Rockchip RK3399 TRM

Revision 1.3
Dec. 2016

Revision History

Date	Revision	Description
2016-12-01	1.3	Update
2016-9-1	1.2	Update
2016-7-28	1.1	Update
2016-5-23	1.0	Update
2016-2-3	0.1	Initial Release

Table of Content

Figure Index

Table Index

Warranty Disclaimer

Rockchip Electronics Co.,Ltd makes no warranty, representation or guarantee (expressed, implied, statutory, or otherwise) by or with respect to anything in this document, and shall not be liable for any implied warranties of non-infringement, merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Information furnished is believed to be accurate and reliable. However, Rockchip Electronics Co.,Ltd assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use.

Rockchip Electronics Co.,Ltd's products are not designed, intended, or authorized for using as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Rockchip Electronics Co.,Ltd's product could create a situation where personal injury or death may occur, should buyer purchase or use Rockchip Electronics Co.,Ltd's products for any such unintended or unauthorized application, buyers shall indemnify and hold Rockchip Electronics Co.,Ltd and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Rockchip Electronics Co.,Ltd was negligent regarding the design or manufacture of the part.

Copyright and Patent Right

Information in this document is provided solely to enable system and software implementers to use Rockchip Electronics Co.,Ltd 's products. There are no expressedand patent or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Rockchip Electronics Co.,Ltd does not convey any license under its copyright and patent rights nor the rights of others.
All copyright and patent rights referenced in this document belong to their respective owners and shall be subject to corresponding copyright and patent licensing requirements.

Trademarks

Rockchip and Rockchip™ logo and the name of Rockchip Electronics Co.,Ltd's products are trademarks of Rockchip Electronics Co.,Ltd. and are exclusively owned by Rockchip Electronics Co.,Ltd. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Confidentiality

The information contained herein (including any attachments) is confidential. The recipient hereby acknowledges the confidentiality of this document, and except for the specific purpose, this document shall not be disclosed to any third party.

Reverse engineering or disassembly is prohibited.

ROCKCHIP ELECTRONICS CO.,LTD. RESERVES THE RIGHT TO MAKE CHANGES IN ITS PRODUCTS OR PRODUCT SPECIFICATIONS WITH THE INTENT TO IMPROVE FUNCTION OR DESIGN AT ANY TIME AND WITHOUT NOTICE AND IS NOT REQUIRED TO UNDATE THIS DOCUMENTATION TO REFLECT SUCH CHANGES.

Copyright © 2016 Rockchip Electronics Co.,Ltd.

Chapter 1 Mobile Storage Host Controller

1

1.1 Overview

The Mobile Storage Host Controller is designed to support Secure Digital memory (SD- max version 3.01) with 1 bits or 4 bits data width, Multimedia Card(MMC-max version 4.51) with 1 bits or 4 bits or 8 bits data width.
The Host Controller is instantiated for SDMMC, SDIO in RK3399. The interface difference between these instances is shown in "Interface Description".
The Host Controller supports following features:

- Bus Interface Features:
  - Support AMBA AHB interface for master and slave
  - Supports internal DMA interface(IDMAC)
    - Supports 16/32-bit data transfers
    - Single-channel; single engine used for Transmit and Receive, which are mutually exclusive
    - Dual-buffer and chained descriptor linked list
    - Each descriptor can transfer up to 4KB of data in chained mode and 8KB of data in dual-buffer mode
    - Programmable burst size for optimal host bus utilization
  - Support combined single FIFO for both transmit and receive operations
  - Support FIFO size of 256x32
  - Support FIFO over-run and under-run prevention by stopping card clock
- Card Interface Features:
  - Support Secure Digital memory protocol commands
  - Support Secure Digital I/O protocol commands
  - Support Multimedia Card protocol commands
  - Support Command Completion Signal and interrupts to host
  - Support CRC generation and error detection
  - Support programmable baud rate
  - Support power management and power switch
  - Support card detection
  - Support write protection
  - Support hardware reset
  - Support SDIO interrupts in 1-bit and 4-bit modes
  - Support 4-bit mode in SDIO3.0
  - Support SDIO suspend and resume operation
  - Support SDIO read wait
  - Support block size of 1 to 65,535 bytes
  - Support 1-bit, 4-bit and 8-bit SDR modes
  - Support 4-bit DDR,8-bit DDR, as defined by SD3.0 and MMC4.41
  - Support boot in 1-bit, 4-bit and 8-bit SDR modes
  - Support Packed Commands, CMD21, CMD49
- Clock Interface Features:
  - Support 0/90/180/270-degree phase shift operation for sample clock (cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock(cclk_in) respectively
  - Support phase tuning using delay line for sample clock(cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock (cclk_in) respectively. The max number of delay element number is 256

1.2 Block Diagram

The Host Controller consists of the following main functional blocks.

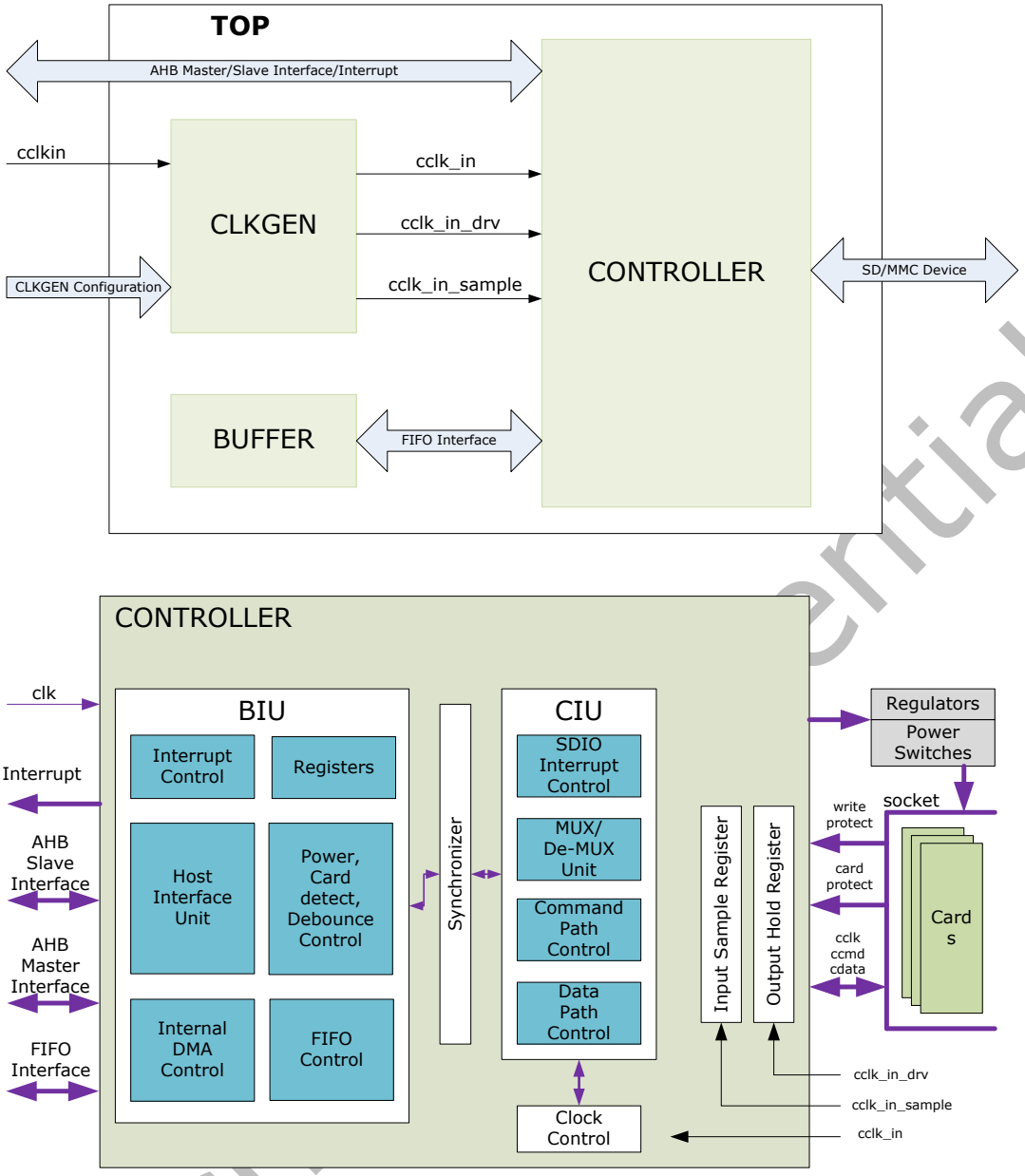


Fig. 1-1 Host Controller Block Diagram

- Clock Generate Unit(CLKGEN): generates card interface clock cclk_in/ cclk_sample/cclk_drv based on cclkin and configuration information.
- Asynchronous dual-port memory(BUFFER): Uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, and the second port is connected to the card clock.
- Bus Interface Unit (BIU): Provides AMBA AHB interfaces for register and data read/writes.
- Card Interface Unit (CIU): Takes care of the SD/MMC protocols and provides clock management.

1.3 Function Description

1.3.1 Bus Interface Unit

The Bus Interface Unit provides the following functions:

- Host interface
- Interrupt control
- Register access
- External FIFO access
- Power control and card detection

1. Host Interface Unit

The Host Interface Unit is an AHB slave interface, which provides the interface between the SD/MMC card and the host bus.

2. Register Unit

The register unit is part of the bus interface unit; it provides read and write access to the registers.

All registers reside in the Bus Interface Unit clock domain. When a command is sent to a card by setting the start_bit, which is bit[31] of the CMD register, all relevant registers needed for the CIU operation are transferred to the CIU block. During this time, the registers that are transferred from the BIU to the CIU should not be written. The software should wait for the hardware to clear the start bit before writing to these registers again. The register unit has a hardware locking feature to prevent illegal writes to registers. The lock is necessary in order to avoid metastability violations, both because the host and card clock domains are different and to prevent illegal software operations.

Once a command start is issued by setting the start_bit of the CMD register, the following registers cannot be reprogrammed until the command is accepted by the card interface unit:

- CMD – Command
- CMDARG – Command Argument
- BYTCNT – Byte Count
- BLKSIZ – Block Size
- CLKDIV – Clock Divider
- CLKENA – Clock Enable
- CLKSRC – Clock Source
- TMOUT – Timeout
- CTYPE – Card Type

The hardware resets the start_bit once the CIU accepts the command. If a host write to any of these registers is attempted during this locked time, then the write is ignored and the hardware lock error bit is set in the raw interrupt status register. Additionally, if the interrupt is enabled and not masked for a hardware lock error, then an interrupt is sent to the host.

When the Card Interface Unit is in an idle state, it typically takes the following number of clocks for the command handshake, where clk is the BIU clock and cclk_in is the CIU clock: 3 (clk) + 3 (cclk_in)

Once a command is accepted, you can send another command to the CIU-which has a one-deep command queue-under the following conditions:

- If the previous command was not a data transfer command, the new command is sent to the SD/MMC card once the previous command completes.
- If the previous command is a data transfer command and if wait_prvdata_complete (bit[13]) of the Command register is set for the new command, the new command is sent to the SD/MMC card only when the data transfer completes.
- If the wait_prvdata_complete is 0, then the new command is sent to the SD/MMC card as soon as the previous command is sent. Typically, you should use this only to stop or abort a previous data transfer or query the card status in the middle of a data transfer.

3. Interrupt Controller Unit

The interrupt controller unit generates an interrupt that depends on the controller raw interrupt status, the interrupt-mask register, and the global interrupt-enable register bit. Once an interrupt condition is detected, it sets the corresponding interrupt bit in the raw interrupt status register. The raw interrupt status bit stays on until the software clears the bit by writing a 1 to the interrupt bit; a 0 leaves the bit untouched.

The interrupt port, int, is an active-high, level-sensitive interrupt. The interrupt port is active only when any bit in the raw interrupt status register is active, the corresponding interrupt mask bit is 1, and the global interrupt enable bit is 1. The interrupt port is registered in order to avoid any combinational glitches.

The int_enable is reset to 0 on power-on, and the interrupt mask bits are set to 32'h0, which masks all the interrupts.

Notes:

Before enabling the interrupt, it is always recommended that you write 32'hffff_ffff to the raw interrupt status register in order to clear any pending unserviced interrupts. When clearing interrupts during normal

operation, ensure that you clear only the interrupt bits that you serviced.
The SDIO Interrupts, Receive FIFO Data Request (RXDR), and Transmit FIFO Data Request (TXDR) are set by level-sensitive interrupt sources. Therefore, the interrupt source should be first cleared before you can clear the interrupt bit of the Raw Interrupt register. For example, on seeing the Receive FIFO Data Request (RXDR) interrupt, the FIFO should be emptied so that the "FIFO count greater than the RX-Watermark" condition, which triggers the interrupt, becomes inactive. The rest of the interrupts are triggered by a single clock-pulse-width source.

Table 1-1 Bits in Interrupt Status Register

Bits	Interrupt	Description
24	sdio_interrupt	Interrupt from SDIO card. In MMC-Ver3.3-only mode, these bits are always 0
16	Card no-busy	If card exit busy status, the interrupt happened
15	End Bit Error (read) /Write no CRC (EBE)	Error in end-bit during read operation, or no data CRC or negative CRC received during write operation. *Notes: For MMC CMD19, there may be no CRC status returned by the card. Hence, EBE is set for CMD19. The application should not treat this as an error.*
14	Auto Command Done (ACD)	Stop/abort commands automatically sent by card unit and not initiated by host; similar to Command Done (CD) interrupt.
13	Start Bit Error (SBE)	Error in data start bit when data is read from a card. In 4-bit mode, if all data bits do not have start bit, then this error is set.
12	Hardware Locked write Error (HLE)	During hardware-lock period, write attempted to one of locked registers.
11	FIFO Underrun/ Overrun Error (FRUN)	Host tried to push data when FIFO was full, or host tried to read data when FIFO was empty. Typically this should not happen, except due to error in software. Card unit never pushes data into FIFO when FIFO is full, and pop data when FIFO is empty.
10	Data Starvation by Host Timeout (HTO)	To avoid data loss, card clock out (cclk_out) is stopped if FIFO is empty when writing to card, or FIFO is full when reading from card. Whenever card clock is stopped to avoid data loss, data-starvation timeout counter is started with data-timeout value. This interrupt is set if host does not fill data into FIFO during write to card, or does not read from FIFO during read from card before timeout period. Even after timeout, card clock stays in stopped state, with CIU state machines waiting. It is responsibility of host to push or pop data into FIFO upon interrupt, which automatically restarts cclk_out and card state machines. Even if host wants to send stop/abort command, it still needs to ensure it has to push or pop FIFO so that clock starts in order for stop/abort command to send on cmd signal along with data that is sent or received on data line.
9	Data Read Timeout (DRTO)	Data timeout occurred. Data Transfer Over (DTO) also set if data timeout occurs.
8	Response Timeout (RTO)	Response timeout occurred. Command Done (CD) also set if response timeout occurs. If command involves data transfer and when response times out, no data transfer is attempted by Host Controller.
7	Data CRC Error (DCRC)	Received Data CRC does not match with locally-generated CRC in CIU.
6	Response CRC Error	Response CRC does not match with locally-generated

Bits	Interrupt	Description
	(RCRC)	CRC in CIU.
5	Receive FIFO Data Request (RXDR)	Interrupt set during read operation from card when FIFO level is greater than Receive-Threshold level.
4	Transmit FIFO Data Request (TXDR)	Interrupt set during write operation to card when FIFO level reaches less than or equal to Transmit-Threshold level.
3	Data Transfer Over (DTO)	Data transfer completed, even if there is Start Bit Error or CRC error. This bit is also set when "read data-timeout" occurs. *Notes: DTO bit is set at the end of the last data block, even if the device asserts MMC busy after the last data block.*
2	Command Done(CD)	Command sent to card and got response from card, even if Response Error or CRC error occurs. Also set when response timeout occurs
1	Response Error (RE)	Error in received response set if one of following occurs: ● Transmission bit != 0 ● Command index mismatch ● End-bit != 1
0	Card-Detect (CDT)	When card inserted or removed, this interrupt occurs. Software should read card-detect register (CDETECT, 0x50) to determine current card status.

4. FIFO Controller Unit

The FIFO controller interfaces the external FIFO to the host interface and the card controller unit. When FIFO overrun and under-run conditions occur, the card clock stops in order to avoid data loss.

The FIFO uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, clk, and the second port is connected to the card clock, cclk_in.

Notes: The FIFO controller does not support simultaneous read/write access from the same port. For debugging purposes, the software may try to write into the FIFO and read back the data; results are indeterminate, since the design does not support read/write access from the same port.

5. Power Control and Card Detection Unit

The register unit has registers that control the power. Power to each card can be selectively turned on or off.

The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value.

On power-on, the controller should read in the card_detect port and store the value in the memory. Upon receiving a card-detect interrupt, it should again read the card_detect port and XOR with the previous card-detect status to find out which card has interrupted. If more than one card is simultaneously removed or inserted, there is only one card-detect interrupt; the XOR value indicates which cards have been disturbed. The memory should be updated with the new card-detect value.

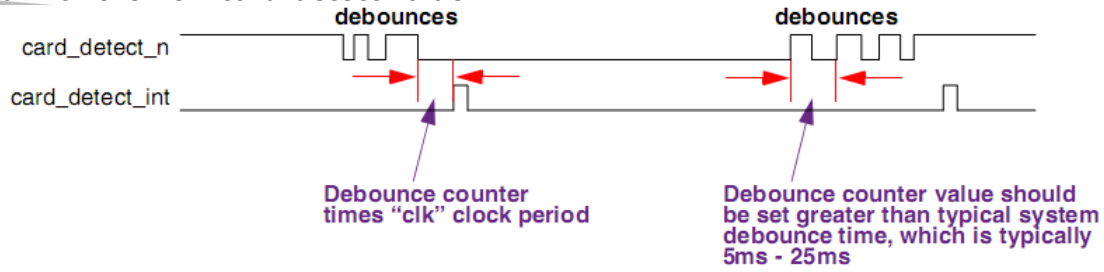


Fig. 1-2 SD/MMC Card-Detect Signal

6. DMA Interface Unit

DMA signals interface the Host Controller to an external DMA controller to reduce the software overhead during FIFO data transfers. The DMA request/acknowledge handshake is

used for only data transfers. The DMA interface provides a connection to the DMA Controller.
On seeing the DMA request, the DMA controller initiates accesses through the host interface to read or write into the data FIFO. The Host Controller has FIFO transmit/receive watermark registers that you can set, depending on system latency. The DMA interface asserts the request in the following cases:

- Read from a card when the data FIFO word count exceeds the Rx-Watermark level
- Write to a card when the FIFO word count is less than or equal to the Tx-Watermark level

When the DMA interface is enabled, you can use normal host read/write to access the data FIFO.

1.3.2 Card Interface Unit

The Card Interface Unit (CIU) interfaces with the Bus Interface Unit (BIU) and the devices. The host writes command parameters to the BIU control registers, and these parameters are then passed to the CIU. Depending on control register values, the CIU generates SD/MMC command and data traffic on a selected card bus according to SD/MMC protocol. The Host Controller accordingly controls the command and data path.
The following software restrictions should be met for proper CIU operation:

- Only one data transfer command can be issued at a time.
- During an open-ended card write operation, if the card clock is stopped because the FIFO is empty, the software must first fill the data into the FIFO and start the card clock. It can then issue only a stop/abort command to the card.
- When issuing card reset commands (CMD0, CMD15 or CMD52_reset) while a card data transfer is in progress, the software must set the stop_abort_cmd bit in the Command register so that the Host Controller can stop the data transfer after issuing the card reset command.
- When the data end bit error is set in the RINTSTS register, the Host Controller does not guarantee SDIO interrupts. The software should ignore the SDIO interrupts and issue the stop/abort command to the card, so that the card stops sending the read data.
- If the card clock is stopped because the FIFO is full during a card read, the software should read at least two FIFO locations to start the card clock.

The CIU block consists of the following primary functional blocks:

- Command path
- Data path
- SDIO interrupt control
- Clock control
- Mux/demux unit

1. Command Path
The command path performs the following functions:

- Loads clock parameters
- Loads card command parameters
- Sends commands to card bus (ccmd_out line)
- Receives responses from card bus (ccmd_in line)
- Sends responses to BIU
- Drives the P-bit on command line

A new command is issued to the Host Controller by programming the BIU registers and setting the start_cmd bit in the Command register. The BIU asserts start_cmd, which indicates that a new command is issued to the SD/MMC device. The command path loads this new command (command, command argument, timeout) and sends acknowledge to the BIU by asserting cmd_taken.
Once the new command is loaded, the command path state machine sends a command to the device bus-including the internally generated CRC7-and receives a response, if any. The state machine then sends the received response and signals to the BIU that the command is done, and then waits for eight clocks before loading a new command.
Load Command Parameters
One of the following commands or responses is loaded in the command path:

- New command from BIU – When start_cmd is asserted, then the start_cmd bit is set in

the Command register.

- Internally-generated auto-stop command – When the data path ends, the stop command request is loaded.
- IRQ response with RCA 0x000 – When the command path is waiting for an IRQ response from the MMC card and a "send irq response" request is signaled by the BIU, then the send_irq_response bit is set in the control register.

Loading a new command from the BIU in the command path depends on the following Command register bit settings:

- update_clock_registers_only – If this bit is set in the Command register, the command path updates only the clock enable, clock divider, and clock source registers. If this bit is not set, the command path loads the command, command argument, and timeout registers; it then starts processing the new command.
- wait_prvdata_complete – If this bit is set, the command path loads the new command under one of the following conditions:
  - Immediately, if the data path is free (that is, there is no data transfer in progress), or if an open-ended data transfer is in progress (byte_count = 0).
  - After completion of the current data transfer, if a predefined data transfer is in progress.

Send Command and Receive Response

Once a new command is loaded in the command path, update_clock_registers_only bit is unset – the command path state machine sends out a command on the device bus; the command path state machine is illustrated in following figure.

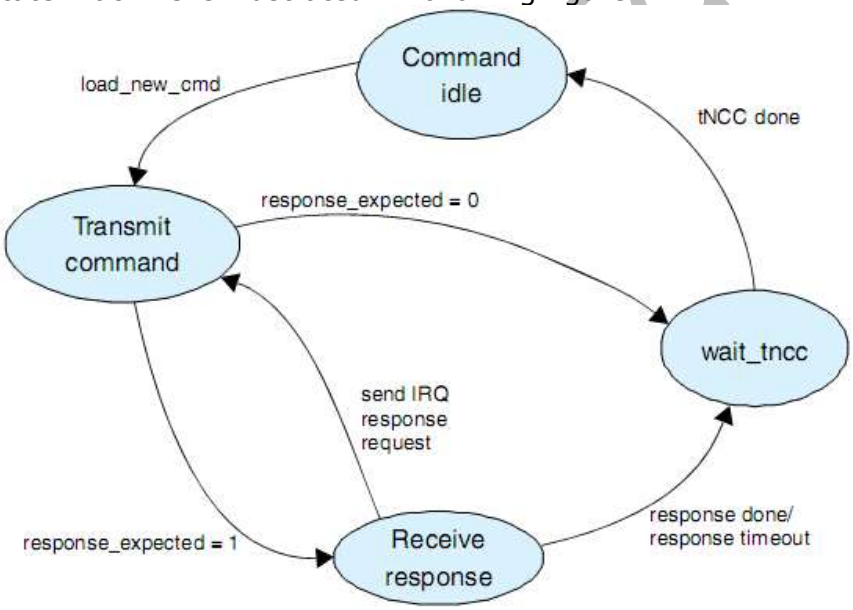


Fig. 1-3 Host Controller Command Path State Machine

The command path state machine performs the following functions, according to Command register bit values:

- send_initialization – Initialization sequence of 80 clocks is sent before sending the command.
- response_expected – Response is expected for the command. After the command is sent out, the command path state machine receives a 48-bit or 136-bit response and sends it to the BIU. If the start bit of the card response is not received within the number of clocks programmed in the timeout register, then the response timeout and command done bit is set in the Raw Interrupt Status register as a signal to the BIU. If the response-expected bit is not set, the command path sends out a command and signals a response done to the BIU; that is, the command done bit is set in the Raw Interrupt Status register.
- response_length – If this bit is set, a 136-bit response is received; if it is not set, a 48-bit response is received.
- check_response_crc – If this bit is set, the command path compares CRC7 received in the response with the internally-generated CRC7. If the two do not match, the

response CRC error is signaled to the BIU; that is, the response CRC error bit is set in the Raw Interrupt Status register.

Send Response to BIU

If the response_expected bit is set in the Command register, the received response is sent to the BIU. The Response0 register is updated for a short response, and the Response3, Response2, Response1, and Response0 registers are updated on a long response, after which the Command Done bit is set. If the response is for an auto_stop command sent by the CIU, the response is saved in the Response1 register, after which the Auto Command Done bit is set.

Additionally, the command path checks for the following:

- Transmission bit = 0
- Command index matches command index of the sent command
- End bit = 1 in received card response

The command index is not checked for a 136-bit response or if the check_response_crc bit is unset. For a 136-bit response and reserved CRC 48-bit responses, the command index is reserved-that is, 111111.

Polling Command Completion Signal

The device generates the Command Completion Signal in order to notify the host controller of the normal command completion or command termination.

Command Completion Signal Detection and Interrupt to Host Processor

If the ccs_expected bit is set in the Command register, the Command Completion Signal (CCS) from the device is indicated by setting the Data Transfer Over (DTO) bit in the RINTSTS register. The Host Controller generates a Data Transfer Over (DTO) interrupt if this interrupt is not masked.

Command Completion Signal Timeout

If the command expects a CCS from the device—if the ccs_expected bit is set in the Command register—the command state machine waits for the CCS and remains in a wait_CCSS state. If the device fails to send out the CCS, the host software should implement a timeout mechanism to free the command and data path. The host controller does not implement a hardware timer; it is the responsibility of the host software to maintain a software timer.

In the event of a CCS timeout, the host should issue a CCSD by setting the send_ccsd bit in the CTRL register. The host controller command state machine sends the CCSD to the device and exits to an idle state. After sending the CCSD, the host should also send a CMD12 to the device in order to abort the outstanding command.

Send Command Completion Signal Disable

If the send_ccsd bit is set in the CTRL register, the host sends a Command Completion Signal Disable (CCSD) pattern on the CMD line. The host can send the CCSD while waiting for the CCS or after a CCS timeout happens.

After sending the CCSD pattern, the host sets the Command Done (CD) bit in RINTSTS and also generates an interrupt to the host if the Command Done interrupt is not masked.

2. Data Path

The data path block pops the data FIFO and transmits data on cdata_out during a write data transfer, or it receives data on cdata_in and pushes it into the FIFO during a read data transfer. The data path loads new data parameters—that is, data expected, read/write data transfer, stream/block transfer, block size, byte count, card type, timeout registers—whenever a data transfer command is not in progress.

If the data_expected bit is set in the Command register, the new command is a data transfer command and the data path starts one of the following:

- Transmit data if the read/write bit = 1
- Data receive if read/write bit = 0

Data Transmit

The data transmit state machine, illustrated in following figure, starts data transmission two clocks after a response for the data write command is received; this occurs even if the command path detects a response error or response CRC error. If a response is not received from the card because of a response timeout, data is not transmitted. Depending upon the value of the transfer_mode bit in the Command register, the data transmit state

machine puts data on the card data bus in a stream or in block(s).

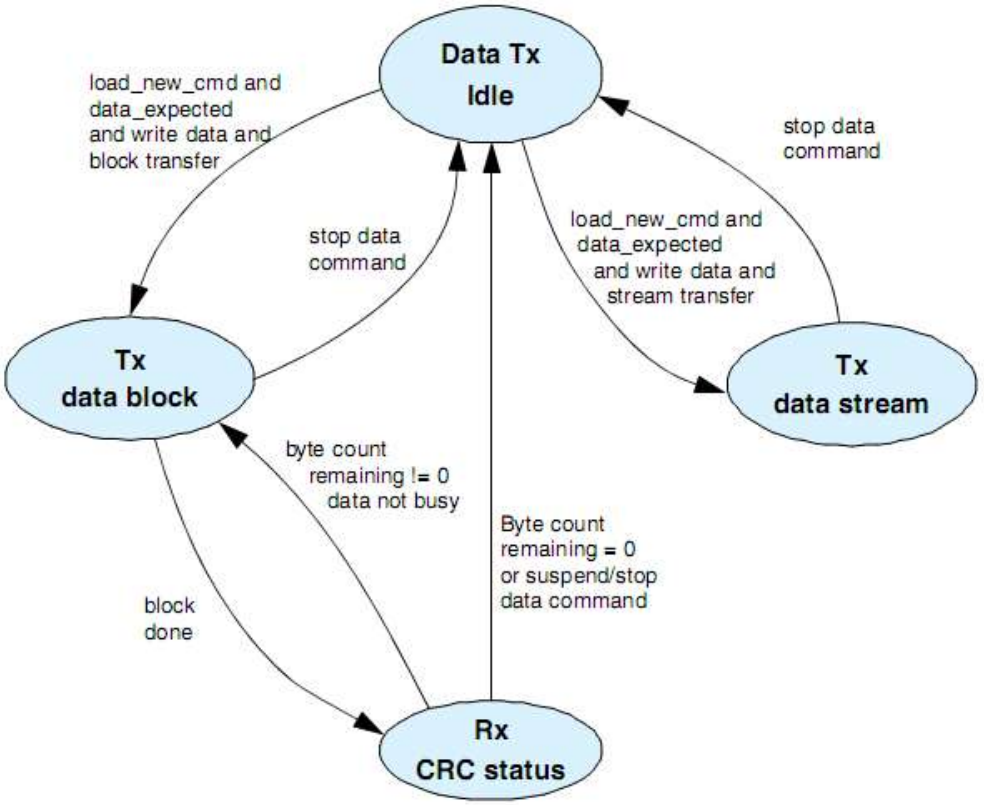


Fig. 1-4 Host Controller Data Transmit State Machine

Stream Data Transmit

If the transfer_mode bit in the Command register is set to 1, it is a stream-write data transfer. The data path pops the FIFO from the BIU and transmits in a stream to the card data bus. If the FIFO becomes empty, the card clock is stopped and restarted once data is available in the FIFO.

If the byte_count register is programmed to 0, it is an open-ended stream-write data transfer. During this data transfer, the data path continuously transmits data in a stream until the host software issues a stop command. A stream data transfer is terminated when the end bit of the stop command and end bit of the data match over two clocks.

If the byte_count register is programmed with a non-zero value and the send_auto_stop bit is set in the Command register, the stop command is internally generated and loaded in the command path when the end bit of the stop command occurs after the last byte of the stream write transfer matches.

This data transfer can also terminate if the host issues a stop command before all the data bytes are transferred to the card bus.

Single Block Data

If the transfer_mode bit in the Command register is set to 0 and the byte_count register value is equal to the value of the block_size register, a single-block write-data transfer occurs. The data transmit state machine sends data in a single block, where the number of bytes equals the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set for a 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted for 1, 4, or 8 data lines, respectively.

After a single data block is transmitted, the data transmit state machine receives the CRC status from the card and signals a data transfer to the BIU; this happens when the data-transfer-over bit is set in the RINTSTS register.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register.

Additionally, if the start bit of the CRC status is not received by two clocks after the end of the data block, a CRC status start bit error is signaled to the BIU by setting the write-no-

CRC bit in the RINTSTS register.

Multiple Block Data

A multiple-block write-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value in the byte_count register is not equal to the value of the block_size register. The data transmit state machine sends data in blocks, where the number of bytes in a block equals the block size, including the internally-generated CRC16. If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted on 1, 4, or 8 data lines, respectively.

After one data block is transmitted, the data transmit state machine receives the CRC status from the card. If the remaining byte_count becomes 0, the data path signals to the BIU that the data transfer is done; this happens when the data-transfer-over bit is set in the RINTSTS register.

If the remaining data bytes are greater than 0, the data path state machine starts to transmit another data block.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register, and continues further data transmission until all the bytes are transmitted.

Additionally, if the CRC status start bit is not received by two clocks after the end of a data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the RINTSTS register; further data transfer is terminated.

If the send_auto_stop bit is set in the Command register, the stop command is internally generated during the transfer of the last data block, where no extra bytes are transferred to the card. The end bit of the stop command may not exactly match the end bit of the CRC status in the last data block.

If the block size is less than 4, 16, or 32 for card data widths of 1 bit, 4 bits, or 8 bits, respectively, the data transmit state machine terminates the data transfer when all the data is transferred, at which time the internally generated stop command is loaded in the command path.

If the byte_count is 0 – the block size must be greater than 0 – it is an open-ended block transfer. The data transmit state machine for this type of data transfer continues the block-write data transfer until the host software issues a stop or abort command.

Data Receive

The data-receive state machine, illustrated in following figure, receives data two clock cycles after the end bit of a data read command, even if the command path detects a response error or response CRC error. If a response is not received from the card because a response timeout occurs, the BIU does not receive a signal that the data transfer is complete; this happens if the command sent by the Host Controller is an illegal operation for the card, which keeps the card from starting a read data transfer.

If data is not received before the data timeout, the data path signals a data timeout to the BIU and an end to the data transfer done. Based on the value of the transfer_mode bit in the Command register, the data-receive state machine gets data from the card data bus in a stream or block(s).

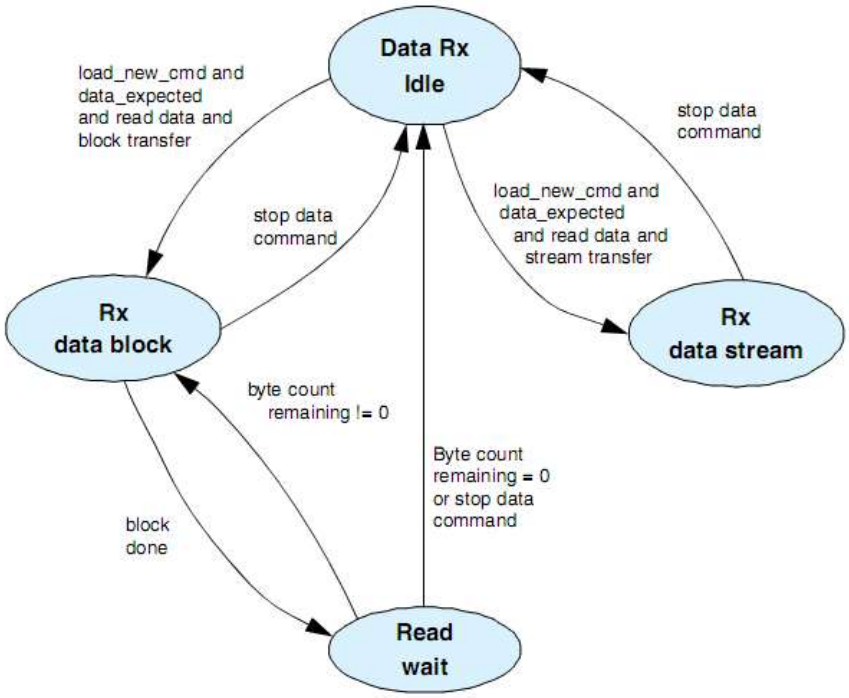


Fig. 1-5 Host Controller Data Receive State Machine

Stream Data Read

A stream-read data transfer occurs if the transfer_mode bit in the Command register equals 1, at which time the data path receives data from the card and pushes it to the FIFO. If the FIFO becomes full, the card clock stops and restarts once the FIFO is no longer full.

An open-ended stream-read data transfer occurs if the byte_count register equals 0. During this type of data transfer, the data path continuously receives data in a stream until the host software issues a stop command. A stream data transfer terminates two clock cycles after the end bit of the stop command.

If the byte_count register contains a non-zero value and the send_auto_stop bit is set in the Command register, a stop command is internally generated and loaded into the command path, where the end bit of the stop command occurs after the last byte of the stream data transfer is received. This data transfer can terminate if the host issues a stop or abort command before all the data bytes are received from the card.

Single-Block Data Read

A single-block read-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is equal to the value of the block_size register. When a start bit is received before the data times out, data bytes equal to the block size and CRC16 are received and checked with the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively. If there is a CRC16 mismatch, the data path signals a data CRC error to the BIU. If the received end bit is not 1, the BIU receives an end-bit error.

Multiple-Block Data Read

If the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is not equal to the value of the block_size register, it is a multiple-block read-data transfer. The data-receive state machine receives data in blocks, where the number of bytes in a block is equal to the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively.

After a data block is received, if the remaining byte_count becomes 0, the data path signals a data transfer to the BIU.

If the remaining data bytes are greater than 0, the data path state machine causes another data block to be received. If CRC16 of a received data block does not match the internally-generated CRC16, a data CRC error to the BIU and data reception continue further data transmission until all bytes are transmitted.
Additionally, if the end of a received data block is not 1, data on the data path signals terminate the bit error to the CIU and the data-receive state machine terminates data reception, waits for data timeout, and signals to the BIU that the data transfer is complete.
If the send_auto_stop bit is set in the Command register, the stop command is internally generated when the last data block is transferred, where no extra bytes are transferred from the card; the end bit of the stop command may not exactly match the end bit of the last data block.
If the requested block size for data transfers to cards is less than 4, 16, or 32 bytes for 1-bit, 4-bit, or 8-bit data transfer modes, respectively, the data-transmit state machine terminates the data transfer when all data is transferred, at which point the internally-generated stop command is loaded in the command path. Data received from the card after that are then ignored by the data path.
If the byte_count is 0—the block size must be greater than 0—it is an open-ended block transfer. For this type of data transfer, the data-receive state machine continues the block-read data transfer until the host software issues a stop or abort command.

Auto-Stop

The Host Controller internally generates a stop command and is loaded in the command path when the send_auto_stop bit is set in the Command register.
The software should set the send_auto_stop bit according to details listed in following table.

Table 1-2 Auto-Stop Generation

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
MMC	Stream read	0	No	Open-ended stream
MMC	Stream read	>0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stream
MMC	Stream write	>0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	>0	No	Byte count =0 is illegal
MMC	Single-block write	>0	No	Byte count =0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	>0	Yes①	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	>0	Yes①	Pre-defined multiple block
SDMEM	Single-block read	>0	No	Byte count =0 is illegal
SDMEM	Single-block write	>0	No	Byte count =0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	>0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	>0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	>0	No	Byte count =0 is illegal
SDIO	Single-block write	>0	No	Byte count =0 illegal
SDIO	Multiple-block read	0	No	Open-ended multiple block

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
SDIO	Multiple-block read	>0	No	Pre-defined multiple block
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block write	>0	No	Pre-defined multiple block

①:*The condition under which the transfer mode is set to block transfer and byte_count is equal to block size is treated as a single-block data transfer command for both MMC and SD cards. If byte_count = n*block_size (n = 2, 3, …), the condition is treated as a predefined multiple-block data transfer command. In the case of an MMC card, the host software can perform a predefined data transfer in two ways: 1) Issue the CMD23 command before issuing CMD18/CMD25 commands to the card – in this case, issue MD18/CMD25 commands without setting the send_auto_stop bit. 2) Issue CMD18/CMD25 commands without issuing CMD23 command to the card, with the send_auto_stop bit set. In this case, the multiple-block data transfer is terminated by an internally-generated auto-stop command after the programmed byte count.*

The following list conditions for the auto-stop command.

- Stream read for MMC card with byte count greater than 0 – The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent out when the last byte of data is read from the card and no extra data byte is received. If the byte count is less than 6 (48 bits), a few extra data bytes are received from the card before the end bit of the stop command is sent.
- Stream write for MMC card with byte count greater than 0 - The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent when the last byte of data is transmitted on the card bus and no extra data byte is transmitted. If the byte count is less than 6 (48 bits), the data path transmits the data last in order to meet the above condition.
- Multiple-block read memory for SD card with byte count greater than 0 – If the block size is less than 4 (single-bit data bus), 16 (4-bit data bus), or 32 (8-bit data bus), the auto-stop command is loaded in the command path after all the bytes are read. Otherwise, the top command is loaded in the command path so that the end bit of the stop command is sent after the last data block is received.
- Multiple-block write memory for SD card with byte count greater than 0 – If the block size is less than 3 (single-bit data bus), 12 (4-bit data bus), or 24 (8-bit data bus), the auto-stop command is loaded in the command path after all data blocks are transmitted. Otherwise, the stop command is loaded in the command path so that the end bit of the stop command is sent after the end bit of the CRC status is received.
- Precaution for host software during auto-stop – Whenever an auto-stop command is issued, the host software should not issue a new command to the SD/MMC device until the auto-stop is sent by the Host Controller and the data transfer is complete. If the host issues a new command during a data transfer with the auto-stop in progress, an auto-stop command may be sent after the new command is sent and its response is received; this can delay sending the stop command, which transfers extra data bytes. For a stream write, extra data bytes are erroneous data that can corrupt the card data. If the host wants to terminate the data transfer before the data transfer is complete, it can issue a stop or abort command, in which case the Host Controller does not generate an auto-stop command.

3. Non-Data Transfer Commands that Use Data Path

Some non-data transfer commands (non-read/write commands) also use the data path. Following table lists the commands and register programming requirements for them.

Table 1-3 Non-data Transfer Commands and Requirements

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
Command register programming						
cmd_index	6'h1B	6'h1E	6'h2A	6'h0D	6'h16	6'h33
response_expect	1	1	1	1	1	1
rResponse_length	0	0	0	0	0	0

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
check_response_crc	1	1	1	1	1	1
data_expected	1	1	1	1	1	1
read/write	1	0	1	0	0	0
transfer_mode	0	0	0	0	0	0
send_auto_stop	0	0	0	0	0	0
wait_prevdata_complete	0	0	0	0	0	0
stop_abort_cmd	0	0	0	0	0	0
Command Argument register programming						
	stuff bits	32-bit write protect data address	stuff bits	stuff bits	stuff bits	stuff bits
Block Size register programming						
	16	4	Num_bytes①	64	4	8
Byte Count register programming						
	16	4	Num_bytes①	64	4	8

①: *Num_bytes = No. of bytes specified as per the lock card data structure (Refer to the SD specification and the MMC specification)*

4. SDIO Interrupt Control

Interrupts for SD cards are reported to the BIU by asserting an interrupt signal for two clock cycles. SDIO cards signal an interrupt by asserting cdata_in low during the interrupt period; an interrupt period for the selected card is determined by the interrupt control state machine. An interrupt period is always valid for non-active or non-selected cards, and 1-bit data mode for the selected card. An interrupt period for a wide-bus active or selected card is valid for the following conditions:

- Card is idle
- Non-data transfer command in progress
- Third clock after end bit of data block between two data blocks
- From two clocks after end bit of last data until end bit of next data transfer command

Bear in mind that, in the following situations, the controller does not sample the SDIO interrupt of the selected card when the card data width is 4 bits. Since the SDIO interrupt is level-triggered, it is sampled in a further interrupt period and the host does not lose any SDIO interrupt from the card.

- Read/Write Resume – The CIU treats the resume command as a normal data transfer command. SDIO interrupts during the resume command are handled similarly to other data commands. According to the SDIO specification, for the normal data command the interrupt period ends after the command end bit of the data command; for the resume command, it ends after the response end bit. In the case of the resume command, the Controller stops the interrupt sampling period after the resume command end bit, instead of stopping after the response end bit of the resume command.
- Suspend during read transfer – If the read data transfer is suspended by the host, the host sets the abort_read_data bit in the controller to reset the data state machine. In the CIU, the SDIO interrupts are handled such that the interrupt sampling starts after the abort_read_data bit is set by the host. In this case the controller does not sample SDIO interrupts between the period from response of the suspend command to setting the abort_read_data bit, and starts sampling after setting the abort_read_data bit.

5. Clock Control

The clock control block provides different clock frequencies required for SD/MMC cards. The cclk_in signal is the source clock (cclk_in >= card max operating frequency) for clock divider of the clock control block. This source clock (cclk_in) is used to generate different card clock frequencies (cclk_out). The card clock can have different clock frequencies, since the card can be a low-speed card or a full-speed card. The Host Controller provides one clock signal (cclk_out).

The clock frequency of a card depends on the following clock control registers:

- Clock Divider register – Internal clock dividers are used to generate different clock frequencies required for card. The division factor for each clock divider can be programmed by writing to the Clock Divider register. The clock divider is an 8-bit value that provides a clock division factor from 1 to 510; a value of 0 represents a clock-divider bypass, a value of 1 represents a divide by 2, a value of 2 represents a divide by 4, and so on.
- Clock Control register – cclk_out can be enabled or disabled for each card under the following conditions:
  - clk_enable – cclk_out for a card is enabled if the clk_enable bit for a card in the Clock Control register is programmed (set to 1) or disabled (set to 0).
  - Low-power mode – Low-power mode of a card can be enabled by setting the low-power mode bit of the Clock Control register to 1. If low-power mode is enabled to save card power, the cclk_out is disabled when the card is idle for at least 8 card clock cycles. It is enabled when a new command is loaded and the command path goes to a non-idle state.

Additionally, cclk_out is disabled when an internal FIFO is full – card read (no more data can be received from card) – or when the FIFO is empty – card write (no data is available for transmission). This helps to avoid FIFO overrun and underrun conditions. It is used by the command and data path to qualify cclk_in for driving outputs and sampling inputs at the programmed clock frequency for the selected card, according to the Clock Divider and Clock Source register values.

Under the following conditions, the card clock is stopped or disabled, along with the active clk_en, for the selected card:

- Clock can be disabled by writing to Clock Enable register (clk_en bit = 1).
- If low-power mode is selected and card is idle, or not selected for 8 clocks.
- FIFO is full and data path cannot accept more data from the card and data transfer is incomplete –to avoid FIFO overrun.
- FIFO is empty and data path cannot transmit more data to the card and data transfer is incomplete – to avoid FIFO underrun.

6. Error Detection

- Response
  - Response timeout – Response expected with response start bit is not received within programmed number of clocks in timeout register.
  - Response CRC error – Response is expected and check response CRC requested; response CRC7 does not match with the internally-generated CRC7.
  - Response error – Response transmission bit is not 0, command index does not match with the command index of the send command, or response end bit is not 1.
- Data transmit
  - No CRC status – During a write data transfer, if the CRC status start bit is not received two clocks after the end bit of the data block is sent out, the data path does the following:
    - Signals no CRC status error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to the BIU
  - Negative CRC – If the CRC status received after the write data block is negative (that is, not 010), a data CRC error is signaled to the BIU and further data transfer is continued.
  - Data starvation due to empty FIFO – If the FIFO becomes empty during a write data transmission, or if the card clock is stopped and the FIFO remains empty for data timeout clocks, then a data-starvation error is signaled to the BIU and the data path continues to wait for data in the FIFO.
- Data receive
  - Data timeout – During a read-data transfer, if the data start bit is not received before the number of clocks that were programmed in the timeout register, the data path does the following:
    - Signals data-timeout error to the BIU
    - Terminates further data transfer

- Signals data transfer done to BIU
- Data start bit error – During a 4-bit or 8-bit read-data transfer, if the all-bit data line does not have a start bit, the data path signals a data start bit error to the BIU and waits for a data timeout, after which it signals that the data transfer is done.
- Data CRC error – During a read-data-block transfer, if the CRC16 received does not match with the internally generated CRC16, the data path signals a data CRC error to the BIU and continues further data transfer.
- Data end-bit error – During a read-data transfer, if the end bit of the received data is not 1, the data path signals an end-bit error to the BIU, terminates further data transfer, and signals to the BIU that the data transfer is done.
- Data starvation due to FIFO full – During a read data transmission and when the FIFO becomes full, the card clock is stopped. If the FIFO remains full for data timeout clocks, a data starvation error is signaled to the BIU (Data Starvation by Host Timeout bit is set in RINTSTS Register) and the data path continues to wait for the FIFO to start to empty.

1.3.3 Internal Direct Memory Access Controller (IDMAC)

The Internal Direct Memory Access Controller (IDMAC) has a Control and Status Register (CSR) and a single Transmit/Receive engine, which transfers data from host memory to the device port and vice versa. The controller utilizes a descriptor to efficiently move data from source to destination with minimal Host CPU intervention. You can program the controller to interrupt the Host CPU in situations such as data Transmit and Receive transfer completion from the card, as well as other normal or error conditions.
The IDMAC and the Host driver communicate through a single data structure. CSR addresses 0x80 to 0x98 are reserved for host programming.
The IDMAC transfers the data received from the card to the Data Buffer in the Host memory, and it transfers Transmit data from the Data Buffer in the Host memory to the FIFO. Descriptors that reside in the Host memory act as pointers to these buffers.
A data buffer resides in physical memory space of the Host and consists of complete data or partial data. Buffers contain only data, while buffer status is maintained in the descriptor. Data chaining refers to data that spans multiple data buffers. However, a single descriptor cannot span multiple data.
A single descriptor is used for both reception and transmission. The base address of the list is written into Descriptor List Base Address Register (DBADDR @0x88). A descriptor list is forward linked. The Last Descriptor can point back to the first entry in order to create a ring structure. The descriptor list resides in the physical memory address space of the Host. Each descriptor can point to a maximum of two data buffers.

1. IDMAC CSR Access

When an IDMAC is introduced, an additional CSR space resides in the IDMAC that controls the IDMAC functionality. The host accesses the new CSR space in addition to the existing control register set in the BIU. The IDMAC CSR primarily contains descriptor information. For a write operation to the CSR, the respective CSR logic of the IDMAC and BIU decodes the address before accepting. For a read operation from the CSR, the appropriate CSR read path is enabled.
You can enable or disable the IDMAC operation by programming bit[25] in the CTRL register of the BIU. This allows the data transfer by accessing the slave interface on the AMBA bus if the IDMAC is present but disabled. When IDMAC is enabled, the FIFO cannot be accessed through the slave interface.

2. Descriptors

- Descriptor structures

The IDMAC uses these types of descriptor structures:

- Dual-Buffer Structure – The distance between two descriptors is determined by the Skip Length value programmed in the Descriptor Skip Length (DSL) field of the Bus Mode Register (BMOD @0x80).

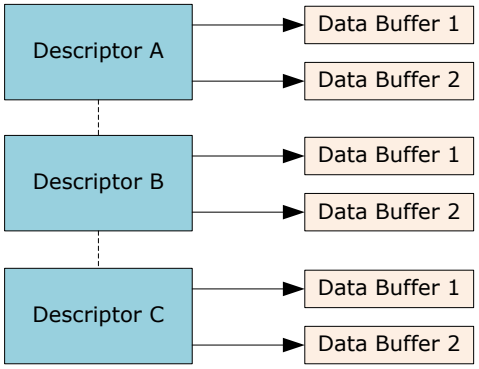


Fig. 1-6 Dual-Buffer Descriptor Structure

- Chain Structure – Each descriptor points to a unique buffer and the next descriptor.

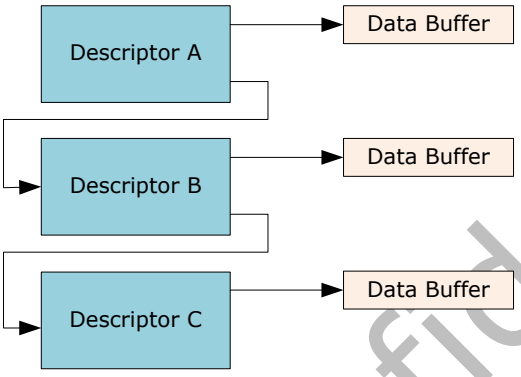


Fig. 1-7 Chain Descriptor Structure

- Descriptor formats

Following figure illustrates the internal formats of a descriptor. The descriptor addresses must be aligned to the bus width used for 32-bit AHB data buses. Each descriptor contains 16 bytes of control and status information. DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, DES3 to denote [127:96] bits.

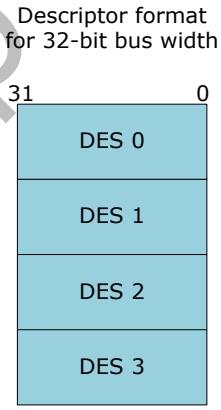


Fig. 1-8 Descriptor Formats for 32-bit AHB Address Bus Width

- The DES0 element in the IDMAC contains control and status information.

Table 1-4 Bits in IDMAC DES0 Element

Bit	Name	Description
31	OWN	When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the Host. The IDMAC clears this bit when it completes the data transfer.
30	Card Error Summary (CES)	These error bits indicate the status of the transaction to or from the card. These bits are also present in RINTSTS Indicates the logical

Bit	Name	Description
		OR of the following bits: ● EBE: End Bit Error ● RTO: Response Time out ● RCRC: Response CRC ● SBE: Start Bit Error ● DRTO: Data Read Timeout ● DCRC: Data CRC for Receive ● RE: Response Error
29:6	Reserved	-
5	End of Ring (ER)	When set, this bit indicates that the descriptor list reached its final descriptor. The IDMAC returns to the base address of the list, creating a Descriptor Ring. This is meaningful for only a dual-buffer descriptor structure.
4	Second Address Chained (CH)	When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When this bit is set, BS2 (DES1[25:13]) should be all zeros.
3	First Descriptor (FS)	When set, this bit indicates that this descriptor contains the first buffer of the data. If the size of the first buffer is 0, next Descriptor contains the beginning of the data.
2	Last Descriptor (LD)	This bit is associated with the last block of a DMA transfer. When set, the bit indicates that the buffers pointed to by this descriptor are the last buffers of the data. After this descriptor is completed, the remaining byte count is 0. In other words, after the descriptor with the LD bit set is completed, the remaining byte count should be 0.
1	Disable Interrupt on Completion (DIC)	When set, this bit will prevent the setting of the TI/RI bit of the IDMAC Status Register (IDSTS) for the data that ends in the buffer pointed to by this descriptor.
0	Reserved	-

■ The DES1 element contains the buffer size.

Table 1-5 Bits in IDMAC DES1 Element

Bit	Name	Description
31:26	Reserved	-
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

■ The DES2 element contains the address pointer to the data buffer.

Table 1-6 Bits in IDMAC DES2 Element

Bit	Name	Description
31:26	Reserved	
25:13	Buffer 2 Size	These bits indicate the second data buffer byte size. The

Bit	Name	Description
	(BS2)	buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

- The DES3 element contains the address pointer to the next descriptor if the present descriptor is not the last descriptor in a chained descriptor structure or the second buffer address for a dual-buffer structure.

Table 1-7 Bits in IDMAC DES3 Element

Bit	Name	Description
31:0	Buffer Address Pointer 2/ Next Descriptor Address (BAP2)	These bits indicate the physical address of the second buffer when the dual-buffer structure is used. If the Second Address Chained (DES0[4]) bit is set, then this address contains the pointer to the physical memory where the Next Descriptor is present. If this is not the last descriptor, then the Next Descriptor address pointer must be bus-width aligned.

3. Initialization

IDMAC initialization occurs as follows:

1) Write to IDMAC Bus Mode Register—BMOD to set Host bus access parameters.
2) Write to IDMAC Interrupt Enable Register—IDINTEN to mask unnecessary interrupt causes.
3) The software driver creates either the Transmit or the Receive descriptor list. Then it writes to IDMAC Descriptor List Base Address Register (DBADDR), providing the IDMAC with the starting address of the list.
4) The IDMAC engine attempts to acquire descriptors from the descriptor lists.

- Host Bus Burst Access

The IDMAC attempts to execute fixed-length burst transfers on the AHB Master interface if configured using the FB bit of the IDMAC Bus Mode register. The maximum burst length is indicated and limited by the PBL field. The descriptors are always accessed in the maximum possible burst-size for the 16-bytes to be read— 16*8/bus-width.

The IDMAC initiates a data transfer only when sufficient space to accommodate the configured burst is available in the FIFO or the number of bytes to the end of data, when less than the configured burst-length.

The IDMAC indicates the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length bursts, then it transfers data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise, in no fixed-length bursts, it transfers data using INCR (undefined length) and SINGLE transactions.

- Host Data Buffer Alignment

The Transmit and Receive data buffers in host memory must be aligned, depending on the data width.

- Buffer Size Calculations

The driver knows the amount of data to transmit or receive. For transmitting to the card, the IDMAC transfers the exact number of bytes to the FIFO, indicated by the buffer size field of DES1.

If a descriptor is not marked as last-LS bit of DES0-then the corresponding buffer(s) of the descriptor are full, and the amount of valid data in a buffer is accurately indicated by its buffer size field. If a descriptor is marked as last, then the buffer cannot be full, as indicated by the buffer size in DES1. The driver is aware of the number of locations that are valid in this case.

- Transmission

IDMAC transmission occurs as follows:

1) The Host sets up the elements (DES0-DES3) for transmission and sets the OWN bit (DES0[31]). The Host also prepares the data buffer.
2) The Host programs the write data command in the CMD register in BIU.
3) The Host will also program the required transmit threshold level (TX_WMark field in FIFOTH register).
4) The IDMAC determines that a write data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the IDMAC enters suspend state and asserts the Descriptor Unable interrupt in the IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed transmit threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB Master Interface.
8) The IDMAC fetches the Transmit data from the data buffer in the Host memory and transfers to the FIFO for transmission to card.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data transmission is complete, status information is updated in IDSTS register by setting Transmit Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Reception

IDMAC reception occurs as follows:

1) The Host sets up the element (DES0-DES3) for reception, sets the OWN (DES0[31]).
2) The Host programs the read data command in the CMD register in BIU.
3) The Host will program the required receive threshold level (RX_WMark field in FIFOTH register).
4) The IDMAC determines that a read data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the DMA enters suspend state and asserts the Descriptor Unable interrupt in the IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed receive threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB.
8) The IDMAC fetches the data from the FIFO and transfer to Host memory.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data reception is complete, status information is updated in IDSTS register by setting Receive Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Interrupts

Interrupts can be generated as a result of various events. IDSTS register contains all the bits that might cause an interrupt. IDINTEN register contains an Enable bit for each of the events that can cause an interrupt.
There are two groups of summary interrupts-Normal and Abnormal-as outlined in IDSTS register. Interrupts are cleared by writing a 1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal dmac_intr_o is de-asserted.
Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt—IDSTS[1] indicates that one or more data was transferred to the Host buffer.
An interrupt is generated only once for simultaneous, multiple events. The driver must scan IDSTS register for the interrupt cause.

1.4 Register Description

1.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SDMMC_CTRL	0x0000	W	0x01000000	Control register
SDMMC_PWREN	0x0004	W	0x00000000	Power-enable register
SDMMC_CLKDIV	0x0008	W	0x00000000	Clock-divider register
SDMMC_CLKSRC	0x000c	W	0x00000000	SD clock source register
SDMMC_CLKENA	0x0010	W	0x00000000	Clock-enable register
SDMMC_TMOUT	0x0014	W	0xffffff40	Time-out register
SDMMC_CTYPE	0x0018	W	0x00000000	Card-type register
SDMMC_BLKSIZ	0x001c	W	0x00000200	Block-size register
SDMMC_BYTCNT	0x0020	W	0x00000200	Byte-count register
SDMMC_INTMASK	0x0024	W	0x00000000	Interrupt-mask register
SDMMC_CMDARG	0x0028	W	0x00000000	Command-argument register
SDMMC_CMD	0x002c	W	0x00000000	Command register
SDMMC_RESP0	0x0030	W	0x00000000	Response-0 register
SDMMC_RESP1	0x0034	W	0x00000000	Response-1 register
SDMMC_RESP2	0x0038	W	0x00000000	Response-2 register
SDMMC_RESP3	0x003c	W	0x00000000	Response-3 register
SDMMC_MINTSTS	0x0040	W	0x00000000	Masked interrupt-status register
SDMMC_RINTSTS	0x0044	W	0x00000000	Raw interrupt-status register
SDMMC_STATUS	0x0048	W	0x00000406	Status register
SDMMC_FIFOTH	0x004c	W	0x00000000	FIFO threshold register
SDMMC_CDETECT	0x0050	W	0x00000000	Card-detect register
SDMMC_WRTPRT	0x0054	W	0x00000000	Write-protect register
SDMMC_TCBCNT	0x005c	W	0x00000000	Transferred CIU card byte count
SDMMC_TBBCNT	0x0060	W	0x00000000	Transferred host/DMA to/from BIU-FIFO byte count
SDMMC_DEBNCE	0x0064	W	0x00ffffff	Card detect debounce register
SDMMC_USRID	0x0068	W	0x07967797	User ID register
SDMMC_VERID	0x006c	W	0x5342270a	Version ID register
SDMMC_HCON	0x0070	W	0x00000000	Hardware configuration register

Name	Offset	Size	Reset Value	Description
SDMMC_UHS_REG	0x0074	W	0x00000000	UHS-1 register
SDMMC_RST_n	0x0078	W	0x00000001	Hardware reset register
SDMMC_BMOD	0x0080	W	0x00000000	Bus mode register
SDMMC_PLDMND	0x0084	W	0x00000000	Poll demand register
SDMMC_DBADDR	0x0088	W	0x00000000	Descriptor list base address register
SDMMC_IDSTS	0x008c	W	0x00000000	Internal DMAC status register
SDMMC_IDINTEN	0x0090	W	0x00000000	Internal DMAC interrupt enable register
SDMMC_DSCADDR	0x0094	W	0x00000000	Current host descriptor address register
SDMMC_BUFADDR	0x0098	W	0x00000000	Current buffer descriptor address register
SDMMC_CARDTHRCTL	0x0100	W	0x00000000	Card read threshold enable register
SDMMC_BACK_END_POWER	0x0104	W	0x00000000	Back-end power register
SDMMC_EMMC_DDR_REG	0x010c	W	0x00000000	eMMC4.5 DDR start bit detection control register
SDMMC_FIFO_BASE	0x0200	W	0x00000000	FIFO base address register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

1.4.2 Detail Register Description

SDMMC_CTRL

Address: Operational Base + offset (0x0000)

Control register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	use_internal_dmac Present only for the Internal DMAC configuration; else, it is reserved. 0: The host performs data transfers through the slave interface 1: Internal DMAC used for data transfe
24:12	RO	0x0	reserved
11	RW	0x0	ceata_device_interrupt_status 0: Interrupts not enabled in CE-ATA device (nIEN = 1 in ATA control register) 1: Interrupts are enabled in CE-ATA device (nIEN = 0 in ATA control register) Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled (nIEN = 1). If the host enables CE-ATA device interrupt, then software should set this bit.

Bit	Attr	Reset Value	Description
10	RW	0x0	send_auto_stop_ccsd 0: Clear bit if Mobile Storage Host Controller does not reset the bit. 1: Send internally generated STOP after sending CCSD to CE-ATA device. NOTE: Always set send_auto_stop_ccsd and send_ccsd bits together send_auto_stop_ccsd should not be set independent of send_ccsd. When set, Mobile Storage Host Controller automatically sends internally- generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) in RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, Mobile Storage Host Controller automatically clears send_auto_stop_ccsd bit.
9	RW	0x0	send_ccsd 0: Clear bit if Mobile Storage Host Controller does not reset the bit. 1: Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, Mobile Storage Host Controller sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, Mobile Storage Host Controller automatically clears send_ccsd bit. It also sets Command Done (CD) bit in RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. NOTE: Once send_ccsd bit is set, it takes two card clock cycles to drive the CCSD on the CMD line. Due to this, during the boundary conditions it may happen that CCSD is sent to the CE-ATA device, even if the device signalled CCS
8	RW	0x0	abort_read_data 0: no change 1: after suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state machine resets to idle. Used in SDIO card suspend sequence.

Bit	Attr	Reset Value	Description
7	RW	0x0	send_irq_response 0: no change 1: send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40, and SDMMC Controller waits for interrupt response from MMC card(s). In meantime, if host wants SDMMC Controller to exit waiting for interrupt state, it can set this bit, at which time SDMMC Controller command state-machine sends CMD40 response on bus and returns to idle state.
6	RW	0x0	read_wait 0: clear read wait 1: assert read wait For sending read-wait to SDIO cards
5	RW	0x0	dma_enable 0: disable DMA transfer mode 1: enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside SDMMC Controller to prioritize simultaneous host/DMA access.
4	RW	0x0	int_enable Global interrupt enable/disable bit: 0: disable interrupts 1: enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.
3	RO	0x0	reserved
2	W1C	0x0	dma_reset 0: no change 1: reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks.
1	W1C	0x0	fifo_reset 0: no change 1: reset to data FIFO To reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation

Bit	Attr	Reset Value	Description
0	W1C	0x0	controller_reset 0: no change 1: reset SDMMC controller To reset controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: a. BIU/CIU interface b. CIU and state machines c. abort_read_data, send_irq_response, and read_wait bits of Control register d. start_cmd bit of Command register Does not affect any registers or DMA interface, or FIFO or host interrupts

SDMMC_PWREN
Address: Operational Base + offset (0x0004)
Power-enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	power_enable Power on/off switch for the card. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card. 0: power off 1: power on Bit values output to card_power_en port.

SDMMC_CLKDIV
Address: Operational Base + offset (0x0008)
Clock-divider register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	clk_divider0 Clock divider-0 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, value of "ff"means divide by 2*255 = 510, and so on

SDMMC_CLKSRC
Address: Operational Base + offset (0x000c)
SD clock source register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	clk_source Clock divider source for up to 16 SD cards supported. Each card has two bits assigned to it. For example, bits[1:0] assigned for card-0, which maps and internally routes clock divider[3:0] outputs to cclk_out[15:0] pins, depending on bit value. 00: Clock divider 0 The cclk_out is always from clock divider 0, and this register is not implemented.

SDMMC_CLKENA
Address: Operational Base + offset (0x0010)
Clock-enable register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	cclk_low_power Low-power control for SD card clock and MMC card clock supported. 0: non-low-power mode 1: low-power mode; stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO cards, if interrupts must be detected, clock should not be stopped).
15:1	RO	0x0	reserved
0	RW	0x0	cclk_enable Clock-enable control for SD card clock and MMC card clock supported. 0: clock disabled 1: clock enabled

SDMMC_TMOUT
Address: Operational Base + offset (0x0014)
Time-out register

Bit	Attr	Reset Value	Description
31:8	RW	0xffffff	data_timeout Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. Value is in number of card output clocks cclk_out of selected card. Note: The software timer should be used if the timeout value is in the order of 100 ms. In this case, read data timeout interrupt needs to be disabled.
7:0	RW	0x40	response_timeout Response timeout value. Value is in number of card output clocks –cclk_out.

SDMMC_CTYPE

Address: Operational Base + offset (0x0018)
Card-type register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	card_width_8 Indicates if card is 8-bit: 0: non 8-bit mode 1: 8-bit mode
15:1	RO	0x0	reserved
0	RW	0x0	card_width Indicates if card is 1-bit or 4-bit: 0: 1-bit mode 1: 4-bit mode

SDMMC_BLKSIZ
Address: Operational Base + offset (0x001c)
Block-size register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0200	block_size Block size

SDMMC_BYTCNT
Address: Operational Base + offset (0x0020)
Byte-count register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000200	byte_count Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.

SDMMC_INTMASK
Address: Operational Base + offset (0x0024)
Interrupt-mask register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sdio_int_mask Mask SDIO interrupts. When masked, SDIO interrupt detection for that card is disabled. A 0 masks an interrupt, and 1 enables an interrupt.
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_mask 0: data no busy interrupt not masked 1: data no busy interrupt masked

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_mask Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. [15]: End-bit error (read)/Write no CRC (EBE) [14]: Auto command done (ACD) [13]: Start-bit error (SBE) [12]: Hardware locked write error (HLE) [11]: FIFO underrun/overrun error (FRUN) [10]: Data starvation-by-host timeout (HTO) /Volt_switch_int [9]: Data read timeout (DRTO) [8]: Response timeout (RTO) [7]: Data CRC error (DCRC) [6]: Response CRC error (RCRC) [5]: Receive FIFO data request (RXDR) [4]: Transmit FIFO data request (TXDR) [3]: Data transfer over (DTO) [2]: Command done (CD) [1]: Response error (RE) [0]: Card detect (CD)

SDMMC_CMDARG
Address: Operational Base + offset (0x0028)
Command-argument register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_arg Value indicates command argument to be passed to card.

SDMMC_CMD
Address: Operational Base + offset (0x002c)
Command register

Bit	Attr	Reset Value	Description
31	RW	0x0	start_cmd Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC cards, Command Done bit is set in raw interrupt register.
30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	use_hold_reg Use Hold Register 0: CMD and DATA sent to card bypassing HOLD Register 1: CMD and DATA sent to card through the HOLD Register Note: a. Set to 1'b1 for SDR12 and SDR25 (with non-zero phase-shifted cclk_in_drv); zero phase shift is not allowed in these modes. b. Set to 1'b0 for SDR50, SDR104, and DDR50 (with zero phase-shifted cclk_in_drv). c. Set to 1'b1 for SDR50, SDR104, and DDR50 (with non-zero phase-shifted cclk_in_drv).
28	RW	0x0	volt_switch Voltage switch bit. 0: no voltage switching 1: voltage switching enabled; must be set for CMD11 only
27	RW	0x0	boot_mode Boot Mode. 0: mandatory Boot operation 1: alternate Boot operation
26	RW	0x0	disable_boot Disable Boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do NOT set disable_boot and enable_boot together.
25	RW	0x0	expect_boot_ack Expect Boot Acknowledge. When Software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.
24	RW	0x0	enable_boot Enable Boot—this bit should be set only for mandatory boot mode.When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do NOT set disable_boot and enable_boot together.
23	RW	0x0	ccs_expected 0: Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device 1: Interrupts are enabled in CE-ATA device (nIEN = 0), and RW_BLK command expects command completion signal from CE-ATA device. If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software should set this control bit. Mobile Storage Host Controller sets Data Transfer Over (DTO) bit in RINTSTS register and generates interrupt to host if Data Transfer Over interrupt is not masked.

Bit	Attr	Reset Value	Description
22	RW	0x0	read_ceata_device 0: Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device 1: Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. Mobile Storage Host Controller should not indicate read data timeout while waiting for data from CE-ATA device.
21	RW	0x0	update_clock_registers_only 0: normal command sequence 1: do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_registers_only = 0, following control registers are transferred from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card. When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC_CEATA cards.
20:16	RO	0x0	reserved
15	RW	0x0	send_initialization 0: do not send initialization sequence (80 clocks of 1) before sending this command 1: send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory).

Bit	Attr	Reset Value	Description
14	RW	0x0	stop_abort_cmd 0: neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1: stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with CMD[26] = disable_boot.
13	RW	0x0	wait_prvdata_complete 0: send command at once, even if previous data transfer has not completed 1: wait for previous data transfer completion before sending command The wait_prvdata_complete = 0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.
12	RW	0x0	send_auto_stop 0: no stop command sent at end of data transfer 1: send stop command at end of data transfer When set, SDMMC Controller sends stop command to SD_MMC cards at end of data transfer. a. when send_auto_stop bit should be set, since some data transfers do not need explicit stop commands b. open-ended transfers that software should explicitly send to stop command Additionally, when "resume" is sent to resume –suspended memory access of SD-Combo card –bit should be set correctly if suspended data transfer needs send_auto_stop. Don't care if no data expected from card.
11	RW	0x0	transfer_mode 0: block data transfer command 1: stream data transfer command Don't care if no data expected.
10	RW	0x0	wr 0: read from card 1: write to card Don't care if no data expected from card.
9	RW	0x0	data_expected 0: no data transfer expected (read/write) 1: data transfer expected (read/write)

Bit	Attr	Reset Value	Description
8	RW	0x0	check_response_crc 0: do not check response CRC 1: check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller
7	RW	0x0	response_length 0: short response expected from card 1: long response expected from card
6	RW	0x0	response_expect 0: no response expected from card 1: response expected from card
5:0	RW	0x00	cmd_index Command index

SDMMC_RESP0
Address: Operational Base + offset (0x0030)
Response-0 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response0 Bit[31:0] of response

SDMMC_RESP1
Address: Operational Base + offset (0x0034)
Response-1 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response Register represents bit[63:32] of long response. When CIU sends auto-stop command, then response is saved in register. Response for previous command sent by host is still preserved in Response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short" for them.

SDMMC_RESP2
Address: Operational Base + offset (0x0038)
Response-2 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response2 Bit[95:64] of long response

SDMMC_RESP3
Address: Operational Base + offset (0x003c)
Response-3 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response3 Bit[127:96] of long response

SDMMC_MINTSTS
Address: Operational Base + offset (0x0040)
Masked interrupt-status register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RO	0x0	sdio_interrupt Interrupt from SDIO card; SDIO interrupt for card enabled only if corresponding sdio_int_mask bit is set in Interrupt mask register (mask bit 1 enables interrupt; 0 masks interrupt). 0: no SDIO interrupt from card 1: SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_status Data no busy Interrupt Status
15:0	RO	0x0000	int_status Interrupt enabled only if corresponding bit in interrupt mask register is set. [15]: End-bit error (read)/Write no CRC (EBE) [14]: Auto command done (ACD) [13]: Start-bit error (SBE) [12]: Hardware locked write error (HLE) [11]: FIFO underrun/overrun error (FRUN) [10]: Data starvation-by-host timeout (HTO) /Volt_switch_int [9]: Data read timeout (DRTO) [8]: Response timeout (RTO) [7]: Data CRC error (DCRC) [6]: Response CRC error (RCRC) [5]: Receive FIFO data request (RXDR) [4]: Transmit FIFO data request (TXDR) [3]: Data transfer over (DTO) [2]: Command done (CD) [1]: Response error (RE) [0]: Card detect (CD)

SDMMC_RINTSTS
Address: Operational Base + offset (0x0044)
Raw interrupt-status register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RO	0x0	sdio_interrupt Interrupt from SDIO card; Writes to these bits clear them. Value of 1 clears bit and 0 leaves bit intact. 0: no SDIO interrupt from card 1: SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_status Data no busy interrupt status
15:0	RO	0x0000	int_status Writes to bits clear status bit. Value of 1 clears status bit, and value of 0 leaves bit intact. Bits are logged regardless of interrupt mask status. [15]: End-bit error (read)/Write no CRC (EBE) [14]: Auto command done (ACD) [13]: Start-bit error (SBE) [12]: Hardware locked write error (HLE) [11]: FIFO underrun/overrun error (FRUN) [10]: Data starvation-by-host timeout (HTO) /Volt_switch_int [9]: Data read timeout (DRTO) [8]: Response timeout (RTO) [7]: Data CRC error (DCRC) [6]: Response CRC error (RCRC) [5]: Receive FIFO data request (RXDR) [4]: Transmit FIFO data request (TXDR) [3]: Data transfer over (DTO) [2]: Command done (CD) [1]: Response error (RE) [0]: Card detect (CD)

SDMMC_STATUS
Address: Operational Base + offset (0x0048)
Status register

Bit	Attr	Reset Value	Description
31	RO	0x0	dma_req DMA request signal state
30	RO	0x0	dma_ack DMA acknowledge signal state
29:17	RO	0x0000	fifo_count Number of filled locations in FIFO
16:11	RO	0x00	response_index Index of previous response, including any auto-stop sent by core
10	RO	0x1	data_state_mc_busy Data transmit or receive state-machine is busy

Bit	Attr	Reset Value	Description
9	RO	0x0	data_busy Inverted version of raw selected card_data[0] 0: card data not busy 1: card data busy default value is 1 or 0 depending on cdata_in
8	RO	0x0	data_3_status Raw selected card_data[3]; checks whether card is present 0: card not present 1: card present default value is 1 or 0 depending on cdata_in
7:4	RO	0x0	command_fsm_states Command FSM states: 0: idle 1: send init sequence 2: Tx cmd start bit 3: Tx cmd tx bit 4: Tx cmd index + arg 5: Tx cmd crc7 6: Tx cmd end bit 7: Rx resp start bit 8: Rx resp IRQ response 9: Rx resp tx bit 10: Rx resp cmd idx 11: Rx resp data 12: Rx resp crc7 13: Rx resp end bit 14: Cmd path wait NCC 15: Wait; CMD-to-response turnaround The command FSM state is represented using 19 bits. The STATUS Register[7:4] has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the STATUS[7:4] register. The three states that are not represented in the STATUS Register[7:4] are: a. Bit 16 –Wait for CCS b. Bit 17 –Send CCSD c. Bit 18 –Boot Mode Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", the Status register indicates status as 0 for the bit field [7:4].
3	RO	0x0	fifo_full FIFO is full status
2	RO	0x1	fifo_empty FIFO is empty status

Bit	Attr	Reset Value	Description
1	RO	0x1	fifo_tx_watermark FIFO reached Transmit watermark level; not qualified with data transfer
0	RO	0x0	fifo_rx_watermark FIFO reached Receive watermark level; not qualified with data transfer

SDMMC_FIFOTH
Address: Operational Base + offset (0x004c)
FIFO threshold register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:28	RW	0x0	dma_mutiple_transaction_size Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE. 0: 1 transfers 1: 4 2: 8 3: 16 4: 32 5: 64 6: 128 7: 256 The unit for transfer is the H_DATA_WIDTH parameter. A single transfer (dw_dma_single assertion in case of Non DW DMA interface) would be signalled based on this value. Value should be sub-multiple of (RX_WMark + 1)* (F_DATA_WIDTH/H_DATA_WIDTH) and (FIFO_DEPTH - TX_WMark)* (F_DATA_WIDTH/ H_DATA_WIDTH) For example, if FIFO_DEPTH = 16, FDATA_WIDTH == H_DATA_WIDTH Allowed combinations for MSize and TX_WMark are: MSize = 1, TX_WMARK = 1-15 MSize = 4, TX_WMark = 8 MSize = 4, TX_WMark = 4 MSize = 4, TX_WMark = 12 MSize = 8, TX_WMark = 8 MSize = 8, TX_WMark = 4 Allowed combinations for MSize and RX_WMark are: MSize = 1, RX_WMARK = 0-14 MSize = 4, RX_WMark = 3 MSize = 4, RX_WMark = 7 MSize = 4, RX_WMark = 11 MSize = 8, RX_WMark = 7 Recommended: MSize = 8, TX_WMark = 8, RX_WMark = 7

Bit	Attr	Reset Value	Description
27:16	RW	0x000	rx_wmark FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits-1 bit less than FIFO-count of status register, which is 13 bits. Limitation: RX_WMark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1) *NOTE: In DMA mode during CCS time-out, the DMA does not generate the request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in the FIFO. It is the responsibility of the application to reset the FIFO after the CCS timeout.*
15:12	RO	0x0	reserved
11:0	RW	0x000	tx_wmark FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits -1 bit less than FIFO-count of status register, which is 13 bits. Limitation: TX_WMark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to FIFO_DEPTH/2)

SDMMC_CDETECT
Address: Operational Base + offset (0x0050)
Card-detect register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	card_detect_n Value on card_detect_n input ports; read-only bits. 0 represents presence of card.

SDMMC_WRTPRT
Address: Operational Base + offset (0x0054)
Write-protect register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	write_protect Value on card_write_prt input port. 1 represents write protection.

SDMMC_TCBCNT
Address: Operational Base + offset (0x005c)
Transferred CIU card byte count

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_card_byte_count Number of bytes transferred by CIU unit to card. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied. Both TCBCNT and TBBCNT share same coherency register. When AREA_OPTIMIZED parameter is 1, register should be read only after data transfer completes; during data transfer, register returns 0.

SDMMC_TBBCNT
Address: Operational Base + offset (0x0060)
Transferred host/DMA to/from BIU-FIFO byte count

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_fifo_byte_count Number of bytes transferred between Host/DMA memory and BIU FIFO. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied. Both TCBCNT and TBBCNT share same coherency register.

SDMMC_DEBNCE
Address: Operational Base + offset (0x0064)
Card detect debounce register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0xffffff	debounce_count Number of host clocks (clk) used by debounce filter logic; typical debounce time is 5-25 ms.

SDMMC_USRID
Address: Operational Base + offset (0x0068)
User ID register

Bit	Attr	Reset Value	Description
31:0	RW	0x07967797	usrid User identification register. The default value is determined by Configuration Value.

SDMMC_VERID
Address: Operational Base + offset (0x006c)
Version ID register

Bit	Attr	Reset Value	Description
31:0	RO	0x5342270a	verid Version identification register; register value is hard-wired. Can be read by firmware to support different versions of core.

SDMMC_HCON
Address: Operational Base + offset (0x0070)
Hardware configuration register

Bit	Attr	Reset Value	Description

31:0	RO	0x00000000	HCON Configuration Dependent. Hardware configurations selected by user before synthesizing core. Register values can be used to develop configuration-independent software drivers. [0]: CARD_TYPE 0: MMC_ONLY 1: SD_MMC [5:1]: NUM_CARDS - 1 [6]: H_BUS_TYPE 0: APB 1: AHB [9:7]: H_DATA_WIDTH 0: 16 bits 1: 32 bits 2: 64 bits others: reserved [15:10]: H_ADDR_WIDTH 0 to 7: reserved 8: 9 bits 9: 10 bits … 31: 32 bits 32 to 63: reserved [17:16]: DMA_INTERFACE 0: none 1: DMA 1 2: DMA 2 3: DMA 3 [20:18]: GE_DMA_DATA_WIDTH 0: 16 bits 1: 32 bits 2: 64 bits others: reserved [21]: FIFO_RAM_INSIDE 0: outside 1: inside [22]: IMPLEMENT_HOLD_REG 0: no hold register 1: hold register [23]: SET_CLK_FALSE_PATH 0: no false path 1: false path set [25:24]: NUM_CLK_DIVIDER-1 [26]: AREA_OPTIMIZED 0: no area optimization 1: Area optimization

SDMMC_UHS_REG
Address: Operational Base + offset (0x0074)
UHS-1 register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ddr_reg DDR mode. Determines the voltage fed to the buffers by an external voltage regulator. 0: non-DDR mode 1: DDR mode UHS_REG [16] should be set for card.
15:0	RO	0x0	reserved

SDMMC_RST_n
Address: Operational Base + offset (0x0078)
Hardware reset register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	card_reset Hardware reset. 0: active mode 1: reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized. CARD_RESET[0] should be set to 1'b1 to reset card.

SDMMC_BMOD
Address: Operational Base + offset (0x0080)
Bus mode register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RO	0x0	PBL Programmable Burst Length. These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of FIFOTH register. In order to change this value, write the required value to FIFOTH register. This is an encode value as follows. 0: 1 transfers 1: 4 transfers 2: 8 transfers 3: 16 transfers 4: 32 transfers 5: 64 transfers 6: 128 transfers 7: 256 transfers Transfer unit is either 16, 32, or 64 bits, based on HDATA_WIDTH. PBL is a read-only value and is applicable only for Data Access; it does not apply to descriptor accesses.
7	RW	0x0	DE IDMAC Enable. When set, the IDMAC is enabled.
6:2	RW	0x00	DSL Descriptor Skip Length. Specifies the number of HWord/Word/Dword (depending on 16/32/64-bit bus) to skip between two unchained descriptors. This is applicable only for dual buffer structure.
1	RW	0x0	FB Fixed Burst. Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.
0	RW	0x0	SWR Software Reset. When set, the DMA Controller resets all its internal registers. It is automatically cleared after 1 clock cycle.

SDMMC_PLDMND
Address: Operational Base + offset (0x0084)
Poll demand register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	PD Poll Demand. If the OWN bit of a descriptor is not set, the FSM goes to the Suspend state. The host needs to write any value into this register for the IDMAC FSM to resume normal descriptor fetch operation. This is a write only register.

SDMMC_DBADDR
Address: Operational Base + offset (0x0088)
Descriptor list base address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SDL Start of Descriptor List. Contains the base address of the First Descriptor. The LSB bits [0/1/2:0] for 16/32/64-bit bus-width) are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only.

SDMMC_IDSTS
Address: Operational Base + offset (0x008c)
Internal DMAC status register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:13	RO	0x0	FSM DMAC FSM present state. 0: DMA_IDLE 1: DMA_SUSPEND 2: DESC_RD 3: DESC_CHK 4: DMA_RD_REQ_WAI 5: DMA_WR_REQ_WAI 6: DMA_RD 7: DMA_WR 8: DESC_CLOSE
12:10	RO	0x0	EB Error Bits. Indicates the type of error that caused a Bus Error. Valid only with atal Bus Error bit—IDSTS[2] (IDSTS64[2], in case of 64-bit address configuration) set. This field does not generate an interrupt. 1: Host Abort received during transmission 2: Host Abort received during reception Others: Reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	AIS Abnormal Interrupt Summary. Logical OR of the following: IDSTS[2] Fatal Bus Interrupt IDSTS[4] DU bit Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit.
8	RW	0x0	NIS Normal Interrupt Summary. Logical OR of the following: IDSTS[0] Transmit Interrupt IDSTS[1] Receive Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit.
7:6	RO	0x0	reserved
5	RW	0x0	CES Card Error Summary. Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits: EBE: End Bit Error RTO: Response Timeout/Boot Ack Timeout RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout/BDS timeout DCRC: Data CRC for Receive RE: Response Error Writing a 1 clears this bit. The abort condition of the IDMAC depends on the setting of this CES bit. If the CES bit is enabled, then the IDMAC aborts on a "response error"; however, it will not abort if the CES bit is cleared.
4	RW	0x0	DU Descriptor Unavailable Interrupt. This bit is set when the descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Writing a 1 clears this bit.
3	RO	0x0	reserved
2	RW	0x0	FBE Fatal Bus Error Interrupt. Indicates that a Bus Error occurred (IDSTS[12:10]) (IDSTS64[12:10], in case of 64-bit address configuration). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit.

Bit	Attr	Reset Value	Description
1	RW	0x0	RI Receive Interrupt. Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit.
0	RW	0x0	TI Transmit Interrupt. Indicates that data transmission is finished for a descriptor. Writing 1 clears this bit.

SDMMC_IDINTEN
Address: Operational Base + offset (0x0090)
Internal DMAC interrupt enable register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	AI Abnormal Interrupt Summary Enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: IDINTEN[2] Fatal Bus Error Interrupt IDINTEN[4] DU Interrupt
8	RW	0x0	NI Normal Interrupt Summary Enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: IDINTEN[0] Transmit Interrupt IDINTEN[1] Receive Interrupt
7:6	RO	0x0	reserved
5	RW	0x0	CES Card Error summary Interrupt Enable. When set, it enables the Card Interrupt summary.
4	RW	0x0	DU Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the DU interrupt is enabled.
3	RO	0x0	reserved
2	RW	0x0	FBE Fatal Bus Error Enable. When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.
1	RW	0x0	RI Receive Interrupt Enable. When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled.
0	RW	0x0	TI Transmit Interrupt Enable. When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled.

SDMMC_DSCADDR
Address: Operational Base + offset (0x0094)
Current host descriptor address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HDA Host Descriptor Address Pointer. Cleared on reset. Pointer updated by IDMAC during operation. This register points to the start address of the current descriptor read by the IDMAC.

SDMMC_BUFADDR
Address: Operational Base + offset (0x0098)
Current buffer descriptor address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HBA Host Buffer Address Pointer. Cleared on Reset. Pointer updated by IDMAC during operation. This register points to the current Data Buffer Address being accessed by the IDMAC.

SDMMC_CARDTHRCTL
Address: Operational Base + offset (0x0100)
Card read threshold enable register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	CardRdThreshold Card Read Threshold size
15:2	RO	0x0	reserved
1	RW	0x0	BsyClrIntEn Busy Clear Interrupt generation: 0: Busy Clear Interrupt disabled 1: Busy Clear Interrupt enabled Note: The application can disable this feature if it does not want to wait for a Busy Clear Interrupt. For example, in a multi-card scenario, the application can switch to the other card without waiting for a busy to be completed. In such cases, the application can use the polling method to determine the status of busy. By default this feature is disabled and backward-compatible to the legacy drivers where polling is used.
0	RW	0x0	CardRdThrEn Card Read Threshold Enable. 0: Card Read Threshold disabled 1: Card Read Threshold enabled. Host Controller initiates Read Transfer only if CardRdThreshold amount of space is available in receive FIFO.

SDMMC_BACK_END_POWER
Address: Operational Base + offset (0x0104)

Back-end power register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	back_end_power Back end power 0: Off; Reset 1: Back-end Power supplied to card application

SDMMC_EMMC_DDR_REG
Address: Operational Base + offset (0x010c)
eMMC4.5 DDR start bit detection control register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	HALF_START_BIT Control for start bit detection mechanism inside Mobile Storage Host Controller based on duration of start bit; each bit refers to one slot. For eMMC 4.5, start bit can be: 0: Full cycle (HALF_START_BIT = 0) 1: Less than one full cycle (HALF_START_BIT = 1) Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications.

SDMMC_FIFO_BASE
Address: Operational Base + offset (0x0200)
FIFO base address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	fifo_base_addr FIFO base address

1.5 Interface Description

The interface and IOMUX setting for SDMMC, SDIO are shown as follows.

Table 1-8 SDMMC Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
sdmmc_cclk	O	IO_SDMMCclkout_MCUJTAGtck_HDCPJTAGtck_SDMMCgpio4b4	GRF_GPIO4B_IOMUX[9:8]=2'b01
sdmmc_ccmd	I/O	IO_SDMMCcmd_MCUJTAGtms_HDCPJTAGtms_SDMMCgpio4b5	GRF_GPIO4B_IOMUX[11:00]=2'b01
sdmmc_cdata0	I/O	IO_SDMMCdata0_UART2DBGAsin_SDMMCgpio4b0	GRF_GPIO4B_IOMUX[1:0]=2'b01
sdmmc_cdata1	I/O	IO_SDMMCdata1_UART2DBGAsout_HDCPJTAGtrstn_SDMMCgpio4b1	GRF_GPIO4B_IOMUX[3:2]=2'b01
sdmmc_cdata2	I/O	IO_SDMMCdata2_CXCSJTAGtck_HDCPJTAGtdi_SDMMCgpio4b2	GRF_GPIO4B_IOMUX[5:4]=2'b01
sdmmc_cdata3	I/O	IO_SDMMCdata3_CXCSJTAGtms_HDCPJTAGtdo_SDMMCgpio4b3	GRF_GPIO4B_IOMUX[7:6]=2'b01
sdmmc_cdetectn	I	IO_SDMMCdectn_PMUdebug5_PMU18gpio0a7	PMUGRF_GPIO0A_IOMUX[15:14]=2'b01
sdmmc_wprt	I	IO_SDMMCwrprt_PMUM0wfi_TESTclkout2_PMU18gpio0b0	PMUGRF_GPIO0B_IOMUX[1:0]=2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

Table 1-9 SDIO Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
sdio_cclk	O	IO_SDIOclkout_TESTclkout1_WIFIBTgpio2d1	GRF_GPIO2D_IOMUX[3:2]=2'b01
sdio_ccmd	I/O	IO_SDIOcmd_WIFIBTgpio2d0	GRF_GPIO2D_IOMUX[1:0]=2'b01
sdio_cdata0	I/O	IO_SDIOdata0_SPI5EXPPLUSrxd_WIFIBTgpio2c4	GRF_GPIO2C_IOMUX[9:8]=2'b01
sdio_cdata1	I/O	IO_SDIOdata1_SPI5EXPPLUStxd_WIFIBTgpio2c5	GRF_GPIO2C_IOMUX[11:10]=2'b01
sdio_cdata2	I/O	IO_SDIOdata2_SPI5EXPPLUSclk_WIFIBTgpio2c6	GRF_GPIO2C_IOMUX[13:12]=2'b01
sdio_cdata3	I/O	IO_SDIOdata3_SPI5EXPPLUScsn0_WIFIBTgpio2c7	GRF_GPIO2C_IOMUX[15:14]=2'b01
sdio_cdetectn	I	IO_SDIOdetectn_PCIEclkreqn_WIFIBTgpio2d2	GRF_GPIO2D_IOMUX[5:4]=2'b01
sdio_wprt	I	IO_SDIOwrprt_PMUdebug1_PMU18gpio0a3	PMUGRF_GPIO0A_IOMUX[7:6]=2'b01
sdio_intn	I	IO_SDIOintn_PMUdebug2_PMU18gpio0a4	PMUGRF_GPIO3C_IOMUX[9:8]=2'b01
sdio_pwren	O	IO_SDIOpwren_WIFIBTgpio2d3	GRF_GPIO2D_IOMUX[7:6]=2'b01
sdio_bkpwr	O	IO_SDIObkpwr_WIFIBTgpio2d4	GRF_GPIO2D_IOMUX[9:8]=2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

1.6 Application Notes

1.6.1 Card-Detect and Write-Protect Mechanism

Following figure illustrates how the SD/MMC card detection and write-protect signals are connected. Most of the SD/MMC sockets have card-detect pins. When no card is present, card_detect_n is 1 due to the pull-up. When the card is inserted, the card-detect pin is shorted to ground, which makes card_detect_n go to 0. Similarly in SD cards, when the write-protect switch is toward the left, it shorts the write_protect port to ground.

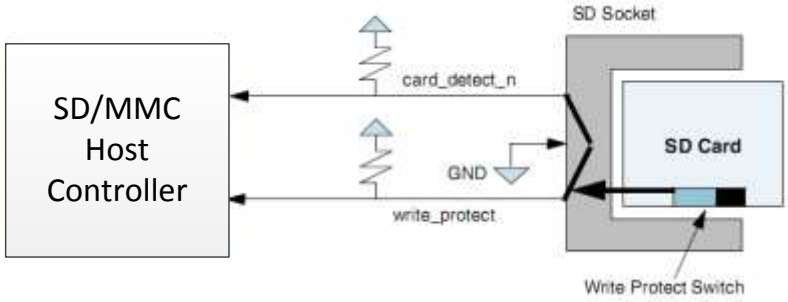


Fig. 1-9 SD/MMC Card-Detect and Write-Protect

1.6.2 SD/MMC Termination Requirement

Following Figure illustrates the SD/MMC termination requirements, which is required to pull up ccmd and cdata lines on the device bus. The recommended specification for pull-up on the ccmd line (Rcmd) is 4.7K - 100K for MMC, and 10K - 100K for an SD. The recommended pull-up on the cdata line (Rdat) is 50K - 100K.

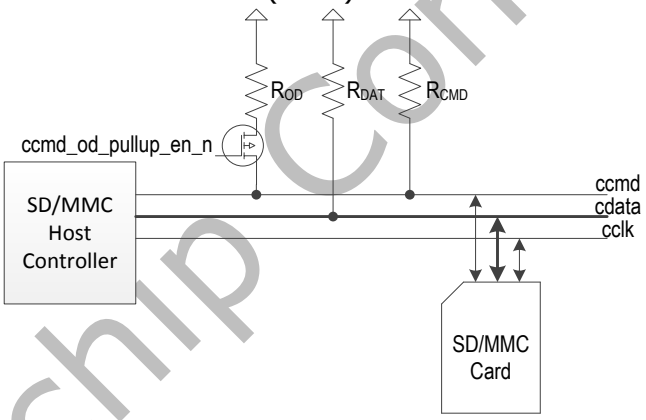


Fig. 1-10 SD/MMC Card Termination

1. Rcmd and Rod Calculation

The SD/MMC card enumeration happens at a very low frequency – 100-400KHz. Since the MMC bus is a shared bus between multiple cards, during enumeration open-drive mode is used to avoid bus conflict. Cards that drive 0 win over cards that drive "z". The pull-up in the command line pulls the bus to 1 when all cards drive "z". During normal data transfer, the host chooses only one card and the card driver switches to push-pull mode.

For example, if enumeration is done at 400KHz and the total bus capacitance is 200 pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz

R = 1/(2.2 * C * 100KHz)

= 1/(2.2 x 200 x 10**-12 x 400 x 10**3)

= 1/(17.6 x 10**-5)

= 5.68K

The ROD and RCMD should be adjusted in such a way that the effective pull-up is at the maximum 5.68K during enumeration. If there are only a few cards in the bus, a fixed RCMD resister is sufficient and there is no need for an additional ROD pull-up during enumeration. You should also ensure the effective pull-up will not violate the Iol rating of the drivers.

In SD mode, since each card has a separate bus, the capacitance is less, typically in the order of 20-30pf (host capacitance + card capacitance + trace + socket capacitance). For example, if enumeration is done at 400KHz and the total bus capacitance is 20pf, the pull-up needed during enumeration is:
2.2 RC = rise-time = 1/400KHz
R = 1/(2.2 * C * 100KHz)
= 1/(2.2 x 20 x 10**-12 x 400 x 10**3)
= 1/(1.76 x 10**-5)
= 56.8K
Therefore, a fixed 56.8K permanent Rcmd is sufficient in SD mode to enumerate the cards. The driver of the SD/MMC on the "command" port needs to be only a push-pull driver. During enumeration, the SD/MMC emulates an open-drain driver by driving only a 0 or a"z" by controlling the ccmd_out and ccmd_out_en signals.

1.6.3 Software/Hardware Restriction

Before issuing a new data transfer command, the software should ensure that the card is not busy due to any previous data transfer command. Before changing the card clock frequency, the software must ensure that there are no data or command transfers in progress.
If the card is enumerated in SDR50, or DDR50 mode, then the application must program the use_hold_reg bit[29] in the CMD register to 1'b0 (phase shift of cclk_in_drv = 0) or 1'b1 (phase shift of cclk_in_drv>0). If the card is enumerated in SDR12 or SDR25 mode, the application must program the use_hold_reg bit[29] in the CMD register to 1'b1.
This programming should be done for all data transfer commands and non-data commands that are sent to the card. When the use_hold_reg bit is programmed to 1'b0, the Host Controller bypasses the Hold Registers in the transmit path. The value of this bit should not be changed when a Command or Data Transfer is in progress. For more details on using use_hold_reg and the implementation requirements for meeting the Card input hold time, refer to "Recommended Usage" in following table.

Table 1-10 Recommended Usage of use_hold_reg

No.	Speed Mode	use_hold_reg	cclk_in (MHz)	clk_in_drv (MHz)	clk_divider	Phase shift
1	SDR104	1'b0	200	200	0	0
2	SDR104	1'b1	200	200	0	Tunable> 0
3	SDR50	1'b0	100	100	0	0
4	SDR50	1'b1	100	100	0	Tunable> 0
5	DDR50 (8bit)	1'b0	100	100	1	0
6	DDR50 (8bit)	1'b1	100	100	1	Tunable> 0
7	DDR50 (4bit)	1'b0	50	50	0	0
8	DDR50 (4bit)	1'b1	50	50	0	Tunable> 0
9	SDR25	1'b1	50	50	0	Tunable> 0
10	SDR12	1'b1	50	50	1	Tunable> 0

To avoid glitches in the card clock outputs, the software should use the following steps when changing the card clock frequency:
1) Before disable the clocks, ensure that the card is not busy due to any previous data command. To determine this, check for 0 in bit9 of STATUS register.
2) Update the Clock Enable register to disable all clocks. To ensure completion of any previous command before this update, send a command to the CIU to update the clock registers by setting:

- start_cmd bit
- "update clock registers only" bits
- "wait_previous data complete" bit

Wait for the CIU to take the command by polling for 0 on the start_cmd bit.
3) Set the start_cmd bit to update the Clock Divider and/or Clock Source registers, and send a command to the CIU in order to update the clock registers; wait for the CIU to take

the command.
4) Set start_cmd to update the Clock Enable register in order to enable the required clocks and send a command to the CIU to update the clock registers; wait for the CIU to take the command.
In non-DMA mode, while reading from a card, the Data Transfer Over (RINTSTS[3]) interrupt occurs as soon as the data transfer from the card is over. There still could be some data left in the FIFO, and the RX_WMark interrupt may or may not occur, depending on the remaining bytes in the FIFO. Software should read any remaining bytes upon seeing the Data Transfer Over (DTO) interrupt. While using the external DMA interface for reading from a card, the DTO interrupt occurs only after all the data is flushed to memory by the DMA interface unit.
While writing to a card in external DMA mode, if an undefined-length transfer is selected by setting the Byte Count Register to 0, the DMA logic will likely request more data than it will send to the card, since it has no way of knowing at which point the software will stop the transfer. The DMA request stops as soon as the DTO is set by the CIU.
If the software issues a controller_reset command by setting control register bit[0] to 1, all the CIU state machines are reset; the FIFO is not cleared. The DMA sends all remaining bytes to the host. In addition to a card-reset, if a FIFO reset is also issued, then:

- Any pending DMA transfer on the bus completes correctly
- DMA data read is ignored
- Write data is unknown(x)

Additionally, if dma_reset is also issued, any pending DMA transfer is abruptly terminated. When the DMA is used, the DMA controller channel should also be reset and reprogrammed.
If any of the previous data commands do not properly terminate, then the software should issue the FIFO reset in order to remove any residual data, if any, in the FIFO. After asserting the FIFO reset, you should wait until this bit is cleared.
One data-transfer requirement between the FIFO and host is that the number of transfers should be a multiple of the FIFO data width (32bits). For example, you want to write only 15 bytes to an SD/MMC card (BYTCNT), the host should write 16 bytes to the FIFO or program the DMA to do 16-byte transfers. The software can still program the Byte Count register to only 15, at which point only 15 bytes will be transferred to the card. Similarly, when 15 bytes are read from a card, the host should still read all 16 bytes from the FIFO. It is recommended that you not change the FIFO threshold register in the middle of data transfers.

1.6.4 Programming Sequence

1. Initialization
Following figure illustrates the initialization flow.

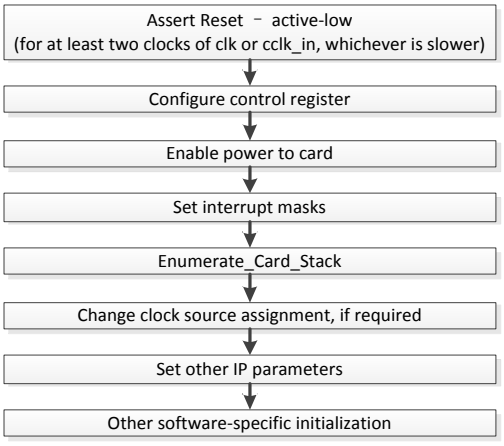


Fig. 1-11 Host Controller Initialization Sequence

Once the power and clocks are stable, reset_n should be asserted(active-low) for at least two clocks of clk or cclk_in, whichever is slower. The reset initializes the registers, ports, FIFO-pointers, DMA interface controls, and state-machines in the design. After power-on reset, the software should do the following:
1) Configure control register – For MMC mode, enable the open-drain pullup by setting

enable_OD_pullup(bit24) in the control register.

2) Enable power to cards – Before enabling the power, confirm that the voltage setting to the voltage regulators is correct. Enable power to the connected cards by setting the corresponding bit to 1 in the Power Enable register. Wait for the power ramp-up time.
3) Set masks for interrupts by clearing appropriate bits in the Interrupt Mask register. Set the global int_enable bit of the Control register. It is recommended that you write 0xffff_ffff to the Raw Interrupt register in order to clear any pending interrupts before setting the int_enable bit.
4) Enumerate card stack – Each card is enumerated according to card type; for details, refer to "Enumerated Card Stack". For enumeration, you should restrict the clock frequency to 400KHz.
5) Changing clock source assignment – set the card frequency using the clock-divider and clock-source registers; for details, refer to "Clock Programming". MMC cards operate at a maximum of 20MHz (at maximum of 52MHz in high-speed mode). SD mode operates at a maximum of 25MHz (at maximum of 50MHz in high-speed mode).
6) Set other parameters, which normally do not need to be changed with every command, with a typical value such as timeout values in cclk_out according to SD/MMC specifications.

- ResponseTimeOut = 0x64
- DataTimeOut = highest of one of the following:
- (10*((TAAC*Fop)+(100*NSAC))
- Host FIFO read/write latency from FIFO empty/full
- Set the debounce value to 25ms(default:0x0fffff) in host clock cycle units in the DEBNCE register.
- FIFO threshold value in bytes in the FIFOTH register.

2. Enumerated Card Stack

The card stack does the following:

- Enumerates all connected cards
- Sets the RCA for the connected cards
- Reads card-specific information
- Stores card-specific information locally

Enumeration depends on the operating mode of the SD/MMC card; the card type is first identified and the appropriate card enumeration routine is called.

1) Check if the card is connected.
2) Clear the card type register to set the card width as a single bit. For the given card number, clear the corresponding bits in the card_type register. Clear the register bit for a 1-bit, 4-bit bus width. For example, for card number=1, clear bit 0 and bit 16 of the card_type register.
3) Set clock frequency to FOD=400KHz, maximum – Program clock divider0 (bits 0-7 in the CLKDIV register) value to one-half of the cclk_in frequency divided by 400KHz. For example, if cclk_in is 20MHz, then the value is 20, 000/(2*400)=25.
4) Identify the card type; that is, SD, MMC, or SDIO.
   a. Send CMD5 first. If a response is received, then the card is SDIO
   b. If not, send CMD8 with the following Argument
      Bit[31:12] = 20'h0 //reserved bits
      Bit[11:8] = 4'b0001 //VHS value
      Bit[7:0] = 8'b10101010 //Preferred Check Pattern by SD2.0
   c. If Response is received the card supports High Capacity SD2.0 then send ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b1; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range
   d. If Response is received for ACMD41 then the card is SD. Otherwise the card is MMC.
   e. If response is not received for initial CMD8 then card does not support High Capacity SD2.0, then issue CMD0 followed by ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits

Bit[30] = 1'b0; //High Capacity Status
Bit[29:24] = 6'h0; //Reserved bits
Bit[23:0] = Supported Voltage Range

5) Enumerate the card according to the card type.
6) Use a clock source with a frequency = Fod (that is, 400KHz) and use the following enumeration command sequence:

- SD card – Send CMD0, CMD8, ACMD41, CMD2, CMD3.
- MMC – Send CMD0, CMD1, CMD2, CMD3.

3. Power Control

You can implement power control using the following registers, along with external circuitry:

- Control register bits card_voltage_a and card_voltage_b – Status of these bits is reflected at the IO pins. The bits can be used to generate or control the supply voltage that the memory cards require.
- Power enable register – Control power to individual cards.

Programming these two register depends on the implemented external circuitry. While turning on or off the power enable, you should confirm that power supply settings are correct. Power to all cards usually should be disabled while switching off the power.

4. Clock Programming

The Host Controller supports one clock sources. The clock to an individual card can be enabled or disabled. Registers that support this are:

- CLKDIV – Programs individual clock source frequency. CLKDIV limited to 0 or 1 is recommended.
- CLKSRC – Assign clock source for each card.
- CLKENA – Enables or disables clock for individual card and enables low-power mode, which automatically stops the clock to a card when the card is idle for more than 8 clocks.

The Host Controller loads each of these registers only when the start_cmd bit and the Update_clk_regs_only bit in the CMD register are set. When a command is successfully loaded, the Host Controller clears this bit, unless the Host Controller already has another command in the queue, at which point it gives an HLE(Hardware Locked Error).
Software should look for the start_cmd and the Update_clk_regs_only bits, and should also set the wait_prvdata_complete bit to ensure that clock parameters do not change during data transfer. Note that even though start_cmd is set for updating clock registers, the Host Controller does not raise a command_done signal upon command completion.
The following shows how to program these registers:

1) Confirm that no card is engaged in any transaction; if there is a transaction, wait until it finishes.
2) Stop all clocks by writing xxxx0000 to the CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
3) Program the CLKDIV and CLKSRC registers, as required. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
4) Re-enable all clocks by programming the CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.

5. No-Data Command With or Without Response Sequence

To send any non-data command, the software needs to program the CMD register @0x2C and the CMDARG register @0x28 with appropriate parameters. Using these two registers, the Host Controller forms the command and sends it to the command bus. The Host Controller reflects the errors in the command response through the error bits of the RINTSTS register.
When a response is received – either erroneous or valid – the Host Controller sets the command_done bit in the RINTSTS register. A short response is copied in Response Register0, while along response is copied to all four response registers @0x30, 0x34, 0x38, and 0x3C. The Response3 register bit 31 represents the MSB, and the Response0 register

bit 0 represents the LSB of a long response.
For basic commands or non-data commands, follow these steps:
1) Program the Command register @0x28 with the appropriate command argument parameter.
2) Program the Command register @0x2C with the settings in following table.

Table 1-11 Command Settings for No-Data Command

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on CMD register
update_clk_regs_only	0	No clock parameters update command
data_expected	0	No data command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
cmd_index	command-index	-
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
wait_prvdata_complete	1	Before sending command on command line, host should wait for completion of any data command in process, if any (recommended to always set this bit, unless the current command is to query status or stop data transfer when transfer is in progress)
check_response_crc	1	If host should crosscheck CRC of response received

1) Wait for command acceptance by host. The following happens when the command is loaded into the Host Controller:
   - Host Controller accepts the command for execution and clears the start_cmd bit in the CMD register, unless one command is in process, at which point the Host Controller can load and keep the second command in the buffer.
   - If the Host Controller is unable to load the command – that is, a command is already in progress, a second command is in the buffer, and a third command is attempted – then it generates an HLE (hardware-locked error).
2) Check if there is an HLE.
3) Wait for command execution to complete. After receiving either a response from a card or response timeout, the Host Controller sets the command_done bit in the RINTSTS register. Software can either poll for this bit or respond to a generated interrupt.

4) Check if response_timeout error, response_CRC error, or response error is set. This can be done either by responding to an interrupt raised by these errors or by polling bits 1, 6, and 8 from the RINTSTS register @0x44. If no response error is received, then the response is valid. If required, the software can copy the response from the response registers @0x30-0x3C.

Software should not modify clock parameters while a command is being executed.

6. Data Transfer Commands

Data transfer commands transfer data between the memory card and the Host Controller. To send a data command, the Host Controller needs a command argument, total data size, and block size. Software can receive or send data through the FIFO.

Before a data transfer command, software should confirm that the card is not busy and is in a transfer state, which can be done using the CMD13 and CMD7 commands, respectively. For the data transfer commands, it is important that the same bus width that is programmed in the card should be set in the card type register @0x18.

The Host Controller generates an interrupt for different conditions during data transfer, which are reflected in the RINTSTS register @0x44 as:

1) Data_Transfer_Over (bit 3) – When data transfer is over or terminated. If there is a response timeout error, then the Host Controller does not attempt any data transfer and the "Data Transfer Over" bit is never set.
2) Transmit_FIFO_Data_request (bit 4) – FIFO threshold for transmitting data was reached; software is expected to write data, if available, in FIFO.
3) Receive_FIFO_Data_request (bit 5) – FIFO threshold for receiving data was reached; software is expected to read data from FIFO.
4) Data starvation by Host timeout (bit 10) – FIFO is empty during transmission or is full during reception. Unless software writes data for empty condition or reads data for full condition, the Host Controller cannot continue with data transfer. The clock to the card has been stopped.
5) Data read timeout error (bit 9) – Card has not sent data within the timeout period.
6) Data CRC error (bit 7) – CRC error occurred during data reception.
7) Start bit error (bit 13) – Start bit was not received during data reception.
8) End bit error (bit 15) – End bit was not received during data reception or for a write operation; a CRC error is indicated by the card.

Conditions 6, 7, and 8 indicate that the received data may have errors. If there was a response timeout, then no data transfer occurred.

7. Single-Block or Multiple-Block Read

Steps involved in a single-block or multiple-block read are:

1) Write the data size in bytes in the BYTCNT register @0x20.
2) Write the block size in bytes in the BLKSIZ register @0x1C. The Host Controller expects data from the card in blocks of size BLKSIZ each.
3) Program the CMDARG register @0x28 with the data address of the beginning of a data read.
4) Program the Command register with the parameters listed in following table. For SD and MMC cards, use CMD17 for a single-block read and CMD18 for a multiple-block read. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 1-12 Command Setting for Single or Multiple-Block Read

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No.0)

Parameter	Value	Description
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	0	Read from card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0- Sends command immediately 1- Sends command after previous data transfer ends
check_response_crc	1	0- Host Controller should not check response CRC 1- Host Controller should check response CRC

After writing to the CMD register, the Host Controller starts executing the command; when the command is sent to the bus, the command_done interrupt is generated.

- Software should look for data error interrupts; that is, bits 7, 9, 13, and 15 of the RINTSTS register. If required, software can terminate the data transfer by sending a STOP command.
- Software should look for Receive_FIFO_Data_request and/or data starvation by host timeout conditions. In both cases, the software should read data from the FIFO and make space in the FIFO for receiving more data.
- When a Data_Transfer_Over interrupt is received, the software should read the remaining data from the FIFO.

8. Single-Block or Multiple-Block Write

Steps involved in a single-block or multiple-block write are:

1) Write the data size in bytes in the BYTCNT register @0x20.
2) Write the block size in bytes in the BLKSIZ register @0x1C; the Host Controller sends data in blocks of size BLKSIZ each.
3) Program CMDARG register @0x28 with the data address to which data should be written.
4) Write data in the FIFO; it is usually best to start filling data the full depth of the FIFO.
5) Program the Command register with the parameters listed in following table.

Table 1-13 Command Settings for Single or Multiple-Block Write

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on CMD register
update_clk_regs_only	0	No clock parameters update command

Parameter	Value	Description
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	1	Write to card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0- Sends command immediately 1- Sends command after previous data transfer ends
check_response_crc	1	0- Host Controller should not check response CRC 1- Host Controller should check response CRC

After writing to the CMD register, Host Controller starts executing a command; when the command is sent to the bus, a command_done interrupt is generated.

- Software should look for data error interrupts; that is, for bits 7, 9, and 15 of the RINTSTS register. If required, software can terminate the data transfer by sending the STOP command.
- Software should look for Transmit_FIFO_Data_Request and/or timeout conditions from data starvation by the host. In both cases, the software should write data into the FIFO.
- When a Data_Transfer_Over interrupt is received, the data command is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the Host Controller should send the STOP command, if necessary. Completion of the AUTO-STOP command is reflected by the Auto_command_done interrupt – bit 14 of the RINTSTS register. A response to AUTO_STOP is stored in RESP1 @0x34.

9. Stream Read

A stream read is like the block read mentioned in "Single-Block or Multiple-Block Read", except for the following bits in the Command register:

transfer_mode = 1; //Stream transfer
cmd_index = CMD20;

A stream transfer is allowed for only a single-bit bus width.

10. Stream Write

A stream write is exactly like the block write mentioned in "Single-Block or Multiple-Block Write", except for the following bits in the Command register:

transfer_mode = 1;//Stream transfer
cmd_index = CMD11;

In a stream transfer, if the byte count is 0, then the software must send the STOP command. If the byte count is not 0, then when a given number of bytes completes a transfer, the Host Controller sends the STOP command. Completion of this AUTO_STOP command is reflected by the Auto_command_done interrupt. A response to an AUTO_STOP is stored in the RESP1 register@0x34.
A stream transfer is allowed for only a single-bit bus width.

11. Packed Commands

In order to reduce overhead, read and write commands can be packed in groups of commands—either all read or all write—that transfer the data for all commands in the group in one transfer on the bus.
Packed commands can be of two types:

- Packed Write: CMD23 →CMD25
- Packed Read: CMD23 → CMD25 → CMD23 → CMD18

Packed commands are put in packets by the application software and are transparent to the core.

12. Sending Stop or Abort in Middle of Transfer

The STOP command can terminate a data transfer between a memory card and the Controller, while the ABORT command can terminate an I/O data transfer for only the SDIO_IOONLY and SDIO_COMBO cards.

- Send STOP command – Can be sent on the command line while a data transfer is in progress; this command can be sent at any time during a data transfer.

You can also use an additional setting for this command in order to set the Command register bits (5-0) to CMD12 and set bit 14 (stop_abort_cmd) to 1. If stop_abort_cmd is not set to 1, the Controller does not know that the user stopped a data transfer. Reset bit 13 of the Command register (wait_prvdata_complete) to 0 in order to make the Controller send the command at once, even though there is a data transfer in progress.

- Send ABORT command – Can be used with only an SDIO_IOONLY or SDIO_COMBO card. To abort the function that is transferring data, program the function number in ASx bits (CCCR register of card, address 0x06, bits (0-2) using CMD52.

13. Suspend or Resume Sequence

In an SDIO card, the data transfer between an I/O function and the Controller can be temporarily halted using the SUSPEND command; this may be required in order to perform a high-priority data transfer with another function. When desired, the data transfer can be resumed using the RESUME command.
The following functions can be implemented by programming the appropriate bits in the CCCR register (Function 0) of the SDIO card. To read from or write to the CCCR register, use the CMD52 command.

- SUSPEND data transfer – Non-data command

1) Check if the SDIO card supports the SUSPEND/RESUME protocol; this can be done through the SBS bit in the CCCR register @0x08 of the card.
2) Check if the data transfer for the required function number is in process; the function number that is currently active is reflected in bits 0-3 of the CCCR register @0x0D. Note that if the BS bit (address 0xc::bit 0) is 1, then only the function number given by the FSx bits is valid.
3) To suspend the transfer, set BR (bit 2) of the CCCR register @0x0C.
4) Poll for clear status of bits BR (bit 1) and BS (bit 0) of the CCCR @0x0C. The BS (Bus Status) bit is 1 when the currently-selected function is using the data bus; the BR (Bus Release) bit remains 1 until the bus release is complete. When the BR and BS bits are 0, the data transfer from the selected function has been suspended.

- RESUME data transfer – This is a data command

1) Check that the card is not in a transfer state, which confirms that the bus is free for data transfer.
2) If the card is in a disconnect state, select it using CMD7. The card status can be retrieved in response to CMD52/CMD53 commands.
3) Check that a function to be resumed is ready for data transfer; this can be confirmed by reading the RFx flag in CCCR @0x0F. If RF = 1, then the function is ready for data transfer.

4) To resume transfer, use CMD52 to write the function number at FSx bits (0-3) in the CCCR register @0x0D. Form the command argument for CMD52 and write it in CMDARG @0x28.
5) Write the block size in the BLKSIZ register @0x1C; data will be transferred in units of this block size.
6) Write the byte count in the BYTCNT register @0x20. This is the total size of the data; that is, the remaining bytes to be transferred. It is the responsibility of the software to handle the data.
7) Program Command register; similar to a block transfer.
8) When the Command register is programmed, the command is sent and the function resumes data transfer. Read the DF flag (Resume Data Flag). If it is 1, then the function has data for the transfer and will begin a data transfer as soon as the function or memory is resumed. If it is 0, then the function has no data for the transfer.
9) If the DF flag is 0, then in case of a read, the Host Controller waits for data. After the data timeout period, it gives a data timeout error.

6. Read_Wait Sequence

Read_wait is used with only the SDIO card and can temporarily stall the data transfer—either from function or memory—and allow the host to send commands to any function within the SDIO device. The host can stall this transfer for as long as required. The Host Controller provides the facility to signal this stall transfer to the card. The steps for doing this are:

1) Check if the card supports the read_wait facility; read SRW (bit 2) of the CCCR register @0x08. If this bit is 1, then all functions in the card support the read_wait facility. Use CMD52 to read this bit.
2) If the card supports the read_wait signal, then assert it by setting the read_wait (bit 6) in the CTRL register @0x00.
3) Clear the read_wait bit in the CTRL register.

14. Controller/DMA/FIFO Reset Usage

- Controller reset – Resets the controller by setting the controller_reset bit (bit 0) in the CTRL register; this resets the CIU and state machines, and also resets the BIU-to-CIU interface. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- FIFO reset - Resets the FIFO by setting the fifo_reset bit (bit 1) in the CTRL register; this resets the FIFO pointers and counters of the FIFO. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.

In external DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO underrun or overrun errors in the RAWINTS register caused by the DMA transfers after the FIFO was reset.

15. Card Read Threshold

When an application needs to perform a Single or Multiple Block Read command, the application must program the CardThrCtl register with the appropriate Card Read Threshold size (CardRdThreshold) and set the Card Read Threshold Enable (CardRdThrEnable) bit to 1'b1. This additional programming ensures that the Host controller sends a Read Command only if there is space equal to the CardRDThreshold available in the Rx FIFO. This in turn ensures that the card clock is not stopped in the middle a block of data being transmitted from the card. The Card Read Threshold can be set to the block size of the transfer, which guarantees that there is a minimum of one block size of space in the RxFIFO before the controller enables the card clock.The Card Read Threshold is required when the Round Trip Delay is greater than 0.5cclk_in period.

16. Error Handling

The Host Controller implements error checking; errors are reflected in the RAWINTS register@0x44 and can be communicated to the software through an interrupt, or the software can poll for these bits. Upon power-on, interrupts are disabled (int_enable in the CTRL register is 0), and all the interrupts are masked (bits 0-31 of the INTMASK register; default is 0).

Error handling:

- Response and data timeout errors – For response timeout, software can retry the command. For data timeout, the Host Controller has not received the data start bit – either for the first block or the intermediate block – within the timeout period, so software can either retry the whole data transfer again or retry from a specified block onwards. By reading the contents of the TCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors – Set when an error is received during response reception. In this case, the response that copied in the response registers is invalid. Software can retry the command.
- Data errors – Set when error in data reception are observed; for example, data CRC, start bit not found, end bit not found, and so on. These errors could be set for any block-first block, intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORT command and retry the command for either whole data or partial data.
- Hardware locked error – Set when the Host Controller cannot load a command issued by software. When software sets the start_cmd bit in the CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
- FIFO underrun/overrun error – If the FIFO is full and software tries to write data in the FIFO, then an overrun error is set. Conversely, if the FIFO is empty and the software tries to read data from the FIFO, an underrun error is set. Before reading or writing data in the FIFO, the software should read the fifo_empty or fifo_full bits in the Status register.
- Data starvation by host timeout – Raised when the Host Controller is waiting for software intervention to transfer the data to or from the FIFO, but the software does not transfer within the stipulated timeout period. Under this condition and when a read transfer is in process, the software should read data from the FIFO and create space for further data reception. When a transmit operation is in process, the software should fill data in the FIFO in order to start transferring data to the card.
- CRC Error on Command – If a CRC error is detected for a command, the CE-ATA device does not send a response, and a response timeout is expected from the Host Controller. The ATA layer is notified that an MMC transport layer error occurred.

Notes: During a multiple-block data transfer, if a negative CRC status is received from the device, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register. It then continues further data transmission until all the bytes are transmitted.

1.6.5 Voltage Switching

The Host Controller supports SD 3.0 Ultra High Speed (UHS-1) and is capable of voltage switching in SD-mode, which can be applied to SD High-Capacity (SDHC) and SD Extended Capacity (SDXC) cards. UHS-1 supports only 4-bit mode.
However, whether the IO voltage of 1.8v supported or not is depended on the SoC design.
SD 3.0 UHS-1 supports the following transfer speed modes for UHS-50 and/or UHS-104 cards:

- DS – default-speed up to 25MHz, 3.3V signaling
- HS – high-speed up to 50MHz, 3.3V signaling
- SDR12 – SDR up to SDR 25MHz, 1.8V signaling
- SDR25 – SDR up to 50MHz, 1.8V signaling
- SDR50 – SDR up to 100MHz, 1.8V signaling
- DDR50 – DDR up to 50MHz, 1.8V signaling

Voltage selection can be done in only SD mode. The first CMD0 selects the bus mode-either SD mode or SPI mode. The card must be in SD mode in order for 1.8V signaling mode to apply, during which time the card cannot be switched to SPI mode or 3.3V signaling without a power cycle.
If the System BIOS in an embedded system already knows that it is connected to an SD 3.0 card, then the driver programs the Controller to initiate ACMD41. The software knows from the response of ACMD41 whether or not the card supports voltage switching to 1.8V.

- If bit 32 of ACMD41 response is 1'b1: card supports voltage switching and next

command-CMD11-invokes voltage switching sequence. After CMD11 is started, the software must program the IO voltage selection register based on the soc architecture.

- If bit 32 of ACMD41 response is 1'b0: card does not support voltage switching and CMD11 should not be started.

If the card and host controller accept voltage switching, then they support UHS-1 modes of data transfer. After the voltage switch to 1.8V, SDR12 is the default speed.
Since the UHS-1 can be used in only 4-bit mode, the software must start ACMD6 and change the card data width to 4-bit mode; ACMD6 is driven in any of the UHS-1 speeds. If the host wants to select the DDR mode of data transfer, then the software must program the DDR_REG register in the CSR space with the appropriate card number.
To choose from any of the SDR or DDR modes, appropriate values should be programmed in the CLKDIV register.

1. Voltage Switch Operation

The Voltage Switch operation must be performed in SD mode only.

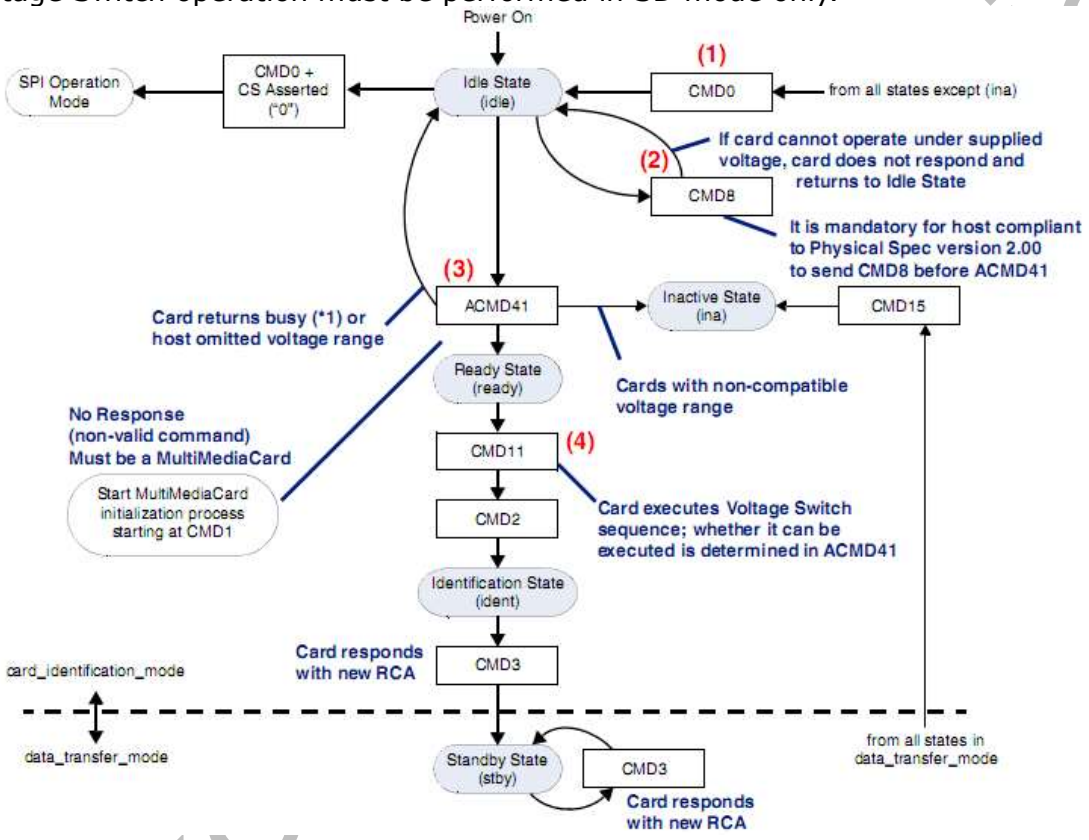


Fig. 1-12 Voltage Switching Command Flow Diagram

The following outlines the steps for the voltage switch programming sequence

1) Software Driver starts CMD0, which selects the bus mode as SD.
2) After the bus is in SD card mode, CMD8 is started in order to verify if the card is compatible with the SD Memory Card Specification, Version 2. 00. CMD8 determines if the card is capable of working within the host supply voltage specified in the VHS (19:16) field of the CMD; the card supports the current host voltage if a response to CMD8 is received.
3) ACMD 41 is started. The response to this command informs the software if the card supports voltage switching; bits 38, 36, and 32 are checked by the card argument of ACMD41; refer to following figure.

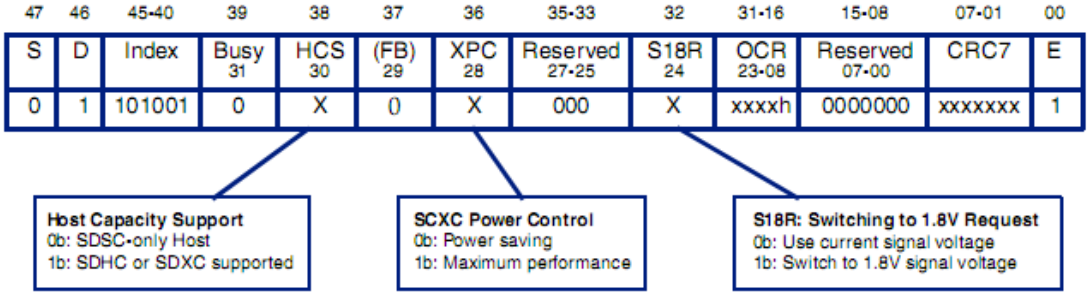


Fig. 1-13 ACMD41 Argument

- Bit 30 informs the card if host supports SDHC/SDXC or not; this bit should be set to 1'b1.
- Bit 28 can be either 1 or 0.
- Bit 24 should be set to 1'b1, indicating that the host is capable of voltage switching; refer to following figure.

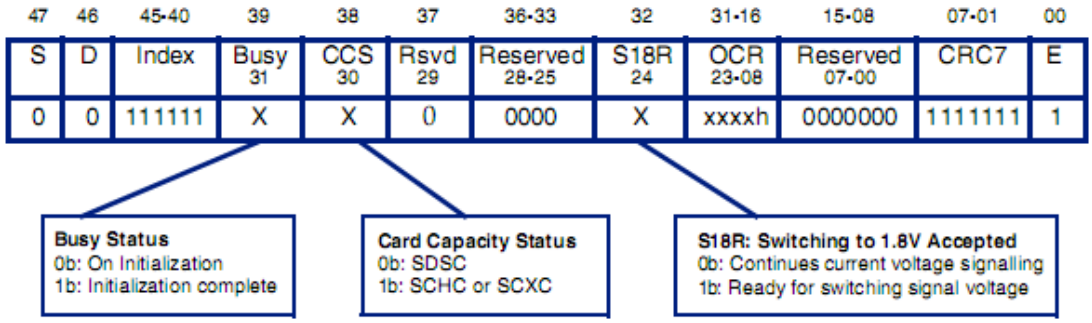


Fig. 1-14 ACMD41 Response(R3)

- Bit 30 – If set to 1'b1, card supports SDHC/SDXC; if set to 1'b0, card supports only SDSC
- Bit 24 – If set to 1'b1, card supports voltage switching and is ready for the switch
- Bit 31 – If set to 1'b1, initialization is over; if set to 1'b0, means initialization in process

4) If the card supports voltage switching, then the software must perform the steps discussed for either the "Voltage Switch Normal Scenario" or the "Voltage Switch Error Scenario".

2. Voltage Switch Normal Scenario

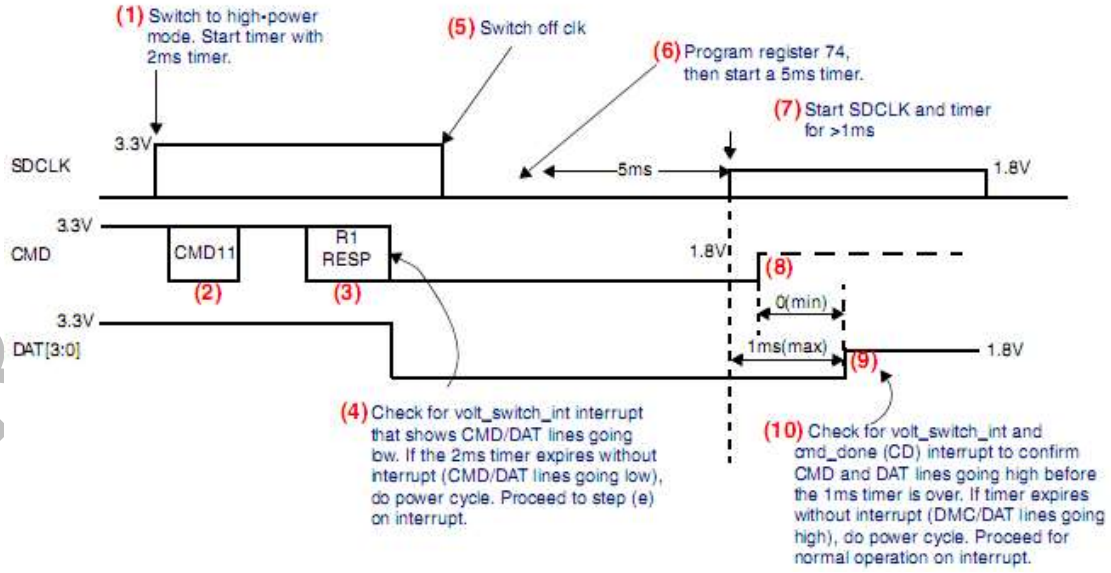


Fig. 1-15 Voltage Switch Normal Scenario

1) The host programs CLKENA—cclk_low_power register—with zero (0) for the corresponding card, which makes the host controller move to high-power mode. The application should start a timer with a recommended value of 2ms; this value of 2 ms is determined as below:Total clk required for CMD11 = 48 clks
Total clk required for RESP R1 = 48 clks

Maximum clk delay between MCD11 end to start of RESP1 = 60 clks
Total = 48+48 + 60 = 160
Minimum frequency during enumeration is 100 KHz; that is, 10us
Total time = 160 * 10us = 1600us = 1. 6ms ~ 2ms

2) The host issues CMD11 to start the voltage switch sequence. Set bit 28 to 1'b1 in CMD when setting CMD11; for more information on setting bits, refer to "Boot Operation".
3) The card returns R1 response; the host controller does not generate cmd_done interrupt on receiving R1 response.
4) The card drives CMD and DAT [3:0] to low immediately after the response. The host controller generates interrupt (VOLT_SWITCH_INT) once the CMD or DAT [3:0] line goes low. The application should wait for this interrupt. If the 2ms timer expires without an interrupt (CMD/DAT lines going low), do a power cycle.

Note: Before doing a power cycle, switch off the card clock by programming CLKENA register

Proceed to step (5) on getting an interrupt (VOLT_SWITCH_INT).

Note: This interrupt must be cleared once this interrupt is received. Additionally, this interrupt should not be masked during the voltage switch sequence.

If the timer expires without interrupt (CMD/DAT lines going low), perform a power cycle.
Proceed to step (5) on interrupt.

1) Program the CLKENA, cclk_enable register, with 0 for the corresponding card; the host stops supplying SDCLK.
2) Program Voltage register to the required values for the corresponding card. The application should start a timer > 5ms.
3) After the 5ms timer expires, the host voltage regulator is stable. Program CLKENA, cclk_enable register, with 1 for the corresponding card; the host starts providing SDCLK at 1. 8V; this can be at zero time after Voltage register has been programmed. When the CLKENA register is programmed, the application should start another timer > 1ms.
4) By detecting SDCLK, the card drives CMD to high at 1. 8V for at least one clock and then stops driving (tri-state); CMD is triggered by the rising edge of SDCLK (SDR timing).
5) If switching to 1. 8V signaling is completed successfully, the card drives DAT [3:0] to high at 1. 8V for at least one clock and then stops driving (tri-state); DAT [3:0] is triggered by the rising edge of SDCLK (SDR timing). DAT[3:0] must be high within 1ms from the start of SDCLK.
6) The host controller generates a voltage switch interrupt (VOLT_SWITCH_INT) and a command done (CD) interrupt once the CMD and DAT[3:0] lines go high. The application should wait for this interrupt to confirm CMD and DAT lines going high before the 1ms timer is done.

If the timer expires without the voltage switch interrupt (VOLT_SWITCH_INT), a power cycle should be performed. Program the CLKENA register to stop the clock for the corresponding card number. Wait for the cmd_done (CD) interrupt. Proceed for normal operation on interrupt. After the sequence is completed, the host and the card start communication in SDR12 timing.

3. Voltage Switch Error Scenario

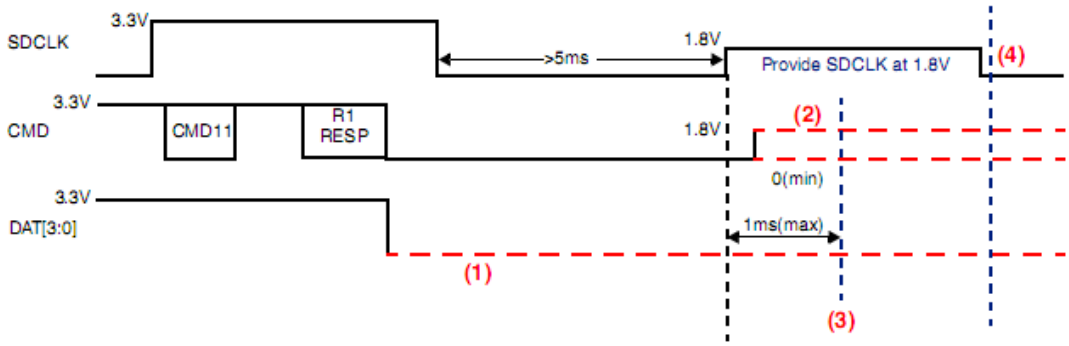


Fig. 1-16 Voltage Switch Error Scenario

1) If the interrupt (VOLT_SWITCH_INT) does not come, then the 2 ms timer should time out and a power cycle should be initiated.

Note: Before performing a power cycle, switch off the card clock by programming CLKENA register; no cmd_done (CD) interrupt is generated.

Additionally, if the card detects a voltage error at any point in between steps (5) and (7) in the card keeps driving DAT[3:0] to low until card power off.

2) CMD can be low or tri-state.
3) The host controller generates a voltage switch interrupt once the CMD and DAT[3:0] lines go high. The application should check for an interrupt to confirm CMD and DAT lines going high before the 1 ms timer is done.

If the 1 ms timer expires without interrupt (VOLT_SWITCH_INT) and cmd_done (CD), a power cycle should be performed. Program the CLKENA register to stop SDCLK of the corresponding card. Wait for the cmd_done interrupt. Proceed for normal operation on interrupt.

4) If DAT[3:0] is low, the host drives SDCLK to low and then stops supplying the card power.

Note: The card checks voltages of its own regulator output and host signals to ensure they are less than 2. 5V. Errors are indicated by (1) and (2).

- If voltage switching is accepted by the card, the default speed is SDR12.
- Command Done is given:
  - If voltage switching is properly done, CMD and DAT line goes high.
  - If switching is not complete, the 1ms timer expires, and the card clk is switched off.

Note: No other CMD should be driven before the voltage switching operation is completed and Command Done is received.

- The application should use CMD6 to check and select the particular function; the function appropriate-speed should be selected.

After the function switches, the application should program the correct value in the CLKDIV register, depending on the function chosen. Additionally, if Function 0x4 of the Access mode is chosen—that is, DDR50, then the application should also program 1'b1 in DDR_REG for the card number that has been selected for DDR50 mode.

1.6.6 Back-End Power

Each device needs one bit to control the back-end power supply for an embedded device; this bit does not control the VDDH of the host controller. A back_end_power register enables software programming for back-end power. The value on this register is output to the back_end_power signal, which can be used to switch power on and off the embedded device.

1.6.7 DDR Operation

1. 4-bit DDR Programming Sequence

DDR programming should be done only after the voltage switch operation has completed. The following outlines the steps for the DDR programming sequence:

1) Once the voltage switch operation is complete, the user must program voltage selection register to the required values for the corresponding card.

- To start a card to work in DDR mode, the application must program a bit of the newly defined UHS_REG[16] register with a value of 1'b1.
- The bit that the user programs depends on which card is to be accessed in DDR mode.

2) To move back to SDR mode, a power cycle should be run on the card—putting the card in SDR12 mode—and only then should UHS_REG[16]be set back to 1'b0 for the appropriate card.

2. 8-bit DDR Programming Sequence

The following outlines the steps for the 8-bit DDR programming sequence:

1) The cclk_in signal should be twice the speed of the required cclk_out. Thus, if the cclk_out signal is required to be 50 MHz, the cclk_in signal should be 100 MHz.
2) The CLKDIV register should always be programmed with a value higher than zero (0); that is, a clock divider should always be used for 8-bit DDR mode.
3) The application must program the UHS_REG[16] register (DDR_REG bits) by assigning it with a value of 1 for the bit corresponding to the card number; this causes the selected card to start working in DDR mode.
4) Depending on the card number, the CTYPE [31:16] bits should be set in order to make the host work in the 8-bit mode.

3. eMMC4.5 DDR START Bit

The eMMC4.5 changes the START bit definition in the following manner:

1) Receiver samples the START bit on the rising edge.
2) On the next rising edge after sampling the START bit, the receiver must sample the data.
3) Removes requirement of the START bit and END bit to be high for one full cycle.

Notes: The Host Controller does not support a START bit duration higher than one clock cycle. START bit durations of one or less than one clock cycle are supported and can be defined at the time of startup by programming the EMMC_DDR_REG register.

Following figure illustrates cases for the definition change of the START bit with eMMC4.5; it also illustrates how some of these cases can fail in sampling when higher-value delays are considered for I/O PADs.

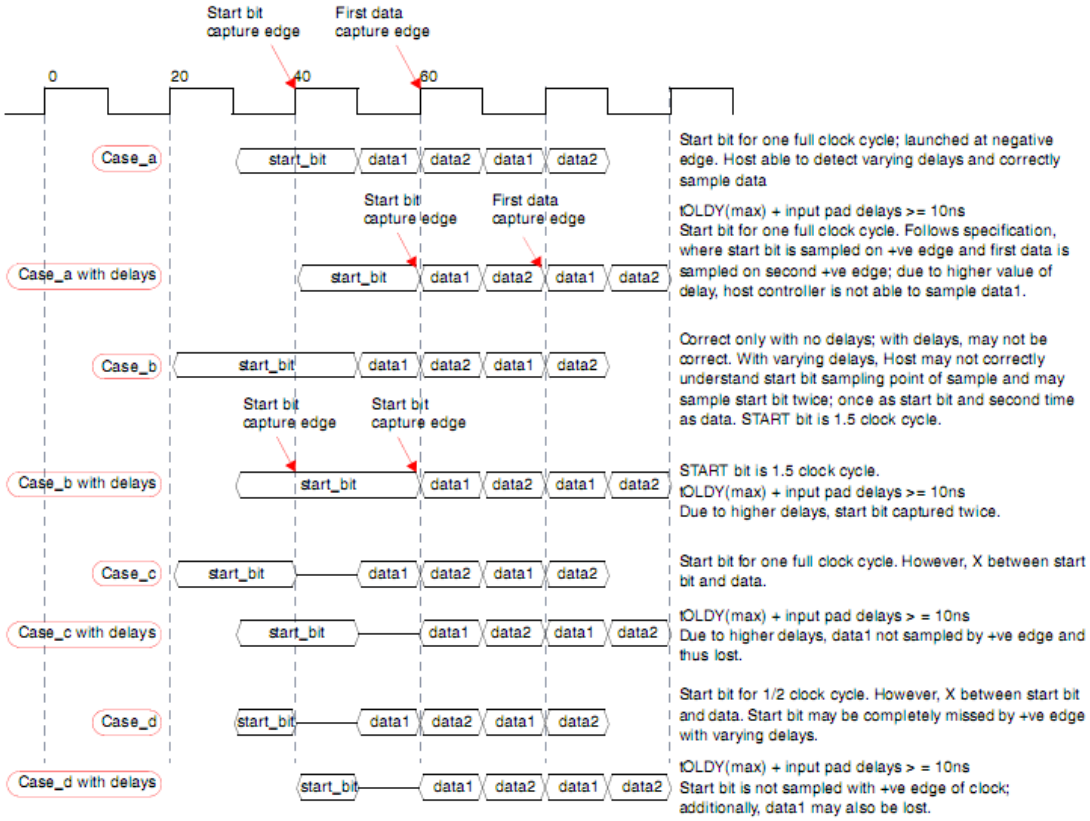


Fig. 1-17 CASES for eMMC 4.5 START bit

4. Reset Command/Moving from DDR50 to SDR12

To reset the mode of operation from DDR50 to SDR12, the following sequence of operations has to be done by the application:

1) Issue CMD0.

When CMD0 is received, the card changes from DDR50 to SDR12.

2) Program the CLKDIV register with an appropriate value.
3) Set DDR_REG to 0.

Note: The Voltage register should not be programmed to 0 while switching from DDR50 to SDR12, since the card is still operating in 1.8V mode after receiving CMD0.

1.6.8 H/W Reset Operation

When the RST_n signal goes low, the card enters a pre-idle state from any state other than the inactive state.

H/W Reset Programming Sequence

The following outlines the steps for the H/W reset programming sequence:

11) Program CMD12 to end any transfer in process.
12) Wait for DTO, even if no response is sent back by the card.
13) Set the following resets:

- DMA reset– CTRL[2]
- FIFO reset – CTRL[1] bits

Note: The above steps are required only if a transfer is in process.

14) Program the CARD_RESET register with a value of 0; this can be done at any time when the card is connected to the controller. This programming asserts the RST_n

signal and resets the card.
15) Wait for minimum of 1 µs or cclk_in period, whichever is greater
16) After a minimum of 1 µs, the application should program a value of 0 into the CARD_RESET register. This de-asserts the RST_n signal and takes the card out of reset.
17) The application can program a new CMD only after a minimum of 200 µs after the de-assertion of the RST_n signal, as per the MMC 4.41 standard.

Note: For backward compatibility, the RST_n signal is temporarily disabled in the card by default. The host may need to set the signal as either permanently enabled or permanently disabled before it uses the card.

1.6.9 FBE Scenarios

An FBE occurs due to an AHB error response on the AHB bus. This is a system error, so the software driver should not perform any further programming to the Host. The only recovery mechanism from such scenarios is to do one of the following:

- Issue a hard reset by asserting the reset_n signal
- Do a program controller reset by writing to the CTRL[0] register

1. FIFO Overflow and Underflow

During normal data transfer conditions, FIFO overflow and underflow will not occur. However if there is a programming error, then FIFO overflow/underflow can result. For example, consider the following scenarios.

- For transmit: PBL=4, Tx watermark = 1. For the above programming values, if the FIFO has only one location empty, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pushes into the FIFO. This will result in a FIFO overflow interrupt.
- For receive: PBL=4, Rx watermark = 1. For the above programming values, if the FIFO has only one location filled, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pops to the FIFO. This will result in a FIFO underflow interrupt.

The driver should ensure that the number of bytes to be transferred as indicated in the descriptor should be a multiple of 4bytes with respect to H_DATA_WIDTH=32. For example, if the BYTCNT = 13, the number of bytes indicated in the descriptor should be 16 for H_DATA_WIDTH=32.

2. Programming of PBL and Watermark Levels

The DMAC performs data transfers depending on the programmed PBL and threshold values.

Table 1-14 PBL and Watermark Levels

PBL (Number of transfers)	Tx/Rx Watermark Value
1	greater than or equal to 1
4	greater than or equal to 4
8	greater than or equal to 8
16	greater than or equal to 16
32	greater than or equal to 32
64	greater than or equal to 64
128	greater than or equal to 128
256	greater than or equal to 256

1.6.10 Variable Delay/Clock Generation

Variable delay mechanism for the cclk_in_drv is optional, but it can be useful in order to meet a range of hold-time requirements across modes. Variable delay mechanism for the cclk_in_sample is mandatory and is required to achieve the correct sampling point for data. cclk_in/cclk_in_sample/ cclk_in_drv is generated by Clock Generation Unit (CLKGEN) with variable delay mechanism, which includes Phase Shift Unit and Delay Line Unit selectable. The Phase Shift Unit can shift cclk_in_sample/cclk_in_drv by 0/90/180/270-degree relative to cclk_in, controlled by sample_degree/drv_degree.

The Delay Line Unit can shift cclk_in_sample/cclk_in_drv in the unit of 40ps~80ps for every delay element. The delay unit number is determined by sample_delaynum/drv_delaynum, and enabled by sample_sel/drv_sel.

cclk_in is generated by cclkin divided by 2. cclk_in_drv and cclk_in_sample clocks are phase-shifted with delayed versions of cclk_in. All clocks are recommended to have a 50%

duty cycle; DDR modes must have 50% duty cycles.
The architecture is as follows.

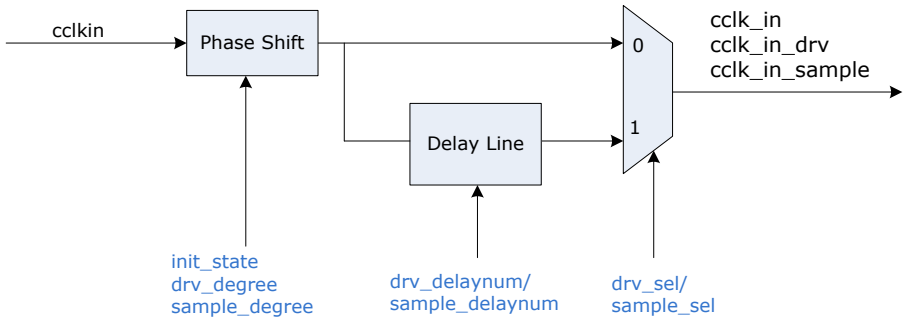


Fig. 1-18 Clock Generation Unit

The control signals for different Host Controller instance are shown as follows:

Table 1-15 Configuration for SDMMC Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_SDMMC_CON0[0]	0	Soft initial state for phase shift.
drv_degree [1:0]	CRU_SDMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum [7:0]	CRU_SDMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree [1:0]	CRU_SDMMC_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum [7:0]	CRU_SDMMC_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDMMC_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

Table 1-16 Configuration for SDIO Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_SDIO0_CON0[0]	0	Soft initial state for phase shift.
drv_degree [1:0]	CRU_SDIO0_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum [7:0]	CRU_SDIO0_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDIO0_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree [1:0]	CRU_SDIO0_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree

Signal Name	Source	Default	Description
			2: 180-degree 3: 270-degree
sample_delaynum [7:0]	CRU_SDIO0_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDIO0_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

The following outlines the steps for clock generation sequence:

1) Assert init_state to soft reset the CLKGEN.
2) Configure drv_degree/sample_degree.
3) If fine adjustment required, delay line can be used by configuring drv_delaynum/sample_delaynum and drv_sel/sample_sel.
4) Dis-assert init_state to start CLKGEN.

1.6.11 Variable Delay Tuning

Tuning is defined by SD and MMC cards to determine the correct sampling point required for the host, especially for the speed modes SDR104 and HS200 where the output delays from the cards can be up to 2 UI. Tuning is required for other speed modes-such as DDR50-even though the output delay from the card is less than one cycle.
Command for tuning is different for different cards.

- SD Memory Card:
  - CMD19 – SD card for SDR50 and SDR104 speed modes. Tuning data is defined by card specifications.
  - CMD6 – SD card for speed modes not supporting CMD19. Tuning data is the 64byte SD status.
- Multimedia Card:
  - CMD21 – MMC card for HS200 speed mode. Tuning data is defined by card specifications.
  - CMD8 – MMC card for speed modes not supporting CMD21. Tuning data is 512 byte ExtCSD data.

The following is the procedure for variable delay tuning:

1) Set a phase shift of 0-degree on cclk_in_sample.
2) Send the Tuning command to the card; the card in turn sends an R1 response on the CMD line and tuning data on the DAT line.
3) If the host sees any of the errors—start bit error, data crc error, end bit error, data read time-out, response crc error, response error—then the sampling point is incorrect.
4) Send CMD12 to bring the host controller state machines to idle.
   - The card may treat CMD12 as an invalid command because the card has successfully sent the tuning data, and it cannot send a response.
   - The host controller may generate a response time-out interrupt that must be cleared by software.
5) Repeat steps 2) to 4) by increasing the phase shift value or delay element number on cclk_in_sample until the correct sampling point is received such that the host does not see any of the errors.
6) Mark this phase shift value as the starting point of the sampling window.
7) Repeat steps 2 to 4 by increasing the phase shift value or delay element number on cclk_in_sample until the host sees the errors starting to come again or the phase shift value reaches 360-degree.
8) Mark the last successful phase shift value as the ending point of the sampling window.

A window is established where the tuning block is matched. For example, for a scenario where the tuning block is received correctly for a phase shift window of 90-degree and 180-degree, then an appropriate sampling point is established as 135-degree. Once a sampling point is established, no errors should be visible in the tuning block.

1.6.12 Package Command

In order to reduce overhead, read and write commands can be packed in groups of commands-either all read or all write-that transfer the data for all commands in the group in one transfer on the bus.
Packed commands can be of two types:

- Packed Write: CMD23 → CMD25
- Packed Read: CMD23 → CMD25 → CMD23 → CMD18

Packed commands are put in packets by the application software and are transparent to the core. For more information on packed commands, refer to the eMMC specification.

1.6.13 Card Detection Method

There are many methods for SDMMC/SDIO card detection.

- Method1: Using CDETECT register, which is value on card_detect_n input port. 0 represents presence of card.
- Method2: Using card detection unit, outputting host interrupt (IRQ_ID[46]). The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value in DEBNCE[23:0].Following figure illustrates the timing for card-detect signals.

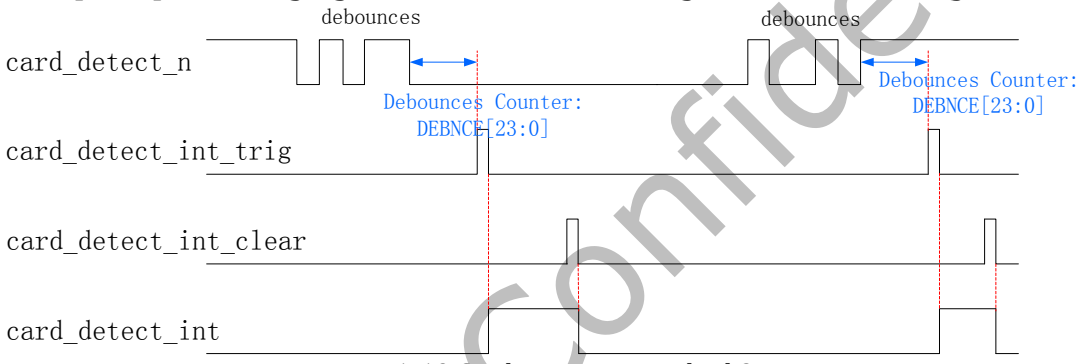


Fig. 1-19 Card Detection Method 2

- Method3: Using card detection unit in GRF, outputting sdmmc_detect_dual edge_int(IRQ_ID[86]), only available for SDMMC. Similar to Method2, except that the debounce is configurable; and the insertion/removal detection interrupt can be enabled or cleared respectively. The detailed register information is:

Table 1-17 Register for SDMMC Card Detection Method 3

Signal Name	Source	Default	Description
sd_detectn_rise_edge_irq_en	GRF_SIG_DETECT_CON[0]	0	sdmmc detect_n signal rise edge interrupt enable. 1: enable 0: disable
sd_detect_fall_edge_detect_en	GRF_SIG_DETECT_CON[1]	0	sd_detect_falling_edge enable 0: disable 1: enable
sd_detect_time[19:0]	{PMUGRF_SOC_CON11[3:0],PMUGRF_SOC_CON10[15:0]}	'h61a8	sd card detection time, in the unit of pmu clock, 24Mhz in normal mode or 375KHz in low power mode
sd_detect_rising_edge_dectect_status	SIG_DETECT_STATUS[0]	0	sd_detect_rising_edge status 0: disable 1: enable
sd_detect_fall_edge_detect_status	SIG_DETECT_STATUS[1]	0	sd_detect_falling_edge status 0: disable 1: enable
sd_detect_rising_edge_dectect_clr	SIG_DETECT_CLR[0]	0	sd_detect_rising_edge clear 0: disable 1: enable

Signal Name	Source	Default	Description
sd_detect_fall_edge_detect_clr	SIG_DETECT_CLR[1]	0	sd_detect_falling_edge clear 0: disable 1: enable

- Method4: Using card_detect_n for interrupt source, connecting to *IRQ* directly.

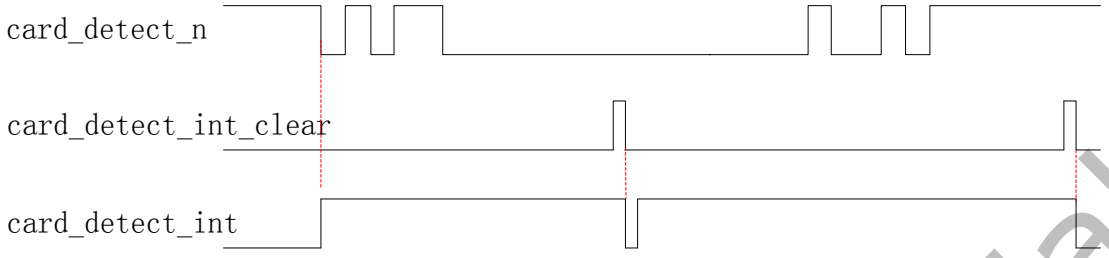


Fig. 1-20 Card Detection Method 4

1.6.14 SDMMC IOMUX With JTAG

The IO for sdmmc_cdata2/sdmmc_cdata3 is shared with jtag_tck/jtag_tms. The condition of usage for SDMMC or JTAG usage is as follows.

- If GRF_SOC_CON7[12](grf_force_jtag) is equal to 1 and sdmmc card is not detected within detection time(in the unit of XIN24M clock), the GPIOs are used for JTAG.
- Otherwise, the GPIOs' usage is defined by IOMUX configuration.

Chapter 2 USB2.0 PHY

2

2.1 Overview

USB2.0 PHY performs low level protocol between UTMI interface and differential signals. In transmitting mode, it serializes data, performs bit stuffing when needed followed with NRZI encoding, generates SYNC and EOP fields. Likewise, in receiving mode, it recovers clock from incoming data, strips the SYNC and EOP field, performs NRZI decoding when needed following by bit un-stuffing and then de-serializes the data. USB2.0 PHY is comprised of one Host port and one OTG port. Host Port is for USB2.0 host controller; OTG port is for USB2.0 part of USB3.0 OTG controller, and as a part to construct a fully feature TypeC subsystem.

USB2.0 PHY supports the following features:

- Fully compliant with USB specification Rev 2.0
- Support 480Mbps/12Mbps/1.5Mbps serial data transmission
- Support Loopback BIST Mode
- Support all test modes defined in USB2.0 Specification
- Host Port support serial mode for 12Mbps/1.5Mbps
- OTG Port support dual-role device, fully support Battery Charge 1.2 Specification

2.2 Block Diagram

USB2.0 PHY comprises with:

- Host Port: used for USB2.0 host controller (see Chapter USB2.0 Host Controller)
- OTG Port: used for USB3.0 OTG controller (see Chapter USB3.0 OTG Controller) with TypeC PHY to comprise as fully feature TypeC

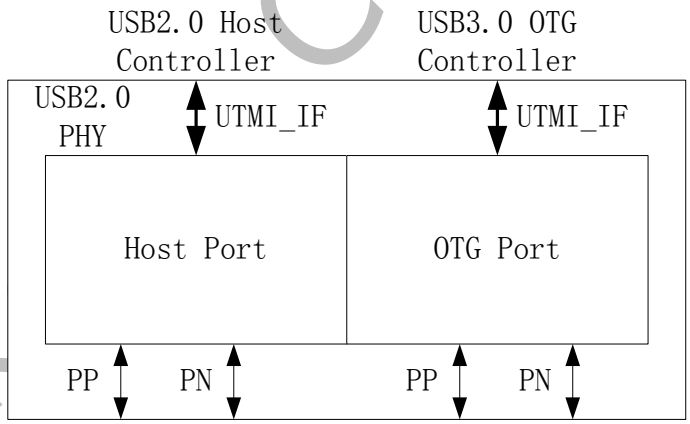


Fig. 2-1 USB2.0 PHY Block Diagram

2.3 Function Description

2.4 Register Description

Please refer to GRF register description.

2.5 Interface Description

Table 2-1 USB2.0 PHY Interface Description

Module Pin	Direction	Pad Name	Descriptions
USB0ID	I	IO_USB0_ID	USB2.0 PHY0 OTG Port ID, left unused for TypeC
USB0PN	I/O	IO_USB0_PN	USB2.0 PHY0 OTG Port PN
USB0PP	I/O	IO_USB0_PP	USB2.0 PHY0 OTG Port PP

Module Pin	Direction	Pad Name	Descriptions
VBUS	I	IO_USB0_VBUS	USB2.0 PHY0 OTG Port VBUS
USB1PN	I/O	IO_USB1_PN	USB2.0 PHY0 Host Port PN
USB1PP	I/O	IO_USB1_PP	USB2.0 PHY0 Host Port PP
USBRBIAS	I/O	IO_USB0_RBIAS	USB2.0 PHY0 Shared RBIAS
USB0ID	I	IO_USB2_ID	USB2.0 PHY1 OTG Port ID, left unused for TypeC
USB0PN	I/O	IO_USB2_PN	USB2.0 PHY1 OTG Port PN
USB0PP	I/O	IO_USB2_PP	USB2.0 PHY1 OTG Port PP
VBUS	I	IO_USB2_VBUS	USB2.0 PHY1 OTG Port VBUS
USB1PN	I/O	IO_USB3_PN	USB2.0 PHY1 Host Port PN
USB1PP	I/O	IO_USB3_PP	USB2.0 PHY1 Host Port PP
USBRBIAS	I/O	IO_USB2_RBIAS	USB2.0 PHY1 Shared RBIAS

Chapter 3 USB2.0 Host Controller

3

3.1 Overview

USB2.0 host controller supports fully USB2.0 functions with one EHCI host controller and one OHCI host controller, and each host controller has one USB port. OHCI host controller only supports full-speed and low-speed mode and is used for full-speed devices and low-speed devices. EHCI only supports high-speed mode and is used for high-speed devices. OHCI host controller and EHCI host controller shares the same USB port, EHCI host controller will auto select the owner (OHCI or EHCI) of this USB port depending on the speed mode of attached devices, when selecting OHCI as owner, OHCI host controller will serve for the attached device; when selecting EHCI as owner, EHCI host controller will serve for the attached device.

USB2.0 Host Controller supports the following features:

- Compatible Specifications
  - Universal Serial Bus Specification, Revision 2.0
  - Enhanced Host Controller Interface Specification (EHCI), Revision 1.0
  - Open Host Controller Interface Specification (OHCI), Revision 1.0a
- Support High-speed (480Mbps), Full-speed (12Mbps) and Low-speed (1.5Mbps)

3.2 Block Diagram

USB2.0 Host Controller comprises with:

- EHCI Host Controller: Perform High-speed transactions
- OHCI Host Controller: Perform full/low-speed transactions
- Port Routing Control: Select EHCI Host Controller or OHCI Host Controller

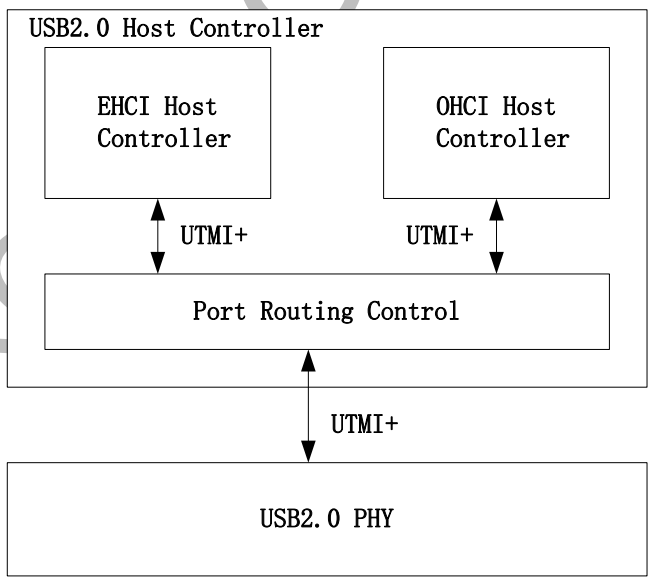


Fig. 3-1 USB2.0 Host Controller Block Diagram

3.3 Function Description

3.3.1 EHCI Host Controller

It performs descriptors and data read or write from or to system memory and packs or un-pack USB transactions from or to UTMI+ interface defined in EHCI specification for high-speed data transmission.

3.3.2 OHCI Host Controller

It performs descriptors and data read/write from/to system memory and packs or un-pack USB transactions from or to UTMI+ interface defined in OHCI specification for full-speed or low-speed data transmission.

3.3.3 Port Routing Control

As part of logic in the EHCI host controller, it is used to auto-select EHCI or OHCI host controller to serve the attached device depending on the speed of the attached device.

3.4 Register Description

3.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 3-1 USB2.0 Host Controller Address Mapping

Base Address[17]	Device	Address Length	Offset Address Range
1'b0	EHCI	128K BYTE	0x00000 ~ 0x1ffff
1'b1	OHCI	128K BYTE	0x20000 ~ 0x3ffff

EHCI and OHCI register definitions, please refer to Enhanced Host Controller Interface Specification (EHCI), Revision 1.0 and Open Host Controller Interface Specification (OHCI), Revision 1.0a.

3.5 Interface Description

Please see chapter USB2.0 PHY

3.6 Application Notes

3.6.1 Special Setting

Set USB2.0 host controller master secure setting (pls refer to Chapter SGRF) before initialization.

3.6.2 Program flow

Please refer to Enhanced Host Controller Interface Specification (EHCI), Revision 1.0 and Open Host Controller Interface Specification (OHCI), Revision 1.0a.

Chapter 4 USB3.0 OTG Controller

4.1 Overview

USB3.0 OTG Controller can act as static host, static device, USB2.0/3.0 OTG A device or B device basing on the status of input ID from USB2.0 PHY and DFP/UFP/Data Role Swap defined in USB TypeC specification. It can perform data transmission between host and device as host or device for Super-Speed/High-Speed/Full-Speed/Low-Speed.

USB3.0 OTG controller supports the following features:

- General Features
  1. Compatible Specification
     - Universal Serial Bus 3.0 Specification, Revision 1.0
     - Universal Serial Bus Specification, Revision 2.0
     - eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1
  2. Support Control/Bulk(including stream)/Interrupt/Isochronous Transfer
  3. Simultaneous IN and OUT transfer for USB3.0, up to 8Gbps bandwidth
  4. Descriptor caching and data pre-fetching used to improve system performance in high-latency systems
  5. LPM protocol in USB 2.0 and U0, U1, U2, and U3 states for USB 3.0
  6. Dynamic FIFO memory allocation for endpoints
  7. Keep-Alive feature in LS mode and (micro-)SOFs in HS/FS modes
  8. Low MIPS requirement
     - Driver involved only in setting up transfers and high-level error recovery
     - Hardware handles data packing and routing to a specific pipe
- Application Interface Features
  1. AHB Slave interface
  2. AXI Master interface
     - Programmable burst lengths up to 16
     - Handle fixed burst address alignment
     - Programmable number of outstanding read/write requests up to 16
     - Concurrent read/write to get best performance of USB3.0 duplex operation
- USB3.0 Device Features
  1. Up to 7 IN endpoints, including control endpoint 0
  2. Up to 6 OUT endpoints, including control endpoint 0
  3. Up to 13 endpoint transfer resources, each one for each endpoint
  4. Flexible endpoint configuration for multiple applications/USB set-configuration modes
  5. Hardware handles ERDY and burst
  6. Stream-based bulk endpoints with controller automatically initiating data movement
  7. Isochronous endpoints with isochronous data in data buffers
  8. Flexible Descriptor with rich set of features to support buffer interrupt moderation, multiple transfers, isochronous, control, and scattered buffering support
- USB Class-Specific Device Features
  1. Stream support for UASP application
  2. Gathering of scattered packet to support Ethernet Over USB
  3. Scheduling of multiple Ethernet packets without interrupt
  4. Variable FIFO buffer allocation for each endpoint
  5. For isochronous applications, scheduling of variable-length payloads for each microframe
  6. Microframe precise scheduling for isochronous applications
  7. Configurable endpoint type selection and dynamic FIFO allocation to facilitate multi-function/composite device implementation. During set-config or alternate-setting, device resources are reconfigured to meet the configuration or alternate setting requirements.

- USB 3.0 xHCI Host Features
  1. Support up to 64 devices
  2. Support 1 interrupter
  3. Support 1 USB2.0 port and 1 Super-Speed port
  4. Support xHCI Debug Capability
  5. Concurrent USB3.0/USB2.0 traffic, up to 8.48Gbps bandwidth
  6. Support standard or open-source xHCI and class driver
- USB 3.0 Dual-Role Device (DRD) Features
  1. Static Device Operation
  2. Static Host Operation
  3. USB3.0/USB2.0 OTG A device and B device basing on ID
  4. UFP/DFP and Data Role Swap Defined in USB TypeC Specification
  5. Not support USB3.0/USB2.0 OTG session request protocol(SRP), host negotiation protocol(HNP) and Role Swap Protocol(RSP)

4.2 Block Diagram

USB3.0 OTG Controller comprises with:

- Bus Interface/List Management: Register Interface/Data and Descriptors DMA management
- HS/FS/LS MAC : USB2.0 part logic
- SS MAC : SS part logic
- USB2.0 PHY: UTMI+ interface USB2.0 PHY, see Chapter USB2.0 PHY
- TypeC PHY: Pipe Interface Super-Speed PHY, see Chapter TypeC PHY

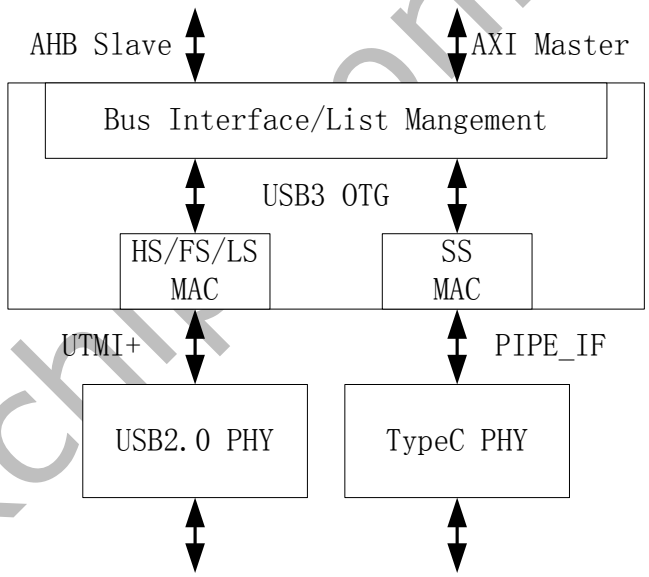


Fig. 4-1 USB3.0 OTG Block Diagram

4.3 Function Description

As a USB3.0 OTG controller, it can act as static xHCI host controller, static device controller, USB3.0/2.0 OTG A device or B device basing on ID of USB2.0 PHY or UFP/DFP defined in TypeC specification.

As device controller/UFP, it can work on either USB2.0 speed or Super-Speed basing on speed of host attached to, and process USB tractions described in the descriptors (read back from external memory by AXI master) to/from UTMI+ interface of USB2.0 PHY or pipe interface of TypeC PHY.

As host controller/DFP, it can work on USB2.0 speed, Super-Speed or both basing on speed or type of attached device, and process USB tractions described in the descriptors (read back from external memory by AXI master) to/from UTMI+ interface of USB2.0 PHY and pipe interface of TypeC PHY.

4.4 Register Description

4.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 4-1 USB3 Address Mapping

Offset Address Range	Register Type
0x00000 ~ 0x07FFF	xHCI Registers, see xHCI spec.
0x0C100 ~ 0x0C6FF	Global Registers
0x0C700 ~ 0x0CBFF	Device Controller Registers
0x0CC00 ~ 0x0CFFF	Unused/Reserved
0x40000 ~ 0x7FFFF	Internal RAM0 – Debug Access (256KB)
0x80000 ~ 0xBFFFF	Internal RAM1 – Debug Access (256KB)
0xC0000 ~ 0xFFFFF	Internal RAM2 – Debug Access (256KB)

4.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
USB3_GSBUSCFG0	0xc100	W	0x00000001	Global SoC Bus Configuration Register 0
USB3_GSBUSCFG1	0xc104	W	0x00000300	Global SoC Bus Configuration Register 1
USB3_GTXTHRCFG	0xc108	W	0x00000000	Global Tx Threshold Control Register
USB3_GRXTHRCFG	0xc10c	W	0x00000000	Global Rx Threshold Control Register
USB3_GCTL	0xc110	W	0x30c12004	Global Core Control Register
USB3_GPMSTS	0xc114	W	0x00000000	Global Power Management Status Register
USB3_GSTS	0xc118	W	0x7e800000	Global Status Register
USB3_GUCTL1	0xc11c	W	0x0004018a	Global User Control Register 1
USB3_GSNPSID	0xc120	W	0x5533290a	Global SNPS ID Register
USB3_GGPIO	0xc124	W	0x00000000	Global General Purpose Input/Output Register
USB3_GUID	0xc128	W	0x12345678	Global User ID Register
USB3_GUCTL	0xc12c	W	0x02008010	Global User Control Register
USB3_GBUSERRADDRLO	0xc130	W	0x00000000	Global SoC Bus Error Address Register - Low
USB3_GBUSERRADDRHI	0xc134	W	0x00000000	Global SoC Bus Error Address Register - High
USB3_GPRTBIMAPLO	0xc138	W	0x00000000	Global SS Port to Bus Instance Mapping Register - Low
USB3_GHWPARAMS0	0xc140	W	0x2020400a	Global Hardware Parameters Register 0
USB3_GHWPARAMS1	0xc144	W	0x0160c93b	Global Hardware Parameters Register 1
USB3_GHWPARAMS2	0xc148	W	0x12345678	Global Hardware Parameters Register 2

Name	Offset	Size	Reset Value	Description
USB3_GHWPARAMS3	0xc14c	W	0x069cd085	Global Hardware Parameters Register 3
USB3_GHWPARAMS4	0xc150	W	0x47822008	Global Hardware Parameters Register 4
USB3_GHWPARAMS5	0xc154	W	0x04202088	Global Hardware Parameters Register 5
USB3_GHWPARAMS6	0xc158	W	0x077c8020	Global Hardware Parameters Register 6
USB3_GHWPARAMS7	0xc15c	W	0x03080756	Global Hardware Parameters Register 7
USB3_GDBGFIFOSPACE	0xc160	W	0x00420000	Global Debug Queue/FIFO Space Available Register
USB3_GDBGLTSSM	0xc164	W	0x41010440	Global Debug LTSSM Register
USB3_GDBGLNMCC	0xc168	W	0x00000000	Global Debug LNMCC Register
USB3_GDBGBMU	0xc16c	W	0x00000000	Global Debug BMU Register
USB3_GDBGLSPMUX	0xc170	W	0x003f0000	Global Debug LSP MUX Register - Device
USB3_GDBGLSP	0xc174	W	0x00000000	Global Debug LSP Register
USB3_GDBGEPINFO0	0xc178	W	0x00000000	Global Debug Endpoint Information Register 0
USB3_GDBGEPINFO1	0xc17c	W	0x00800000	Global Debug Endpoint Information Register 1
USB3_GPRTBIMAP_HSLO	0xc180	W	0x00000000	Global High-Speed Port to Bus Instance Mapping Register - Low
USB3_GPRTBIMAP_FSLO	0xc188	W	0x00000000	Global Full-Speed Port to Bus Instance Mapping Register - Low
USB3_GUSB2PHYCFG0	0xc200	W	0x40102400	Global USB2 PHY Configuration Register 0
USB3_GUSB3PIPECTL0	0xc2c0	W	0x010c0002	Global USB3 PIPE Control Register 0
USB3_GTXFIFOSIZ0~6	0xc300~ 0xc318	W	0x00000042	Global Transmit FIFO Size Register n
USB3_GRXFIFOSIZ0~2	0xc380~ 0xC388	W	0x00000285	Global Receive FIFO Size Register n
USB3_GEVNTADRLO0	0xc400	W	0x00000000	Global Event Buffer Address (Low) Register 0
USB3_GEVNTADRHI0	0xc404	W	0x00000000	Global Event Buffer Address (High) Register 0
USB3_GEVNTSIZ0	0xc408	W	0x00000000	Global Event Buffer Size Register 0
USB3_GEVNTCOUNT0	0xc40c	W	0x00000000	Global Event Buffer Count Register 0
USB3_GHWPARAMS8	0xc600	W	0x0000077c	Global Hardware Parameters Register 8

Name	Offset	Size	Reset Value	Description
USB3_GTXFIFOPRIDEV	0xc610	W	0x00000000	Global Device TX FIFO DMA Priority Register
USB3_GTXFIFOPRIHST	0xc618	W	0x00000000	Global Host TX FIFO DMA Priority Register
USB3_GRXFIFOPRIHST	0xc61c	W	0x00000000	Global Host RX FIFO DMA Priority Register
USB3_GFIFOPRIDBC	0xc620	W	0x00000000	Global Host Debug Capability DMA Priority Register
USB3_GDMAHLRATIO	0xc624	W	0x00000008	Global Host FIFO DMA High-Low Priority Ratio Register
USB3_GFLADJ	0xc630	W	0x00000000	Global Frame Length Adjustment Register
USB3_DCFG	0xc700	W	0x00080004	Device Configuration Register
USB3_DCTL	0xc704	W	0x00f00000	Device Control Register
USB3_DEVTEN	0xc708	W	0x00000000	Device Event Enable Register
USB3_DSTS	0xc70c	W	0x00500004	Device Status Register
USB3_DGCMDPAR	0xc710	W	0x00000000	Device Generic Command Parameter Register
USB3_DGCMD	0xc714	W	0x00000000	Device Generic Command Register
USB3_DALEPENA	0xc720	W	0x00000000	Device Active USB Endpoint Enable Register
USB3_DEPnCMDPAR2	0xc800~ 0xc8c0	W	0x00000000	Device Physical Endpoint-n Command Parameter 2 Register
USB3_DEPnCMDPAR1	0xc804~ 0xc8c4	W	0x00000000	Device Physical Endpoint-n Command Parameter 1 Register
USB3_DEPnCMDPAR0	0xc808~ 0xc8c8	W	0x00000000	Device Physical Endpoint-n Command Parameter 0 Register
USB3_DEPnCMD	0xc80c~ 0xc8cc	W	0x00000000	Device Physical Endpoint-n Command Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

4.4.3 Detail Register Description

USB3_GSBUSCFG0

Address: Operational Base + offset (0xc100)

Global SoC Bus Configuration Register 0

Bit	Attr	Reset Value	Description
31:28	RW	0x0	DATRDREQINFO DATRDREQINFO AXI-cache for Data Read (DatRdReqInfo)
27:24	RW	0x0	DESRDREQINFO DESRDREQINFO AXI-cache for Descriptor Read (DesRdReqInfo).

Bit	Attr	Reset Value	Description
23:20	RW	0x0	DATWRREQINFO DATWRREQINFO AXI-cache for Data Write (DatWrReqInfo).
19:16	RW	0x0	DESWRREQINFO DESWRREQINFO AXI-cache for Descriptor Write (DesWrReqInfo)
15:12	RO	0x0	reserved
11	RW	0x0	DATBIGEND Data Access is Big Endian This bit controls the endian mode for data accesses.0, Little-endian (default); 1, Big-endian;
10	RW	0x0	DESBIGEND Descriptor Access is Big Endian This bit controls the endian mode for descriptor accesses. 0, Little-endian (default); 1, Big-endian.
9:8	RO	0x0	reserved
7	RW	0x0	INCR256BRSTENA INCR256 Burst Type Enable If software set this bit to 1, the AXI master uses INCR to do the 256-beat burst.
6	RW	0x0	INCR128BRSTENA INCR128 Burst Type Enable If software set this bit to 1, the AXI master uses INCR to do the 128-beat burst.
5	RW	0x0	INCR64BRSTENA INCR64 Burst Type Enable If software set this bit to 1, AXI master uses INCR to do the 64-beat burst.
4	RW	0x0	INCR32BRSTENA INCR32 Burst Type Enable If software set this bit to 1, the AXI master uses INCR to do the 32-beat burst.
3	RW	0x0	INCR16BRSTENA INCR16 Burst Type Enable If software set this bit to 1, the AXI master uses INCR to do the 16-beat burst.
2	RW	0x0	INCR8BRSTENA INCR8 Burst Type Enable If software set this bit to 1, the AXI master uses INCR to do the 8-beat burst.

Bit	Attr	Reset Value	Description
1	RW	0x0	INCR4BRSTENA INCR4 Burst Type Enable When this bit is enabled the controller is allowed to do bursts of beat length 1, 2, 3, and 4. It is highly recommended that this bit is enabled to prevent descriptor reads and writes from being broken up into separate transfers.
0	RW	0x1	INCRBRSTENA Undefined Length INCR Burst Type Enable This bit determines the set of burst lengths the master interface uses. It works in conjunction with the GSBUSCFG0[7:1] enables (INCR256/128/64/32/16/8/4). 0: INCRX burst mode ARLEN/AWLEN do not use INCR. They use only the following burst lengths: 1; 4 (if GSBUSCFG0.INCR4BrstEna = 1); 8 (if GSBUSCFG0.INCR8BrstEna = 1); 16 (if GSBUSCFG0.INCR16BrstEna = 1); 32 (if GSBUSCFG0.INCR32BrstEna = 1); 64 (if GSBUSCFG0.INCR64BrstEna = 1); 128 (if GSBUSCFG0.INCR128BrstEna = 1); 256 (if GSBUSCFG0.INCR256BrstEna = 1); 1: INCR (undefined length) burst mode ARLEN/AWLEN uses any length less than or equal to the largest-enabled burst length of INCR4/8/16/32/64/128/256. For cache line-aligned applications, this bit is typically set to 0 to ensure that the master interface uses only power-of-2 burst lengths (as enabled via GSBUSCFG0[7:0]).

USB3_GSBUSCFG1

Address: Operational Base + offset (0xc104)

Global SoC Bus Configuration Register 1

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	EN1KPAGE 1K Page Boundary Enable By default (this bit is disabled) the AXI breaks transfers at the 4k page boundary. When this bit is enabled, the AXI master (DMA data) breaks transfers at the 1k page boundary.

Bit	Attr	Reset Value	Description
11:8	RW	0x3	PipeTransLimit AXI Pipelined Transfers Burst Request Limit The field controls the number of outstanding pipelined transfer requests the AXI master pushes to the AXI slave. When the AXI master reaches this limit, it does not make any more requests on the AXI ARADDR and AWADDR buses until the associated data phases complete. This field is encoded as follows: 0: 1 request 1: 2 requests 2: 3 requests 3: 4 requests ... F: 16 requests
7:0	RO	0x0	reserved

USB3_GTXTHRCFG
Address: Operational Base + offset (0xc108)
Global Tx Threshold Control Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	USBTxPktCntSel USB Transmit Packet Count Enable This field enables/disables the USB transmission multi-packet thresholding: 0: USB transmission multi-packet thresholding is disabled; the core can only start transmission on the USB after the entire packet has been fetched into the corresponding TXFIFO. 1: USB transmission multi-packet thresholding is enabled. The core can only start transmission on the USB after USB Transmit Packet Count amount of packets for the USB transaction (burst) are already in the corresponding TXFIFO. This mode is only valid in the host mode. It is only used for SuperSpeed.
28	RO	0x0	reserved
27:24	RW	0x0	USBTxPktCnt USB Transmit Packet Count This field specifies the number of packets that must be in the TXFIFO before the core can start transmission for the corresponding USB transaction (burst). This field is only valid when the USB Transmit Packet Count Enable field is set to one. Valid values are from 1 to 15. Note: This field must be less than or equal to the USB Maximum TX Burst Size field.

Bit	Attr	Reset Value	Description
23:16	RW	0x00	USBMaxTxBurstSize USB Maximum TX Burst Size When USBTxPktCntSel is 1, this field specifies the Maximum Bulk OUT burst the core can execute. When the system bus is slower than the USB, TX FIFO can underrun during a long burst. You can program a smaller value to this field to limit the TX burst size that the core can execute. It only applies to SS Bulk, Isochronous, and Interrupt OUT endpoints in the host mode. Valid values are from 1 to 16.
15:0	RO	0x0	reserved

USB3_GRXTHRCFG
Address: Operational Base + offset (0xc10c)
Global Rx Threshold Control Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	USBRxPktCntSel USB ReceivePacket Count Enable This field enables/disables the USB reception multi-packet thresholding: 0: The core can only start reception on the USB when the RX FIFO has space for at least one packet. 1: The core can only start reception on the USB when the RX FIFO has space for at least USBRxPktCnt amount of packets. This mode is valid in both host and device mode. It is only used for SuperSpeed. In device mode, Setting this bit to 1 also enables the functionality of reporting NUMP in the ACK TP based on the RX FIFO space instead of reporting a fixed NUMP derived from DCFG.NUMP If you are using external buffer control (EBC) feature, disable this mode by setting USBRxPktCntSel to 0.
28	RO	0x0	reserved
27:24	RW	0x0	USBRxPktCnt USB Receive Packet Count In host mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the core can start the corresponding USB RX transaction (burst). In device mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the core can send ERDY for a flow-controlled endpoint. This field is valid only when the USB Receive Packet Count Enable field is set to 1. The valid values for this field are from 1 to 15. Note: This field must be less than or equal to the USB Maximum Receive Burst Size field.

Bit	Attr	Reset Value	Description
23:19	RW	0x00	USBMaxRxBurstSize USB Maximum Receive Burst Size In host mode, this field specifies the Maximum Bulk IN burst the usb3 controller can perform. When the system bus is slower than the USB, RX FIFO can overrun during a long burst. You can program a smaller value to this field to limit the RX burst size that the core can perform. It only applies to SS Bulk, Isochronous, and Interrupt IN endpoints in the host mode. In device mode, this field specifies the NUMP value that is sent in ERDY for an OUT endpoint. This field is valid only when USBRxPktCntSel is one. The valid values for this field are from 1 to 16.
18:0	RO	0x0	reserved

USB3_GCTL
Address: Operational Base + offset (0xc110)
Global Core Control Register

Bit	Attr	Reset Value	Description
31:19	RW	0x0618	PWRDNSCALE Power Down Scale (PwrDnScale) The USB3 suspend_clk input replaces pipe3_rx_pclk as a clock source to a small part of the USB3 core that operates when the SS PHY is in its lowest power (P3) state, and therefore does not provide a clock. The Power Down Scale field specifies how many suspend_clk periods fit into a 16 kHz clock period. When performing the division, round up the remainder. For example, when using an 8-bit/16-bit/32-bit PHY and 25-MHz Suspend clock, Power Down Scale = 25000 kHz/16 kHz = 13'd1563 (rounder up) Note: Minimum Suspend clock frequency is 32 kHz Maximum Suspend clock frequency is 125 MHz The LTSSM uses Suspend clock for 12-ms and 100-ms timers during suspend mode. According to the USB 3.0 specification, the accuracy on these timers is 0% to +50%. 12 ms + 0~+50% accuracy = 18 ms (Range is 12 ms - 18 ms) 100 ms + 0~+50% accuracy = 150 ms (Range is 100 ms - 150ms). The suspend clock accuracy requirement is: (12,000/62.5) * (GCTL[31:19]) * actual suspend_clk_period must be between 12,000 and 18,000 (100,0000/62.5) * (GCTL[31:19]) * actual suspend_clk_period must be between 100,000 and 150,000 For example, if your suspend_clk frequency varies from 7.5 MHz to 10.5MHz, then the value needs to programmed is: Power Down Scale = 10500/16 = 657 (rounded up; and fastest frequency used).
18	RW	0x0	MASTERFILTBYPASS Master Filter Bypass When this bit is set to 1'b1, all the filters are bypassed. The double synchronizers to mac_clk preceding the filters are also bypassed. For enabling the filters, this bit must be 1'b0.

Bit	Attr	Reset Value	Description
17	RW	0x0	BYPSSETADDR Bypass SetAddress in Device Mode. When BYPSSETADDR bit is set, the device core uses the value in the DCFG[DevAddr] bits directly for comparing the device address in the tokens. For simulation, you can use this feature to avoid sending an actual SET ADDRESS control transfer on the USB, and make the device core respond to a new address. When the xHCI Debug capability is enabled and this bit is set, the Debug Target immediately enters the configured state without requiring the Debug Host to send a SetAddress or SetConfig request. Note: You can set this bit for simulation purposes only. In the actual hardware, this bit must be set to 1'b0.
16	RW	0x1	U2RSTECN U2RSTECN If the SuperSpeed connection fails during POLL or LMP exchange, the device connects at non-SS mode. If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode. For each attempt, the device checks receiver termination eight times. From 2.60a release, this bit controls whether to check for Rx.Detect eight times or one time for every attempt. Device controller on USB 2.0 reset checks for receiver termination eight times per attempt if this bit is set to zero, or only once per attempt if the bit is set to one. Note: This bit is applicable only in device mode.

Bit	Attr	Reset Value	Description
15:14	RW	0x0	FRMSCLDWN FRMSCLDWN This field scales down device view of a SOF/USOF/ITP duration. For SS/HS mode: Value of 2'h3 implements interval to be 15.625 us Value of 2'h2 implements interval to be 31.25 us Value of 2'h1 implements interval to be 62.5 us Value of 2'h0 implements interval to be 125us For FS mode, the scale-down value is multiplied by 8. When xHCI Debug Capability is enabled, this field also scales down the MaxPacketSize of the IN and OUT bulk endpoint to allow more traffic during simulation. It can only be changed from a non-zero value during simulation. 2'h0: 1024 bytes 2'h1: 512 bytes 2'h2: 256 bytes 2'h3: 128 bytes
13:12	RW	0x2	PRTCAPDIR PRTCAPDIR: Port Capability Direction (PrtCapDir) 2'b01: for Host configurations 2'b10: for Device configurations SW should base on IDDIG input to set usb3 controller as an OTG 2.0/3.0 device with A-device or B-device.
11	RW	0x0	CORESOFTRESET Core Soft Reset (CoreSoftReset) 1'b0 - No soft reset; 1'b1 - Soft reset to core Clears the interrupts and all the CSRs except the following registers: GCTL; GUCTL; GSTS; GSNPSID; GGPIO; GUID; GUSB2PHYCFGn registers; GUSB3PIPECTLn registers; DCFG; DCTL; DEVTEN; DSTS. When you reset PHYs (using GUBS3PHYCFG or GUSB3PIPECTL registers), you must keep the core in reset state until PHY clocks are stable. This controls the bus, ram, and mac domain resets. Note: This bit is for debug purposes only. Use USBCMD.HCRESET in xHCI Mode and DCTL.SoftReset in device mode for soft reset.

Bit	Attr	Reset Value	Description
10	RW	0x0	SOFITPSYNC SOFITPSYNC If this bit is set to 0 operating in host mode, the core keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever there is a SuperSpeed port that is not in Rx.Detect, SS.Disable and U3. If this bit is set to 1 operating in host mode, the core keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever the other non-SuperSpeed ports are not in a suspended state. This feature is useful because it saves power by suspending UTMI/ULPI when SuperSpeed only is active, and it helps resolve when the PHY does not transmit a host resume unless it is placed in suspend state. This bit must be programmed as a part of initialization at power-on reset, and must not be dynamically changed afterwards. Note: USB2PHYCFGn[6].PhySusp eventually decides to put the UTMI/ULPI PHY in to suspend state. In addition, when this bit is set to 1, the core generates ITP from the ref_clk based counter. Otherwise, ITP and SOF are generated from utmi/ulpi_clk[0] based counter. To program the reference clock period inside the core, refer to GUCTL[31:22].REFCLKPER. This feature is valid in Host and DRD/OTG configurations and used only in Host mode operation. If you never use this feature or the GFLADJ.GFLADJ_REFCLK_LPM_SEL, the minimum frequency for the ref_clk can be as low as 32KHz. You can connect the suspend_clk (as low as 32 KHz) to the ref_clk. If you plan to enable hardware-based LPM or software-based LPM (PORTPMSC. HLE=1), then you cannot use this feature. Turn off this feature by setting this bit to 0 and use the GFLADJ.GFLADJ_REFCLK_LPM_SEL feature. If you set this bit to 1, the GUSB2PHYCFG.U2_FREECLK_EXISTS bit and the DWC_USB3_FREECLK_USB2_EXIST parameter must be set to 0. Program this bit to 0 if the core is intended to be operated in USB 3.0 mode.
9	RW	0x0	U1U2TimerScale Disable U1/U2 timer Scaledown (U1U2TimerScale). If set to 1 along with GCTL[5:4] (ScaleDown) = 2'bX1, disables the scale down of U1/U2 inactive timer values. This is for simulation mode only.

Bit	Attr	Reset Value	Description
8	RW	0x0	DEBUGATTACH Debug Attach When this bit is set: 1. SS Link proceeds directly to the Polling link state (after RUN/STOP in the DCTL register is asserted) without checking remote termination; 2. Link LFPS polling timeout is infinite; 3. Polling timeout during TS1 is infinite (in case link is waiting for TXEQ to finish).
7:6	RW	0x0	RAMCLKSEL RAM Clock Select (RAMClkSel) 2'b00: bus clock 2'b01: pipe clock (Only used in device mode) 2'b10: In device mode, pipe/2 clock. In Host mode, controller switches ram_clk between pipe/2 clock, mac2_clk and bus_clk based on the status of the U2/U3 ports 2'b11: In device mode, selects mac2_clk as ram_clk (when 8-bit UTMI or ULPI used. Not supported in 16-bit UTMI mode); In Host mode, controller switches ram_clk between pipe_clk, mac2_clk and bus_clk based on the status of the U2/U3 ports. In device mode, upon a USB reset and USB disconnect, the hardware clears these bits to 2'b00.
5:4	RW	0x0	SCALEDOWN Scale-Down Mode (ScaleDown) When Scale-Down mode is enabled for simulation, the core uses scaled-down timing values, resulting in faster simulations. When Scale-Down mode is disabled, actual timing values are used. This is required for hardware operation. HS/FS/LS Modes: 2'b00: Disables all scale-downs. Actual timing values are used. 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include Speed enumeration, HNP/SRP, and Host mode suspend and resume 2'b10: Enables scale-down of Device mode suspend and resume timing values only. 2'b11: Enables bit 0 and bit 1 scale-down timing values. SS Mode: 2'b00: Disables all scale-downs. Actual timing values are used. 2'b01: Enables scaled down SS timing and repeat values including: (1) Number of TxEq training sequences reduce to 8; (2) LFPS polling burst time reduce to 256 nS; (3) LFPS warm reset receive reduce to 30 uS. 2'b10: No TxEq training sequences are sent. Overrides Bit 4. 2'b11: Enables bit 0 and bit 1 scale-down timing values.

Bit	Attr	Reset Value	Description
3	RW	0x0	DISSCRAMBLE Disable Scrambling (DisScramble) Transmit request to Link Partner on next transition to Recovery or Polling.
2	RW	0x1	U2EXIT_LFPS U2EXIT_LFPS If this bit is: 0: the link treats 248ns LFPS as a valid U2 exit. 1: the link waits for 8us of LFPS before it detects a valid U2 exit. This bit is added to improve interoperability with a third party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller.
1	RO	0x0	GblHibernationEn GblHibernationEn This bit enables hibernation at the global level. If hibernation is not enabled through this bit, the PMU immediately accepts the D0->D3 and D3->D0 power state change requests, but does not save or restore any core state. In addition, the PMUs never drive the PHY interfaces and let the core continue to drive the PHY interfaces.
0	RW	0x0	DSBLCLKGTNG Disable Clock Gating (DsblClkGtng) This bit is set to 1 and the core is in Low Power mode, internal clock gating is disabled. You can set this bit to 1'b1 after Power On Reset.

USB3_GPMSTS
Address: Operational Base + offset (0xc114)
Global Power Management Status Register

Bit	Attr	Reset Value	Description
31:28	WO	0x0	PortSel Global Power Management Status Register This field selects the port number.
27:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:12	RO	0x00	U3Wakeup U3Wakeup This field gives the following USB 3.0 port wakeup conditions: Bit [12]: Overcurrent Detected Bit [13]: Resume Detected Bit [14]: Connect Detected Bit [15]: Disconnect Detected Bit [16]: Last Connection State
11:10	RO	0x0	reserved
9:0	RO	0x000	U2Wakeup U2Wakeup This field indicates the following USB 2.0 port wakeup conditions: Bit [0]: Overcurrent Detected Bit [1]: Resume Detected Bit [2]: Connect Detected Bit [3]: Disconnect Detected Bit [4]: Last Connection State Bit [5]: ID Change Detected Bit [6]: SRP Request Detected Bit [7]: ULPI Interrupt Detected Bit [8]: USB Reset Detected Bit [9]: Resume Detected Changed

USB3_GSTS
Address: Operational Base + offset (0xc118)
Global Status Register

Bit	Attr	Reset Value	Description
31:20	RO	0x7e8	CBELT Current BELT Value In Host mode, this field indicates the minimum value of all received device BELT values and the BELT value that is set by the Set Latency Tolerance Value command.
19:12	RO	0x0	reserved
11	RO	0x0	SSIC_IP SSIC interrupt pending This field indicates that there is a pending interrupt related to SSIC in the SEVT register. Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is reserved.
10	RO	0x0	OTG_IP OTG Interrupt Pending This field indicates that there is a pending interrupt pertaining to OTG in OEVT register.

Bit	Attr	Reset Value	Description
9	RO	0x0	BC_IP Battery Charger Interrupt Pending This field indicates that there is a pending interrupt pertaining to BC in BCEVT register.
8	RO	0x0	ADP_IP ADP Interrupt Pending his field indicates that there is a pending interrupt pertaining to ADP in ADPEVT register.
7	RO	0x0	Host_IP Host Interrupt Pending This field indicates that there is a pending interrupt pertaining to xHC in the Host event queue.
6	RO	0x0	Device_IP Device Interrupt Pending This field indicates that there is a pending interrupt pertaining to peripheral (device) operation in the Device event queue.
5	W1C	0x0	CSRTimeout CSR Timeout When this bit is 1'b1, it indicates that the software performed a write or read to a core register that could not be completed within DWC_USB3_CSR_ACCESS_TIMEOUT bus clock cycles (default: h1FFFF).
4	W1C	0x0	BUSERRADDRVLD Bus Error Address Valid Indicates that the GBUSERRADDR register is valid and reports the first bus address that encounters a bus error.
3:2	RO	0x0	reserved
1:0	RO	0x0	CURMOD Current Mode of Operation Current Mode of Operation

USB3_GUCTL1

Address: Operational Base + offset (0xc11c)

Global User Control Register 1

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	FILTER_SE0_FSLS_EOP FILTER_SE0_FSLS_EOP 0: Default behaviour, no change in Linestate check for SE0 detection in FS/LS 1: Feature enabled, FS/LS SE0 is filtered for 2 clocks for detecting EOP This bit is applicable for FS/LS operation. If this feature is enabled, then SE0 on the linestate is validated for 2 consecutive utmi/ulpi clock edges for EOP detection. This feature is applicable only in FS in device mode and FS/LS mode of operation in host mode. Device mode: FS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then for device LPM hanshake, the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS. Host mode: FS/LS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS/LS port. Enable this feature if the LineState has SE0 glitches during transmission. This bit is quasi-static, i.e., should not be changed during device operation.
28	RW	0x0	TX_IPGAP_LINECHECK_DIS TX_IPGAP_LINECHECK_DIS 0: Default behaviour, no change in Linestate check 1: Feature enabled, 2.0 MAC disables Linestate check during HS transmit This bit is applicable for HS operation of u2mac. If this feature is enabled, then the 2.0 mac operating in HS ignores the UTMI/ULPI Linestate during the transmit of a token (during token-to-token and token-to-data IPGAP). When enabled, the controller implements a fixed 40-bit TxEndDelay after the packet is given on UTMI and ignores the Linestate during this time. This fetaure is applicable only in HS mode of operation. Device mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then for device LPM handshake, the core will ignore the linestate after TX and wait for a fixed clocks (40 bit times equivalent) after transmiting ACK on utmi. Host mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then the ipgap between (tkn to tkn/data) is added by 40 bit times of TXENDDELAY, and linestate is ignored during this 40 bit times delay. Enable this bit if the LineState will not reflect the expected line state (J) during transmission. This bit is quasi-static, i.e., should not be changed during device operation.

Bit	Attr	Reset Value	Description
27	RW	0x0	DEV_TRB_OUT_SPR_IND DEV_TRB_OUT_SPR_IND 0: Default behaviour, no change in TRB status dword 1: Feature enabled, OUT TRB status indicates Short Packet This bit is applicable for device mode only (and ignored in host mode). If the device application (SW/HW) wants to know if a short packet was received for an OUT in the TRB status itself, then this feature can be enabled, so that a bit is set in the TRB writeback in the buf_size dword. Bit[26] - SPR of the trbstatus, RSVD, SPR,PCM1, bufsize dword will be set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. This bit is quasi-static, i.e., should not be changed during device operation.
26	RW	0x0	DEV_FORCE_20_CLK_FOR_30_CLK DEV_FORCE_20_CLK_FOR_30_CLK 0: Default behaviour, Uses 3.0 clock when operating in 2.0 mode 1: Feature enabled This bit is applicable (and to be set) for device mode (DCFG.Speed!= SS) only. In the 3.0 device core, if the core is programmed to operate in 2.0 only (i.e., Device Speed is programmed to 2.0 speeds in DCFG[Speed]), then setting this bit makes the internal 2.0 (utmi/ulpi) clock to be routed as the 3.0 (pipe) clock. Enabling this feature allows the pipe3 clock to be not-running when forcibily operating in 2.0 device mode. Note: When using this feature, all pipe3 inputs must be in inactive mode, esp. pipe3 clocks not running and pipe3_physatus_async must be tied to 0. This bit should not be set if the core is programmed to operate in SuperSpeed mode (even when it falls back to 2.0). This bit is quasi-static, i.e., should not be changed during operation.
25	RW	0x0	P3_IN_U2 P3_IN_U2 0: Default behaviour, When SuperSpeed link is in U2 , PowerState P2 is attempted on the PIPE Interface. 1: When SuperSpeed link is in U2, PowerState P3 is attempted if GUSB3PIPECTL[17] is set. Setting this bit enables P3 Power State when the SuperSpeed link is in U2. Another Power Saving option. When setting this bit to 1 to enable P3 in P2, GUSB3PIPECTL[27] should be set to 0 to make sure that the U2 exit is attempted in P0.

Bit	Attr	Reset Value	Description
24	RW	0x0	DEV_L1_EXIT_BY_HW DEV_L1_EXIT_BY_HW 0: Default behaviour, disables device L1 hardware exit logic 1: feature enabled This bit is applicable for device mode (2.0) only. This field enables device controller sending remote wakeup for L1 if the device becomes ready for sending/accepting data when in L1 state. If the host expects the device to send remote wkp signalling to resume after going into L1 in flow controlled state, then this bit can be set to send the remote wake signal automatically when the device controller becomes ready. This HW remote wake feature is applicable only to bulk and interrupt transfers, and not for Isoch/Control When control transfers are in progress, the LPM will be rejected (NYET response). Only after control transfers are completed (either with ACK/STALL), LPM will be accepted For Isoch transfers, the host needs to do the wake-up and start the transfer. Device controller will not do remote-wakeup when Isoch endpoints get ready. The device SW needs to keep the GUSB2PHYCFG[EnblSlpM] reset in order to keep the PHY clock to be running for keeping track of SOF intervals. When L1 hibernation is enabled, the controller will not do automatic exit for hibernation requests thru L1. This bit is quasi-static, i.e., should not be changed during device operation.
23:21	RW	0x0	IP_GAP_ADD_ON IP_GAP_ADD_ON This register field is used to add on to the default inter packet gap setting in the USB 2.0 MAC. This should be programmed to a non zero value only in case where you need to increase the default inter packet delay calculations in the USB 2.0 MAC module DWC_usb3_u2mac.v
20	RW	0x0	DEV_LSP_TAIL_LOCK_DIS DEV_LSP_TAIL_LOCK_DIS 0: Default behaviour, enables device lsp lock logic for tail TRB update 1: Fix disabled This is a bug fix for STAR 9000716195 that affects the CSP mode for OUT endpoints in device mode. The issue is that tail TRB index is not synchronized with the cache Scratchpad bytecount update. If the fast-forward request comes in-between the bytecount update ona newly fetched TRB and the tail-index write update in TPF, the RDP works on an incorrect tail index and misses the byte count decrement for the newly fetched TRB in the fast-forwarding process. This fix needs to be present all the times.

Bit	Attr	Reset Value	Description
19	RW	0x0	NAK_PER_ENH_FS NAK_PER_ENH_FS 1: Enables performance enhancement for FS async endpoints in the presence of NAKs 0: Enhancement not applied If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other Full Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your FullSpeed application. Setting this bit will only control, and is only required for Full Speed transfers.
18	RW	0x1	NAK_PER_ENH_HS NAK_PER_ENH_HS 1: Enables performance enhancement for HS async endpoints in the presence of NAKs 0: Enhancement not applied If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other High Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your HighSpeed application. Setting this bit will only control, and is only required for High Speed transfers.
17	RW	0x0	PARKMODE_DISABLE_SS PARKMODE_DISABLE_SS This bit is used only in host mode, and is for debug purpose only. When this bit is set to 1 all SS bus instances in park mode are disabled.

Bit	Attr	Reset Value	Description
16	RW	0x0	PARKMODE_DISABLE_HS PARKMODE_DISABLE_HS This bit is used only in host mode. When this bit is set to 1 all HS bus instances park mode are disabled. To improve performance in park mode, the xHCI scheduler queues in three requests of 4 packets each for High Speed asynchronous endpoints in a micro-frame. But if a device is slow and if it NAKs more than 3 times, then it is rescheduled only in the next micro-frame. This could decrease the performance of a slow device even further. In a few high speed devices (such as Sandisk Cruzer Blade 4GB VID:1921, PID:21863 and Flex Drive VID:3744, PID:8552) when an IN request is sent within 900ns of the ACK of the previous packet, these devices send a NAK. When connected to these devices, if required, the software can disable the park mode if you see performance drop in your system. When park mode is disabled, pipelining of multiple packet is disabled and instead one packet at a time is requested by the scheduler. This allows up to 12 NAKs in a micro-frame and improves performance of these slow devices.
15	RW	0x0	PARKMODE_DISABLE_FSLS PARKMODE_DISABLE_FSLS This bit is used only in host mode, and is for debug purpose only. When this bit is set to 1 all FS/LS bus instances in park mode disabled.
14:9	RO	0x0	reserved
8	RW	0x1	L1_SUSP_THRLD_EN_FOR_HOST L1_SUSP_THRLD_EN_FOR_HOST This bit is used only in host mode. The host controller asserts the utmi_l1_suspend_n and utmi_sleep_n output signals (see LPM Interface Signals table in the Databook) as follows: The controller asserts the utmi_l1_suspend_n signal to put the PHY into deep low-power mode in L1 when both of the following are true: The HIRD/BESL value used is greater than or equal to the value in L1_SUSP_THRLD_FOR_HOST field. The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b1. The controller asserts utmi_sleep_n on L1 when one of the following is true: The HIRD/BESL value used is less than the value in L1_SUSP_THRLD_FOR_HOST field. The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b0.

Bit	Attr	Reset Value	Description
7:4	RW	0x8	L1_SUSP_THRLD_FOR_HOST L1_SUSP_THRLD_FOR_HOST This field is effective only when the L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1. For more details, refer to the description of the L1_SUSP_THRLD_EN_FOR_HOST bit.
3	RW	0x1	HC_ERRATA_ENABLE Host ELD Enable When this bit is set to 1, it enables the Exit Latency Delta (ELD) support defined in the xHCI 1.0 Errata. This bit is used only in the host mode. This bit has to be set to 1 in Host mode.
2	RW	0x0	HC_PARCHK_DISABLE Host Parameter Check Disable When this bit is set to 0 (by default), the xHC checks that the input slot/EP context fields comply to the xHCI Specification. Upon detection of a parameter error during command execution, the xHC generates an event TRB with completion code indicating PARAMETER ERROR. When the bit is set to 1, the xHC does not perform parameter checks and does not generate PARAMETER ERROR completion code.
1	RW	0x1	OVRLD_L1_SUSP_COM OVRLD_L1_SUSP_COM If this bit is set, the utmi_l1_suspend_com_n is overloaded with the utmi_sleep_n signal. This bit is usually set if the PHY stops the port clock during L1 sleep condition.
0	RW	0x0	LOA_FILTER_EN LOA_FILTER_EN If this bit is set, the USB 2.0 port babble is checked at least three consecutive times before the port is disabled. This prevents false triggering of the babble condition when using low quality cables. Note: This bit is valid only in host mode.

USB3_GSNPSID

Address: Operational Base + offset (0xc120)

Global SNPS ID Register

Bit	Attr	Reset Value	Description
31:0	RO	0x5533300a	SNPSID SNPSID SNPSID[31:16] indicates Core Identification Number. 0x5533 is ASCII for U3 (DWC_usb3). SNPSID[15:0] indicates the release number. Current Release is 3.00a. Software uses this register to configure release-specific features in the driver.

USB3_GGPIO
Address: Operational Base + offset (0xc124)
Global General Purpose Input/Output Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	GPO General Purpose Output This field's value is driven out on the gp_out[15:0] core output port.
15:0	RO	0x0000	GPI General Purpose Input This field's read value reflects the gp_in[15:0] core input value.

USB3_GUID
Address: Operational Base + offset (0xc128)
Global User ID Register

Bit	Attr	Reset Value	Description
31:0	RW	0x12345678	USERID USERID Application-programmable ID field.

USB3_GUCTL
Address: Operational Base + offset (0xc12c)
Global User Control Register

Bit	Attr	Reset Value	Description
31:22	RW	0x008	REFCLKPER REFCLKPER This field indicates in terms of nano seconds the period of ref_clk. The default value of this register is set to 'h8 (8ns/125 MHz). This field needs to be updated during power-on initialization, if GCTL.SOFITPSYNC or GFLADJ.GFLADJ_REFCLK_LPM_SEL is set to 1. The programmable maximum value is 62ns, and the minimum value is 8ns. You must use a reference clock with a period that is an integer multiple, so that ITP can meet the jitter margin of 32ns. The allowable ref_clk frequencies whose period is not integer multiples are 16/17/19.2/24/39.7MHz. This field must not be set to 0 at any time. If you never plan to use this feature, then set this field to 'h8, the default value.
21	RW	0x0	NoExtrDl No Extra Delay Between SOF and the First Some HS devices misbehave when the host sends a packet immediately after a SOF. However, adding an extra delay between a SOF and the first packet can reduce the USB data rate and performance. This bit is used to control whether the host must wait for 2 microseconds before it sends the first packet after a SOF, or not. User can set this bit to one to improve the performance if those problematic devices are not a concern in the user's host environment. 1'b0: Host waits for 2 microseconds after a SOF before it sends the first USB packet. 1'b1: Host doesn't wait after a SOF before it sends the first USB packet.
20:18	RO	0x0	reserved
17	RW	0x0	SprsCtrlTransEn Sparse Control Transaction Enable Some devices are slow in responding to Control transfers. Scheduling multiple transactions in one microframe/frame can cause these devices to misbehave. If this bit is set to 1'b1, the host controller schedules transactions for a Control transfer in different microframes/frames.

Bit	Attr	Reset Value	Description
16	RW	0x0	ResBwHSEPS Reserving 85% Bandwidth for HS Periodic EPs By default, HC reserves 80% of the bandwidth for periodic EPs. If this bit is set, the bandwidth is relaxed to 85% to accommodate two high speed, high bandwidth ISOC EPs. USB 2.0 required 80% bandwidth allocated for ISOC traffic. If two High-bandwidth ISOC devices (HD Webcams) are connected, and if each requires 1024-bytes X 3 packets per Micro-Frame, then the bandwidth required is around 82%. If this bit is set, then it is possible to connect two Webcams of 1024bytes X 3 paylod per Micro-Frame each. Otherwise, you may have to reduce the resolution of the Webcams. This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
15	RW	0x1	CMdevAddr Compliance Mode for Device Address When this bit is 1'b1, Slot ID may have different value than Device Address if max_slot_enabled < 128. 1'b1: Increment Device Address on each Address Device command. 1'b0: Device Address is equal to Slot ID. The xHCI compliance requires this bit to be set to 1. The 0 mode is for debug purpose only. This allows you to easily identify a device connected to a port in the Lecroy or Eliisys trace during hardware debug. This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
14	RW	0x0	USBHstInAutoRetryEn Host IN Auto Retry When set, this field enables the Auto Retry feature. For IN transfers (non-isochronous) that encounter data packets with CRC errors or internal overrun scenarios, the auto retry feature causes the Host core to reply to the device with a non-terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP != 0). If the Auto Retry feature is disabled (default), the core will respond with a terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP = 0). 1'b0: Auto Retry Disabled 1'b1: Auto Retry Enabled Note: This bit is also applicable to the device mode.

Bit	Attr	Reset Value	Description
13	RW	0x0	EnOverlapChk Enable Check for LFPS Overlap During Remote Ux 1'b1: The SuperSpeed link when exiting U1/U2/U3 waits for either the remote link LFPS or TS1/TS2 training symbols before it confirms that the LFPS handshake is complete. This is done to handle the case where the LFPS glitch causes the link to start exiting from the low power state. Looking for the LFPS overlap makes sure that the link partner also sees the LFPS. 1'b0: When the link exists U1/U2/U3 because of a remote exit, it does not look for an LFPS overlap.
12	RW	0x0	ExtCapSupptEN External Extended Capability Support Enable When set, this field enables extended capabilities to be implemented outside the core. When the ExtCapSupEN is set and the Debug Capability is enabled, the Next Capability pointer in Debug Capability returns 16. A read to the first DWORD of the last internal extended capability (the "xHCI Supported Protocol Capability for USB 3.0" when the Debug Capability is not enabled) returns a value of 4 in the Next Capability Pointer field. This indicates to software that there is another capability four DWORDs after this capability (for example, at address N+16 where N is the address of this DWORD). If enabled, an external address decoder that snoops the xHC slave interface must be implemented. If it sees an access to N+16 or greater, the slave access is re-routed to a piece of hardware which returns the external capability pointer register of the new capability and also handles reads/writes to this new capability and the side effects. If disabled, a read to the first DWORD of the last internal extended capability returns 0 in the 'Next Capability Pointer' field. This indicates there are no more capabilities.

Bit	Attr	Reset Value	Description
11	RW	0x0	InsrtExtrFSBODI Insert Extra Delay Between FS Bulk OUT Some FS devices are slow to receive Bulk OUT data and can get stuck when there are consecutive Bulk OUT transactions with short inter-transaction delays. This bit is used to control whether the host inserts extra delay between consecutive Bulk OUT transactions to a FS Endpoint. 1'b0: Host doesn't insert extra delay between consecutive Bulk OUT transactions to a FS Endpoint. 1'b1: Host inserts about 12us extra delay between consecutive Bulk OUT transactions to a FS Endpoint to work around the device issue. Note: Setting this bit to one will reduce the Bulk OUT transfer performance for most of the FS devices.
10:9	RW	0x0	DTCT Device Timeout Coarse Tuning This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. The core first checks the DTCT value. If it is 0, then the timeout value is defined by the DTFT. If it is non-zero, then it uses the following timeout values: 2'b00: 0 usec -> use DTFT value instead 2'b01: 500 usec 2'b10: 1.5 msec 2'b11: 6.5 msec
8:0	RW	0x010	DTFT Device Timeout Fine Tuning This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. For the DTFT field to take effect, DTCT must be set to 2'b00. The DTFT value is the number of 125 MHz clocks * 256 to count before considering a device timeout. The minimum value of DTFT is 2. For example, if the mac3_clk is 125 MHz clk (8 ns period), this is calculated as follows: (DTFT value) * 256 * (8 ns) Quick Reference: if DTFT = 0x2, 2*256*8 = 4usec timeout if DTFT = 0x5, 5*256*8 = 10usec timeout if DTFT = 0xA, 10*256*8 = 20usec timeout if DTFT = 0x10, 16*256*8 = 32usec timeout if DTFT = 0x19, 25*256*8 = 51usec timeout if DTFT = 0x31, 49*256*8 = 100usec timeout if DTFT = 0x62, 98*256*8 = 200usec timeout

USB3_GBUSERRADDRLO
Address: Operational Base + offset (0xc130)
Gobal SoC Bus Error Address Register - Low

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	BUSERRADDR Bus Address - Low This register contains the lower 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

USB3_GBUSERRADDRHI
Address: Operational Base + offset (0xc134)
Gobal SoC Bus Error Address Register - High

Bit	Attr	Reset Value	Description
31:0	RU	0x00000000	BUSERRADDR Bus Address - High his register contains the higher 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

USB3_GPRTBIMAPLO
Address: Operational Base + offset (0xc138)
Global SS Port to Bus Instance Mapping Register - Low

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	BINUM1 SS USB Instance Number for Port 1 Application-programmable ID field.

USB3_GHWPARAMS0
Address: Operational Base + offset (0xc140)
Global Hardware Parameters Register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x2020400a	GHWPARAMS0 Global Hardware Parameters Register 0 Global Hardware Parameters Register 0

USB3_GHWPARAMS1
Address: Operational Base + offset (0xc144)
Global Hardware Parameters Register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x0160c93b	GHWPARAMS1 Global Hardware Parameters Register 1 Global Hardware Parameters Register 1

USB3_GHWPARAMS2
Address: Operational Base + offset (0xc148)
Global Hardware Parameters Register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x12345678	GHWPARAMS2 Global Hardware Parameters Register 2 Global Hardware Parameters Register 2

USB3_GHWPARAMS3
Address: Operational Base + offset (0xc14c)
Global Hardware Parameters Register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x069cd085	GHWPARAMS3 Global Hardware Parameters Register 3 Global Hardware Parameters Register 3

USB3_GHWPARAMS4
Address: Operational Base + offset (0xc150)
Global Hardware Parameters Register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x47822008	GHWPARAMS4 Global Hardware Parameters Register 4 Global Hardware Parameters Register 4

USB3_GHWPARAMS5
Address: Operational Base + offset (0xc154)
Global Hardware Parameters Register 5

Bit	Attr	Reset Value	Description
31:0	RO	0x04202088	GHWPARAMS5 Global Hardware Parameters Register 5 Global Hardware Parameters Register 5

USB3_GHWPARAMS6
Address: Operational Base + offset (0xc158)
Global Hardware Parameters Register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x077c8020	GHWPARAMS6 Global Hardware Parameters Register 6 Global Hardware Parameters Register 6

USB3_GHWPARAMS7
Address: Operational Base + offset (0xc15c)
Global Hardware Parameters Register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x03080756	GHWPARAMS7 Global Hardware Parameters Register 7 Global Hardware Parameters Register 7

USB3_GDBGFIFOSPACE
Address: Operational Base + offset (0xc160)
Global Debug Queue/FIFO Space Available Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0042	SPACE_AVAILABLE Space Avalible Space Avalible
15:9	RO	0x0	reserved
8:0	RW	0x000	FIFO_QUEUE_SELECT FIFO/Queue Select (or) Port-Select FIFO/Queue Select[8:5] indicates the FIFO/Queue Type FIFO/Queue Select[4:0] indicates the FIFO/Queue Number Port-Select[3:0] selects the port-number when accessing GDBGLTSSM register.

USB3_GDBGLTSSM
Address: Operational Base + offset (0xc164)
Global Debug LTSSM Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x1	RxElecidle RxElecidle Reflect status of Pipe interface.
29:27	RO	0x0	reserved
26	RW	0x0	LTDBTIMEOUT LTDB Timeout LTDB Timeout
25:22	RO	0x4	LTDBLINKSTATE LTDB Link State LTDB Link State
21:18	RO	0x0	LTDBSUBSTATE LTDB Sub-State LTDB Sub-State
17	RO	0x0	ELASTICBUFFERMODE ELASTICBUFFERMODE Reflect status of Pipe interface.

Bit	Attr	Reset Value	Description
16	RO	0x1	TXELECLDLE TXELECLDLE Reflect status of Pipe interface.
15	RO	0x0	RXPOLARITY RXPOLARITY Reflect status of Pipe interface.
14	RO	0x0	TxDetRxLoopback Tx Detect Rx/Loopback Reflect status of Pipe interface.
13:11	RO	0x0	LTDBPhyCmdState LTSSM PHY command State 000: PHY_IDLE (PHY command state is in IDLE. No PHY request pending) 001: PHY_DET (Request to start Receiver detection) 010: PHY_DET_3 (Wait for Phy_Status (Receiver detection)) 011: PHY_PWR_DLY (Delay Pipe3_PowerDown P0 -> P1/P2/P3 request) 100: PHY_PWR_A (Delay for internal logic) 101: PHY_PWR_B (Wait for Phy_Status(Power state change request))
10:9	RO	0x2	POWERDOWN POWERDOWN Reflect status of Pipe interface.
8	RO	0x0	RXEQTRAIN RXEQTRAIN Reflect status of Pipe interface.
7:6	RO	0x1	TXDEEMPHASIS TXDEEMPHASIS Reflect status of Pipe interface.
5:3	RO	0x0	LTDBClkState LTSSM Clock State In multi-port host configuration, the port number is defined by Port-Select[3:0] field in the GDBGFIFOSPACE register. Note:GDBGLTSSM register is not applicable for USB 2.0-only mode. 000: CLK_NORM (PHY is in non-P3 state and PCLK is running) 001: CLK_TO_P3 (P3 entry request to PHY); 010: CLK_WAIT1 (Wait for Phy_Status (P3 request)); 011: CLK_P3 (PHY is in P3 and PCLK is not running); 100: CLK_TO_P0 (P3 exit request to PHY); 101: CLK_WAIT2 (Wait for Phy_Status (P3 exit request))
2	RO	0x0	TXSWING TXSWING Reflect status of Pipe interface.

Bit	Attr	Reset Value	Description
1	RO	0x0	RXTERMINATION RXTERMINATION Reflect status of Pipe interface.
0	RO	0x0	TXONESZEROS TXONESZEROS Reflect status of Pipe interface.

USB3_GDBGLNMCC
Address: Operational Base + offset (0xc168)
Global Debug LNMCC Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RO	0x000	LNMCC_BERC LNMCC_BERC This field indicates the bit error rate information for the port selected in the GDBGFIFOSPACE.PortSelect field.This field is for debug purposes only.

USB3_GDBGBMU
Address: Operational Base + offset (0xc16c)
Global Debug BMU Register

Bit	Attr	Reset Value	Description
31:8	RW	0x000000	BMU_BCU BMU_BCU Debug information BMU_BCU Debug information
7:4	RO	0x0	BMU_DCU BMU_DCU Debug information BMU_DCU Debug information
3:0	RO	0x0	BMU_CCU BMU_CCU Debug information BMU_CCU Debug information

USB3_GDBGLSPMUX
Address: Operational Base + offset (0xc170)
Global Debug LSP MUX Register - Device

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x3f	logic_analyzer_trace Logic Analyzer Trace Port MUX Select Currently only bits[21:16] are used. A value of 6'h3F drives "0"s on the logic_analyzer_trace signal. If you plan to OR (instead using a mux) this signal with other trace signals in your system to generate a common trace signal, you can use this feature.

Bit	Attr	Reset Value	Description
15	RW	0x0	EnDbc EnDbc Enable debugging of Debug capablity LSP in Host mode. Use HostSelect to select DbC LSP debug information presented in the GDBGLSP register.
14	RO	0x0	reserved
13:8	RW	0x00	HOSTSELECT Host LSP Select Selects the LSP debug information presented in the GDBGLSP register in host mode.
7:4	RW	0x0	DEVSELECT Device LSP Select Selects the LSP debug information presented in the GDBGLSP register in device mode. Or bit[7:4] of HOSTSELECT, Selects the LSP debug information presented in the GDBGLSP register in host mode.
3:0	RW	0x0	EPSELECT Device Endpoint Select Selects the Endpoint debug information presented in the GDBGEPINFO registers in device mode. Or bit[3:0] of HOSTSELECT, Selects the LSP debug information presented in the GDBGLSP register in host mode.

USB3_GDBGLSP
Address: Operational Base + offset (0xc174)
Global Debug LSP Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	LSPDEBUG LSP Debug Information LSP Debug Information

USB3_GDBGEPINFO0
Address: Operational Base + offset (0xc178)
Global Debug Endpoint Information Register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	EPDEBUG Endpoint Debug Information Low 32-bit Endpoint Debug Information Low 32-bit

USB3_GDBGEPINFO1
Address: Operational Base + offset (0xc17c)
Global Debug Endpoint Information Register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00800000	EPDEBUG Endpoint Debug Information High 32-bit Endpoint Debug Information High 32-bit

USB3_GPRTBIMAP_HSLO
Address: Operational Base + offset (0xc180)
Global High-Speed Port to Bus Instance Mapping Register - Low

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	BINUM1 HS USB Instance Number for Port 1 Application-programmable ID field.

USB3_GPRTBIMAP_FSLO
Address: Operational Base + offset (0xc188)
Global Full-Speed Port to Bus Instance Mapping Register - Low

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	BINUM1 FS USB Instance Number for Port 1 Application-programmable ID field.

USB3_GUSB2PHYCFG0
Address: Operational Base + offset (0xc200)
Global USB2 PHY Configuration Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	PHYSOFTRST UTMI PHY Soft Reset Causes the usb2phy_reset signal to be asserted to reset a UTMI PHY. Not applicable to ULPI because ULPI PHYs are reset via their FunctionControl.Reset register, and the core automatically writes to this register when the core is reset (vcc_reset_n, USBCMD.HCRST, DCTL.SoftReset, or GCTL.SoftReset)
30	RW	0x1	U2_FREECLK_EXISTS U2_FREECLK_EXISTS Specifies whether your USB 2.0 PHY provides a free-running PHY clock, which is active when the clock control input is active. If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input. The remaining utmi_clk[n] must be connected to the respective port clocks. The core uses the Port-0 clock for generating the internal mac2 clock. 1'b0: USB 2.0 free clock does not exist 1'b1: USB 2.0 free clock exists Note: When the core is configured as device-only, do not set this bit to 1.

Bit	Attr	Reset Value	Description
29:25	RO	0x0	reserved
24:22	RW	0x0	LSTRD LS Turnaround Time This field indicates the value of the Rx-to-Tx packet gap for LS devices. The encoding is as follows: 0: 2 bit times 1: 2.5 bit times 2: 3 bit times 3: 3.5 bit times 4: 4 bit times 5: 4.5 bit times 6: 5 bit times 7: 5.5 bit times Note: This field is applicable only in Host mode. For normal operation (to work with most LS devices), set the default value of this field to 3'h0 (2 bit times). The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3 bit times of inter-packet gap to work correctly. Include your PHY delays when programming the LSIPD/LSTRDTIM values. For example, if your PHY's TxEndDelay in LS mode is 30 UTMI/ULPI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement.

Bit	Attr	Reset Value	Description
21:19	RW	0x2	LSIPD LS Inter-Packet Time This field indicates the value of Tx-to-Tx packet gap for LS devices. The encoding is as follows: 0: 2 bit times 1: 2.5 bit times 2: 3 bit times 3: 3.5 bit times 4: 4 bit times 5: 4.5 bit times 6: 5 bit times 7: 5.5 bit times Note: This field is applicable only in Host mode. For normal operation (to work with most LS devices), set the default value of this field to 3'h2 (3 bit times). The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the AOpen LS mouse requires 3 bit times of inter-packet gap to work correctly. Include your PHY delays when programming the LSIPD/LSTRDTIM values. For example, if your PHY's TxEndDelay in LS mode is 30 UTMI/ULPI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement.
18:14	RO	0x0	reserved
13:10	RW	0x9	USBTRDTIM USB 2.0 Turnaround Time Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). The following are the required values for the minimum SoC bus frequency of 60 MHz. USB turnaround time is a critical certification criteria when using long cables and five hub levels. The required values for this field: 4'h5: When the MAC interface is 16-bit UTMI+. 4'h9: When the MAC interface is 8-bit UTMI+/ULPI. If SoC bus clock is less than 60 MHz, and USB turnaround time is not critical, this field can be set to a larger value. Note: This field is valid only in device mode.

Bit	Attr	Reset Value	Description
9	RW	0x0	XCVRDLY Transceiver Delay Enables a delay between the assertion of the UTMI/ULPI Transceiver Select signal (for HS) and the assertion of the TxValid signal during a HS Chirp. When this bit is set to 1, a delay (of approximately 2.5 us) is introduced from the time when the Transceiver Select is set to 2'b00 (HS) to the time the TxValid is driven to 0 for sending the chirp-K. This delay is required for some UTMI/ULPI PHYs. Note: If you enable the hibernation feature when the device core comes out of power-off, you must re-initialize this bit with the appropriate value because the core does not save and restore this bit value during hibernation. This bit is valid only in device mode.
8	RW	0x0	ENBLSLPM Enable utmi_sleep_n and utmi_l1_suspend_n The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state. 1'b0: utmi_sleep_n and utmi_l1_suspend_n assertion from the core is not transferred to the external PHY. 1'b1: utmi_sleep_n and utmi_l1_suspend_n assertion from the core is transferred to the external PHY. Note: In Device mode - Before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. Without disabling this bit, if a command is issued when the device is in L1 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not get completed.
7	RO	0x0	PHYSEL USB 2.0 High-Speed PHY or USB 1.1 Full-Speed 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY. 1'b1: USB 1.1 full-speed serial transceiver.

Bit	Attr	Reset Value	Description
6	RW	0x0	SUSPENDUSB20 Suspend USB2.0 HS/FS/LS PHY When set, USB2.0 PHY enters Suspend mode if Suspend conditions are valid. For DRD/OTG configurations, it is recommended that this bit is set to 0 during coreConsultant configuration. If it is set to 1, then the application must clear this bit after power-on reset. Application needs to set it to 1 after the core initialization completes. For all other configurations, this bit can be set to 1 during core configuration. Note: In host mode, on reset, this bit is set to 1. Software can override this bit after reset. In device mode, before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. If you issue a command without disabling this bit when the device is in L2 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not get completed.
5	RO	0x0	reserved
4	RO	0x0	ULPI_UTMI_Sel ULPI or UTMI+ Select 1'b0: UTMI+ Interface 1'b1: ULPI Interface
3	RW	0x0	PHYIF PHY Interface If UTMI+ is selected, the application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. 1'b0: 8 bits 1'b1: 16 bits

Bit	Attr	Reset Value	Description
2:0	RW	0x0	TOutCal HS/FS Timeout Calibration The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the core to account for additional delays introduced by the PHY. This may be required, since the delay introduced by the PHY in generating the linestate condition may vary among PHYs. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of connection. The number of bit times added per PHY clock are: High-speed operation: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full-speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times

USB3_GUSB3PIPECTL0
Address: Operational Base + offset (0xc2c0)
Global USB3 PIPE Control Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	PHYSoftRst USB3 PHY Soft Reset After setting this bit to 1, the software needs to clear this bit.

Bit	Attr	Reset Value	Description
30	RW	0x0	HstPrtCmpl HstPrtCmpl This feature tests the PIPE PHY compliance patterns without having to have a test fixture on the USB 3.0 cable. This bit enables placing the SS port link into a compliance state. By default, this bit must be set to 1'b0. In compliance lab testing, the SS port link enters compliance after failing the first polling sequence after power on. Set this bit to 0, when you run compliance tests. The sequence for using this functionality is as follows: 1. Disconnect any plugged in devices. 2. Perform USBCMD.HCRST or power-on-chip reset. 3. Set PORTSC.PP=0. 4. Set GUSB3PIPECTL. HstPrtCmpl=1. This places the link into compliance state. To advance the compliance pattern, follow this sequence (toggle the set GUSB3PIPECTL. HstPrtCmpl): 1. Set GUSB3PIPECTL.HstPrtCmpl=0. 2. Set GUSB3PIPECTL.HstPrtCmpl=1. This advances the link to the next compliance pattern. To exit from the compliance state perform USBCMD.HCRST or power-on-chip reset.
29	RW	0x0	U2SSInactP3ok P3 OK for U2/SSInactive 0: During link state U2/SS.Inactive, put PHY in P2 (Default) 1: During link state U2/SS.Inactive, put PHY in P3. Note: For a port, if GUSB3PIPECTL[7]=1 and GUSB3PIPECTL[29]=1, set GUSB3PIPECTL[11] to 1.
28	RW	0x0	DisRxDetP3 Disabled receiver detection in P3 0: If PHY is in P3 and Core needs to perform receiver detection, The core performs receiver detection in P3. (Default) 1: If PHY is in P3 and Core needs to perform receiver detection, The core changes the PHY power state to P2 and then performs receiver detection. After receiver detection, the cores changes PHY power state to P3.
27	RW	0x0	Ux_exit_in_Px Ux Exit in Px 0: The core does U1/U2/U3 exit in PHY power state P0 (default behavior). 1: The core does U1/U2/U3 exit in PHY power state P1/P2/P3 respectively. This bit is added for SS PHY workaround where SS PHY injects a glitch on pipe3_RxElecIdle while receiving Ux exit LFPS, and pipe3_PowerDown change is in progress.

Bit	Attr	Reset Value	Description
26	RW	0x0	ping_enhancement_en Ping Enhancement Enable When set, the Downstream port U1 ping receive timeout becomes 500 ms instead of 300 ms. Minimum Ping.LFPS receive duration is 8 ns (one mac3_clk). This field is valid for the downstream port only.
25	RW	0x0	u1u2exitfail_to_recov U1U2exitfail to Recovery When set, and U1/U2 LFPS handshake fails, the LTSSM transitions from U1/U2 to Recovery instead of SS Inactive. If Recovery fails, then the LTSSM can enter SS.Inactive. This is an enhancement only. It prevents interoperability issue if the remote link does not do proper handshake.
24	RW	0x1	request_p1p2p3 Always Request P1/P2/P3 for U1/U2/U3 When set, the core always requests PHY power change from P0 to P1/P2/P3 during U0 to U1/U2/U3 transition. If this bit is 0, and immediate Ux exit (remotely initiated, or locally initiated) happens, the core does not request P1/P2/P3 power state change. Note: This bit must be set to 1 for Synopsys PHY. For third-party SS PHY, check with your PHY vendor.
23	RW	0x0	StartRxDetU3RxDet Start Receiver Detection in U3/Rx.Detect If DWC_USB3_GUSB3PIPECTL_INIT[22] is set, and the link is in either U3 or Rx.Detect state, the core starts receiver detection on the rising edge of this bit. This can only be used for Downstream ports. This bit must be set to 0 for Upstream ports.
22	RW	0x0	DisRxDetU3RxDet Disable Receiver Detection in U3/Rx.Det When set, the core does not handle receiver detection in either U3 or Rx.Detect states. DWC_USB3_GUSB3PIPECTL_INIT[23] must be used to start receiver detection manually. This bit can only be used for the downstream port. This bit must be set to 0 for Upstream ports.
21:19	RW	0x1	DelayP1P2P3 Delay P1P2P3 Delay P0 to P1/P2/P3 request when entering U1/U2/U3 until (DWC_USB3_GUSB3PIPECTL_INIT[21:19]*8) 8B10B error occurs, or Pipe3_RxValid drops to 0. DWC_USB3_GUSB3PIPECTL_INIT[18] must be 1 to enable this functionality.

Bit	Attr	Reset Value	Description
18	RW	0x1	DELAYP1TRANS DELAYP1TRANS Delay PHY power change from P0 to P1/P2/P3 when link state changing from U0 to U1/U2/U3 respectively. 1'b1: When entering U1/U2/U3, delay the transition to P1/P2/P3 until the pipe3 signals, Pipe3_RxElecIlde is 1 and pipe3_RxValid is 0 1'b0: When entering U1/U2/U3, transition to P1/P2/P3 without checking for Pipe3_RxElecIlde and pipe3_RxValid.
17	RW	0x0	SUSPENDENABLE Suspend USB3.0 SS PHY When set, and if Suspend conditions are valid, the USB 3.0 PHY enters Suspend mode. For DRD/OTG configurations, it is recommended that this bit is set to '0' during coreConsultant configuration. If it is set to '1', then the application must clear this bit after power-on reset. Application needs to set it to '1' after the core initialization is completed. For all other configurations, this bit can be set to '1' during core configuration.
16:15	RW	0x0	DATWIDTH PIPE Data Width 2'b00: 32 bits 2'b01: 16 bits 2'b10: 8 bits Note: USB3 controller only support 32-bit width pipe interface.
14	RW	0x0	AbortRxDetInU2 Abort Rx Detect in U2 When set and the link state is U2, then the core will abort receiver detection if it receives U2 exit LFPS from the remote link partner. This bit is for the downstream port only.
13	RW	0x0	SkipRxDet Skip Rx Detect When set, the core skips Rx Detection if pipe3_RxElecIdle is low. Skip is defined as waiting for the appropriate timeout, then repeating the operation.
12	RW	0x0	LFPSP0Algn LFPS P0 Align When set: 1. The core deasserts LFPS transmission on the clock edge that it requests Phy power state 0 when exiting U1, U2, or U3 low power states. Otherwise, LFPS transmission is asserted one clock earlier. 2. The core requests symbol transmission two pipe3_rx_pclks periods after the PHY asserts PhyStatus as a result of the PHY switching from P1 or P2 state to P0 state.

Bit	Attr	Reset Value	Description
11	RW	0x0	P3P2TranOK P3 P2 Transitions OK When set, the core transitions directly from Phy power state P2 to P3 or from state P3 to P2. When not set, P0 is always entered as an intermediate state during transitions between P2 and P3, as defined in the PIPE3 Specification. According to the PIPE3 Specification, any direct transition between P3 and P2 is illegal.
10	RW	0x0	P3ExSigP2 P3 Exit Signal in P2 When this bit is set, the core always changes the PHY power state to P2, before attempting a U3 exit handshake.
9	RW	0x0	LFPSFILTER LFPS Filter When set, filter LFPS reception with pipe3_RxValid in PHY power state P0, that is, ignore LFPS reception from the PHY unless both pipe3_Rxelecidle and pipe3_RxValid are deasserted.
8	RW	0x0	RX_DETECT_to_Polling_L RX_DETECT to Polling.LFPS Control 1'b0 (Default): Enables a 400us delay to start Polling LFPS after RX_DETECT. This allows VCM offset to settle to a proper level. 1'b1: Disables the 400us delay to start Polling LFPS after RX_DETECT. During controller certification with third party PHY it is observed that the PHY is not able to meet the Tx AC common mode voltage active (VTX-CM-ACPP_ACTIVE <100mv) if the link starts polling within 80us from the time rx.detect is performed. To meet this VTX-CM-ACPP_ACTIVE specification, the polling must be delayed further. If the PHY does not have issue then they can set this bit to 1 which allows polling to start within 80us.
7	RO	0x0	reserved
6	RW	0x0	TX_SWING Tx Swing Drive the setting value to the pipe interface of PHY.
5:3	RW	0x0	TX_MARGIN Tx Margin[2:0] Drive the setting value to the pipe interface of PHY.
2:1	RW	0x1	TX_DE_EPPHASIS Tx Deemphasis The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode.
0	RW	0x0	ELASTIC_BUFFER_MODE Elastic Buffer Mode Drive the setting value to the pipe interface of PHY.

USB3_GTXFIFOSIZn
Address: Operational Base + offset (0xc300 + 4*n), n=0~6
Global Transmit FIFO Size Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	TXFSTADDR_N Transmit FIFOn RAM Start Address This field contains the memory start address for TxFIFOn in 64-bit words.
15:0	RW	0x0042	TXFDEP_N TxFIFO Depth This field contains the depth of TxFIFOn in 64-bit words. Minimum value: 32; Maximum value: 32,768

USB3_GRXFIFOSIZn
Address: Operational Base + offset (0xc380 + 4*n), n=0~2
Global Receive FIFO Size Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	RXFSTADDR_N RxFIFOn RAM Start Address This field contains the memory start address for RxFIFOn in 64-bit words.
15:0	RW	0x0285	RXFDEP_N RxFIFO Depth This field contains the depth of RxFIFOn in 64-bit words. Minimum value: 32; Maximum value: 16,384

USB3_GEVNTADRLO0
Address: Operational Base + offset (0xc400)
Global Event Buffer Address (Low) Register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EVNTADRLO Event Buffer Address Holds the lower 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.

USB3_GEVNTADRHI0
Address: Operational Base + offset (0xc404)
Global Event Buffer Address (High) Register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EVNTADRHI Event Buffer Address Holds the higher 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.

USB3_GEVNTSIZ0
Address: Operational Base + offset (0xc408)
Global Event Buffer Size Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	EVNTINTRPTMASK Event Interrupt Mask When set to '1', this prevents the interrupt from being generated. However, even when the mask is set, the events are queued.
30:16	RO	0x0	reserved
15:0	RW	0x0000	EVENTSIZ Event Buffer Size in bytes Holds the size of the Event Buffer in bytes; must be a multiple of four. This is programmed by software once during initialization. The minimum size of the event buffer is 32 bytes.

USB3_GEVNTCOUNT0
Address: Operational Base + offset (0xc40c)
Global Event Buffer Count Register 0

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	EVNTCOUNT Event Count When read, returns the number of valid events in the Event Buffer (in bytes). When written, hardware decrements the count by the value written. The interrupt line remains high when count is not 0.

USB3_GHWPARAMS8
Address: Operational Base + offset (0xc600)
Global Hardware Parameters Register 8

Bit	Attr	Reset Value	Description
31:0	RO	0x0000077c	ghwparams8_32_0 ghwparams8_32_0 ghwparams8_32_0

USB3_GTXFIFOPRIDEV
Address: Operational Base + offset (0xc610)
Global Device TX FIFO DMA Priority Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	gtxfifopridev Device TxFIFO priority This register specifies the relative DMA priority level among the Device TXFIFOs (one per IN endpoint). Each register bit[n] controls the priority (1: high, 0: low) of each TXFIFO[n]. When multiple TXFIFOs compete for DMA service at a given time (that is, multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner: 1. High-priority TXFIFOs are granted access using round-robin arbitration 2. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full). For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. When configuring periodic IN endpoints, software must set register bit[n]=1, where n is the TXFIFO assignment. This ensures that the DMA for isochronous or interrupt IN endpoints are prioritized over bulk or control IN endpoints. This register is present only when the core is configured to operate in the device mode. The register size corresponds to the number of Device IN endpoints.

USB3_GTXFIFOPRIHST
Address: Operational Base + offset (0xc618)
Global Host TX FIFO DMA Priority Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gtxfifoprihst Host TxFIFO priority This register specifies the relative DMA priority level among the Host TXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS). Each register bit[n] controls the priority (1: high, 0: low) of TXFIFO[n] within a speed group. When multiple TXFIFOs compete for DMA service at a given time (i.e., multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner: 1. Among the FIFOs in the same speed group (SS or HS/FSLS): a. High-priority TXFIFOs are granted access using round-robin arbitration b. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full). 2. The TX DMA arbiter prioritizes the SS speed group or HS/FSLS speed group according to the ratio programmed in the GDMAHLRATIO register. For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. This register is present only when the core is configured to operate in the host mode (includes DRD and OTG modes). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 3 USB bus instances (1 SS, 1 HS, and 1 FSLS).

USB3_GRXFIFOPRIHST
Address: Operational Base + offset (0xc61c)
Global Host RX FIFO DMA Priority Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	grxfifoprihst Host RxFIFO priority This register specifies the relative DMA priority level among the Host RXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS). Each register bit[n] controls the priority (1: high, 0: low) of RXFIFO[n] within a speed group. When multiple RXFIFOs compete for DMA service at a given time (i.e., multiple RXQs contain RX DMA requests and their corresponding RXFIFOs have data available), the RX DMA arbiter grants access on a packet-basis in the following manner: 1. Among the FIFOs in the same speed group (SS or HS/FSLS): a. High-priority RXFIFOs are granted access using round-robin arbitration b. Low-priority RXFIFOs are granted access using round-robin arbitration only after high-priority RXFIFOs have no further processing to do (that is, either the RXQs are empty or the corresponding RXFIFOs do not have the required data). 2. The RX DMA arbiter prioritizes the SS speed group or HS/FSLS speed group according to the ratio programmed in the GDMAHLRATIO register. For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. This register is present only when the core is configured to operate in the host mode (includes DRD and OTG modes). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 3 USB bus instances (1 SS, 1 HS, and 1 FSLS).

USB3_GFIFOPRIDBC
Address: Operational Base + offset (0xc620)
Global Host Debug Capability DMA Priority Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gfifopridbc Host DbC DMA priority This register specifies the relative priority of the RXFIFOs and TXFIFOs associated with the DbC mode. It overrides the priority assigned in the corresponding indexes of the Host RXFIFO and TXFIFO DMA priority registers, when the DbC mode is enabled. Priority settings are specified in relation to the low-priority SS speed group: 1. Normal priority indicates that the DbC FIFOs are considered identical to the Host SS low-priority FIFOs. 2. Low priority indicates that the DbC FIFOs are considered to have lower priority than all Host SS FIFOs. 3. High priority indicates that the DbC FIFOs are considered higher priority than the Host SS low-priority FIFOs but lower priority than the Host SS high-priority FIFOs. This register is present only when the core is configured to operate in Host Debug Capability (DbC) mode.

USB3_GDMAHLRATIO
Address: Operational Base + offset (0xc624)
Global Host FIFO DMA High-Low Priority Ratio Register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:8	RW	0x00	hstrxfifo Host RXFIFO DMA High-Low Priority This register specifies the relative priority of the SS FIFOs with respect to the HS/FSLS FIFOs. The DMA arbiter prioritizes the HS/FSLS round-robin arbiter group every DMA High-Low Priority Ratio grants as indicated in the register separately for TX and RX. To illustrate, consider that all FIFOs are requesting access simultaneously, and the ratio is 4. SS gets priority for 4 packets, HS/FSLS gets priority for 1 packet, SS gets priority for 4 packets, HS/FSLS gets priority for 1 packet, and so on. If FIFOs from both speed groups are not requesting access simultaneously then, 1. if SS got grants 4 out of the last 4 times, then HS/FSLS get the priority on any future request. 2. if HS/FSLS got the grant last time, SS gets the priority on the next request. 3. if there is a valid request on either SS or HS/FSLS, a grant is always awarded; there is no idle. This register is present if the core is configured to operate in host mode.
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x08	hsttxfifo Host TXFIFO DMA High-Low Priority This register specifies the relative priority of the SS FIFOs with respect to the HS/FSLS FIFOs. The DMA arbiter prioritizes the HS/FSLS round-robin arbiter group every DMA High-Low Priority Ratio grants as indicated in the register separately for TX and RX. To illustrate, consider that all FIFOs are requesting access simultaneously, and the ratio is 4. SS gets priority for 4 packets, HS/FSLS gets priority for 1 packet, SS gets priority for 4 packets, HS/FSLS gets priority for 1 packet, and so on. If FIFOs from both speed groups are not requesting access simultaneously then, 1. if SS got grants 4 out of the last 4 times, then HS/FSLS get the priority on any future request. 2. if HS/FSLS got the grant last time, SS gets the priority on the next request. 3. if there is a valid request on either SS or HS/FSLS, a grant is always awarded; there is no idle. This register is present if the core is configured to operate in host mode.

USB3_GFLADJ

Address: Operational Base + offset (0xc630)

Global Frame Length Adjustment Register

Bit	Attr	Reset Value	Description
31	RW	0x0	GFLADJ_REFCLK_240MHZDECR_PLS1 GFLADJ_REFCLK_240MHZDECR_PLS1 This field indicates that the decrement value that the controller applies for each ref_clk must be GFLADJ_REFCLK_240MHZ_DECR and GFLADJ_REFCLK_240MHZ_DECR +1 alternatively on each ref_clk. Set this bit to a 1 only if GFLADJ_REFCLK_LPM_SEL is set to 1 and the fractional component of 240/ref_frequency is greater than or equal to 0.5. Examples: If the ref_clk is 24 MHz then 1. GUCTL.REF_CLK_PERIOD = 41 2. GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = (240/24) = 10 3. GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 0

Bit	Attr	Reset Value	Description
30:24	RW	0x00	GFLADJ_REFCLK_240MHZ_DECR GFLADJ_REFCLK_240MHZ_DECR This field indicates the decrement value that the controller applies for each ref_clk in order to derive a frame timer in terms of a 240-MHz clock. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to 1. The value is derived as follows: GFLADJ_REFCLK_240MHZ_DECR = 240/ref_clk_frequency Examples: If the ref_clk is 24 MHz then 1. GUCTL.REF_CLK_PERIOD = 41 2. GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/24 = 10
23	RW	0x0	GFLADJ_REFCLK_LPM_SEL GFLADJ_REFCLK_LPM_SEL This bit enables the functionality of running SOF/ITP counters on the ref_clk. This bit must not be set to 1 if GCTL.SOFITPSYNC bit is set to 1. Similarly, if GFLADJ_REFCLK_LPM_SEL set to 1, GCTL.SOFITPSYNC must not be set to 1. When GFLADJ_REFCLK_LPM_SEL is set to 1 the overloading of the suspend control of the USB 2.0 first port PHY (UTMI/ULPI) with USB 3.0 port states is removed. Note that the ref_clk frequencies supported in this mode are 16/17/19.2/20/24/39.7/40 MHz. The utmi_clk[0] signal of the core must be connected to the FREECLK of the PHY. Note: If you set this bit to 1, the GUSB2PHYCFG.U2_FREECLK_EXISTS bit must be set to 0.
22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21:8	RW	0x0000	GFLADJ_REFCLK_FLADJ GFLADJ_REFCLK_FLADJ This field indicates the frame length adjustment to be applied when SOF/ITP counter is running on the ref_clk. This register value is used to adjust the ITP interval when GCTL[SOFITPSYNC] is set to 1; SOF and ITP interval when GLADJ.GFLADJ_REFCLK_LPM_SEL is set to 1. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to 1 or GCTL.SOFITPSYNC is set to 1. The value is derived as follows: FLADJ_REF_CLK_FLADJ=((125000/ref_clk_period_integer)-(125000/ref_clk_period)) * ref_clk_period where: 1. The ref_clk_period_integer is the integer value of the ref_clk period got by truncating the decimal (fractional) value that is programmed in the GUCTL.REF_CLK_PERIOD field. 2. The ref_clk_period is the ref_clk period including the fractional value. Examples: If the ref_clk is 24 MHz then 1. GUCTL.REF_CLK_PERIOD = 41 2. GFLADJ.GLADJ_REFCLK_FLADJ = ((125000/41)-(125000/41.6666))*41.6666 = 2032 (ignoring the fractional value)
7	RW	0x0	GFLADJ_30MHZ_SDBND_SEL GFLADJ_30MHZ_SDBND_SEL This field selects whether to use the input signal fladj_30mhz_reg or the GFLADJ.GFLADJ_30MHZ to adjust the frame length for the SOF/ITP. When this bit is set to: 1, the controller uses the register field GFLADJ.GFLADJ_30MHZ value 0, the controller uses the input signal fladj_30mhz_reg value
6	RO	0x0	reserved
5:0	RW	0x00	GFLADJ_30MHZ GFLADJ_30MHZ This field indicates the value that is used for frame length adjustment instead of considering from the sideband input signal fladj_30mhz_reg. This enables post-silicon frame length adjustment in case the input signal fladj_30mhz_reg is connected to a wrong value or is not valid. For details on how to set this value, refer to section 5.2.4, "Frame Length Adjustment Register (FLADJ)," of the xHCI Specification.

USB3_DCFG

Address: Operational Base + offset (0xc700)
Device Configuration Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	IgnStrmPP IgnoreStreamPP This bit only affects stream-capable bulk endpoints. When this bit is set to 0 and the controller receives a Data Packet with the Packet Pending (PP) bit set to 0 for OUT endpoints, or it receives an ACK with the NumP field set to 0 and PP set to 0 for IN endpoints, the core attempts to search for another stream (CStream) to initiate to the host. However, there are two situations where this behavior is not optimal: 1. When the host is setting PP=0 even though it has not finished the stream, or 2. When the endpoint on the device is configured with one transfer resource and therefore does not have any other streams to initiate to the host. When this bit is set to 1, the core ignores the Packet Pending bit for the purposes of stream selection and does not search for another stream when it receives DP(PP=0) or ACK(NumP=0, PP=0). This can enhance the performance when the device system bus bandwidth is low or the host responds to the core's ERDY transmission very quickly.
22	RW	0x0	LPMCAP LPM Capable The application uses this bit to control the DWC_usb3 core LPM capabilities. If the core operates as a non-LPM-capable device, it cannot respond to LPM transactions. 1'b0: LPM capability is not enabled. 1'b1: LPM capability is enabled.
21:17	RW	0x04	NUMP Number of Receive Buffers. This bit indicates the number of receive buffers to be reported in the ACK TP. The DWC_usb3 controller uses this field if GRXTHRCFG.USBRxPktCntSel is set to 0. The application can program this value based on RxFIFO size, buffer sizes programmed in descriptors, and system latency. For an OUT endpoint, this field controls the number of receive buffers reported in the NumP field of the ACK TP transmitted by the core. Note: This bit is used in host mode when Debug Capability is enabled.

Bit	Attr	Reset Value	Description
16:12	RW	0x00	INTRNUM Interrupt number Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated.
11:10	RO	0x0	reserved
9:3	RW	0x00	DEVADDR Device Address. The application must perform the following: 1. Program this field after every SetAddress request. 2. Reset this field to zero after USB reset.
2:0	RW	0x4	DEVSPD Device Speed Indicates the speed at which the application requires the core to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. 3'b100: SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) 3'b000: High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 3'b001: Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)

USB3_DCTL
Address: Operational Base + offset (0xc704)
Device Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	RUN_STOP Run/Stop The software writes 1 to this bit to start the device controller operation. To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DSTS.DevCtrlHlt bit when the core is idle and the lower layer finishes the disconnect process. The Run/Stop bit must be used in following cases as specified: 1. After power-on reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. 2. The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. The minimum duration of keeping this bit cleared is specified in the Note below. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DCTL[8:5] to 5 before reasserting the Run/Stop bit. 3. When the USB or Link is in a lower power state and the Two Power Rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the Core Power Rail. After the software turns on the Core Power Rail again and re-initializes the device controller, it must set this bit to start the device controller. Note: The following is the minimum duration under various conditions for which the soft disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect: 30ms: For SuperSpeed, when the device state is Suspended, Idle, Transmit, or Receive. 10ms: For high-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions); For full-speed/low-speed, when the device state is Suspended, Idle, or not Idle/Supended (performing transactions) To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration.

Bit	Attr	Reset Value	Description
30	R/W SC	0x0	CSFTRST Core Soft Reset Reset all clock domains as follows: 1. This bit clears the interrupts and all the CSRs except GSTS, GSNPSID, GGPIO, GUID, GUSB2PHYCFGn registers, GUSB3PIPECTLn registers, DCFG, DCTL, DEVTEN, and DSTS registers. 2. All module state machines (except the SoC Bus Slave Unit) are reset to the IDLE state, and all the TxFIFOs and the RxFIFO are flushed. 3. Any transactions on the SoC bus Master are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately. The application can write this bit at any time to reset the core. This is a self-clearing bit; the core clears this bit after all necessary logic is reset in the core, which may take several clocks depending on the core's current state. Once this bit is cleared, the software must wait at least 3 PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation.
29	RO	0x0	reserved
28:24	RW	0x00	HIRDTHRES HIRD Threshold The core asserts output signals utmi_l1_suspend_n and utmi_sleep_n on the basis of this signal: The core asserts utmi_l1_suspend_n to put the PHY into Deep Low-Power mode in L1 when both of the following are true: 1. HIRD value is greater than or equal to the value in DCTL.HIRD_Thres[3:0] 2. HIRD_Thres[4] is set to 1'b1. The core asserts utmi_sleep_n on L1 when one of the following is true: 1. If the HIRD value is less than HIRD_Thres[3:0] or 2. HIRD_Thres[4] is set to 1'b0. Note: This field must be set to '0' during SuperSpeed mode of operation.

Bit	Attr	Reset Value	Description
23:20	RW	0xf	LPM_NYET_thres LPM NYET Threshold Handshake response to LPM token specified by device application. Response depends on DCFG.LPMCap. DCFG.LPMCap is 1'b0 - The core always responds with Timeout (that is, no response). DCFG.LPMCap is 1'b1 - The core responds with an ACK on successful LPM transaction, which requires that all of the following are satisfied: 1. There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). 2. No data is pending in the Transmit FIFO and OUT endpoints not in flow controlled state (else NYET). 3. The BESL value in the LPM token is less than or equal to LPM_NYET_thres[3:0]
19	RW	0x0	KeepConnect KeepConnect When 1, this bit enables the save and restore programming model by preventing the core from disconnecting from the host when DCTL.RunStop is set to 0. It also enables the Hibernation Request Event to be generated when the link goes to U3 or L2. The device core disconnects from the host when DCTL.RunStop is set to 0. This bit indicates whether to preserve this behavior (0), or if the core must not disconnect when RunStop is set to 0 (1). This bit also prevents the LTSSM from automatically going to U0/L0 when the host requests resume from U3/L2.
18	RW	0x0	L1HibernationEn L1HibernationEn When this bit is set along with KeepConnect, the device core generates a Hibernation Request Event if L1 is enabled and the HIRD value in the LPM token is larger than the threshold programmed in DCTL.HIRD_Thres. The core does not exit the LPM L1 state until software writes Recovery into the DCTL.ULStChngReq field. This prevents corner cases where the device is entering hibernation at the same time the host is attempting to exit L1.
17	RW	0x0	CRS Controller Restore State This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to 1, the controller immediately sets DSTS.RSS to 1. When the controller has finished the restore process, it sets DSTS.RSS to 0. Note: When read, this field always returns 0.

Bit	Attr	Reset Value	Description
16	RW	0x0	CSS Controller Save State This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to 1, the controller immediately sets DSTS.SSS to 1. When the controller has finished the save process, it sets DSTS.SSS to 0. Note: When read, this field always returns 0.
15:13	RO	0x0	reserved
12	RW	0x0	INITU2ENA Initiate U2 Enable 1'b0: May not initiate U2 (default) 1'b1: May initiate U2 On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U2_ENABLE), and clears this bit when ClearFeature(U2_ENABLE) is received. If DCTL[11] (AcceptU2Ena) is 0, the link immediately exits U2 state.
11	RW	0x0	ACCEPTU2ENA Accept U2 Enable 1'b0: Reject U2 except when Force_LinkPM_Accept bit is set (default) 1'b1: Core accepts transition to U2 state if nothing is pending on the application side. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.
10	RW	0x0	INITU1ENA Initiate U1 Enable 1'b0: May not initiate U1 (default); 1'b1: May initiate U1. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U1_ENABLE), and clears this bit when ClearFeature(U1_ENABLE) is received. If DCTL[9] (AcceptU1Ena) is 0, the link immediately exits U1 state.
9	RW	0x0	ACCEPTU1ENA Accept U1 Enable 1'b0: Core rejects U1 except when Force_LinkPM_Accept bit is set (default) 1'b1: Core accepts transition to U1 state if nothing is pending on the application side. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.

Bit	Attr	Reset Value	Description
8:5	RW	0x0	ULSTCHNGREQ ULSTCHNGREQ Software writes this field to issue a USB/Link state change request. A change in this field indicates a new request to the core. If software wants to issue the same request back-to-back, it must write a 0 to this field between the two requests. The result of the state change request is reflected in the USB/Link State in DSTS. These bits are self-cleared on the MAC Layer exiting suspended state. If software is updating other fields of the DCTL register and not intending to force any link state change, then it must write a 0 to this field. SS Compliance mode is normally entered and controlled by the remote link partner. Refer to the USB 3.0 specification. Alternatively, you can force the local link directly into compliance mode, by resetting the SS link with the RUN/STOP bit set to zero. If you then write 10 to the USB/Link State Change field and 1 to RUN/STOP, the link goes to compliance mode. Once you are in compliance, you may alternately write zero and 10 to this field to advance the compliance pattern. In SS mode: Value: Requested Link State Transition/Action 0: No Action 4: SS.Disabled 5: Rx.Detect 6: SS.Inactive 8: Recovery 10: Compliance Others: Reserved In HS/FS/LS mode: Value:Requested USB state transition 8: Remote wakeup request Others: Reserved The Remote wakeup request must be issued 2us after the device goes into suspend state (DSTS[21:18] is 3). Note: After coming out of hibernation, software must write 8 (Recovery) into this field to confirm exit from the suspended state.

Bit	Attr	Reset Value	Description
4:1	RW	0x0	TSTCTL Test Control 4'b000: Test mode disabled 4'b001: Test_J mode 4'b010: Test_K mode 4'b011: Test_SE0_NAK mode 4'b100: Test_Packet mode 4'b101: Test_Force_Enable Others: Reserved
0	RO	0x0	reserved

USB3_DEVTEN
Address: Operational Base + offset (0xc708)
Device Event Enable Register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	VENDEVTSTRCVDEN Vendor Device Test LMP Received Event 1: Enable this event 0: Disable this event
11:10	RO	0x0	reserved
9	RW	0x0	ERRTICERREVTEN Erratic Error Event Enable 1: Enable this event 0: Disable this event
8	RO	0x0	reserved
7	RW	0x0	SOFTEVTEN Start of (u)frame Event Enable 1: Enable this event 0: Disable this event
6	RW	0x0	U3L2L1SuspEn U3/L2-L1 Suspend Event Enable 1: Enable this event 0: Disable this event
5	RW	0x0	HibernationReqEvtEn Hibernation Request Event Enable 1: Enable this event 0: Disable this event
4	RW	0x0	WKUPEVTEN Resume/Remote Wakeup Detected Event Enable 1: Enable this event 0: Disable this event

Bit	Attr	Reset Value	Description
3	RW	0x0	ULSTCNGEN USB/Link State Change Event Enable 1: Enable this event 0: Disable this event
2	RW	0x0	CONNECTDONEEVTEN Connection Done Event Enable 1: Enable this event 0: Disable this event
1	RW	0x0	USBRSTEVTEN USB Reset Event Enable 1: Enable this event 0: Disable this event
0	RW	0x0	DISSCONNEVTEN Disconnect Detected Event Enable 1: Enable this event 0: Disable this event

USB3_DSTS
Address: Operational Base + offset (0xc70c)
Device Status Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	DCNRD Device Controller Not Ready The bit indicates that the core is in the process of completing the state transitions after exiting from hibernation. To complete the state transitions, it takes 256 bus clock cycles from the time DCTL[31].Run/Stop is set. During hibernation, if the UTMI/ULPI PHY is in suspended state, then the 256-bus clock cycle delay starts after the PHY exited suspended state. Software must set DCTL[31].Run/Stop to 1 and wait for this bit to be de-asserted to zero before processing DSTS.USBLnkSt.
28:26	RO	0x0	reserved
25	RO	0x0	RSS RSS Restore State Status This bit is similar to the USBSTS.RSS in host mode. When the controller finishes the restore process, it completes the command by setting DSTS.RSS to 0.
24	RO	0x0	SSS SSS Save State Status This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DSTS.SSS to 0.

Bit	Attr	Reset Value	Description
23	RO	0x0	COREIDLE Core Idle The bit indicates that the core finished transferring all RxFIFO data to system memory, writing out all completed descriptors, and all Event Counts are zero. Note: While testing for Reset values, mask out the read value. This bit represents the changing state of the core and does not hold a static value.
22	RO	0x1	DEVCTRLHLT Device Controller Halted This bit is set to 0 when the Run/Stop bit in the DCTL register is set to 1. The core sets this bit to 1 when, after SW sets Run/Stop to 0, the core is idle and the lower layer finishes the disconnect process. When Halted=1, the core does not generate Device events. Note: The core does not set this bit to 1 if GEVNTCOUNTn has some valid value. Software needs to acknowledge the events that are generated (by writing to GEVNTCOUNTn) while it is waiting for this bit to be set to 1.

Bit	Attr	Reset Value	Description
21:18	RO	0x4	USBLNKST USBLNKST SS mode: LTSSM State 4'h0: U0 4'h1: U1 4'h2: U2 4'h3: U3 4'h5: RX_DET 4'h6: SS_INACT 4'h7: POLL 4'h8: RECOV 4'h9: HRESET 4'ha: CMPLY 4'hb: LPBK 4'hf: Resume/Reset In HS/FS/LS mode: 4'h0: On state 4'h2: Sleep (L1) state 4'h3: Suspend (L2) state 4'h4: Disconnected state (Default state) 4'h5: Early Suspend state (valid only when Hibernation is disabled, GCTL[1].GblHibernationEn = 0) 4'he: Reset (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) 4'hf: Resume (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) The link state Resume/Reset indicates that the core received a resume or USB reset request from the host while the link was in hibernation. Software must write 8 (Recovery) to the DCTL.ULStChngReq field to acknowledge the resume/reset request. When Hibernation is enabled, GCTL[1].GblHibernationEn = 1, this field USBLnkSt is valid only when DCTL[31].Run/Stop set to 1 and DSTS[29].DCNRD = 0.
17:3	RO	0x0	reserved
2:0	RU	0x4	CONNECTSPD Connected Speed Indicates the speed at which the DWC_usb3 core has come up after speed detection through a chirp sequence. 3'b100: SuperSpeed (PHY clock is running at 125 or 250 MHz) 3'b000: High-speed (PHY clock is running at 30 or 60 MHz) 3'b001: Full-speed (PHY clock is running at 30 or 60 MHz)

USB3_DGCMDPAR
Address: Operational Base + offset (0xc710)
Device Generic Command Parameter Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PARAMETER PARAMETER This register indicates the device command parameter. This must be programmed before or along with the device command. The available device commands are listed in DGCMD register.

USB3_DGCMD
Address: Operational Base + offset (0xc714)
Device Generic Command Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:12	RO	0x0	CMDSTATUS Command Status 1: CmdErr: Indicates that the device controller encountered an error while processing the command. 0: Indicates command success
11	RO	0x0	reserved
10	R/WSC	0x0	CMDACT Command Active The software sets this bit to 1 to enable the device controller to execute the generic command. The device controller sets this bit to 0 after executing the command.
9	RO	0x0	reserved
8	RW	0x0	CMDIOC Command Interrupt on Complete When this bit is set, the device controller issues a Generic Command Completion event after executing the command. Note that this interrupt is mapped to DCFG.IntrNum. Note: This field must not set to 1 if the DCTL.RunStop field is 0.
7:0	RW	0x00	CMDTYP Command Type Specifies the type of command the software driver is requesting the core to perform. 00h: Reserved 01h: Set Endpoint Configuration - 64 or 96-bit Parameter 02h: Set Endpoint Transfer Resource Configuration - 32-bit Parameter 03h: Get Endpoint State - No Parameter Needed 05h: Clear Stall (see Set Stall) - No Parameter Needed 06h: Start Transfer - 64-bit Parameter 07h: Update Transfer - No Parameter Needed 08h: End Transfer - No Parameter Needed 09h: Start New Configuration - No Parameter Needed

USB3_DALEPENA
Address: Operational Base + offset (0xc720)
Device Active USB Endpoint Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	USBACTEP USB Active Endpoints This field indicates if a USB endpoint is active in the current configuration and interface. It applies to USB IN endpoints 0~15 and OUT endpoints 0~15, with one bit for each of the 32 possible endpoints. Even numbers are for USB OUT endpoints, and odd numbers are for USB IN endpoints, as follows: Bit[0]: USB EP0-OUT Bit[1]: USB EP0-IN Bit[2]: USB EP1-OUT Bit[3]: USB EP1-IN ... The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USBReset. Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set these bits. For more information, Pls see "Flexible Endpoint Mapping" section.

USB3_DEPnCMDPAR2
Address: Operational Base + offset (0xc800 + 4*n), n=0~12
Device Physical Endpoint-n Command Parameter 2 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PARAMETER PARAMETER This register indicates the physical endpoint command Parameter 2. It must be programmed before issuing the command.

USB3_DEPnCMDPAR1
Address: Operational Base + offset (0xc804 + 4*n), n=0~12
Device Physical Endpoint-n Command Parameter 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PARAMETER PARAMETER This register indicates the physical endpoint command Parameter 1. It must be programmed before issuing the command.

USB3_DEPnCMDPAR0
Address: Operational Base + offset (0xc808 + 4*n), n=0~12
Device Physical Endpoint-n Command Parameter 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PARAMETER PARAMETER This register indicates the physical endpoint command Parameter 0. It must be programmed before issuing the command.

USB3_DEPnCMD

Address: Operational Base + offset (0xc80c + 4*n), n=0~12

Device Physical Endpoint-n Command Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	COMMANDPARAM Command Parameters or Event Parameters when this register is written: For Start Transfer command: The 16-bit StreamID assigned to this transfer For Start Transfer command applied to an isochronous endpoint: StartMicroFramNum, Indicates the (micro)frame number to which the first TRB applies. For Update Transfer, End Transfer, and Start New Configuration commands: [22:16]: Transfer Resource Index (XferRscIdx). The hardware-assigned transfer resource index for the transfer, which was returned in response to the Start Transfer command. The application software-assigned transfer resource index for a Start New Configuration command. Event Parameters (EventParam), when this register is read. Please see bits [31:16] in the "Device Endpoint-n Events: DEPEVT" of the Databook.
15:12	RW	0x0	CMDSTATUS Command Completion Status The information is in the same format as bits 15:12 of the Endpoint Command Complete event, Please see "Device Endpoint-n Events: DEPEVT" in the Databook.
11	RW	0x0	HIPRI_FORCERM HighPriority/ForceRM HighPriority: Only valid for Start Transfer command ForceRM: Only valid for End Transfer command ClearPendIN: Only valid for Clear Stall command . Software sets this bit to clear any pending IN transaction (on that endpoint) stuck at the lower layers when a Clear Stall command is issued.

Bit	Attr	Reset Value	Description
10	RW	0x0	CMDACT Command Active Software sets this bit to 1 to enable the device endpoint controller to execute the generic command. The device controller sets this bit to 0 when the CmdStatus field is valid and the endpoint is ready to accept another command. This does not imply that all the effects of the previously-issued command have taken place.
9	RO	0x0	reserved
8	RW	0x0	CMDIOC Command Interrupt on Complete When this bit is set, the device controller issues a generic Endpoint Command Complete event after executing the command. Note that this interrupt is mapped to DEPCFG.IntrNum. When the DEPCFG command is executed, the command interrupt on completion goes to the interrupt pointed by the DEPCFG.IntrNum in the current command. Note: This field must not set to 1 if the DCTL.RunStop field is 0.
7:4	RO	0x0	reserved
3:0	RW	0x0	CMDTYP Command Type Specifies the type of command the software driver is requesting the core to perform. 00h: Reserved 01h: Set Endpoint Configuration - -64 or 96-bit Parameter 02h: Set Endpoint Transfer Resource Configuration - 32-bitparameter 03h: Get Endpoint State - No Parameter Needed 04h: Set Stall - No Parameter Needed 05h: Clear Stall (see Set Stall) - No Parameter Needed 06h: Start Transfer - 64-bit Parameter 07h: Update Transfer - No Parameter Needed 08h: End Transfer - No Parameter Needed 09h: Start New Configuration - No Parameter Needed

4.5 Interface Description

Please see Chapter TypeC PHY and USB2.0 PHY.

4.6 Application Notes

4.6.1 Some Special Settings before Initialization

Set USB3.0 OTG controller AXI master setting.
Clear USB2.0 only mode setting (bit 3 of register GRF_USB3PHY0/1_CON0 in Chapter GRF)
USB3.0 OTG controller should be hold in reset during the initialization of the corresponding

TypeC PHY until TypeC PHY is ready for USB operation.
Set PHYIF to 1 to use 16-bit UTMI+ interface (see register GUSB2PHYCFG0)
Clear ENBLSLPM to 0 to disable sleep and l1 suspend (see register GUSB2PHYCFG0)
Clear U2_FREECLK_EXITSTS to 0 (see register GUSB2PHYCFG0)
Clear DEV_FORCE_20_CLK_FOR_30_CLK to 0 (see register GUCTL1)
Clear DELAYP1TRANS to 0 (see register GUSB3PIPECTL0)

4.6.2 OTG Programming Model

When detect ID change event (bit 4/5 of GRF_SIG_DETECT_STATUS in Chapter GRF for USB3.0 OTG controller 0; bit 9/10 of GRF_SIG_DETECT_STATUS in Chapter GRF for USB3.0 OTG controller 1) or VBUS change event (bit 3 of GRF_SIG_DETECT_STATUS in Chapter GRF for USB3.0 OTG controller 0; bit 8 of GRF_SIG_DETECT_STATUS in Chapter GRF for USB3.0 OTG controller 1) after disconnect, it means a USB3/2 OTG A device or B device may connect, then check status of ID (bit 8 or bit 11 of GRF_SOC_STATUS3). If ID==0, it will work as A device, then follow host programming flow; if ID==1, it will work as B device, then it follow device programming flow.
Note: USB3.0 OTG doesn't support host/device mode swapping through HNP and RSP.

4.6.3 TypeC Programming Model

After data role contract of TypeC PD controller is finished by detecting a valid DFP-to-UFP connection or DR_Swap message by PD communication, if data role is UFP, then re-initial USB3.0 OTG controller following static device programming flow; if data role is DFP, then re-initial USB3.0 OTG controller following static host programming flow.

Chapter 5 HDMI TX

5.1 Overview

HDMI TX is fully compliant with HDMI 1.4a and 2.0a specification. It offers a simple implementation for consumer electronics like DVD/player/recorder and camcorder. HDMI TX consists of one HMDI transmitter controller and one HMDI transmitter PHY.
It supports following features:

- Video formats:
  - All CEA-861-E video formats up to 1080p at 60 Hz and 720p/1080i at 120 Hz
  - HDMI 1.4b video formats
    - All CEA-861-E video formats up to 1080p at 120 Hz
    - HDMI 1.4b 4K x 2K video formats
    - HDMI 1.4b 3D video modes with up to 340 MHz (TMDS clock)
  - HDMI 2.0 video formats, All CEA-861-F video formats
- Colorimetry, 24-bit RGB 4:4:4
- Pixel clock from 13.5 MHz up to 600 MHz
- Up to 192 kHz IEC60958 audio sampling rate
- Flexible synchronous enable per clock domain to set functional power down modes
- AMBA APB 3.0 register access
- I2C DDC, EDID block read mode
- SCDC I2C DDC access
- TMDS Scrambler to enable support for 2160p@60Hz with RGB 4:4:4
- Integrated CEC hardware engine

5.2 Block Diagram

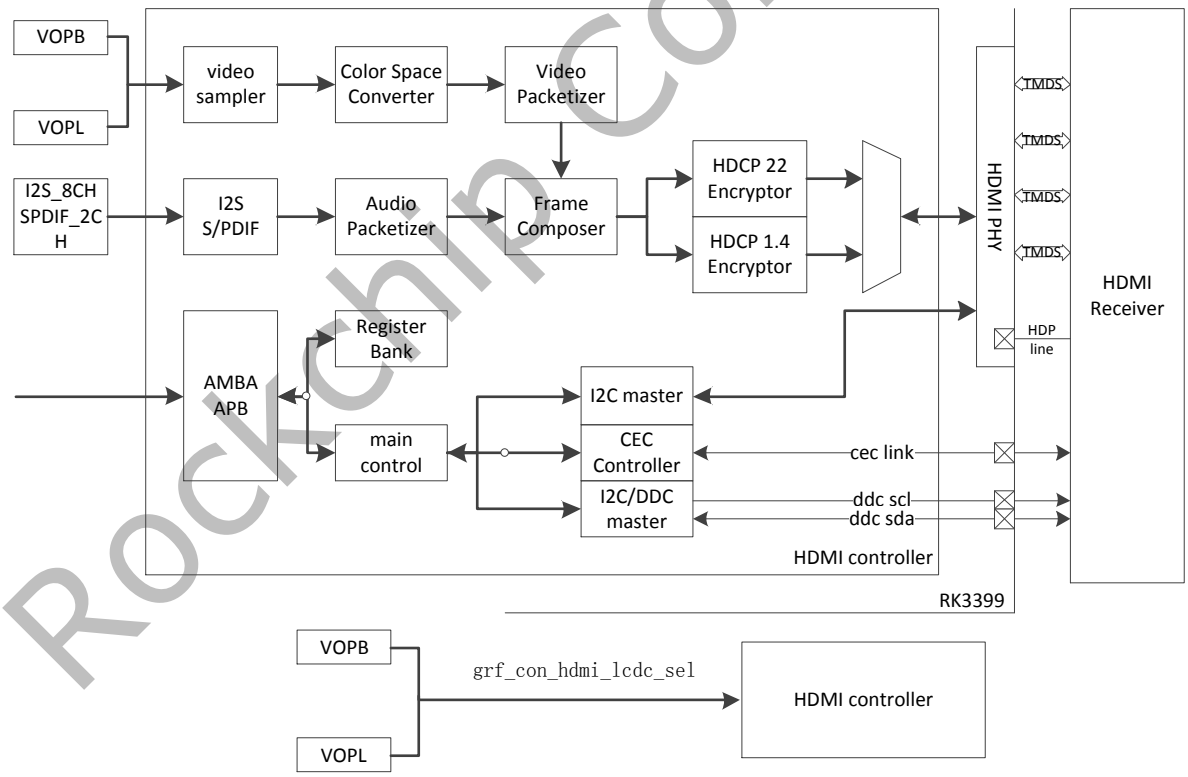


Fig. 5-1 HDMI TX Block Diagram

5.3 Function Description

5.3.1 IOMUX

Customer should notice that CEC channel use

IO_HDMIcecinout_EDPhotplug_GPIO1830gpio4c7 and DDC channel use PAD IO_I2C3HDMIscl_UART2DBGBsout_HDMII2Cscl_GPIO1830gpio4c1 and IO_I2C3HDMIsda_UART2DBGBsin_HDMII2Csda_GPIO1830gpio4c0.

Note: PAD IO_HDMI_DDCCEC is Ground reference for the Hot Plug Detect signal.

5.3.2 Video Data Processing

The video processing contains video format timings, pixel encodings (RGB to YCbCr, or YCbCr to RGB), colorimetry and corresponding requirements. This function is implemented by some functional blocks, Video Capture block, Color Space Conversion block, and Deep Color block.

The input video pixels can be encoded in either RGB, YCBCR 4:4:4 or YCBCR 4:2:2 formats by Color Space Conversion block.

The input Video data can have a pixel size of 24bits. The deep color block is used to deal with different pixel size. Video at the default 24-bit color depth is carried at a TMDS clock rate equal to the pixel clock rate. Higher color depths are carried using a correspondingly higher TMDS clock rate. HDMI Transmitter support video formats with TMDS rates below 25MHz (e.g. 13.5MHz for 480i/NTSC) that can be transmitted using a pixel-repetition scheme by setting relative registers.

Color Space Conversion

HDMI Transmitter Color space conversion (CSC) is responsible for carrying out the following video color space conversion functions:

- RGB to/from YCbCr
- 4:2:2 to/from 4:4:4 up (pixel repetition or linear interpolation)/down-converter
- Limited to/from full quantization range conversion
- The CSC supports all the timings reported in the CEA-861-D specification and the following pixel modes:
- RGB 444 and YCbCr 444: 24, 30, 36, and 48 bits
- YCbCr 422: 16, 20, and 24 bits

The color space conversion matrix is ruled by the following equations listed in below figure.

$$out_1 = (X_1 \times in_1/4096 + X_2 \times in_2/4096 + X_3 \times in_3/4096 + X_4) \times 2^{scale}$$
$$out_2 = (Y_1 \times in_1/4096 + Y_2 \times in_2/4096 + Y_3 \times in_3/4096 + Y_4) \times 2^{scale}$$
$$out_3 = (Z_1 \times in_1/4096 + Z_2 \times in_2/4096 + Z_3 \times in_3/4096 + Z_4) \times 2^{scale}$$

Fig. 5-2 HDMI Color Space Conversion Matrix Equations

Note: Color Space Conversion to and from YCrCb 4:2:0 is not supported.

5.3.3 Audio Data Processing

The HDMI TX audio process contain audio clock regeneration, placement of audio samples within packets, packet timing control, audio sample rates setting, and channel/speaker assignments. This function is implemented by Audio Capture blocks

The Audio Capture support either SPDIF or four channel I2S input. SPDIF input supports audio sampling rates from 32 to 192 KHz. The I2S input supports from 2-channel to 8-channel audio up to 192 KHz.

The scheme of audio processing as shown in the figure below:

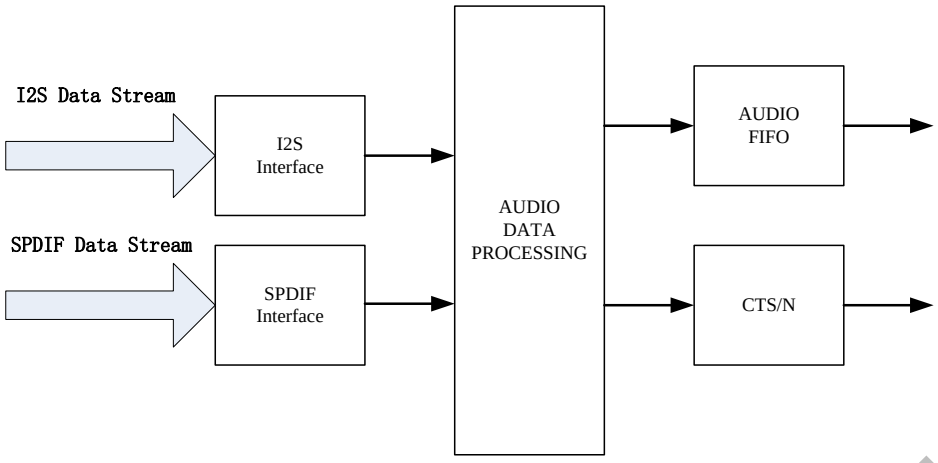


Fig. 5-3 HDMI Audio Data Processing Diagram

1.I2S

The function of this module is to implement I2S audio input feature. The incoming audio stream is captured, processed then transmitted into the TMDS link. Four I2S inputs also allow transmission of DVD-Audio and decoded Dolby Digital to A/V Receivers and high-end displays. The interface supports from 2-channel to 8-channel audio up to 192 kHz. The I2S pins must also be coherent with mclk. The appropriate registers must be configured to describe the format of audio being input. This information is passed over the HDMI link in the CEA-861D Audio Info (AI) packets. Table shows the I2S 8 channel audio formats that are supported for each of the video formats.

Table 5-1 HDMI TX I2S 2 Channel Audio Sampling Frequency

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
720x480p /720x576p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1440x480i/ 1440x576i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 5-2 HDMI TX I2S 8 Channel Audio Sampling Frequency

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
720x480p /720x576p	Yes	Yes	Yes	No	No	No	No
1440x480i/ 1440x576i	Yes	Yes	Yes	Yes	No	No	No
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

2.SPDIF

The function of this module is to implement SPDIF audio input feature. The incoming audio stream is captured, processed then transmitted into the TMDS link. SPDIF stream can carry 2-channel uncompressed PCM data (IEC 60958) or a compressed bit stream for multi-channel (IEC 61937) formats. The audio data capture logic forms the audio data into packets in accordance with the HDMI specification. SPDIF input supports audio sampling rates from 32 to 192 KHz. The following shows the SPDIF audio formats that are supported for each of the video formats

Table 5-3 HDMI SPDIF Sampling Frequency at Each Video Format

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
720x480p /720x576p	Yes	Yes	Yes	Yes	Yes	No	No
1440x480i/ 1440x576i	Yes	Yes	Yes	Yes	Yes	No	No

720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

3.Audio Sample Clock Capture and Regeneration

Audio data is carried across the HDMI link, which is driven by a TMDS clock running at a rate corresponding to the video pixel rate, does not retain the original audio sample clock. The task of recreating this clock at the Sink is called Audio Clock Regeneration.
The HDMI Transmitter determine the fractional relationship between the TMDS clock and an audio reference clock (128 audio sample rate [fs]) and pass the numerator and denominator of that fraction to the HDMI Sink across the HDMI link. The Sink then re-create the audio clock from the TMDS clock by using a clock divider and a clock multiplier. The exact relationship between the two clocks will be.

$128 \cdot f_s = f_{TMDS\_clock} * N / CTS$.

The scheme of the Audio Sample Clock Capture and Regeneration as shown below:

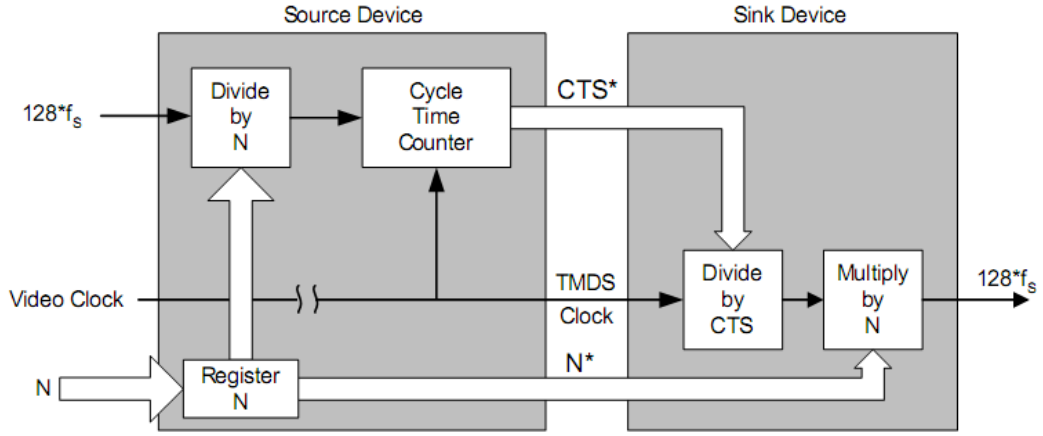


Fig. 5-4 HDMI Audio Clock Regeneration Model

Because there is no audio clock carried through the HDMI link, only the TMDS clock is used. Software sets the CTS/N with a value taken form the below table, which shows the CTS and N value for the supported standard. All other TMDS clocks are not supported; The TMDS clocks divided or multiplied by 1,001 coefficients are not supported.

Table 5-4 HDMI CTS and N table

	TMDS Clock (MHz)													
	25.2		27		54		74.25		148.5		297		597	
Fs (kHZ)	N	CTS	N	CTS	N	CTS	N	CTS	N	CTS	N	CTS	N	CTS
32	4096	25200	4096	27000	4096	54000	4096	74250	4096	148500	3072	222750	3072	445500
44.1	6272	28000	6272	30000	6272	60000	6272	82500	6272	165000	4704	247500	9408	990000
48	6144	25200	6144	27000	6144	54000	6144	74250	6144	148500	5120	247500	6144	495000
88.2	12544	28000	12544	30000	12544	60000	12544	82500	12544	165000	9408	247500	18816	990000
96	12288	25200	12288	27000	12288	54000	12288	74250	12288	148500	10240	247500	12288	495000
176.4	25088	28000	25088	30000	25088	60000	25088	82500	25088	165000	18816	247500	37632	990000
192	24576	25200	24576	27000	24576	54000	24576	74250	24576	148500	20480	247500	24576	4950000

5.3.4 DDC

The DDC functional block is used for configuration and status exchange between the HDMI Source and HDMI Sink. HMDI Transmitter Controller has I2C Master Interface for DDC

transactions. It enables for host controller to read EDID, HDCP authentication by issuing simple register access. The I2C bus speed is limited by DDC specification. DDC bus access frequency can be controlled.

5.3.5 EDID

Extended Display Identification Data (EDID) was created by VESA to enable plug and play capabilities of monitors. This data, which is stored in the sink device, describes video formats that the DTV Monitor is capable of receiving and rendering. The information is supplied to the source device, over the interface, upon the request of the source device. The source device then chooses its output format, taking into account the format of the original video stream and the formats supported by the DTV Monitor. The function of this module is to implement EDID feature.

5.3.6 HDCP

HDMI Transmitter has a capability for HDCP authentication by hardware. The function of this module is to implement HDCP encryption feature. This feature can be turned on or off depending on register setting.
RK3399 supports up to HDCP 2.2. HDCP 1.4 is done by HDMI Transmitter itself. HDCP22 is descripted in another chapter.

5.3.7 Hot Plug Detect

HDMI Transmitter has a capability for detecting the Sink plug in or plug out, and launch an interrupt and registers state indicating for software controlling.

5.3.8 TMDS encoder

The TMDS encoder converts the 2/4/8 bits data into the 10 bit DC-balanced TMDS data. HDMI TX put the TMDS encoding on the audio /video /aux data received from the HDCP XOR mask. This data is output onto three TMDS differential data lines along with a TMDS differential clock.

5.3.9 CEC

The CEC functional block provides high-level control functions between all of the various audiovisual products in a user's environment through one line.

5.4 HDMI PHY

The HDMI Tx PHY is the physical layer of an HDMI digital transmitter (source), capable of encoding and transmitting high-speed data streams carrying RGB video, audio, and control information.
The HDMI Tx PHY includes one PLL that synthesizes the high-speed serial bit clock (required by the transmitter) from a reference pixel clock with a frequency that can vary between 13.5-600 MHz for HDMI operation. The transmitter is capable of transmitting up to 18 Gbps using three lanes for HDMI operation.
The HDMI Tx PHY drives audio and video across the TMDS data channels. For HDMI operation, each serial TMDS link has a data-rate range of 0.25-6 Gbps. The HDMI Tx PHY also drives the TMDS clock at 1/10th of the serial data rate with a frequency range of 25-340 MHz for data rates below 3.4Gbps. For a serial data rate between 3.4Gbps and 6Gbps, the HDMI Tx PHY drives the TMDS clock at 1/40th of the serial data rate.
Within the HDMI Tx PHY, additional support blocks exist:

- Bandgap block (for blocks biasing)
- Resistor Calibration block
- ADC
- Analog test bus

An external, 1.62-kΩ reference resistor (connected to ground) is required for precise current and biasing (within each block).

5.4.1 I2C interface

The HDMI Tx PHY's status and configuration is accessed through internal I2C interface. The internal I2C interface is mapped to PHY Configuration Registers in HDMI TX registers at address 0x3020. Customers can access these registers though HDMI controller's Register.

Operating to these register can trigger one i2c write or i2c read.

5.4.2 PLL Operation

The PLL is responsible for the generation of the high-speed serialization clocks, as well as for the other clocks involved on the data transmission (clocks on the TMDS lanes) or involved on the supported testability.
The PLL can be configured for the video modes required for each of the supported modes of operation. For additional information about the PLL configuration, please refer "PHY MPLL Configuration".

5.4.3 HPD/RXSENSE Operation

For HDMI operation, to detect the presence of an HDMI sink system, the HDMI Tx PHY supports two different mechanisms to determine whether a Hot Plug Detect (HPD) signal is available and whether HDMI sink terminations are present and powered.
If voltage is present on the HPD line provided by the HDMI sink system, the HPD circuit identifies and asserts the SNKDET signal on the PHY's digital interface, provided the voltage level is 2.4-5.3 V, as specified by the HDMI specification. The HPD circuit enables the HDMI Tx PHY to identify whether the HDMI sink system is connected and is providing a correct HPD voltage level.
The RXSENSE circuit determines, for each TMDS data and clock lane for the HDMI operation, whether the HDMI sink terminations are present. Users can observe RXSENSE that reflects the state of the TMDS lanes evaluated from phy_stat0 in HDMI controller register.

5.4.4 Power Collapsing

The HDMI Tx PHY supports power collapsing. This feature enables the HDMI Tx PHY to stay—in any given combination of power supply availability (present, floating, or grounded)—in a known state with low current consumption. This low current consumption remains similar to, or below, the current consumption in power-down mode.
Power collapsing provides customers who require control over the power lines with independent control of the PHY power supplies.

5.4.5 Operating Modes

The HDMI Tx PHY can be placed in two different operating modes: Power-Down and Active. Customers can assert a soft reset through the HDMI control registers. This soft reset clears all system FSMs except I2C, JTAG, and control registers.
For each separate video mode in which the HDMI Tx PHY is set to transmit, due to different operating frequency, color depth and pixel repetition that characterizes each one, you need to configure the HDMI Tx PHY block for correct operation and optimized performance. It is recommended for the PHY configuration through the I2C or JTAG interface to be done while the PHY is in power-down mode.
Configuration involves programming the PLL dividers and analog configuration as well as the analog drivers' source termination value, signals voltage level, pre-emphasis and slope boosting. This programming is done through the I2C interface.

5.5 Register Description

The address offset of the HDMI TX is 0xff980000,it contains 16 address section. The offset of the table of Register Summary must multiple with 4 when software configure it。Like the Interrupt registers，its base address is 0x0100.If we want to configure it, its real address is 0xff980000+0x0100*4.

5.5.1 Register Summary

Name	Offset	Size	Reset Value	Description
Identification Registers	0x0000	B		Identification releated registers
Interrupt registers	0x0100	B		Interrupt releated registers

Name	Offset	Size	Reset Value	Description
Video Sampler registers	0x0200	B		Video Sampler registers
Video Packetizer registers	0x0800	B		Video Packetizer registers
Frame Composer Registers	0x1000	B		Frame Composer Registers
HDMI Source PHY Registers	0x3000	B		HDMI Source PHY Registers
I2C Master PHY Registers	0x3020	B		I2C Master PHY Registers
Audio Sampler Registers	0x3100	B		Audio Sampler Registers
Main Controller Registers	0x4000	B		Main Controller Registers
Color Space Converter Registers	0x4100	B		Color Space Converter Registers
HDCP Encryption Engine Registers	0x5000	B		HDCP Encryption Engine Registers
HDCP BKSV Registers	0x7800	B		HDCP BKSV Registers
HDCP AN Registers	0x7805	B		HDCP AN Registers
Encrypted DPK Embedded Storage Registers	0x780E	B		Encrypted DPK Embedded Storage Registers
CEC Engine Registers	0x7D00	B		CEC Engine Registers
I2C Master Registers	0x7E00	B		I2C Master Registers for E-DDC/SCDC

Identification Registers

Identification Registers. Follow the link for the register to see a detailed description of the register.

Registers for Address Block: Identification

Register	Offset	Description
design_id	0x0	Design Identification Register
revision_id	0x1	Revision Identification Register
product_id0	0x2	Product Identification Register 0
product_id1	0x3	Product Identification Register 1
config0_id	0x4	Configuration Identification Register 0
config1_id	0x5	Configuration Identification Register 1
config2_id	0x6	Configuration Identification Register 2
config3_id	0x7	Configuration Identification Register 3

design_id

Description: Design Identification Register

Size: 8 bits

Offset: 0x0

Bits	Name	Attr	Description
7:0	design_id	R	Design ID code fixed by HDMI that Identifies the instantiated DWC_hdmi_tx controller. For example, DWC_hdmi_tx 2.11a, DESIGN_ID = 21 Value After Reset: 0x21

revision_id

Description: Revision Identification Register

Size: 8 bits

Offset: 0x1

Bits	Name	Attr	Description
7:0	revision_id	R	Revision ID code fixed by HDMI that Identifies the instantiated DWC_hdmi_tx controller. Value After Reset: 0x1a

product_id0

Description: Product Identification Register 0

Size: 8 bits

Offset: 0x2

Bits	Name	Attr	Description
7:0	product_id0	R	This one byte fixed code Identifies HDMI 's product line ("A0h" for DWC_hdmi_tx products). Value After Reset: 0xa0

product_id1

Description: Product Identification Register 1

Size: 8 bits

Offset: 0x3

Bits	Name	Attr	Description
7:6	product_id1_hdcp	R	These bits identify a HDMI Controller with HDCP encryption according to HDMI product line. Value After Reset: "(HDCP== 1) ? 3 : 0"
5:2			Reserved for future use.
1	product_id1_rx	R	This bit Identifies HDMI 's DWC_hdmi_rx Controller according to HDMI product line. Value After Reset: 0x0
0	product_id1_tx	R	This bit Identifies H Controller according to HDMI product line. Value After Reset: 0x1

config0_id

Description: Configuration Identification Register 0

Size: 8 bits

Offset: 0x4

Bits	Name	Attr	Description
7	prepen	R	Indicates if it is possible to use internal pixel

			repetition Value After Reset: "(HDMI_TX_INTPREPEN== 1) ? 1 : 0"
6			Reserved for future use.
5	audspdif	R	Indicates if the SPDIF audio interface is present Value After Reset: "(SPDIFPORTS== 1) ? 1 : 0"
4	audi2s	R	Indicates if I2S interface is present Value After Reset: "(I2SPORTS== 1) ? 1 : 0"
3	hdmi14	R	Indicates if HDMI 1.4 features are present Value After Reset: "(HDMI_TX_14== 1) ? 1 : 0"
2	csc	R	Indicates if Color Space Conversion block is present Value After Reset: "(CSC== 1) ? 1 : 0"
1	cec	R	Indicates if CEC is present Value After Reset: "(CEC== 1) ? 1 : 0"
0	hdcp	R	Indicates if HDCP is present Value After Reset: "(HDCP== 1) ? 1 : 0"

config1_id

Description: Configuration Identification Register 1

Size: 8 bits

Offset: 0x5

Bits	Name	Attr	Description
7	hdcp22_snps	R	Indicates if HDCP 2.2 SNPS solution is present Value After Reset: (HTX_HDCP22_SNPS== 1) ? 1 : 0
6	hdcp22_ext	R	Indicates if external HDCP 2.2 interface support is present Value After Reset: "(HTX_HDCP22_EXTERNAL== 1) ? 1 : 0"
5	hdmi20	R	Indicates if HDMI 2.0 features are present Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0"
4:2			Reserved for future use.
1	confapb	R	Indicates that configuration interface is APB interface Value After Reset: 0x1
0			Reserved for future use.

config2_id

Description: Configuration Identification Register 2

Size: 8 bits

Offset: 0x6

Bits	Name	Attr	Description
7:0	phytype	R	Indicates the type of PHY interface selected: 0x00: Legacy PHY (HDMI Tx PHY) 0xF2: PHY GEN2 (HDMI 3D TX PHY) 0xE2: PHY GEN2 (HDMI 3D TX PHY) + HEAC PHY 0xC2: PHY MHL COMBO (MHL+HDMI 2.0 TX PHY) 0xB2: PHY MHL COMBO (MHL+HDMI 2.0 TX PHY) +

			HEAC PHY 0xF3: PHY HDMI 20 (HDMI 2.0 TX PHY) 0xE3: PHY HDMI 20 (HDMI 2.0 TX PHY) + HEAC PHY 0xFE: External PHY Value After Reset: "(PHY_HDMI20==1) ? ((HDMI_HEAC_PHY_EN==1)? 0xE3 : 0xF3) : (PHY_MHL_COMBO==1) ? ((HDMI_HEAC_PHY_EN==1)? 0xB2 : 0xC2) : (PHY_GEN2==1) ? ((HDMI_HEAC_PHY_EN==1)? 0xE2 : 0xF2) : (PHY_EXTERNAL==1)? 0xFE : 0x00"

config3_id

Description: Configuration Identification Register 3

Size: 8 bits

Offset: 0x7

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	confahbauddma	R	Indicates that the audio interface is AHB AUD DMA Value After Reset: "(AHBAUDDMAIF== 1) ? 1 : 0"
0	confgpaud	R	Indicates that the audio interface is Generic Parallel Audio (GPAUD) Value After Reset: "(GPAUDPORTS== 1) ? 1 : 0"

Interrupt Registers

Interrupt Registers. Follow the link for the register to see a detailed description of the register.

Registers for Address Block: Interrupt

Register	Offset	Description
ih_fc_stat0	0x100	Frame Composer Interrupt Status Register 0 (Packet Interrupts)
ih_fc_stat1	0x101	Frame Composer Interrupt Status Register 1 (Packet Interrupts)
ih_fc_stat2	0x102	Frame Composer Interrupt Status Register 2 (Packet Interrupts)
ih_as_stat0	0x103	Audio Sampler Interrupt Status Register (FIFO Threshold, Underflow and Overflow Interrupts)
ih_phy_stat0	0x104	PHY Interface Interrupt Status Register (RXSENSE, PLL Lock and HPD Interrupts)
ih_i2cm_stat0	0x105	E-DDC I2C Master Interrupt Status Register (Done and Error Interrupts)
ih_cec_stat0	0x106	CEC Interrupt Status Register (Functional Operation Interrupts)
ih_vp_stat0	0x107	Video Packetizer Interrupt Status Register (FIFO Full and Empty Interrupts)
ih_i2cmphy_stat0	0x108	PHY GEN2 I2C Master Interrupt Status Register

Register	Offset	Description
		(Done and Error Interrupts)
ih_ahbdmaaud_stat0	0x109	AHB Audio DMA Interrupt Status Register (Functional Operation, Buffer Full and Empty...
ih_decode	0x170	Interruption Handler Decode Assist Register
ih_mute_fc_stat0	0x180	Frame Composer Interrupt Mute Control Register 0
ih_mute_fc_stat1	0x181	Frame Composer Interrupt Mute Control Register 1
ih_mute_fc_stat2	0x182	Frame Composer Interrupt Mute Control Register 2
ih_mute_as_stat0	0x183	Audio Sampler Interrupt Mute Control Register
ih_mute_phy_stat0	0x184	PHY Interface Interrupt Mute Control Register
ih_mute_i2cm_stat0	0x185	E-DDC I2C Master Interrupt Mute Control Register
ih_mute_cec_stat0	0x186	CEC Interrupt Mute Control Register
ih_mute_vp_stat0	0x187	Video Packetizer Interrupt Mute Control Register
ih_mute_i2cmphy_stat0	0x188	PHY GEN2 I2C Master Interrupt Mute Control Register
ih_mute_ahbdmaaud_stat0	0x189	AHB Audio DMA Interrupt Mute Control Register
ih_mute	0x1ff	Global Interrupt Mute Control Register

ih_fc_stat0

Description: Frame Composer Interrupt Status Register 0 (Packet Interrupts)

Size: 8 bits

Offset: 0x100

Bits	Name	Attr	Description
7	AUDI	R/W1C	Active after successful transmission of an Audio InfoFrame packet. Value After Reset: 0x0
6	ACP	R/W1C	Active after successful transmission of an Audio Content Protection packet. Value After Reset: 0x0
5	HBR	R/W1C	Active after successful transmission of an Audio HBR packet. Value After Reset: 0x0
4	MAS	R/W1C	Active after successful transmission of an MultiStream Audio packet Value After Reset: 0x0
3	NVBI	R/W1C	Active after successful transmission of an NTSC VBI packet Value After Reset: 0x0
2	AUDS	R/W1C	Active after successful transmission of an Audio Sample packet. Due to high number of audio sample packets transmitted, this interrupt is by default masked at frame composer. Value After Reset: 0x0
1	ACR	R/W1C	Active after successful transmission of an Audio Clock Regeneration (N/ CTS transmission) packet. Value After Reset: 0x0

0	NULL	R/W1C	Active after successful transmission of an Null packet. Due to high number of audio sample packets transmitted, this interrupt is by default masked at frame composer. Value After Reset: 0x0

ih_fc_stat1

Description: Frame Composer Interrupt Status Register 1 (Packet Interrupts)

Size: 8 bits

Offset: 0x101

Bits	Name	Attr	Description
7	GMD	R/W1C	Active after successful transmission of an Gamut metadata packet. Value After Reset: 0x0
6	ISCR1	R/W1C	Active after successful transmission of an International Standard Recording Code 1 packet. Value After Reset: 0x0
5	ISCR2	R/W1C	Active after successful transmission of an International Standard Recording Code 2 packet Value After Reset: 0x0
4	VSD	R/W1C	Active after successful transmission of an Vendor Specific Data InfoFrame packet. Value After Reset: 0x0
3	SPD	R/W1C	Active after successful transmission of an Source Product Descriptor InfoFrame packet. Value After Reset: 0x0
2	AMP	R/W1C	Active after successful transmission of an Audio Metadata packet Value After Reset: 0x0
1	AVI	R/W1C	Active after successful transmission of an AVI InfoFrame packet. Value After Reset: 0x0
0	GCP	R/W1C	Active after successful transmission of an General Control Packet. Value After Reset: 0x0

ih_fc_stat2

Description: Frame Composer Interrupt Status Register 2 (Packet Interrupts)

Size: 8 bits

Offset: 0x102

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	DRM	R/W1C	Active after successful transmission of an DRM packet Value After Reset: 0x0
1	LowPriority_overflow	R/W1C	Frame Composer low priority packet queue

			descriptor overflow indication Value After Reset: 0x0
0	HighPriority_overflow	R/W1C	Frame Composer high priority packet queue descriptor overflow indication Value After Reset: 0x0

ih_as_stat0

Description: Audio Sampler Interrupt Status Register (FIFO Threshold, Underflow and Overflow Interrupts)

Size: 8 bits

Offset: 0x103

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	fifo_underrun	R/W1C	Indicates an underrun on the audio FIFO Value After Reset: 0x0
3	fifo_overrun	R/W1C	Indicates an overrun on the audio FIFO. Value After Reset: 0x0
2	Aud_fifo_underflow_thr	R/W1C	Audio Sampler audio FIFO empty threshold (four samples) indication for the legacy HBR audio interface. For AHB_DMA, this bit indicates that the number of samples in the FIFO is equal to (or less) than the number of active audio channels. This bit is not relevant for I2S, SPDIF, and GPA interfaces. Value After Reset: 0x0
1	Aud_fifo_underflow	R/W1C	Audio Sampler audio FIFO empty indication. Value After Reset: 0x0
0	Aud_fifo_overflow	R/W1C	Audio Sampler audio FIFO full indication. Value After Reset: 0x0

ih_phy_stat0

Description: PHY Interface Interrupt Status Register (RXSENSE, PLL Lock and HPD Interrupts)

Size: 8 bits

Offset: 0x104

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	RX_SENSE_3	R/W1C	TX PHY RX_SENSE indication for driver 3. You may need to mask or change polarity of this interrupt after it has become active. Value After Reset: 0x0
4	RX_SENSE_2	R/W1C	TX PHY RX_SENSE indication for driver 2. You may need to mask or change polarity of this interrupt after it has become active. Value After Reset: 0x0

3	RX_SENSE_1	R/W1C	TX PHY RX_SENSE indication for driver 1. You may need to mask or change polarity of this interrupt after it has become active. Value After Reset: 0x0
2	RX_SENSE_0	R/W1C	TX PHY RX_SENSE indication for driver 0. You may need to mask or change polarity of this interrupt after it has become active. Value After Reset: 0x0
1	TX_PHY_LOCK	R/W1C	TX PHY PLL lock indication. Value After Reset: 0x0
0	HPD	R/W1C	HDMI Hot Plug Detect indication. You may need to mask or change polarity of this interrupt after it has become active. Value After Reset: 0x0

ih_i2cm_stat0

Description: E-DDC I2C Master Interrupt Status Register (Done and Error Interrupts)

Size: 8 bits

Offset: 0x105

Bits	Name	Attr	Description
7:3			Reserved for future use.
2	scdc_readreq	R/W1C	I2C Master SCDC read request indication. Value After Reset: 0x0
1	I2Cmasterdone	R/W1C	I2C Master done indication Value After Reset: 0x0
0	I2Cmastererror	R/W1C	I2C Master error indication Value After Reset: 0x0

ih_cec_stat0

Description: CEC Interrupt Status Register (Functional Operation Interrupts)

Size: 8 bits

Offset: 0x106

Bits	Name	Attr	Description
7			Reserved for future use.
6	WAKEUP	R/W1C	CEC Wake-up indication Value After Reset: 0x0
5	ERROR_FOLLOW	R/W1C	CEC Error Follow indication Value After Reset: 0x0
4	ERROR_INITIATOR	R/W1C	CEC Error Initiator indication Value After Reset: 0x0
3	ARB_LOST	R/W1C	CEC Arbitration Lost indication Value After Reset: 0x0
2	NACK	R/W1C	CEC Not Acknowledge indication Value After Reset: 0x0
1	EOM	R/W1C	CEC End of Message Indication Value After Reset: 0x0
0	DONE	R/W1C	CEC Done Indication Value After Reset: 0x0

ih_vp_stat0

Description: Video Packetizer Interrupt Status Register (FIFO Full and Empty Interrupts)

Size: 8 bits

Offset: 0x107

Bits	Name	Attr	Description
7	fifofullrepet	R/W1C	Video Packetizer pixel repeater FIFO full interrupt Value After Reset: 0x0
6	fifoemptyrepet	R/W1C	Video Packetizer pixel repeater FIFO empty interrupt Value After Reset: 0x0
5	fifofullpp	R/W1C	Video Packetizer pixel packing FIFO full interrupt Value After Reset: 0x0
4	fifoemptypp	R/W1C	Video Packetizer pixel packing FIFO empty interrupt Value After Reset: 0x0
3	fifofullremap	R/W1C	Video Packetizer pixel YCC 422 re-mapper FIFO full interrupt Value After Reset: 0x0
2	fifoemptyremap	R/W1C	Video Packetizer pixel YCC 422 re-mapper FIFO empty interrupt Value After Reset: 0x0
1:0		R/W1C	Reserved and read as zero

ih_i2cmphy_stat0

Description: PHY GEN2 I2C Master Interrupt Status Register (Done and Error Interrupts)

Size: 8 bits

Offset: 0x108

Bits	**Name**	**Attr**	**Description**
7:2			Reserved for future use.
1	I2Cmphydone	R/W1C	I2C Master PHY done indication Value After Reset: 0x0
0	I2Cmphyerror	R/W1C	I2C Master PHY error indication Value After Reset: 0x0

ih_ahbdmaaud_stat0

Description: AHB Audio DMA Interrupt Status Register (Functional Operation, Buffer Full and Empty Interrupts)

Size: 8 bits

Offset: 0x109

Bits	**Name**	**Attr**	**Description**
7			Reserved for future use.
6	ahbdmaaud_intbuffoverrun	R/W1C	AHB audio DMA Buffer overrun interruption Value After Reset: 0x0
5	ahbdmaaud_interror	R/W1C	AHB audio DMA error interrupt Value After Reset: 0x0
4	ahbdmaaud_intlostownership	R/W1C	AHB audio DMA lost ownership interrupt Value After Reset: 0x0
3	ahbdmaaud_intretrysplit	R/W1C	AHB audio DMA RETRY/SPLIT interrupt

			Value After Reset: 0x0
2	ahbdmaaud_intdone	R/W1C	AHB audio DMA done interrupt Value After Reset: 0x0
1	ahbdmaaud_intbufffull	R/W1C	AHB audio DMA Buffer full interrupt Value After Reset: 0x0
0	ahbdmaaud_intbuffempty	R/W1C	AHB audio DMA Buffer empty interrupt Value After Reset: 0x0

ih_decode

Description: Interruption Handler Decode Assist Register

Size: 8 bits

Offset: 0x170

Bits	Name	Attr	Description
7	ih_fc_stat0	R	Interruption active at the ih_fc_stat0 register Value After Reset: 0x0
6	ih_fc_stat1	R	Interruption active at the ih_fc_stat1 register Value After Reset: 0x0
5	ih_fc_stat2_vp	R	Interruption active at the ih_fc_stat2 or ih_vp_stat0 register Value After Reset: 0x0
4	ih_as_stat0	R	Interruption active at the ih_as_stat0 register Value After Reset: 0x0
3	ih_phy	R	Interruption active at the ih_phy_stat0 or ih_i2cmphy_stat0 register Value After Reset: 0x0
2	ih_i2cm_stat0	R	Interruption active at the ih_i2cm_stat0 register Value After Reset: 0x0
1	ih_cec_stat0	R	Interruption active at the ih_cec_stat0 register Value After Reset: 0x0
0	ih_ahbdmaaud_stat0	R	Interruption active at the ih_ahbdmaaud_stat0 register Value After Reset: 0x0

ih_mute_fc_stat0

Description: Frame Composer Interrupt Mute Control Register 0

Size: 8 bits

Offset: 0x180

Bits	Name	Attr	Description
7	AUDI	R/W	When set to 1, mutes ih_fc_stat0[7] Value After Reset: 0x0
6	ACP	R/W	When set to 1, mutes ih_fc_stat0[6] Value After Reset: 0x0
5	HBR	R/W	When set to 1, mutes ih_fc_stat0[5] Value After Reset: 0x0
4	MAS	R/W	When set to 1, mutes ih_fc_stat0[4]. Otherwise, this field is a "spare" bit with no associated

Bits	Name	Attr	Description
			functionality. Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0"
3	NVBI	R/W	When set to 1, mutes ih_fc_stat0[3]. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0"
2	AUDS	R/W	When set to 1, mutes ih_fc_stat0[2] Value After Reset: 0x0
1	ACR	R/W	When set to 1, mutes ih_fc_stat0[1] Value After Reset: 0x0
0	NULL	R/W	When set to 1, mutes ih_fc_stat0[0] Value After Reset: 0x0

ih_mute_fc_stat1

Description: Frame Composer Interrupt Mute Control Register 1

Size: 8 bits

Offset: 0x181

Bits	Name	Attr	Description
7	GMD	R/W	When set to 1, mutes ih_fc_stat1[7] Value After Reset: 0x0
6	ISCR1	R/W	When set to 1, mutes ih_fc_stat1[6] Value After Reset: 0x0
5	ISCR2	R/W	When set to 1, mutes ih_fc_stat1[5] Value After Reset: 0x0
4	VSD	R/W	When set to 1, mutes ih_fc_stat1[4] Value After Reset: 0x0
3	SPD	R/W	When set to 1, mutes ih_fc_stat1[3] Value After Reset: 0x0
2	AMP	R/W	When set to 1, mutes ih_fc_stat1[2]. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0" Exists: HDMI_TX_20==1
1	AVI	R/W	When set to 1, mutes ih_fc_stat1[1] Value After Reset: 0x0
0	GCP	R/W	When set to 1, mutes ih_fc_stat1[0] Value After Reset: 0x0

ih_mute_fc_stat2

Description: Frame Composer Interrupt Mute Control Register 2

Size: 8 bits

Offset: 0x182

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	DRM	R/W	When set to 1, mutes ih_fc_stat2[4].

			Value After Reset: (HDMI_TX_20== 1) ? 1 : 0
3:2			Reserved for future use
1	LowPriority_overflow	R/W	When set to 1, mutes ih_fc_stat2[1] Value After Reset: 0x0
0	HighPriority_overflow	R/W	When set to 1, mutes ih_fc_stat2[0] Value After Reset: 0x0

ih_mute_as_stat0

Description: Audio Sampler Interrupt Mute Control Register

Size: 8 bits

Offset: 0x183

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	fifo_underrun	R/W	When set to 1, mutes ih_as_stat0[4] Value After Reset: 0x1
3	fifo_overrun	R/W	When set to 1, mutes ih_as_stat0[3] Value After Reset: 0x1
2	Aud_fifo_underflow_thr	R/W	When set to 1, mutes ih_as_stat0[2] Value After Reset: 0x0
1	Aud_fifo_underflow	R/W	When set to 1, mutes ih_as_stat0[1] Value After Reset: 0x0
0	Aud_fifo_overflow	R/W	When set to 1, mutes ih_as_stat0[0] Value After Reset: 0x0

ih_mute_phy_stat0

Description: PHY Interface Interrupt Mute Control Register

Size: 8 bits

Offset: 0x184

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	RX_SENSE_3	R/W	When set to 1, mutes ih_phy_stat0[5] Value After Reset: 0x0
4	RX_SENSE_2	R/W	When set to 1, mutes ih_phy_stat0[4] Value After Reset: 0x0
3	RX_SENSE_1	R/W	When set to 1, mutes ih_phy_stat0[3] Value After Reset: 0x0
2	RX_SENSE_0	R/W	When set to 1, mutes ih_phy_stat0[2] Value After Reset: 0x0
1	TX_PHY_LOCK	R/W	When set to 1, mutes ih_phy_stat0[1] Value After Reset: 0x0
0	HPD	R/W	When set to 1, mutes ih_phy_stat0[0] Value After Reset: 0x0

ih_mute_i2cm_stat0

Description: E-DDC I2C Master Interrupt Mute Control Register

Size: 8 bits

Offset: 0x185

Bits	Name	Attr	Description
7:3			Reserved for future use.
2	scdc_readreq	R/W	When set to 1, mutes ih_i2cm_stat0[2] Value After Reset: 0x1
1	I2Cmasterdone	R/W	When set to 1, mutes ih_i2cm_stat0[1] Value After Reset: 0x0
0	I2Cmastererror	R/W	When set to 1, mutes ih_i2cm_stat0[0] Value After Reset: 0x0

ih_mute_cec_stat0

Description: CEC Interrupt Mute Control Register

Size: 8 bits

Offset: 0x186

Bits	Name	Attr	Description
7			Reserved for future use.
6	WAKEUP	R/W	When set to 1, mutes ih_cec_stat0[6] Value After Reset: 0x0
5	ERROR_FOLLOW	R/W	When set to 1, mutes ih_cec_stat0[5] Value After Reset: 0x0
4	ERROR_INITIATOR	R/W	When set to 1, mutes ih_cec_stat0[4] Value After Reset: 0x0
3	ARB_LOST	R/W	When set to 1, mutes ih_cec_stat0[3] Value After Reset: 0x0
2	NACK	R/W	When set to 1, mutes ih_cec_stat0[2] Value After Reset: 0x0
1	EOM	R/W	When set to 1, mutes ih_cec_stat0[1] Value After Reset: 0x0
0	DONE	R/W	When set to 1, mutes ih_cec_stat0[0] Value After Reset: 0x0

ih_mute_vp_stat0

Description: Video Packetizer Interrupt Mute Control Register

Size: 8 bits

Offset: 0x187

Bits	Name	Attr	Description
7	fifofullrepet	R/W	When set to 1, mutes ih_vp_stat0[7] Value After Reset: 0x0
6	fifoemptyrepet	R/W	When set to 1, mutes ih_vp_stat0[6] Value After Reset: 0x0
5	fifofullpp	R/W	When set to 1, mutes ih_vp_stat0[5] Value After Reset: 0x0
4	fifoemptypp	R/W	When set to 1, mutes ih_vp_stat0[4] Value After Reset: 0x0
3	fifofullremap	R/W	When set to 1, mutes ih_vp_stat0[3] Value After Reset: 0x0

2	fifoemptyremap	R/W	When set to 1, mutes ih_vp_stat0[2] Value After Reset: 0x0
1	spare_2	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
0	spare_1	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0

ih_mute_i2cmphy_stat0

Description: PHY GEN2 I2C Master Interrupt Mute Control Register

Size: 8 bits

Offset: 0x188

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	I2Cmphydone	R/W	When set to 1, mutes ih_i2cmphy_stat0[1] Value After Reset: 0x0
0	I2Cmphyerror	R/W	When set to 1, mutes ih_i2cmphy_stat0[0] Value After Reset: 0x0

ih_mute_ahbdmaaud_stat0

Description: AHB Audio DMA Interrupt Mute Control Register

Size: 8 bits

Offset: 0x189

Bits	Name	Attr	Description
7			Reserved for future use.
6	ahbdmaaud_intbuffoverrun	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[6] Value After Reset: 0x1
5	ahbdmaaud_interror	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[5] Value After Reset: 0x0
4	ahbdmaaud_intlostownership	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[4] Value After Reset: 0x0
3	ahbdmaaud_intretrysplit	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[3] Value After Reset: 0x0
2	ahbdmaaud_intdone	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[2] Value After Reset: 0x0
1	ahbdmaaud_intbufffull	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[1] Value After Reset: 0x0
0	ahbdmaaud_intbuffempty	R/W	When set to 1, mutes ih_ahbdmaaud_stat0[0] Value After Reset: 0x0

ih_mute

Description: Global Interrupt Mute Control Register

Size: 8 bits

Offset: 0x1ff

Bits	Name	Attr	Description
7:2			Reserved for future use.

1	mute_wakeup_interrupt	R/W	When set to 1, mutes the main interrupt output port. The sticky bit interrupts continue with their state accessible through the configuration bus, only the main interrupt line is muted. Value After Reset: 0x1
0	mute_all_interrupt	R/W	When set to 1, mutes the main interrupt line (where all interrupts are ORed). The sticky bit interrupts continue with their state; only the main interrupt line is muted. Value After Reset: 0x1

VideoSampler Registers

Video Sampler Registers. Follow the link for the register to see a detailed description of the register.

Registers for Address Block: VideoSampler

Register	Offset	Description
tx_invid0	0x200	Video Input Mapping and Internal Data Enable Configuration Register
tx_instuffing	0x201	Video Input Stuffing Enable Register
tx_gydata0	0x202	Video Input gy Data Channel Stuffing Register 0
tx_gydata1	0x203	Video Input gy Data Channel Stuffing Register 1
tx_rcrdata0	0x204	Video Input rcr Data Channel Stuffing Register 0
tx_rcrdata1	0x205	Video Input rcr Data Channel Stuffing Register 1
tx_bcbdata0	0x206	Video Input bcb Data Channel Stuffing Register 0
tx_bcbdata1	0x207	Video Input bcb Data Channel Stuffing Register 1

tx_invid0

Description: Video Input Mapping and Internal Data Enable Configuration Register

Size: 8 bits

Offset: 0x200

Bits	Name	Attr	Description
7	internal_de_generator	R/W	Internal data enable (DE) generator enable. If data enable is not available for the input video, set this bit to one to activate the internal data enable generator. Attention: This feature only works for input video modes that have native repetition (such as, all CEA videos). No desired pixel repetition can be used with this feature because these configurations only affect the Frame Composer and not this block. The DE Generator does not work for the following conditions: Transmission of video with CEA VIC 39 Transmission of 3D video using the field alternative structure Value After Reset: 0x0

Bits	Name	Attr	Description
6:5			Reserved for future use.
4:0	video_mapping	R/W	Video Input mapping (color space/color depth): 0x01: RGB 4:4:4/8 bits 0x03: RGB 4:4:4/10 bits 0x05: RGB 4:4:4/12 bits 0x07: RGB 4:4:4/16 bits 0x09: YCbCr 4:4:4 or 4:2:0/8 bits 0x0B: YCbCr 4:4:4 or 4:2:0/10 bits 0x0D: YCbCr 4:4:4 or 4:2:0/12 bits 0x0F: YCbCr 4:4:4 or 4:2:0/16 bits 0x16: YCbCr 4:2:2/8 bits 0x14: YCbCr 4:2:2/10 bits 0x12: YCbCr 4:2:2/12 bits 0x17: YCbCr 4:4:4 (IPI)/8 bits 0x18: YCbCr 4:4:4 (IPI)/10 bits 0x19: YCbCr 4:4:4 (IPI)/12 bits 0x1A: YCbCr 4:4:4 (IPI)/16 bits 0x1B: YCbCr 4:2:2 (IPI)/12 bits 0x1C: YCbCr 4:2:0 (IPI)/8 bits 0x1D: YCbCr 4:2:0 (IPI)/10 bits 0x1E: YCbCr 4:2:0 (IPI)/12 bits 0x1F: YCbCr 4:2:0 (IPI)/16 bits Value After Reset: 0x1

tx_instuffing

Description: Video Input Stuffing Enable Register

Size: 8 bits

Offset: 0x201

Bits	Name	Attr	Description
7:3			Reserved for future use.
2	bcbdata_stuffing	R/W	0b: When the dataen signal is low, the value in the bcbdata[15:0] output is the one sampled from the corresponding input data. 1b: When the dataen signal is low, the value in the bcbdata[15:0] output is given by the values in the TX_BCBDTA0 and TX_BCBDATA1 registers. Value After Reset: 0x0
1	rcrdata_stuffing	R/W	0b: When the dataen signal is low, the value in the rcrdata[15:0] output is the one sampled from the corresponding input data. 1b: When the dataen signal is low, the value in the rcrdata[15:0] output is given by the values in TX_RCRDTA0 and TX_RCRDATA1 registers. Value After Reset: 0x0
0	gydata_stuffing	R/W	0b: When the dataen signal is low, the value in the gydata[15:0] output is the one sampled from the corresponding input data. 1b: When the dataen signal is low, the value in the

			gydata[15:0] output is given by the values in TX_GYDTA0 and TX_GYDATA1 registers. Value After Reset: 0x0

tx_gydata0

Description: Video Input gy Data Channel Stuffing Register 0

Size: 8 bits

Offset: 0x202

Bits	Name	Attr	Description
7:0	gydata	R/W	This register defines the value of gydata[7:0] when TX_INSTUFFING[0] (gydata_stuffing) is set to 1b. Value After Reset: 0x0

tx_gydata1

Description: Video Input gy Data Channel Stuffing Register 1

Size: 8 bits

Offset: 0x203

Bits	Name	Attr	Description
7:0	gydata	R/W	This register defines the value of gydata[15:8] when TX_INSTUFFING[0] (gydata_stuffing) is set to 1b. Value After Reset: 0x0

tx_rcrdata0

Description: Video Input rcr Data Channel Stuffing Register 0

Size: 8 bits

Offset: 0x204

Bits	Name	Attr	Description
7:0	rcrdata	R/W	This register defines the value of rcrydata[7:0] when TX_INSTUFFING[1] (rcrdata_stuffing) is set to 1b. Value After Reset: 0x0

tx_rcrdata1

Description: Video Input rcr Data Channel Stuffing Register 1

Size: 8 bits

Offset: 0x205

Bits	Name	Attr	Description
7:0	rcrdata	R/W	This register defines the value of rcrydata[15:8] when TX_INSTUFFING[1] (rcrdata_stuffing) is set to 1b. Value After Reset: 0x0

tx_bcbdata0

Description: Video Input bcb Data Channel Stuffing Register 0

Size: 8 bits

Offset: 0x206

Bits	Name	Attr	Description
7:0	bcbdata	R/W	This register defines the value of bcbdata[7:0] when TX_INSTUFFING[2] (bcbdata_stuffing) is set to 1b. Value After Reset: 0x0

tx_bcbdata1

Description: Video Input bcb Data Channel Stuffing Register 1

Size: 8 bits

Offset: 0x207

Bits	Name	Attr	Description
7:0	bcbdata	R/W	This register defines the value of bcbdata[15:8] when TX_INSTUFFING[2] (bcbdata_stuffing) is set to 1b. Value After Reset: 0x0

VideoPacketizer Registers

Video Packetizer Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
vp_status	0x800	Video Packetizer Packing Phase Status Register
vp_pr_cd	0x801	Video Packetizer Pixel Repetition and Color Depth Register
vp_stuff	0x802	Video Packetizer Stuffing and Default Packing Phase Register
vp_remap	0x803	Video Packetizer YCC422 Remapping Register
vp_conf	0x804	Video Packetizer Output and Enable Configuration Register
vp_mask	0x807	Video Packetizer Interrupt Mask Register

vp_status

Description: Video Packetizer Packing Phase Status Register

Size: 8 bits

Offset: 0x800

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	packing_phase	R	Read only register that holds the "packing phase" output of the Video Packetizer block. Value After Reset: 0x0

vp_pr_cd

Description: Video Packetizer Pixel Repetition and Color Depth Register

Size: 8 bits

Offset: 0x801

Bits	Name	Attr	Description
7:4	color_depth	R/W	The Color depth configuration is described as the following, with the action stated corresponding to color_depth[3:0]: 0000b: 24 bits per pixel video (8 bits per component). 8-bit packing mode. 0001b-0011b: Reserved. Not used.

Bits	Name	Attr	Description
			0100b: 24 bits per pixel video (8 bits per component). 8-bit packing mode. 0101b: 30 bits per pixel video (10 bits per component). 10-bit packing mode. 0110b: 36 bits per pixel video (12 bits per component). 12-bit packing mode. 0111b: 48 bits per pixel video (16 bits per component). 16-bit packing mode. Other: Reserved. Not used. Value After Reset: 0x0
3:0	desired_pr_factor	R/W	Desired pixel repetition factor configuration. The configured value sets H13T PHY PLL to multiply pixel clock by the factor in order to obtain the desired repetition clock. For the CEA modes some are already defined with pixel repetition in the input video. So for CEA modes this shall be always 0. Shall only be used if the user wants to do pixel repetition using H13TCTRL controller. The action is stated corresponding to desired_pr_factor[3:0]: 0000b: No pixel repetition (pixel sent only once) 0001b: Pixel sent two times (pixel repeated once) 0010b: Pixel sent three times 0011b: Pixel sent four times 0100b: Pixel sent five times 0101b: Pixel sent six times 0110b: Pixel sent seven times 0111b: Pixel sent eight times 1000b: Pixel sent nine times 1001b: Pixel sent 10 times Other: Reserved. Not used Value After Reset: 0x0

vp_stuff

Description: Video Packetizer Stuffing and Default Packing Phase Register

Size: 8 bits

Offset: 0x802

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	idefault_phase	R/W	Controls the default phase packing machine used according to HDMI 1.4b specification: "If the transmitted video format has timing such that the phase of the first pixel of every Video Data Period corresponds to pixel packing phase 0 (e.g. 10P0, 12P0, 16P0), the Source may set the Default_Phase bit in the GCP. The Sink may use this bit to optimize its filtering or handling of the

Bits	Name	Attr	Description
			PP field." This means that for 10-bit mode the Htotal must be dividable by 4; for 12- bit mode, the Htotal must be divisible by 2. Value After Reset: 0x0
4	ifix_pp_to_last	R/W	Reserved. Controls packing machine strategy Value After Reset: 0x0
3	icx_goto_p0_st	R/W	Reserved. Controls packing machine strategy Value After Reset: 0x0
2	ycc422_stuffing	R/W	YCC 422 remap stuffing control. For horizontal blanking, the action is stated corresponding to ycc422_stuffing: 0b: YCC 422 remap block in direct mode (input blanking data goes directly to output). 1b: YCC 422 remap block in stuffing mode. When "de" goes to low the outputs are fixed to 0x00. Value After Reset: 0x0
1	pp_stuffing	R/W	Pixel packing stuffing control. The action is stated corresponding to pp_stuffing: 0b: Pixel packing block in direct mode (input blanking data goes directly to output). 1b: Pixel packing block in stuffing mode. When "de_rep" goes to low the outputs are fixed to 0x00. Value After Reset: 0x0

Fields for Register: vp_stuff (Continued)

Bits	Name	Attr	Description
0	pr_stuffing	R/W	Pixel repeater stuffing control. The action is stated corresponding to pp_stuffing: 0b: Pixel repeater block in direct mode (input blanking data goes directly to output). 1b: Pixel repeater block in stuffing mode. When "de" goes to low the outputs are fixed to 0x00. Value After Reset: 0x0

vp_remap

Description: Video Packetizer YCC422 Remapping Register

Size: 8 bits

Offset: 0x803

Bits	Name	Attr	Description
7:2			Reserved for future use.
1:0	ycc422_size	R/W	YCC 422 remap input video size ycc422_size[1:0] 00b: YCC 422 16-bit input video (8 bits per component) 01b: YCC 422 20-bit input video (10

			bits per component) 10b: YCC 422 24-bit input video (12 bits per component) 11b: Reserved. Not used Value After Reset: 0x0

vp_conf

Description: Video Packetizer Output and Enable Configuration Register

Size: 8 bits

Offset: 0x804

Bits	Name	Attr	Description
7			Reserved for future use.
6	bypass_en	R/W	When set to 1'b1, Pixel packing enable. When set to 1b'0, the pixel packing block is controlled by pp_en. Value After Reset: 0x0
5	pp_en	R/W	Pixel packing enable. When set to 0, the pixel packing block is disabled if bypass_en is 1’b0. Value After Reset: 0x1
4	pr_en	R/W	Pixel repeater enable. When set to 0, the pixel repetition block is disabled. Value After Reset: 0x0
3	ycc422_en	R/W	YCC 422 select enable. Disabling forces bypass module to output always zeros. Value After Reset: 0x0
2	bypass_select	R/W	bypass_select 0b: Data from pixel repeater block 1b: Data from input of Video Packetizer block Value After Reset: 0x1
1	output_selector	R/W	When set to 1'b1, Data from pixel packing block. Value After Reset: 0x0
0	output_selector_0	R/W	Video Packetizer output selection 0b: Data from pixel packing block 1b: Data from YCC422 remap block Value After Reset: 0x0

vp_mask

Description: Video Packetizer Interrupt Mask Register

Size: 8 bits

Offset: 0x807

Bits	Name	Attr	Description
7	ointfullrepet	R/W	Mask bit for Video Packetizer pixel repeater FIFO full Value After Reset: 0x0
6	ointemptyrepet	R/W	Mask bit for Video Packetizer pixel repeater FIFO empty Value After Reset: 0x0
5	ointfullpp	R/W	Mask bit for Video Packetizer pixel packing FIFO

			full Value After Reset: 0x0
4	ointemptypp	R/W	Mask bit for Video Packetizer pixel packing FIFO empty Value After Reset: 0x0
3	ointfullremap	R/W	Mask bit for Video Packetizer pixel YCC 422 re-mapper FIFO full Value After Reset: 0x0
2	ointemptyremap	R/W	Mask bit for Video Packetizer pixel YCC 422 re-mapper FIFO empty Value After Reset: 0x0
1	spare_2	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
0	spare_1	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0

FrameComposer Registers

Frame Composer Registers. Follow the link for the register to see a detailed description of the register.

Registers for Address Block: FrameComposer

Register	Offset	Description
fc_invidconf	0x1000	Frame Composer Input Video Configuration and HDCP Keepout Register
fc_inhactiv0	0x1001	Frame Composer Input Video HActive Pixels Register 0
fc_inhactiv1	0x1002	Frame Composer Input Video HActive Pixels Register 1
fc_inhblank0	0x1003	Frame Composer Input Video HBlank Pixels Register 0
fc_inhblank1	0x1004	Frame Composer Input Video HBlank Pixels Register 1
fc_invactiv0	0x1005	Frame Composer Input Video VActive Pixels Register 0
fc_invactiv1	0x1006	Frame Composer Input Video VActive Pixels Register 1
fc_invblank	0x1007	Frame Composer Input Video VBlank Pixels Register
fc_hsyncindelay0	0x1008	Frame Composer Input Video HSync Front Porch Register 0
fc_hsyncindelay1	0x1009	Frame Composer Input Video HSync Front Porch Register 1
fc_hsyncinwidth0	0x100a	Frame Composer Input Video HSync Width Register 0
fc_hsyncinwidth1	0x100b	Frame Composer Input Video HSync Width Register 1
fc_vsyncindelay	0x100c	Frame Composer Input Video VSync Front Porch Register

fc_vsyncinwidth	0x100d	Frame Composer Input Video VSync Width Register
fc_infreq0	0x100e	Frame Composer Input Video Refresh Rate Register 0
fc_infreq1	0x100f	Frame Composer Input Video Refresh Rate Register 1
fc_infreq2	0x1010	Frame Composer Input Video Refresh Rate Register 2
fc_ctrldur	0x1011	Frame Composer Control Period Duration Register
fc_exctrldur	0x1012	Frame Composer Extended Control Period Duration Register
fc_exctrlspac	0x1013	Frame Composer Extended Control Period Maximum Spacing Register
fc_ch0pream	0x1014	Frame Composer Channel 0 Non-Preamble Data Register
fc_ch1pream	0x1015	Frame Composer Channel 1 Non-Preamble Data Register
fc_ch2pream	0x1016	Frame Composer Channel 2 Non-Preamble Data Register
fc_aviconf3	0x1017	Frame Composer AVI Packet Configuration Register 3
fc_gcp	0x1018	Frame Composer GCP Packet Configuration Register
fc_aviconf0	0x1019	Frame Composer AVI Packet Configuration Register 0
fc_aviconf1	0x101a	Frame Composer AVI Packet Configuration Register 1

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_aviconf2	0x101b	Frame Composer AVI Packet Configuration Register 2
fc_avivid	0x101c	Frame Composer AVI Packet VIC Register
fc_avietb[0:1]	0x101d + (i * 0x1)	Frame Composer AVI Packet End of Top Bar Register Array
fc_avisbb[0:1]	0x101f + (i * 0x1)	Frame Composer AVI Packet Start of Bottom Bar Register Array
fc_avielb[0:1]	0x1021 + (i * 0x1)	Frame Composer AVI Packet End of Left Bar Register Array
fc_avisrb[0:1]	0x1023 + (i * 0x1)	Frame Composer AVI Packet Start of Right Bar Register Array
fc_audiconf0	0x1025	Frame Composer AUD Packet Configuration Register 0
fc_audiconf1	0x1026	Frame Composer AUD Packet Configuration Register 1
fc_audiconf2	0x1027	Frame Composer AUD Packet Configuration Register 2
fc_audiconf3	0x1028	Frame Composer AUD Packet Configuration Register 3
fc_vsdieeeid2	0x1029	Frame Composer VSI Packet Data IEEE Register 2
fc_vsdsize	0x102a	Frame Composer VSI Packet Data Size Register
fc_vsdieeeid1	0x1030	Frame Composer VSI Packet Data IEEE Register 1
fc_vsdieeeid0	0x1031	Frame Composer VSI Packet Data IEEE Register 0
fc_vsdpayload[0:23]	0x1032 + (i * 0x1)	Frame Composer VSI Packet Data Payload Register Array

Register	Offset	Description
fc_spdvendorname[0:7]	0x104a + (i * 0x1)	Frame Composer SPD Packet Data Vendor Name Register Array
fc_spdproductname[0:15]	0x1052 + (i * 0x1)	Frame Composer SPD packet Data Product Name Register Array
fc_spddeviceinf	0x1062	Frame Composer SPD Packet Data Source Product Descriptor Register
fc_audsconf	0x1063	Frame Composer Audio Sample Flat and Layout Configuration Register
fc_audsstat	0x1064	Frame Composer Audio Sample Flat and Layout Status Register
fc_audsv	0x1065	Frame Composer Audio Sample Validity Flag Register
fc_audsu	0x1066	Frame Composer Audio Sample User Flag Register
fc_audschnl0	0x1067	Frame Composer Audio Sample Channel Status Configuration Register 0
fc_audschnl1	0x1068	Frame Composer Audio Sample Channel Status Configuration Register 1

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_audschnl2	0x1069	Frame Composer Audio Sample Channel Status Configuration Register 2
fc_audschnl3	0x106a	Frame Composer Audio Sample Channel Status Configuration Register 3
fc_audschnl4	0x106b	Frame Composer Audio Sample Channel Status Configuration Register 4
fc_audschnl5	0x106c	Frame Composer Audio Sample Channel Status Configuration Register 5
fc_audschnl6	0x106d	Frame Composer Audio Sample Channel Status Configuration Register 6
fc_audschnl7	0x106e	Frame Composer Audio Sample Channel Status Configuration Register 7
fc_audschnl8	0x106f	Frame Composer Audio Sample Channel Status Configuration Register 8
fc_ctrlqhigh	0x1073	Frame Composer Number of High Priority Packets Attended Configuration Register
fc_ctrlqlow	0x1074	Frame Composer Number of Low Priority Packets Attended Configuration Register
fc_acp0	0x1075	Frame Composer ACP Packet Type Configuration Register 0
fc_acp16	0x1082	Frame Composer ACP Packet Body Configuration Register 16
fc_acp15	0x1083	Frame Composer ACP Packet Body Configuration Register 15
fc_acp14	0x1084	Frame Composer ACP Packet Body Configuration Register 14
fc_acp13	0x1085	Frame Composer ACP Packet Body Configuration

Register	Offset	Description
		Register 13
fc_acp12	0x1086	Frame Composer ACP Packet Body Configuration Register 12
fc_acp11	0x1087	Frame Composer ACP Packet Body Configuration Register 11
fc_acp10	0x1088	Frame Composer ACP Packet Body Configuration Register 10
fc_acp9	0x1089	Frame Composer ACP Packet Body Configuration Register 9
fc_acp8	0x108a	Frame Composer ACP Packet Body Configuration Register 8
fc_acp7	0x108b	Frame Composer ACP Packet Body Configuration Register 7
fc_acp6	0x108c	Frame Composer ACP Packet Body Configuration Register 6

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_acp5	0x108d	Frame Composer ACP Packet Body Configuration Register 5
fc_acp4	0x108e	Frame Composer ACP Packet Body Configuration Register 4
fc_acp3	0x108f	Frame Composer ACP Packet Body Configuration Register 3
fc_acp2	0x1090	Frame Composer ACP Packet Body Configuration Register 2
fc_acp1	0x1091	Frame Composer ACP Packet Body Configuration Register 1
fc_iscr1_0	0x1092	Frame Composer ISRC1 Packet Status, Valid, and Continue Configuration Register
fc_iscr1_16	0x1093	Frame Composer ISRC1 Packet Body Register 16
fc_iscr1_15	0x1094	Frame Composer ISRC1 Packet Body Register 15
fc_iscr1_14	0x1095	Frame Composer ISRC1 Packet Body Register 14
fc_iscr1_13	0x1096	Frame Composer ISRC1 Packet Body Register 13
fc_iscr1_12	0x1097	Frame Composer ISRC1 Packet Body Register 12
fc_iscr1_11	0x1098	Frame Composer ISRC1 Packet Body Register 11
fc_iscr1_10	0x1099	Frame Composer ISRC1 Packet Body Register 10
fc_iscr1_9	0x109a	Frame Composer ISRC1 Packet Body Register 9
fc_iscr1_8	0x109b	Frame Composer ISRC1 Packet Body Register 8
fc_iscr1_7	0x109c	Frame Composer ISRC1 Packet Body Register 7
fc_iscr1_6	0x109d	Frame Composer ISRC1 Packet Body Register 6
fc_iscr1_5	0x109e	Frame Composer ISRC1 Packet Body Register 5
fc_iscr1_4	0x109f	Frame Composer ISRC1 Packet Body Register 4
fc_iscr1_3	0x10a0	Frame Composer ISRC1 Packet Body Register 3
fc_iscr1_2	0x10a1	Frame Composer ISRC1 Packet Body Register 2
fc_iscr1_1	0x10a2	Frame Composer ISRC1 Packet Body Register 1

Register	Offset	Description
fc_iscr2_15	0x10a3	Frame Composer ISRC2 Packet Body Register 15
fc_iscr2_14	0x10a4	Frame Composer ISRC2 Packet Body Register 14
fc_iscr2_13	0x10a5	Frame Composer ISRC2 Packet Body Register 13
fc_iscr2_12	0x10a6	Frame Composer ISRC2 Packet Body Register 12
fc_iscr2_11	0x10a7	Frame Composer ISRC2 Packet Body Register 11
fc_iscr2_10	0x10a8	Frame Composer ISRC2 Packet Body Register 10
fc_iscr2_9	0x10a9	Frame Composer ISRC2 Packet Body Register 9
fc_iscr2_8	0x10aa	Frame Composer ISRC2 Packet Body Register 8
fc_iscr2_7	0x10ab	Frame Composer ISRC2 Packet Body Register 7

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_iscr2_6	0x10ac	Frame Composer ISRC2 Packet Body Register 6
fc_iscr2_5	0x10ad	Frame Composer ISRC2 Packet Body Register 5
fc_iscr2_4	0x10ae	Frame Composer ISRC2 Packet Body Register 4
fc_iscr2_3	0x10af	Frame Composer ISRC2 Packet Body Register 3
fc_iscr2_2	0x10b0	Frame Composer ISRC2 Packet Body Register 2
fc_iscr2_1	0x10b1	Frame Composer ISRC2 Packet Body Register 1
fc_iscr2_0	0x10b2	Frame Composer ISRC2 Packet Body Register 0
fc_datauto0	0x10b3	Frame Composer Data Island Auto Packet Scheduling Register 0 Configures the Frame Composer RDRB(1)/ Manual(0)...
fc_datauto1	0x10b4	Frame Composer Data Island Auto Packet Scheduling Register 1 Configures the Frame Composer (FC)...
fc_datauto2	0x10b5	Frame Composer Data Island Auto packet scheduling Register 2 Configures the Frame Composer (FC)...
fc_datman	0x10b6	Frame Composer Data Island Manual Packet Request Register Requests to the Frame Composer the data...
fc_datauto3	0x10b7	Frame Composer Data Island Auto Packet Scheduling Register 3 Configures the Frame Composer Automatic(1)/ RDRB(0)...
fc_rdrb0	0x10b8	Frame Composer Round Robin ACR Packet Insertion Register 0 Configures the Frame Composer (FC) RDRB...
fc_rdrb1	0x10b9	Frame Composer Round Robin ACR Packet Insertion Register 1 Configures the Frame Composer (FC) RDRB...
fc_rdrb2	0x10ba	Frame Composer Round Robin AUDI Packet Insertion Register 2 Configures the Frame Composer (FC)...
fc_rdrb3	0x10bb	Frame Composer Round Robin AUDI Packet Insertion Register 3 Configures the Frame Composer (FC)...
fc_rdrb4	0x10bc	Frame Composer Round Robin GCP Packet Insertion Register 4 Configures the Frame Composer (FC) RDRB...
fc_rdrb5	0x10bd	Frame Composer Round Robin GCP Packet Insertion

Register	Offset	Description
		Register 5 Configures the Frame Composer (FC) RDRB...
fc_rdrb6	0x10be	Frame Composer Round Robin AVI Packet Insertion Register 6 Configures the Frame Composer (FC) RDRB...
fc_rdrb7	0x10bf	Frame Composer Round Robin AVI Packet Insertion Register 7 Configures the Frame Composer (FC) RDRB...
fc_rdrb8	0x10c0	Frame Composer Round Robin AMP Packet Insertion Register 8

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_rdrb9	0x10c1	Frame Composer Round Robin AMP Packet Insertion Register 9
fc_rdrb10	0x10c2	Frame Composer Round Robin NTSC VBI Packet Insertion Register 10
fc_rdrb11	0x10c3	Frame Composer Round Robin NTSC VBI Packet Insertion Register 11
fc_rdrb12	0x10c4	Frame Composer Round Robin DRM Packet Insertion Register 12
fc_rdrb13	0x10c5	Frame Composer Round Robin DRM Packet Insertion Register 13
fc_mask0	0x10d2	Frame Composer Packet Interrupt Mask Register 0
fc_mask1	0x10d6	Frame Composer Packet Interrupt Mask Register 1
fc_mask2	0x10da	Frame Composer High/Low Priority Overflow Interrupt Mask Register 2
fc_prconf	0x10e0	Frame Composer Pixel Repetition Configuration Register
fc_scrambler_ctrl	0x10e1	Frame Composer Scrambler Control
fc_multistream_ctrl	0x10e2	Frame Composer Multi-Stream Audio Control
fc_packet_tx_en	0x10e3	Frame Composer Packet Transmission Control
fc_actspc_hdlr_cfg	0x10e8	Frame Composer Active Space Control
fc_invact_2d_0	0x10e9	Frame Composer Input Video 2D VActive Pixels Register 0
fc_invact_2d_1	0x10ea	Frame Composer Input Video VActive pixels Register 1
fc_gmd_stat	0x1100	Frame Composer GMD Packet Status Register Gamut metadata packet status bit information for no_current_gmd,...
fc_gmd_en	0x1101	Frame Composer GMD Packet Enable Register This register enables Gamut metadata (GMD) packet transmission....
fc_gmd_up	0x1102	Frame Composer GMD Packet Update Register This register performs an GMD packet content update

Register	Offset	Description
		according...
fc_gmd_conf	0x1103	Frame Composer GMD Packet Schedule Configuration Register This register configures the number of...
fc_gmd_hb	0x1104	Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register This register configures...
fc_gmd_pb[0:27]	0x1105 + (i * 0x1)	Frame Composer GMD Packet Body Register Array Configures the GMD packet body of the GMD...
fc_amp_hb1	0x1128	Frame Composer AMP Packet Header Register 1
fc_amp_hb2	0x1129	Frame Composer AMP Packet Header Register 2

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_amp_pb[0:27]	0x112a + (i * 0x1)	Frame Composer AMP Packet Body Register Array
fc_nvbi_hb1	0x1148	Frame Composer NTSC VBI Packet Header Register 1
fc_nvbi_hb2	0x1149	Frame Composer NTSC VBI Packet Header Register 2
fc_nvbi_pb[0:26]	0x114a + (i * 0x1)	Frame Composer NTSC VBI Packet Body Register Array
fc_drm_up	0x1167	Frame Composer DRM Packet Update Register
fc_drm_hb[0:1]	0x1168 + (i * 0x1)	Frame Composer DRM Packet Header Register Array
fc_drm_pb[0:26]	0x116a + (i * 0x1)	Frame Composer DRM Packet Body Register Array
fc_dbgforce	0x1200	Frame Composer video/audio Force Enable Register This register allows to force the controller to...
fc_dbgaud0ch0	0x1201	Frame Composer Audio Data Channel 0 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch0	0x1202	Frame Composer Audio Data Channel 0 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch0	0x1203	Frame Composer Audio Data Channel 0 Register 2 Configures the audio fixed data to be used in channel...
fc_dbgaud0ch1	0x1204	Frame Composer Audio Data Channel 1 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch1	0x1205	Frame Composer Audio Data Channel 1 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch1	0x1206	Frame Composer Audio Data Channel 1 Register 2 Configures the audio fixed data to be used in

Register	Offset	Description
		channel...
fc_dbgaud0ch2	0x1207	Frame Composer Audio Data Channel 2 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch2	0x1208	Frame Composer Audio Data Channel 2 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch2	0x1209	Frame Composer Audio Data Channel 2 Register 2 Configures the audio fixed data to be used in channel...
fc_dbgaud0ch3	0x120a	Frame Composer Audio Data Channel 3 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch3	0x120b	Frame Composer Audio Data Channel 3 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch3	0x120c	Frame Composer Audio Data Channel 3 Register 2 Configures the audio fixed data to be used in channel...

Registers for Address Block: FrameComposer (Continued)

Register	Offset	Description
fc_dbgaud0ch4	0x120d	Frame Composer Audio Data Channel 4 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch4	0x120e	Frame Composer Audio Data Channel 4 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch4	0x120f	Frame Composer Audio Data Channel 4 Register 2 Configures the audio fixed data to be used in channel...
fc_dbgaud0ch5	0x1210	Frame Composer Audio Data Channel 5 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch5	0x1211	Frame Composer Audio Data Channel 5 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch5	0x1212	Frame Composer Audio Data Channel 5 Register 2 Configures the audio fixed data to be used in channel...
fc_dbgaud0ch6	0x1213	Frame Composer Audio Data Channel 6 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch6	0x1214	Frame Composer Audio Data Channel 6 Register 1 Configures the audio fixed data to be used in channel...

Register	Offset	Description
fc_dbgaud2ch6	0x1215	Frame Composer Audio Data Channel 6 Register 2 Configures the audio fixed data to be used in channel...
fc_dbgaud0ch7	0x1216	Frame Composer Audio Data Channel 7 Register 0 Configures the audio fixed data to be used in channel...
fc_dbgaud1ch7	0x1217	Frame Composer Audio Data Channel 7 Register 1 Configures the audio fixed data to be used in channel...
fc_dbgaud2ch7	0x1218	Frame Composer Audio Data Channel 7 Register 2 Configures the audio fixed data to be used in channel...
fc_dbgtmds[0:2]	0x1219 + (i * 0x1)	Frame Composer TMDS Data Channel Register Array Configures the video fixed data to be used in TMDS...

fc_invidconf

Description: Frame Composer Input Video Configuration and HDCP Keepout Register
Size: 8 bits
Offset: 0x1000

Bits	Name	Attr	Description
7	HDCP_keepout	R/W	Start/stop HDCP keepout window generation 1b: Active Value After Reset: 0x0
6	vsync_in_polarity	R/W	Vsync input polarity 1b: Active high 0b: Active low Value After Reset: 0x1
5	hsync_in_polarity	R/W	Hsync input polarity 1b: Active high 0b: Active low Value After Reset: 0x1
4	de_in_polarity	R/W	Data enable input polarity 1b: Active high 0b: Active low Value After Reset: 0x1
3	DVI_modez	R/W	Active low 0b: DVI mode selected 1b: HDMI mode selected Value After Reset: 0x0
2			Reserved for future use.
1	r_v_blank_in_osc	R/W	Used for CEA861-D modes with fractional Vblank (for example, modes 5, 6, 7, 10, 11, 20, 21, and 22). For more modes, see the CEA861-D specification. Note: Set this field to 1 for video mode 39, although there is no Vblank oscillation. 1b: Active high Value After Reset: 0x0

Fields for Register: fc_invidconf (Continued)

Bits	Name	Attr	Description
0	in_I_P	R/W	Input video mode: 1b: Interlaced 0b: Progressive Value After Reset: 0x0

fc_inhactiv0

Description: Frame Composer Input Video HActive Pixels Register 0
Size: 8 bits
Offset: 0x1001

Bits	Name	Attr	Description
7:0	H_in_activ	R/W	Input video Horizontal active pixel region width. Number of Horizontal active pixels [0...8191]. Value After Reset: 0x0

fc_inhactiv1

Description: Frame Composer Input Video HActive Pixels Register 1
Size: 8 bits
Offset: 0x1002

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	H_in_activ_13	R/W	Input video Horizontal active pixel region width (0 .. 16383) If the configuration parameter HDMI_TX_20 = True (1), this bit field holds bit 13. Value After Reset: 0x0

Fields for Register: fc_inhactiv1 (Continued)

Bits	Name	Attr	Description
4	H_in_activ_12	R/W	Input video Horizontal active pixel region width (0 . . 8191) If configuration parameter HDMI_TX_14 = True (1), this bit field holds bit 12. Value After Reset: 0x0
3:0	H_in_activ	R/W	Input video Horizontal active pixel region width Value After Reset: 0x0

fc_inhblank0

Description: Frame Composer Input Video HBlank Pixels Register 0
Size: 8 bits
Offset: 0x1003

Bits	Name	Attr	Description
7:0	H_in_blank	R/W	Input video Horizontal blanking pixel region width. Number of Horizontal blanking pixels [0...4095]. Value After Reset: 0x0

fc_inhblank1

Description: Frame Composer Input Video HBlank Pixels Register 1
Size: 8 bits
Offset: 0x1004

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:2	H_in_blank_12	R/W	Input video Horizontal blanking pixel region width If configuration parameter HDMI_TX_14 = True (1), this bit field holds bit 12:10 of number of horizontal blanking pixels. Value After Reset: 0x0
1:0	H_in_blank	R/W	Input video Horizontal blanking pixel region width this bit field holds bits 9:8 of number of Horizontal blanking pixels. Value After Reset: 0x0

fc_invactiv0
Description: Frame Composer Input Video VActive Pixels Register 0
Size: 8 bits
Offset: 0x1005

Bits	Name	Attr	Description
7:0	V_in_activ	R/W	Input video Vertical active pixel region width. This bit field holds bits 7:0 of number of Vertical active pixels. Value After Reset: 0x0

fc_invactiv1
Description: Frame Composer Input Video VActive Pixels Register 1
Size: 8 bits
Offset: 0x1006

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:3	V_in_activ_12_11	R/W	Input video Vertical active pixel region width. If the configuration parameter HDMI_TX_14 = True (1), this bit field holds bits 12:10 of number of Vertical active pixels. Value After Reset: 0x0
2:0	V_in_activ	R/W	Input video Vertical active pixel region width. This bit field holds bits 9:8 of number of Vertical active pixels. Value After Reset: 0x0

fc_invblank
Description: Frame Composer Input Video VBlank Pixels Register
Size: 8 bits
Offset: 0x1007

Bits	Name	Attr	Description

7:0	V_in_blank	R/W	Input video Vertical blanking pixel region width. Number of Vertical blanking lines [0...255]. Value After Reset: 0x0

fc_hsyncindelay0

Description: Frame Composer Input Video HSync Front Porch Register 0

Size: 8 bits

Offset: 0x1008

Bits	Name	Attr	Description
7:0	H_in_delay	R/W	Input video Hsync active edge delay. Integer number of pixel clock cycles from "de" non active edge of the last "de" valid period [0...4095]. Value After Reset: 0x0

fc_hsyncindelay1

Description: Frame Composer Input Video HSync Front Porch Register 1

Size: 8 bits

Offset: 0x1009

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:3	H_in_delay_12	R/W	Input video Horizontal active edge delay. If configuration parameter HDMI_TX_14 = True (1), this bit field holds bit 12. Integer number of pixel clock cycles from "de" non-active edge of the last "de" valid period [0...8191]. Value After Reset: 0x0
2:0	H_in_delay	R/W	Input video Horizontal active edge delay. Value After Reset: 0x0

fc_hsyncinwidth0

Description: Frame Composer Input Video HSync Width Register 0

Size: 8 bits

Offset: 0x100a

Bits	Name	Attr	Description
7:0	H_in_width	R/W	Input video Hsync active pulse width. Integer number of pixel clock cycles [0...511]. Value After Reset: 0x0

fc_hsyncinwidth1

Description: Frame Composer Input Video HSync Width Register 1

Size: 8 bits

Offset: 0x100b

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	H_in_width_9	R/W	Input video Hsync active pulse width. If configuration parameter HDMI_TX_14 = True

			(1), then this bit field holds bit 9. Number of Horizontal active pixels [0...1024]. Value After Reset: 0x0
0	H_in_width	R/W	Input video Hsync active pulse width. Value After Reset: 0x0

fc_vsyncindelay

Description: Frame Composer Input Video VSync Front Porch Register

Size: 8 bits

Offset: 0x100c

Bits	Name	Attr	Description
7:0	V_in_delay	R/W	Input video Vsync active edge delay. Integer number of Hsync pulses from "de" non active edge of the last "de" valid period. [0...255]. Value After Reset: 0x0

fc_vsyncinwidth

Description: Frame Composer Input Video VSync Width Register

Size: 8 bits

Offset: 0x100d

Bits	Name	Attr	Description
7:6			Reserved for future use.
5:0	V_in_width	R/W	Description: Input video Vsync active pulse width. Integer number of video lines [0...63]. Value After Reset: 0x0

fc_infreq0

Description: Frame Composer Input Video Refresh Rate Register 0

Size: 8 bits

Offset: 0x100e

Bits	Name	Attr	Description
7:0	infreq	R/W	Video refresh rate in Hz*1E3 format. This register is provided for debug and informative purposes. The Hdmi_tx does not write any data to this register; the data written by software is not used by the Hdmi_tx. Value After Reset: 0x0

fc_infreq1

Description: Frame Composer Input Video Refresh Rate Register 1

Size: 8 bits

Offset: 0x100f

Bits	Name	Attr	Description
7:0	infreq	R/W	Video refresh rate in Hz*1E3 format. This register is provided for debug and informative purposes. The Hdmi_tx does not write any data to this

			register; the data written by software is not used by the Hdmi_tx. Value After Reset: 0x0

fc_infreq2

Description: Frame Composer Input Video Refresh Rate Register 2

Size: 8 bits

Offset: 0x1010

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	infreq	R/W	Video refresh rate in Hz*1E3 format. This register is provided for debug and informative purposes. The Hdmi_tx does not write any data to this register; the data written by software is not used by the Hdmi_tx. Value After Reset: 0x0

fc_ctrldur

Description: Frame Composer Control Period Duration Register

Size: 8 bits

Offset: 0x1011

Bits	Name	Attr	Description
7:0	ctrlperiodduration	R/W	Configuration of the control period minimum duration (minimum of 12 pixel clock cycles; refer to HDMI 1.4b specification). Integer number of pixel clocks cycles [0..223]. Value After Reset: 0x0

fc_exctrldur

Description: Frame Composer Extended Control Period Duration Register

Size: 8 bits

Offset: 0x1012

Bits	Name	Attr	Description
7:0	exctrlperiodduration	R/W	Configuration of the extended control period minimum duration (minimum of 32 pixel clock cycles; refer to HDMI 1.4b specification). Integer number of pixel clocks cycles [0..223]. Value After Reset: 0x0

fc_exctrlspac

Description: Frame Composer Extended Control Period Maximum Spacing Register

Size: 8 bits

Offset: 0x1013

Bits	Name	Attr	Description
7:0	exctrlperiodspacing	R/W	Configuration of the maximum spacing between consecutive extended control periods (maximum of

			50ms; refer to the applicable HDMI specification). When using the HDMI 2.0 supported features (HDMI_TX_20 = 1): generated spacing = (1/freq tmds clock)*256*512*(extctrlperiodspacing +1) else generated spacing = (1/freq tmds clock)*256*256*(extctrlperiodspacing +1) Value After Reset: 0x0

fc_ch0pream

Description: Frame Composer Channel 0 Non-Preamble Data Register

Size: 8 bits

Offset: 0x1014

Bits	Name	Attr	Description
7:0	ch0_preamble_filter	R/W	When in control mode, configures 8 bits that fill the channel 0 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1.4b specification). Value After Reset: 0x0

fc_ch1pream

Description: Frame Composer Channel 1 Non-Preamble Data Register

Size: 8 bits

Offset: 0x1015

Bits	Name	Attr	Description
7:6			Reserved for future use.
5:0	ch1_preamble_filter	R/W	When in control mode, configures 6 bits that fill the channel 1 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1.4b specification). Value After Reset: 0x0

fc_ch2pream

Description: Frame Composer Channel 2 Non-Preamble Data Register

Size: 8 bits

Offset: 0x1016

Bits	Name	Attr	Description
7:6			Reserved for future use.
5:0	ch2_preamble_filter	R/W	When in control mode, configures 6 bits that fill the channel 2 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1.4b specification). Value After Reset: 0x0

fc_aviconf3

Description: Frame Composer AVI Packet Configuration Register 3

Size: 8 bits
Offset: 0x1017

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:2	YQ	R/W	YCC Quantization range according to the CEA specification Value After Reset: 0x0
1:0	CN	R/W	IT content type according to CEA the specification Value After Reset: 0x0

fc_gcp
Description: Frame Composer GCP Packet Configuration Register
Size: 8 bits
Offset: 0x1018

Bits	Name	Attr	Description
7:3			Reserved for future use.
2	default_phase	R/W	Value of "default_phase" in the GCP packet. This data must be equal to the default phase used at Video Packetizer packing machine. Value After Reset: 0x0
1	set_avmute	R/W	Value of "set_avmute" in the GCP packet Once the AVmute is set, the frame composer schedules the GCP packet with AVmute set in the packet scheduler to be sent once (may only be transmitted between the active edge of VSYNC and 384 pixels following this edge). Value After Reset: 0x0
0	clear_avmute	R/W	Value of "clear_avmute" in the GCP packet Value After Reset: 0x0

fc_aviconf0
Description: Frame Composer AVI Packet Configuration Register 0
Size: 8 bits
Offset: 0x1019

Bits	Name	Attr	Description
7	rgc_ycc_indication_2	R/W	Y2, Bit 2 of rgc_ycc_indication Value After Reset: 0x0
6	active_format_present	R/W	Active format present Value After Reset: 0x0
5:4	scan_information	R/W	Scan information Value After Reset: 0x0
3:2	bar_information	R/W	Bar information data valid Value After Reset: 0x0
1:0	rgc_ycc_indication	R/W	Y1,Y0 RGB or YCC indicator Value After Reset: 0x0

fc_aviconf1
Description: Frame Composer AVI Packet Configuration Register 1
Size: 8 bits

Offset: 0x101a

Bits	Name	Attr	Description
7:6	Colorimetry	R/W	Colorimetry Value After Reset: 0x0
5:4	picture_aspect_ratio	R/W	Picture aspect ratio Value After Reset: 0x0
3:0	active_aspect_ratio	R/W	Active aspect ratio Value After Reset: 0x0

fc_aviconf2

Description: Frame Composer AVI Packet Configuration Register 2

Size: 8 bits

Offset: 0x101b

Bits	Name	Attr	Description
7	it_content	R/W	IT content Value After Reset: 0x0
6:4	extended_colorimetry	R/W	Extended colorimetry Value After Reset: 0x0
3:2	quantization_range	R/W	Quantization range Value After Reset: 0x0
1:0	non_uniform_picture_scaling	R/W	Non-uniform picture scaling Value After Reset: 0x0

fc_avivid

Description: Frame Composer AVI Packet VIC Register

Size: 8 bits

Offset: 0x101c

Bits	Name	Attr	Description
7	fc_avivid_7	R/W	Bit 7 of fc_avivid register Value After Reset: 0x0
6:0	fc_avivid	R/W	Configures the AVI InfoFrame Video Identification code. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_avietb[0:1]

Description: Frame Composer AVI Packet End of Top Bar Register Array

Size: 8 bits

Offset: 0x101d + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_avietb	R/W	Defines the AVI InfoFrame End of Top Bar value. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_avisbb[0:1]

Description: Frame Composer AVI Packet Start of Bottom Bar Register Array

Size: 8 bits

Offset: 0x101f + (i * 0x1)

Bits	Name	Attr	Description

7:0	fc_avisbb	R/W	This register defines the AVI InfoFrame Start of Bottom Bar value. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_avielb[0:1]

Description: Frame Composer AVI Packet End of Left Bar Register Array

Size: 8 bits

Offset: 0x1021 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_avielb	R/W	This register defines the AVI InfoFrame End of Left Bar value. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_avisrb[0:1]

Description: Frame Composer AVI Packet Start of Right Bar Register Array

Size: 8 bits

Offset: 0x1023 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_avisrb	R/W	This register defines the AVI InfoFrame Start of Right Bar value. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_audiconf0

Description: Frame Composer AUD Packet Configuration Register 0

Size: 8 bits

Offset: 0x1025

Bits	Name	Attr	Description
7			Reserved for future use.
6:4	CC	R/W	Channel count Value After Reset: 0x0
3:0	CT	R/W	Coding Type Value After Reset: 0x0

fc_audiconf1

Description: Frame Composer AUD Packet Configuration Register 1

Size: 8 bits

Offset: 0x1026

Bits	Name	Attr	Description
7:6			Reserved for future use.
5:4	SS	R/W	Sampling size Value After Reset: 0x0
3			Reserved for future use.
2:0	SF	R/W	Sampling frequency Value After Reset: 0x0

fc_audiconf2

Description: Frame Composer AUD Packet Configuration Register 2

Size: 8 bits

Offset: 0x1027

Bits	Name	Attr	Description
7:0	CA	R/W	Channel allocation Value After Reset: 0x0

fc_audiconf3

Description: Frame Composer AUD Packet Configuration Register 3

Size: 8 bits

Offset: 0x1028

Bits	Name	Attr	Description
7			Reserved for future use.
6:5	LFEPBL	R/W	LFE playback information LFEPBL1, LFEPBL0 LFE playback level as compared to the other channels. Value After Reset: 0x0
4	DM_INH	R/W	Down mix enable Value After Reset: 0x0
3:0	LSV	R/W	Level shift value (for down mixing) Value After Reset: 0x0

fc_vsdieeeid2

Description: Frame Composer VSI Packet Data IEEE Register 2

Size: 8 bits

Offset: 0x1029

Bits	Name	Attr	Description
7:0	IEEE	R/W	This register configures the Vendor Specific InfoFrame IEEE registration identifier. For more information, refer to the CEA- 861-E specification. Value After Reset: 0x0

fc_vsdsize

Description: Frame Composer VSI Packet Data Size Register

Size: 8 bits

Offset: 0x102a

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:0	VSDSIZE	R/W	Packet size as described in the HDMI Vendor Specific InfoFrame (from the HDMI specification). Value After Reset: 0x1b

fc_vsdieeeid1

Description: Frame Composer VSI Packet Data IEEE Register 1

Size: 8 bits

Offset: 0x1030

Bits	Name	Attr	Description
7:0	IEEE	R/W	This register configures the Vendor Specific InfoFrame IEEE registration identifier. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_vsdieeeid0

Description: Frame Composer VSI Packet Data IEEE Register 0

Size: 8 bits

Offset: 0x1031

Bits	Name	Attr	Description
7:0	IEEE	R/W	This register configures the Vendor Specific InfoFrame IEEE registration identifier. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_vsdpayload[0:23]

Description: Frame Composer VSI Packet Data Payload Register Array

Size: 8 bits

Offset: 0x1032 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_vsdpayload	R/W	Frame Composer VSI Packet Data Payload Register Array Configures the Vendor Specific infoFrame 24 bytes specific payload. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_spdvendorname[0:7]

Description: Frame Composer SPD Packet Data Vendor Name Register Array

Size: 8 bits

Offset: 0x104a + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_spdvendorname	R/W	Frame Composer SPD Packet Data Vendor Name Register Array Configures the Source Product Descriptor infoFrame 8 bytes Vendor name. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_spdproductname[0:15]

Description: Frame Composer SPD packet Data Product Name Register Array

Size: 8 bits

Offset: 0x1052 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_spdproductname	R/W	Frame Composer SPD packet Data Product Name Register Array Configures the Source Product

			Descriptor infoFrame 16 bytes Product name. For more information, refer to the CEA-861-E specification. Value After Reset: 0x0

fc_spddeviceinf

Description: Frame Composer SPD Packet Data Source Product Descriptor Register

Size: 8 bits

Offset: 0x1062

Bits	Name	Attr	Description
7:0	fc_spddeviceinf	R/W	Frame Composer SPD Packet Data Source Product Descriptor Register Value After Reset: 0x0

fc_audsconf

Description: Frame Composer Audio Sample Flat and Layout Configuration Register

Size: 8 bits

Offset: 0x1063

Bits	Name	Attr	Description
7:4	aud_packet_sampflt	R/W	Set the audio packet sample flat value to be sent on the packet. Value After Reset: 0x0
3:1			Reserved for future use.
0	aud_packet_layout	R/W	Set the audio packet layout to be sent in the packet: 1b: layout 1 0b: layout 0 If HDMI_TX_20 is defined and register field fc_multistream_ctrl.fc_mas_packet_en is active, this bit has no effect. Value After Reset: 0x0

fc_audsstat

Description: Frame Composer Audio Sample Flat and Layout Status Register

Size: 8 bits

Offset: 0x1064

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	packet_sampprs	R	Shows the data sample present indication of the last Audio sample packet sent by the HDMI Tx Controller. This register information is at TMDS clock rate. Value After Reset: 0x0

fc_audsv

Description: Frame Composer Audio Sample Validity Flag Register

Size: 8 bits

Offset: 0x1065

Bits	Name	Attr	Description
7	V3r	R/W	Set validity bit "V" for Channel 3, Right Value After Reset: 0x0
6	V2r	R/W	Set validity bit "V" for Channel 2, Right Value After Reset: 0x0
5	V1r	R/W	Set validity bit "V" for Channel 1, Right Value After Reset: 0x0
4	V0r	R/W	Set validity bit "V" for Channel 0, Right Value After Reset: 0x0
3	V3l	R/W	Set validity bit "V" for Channel 3, Left Value After Reset: 0x0
2	V2l	R/W	Set validity bit "V" for Channel 2, Left Value After Reset: 0x0
1	V1l	R/W	Set validity bit "V" for Channel 1, Left Value After Reset: 0x0
0	V0l	R/W	Set validity bit "V" for Channel 0, Left Value After Reset: 0x0

fc_audsu

Description: Frame Composer Audio Sample User Flag Register

Size: 8 bits

Offset: 0x1066

Bits	Name	Attr	Description
7	U3r	R/W	Set user bit "U" for Channel 3, Right Value After Reset: 0x0
6	U2r	R/W	Set user bit "U" for Channel 2, Right Value After Reset: 0x0
5	U1r	R/W	Set user bit "U" for Channel 1, Right Value After Reset: 0x0
4	U0r	R/W	Set user bit "U" for Channel 0, Right Value After Reset: 0x0
3	U3l	R/W	Set user bit "U" for Channel 3, Left Value After Reset: 0x0
2	U2l	R/W	Set user bit "U" for Channel 2, Left Value After Reset: 0x0
1	U1l	R/W	Set user bit "U" for Channel 1, Left Value After Reset: 0x0
0	U0l	R/W	Set user bit "U" for Channel 0, Left Value After Reset: 0x0

fc_audschnl0

Description: Frame Composer Audio Sample Channel Status Configuration Register 0

Size: 8 bits

Offset: 0x1067

Bits	Name	Attr	Description

7:6			Reserved for future use.
5:4	oiec_cgmsa	R/W	CGMS-A Value After Reset: 0x0
3:1			Reserved for future use.
0	oiec_copyright	R/W	IEC Copyright indication Value After Reset: 0x0

fc_audschnl1

Description: Frame Composer Audio Sample Channel Status Configuration Register 1

Size: 8 bits

Offset: 0x1068

Bits	Name	Attr	Description
7:0	oiec_categorycode	R/W	Category code Value After Reset: 0x0

fc_audschnl2

Description: Frame Composer Audio Sample Channel Status Configuration Register 2

Size: 8 bits

Offset: 0x1069

Bits	Name	Attr	Description
7			Reserved for future use.
6:4	oiec_pcmaudiomode	R/W	PCM audio mode Value After Reset: 0x0
3:0	oiec_sourcenumber	R/W	Source number Value After Reset: 0x0

fc_audschnl3

Description: Frame Composer Audio Sample Channel Status Configuration Register 3

Size: 8 bits

Offset: 0x106a

Bits	Name	Attr	Description
7:4	oiec_channelnumcr1	R/W	Channel number for second right sample Value After Reset: 0x0
3:0	oiec_channelnumcr0	R/W	Channel number for first right sample Value After Reset: 0x0

fc_audschnl4

Description: Frame Composer Audio Sample Channel Status Configuration Register 4

Size: 8 bits

Offset: 0x106b

Bits	Name	Attr	Description
7:4	oiec_channelnumcr3	R/W	Channel number for fourth right sample Value After Reset: 0x0
3:0	oiec_channelnumcr2	R/W	Channel number for third right sample Value After Reset: 0x0

fc_audschnl5

Description: Frame Composer Audio Sample Channel Status Configuration Register 5
Size: 8 bits
Offset: 0x106c

Bits	Name	Attr	Description
7:4	oiec_channelnumcl1	R/W	Channel number for second left sample Value After Reset: 0x0
3:0	oiec_channelnumcl0	R/W	Channel number for first left sample Value After Reset: 0x0

fc_audschnl6
Description: Frame Composer Audio Sample Channel Status Configuration Register 6
Size: 8 bits
Offset: 0x106d

Bits	Name	Attr	Description
7:4	oiec_channelnumcl3	R/W	Channel number for fourth left sample Value After Reset: 0x0
3:0	oiec_channelnumcl2	R/W	Channel number for third left sample Value After Reset: 0x0

fc_audschnl7
Description: Frame Composer Audio Sample Channel Status Configuration Register 7
Size: 8 bits
Offset: 0x106e

Bits	Name	Attr	Description
7:6	oiec_sampfreq_ext	R/W	Sampling frequency (channel status bits 31 and 30) Value After Reset: 0x0
5:4	oiec_clkaccuracy	R/W	Clock accuracy Value After Reset: 0x0
3:0	oiec_sampfreq	R/W	Sampling frequency Value After Reset: 0x0

fc_audschnl8
Description: Frame Composer Audio Sample Channel Status Configuration Register 8
Size: 8 bits
Offset: 0x106f

Bits	Name	Attr	Description
7:4	oiec_origsampfreq	R/W	Original sampling frequency Value After Reset: 0x0
3:0	oiec_wordlength	R/W	Word length configuration Value After Reset: 0x0

fc_ctrlqhigh
Description: Frame Composer Number of High Priority Packets Attended Configuration Register
Size: 8 bits
Offset: 0x1073

Bits	Name	Attr	Description

7:5			Reserved for future use.
4:0	onhighattended	R/W	Configures the number of high priority packets or audio sample packets consecutively attended before checking low priority queue status. Valid range is from 5'd1 to 5'd31. Value After Reset: 0xf

fc_ctrlqlow

Description: Frame Composer Number of Low Priority Packets Attended Configuration Register

Size: 8 bits

Offset: 0x1074

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:0	onlowattended	R/W	Configures the number of low priority packets or null packets consecutively attended before checking high priority queue status or audio samples availability. Valid range is from 5'd1 to 5'd31. Value After Reset: 0x3

fc_acp0

Description: Frame Composer ACP Packet Type Configuration Register 0

Size: 8 bits

Offset: 0x1075

Bits	Name	Attr	Description
7:0	acptype	R/W	Configures the ACP packet type. Value After Reset: 0x0

fc_acp16

Description: Frame Composer ACP Packet Body Configuration Register 16

Size: 8 bits

Offset: 0x1082

Bits	Name	Attr	Description
7:0	fc_acp16	R/W	Frame Composer ACP Packet Body Configuration Register 16 Value After Reset: 0x0

fc_acp15

Description: Frame Composer ACP Packet Body Configuration Register 15

Size: 8 bits

Offset: 0x1083

Bits	Name	Attr	Description
7:0	fc_acp15	R/W	Frame Composer ACP Packet Body Configuration Register 15 Value After Reset: 0x0

fc_acp14

Description: Frame Composer ACP Packet Body Configuration Register 14

Size: 8 bits

Offset: 0x1084

Bits	Name	Attr	Description
7:0	fc_acp14	R/W	Frame Composer ACP Packet Body Configuration Register 14 Value After Reset: 0x0

fc_acp13

Description: Frame Composer ACP Packet Body Configuration Register 13

Size: 8 bits

Offset: 0x1085

Bits	Name	Attr	Description
7:0	fc_acp13	R/W	Frame Composer ACP Packet Body Configuration Register 13 Value After Reset: 0x0

fc_acp12

Description: Frame Composer ACP Packet Body Configuration Register 12

Size: 8 bits

Offset: 0x1086

Bits	Name	Attr	Description
7:0	fc_acp12	R/W	Frame Composer ACP Packet Body Configuration Register 12 Value After Reset: 0x0

fc_acp11

Description: Frame Composer ACP Packet Body Configuration Register 11

Size: 8 bits

Offset: 0x1087

Bits	Name	Attr	Description
7:0	fc_acp11	R/W	Frame Composer ACP Packet Body Configuration Register 11 Value After Reset: 0x0

fc_acp10

Description: Frame Composer ACP Packet Body Configuration Register 10

Size: 8 bits

Offset: 0x1088

Bits	Name	Attr	Description
7:0	fc_acp10	R/W	Frame Composer ACP Packet Body Configuration Register 10 Value After Reset: 0x0

fc_acp9

Description: Frame Composer ACP Packet Body Configuration Register 9

Size: 8 bits
Offset: 0x1089

Bits	Name	Attr	Description
7:0	fc_acp9	R/W	Frame Composer ACP Packet Body Configuration Register 9 Value After Reset: 0x0

fc_acp8
Description: Frame Composer ACP Packet Body Configuration Register 8
Size: 8 bits
Offset: 0x108a

Bits	Name	Attr	Description
7:0	fc_acp8	R/W	Frame Composer ACP Packet Body Configuration Register 8 Value After Reset: 0x0

fc_acp7
Description: Frame Composer ACP Packet Body Configuration Register 7
Size: 8 bits
Offset: 0x108b

Bits	Name	Attr	Description
7:0	fc_acp7	R/W	Frame Composer ACP Packet Body Configuration Register 7 Value After Reset: 0x0

fc_acp6
Description: Frame Composer ACP Packet Body Configuration Register 6
Size: 8 bits
Offset: 0x108c

Bits	Name	Attr	Description
7:0	fc_acp6	R/W	Frame Composer ACP Packet Body Configuration Register 6 Value After Reset: 0x0

fc_acp5
Description: Frame Composer ACP Packet Body Configuration Register 5
Size: 8 bits
Offset: 0x108d

Bits	Name	Attr	Description
7:0	fc_acp5	R/W	Frame Composer ACP Packet Body Configuration Register 5 Value After Reset: 0x0

fc_acp4
Description: Frame Composer ACP Packet Body Configuration Register 4
Size: 8 bits
Offset: 0x108e

Bits	Name	Attr	Description
7:0	fc_acp4	R/W	Frame Composer ACP Packet Body Configuration Register 4 Value After Reset: 0x0

fc_acp3

Description: Frame Composer ACP Packet Body Configuration Register 3

Size: 8 bits

Offset: 0x108f

Bits	Name	Attr	Description
7:0	fc_acp3	R/W	Frame Composer ACP Packet Body Configuration Register 3 Value After Reset: 0x0

fc_acp2

Description: Frame Composer ACP Packet Body Configuration Register 2

Size: 8 bits

Offset: 0x1090

Bits	Name	Attr	Description
7:0	fc_acp2	R/W	Frame Composer ACP Packet Body Configuration Register 2 Value After Reset: 0x0

fc_acp1

Description: Frame Composer ACP Packet Body Configuration Register 1

Size: 8 bits

Offset: 0x1091

Bits	Name	Attr	Description
7:0	fc_acp1	R/W	Frame Composer ACP Packet Body Configuration Register 1 Value After Reset: 0x0

fc_iscr1_0

Description: Frame Composer ISRC1 Packet Status, Valid, and Continue Configuration Register

Size: 8 bits

Offset: 0x1092

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:2	isrc_status	R/W	ISRC1 Status signal Value After Reset: 0x0
1	isrc_valid	R/W	ISRC1 Valid control signal Value After Reset: 0x0
0	isrc_cont	R/W	ISRC1 Indication of packet continuation (ISRC2 will be transmitted) Value After Reset: 0x0

fc_iscr1_16

Description: Frame Composer ISRC1 Packet Body Register 16
Size: 8 bits
Offset: 0x1093

Bits	Name	Attr	Description
7:0	fc_iscr1_16	R/W	Frame Composer ISRC1 Packet Body Register 16; configures ISRC1 packet body of the ISRC1 packet Value After Reset: 0x0

fc_iscr1_15
Description: Frame Composer ISRC1 Packet Body Register 15
Size: 8 bits
Offset: 0x1094

Bits	Name	Attr	Description
7:0	fc_iscr1_15	R/W	Frame Composer ISRC1 Packet Body Register 15 Value After Reset: 0x0

fc_iscr1_14
Description: Frame Composer ISRC1 Packet Body Register 14
Size: 8 bits
Offset: 0x1095

Bits	Name	Attr	Description
7:0	fc_iscr1_14	R/W	Frame Composer ISRC1 Packet Body Register 14 Value After Reset: 0x0

fc_iscr1_13
Description: Frame Composer ISRC1 Packet Body Register 13
Size: 8 bits
Offset: 0x1096

Bits	Name	Attr	Description
7:0	fc_iscr1_13	R/W	Frame Composer ISRC1 Packet Body Register 13 Value After Reset: 0x0

fc_iscr1_12
Description: Frame Composer ISRC1 Packet Body Register 12
Size: 8 bits
Offset: 0x1097

Bits	Name	Attr	Description
7:0	fc_iscr1_12	R/W	Frame Composer ISRC1 Packet Body Register 12 Value After Reset: 0x0

fc_iscr1_11
Description: Frame Composer ISRC1 Packet Body Register 11
Size: 8 bits
Offset: 0x1098

Bits	Name	Attr	Description
7:0	fc_iscr1_11	R/W	Frame Composer ISRC1 Packet Body Register 11

			Value After Reset: 0x0

fc_iscr1_10
Description: Frame Composer ISRC1 Packet Body Register 10
Size: 8 bits
Offset: 0x1099

Bits	Name	Attr	Description
7:0	fc_iscr1_10	R/W	Frame Composer ISRC1 Packet Body Register 10 Value After Reset: 0x0

fc_iscr1_9
Description: Frame Composer ISRC1 Packet Body Register 9
Size: 8 bits
Offset: 0x109a

Bits	Name	Attr	Description
7:0	fc_iscr1_9	R/W	Frame Composer ISRC1 Packet Body Register 9 Value After Reset: 0x0

fc_iscr1_8
Description: Frame Composer ISRC1 Packet Body Register 8
Size: 8 bits
Offset: 0x109b

Bits	Name	Attr	Description
7:0	fc_iscr1_8	R/W	Frame Composer ISRC1 Packet Body Register 8 Value After Reset: 0x0

fc_iscr1_7
Description: Frame Composer ISRC1 Packet Body Register 7
Size: 8 bits
Offset: 0x109c

Bits	Name	Attr	Description
7:0	fc_iscr1_7	R/W	Frame Composer ISRC1 Packet Body Register 7 Value After Reset: 0x0

fc_iscr1_6
Description: Frame Composer ISRC1 Packet Body Register 6
Size: 8 bits
Offset: 0x109d

Bits	Name	Attr	Description
7:0	fc_iscr1_6	R/W	Frame Composer ISRC1 Packet Body Register 6 Value After Reset: 0x0

fc_iscr1_5
Description: Frame Composer ISRC1 Packet Body Register 5
Size: 8 bits
Offset: 0x109e

Bits	Name	Attr	Description
7:0	fc_iscr1_5	R/W	Frame Composer ISRC1 Packet Body Register 5 Value After Reset: 0x0

fc_iscr1_4

Description: Frame Composer ISRC1 Packet Body Register 4

Size: 8 bits

Offset: 0x109f

Bits	Name	Attr	Description
7:0	fc_iscr1_4	R/W	Frame Composer ISRC1 Packet Body Register 4 Value After Reset: 0x0

fc_iscr1_3

Description: Frame Composer ISRC1 Packet Body Register 3

Size: 8 bits

Offset: 0x10a0

Bits	Name	Attr	Description
7:0	fc_iscr1_3	R/W	Frame Composer ISRC1 Packet Body Register 3 Value After Reset: 0x0

fc_iscr1_2

Description: Frame Composer ISRC1 Packet Body Register 2

Size: 8 bits

Offset: 0x10a1

Bits	Name	Attr	Description
7:0	fc_iscr1_2	R/W	Frame Composer ISRC1 Packet Body Register 2 Value After Reset: 0x0

fc_iscr1_1

Description: Frame Composer ISRC1 Packet Body Register 1

Size: 8 bits

Offset: 0x10a2

Bits	Name	Attr	Description
7:0	fc_iscr1_1	R/W	Frame Composer ISRC1 Packet Body Register 1 Value After Reset: 0x0

fc_iscr2_15

Description: Frame Composer ISRC2 Packet Body Register 15

Size: 8 bits

Offset: 0x10a3

Bits	Name	Attr	Description
7:0	fc_iscr2_15	R/W	Frame Composer ISRC2 Packet Body Register 15; configures the ISRC2 packet body of the ISRC2 packet Value After Reset: 0x0

fc_iscr2_14

Description: Frame Composer ISRC2 Packet Body Register 14
Size: 8 bits
Offset: 0x10a4

Bits	Name	Attr	Description
7:0	fc_iscr2_14	R/W	Frame Composer ISRC2 Packet Body Register 14 Value After Reset: 0x0

fc_iscr2_13
Description: Frame Composer ISRC2 Packet Body Register 13
Size: 8 bits
Offset: 0x10a5

Bits	Name	Attr	Description
7:0	fc_iscr2_13	R/W	Frame Composer ISRC2 Packet Body Register 13 Value After Reset: 0x0

fc_iscr2_12
Description: Frame Composer ISRC2 Packet Body Register 12
Size: 8 bits
Offset: 0x10a6

Bits	Name	Attr	Description
7:0	fc_iscr2_12	R/W	Frame Composer ISRC2 Packet Body Register 12 Value After Reset: 0x0

fc_iscr2_11
Description: Frame Composer ISRC2 Packet Body Register 11
Size: 8 bits
Offset: 0x10a7

Bits	Name	Attr	Description
7:0	fc_iscr2_11	R/W	Frame Composer ISRC2 Packet Body Register 11 Value After Reset: 0x0

fc_iscr2_10
Description: Frame Composer ISRC2 Packet Body Register 10
Size: 8 bits
Offset: 0x10a8

Bits	Name	Attr	Description
7:0	fc_iscr2_10	R/W	Frame Composer ISRC2 Packet Body Register 10 Value After Reset: 0x0

fc_iscr2_9
Description: Frame Composer ISRC2 Packet Body Register 9
Size: 8 bits
Offset: 0x10a9

Bits	Name	Attr	Description
7:0	fc_iscr2_9	R/W	Frame Composer ISRC2 Packet Body Register 9 Value After Reset: 0x0

fc_iscr2_8
Description: Frame Composer ISRC2 Packet Body Register 8
Size: 8 bits
Offset: 0x10aa

Bits	Name	Attr	Description
7:0	fc_iscr2_8	R/W	Frame Composer ISRC2 Packet Body Register 8 Value After Reset: 0x0

fc_iscr2_7
Description: Frame Composer ISRC2 Packet Body Register 7
Size: 8 bits
Offset: 0x10ab

Bits	Name	Attr	Description
7:0	fc_iscr2_7	R/W	Frame Composer ISRC2 Packet Body Register 7 Value After Reset: 0x0

fc_iscr2_6
Description: Frame Composer ISRC2 Packet Body Register 6
Size: 8 bits
Offset: 0x10ac

Bits	Name	Attr	Description
7:0	fc_iscr2_6	R/W	Frame Composer ISRC2 Packet Body Register 6 Value After Reset: 0x0

fc_iscr2_5
Description: Frame Composer ISRC2 Packet Body Register 5
Size: 8 bits
Offset: 0x10ad

Bits	Name	Attr	Description
7:0	fc_iscr2_5	R/W	Frame Composer ISRC2 Packet Body Register 5 Value After Reset: 0x0

fc_iscr2_4
Description: Frame Composer ISRC2 Packet Body Register 4
Size: 8 bits
Offset: 0x10ae

Bits	Name	Attr	Description
7:0	fc_iscr2_4	R/W	Frame Composer ISRC2 Packet Body Register 4 Value After Reset: 0x0

fc_iscr2_3
Description: Frame Composer ISRC2 Packet Body Register 3
Size: 8 bits
Offset: 0x10af

Bits	Name	Attr	Description

7:0	fc_iscr2_3	R/W	Frame Composer ISRC2 Packet Body Register 3 Value After Reset: 0x0

fc_iscr2_2

Description: Frame Composer ISRC2 Packet Body Register 2

Size: 8 bits

Offset: 0x10b0

Bits	Name	Attr	Description
7:0	fc_iscr2_2	R/W	Frame Composer ISRC2 Packet Body Register 2 Value After Reset: 0x0

fc_iscr2_1

Description: Frame Composer ISRC2 Packet Body Register 1

Size: 8 bits

Offset: 0x10b1

Bits	Name	Attr	Description
7:0	fc_iscr2_1	R/W	Frame Composer ISRC2 Packet Body Register 1 Value After Reset: 0x0

fc_iscr2_0

Description: Frame Composer ISRC2 Packet Body Register 0

Size: 8 bits

Offset: 0x10b2

Bits	Name	Attr	Description
7:0	fc_iscr2_0	R/W	Frame Composer ISRC2 Packet Body Register 0 Value After Reset: 0x0

fc_datauto0

Description: Frame Composer Data Island Auto Packet Scheduling Register 0

Configures the Frame Composer RDRB(1)/Manual(0) data island packet insertion for SPD, VSD, ISRC2, ISRC1 and ACP packets. On RDRB mode the described packet scheduling is controlled by registers FC_DATAUTO1 and FC_DATAUTO2, while in Manual mode register FC_DATMAN requests to FC the insertion of the requested packet.

Size: 8 bits

Offset: 0x10b3

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	spd_auto	R/W	Enables SPD automatic packet scheduling Value After Reset: 0x0
3	vsd_auto	R/W	Enables VSD automatic packet scheduling Value After Reset: 0x0
2	iscr2_auto	R/W	Enables ISRC2 automatic packet scheduling Value After Reset: 0x0
1	iscr1_auto	R/W	Enables ISRC1 automatic packet scheduling Value After Reset: 0x0
0	acp_auto	R/W	Enables ACP automatic packet scheduling

			Value After Reset: 0x0

fc_datauto1

Description: Frame Composer Data Island Auto Packet Scheduling Register 1
Configures the Frame Composer (FC) RDRB frame interpolation for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets.
Size: 8 bits
Offset: 0x10b4

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	auto_frame_interpolation	R/W	Packet frame interpolation for automatic packet scheduling Value After Reset: 0x0

fc_datauto2

Description: Frame Composer Data Island Auto packet scheduling Register 2
Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets.
Size: 8 bits
Offset: 0x10b5

Bits	Name	Attr	Description
7:4	auto_frame_packets	R/W	Packets per frame, for automatic packet scheduling Value After Reset: 0x0
3:0	auto_line_spacing	R/W	Packets line spacing, for automatic packet scheduling Value After Reset: 0x0

fc_datman

Description: Frame Composer Data Island Manual Packet Request Register
Requests to the Frame Composer the data island packet insertion for NULL, SPD, VSD, ISRC2, ISRC1 and ACP packets when FC_DATAUTO0 bit is in manual mode for the packet requested.
Size: 8 bits
Offset: 0x10b6

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	null_tx	W	Null packet Value After Reset: 0x0
4	spd_tx	W	SPD packet Value After Reset: 0x0
3	vsd_tx	W	VSD packet Value After Reset: 0x0
2	iscr2_tx	W	ISRC2 packet Value After Reset: 0x0

1	iscr1_tx	W	ISRC1 packet Value After Reset: 0x0
0	acp_tx	W	ACP packet Value After Reset: 0x0

fc_datauto3

Description: Frame Composer Data Island Auto Packet Scheduling Register 3
Configures the Frame Composer Automatic(1)/RDRB(0) data island packet insertion for AVI, GCP, AUDI and ACR packets. In Automatic mode, the packet is inserted on Vblanking when first line with active Vsync appears.
Size: 8 bits
Offset: 0x10b7

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	nvbi_auto	R/W	Enables NTSC VBI packet insertion Value After Reset: 0x1
4	amp_auto	R/W	Enables AMP packet insertion Value After Reset: 0x1
3	avi_auto	R/W	Enables AVI packet insertion Value After Reset: 0x1
2	gcp_auto	R/W	Enables GCP packet insertion Value After Reset: 0x1
1	audi_auto	R/W	Enables AUDI packet insertion Value After Reset: 0x1
0	acr_auto	R/W	Enables ACR packet insertion Value After Reset: 0x1

fc_rdrb0

Description: Frame Composer Round Robin ACR Packet Insertion Register 0
Configures the Frame Composer (FC) RDRB frame interpolation for ACR packet insertion on data island when FC is on RDRB mode for this packet.
Size: 8 bits
Offset: 0x10b8

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	ACRframeinterpolation	R/W	ACR Frame interpolation Value After Reset: 0x0

fc_rdrb1

Description: Frame Composer Round Robin ACR Packet Insertion Register 1
Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the ACR packet insertion on data island when FC is on RDRB mode this packet.
Size: 8 bits
Offset: 0x10b9

Bits	Name	Attr	Description
7:4	ACRpacketsinframe	R/W	ACR packets in frame Value After Reset: 0x0
3:0	ACRpacketlinespacing	R/W	ACR packet line spacing Value After Reset: 0x0

fc_rdrb2

Description: Frame Composer Round Robin AUDI Packet Insertion Register 2
Configures the Frame Composer (FC) RDRB frame interpolation for AUDI packet insertion on data island when FC is on RDRB mode for this packet.
Size: 8 bits
Offset: 0x10ba

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	AUDIframeinterpolation	R/W	Audio frame interpolation Value After Reset: 0x0

fc_rdrb3
Description: Frame Composer Round Robin AUDI Packet Insertion Register 3
Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AUDI packet insertion on data island when FC is on RDRB mode this packet.
Size: 8 bits
Offset: 0x10bb

Bits	Name	Attr	Description
7:4	AUDIpacketsinframe	R/W	Audio packets per frame Value After Reset: 0x0
3:0	AUDIpacketlinespacing	R/W	Audio packets line spacing Value After Reset: 0x0

fc_rdrb4
Description: Frame Composer Round Robin GCP Packet Insertion Register 4
Configures the Frame Composer (FC) RDRB frame interpolation for GCP packet insertion on data island when FC is on RDRB mode for this packet.
Size: 8 bits
Offset: 0x10bc

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	GCPframeinterpolation	R/W	Frames interpolated between GCP packets Value After Reset: 0x0

fc_rdrb5
Description: Frame Composer Round Robin GCP Packet Insertion Register 5
Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the GCP packet insertion on data island when FC is on RDRB mode this packet.
Size: 8 bits
Offset: 0x10bd

Bits	Name	Attr	Description
7:4	GCPpacketsinframe	R/W	GCP packets per frame Value After Reset: 0x0
3:0	GCPpacketlinespacing	R/W	GCP packets line spacing Value After Reset: 0x0

fc_rdrb6
Description: Frame Composer Round Robin AVI Packet Insertion Register 6
Configures the Frame Composer (FC) RDRB frame interpolation for AVI packet insertion on data island when FC is on RDRB mode for this packet.
Size: 8 bits
Offset: 0x10be

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	AVIframeinterpolation	R/W	Frames interpolated between AVI packets Value After Reset: 0x0

fc_rdrb7

Description: Frame Composer Round Robin AVI Packet Insertion Register 7
Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AVI packet insertion on data island when FC is on RDRB mode this packet.
Size: 8 bits
Offset: 0x10bf

Bits	Name	Attr	Description
7:4	AVIpacketsinframe	R/W	AVI packets per frame Value After Reset: 0x0
3:0	AVIpacketlinespacing	R/W	AVI packets line spacing Value After Reset: 0x0

fc_rdrb8

Description: Frame Composer Round Robin AMP Packet Insertion Register 8
Size: 8 bits
Offset: 0x10c0

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	AMPframeinterpolation	R/W	AMP frame interpolation Value After Reset: 0x0

fc_rdrb9

Description: Frame Composer Round Robin AMP Packet Insertion Register 9
Size: 8 bits
Offset: 0x10c1

Bits	Name	Attr	Description
7:4	AMPpacketsinframe	R/W	AMP packets per frame Value After Reset: 0x0
3:0	AMPpacketlinespacing	R/W	AMP packets line spacing Value After Reset: 0x0

fc_rdrb10

Description: Frame Composer Round Robin NTSC VBI Packet Insertion Register 10
Size: 8 bits
Offset: 0x10c2

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	NVBIframeinterpolation	R/W	NTSC VBI frame interpolation Value After Reset: 0x0

fc_rdrb11

Description: Frame Composer Round Robin NTSC VBI Packet Insertion Register 11
Size: 8 bits
Offset: 0x10c3

Bits	Name	Attr	Description
7:4	NVBIpacketsinframe	R/W	NTSC VBI packets per frame Value After Reset: 0x0
3:0	NVBIpacketlinespacing	R/W	NTSC VBI packets line spacing Value After Reset:

			0x0

fc_rdrb12

Description: Frame Composer Round Robin DRM Packet Insertion Register 12

Size: 8 bits

Offset: 0x10c4

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	DRMframeinterpolation	R/W	Description: DRM frame interpolation Value After Reset: 0x0

fc_rdrb13

Description: Frame Composer Round Robin DRM Packet Insertion Register 13

Size: 8 bits

Offset: 0x10c5

Bits	Name	Attr	Description
7:4	DRMpacketsinframe	R/W	DRM packets per frame Value After Reset: 0x0
3:0	DRMpacketlinespacing	R/W	DRM packets line spacing Value After Reset: 0x0

fc_mask0

Description: Frame Composer Packet Interrupt Mask Register 0

Size: 8 bits

Offset: 0x10d2

Bits	Name	Attr	Description
7	AUDI	R/W	Mask bit for FC_INT0.AUDI interrupt bit Value After Reset: 0x0
6	ACP	R/W	Mask bit for FC_INT0.ACP interrupt bit Value After Reset: 0x0
5	HBR	R/W	Mask bit for FC_INT0.HBR interrupt bit Value After Reset: 0x1
4	MAS	R/W	Mask bit for FC_INT0.MAS interrupt bit. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0"
3	NVBI	R/W	Mask bit for FC_INT0.NVBI interrupt bit. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0"
2	AUDS	R/W	Mask bit for FC_INT0.AUDS interrupt bit Value After Reset: 0x1
1	ACR	R/W	Mask bit for FC_INT0.ACR interrupt bit Value After Reset: 0x0
0	NULL	R/W	Mask bit for FC_INT0.NULL interrupt bit Value After Reset: 0x1

fc_mask1

Description: Frame Composer Packet Interrupt Mask Register 1
Size: 8 bits
Offset: 0x10d6

Bits	Name	Attr	Description
7	GMD	R/W	Mask bit for FC_INT1.GMD interrupt bit Value After Reset: 0x0
6	ISCR1	R/W	Mask bit for FC_INT1.ISRC1 interrupt bit Value After Reset: 0x0
5	ISCR2	R/W	Mask bit for FC_INT1.ISRC2 interrupt bit Value After Reset: 0x0
4	VSD	R/W	Mask bit for FC_INT1.VSD interrupt bit Value After Reset: 0x0
3	SPD	R/W	Mask bit for FC_INT1.SPD interrupt bit Value After Reset: 0x0
2	AMP	R/W	Mask bit for FC_INT1.AMP interrupt bit. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: "(HDMI_TX_20== 1) ? 1 : 0"
1	AVI	R/W	Mask bit for FC_INT1.AVI interrupt bit Value After Reset: 0x0
0	GCP	R/W	Mask bit for FC_INT1.GCP interrupt bit Value After Reset: 0x0

fc_mask2
Description: Frame Composer High/Low Priority Overflow and DRM Interrupt Mask Register 2
Size: 8 bits
Offset: 0x10da

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	DRM	R/W	Mask bit for FC_INT2.DRM interrupt bit. Value After Reset: (HDMI_TX_20== 1) ? 1 : 0
3:2			Reserved for future use.
1	LowPriority_overflow	R/W	Mask bit for FC_INT2.LowPriority_overflow interrupt bit Value After Reset: 0x0
0	HighPriority_overflow	R/W	Mask bit for FC_INT2.HighPriority_overflow interrupt bit Value After Reset: 0x0

fc_prconf
Description: Frame Composer Pixel Repetition Configuration Register
Size: 8 bits
Offset: 0x10e0

Bits	Name	Attr	Description
7:4	incoming_pr_factor	R/W	Configures the input video pixel repetition. For CEA

Bits	Name	Attr	Description
			modes, this value must be extracted from the CEA specification for the video mode being input. incoming_pr_factor[3:0] 0000b: No action. Not used. 0001b: No pixel repetition (pixel sent only once) 0010b: Pixel sent two times (pixel repeated once) 0011b: Pixel sent three times 0100b: Pixel sent four times 0101b: Pixel sent five times 0110b: Pixel sent six times 0111b: Pixel sent seven times 1000b: Pixel sent eight times 1001b: Pixel sent nine times 1010b: Pixel sent 10 times Other: Reserved. Not used Value After Reset: 0x1
3:0	output_pr_factor	R/W	Configures the video pixel repetition ratio to be sent on the AVI InfoFrame. This value must be valid according to the HDMI specification. The output_pr_factor = incoming_pr_factor * (desired_pr_factor + 1) – 1. output_pr_factor[3:0] 0000b: No action. Not used. 0001b: Pixel sent two times (pixel repeated once) 0010b: Pixel sent three times 0011b: Pixel sent four times 0100b: Pixel sent five times 0101b: Pixel sent six times 0110b: Pixel sent seven times 0111b: Pixel sent eight times 1000b: Pixel sent nine times 1001b: Pixel sent 10 times Other: Reserved. Not used Note: When working in YCC422 video, the actual repetition of the stream is Incoming_pr_factor * (desired_pr_factor + 1). This calculation is done internally in the H13TCTRL and no hardware overflow protection is available. Care must be taken to avoid this result passes the maximum number of 10 pixels repeated because no HDMI support is available for this in the specification and the H13TPHY does not support this higher repetition values. Value After Reset: 0x0

fc_scrambler_ctrl

Description: Frame Composer Scrambler Control

Size: 8 bits

Offset: 0x10e1

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	scrambler_ucp_line	R/W	Debug register. When active (1'b1), the Unscrambled Control Period is generated after each active video line (non-compliant behavior). This is

			quasi-static field which requires a a mc_swrstzreq.tmdsswrst_req reset request to be performed after the change of this configuration bit. Value After Reset: 0x0
3:1			Reserved for future use.
0	scrambler_on	R/W	When set (1'b1), this field activates the HDMI 2.0 scrambler feature. When disabled (1'b0) the scrambler feature is bypassed, placing Hdmi_tx in HDMI 1.4b compatible mode. To activate the scrambler feature, you must ensure that the quasi-static configuration bit fc_invidconf.HDCP_keepout is set (1'b1) at configuration time, before the required mc_swrstzreq.tmdsswrst_req reset request is issued. This is field can be changed in runtime. Value After Reset: 0x0

fc_multistream_ctrl

Description: Frame Composer Multi-Stream Audio Control

Size: 8 bits

Offset: 0x10e2

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	fc_mas_packet_en	R/W	This field, when set (1'b1), activates the HDMI 2.0 Multi- Stream support. The audio stream present at the input of the Hdmi_tx controller is transported using Multi-Stream Audio Sample Packets. Value After Reset: 0x0

fc_packet_tx_en

Description: Frame Composer Packet Transmission Control

Size: 8 bits

Offset: 0x10e3

Bits	Name	Attr	Description
7	drm_tx_en	R/W	DRM transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x0
6	nvbi_tx_en	R/W	NTSC VBI transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x0
5	amp_tx_en	R/W	AMP transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x0
4	aut_tx_en	R/W	ACP, SPD, VSIF, ISRC1, and SRC2 packet transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x1

3	audi_tx_en	R/W	AUDI packet transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x1
2	avi_tx_en	R/W	AVI packet transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x1
1	gcp_tx_en	R/W	GCP transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x1
0	acr_tx_en	R/W	ACR packet transmission control 1b: Transmission enabled 0b: Transmission disabled Value After Reset: 0x1

fc_actspc_hdlr_cfg

Description: Frame Composer Active Space Control

Size: 8 bits

Offset: 0x10e8

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	actspc_hdlr_tgl	R/W	Active Space handler control 1b: Active space 1 value is different from Active Space 2 value. Refer to Figure 8-4: 3D Structure of the HDMI 1.4b specification. 0b: Active space not oscillating Value After Reset: 0x0
0	actspc_hdlr_en	R/W	Active Space Handler Control 1b: Fixed active space value mode enabled. During active space, a fixed value of 0xAA is applied to all TMDS channels. 0b: Fixed active space value mode disabled Value After Reset: 0x0

fc_invact_2d_0

Description: Frame Composer Input Video 2D VActive Pixels Register 0

Size: 8 bits

Offset: 0x10e9

Bits	Name	Attr	Description
7:0	fc_invact_2d_0	R/W	2D Input video vertical active pixel region width. Number of 2D video vertical active lines [7:0]. Value After Reset: 0x0

fc_invact_2d_1

Description: Frame Composer Input Video VActive pixels Register 1

Size: 8 bits

Offset: 0x10ea

Bits	Name	Attr	Description
7:4			Reserved for future use.

3:0	fc_invact_2d_1	R/W	2D Input video vertical active pixel region width. Number of 2D video vertical active lines [11:8]. Value After Reset: 0x0

fc_gmd_stat

Description: Frame Composer GMD Packet Status Register

Gamut metadata packet status bit information for no_current_gmd, next_gmd_field, gmd_packet_sequence and current_gamut_seq_num. For more information, refer to the HDMI 1.4b specification.

Size: 8 bits

Offset: 0x1100

Bits	Name	Attr	Description
7	igmdno_crnt_gbd	R	Gamut scheduling: No current gamut data Value After Reset: 0x0
6	igmddnext_field	R	Gamut scheduling: Gamut Next field Value After Reset: 0x0
5:4	igmdpacket_seq	R	Gamut scheduling: Gamut packet sequence Value After Reset: 0x0
3:0	igmdcurrent_gamut_seq_num	R	Gamut scheduling: Current Gamut packet sequence number Value After Reset: 0x0

fc_gmd_en

Description: Frame Composer GMD Packet Enable Register

This register enables Gamut metadata (GMD) packet transmission. Packets are inserted in the incoming frame, starting in the line where active Vsync indication starts. After enable of GMD packets the outgoing packet is sent with no_current_gmd active indication until update GMD request is performed in the controller.

Size: 8 bits

Offset: 0x1101

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	gmdenabletx	R/W	Gamut Metadata packet transmission enable (1b) Value After Reset: 0x0

fc_gmd_up

Description: Frame Composer GMD Packet Update Register

This register performs an GMD packet content update according to the configured packet body (FC_GMD_PB0 to FC_GMD_PB27) and packet header (FC_GMD_HB). This active high auto clear register reflects the body and header configurations on the GMD packets sent arbitrating the current_gamut_seq_num, gmd_packet_sequence and next_gmd_field bits on packet to correctly indicate to source the Gamut change to be performed. After enable GMD packets the first update request is also responsible for deactivating the no_current_gmd indication bit.

Attention packet update request must only be done after correct configuration of GMD packet body and header registers. Correct affected_gamut_seq_num and gmd_profile configuration is user responsibility and must convey with HDMI 1.4b standard gamut rules.

Size: 8 bits
Offset: 0x1102

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	gmdupdatepacket	W	Gamut Metadata packet update Value After Reset: 0x0

fc_gmd_conf

Description: Frame Composer GMD Packet Schedule Configuration Register
This register configures the number of GMD packets to be inserted per frame (starting always in the line where the active Vsync appears) and the line spacing between the transmitted GMD packets.
Note that for profile P0 (refer to the HDMI 1.4b specification) this register should only indicate one GMD packet to be inserted per video field.
Size: 8 bits
Offset: 0x1103

Bits	Name	Attr	Description
7:4	gmdpacketsinframe	R/W	Number of GMD packets per frame or video field (profile P0) Value After Reset: 0x1
3:0	gmdpacketlinespacing	R/W	Number of line spacing between the transmitted GMD packets Value After Reset: 0x0

fc_gmd_hb

Description: Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register
This register configures the GMD packet header affected_gamut_seq_num and gmd_profile bits. For more information, refer to the HDMI 1.4b specification.
Size: 8 bits
Offset: 0x1104

Bits	Name	Attr	Description
7			Reserved for future use.
6:4	gmdgbd_profile	R/W	GMD profile bits. Hdmi_tx only supports Profile 0 (P0) of the Gamut Boundary Description Metadata Profiles described in the HDMI 1.4 Specification (which defines four profiles, P0-P4). Value After Reset: 0x0
3:0	gmdaffected_gamut_seq_num	R/W	Affected gamut sequence number Value After Reset: 0x0

fc_gmd_pb[0:27]

Description: Frame Composer GMD Packet Body Register Array Configures the GMD packet body of the GMD packet.
Size: 8 bits
Offset: 0x1105 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_gmd_pb	R/W	Frame Composer GMD Packet Body Register Array Value After Reset: 0x0

fc_amp_hb1

Description: Frame Composer AMP Packet Header Register 1

Size: 8 bits

Offset: 0x1128

Bits	Name	Attr	Description
7:0	fc_amp_hb0	R/W	Frame Composer AMP Packet Header Register 1 Value After Reset: 0x0

fc_amp_hb2

Description: Frame Composer AMP Packet Header Register 2

Size: 8 bits

Offset: 0x1129

Bits	Name	Attr	Description
7:0	fc_amp_hb1	R/W	Frame Composer AMP Packet Header Register 2 Value After Reset: 0x0

fc_amp_pb[0:27]

Description: Frame Composer AMP Packet Body Register Array

Size: 8 bits

Offset: 0x112a + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_amp_pb	R/W	Frame Composer AMP Packet Body Register Array Value After Reset: 0x0

fc_nvbi_hb1

Description: Frame Composer NTSC VBI Packet Header Register 1

Size: 8 bits

Offset: 0x1148

Bits	Name	Attr	description
7:0	fc_nvbi_hb0	R/W	Frame Composer NTSC VBI Packet Header Register 1 Value After Reset: 0x0

fc_nvbi_hb2

Description: Frame Composer NTSC VBI Packet Header Register 2

Size: 8 bits

Offset: 0x1149

Bits	Name	Attr	Description
7:0	fc_nvbi_hb1	R/W	Frame Composer NTSC VBI Packet Header Register 2 Value After Reset: 0x0

fc_nvbi_pb[0:26]

Description: Frame Composer NTSC VBI Packet Body Register Array

Size: 8 bits
Offset: 0x114a + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_nvbi_pb	R/W	Frame Composer NTSC VBI Packet Body Register Array Value After Reset: 0x0

fc_drm_up
Description: Frame Composer DRM Packet Update Register
This register performs an DRM packet content update according to the configured packet body (FC_DRM_PB0 to FC_DRM_PB27) and packet header (FC_DRM_HB). This active high auto clear register reflects the body and header configurations on the DRM packets change to be performed.
Attention packet update request must only be done after correct configuration of DRM packet body and header registers.
Size: 8 bits
Offset: 0x1167

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	drmpacketupdate	W	DRM packet update Value After Reset: 0x0

fc_drm_hb[0:1]
Description: Frame Composer DRM Packet Header Register Array
Size: 8 bits
Offset: 0x1168 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_drm_hb	R/W	Frame Composer DRM Packet Header Register Array Value After Reset: 0x0

fc_drm_pb[0:26]
Description: Frame Composer DRM Packet Body Register Array
Size: 8 bits
Offset: 0x116a + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_drm_pb	R/W	Frame Composer DRM Packet Body Register Array Value After Reset: 0x0

fc_dbgforce
Description: Frame Composer video/audio Force Enable Register
This register allows to force the controller to output audio and video data the values configured in the FC_DBGAUD and FC_DBGTMDS registers.
Size: 8 bits
Offset: 0x1200

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	forceaudio	R/W	Force fixed audio output with FC_DBGAUDxCHx

			register contents. Value After Reset: 0x0
3:1			Reserved for future use.
0	forcevideo	R/W	Force fixed video output with FC_DBGTMDSx register contents. Value After Reset: 0x0

fc_dbgaud0ch0

Description: Frame Composer Audio Data Channel 0 Register 0

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

Size: 8 bits

Offset: 0x1201

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch0	R/W	Frame Composer Audio Data Channel 0 Register 0 Value After Reset: 0x0

fc_dbgaud1ch0

Description: Frame Composer Audio Data Channel 0 Register 1

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

Size: 8 bits

Offset: 0x1202

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch0	R/W	Frame Composer Audio Data Channel 0 Register 1 Value After Reset: 0x0

fc_dbgaud2ch0

Description: Frame Composer Audio Data Channel 0 Register 2

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

Size: 8 bits

Offset: 0x1203

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch0	R/W	Frame Composer Audio Data Channel 0 Register 2 Value After Reset: 0x0

fc_dbgaud0ch1

Description: Frame Composer Audio Data Channel 1 Register 0

Configures the audio fixed data to be used in channel 1 when in fixed audio selection.

Size: 8 bits

Offset: 0x1204

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch1	R/W	Frame Composer Audio Data Channel 1 Register 0 Value After Reset: 0x0

fc_dbgaud1ch1

Description: Frame Composer Audio Data Channel 1 Register 1

Configures the audio fixed data to be used in channel 1 when in fixed audio selection.

Size: 8 bits

Offset: 0x1205

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch1	R/W	Frame Composer Audio Data Channel 1 Register 1 Value After Reset: 0x0

fc_dbgaud2ch1

Description: Frame Composer Audio Data Channel 1 Register 2

Configures the audio fixed data to be used in channel 1 when in fixed audio selection.

Size: 8 bits

Offset: 0x1206

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch1	R/W	Frame Composer Audio Data Channel 1 Register 2 Value After Reset: 0x0

fc_dbgaud0ch2

Description: Frame Composer Audio Data Channel 2 Register 0

Configures the audio fixed data to be used in channel 2 when in fixed audio selection.

Size: 8 bits

Offset: 0x1207

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch2	R/W	Frame Composer Audio Data Channel 2 Register 0 Value After Reset: 0x0

fc_dbgaud1ch2

Description: Frame Composer Audio Data Channel 2 Register 1

Configures the audio fixed data to be used in channel 2 when in fixed audio selection.

Size: 8 bits Offset: 0x1208

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch2	R/W	Frame Composer Audio Data Channel 2 Register 1 Value After Reset: 0x0

fc_dbgaud2ch2

Description: Frame Composer Audio Data Channel 2 Register 2

Configures the audio fixed data to be used in channel 2 when in fixed audio selection.

Size: 8 bits

Offset: 0x1209

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch2	R/W	Frame Composer Audio Data Channel 2 Register 2 Value After Reset: 0x0

fc_dbgaud0ch3

Description: Frame Composer Audio Data Channel 3 Register 0

Configures the audio fixed data to be used in channel 3 when in fixed audio selection.

Size: 8 bits

Offset: 0x120a

Bits	Name	Attr	Description

7:0	fc_dbgaud0ch3	R/W	Frame Composer Audio Data Channel 3 Register 0 Value After Reset: 0x0

fc_dbgaud1ch3

Description: Frame Composer Audio Data Channel 3 Register 1

Configures the audio fixed data to be used in channel 3 when in fixed audio selection.

Size: 8 bits

Offset: 0x120b

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch3	R/W	Frame Composer Audio Data Channel 3 Register 1 Value After Reset: 0x0

fc_dbgaud2ch3

Description: Frame Composer Audio Data Channel 3 Register 2

Configures the audio fixed data to be used in channel 3 when in fixed audio selection.

Size: 8 bits

Offset: 0x120c

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch3	R/W	Frame Composer Audio Data Channel 3 Register 2 Value After Reset: 0x0

fc_dbgaud0ch4

Description: Frame Composer Audio Data Channel 4 Register 0

Configures the audio fixed data to be used in channel 4 when in fixed audio selection.

Size: 8 bits

Offset: 0x120d

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch4	R/W	Frame Composer Audio Data Channel 4 Register 0 Value After Reset: 0x0

fc_dbgaud1ch4

Description: Frame Composer Audio Data Channel 4 Register 1

Configures the audio fixed data to be used in channel 4 when in fixed audio selection.

Size: 8 bits

Offset: 0x120e

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch4	R/W	Frame Composer Audio Data Channel 4 Register 1 Value After Reset: 0x0

fc_dbgaud2ch4

Description: Frame Composer Audio Data Channel 4 Register 2

Configures the audio fixed data to be used in channel 4 when in fixed audio selection.

Size: 8 bits

Offset: 0x120f

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch4	R/W	Frame Composer Audio Data Channel 4 Register 2 Value After Reset: 0x0

fc_dbgaud0ch5

Description: Frame Composer Audio Data Channel 5 Register 0

Configures the audio fixed data to be used in channel 5 when in fixed audio selection.

Size: 8 bits

Offset: 0x1210

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch5	R/W	Frame Composer Audio Data Channel 5 Register 0 Value After Reset: 0x0

fc_dbgaud1ch5

Description: Frame Composer Audio Data Channel 5 Register 1

Configures the audio fixed data to be used in channel 5 when in fixed audio selection.

Size: 8 bits

Offset: 0x1211

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch5	R/W	Frame Composer Audio Data Channel 5 Register 1 Value After Reset: 0x0

fc_dbgaud2ch5

Description: Frame Composer Audio Data Channel 5 Register 2

Configures the audio fixed data to be used in channel 5 when in fixed audio selection.

Size: 8 bits

Offset: 0x1212

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch5	R/W	Frame Composer Audio Data Channel 5 Register 2 Value After Reset: 0x0

fc_dbgaud0ch6

Description: Frame Composer Audio Data Channel 6 Register 0

Configures the audio fixed data to be used in channel 6 when in fixed audio selection.

Size: 8 bits

Offset: 0x1213

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch6	R/W	Frame Composer Audio Data Channel 6 Register 0 Value After Reset: 0x0

fc_dbgaud1ch6

Description: Frame Composer Audio Data Channel 6 Register 1

Configures the audio fixed data to be used in channel 6 when in fixed audio selection.

Size: 8 bits

Offset: 0x1214

Bits	Name	Attr	Description

7:0	fc_dbgaud1ch6	R/W	Frame Composer Audio Data Channel 6 Register 1 Value After Reset: 0x0

fc_dbgaud2ch6

Description: Frame Composer Audio Data Channel 6 Register 2

Configures the audio fixed data to be used in channel 6 when in fixed audio selection.

Size: 8 bits

Offset: 0x1215

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch6	R/W	Frame Composer Audio Data Channel 6 Register 2 Value After Reset: 0x0

fc_dbgaud0ch7

Description: Frame Composer Audio Data Channel 7 Register 0

Configures the audio fixed data to be used in channel 7 when in fixed audio selection.

Size: 8 bits

Offset: 0x1216

Bits	Name	Attr	Description
7:0	fc_dbgaud0ch7	R/W	Frame Composer Audio Data Channel 7 Register 0 Value After Reset: 0x0

fc_dbgaud1ch7

Description: Frame Composer Audio Data Channel 7 Register 1

Configures the audio fixed data to be used in channel 7 when in fixed audio selection.

Size: 8 bits

Offset: 0x1217

Bits	Name	Attr	Description
7:0	fc_dbgaud1ch7	R/W	Frame Composer Audio Data Channel 7 Register 1 Value After Reset: 0x0

fc_dbgaud2ch7

Description: Frame Composer Audio Data Channel 7 Register 2

Configures the audio fixed data to be used in channel 7 when in fixed audio selection.

Size: 8 bits

Offset: 0x1218

Bits	Name	Attr	Description
7:0	fc_dbgaud2ch7	R/W	Frame Composer Audio Data Channel 7 Register 2 Value After Reset: 0x0

fc_dbgtmds[0:2]

Description: Frame Composer TMDS Data Channel Register Array

Configures the video fixed data to be used in TMDS channel x (where x is 0 to 2) when in fixed video selection.

For Channel 0, this equals to set B pixel component value in RGB video or Cb pixel component value in YCbCr.

For Channel 1, this equals set G pixel component value in RGB video or Y pixel component value in YCbCr.
For Channel 2, this equals to set R pixel component value in RGB video or Cr pixel component value in YCbCr.
Size: 8 bits
Offset: 0x1219 + (i * 0x1)

Bits	Name	Attr	Description
7:0	fc_dbgtmds	R/W	Frame Composer TMDS Data Channel 0 Register Value After Reset: 0x0

PHYConfiguration Registers

PHY Configuration Registers. Follow the link for the register to see a detailed description of the register.
Registers for Address Block: PHYConfiguration

Register	Offset	Description
phy_conf0	0x3000	PHY Configuration Register This register holds the power down, data enable polarity, and interface...
phy_tst0	0x3001	PHY Test Interface Register 0 PHY TX mapped test interface (control). For more information, refer...
phy_tst1	0x3002	PHY Test Interface Register 1 PHY TX mapped text interface (data in). For more information, refer...
phy_tst2	0x3003	PHY Test Interface Register 2 PHY TX mapped text interface (data out). For more information, refer...
phy_stat0	0x3004	PHY RXSENSE, PLL Lock, and HPD Status Register This register contains the following active high...
phy_int0	0x3005	PHY RXSENSE, PLL Lock, and HPD Interrupt Register This register contains the interrupt indication...
phy_mask0	0x3006	PHY RXSENSE, PLL Lock, and HPD Mask Register Mask register for generation of PHY_INT0...
phy_pol0	0x3007	PHY RXSENSE, PLL Lock, and HPD Polarity Register Polarity register for generation of PHY_INT0...
PHY_PCLFREQ0	0x3008	PHY Test Interface Register 0
PHY_PCLFREQ1	0x3009	PHY Test Interface Register 1
PHY_PLLCFGFREQ0	0x300a	PHY PLL Test Interface Register 0
PHY_PLLCFGFREQ1	0x300b	PHY PLL Test Interface Register 1
PHY_PLLCFGFREQ2	0x300c	PHY PLL Test Interface Register 2
phy_i2cm_slave	0x3020	PHY I2C Slave Address Configuration Register
phy_i2cm_address	0x3021	PHY I2C Address Configuration Register This register writes the address for read and write...
phy_i2cm_datao_1	0x3022	PHY I2C Data Write Register 1
phy_i2cm_datao_0	0x3023	PHY I2C Data Write Register 0
phy_i2cm_datai_1	0x3024	PHY I2C Data Read Register 1
phy_i2cm_datai_0	0x3025	PHY I2C Data Read Register 0
phy_i2cm_operation	0x3026	PHY I2C RD/RD_EXT/WR Operation Register This register requests read and write operations from the...

phy_i2cm_int	0x3027	PHY I2C Done Interrupt Register This register contains and configures I2C master PHY done...

Registers for Address Block: PHYConfiguration (Continued)

Register	Offset	Description
phy_i2cm_ctlint	0x3028	PHY I2C error Interrupt Register This register contains and configures the I2C master PHY error...
phy_i2cm_div	0x3029	PHY I2C Speed control Register This register wets the I2C Master PHY to work in either Fast or...
phy_i2cm_softrstz	0x302a	PHY I2C SW reset control register This register sets the I2C Master PHY software reset.
phy_i2cm_ss_scl_hcnt_1_addr	0x302b	PHY I2C Slow Speed SCL High Level Control Register 1
phy_i2cm_ss_scl_hcnt_0_addr	0x302c	PHY I2C Slow Speed SCL High Level Control Register 0
phy_i2cm_ss_scl_lcnt_1_addr	0x302d	PHY I2C Slow Speed SCL Low Level Control Register 1
phy_i2cm_ss_scl_lcnt_0_addr	0x302e	PHY I2C Slow Speed SCL Low Level Control Register 0
phy_i2cm_fs_scl_hcnt_1_addr	0x302f	PHY I2C Fast Speed SCL High Level Control Register 1
phy_i2cm_fs_scl_hcnt_0_addr	0x3030	PHY I2C Fast Speed SCL High Level Control Register 0
phy_i2cm_fs_scl_lcnt_1_addr	0x3031	PHY I2C Fast Speed SCL Low Level Control Register 1
phy_i2cm_fs_scl_lcnt_0_addr	0x3032	PHY I2C Fast Speed SCL Low Level Control Register 0
phy_i2cm_sda_hold	0x3033	PHY I2C SDA HOLD Control Register
jtag_phy_config	0x3034	PHY I2C/JTAG I/O Configuration Control Register
jtag_phy_tap_tck	0x3035	PHY JTAG Clock Control Register
jtag_phy_tap_in	0x3036	PHY JTAG TAP In Control Register
jtag_phy_tap_out	0x3037	PHY JTAG TAP Out Control Register
jtag_phy_addr	0x3038	PHY JTAG Address Control Register

phy_conf0

Description: PHY Configuration Register

This register holds the power down, data enable polarity, and interface control of the HDMI Source PHY control.

Size: 8 bits

Offset: 0x3000

Bits	Name	Attr	description
7	PDZ	R/W	Power-down enable (active low 0b). Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
6	ENTMDS	R/W	Enable TMDS drivers, bias, and TMDS digital logic. Otherwise, this field is a "spare" bit with no

			associated functionality. Value After Reset: 0x0
5	svsret	R/W	PHY SVSRET signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
4	pddq	R/W	PHY PDDQ signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: "(PHY_MHL_COMBO== 1) ? 1 : 0"
3	txpwron	R/W	PHY TXPWRON signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
2	enhpdrxsense	R/W	PHY ENHPDRXSENSE signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x1
1	seldataenpol	R/W	Select data enable polarity. Value After Reset: 0x1
0	seldipif	R/W	Select interface control. Value After Reset: 0x0

phy_tst0

Description: PHY Test Interface Register 0

PHY TX mapped test interface (control).

Size: 8 bits

Offset: 0x3001

Bits	Name	Attr	Description
7:6	spare_2	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
5	testclr	R/W	Test Clear signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
4	testen	R/W	Test Enable signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
3:1	spare_1	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
0	testclk	R/W	Test Clock signal. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0

phy_tst1

Description: PHY Test Interface Register 1

PHY TX mapped text interface (data in).

Size: 8 bits

Offset: 0x3002

Bits	Name	Attr	Description

7:0	testdin	R/W	Test Data input Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0

phy_tst2

Description: PHY Test Interface Register 2

PHY TX mapped text interface (data out).

Size: 8 bits

Offset: 0x3003

Bits	Name	Attr	Description
7:0	testdout	R	Test Data output. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0

phy_stat0

Description: PHY RXSENSE, PLL Lock, and HPD Status Register

This register contains the following active high packet sent status indications.

Size: 8 bits

Offset: 0x3004

Bits	Name	Attr	Description
7	RX_SENSE_3	R	Status bit. TX PHY RX_SENSE indication for TMDS channel 3 driver. You may need to mask or change polarity of this interrupt after it has became active. Value After Reset: 0x0
6	RX_SENSE_2	R	Status bit. TX PHY RX_SENSE indication for TMDS channel 2 driver. You may need to mask or change polarity of this interrupt after it has became active. Value After Reset: 0x0
5	RX_SENSE_1	R	Status bit. TX PHY RX_SENSE indication for TMDS channel 1 driver. You may need to mask or change polarity of this interrupt after it has became active. Value After Reset: 0x0
4	RX_SENSE_0	R	Status bit. TX PHY RX_SENSE indication for TMDS channel 0 driver. You may need to mask or change polarity of this interrupt after it has became active. Value After Reset: 0x0
3:2			Reserved for future use.
1	HPD	R	Status bit. HDMI Hot Plug Detect indication. You may need to mask or change polarity of this interrupt after it has became active. Value After Reset: 0x0
0	TX_PHY_LOCK	R	Status bit. TX PHY PLL lock indication. You may need to mask or change polarity of this interrupt after it has became active. Value After Reset: 0x0

phy_int0

Description: PHY RXSENSE, PLL Lock, and HPD Interrupt Register
This register contains the interrupt indication of the PHY_STAT0 status interrupts. Interrupt generation is accomplished in the following way:
interrupt = (mask == 1'b0) && (polarity == status);
All these interrupts are forwarded to the Interrupt Handler sticky bit register ih_phy_stat0 and after ORed to a single main interrupt line to micro controller. Assertion of this interrupt implies that data related with the corresponding packet has been sent through the HDMI interface.
Size: 8 bits
Offset: 0x3005

Bits	Name	Attr	Description
7	RX_SENSE_3	R	Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS CLK driver. Value After Reset: 0x0
6	RX_SENSE_2	R	Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS channel 2 driver. Value After Reset: 0x0
5	RX_SENSE_1	R	Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS channel 1 driver. Value After Reset: 0x0
4	RX_SENSE_0	R	Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS channel 0 driver. Value After Reset: 0x0
3:2			Reserved for future use.
1	HPD	R	Interrupt indication bit. HDMI Hot Plug Detect indication interrupt. Value After Reset: 0x0
0	TX_PHY_LOCK	R	Interrupt indication bit. TX PHY PLL lock indication interrupt. Value After Reset: 0x0

phy_mask0
Description: PHY RXSENSE, PLL Lock, and HPD Mask Register Mask register for generation of PHY_INT0 interrupts.
Size: 8 bits
Offset: 0x3006

Bits	Name	Attr	Description
7	RX_SENSE_3	R/W	Mask bit for PHY_INT0.RX_SENSE[3] interrupt bit Value After Reset: 0x0
6	RX_SENSE_2	R/W	Mask bit for PHY_INT0.RX_SENSE[2] interrupt bit Value After Reset: 0x0
5	RX_SENSE_1	R/W	Mask bit for PHY_INT0.RX_SENSE[1] interrupt bit Value After Reset: 0x0
4	RX_SENSE_0	R/W	Mask bit for PHY_INT0.RX_SENSE[0] interrupt bit Value After Reset: 0x0
3:2			Reserved for future use.

1	HPD	R/W	Mask bit for PHY_INT0.HPD interrupt bit Value After Reset: 0x0
0	TX_PHY_LOCK	R/W	Mask bit for PHY_INT0.TX_PHY_LOCK interrupt bit Value After Reset: 0x0

phy_pol0

Description: PHY RXSENSE, PLL Lock, and HPD Polarity Register Polarity register for generation of PHY_INT0 interrupts.

Size: 8 bits

Offset: 0x3007

Bits	Name	Attr	Description
7	RX_SENSE_3	R/W	Polarity bit for PHY_INT0.RX_SENSE[3] interrupt bit Value After Reset: 0x1
6	RX_SENSE_2	R/W	Polarity bit for PHY_INT0.RX_SENSE[2] interrupt bit Value After Reset: 0x1
5	RX_SENSE_1	R/W	Polarity bit for PHY_INT0.RX_SENSE[1] interrupt bit Value After Reset: 0x1
4	RX_SENSE_0	R/W	Polarity bit for PHY_INT0.RX_SENSE[0] interrupt bit Value After Reset: 0x1
3:2			Reserved for future use.
1	HPD	R/W	Polarity bit for PHY_INT0.HPD interrupt bit Value After Reset: 0x1
0	TX_PHY_LOCK	R/W	Polarity bit for PHY_INT0.TX_PHY_LOCK interrupt bit Value After Reset: 0x1

PHY_PCLFREQ0

Description: PHY Test Interface Register 0

Size: 8 bits

Offset: 0x3008

Bits	Name	Attr	Description
7:0	pclk_freq	R/W	Pixel Clock Frequency (pclk_freq[7:0]). Value After Reset: 0x32

PHY_PCLFREQ1

Description: PHY Test Interface Register 1

Size: 8 bits

Offset: 0x3009

Bits	Name	Attr	Description
7:2			Reserved for future use.
1:0	pclk_freq	R/W	Pixel Clock Frequency (pclk_freq[9:8]). Value After Reset: 0x0

PHY_PLLCFGFREQ0

Description: PHY PLL Test Interface Register 0

Size: 8 bits

Offset: 0x300a

Bits	Name	Attr	Description
7:0	pllcfgfreq	R/W	PLL Configuration Frequency (pllcfgfreq[7:0]). Value After Reset: 0x20

PHY_PLLCFGFREQ1

Description: PHY PLL Test Interface Register 1

Size: 8 bits

Offset: 0x300b

Bits	Name	Attr	Description
7:0	pllcfgfreq	R/W	PLL Configuration Frequency (pllcfgfreq[15:8]). Value After Reset: 0x27

PHY_PLLCFGFREQ2

Description: PHY PLL Test Interface Register 2

Size: 8 bits

Offset: 0x300c

Bits	Name	Attr	Description
7:0	pllcfgfreq	R/W	PLL Configuration Frequency (pllcfgfreq[23:16]). Value After Reset: 0x0

phy_i2cm_slave

Description: PHY I2C Slave Address Configuration Register

Size: 8 bits

Offset: 0x3020

Bits	Name	Attr	Description
7			Reserved for future use.
6:0	slaveaddr	R/W	Slave address to be sent during read and write operations. PHY Gen2 slave address: 7'h69 HEAC PHY slave address: 7'h49 Value After Reset: 0x0

phy_i2cm_address

Description: PHY I2C Address Configuration Register

This register writes the address for read and write operations.

Size: 8 bits

Offset: 0x3021

Bits	Name	Attr	Description
7:0	address	R/W	Register address for read and write operations Value After Reset: 0x0

phy_i2cm_datao_1

Description: PHY I2C Data Write Register 1

Size: 8 bits

Offset: 0x3022

Bits	Name	Attr	Description
7:0	datao	R/W	Data MSB (datao[15:8]) to be written on register pointed by phy_i2cm_address [7:0]. Value After Reset: 0x0

phy_i2cm_datao_0

Description: PHY I2C Data Write Register 0

Size: 8 bits

Offset: 0x3023

Bits	Name	Attr	Description
7:0	datao	R/W	Data LSB (datao[7:0]) to be written on register pointed by phy_i2cm_address [7:0]. Value After Reset: 0x0

phy_i2cm_datai_1

Description: PHY I2C Data Read Register 1

Size: 8 bits

Offset: 0x3024

Bits	Name	Attr	Description
7:0	datai	R	Data MSB (datai[15:8]) read from register pointed by phy_i2cm_address[7:0]. Value After Reset: 0x0

phy_i2cm_datai_0

Description: PHY I2C Data Read Register 0

Size: 8 bits

Offset: 0x3025

Bits	Name	Attr	Description
7:0	datai	R	Data LSB (datai[7:0]) read from register pointed by phy_i2cm_address[7:0]. Value After Reset: 0x0

phy_i2cm_operation

Description: PHY I2C RD/RD_EXT/WR Operation Register

This register requests read and write operations from the I2C Master PHY. This register can only be written; reading this register always results in 00h. Writing 1'b1 simultaneously to read and write requests is considered a read request.

Size: 8 bits

Offset: 0x3026

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	wr	W	Write operation request Value After Reset: 0x0
3:1			Reserved for future use.
0	rd	W	Read operation request Value After Reset: 0x0

phy_i2cm_int

Description: PHY I2C Done Interrupt Register

This register contains and configures I2C master PHY done interrupt.

Size: 8 bits

Offset: 0x3027

Bits	Name	Attr	Description
7:4			Reserved for future use.
3	done_pol	R/W	Done interrupt polarity configuration Value After Reset: 0x1
2	done_mask	R/W	Done interrupt mask signal Value After Reset: 0x0
1	done_interrupt	R	Operation done interrupt bit. Only lasts for 1 SFR clock cycle and is auto cleared after it. {done_interrupt = (done_mask==0b) && (done_status==done_pol)} Value After Reset: 0x0
0	done_status	R	Operation done status bit. Marks the end of a read or write operation. Value After Reset: 0x0

phy_i2cm_ctlint

Description: PHY I2C error Interrupt Register

This register contains and configures the I2C master PHY error interrupts.

Size: 8 bits

Offset: 0x3028

Bits	Name	Attr	Description
7	nack_pol	R/W	Not acknowledge error interrupt polarity configuration Value After Reset: 0x1
6	nack_mask	R/W	Not acknowledge error interrupt mask signal Value After Reset: 0x0
5	nack_interrupt	R	Not acknowledge error interrupt bit. Only lasts for one SFR clock cycle and is auto cleared after it. {nack_interrupt = (nack_mask==0b) && (nack_status==nack_pol)}. Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register. Value After Reset: 0x0
4	nack_status	R	Not acknowledge error status bit. Error on I2C not acknowledge. Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register. Value After Reset: 0x0
3	arbitration_pol	R/W	Arbitration error interrupt polarity configuration. Value After Reset: 0x1
2	arbitration_mask	R/W	Arbitration error interrupt mask signal. Value After Reset: 0x0
1	arbitration_interrupt	R	Arbitration error interrupt bit

Bits	Name	Attr	Description
			{arbitration_interrupt = (arbitration_mask==0b) && (arbitration_status==arbitration_pol)} Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register. Value After Reset: 0x0

Fields for Register: phy_i2cm_ctlint (Continued)

Bits	Name	Attr	Description
0	arbitration_status	R	Arbitration error status bit. Error on master I2C protocol arbitration. Only lasts for one SFR clock cycle and is auto cleared after it. Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register. Value After Reset: 0x0

phy_i2cm_div

Description: PHY I2C Speed control Register

This register wets the I2C Master PHY to work in either Fast or Standard mode.

Size: 8 bits

Offset: 0x3029

Bits	Name	Attr	Description
7:4			Reserved for future use.
3	fast_std_mode	R/W	Sets the I2C Master to work in Fast Mode or Standard Mode: 1: Fast Mode 0: Standard Mode Value After Reset: 0x1
2:0	spare	R/W	Reserved as "spare" register with no associated functionality. Value After Reset: 0x3

phy_i2cm_softrstz

Description: PHY I2C SW reset control register

This register sets the I2C Master PHY software reset.

Size: 8 bits

Offset: 0x302a

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	i2c_softrstz	R/W	I2C Master Software Reset. Active by writing a zero and auto cleared to one in the following cycle. Value After Reset: 0x1

phy_i2cm_ss_scl_hcnt_1_addr

Description: PHY I2C Slow Speed SCL High Level Control Register 1

Size: 8 bits

Offset: 0x302b

Bits	Name	Attr	Description

7:0	i2cmp_ss_scl_hcnt1	R/W	PHY I2C Slow Speed SCL High Level Control Register 1 Value After Reset: 0x0

phy_i2cm_ss_scl_hcnt_0_addr

Description: PHY I2C Slow Speed SCL High Level Control Register 0

Size: 8 bits

Offset: 0x302c

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_hcnt0	R/W	PHY I2C Slow Speed SCL High Level Control Register 0 Value After Reset: 0x6c

phy_i2cm_ss_scl_lcnt_1_addr

Description: PHY I2C Slow Speed SCL Low Level Control Register 1

Size: 8 bits

Offset: 0x302d

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_lcnt1	R/W	PHY I2C Slow Speed SCL Low Level Control Register 1 Value After Reset: 0x0

phy_i2cm_ss_scl_lcnt_0_addr

Description: PHY I2C Slow Speed SCL Low Level Control Register 0

Size: 8 bits

Offset: 0x302e

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_lcnt0	R/W	PHY I2C Slow Speed SCL Low Level Control Register 0 Value After Reset: 0x7f

phy_i2cm_fs_scl_hcnt_1_addr

Description: PHY I2C Fast Speed SCL High Level Control Register 1

Size: 8 bits

Offset: 0x302f

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_hcnt1	R/W	PHY I2C Fast Speed SCL High Level Control Register 1 Value After Reset: 0x0

phy_i2cm_fs_scl_hcnt_0_addr

Description: PHY I2C Fast Speed SCL High Level Control Register 0

Size: 8 bits

Offset: 0x3030

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_hcnt0	R/W	PHY I2C Fast Speed SCL High Level Control Register

Bits	Name	Attr	Description
			0 Value After Reset: 0x11

phy_i2cm_fs_scl_lcnt_1_addr

Description: PHY I2C Fast Speed SCL Low Level Control Register 1

Size: 8 bits

Offset: 0x3031

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_lcnt1	R/W	PHY I2C Fast Speed SCL Low Level Control Register 1 Value After Reset: 0x0

phy_i2cm_fs_scl_lcnt_0_addr

Description: PHY I2C Fast Speed SCL Low Level Control Register 0

Size: 8 bits

Offset: 0x3032

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_lcnt0	R/W	PHY I2C Fast Speed SCL Low Level Control Register 0 Value After Reset: 0x24

phy_i2cm_sda_hold

Description: PHY I2C SDA HOLD Control Register

Size: 8 bits

Offset: 0x3033

Bits	Name	Attr	Description
7:0	osda_hold	R/W	Defines the number of SFR clock cycles to meet tHD:DAT (300 ns) osda_hold = round_to_high_integer (300 ns / (1/ isfrclk_frequency)) Value After Reset: 0x9

jtag_phy_config

Description: PHY I2C/JTAG I/O Configuration Control Register

Size: 8 bits

Offset: 0x3034

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	i2c_jtagz	R/W	Configures the JTAG PHY interface output pin I2C_JTAGZ to select the PHY configuration interface when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_i2c_jtagz when PHY_EXTERNAL=1. 1'b0: JTAG configuration Interface 1'b1: I2C configuration Interface Value After Reset: 0x1
3:1			Reserved for future use.
0	jtag_trst_n	R/W	Configures the JTAG PHY interface output pin

Bits	Name	Attr	Description
			JTAG_TRST_N when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_trst_n when PHY_EXTERNAL=1. Value After Reset: 0x1

jtag_phy_tap_tck

Description: PHY JTAG Clock Control Register

Size: 8 bits

Offset: 0x3035

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	jtag_tck	R/W	Configures the JTAG PHY interface pin JTAG_TCK when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_tck when PHY_EXTERNAL=1. Value After Reset: 0x0

jtag_phy_tap_in

Description: PHY JTAG TAP In Control Register

Size: 8 bits

Offset: 0x3036

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	jtag_tms	R/W	Configures the JTAG PHY interface pin JTAG_TMS when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_tms when PHY_EXTERNAL=1. Value After Reset: 0x1
3:1			Reserved for future use.
0	jtag_tdi	R/W	Configures the JTAG PHY interface pin JTAG_TDI when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_tdi when PHY_EXTERNAL=1. Value After Reset: 0x0

jtag_phy_tap_out

Description: PHY JTAG TAP Out Control Register

Size: 8 bits

Offset: 0x3037

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	jtag_tdo_en	R	Read JTAG PHY interface input pin JTAG_TDO_EN when in internal control mode (iphy_ext_ctrl=1'b0) or iphyext_jtag_tdo_en when PHY_EXTERNAL=1 Value After Reset: 0x0
3:1			Reserved for future use.
0	jtag_tdo	R	Read JTAG PHY interface input pin JTAG_TDO when in internal control mode (iphy_ext_ctrl=1'b0) or

Bits	Name	Attr	Description
			iphyext_jtag_tdo when PHY_EXTERNAL=1 Value After Reset: 0x0

jtag_phy_addr

Description: PHY JTAG Address Control Register

Size: 8 bits

Offset: 0x3038

Bits	Name	Attr	Description
7:0	jtag_addr	R/W	Configures the JTAG PHY interface pin JTAG_ADDR[7:0] when in internal control mode (iphy_ext_ctrl=1'b0) or iphyext_jtag_addr[7:0] when PHY_EXTERNAL=1 Value After Reset: 0x0

AudioSample Registers

Audio Sample Registers. Follow the link for the register to see a detailed description of the register.

Registers for Address Block: AudioSample

Register	Offset	Description
aud_conf0	0x3100	Audio I2S Software FIFO Reset, Select, and Enable Control Register 0 This register configures the...
aud_conf1	0x3101	Audio I2S Width Configuration Register 1 This register configures the data width of the input...
aud_int	0x3102	I2S FIFO status and interrupts. This register configures the I2S FIFO status and interrupts.
aud_conf2	0x3103	Audio I2S PCUV, NLPCM and HBR configuration Register 2 This register configures the I2S Audio
aud_int1	0x3104	I2S Mask Interrupt Register This register masks the interrupts present in the I2S module.

aud_conf0

Description: Audio I2S Software FIFO Reset, Select, and Enable Control Register 0

This register configures the I2S input enable that indicates which input I2S channels have valid data. It also allows the system processor to reset audio FIFOs upon underflow/overflow error detection.

Size: 8 bits

Offset: 0x3100

Bits	Name	Attr	Description
7	sw_audio_fifo_rst	R/W	Audio FIFOs software reset Writing 0b: no action taken Writing 1b: Resets all audio FIFOs Reading from this register always returns 0b. Note: If a FIFO reset request (via SFR command) lands in the middle of an I2S transaction, the samples become

Bits	Name	Attr	Description
			misaligned (left-right sequence lost). As a solution, for each FIFO reset, an associated I2S reset must be issued (writing 8'hF7 to MC_SWRSTZ register). Value After Reset: 0x0
6	spare_2	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
5	i2s_select	R/W	1b: Selects I2S Audio Interface 0b: Selects the second (SPDIF/GPA) interface, in configurations with more that one audio interface (DOUBLE/GDOUBLE) Value After Reset: 0x1
4	spare_1	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
3:0	I2S_in_en	R/W	Action I2S_in_en[0] - I2Sdata[0] enable I2S_in_en[1] - I2Sdata[1] enable I2S_in_en[2] - I2Sdata[2] enable I2S_in_en[3] - I2Sdata[3] enable Value After Reset: 0xf

aud_conf1

Description: Audio I2S Width Configuration Register 1 This register configures the data width of the input data.

Size: 8 bits

Offset: 0x3101

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:0	I2S_width	R/W	I2S input data width I2S_width[4:0] | Action 00000b-01111b | Not used 10000b | 16 bit data samples at input 10001b | 17 bit data samples at input 10010b | 18 bit data samples at input 10011b | 19 bit data samples at input 10100b | 20 bit data samples at input 10101b | 21 bit data samples at input 10110b | 22 bit data samples at input 10111b | 23 bit data samples at input 11000b | 24 bit data samples at input 11001b-11111b | Not Used Value After Reset: 0x18

aud_int

Description: I2S FIFO status and interrupts.

This register configures the I2S FIFO status and interrupts.

Size: 8 bits

Offset: 0x3102

Bits	Name	Attr	Description
7:4			Reserved for future use.
3	fifo_empty_mask	R/W	FIFO empty mask. Value After Reset: 0x0
2	fifo_full_mask	R/W	FIFO full mask. Value After Reset: 0x0
1:0			Reserved for future use.

aud_conf2

Description: Audio I2S PCUV, NLPCM and HBR configuration Register 2

This register configures the I2S Audio Data mapping. By default, audio data mapping is the standard I2S Linear PCM (L-PCM) mapping. You can choose to use the I2S interface to transport HBR or Non- Linear PCM (NL-PCM) audio, by setting the relevant bit in this register.

Size: 8 bits

Offset: 0x3103

Bits	Name	Attr	Description
7:3			Reserved for future use.
2	insert_pcuv	R/W	When set (1'b1), this bit enables the insertion of the PCUV (Parity, Channel Status, User bit and Validity) bits on the incoming audio stream (support limited to Linear PCM audio). If disabled, the incoming audio stream must contain the PCUV bits, mapped according to Databook. Value After Reset: 0x1
1	NLPCM	R/W	I2S NLPCM Mode Enable. When enabled, this bit assumes that PCUV data is included on the I2S audio stream according to the description located in the "I2S Interface" section of Chapter 2, "Functional Description." Value After Reset: 0x0
0	HBR	R/W	I2S HBR Mode Enable. When enabled, the I2S audio stream is transmitted using HBR packets. Value After Reset: 0x0

aud_int1

Description: I2S Mask Interrupt Register

This register masks the interrupts present in the I2S module.

Size: 8 bits

Offset: 0x3104

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	fifo_overrun_mask	R/W	FIFO overrun mask Value After Reset: 0x1
3:0			Reserved for future use.

AudioPacketizer Registers

Audio Packetizer Registers. Follow the link for the register to see a detailed description of

the register.
Registers for Address Block: AudioPacketizer

Register	Offset	Description
aud_n1	0x3200	Audio Clock Regenerator N Value Register 1 For N expected values, refer to the HDMI 1.4b...
aud_n2	0x3201	Audio Clock Regenerator N Value Register 2 For N expected values, refer to the HDMI 1.4b...
aud_n3	0x3202	Audio Clock Regenerator N Value Register 3 For N expected values, refer to the HDMI 1.4b...
aud_cts1	0x3203	Audio Clock Regenerator CTS Value Register 1 For CTS expected values, refer to the HDMI 1.4b...
aud_cts2	0x3204	Audio Clock Regenerator CTS Register 2 For CTS expected values, refer to the HDMI 1.4b...
aud_cts3	0x3205	Audio Clock Regenerator CTS value Register 3. For CTS expected values, refer to the HDMI 1.4b...
aud_inputclkfs	0x3206	Audio Input Clock FS Factor Register
aud_cts_dither	0x3207	Audio CTS Dither Register

aud_n1
Description: Audio Clock Regenerator N Value Register 1 For N expected values, refer to the HDMI 1.4b specification.
Size: 8 bits
Offset: 0x3200

Bits	Name	Attr	Description
7:0	AudN	R/W	HDMI Audio Clock Regenerator N value Value After Reset: 0x0

aud_n2
Description: Audio Clock Regenerator N Value Register 2 For N expected values, refer to the HDMI 1.4b specification.
Size: 8 bits
Offset: 0x3201

Bits	Name	Attr	Description
7:0	AudN	R/W	HDMI Audio Clock Regenerator N value Value After Reset: 0x0

aud_n3
Description: Audio Clock Regenerator N Value Register 3 For N expected values, refer to the HDMI 1.4b specification.
Size: 8 bits
Offset: 0x3202

Bits	Name	Attr	Description
7	ncts_atomic_write	R/W	When set, the new N and CTS values are only used when aud_n1 register is written. If clear, N and CTS data is updated each time a new N or CTS byte is

Bits	Name	Attr	Description
			written. The following write sequence is recommended: aud_n3 (set bit ncts_atomic_write if desired) aud_cts3 (set CTS_manual and CTS value if desired/enabled) aud_cts2 (required in CTS_manual) aud_cts1 (required in CTS_manual) aud_n3 (bit ncts_atomic_write with same value as in step 1.) aud_n2 aud_n1 For dynamic N/CTS changes, perform only steps from 2-7 or 5-7 depending on the state of CTS_manual. Value After Reset: 0x0
6:4			Reserved for future use.
3:0	AudN	R/W	HDMI Audio Clock Regenerator N value Value After Reset: 0x0

aud_cts1

Description: Audio Clock Regenerator CTS Value Register 1 For CTS expected values, refer to the HDMI 1.4b specification.

Size: 8 bits

Offset: 0x3203

Bits	Name	Attr	Description
7:0	AudCTS	R/W	HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism. Value After Reset: 0x0

aud_cts2

Description: Audio Clock Regenerator CTS Register 2

For CTS expected values, refer to the HDMI 1.4b specification.

Size: 8 bits

Offset: 0x3204

Bits	Name	Attr	Description
7:0	AudCTS	R/W	HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism. Value After Reset: 0x0

aud_cts3

Description: Audio Clock Regenerator CTS value Register 3. For CTS expected values, refer to the HDMI 1.4b specification.

Size: 8 bits

Offset: 0x3205

Bits	Name	Attr	Description
7:5	Spare_bits	R/W	Reserved as "spare" bits with no associated functionality. Value After Reset: 0x0
4	CTS_manual	R/W	If the CTS_manual bit equals 0b, this registers contains audCTS[19:0] generated by the Cycle time counter according to the specified timing. If the CTS_manual bit equals 1b, this register is configured with the audCTS[7:0] value that is output by the Audio Packetizer. Note: When the General Parallel Audio Interface (GPAUD) is enabled (AUDIO_IF = 6) or the AHB DMA Audio Interface is enabled (AUDIO_IF = 8), writing to these bits has no effect; reading these bits always return 0. Value After Reset: 0x0
3:0	AudCTS	R/W	HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism. Value After Reset: 0x0

aud_inputclkfs

Description: Audio Input Clock FS Factor Register

Size: 8 bits

Offset: 0x3206

Bits	Name	Attr	Description
7:3			Reserved for future use.
2:0	ifsfactor	R/W	Fs factor configuration: ifsfactor[2:0] | Audio Clock | Action 0 | 128xFs | If you select the Bypass SPDIF DRU unit in coreConsultant, the input audio clock (either I2S or SPDIF according to configuration) is used at the audio packetizer to calculate the CTS value and ACR packet insertion rate. | 256xFs | The input audio clock (I2S only) is divided by 2 and then used at audio packetizer to calculate the CTS value and ACR packet insertion rate. | 512xFs | The input audio clock (either I2S or SPDIF according to configuration) used divided by 4 and then used at the audio packetizer to calculate the CTS value and ACR packet insertion rate. Note: When the SPDIF interface is receiving an HBR audio stream ("Support for HBR over SDPIF" parameter must be enabled), it is required that the selected IFSFACTOR to be set at 512xFs in order to comply with the HDMI ACR requirements for HBR

Bits	Name	Attr	Description
			audio streams. | Reserved | 64xFs | The input audio clock (I2S only) is multiplied by 2 and then used at the audio packetizer to calculate the CTS value and ACR packet insertion rate. others | 128xFs | If you select the Bypass SPDIF DRU unit in coreConsultant, the input audio clock (either I2S or SPDIF according to configuration) is used at the audio packetizer to calculate the CTS value and ACR packet insertion rate. The SPDIF interface, for non HBR audio, requires that the configured oversampling value to be 128xFs when HTX_SPDIFBYPDRU is enabled and 512xFs if not. When the SPDIF interface is receiving HBR audio (HBR_ON_SPDIF must be enabled), in order to comply with the HDMI ACR requirements for HBR audio streams. Value After Reset: 0x0

aud_cts_dither

Description: Audio CTS Dither Register

Size: 8 bits

Offset: 0x3207

Bits	Name	Attr	Description
7:4	divisor	R/W	Dither divisor (4'd1 if no CTS Dither). This field should be configured with the value of divisor from the HDMI specification. Value After Reset: 0x1
3:0	dividend	R/W	Dither dividend (4'd1 if no CTS Dither). This field should be configured with the value of dividend from the HDMI specification. Value After Reset: 0x1

AudioSampleSPDIF Registers

Audio Sample SPDIF Registers. Follow the link for the register to see a detailed description of the register.

Registers for Address Block: AudioSampleSPDIF

Register	Offset	Description
aud_spdif0	0x3300	Audio SPDIF Software FIFO Reset Control Register 0 This register allows the system processor to...
aud_spdif1	0x3301	Audio SPDIF NLPCM and Width Configuration Register 1 This register configures the SPDIF data...
aud_spdifint	0x3302	Audio SPDIF FIFO Empty/Full Mask Register
aud_spdifint1	0x3303	Audio SPDIF Mask Interrupt Register 1 This register

Register	Offset	Description
		masks interrupts present in the SPDIF...
aud_spdif2	0x3304	Audio SPDIF Enable Confiiguration Register 2 This register configures the SPDIF input enable that...

aud_spdif0

Description: Audio SPDIF Software FIFO Reset Control Register 0
This register allows the system processor to reset audio FIFOs upon underflow/overflow error detection.
Size: 8 bits
Offset: 0x3300

Bits	Name	Attr	Description
7	sw_audio_fifo_rst	R/W	Audio FIFOs software reset Writing 0b: no action taken Writing 1b: Resets all audio FIFOs Reading from this register always returns 0b. Note: If a FIFO reset request (via register write command) lands in the middle of an SPDIF audio transaction, the samples become misaligned (left-right sequence lost). As a solution, for each FIFO reset, an associated SPDIF reset must be issued (writing 8'hEF to MC_SWRSTZ register). Value After Reset: 0x0
6:0	spare	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0xf

aud_spdif1

Description: Audio SPDIF NLPCM and Width Configuration Register 1 This register configures the SPDIF data width.
Size: 8 bits
Offset: 0x3301

Bits	Name	Attr	Description
7	setnlpcm	R/W	Select Non-Linear (1b) / Linear (0b) PCM mode Value After Reset: 0x0
6	spdif_hbr_mode	R/W	When set to 1'b1, this bit field indicates that the input stream has a High Bit Rate (HBR) to be transmitted in HDMI HBR packets. When clear (1b'0), the audio is transmitted in HDMI AUDS packets.Note:< Otherwise, this field is a "spare" bit with no associated functionality.

aud_spdifint

Description: Audio SPDIF FIFO Empty/Full Mask Register
Size: 8 bits
Offset: 0x3302

Bits	Name	Attr	Description
7:4			Reserved for future use.
3	spdif_fifo_empty_mask	R/W	Description: SPDIF FIFO empty mask Value After Reset: 0x0
2	spdif_fifo_full_mask	R/W	Description: SPDIF FIFO full mask Value After Reset: 0x0
1:0			Reserved for future use.

aud_spdifint1

Description: Audio SPDIF Mask Interrupt Register 1

This register masks interrupts present in the SPDIF module.

Size: 8 bits

Offset: 0x3303

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	fifo_overrun_mask	R/W	FIFO overrun mask Value After Reset: 0x1
3:0			Reserved for future use.

aud_spdif2

Description: Audio SPDIF Enable Confiiguration Register 2

This register configures the SPDIF input enable that indicates which input SPDIF channels have valid data.

Size: 8 bits

Offset: 0x3304

Bits	Name	Attr	Description
7:4			Reserved for future use.
3:0	SPDIF_in_en	R/W	Action SPDIF_in_en[0] - ispdifdata[0] enable SPDIF_in_en[1] - ispdifdata[1] enable SPDIF_in_en[2] - ispdifdata[2] enable SPDIF_in_en[3] - ispdifdata[3] enable Value After Reset: 0x1

AudioSampleGP Registers

Audio Sample GP Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
gp_conf0	0x3500	Audio GPA Software FIFO Reset Control Register 0
gp_conf1	0x3501	Audio GPA Channel Enable Configuration Register 1
gp_conf2	0x3502	Audio GPA HBR Enable Register 2
gp_mask	0x3506	Audio GPA FIFO Full and Empty Mask Interrupt Register

gp_conf0

Description: Audio GPA Software FIFO Reset Control Register 0

Size: 8 bits

Offset: 0x3500

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	sw_audio_fifo_rst	R/W	Audio FIFOs software reset Writing 0b: no action taken Writing 1b: Resets all audio FIFOs Reading from this register always returns 0b. Note: If a FIFO reset request (via register write command) lands in the middle of an GPAUD audio transaction, the samples become misaligned (left-right sequence lost). As a solution, for each FIFO reset, an associated SPDIF reset must be issued (writing 8'h7F to MC_SWRSTZ register). Value After Reset: 0x0

gp_conf1

Description: Audio GPA Channel Enable Configuration Register 1

Size: 8 bits

Offset: 0x3501

Bits	Name	Attr	Description
7:0	ch_in_en	R/W	Each bit controls the enabling of the respective audio channel. For instance, bit 1, when set (1'b1), the audio Channel 1 is enabled. When cleared, the referred channel is disabled. Value After Reset: 0x0

gp_conf2

Description: Audio GPA HBR Enable Register 2

Size: 8 bits

Offset: 0x3502

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	insert_pcuv	R/W	When set (1'b1), this bit enables the insertion of the PCUV (Parity, Channel Status, User bit and Validity) bits on the incoming audio stream (support limited to Linear PCM audio). If disabled, the incoming audio stream must contain the PCUV bits, mapped according to 2.6.4.2 Data Mapping Examples. Value After Reset: 0x0
0	HBR	R/W	HBR packets enable. The Hdmi_tx sends the HBR packets. This bit is enabled when the audio frequency is higher than 192 kHz. If this bit is enabled, the number of channels configured in GP_CONF1 must be set to 8'hFF. Value After Reset: 0x0

gp_mask

Description: Audio GPA FIFO Full and Empty Mask Interrupt Register
Size: 8 bits
Offset: 0x3506

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	fifo_overrun_mask	R/W	FIFO overrun mask Value After Reset: 0x1
3:2			Reserved for future use.
1	fifo_empty_mask	R/W	FIFO empty flag mask Value After Reset: 0x0
0	fifo_full_mask	R/W	FIFO full flag mask Value After Reset: 0x0

AudioDMA Registers

Audio DMA Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
ahb_dma_conf0	0x3600	Audio DMA SW FIFO reset and DMA Configuration Register 0 This register contains the software reset...
ahb_dma_start	0x3601	Audio DMA Start Register The start_dma_transaction bit field signals the AHB audio DMA to start...
ahb_dma_stop	0x3602	Audio DMA Stop Register The stop_dma_transaction bit field signals the AHB audio DMA to stop current...
ahb_dma_thrsld	0x3603	Audio DMA FIFO Threshold Register This register defines the FIFO medium threshold occupation value....
ahb_dma_straddr_set0[0:3]	0x3604 + (i * 0x1)	Audio DMA Start Address Set0 Register Array Address offset: i = 0 to 3 These registers define...
ahb_dma_stpaddr_set0[0:3]	0x3608 + (i * 0x1)	Audio DMA Stop Address Set0 Register Array Address offset: i = 0 to 3 This registers define the...
ahb_dma_bstraddr[0:3]	0x360c + (i * 0x1)	Audio DMA Burst Start Address Register Array Address offset: i = 0 to 3 These read-only registers...
ahb_dma_mblength0	0x3610	Audio DMA Burst Length Register 0 This registers holds the length of the current burst operation....
ahb_dma_mblength	0x3611	Audio DMA Burst Length Register 1 This registers holds the length of the current burst operation....
ahb_dma_mask	0x3614	Audio DMA Mask Interrupt Register This register masks each of the interrupts present in the AHB...
ahb_dma_conf	0x3616	Audio DMA Channel Enable Configuration Register 1 In AUDS packet configuration with layout 0 selected,...
ahb_dma_buffmask	0x3619	Audio DMA Buffer Mask Interrupt Register
ahb_dma_mask1	0x361b	Audio DMA Mask Interrupt Register 1 This register masks interrupts present in the AHB audio DMA...

Register	Offset	Description
ahb_dma_status	0x361c	Audio DMA Status
ahb_dma_conf2	0x361d	Audio DMA Configuration Register 2
ahb_dma_straddr_set1[0:3]	0x3620 + (i * 0x1)	Audio DMA Start Address Set 1 Register Array Address offset: i = 0 to 3 These registers define...
ahb_dma_stpaddr_set1[0:3]	0x3624 + (i * 0x1)	Audio DMA Stop Address Set 1 Register Array Address offset: i = 0 to 3 These registers define...

ahb_dma_conf0

Description: Audio DMA SW FIFO reset and DMA Configuration Register 0
This register contains the software reset bit for the audio FIFOs. It also configures operating modes of the AHB master.
Size: 8 bits
Offset: 0x3600

Bits	Name	Attr	Description
7	sw_fifo_rst	R/W	This is the software reset bit for the audio and FIFO clear. Writing 0'b does not result in any action. Writing 1'b to this register resets all audio FIFOs. Reading from this register always returns 0'b. Value After Reset: 0x0
6	insert_pcuv	R/W	Enables the insertion of PCUV data Value After Reset: 0x0
5			Reserved for future use.
4	hbr	R/W	HBR packet enable The Hdmi_tx sends the HBR packets. This bit must be enabled when transmitting non-linear audio of frequency higher than 192 kHz. If this bit is enabled, the number of channels configured in AHB_DMA_CONF1 is always 8. Value After Reset: 0x0
3	enable_hlock	R/W	Enable request of locked burst AHB mechanism. 1'b: Enables the usage of hlock for master request to arbiter of a locked complete burst. 0'b: Disables request of locked burst AHB mechanism Value After Reset: 0x0
2:1	incr_type	R/W	Selects the preferred burst length size 00'b: Corresponds to INCR4 fixed four beat, incremental AHB burst mode. Only valid when burst_mode is high. 01'b: Corresponds to INCR8 fixed eight beat incremental AHB burst mode. Only valid when burst_mode is high. 10'b: Corresponds to INCR16 fixed 16 beat incremental AHB burst mode. Only valid when burst_mode is high. 11'b: Corresponds to INCR16 fixed 16 beat incremental AHB burst mode. Only valid when burst_mode is high.

			Value After Reset: 0x0
0	burst_mode	R/W	1'b: Forces the burst mode to be fixed beat, incremental burst mode designated by the incr_type[1:0] signal. 0'b: Normal operation is unspecified length, incremental burst. It corresponds to INCR AHB burst mode. Value After Reset: 0x0

ahb_dma_start

Description: Audio DMA Start Register

The start_dma_transaction bit field signals the AHB audio DMA to start accessing system memory in order to fetch data samples to store in the FIFO. After the operation starts, a new request for a DMA start is ignored until the DMA is stopped or it reaches the end address. Only in one of these situations is a new start request acknowledged.

The first DMA burst request after start_dma_transaction configuration uses initial_addr[31:0] as ohaddr[31:0] value; mburstlength[8:0] is set to the maximum admissible value. This maximum value is constrained by the size of buffer provided, the instantiated FIFO depth, or/and the number of words up to the next 1 Kbyte boundary.

Size: 8 bits

Offset: 0x3601

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	start_dma_transaction	R/W	Start DMA transaction This register is auto-cleared when the transfer operation is completed (done). Value After Reset: 0x0

ahb_dma_stop

Description: Audio DMA Stop Register

The stop_dma_transaction bit field signals the AHB audio DMA to stop current Attr. After it stops, if a new start DMA operation is requested, the DMA engine restarts the Attr using the initial_addr[31:0], which is programmed at ahb_dma_straddr0 to ahb_dma_straddr3.

Size: 8 bits

Offset: 0x3602

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	stop_dma_transaction	R/W	Stop DMA transaction This register is auto-cleared when the transfer operation is stopped (done). Value After Reset: 0x0

ahb_dma_thrsld

Description: Audio DMA FIFO Threshold Register

This register defines the FIFO medium threshold occupation value.

After the AHB master completes a burst transaction successfully, the FIFO may remain full till the data fetch interface requests samples. Each data fetch operation reduces the number of samples stored in the FIFO by the number of channels enabled.

Therefore, the fifo_threshold[7:0] is the medium number of samples that should be

available in the audio FIFO across the DMA operation.
As soon as the number of samples in the FIFO drops lower than the fifo_threshold[7:0], the DMA engine requests a new burst of samples for the AHB master. The length is constrained by the size of buffer provided, the instantiated FIFO depth minus fifo_threshold[7:0], and/or the number of words up to the next 1 kbyte boundary.
Size: 8 bits
Offset: 0x3603

Bits	Name	Attr	Description
7:0	fifo_threshold	R/W	FIFO medium threshold occupation value Value After Reset: 0x0

ahb_dma_straddr_set0[0:3]
Description: Audio DMA Start Address Set0 Register Array Address offset: i = 0 to 3
These registers define the initial_addr[31:0] used to initiate the DMA burst read transactions upon start_dma_transaction configuration.
Size: 8 bits
Offset: 0x3604 + (i * 0x1)

Bits	Name	Attr	Description
7:0	initial_addr	R/W	Defines init_addr[7:0] to initiate DMA burst transactions Value After Reset: 0x0

ahb_dma_stpaddr_set0[0:3]
Description: Audio DMA Stop Address Set0 Register Array Address offset: i = 0 to 3
This registers define the final_addr[31:0] used as the final point to the DMA burst read transactions.
Upon start_dma_transaction configuration, the DMA engine starts requesting burst reads from the external system memory. Each burst read can have a maximum theoretical length of 256 words (due to the AMBA AHB specification 1 Kbyte boundary burst limitation).
The DMA engine is responsible for incrementing the burst starting address and defining its corresponding burst length to reach the final_addr[31:0] address. The last burst request issued by the DMA engine takes into account that it should only request data until the final_addr[31:0] address (included) and for that should calculate the correct burst length.
After reaching the final_addr[31:0] address, the done interrupt is active to signal completion of DMA operation.
Size: 8 bits
Offset: 0x3608 + (i * 0x1)

Bits	Name	Attr	Description
7:0	final_addr	R/W	Defines final_addr[7:0] to end DMA burst transactions Value After Reset: 0x0

ahb_dma_bstraddr[0:3]
Description: Audio DMA Burst Start Address Register Array Address offset: i = 0 to 3
These read-only registers compose the start address of the current burst operation.
burst_start_addr[31:0] = haddr[31:0] = initial_addr[31:0] + 16.
Size: 8 bits
Offset: 0x360c + (i * 0x1)

Bits	Name	Attr	Description

7:0	burst_addr	R	Start address for the current burst operation Value After Reset: 0x0

ahb_dma_mblength0

Description: Audio DMA Burst Length Register 0

This registers holds the length of the current burst operation. As an example, if the first burst transaction of the AHB audio DMA is a length of 8, then the second burst should start at address ohaddr[31:0] = initial_addr[31:0] + 32.

Size: 8 bits

Offset: 0x3610

Bits	Name	Attr	Description
7:0	mburstlength	R	Requested burst length (mburstlength[7:0]) Value After Reset: 0x0

ahb_dma_mblength1

Description: Audio DMA Burst Length Register 1

This registers holds the length of the current burst operation. As an example, if the first burst transaction of the AHB audio DMA is a length of 8, then the second burst should start at address ohaddr[31:0] = initial_addr[31:0] + 32.

Size: 8 bits

Offset: 0x3611

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	mburstlength	R	Requested burst length Value After Reset: 0x0

ahb_dma_mask

Description: Audio DMA Mask Interrupt Register

This register masks each of the interrupts present in the AHB audio DMA module.

Size: 8 bits

Offset: 0x3614

Bits	Name	Attr	Description
7	done_mask	R/W	DMA end of operation interrupt mask. Active when DMA engine reaches final_addr[15:0] or when stop DMA operation is activated. Value After Reset: 0x1
6	retrysplit_mask	R/W	Retry/split interrupt mask. Active when AHB master receives a RETRY or SPLIT response from slave. Value After Reset: 0x1
5	lostownership_mask	R/W	Master lost ownership interrupt mask when in burst transfer. Active when AHB master loses BUS ownership within the course of a burst transfer. Value After Reset: 0x1
4	error_mask	R/W	Error interrupt mask. Active when slave indicates error through the isresp[1:0]. Value After Reset: 0x1
3			Reserved for future use.
2	fifo_thrempty_mask	R/W	Audio FIFO empty interrupt mask when audio FIFO has

			less than the number of enabled audio channels. Value After Reset: 0x1
1	fifo_full_mask	R/W	Audio FIFO full interrupt mask. Value After Reset: 0x1
0	fifo_empty_mask	R/W	Audio FIFO empty interrupt mask. Value After Reset: 0x1

ahb_dma_conf1

Description: Audio DMA Channel Enable Configuration Register 1

In AUDS packet configuration with layout 0 selected, the maximum number of active channels is 2.

Size: 8 bits

Offset: 0x3616

Bits	Name	Attr	Description
7:0	ch_in_en	R/W	Each bit controls the enabling of the respective audio channel. For instance, when bit 1 is set (1'b1) the audio Channel 1 is enabled. When cleared, the referred channel is disabled. Value After Reset: 0x0

ahb_dma_buffmask

Description: Audio DMA Buffer Mask Interrupt Register

Size: 8 bits

Offset: 0x3619

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	mask_fifo_overrun	R/W	Buffer overrun flag mask Value After Reset: 0x1
3:2			Reserved for future use.
1	mask_buff_full	R/W	Buffer full flag mask Value After Reset: 0x1
0	mask_buff_empty	R/W	Buffer empty flag mask Value After Reset: 0x1

ahb_dma_mask1

Description: Audio DMA Mask Interrupt Register 1

This register masks interrupts present in the AHB audio DMA module.

Size: 8 bits

Offset: 0x361b

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	fifo_underrun_mask	R/W	AHB DMA FIFO underrun mask Value After Reset: 0x1
0	fifo_overrun_mask	R/W	AHB DMA FIFO overrun mask Value After Reset: 0x1

ahb_dma_status

Description: Audio DMA Status

Size: 8 bits

Offset: 0x361c

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	autostart_status	R	Indicates the set of start and stop addresses currently used by the AHB audio DMA. If cleared (1'b0), the start and stop addresses configured in the address range 0x3604 to 0x360B are being used. When set (1'b1), the configurations at address range 0x3620 to 0x3627 are being used. This bit is always at zero when autostart_enable is cleared (1'b0). Value After Reset: 0x0

ahb_dma_conf2

Description: Audio DMA Configuration Register 2

Size: 8 bits

Offset: 0x361d

Bits	Name	Attr	Description
7:2			Reserved for future use.
1	autostart_loop	R/W	Enables the AHB audio DMA auto-start loop mode Value After Reset: 0x1
0	autostart_enable	R/W	Enables the AHB audio DMA auto-start feature Value After Reset: 0x0

ahb_dma_straddr_set1[0:3]

Description: Audio DMA Start Address Set 1 Register Array Address offset: i = 0 to 3

These registers define the initial_addr_1[31:0] used to initiate the DMA burst read transactions upon start_dma_transaction configuration.

Size: 8 bits

Offset: 0x3620 + (i * 0x1)

Bits	Name	Attr	Description
7:0	initial_addr_1	R/W	Defines init_addr_1[7:0] to initiate DMA burst transactions Value After Reset: 0x0

ahb_dma_stpaddr_set1[0:3]

Description: Audio DMA Stop Address Set 1 Register Array Address offset: i = 0 to 3

These registers define the final_addr_1[31:0] used as the final point to the DMA burst read transactions. Upon start_dma_transaction configuration, the DMA engine starts requesting burst reads from the external system memory. Each burst read can have a maximum theoretical length of 256 words (due to the AMBA AHB specification 1 Kbyte boundary burst limitation).

The DMA engine is responsible for incrementing the burst starting address and defining its corresponding burst length to reach the final_addr[31:0] address. The last burst request issued by the DMA engine takes into account that it should only request data until the final_addr[31:0] address (included) and for that should calculate the correct burst length. After reaching the final_addr_1[31:0] address, the done interrupt is active to indicate completion of the DMA operation.

Size: 8 bits
Offset: 0x3624 + (i * 0x1)

Bits	Name	Attr	Description
7:0	final_addr_1	R/W	Defines final_addr_1[7:0] to end DMA burst transactions Value After Reset: 0x0

MainController Registers

Main Controller Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
mc_clkdis	0x4001	Main Controller Synchronous Clock Domain Disable Register
mc_swrstzreq	0x4002	Main Controller Software Reset Register Main controller software reset request per clock domain....
mc_opctrl	0x4003	Main Controller HDCP Bypass Control Register
mc_flowctrl	0x4004	Main Controller Feed Through Control Register
mc_phyrstz	0x4005	Main Controller PHY Reset Register
mc_lockonclock	0x4006	Main Controller Clock Present Register
mc_heacphy_rst	0x4007	Main Controller HEAC PHY Reset Register
mc_lockonclock_2	0x4009	Main Controller Clock Present Register 2
mc_swrstzreq_2	0x400a	Main Controller Software Reset Register 2 Main controller software reset request per clock domain....
mc_opsts	0x4010	Main Controller Status Register This register contains the information regarding the status of...
base_sfrdivlow	0x4018	SFR Clock Base Time Register Low
base_sfrdivhigh	0x4019	SFR Clock Base Time Register High

mc_clkdis

Description: Main Controller Synchronous Clock Domain Disable Register
Size: 8 bits
Offset: 0x4001

Bits	Name	Attr	Description
7	h22sclk_disable	R/W	HDCP22 clock synchronous disable signal. When active (1b), simultaneously bypasses HDCP22. Value After Reset: 0x0
6	hdcpclk_disable	R/W	HDCP clock synchronous disable signal. When active (1b), simultaneously bypasses HDCP. Value After Reset: 0x0
5	cecclk_disable	R/W	CEC Engine clock synchronous disable signal. Value After Reset: 0x0
4	cscclk_disable	R/W	Color Space Converter clock synchronous disable signal. Value After Reset: 0x0
3	audclk_disable	R/W	Audio Sampler clock synchronous disable signal. Value After Reset: 0x0

Bits	Name	Attr	Description
2	prepclk_disable	R/W	Pixel Repetition clock synchronous disable signal. Value After Reset: 0x0
1	tmdsclk_disable	R/W	TMDS clock synchronous disable signal. It is required to perform a write action on one of the following registers: fc_invidconf, fc_inhactiv0, fc_inhactiv1, fc_inhblank0, fc_inhblank1, fc_invactiv0 fc_invactiv1, fc_invblank, fc_hsyncindelay0, fc_hsyncindelay1, fc_hsyncinwidth0 fc_hsyncinwidth1, fc_vsyncindelay, fc_vsyncinwidth, fc_ctrldur, fc_exctrldur, fc_exctrlspac Value After Reset: 0x0
0	pixelclk_disable	R/W	Pixel clock synchronous disable signal. Value After Reset: 0x0

mc_swrstzreq

Description: Main Controller Software Reset Register

Main controller software reset request per clock domain. Writing zero to a bit of this register results in an NRZ signal toggle at sfrclk rate to an output signal that indicates a software reset request. This toggle must be used to generate a synchronized reset to de corresponding domain, with at least 1 clock cycle.

Size: 8 bits

Offset: 0x4002

Bits	Name	Attr	Description
7	igpaswrst_req	R/W	GPAUD interface soft reset request. This bit is enabled when the Generic Parallel Audio (GPAUD) interface is enabled (AUDIO_IF = 6). Otherwise, this bit returns zero. Value After Reset: 0x1
6	cecswrst_req	R/W	CEC software reset request. Defaults back to 1b after reset request. Note: After you configure cecswrst_req, set the value of the bit csc_clk_disable of the register mc_clkdis to 1, 0, and then 1 again. Value After Reset: 0x1
5			Reserved for future use.
4	ispdifswrst_req	R/W	SPDIF audio software reset request. Value After Reset: 0x1
3	ii2sswrst_req	R/W	I2S audio software reset request. Value After Reset: 0x1
2	prepswrst_req	R/W	Pixel Repetition software reset request. Value After Reset: 0x1
1	tmdsswrst_req	R/W	TMDS software reset request. It is required to perform a write action on one of the following registers:

			fc_invidconf, fc_inhactiv0, fc_inhactiv1, fc_inhblank0, fc_inhblank1, fc_invactiv0 fc_invactiv1, fc_invblank, fc_hsyncindelay0, fc_hsyncindelay1, fc_hsyncinwidth0 fc_hsyncinwidth1, fc_vsyncindelay, fc_vsyncinwidth, fc_ctrldur, fc_exctrldur, fc_exctrlspac Value After Reset: 0x1
0	pixelswrst_req	R/W	Pixel software reset request. Value After Reset: 0x1

mc_opctrl

Description: Main Controller HDCP Bypass Control Register

Size: 8 bits

Offset: 0x4003

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	h22s_ovr_val	R/W	HDCP SNPS 2.2 versus 1.4 switch override value 1'b0: The switch is routed to HDCP 1.4 signals when hdcp22snps_switch_lock is not set to 1'b1. 1'b1: The switch is routed to HDCP 2.2 SNPS signals when hdcp22snps_switch_lock is not set to 1'b1. Value After Reset: 0x1
4	h22s_switch_lck	R/W	HDCP 2.2 SNPS switch lock 1'b0: Enables you to change the direction of the HDCP 2.2 SNPS versus 1.4 switch by using the hdcp22snps_ovr_val. 1'b1: You can still write to hdcp22snps_ovr_val but has no effect over the HDCP 2.2 SNPS versus 1.4 switch, that keeps as it was configured by hdcp22snps_ovr_val at the time the 1'b1 was written to this bit field. Once you set the value to 1'b1, you can change the value back to 1'b0 only by issuing a master reset to the Hdmi_tx. Value After Reset: 0x0
3:1			Reserved for future use.
0	hdcp_block_byp	R/W	Block HDCP bypass mechanism 1'b0: This is the default value. You can write to the hdcp_clkdisable bit of the register mc_clkdis and bypass HDCP by acting on the register mc_clkdis bit 6 (hdcp_clkdisable) 1'b1: You can still write to the hdcp_clkdisable bit of the register mc_clkdis but this action disables the HDCP module and blocks the bypass mechanism. The output data is frozen and the HDMI Tx and RX fail authentication. Once you set the value to 1'b1, you can change the value back to 1'b0 only by issuing a master reset to the

Bits	Name	Attr	Description
			Hdmi_tx. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0

mc_flowctrl

Description: Main Controller Feed Through Control Register

Size: 8 bits

Offset: 0x4004

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	Feed_through_off	R/W	Video path Feed Through enable bit: 1b: Color Space Converter is in the video data path. 0b: Color Space Converter is bypassed (not in the video data path). Value After Reset: 0x0

mc_phyrstz

Description: Main Controller PHY Reset Register

Size: 8 bits

Offset: 0x4005

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	phyrstz	R/W	HDMI Source PHY active low reset control for PHY GEN1, active high reset control for PHY GEN2. Value After Reset: "(PHY_GEN2== 1) ? 1 : 0"

mc_lockonclock

Description: Main Controller Clock Present Register

Size: 8 bits

Offset: 0x4006

Bits	Name	Attr	Description
7	igpaclk	R/W1C	GPAUD interface clock status. This bit is enabled when the Generic Parallel Audio (GPAUD) interface is enabled (AUDIO_IF = 6). Otherwise, this bit returns zero. This bit indicates the clock is present in the system. It is cleared by writing 1 to this bit. Value After Reset: 0x0
6	pclk	R/W1C	Pixel clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position. Value After Reset: 0x0
5	tclk	R/W1C	TMDS clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position. Value After Reset: 0x0
4	prepclk	R/W1C	Pixel Repetition clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.

			Value After Reset: 0x0
3	i2sclk	R/W1C	I2S clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position. Value After Reset: 0x0
2	audiospdifclk	R/W1C	SPDIF clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position. Value After Reset: 0x0
1			Reserved for future use.
0	cecclk	R/W1C	CEC clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position. Value After Reset: 0x0

mc_heacphy_rst

Description: Main Controller HEAC PHY Reset Register

Size: 8 bits

Offset: 0x4007

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	heacphyrst	R/W	HEAC PHY reset (active high) Value After Reset: 0x1

mc_lockonclock_2

Description: Main Controller Clock Present Register 2

Size: 8 bits

Offset: 0x4009

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	ahbdmaclk	R/W1C	AHB audio DMA clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position. Value After Reset: 0x0

mc_swrstzreq_2

Description: Main Controller Software Reset Register 2

Main controller software reset request per clock domain. Writing zero to a bit of this register results in a signal toggle that indicates a software reset request. This toggle is used to generate a synchronized reset to the corresponding domain, with one or more clock cycles.

Size: 8 bits

Offset: 0x400a

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	ahbdmaswrst_req	R/W	AHB audio DMA software reset request. Writing 1'b1 does not result in any action. Writing 1'b0 to this register resets all AHB audio logic. Value After Reset: 0x0

mc_opsts

Description: Main Controller Status Register

This register contains the information regarding the status of the HDCP SNPS 2.2 versus 1.4 switch.
Size: 8 bits
Offset: 0x4010

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	h22s_switch_sts	R	HDCP SNPS 2.2 versus 1.4 switch value status. 1'b0: HDCP 1.4 selected 1'b1: HDCP 2.2 selected Value After Reset: 0x1

base_sfrdivlow
Description: SFR Clock Base Time Register Low
Size: 8 bits
Offset: 0x4018

Bits	Name	Attr	Description
7:0	base_sfrdiv_lo	R/W	SFR clock divider Low This register must be configured with the 8 least-significant bits of the value sfrclk frequency divided by 1000 (for example, for 27 MHz base_sfrdiv[14:0] = 27027). The configured data is used to generate a reference pulse of 1ms period that is needed by several timers within the controller. Value After Reset: 0x93

base_sfrdivhigh
Description: SFR Clock Base Time Register High
Size: 8 bits
Offset: 0x4019

Bits	Name	Attr	Description
7			Reserved for future use.
6:0	base_sfrdiv_hi	R/W	SFR clock divider High This register must be configured with the 7 most-significant bits of the value sfrclk frequency divided by 1000 (for example, for 27 MHz base_sfrdiv[14:0] = 27027). The configured data is used to generate a reference pulse of 1ms period that is needed by several timers within the controller. Value After Reset: 0x69

ColorSpaceConverter Registers
Color Space Converter Registers Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
csc_cfg	0x4100	Color Space Converter Interpolation and Decimation Configuration Register
csc_scale	0x4101	Color Space Converter Scale and Deep Color

Register	Offset	Description
		Configuration Register
csc_coef_a1_msb	0x4102	Color Space Converter Matrix A1 Coefficient Register MSB Notes: - The coefficients used in the...
csc_coef_a1_lsb	0x4103	Color Space Converter Matrix A1 Coefficient Register LSB Notes: - The coefficients used in the...
csc_coef_a2_msb	0x4104	Color Space Converter Matrix A2 Coefficient Register MSB Color Space Conversion A2...
csc_coef_a2_lsb	0x4105	Color Space Converter Matrix A2 Coefficient Register LSB Color Space Conversion A2...
csc_coef_a3_msb	0x4106	Color Space Converter Matrix A3 Coefficient Register MSB Color Space Conversion A3...
csc_coef_a3_lsb	0x4107	Color Space Converter Matrix A3 Coefficient Register LSB Color Space Conversion A3...
csc_coef_a4_msb	0x4108	Color Space Converter Matrix A4 Coefficient Register MSB Color Space Conversion A4...
csc_coef_a4_lsb	0x4109	Color Space Converter Matrix A4 Coefficient Register LSB Color Space Conversion A4...
csc_coef_b1_msb	0x410a	Color Space Converter Matrix B1 Coefficient Register MSB Color Space Conversion B1...
csc_coef_b1_lsb	0x410b	Color Space Converter Matrix B1 Coefficient Register LSB Color Space Conversion B1...
csc_coef_b2_msb	0x410c	Color Space Converter Matrix B2 Coefficient Register MSB Color Space Conversion B2...
csc_coef_b2_lsb	0x410d	Color Space Converter Matrix B2 Coefficient Register LSB Color Space Conversion B2...
csc_coef_b3_msb	0x410e	Color Space Converter Matrix B3 Coefficient Register MSB Color Space Conversion B3...
csc_coef_b3_lsb	0x410f	Color Space Converter Matrix B3 Coefficient Register LSB Color Space Conversion B3...
csc_coef_b4_msb	0x4110	Color Space Converter Matrix B4 Coefficient Register MSB Color Space Conversion B4...

Registers for Address Block: ColorSpaceConverter (Continued)

Register	Offset	Description
csc_coef_b4_lsb	0x4111	Color Space Converter Matrix B4 Coefficient Register LSB Color Space Conversion B4...
csc_coef_c1_msb	0x4112	Color Space Converter Matrix C1 Coefficient Register MSB Color Space Conversion C1...
csc_coef_c1_lsb	0x4113	Color Space Converter Matrix C1 Coefficient Register LSB Color Space Conversion C1...
csc_coef_c2_msb	0x4114	Color Space Converter Matrix C2 Coefficient Register MSB Color Space Conversion C2...
csc_coef_c2_lsb	0x4115	Color Space Converter Matrix C2 Coefficient Register LSB Color Space Conversion C2...
csc_coef_c3_msb	0x4116	Color Space Converter Matrix C3 Coefficient Register MSB Color Space Conversion C3...

csc_coef_c3_lsb	0x4117	Color Space Converter Matrix C3 Coefficient Register LSB Color Space Conversion C3...
csc_coef_c4_msb	0x4118	Color Space Converter Matrix C4 Coefficient Register MSB Color Space Conversion C4...
csc_coef_c4_lsb	0x4119	Color Space Converter Matrix C4 Coefficient Register LSB Color Space Conversion C4...
csc_limit_up_msb	0x411a	Color Space Converter Matrix Output Up Limit Register MSB For more details, refer to the HDMI 1.4...
csc_limit_up_lsb	0x411b	Color Space Converter Matrix output Up Limit Register LSB For more details, refer to the HDMI 1.4...
csc_limit_dn_msb	0x411c	Color Space Converter Matrix output Down Limit Register MSB For more details, refer to the HDMI...
csc_limit_dn_lsb	0x411d	Color Space Converter Matrix output Down Limit Register LSB For more details, refer to the HDMI...

csc_cfg

Description: Color Space Converter Interpolation and Decimation Configuration Register

Size: 8 bits

Offset: 0x4100

Bits	Name	Attr	Description
7	csc_limit	R/W	When set (1'b1), the range limitation values defined in registers csc_mat_uplim and csc_mat_dnlim are applied to the output of the Color Space Conversion matrix. This feature ensures that the video output range is always respected, independently of the matrix coefficient configuration or of the video input stream. Value After Reset: 0x0
6	spare_2	R/W	Reserved as "spare" register with no associated functionality. Value After Reset: 0x0
5:4	intmode	R/W	Chroma interpolation configuration: intmode[1:0] | Chroma Interpolation 00 | interpolation disabled 01 | Hu (z) =1 + z^(-1) 10 | Hu(z)=1/ 2 + z^(-11)+1/2 z^(-2) 11 | interpolation disabled Value After Reset: 0x0
3:2	spare_1	R/W	Reserved as "spare" register with no associated functionality. Value After Reset: 0x0
1:0	decmode	R/W	Chroma decimation configuration: decmode[1:0] | Chroma Decimation 00 | decimation disabled 01 | Hd (z) =1 10 | Hd(z)=1/ 4 + 1/2z^(-1)+1/4 z^(-2) 11 | Hd(z)x2^(11)= -5+12z^(-2) - 22z^(-

			4)+39z^(-8) +109z^(-10) -204z^(-12)+648z^(-14) + 1024z^(-15) +648z^(-16) - 204z^(-18) +109z^(-20)- 65z^(-22) +39z^(-24) - 22z^(-26) +12z^(- 28)-5z^(-30) Value After Reset: 0x0

csc_scale

Description: Color Space Converter Scale and Deep Color Configuration Register
Size: 8 bits
Offset: 0x4101

Bits	Name	Attr	Description
7:4	csc_color_depth	R/W	Color space converter color depth configuration: csc_colordepth[3:0] | Action 0000 | 24 bit per pixel video (8 bit per component). 0001-0011 | Reserved. Not used. 0100 | 24 bit per pixel video (8 bit per component). 0101 | 30 bit per pixel video (10 bit per component). 0110 | 36 bit per pixel video (12 bit per component). 0111 | 48 bit per pixel video (16 bit per component). other | Reserved. Not used. Value After Reset: 0x0
3:2	spare	R/W	The is a Reserved as "spare" register with no associated functionality. Value After Reset: 0x0
1:0	cscscale	R/W	Defines the cscscale[1:0] scale factor to apply to all coefficients in Color Space Conversion. This scale factor is expressed in the number of left shifts to apply to each of the coefficients, ranging from 0 to 2. Value After Reset: 0x1

csc_coef_a1_msb

Description: Color Space Converter Matrix A1 Coefficient Register MSB Notes:
The coefficients used in the CSC matrix use only 15 bits for the internal computations.
Coefficients are represented in 2's complementary format and stored in two registers:
csc_coef_*_lsb[7:0]: coefficient bits 7 to 0
csc_coef_*_msb[7]: spare bit
csc_coef_*_msb[6:0]: coefficient bits 14 to 8
Examples for standard ITU601 and ITU709 RGB/YCC conversion CSC coefficients exist in

the Video Datapath Application Note.
Size: 8 bits
Offset: 0x4102

Bits	Name	Attr	Description
7:0	csc_coef_a1_msb	R/W	Color Space Converter Matrix A1 Coefficient Register MSB Value After Reset: 0x20

csc_coef_a1_lsb
Description: Color Space Converter Matrix A1 Coefficient Register LSB Notes:
The coefficients used in the CSC matrix use only 15 bits for the internal computations.
Coefficients are represented in 2's complementary format and stored in two registers:
csc_coef_*_lsb[7:0]: coefficient bits 7 to 0
csc_coef_*_msb[7]: spare bit
csc_coef_*_msb[6:0]: coefficient bits 14 to 8
Examples for standard ITU601 and ITU709 RGB/YCC conversion CSC coefficients exist in the Video Datapath Application Note.
Size: 8 bits
Offset: 0x4103

Bits	Name	Attr	Description
7:0	csc_coef_a1_lsb	R/W	Color Space Converter Matrix A1 Coefficient Register LSB Value After Reset: 0x0

csc_coef_a2_msb
Description: Color Space Converter Matrix A2 Coefficient Register MSB Color Space Conversion A2 coefficient.
Size: 8 bits
Offset: 0x4104

Bits	Name	Attr	Description
7:0	csc_coef_a2_msb	R/W	Color Space Converter Matrix A2 Coefficient Register MSB Value After Reset: 0x0

csc_coef_a2_lsb
Description: Color Space Converter Matrix A2 Coefficient Register LSB Color Space Conversion A2 coefficient.
Size: 8 bits
Offset: 0x4105

Bits	Name	Attr	Description
7:0	csc_coef_a2_lsb	R/W	Color Space Converter Matrix A2 Coefficient Register LSB Value After Reset: 0x0

csc_coef_a3_msb
Description: Color Space Converter Matrix A3 Coefficient Register MSB Color Space Conversion A3 coefficient.

Size: 8 bits
Offset: 0x4106

Bits	Name	Attr	Description
7:0	csc_coef_a3_msb	R/W	Color Space Converter Matrix A3 Coefficient Register MSB Value After Reset: 0x0

csc_coef_a3_lsb
Description: Color Space Converter Matrix A3 Coefficient Register LSB Color Space Conversion A3 coefficient.
Size: 8 bits
Offset: 0x4107

Bits	Name	Attr	Description
7:0	csc_coef_a3_lsb	R/W	Color Space Converter Matrix A3 Coefficient Register LSB Value After Reset: 0x0

csc_coef_a4_msb
Description: Color Space Converter Matrix A4 Coefficient Register MSB Color Space Conversion A4 coefficient.
Size: 8 bits
Offset: 0x4108

Bits	Name	Attr	Description
7:0	csc_coef_a4_msb	R/W	Color Space Converter Matrix A4 Coefficient Register MSB Value After Reset: 0x0

csc_coef_a4_lsb
Description: Color Space Converter Matrix A4 Coefficient Register LSB Color Space Conversion A4 coefficient.
Size: 8 bits
Offset: 0x4109

Bits	Name	Attr	Description
7:0	csc_coef_a4_lsb	R/W	Color Space Converter Matrix A4 Coefficient Register LSB Value After Reset: 0x0

csc_coef_b1_msb
Description: Color Space Converter Matrix B1 Coefficient Register MSB Color Space Conversion B1 coefficient.
Size: 8 bits
Offset: 0x410a

Bits	Name	Attr	Description
7:0	csc_coef_b1_msb	R/W	Color Space Converter Matrix B1 Coefficient Register MSB

			Value After Reset: 0x0

csc_coef_b1_lsb

Description: Color Space Converter Matrix B1 Coefficient Register LSB Color Space Conversion B1 coefficient.

Size: 8 bits

Offset: 0x410b

Bits	Name	Attr	Description
7:0	csc_coef_b1_lsb	R/W	Color Space Converter Matrix B1 Coefficient Register LSB Value After Reset: 0x0

csc_coef_b2_msb

Description: Color Space Converter Matrix B2 Coefficient Register MSB Color Space Conversion B2 coefficient.

Size: 8 bits

Offset: 0x410c

Bits	Name	Attr	Description
7:0	csc_coef_b2_msb	R/W	Color Space Converter Matrix B2 Coefficient Register MSB Value After Reset: 0x20

csc_coef_b2_lsb

Description: Color Space Converter Matrix B2 Coefficient Register LSB Color Space Conversion B2 coefficient.

Size: 8 bits

Offset: 0x410d

Bits	Name	Attr	Description
7:0	csc_coef_b2_lsb	R/W	Color Space Converter Matrix B2 Coefficient Register LSB Value After Reset: 0x0

csc_coef_b3_msb

Description: Color Space Converter Matrix B3 Coefficient Register MSB Color Space Conversion B3 coefficient.

Size: 8 bits

Offset: 0x410e

Bits	Name	Attr	Description
7:0	csc_coef_b3_msb	R/W	Color Space Converter Matrix B3 Coefficient Register MSB Value After Reset: 0x0

csc_coef_b3_lsb

Description: Color Space Converter Matrix B3 Coefficient Register LSB Color Space Conversion B3 coefficient.

Size: 8 bits
Offset: 0x410f

Bits	Name	Attr	Description
7:0	csc_coef_b3_lsb	R/W	Color Space Converter Matrix B3 Coefficient Register LSB Value After Reset: 0x0

csc_coef_b4_msb
Description: Color Space Converter Matrix B4 Coefficient Register MSB Color Space Conversion B4 coefficient.
Size: 8 bits
Offset: 0x4110

Bits	Name	Attr	Description
7:0	csc_coef_b4_msb	R/W	Color Space Converter Matrix B4 Coefficient Register MSB Value After Reset: 0x0

csc_coef_b4_lsb
Description: Color Space Converter Matrix B4 Coefficient Register LSB Color Space Conversion B4 coefficient.
Size: 8 bits
Offset: 0x4111

Bits	Name	Attr	Description
7:0	csc_coef_b4_lsb	R/W	Color Space Converter Matrix B4 Coefficient Register LSB Value After Reset: 0x0

csc_coef_c1_msb
Description: Color Space Converter Matrix C1 Coefficient Register MSB Color Space Conversion C1 coefficient.
Size: 8 bits
Offset: 0x4112

Bits	Name	Attr	Description
7:0	csc_coef_c1_msb	R/W	Color Space Converter Matrix C1 Coefficient Register MSB Value After Reset: 0x0

csc_coef_c1_lsb
Description: Color Space Converter Matrix C1 Coefficient Register LSB Color Space Conversion C1 coefficient.
Size: 8 bits
Offset: 0x4113

Bits	Name	Attr	Description
7:0	csc_coef_c1_lsb	R/W	Color Space Converter Matrix C1 Coefficient Register LSB Value After Reset: 0x0

csc_coef_c2_msb

Description: Color Space Converter Matrix C2 Coefficient Register MSB Color Space Conversion C2 coefficient.

Size: 8 bits

Offset: 0x4114

Bits	Name	Attr	Description
7:0	csc_coef_c2_msb	R/W	Color Space Converter Matrix C2 Coefficient Register MSB Value After Reset: 0x0

csc_coef_c2_lsb

Description: Color Space Converter Matrix C2 Coefficient Register LSB Color Space Conversion C2 coefficient.

Size: 8 bits

Offset: 0x4115

Bits	Name	Attr	Description
7:0	csc_coef_c2_lsb	R/W	Color Space Converter Matrix C2 Coefficient Register LSB Value After Reset: 0x0

csc_coef_c3_msb

Description: Color Space Converter Matrix C3 Coefficient Register MSB Color Space Conversion C3 coefficient.

Size: 8 bits

Offset: 0x4116

Bits	Name	Attr	Description
7:0	csc_coef_c3_msb	R/W	Color Space Converter Matrix C3 Coefficient Register MSB Value After Reset: 0x20

csc_coef_c3_lsb

Description: Color Space Converter Matrix C3 Coefficient Register LSB Color Space Conversion C3 coefficient.

Size: 8 bits

Offset: 0x4117

Bits	Name	Attr	Description
7:0	csc_coef_c3_lsb	R/W	Color Space Converter Matrix C3 Coefficient Register LSB Value After Reset: 0x0

csc_coef_c4_msb

Description: Color Space Converter Matrix C4 Coefficient Register MSB Color Space Conversion C4 coefficient.

Size: 8 bits

Offset: 0x4118

Bits	Name	Attr	Description

7:0	csc_coef_c4_msb	R/W	Description: Color Space Converter Matrix C4 Coefficient Register MSB Value After Reset: 0x0

csc_coef_c4_lsb

Description: Color Space Converter Matrix C4 Coefficient Register LSB Color Space Conversion C4 coefficient.

Size: 8 bits

Offset: 0x4119

Bits	Name	Attr	Description
7:0	csc_coef_c4_lsb	R/W	Color Space Converter Matrix C4 Coefficient Register LSB Value After Reset: 0x0

csc_limit_up_msb

Description: Color Space Converter Matrix Output Up Limit Register MSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 254, and this register must be configured with 0x00.

Size: 8 bits

Offset: 0x411a

Bits	Name	Attr	Description
7:0	csc_limit_up_msb	R/W	Color Space Converter Matrix Output Upper Limit Register MSB Value After Reset: 0xff

csc_limit_up_lsb

Description: Color Space Converter Matrix output Up Limit Register LSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 254, and this register must be configured with 0xFE.

Size: 8 bits

Offset: 0x411b

Bits	Name	Attr	Description
7:0	csc_limit_up_lsb	R/W	Color Space Converter Matrix Output Upper Limit Register LSB Value After Reset: 0xff

csc_limit_dn_msb

Description: Color Space Converter Matrix output Down Limit Register MSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 1, and this register must be configured with 0x00.

Size: 8 bits

Offset: 0x411c

Bits	Name	Attr	Description

7:0	csc_limit_dn_msb	R/W	Color Space Converter Matrix output Down Limit Register MSB Value After Reset: 0x0

csc_limit_dn_lsb

Description: Color Space Converter Matrix output Down Limit Register LSB
For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 1, and this register must be configured with 0x01.
Size: 8 bits
Offset: 0x411d

Bits	Name	Attr	Description
7:0	csc_limit_dn_lsb	R/W	Color Space Converter Matrix Output Down Limit Register LSB Value After Reset: 0x0

HDCP Registers

HDCP Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
a_hdcpcfg0	0x5000	HDCP Enable and Functional Control Configuration Register 0
a_hdcpcfg1	0x5001	HDCP Software Reset and Functional Control Configuration Register 1
a_hdcpobs0	0x5002	HDCP Observation Register 0
a_hdcpobs1	0x5003	HDCP Observation Register 1
a_hdcpobs2	0x5004	HDCP Observation Register 2
a_hdcpobs3	0x5005	HDCP Observation Register 3
a_apiintclr	0x5006	HDCP Interrupt Clear Register Write only register, active high and auto cleared, cleans the respective...
a_apiintstat	0x5007	HDCP Interrupt Status Register Read only register, reports the interruption which caused the activation...
a_apiintmsk	0x5008	HDCP Interrupt Mask Register The configuration of this register mask a given setup of interruption,...
a_vidpolcfg	0x5009	HDCP Video Polarity Configuration Register
a_oesswcfg	0x500a	HDCP OESS WOO Configuration Register Pulse width of the encryption enable (CTL3) signal in the...
a_coreverlsb	0x5014	HDCP Controller Version Register LSB Design ID number.
a_corevermsb	0x5015	HDCP Controller Version Register MSB Revision ID number.
a_ksvmemctrl	0x5016	HDCP KSV Memory Control Register The KSVCTRLupd bit is a notification flag. This flag changes polarity...

Register	Offset	Description
hdcp_bstatus[0:1]	0x5020 + (i * 0x1)	HDCP BStatus Register Array
hdcp_m0[0:7]	0x5022 + (i * 0x1)	HDCP M0 Register Array
hdcp_ksv[0:634]	0x502a + (i * 0x1)	HDCP KSV Registers
hdcp_vh[0:19]	0x52a5 + (i * 0x1)	HDCP SHA-1 VH Registers
hdcp_revoc_size_0	0x52b9	HDCP Revocation KSV List Size Register 0
hdcp_revoc_size_1	0x52ba	HDCP Revocation KSV List Size Register 1
hdcp_revoc_list[0:5059]	0x52bb + (i * 0x1)	HDCP Revocation KSV Registers
hdcpreg_bksv0	0x7800	HDCP KSV Status Register 0

Registers for Address Block: HDCP (Continued)

Register	Offset	Description
hdcpreg_bksv1	0x7801	HDCP KSV Status Register 1
hdcpreg_bksv2	0x7802	HDCP KSV Status Register 2
hdcpreg_bksv3	0x7803	HDCP KSV Status Register 3
hdcpreg_bksv4	0x7804	HDCP KSV Status Register 4
hdcpreg_anconf	0x7805	HDCP AN Bypass Control Register
hdcpreg_an0	0x7806	HDCP Forced AN Register 0
hdcpreg_an1	0x7807	HDCP Forced AN Register 1
hdcpreg_an2	0x7808	HDCP forced AN Register 2
hdcpreg_an3	0x7809	HDCP Forced AN Register 3
hdcpreg_an4	0x780a	HDCP Forced AN Register 4
hdcpreg_an5	0x780b	HDCP Forced AN Register 5
hdcpreg_an6	0x780c	HDCP Forced AN Register 6
hdcpreg_an7	0x780d	HDCP Forced AN Register 7
hdcpreg_rmlctl	0x780e	HDCP Encrypted Device Private Keys Control Register This register is the control register for the...
hdcpreg_rmlsts	0x780f	HDCP Encrypted DPK Status Register The required software configuration sequence is documented in...
hdcpreg_seed0	0x7810	HDCP Encrypted DPK Seed Register 0 This register contains a byte of the HDCP Encrypted DPK seed...

Register	Offset	Description
hdcpreg_seed1	0x7811	HDCP Encrypted DPK Seed Register 1 This register contains a byte of the HDCP Encrypted DPK seed...
hdcpreg_dpk0	0x7812	HDCP Encrypted DPK Data Register 0 This register contains an HDCP DPK byte. The required software...
hdcpreg_dpk1	0x7813	HDCP Encrypted DPK Data Register 1 This register contains an HDCP DPK byte. The required software...
hdcpreg_dpk2	0x7814	HDCP Encrypted DPK Data Register 2 This register contains an HDCP DPK byte. The required software...
hdcpreg_dpk3	0x7815	HDCP Encrypted DPK Data Register 3 This register contains an HDCP DPK byte. The required software...
hdcpreg_dpk4	0x7816	HDCP Encrypted DPK Data Register 4 This register contains an HDCP DPK byte. The required software...
hdcpreg_dpk5	0x7817	HDCP Encrypted DPK Data Register 5 This register contains an HDCP DPK byte. The required software...
hdcpreg_dpk6	0x7818	HDCP Encrypted DPK Data Register 6 This register contains an HDCP DPK byte. The required software...

a_hdcpcfg0

Description: HDCP Enable and Functional Control Configuration Register 0

Size: 8 bits

Offset: 0x5000

Bits	Name	Attr	Description
7	ELVena	R/W	Enables the Enhanced Link Verification from the transmitter's side Value After Reset: 0x0
6	I2Cfastmode	R/W	Enable the I2C fast mode option from the transmitter's side. Value After Reset: 0x0
5	bypencryption	R/W	Bypasses all the data encryption stages Value After Reset: "(HDMI_HDCP_BYPASS== 1) ? 1 : 0"
4	syncricheck	R/W	Configures if the Ri check should be done at every 2s even or synchronously to every 128 encrypted frame. Value After Reset: 0x0
3	avmute	R	This register holds the current AVMUTE state of the Hdmi_tx controller, as expected to be perceived by the connected HDMI/HDCP sink device.

			Value After Reset: 0x0
2	rxdetect	R/W	Information that a sink device was detected connected to the HDMI port Value After Reset: 0x0
1	en11feature	R/W	Enable the use of features 1.1 from the transmitter's side Value After Reset: 0x0
0	hdmidvi	R/W	Configures the transmitter to operate with a HDMI capable device or with a DVI device. Value After Reset: 0x0

a_hdcpcfg1

Description: HDCP Software Reset and Functional Control Configuration Register 1

Size: 8 bits

Offset: 0x5001

Bits	Name	Attr	Description
7:5	spare	R/W	Reserved as "spare" register with no associated functionality. Value After Reset: 0x0
4	hdcp_lock	R/W	Lock the HDCP bypass and encryption disable mechanisms: 1'b0: The default 1'b0 value enables you to bypass HDCP through bit 5 (bypencryption) of the A_HDCPCFG0 register or to disable the encryption through bit 1 (encryptiondisable) of A_HDCPCFG1. 1'b1: You can still write to the bit bypencryption of A_HDCPCFG0 or encryptiondisable bit of A_HDCPCFG1 but you cannot enable the bypass. Once you set the value to 1'b1, you can change the value back to 1'b0 only by issuing a master reset to the Hdmi_tx. Value After Reset: 0x0
3	dissha1check	R/W	Disables the request to the API processor to verify the SHA1 message digest of a received KSV List Value After Reset: 0x0
2	ph2upshftenc	R/W	Enables the encoding of packet header in the tmdsch0 bit[0] with cipher[2] instead of the tmdsch0 bit[2] Note: This bit must always be set to 1 for all PHYs. Value After Reset: 0x0
1	encryptiondisable	R/W	Disable encryption without losing authentication Value After Reset: 0x0
0	swreset	R/W	Software reset signal, active by writing a zero and auto cleared to 1 in the following cycle. Value After Reset: 0x1

a_hdcpobs0

Description: HDCP Observation Register 0
Size: 8 bits
Offset: 0x5002

Bits	Name	Attr	Description
7:4	STATEA	R	Observability register informs in which state the authentication machine is on. Value After Reset: 0x0
3:1	SUBSTATEA	R	Observability register informs in which sub-state the authentication is on. Value After Reset: 0x0
0	hdcpengaged	R	Informs that the current HDMI link has the HDCP protocol fully engaged. Value After Reset: 0x0

a_hdcpobs1
Description: HDCP Observation Register 1
Size: 8 bits
Offset: 0x5003

Bits	Name	Attr	Description
7			Reserved for future use.
6:4	STATEOEG	R	Observability register informs in which state the OESS machine is on. Value After Reset: 0x0
3:0	STATER	R	Observability register informs in which state the revocation machine is on. Value After Reset: 0x0

a_hdcpobs2
Description: HDCP Observation Register 2
Size: 8 bits
Offset: 0x5004

Bits	Name	Attr	Description
7:6			Reserved for future use.
5:3	STATEE	R	Observability register informs in which state the cipher machine is on. Value After Reset: 0x0
2:0	STATEEEG	R	Observability register informs in which state the EESS machine is on. Value After Reset: 0x0

a_hdcpobs3
Description: HDCP Observation Register 3
Size: 8 bits
Offset: 0x5005

Bits	Name	Attr	Description
7	HDMI_RESERVED_1	R	Register read from attached sink device:

Bits	Name	Attr	Description
			Bcap(0x40) bit 7. Value After Reset: 0x0
6	REPEATER	R	Register read from attached sink device: Bcap(0x40) bit 6. Value After Reset: 0x0
5	KSV_FIFO_READY	R	Register read from attached sink device: Bcap(0x40) bit 5. Value After Reset: 0x0
4	FAST_I2C	R	Register read from attached sink device: Bcap(0x40) bit 4. Value After Reset: 0x0
3	HDMI_RESERVED_2	R	Register read from attached sink device: Bstatus(0x41) bit 13. Value After Reset: 0x0
2	HDMI_MODE	R	Register read from attached sink device: Bstatus(0x41) bit 12. Value After Reset: 0x0
1	FEATURES_1_1	R	Register read from attached sink device: Bcap(0x40) bit 1. Value After Reset: 0x0
0	FAST_REAUTHENTICATION	R	Register read from attached sink device: Bcap(0x40) bit 0. Value After Reset: 0x0

a_apiintclr

Description: HDCP Interrupt Clear Register

Write only register, active high and auto cleared, cleans the respective interruption in the interrupt status register.

Size: 8 bits

Offset: 0x5006

Bits	Name	Attr	Description
7	HDCP_engaged	W	Clears the interruption related to HDCP authentication process successful. Value After Reset: 0x0
6	HDCP_failed	W	Clears the interruption related to HDCP authentication process failed. Value After Reset: 0x0
5	KSVsha1calcdoneint	W	Clears the interruption related to SHA1 verification has been done Value After Reset: 0x0
4	I2Cnack	W	Clears the interruption related to I2C NACK reception. Value After Reset: 0x0
3	Lostarbitration	W	Clears the interruption related to I2C arbitration lost.

			Value After Reset: 0x0
2	Keepouterrorint	W	Clears the interruption related to keep out window error. Value After Reset: 0x0
1	KSVsha1calcint	W	Clears the interruption related to KSV list update in memory that needs to be SHA1 verified. Value After Reset: 0x0
0	KSVaccessint	W	Clears the interruption related to KSV Attr grant for Read-Write access. Value After Reset: 0x0

a_apiintstat

Description: HDCP Interrupt Status Register

Read only register, reports the interruption which caused the activation of the interruption output pin.

Size: 8 bits

Offset: 0x5007

Bits	Name	Attr	Description
7	HDCP_engaged	R	Notifies that the HDCP authentication process was successful Value After Reset: 0x0
6	HDCP_failed	R	Notifies that the HDCP authentication process was failed. Value After Reset: 0x0
5	KSVsha1calcdoneint	R	Notifies that the HDCP13TCTRL controller SHA1 verification has been done. The status ready to be read. Value After Reset: 0x0
4	I2Cnack	R	Notifies that the I2C received a NACK from slave device. Value After Reset: 0x0
3	Lostarbitration	R	Notifies that the I2C lost the arbitration to communicate. Another master gained arbitration. Value After Reset: 0x0
2	Keepouterrorint	R	Notifies that during the keep out window, the ctlout[3:0] bus was used besides control period. Value After Reset: 0x0
1	KSVsha1calcint	R	Notifies that the HDCP13TCTRL controller as updated a KSV list in memory that needs to be SHA1 verified. Value After Reset: 0x0
0	KSVaccessint	R	Notifies that the KSV Attr as been guaranteed for Read-Write access. Value After Reset: 0x0

a_apiintmsk

Description: HDCP Interrupt Mask Register

The configuration of this register mask a given setup of interruption, disabling them from generating interruption pulses in the interruption output pin.
Size: 8 bits
Offset: 0x5008

Bits	Name	Attr	Description
7	HDCP_engaged	R/W	Masks the interruption related to HDCP authentication process successful. Value After Reset: 0x0
6	HDCP_failed	R/W	Masks the interruption related to HDCP authentication process failed. Value After Reset: 0x0
5	KSVsha1calcdoneint	R/W	Masks the interruption related to SHA1 verification has been done Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
4	I2Cnack	R/W	Masks the interruption related to I2C NACK reception. Value After Reset: 0x0
3	Lostarbitration	R/W	Masks the interruption related to I2C arbitration lost. Value After Reset: 0x0
2	Keepouterrorint	R/W	Masks the interruption related to keep out window error. Value After Reset: 0x0
1	KSVsha1calcint	R/W	Masks the interruption related to KSV list update in memory that needs to be SHA1 verified. Otherwise, this field is a "spare" bit with no associated functionality. Value After Reset: 0x0
0	KSVaccessint	R/W	Masks the interruption related to KSV Attr grant for Read-Write access. Value After Reset: 0x0

a_vidpolcfg
Description: HDCP Video Polarity Configuration Register
Size: 8 bits
Offset: 0x5009

Bits	Name	Attr	Description
7			Reserved for future use.
6:5	unencryptconf	R/W	Configuration of the color sent when sending unencrypted video data Value After Reset: 0x0
4	dataenpol	R/W	Configuration of the video data enable polarity Value After Reset: 0x0
3	vsyncpol	R/W	Configuration of the video Vertical synchronism polarity

			Value After Reset: 0x0
2	spare_2	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0
1	hsyncpol	R/W	Configuration of the video Horizontal synchronism polarity. Value After Reset: 0x0
0	spare_1	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x0

a_oesswcfg

Description: HDCP OESS WOO Configuration Register

Pulse width of the encryption enable (CTL3) signal in the HDCP OESS mode. The window of opportunity for the Original Encryption Status Signaling starts at the active edge of the Vertical synchronism and stops after oesswindowoffset[7:0]*4 clock cycles of TMDS clock. According to the HDCP specification, the CTL3 signal must be asserted at least for eight TMDS clock cycles (oesswindowoffset[7:0] must be greater than 1), and it is recommended to transmit a larger pulse width for enhanced link reliability.

Size: 8 bits

Offset: 0x500a

Bits	Name	Attr	Description
7:0	a_oesswcfg	R/W	HDCP OESS WOO Configuration Register Value After Reset: 0x80

a_coreverlsb

Description: HDCP Controller Version Register LSB Design ID number.

Size: 8 bits

Offset: 0x5014

Bits	Name	Attr	Description
7:0	a_coreverlsb	R	HDCP Controller Version Register LSB Value After Reset: 0x2

a_corevermsb

Description: HDCP Controller Version Register MSB Revision ID number.

Size: 8 bits

Offset: 0x5015

Bits	Name	Attr	Description
7:0	a_corevermsb	R	HDCP Controller Version Register MSB Value After Reset: 0x3

a_ksvmemctrl

Description: HDCP KSV Memory Control Register

The KSVCTRLupd bit is a notification flag. This flag changes polarity whenever the register is written. This flag acts as a trigger to other blocks that processes this data. Upon reset the flag returns to low default value.

Size: 8 bits

Offset: 0x5016

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	KSVsha1status	R	Notification whether the KSV list message digest is correct from the controller: 1'b1 if digest message verification failed 1'b0 if digest message verification succeeded Value After Reset: 0x0
3	SHA1fail	R/W	Notification whether the KSV list message digest is correct. Value After Reset: 0x0
2	KSVCTRLupd	R/W	Set to inform that the KSV list in memory has been analyzed and the response to the Message Digest has been updated if on configurations on software SHA-1 calculation. Value After Reset: 0x0
1	KSVMEMaccess	R	Notification that the KSV Attr as been guaranteed. Value After Reset: 0x0
0	KSVMEMrequest	R/W	Request access to the KSV memory; must be de-asserted after the access is completed by the system. Value After Reset: 0x0

hdcp_bstatus[0:1]

Description: HDCP BStatus Register Array

Size: 8 bits

Offset: 0x5020 + (i * 0x1)

Bits	Name	Attr	Description
7:0	bstatus	R/W	HDCP BSTATUS[15:0]. If Attr has not been granted (see register a_ksvmemctrl), the value read will be 8'hff. Value After Reset: 0xff

hdcp_m0[0:7]

Description: HDCP M0 Register Array

Size: 8 bits

Offset: 0x5022 + (i * 0x1)

Bits	Name	Attr	Description
7:0	M0	R/W	HDCP M0[32:0]. If Attr has not been granted (see register a_ksvmemctrl) , the value read will be 8'hff. These values are only available on a configuration that has the SHA1 calculation by software. Value After Reset: 0xff

hdcp_ksv[0:634]

Description: HDCP KSV Registers.

Size: 8 bits
Offset: 0x502a + (i * 0x1)

Bits	Name	Attr	Description
7:0	hdcp_ksv_byte	R/W	Sink KSV FIFO byte, ordered in little endian (byte at address 0x502a belongs to byte 0 of KSV0). If Attr has not been granted (see register a_ksvmemctrl), the value read is 8'hff. In this address space, 635 KSV FIFO bytes are mapped, which allow for 127 KSV values, each with 5 bytes (40 bits). Value After Reset: 0xff

hdcp_vh[0:19]
Description: HDCP SHA-1 VH Registers.
Size: 8 bits
Offset: 0x52a5 + (i * 0x1)

Bits	Name	Attr	Description
7:0	hdcp_vh_byte	R/W	Sink VH' byte, ordered in little endian (byte at address 0x525a belongs to byte 0 of VH0). If Attr has not been granted (see register a_ksvmemctrl), the value read is 8'hff. In this address space 20 VH bytes are mapped, which allow for 5 VH values, each with 4 bytes (32bits). Value After Reset: 0xff

hdcp_revoc_size_0
Description: HDCP Revocation KSV List Size Register 0
Size: 8 bits
Offset: 0x52b9

Bits	Name	Attr	Description
7:0	hdcp_revoc_size_0	R/W	Register containing the LSB of KSV list size (ksv_list_size[7:0]). If Attr has not been granted (see register a_ksvmemctrl), the value read is 8'hff. Value After Reset: 0xff

hdcp_revoc_size_1
Description: HDCP Revocation KSV List Size Register 1
Size: 8 bits
Offset: 0x52ba

Bits	Name	Attr	Description
7:0	hdcp_revoc_size_1	R/W	Register containing the MSB of KSV list size (ksv_list_size[15:8]). If Attr has not been granted (see register a_ksvmemctrl), the value read is 8'hff. Value After Reset: 0xff

hdcp_revoc_list[0:5059]

Description: HDCP Revocation KSV Registers.
Size: 8 bits
Offset: 0x52bb + (i * 0x1)

Bits	Name	Attr	Description
7:0	hdcp_revoc_list_ksv_byte	R/W	Revocation KSV byte, ordered in little endian (byte at address 0x52bb belongs to byte 0 of the first revoked KSV). If Attr has not been granted (see register a_ksvmemctrl), the value read is 8'hff. In this address space 5060 revoked KSV bytes are mapped, which allow for 1012 KSV values, each with 5 bytes (40 bits). Value After Reset: 0xff

hdcpreg_bksv0
Description: HDCP KSV Status Register 0
Size: 8 bits
Offset: 0x7800

Bits	Name	Attr	Description
7:0	hdcpreg_bksv0	R	Contains the value of BKSV[7:0]. Value After Reset: 0x0

hdcpreg_bksv1
Description: HDCP KSV Status Register 1
Size: 8 bits
Offset: 0x7801

Bits	Name	Attr	Description
7:0	hdcpreg_bksv1	R	Description: Contains the value of BKSV[15:8]. Value After Reset: 0x0

hdcpreg_bksv2
Description: HDCP KSV Status Register 2
Size: 8 bits
Offset: 0x7802

Bits	Name	Attr	Description
7:0	hdcpreg_bksv2	R	Contains the value of BKSV[23:16]. Value After Reset: 0x0

hdcpreg_bksv3
Description: HDCP KSV Status Register 3
Size: 8 bits
Offset: 0x7803

Bits	Name	Attr	Description
7:0	hdcpreg_bksv3	R	Contains the value of BKSV[31:24]. Value After Reset: 0x0

hdcpreg_bksv4
Description: HDCP KSV Status Register 4

Size: 8 bits
Offset: 0x7804

Bits	Name	Attr	Description
7:0	hdcpreg_bksv4	R	Contains the value of BKSV[39:32]. Value After Reset: 0x0

hdcpreg_anconf
Description: HDCP AN Bypass Control Register
Size: 8 bits
Offset: 0x7805

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	oanbypass	R/W	When oanbypass=1, the value of AN used in the HDCP engine comes from the hdcpreg_an0 to hdcpreg_an7 registers. When oanbypass=0, the value of AN used in the HDCP engine comes from the random number input. Value After Reset: 0x0

hdcpreg_an0
Description: HDCP Forced AN Register 0
Size: 8 bits
Offset: 0x7806

Bits	Name	Attr	Description
7:0	hdcpreg_an0	R/W	Contains the value of AN[7:0] Value After Reset: 0x0

hdcpreg_an1
Description: HDCP Forced AN Register 1
Size: 8 bits
Offset: 0x7807

Bits	Name	Attr	Description
7:0	hdcpreg_an1	R/W	Contains the value of AN[15:8] Value After Reset: 0x0

hdcpreg_an2
Description: HDCP forced AN Register 2
Size: 8 bits
Offset: 0x7808

Bits	Name	Attr	Description
7:0	hdcpreg_an2	R/W	Contains the value of AN[23:16] Value After Reset: 0x0

hdcpreg_an3
Description: HDCP Forced AN Register 3
Size: 8 bits
Offset: 0x7809

Bits	Name	Attr	Description
7:0	hdcpreg_an3	R/W	Contains the value of AN[31:24] Value After Reset: 0x0

hdcpreg_an4

Description: HDCP Forced AN Register 4

Size: 8 bits

Offset: 0x780a

Bits	Name	Attr	Description
7:0	hdcpreg_an4	R/W	Contains the value of AN[39:32] Value After Reset: 0x0

hdcpreg_an5

Description: HDCP Forced AN Register 5

Size: 8 bits

Offset: 0x780b

Bits	Name	Attr	Description
7:0	hdcpreg_an5	R/W	Contains the value of AN[47:40] Value After Reset: 0x0

hdcpreg_an6

Description: HDCP Forced AN Register 6

Size: 8 bits

Offset: 0x780c

Bits	Name	Attr	Description
7:0	hdcpreg_an6	R/W	Contains the value of AN[55:48] Value After Reset: 0x0

hdcpreg_an7

Description: HDCP Forced AN Register 7

Size: 8 bits

Offset: 0x780d

Bits	Name	Attr	Description
7:0	hdcpreg_an7	R/W	Contains the value of BKSV[63:56] Value After Reset: 0x0

hdcpreg_rmlctl

Description: HDCP Encrypted Device Private Keys Control Register

This register is the control register for the software programmable encrypted DPK embedded storage feature. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."

Size: 8 bits

Offset: 0x780e

Bits	Name	Attr	Description
7:1			Reserved for future use.

0	odpk_decrypt_enable	R/W	When set (1'b1), this bit activates the decryption of the Device Private keys. Value After Reset: 0x0

hdcpreg_rmlsts

Description: HDCP Encrypted DPK Status Register

The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."

Size: 8 bits

Offset: 0x780f

Bits	Name	Attr	Description
7			Reserved for future use.
6	idpk_wr_ok_sts	R	When high (1'b1), it indicates that a DPK write is allowed. Value After Reset: 0x0
5:0	idpk_data_index	R	Current Device Private Key being written plus one. Position 0 is occupied by the AKSV. Value After Reset: 0x0

hdcpreg_seed0

Description: HDCP Encrypted DPK Seed Register 0

This register contains a byte of the HDCP Encrypted DPK seed value used to encrypt the Device Private Keys. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."

Size: 8 bits

Offset: 0x7810

Bits	Name	Attr	Description
7:0	hdcpreg_seed0	W	Least significant byte of the decryption seed value (dpk_decrypt_seed[7:0]). Value After Reset: 0x0

hdcpreg_seed1

Description: HDCP Encrypted DPK Seed Register 1

This register contains a byte of the HDCP Encrypted DPK seed value used to encrypt the Device Private Keys. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."

Size: 8 bits

Offset: 0x7811

Bits	Name	Attr	Description
7:0	hdcpreg_seed1	W	Most significant byte of the decryption seed value (dpk_decrypt_seed[15:8]). Value After Reset: 0x0

hdcpreg_dpk0

Description: HDCP Encrypted DPK Data Register 0

This register contains an HDCP DPK byte. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."
Size: 8 bits
Offset: 0x7812

Bits	Name	Attr	Description
7:0	dpk_data	W	Byte of the encrypted DPK value. dpk[7:0] When this byte is written, a strobe signal is generated that triggers the decryption and/or storage of the DPK word on the DPK internal RAM memory. Value After Reset: 0x0

hdcpreg_dpk1
Description: HDCP Encrypted DPK Data Register 1
This register contains an HDCP DPK byte. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."
Size: 8 bits
Offset: 0x7813

Bits	Name	Attr	Description
7:0	dpk_data	W	Byte of the encrypted DPK value. dpk[15:8] Value After Reset: 0x0

hdcpreg_dpk2
Description: HDCP Encrypted DPK Data Register 2
This register contains an HDCP DPK byte. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."
Size: 8 bits
Offset: 0x7814

Bits	Name	Attr	Description
7:0	dpk_data	W	Byte of the encrypted DPK value. dpk[23:16] Value After Reset: 0x0

hdcpreg_dpk3
Description: HDCP Encrypted DPK Data Register 3
This register contains an HDCP DPK byte. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."
Size: 8 bits
Offset: 0x7815

Bits	Name	Attr	Description
7:0	dpk_data	W	Byte of the encrypted DPK value. dpk[31:24] Value After Reset: 0x0

hdcpreg_dpk4
Description: HDCP Encrypted DPK Data Register 4

This register contains an HDCP DPK byte.
Size: 8 bits
Offset: 0x7816

Bits	Name	Attr	Description
7:0	dpk_data	W	Byte of the encrypted DPK value. dpk[39:32] Value After Reset: 0x0

hdcpreg_dpk5
Description: HDCP Encrypted DPK Data Register 5
This register contains an HDCP DPK byte. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."
Size: 8 bits
Offset: 0x7817

Bits	Name	Attr	Description
7:0	dpk_data	W	Contains the value of DPK[x][47:40] Value After Reset: 0x0

hdcpreg_dpk6
Description: HDCP Encrypted DPK Data Register 6
This register contains an HDCP DPK byte. The required software configuration sequence is documented in the Cores HDMI Transmitter User Guide in the "Programming" chapter, Section 3.2.4, "Configure HDCP."
Size: 8 bits
Offset: 0x7818

Bits	Name	Attr	Description
7:0	dpk_data	W	Contains the value of DPK[x][55:48] Value After Reset: 0x0

HDCP22 Registers
HDCP22 Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
hdcp22reg_id	0x7900	HDCP 2.2 Identification Register
hdcp22reg_ctrl	0x7904	HDCP 2.2 Control Register
hdcp22reg_ctrl1	0x7905	HDCP 2.2 Control Register 1
hdcp22reg_sts	0x7908	HDCP 2.2 Status Register
hdcp22reg_mask	0x790c	HDCP 2.2 Interrupt Mask Register
hdcp22reg_stat	0x790d	HDCP 2.2 interrupt Sticky Bit Status Register
hdcp22reg_mute	0x790e	HDCP 2.2 Interrupt Mute Vector

hdcp22reg_id
Description: HDCP 2.2 Identification Register
Size: 8 bits
Offset: 0x7900

Bits	Name	Attr	Description
7:3			Reserved for future use.
2	hdcp22_3rdparty	R	Indicates that External HDCP 2.2 interface is present and 3rd party HDCP 2.2 module is connected to this interface. Value After Reset: "(HTX_HDCP22_EXTERNAL_ELLP== 1) ? 1 : 0"
1	hdcp22_externalif	R	Indicates that External HDCP 2.2 interface is present. Value After Reset: "(HTX_HDCP22_EXTERNAL_NONE== 1) ? 1 : 0"
0			Reserved for future use.

hdcp22reg_ctrl

Description: HDCP 2.2 Control Register

Size: 8 bits

Offset: 0x7904

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	hpd_ovr_val	R/W	HPD Override Value 1'b0: If hpd_ovr_en is 1'b1 the HPD value to the HDCP 2.2 external interface is set to 1'b0. 1'b1: If hpd_ovr_en is 1'b1 the HPD value to the HDCP 2.2 external interface is set to 1'b1. Value After Reset: 0x0
4	hpd_ovr_en	R/W	HPD Override enable 1'b0: The HPD value to the HDCP 2.2 external interface comes from the PHY as in phy_stat0.HPD. 1'b1: The HPD value to the HDCP 2.2 external interface comes from hpd_ovr_val bit field. Value After Reset: 0x0
3			Reserved for future use.
2	hdcp22_ovr_val	R/W	HDCP 2.2 versus 1.4 switch override value 1'b0: The switch is routed to HDCP 1.4 signals when hdcp22_ovr_en is 1'b1 and hdcp22_switch_lock is not set to 1'b1. 1'b1: The switch is routed to HDCP 2.2 signals when hdcp22_ovr_en is 1'b1 and hdcp22_switch_lock is not set to 1'b1. Value After Reset: 0x0
1	hdcp22_ovr_en	R/W	HDCP 2.2 versus 1.4 switch override enable 1'b0: The switch is automatically controlled by the HDCP 2.2 controller using the ist_hdcp2_capable and ist_hdcp2_not_capable status level indications. If the HDCP 2.2 controller indicates ist_hdcp2_capable at 1'b1, the switch is routed to

Bits	Name	Attr	Description
			HDCP 2.2 signals. If the HDCP 2.2 controller indicates ist_hdcp2_not_capable at 1'b1, the switch is routed to HDCP 1.4 signals. 1'b1: The HDCP 2.2 ist_hdcp2_capable and ist_hdcp2_not_capable values are ignored, and the direction of the HDCP 2.2 versus 1.4 switch is directly controlled by the hdcp22_ovr_val. Value After Reset: 0x0
0	hdcp22_switch_lck	R/W	HDCP 2.2 switch lock 1'b0: Enables you to change the direction of the HDCP 2.2 versus 1.4 switch by using the hdcp22_ovr_en and hdcp22_ovr_val. 1'b1: You can still write to hdcp22_ovr_en and hdcp22_ovr_val but has no effect over the HDCP 2.2 versus 1.4 switch, that keeps as it was configured by hdcp22_ovr_en and hdcp22_ovr_val at the time the 1'b1 was written to this bit field. Once you set the value to 1'b1, you can change the value back to 1'b0 only by issuing a master reset to the Hdmi_tx. Value After Reset: 0x0

hdcp22reg_ctrl1

Description: HDCP 2.2 Control Register 1

Size: 8 bits

Offset: 0x7905

Bits	Name	Attr	Description
7:4	hdcp22_cd_ovr_val	R/W	HDCP color depth override value, which is sent through the HDCP 2.2 external interface when hdcp22reg_ctrl1.hdcp22_cd_ovr_en is set. For reference on the HDMI allowed values consult the HDMI 1.4b specification, Table 6-1. Value After Reset: 0x0
3	hdcp22_cd_ovr_en	R/W	HDCP 2.2 versus 1.4 color depth override enable: 1'b0: The default 1'b0 value indicates that the color depth sent to the external interface is the one configured in the vp_pr_cd.color_depth register field. 1'b1: Although the used color depth for pixel encoding is defined by the field vp_pr_cd.color_depth register, the color depth sent to the external interface is the one defined in register field hdcp22reg_ctrl1.hdcp22_cd_ovr_val. Value After Reset: 0x0

Bits	Name	Attr	Description
2			Reserved for future use.
1	hdcp22_avmute_ovr_val	R/W	HDCP AV_MUTE override value, which is sent through the HDCP 2.2 external interface when hdcp22reg_ctrl1.hdcp22_avmute_ovr_en is set. Value After Reset: 0x0
0	hdcp22_avmute_ovr_en	R/W	HDCP 2.2 versus 1.4 avmute override enable 1'b0: The default 1'b0 value indicates that the AVMUTE sent to the external interface is the one configured through register fields fc_gcp.set_avmute and fc_gcp.clear_avmute. 1'b1: Although the GCP packet sends the set_avmute or clear_avmute as configured in register fc_gcp, the AV_MUTE sent to the external interface is the one defined in register field hdcp22reg_ctrl1.hdcp22_avmute_ovr_val. Value After Reset: 0x0

hdcp22reg_sts

Description: HDCP 2.2 Status Register

Size: 8 bits

Offset: 0x7908

Bits	Name	Attr	Description
7:4			Reserved for future use.
3	hdcp_decrypted_sts	R	Value of HDCP 2.2 ist_hdcp_decrypted line. Provided for debug only Value After Reset: 0x0
2	hdcp22_switch_sts	R	HDCP 2.2 HDCP 2.2 versus 1.4 switch status after lock mechanism (hdcp22reg_ctrl.hdcp22_switch_lock, hdcp22reg_ctrl.hdcp22_ovr_en and hdcp22reg_ctrl.hdcp22_ovr_val). 1'b0: HDCP 1.4 selected 1'b1: HDCP 2.2 selected Value After Reset: 0x0
1	hdcp_avmute_sts	R	HDCP 2.2 AVMUTE external interface status. 1'b0: External HDCP used AVMUTE is clear 1'b1: External HDCP AVMUTE is set (audio/video should be muted) Value After Reset: 0x0
0	hdmi_hpd_sts	R	HDCP 2.2 HPD external interface status after lock mechanism (hdcp22reg_ctrl.hdcp22_switch_lock, hdcp22reg_ctrl.hdcp22_ovr_en and hdcp22reg_ctrl.hdcp22_ovr_val). 1'b0: Sink not detected (HPD line clear) 1'b1: Sink detected (HPD line set) Value After Reset: 0x0

hdcp22reg_mask

Description: HDCP 2.2 Interrupt Mask Register

Size: 8 bits

Offset: 0x790c

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	mask_hdcp_decrypted_chg	R/W	Active high interrupt mask to HDCP 2.2 decrypted value change interrupt status Value After Reset: 0x1
4	mask_hdcp_authentication_fail	R/W	Active high interrupt mask to HDCP 2.2 authentication fail interrupt status Value After Reset: 0x1
3	mask_hdcp_authenticated	R/W	Active high interrupt mask to HDCP 2.2 authenticated interrupt status Value After Reset: 0x1
2	mask_hdcp_authentication_lost	R/W	Active high interrupt mask to HDCP 2.2 authentication lost interrupt status Value After Reset: 0x1
1	mask_hdcp2_not_capable	R/W	Active high interrupt mask to HDCP 2.2 not capable rise interrupt status Value After Reset: 0x1
0	mask_hdcp2_capable	R/W	Active high interrupt mask to HDCP 2.2 capable rise interrupt status Value After Reset: 0x1

hdcp22reg_stat

Description: HDCP 2.2 interrupt Sticky Bit Status Register

Size: 8 bits

Offset: 0x790d

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	st_hdcp_decrypted_chg	R/W1C	HDCP 2.2 decrypted value change interrupt status sticky bit. Clear by Write 1 to this bit field Value After Reset: 0x0
4	st_hdcp_authentication_fail	R/W1C	HDCP 2.2 authentication fail interrupt status sticky bit. Clear by Write 1 to this bit field Value After Reset: 0x0
3	st_hdcp_authenticated	R/W1C	HDCP 2.2 authenticated interrupt status sticky bit. Clear by Write 1 to this bit field Value After Reset: 0x0
2	st_hdcp_authentication_lost	R/W1C	HDCP 2.2 authentication lost interrupt status sticky bit. Clear by Write 1 to this bit field Value After Reset: 0x0
1	st_hdcp2_not_capable	R/W1C	HDCP 2.2 not capable rise interrupt status sticky bit. Clear by Write 1 to this bit field

Bits	Name	Attr	Description
			Value After Reset: 0x0
0	st_hdcp2_capable	R/W1C	HDCP 2.2 capable rise interrupt status sticky bit. Clear by Write 1 to this bit field Value After Reset: 0x0

hdcp22reg_mute

Description: HDCP 2.2 Interrupt Mute Vector

Size: 8 bits

Offset: 0x790e

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	mute_hdcp_decrypted_chg	R/W	Active high interrupt mute to HDCP 2.2 decrypted value change interrupt status Value After Reset: 0x1
4	mute_hdcp_authentication_fail	R/W	Active high interrupt mute to HDCP 2.2 authentication fail interrupt status Value After Reset: 0x1
3	mute_hdcp_authenticated	R/W	Active high interrupt mute to HDCP 2.2 authenticated interrupt status Value After Reset: 0x1
2	mute_hdcp_authentication_lost	R/W	Active high interrupt mute to HDCP 2.2 authentication lost interrupt status Value After Reset: 0x1
1	mute_hdcp2_not_capable	R/W	Active high interrupt mute to HDCP 2.2 not capable rise interrupt status Value After Reset: 0x1
0	mute_hdcp2_capable	R/W	Active high interrupt mute to HDCP 2.2 capable rise interrupt status Value After Reset: 0x1

CEC Registers

CEC Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
cec_ctrl	0x7d00	CEC Control Register This register handles the main control of the CEC initiator.
cec_mask	0x7d02	CEC Interrupt Mask Register This read/write register masks/unmasks the interrupt events. When the...
cec_addr_l	0x7d05	CEC Logical Address Register Low This register indicates the logical address(es) allocated to the...
cec_addr_h	0x7d06	CEC Logical Address Register High This register indicates the logical address(es) allocated to...
cec_tx_cnt	0x7d07	CEC TX Frame Size Register This register indicates the size of the frame in bytes (including header...
cec_rx_cnt	0x7d08	CEC RX Frame Size Register This register indicates the size of the frame in bytes (including header...

Register	Offset	Description
cec_tx_data[0:15]	0x7d10 + (i * 0x1)	CEC TX Data Register Array Address offset: i = 0 to 15 These registers (8 bits each) are the buffers...
cec_rx_data[0:15]	0x7d20 + (i * 0x1)	CEC RX Data Register Array Address offset: i =0 to 15 These registers (8 bit each) are the buffers...
cec_lock	0x7d30	CEC Buffer Lock Register
cec_wakeupctrl	0x7d31	CEC Wake-up Control Register After receiving a message in the CEC_RX_DATA1 (OPCODE) registers,...

cec_ctrl

Description: CEC Control Register

This register handles the main control of the CEC initiator.

Size: 8 bits

Offset: 0x7d00

Bits	Name	Attr	Description
7:5			Reserved for future use.
4	standby	R/W	1: CEC controller responds with a NACK to all messages and generates a wakeup status for opcode. It only responds with a NACK when the EOM is received. This means only the last block of a frame responds with NACK. The follower sends an ACK to the message when there is only one head block pointed to the follower, if the follower is in the standby mode. 0: CEC controller responds the ACK to all messages. Value After Reset: 0x0
3	bc_nack	R/W	1'b1: Set by software to NACK the received broadcast message. This bit holds until software resets. The broadcasts is answered with 1'b0, indicating the follower reject the message. 1'b0: Reset by software to ACK the received broadcast message. Value After Reset: 0x0
2:1	frame_typ	R/W	2'b00: Signal Free Time = 3-bit periods. Previous attempt to send frame is unsuccessful. 2'b01: Signal Free Time = 5-bit periods. New initiator wants to send a frame. 2'b10: Signal Free Time = 7-bit periods. Present initiator wants to send another frame immediately after its previous frame. (specification CEC 9.1). 2'b11: Illegal value. If software writes this value, hardware sets the value to the default 2'b01. Value After Reset: 0x1

Bits	Name	Attr	Description
0	send	R/W	1'b1: Set by software to trigger CEC sending a frame as an initiator. This bit keeps at 1'b1 while the transmission is going on. 1'b0: Reset to 1'b0 by hardware when the CEC transmission is done (no matter successful or failed). It can also work as an indicator checked by software to see whether the transmission is finished. Value After Reset: 0x0

cec_mask

Description: CEC Interrupt Mask Register

This read/write register masks/unmasks the interrupt events. When the bit is set to 1 (masked), the corresponding event does not trigger an interrupt signal at the system interface. When the bit is reset to 0, the interrupt event is unmasked.

Size: 8 bits

Offset: 0x7d02

Bits	Name	Attr	Description
7			Reserved for future use.
6	wakeup	R/W	Follower wake-up signal mask Value After Reset: 0x0
5	error_flow	R/W	An error is notified by a follower. Abnormal logic data bit error (for follower) Value After Reset: 0x0
4	error_initiator	R/W	An error is detected on a CEC line (for initiator only). Value After Reset: 0x0
3	arb_lost	R/W	The initiator losses the CEC line arbitration to a second initiator. (specification CEC 9) Value After Reset: 0x0
2	nack	R/W	A frame is not acknowledged in a directly addressed message. Or a frame is negatively acknowledged in a broadcast message (for initiator only) Value After Reset: 0x0
1	eom	R/W	EOM is detected so that the received data is ready in the receiver data buffer (for follower only) Value After Reset: 0x0
0	done	R/W	The current transmission is successful (for initiator only) Value After Reset: 0x0

cec_addr_l

Description: CEC Logical Address Register Low

This register indicates the logical address(es) allocated to the CEC device.

This register is written by software when the logical allocation is finished. Bit value 1 means the corresponding logical address is allocated to this device. Bit value 0 means the corresponding logical address is not allocated to this device.

Size: 8 bits

Offset: 0x7d05

Bits	Name	Attr	Description
7	cec_addr_l_7	R/W	Logical address 7 - Tuner 3 Value After Reset: 0x0
6	cec_addr_l_6	R/W	Logical address 6 - Tuner 2 Value After Reset: 0x0
5	cec_addr_l_5	R/W	Logical address 5 - Audio System Value After Reset: 0x0
4	cec_addr_l_4	R/W	Logical address 4 - Playback Device 1 Value After Reset: 0x0
3	cec_addr_l_3	R/W	Logical address 3 - Tuner 1 Value After Reset: 0x0
2	cec_addr_l_2	R/W	Logical address 2 - Recording Device 2 Value After Reset: 0x0
1	cec_addr_l_1	R/W	Logical address 1 - Recording Device 1 Value After Reset: 0x0
0	cec_addr_l_0	R/W	Logical address 0 - Device TV Value After Reset: 0x0

cec_addr_h

Description: CEC Logical Address Register High

This register indicates the logical address(es) allocated to the CEC device.

This register is written by software when the logical allocation is finished. Bit value 1 means the corresponding logical address is allocated to this device. Bit value 0 means the corresponding logical address is not allocated to this device.

Size: 8 bits

Offset: 0x7d06

Bits	Name	Attr	Description
7	cec_addr_h_7	R/W	Logical address 15 - Unregistered (as initiator address), Broadcast (as destination address) Value After Reset: 0x1
6	cec_addr_h_6	R/W	Logical address 14 - Free use Value After Reset: 0x0
5	cec_addr_h_5	R/W	Logical address 13 - Reserved Value After Reset: 0x0
4	cec_addr_h_4	R/W	Logical address 12 - Reserved Value After Reset: 0x0
3	cec_addr_h_3	R/W	Logical address 11 - Playback Device 3 Value After Reset: 0x0
2	cec_addr_h_2	R/W	Logical address 10 - Tuner 4 Value After Reset: 0x0
1	cec_addr_h_1	R/W	Logical address 9 - Playback Device 3 Value After Reset: 0x0
0	cec_addr_h_0	R/W	Logical address 8 - Playback Device 2 Value After Reset: 0x0

cec_tx_cnt

Description: CEC TX Frame Size Register

This register indicates the size of the frame in bytes (including header and data blocks), which are available in the transmitter data buffer.

Note: When the value is zero, the CEC controller ignores the send command triggered by software. When the transmission is done (no matter success or not), the current value is held until it is overwritten by software.

Size: 8 bits

Offset: 0x7d07

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:0	cec_tx_cnt	R/W	CEC Transmitter Counter register 5'd0: No data needs to be transmitted 5'd1: Frame size is 1 byte 5'd16: Frame size is 16 bytes Value After Reset: 0x0

cec_rx_cnt

Description: CEC RX Frame Size Register

This register indicates the size of the frame in bytes (including header and data blocks), which are available in the receiver data buffer.

Note: Only after the whole receiving process is finished successfully, the counter is refreshed to the value which indicates the total number of data bytes in the Receiver Data Register.

Size: 8 bits

Offset: 0x7d08

Bits	Name	Attr	Description
7:5			Reserved for future use.
4:0	cec_rx_cnt	R	CEC Receiver Counter register: 5'd0: No data received 5'd1: 1-byte data is received 5'd16: 16-byte data is received Value After Reset: 0x0

cec_tx_data[0:15]

Description: CEC TX Data Register Array Address offset: i = 0 to 15

These registers (8 bits each) are the buffers used for storing the data waiting for transmission (including header and data blocks).

Size: 8 bits

Offset: 0x7d10 + (i * 0x1)

Bits	Name	Attr	Description
7:0	databyte	R/W	Data byte[x], where x is 0 to 15 Value After Reset: 0x0

cec_rx_data[0:15]

Description: CEC RX Data Register Array Address offset: i =0 to 15

These registers (8 bit each) are the buffers used for storing the received data (including header and data blocks).

Size: 8 bits

Offset: 0x7d20 + (i * 0x1)

Bits	Name	Attr	Description
7:0	databyte	R	Data byte[x], where x is 0 to 15 Value After Reset: 0x0

cec_lock

Description: CEC Buffer Lock Register

Size: 8 bits

Offset: 0x7d30

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	locked_buffer	R/W	When a frame is received, this bit would be active. The

			CEC controller answers to all the messages with NACK until the CPU writes it to '0'. Value After Reset: 0x0

cec_wakeupctrl

Description: CEC Wake-up Control Register

After receiving a message in the CEC_RX_DATA1 (OPCODE) registers, the CEC engine verifies the message opcode[7:0] against one of the previously defined values to generate the wake-up status:

Wakeupstatus is 1 when:

received opcode is 0x04 and opcode0x04en is 1 or
received opcode is 0x0D and opcode0x0Den is 1 or
received opcode is 0x41 and opcode0x41en is 1 or
received opcode is 0x42 and opcode0x42en is 1 or
received opcode is 0x44 and opcode0x44en is 1 or
received opcode is 0x70 and opcode0x70en is 1 or
received opcode is 0x82 and opcode0x82en is 1 or
received opcode is 0x86 and opcode0x86en is 1

Wakeupstatus is 0 when none of the previous conditions are true.

This formula means that the wake-up status (on CEC_STAT[6] register) is only '1' if the opcode[7:0] received is equal to one of the defined values and the corresponding enable bit of that defined value is set to '1'.

Size: 8 bits

Offset: 0x7d31

Bits	Name	Attr	Description
7	opcode0x86en	R/W	OPCODE 0x86 wake up enable Value After Reset: 0x1
6	opcode0x82en	R/W	OPCODE 0x82 wake up enable Value After Reset: 0x1
5	opcode0x70en	R/W	OPCODE 0x70 wake up enable Value After Reset: 0x1
4	opcode0x44en	R/W	OPCODE 0x44 wake up enable Value After Reset: 0x1
3	opcode0x42en	R/W	OPCODE 0x42 wake up enable Value After Reset: 0x1
2	opcode0x41en	R/W	OPCODE 0x41 wake up enable Value After Reset: 0x1
1	opcode0x0Den	R/W	OPCODE 0x0D wake up enable Value After Reset: 0x1
0	opcode0x04en	R/W	OPCODE 0x04 wake up enable Value After Reset: 0x1

EDDC Registers

E-DDC Registers. Follow the link for the register to see a detailed description of the register.

Register	Offset	Description
i2cm_slave	0x7e00	I2C DDC Slave address Configuration Register

Register	Offset	Description
i2cm_address	0x7e01	I2C DDC Address Configuration Register
i2cm_datao	0x7e02	I2C DDC Data Write Register
i2cm_datai	0x7e03	I2C DDC Data read Register
i2cm_operation	0x7e04	I2C DDC RD/RD_EXT/WR Operation Register Read and write operation request. This register can only...
i2cm_int	0x7e05	I2C DDC Done Interrupt Register This register configures the I2C master interrupts.
i2cm_ctlint	0x7e06	I2C DDC error Interrupt Register This register configures the I2C master arbitration lost and not...
i2cm_div	0x7e07	I2C DDC Speed Control Register This register configures the division relation between master and...
i2cm_segaddr	0x7e08	I2C DDC Segment Address Configuration Register This register configures the segment address for...
i2cm_softrstz	0x7e09	I2C DDC Software Reset Control Register This register resets the I2C master.
i2cm_segptr	0x7e0a	I2C DDC Segment Pointer Register This register configures the segment pointer for extended RD/WR...
i2cm_ss_scl_hcnt_1_addr	0x7e0b	I2C DDC Slow Speed SCL High Level Control Register 1
i2cm_ss_scl_hcnt_0_addr	0x7e0c	I2C DDC Slow Speed SCL High Level Control Register 0
i2cm_ss_scl_lcnt_1_addr	0x7e0d	I2C DDC Slow Speed SCL Low Level Control Register 1
i2cm_ss_scl_lcnt_0_addr	0x7e0e	I2C DDC Slow Speed SCL Low Level Control Register 0
i2cm_fs_scl_hcnt_1_addr	0x7e0f	I2C DDC Fast Speed SCL High Level Control Register 1
i2cm_fs_scl_hcnt_0_addr	0x7e10	I2C DDC Fast Speed SCL High Level Control Register 0
i2cm_fs_scl_lcnt_1_addr	0x7e11	I2C DDC Fast Speed SCL Low Level Control Register 1
i2cm_fs_scl_lcnt_0_addr	0x7e12	I2C DDC Fast Speed SCL Low Level Control Register 0
i2cm_sda_hold	0x7e13	I2C DDC SDA Hold Register
i2cm_scdc_read_update	0x7e14	SCDC Control Register This register configures the SCDC update status read through the I2C master...
i2cm_read_buff0	0x7e20	I2C Master Sequential Read Buffer Register 0
i2cm_read_buff1	0x7e21	I2C Master Sequential Read Buffer Register 1

Registers for Address Block: EDDC (Continued)

Register	Offset	Description
i2cm_read_buff2	0x7e22	I2C Master Sequential Read Buffer Register 2
i2cm_read_buff3	0x7e23	I2C Master Sequential Read Buffer Register 3
i2cm_read_buff4	0x7e24	I2C Master Sequential Read Buffer Register 4
i2cm_read_buff5	0x7e25	I2C Master Sequential Read Buffer Register 5
i2cm_read_buff6	0x7e26	I2C Master Sequential Read Buffer Register 6
i2cm_read_buff7	0x7e27	I2C Master Sequential Read Buffer Register 7
i2cm_scdc_update0	0x7e30	I2C SCDC Read Update Register 0

Register	Offset	Description
i2cm_scdc_update1	0x7e31	I2C SCDC Read Update Register 1

i2cm_slave

Description: I2C DDC Slave address Configuration Register

Size: 8 bits

Offset: 0x7e00

Bits	Name	Attr	Description
7			Reserved for future use.
6:0	slaveaddr	R/W	Slave address to be sent during read and write normal operations. Value After Reset: 0x0

i2cm_address

Description: I2C DDC Address Configuration Register

Size: 8 bits

Offset: 0x7e01

Bits	Name	Attr	Description
7:0	address	R/W	Register address for read and write operations Value After Reset: 0x0

i2cm_datao

Description: I2C DDC Data Write Register

Size: 8 bits

Offset: 0x7e02

Bits	Name	Attr	Description
7:0	datao	R/W	Data to be written on register pointed by address[7:0]. Value After Reset: 0x0

i2cm_datai

Description: I2C DDC Data read Register

Size: 8 bits

Offset: 0x7e03

Bits	Name	Attr	Description
7:0	datai	R	Data read from register pointed by address[7:0]. Value After Reset: 0x0

i2cm_operation

Description: I2C DDC RD/RD_EXT/WR Operation Register

Read and write operation request. This register can only be written; reading this register always results in 00h. Writing 1'b1 simultaneously to rd, rd_ext and wr requests is considered as a read (rd) request.

Size: 8 bits

Offset: 0x7e04

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	busclear	W	Bus clear operation request. Value After Reset: 0x0
4	wr	W	Single byte write operation request. Value After Reset: 0x0
3	rd8_ext	W	Extended sequential read operation request. Eight bytes are read starting at the address defined in register field i2cm_address.address and stored in registers i2cm_read_buffx. Value After Reset: 0x0
2	rd8	W	Sequential read operation request. Eight bytes are read starting at the address defined in the i2cm_address.address register field and stored in the i2cm_read_buffx registers. Value After Reset: 0x0
1	rd_ext	W	After writing 1'b1 to rd_ext bit a extended data read operation is started (E-DDC read operation). Value After Reset: 0x0
0	rd	W	Single byte read operation request Value After Reset: 0x0

i2cm_int

Description: I2C DDC Done Interrupt Register This register configures the I2C master interrupts.

Size: 8 bits

Offset: 0x7e05

Bits	Name	Attr	Description
7			Reserved for future use.
6	read_req_mask	R/W	Read request interruption mask signal. Value After Reset: 0x1
5:3			Reserved for future use.
2	done_mask	R/W	Done interrupt mask signal. Value After Reset: 0x0
1:0			Reserved for future use.

i2cm_ctlint

Description: I2C DDC error Interrupt Register

This register configures the I2C master arbitration lost and not acknowledge error interrupts.

Size: 8 bits

Offset: 0x7e06

Bits	Name	Attr	Description
7			Reserved for future use.
6	nack_mask	R/W	Not acknowledge error interrupt mask signal. Value After Reset: 0x0
5:3			Reserved for future use.
2	arbitration_mask	R/W	Arbitration error interrupt mask signal. Value After Reset: 0x0

Bits	Name	Attr	Description
1:0			Reserved for future use.

i2cm_div

Description: I2C DDC Speed Control Register

This register configures the division relation between master and scl clock.

Size: 8 bits

Offset: 0x7e07

Bits	Name	Attr	Description
7:4			Reserved for future use.
3	fast_std_mode	R/W	Sets the I2C Master to work in Fast Mode or Standard Mode: 1: Fast Mode 0: Standard Mode Value After Reset: 0x1
2:0	spare	R/W	Reserved as "spare" bit with no associated functionality. Value After Reset: 0x3

i2cm_segaddr

Description: I2C DDC Segment Address Configuration Register

This register configures the segment address for extended R/W destination and is used for EDID reading operations, particularly for the Extended Data Read Operation for Enhanced DDC.

Size: 8 bits

Offset: 0x7e08

Bits	Name	Attr	Description
7			Reserved for future use.
6:0	seg_addr	R/W	I2C DDC Segment Address Configuration Register Value After Reset: 0x0

i2cm_softrstz

Description: I2C DDC Software Reset Control Register This register resets the I2C master.

Size: 8 bits

Offset: 0x7e09

Bits	Name	Attr	Description
7:1			Reserved for future use.
0	i2c_softrstz	R/W	I2C Master Software Reset. Active by writing a zero and auto cleared to one in the following cycle. Value After Reset: 0x1

i2cm_segptr

Description: I2C DDC Segment Pointer Register

This register configures the segment pointer for extended RD/WR request.

Size: 8 bits

Offset: 0x7e0a

Bits	Name	Attr	Description
7:0	segptr	R/W	I2C DDC Segment Pointer Register

			Value After Reset: 0x0

i2cm_ss_scl_hcnt_1_addr

Description: I2C DDC Slow Speed SCL High Level Control Register 1

Size: 8 bits

Offset: 0x7e0b

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_hcnt1	R/W	I2C DDC Slow Speed SCL High Level Control Register 1 Value After Reset: 0x0

i2cm_ss_scl_hcnt_0_addr

Description: I2C DDC Slow Speed SCL High Level Control Register 0

Size: 8 bits

Offset: 0x7e0c

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_hcnt0	R/W	I2C DDC Slow Speed SCL High Level Control Register 0 Value After Reset: 0x6c

i2cm_ss_scl_lcnt_1_addr

Description: I2C DDC Slow Speed SCL Low Level Control Register 1

Size: 8 bits

Offset: 0x7e0d

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_lcnt1	R/W	I2C DDC Slow Speed SCL Low Level Control Register 1 Value After Reset: 0x0

i2cm_ss_scl_lcnt_0_addr

Description: I2C DDC Slow Speed SCL Low Level Control Register 0

Size: 8 bits

Offset: 0x7e0e

Bits	Name	Attr	Description
7:0	i2cmp_ss_scl_lcnt0	R/W	I2C DDC Slow Speed SCL Low Level Control Register 0 Value After Reset: 0x7f

i2cm_fs_scl_hcnt_1_addr

Description: I2C DDC Fast Speed SCL High Level Control Register 1

Size: 8 bits

Offset: 0x7e0f

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_hcnt1	R/W	I2C DDC Fast Speed SCL High Level Control Register 1 Value After Reset: 0x0

i2cm_fs_scl_hcnt_0_addr

Description: I2C DDC Fast Speed SCL High Level Control Register 0

Size: 8 bits

Offset: 0x7e10

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_hcnt0	R/W	I2C DDC Fast Speed SCL High Level Control Register 0 Value After Reset: 0x11

i2cm_fs_scl_lcnt_1_addr

Description: I2C DDC Fast Speed SCL Low Level Control Register 1

Size: 8 bits

Offset: 0x7e11

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_lcnt1	R/W	I2C DDC Fast Speed SCL Low Level Control Register 1 Value After Reset: 0x0

i2cm_fs_scl_lcnt_0_addr

Description: I2C DDC Fast Speed SCL Low Level Control Register 0

Size: 8 bits

Offset: 0x7e12

Bits	Name	Attr	Description
7:0	i2cmp_fs_scl_lcnt0	R/W	I2C DDC Fast Speed SCL Low Level Control Register 0 Value After Reset: 0x24

i2cm_sda_hold

Description: I2C DDC SDA Hold Register

Size: 8 bits

Offset: 0x7e13

Bits	Name	Attr	Description
7:0	osda_hold	R/W	Defines the number of SFR clock cycles to meet tHD;DAT (300 ns) osda_hold = round_to_high_integer (300 ns / (1 / isfrclk_frequency)) Value After Reset: 0x9

i2cm_scdc_read_update

Description: SCDC Control Register

This register configures the SCDC update status read through the I2C master interface.

Size: 8 bits

Offset: 0x7e14

Bits	Name	Attr	Description
7:6			Reserved for future use.
5	updtrd_vsyncpoll_en	R/W	Update read polling enabled. When active (1'b1), an SCDC Update Read is performed at the fall of the active edge of the vertical sync pulse. Value After Reset: 0x0
4	read_request_en	R/W	Read request enabled. When active (1'b1) an SCDC Update Read shall be performed whenever a SCDC read request is detected.

			Value After Reset: 0x0
3:1			Reserved for future use.
0	read_update	W	When set to 1'b1, a SCDC Update Read is performed and the read data loaded into registers i2cm_scdc_update0 and i2cm_scdc_update1. Value After Reset: 0x0

i2cm_read_buff0
Description: I2C Master Sequential Read Buffer Register 0
Size: 8 bits
Offset: 0x7e20

Bits	Name	Attr	Description
7:0	i2cm_read_buff0	R	Byte 0 of a I2C read buffer sequential read (from address i2cm_address) Value After Reset: 0x0

i2cm_read_buff1
Description: I2C Master Sequential Read Buffer Register 1
Size: 8 bits
Offset: 0x7e21

Bits	Name	Attr	Description
7:0	i2cm_read_buff1	R	Byte 1 of a I2C read buffer sequential read (from address i2cm_address+1) Value After Reset: 0x0

i2cm_read_buff2
Description: I2C Master Sequential Read Buffer Register 2
Size: 8 bits
Offset: 0x7e22

Bits	Name	Attr	Description
7:0	i2cm_read_buff2	R	Byte 2 of a I2C read buffer sequential read (from address i2cm_address+2) Value After Reset: 0x0

i2cm_read_buff3
Description: I2C Master Sequential Read Buffer Register 3
Size: 8 bits
Offset: 0x7e23

Bits	Name	Attr	Description
7:0	i2cm_read_buff3	R	Byte 3 of a I2C read buffer sequential read (from address i2cm_address+3) Value After Reset: 0x0

i2cm_read_buff4
Description: I2C Master Sequential Read Buffer Register 4
Size: 8 bits
Offset: 0x7e24

Bits	Name	Attr	Description
7:0	i2cm_read_buff4	R	Byte 4 of a I2C read buffer sequential read (from address i2cm_address+4) Value After Reset: 0x0

i2cm_read_buff5

Description: I2C Master Sequential Read Buffer Register 5

Size: 8 bits

Offset: 0x7e25

Bits	Name	Attr	Description
7:0	i2cm_read_buff5	R	Byte 5 of a I2C read buffer sequential read (from address i2cm_address+5) Value After Reset: 0x0

i2cm_read_buff6

Description: I2C Master Sequential Read Buffer Register 6

Size: 8 bits

Offset: 0x7e26

Bits	Name	Attr	Description
7:0	i2cm_read_buff6	R	Byte 6 of a I2C read buffer sequential read (from address i2cm_address+6) Value After Reset: 0x0

i2cm_read_buff7

Description: I2C Master Sequential Read Buffer Register 7

Size: 8 bits

Offset: 0x7e27

Bits	Name	Attr	Description
7:0	i2cm_read_buff7	R	Byte 7 of a I2C read buffer sequential read (from address i2cm_address+7) Value After Reset: 0x0

i2cm_scdc_update0

Description: I2C SCDC Read Update Register 0

Size: 8 bits

Offset: 0x7e30

Bits	Name	Attr	Description
7:0	i2cm_scdc_update0	R	Byte 0 of a SCDC I2C update sequential read Value After Reset: 0x0

i2cm_scdc_update1

Description: I2C SCDC Read Update Register 1

Size: 8 bits

Offset: 0x7e31

Bits	Name	Attr	Description
7:0	i2cm_scdc_update1	R	Byte 1 of a SCDC I2C update sequential read

Bits	Name	Attr	Description
			Value After Reset: 0x0

5.6 Application Notes

This chapter describes how to bring up HDMI transmitter in your system. As shown few examples below, these introduce the basically HDMI transmitter application, likes, the Hot Plug Detect, EDID read back, multiple audio format input and different video resolution displaying.

5.6.1 Initial Operation

The default HDMI transmitter is configured to 24bit RGB 1080P resolution video with 8 channel 48K sample I2S format audio input. It is easily for customer to turn on HDMI transmitter without doing more complex operation. Just do the step, reset the HDMI TX.

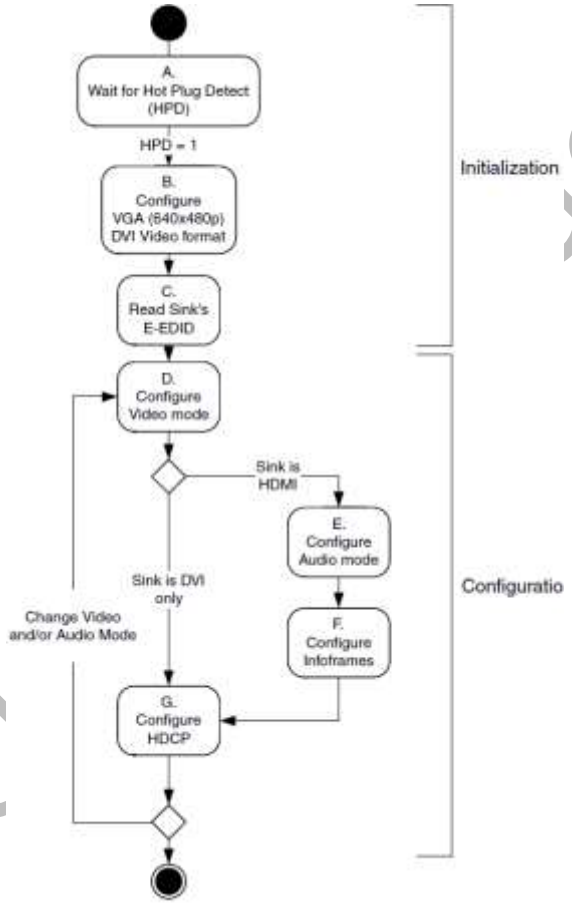


Fig. 5-5 HDMI programming sequence

5.6.2 Step A: Hot Plug Detection

The Hot Plug Detect information notifies the HDMI TX controller when the cable is plugged into a Sink (Receiver) device and when the device is ready to receive video content.
The Hot-Plug Detection indication is a +5V signal on the HDMI cable. The HDMI TX PHY performs level-shifting to a low-level digital signal.
For this step, registers are used on the HDMI TX PHY.
To check the Hot Plug Detect status, the following steps have to be followed:
1. Power-on the HDMI TX PHY HPD Detector. Write 1'b1 in the phy_conf0.enhpdrxsense bit field register.
2. Check the HPD status bit.

- Read the phy_stat0.HPD bit field register.
  - If HPD = 0, the Hot Plug signal is low (no Sink (Receiver) detected).

- If HPD = 1, the Hot Plug signal is high (Sink (Receiver) detected).

Note: Some receivers may turn off the Hot-Plug Detect indication when the HDMI input is not selected, even when the cable is plugged in.

You can control this function by using the interrupt signal and proper registers from the HDMI transmitter with few operations.

5.6.3 Step B: Configure Video Mode

Once a receiver is detected, it is necessary to start sending video content in DVI mode. Because some receivers need to receive a video signal (TMDS clock more precisely), it is important to clock all the digital logic and be able to reply on an E-EDID read access (Step C). Following flow is a configuration example of VGA (640x480p).

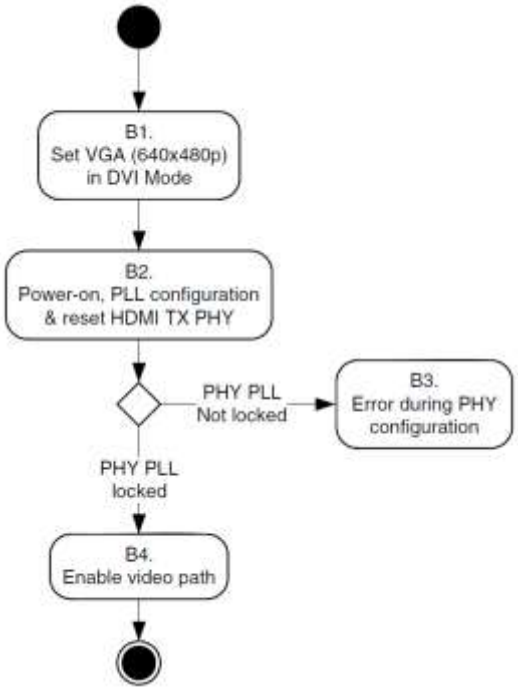


Fig. 5-6 Configure Video Mode

The VGA (640x480p) mode is mandatory for all receivers and transmitters, because it requires the lowest data rate possible on the HDMI specification.

1. Set VGA (640x480p) in DVI mode.

- To select the Video Mapping input RGB444, 8-bits per color components (24-bits RGB), write "1" in the tx_invid0.video_mapping register.
- To configure VGA timing information (CEA mode 1):
  - Write "0" in the fc_invidconf.vsync_in_polarity register.
  - Write "0" in the fc_invidconf.hsync_in_polarity register.
  - Write "1" in the fc_invidconf.de_in_polarity register.
  - Write "0" in the fc_invidconf.r_v_blank_in_osc register.
  - Write "0" in the fc_invidconf.in_I_P register.
  - 640 H active pixels: 0x280
    - Write "2" in the fc_inhactiv1.H_in_activ register.
    - Write "0x80" in the fc_inhactiv0.H_in_activ register.
  - 480 V active pixels: 0x1E0
    - Write "1" in the fc_invactiv1.V_in_activ register.
    - Write "0xE0" in the fc_invactiv0.V_in_activ register.
  - 160 H blanking pixels: 0xA0
    Write "0xA0" in thefc_inhblank0.H_in_blank.
  - 45 V blanking pixels: 0x2D

Write "0x2D" in the fc_invblank.V_in_blank register.

- 16 Sync offset: 0x10
  Write "0x10" in the fc_hsyncindelay0.H_in_delay register.
- 10 Vsync offset: 0x0A
  Write "0x0A" in the fc_vsyncindelay0.V_in_delay register.
- 96 HSync pulse width: 0x60
  Write "0x60" in the fc_hsyncinwidth0.H_in_width register.
- 2 VSync pulse width: 0x02
  Write "0x02" in the fc_vsyncinwidth0.V_in_width register.

- To select the HDMI mode:
  Write "1" in the fc_invidconf.DVI_modez register.

2. Power-on, configure the PLL, and reset the HDMI TX PHY.

The HDMI TX PHY has an internal PLL that generates TMDS clock from Pixel clock input. The power-on sequence and PLL configuration is part of the HDMI TX PHY documentation. To configure HDMI TX PHY, you are generally required to:

a. Place the PHY in reset by writing 0x01 in the mc_phyrstz register.

b. Write the desired color depth and the pixel repetition in the vp_pr_cd register.

c. After a PHY-dependent time, it is required to lift the reset by writing 0x00 to the mc_phyrstz register.

d. Clear the pddq, txpwron configuration bits, define the data enable polarity and clear the interface control selection by writing these values in the phy_cfg0 register.

e. Set the PHY slave address by writing 0x69 in the phy_i2cm_slave register.

f. You are required to look up the configuration for your intended video mode and write those values on the PHY I2C interface. The baseline flow to write to the PHY through the I2C interface is:

- Write register address in the phy_i2cm_address register.
- Write data in the phy_i2cm_datao_1 (MSB, [15:8]) and phy_i2cm_datao_0 (LSB, [7:0]) registers.
- Initialize the write operation by writing 8'h10 in the phy_i2cm_operation register.
- Wait for a done interruption from the I2C master.

g. After all of the required PHY I2C registers have been configured, you now need to set the txpwron bit in the PHY_CONF0 register, leaving the remaining bits in the state defined previously.

3. Handle errors during PHY configuration.

If the PHY is not correctly configured, the PLL does not lock and no activity can happen on the HDMI output.

4. Enable the video path.

a. Set default parameters in the HDMI TX Controller, Control period duration: 12: 0x0C.
Write "0x0C" in the fc_ctrldur.ctrlperiodduration bit field register.

b. Set default parameters in the HDMI TX Controller, Extended Control period duration: 32: 0x20.
Write "0x20" in the fc_exctrldur.exctrlperiodduration bit field register.

c. Set default parameters in the HDMI TX Controller, Max spacing between extended Control period duration: 1: 0x1.
Write "0x01" in the fc_exctrlspac.exctrlperiodspacing bit field register.

d. Set default parameters in the HDMI TX Controller, Preamble filters to fill TMDS data channels.

- Write "0x0B" in the fc_ch0pream.ch0_preamble_filter bit field register.

- Write “0x16” in the fc_ch1pream.ch1_preamble_filter bit field register.
- Write “0x21” in the fc_ch2pream.ch2_preamble_filter bit field register.

e. Enable pixel clock data path:

Write “0” in the mc_clkdis.pixelclk_disable bit field register.

f. Enable TMDS clock data path:

i. Write “0” in the mc_clkdis.tmdsclk_disable bit field register.

ii. Re-Write the VSync pulse width in the fc_vsyncinwidth0.V_in_width bit field register.

After these steps, it is expected to observe a video picture in VGA mode on the receiver side.

5.6.4 Step C: Reading E-EDID

The E-EDID is a memory present on the receiver side. It contains the video and audio capabilities of the receiver. It is important to read this information and parse it in order to configure the transmitter system in accordance to what the receiver supports. The E-EDID is read using the E-DDC channel present on the HDMI cable.

The E-DDC channel protocol is based on I2C, with segment pointer addressing extension (such as, Extended Read). The hdmi tx controller has an I2C Master controller for this purpose.

1. To perform a “normal” read operation

- Set I2C slave address.
  Write i2cm_slave.slaveaddr[6:0] bit field register.
- Set I2C register address.
  Write i2cm_address.address[7:0] bit field register.
- Activate Read operation.
  Write “1” in the i2cm_operation.rd bit field register.
- Wait for interruption.
  Wait for the ii2cmasterdone interrupt in the ih_i2cm_stat0 register.
- Read data result.
  Read data in the i2cm_datai.datai[7:0] bit field.

2. To perform an E-DDC extended read operation

- Set I2C slave address.
  Write i2cm_slave.slaveaddr[6:0] bit field register.
- Set I2C segment address.
  Write the i2cm_segaddr.seg_addr bit field register.
- Set I2C segment pointer.
  Write i2cm_segptr.segptr bit field register.
- Activate Read operation.
  Write “1”in the i2cm_operation.rd_ext bit field register.
- Wait for interruption.
  Wait for the ii2cmasterdone interrupt in the ih_i2cm_stat0 register.
- Read data result.
  Read data in the i2cm_datai.datai[7:0] bit field register.

5.6.5 Step D: Configure Video Mode

The video mode selected has to match what is supported by the source of the video (Transmitter side) and the sink of the video (receiver side). The video capabilities of the receiver are extracted from the E-EDID information (refer to step C).

To configure any video mode, a similar procedure in step B has to be followed.

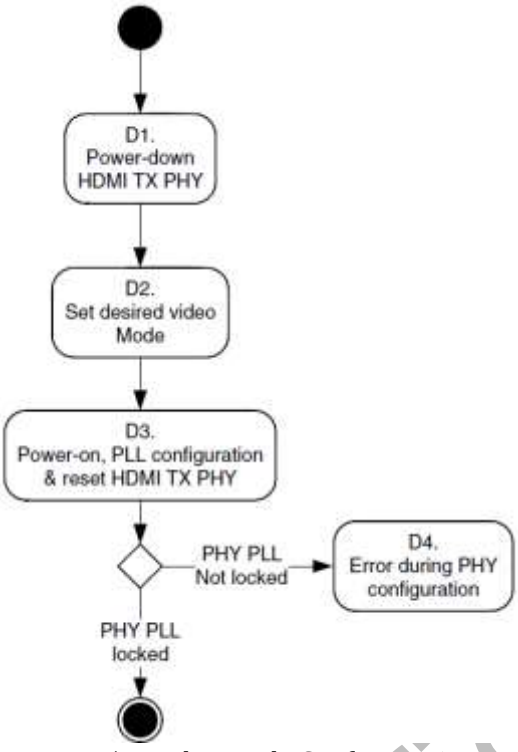


Fig. 5-7 Video Mode Configuration

Before any change is made on the video modes, is it recommended you power-down the HDMI TX PHY to avoid any unexpected behavior on the receiver side.

1. Power-down HDMI TX PHY Detector

- Place the PHY in reset by writing 0x01 in the mc_phyrstz register.
- Then proceed to set SVSRET_MODEZ to 1'b0, PDDQ to 1'b1 and TX_PWRON to 1b0, by writing 2'h12 in the phy_conf0 register.

2. Set desired video mode.

- To select the Video Mapping input mode (RGB444, YCC444, YCC422). Write the video code in the tx_invid0.video_mapping bit field register.
- Set video timing information configuration:
  - Write the fc_invidconf.vsync_in_polarity register.
  - Write the fc_invidconf.hsync_in_polarity register.
  - Write the fc_invidconf.de_in_polarity register.
  - Write the fc_invidconf.r_v_blank_in_osc register.
  - Write the fc_invidconf.in_I_P register.
  - H active pixels
    - Write the fc_inhactiv1.H_in_activ register.
    - Write the fc_inhactiv0.H_in_activ register.
  - V active pixels

    If the desired video mode is 3D, the fc_invact_2d_1, fc_invact_2d_0, and all registers in the following table must be written.

Table 5-5 HDMI 3D structure table

3D STRUCTURE	fc_actspc_hdlr_cfg	
	fc_actspc_hdlr_tgl	fc_actspc_hdlr_en
Frame Packing for interlaced format	1	1
Frame Packing for progressive format	0	1
Field alternative	0	0
Line Alternative	0	0
Side-by-Side(FULL)	0	0
L+depth	0	1

L+depth+graphics+graphics-depth	0	1
Top-and-Bottom	0	0
Reserved for Future Use	0	0
Side-by-Side(FULL)	0	0
Reserved for Future Use	0	0
Not in use	0	0

- Write the fc_invact_2D_1 register with the 2D Vertical video active[11:8].
- Write the fc_invact_2D_0 register with the 2D Vertical video active[7:0]

The following registers must always be written with the full V active of the desired video
regardless of the type of the video mode.

- Write the fc_invactiv1.V_in_activ register.
- Write the fc_invactiv0.V_in_activ register.

■ H blanking pixels

Write the fc_inhblank0.H_in_blank register.

■ V blanking pixels

Write the fc_invblank.V_in_blank register.

■ HSync offset

Write the fc_hsyncindelay0.H_in_delay register.

■ VSync offset

Write the fc_vsyncindelay0.V_in_delay register.

■ HSync pulse width

Write the fc_hsyncinwidth0.H_in_width register.

■ VSync pulse width

Write the fc_vsyncinwidth0.V_in_width register.

- Select DVI or HDMI mode:
  Write "1" for HDMI in the fc_invidconf.DVI_modez bit field register.

3. Power-on, configure the PLL, and reset the HDMI TX PHY (same as Step B2).

The HDMI TX PHY has an internal PLL that generates TMDS clock from Pixel clock input. All the power-on sequence and PLL configuration is part of the HDMI TX PHY documentation. The HDMI TX Controller provides all registers necessary for the PHY sequence implementation, which are presented in the "HDMI PHY Registers".

- Place the PHY in reset by writing 0x01 in the mc_phyrstz register.
- Write in the desired color depth and the pixel repetition in the vp_pr_cd register.
- After a PHY-dependent time, it is required to lift the reset by writing 0x00 to the mc_phyrstz register.
- Clear the pddq, txpwron configuration bits, define the data enable polarity and clear the interface control selection by writing these values in the phy_conf0 register.
- Set the PHY slave address by writing 0x69 to the phy_i2cm_slave register.
- Look up the configuration for your intended video mode and write those values on the PHY I2C interface. The baseline flow to write in the PHY through the I2C interface is
  ■ Write the register address in the phy_i2cm_address register.
  ■ Write data into phy_i2cm_datao_1 (MSB, [7:0]) and phy_i2cm_datao_0 (LSB, [7:0]) registers.
  ■ Initialize the write operation by writing 8'h10 in the phy_i2cm_operation register.
  ■ Wait for a done interrupt from the I2C master.
- After all of the required PHY I2C registers have been configured, you now need to set the phy_conf0.txpwron register, leaving the remaining bits in the state defined previously.

5.6.6 Step E: Audio input configuration

HDMI transmitter audio support either SPDIF or four channel I2S input. SPDIF input supports audio sampling rates from 32 to 192 KHz. The I2S input supports from 2-channel to 8-channel audio up to 192 KHz. The default audio format is I2S input with 8 channels. The audio sample rate is 48K.

- Configure Audio Input Format with I2S Steps:
  - ■ Select I2S input.
  Write "1" in the aud_conf0.i2s_select bit field register.
  - ■ Enable I2S inputs:
  Write "1" in the aud_conf0.i2s_in_en[3:0] bit field register.
  - ■ Set I2S Mode [Standard | Right-justified | Left-justified | Burst1 | Burst2]:
  Write the aud_conf1.i2s_mode[2:0] bit field register.
  - ■ Set I2S data width [16 bits up to 24 bits]:
  Write the aud_conf1.i2s_width[4:0] bit field.
- Configure Audio Input Format with SPDIF Steps:
  - ■ Select SPDIF input.
  Write "0" in the aud_conf0.i2s_select bit field register.
  - ■ Set S/PDIF Linear-PCM or Non-Linear PCM audio samples:
  Write the aud_spdif1.setnlpcm bit field register.
  - ■ Set SPDIF data width [16 bits up to 24 bits]:
  Write the aud_spdif1.spdif_width[4:0] bit field.
- Configure Audio Parameters Steps:
  - ■ Set Audio input frequency clock FS ratio factor [128 Fs | 256 Fs |512 Fs]:
  Write the aud_inputclkfs.lfsfactor bit field register.
  - ■ Set Audio fixed N factor for Audio Clock Regeneration. This factor depends on the audio sampling rate and video mode.
  Write the aud_n1.audN, aud_n2.audN, and aud_n3.audN bit field registers.
  - ■ Set Audio CTS factor for Audio Clock Regeneration. This factor can be generated automatically or manually.
  For Automatic CTS generation
  Write "0" on the bit field "CTS_manual", Register 0x3205: AUD_CTS3
  For Manual CTS setting
  Write "1" in the aud_cts3.CTS_manual register bit field.
  Write the aud_cts1.audCTS, aud_cts2.audCTS, aud_cts3.audCTS bit field registers.
  - ■ Enable Audio sampler block:
  Write "0" in the mc_clkdis.audclk_disable bit field register.

5.6.7 Step F: Configure Infoframes

This step is only relevant when the HDMI TX controller is configured in HDMI mode (refer to "Step D: Configure Video Mode"). In DVI mode, no Infoframes are transmitted. The configuration of the Infoframes is essential to correctly inform the HDMI Receiver about the content of the video and audio format been transmitted. All the detailed information is provided in the HDMI 1.4b Specification.

5.6.8 Step G: Configure HDCP 1.4

Following flow is an initialization of HDCP 1.4. To detect if the HDMI TX Controller is HDCP capable:

1. Read the product_id1.product_id1 bit field register:

- If product_id1 = 01, HDCP is not present.
- If product_id1 = C1, HDCP is present.

2. Select HDMI mode:

- Write "1" for HDMI in the a_hdcpcfg0.hdmidvi bit field register.

3. Set the Data enable, Hsync and VSync polarity:

Write the dataenpol, vsyncpol, and hsyncpol bit fields of the a_vidpolcfg register.

4. Set encryption on:

- Write "64 (0x40)" for "OesswindowSize" in the a_oesswcfg register.
- Write "0" for "no HDCP bypass" a_hdcpcfg0.BypassEncryption bit field register.
- Write "0" for "HDCP enable" in the a_hdcpcfg1.encryptiondisable bit field register.

5. Reset HDCP engine:

- Write "0" in the a_hdcpcfg1.swreset bit field register.

6. Configure Device Private Keys, which is illustrated in the flow chart Figure 3-6

7. Enable the encryption:

- Write "1" in the a_hdcpcfg0.rxdetect bit field register.

8. When connected to a repeater the SHA-1 calculation is needed to be done by software, which are indicated in the following steps

a. Wait for an interrupt to be triggered (a_apiintstat.KSVSha1calcint).

b. Request access to KSV memory through setting a_ksvmemctrl.KSVMEMrequest to 1'b1 and pool a_ksvmemctrl.KSVMEMaccess until this value is 1'b1 (access granted).

c. Read VH', M0, Bstatus, and the KSV FIFO.

The data is stored in the revocation memory.

d. Calculate the SHA-1 checksum (VH) over M0, Bstatus, and the KSV FIFO.

e. If the calculated VH equals the VH`, set a_ksvmemctrl.SHA1fail to 0 and set a_ksvmemctrl.KSVCTRLupd to 1. If the calculated VH is different from VH` then set a_ksvmemctrl.SHA1fail to 1 and set a_ksvmemctrl.KSVCTRLupd to 1, forcing the controller to re-authenticate from the beginning.

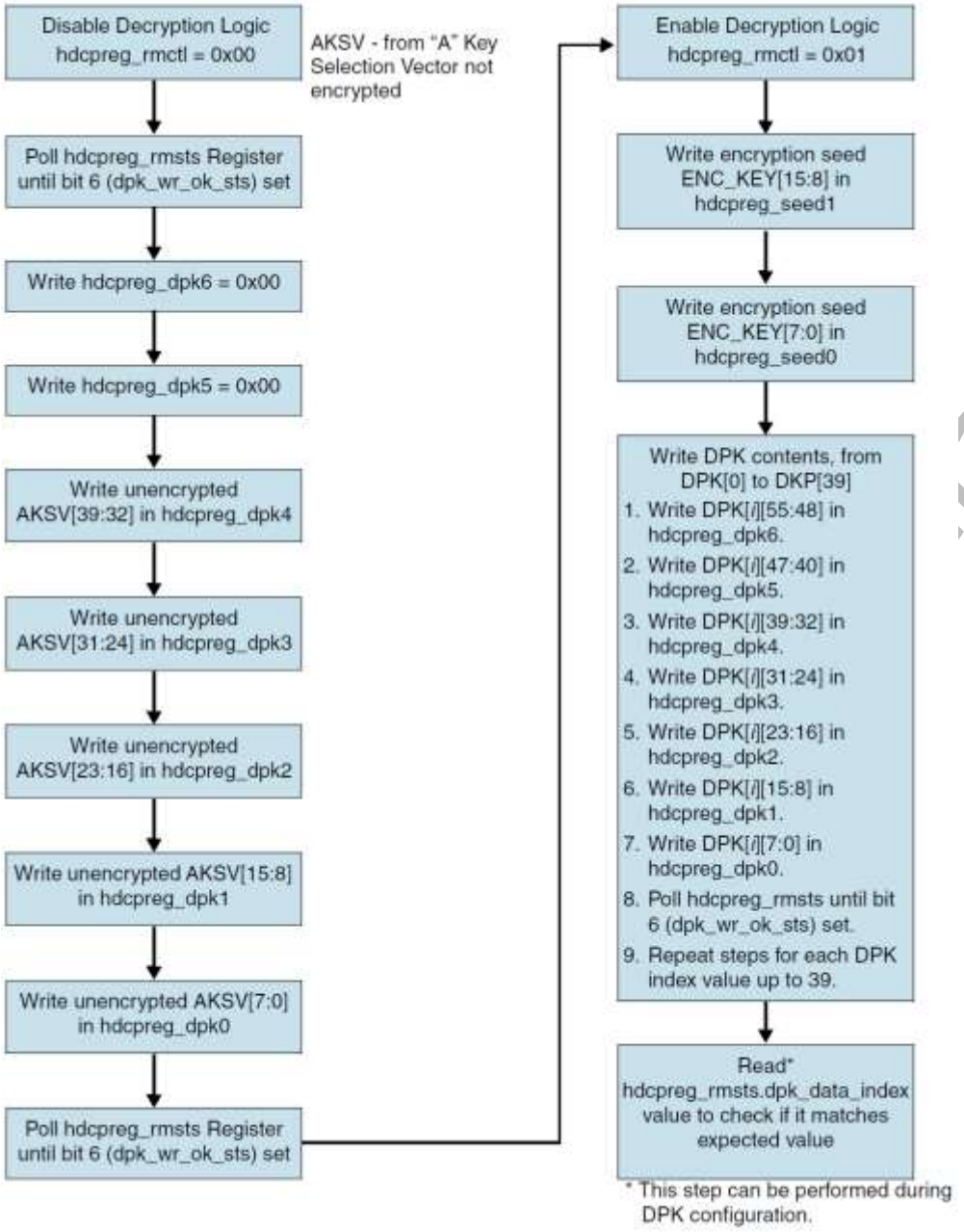


Fig. 5-8 Configuring Device Protection Key Process

5.6.9 CEC OPERATION

The CEC line is used for high-level user control of HDMI-connected devices.The HDMI TX contain CEC TX operations and CEC RX operations.

You can control this function by using the interrupt signal and proper registers from the HDMI transmitter with few operations. The register offset is from 0x7D00.

- Configure The CEC Step:
  - ■ Write the CEC logical address to cec_addr_l,cec_addr_h register
  - ■ Write the size of the frame in bytes which are available in the transmitter data buffer to cec_tx_cnt register
  - ■ Write the desired CEC data(including header and data blocks) to cec_tx_data0 to cec_tx_data15
  - ■ Write 1 to cec_ctrl.send register, to start the cec transmit.

5.6.10 HDMI PHY MPLL configuration

Table 5-6 HDMI PHY MPLL Generic Configuration Settings

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl		fbdiv1_cntrl		ref_cntrl		n_cntrl		prop_cntrl		int_cntrl				gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
13.5	1	8	1.4	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	27
13.5	1	10	1.4	0	1	0	0	0	0	1	0	1	1	1	0	0	1	1	0	1	1	0	0	0	0	0	33.75
13.5	1	12	1.4	1	0	0	0	0	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	40.5
13.5	1	16	1.4	1	1	0	0	0	0	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	54
13.5	3	8	1.4	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	54
13.5	3	10	1.4	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	67.5
13.5	3	12	1.4	1	0	0	0	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	81
13.5	3	16	1.4	1	1	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
13.5	7	8	1.4	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
13.5	7	10	1.4	0	1	0	0	0	0	1	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	135
13.5	7	12	1.4	1	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	162
13.5	7	16	1.4	1	1	0	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	216
18	2	8	1.4	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	54
18	2	10	1.4	0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	1	0	0	0	0	1	67.5
18	2	12	1.4	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	1	81
18	2	16	1.4	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
18	5	8	1.4	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
18	5	10	1.4	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	135
18	5	12	1.4	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	0	162
18	5	16	1.4	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	216
21.6	4	8	1.4	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	108
21.6	4	12	1.4	1	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	162
21.6	4	16	1.4	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	216
21.6	9	8	1.4	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	216
21.6	9	12	1.4	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	324
21.6	9	16	2.0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	432
24	8	8	1.4	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	216
24	8	16	2.0	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	432
25.175	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	25.175
25.175	No	10	1.4	0	1	0	0	0	0	1	0	1	0	1	0	0	1	1	0	1	1	0	0	0	0	0	31.46875
25.175	No	12	1.4	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	1	1	0	0	0	0	0	37.7625

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl			fbdiv1_cntrl		ref_cntrl		n_cntrl	prop_cntrl		int_cntrl				gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
25.175	No	16	1.4	1	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1	50.35
27	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	27
27	No	10	1.4	0	1	0	0	0	0	1	0	1	0	1	0	0	1	1	0	1	1	0	0	0	0	33.75
27	No	12	1.4	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	1	1	0	0	0	0	40.5
27	No	16	1.4	1	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1	54
27	1	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1	54
27	1	10	1.4	0	1	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	0	0	1	67.5
27	1	12	1.4	1	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	0	0	1	81
27	1	16	1.4	1	1	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
27	3	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
27	3	10	1.4	0	1	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	135
27	3	12	1.4	1	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	162
27	3	16	1.4	1	1	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	216
27	7	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	216
27	7	10	1.4	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	270
27	7	12	1.4	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	324
27	7	16	2.0	1	1	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	432
31.5	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	31.5
33.75	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	33.75
35.5	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	35.5
36	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	36
36	2	8	1.4	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	0	108
36	2	10	1.4	0	1	0	0	0	0	1	0	1	1	0	0	1	0	1	1	1	1	0	0	0	1	0	135
36	2	12	1.4	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	162
36	2	16	1.4	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	216
36	5	8	1.4	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	216
36	5	10	1.4	0	1	0	0	0	0	1	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	270
36	5	12	1.4	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	324
36	5	16	2.0	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	432
40	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	40
43.2	4	8	1.4	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	216
43.2	4	12	1.4	1	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	324
43.2	4	16	2.0	1	1	1	1	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	432

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl			fbdiv1_cntrl		ref_cntrl		n_cntrl	prop_cntrl		int_cntrl				gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
44.9	No	8	1.4	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	44.9
49.5	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	1	49.5
50	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	1	50
50.35	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	1	50.35
50.35	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	1	62.9375
50.35	No	12	1.4	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	75.525
50.35	No	16	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	100.7
54	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	1	54
54	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	1	67.5
54	No	12	1.4	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	81
54	No	16	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
54	1	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	108
54	1	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	1	0	135
54	1	12	1.4	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	0	162
54	1	16	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	216
54	3	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	216
54	3	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	270
54	3	12	1.4	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	324
54	3	16	2.0	1	1	1	1	0	1	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	432
56.25	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	56.25
59.4	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	59.4
59.4	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	74.25
59.4	No	12	1.4	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	1	89.1
59.4	No	16	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	118.8
65	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	65
68.25	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	68.25
71	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	71
72	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	72
72	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	90
72	No	12	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	108
72	No	16	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	144
72	2	8	1.4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	216
72	2	10	1.4	0	1	0	0	0	0	1	0	1	1	0	1	1	0	0	1	1	1	0	0	0	1	1	270

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl			fbdiv1_cntrl		ref_cntrl		n_cntrl	prop_cntrl			int_cntrl			gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
7 2	2	1 2	1.4	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	324
7 2	2	1 6	2.0	1	1	1	1	0	1	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	432
73.25	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	73.25
74.25	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	74.25
74.25	No	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	92.8125
74.25	No	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	111.375
74.25	No	1 6	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	148.5
7 5	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	7 5
78.75	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	78.75
79.5	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	79.5
82.5	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	82.5
82.5	No	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	103.125
82.5	No	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	123.75
82.5	No	1 6	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	165
83.5	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	83.5
85.5	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	85.5
88.75	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	88.75
9 0	No	8	1.4	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	1	9 0
9 0	No	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	112.5
9 0	No	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	135
9 0	No	1 6	1.4	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	180
94.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	94.5
9 9	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	9 9
9 9	No	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	123.75
9 9	No	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	148.5
9 9	No	1 6	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	198
100.7	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	100.7
100.7	No	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	125.875
100.7	No	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	151.05
100.7	No	1 6	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	201.4
101	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	101
102.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	102.25
106.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	106.5

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl			fbdiv1_cntrl		ref_cntrl		n_cntrl	prop_cntrl		int_cntrl				gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
108	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	108
108	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	135
108	No	12	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	162
108	No	16	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	216
108	1	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	216
108	1	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	270
108	1	12	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	324
108	1	16	2.0	1	1	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	432
115.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	115.5
117.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	117.5
118.8	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	118.8
118.8	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	148.5
118.8	No	12	1.4	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	178.2
118.8	No	16	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	237.6
119	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	119
121.75	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	121.75
122.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	122.5
135	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	135
136.75	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	136.75
140.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	140.25
144	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	144
144	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	180
144	No	12	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	216
144	No	16	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	288
146.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	146.25
148.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	148.25
148.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	148.5
148.5	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	185.625
148.5	No	12	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	222.75
148.5	No	16	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	297
154	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	154
156	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	156
157	No	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	157

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl			fbdiv1_cntrl		ref_cntrl		n_cntrl	prop_cntrl			int_cntrl			gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
157.5	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	157.5
162	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	162
165	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	165
165	N o	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	206.25
165	N o	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	247.5
165	N o	1 6	1.4	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	330
175.5	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	175.5
179.5	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	179.5
180	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	180
180	N o	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	225
180	N o	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	270
180	N o	1 6	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	360
182.75	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	182.75
185.625	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	185.625
185.625	N o	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	232.03125
185.625	N o	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	278.4375
185.625	N o	1 6	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	371.25
187	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	187
187.25	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	187.25
189	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	189
193.25	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	193.25
198	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	198
198	N o	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	247.5
198	N o	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	297
198	N o	1 6	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	396
202.5	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	202.5
204.75	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	204.75
208	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	208
214.75	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	214.75
216	N o	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	216
216	N o	1 0	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	270
216	N o	1 2	1.4	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	324
216	N o	1 6	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	432

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl			fbdiv1_cntrl		ref_cntrl		n_cntrl	prop_cntrl		int_cntrl				gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
218.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	218.25
229.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	229.5
234	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	234
237.6	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	237.6
237.6	No	10	1.4	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	297
237.6	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	356.4
237.6	No	16	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	475.2
245.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	245.25
245.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	245.5
261	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	261
268.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	268.25
268.5	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	268.5
281.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	281.25
288	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	288
288	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	360
288	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	432
288	No	16	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	576
297	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	297
297	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	371.25
297	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	445.5
297	No	16	2.0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	594
317	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	317
330	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	330
330	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	412.5
330	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	495
333.25	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	333.25
340	No	8	1.4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	340
348.5	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	348.5
356.5	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	356.5
360	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	360
360	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	450
360	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	540
371.25	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	371.25

Pixel Clock (MHz)	Pixel Repetition	Color depth [bits]	OPMODE(HDMI1.4 OR 2.0)	Divider Settings				opmode		Divider settings								MPLL Charge pump settings								TMDSCLKSRC (MHz)
				prep_div		tmdsmhl_cntrl				fbdiv2_cntrl		fbdiv1_cntrl		ref_cntrl		n_cntrl		prop_cntrl			int_cntrl			gmp_cntrl		
				0006:Bit 14	0006:Bit 13	0006:Bit 12	0006:Bit 11	0006:Bit 10	0006:Bit 9	0006:Bit 8	0006:Bit 7	0006:Bit 6	0006:Bit 5	0006:Bit 4	0006:Bit 3	0006:Bit 2	0006:Bit 1	0006:Bit 0	0010:Bit 5	0010:Bit 4	0010:Bit 3	0010:Bit 2	0010:Bit 1	0010:Bit 0	0015:Bit 1	0015:Bit 0	
371.25	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	464.0625
371.25	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	556.875
380.5	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	380.5
396	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	396
396	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	495
396	No	12	2.0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	594
432	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	432
432	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	540
443.25	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	443.25
475.2	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	475.2
475.2	No	10	2.0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	594
495	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	495
505.25	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	505.25
552.75	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	552.75
594	No	8	2.0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	594

Chapter 6 eDP TX Controller

6.1 Overview

eDP TX Controller is compliant with DisplayPort standard 1.2a and eDP 1.3. DisplayPort is an industry standard to accommodate the growing broad adoption of digital display technology within the PC and consumer electronics (CE) industries. It consolidates the internal and external connection methods to reduce device complexity and cost, supports necessary features for key cross industry applications, and provides performance scalability to enable the next generation of displays featuring higher color depths, refresh rates, and display resolutions.

It supports following features:

- Compliant with DisplayPortTM Specification, Version 1.2
- Compliant with eDPTM Specification, Version 1.3
- Main link containing 4 physical lanes of 2.7/1.62 Gbps/lane
- TX PHY lanes, control pins and hot-plug pins are shared by the DisplayPort Source
- Bi-directional auxiliary link with up to 1Mbps speed
- RGB 8/10 bit per component video format
- Video slave mode
- Support PSR
- APB slave bus interface
- Hot plug and unplug detection and link status monitor
- Support VESA DMT and CVT timing standards
- Fully support EIA/CEA-861D video timing and Info Frame structure
- Supports reading of the display EDID whenever the display is connected to power, even an AC-trickle power
- Up to 0.5% down-spreading support at high-speed link
- PRBS or programmable transmitter pattern for main link quality test
- 24 Mhz crystal clock input
- Built-in video BIST patterns

6.2 Block Diagram

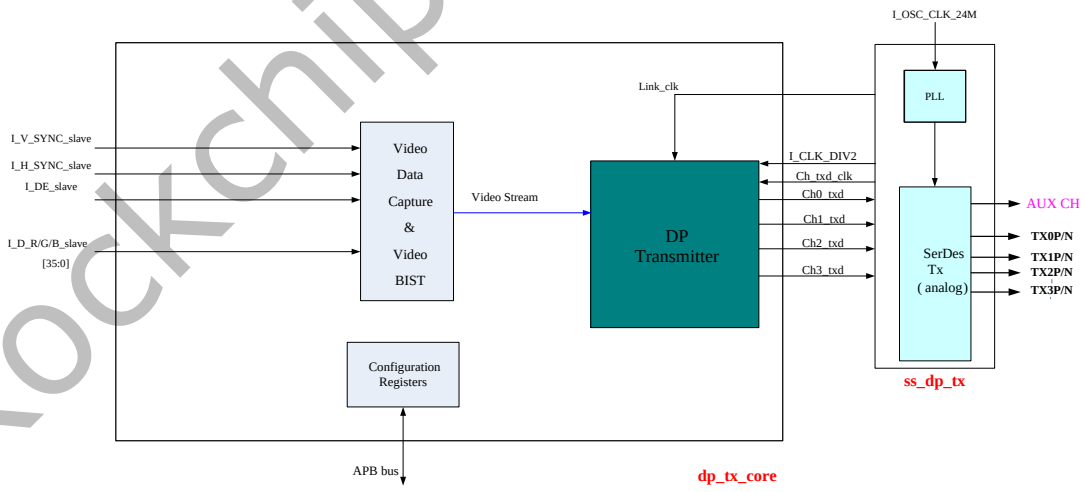


Fig. 6-1 eDP TX controller Block Diagram

Above figure shows the block diagram of eDP TX controller in top level. The video data and clock are sent directly from the video output processor.

The video data capture & video BIST block is separated as video_capture and display_bist module. The block before SerDes is DP_TX main module. Following Table shows the brief function description of each sub-module.

Table 6-1 Brief function description of each module in eDP TX controller

Module Name in Top Level	Brief Module Function Description
video_capture	Capture block of video data.
display_bist	Generation of arbitrary video format with three types of video data. The output of display_bist module will input to video_capture module directly if display BIST mode is active.
dp_tx	DisplayPort transmitter block.
apb_slave_top	APB slave Bus interface

6.3 Function Description

6.3.1 eDP TX controller in SoC

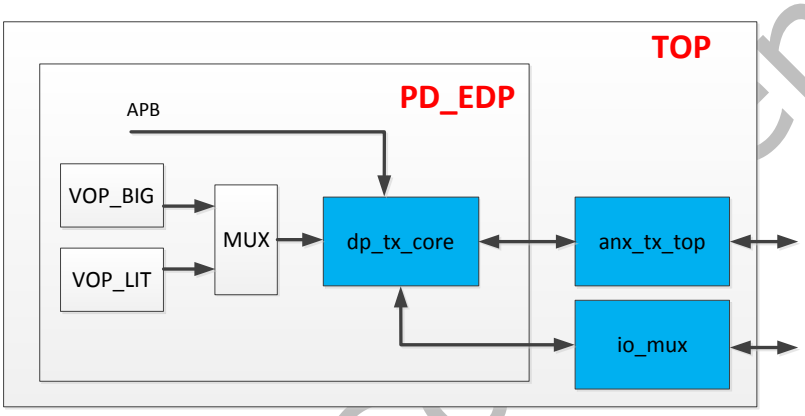


Fig. 6-2 edp TX controller in SoC

6.3.2 Video Capture

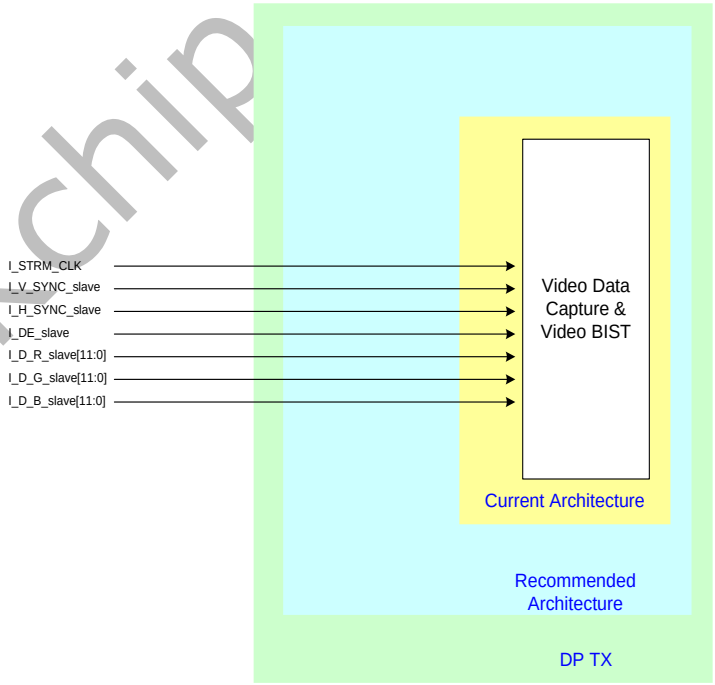


Fig. 6-3 Block diagram of Video Interface

6.3.3 Video slave mode interface

In this slave mode, all timing signals, video data and clock are provided to the DP_TX. All such signals are synchronous to the input clock (I_STRM_CLK) for slave mode. DP_TX receives such signals and stores the video signals into the buffer. Afterward, the video data

packetizer which operates on LS_CLK clock (135/81MHz) reads from the buffer asynchronously.

6.3.4 DisplayPort AUX CH

The auxiliary link is a low speed (1Mbps) half-duplex bi-directional link used to get state/attributes, such as link capabilities, link status and the EDID, from the receiver. The transmitter will also initiate the link training through the AUX link.
It is used to perform the native AUX CH link service and device services, such as DPCD reading/writing, link initialization, training and monitoring. It is also used to bridge the APB transactions (such as the EDID read) between the GPU and the display devices.
The DisplayPort Source Device must weakly pull up the AUX CH lines between the AC-coupling capacitors and the Source Connector to assist source detection by the Sink Device. Source detection capable Sink Device must have ac-coupling capacitors. The Sink Device must pull down the AUX CH lines with 1MΩ resistors between the sink connector and the ac-coupling capacitors, and monitor the DC voltage of AUX CH line.
The AUX CH uses the Manchester-II code for the self-clocked transmission of signals as shown below in follows.
The data format of request transaction is shown as following figure,

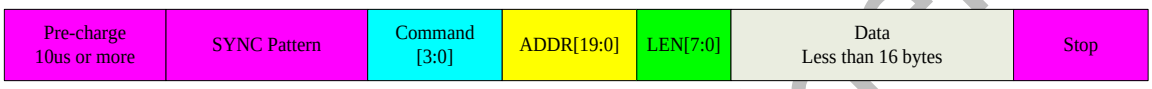


Fig. 6-4 AUX CH Request Transaction Data Format

The reply transaction is shown as following figure

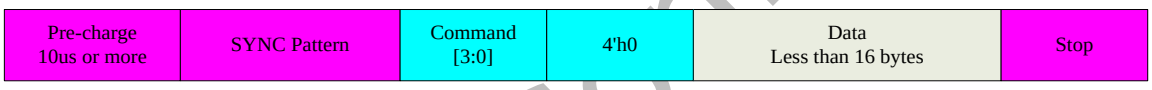


Fig. 6-5 AUX CH Reply Transaction Data Format

Pre-charge: 10 to 16 consecutive 0's in Manchester II code to pre-charge the AUX-CH+ and AUX-CH- to a common mode voltage.
Sync Pattern: Start with 16 consecutive 0's in Manchester II code, end with the AUX-CH+ driven to high for a two bit period and low for a two bit period, which is illegal in Manchester II code. The AUX-CH- must be driven to the opposite polarity.

Request Command:

COMM[3]	COMM[2]	COMM[1]	COMM[0]
1: DisplayPort transaction	000: Request type is write 001: Request type is read Others: Reserved		
0: I2C transaction	1: MOT (Middle-of-Transaction) = 1 0: MOT (Middle-of-Transaction) = 0	00 : write 01 : read 00 : write_status_request 01 : Reserved	

Reply Command:

COMM[3]	COMM[2]	COMM[1]	COMM[0]
I^2C-over-AUX Reply field: 00: I^2C_ACK. 01: I^2C_NACK. 10: I^2C_DEFER. 11: Reserved.		Native AUX CH Reply field: 00: AUX_ACK. 01: AUX_NACK. 10: AUX_DEFER. 11: Reserved.	

Stop: AUX-CH+ driven to high for a two bit period and low for a two bit period, which is illegal in Manchester II code. The AUX-CH- must be driven to the opposite polarity.

6.4 Register Description

Terminology

	Description
R/C	Read-clear, field will return read value and be cleared by hardware.
RO	Read only register, field may be read but not written. Write have no effect.
R/W	Read/Write accessible register, filed may be written or read. Read-data will equal write data.
S/C or SC	Write-auto-clear, field may be written to and will be cleared by hardware.
W/C1	Write-once-clear, field will be cleared to zero if written with a one.
WO	Write-only, field may be written to, but reads will always return zero.
W/OC	Write-once, field may be written to, but hardware will update read value.
R/U	Read-update, field will be updated by hardware after read
Reserved	A reserved register bit may not be physically implemented in the device or could be implemented to control a proprietary function. Therefore, register bits labeled "Reserved" should not be changed by software. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is to say, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. For conciseness, this document notes the default value of a reserved register bit as "0" even though default values for reserved bits may actually be unspecified.
[mm:nn]	This denotes a group of register bits: mm represents the most significant bits and nn the least significant among the quantity referenced. By convention, the less significant bits have lower offset, so that bit 0 in a register is always the least significant bit.
Table X-X	Blue text indicates a linked cross-reference. Click on the link to jump to the reference information.

6.4.1 Registers Summary

Register MAP

General Control Register Definition

General Control Register Definition				
Name	**Address**	**Type**	**Description**	**Reset Value**
DP_TX_VERSION	Base + 0x0010	RO	DP_TX version register	0x0000_0060
FUNC_EN_1	Base + 0x0018	R/W	Function Enable Register 1	0x0000_007D
FUNC_EN_2	Base + 0x001C	R/W	Function Enable Register 2	0x0000_0087
VIDEO_CTL_1	Base + 0x0020	R/W	Video Control 1	0x0000_0000
VIDEO_CTL_2	Base + 0x0024	R/W	Video Control 2	0x0000_0010

General Control Register Definition				
Name	**Address**	**Type**	**Description**	**Reset Value**
VIDEO_CTL_3	Base + 0x0028	R/W	Video Control 3	0x0000_0000
VIDEO_CTL_4	Base + 0x002C	R/W	Video Control 4	0x0000_0000
VIDEO_CTL_8	Base + 0x003C	R/W	Video Control 8	0x0000_0020
VIDEO_CTL_10	Base + 0x0044	R/W	Video Control 10	0x0000_0000
TOTAL_LINE_CFG_L	Base + 0x0048	R/W	Total Line Low Byte Configure Register	0x0000_0000
TOTAL_LINE_CFG_H	Base + 0x004C	R/W	Total Line High Byte Configure Register	0x0000_0000
ACTIVE_LINE_CFG_L	Base + 0x0050	R/W	Active Line Low Byte Configure Register	0x0000_0000
ACTIVE_LINE_CFG_H	Base + 0x0054	R/W	Active Line High Byte Configure Register	0x0000_0000
V_F_PORCH_CFG	Base + 0x0058	R/W	Vertical Front Porch Configure Register	0x0000_0000
V_SYNC_WIDTH_CFG	Base + 0x005C	R/W	Vertical Sync Width Configure Register	0x0000_0000
V_B_PORCH_CFG	Base + 0x0060	R/W	Vertical Back Porch Configure Register	0x0000_0000
TOTAL_PIXEL_CFG_L	Base + 0x0064	R/W	Total Pixel Low Byte Configure Register	0x0000_0000
TOTAL_PIXEL_CFG_H	Base + 0x0068	R/W	Total Pixel High Byte Configure Register	0x0000_0000
ACTIVE_PIXEL_CFG_L	Base + 0x006C	R/W	Active Pixel Low Byte Configure Register	0x0000_0000
ACTIVE_PIXEL_CFG_H	Base + 0x0070	R/W	Active Pixel High Byte Configure Register	0x0000_0000
H_F_PORCH_CFG_L	Base + 0x0074	R/W	Horizon Front Porch Low Byte Configure Register	0x0000_0000
H_F_PORCH_CFG_H	Base + 0x0078	R/W	Horizon Front Porch High Byte Configure Register	0x0000_0000
H_SYNC_CFG_L	Base + 0x007C	R/W	Horizon Sync Width Low Byte Configure Register	0x0000_0000
H_SYNC_CFG_H	Base + 0x0080	R/W	Horizon Sync Width High Byte Configure Register	0x0000_0000
H_B_PORCH_CFG_L	Base + 0x0084	R/W	Horizon Back Porch Low Byte Configure Register	0x0000_0000
H_B_PORCH_CFG_H	Base + 0x0088	R/W	Horizon Back Porch High Byte Configure Register	0x0000_0000
VIDEO_STATUS	Base + 0x008C	RO	Video Status Register	0x0000_0003
TOTAL_LINE_STA_L	Base + 0x0090	RO	Total Line Status Low Byte Register	0x0000_0001
TOTAL_LINE_STA_H	Base + 0x0094	RO	Total Line Status High Byte Register	0x0000_0000
ACTIVE_LINE_STA_L	Base + 0x0098	RO	Active Line Status Low Byte Register	0x0000_0000
ACTIVE_LINE_STA_H	Base + 0x009C	RO	Active Line Status High Byte Register	0x0000_0000
V_F_PORCH_STA	Base + 0x00A0	RO	Vertical Front Porch Status Register	0x0000_0001
V_SYNC_STA	Base + 0x00A4	RO	Vertical Sync Width Status Register	0x0000_0000
V_B_PORCH_STA	Base + 0x00A8	RO	Vertical Back Porch Status Register	0x0000_0000
TOTAL_PIXEL_STA_L	Base + 0x00AC	RO	Total Pixel Status Low	0x0000_0000

General Control Register Definition				
Name	**Address**	**Type**	**Description**	**Reset Value**
			Byte Register	
TOTAL_PIXEL_STA_H	Base + 0x00B0	RO	Total Pixel Status High Byte Register	0x0000_0000
ACTIVE_PIXEL_STA_L	Base + 0x00B4	RO	Active Pixel Status Low Byte Register	0x0000_0000
ACTIVE_PIXEL_STA_H	Base + 0x00B8	RO	Active Pixel Status High Byte Register	0x0000_0000
H_F_PORCH_STA_L	Base + 0x00BC	RO	Horizon Front Porch Status Low Byte Register	0x0000_0000
H_F_PORCH_STA_H	Base + 0x00C0	RO	Horizon Front Porch Status High Byte Register	0x0000_0000
H_SYNC_STA_L	Base + 0x00C4	RO	Horizon Sync Width Status Low Byte Register	0x0000_0000
H_SYNC_STA_H	Base + 0x00C8	RO	Horizon Sync Width Status High Byte Register	0x0000_0000
H_B_PORCH_STA_L	Base + 0x00CC	RO	Horizon Back Porch Status Low Byte Register	0x0000_0000
H_B_PORCH_STA_H	Base + 0x00D0	RO	Horizon Back Porch Status High Byte Register	0x0000_0000
AVI_DB1 ~ AVI_DB13	Base + 0x01D0~ Base + 0x0200	R/W	AVI InfoFrame Packet Data Byte	0x0000_0000
IF_TYPE	Base + 0x0244	R/W	InfoFrame Packet Type Code.	0x0000_0000
IF_PKT_DB1~ IF_PKT_DB25	Base + 0x0254~ Base + 0x02B4	R/W	InfoFrame Packet Data Byte	0x0000_0000
MPEG_DB1~ MPEG_DB10	Base + 0x02D0~ Base + 0x02F4	R/W	MPEG Source InfoFrame Packet Data Byte	0x0000_0000
PSR_FRAME_UPDATA_CTRL	Base + 0x0318	R/W	Frame update control for PSR	0x0000_0000
VSC_SHADOW_DB0~ VSC_SHADOW_DB7	Base+ 0x031C~ Base+ 0x0338	R/W	VSC shadow data bytes 0 ~ 7	0x0000_0000
VSC_SHADOW_PB0~ VSC_SHADOW_PB1	Base+ 0x033C~ Base+ 0x0340	R/W	VSC shadow parity byte 0 ~ 1	0x0000_0000
LANE_MAP	Base + 0x035C	R/W	Lane Map Register	0x0000_00E4
ANALOG_CTL_2	Base + 0x0374	R/W	Analog Control Register 2	0x0000_0008
HIDDEN_REG	Base + 0x0390	R/W	Debug Register	0x0000_0003
INT_STATE	Base + 0x03C0	RO	Interrupt Status Register	0x0000_0000
COMMON_INT_STA_1	Base + 0x03C4	R/W	Common Interrupt Status Register 1	0x0000_0000
COMMON_INT_STA_3	Base + 0x03CC	R/W/C 1	Common Interrupt Status Register 3	0x0000_0000
COMMON_INT_STA_4	Base + 0x03D0	R/W/C 1	Common Interrupt Status Register 4	0x0000_0000
DP_INT_STA	Base + 0x03DC	R/W/C 1	DisplayPort Interrupt Status Register	0x0000_0000
COMMON_INT_MASK_1	Base + 0x03E0	R/W	Common Interrupt Mask Register1	0x0000_0000
COMMON_INT_MASK_3	Base + 0x03E8	R/W	Common Interrupt Mask Register3	0x0000_0000
COMMON_INT_MASK_4	Base + 0x03EC	R/W	Common Interrupt Mask Register4	0x0000_0000
DP_INT_STA_MASK	Base + 0x03F8	R/W	DisplayPort Interrupt enable Register	0x0000_0000
INT_CTL	Base + 0x03FC	R/W	Interrupt Control Register	0x0000_0000

DisplayPort Register Definition

Register Definition for DisplayPort Function				
Name	**Address**	**Type**	**Description**	**Reset Value**
SYS_CTL_1	Base + 0x0600	R/W	System Control Register #1	0x0000_0000
SYS_CTL_2	Base + 0x0604	R/W	System Control Register #2	0X0000_0040
SYS_CTL_3	Base + 0x0608	R/W, RO	System Control Register #3	0x0000_0000
SYS_CTL_4	Base + 0x060C	R/W	System Control Register #4	0x0000_0000
DP_VID_CTL	Base + 0x0610	RO	DP Video Control Register	0x0000_0020
PKT_SEND_CTL	Base + 0x0640	R/W	Packet Send Control Register	0x0000_0000
LINK_BW_SET	Base + 0x0680	R/W	Main Link Bandwidth Setting Register	0x0000_000A
LANE_COUNT_SET	Base + 0x0684	R/W	DP Main Link Lane Number Register	0x0000_0004
DP_TRAINING_PTN_SET	Base + 0x0688	R/W	DP Training Pattern Set Register	0x0000_0000
DP_LN0_LINK_TRAINING_CTL	Base + 0x068C	R/W, RO	DP Lane 0 Link Training Control Register	0x0000_0000
DP_LN1_LINK_TRAINING_CTL	Base + 0x0690	R/W, RO	DP Lane 1 Link Training Control Register	0x0000_0000
DP_LN2_LINK_TRAINING_CTL	Base + 0x0694	R/W, RO	DP Lane 2 Link Training Control Register	0x0000_0000
DP_LN3_LINK_TRAINING_CTL	Base + 0x0698	R/W, RO	DP Lane 3 Link Training Control Register	0x0000_0000
DP_HW_LINK_TRAINING_CTL	Base + 0x06A0	R/W, RO	DP HW LINK TRAINING_CONTROL Register	0x0000_0000
DP_DEBUG_CTL	Base + 0x06C0	R R/W	DP Debug Control Register #1	0x0000_0000
HPD_DEGLITCH_L	Base + 0x06C4	R/W	DP HPD De-glitch Low Byte Register	0x0000_005E
HPD_DEGLITCH_H	Base + 0x06C8	R/W	DP HPD De-glitch High Byte Register	0x0000_001A
POLLING_PERIOD	Base + 0x06CC	R/W	DP polling period	0x0000_000E
DP_LINK_DEBUG_CTL	Base + 0x06E0	R/W	DP Link Debug Control Register	0x0000_0010
DP_SINK_COUNT	Base + 0x06E4	RO	DP Sink Count	0x0000_0000
DP_IRD_VECTOR	Base + 0x06E8	RO	DP Irq Vector	0x0000_0000
DP_LINK_STATUS0	Base + 0x06EC	RO	DP Lane0 and Lane1 Status	0x0000_0000
DP_LINK_STATUS1	Base + 0x06F0	RO	DP Lane2 and Lane3 Status	0x0000_0000
DP_ALIGN_STATUS`	Base + 0x06F4	RO	DP Align Status	0x0000_0000
DP_SINK_STATUS	Base + 0x06F8	RO	DP Sink Status	0x0000_0000
M_VID_0	Base + 0x0700	R/W	DP M_VID Configure Register #0	0x0000_0000
M_VID_1	Base + 0x0704	R/W	DP M_VID Configure Register #1	.0x0000_0000
M_VID_2	Base + 0x0708	R/W	DP M_VID Configure	0x0000_0000

Register Definition for DisplayPort Function				
Name	**Address**	**Type**	**Description**	**Reset Value**
			Register #2	
N_VID_0	Base + 0x070C	R/W	DP N_VID Configure Register #0	0x0000_0000
N_VID_1	Base + 0x0710	R/W	DP N_VID Configure Register #1	0x0000_0080
N_VID_2	Base + 0x0714	R/W	DP N_VID Configure Register #2	0x0000_0000
M_VID_MON	Base + 0x0718	RO	DP M_VID value monitoring register	0x0000_0000
DP_VIDEO_FIFO_THRD	Base + 0x0730	R/W	DP FIFO Threshold Register	0x0000_0000
DP_M_CAL_CTL	Base + 0x0760	R/W	DP M Value Calculation Control Register	0x0000_0000
M_VID_GEN_FILTER_TH	Base + 0x0764	R/W	DP M_VID Value Calculation Control Register	0x0000_0004
AUX_CH_STA	Base + 0x0780	R/W	AUX Channel Access Status Register	0x0000_0000
AUX_ERR_NUM	Base + 0x0784	RO	AUX Channel Access Error Code Register	0x0000_0000
AUX_CH_DEFER_CTL	Base + 0x0788	R/W	DP AUX CH DEFER Control Register	0x0000_007F
AUX_RX_COMM	Base + 0x078C	RO	DP AUX RX Command Register	0x0000_0000
BUFFER_DATA_CTL	Base + 0x0790	R/W, RO	DP Buffer Data Count Register	0x0000_0000
AUX_CH_CTL_1	Base + 0x0794	R/W	DP AUX Channel Control Register 1	0x0000_0000
AUX_ADDR_7_0	Base + 0x0798	R/W	DP AUX CH Address Register #0	0x0000_0000
AUX_ADDR_15_8	Base + 0x079C	R/W	DP AUX CH Address Register #1	0x0000_0000
AUX_ADDR_19_16	Base + 0x07A0	R/W	DP AUX CH Address Register #2	0x0000_0000
AUX_CH_CTL_2	Base + 0x07A4	R/W	DP AUX CH Control Register 2	0x0000_0000
BUF_DATA_0	Base + 0x07C0	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_1	Base + 0x07C4	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_2	Base + 0x07C8	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_3	Base + 0x07CC	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_4	Base + 0x07D0	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_5	Base + 0x07D4	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_6	Base + 0x07D8	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_7	Base + 0x07DC	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_8	Base + 0x07E0	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_9	Base + 0x07E4	R/W	AUX CH buffer data	0x0000_00FF

Register Definition for DisplayPort Function				
Name	**Address**	**Type**	**Description**	**Reset Value**
BUF_DATA_10	Base + 0x07E8	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_11	Base + 0x07EC	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_12	Base + 0x07F0	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_13	Base + 0x07F4	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_14	Base + 0x07F8	R/W	AUX CH buffer data	0x0000_00FF
BUF_DATA_15	Base + 0x07FC	R/W	AUX CH buffer data	0x0000_00FF

Register definitions for SoC IP implementation

Register definitions for SoC IP implementation				
Name	**Address**	**Type**	**Description**	**Reset Value**
ATE_TEST_CTL	Base + 0x0804	R/W SC	ATE test control register	0x0000_0000
ATE_TEST_STATUS	Base + 0x0808	RO	ATE test status register	0x0000_0000
ATE_TEST_ERR_CNT0	Base + 0x080C	RO	Lane0 ATE test error counter register	0x0000_0000
ATE_TEST_ERR_CNT1	Base + 0x0810	RO	Lane1 ATE test error counter register	0x0000_0000
ATE_TEST_ERR_CNT2	Base + 0x0814	RO	Lane2 ATE test error counter register	0x0000_0000
ATE_TEST_ERR_CNT3	Base + 0x0818	RO	Lane3 ATE test error counter register	0x0000_0000
DP_TEST_80B_PATTERN0	Base + 0x081C	R/W	80b pattern [29:0]	0x0000_0000
DP_TEST_80B_PATTERN1	Base + 0x0820	R/W	80b pattern [59:30]	0x0000_0000
DP_TEST_80B_PATTERN2	Base + 0x0824	R/W	80b pattern [79:60]	0x0000_0000
DP_TEST_HBR2_PATTERN	Base + 0x0828	R/W	Hbr2 compliance SR count	0x0000_0000
CRC_CON	Base + 0x0890	R/W	CRC control register	0x0000_0000
ANALOG_CTL_5	Base + 0x0914	R/W	PC2 Control Register	0x0000_0000
ANALOG_CTL_6	Base + 0x0918	R/W	AMP_400MV_0DB	0x0000_0050
ANALOG_CTL_7	Base + 0x091C	R/W	AMP_600MV_0DB	0x0000_0078
ANALOG_CTL_8	Base + 0x0920	R/W	AMP_800MV_0DB	0x0000_00A0
ANALOG_CTL_9	Base + 0x0924	R/W	AMP_1200MV_0DB	0x0000_00F0
ANALOG_CTL_10	Base + 0x0928	R/W	AMP_400MV_3P5DB	0x0000_0064
ANALOG_CTL_11	Base + 0x092C	R/W	AMP_600MV_3P5DB	0x0000_0096
ANALOG_CTL_12	Base + 0x0930	R/W	AMP_800MV_3P5DB	0x0000_00C8
ANALOG_CTL_13	Base + 0x0934	R/W	AMP_400MV_6DB	0x0000_0078
ANALOG_CTL_14	Base + 0x0938	R/W	AMP_600MV_6DB	0x0000_00B4
ANALOG_CTL_15	Base + 0x093C	R/W	AMP_400MV_9DB	0x0000_00A0
ANALOG_CTL_16	Base + 0x0940	R/W	EMP_400MV_0DB	0x0000_0000
ANALOG_CTL_17	Base + 0x0944	R/W	EMP_600MV_0DB	0x0000_0000
ANALOG_CTL_18	Base + 0x0948	R/W	EMP_800MV_0DB	0x0000_0000
ANALOG_CTL_19	Base + 0x094C	R/W	EMP_1200MV_0DB	0x0000_0000
ANALOG_CTL_20	Base + 0x0950	R/W	EMP_400MV_3P5DB	0x0000_0028
ANALOG_CTL_21	Base + 0x0954	R/W	EMP_600MV_3P5DB	0x0000_003C
ANALOG_CTL_22	Base + 0x0958	R/W	EMP_800MV_3P5DB	0x0000_0050
ANALOG_CTL_23	Base + 0x095C	R/W	EMP_400MV_6DB	0x0000_0050
ANALOG_CTL_24	Base + 0x0960	R/W	EMP_600MV_6DB	0x0000_0078
ANALOG_CTL_25	Base + 0x0964	R/W	EMP_400MV_9DB	0x0000_00A0
ANALOG_CTL_26	Base + 0x0968	R/W	PC2_400MV_0DB	0x0000_0004

Register definitions for SoC IP implementation				
Name	**Address**	**Type**	**Description**	**Reset Value**
ANALOG_CTL_27	Base + 0x096C	R/W	PC2_600MV_0DB	0x0000_0006
ANALOG_CTL_28	Base + 0x0970	R/W	PC2_800MV_0DB	0x0000_0008
ANALOG_CTL_29	Base + 0x0974	R/W	PC2_1200MV_0DB	0x0000_000C
ANALOG_CTL_30	Base + 0x0978	R/W	PC2_400MV_3P5DB	0x0000_0006
ANALOG_CTL_31	Base + 0x097C	R/W	PC2_600MV_3P5DB	0x0000_0009
ANALOG_CTL_32	Base + 0x0980	R/W	PC2_800MV_3P5DB	0x0000_000C
ANALOG_CTL_33	Base + 0x0984	R/W	PC2_400MV_6DB	0x0000_0008
ANALOG_CTL_34	Base + 0x0988	R/W	PC2_600MV_6DB	0x0000_000C
ANALOG_CTL_35	Base + 0x098C	R/W	PC2_400MV_9DB	0x0000_000C
ANALOG_CTL_36	Base + 0x0990	R/W	CH0_AMP_FORCE_VALUE	0x0000_0050
ANALOG_CTL_37	Base + 0x0994	R/W	CH0_EMP_FORCE_VALUE	0x0000_0000
ANALOG_CTL_38	Base + 0x0998	R/W	CH0_PC2_FORCE_VALUE	0x0000_0004
ANALOG_CTL_39	Base + 0x099C	R/W	CH1_AMP_FORCE_VALUE	0x0000_0050
ANALOG_CTL_40	Base + 0x09A0	R/W	CH1_EMP_FORCE_VALUE	0x0000_0000
ANALOG_CTL_41	Base + 0x09A4	R/W	CH1_PC2_FORCE_VALUE	0x0000_0004
ANALOG_CTL_42	Base + 0x09A8	R/W	CH0_CH1_FORCE_CTRL	0x0000_0000
ANALOG_CTL_43	Base + 0x09AC	R/W	CH2_AMP_FORCE_VALUE	0x0000_0050
ANALOG_CTL_44	Base + 0x09B0	R/W	CH2_EMP_FORCE_VALUE	0x0000_0000
ANALOG_CTL_45	Base + 0x09B4	R/W	CH2_PC2_FORCE_VALUE	0x0000_0004
ANALOG_CTL_46	Base + 0x09B8	R/W	CH3_AMP_FORCE_VALUE	0x0000_0050
ANALOG_CTL_47	Base + 0x09BC	R/W	CH3_EMP_FORCE_VALUE	0x0000_0000
ANALOG_CTL_48	Base + 0x09C0	R/W	CH3_PC2_FORCE_VALUE	0x0000_0004
ANALOG_CTL_49	Base + 0x09C4	R/W	CH2_CH3_FORCE_CTRL	0x0000_0000
LINK_POLICY	Base + 0x09D8	R/W	Dp Link Policy	0x0000_0050
PLL_REG_1	Base + 0x00FC	R/W	Pll_control_1	0x0000_0011
PLL_REG_2	Base + 0x09E4	R/W	Pll_control_2	0x0000_0011
PLL_REG_3	Base + 0x09E8	R/W	Pll_control_3	0x0000_002B
PLL_REG_4	Base + 0x09EC	R/W	Pll_control_4	0x0000_0023
PLL_REG_5	Base + 0x0A00	R/W	Pll_control_5	0x0000_0000
PLL_REG_MAC	Base + 0x0A04	R/W	Pll_control_MAC	0x0000_0000
FREQ_IN_REG	Base + 0x0A10	R/W	freq_in_reg	0x0000_0080
P_REG_FRQ	Base + 0x0A14	RO	frequency counter ,digital output for debug	0x0000_0000
P_REG_FRQ_COUNT_RDY	Base + 0x0A18	RO	frequency counter ready indicator	0x0000_0000
P_BAND_DEC_RESET	Base + 0x0A1C	WO	reset band decoder	0x0000_0000
SSC_REG	Base + 0x0104	R/W	SSC control	0x0000_000A
TX_COMMON	Base + 0x0114	R/W	Tx terminal resistor control	0x0000_003A
TX_COMMON2	Base + 0x0118	R/W	Tx terminal resistor control2	0x0000_0050
TX_COMMON3	Base + 0x0A08	R/W	Tx terminal resistor control3	0x0000_0000
DP_AUX	Base + 0x0120	R/W	Aux control	0x0000_0007
DP_BIAS	Base + 0x0124	R/W	Bias control	0x0000_0034
DP_TEST	Base + 0x0128	R/W	Test mode	0x0000_0000
DP_PD	Base + 0x012C	R/W	Power down control	0x0000_00FF
DP_RESERV1	Base + 0x0130	R/W	RESERVD1	0x0000_0000
DP_RESERV2	Base + 0x0134	R/W	RESERVD2	0x0000_0011

6.4.2 Detail Register Description

General Control Register Definition

DP_TX Version (DP_TX_VERSION)

Register	Address	Type	Description	Reset Value
DP_TX_VERSION	Base + 0x0010	RO	DP_TX_VERSION	0x0000_0060

DP_TX_VERSION	Bit	Description	Initial State
DP_TX_VERSION	[31:0]	<7:5> process 011 (Global Foundry 28nm) <4:3>: Version 00: Rev A <2:0>: Minor revision 000: rev .1 Notion: It is effective when PAD_DVDD is supplied	0x60

Function Enable Register 1(FUNC_EN_1)

Register	Address	Type	Description	Reset Value
FUNC_EN_1	Base + 0x0018	R/W	Function Enable Register 1	0x0000_007D

FUNC_EN_1	Bit	Description	Initial State
-	[31:7]	Reserved	0
VID_CAP_FUNC_EN_N	[6]	Video capture functions enable. 0: Normal operation, 1: Disable video capture.	1
VID_FIFO_FUNC_EN_N	[5]	Video FIFO functions enable. 0: Normal operation, 1: Disable video FIFO.	1
-	[4:1]	Reserved	0
SW_FUNC_EN_N	[0]	Software defined function enable. 0: Normal operation, 1: Disable All the function modules. The bit has the highest priority, if the bit is 1, other function enable bits does not work.	1

Function Enable Register 2 (FUNC_EN_2)

Register	Address	Type	Description	Reset Value
FUNC_EN_2	Base + 0x001C	R/W	Function Enable Register 2	0x0000_0087

FUNC_EN_2	Bit	Description	Initial State
-	[31:8]	Reserved	0
SSC_FUNC_EN_N	[7]	SSC module enable. 0: Normal mode, 1: Disable SSC module. To apply updated SSC parameters into SSC operation, firmware must disable and enable this bit.	1
-	[6:3]	Reserved	0
AUX_FUNC_EN_N	[2]	AUX channel module function enable. 0: Normal operation, 1: Disable AUX channel module.	1

FUNC_EN_2	Bit	Description	Initial State
SERDES_FIFO_FUNC_EN_N	[1]	Serdes FIFO function enable. 0: Normal mode, 1: Disable Serdes FIFO. To reset the serdes fifo, firmware must disable and enable this bit.	1
LS_CLK_DOMAIN_FUNC_EN_N	[0]	Link symbol clock domain modules functions enable. 0: Normal mode, 1: Disable the modules in link symbol clock domain. To reset the modules in link symbol clock domain, firmware must disable and enable this bit.	1

Video Control Register 1 (VIDEO_CTL_1)

Register	Address	Type	Description	Reset Value
VIDEO_CTL_1	Base + 0x0020	R/W	Video Control Register 1	0x0000_0000

VIDEO_CTL_1	Bit	Description	Initial State
-	[31:8]	Reserved	0
VIDEO_EN	[7]	Video data input enable. 0: Disable video data input. 1: Enable video data input, It takes effect at next video frame.	0
VIDEO_MUTE	[6]	Video mute enable. In video mute mode, the solid color, specified in Base + 0x04A8 ~ Base + 0x04B0, is displayed. 0: Disable, 1: Enable. Output video data is changed properly as soon as this bit is configured.	0
-	[5:0]	Reserved	0

Video Control Register 2 (VIDEO_CTL_2)

Register	Address	Type	Description	Reset Value
VIDEO_CTL_2	Base + 0x0024	R/W	Video Control Register 2	0x0000_0010

VIDEO_CTL_2	Bit	Description	Initial State
-	[31:8]	Reserved	0
IN_D_RANGE	[7]	Dynamic range. This bit field is used to specify video data format in main stream attribute data. 1: CEA range (16 ~ 235), 0: VESA range (0 ~ 255).	0
IN_BPC	[6:4]	Video input bit per color/ component (bpc). This bit field is used to specify video data format in main stream attribute data. Note that 6 bpc mode is invalid in YCbCr 422 mode. 100, 101, 110, 111: Reserved, 011: 12 bits, 010: 10 bits, 001: 8 bits, 000: 6 bits.	001
-	[3:2]	Reserved	0

VIDEO_CTL_2	Bit	Description	Initial State
IN_COLOR_F	[1:0]	Colorimetric format of input video. This is used to specify video data format in main stream attribute data. 11: Reserved, 10: YCbCr444, 01: YcbCr422, 00: RGB.	0

Video Control Register 3 (VIDEO_CTL_3)

Register	Address	Type	Description	Reset Value
VIDEO_CTL_3	Base + 0x0028	R/W	Video Control Register 3	0x0000_0000

VIDEO_CTL_3	Bit	Description	Initial State
-	[31:8]	Reserved	0
IN_YC_COEFFI	[7]	YCbCr Coefficients of input video. This is used to specify video data format in main stream attribute data. 1: ITU709. 0: ITU601.	0
-	[6:5]	Reserved	0
VID_CHK_UPDATE_TYPE	[4]	Select video format stability check method in video capture block. 1: Check stability with the difference between adjacent frames. 0: Check stability with the difference of differences between adjacent frames. Compares difference of 1st and 2nd to difference of 3rd and 4th frame.	0
-	[3:0]	Reserved	0

Video Control Register 4 (VIDEO_CTL_4)

Register	Address	Type	Description	Reset Value
VIDEO_CTL_4	Base + 0x002C	R/W	Video Control Register 4	0x0000_0000

VIDEO_CTL_4	Bit	Description	Initial State
-	[31:4]	Reserved	0
BIST_EN	[3]	Video BIST enable. 1: Enable video BIST, 0: Normal operation mode.	0
BIST_WIDTH	[2]	Control display BIST color bar width. 1: Each bar is 64 pixel width, 0: Each bar is 32 pixel width.	0
BIST_TYPE	[1:0]	Display BIST type. 00: Color bar, 01: White, gray and black bar, 10: Mobile white bar, 11: Reserved.	0

Video Control Register 8 (VIDEO_CTL_8)

Register	Address	Type	Description	Reset Value
VIDEO_CTL_8	Base + 0x003C	R/W	Video Control Register 8	0x0000_0020

VIDEO_CTL_8	Bit	Description	Initial State
-	[31:8]	Reserved	0
VID_HRES_TH	[7:4]	Video Frame Horizontal Resolution variation threshold for video capture block. This bit field is used by CAPTURE block to determine whether STRM_VALID should be asserted.	2
VID_VRES_TH	[3:0]	Video Frame Vertical Resolution variation threshold for video capture block. This bit field is used by CAPTURE block to determine whether STRM_VALID should be asserted.	0

Video Control Register 10 (VIDEO_CTL_10)

Register	Address	Type	Description	Reset Value
VIDEO_CTL_10	Base + 0x0044	R/W	Video Control Register 10	0x0000_0000

VIDEO_CTL_10	Bit	Description	Initial State
-	[31:5]	Reserved	0
F_SEL	[4]	Video format select. 1: Video format information from register, 0: Video format information from video_capture module. According to the configuration of this bit field, the values of video format status registers in Base + 0x008C~ 0x00D0 are determined, which are transferred as main stream attribute packet. Note that if BIST_EN is set to 1, F_SEL must be cleared to 0 although video format information comes from registers set by user.	0
-	[3]	Reserved	0
SLAVE_I_SCAN_CFG	[2]	Interlace scan mode configuration. 0: Progressive, 1: Interlace. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0
SLAVE_VSYNC_P_CFG	[1]	Slave mode VSYNC polarity configuration. 1: Low is active, 0: High is active. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0
SLAVE_HSYNC_P_CFG	[0]	Slave mode HSYNC polarity configuration. 1: Low is active, 0: High is active. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Total Line Low Byte Configure Register (TOTAL_LINE_CFG_L)

Register	Address	Type	Description	Reset Value

TOTAL_LINE_CFG_L	Base + 0x0048	R/W	Total Line Byte Configure Register	0x0000_0000

TOTAL_LINE_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
TOTAL_LINE_CFG_L	[7:0]	TOTAL_LINE_CFG is used to specify the number of lines in each frame. This register is TOTAL_LINE_CFG [7:0]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Total Line High Byte Configure Register (TOTAL_LINE_CFG_H)

Register	Address	Type	Description	Reset Value
TOTAL_LINE_CFG_H	Base + 0x004C	R/W	Total Line High Byte Configure Register	0x0000_0000

TOTAL_LINE_CFG_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
TOTAL_LINE_CFG_H	[3:0]	TOTAL_LINE_CFG is used to specify the number of lines in each frame. This register is TOTAL_LINE_CFG [11:8]. When F_SEL is 1, this value is sent in main stream attribute packet. When Video BIST_EN is enabled, this bit must be configured right to generate right video format.	0

Active Line Low Byte Configure Register (ACTIVE_LINE_CFG_L)

Register	Address	Type	Description	Reset Value
ACTIVE_LINE_CFG_L	Base + 0x0050	R/W	Active Line Low Byte Configure Register	0x0000_0000

ACTIVE_LINE_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
ACTIVE_LINE_CFG_L	[7:0]	ACTIVE_LINE_CFG is used to specify the number of active lines in each frame. This register is ACTIVE_LINE_CFG [7:0]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Active Line High Byte Configure Register (ACTIVE_LINE_CFG_H)

Register	Address	Type	Description	Reset Value
ACTIVE_LINE_CFG_H	Base + 0x0054	R/W	Active Line High Byte Configure Register	0x0000_0000

ACTIVE_LINE_CFG_H	Bit	Description	Initial State
-	[31:4]	Reserved	0

ACTIVE_LINE_CFG_H	[3:0]	ACTIVE_LINE_CFG is used to specify the number of active lines in each frame. This register is ACTIVE_LINE_CFG [11:8]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Vertical Front Porch Configure Register (V_F_PORCH_CFG)

Register	Address	Type	Description	Reset Value
V_F_PORCH_CFG	Base + 0x0058	R/W	Vertical Front Porch Configure Register	0x0000_0000

V_F_PORCH_CFG	Bit	Description	Initial State
-	[31:8]	Reserved	0
V_F_PORCH_CFG	[7:0]	This is used to specify the number of lines in vertical front porch part. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Vertical Sync Width Configure Register (V_SYNC_WIDTH_CFG)

Register	Address	Type	Description	Reset Value
V_SYNC_WIDTH_CFG	Base + 0x005C	R/W	Vertical Sync Width Configure Register	0x0000_0000

V_SYNC_WIDTH_CFG	Bit	Description	Initial State
-	[31:8]	Reserved	0
V_SYNC_WIDTH_CFG	[7:0]	This is used to specify the number of lines in VSYNC period. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Vertical Back Porch Configure Register (V_B_PORCH_CFG)

Register	Address	Type	Description	Reset Value
V_B_PORCH_CFG	Base + 0x0060	R/W	Vertical Back Porch Configure Register	0x0000_0000

V_B_PORCH_CFG	Bit	Description	Initial State
-	[31:8]	Reserved	0
V_B_PORCH_CFG	[7:0]	This is used to specify the number of lines in frame back porch part. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Total Pixel Low Byte Configure Register (TOTAL_PIXEL_CFG_L)

Register	Address	Type	Description	Reset Value
TOTAL_PIXEL_CFG_L	Base + 0x0064	R/W	Total Pixel Low Byte Configure Register	0x0000_0000

TOTAL_PIXEL_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
TOTAL_PIXEL_CFG_L	[7:0]	TOTAL_PIXEL_CFG is used to specify the number of pixels in each line. This register is TOTAL_PIXEL_CFG[7:0]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Total Pixel High Byte Configure Register (TOTAL_PIXEL_CFG_H)

Register	Address	Type	Description	Reset Value
TOTAL_PIXEL_CFG_H	Base + 0x0068	R/W	Total Pixel High Byte Configure Register	0x0000_0000

TOTAL_PIXEL_CFG_H	Bit	Description	Initial State
-	[31:6]	Reserved	0
TOTAL_PIXEL_CFG_H	[5:0]	TOTAL_PIXEL_CFG is used to specify the number of pixels in each line. This register is TOTAL_PIXEL_CFG [13:8]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Active Pixel Low Byte Configure Register (ACTIVE_PIXEL_CFG_L)

Register	Address	Type	Description	Reset Value
ACTIVE_PIXEL_CFG_L	Base + 0x006C	R/W	Active Pixel Low Byte Configure Register	0x0000_0000

ACTIVE_PIXEL_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
ACTIVE_PIXEL_CFG_L	[7:0]	ACTIVE_PIXEL_CFG is used to specify the number of active pixels in each line. This register is ACTIVE_PIXEL_CFG [7:0]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Active Pixel High Byte Configure Register (ACTIVE_PIXEL_CFG_H)

Register	Address	Type	Description	Reset Value
ACTIVE_PIXEL_CFG_H	Base + 0x0070	R/W	Active Pixel High Byte Configure Register	0x0000_0000

ACTIVE_PIXEL_CFG_H	Bit	Description	Initial State
-	[31:6]	Reserved	0

ACTIVE_PIXEL_CFG_H	[5:0]	ACTIVE_PIXEL_CFG is used to specify the number of active pixels in each line. This register is ACTIVE_PIXEL_CFG [13:8]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Horizon Front Porch Low Byte Configure Register (H_F_PORCH_CFG_L)

Register	Address	Type	Description	Reset Value
H_F_PORCH_CFG_L	Base + 0x0074	R/W	Horizon Front Porch Low Byte Configure Register	0x0000_0000

H_F_PORCH_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
H_F_PORCH_CFG_L	[7:0]	H_F_PORCH_CFG is used to specify the number of pixels in frame horizon front porch part. This register is H_F_PORCH_CFG[7:0] When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Horizon Front Porch High Byte Configure Register (H_F_PORCH_CFG_H)

Register	Address	Type	Description	Reset Value
H_F_PORCH_CFG_H	Base + 0x0078	R/W	Horizon Front Porch High Byte Configure Register	0x0000_0000

H_F_PORCH_CFG_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
H_F_PORCH_CFG_H	[3:0]	H_F_PORCH_CFG is used to specify the number of pixels in frame horizon front porch part. This register is H_F_PORCH_CFG [11:8] When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Horizon Sync Width Low Byte Configure Register (H_SYNC_CFG_L)

Register	Address	Type	Description	Reset Value
H_SYNC_CFG_L	Base + 0x007C	R/W	Horizon Sync Width Low Byte Configure Register	0x0000_0000

H_SYNC_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
H_SYNC_CFG_L	[7:0]	H_SYNC_CFG is used to specify the number of pixels in HSYNC period. This register is H_SYNC_CFG [7:0]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Horizon Sync Width High Byte Configure Register (H_SYNC_CFG_H)

Register	Address	Type	Description	Reset Value
H_SYNC_CFG_H	Base + 0x0080	R/W	Horizon Sync Width High Byte Configure Register	0x0000_0000

H_SYNC_CFG_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
H_SYNC_CFG_H	[3:0]	H_SYNC_CFG is used to specify the number of pixels in HSYNC period. This register is H_SYNC_CFG [11:8]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Horizon Back Porch Low Byte Configure Register (H_B_PORCH_CFG_L)

Register	Address	Type	Description	Reset Value
H_B_PORCH_CFG_L	Base + 0x0084	R/W	Horizon Back Porch Low Byte Configure Register	0x0000_0000

H_B_PORCH_CFG_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
H_B_PORCH_CFG_L	[7:0]	H_B_PORCH_CFG is used to specify the number of pixel in frame horizon back porch part. This register is H_B_PORCH_CFG [7:0]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Horizon Back Porch High Byte Configure Register (H_B_PORCH_CFG_H)

Register	Address	Type	Description	Reset Value
H_B_PORCH_CFG_H	0x0000_0088	R/W	Horizon Back Porch High Byte Configure Register	0x0000_0000

H_B_PORCH_CFG_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
H_B_PORCH_CFG_H	[3:0]	H_B_PORCH_CFG is used to specify the number of pixel in frame horizon back porch part. This register is H_B_PORCH_CFG [11:8]. When F_SEL is 1, this value is sent in main stream attribute packet. When BIST_EN is 1, this bit field is used to specify the BIST video stream format.	0

Video Status Register (VIDEO_STATUS)

Register	Address	Type	Description	Reset Value
VIDEO_STATUS	Base + 0x008C	RO	Input Video Status Register	0x0000_0003

VIDEO_STATUS	Bit	Description	Initial State
-	[31:4]	Reserved	0

VIDEO_STATUS	Bit	Description	Initial State
FIELD_S	[3]	Interlace scan field status. 1: Second field, 0: First field. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0
I_SCAN_S	[2]	Auto-detect interlace or progressive scan status: 1: Interlace scan, 0: Progressive scan. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0
VSYNC_P_S	[1]	Auto-detect VSYNC polarity: 1: Low is active, 0: High is active. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	1
HSYNC_P_S	[0]	Auto-detect HSYNC polarity: 1: Low is active, 0: High is active. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	1

Total Line Status Low Byte Register (TOTAL_LINE_STA_L)

Register	Address	Type	Description	Reset Value
TOTAL_LINE_STA_L	Base + 0x0090	RO	Total Line Status Low Byte Register	0x0000_0001

TOTAL_LINE_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
TOTAL_LINE_STA_L	[7:0]	TOTAL_LINE [7:0] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	1

Total Line Status High Byte Register (TOTAL_LINE_STA_H)

Register	Address	Type	Description	Reset Value
TOTAL_LINE_STA_H	Base + 0x0094	RO	Total Line Status High Byte Register	0x0000_0000

TOTAL_LINE_STA_H	Bit	Description	Initial State
-	[31:5]	Reserved	0
TOTAL_LINE_STA_H	[4:0]	TOTAL_LINE [11:8] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Active Line Status Low Byte Register (ACTIVE_LINE_STA_L)

Register	Address	Type	Description	Reset Value
ACTIVE_LINE_STA_L	Base + 0x0098	RO	Active Line Status Low Byte Register	0x0000_0000

ACTIVE_LINE_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
ACTIVE_LINE_STA_L	[7:0]	ACTIVE_LINE [7:0] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Active Line Status High Byte Register (ACTIVE_LINE_STA_H)

Register	Address	Type	Description	Reset Value
ACTIVE_LINE_STA_H	Base + 0x009C	RO	Active Line Status High Byte Register	0x0000_0000

ACTIVE_LINE_STA_H	Bit	Description	Initial State
-	[31:5]	Reserved	0
ACTIVE_LINE_STA_H	[4:0]	ACTIVE_LINE [11:8] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Vertical Front Porch Status Register (V_F_PORCH_STA)

Register	Address	Type	Description	Reset Value
V_F_PORCH_STA	Base + 0x00A0	RO	Vertical Front Porch Status Register	0x0000_0001

V_F_PORCH_STA	Bit	Description	Initial State
-	[31:8]	Reserved	0
V_F_PORCH_STA	[7:0]	V_F_PORCH (vertical front porch) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	1

Vertical Sync Width Status Register (V_SYNC_STA)

Register	Address	Type	Description	Reset Value
V_SYNC_STA	Base + 0x00A4	RO	Vertical Sync Width Status Register	0x0000_0000

V_SYNC_STA	Bit	Description	Initial State
-	[31:8]	Reserved	0
V_SYNC_STA	[7:0]	V_SYNC_WIDTH (vertical sync width) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Vertical Back Porch Status Register (V_B_PORCH_STA)

Register	Address	Type	Description	Reset Value
V_B_PORCH_STA	Base + 0x00A8	RO	Vertical Back Porch Status Register	0x0000_0000

V_B_PORCH_STA	Bit	Description	Initial State
-	[31:8]	Reserved	0
V_B_PORCH_STA	[7:0]	V_B_PORCH (vertical back porch) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Total Pixel Status Low Byte Register (TOTAL_PIXEL_STA_L)

Register	Address	Type	Description	Reset Value
TOTAL_PIXEL_STA_L	Base + 0x00Ac	RO	Total Pixel Status Low Byte Register	0x0000_0000

TOTAL_PIXEL_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
TOTAL_PIXEL_STA_L	[7:0]	TOTAL_PIXEL [7:0] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Total Pixel Status High Byte Register (TOTAL_PIXEL_STA_H)

Register	Address	Type	Description	Reset Value
TOTAL_PIXEL_STA_H	Base + 0x00B0	RO	Total Pixel Status High Byte Register	0x0000_0000

TOTAL_PIXEL_STA_H	Bit	Description	Initial State
-	[31:6]	Reserved	0
TOTAL_PIXEL_STA_H	[5:0]	TOTAL_PIXEL [13:8] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Active Pixel Status Low Byte Register (ACTIVE_PIXEL_STA_L)

Register	Address	Type	Description	Reset Value
ACTIVE_PIXEL_STA_L	Base + 0x00B4	RO	Active Pixel Status Low Byte Register	0x0000_0000

ACTIVE_PIXEL_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0

ACTIVE_PIXEL_STA_L	[7:0]	ACTIVE_PIXEL [7:0] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Active Pixel Status High Byte Register (ACTIVE_PIXEL_STA_H)

Register	Address	Type	Description	Reset Value
ACTIVE_PIXEL_STA_H	Base + 0x00B8	RO	Active Pixel Status High Byte Register	0x0000_0000

ACTIVE_PIXEL_STA_H	Bit	Description	Initial State
-	[31:6]	Reserved	0
ACTIVE_PIXEL_STA_H	[5:0]	ACTIVE_PIXEL [13:8] which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Horizon Front Porch Status Low Byte Register (H_F_PORCH_STA_L)

Register	Address	Type	Description	Reset Value
H_F_PORCH_STA_L	Base + 0x00BC	RO	Horizon Front Porch Status Low Byte Register	0x0000_0000

H_F_PORCH_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
H_F_PORCH_STA_L	[7:0]	H_F_PORCH [7:0] (horizon front porch) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Horizon Front Porch Status High Byte Register (H_F_PORCH_STA_H)

Register	Address	Type	Description	Reset Value
H_F_PORCH_STA_H	Base + 0x00C0	RO	Horizon Front Porch Status High Byte Register	0x0000_0000

H_F_PORCH_STA_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
H_F_PORCH_STA_H	[3:0]	H_F_PORCH [11:8] (horizon front porch) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Horizon Sync Width Status Low Byte Register (H_SYNC_STA_L)

Register	Address	Type	Description	Reset Value
H_SYNC_STA_L	Base + 0x00C4	RO	Horizon Sync Width Status Low Byte Register	0x0000_0000

H_SYNC_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
H_SYNC_STA_L	[7:0]	H_SYNC [7:0] (horizon sync width) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Horizon Sync Width Status High Byte Register (H_SYNC_STA_H)

Register	Address	Type	Description	Reset Value
H_SYNC_STA_H	Base + 0x00C8	RO	Horizon Sync Width Status High Byte Register	0x0000_0000

H_SYNC_STA_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
H_SYNC_STA_H	[3:0]	H_SYNC [11:8] (horizon sync width) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Horizon Back Porch Status Low Byte Register (H_B_PORCH_STA_L)

Register	Address	Type	Description	Reset Value
H_B_PORCH_STA_L	Base + 0x00CC	RO	Horizon Back Porch Status Low Byte Register	0x0000_0000

H_B_PORCH_STA_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
H_B_PORCH_STA_L	[7:0]	H_B_PORCH [7:0] (horizon back porch) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

Horizon Back Porch Status High Byte Register (H_B_PORCH_STA_H)

Register	Address	Type	Description	Reset Value
H_B_PORCH_STA_H	Base + 0x00D0	RO	Horizon Back Porch Status High Byte Register	0x0000_0000

H_B_PORCH_STA_H	Bit	Description	Initial State
-	[31:4]	Reserved	0
H_B_PORCH_STA_H	[3:0]	H_B_PORCH [11:8] (horizon back porch) which is detected by video capture module. This bit field is valid only when STRM_VALID is high. And STRM_VALID becomes high when two successive frames are determined as stable.	0

PLL control Register_1(PLL_REG_1)

Register	Address	Type	Description	Reset Value
PLL_REG_1	Base + 0x00FC	R/W	PLL control 1	0x0000_0011

PLL_REG_1	Bit	Description	Initial State
-	[31:6]	Reserved	0

LINK_SPEED(RO)	[5:4]	FVCO: 00:1.62G 01:2.7G 1x:Reserved	1
-	[3:1]	Reserved	0
PLL_REF_CLK_FREQ	[0]	reference CLOCK frequency: 1(default):24MHz 0:27MHz	0

PLL control Register_2(PLL_REG_2)

Register	Address	Type	Description	Reset Value
PLL_REG_2	**Base + 0x09E4**	**R/W**	**Pll_control_2**	**0x0000_0011**

PLL_REG_2	Bit	Description	Initial State
-	[31:8]	Reserved	0
LDO_OUTPUT_V_SEL	[7:6]	1.5v LDO output voltage select 00:1.35v 01:1.40v 10:1.45v(default) 11:1.50v	2'b10
KVCO	[5:4]	KVCO to control VCO band 00: decrease KVCO by 15% 01:(default) 10:increase KVCO by 10% 11:increase KVCO by 20%	2'b01
CHG_PUMP_CURRENT_SEL	[3:2]	charge pump current select 00:2.5u 01(default):5u 10:7.5u 11:10u	2'b01
V2I_CURRENT_SEL	[1:0]	v2i current select 00: no adding current 01: adding 1mA current(default) 10: adding 2mA current 11: adding 4.5mA current	2'b01

PLL control Register_3(PLL_REG_3)

Register	Address	Type	Description	Reset Value
PLL_REG_3	Base + 0x09E8	R/W	Pll_control_3	0x0000_002B

PLL_REG_3	Bit	Description	Initial State
-	[31:7]	Reserved	0
LOCK_DET_CNT_SEL	[6:5]	lock detector output counter select, counter period is twice of reference clock 00: 64 cycle 01: 128 cycle 10: 256 cycle (default) 11: 512 cycle	2'b10
LOOP_FILTER_RESET_SEL	[4]	loop filter control voltage reset select 1: reset to the value below DVDD 0: reset to DVDD (default)	0
PALL_SSC_RESET	[3]	PLL and ssc reset control 1: reset 0: normal	0

PLL_REG_3	Bit	Description	Initial State
LOCK_DET_BYPASS	[2]	lock detector bypass select 0: not bypass (default) 1: bypass lock detector in ssc	0
PLL_LOCK_DET_MODE	[1]	PLL lock detector mode select 0: fractional N (default) 1: integer N	0
PLL_LOCK_DET_FORCE	[0]	force PLL lock detector lock 0: not force lock (default) 1: force PLL lock	0

PLL control Register_4(PLL_REG_4)

Register	Address	Type	Description	Reset Value
PLL_REG_4	Base + 0x09EC	R/W	Pll_control_4	0x0000_0023

PLL_REG_4	Bit	Description	Initial State
-	[31:8]	Reserved	0
-	[7:0]	Reserved	0

PLL control Register_5(PLL_REG_5)

Register	Address	Type	Description	Reset Value
PLL_REG_5	Base + 0x0A00	R/W	Pll_control_5	0x0000_0000

PLL_REG_5	Bit	Description	Initial State
-	[31:7]	Reserved	0
REGULATOR_V_SEL	[6:4]	slave regulator output voltage select 000900V 0010.925V 0100.950V(default) 0110.975V 1001.000V 1011.025V 1101.050V	3'b010
STANDBY_CURRENT_SEL	[3]	slave standby current select 1: adding 200uA standby current 0: keep 300uA standby current (default)	0
CHG_PUMP_INPUT_CTRL	[2:1]	control charge pump input voltage for 0.95V master regulator 00: 1.1V 01: 1.2V(default) 10: 1.3V 11: 1.4V	2'b01
CHG_PUMP_INPUT_CTRL_OP	[0]	option to control charge pump input voltage for 0.95V master regulator 0: set by pll_reg5<2:1>(default) 1: 1.8V	0

PLL control Register_mac(PLL_REG_mac)

Register	Address	Type	Description	Reset Value
PLL_REG_MAC	Base + 0x0A04	R/W	Pll_control_mac	0x0000_0000

PLL_REG_MAC	Bit	Description	Initial State
-	[31:8]	Reserved	0
ANALOG_BACKUP1	[7:0]	Reserved	0

Freq Register (FREQ_IN_REG)

Register	Address	Type	Description	Reset Value
FREQ_IN_REG	Base + 0x0A10	R/W	frequency set from register for freq counter	0x0000_0080

FREQ_IN_REG	Bit	Description	Initial State
-	[31:8]	Reserved	0
FREQ_REG	[7:0]	frequency set from register for freq counter	80

Freq Register (P_REG_FRQ)

Register	Address	Type	Description	Reset Value
P_REG_FRQ	Base + 0x0A14	RO	digital output for debug	0x0000_0000

P_REG_FRQ	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRQ	[7:0]	digital output for debug, controlled by pll_reg1<7> and pll_reg4<4> when pll_reg1<7>=0 & pll_reg4<4>=0: half video clock frequency calculated by frequency counter When pll_reg1<7>=0 & pll_reg4<4>=1: half video clock frequency calculated frq_vid_ck_in<8:0> when pll_reg1<7>=1: frq<1:0>: <n_over, n_under> frq<3:2>: band<1:0> frq<7:4>: 0	00

Freq Register (P_REG_FRQ_COUNT_RDY)

Register	Address	Type	Description	Reset Value
P_REG_FRQ_COUNT_RDY	Base + 0x0A18	RO	frequency counter ready indicator (frequency counter for VCO band selection)	0x0000_0000

P_REG_FRQ_COUNT_RDY	Bit	Description	Initial State
-	[31:1]	Reserved	0
FRQ_COUNT_RDY	[0]	frequency counter ready indicator (frequency counter for VCO band selection) 1: frequency counter ready, its output is the real value of video PLL 0: frequency counter not ready, its output is not the real value	0

Freq Register (P_BAND_DEC_RESET)

Register	Address	Type	Description	Reset Value

P_BAND_DEC_RESET	Base + 0x0A1C	WO	band decoder reset	0x0000_0000

P_BAND_DEC_RESET	Bit	Description	Initial State
-	[31:1]	Reserved	0
R_BAND_DEC_RESET	[0]	1: reset band decoder 0: band decoder works	0

SSC control Register_2(SSC_REG)

Register	Address	Type	Description	Reset Value
SSC_REG	Base + 0x0104	R/W	SSC control	0x0000_000A

SSC_REG	Bit	Description	Initial State
-	[31:8]	Reserved	0
SSC_OFFSET	[7:6]	00: no 01: up 100ppm 10: down 100ppm 11: down 200ppm	0
SSC_MODE	[5:4]	00:disable 01:down spread 10:center spread 11:up spread	2'b01
SSC_DEPTH	[3:0]	0000:disable 0001:500ppm 0010:1000ppm 0011:1500ppm 0100:2000ppm 0101:2500ppm 0110:3000ppm 0111:3500ppm 1000:4000ppm 1001:4500ppm 1010:5000ppm 1011:5500ppm 1100-1111:6000ppm	4'h9

TX_COMMON Register (TX_COMMON)

Register	Address	Type	Description	Reset Value
TX_REG_COMMON	Base + 0x0114	R/W	TX COMMON 1 register	0x0000_003A

TX_COMMON	Bit	Description	Initial State
-	[31:8]	Reserved	0
TX_SWING_PRE_EMP_MODE_SEL	[7]	TX swing and pre emphasis control mode selection 1: TX swing and pre emphasis control by register dp_reserv2<7:0> 0: TX swing and pre emphasis control by register chx_reg_swing<7:0> and chx_reg_pre<7:0>	1
PRE_DRIVER_PW_CTRL1	[6:5]	Pre-driver extra power control 0: disable 1: enable	0

LP_MODE_CLK_REGULATOR	[4]	Low power mode control for clock regulator 0:low power mode 1:high power mode	0
RESISTOR_MSB_CTRL	[3]	TX terminal resistor MSB control	0
RESISTOR_CTRL	[2:0]	TX terminal resistor control when tx_common<3>=0 000: 58.54 . . . . . . 011:54.6 . . . . . . 111:50 when tx_common<6>=1 000: 49 . . . . . . 011:46 . . . . . . 111:42.6	3'h7

TX_COMMON2 Register (TX_COMMON2)

Register	Address	Type	Description	Reset Value
TX_COMMON2	Base + 0x0118	R/W	TX_COMMON2	0x0000_0050

TX_COMMON2	Bit	Description	Initial State
-	[31:8]	Reserved	0
TX_OUTPUT_PN_INVERSE_CH3	[7]	TX ch3 output p-n inverse control: 0: not inverse 1: output p and n inverse	0
TX_OUTPUT_PN_INVERSE_CH2	[6]	TX ch2 output p-n inverse control: 0: not inverse 1: output p and n inverse	1
TX_OUTPUT_PN_INVERSE_CH1	[5]	TX ch1 output p-n inverse control: 0: not inverse 1: output p and n inverse	0
TX_OUTPUT_PN_INVERSE_CH0	[4]	TX ch0 output p-n inverse control: 0: not inverse 1: output p and n inverse	1
TX_OUT_PATTERN_EN	[3]	TX output pattern enable 0: normal TX 1: dedicate pattern	0
TX_DATA_PATTEN	[2:0]	TX data Patten 000:all zero 001:all one 010:D10.2 011:1100 100:K28.5 101:K28.7 110:1111100000 111:11111111110000000000	0

TX_COMMON3 Register (TX_COMMON3)

Register	Address	Type	Description	Reset Value
TX_COMMON3	Base + 0x0A08	R/W	TX_COMMON3	0x0000_0000

TX_COMMON3	Bit	Description	Initial State
-	[31:8]	Reserved	0
CLK_DLY_SEL	[7:3]	Select /20 clock delay (clk_div2_ssc & tx_txd_clk) 00000 : delay=150ps 00001 : delay=150ps+1*70ps 00010 : delay=150ps+2*70ps 00011: delay=150ps+3*70ps	0
CLK_INVERSE_EN	[2]	TX input clock inverse enable 0: normal 1: TX input clock inverse	0
SCAN_CLK_SEL	[1]	ch0 select i_ref_clk_24m for scan 0:select tx_bscan_data<0> 1:select i_ref_clk_24m	0
-	[0]	Reserved	0

DP_AUX Register (DP_AUX)

Register	Address	Type	Description	Reset Value
DP_AUX	Base + 0x0120	R/W	Aux control	0x0000_0007

DP_AUX	Bit	Description	Initial State
-	[31:5]	Reserved	0
DP_AUX_COMMON_MODE	[4]	AUX RX CM voltage control 0: AUX CH use VCC1/2 as CM voltage (have static current consumption) 1: use VCC1 as CM voltage	0
DP_AUX_EN(R/O)	[3]	AUX TX enable 0: AUX CH configured as RX 1: AUX CH configured as TX	0
-	[2]	Reserved	0
AUX_TERM	[1:0]	AUX CH impedance control bits: only control TX impedance 00 : 500ohm 01 : 250ohm 10 : 100ohm 11 : 50ohm	3

DP_BIAS Register (DP_BIAS)

Register	Address	Type	Description	Reset Value
DP_BIAS	Base + 0x0124	R/W	BIAS control	0x0000_0034

DP_BIAS	Bit	Description	Initial State
-	[31:7]	Reserved	0
DP_BG_OUT_SEL	[6:4]	Select band gap out 000:0.56V 100:0.6V(default) 111:0.63V	4
DP_DB_CUREENT_CTRL	[3]	Band gap start up current control 0: balance 1: unbalance	0

DP_BG_SEL	[2]	Select band gap 0:sel Register 1:sel Band gap	1
DP_RESISTOR_TUNE_BG_CTRL	[1:0]	Resistor tune for band gap TC control 00 : 25uV/℃ 01: 10uV/℃ 10: -10uV/℃ 11: -25uV/℃	0

DP_TEST Register (DP_TEST)

Register	Address	Type	Description	Reset Value
DP_TEST	Base + 0x0128	R/W	TEST	0x0000_0000

DP_TEST	Bit	Description	Initial State
-	[31:8]	Reserved	0
DP_TEST_MODE	[7:6]	00&01: test disables dc_tp/atesto/dtesto output hiz. 10: atesto test enable 11: dtesto test enable	0
	[5:3]	When <7:6> ==10, test analog blocks: 000-- disable analog test 001: enable ch0 analog test mux <1:0>(00/01/10/11)-- ->(avdd10_pre_drv,avdd10_p2s, vddu_p2s,avss) 010: enable ch1 analog test mux <1:0>(00/01/10/11)-- ->(avdd10_pre_drv,avdd10_p2s, vddu_p2s,avss) 011: enable ch2 analog test mux <1:0>(00/01/10/11)-- ->(avdd10_pre_drv,avdd10_p2s, vddu_p2s,avss) 100: enable ch3 analog test mux <1:0>(00/01/10/11)-- ->(avdd10_pre_drv,avdd10_p2s, vddu_p2s,avss) 101: enable pll analog test mux <1:0>(00/01/10/11)--->(vdd10_cln,v1p45_v2i, avdd18,vco_ctrl) 110: enable charge pump regulator analog test mux <1:0> (00/01/10/11)---->(v1v_regu, vregu_out,v0.5_ref,null) 111: test band gap output When<7:6>==11, test digital blocks: 000-- disable digital test 001: enable pll digital test mux <1:0>--(00/01/10/11)--->pll_ref,vco_fb, vss,vss. 010: test charge pump regulator OSC clock.	0
-	[2:0]	Reserved	0

DP_PD Register (DP_PD)

Register	Address	Type	Description	Reset Value
DP_PD	Base + 0x012C	R/W	Power down. Power down sequence: dp_pd=ff => dp_pd=7f => wait 10us => dp_pd=00	0x0000_00FF

DP_PD	Bit	Description	Initial State
-	[31:8]	Reserved	0
PD_INC_BG	[7]	Power down all including band gap	1
PD_EXP_BG	[6]	Power down all except band gap	1
PD_AUX_CH	[5]	Power down AUX channel	1
PD_PLL	[4]	Power down PLL	1
PD_CH3	[3]	Power down ch3	1
PD_CH2	[2]	Power down ch2	1
PD_CH1	[1]	Power down ch1	1
PD_CH0	[0]	Power down ch0	1

DP_RESERV1 Register (DP_RESERV1)

Register	Address	Type	Description	Reset Value
RESERVE1	Base + 0x0130	R/W	ATE Test enable	0x0000_0000

DP_RESERV1	Bit	Description	Initial State
-	[31:4]	Reserved	0
SSC_MODE_LOCK	[7]	SSC mode lock	0
-	[6]	Reserved	0
PRE_DRIVER_PW_CTRL2	[5:4]	Pre-driver extra power control 0: disable 1:enable	0
ATE_EN_CH3	[3]	Set 1 to enable CH3 ATE test	0
ATE_EN_CH2	[2]	Set 1 to enable CH2 ATE test	0
ATE_EN_CH1	[1]	Set 1 to enable CH1 ATE test	0
ATE_EN_CH0	[0]	Set 1 to enable CH0 ATE test	0

DP_RESERV2 Register (DP_RESERV2)

Register	Address	Type	Description	Reset Value
DP_RESERV2	Base + 0x0134	R/W	RESERVED	0x0000_0000

DP_RESERV2	Bit	Description	Initial State
-	[31:8]	Reserved	0
CH1_CH3_SWING_EMP_CTRL	[7:4]	ch1,3 swing and pre emphasis control for firmware when tx_common<7>=1 0000 : swing0 pre emphasis 0 dB 0001 : swing1 pre emphasis 0 dB 0010 : swing2 pre emphasis 0 dB 0011 : swing3 pre emphasis 0 dB 0100 : swing0 pre emphasis 3.5 dB 0101 : swing1 pre emphasis 3.5 dB 0110 : swing2 pre emphasis 3.5 dB 1000 : swing0 pre emphasis 6 dB 1001 : swing1 pre emphasis 6 dB 1100 : swing0 pre emphasis 9.5 dB others : swing0 pre emphasis 9.5 dB	5

DP_RESERV2	Bit	Description	Initial State
CH0_CH2_SWING_EMP_CTRL	[3:0]	ch0,2 swing and pre emphasis control for firmware when tx_common<7>=1 0000 : swing0 pre emphasis 0 dB 0001 : swing1 pre emphasis 0 dB 0010 : swing2 pre emphasis 0 dB 0011 : swing3 pre emphasis 0 dB 0100 : swing0 pre emphasis 3.5 dB 0101 : swing1 pre emphasis 3.5 dB 0110 : swing2 pre emphasis 3.5 dB 1000 : swing0 pre emphasis 6 dB 1001 : swing1 pre emphasis 6 dB 1100 : swing0 pre emphasis 9.5 dB others : swing0 pre emphasis 9.5 dB	5

AVI InfoFrame Packet Data Byte (AVI_DB1 ~ AVI_DB13)

Register	Address	Type	Description	Reset Value
AVI_DB1 ~ AVI_DB13	Base + 0x01D0~ Base + 0x0200	R/W	AVI InfoFrame Packet Data Byte	0x0000_0000

AVI_DB1 ~ AVI_DB13	Bit	Description	Initial State
-	[31:8]	Reserved	0
AVI_DB1~ AVI_DB13	[7:0]	AVI Data Byte 1 ~ 13	0

InfoFrame Packet Type Code (IF_TYPE)

Register	Address	Type	Description	Reset Value
IF_TYPE	Base + 0x0244	R/W	InfoFrame Packet Type Code.	0x0000_0000

IF_TYPE	Bit	Description	Initial State
-	[31:8]	Reserved	0
IF_TYPE	[7:0]	InfoFrame Packet Type Code. It can be set as (0x80 + InfoFrame Type Code) and send any type of infoframe defined in CEA-861C. Commonly, we set it as 0x83(0x80 + 0x03, 0x03 is the type code of SPD InfoFrame) and send SPD infoframe.	0

InfoFrame Packet Data Byte (IF_PKT_DB1~25)

Register	Address	Type	Description	Reset Value
IF_PKT_DB1 ~ IF_PKT_DB25	Base + 0x0254 ~ Base + 0x02B4	R/W	InfoFrame Packet Data Byte	0x0000_0000

IF_PKT_DB1 ~ IF_PKT_DB25	Bit	Description	Initial State
-	[31:8]	Reserved	0
IF_PKT_DB1 ~ IF_PKT_DB25	[7:0]	InfoFrame Packet Data Byte 1 ~ 25. The registers define the data in the InfoFrame and the InfoFrame type is defined by IF_TYPE.	0

MPEG Source InfoFrame Packet Data Byte (MPEG_DB1 ~ MPEG_DB10)

Register	Address	Type	Description	Reset Value
MPEG_DB1 ~ MPEG_DB10	Base + 0x02D0 ~ Base + 0x02F4	R/W	MPEG Source InfoFrame Packet Data Byte	0x0000_0000

MPEG_DB1 ~ MPEG_DB10	Bit	Description	Initial State
-	[31:8]	Reserved	0
MPEG_DB1 ~ MPEG_DB10	[7:0]	MPEG InfoFrame Data Byte 1 ~ 10	0

PSR Frame Update Control Register

Register	Address	Type	Description	Reset Value
PSR_FRAME_UPDATA_CTRL	Base + 0x0318	R/W	PSR frame update control	0x0000_0000

PSR_FRAME_UPDATA_CTRL	Bit	Description	Initial State
-	[31:2]	Reserved	0
PSR_FRAME_UP_TYPE	[1]	Select PSR Frame Update type. 1 = Burst single frame update. 0 = Single frame update. VSC packet is only sent once after PSR_FRAME_UPDATE is written with 1. IF_EN bit will be self-cleared after the VSYNC leading edge.	0
PSR_VSC_PACKET_VERSION	[0]	PSR VSC packet version select. 1 = PSR 2. 0 = PSR 1.	0

VSC Shadow Data Bytes Register

Register	Address	Type	Description	Reset Value
VSC_SHADOW_DB0~ VSC_SHADOW_DB7	Base+ 0x031C ~ Base+ 0x0338	R/W	VSC shadow data bytes 0 ~ 7	0x0000_0000

VSC_SHADOW_DATA_BYTES	Bit	Description	Initial State
-	[31:8]	Reserved	0
VSC_SHADOW_DB0~ VSC_SHADOW_DB7	[7:0]	VSC shadow data bytes 0 ~ 7	0

VSC Shadow Parity Bytes Register

Register	Address	Type	Description	Reset Value
VSC_SHADOW_PB0~ VSC_SHADOW_PB1	Base+ 0x033C ~ Base+ 0x0340	R/W	VSC shadow parity bytes 0 ~ 1	0x0000_0000

VSC_SHADOW_PARITY_BYTES	Bit	Description	Initial State
-	[31:8]	Reserved	0

VSC_SHADOW_PB0~ VSC_SHADOW_PB1	[7:0]	VSC shadow parity bytes 0 ~ 1	0

Lane Map Register (LANE_MAP)

Register	Address	Type	Description	Reset Value
LANE_MAP	Base + 0x035C	R/W	Lane Map Register	0x0000_00E4

LANE_MAP	Bit	Description	Initial State
-	[31:8]	Reserved	0
LANE3_MAP	[7:6]	Control physical lane 3 will map to which logic lane: 0x11: Logic lane 3, 0x10: Logic lane 2, 0x01: Logic lane 1, 0x00: Logic lane 0,	3
LANE2_MAP	[5:4]	Control physical lane 2 will map to which logic lane: 0x11: Logic lane 3, 0x10: Logic lane 2, 0x01: Logic lane 1, 0x00: Logic lane 0,	2
LANE1_MAP	[3:2]	Control physical lane 1 will map to which logic lane: 0x11: Logic lane 3, 0x10: Logic lane 2, 0x01: Logic lane 1, 0x00: Logic lane 0,	1
LANE0_MAP	[1:0]	Control physical lane 0 will map to which logic lane: 0x11: Logic lane 3, 0x10: Logic lane 2, 0x01: Logic lane 1, 0x00: Logic lane 0,	0

Analog Control Register 2 (ANALOG_CTL_2)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_2	Base + 0x0374	R/W	Analog Control Register 2	0x0000_0008

ANALOG_CTL_2	Bit	Description	Initial State
-	[31:4]	Reserved	0
SEL_24M	[3]	Choose the reference clock of PHY use 24M or 27M: 1: Use 24M clock, 0: Use 27M clock.	1

Hidden Register (INT_STATE_0)

Register	Address	Type	Description	Reset Value
INT_STATE_0	Base + 0x0390	R/W	Hidden Register for debug	0x0000_0003

INT_STATE_0	Bit	Description	Initial State
-	[31:12]	Reserved	0
AUX_CH_DATA_IN(RO)	[11]	AUX received data for debug when AUX_CH_TEST_MODE = 1 and AUX_CH_EN_TEST=0 This bit is read only	0

INT_STATE_0	Bit	Description	Initial State
AUX_SEND_0_1_EN	[10]	1: Force 0/1 toggle in AUX CH when AUX_CH_TEST_MODE = 0 0: normal AUX data transmitting in AUX CH when AUX_CH_TEST_MODE = 0	0
AUX_CH_TEST_MODE	[9]	1: AUX CH is in test mode. 0: AUX CH is in normal mode.	0
AUX_CH_T_TEST	[8]	AUX transmitted data when AUX_CH_TEST_MODE = 1	0
AUX_CH_EN_TEST	[7]	AUX TX enable when AUX_CH_TEST_MODE = 1. 0: disable 1: enable	0
M_VID_DEBUG_EN	[6]	Enable M_VID debugging	0
BIST_YCBCR422_CRL	[5]	For YCbCr422 BIST control	0
AUX_TC	[4:3]	AUX TC Register	0
AUX_RETRY_TIMER	[2:0]	AUX Retry Timer Register	3

Interrupt Status Register (INT_STATE_1)

Register	Address	Type	Description	Reset Value
INT_STATE_1	Base + 0x03C0	RO	Interrupt Status Register	0x0000_0000

INT_STATE_1	Bit	Description	Initial State
-	[31:1]	Reserved	0
INT_STATE	[0]	Interrupt request status 1: Interrupt service is requested, 0: No interrupt service is requested.	0

Common Interrupt Status Register 1 (COMMON_INT_STA_1)

Register	Address	Type	Description	Reset Value
COMMON_INT_STA_1	Base + 0x03C4	R/W C1	Common Interrupt Status Register 1	0x0000_0000

COMMON_INT_STA_1	Bit	Description	Initial State
-	[31:8]	Reserved	0
VSYNC_DET	[7]	1: VSYNC active edge has been detected. Write 1 to clear.	0
PLL_LOCK_CHG	[6]	1: PLL lock state is changed. Write 1 to clear. Check PLL_LOCK of register DP_DEBUG_CTL for PLL lock status.	0
-	[5:0]	Reserved	0
VID_FORMAT_CHG	[3]	1: Video input format change is detected. Write 1 to clear.	0
PSR_VID_CRC_VALID	[2]	1: PSR video CRC value is valid.	0
VID_CLK_CHG	[1]	1: Video input clock change is detected.	0
SW_INT	[0]	1: Software-induced interrupt. Write 1 to clear.	0

Note: All of interrupt status bits are edge triggered.
Note: Interrupt status bits are set regardless of the value of the corresponding interrupt mask bits. But if a mask bit is set, then the corresponding interrupt status is not routed to INT_STATE, which is connected to the system.

Common Interrupt Status Register 3 (COMMON_INT_STA_3)

Register	Address	Type	Description	Reset Value
COMMON_INT_STA_3	Base + 0x03CC	R/W C1	Common Interrupt Status Register 3	0x0000_0000

COMMON_INT_STA_3	Bit	Description	Initial State
-	[31:5]	Reserved	0
DPCD_SPECIFIC_IRQ	[4]	1: Sink specific interrupt in DPCD is detected. Write 1 to clear	0
MYDP_PLUG_IN	[3]	1: MYDP plug out event is detected. Write 1 to clear	0
MYDP_PLUG_OUT	[2]	1: MYDP plug out event is detected. Write 1 to clear	0
MYDP_HPD_IRQ	[1]	1: MYDP HPD interrupt is detected. Write 1 to clear	0
-	[0]	Reserved	0

Common Interrupt Status Register 4 (COMMON_INT_STA_4)

Register	Address	Type	Description	Reset Value
COMMON_INT_STA_4	Base + 0x03D0	R/W C1	Common Interrupt Status Register 4	0x0000_0000

COMMON_INT_STA_4	Bit	Description	Initial State
-	[31:3]	Reserved	0
HOTPLUG_CHG	[2]	1: Hot plug change detected. Write 1 to clear. HOTPLUG_CHG happens whenever the pin I_DP_HDP changes and the change remains for at least hot plug deglitch time. And the hot plug deglitch time is defined in HPD_DEGLITCH_L and HPD_DEGLITCH_H. When HOTPLUG_CHG is high, software shall check the status of HPD signal on register HPD_STATUS.	0
HPD_LOST	[1]	Hot plug detect signal lost timer larger than 2ms, that means cable is plugged out: 1: Interrupt assert, 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0
PLUG	[0]	Hot plug detect signal lost time is larger than 2ms before cable plugged, it means cable is plugged in: 1: Interrupt assert, 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0

DisplayPort Interrupt Status Register (DP_INT_STA)

Register	Address	Type	Description	Reset Value
DP_INT_STA	Base + 0x03DC	R/W C1	DisplayPort Interrupt Status Register	0x0000_0000

DP_INT_STA	Bit	Description	Initial State
-	[31:7]	Reserved	0

INT_HPD	[6]	IRQ (HPD de-asserted less than 2ms) detect interrupt: 1: IRQ interrupt assert, 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0
HW_TRAINING_FINISH	[5]	Training FSM module finish link training procedure: 1: Hardware link training finished, 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0
-	[4]	Reserved	0
SINK_LOST	[3]	Sink lost interrupt 1: Sink lost occurred 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0
LINK_LOST	[2]	Link lost interrupt 1: Link lost occurred 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0
RPLY_RECEIV	[1]	AUX channel command reply is received: 1: Interrupt assert, 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0
AUX_ERR	[0]	AUX channel access error interrupt: 1: Interrupt assert, 0: Not interrupt occurred Write 1 to this bit to clear this interrupt source.	0

Interrupt Mask Register (COMMON_INT_MASK_1)

Register	Address	Type	Description	Reset Value
COMMON_INT_MASK_1	Base + 0x03E0	R/W	Interrupt Mask Register	0x0000_0000

COMMON_INT_MASK_1	Bit	Description	Initial State
-	[31:8]	Reserved	0
COMMON_INT_MASK_1[7:0]	[7:0]	Each bit corresponds to the same bit in Common Interrupt Status Register 1. 0: Mask interrupt 1: Enable interrupt	0

Interrupt Mask Register (COMMON_INT_MASK_3)

Register	Address	Type	Description	Reset Value
COMMON_INT_MASK_3	Base + 0x03E8	R/W	Interrupt Mask Register	0x0000_0000

COMMON_INT_MASK_3	Bit	Description	Initial State
-	[31:5]	Reserved	0

COMMON_INT_MASK_3[4:1]	[4:1]	Each bit corresponds to the same bit in Common Interrupt Status Register 3. 0: Mask interrupt 1: Enable interrupt	0
-	[0]	Reserved	

Interrupt Mask Register (COMMON_INT_MASK_4)

Register	Address	Type	Description	Reset Value
COMMON_INT_MASK_4	Base + 0x03EC	R/W	Interrupt Mask Register	0x0000_0000

COMMON_INT_MASK_4	Bit	Description	Initial State
-	[31:3]	Reserved	0
COMMON_INT_MASK_4[2:0]	[2:0]	Each bit corresponds to the same bit in Common Interrupt Status Register 3. 0: Mask interrupt 1: Enable interrupt	0

DP Interrupt Mask Register (DP_INT_STA_MASK)

Register	Address	Type	Description	Reset Value
DP_INT_STA_MASK	Base + 0x03F8	R/W	Interrupt enable Register	0x0000_0000

DP_INT_STA_MASK	Bit	Description	Initial State
-	[31:7]	Reserved	0
DP_INT_STA_MASK	[6:0]	Each bit corresponds to the same bit in DisplayPort Interrupt Status Register (DP_INT_STA). 1: Enable interrupt. 0: Mask interrupt.	0

Interrupt Control Register (INT_CTL)

Register	Address	Type	Description	Reset Value
INT_CTL	Base + 0x03FC	R/W	Interrupt Control Register	0x0000_0001

INT_CTL	Bit	Description	Initial State
-	[31:6]	Reserved	0
SERDES_OVERFLOW_CLEAR	[5]	1: clear SerDes FIFO overflow flag	0
SERDES_UNDERFLOW_CLEAR	[4]	1: clear SerDes FIFO underflow flag	0
-	[3]	Reserved	0
SOFT_INT_CTRL	[2]	Set Software Interrupt: 1: Set interrupt, 0: Do not set interrupt,	0
-	[1]	Reserved	0
INT_POL	[0]	INT pin assertion polarity: 1: Assert high, 0: Assert low	1

Register Definition for DisplayPort Function

System Control Register #1 (SYS_CTL_1)

Register	Address	Type	Description	Reset Value

SYS_CTL_1	Base + 0x0600	R/W	System Control Register #1.	0x0000_0000

SYS_CTL_1	Bit	Description	Initial State
-	[31:5]	Reserved	0
HBR2_EYE_SY_CTRL	[4:3]	HBR2 pattern control	0
DET_STA	[2]	Video stream clock detect status, It will not affect video output. 1: Stream clock detected 0: Stream clock not detected Write any value to update the current status.	0
FORCE_DET	[1]	Force video stream clock detect, this bit is only active when DET_CTRL is 1 1: Force video stream clock detected 0: Force video stream clock not detected This bit's type is R/W.	0
DET_CTRL	[0]	Video stream clock detect status control: 1: Use force detect status 0: Use auto-detected status This bit's type is R/W.	0

DP System Control Register #2 (SYS_CTL_2)

Register	Address	Type	Description	Reset Value
SYS_CTL_2	Base + 0x0604	R/W	System Control Register #2	0x0000_0040

SYS_CTL_2	Bit	Description	Initial State
-	[31:8]	Reserved	0
CHA_CRI	[7:4]	Pixel clock change detection threshold. The incoming pixel clock input is counted continuously by the 24Mhz reference clock. This register defines a number, if the counter number change is more than this value for 2 pixel clock edges, the CHA_STA bit is asserted. This bit's type is R/W.	4
-	[3]	Reserved	0
CHA_STA	[2]	Video stream clock change status, It will not affect video output 1: Clock frequency changed 0: Clock frequency not changed Write any value to update the current status.	0
FORCE_CHA	[1]	Force stream clock change status, this bit only active when CHA_CTRL is 1 1: Force clock change. When asserted, CHA_STA is '1'. 0: Force clock not change This bit's type is R/W.	0

SYS_CTL_2	Bit	Description	Initial State
CHA_CTRL	[0]	Pixel clock frequency change status control 1: Use force change status 0: Use auto-detected status This bit's type is R/W.	0

DP System Control Register #3 (SYS_CTL_3)

Register	Address	Type	Description	Reset Value
SYS_CTL_3	Base + 0x0608	R/W, RO	System Control Register #3.	0x0000_0000

SYS_CTL_3	Bit	Description	Initial State
-	[31:7]	Reserved	0
HPD_STATUS(RO)	[6]	Hot plug detect status. 1: HPD is 1, 0: HPD is 0. This bit's type is RO. When this bit is 0, AUX CH will not work. Note that the HPD_STATUS is only changed after the change of the pin I_DP_HPD remains for no less than hot plug deglitch time. And the hot plug deglitch time is defined in HPD_DEGLITCH_L and HPD_DEGLITCH_H.	0
F_HPD	[5]	Force hot plug detect. 1: Force HPD 1, 0: Force HPD 0. This bit's type is R/W.	0
HPD_CTRL	[4]	Hot plug detect manual control. 1: Force HPD with F_HPD, 0: Use PIN_HPD state. This bit's type is R/W.	0
HDCP_RDY(RO)	[3]	HDCP ready status. 1: HDCP is ready, 0: HDCP is not ready. This bit's type is RO. This bit is an indicator of whether HDCP is ready to perform. Usually, it is set as soon as HPD signal is detected as plugged.	0
STRM_VALID	[2]	Input stream have constant video format, and this stream is valid to send out through link. 1: Input stream is valid, 0: Input stream is not valid. Write any value to update the current status. Hardware will not send out video through link when this bit is 0.	0
F_VALID	[1]	Force stream valid, this bit only active when VALID_CTRL is 1. 1: Force input video stream valid, 0: Force input video stream not valid. This bit's type is R/W.	0

SYS_CTL_3	Bit	Description	Initial State
VALID_CTRL	[0]	Stream valid control. 1: Use F_VALID bit to control video stream valid status 0: Use video stream valid auto-detect This bit's type is R/W.	0

DP System Control Register #4 (SYS_CTL_4)

Register	Address	Type	Description	Reset Value
SYS_CTL_4	Base + 0x060C	R/W	System Control Register #2.	0x0000_0000

SYS_CTL_4	Bit	Description	Initial State
-	[31:4]	Reserved	0
ENHANCED	[3]	DisplayPort Enhanced mode enable 1: Enhanced mode, 0: Normal mode.	0
FIX_M_VID	[2]	Fix M_VID value 1: Use register M_VID value to be sent out, 0: Use calculates M_VID value to be sent out.	0
M_VID_UPDATE_CTRL	[1:0]	Control M_VID update frequency 11: 1/8 X update rate, 10: 1/4 X update rate, 01: 1/2 X update rate, 00: Normal rate.	0

DP Video Control Register (DP_VID_CTL)

Register	Address	Type	Description	Reset Value
DP_VID_CTL	Base + 0x0610	RO	DP Video Control Register	0x0000_0020

DP_VID_CTL	Bit	Description	Initial State
-	[31:8]	Reserved	0
BPC	[7:5]	Bit per color/ component with video which transferred via DP main link 101, 110, 111, 100: Reserved, 011: 12 bits, 010: 10 bits, 001: 8 bits, 000: 6 bits.	1
YC_COEFF	[4]	YcbCr Coefficients with video which transferred via DP main link 1: ITU709, 0: ITU601.	0
D_RANGE	[3]	Dynamic range 1: CEA range, 0: VESA range (from 0 to the maximum).	0

DP_VID_CTL	Bit	Description	Initial State
COLOR_F	[2:1]	Colorimetric format with video which transferred via DP main link 11: Reserved, 10: YcbCr444, 01: YcbCr422, 00: RGB.	0
-	[0]	Reserved	0

Packet Send Control Register (PKT_SEND_CTL)

Register	Address	Type	Description	Reset Value
PKT_SEND_CTL	Base + 0x0640	R/W	Packet Send Control Register.	0x0000_0000

PKT_SEND_CTL	Bit	Description	Initial State
-	[31:8]	Reserved	0
AUDIO_INFO_UP(C)	[7]	Audio InfoFrame content has been updated. 1: Updated, 0: Don't care. Write 1 to this bit after Audio Packet Content Registers have been configured as Audio InfoFrame content has been updated. This bit's type is R/W. This bit is self cleared after the register configured content has been used to update the InfoFrame.	0
AVI_INFO_UP(C)	[6]	AVI InfoFrame content has been updated. 1: Updated, 0: Don't care. Write 1 to this bit after AVI Packet Content Registers have been configured as AVI InfoFrame content has been updated. This bit's type is R/W. This bit is self cleared after the register configured content has been used to update the InfoFrame.	0
MPEG_INFO_UP(C)	[5]	MPEG InfoFrame content has been updated. 1: Updated, 0: Don't care. Write 1 to this bit after MPEG Packet Content Registers have been configured as MPEG InfoFrame content has been updated. This bit's type is R/W. This bit is self cleared after the register configured content has been used to update the InfoFrame.	0

PKT_SEND_CTL	Bit	Description	Initial State
IF_UP(C)	[4]	Configurable InfoFrame content has been updated. 1: Updated, 0: Don't care. Write 1 to this bit after IF_TYPE and IF_PKT_DB1~25 Registers have been configured as configurable InfoFrame content have been updated. This bit's type is R/W. This bit is self cleared after the register configured content has been used to update the InfoFrame.	0
AUDIO_INFO_EN	[3]	Audio InfoFrame send enable. 1: Send Audio InfoFrame, 0: Don't send Audio InfoFrame. Make sure that the Audio Packet Content Registers had been configured correctly and the AUDIO_INFO_UP had been written with 1. This bit's type is R/W.	0
AVI_INFO_EN	[2]	AVI InfoFrame send enable. 1: Send AVI InfoFrame, 0: Don't send AVI InfoFrame. Make sure that the AVI Packet Content Registers had been configured correctly and the AVI_INFO_UP had been written with 1. This bit's type is R/W.	0
MPEG_INFO_EN	[1]	MPEG InfoFrame send enable. 1: Send MPEG InfoFrame, 0: Don't send MPEG InfoFrame. Make sure that the MPEG Packet Content Registers had been configured correctly and the MPEG_INFO_UP had been written with 1. This bit's type is R/W.	0
IF_EN	[0]	Configurable InfoFrame send enable. 1: Send InfoFrame defined in IF_TYPE and IF_PKT_DB1~25, 0: Don't send InfoFrame. Make sure that the IF_TYPE and IF_PKT_DB1~25 Registers had been configured correctly and the IF_UP had been written with 1. This bit's type is R/W.	0

DP Main Link Bandwidth Setting Register (LINK_BW_SET)

Register	Address	Type	Description	Reset Value
LINK_BW_SET	Base + 0x0680	R/W	Main link bandwidth setting	0x0000_000A

LINK_BW_SET	Bit	Description	Initial State
-	[31:4]	Reserved	0
LINK_BW_SET	[3:0]	Main link bandwidth setting: 0x06: 1.62Gpbs per lane 0x0a: 2.7Gpbs per lane other: Reserved	A

DP Main Link Lane Count Register (LANE_COUNT_SET)

Register	Address	Type	Description	Reset Value
LANE_COUNT_SET	Base + 0x0684	R/W	Main link lane count	0x0000_0004

LANE_COUNT_SET	Bit	Description	Initial State
-	[31:3]	Reserved	0
LANE_COUNT_SET	[2:0]	Main link lane count 0x1: one lane 0x2: two lanes 0x4:four lanes other: Reserved	4

DP Training Pattern Set Register (DP_TRAINING_PTN_SET)

Register	Address	Type	Description	Reset Value
DP_TRAINING_PTN_SET	Base + 0x0688	R/W	DP Training Pattern Set Register	0x0000_0000

DP_TRAINING_PTN_SET	Bit	Description	Initial State
-	[31:6]	Reserved	0
SCRAMBLING_DISABLE	[5]	Disable scramble 1: Disable 0: Normal operation	0
LINK_QUAL_PATTERN_SET	[4:2]	Link quality pattern setting. 101 = HBR2 Compliance 100 = 80 bit test pattern 011 = PRBS 7 bit 010 = symbol error rate measurement pattern is sent; 001 = D10.2 test pattern is sent; 000= link quality test pattern not sent	0
SW_TRAINING_PATTERN_SET	[1:0]	Link training pattern setting. SW_TRAINING_PATTERN_SET has higher priority than LINK_QUAL_PATTER_SET. 11: Reserved 10: Sending training pattern 2 01: Sending training pattern 1 00: Training pattern not sent	0

DP Lane 0 Link Training Control Register (DP_LN0_LINK_TRAINING_CTL)

Register	Address	Type	Description	Reset Value
DP_LN0_LINK_TRAINING_CTL	Base + 0x068C	R/W, RO	DP Lane 0 Link Training Control Register.	0x0000_0000

<table>
<tr><th>DP_LN0_LINK_TRAINING_CTL</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td>-</td><td>[31:6]</td><td>Reserved</td><td>0</td></tr>
<tr><td>MAX_PRE_REACH_0(RO)</td><td>[5]</td><td>This bit field is set to 1 automatically when max pre-emphasis level of DP Tx is reached.
This bit’s type is RO.
Note that the MAX_PRE_REACH_0 and MAX_DRIVE_REACH_0 have the same value like the following table.
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Pre-emphasis (dB)</td></tr>
<tr><td>0</td><td>3.5</td><td>6.0</td><td>9.5</td></tr>
<tr><td rowspan="4">Driving Current (mV)</td><td>400</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>600</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>800</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1200</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Both of MAX_PRE_REACH_0 and MAX_DRIVE_REACH_0 are for test purpose only.</td><td>0</td></tr>
<tr><td>PRE_EMPHASIS_SET_0</td><td>[4:3]</td><td>Lane 0 pre-emphasis level setting
11: 9.5 dB,
10: 6.0 dB,
01: 3.5 dB,
00: 0 dB (No pre-emphasis).
This bit’s type is R/W.</td><td>0</td></tr>
<tr><td>MAX_DRIVE_REACH_0(RO)</td><td>[2]</td><td>This bit field is set to 1 automatically when max driving current level of DP Tx is reached. For test purpose only. This bit’s type is RO. For more information, refer to MAX_PRE_REACH_0.</td><td>0</td></tr>
<tr><td>DRIVE_CURRENT_SET_0</td><td>[1:0]</td><td>Lane 0 output amplitude setting
11: 1200 mV,
10: 800 mV,
01: 600 mV,
00: 400 mV.
This bit’s type is R/W.</td><td>0</td></tr>
</table>

DP Lane 1 Link Training Control Register (DP_LN1_LINK_TRAINING_CTL)

Register	Address	Type	Description	Reset Value
DP_LN1_LINK_TRAINING_CTL	Base + 0x0690	R/W, RO	DP Lane 1 Link Training Control Register.	0x0000_0000

DP_LN1_LINK_TRAINING_CTL	Bit	Description	Initial State
-	[31:6]	Reserved	0

<table>
<tr><th>DP_LN1_LINK_TRAINING_CTL</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td>MAX_PRE_REACH_1(RO)</td><td>[5]</td><td>This bit field is set to 1 automatically when max pre-emphasis level of DP Tx is reached.
Note that the MAX_PRE_REACH_1 and MAX_DRIVE_REACH_1 have the same value like the following table.
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Pre-emphasis (dB)</td></tr>
<tr><td>0</td><td>3.5</td><td>6.0</td><td>9.5</td></tr>
<tr><td rowspan="4">Driving Current (mV)</td><td>400</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>600</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>800</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1200</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Both of MAX_PRE_REACH_1 and MAX_DRIVE_REACH_1 are for test purpose only.
This bit’s type is RO.</td><td>0</td></tr>
<tr><td>PRE_EMPHASIS_SET_1</td><td>[4:3]</td><td>Lane 1 pre-emphasis level setting
11: 9.5 dB,
10: 6.0 dB,
01: 3.5 dB,
00: 0 dB (No pre-emphasis).
This bit’s type is R/W.</td><td>0</td></tr>
<tr><td>MAX_DRIVE_REACH_1(RO)</td><td>[2]</td><td>This bit field is set to 1 automatically when max driving current level of DP Tx is reached. For more information, refer to MAX_PRE_REACH_1. For test purpose only.
This bit’s type is RO.</td><td>0</td></tr>
<tr><td>DRIVE_CURRENT_SET_1</td><td>[1:0]</td><td>Lane 1 output amplitude setting
11: 1200 mV,
10: 800 mV,
01: 600 mV,
00: 400 mV.
This bit’s type is R/W.</td><td>0</td></tr>
</table>

DP Lane 2 Link Training Control Register (DP_LN2_LINK_TRAINING_CTL)

Register	Address	Type	Description	Reset Value
DP_LN2_LINK_TRAINING_CTL	Base + 0x0694	R/W, RO	DP Lane 2 Link Training Control Register.	0x0000_0000

DP_LN2_LINK_TRAINING_CTL	Bit	Description	Initial State
-	[31:6]	Reserved	0

<table>
<tr><th>DP_LN2_LINK_TRAINING_CTL</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td>MAX_PRE_REACH_2(RO)</td><td>[5]</td><td>This bit field is set to 1 automatically when max pre-emphasis level of DP Tx is reached.
Note that the MAX_PRE_REACH_2 and MAX_DRIVE_REACH_2 have the same value like the following table.
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Pre-emphasis (dB)</td></tr>
<tr><td>0</td><td>3.5</td><td>6.0</td><td>9.5</td></tr>
<tr><td rowspan="4">Driving Current (mV)</td><td>400</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>600</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>800</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1200</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Both of MAX_PRE_REACH_1 and MAX_DRIVE_REACH_2 are for test purpose only.
This bit’s type is RO.</td><td>0</td></tr>
<tr><td>PRE_EMPHASIS_SET_2</td><td>[4:3]</td><td>Lane 2 pre-emphasis level setting
11: 9.5 dB,
10: 6.0 dB,
01: 3.5 dB,
00: 0 dB (No pre-emphasis).
This bit’s type is R/W.</td><td>0</td></tr>
<tr><td>MAX_DRIVE_REACH_2(RO)</td><td>[2]</td><td>This bit field is set to 1 automatically when max driving current level of DP Tx is reached. For more information, refer to MAX_PRE_REACH_2. For test purpose only.
This bit’s type is RO.</td><td>0</td></tr>
<tr><td>DRIVE_CURRENT_SET_2</td><td>[1:0]</td><td>Lane 2 output amplitude setting
11: 1200 mV,
10: 800 mV,
01: 600 mV,
00: 400 mV.
This bit’s type is R/W.</td><td>0</td></tr>
</table>

DP Lane 3 Link Training Control Register (DP_LN3_LINK_TRAINING_CTL)

Register	Address	Type	Description	Reset Value
DP_LN3_LINK_TRAINING_CTL	Base + 0x0698	R/W, RO	DP Lane 3 Link Training Control Register.	0x0000_0000

DP_LN3_LINK_TRAINING_CTL	Bit	Description	Initial State
-	[31:6]	Reserved	0

<table>
<tr><th>DP_LN3_LINK_TRAINING_CTL</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td>MAX_PRE_REACH_3(RO)</td><td>[5]</td><td>This bit field is set to 1 automatically when max pre-emphasis level of DP Tx is reached.
Note that the MAX_PRE_REACH_3 and MAX_DRIVE_REACH_3 have the same value like the following table.
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Pre-emphasis (dB)</td></tr>
<tr><td>0</td><td>3.5</td><td>6.0</td><td>9.5</td></tr>
<tr><td rowspan="4">Driving Current (mV)</td><td>400</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>600</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>800</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1200</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Both of MAX_PRE_REACH_3 and MAX_DRIVE_REACH_3 are for test purpose only.
This bit’s type is RO.</td><td>0</td></tr>
<tr><td>PRE_EMPHASIS_SET_3</td><td>[4:3]</td><td>Lane 3 pre-emphasis level setting
11: 9.5 dB,
10: 6.0 dB,
01: 3.5 dB,
00: 0 dB (No pre-emphasis).
This bit’s type is R/W.</td><td>0</td></tr>
<tr><td>MAX_DRIVE_REACH_3(RO)</td><td>[2]</td><td>This bit field is set to 1 automatically when max driving current level of DP Tx is reached. For more information, refer to MAX_PRE_REACH_3. For test purpose only.
This bit’s type is RO.</td><td>0</td></tr>
<tr><td>DRIVE_CURRENT_SET_3</td><td>[1:0]</td><td>Lane 3 output amplitude setting
11: 1200 mV,
10: 800 mV,
01: 600 mV,
00: 400 mV.
This bit’s type is R/W.</td><td>0</td></tr>
</table>

DP HW LINK TRAINING_CONTROL Register (DP_HW_LINK_TRAINING_CTL)

Register	Address	Type	Description	Reset Value
DP_HW_LINK_TRAINING_CTL	Base + 0x06A0	R/W, RO	DP hardware training control registers.	0x0000_0000

DP_HW_LINK_TRAINING_CTL	Bit	Description	Initial State
-	[31:7]	Reserved	0
HW_TRAINING_ERROR_CODE(RO)	[6:4]	Training error code 0: OK 1: AUX_WRITE_ERROR 2: MAX_DRIVE_REACHED 3: WRONG_LANE_COUNT_SETTING 4: LOOP_SAME_5_TIME 5: CR_FAIL_IN_EQ 6: EQ_LOOP_5_TIME This bit’s type is RO.	0

DP_HW_LINK_TRAINING_CTL	Bit	Description	Initial State
-	[3:1]	Reserved	0
HW_TRAINING_EN(C)	[0]	Link training sequence enable Write 1 to enable training sequence, write 0 to force training sequence stop, this bit will self-clear when training done. This bit's type is R/W. This bit is self cleared.	0

DP Debug Register Register #1 (DP_DEBUG_CTL)

Register	Address	Type	Description	Reset Value
DP_DEBUG_CTL	Base + 0x06C0	R/W, RO	DP Debug Control Register #1.	0x0000_0000

DP_DEBUG_CTL	Bit	Description	Initial State
-	[31:7]	Reserved	0
MYDP_HPD_POLLIN_EN	[6]	Enable the MYDP HPD status polling. If this bit and POLLING_EN are enabled and BYPASS_STATUS_POLLING is 0, hardware polling both of link status and MYDP HPD status. 1: Enabled 0: Disabled. This bit's type is RW.	0
BYPASS_STATUS_POLLING	[5]	Bypass link status polling. If this bit, MYDP_HPD_POLLIN_EN and POLLING_EN are all enabled, hardware only polling MYDP HPD status. 1: Enabled 0: Disabled. This bit's type is RW.	0
PLL_LOCK(RO)	[4]	PLL lock status 1: PLL lock, 0: PLL unlock. This bit's type is RO.	0
F_PLL_LOCK	[3]	Force PLL lock, this bit is active when PLL_LOCK_CTRL is 1: 1: Force PLL lock, 0: Force PLL non-lock. This bit's type is R/W	0
PLL_LOCK_CTRL	[2]	PLL lock register control enable 1: PLL lock signal is controlled by register, 0: PLL lock signal is controlled by PLL. This bit's type is R/W	0
POLLING_EN	[1]	Enable hardware state machine to polling the HPD status or link status. The interval of each polling is controlled by POLLING_PERIOD 1: Enable polling function. 0: Disable polling function This bit's type is R/W.	0

PN_INV	[0]	Invert SERDES output polarity 1: Invert output polarity, 0: Normal operation. This bit's type is R/W.	0

DP HPD De-glitch Low Byte Register (HPD_DEGLITCH_L)

Register	Address	Type	Description	Reset Value
HPD_DEGLITCH_L	Base + 0x06C4	R/W	HPD_DEGLITCH is used to de-glitch the HPD signal	0x0000_005E

HPD_DEGLITCH_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
HPD_DEGLITCH_L	[7:0]	HPD_DEGLITCH, which is counted at 24 MHz, is used to de-glitch the HPD signal This register is HPD_DEGLITCH [7:0]. The default value is 0x5E for 280.75 us deglitch time.	0x5E

DP HPD De-glitch High Byte Register (HPD_DEGLITCH_H)

Register	Address	Type	Description	Reset Value
HPD_DEGLITCH_H	Base + 0x06C8	R/W	HPD_DEGLITCH is used to de-glitch the HPD signal	0x0000_001A

HPD_DEGLITCH_H	Bit	Description	Initial State
-	[31:6]	Reserved	0
HPD_DEGLITCH_H	[5:0]	HPD_DEGLITCH, which is counted at 24 MHz, is used to de-glitch the HPD signal. This register is HPD_DEGLITCH [13:8]. The default value is 0x1A for 280.75 us deglitch time.	0x1A

DP POLLING_PERIOD Register (POLLING_PERIOD)

Register	Address	Type	Description	Reset Value
POLLING_PERIOD	Base + 0x06CC	R/W	POLLING_PERIOD	0x0000_000E

POLLING_PERIOD	Bit	Description	Initial State
-	[31:8]	Reserved	0
POLLING_PERIOD	[7:0]	This register controls the interval between each time of polling operation. Interval time = POLLING_PERIOD * 2^16 * Period of 24M clock.	0xE

DP Link Debug Control Register (DP_LINK_DEBUG_CTL)

Register	Address	Type	Description	Reset Value
DP_LINK_DEBUG_CTL	0x0000_06E0	R/W	DP Link Debug Control Register	0x0000_0010

DP_LINK_DEBUG_CTL	Bit	Description	Initial State
-	[31:5]	Reserved	0
NEW_PRBS7	[4]	Control the PRBS 7 formula. 1: Use new PRBS7 formula in DP 1.1 version 0: Use old PRBS7 formula in DP 1.0 version	1
DIS_FIFO_RST	[3]	Disable video FIFO reset every line 1: Disable, 0: Reset video FIFO every line.	0
DISABLE_AUTO_RESET_ENCODER	[2]	Disable 8b/10 encoder auto reset 1: Disabled auto reset 8b/10 encode before sending Link Training Pattern 2 0: Auto reset 8b/10 encode before sending Link Training Pattern 2	0
-	[1]	Reserved	0
PRBS31_EN	[0]	Enable DisplayPort PRBS 31. 1: Enabled, 0: Normal mode.	0

DP SINK_COUNT Register (SINK_COUNT)

Register	Address	Type	Description	Reset Value
SINK_COUNT	Base + 0x06E4	RO	SINK_COUNT	0x0000_0000

SINK_COUNT	Bit	Description	Initial State
-	[31:8]	Reserved	0
SINK_COUNT	[7:0]	Sink Count	0x0

DP IRQ_VECTOR Register (IRQ_VECTOR)

Register	Address	Type	Description	Reset Value
IRQ_VECTOR	Base + 0x06E8	RO	IRQ_VECTOR	0x0000_0000

IRQ_VECTOR	Bit	Description	Initial State
-	[31:8]	Reserved	0
IRQ_VECTOR	[7:0]	Irq_vector	0x0

DP_LINK_STATUS0 Register (DP_LINK_STATUS0)

Register	Address	Type	Description	Reset Value
DP_LINK_STATUS0	Base + 0x06EC	RO	DP_LINK_STATUS0	0x0000_0000

DP_LINK_STATUS0	Bit	Description	Initial State
-	[31:7]	Reserved	0
LN1_SYBOL_LOCK	[6]	Lane1 symbol lock	0x0
LN_EQ_DONE	[5]	Lane1 EQ done	0x0
LN_CR_DONE	[4]	Lane1 CR done	0x0
-	[3]	Reserved	0x0
LN0_SYBOL_LOCK	[2]	Lane0 symbol lock	0x0

LN0_EQ_DONE	[1]	Lane0 EQ done	0x0
LN0_CR_DONE	[0]	Lane0 CR done	0x0

DP_LINK_STATUS1 Register (DP_LINK_STATUS1)

Register	Address	Type	Description	Reset Value
DP_LINK_STATUS1	Base + 0x06F0	RO	DP_LINK_STATUS1	0x0000_0000

DP_LINK_STATUS1	Bit	Description	Initial State
-	[31:8]	Reserved	0
INTER_LN_ALIGN	[7]	Interlace align	0x0
LN3_SYMBOL_LOCK	[6]	Lane3 symbol lock	0x0
LN3_EQ_DONE	[5]	Lane3 EQ done	0x0
LN3_CR_DONE	[4]	Lane3 CR done	0x0
-	[3]	Reserved	0x0
LN2_SYMBOL_LOCK	[2]	Lane2 symbol lock	0x0
LN2_EQ_DONE	[1]	Lane2 EQ done	0x0
LN2_CR_DONE	[0]	Lane2 CR done	0x0

DP ALIGN_STATUS Register (ALIGN_STATUS)

Register	Address	Type	Description	Reset Value
ALIGN_STATUS	Base + 0x06F4	RO	ALIGN_STATUS	0x0000_0000

ALIGN_STATUS	Bit	Description	Initial State
-	[31:8]	Reserved	0
ALIGN_STATUS	[7:0]	ALIGN_STATUS	0

DP DP_SINK_STATUS Register (DP_SINK_STATUS)

Register	Address	Type	Description	Reset Value
DP_SINK_STATUS	Base + 0x06F8	RO	DP_SINK_STATUS	0x0000_0000

DP_SINK_STATUS	Bit	Description	Initial State
-	[31:2]	Reserved	0
SINK_STA_1	[1]	Debug register	0
SINK_STA_0	[0]	Debug register	0

M_VID Configure Register #0 (M_VID_0)

Register	Address	Type	Description	Reset Value
M_VID_0	Base + 0x0700	R/W	M_VID[7:0]	0x0000_0000

M_VID_0	Bit	Description	Initial State
-	[31:8]	Reserved	0

M_VID_0	[7:0]	M_VID [7:0]. If FIX_M_VID is 1, this M_VID is used. Otherwise the M_VID value which chip calculated is used	0

DP M_VID Configure Register #1 (M_VID_1)

Register	Address	Type	Description	Reset Value
M_VID_1	Base + 0x0704	R/W	M_VID[15:8]	0x0000_0000

M_VID_1	Bit	Description	Initial State
-	[31:8]	Reserved	0
M_VID_1	[7:0]	M_VID [15:8]. If FIX_M_VID is 1, this M_VID is used. Otherwise the M_VID value which chip calculated is used	0

DP M_VID Configure Register #2 (M_VID_2)

Register	Address	Type	Description	Reset Value
M_VID_2	Base + 0x0708	R/W	M_VID[23:16]	0x0000_0000

M_VID_2	Bit	Description	Initial State
-	[31:8]	Reserved	0
M_VID_2	[7:0]	M_VID [23:16]. If FIX_M_VID is 1, this M_VID is used. Otherwise the M_VID value which chip calculated is used	0

DP N_VID Configure Register #0 (N_VID_0)

Register	Address	Type	Description	Reset Value
N_VID_0	Base + 0x070C	R/W	N_VID[7:0]	0x0000_0000

N_VID_0	Bit	Description	Initial State
-	[31:8]	Reserved	0
N_VID_0	[7:0]	N_VID[7:0] The maximum value of M_VID is 0xFFFF in ASYNC mode.	0

DP N_VID Configure Register #1 (N_VID_1)

Register	Address	Type	Description	Reset Value
N_VID_1	Base + 0x0710	R/W	N_VID[15:8]	0x0000_0080

N_VID_1	Bit	Description	Initial State
-	[31:8]	Reserved	0
N_VID_1	[7:0]	N_VID[15:8]	0x80

DP N_VID Configure Register #2 (N_VID_2)

Register	Address	Type	Description	Reset Value
N_VID_2	Base + 0x0714	R/W	N_VID[23:16]	0x0000_0000

N_VID_2	Bit	Description	Initial State
-	[31:8]	Reserved	0
N_VID_2	[7:0]	N_VID[23:16]	0

DP M_VID_MON register (M_VID_MON)

Register	Address	Type	Description	Reset Value
M_VID_MON	Base + 0x0718	RO	M_VID value monitoring register	0x0000_0000

M_VID_MON	Bit	Description	Initial State
-	[31:24]	Reserved	0
M_VID_MON	[23:0]	This register shows M_VID value which is actually transmitted to Rx for monitoring purpose.	0x0

DP FIFO Threshold Register (DP_VIDEO_FIFO_THRD)

Register	Address	Type	Description	Reset Value
DP_VIDEO_FIFO_THRD	Base + 0x0730	R/W	DP Video Data FIFO Threshold Register	0x0000_0000

DP_VIDEO_FIFO_THRD	Bit	Description	Initial State
-	[31:5]	Reserved	0
VIDEO_TH_CTRL	[4]	Video Data FIFO threshold control enables. 1: Video Data FIFO threshold uses VIDEO_TH_VALUE. 0: Video Data FIFO threshold uses internal calculate value automatically.	0
VIDEO_TH_VALUE	[3:0]	Video Data FIFO threshold value. If VIDEO_TH_CTRL is 1, and data count in video data FIFO have reached FIFO threshold value, video data is read out from FIFO.	0

DP GNS Control Register (DP_GNS_CTRL)

Register	Address	Type	Description	Reset Value
DP_GNS_CTRL	Base + 0x0734	R/W	DP GNS CONTROL REGISTER	0x0000_0018

DP_GNS_CTRL	Bit	Description	Initial State
-	[31:7]	Reserved	0
EQ_TRAINING_LOOP_CONTROL	[6]	1: enable 0: disable	0
-	[5]	Reserved	0
SCRAMBLE_CTRL	[4]	Scramble formula control: 1 = new formula; 0 = old formula;	1
IN_EX	[3]	Control scrambler structure: 1 = Internal type. 0 = External type.	1
DISABLE_SERDES_FIFO_RSET	[2]	1 = Disable serdes FIFO auto reset. 0 = Enable serdes FIFO auto reset	0
VIDEO_MAP_CTRL	[1]	Control use or not the video data map in YCbCr 4:2:2 mode: 1 = use video data map in YCbCr 4:2:2 mode. 0 = don't use.	0

RS_CTRL	[0]	Control RS parameter: 1 = parameter define by V1.0. 0 = parameter in GNS	0

DP M Value Calculation Control Register (DP_M_CAL_CTL)

Register	Address	Type	Description	Reset Value
DP_M_CAL_CTL	Base + 0x0760	R/W	DP M Value Calculation Control Register	0x0000_0000

DP_M_CAL_CTL	Bit	Description	Initial State
-	[31:3]	Reserved	0
M_VID_GEN_FILTER_EN	[2]	Enable M_VID value generation filter to reduce the variation of M_VID value. This filter is a low-pass filter to smooth out the M_VID variation 1: Enable the filter 0: Disable the filter Note: Refer to page 22 for details.	0
-	[1]	Reserved	0
M_GEN_CLK_SEL	[0]	Select which link clock is used to generate the M value 1: Clock with down spreading is used 0: Clock without down spreading is used	0

DP M_VID Value Calculation Control Register (M_VID_GEN_FILTER_TH)

Register	Address	Type	Description	Reset Value
M_VID_GEN_FILTER_TH	Base + 0x0764	R/W	The threshold of M_VID generation filter	0x0000_0004

M_VID_GEN_FILTER_TH	Bit	Description	Initial State
-	[31:8]	Reserved	0
M_VID_GEN_FILTER_TH	[7:0]	The threshold of M_VID generation filter It only takes effect when M_VID_GEN_FILTER_EN is set to 1	4

AUX Channel Access Status Register (AUX_CH_STA)

Register	Address	Type	Description	Reset Value
AUX_CH_STA	Base + 0x0780	RO	AUX Channel Access Status Register	0x0000_0000

AUX_CH_STA	Bit	Description	Initial State
-	[31:5]	Reserved	0
AUX_BUSY	[4]	AUX channel status bit. If this bit is read as 1, AUX channel access should be halted. 1: AUX CH is busy 0: AUX CH is idle	0

AUX_STATUS	[3:0]	This register indicate the AUX channel access status 0: OK 1: NACK_ERROR 2: TIMEOUT_ERROR 3: UNKNOWN_ERROR 4: MUCH_DEFER_ERROR 5: TX_SHORT_ERROR 6: RX_SHORT_ERROR 7: NACK_WITHOUT_M_ERROR 8: I2C_NACK_ERROR Other: Reserved.	0

AUX Channel Access Error Code Register (AUX_ERR_NUM)

Register	Address	Type	Description	Reset Value
AUX_ERR_NUM	Base + 0x0784	RO	AUX Channel Access Error Code Register	0x0000_0000

AUX_ERR_NUM	Bit	Description	Initial State
-	[31:8]	Reserved	0
AUX_ERR_NUM	[7:0]	The error number counter of AUX channel counts when AUX channel access failed. In AUX CH reading, this number indicates the number of read back byte. In AUX CH writing, this number indicates the number of reply command.	0

DP AUX CH DEFER Control Register (AUX_CH_DEFER_CTL)

Register	Address	Type	Description	Reset Value
AUX_CH_DEFER_CTL	Base + 0x0788	R/W	DP AUX CH DEFER Control Register	0x0000_007F

AUX_CH_DEFER_CTL	Bit	Description	Initial State
-	[31:8]	Reserved	0
DEFER_CTRL_EN	[7]	AUX CH received DEFER command count control enable 1: If the count that AUX CH receive DEFER command equal to (DEFER_COUNT * 64), the AUX CH transaction is terminated, and the AUX_STATUS is 0100 0: The count that AUX CH receive DEFER command is unlimited	0
DEFER_COUNT	[6:0]	The count is defined to limit the max count AUX CH receive DEFER command When DEFER_CTRL_EN is 1 and AUX CH received (DEFER_COUNT * 64) DEFER command, the AUX CH will terminate the transaction	0x7F

DP AUX RX Command Register (AUX_RX_COMM)

Register	Address	Type	Description	Reset Value
AUX_RX_COMM	Base + 0x078C	RO	AUX CH received command	0x0000_0000

AUX_RX_COMM	Bit	Description	Initial State
-	[31:4]	Reserved	0
AUX_RX_COMM	[3:0]	AUX CH received command	0

DP Buffer Data Count Register (BUFFER_DATA_CTL)

Register	Address	Type	Description	Reset Value
BUFFER_DATA_CTL	Base + 0x0790	RO/C	DP Buffer Data Count Register	0x0000_0000

BUFFER_DATA_CTL	Bit	Description	Initial State
-	[31:8]	Reserved	0
BUF_CLR(C)	[7]	Write 1 to this bit to clear AUX CH data buffer (BUF_DATA_0 ~ BUF_DATA_15). Always read back 0 from this bit. This bit's type is R/W. This bit is self cleared. Note: For the write operation, set this bit to 1 before writing data to BUF_DATA_0~15. And for READ operation, this bit has only to be set before starting data transfer by setting AUX_EN.	0
-	[6:5]	Reserved	0
BUF_HAVE_DATA(RO)	[4]	0:buffer have data,1:buffer have not data	0
BUF_DATA_COUNT(RO)	[3:0]	The counts of data AUX CH buffer have. This bit's type is RO.	0

DP AUX Channel Control Register 1 (AUX_CH_CTL_1)

Register	Address	Type	Description	Reset Value
AUX_CH_CTL_1	Base + 0x0794	R/W	DP AUX Channel Control Register 1	0x0000_0000

AUX_CH_CTL_1	Bit	Description	Initial State
-	[31:8]	Reserved	0
AUX_LENGTH	[7:4]	Register control AUX CH transaction length.	0
AUX_TX_COMM	[3:0]	Register control AUX CH transaction command.	0

DP AUX CH Address Register #0 (AUX_ADDR_7_0)

Register	Address	Type	Description	Reset Value
AUX_ADDR_7_0	Base + 0x0798	R/W	AUX_ADDR[7:0]	0x0000_0000

AUX_ADDR_7_0	Bit	Description	Initial State
-	[31:8]	Reserved	0
AUX_ADDR_7_0	[7:0]	AUX_ADDR[7:0], Register control AUX CH address	0

DP AUX CH Address Register #1 (AUX_ADDR_15_8)

Register	Address	Type	Description	Reset Value
AUX_ADDR_15_8	Base + 0x079C	R/W	AUX_ADDR[15:8]	0x0000_0000

AUX_ADDR_15_8	Bit	Description	Initial State
-	[31:8]	Reserved	0
AUX_ADDR_15_8	[7:0]	AUX_ADDR[15:8], Register control AUX CH address	0

DP AUX CH Address Register #2 (AUX_ADDR_19_16)

Register	Address	Type	Description	Reset Value
AUX_ADDR_19_16	Base + 0x07A0	R/W	AUX_ADDR[19:16]	0x0000_0000

AUX_ADDR_19_16	Bit	Description	Initial State
-	[31:4]	Reserved	0
AUX_ADDR_19_16	[3:0]	AUX_ADDR[7:0], Register control AUX CH address	0

DP AUX CH Control Register 2 (AUX_CH_CTL_2)

Register	Address	Type	Description	Reset Value
AUX_CH_CTL_2	Base + 0x07A4	R/W /C	DP AUX CH Control Register 2	0x0000_0000

AUX_CH_CTL_2	Bit	Description	Initial State
-	[31:4]	Reserved	0
PD_AUX_IDLE	[3]	Power down AUX CH when AUX CH is in idle state. 1 : Power down AUX CH in idle state. 0 : Keep AUX CH power up in idle state.	0
AUX_PN_INV	[2]	Invert AUX CH PN 1: Invert PN 0: Normal mode	0
ADDR_ONLY	[1]	AUX CH issue "address only" command 1: Issue "address only" command 0: Normal AUX CH command	0
AUX_EN(C)	[0]	Register control AUX CH operation enable Write 1 to this bit to enable AUX CH operation This bit will self-clear when AUX CH operation is finished. This bit is self cleared.	0

DP AUX Buffer Data Register (BUF_DATA_0 ~ BUF_DATA_15)

Register	Address	Type	Description	Reset Value
BUF_DATA_0 ~ BUF_DATA_15	Base + 0x07C0 ~ Base+0x07FC	R/W	AUX CH buffer data 0 ~ 15	0x0000_00FF

BUF_DATA_0 ~ BUF_DATA_15	Bit	Description	Initial State
-	[31:8]	Reserved	0
BUF_DATA_0 ~ BUF_DATA_15	[7:0]	AUX CH buffer data 0 ~ 15	0xFF

Register definitions for SoC IP implementation

ATE test control register (ATE_TEST_CTL)

Register	Address	Type	Description	Reset Value
ATE_TEST_CTL	Base + 0x0804	R/W C	ATE test control register	0x0000_0000

ATE_TEST_CTL	Bit	Description	Initial State
-	[31:9]	Reserved	0
TX_ATE	[15:8]	ATE test enable Bit 15~12: Reserved for analog test. Bit 11: ate_en in ch3, Bit 10: ate_en in ch2, Bit 9: ate_en in ch1 Bit 8: ate_en in ch0	0
-	[7:6]	Reserved	0
ATE_TEST_DATA_INV	[5]	Invert ate test data	0
ATE_ERR_GEN_EN_IN(C)	[4]	Insert a ERR for PHY ATE test. Self clear	0
ATE_CLR_ERR(C)	[3:0]	Clear error counter [3]:lane3,[2]:lane2,[1]:lane1,[0]:lane0	0

ATE test status register (ATE_TEST_STATUS)

Register	Address	Type	Description	Reset Value
ATE_TEST_STATUS	Base + 0x0808	RO	ATE test control register	0x0000_0000

ATE_TEST_STATUS	Bit	Description	Initial State
-	[31:20]	Reserved	0
ERROR_INC	[19:16]	ERROR indicator [19]:lane3,[18]:lane2,[17]:lane1,[16]:lane0	0
PRBS7 CHECK FSM STATE	[15:0]	PRBS7 check FSM state [15:12]:lane3,[11:8]:lane2,[7:4]:lane1,[3:0]:lane0	0

ATE test error counter register (ATE_TEST_ERR_CNT)

Register	Address	Type	Description	Reset Value
ATE_TEST_ERR_CNT	Base + 0x080C~ Base + 0x0818	RO	ATE test error counter register	0x0000_0000

ATE_TEST_STATUS	Bit	Description	Initial State
ATE_TEST_ERR_CNT	[31:0]	ATE test error counter register.0x080C—lane0, 0x0810—lane1, 0x0814—lane2, 0x0818—lane3	0

DP test 80bit pattern0 (DP_TEST_80B_PATTERN0)

Register	Address	Type	Description	Reset Value
DP_TEST_80B_PATTERN0	Base + 0x081C	R/W	DP test 80bit pattern0	0x0000_0000

DP_TEST_80B_PATTERN0	Bit	Description	Initial State
-	[31:30]	Reserved	0
DP_TEST_80B_PATTERN0	[29:0]	DP test 80bit pattern0[29:0]	0

DP test 80bit pattern1 (DP_TEST_80B_PATTERN1)

Register	Address	Type	Description	Reset Value
DP_TEST_80B_PATTERN1	Base + 0x0820	R/W	DP test 80bit pattern1	0x0000_0000

DP_TEST_80B_PATTERN1	Bit	Description	Initial State
-	[31:30]	Reserved	0
DP_TEST_80B_PATTERN1	[29:0]	DP test 80bit pattern0[59:30]	0x0000_0000

DP test 80bit pattern2 (DP_TEST_80B_PATTERN2)

Register	Address	Type	Description	Reset Value
DP_TEST_80B_PATTERN2	Base + 0x0824	R/W	DP test 80bit pattern2	0x0000_0000

DP_TEST_80B_PATTERN0	Bit	Description	Initial State
-	[31:20]	Reserved	0
DP_TEST_80B_PATTERN0	[19:0]	DP test 80bit pattern0[79:60]	0x0000_0000

DP test HBR2 SR COUNT (DP_TEST_HBR2_PATTERN)

Register	Address	Type	Description	Reset Value
DP_TEST_HBR2_PATTERN	Base + 0x0828	R/W	Hbr2 compliance SR count	0x0000_0010

DP_TEST_80B_PATTERN0	Bit	Description	Initial State
-	[31:16]	Reserved	0
DP_TEST_HBR2_PATTERN	[15:0]	Hbr2 compliance SR count	0x0000_0010

CRC check control register (CRC_CON)

Register	Address	Type	Description	Reset Value
CRC_CON	Base + 0x0890	R/W	CRC check control	0x0000_0000

CRC_CON	Bit	Description	Initial State
-	[31:3]	Reserved	0
PSR_VID_CRC_FLUSH	[2]	PSR Video CRC flush enable. The PSR video CRC value is initialized at every v-sync leading edge.	0
-	[1]	Reserved	0
PSR_VID_CRC_ENABLE	[0]	PSR Video CRC enable. 0: Disable, 1: Enable	0

PSR frame CRC R/CR component low byte (FRAME_CRC_R_CR_L)

Register	Address	Type	Description	Reset Value
FRAME_CRC_R_CR_L	Base + 0x0894	R/W	PSR frame CRC value of R/CR low byte	0x0000_0000

FRAME_CRC_R_CR_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRAME_CRC_R_CR_L	[7:0]	PSR frame CRC value of R/CR low byte	0

PSR frame CRC R/CR component high byte (FRAME_CRC_R_CR_H)

Register	Address	Type	Description	Reset Value
FRAME_CRC_R_CR_H	Base + 0x0898	R/W	PSR frame CRC value of R/CR high byte	0x0000_0000

FRAME_CRC_R_CR_H	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRAME_CRC_R_CR_H	[7:0]	PSR frame CRC value of R/CR high byte	0

PSR frame CRC G/Y component low byte (FRAME_CRC_G_Y_L)

Register	Address	Type	Description	Reset Value
FRAME_CRC_G_Y_L	Base + 0x089C	R/W	PSR frame CRC value of G/Y low byte	0x0000_0000

FRAME_CRC_G_Y_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRAME_CRC_G_Y_L	[7:0]	PSR frame CRC value of G/Y low byte	0

PSR frame CRC G/Y component high byte (FRAME_CRC_G_Y_H)

Register	Address	Type	Description	Reset Value
FRAME_CRC_G_Y_H	Base + 0x08A0	R/W	PSR frame CRC value of G/Y high byte	0x0000_0000

FRAME_CRC_G_Y_H	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRAME_CRC_G_Y_H	[7:0]	PSR frame CRC value of G/Y high byte	0

PSR frame CRC B/CB component low byte (FRAME_CRC_B_CB_L)

Register	Address	Type	Description	Reset Value
FRAME_CRC_B_CB_L	Base + 0x08A4	R/W	PSR frame CRC value of B/CB low byte	0x0000_0000

FRAME_CRC_B_CB_L	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRAME_CRC_B_CB_L	[7:0]	PSR frame CRC value of B/CB low byte	0

PSR frame CRC B/CB component high byte (FRAME_CRC_B_CB_H)

Register	Address	Type	Description	Reset Value
FRAME_CRC_B_CB_H	Base + 0x08A8	R/W	PSR frame CRC value of B/CB high byte	0x0000_0000

FRAME_CRC_G_Y_H	Bit	Description	Initial State
-	[31:8]	Reserved	0
FRAME_CRC_B_CB_H	[7:0]	PSR frame CRC value of B/CB high byte	0

ATE CRC Result (ATE_CRC_RESULT)

Register	Address	Type	Description	Reset Value
ATE_CRC_RESULT	Base + 0x08AC	RO	ATE CRC result	0x0000_0000

ATE_CRC_RESULT	Bit	Description	Initial State
-	[31:16]	Reserved	0
ATE_VID_CRC_RESULT	[15:0]	ATE Video CRC result	0

Analog Control Register 5 (ANALOG_CTL_5)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_5	Base + 0x0914	R/W	Analog Control Register 5	0x0000_0000

ANALOG_CTL_5	Bit	Description	Initial State
-	[31:8]	Reserved	0
CH3_PC2_SEL	[7:6]	Ch3 post cursor2 setting: CH3_PC2_SEL: post cursor2 0x0: 0 0x1: 0.05 0x2: 0.1 0x3: 0.15	0
CH3_PC2_SEL	[5:4]	Ch2 post cursor2 setting: CH2_PC2_SEL: post cursor2 0x0: 0 0x1: 0.05 0x2: 0.1 0x3: 0.15	0
CH3_PC2_SEL	[3:2]	Ch1 post cursor2 setting: CH1_PC2_SEL: post cursor2 0x0: 0 0x1: 0.05 0x2: 0.1 0x3: 0.15	0
CH3_PC2_SEL	[1:0]	Ch0 post cursor2 setting: CH0_PC2_SEL: post cursor2 0x0: 0 0x1: 0.05 0x2: 0.1 0x3: 0.15	0

Analog Control Register 6 (ANALOG_CTL_6)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_6	Base + 0x0918	R/W	Analog Control Register 6	0x0000_0050

ANALOG_CTL_6	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_400MV_0DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 400mv and Pre_emphasis is 0 db.	0x50

Analog Control Register 7 (ANALOG_CTL_7)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_7	Base + 0x091C	R/W	Analog Control Register 7	0x0000_0078

ANALOG_CTL_7	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_600MV_0DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 600mv and Pre_emphasis is 0 db.	0x78

Analog Control Register 8 (ANALOG_CTL_8)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_8	Base + 0x0920	R/W	Analog Control Register 8	0x0000_00A0

ANALOG_CTL_8	Bit	Description	Initial State

-	[31:8]	Reserved	0
R_AMP_800MV_0DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 800mv and Pre_emphasis is 0 db.	0xA0

Analog Control Register 9 (ANALOG_CTL_9)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_9	Base + 0x0924	R/W	Analog Control Register 9	0x0000_00F0

ANALOG_CTL_9	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_1200MV_0DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 1200mv and Pre_emphasis is 0 db.	0xF0

Analog Control Register 10 (ANALOG_CTL_10)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_10	Base + 0x0928	R/W	Analog Control Register 10	0x0000_0064

ANALOG_CTL_10	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_400MV_3P5DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 400mv and Pre_emphasis is 3.5 db.	0x64

Analog Control Register 11 (ANALOG_CTL_11)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_11	Base + 0x092C	R/W	Analog Control Register 11	0x0000_0096

ANALOG_CTL_11	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_600MV_3P5DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 600mv and Pre_emphasis is 3.5 db.	0x96

Analog Control Register 12 (ANALOG_CTL_12)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_12	Base + 0x0930	R/W	Analog Control Register 12	0x0000_00C8

ANALOG_CTL_12	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_800MV_3P5DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 800mv and Pre_emphasis is 3.5 db.	0XC8

Analog Control Register 13 (ANALOG_CTL_13)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_13	Base + 0x0934	R/W	Analog Control Register 13	0x0000_0078

ANALOG_CTL_13	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_400MV_6DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 400mv and Pre_emphasis is 6db.	0x78

Analog Control Register 14 (ANALOG_CTL_14)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_14	Base + 0x0938	R/W	Analog Control Register 14	0x0000_00B4

ANALOG_CTL_14	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_600MV_6DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 600mv and Pre_emphasis is 6db.	0xB4

Analog Control Register 15 (ANALOG_CTL_15)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_15	Base + 0x093C	R/W	Analog Control Register 15	0x0000_00A0

ANALOG_CTL_15	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_AMP_400MV_9DB	[7:0]	The lookup-table 1(for calculating chx_swing_bit) value when V_diff is 400mv and Pre_emphasis is 9db.	0xA0

Analog Control Register 16 (ANALOG_CTL_16)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_16	Base + 0x0940	R/W	Analog Control Register 16	0x0000_0000

ANALOG_CTL_16	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_400MV_0DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 400mv and Pre_emphasis is 0db.	0x00

Analog Control Register 17 (ANALOG_CTL_17)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_17	Base + 0x0944	R/W	Analog Control Register 17	0x0000_0000

ANALOG_CTL_17	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_600MV_0DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 600mv and Pre_emphasis is 0 db.	0x00

Analog Control Register 18 (ANALOG_CTL_18)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_18	Base + 0x0948	R/W	Analog Control Register 18	0x0000_0000

ANALOG_CTL_18	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_800MV_0DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 800mv and Pre_emphasis is 0 db.	0x00

Analog Control Register 19 (ANALOG_CTL_19)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_19	Base + 0x094C	R/W	Analog Control Register 19	0x0000_0000

ANALOG_CTL_19	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_1200MV_0DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 1200mv and Pre_emphasis is 0 db.	0x00

Analog Control Register 20 (ANALOG_CTL_20)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_20	Base + 0x0950	R/W	Analog Control Register 20	0x0000_0028

ANALOG_CTL_20	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_400MV_3P5DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 400mv and Pre_emphasis is 3.5 db.	0x28

Analog Control Register 21 (ANALOG_CTL_21)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_21	Base + 0x0954	R/W	Analog Control Register 21	0x0000_003C

ANALOG_CTL_21	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_600MV_3P5DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 600mv and Pre_emphasis is 3.5 db.	0x3C

Analog Control Register 22 (ANALOG_CTL_22)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_22	Base + 0x0958	R/W	Analog Control Register 22	0x0000_0050

ANALOG_CTL_22	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_800MV_3P5DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 800mv and Pre_emphasis is 3.5 db.	0X50

Analog Control Register 23 (ANALOG_CTL_23)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_23	Base + 0x095C	R/W	Analog Control Register 23	0x0000_0050

ANALOG_CTL_23	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_400MV_6DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 400mv and Pre_emphasis is 6db.	0x50

Analog Control Register 24 (ANALOG_CTL_24)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_24	Base + 0x0960	R/W	Analog Control Register 24	0x0000_0078

ANALOG_CTL_24	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_600MV_6DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 600mv and Pre_emphasis is 6db.	0x78

Analog Control Register 25 (ANALOG_CTL_25)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_25	Base + 0x0964	R/W	Analog Control Register 25	0x0000_00A0

ANALOG_CTL_25	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_EMP_400MV_9DB	[7:0]	The lookup-table 2(for calculating chx_pre_emp_bit) value when V_diff is 400mv and Pre_emphasis is 9db.	0xA0

Analog Control Register 26 (ANALOG_CTL_26)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_26	Base + 0x0968	R/W	Analog Control Register 26	0x0000_0004

ANALOG_CTL_26	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_400MV_0DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 400mv and Pre_emphasis is 0db.	0x04

Analog Control Register 27 (ANALOG_CTL_27)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_27	Base + 0x096C	R/W	Analog Control Register 27	0x0000_0006

ANALOG_CTL_27	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_600MV_0DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 600mv and Pre_emphasis is 0 db.	0x06

Analog Control Register 28 (ANALOG_CTL_28)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_28	Base + 0x0970	R/W	Analog Control Register 28	0x0000_0008

ANALOG_CTL_28	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_800MV_0DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 800mv and Pre_emphasis is 0 db.	0x08

Analog Control Register 29 (ANALOG_CTL_29)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_29	Base + 0x0974	R/W	Analog Control Register 29	0x0000_000C

ANALOG_CTL_29	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_1200MV_0DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 1200mv and Pre_emphasis is 0 db.	0x0C

Analog Control Register 30 (ANALOG_CTL_30)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_30	Base + 0x0978	R/W	Analog Control Register 30	0x0000_0006

ANALOG_CTL_30	Bit	Description	Initial State

-	[31:6]	Reserved	0
R_PC2_400MV_3P5DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 400mv and Pre_emphasis is 3.5 db.	0x06

Analog Control Register 31 (ANALOG_CTL_31)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_31	Base + 0x097C	R/W	Analog Control Register 31	0x0000_0009

ANALOG_CTL_31	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_600MV_3P5DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 600mv and Pre_emphasis is 3.5 db.	0x09

Analog Control Register 32 (ANALOG_CTL_32)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_32	Base + 0x0980	R/W	Analog Control Register 32	0x0000_000C

ANALOG_CTL_32	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_800MV_3P5DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 800mv and Pre_emphasis is 3.5 db.	0X0C

Analog Control Register 33 (ANALOG_CTL_33)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_33	Base + 0x0984	R/W	Analog Control Register 33	0x0000_0008

ANALOG_CTL_33	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_400MV_6DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 400mv and Pre_emphasis is 6db.	0x08

Analog Control Register 34 (ANALOG_CTL_34)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_34	Base + 0x0988	R/W	Analog Control Register 34	0x0000_000C

ANALOG_CTL_34	Bit	Description	Initial State
-	[31:6]	Reserved	0

R_PC2_600MV_6DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 600mv and Pre_emphasis is 6db.	0x0C

Analog Control Register 35 (ANALOG_CTL_35)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_35	Base + 0x098C	R/W	Analog Control Register 35	0x0000_000C

ANALOG_CTL_35	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_PC2_400MV_9DB	[5:0]	The lookup-table 3(for calculating chx_swing_bit and chx_pc2_bit) value when V_diff is 400mv and Pre_emphasis is 9db.	0x0C

Analog Control Register 36 (ANALOG_CTL_36)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_36	Base + 0x0990	R/W	Analog Control Register 36	0x0000_0050

ANALOG_CTL_36	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_CH0_AMP_FORCE_VALUE	[7:0]	The forced ch0 amp value (for calculating ch0_swing_bit) value in specific V_diff and Pre_emphasis.	0x50

Analog Control Register 37 (ANALOG_CTL_37)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_37	Base + 0x0994	R/W	Analog Control Register 37	0x0000_0000

ANALOG_CTL_37	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_CH0_EMP_FORCE_VALUE	[7:0]	The forced ch0 emp value (for calculating ch0_pre_emphasis_bit) value in specific V_diff and Pre_emphasis.	0x00

Analog Control Register 38 (ANALOG_CTL_38)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_38	Base + 0x0998	R/W	Analog Control Register 38	0x0000_0004

ANALOG_CTL_38	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_CH0_PC2_FORCE_VALUE	[5:0]	The forced ch0 PC2 value (for calculating ch0_swing_bit and ch0_pc2_bit) value in specific V_diff and Pre_emphasis.	0x04

Analog Control Register 39 (ANALOG_CTL_39)

Register	Address	Type	Description	Reset Value

ANALOG_CTL_39	Base + 0x099C	R/W	Analog Control Register 39	0x0000_0050

ANALOG_CTL_39	**Bit**	**Description**	**Initial State**
-	[31:8]	Reserved	0
R_CH1_AMP_FORCE_VALUE	[7:0]	The forced ch1 amp value (for calculating ch1_swing_bit) value in specific V_diff and Pre_emphasis.	0x50

Analog Control Register 40 (ANALOG_CTL_40)

Register	**Address**	**Type**	**Description**	**Reset Value**
ANALOG_CTL_40	Base + 0x09A0	R/W	Analog Control Register 40	0x0000_0000

ANALOG_CTL_40	**Bit**	**Description**	**Initial State**
-	[31:8]	Reserved	0
R_CH1_EMP_FORCE_VALUE	[7:0]	The forced ch1 emp value (for calculating ch1_pre_emphasis_bit) value in specific V_diff and Pre_emphasis.	0x00

Analog Control Register 41 (ANALOG_CTL_41)

Register	**Address**	**Type**	**Description**	**Reset Value**
ANALOG_CTL_41	Base + 0x09A4	R/W	Analog Control Register 41	0x0000_0004

ANALOG_CTL_41	**Bit**	**Description**	**Initial State**
-	[31:6]	Reserved	0
R_CH1_PC2_FORCE_VALUE	[5:0]	The forced ch1 PC2 value (for calculating ch1_swing_bit and ch1_pc2_bit) value in specific V_diff and Pre_emphasis.	0x04

Analog Control Register 42 (ANALOG_CTL_42)

Register	**Address**	**Type**	**Description**	**Reset Value**
ANALOG_CTL_42	Base + 0x09A8	R/W	Analog Control Register 42	0x0000_0000

ANALOG_CTL_42	**Bit**	**Description**	**Initial State**
-	[31:6]	Reserved	0
R_FORCE_CH1_AMP	[5]	0x1: The result of ch1 swing bit is decide by R_CH1_AMP_FORCE_VALUE value 0x0: The result of ch1 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH1_EMP	[4]	0x1: The result of ch1 pre emphasis bit is decide by R_CH1_EMP_FORCE_VALUE value 0x0: The result of ch1 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH1_PC2	[3]	0x1: The result of ch1 pc2 bit is decide by R_CH1_PC2_FORCE_VALUE value 0x0: The result of ch1 swing bit is decide by different V_diff and Pre_emphasis	0

R_FORCE_CH0_AMP	[2]	0x1: The result of ch0 swing bit is decide by R_CH1_AMP_FORCE_VALUE value 0x0: The result of ch0 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH0_EMP	[1]	0x1: The result of ch0 pre emphasis bit is decide by R_CH1_EMP_FORCE_VALUE value 0x0: The result of ch0 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH0_PC2	[0]	0x1: The result of ch0 pc2 bit is decide by R_CH1_PC2_FORCE_VALUE value 0x0: The result of ch0 swing bit is decide by different V_diff and Pre_emphasis	0

Analog Control Register 43 (ANALOG_CTL_43)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_43	Base + 0x09AC	R/W	Analog Control Register 43	0x0000_0050

ANALOG_CTL_43	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_CH2_AMP_FORCE_VALUE	[7:0]	The forced ch2 amp value (for calculating ch2_swing_bit) value in specific V_diff and Pre_emphasis.	0x50

Analog Control Register 44 (ANALOG_CTL_44)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_44	Base + 0x09B0	R/W	Analog Control Register 44	0x0000_0000

ANALOG_CTL_44	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_CH2_EMP_FORCE_VALUE	[7:0]	The forced ch2 emp value (for calculating ch2_pre_emphasis_bit) value in specific V_diff and Pre_emphasis.	0x00

Analog Control Register 45 (ANALOG_CTL_45)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_45	Base + 0x09B4	R/W	Analog Control Register 45	0x0000_0004

ANALOG_CTL_45	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_CH2_PC2_FORCE_VALUE	[5:0]	The forced ch2 PC2 value (for calculating ch2_swing_bit and ch2_pc2_bit) value in specific V_diff and Pre_emphasis.	0x04

Analog Control Register 46 (ANALOG_CTL_46)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_46	Base + 0x09B8	R/W	Analog Control Register 46	0x0000_0050

ANALOG_CTL_46	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_CH3_AMP_FORCE_VALUE	[7:0]	The forced ch3 amp value (for calculating ch3_swing_bit) value in specific V_diff and Pre_emphasis.	0x50

Analog Control Register 47 (ANALOG_CTL_47)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_47	Base + 0x09BC	R/W	Analog Control Register 47	0x0000_0000

ANALOG_CTL_47	Bit	Description	Initial State
-	[31:8]	Reserved	0
R_CH3_EMP_FORCE_VALUE	[7:0]	The forced ch3 emp value (for calculating ch3_pre_emphasis_bit) value in specific V_diff and Pre_emphasis.	0x00

Analog Control Register 48 (ANALOG_CTL_48)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_48	Base + 0x09C0	R/W	Analog Control Register 48	0x0000_0004

ANALOG_CTL_48	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_CH3_PC2_FORCE_VALUE	[5:0]	The forced ch3 PC2 value (for calculating ch3_swing_bit and ch3_pc2_bit) value in specific V_diff and Pre_emphasis.	0x04

Analog Control Register 49 (ANALOG_CTL_49)

Register	Address	Type	Description	Reset Value
ANALOG_CTL_49	Base + 0x09C4	R/W	Analog Control Register 49	0x0000_0000

ANALOG_CTL_49	Bit	Description	Initial State
-	[31:6]	Reserved	0
R_FORCE_CH3_AMP	[5]	0x1: The result of ch3 swing bit is decide by R_CH1_AMP_FORCE_VALUE value 0x0: The result of ch3 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH3_EMP	[4]	0x1: The result of ch3 pre emphasis bit is decide by R_CH1_EMP_FORCE_VALUE value 0x0: The result of ch3 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH3_PC2	[3]	0x1: The result of ch3 pc2 bit is decide by R_CH1_PC2_FORCE_VALUE value 0x0: The result of ch3 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH2_AMP	[2]	0x1: The result of ch2 swing bit is decide by R_CH1_AMP_FORCE_VALUE value 0x0: The result of ch2 swing bit is decide by different V_diff and Pre_emphasis	0

R_FORCE_CH2_EMP	[1]	0x1: The result of ch2 pre emphasis bit is decide by R_CH1_EMP_FORCE_VALUE value 0x0: The result of ch2 swing bit is decide by different V_diff and Pre_emphasis	0
R_FORCE_CH2_PC2	[0]	0x1: The result of ch2 pc2 bit is decide by R_CH1_PC2_FORCE_VALUE value 0x0: The result of ch2 swing bit is decide by different V_diff and Pre_emphasis	0

LINK_POLICY (LINK_POLICY)

Register	Address	Type	Description	Reset Value
LINK_POLICY	Base + 0x09D8	R/W	Link_Policy	0x0000_0050

LINK_POLICY	Bit	Description	Initial State
-	[31:8]	Reserved	0
ALTERNATE_SR_EN	[7]	Alternate SR enable	0
LINK_TRAIN_CR_LP_IN	[6:4]	Link training CR loop in	5
LINK_TRAIN_WR_EN	[3]	Training first write en	0
LINK_TRAIN_405G	[2]	405g training enable	0
LINK_TRAIN_INV	[1]	Invert training bit enable	0
FRAME_CHANGE_EN	[0]	Framing change enable	0

6.5 Interface Description

6.5.1 Video Input Source

In RK3399, the eDP TX video source can come from vop_big or vop_lit.

- GRF_SOC_CON20[5] == 1'b0, video source from vop_big
- GRF_SOC_CON20[5] == 1'b1, video source from vop_lit

6.5.2 Hot plug

There is a hot plug input signal to eDP TX controller. This signal is muxed with GPIO4C[7], and is enabled by "GRF_GPIO4C_IOMUX[15:14] = 2'b10".

Chapter 7 MIPI D-PHY

7

7.1 Overview

The DesignWare Cores MIPI D-PHY Bidirectional 4-Lane macro for TMSC 28-nm HPC /1.8V （referred to as PHY) implements the physical layer of bidirectional universal lanes for the MIPI D-PHY interface. The PHY is stacked in a configuration with four data lanes and one clock lane. The PHY can be reused for both master and slave applications. The lane modules are bidirectional with HS-TX, HS-RX, LP-TX, LP-RX, and LP-CD functions, but with no support for high-speed reverse communication. The PHY also includes a clock multiplier PLL for high-speed (HS) clock generation needed in a master-side application. It is targeted for the digital data transmission between a host processor and display drivers or camera interfaces in mobile applications, supporting a maximum effective bit rate of 1.5 Gbps per lane. The assembled four-data-lane system enables up to 6 Gbps aggregate communication throughputs, delivering the bandwidth needed for high-throughput data transfer. Because of its dual master/slave reusability, the PHY builds a bidirectional high-speed differential interface for serial data transmission. There is an additional reduced-throughput, low-power data transfer mode in each differential pair, which reduces line count and minimizes cable wires and EMI shielding requirements.
The following shows a typical application for PHY.

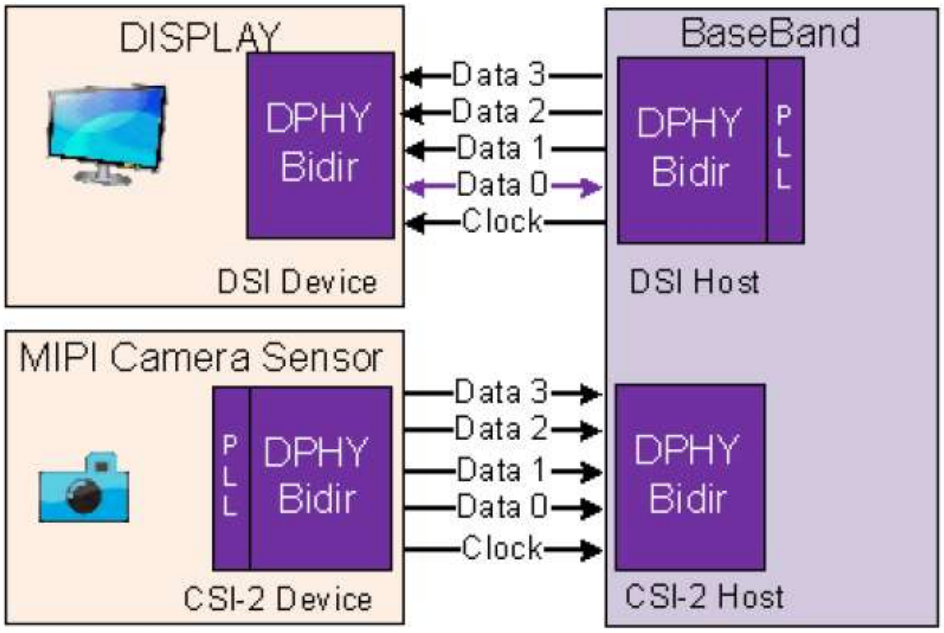


Fig. 7-1 Typical Application for the DWC MIPI D-PHY Bidir 4L IP

The MIPI D-PHY supports the following features:

- Attachable PLL clock multiplication unit for master-side functionality
- Flexible input clock reference — 5 MHz to 500 MHz
- 50% DDR output clock duty-cycle
- Lane operation ranging from 80 Mbps to 1.5 Gbps in forward direction
- Aggregate throughput up to 6 Gbps with four data lanes
- PHY-Protocol Interface (PPI) for clock and data lanes
- Low-power Escape modes and Ultra Low Power state
- 1.8 V +-10% analog supply operation
- 0.9 V +-10% digital supply operation
- Core Area:
  - For Slave configuration: 0.477 mm^2
  - For Master configuration: 0.576 mm^2

7.2 Block Diagram

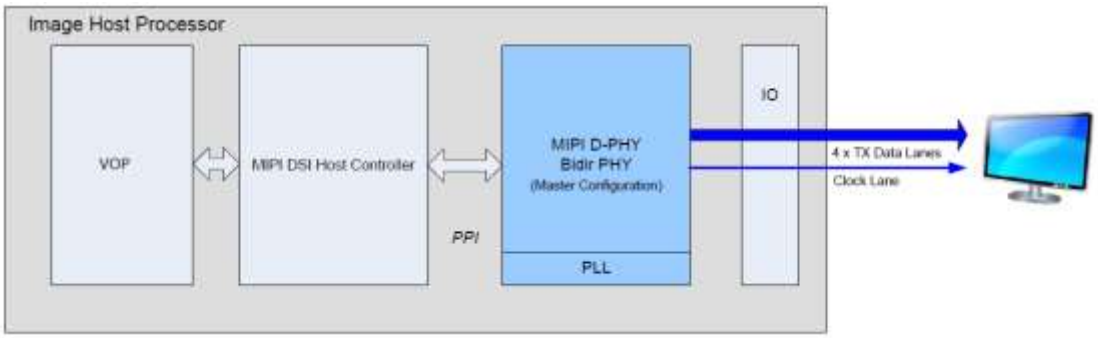


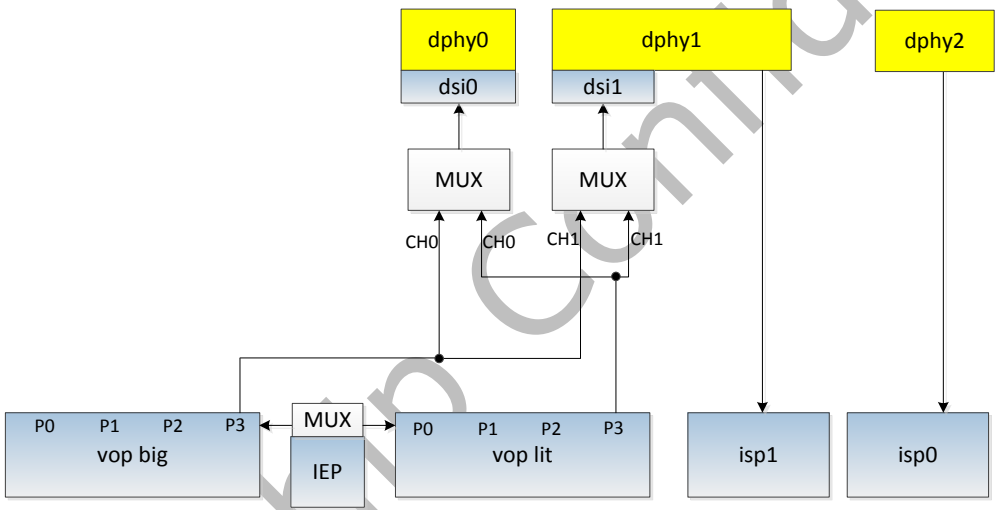
Fig. 7-2 MIPI D-PHY detailed block diagram

MIPI D-PHY configuration contains one Clock Lane Module and four Data Lane Modules. Each of these PHY Lane Modules communicates via two Lines to a complementary part at the other side of the Lane Interconnect.

7.3 Function Description

7.3.1 System connection

There are three D-PHY instances in RK3399, their connection are shown as following figure:



In above figure, dphy0 is D-PHY TX0, dphy1 is D-PHY TX1RX1 and dphy2 is D-PHY RX0.

D-PHY RX0

D-PHY RX0 is only used for RX, receive the Mipi Camera data then send to ISP0.

D-PHY TX0

D-PHY TX0 is only used for TX, send the data from VOP_BIG or VOP_LIT to the Mipi Panel. You can select data from VOP_BIG or VOP_LIT by setting grf_con_dsi0_lcdc_sel (GRF_SOC_CON20[0])

D-PHY TX1RX1

D-PHY TX1RX1 can configure to for TX or for RX.

The D-PHY can be configured to for TX by setting as below:

grf_dphy_tx1rx1_masterslavez (GRF_SOC_CON24[7]) = 1'b1

grf_dphy_tx1rx1_basedir(GRF_SOC_CON24[5]) = 1'b0

grf_con_dsi1_lcdc_sel (GRF_SOC_CON20[4]) to select the data from VOP_BIG or from VOP_LIT.

The D-PHY can be configured to for RX by setting as below:

grf_dphy_tx1rx1_masterslavez (GRF_SOC_CON24[7]) = 1'b0

grf_dphy_tx1rx1_basedir(GRF_SOC_CON24[5]) = 1'b1.

In this case, it receives the Mipi Camera data then send to ISP1.

The detail register setting is as following table:

Table 7-1 Register Config For D-PHY Mode Select

VOP_BIG +DSI0+TX0	VOP_LIT +DSI0+TX0	VOP_BIG +DSI1+ TX1 RX1	VOP_LIT +DSI1+TX1RX1
GRF_SOC_CON20[0] =1'b0	GRF_SOC_CON20[0] =1'b1	GRF_SOC_CON20[4] =1'b0	GRF_SOC_CON20[4] =1'b1
GRF_SOC_CON22[3:0]=4'h0 GRF_SOC_CON22[7:4]=4'h0	GRF_SOC_CON22[3:0]=4'h0 GRF_SOC_CON22[7:4]=4'h0	GRF_SOC_CON23[7:4]=4'h0 GRF_SOC_CON23[11:8]=4'h0	GRF_SOC_CON23[7:4]=4'h0 GRF_SOC_CON23[11:8]=4'h0
		GRF_SOC_CON24[5] = 1'b0 GRF_SOC_CON24[7] = 1'b1	GRF_SOC_CON24[5] = 1'b0 GRF_SOC_CON24[7] = 1'b1
RX0 + ISP0	**TX1RX1 + ISP1**		
GRF_SOC_CON21[7:4]=4'h0 GRF_SOC_CON21[11:8]=4'h0	GRF_SOC_CON23[7:4]=4'h0 GRF_SOC_CON23[11:8]=4'h0		
	GRF_SOC_CON24[5] =1'b1 GRF_SOC_CON24[7] =1'b0		

7.3.2 Test and control code

The MIPI D-PHY contains a set of test and control codes that can be used for testing either under the scope of normal silicon characterization or for production test in the ATE environment. These control codes are primarily used for the configuration of normal operation of the MIPI D-PHY, but for the test purposes they are referred to as test codes. The MIPI DPHY test interface in RK3399 can be accessed from configuration and test interface by MIPI DSI HOST controller.

Interface Timing

The standard procedure is two-folded; first the necessary test code is programmed and then the related test data words are fed to the tester's inputs.

To configure a test code, have the MIPI D-PHY in shutdown mode (SHUTDOWNZ=0) and then reset it (RSTZ=0). This avoids the transient periods in the operation during re-configuration procedures. It is also recommended to apply a tester reset pulse (TESTCLR = 1) before any test code configuration.

The test code programming is done using the following steps:

1) Set the desired test code
   a. Ensure that TESTCLK is set to high.
   b. Place the 8-bit word corresponding to the test code in TESTDIN.
   c. Set TESTEN to high.
   d. Set TESTCLK to low. With the falling edge on TESTCLK, the TESTDIN[7:0] signal content is latched internally as the current test code.
   e. Set testen to low.
2) Enter the necessary test data
   a. Set TESTCLK to low, if not done already.
   b. Place the 8-bit word corresponding to the required test data in TESTDIN.
   c. Set TESTCLK to high. Test data is programmed internally.
   d. Repeat the steps to add more test data for the same test code.

Repeat the above procedure to program subsequent test codes. Additionally, a test reset procedure (TESTCLR = 1) is only needed prior to the first programming operation or if you wish to reset the MIPI D-PHY's configuration to its default values and override any changes made meanwhile. Figure bellow shows a generic timing diagram for test operation. After a test code is effectively programmed, TESTDOUT[7:0] asynchronously outputs relevant data for that specific test code, whether it is pure read-back data or other meaningful signals.

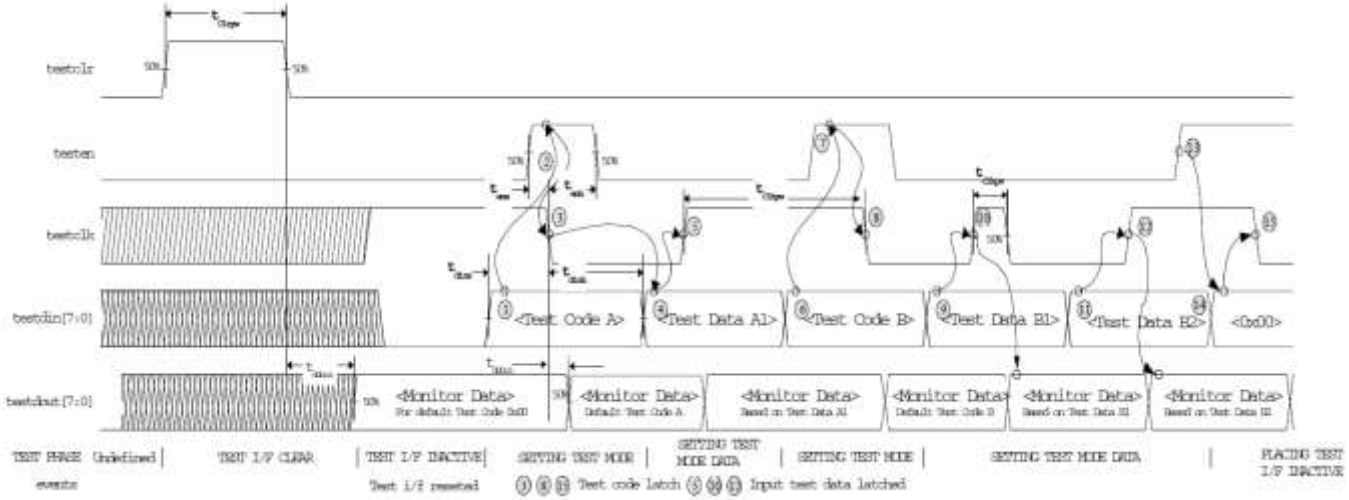


Fig. 7-3 Testability Interface Timing Diagram

7.3.3 Operating Modes

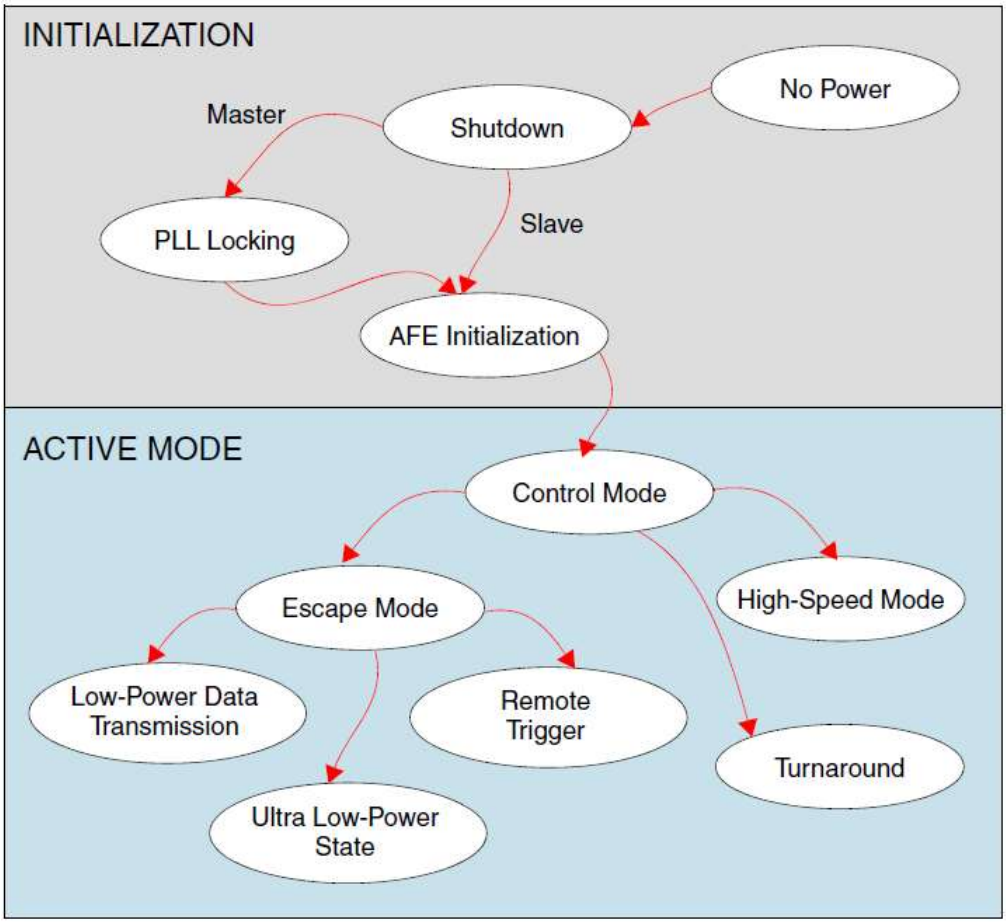


Fig. 7-4 MIPI D-PHY Initialization from Shutdown to Control Modes

7.3.4 Initialization

- **NO POWER MODE**

The No Power mode is characterized by the non-existence of any supply voltage applied to MIPI D-PHY Bidir 4L. In order to get to the powered modes, proper voltages should be applied sequentially to MIPI D-PHY, this task is usually done by the SoC PMU or eventually by global powering up sequence.

The recommended powering up sequence is that the core voltage (VDD) powers up first and the I/O voltage (AVDD) powers up next. This is not considered as a constraint, but instead a guideline, as it results in the best-case operating scenario, where power-down currents are kept to a minimum.

- **Shutdown Mode**

Shutdown mode is the lowest power consumption mode, where all analog blocks are disabled, and digital logic is reset. The current consumption is given by the analog stand-by current and the digital logic leakage current. It is entered asynchronously when RSTZ and SHUTDOWNZ are in low state. It should be ensured that the TESTCLR signal is asserted by default, as it acts as an active high reset to the control block responsible for the configuration values preset.

In this mode, the differential lines of DATAN/DATAP and CLKN/CLKP are high impedance (Hi-Z).

Depending on the MIPI D-PHY usage, some additional steps can be performed. By default, MIPI D-PHY is configured to work only on the lower operation range of 80-110 Mbps. If higher bit rate operation is required, you should set the register hsfreqrange (HS RX Control of Lane 0) with the proper code. If MIPI D-PHY is expected to work always at the same bit rate, this additional step can be performed while in Shutdown mode as the control interface is independent of the rest of MIPI D-PHY. Conversely, if the MIPI D-PHY is expected to change the bit rate after initialization, hsfreqrange should be updated while in

Control mode.
In addition, when working in Master mode, the PLL must be configured to the function of the REFCLK input frequency and the desired output frequency, which determines the bit rate on the transmission path.
When RSTZ and SHUTDOWNZ are set to logic high level, MIPI D-PHY leaves this state and starts an initialization procedure.

- **PLL Locking Mode and AFE Initialization**

The MIPI D-PHY consists of four data lanes, but applications can use four or lesser number data lanes. In such cases, you are granted access to individual enabling signals (ENABLE_N) that control which lanes should be used and evolve through all the necessary initialization steps. It is assumed that such configurations are static or at least are stable prior to leaving the Shutdown mode.
The D-PHY specification has several timing intervals which have to be followed to ensure proper operation. The fact that some of those timing intervals often have absolute timing components—some are combined with UI values and some with only absolute timing—makes it difficult to meet the maximum and minimum values across the complete data rate range (80 Mbps-1.5 Gbps) by just using default settings. To cope with this situation, the MIPI D-PHY implements a set of frequency ranges that needs to be configured prior to starting normal operation, either in Master or Slave operation using the hsfreqrange register.

Table 7-2 Frequency Ranges

Range (Mbps)	hsfreqrange[5:0]
80-89	000000
90-99	010000
100-109	100000
110-129	000001
130-139	010001
140-149	100001
150-169	000010
170-179	010010
180-199	100010
200-219	000011
220-239	010011
240-249	100011
250-269	000100
270-299	010100
300-329	000101
330-359	010101
360-399	100101
400-449	000110
450-499	010110
500-549	000111
550-599	010111
600-649	001000
650-699	011000
700-749	001001
750-799	011001
800-849	101001
850-899	111001
900-949	001010
950-999	011010
1000-1049	101010
1050-1099	111010
1100-1149	001011
1150-1199	011011
1200-1249	101011

Range (Mbps)	hsfreqrange[5:0]
1250-1299	111011
1300-1349	001100
1350-1399	011100
1400-1449	101100
1450-1500	111100

The hsfreqrange field is accessible through Test Code 0x44 when TESTDIN[7] = 0 and TESTDIN[0] = 0. The hsfreqrange[5:0] field is programmed with the contents of TESTDIN[6:1] at every rising edge of TESTCLK.

After the reset signals—RSTZ and SHUTDOWNZ—are released, the MIPI D-PHY begins an initialization sequence that allows its correct operation. The sequence of the release of signals is not critical, but it is recommended that SHUTDOWNZ should precede RSTZ; it is also assumed that the CFG_CLK signal is available and stable by that time.

In a Master configuration, the PLL becomes active and MIPI D-PHY goes through the PLL Locking mode, in which the MIPI D-PHY waits for the PLL to acquire lock, indicated by the LOCK output going high. A valid REFCLK (FREFCLK) should be provided.

Following the PLL lock, the rest of the AFE is initialized leading to the enabling of the low-power drivers. After completing these transitory states, the lines go to the Stop state (LP = 11) and the TX achieves active mode.

In the case of a Slave configuration (MASTERSLAVEZ = 1'b0), PLL is inactive, therefore only the rest of AFE initialization takes place.

- **Active Modes**
  - Control Mode

Control mode is the default operating mode. After the initialization is completed (analog calibrations and PLL locking for Master configurations), the MIPI D-PHY remains in this default mode until some request is placed. The request is placed either by the protocol layer for TX, or directly through the sequence of low-power signals in the lanes in case of RX. While in control mode, the transmitter side sets the LP-11 state in the lines - this is called the Stop state. The receiver side remains in control mode while receiving LP-11 in the lines. Any request must start from and end in Stop state. Following a request, a lane can leave control mode for either high-speed data transfer mode, Escape mode, Ultra Low Power state, or turnaround operation.

  - High-Speed Data Transfer Mode

Once the initialization sequence is completed, the MIPI D-PHY remains in control mode, which is the default operating mode, until some request appears. High-speed is one of the possible requests at this point. High-speed data transfer occurs in bursts. Only during these bursts the lane is in high-speed mode. A high-speed burst must start from and return to a Stop state (control mode). A high-speed burst allows for the transmission of payload data by the data lanes. Inherent to such data transmission is the existence of a valid DDR clock in the clock lane. High-speed data bursts are independent for each lane, which means that each data lane can start and end a high-speed transmission independently of the state of the remaining data lanes.

A burst contains the low-power initialization sequence, the high-speed data payload, and also the end of transmission sequence.

  - Escape Mode

Escape mode is a special mode of operation that uses the data lanes to communicate asynchronously using the low-power states at low-speed. The MIPI D-PHY supports this mode in both directions. A Data Lane enters the Escape mode through an Escape mode entry procedure (LP-11, LP-10, LP-00, LP-01, LP-00), if an LP-11 is detected before reaching LP-00 state, the entry is aborted and the receiver returns to the Stop state. Once the sequence is correctly completed, the transmitter sends an 8-bit command to indicate a requested action. The MIPI D-PHY applies Spaced-One-Hot encoding (a Mark state is interleaved with a Space state) on commands and data.

Each symbol consists of the following two parts: One-Hot phase and Space state. To transmit one bit, a Mark-1 should be sent followed by the Space state. In the case of a zero bit, a Mark-0 should be sent followed by Space state.

  - Turnaround

The MIPI D-PHY allows the transmission direction of the data lanes to be swapped by means of a turnaround request.
Initialization

7.3.5 PLL Requirements

Because the MIPI D-PHY is expected to be used in applications where the lane bit rate requirements can change from system to system, it features a flexible clock-multiplying PLL architecture. This PLL uses a VCO comprised of a pseudo-differential oscillator ring.

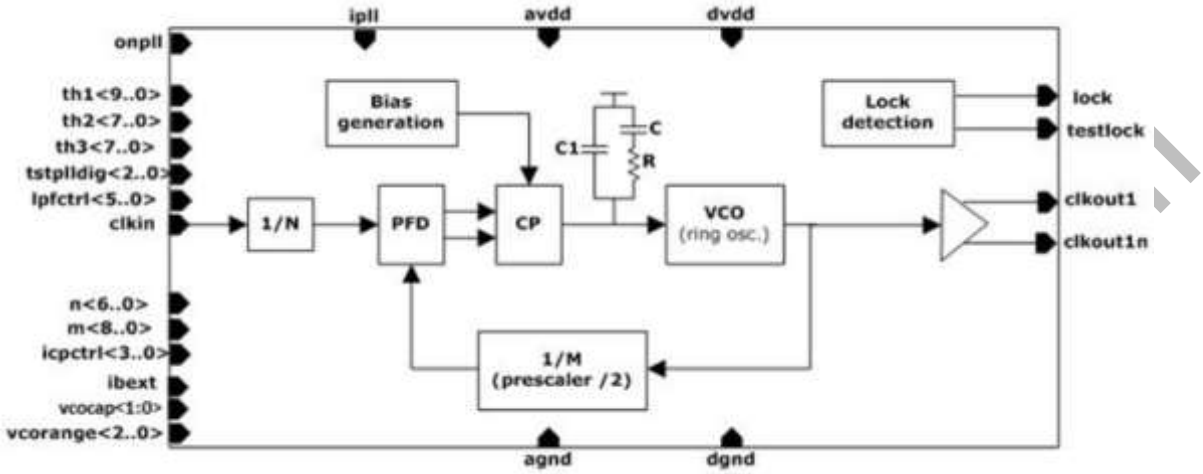


Fig. 7-5 PLL System-Level Block Diagram

The VCO output frequency is divided by a programmable counter (M) before being compared to a crystal-based, reference frequency by means of a three-state phase-frequency detector (PFD) with no dead-zone. The digital PFD commands a charge-pump (CP), which in turn delivers charge to, or extracts charge from, the loop filter. The output clock frequency is equal to the VCO operating frequency.
The integrated loop filter is a second-order low-pass filter with a pole at the origin and pole-zero compensation that stabilizes the PLL. The differential VCO architecture, together with a large loop bandwidth, ensures that there is little internal PLL phase noise.

Output Frequency

The VCO oscillating frequency is a function of the input reference frequency and the multiplication and division ratios. It can be calculated as follows:

Where:

- – frequency output, full-rate clock used for bit serialization
- M – Feedback Multiplication Ratio
- N – Input Frequency Division Ratio

The VCO ranges are selected as shown in following table.

Table 7-3 VCO Ranges

vcorange<2..0>	VCO Frequency (MHz)
000	80-200
001	200-300
010	300-500
011	500-700
100	700-900
101	900-1100
110	1100-1300
111	1300-1500

is the output, full-rate clock used for bit serialization. A 1500 Mbps bit rate on the data lanes assumes to be equal to 1500 MHz.
Finally, the output clock frequency is derived from a programmable division of the VCO frequency:

The following limit applies:
Normal mode

PLL Lock Detector

The PLL has an internal digital circuit that performs lock detection. This internal digital block uses two counters to detect the phase difference between clkin and fbclk.
Where:

fbclk – VCO frequency (clkout1) divided by M/2

The first counter is clocked at the VCO frequency (clkout1) and counts the phase error between clkin and fbclk.
Where:

phase_error_max < ((th1+1)/m)*360°

The second counter is clocked when the phase error is within the specified phase error. The lock condition is reached when th2 consecutive cycles of phase error become smaller than the maximum phase error.
The same principle applies for an unlock condition after a lock condition occurs. However, the second counter has a th3 threshold that declares the PLL unlock. The unlock condition is reached when th3 consecutive cycles of phase error become bigger than the maximum phase error max.

PLL Programming

Table 7-4 Division Ratios for the Attachable PLL

m<8..0> = M - 1	M	n<6..0> = N - 1	N
9'h1	2	7'h0	1
9'h3	4	7'h1	2
...	...	...	...
9'h127	296	7'h61	98
9'h129	298	7'h62	99
9'h12B	300	7'h63	100

Some combinations of N and M are not allowed, since they violate the limits of operation of the VCO or the minimum allowed comparison frequency. Due to the use of a "by 2 pre-scaler," the range of the feedback multiplication value M is limited to even division numbers. These N and M values should be programmed on the meaningful control registers Test Code: 0x17 and Test Code: 0x18.
To ensure proper operation of the PLL, the loop bandwidth should be configured depending on the selected frequency. The control over the CP current (icpctrl[3:0]), the LPF characteristics (lpfctrl[5..0]), and vcorange control signals is granted. Following table presents the bits correspondence.

Table 7-5 PLL CP and LPF Control Bits

VCO fmin (MHz)	VCO fmax (MHz)	vcorange<2..0>	icpctrl<3..0>	lpfctrl<5:0>	vcocap<1:0>
80	110	000	0001	000010	00
110	150	000	0001	000001	00
150	200	000	1001	000010	00
200	250	001	0010	000010	00
250	300	001	1001	000100	00
300	400	010	0001	000001	00
400	500	010	0110	000100	00
500	600	011	0110	001000	00
600	700	011	0110	000100	00
700	900	100	0110	000100	00
900	1100	101	1011	010000	00
1100	1300	110	1011	001000	00
1300	1500	111	1011	001000	00

Following table shows an example of PLL settings for different values of hsfreqrange[5:0]. Each parameter must be programmed through its respective test code.

Table 7-6 PLL Settings for 27 MHz Reference Clock and Selectable Ranges

Range (MHz)	m[8:0]	n[6:0]	icpctrl<3..0>	lpfctrl<5..0>	vcorange<2..0>	foutc [MHz]
80-89	5	1	0001	000010	000	81
90-99	9	2	0001	000010	000	90
100-109	11	2	0001	000010	000	108
110-129	13	2	0001	000001	000	126
130-139	9	1	0001	000001	000	135
140-149	15	2	0001	000001	000	144
150-169	17	2	1001	000010	000	162
170-179	19	2	1001	000010	000	180
180-199	21	2	1001	000010	000	198
200-219	23	2	0010	000010	001	216
220-239	25	2	0010	000010	001	234
240-249	17	1	0010	000010	001	243
250-269	29	2	1001	000100	001	270
270-299	21	1	1001	000100	001	297
300-329	23	1	0001	000001	010	324
330-359	39	2	0001	000001	010	360
360-399	43	2	0001	000001	010	396
400-449	49	2	0110	000100	010	450
450-499	17	0	0110	000100	010	486
500-549	59	2	0110	001000	011	540
550-599	43	1	0110	001000	011	594
600-649	23	0	0110	000100	011	648
650-699	75	2	0110	000100	011	684
700-749	81	2	0110	000100	100	738
750-799	57	1	0110	000100	100	783
800-849	93	2	0110	000100	100	846
850-899	99	2	0110	000100	100	900
900-949	69	1	1011	010000	101	945
950-999	73	1	1011	010000	101	999
1000-1049	115	2	1011	010000	101	1044
1050-1099	39	0	1011	010000	101	1080
1100-1149	125	2	1011	001000	110	1134
1150-1199	131	2	1011	001000	110	1188
1200-1249	45	0	1011	001000	110	1242
1250-1299	47	0	1011	001000	110	1296
1349	149	2	1011	001000	111	1350
1350-1399	153	2	1011	001000	111	1386
1400-1449	159	2	1011	001000	111	1440
1450-1500	165	2	1011	001000	111	1494

Notes:

a. m = M-1 represents the feedback divider, the control word in decimal notation.

b. n = N-1 represents the input divider, the control word in decimal notation.

c. foutc is the frequency visible at the clkout1 and clkout1n PLL clock outputs.

To correctly configure the PLL, the following parameters should be set using the test codes.

- PLL Input Divider Ratio (N): Test Code 0x17
- PLL Loop Divider Ratio (M): Test Code 0x18

  Make the previously configured N and M factors effective: Register 0x19: TESTDIN[5:4] = 2'b11
- VCO Control (vcorange and vcocap): Test Code 0x10: TESTDIN[5:1]

Make the previously configured vcorange and vcocap factors effective: Test Code 0x10 : TESTDIN[7] =1'b1

- PLL Control (icpctrl): Test Code 0x11: TESTDIN[3:0]
- PLL Control (lpfctrl): Test Code 0x12: TESTDIN[5:0]

Make the previously configured icpctrl and lpfctrl factors effective: Test Code 0x12: TESTDIN[7:6] = 2'b11

PLL Programming Examples

This section illustrates PLL configurations using a REFCLK of 27 MHz, and 24 MHz for an operating bit rate at 324 MHz and 756 MHz.

- REFCLK = 27 MHz

Assuming a REFCLK equal to 27 MHz, the following are cases for an operating bit rate at 324 MHz and 756 MHz.

- Case 1 (324 MHz)

  To get an operating frequency range of 324 MHz, configure the following:
  1. hsfreqrange[5:0] = 000101 (0x44 = 0x0A, listed in Table 5-1 on page 67)
  2. PLL Input Divider Ratio (N) programmed: 0x17 = 0x01
  N = 2
  3. PLL Loop Divider Ratio (M) programmed: 0x18 = 0x17 (LSB)
  4. PLL Loop Divider Ratio (M) programmed: 0x18 = 0x80 (MSB)
  M = 24
  Make the previously configured N and M factors effective: 0x19 = 0x30
  5. VCO Control (vcorange and vcocap) programmed: 0x10 = 0x91
  VCO range= 3'b010
  Vcocap = 2'b00
  6. PLL Control (icpctrl) programmed: 0x11=0x01
  icpctrl = 1101 (14 μA)
  7. PLL Control (icpctrl) programmed and allow 0x11 to be effective: 0x12= 0xC1
  icpctrl = 010000 (6 KΩ)

- Case 2 (756 MHz)

  To get an operating frequency range of 756 MHz, configure the following:
  1. hsfreqrange[5:0] = 011001 (0x44 = 0x32, listed in Table 5-1 on page 65)
  2. PLL Input Divider Ratio (N) programmed: 0x17 = 0x01
  N = 2
  3. PLL Loop Divider Ratio (M) programmed: 0x18 = 0x17 (LSB)
  4. PLL Loop Divider Ratio (M) programmed: 0x18 = 0x81 (MSB)
  M= 56
  Make the previously configured N and M factors effective: 0x19 = 0x30
  5. VCO Control (vcorange and vcocap) programmed: 0x10 = 0xA1
  VCO range= 3'b100
  Vcocap = 2'b00
  6. PLL Control (icpctrl) programmed: 0x11=0x06
  icpctrl = 4'b1111 (16 μA)
  7. PLL Control (icpctrl) programmed and allow 0x11 to be effective: 0x12 = 0xC4
  icpctrl = 6'b000100 (7 KΩ)

- REFCLK = 24 MHz

Assuming a REFCLK not equal to 27 MHz—that is, assuming 24 MHz—the following is a case for an operating bit rate at 324 MHz:

- Case (324MHz)

  To get an operation frequency range of 324 MHz, configure the following:
  1. hsfreqrange[5:0] = 000101 (0x44 = 0x0A, listed in Table 5-1 on page 65)
  2. PLL Input Divider Ratio (N) programmed: 0x17 = 0x01
  N = 2
  3. PLL Loop Divider Ratio (M) programmed: 0x18 = 0x1A (LSB)
  4. PLL Loop Divider Ratio (M) programmed: 0x18 = 0x80 (MSB)
  M = 27
  Make the previously configured N and M factors effective: 0x19 = 0x30
  5. VCO Control (vcorange and vcocap) programmed: 0x10 = 0x91

VCO range= 3'b010
Vcocap = 2'b00
6. PLL Control (icpctrl) programmed: 0x11=0x01
icpctrl = 4'b1101 (14 μA)
7. PLL Control (icpctrl) programmed and allow 0x11 to be effective: 0x12= 0xC1
Icpctrl = 6'b010000 (6 KΩ)

Chapter 8 Pulse Width Modulation (PWM)

8

8.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.
The PWM Module supports the following features:

- 4-built-in PWM channels
- Configurable to operate in capture mode
  - Measures the high/low polarity effective cycles of this input waveform
  - Generates a single interrupt at the transition of input waveform polarity
  - 32-bit high polarity capture register
  - 32-bit low polarity capture register
  - 32-bit current value register
  - The capture result of channel 3 can be stored in a FIFO. The depth of FIFO is 8, and the data in FIFO can be read through DMA. It also supports timeout interrupt when the data in FIFO has not been read in a time threshold.
- Configurable to operate in continuous mode or one-shot mode
  - 32-bit period counter
  - 32-bit duty register
  - 32-bit current value register
  - Configurable PWM output polarity in inactive state and duty period pulse polarity
  - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
  - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
  - 8-bit repeat counter for one-shot operation. One-shot operation will produce N + 1 periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
  - Continuous mode generates the waveform continuously, and does not generates any interrupts
- pre-scaled operation to bus clock and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

8.2 Block Diagram

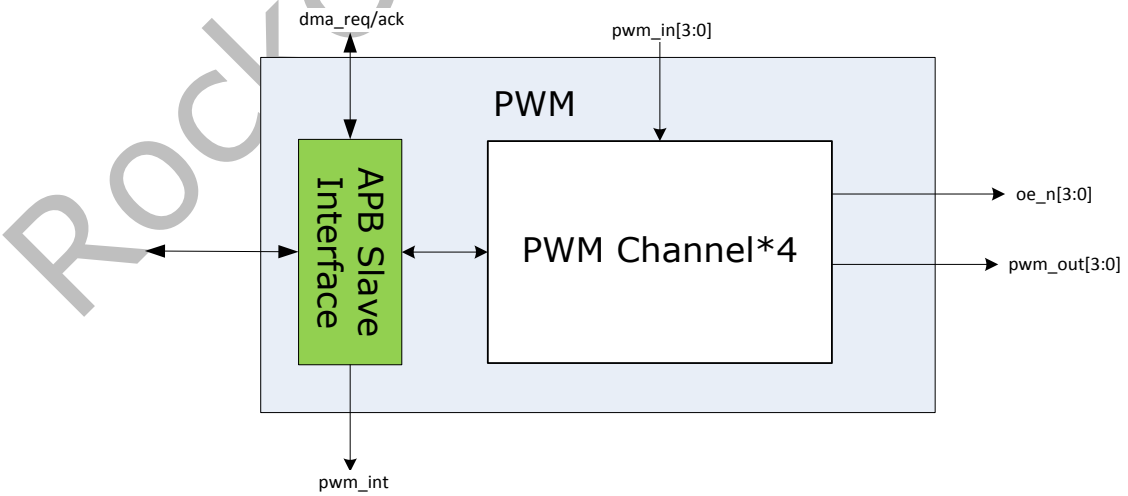


Fig. 8-1 PWM Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface with 32-bit bus width, and asserts the active-high level interrupt. PWM only supports one interrupt output, please refer to interrupt register to know the raw interrupt status when an interrupt is asserted.

PWM Channel is the control logic of PWM module, and controls the operation of PWM module according to the configured working mode.

8.3 Function Description

The PWM supports three operation modes: capture mode, one-shot mode and continuous mode. For the one-shot mode and the continuous mode, the PWM output can be configured as the left-aligned mode or the center-aligned mode.

8.3.1 Capture mode

The capture mode is used to measure the PWM channel input waveform high/low effective cycles with the PWM channel clock, and asserts an interrupt when the polarity of the input waveform changes. The number of the high effective cycles is recorded in the PWMx_PERIOD_HPC register, while the number of the low effective cycles is recorded in the PWMx_DUTY_LPC register.

Notes: the PWM input waveform is doubled buffered when the PWM channel is working in order to filter unexpected shot-time polarity transition, and therefore the interrupt is asserted several cycles after the input waveform polarity changes, and so does the change of the values of PWMx_PERIOD_HPC and PWMx_DUTY_LPC.

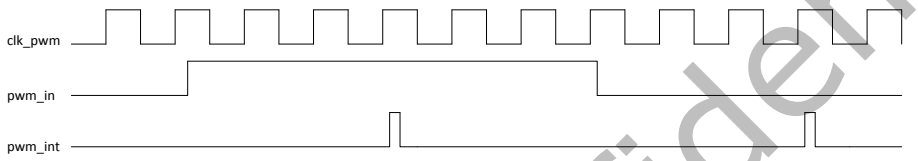


Fig. 8-2 PWM Capture Mode

8.3.2 Continuous mode

The PWM channel generates a series of the pulses continuously as expected once the channel is enabled with continuous mode.

In the continuous mode, the PWM output waveforms can be in one form of the two output mode: left-aligned mode or center-aligned mode.

For the left-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. After the period number (PWMx_PERIOD_HPC) is reached, the output is again switched to the opposite polarity to start another period of desired pulse.

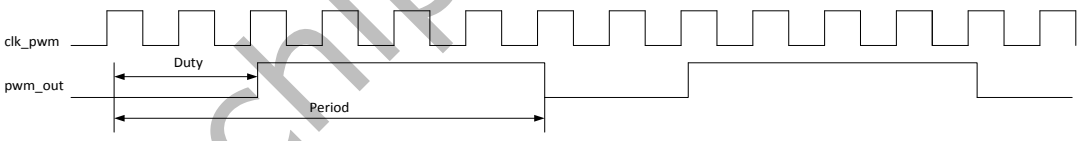


Fig. 8-3 PWM Continuous Left-aligned Output Mode

For the center-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once one half of duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. Then if there is one half of duty cycle left for the whole period, the output is again switched to the opposite polarity. Finally after the period number (PWMx_PERIOD_HPC) is reached, the output starts another period of desired pulse.

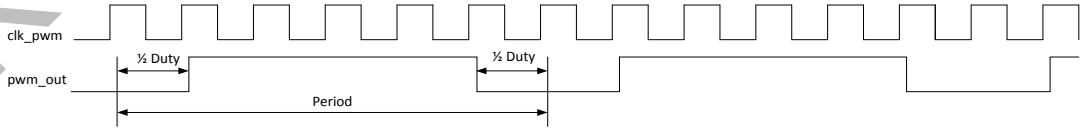


Fig. 8-4 PWM Continuous Center-aligned Output Mode

Once disable the PWM channel, the channel stops generating the output waveforms and output polarity is fixed as the configured inactive polarity (PWMx_CTRL.inactive_pol).

8.3.3 One-shot mode

Unlike the continuous mode, the PWM channel generates the output waveforms within the configured periods (PWM_CTRL.rpt + 1), and then stops. At the same times, an interrupt is asserted to inform that the operation has been finished.

There are also two output modes for the one-shot mode: the left-aligned mode and the

center-aligned mode.

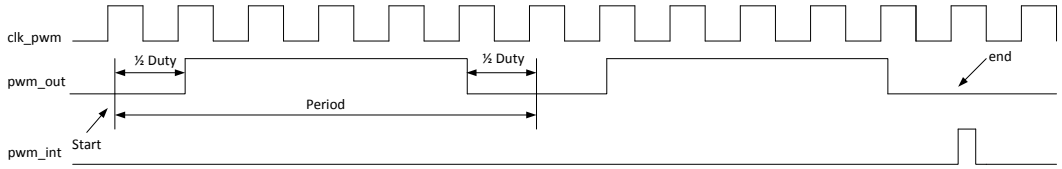


Fig. 8-5 PWM One-shot Center-aligned Output Mode

8.4 Register Description

8.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002c	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
PWM_PWM3_DUTY_LPR	0x0038	W	0x00000000	PWM Channel 3 Duty Register/Low Polarity Capture Register
PWM_PWM3_CTRL	0x003c	W	0x00000000	PWM Channel 3 Control Register
PWM_INTSTS	0x0040	W	0x00000000	Interrupt Status Register
PWM_INT_EN	0x0044	W	0x00000000	Interrupt Enable Register

Name	Offset	Size	Reset Value	Description
PWM_PWM_FIFO_CTRL	0x0050	W	0x00000000	PWM Channel 3 FIFO Mode Control Register
PWM_PWM_FIFO_INTSTS	0x0054	W	0x00000000	FIFO Interrupts Status Register
PWM_PWM_FIFO_TOUTTHR	0x0058	W	0x00000000	FIFO Timeout Threshold Register
PWM_PWM_FIFO	0x0060 ~0x007C	W	0x00000000	FIFO Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

8.4.2 Detail Register Description

PWM_PWM0_CNT
Address: Operational Base + offset (0x0000)
PWM Channel 0 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_PERIOD_HPR
Address: Operational Base + offset (0x0004)
PWM Channel 0 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_DUTY_LPR
Address: Operational Base + offset (0x0008)
PWM Channel 0 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_CTRL
Address: Operational Base + offset (0x000c)
PWM Channel 0 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.

Bit	Attr	Reset Value	Description
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM1_CNT
Address: Operational Base + offset (0x0010)
PWM Channel 1 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_PERIOD_HPR
Address: Operational Base + offset (0x0014)

PWM Channel 1 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_DUTY_LPR

Address: Operational Base + offset (0x0018)

PWM Channel 1 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_CTRL

Address: Operational Base + offset (0x001c)

PWM Channel 1 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM2_CNT
Address: Operational Base + offset (0x0020)
PWM Channel 2 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_PERIOD_HPR
Address: Operational Base + offset (0x0024)
PWM Channel 2 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_DUTY_LPR
Address: Operational Base + offset (0x0028)
PWM Channel 2 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_CTRL
Address: Operational Base + offset (0x002c)
PWM Channel 2 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.

Bit	Attr	Reset Value	Description
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM3_CNT
Address: Operational Base + offset (0x0030)
PWM Channel 3 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_PERIOD_HPR
Address: Operational Base + offset (0x0034)

PWM Channel 3 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_DUTY_LPR

Address: Operational Base + offset (0x0038)

PWM Channel 3 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_CTRL

Address: Operational Base + offset (0x003c)

PWM Channel 3 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_INTSTS

Address: Operational Base + offset (0x0040)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RO	0x0	CH3_Pol Channel 3 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 3 input waveform. Otherwise, please refer to PWM3_PERIOD_LPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit.
10	RO	0x0	CH2_Pol Channel 2 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 2 input waveform. Otherwise, please refer to PWM2_PERIOD_LPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.
9	RO	0x0	CH1_Pol Channel 1 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 1 input waveform. Otherwise, please refer to PWM1_PERIOD_LPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.
8	RO	0x0	CH0_Pol Channel 0 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	CH3_IntSts Channel 3 Interrupt Status 0: Channel 3 Interrupt not generated 1: Channel 3 Interrupt generated
2	RW	0x0	CH2_IntSts Channel 2 Interrupt Status 0: Channel 2 Interrupt not generated 1: Channel 2 Interrupt generated
1	RW	0x0	CH1_IntSts Channel 1 Interrupt Status 0: Channel 1 Interrupt not generated 1: Channel 1 Interrupt generated
0	RW	0x0	CH0_IntSts Channel 0 Raw Interrupt Status 0: Channel 0 Interrupt not generated 1: Channel 0 Interrupt generated

PWM_INT_EN
Address: Operational Base + offset (0x0044)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	CH3_Int_en Channel 3 Interrupt Enable 0: Channel 3 Interrupt disabled 1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en Channel 2 Interrupt Enable 0: Channel 2 Interrupt disabled 1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en Channel 1 Interrupt Enable 0: Channel 1 Interrupt disabled 1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en Channel 0 Interrupt Enable 0: Channel 0 Interrupt disabled 1: Channel 0 Interrupt enabled

PWM_PWM_FIFO_CTRL
Address: Operational Base + offset (0x0050)
PWM Channel 3 FIFO Mode Control Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	timeout_en FIFO Timeout Enable
8	RW	0x0	dma_mode_en DMA Mode Enable 1'b1: enable 1'b0: disable
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost Full Watermark Level
3	RW	0x0	watermark_int_en Watermark Full Interrupt
2	RW	0x0	overflow_int_en FIFO Overflow Interrupt Enable When high, an interrupt asserts when the channel 3 FIFO is overflow.
1	RW	0x0	full_int_en FIFO Full Interrupt Enable When high, an interrupt asserts when the channel 3 FIFO is full.
0	RW	0x0	fifo_mode_sel FIFO MODE Sel When high, PWM FIFO mode is activated

PWM_PWM_FIFO_INTSTS
Address: Operational Base + offset (0x0054)
FIFO Interrupts Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	fifo_empty_status FIFO Empty Status This bit indicates the FIFO is empty
3	W1C	0x0	timeout_intsts Timeout Interrupt
2	W1C	0x0	fifo_watermark_full_intsts FIFO Watermark Full Interrupt Status This bit indicates the FIFO is Watermark Full
1	W1C	0x0	fifo_overflow_intsts FIFO Overflow Interrupt Status This bit indicates the FIFO is overflow
0	W1C	0x0	fifo_full_intsts FIFO Full Interrupt Status This bit indicates the FIFO is full

PWM_PWM_FIFO_TOUTTHR
Address: Operational Base + offset (0x0058)
FIFO Timeout Threshold Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RO	0x00000	timeout_threshold FIFO Timeout Value(unit pwmclk)

PWM_PWM_FIFO
Address: Operational Base + offset (0x0060~0x007C)
FIFO Register

Bit	Attr	Reset Value	Description
31	RW	0x0	pol Polarity This bit indicates the polarity of the lower 31-bit counter. 0: Low 1: High
30:0	RO	0x00000000	cycle_cnt High/Low Cycle Counter This 31-bit counter indicates the effective cycles of high/low waveform.

8.5 Interface Description

Table 8-1 PWM Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
PWM0	I/O	IO_PWM0_VOP0pwm_VOP1pwm_GPIO1830gpio4c2	GRF_GPIO4C_IOMUX[5:4]=2'b01
PWM1	I/O	IO_PWM1_GPIO1830gpio4c6	GRF_GPIO4C_IOMUX[13:12]=2'b01
PWM2	I/O	IO_PWM2_PMU1830gpio1c3	GRF_GPIO1C_IOMUX[7:6]=2'b01
PWM3	I/O	IO_PWMA3_PMUdebug4_PMU18gpio0a6	GRF_GPIO0A_IOMUX[13:12]=2'b01 PMUGRF_SOC_CON0[5]=1'b0
		IO_PWMB3_PMU1830gpio1b6	GRF_GPIO1B_IOMUX[13:12]=2'b01 PMUGRF_SOC_CON0[5]=1'b1

Notes: I=input, O=output, I/O=input/output.

8.6 Application Notes

8.6.1 PWM Capture Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Configure the channel to work in the capture mode.
4. Enable the INT_EN.chx_int_en to enable the interrupt generation.
5. Enable the channel by writing '1' to PWMx_CTRL.pwm_en bit to start the channel.
6. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status. If the corresponding polarity flag is set, turn to PWMx_PERIOD_HPC register to know the

effective high cycles of input waveforms, otherwise turn to PWMx_DUTY_LPC register to know the effective low cycles.
7. Write '0' to PWMx_CTRL.pwm_en to disable the channel.

8.6.2 PWM Capture DMA Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Configure the channel 3 to work in the capture mode.
4. Configure the PWM_FIFO_CTRL.dma_mode_en and PWM_FIFO_CTRL.fifo_mode_sel to enable the DMA mode. Configure PWM_FIFO_CTRL.almost_full_watermark at appropriate value.
5. Configure DMAC_BUS to tansfer data from PWM to DDR.
6. Enable the channel by writing '1' to PWMx_CTRL.pwm_en bit to start the channel.
7. When an dma_req is asserted, DMAC_BUS transfer the data of effective high cycles and low cycles of input waveforms to DDR.
8. Write '0' to PWMx_CTRL.pwm_en to disable the channel.

8.6.3 PWM One-shot Mode/Continuous Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Choose the output mode by setting PWMx_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWMx_CTRL.duty_pol and PWMx_CTRL.inactive_pol.
4. Set the PWMx_CTRL.rpt if the channel is desired to work in the one-shot mode.
5. Configure the channel to work in the one-shot mode or the continuous mode.
6. Enable the INT_EN.chx_int_en to enable the interrupt generation if if the channel is desired to work in the one-shot mode.
7. If the channel is working in the one-shot mode, an interrupt is asserted after the end of operation, and the PWMx_CTRL.pwm_en is automatically cleared. Whatever mode the channel is working in, write '0' to PWMx_CTRL.pwm_en bit to disable the PWM channel.

8.6.4 Low-power mode

Setting PWMx_CTRL.lp_en to '1' makes the channel enter the low-power mode. When the PWM channel is inactive, the APB bus clock to the clock prescale module is gated in order to reduce the power consumption. It is recommended to disable the channel before entering the low-power mode, and quit the low-power mode before enabling the channel.

8.6.5 Other notes

When the channel is active to produce waveforms, it is free to program the PWMx_PERIOD_HPC and PWMx_DUTY_LPC register. The change will not take effect immediately until the current period ends.
An active channel can be changed to another operation mode without disable the PWM channel. However, during the transition of the operation mode there may be some irregular output waveforms. So does changing the clock division factor when the channel is active.

Chapter 9 UART

9

9.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.
UART Controller supports the following features:

- Support 5 independent UART controller: UART0, UART1, UART2, UART3, UART4
- All contain two 64Bytes FIFOs for data receive and transmit
- UART0/UART3 support auto flow-control
- Support bit rates 115.2Kbps, 460.8Kbps, 921.6Kbps, 1.5Mbps, 3Mbps, 4Mbps
- Support programmable baud rates, even with non-integer clock divider
- Standard asynchronous communication bits (start, stop and parity)
- Support interrupt-based or DMA-based mode
- Support 5-8 bits width transfer

9.2 Block Diagram

This section provides a description about the functions and behavior under various conditions. The UART Controller comprises with:

- AMBA APB interface
- FIFO controllers
- Register block
- Modem synchronization block and baud clock generation block
- Serial receiver and serial transmitter

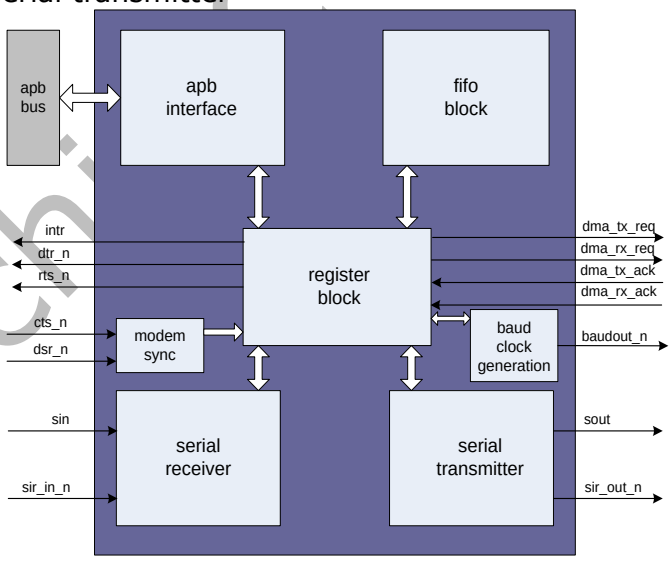


Fig. 9-1 UART Architecture

APB INTERFACE
The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus widths of 8, 16, and 32 bits.
Register block
Be responsible for the main UART functionality including control, status and interrupt generation.
Modem Synchronization block
Synchronizes the modem input signal.
FIFO block
Be responsible for FIFO control and storage (when using internal RAM) or signaling to control external RAM (when used).

Baud Clock Generator
Generates the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).
Serial Transmitter
Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.
Serial Receiver
Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

9.3 Function Description

UART (RS232) Serial Protocol
Because the serial communication is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to perform simple error checking on the received data, as shown in Figure.

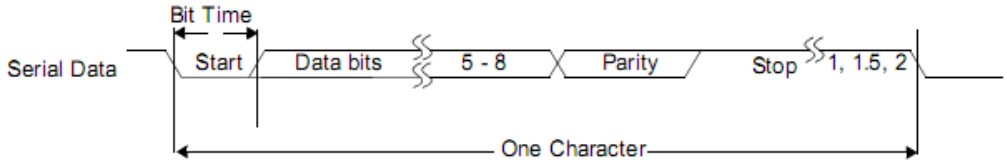


Fig. 9-2 UART Serial protocol

IrDA 1.0 SIR Protocol
The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.
Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

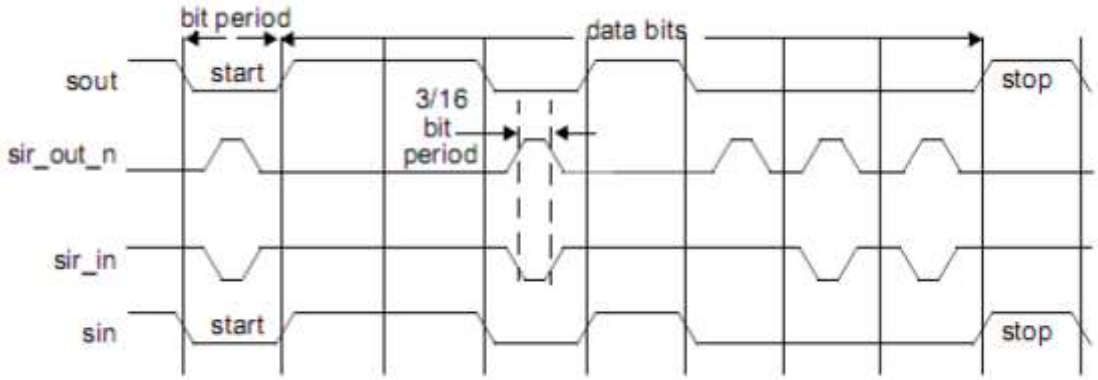


Fig. 9-3 IrDA 1.0

Baud Clock
The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit.

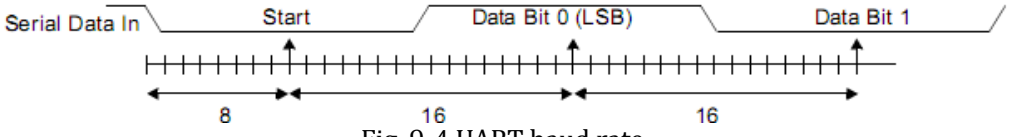


Fig. 9-4 UART baud rate

FIFO Support

1. NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2. FIFO MODE

The FIFO depth of UART0/UART1/UART2 is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The UART supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

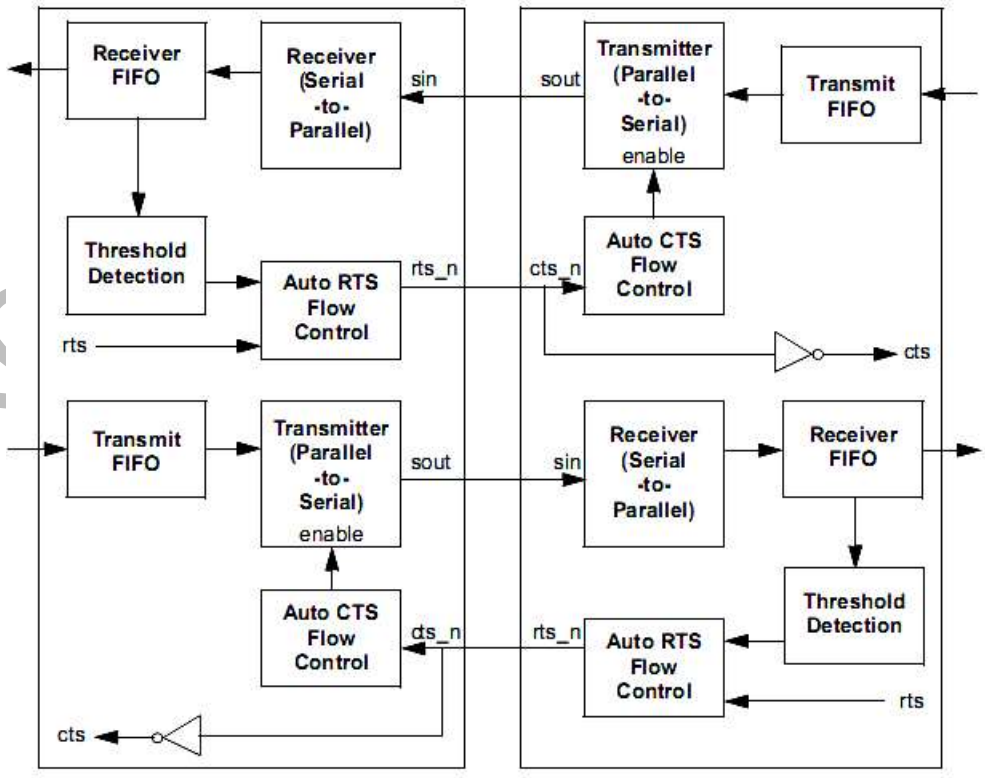


Fig. 9-5 UART Auto flow control block diagram

Auto RTS – Becomes active when the following occurs:
- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

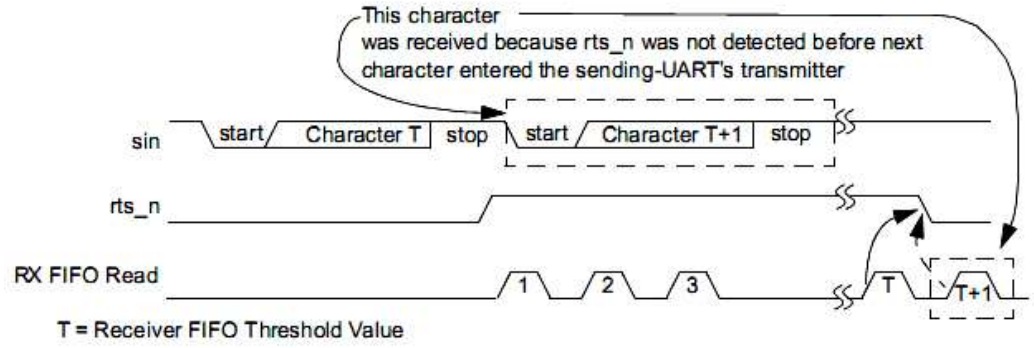


Fig. 9-6 UART AUTO RTS TIMING

Auto CTS – becomes active when the following occurs:
- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit is set)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit is not set)

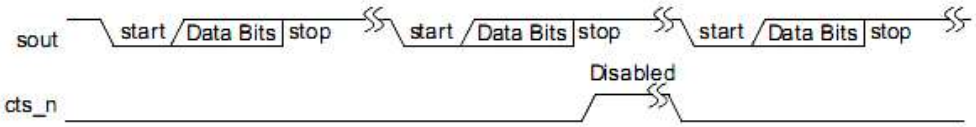


Fig. 9-7 UART AUTO CTS TIMING

9.4 Register Description

This section describes the control/status registers of the design. There are 3 UARTs in RK3228, and each one has its own base address.

9.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000000	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000000	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x006c	W	0x00000000	Shadow Transmit Holding Register

Name	Offset	Size	Reset Value	Description
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000000	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x0330372a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

9.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)

Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_input Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

UART_THR

Address: Operational Base + offset (0x0000)
Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_output Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLL
Address: Operational Base + offset (0x0000)
Divisor Latch (Low)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_L Lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

UART_DLH
Address: Operational Base + offset (0x0004)
Divisor Latch (High)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_H Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.

UART_IER
Address: Operational Base + offset (0x0004)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
6:4	RO	0x0	reserved
3	RW	0x0	modem_status_int_en Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled
2	RW	0x0	receive_line_status_int_en Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt.
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

UART_IIR
Address: Operational Base + offset (0x0008)
Interrupt Identification Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled

Bit	Attr	Reset Value	Description
5:4	RO	0x0	reserved
3:0	RO	0x0	int_id Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

UART_FCR
Address: Operational Base + offset (0x0008)
FIFO Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	WO	0x0	rcvr_trigger RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than ful
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full

Bit	Attr	Reset Value	Description
3	WO	0x0	dma_mode DMA Mode This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected . 0 = mode 0 1 = mode 11100 = character timeout.
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	WO	0x0	fifo_en FIFO Enable. FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_LCR

Address: Operational Base + offset (0x000c)

Line Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	div_lat_access Divisor Latch Access Bit. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.

Bit	Attr	Reset Value	Description
6	RW	0x0	break_ctrl Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	RO	0x0	reserved
4	RW	0x0	even_parity_sel Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.
3	RW	0x0	parity_en Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled
2	RW	0x0	stop_bits_num Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	data_length_sel Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

UART_MCR
Address: Operational Base + offset (0x0010)
Modem Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	sir_mode_en SIR Mode Enable. SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode . 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled
5	RW	0x0	auto_flow_ctrl_en Auto Flow Control Enable. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
4	RW	0x0	loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0)
2	RW	0x0	out1 OUT1 This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 1'b0: out2_n de-asserted (logic 1) 1'b1: out2_n asserted (logic 0)

Bit	Attr	Reset Value	Description
1	RW	0x0	req_to_send Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data.
0	RW	0x0	data_terminal_ready Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0)

UART_LSR
Address: Operational Base + offset (0x0014)
Line Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	receiver_fifo_error Receiver FIFO Error bit. This bit is relevant FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO
6	RO	0x0	trans_empty Transmitter Empty bit. Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	RO	0x0	trans_hold_reg_empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.

Bit	Attr	Reset Value	Description
4	RO	0x0	break_int Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2	RO	0x0	parity_eror Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready

UART_MSR
Address: Operational Base + offset (0x0018)
Modem Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	data_carrior_detect Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n.
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n.
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.

Bit	Attr	Reset Value	Description
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n.
3	RO	0x0	delta_data_carrier_detect Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.
2	RO	0x0	trailing_edge_ring_indicator Trailing Edge of Ring Indicator. Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.
1	RO	0x0	delta_data_set_ready Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.
0	RO	0x0	delta_clear_to_send Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR
Address: Operational Base + offset (0x001c)
Scratchpad Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space.

UART_SRBR
Address: Operational Base + offset (0x0030)
Shadow Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR
Address: Operational Base + offset (0x006c)
Shadow Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_thr This is a shadow register for the THR.

UART_FAR
Address: Operational Base + offset (0x0070)
FIFO Access Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	fifo_access_test_en This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled

UART_TFR
Address: Operational Base + offset (0x0074)
Transmit FIFO Read

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	trans_fifo_read Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO.

UART_RFW
Address: Operational Base + offset (0x0078)
Receive FIFO Write

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	receive_fifo_framing_error Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
8	WO	0x0	receive_fifo_parity_error Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
7:0	WO	0x00	receive_fifo_write Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR.

UART_USR
Address: Operational Base + offset (0x007c)
UART Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.

Bit	Attr	Reset Value	Description
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	RO	0x0	trasn_fifo_empty Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty
1	RO	0x0	trans_fifo_not_full Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	RO	0x0	uart_busy UART Busy. UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the UART is idle or inactive. 0 = UART is idle or inactive 1 = UART is busy (actively transferring data)

UART_TFL
Address: Operational Base + offset (0x0080)
Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	trans_fifo_level Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

UART_RFL
Address: Operational Base + offset (0x0084)
Receive FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	receive_fifo_level Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

UART_SRR
Address: Operational Base + offset (0x0088)
Software Reset Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	WO	0x0	uart_reset UART Reset. This asynchronously resets the UART and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS
Address: Operational Base + offset (0x008c)
Shadow Request to Send

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR
Address: Operational Base + offset (0x0090)
Shadow Break Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM
Address: Operational Base + offset (0x0094)
Shadow DMA Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).

UART_SFE
Address: Operational Base + offset (0x0098)
Shadow FIFO Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT
Address: Operational Base + offset (0x009c)
Shadow RCVR Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET
Address: Operational Base + offset (0x00a0)
Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX
Address: Operational Base + offset (0x00a4)
Halt TX

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled

UART_DMASA
Address: Operational Base + offset (0x00a8)
DMA Software Acknowledge

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_CPR
Address: Operational Base + offset (0x00f4)
Component Parameter Register
UART_CPR is UART0's own unique register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	FIFO_MODE 0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	RO	0x0	reserved
13	RO	0x0	DMA_EXTRA 0 = FALSE 1 = TRUE
12	RO	0x0	UART_ADD_ENCODED_PARAMS 0 = FALSE 1 = TRUE
11	RO	0x0	SHADOW 0 = FALSE 1 = TRUE
10	RO	0x0	FIFO_STAT 0 = FALSE 1 = TRUE
9	RO	0x0	FIFO_ACCESS 0 = FALSE 1 = TRUE
8	RO	0x0	NEW_FEAT 0 = FALSE 1 = TRUE
7	RO	0x0	SIR_LP_MODE 0 = FALSE 1 = TRUE

Bit	Attr	Reset Value	Description
6	RO	0x0	SIR_MODE 0 = FALSE 1 = TRUE
5	RO	0x0	THRE_MODE 0 = FALSE 1 = TRUE
4	RO	0x0	AFCE_MODE 0 = FALSE 1 = TRUE
3:2	RO	0x0	reserved
1:0	RO	0x0	APB_DATA_WIDTH 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

UART_UCV
Address: Operational Base + offset (0x00f8)
UART Component Version

Bit	Attr	Reset Value	Description
31:0	RO	0x0330372a	ver ASCII value for each number in the version

UART_CTR
Address: Operational Base + offset (0x00fc)
Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

9.5 Interface Description

Table 9-1 UART Interface Description

Module pin	Dir	Pad name	IOMUX
UART0 Interface			
uart0_sin	I	IO_UART0BTsin_WIFIBTgpio2c0	GRF_GPIO2C_IOMUX[1:0]=2'b01
uart0_sout	O	IO_UART0BTsout_WIFIBTgpio2c1	GRF_GPIO2C_IOMUX[3:2]=2'b01
uart0_cts_n	I	IO_UART0BTctsn_WIFIBTgpio2c2	GRF_GPIO2C_IOMUX[5:4]=2'b01
uart0_rts_n	O	IO_UART0BTrtsn_WIFIBTgpio2c3	GRF_GPIO2C_IOMUX[7:6]=2'b01
UART1 Interface			
uart1_sin	I	IO_MACtxen_UART1BBsin_GMACgpio3b4	GRF_GPIO3B_IOMUX[9:8]=2'b10
uart1_s	O	IO_MACmdio_UART1BBsout_GMACgpio	GRF_GPIO3B_IOMUX[11:10]=

Module pin	Dir	Pad name	IOMUX
out		3b5	2'b10
UART2A Interface			
uart2a_sin	I	IO_SDMMCdata0_UART2DBGAsin_SDMMCgpio4b0	GRF_GPIO4B_IOMUX[1:0]=2'b10
uart2a_sout	O	IO_SDMMCdata1_UART2DBGAsout_HDCPJTAGtrstn_SDMMCgpio4b1	GRF_GPIO4B_IOMUX[3:2]=2'b10
UART2B Interface			
uart2b_sin	I	IO_I2C3HDMIsda_UART2DBGBsin_HDMII2Csda_GPIO1830gpio4c0	GRF_GPIO4C_IOMUX[1:0]=2'b10
uart2b_sout	O	IO_I2C3HDMIscl_UART2DBGBsout_HDMII2Cscl_GPIO1830gpio4c1	GRF_GPIO4C_IOMUX[3:2]=2'b10
UART2C Interface			
uart2c_sin	I	IO_UART2DBGCsin_UARTHDCPsin_GPIO1830gpio4c3	GRF_GPIO4C_IOMUX[7:6]=2'b10
uart2c_sout	O	IO_UART2DBGCsout_UARTHDCPsout_GPIO1830gpio4c4	GRF_GPIO4C_IOMUX[9:8]=2'b10
UART3 Interface			
uart3_sin	I	IO_MACrxclk_UART3GPSsin_GMACgpio3b6	GRF_GPIO3B_IOMUX[13:12]=2'b10
uart3_sout	O	IO_MACcrs_UART3GPSsout_CIFclkoutb_GMACgpio3b7	GRF_GPIO3B_IOMUX[15:14]=2'b10
uart3_cts_n	I	IO_MACcol_UART3GPSctsn_SPDIFtxb_GMACgpio3c0	GRF_GPIO3C_IOMUX[1:0]=2'b10
uart3_rts_n	O	IO_MACtxclk_UART3GPSrtsn_GMACgpio3c1	GRF_GPIO3C_IOMUX[3:2]=2'b10
UART4 Interface			
uart4_sin	I	IO_UART4M0sin_SPI1ECrxd_PMU1830gpio1a7	PMUGRF_GPIO1A_IOMUX[15:14]=2'b01
uart4_sout	O	IO_UART4M0sout_SPI1ECtxd_PMU1830gpio1b0	PMUGRF_GPIO1B_IOMUX[1:0]=2'b01

The I/O interface of UART1 can be chosen by setting GRF_CON_IOMUX[11](uart1sel) bit, if this bit is set to 1, UART1 uses the UART11 I/O interface. The I/O interface of UART2 can be chosen by setting GRF_CON_IOMUX[8](uart2sel) bit, if this bit is set to 1, UART2 uses the UART21 I/O interface.

9.6 Application Notes

9.6.1 None FIFO Mode Transfer Flow

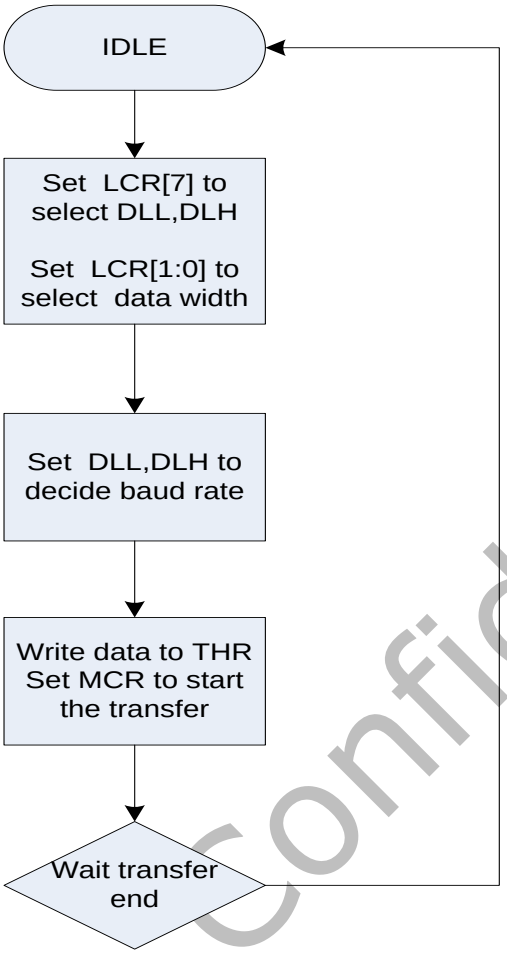


Fig. 9-8 UART none fifo mode

9.6.2 FIFO Mode Transfer Flow

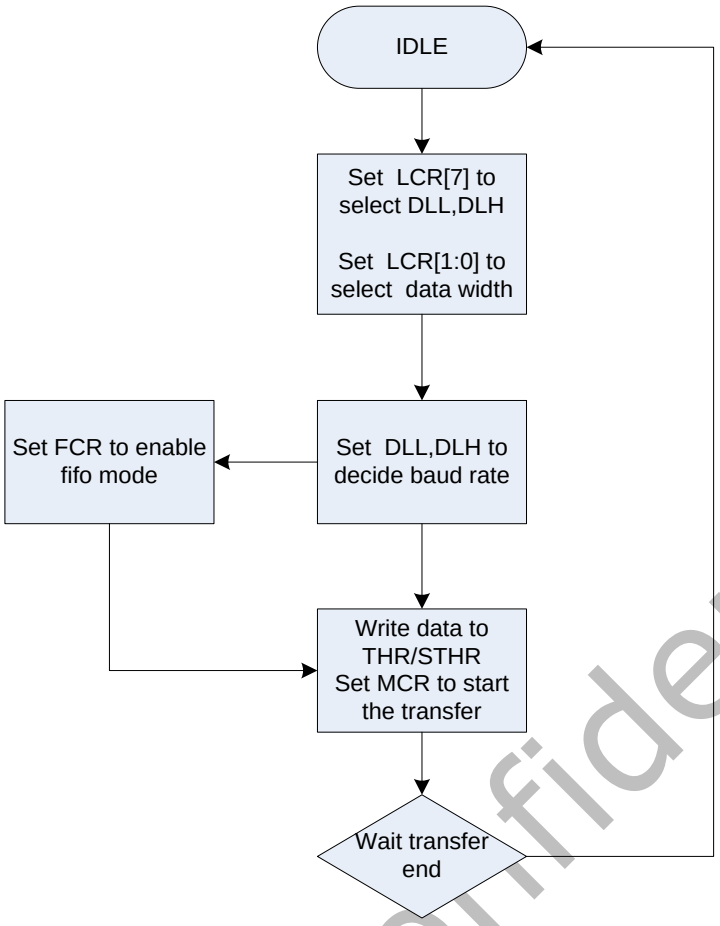


Fig. 9-9 UART fifo mode

The UART is an APB slave performing:
Serial-to-parallel conversion on data received from a peripheral device.
Parallel-to-serial conversion on data transmitted to the peripheral device.
The CPU reads and writes data and control/status information through the APB interface.
The transmitting and receiving paths are buffered with internal FIFO memories enabling up to 64-bytes to be stored independently in both transmit and receive modes. A baud rate generator can generate a common transmit and receive internal clock input. The baud rates will depend on the internal clock frequency. The UART will also provide transmit, receive and exception interrupts to system. A DMA interface is implemented for improving the system performance.

9.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.
UART0, UART1, UART2, UART3 source clocks can be selected from three PLL outputs (CODEC PLL/GENERAL PLL/USBPHY_480M). UART4 source clocks can be selected from only one PLL outputs (PMU PLL). UART clocks can be generated by 1 to 64 division of its source clock, or can be fractionally divided again, or be provided by XIN24M.

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 9-2 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 8.
460.8 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 2.
921.6 Kbps	Configure GENERAL PLL to get 648MHz clock output;

Baud Rate	Reference Configuration
	Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 1.
1.5 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 16 to get 24MHz clock; Configure UART_DLL to 1
3 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 8 to get 48MHz clock; Configure UART_DLL to 1
4 Mbps	Configure GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 6 to get 64MHz clock; Configure UART_DLL to 1

1.6.4 CTS_n and RTS_n Polarity Configurable

The polarity of cts_n and rts_n ports can be configured by GRF registers.

- grf_uart_cts_sel[*] used to configure the polarity of cts_n. Every bit for one UART.
- grf_uart_rts_sel[*] used to configure the polarity of rts_n. Every bit for one UART.
- When grf_uart_cts_sel[*] is configured as 1'b1, cts_n is high active. Otherwise, low active.
- When grf_uart_rts_sel[*] is configured as 1'b1, rts_n is high active. Otherwise, low active.

Chapter 10 GPIO

10

10.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is an APB slave device. GPIO controls the output data and direction of external I/O pads. It also can read back the data on external pads using memory-mapped registers.
GPIO supports the following features:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode

10.2 Block Diagram

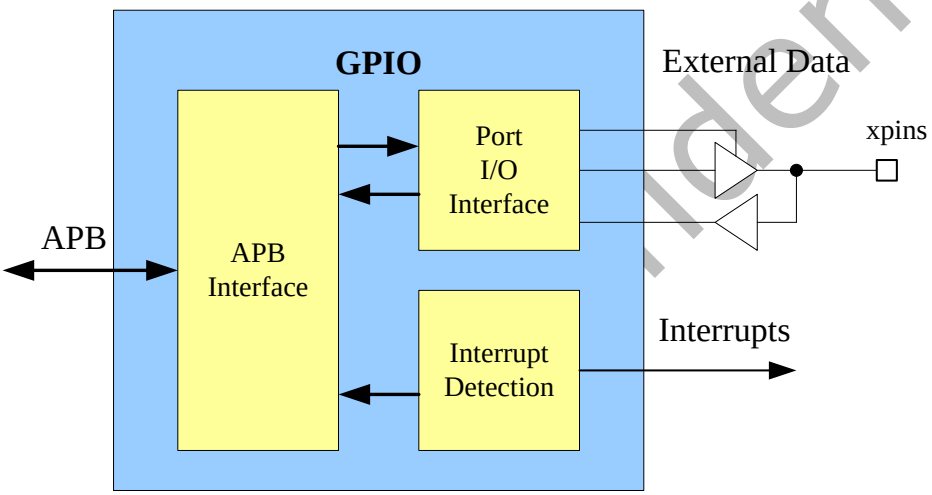


Fig. 10-1 GPIO block diagram

Block descriptions:
APB Interface
The APB Interface implements the APB slave operation. Its data bus width is 32 bits.
Port I/O Interface
External data Interface to or from I/O pads.
Interrupt Detection
Interrupt interface to or from interrupt controller.

10.3 Function Description

10.3.1 Operation

Control Mode (software)
Under software control, the data and direction control for the signal are sourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR). The direction of the external I/O pad is controlled by a write to the Porta data direction register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mapped onto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls the direction of an external I/O pad.
The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad. External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register (GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only, meaning that it cannot be written from the APB software interface.

Reading External Signals

The data on the GPIO_EXT_PORTA external signal can always be read. The data on the external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.

An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.

The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost. However, no new interrupts are generated.

For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

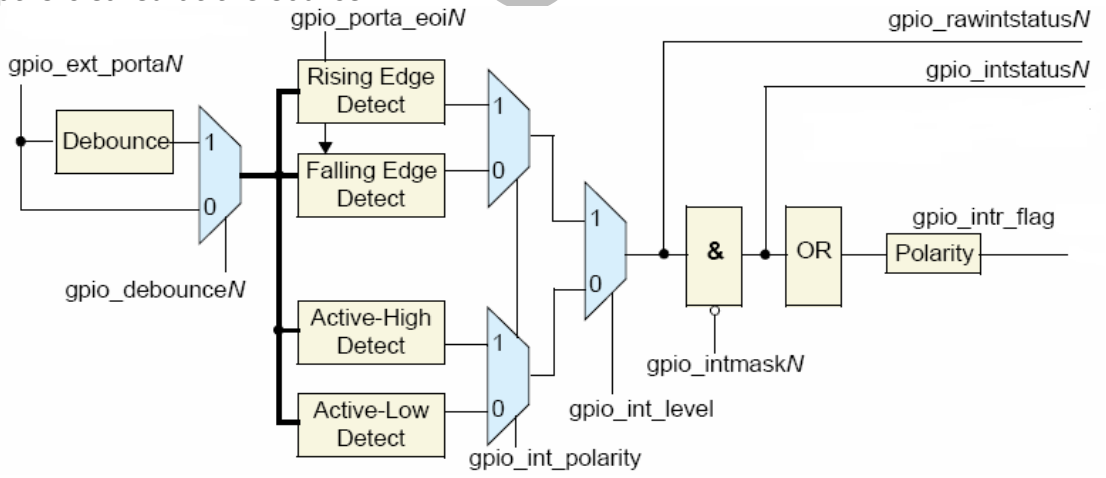


Fig. 10-2 GPIO Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When input interrupt signals are debounced using a debounce clock (pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one rising edge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization to pclk must occur for edge-detect signals. With level-sensitive interrupts, synchronization is optional and under software control (GPIO_LS_SYNC).

10.3.2 Programming

Programming Considerations

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitive interrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A in order to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on a level-sensitive interrupt.

9 GPIOs' hierarchy in the chip

GPIO0, GPIO1, GPIO2 are in PD_PERI subsystem.

10.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are 3 GPIOs (GPIO0 ~ GPIO2), and each of them has same register group. Therefore, 3 GPIOs' register groups have 3 different base addresses.

10.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003c	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004c	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

10.4.2 Detail Register Description

GPIO_SWPORTA_DR

Address: Operational Base + offset (0x0000)

Port A data register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_dr Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORTA_DDR
Address: Operational Base + offset (0x0004)
Port A data direction register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 0: Input (default) 1: Output

GPIO_INTEN
Address: Operational Base + offset (0x0030)
Interrupt enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_en Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 0: Configure Port A bit as normal GPIO signal (default) 1: Configure Port A bit as interrupt

GPIO_INTMASK
Address: Operational Base + offset (0x0034)
Interrupt mask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_mask Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 0: Interrupt bits are unmasked (default) 1: Mask interrupt

GPIO_INTTYPE_LEVEL
Address: Operational Base + offset (0x0038)
Interrupt level register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 0: Level-sensitive (default) 1: Edge-sensitive

GPIO_INT_POLARITY
Address: Operational Base + offset (0x003c)
Interrupt polarity register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_polarity Controls the polarity of edge or level sensitivity that can occur on input of Port A. 0: Active-low (default) 1: Active-high

GPIO_INT_STATUS
Address: Operational Base + offset (0x0040)
Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_status Interrupt status of Port A

GPIO_INT_RAWSTATUS
Address: Operational Base + offset (0x0044)
Raw Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Raw interrupt of status of Port A (premasking bits)

GPIO_DEBOUNCE
Address: Operational Base + offset (0x0048)
Debounce enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0: No debounce (default) 1: Enable debounce

GPIO_PORTA_EOI
Address: Operational Base + offset (0x004c)
Port A clear interrupt register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	gpio_porta_eoi Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0: No interrupt clear (default) 1: Clear interrupt

GPIO_EXT_PORTA
Address: Operational Base + offset (0x0050)
Port A external port register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_ext_porta When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.

GPIO_LS_SYNC
Address: Operational Base + offset (0x0060)
Level_sensitive synchronization enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0: No synchronization to pclk_intr (default) 1: Synchronize to pclk_intr

10.5 Interface Description

Table 10-1 GPIO interface description

Module Pin	Dir	Pad Name	IOMUX Setting
GPIO0 Interface			
gpio0_porta[6:0]	I/O	GPIO0_A[6:0]	GRF_GPIO0A_IOMUX[13:0]=14'h0
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	GRF_GPIO0B_IOMUX[15:0]=16'h0
gpio0_porta[22:16]	I/O	GPIO0_C[6:0]	GRF_GPIO0C_IOMUX[11:0]=12'h0
gpio0_porta[31:24]	I/O	GPIO0_D[7:0]	GRF_GPIO0D_IOMUX[15:4]=16'h0
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[7:0]	GRF_GPIO1A_IOMUX[15:0]=16'h0
gpio1_porta[9:8]	I/O	GPIO1_B[1:0]	GRF_GPIO1B_IOMUX[3:0]=4'h0
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[7:0]	GRF_GPIO2A_IOMUX[15:0]=16'h0
gpio2_porta[15:8]	I/O	GPIO2_B[7:0]	GRF_GPIO2B_IOMUX[15:0]=16'h0

Module Pin	Dir	Pad Name	IOMUX Setting
gpio2_porta[23:16]	I/O	GPIO2_C[7:0]	GRF_GPIO2C_IOMUX[15:0]=16'h0
gpio2_porta[31:24]	I/O	GPIO2_D[7:0]	GRF_GPIO2D_IOMUX[15:0]=16'h0
GPIO3 Interface			
gpio3_porta[4:0]	I/O	GPIO3_A[4:0]	GRF_GPIO3A_IOMUX[9:0]=10'h0
gpio3_porta[15:8]	I/O	GPIO3_B[7:0]	GRF_GPIO3B_IOMUX[15:0]=16'h0
gpio3_porta[22:16]	I/O	GPIO3_C[6:0]	GRF_GPIO3C_IOMUX[13:0]=14'h0
GPIO4 Interface			
gpio4_porta[23:16]	I/O	GPIO4_C[7:0]	GRF_GPIO4C_IOMUX[15:0]=16'h0
gpio4_porta[26:24]	I/O	GPIO4_D[2:0]	GRF_GPIO4D_IOMUX[5:0]=6'h0

10.6 Application Notes

Steps to set GPIO's direction

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction and Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Default GPIO's direction is input direction.

Steps to set GPIO's level

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction.
- Write GPIO_SWPORT_DR[x] as v to set this GPIO's value.

Steps to get GPIO's level

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Read from GPIO_EXT_PORT[x] to get GPIO's value

Steps to set GPIO as interrupt source

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Write GPIO_INTTYPE_LEVEL[x] as v1 and write GPIO_INT_POLARITY[x] as v2 to set interrupt type
- Write GPIO_INTEN[x] as 1 to enable GPIO's interrupt

Note: Please switch iomux to GPIO mode first!

Chapter 11 I2C Interface

11

11.1 Overview

The Inter-Integrated Circuit (I2C) is a two wired (SCL and SDA), bi-directional serial bus that provides an efficient and simple method of information exchange between devices. This I2C bus controller supports master mode acting as a bridge between AMBA protocol and generic I2C bus system.

I2C Controller supports the following features:

- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation

11.2 Block Diagram

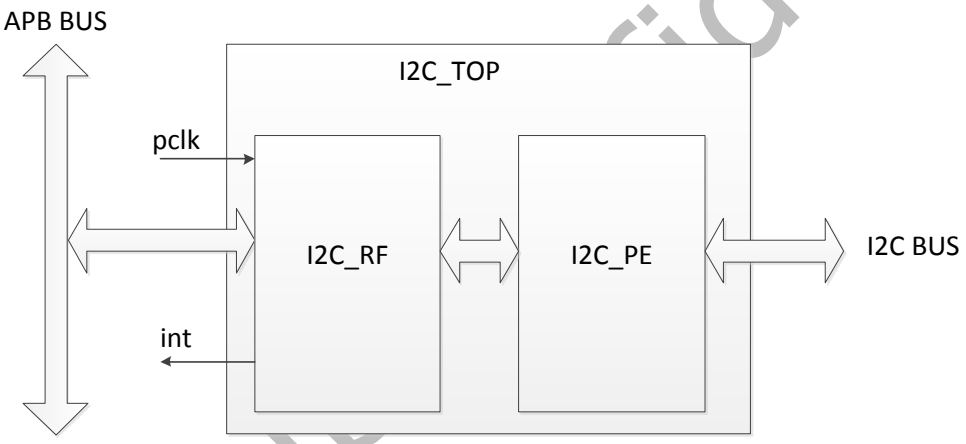


Fig. 11-1 I2C architecture

11.2.1 I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

11.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the clk_i2c.

11.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

11.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.

The I2C controller supports only Master function. It supports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 1Mbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

11.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting and configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock/clk_i2c as the working clock. The APB clock will determine the I2C bus clock, clk_i2c is the function clk, up to 200MHz. The correct register setting is subject to the system requirement.

11.3.2 Master Mode Programming

- SCL Clock
  When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:
  SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1)
  SCL = clk_i2c/ SCLK Divisor

- Data Receiver Register Access
  When the I2C controller received MRXCNT bytes data， CPU can get the data through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 bytes' data in one transaction.
  When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

- Transmit Transmitter Register
  Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 bytes' data in one transaction. The lower byte will be transmitted first.
  When MTXCNT register is written, the I2C controller will start to transmit data.

- Start Command
  Write 1 to I2C_CON[3], the controller will send I2C start command.

- Stop Command
  Write 1 to I2C_CON[4], the controller will send I2C stop command

- I2C Operation mode
  There are four i2c operation modes.
  - When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.
  - When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.
  - When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will trigger clock to read MRXCNT byte data.
  - When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

- Read/Write Command
  - When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.
  - In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].

- In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

- Master Interrupt Condition
  There are 7 interrupt bits in I2C_ISR register related to master mode.
  - Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master completed transmitting a byte.
  - Byte received finish interrupt (Bit 1): The bit is asserted when Master completed receiving a byte.
  - MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master completed transmitting MTXCNT bytes.
  - MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master completed receiving MRXCNT bytes.
  - Start interrupt (Bit 4): The bit is asserted when Master finished asserting start command to I2C bus.
  - Stop interrupt (Bit 5): The bit is asserted when Master finished asserting stop command to I2C bus.
  - NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

- Last byte acknowledge control
  - If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.
  - If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

- How to handle NAK handshake received
  - If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.
  - If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.

- I2C controller data transfer waveform
  - Bit transferring
    - Data Validity
      The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

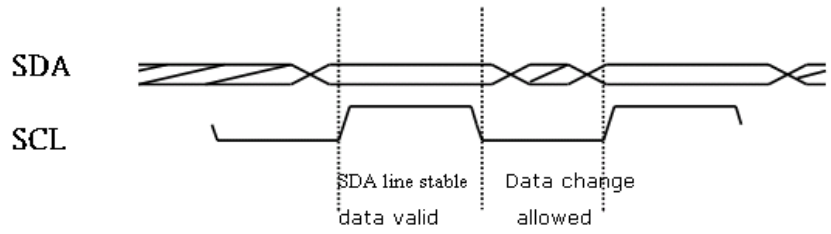


Fig. 11-2 I2C DATA Validity

    - START and STOP conditions
      START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.

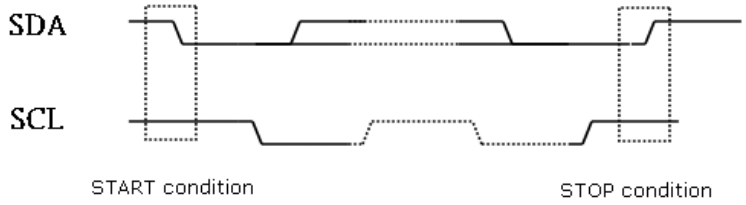


Fig. 11-3 I2C Start and stop conditions

- Data transfer
  - Acknowledge
    After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means "ACK", on the contrary, it's "NOT ACK".

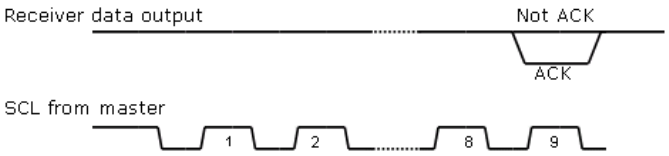


Fig. 11-4 I2C Acknowledge

  - Byte transfer
    The master own I2C bus might initiate multi byte to transfer to a slave.The transfer starts from a "START" command and ends in a "STOP"command. After every byte transfer, the receiver must reply an ACK to transmitter.

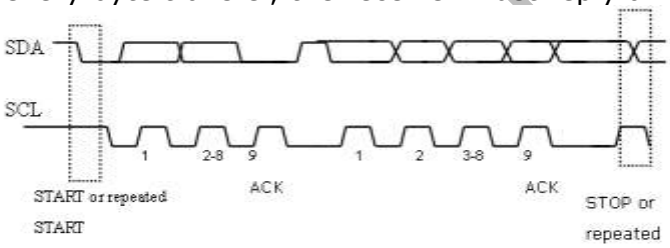


Fig. 11-5 I2C byte transfer

11.4 Register Description

11.4.1 Registers Summary Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00000000	control register
RKI2C_CLKDIV	0x0004	W	0x00000001	clock divider register
RKI2C_MRXADDR	0x0008	W	0x00000000	the slave address accessed for master rx mode
RKI2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master rx mode
RKI2C_MTXCNT	0x0010	W	0x00000000	master transmit count
RKI2C_MRXCNT	0x0014	W	0x00000000	master rx count
RKI2C_IEN	0x0018	W	0x00000000	interrupt enable register
RKI2C_IPD	0x001c	W	0x00000000	interrupt pending register
RKI2C_FCNT	0x0020	W	0x00000000	finished count
RKI2C_SCL_OE_DB	0x0024	W	0x00000020	slave hold debounce configure register
RKI2C_TXDATA0	0x0100	W	0x00000000	I2C tx data register 0

Name	Offset	Size	Reset Value	Description
RKI2C_TXDATA1	0x0104	W	0x00000000	I2C tx data register 1
RKI2C_TXDATA2	0x0108	W	0x00000000	I2C tx data register 2
RKI2C_TXDATA3	0x010c	W	0x00000000	I2C tx data register 3
RKI2C_TXDATA4	0x0110	W	0x00000000	I2C tx data register 4
RKI2C_TXDATA5	0x0114	W	0x00000000	I2C tx data register 5
RKI2C_TXDATA6	0x0118	W	0x00000000	I2C tx data register 6
RKI2C_TXDATA7	0x011c	W	0x00000000	I2C tx data register 7
RKI2C_RXDATA0	0x0200	W	0x00000000	I2C rx data register 0
RKI2C_RXDATA1	0x0204	W	0x00000000	I2C rx data register 1
RKI2C_RXDATA2	0x0208	W	0x00000000	I2C rx data register 2
RKI2C_RXDATA3	0x020c	W	0x00000000	I2C rx data register 3
RKI2C_RXDATA4	0x0210	W	0x00000000	I2C rx data register 4
RKI2C_RXDATA5	0x0214	W	0x00000000	I2C rx data register 5
RKI2C_RXDATA6	0x0218	W	0x00000000	I2C rx data register 6
RKI2C_RXDATA7	0x021c	W	0x00000000	I2C rx data register 7
RKI2C_ST	0x0220	W	0x00000000	status debug register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

11.4.2 Detail Register Description

RKI2C_CON

Address: Operational Base + offset (0x0000)
control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	version rki2c version version information
15:14	RW	0x0	stop_setup staop setup config TSU;sto = (stop_setup + 1) * T(SCL_HIGH) + Tclk_i2c
13:12	RW	0x0	start_setup start setup config TSU;sta = (start_setup + 1) * T(SCL_HIGH) + Tclk_i2c THD;sta = (start_setup + 2) * T(SCL_HIGH) - Tclk_i2c
11	RO	0x0	reserved
10:8	RW	0x0	data_upd_st SDA update point config Used to config sda change state when scl is low, used to adjust setup/hold time 4'bn:Thold = (n + 1) * Tclk_i2c Note: 0 <= n <= 5
7	RO	0x0	reserved
6	RW	0x0	act2nak operation when NAK handshake is received 1'b0: ignored 1'b1: stop transaction

Bit	Attr	Reset Value	Description
5	RW	0x0	ack last byte acknowledge control in master receive mode 1'b0: ACK 1'b1: NAK
4	RW	0x0	stop stop enable stop enable, when this bit is written to 1, I2C will generate stop signal.
3	RW	0x0	start start enable start enable, when this bit is written to 1, I2C will generate start signal.
2:1	RW	0x0	i2c_mode i2c mode select 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart --> transmit address -> receive only 2'b10: receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en i2c module enable 1'b0:not enable 1'b1:enable

RKI2C_CLKDIV
Address: Operational Base + offset (0x0004)
clock divider register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH scl high level clock count T(SCL_HIGH) = Tclk_i2c * (CLKDIVH + 1) * 8
15:0	RW	0x0001	CLKDIVL scl low level clock count T(SCL_LOW) = Tclk_i2c * (CLKDIVL + 1) * 8

RKI2C_MRXADDR
Address: Operational Base + offset (0x0008)
the slave address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RW	0x0	addhvld address high byte valid 1'b0:invalid 1'b1:valid
25	RW	0x0	addmvld address middle byte valid 1'b0:invalid 1'b1:valid
24	RW	0x0	addlvld address low byte valid 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	saddr master address register the lowest bit indicate write or read 24 bits address register

RKI2C_MRXRADDR

Address: Operational Base + offset (0x000c)

the slave register address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid 1'b0:invalid 1'b1:valid
25	RW	0x0	sraddmvld address middle byte valid 1'b0:invalid 1'b1:valid
24	RW	0x0	sraddlvld address low byte valid 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	sraddr slave register address accessed 24 bits register address

RKI2C_MTXCNT

Address: Operational Base + offset (0x0010)

master transmit count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	mtxcnt master transmit count 6 bits counter

RKI2C_MRXCNT
Address: Operational Base + offset (0x0014)
master rx count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master rx count 6 bits counter

RKI2C_IEN
Address: Operational Base + offset (0x0018)
interrupt enable register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclen slave hold scl interrupt enable 1'b0:disable 1'b1:enable
6	RW	0x0	nakrcvien NAK handshake received interrupt enable 1'b0:disable 1'b1:enable
5	RW	0x0	stopien stop operation finished interrupt enable 1'b0:disable 1'b1:enable
4	RW	0x0	startien start operation finished interrupt enable 1'b0:disable 1'b1:enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable 1'b0:disable 1'b1:enable
2	RW	0x0	mbtfien MTXCNT data transfer finished interrupt enable 1'b0:disable 1'b1:enable

Bit	Attr	Reset Value	Description
1	RW	0x0	brfien byte rx finished interrupt enable 1'b0:disable 1'b1:enable
0	RW	0x0	btfien byte tx finished interrupt enable 1'b0:disable 1'b1:enable

RKI2C_IPD
Address: Operational Base + offset (0x001c)
interrupt pending register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclipd slave hold scl interrupt pending bit 1'b0:no interrupt available 1'b1:slave hold scl interrupt appear, write 1 to clear
6	W1C	0x0	nakrcvipd NAK handshake received interrupt pending bit 1'b0:no interrupt available 1'b1:NAK handshake received interrupt appear, write 1 to clear
5	W1C	0x0	stopipd stop operation finished interrupt pending bit 1'b0:no interrupt available 1'b1:stop operation finished interrupt appear, write 1 to clear
4	W1C	0x0	startipd start operation finished interrupt pending bit 1'b0:no interrupt available 1'b1:start operation finished interrupt appear, write 1 to clear
3	W1C	0x0	mbrfipd MRXCNT data received finished interrupt pending bit 1'b0:no interrupt available 1'b1:MRXCNT data received finished interrupt appear, write 1 to clear
2	W1C	0x0	mbtfipd MTXCNT data transfer finished interrupt pending bit 1'b0:no interrupt available 1'b1:MTXCNT data transfer finished interrupt appear, write 1 to clear
1	W1C	0x0	brfipd byte rx finished interrupt pending bit 1'b0:no interrupt available 1'b1:byte rx finished interrupt appear, write 1 to clear

Bit	Attr	Reset Value	Description
0	W1C	0x0	btfipd byte tx finished interrupt pending bit 1'b0:no interrupt available 1'b1:byte tx finished interrupt appear, write 1 to clear

RKI2C_FCNT
Address: Operational Base + offset (0x0020)
finished count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	fcnt finished count the count of data which has been transmitted or received for debug purpose

RKI2C_SCL_OE_DB
Address: Operational Base + offset (0x0024)
slave hold debounce configure register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x20	scl_oe_db slave hold scl debounce cycles for debounce (unit: Tclk_i2c)

RKI2C_TXDATA0
Address: Operational Base + offset (0x0100)
I2C tx data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0 data0 to be transmitted 32 bits data

RKI2C_TXDATA1
Address: Operational Base + offset (0x0104)
I2C tx data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1 data1 to be transmitted 32 bits data

RKI2C_TXDATA2
Address: Operational Base + offset (0x0108)
I2C tx data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2 data2 to be transmitted 32 bits data

RKI2C_TXDATA3
Address: Operational Base + offset (0x010c)
I2C tx data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3 data3 to be transmitted 32 bits data

RKI2C_TXDATA4
Address: Operational Base + offset (0x0110)
I2C tx data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4 data4 to be transmitted 32 bits data

RKI2C_TXDATA5
Address: Operational Base + offset (0x0114)
I2C tx data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5 data5 to be transmitted 32 bits data

RKI2C_TXDATA6
Address: Operational Base + offset (0x0118)
I2C tx data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6 data6 to be transmitted 32 bits data

RKI2C_TXDATA7
Address: Operational Base + offset (0x011c)
I2C tx data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7 data7 to be transmitted 32 bits data

RKI2C_RXDATA0
Address: Operational Base + offset (0x0200)
I2C rx data register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0 data0 received 32 bits data

RKI2C_RXDATA1
Address: Operational Base + offset (0x0204)
I2C rx data register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1 data1 received 32 bits data

RKI2C_RXDATA2
Address: Operational Base + offset (0x0208)
I2C rx data register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2 data2 received 32 bits data

RKI2C_RXDATA3
Address: Operational Base + offset (0x020c)
I2C rx data register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3 data3 received 32 bits data

RKI2C_RXDATA4
Address: Operational Base + offset (0x0210)
I2C rx data register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4 data4 received 32 bits data

RKI2C_RXDATA5
Address: Operational Base + offset (0x0214)
I2C rx data register 5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5 data5 received 32 bits data

RKI2C_RXDATA6
Address: Operational Base + offset (0x0218)
I2C rx data register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6 data6 received 32 bits data

RKI2C_RXDATA7
Address: Operational Base + offset (0x021c)
I2C rx data register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7 data7 received 32 bits data

RKI2C_ST
Address: Operational Base + offset (0x0220)
status debug register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	scl_st scl status 1'b0: scl status low 1'b0: scl status high
0	RO	0x0	sda_st sda status 1'b0: sda status low 1'b0: sda status high

11.5 Interface Description

Table 11-1 I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	IO_SPI3PMUrxd_I2C0PMUsda_PMU1830gpio1b7	PMUGRF_GPIO1B_IOMUX[15:14]=2'b10
i2c0_scl	I/O	IO_SPI3PMUtxd_I2C0PMUscl_PMU1830gpio1c0	PMUGRF_GPIO1C_IOMUX[1:0]=2'b10
I2C1 Interface			
i2c1_sda	I/O	IO_I2C1AUDIOCAMsda_TRACEclk_AUDIOgpio4a1	GRF_GPIO4A_IOMUX[3:2]=2'b01

Module pin	Direction	Pad name	IOMUX
i2c1_scl	I/O	IO_I2C1AUDIOCAMscl_TRACEdata8_AUDIOgpio4a2	GRF_GPIO4A_IOMUX[5:4]=2'b01
I2C2 Interface			
i2c2_sda	I/O	IO_VOPdata0_I2C2TPsda_CIFdata0_BT656gpio2a0	GRF_GPIO2A_IOMUX[1:0]=2'b10
i2c2_scl	I/O	IO_VOPdata1_I2C2TPscl_CIFdata1_BT656gpio2a1	GRF_GPIO2A_IOMUX[3:2]=2'b10
I2C3 Interface			
i2c3_sda	I/O	IO_I2C3HDMIsda_UART2DBGBsin_HDMII2Csda_GPIO1830gpio4c0	GRF_GPIO4C_IOMUX[1:0]=2'b01
i2c3_scl	I/O	IO_I2C3HDMIscl_UART2DBGBsout_HDMII2Cscl_GPIO1830gpio4c1	GRF_GPIO4C_IOMUX[3:2]=2'b01
I2C4 Interface			
i2c4_sda	I/O	IO_I2C4SENSORsda_PMU1830gpio1b3	PMUGRF_GPIO1B_IOMUX[7:6]=2'b01
i2c4_scl	I/O	IO_I2C4SENSORscl_PMU1830gpio1b4	PMUGRF_GPIO1B_IOMUX[9:8]=2'b01
I2C5 Interface			
i2c5_sda	I/O	IO_MACrxer_I2C5TRACKPADsda_GMACgpio3b2	GRF_GPIO3B_IOMUX[5:4]=2'b10
i2c5_scl	I/O	IO_MACclk_I2C5TRACKPADscl_GMACgpio3b3	GRF_GPIO3B_IOMUX[7:6]=2'b10
I2C6 Interface			
i2c6_sda	I/O	IO_SPI2TPMrxd_I2C6TPMsda_CIFhref_BT656gpio2b1	GRF_GPIO2B_IOMUX[3:2]=2'b10
i2c6_scl	I/O	IO_SPI2TPMtxd_I2C6TPMscl_CIFclkin_BT656gpio2b2	GRF_GPIO2B_IOMUX[5:4]=2'b10
I2C7 Interface			
i2c7_sda	I/O	IO_VOPdata7_I2C7NFCsda_CIFdata7_BT656gpio2a7	GRF_GPIO2A_IOMUX[15:14]=2'b10
i2c7_scl	I/O	IO_VOPdclk_I2C7NFCscl_CIFvsync_BT656gpio2b0	GRF_GPIO2B_IOMUX[1:0]=2'b10
I2C8 Interface			
i2c8_sda	I/O	IO_I2C8DCDCsda_PMU1830gpio1c4	PMUGRF_GPIO1C_IOMUX[9:8]=2'b01
i2c8_scl	I/O	IO_I2C8DCDCscl_PMU1830gpio1c5	PMUGRF_GPIO1C_IOMUX[11:10]=2'b01

11.6 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections, transmit only mode, receive only mode, and mix mode. Users are strongly advised to follow

- Transmit only mode (I2C_CON[1:0]=2'b00)

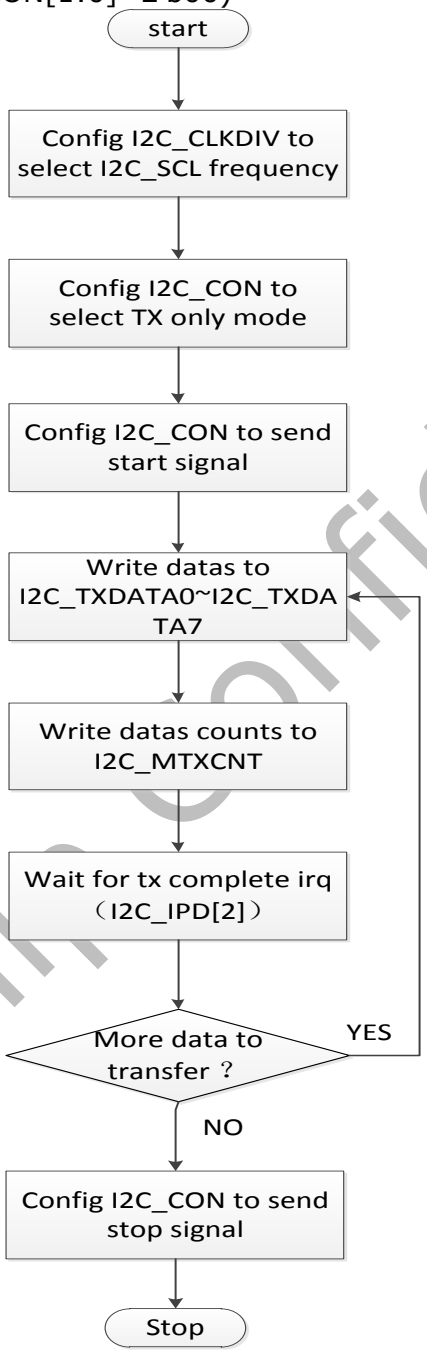


Fig. 11-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

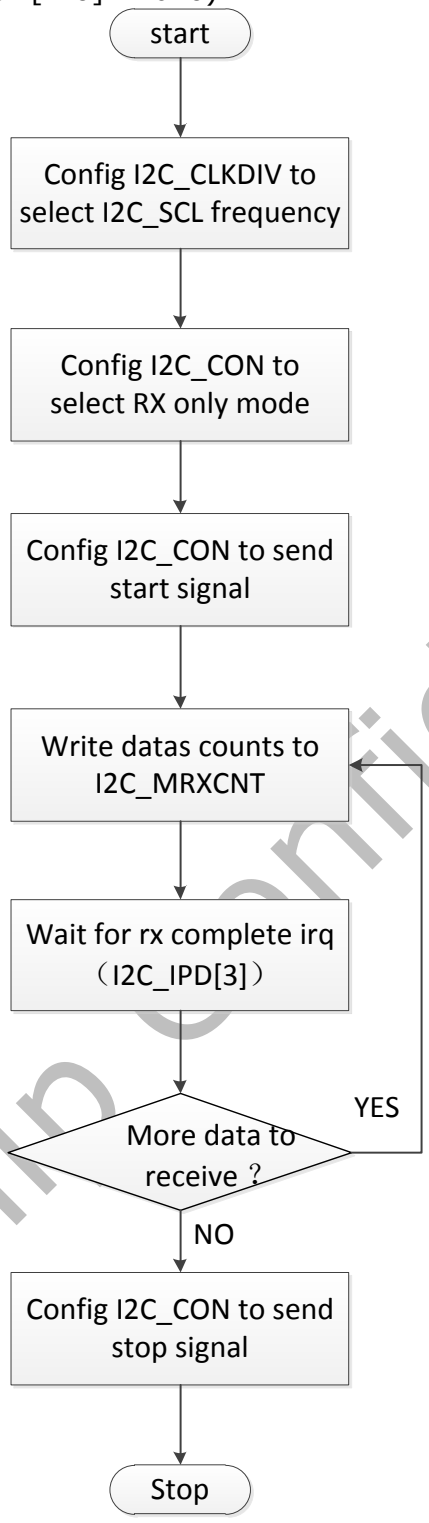


Fig. 11-7 I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

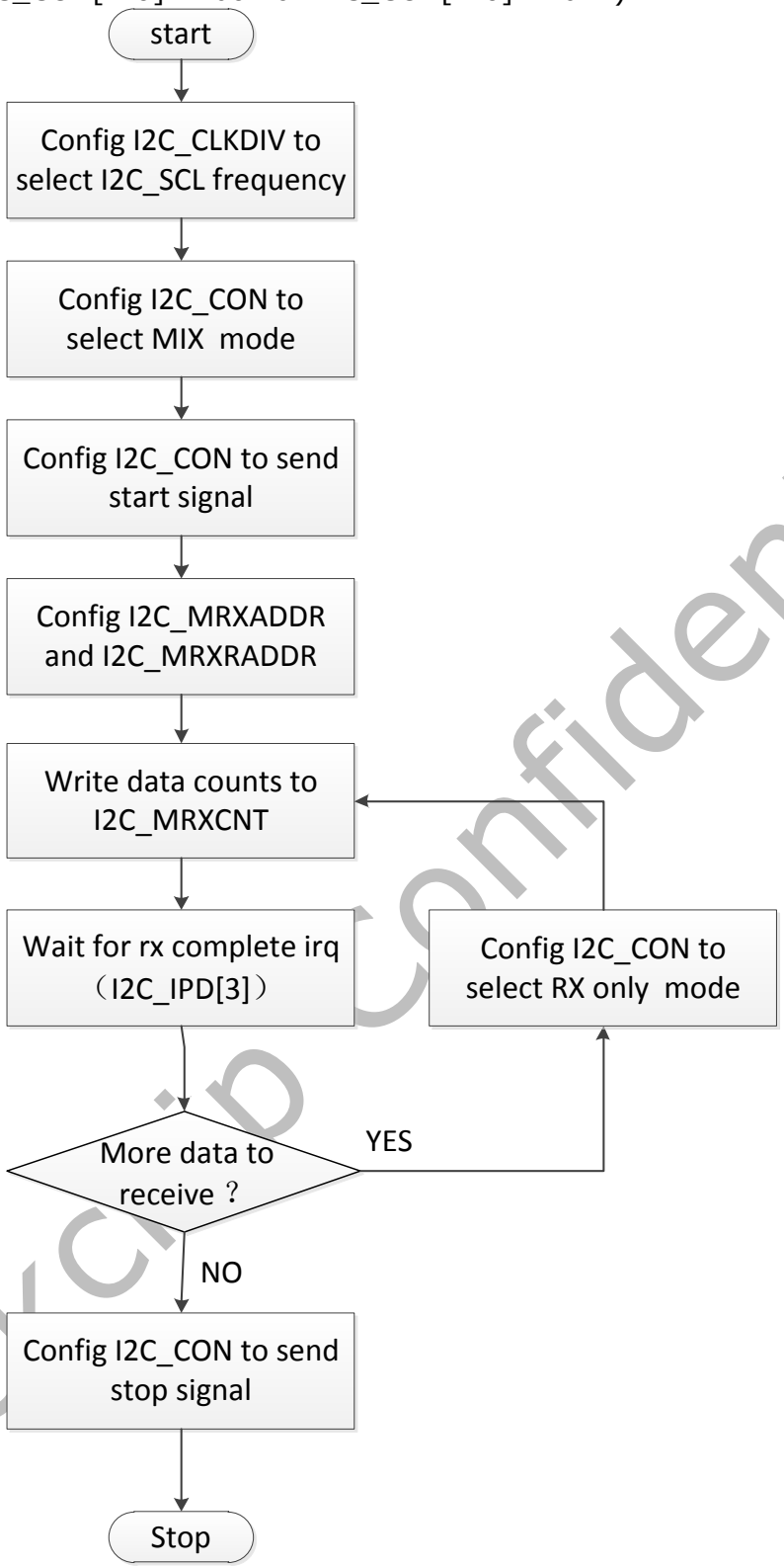


Fig. 11-8 I2C Flow chat for mix mode

Chapter 12 I2S/PCM Controller (8 channel)

12

12.1 Overview

The I2S/PCM controller is designed for interfacing between the AHB bus and the I2S bus. The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.
Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.
Not only I2S but also PCM mode surround audio output and stereo input are supported in I2S/PCM controller.
There are three I2S/PCM controllers embedded in the design, I2S0, I2S1 and I2S2.
Different features between I2S/PCM controllers are as follows.

- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and four for receiving audio data for I2S0
- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and one for receiving audio data for I2S1
- Support four internal 32-bit wide and 32-location deep FIFOs, four for transmitting audio data for I2S2
- Support 10 channels audio data transmitting and receiving in total in I2S mode for I2S0, 2 channels audio data transmitting and 2 channels audio data receiving for I2S1, 8 channels audio data transmitting for I2S2
- Support up to 192kHz sample rate for I2S0 and I2S1, 768kHz sample rate for I2S2

Common features for I2S0, I2S1 and I2S2 are as follows.

- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2 channels audio receiving in PCM mode
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and one for transmitting audio data. Single LRCK can be used for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity

12.2 Block Diagram

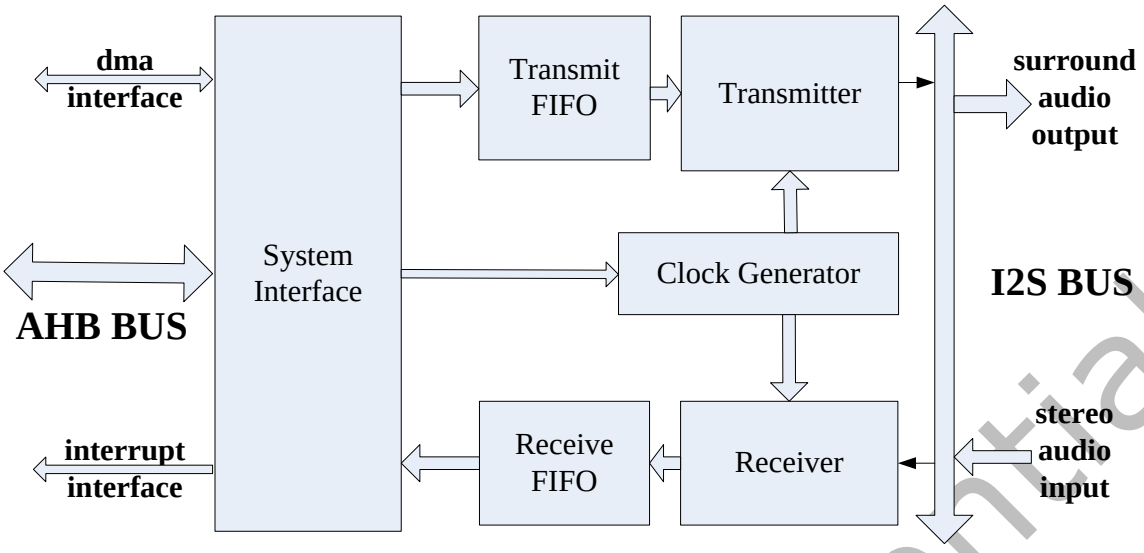


Fig. 12-1 I2S/PCM controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitter and receiver inside but also interrupt and DMA handshake interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitter

The Transmitter implements transmission operation. The transmitter can act as either master or slave, with I2S or PCM mode surround serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

12.3 Function description

In the I2S/PCM controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

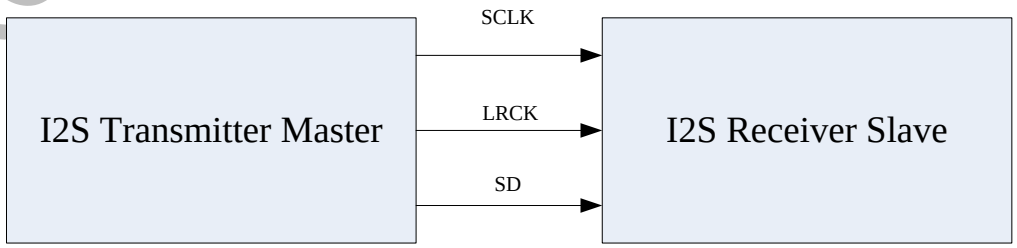


Fig. 12-2 I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready

before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

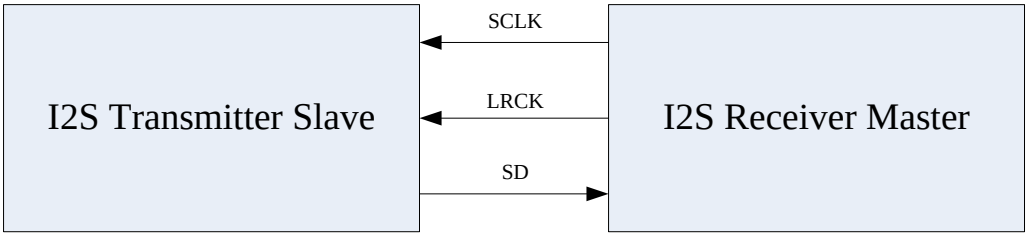


Fig. 12-3 I2S transmitter-slave& receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then the receiver start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

12.3.1 i2s normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

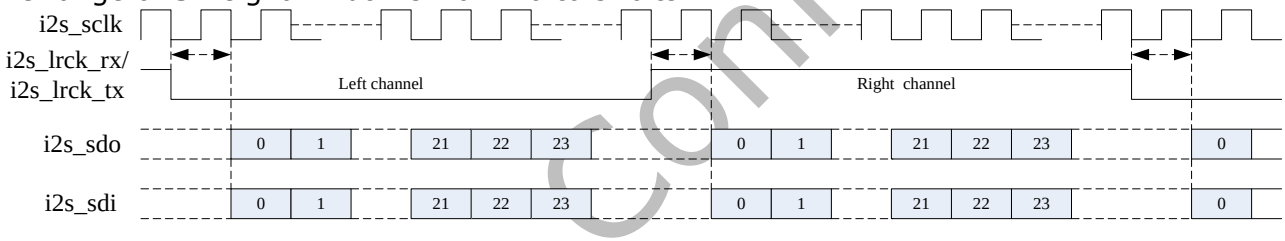


Fig. 12-4 I2S normal mode timing format

12.3.2 i2s left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

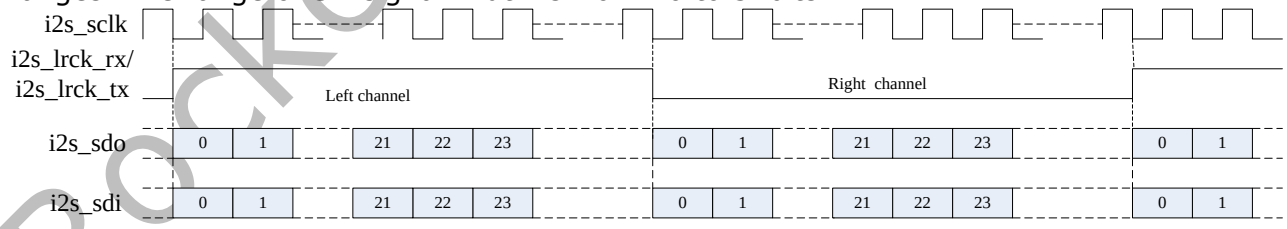


Fig. 12-5 I2S left justified mode timing format

12.3.3 i2s right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

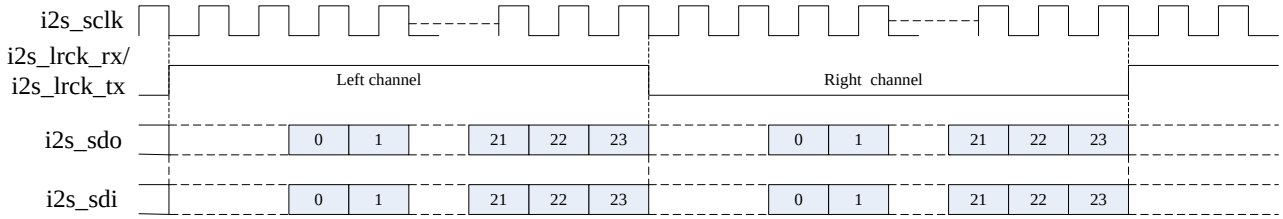


Fig. 12-6 I2S right justified mode timing format

12.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

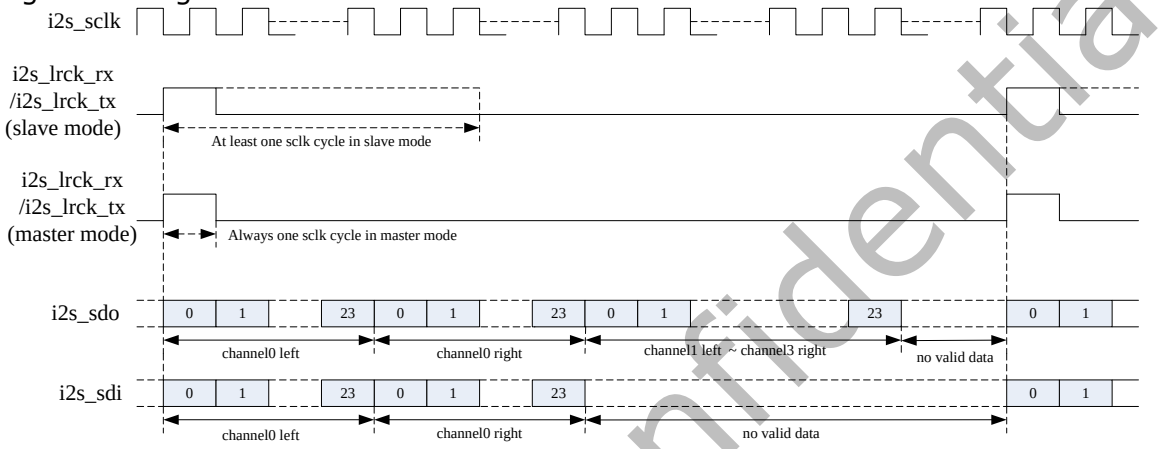


Fig. 12-7 PCM early mode timing format

12.3.5 PCM late1 mode

This is the waveform of PCM late1 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

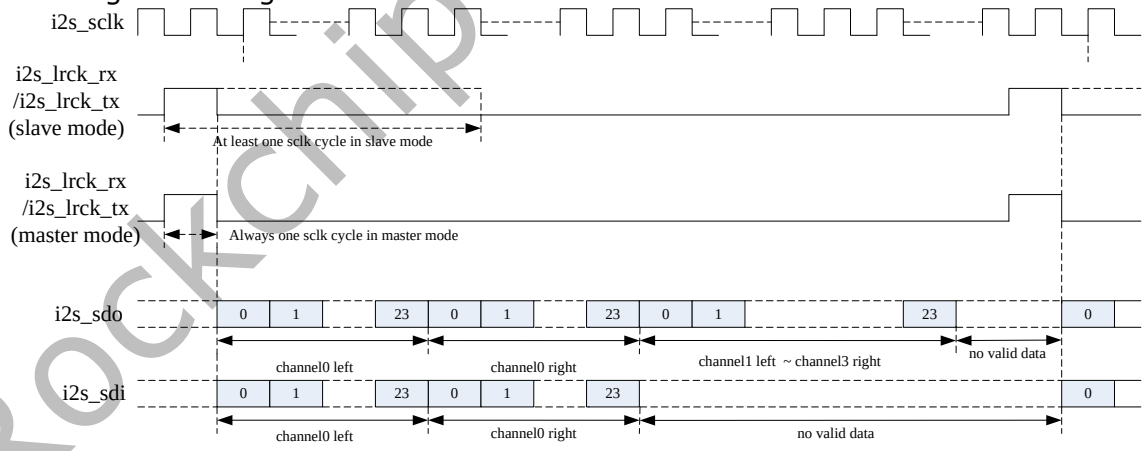


Fig. 12-8 PCM late1 mode timing format

12.3.6 PCM late2 mode

This is the waveform of PCM late2 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

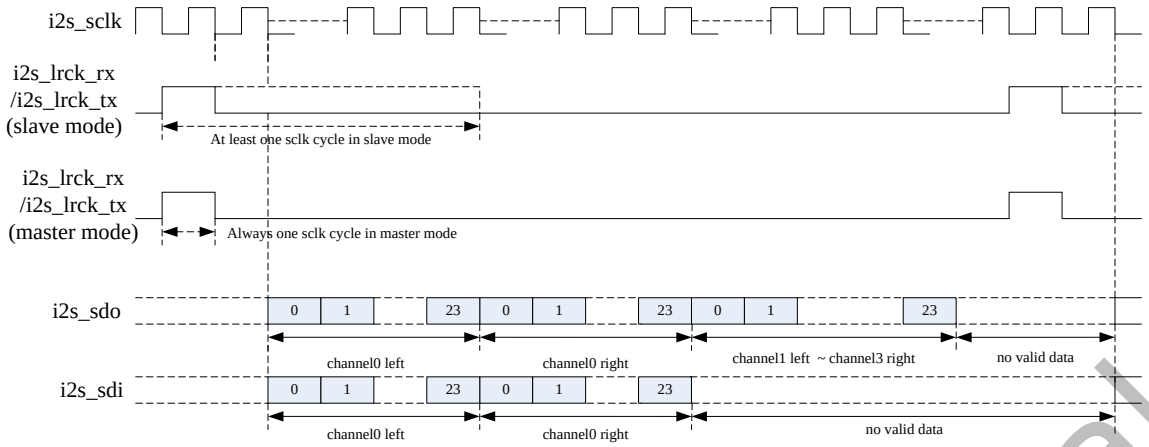


Fig. 12-9 PCM late2 mode timing format

12.3.7 PCM late3 mode

This is the waveform of PCM late3 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

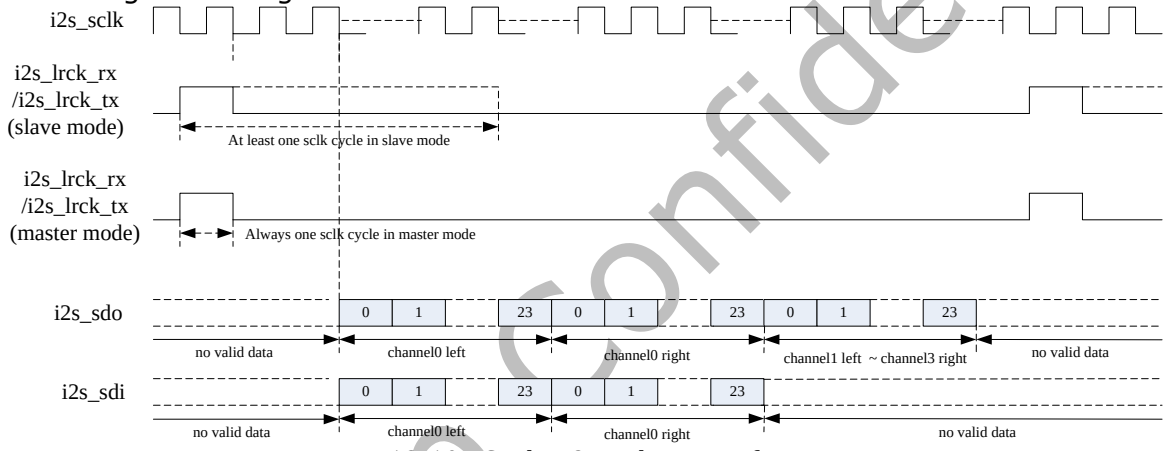


Fig. 12-10 PCM late3 mode timing format

12.4 Register Description

This section describes the control/status registers of the design.

12.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2S_RXCR	0x0004	W	0x0000000f	receive operation control register
I2S_CKR	0x0008	W	0x00071f1f	clock generation register
I2S_TXFIFOLR	0x000c	W	0x00000000	TX FIFO level register
I2S_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_INTCR	0x0014	W	0x00000000	interrupt control register
I2S_INTSR	0x0018	W	0x00000000	interrupt status register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register
I2S_RXFIFOLR	0x002c	W	0x00000000	RX FIFO level register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

12.4.2 Detail Register Description

I2S_TXCR
Address: Operational Base + offset (0x0000)
transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right justified counter (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	TCSR TX Channel select register 2'b00:two channel 2'b01:four channel 2'b10:six channel 2'b11:eight channel
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. This bit is invalid if VDW select 16bit data and HWT select 0, Because every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB

Bit	Attr	Reset Value	Description
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_RXCR

Address: Operational Base + offset (0x0004)

receive operation control register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:15	RW	0x0	RCSR RX Channel select register 2'b00:two channel 2'b01:four channel 2'b10:six channel 2'b11:eight channel
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_CKR
Address: Operational Base + offset (0x0008)
clock generation register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM Tx and Rx Common Use 2'b00/2'b11:tx_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01:only tx_lrck is used as synchronous signal for TX and RX. 2'b10:only rx_lrck is used as synchronous signal for TX and RX.
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk

Bit	Attr	Reset Value	Description
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
24	RW	0x0	TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
23:16	RW	0x07	MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclk frequecy / txsclk frequecy-1) 0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; …… 2n,2n+1:Fmclk=(2n+2)*Ftxsclk; …… 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; …… 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description
15:8	RW	0x1f	RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs
7:0	RW	0x1f	TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs

I2S_TXFIFOLR

Address: Operational Base + offset (0x000c)

TX FIFO level register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.
17:12	RO	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RO	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.

Bit	Attr	Reset Value	Description
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S_DMACR
Address: Operational Base + offset (0x0010)
DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO (RXFIFO0 if RCSR=00;RXFIFO1 if RCSR=01,RXFIFO2 if RCSR=10,RXFIFO3 if RCSR=11)is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if TCSR=00;TXFIFO1 if TCSR=01,TXFIFO2 if TCSR=10,TXFIFO3 if TCSR=11)is equal to or below this field value.

I2S_INTCR
Address: Operational Base + offset (0x0014)
interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:20	RW	0x00	RFT Receive FIFO Threshold When the number of receive FIFO entries (RXFIFO0 if RCSR=00; RXFIFO1 if RCSR=01, RXFIFO2 if RCSR=10, RXFIFO3 if RCSR=11) is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if TCSR=00; TXFIFO1 if TCSR=01, TXFIFO2 if TCSR=10, TXFIFO3 if TCSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2S_INTSR

Address: Operational Base + offset (0x0018)

interrupt status register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2S_XFER
Address: Operational Base + offset (0x001c)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2S_CLR
Address: Operational Base + offset (0x0020)
SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX logic clear This is a self cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX logic clear This is a self cleared bit. Write 1 to clear all transmit logic.

I2S_TXDR
Address: Operational Base + offset (0x0024)
Transmit FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_RXDR
Address: Operational Base + offset (0x0028)
Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

I2S_RXFIFOLR
Address: Operational Base + offset (0x002c)
RX FIFO level register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RO	0x00	RFL3 Receive FIFO3 Level Contains the number of valid data entries in the receive FIFO3.
17:12	RO	0x00	RFL2 Receive FIFO2 Level Contains the number of valid data entries in the receive FIFO2.
11:6	RU	0x00	RFL1 Receive FIFO1 Level Contains the number of valid data entries in the receive FIFO1.
5:0	RO	0x00	RFL0 Receive FIFO0 Level Contains the number of valid data entries in the receive FIFO0.

12.5 Interface description

Table 12-1 I2S Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
Interface for i2s0			
i2s_mclk	I/O	IO_I2Sclk_TRACEctl_LPM0wfi_AUDIOgpio4a0	GRF_GPIO4A_IOMUX[1:0]=2'b01
i2s0_sclk	I/O	IO_I2S0sclk_TRACEdata0_A72CORE0wfi_AUDIOgpio3d0	GRF_GPIO3D_IOMUX[1:0]=2'b01
i2s0_lrck_rx	I/O	IO_I2S0lrckrx_TRACEdata1_A72CORE1wfi_AUDIOgpio3d1	GRF_GPIO3D_IOMUX[3:2]=2'b01
i2s0_lrck_tx	I/O	IO_I2S0lrcktx_TRACEdata2_A53CORE0wfi_AUDIOgpio3d2	GRF_GPIO3D_IOMUX[5:4]=2'b01
i2s0_sdo0	O	IO_I2S0sdo0_TRACEdata7_A53L2wfi_AUDIOgpio3d7	GRF_GPIO3D_IOMUX[15:14]=2'b01
i2s0_sdo1	O	IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6	GRF_GPIO3D_IOMUX[13:12]=2'b01
i2s0_sdo2	O	IO_I2S0sdi2sdo2_TRACEdata5_A	GRF_GPIO3D_IOMUX[11:10]=2'b01

Module Pin	Direction	Pad Name	IOMUX Setting
		53CORE3wfi_AUDIOgpio3d5	
i2s0_sdo3	O	IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4	GRF_GPIO3D_IOMUX[9:8]=2'b01
i2s0_sdi0	I	IO_I2S0sdi0_TRACEdata3_A53CORE1wfi_AUDIOgpio3d3	GRF_GPIO3D_IOMUX[7:6]=2'b01
i2s0_sdi1	I	IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4	GRF_GPIO3D_IOMUX[9:8]=2'b01
i2s0_sdi2	I	IO_I2S0sdi2sdo2_TRACEdata5_A53CORE3wfi_AUDIOgpio3d5	GRF_GPIO3D_IOMUX[11:10]=2'b01
i2s0_sdi3	I	IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6	GRF_GPIO3D_IOMUX[13:12]=2'b01
Interface for i2s1			
i2s1_mclk	I/O	IO_I2Sclk_TRACEctl_LPM0wfi_AUDIOgpio4a0	GRF_GPIO4A_IOMUX[1:0]=2'b01
i2s1_sclk	I/O	IO_I2S1sclk_TRACEdata9_AUDIOgpio4a3	GRF_GPIO4A_IOMUX[7:6]=2'b01
i2s1_lrck_rx	I/O	IO_I2S1lrckrx_TRACEdata10_AUDIOgpio4a4	GRF_GPIO4A_IOMUX[9:8]=2'b01
i2s1_lrck_tx	I/O	IO_I2S1lrcktx_TRACEdata11_AUDIOgpio4a5	GRF_GPIO4A_IOMUX[11:10]=2'b01
i2s1_sdi	I	IO_I2S1sdi0_AUDIOgpio4a6	GRF_GPIO4A_IOMUX[13:12]=2'b01
i2s1_sdo	O	IO_I2S1sdo0_AUDIOgpio4a7	GRF_GPIO4A_IOMUX[15:14]=2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

The i2s0_sdix(x=1,2,3) and i2s0_sdox(x=1,2,3) signals shares the same IO, the direction is configured by setting GRF_SOC_CON8[13:11]. GRF_SOC_CON8[11] controls the direction of IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6, GRF_SOC_CON8[12] corresponds to IO_I2S0sdi2sdo2_TRACEdata5_A53CORE3wfi_AUDIOgpio3d5 and GRF_SOC_CON8[13] corresponds to IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4.

The I2S2 module is connected to the audio interface of HDMI and DP, which supports 8 channels audio data transmitting.

Table 12-3 I2S Interface Between I2S2 and HDMI

Module Pin	Direction	Module Pin	Direction
i2s2_sclk_out	O	ii2sclk	I
i2s2_tx_lrck_out	O	ii2slrclk	I
i2s2_sdo[3:0]	O	ii2sdata[3:0]	I

Table 12-4 I2S Interface Between I2S2 and DP

Module Pin	Direction	Module Pin	Direction
i2s2_sclk_out	O	source_i2s_clk	I
i2s2_tx_lrck_out	O	source_i2s_ws	I
i2s2_8ch_sdo[3:0]	O	source_i2s_data[3:0]	I

12.6 Application Notes

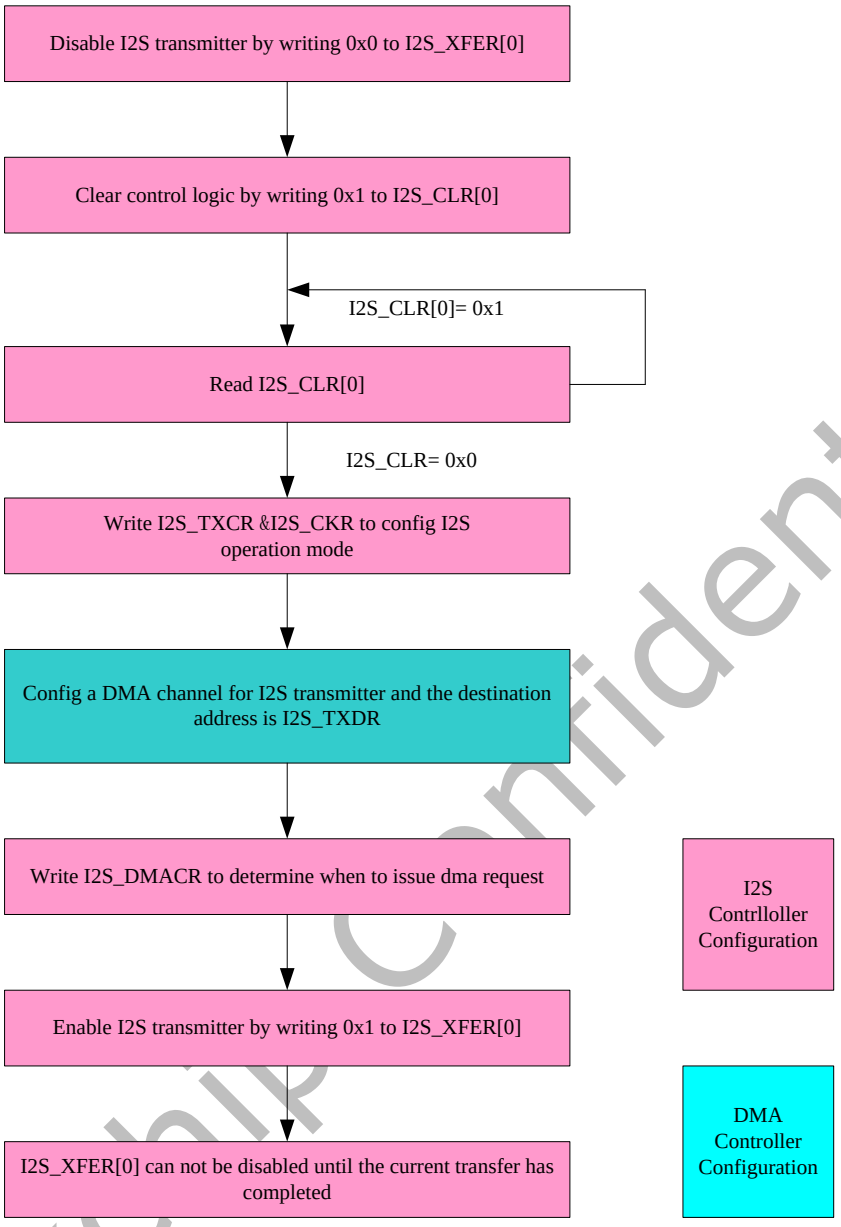


Fig. 12-11 I2S/PCM controller transmit operation flow chart

Chapter 13 Serial Peripheral Interface (SPI)

13.1 Overview

The serial peripheral interface is an APB slave device. A four wire full duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

13.2 Block Diagram

The SPI Controller comprises with:

- AMBA APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt

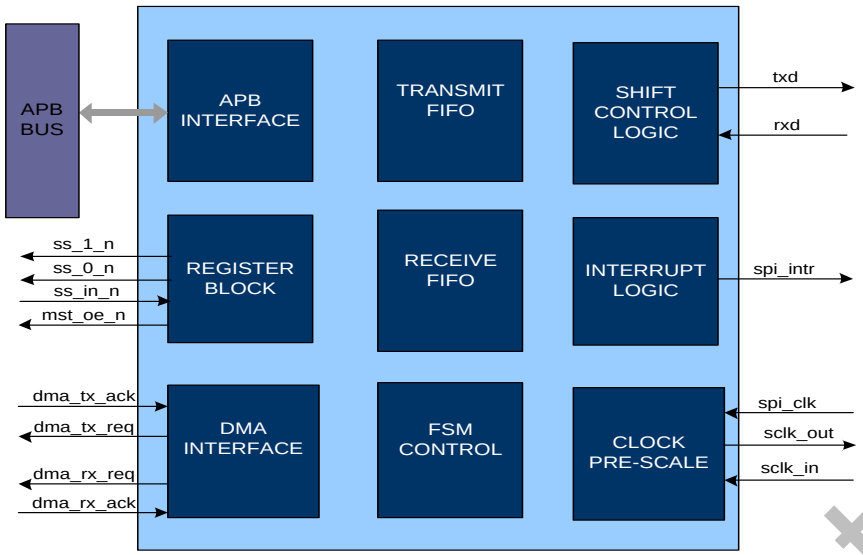


Fig. 13-1 SPI Controller Block diagram

APB INTERFACE

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 32 bits and 8 or 16 bits when reading or writing internal FIFO if data frame size(SPI_CTRL0[1:0]) is set to 8 bits.

DMA INTERFACE

This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC

For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Both fifos are 32x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

13.3 Function Description

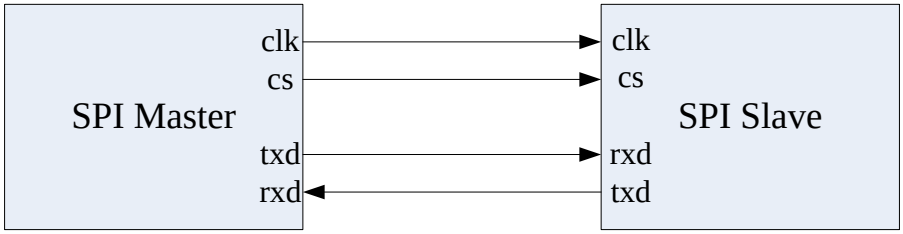


Fig. 13-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.

1). Transmit and Receive

When SPI_CTRLR0 [19:18]== 2'b00, both transmit and receive logic are valid.

2).Transmit Only

When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3).Receive Only

When SPI_CTRLR0 [19:18]== 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the F_{spi_clk}>= 2 × (maximum F_{sclk_out})

When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

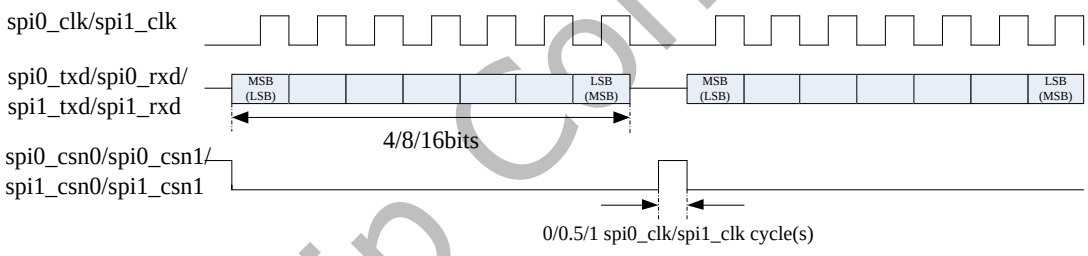


Fig. 13-3 SPI Format (SCPH=0 SCPOL=0)

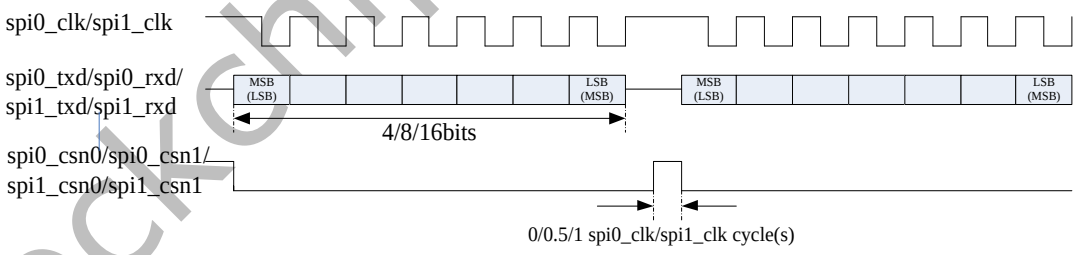


Fig. 13-4 SPI Format (SCPH=0 SCPOL=1)

When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. The following two figures show the timing diagram for the SPI format when the configuration parameter SCPH = 1.

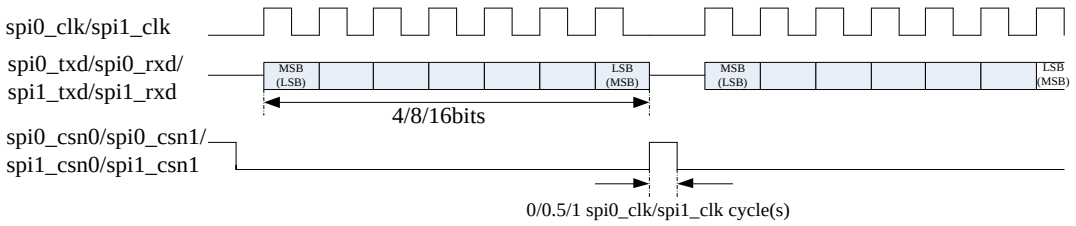


Fig. 13-5 SPI Format (SCPH=1 SCPOL=0)

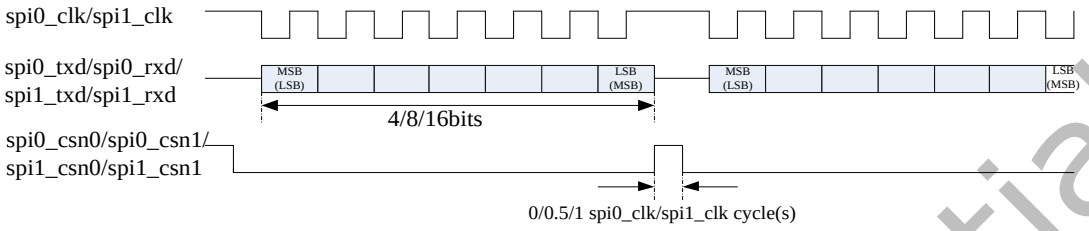


Fig. 13-6 SPI Format (SCPH=1 SCPOL=1)

13.4 Register Description

13.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable
SPI_SER	0x000c	W	0x00000000	Slave Enable Register
SPI_BAUDR	0x0010	W	0x00000000	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x00000000	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x00000000	Receive FIFO Threshold Level
SPI_TXFLR	0x001c	W	0x00000000	Transmit FIFO Level
SPI_RXFLR	0x0020	W	0x00000000	Receive FIFO Level
SPI_SR	0x0024	W	0x0000000c	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TXDR	0x0400	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800	W	0x00000000	Receive FIFO Data

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

13.4.2 Detail Register Description

SPI_CTRLR0
Address: Operational Base + offset (0x0000)
Control Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	MTM Microwire Transfer Mode Valid when frame format is set to National Semiconductors Microwire. 1'b0: non-sequential transfer 1'b1: sequential transfer
20	RW	0x0	OPM Operation Mode 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	XFM Transfer Mode 2'b00 :Transmit & Receive 2'b01 : Transmit Only 2'b10 : Receive Only 2'b11 :reserved
17:16	RW	0x0	FRF Frame Format 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11 : Reserved
15:14	RW	0x0	RSD Rxd Sample Delay When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00:do not delay 2'b01:1 cycle delay 2'b10:2 cycles delay 2'b11:3 cycles delay
13	RW	0x0	BHT Byte and Halfword Transform Valid when data frame size is 8bit. 1'b0:apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM First Bit Mode 1'b0:first bit is MSB 1'b1:first bit is LSB

Bit	Attr	Reset Value	Description
11	RW	0x0	EM Endian Mode Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1'b0:little endian 1'b1:big endian
10	RW	0x0	SSD ss_n to sclk_out delay Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: the period between ss_n active and sclk_out active is one sclk_out cycle.
9:8	RW	0x0	CSM Chip Select Mode Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01:ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11:reserved
7	RW	0x0	SCPOL Serial Clock Polarity Valid when the frame format is set to Motorola SPI. 1'b0: Inactive state of serial clock is low 1'b1: Inactive state of serial clock is high
6	RW	0x0	SCPH Serial Clock Phase Valid when the frame format is set to Motorola SPI. 1'b0: Serial clock toggles in middle of first data bit 1'b1: Serial clock toggles at start of first data bit

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CFS Control Frame Size Selects the length of the control word for the Microwire frame format. 4'b0000~0010:reserved 4'b0011:4-bit serial data transfer 4'b0100:5-bit serial data transfer 4'b0101:6-bit serial data transfer 4'b0110:7-bit serial data transfer 4'b0111:8-bit serial data transfer 4'b1000:9-bit serial data transfer 4'b1001:10-bit serial data transfer 4'b1010:11-bit serial data transfer 4'b1011:12-bit serial data transfer 4'b1100:13-bit serial data transfer 4'b1101:14-bit serial data transfer 4'b1110:15-bit serial data transfer 4'b1111:16-bit serial data transfer
1:0	RW	0x2	DFS Data Frame Size Selects the data frame length. 2'b00:4bit data 2'b01:8bit data 2'b10:16bit data 2'b11:reserved

SPI_CTRLR1
Address: Operational Base + offset (0x0004)
Control Register 1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	NDM Number of Data Frames When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

SPI_ENR
Address: Operational Base + offset (0x0008)
SPI Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	ENR SPI Enable 1'b1: Enable all SPI operations. 1'b0: Disable all SPI operations Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER
Address: Operational Base + offset (0x000c)
Slave Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	SER1 Slave 1 Select Enable 1'b1: Enable chip select 1 1'b0: Disable chip select 1 This register is valid only when SPI is configured as a master device.
0	RW	0x0	SER0 Slave Select Enable 1'b1: Enable chip select 0 1'b0: Disable chip select 0 This register is valid only when SPI is configured as a master device.

SPI_BAUDR
Address: Operational Base + offset (0x0010)
Baud Rate Select

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	BAUDR Baud Rate Select SPI Clock Divider. This register is valid only when the SPI is configured as a master device. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fspi_clk/ SCKDV Where SCKDV is any even value between 2 and 65534. For example: for Fspi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2= 1.8432MHz

SPI_TXFTLR
Address: Operational Base + offset (0x0014)
Transmit FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TXFTLR Transmit FIFO Threshold Level When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SPI_RXFTLR
Address: Operational Base + offset (0x0018)
Receive FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RXFTLR Receive FIFO Threshold Level When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

SPI_TXFLR
Address: Operational Base + offset (0x001c)
Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	TXFLR Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR
Address: Operational Base + offset (0x0020)
Receive FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	RXFLR Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

SPI_SR
Address: Operational Base + offset (0x0024)
SPI Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFF Receive FIFO Full 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RO	0x1	RFE Receive FIFO Empty 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RO	0x1	TFE Transmit FIFO Empty 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RO	0x0	TFF Transmit FIFO Full 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full
0	RO	0x0	BSF SPI Busy Flag When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled 1'b1: SPI is actively transferring data

SPI_IPR
Address: Operational Base + offset (0x0028)
Interrupt Polarity

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	IPR Interrupt Polarity Interrupt Polarity Register 1'b0:Active Interrupt Polarity Level is HIGH 1'b1: Active Interrupt Polarity Level is LOW

SPI_IMR
Address: Operational Base + offset (0x002c)
Interrupt Mask

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFIM Receive FIFO Full Interrupt Mask 1'b0: spi_rxf_intr interrupt is masked 1'b1: spi_rxf_intr interrupt is not masked
3	RW	0x0	RFOIM Receive FIFO Overflow Interrupt Mask 1'b0: spi_rxo_intr interrupt is masked 1'b1: spi_rxo_intr interrupt is not masked
2	RW	0x0	RFUIM Receive FIFO Underflow Interrupt Mask 1'b0: spi_rxu_intr interrupt is masked 1'b1: spi_rxu_intr interrupt is not masked
1	RW	0x0	TFOIM Transmit FIFO Overflow Interrupt Mask 1'b0: spi_txo_intr interrupt is masked 1'b1: spi_txo_intr interrupt is not masked
0	RW	0x0	TFEIM Transmit FIFO Empty Interrupt Mask 1'b0: spi_txe_intr interrupt is masked 1'b1: spi_txe_intr interrupt is not masked

SPI_ISR
Address: Operational Base + offset (0x0030)
Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFIS Receive FIFO Full Interrupt Status 1'b0: spi_rxf_intr interrupt is not active after masking 1'b1: spi_rxf_intr interrupt is full after masking
3	RO	0x0	RFOIS Receive FIFO Overflow Interrupt Status 1'b0: spi_rxo_intr interrupt is not active after masking 1'b1: spi_rxo_intr interrupt is active after masking

Bit	Attr	Reset Value	Description
2	RO	0x0	RFUIS Receive FIFO Underflow Interrupt Status 1'b0: spi_rxu_intr interrupt is not active after masking 1'b1: spi_rxu_intr interrupt is active after masking
1	RO	0x0	TFOIS Transmit FIFO Overflow Interrupt Status 1'b0: spi_txo_intr interrupt is not active after masking 1'b1: spi_txo_intr interrupt is active after masking
0	RO	0x0	TFEIS Transmit FIFO Empty Interrupt Status 1'b0: spi_txe_intr interrupt is not active after masking 1'b1: spi_txe_intr interrupt is active after masking

SPI_RISR
Address: Operational Base + offset (0x0034)
Raw Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFRIS Receive FIFO Full Raw Interrupt Status 1'b0: spi_rxf_intr interrupt is not active prior to masking 1'b1: spi_rxf_intr interrupt is full prior to masking
3	RO	0x0	RFORIS Receive FIFO Overflow Raw Interrupt Status 1'b0 = spi_rxo_intr interrupt is not active prior to masking 1'b1 = spi_rxo_intr interrupt is active prior to masking
2	RO	0x0	RFURIS Receive FIFO Underflow Raw Interrupt Status 1'b0: spi_rxu_intr interrupt is not active prior to masking 1'b1: spi_rxu_intr interrupt is active prior to masking
1	RO	0x0	TFORIS Transmit FIFO Overflow Raw Interrupt Status 1'b0: spi_txo_intr interrupt is not active prior to masking 1'b1: spi_txo_intr interrupt is active prior to masking
0	RO	0x1	TFERIS Transmit FIFO Empty Raw Interrupt Status 1'b0: spi_txe_intr interrupt is not active prior to masking 1'b1: spi_txe_intr interrupt is active prior to masking

SPI_ICR
Address: Operational Base + offset (0x0038)
Interrupt Clear

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	WO	0x0	CTFOI Clear Transmit FIFO Overflow Interrupt Write 1 to Clear Transmit FIFO Overflow Interrupt
2	WO	0x0	CRFOI Clear Receive FIFO Overflow Interrupt Write 1 to Clear Receive FIFO Overflow Interrupt
1	WO	0x0	CRFUI Clear Receive FIFO Underflow Interrupt Write 1 to Clear Receive FIFO Underflow Interrupt
0	WO	0x0	CCI Clear Combined Interrupt Write 1 to Clear Combined Interrupt

SPI_DMACR
Address: Operational Base + offset (0x003c)
DMA Control

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
0	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled

SPI_DMATDLR
Address: Operational Base + offset (0x0040)
DMA Transmit Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI_DMARDLR
Address: Operational Base + offset (0x0044)
DMA Receive Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI_TXDR
Address: Operational Base + offset (0x0048)
Transmit FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	WO	0x0000	TXDR Transimt FIFO Data Register. When it is written to, data are moved into the transmit FIFO.

SPI_RXDR
Address: Operational Base + offset (0x004c)
Receive FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RXDR Receive FIFO Data Register. When the register is read, data in the receive FIFO is accessed.

13.5 Interface Description

Table 13-1 SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_sclk	I/O	IO_MACrxd0_SPI0NORCODECclk_GMACgpio3a6	GRF_GPI03A_IOMUX[13:12]=2'b10
spi0_rxd	I	IO_MACtxd0_SPI0NORCODECrxd_GMACgpio3a4	GRF_GPI03A_IOMUX[9:8]=2'b10
spi0_txd	O	IO_MACtxd1_SPI0NORCODECtxd_GMACgpio3a5	GRF_GPI03A_IOMUX[11:10]=2'b10
spi0_csn0	I/O	IO_MACrxd1_SPI0NORCODECcsn0_GMACgpio3a7	GRF_GPI03A_IOMUX[15:14]=2'b10
spi0_csn1	O	IO_MACmdc_SPI0NORCODECcsn1_GMACgpio3b0	GRF_GPI03B_IOMUX[1:0]=2'b10
spi1_sclk	I/O	IO_PMUM0JTAGtck_SPI1ECclk_PMU1830gpio1b1	PMUGRF_GPIO1B_IOMUX[3:2]=2'b10
spi1_rxd	I	IO_UART4M0sin_SPI1ECrxd_PMU1830gpio1a7	PMUGRF_GPIO1A_IOMUX[15:14]=2'b10
spi1_txd	O	IO_UART4M0sout_SPI1ECtxd_PMU1830gpio1b0	PMUGRF_GPIO1B_IOMUX[1:0]=2'b10
spi1_csn0	I/O	IO_PMUM0JTAGtms_SPI1ECcsn	PMUGRF_GPIO1B_IOMUX[5:4]

Module Pin	Direction	Pad Name	IOMUX Setting
		0_PMU1830gpio1b2	=2'b10
spi2_sclk	I/O	IO_SPI2TPMclk_VOPden_CIFclkouta_BT656gpio2b3	GRF_GPI02B_IOMUX[7:6]=2'b01
spi2_rxd	I	IO_SPI2TPMrxd_I2C6TPMsda_CIFhref_BT656gpio2b1	GRF_GPI02B_IOMUX[3:2]=2'b01
spi2_txd	O	IO_SPI2TPMtxd_I2C6TPMscl_CIFclkin_BT656gpio2b2	GRF_GPI02B_IOMUX[5:4]=2'b01
spi2_csn0	I/O	IO_SPI2TPMcsn0_BT656gpio2b4	GRF_GPI02B_IOMUX[9:8]=2'b01
spi3_sclk	I/O	IO_SPI3PMUclk_PMU1830gpio1c1	PMUGRF_GPIO1C_IOMUX[3:2]=2'b10
spi3_rxd	I	IO_SPI3PMUrxd_I2C0PMUsda_PMU1830gpio1b7	PMUGRF_GPIO1B_IOMUX[15:14]=2'b10
spi3_txd	O	IO_SPI3PMUtxd_I2C0PMUscl_PMU1830gpio1c0	PMUGRF_GPIO1C_IOMUX[1:0]=2'b10
spi3_csn0	I/O	IO_SPI3PMUcsn0_PMU1830gpio1c2	PMUGRF_GPIO1C_IOMUX[5:4]=2'b10
spi4_sclk	I/O	IO_MACrxd2_SPI4EXPclk_TRACEdata14_GMACgpio3a2	GRF_GPI03A_IOMUX[5:4]=2'b10
spi4_rxd	I	IO_MACtxd2_SPI4EXPrxd_TRACEdata12_GMACgpio3a0	GRF_GPI03A_IOMUX[1:0]=2'b10
spi4_txd	O	IO_MACtxd3_SPI4EXPtxd_TRACEdata13_GMACgpio3a1	GRF_GPI03A_IOMUX[3:2]=2'b10
spi4_csn0	I/O	IO_MACrxd3_SPI4EXPcsn0_TRACEdata15_GMACgpio3a3	GRF_GPI03A_IOMUX[7:6]=2'b10
spi5_sclk	I/O	IO_SDIOdata2_SPI5EXPPLUSclk_WIFIBTgpio2c6	GRF_GPI02C_IOMUX[13:12]=2'b10
spi5_rxd	I	IO_SDIOdata0_SPI5EXPPLUSrxd_WIFIBTgpio2c4	GRF_GPI02C_IOMUX[9:8]=2'b10
spi5_txd	O	IO_SDIOdata1_SPI5EXPPLUStxd_WIFIBTgpio2c5	GRF_GPI02C_IOMUX[11:10]=2'b10
spi5_csn0	I/O	IO_SDIOdata3_SPI5EXPPLUScsn0_WIFIBTgpio2c7	GRF_GPI02C_IOMUX[15:14]=2'b10

Notes: I=input, O=output, I/O=input/output, bidirectional. spi_csn1 can only be used in master mode

13.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the Fspi_clk>= 2 × (maximum Fsclk_out)

When SPI Controller works as slave, the Fspi_clk>= 6 × (maximum Fsclk_in)

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

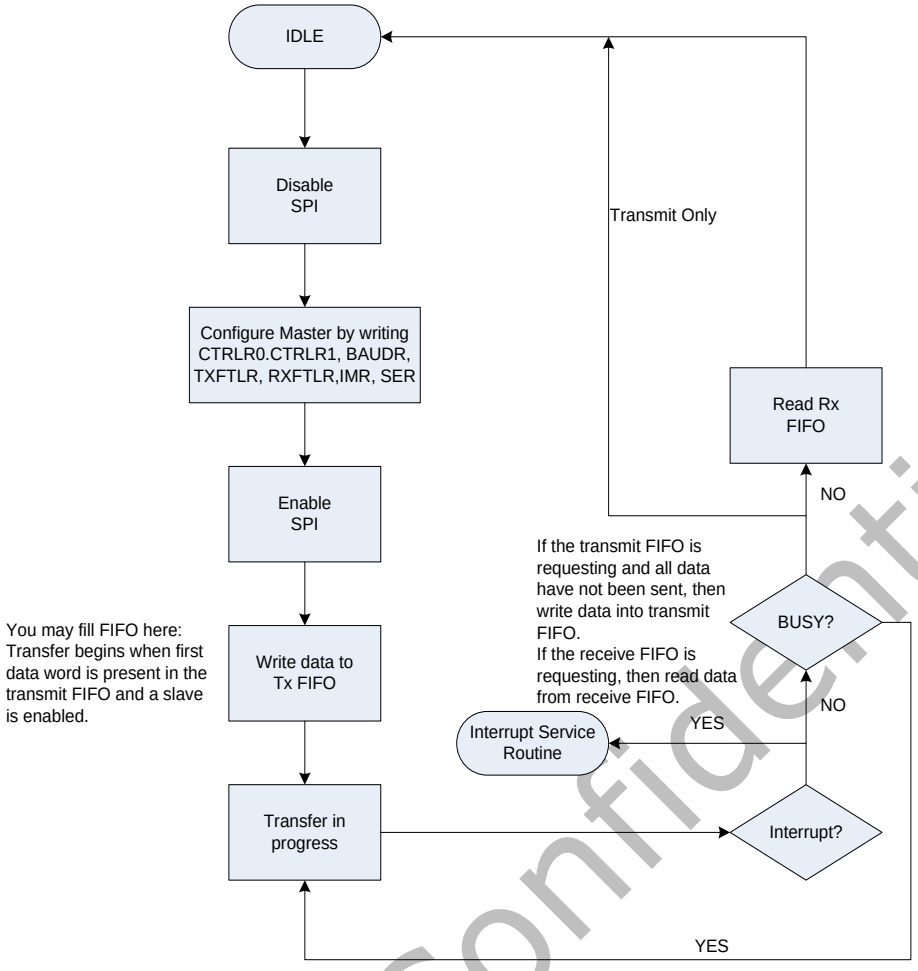


Fig. 13-7 SPI Master transfer flow diagram

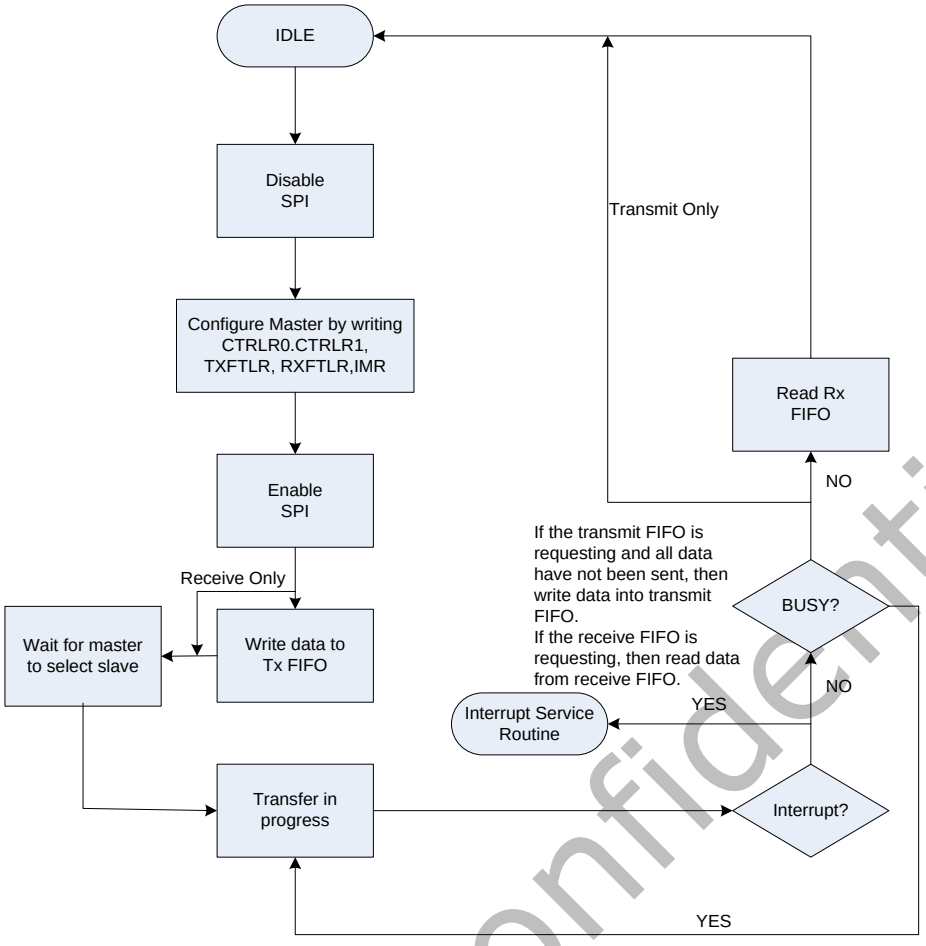


Fig. 13-8 SPI Slave transfer flow diagram

Chapter 14 SPDIF transmitter

14.1 Overview

The SPDIF transmitter is a self-clocking, serial and unidirectional interface for the interconnection of digital audio equipment in consumer and professional applications which uses linear PCM coded audio samples.
When used in professional application, the interface is primarily intended to carry monophonic or stereophonic programmes at a 48 kHz sampling frequency with a resolution of up to 24bits per sample. It may alternatively be used to carry signals sampled at 32 kHz or 44.1 kHz.
When used in consumer application, the interface is primarily intended to carry stereophonic programmes with a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.
When used for other purposes, the interface is primarily intended to carry audio data coded other than linear PCM coded audio samples. Provision is also made to allow the interface to carry data related to computer software or signals coded using non-linear PCM. The maximum sample frequency can be up to 768 kHz for the non-linear PCM mode.
In all cases, the clock references and auxiliary information are transmitted along with the programme.

- Supports one internal 32-bit wide and 32-location deep sample data buffer
- Supports two 16-bit audio data store together in one 32-bit wide location
- Supports AHB bus interface
- Supports biphase format stereo audio data output
- Supports DMA handshake interface and configurable DMA water level
- Supports sample data buffer empty, block terminate and user data interrupt
- Supports combine interrupt output
- Supports 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
- Support 16, 20, 24 bits audio data transfer in linear PCM mode
- Support non-linear PCM transfer

14.2 Block Diagram

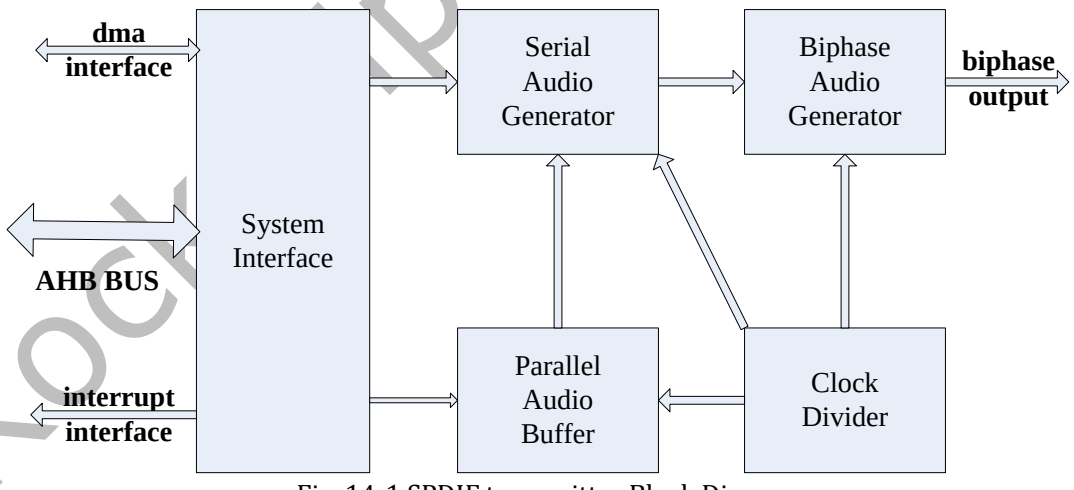


Fig. 14-1 SPDIF transmitter Block Diagram

System Interface
The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshake interface.
Clock Divider
The Clock Divider implements clock generation function. The input source clock to the module is MCLK. By the divider of the module, the clock divider generates work clock for digital audio data transformation.
Parallel Audio Buffer

The Parallel Audio Buffer is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Serial Audio Converter

The Serial Audio Converter reads parallel audio data from the Parallel Audio Buffer and converts it to serial audio data.

Biphase Audio Generator

The Biphase Audio Generator reads serial audio data from the Serial Audio Converter and generates biphase audio data based on IEC-60958 standard.

14.3 Function description

14.3.1 Frame Format

A frame is uniquely composed of two sub-frames. For linear coded audio applications, the rate of transmission of frames corresponds exactly to the source sampling frequency.
In the 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames. The first sub-frame(left channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once every 192 frame to identify the start of the block structure used to organize the channel status information.
The second sub-frame (right in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

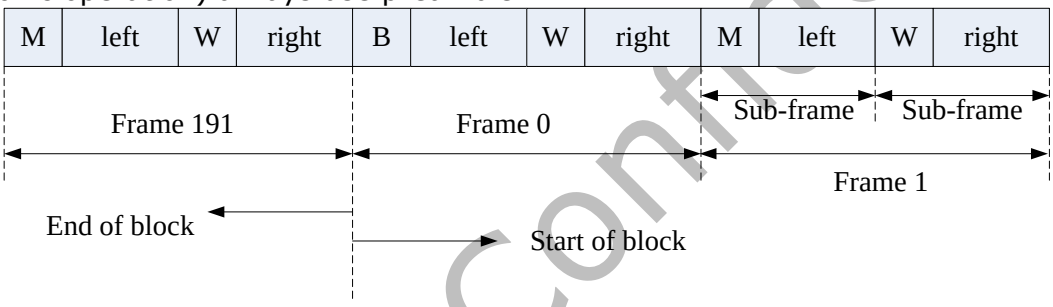


Fig. 14-2 SPDIF Frame Format

In the single channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried only in the first sub-frame and may be duplicated in the second sub-frame. If the second sub-frame is not carrying duplicate data, then time slot 28 (validity flag) shall be set to logical '1' (not valid).

14.3.2 Sub-frame Format

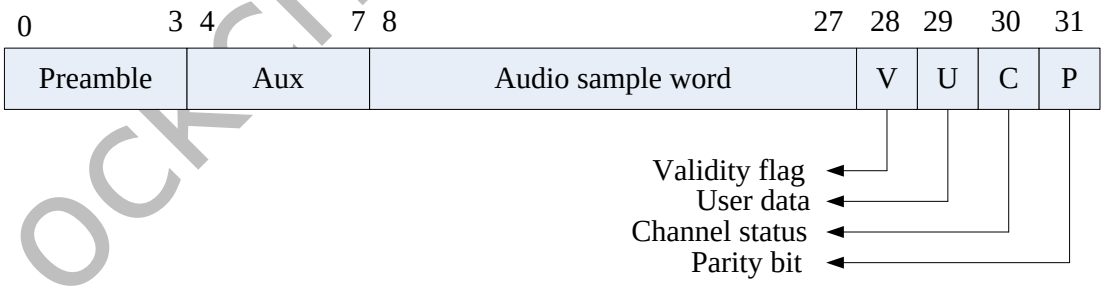


Fig. 14-3 SPDIF Sub-frame Format

Each sub-frame is divided into 32 time slots, numbered from 0 to 31. Time slot 0 to 3 carries one of the three permitted preambles. Time slot 4 to 27 carry the audio sample word in linear 2's complement representation. The MSB is carried by time slot 27. When a 24-bit coding range is used, the LSB is in time slot 4. When a 20-bit coding range is used, time slot 8 to 27 carry the audio sample word with the LSB in time slot 8.Time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.
If the source provides fewer bits than the interface allows (either 24 or 20), the unused LSBs are set to a logical '0'. For a non-linear PCM audio application or a data application the main data field may carry any other information. Time slot 28 carries the validity flag associated with the main data field. Time slot 29 carries 1 bit of the user data associated

with the audio channel transmitted in the same sub-frame. Time slot 30 carries one bit of the channel status words associated with the main data field channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that time slots 4 to 31 inclusive carries an even number of ones and an even number of zeros.

14.3.3 Channel Coding

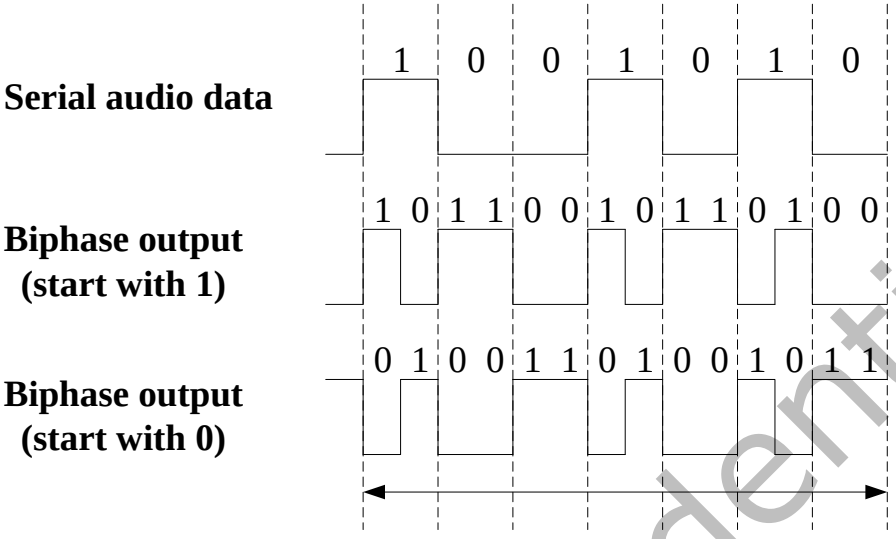


Fig. 14-4 SPDIF Channel Coding

To minimize the direct current component on the transmission line, to facilitate clock recovery from the data stream and to make the interface insensitive to the polarity of connections, time slots 4 to 31 are encoded in biphase-mark.

Each bit to be transmitted is represented by a symbol comprising two consecutive binary states. The first state of a symbol is always different from the second state of the previous symbol. The second state of the symbol is identical to the first if the bit to be transmitted is logical '0'.However, it is different from the first if the bit is logical '1'.

14.3.4 Preamble

Preambles are specific patterns providing synchronization and identification of the sub-frames and blocks.

To achieve synchronization within one sampling period and to make this process completely reliable, these patterns violate the biphase-mark code rules, thereby avoiding the possibility of data imitating the preambles.

A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

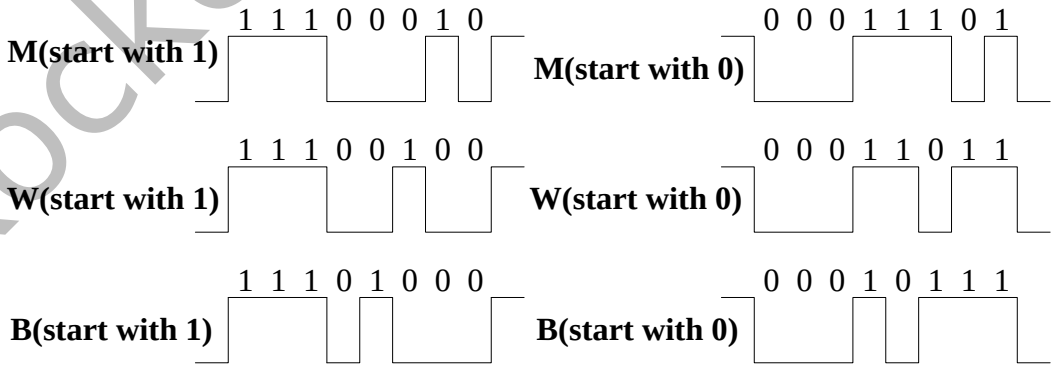


Fig. 14-5 SPDIF Preamble

Like biphase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphase sequence.

14.3.5 NON-LINEAR PCM ENCODED SOURCE(IEC 61937)

The non-linear PCM encoded audio bitstream is transferred using the basic 16-bit data area of the IEC 60958subframes, i.e. in time slots 12 to 27. Each IEC 60958 frame transfers 32-bit of the non-PCM data in consumer application mode.

If the SPDIF bitstream conveys linear PCM audio, the symbol frequency is 64 times the PCM sampling frequency(32 time slots per PCM sample times two channels). If a non-linear PCM encoded audio bitstream is conveyed by the interface, the symbol frequency is 64 times the sampling rate of the encoded audio within that bitstream. But in the case where a non-linear PCM encoded audio bitstream is conveyed by the interface containing audio with low sampling frequency, the symbol frequency is 128 times the sampling rate of the encoded audio within that bitstream.
Each data burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc, Pd), followed by the burst payload which contains data of an encoded audio frame.
The burst-preamble consists of four mandatory fields. Pa and Pb represent a synchronization word. Pc gives information about the type of data and some information/control for the receiver. Pd gives the length of the burst payload, the number of bits or number of bytes according to data-type.
The four preamble words are contained in two sequential SPDIF frames. The frame beginning the data-burst contains preamble word Pa in subframe 0 and Pb in subframe 1. The next frame contains Pc in subframe 0 and Pd in subframe 1. When placed into a SPDIF subframe, the MSB of a 16-bit burst-preamble is placed into timeslot 27 and the LSB is placed into time slot 12.

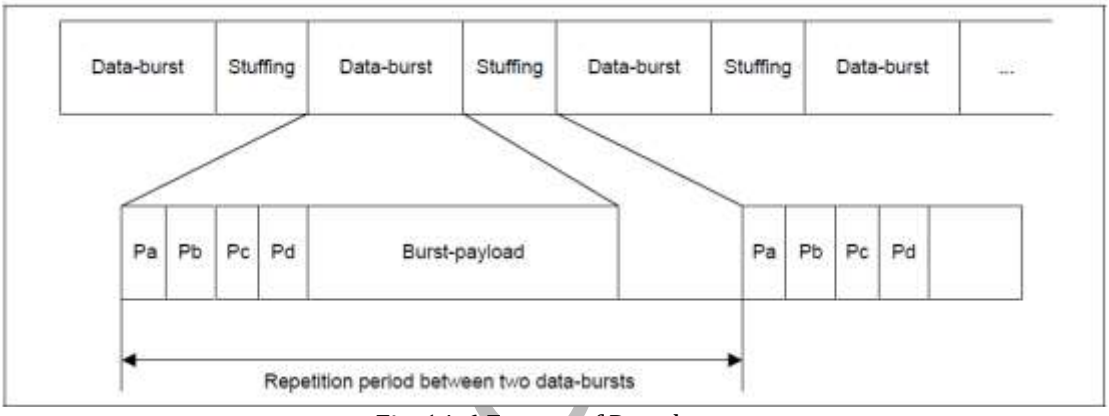


Fig. 14-6 Format of Data-burst

14.4 Register description

14.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SPDIF_CFGR	0x0000	W	0x00000000	Transfer Configuration Register
SPDIF_SDBLR	0x0004	W	0x00000000	Sample Date Buffer Level Register
SPDIF_DMACR	0x0008	W	0x00000000	DMA Control Register
SPDIF_INTCR	0x000c	W	0x00000000	Interrupt Control Register
SPDIF_INTSR	0x0010	W	0x00000000	Interrupt Status Register
SPDIF_XFER	0x0018	W	0x00000000	Transfer Start Register
SPDIF_SMPDR	0x0020	W	0x00000000	Sample Data Register
SPDIF_VLDFRn	0x0060	W	0x00000000	Validity Flag Register n
SPDIF_USRDRn	0x0090	W	0x00000000	User Data Register n
SPDIF_CHNSRn	0x00c0	W	0x00000000	Channel Status Register n
SPDIF_BURTSINFO	0x0100	W	0x00000000	Channel Burst Info Register
SPDIF_REPETTION	0x0104	W	0x00000000	Channel Repetition Register
SPDIF_BURTSINFO_SHD	0x0108	W	0x00000000	Shadow Channel Burst Info Register
SPDIF_REPETTION_SHD	0x010c	W	0x00000000	Shadow Channel Repetition Register
SPDIF_USRDR_SHDn	0x0190	W	0x00000000	Shadow User Data Register n

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

14.4.2 Detail Register Description

SPDIF_CFGR
Address: Operational Base + offset (0x0000)
Transfer Configuration Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	MCD mclk divider Fmclk/Fsdo This parameter can be calculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:9	RO	0x0	reserved
8	RW	0x0	PCMTYPE PCM type 0: linear PCM 1: non-linear PCM
7	WO	0x0	CLR mclk domain logic clear Write 1 to clear mclk domain logic. Read return zero.
6	RW	0x0	CSE Channel status enable 0: disable 1: enable The bit should be set to 1 when the channel conveys non-linear PCM
5	RW	0x0	UDE User data enable 0: disable 1: enable
4	RW	0x0	VFE Validity flag enable 0: disable 1: enable
3	RW	0x0	ADJ audio data justified 0: Right justified 1: Left justified
2	RW	0x0	HWT Halfword word transform enable 0: disable 1: enable It is valid only when the valid data width is 16bit.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	VDW Valid data width 00: 16bit 01: 20bit 10: 24bit 11: reserved The valid data width is 16bit only for non-linear PCM

SPDIF_SDBLR
Address: Operational Base + offset (0x0004)
Sample Date Buffer Level Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	SDBLR Sample Date Buffer Level Register Contains the number of valid data entries in the sample data buffer.

SPDIF_DMACR
Address: Operational Base + offset (0x0008)
DMA Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	TDE Transmit DMA Enable 0: Transmit DMA disabled 1: Transmit DMA enabled
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the Sample Date Buffer is equal to or below this field value

SPDIF_INTCR
Address: Operational Base + offset (0x000c)
Interrupt Control Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	W1C	0x0	UDTIC User Data Interrupt Clear Write '1' to clear the user data interrupt.

Bit	Attr	Reset Value	Description
16	W1C	0x0	BTTIC Block/Data burst transfer finish interrupt clear Write 1 to clear the interrupt.
15:10	RO	0x0	reserved
9:5	RW	0x00	SDBT Sample Date Buffer Threshold Sample Date Buffer Threshold for empty interrupt
4	RW	0x0	SDBEIE Sample Date Buffer empty interrupt enable 0: disable 1: enable
3	RW	0x0	BTTIE Block transfer/repetition period end interrupt enable When enabled, an interrupt will be asserted when the block transfer is finished if the channel conveys linear PCM or when the repetition period is reached if the channel conveys non-linear PCM. 0: disable 1: enable
2	RW	0x0	UDTIE User Data Interrupt 0: disable 1: enable If enabled, an interrupt will be asserted when the content of the user data register is fed into the corresponding shadow register
1:0	RO	0x0	reserved

SPDIF_INTSR

Address: Operational Base + offset (0x0010)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	SDBEIS Sample Date Buffer empty interrupt status 0: inactive 1: active
3	RW	0x0	BTTIS Block/Data burst transfer interrupt status 0: inactive 1: active
2	RW	0x0	UDTIS User Data Interrupt Status 0: inactive 1: active
1:0	RO	0x0	reserved

SPDIF_XFER
Address: Operational Base + offset (0x0018)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	XFER Transfer Start Register Transfer Start Register

SPDIF_SMPDR
Address: Operational Base + offset (0x0020)
Sample Data Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SMPDR Sample Data Register Sample Data Register

SPDIF_VLDFRn
Address: Operational Base + offset (0x0060)
Validity Flag Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	VLDFR_SUB_1 Validity Flag Subframe 1 Validity Flag Register 0
15:0	RW	0x0000	VLDFR_SUB_0 Validity Flag Subframe 0 Validity Flag for Subframe 0

SPDIF_USRDRn
Address: Operational Base + offset (0x0090)
User Data Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	USR_SUB_1 User Data Subframe 1 User Data Bit for Subframe 1
15:0	RW	0x0000	USR_SUB_0 User Data Subframe 0 User Data Bit for Subframe 0

SPDIF_CHNSRn
Address: Operational Base + offset (0x00c0)
Channel Status Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CHNSR_SUB_1 Channel Status Subframe 1 Channel Status Bit for Subframe 1
15:0	RW	0x0000	CHNSR_SUB_0 Channel Status Subframe 0 Channel Status Bit for Subframe 0

SPDIF_BURTSINFO
Address: Operational Base + offset (0x00d0)
Channel Burst Info Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PD pd Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RW	0x0	BSNUM Bitstream Number This field indicates the bitstream number. Usually the bitstream number is 0.
12:8	RW	0x00	DATAINFO Data-type-dependent info This field gives the data-type-dependent info
7	RW	0x0	ERRFLAG Error Flag 0: indicates a valid burst-payload 1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RW	0x00	DATATYPE Data type 0000000: null data 0000001: AC-3 data 0000011: Pause data 0000100: MPEG-1 layer 1 data 0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 0000110: MPEG-2 data with extension 0000111: MPEG-2 AAC 0001000: MPEG-2, layer-1 low sampling frequency 0001001: MPEG-2, layer-2 low sampling frequency 0001010: MPEG-2, layer-3 low sampling frequency 0001011: DTS type I 0001100: DTS type II 0001101: DTS type III 0001110: ATRAC 0001111: ATRAC 2/3 0010000: ATRAC-X 0010001: DTS type IV 0010010: WMA professional type I 0110010: WMA professional type II 1010010: WMA professional type III 1110010: WMA professional type IV 0010011: MPEG-2 AAC low sampling frequency 0110011: MPEG-2 AAC low sampling frequency 1010011: MPEG-2 AAC low sampling frequency 1110011: MPEG-2 AAC low sampling frequency 0010100: MPEG-4 AAC 0110100: MPEG-4 AAC 1010100: MPEG-4 AAC 1110100: MPEG-4 AAC 0010101: Enhanced AC-3 0010110: MAT others: reserved

SPDIF_REPETTION
Address: Operational Base + offset (0x0104)
Channel Repetition Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	REPETTION Repetition This define the repetition period when the channel conveys non-linear PCM

SPDIF_BURTSINFO_SHD
Address: Operational Base + offset (0x0108)
Shadow Channel Burst Info Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	PD pd Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RO	0x0	BSNUM Bitstream Number This field indicates the bitstream number. Usually the birstream number is 0.
12:8	RO	0x00	DATAINFO Data-type-dependent info This field gives the data-type-dependent info
7	RO	0x0	ERRFLAG Error Flag 0: indicates a valid burst-payload 1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RO	0x00	DATATYPE Data type 0000000: null data 0000001: AC-3 data 0000011: Pause data 0000100: MPEG-1 layer 1 data 0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 0000110: MPEG-2 data with extension 0000111: MPEG-2 AAC 0001000: MPEG-2, layer-1 low sampling frequency 0001001: MPEG-2, layer-2 low sampling frequency 0001010: MPEG-2, layer-3 low sampling frequency 0001011: DTS type I 0001100: DTS type II 0001101: DTS type III 0001110: ATRAC 0001111: ATRAC 2/3 0010000: ATRAC-X 0010001: DTS type IV 0010010: WMA professional type I 0110010: WMA professional type II 1010010: WMA professional type III 1110010: WMA professional type IV 0010011: MPEG-2 AAC low sampling frequency 0110011: MPEG-2 AAC low sampling frequency 1010011: MPEG-2 AAC low sampling frequency 1110011: MPEG-2 AAC low sampling frequency 0010100: MPEG-4 AAC 0110100: MPEG-4 AAC 1010100: MPEG-4 AAC 1110100: MPEG-4 AAC 0010101: Enhanced AC-3 0010110: MAT others: reserved

SPDIF_REPETTION_SHD
Address: Operational Base + offset (0x010c)
Shadow Channel Repetition Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	REPETTION Repetition This register provides the repetition of the bitstream when channel conveys non-linear PCM. In the design, it defines the length between Pa of the two consecutive data-burst. For the same audio format, the definition is different. Please convert the actual repetition in order to comply with the design.

SPDIF_USRDR_SHDn
Address: Operational Base + offset (0x0190)
Shadow User Data Register n

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	USR_SUB_1 User Data Subframe 1 User Data Bit for Subframe 1
15:0	RO	0x0000	USR_SUB_0 User Data Subframe 0 User Data Bit for Subframe 0

14.5 Interface description

Table 14-1 SPDIF Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
spdif_8ch_sdo	O	IO_SPDIFtx_GPIO1830gpio4c5	GRF_GPIO4C_IOMUX[11:10]=2'b01
spdif_8ch_sdo	O	IO_MACcol_UART3GPSctsn_SPDIFtxb_GMACgpio3c0	GRF_GPIO3C_IOMUX[1:0]=2'b11

The output of SPDIF module which signals as spdif_8ch_sdo is also connected to the audio interface of HDMI and DP.

Table 14-2 Interface Between SPDIF and HDMI

Module Pin	Direction	Module Pin	Direction
mclk_spdif_8ch	O	ispdifclk	I
spdif_8ch_sdo	O	ispdifdata	I

Table 14-3 Interface Between SPDIF and DP

Module Pin	Direction	Module Pin	Direction
spdif_8ch_sdo	O	source_spdif_din	I

14.6 Application Notes

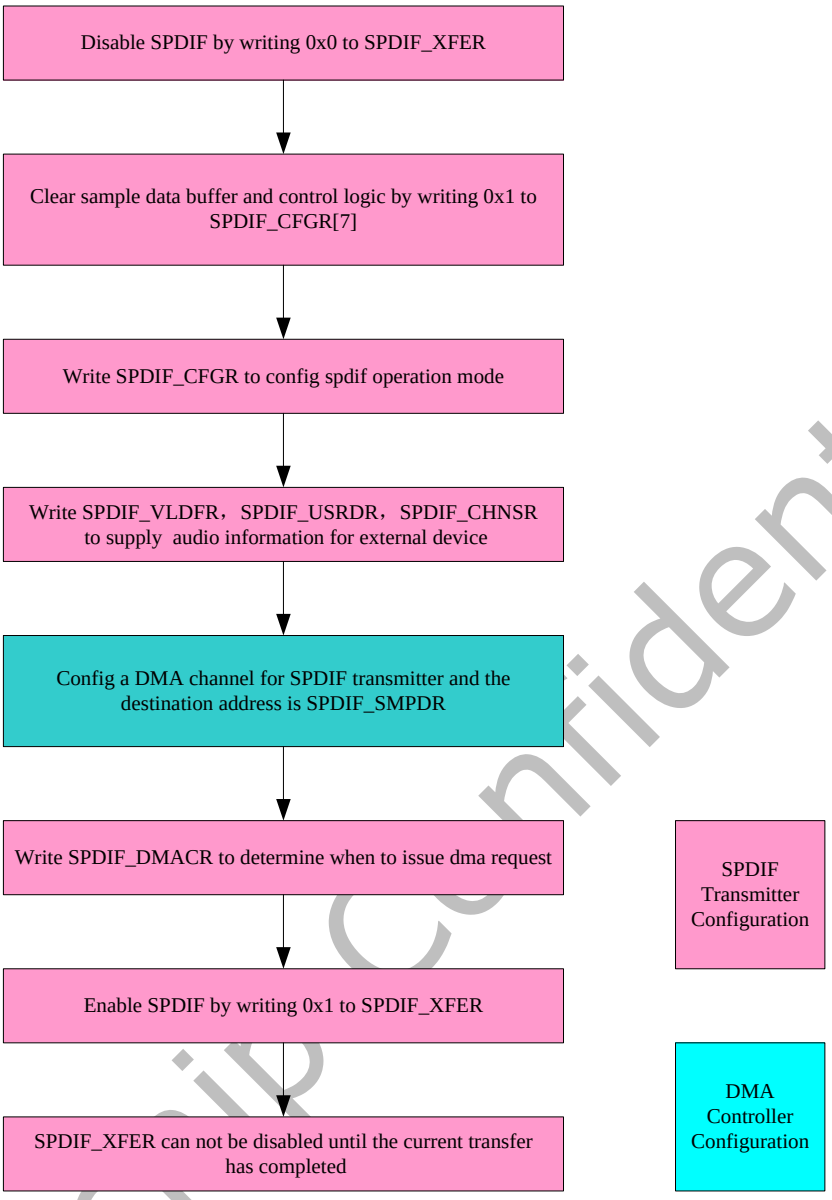


Fig. 14-7 SPDIF transmitter operation flow chart

14.6.1 Channel Status Bit and Validity Flag Bit

Normally the channel status bits and validity flag bits are not necessarily updated frequently. If it is desired to change the channel status bits or validity flag, please write to the corresponding register after a block termination interrupt is asserted. The new value will take effect immediately.

14.6.2 User Data Bit

As the user data bits are updated frequently, the design takes use of the shadow register mechanism to store and convey the user data bit. When the SPDIF interface is disabled, the values of the shadow user data registers keeps the same with the corresponding user data registers. After the SPDIF starts, any change of the user data register will not go to the corresponding shadow user data registers until an user data interrupt is asserted. Therefore before the SPDIF transfer starts, prepare the first 384 user data bits by writing them to the SPDIF_USRDR registers. After the SPDIF transfer starts, writing the second 384 user data bits to the SPDIF_USRDR registers. Then wait for the assertion of user data interrupt. The second 384 user data bits goes to the shadow registers, and then third 384 user bits are written to SPDIF_USRDR.

14.6.3 Burst Info and Repetition

The shadow register mechanism is also applied to the data of burst info and repetition as the user data. The difference is that the update of shadow register will be taken after assertion of the block termination interrupt.
It is important to note that the repetition defined in the design is a little different from the repetition defined in IEC-61957. The repetition is always defined as the length (measured in IEC-60958 frame) between Pa of two consecutive data-bursts. Therefore the user needs to calculate the new repetition value if the definition of the repetition is different for some audio formats such as AC-3.

Chapter 15 GMAC Ethernet Interface

15.1 Overview

The GMAC Ethernet Controller provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) and Reduced Gigabit Media Independent Interface (RGMII) compliant Ethernet PHY.
The GMAC includes a DMA controller. The DMA controller efficiently moves packet data from microprocessor's RAM, formats the data for an IEEE 802.3-2002 compliant packet and transmits the data to an Ethernet Physical Interface (PHY). It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

15.1.1 Feature

- Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces
- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Supports both full-duplex and half-duplex operation
  - Supports CSMA/CD Protocol for half-duplex operation
  - Supports packet bursting and frame extension in 1000 Mbps half-duplex operation
  - Supports IEEE 802.3x flow control for full-duplex operation
  - Optional forwarding of received pause control frames to the user application in full-duplex operation
  - Back-pressure support for half-duplex operation
  - Automatic transmission of zero-quanta pause frame on de-assertion of flow control input in full-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard Ethernet frames
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes:
  - 64-bit Hash filter (optional) for multicast and uni-cast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- MDIO Master interface for PHY device configuration and management
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- Comprehensive status reporting for normal operation and transfers with errors
- Support per-frame Transmit/Receive complete interrupt control
- Supports 4-KB receive FIFO depths on reception.
- Supports 2-KB FIFO depth on transmission
- Automatic generation of PAUSE frame control or backpressure signal to the GMAC core based on Receive FIFO-fill (threshold configurable) level
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- AXI interface to any CPU or memory
- Software can select the type of AXI burst (fixed and variable length burst) in the AXI Master interface
- Supports internal loopback on the RGMII/RMII for debugging

- Debug status register that gives status of FSMs in Transmit and Receive data-paths and FIFO fill-levels.

15.2 Block Diagram

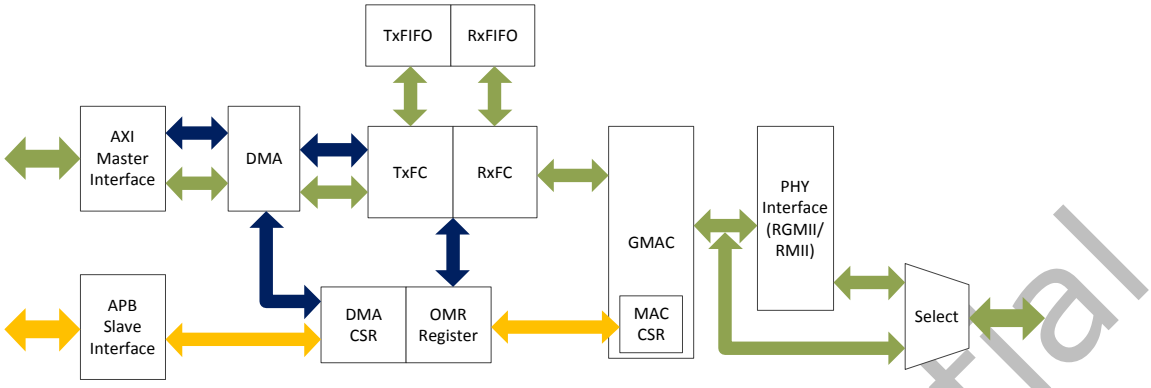


Fig. 15-1 GMACArchitecture

The GMAC is broken up into multiple separate functional units. These blocks are interconnected in the MAC module. The block diagram shows the general flow of data and control signals between these blocks.
The GMAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the GMAC subsystem's control and status registers (CSRs).
The GMAC supports the PHY interfaces of reduced GMII (RGMII) and reduced MII (RMII).
The Transmit FIFO (Tx FIFO) buffers data read from system memory by the DMA before transmission by the GMAC Core. Similarly, the Receive FIFO (Rx FIFO) stores the Ethernet frames received from the line until they are transferred to system memory by the DMA. These are asynchronous FIFOs, as they also transfer the data between the application clock and the GMAC line clocks.

15.3 Function Description

15.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig. 25-2.

Fig. 15-2 MAC Block Diagram

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:
10101010 10101010 10101010 10101010 10101010 10101010 10101010
The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble.The bit value is 10101011.
The data in a well formed frame shall consist of N octet's data.

15.3.2 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.
The RMII module is instantiated between the GMAC and the PHY. This helps translation of the MAC's MII into the RMII. The RMII block has the following characteristics:

- Supports 10-Mbps and 100-Mbps operating rates. It does not support 1000-Mbps operation.
- Two clock references are sourced externally or CRU, providing independent, 2-bit wide transmit and receive paths.

Transmit Bit Ordering

Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Fig.1-3. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

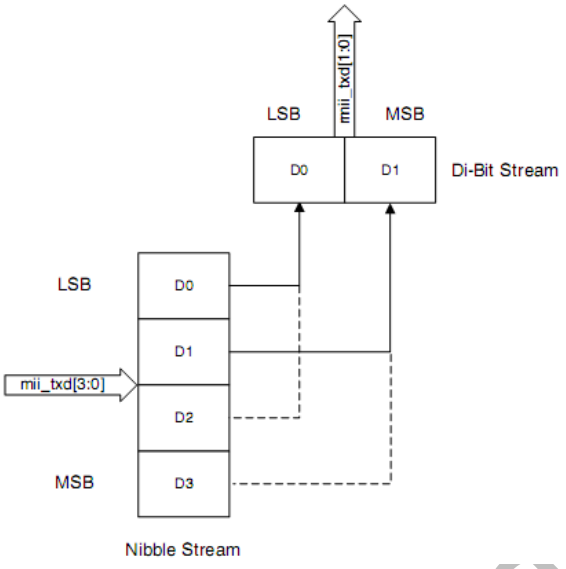


Fig. 15-3 RMII transmission bit ordering

RMII Transmit Timing Diagrams

Fig.1-4 through 1-7 show MII-to-RMII transaction timing.The clk_rmii_i (REF_CLK) frequency is 50MHz in RMII interface.In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on rmii_txd_o[1:0] (TXD[1:0]) shall be valid such that TXD[1:0] may be sampled every 10th cycle,regard-less of the starting cycle within the group and yield the correct frame data.

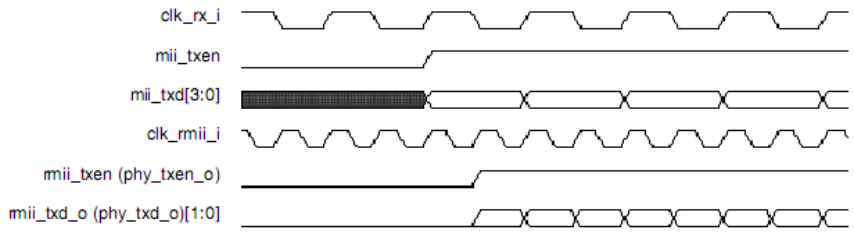


Fig. 15-4 Start of MII and RMII transmission in 100-Mbps mode

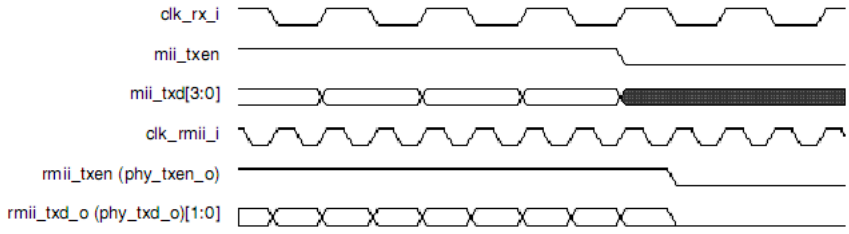


Fig. 15-5 End of MII and RMII Transmission in 100-Mbps Mode

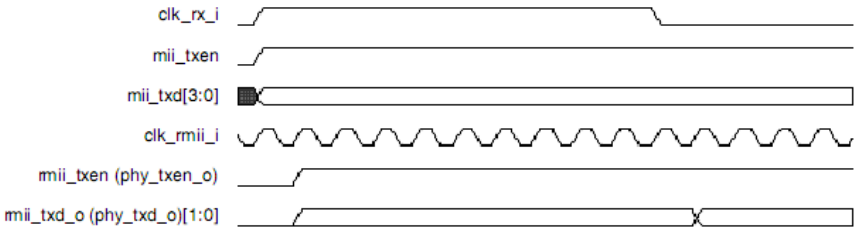


Fig. 15-6Start of MII and RMII Transmission in 10-Mbps Mode

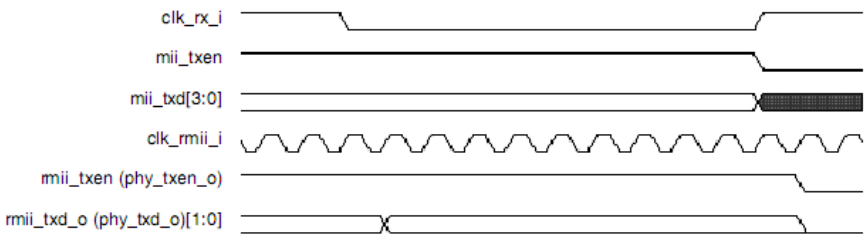


Fig. 15-7End of MII and RMII Transmission in 10-Mbps Mode

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig.1-8. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

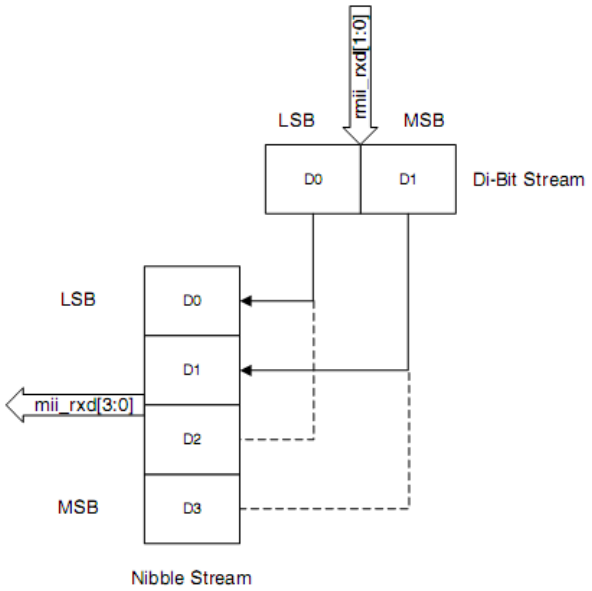


Fig. 15-8 RMII receive bit ordering

15.3.3 RGMII interface

The Reduced Gigabit Media Independent Interface (RGMII) specification reduces the pin count of the interconnection between the GMAC 10/100/1000 controller and the PHY for GMII and MII interfaces. To achieve this, the data path and control signals are reduced and multiplexed together with both the edges of the transmission and receive clocks. For gigabit operation the clocks operate at 125 MHz; for 10/100 operation, the clock rates are 2.5 MHz/25 MHz.

In the GMAC 10/100/1000 controller, the RGMII module is instantiated between the GMAC core's GMII and the PHY to translate the control and data signals between the GMII and RGMII protocols.

The RGMII block has the following characteristics:

- Supports 10-Mbps, 100-Mbps, and 1000-Mbps operation rates.
- For the RGMII block, no extra clock is required because both the edges of the incoming clocks are used.
- The RGMII block extracts the in-band (link speed, duplex mode and link status) status signals from the PHY and provides them to the GMAC core logic for link detection.

15.3.4 Management Interface

The MAC management interface provides a simple, two-wire, serial interface to connect the GMAC and a managed PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information across the MII bus: MDIO and MDC.

The GMAC initiates the management write/read operation. The clock gmii_mdc_o(MDC) is a divided clock from the application clock pclk_gmac. The divide factor depends on the clock range setting in the GMII address register. Clock range is set as follows:

Selection	pclk_gmac	MDC Clock

0000	60-100 MHz	pclk_gmac/42
0001	100-150 MHz	pclk_gmac/62
0010	20-35 MHz	pclk_gmac/16
0011	35-60 MHz	pclk_gmac/26
0100	150-250 MHz	pclk_gmac/102
0101	250-300 MHz	pclk_gmac/124
0110, 0111	Reserved	

The MDC is the derivative of the application clock pclk_gmac. The management operation is performed through the gmii_mdi_i, gmii_mdo_o and gmii_mdo_o_e signals. A three-state buffer is implemented in the PAD.

The frame structure on the MDIO line is shown below.

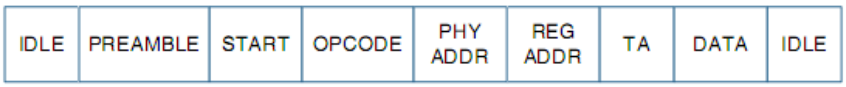


Fig. 15-9 MDIO frame structure

IDLE: The mdio line is three-state; there is no clock on gmii_mdc_o
PREAMBLE: 32 continuous bits of value 1
START: Start-of-frame is 2'b01
OPCODE: 2'b10 for read and 2'b01 for write
PHY ADDR: 5-bit address select for one of 32 PHYs
REG ADDR: Register address in the selected PHY
TA: Turnaround is 2'bZ0 for read and 2'b10 for Write
DATA: Any 16-bit value. In a write operation, the GMAC drives mdio; in a read operation, PHY drives it.

15.3.5 Power Management Block

Power management (PMT) supports the reception of network (remote) wake-up frames and Magic Packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and Magic Packets received by the GMAC. The PMT block sits on the receiver path of the GMAC and is enabled with remote wake-up frame enable and Magic Packet enable. These enables are in the PMT control and status register and are programmed by the application.

When the power down mode is enabled in the PMT, then all received frames are dropped by the core and they are not forwarded to the application. The core comes out of the power down mode only when either a Magic Packet or a Remote Wake-up frame is received and the corresponding detection is enabled.

Remote Wake-Up Frame Detection

When the GMAC is in sleep mode and the remote wake-up bit is enabled in register GMAC_PMT_CTRL_STA (0x002C), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers, by performing a sequential write to address (0028). The application enables remote wake-up by writing a 1 to bit 2 of the register GMAC_PMT_CTRL_STA.

PMT supports four programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of Filter Command, and if Filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received. Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter Byte Mask determines which bytes of the frame must be examined. The thirty-first bit of Byte Mask must be set to zero.

The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, GMII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the register GMAC_PMT_CTRL_STA for every remote Wake-up frame received. A PMT interrupt to the application triggers a read to the GMAC_PMT_CTRL_STA register to determine reception of a wake-up frame.

Magic Packet Detection

The Magic Packet frame is based on a method that uses Advanced Micro Device's Magic Packet technology to power up the sleeping device on the network. The GMAC receives a

specific packet of information, called a Magic Packet, addressed to the node on the network.
Only Magic Packets that are addressed to the device or a broadcast address will be checked to determine whether they meet the wake-up requirements. Magic Packets that pass the address filtering (unicast or broadcast) will be checked to determine whether they meet the remote Wake-on-LAN data format of 6 bytes of all ones followed by a GMAC Address appearing 16 times.
The application enables Magic Packet wake-up by writing a 1 to Bit 1 of the register GMAC_PMT_CTRL_STA. The PMT block constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Each frame received is checked for a 48'hFF_FF_FF_FF_FF_FF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the GMAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 48'hFF_FF_FF_FF_FF_FF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48'hFF_FF_FF_FF_FF_FF). The device will also accept a multicast frame, as long as the 16 duplications of the GMAC address are detected.
If the MAC address of a node is 48'h00_11_22_33_44_55, then the GMAC scans for the data sequence:
Destination Address Source Address …….. FF FFFFFFFFFF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC
Magic Packet detection is updated in the PMT Control and Status register for Magic Packet received. A PMT interrupt to the Application triggers a read to the PMT CSR to determine whether a Magic Packet frame has been received.

15.3.6 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Non-32-bit accesses are allowed as long as the address is word-aligned.
The organization of these registers is shown in Register Description. The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The Register Description in this chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.
The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet frames.

15.4 Register Description

15.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GMAC_MAC_CONF	0x0000	W	0x00000000	MAC Configuration Register
GMAC_MAC_FRM_FILT	0x0004	W	0x00000000	MAC Frame Filter
GMAC_HASH_TAB_HI	0x0008	W	0x00000000	Hash Table High Register
GMAC_HASH_TAB_LO	0x000c	W	0x00000000	Hash Table Low Register
GMAC_GMII_ADDR	0x0010	W	0x00000000	GMII Address Register

Name	Offset	Size	Reset Value	Description
GMAC_GMII_DATA	0x0014	W	0x00000000	GMII Data Register
GMAC_FLOW_CTRL	0x0018	W	0x00000000	Flow Control Register
GMAC_VLAN_TAG	0x001c	W	0x00000000	VLAN Tag Register
GMAC_DEBUG	0x0024	W	0x00000000	Debug register
GMAC_PMT_CTRL_STA	0x002c	W	0x00000000	PMT Control and Status Register
GMAC_INT_STATUS	0x0038	W	0x00000000	Interrupt Status Register
GMAC_INT_MASK	0x003c	W	0x00000000	Interrupt Mask Register
GMAC_MAC_ADDR0_HI	0x0040	W	0x0000ffff	MAC Address0 High Register
GMAC_MAC_ADDR0_LO	0x0044	W	0xffffffff	MAC Address0 Low Register
GMAC_AN_CTRL	0x00c0	W	0x00000000	AN Control Register
GMAC_AN_STATUS	0x00c4	W	0x00000008	AN Status Register
GMAC_AN_ADV	0x00c8	W	0x000001e0	Auto Negotiation Advertisement Register
GMAC_AN_LINK_PART_AB	0x00cc	W	0x00000000	Auto Negotiation Link Partner Ability Register
GMAC_AN_EXP	0x00d0	W	0x00000000	Auto Negotiation Expansion Register
GMAC_INTF_MODE_STA	0x00d8	W	0x00000000	RGMII Status Register
GMAC_MMC_CTRL	0x0100	W	0x00000000	MMC Control Register
GMAC_MMC_RX_INTR	0x0104	W	0x00000000	MMC Receive Interrupt Register
GMAC_MMC_TX_INTR	0x0108	W	0x00000000	MMC Transmit Interrupt Register
GMAC_MMC_RX_INT_MSK	0x010c	W	0x00000000	MMC Receive Interrupt Mask Register
GMAC_MMC_TX_INT_MSK	0x0110	W	0x00000000	MMC Transmit Interrupt Mask Register
GMAC_MMC_TXOCTETCNT_GB	0x0114	W	0x00000000	MMC TX OCTET Good and Bad Counter
GMAC_MMC_TXFRMCNT_GB	0x0118	W	0x00000000	MMC TX Frame Good and Bad Counter
GMAC_MMC_TXUNDFLWERR	0x0148	W	0x00000000	MMC TX Underflow Error
GMAC_MMC_TXCARERR	0x0160	W	0x00000000	MMC TX Carrier Error
GMAC_MMC_TXOCTETCNT_G	0x0164	W	0x00000000	MMC TX OCTET Good Counter
GMAC_MMC_TXFRMCNT_G	0x0168	W	0x00000000	MMC TX Frame Good Counter
GMAC_MMC_RXFRMCNT_GB	0x0180	W	0x00000000	MMC RX Frame Good and Bad Counter
GMAC_MMC_RXOCTETCNT_GB	0x0184	W	0x00000000	MMC RX OCTET Good and Bad Counter
GMAC_MMC_RXOCTETCNT_G	0x0188	W	0x00000000	MMC RX OCTET Good Counter

Name	Offset	Size	Reset Value	Description
GMAC_MMC_RXMCFRMCNT_G	0x0190	W	0x00000000	MMC RX Multicast Frame Good Counter
GMAC_MMC_RXCRCERR	0x0194	W	0x00000000	MMC RX Carrier
GMAC_MMC_RXLENERR	0x01c8	W	0x00000000	MMC RX Length Error
GMAC_MMC_RXFIFOOVRFLW	0x01d4	W	0x00000000	MMC RX FIFO Overflow
GMAC_MMC_IPC_INT_MSK	0x0200	W	0x00000000	MMC Receive Checksum Offload Interrupt Mask Register
GMAC_MMC_IPC_INTR	0x0208	W	0x00000000	MMC Receive Checksum Offload Interrupt Register
GMAC_MMC_RXIPV4GFRM	0x0210	W	0x00000000	MMC RX IPV4 Good Frame
GMAC_MMC_RXIPV4HDERRFRM	0x0214	W	0x00000000	MMC RX IPV4 Head Error Frame
GMAC_MMC_RXIPV6GFRM	0x0224	W	0x00000000	MMC RX IPV6 Good Frame
GMAC_MMC_RXIPV6HDERRFRM	0x0228	W	0x00000000	MMC RX IPV6 Head Error Frame
GMAC_MMC_RXUDPERRFRM	0x0234	W	0x00000000	MMC RX UDP Error Frame
GMAC_MMC_RXTCPERRFRM	0x023c	W	0x00000000	MMC RX TCP Error Frame
GMAC_MMC_RXICMPERRFRM	0x0244	W	0x00000000	MMC RX ICMP Error Frame
GMAC_MMC_RXIPV4HDERROCT	0x0254	W	0x00000000	MMC RX OCTET IPV4 Head Error
GMAC_MMC_RXIPV6HDERROCT	0x0268	W	0x00000000	MMC RX OCTET IPV6 Head Error
GMAC_MMC_RXUDPERROCT	0x0274	W	0x00000000	MMC RX OCTET UDP Error
GMAC_MMC_RXTCPERROCT	0x027c	W	0x00000000	MMC RX OCTET TCP Error
GMAC_MMC_RXICMPERROCT	0x0284	W	0x00000000	MMC RX OCTET ICMP Error
GMAC_BUS_MODE	0x1000	W	0x00020101	Bus Mode Register
GMAC_TX_POLL_DEMAND	0x1004	W	0x00000000	Transmit Poll Demand Register
GMAC_RX_POLL_DEMAND	0x1008	W	0x00000000	Receive Poll Demand Register
GMAC_RX_DESC_LIST_ADDR	0x100c	W	0x00000000	Receive Descriptor List Address Register
GMAC_TX_DESC_LIST_ADDR	0x1010	W	0x00000000	Transmit Descriptor List Address Register
GMAC_STATUS	0x1014	W	0x00000000	Status Register
GMAC_OP_MODE	0x1018	W	0x00000000	Operation Mode Register
GMAC_INT_ENA	0x101c	W	0x00000000	Interrupt Enable Register
GMAC_OVERFLOW_CNT	0x1020	W	0x00000000	Missed Frame and Buffer Overflow Counter Register

Name	Offset	Size	Reset Value	Description
GMAC_REC_INT_WDT_TIMER	0x1024	W	0x00000000	Receive Interrupt Watchdog Timer Register
GMAC_AXI_BUS_MODE	0x1028	W	0x00110001	AXI Bus Mode Register
GMAC_AXI_STATUS	0x102c	W	0x00000000	AXI Status Register
GMAC_CUR_HOST_TX_DESC	0x1048	W	0x00000000	Current Host Transmit Descriptor Register
GMAC_CUR_HOST_RX_DESC	0x104c	W	0x00000000	Current Host Receive Descriptor Register
GMAC_CUR_HOST_TX_BUF_ADDR	0x1050	W	0x00000000	Current Host Transmit Buffer Address Register
GMAC_CUR_HOST_RX_BUF_ADDR	0x1054	W	0x00000000	Current Host Receive Buffer Address Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

15.4.2 Detail Register Description

GMAC_MAC_CONF
Address: Operational Base + offset (0x0000)
MAC Configuration Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	TC Transmit Configuration in RGMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY.
23	RW	0x0	WD Watchdog Disable When this bit is set, the GMAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that.
22	RW	0x0	JD Jabber Disable When this bit is set, the GMAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the GMAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.
21	RW	0x0	BE Frame Burst Enable When this bit is set, the GMAC allows frame bursting during transmission in GMII Half-Duplex mode.
20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19:17	RW	0x0	IFG Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96 bit times 3'b001: 88 bit times 3'b010: 80 bit times ... 3'b111: 40 bit times
16	RW	0x0	DCRS Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.
15	RW	0x0	PS Port Select Selects between GMII and MII: 1'b0: GMII (1000 Mbps) 1'b1: MII (10/100 Mbps)
14	RW	0x0	FES Speed Indicates the speed in Fast Ethernet (MII) mode: 1'b0: 10 Mbps 1'b1: 100 Mbps
13	RW	0x0	DO Disable Receive Own When this bit is set, the GMAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the GMAC receives all packets that are given by the PHY while transmitting.
12	RW	0x0	LM Loopback Mode When this bit is set, the GMAC operates in loopback mode at GMII/MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally.
11	RW	0x0	DM Duplex Mode When this bit is set, the GMAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration.

Bit	Attr	Reset Value	Description
10	RW	0x0	IPC Checksum Offload When this bit is set, the GMAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The GMAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.
9	RW	0x0	DR Disable Retry When this bit is set, the GMAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the GMAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the GMAC will attempt retries based on the settings of BL.
8	RW	0x0	LUD Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RGMII interface: 1'b0: Link Down 1'b1: Link Up
7	RW	0x0	ACS Automatic Pad/CRC Stripping When this bit is set, the GMAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the GMAC will pass all incoming frames to the Host unmodified.

Bit	Attr	Reset Value	Description
6:5	RW	0x0	BL Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is reserved (RO) in Full-Duplex-only configuration. 2'b00: k = min (n, 10) 2'b01: k = min (n, 8) 2'b10: k = min (n, 4) 2'b11: k = min (n, 1), Where n = retransmission attempt. The random integer r takes the value in the range 0 = r < 2^k
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the GMAC. The GMAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmission state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive.
3	RW	0x0	TE Transmitter Enable When this bit is set, the transmission state machine of the GMAC is enabled for transmission on the GMII/MII. When this bit is reset, the GMAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.
2	RW	0x0	RE Receiver Enable When this bit is set, the receiver state machine of the GMAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the GMAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII.
1:0	RO	0x0	reserved

GMAC_MAC_FRM_FILT
Address: Operational Base + offset (0x0004)
MAC Frame Filter

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the GMAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter.
30:11	RO	0x0	reserved
10	RW	0x0	HPF Hash or Perfect Filter When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter.
9	RW	0x0	SAF Source Address Filter Enable The GMAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the GMAC drops the frame. When this bit is reset, then the GMAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison.
8	RW	0x0	SAIF SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.

Bit	Attr	Reset Value	Description
7:6	RW	0x0	PCF Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Register GMAC_FLOW_CTRL[2]. 2'b00: GMAC filters all control frames from reaching the application. 2'b01: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 2'b10: GMAC forwards all control frames to application even if they fail the Address Filter. 2'b11: GMAC forwards control frames that pass the Address Filter.
5	RW	0x0	DBF Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames.
4	RW	0x0	PM Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.
3	RW	0x0	DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.
2	RW	0x0	HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.

Bit	Attr	Reset Value	Description
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.

GMAC_HASH_TAB_HI
Address: Operational Base + offset (0x0008)
Hash Table High Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTH Hash Table High This field contains the upper 32 bits of Hash table

GMAC_HASH_TAB_LO
Address: Operational Base + offset (0x000c)
Hash Table Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTL Hash Table Low This field contains the lower 32 bits of Hash table

GMAC_GMII_ADDR
Address: Operational Base + offset (0x0010)
GMII Address Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:11	RW	0x00	PA Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed
10:6	RW	0x00	GR GMII Register These bits select the desired GMII register in the selected PHY device

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CR APB Clock Range The APB Clock Range selection determines the frequency of the MDC clock as per the pclk_gmac frequency used in your design. The suggested range of pclk_gmac frequency applicable for each value below (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz. **Selection pclk_gmacMDC Clock** 0000 60-100 MHz pclk_gmac/42 0001 100-150 MHz pclk_gmac/62 0010 20-35 MHz pclk_gmac/16 0011 35-60 MHz pclk_gmac/26 0100 150-250 MHz pclk_gmac/102 0101 250-300 MHz pclk_gmac/124 0110, 0111 Reserved When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_gmac is of frequency 100 MHz and you program these bits as "1010", then the resultant MDC clock will be of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Please program the values given below only if the interfacing chips supports faster MDC clocks. **Selection MDC Clock** 1000 pclk_gmac/4 1001 pclk_gmac/6 1010 pclk_gmac/8 1011 pclk_gmac/10 1100 pclk_gmac/12 1101 pclk_gmac/14 1110 pclk_gmac/16 1111 pclk_gmac/18
1	RW	0x0	GW GMII Write When set, this bit tells the PHY that this will be a Write operation using register GMAC_GMII_DATA. If this bit is not set, this will be a Read operation, placing the data in register GMAC_GMII_DATA.

Bit	Attr	Reset Value	Description
0	W1C	0x0	GB GMII Busy This bit should read a logic 0 before writing to Register GMII_ADDR and Register GMII_DATA. This bit must also be set to 0 during a Write to Register GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register GMII_DATA (GMII Data) should be kept valid until this bit is cleared by the GMAC during a PHY Write operation. The Register GMII_DATA is invalid until this bit is cleared by the GMAC during a PHY Read operation. The Register GMII_ADDR (GMII Address) should not be written to until this bit is cleared.

GMAC_GMII_DATA
Address: Operational Base + offset (0x0014)
GMII Data Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GD GMII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.

GMAC_FLOW_CTRL
Address: Operational Base + offset (0x0018)
Flow Control Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.
15:8	RO	0x0	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	PLT Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot-times after the first PAUSE frame is transmitted. **Selection Threshold** 00 Pause time minus 4 slot times 01 Pause time minus 28 slot times 10 Pause time minus 144 slot times 11 Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the GMII/MII interface.
3	RW	0x0	UP Unicast Pause Frame Detect When this bit is set, the GMAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the GMAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.
2	RW	0x0	RFE Receive Flow Control Enable When this bit is set, the GMAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled.
1	RW	0x0	TFE Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the GMAC is disabled, and the GMAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the GMAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled.

Bit	Attr	Reset Value	Description
0	RW	0x0	FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register GMAC_FLOW_CTRL. To initiate a pause control frame, the application must set this bit to 1'b1. During a transfer of the control frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC will reset this bit to 1'b0. The register GMAC_FLOW_CTRL should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the GMAC Core. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function.

GMAC_VLAN_TAG
Address: Operational Base + offset (0x001c)
VLAN Tag Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Frames This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the GMAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.

GMAC_DEBUG
Address: Operational Base + offset (0x0024)
Debug register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	TFIFO3 When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission.
24	RW	0x0	TFIFO2 When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission.
23	RO	0x0	reserved
22	RW	0x0	TFIFO1 When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO.
21:20	RW	0x0	TFIFOSTA This indicates the state of the TxFIFO read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to MAC transmitter) 2'b10: Waiting for TxStatus from MAC transmitter 2'b11: Writing the received TxStatus or flushing the TxFIFO
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state.
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	RFIFORD It gives the state of the RxFIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or time-stamp) 2'b11: Flushing the frame data and Status
4	RW	0x0	RFIFOWR When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO.
3	RO	0x0	reserved
2:1	RW	0x0	ACT When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module
0	RW	0x0	RDB When high, it indicates that the MAC GMII/MII receive protocol engine is actively receiving data and not in IDLE state.

GMAC_PMT_CTRL_STA
Address: Operational Base + offset (0x002c)
PMT Control and Status Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	WFFRPR Wake-Up Frame Filter Register Pointer Reset When set, resets the Remote Wake-up Frame Filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle.
30:10	RO	0x0	reserved
9	RW	0x0	GU Global Unicast When set, enables any unicast packet filtered by the GMAC (DAF) address recognition to be a wake-up frame.
8:7	RO	0x0	reserved
6	RC	0x0	WFR Wake-Up Frame Received When set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a read into this register.
5	RC	0x0	MPR Magic Packet Received When set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a read into this register.
4:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	WFE Wake-Up Frame Enable When set, enables generation of a power management event due to wake-up frame reception.
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception.
0	R/WSC	0x0	PD Power Down When set, all received frames will be dropped. This bit is cleared automatically when a magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must only be set when either the Magic Packet Enable or Wake-Up Frame Enable bit is set high.

GMAC_INT_STATUS
Address: Operational Base + offset (0x0038)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	MRCOIS MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.
6	RO	0x0	MTIS MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.
5	RO	0x0	MRIS MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.
4	RO	0x0	MIS MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low. This bit is valid only when the optional MMC module is selected during configuration.

Bit	Attr	Reset Value	Description
3	RO	0x0	PIS PMT Interrupt Status This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power-Down mode). This bit is cleared when both bits[6:5] are cleared due to a read operation to the register GMAC_PMT_CTRL_STA.
2:1	RO	0x0	reserved
0	RO	0x0	RIS RGMII Interrupt Status This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation the RGMII Status register.

GMAC_INT_MASK
Address: Operational Base + offset (0x003c)
Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	PIM PMT Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PMT Interrupt Status bit in Register GMAC_INT_STATUS.
2:1	RO	0x0	reserved
0	RW	0x0	RIM RGMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit in Register GMAC_INT_STATUS.

GMAC_MAC_ADDR0_HI
Address: Operational Base + offset (0x0040)
MAC Address0 High Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0xffff	A47_A32 MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_MAC_ADDR0_LO
Address: Operational Base + offset (0x0044)
MAC Address0 Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	A31_A0 MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_AN_CTRL
Address: Operational Base + offset (0x00c0)
AN Control Register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	ANE Auto-Negotiation Enable When set, will enable the GMAC to perform auto-negotiation with the link partner. Clearing this bit will disable auto-negotiation.
11:10	RO	0x0	reserved
9	R/W SC	0x0	RAN Restart Auto-Negotiation When set, will cause auto-negotiation to restart if the ANE is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation.
8:0	RO	0x0	reserved

GMAC_AN_STATUS
Address: Operational Base + offset (0x00c4)
AN Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	ANC Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is completed. This bit is cleared when auto-negotiation is reinitiated.
4	RO	0x0	reserved
3	RO	0x1	ANA Auto-Negotiation Ability This bit is always high, because the GMAC supports auto-negotiation.
2	R/W SC	0x0	LS Link Status When set, this bit indicates that the link is up. When cleared, this bit indicates that the link is down.
1:0	RO	0x0	reserved

GMAC_AN_ADV
Address: Operational Base + offset (0x00c8)
Auto Negotiation Advertisement Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support This bit is tied to low, because the GMAC does not support the next page.
14	RO	0x0	reserved
13:12	RW	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating to a link partner that a fault or error condition has occurred.
11:9	RO	0x0	reserved
8:7	RW	0x3	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the GMAC is capable of configuring the PAUSE function as defined in IEEE 802.3x.
6	RW	0x1	HD Half-Duplex This bit, when set high, indicates that the GMAC supports Half-Duplex. This bit is tied to low (and RO) when the GMAC is configured for Full-Duplex-only operation.
5	RW	0x1	FD Full-Duplex This bit, when set high, indicates that the GMAC supports Full-Duplex.
4:0	RO	0x0	reserved

GMAC_AN_LINK_PART_AB
Address: Operational Base + offset (0x00cc)
Auto Negotiation Link Partner Ability Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired.

Bit	Attr	Reset Value	Description
14	RO	0x0	ACK Acknowledge When set, this bit is used by the auto-negotiation function to indicate that the link partner has successfully received the GMAC's base page. When cleared, it indicates that a successful receipt of the base page has not been achieved.
13:12	RO	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating a fault or error condition of the link partner.
11:9	RO	0x0	reserved
8:7	RO	0x0	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the link partner's capability of configuring the PAUSE function as defined in IEEE 802.3x.
6	RO	0x0	HD Half-Duplex When set, this bit indicates that the link partner has the ability to operate in Half-Duplex mode. When cleared, the link partner does not have the ability to operate in Half-Duplex mode.
5	RO	0x0	FD Full-Duplex When set, this bit indicates that the link partner has the ability to operate in Full-Duplex mode. When cleared, the link partner does not have the ability to operate in Full-Duplex mode.
4:0	RO	0x0	reserved

GMAC_AN_EXP
Address: Operational Base + offset (0x00d0)
Auto Negotiation Expansion Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	NPA Next Page Ability This bit is tied to low, because the GMAC does not support next page function.
1	RO	0x0	NPR New Page Received When set, this bit indicates that a new page has been received by the GMAC. This bit will be cleared when read.
0	RO	0x0	reserved

GMAC_INTF_MODE_STA
Address: Operational Base + offset (0x00d8)

RGMII Status Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	LST Link Status Indicates whether the link is up (1'b1) or down (1'b0)
2:1	RO	0x0	LSD Link Speed Indicates the current speed of the link: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz
0	RW	0x0	LM Link Mode Indicates the current mode of operation of the link: 1'b0: Half-Duplex mode 1'b1: Full-Duplex mode

GMAC_MMC_CTRL

Address: Operational Base + offset (0x0100)

MMC Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	FHP Full-Half preset When low and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2K Bytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16) When high and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2K Bytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/W SC	0x0	CP Counters Preset When set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full.

Bit	Attr	Reset Value	Description
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	R/WSC	0x0	CR Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle

GMAC_MMC_RX_INTR
Address: Operational Base + offset (0x0104)
MMC Receive Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 The bit is set when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved
18	RC	0x0	INT18 The bit is set when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 The bit is set when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RC	0x0	INT4 The bit is set when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RC	0x0	INT2 The bit is set when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
1	RC	0x0	INT1 The bit is set when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INTR
Address: Operational Base + offset (0x0108)
MMC Transmit Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RC	0x0	INT21 The bit is set when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RC	0x0	INT20 The bit is set when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RC	0x0	INT19 The bit is set when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RX_INT_MSK
Address: Operational Base + offset (0x010c)
MMC Receive Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RW	0x0	INT18 Setting this bit masks the interrupt when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RW	0x0	INT4 Setting this bit masks the interrupt when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RW	0x0	INT2 Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INT_MSK
Address: Operational Base + offset (0x0110)
MMC Transmit Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RW	0x0	INT20 Setting this bit masks the interrupt when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RW	0x0	INT19 Setting this bit masks the interrupt when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	INT13 Setting this bit masks the interrupt when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TXOCTETCNT_GB
Address: Operational Base + offset (0x0114)
MMC TX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_gb Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames.

GMAC_MMC_TXFRMCNT_GB
Address: Operational Base + offset (0x0118)
MMC TX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_gb Number of good and bad frames transmitted, exclusive of retried frames.

GMAC_MMC_TXUNDFLWERR
Address: Operational Base + offset (0x0148)
MMC TX Underflow Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txunderflowerror Number of frames aborted due to frame underflow error.

GMAC_MMC_TXCARERR
Address: Operational Base + offset (0x0160)
MMC TX Carrier Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txcarriererror Number of frames aborted due to carrier sense error (no carrier or loss of carrier).

GMAC_MMC_TXOCTETCNT_G
Address: Operational Base + offset (0x0164)
MMC TX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_g Number of bytes transmitted, exclusive of preamble, in good frames only.

GMAC_MMC_TXFRMCNT_G
Address: Operational Base + offset (0x0168)
MMC TX Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_g Number of good frames transmitted.

GMAC_MMC_RXFRMCNT_GB
Address: Operational Base + offset (0x0180)
MMC RX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxframecount_gb Number of good and bad frames received.

GMAC_MMC_RXOCTETCNT_GB
Address: Operational Base + offset (0x0184)
MMC RX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_gb Number of bytes received, exclusive of preamble, in good and bad frames.

GMAC_MMC_RXOCTETCNT_G
Address: Operational Base + offset (0x0188)
MMC RX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_g Number of bytes received, exclusive of preamble, only in good frames.

GMAC_MMC_RXMCFRMCNT_G
Address: Operational Base + offset (0x0190)
MMC RX Mulitcast Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxmulticastframes_g Number of good multicast frames received.

GMAC_MMC_RXCRCERR
Address: Operational Base + offset (0x0194)
MMC RX Carrier

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxcrcerror Number of frames received with CRC error.

GMAC_MMC_RXLENERR
Address: Operational Base + offset (0x01c8)
MMC RX Length Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxlengtherror Number of frames received with length error (Length type field ≠frame size), for all frames with valid length field.

GMAC_MMC_RXFIFOOVRFLW
Address: Operational Base + offset (0x01d4)
MMC RX FIFO Overflow

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxfifooverflow Number of missed received frames due to FIFO overflow.

GMAC_MMC_IPC_INT_MSK
Address: Operational Base + offset (0x0200)
MMC Receive Checksum Offload Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	INT29 Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RW	0x0	INT27 Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RW	0x0	INT25 Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RW	0x0	INT22 Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
21:18	RO	0x0	reserved
17	RW	0x0	INT17 Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RW	0x0	INT11 Setting this bit masks the interrupt when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RW	0x0	INT9 Setting this bit masks the interrupt when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RW	0x0	INT6 Setting this bit masks the interrupt when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_IPC_INTR
Address: Operational Base + offset (0x0208)
MMC Receive Checksum Offload Interrupt Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RC	0x0	INT29 The bit is set when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RC	0x0	INT27 The bit is set when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RC	0x0	INT25 The bit is set when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RC	0x0	INT22 The bit is set when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21:18	RO	0x0	reserved
17	RC	0x0	INT17 The bit is set when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RC	0x0	INT11 The bit is set when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RC	0x0	INT9 The bit is set when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RC	0x0	INT6 The bit is set when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RC	0x0	INT5 The bit is set when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RC	0x0	INT1 The bit is set when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RXIPV4GFRM
Address: Operational Base + offset (0x0210)
MMC RX IPV4 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_gd_frms Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

GMAC_MMC_RXIPV4HDERRFRM
Address: Operational Base + offset (0x0214)
MMC RX IPV4 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_frms Number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors

GMAC_MMC_RXIPV6GFRM
Address: Operational Base + offset (0x0224)
MMC RX IPV6 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_gd_frms Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads.

GMAC_MMC_RXIPV6HDERRFRM
Address: Operational Base + offset (0x0228)
MMC RX IPV6 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_frms Number of IPv6 datagrams received with header errors (length or version mismatch).

GMAC_MMC_RXUDPERRFRM
Address: Operational Base + offset (0x0234)
MMC RX UDP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_frms Number of good IP datagrams whose UDP payload has a checksum error.

GMAC_MMC_RXTCPERRFRM
Address: Operational Base + offset (0x023c)
MMC RX TCP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_frms Number of good IP datagrams whose TCP payload has a checksum error.

GMAC_MMC_RXICMPERRFRM
Address: Operational Base + offset (0x0244)
MMC RX ICMP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_frms Number of good IP datagrams whose ICMP payload has a checksum error.

GMAC_MMC_RXIPV4HDERROCT
Address: Operational Base + offset (0x0254)
MMC RX OCTET IPV4 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_octets Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter.

GMAC_MMC_RXIPV6HDERROCT
Address: Operational Base + offset (0x0268)
MMC RX OCTET IPV6 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_octets Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter.

GMAC_MMC_RXUDPERROCT
Address: Operational Base + offset (0x0274)
MMC RX OCTET UDP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_octets Number of bytes received in a UDP segment that had checksum errors.

GMAC_MMC_RXTCPERROCT
Address: Operational Base + offset (0x027c)
MMC RX OCTET TCP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_octets Number of bytes received in a TCP segment with checksum errors.

GMAC_MMC_RXICMPERROCT
Address: Operational Base + offset (0x0284)
MMC RX OCTET ICMP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_octets Number of bytes received in an ICMP segment with checksum errors.

GMAC_BUS_MODE
Address: Operational Base + offset (0x1000)
Bus Mode Register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	AAL Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.
24	RW	0x0	PBL_Mode 8xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
23	RW	0x0	USP Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines.

Bit	Attr	Reset Value	Description
22:17	RW	0x01	RPBL RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high.
16	RW	0x0	FB Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AXI will use SINGLE and INCR burst transfer operations.
15:14	RO	0x0	reserved
13:8	RW	0x01	PBL Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only. The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified (as given below). For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly. For TxFIFO, valid PBL range in full duplex mode and duplex mode is 128 or less. For RxFIFO, valid PBL range in full duplex mode is all.
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:2	RW	0x00	DSL Descriptor Skip Length This bit specifies the number of dword to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.
1	RO	0x0	reserved
0	R/W SC	0x1	SWR Software Reset When this bit is set, the MAC DMA Controller resets all GMAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion.

GMAC_TX_POLL_DEMAND
Address: Operational Base + offset (0x1004)
Transmit Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TPD Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_TX_DESC. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register GMAC_STATUS[2] is asserted. If the descriptor is available, transmission resumes.

GMAC_RX_POLL_DEMAND
Address: Operational Base + offset (0x1008)
Receive Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RPD Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_RX_DESC. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register GMAC_STATUS[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state.

GMAC_RX_DESC_LIST_ADDR
Address: Operational Base + offset (0x100c)
Receive Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SRL Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_TX_DESC_LIST_ADDR
Address: Operational Base + offset (0x1010)
Transmit Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STL Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_STATUS
Address: Operational Base + offset (0x1014)
Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	GPI GMAC PMT Interrupt This bit indicates an interrupt event in the GMAC core's PMT module. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
27	RO	0x0	GMI GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the GMAC core. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.

Bit	Attr	Reset Value	Description
26	RO	0x0	GLI GMAC Line interface Interrupt This bit reflects an interrupt event in the GMAC Core's PCS or RGMII interface block. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
25:23	RO	0x0	EB Error Bits These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (Register GMAC_STATUS[13]) set. This field does not generate an interrupt. Bit 23: 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA Bit 24: 1'b1 Error during read transfer 1'b0 Error during write transfer Bit 25: 1'b1 Error during descriptor access 1'b0 Error during data buffer access
22:20	RO	0x0	TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor.

Bit	Attr	Reset Value	Description
19:17	RO	0x0	RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory.
16	W1C	0x0	NIS Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.
15	W1C	0x0	AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive FIFO Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.

Bit	Attr	Reset Value	Description
14	W1C	0x0	ERI Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register GMAC_STATUS[6] automatically clears this bit.
13	W1C	0x0	FBI Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses.
12:11	RO	0x0	reserved
10	W1C	0x0	ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.
9	W1C	0x0	RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received.
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.
7	W1C	0x0	RU Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register GMAC_STATUS[7] is set only when the previous Receive Descriptor was owned by the DMA.
6	W1C	0x0	RI Receive Interrupt This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.
5	W1C	0x0	UNF Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.

Bit	Attr	Reset Value	Description
4	W1C	0x0	OVF Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11].
3	W1C	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.
2	W1C	0x0	TU Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command.
1	W1C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped.
0	W1C	0x0	TI Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor.

GMAC_OP_MODE
Address: Operational Base + offset (0x1018)
Operation Mode Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	DT Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset.

Bit	Attr	Reset Value	Description
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.
23:22	RO	0x0	reserved
21	RW	0x0	TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register GMAC_OP_MODE[16:14] are ignored. This bit should be changed only when transmission is stopped.
20	W1C	0x0	FTF Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.
19:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TTC Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16
13	RW	0x0	ST Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register GMAC_TX_DESC_LIST_ADDR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register GMAC_STATUS[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register TX_DESC_LIST_ADDR, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state.
12:11	RW	0x0	RFD Threshold for deactivating flow control (in both HD and FD) These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is de-asserted after activation. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the de-assertion is effective only after flow control is asserted.

Bit	Attr	Reset Value	Description
10:9	RW	0x0	RFA Threshold for activating flow control (in both HD and FD) These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high.
8	RW	0x0	EFC Enable HW flow control When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled.
7	RW	0x0	FEF Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is written, then such frames are dropped even when FEF is set.
6	RW	0x0	FUF Forward Undersized Good Frames When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01).
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:3	RW	0x0	RTC Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 2'b00: 64 2'b01: 32 2'b10: 96 2'b11: 128
2	RW	0x0	OSF Operate on Second Frame When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained.
1	RW	0x0	SR Start/Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by register GMAC_RX_DESC_LIST_ADDR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register GMAC_STATUS[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting register GMAC_RX_DESC_LIST_ADDR, DMA behavior is unpredictable. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.
0	RO	0x0	reserved

GMAC_INT_ENA
Address: Operational Base + offset (0x101c)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NIE Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt
15	RW	0x0	AIE Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error
14	RW	0x0	ERE Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled.
13	RW	0x0	FBE Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.
12:11	RO	0x0	reserved
10	RW	0x0	ETE Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (BIT 15), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.

Bit	Attr	Reset Value	Description
9	RW	0x0	RWE Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.
8	RW	0x0	RSE Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.
7	RW	0x0	RUE Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.
5	RW	0x0	UNE Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.
4	RW	0x0	OVE Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled
3	RW	0x0	TJE Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.
2	RW	0x0	TUE Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.

Bit	Attr	Reset Value	Description
1	RW	0x0	TSE Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.

GMAC_OVERFLOW_CNT
Address: Operational Base + offset (0x1020)
Missed Frame and Buffer Overflow Counter Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RC	0x0	FIFO_overflow_bit Overflow bit for FIFO Overflow Counter
27:17	RC	0x000	Frame_miss_number Indicates the number of frames missed by the application This counter is incremented each time the MTL asserts the sideband signal mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.
16	RC	0x0	Miss_frame_overflow_bit Overflow bit for Missed Frame Counter
15:0	RC	0x0000	Frame_miss_number_2 Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.

GMAC_REC_INT_WDT_TIMER
Address: Operational Base + offset (0x1024)
Receive Interrupt Watchdog Timer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	RIWT RI Watchdog Timer count Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame.

GMAC_AXI_BUS_MODE
Address: Operational Base + offset (0x1028)
AXI Bus Mode Register

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable LPI (Low Power Interface) When set to 1, enable the LPI (Low Power Interface) supported by the GMAC and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller.
30	RW	0x0	UNLCK_ON_MGK_RWK Unlock on Magic Packet or Remote Wake Up When set to 1, enables it to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enables it requests to come out of Low Power mode when any frame is received.
29:22	RO	0x0	reserved
21:20	RW	0x1	WR_OSR_LMT AXI Maximum Write Out Standing Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1
19:18	RO	0x0	reserved
17:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Out Standing Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RO	0x0	AXI_AAL Address-Aligned Beats This bit is read-only bit and reflects the AAL bit (register GMAC_BUS_MODE[25]). When this bit set to 1, it performs address-aligned burst transfers on both read and write channels.
11:4	RO	0x0	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16.
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8.
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4.
0	RO	0x1	UNDEF AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of FB bit in register GMAC_BUS_MODE[16]. When this bit is set to 1, it is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1]; When this bit is set to 0, it is allowed to perform only fixed burst lengths as indicated by BLEN256/128/64/32/16/8/4, or a burst length of 1.

GMAC_AXI_STATUS
Address: Operational Base + offset (0x102c)
AXI Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	RD_CH_STA When high, it indicates that AXI Master's read channel is active and transferring data.
0	RO	0x0	WR_CH_STA When high, it indicates that AXI Master's write channel is active and transferring data.

GMAC_CUR_HOST_TX_DESC
Address: Operational Base + offset (0x1048)
Current Host Transmit Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTDAP Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_DESC
Address: Operational Base + offset (0x104c)
Current Host Receive Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRDAP Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_TX_Buf_ADDR
Address: Operational Base + offset (0x1050)
Current Host Transmit Buffer Address Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTBAP Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_BUF_ADDR
Address: Operational Base + offset (0x1054)
Current Host Receive Buffer Adderss Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRBAP Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

15.5 Interface Description

Table 15-1 RMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RMII interface			
mac_clk	I/O	IO_MACclk_I2C5TRACKPADscl_GMACgpio3b3	GRF_GPIO3B_IOMUX[7:6]=2'b10
mac_txen	O	IO_MACtxen_UART1BBsin_GMACgpio3b4	GRF_GPIO3B_IOMUX[9:8]=2'b10
mac_txd1	O	IO_MACtxd1_SPI0NORCODECtxd_GMACgpio3a5	GRF_GPIO3A_IOMUX[11:10]=2'b10
mac_txd0	O	IO_MACtxd0_SPI0NORCODECrxd_GMACgpio3a4	GRF_GPIO3A_IOMUX[9:8]=2'b10
mac_rxdv	I	IO_MACrxdv_GMACgpio3b1	GRF_GPIO3B_IOMUX[3:2]=2'b10
mac_rxer	I	IO_MACrxer_I2C5TRACKPADsda_GMACgpio3b2	GRF_GPIO3B_IOMUX[5:4]=2'b10
mac_rxd1	I	IO_MACrxd1_SPI0NORCODECcsn0_GMACgpio3a7	GRF_GPIO3A_IOMUX[15:14]=2'b10
mac_rxd0	I	IO_MACrxd0_SPI0NORCODECclk_GMACgpio3a6	GRF_GPIO3A_IOMUX[13:12]=2'b10
Management interface			

mac_mdio	I/O	IO_MACmdio_UART1BBsout_GMACgpio3b5	GRF_GPIO3B_IOMUX[11:10]=2'b10
mac_mdc	O	IO_MACmdc_SPI0NORCODECcsn1_GMACgpio3b0	GRF_GPIO3B_IOMUX[1:0]=2'b10

Table 15-2 RGMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RGMII/RMII interface			
mac_clk	I/O	IO_MACclk_I2C5TRACKPADscl_GMACgpio3b3	GRF_GPIO3B_IOMUX[7:6]=2'b10
mac_txclk	O	IO_MACtxclk_UART3GPSrtsn_GMACgpio3c1	GRF_GPIO3C_IOMUX[3:2]=2'b10
mac_txen	O	IO_MACtxen_UART1BBsin_GMACgpio3b4	GRF_GPIO3B_IOMUX[9:8]=2'b10
mac_txd3	O	IO_MACtxd3_SPI4EXPtxd_TRACEdata13_GMACgpio3a1	GRF_GPIO3A_IOMUX[3:2]=2'b10
mac_txd2	O	IO_MACtxd2_SPI4EXPrxd_TRACEdata12_GMACgpio3a0	GRF_GPIO3A_IOMUX[1:0]=2'b10
mac_txd1	O	IO_MACtxd1_SPI0NORCODECtxd_GMACgpio3a5	GRF_GPIO3A_IOMUX[11:10]=2'b10
mac_txd0	O	IO_MACtxd0_SPI0NORCODECrxd_GMACgpio3a4	GRF_GPIO3A_IOMUX[9:8]=2'b10
mac_rxclk	I	IO_MACrxclk_UART3GPSsin_GMACgpio3b6	GRF_GPIO3B_IOMUX[13:12]=2'b10
mac_rxdv	I	IO_MACrxdv_GMACgpio3b1	GRF_GPIO3B_IOMUX[3:2]=2'b10
mac_rxd3	I	IO_MACrxd3_SPI4EXPcsn0_TRACEdata15_GMACgpio3a3	GRF_GPIO3A_IOMUX[7:6]=2'b10
mac_rxd2	I	IO_MACrxd2_SPI4EXPclk_TRACEdata14_GMACgpio3a2	GRF_GPIO3A_IOMUX[5:4]=2'b10
mac_rxd1	I	IO_MACrxd1_SPI0NORCODECcsn0_GMACgpio3a7	GRF_GPIO3A_IOMUX[15:14]=2'b10
mac_rxd0	I	IO_MACrxd0_SPI0NORCODECclk_GMACgpio3a6	GRF_GPIO3A_IOMUX[13:12]=2'b10
mac_crs	I	IO_MACcrs_UART3GPSsout_CIFclkoutb_GMACgpio3b7	GRF_GPIO3B_IOMUX[15:14]=2'b10
mac_col	I	IO_MACcol_UART3GPSctsn_SPDIFtxb_GMACgpio3c0	GRF_GPIO3C_IOMUX[1:0]=2'b10
Management interface			
mac_mdio	I/O	IO_MACmdio_UART1BBsout_GMACgpio3b5	GRF_GPIO3B_IOMUX[11:10]=2'b10
mac_mdc	O	IO_MACmdc_SPI0NORCODECcsn1_GMACgpio3b0	GRF_GPIO3B_IOMUX[1:0]=2'b10

Notes: I=input, O=output, I/O=input/output, bidirectional

15.6 Application Notes

15.6.1 Descriptors

The DMA in GMAC can communicate with Host driver through descriptor lists and data buffers. The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.
There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Registers RX_DESC_LIST_ADDR and TX_DESC_LIST_ADDR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists

resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.
A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled
The descriptor ring and chain structure is shown in following figure.

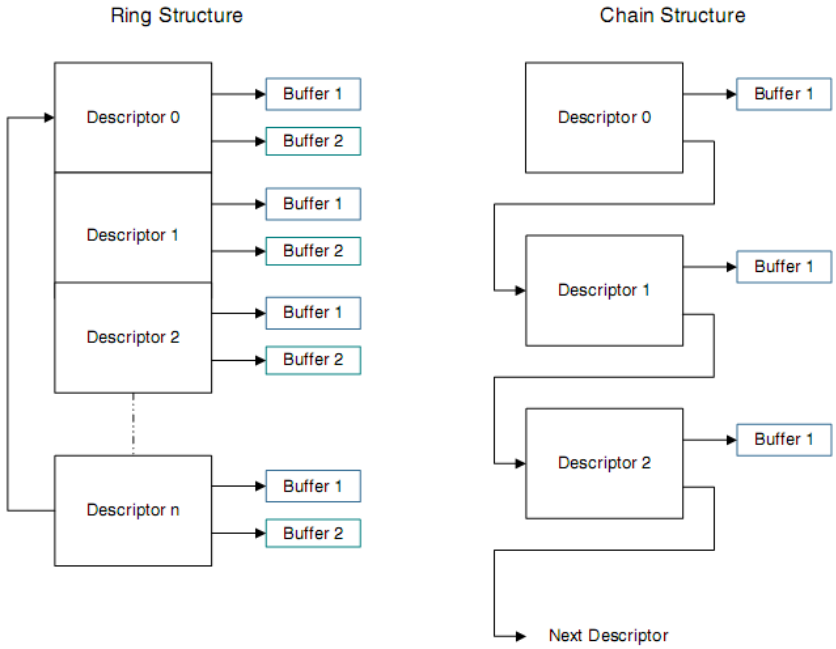


Fig. 15-10 Descriptor Ring and Chain Structure

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. The descriptor addresses must be aligned to the bus width used (Word/Dword/Lword for 32/64/128-bit buses).

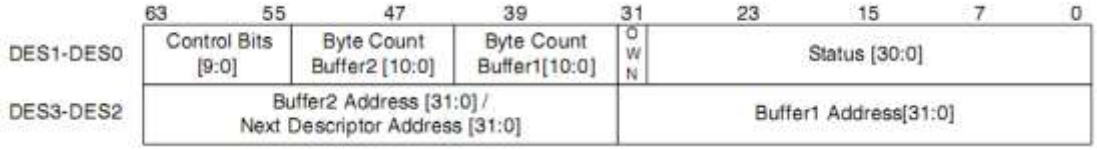


Fig. 15-11 Rx/Tx Descriptors definition

15.6.2 Receive Descriptor

The GMAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMAalways attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received.
In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Receive Descriptor 0 (RDES0)
RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 15-3 Receive Descriptor 0

Bit	Description

31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC Core.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES: Error Summary Indicates the logical OR of the following bits: • RDES0[0]: Payload Checksum Error • RDES0[1]: CRC Error • RDES0[3]: Receive Error • RDES0[4]: Watchdog Timeout • RDES0[6]: Late Collision • RDES0[7]: IPC Checksum • RDES0[11]: Overflow Error • RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC Core.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the GMAC Core.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
7	IPC Checksum Error/Giant Frame When IP Checksum Engine is enabled, this bit, when set, indicates that the 16-bit

	IPv4 Header checksum calculated by the core did not match the received checksum bytes. The Error Summary bit[15] is NOT set when this bit is set in this mode.
6	LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.
5	FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
4	RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
3	RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension, or error (rxd≠ 0f) during extension.
2	DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.
1	CE: CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	Rx MAC Address/Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.

Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain/ring.

Table 15-4 Receive Descriptor 1

Bit	Description
31	Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the GMAC_STATUS Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30:26	Reserved.
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a "don't care" value. RDES1[25] takes precedence over RDES1[24].

23:22	Reserved.
21:11	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. This field is not valid if RDES1[24] is set.
10:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Receive Descriptor 2 (RDES2)

RDES2 contains the address pointer to the first data buffer in the descriptor.

Table 15-5 Receive Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[2:0] (corresponding to bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Receive Descriptor 3 (RDES3)

RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Table 15-6 Receive Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[2:0] = 0, corresponding to a bus width of 64. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[2:0] (corresponding to a bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

15.6.3 Transmit Descriptor

The descriptor addresses must be aligned to the bus width used (64). Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.

Transmit Descriptor 0 (TDES0)

TDES0 contains the transmitted frame status and the descriptor ownership information.

Table 15-7 Transmit Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30:17	Reserved.
16	IHE: IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion.
15	ES: Error Summary Indicates the logical OR of the following bits: • TDES0[14]: Jabber Timeout • TDES0[13]: Frame Flush • TDES0[11]: Loss of Carrier • TDES0[10]: No Carrier • TDES0[9]: Late Collision • TDES0[8]: Excessive Collision • TDES0[2]: Excessive Deferral • TDES0[1]: Underflow Error
14	JT: Jabber Timeout When set, this bit indicates the GMAC transmitter has experienced a jabber time-out.
13	FF: Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode.
11	LC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission. This is valid only for the frames transmitted without collision and when the GMAC operates in Half-Duplex Mode.
10	NC: No Carrier When set, this bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in RMII Mode and 512 byte times including Preamble and Carrier Extension in RGMII Mode). Not valid if Underflow Error is set.
8	EC: Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable

	Retry) bit in the GMAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode) if the Deferral Check (DC) bit is set high in the GMAC Control Register.
1	UF: Underflow Error When set, this bit indicates that the GMAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register GMAC_STATUS[5]) and Transmit Interrupt (Register GMAC_STATUS [0]).
0	DB: Deferred Bit When set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Table 15-8 Transmit Descriptor 1

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.
26	DC: Disable CRC When set, the GMAC does not append the Cyclic Redundancy Check (CRC) to the

	end of the transmitted frame. This is valid only when the first segment (TDES1[29]).
25	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
24	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are "don't care" values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the GMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	Reserved.
21:11	TBS2: Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Table 15-9 Transmit Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 15-10 Transmit Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

15.6.4 Programming Guide

DMA Initialization – Descriptors

The following operations must be performed to initialize the DMA.

1. Provide a software reset. This will reset all of the GMAC internal registers and logic. (GMAC_OP_MODE[0]).
2. Wait for the completion of the reset process (poll GMAC_OP_MODE[0], which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the Bus Mode Register by setting values in register GMAC_BUS_MODE
   a. Mixed Burst and AAL
   b. Fixed burst or undefined burst

c. Burst length values and burst mode values.
d. Descriptor Length (only valid if Ring Mode is used)
e. Tx and Rx DMA Arbitration scheme

4. Program the AXI Interface options in the register GMAC_BUS_MODE
a. If fixed burst-length is enabled, then select the maximum burst-length possible on the AXI bus (Bits[7:1])

5. A proper descriptor chain for transmit and receive must be created. It should also ensure that the receive descriptors are owned by DMA (bit 31 of descriptor should be set). When OSF mode is used, at least two descriptors are required.

6. Software should create three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.

7. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (register GMAC_RX_DESC_LIST_ADDR and GMAC_TX_DESC_LIST_ADDR).

8. Program the following fields to initialize the mode of operation by setting values in register GMAC_OP_MODE
a. Receive and Transmit Store And Forward
b. Receive and Transmit Threshold Control (RTC and TTC)
c. Hardware Flow Control enable
d. Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
e. Error Frame and undersized good frame forwarding enable
f. OSF Mode

9. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit 16 - normal interrupt summary will clear this bit (register GMAC_STATUS).

10. Enable the interrupts by programming the interrupt enable register GMAC_INT_ENA.

11. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the control register GMAC_OP_MODE.

MAC Initialization

The following MAC Initialization operations can be performed after the DMA initialization sequence. If the MAC Initialization is done before the DMA is set-up, then enable the MAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the RxFIFO and overflow.

1. Program the register GMAC_GMII_ADDR for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.

2. Read the 16-bit data of (GMAC_GMII_DATA) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in registerGMAC_GMII_ADDR (bits 15-11).

3. Provide the MAC address registers (GMAC_MAC_ADDR0_HI and GMAC_MAC_ADDR0_LO).

4. If Hash filtering is enabled in your configuration, program the Hash filter register (GMAC_HASH_TAB_HI and GMAC_HASH_TAB_LO).

5. Program the following fields to set the appropriate filters for the incoming frames in register GMAC_MAC_FRM_FILT
a. Receive All
b. Promiscuous mode
c. Hash or Perfect Filter
d. Unicast, Multicast, broad cast and control frames filter settings etc.

6. Program the following fields for proper flow control in register GMAC_FLOW_CTRL.
a. Pause time and other pause frame control bits
b. Receive and Transmit Flow control bits
c. Flow Control Busy/Backpressure Activate

7. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.

8. Program the appropriate fields in register GMAC_MAC_CONF for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the Duplex mode (bit 11), port select (bit 15), etc.
9. Set the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.

Normal Receive and Transmit Operation
For normal operation, the following steps can be followed.

- For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
- On completion of the above step, set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
- If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (GMAC_TX_POLL_DEMAND and GMAC_RX_POLL_DEMAND).
- The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (GMAC_CUR_HOST_TX_DESC and GMAC_CUR_HOST_RX_DESC).
- The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (GMAC_CUR_HOST_TX_Buf_ADDR and GMAC_CUR_HOST_RX_BUF_ADDR).

Stop and Start Operation
When the transmission is required to be paused for some time then the following steps can be followed.
1. Disable the Transmit DMA (if applicable), by clearing ST (bit 13) of the control register GMAC_OP_MODE.
2. Wait for any previous frame transmissions to complete. This can be checked by reading the appropriate bits of MAC Debug register.
3. Disable the MAC transmitter and MAC receiver by clearing the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.
4. Disable the Receive DMA (if applicable), after making sure the data in the RX FIFO is transferred to the system memory (by reading the register GMAC_DEBUG).
5. Make sure both the TX FIFO and RX FIFO are empty.
6. To re-start the operation, start the DMAs first, before enabling the MAC Transmitter and Receiver.

15.6.5 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure.
The mux select rmii_speed is GRF_SOC_CON1[11].

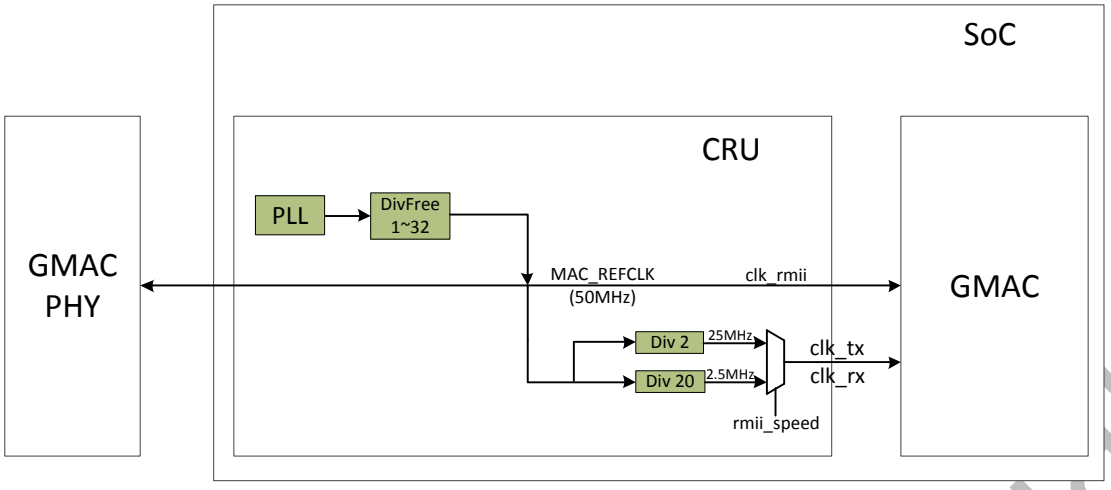


Fig. 15-12 RMII clock architecture when clock source from CRU

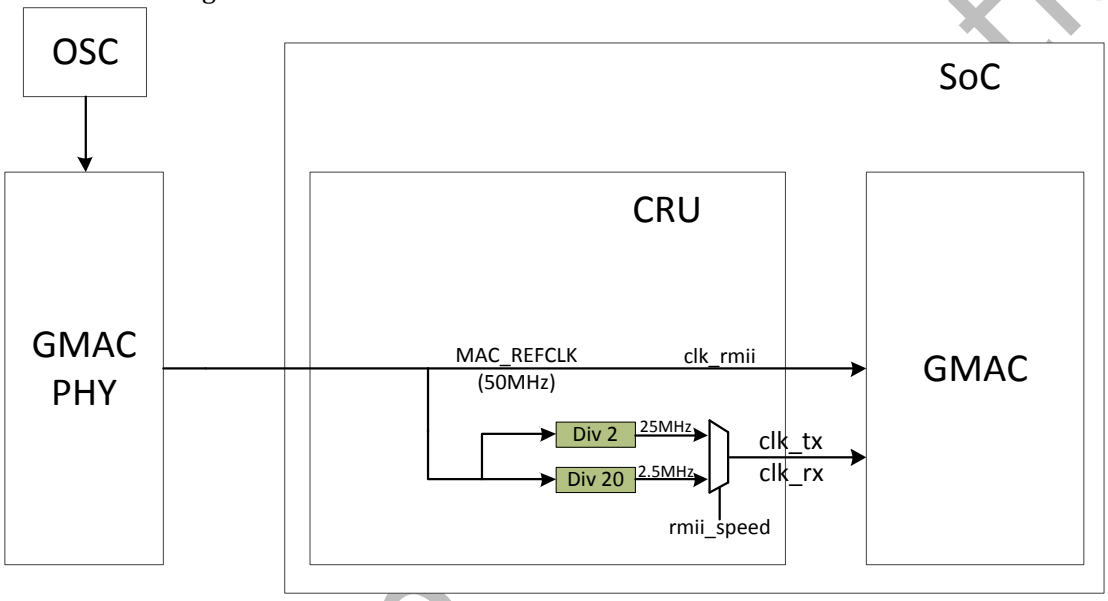


Fig. 15-13 RMII clock architecture when clock source from external OSC

In RGMII mode, clock architecture only supports that TX clock source is from CRU as following figure.

In order to dynamically adjust the timing between TX/RX clocks with data, deleyline is integrated in TX and RX clock path. Register GRF_SOC_CON3[15:14] can enable the deleylines, and GRF_SOC_CON3[13:0] is used to determine the delay length. There are 100 deley elements in each delayline.

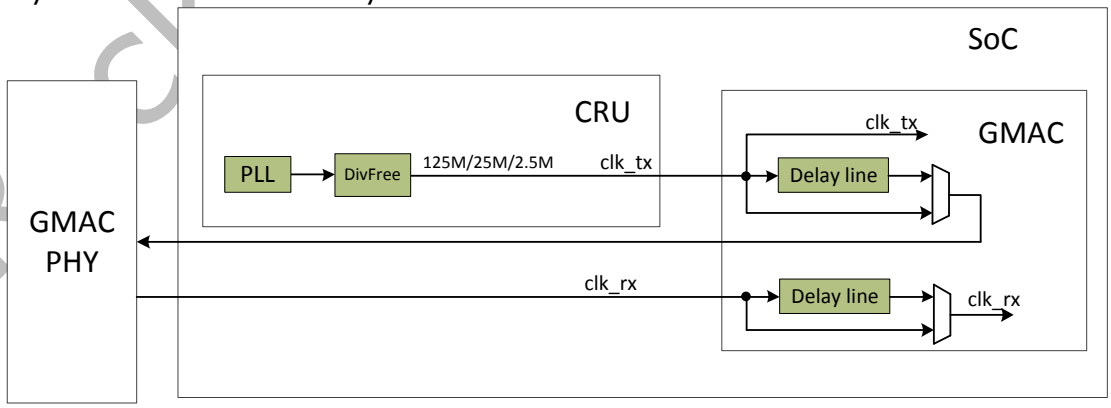


Fig. 15-14 RGMII clock architecture when clock source from CRU

15.6.6 Remote Wake-Up Frame Filter Register

The register wkupfmfilter_reg, address (028H), loads the Wake-up Frame Filter register. To load values in a Wake-up Frame Filter register, the entire register (wkupfmfilter_reg) must

be written. The wkupfmfilter_reg register is loaded by sequentially loading the eight register values in address (028) for wkupfmfilter_reg0, wkupfmfilter_reg1, ..., wkupfmfilter_reg7, respectively. Wkupfmfilter_reg is read in the same way.
The internal counter to access the appropriate wkupfmfilter_reg is incremented when lane3 (or lane 0 in big-endian) is accessed by the CPU. This should be kept in mind if you are accessing these registers in byte or half-word mode.

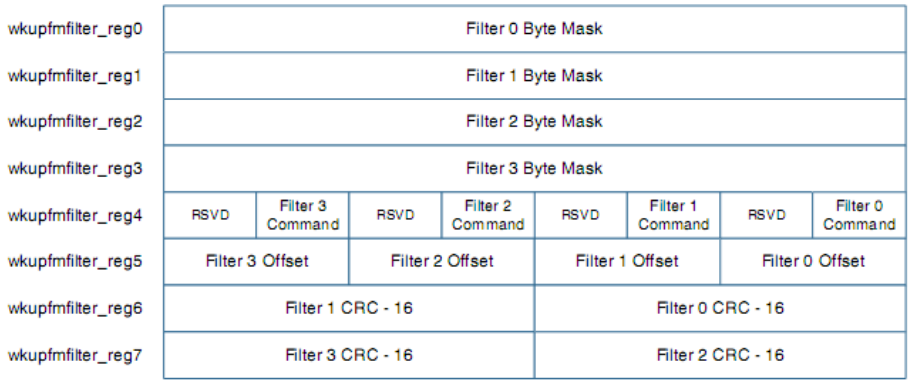


Fig. 15-15 Wake-Up Frame Filter Register

Filter i Byte Mask
This register defines which bytes of the frame are examined by filter i (0, 1, 2, and 3) in order to determine whether or not the frame is a wake-up frame. The MSB (thirty-first bit) must be zero. Bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then Filter i Offset + j of the incoming frame is processed by the CRC block; otherwise Filter i Offset + j is ignored.
Filter i Command
This 4-bit command controls the filter i operation. Bit 3 specifies the address type, defining the pattern's destination address type. When the bit is set, the pattern applies to only multicast frames; when the bit is reset, the pattern applies only to unicast frame. Bit 2 and Bit 1 are reserved. Bit 0 is the enable for filter i; if Bit 0 is not set, filter i is disabled.
Filter i Offset
This register defines the offset (within the frame) from which the frames are examined by filter i. This 8-bit pattern-offset is the offset for the filter i first byte to examined. The minimum allowed is 12, which refers to the 13th byte of the frame (offset value 0 refers to the first byte of the frame).
Filter i CRC-16
This register contains the CRC_16 value calculated from the pattern, as well as the byte mask programmed to the wake-up filter register block.

15.6.7 System Consideration During Power-Down

GMAC neither gates nor stops clocks when Power-Down mode is enabled. Power saving by clock gating must be done outside the core by the CRU. The receive data path must be clocked with clk_rx_i during Power-Down mode, because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the APB path clocks can be gated off during Power-Down mode.
The PMT interrupt is asserted when a valid wake-up frame is received. This interrupt is generated in the clk_rx domain.
The recommended power-down and wake-up sequence is as follows.
1. Disable the Transmit DMA (if applicable) and wait for any previous frame transmissions to complete. These transmissions can be detected when Transmit Interrupt (TI - Register GMAC_STATUS[0]) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC Configuration register.
3. Wait until the Receive DMA empties all the frames from the Rx FIFO (a software timer may be required).
4. Enable Power-Down mode by appropriately configuring the PMT registers.
5. Enable the MAC Receiver and enter Power-Down mode.

6. Gate the APB and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter Sleep mode.
7. On receiving a valid wake-up frame, the GMAC asserts the PMT interrupt signal and exits Power-Down mode.
8. On receiving the interrupt, the system must enable the APB and transmit clock inputs to the core.
9. Read the register GMAC_PMT_CTRL_STA to clear the interrupt, then enable the other modules in the system and resume normal operation.

15.6.8 GRF Register Summary

GRF Register	Register Description
GRF_MAC_CON1[6:4]	PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
GRF_MAC_CON1[3]	GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
GRF_MAC_CON1[2]	GMACspeed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON1[7]	RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
GRF_MAC_CON1[9:8]	RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
GRF_MAC_CON1[10]	RMII mode selection 1'b1: RMII mode 1'b0: Reserved
GRF_MAC_CON0[6:0]	RGMII TX clock delayline value
GRF_MAC_CON0[13:7]	RGMII RX clock delayline value
GRF_MAC_CON1[0]	RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
GRF_MAC_CON1[1]	RGMII RX clock delayline enable 1'b1: enable 1'b0: disable

Chapter 16 eMMC Host Controller

16.1 Overview

The "eMMC Host Controller" is a Host Controller with an AXI processor interface. This product conforms to JEDEC's eMMC Specification Version 5.0 and SD Host Controller Standard Specification Version 3.00 and JEDEC's MMC Specification Version 5.1 (Draft). The Host Controller handles eMMC/SDIO/SD Protocol at transmission level, packing data, adding cyclic redundancy check (CRC), Start/End bit, and checking for transaction format correctness.

The Host Controller provides Programmed IO method and DMA data transfer method. In programmed IO method, the Host processor transfers data using the Buffer Data Port Register. Host controller support for DMA can be determined by checking the DMA support in the Capabilities register. DMA allows a peripheral to read or write memory without the intervention from the CPU. The Host Controller's system address register points to the first data address, and data is then accessed sequentially from that address.

The Host Controller supports following features:

- Compliance
  - eMMC Specification version 5.0 JESD84_b50
  - SD Host Controller Standard Specification Version 3.00
  - SDIO card specification Version 3.0
  - SD Memory Card Specification Version 3.01
  - SD Memory Card Security Specification version 1.01
  - CQHCI specification version 0.7 (Draft)
  - AMBA AXI Specification version 3.00
- System/Host Interface
  - Supports AXI Host Bus Master interface and AHB Host Bus Slave interface
  - Data transfer using PIO mode on the Host Bus Slave interface, using DMA mode on the Host Bus Master interface
- eMMC Card Interface
  - Host clock rate variable between 0 and 200 MHz
  - Up to 3200Mbits per second data rate using 8 parallel data lines (eMMC HS400)
  - Up to 1600Mbits per second data rate using 8 parallel data lines (eMMC HS200)
  - Up to 832Mbits per second data rate using 8 parallel data lines (eMMC DDR52 mode)
  - Transfers the data in 1 bit, 4 bit and 8 bit modes
  - Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
  - Supports eMMC Plus and eMMC Mobile
- SD/ SDIO Card interface
  - Host clock rate variable between 0 and 200 MHz
  - Up to 832Mbits per second data rate using 4 parallel data lines (SD's SDR104 mode)
  - Transfers the data in 1 bit and 4 bit SD modes
  - Transfers the data in SDR104, SDR50, DDR50 modes
  - Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
  - Variable-length data transfers
  - Performs Read wait Control, Suspend/Resume operation SDIO CARD
  - Designed to work with I/O cards, Read-only cards and Read/Write cards
  - Supports Read wait Control, Suspend/Resume operation
- Miscellaneous
  - Configurable FIFO used to aid data transfer between the CPU and the Controller, with FIFO depth 128 and width 64
  - Handle the FIFO Overrun and Underrun condition by stopping SD clock

16.2 Block Diagram

The block diagram of Host Controller is shown below.

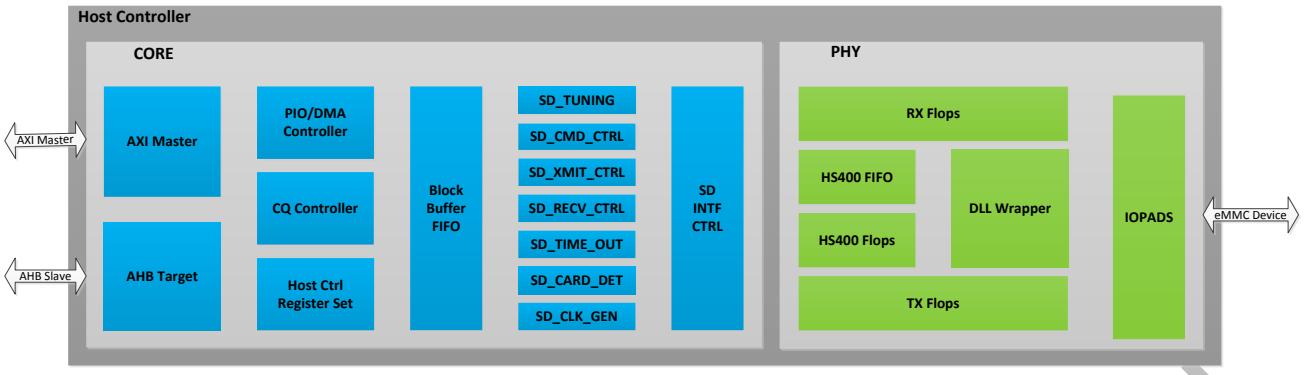


Fig. 16-1 Host Controller Block Diagram

16.3 Function Description

16.3.1 Host Controller Core Block Description

Host interface (Master/Target)

The Host Controller interfaces to the System bus using the AXI Master and Target Interface. The Target Bus is used to access the Registers inside the Host controller. Also when operating in PIO mode, the Driver can access the SD Data Port Register thru this interface. This is the PIO method in which the Host Driver transfers data using the Buffer Data Port Register. The Target Bus supports only single transfer access (no Burst Support). Also in case of AXI Interface, the Target Bus supports only one outstanding read/write transaction. The Master Bus is used by the DMA Controller (when using DMA or ADMA2 Modes). The DMA Controller uses the Master DMA Interface to transfer data between the internal Block Buffer and the System Memory and vice-versa. The DMA Controller also uses the Master Interface to fetch the descriptors while operating in ADMA2 mode. The CQ controller also uses the Master Interface to fetch task descriptors when CQ is enabled.

Host Controller Register Set

The Host Controller Register set implements the Registers defined by the SD Host Controller Specification (Version 3.00). The Registers are Byte/DWORD accessible from the Target Interface. The Host Controller Register Set also implements the Data Port Registers for the PIO Mode transfers. The Register Set provides the Control Signals to rest of the Blocks in design and monitors the status signals from the blocks to set Interrupt Status Bits and eventually generate Interrupt signal to the Host Bus.

The Host Controller Register Set acts as the bridge between CPU and Host Controller. The controller registers are programmed by the Host Processor through Host Target interface. Interrupts are generated to the Host Processor based on the values set in the Interrupt status register and Interrupt enable registers.

PIO/DMA Controller

The PIO/DMA Controller Module implements the SDMA and ADMA2 Engines as defined in the SD Host Controller Specification and maintains the block transfer counts for PIO operation. It interacts with the Registers Set and starts the DMA Engine when a Command with Data Transfer is involved. The DMA Controller interfaces to the Host Master Module to generate Transfers and on the other side it interfaces with the Block Buffer to store/fetch block data. The DMA Controller implements a Separate DMA for SDMA Operation and Separate DMA for the ADMA2 Operation. In addition it implements Host Transaction Generator that generates controls for the Host Master Interface Module.

CQ Controller

CQ Controller Module implements Command Queuing engine as defined the CQHCI specification. Once the CQE is enabled its main functions are: processing task information provided by software, communication with the device using the bus protocol for issuing tasks and ordering task execution, copying data to/from the system memory, and generation of interrupts.

The CQE interfaces with CQ registers and receives tasks from software via task descriptors in system memory. The CQE issues CQ commands to the eMMC device and also stores the task's information. CQE also reads the device's queue status register, decides which task to execute and issues the EXECUTE(Data transfer) commands. CQE feeds the task's transfer

descriptor(s) to the DMA engine as a pointer to the data buffer in the host memory. The CQ interfaces to the Host Master Module via the DMA Controller to generate Transfers.

Block Buffer

The Host Controller uses a Dual Port Block Buffer (read/write on both ports) that is used to store the Block Data during SD Transfers. The Block Buffer uses Circular Buffer Architecture. One side of the Block Buffer is interfaced with the DMA Controller and operates on the Host Clock. The other side of the Block Buffer interfaces with SD Control Logic and operates on SD Clock. During a write transaction (data transferred from CPU Processor to SD3.0 / SDIO3.0 / eMMC5.1 card), the data is fetched from the System Memory and is stored in the Block Buffer. When a Block of data is available, the SD Control logic will transfer it onto the SD Interface. The DMA Controller continues to fetch additional block of data when the Block Buffer has space. During a read transaction (data transferred from card to CPU Processor), the data from card will be written in to block buffer and at the end when the CRC of the Block is valid, the data is committed. When a block of data is available, then the DMA Controller transfers this data to the System Memory. The SD Control logic meanwhile receives the next block of data provided there is space in the Block Buffer. If the Host controller cannot accept any data from card, then it will issue read wait (if card supports read wait mechanism) to stop the data transfer from card or by stopping the clock.

eMMC Clock Generator

The Clock Generator module generates the SD Clock from the Reference Clock (xin_clk), based on the Controls programmed in the Clock Control Register. These include the Clock Divide Value, SD Clock Enable etc. The outputs from this module are the SD_CLK and the SD_CARD Clock. The SD_CLK is used by the SD Control Logic and the SD_CARD Clock connected to the "CLK" Pin on the SD Interface. This module also generates system resets to various clock domains.

eMMC/SD Timeout Control

The SD Timeout Control logic implements the Timeout Check between Block Transfers. It uses the Contents of the Timeout Control Register to implement timeout between Blocks. This module operates under the control of the Transmit Control and Receive Control Modules (based on direction) When the Timeout is detected the event is reported to Transmit Control or Receive Control module.

eMMC/SD Command Control

The SD Command Control module generates the Command Sequence on the CMD line of the SD Interface for every new command programmed by the Software. The Command Control module also implements the Response Reception and checking the validity of the Response. It uses the Response type field to determine the length of the response and the presence of CRC7 field. The Response is received on the Receive Clock (which is either the looped back clock or the tuned Clock). Once the response is received the contents of the Response (Start Bit, Command Index, CRC7, End Bit) are verified and response status is forwarded to the Register set module for setting various status bits. It also implements the Timeout Check on the Response Reception to make sure that the Response is received with in the defined time (5 or 64 Clocks based on Command Type). The received Response is then stored into appropriate bit position in the Response Register.

The SD Command Control module generates controls to the SD Transmit Control and SD Receive Control based on the Transfer Direction The SD Command Control module also generates Auto Command (AutoCMD12 or AutoCMD23) when enabled.

eMMC/SD Transmit Control

The SD Transmit control Module is used for write transfers for transferring data to the Card. Once the Command is issued, this module waits for the Block of data to be available in the Block Buffer and transfers this onto the SD DAT lines. Based on the configuration of data lines, the (1-bit, 4-bit or 8-bits), the data from Block Buffer is appropriately routed. The CRC16 is individually calculated on per lane basis and is attached at the end of block transfer before the END Bit. In case of DDR operation, it implements separate CRC16 for each edge of the clock.

At the end of Block transfer, it waits for the CRC Response on DAT0 line and reports the result of the CRC check to the Register Set.

Also this module checks for the Write Busy indication (DAT0 Line) before transferring next block of data. The Timeout Check is implemented to make sure that the Write Busy is asserted no more than the required limit.

eMMC/SD Receive Control

The SD Receive control Module is used for read transfers for receiving data from the Card. Once the Command is issued, this module waits for the Block of data to be received from the Card. Based on the configuration of data lines, the (1-bit, 4-bit or 8-bits), the data from SD interface is assembled into byte and eventually into 32-bit word before it is being written into Block Buffer is. The CRC16 is individually calculated on per lane basis and is checked against the received CRC16 at the end of block transfer before the END Bit. In case of DDR operation, it implements separate CRC16 checker for each edge of the clock. The data is received on the receive clock. This receive clock is either the Looped back Clock (SDCARD_CLK from the IO_BUF) or the Tuned Clock using DLL or DLY elements.
The Timeout Check is implemented to make sure that the gap between the block no more than the required limit.

eMMC/SD Tuning Block

The SD Tuning Block is used for SDR104 or SDR50 (optionally when enabled) and eMMC HS200 modes to tune the receive clock. The Tuning block generates the Delay Controls to the external Delay Controller module. The Tuning module receives the 64-byte Tuning Block (SD Mode) or 128-byte Tuning Block (eMMC Mode) and maintains a Tuning vector to determine the optimal Delay. The Tuning Block can be configured with number of Delay Taps (maximum 32) that are supported. Using the Tuning Block performs Tuning and selects the optimal Tap Point for the Receive Clock.

eMMC/SD Intf Control

The SD Interface Control block maps the internal signals to the External SD Interface and vice versa. Based on the Bus Width (1/4/8) the internal signals are driven out appropriately. In case of DS, the outputs are driven on the negative edge of the sd_clk. The inputs from RxFlops module are latched on the rx_clk (looped back or tuned clock) and output to the Receive Control Module for further processing.

16.3.2 Host Controller PHY Block Description

The blocks external to this Host Controller Core are the RX Clock Delay (DLL or DLY_BUF) module, the Rx Flops Module, Tx Clock Delay (DLL or DLY BUF) and TX Flops Module.

RxClk Delay Module

The RxClk Delay Module is used to support Receive Clock tuning to center align the receive data to the receive clock. There are two modes of Delaying the Receive Clock. First one is the Automatic Tuning of the Receive Clock when operating in SDR104 mode in SD 3.0 or HS200 and HS400 modes in eMMC 5.0, or optionally in SDR50 mode (SD3.0) when the Tuning is implemented. The second one is under manual controls to offset for Post Silicon Board Delays etc. The manual control is implemented for HS mode and SDR25/SDR50/DDR50 modes using the *phyctrl_itapdlysel* and *phyctrl_itapdlyena* signals. This module can be implemented with either using a DLL or Tap Delay Lines for generating various phases of clock and selecting one of the phases of the clock. The Maximum number of Tap Delay (phases of the clock) is 32. A typical implementation will use either 4 or 8 Tap Delay lines (phases of clock).
The preferred method is to use the Looped back sdcard_clk (rxclk_in) to generate multiple phases of the clock. In case of DLL based approach, this looped back clock is not ideal as the clock itself can dynamically be stopped by the Host Controller to Pause the data reception from the SD/eMMC Card. As the DLL takes longer times to lock the clock, we need a continuous clock.

RX Flops Module

The RX Flops Module is a where the CMD/DAT lines from the SD Interface are flopped on the output of the RxClk Delay Module. This module can be placed closer to the IO to achieve better timing on the Receive Signals. To Support both DDR Mode of operation, the signals are latched on both positive edge of the Receive clock and the negative edge of the receive clock. The outputs from these flops are passed onto to SD Host Controller Core for further processing.

TxClk Delay Module

The CMD and DAT outputs need to delayed w.r.t to the output SD_CLK signal to meet the hold time requirements in various modes of operation. The out going SD Clock is delayed and the delayed clock is used to flop the CMD/DAT lines and use this output to drive the SD Interface. The SD_CLK output itself is not delayed.

TX Flops Module

The SD Outputs from the SD Host Controller Core are flopped on the Delayed Tx Clock. Also to support the DDR Mode of operation, the separate set of outputs is received from the Core to be driven on positive and negative edge of the Delayed Clock.

Data Strobe Delay (DLL) Module

This module is used when operating in eMMC's HS400 Mode in which the data strobe from the card is delayed to properly latch the read data. The DLL should provide a 90 degree and 180 degree phase shifted outputs of the incoming Data Strobe Signal. The internal module uses these two signals to latch data in HS400 Mode.

HS400 Flops Module

The HS400 Flops Module is a where the CMD/DAT lines from the SD Interface are flopped on the 90 degree phase shifted Data Strobe. This module has to be placed closer to the IO to achieve better timing on the Receive Signals and to meet the timing requirements on the Phase shifted Strobe signal. To Support both DDR Mode of operation, the signals are latched on both positive edge of the Receive clock and the negative edge of the 90 degree phase shifted Data Strobe. The outputs from these flops are passed onto to SD Host Controller Core for further processing using the HS400 FIFO module.

HS400 FIFO Module

This module implements a 4 Deep FIFO to transfer the outputs from HS400 Flops Module into the receive Clock domain. The FIFO is written of the falling edge of the 180 degree phase shifted Data Strobe signal. The FIFO is read out on the Tuned Receive clock (every time the FIFO is not empty). The output form this module is provided to the SD Host Controller Core module for further processing.

DLL

Analog DLL is added as part of the eMMC50_PHY_TOP for tuning the clock using analog circuits.

16.4 Register Description

16.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
EMMCCORE_SADDR	0x0000	W	0x00000000	System address/ Argument 2 register
EMMCCORE_BLKSIZ	0x0004	HW	0x0000	Block size register
EMMCCORE_BLKCNT	0x0006	HW	0x0000	Block count register
EMMCCORE_ARG	0x0008	W	0x00000000	Argument register
EMMCCORE_TRANSMOD	0x000c	HW	0x0000	Transfer mode register
EMMCCORE_CMD	0x000e	HW	0x0000	Command register
EMMCCORE_RESP0	0x0010	W	0x00000000	Response register bit [31:0]
EMMCCORE_RESP1	0x0014	W	0x00000000	Response register bit [63:32]
EMMCCORE_RESP2	0x0018	W	0x00000000	Response register bit [95:64]
EMMCCORE_RESP3	0x001c	W	0x00000000	Response register bit [127:98]
EMMCCORE_BUFFER	0x0020	W	0x00000000	Buffer data port register
EMMCCORE_PRESTS	0x0024	W	0x1fff0000	Present state register
EMMCCORE_HOSTCTRL1	0x0028	B	0x00	Host control 1 register
EMMCCORE_PWRCTRL	0x0029	B	0x00	Power control register
EMMCCORE_BLKGAPCTRL	0x002a	B	0x00	Block gap control register
EMMCCORE_CLKCTRL	0x002c	HW	0x0000	Clock control Register
EMMCCORE_TIMEOUT	0x002e	B	0x00	Timeout control register
EMMCCORE_SWRST	0x002f	B	0x00	Software reset register
EMMCCORE_NORINTSTS	0x0030	HW	0x0000	Normal interrupt status register

Name	Offset	Size	Reset Value	Description
EMMCCORE_ERRINTSTS	0x0032	HW	0x0000	Error interrupt status register
EMMCCORE_NORINTSTSENA	0x0034	HW	0x0000	Normal interrupt status enable register
EMMCCORE_ERRINTSTSENA	0x0036	HW	0x0000	Error interrupt status enable register
EMMCCORE_NORINTSIGENA	0x0038	HW	0x0000	Normal interrupt signal enable register
EMMCCORE_ERRINTSIGENA	0x003a	HW	0x0000	Error interrupt signal enable register
EMMCCORE_ACMDERRSTS	0x003c	HW	0x0000	Auto CMD error status register
EMMCCORE_HOSTCTRL2	0x003e	HW	0x0000	Host Control 2 Register
EMMCCORE_CAP	0x0040	DW	0x8000200744ed0000	Capabilities register
EMMCCORE_FEACMD	0x0050	HW	0x0000	Force event register for Auto CMD error status
EMMCCORE_FEERRINT	0x0052	HW	0x0000	Force event register for error interrupt status
EMMCCORE_ADMAERRSTS	0x0054	HW	0x0000	ADMA error status register
EMMCCORE_ADMAADDR	0x0058	DW	0x0000000000000000	ADMA system address register
EMMCCORE_PVALINIT	0x0060	HW	0x0000	Preset value register for Initialization
EMMCCORE_PVALDS	0x0062	HW	0x0000	Preset value register for Default Speed
EMMCCORE_PVALHS	0x0064	HW	0x0000	Preset value register for High Speed
EMMCCORE_PVALSDR12	0x0066	HW	0x0000	Preset value register for SDR12
EMMCCORE_PVALSDR25	0x0068	HW	0x0000	Preset value register for SDR25
EMMCCORE_PVALSDR50	0x006a	HW	0x0000	Preset value register for SDR50
EMMCCORE_PVALSDR104	0x006c	HW	0x0000	Preset value register for SDR104
EMMCCORE_PVALDDR50	0x006e	HW	0x0000	Preset value register for DDR50
EMMCCORE_BOOTTIMEOUT	0x0070	W	0x00000000	Boot timeout control register
EMMCCORE_PVALHS400	0x0074	HW	0x0000	Preset value register for HS400
EMMCCORE_VENDOR	0x0078	HW	0x0000	Vendor register
EMMCCORE_SLOTINTSTS	0x00fc	HW	0x0000	Slot interrupt status register
EMMCCORE_VERSION	0x00fe	HW	0x1002	Host controller version register
EMMCCORE_CQVER	0x0200	W	0x00000510	Command queueing version register
EMMCCORE_CQCAP	0x0204	W	0x00000000	Command queueing capabilities register
EMMCCORE_CQCFG	0x0208	W	0x00000000	Command queueing configuration register
EMMCCORE_CQCTRL	0x020c	W	0x00000000	Command queueing control register
EMMCCORE_CQINTSTS	0x0210	W	0x00000000	Command queueing interrupt status register
EMMCCORE_CQINTSTSENA	0x0214	W	0x00000000	Command queueing interrupt status enable register
EMMCCORE_CQINTSIGENA	0x0218	W	0x00000000	Command queueing interrupt signal enable register

Name	Offset	Size	Reset Value	Description
EMMCCORE_CQINTCOAL	0x021c	W	0x00000000	Command queueing interrupt coalescing register
EMMCCORE_CQTDLBA	0x0220	W	0x00000000	Command queueing task descriptor list base address register
EMMCCORE_CQTDLBAU	0x0224	W	0x00000000	Command queueing task descriptor list base address upper 32bits register
EMMCCORE_CQTDB	0x0228	W	0x00000000	Command queueing task doorbell register
EMMCCORE_CQTDBN	0x022c	W	0x00000000	Command queueing task doorbell notification register
EMMCCORE_CQDQSTS	0x0230	W	0x00000000	Command queueing device queue status register
EMMCCORE_CQDPT	0x0234	W	0x00000000	Command queueing device pending tasks register
EMMCCORE_CQTCLR	0x0238	W	0x00000000	Command queueing task clear register
EMMCCORE_CQSSC1	0x0240	W	0x00011000	Command queueing send status configuration register 1
EMMCCORE_CQSSC2	0x0244	W	0x00000000	Command queueing send status configuration register 2
EMMCCORE_CQCRDT	0x0248	W	0x00000000	Command queueing command response for direct-command task register
EMMCCORE_CQRMEM	0x0250	W	0xfdf9a080	Command queueing response mode error mask register
EMMCCORE_CQTEI	0x0254	W	0x00000000	Command queueing task error information register
EMMCCORE_CQCRI	0x0258	W	0x00000000	Command queueing command response index register
EMMCCORE_CQCRA	0x025c	W	0x00000000	Command queueing command response argument register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

16.4.2 Detail Register Description

EMMCCORE_SADDR
Address: Operational Base + offset (0x0000)
System address/ Argument 2 register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	saddr This register contains the physical system memory address used for DMA transfers or the second argument for the Auto CMD23. (1) SDMA System Address This register contains the system memory address for a SDMA transfer. When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by the Resume command or by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the SDMA System Address register. ADMA does not use this register. (2) Argument 2 This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without AMDA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.

EMMCCORE_BLKSIZ

Address: Operational Base + offset (0x0004)

Block size register

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	HostSDMABufferSize To perform long DMA transfer, System Address register shall be updated at every system boundary during DMA transfer. These bits specify the size of contiguous buffer in the system memory. The DMA transfer shall wait at the every boundary specified by these fields and the HC generates the DMA Interrupt to request the HD to update the System Address register. These bits shall support when the DMA Support in the Capabilities register is set to 1 and this function is active when the DMA Enable in the Transfer Mode register is set to 1. 3'h0: 4KB(Detects A11 Carry out) 3'h1: 8KB(Detects A12 Carry out) 3'h2: 16KB(Detects A13 Carry out) 3'h3: 32KB(Detects A14 Carry out) 3'h4: 64KB(Detects A15 Carry out) 3'h5: 128KB(Detects A16 Carry out) 3'h6: 256KB(Detects A17 Carry out) 3'h7: 512KB(Detects A18 Carry out)
11:0	RW	0x000	TransferBlockSize This register specifies the block size for block data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. It can be accessed only if no transaction is executing (i.e after a transaction has stopped). Read operations during transfer return an invalid value and write operations shall be ignored. 12'h0000: No Data Transfer 12'h0001: 1 Byte 12'h0002: 2 Bytes 12'h0003: 3 Bytes 12'h0004: 4 Bytes ..... 12'h01FF: 511 Bytes 12'h0200: 512 Bytes ..... 12'h0800: 2048 Bytes

EMMCCORE_BLKCNT
Address: Operational Base + offset (0x0006)
Block count register

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	BlockCountForCurrentTransfer This register is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. The HC decrements the block count after each block transfer and stops when the count reaches zero. It can be accessed only if no transaction is executing (i.e. after a transaction has stopped). Read operations during transfer return an invalid value and write operations shall be ignored. When saving transfer context as a result of Suspend command, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the HD shall restore the previously save block count. 16'h0000: Stop Count 16'h0001: 1 block 16'h0002: 2 blocks ..... 16'hFFFF: 65535 blocks

EMMCCORE_ARG
Address: Operational Base + offset (0x0008)
Argument register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CommandArgument1 The SD Command Argument is specified as bit39-8 of Command-Format.

EMMCCORE_TRANSMOD
Address: Operational Base + offset (0x000c)
Transfer mode register

Bit	Attr	Reset Value	Description
15:6	RO	0x0	reserved
5	RW	0x0	MultiBlockSelect This bit enables multiple block data transfers. 0: Single Block 1: Multiple Block
4	RW	0x0	DataTransferDirectionSelect This bit defines the direction of data transfers. 0: Write (Host to Card) 1: Read (Card to Host)

Bit	Attr	Reset Value	Description
3:2	RW	0x0	AutoCmdEnable This field determines use of auto command functions 0: Auto Command Disabled 1: Auto CMD12 Enable 2: Auto CMD23 Enable 3: Reserved There are two methods to stop Multiple-block read and write operation. (1) Auto CMD12 Enable Multiple-block read and write commands for memory require CMD12 to stop the operation. When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transferis completed. Auto CMD12 error is indicated to the Auto CMD Error Status register. The Host Driver shall not set this bit if the command does not require CMD12. (2) Auto CMD23 Enable When this bit field is set to 10b, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command Register The following conditions are required to use the Auto CMD23. a. Auto CMD23 Supported (Host Controller Version is 3.00 or later) b. A memory card that supports CMD23 (SCR[33]=1) c. If DMA is used, it shall be ADMA d. Only when CMD18 or CMD25 is issued By writing the Command register, the Host Controller issues a CMD23 first and then issues a command specified by the Command Index in Command register 32-bit block count value for CMD23 is set to SDMA System Address / Argument 2 register
1	RW	0x0	BlockCountEnable This bit is used to enable the Block count register, which is only relevant for multiple block transfers. When this bit is 0, the Block Count register is disabled, which is useful in executing an infinite transfer. 0: Disable 1: Enable
0	RW	0x0	DMAEnable DMA can be enabled only if DMA Support bit in the Capabilities register is set. If this bit is set to 1, a DMA operation shall begin when the HD writes to the upper byte of Command register (00Fh). 0: Disable 1: Enable

EMMCCORE_CMD

Address: Operational Base + offset (0x000e)
Command register

Bit	Attr	Reset Value	Description
15:8	RO	0x0	reserved
7:6	RW	0x0	CmdType 0: Normal 1: Suspend 2: Resume 3: Abort There are three types of special commands. Suspend, Resume and Abort. These bits shall bet set to 00b for all other commands. Suspend Command If the Suspend command succeeds, the HC shall assume the SD Bus has been released and that it is possible to issue the next command which uses the DAT line. The HC shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The Interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the HC shall maintain its current state. and the HD shall restart the transfer by setting Continue Request in the Block Gap Control Register. Resume Command The HD re-starts the data transfer by restoring the registers in the range of 000-00Dh. The HC shall check for busy before starting write transfers. Abort Command If this command is set when executing a read transfer, the HC shall stop reads to the buffer. If this command is set when executing a write transfer, the HC shall stop driving the DAT line. After issuing the Abort command, the HD should issue a software reset.
5	RW	0x0	DataPresentSel 0: No Data Present 1: Data Present This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. If is set to 0 for the following: a. Commands using only CMD line (ex. CMD52) b. Commands with no data transfer but using busy signal on DAT[0] line (R1b or R5b ex. CMD38) c Resume Command
4	RW	0x0	CmdIndexChkEna If this bit is set to 1, the HC shall check the index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 0: Disable 1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x0	CmdCrcChkEna If this bit is set to 1, the HC shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. 0: Disable 1: Enable
2	RO	0x0	reserved
1:0	RW	0x0	RespTypeSel Response Type Select 0: No Response 1: Response length 136 2: Response length 48 3: Response length 48 check Busy after response

EMMCCORE_RESP0
Address: Operational Base + offset (0x0010)
Response register bit [31:0]

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Resp Response register bit [31:0]

EMMCCORE_RESP1
Address: Operational Base + offset (0x0014)
Response register bit [63:32]

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Resp Response register bit [63:32]

EMMCCORE_RESP2
Address: Operational Base + offset (0x0018)
Response register bit [95:64]

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Resp Response register bit [95:64]

EMMCCORE_RESP3
Address: Operational Base + offset (0x001c)
Response register bit [127:98]

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Resp Response register bit [127:98]

EMMCCORE_BUFFER
Address: Operational Base + offset (0x0020)
Buffer data port register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	BufferData The Host Controller Buffer can be accessed through this 32-bit Data Port Register.

EMMCCORE_PRESTS

Address: Operational Base + offset (0x0024)

Present state register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:25	RW	0xf	DAT74LineSignalLevel This status is used to check DAT line level to recover from errors, and for debugging. [28]: DAT[7] [27]: DAT[6] [26]: DAT[5] [25]: DAT[4]
24	RO	0x1	CMDLineSignalLevel This status is used to check CMD line level to recover from errors, and for debugging.
23:20	RO	0xf	DAT30LineSignalLevel DAT[3:0] Line Signal Level This status is used to check DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0]. [23]: DAT[3] [22]: DAT[2] [21]: DAT[1] [20]: DAT[0]
19	RO	0x1	WrPrtSwPinLvl Write Protect Switch Pin Level. The Write Protect Switch is supported for memory and combo cards. This bit reflects the SDWP# pin. 0: Write protected (SDWP# = 0) 1: Write enabled (SDWP# = 1)
18	RO	0x1	CardDetectPinLevel This bit reflects the inverse value of the SDCD# pin. 0: No Card present (SDCD# = 1) 1: Card present (SDCD# = 0)
17	RO	0x1	CardStateStable 0: Reset of Debouncing 1: No Card or Inserted This bit is used for testing. If it is 0, the Card Detect Pin Level is not stable. If this bit is set to 1, it means the Card Detect Pin Level is stable. The Software Reset For All in the Software Reset Register shall not affect this bit.

Bit	Attr	Reset Value	Description
16	RO	0x1	CardInserted 0: Reset or Debouncing or NoCard 1: Card Inserted This bit indicates whether a card has been inserted. Changing from 0 to 1 generates a Card Insertion interrupt in the Normal Interrupt Status register and changing from 1 to 0 generates a Card Removal Interrupt in the Normal Interrupt Status register. The Software Reset For All in the Software Reset register shall not affect this bit. If a Card is removed while its power is on and its clock is oscillating, the HC shall clear SD Bus Power in the Power Control register and SD Clock Enable in the Clock control register. In addition the HD should clear the HC by the Software Reset For All in Software register. The card detect is active regardless of the SD Bus Power.
15:12	RO	0x0	reserved
11	RO	0x0	BufferReadEnable 0: Read Disable 1: Read Enable This status is used for non-DMA read transfers. This read only flag indicates that valid data exists in the host side buffer status. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when all the block data is ready in the buffer and generates the Buffer Read Ready Interrupt.
10	RO	0x0	BufferWriteEnable 0: Write Disable 1: Write Enable This status is used for non-DMA write transfers. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when top of block data can be written to the buffer and generates the Buffer Write Ready Interrupt.

Bit	Attr	Reset Value	Description
9	RO	0x0	ReadTransActive 1: Transferring data 0: No valid data This status is used for detecting completion of a read transfer. This bit is set to 1 for either of the following conditions: a. After the end bit of the read command b. When writing a 1 to continue Request in the Block Gap Control register to restart a read transfer This bit is cleared to 0 for either of the following conditions: a. When the last data block as specified by block length is transferred to the system. b. When all valid data blocks have been transferred to the system and no current block transfers are being sent as a result of the Stop At Block Gap Request set to 1. A transfer complete interrupt is generated when this bit changes to 0.
8	RO	0x0	WriteTransActive 1: transferring data 0: No valid data This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the HC. This bit is set in either of the following cases: a. After the end bit of the write command. b. When writing a 1 to Continue Request in the Block Gap Control register to restart a write transfer. This bit is cleared in either of the following cases: a. After getting the CRC status of the last data block as specified by the transfer count (Single or Multiple). b. After getting a CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request. During a write transaction, a Block Gap Event interrupt is generated when this bit is changed to 0, as a result of the Stop At Block Gap Request being set. This status is useful for the HD in determining when to issue commands during write busy.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RO	0x0	ReTuningReq Re-Tuning Request 1: Sampling clock needs re-tuning 0: Fixed or well tuned sampling clock Host Controller may request Host Driver to execute re-tuning sequence by setting this bit when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This bit is cleared when a command is issued with setting Execute Tuning in the Host Control 2 register. Changing of this bit from 0 to 1 generates Re-Tuning Event. Refer to Normal Interrupt registers for more detail. This bit isn't set to 1 if Sampling Clock Select in the Host Control 2 register is set to 0 (using fixed sampling clock).
2	RO	0x0	DATLineActive This bit indicates whether one of the DAT line on SD bus is in use. 1: DAT line active 0: DAT line inactive
1	RO	0x0	DatInhibit 1: cannot issue command which uses the DAT line 0: Can issue command which uses the DAT line This status bit is generated if either the DAT Line Active or the Read transfer Active is set to 1. If this bit is 0, it indicates the HC can issue the next SD command. Commands with busy signal belong to Command Inhibit (DAT) (ex. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Normal interrupt status register. Note: The SD Host Driver can save registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0.

Bit	Attr	Reset Value	Description
0	RO	0x0	CmdInhibit If this bit is 0, it indicates the CMD line is not in use and the HC can issue a SD command using the CMD line. This bit is set immediately after the Command register is written. This bit is cleared when the command response is received. Even if the Command Inhibit (DAT) is set to 1, Commands using only the CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a Command complete interrupt in the Normal Interrupt Status register. If the HC cannot issue the command because of a command conflict error or because of Command Not Issued By Auto CMD12 Error, this bit shall remain 1 and the Command Complete is not set. Status issuing Auto CMD12 is not read from this bit. Auto CMD12 and Auto CMD23 consist of two responses. In this case, this bit is not cleared by the response of CMD12 or CMD23 but cleared by the response of a read/write command. Status issuing Auto CMD12 is not read from this bit. So if a command is issued during Auto CMD12 operation, Host Controller shall manage to issue two commands: CMD12 and a command set by Command register.

EMMCCORE_HOSTCTRL1
Address: Operational Base + offset (0x0028)
Host control 1 register

Bit	Attr	Reset Value	Description
7	RW	0x0	CardDetSginalDet This bit selects source for card detection. 1: The card detect test level is selected 0: SDCD# is selected (for normal use)
6	RW	0x0	CardDetTestLevel This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not. Generates (card ins or card removal) interrupt when the normal int sts enable bit is set. 1: Card Inserted 0: No Card

Bit	Attr	Reset Value	Description
5	RW	0x0	ExtendedDataTransWidth This bit controls 8-bit bus width mode for embedded device. Support of this function is indicated in 8-bit Support for Embedded Device in the Capabilities register. If a device supports 8-bit bus mode, this bit may be set to 1. If this bit is 0, bus width is controlled by Data Transfer Width in the Host Control 1 register.This bit is not effective when multiple devices are installed on a bus slot (Slot Type is set to 10b in the Capabilities register). In this case, each device bus width is controlled by Bus Width Preset field in the Shared Bus register. 1: 8-bit Bus Width 0: Bus Width is Selected by Data Transfer Width
4:3	RW	0x0	DMASelect One of supported DMA modes can be selected. The host driver shall check support of DMA modes by referring the Capabilities register. 0: SDMA is selected 1: 32-bit Address ADMA1 is selected 2: 32-bit Address ADMA2 is selected 3: 64-bit Address ADMA2 is selected
2	RW	0x0	HighSpeedEna 1: High Speed Mode 0: Normal Speed Mode This bit is optional. Before setting this bit, the HD shall check the High Speed Support in the capabilities register. If this bit is set to 0 (default), the HC outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz/20MHz for eMMC). If thisbit is set to 1, the HC outputs CMD line and DAT lines at the rising edge of the SD clock (up to 50 MHz for SD/52MHz for eMMC)/ 208Mhz (for SD3.0). If Preset Value Enable in the Host Control 2 register is set to 1, Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitches. After setting this field, the Host Driver sets SD Clock Enable again
1	RW	0x0	DataTransWidth 1: 4 bit mode 0: 1 bit mode This bit selects the data width of the HC. The HD shall select it to match the data width of the SD card.
0	RO	0x0	reserved

EMMCCORE_PWRCTRL
Address: Operational Base + offset (0x0029)
Power control register

Bit	Attr	Reset Value	Description
7:1	RO	0x0	reserved
0	RW	0x0	SDBusPower 1: Power on 0: Power off

EMMCCORE_BLKGAPCTRL
Address: Operational Base + offset (0x002a)
Block gap control register

Bit	Attr	Reset Value	Description
7	RW	0x0	BootAckChk To check for the boot acknowledge in boot operation. 1: wait for boot ack from eMMC card 0: Will not wait for boot ack from eMMC card
6	RW	0x0	AltBootEn To start boot code access in alternative mode. 1: To start alternate boot mode access 0: To stop alternate boot mode access
5	RW	0x0	BootEn To start boot code access. 1: To start boot code access 0: To stop boot code access
4	RW	0x0	SpiMode SPI mode enable bit. 1: SPI mode 0: SD mode
3	RW	0x0	IntAtBlkGap Interrupt At Block Gap. This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the HD detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card.

Bit	Attr	Reset Value	Description
2	RW	0x0	ReadWaitControl 1: Enable Read Wait Control 0: Disable Read Wait Control The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using DAT[2] line. Otherwise the HC has to stop the SD clock to hold read data, which restricts commands generation. When the HD detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise DAT line conflict may occur. If this bit is set to 0, Suspend / Resume cannot be supported
1	R/W SC	0x0	ContinueRequest 1: Restart 0: Ignored This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request. To cancel stop at the block gap, set Stop At block Gap Request to 0 and set this bit to restart the transfer. The HC automatically clears this bit in either of the following cases: a. In the case of a read transaction, the DAT Line Active changes from 0 to 1 as a read transaction restarts. b. In the case of a write transaction, the Write transfer active changes from 0 to 1 as the write transaction restarts. Therefore it is not necessary for Host driver to set this bit to 0. If Stop At Block Gap Request is set to 1, any write to this bit is ignored.

Bit	Attr	Reset Value	Description
0	RW	0x0	StopAtBlkGapReq 1: Stop 0: Transfer This bit is used to stop executing a transaction at the next block gap for non-DMA,SDMA and ADMA transfers. Until the transfer complete is set to 1, indicating a transfer completion the HD shall leave this bit set to 1. Clearing both the Stop At Block Gap Request and Continue Request shall not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The HC shall honour Stop At Block Gap Request for write transfers, but for read transfers it requires that the SD card support Read Wait. Therefore the HD shall not set this bit during read transfers unless the SD card supports Read Wait and has set Read Wait Control to 1. In case ofwrite transfers in which the HD writes data to the Buffer Data Port register, the HD shall set this bit after all block data is written. If this bit is set to 1, the HD shall not write data to Buffer data port register. This bit affects Read Transfer Active, Write Transfer Active, DAT line active and Command Inhibit (DAT) in the Present State register.

EMMCCORE_CLKCTRL
Address: Operational Base + offset (0x002c)
Clock control Register

Bit	Attr	Reset Value	Description
15:8	RW	0x00	SDCLKFreqSel SDCLK Frequency Select. This register is used to select the frequency of the SDCLK pin. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD clock in the capabilities register. Only the following settings are allowed. (1) 8-bit Divided Clock Mode 8'h80: base clock divided by 256 8'h40: base clock divided by 128 8'h20: base clock divided by 64 8'h10: base clock divided by 32 8'h08: base clock divided by 16 8'h04: base clock divided by 8 8'h02: base clock divided by 4 8'h01: base clock divided by 2 8'h00: base clock(10MHz-63MHz) Setting 8'h00 specifies the highest frequency of the SD Clock. When setting multiple bits, the most significant bit is used as the divisor. But multiple bits should not be set. The two default divider values can be calculated by the frequency that is defined by the Base Clock Frequency For SD Clock in the Capabilities register. a. 25 MHz divider value b. 400 KHz divider value The frequency of the SDCLK is set by the following formula: Clock Frequency = (Baseclock) / divisor. Thus choose the smallest possible divisor which results in a clock frequency that is less than or equal to the target frequency. Maximum Frequency for SD = 50Mhz (base clock) Maximum Frequency for eMMC = 52Mhz (base clock) Minimum Frequency = 195.3125Khz (50Mhz / 256), same calculation for eMMC also. (2) 10-bit Divided Clock Mode Host Controller supports this mandatory mode instead of the 8-bit Divided Clock Mode. The length of divider is extended to10 bits and all divider values shall be supported. 10'h3FF:1/2046 Divided Clock N: 1/2N Divided Clock (Duty 50%) 10'h002: 1/4 Divided Clock 10'h001: 1/2 Divided Clock 10'h000: Base Clock (10MHz-254MHz)
7:6	RW	0x0	SDCLKFreqSelUpper Field0001 Abstract Upper Bits of SDCLK Frequency Select. Bit 07-06 is assigned to bit 09-08 of clock divider in SDCLK Frequency Select

Bit	Attr	Reset Value	Description
5	RW	0x0	ClkGenSel Clock Generator Select. 1: Programmable Clock Mode 0: Divided Clock Mode This bit is used to select the clock generator mode in SDCLK Frequency Select. If the Programmable Clock Mode is supported (non-zero value is set to Clock Multiplier in the Capabilities register), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read. This bit depends on the setting of Preset Value Enable in the Host Control 2 register. If the Preset Value Enable= 0, this bit is set by Host Driver. If the Preset Value Enable= 1, this bit is automatically set to a value specified in one of Preset Value registers.
4:3	RO	0x0	reserved
2	RW	0x0	SDClkEna SD Clock Enable. 1: Enable 0: Disable The HC shall stop SDCLK when writing this bit to 0. SDCLK frequency Select can be changed when this bit is 0. Then, the HC shall maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK = 0). If the HC detects the No Card state, this bit shall be cleared.
1	RO	0x0	InternalClockStable Internal Clock Stable. 1: Ready 0: Not Ready This bit is set to 1 when SD clock is stable after writing to Internal Clock Enable in this register to 1. The SD Host Driver shall wait to set SD Clock Enable until this bit is set to 1. Note: This is useful when using PLL for a clock oscillator that requires setup time.
0	RW	0x0	InternalClockEnable 1: Oscillate 0: Stop This bit is set to 0 when the HD is not using the HC or the HC awaits a wakeup event. The HC should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the HC shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.

EMMCCORE_TIMEOUT

Address: Operational Base + offset (0x002e)
Timeout control register

Bit	Attr	Reset Value	Description
7:4	RO	0x0	reserved
3:0	RW	0x0	DataTimeoutCounterValue This value determines the interval by which DAT line time-outs are detected. Refer to the Data Time-out Error in the Error Interrupt Statusregister for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the sdclockTMCLK by this value. When setting this register, prevent inadvertent time-out events by clearing the Data Time-out Error Status Enable (in the Error Interrupt Status Enable register). 4'hf: Reserved 4'he: TMCLK * 2^27 ..... 4'h1: TMCLK * 2^14 4'h0: TMCLK * 2^13

EMMCCORE_SWRST
Address: Operational Base + offset (0x002f)
Software reset register

Bit	Attr	Reset Value	Description
7:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	R/WSC	0x0	SoftwareResetDAT Software Reset for DAT Line. 1: Reset 0: Work Only part of data circuit is reset. The following registers and bits are cleared by this bit: a. Buffer Data Port Register: Buffer is cleared and Initialized b. Present State register: Buffer read Enable Buffer write Enable Read Transfer Active Write Transfer Active DAT Line Active Command Inhibit (DAT) c. Block Gap Control register: Continue Request Stop At Block Gap Request d. Normal Interrupt Status register: Buffer Read Ready Buffer Write Ready Block Gap Event Transfer Complete
1	R/WSC	0x0	SoftwareResetCMD Only part of command circuit is reset. 1: Reset 0: Work The following registers and bits are cleared by this bit: a. Present State register: Command Inhibit (CMD) b. Normal Interrupt Status register: Command Complete
0	R/WSC	0x0	SoftwareResetALL Software Reset for All 1: Reset 0: Work This reset affects the entire HC except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the HD shall set this bit to 1 to reset the HC. The HC shall reset this bit to 0 when capabilities registers are valid and the HD can read them. Additional use of Software Reset For All may not affect the value of the Capabilities registers. If this bit is set to 1, the SD card shall reset itself and must be reinitialized by the HD.

EMMCCORE_NORINTSTS
Address: Operational Base + offset (0x0030)
Normal interrupt status register

Bit	Attr	Reset Value	Description
15	RO	0x0	ErrorInterrupt 0: No Error 1: Error If any of the bits inthe Error Interrupt Status Register are set, then this bit is set. Therefore the HD can test for an error by checking this bit first.
14	W1C	0x0	BootTerminateInterrupt This status is set if the boot operation get terminated 0: Boot operation is not terminated 1: Boot operation is terminated
13	W1C	0x0	BootAckRcv This status is set if the boot acknowledge is received from device. 0: Boot ack is not received 1: Boot ack is received
12	RO	0x0	ReTuningEvent 1: Re-Tuning should be performed 0: Re-Tuning is not required This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. Host Controller requestsHost Driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning.
11:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RO	0x0	CardInterrupt 0: No Card Interrupt 1: Generate Card Interrupt Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the HC shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the card and the interrupt to the Host system. when this status has been set and the HD needs to start this interrupt service, Card Interrupt Status Enable in the Normal Interrupt Status register shall be set to 0 in order to clear the card interrupt statuses latched in the HC and stop driving the Host System. After completion of the card interrupt service (the reset factor in the SD card and the interrupt signal may not be asserted), set Card Interrupt Status Enable to 1 and start sampling the interrupt signal again. Interrupt detected by DAT[1] is supported when there is a card per slot.
7	W1C	0x0	CardRemoval 0: Card State Stable or Debouncing 1: Card Removed This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the HD writes this bit to 1 to clear this status the status of the Card Inserted in the Present State registershould be confirmed. Because the card detect may possibly be changed when the HD clear this bit an Interrupt event may not be generated.
6	W1C	0x0	CardInsertion 0: Card State Stable or Debouncing 1: Card Inserted This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the HD writes this bit to 1 to clear this status the status of the Card Inserted in the Present State registershould be confirmed. Because the card detect may possibly be changed when the HD clear this bit an Interrupt event may not be generated.
5	W1C	0x0	BufferReadReady 0: Not Ready to read Buffer 1: Ready to read Buffer This status is set if the Buffer Read Enable changes from 0 to 1. Buffer Read Ready is set to 1 for every CMD19 execution in tuning procedure.

Bit	Attr	Reset Value	Description
4	W1C	0x0	BufferWriteReady 0: Not Ready to Write Buffer 1: Ready to Write Buffer This status is set if the Buffer Write Enable changes from 0 to 1.
3	W1C	0x0	DMAInterrupt 0: No DMA Interrupt 1: DMA Interrupt is Generated This status is set if the HC detects the Host DMA Buffer Boundary in the Block Size regiser.
2	W1C	0x0	BlockGapEvent 0: No Block Gap Event 1: Transaction stopped at Block Gap If the Stop At Block Gap Request in the Block Gap Control Register is set, this bit is set. a. Read Transaction: This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing. The Read Wait must be supported inorder to use this function). b. Write Transaction: This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing).

Bit	Attr	Reset Value	Description
1	W1C	0x0	TransferComplete 0: No Data Transfer Complete 1: Data Transfer Complete This bit is set when a read / write transaction is completed. a. Read Transaction: This bit is set at the falling edge of Read Transfer Active Status. There are two cases in which the Interrupt is generated. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request in the Block Gap Control Register (After valid data has been read to the Host System). b. Write Transaction: This bit is set at the falling edge of the DAT Line Active Status. There are two cases in which the Interrupt is generated. The first is when the last data is written to the card as specified by data length and Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request in the Block Gap Control Register and data transfers completed. (Aftervalid data is written to the SD card and the busy signal is released). c. In case of command with busy This bit is set when busy is deasserted. Note: a. Transfer Complete has higher priority than Data Time-out Error. If both bits are set to 1, the data transfer can be considered complete b. While performing tuning procedure (Execute Tuning is set to 1), Transfer Complete is not set to 1
0	W1C	0x0	CommandComplete 0: No Command Complete 1: Command Complete This bit is set when we get the end bit of the command response (Except Auto CMD12 and Auto CMD23) Note: Command Time-out Error has higher priority than Command Complete. If both are set to 1, it can be considered that the response was not received correctly.

EMMCCORE_ERRINTSTS

Address: Operational Base + offset (0x0032)

Error interrupt status register

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	W1C	0x0	TargetRespErr 0: no error 1: error Occurs when detecting ERROR in m_hresp(dma transaction)
11:10	RO	0x0	reserved
9	W1C	0x0	ADMAErr 1: Error 0: No error This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register.
8	W1C	0x0	AutoCMDErr 0: No Error 1: Error Auto CMD12 and Auto CMD23 use this error status. This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register has changed from 0 to 1. In case of Auto CMD12, this bit is set to 1, not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error.
7	W1C	0x0	CurrentLimitErr 0: No Error 1: Power Fail By setting the SD Bus Power bit in the Power Control Register, the HC is requested to supply power for the SD Bus. If the HC supports the Current Limit Function, it can be protected from an Illegal card by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the HC is not supplying power to SD card due to some failure. Reading 0 means that the HC is supplying power and no error has occurred. This bit shall always set to be 0, if the HC does not support this function. Note: The current_Limit_Error is to be implemented if customer application requires it.. By default it is not implementedas there is no specific requirement from Customers.
6	W1C	0x0	DataEndBitErr 0: No Error 1: Error Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status.

Bit	Attr	Reset Value	Description
5	W1C	0x0	DataCRCErr 0: No Error 1: Error Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than "010".
4	W1C	0x0	DataTimeoutErr 0: No Error 1: Timeout Occurs when detecting one of following timeout conditions. a. Busy Timeout for R1b, R5b type. b. Busy Timeout after Write CRC status c. Write CRC status Timeout d. Read Data Timeout
3	W1C	0x0	CmdIndexErr 0: No Error 1: Error Occurs if a Command Index error occurs in the Command Response.
2	RW	0x0	CmdEndBitErr 0: No Error 1: End Bit Error Generated Occurs when detecting that the end bit of a command response is 0.
1	W1C	0x0	CmdCRCErr 0: No Error 1: CRC Error Generated Command CRC Error is generated in two cases. a. If a response is returned and the Command Time-out Error is set to 0, this bit is set to 1 when detecting a CRT error in the command response b. The HC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the HC drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the HC shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict.
0	W1C	0x0	CmdTimeoutErr 0: No Error 1: Timeout Occurs only if the no response is returned within 64 SDCLK cycles from the end bit of the command. If the HC detects a CMD line conflict, in which case Command CRC Error shall also be set. This bit shall be set without waiting for 64 SDCLK cycles because the command will be aborted by the HC.

EMMCCORE_NORINTSTSENA
Address: Operational Base + offset (0x0034)
Normal interrupt status enable register

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14	RW	0x0	BootTerminateInterrupt 0: Masked 1: Enabled
13	RW	0x0	BootAckRcv 0: Masked 1: Enabled
12	RW	0x0	ReTuningEvent 0: Masked 1: Enabled
11:9	RO	0x0	reserved
8	RW	0x0	CardInterrupt 0: Masked 1: Enabled
7	RW	0x0	CardRemoval 0: Masked 1: Enabled
6	RW	0x0	CardInsertion 0: Masked 1: Enabled
5	RW	0x0	BufferReadReady 0: Masked 1: Enabled
4	RW	0x0	BufferWriteReady 0: Masked 1: Enabled
3	RW	0x0	DMAInterrupt 0: Masked 1: Enabled
2	RW	0x0	BlockGapEvent 0: Masked 1: Enabled
1	RW	0x0	TransferComplete 0: Masked 1: Enabled
0	RW	0x0	CommandComplete 0: Masked 1: Enabled

EMMCCORE_ERRINTSTSENA
Address: Operational Base + offset (0x0036)
Error interrupt status enable register

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12	RW	0x0	TargetRespErr 0: Masked 1: Enabled
11:10	RO	0x0	reserved
9	RW	0x0	ADMAErr 0: Masked 1: Enabled
8	RW	0x0	AutoCMDErr 0: Masked 1: Enabled
7	RW	0x0	CurrentLimitErr 0: Masked 1: Enabled
6	RW	0x0	DataEndBitErr 0: Masked 1: Enabled
5	RW	0x0	DataCRCErr 0: Masked 1: Enabled
4	RW	0x0	DataTimeoutErr 0: Masked 1: Enabled
3	RW	0x0	CmdIndexErr 0: Masked 1: Enabled
2	RW	0x0	CmdEndBitErr 0: Masked 1: Enabled
1	RW	0x0	CmdCRCErr 0: Masked 1: Enabled
0	RW	0x0	CmdTimeoutErr 0: Masked 1: Enabled

EMMCCORE_NORINTSIGENA

Address: Operational Base + offset (0x0038)

Normal interrupt signal enable register

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14	RW	0x0	BootTerminateInterrupt 0: Masked 1: Enabled

Bit	Attr	Reset Value	Description
13	RW	0x0	BootAckRcv 0: Masked 1: Enabled
12	RW	0x0	ReTuningEvent 0: Masked 1: Enabled
11:9	RO	0x0	reserved
8	RW	0x0	CardInterrupt 0: Masked 1: Enabled
7	RW	0x0	CardRemoval 0: Masked 1: Enabled
6	RW	0x0	CardInsertion 0: Masked 1: Enabled
5	RW	0x0	BufferReadReady 0: Masked 1: Enabled
4	RW	0x0	BufferWriteReady 0: Masked 1: Enabled
3	RW	0x0	DMAInterrupt 0: Masked 1: Enabled
2	RW	0x0	BlockGapEvent 0: Masked 1: Enabled
1	RW	0x0	TransferComplete 0: Masked 1: Enabled
0	RW	0x0	CommandComplete 0: Masked 1: Enabled

EMMCCORE_ERRINTSIGENA

Address: Operational Base + offset (0x003a)

Error interrupt signal enable register

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12	RW	0x0	TargetRespErr 0: Masked 1: Enabled
11:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	ADMAErr 0: Masked 1: Enabled
8	RW	0x0	AutoCMDErr 0: Masked 1: Enabled
7	RW	0x0	CurrentLimitErr 0: Masked 1: Enabled
6	RW	0x0	DataEndBitErr 0: Masked 1: Enabled
5	RW	0x0	DataCRCErr 0: Masked 1: Enabled
4	RW	0x0	DataTimeoutErr 0: Masked 1: Enabled
3	RW	0x0	CmdIndexErr 0: Masked 1: Enabled
2	RW	0x0	CmdEndBitErr 0: Masked 1: Enabled
1	RW	0x0	CmdCRCErr 0: Masked 1: Enabled
0	RW	0x0	CmdTimeoutErr 0: Masked 1: Enabled

EMMCCORE_ACMDERRSTS
Address: Operational Base + offset (0x003c)
Auto CMD error status register

Bit	Attr	Reset Value	Description
15:8	RO	0x0	reserved
7	RO	0x0	CmdNotIssByACMD12Err Command Not Issued By Auto CMD12 Error. Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04 - D01) in this register. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. 0: No Error 1: Not Issued
6:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RO	0x0	ACMDIndexErr Auto CMD Index Error. Occurs if the Command Index error occurs in response to a command. 0: No Error 1: Error
3	RO	0x0	ACMDEndBitErr Auto CMD End Bit Error. Occurs when detecting that the end bit of command response is 0. 0: No Error 1: End Bit Error Generated
2	RO	0x0	ACMDCRCErr Auto CMD CRC Error Occurs when detecting a CRC error in the command response. 0: No Error 1: CRC Error Generated
1	RO	0x0	ACMDTimeoutErr Auto CMD Timeout Error Occurs if the no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (D04 - D02) are meaningless. 0: No Error 1: Timeout
0	RO	0x0	ACMD12notExe Auto CMD12 not Executed If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the HC cannot issue Auto CMD12 to stop memory multiple block transfer due to some error. If this bit isset to 1, other error status bits (D04 - D01) are meaningless. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. 0: Executed 1: Not Executed

EMMCCORE_HOSTCTRL2
Address: Operational Base + offset (0x003e)
Host Control 2 Register

Bit	Attr	Reset Value	Description
15	RW	0x0	PresetValueEnable 1: Automatic Selection by Preset Value are Enabled 0: SDCLK and Driver Strength are controlled by Host Driver As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set to automatic. This bit enablesthe functions defined in the Preset Value registers. If this bit is set to 0, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Driver. If this bit is set to 1, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Controller as specified in the Preset Value registers.
14	RW	0x0	AsynIntEn Asynchronous Interrupt Enable This bit can be set to 1 if a card support asynchronous interrupt and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode(and zero is set to Interrupt Pin Select in the Shared Bus Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver CardInterrupt to the host when it is asserted by the card. 1: Enabled 0: Disabled
13:8	RO	0x0	reserved
7	RW	0x0	SamplingClockSelect Sampling Clock Select This bit is set by tuning procedure when Execute Tuning is cleared. Writing 1 to this bit is meaningless and ignored. Setting 1 means that tuning is completed successfully and setting 0 means that tuning is failed. Host Controller uses this bit to select sampling clock to receive CMD and DAT. This bit is cleared by writing 0. Change of this bit is not allowed while the Host Controller is receiving response or a read data block. 1: Tuned clock is used to sample data 0: Fixed clock is used to sample data

Bit	Attr	Reset Value	Description
6	R/WSC	0x0	ExecuteTuning This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select. Tuning procedure is aborted by writing 0 for more detail about tuning procedure. 1: Execute Tuning 0: Not Tuned or Tuning Completed
5:3	RO	0x0	reserved
2:0	RW	0x0	UHSModeSelect UHS Mode Select. This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1. If Preset Value Enable in the Host Control 2 register is set to 1, Host Controller sets SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select according to Preset Value registers. In this case, one of preset value registers is selected by this field. Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitch. After setting this field, Host Driver sets SD Clock Enable again. 3'h0: SDR12 3'h1: SDR25 3'h2: SDR50 3'h3: SDR104 3'h4: DDR50 3'h5: HS400 others: Reserved When SDR50, SDR104 or DDR50 is selected for SDIO card, interrupt detection at the block gap shall not be used. Read Wait timing is changed for these modes. Refer to the SDIO Specification Version 3.00 for more detail.

EMMCCORE_CAP

Address: Operational Base + offset (0x0040)

Capabilities register

Bit	Attr	Reset Value	Description
63	RO	0x1	HS400Support HS400 Support 0: Not Supported 1: Supported
62:58	RO	0x0	reserved
57	RO	0x0	SPIBlockMode SPI block mode 0: Not Supported 1: Supported

Bit	Attr	Reset Value	Description
56	RO	0x0	reserved
55:48	RO	0x10	ClockMultiplier This field indicates clock multiplier value of programmable clock generator. Refer to Clock Control register. Setting 00h means that Host Controller does not support programmable clock generator. 8'hFF: Clock Multiplier M = 256 .... 8'h02: Clock Multiplier M = 3 8'h01: Clock Multiplier M = 2 8'h00: Clock Multiplier is Not Supported
47:46	RO	0x0	RetuningMode Re-tuning modes This field defines the re-tuning capability of a Host Controller and how to manage the data transfer length and a Re-Tuning Timer by the Host Driver 0: Mode1 1: Mode2 2: Mode3 3: Reserved There are two re-tuning timings: Re-Tuning Request and expiration of a Re-Tuning Timer. By receiving either timing, the Host Driver executes the re-tuning procedure just before a next command issue
45	RO	0x1	UseTuningForSDR50 Use Tuning for SDR50 If this bit is set to 1, this Host Controller requires tuning to operate SDR50. (Tuning is always required to operate SDR104.) 1: SDR50 requires tuning 0: SDR50 does not require tuning
44	RO	0x0	reserved
43:40	RO	0x0	TimerCountForRetuning Timer count for ReTuning This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3. 4'h0 - Get information via other source 4'h1 = 1 seconds 4'h2 = 2 seconds 4'h3 = 4 seconds 4'h4 = 8 seconds ..... 4'hB = 1024 seconds 4'hF - Ch = Reserved

Bit	Attr	Reset Value	Description
39	RO	0x1	DriverType4Support Driver Type 4 Support 1: Driver Type 4 is Supported 0: Driver Type 4 is Not Supported
38	RO	0x1	DriverTypeDSupport This bit indicates support of Driver Type D for 1.8 Signaling. 1: Driver Type D is Supported 0: Driver Type D is Not Supported
37	RW	0x1	DriverTypeCSupport This bit indicates support of Driver Type C for 1.8 Signaling. 1: Driver Type C is Supported 0: Driver Type C is Not Supported
36	RO	0x1	DriverTypeASupport This bit indicates support of Driver Type A for 1.8 Signaling. 1: Driver Type A is Supported 0: Driver Type A is Not Supported
35	RO	0x0	reserved
34	RO	0x1	DDR50Support DDR50 Support 1: DDR50 is Supported 0: DDR50 is Not Supported
33	RO	0x1	SDR104Support SDR104 Support. 1: SDR104 is Supported 0: SDR104 is Not Supported
32	RO	0x1	SDR50Support SDR50 Support 1: SDR50 is Supported 0: SDR50 is Not Supported
31:30	RO	0x1	SlotType This field indicates usage of a slot by a specific Host System. (A host controller register set is defined per slot.) Embedded slot for one device (01b) means that only one non-removable device is connected to a SD bus slot. Shared Bus Slot (10b) can be set if Host Controller supports Shared Bus Control register. The Standard Host Driver controls only a removable card or one embedded device is onnected to a SD bus slot. If a slot is configured for shared bus (10b), the Standard Host Driver does not control embedded devices connected to a shared bus. Shared bus slot is controlled by a specific host driver developed by a Host System. 0: Removable Card Slot 1: Embedded Slot for One Device 2: Shared Bus Slot 3: Reserved

Bit	Attr	Reset Value	Description
29	RO	0x0	AsynIntSupport Asynchronous Interrupt Support Refer to SDIO Specification Version 3.00 about asynchronous interrupt. 1: Asynchronous Interrupt Supported 0: Asynchronous Interrupt Not Supported
28	RO	0x0	SystemBusSupport 64-bit System Bus Support 1: Supports 64 bit system address 0: Does not support 64 bit system address
27	RO	0x0	reserved
26	RO	0x1	Voltage18vSupport Voltage Support 1.8 V 0: Not Supported 1: Supported
25	RO	0x0	Voltage30vSupport Voltage Support 3.0 V 0: Not Supported 1: Supported
24	RW	0x0	Voltage33vSupport Voltage Support 3.3 V 0: Not Supported 1: Supported
23	RW	0x1	SuspendResumeSupport Suspend / Resume Support This bit indicates whether the HC supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism are not supported and the HD shall not issue either Suspend / Resume commands. 0: Not Supported 1: Supported
22	RO	0x1	SDMASupport This bit indicates whether the HC is capable of using DMA to transfer data between system memory and the HC directly. 0: SDMA Not Supported 1: SDMA Supported.
21	RO	0x1	HighSpeedSupport High Speed Support This bit indicates whether the HC and the Host System support High Speed mode and they can supply SD Clock frequency from 25Mhz to 50 Mhz (for SD)/ 20MHz to 52MHz (for eMMC). 0: High Speed Not Supported 1: High Speed Supported
20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19	RO	0x1	ADMA2Support ADMA2 Support 1: ADMA2 support 0: ADMA2 not support
18	RO	0x1	ExtendedMediaBusSupport Extended Media Bus Support This bit indicates whether the Host Controller is capable of using 8-bit bus width mode. This bit is not effective when Slot Type is set to 10b. In this case, refer to Bus Width Preset in the Shared Bus resister. 1: Extended Media Bus Supported 0: Extended Media Bus not Supported
17:16	RO	0x1	MaxBlockLength Max Block Length This value indicates the maximum block size that the HD can read and write to the buffer in the HC. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. 0: 512 byte 1: 1024 byte 2: 2048 byte 3: 4096 byte

Bit	Attr	Reset Value	Description
15:8	RO	0xc8	BaseClockFreqSDClock Base Clock Frequency for SD Clock (1) 6-bit Base Clock Frequency This mode is supported by the Host Controller Version 1.00 and 2.00. Upper 2-bit is not effective and always 0. Unit values are 1MHz. The supported clock range is 10MHz to 63MHz. 8'h00: Get information via another method 8'h01: 1MHz 8'h02: 2MHz ..... 8'h3f: 63MHz others: not supported (2) 8-bit Base Clock Frequency This mode is supported by the Host Controller Version 3.00. Unit values are 1MHz. The supported clock range is 10MHz to 255MHz. 8'h00: Get information via another method 8'h01: 1MHz 8'h02: 2MHz ..... 8'hff: 255MHz If the real frequency is 16.5MHz, the lager value shall be set 0001 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the SDCLK Frequency Select in the Clock Control register.) and it shall not exceed upper limit of the SD Clock frequency. If these bits are all 0, the Host System has to get information via another method.
7	RO	0x1	TimeoutClockUnit This bit shows the unit of base clock frequency used to detect Data Timeout Error. 0: Khz 1: Mhz
6	RO	0x0	reserved
5:0	RO	0x00	TimeoutClockFrequency This bit shows the base clock frequency used to detect Data Timeout Error. Not 0: 1Khz to 63Khz or 1Mhz to 63Mhz 0: Get Information viaanother method

EMMCCORE_FEACMD

Address: Operational Base + offset (0x0050)

Force event register for Auto CMD error status

Bit	Attr	Reset Value	Description
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	WO	0x0	CmdErr Force Event for command not issued by Auto CMD12 Error 1: Interrupt is generated 0: No interrupt
6:5	RO	0x0	reserved
4	WO	0x0	IndexErr Force Event for Auto CMD Index Error 1: Interrupt is generated 0: No interrupt
3	WO	0x0	EndErr Force Event for Auto CMD End bit Error 1: Interrupt is generated 0: No interrupt
2	WO	0x0	CrcErr Force Event for Auto CMD CRC Error 1: Interrupt is generated 0: No interrupt
1	WO	0x0	TimeoutErr Force Event for Auto CMD timeout Error 1: Interrupt is generated 0: No interrupt
0	WO	0x0	NotExe Force Event for Auto CMD12 NOT Executed 1: Interrupt is generated 0: No interrupt

EMMCCORE_FEERRINT
Address: Operational Base + offset (0x0052)
Force event register for error interrupt status

Bit	Attr	Reset Value	Description
15:12	RO	0x0	VendorErr Force Event for Vendor Specific Error Status
11:10	RO	0x0	reserved
9	WO	0x0	ADMAErr Force Event for ADMA Error 1: Interrupt is generated 0: No interrupt
8	WO	0x0	ACMDErr Force Event for Auto CMD Error 1: Interrupt is generated 0: No interrupt
7	WO	0x0	CurrentErr Force Event for Current Limit Error 1: Interrupt is generated 0: No interrupt

Bit	Attr	Reset Value	Description
6	WO	0x0	DatEndBitErr Force Event for Data End Bit Error 1: Interrupt is generated 0: No interrupt
5	WO	0x0	DatCRCErr Force Event for Data CRC Error 1: Interrupt is generated 0: No interrupt
4	WO	0x0	DatTimeoutErr Force Event for Data Timeout Error 1: Interrupt is generated 0: No interrupt
3	WO	0x0	CmdIndexErr Force Event for Command Index Error 1: Interrupt is generated 0: No interrupt
2	WO	0x0	CmdEndBitErr Force Event for Command End Bit Error 1: Interrupt is generated 0: No interrupt
1	WO	0x0	CmdCRCErr Force Event for Command CRC Error 1: Interrupt is generated 0: No interrupt
0	WO	0x0	CmdTimeoutErr Force Event for Command Timeout Error 1: Interrupt is generated 0: No interrupt

EMMCCORE_ADMAERRSTS

Address: Operational Base + offset (0x0054)

ADMA error status register

Bit	Attr	Reset Value	Description
15:3	RO	0x0	reserved
2	RO	0x0	LenMismatch ADMA Length Mismatch Error. While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. Total data length can not be divided by the block length. 1: Error 0: No error

Bit	Attr	Reset Value	Description
1:0	RO	0x0	ADMAErrorState This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. 0: ST_STOP (Stop DMA) Points to next of the error descriptor 1: ST_FDS (Fetch Descriptor) Points to the error descriptor 2: Never set this state (Not used) 3: ST_TFR (Transfer Data) Points to the next of the error descriptor

EMMCCORE_ADMAADDR
Address: Operational Base + offset (0x0058)
ADMA system address register

Bit	Attr	Reset Value	Description
63:32	RW	0x00000000	AddrH32 ADMA System Address [63:32].
31:0	RW	0x00000000	AddrL32 ADMA System Address [31:0]. This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 0.

EMMCCORE_PVALINIT
Address: Operational Base + offset (0x0060)
Preset value register for Initialization

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALDS
Address: Operational Base + offset (0x0062)

Preset value register for Default Speed

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALHS

Address: Operational Base + offset (0x0064)

Preset value register for High Speed

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALSDR12

Address: Operational Base + offset (0x0066)

Preset value register for SDR12

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALSDR25

Address: Operational Base + offset (0x0068)

Preset value register for SDR25

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALSDR50
Address: Operational Base + offset (0x006a)
Preset value register for SDR50

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALSDR104
Address: Operational Base + offset (0x006c)
Preset value register for SDR104

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_PVALDDR50
Address: Operational Base + offset (0x006e)
Preset value register for DDR50

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_BOOTTIMEOUT
Address: Operational Base + offset (0x0070)
Boot timeout control register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	BootTimeout Boot Data Timeout Counter Value This value determines the interval by which DAT line time-outs are detected during boot operation for eMMC card. The value is in number of sd clock.

EMMCCORE_PVALHS400
Address: Operational Base + offset (0x0074)
Preset value register for HS400

Bit	Attr	Reset Value	Description
15:14	RO	0x0	DriverStrengthSelectValue Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. 2'b11: Driver Type D is Selected 2'b10: Driver Type C is Selected 2'b01: Driver Type A is Selected 2'b00: Driver Type B is Selected
13:11	RO	0x0	reserved
10	RO	0x0	ClockGeneratorSelectValue This bit is effective when Host Controller supports programmable clockgenerator. 1: Programmable Clock Generator 0: Host Controller Ver2.00 Compatible Clock Generator
9:0	RO	0x000	SDCLKFrequencySelectValue 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

EMMCCORE_VENDOR
Address: Operational Base + offset (0x0078)
Vendor register

Bit	Attr	Reset Value	Description
15:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	EnhancedStrobe This bit enables the enhanced strobe logic of the Host Controller

EMMCCORE_SLOTINTSTS
Address: Operational Base + offset (0x00fc)
Slot interrupt status register

Bit	Attr	Reset Value	Description
15:1	RO	0x0	reserved
0	RO	0x0	IntSlot0 This status bit indicates the OR of Interrupt signal and Wakeup signal for slot

EMMCCORE_VERSION
Address: Operational Base + offset (0x00fe)
Host controller version register

Bit	Attr	Reset Value	Description
15:8	RO	0x10	VendorVersion The Vendor Version Number is set to 0x10 (1.0)
7:0	RO	0x02	SpecificationVersion The Host Controller Version Number is set to 0x02 (SD Host Specification Version 3.00).

EMMCCORE_CQVER
Address: Operational Base + offset (0x0200)
Command queueing version register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RO	0x5	Major eMMC Major Version Number (digit left of decimal point), in BCD format
7:4	RO	0x1	Minor eMMC Minor Version Number(digit right of decimal point), in BCD format
3:0	RO	0x0	Suffix eMMC Version Suffix (2nd digit right of decimal point), in BCD format

EMMCCORE_CQCAP
Address: Operational Base + offset (0x0204)
Command queueing capabilities register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:12	RO	0x0	ITCFMUL nternal Timer Clock Frequency Multiplier ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details. 4'h0: 0.001 MHz 4'h1: 0.01 MHz 4'h2: 0.1 MHz 4'h3: 1 MHz 4'h4: 10 MHz Other values are reserved
11:10	RO	0x0	reserved
9:0	RO	0x000	ITCFVAL Internal Timer Clock Frequency Value TCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_ STATUS (CMD13) polling. The clock frequency is calculated as ITCFVAL* ITCFMUL. For example, to encode 19.2 MHz, ITCFVAL shall be C0h (= 192 decimal) and ITCFMUL shall be 2h (0.1 MHz) 192 * 0.1 MHz=19.2 MHz

EMMCCORE_CQCFG
Address: Operational Base + offset (0x0208)
Command queueing configuration register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	DCMDEna Direct Command (DCMD) Enable This bit indicates to the hardware whether the Task Descriptor in slot #31 of the TDL is a Data Transfer Task Descriptor, or a Direct Command Task Descriptor. CQE uses this bit when a task is issued in slot #31, to determine how to decode the Task Descriptor. 1: Task descriptor in slot #31 is a DCMD Task Descriptor 0: Task descriptor in slot #31 is a Data Transfer Task Descriptor
11:9	RO	0x0	reserved
8	RW	0x0	TaskDescriptorSize This bit indicates whether the task descriptor size is 128 bits or 64 bits as detailed in Data Structures section. This bit can only be configured when Command Queueing Enable bit is 0 (command queueing is disabled) 1: Task descriptor size is 128 bits 0: Task descriptor size is 64 bits
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	CQEna Command Queueing Enable Software shall write 1 this bit when in order to enable command queueing mode (i.e. enable CQE). When this bit is 0, CQE is disabled and software controls the eMMC bus using the legacy eMMC host controller. Before software writes 1 to this bit, software shall verify that the eMMC host controller is in idle state and there are no commands or data transfers ongoing. When software wants to exit command queueing mode, it shall clear all previous tasks if such exist before setting this bit to 0.

EMMCCORE_CQCTRL
Address: Operational Base + offset (0x020c)
Command queueing control register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	R/WSC	0x0	ClearAllTasks Software shall write 1 this bit when it wants to clear all the tasks sent to the device. This bit can only be written when CQE is in halt state (i.e. Halt bit is 1). When software writes 1, the value of the register is updated to 1, and CQE shall reset CQTDBR register and all other context information for all unfinished tasks. Then CQE will clear this bit. Software should poll on this bit until it is set to back 0 and may then resume normal operation, by clearing the Halt bit. CQE does not communicate to the device that the tasks were cleared. It is software's responsibility to order the device to discard the tasks in its queue using CMDQ_TASK_MGMT command. Writing 0 to this register shall have no effect.
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	Halt Host software shall write 1 to the bit when it wants to acquire software control over the eMMC bus and disable CQE from issuing commands on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT). When software writes 1, CQE shall complete the ongoing task if such a task is in progress. Once the task is completed and CQE is in idle state, CQE shall not issue new commands and shall indicate so to software by setting this bit to 1. Software may poll on this bit until it is set to 1, and may only then send commands on the eMMC bus. In order to exit halt state (i.e. resume CQE activity), software shall clear this bit (write 0). Writing 0 when the value is already 0 shall have no effect.

EMMCCORE_CQINTSTS
Address: Operational Base + offset (0x0210)
Command queueing interrupt status register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	W1C	0x0	TERR Task Error Interrupt This bit is asserted when task error is detected due to invalid task descriptor
3	W1C	0x0	TCL Task Cleared This status bit is asserted (if CQISTE.TCL=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (CQTCLR) or clearing of all tasks (CQCTL).
2	W1C	0x0	RED Response Error Detected Interrupt This status bit is asserted (if CQISTE.RED=1) when a response is received with an error bit set in the device status field. Software uses CQRMEM register to configure which device status bit fields may trigger an interrupt, and which are masked.
1	W1C	0x0	TCC Task Complete Interrupt This status bit is asserted(if CQISTE.TCC=1) when at least one of the following two conditions are met: a. A task is completed and the INT bit is set in its Task Descriptor b. Interrupt caused by Interrupt Coalescing logic

Bit	Attr	Reset Value	Description
0	W1C	0x0	HAC Halt Complete Interrupt This status bit is asserted (if CQISTE.HAC=1) when halt bit in CQCTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state.

EMMCCORE_CQINTSTSENA
Address: Operational Base + offset (0x0214)
Command queueing interrupt status enable register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	TERR Task Error Interrupt 1: enable 0: disable
3	RW	0x0	TCL Task Cleared 1: enable 0: disable
2	RW	0x0	RED Response Error Detected Interrupt 1: enable 0: disable
1	RW	0x0	TCC Task Complete Interrupt 1: enable 0: disable
0	RW	0x0	HAC Halt Complete Interrupt 1: enable 0: disable

EMMCCORE_CQINTSIGENA
Address: Operational Base + offset (0x0218)
Command queueing interrupt signal enable register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	TERR Task Error Interrupt 1: enable 0: disable

Bit	Attr	Reset Value	Description
3	RW	0x0	TCL Task Cleared 1: enable 0: disable
2	RW	0x0	RED Response Error Detected Interrupt 1: enable 0: disable
1	RW	0x0	TCC Task Complete Interrupt 1: enable 0: disable
0	RW	0x0	HAC Halt Complete Interrupt 1: enable 0: disable

EMMCCORE_CQINTCOAL
Address: Operational Base + offset (0x021c)
Command queueing interrupt coalescing register

Bit	Attr	Reset Value	Description
31	RW	0x0	IntCoalEna Interrupt Coalescing Enable/Disable: When set to 0 by software, command responses are neither counted nor timed. Interrupts are still triggered by completion of tasks with INT=1 in the Task Descriptor. When set to 1, the interrupt coalescing mechanism is enabled and coalesced interrupts are generated.
30:21	RO	0x0	reserved
20	RO	0x0	ICSB Interrupt Coalescing Status Bit : This bit indicates to software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (i.e., ICSB is set ifand only if IC counter > 0). 1: At least one task completion has been counted (IC counter >0) 0: No task completions have occurred since last counter reset (IC counter =0)
19:17	RO	0x0	reserved
16	WO	0x0	Reset Counter and Timer Reset(ICCTR): When host driver writes 1, the interrupt coalescing timer and counter are reset

Bit	Attr	Reset Value	Description
15	WO	0x0	ICCTHWEN Interrupt Coalescing Counter Threshold Write Enable: When software writes 1, the value ICCTH is updated with the contents written at the same cycle. When software writes 0, the value in ICCTH is not updated. NOTE: Write operations to ICCTH are only allowed when the task queue is empty.
14:13	RO	0x0	reserved
12:8	RW	0x00	ICCTH Interrupt Coalescing Counter Threshold (ICCTH): Software uses this field to configure the number of task completions (only tasks withINT=0 in the Task Descriptor) which are required in order to generate an interrupt. Counter Operation: As data transfer tasks with INT=0 complete, they are counted byCQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in ICCTH. The maximum allowed value is 31 NOTE : When ICCTH is 0, task completions are not counted, and counting-based interrupts are not generated. In order to write to this field, the ICCTHWEN bit must be set at the same write operation.
7	WO	0x0	ICTOVALWEN Interrupt Coalescing Timeout Value Write Enable: When software writes 1, the value ICTOVAL is updated with the contents written at the same cycle. When software writes 0, the value in ICTOVAL is not updated. NOTE: Write operations to ICTOVAL are only allowed when the task queue is empty.

Bit	Attr	Reset Value	Description
6:0	RW	0x00	ICTOVAL Interrupt Coalescing Timeout Value (ICTOVAL): Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when a data transfer task with INT=0 is completed, after the timer was reset. When the timer reaches the value configured in ICTOVAL field it generates an interrupt and stops. The timer's unit is equal to 1024 clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 01h (1024 clock periods) and the maximum value is 7Fh (127*1024 clock periods). For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in ICTOVAL is 10h, the calculated polling period is 16*1024*52.08 ns= 853.33 us. NOTE:When ICTOVAL is 0, the timer is not running, and timer-based interrupts are not generated.In order to write to this field, the ICTOVALWEN bit must be set at the same write operation.

EMMCCORE_CQTDLBA
Address: Operational Base + offset (0x0220)
Command queueing task descriptor list base address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDLBA Task Descriptor List Base Address This register stores the LSB bits (bits 31:0) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address shall be set on Byte1 KByte boundary. The lower 10 bits of this register shall be set to 0 by software and shall be ignored by CQE.

EMMCCORE_CQTDLBAU
Address: Operational Base + offset (0x0224)
Command queueing task descriptor list base address upper 32bits register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TDLBA Task Descriptor List Base Address This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This register is reserved when using 32-bit addressing mode.

EMMCCORE_CQTDB
Address: Operational Base + offset (0x0228)
Command queueing task doorbell register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TaskDoorbell Command Queueing Task Doorbell Software shall configure TDLBA and TDLBAU, and enable CQE in CQCFG before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. CQE always processes tasks in-order according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device. CQE processes DCMD tasks (in slot #31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to 0 by CQE in one of the following events: a. When a task execution is completed (with success or error) b. The task is cleared using CQTCLR register c. All tasks are cleared using CQCTL register d. CQE is disabled using CQCFG register Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission: CQE shall process the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution will be based on said processing order. Writing 0 by software shall have no impact on the hardware, and will not change the value of the register bit.

EMMCCORE_CQTDBN
Address: Operational Base + offset (0x022c)

Command queueing task doorbell notification register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TCN Task Complete Notification CQE shall set bit n of this register (at the same time it clears bit n of CQTDBR) when a task execution is completed (with success or error). When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

EMMCCORE_CQDQSTS

Address: Operational Base + offset (0x0230)

Command queueing device queue status register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DQS Device Queue Status Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command, i.e. the device's queue status.

EMMCCORE_CQDPT

Address: Operational Base + offset (0x0234)

Command queueing device pending tasks register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DPT Device Pending Tasks Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task hasn’t been executed yet. CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution. Software needs to read this register in the task-discard procedure, when the controlleris halted, to determine if the task is queued in the device. If the task is queued, the driver sends a CMDQ_TASK_MGMT (CMD48) to the device ordering it to discard the task. Then software clears the task in the CQE. Only then the software orders CQE to resume its operation using CQCTL register.

EMMCCORE_CQTCLR

Address: Operational Base + offset (0x0238)

Command queueing task clear register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CQTC Command Queueing Task Clear Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFG register Halt bit. When software writes 1 to a bit in this register, CQE updates the value to 1, and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once clear operation is complete. Software should poll on the CQTCLR until it is cleared to verify clear operation was complete. Writing to this register only clears the task in the CQE and does not have impact on the device. In order to discard the task in the device, host softwareshall send CMDQ_TASK _MGMT while CQE is still in Halt state. Host driver is not allowed to use this register to clear multiple tasks at the same time. Clearing multiple tasks can be done using CQCTL register. Writing 0 to a register bit shall have no impact.

EMMCCORE_CQSSC1
Address: Operational Base + offset (0x0240)
Command queueing send status configuration register 1

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x1	SSCBC Send Status Command Block Counter This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device's task queue. A value of n means CQE shall send status command on the CMD line, during the transfer of data block BLOCK_CNT-n, on the data lines, where BLOCK_CNT is the number of blocks inthe current transaction. A value of 0 means that SEND_QUEUE_STATUS (CMD13) command shall not be sent during the transaction. Instead it will be sentonly when the data lines are idle. A value of 1 means that STATUS command is to be sent during the last block of the transaction.

Bit	Attr	Reset Value	Description
15:0	RW	0x1000	SSCIT Send Status Command Idle Timer This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicating that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS. Timer units are clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP egister. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). Default interval is: 4096 clock periods. For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in CQSST is 1000h, the calculated polling period is 4096*52.08 ns= 213.33 us.

EMMCCORE_CQSSC2

Address: Operational Base + offset (0x0244)

Command queueing send status configuration register 2

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	SQRCA Send Queue RCA This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_ STATUS (CMD13) command. argument. CQE shall copy this field to bits 31:16 of the argument when transmitting SEND_ QUEUE_STATUS (CMD13) command

EMMCCORE_CQCRDT

Address: Operational Base + offset (0x0248)

Command queueing command response for direct-command task register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DCLR Direct Command Last Response This register contains the response of the command generated by the last direct-command (DCMD) task which was sent. CQE shall update this register when it receives the response for a DCMD task. This register is considered valid only after bit 31 of CQTDBR register is cleared by CQE.

EMMCCORE_CQRMEM

Address: Operational Base + offset (0x0250)
Command queueing response mode error mask register

Bit	Attr	Reset Value	Description
31:0	RO	0xfdf9a080	RMEM Response Mode Error Mask This bit is used as in interrupt mask on the device status filed which is received in R1/R1b responses. Bit Value Description (for any bit i): 1: When a R1/R1b response is received, with bit i in the device status set, a RED interrupt is generated 0: When a R1/R1b response is received, bit i in the device status is ignored The reset value of this register is set to trigger an interrupt on all 'Error' type bits in the device status. NOTE: Responses to CMD13 (SQS) encode the QSR, so they are ignored by this logic.

EMMCCORE_CQTEI
Address: Operational Base + offset (0x0254)
Command queueing task error information register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:16	RO	0x00	DTECI Data Transfer Error Command Index This field indicates the index of the command which was executed on the data lines when an error occurred. The index shall be set to EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) according to the data direction. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
15	RO	0x0	RMEFV Response Mode Error Fields Valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a command transaction is in progress when the error is detected/indicated, the bit is set to 1. If a no command transaction is in progress when the error is detected/indicated, the bit is cleared to 0.
14:13	RO	0x0	reserved
12:8	RO	0x00	RMETID Response Mode Error Task ID This field indicates the ID of the task which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RO	0x00	RMECI Response Mode Error Command Index This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.

EMMCCORE_CQCRI
Address: Operational Base + offset (0x0258)
Command queueing command response index register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	LCRI Last Command Response Index This field stores the index of the last received command response. CQE shall update the value every time a command response is received.

EMMCCORE_CQCRA
Address: Operational Base + offset (0x025c)
Command queueing command response argument register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	LCRA Last Command Response Argument This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

16.5 Miscellaneous Signals

16.5.1 Miscellaneous Configure Signals for CORE

Table 16-1 Miscellaneous Configure Signals for CORE

Signal Name	Source	Def	Description
corecfg_tuningcount[5:0]	GRF_EMMCCORE_CON0[5:0]	6'h20	Tuning Count. The Tuning State machine uses this information to select one of the Taps (Phases) of the rxclk_in during the Tuning Procedure.
corecfg_timeoutclkunit	GRF_EMMCCORE_CON0[7]	1	Timeout Clock Unit.0-1ms/KHz，1-1us/MHz
corecfg_baseclkfreq[7:0]	GRF_EMMCCORE_CON0[15:8]	8’hc8	Base Clock Frequency. This is the frequency of the xin_clk.
corecfg_initpresetval[12:0]	GRF_EMMCCORE_CON1[12:0]	0	Initialization Phase Preset Val
corecfg_dspdpresetval[12:0]	GRF_EMMCCORE_CON2[12:0]	13'h804	Default Speed preset value. [9:0]: SDCLK Frequency Select Value [10]: Clock Generator Select Value.0: Host Controller Ver2.00 Compatible clock Generator; 1: Programmable Clock Generator [12:11]: reserved

Signal Name	Source	Def	Description
corecfg_hspdpresetval[12:0]	GRF_EMMCCORE_CON3[12:0]	13'h002	High Speed preset value
corecfg_sdr12presetval[12:0]	GRF_EMMCCORE_CON4[12:0]	13'h804	SDR12 Preset value
corecfg_sdr25presetval[12:0]	GRF_EMMCCORE_CON5[12:0]	13'h002	SDR25 Preset value
corecfg_sdr50presetval[12:0]	GRF_EMMCCORE_CON6[12:0]	13'h801	SDR50 Preset value
corecfg_sdr104presetval[12:0]	GRF_EMMCCORE_CON7[12:0]	13'h000	SDR104 Preset value
corecfg_ddr50presetval[12:0]	GRF_EMMCCORE_CON8[12:0]	13'h802	DDR50 Preset value
corecfg_hs400presetval[12:0]	GRF_EMMCCORE_CON9[12:0]	13'h000	HS400 Preset value
corecfg_clockmultiplier[7:0]	GRF_EMMCCORE_CON11[7:0]	8'h10	This field indicates clock multiplier value of programmable clock generator. Refer to Clock Control register. Setting 00h means that Host Controller does not support programmable clock generator. FFh Clock Multiplier M = 256 .... 02h Clock Multiplier M = 3 01h Clock Multiplier M = 2 00h Clock Multiplier is Not Supported

16.5.2 Miscellaneous Configure Signals for PHY

Table 16-2 Miscellaneous Configure Signals for PHY

Signal Name	Source	Def	Description
phyctrl_itapdlyena	GRF_EMMCPHY_CON0[0]	0	Input Tap Delay Enable. This is used for the manual control of the rxclk Tap Delay in non HS200/HS400 modes.
phyctrl_itapdlysel[4:0]	GRF_EMMCPHY_CON0[5:1]	0	Input Tap Delay Select. This is used for the manual control of the rxclk Tap Delay in the non HS200/HS400 modes.
phyctrl_itapchgwin	GRF_EMMCPHY_CON0[6]	0	Input Tap Change Window. This is to be asserted while changing the phyctrl_itapdlysel. This is used to gate of the rxclk to avoid clock glitches while the tap is changing.
phyctrl_otapdlysel[3:0]	GRF_EMMCPHY_CON0[10:7]	0	Output Tap Delay Select (0 to 15). This is used for the manual control of the txclk Tap Delay, for flopping the final stage flops for maintaining Hold requirements on eMMC Interface.
phyctrl_otapdlyena	GRF_EMMCPHY_CON0[11]	0	Output Tap Delay Enable
phyctrl_frqsel[1:0]	GRF_EMMCPHY_CON0[13:12]	0	Select the frequency range of DLL operation: 0: 200MHz 1: 50MHz 2: 100MHz 3: 150MHz
phyctrl_strbsel[3:0]	GRF_EMMCPHY_CON1[3:0]	0	Tap Select for STRB_90 and STRB_180. [1:0]: strb90 tap point. 0: DLLSTRBPH[4] 1: DLLSTRBPH[5] 2: DLLSTRBPH[6] 3: DLLSTRBPH[7] [3:2]: strb180 tap point. 0: DLLSTRBPH[12] 1: DLLSTRBPH[13] 2: DLLSTRBPH[14] 3: DLLSTRBPH[15]
phyctrl_seldlyrxclk	GRF_EMMCPHY_CON1[4]	0	Select DLY Chain based RX_CLK. When enabled the Delay Chain based rxclk is selected rather than DLL based. 0: rxclk = dll_rxclk

Signal Name	Source	Def	Description
			1: rxclk = dly_rxclk
phyctrl_seldlytxclk	GRF_EMMCPHY_CON1[5]	0	Select DLY Chain based TX_CLK. When enabled the Delay Chain based txclk is selected rather than DLL based. 0: rxclk = dll_txclk 1: rxclk = dly_txclk
phyctrl_clkbufsel[2:0]	GRF_EMMCPHY_CON1[8:6]	0	Clock buffer select for delay buffer.
phyctrl_ren_dat[7:0]	GRF_EMMCPHY_CON2[7:0]	8'hff	REN Enable on DAT Lines
phyctrl_ren_cmd	GRF_EMMCPHY_CON2[8]	1	REN Enable on CMD Line
phyctrl_ren_strb	GRF_EMMCPHY_CON2[9]	0	REN Enable on STRB Line
phyctrl_pu_dat[7:0]	GRF_EMMCPHY_CON3[7:0]	8'hff	Pullup Enable on DAT Lines
phyctrl_pu_cmd	GRF_EMMCPHY_CON3[8]	1	Pullup Enable on CMD Line
phyctrl_pu_strb	GRF_EMMCPHY_CON3[9]	0	Pullup Enable on STRB Line
phyctrl_OD_release_dat[7:0]	GRF_EMMCPHY_CON4[7:0]	0	OD release for DAT Lines. Disable an internal 4.7K pull up resistor in open drain mode on DAT line
phyctrl_OD_release_cmd	GRF_EMMCPHY_CON4[9]	0	OD release for CMD Line. Disable an internal 4.7K pull up resistor in open drain mode on CMD line
phyctrl_OD_release_strb	GRF_EMMCPHY_CON4[8]	0	OD release for STRB Line. Disable an internal 4.7K pull up resistor in open drain mode on STRB line
phyctrl_oden_dat[7:0]	GRF_EMMCPHY_CON5[7:0]	0	OPEN Drain Enable on DAT Lines
phyctrl_oden_cmd	GRF_EMMCPHY_CON5[8]	0	OPEN Drain Enable on CMD Line
phyctrl_oden_strb	GRF_EMMCPHY_CON5[9]	0	OPEN Drain Enable on STRB Line
phyctrl_pdb	GRF_EMMCPHY_CON6[0]	0	PowerDown# from SOC. SOC asserts after power up sequence is completed.
phyctrl_endll	GRF_EMMCPHY_CON6[1]	0	Enable DLL
phyctrl_reten	GRF_EMMCPHY_CON6[2]	0	Retention Enable. Enable retention mode. EMMCIOs will retain its input programming state during VCORE is off in sleep mode.
phyctrl_retenb	GRF_EMMCPHY_CON6[3]	1	Retention Enable#. Retention mode enable bar.
phyctrl_dr_ty[2:0]	GRF_EMMCPHY_CON6[6:4]	0	Drive Type. Drive Source/Sink impedance programming
phyctrl_retrim	GRF_EMMCPHY_CON6[7]	0	Retrim. Positive Edge initiates a CALIO calibration cycle.
phyctrl_en_rtrim	GRF_EMMCPHY_CON6[8]	1	Enable CALIO Pad. Enable CALIO to trim eMMC IOs source/sink impedance
phyctrl_dll_trm_icp[3:0]	GRF_EMMCPHY_CON6[12:9]	8	DLL Trim Bits.

16.5.3 Miscellaneous Status Signals for CORE

Table 16-3 Miscellaneous Status Signals for CORE

Signal Name	Destination	Description
emmccore_dmadebugbus[15:0]	GRF_EMMCCORE_STATUS0[15:0]	DMA_CTRL Debug Bus. **[15]: hostintf_blocknextcmd** – Command Complete indication from SDHC_CMDCTRL is set, the next Command Indication should be blocked **[14]: hostintf_abortcmdmode** - When the Abort command is issued, the flag is set

Signal Name	Destination	Description
		[13]: hostintf_rdxferactive - This is set from the time the Command Last Bit is issued or Block Gap Continue to the time the last block of data is sent to Host **[12]: hostintf_enddataxfer** - It is set when the end Data Transfer Complete Indications from the PIO/SDMA and ADMA2 State Machines **[11]: hostintf_stopatblkgap** - For write transfer, It is set from the PIO/SDMA and ADMA2 State Machines. For read transfer, RegSet is used to generate Stop at BlkGap **[10]: hostwrdat_state** - The Write Data State machine Transfers the Data to Host Interface **[9]: hostrddat_state** - The Read Data State machine receives the Data from Host Interface **[8:7]: hosttrans_state[1:0]** - Host Transfer State Machine. 0-IDLE, 1-PRE, 2-REQ, 3-EOF **[6:3]: adma2_state[3:0]** - ADMA2 State Machine. 0-IDLE, 1-CMDRESP, 2-DESCRD0, 3-DESCRD1, 4-DATXFER0, 5-DATXFER1, 6-BLOCKGAP, 7-COMPLETE **[2:0]: piosdma_state[2:0]** - PIO/SDMA State Machine. 0-IDLE, 1-CMDRESP, 2-DATXFER0, 3-DATXFER1, 4-BLOCKGAP, 5-SDMAINTR, 6-COMPLETE, 7-NEXTBOOT
emmccore_cmddebugbus[15:0]	GRF_EMMCCORE_STATUS0[31:16]	CMD_CTRL Debug Bus. **[15:12]: cmdfsm_cmdrespstatus [3:0]**- Command Response Status **[11]: sdhcregset_bootena_sdclk** - Double Synchronization of sdhcregset_bootena Signal **[10]: sdhcregset_cmdexecute_sdclk** - Double Synchronization of sdhcregset_cmdexecute Signal **[9]: cmdfsm_autocmd23** - It is set when AutoCMD 23 Qualified with CMD_Data Present **[8]: cmdfsm_autocmd12** - Auto CMD12 Indication **[7]: cmdfsm_cmdissued** - CMD Issued Signal **[6]: cmdfsm_cmdcomplete** - Command complete bit **[5]: cmdfsm_cmdena** - CMD Output Enable **[4]: cmdfsm_cmdout** - CMD Output **[3:0]: cmdfsm_state[3:0]** - Command State Machine. 0-IDLE, 1-WAITCLK, 2-SENDSEQ, 3-SENDCRC, 4-SENDEND, 5-WTFERRESP, 7-CMDBOOT, 8-SPISTOP, 9-NORESPCPL
emmccore_txddebugbus[15:0]	GRF_EMMCCORE_STATUS1[15:0]	TXD_CTRL Debug Bus. **[15]: txdfsm_enatimeoutchk** - Enable Data Timeout Check **[14]: txdfsm_stopsdcardclk** - Stop SDCard Clock **[13:11]: txdfsm_xmitstatus[2:0]** - Transmit Status ENDBIT Err, CRC Error, Timeout Error} **[10]: txdfsm_xmitstsvld** - Transmit Status Valid Indication **[9]: txdfsm_rcvcrcsts** - Receive CRC Status Indication to RXCRC State Machine **[8]: txdfsm_readeob** - Read EOB **[7]: txdfsm_readbuffer** - Read Buffer **[6]: txdfsm_sddataena** - SD Data Enable **[5]: txdfsm_wrxferactive** - Wr Transfer Active Indication **[4]: txdfsm_datalineactive** - Data Line Active Indicaiton **[3:0]: txdfsm_state[3:0]** - Transmit Data State Machine. 0-IDLE, 1-BLKWAIT, 2-BLKSTART, 3-SENDBLK, 4-SENDCRC, 5-SENDEND, 6-WAITCRC, 7-WAITBSY, 8-STOPBLKGAP
emmccore_rxddebugbus0[15:0]	GRF_EMMCCORE_STATUS1[31:16]	RXD_CTRL Debug Bus (SD CLK) [15:8]: 8'b0 **[7:5]: rxctrl_rcvstatus[2:0]** - Receive Status {EndBit Err, CRC Err, TimeoutErr} **[4]: rxctrl_rcvstsvld** - Receive Status Valid **[3]: rxctrl_stopafterblk** - Stop Receiving After this Block (Stop at Blk Gap) **[2]: rxctrl_stopsdcardclk2** - Stop SD Clock Indication (Stop at Block Gap) **[1]: rxctrl_stopsdcardclk1** - Stop SD Clock Indication (Normal Operation, Buffer Full) **[0]: rxctrl_rcvdata** - Receive Data Indication

Signal Name	Destination	Description
emmccore_rxddebugbus1[15:0]	GRF_EMMCCORE_STATUS2[15:0]	RXD_CTRL Debug Bus (RX CLK) [15:8]: 8'b0 **[7]: rxdfsm_reachingeob** - Reaching End of Block Indication **[6]: rxdfsm_okstopclk** - OK to Stop the Clock (End of Block Flow Control) **[5]: rxdfsm_dataeob** - Data Byte/Word is End Of Block Indication **[4]: rxdfsm_datawrite** - Data Byte/Word Write **[3]: rxdfsm_wtforblk** - Wait for Block Indication **[2:0]: rxdfsm_state [2:0]** - Receive Data State Machine. 0-RCVSTART, 1-RCVDATA, 2-RCVEND, 4-RDWAIT, 5-RCVBOOTACK
emmccore_tundebugbus[15:0]	GRF_EMMCCORE_STATUS2[31:16]	TUN_CTRL Debug Bus **[15]: tuningfsm_done** - Tuning Done Indication **[14:9]: tuningfsm_numseqmatch [5:0]** - Number of Sequential Matches **[8:3]: tuningfsm_count [5:0]** - Tuning Count **[2:0]: tuningfsm_state[2:0]** - The Tuning FSM waits for Execute_Tuning Request from Software and starts the Tuning Procedure. 0-IDLE, 1-WTFORCMD, 2-WTCORDAT, 3-WTFORCRC, 4-NEXTITER, 5-SELCLOCK0, 6-SELCLOCK1
emmccore_cmddir	GRF_EMMCCORE_STATUS3[0]	To indicate the command Direction 0: Write (Host to Card) 1: Read (Card to Host)
emmccore_cmdidle	GRF_EMMCCORE_STATUS3[1]	Idle signal to enable S/W to gate off the clocks: 0: active 1: idle

16.5.4 Miscellaneous Status Signals for PHY

Table 16-4 Miscellaneous Status Signals for PHY

Signal Name	Destination	Description
phyctrl_caldone	GRF_EMMCPHY_STATUS[6]	Indicate that CALIO Calibration is completed successfully. Power on default 1b'0.
phyctrl_dllrdy	GRF_EMMCPHY_STATUS[5]	DLL ready. Indicates that DLL loop is locked. Power on Default 1b'0
phyctrl_rtrim[3:0]	GRF_EMMCPHY_STATUS[4:1]	CALIO Calibration Result. Holds the content of CALIO Impedance Calibration Result. Power on default 4b'1110.
phyctrl_exr_ninst	GRF_EMMCPHY_STATUS[0]	External Resistor on CALIO absent. Indicates trim cycle started and external resistor is absent. Power on Default 1b'0.

16.6 Interface Description

Table 16-5 EMMC Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
emmc_cclk	O	IO_EMMC_CLK	N/A
emmc_ccmd	I/O	IO_EMMC_CMD	N/A
emmc_cdata0	I/O	IO_EMMC_DATA0	N/A
emmc_cdata1	I/O	IO_EMMC_DATA1	N/A
emmc_cdata2	I/O	IO_EMMC_DATA2	N/A
emmc_cdata3	I/O	IO_EMMC_DATA3	N/A
emmc_cdata4	I/O	IO_EMMC_DATA4	N/A
emmc_cdata5	I/O	IO_EMMC_DATA5	N/A
emmc_cdata6	I/O	IO_EMMC_DATA6	N/A
emmc_cdata7	I/O	IO_EMMC_DATA7	N/A
emmc_strb	I	IO_EMMC_STRB	N/A
emmc_pwren	O	IO_EMMCpwren_PMUdebug3_PMU18gpio0a5	GRF_GPIO0A_IOMUX[11:10]=2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

16.7 Application Notes

16.7.1 SD Clock Control

1. SD Clock Supply Sequence

The clock shall be supplied to the card before either of the following actions is taken.

- Issuing a SD command
- Detect an interrupt from a SD card in 4-bit mode

The sequence for supplying SD Clock to a SD card is described as follows.

7) Calculate a divisor to determine SD Clock frequency by reading Base Clock Frequency For SD Clock in the Capabilities register.
8) Set Internal Clock Enable and SDCLK Frequency Select in the Clock Control register.
9) Check Internal Clock Stable in the Clock Control register. Repeat this step until Clock Stable is 1.
10) Set SD Clock Enable in the Clock Control register to 1. Then, the Host Controller starts to supply the SD Clock.

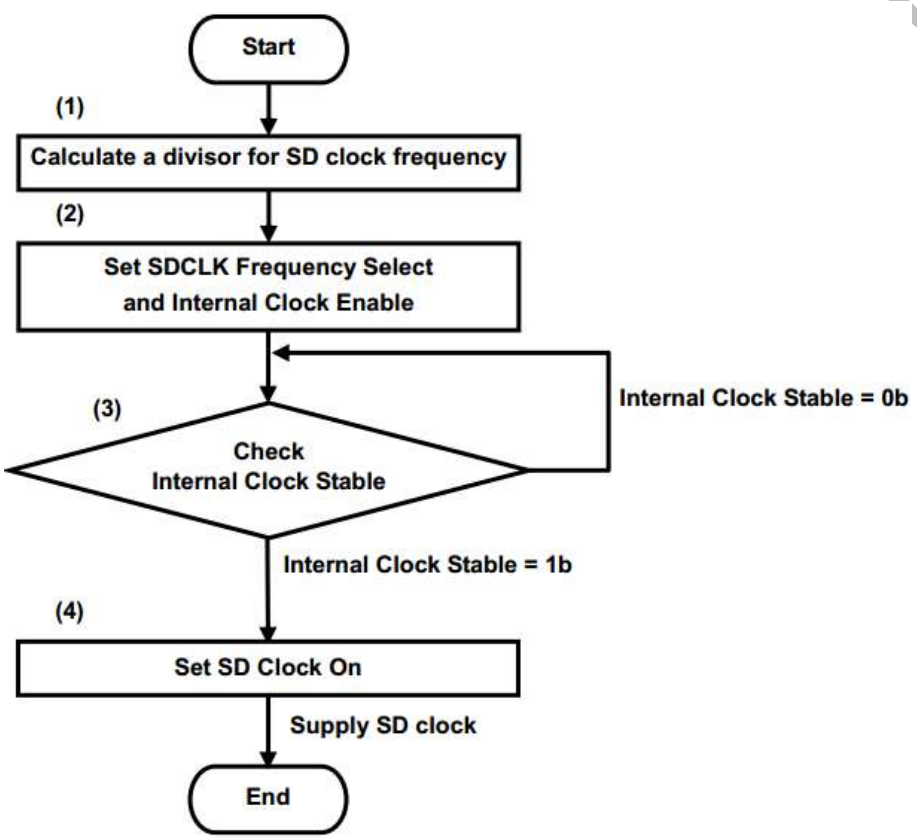


Fig. 16-2 SD Clock Supply Sequence

2. SD Clock Stop Sequence

Set SD Clock Enable in the Clock Control register to 0. Then, the Host Controller stops supplying the SD Clock.

The Host Driver shall not stop the SD Clock when a SD transaction is occurring on the SD Bus -- namely, when either Command Inhibit (DAT) or Command Inhibit (CMD) in the Present State register is set to 1.

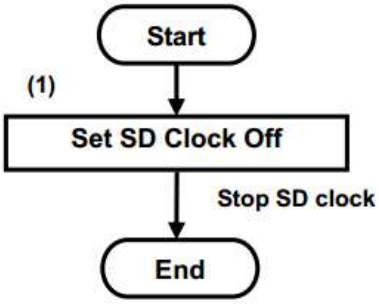


Fig. 16-3 SD Clock Stop Sequence

3. SD Clock Frequency Change Sequence

The sequence for changing SD Clock frequency is shown in following figure. When SD Clock is still off, step (1) is omitted.

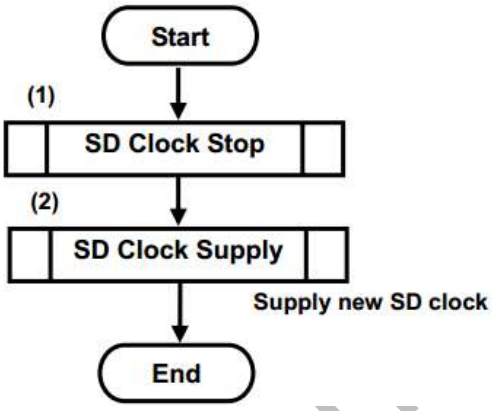


Fig. 16-4 SD Clock Change Sequence

16.7.2 Transaction Control without Data Transfer Using DAT Line

1. The Sequence to Issue a SD Command

The sequence to issue the SD Command is detailed below.

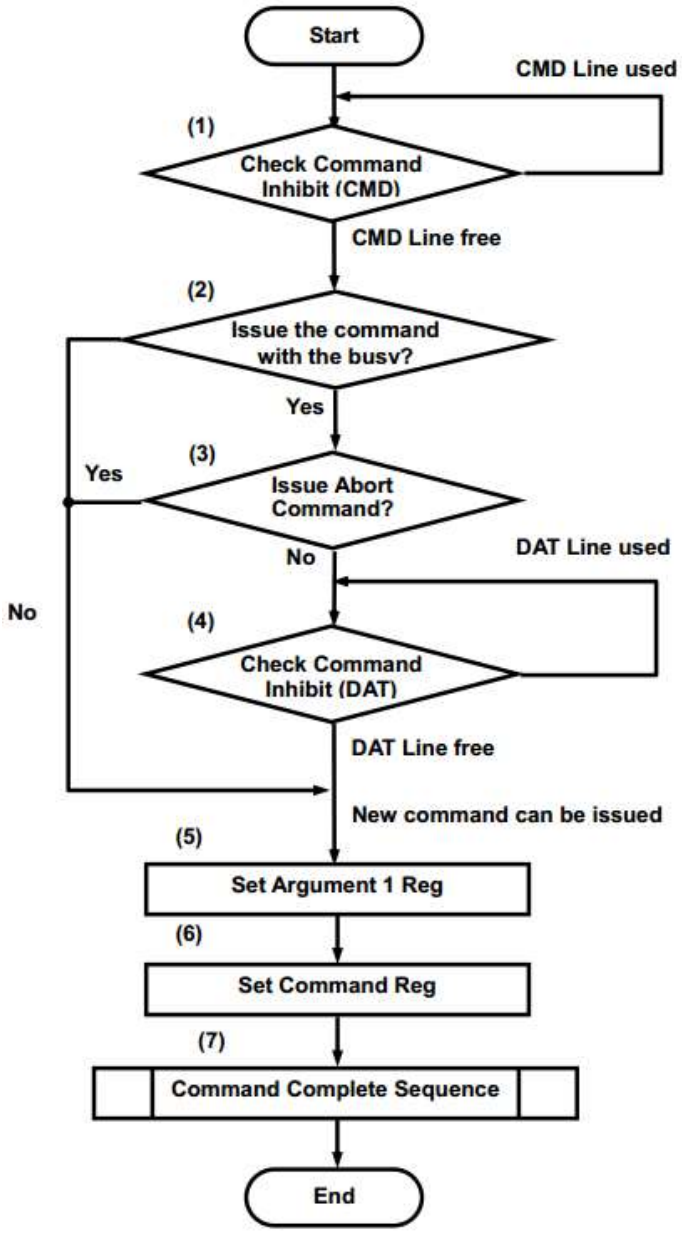


Fig. 16-5 SD Command Issue Sequence

1) Check Command Inhibit (CMD) in the Present State register. Repeat this step until Command Inhibit (CMD) is 0. That is, when Command Inhibit (CMD) is 1, the Host Driver shall not issue a SD Command.
2) If the Host Driver issues a SD Command with busy signal, go to step (3). If without busy signal, go to step (5).
3) If the Host Driver issues an abort command, go to step (5). In the case of no abort command, go to step (4).
4) Check Command Inhibit (DAT) in the Present State register. Repeat this step until Command Inhibit (DAT) is set to 0.
5) Set the value of command argument to the Argument 1 register.
6) Set the Command register.
7) Perform Command Completion Sequence.

2. The Sequence to Finalize a Command

Following figure shows the sequence to finalize a SD Command. There is a possibility that some errors (Command Index/End bit/CRC/Timeout Error) occur during this sequence.

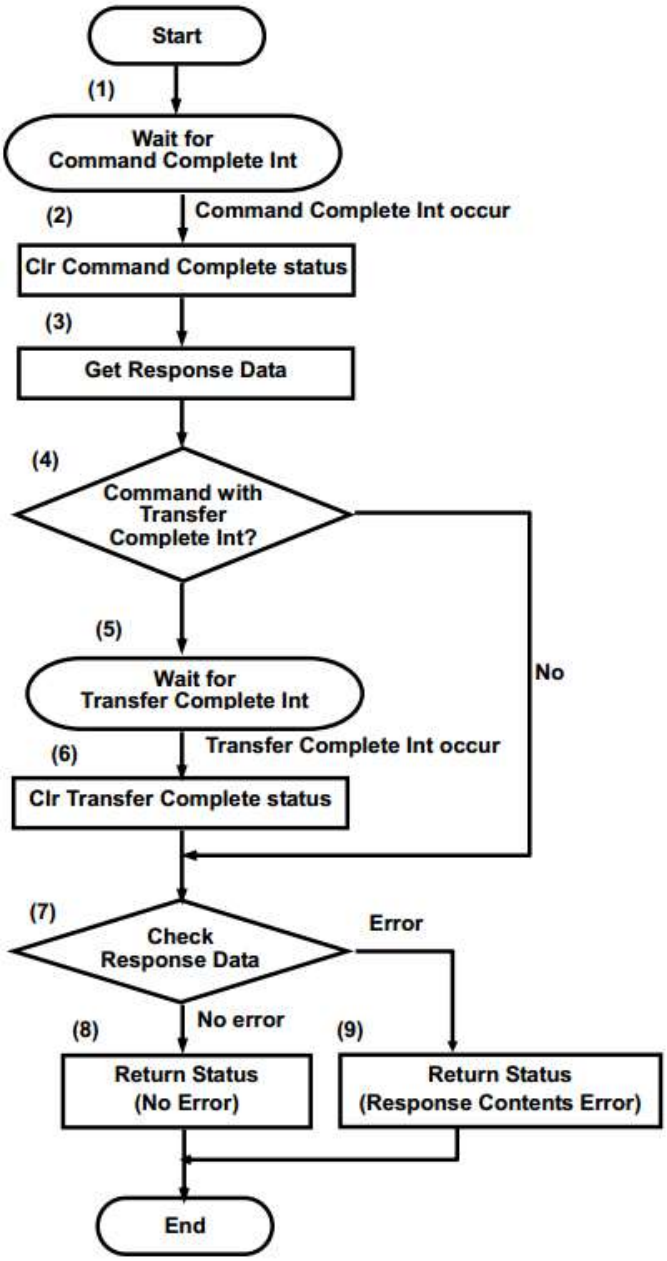


Fig. 16-6 Command Complete Sequence

1) Wait for the Command Complete Interrupt. If the Command Complete Interrupt has occurred, go to step (2).
2) Write 1 to Command Complete in the Normal Interrupt Status register to clear this bit.
3) Read the Response register and get necessary information of the issued command.
4) Judge whether the command uses the Transfer Complete Interrupt or not. If it uses Transfer Complete, go to step (5). If not, go to step (7).
5) Wait for the Transfer Complete Interrupt. If the Transfer Complete Interrupt has occurred, go to step (6).
6) Write 1 to Transfer Complete in the Normal Interrupt Status register to clear this bit.
7) Check for errors in Response Data. If there is no error, go to step (8). If there is an error, go to step (9).
8) Return Status of "No Error".
9) Return Status of "Response Contents Error".

Note:

- *While waiting for the Transfer Complete interrupt, the Host Driver shall only issue commands that do not use the busy signal.*
- *The Host Driver shall judge the Auto CMD12 complete by monitoring Transfer Complete.*
- *When the last block of un-protected area is read using memory multiple block read command (CMD18), OUT_OF_RANGE error may occur even if the sequence is correct. The Host Driver should ignore it. This error will appear in the response of Auto CMD12 or in the response of the next memory*

command.

16.7.3 Transaction Control with Data Transfer Using DAT Line

1. Determination of Transfer Type

Table 16-6 Determination of Transfer Type

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

2. Not using DMA

The sequence for not using DMA is shown below.

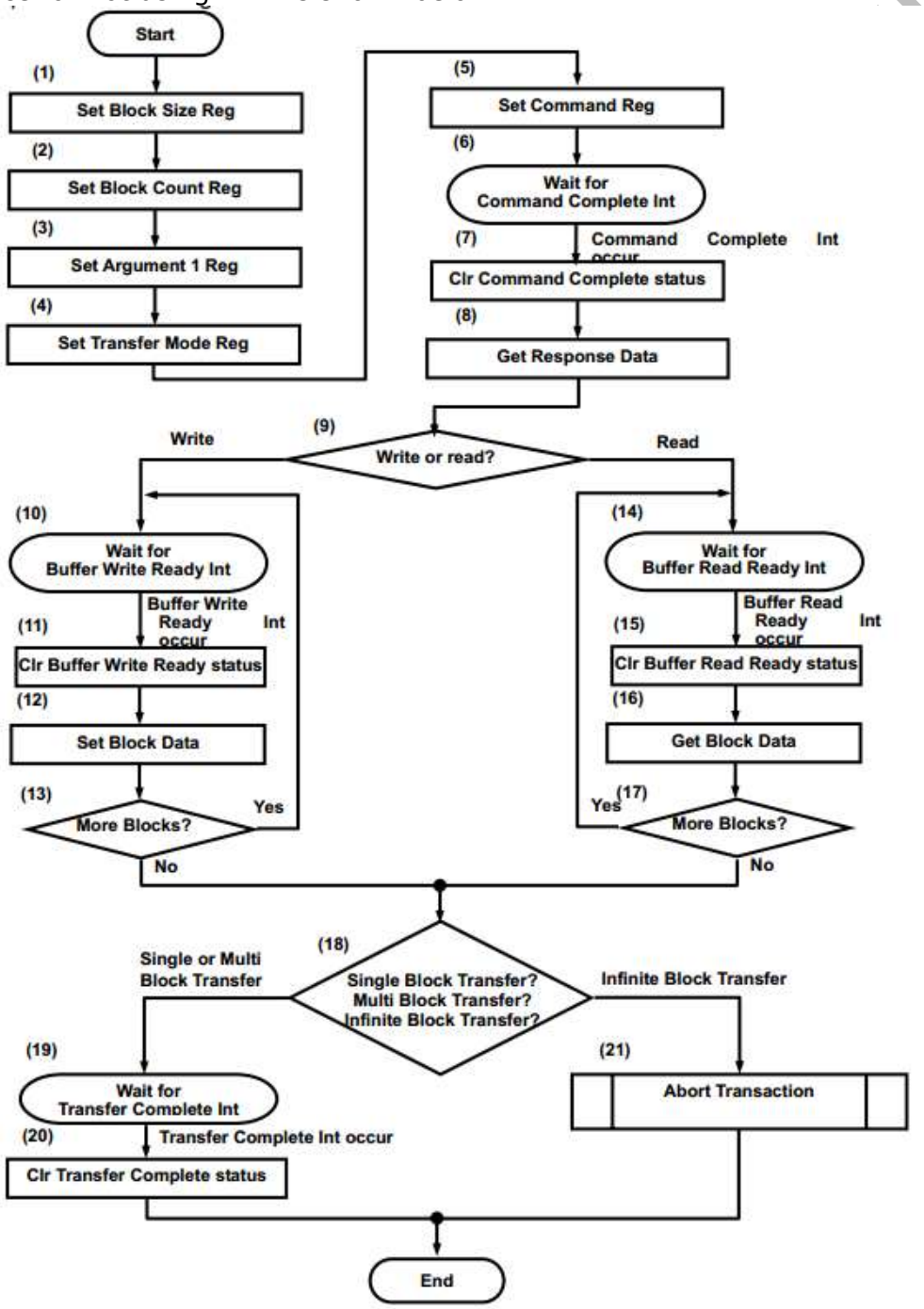


Fig. 16-7 Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)

1) Set the value corresponding to the executed data byte length of one block to Block Size

register.
2) Set the value corresponding to the executed data block count to Block Count register.
3) Set the value corresponding to the issued command to Argument register.
4) Set the value to Multi / Single Block Select and Block Count Enable. Set the value corresponding to the issued command to Data Transfer Direction, Auto CMD12 Enable and DMA Enable.
5) Set the value corresponding to the issued command to Command register. When writing the upper byte of Command register, SD command is issued.
6) Then, wait for the Command Complete Interrupt.
7) Write 1 to the Command Complete in the Normal Interrupt Status register for clearing this bit.
8) Read Response register and get necessary information of the issued command.
9) In the case where this sequence is for write to a card, go to step (10). In case of read from a card, go to step (14).
10) Then wait for Buffer Write Ready Interrupt.
11) Write 1 to the Buffer Write Ready in the Normal Interrupt Status register for clearing this bit.
12) Write block data (in according to the number of bytes specified at the step (1)) to Buffer Data Port register.
13) Repeat until all blocks are sent and then go to step (18).
14) Then wait for the Buffer Read Ready Interrupt.
15) Write 1 to the Buffer Read Ready in the Normal Interrupt Status register for clearing this bit.
16) Read block data (in according to the number of bytes specified at the step (1)) from the Buffer Data Port register.
17) Repeat until all blocks are received and then go to step (18).
18) If this sequence is for Single or Multiple Block Transfer, go to step (19). In case of Infinite Block Transfer, go to step (21).
19) Wait for Transfer Complete Interrupt.
20) Write 1 to the Transfer Complete in the Normal Interrupt Status register for clearing this bit.
21) Perform the sequence for Abort Transaction.

Note:

- *Step (1) and Step (2) can be executed at same time. Step (4) and Step (5) can be executed at same time.*
- *On receiving the Buffer Write Ready interrupt the CPU Processor will act as a master and start transferring the data via Buffer data port register (fifo_1). Transmitter starts sending the data in SD bus when a block of data is ready in fifo_1. While transmitting the data in sd bus the buffer write ready interrupt is sent to the CPU Processor for the second block of data. The CPU Processor will act as a master and start sending the second block of data via Buffer data port register to fifo_2. Buffer write ready interrupt will be asserted only when a fifo is empty to receive a block of data.*
- *Buffer Read Ready interrupt is asserted whenever a block of data is ready in one of the fifo's. On receiving the Buffer Read Ready interrupt the CPU Processor will act as a master and start reading the data via Buffer data port register (fifo_1). Receiver starts reading the data from SD bus only when a fifo is empty to receive a block of data. When both the fifo's are full the host controller will stop the data coming from the card through read wait mechanism (if card supports read wait) or through clock stopping.*

2. Using SDMA

The sequence for using SDMA is shown below.

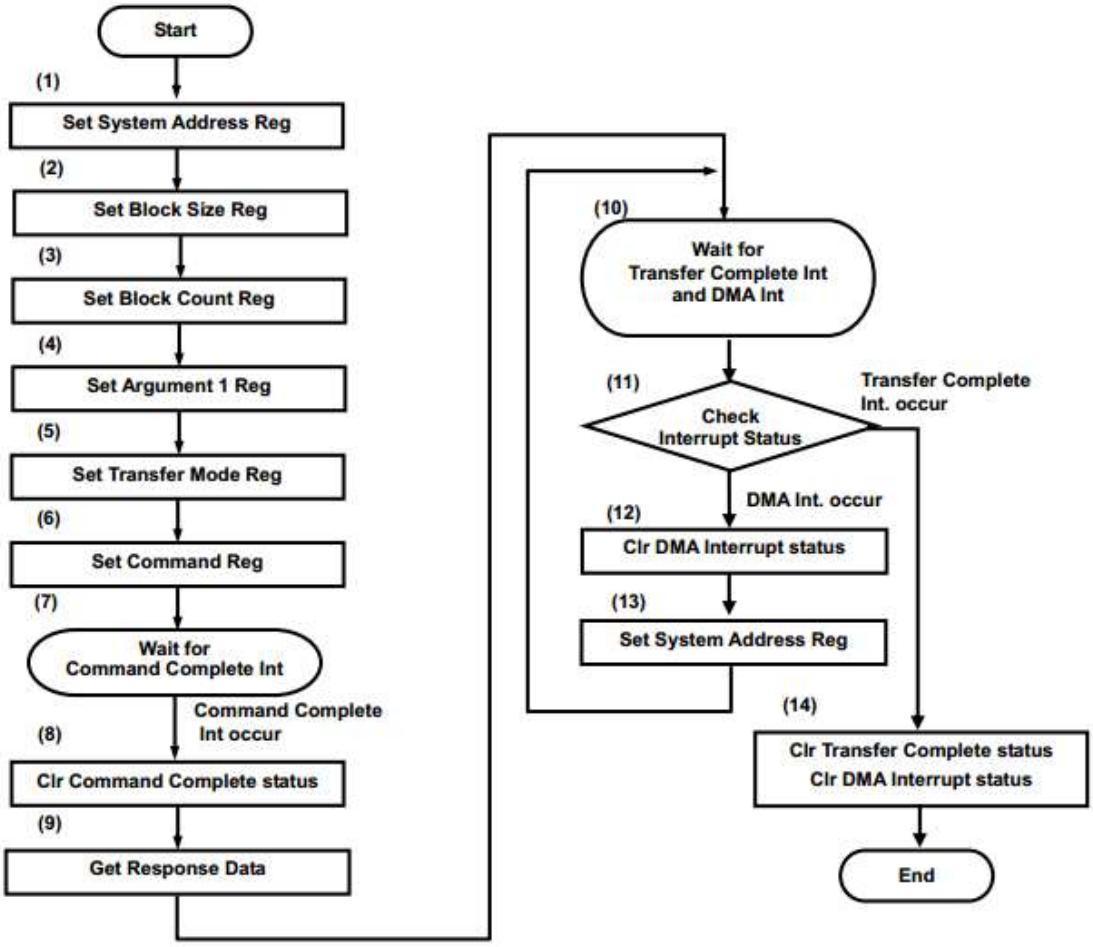


Fig. 16-8 Transaction Control with Data Transfer Using DAT Line Sequence (Using SDMA)

1) Set the system address for DMA in the System Address register.
2) Set the value corresponding to the executed data byte length of one block in the Block Size register.
3) Set the value corresponding to the executed data block count in the Block Count register.
4) Set the value corresponding to the issued command to Argument register.
5) Set the value to Multi / Single Block Select and Block Count Enable. Set the value corresponding to the issued command to Data Transfer Direction, Auto CMD12 Enable and DMA Enable.
6) Set the value corresponding to the issued command to Command register. When writing the upper byte of Command register, SD command is issued.
7) Then wait for the Command Complete Interrupt.
8) Write 1 to the Command Complete in the Normal Interrupt Status register to clear this bit.
9) Read Response register and get necessary information of the issued command.
10) Wait for the Transfer Complete Interrupt and DMA Interrupt.
11) If Transfer Complete is set 1, go to Step (14) else if DMA Interrupt is set to 1, go to Step (12).Transfer Complete is higher priority than DMA Interrupt.
12) Write 1 to the DMA Interrupt in the Normal Interrupt Status register to clear this bit.
13) Set the next system address of the next data position to the System Address register and go to Step (10).
14) Write 1 to the Transfer Complete and DMA Interrupt in the Normal Interrupt Status register to clear this bit.

Note:

- *Step (2) and Step (3) can be executed simultaneously. Step (5) and Step (6) can also be executed simultaneously*
- *DMA read transfer: On receiving the response end bit from the card for the write command (data flowing from Host to Card) the SD Host controller will act as the master and request the System/Host bus. After receiving the grant the host controller will start reading a block of data from the system*

memory and fills the first fifo. Whenever a block of data is ready the transmitter will start sending the data in SD bus. While transmitting the data in SD bus the host controller requests the bus to fill the second block in second fifo. Ping Pong fifo's are used to increase the throughput. Similarly the host controller reads a block of data from the system memory whenever a fifo is empty. This will continue till all the blocks are read from the System memory. Transfer complete Interrupt will be set only after transferring all the blocks of data to the card.

- *DMA write transfer: The block of data received from the Card (data flowing from Card to Host) is stored in first half of the fifo. Whenever a block of data is ready the SD Host controller will act as the master and request the System/Host bus. After receiving the grant the host controller will start writing a block of data into the system memory from the first fifo. While transmitting the data into System memory the host controller will receive the second block of data and store in second fifo. Similarly the host controller writes a block of data into the system memory whenever data is ready. This will continue till all the blocks are transferred to the System memory. Transfer complete Interrupt will be set only after transferring all the blocks of data to the System memory.*
- *Host controller will receive a block of data from the card only when it has room to store a block of data in fifo. When both the fifo's are full the host controller will stop the data coming from the card through read wait mechanism (if card supports read wait) or through clock stopping.*

3. Using ADMA

The sequence for using ADMA is shown below.

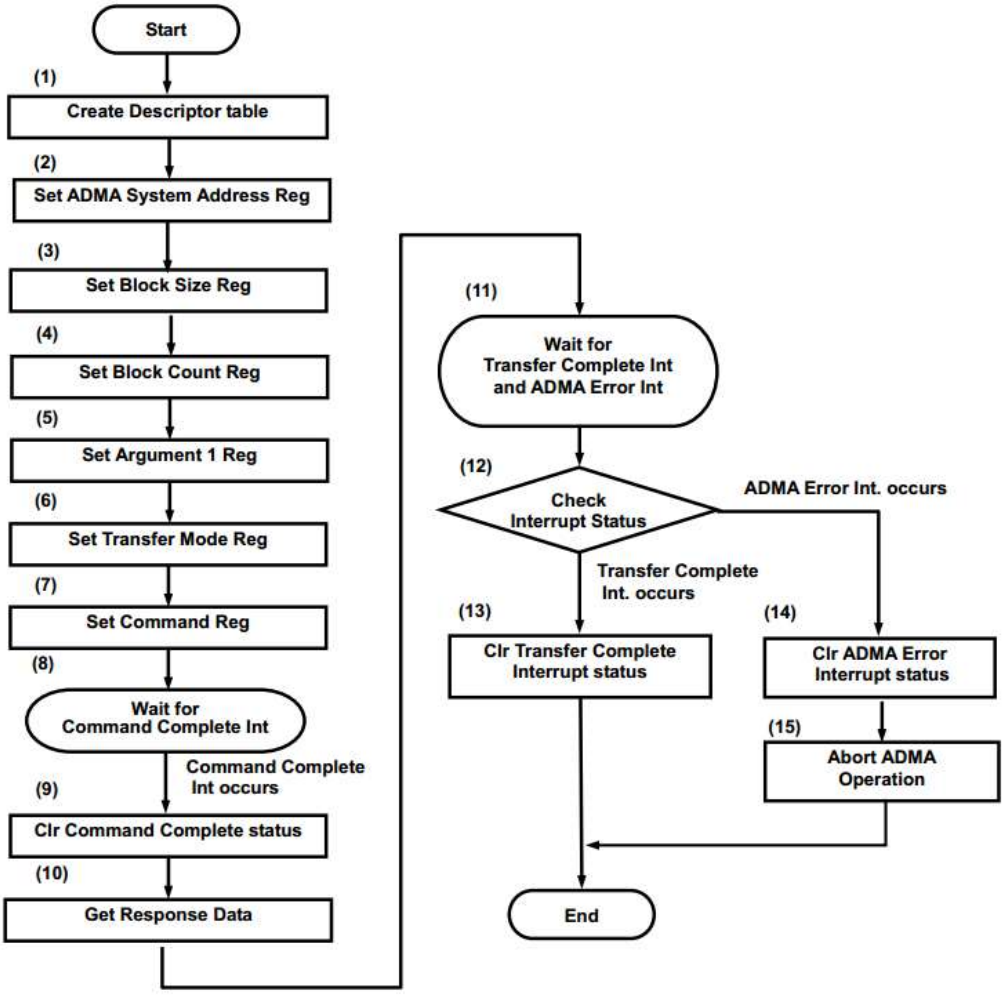


Fig. 16-9 Transaction Control with Data Transfer Using DAT Line Sequence (Using ADMA)

1) Create Descriptor table for ADMA in the system memory.
2) Set the Descriptor address for ADMA in the ADMA System Address register.
3) Set the value corresponding to the executed data byte length of one block in the Block Size register.
4) Set the value corresponding to the executed data block count in the Block Count register. If the Block Count Enable in the Transfer Mode register is set to 1, total data length can be designated by the Block Count register and the Descriptor Table. These two parameters shall indicate same data length. However, transfer length is limited by the Block Count register. If the Block Count Enable in the Transfer Mode register is set to 0, total data length is designated by not Block Count register but the Descriptor Table. In this case, ADMA reads more data than length programmed in descriptor from

SD card. Too much read operation is aborted asynchronously and extra read data is discarded when the ADMA is completed.

5) Set the argument value to the Argument 1 register.
6) Set the value to the Transfer Mode register. The host driver determines Multi / Single Block Select, Block Count Enable, Data Transfer Direction, Auto CMD12 Enable and DMA Enable.
7) Set the value to the Command register.

Note: When writing to the upper byte [3] of the Command register, the SD command is issued and DMA is started.

8) Then wait for the Command Complete Interrupt.
9) Write 1 to the Command Complete in the Normal Interrupt Status register to clear this bit.
10) Read Response register and get necessary information of the issued command.
11) Wait for the Transfer Complete Interrupt and ADMA Error Interrupt.
12) If Transfer Complete is set 1, go to Step (13) else if ADMA Error Interrupt is set to 1, go to Step (14).
13) Write 1 to the Transfer Complete Status in the Normal Interrupt Status register to clear this bit.
14) Write 1 to the ADMA Error Interrupt Status in the Error Interrupt Status register to clear this bit.
15) Abort ADMA operation. SD card operation should be stopped by issuing abort command. If necessary, the host driver checks ADMA Error Status register to detect why ADMA error is generated.

Note: Step (3) and Step (4) can be executed simultaneously. Step (6) and Step (7) can also be executed simultaneously.

16.7.4 Advanced DMA(ADMA2)

1. Block Diagram of ADMA2

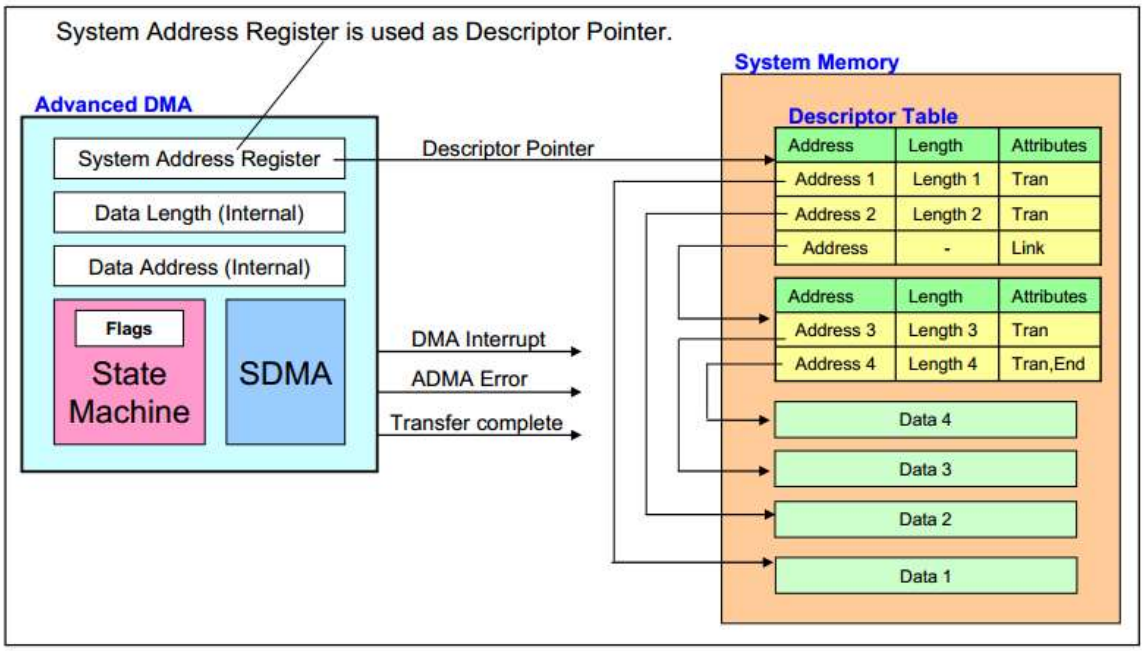


Fig. 16-10 Block Diagram of ADMA2

Figure above shows block diagram of ADMA2. The Descriptor Table is created in system memory by the Host Driver. 32-bit Address Descriptor Table is used for the system with 32-bit addressing. Each descriptor line (one executable unit) consists with address, length and attribute field. The attribute specifies operation of the descriptor line. ADMA2 includes SDMA, State Machine and Registers circuits. ADMA2 does not use 2-bit *SDMA System Address* Register (offset 0) but uses the 64-bit *Advanced DMA System Address* register (offset 058h) for descriptor pointer. Writing Command register triggers off ADMA2 transfer. ADMA2 fetches one descriptor line and execute it. This procedure is repeated until end of descriptor is found (End=1 in attribute).

2. Data Address and Data Length Requirements

There are 3 requirements to program the descriptor.

- The minimum unit of address is 4 bytes.
- The maximum data length of each descriptor line is less than 64KB.
- Total Length = Length 1 + Length 2 + Length 3 + ... + Length n = multiple of Block Size.

If total length of a descriptor were not multiple of block size, ADMA2 transfer might not be terminated. In this case, the transfer should be aborted by data timeout.
Block Count register limits the maximum of 65535 blocks transfer. If ADMA2 operation is less than or equal 65535 blocks transfer, Block Count register can be used. In this case, total length of Descriptor Table shall be equivalent to multiply block size and block count. If ADMA2 operation is more than 65535 blocks transfer, Block Count Register shall be disabled by setting 0 to Block Count Enable in the Transfer Mode Register. In this case, length of data transfer is not designated by block count but Descriptor Table. Therefore, the timing of detecting the last block on SD bus may be different and it affects the control of Read Transfer Active, Write Transfer Active and DAT line Active in the Present State register. In case of read operation, several blocks may be read more than required. The Host Driver shall ignore out of range error if the read operation is for the last block of memory area.

3. Descriptor Table

Address Field	Length	Reserved	Attribute					
63 32	31 16	15 06	05	04	03	02	01	00
32-bit Address	16-bit Length	000000	Act2	Act1	0	Int	End	Valid

Valid	Valid=1 indicates this line of descriptor is effective. If Valid=0 generate ADMA Error Interrupt and stop ADMA to prevent runaway.
End	End=1 indicates end of descriptor. The Transfer Complete Interrupt is generated when the operation of the descriptor line is completed.
Int	INT=1 generates DMA Interrupt when the operation of the descriptor line is completed.

Act2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Do not execute current line and go to next line.
0	1	rsv	reserved	(Same as Nop. Do not execute current line and go to next line.)
1	0	Tran	Transfer Data	Transfer data of one descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Length Field	Value of Length
0000h	65536 bytes
0001h	1 byte
0002h	2 bytes
.....	.....
FFFFh	65535 bytes

Fig. 16-11 32-bit Address Descriptor Table

16.7.5 Tuning Procedure

1. Sampling clock tuning

The SD bus can be operating in high clock frequency mode and then the data window from the card on CMD and DAT lines gets smaller. The position of the data window will vary depending on the card and host system implementation. The Host Controller shall support a tuning circuit by executing the tuning procedure and adjusting the sampling clock. Execute Tuning and Sampling Clock Select in the Host Control 2 register are used to control the tuning circuit.

2. Clock Tuning procedure

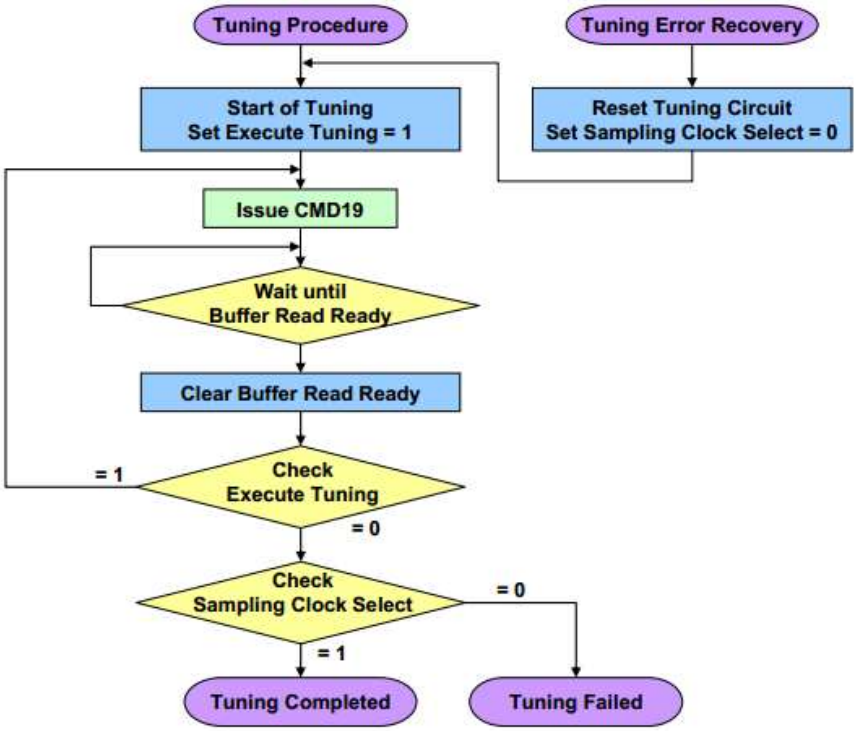


Fig. 16-12 Sampling Clock Tuning Procedure

The above figure defines sampling clock tuning procedure supported by Host Controller. In default, for lower frequency operation, fixed sampling clock is used to receive signals on CMD and DAT. Before using SDR104, sampling clock tuning is required. Start of sampling clock tuning is requested by setting Execute Tuning to 1 and Sampling Clock Select to 0. Host driver issue CMD19(SD)/CDM21(MMC) repeatedly until the host controller resets Execute Tuning to 0. Host Controller resets Execute Tuning to 0 when tuning is completed or tuning is not completed within 40 times. Host Driver can abort this loop by 40 times CMD19/CMD21 issue or 150ms time-out. If tuning is completed successfully, Host Controller set Sampling Clock Select to 1 and this means the Host Controller start to use tuned sampling clock. If tuning is failed, Host Controller keeps Sampling Clock Select to 0. By writing Sampling Clock Select to 0, sampling clock is switched from tuned sampling clock to fixed sampling clock. Re-tuning time would be smaller than the first tuning time. CMD19/CMD21 response errors are not indicated while tuning is performed.

The clock tuning tap delay values are selected using Variable sampling point detection. Fixed tap delay value is used for fixed tuning clock method.

16.7.6 Abort Transaction

An abort transaction is performed by issuing CMD12 for a SD memory card and by issuing CMD52 for a SDIO card. There are two cases where the Host Driver needs to do an Abort Transaction. The first case is when the Host Driver stops Infinite Block Transfers. The second case is when the Host Driver stops transfers while a Multiple Block Transfer is executing.

There are two ways to issue an Abort Command. The first is an asynchronous abort. The second is a synchronous abort. In an asynchronous abort sequence, the Host Driver can issue an Abort Command at anytime unless Command Inhibit (CMD) in the Present State register is set to 1. In a synchronous abort, the Host Driver shall issue an Abort Command after the data transfer stopped by using Stop At Block Gap Request in the Block Gap Control register.

1. Asynchronous Abort

The sequence for Asynchronous Abort is shown below.

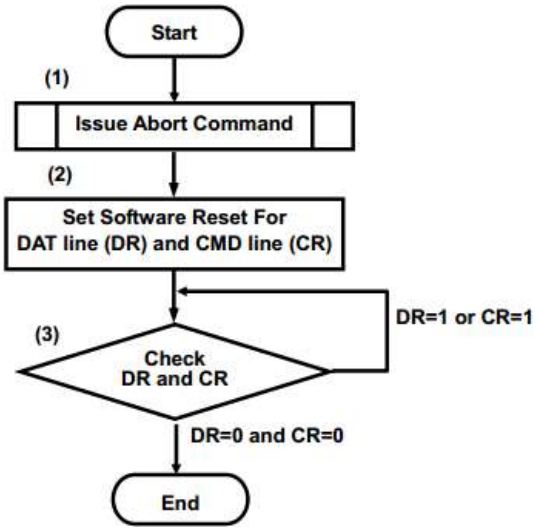


Fig. 16-13 Asynchronous Abort Sequence

1) Issue Abort Command.
2) Set both Software Reset For DAT Line and Software Reset For CMD Line to 1 in the Software Reset register to do software reset.
3) Check Software Reset For DAT Line and Software Reset For CMD Line in the Software Reset register. If both Software Reset For DAT Line and Software Reset For CMD Line are 0, go to "End". If either Software Reset For DAT Line or Software Reset For CMD Line is 1, go to step (3).

2. Synchronous Abort

The sequence for Synchronous Abort is shown below.

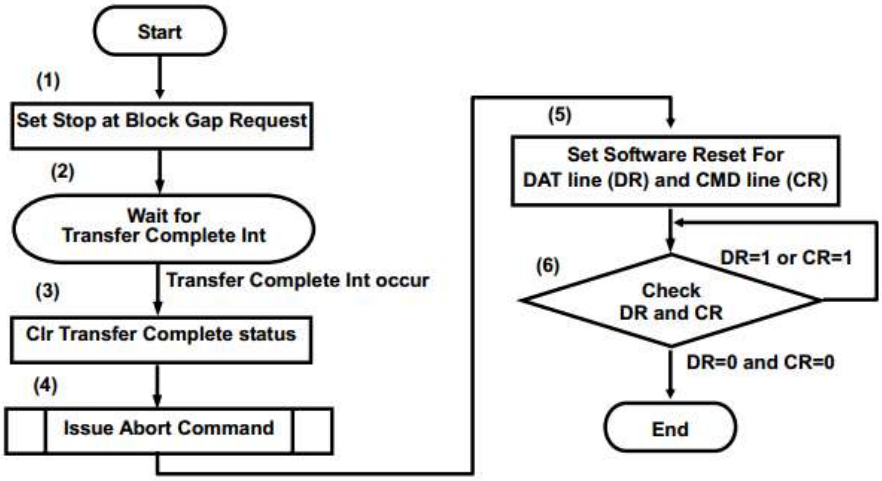


Fig. 16-14 Synchronous Abort Sequence

1) Set the Stop At Block Gap Request in the Block Gap Control register to 1 to stop SD transactions.
2) Wait for the Transfer Complete Interrupt.
3) Set the Transfer Complete to 1 in the Normal Interrupt Status register to clear this bit.
4) Issue the Abort Command.
5) Set both Software Reset For DAT Line and Software Reset For CMD Line to 1 in the Software Reset register to do software reset.
6) Check both Software Reset For DAT Line and Software Reset For CMD Line in the Software Reset register. If both Software Reset For DAT Line and Software Reset For CMD Line are 0, go to 'End'. If either Software Reset For DAT Line or Software Reset For CMD Line is 1, go to step (6).

16.7.7 Error Recovery

1. Error Interrupt Recovery

The sequence for Error Interrupt Recovery is shown below.

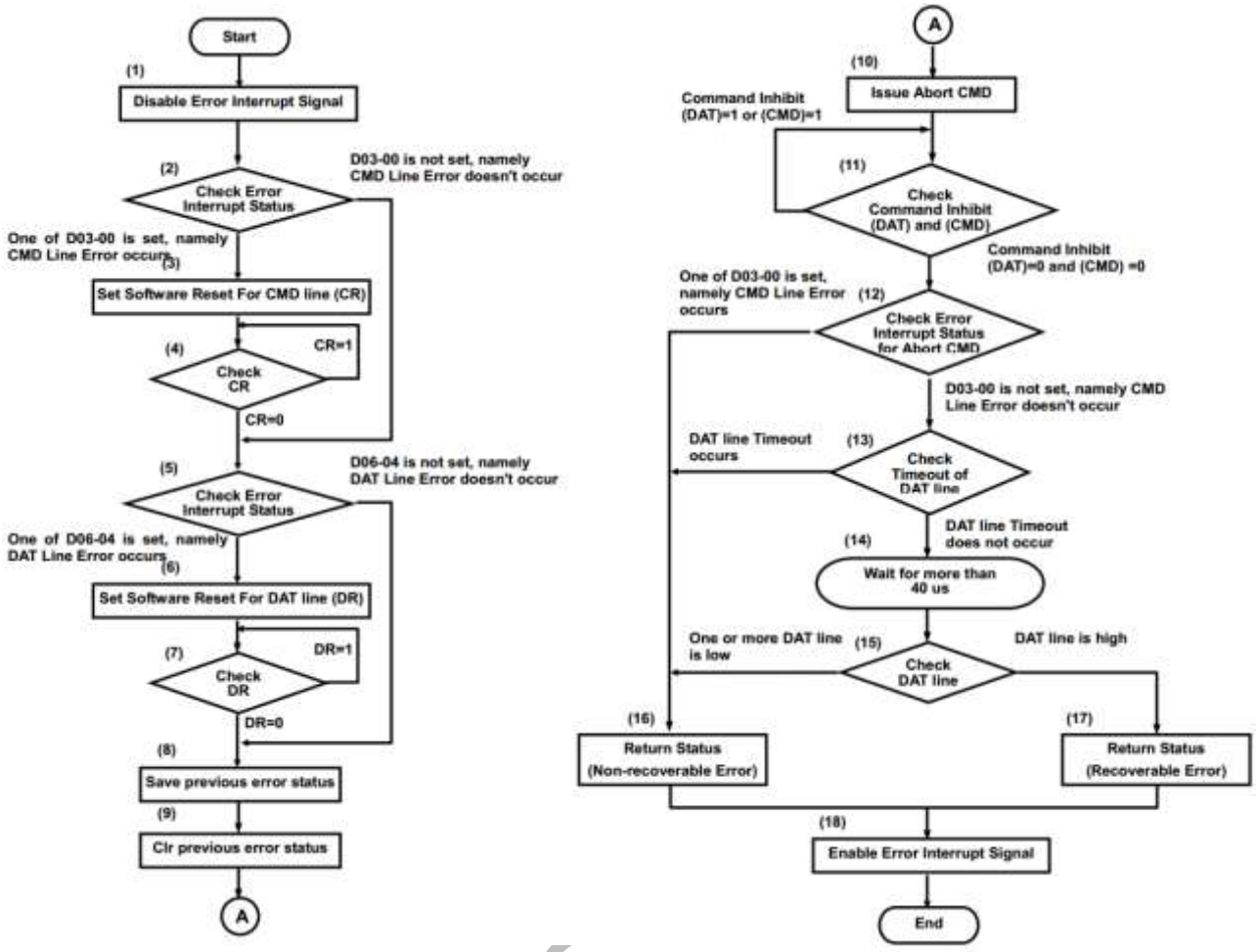


Fig. 16-15 Error Interrupt Recovery Sequence

1) Disable the Error Interrupt Signal.
2) Check bits D03-00 in the Error Interrupt Status register. If one of these bits (D03-00) is set to 1, go to step (3). If none are set to 1 (all are 0), go to step (5).
3) Set Software Reset For CMD Line to 1 in the Software Reset register for software reset of the CMD line.
4) Check Software Reset For CMD Line in the Software Reset register. If Software Reset For CMD Line is 0, go to step (5). If it is 1, go to step (4).
5) Check bits D06-04 in the Error Interrupt Status register. If one of these bits (D06-04) is set to 1, go to step (6). If none are set to 1 (all are 0), go to step (8).
6) Set Software Reset For DAT Line to 1 in the Software Reset register for software reset of the DAT line.
7) Check Software Reset For DAT Line in the Software Reset register. If Software Reset For DAT Line is 0, go to step (8). If it is 1, go to step (7).
8) Save previous error status.
9) Clear previous error status with setting them to 1.
10) Issue Abort Command.
11) Check Command Inhibit (DAT) and Command Inhibit (CMD) in the Present State register. Repeat this step until both Command Inhibit (DAT) and Command Inhibit (CMD) are set to 0.
12) Check bits D03-00 in the Error Interrupt Status register for Abort Command. If one of these bits is set to 1, go to step (16). If none of these bits are set to 1 (all are 0), go to step (13).
13) Check Data Timeout Error in the Error Interrupt Status register. If this bit is set to 1, go to step (16). If it is 0, go to step (14).
14) Wait for more than 40 us.
15) By monitoring the DAT [3:0] Line Signal Level in the Present State register, judge whether the level of the DAT line is low or not. If one or more DAT lines are low, go to

step (16). If the DAT lines are high, go to step (17).
16) Return Status of "Non-recoverable Error".
17) Return Status of "Recoverable Error".
18) Enable the Error Interrupt Signal.

2. Auto CMD12 Error Recovery

The sequence for Auto CMD12 Error Recovery is shown below. Following four cases A-D shall be covered.
A: An error occurred in CMD_wo_DAT, but not in the SD memory transfer.
B: An error occurred in CMD_wo_DAT, and also occurred in the SD memory transfer.
C: An error did not occur in CMD_wo_DAT, but an error occurred in the SD memory transfer.
D: CMD_wo_DAT was not issued, and an error occurred in the SD memory transfer.

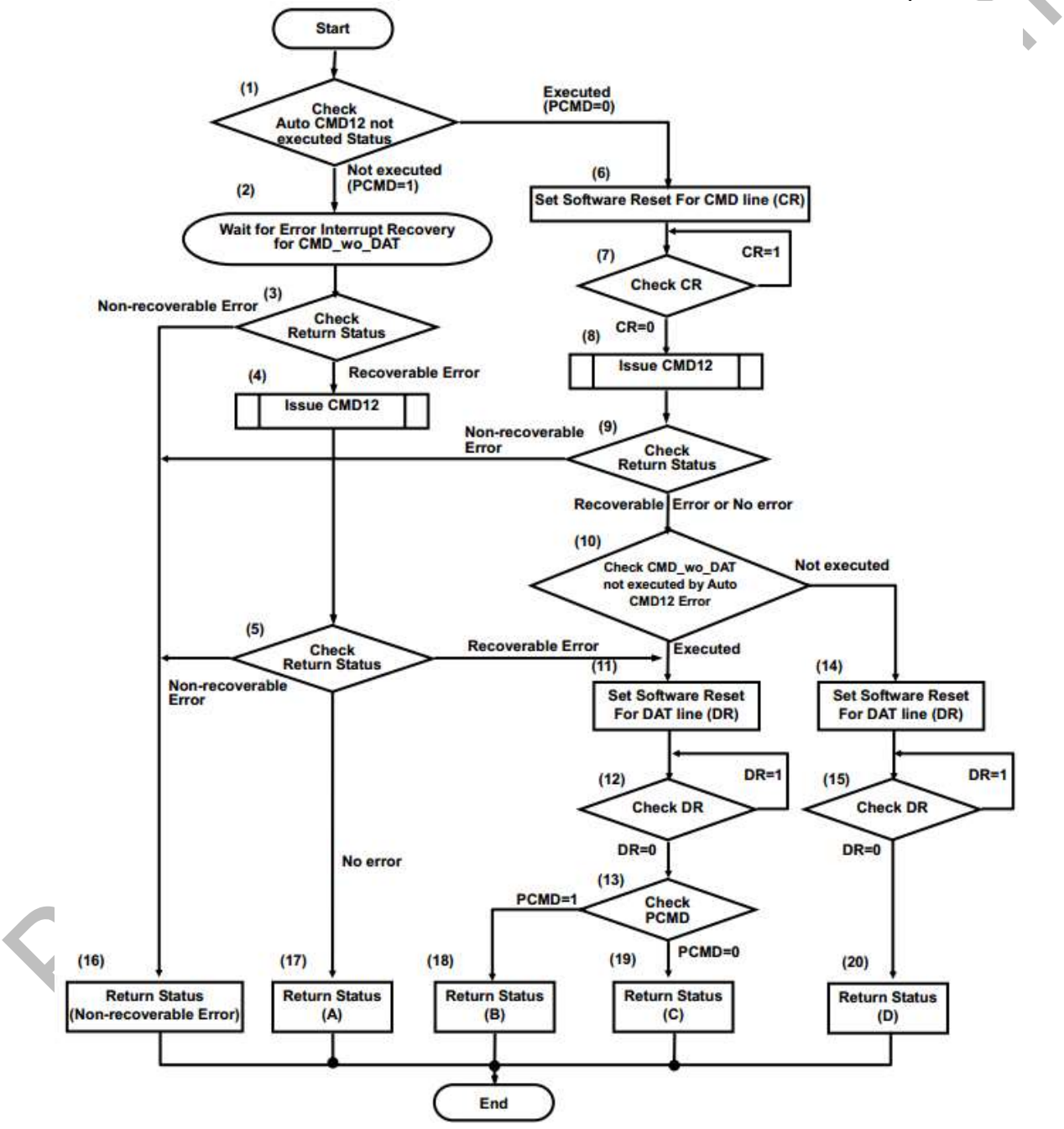


Fig. 16-16 Auto CMD12 Error Recovery Sequence

1) Check Auto CMD12 Not Executed in the Auto CMD Error Status register. If this bit is set to 1, go to step (2). If this bit is set to 0, go to step (6). In addition, the Host Driver shall define PCMD flag, which changes to 1 if Auto CMD12 Not Executed is set to 1.
2) Wait for Error Interrupt Recovery for CMD_wo_DAT.
3) Check "Return Status". In the case of "Non-recoverable Error", go to step (16). In the

case of "Recoverable Error", go to step (4).
4) Issue CMD12.
5) (5) If the CMD line errors occur for the CMD12 (One of D03-00 is set in the Error Interrupt Status register), "Return Status" is "Non-recoverable Error" and go to step (16). If not CMD line error and busy timeout error occur (D04 is set in the Error Interrupt Status register), "Return Status" is "Recoverable Error" and go to step (11). Otherwise, "Return Status" is "No error" and go to step (17).
6) Set Software Reset For CMD Line to 1 in the Software Reset register for software reset of the CMD line.
7) Check Software Reset For CMD Line in the Software Reset register. If Software Reset For CMD Line is 0, go to step (8). If it is 1, go to step (7).
8) Issue CMD12 according to Section 3.7.1. Acceptance of CMD12 depends on the state of the card. CMD12 may make the card to return to tran state. If the card is already in tran state, the card does not response to CMD12.
9) Check "Return Status" for CMD12. If "Return Status" returns "Non-recoverable Error", go to step (16). In the case of "Recoverable Error" or "No error", go to step (10).
10) Check the Command Not Issued By Auto CMD12 Error in the Auto CMD Error Status register. If this bit is 0, go to step (11). If it is 1, go to step (14).
11) Set Software Reset For DAT Line to 1 in the Software Reset register for software reset of the DAT line.
12) Check Software Reset For DAT Line in the Software Reset register. If Software Reset For DAT Line is 0, go to step (13). If it is 1, go to step (12).
13) Check the PCMD flag. If PCMD is 1, go to step (18). If it is 0, go to step (19).
14) Set Software Reset For DAT Line to 1 in the Software Reset register for software reset of the DAT line.
15) Check Software Reset For DAT Line in the Software Reset register. If Software Reset For DAT Line is 0, go to step (20). If it is 1, go to step (15).
16) Return Status of "Non-recoverable Error".
17) Return Status that an error has occurred in CMD_wo_DAT, but not in the SD memory transfer.
18) Return Status that an error has occurred in both CMD_wo_DAT, and the SD memory transfer.
19) Return Status that an error has not occurred in CMD_wo_DAT, but has occurred in the SD memory transfer.
20) Return Status that CMD_wo_DAT has not been issued, and an error has occurred in the SD memory transfer.

16.7.8 Suspend/Resume

If a SD card supports suspend and resume functionality, then the Host Controller can initiate suspend and resume. It is necessary for both the Host Controller and the SD card to support the function of "Read Wait". ADMA operation does not support this function.

1. Suspend Sequence

The sequence for suspend is shown below.

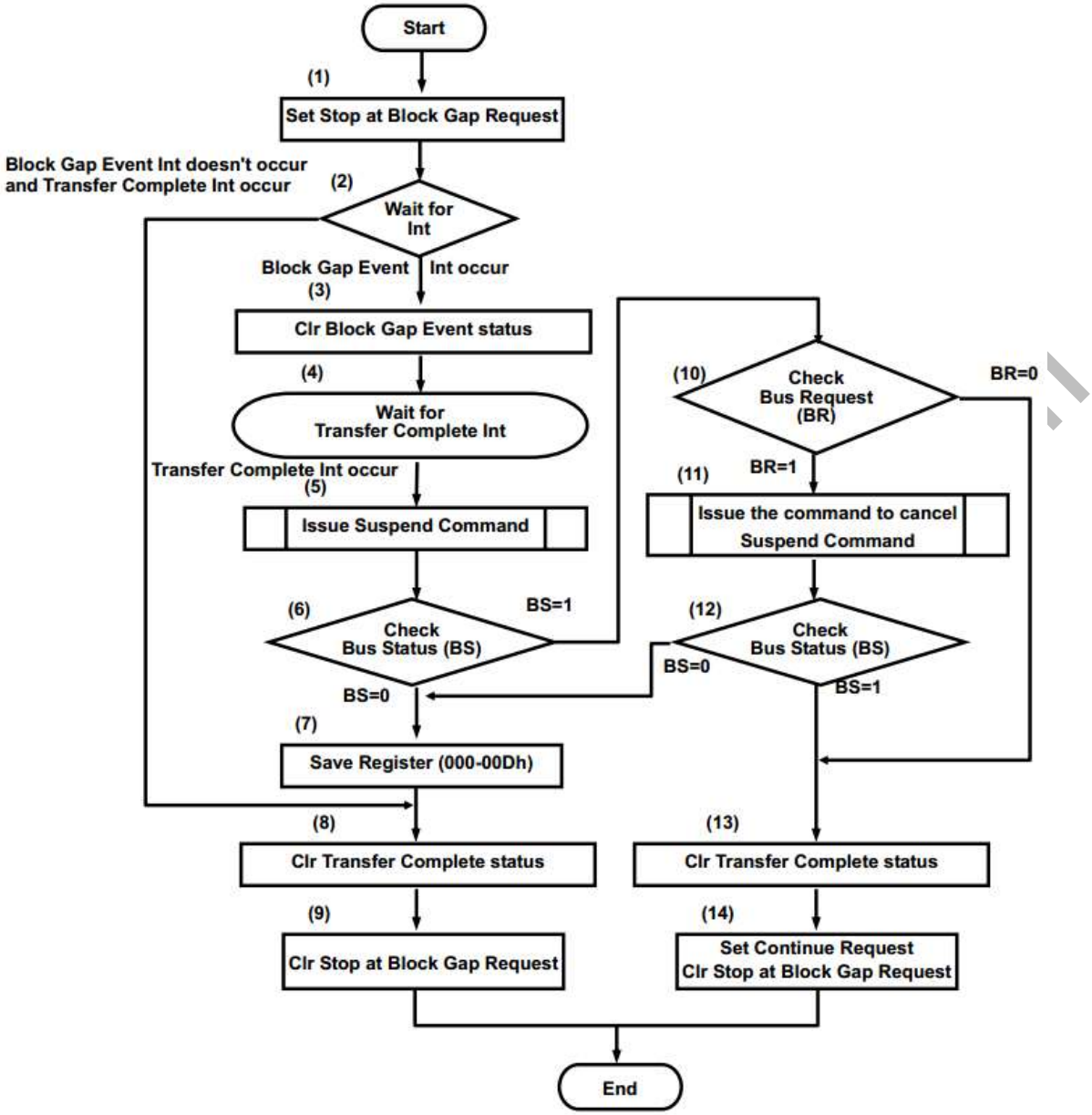


Fig. 16-17 The Sequence for Suspend

1) Set Stop At Block Gap Request to 1 in the Block Gap Control register to stop the SD transaction.
2) Wait for an Interrupt. If Block Gap Event is set to 0 and Transfer Complete is set to 1 in the Normal Interrupt Status register, go to step (8). If Block Gap Event is set to 1, go to step (3).
3) Set Block Gap Event to 1 in the Normal Interrupt Status register to clear this bit.
4) Wait for the Transfer Complete Interrupt.
5) Issue the Suspend Command.
6) Check the BS value of the response data. If BS is 0, go to step (7). If BS is 1, go to step (10).
7) Save the register (000h-00Dh).
8) Set Transfer Complete to 1 in the Normal Interrupt Status register to clear this bit.
9) Set Stop At Block Gap Request to 0 in the Block Gap Control register to clear this bit.
10) Check the BR value of the response data. If BR is 1, go to step (11). If BR is 0, go to step (13).
11) Issues the command to cancel the previous suspend command in accordance with Transaction Control without Data Transfer Using DAT Line.
12) Check the BS value of the response data. If BS is 0, go to step (7). If BS is 1, go to step (13).
13) Set Transfer Complete to 1 in the Normal Interrupt Status register to clear this bit.

14) Set Continue Request to 1 in the Block Gap Control register to continue the transaction. At the same time, write 0 to Stop At Block Gap Request to clear this bit.

2. Resume Sequence

The sequence for resume is shown below.

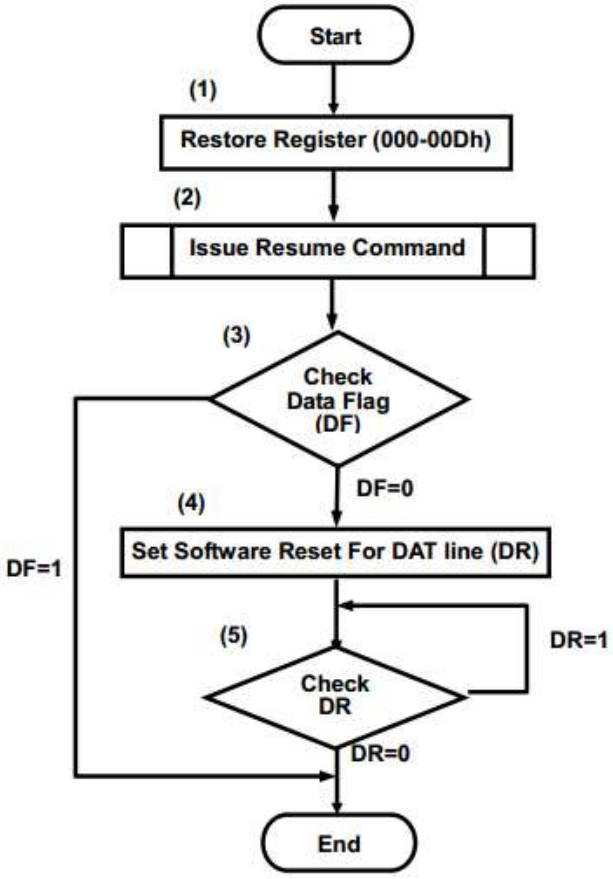


Fig. 16-18 The Sequence for Resume

1) Restore the register (000h-00Dh).
2) Issue the Resume Command.
3) Check the DF value of the response data. If DF is 0, go to step (4). If DF is 1, go to 'End'.
4) Set Software Reset For DAT Line to 1 in the Software Reset register for software reset of the DAT line.
5) Check Software Reset For DAT Line in the Software Reset register. If Software Reset For DAT Line is 0, go to 'End'. If it is 1, go to step (5).

3. Read Transaction Wait / Continue Timing

Note:

- *Read Wait, DAT Line Active and Read Transfer Active shall be set and cleared by the Host Controller.*
- *Stop At Block Gap Request shall be set and cleared by the Host Driver.*
- *Continue Request shall be set by the Host Driver and be cleared by the Host Controller.*
- *Block Gap Event and Transfer Complete shall be set by the Host Controller and be cleared by the Host Driver.*

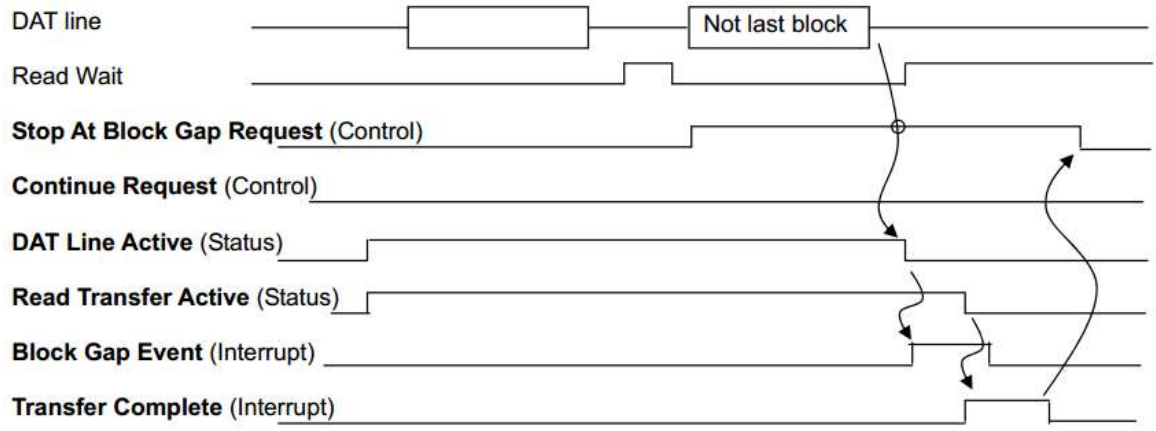


Fig. 16-19 Wait Read Transfer by Stop At Block Gap Request

The Host Controller can accept a Stop At Block Gap Request when all the following conditions are met.

- It is at the block gap.
- The Host Controller can assert read wait or it is already asserted.
- Read Wait Control is set to 1.

After accepting the Stop At Block Gap Request,

1) Clear DAT Line Active status and generate the Block Gap Event Interrupt
2) After all valid data has been read (No valid read data remains in the Host Controller), clear the Read Transfer Active status and generate the Transfer Complete Interrupt.
3) After accepting Transfer Complete Interrupt, clear the Stop At Block Gap Request

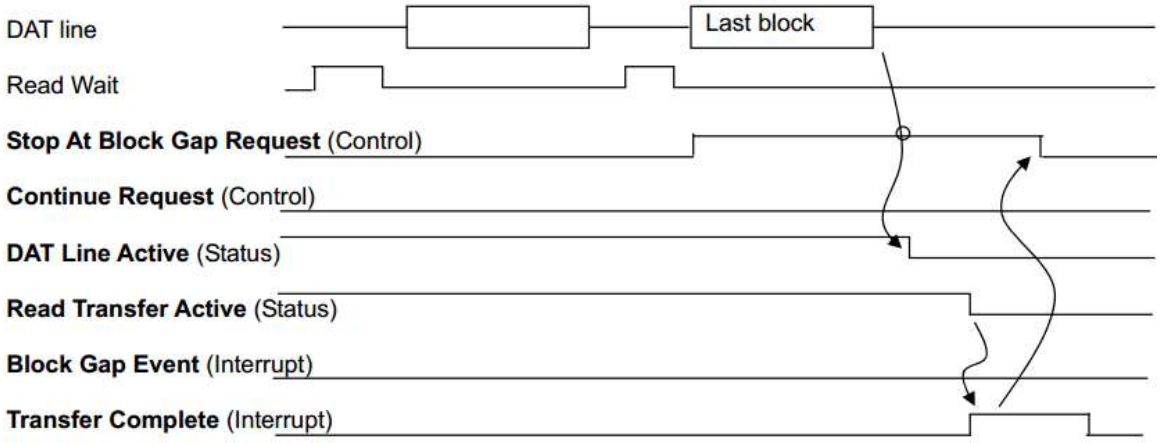


Fig. 16-20 Stop At Block Gap Request is Not Accepted at the Last Block of the Read Transfer

If the Stop At Block Gap Request is set to 1 during the last block transfer, the Host Controller shall not accept the Stop At Block Gap Request and stops the transaction normally. The Block Gap Event Interrupt is not generated. When the Transfer Complete Interrupt is generated, and if the Block Gap Event status is not set to 1, the driver shall clear the Stop At Block Gap Request.

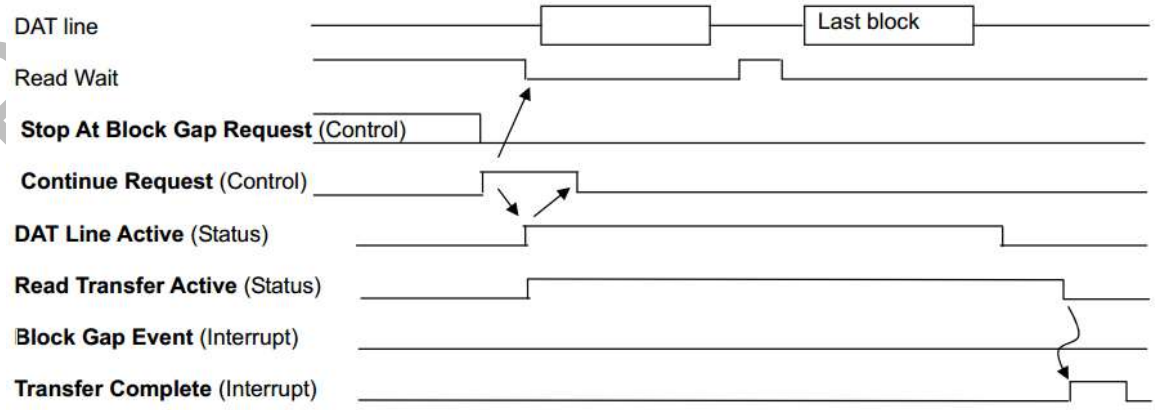


Fig. 16-21 Continue Read Transfer by Continue Request

To restart a stopped data transfer, set the Continue Request to 1. (The Stop At Block Gap Request shall be set to 0.)

After accepting the Continue Request,
1) Release Read Wait (if the data block can accept the next data.)
2) Set the DAT Line Active status and the Read Transfer Active status
3) The Continue Request is automatically cleared by (2).

The end of the read transfer is specified by data length.
1) Clear the DAT Line Active status and do not generate the Block Gap Event Interrupt.
2) After all valid data has been read (No valid read data remains in the Host Controller), clear the Read Transfer Active status and generate the Transfer Complete Interrupt.

4. Write Transaction Wait / Continue Timing

Note:
- *DAT Line Active and Write Transfer Active shall be set and cleared by the Host Controller.*
- *Stop At Block Gap Request shall be set and cleared by the Host Driver.*
- *Continue Request shall be set by the Host Driver and be cleared by the Host Controller.*
- *Block Gap Event and Transfer Complete shall be set by the Host Controller and be cleared by the Host Driver.*

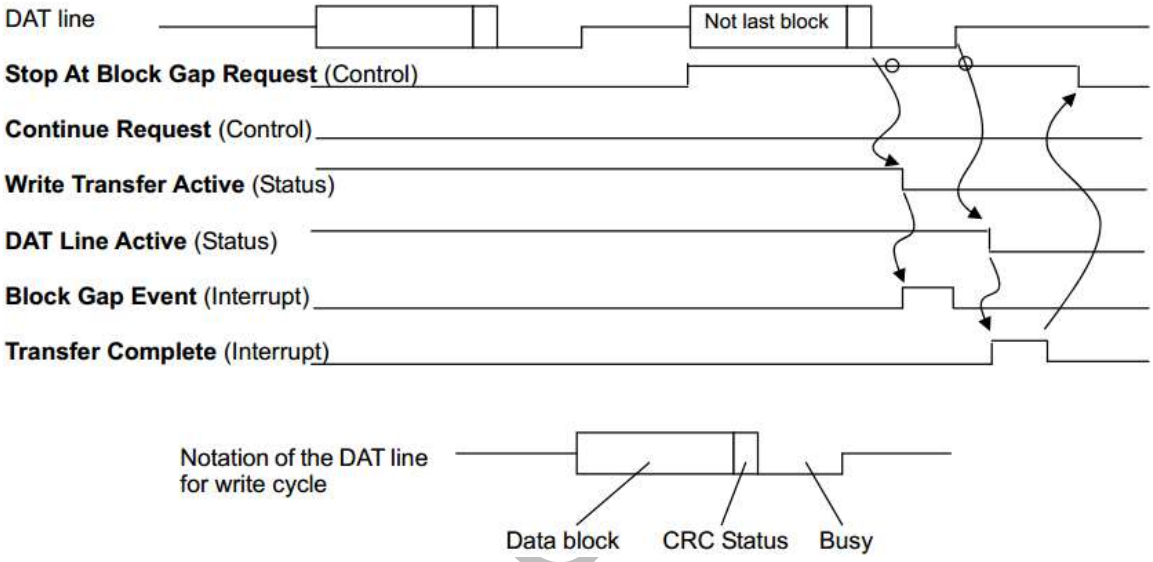


Fig. 16-22 Wait Write Transfer by Stop At Block Gap Request

The Host Controller can accept the Stop At Block Gap Request when matches all following conditions
- It is at the block gap.
- No valid write data remains in the Host Controller

After accepting the Stop At Block Gap Request
1) Clear the Write Transfer Active Status and generate the Block Gap Event Interrupt.
2) After the busy signal is released, clear the DAT Line Active status and generate the Transfer Complete Interrupt.
3) After accepting the Transfer Complete Interrupt, clear the Stop At Block Gap Request

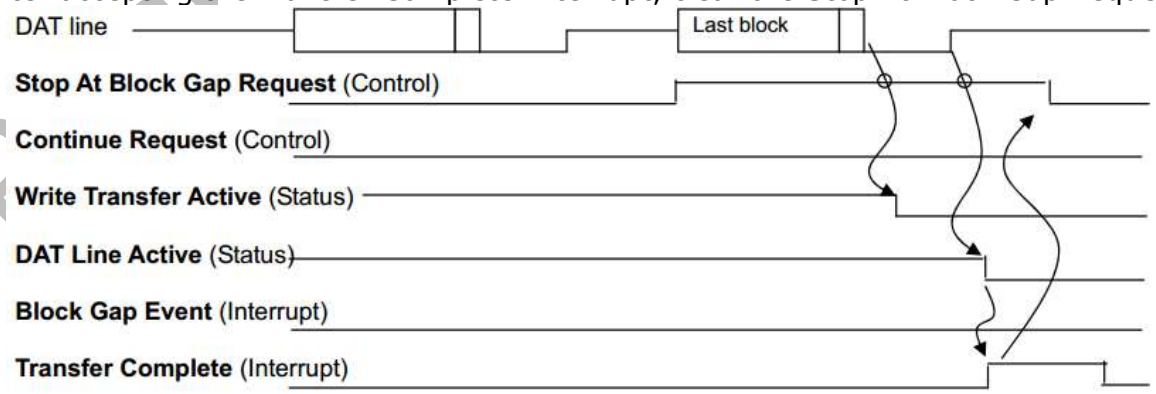


Fig. 16-23 Stop At Block Gap Request is Not Accepted at the Last Block of the Write Transfer

If the Stop At Block Gap Request is set to 1 during the last block transfer, the Host Controller shall not accept the Stop At Block Gap Request and terminates the transaction normally. The Block Gap Event Interrupt is not generated. When the Transfer Complete Interrupt is generated, and if the Block Gap Event Interrupt Status is not set to 1, the

driver shall clear the Stop At Block Gap Request.

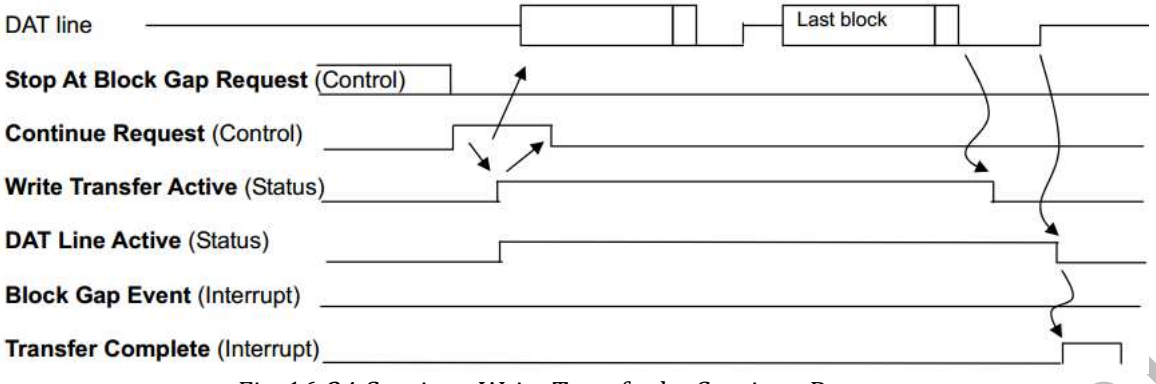


Fig. 16-24 Continue Write Transfer by Continue Request

To restart a stopped data transfer, set the Continue Request to 1. (Stop At Block Gap Request shall be set to 0.)

After accepting the Continue Request:

1) Set the DAT Line Active status and the Write Transfer Active Status
2) The Continue Request is automatically cleared by (1).

The end of transfer is specified by data length.

1) Clear the Write Transfer Active Status, and do not generate the Block Gap Event Interrupt
2) After the busy signal is released, clear the DAT Line Active status and generates the Transfer Complete Interrupt.

16.7.9 Relationship between Interrupt Control Registers

The Host Controller implements a number of interrupt sources. Interrupt sources can be enabled as interrupts as shown in following figure. If the interrupt source's corresponding bit in the Normal Interrupt Status Enable or Error Interrupt Status Enable register is 1 and the interrupt becomes active its active state is latched and made available to the Host Driver in the Normal Interrupt Status register or the Error Interrupt Status register. Interrupt Status shall be cleared when Interrupt Status Enable is cleared.

An interrupt source with its bit set in an interrupt status register shall assert a system interrupt signal if its corresponding bit is also set in the Normal Interrupt Signal Enable register or the Error Interrupt Signal Enable register. Once signaled, most interrupts are cleared by writing a 1 to the associated bit in the interrupt status register. Card interrupts, however, shall be cleared by the Card Driver. If the Card Interrupt is generated, the Host Driver may clear Card Interrupt Status Enable to disable card interrupts while the Card Driver is processing them. After all interrupt sources are cleared, the Host Driver sets it again to enable another card interrupt. Disabling the Card Interrupt Status Enable avoids generating multiple interrupts during processing interrupt service.

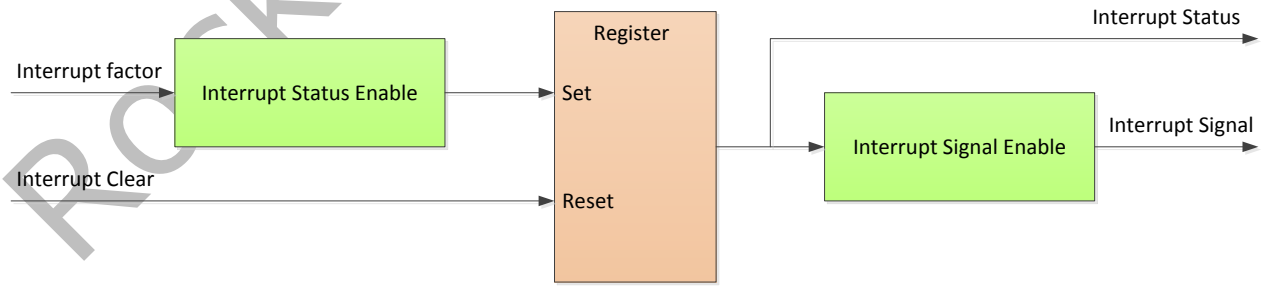


Fig. 16-25 Logical Relation for Interrupt Registers

16.7.10 Command Queuing Driver Flow Sequence

1. Command Queuing Initialization Sequence

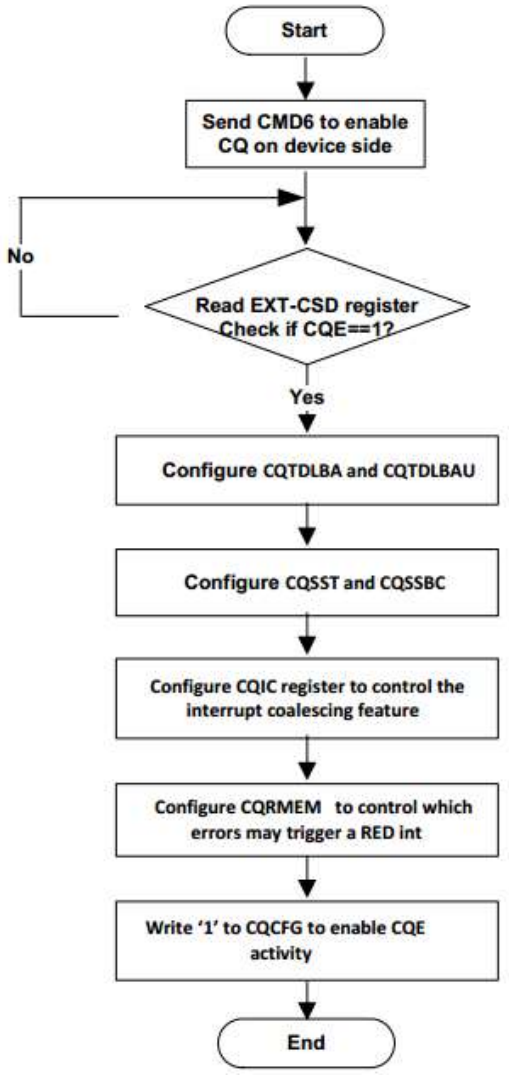


Fig. 16-26 Command Queuing Initialization Sequence

In order to start using Command Queuing and initialize CQE hardware, the following steps should be executed:

1) Initialize and enable Command Queuing in the device (beyond the scope of this section).
2) Configure Task Descriptor size in CQCFG register.
3) Configure CQTDLBA and CQTDLBAU to point to the memory location allocated to the TDL in host memory.
4) Configure CQSSC1 to control when SEND_QUEUE_STATUS commands are sent to the device by CQE.
5) Configure CQIC register to control the interrupt coalescing feature: enable/disable, set interrupt count and timer protection.
6) Configure CQRMEM to control which errors may trigger a RED interrupt (if different from reset values).
7) Write '1' to CQCFG to enable CQE activity.

2. Task Issuance Sequence

When host software needs to issue a data Transfer Task to the CQE and eMMC device, it utilizes Task Descriptor List.

The following is the steps for host software to build a Task and issue it:

1) Find an empty transfer request slot by reading the CQTDBR. An empty transfer request slot has its respective bit cleared to '0' in the CQTDBR.
2) Build a Task Descriptor at the 1 st entry of the empty slot. Task Descriptor field values:
   a. Valid =1, to indicate the descriptor is effective.
   b. End =1, as required for Task Descriptors.
   c. Int=1, if an interrupt on completion is required. Otherwise, Int=0.

d. Act = b101, to indicate a Task Descriptor
e. Data Direction =1 for read commands, and 0 for write commands.
f. Priority =1 for high priority, and 0 for simple requests.
g. QBR =1 if Queue Barrier functionality is required, 0 otherwise.
h. Forced Programming, Context ID, Tag Request, Reliable Write, Block Count, and Block Address programmed according to application requirements.

3) Build a Transfer Descriptor at the 2nd entry of the empty slot. Transfer Descriptor field values:
a. Valid =1, to indicate the descriptor is effective.
b. End =1, if a TRAN descriptor is used, to indicate the task only has one data buffer. End =0 if LINK descriptor is used.
c. Int=0. Ignore in Command queuing.
d. Act = b100, for TRAN descriptors, pointing directly to the task's single data buffer. Act = b110, for LINK descriptors, point to a scatter/gather list (indirect)
e. Address and Length programmed according to the data buffer supplied by the application.
4) If more than one transfer is requested, repeat step 1 -3 for all needed transfers
5) Set CQTDBR to ring the doorbell register to indicate to the CQE that one or more transfer requests are ready to be sent to the attached device. Host software shall only write a '1' to the bit position that corresponds to new tasks; all other bit positions within CQTDBR should be written with a '0', which indicates no change to their current values.

Following the steps below, the host hardware starts processing the task and sends it to the device. The sequence for queuing a task, between the host software, CQE and the e•MMC device, is given in following figure.

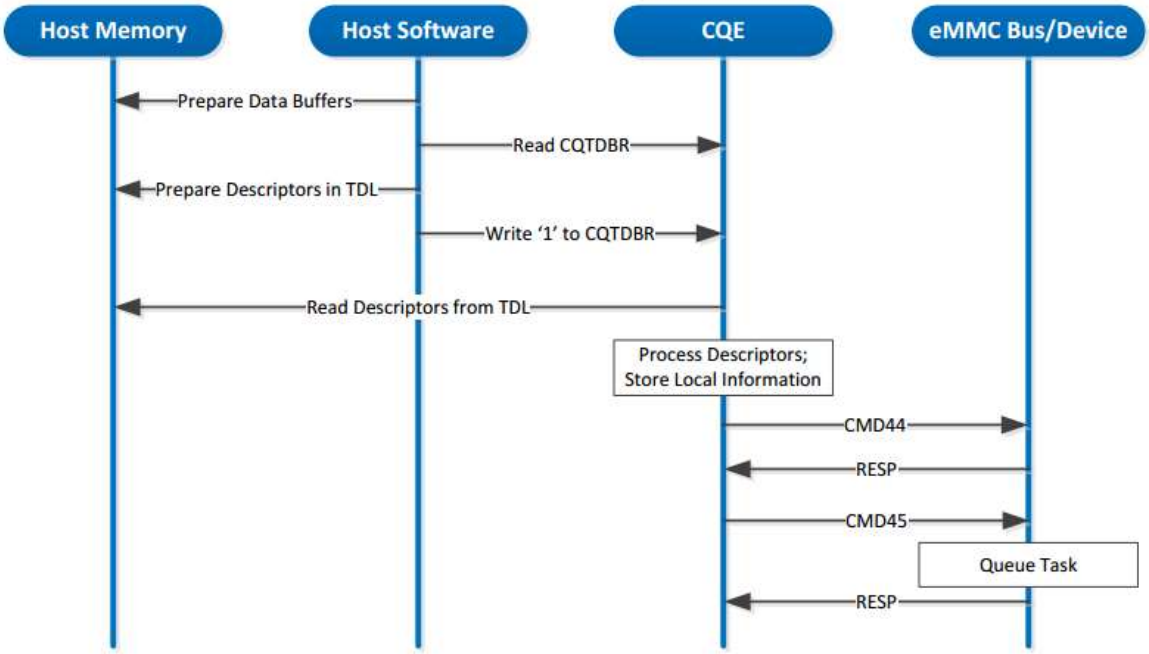


Fig. 16-27 Task Queuing Sequence

3. Task Execution and Completion Sequence

The CQE is responsible for task execution, communication with the device and moving the data to the buffers in the host memory. When the task execution is completed, an interrupt may be generated, if requested, or as determined by Interrupt Coalescing mechanism.

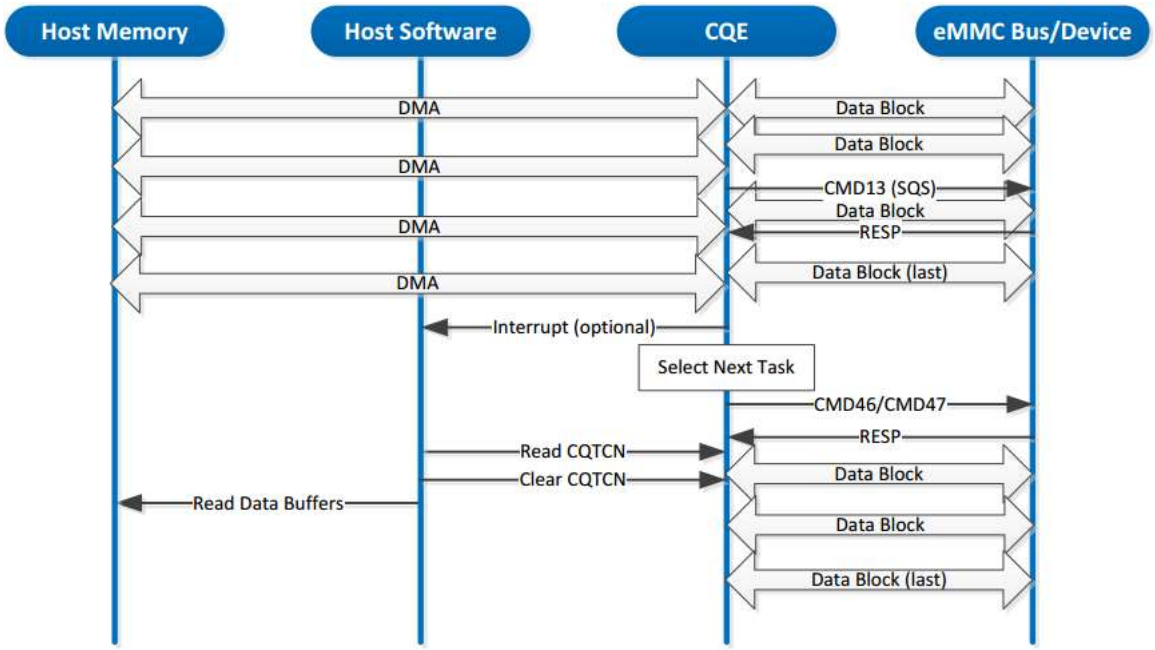


Fig. 16-28 Task Execution and Completion Sequence

In the example in Figure above, CQE sends a CMD13 during the 2nd-to-last block of a data transfer to determine the queue status, in order to decide which task to queue next. The response to CMD13 may indicate that no task is ready for execution. In such a case the host is required to send CMD13 again, at a later time, which is controlled by CQSSC1.CIT register.

When the task is completed, an interrupt may be generated to host software, and an EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) is sent to the device, ordering it to execute the next task.

When receiving the interrupt, host software takes the following steps:

1) Read CQTCN to determine which task(s) has (have) completed. Each bit which is set in CQTCN represents (by its index) a task which has completed but wasn't yet served by software.
2) For every task completed:
   a. Clear appropriate CQTCN bit
   b. Pass a completion notification to the requesting application

4. Task Discard & Clear Sequence

In order to discard a task, host software should notify the device and the CQE separately. The task is discarded by the device when a CMDQ_TASK_MGMT (CMD48) is sent. The CQE is issued a "task clear" command using the CQTCLR register. Both operations need to be done, without a specific ordering between them, as long as the CQE is kept in Halt state. The procedure is illustrated in figure below.

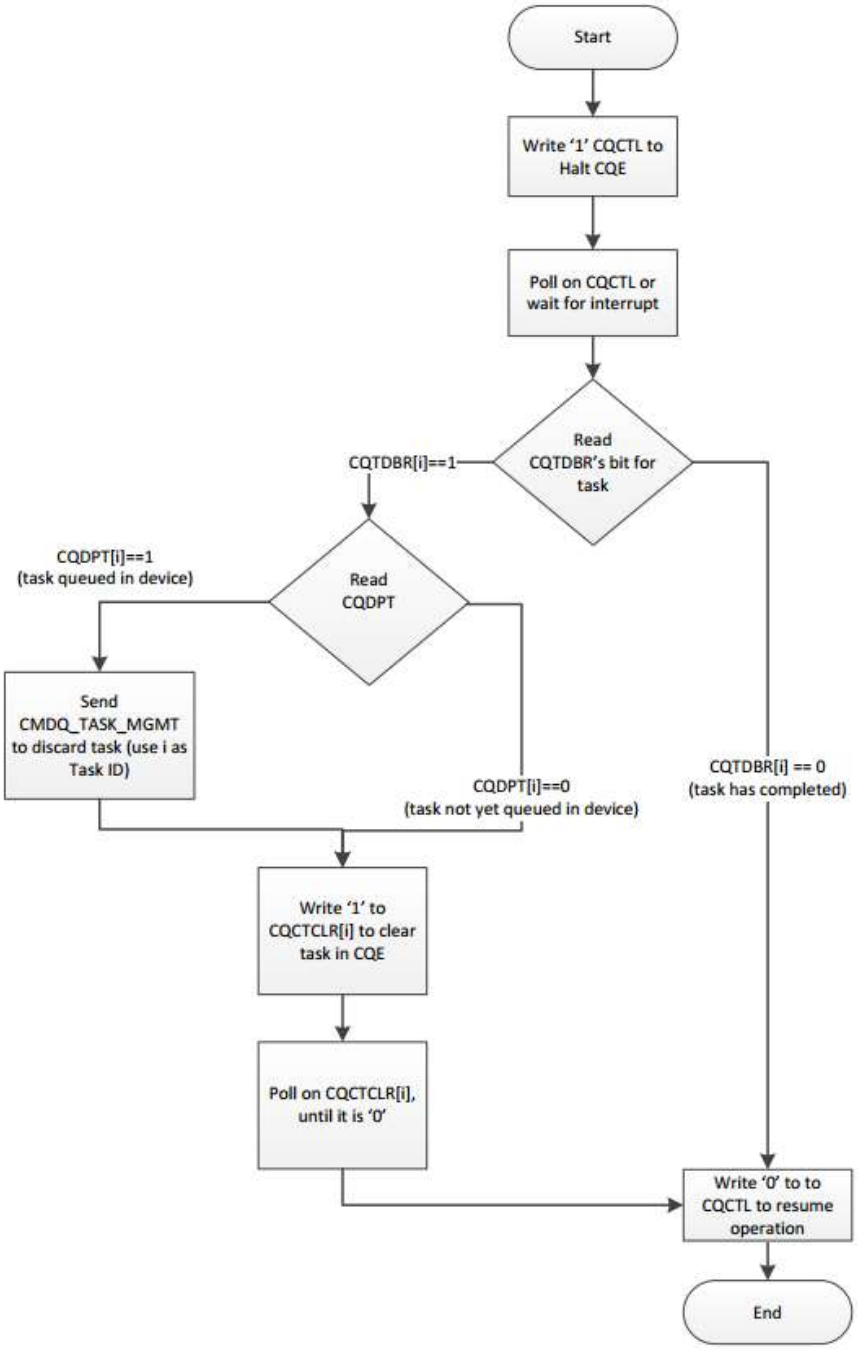


Fig. 16-29 Task Discard and Clear Sequence Diagram

5. Error Detect and Recovery when CQ is enabled

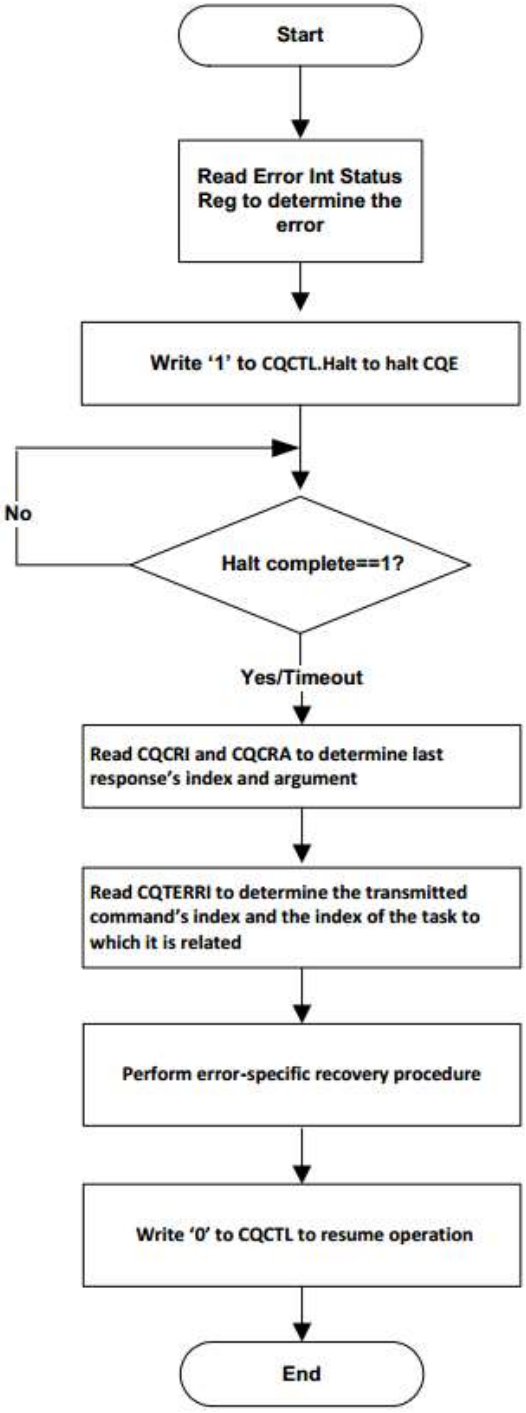


Fig. 16-30 Error Detect and Recovery

The following steps should be taken by software when responding error interrupts from CQE and eMMC host controller, when CQ mode is in use:

1) Read eMMC host controller Error Interrupt Status register and determine error is related to CQE
2) Write '1' to CQCTL.Halt to halt CQE
3) Wait for CQCTL.Halt to read '1'. In some error cases, this may not happen, so software should proceed to the next step after a sufficient time-out.
4) Read CQCRI and CQCRA to determine last response's index and argument
5) Read CQTERRI to determine the transmitted command's index and the index of the task to which it is related
6) Perform error-specific recovery procedure
7) Write '0' to CQCTL to resume operation

16.7.11 PHY Power Management

1. PHY Power up Sequence

The order of Powering up VDDQ, VCORE, VCORE_DLL and Always on VCORE does not matter for eMMC PHY.
The '*phyctrl_pdb*' should be asserted after VDDQ, VCORE and always on VCORE is ready.
The CALIO output signal from ACS PHY to the SOC '*phyctrl_caldone*' indicates that the calibration cycle is completed successfully. The SOC should wait for '*phyctrl_caldone*' to be asserted to start data transfer on EMMC51 data lines. The duration of the calibration cycle depends on CALIO internal clock generator frequency and varies with PVT. The CALIO design is optimized for calibration duration of less than 2 us.
The ACS PHY DLL can be enabled by asserting '*phyctrl_endll*' after VCORE and VCORE_DLL is ready during power up sequence. The '*phyctrl_endll*' power on default must be set low while VCORE and VCORE_DLL is on to allow initialization of DLL DFFs.
After the DLL control loop reaches steady state a DLL ready signal is generated by the DLL circuits '*phyctrl_dllrdy*'. The time from '*phyctrl_endll*' to DLL ready signal '*phyctrl_dllrdy*' varies with the clock frequency. At 200MHz clock frequency the DLL ready delay is 2.56us, at 100MHz clock frequency the DLL ready delay is 5.112us and at 50 MHz clock frequency the DLL ready delay is 10.231us. .
During initialization the ACS eMMC5.1 Host expects the eMMC5.1 data lines and CMD line to be in high state. The power on default of eMMC5.1 DATA and CMD I/Os should be programmed in weak pull up state by setting the power on default '*phyctrl_ren_dat[7:0]*' to 8b'11111111 , '*phyctrl_ren_cmd*' to 1b'1, '*phyctrl_up_dat[7:0]*' to 8b'11111111 and '*phyctrl_up_cmd* 'to 1b'1.

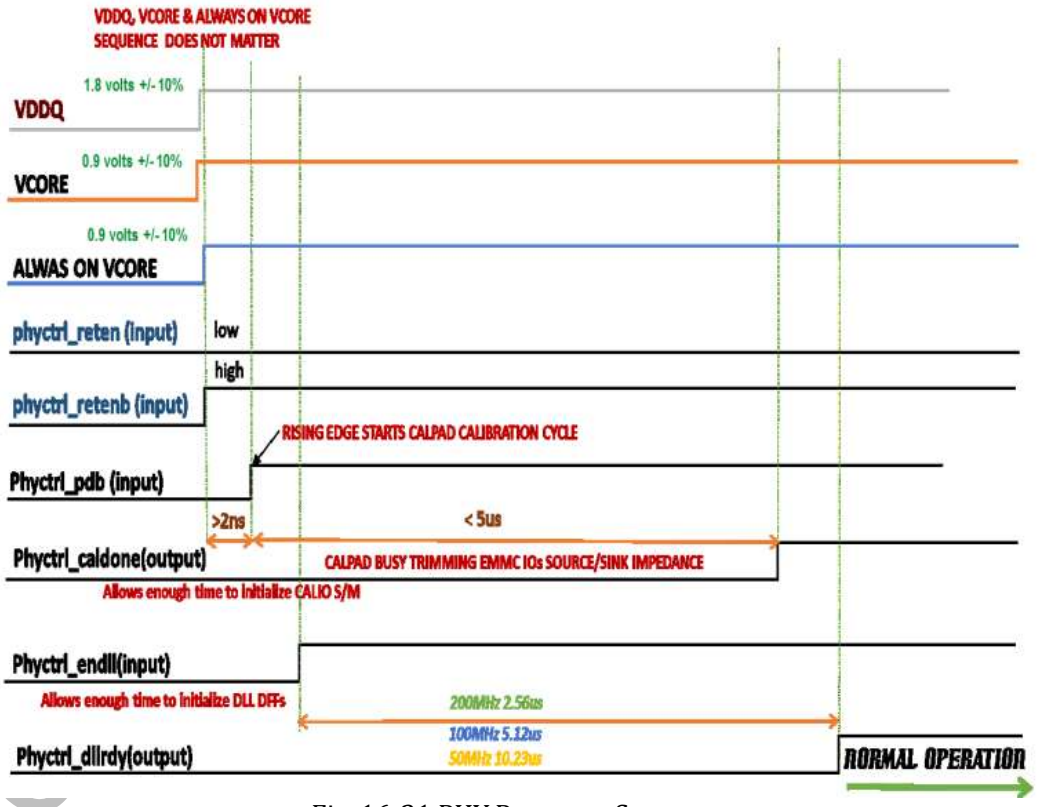


Fig. 16-31 PHY Power up Sequence

2. Sleep Mode and Retention Mode

When the SOC controller enter sleep mode it is desirable to switch off VCORE power supply to save power. The PHY IP includes a retention mode feature. If the retention mode is enabled '*phyctrl_reten*' 1b'1 and '*phyctrl_retenb*' 1b'0 the EMMC IOs latches and preserve all their control input in VDDQ power domain.
When the SOC exit sleep mode and enable VCORE, as soon as retention mode is disabled '*phyctrl_reten*'== 1b'0 and '*phyctrl_retenb*' ==1b'1. The EMMC I/Os would revert to its original state before sleep mode entry. The '*phyctrl_pdb*' must be held low for few nano seconds to allow initializing the RTRIM hold registers. After '*phyctrl_pdb*' is asserted the CALIO starts a calibration cycle to correct the Source / Sink impedance of EMMC IOs.
When the SOC exit sleep mode and after VCORE is switched on, the SOC has to update the configuration before retention is de-asserted.

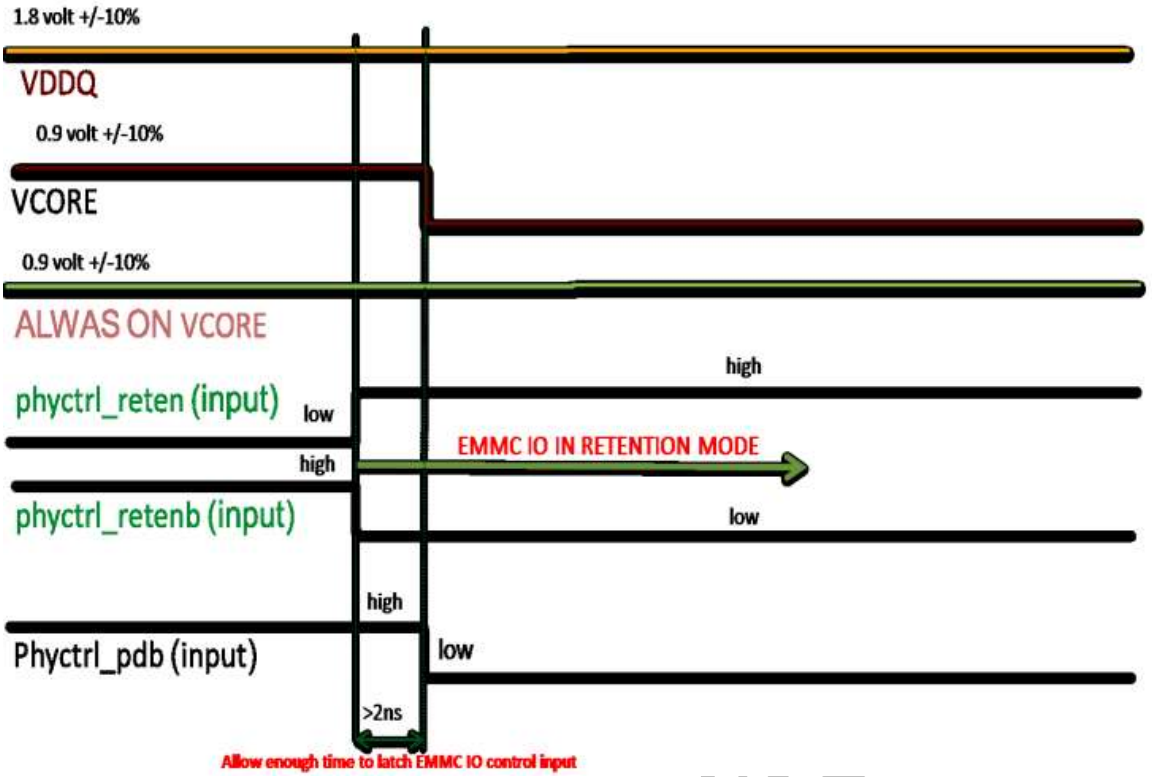


Fig. 16-32 Sleep Entry with Retention

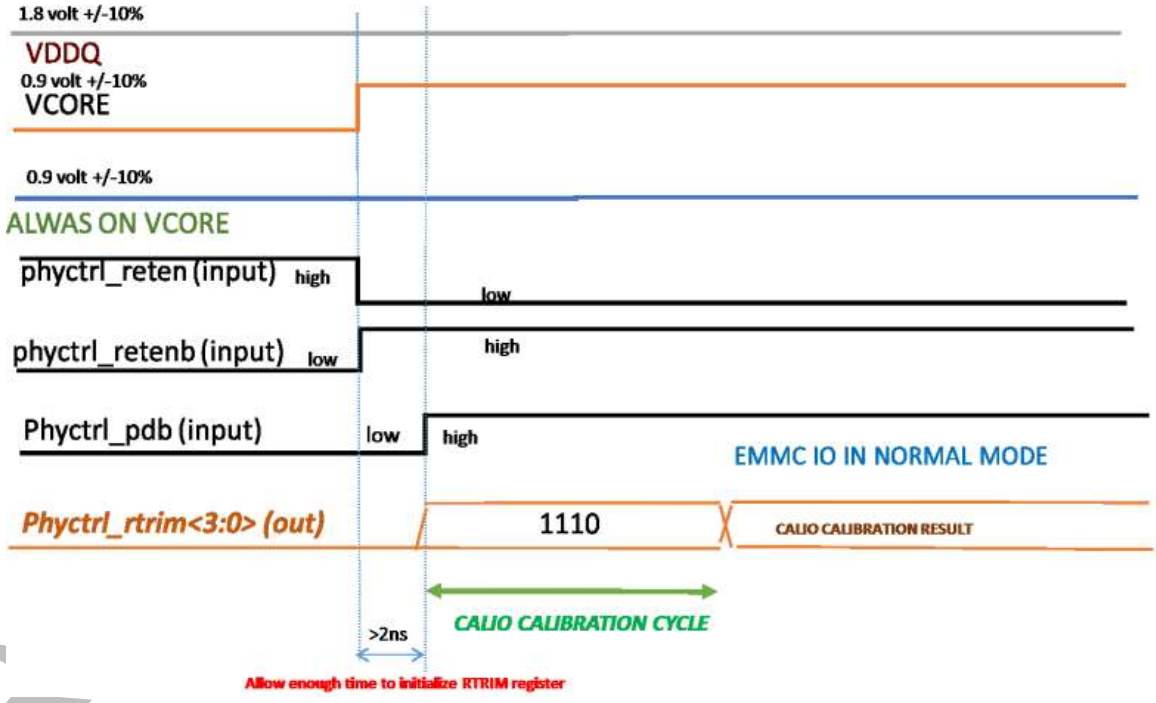


Fig. 16-33 Sleep Exit after Retention

Chapter 17 PCIe Controller

17.1 Overview

The PCI Express(PCIe) is a high performance, general purpose I/O interconnect defined for a wide variety of future computing and communication platforms. The PCIe designed to be used as a general purpose serial I/O interconnect in multiple market segments, including desktop, mobile, server, storage and embedded communications. It is compliant with PCI Express Specifications 1.1, 2.1.

PCIe supports the following features:

- Compatible with PCI Express Base Specification Revision 2.1
- Dual operation mode: Root Complex(RC)and End Point(EP)
- Maximum link width is 4, single bi-directional Link interface
- Maximum Payload Size of 256 bytes
- Maximum Read Request Size of 256 bytes
- Maximum 16 Non-Posted outstanding transactions
- Support 2.5GT/s and 5.0 GT/s serial data transmission rate per lane per direction
- Support 100MHz differential clock output (optional with SSC) for system application
- Support Single-root I/O virtualization(SR-IOV)
- Support DMA within the module, 2 channels, 2 RAM partitions, 2K bytes depth
- Support Transaction Processing Hints(TPH), ST Table Location at 1, ST Table Size is 7
- Support Latency Tolerance Reporting(LTR)
- Support Optimized Buffer Flush and Fill(OBFF)
- Support Resizable BAR Capability
- Support performance monitor signals at client
- Support Single Physical PCI Functions in Endpoint Mode
- Support Legacy Interrupt
- Support MSI and MSI-X interrupt
- Support ECRC Generation and Checking
- Support Outbound and Inbound address translation
- Support 8 Virtual Functions attached to Physical Function
- Support Power Budgeting Capability
- Support Dynamic Power Allocation Capability(DPA), Max Number Of Substate is 7
- Support PCI Express Active State Power Management (ASPM) state L0s and L1
- Support L1 Power Management Substate
- Support PCI Function power states D0, D1 and D3, and the corresponding link power states L0, L1 and L2

17.2 Block Diagram

The diagram of PCIe module is shown below:

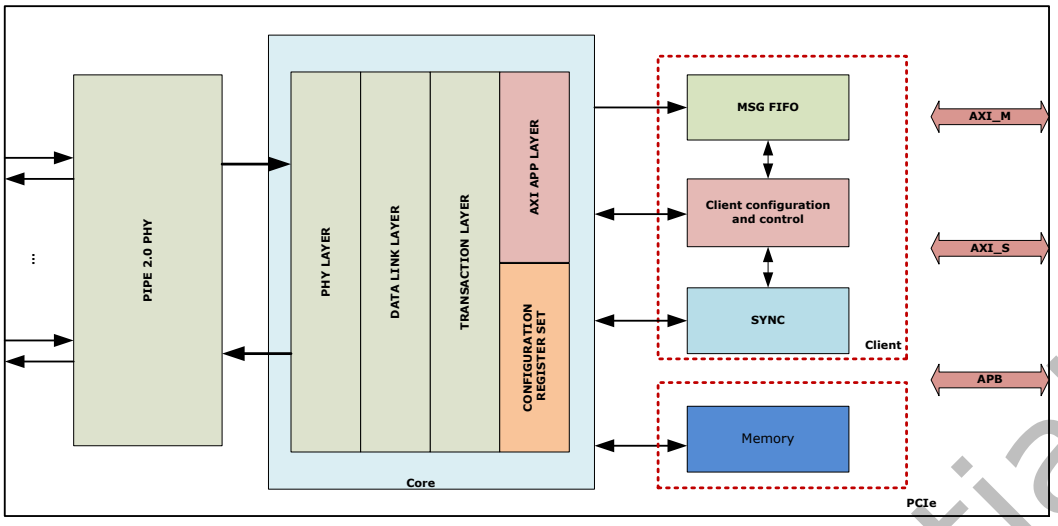


Fig. 17-1 PCIe Block Diagram

17.3 Function Description

17.3.1 Application Layer

The Application Layer provides a simple interface to a host bus or DMA engine on the user side. It consists of four separate interfaces to the user's logic on the device:

- Target memory read/write interface: Enables easy interfacing to the user's memory controller or DMA engine. This interface is also used to deliver I/O requests and messages received from the link to the client. This interface is needed for all Endpoints.
- Master read/write interface: Enables an Endpoint to generate memory transactions to the host as bus master; or a Root Port to generate memory, I/O, configuration and message requests. This interface is needed only in devices that require bus master capability, that is, in all Root Ports and in Endpoints that implement master capability.
- Interrupt interface: Provides interface to signal interrupt state between the Local Client and the core, the Local Client will generate three types interrupt to interrupt controller.
- Configuration interface: Provides interface to configure the Client strap configuration and status inquiry. Beyond that, a local processor may access and modify the contents of the configuration registers associated with the PCI Functions, as well as other management registers within the PCIe Core.

The Application interface contains all the logic required to maintain the state of up to 16 non-posted transactions (memory reads, I/O reads and writes, configuration reads and writes) generated on the master side, so that their Completions can be matched against the requests.
The below shows the diagram of the AXI module.

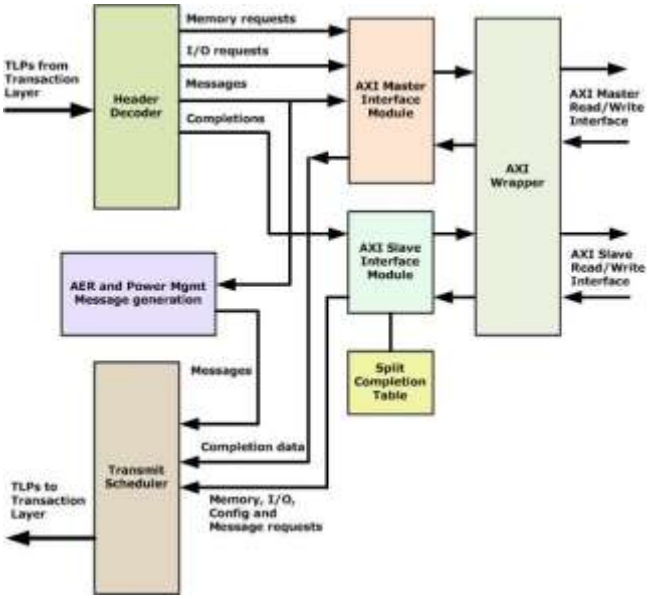


Fig. 17-2 PCIe AXI module

17.3.2 Local Client

The generalized application Client consists of system and local client. The local client implements the message receiver(MSG FIFO) and Client configuration and control. The MSG FIFO can optionally store the received message into FIFO and then generate an interrupt, it also records the length of the message in double word. The Client configuration and control module mainly contains the strap configuration, such as PCIe mode, link width, and so on. It also provides core internal status polling. Beyond that, the Client is in charge of the link control and interrupt generation.

17.3.3 Transaction Layer

The following diagram is an illustration of the receive side of the Transaction layer.

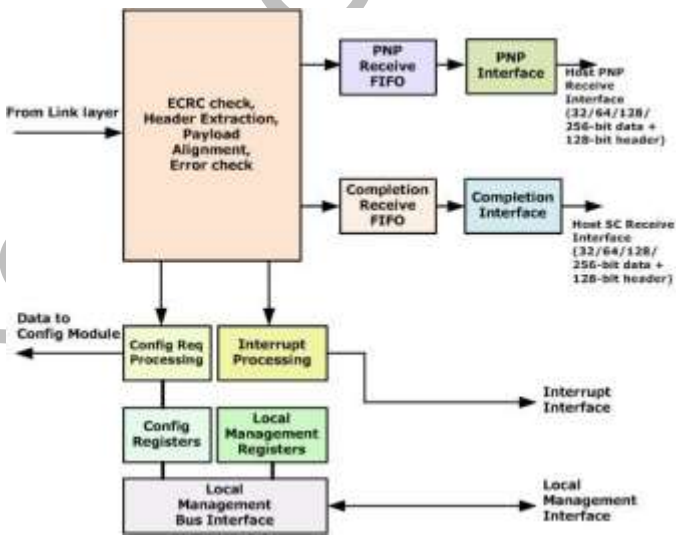


Fig. 17-3 PCIe Transaction Layer Receiver

On the receive side, data arrives from the Data link layer over a data path of size 128 bits. Logic within the transaction layer decodes the packet header, performs ECRC check when the packet is accompanied by a TLP Digest, and aligns the payload on the data path. The data then goes through store-and-forward FIFOs. There is a separate FIFO for Posted/Non-Posted and Completion packets.

A packet is read out from the FIFO only when the entire packet has already been received. The decoding logic classifies packets based on their TLP header and forwards them to the appropriate modules.

All read/write requests to configuration registers are processed within the Transaction layer. Logic within the Transaction layer routes these requests to the register set of the function addressed by the request, and returns completion packets back to the link. All interrupt related messages are processed by a separate interrupt processing module, which controls the Interrupt interface of the core. The error messages are processed by an error handling module.

The receive flow control parameters of the core (payload and header credit for the three types of packets: Posted, Non- Posted and Completion) are set based on the available space in the receive FIFOs. The flow control protocol then ensures that the FIFOs do not overflow. The FIFOs communicate their state to the flow control module so that when the packet is forwarded out of the FIFO, the corresponding credit becomes available and can be advertised to the link.

The following diagram is an illustration of the transmit side of Transaction layer. PNP requests and Completion (SC) packets arrive from the host over separate interfaces. Logic within the Transaction Layer multiplexes the packets, inserts the TLP header (and optionally the ECRC) , and forwards them to the Data link layer. The Completions and messages generated within the core are also multiplexed on the same data path to the Data link layer.

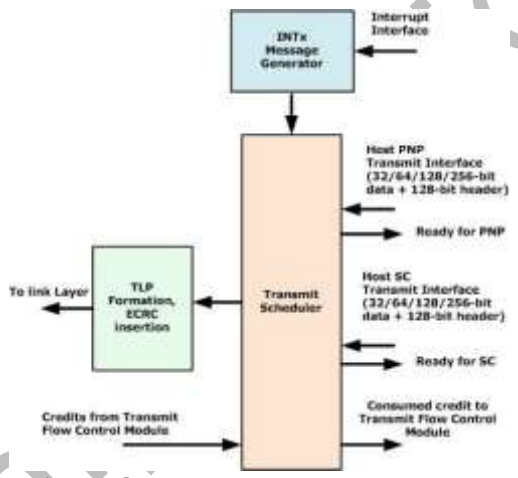


Fig. 17-4 PCIe Transaction Layer Transmitter

17.3.4 Data Link Layer

The following diagram illustrates the link layer data flow within the core on the receive side.

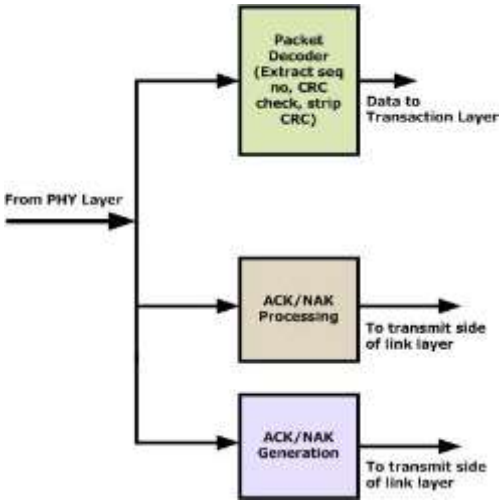


Fig. 17-5 PCIe Link Layer Receive Side

On the receive side, the data from the link passes through a decoder state machine for each link. The decoders verify the integrity of packets by matching the received CRC with the generated CRC, and comparing their sequence numbers with the expected values (for TLPs). There is a separate CRC module for TLPs (32-bit CRC) and DLLPs (16-bit CRC). After comparing and removing the link-layer CRC, the DLLP decoders pass on the received packet to its target module based on the packet type. All received TLPs are passed on to the Transaction Layer (after removing their sequence number and LCRC fields). Data link layer acknowledgements (ACKs and NAKs) are processed within the Data link layer itself, and credit DLLPs are sent to the flow control module.

After the CRC check, the arriving Data link layer acknowledgements are sent to the transmit side of the Data link layer for processing. Logic on the transmit side match the acknowledgements with outstanding packets and deal with all the error conditions.

The receive side of the Data link layer also generate acknowledgements (ACKs and NAKs) for the received TLPs. These packets are sent to the transmit side of the Data link layer, where they are multiplexed with outgoing TLPs.

The following diagram shows the data flow on the transmit side.

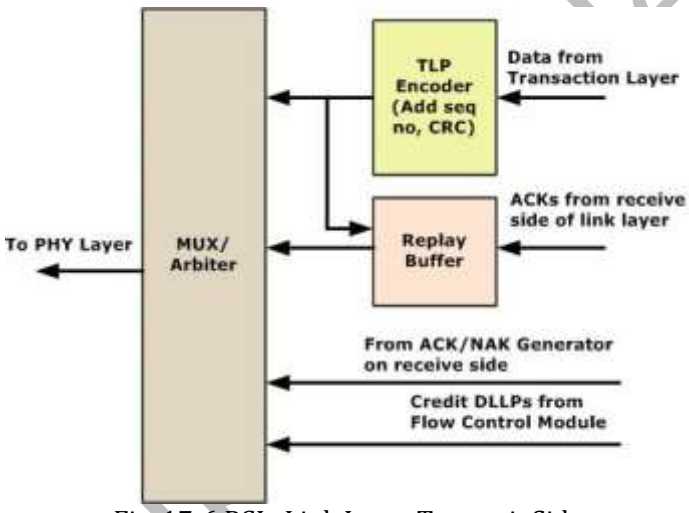


Fig. 17-6 PCIe Link Layer Transmit Side

17.3.5 Physical Layer

The following illustration is the block diagram of the receive side of the Physical layer.

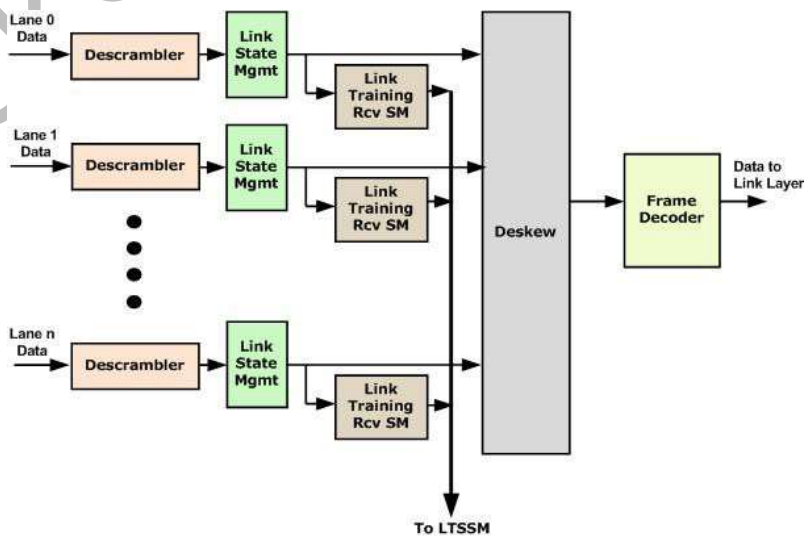


Fig. 17-7 PCIe Physical Layer Receive Side

Data arrives from the PIPE interface over one or more lanes, over a 16-bit interface per lane, at a clock frequency(125,250MHz) depending on the speed of operation of the link. The data is first converted to the core clock domain. The data from each lane is de-scrambled independently. The data then goes through logic to detect link power state transitions. The lanes are then de-skewed by passing the data through FIFOs that are aligned on SKP sequences. The lanes are aligned as a single unit.

The de-skewed data is then decoded by a frame decoder. The frame decoder removes the SOP/EOP framing delimiters from the packet and aligns them on the internal data path. The frame decoder is designed to deal with varying link widths and all possible alignments of packets on the lanes. The decoded data is then passed to the Data link layer with indications such as packet-type and errors detected.

The received data from each lane is also routed to a Link Training Receive State Machine, which is responsible for detecting and decoding training sequences received from the lane. Each of these state machines pass on information extracted from the training sequences to the LTSSM

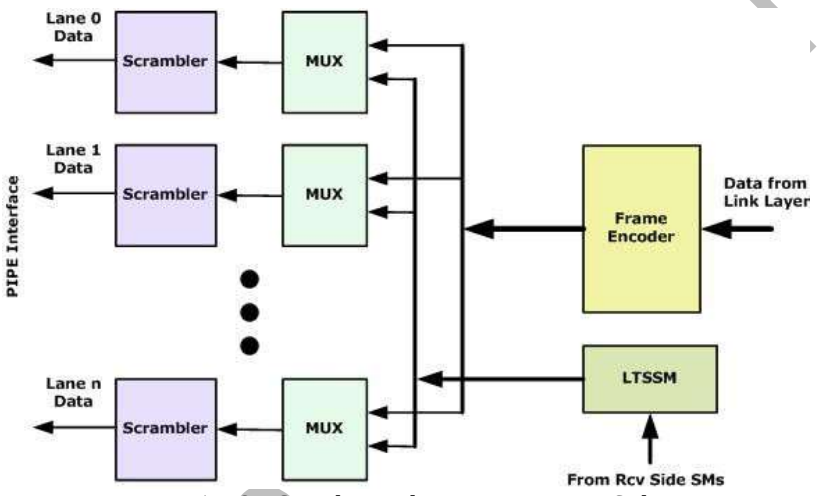


Fig. 17-8 PCIe Physical Layer Transmit Side

17.3.6 PIPE2.0 PHY Controller

The PCIe PHY, which consists of Physical Coding Sub-layer(PCS) and Physical Media Attachment Layer(PMA), includes all circuitry for interface operation, including 8/10 encoding/decoding, driver and input buffers, parallel-to-serial and serial-to-parallel conversion, PLL(s), and impedance matching circuitry. The PHY provides standard PIPE interface with the Media Access Layer for exchanging information. This PHY is responsible for converting information received from the MAC into an appropriate serialized format and transmitting it across the PCIe Link at a frequency and width compatible with the device connected to the other side of the Link.

17.4 Interface Description

17.4.1 Signal Description

Table 17-1 PCIe Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
A_TX_P	O	IO_PCIE_A_TX_P	NS
A_TX_N	O	IO_PCIE_A_TX_N	NS
A_RX_P	I	IO_PCIE_A_RX_P	NS
A_RX_N	I	IO_PCIE_A_RX_N	NS
B_TX_P	O	IO_PCIE_B_TX_P	NS
B_TX_N	O	IO_PCIE_B_TX_N	NS

Module Pin	Direction	Pad Name	IOMUX Setting
B_RX_P	I	IO_PCIE_B_RX_P	NS
B_RX_N	I	IO_PCIE_B_RX_N	NS
C_TX_P	O	IO_PCIE_C_TX_P	NS
C_TX_N	O	IO_PCIE_C_TX_N	NS
C_RX_P	I	IO_PCIE_C_RX_P	NS
C_RX_N	I	IO_PCIE_C_RX_N	NS
D_TX_P	O	IO_PCIE_D_TX_P	NS
D_TX_N	O	IO_PCIE_D_TX_N	NS
D_RX_P	I	IO_PCIE_D_RX_P	NS
D_RX_N	I	IO_PCIE_D_RX_N	NS
REF_CLK_100M_P	O	IO_PCIE_ CLK_100M_P	NS
REF_CLK_100M_N	O	IO_PCIE_ CLK_100M_N	NS
CLKREQN_	I/O	IO_SDIOdetectn_PCIEclkreqn_WIFIBTgpio2d2	GRF_SOC_CON7[14]=1'b0 GRF_GPIO2D_IOMUX[5:4]=2'b10
		IO_PCIEclkreqnb_GPIO1830gpio4d0	GRF_SOC_CON7[14]=1'b1 GRF_GPIO4D_IOMUX[1:0]=2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

17.5 Application Notes

17.5.1 Protocol Introduction

17.5.1.1 PCIe Topology

A fabric is composed of point-to-point Links that interconnect a set of components – an example fabric topology is shown below. This figure illustrates a single fabric instance referred to as a hierarchy – composed of a Root Complex (RC), multiple Endpoints (I/O devices), a Switch, and a PCI Express to PCI/PCI-X Bridge, all interconnected via PCI Express Links.

- Root Complex: The Root Complex can be understood as the interface between the system CPU and the PCIe topology, with PCIe Ports labeled as "Root Ports" in configuration space
- Switch: Fan-out or aggregation capability, act as router
- Bridge: Interface to other buses, such as PCI/PCI-X or PCIe
- Endpoint: Act as initiators and Completers of transactions on the bus. Have legacy EP and native EP

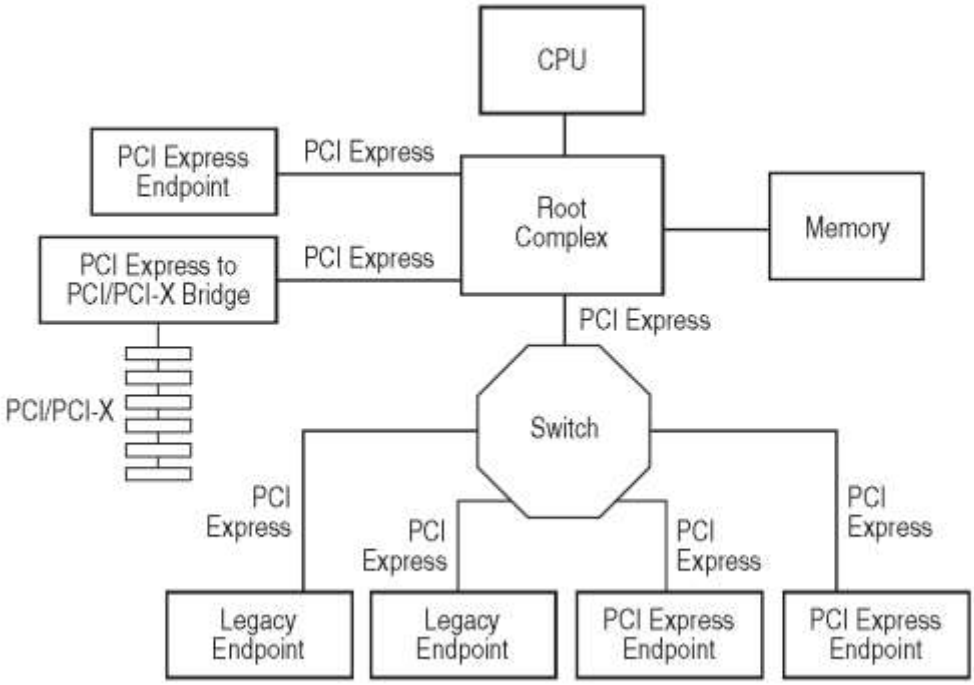


Fig. 17-9 Example Topology

17.5.1.2 Serial Link

PCI Express is a point-to-point serial signaling protocol. Each link consists of one TX and one RX differential pair. On each link lane, depending upon the protocol version, 2.5 GT/s or 5.0 GT/s can be transported in each direction. Accounting for the 8b/10b encoding used over the serial link, the data rates translate to 2.0 Gbps and 4.0 Gbps of throughput at higher layers of PCI Express protocol. Data is transferred in packets, which include an address and a variable size data payload. The PCIe protocol maximum supports 32 lanes, but current designed to support 1/2/4 lanes for different usage.

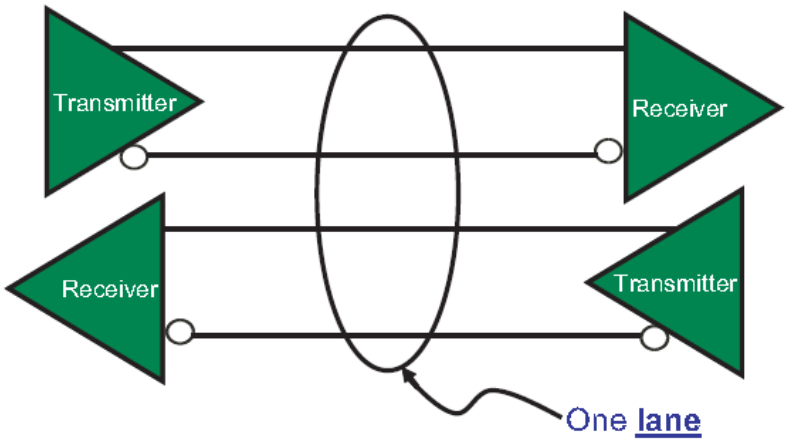


Fig. 17-10 One lane

17.5.1.3 Supported Transaction

All of the PCI Express transactions defined, Posted and Non-posted, are supported include the Locked Memory Read request transaction and its subsequent completion Locked response transaction. Inbound I/O read/write transactions also are not supported.
The non-posted transactions comprise of a transaction layer packet (TLP) from the requester to completer. The completer, at a later time, sends a completion TLP to the requester. The completion TLP is used to inform the requester that the completer has received the request. In addition, the completion TLP also contains the data if the transaction was a read transaction. Non-posted write transactions contain the data in request TLP.

For posted transactions, the request TLP is sent but there is no response TLP sent from the completer to the requester.

The below table shows the supported TLP:

Table 17-2 TLP Basics

Address Space	Types	Basic usage	N/P types
Memory	Write	Transfer data to/from a memory mapped location	Posted
	Read		Non-Posted
	Read Lock		Non-Posted
IO	Write	Transfer data to/from an I/O-mapped location	Non-Posted
	Read		Non-Posted
Configuration	Write	Device Function configuration/setup	Non-Posted
	Read		Non-Posted
Message	Write	From event signaling mechanism to general purpose messaging	Posted

17.5.2 Clock And Reset Control

17.5.2.1 Clock Application

The PCIe module uses multiple clock domains, it consists of system bus clock (APB and AXI), Core function clock, PIPE interface clock, Power management clock, and PCIe PHY reference clock. The system bus clock is generated from SOC main PLL internally, please see the relative data manual for details. The Power management clock which should be fixed with 24MHz driven by crystal input clock.

The PIPE interface clock is generated by PCIe PHY internal PLL, Core function clock source is configurable, which can be from PCIe PHY internal PLL or SOC main PLL. If the Core function clock is from SOC main PLL, user should make sure the PLL use the same reference clock with PCIe PHY PLL. The clock input to PCIe PHY is single-end clock which can be derived from crystal input(only support 24MHz) or SOC main PLL output(only support 100MHz), user should configure the CRU_CLKSEL18_CON that relative to PCIe before initialization and keep constant. Core function clock also can source from SOC main PLL.

The PCIe module doesn't comply with the PCIe specifications, it is unacceptable that a reference clock of 100 MHz that meets the requirements for REFCLK as described in the PCI Express Card Electromechanical Specification be driven as a differential signal into PCIe reference clock pads of the device.

Many PCIe connections, especially backplane connections, require a synchronous reference clock between the two link partners. To achieve this a common clock source, referred to as REFCLK in the PCI Express Card Electromechanical Specification, should be used by both ends of the PCIe link, the PCIe PHY provides 100MHz differential clock output (optional with SSC) for system application. If Spread Spectrum Clocking (SSC) is used it is required that a common reference clock be used by the link partners. Most commercially available platforms with PCIe backplanes use Spread Spectrum Clocking to reduce EMI. If common clock architecture is used, the user driver should configure the PHY internal configuration bits to enable the feature of 100MHz differential clock output. For detail information, please see PCIe PHY configuration.

The PCIe operation is completely dependent upon availability of clock from the PLL that is inside the PHY. All registers in the PCIe Core are located in the clock domain that is dependent upon PLL to be in lock and properly configured. So, transactions initiated before ensuring that the PLL is locked. The device level and Client registers can be used to enable PLL and verify lock status. Please see the PCIe PHY Configuration section for details.

17.5.2.2 Reset Application

17.5.2.2.1 Power-On Reset

The power-on reset is used as cold reset of the PCIe. The entire module is reset when power-on reset is asserted. After de-assertion of the power-on reset, the PCIe PHY and PCIe Core function reset keep until the software release them.

17.5.2.2.2 System Reset

The PCIe has the following distinct reset, all of these are configurable through software driver. This section describes the function of each of the reset inputs and the recommended sequences in which these should be activated. All in all, after power up, the software driver should de-assert the reset of PCIe PHY, then wait the PLL locked by polling the status, if PLL has locked, then can de-assert the rest reset simultaneously.

- RESET_N: This is the main reset for the core. It resets all the logic in the core running in the CORE_CLK domain, except management registers and Capability registers in the configuration spaces of the core. It keeps reset after power-up reset by default.
- MGMT_RESET_N: This resets all the local management registers of the core as well as all the Capability register fields in the configuration spaces to their default settings. The Capability register fields can only be read through a configuration transaction, but many of these can be modified through the local management bus. By separating this reset from the main reset, any re-programming of these registers needs to be done only once after power-on. The registers retain their settings when RESET_N is pulsed. It keeps reset after power-up reset by default. The MGMT_RESET_N is provided in order to give the user more flexibility in how the configuration and management registers are reset with respect to the controller itself. Having a separate reset allows the configuration and management register settings to be changed before bringing up the core. The core can then be reset without affecting the configuration parameters.
- MGMT_STICKY_RESET_N: This reset is similar to MGMT_RESET_N, except that it is connected only to the PCI/PCIe configuration register fields that are designated as sticky. Examples include the error status bits in the Advanced Error Reporting Capability Structure. These register fields are meant to be reset only on power-on. It keeps reset after power-up reset by default.
- PIPE_RESET_N: This reset signal resets the logic in the controller that interfaces to the PIPE interface in PIPE_PCLK domain. It keeps reset after power-up reset by default.
- AXI_RESET_N: This resets all logic in the AXI_CLK domain. Typically, this reset should be asserted whenever RESET_N is asserted. The client logic shall decide when it is appropriate to drive the reset to this AXI_RESET_N. It is expected that the AXI_RESET_N shall be driven active at Power on Reset. The client logic shall have knowledge of when an AXI request is in flight across the AXI interface and can choose an appropriate time to drive this reset in order to avoid times when a transaction is in flight
- APB_RESET_N: This resets all logic in the APB_CLK domain. This reset should be asserted whenever RESET_N is asserted.
- PM_RESET_N: This resets all logic in the PM_CLK domain. Typically, this reset should be asserted whenever RESET_N is asserted. A separate PM_RESET_N is therefore used to reset the L1 substates logic in the PCIe Controller, for those Configuration that implements the L1 substates ECN
- PCIE_PHY_RESET_N: This input resets all logic in the PHY. Typically, this reset should be asserted whenever PIPE_RESET_N is asserted. It keeps reset after power-up reset by default.

17.5.2.2.3 Hot Reset

The PCIe module contains a software reset (Hot_reset_in) bit in the hot reset control register that is used to issue a hot reset. In general, this reset is software controlled

procedure and can be issued only by a root complex in a PCIe network as the propagation is downstream only. When RC issued a hot reset, the relative bit should keep asserted till the Hot reset interrupt generates.

17.5.3 PCIe PHY Configuration

The PCIe PHY configuration contains two parts, one is device level that relative to reset and clock control, this section mainly introduces the PHY configuration.

PCIe PHY configuration should complete before Core reset de-assertion, its internal register can't be accessed by CPU directly. The software driver monitors the config bus sequence to access the register by writing to GRF_SOC_CON8, the timing sequence and configuration is shown in below Fig and Table.

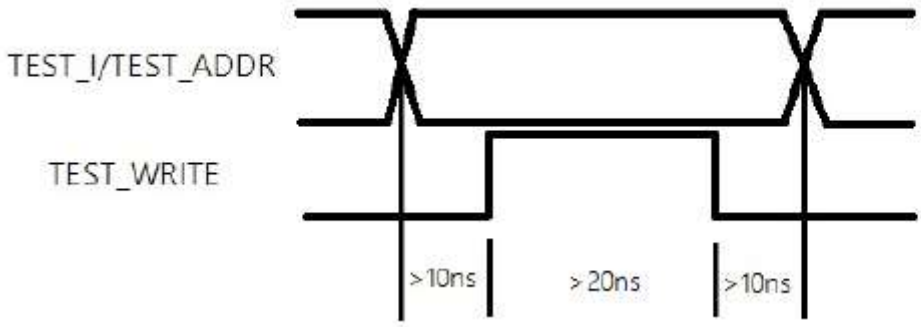


Fig. 17-11 PCIe Configure Timing sequence

Table 17-3 PCIe PHY Main Configuration Table

TEST_ADDR[4:0]	TEST_I[3]	TEST_I[2]	TEST_I[1]	TEST_I[0]
10000	separate_rate (1'b0)	1'b0	1'b0	1'b0
10001	SSCG_DISABLE (1'b1)	SSC_REDUCE (1'b0)	SSC_OFFSET (2'b00)	
10010	SEL_PLL_100M (1'b0)	GATE_100M (1'b1)	Reserved	
Bit description	*separate_rate: Forces PLL VCO to always run at 5Gbps with additional divider, while only CDR rate is controlled by RATE_I. When set low, both PLL and CDR run at RATE_I. *SSCG_DISABLE: Disables SSC modulation. Shall only be enabled when all the conditions are met: 1) in systems with common clocked architecture, 2) accuracy of REF_CLK frequency is within +/-300ppm, 3) the PHY is in the device acting as Root Complex, 4) 100MHz differential output clock is enabled and used for reference clock of EndPoint, 5) PLL is selected as the source of the differential 100MHz output clock. *SSC_REDUCE: Reduces SSC swing from 0~-4500ppm to -2000ppm~-4500ppm. *SSC_OFFSET: Adjusts SSC offset: 01 down 100ppm, 10 down 200ppm, 11 up100ppm. *SEL_PLL_100M: 100MHz differential clock is generated by PLL when this bit is high, otherwise it is from input port REF_CLK_100M. Note: the 100MHz clock from PLL would not be active until it has locked. *GATE_100M: The differential 100MHz output clock is gated asynchronously (high impedance) when this bit is high. By reset, this bit is high and the LVDS clock output is off regardless of which source the 100MHz clock is from. When this bit is low and PLL is selected as source of the 100MHz output but has not locked, LVDS clock output will keep constant and would NOT be off automatically, i.e. it is not high impedance and will draw current			

17.5.4 PCIe Address Space

From the PCIe perspective, the internal bus addressable resources are categorized into two address spaces. That is, the PCIe has two address spaces. The first is dedicated for internal registers, the second is dedicated for remote device. The address mapping about the internal register, please refer to section " Internal Register Address Mapping"

This section describes the mapping for remote device. The total size is 64 Mbytes and base address is 0xF8000000, it is divided into total 33 regions, the following equation describes how the region from 0 to 32 can be selected.

Region 0 is decoded by A[RW]ADDR [25] == 0, the region size is 32 Mbytes.

Region 1 to 32 (configured number of regions) is decoded by Region x = A[RW]ADDR [24:20] + 1, when A[RW]ADDR [25] == 1

Table 17-4 PCIe Outbound Region Select

A[25]	A[24:20]	Size(Mbytes)	Address Space[11:2]
0	X	32	Region 0
1	0	1	Region 1
1	1	1	Region 2
1	2	1	Region 3
1	3	1	Region 4
1	...	1	...
1	30	1	Region 31
1	31	1	Region 32

17.5.5 PCIe Address Translation

PCIe TLP transactions use PCIe addresses. There is a mapping requirement between a PCIe address and a local internal bus address and to accommodate this address mapping. An outbound or inbound address translation is required and is performed accordingly to map internal bus address to PCIe address or vice-versa using address translators. Address translations for outbound and inbound transactions are discussed detail below.

17.5.5.1 Outbound Address Translation Application

The Core uses the value of some of the incoming address bits to select from a bank of internal preprogrammed registers. The register values determine,

- the values of the PCIe header of the TLP
- the values on the upper bits of AWADDR and ARADDR
- how many AWADDR and ARADDR bits are passed through from the SoC AXI bus

The address space occupied by the AXI slave port is decoded within the Core into smaller address ranges. There are two modes for this decoding. Mode 0 decodes the address space into equally sized regions, with the number of regions configurable between 1 and 32. Mode 1 assigns the lower half of the address space to region zero and decodes the upper half into the configured number of equally sized regions. Mode 1 is intended for use in a root port supporting ECAM where it is beneficial to pass through a large number of SoC address bus bits for configuration reads and writes. Mode 1 supports this while minimizing the total address space the PCIe core occupies in the SoC. If the AXI Slave is configured for mode 1 and the PCIe core occupies 32MByte of SoC address space, then region 0 will occupy 16Mbytes and the bottom 24 bits of AWADDR and ARADDR can be passed through to the PCIe address for configuration reads and writes. This meets the PCIe recommendation for ECAM addressing with a 32-bit native bus.

Each region has a register bank supporting PCIe header descriptor of up to 64 bits and an address bus width of 64-bits. From a software perspective, the register bank is organized as 4 registers and each register bank is located on a 32- byte boundary. The register contents are interpreted differently by the AXI Slave depending on whether a memory, IO,

configuration or message TLP is being generated. It is assumed that no SoC read accesses will take place to a region which has been programmed for message writes

17.5.5.1.1 Memory and IO Accesses

When a SoC AXI access is made to a region holding a descriptor for memory or IO transactions then the PCIe header fields are driven from the register descriptor values and the SoC AXI transfer is passed to the PCIe transaction layer with the address translation as programmed in the register bank.

The AXI Slave drives a 64-bit address to the PCIe transaction layer from a configurable width SoC AXI address bus. The address driven to PCIe transaction layer will comprise lower bits passed through from the SoC AXI bus and higher bits driven from the address translation registers.

The maximum number of address bits which should be passed through is dependent on the region size the SoC CPU is accessing within the SoC address space allocated to the PCIe core. Some examples are:

- PCIe core occupies 256MByte, 32 regions, region size is 8MByte. AXI Slave should not pass more than 23 bits from SoC address to PCIe transaction layer
- PCIe core occupies 2MByte, 2 regions, region size is 1MByte. AXI Slave should not pass more than 20 bits from SoC address to PCIe transaction layer

It is possible that the AXI Slave may be configured to pass fewer address bits from the SoC address to the PCIe transaction layer than those corresponding to the region size. This would occur where the BAR block size for the TLP is less than the SoC AXI region size. In this case the number of address bits passed through should be configured to match the BAR block size.

The following table "Outbound TLP Register Bank - Memory and IO TLPs" describes the configuration register allocation when used for Memory and IO TLPs. For accesses below 4GByte in PCIe address space addr1 must be programmed to zero and addr0[5:0] must be programmed to pass 32 bits or less.

Table 17-5 Outbound TLP Register Bank - Memory and IO TLPs

Register	Bits	Allocation
desc3	31:0	Reserved
desc2	31:16	Reserved
desc2	15:0	PCIe header descriptor [79:64]
desc1	31:0	PCIe header descriptor[63:32]
desc0	31:0	PCIe header descriptor[31:0]
addr1	31:0	Address bits applied to ARADDR[63:32] and AWADDR[63:32], depending on addr0[5:0], when transaction is generated to PCIe transaction layer.
addr0	31:8	Address bits applied to ARADDR[31:8] and AWADDR[31:8] depending on addr0[5:0], when transaction is generated to PCIe transaction layer.
	7:6	Reserved
	5:0	Number of address bits passed through from SoC AXI to PCIe transaction layer. AXI Slave passes programmed value + 1 bits. This field must be programmed to pass at least 8 bits.

Table 17-6 PCIe Header Descriptors with description

PCIe Header descriptor	Description
[3:0]	Transaction type: ● 0010 = memory IO; ● 0110 = IO; All other values are reserved.
[6:4]	PCIe Attributes associated with the request.
[16:7]	Reserved for future use.
[19:17]	PCIe Traffic Class (TC) associated with the request.
[20]	When the request is a memory write transaction, setting this bit causes the core to poison the Memory Write TLP being sent. This bit has no effect for other types of transactions.
[21]	Force ECRC insertion. Setting this bit to 1 force the core to append a TLP Digest containing ECRC to the TLP, even when ECRC is not enabled for the Function generating the request.
[22]	Reserved for future use.
[23]	This input enables the client to supply the bus and device numbers to be used in the requester ID. If this input is 0, the core will use the captured values of the bus and device numbers to form the Requester ID. If this input is 1, the core will use the bus and device numbers supplied by the client on bits [39:32] and [31:27] to form the Requester ID. This bit must always be set while originating requests in the RC mode, and the corresponding Requester ID must be placed on bits [39:32].
[31:24]	PCI Function Number associated with the request. In the ARI mode, all the 8 bits are used to indicate the requesting Function. In the legacy mode, only bits [26:24] are used, and bits [31:27] are used to specify the device number to be used within the Requester ID, when bit [23] is set.
[39:32]	When bit [23] is set, this field must specify the bus number to be used for the Requester ID. Otherwise, this field is ignored by the core.
[47:40]	Reserved.
[63:48]	Reserved.
[71:64]	8-bit Steering Tag for the Hint.
[73:72]	Value of PH[1:0] associated with the Hint.
[75:74]	Reserved.
[76]	Set when the request has a Transaction Processing Hint associated with it.
[79:77]	Reserved.

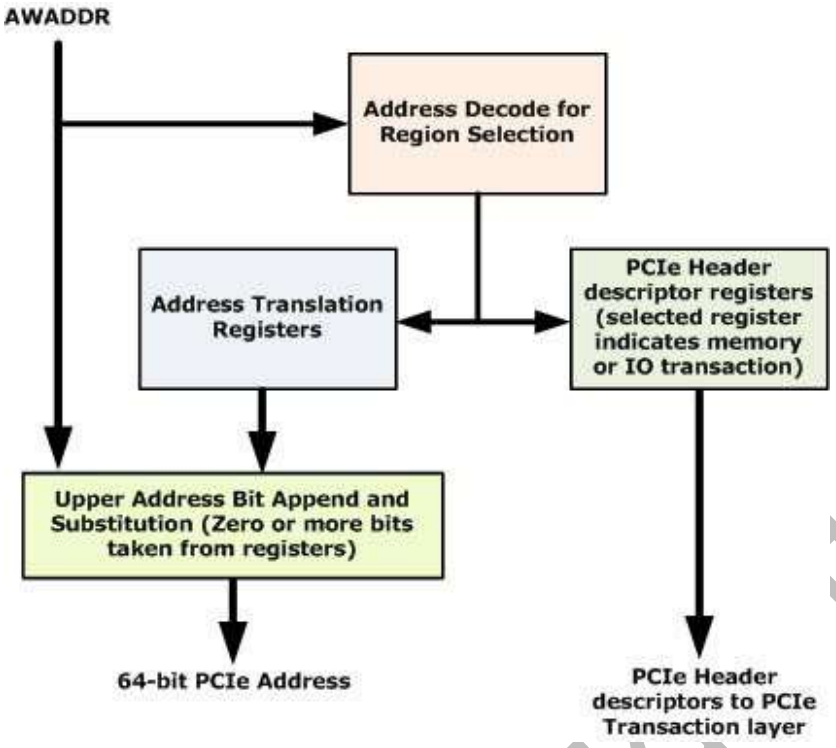


Fig. 17-12 Outbound Memory or IO Write Generation

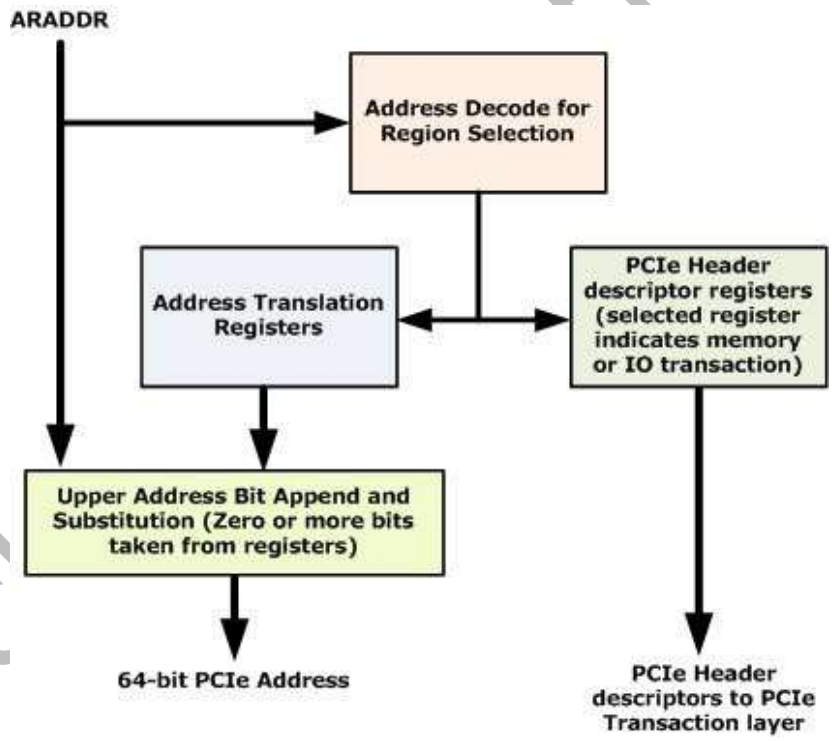


Fig. 17-13 Outbound Memory or IO Read Generation

17.5.5.1.2 Configuration Accesses

When a SoC AXI access is made to a region holding a descriptor for a configuration transaction then the PCIe Header fields are driven mainly from the Address translation registers. The PCI Completer ID field of the PCIe Header should be driven by bits [27:12] of the incoming AXI transaction. The SoC AXI transfer is passed to the PCIe Transaction layer with the address translation as programmed in the register bank. This is shown in Figure "Outbound Configuration Write Generation" and Figure "Outbound Configuration Read Generation".

Ideally the address translation registers will be programmed so that all address bits up to and including those fully defining the PCI completer ID will be passed through from the SoC

AXI bus instead of being supplied by the address translation IDs. This will enable the SOC CPU to address the entire ECAM address space as an address space in its own memory map and without the need to reprogram registers to reach different areas of the ECAM address space. The translation registers should not be programmed to pass more than 28-bits, and bits [63:28] in the translation registers should be programmed as zero.

Table 17-7 Outbound TLP Register Bank - Configuration TLPs

Register	Bits	Allocation
desc3	31:0	Reserved
desc2	31:0	Reserved
desc1	31:0	PCIe header descriptor[63:32]
desc0	31:0	PCIe header descriptor[31:0]
addr1	31:0	Should be programmed to zero.
addr0	31:28	Should be programmed to zero.
	27:20	Bus Number passed to write or read descriptor if addr0[5:0] is programmed to pass less than 28 bits.
	19:15	Device Number (or ARI ID[7:3])passed to write or read descriptor if addr0[5:0] is programmed to pass less than 20 bits.
	14:12	Function Number(or ARIID[2:0])passed to write or read descriptor if addr0[5:0] is programmed to pass less than 15 bits.
	11:8	Extended Register number passed to AXI2HAL address bus if addr0[5:0] is programmed to pass less than 12 bits.
	7:6	Reserved.
	5:0	Number of address bits passed through from SoC AXI to AXI2HAL. Must be programmed to pass at least 8 bits. Wrapper passes programmed value + 1 bits. This field must be programmed to pass at least 8 bits.

The following table describes the register allocation when used for Configuration TLPs.

Table 17-8 PCIe Header Descriptors with description

PCIe Header descriptor	Description
[3:0]	Transaction type: 1010 = Type 0 config; 1011 = Type 1 config; All other values are reserved.
[6:4]	PCIe Attributes associated with the request.
[16:7]	Reserved for future use.
[19:17]	PCIe Traffic Class (TC) associated with the request.
[20]	Reserved.
[21]	Force ECRC insertion. Setting this bit to 1 force the core to append a TLP Digest containing ECRC to the TLP, even when ECRC is not enabled for the Function generating the request.
[22]	Reserved for future use.

PCIe Header descriptor	Description
[23]	This input enables the client to supply the bus and device numbers to be used in the requester ID. If this input is 0, the core uses the captured values of the bus and device numbers to form the Requester ID. If this input is 1, the core uses the bus and device numbers supplied by the client on bits [39:32] and [31:27] to form the Requester ID. This bit must always be set while originating requests in the RC mode, and the corresponding Requester ID must be placed on bits [39:32].
[31:24]	PCI Function Number associated with the request. In the ARI mode, all the 8 bits are used to indicate the requesting Function. In the legacy mode, only bits [26:24] are used, and bits [31:27] are used to specify the device number to be used within the Requester ID, when bit [23] is set.
[39:32]	When bit [23] is set, this field must specify the bus number to be used for the Requester ID. Otherwise, this field is ignored by the core.
[55:40]	Reserved.
[60:56]	Reserved.
[63:61]	Reserved.
[71:64]	8-bit Steering Tag for the Hint.
[73:72]	Value of PH[1:0] associated with the Hint.
[75:74]	Reserved.
[76]	Set when the request has a Transaction Processing Hint associated with it.
[79:77]	Reserved.

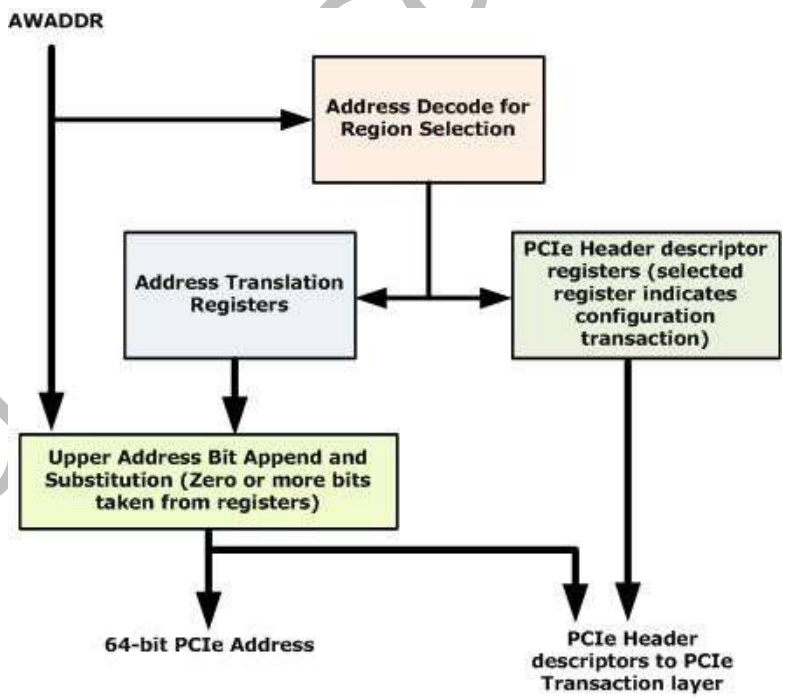


Fig. 17-14 Outbound Configuration Write Generation

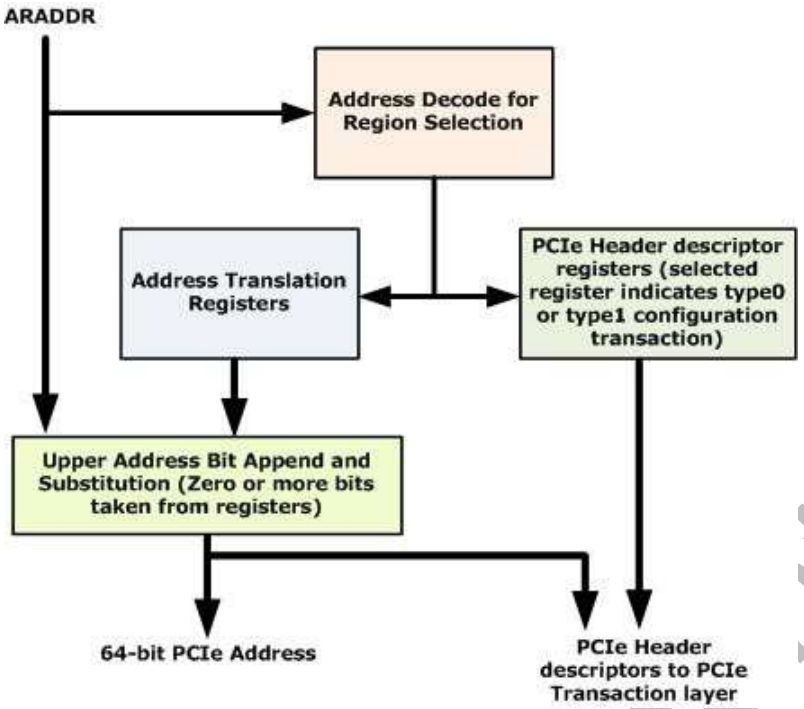


Fig. 17-15 Outbound Configuration Read Generation

17.5.5.1.3 Message Access

When a SoC AXI access is made to a region holding a descriptor for a message transaction then the PCIe Header fields are driven mainly from the register value. However the Message Code and Message Routing fields are driven by bits of the incoming AWADDR as follows:

- Normal Messages
  - [15:8] - Message Code
  - [7:5] - Message Routing
- Vendor Defined Messages
  - [15] - Message Code is 0x7E when bit is clear, 0x7F when bit is set
  - [14:12] - Message Routing

The SoC AXI transfer is passed to the PCIe Transaction layer, but all WSTRB strobes are negated if bit 16 of AWADDR was asserted on the oncoming transaction. The AXI Slave uses AWADDR bits [63:8] to receive bits [127:72] of the TLP header for vendor defined messages, and automatically drives these bits from the configuration registers for message TLPs. They should be programmed with the required values for Vendor Defined Messages and programmed to zero for Normal Messages.

When an AXI transaction is made with AWADDR[16] set, AXI Slave will negate WSTRB when forwarding the transaction, therefore generating a zero-data message. When an AXI transaction is made with AWADDR[16] set, the transaction length must be 1 beat (AWLEN = 0). If the AXI transaction is made to an address with AWADDR[16] clear WSTRB is passed unmodified and a message with data is generated. This is shown in Figure "Outbound Message Write Generation".

It is critical that the AXI transaction for a message with data is implemented as a single AXI burst from the SoC. This is straightforward for normal messages with data as they are 1DW in length. However, if the user requires a Vendor Defined Message with a significant payload then it is unlikely that a CPU could generate the required AXI transaction as a single AXI burst, and they would have to implement custom logic to generate the AXI bursts associated with Vendor Defined Messages.

The following table describes the register allocation when used for Message TLPs.

Table 17-9 Outbound TLP Register Bank - Message TLPs

Register	Bits	Allocation
desc3	31:0	Reserved
desc2	31:0	Reserved
desc1	31:0	PCIe header descriptor[63:32]
desc0	31:0	PCIe header descriptor[31:0]
addr1	31:0	Bits [127:96] of vendor defined message. Should be programmed to zero for normal messages.
addr0	31:28	Bits [95:72] of vendor defined message. Should be programmed to zero for normal messages.
	7:0	Reserved.

Table 17-10 PCIe Header Descriptors with description

PCIe Header descriptor	Description
[3:0]	Transaction type: 1100 = Normal message; 1101 = Vendor-Defined message; All other values are reserved.
[6:4]	PCIe Attributes associated with the request.
[7]	Reserved for future use.
[15:8]	For vendor defined messages, this field carries bits [71:64] of the message header. For all other requests, this field is reserved.
[16]	Reserved for future use.
[19:17]	PCIe Traffic Class (TC) associated with the request.
[20]	Reserved for future use.
[21]	Force ECRC insertion. Setting this bit to 1 force the core to append a TLP Digest containing ECRC to the TLP, even when ECRC is not enabled for the Function generating the request.
[23]	This input enables the client to supply the bus and device numbers to be used in the requester ID. If this input is 0, the core will use the captured values of the bus and device numbers to form the Requester ID. If this input is 1, the core will use the bus and device numbers supplied by the client on bits [39:32] and [31:27] to form the Requester ID. This bit must always be set while originating requests in the RC mode, and the corresponding Requester ID must be placed on bits [39:32].
[22]	Reserved for future use.
[31:24]	PCI Function Number associated with the request. In the ARI mode, all the 8 bits are used to indicate the requesting Function. In the legacy mode, only bits [26:24] are used, and bits [31:27] are used to specify the device number to be used within the Requester ID, when bit [23] is set.
[39:32]	When bit [23] is set, this field must specify the bus number to be used for the Requester ID. Otherwise, this field is ignored by the core.
[55:40]	Reserved for future use.
[63:56]	Reserved for future use.
[71:64]	8-bit Steering Tag for the Hint.
[73:72]	Value of PH[1:0] associated with the Hint.
[75:74]	Reserved.
[76]	Set when the request has a Transaction Processing Hint associated with it.
[79:77]	Reserved.

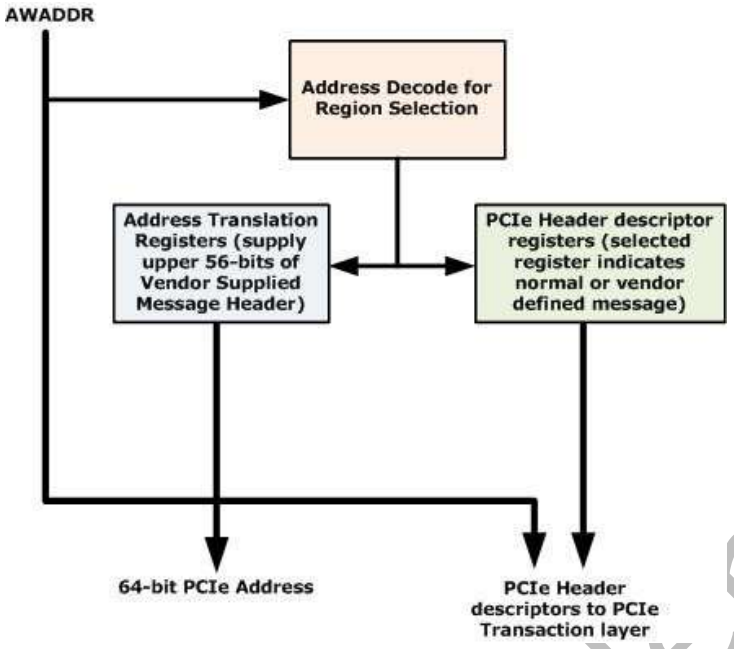


Fig. 17-16 Outbound Message Write Generation

17.5.5.2 Inbound Address Translation Application

17.5.5.2.1 Root Port Address Translation

Inbound address translation is performed on memory and IO TLPs. The selection of which address translation registers to use in the translation process is dependent on the BAR match of the incoming TLP. In Root Port mode there are 2 bars, so only 2 sets of registers are implemented (if BAR checking is disable, 3 sets of register should be concern). A set of registers corresponding to one BAR is shown in the table below

Table 17-11 Inbound Ordering

Register	Bits	Allocation
addr1	31:0	Address bits applied to ARADDR[63:32] and AWADDR[63:32], depending on addr0[5:0], when transaction is generated to SoC.
addr0	31:8	Address bits applied to ARADDR[31:8] and AWADDR[31:8] depending on addr0[5:0], when transaction is generated to SoC.
	7:6	Reserved
	5:0	Number of Address Bits passed through from HAL2AXI to SoC. Wrapper passes programmed value +1 bits.

The translation process for write and read TLPs is shown in Figure “Root Port Inbound Write Access Address Translation” and Figure “Root Port Inbound Read Access Address Translation”

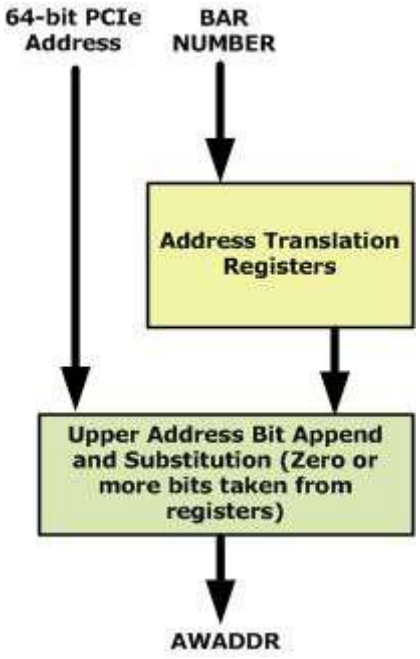


Fig. 17-17 Root Port Inbound Write Access Address Translation

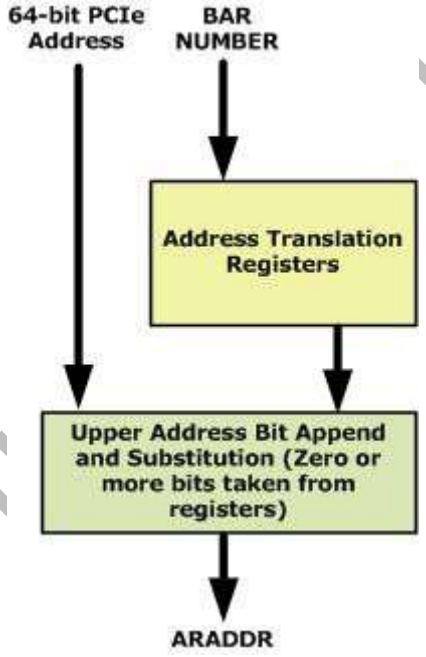


Fig. 17-18 Root Port Inbound Read Access Address Translation

17.5.5.2.2 End Point Address Translation

The address is modified with values programmed in the AXI address translation registers, please see Section "Address Translation Registers" for reference.

Based on the function and bar number of a request, the application layer of the PCIe Core translates the address from the PCIe address to the SoC address making use of the base address given in the address translation registers. In current PCIe module, only one physical is implemented.

17.5.5.3 Usage of Base Address Registers(BARs)

For an RC/EP port, the use of BAR register is applicable only when itself has a memory that can be targeted from the PCIe link. In a normal system setup, the system memory would be connected on the host/CPU bus. This host/CPU is the initiator/completer for TLPs. To access a downstream device or upstream port, the host/CPU would initiate a request that is forwarded to remote target. Completions (if any), for this request are given to the host/CPU application. If a request with an address within the range of the target BAR space. That TLP will be forwarded to the host/CPU interface. In RC mode, the software can

disable BAR checking by clearing the RC BAR Check Enable bit. When this bit is set to 0, the core will forward all incoming memory requests to the client logic without checking their address ranges. The BAR n Aperture setting decides the writable bits of the BARs.

Detail register description, please refer to register description section "Root Complex Base Address Register 0/1", "Base Address Register n", "Physical/Virtual Function BAR Configuration Register", "Root Complex BAR Configuration Register"

17.5.6 PCIe DMA

17.5.6.1 Introduction

The PCIe module contains the uDMA core integrated with the existing PCIe Controller and AXI bridges to provide a PCIe-AXI bridge with integrated DMA.

The existing bridges provide a means of translating transactions on the PCIe bus to equivalent transactions on the AXI bus and vice-versa. The DMA functionality adds a method of initiating and maintaining transfers between the two memories: the local memory accessed by the AXI infrastructure (“System Memory”) and remote memory in the PCIe link partner (“External Memory”).

In Current implementation, the DMAC module also can initiate transaction through AXI salve interface instead of master. This operation usage is similar with CPU access normally. Please see the DMAC section for detail usage.
This section introduces the uDMA usage only within the PCIe Core module.

17.5.6.2 Basic DMA Operation

The DMA Operation is controlled by a linked list of transfer descriptors maintained in local system memory. These descriptors describe the details such as, addresses in the System and External memory, length and attributes of the transfer (e.g. Memory, IO or Configuration, Requestor ID etc… for PCIe, Memory types and Trust Zone for AXI).

The firmware running on the local processor is expected to generate the linked-list and program the DMA.

The DMA is configured to contain a number of independent DMA Channels. Each Channel is programmed with the address of the linked list and initiated. The DMA Channel has access to a local dual-port RAM to store the data being transferred.

The DMA Channel fetches the first descriptor in the linked list. Based on the information in the descriptor the transfer will either be inbound (i.e. from the remote system to the local system) or outbound.

The AXI attributes to fetch the descriptor are listed below:

- arbar = {1'b0, ar_attr[22]};
- arsnoop = ar_attr[18:16];
- ardomain = ar_attr[21:20];
- arregion = ar_attr[15:12];
- arqos = ar_attr[11:8];
- arprot = ar_attr[2:0];
- arcache = ar_attr[6:3];
- arlock = ar_attr[7];
- ar_attr is the concatenated value programmed in the 2 attribute registers.

Please see PCIe DMA Channel Attribute Lower/Upper Register in Section “PCIe DMA registers” for register descriptions.

17.5.6.2.1 Inbound transfer

For an inbound transfer, the Channel issues a read request onto the PCIe Link. The read request does not have understanding of any address or length restrictions on the interface to the PCIe Controller. The Channel issues the read request to the address programmed in the descriptor for a length that can be a value up to that value supported by its dual- port

RAM. At the bridge the read request is fragmented according to the PCIe rules. For example, the channel breaks a DMA transfer into individual 4 K-byte transfers – not necessarily on 4K boundaries. The bridge then fragments it further depending on MPS, MRRS and at 4K boundaries.

As each request made to the PCIe Controller is allocated a different PCIe Tag, the read data may be returned to the bridge in the following fashion:

- Data returned for the same PCIe request may be returned in several completions. The completions will be in address order for that request.
- A completion with data for a different PCIe request (even if it's for the same Channel request) may precede the first PCIe request.

As there is a requirement to avoid additional buffering in the bridge the data is returned to the Channel with this potential disorder. The Channel uses the dual-port RAM to correct the ordering. When it has got enough data accumulated, it can then issue the write request on the AXI Master interface. Again the Channel does not know the address and length restrictions on the AXI bus. The DMA fragments requests according to AXI bus address and length restrictions and aggregates any write responses.

17.5.6.2.2 Outbound transfer

For an outbound transfer, the reverse of inbound operation happens. Data is read through the AXI Master interface and transferred over to the PCIe Controller. On the AXI, multiple requests with the same tag (ARID) can be issued and the AXI ordering rules will keep the data in order. Therefore, data is maintained in order for outbound transfers. The PCIe Controller requires write data to be always available once the transfer has started and this ensures that a write request is not issued until the complete write data is in the dual-port RAM.

Once the write has been acknowledged the Channel considers that transfer to be complete. It then checks if there is another descriptor in the linked-list and fetch and execute that. On the last descriptor in the linked list, the Channel interrupts the local system firmware to indicate complete.

17.5.6.2.3 ECC width

If the PCIe Controller is configured with ECC, the DMA RAMs will also have ECC. ECC requires an additional 7 bits for every 32-bits of payload data. Selecting ECC imposes a restriction on DMA transfers that the length is a multiple of 4 bytes and addresses in both the System memory and Host memory are aligned on 4 bytes.

17.5.6.3 Features of the Universal DMA Engine

The Universal DMA Engine consists of the following features:

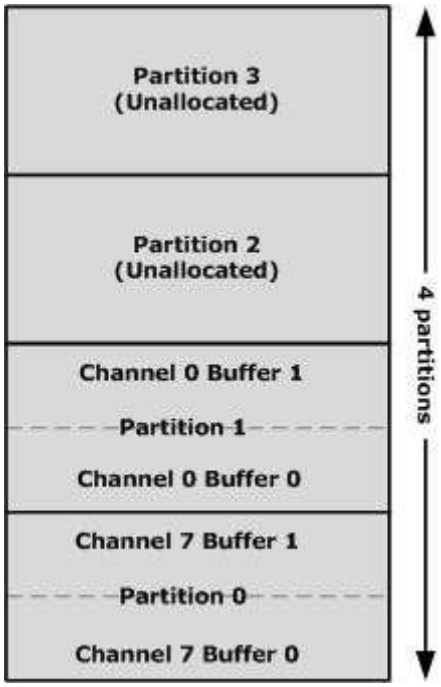


Fig. 17-19 Example Buffer Location

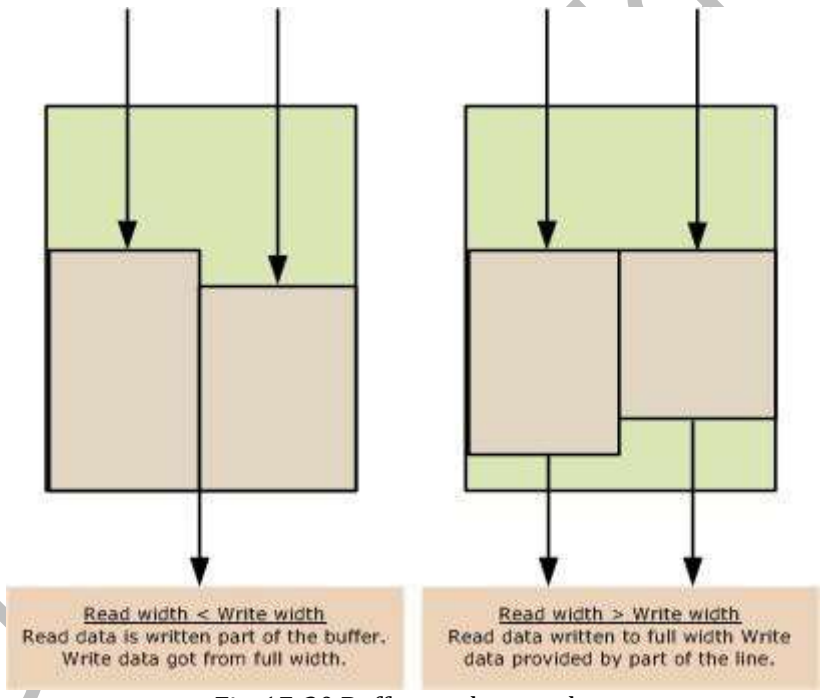


Fig. 17-20 Buffer used as gearbox

- Multiple channels - up to 8. The buffer is maintained in a dual-port RAM shared between channels. There are separate dual-port RAMs for inbound and outbound transfers.
- Each channel is controlled by a linked list of channel descriptors in the system memory.
  Note: Some descriptor fields are either 4 or 8 bytes. The total number of bytes required is optimised for the required configuration.

Each descriptor provides:

- 32...64-bit system bus address - 4 or 8 bytes.
- Configurable width system bus command attributes - 4 or (rarely 8) bytes.
- 32...64-bit external interface address - 4 or 8 bytes.
- Configurable width external interface command attributes - 4 or 8 bytes.
- Transfer length - 2 bytes.
- Control (function) - 1 byte.
  - Inbound transfer

  - Outbound transfer
  - Transfer continuity
- Status (separate status for system bus, external interface) - 2 bytes.
- Channel status - 1 byte, bit-significant indicating:
  - Transfer completed OK.
  - Invalid descriptor.
  - Uncorrectable buffer error.
- 32..64-bit system bus address of the next descriptor in the linked list - 4 or 8 bytes
- Each channel has separate control registers. These registers are accessed through an APB slave port.
  - 32...64-bit system bus address of head of linked-list.
  - Doorbell ("Go") and status.
  - Interrupt registers.
- The data widths of the system bus and external interface are configurable parameters, and need not be the same, though one will be two, four or eight times the other. In the initial version this will be limited to one or two times.
- There are two dual-port SRAMs: one for inbound transfers and one for outbound transfers. Each SRAM is arranged as a configurable number of partitions. Each partition will be allocated to a channel to use as two buffers. When a channel has been programmed with a descriptor list it can request one partition. In the figure below, with 4 partitions and 8 channels, Channel 7 has requested a pair of buffers, followed by channel 0.
- Each Channel can request data to fill the two buffers it has been allocated. When the first buffer is fully fetched then the data can be written. The buffers form both the gearbox between the two bus widths as well as the clock domain crossing between the system clock and interface clock. It also provides alignment with the first byte of the transfer aligned on byte 0 in the buffer, whereas the read or write data may have it aligned elsewhere. The number of cycles to read the SRAM will be a configuration parameter to allow a variety of SRAM implementations to be used.
- Data transfer will be supported in three modes (Scatter, Gather or Bulk).

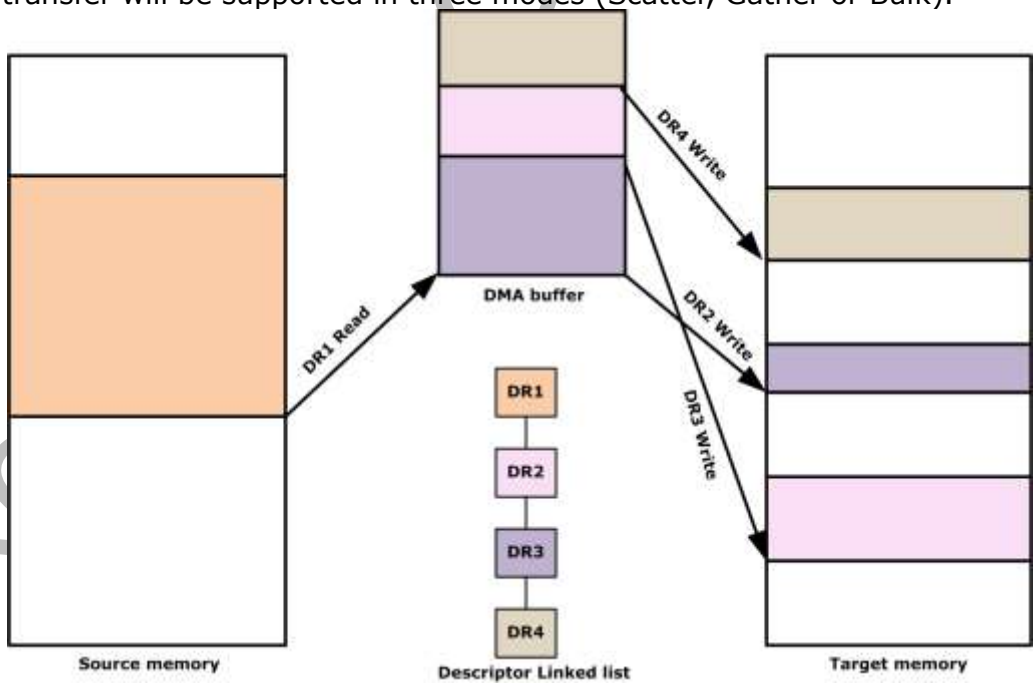


Fig. 17-21 Scattering data

The first descriptor is used to fetch the entire data and subsequent descriptors are used to write each set of data to target memory. The overall data size is restricted to RAM Partition size.

  - In Bulk mode, data is transferred from a single area in source memory to a single area in target memory. Each descriptor in the linked-list represents some data as part of this transfer. The overall data size is restricted to 224 bytes.
  - In Gather mode, data is transferred from a number of areas in source memory to a

single area in target memory – see the figure below.

- The initial descriptors in the linked list are used to fetch the separate sets of data from source memory: the last descriptor is used write all the data to target memory. The overall data size is restricted to RAM Partition size.

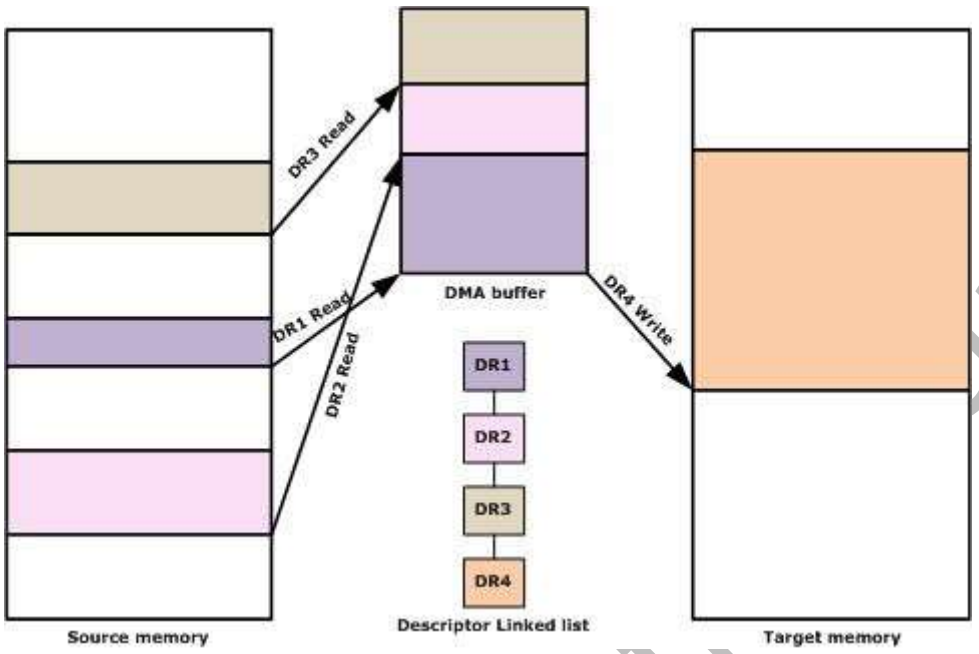


Fig. 17-22 Gathering data

- In Scatter mode, data is transferred from a single area in source memory to a number of areas in target memory – see the following image.

- The uDMA is completely agnostic to the contents of the attributes fields of the channel descriptor. It forwards these to the PCIe Controller.
- The uDMA is mostly agnostic to the value of the address fields of the channel descriptor. It forwards these to the appropriate AXI/PCIe Controller port. It is, however, aware of the value of the addresses when writing incomplete and disordered responses into the buffer.
- The uDMA is completely agnostic to the contents of the status field of the channel descriptor. It takes the attributes from the response on the appropriate AXI/PCIe Controller port.
- The uDMA has no understanding of any burst boundary requirements of either the external interface or the system bus. It is the responsibility of the appropriate bus bridge to break bursts as required by the bus architecture.
- Each channel drives the AXI ID field with a separate value, but consistently uses that value.
- The requests on each AXI/PCIe Controller port from all the channels are then arbitrated using round-robin to generate requests to the system bus and external interface.
- Data integrity is provided as a configurable option:
  - No error detection or correction.
  - Single bit error detects – byte parity.
- Data integrity is provided for the channel buffers. A byte parity or a single-bit correct, double-bit detect code will be used.
- If the channel detects a descriptor that is incompatible with the mode of transfer or is not consistent with other descriptors in the linked list it will write this information to the status field of the descriptor.
- If the channel detects an uncorrectable error while reading the buffer it will write this information to the status field of the descriptor.
- If the channel detects a correctable error while reading the buffer a register will be incremented with this fact.
- When the transfer is complete for the descriptor list, because all transfers completed successfully or a transfer completed unsuccessfully, a bit in the status register will be set and an interrupt is raised to the system processor.

17.5.6.4 Command Descriptor

The command descriptor is written to system memory. It provides a means of programming the DMA Channel. The size of the descriptor is optimized for the bus widths of the addresses and attributes of the system and external bus. The descriptors may be combined into a linked list in order to affect a complete transfer of data between system (SOC) memory and external (link partner).

For the DMA, the bus configuration is as follows:

Table 17-12 PCIe-AXI with DMA Command Descriptor

Displacement from descriptor base	Size (in bytes)	Bits	Description
0	4	31:0	AXI Base Address.
4	4		AXI Address Phase (AR or AW) controls.
		31:8	Reserved.
		7	AxLOCK[0]
		6:3	AxCACHE[3:0]
		2:0	AxPROT[2:0]
8	8	63:0	PCIe Base Address.
16	8		PCIe TLP Header Attributes.
		63:45	Reserved
		44	TPH Present
		43	Reserved.
		42:41	TPH ST Hint
		40	Reserved
		39:32	TPH Steering Tag
		31:26	Reserved
		25:10	Requester ID
		9	Use 25:10 as Requester ID
		8:6	Reserved
		5:3	PCIe Transfer Class
		2	ID-Based Ordering
		1	Relaxed Ordering
		0	No Snoop
		8:6	Reserved
24	3	23:0	Length of transfer in bytes (0 indicates maximum length transfer 2^24 bytes).
27	1		Control Byte
		7:6	Reserved
		5	Continue to execute linked-list
		4:3	Reserved
		2:1	Continuity: 00 - Read and write data for bulk operation 01 - Prefetch: read-only for scatter/gather operation 10 - Write data for scatter/gather operatio
		0	Interrupt after execution of descriptor
28	1		AXI Bus Status
		7:3	Reserved
		2	Internal data integrity error detected (when generating AXI request)
		1:0	BRESP[1:0] or RRESP[1:0]
29	1		PCIe Bus Status

Displacement from descriptor base	Size (in bytes)	Bits	Description
		7:4	Reserved
		3:0	TLP Completion status or internal error: 0000 - Normal Completion 0001 - Completion TLP was poisoned 0010 - Request termintaed by a completion with UR, CA or CRS status 0011 - Request terminated by a completion with no data 0100 - The current completion has the same tag as an outstanding request but its requester ID, traffic class or Attributes fields do not match 0101 - The low address bits of the completion.TLP header do not match the starting address of the next expected byte. 0110 - The tag of the current completion does not match that of any outstanding request 0111 - Request terminated by completion timeout or by an FLR targeted at the function which generated the request 1000 - Internal error: returning byte count of the completion does not match the expected 1001 to 1011 - Reserved 1100 - Internal data integrity error detected (when generating controller request) 1101 to 1110 - Reserved 1111 - Multiple error conditions detected
30	1		Channel Status
		7	Buffer Not Empty: There is outstanding data in the buffer at the end of executing a linked-list.
		6	Buffer Underflow: When scattering data more data is required than the size of the buffer.
		5	Buffer Overflow: When gathering data more data is required than the size of the buffer.
		4	Descriptor Error: An invalid decode of the descriptor was detected.
		3	Data Integrity Error: Internal data integrity detected (when accessing internal RAMs).
		2	AXI transfer completed early with incomplete data.
		1	PCIe transfer completed early with incomplete data.
		0	Descriptor action completed.
31	1	7:0	Reserved
32	4	31:0	Pointer to Next Descriptor in linked-list

17.5.6.5 Firmware Use Model

This section describes the firmware actions required to support the use of the Universal DMA Engine. This covers initialization requirements, the establishment of a linked list of descriptors, the programming of an individual channel and handling of the status of transfer completions.

17.5.6.5.1 Initialization

Firmware needs to query the Universal DMA hardware to find out the hardware version and its capabilities and configuration.

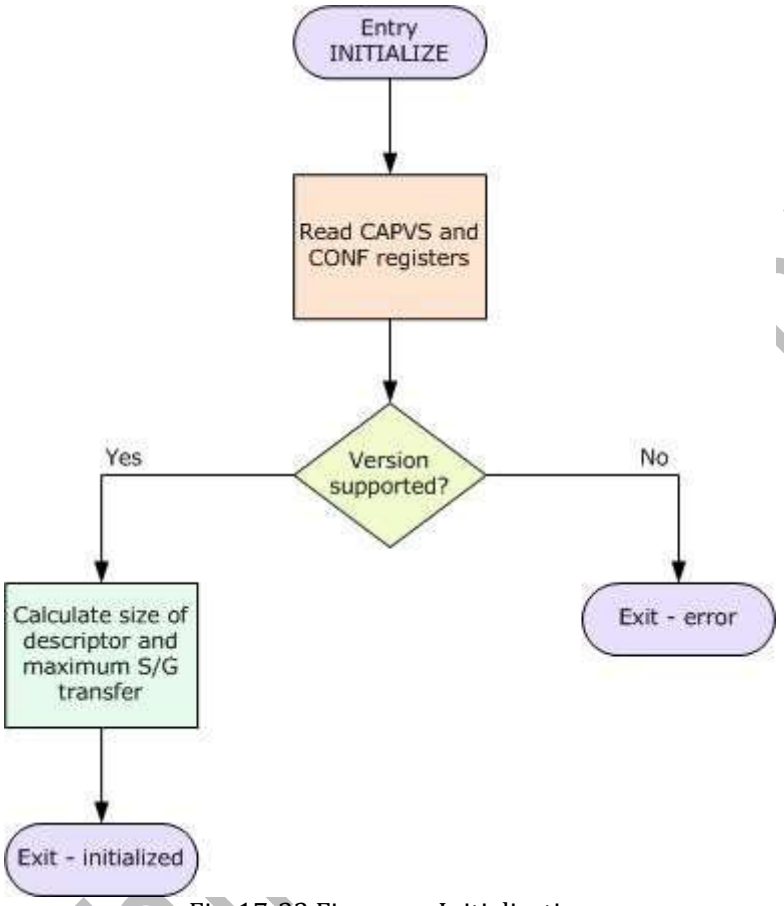


Fig. 17-23 Firmware Initialization

Register CAPVS provides the version and capabilities of the hardware. The current version supported by the algorithms in this section is 1.0 (major version 1, minor version 0). There are no optional capabilities defined at this version.

Register CONF provides the configuration of the hardware. This provides information that the firmware requires:

- Overall size and field displacements of the command descriptor.
- The maximum transfer size for Scatter or Gather Transfers is limited to a single RAM partition = 128 * 2CONF.PZ.
- The number of DMA Channels provided by the hardware = CONF.NUMC.
- The number of partitions per RAM provided by the hardware = CONF.NUMP. Firmware can make use of this information in order to not request execution of a transfer if the RAM resource is unavailable: however, for firmware to make the request, the hardware waits until the resource becomes available.

In addition firmware needs to enable all the interrupts it expects to handle. There is a mask register which resets to all interrupts disabled. Firmware needs to write to a bit-significant enable register to enable individual interrupts. There is also a bit-significant disable register to disable individual interrupts.

17.5.6.5.2 Creating Linked Lists

When setting up the Universal DMA to execute a transfer, the firmware needs to create a

linked list of descriptors. There are three types of transfers supported by the Universal DMA:

- Bulk Transfers: Data is transferred between system memory and external memory. The individual data buffers in both the memories are large compared with the size of the individual link data transfers (and a single RAM partition).
- Scatter Transfers: Data is read from a single data area and written to a number of data areas. Typically, this will be used for reading from the external memory and writing to the system memory to minimize the number of link transfers required. The overall amount of data is restricted to a single RAM partition.
- Gather Transfers: Data is read from multiple data areas and written to a single area. Typically, this will be used for reading from the system memory and writing to the external memory to minimize the number of link transfers required. The overall amount of data is restricted to a single RAM partition.

In order to create the linked list, firmware can use three descriptor types:

- A Read/Write descriptor: The entire data is transferred between the two memories. The descriptor can be used to transfer up to 224 bytes of data. The hardware channel will request data transfers in quanta of a RAM buffer, with two outstanding transfers at any one time. The RAM partition allocated to a channel is divided into the two buffers required to support this.
- A Pre-fetch descriptor: The data specified in the channel descriptor is read from the source memory and written into the RAM partition allocated to the channel. If there is not enough space in the RAM partition to hold the data being read, this is considered an overflow condition and the appropriate status bit written back to the descriptor.
- A Post-write descriptor: The data specified is written to the target memory from the RAM partition allocated to the channel. If there is insufficient data in the RAM partition to provide all the data for the write, this is considered an underflow condition and the appropriate status bit written back to the descriptor. If on the last descriptor of a linked- list there is still data in the RAM partition, this is considered a "not empty" condition and the appropriate status bit written back to the descriptor.

When building a linked-list the Next Descriptor Pointer field of the descriptor will point to the next descriptor in the list. In the last descriptor of the list the Next Descriptor Pointer will be zero. In descriptors other than the last in the list the control bit "continue" is set. In the last descriptor the "continue" bit is cleared.

Any descriptor can request that an interrupt occurs by setting the "interrupt" bit in the control field. This will result in the done interrupt occurring when the transfer is correctly executed.

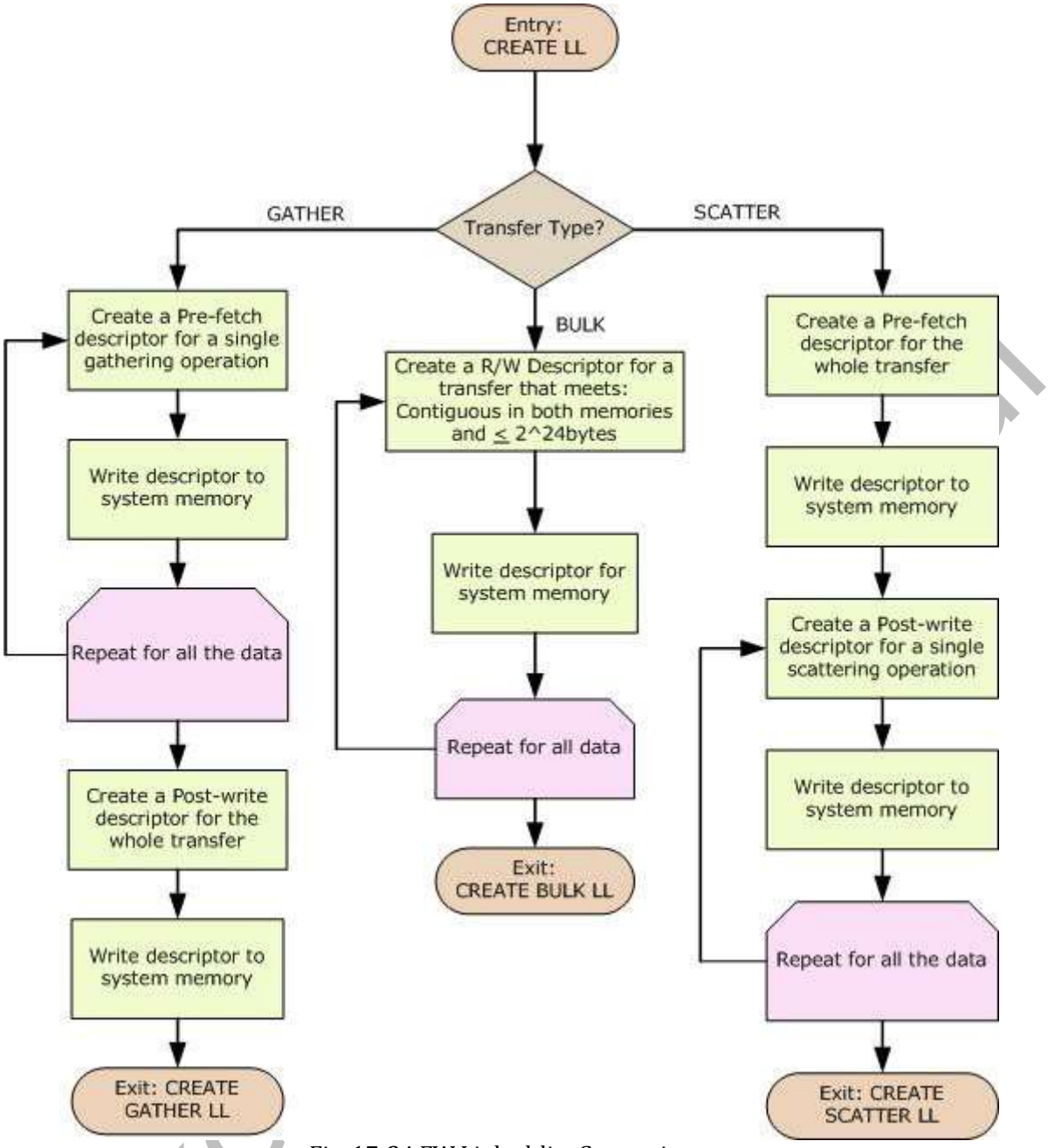


Fig. 17-24 FW Linked-list Generation

17.5.6.5.3 Bulk Transfers

A Bulk Transfer describes the transfer of a large amount of data between system memory and external memory. Firmware needs to understand how the data areas in the system memory and external memory map to each other, and break down into individual transfers that meet the following three criteria:

- The data is contiguous in the system memory;
- The data is contiguous in the external memory;
- The data is less than or equal to the maximum definable in a descriptor (224 bytes).

See the center branch in the figure "FW Linked-list generation".

17.5.6.5.4 Scatter Transfers

A Scatter Transfer describes the transfer of a small amount of data between system memory and external memory. Data is in a single contiguous area in the source memory (typically the external memory) and is scattered across several non- contiguous areas of the destination memory (typically the system memory). See the right branch in the figure "FW Linked-list generation".

17.5.6.5.5 Gather Transfers

A Gather Transfer describes the transfer of a small amount of data between system

memory and external memory. Data is gathered from several non-contiguous areas of the source memory (typically the system memory) and written to a single contiguous area of the destination memory (typically the external memory). See the left branch in the figure "FW Linked-list generation".

17.5.6.6 Appending Linked Lists

During the execution of a Bulk Transfer linked list, firmware can extend it by creating new descriptors and then writing the Next Pointer field and setting the "continue" bit of the command field in the last descriptor of the list being executed.

It then writes the start pointer register and set the go for the command. Hardware ensures that multiple writes to the start pointer are disabled if the first has not been acknowledged by the channel. Note that it is not possible to change the direction of transfer or the system bus attributes register for the descriptor once a channel has started executing: the result of doing so is undefined. Because a linked list describes a single Scatter or Gather operation, extending linked-lists for Scatter or Gather transfers will not be supported and operation is not defined.

This feature requires that firmware can update a descriptor while the hardware is potentially reading it. This can result in inconsistencies where the hardware and firmware do not have the same image of the last descriptor of a linked list. The views may differ because:

- Hardware reads the descriptor as a single transfer. However, the system fabric may split this into a number of smaller accesses.
- Firmware has no control over the way the system fabric handles writes to the descriptor, either when it is initially writing a complete descriptor or when it is updating the control and next pointer fields when appending to a linked list.
- It can be assumed that when firmware writes to the hardware registers, hardware will see these writes in the order firmware issued the writes, thus when hardware sees the go signal, and then the register sp will have the intended value.

Also hardware reads the descriptor and then executes the transfer. When it completes the transfer it may have an out-of- date value of the control and next-pointer values however go and spare both valid. See the table below for a description of each scenario.

Table 17-13 Linked-list consistencies

cont	NP	go	sp	Situation	Hardware Action
0	0	0	-	HW has read the last entry in a linked-list. It has not detected any action by FW to update it.	Go IDLE to wait for FW to set go which will indicate either an update to the linked list or the start of a new linked list.
0	0	1	SP	When HW read the DR it was the last in the linked list. FW has either extended it or written a new list.	Start reading the descriptor pointed to by SP. Clear go.
0	!=0	0	-	FW was in the process of updating the last entry of the linked list when HW read it at the start of the transfer. FW has not yet written go even though the HW has executed the transfer.	Go IDLE to wait for FW to set go to complete the update of the linked list.

cont	NP	go	sp	Situation	Hardware Action
0	!=0	1	SP	FW was in the process of updating the last entry of the linked list when HW read it at the start of the transfer and has now completed it by setting go.	Start reading the descriptor pointed to by SP. Clear go.
1	0	0	-	FW was in the process of updating the last entry of the linked list when HW read it at the start of the transfer. FW has not yet written go even though the HW has executed the transfer. Note that HW has seen the update to "continue" and not the update to "NP": this would be caused by a problem with the order in which FW writes are observed by the HW and should be impossible.	Go IDLE to wait for FW to set go to complete the update of the linked list.
1	0	1	SP	As above, but now FW has completed the update by setting go.	Start reading the descriptor pointed to by SP. Clear go.
1	NZ	0	-	Either FW was in the process of updating the last entry and has not yet set go, or this is not the last entry of a linked-list.	Start reading the descriptor pointed to by next pointer.
1	NZ	1	SP (SP = NP)	FW has updated the last entry and set go. HW has processed the last entry.	Start reading the descriptor pointed to by next pointer or SP (they are the same).
1	NZ	1	SP (SP != NP)	FW has updated the last entry and set go. HW has processed the last entry that is not the last.	Start reading the descriptor pointed to by next pointer.

17.5.6.7 Programming a Channel to Execute a Transfer

Once a linked-list is written into system memory, a channel is programmed to execute it. Based on the information enumerated during initialization, firmware allocates a specific channel on which the transfer needs to be executed. Internal arbitration between the channels to access the internal RAM and the system and external bus are scrupulously fair, so firmware can arbitrarily allocate the channel to be used.

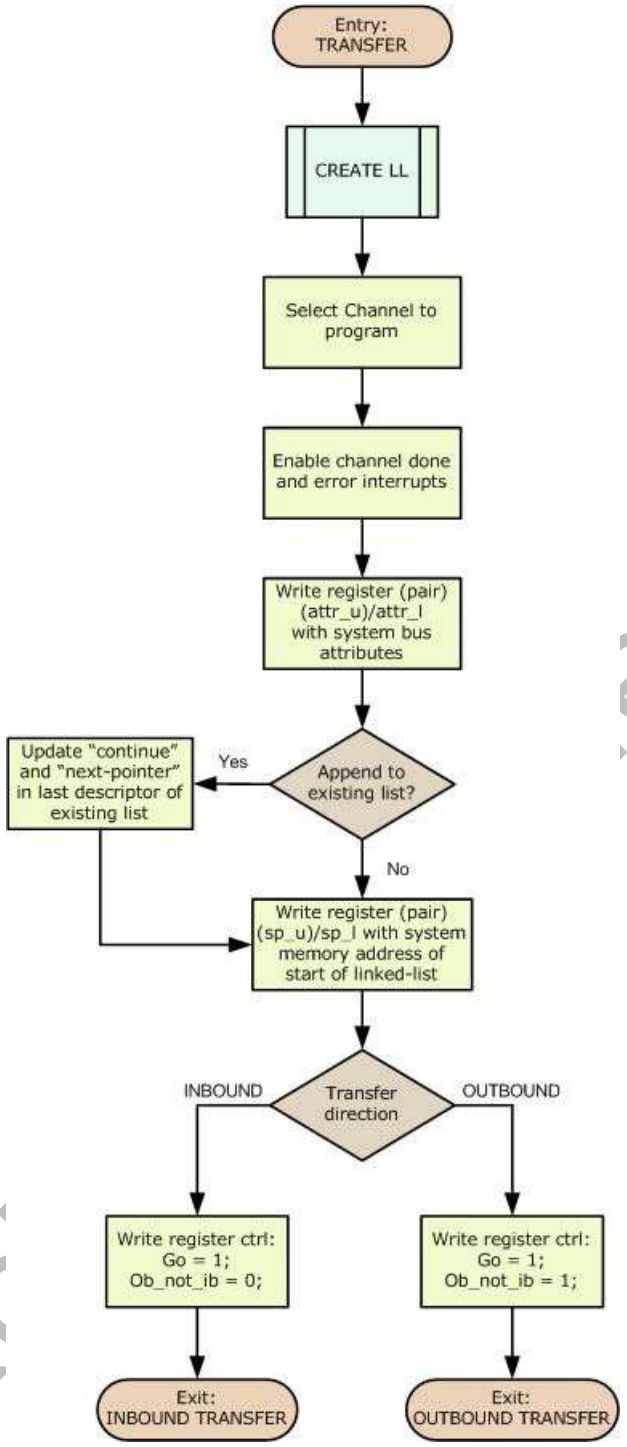


Fig. 17-25 Programming a channel

When the system attribute bus is > 32-bits wide the firmware needs to program both attr_u and attr_l registers for the selected channel, otherwise just attr_l is programmed. These registers define the system bus attributes to be used when fetching all the descriptors of the linked-list and when writing back the status value. When the system address bus is > 32-bits wide the firmware needs to program both sp_u and sp_l register s for the selected channel, otherwise just sp_l is programmed.

When programmed to transfer the data, the firmware must not access any register (except as indicated in Appending Linked Lists) for that channel until the channel responds by setting one of the done or error interrupt.

17.5.6.8 Handling Transfer Complete

The channel indicates the firmware that it has finished processing the transfer by setting either done interrupt or error interrupt. In both cases the firmware should read the status

fields of each descriptor in the linked list to ascertain the progress that the transfer made. The firmware can stop scanning the linked list at the first descriptor whose channel status field is zero, indicating that the transfer has not been attempted – i.e. has neither completed nor contains error in some way.

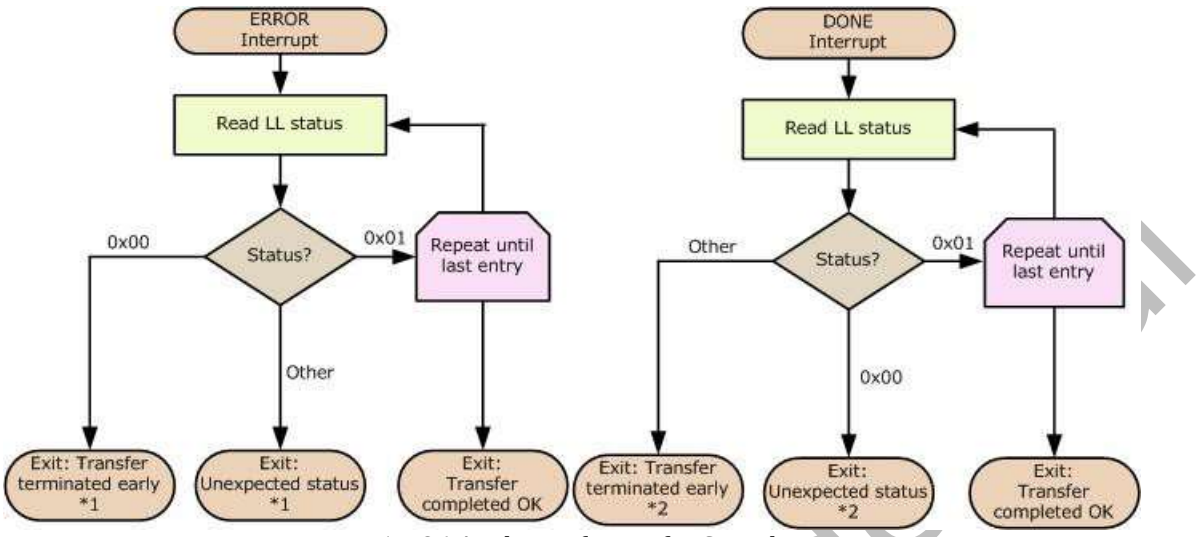


Fig. 17-26 Analysis of Transfer Complete

The cases of an early termination of a transfer indicate:

- The transfer may have successfully completed several descriptors – indicated by the count on the previous loop – and then failed to fetch the descriptor for the next transfer. No indication of the reason for this fail is possible.
- The transfer may have successfully completed several descriptors – indicated by the count on the previous loop – then failed to complete the next transfer. The reason for this is indicated by the remaining bits of the channel status field:
  - Bit 1: The external bus responded with an early completion – read the external status field for the specific reason.
  - Bit 2: The system bus responded with an early completion – read the system status field for the specific reason.
  - Bit 3: There was a data integrity (parity) error: either accessing the RAM or either system or external bus.
  - Bit 4: The descriptor used was invalid.
  - Bit 5: Buffer Overflow: More data was read than there is room for in the RAM partition.
  - Bit 6: Buffer Underflow: The post-write descriptor attempted to write more data than had been fetched in the previous pre-fetch descriptors.
  - Bit 7: Buffer Not Empty: The last post-write descriptor failed to empty the RAM partition.

In addition, the following failure case needs to be added:

- The transfer may have successfully completed several descriptors – indicated by the count on the previous loop – and on the next transfer the write-back of status to the descriptor failed.
- The cases indicating unexpected status should not occur:
  - In this case the descriptor has not been fetched but the status field has been written back to it.
  - In this case the transfer has been reported complete by done interrupt but the status field has not been written back.

Neither cases can be a result of the Universal DMA alone accessing the status field; some other system component must have interfered with the content of the descriptor in system memory.

17.5.6.9 Interrupt Service Routine

When an interrupt is received from the uDMA, the interrupt status is read and the appropriate routine is followed as described in Section 5.3.8 for each bit set in the register. This register should then be written back with the same value to clear all handled interrupts. Note: A second interrupt should not be received from any channel until it is re-programmed for the next transfer.

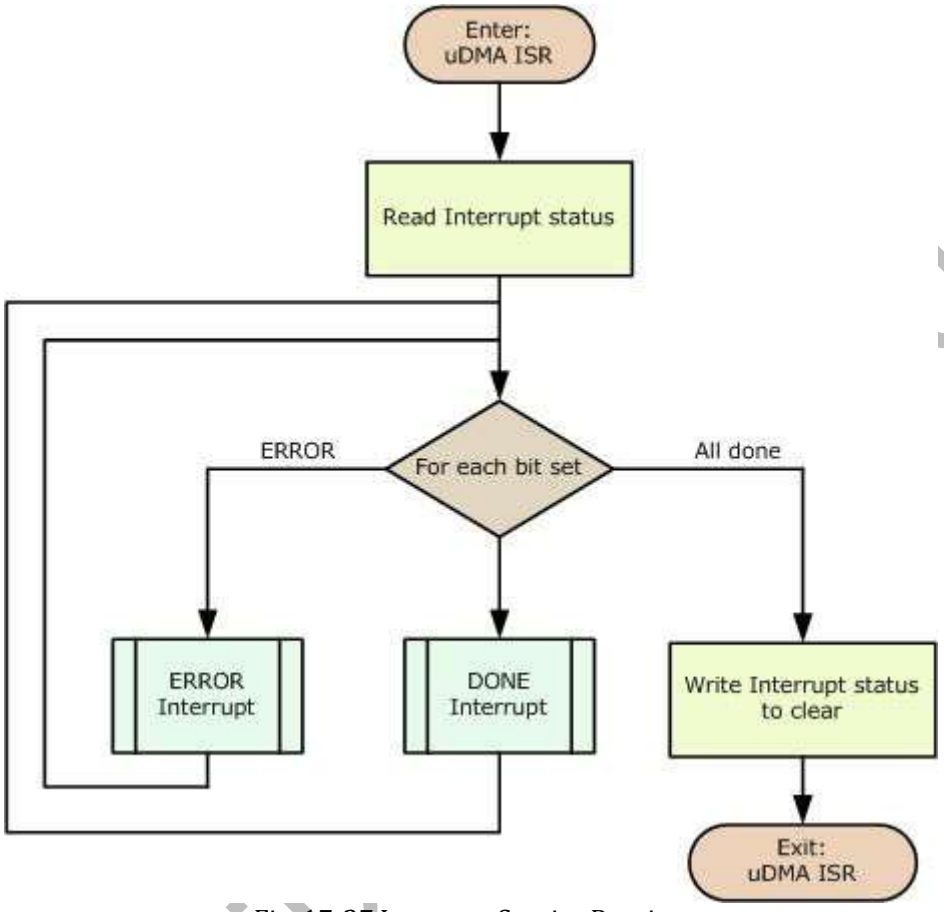


Fig. 17-27 Interrupt Service Routine

17.5.7 PCIe Transaction Requirements

The PCIe permits a client Endpoint application to initiate PCI transactions as a bus master across the PCIe link to the host memory. For Root Ports, it is also used to initiate I/O and configuration requests. For Endpoints, it needs to be connected to client logic only when the client has bus master capability. This PCIe can also be used by both Endpoints and Root Ports to send messages on the PCIe Link.

Some requirements that must be adhered to while initiating transactions to the PCIe internal bus interface. The client logic must check the following conditions before making a request.

- Only Root Ports can initiate I/O or Configuration Requests.
- The PCIe module operates in little-endian mode.
- An Endpoint can initiate a memory read or write request only when the Bus Master Enable bit of the PCI Command Register associated with the requesting Function is set. These bits are accessible at the Client BASIC_STATUS1[fc_st] or Core i_command_status[BE].
- An Endpoint can send requests only when the power state of the Function originating the request is D0-Active. The power state of each Function is available at the Client POWER_STATUS[fc_pwr_st].
- The originating Function is not currently processing a Function-Level Reset (when FLR is implemented in the core).
- Transactions only support incrementing burst type. It doesn't support fixed or wrap burst types. Issue an unsupported transaction will result in unexpected behavior.

- Neither the PCIe bus master nor slave can support the bus interleave operation. This means the write/read data interleaving depth of the core is 1. Client logic must complete data transaction for a particular request before another request.

17.5.8 PCIe Operation

17.5.8.1 PCIe Initialization Sequence

17.5.8.1.1 PCIe as Root Complex

The initialization sequence is as follows:

1. Configure the GPIO configure for CLKREQ_IN_N, please see section "Interface Description".
2. Configure the CRU_CLKSEL18_CON, provide reference clock to PCIe PHY and select the PCIe Core clock source. If the PCIe PHY reference clock is selected from internal PLL's output (100MHz reference clock), the software should configure the CRU_NPLL_CON3 and CRU_CLKSEL17_CON. Please refer to CRU register section for more detail description.
3. Set BASIC_STRAP_CONF to relative values in the PCIe Client level register to operate the PCIe in EP mode. The software can also select the PCIe generation support and lane count support except the link training enable (the link_train_en is default disable after power-up reset).
4. De-assert the PCIe PHY reset by setting CRU_SOFTRST8_CON [7] to enable the PCIe PHY PLL internally, then Wait until PLL is locked by polling the SIDE_BAND_STATUS[phy_st] bit.
5. If the remote device requests the common differential reference clock, the software should configure the PCIe PHY internal register bits through PCIe PHY configuration interface, the software can enable the PCIe PHY provide the 100M differential reference clock (optional with SSC) to remote device. For detail configuration descriptions, please see the PCIe PHY Configuration section.
6. De-assert the PIPE_RESET_N/MGMT_STICKY_RESET_N/MGMT_RESET_N/RESET_N simultaneously.
7. Program the configuration registers in the PCIe to desired values.
8. Initiate link training can be initiated by asserting BASIC_STRAP_CONF[link_train_en] bit. Insure link training completion and success by observing link_st field in PCIe Client BASIC_STATUS1 register change to 0x11. If both side support PCIe Gne2 speed, re-train can be Initiated by asserting the Retrain Link field in Link Control and Status Register. The software should insure the BASIC_STATUS0[negotiated_speed] change to "1", that indicates re-train to Gen2 successfully.
9. In conjunction with the system software, start bus enumeration and setup configuration space on downstream ports, and then continue software handshake and initialization on the remote devices.

17.5.8.1.2 PCIe as Endpoint

The initialization sequence is as follows:

1. Configure the GPIO configure for CLKREQ_IN_N, please see section "Interface Description".
2. Configure the CRU_CLKSEL18_CON, provide reference clock to PCIe PHY and select the PCIe Core clock source. If the PCIe PHY reference clock is selected from internal PLL's output (100MHz reference clock), the software should configure the CRU_NPLL_CON3 and CRU_CLKSEL17_CON. Please refer to CRU register section for more detail description. Current PCIe module only support separate reference clock architecture if it works in EP mode.
3. Set BASIC_STRAP_CONF to relative values in the PCIe Client level register to operate the PCIe in RC mode. The software can also select the PCIe generation support and lane count support except the link training enable (the link_train_en is default disable after power-up reset). In EP Mode, it is advisable that set the conf_en to "0" to disable the remote configuration request, the core will generate CRS completion to in response to configuration requests until it is asserted.
4. De-assert the PCIe PHY reset by setting CRU_SOFTRST8_CON [7] to enable the PCIe PHY PLL internally, then Wait until PLL is locked by polling the SIDE_BAND_STATUS[phy_st] bit.

5. De-assert the PIPE_RESET_N/MGMT_STICKY_RESET_N/MGMT_RESET_N/RESET_N simultaneously.
6. Program the configuration registers in the PCIe to desired values. If no n
7. Initiate link training can be initiated by asserting BASIC_STRAP_CONF[link_train_en] bit. Insure link training completion and success by observing link_st field in PCIe Client BASIC_STATUS1 register change to 2'b11. If both side support PCIe Gne2 speed, re-train can be Initiated by asserting the Retrain Link field in Link Control and Status Register. The software should insure the BASIC_STATUS0[negotiated_speed] change to "1", that indicates re-train to Gen2 successfully.
8. If further configuration register initialization is required, the application request retry bit BASIC_STRAP_CONF[conf_en] should be keep de-assert. This will lead to incoming accesses to be responded with the retry response. This feature allows slow devices extra time before the root port assumes the devices to be inactive. Once programming is complete, assert conf_en bit to allow transactions from the root complex.
9. The PCIe can accept configuration setup access, after emulation done, the transactions can be initiated.

17.5.8.2 PCIe Wrapper Programming

The following table describes the register programming in AXI.

Table 17-14 Register Programming for AXI

Type of Packet	Memory/IO	Config	Vendor Defined Message	Normal Message
desc0(32bit)	PCIe_Descriptor [31:0]	PCIe_Descriptor[31:0]	{PCIe_Descriptor[31:16],VDH[71:64], PCIe_Descriptor[7:0]}	PCIe_Descriptor[31:0]
desc1(32bit)	PCIe_Descriptor [63:32]	{PCIe_Descriptor [63:56], 16'h0, PCIe_Descriptor [39:32]}	{MSG_CODE, PCIe_Descriptor[55:48], MSG_ROUTING, PCIe_Descriptor[39:0]}	{MSG_CODE, PCIe_Descriptor[55:48], MSG_ROUTING, PCIe_Descriptor[39:0]}
desc2(32bit)	TPH_VECTOR	TPH_VECTOR	TPH_VECTOR	TPH_VECTOR
addr0(32bit)	{PCIe_Base Address [31:8], 2'h0, Pass_bits}	{4'h0,BUS_NUM_DEVICE_NUM, 4'h0, 2'h0, Pass_bits}	{VDMH[95:72], 8'h0}	32'h0
addr1(32bit)	PCIe_Base_Address [63:32]	32'h0	VDM[127:96]	32'h0
SOC_Address (32bit/64bit)	SOC_Address[RS +NR-1:RS]= region number, Lower Address is the offset address.	SOC_Address[RS +NR-1:RS]= region number, Lower Address is the offset address.	SOC_Address[RS+NR-1:RS]= region number, Lower Address = 0. SOC_Address[16]= 0 for MSG with Data, SOC_Address[16] = 1 for MSG without data, SOC_Address[14:12] = MSG_ROUTING, SOC_Address[15:8] = TYPE1,TYPE0.	SOC_Address[RS +NR-1:RS]= region number, Lower Address = 0. SOC_Address[16]= 0 for MSG with Data, SOC_Address[16] = 1 for MSG without data, SOC_Address[7:5] = MSG_ROUTING, SOC_Address[15:8] = MSG_CODE.
Pass_bits(6 bits)	RS-1	RS-1	0	0

NOTE:RS - Region Size in bits , NR - Number of Regions, VDMH - Vendor Defined Message Header, MSG_CODE - Message Code, MSG_ROUTING - Message routing information

Table 17-15 TPH_VECTOR

TPH_VECTOR					
TPH_ST_TAG[15:8]	TPH_PRESENT	TPH_LENGTH	TPH_TYPE[1:0]	TPH_INDEX	TPH_ST_TAG[7:0]

Please refer to Section "Configuration Accesses", Section "Memory and IO Accesses" and Section "Message Access" for their respective PCIe_Descriptor.

17.5.8.2.1 Configuration Descriptor

The following is an example that describes the procedure to send configuration messages.

Address registers

- For config access, the translated address bits [11:0] is passed to the PCIe address fields.
- Addr0[5:0] should be 11.

Descriptor registers

The PCIe descriptor is formed by

- PCIe_Descriptor = {Desc_regs [max - 1:56], PCIe_address[27:12], Desc_regs[39:0]};
- Desc_regs = {desc3,desc2,desc1,desc0};
- Config - Bus Number and device ID driven from address, remainder driven from registers.

Config Legacy Mode

- bus number = PCIe_descriptor[55:48]
- device number = PCIe_descriptor[47:43]

function number = PCIe_descriptor[42:40]

Config ARI Mode
- bus number = PCIe_descriptor[55:48]
- function number = PCIe_descriptor[47:40]

Please refer to PCIe_Descriptor fields for setup of Desc_regs.

Example:
A configuration is composed of 34 bits as region size and it has 32 regions in total. How do we send a config packet ?
- Method 1 (Bus number device number from registers)
  - Region setup
    Program the region 0 through the APB using the address given in userguide.
    Program the Addr0[5:0] as 11. Program the bits [27:12] of addr0 as follows:
    Config Legacy Mode
    - bus number = addr0[27:20]
    - device number = addr0[19:15]
    - function number = addr0[14:12]

    Config ARI Mode
    - bus number = addr0[27:20]
    - function number = addr0[19:12]
  - Region Access
    Pass the SoC address [11:0] as the config reg address. Pass the SoC address [38:34] as 0 (region select).
- Method 2 (Bus number device number from the AXI SoC address)
  - Region setup
    Program the region 0 through the APB using the address given in the User guide.
    Program the Addr0[5:0] as 27.
    In this case the translated address will be formed by using the lower 28 bits from the SoC address. Hence the Translated address [27:0] = SoC address [27:0].
  - Region Access
    Program the bits [27:12] of SoC address as follows:
    Config Legacy Mode
    - bus number = SoC address [27:20]
    - device number = SoC address [19:15]
    - function number = SoC address [14:12]

    Config ARI Mode
    - bus number = SoC address [27:20]
    - function number = SoC address [19:12]
    - Pass the SoC address [11:0] as the config register address.
    - Pass the SoC address [38:34] as 0. (region select)

You can combine Method 1 and Method 2 to make use of both features. For outbound transfer, the SoC address refers to the pcie_master_axi_AXADDR in the controller core.

For PCIe Header descriptors with their descriptions for configuration access, please see Section "Configuration Accesses".

17.5.8.2.2 Memory or IO Descriptor

Address registers
In this example we assume that the PCIe Controller itself occupies 2 MB of address space on the AXI SoC Bus. Each of the 8 Regions in this example is configured to consume 256Kbytes.
Let us assume that the PCIe Controller's SoC Starting Address is 32'h3CE0_0000; - Region 0 = 32'h3CE0_0000 to 32'h3CE3_FFFF; (256Kbytes).
Let us assume that the PCIe Host Address we want to write to is 32'hFF00_0000;
The Address Translation can be programmed to form the PCIe Address from a combination of the pre-programmed register value, and the incoming AXI address. We can take a maximum of 18 bits from the incoming AXI Address to pass through as the lower PCIe Host Address fields. This maximum is derived from the Region size of 256Kbytes.
Addr0 = {24'hFF00_00,2'h0,6'd17};

Addr1 = 32'h0;
- PCIe Host Address [63:32] - are all zero and come from Region 0 Addr 1 register
- PCIe Host Address [31:18] - come from Region 0 Addr 0 register [31:18]
- PCIe Host Address[17:0] - come directly from the AXI_AWADDR SoC address bus [17:0]

Descriptor registers
Please follow the User guide to setup the descriptors. Desc_regs= {desc3,desc2,desc1,desc0};

For PCIe Header descriptors with their descriptions for memory and IO access, please see Section "Memory and IO Accesses".

17.5.8.2.3 Message Descriptor

When a SoC AXI access is made to a region holding a descriptor for a message transaction then the PCIe Header fields are driven mainly from the register value. However, the Message Code and Message Routing fields are driven by bits [15:8] and [7:5] respectively of the incoming SOC_AWADDR.

SOC_AWADDR [16] = 1 Message with no Data SOC_AWADDR [16] = 0 Message with Data
Step by Step programming of register:
1. Set the PCIe Header Descriptor fields [63:32].
   Write to desc1 to set PCIe Header Descriptor 1 - "PCIe Header Descriptor [63:32]".
   - Bits [63:56] = Message Code
   - Bits [47:45] = Message Routing
2. Set the PCIe Header Descriptor fields [31:0].
   Write to desc0 to set PCIe Header Descriptor 0 - "PCIe Header Descriptor [31:0]".
   - Bits [2:0] = 3'b101 for Vendor Defined Message for example.
   - Bits [15:8] = Supply bits [71:64] of the Vendor Defined Message Header.
3. Set Bits [127:96] of the Vendor Defined Message.
   Write to Addr1 to set bits [127:96]
4. Set bits [95:72] of the Vendor Defined Message.
   Write to Addr0 to set bits [95:72].
5. Send Vendor Defined Message with Data:
   - Perform a AXI Write to Region 7, setting AWADDR [16] = 1'b0 to signify Message with Data
   - Supply the Vendor Data on the AWDATA Channel.

Note: It is critical that the AXI transaction for a message with data is implemented as a single AXI burst from the SoC. This is straightforward for normal messages with data as they are 1DW in length, AWLEN=0. However, if the user requires a Vendor Defined Message with a significant payload then it is unlikely that a CPU could generate the required AXI transaction as a single AXI burst, and they would have to implement custom logic to generate the AXI bursts associated with Vendor Defined Messages.

For PCIe Header descriptors with their descriptions for message access, please see Section "Message Access"

17.5.8.3 PCIe Message Handling

17.5.8.3.1 Message FIFO

The PCIe Core includes a dedicated interface for the output of inbound messages. The PCIe client processes the message by receiving it into a 32X128(depth 32, width 128bits) FIFO. Once message received done, an interrupt generates, and the effective length recorded in the MSG_DATA_LEN register. Currently, PCIe can support four message pending in FIFO. If the length is not equal to zero, it means a message had received. When CPU services the message receive done interrupt and read the MSG_DATA_LEN, the MSG_DATA_LEN is cleared automatically. The message process does not support back pressuring; the software must read out all the data based on the effective lengths. Because the APB bus read transaction bus width is 32bits and message write transaction bus width is 128bits, the FIFO read transactions number should be the multiples of 4, otherwise, it will lead to abnormal message processing. That is, if it's possible that the recorded effective length is not integral multiple of DWORD, so the software should add APB bus read dummy cycles to

complete the read FIFO transaction. The first cycle accesses the low DWORD of the 128bits message data and so on.

A software option named MSG_CTRL[msg_fifo_rx_mode] is added to message receive controller. If it asserted, all the message received. Otherwise, only the messages that message_code [7:0] (derived from PCIe link) matched with MSG_CODE0/1 are accepted. Due to the message is low frequency events, it's advisable for software to process the message receive done interrupt events. Below illustration shows the diagram of message receiver. In addition, if the legacy interrupt message can be received by message FIFO controller and the legacy interrupt is enable, PCIe client interrupt and Legacy interrupt will generate almost at the same time.

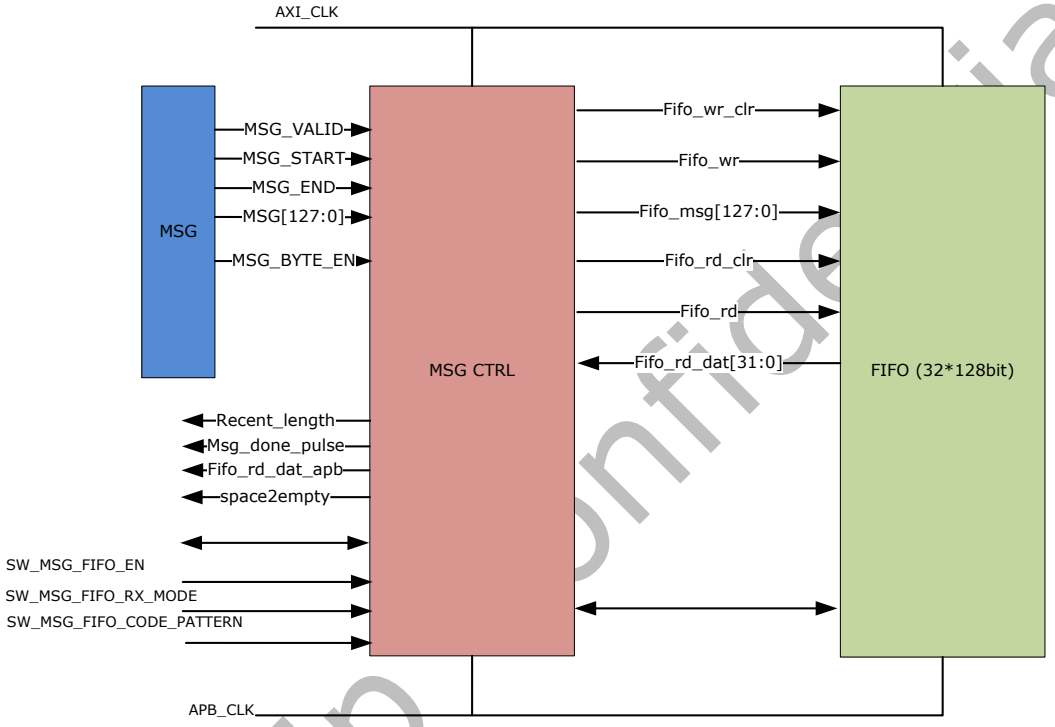


Fig. 17-28 Message Handling diagram

17.5.8.3.2 Message Header Elements

This section mainly describes the bit allocation of the message header, the software distinguish the message type based on the header. The message header will always occupy 64 bits, with an additional 64 bits for header bits [127:64] of a vendor defined message. The bit allocation of the message header is as follows:

Table 17-16 Message Header Bit Allocation

Bits	Allocation
63:52	Unused
51:36	Steering Tag
35:34	Processing Hint
33	1: 16-bit steering tag 0: 8-bit steering tag
32	TPH present
3:1	Attributes
31:24	Message code
23:8	Requester ID
7	Unused
6:4	Routing
3:1	Attributes

Bits	Allocation
0	1: vendor defined message 0: normal message

Message data (if present) always starts on a new stripe. Examples for different data bus widths are shown in the tables below.

Table 17-17 128-bit Message Interface, Normal Message with Data

Stripe	MSG_START	MSG_END	MSG_VDH	MSG_DATA	MSG[127:0]
Stripe 1	0	1	0	1	Data
Stripe 0	1	0	0	0	Header[63:0]

Table 17-18 128-bit Message Interface, Normal Message without Data

Stripe	MSG_START	MSG_END	MSG_VDH	MSG_DATA	MSG[127:0]
Stripe 0	1	1	0	0	Header[63:0]

Table 17-19 128-bit Message Interface, Vendor Defined Message with Data

Stripe	MSG_START	MSG_END	MSG_VDH	MSG_DATA	MSG[127:0]
Stripe 1	0	1	0	1	Data
Stripe 0	1	0	1	0	{Vendor Defined Header[127:64], Header[63:0]}

Table 17-20 128-bit Message Interface, Vendor Defined Message without Data

Stripe	MSG_START	MSG_END	MSG_VDH	MSG_DATA	MSG[127:0]
Stripe 0	1	1	1	0	{Vendor Defined Header[127:64], Header[63:0]}

17.5.9 Interrupt Support

The PCIe provides five types interrupt to system interrupt controller. It can be divided into MSI/MSI-X, Legacy interrupt, PCIe subsystem interrupt, PCIe client interrupt, PCIe wake up interrupt. When operating as RC, the PCIe is capable of handling both MSI/MSI-X and legacy interrupts. This is because when operating as RC it should be able to service both PCIe end points as well as legacy end points. It is capable of generating MSI or Legacy interrupt if the PCIe is configured to be EP. Notes that one PCIe component can't generate both types of interrupts. It is either one or the other. The interrupt type an EP generates is configured during configuration time. PCIe subsystem and client interrupt consist of multiple interrupts that generated by PCIe core and client directly, the software can inquire the interrupt status to acquire the interrupt events, besides, some interrupts generated by some deeper events, the software should clear the root cause to service the interrupt events. Some interrupt events are meaningful based on the role the PCIe assumes (RC or EP). Besides, if PCIe PHY is in P2 low power mode and de-assert electrical idle, it will generate a wake up interrupt to interrupt controller and their underlying events are listed below table.

Table 17-21 PCIe Interrupt table

System Interrupt Event ID	Interrupt description (level 1)	Interrupt subset (level 2)	Support mode	Note
81	PCIe subsystem interrupt	Power state change interrupt	EP only	Should acquire Interrupt subset status(level 3) if necessary, see more details from
		Hot plug interrupt	RC only	
		Phy interrupt	RC only	
		uDMA interrupt	RC and EP	

System Interrupt Event ID	Interrupt description (level 1)	Interrupt subset (level 2)	Support mode	Note
		Local interrupt	RC and EP	Core register description
82	PCIe Legacy interrupt	INTA interrupt	RC only	
		INTB interrupt	RC only	
		INTC interrupt	RC only	
		INTD interrupt	RC only	
83	PCIe client interrupt	Correctable error interrupt	RC and EP	
		Non-fatal error interrupt	RC and EP	
		Fatal error interrupt	RC and EP	
		DPA interrupt	EP only	
		Hot reset interrupt	RC only	
		Message receive done interrupt	RC and EP	
		Legacy interrupt send done interrupt	EP only	
169	PCIe wake up interrupt	PCIe P2 low power wake up interrupt	RC only	
XXX	MSI/MSI-X	message signaled interrupt	RC service and EP initiate	Allocate Event ID for PCIe device when emulation

When MSI/MSI-X handled in RC, the MSI/MSI-X TLP address should map to relative GIC address space, the Host CPU support GIC500 SPI or LPI interrupt control flow, detail software operation, please see GIC section and the MSI/MSI-X register description.

17.5.10 PCIe Power Management

The PCIe core supports both Active State Power Management (ASPM) and PCI Power Management, as described in Chapter 5 of the PCI Express Specifications. This chapter describes the details of the implementation. The following topics are covered in this chapter:

- Active State Power Management
- PCI power management
- L1 Power Management Substates

17.5.10.1 Active State Power Management

This version of the PCIe core supports ASPM L0s and ASPM L1.

ASPM L0s can be enabled/disabled in the "Link Control and Status Register" configuration register bit "Active State Power Management Control [0]". The Controller automatically initiates entry into ASPM L0s if the TX side is idle (i.e. No TLPs and No DLLPs to be transmitted) for a programmable time period. This time period is programmable through the Local Management Register "L0s Timeout Limit Register" The transition from L0 to L0s happens after a time period programmed in the "L0s Timeout Limit Register" elapses with No TLP or DLLP being transmitted. Setting the "L0s Timeout Limit Register" register to 0 disables the transition to L0s state.

In the EndPoint mode, the ASPM L0s must be enabled in the Link Control Register of the configuration spaces of all enabled Functions, for the transition to occur. In the Root Port mode, the ASPM L0s must be enabled in the Link Control Register of the Root Port Register Set of the core, for the link to transition to L0s to occur.

ASPM L1 can be enabled/disabled in the "Link Control and Status Register" configuration register bit "Active State Power Management Control [1]". The Controller automatically initiates entry into ASPM L1 if the TX side is idle (i.e. No TLPs from the Client and no replay

TLPs pending) for a programmable time period.

This time period is programmable through the Local Management Register "ASPM L1 Entry Timeout Delay Register". The field "L1 Timeout [19:0]" contains the timeout value (in units of 4 ns) for transitioning to the ASPM L1 power state. Setting it to 0 disables the transition to the ASPM L1 power state. The following sequence illustrates the operation of ASPM L1 Entry when the core is configured as an EndPoint.

1. When the Link TX is idle, (i.e. no transmit TLPs from client interface and no replay TLPs pending), the Controller starts incrementing the ASPM L1 entry timer internally.
   (1) If the Client requests to transmit a TLP, the timer is immediately cleared.
2. When the ASPM L1 entry timer reaches the programmed value of the "ASPM L1 Entry Timeout Delay Register", the Controller checks if sufficient credits are accumulated.
3. The Controller then blocks new TLPs and initiates ASPM L1 entry by transmitting "PM_Active_State_Request_L1" DLLPs onto its Transmit Lanes.
4. The Controller continuously transmits "PM_Active_State_Request_L1" DLLP until it receives a response from the Upstream device.
5. The Upstream component must immediately respond to the request with either an acceptance or a rejection of the request.
6. If the Upstream component rejects by sending a "PM_Active_State_Nak" Message, the Controller aborts the ASPM L1 entry and continues to send TLPs normally.
7. If the Upstream component accepts by sending "PM_Request_Ack" DLLPs, then the Controller (EP) puts its TX into electrical idle and enters ASPM L1.
8. The Upstream component also detects electrical idle and puts its TX also into electrical idle.

In the EndPoint mode, the ASPM L1 must be enabled in the Link Control Register of all the enabled Functions, for the transition to ASPM L1 to occur.

In the Root Port mode, the L1 power state must be enabled in the Link Control Register of the Root Port Register Set of the core, for the link to transition to ASPM L1 to occur.

17.5.10.2 PCI Power Management

This PCIe core supports PCI Function power states D0 (uninitialized and active), D1 and D3, and the corresponding link power states L0, L1 and L2.

17.5.10.2.1 PCI Power Management in EndPoint Mode

The following sequence illustrates the operation of PCI power management when the core is configured as an EndPoint.

1. Assume the core is operating normally, with all Functions in D0active state.
2. The remote Root Port writes into the Power Management Control Register of one or more Functions to transition the Function to the D3 power state.
3. When all enabled Functions in the core are in D3 state, the core initiates a transition of the link power state to L1 by transmitting PM_Enter_L1 DLLPs.
4. After the Data Link Layer handshake, the link transitions to L1 state.
5. The remote Root Port sends a PME_Turn_Off message to the core.
6. The core delivers the PME_Turn_Off message to the client logic through the AXI Message Interface.
7. When ready, client logic transmits the "PME_TO_Ack" message to the Root Port via the Client Master Interface. The steps for the Client logic to transmit "PME_TO_Ack" are described below:
   (1) Wait for a "PME_Turn_Off" message to be received on the Client Target Request Interface.
   (2) Read the function "Power State" from the configuration register "Power Management Control/Status Register".
   (3) Check the programmed value of "PME Turnoff Ack Delay [15:0]" in the local management register space.
   (4) If the "Function Power State" == "D0", OR, if the "PME Turnoff Ack

Delay"==0x0000, then the Controller does not transmit the "PME_TO_Ack" message. (Please see Note*).
1) Client firmware should ensure that there are no PCIe transfers active in the PCIe subsystem.
2) Client then sends a "PME_TO_ACK" message over the Client Master Request Interface.

(5) If the condition in (d) above is not true, then the Controller automatically transmits a "PME_TO_Ack" message after the "PME Turnoff Ack Delay" time.
1) The Client logic must not send the "PME_TO_ACK" in this case.

*Note:*If any enabled PF is in "D0" power state, then there may be PCIe transfers outstanding in the system for that PF. Hence the Controller does not automatically transmit "PME_TO_Ack" in this case.*

8. Client logic may now optionally change the power state of the core to L23_Ready by asserting the input POWER_CTRL[req_trn_l23ready]. This causes the LTSSM of the core to transition to the L2 power state, and enables the client to power down the core completely if desired. A power-on reset is required upon restoring power to bring the link back up to L0. The transition to the L2 state can be suppressed by keeping POWER_CTRL[req_trn_l23ready] de-asserted permanently.

By default, the core sets the No_Soft_Reset bit of the Power Management Control Register of all enabled Functions to1. This implies that the state of a Function is not lost when it is in the D3 power state, and its registers need not be re- configured when the Function is set back to D0. The PCI-SIG recommends setting this bit for all Functions.

While the link is in L1 state and all the core's functions are in D1 or D3 state, the link partner may transition the link any time from L1 to L0. The core may then optionally initiate a re-entry back to L1 if the link has been idle for a set interval and the core ‘s Functions still remain in D3 state. The re-entry to L1 is controlled by the delay programmed in the L1 State Re-entry Delay Register. Setting this register to a non-zero value causes the core to initiate entry back to L1 when a delay equal to the number of clock cycles programmed in this register has elapsed with no link activity. Setting this register to 0 prevents re-entry to L1. The initial transition to L1 (Step 3 above) is not affected by the setting of this register.

17.5.10.2.2 Wakeup Support

The PCIe Controller supports systems that use a wakeup mechanism.

The PME context of the PCIe controller shall be captured by the Client logic before sending the PME_TurnOff Acknowledge message to the Root Complex. The client logic shall specify the Requester ID of the PME message sent when power is re-applied to the controller and the link reaches L0. The following sequence describes the process of supporting WAKE#.

1. Assume the core is operating normally, with all Functions in D0active state.
2. The remote Root Complex writes into the Power Management Control Register of one or more Functions to transition the Function to the D3 power state.
3. When all enabled Functions in the core are in D3 state, the core initiates a transition of the link power state to L1 by transmitting DLLPs.
4. After the Data Link Layer handshake, the link transitions to L1 state.
5. The remote Root Complex sends a PME_Turn_Off message to the core.
6. The core delivers the PME_Turn_Off message to the client logic.
7. The client logic captured the PME Context of the PCIe controller, such as Requester ID. When ready, client logic transmits the ack to the PME_Turn_Off message to the Root Complex.
8. The client logic shall maintain the PME context using client application logic that is powered, either by Vaux or full power.
9. Client logic may now change the power state of the core to L23_Ready by asserting the input POWER_CTRL[req_trn_l23ready]. This causes the LTSSM of the core to transition to the L2 power state, and enables the client to power down the core completely if desired.
10. Some time later the client logic may decide to wake up the controller. The client logic

shall drive the WAKE# out-of- band signal to signal to the Power Management Controller that the controller requires power to be reapplied.

Note: The WAKE# signal is external to the PCIe Controller and is not used by the PCIe Controller.

11. Power is reapplied to the controller.
12. A hard reset is required upon restoring power to bring the link back up to L0.
13. The client logic then restores any PME Context to the controller registers via the local management interface.
    (1) Prevent the Controller from automatically transmitting PM_PME message by programming the bit-20 "disable_pme_message_on_pm_status" to '1' in Local Management Register "PME Service Timeout Delay Register".
    (2) The "disable_pme_message_on_pm_status" bit needs to be programmed to '1' before setting the "PME Status" bit to '1'.
14. The client logic then sends a PM_PME message over the client interface using the Requester ID that was captured before entering L2/L3.

The Root Complex may then perform a configuration write to the controller in order to move the device from the D3 state.

17.5.10.2.3 PCI Power Management in Root Port Mode

When the core is brought up as a Root Port, the following sequence illustrates the operation of PCI power management.

1. Assume the core is connected to a remote EndPoint, with all its Functions in D0active state.
2. The core writes into the Power Management Control Register of one or more Functions of the remote device to transition the Function to the D3 power state.
3. When all enabled Functions in the remote device are in D3 state, the remote device will initiate a transition of the link power state to L1 by transmitting PM_Enter_L1 DLLPs. The core responds to this by sending acknowledgments.
4. After the Data Link Layer handshake, the link transitions to L1 state.
5. Client logic may send a PME_Turn_Off message to the remote device through the AXI Message Interface.
6. When the remote device returns with the ack (PME_TO_Ack TLP) message, the core delivers it to the client logic through the AXI Message Interface.
7. The remote device initiates entry to L2/L3 Ready state by sending PM_Enter_L23 DLLPs. The core will then respond by transmitting PM_Request_Ack DLLPs. A power-on reset will then be required to return the link to L0.

17.5.10.3 L1 Power Management Substates

The L1 Power Management Substates ECN defines an optional mechanism to reduce the idle power in the L1 link state. This is achieved by defining new substates within the L1 state to facilitate the removal of power to the phy, and clocks to the controller. The L1 PM substates are enabled when the link enters L1, due to either PCI Power Management or ASPM. The ECN provides two options when the link is in the L1 state: The L1.1 substate allows clocks and most of the phy power to be turned off, but requires the phy to maintain common-mode voltages on the transmit side. The L1.2 substate enables further reduction in idle power by not requiring common-mode voltages to be maintained. Both L1.1 and L1.2 states allow the Electrical Idle detection circuitry in the phy to be turned off.

The L1 PM Substates use the CLKREQ#(CLKREQN_) sideband signal to control the clocks. The CLKREQ# signal is an open-drain active-low signal shared by the Upstream and Downstream Ports, and can be asserted by either side driving it low. This enables the clock generator. The core clock is turned off when both sides de-assert their CLKREQ# outputs.

The PCIe core provides a CLKREQ_IN_N input and a CLKREQ_OUT_N output to implement the tri-state CLKREQ# pin. The CLKREQ_OUT_N output, when low, enables the tri-state driver driving the CLKREQ# pin, causing assertion of the shared signal. The CLKREQ# signal can also be asserted by the Port on the other side driving it low. The state of this shared signal is monitored by the PCIe core through the CLKREQ_IN_N input, as shown in the following diagram.

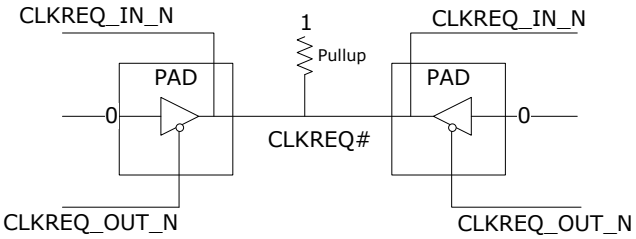


Fig. 17-29 Block Diagram of CLKREQ#

Because the core clock is turned off in the L1.1 and L1.2 substates, a separate power management clock (PM_CLK) is used to drive the L1 PM Substates state machine. This clock must always be active, regardless of the power state of the link. The frequency of the clock can be chosen in the range 1 - 30MHz at the time of Core parameter configuration. The PM_CLK should be driven with the chosen frequency and should not be changed during operation. There is no requirement on the relative phase of this clock with respect to the other clocks used by the core.

The entry to L1 state is governed by the conditions described in the PCI Express Base Specifications. When the LTSSM enters L1, the L1 PM substates State Machine is initially placed in the L1.0 substate. The substate may transition from L1.0 to either L1.1 or L1.2, based on the conditions described in the L1 Power Management Substates ECN. The CLKREQ_OUT_N is de-asserted in both cases, turning off the core clock to the LTSSM.

Either side may initiate a transition out of the L1 state. The remote side initiates the transition by asserting its CLKREQ# output, which turns on the core clock. This also asserts the CLKREQ_IN_N input to the core, causing its L1 PM substate to change to L1.0, and enabling the transition of the LTSSM from L1 into Recovery.

The L1 PM Substates state machine also provides handshake signals PHY_ENT_L1_X and PHY_ACK_L1_X to prepare the local phy module for the removal of the reference clock. The state machine asserts the PHY_ENT_L1_X output in the

L1.0 substate when it has determined that the conditions for transition to the L1.1 or L1.2 substates are met. It then wait for the phy to assert PHY_ACK_L1_X before de-asserting CLKREQ_OUT_N and entering L1.1 or L1.2 substates.

The local client logic may initiate a transition out of the L1 state by activating the POWER_CTRL[clt_req_exit_l1] input to the core. This input is sampled by the L1 PM Substates State Machine in the core in both L1.1 and L1.2 substates. On sensing this input high, the state machine turns on the core clock by asserting CLKREQ_OUT_N, and transitions to the L1.0 substate. This enabled the LTSSM to move out the L1 state into Recovery.

Handshake is required with the PHY for exiting L1 sub-states. If the exit is initiated by the local client logic, CLKREQ_OUT_N should be asserted to request that reference clocks be restored. If the exit was initiated by the remote side, then CLKREQ_IN_N will have been asserted and reference clocks will be restored within the specified time quoted in the ECN.

The controller will de-assert PHY_ENT_L1_X to the PHY and wait for corresponding de-assertion of PHY_ACK_L1_X before transitioning to the L1.0 state. This is required to ensure that the PHY is fully operational and clocks are stable before entering L1.0. For the case of L1.2, the PHY handshake is performed while in the L1.2 Exit substate.

17.5.10.3.1 L1 PM Substates State Machine Operation

The L1 Power Management Substates State Machine is responsible for managing the L1 PM substates associated with the link and activating or inactivating the core clock based on the substate. This state machine is inactive when the LTSSM is not in L1 state. The L1 PM

Substates State Machine enters the L1.0 substate when the LTSSM enters L1, caused by either Autonomous State Power Management (ASPM) or PCI Power Management (PCI-PM). When L1 PM Substates are enabled, it may transition from the L1.0 substate to either the L1.1 substate, or the L1.2.Entry substate. The conditions for each of these transitions are described in Section 5.5.1 of the ECN, L1 Power Management Substates with CLKREQ.

17.5.10.3.1.1 L1.1 Operation

17.5.10.3.1.1.1 Core Configured as EndPoint (Upstream Port)

The following figure illustrates the operation of the L1.1 substate when the core is configured as EP, and when the exit from L1 is initiated locally using the POWER_CTRL[clt_req_exit_l1] input to the core. The core enters L1.1 from L1.0 when the entry conditions for L1.2 are not satisfied and the entry conditions for L1.1 are satisfied.

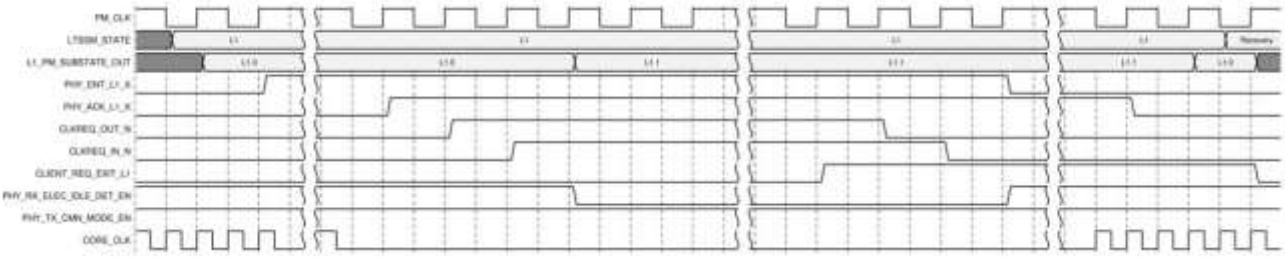


Fig. 17-30 L1.1 Substate Operation: Core as EP, Locally Initiated Exit

On finding that the conditions for entering the L1.1 Substate are met, the L1 PM Substates State Machine first performs the handshake with the phy using the PHY_ENT_L1_X and PHY_ACK_L1_X signals to prepare the phy for the removal of the reference clock. Once the phy has asserted PHY_ACK_L1_X, the core de-asserts CLKREQ_OUT_N. If the link partner also de-asserts its CLKREQ# output, the core clock will become inactive and the CLKREQ_IN_N input to the core will be de-asserted. The L1 PM Substates State Machine transitions to L1.1 on sensing the CLKREQ_IN_N input high.

The local client may request a transition of the link from L1 state by asserting the POWER_CTRL[clt_req_exit_l1] input. When the L1 PM Substates State Machine senses this input high, it asserts CLKREQ_OUT_N to turn on the core clock. This results in CLKREQ_IN_N becoming asserted. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine performs another handshake with the phy by de-asserting PHY_ENT_L1_X and waiting for the phy to respond with the de-assertion of PHY_ACK_L1_X. This handshake is necessary to prepare the phy for the re-activation of the reference

clock. Once this handshake has been completed, the phy transitions back to the L1.0 substate. Subsequently, when the core clock becomes stable, the LTSSM transitions the link to the Recovery state and from there to L0.

The client must de-assert POWER_CTRL[clt_req_exit_l1] once the LTSSM has transitioned out of the L1 state, and before the next entry of the link into L1.

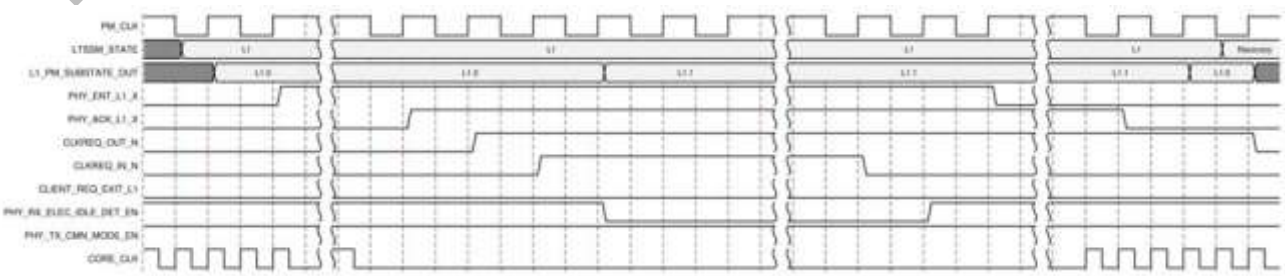


Fig. 17-31 L1.1 Substate Operation: Core as EP, Exit Initiated by Link Partner

The above figure illustrates the operation when the exit from L1 is initiated by the link partner (Downstream Port) that the core is connected to. The core enters L1.1 from L1.0 when the entry conditions for L1.2 are not satisfied and the entry conditions for L1.1 are satisfied. After completing the PHY_ENT_L1_X/ PHY_ACK_L1_X handshake with the phy, the core de-asserts CLKREQ_OUT_N. If the link partner also de-asserts its CLKREQ# output, the core clock will become inactive and the CLKREQ_IN_N input to the core will be de-asserted, causing the L1 PM Substates State Machine to enter the L1.1 state.

The Downstream Port initiates the transition of the link from L1 state by asserting its CLKREQ# output. This results in assertion of the CLKREQ_IN_N input to the core. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine prepares the phy for exit from L1.1 by de-asserting PHY_ENT_L1_X and waiting for the phy to de-assert PHY_ACK_L1_X. Once this handshake has been completed, the L1 PM Substates State Machine transitions to the L1.0 substate. Meanwhile, the de-assertion of CLKREQ# by the Downstream Port results in the core clock becoming active. This enables the LTSSM to move out of L1 into Recovery.

The L1 PM Substates State Machine does not assert CLKREQ_OUT_N until the LTSSM reaches the Recovery state, but this does not affect the operation of the core clock generator because of the Downstream Port maintaining its CLKREQ# output low.

17.5.10.3.1.1.2 Core Configured as Root Complex (Downstream Port)

The figure below illustrates the operation of the L1.1 substate when the core is configured as Root Complex, and when the exit from L1 is initiated locally using the CLIENT_REQ_EXIT_L1 input to the core. The core enters L1.1 from L1.0 when the entry conditions for L1.2 are not satisfied and the entry conditions for L1.1 are satisfied. The entry sequence is identical to that described in Section 1.2.1.1, for the EndPoint case.

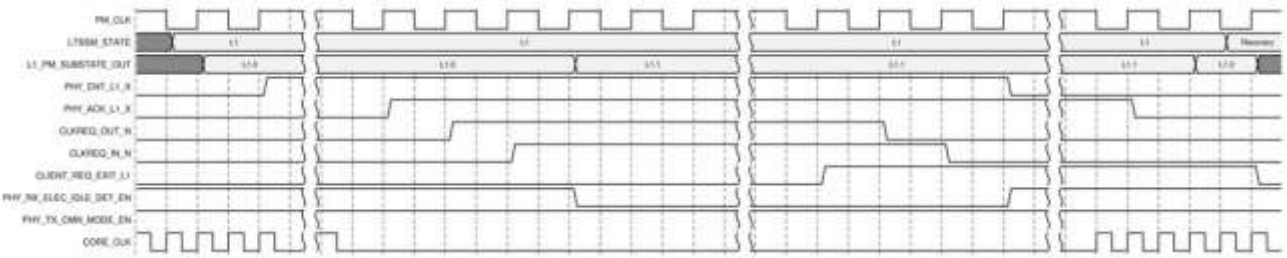


Fig. 17-32 L1.1 Substate Operation: Core as RC, Locally Initiated Exit

The local client may request a transition of the link from L1 state by asserting the POWER_CTRL[clt_req_exit_l1] input. When the L1 PM Substates State Machine senses this input high, it asserts CLKREQ_OUT_N to turn on the core clock. This results in CLKREQ_IN_N becoming asserted. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine performs another handshake with the phy by de-asserting PHY_ENT_L1_X and waiting for the phy to respond with the de-assertion of PHY_ACK_L1_X. This handshake is necessary to prepare the phy for the re-activation of the reference

clock. Once this handshake has been completed, the phy transitions back to the L1.0 substate. Subsequently, when the core clock becomes stable, the LTSSM transitions the link to the Recovery state and from there to L0.

The client must de-assert POWER_CTRL[clt_req_exit_l1] once the LTSSM has transitioned out of the L1 state, and before the next entry of the link into L1.

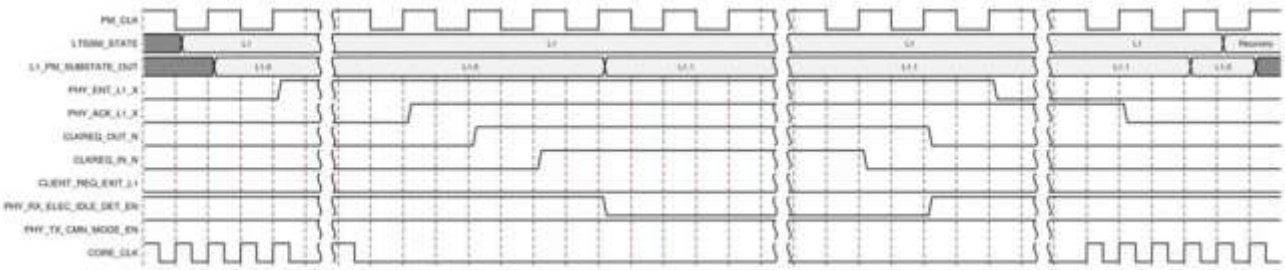


Fig. 17-33 L1.1 Substate Operation: Core as RC, Exit Initiated by Link Partner

The above figure illustrates the operation when the exit from L1 is initiated by the link partner (Upstream Port) that the core is connected to. The core enters L1.1 from L1.0 when the entry conditions for L1.2 are not satisfied and the entry conditions for L1.1 are satisfied. After completing the PHY_ENT_L1_X/ PHY_ACK_L1_X handshake with the phy, the core de-asserts CLKREQ_OUT_N. If the link partner also de-asserts its CLKREQ# output, the core clock will become inactive and the CLKREQ_IN_N input to the core will be de-asserted, causing the L1 PM Substates State Machine to enter the L1.1 state.

The Upstream Port initiates the transition of the link from L1 state by asserting its CLKREQ# output. This results in assertion of the CLKREQ_IN_N input to the core. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine first asserts CLKREQ_OUT_N, and proceeds to prepare the phy for exit from L1.1 by de-asserting PHY_ENT_L1_X and waiting for the phy to de-assert PHY_ACK_L1_X. Once this handshake has been completed, the L1 PM Substates State Machine transitions to the L1.0 substate. Meanwhile, the de-assertion of CLKREQ# results in the core clock becoming active. This enables the LTSSM to move out of L1 into Recovery.

The L1 PM Substates State Machine does not assert CLKREQ_OUT_N until the LTSSM reaches the Recovery state, but this does not affect the operation of the core clock generator because of the Downstream Port maintaining its CLKREQ# output low.

17.5.10.3.1.2 L1.2 Operation

17.5.10.3.1.2.1 L1.2 Entry Sequence

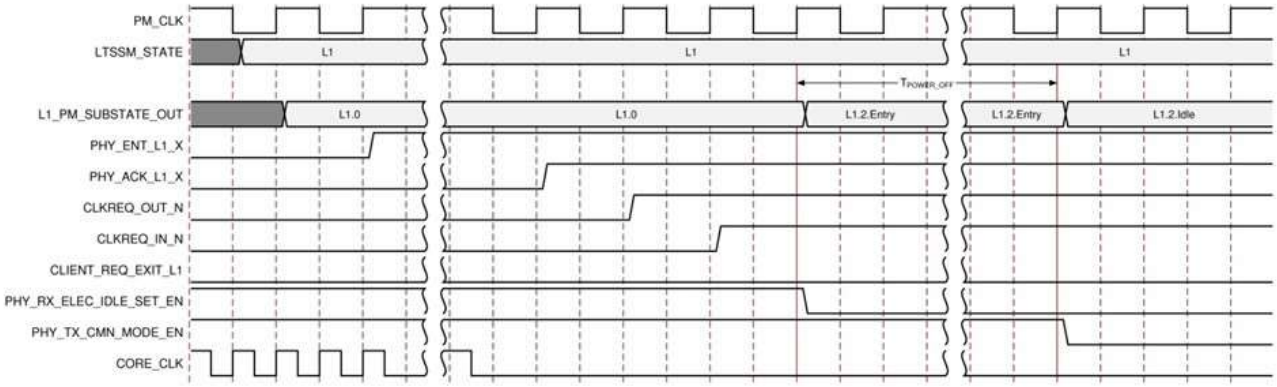


Fig. 17-34 L1.2 Substate Entry Operation

The above figure illustrates the sequence for the L1 PM Substates State Machine to enter the L1.2 substate, when the core is configured as EP or RC. If the entry conditions for L1.2 are satisfied, it first performs the handshake with the phy using the PHY_ENT_L1_X and PHY_ACK_L1_X signals to prepare the phy for the removal of the reference clock.

Once the phy has asserted PHY_ACK_L1_X, the core de-asserts CLKREQ_OUT_N. If the link partner also de-asserts its CLKREQ# output, the core clock will become inactive and the CLKREQ_IN_N input to the core will be de-asserted. The L1 PM Substates State Machine transitions to L1.2. Entry when it senses the CLKREQ_IN_N input high.

While the L1 PM Substates State Machine is in the L1.2. Entry substate, it monitors the CLKREQ_IN_N input and transitions back to the L1.0 substate if it is found asserted. If CLKREQ_IN_N remains de-asserted, the state machine will stay in the L1.2. Entry substate for an interval TPOWER_OFF (set to 2 microseconds) and then transition to L1.2. Idle substate.

17.5.10.3.1.2.2 L1.2 Exit Sequence for EndPoint (Upstream Port)

When the L1 PM Substates State Machine is in the L1.2. Idle substate, either the local client or the link partner may initiate a transition of the link out of the L1 state. The figure below illustrates the operation of the L1.2 substates when the core is configured as EP, and when the exit from L1 is initiated locally using the POWER_CTRL[clt_req_exit_l1] input to the core.

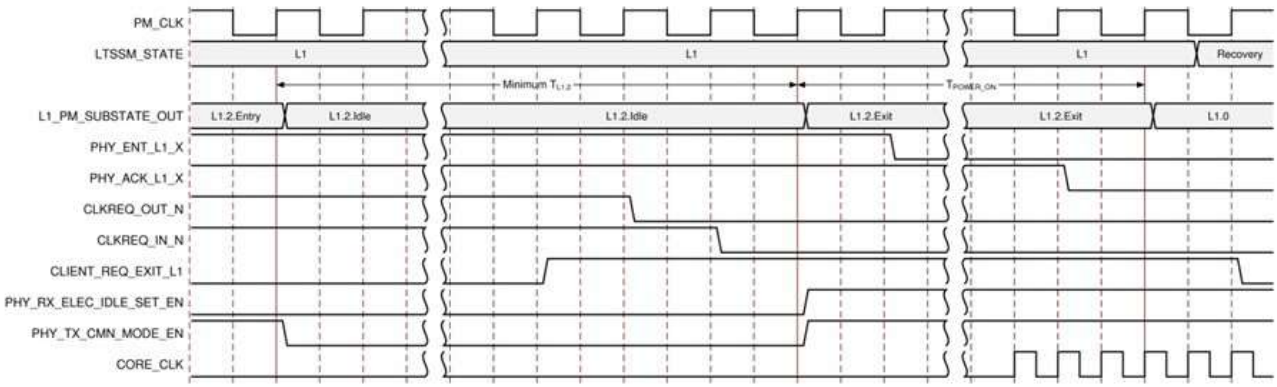


Fig. 17-35 L1.2 Substate Operation: Core as EP, Locally Initiated Exit

The local client requests a transition of the link from L1 state by asserting the CLIENT_REQ_ EXIT_L1 input. When the L1 PM Substates State Machine senses this input high, it asserts CLKREQ_OUT_N to turn on the core clock. This results in CLKREQ_IN_N becoming asserted. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine transitions to the L1.2. Exit substate.

While in the L1.2. Exit substate, the L1 PM Substates State Machine performs the PHY_ENT_L1_X/ PHY_ACK_L1_X handshake with the phy to prepare the phy for the re-introduction of the clocks, and subsequently transitions back to the L1.0 substate. The L1 PM Substates State Machine must stay in the L1.2. Exit substate for a minimum interval of TPOWER_ON. The duration of this interval is determined by the setting of the TPOWER_ON value and scale parameters in the L1 PM Substates Control 2 Register. The interval can vary of 0 to 3100 microseconds based on the setting of these register fields.

Meanwhile, the de-assertion of CLKREQ# by the L1 PM Substates State Machine results in the core clock becoming active. This enables the LTSSM. If the core clock become stable before the L1 PM Substates State Machine has reached the L1.0 substate, the LTSSM waits in the L1 state for the latter to reach L1.0, and then moves into Recovery.

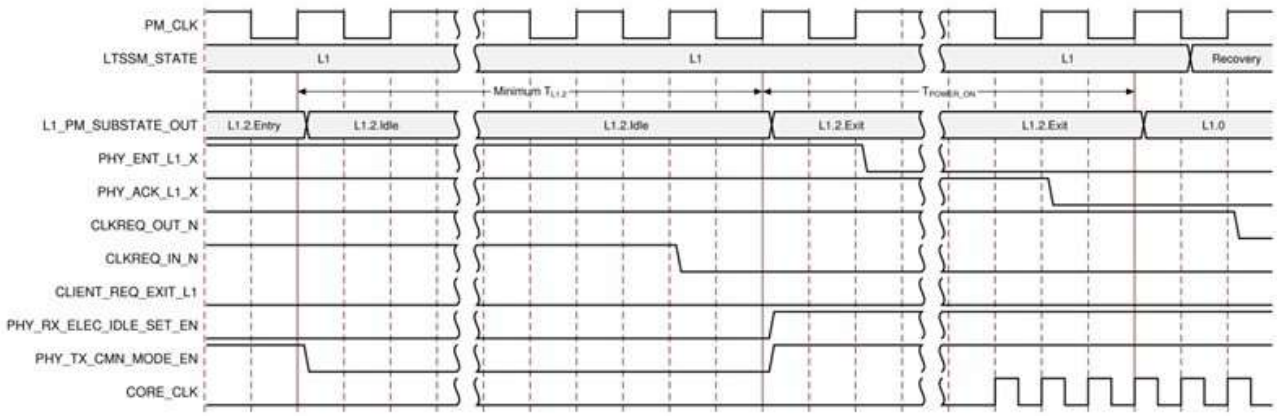


Fig. 17-36 L1.2 Substate Operation: Core as EP, Exit Initiated by Link Partner

The above figure illustrates the operation when the exit from L1 is initiated by the link partner (Downstream Port) that the core is connected to. When in the L1.2. Idle substate, the Downstream Port initiates the transition of the link from L1 state by asserting its CLKREQ# output. This results in assertion of the CLKREQ_IN_N input to the core. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine transitions to the L1.2. Exit substate (after satisfying the requirement of a minimum 4 us stay in L1.2. Idle). The L1 PM Substates State Machine then transitions to the L1.0 substate, after completing the handshake with the phy for re-enabling its clocks, and staying in the L1.2. Exit substate for a minimum interval of TPOWER_ON.

Meanwhile, the de-assertion of CLKREQ# by the Downstream Port results in the core clock becoming active. This enables the LTSSM to move out of L1 into Recovery. The L1 PM Substates State Machine does not assert CLKREQ_OUT_N until the LTSSM reaches the Recovery state, but this does not affect the operation of the core clock generator because of the Downstream Port maintaining its CLKREQ# output low.

17.5.10.3.1.2.3 Core Configured as Root Complex (Downstream Port)

Figure 9 illustrates the operation of the L1.2 substates when the core is configured as Root Complex, and when the exit from L1 is initiated locally using the POWER_CTRL[clt_req_exit_l1] input to the core. The exit sequence in this case is identical to that of the EndPoint, described in Section 1.2.2.2.

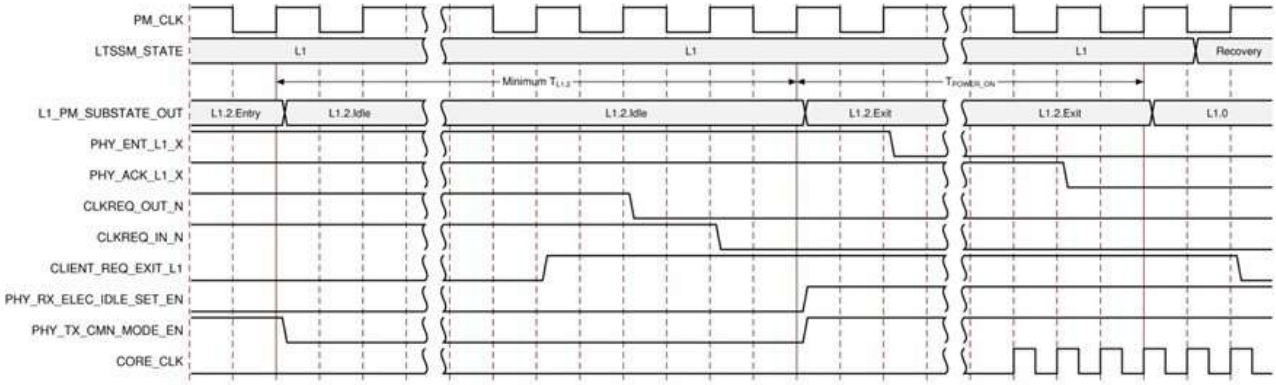


Fig. 17-37 L1.2 Substate Operation: Core as RC, Locally Initiated Exit

Figure 10 illustrates the operation when the exit from L1 is initiated by the link partner (Upstream Port) that the core is connected to. When in the L1.2. Idle substate, the Upstream Port initiates the transition of the link from L1 state by asserting its CLKREQ# output. This results in assertion of the CLKREQ_IN_N input to the core. On sensing CLKREQ_IN_N low, the L1 PM Substates State Machine transitions to the L1.2. Exit substate (after satisfying the requirement of a minimum 4 us stay in L1.2.Idle).

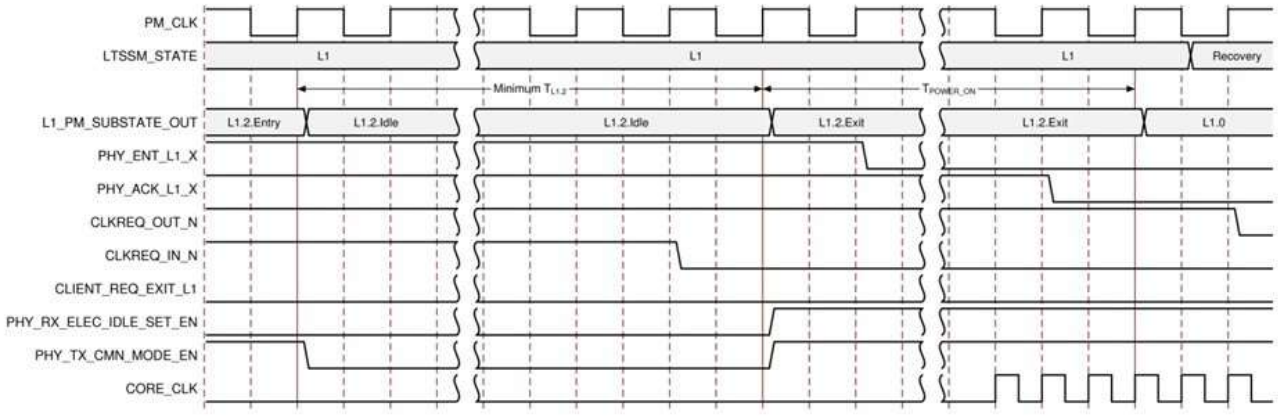


Fig. 17-38 L1.2 Substate Operation: Core as RC, Exit Initiated by Link Partner

After staying in the L1.2. Exit substate for an interval of TPOWER_ON, the L1 PM Substates State Machine transitions to the L1.0 substate. While in the L1.2. Exit Substate, the state machine also completes the handshake with the phy to prepare it for the re-introduction of the clocks.

Meanwhile, the de-assertion of CLKREQ# by the Downstream Port results in the core clock becoming active. This enables the LTSSM to move out of L1 into Recovery. The L1 PM Substates State Machine does not assert CLKREQ_OUT_N until the LTSSM reaches the Recovery state, but this does not affect the operation of the core clock generator because of the Downstream Port maintaining its CLKREQ# output low.

17.6 Register Description

17.6.1 Internal Register Address Mapping

This section describes the PCIe local address mapping. The base address of Local Client register and Core register is 0xFD000000, the size is 16 Mbytes. More detail partition is shown below.

Table 17-22 PCIe Client and Core Register Address Mapping

Mode	A[23]	Address Space Select	Address region
EP/RC	0	Client Register Set	0xFD00_0000~0xFD7f_ffff
EP/RC	1	Core Register Set	0xFD80_0000~0xFDff_ffff

The following diagram illustrates the organization of the configuration and management registers in the PCIe core(Core Register Set).

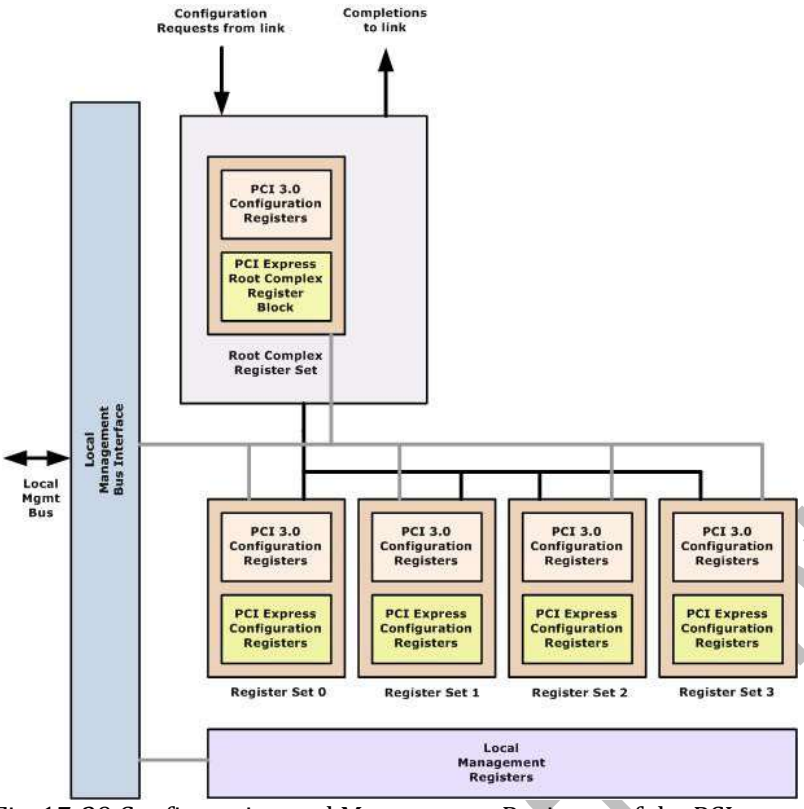


Fig. 17-39 Configuration and Management Registers of the PCIe core

The example shows a core supporting four distinct PCI Functions. The registers can be divided into the following groups:

- Each PCI Function has a set of registers defining its configuration space. These register sets are referred to as Configuration Register Sets. In the example shown, there are four Configuration Register Sets, which together support four distinct PCI Functions. The individual registers in each of these register sets can be accessed by the configuration requests arriving from the link to which the register set is mapped. All the registers are also accessible to a local processor through the local management bus. The registers in each Configuration Register Set are further sub-divided into the PCI 3.0-compatible Configuration Registers and PCI Express Configuration Registers. The former includes the configuration registers defined in the PCI3.0 Specification and registers in the PCI Express Capability Structure. These have register addresses in the range 0 - 63 (decimal). The latter are Extended Capability Registers unique to PCI Express, and are defined in the PCI Express Base Specification 1.1 or 2.0. These register addresses are in the range 64 to 1023 (decimal).In a dual-mode core, these registers are accessible only when the core is strapped as an Endpoint (MODE_SELECT input set to 0).
- For Root Port cores, there is one set of Root Port registers associated with each of the links. These registers are not accessible via configuration requests from the link, but can be read and written through the local management bus. In a dual-mode core, these registers are accessible only when the core is strapped as a Root Port (MODE_SELECT input set to 1).
- There is a set of local management registers storing configuration parameters and other diagnostic information for the core. These registers are accessible only to the local processor via the local management bus, and are not visible from the external PCI links. These registers are accessible in both EP and RP modes.

The following table shows the global address map of the core as observed from the local management bus, as a function of the management address bits and the RC/EP setting

Table 17-23 Global Address Map for Core Local Management Bus

Mode	A[22]	A[21]	A[20]	A[19:12]	Address Space[11:2]	Address Base
0(EP)	0	x	0	0	PCI/PCIe Function 0 Configuration Registers	0xFD80_0000
0(EP)	0	x	0	1~8	PCI/PCIe Virtual Function 0~7 Configuration Registers	0xFD81_0000
0(EP)	0	x	0	9~255	reserved	
1(RP)	0	0	0	0	Root Port Registers, normal access	0xFD80_0000
1(RP)	0	1	0	0	Root Port Registers. In this mode, certain RO fields in the configuration space can be written. Please see documentation of the RC mode registers for more information.	0xFDa0_0000
X(EP/RP)	0	x	1	0	Local management registers	0xFD90_0000
X(EP/RP)	1	0	x	x	Address Translation registers	0xFDC0_0000
X(EP/RP)	1	1	x	x	PCIe DMA registers	0xFDE0_0000

Register addresses stated in this section are DWORD addresses. In write operations, individual bytes can be addressed by the use of byte-enable bits. The addresses not defined are reserved. A configuration access from the link to a reserved address causes the core to return a completion packet with the UR (Unsupported Request) completion code. A read from the local management bus to a reserved address returns all zero's, and a write to a reserved address does not modify any of the registers. All registers (with the exception of reserved or hardwired fields) are writable from the local management bus

17.6.2 PCIe Client Registers Summary

Name	Offset	Size	Reset Value	Description
PCIE_CLIENT_BASIC_STRAP_CONF	0x0000	W	0x000000c1	Basic strap configuration register
PCIE_CLIENT_POWER_CTRL	0x0004	W	0x00000000	PCIe client power control configuration
PCIE_CLIENT_POWER_STATUS	0x0008	W	0x00000000	PCIe power management status
PCIE_CLIENT_LEGACY_INT_CTRL	0x000c	W	0x00000000	Legacy interrupt control
PCIE_CLIENT_ERR_CTRL	0x0010	W	0x00000000	Error control register
PCIE_CLIENT_ERR_CNT	0x0014	W	0x00000000	Error counter
PCIE_CLIENT_HOT_RESET_CTRL	0x0018	W	0x00000000	Hot reset control
PCIE_CLIENT_SIDE_BAND_CTRL	0x001c	W	0x00000004	Side band control configuration
PCIE_CLIENT_SIDE_BAND_STATUS	0x0020	W	0x000011f1	Side band status
PCIE_CLIENT_FC_LEVEL_RST_DONE	0x0024	W	0x00000000	Generate function level reset done pulse
PCIE_CLIENT_FLR_STATUS	0x0028	W	0x00000000	Function level reset status
PCIE_CLIENT_VF_STATUS	0x002c	W	0x00000000	Virtual function status
PCIE_CLIENT_VF_PWR_STATUS	0x0030	W	0x00000000	Virtual function power status

Name	Offset	Size	Reset Value	Description
PCIE_CLIENT_VF_TPH_STATUS	0x0034	W	0x00000000	Virtual function TPH status
PCIE_CLIENT_TPH_STATUS	0x0038	W	0x00000000	Physical TPH status
PCIE_CLIENT_DEBUG_OUT_0	0x003c	W	0x00000000	Debug information 0
PCIE_CLIENT_DEBUG_OUT_1	0x0040	W	0x00000000	Debug information 1
PCIE_CLIENT_BASIC_STATUS0	0x0044	W	0x00000280	Basic status 0
PCIE_CLIENT_BASIC_STATUS1	0x0048	W	0x00080001	Basic status 1
PCIE_CLIENT_INT_MASK	0x004c	W	0x0000ffff	Interrupt mask
PCIE_CLIENT_INT_STATUS	0x0050	W	0x00000000	Interrupt status
PCIE_CLIENT_MSG_CTRL	0x0054	W	0x00000000	Message receive control register
PCIE_CLIENT_MSG_STATUS	0x0058	W	0x00000000	Message control status
PCIE_CLIENT_MSG_CODE0	0x005c	W	0x00000000	Message code 0
PCIE_CLIENT_MSG_CODE1	0x0060	W	0x00000000	Message code 1
PCIE_CLIENT_MSG_DATA_LEN	0x0064	W	0x00000000	Message data length
PCIE_CLIENT_MSG_FIFO_RD_DATA	0x0100	W	0x00000000	Message fifo read data
PCIE_CLIENT_CONF_NU0	0x0200	W	0x00000000	Configuration no used
PCIE_CLIENT_CONF_NU1	0x0204	W	0x00000000	Configuration no used

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

17.6.3 PCIe Client Detail Register Description

PCIE_CLIENT_BASIC_STRAP_CONF

Address: Operational Base + offset (0x0000)

Basic strap configuration register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write mask bits For each served bit 1'b0: write mask 1'b1: write enable
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x1	pcie_gen_sel Generation support select 1'b0: Gen1 mode 1'b1: Gen2 mode This strap input selects the generation of the PCI Express protocol supported by the core. If Gen1 mode. The core advertises only Gen1 capability in this mode, and will always operate at Gen1 speed. If Gen2 mode. The core advertises Gen1 and Gen2 capabilities in this mode, but not Gen3. The link may operate at Gen1 or Gen2 speed.
6	RW	0x1	mode_select Controller operation mode select 1'b0: Endpoint operation 1'b1: Root Port operation
5:4	RW	0x2	lane count in configure the lane count supported 2’b11: reserved 2’b10: X4 2’b01: X2 2’b00: X1
3	RW	0x0	ari_en Alternate interpretation enable 1'b0: legacy interpretation of the PCI Routing ID 1'b1: alternate interpretation of the PCI Routing ID This input is strapped to 0 for legacy interpretation of the PCI Routing ID (8-bit Bus + 5-bit Device + 3- bit Function). A 1 at this input enables the alternate interpretation (8-bit Bus + 8-bit Function).
2	RW	0x0	sr_iov_en Single root I/O virtualization feature enable 1'b0: disable SR-IOV feature 1'b1: enable SR-IOV feature In a core supporting the Single Root I/O Virtualization feature, this strap input must be tied high to enable the SR-IOV feature. The ari_en input must also be strapped high to enable the SR-IOV feature.
1	RW	0x0	link_train_en Link training enable 1'b0: Quite state 1'b1: enable link training This input must be set to 1 to enable the LTSSM to bring up the link. Setting it to 0 forces the LTSSM to stay in the Detect Quiet state.

Bit	Attr	Reset Value	Description
0	RW	0x1	conf_en Config enable 1'b0: disable 1'b1: enable When this input is set to 0 in the EP mode, the core will generate a CRS Completion in response to Configuration Requests. In systems where the core configuration registers are loaded from RAM on power-up, this prevents the core from responding to Configuration Requests before all the registers are loaded. This input can be strapped high when the power-on default values of the Configuration Registers do not need to be modified before Configuration Space enumeration.

PCIE_CLIENT_POWER_CTRL
Address: Operational Base + offset (0x0004)
PCIe client power control configuration

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write mask bits For each served bit 1'b0: write mask 1'b1: write enable
15:10	RO	0x0	reserved
9	RW	0x0	pwr_stcg_ack_mode Power state change ack mode select 1'b0: select power state change ack source from bit pwr_stcg_ack 1'b1: select power state change ack source from constant, it always keeps high.
8	WO	0x0	pwr_stcg_ack Power state change ack 1'b0: keep 0 1'b1: write 1 to generate one high pulse ack to controller The client must assert this input to the core for one cycle in response to the assertion of power state change interrupt, when it is ready to transition to the low-power state requested by the configuration write request. The client may permanently maintain this input high if it does not need to delay the return of the completions for the configuration write transactions causing power-state changes.
7	RO	0x0	reserved
6	RW	0x0	hwclr_exit_l1_req Hardware clear exit L1 request 1'b0: software polling and write clear mode 1'b1: hardware polling and auto-clear mode

Bit	Attr	Reset Value	Description
5	RO	0x0	reserved
4	RW	0x0	hwclr_exit_l2_req Hardware clear exit L2 request 1'b0: software polling and write clear mode 1'b1: hardware polling and auto-clear mode
3	RO	0x0	reserved
2	RW	0x0	clt_req_exit_l1 Client request exit L1 power state 1'b0: keep 1'b1: request to exit the L1.1 or L1.2.Idle substate Client request to exit the L1.1 or L1.2.Idle substate. When the core clock is turned off, the client must activate this input to request the L1 PM substate state machine to de-assert CLKREQ_OUT and transition the link out of L1. If the core clock is not turned off in the L1.1 and L1.2 substates, this input can be permanently kept low.
1	RW	0x0	req_trn_l23ready Request transition to L23_Ready state 1'b0: keep 1'b1: transition the power management state of the core to L23_READY When the core is configured as Endpoint, the client may assert this input to transition the power management state of the core to L23_READY (see Chapter 5 of PCI Express Specifications for a detailed description of power management). This is done after the PCI Functions in the core have been placed in the D3 state and after the client has acknowledged the PME_Turn_Off message from the Root Port. Asserting this input causes the link to transition to the L2 state, and requires a power-on reset to resume operation. This input can be hardwired to 0 if the link is not required to transition to L2. This input is not used in the Root Port mode.
0	RW	0x0	clt_req_exit_l2 Client request exit L2 power state 1'b0: keep 1'b1: Exit from L2_IDLE This input can be asserted by the client only in the short interval of time after the link enters L2 and before the system is powered OFF. While the power and clocks are still ON, the client can assert this input to initiate an exit from L2_IDLE->DETECT.

PCIE_CLIENT_POWER_STATUS
Address: Operational Base + offset (0x0008)
PCIe power management status

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	pwr_stcg_fc_num Power state change function number Function number of the function for which a power state change occurred.
15	RO	0x0	reserved
14:12	RO	0x0	l1_pm_subst L1 power management substate This output provides the current state of the L1 PM substates state machine. This output is in the PM_CLK clock domain. Its encodings are: 3'b000 = LTSSM not in L1 state 3'b001 = L1.0 substate 3'b010 = L1.1 substate 3'b011 = Reserved 3'b100 = L1.2.Entry substate 3'b101 = L1.2.Idle substate 3'b110 = L1.2.Exit substate 3'b111 = Reserved
11	RO	0x0	reserved
10:8	RO	0x0	fc_pwr_st Function power state These outputs provide the current power state of the Physical Functions. Bits [2:0] capture the power state of Function 0 The possible power states are: 3'b000: D0_uninitialized 3'b001: D0_active 3'b010: D1 3'b100: D3_hot
7:4	RO	0x0	reserved
3:0	RO	0x0	link_pwr_st Link power state Current power state of the PCIe link: 4'b0001 = L0 4'b0010 = L0s 4'b0100 = L1 4'b1000 = L2

PCIE_CLIENT_LEGACY_INT_CTRL

Address: Operational Base + offset (0x000c)

Legacy interrupt control

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17:16	WO	0x0	write_mask Write mask For each served bit 1'b0: write mask 1'b1: write enable
15:2	RO	0x0	reserved
1	RW	0x0	int_in Legacy interrupt input 1'b0: de-assert 1'b1: assert When the core is configured as EP, this input is used by the client application to signal an interrupt from any of its PCI Functions to the RC using the Legacy PCI Express Interrupt Delivery mechanism of PCI Express. This input corresponds to INTA of the PCI bus. Asserting this signal causes the core to send out an Assert_INTx message, and de-asserting this signal causes the core to transmit a Deassert_INTx message.
0	RW	0x0	int_pend_st Legacy interrupt pending status 1'b0: normal 1'b1: pending When using legacy interrupts, this input is used to indicate the interrupt pending status of the Physical Functions. The input i must be set when an interrupt is pending in Function i.

PCIE_CLIENT_ERR_CTRL
Address: Operational Base + offset (0x0010)
Error control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write mask For each served bit 1'b0: write mask 1'b1: write enable
15:11	RO	0x0	reserved
10	RW	0x0	corr_err_cnt_en Enable correctable error counter 1'b0: disable counter 1'b1: enable counter
9	RW	0x0	nfatal_err_cnt_en Enable non-fatal error counter 1'b0: disable counter 1'b1: enable counter

Bit	Attr	Reset Value	Description
8	RW	0x0	fatal_err_cnt_en Enable fatal error counter 1'b0: disable counter 1'b1: enable counter
7:2	RO	0x0	reserved
1	WO	0x0	corr_err_in_en Assert a correctable error input to core 1'b0: no error 1'b1: write one to generate one pulse The client may activate this input for one cycle to indicate a correctable error detected within the client logic that needs to be reported as an internal error through the PCI Express Advanced Error Reporting mechanism. In response, the core sets the Corrected Internal Error Status bit in the AER Correctable Error Status Register of all enabled Functions, and in EP mode also sends an error message if enabled to do so. This error is not considered Function-specific.
0	WO	0x0	uncorr_err_in_en Assert an uncorrectable error input to core 1'b0: no error 1'b1: write one to generate one pulse The client may activate this input for one cycle to indicate an uncorrectable error detected within the client logic that needs to be reported as an internal error through the PCI Express Advanced Error Reporting mechanism. In response, the core sets the Uncorrectable Internal Error Status bit in the AER Uncorrectable Error Status Register of all enabled Functions, and in EP mode also sends an error message if enabled to do so. This error is not considered Function-specific.

PCIE_CLIENT_ERR_CNT

Address: Operational Base + offset (0x0014)

Error counter

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	W1C	0x00	corr_err_cnt Correctable error counter Correctable error counter, write all one(8'hff) clear the counter.
15:8	W1C	0x00	nfatal_err_cnt Non-fatal error counter Non-fatal error counter, write all one(8'hff) clear the counter.
7:0	W1C	0x00	fatal_err_cnt Fatal error counter Fatal error counter, write all one(8'hff) clear the counter.

PCIE_CLIENT_HOT_RESET_CTRL
Address: Operational Base + offset (0x0018)
Hot reset control

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	WO	0x0	write_mask Write mask For each served bit 1'b0: write mask 1'b1: write enable
15:2	RO	0x0	reserved
1	RW	0x0	link_down_rst_clt_mask Mask link down reset client logic 1'b0: disable link down reset client logic 1'b1: enable link down reset client logic
0	RW	0x0	hot_reset_in Assert hot reset to remote device 0: de-assert 1: assert When this input is asserted in the RC mode, the core initiates a Hot Reset sequence on the PCIe link. This signal should be driven synchronous to the CORE_CLK domain. The controller will keep the PCIe link in hot reset till the time this signal is driven asserted. When de-asserted, controller will bring the PCIe link out of hot reset and initiate link training

PCIE_CLIENT_SIDE_BAND_CTRL
Address: Operational Base + offset (0x001c)
Side band control configuration

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write mask For each served bit 1'b0: write mask 1'b1: write enable
15:12	RO	0x0	reserved
11:8	RW	0x0	rx_standby PCIe phy receiver control Controls whether the PHY RX is active when the PHY is in P0 or P0s states. 1'b0: Active 1'b1: Standby In other modes not mentioned above, this signal is ignored. One bit for each lane.
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	bypass_codec PIPE bypass codec configuration Controls whether the PHY performs 8b/10b encode and decode: 1'b0: 8b/10b encode/decode performed normally 1'b1: 8b/10b encode/decode bypassed Data bus width is 20 bits, TxDataK and RxDataK interfaces are not used, if encode/decode bypassed, and WIDTH_I shall be set high.
5:4	RW	0x0	tx_deemphasis_ext PIPE phy extended de-emphasis configuration, it combine with the standard pipe de-emphasis.
3:1	RO	0x2	pwdn Power state of the phy Power up or down the transceiver. 3'b000: P0, normal operation 3'b001: P0s, power saving state 3'b010: P1, lower power state 3'b011: P2, lowest power state, PLL not powered 3'b111: L1SS.2, common mode off others : L1SS.1, common mode on
0	RW	0x0	non_posted_rej PCIe target non posted reject 1'b0: normal operation 1'b1: reject non posted request This is a single bit input signal which can be asserted by client logic when it cannot service a non-posted request. The core will not present any non-posted requests that it receives from the PCIe Link. It will hold them in the PNP FIFO RAM till the signal is de-asserted. If a non-posted TLP has already been queued from the PNP FIFO and this signal is asserted, the core will place it on the AXI bridge. The client logic must accept the non-posted TLP. The in-flight non-posted TLPs in the core from the PNP FIFO cannot be stopped. However, non-posted TLPs that are in the PNP FIFO RAM when this signal is asserted or come in after the signal is asserted will not be forwarded to the AXI interface. The client must assert this signal when it still can process two or three non-posted TLPs. This will allow posted TLPs to go past non-posted TLPs at the AXI master write interface due to client not being able to service non-posted TLPs.

PCIE_CLIENT_SIDE_BAND_STATUS
Address: Operational Base + offset (0x0020)
Side band status

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x1	phy_st PIPE phy status It indicates completion of several PHY functions including power management state transition and receiver detection. When this signal transitions during entry and exit from any PHY state where PCLK is not provided, then the signaling is asynchronous. When phy power up, "0" state can indicates pll locked
11:10	RO	0x0	reserved
9:8	RO	0x1	data_bus_width PIPE interface data bus width Reports the width of the data bus that the PHY is configured for: 1'b0: 32-bit mode 1'b1: 16-bit mode 0thers: reserved When bypass_codec is high, the interface is 20-bit and these two bits report a value of 2'b01.
7:4	RO	0xf	rx_standby_st RX standby status Indicates PHY's RxStandby state 1'b0: Active 1'b1: Standby Always high during P1/P2/L1SS state.
3	RO	0x0	reserved
2:0	RO	0x1	tx_deemphasis PIPE phy de-emphasis status Transmitter de-emphasis selection, it combined by tx_deemphasis_ext,tx_deemphasis. 3'b000: -6dB de-emphasis 3'b001: -3.5dB de-emphasis 3'b010: 0dB de-emphasis 3'b011: -5.5dB de-emphasis 3'b100: -6.5dB de-emphasis 3'b101: -4dB de-emphasis 3'b110: -1dB de-emphasis 3'b111: -3dB de-emphasis

PCIE_CLIENT_FC_LEVEL_RST_DONE

Address: Operational Base + offset (0x0024)

Generate function level reset done pulse

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:8	WO	0x00	vf_flr_done Virtual function level reset done pulse generate The client must assert bit i of this bus when it has completed the reset operation of Virtual Function i. This causes the core to de-assert FLR_IN_PROGRESS for VF i and to re-enable configuration accesses to the VF. Write one to generate one high pulse.
7:1	RO	0x0	reserved
0	WO	0x0	flr_done Physical function level reset done pulse generate The client must assert bit i of this bus when it has completed the reset operation of Function i. This causes the core to de-assert FLR_IN_PROGRESS for Function i and to re-enable configuration accesses to the Function. Write one to generate one high pulse.

PCIE_CLIENT_FLR_STATUS
Address: Operational Base + offset (0x0028)
Function level reset status

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RO	0x00	vf_flr_in_prog Virtual function level reset in progress The core asserts bit i of this bus when the host initiates a reset of Virtual Function i though its FLR bit in the configuration space. The core continues to maintain the output high until the client sets the FLR_DONE input for the corresponding VF to indicate the completion of the reset operation. One bit for each function 1'b0: normal 1'b1: function level reset in progress
7:1	RO	0x0	reserved
0	RO	0x0	flr_in_prog Function level reset in progress The core asserts bit i of this bus when the host initiates a reset of Function i though its FLR bit in the configuration space. The core continues to maintain the output high until the client sets the FLR_DONE input for the corresponding Function to indicate the completion of the reset operation. 1'b0: normal 1'b1: function level reset in progress

PCIE_CLIENT_VF_STATUS
Address: Operational Base + offset (0x002c)
Virtual function status

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	vf_en Virtual function enable The core sets bit i of this bus when the host has configured the corresponding Virtual Function i. Client logic must check the state of this bit before initiating any request from the VF. 1'b0: disable 1'b1: enable One bit for each function
15:8	RO	0x0	reserved
7:0	RO	0x00	vf_bus_master_en Virtual function bus master enable Bit i of this bus reflects the setting of the Bus Master Enable bit of the PCI Command Register of Virtual Function i. Client logic must check the state of this bit before initiating any memory read or write transactions from the VF. 1'b0: disable 1'b1: enable One bit for each function

PCIE_CLIENT_VF_PWR_STATUS
Address: Operational Base + offset (0x0030)
Virtual function power status

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RO	0x000000	vf_pwr_st Virtual function power status These outputs provide the current power state of the Virtual Functions. Bits [2:0] capture the power state of Virtual Function 0, bits [5:3] capture that of Virtual Function 1, and so on. The possible power states are: 3'b000: D0_uninitialized 3'b001: D0_active 3'b010: D1 3'b100: D3_hot

PCIE_CLIENT_VF_TPH_STATUS
Address: Operational Base + offset (0x0034)
Virtual function TPH status

Bit	Attr	Reset Value	Description
31:24	RO	0x00	vf_tph_reqr_en Virtual function TPH requester enable Each of the 16 bits of this output is driven the TPH Requester Enable bit [8] of the TPH Requester Control Register in the TPH Requester Capability Structure of the corresponding Virtual Function. These bits are active only in the Endpoint mode when SR-IOV is enabled. They indicate whether the software has enabled the device to generate requests with TPH Hints from the associated Virtual Function.
23:0	RO	0x000000	vf_tph_st_mode Virtual function TPH steering tag mode Bits [2:0] of this output reflect the setting of the ST Mode Select bits in the TPH Requester Control Register of Virtual Function 0. Bits [5:3] reflect the setting of the same register field of VF 1, and so on. These bits are active only in the Endpoint mode. They indicate the allowed modes for generation of TPH Hints by the corresponding VF.

PCIE_CLIENT_TPH_STATUS
Address: Operational Base + offset (0x0038)
Physical TPH status

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RO	0x0	tph_reqr_en Physical function TPH requester enable Bit 0 of this output is drives the TPH Requester Enable bit [8] of the TPH Requester Control Register in the TPH Requester Capability Structure of the Physical Function 0. These bits are active only in the Endpoint mode. They indicate whether the software has enabled the device to generate requests with TPH Hints from the associated Physical Function.
7:3	RO	0x0	reserved
2:0	RO	0x0	tph_st_mode Physical function TPH steering tag mode Bits [2:0] of this output reflect the setting of the ST Mode Select bits in the TPH Requester Control Register of Physical Function 0. These bits are active only in the Endpoint mode. They indicate the allowed modes for generation of TPH Hints by the corresponding Physical Function.

PCIE_CLIENT_DEBUG_OUT_0
Address: Operational Base + offset (0x003c)

Debug information 0

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	debug_data_out Output data from the debug bus 16-bit output data from the debug bus, described in Appendix A
15:6	RO	0x0	reserved
5:0	RO	0x00	ltssm_state Link training and status state Current state of the Link Training and Status State Machine within the core. The encodings of this output are described in Appendix B

PCIE_CLIENT_DEBUG_OUT_1

Address: Operational Base + offset (0x0040)

Debug information 1

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:0	RO	0x00000	perf_data_out Performance data out Each of the bits of this vector is explained below: Bit[17]:Pulse appears when event happens,described in Appendix B

PCIE_CLIENT_BASIC_STATUS0

Address: Operational Base + offset (0x0044)

Basic status 0

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:12	RO	0x0	max_payload_size Max payload size The maximum payload size field programmed in the PCI Express Device Control Register. In multiple function cores, this output provides the minimum of the max-payload-size field in the Device Control Registers of all the enabled Physical Functions. The client logic must limit the size of Outgoing Completion payloads to this value. The 3-bit codes are the same as those defined in PCIe Specifications: 3'b000 = 128 bytes 3'b001 = 256 bytes 3'b010 = 512 bytes 3'b011 = 1024 bytes 3'b100 = 2048 bytes 3'b101 = 4096 bytes
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RO	0x2	max_rdreq_size Max read request size The maximum request size field programmed in the PCI Express Device Control Register. In multi?Function cores, this output provides the minimum of the max-read-request field in the Device Control Registers of all the Physical Functions. The client logic must limit the size of outgoing read requests to this value. The 3-bit codes are the same as those defined in PCIe Specifications: 3'b000 = 128 bytes 3'b001 = 256 bytes 3'b010 = 512 bytes 3'b011 = 1024 bytes 3'b100 = 2048 bytes 3'b101 = 4096 bytes
7:6	RO	0x2	negotiated_link_width Negotiated link width Current link width are as follows: 2'b10: x4 2'b01: x2 2'b00: x1 others: Reserved
5	RO	0x0	negotiated_speed Operation speed after negotiation Current operating speed of the link is as follows: 1'b0: 2.5GT/s 1'b1: 5GT/s
4	RO	0x0	reserved
3	RO	0x0	rcb_st Read completion boundary status Provides the setting of the Read Completion Boundary (RCB) bit in the Link Control Register of each Physical Function. In the Endpoint mode, bit 0 indicates the RCB for PF 0 and so on. In the RC mode, bit 0 indicates the RCB setting of the Link Control Register of the RC. For each bit, a value of 1'b0: indicates an RCB of 64 bytes 1'b1: indicates 128 bytes

Bit	Attr	Reset Value	Description
2	RO	0x0	ltr_en Latency tolerance reporting mechanism enable The state of this output reflects the setting of the LTR Mechanism Enable bit in the Device Control 2 Register of Physical Function 0. When the core is configured as an Endpoint, client logic uses this output to enable the generation of LTR messages. This output is not to be used when the core is configured as a Root Complex. 1'b0: disable 1'b1: enable
1:0	RO	0x0	obff_en Optimized buffer flush and fill enable This output reflects the setting of the OBFF Enable field in the Device Control 2 Register 2'b00: OBFF disabled, 2'b01: OBFF enabled using message signaling, Variation A, 2'b10: OBFF enabled using message signaling, Variation B, 2'b11: OBFF enabled using WAKE# signaling.

PCIE_CLIENT_BASIC_STATUS1

Address: Operational Base + offset (0x0048)

Basic status 1

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:20	RO	0x0	link_st Link status Status of the PCI Express link. 2'b00 = No receivers detected. 2'b01 = Link training in progress. 2'b10 = Link up , DL initialization in progress. 2'b11 = Link up, DL initialization completed.
19:16	RO	0x8	fc_st Function status These outputs indicate the states of the Command Register bits in the PCI configuration space of each Function. These outputs are used to enable requests and completions from the host logic. The assignment of bits is as follows: Bit 0: Function 0 IO Space Enable Bit 1: Function 0 Memory Space Enable Bit 2: Function 0 Bus Master Enable Bit 3: Function 0 INTx Disable and so on depending on the number of functions.

Bit	Attr	Reset Value	Description
15:0	RO	0x0001	sys_page_size System page size These bits reflect the setting of the System Page Size Register in the SR IOV capability of each PF. Bits [15:0] reflect bits [15:0] of System Page Size register of PF0

PCIE_CLIENT_INT_MASK
Address: Operational Base + offset (0x004c)
Interrupt mask

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write mask For each served bit 1'b0: write mask 1'b1: write enable
15	RW	0x1	legacy_done_int_mask Legacy interrupt send done interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
14	RW	0x1	msg_int_mask Message receive done interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
13	RW	0x1	hot_reset_int_mask Hot reset interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
12	RW	0x1	dpa_int_mask DPA interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
11	RW	0x1	fatal_err_int_mask Fatal error interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
10	RW	0x1	nfatal_err_int_mask Non-fatal error interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
9	RW	0x1	corr_err_int_mask Correctable error interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask

Bit	Attr	Reset Value	Description
8	RW	0x1	intd_mask INTD interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
7	RW	0x1	intc_mask INTC interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
6	RW	0x1	intb_mask INTB interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
5	RW	0x1	inta_mask INTA interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
4	RW	0x1	local_int_mask Local interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
3	RW	0x1	udma_int_mask uDMA interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
2	RW	0x1	phy_int_mask Phy interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask
1	RW	0x1	hot_plug_int_mask Hot plug interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask reserved
0	RW	0x1	pwr_stcg_int_mask Power state change interrupt mask 1'b0: interrupt enable 1'b1: interrupt mask

PCIE_CLIENT_INT_STATUS

Address: Operational Base + offset (0x0050)

Interrupt status

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15	W1C	0x0	legacy_done_int Legacy interrupt send done interrupt 1'b0: no interrupt 1'b1: interrupt A pulse on this output indicates that the core has sent an INTx Assert or Deassert message in response to a change in the state of one of the int_in
14	W1C	0x0	msg_int Message receive done interrupt 1'b0: no interrupt 1'b1: interrupt When a message received done by Client message FIFO, an interrupt will generate
13	W1C	0x0	hot_reset_int Hot reset interrupt 1'b0: no interrupt 1'b1: interrupt When a hot reset send done interrupt generated in RC mode, it indicates that the Endpoint Device has also received the Hot Reset, and then the hot_reset_in can be de-assert
12	W1C	0x0	dpa_int DPA interrupt 1'b0: no interrupt 1'b1: interrupt The core generates an interrupt when a Configuration Write transaction writes into the Dynamic Power Allocation Control Register to modify the DPA power state of the device. A interrupt indicates such a DPA event for PF 0, and so on. The local software running on the End Point must read the DPA Control Register of the corresponding Function to determine the DPA substate requested by the host and set the power state of the device accordingly
11	W1C	0x0	fatal_err_int Fatal error interrupt 1'b0: no interrupt 1'b1: interrupt In the EP mode, the core activates this output for one cycle when it has detected a fatal error and its reporting is not masked. In multi-Function cores, this is the logical OR of the fatal error status bits in the Device Status Registers of all Functions. In the RP mode, this output is activated on detection of a local fatal error, when its reporting is not masked. This signal also gets activated in response to an error message received from the link if Fatal Error Reporting is enabled in the Root Error Command register.

Bit	Attr	Reset Value	Description
10	W1C	0x0	nfatal_err_int Non-fatal error interrupt 1'b0: no interrupt 1'b1: interrupt In the EP mode, the core activates this output for one cycle when it has detected a non-fatal error and its reporting is not masked. In multi-Function cores, this is the logical OR of the non-fatal error status bits in the Device Status Registers of all Functions. In the RC mode, this output is activated on detection of a local fatal error, when its reporting is not masked. This signal also gets activated in response to an error message received from the link if Fatal Error Reporting is enabled in the Root Error Command register.
9	W1C	0x0	corr_err_int Correctable error interrupt 1'b0: no interrupt 1'b1: interrupt In the EP mode, the core activates this output for one cycle when it has detected a correctable error and its reporting is not masked. In multi-Function cores, this is the logical OR of the correctable error status bits in the Device Status Registers of all Functions. In the RC mode, this output is activated on detection of a local correctable error, when its reporting is not masked. This signal also gets activated in response to an error message received from the link if Correctable Error Reporting is enabled in the Root Error Command register.
8	RO	0x0	intd INTD interrupt 1'b0: no interrupt 1'b1: interrupt When the core is configured as RC, this interrupt emulate the PCI legacy interrupts INTD. The core asserts an interrupt output when it has received an Assert_INTD message from the link, and deasserts it when it receives a Deassert_INTD message.
7	RO	0x0	intc INTC interrupt 1'b0: no interrupt 1'b1: interrupt When the core is configured as RC, this interrupt emulate the PCI legacy interrupts INTC. The core asserts an interrupt output when it has received an Assert_INTC message from the link, and deasserts it when it receives a Deassert_INTC message.

Bit	Attr	Reset Value	Description
6	RO	0x0	intb INTB interrupt 1'b0: no interrupt 1'b1: interrupt When the core is configured as RC, this interrupt emulate the PCI legacy interrupts INTB. The core asserts an interrupt output when it has received an Assert_INTB message from the link, and deasserts it when it receives a Deassert_INTB message.
5	RO	0x0	inta INTA interrupt 1'b0: no interrupt 1'b1: interrupt When the core is configured as RC, this interrupt emulate the PCI legacy interrupts INTA. The core asserts an interrupt output when it has received an Assert_INTA message from the link, and deasserts it when it receives a Deassert_INTA message.
4	RO	0x0	local_int Local interrupt 1'b0: no interrupt 1'b1: interrupt Local Error and Status Register Interrupt. This is a level interrupt till cleared by software Detail information refers to Local Error and Status Register description in PCIe Core register section "Local Management Registers"
3	RO	0x0	udma_int uDMA interrupt 1'b0: no interrupt 1'b1: interrupt DMA Interrupt to the system processor. Will be asserted on a "DMA Done" or a "DMA Error" event

Bit	Attr	Reset Value	Description
2	RO	0x0	phy_int Phy interrupt 1'b0: no interrupt 1'b1: interrupt This interrupt is used by the core in the RP mode to signal one of the following link training-related events: 1. The link bandwidth changed as a result of the change in the link width or operating speed and the change was initiated locally (not by the link partner), without the link going down. This interrupt is enabled by the Link Bandwidth Management Interrupt Enable bit in the Link Control Register. The status of this interrupt can be read from the Link Bandwidth Management Status bit of the Link Status Register. 2. The link bandwidth changed autonomously as a result of the change in the link width or operating speed and the change was initiated by the remote node. This interrupt is enabled by the Link Autonomous Bandwidth Interrupt Enable bit in the Link Control Register. The status of this interrupt can be read from the Link Autonomous Bandwidth Status bit of the Link Status Register. The phy interrupt is not active when the core is configured as an EndPoint.
1	RO	0x0	hot_plug_int Hot plug interrupt 1'b0: no interrupt 1'b1: interrupt Hot Plug Interrupt Output for Software Notification of Hot Plug events. Currently, this interrupt reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	pwr_stcg_int Power state change interrupt 1'b0: no interrupt 1'b1: interrupt The core asserts this output when the power state of a Physical or Virtual Function is being changed to the D1 or D3 states by a write into its Power Management Control Register. The core maintains this output high until the client asserts the pwr_stcg_ack input to the core. While interrupt remains high, the core will not return completions for any pending configuration read or write transaction received by the core. The intent is to delay the completion for the configuration write transaction that caused the state change until the client is ready to transition to the low- power state. When interrupt is asserted, the Function number associated with the configuration write transaction is provided on the pwr_stcg_fc_num. When the client asserts pwr_stcg_ack, the new state of the Function that underwent the state change will be reflected on the fc_pwr_st (for PFs) or the vf_pwr_st (for VFs) outputs of the core.

PCIE_CLIENT_MSG_CTRL
Address: Operational Base + offset (0x0054)
Message receive control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write mask For each served bit 1'b0: write mask 1'b1: write enable
15:13	RO	0x0	reserved
12:8	RW	0x00	almfull_water_mark Almost full water mark almost full water mark configuration
7:2	RO	0x0	reserved
1	RW	0x0	msg_fifo_rx_mode Message fifo receive mode select 1'b0: partial mode 1'b1: full mode
0	RW	0x0	msg_fifo_en Message fifo receive enable 1'b0: disable message receive 1'b1: enable client message receive

PCIE_CLIENT_MSG_STATUS

Address: Operational Base + offset (0x0058)
Message control status

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:8	RO	0x00	space2empty Space to empty Data length before empty
7:3	RO	0x0	reserved
2	RO	0x0	fifo_full Message fifo full 1'b0: non-full 1'b1: fifo full
1	RO	0x0	fifo_empty Message fifo empty 1'b0: non-full empty 1'b1: fifo_empty
0	RO	0x0	almost_full Message fifo almost full flag 1'b0: non-almost full 1'b1: almost full

PCIE_CLIENT_MSG_CODE0
Address: Operational Base + offset (0x005c)
Message code 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	mtpat4 Match pattern 4 Pattern4
23:16	RW	0x00	mtpat3 Match pattern 3 Pattern3
15:8	RW	0x00	mtpat2 Match pattern 2 Pattern2
7:0	RW	0x00	mtpat1 Match pattern 1 Pattern 1

PCIE_CLIENT_MSG_CODE1
Address: Operational Base + offset (0x0060)
Message code 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	mtpat8 Match pattern 8 Pattern8

Bit	Attr	Reset Value	Description
23:16	RW	0x00	mtpat7 Match pattern 7 Pattern7
15:8	RW	0x00	mtpat6 Match pattern 6 Pattern6
7:0	RW	0x00	mtpat5 Match pattern 5 Pattern5

PCIE_CLIENT_MSG_DATA_LEN
Address: Operational Base + offset (0x0064)
Message data length

Bit	Attr	Reset Value	Description
31:24	RC	0x00	length4 Length4 Length4, record the 4th recently received message length.
23:16	RC	0x00	length3 Length3 Length3, record the 3rd recently received message length.
15:8	RC	0x00	length2 Length2 Length2, record the 2nd recently received message length.
7:0	RC	0x00	length1 Length1 Length1, record the recently received message length.

PCIE_CLIENT_MSG_FIFO_RD_DATA
Address: Operational Base + offset (0x0100)
Message fifo read data

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RD_DATA Message fifo read data Message fifo read data

PCIE_CLIENT_CONF_NU0
Address: Operational Base + offset (0x0200)
Configuration no used

Bit	Attr	Reset Value	Description
31:0	RO	0x0	reserved

PCIE_CLIENT_CONF_NU1
Address: Operational Base + offset (0x0204)
Configuration no used

Bit	Attr	Reset Value	Description
31:0	RO	0x0	reserved

17.6.4 Physical Function Configuration Register Description

There are 1 Physical Functions, which are assigned Function numbers 0. The registers of these Functions are accessed by setting MGMTADDR[18] to 0 and MGMT_ADDR[17:10] to the Function number.

The Physical Function registers are described in detail below.

17.6.4.1 i_pcie_base

17.6.4.1.1 Vendor ID and Device ID

Propname：Vendor ID and Device ID

Address：@0x0

Description：16-bit Vendor ID register and 16-bit Device ID register

Bits	SW	Name	Description	Reset
15:0	R	Vendor ID [VID]	This is the Vendor ID assigned by PCI SIG to the manufacturer of the device. The Vendor ID is set in the Vendor ID Register within the local management register block.	16'h17cd
31:16	R	Device ID [DID]	Device ID assigned by the manufacturer of the device. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be rewritten independently for each Function from the local management bus.	16'h100

17.6.4.1.2 Command and Status Register

Propname：Command and Status Register

Address：@0x4

Description：16-bit Command Register and 16-bit Status Register.

Bits	SW	Name	Description	Reset
0	R/W	IO-Space Enable [ISE]	Enables IO accesses through the core for this PCI Function. This field can be written from the local management bus.	0x0
1	R/W	Mem-Space Enable [MSE]	Enables memory accesses through the core for this PCI Function. This field can be written from the local management bus.	0x0
2	R/W	Bus-Master Enable [BE]	Enables the device to issue memory and I/O requests from this Function. This field can be written from the local management bus.	0x0
5:3	R	Reserved [R0]	Reserved	0x0

Bits	SW	Name	Description	Reset
6	R/W	Parity Error Response Enable [PERE]	When this bit is 1, the core sets the Master Data Parity Error status bit when it detects the following error conditions: (i) The core receives a poisoned completion from the link in response to a request. (ii) The core sends out a poisoned write request on the link (this may be because an underflow occurred during the packet transfer at the host interface of the core.). When this bit is 0, the Master Data Parity Error status bit is never set. This field can be written from the local management bus.	0x0
7	R	Reserved [R1]	Reserved	0x0
8	R/W	SERR Enable [SE]	Enables the reporting of fatal and non-fatal errors detected by the core to the Root Complex. This field can be written from the local management bus.	0x0
9	R	Reserved [R2]	Reserved	0x0
10	R/W	INTx Message Disabled [IMD]	Enables or disables the transmission of INTx Assert and De-assert messages from the core. Setting this bit to 1 disables generation of INTx assert/de-assert messages in the core. This field can be written from the local management bus.	0x0
15:11	R	Reserved [R3]	Reserved	0x0
18:16	R	Reserved [R4]	Reserved	0x0
19	R	Interrupt Status [IS]	This bit is valid only when the core is configured to support legacy interrupts. Indicates that the core has a pending interrupt, that is, the core has sent an Assert_INTx message but has not transmitted a corresponding Deassert_INTx message.	0x0
20	R	Capabilities List [CL]	Indicates the presence of PCI Extended Capabilities registers. This bit is hardwired to 1.	0x1
23:21	R	Reserved [R5]	Reserved	0x0

Bits	SW	Name	Description	Reset
24	R/WOCLR	Master Data Parity Error [MDPE]	When the Parity Error Response enable bit is 1, the core sets this bit when it detects the following error conditions: (i) The core receives a poisoned completion from the link in response to a request. (ii) The core sends out a poisoned write request on the link (this may be because an underflow occurred during the packet transfer at the host interface of the core.). This bit remains 0 when the Parity Error Response enable bit is 0. This field can also be cleared from the local management bus by writing a 1 into this bit position.	0x0
26:25	R	Reserved [R6]	Reserved	0x0
27	R/WOCLR	Signaled Target Abort [STA]	This bit is set when the core has sent a completion to the link with the Completer Abort status. This field can also be cleared from the local management bus by writing a 1 into this bit position.	0x0
28	R/WOCLR	Received Target Abort [RTA]	This bit is set when the core has received a completion from the link with the Completer Abort status. This field can also be cleared from the local management bus by writing a 1 into this bit position	0x0
29	R/WOCLR	Received Master Abort [RMA]	This bit is set when the core has received a completion from the link with the Unsupported Request status. This field can also be cleared from the local management bus by writing a 1 into this bit position	0x0
30	R/WOCLR	Signaled System Error [SSE]	If the SERR enable bit is 1, this bit is set when the core has sent out a fatal or non-fatal error message on the link to the Root Complex. If the SERR enable bit is 0, this bit remains 0. This field can also be cleared from the local management bus by writing a 1 into this bit position.	0x0
31	R/WOCLR	Detected Parity Error [DPE]	This bit is set when the core has received a poisoned TLP. The Parity Error Response enable bit (bit 6) has no effect on the setting of this bit. This field can also be cleared from the local management bus by writing a 1 into this bit position.	0x0

17.6.4.1.3 Revision ID and Class Code Register

Propname：Revision ID and Class Code Register
Address：@0x8
Description：This register contains the Revision ID and Class Code associated with the

device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
7:0	R	Revision ID [RID]	Assigned by the manufacturer of the device to identify the revision number of the device. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be re- written independently for each Function from the local management bus.	8'h0
15:8	R	Programming Interface Byte [PIB]	Identifies the register set layout of the device. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be re- written independently for each Function from the local management bus.	8'h0
23:16	R	Sub-Class Code [SCC]	Identifies a sub-category within the selected function. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be re-written independently for each Function from the local management bus.	8'h0
31:24	R	Class Code [CC]	Identifies the function of the device. On power- up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be rewritten independently for each Function from the local management bus	8'h0

17.6.4.1.4 BIST, Header Type, Latency Timer and Cache Line Size Registers

Propname：BIST, Header Type, Latency Timer and Cache Line Size Registers
Address：@0xc
Description：This location contains the BIST, header-type, Latency Timer and Cache Line Size Registers.

Bits	SW	Name	Description	Reset
7:0	R/W	Cache Line Size [CLS]	Cache Line Size Register defined in PCI Specifications 3.0. This field can be read or written, both from the link and from the local management bus, but its value is not used.	0x0
15:8	R	Latency Timer [LT]	This is an unused field and is hardwired to 0.	0x0
22:16	R	Header Type [HT]	Identifies format of header. This field is hardwired to 0.	0x0
23	R	Device Type [DT]	Identifies whether the device supports a single Function or multiple Functions. This bit is read as 0 when only Function 0 has been enabled in the Physical Function Configuration Register (in the local management block), and as 1 when more than one Function has been enabled.	0x0

Bits	SW	Name	Description	Reset
31:24	R	BIST Register [BR]	BIST control register.It can be accessed using local management bus.	0x0

17.6.4.1.5 Base Address Register 0

Propname：Base Address Register 0
Address：@0x10
Description：This is one of the six Base Address Registers defined by the PCI Specifications 3.0. These registers are used to define address ranges for memory and I/O accesses to the Endpoint device. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1s into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if BAR 0 is not configured. Otherwise, the number of 1s returned is based on the size of the BAR. BAR0 can be setup as 32-bit memory or IO BAR, or can be paired with BAR 1 to form a 64bit memory BAR. The settings of this BAR is defined in the BAR Configuration Register associated with this PF. The BAR aperture can be controller in two different ways: (i) When the Resizable BAR Capability is enabled, the aperture is controlled by the setting of the BAR width field in Resizable BAR Control Register. The Resizable BAR Capability is enabled by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register 1. (ii) When the Resizable BAR Capability is disabled for the Physical Function, the aperture is controlled by the setting of the PF BAR Configuration Register.

Bits	SW	Name	Description	Reset
0	R	Memory Space Indicator [MSI0]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O BARs.	0x0
2	R	Size [S0]	When the BAR is used to define a memory address range, this field indicates whether the address range is 32-bit or 64-bit (0 = 32-bit, 1 = 64 bit). For 64-bit address ranges, the value in BAR 1 is treated as a continuation of the base address in BAR 0. The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function.	0x1
3	R	Prefetchability [P0]	When the BAR is used to define a memory address range, this field declares whether data from the address range is prefetchable (0 = non- prefetchable, 1 = prefetchable). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
7:4	R	Reserved [R8]	These bits are hardwired to 0	0x0

Bits	SW	Name	Description	Reset
21:8	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function.	0x0

17.6.4.1.6 Base Address Register 1

Propname：Base Address Register 1
Address：@0x14
Description：This is the one of the six Base Address Registers defined by the PCI Specifications 3.0. BAR1 can be setup as 32-bit memory or IO BAR, or can be paired with BAR 0 to form a 64bit memory BAR. This register can be used in two distinct ways: (i) When BAR 0 defines a 64-bit memory address range, this register is used to define the high-order bits of the base address. The number of writable bits in this field is based on the aperture setting of the BAR. (ii) When the BAR 0 is used to define a 32-bit memory address range or an I/O address range, this register can be used to define a new 32-bit memory address range or an I/O address range. The individual fields in the register have the same format as those of BAR 0 and is described below. The settings of this BAR is defined in the BAR Configuration Register associated with this PF. When configured as a 32-bit memory or IO BAR, the BAR aperture can be controller in two different ways: (a) When the Resizable BAR Capability is enabled, the aperture is controlled by the setting of the BAR width field in Resizable BAR Control Register 1. The Resizable BAR Capability is enabled by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. (b) When the Resizable BAR Capability is disabled for the Physical Function, the aperture is controlled by the setting of the Physical Function BAR Configuration Register.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture setting of BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

17.6.4.1.7 Base Address Register 2

Propname：Base Address Register 2
Address：@0x18
Description：This is one of the six Base Address Registers defined by the PCI Specifications 3.0. These registers are used to define address ranges for memory and I/O accesses to the Endpoint device. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1s into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if BAR 2 is not configured. Otherwise, the number of 1s returned is based on the size of the BAR. BAR2

can be setup as 32-bit memory or IO BAR, or can be paired with BAR 3 to form a 64bit memory BAR. The settings of this BAR is defined in the BAR Configuration Register associated with this PF. The BAR aperture can be controller in two different ways: (i) When the Resizable BAR Capability is enabled, the aperture is controlled by the setting of the BAR width field in Resizable BAR Control Register. The Resizable BAR Capability is enabled by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register 1. (ii) When the Resizable BAR Capability is disabled for the Physical Function, the aperture is controlled by the setting of the PF BAR Configuration Register.

Bits	SW	Name	Description	Reset
0	R	Memory Space Indicator [MSI0]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O BARs.	0x0
2	R	Size [S0]	When the BAR is used to define a memory address range, this field indicates whether the address range is 32-bit or 64-bit (0 = 32-bit, 1 = 64 bit). For 64-bit address ranges, the value in BAR 3 is treated as a continuation of the base address in BAR 2. The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function.	0x1
3	R	Prefetchability [P0]	When the BAR is used to define a memory address range, this field declares whether data from the address range is prefetchable (0 = non- prefetchable, 1 = prefetchable). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
7:4	R	Reserved [R8]	These bits are hardwired to 0	0x0
21:8	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

Bits	SW	Name	Description	Reset
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function.	0x0

17.6.4.1.8 Base Address Register 3

Propname：Base Address Register 3
Address：@0x1c
Description：This is the one of the six Base Address Registers defined by the PCI Specifications 3.0. BAR3 can be setup as 32-bit memory or IO BAR, or can be paired with BAR 2 to form a 64bit memory BAR. This register can be used in two distinct ways: (i) When BAR 2 defines a 64-bit memory address range, this register is used to define the high-order bits of the base address. The number of writable bits in this field is based on the aperture setting of the BAR. (ii) When the BAR 2 is used to define a 32-bit memory address range or an I/O address range, this register can be used to define a new 32-bit memory address range or an I/O address range. The individual fields in the register have the same format as those of BAR2 and is described below. The settings of this BAR is defined in the BAR Configuration Register associated with this PF. When configured as a 32-bit memory or IO BAR, the BAR aperture can be controller in two different ways: (a) When the Resizable BAR Capability is enabled, the aperture is controlled by the setting of the BAR width field in Resizable BAR Control Register 3. The Resizable BAR Capability is enabled by setting the Enable Resizable BAR Capability bit (bit 31)of the associated Physical Function BAR Configuration Register. (b) When the Resizable BAR Capability is disabled for the Physical Function, the aperture is controlled by the setting of the Physical Function BAR Configuration Register.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture setting of BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

17.6.4.1.9 Base Address Register 4

Propname：Base Address Register 4
Address：@0x20
Description：This is one of the six Base Address Registers defined by the PCI Specifications 3.0. These registers are used to define address ranges for memory and I/O accesses to the Endpoint device. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1s into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if BAR 4 is not configured. Otherwise, the number of 1s returned is based on the size of the BAR. BAR4 can be setup as 32-bit memory or IO BAR, or can be paired with BAR 5 to form a 64bit memory BAR. The settings of this BAR is defined in the BAR Configuration Register associated with this PF. The BAR aperture can be controller in two different ways: (i) When the Resizable BAR Capability is enabled, the aperture is controlled by the setting of the BAR width field in Resizable BAR Control Register. The Resizable BAR Capability is enabled by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register 1. (ii) When the Resizable BAR Capability is disabled for the Physical Function, the aperture is controlled by the setting of the PF BAR Configuration

Register.

Bits	SW	Name	Description	Reset
0	R	Memory Space Indicator [MSI0]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O	0x0
2	R	Size [S0]	When the BAR is used to define a memory address range, this field indicates whether the address range is 32-bit or 64-bit (0 = 32-bit, 1 = 64 bit). For 64-bit address ranges, the value in BAR 5 is treated as a continuation of the base address in BAR 4. The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function.	0x1
3	R	Prefetchability [P0]	When the BAR is used to define a memory address range, this field declares whether data from the address range is prefetchable (0 = non- prefetchable, 1 = prefetchable). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
7:4	R	Reserved [R8]	These bits are hardwired to 0	0x0
21:8	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function.	0x0

17.6.4.1.10 Base Address Register 5

Propname：Base Address Register 5

Address：@0x24

Description：This is the one of the six Base Address Registers defined by the PCI Specifications 3.0. BAR5 can be setup as 32-bit memory or IO BAR, or can be paired with BAR 4 to form a 64bit memory BAR. This register can be used in two distinct ways: (i) When BAR 4 defines a 64-bit memory address range, this register is used to define the

high-order bits of the base address. The number of writable bits in this field is based on the aperture setting of the BAR. (ii) When the BAR 4 is used to define a 32-bit memory address range or an I/O address range, this register can be used to define a new 32-bit memory address range or an I/O address range. The individual fields in the register have the same format as those of BAR 4 and is described below. The settings of this BAR is defined in the BAR Configuration Register associated with this PF. When configured as a 32-bit memory or IO BAR, the BAR aperture can be controller in two different ways: (a) When the Resizable BAR Capability is enabled, the aperture is controlled by the setting of the BAR width field in Resizable BAR Control Register 5. The Resizable BAR Capability is enabled by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. (b) When the Resizable BAR Capability is disabled for the Physical Function, the aperture is controlled by the setting of the Physical Function BAR Configuration Register.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address-RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture setting of BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

17.6.4.1.11 Reserved

Propname：Reserved
Address：@0x28
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.1.12 Subsystem Vendor ID and Subsystem ID Register

Propname：Subsystem Vendor ID and Subsystem ID Register
Address：@0x2c
Description：This register contains the Subsystem Vendor ID and Subsystem ID associated with the device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
15:0	R	Subsystem Vendor ID [SVID]	Specifies the Subsystem Vendor ID assigned by the PCI SIG to the manufacturer of the device. Its value comes from the Subsystem Vendor ID Register in the local management register block.	16'h17cd
31:16	R	Subsystem ID [SID]	Specifies the Subsystem ID assigned by the manufacturer of the device. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be re-written independently for each Function from the local management bus.	16'h0

17.6.4.1.13 Reserved

Propname：Reserved
Address：@0x30

Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.1.14 Capabilities Pointer

Propname：Capabilities Pointer
Address：@0x34
Description：This location contains the pointer to the first PCI Capabilities Structure. Its default value points to the Power Management Capability Structure (register number 080 hex).

Bits	SW	Name	Description	Reset
7:0	R	Capabilities Pointer [CP]	Contains pointer to the first PCI Capability Structure. This field is set by default to the value defined in the RTL file reg_defaults.h. It can be re-written independently for every Function from the local management bus.	0x80
31:8	R	Reserved [R15]	Reserved	0x0

17.6.4.1.15 Reserved

Propname：Reserved
Address：@0x38
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.1.16 Interrupt Line and Interrupt Pin Register

Propname：Interrupt Line and Interrupt Pin Register
Address：@0x3c
Description：This location contains the PCI 3.0 Interrupt Line and Interrupt Pin Registers. These registers are used only when the core is configured to support PCI legacy interrupts. If the legacy interrupt mode is configured, the core receives interrupt indications from the client logic on its INTA_IN, INTB_IN, INTC_IN and INTD_IN inputs, and sends out Assert_INTx or Deassert_INTx messages on the link in response to their activation or deactivation, respectively. The Interrupt Pin Register defines which of the four inputs is connected to the Function corresponding to this register set. The Interrupt Line register defines the input of the interrupt controller (IRQ0 - IRQ15) in the Root Complex that is activated by each Assert_INTx message.

Bits	SW	Name	Description	Reset
7:0	R/W	Interrupt Line Register [ILR]	Identifies the IRQx input of the interrupt controller at the Root Complex that is activated by this Functions interrupt (00 = IRQ0, ... , 0F = IRQ15, FF = unknown or not connected). This field is writable from the local management bus.	8'hff

Bits	SW	Name	Description	Reset
10:8	R	Interrupt Pin Register [IPR]	Identifies the interrupt input (A, B, C, D) to which this Functions interrupt output is connected to (01= INTA, 02 = INTB, 03 = INTC, 04 = INTD). The assignment of interrupt inputs to Functions is fixed when the core is configured. This field can be re-written independently for each Function from the local management bus.	0x1
31:11	R	Reserved [R16]	Reserved	0x0

17.6.4.1.17 Reserved

Propname：Reserved
Address：@0x40 + [0..15 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.2 i_power_mgmt_cap_struct

17.6.4.2.1 Power Management Capabilities Register

Propname：Power Management Capabilities Register
Address：@0x80
Description：This location contains the Power Management Capabilities Register, its Capability ID, and a pointer to the next capability. This version of the core supports the PCI power states D0, D1 and D3.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Identifies that the capability structure is for Power Management. This field is set by default to 01 hex. It can be re-written independently for each Function from the local management bus.	0x01
15:8	R	Capabilities Pointer [CP]	Contains pointer to the next PCI Capability Structure. The core sets it to the value defined in the RTL file reg_defaults.h. This field can be re-written independently for each Function from the local management bus.	8'h90
18:16	R	Version ID [VID]	Indicates the version of the PCI Bus Power Management Specifications that the Function implements. This field is set by default to 011 (Version 1.2). It can be re-written independently for each Function from the local management bus.	0x3
19	R	PME Clock [PC]	Not applicable to PCI Express. This bit is hardwired to 0.	0x0
20	R	Reserved [R0]	Reserved	0x0

Bits	SW	Name	Description	Reset
21	R	Device Specific Initialization Bit [DSI]	This bit, when set, indicates that the device requires additional configuration steps beyond setting up its PCI configuration space, to bring it to the D0 active state from the D0 uninitialized state. This bit is hardwired to 0.	0x0
24:22	R	Max Current Required from Aux Power Supply [MCRAPS]	Specifies the maximum current drawn by the device from the aux power source in the D3cold state. This field is not implemented in devices not supporting PME notification when in the D3cold state, and is therefore hardwired to 0.	0x0
25	R	D1 Support [D1S]	Set if the Function supports the D1 power state. This bit can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x1
26	R	D2 Support [D2S]	Set if the Function supports the D2 power state. Currently hardwired to 0.	0x0
27	R	PME Support for D0 State [PSD0S]	Indicates whether the Function is capable of sending PME messages when in the D0 state. This bit is set to 1 by default, but can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x01
28	R	PME Support for D1 State [PSD1S]	Indicates whether the Function is capable of sending PME messages when in the D1 state. This bit can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x1
29	R	PME Support for D2 State [PSD2S]	Indicates whether the Function is capable of sending PME messages when in the D2 state. This bit is hardwired to 0 because D2 state is not supported.	0x0
30	R	PME Support for D3(hot) Statue [PSDHS]	Indicates whether the Function is capable of sending PME messages when in the D3hot state. This bit is set to 1 by default, but can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x01

Bits	SW	Name	Description	Reset
31	R	PME Support for D3(cold) State [PSDCS]	Indicates whether the Function is capable of sending PME messages when in the D3cold state. Because the device does not have aux power, this bit is hardwired to 0.	0x0

17.6.4.2.2 Power Management Control/Status Report

Propname：Power Management Control/Status Report
Address：@0x84
Description：This location contains the Power Management Control/Status and Data Registers.

Bits	SW	Name	Description	Reset
1:0	R/W	Power State [PS]	Indicates the power state this Function is currently in. This field can be read by the software to monitor the current power state, or can be written to cause a transition to a new state. The valid settings are 00 (state D0), 01 (state D1) and 11 (state D3hot). The software should not write any other value into this field. This field can also be written from the local management bus independently for each Function.	0x0
2	R	Reserved [R4]	Reserved	0x0
3	R	No Soft Reset [NSR]	When this bit is set to 1, the Function will maintain all its state in the PM state D3hot. The software is not required to re-initialize the Function registers on the transition back to D0. This bit is set to 1 by default, but can be modified independently for each PF from the local management bus.	0x01
7:4	R	Reserved [R3]	Reserved	0x0
8	R/W	PME Enable [PE]	Setting this bit enables the notification of PME events from the associated Function. This bit can be set also by writing into this register from the local management bus.	0x0
14:9	R	Reserved [R2]	Reserved	0x0

Bits	SW	Name	Description	Reset
15	R/WOCLR	PME Status [PMES]	When PME notification is enabled, writing a 1 into this bit position from the local management bus sets this bit and causes the core to send a PME message from the associated Function. When the Root Complex processes this message, it will turn off this bit by writing a 1 into this bit position through a Config Write. This bit can be set or cleared from the local management bus, by writing a 1 or 0, respectively. It can only be cleared from the configuration path (by writing a 1).	0x0
23:16	R	Reserved [R1]	Reserved	0x0
31:24	R	Data Register [DR]	This optional register is not implemented in the PCIe core. This field is hardwired to 0.	0x0

17.6.4.2.3 Reserved

Propname：Reserved
Address：@0x88 + [0..1 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.3 i_MSI_cap_struct

17.6.4.3.1 MSI Control Register

Propname：MSI Control Register
Address：@0x90
Description：This register is used only when the core is configured to support Message Signaled Interrupts (MSIs). In addition to the MSI control bits, this location also contains the Capability ID for MSI and the pointer to the next PCI Capability Structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID1]	Specifies that the capability structure is for MSI. Hardwired to 05 hex.	0x05
15:8	R	Capabilities Pointer [CP1]	Pointer to the next PCI Capability Structure. This can be modified from the local management bus. This field can be written from the local management bus.	8'hb0
16	R/W	MSI Enable [ME]	Set by the configuration program to enable the MSI feature. This field can also be written from the local management bus.	0x0

Bits	SW	Name	Description	Reset
19:17	R	Multiple Message Capable [MMC]	Encodes the number of distinct messages that the core is capable of generating for this Function (000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101= 32). Thus, this field defines the number of the interrupt vectors for this Function. The core allows up to 32 distinct messages, but the setting of this field must be based on the number of interrupt inputs of the core that are actually used by the client. For example, if the client logic uses 8 of the 32 distinct MSI interrupt inputs of the core for this Function, then the value of this field must be set to 011. This field can be written from the local management bus.	0x0
22:20	R/W	Multiple Message Enable [MME]	Encodes the number of distinct messages that the core is programmed to generate for this Function (000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101= 32). This setting must be based on the number of interrupt inputs of the core that are actually used by this Function. This field can be written from the local management bus.	0x0
23	R	64-Bit Address Capable [BAC64]	Set to 1 to indicate that the device is capable of generating 64-bit addresses for MSI messages. Can be modified using local management interface	0x1
24	R	MSI masking capable [MC]	can be modified using local management interface	0x1
31:25	R	Reserved [R0]	Reserved	0x0

17.6.4.3.2 MSI Message Low Address Register

Propname：MSI Message Low Address Register
Address：@0x94
Description：This register contains the first 32 bits of the address to be used in the MSI messages generated by the core for this Function. This address is taken as a 32-bit address if the value programmed in the MSI Message High Address Register is 0. Otherwise, this address is taken as the least significant 32 bits of the 64-bit address sent in MSI messages.

Bits	SW	Name	Description	Reset
1:0	R	Reserved [R1]	The two lower bits of the address are hardwired to 0 to align the address on a double-word boundary.	0x0
31:2	R/W	Message Address Low [MAL]	Lower bits of the address to be used in MSI messages. This field can also be written from the local management bus.	0x0

17.6.4.3.3 MSI Message High Address Register

Propname：MSI Message High Address Register
Address：@0x98
Description：This register contains the most significant 32 bits of the 64-bit address sent

by the core in MSI messages. A value of all zeroes in this register is taken to mean that the core should use 32-bit addresses in the messages.

Bits	SW	Name	Description	Reset
31:0	R/W	Message Address High [MAH]	Contains bits 63:32 of the 64-bit address to be used in MSI Messages. A value of 0 specifies that 32-bit addresses are to be used in the messages. This field can also be written from the local management bus.	0x0

17.6.4.3.4 MSI Message Data Register

Propname：MSI Message Data Register

Address：@0x9c

Description This register contains the write data to be used in the MSI messages to be generated for the associated PCI Function. When the number of distinct messages programmed in the MSI Control Register is 1, the 32-bit value from this register is used as the data value in the MSI packets generated by the core for this Function. If the number of distinct messages is more than 1, the least significant bits of the programmed value are replaced with the encoded interrupt vector [31:0] of the specific message to generate the write data value for the message.

Bits	SW	Name	Description	Reset
15:0	R/W	Message Data [MD]	Message data to be used for this Function. This field can also be written from the local management bus.	0x0
31:16	R	Reserved [R2]	Hardwired to 0	0x0

17.6.4.3.5 MSI Mask Register

Propname：MSI Mask Register

Address：@0xa0

Description：This register contains the MSI mask bits, one for each of the interrupt levels.

Bits	SW	Name	Description	Reset
0	R/W	MSI Mask [MM]	Mask bits for MSI interrupts. The Multiple Message Capable field of the MSI Control Register specifies the number of distinct interrupts forthe Function, which determines the number of valid mask bits. Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of the MSI Mask field also changes correspondingly	0x0
31:1	R	Reserved [R0]	Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of this field also changes correspondingly	0x0

17.6.4.3.6 MSI Pending Bits Register

Propname：MSI Pending Bits Register

Address：@0xa4

Description：This register contains the MSI pending interrupt bits, one for each of the interrupt levels.

Bits	SW	Name	Description	Reset
0	R	MSI Pending Bits [MP]	Pending bits for MSI interrupts. This field can be written from the APB interface to reflect the current pending status. The Multiple Message Capable field of the MSI Control Register specifies the number of distinct interrupts for the Function, which determines the number of valid pending bits. Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of the MSI Pending Bits field also changes correspondingly	0x0
31:1	R	Reserved [R0]	Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of this field also changes correspondingly	0x0

17.6.4.3.7 Reserved

Propname：Reserved
Address：@0xa8 + [0..1 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.4 i_MSIX_cap_struct

17.6.4.4.1 MSI-X Control Register

Propname：MSI-X Control Register
Address：@0xb0
Description：This register contains the MSI-X configuration bits, the Capability ID for MSI-X and the pointer to the next PCI Capability Structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Identifies that the capability structure is for MSI-X. This field is set by default to 11 hex. It can be rewritten independently for each Function from the local management bus.	0x11
15:8	R	Capabilities Pointer [CP]	Contains pointer to the next PCI Capability Structure. This is set to point to the PCI Express Capability Structure at 30 hex. This can be rewritten independently for each Function from the local management bus.	8'hc0

Bits	SW	Name	Description	Reset
26:16	R	MSI-X Table Size [MSIXTS]	Specifies the size of the MSI-X Table, that is, the number of interrupt vectors definedfor the Function. The programmed value is 1 minus the size of the table (that is, this field is set to 0 if the table size is 1.). It can be re- written independently for each Function from the local management bus.	11'h0
29:27	R	Reserved [R0]	Reserved	0x0
30	R/W	Function Mask [FM]	This bit serves as a global mask to all the interrupt conditions associated with this Function. When this bit is set, the core will not send out MSI-X messages from this Function. This field can also be written from the local management bus.	0x0
31	R/W	MSI-X Enable [MSIXE]	Set by the configuration program to enable the MSI-X feature. This field can also be written from the local management bus.	0x0

17.6.4.4.2 MSI-X Table Offset Register

Propname：MSI-X Table Offset Register
Address：@0xb4
Description：This register is used to specify the location of the MSI-X Table in memory. All the 32 bits of this register can be re-written independently for each Function from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Indicator Register [BARI]	Identifies the BAR corresponding to the memory address range where the MSI-X Table is located (000 = BAR 0, 001 = BAR 1, ... , 101 = BAR 5).	3'd0
31:3	R	Table Offset [TO]	Offset of the memory address where the MSI- X Table is located, relative to the selected BAR. The three least significant bits of the address are omitted, as the addresses are QWORD aligned.	29'h0

17.6.4.4.3 MSI-X Pending Interrupt Register

Propname：MSI-X Pending Interrupt Register
Address：@0xb8
Description：This register is used to specify the location of the MSI-X Pending Bit Array (PBA). The PBA is a structure in memory containing the pending interrupt bits. All the 32 bits of this register can be rewritten independently for each Function from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Indicator Register [BARI1]	Identifies the BAR corresponding to the memory address range where the PBA Structure is located (000 = BAR 0, 001 = BAR 1, ... , 101 = BAR 5). The value programmed must be the same as the BAR Indicator configured in the MSI-X Table Offset Register.Identifies the BAR corresponding to the memory address range where the PBA Structure is located (000 = BAR 0, 001 = BAR1, ... , 101 = BAR 5). The value programmed must be the same as the BAR Indicator configured in the MSI-X Table Offset Register.	3'd0
31:3	R	PBA Offset [PBAO]	Offset of the memory address where the PBA is located, relative to the selected BAR. The three least significant bits of the address are omitted, as the addresses are QWORD aligned.	29'h1

17.6.4.4.4 Reserved

Propname：Reserved
Address：@0xbc
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.5 i_PCIE_cap_struct

17.6.4.5.1 PCI Express Capability List Register

Propname：PCI Express Capability List Register
Address：@0xc0
Description：This location identifies the PCI Express device type and its capabilities. It also contains the Capability ID for the PCI Express Structure and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Specifies Capability ID assigned by PCI SIG for this structure. This field is hardwired to 10 hex.	0x010
15:8	R	Next Capability Pointer [NCP]	Points to the next PCI capability structure. Set to 0 because this is the last capability structure.	0x0
19:16	R	Capability Version [PCV]	Identifies the version number of the capability structure. The value depends on the value of the strap input PCIE_GENERATION_SEL If PCIE_GENERATION_SEL indicates Gen 2 or later generations, then the value is 2 else 1. It can be modified through local management interface.	0x2

Bits	SW	Name	Description	Reset
23:20	R	Device Type [DT]	Indicates the type of device implementing this Function. This field is hardwired to 0 in the EP mode.	0x0
24	R	Slot Status [SS]	Set to 1 when the link connected to a slot. Hardwired to 0.	0x0
29:25	R	Interrupt Message Number [IMN]	Identifies the MSI or MSI-X interrupt vector for the interrupt message generated corresponding to the status bits in the Slot Status Register, Root Status Register, or this capability structure. This field must be defined based on the chosen interrupt mode - MSI or MSI-X. This field is hardwired to 0.	0x0
30	R	TCS Routing Supported [TRS]	When set to 1, this bit indicates that the device supports routing of Trusted Configuration Requests. Not valid for Endpoints. Hardwired to 0.	0x0
31	R	Reserved [R0]	Reserved	0x0

17.6.4.5.2 PCI Express Device Capabilities Register

Propname：PCI Express Device Capabilities Register

Address：@0xc4

Description：This register advertises the capabilities of the PCI Express device encompassing this Function.

Bits	SW	Name	Description	Reset
2:0	R	Max Payload Size [MPS]	Specifies maximum payload size supported by the device.	3'b001
4:3	R	Phantom Functions Supported [PFS]	This field is used to extend the tag field by combining unused Function bits with the tag bits. This field is hardwired to 00 to disable this feature.	0x0
5	R	Extended Tag Field Supported [ETFS]	Extended Tag Field Not Supported. Hard coded to 0.	0x0
8:6	R	Acceptable L0S Latency [AL0SL]	Specifies acceptable latency that the Endpoint can tolerate while transitioning from L0S to L0. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x4
11:9	R	Acceptable L1 Latency [AL1SL]	Specifies acceptable latency that the Endpoint can tolerate while transitioning from L1 to L0. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x0
14:12	R	Reserved [R1]	Reserved	0x0

Bits	SW	Name	Description	Reset
15	R	Role-Based Error Reporting [RBER]	Enables role-based error reporting. It is hardwired to 1.It can be re-written independently for each Function from the local management bus.	0x01
17:16	R	Reserved [R2]	Reserved	0x0
25:18	R	Captured Slot Power Limit Value [CSPLV]	Specifies upper limit on power supplied by slot. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x0
27:26	R	Captured Power Limit Scale [CPLS]	Specifies the scale used by Slot Power Limit Value. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x0
28	R	FLR Capable [FC]	Set when device has Function-Level Reset capability. It is set by default to 1. It can be re- written independently for each Function from the local management bus.	0x01
31:29	R	Reserved [R3]	Reserved	0x0

17.6.4.5.3 PCI Express Device Control and Status Register

Propname：PCI Express Device Control and Status Register
Address：@0xc8
Description：This register contains control and status bits associated with the device implementing this Function. All the read-write bits in this register can also be written from the local management bus. Likewise, bits designated as RW1C can also be cleared by writing a 1 from the local management bus.

Bits	SW	Name	Description	Reset
0	R/W	Enable Correctable Error Reporting [ECER]	Enables the sending of ERR_COR messages by the core on the detection of correctable errors.	0x0
1	R/W	Enable Non-Fatal Error Reporting [ENFER]	Enables the sending of ERR_NONFATAL messages by the core on the detection of non-fatal errors.	0x0
2	R/W	Enable Fatal Error Reporting [EFER]	Enables the sending of ERR_FATAL messages by the core on the detection of fatal errors.	0x0
3	R/W	Enable Unsupported Request Reporting [EURR]	Enables the sending of error messages by the core on receiving unsupported requests.	0x0

Bits	SW	Name	Description	Reset
4	R/W	Enable Relaxed Ordering [ERO]	When set, this bit indicates that the device is allowed to set the Relaxed Ordering bit in the Attributes field of transactions initiated from it, when the transactions do not require Strong Ordering.	0x01
7:5	R/W	Max Payload Size [MPS]	Specifies the maximum TLP payload size configured. The device must be able to receive a TLP of this maximum size, and should not generate TLPs larger than this value. The configuration program sets this field based on the maximum payload size in the Device Capabilities Register, and the capability of the other side.	0x0
8	R	Extended Tag Field Enable [ETFE]	Enables the extension of the tag field from 5 to 8 bits.	0x0
9	R	Enable Phantom Functions [EPH]	This field is hardwired to 0 as the core does not support this feature.	0x0
10	R	Enable Aux Power [EAP]	Used only when device used aux power. This field is hardwired to 0.	0x0
11	R/W	Enable No Snoop [ENS]	When set to 1, the device is allowed to set the No Snoop bit in initiated transactions in which cache coherency is not needed.	0x1
14:12	R/W	Max Read Request Size [MRRS]	Specifies the maximum size allowed in read requests generated by the device.	0x02
15	R/W	Function-Level Reset [FLR]	Writing a 1 into this bit position generates a Function-Level Reset for the selected Function. This bit reads as 0.	0x0
16	R/WOCLR	Correctable Error Detected [CED]	Set to 1 by the core when it detects a correctable error, regardless of whether error reporting is enabled or not, and regardless of whether the error is masked.	0x0
17	R/WOCLR	Non-Fatal Error Detected [NFED]	Set to 1 by the core when it detects a non-fatal error, regardless of whether error reporting is enabled or not, and regardless of whether the error is masked.	0x0
18	R/WOCLR	Fatal Error Detected [FED]	Set to 1 by the core when it detects a fatal error, regardless of whether error reporting is enabled or not, and regardless of whether the error is masked.	0x0
19	R/WOCLR	Unsupported Request Detected [URD]	Set to 1 by the core when it receives an unsupported request, regardless of whether its reporting is enabled or not.	0x0

Bits	SW	Name	Description	Reset
20	R	Aux Power Detected [APD]	Set when auxiliary power is detected by the device. This is an unused field.	0x0
21	R	Transaction Pending [TP]	Indicates if any of the Non-Posted requests issued by the Function are still pending.	0x0
31:22	R	Reserved [R4]	Reserved	0x0

17.6.4.5.4 Link Capabilities Register

Propname：Link Capabilities Register
Address：@0xcc
Description：This register advertises the link-specific capabilities of the device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
3:0	R	Maximum Link Speed [MLS]	Indicates the maximum speed supported by the link. (2.5 GT/s, 5 GT/s per lane). This field is hardwired to 0001 (2.5GT/s) when the strap input PCIE_GENERATION_SEL is set to 0, to 0010 (5 GT/s) when the strap is set to 1.	0x2
9:4	R	Maximum Link Width [MLW]	Indicates the maximum number of lanes supported by the device. This field is hardwired based on the setting of the LANE_COUNT_IN strap input.	0x4
11:10	R	Active State Power Management [ASPM]	Indicates the level of ASPM support provided by the device. This field can be re-written independently for each Function from the local management bus. When SRIS is enabled in local management register bit, L0s capability is not supported and is forced low.	0x3
14:12	R	L0S Exit Latency [L0SEL]	Specifies the time required for the device to transition from L0S to L0. This parameter is dependent on the Physical Layer implementation. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x2
17:15	R	L1 Exit Latency [L1EL]	Specifies the exit latency from L1 state. This parameter is dependent on the Physical Layer implementation. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x3

Bits	SW	Name	Description	Reset
18	R	Clock Power Management [CPM]	Indicates that the device supports removal of referenc clocks. It is set by default to the value of the define in reg_defaults.h. It can be re-written independently for each function from the local management bus.	0x0
19	R	Surprise Down Error Reporting Capability [SDERC]	Indicates the capability of the device to report a Surprise Down error condition. This bit is hardwired to 0, as this version of the core does not support the feature.	0x0
20	R	Data Link Layer Active Reporting Capability [DLLARC]	Set to 1 if the device is capable of reporting that the DL Control and Management State Machine has reached the DL_Active state. This bit is hardwired to 0, as this version of the core does not support the feature.	0x0
21	R	Link Bandwidth Notification Capability [LBNC]	A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. Reserved for Endpoint.	0x0
22	R	ASPM Optionality Compliance [AOC]	Setting this bit indicates that the device supports the ASPM Optionality feature. It can be turned off by writing a 0 to this bit position through the local management bus.	0x1
23	R	Reserved [R5]	Reserved	0x0
31:24	R	Port Number [PN]	Specifies the port number assigned to the PCI Express link connected to this device.	8'h0

17.6.4.5.5 Link Control and Status Register

Propname：Link Control and Status Register
Address：@0xd0
Description：This register contains control and status bits specific to the PCI Express link. All the read-write bits in this register can also be written from the local management bus.

Bits	SW	Name	Description	Reset
1:0	R/W	Active State Power Management Control [ASPMC]	Controls the level of ASPM support on the PCI Express link associated with this Function. The valid setting are 00: ASPM disabled 01: L0s entry enabled, L1 disabled 10: L1 entry enabled, L0s disabled 11: Both L0s and L1 enabled.	0x0
2	R	Reserved [R6]	Reserved	0x0

Bits	SW	Name	Description	Reset
3	R/W	Read Completion Boundary [RCB]	Indicates the Read Completion Boundary of the Root Port connected to this Endpoint (0 = 64 bytes, 1 = 128 bytes). This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
4	R	Link Disable [LD]	Writing a 1 to this bit position causes the LTSSM to go to the Disable Link state. The LTSSM stays in the Disable Link state while this bit is set. Reserved for Endpoint mode.	0x0
5	R	Retrain Link [RL]	Setting this bit to 1 causes the LTSSM to initiate link training. Reserved for Endpoint mode. This bit always reads as 0	0x0
6	R/W	Common Clock Configuration [CCC]	A value of 0 indicates that the reference clock of this device is asynchronous to that of the upstream device. A value of 1 indicates that the reference clock is common.	0x0
7	R/W	Extended Synch [ES]	Set to 1 to extend the sequence of ordered sets transmitted while exiting from the L0S state.	0x0
8	R	Enable Clock Power Management [ECPM]	When this bit is set to 1, the device may use the CLKREQ# pin on the PCIe connector to power manage the Link clock. This bit is writeable only when the Clock Power Management bit in the Link Capability Register is set to 1.	0x0
9	R	Reserved [R9]	Reserved	0x0
10	R	Link Bandwidth Management Interrupt Enable [LBMIE]	When Set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been Set. This enables an interrupt to be generated through PHY INTERRUPT_OUT if triggered. Hardwired to 0 if Link Bandwidth Notification Capability is 0. Not applicable to Endpoints where field is hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
11	R	Link Autonomous Bandwidth Interrupt Enable [LABIE]	When Set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been Set. This enables an interrupt to be generated through PHY_INTERRUPT_OUT if triggered. Hardwired to 0 if Link Bandwidth Notification Capability is 0. Not applicable to Endpoints where field is hardwired to 0.	0x0
15:12	R	Reserved [R15_12]	Reserved	0x0
19:16	R	Negotiated Link Speed [NLS]	Negotiated link speed of the device. The only supported speed ids are 2.5 GT/s per lane (0001),5 GT/s per lane (0010).	0x2
25:20	R	Negotiated Link Width [NLW]	Set at the end of link training to the actual link width negotiated between the two sides. Value is undefined if this register is accessed before link training.	0x4
26	R	Reserved [R8]	Reserved	0x0
27	R	Link Training Status [LTS]	This read-only bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state, or that 1b was written to the Retrain Link bit but Link training has not yet begun. Hardware clears this bit when the LTSSM exits the Configuration/ Recovery state. Not applicable to Endpoints where field is hardwired to 0.	0x0
28	R	Slot Clock Configuration [SCC]	Indicates that the device uses the reference clock provided by the connector.	0x0
29	R	Data Link Layer Active [DLLA]	Indicates the status of the Data Link Layer. Set to 1 when the DL Control and Management State Machine has reached the DL_Active state. This bit is hardwired to 0 in this version of the core.	0x0
30	R/WOCLR	Link Bandwidth Management Status [LBMS]	This bit is Set by hardware to indicate that either link training has completed following write to retrain link bit, or when HW has changed link speed or width to attempt to correct unreliable link operation. This triggers an interrupt to be generated through PHY_INTERRUPT_OUT if enabled. Hardwired to 0 if Link Bandwidth Notification Capability is 0. Not applicable to Endpoints where field is hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
31	R/WOCLR	Link Autonomous Bandwidth Status [LABS]	This bit is Set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This triggers an interrupt to be generated through PHY_INTERRUPT_OUT if enabled. Hardwired to 0 if Link Bandwidth Notification Capability is 0. Not applicable to Endpoints where field is hardwired to 0.	0x0

17.6.4.5.6 Reserved

Propname：Reserved
Address：@0xd8
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.5.7 Reserved

Propname：Reserved
Address：@0xdc + [0..1 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.5.8 PCI Express Device Capabilities Register 2

Propname：PCI Express Device Capabilities Register 2
Address：@0xe4
Description：This register advertises the capabilities of the PCI Express device encompassing this Function.

Bits	SW	Name	Description	Reset
3:0	R	Completion Timeout Ranges [CTR]	Specifies the Completion Timeout values supported by the device. This field is set by default to 0010 (10 ms - 250 ms). The actual timeout values are in two programmable local management registers, which allow the timeout settings of the two sub-ranges within Range B to be programmed independently.	0x02
4	R	Completion Timeout Disable Supported [CTDS]	A 1 in this field indicates that the associated Function supports the capability to turn off its Completion timeout. This bit is set to 1 by default, but can be re-written independently for each Function from the local management bus.	0x01

Bits	SW	Name	Description	Reset
5	R	ARI forwarding support [AFS]	ARI forwarding supported.	0x0
6	R	OP routing supported [OPRS]	Atomic OP routing supported.	0x0
7	R	32-Bit Atomic Op Completer Supported [BAOCS32]	Hardwired to 0.	0x0
8	R	64-Bit Atomic Op Completer Supported [BAOCS64]	Hardwired to 0.	0x0
9	R	128-Bit CAS Atomic Op Completer Supported [BAOCS128]	Hardwired to 0.	0x0
10	R	Reserved [R12]	Reserved	0x0
11	R	LTR Mechanism Supported [LMS]	A 1 in this bit position indicates that the Function supports the Latency Tolerance Reporting (LTR) Capability. This bit is set to 1 by default, but can be turned off for all Physical Functions by writing into PF 0.	0x01
13:12	R	TPH Completer Supported [TCS]	These bits, when set, indicate that the Function is capable of serving as a completer for requests with Transaction Processing Hints (TPH). It can be turned off for all Physical Functions by writing into PF 0. Defined Encodings are: 00b TPH and Extended TPH Completer not supported. 01b TPH Completer supported; Extended TPH Completer not supported. 10b Reserved. 11b Both TPH and Extended TPH Completer supported.	0x01
17:14	R	Reserved [R13]	Reserved	0x0
19:18	R	OBFF Supported [OPFFS]	A 1 in this bit position indicates that the Function supports the Optimized Buffer Flush/Fill (OBFF) capability using message signaling.	0x1
20	R	Extended Format Field Supported [EXFS]	Indicates that the Function supports the 3-bit definition of the Fmt field in the TLP header. This bit is hardwired to 1 for all Physical Functions.	0x0

Bits	SW	Name	Description	Reset
21	R	End-End TLP Prefix Supported [EEPS]	Indicates whether the Function supports End-End TLP Prefixes. A 1 in this field indicates that the Function supports receiving TLPs containing End- End TLP Prefixes.	0x0
23:22	R	Max End-End TLP Prefixes [MEEP]	Indicates the maximum number of End-End TLP Prefixes supported by the Function. The supported values are: 01b 1 End-End TLP Prefix 10b 2 End- End TLP Prefixes	0x0
31:24	R	Reserved [R14]	Reserved	0x0

17.6.4.5.9 PCI Express Device Control and Status Register 2

Propname：PCI Express Device Control and Status Register 2
Address：@0xe8
Description：This register contains control and status bits associated with the device implementing this Function.

Bits	SW	Name	Description	Reset
3:0	R/W	Completion Timeout Value [CTV]	Specifies the Completion Timeout value for the device. Allowable values are 0101 (sub-range 1) and 0110 (sub-range 2). The corresponding timeout values are stored in the local management registers Completion Timeout Interval Registers 0 and 1, respectively.	0x0
4	R/W	Completion Timeout Disable [CTD]	Setting this bit disables Completion Timeout in the device. This bit can also be written from the local management bus.	0x0
5	R	ARI forwarding enable [AFE]	ARI forwarding enable	0x0
6	R	Atomic Op Requester Enable [AORE]	This bit must be set to enable the generation of Atomic Op Requests from the Function. If the client logic attempts to send an Atomic Op for a Function for which this bit is not set, logic in the core will nullify the TLP on its way to the link. This bit can also be written from the local management bus.	0x0
7	R	Reserved [R16]	Reserved	0x0
8	R	IDO Request Enable [IDORE]	When this bit is 1, the Function is allowed to set the ID-based Ordering (IDO) Attribute bit in the requests it generates.	0x0
9	R	IDO Completion Enable [IDOCE]	When this bit is 1, the Function is allowed to set the ID-based Ordering (IDO) Attribute bit in the Completions it generates.	0x0

Bits	SW	Name	Description	Reset
10	R/W	LTR Mechanism Enable [LTRME]	This must be set to 1 to enable the Latency Tolerance Reporting Mechanism. This bit is implemented only in PF 0. Its default value is 1, but can be modified from the local management bus. This bit is read-only in PF 1.	0x0
12:11	R	Reserved [R17]	Reserved	0x0
14:13	R/W	OBFF Enable [OBFFE]	Enables the Optimized Buffer Flush/Fill (OBFF) capability in the device. This field is implemented only in PF 0. Valid settings are 00 (disabled), 01 (Variation A) and 10 (Variation B). This field can also be written from the local management bus. RW if OBFF capability is supported, RO otherwise.	0x0
31:15	R	Reserved [R18]	Reserved	0x0

17.6.4.5.10 Link Capabilities Register 2

Propname：Link Capabilities Register 2

Address：@0xec

Description：This register advertises the supported link speeds of the core.

Bits	SW	Name	Description	Reset
0	R	RSVD	RSVD	1'h0
2:1	R	Supported Link Speeds Vector [SLSV]	This field indicates the supported link speeds of the core. For each bit, a value of 1 indicates that the corresponding link speed is supported, while a value of 0 indicates that the corresponding speed is not supported. The bits corresponding to various link speeds are: Bit 1 = Link Speed 2.5 GT/s, Bit 2= Link Speed 5 GT/s, Bit 3 = Link Speed 8 GT/s. This field is hardwired to 001 (2.5 GT/s) when the PCIE_GENERATION_SEL strap pins of the core are set to 0, 011 (2.5 and 5 GT/s) when the strap is set to 1. This field is RsvrdP for the selected configuration.	0x0
31:3	R	RSVD	RSVD	29'h00000000

17.6.4.5.11 Link Control and Status Register 2

Propname：Link Control and Status Register 2

Address：@0xf0

Description：This register contains control and status bits specific to the PCI Express link. All the fields marked RW or RW(STICKY) can also be written from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R/W	Target Link Speed [TLS]	For an upstream component, this field sets an upper limit on Link operational speed during reconfiguration. Additionally for both upstream and downstream components, this field sets the target speed when the software forces the link into Compliance mode by setting the Enter Compliance bit in this register (0001 = 2.5 GT/ s, 0010 = 5 GT/s, 0100 = 8 GT/s). The default value of this field is 0001 (2.5 GT/s) when the PCIE_GENERATION_SEL strap pins of the core are set to 0, 0010 (5 GT/s) when the strap is set to 1. These bits are STICKY.	4'd2
4	R/W	Enter Compliance [EC]	into the Compliance mode. The target speed for the Compliance mode is determined by the Target Link Speed field of this register. STICKY.	0x0
5	R/W	Hardware Autonomous Speed Disable [HASD]	When this bit is set, the LTSSM is prevented from changing the operating speed of the link, other than reducing the speed to correct unreliable operation of the link. STICKY	0x0
6	R	Selectable De-emphasis [SDE]	This bit selects the de-emphasis level when the core is operating at 5 GT/s (0 = -6 dB, 1 = -3.5 dB). This is reserved for Endpoints.	0x0
9:7	R/W	Transmit Margin [TM]	This field is intended for debug and compliance testing purposes only. It controls the non-de- emphasized voltage level at the transmitter outputs. Its encodings are: 000: Normal operating range. 001: 800 - 1200 mV for full swing and 400 - 700 mV for half swing. 010 - 111: See PCI Express Base Specification 2.0. This field is reset to 0 when the LTSSM enters the Polling Configuration substate during link training. STICKY.	0x0
10	R/W	Enter Modified Compliance [EMC]	This field is intended for debug and compliance testing purposes only. If this bit is set to 1, the device will transmit the Modified Compliance Pattern when the LTSSM enters the Polling. Compliance substate. STICKY.	0x0
11	R/W	Compliance SOS [CS]	When this bit is set to 1, the device will transmit SKP ordered sets between compliance patterns. STICKY.	0x0

Bits	SW	Name	Description	Reset
15:12	R/W	Compliance De-Emphasis [CDE]	This bit sets the de-emphasis level (for 5GT/s operation) or the Transmitter Preset level (for 8 GT/s operation) when the LTSSM enters the Polling Compliance state because of software setting the Enter Compliance bit in this register. It is used only when the link is running at 5 GT/s or 8 GT/s. At 5 GT/s, the only valid setting are 0 (-6dB) and 1 (-3.5 dB). STICKY.	0x0
16	R	Current De-Emphasis Level [CDEL]	This status bit indicates the current operating de- emphasis level of the transmitter (0 = -6 dB, 1 = -3.5 dB).This field is undefined when link is not at Gen2 speed.	0x0
21:17	R	Reserved [R20]	Reserved	0x0
31:22	R	Reserved [R19]	Reserved	0x0

17.6.4.5.12 Reserved

Propname：Reserved
Address：@0xf4 + [0..2 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.6 i_AER_cap_struct

17.6.4.6.1 Advanced Error Reporting (AER) Enhanced Capability Header Register

Propname：Advanced Error Reporting (AER) Enhanced Capability Header Register
Address：@0x100
Description：This is the first register in the PCI Express Advanced Error Reporting Capability Structure. This register contains the PCI Express Extended Capability ID, the capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PEECI]	This field is hardwired to the Capability ID assigned by PCI SIG to the PCI Express AER Extended Capability Structure (0001 hex).	0x01
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 4'h2, but can be modified from the local management bus.	4'h2
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h140

17.6.4.6.2 Uncorrectable Error Status Register

Propname：Uncorrectable Error Status Register
Address：@0x104
Description：This register provides the status of the various uncorrectable errors detected by the PCI Express core. Software may clear any error bit by writing a 1 into the corresponding bit position. The states of the bits in the Uncorrectable Error Mask Register have no effect on the status bits of this register. The setting of an uncorrectable error status bit causes the core to generate an ERR_FATAL message if the corresponding severity bit of the Uncorrectable Error Severity Register is 1. If the severity bit is 0, however, there are two separate ways the error could be processed:(i) In certain cases, the uncorrectable error is treated as an Advisory Non-Fatal Error. These cases are treated as similar to correctable errors, causing the core to generate an ERR_COR message instead of an ERR_NONFATAL message. For details on these special cases, refer to Section 6.2.3.2.4 of the PCI Express Base Specifications, Version 1.1. (ii) In all other cases, the core sends an ERR_NONFATAL message when the error is detected. In all cases, the sending of the error message can be suppressed by setting the bit corresponding to the error type in the Uncorrectable Error Mask Register. For errors that are not Function-specific, the error status bus is set in the registers belonging to all the Functions associated with the link, but only a single message is generated for the entire link. In the case of certain errors detected by the Transaction Layer, the associated TLP header is logged in the Header Log Registers. All the RW1C bits can also be cleared from the local management bus by writing a 1 into the bit position.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R/WOCLR	Data Link Protocol Error Status [DLPES]	This bit is set when the core receives an Ack or Nak DLLP whose sequence number does not correspond to that of an unacknowledged TLP or that of the last acknowledged TLP (for details, refer to PCI Express Base Specification 1.1, Section 3.5.2). This error is not Function-specific, and is reported by Function 0. STICKY.	0x0
11:5	R	Reserved [R1]	Reserved	0x0
12	R/WOCLR	Poisoned TLP Status [PTS]	This bit is set when the core receives a poisoned TLP from the link. This error is Function-specific. This error is considered non-fatal by default. The error is reported by sending an ERR_NONFATAL message. The header of the received TLP with error is logged in the Header Log Registers. STICKY.	0x0
13	R/WOCLR	Flow Control Protocol Error Status [FCPES]	This bit is set when certain violations of the flow control protocol are detected by the core. See Section 2.6.1 for details. This error is not Function- specific STICKY.	0x0

Bits	SW	Name	Description	Reset
14	R/WOCLR	Completion Timeout Status [CTS]	This bit is set when the completion timer associated with an outstanding request times out. This error is Function-specific. This error is considered non-fatal by default. STICKY.	0x0
15	R/WOCLR	Completer Abort Status [CAS]	This bit is set when the core has returned the Completer Abort (CA) status to a request received from the link. This error is Function-specific. The header of the received request that caused the error is logged in the Header Log Registers. STICKY.	0x0
16	R/WOCLR	Unexpected Completion Status [UCS]	This bit is set when the core has received an unexpected Completion packet from the link. This error is not Function-specific. STICKY.	0x0
17	R/WOCLR	Receiver Overflow Status [ROS]	This bit is set when the core receives a TLP in violation of the receive credit currently available. This error is not Function-specific. STICKY.	0x0
18	R/WOCLR	Malformed TLP Status [MTS]	This bit is set when the core receives a malformed TLP from the link. This error is not Function-specific. This error is considered fatal by default, and is reported by sending an ERR_FATAL message. The header of the received TLP with error is logged in the Header Log Registers. STICKY.	0x0
19	R/WOCLR	ECRC Error Status [EES]	This bit is set when the core has detected an ECRC error in a received TLP. This error is not Function-specific. The header of the received TLP with error is logged in the Header Log Registers. STICKY.	0x0
20	R/WOCLR	Unsupported Request Error Status [URES]	This bit is set when the core has received a request from the link that it does not support. This error is not Function-specific. This error is considered non-fatal by default. In the special case described in Sections 6.2.3.2.4.1 of the PCI Express Specifications, the error is reported by sending an ERR_COR message. In all other cases, the error is reported by sending an ERR_NONFATAL message. The header of the received request that caused the error is logged in the Header Log Registers. STICKY.	0x0

Bits	SW	Name	Description	Reset
21	R	Reserved [R2]	Reserved	0x0
22	R/WOCLR	Uncorrectable Internal Error Status [UIES]	This bit is set when the core has detected an internal uncorrectable error (HAL parity error or an uncorrectable ECC error while reading from any of the RAMs). This bit is also set in response to the client signaling an internal error through the input UNCORRECTABLE_ERROR_IN. This error is not Function-specific. This error is considered fatal by default, and is reported by sending an ERR_FATAL message. STICKY.	0x0
31:23	R	Reserved [R3]	(no description)	0x0

17.6.4.6.3 Uncorrectable Error Mask Register

Propname：Uncorrectable Error Mask Register
Address：@0x108
Description：The mask bits in this register control the reporting of uncorrectable errors. For each error type in the Uncorrectable Error Status Register, there is a corresponding bit in this register to mask its reporting. Setting the mask bit has the following effects: (i) The occurrence of the error is not reported to the Root Complex (by a PCI Express error message). (ii) The header of the TLP in which the error was detected is not logged in the Header Log Registers. (iii) The First Error Pointer in the Advanced Error Capabilities and Control Register is not updated on detection of the error. The individual bits of the mask register are described below. The bits marked RW can also be written from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R4]	Reserved	0x0
4	R/W	Data Link Protocol Error Mask [DLPEM]	This bit is set to mask the reporting of Data Link Protocol Errors. STICKY.	0x0
11:5	R	Reserved [R5]	Reserved	0x0
12	R/W	Poisoned TLP Mask [PTM]	This bit is set to mask the reporting of a Poisoned TLP. STICKY.	0x0
13	R/W	Flow Control Protocol Error Mask [FCPEM]	This bit is set to mask the reporting of Flow Control Protocol Errors. STICKY.	0x0
14	R/W	Completion Timeout Mask [CTM]	This bit is set to mask the reporting of Completion Timeouts. STICKY.	0x0
15	R/W	Completer Abort Mask [CAM]	This bit is set to mask the reporting of the core sending a Completer Abort. STICKY.	0x0
16	R/W	Unexpected Completion Mask [UCM]	This bit is set to mask the reporting of unexpected Completions received by the core. STICKY.	0x0

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R4]	Reserved	0x0
4	R/W	Data Link Protocol Error Mask [DLPEM]	This bit is set to mask the reporting of Data Link Protocol Errors. STICKY.	0x0
11:5	R	Reserved [R5]	Reserved	0x0
12	R/W	Poisoned TLP Mask [PTM]	This bit is set to mask the reporting of a Poisoned TLP. STICKY.	0x0
17	R/W	Receiver Overflow Mask [ROM]	This bit is set to mask the reporting of violations of receive credit. STICKY.	0x0
18	R/W	Malformed TLP Mask [MTM]	This bit is set to mask the reporting of malformed TLPs received from the link. STICKY.	0x0
19	R/W	ECRC Error Mask [EEM]	This bit is set to mask the reporting of ECRC errors. STICKY.	0x0
20	R/W	Unsupported Request Error Mask [UREM]	This bit is set to mask the reporting of unexpected requests received from the link. STICKY.	0x0
21	R	Reserved [R6]	Reserved	0x0
22	R/W	Uncorrectable Internal Error Mask [UIEM]	This bit is set to mask the reporting of internal errors. STICKY.	0x1
31:23	R	Reserved [R7]	Reserved	0x0

17.6.4.6.4 Uncorrectable Error Severity Register

Propname：Uncorrectable Error Severity Register

Address：@0x10c

Description：The setting of this register determines whether an uncorrectable error is reported as a fatal error on non-fatal error to the Root Complex. If a severity bit of this register is 0, the corresponding error is reported by the core using an ERR_NONFATAL message. Otherwise, it is reported using an ERR_FATAL message. The bits marked RW can also be written from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R8]	Reserved	0x0
4	R/W	Data Link Protocol Error Severity [DLPER]	Severity of Data Link Protocol Errors (0 = Non-Fatal, 1 = Fatal). STICKY.	0x01
5	R	Surprise Down Error Severity [SDES]	hard coded to 1	0x1
11:6	R	Reserved [R10]	Reserved	0x0
12	R/W	Poisoned TLP Severity [PTS]	Severity of a Poisoned TLP error (0 = Non-Fatal, 1= Fatal). STICKY.	0x0

Bits	SW	Name	Description	Reset
13	R/W	Flow Control Protocol Error Severity [FCPES]	Severity of Flow Control Protocol Errors (0 = Non- Fatal, 1 = Fatal). STICKY.	0x01
14	R/W	Completion Timeout Severity [CTS]	Severity of Completion Timeouts (0 = Non-Fatal, 1= Fatal). STICKY.	0x0
15	R/W	Completer Abort Severity [CAS]	Severity of sending a Completer Abort (0 = Non- Fatal, 1 = Fatal). STICKY.	0x0
16	R/W	Unexpected Completion Severity [UCS]	Severity of unexpected Completions received by the core (0 = Non-Fatal, 1 = Fatal). STICKY.	0x0
17	R/W	Receiver Overflow Severity [ROS]	Severity of receive credit violations (0 = Non- Fatal, 1 = Fatal). STICKY.	0x01
18	R/W	Malformed TLP Severity [MTS]	Severity of malformed TLPs received from the link (0 = Non- Fatal, 1 = Fatal). STICKY.	0x01
19	R/W	ECRC Error Severity [EES]	Severity of ECRC errors (0 = Non-Fatal, 1 = Fatal). STICKY.	0x0
20	R/W	Unsupported Requeset Error Severity [URES]	Severity of unexpected requests received from the link (0 = Non-Fatal, 1 = Fatal). STICKY.	0x0
21	R	Reserved [R11]	Reserved	0x0
22	R/W	Uncorrectable Internal Error Severity [UIES]	Severity of internal errors (0 = Non-Fatal, 1 = Fatal). STICKY.	0x01
31:23	R	Reserved [R12]	Reserved	0x0

17.6.4.6.5 Correctable Error Status Register

Propname：Correctable Error Status Register
Address：@0x110
Description This register provides the status of the various correctable errors detected by the PCI Express core. Software may clear any error bit by writing a 1 into the corresponding bit position. The states of the bits in the Correctable Error Mask Register have no effect on the status bits of this register. The setting of a correctable error status bit causes the core to generate an ERR_COR error message to the Root Complex if the error is not masked in the Correctable Error Mask Register. For errors that are not Function-specific, the error status bus is set in the registers belonging to all the Functions associated with the link, but only a single message is generated for the entire link. Header logging of received TLPs does not apply to correctable errors. All the RW1C bits can also be cleared from the local management bus by writing a 1 into the bit position.

Bits	SW	Name	Description	Reset
0	R/WOCLR	Receiver Error Status [RES]	This bit is set when an error is detected in the receive side of the Physical Layer of the core (e.g. a bit error or coding violation). This error is not Function-specific. STICKY.	0x0
5:1	R	Reserved [R12]	Reserved	0x0
6	R/WOCLR	Bad TP Status [BTS]	This bit is set when an error is detected in a received TLP by the Data Link Layer of the core. The conditions causing this error are: (i) An LCRC error (ii) The packet terminates with EDB symbol, but its LCRC field does not equal the inverted value of the calculated CRC. This error is not Function-specific. STICKY.	0x0
7	R/WOCLR	Bad DLLP Status [BDS]	This bit is set when an LCRC error is detected in a received DLLP, and no errors were detected by the Physical Layer. This error is not Function-specific. STICKY.	0x0
8	R/WOCLR	Replay Number Rollover Status [RNRS]	This bit is set when the replay count rolls over after three re-transmissions of a TLP at the Data Link Layer of the core. This error is not Function- specific STICKY.	0x0
11:9	R	Reserved [R13]	Reserved	0x0
12	R/WOCLR	Replay Timer Timeout Status [RTTS]	This bit is set when the replay timer in the Data Link Layer of the core times out, causing the core to retransmit a TLP. This error is not Function- specific. STICKY.	0x0
13	R/WOCLR	Advisory Non- Fatal Error Status [ANFES]	This bit is set when an uncorrectable error occurs, which is determined to belong to one of the special cases described in Section 6.2.3.2.4 of the PCI Express 2.0 Specifications. This causes the core to generate an ERR_COR message in place of an ERR_NONFATAL message. STICKY.	0x0
14	R/WOCLR	Corrected Internal Error Status [CIES]	This bit is set when the core has detected an internal correctable error condition (a correctable ECC error while reading from any of the RAMs). This bit is also set in response to the client signaling an internal error through the input CORRECTABLE_ERROR_IN. This error is not Function-specific. STICKY.	0x0

Bits	SW	Name	Description	Reset
15	R/WOCLR	Header Log Overflow Status [HLOS]	This bit is set on a Header Log Register overflow, that is, when the header could not be logged in the Header Log Register because it is occupied by a previous header. STICKY.	0x0
31:16	R	Reserved [R14]	Reserved	0x0

17.6.4.6.6 Correctable Error Mask Register

Propname：Correctable Error Mask Register
Address：@0x114
Description：The mask bits in this register control the reporting of correctable errors. For each error type in the Correctable Error Status Register, there is a corresponding bit in this register to mask its reporting. When a mask bit is set, the occurrence of the error is not reported to the Root Complex (by a PCI Express error message). The individual bits of the mask register are described below. The bits marked RW can also be written from the local management bus.

Bits	SW	Name	Description	Reset
0	R/W	Receiver Error Mask [REM]	This bit, when set, masks the generation of error messages in response to the Physical Layer errors STICKY.	0x0
5:1	R	Reserved [R15]	Reserved	0x0
6	R/W	Bad TLP Mask [BTM]	This bit, when set, masks the generation of error messages in response to a 'Bad TLP' received. STICKY.	0x0
7	R/W	Bad DLLP Mask [BDM]	This bit, when set, masks the generation of error messages in response to a 'Bad DLLP' received. STICKY.	0x0
8	R/W	Replay Number Rollover Mask [RNRM]	This bit, when set, masks the generation of error messages in response to a Replay Number Rollover event. STICKY.	0x0
11:9	R	Reserved [R16]	Reserved	0x0
12	R/W	Replay Timer Timeout Mask [RTTM]	This bit, when set, masks the generation of error messages in response to a Replay Timer timeout event. STICKY.	0x0
13	R/W	Advisory Non-Fatal Error Mask [ANFEM]	This bit, when set, masks the generation of error messages in response to an uncorrectable error occur, which is determined to belong to one of the special cases (as described in Section 6.2.3.2.4 of the PCI Express 2.0 Specifications). STICKY.	0x01

Bits	SW	Name	Description	Reset
14	R/W	Corrected Internal Error Mask [CIEM]	This bit, when set, masks the generation of error messages in response to a corrected internal error condition. STICKY.	0x01
15	R/W	Header Log Overflow Mask [HLOM]	This bit, when set, masks the generation of error messages in response to a Header Log register overflow. STICKY.	0x01
31:16	R	Reserved [R17]	Reserved	0x0

17.6.4.6.7 Advanced Error Capabilities and Control Register

Propname：Advanced Error Capabilities and Control Register
Address：@0x118
Description：This register contains a pointer to the first error that is reported in the Uncorrectable Error Status Register, and bits to enable ECRC generation and checking

Bits	SW	Name	Description	Reset
4:0	R	First Error Pointer [FER]	This is a 5-bit pointer to the bit position in the Uncorrectable Error Status Register corresponding to the error that was detected first. When there are multiple bits set in the Uncorrectable Error Status Register, this field informs the software which error was observed first. To prevent the field from being overwritten before software was able to read it, this field is not updated while the status bit pointed by it in the Uncorrectable Error Status Register remains set. After the software clears this status bit, a subsequent error condition that sets any bit in the Uncorrectable Error Status Register will update the First Error Pointer. Any uncorrectable error type, including the special cases where the error is reported using an ERR_COR message, will set the First Error Pointer (assuming the software has reset the error pointed by it in the Uncorrectable Error Status Register). STICKY.	0x0
5	R	ECRC Generation Capability [EGC]	This read-only bit indicates to the software that the device is capable of generating ECRC in packets transmitted on the link. This bit is writable from the local management bus.	0x1
6	R/W	Enable ECRC Generation [EEG]	Setting this bit enables the ECRC generation on the transmit side of the core. This bit is writable from the local management bus. STICKY.	0x0

Bits	SW	Name	Description	Reset
7	R	ECRC Check Capability [ECC]	This read-only bit indicates to the software that the device is capable of checking ECRC in packets received from the link. This bit is writable from the local management bus.	0x1
8	R/W	Enable ECRC Check [EEC]	Setting this bit enables ECRC checking on the receive side of the core. This bit is writable from the local management bus. STICKY.	0x0
9	R	Multiple Header Recording Capable [MHRC]	This bit is set when the Function has the capability to log more than one error header in its Header Log Registers. It is hardwired to 0.	0x0
10	R	Multiple Header Recording Enable [MHRE]	Setting this bit enables the Function to log multiple error headers in its Header Log Registers. It is hardwired to 0	0x0
31:11	R	Reserved [R18]	Reserved	0x0

17.6.4.6.8 Header Log Register 0

Propname：Header Log Register 0
Address：@0x11c
Description：This is the first of a set of four registers used to capture the header of a TLP received by the core from the link upon detection of an uncorrectable error. When multiple bits are set in the Uncorrectable Error Status Register, the captured header corresponds to the error that was detected first, that is, the error pointed by the First Error Pointer. To prevent the captured header from being over-written before software was able to read it, this register is not updated while the status bit pointed by the First Error Pointer in the Uncorrectable Error Status Register remains set. After the software clears this status bit, a subsequent error condition that sets any bit in the Uncorrectable Error Status Register will also cause the Header Log Registers to be updated. The double words of the TLP header are stored in the Header Log Registers with their bytes transposed. That is, the byte containing the Type/Format fields of the header is stored at bit positions 31:24 of the Header Log Register 0.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 0 [HD0]	First DWORD of captured TLP header STICKY.	0x0

17.6.4.6.9 Header Log Register 1

Propname：Header Log Register 1
Address：@0x120
Description：This register contains the second DWORD of the captured TLP header. The bytes are stored in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 1 [HD1]	Second DWORD of captured TLP header STICKY.	0x0

17.6.4.6.10 Header Log Register 2

Propname：Header Log Register 2
Address：@0x124
Description：This register contains the third DWORD of the captured TLP header. The bytes

are stored in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 2 [HD2]	Third DWORD of captured TLP header STICKY.	0x0

17.6.4.6.11 Header Log Register 3

Propname：Header Log Register 3
Address：@0x128
Description：If the captured TLP header is 4 DWORDs long, this register contains its fourth DWORD. If the captured header is a 3-DWORD header, this register is unused. The bytes of the DWORD are stored in this register in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 3 [HD3]	Fourth DWORD of captured TLP header STICKY.	0x0

17.6.4.6.12 Reserved

Propname：Reserved
Address：@0x12c + [0..2 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.7 i_ARI_cap_struct

17.6.4.7.1 ARI Extended Capability Header Register

Propname：ARI Extended Capability Header Register
Address：@0x140
Description：This register is used to enable the Alternate Routing ID interpretation. This register contains the PCI Express Extended Capability ID, the capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI-SIG to the ARI Extended Capability (000E hex).	0x0E
19:16	R	Capability Version [ARICV]	Specifies the SIG-assigned value for the version of the capability structure. This field is set to 1 by default, but can be modified independently for each Function from the local management bus	0x01
31:20	R	Next Capability Offset [ARINCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h160

17.6.4.7.2 ARI Capability Register and ARI Control Register

Propname：ARI Capability Register and ARI Control Register
Address：@0x144
Description：This location contains the ARI Capability Register and the ARI Control Register. The individual fields are described below.

Bits	SW	Name	Description	Reset
0	R	MFVC Function Groups Capability [MFGC]	Set when device supports arbitration at the Function Group-level. This field is hardwired to 0.	0x0
1	R	ACS Function Groups Capability [AFGC]	Relevant only when ACS Capability is supported. This field is hardwired to 0.	0x0
7:2	R	RSVD	RSVD	6'h00
15:8	R	Next Function [NF]	Points to the next Physical Function in the device. This field is set by default to point to the next Physical Function, 0 for last Function. It can be rewritten from the local management bus.	0x0
31:16	R	ARI Control Register [ACR]	ARI Control Register not implemented in this core. This field is hardwired to 0.	0x0

17.6.4.7.3 Reserved

Propname：Reserved
Address：@0x148 + [0..1 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.8 i_power_budgeting_cap_struct

17.6.4.8.1 Power Budgeting Enhanced Capability Header

Propname：Power Budgeting Enhanced Capability Header
Address：@0x160
Description：This register contains the PCI Express Extended Capability ID for Power Budgeting Capability, its capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the PCI Express Power Budgeting Capability (0004 hex).	0x04
19:16	R	Capability Version [PCV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified from the local management bus by writing into Function 0 from the local management bus.	0x01
31:20	R	Next Capability Offset [PBNCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h1b8

17.6.4.8.2 Power Budgeting Data Select Register

Propname：Power Budgeting Data Select Register

Address：@0x164
Description：This register is used to select the specific word of specific power-budgeting data returned on a read from the Power Budgeting Data Register. This version of the core stores power budgeting data for three distinct power states(D0, D1 and D3hot) for each Physical Function, which can be read from the Power Budgeting Data Register by indexing through this register, as described below.

Bits	SW	Name	Description	Reset
7:0	R/W	Power Budgeting Data Index [PBDN]	This field selects the power budgeting data read from the Power Budgeting Data Register. Its settings are: 00: Selects power budgeting data for power state D0 MAX for the associated PF. 01: Selects power budgeting data for power state D0 SUSTAINED for the associated PF. 10: Selects power budgeting data for power state D3hot for the associated PF. 11: Selects power budgeting data for power state D1 for the associated PF. Others: Not a valid setting. A read from the Power Budgeting Data Register returns all zeroes.	0x0
31:8	R	Reserved [R0]	(no description)	0x0

17.6.4.8.3 Power Budgeting Data Register
Propname：Power Budgeting Data Register
Address：@0x168
Description：This read-only register returns the DWORD of Power Budgeting Data selected by the Data Select register. Each DWORD of the Power Budgeting Data describes the power usage of the device in a particular operating condition. All the fields can be modified independently for each PF by writing from the local management bus.

Bits	SW	Name	Description	Reset
7:0	R	Base Power [BP]	Specifies base power(in watts) of the selected power state	8'd240
9:8	R	Data Scale [DS]	Scale factor applicable to the Base Power field.	0x0
12:10	R	PM Sub-State [PSS]	Specifies the power management sub-state of the selected power state	0x0
14:13	R	PM State [PS]	Specifies the power management state of the Function, for which this power management data applies.	0x0
17:15	R	Type [TYPE]	Specifies the operation condition for which the data applies.	0x7
20:18	R	Power Rail [PR]	Specifies the power rail corresponding to the power management data in this register.	0x2
31:21	R	Reserved [R1]	Reserved	0x0

17.6.4.8.4 Power Budget Capability Register
Propname：Power Budget Capability Register
Address：@0x16c
Description：This register specifies whether the device power specified by this Capability Structure is included in the system power budget.

Bits	SW	Name	Description	Reset
0	R	System Allocated [SA]	This bit is set to indicate that the device power specified by this Power Management Capability Structure is included in the system power budget. When this bit set, the software must exclude the device power reported by this Capability Structure from power calculations, when making power budgeting decisions. This bit is set to 0 by default, but its setting can be modified individually for each PF from the local management bus.	0x0
31:1	R	Reserved [R4]	Reserved	0x0

17.6.4.8.5 Reserved

Propname：Reserved
Address：@0x170 + [0..3 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.9 i_resizable_BAR_cap_struct

17.6.4.9.1 Resizable BAR Extended Capability Header Register

Propname：Resizable BAR Extended Capability Header Register
Address：@0x180
Description：This register contains the PCI Express Extended Capability ID for the Resizable BAR Capability, its capability version, and the pointer to the next capability structure. This register is enabled only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register (Section 8.4.2.24). When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the Resizable BAR Capability (0015 hex).	0x0
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified from the local management bus.	0x0
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h0

17.6.4.9.2 Resizable BAR Capability Register 0

Propname：Resizable BAR Capability Register 0
Address：@0x184
Description：This register advertises the available aperture settings of the first memory

BAR of the associated Physical Function. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability (bit 31) of the associated Physical Function BAR Configuration Register 1. When the Resizable BAR Capability is not enabled, a read from this location returns all zeros. When the Resizable BAR Capability is enabled, any of the bits 4-22 can be modified from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R	Aperture 1M [A1M]	Indicates that the BAR aperture can be set to 1M.	0x0
5	R	Aperture 2M [A2M]	Indicates that the BAR aperture can be set to 2M.	0x0
6	R	Aperture 4M [A4M]	Indicates that the BAR aperture can be set to 4M.	0x0
7	R	Aperture 8M [A8M]	Indicates that the BAR aperture can be set to 8M.	0x0
8	R	Aperture 16M [A16M]	Indicates that the BAR aperture can be set to 16M.	0x0
9	R	Aperture 32M [A32M]	Indicates that the BAR aperture can be set to 32M.	0x0
10	R	Aperture 64M [A64M]	Indicates that the BAR aperture can be set to 64M.	0x0
11	R	Aperture 128M [A128M]	Indicates that the BAR aperture can be set to 128M.	0x0
12	R	Aperture 256M [A256M]	Indicates that the BAR aperture can be set to 256M.	0x0
13	R	Aperture 512M [A512M]	Indicates that the BAR aperture can be set to 512M.	0x0
14	R	Aperture 1G [A1G]	Indicates that the BAR aperture can be set to 1G.	0x0
15	R	Aperture 2G [A2G]	Indicates that the BAR aperture can be set to 2G.	0x0
16	R	Aperture 4G [A4G]	Indicates that the BAR aperture can be set to 4G.	0x0
17	R	Aperture 8G [A8G]	Indicates that the BAR aperture can be set to 8G.	0x0
18	R	Aperture 16G [A16G]	Indicates that the BAR aperture can be set to 16G.	0x0
19	R	Aperture 32G [A32G]	Indicates that the BAR aperture can be set to 32G.	0x0
20	R	Aperture 64G [A64G]	Indicates that the BAR aperture can be set to 64G.	0x0
21	R	Aperture 128G [A128G]	Indicates that the BAR aperture can be set to 128G.	0x0

Bits	SW	Name	Description	Reset
22	R	Aperture 256G [A256G]	Indicates that the BAR aperture can be set to 256G.	0x0
23	R	Aperture 512G [A512G]	Indicates that the BAR aperture can be set to 512G.	0x0
31:24	R	Reserved [R1]	Reserved	0x0

17.6.4.9.3 Resizable BAR Control Register 0

Propname：Resizable BAR Control Register 0
Address：@0x188
Description：This register controls the aperture setting of the first memory BAR of the associated Physical Function, and also has a field that specifies the number of resizable BARs configurable through the Resizable BAR Capability Structure. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes. When the Resizable BAR Capability is enabled, all valid fields of this register can be modified from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Index [BARI]	Specifies the index of the BAR controlled by this register. This field can be modified independently for each PF from the local management bus.	0x0
4:3	R	Reserved [R2]	Reserved	0x0
7:5	R	Resizable BAR Count [RBARC]	Specifies the number of BARs that can be configured through the Resizable BAR Capability Structure for this PF. This field can be modified independently for each PF from the local management bus.	0x0
12:8	R	BAR Size [BARS]	When the Resizable BAR Capability is enabled for the Physical Function, this field controls the BAR aperture for the first BAR of the PF (0 = 1M, 1 = 2M, ... , 12 = 4G). This field can be modified independently for each PF from the local management bus.	0x0
31:13	R	Reserved [R3]	Reserved	0x0

17.6.4.9.4 Resizable BAR Capability Register 1

Propname：Resizable BAR Capability Register 1
Address：@0x18c
Description：This register advertises the available aperture settings of the first memory BAR of the associated Physical Function. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability (bit 31) of the associated Physical Function BAR Configuration Register 1. When the Resizable BAR Capability is not enabled, a read from this location returns all zeros. When the Resizable BAR Capability is enabled, any of the bits 4-22 can be modified from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0

Bits	SW	Name	Description	Reset
4	R	Aperture 1M [A1M]	Indicates that the BAR aperture can be set to 1M.	0x0
5	R	Aperture 2M [A2M]	Indicates that the BAR aperture can be set to 2M.	0x0
6	R	Aperture 4M [A4M]	Indicates that the BAR aperture can be set to 4M.	0x0
7	R	Aperture 8M [A8M]	Indicates that the BAR aperture can be set to 8M.	0x0
8	R	Aperture 16M [A16M]	Indicates that the BAR aperture can be set to 16M.	0x0
9	R	Aperture 32M [A32M]	Indicates that the BAR aperture can be set to 32M.	0x0
10	R	Aperture 64M [A64M]	Indicates that the BAR aperture can be set to 64M.	0x0
11	R	Aperture 128M [A128M]	Indicates that the BAR aperture can be set to 128M.	0x0
12	R	Aperture 256M [A256M]	Indicates that the BAR aperture can be set to 256M.	0x0
13	R	Aperture 512M [A512M]	Indicates that the BAR aperture can be set to 512M.	0x0
14	R	Aperture 1G [A1G]	Indicates that the BAR aperture can be set to 1G.	0x0
15	R	Aperture 2G [A2G]	Indicates that the BAR aperture can be set to 2G.	0x0
16	R	Aperture 4G [A4G]	Indicates that the BAR aperture can be set to 4G.	0x0
17	R	Aperture 8G [A8G]	Indicates that the BAR aperture can be set to 8G.	0x0
18	R	Aperture 16G [A16G]	Indicates that the BAR aperture can be set to 16G.	0x0
19	R	Aperture 32G [A32G]	Indicates that the BAR aperture can be set to 32G.	0x0
20	R	Aperture 64G [A64G]	Indicates that the BAR aperture can be set to 64G.	0x0
21	R	Aperture 128G [A128G]	Indicates that the BAR aperture can be set to 128G.	0x0
22	R	Aperture 256G [A256G]	Indicates that the BAR aperture can be set to 256G.	0x0
23	R	Aperture 512G [A512G]	Indicates that the BAR aperture can be set to 512G.	0x0
31:24	R	Reserved [R1]	Reserved	0x0

17.6.4.9.5 Resizable BAR Control Register 1

Propname：Resizable BAR Control Register 1
Address：@0x190
Description：This register controls the aperture setting of the first memory BAR of the associated Physical Function, and also has a field that specifies the number of resizable BARs configurable through the Resizable BAR Capability Structure. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes. When the Resizable BAR Capability is enabled, all valid fields of this register can be modified from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Index [BARI]	Specifies the index of the BAR controlled by this register. This field can be modified independently for each PF from the local management bus.	0x0
4:3	R	Reserved [R2]	Reserved	0x0
7:5	R	Resizable BAR Count [RBARC]	Specifies the number of BARs that can be configured through the Resizable BAR Capability Structure for this PF. This field can be modified independently for each PF from the local management bus.	0x0
12:8	R	BAR Size [BARS]	When the Resizable BAR Capability is enabled for the Physical Function, this field controls the BAR aperture for the first BAR of the PF (0 = 1M, 1 = 2M, ... , 12 = 4G). This field can be modified independently for each PF from the local management bus.	0x0
31:13	R	Reserved [R3]	Reserved	0x0

17.6.4.9.6 Resizable BAR Capability Register 2

Propname：Resizable BAR Capability Register 2
Address：@0x194
Description：This register advertises the available aperture settings of the first memory BAR of the associated Physical Function. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability (bit 31) of the associated Physical Function BAR Configuration Register 1. When the Resizable BAR Capability is not enabled, a read from this location returns all zeros. When the Resizable BAR Capability is enabled, any of the bits 4-22 can be modified from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R	Aperture 1M [A1M]	Indicates that the BAR aperture can be set to 1M.	0x0
5	R	Aperture 2M [A2M]	Indicates that the BAR aperture can be set to 2M.	0x0
6	R	Aperture 4M [A4M]	Indicates that the BAR aperture can be set to 4M.	0x0
7	R	Aperture 8M [A8M]	Indicates that the BAR aperture can be set to 8M.	0x0

Bits	SW	Name	Description	Reset
8	R	Aperture 16M [A16M]	Indicates that the BAR aperture can be set to 16M.	0x0
9	R	Aperture 32M [A32M]	Indicates that the BAR aperture can be set to 32M.	0x0
10	R	Aperture 64M [A64M]	Indicates that the BAR aperture can be set to 64M.	0x0
11	R	Aperture 128M [A128M]	Indicates that the BAR aperture can be set to 128M.	0x0
12	R	Aperture 256M [A256M]	Indicates that the BAR aperture can be set to 256M.	0x0
13	R	Aperture 512M [A512M]	Indicates that the BAR aperture can be set to 512M.	0x0
14	R	Aperture 1G [A1G]	Indicates that the BAR aperture can be set to 1G.	0x0
15	R	Aperture 2G [A2G]	Indicates that the BAR aperture can be set to 2G.	0x0
16	R	Aperture 4G [A4G]	Indicates that the BAR aperture can be set to 4G.	0x0
17	R	Aperture 8G [A8G]	Indicates that the BAR aperture can be set to 8G.	0x0
18	R	Aperture 16G [A16G]	Indicates that the BAR aperture can be set to 16G.	0x0
19	R	Aperture 32G [A32G]	Indicates that the BAR aperture can be set to 32G.	0x0
20	R	Aperture 64G [A64G]	Indicates that the BAR aperture can be set to 64G.	0x0
21	R	Aperture 128G [A128G]	Indicates that the BAR aperture can be set to 128G.	0x0
22	R	Aperture 256G [A256G]	Indicates that the BAR aperture can be set to 256G.	0x0
23	R	Aperture 512G [A512G]	Indicates that the BAR aperture can be set to 512G.	0x0
31:24	R	Reserved [R1]	Reserved	0x0

17.6.4.9.7 Resizable BAR Control Register 2

Propname：Resizable BAR Control Register 2
Address：@0x198
Description：This register controls the aperture setting of the first memory BAR of the associated Physical Function, and also has a field that specifies the number of resizable BARs configurable through the Resizable BAR Capability Structure. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR

Configuration Register. When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes. When the Resizable BAR Capability is enabled, all valid fields of this register can be modified from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Index [BARI]	Specifies the index of the BAR controlled by this register. This field can be modified independently for each PF from the local management bus.	0x0
4:3	R	Reserved [R2]	Reserved	0x0
7:5	R	Resizable BAR Count [RBARC]	Specifies the number of BARs that can be configured through the Resizable BAR Capability Structure for this PF. This field can be modified independently for each PF from the local management bus.	0x0
12:8	R	BAR Size [BARS]	When the Resizable BAR Capability is enabled for the Physical Function, this field controls the BAR aperture for the first BAR of the PF (0 = 1M, 1 = 2M, ... , 12 = 4G). This field can be modified independently for each PF from the local management bus.	0x0
31:13	R	Reserved [R3]	Reserved	0x0

17.6.4.9.8 Resizable BAR Capability Register 3

Propname：Resizable BAR Capability Register 3
Address：@0x19c
Description：This register advertises the available aperture settings of the first memory BAR of the associated Physical Function. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability (bit 31) of the associated Physical Function BAR Configuration Register 1. When the Resizable BAR Capability is not enabled, a read from this location returns all zeros. When the Resizable BAR Capability is enabled, any of the bits 4-22 can be modified from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R	Aperture 1M [A1M]	Indicates that the BAR aperture can be set to 1M.	0x0
5	R	Aperture 2M [A2M]	Indicates that the BAR aperture can be set to 2M.	0x0
6	R	Aperture 4M [A4M]	Indicates that the BAR aperture can be set to 4M.	0x0
7	R	Aperture 8M [A8M]	Indicates that the BAR aperture can be set to 8M.	0x0
8	R	Aperture 16M [A16M]	Indicates that the BAR aperture can be set to 16M.	0x0
9	R	Aperture 32M [A32M]	Indicates that the BAR aperture can be set to 32M.	0x0
10	R	Aperture 64M [A64M]	Indicates that the BAR aperture can be set to 64M.	0x0

Bits	SW	Name	Description	Reset
11	R	Aperture 128M [A128M]	Indicates that the BAR aperture can be set to 128M.	0x0
12	R	Aperture 256M [A256M]	Indicates that the BAR aperture can be set to 256M.	0x0
13	R	Aperture 512M [A512M]	Indicates that the BAR aperture can be set to 512M.	0x0
14	R	Aperture 1G [A1G]	Indicates that the BAR aperture can be set to 1G.	0x0
15	R	Aperture 2G [A2G]	Indicates that the BAR aperture can be set to 2G.	0x0
16	R	Aperture 4G [A4G]	Indicates that the BAR aperture can be set to 4G.	0x0
17	R	Aperture 8G [A8G]	Indicates that the BAR aperture can be set to 8G.	0x0
18	R	Aperture 16G [A16G]	Indicates that the BAR aperture can be set to 16G.	0x0
19	R	Aperture 32G [A32G]	Indicates that the BAR aperture can be set to 32G.	0x0
20	R	Aperture 64G [A64G]	Indicates that the BAR aperture can be set to 64G.	0x0
21	R	Aperture 128G [A128G]	Indicates that the BAR aperture can be set to 128G.	0x0
22	R	Aperture 256G [A256G]	Indicates that the BAR aperture can be set to 256G.	0x0
23	R	Aperture 512G [A512G]	Indicates that the BAR aperture can be set to 512G.	0x0
31:24	R	Reserved [R1]	Reserved	0x0

17.6.4.9.9 Resizable BAR Control Register 3

Propname：Resizable BAR Control Register 3
Address：@0x1a0
Description：This register controls the aperture setting of the first memory BAR of the associated Physical Function, and also has a field that specifies the number of resizable BARs configurable through the Resizable BAR Capability Structure. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes. When the Resizable BAR Capability is enabled, all valid fields of this register can be modified from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Index [BARI]	Specifies the index of the BAR controlled by this register. This field can be modified independently for each PF from the local management bus.	0x0

Bits	SW	Name	Description	Reset
4:3	R	Reserved [R2]	Reserved	0x0
7:5	R	Resizable BAR Count [RBARC]	Specifies the number of BARs that can be configured through the Resizable BAR Capability Structure for this PF. This field can be modified independently for each PF from the local management bus.	0x0
12:8	R	BAR Size [BARS]	When the Resizable BAR Capability is enabled for the Physical Function, this field controls the BAR aperture for the first BAR of the PF (0 = 1M, 1 = 2M, ... , 12 = 4G). This field can be modified independently for each PF from the local management bus.	0x0
31:13	R	Reserved [R3]	Reserved	0x0

17.6.4.9.10 Resizable BAR Capability Register 4

Propname：Resizable BAR Capability Register 4

Address：@0x1a4

Description：This register advertises the available aperture settings of the first memory BAR of the associated Physical Function. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability (bit 31) of the associated Physical Function BAR Configuration Register 1. When the Resizable BAR Capability is not enabled, a read from this location returns all zeros. When the Resizable BAR Capability is enabled, any of the bits 4-22 can be modified from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R	Aperture 1M [A1M]	Indicates that the BAR aperture can be set to 1M.	0x0
5	R	Aperture 2M [A2M]	Indicates that the BAR aperture can be set to 2M.	0x0
6	R	Aperture 4M [A4M]	Indicates that the BAR aperture can be set to 4M.	0x0
7	R	Aperture 8M [A8M]	Indicates that the BAR aperture can be set to 8M.	0x0
8	R	Aperture 16M [A16M]	Indicates that the BAR aperture can be set to 16M.	0x0
9	R	Aperture 32M [A32M]	Indicates that the BAR aperture can be set to 32M.	0x0
10	R	Aperture 64M [A64M]	Indicates that the BAR aperture can be set to 64M.	0x0
11	R	Aperture 128M [A128M]	Indicates that the BAR aperture can be set to 128M.	0x0
12	R	Aperture 256M [A256M]	Indicates that the BAR aperture can be set to 256M.	0x0
13	R	Aperture 512M [A512M]	Indicates that the BAR aperture can be set to 512M.	0x0

Bits	SW	Name	Description	Reset
14	R	Aperture 1G [A1G]	Indicates that the BAR aperture can be set to 1G.	0x0
15	R	Aperture 2G [A2G]	Indicates that the BAR aperture can be set to 2G.	0x0
16	R	Aperture 4G [A4G]	Indicates that the BAR aperture can be set to 4G.	0x0
17	R	Aperture 8G [A8G]	Indicates that the BAR aperture can be set to 8G.	0x0
18	R	Aperture 16G [A16G]	Indicates that the BAR aperture can be set to 16G.	0x0
19	R	Aperture 32G [A32G]	Indicates that the BAR aperture can be set to 32G.	0x0
20	R	Aperture 64G [A64G]	Indicates that the BAR aperture can be set to 64G.	0x0
21	R	Aperture 128G [A128G]	Indicates that the BAR aperture can be set to 128G.	0x0
22	R	Aperture 256G [A256G]	Indicates that the BAR aperture can be set to 256G.	0x0
23	R	Aperture 512G [A512G]	Indicates that the BAR aperture can be set to 512G.	0x0
31:24	R	Reserved [R1]	Reserved	0x0

17.6.4.9.11 Resizable BAR Control Register 4

Propname：Resizable BAR Control Register 4

Address：@0x1a8

Description：This register controls the aperture setting of the first memory BAR of the associated Physical Function, and also has a field that specifies the number of resizable BARs configurable through the Resizable BAR Capability Structure. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes. When the Resizable BAR Capability is enabled, all valid fields of this register can be modified from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Index [BARI]	Specifies the index of the BAR controlled by this register. This field can be modified independently for each PF from the local management bus.	0x0
4:3	R	Reserved [R2]	Reserved	0x0
7:5	R	Resizable BAR Count [RBARC]	Specifies the number of BARs that can be configured through the Resizable BAR Capability Structure for this PF. This field can be modified independently for each PF from the local management bus.	0x0

Bits	SW	Name	Description	Reset
12:8	R	BAR Size [BARS]	When the Resizable BAR Capability is enabled for the Physical Function, this field controls the BAR aperture for the first BAR of the PF (0 = 1M, 1 = 2M, ... , 12 = 4G). This field can be modified independently for each PF from the local management bus.	0x0
31:13	R	Reserved [R3]	Reserved	0x0

17.6.4.9.12 Resizable BAR Capability Register 5

Propname：Resizable BAR Capability Register 5

Address：@0x1ac

Description：This register advertises the available aperture settings of the first memory BAR of the associated Physical Function. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability (bit 31) of the associated Physical Function BAR Configuration Register 1. When the Resizable BAR Capability is not enabled, a read from this location returns all zeros. When the Resizable BAR Capability is enabled, any of the bits 4-22 can be modified from the local management bus.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R	Aperture 1M [A1M]	Indicates that the BAR aperture can be set to 1M.	0x0
5	R	Aperture 2M [A2M]	Indicates that the BAR aperture can be set to 2M.	0x0
6	R	Aperture 4M [A4M]	Indicates that the BAR aperture can be set to 4M.	0x0
7	R	Aperture 8M [A8M]	Indicates that the BAR aperture can be set to 8M.	0x0
8	R	Aperture 16M [A16M]	Indicates that the BAR aperture can be set to 16M.	0x0
9	R	Aperture 32M [A32M]	Indicates that the BAR aperture can be set to 32M.	0x0
10	R	Aperture 64M [A64M]	Indicates that the BAR aperture can be set to 64M.	0x0
11	R	Aperture 128M [A128M]	Indicates that the BAR aperture can be set to 128M.	0x0
12	R	Aperture 256M [A256M]	Indicates that the BAR aperture can be set to 256M.	0x0
13	R	Aperture 512M [A512M]	Indicates that the BAR aperture can be set to 512M.	0x0
14	R	Aperture 1G [A1G]	Indicates that the BAR aperture can be set to 1G.	0x0
15	R	Aperture 2G [A2G]	Indicates that the BAR aperture can be set to 2G.	0x0
16	R	Aperture 4G [A4G]	Indicates that the BAR aperture can be set to 4G.	0x0

Bits	SW	Name	Description	Reset
17	R	Aperture 8G [A8G]	Indicates that the BAR aperture can be set to 8G.	0x0
18	R	Aperture 16G [A16G]	Indicates that the BAR aperture can be set to 16G.	0x0
19	R	Aperture 32G [A32G]	Indicates that the BAR aperture can be set to 32G.	0x0
20	R	Aperture 64G [A64G]	Indicates that the BAR aperture can be set to 64G.	0x0
21	R	Aperture 128G [A128G]	Indicates that the BAR aperture can be set to 128G.	0x0
22	R	Aperture 256G [A256G]	Indicates that the BAR aperture can be set to 256G.	0x0
23	R	Aperture 512G [A512G]	Indicates that the BAR aperture can be set to 512G.	0x0
31:24	R	Reserved [R1]	Reserved	0x0

17.6.4.9.13 Resizable BAR Control Register 5

Propname：Resizable BAR Control Register 5
Address：@0x1b0
Description：This register controls the aperture setting of the first memory BAR of the associated Physical Function, and also has a field that specifies the number of resizable BARs configurable through the Resizable BAR Capability Structure. This register is active only when the Resizable BAR Capability is enabled for the Physical Function by setting the Enable Resizable BAR Capability bit (bit 31) of the associated Physical Function BAR Configuration Register. When the Resizable BAR Capability is not enabled, a read from this location returns all zeroes. When the Resizable BAR Capability is enabled, all valid fields of this register can be modified from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Index [BARI]	Specifies the index of the BAR controlled by this register. This field can be modified independently for each PF from the local management bus.	0x0
4:3	R	Reserved [R2]	Reserved	0x0
7:5	R	Resizable BAR Count [RBARC]	Specifies the number of BARs that can be configured through the Resizable BAR Capability Structure for this PF. This field can be modified independently for each PF from the local management bus.	0x0
12:8	R	BAR Size [BARS]	When the Resizable BAR Capability is enabled for the Physical Function, this field controls the BAR aperture for the first BAR of the PF (0 = 1M, 1 = 2M, ... , 12 = 4G). This field can be modified independently for each PF from the local management bus.	0x0
31:13	R	Reserved [R3]	Reserved	0x0

17.6.4.10 i_LTR_cap_struct

17.6.4.10.1 Latency Tolerance Reporting (LTR) Extended Capability Header Register

Propname：Latency Tolerance Reporting (LTR) Extended Capability Header Register
Address：@0x1b8
Description：This register contains the PCI Express Extended Capability ID for the Latency Tolerance Reporting (LTR) Capability, its capability version, and the pointer to the next capability structure. This register is implemented only for Physical Function 0. A read from this address of other Physical Functions configuration space returns all zeroes.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the Latency Tolerance Reporting Capability (0018 hex).	0x018
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified from the local management bus.	0x01
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h1c0

17.6.4.10.2 LTR Max Snoop/Max No-Snoop Latency Register

Propname：LTR Max Snoop/Max No-Snoop Latency Register
Address：@0x1bc
Description：This register contains the maximum snoop latency and the maximum no-snoop latency that the device is allowed to request in an LTR message it originates.

Bits	SW	Name	Description	Reset
9:0	R/W	Max Snoop Latency [MSL]	When multiplied by the value of the Max Snoop Latency Scale, this field defines the maximum snoop value the device is permitted to request in an LTR message. This field can be written independently for each Physical Function from the local management bus.	0x0
12:10	R/W	Max Snoop Latency Scale [MSLS]	Specifies the scale value for the Max Snoop Latency. When the setting of this field is non- zero, the actual snoop latency is determined by multiplying the Max Snoop Latency by the following scale factors: 001: 32 ns, 010: 1024 ns, 011: 32,768 ns, 100: 1,047,576 ns, 101: 33,554,432ns, 110-111: Reserved	0x0
15:13	R	Reserved [R0]	Reserved	0x0

Bits	SW	Name	Description	Reset
25:16	R/W	Max No-Snoop Latency [MNSL]	When multiplied by the value of the Max No- Snoop Latency Scale, this field defines the maximum no-snoop value the device is permitted to request in an LTR message. This field can be written independently for each Physical Function from the local management bus.	0x0
28:26	R/W	Max No-Snoop Latency Scale [MNSLS]	Specifies the scale value for the Max No-Snoop Latency. When the setting of this field is non- zero, the actual snoop latency is determined by multiplying the Max No-Snoop Latency by the following scale factors: 001: 32 ns, 010: 1024 ns, 011: 32,768 ns, 100: 1,047,576 ns, 101: 33,554,432ns, 110-111: Reserved	0x0
31:29	R	Reserved [R1]	Reserved	0x0

17.6.4.11 i_DPA_cap_struct

17.6.4.11.1 DPA Extended Capability Header Register

Propname：DPA Extended Capability Header Register
Address：@0x1c0
Description：This location contains the PCI Express Extended Capability ID for DPA Capability and the offset to the next capability block.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the Dynamic Power Allocation Reporting Capability	0x0016
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified from the local management bus.	0x1
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h200

17.6.4.11.2 DPA Capability Register

Propname：DPA Capability Register
Address：@0x1c4
Description：This register contains the DPA capability parameters for the associated Function.

Bits	SW	Name	Description	Reset
4:0	R	Maximum Number of Substates [MNS]	Maximum number of DPA substates supported by the Function (the value in this field is the number of substates minus 1).	5'd7
7:5	R	Reserved [R0]	Reserved	0x0

Bits	SW	Name	Description	Reset
9:8	R	Transition Latency Unit [TLU]	This is the unit of the transition latencies specified in the Transition Latency Value 0 and Transition Latency Value 1 fields of this register (00 = 1ms, 01 = 10ms, 10 = 100ms, 11 = reserved).	0x0
11:10	R	Reserved [R1]	Reserved	0x0
13:12	R	Power Allocation Scale [PAS]	This is the scale used to compute the actual power from the values specified in the Dynamic Power Allocation Array Registers 0 - 7. The actual power in Watts is obtained by multiplying the value in the Dynamic Power Allocation Array Register by this scale factor (00 = 10x, 01 = 1x, 10 = 0.1x, 11 = 0.01x).	0x0
15:14	R	Reserved [R2]	Reserved	0x0
23:16	R	Transition Latency Value 0 [TLV0]	Specifies the transition latency for the substate. Each of the 32 substates may specify one of the two transition latency values. This field contains the first of the two latency values. The unit of latency is specified by the Transition Latency Unit field of this register.	8'h10
31:24	R	Transition Latency Value 1 [TLV1]	Specifies the second of the two transition latency values for the substates. The unit of latency is specified by the Transition Latency Unit field of this register.	

17.6.4.11.3 DPA Latency Indicator Register

Propname： DPA Latency Indicator Register
Address：@0x1c8
Description：This location contains Transition Latency Indicator bits for the DPA substates.

Bits	SW	Name	Description	Reset
31:0	R	Transition Latency Indicator Bits [TLIN]	Bit i of this register indicates the choice of the transition latency value for substate i. A setting of 0 indicates that Transition Latency Value 0 from the DPA Capability Register applies to this substate; a setting of 1 indicates that Transition Latency Value 1 applies.	See Description

17.6.4.11.4 DPA Control and Status Registers

Propname：DPA Control and Status Registers
Address：@0x1cc
Description：This location contains the DPA Control Register and the DPA Status Register.

Bits	SW	Name	Description	Reset
4:0	R	Substate Status [SS]	This field provides the current DPA substate of this Function. This field is writable from the local management bus, and must be updated by the local software running on the EndPoint upon completion of a DPA transition to a new substate.	0x0
7:5	R	Reserved [R3]	Reserved	0x0
8	R/WOCLR	Substate Control Enabled [SCE]	This bit enables the Substate Control field. This bit is initialized to 1 by the hardware on a power- on reset or a Function-Level Reset. Software may clear this bit by writing a 1 to this bit position, but cannot set this bit directly through a configuration write. Clearing this bit disables the Substate Control field, thus preventing further substate transitions for this Function. This bit can be set to 0 or 1 through the local management bus, by writing a 0 or 1, respectively.	0x1
15:9	R	Reserved [R4]	Reserved	0x0
20:16	R/W	Substate Control [SC]	This field is used to initiate a transition of the Function's DPA to a new substate. To initiate the transition, software must write the desired substate value into this field and wait for the transition latency of the substate for the Function to complete the transition. This field can also be written from the local management bus. All substate transitions are disabled when the Substate Control Enabled bit is 0. The core generates a one-cycle pulse on the output DPA_INTERRUPT when the value if this field is changed (bit 0 is for PF 0 and so on) This interrupt informs the client of the request from software to change the DPA substate. In response, the client must read the Substate Control field from this register to determine the new substate, and perform the actions necessary to effect the substate change. On completion of the substate change, the client must update the Substate Status field to reflect the new substate the function is in.	0x0
31:21	R	Reserved [R5]	Reserved	0x0

17.6.4.11.5 Dynamic Power Allocation Array Register 0

Propname：Dynamic Power Allocation Array Register 0
Address：@0x1d0
Description：This is a register in an array of 2registers that contain the power allocations for the DPA substates. Each location contains power allocation values for four substates, 8 bits per substate. The value in each 8-bit field, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.

Bits	SW	Name	Description	Reset
7:0	R	Substate Power Allocation 0 [SPA0_0]	This field contains the power allocation for the DPA substate #0 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h0
15:8	R	Substate Power Allocation 1 [SPA1_0]	This field contains the power allocation for the DPA substate #1 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h1
23:16	R	Substate Power Allocation 2 [SPA2_0]	This field contains the power allocation for the DPA substate #2 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h2
31:24	R	Substate Power Allocation 3 [SPA3_0]	This field contains the power allocation for the DPA substate #3 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h3

17.6.4.11.6 Dynamic Power Allocation Array Register 1

Propname：Dynamic Power Allocation Array Register 1
Address：@0x1d4
Description This is a register in an array of 2registers that contain the power allocations for the DPA substates. Each location contains power allocation values for four substates, 8 bits per substate. The value in each 8-bit field, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.

Bits	SW	Name	Description	Reset
7:0	R	Substate Power Allocation 4 [SPA0_1]	This field contains the power allocation for the DPA substate #4 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h4
15:8	R	Substate Power Allocation 5 [SPA1_1]	This field contains the power allocation for the DPA substate #5 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h5
23:16	R	Substate Power Allocation 6 [SPA2_1]	This field contains the power allocation for the DPA substate #6 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h6
31:24	R	Substate Power Allocation 7 [SPA3_1]	This field contains the power allocation for the DPA substate #7 covered by this register. This value, when multiplied by the Power Allocation Scale programmed in the DPA Capability Register, provides the power associated with the corresponding substate in Watts.	8'h7

17.6.4.11.7 Reserved

Propname：Reserved
Address：@0x1d8 + [0..3 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.12 i_SRIOV_cap_struct

17.6.4.12.1 SR-IOV Extended Capability Header Register

Propname：SR-IOV Extended Capability Header Register
Address：@0x200
Description：This location contains the PCI Express Extended Capability ID for SR-IOV and the offset to the next capability block.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI-SIG to the SR-IOV Extended Capability Structure (0010 hex).	0x0010

Bits	SW	Name	Description	Reset
19:16	R	Capability Version [CV]	Specifies the SIG-assigned value for the version of the capability structure. This field is set by default to 1, but can be modified independently for each Function from the local management bus.	0x1
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h274

17.6.4.12.2 SR-IOV Capabilities Register

Propname：SR-IOV Capabilities Register
Address：@0x204
Description：This register defines various capabilities of the SR-IOV implementation.

Bits	SW	Name	Description	Reset
0	R	VF Migration Capable [VFMC]	Set when the core supports VF migration. Hardwired to 0.	0x0
1	R	ARI Capable Hierarchy Preserved [ACHP]	A 1 in this bit position indicates that the ARI Capable Hierarchy bit in the SR-IOV Control Register is preserved across certain power state transitions (see the PCI-SIG Single Root IO Virtualization and Sharing Specifications, Version 1.1, Section 3.3.3.5 for details). This bit is set to 1 by default, but can be modified from the local management bus.	0x1
31:2	R	Reserved [R0]	Reserved	0x0

17.6.4.12.3 SR-IOV Control and Status Registers

Propname：SR-IOV Control and Status Registers
Address：@0x208
Description：This location contains the SR-IOV Control Register and the SR-IOV Status Register.

Bits	SW	Name	Description	Reset
0	R/W	VF Enable [VFE]	This bit must be set to enable the VFs associated with this PF.	0x0
1	R	VF Migration Enable [VFME]	Not supported. Hardwired to 0	0x0
2	R	VF Migration Interrupt Enable [VFMIE]	Not supported. Hardwired to 0	0x0

Bits	SW	Name	Description	Reset
3	R/W	VF Memory Space Enable [VFMSE]	This bit must be set to allow access to the memory space of the VFs associated with this PF.	0x0
4	R/W	ARI Capable Hierarchy [ARIE]	This bit enables the ARI mode for Virtual Functions. This bit must be set when VF Enable is set. Valid only for PF0	0x0
15:5	R	Reserved [R1]	Reserved	0x0
31:16	R	SRIOV Status Register [SSR]	Not implemented.	0x0

17.6.4.12.4 Initial VFs/Total VFs Register

Propname：Initial VFs/Total VFs Register
Address：@0x20c
Description：This location contains registers that specify the initial and the total number Virtual Functions (VFs) in the device.

Bits	SW	Name	Description	Reset
15:0	R	Initial VFs [IVF]	This field contains the initial number of VFs configured for each PF. This field can be modified using local management registers.	0x8
31:16	R	Total VFs [TVF]	This field contains the total number of VFs per PF. Its default setting is identical to that of InitialVFs. This field can be modified using local management registers.	0x8

17.6.4.12.5 Function Dependency Link/NumVFs Register

Propname：Function Dependency Link/NumVFs Register
Address：@0x210
Description：This location contains the Function Dependency Link that defines VF dependencies, and the NumVFs register that stores the number of VFs configured.

Bits	SW	Name	Description	Reset
15:0	R/W	NumVFs [NVF]	This field must be set by the software to the number of VFs that it wants to enable for each PF. This field can be changed only when the VF Enable bit in the SR-IOV Control Register is 0. Its value should not exceed the setting of TotalVFs for the corresponding Physical Function. This field can also be written from the local management bus.	0x0
31:16	R	Function Dependency Link [FDL]	This field is used to specify dependencies between PFs. It can be modified independently for each Function from the local management bus.	0x0

17.6.4.12.6 VF Offset/Stride Register

Propname：VF Offset/Stride Register
Address：@0x214
Description：Specifies the offset and stride values for VF address assignment.

Bits	SW	Name	Description	Reset
15:0	R	First VF Offset [FVFO]	Offset of First VF relative to its PF. This field can be re-written independently for each PF from the local management bus.	16'd1
31:16	R	VF Stride [VFS]	Stride value used to assign RIDs for VFs. The stride value is hardwired to 1 for all Physical Functions.	0x1

17.6.4.12.7 VF Device ID Register

Propname：VF Device ID Register
Address：@0x218
Description：This register specifies the VF device id for the device.

Bits	SW	Name	Description	Reset
15:0	R	Reserved [R2]	Reserved	0x0
31:16	R	VF Device ID [VFDI]	VF device id assigned to the device. Its default value is specified in reg_defaults.h, but can be re-written independently for each PF from the local management bus.	0x0100

17.6.4.12.8 Supported Page Sizes Register

Propname：Supported Page Sizes Register
Address：@0x21c
Description：This register specifies all the page sizes supported by the device.

Bits	SW	Name	Description	Reset
15:0	R	Page Sizes [PS]	Page sizes supported by the device (one bit for each page size). The core implements only bits 15:0 of this register. The default value of this field is specified in reg_defaults.h, but can be re- written independently for each PF from the local management bus.	16'h553
31:16	R	Reserved [R0]	Reserved	0x0

17.6.4.12.9 System Page Size Register

Propname：System Page Size Register
Address：@0x220
Description：This register identifies the page size currently used by the system.

Bits	SW	Name	Description	Reset
15:0	R/W	System Page Size [SPS]	This field must be programmed by software to the current page size in use. The core implements only bits 15:0 of this register. This field can also be written from the local management bus.	0x1

Bits	SW	Name	Description	Reset
31:16	R	Reserved [R0]	Reserved	0x0

17.6.4.12.10 VF Base Address Register 0

Propname：VF Base Address Register 0
Address：@0x224
Description：This is part of the set of six Virtual Function Base Address Registers defined by the SR-IOV Specifications. These registers are used to define address ranges for memory accesses to the Endpoint device. This register may be used to define a range of 32-bit addresses, or paired with the next adjacent register to define a 64-bit address range. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1's into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if this BAR is not configured. Otherwise, the number of 1's returned is based on the length of the BAR.

Bits	SW	Name	Description	Reset
0	R	Memory Space Indicator [MSI]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O BARs.	0x0
2	R	Size [S0]	When the BAR is used to define a memory address range, this field indicates whether the address range is 32-bit or 64-bit (0 = 32-bit, 1 = 64 bit). For 64-bit address ranges, the value in BAR 1 is treated as a continuation of the base address in BAR 0. The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function.	0x1
3	R	Prefetchability [P0]	When the BAR is used to define a memory address range, this field declares whether data from the address range is prefetchable (0 = non- prefetchable, 1 = prefetchable). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
7:4	R	Reserved [R8]	These bits are hardwired to 0	0x0

Bits	SW	Name	Description	Reset
21:8	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function.	0x0

17.6.4.12.11 VF Base Address Register 1

Propname：VF Base Address Register 1
Address：@0x228
Description：This is part of the set of six Virtual Function Base Address Registers defined by the SR-IOV Specifications. These registers are used to define address ranges for memory accesses to the Endpoint device. This register may be used to define a range of 32-bit addresses. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1's into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if this BAR is not configured. Otherwise, the number of 1's returned is based on the length of the BAR.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture setting of BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

17.6.4.12.12 VF Base Address Register 2

Propname：VF Base Address Register 2
Address：@0x22c
Description：This is part of the set of six Virtual Function Base Address Registers defined by the SR-IOV Specifications. These registers are used to define address ranges for memory accesses to the Endpoint device. This register may be used to define a range of 32-bit addresses, or paired with the next adjacent register to define a 64-bit address range. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1's into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if this BAR is not configured. Otherwise, the number of 1's returned is based on the length of the BAR.

Bits	SW	Name	Description	Reset
0	R	Memory Space Indicator [MSI]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O BARs.	0x0
2	R	Size [S0]	When the BAR is used to define a memory address range, this field indicates whether the address range is 32-bit or 64-bit (0 = 32-bit, 1 = 64 bit). For 64-bit address ranges, the value in BAR 1 is treated as a continuation of the base address in BAR 0. The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function.	0x1
3	R	Prefetchability [P0]	When the BAR is used to define a memory address range, this field declares whether data from the address range is prefetchable (0 = non- prefetchable, 1 = prefetchable). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
7:4	R	Reserved [R8]	These bits are hardwired to 0	0x0
21:8	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function.	0x0

17.6.4.12.13 VF Base Address Register 3

Propname：VF Base Address Register 3
Address：@0x230
Description：This is part of the set of six Virtual Function Base Address Registers defined by the SR-IOV Specifications. These registers are used to define address ranges for memory accesses to the Endpoint device. This register may be used to define a range of 32-bit addresses. During the initial configuration of the device, the configuration program

determines the size of the address range defined by the BAR by writing a pattern of all 1's into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if this BAR is not configured. Otherwise, the number of 1's returned is based on the length of the BAR.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture setting of BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

17.6.4.12.14 VF Base Address Register 4

Propname：VF Base Address Register 4

Address：@0x234

Description：This is part of the set of six Virtual Function Base Address Registers defined by the SR-IOV Specifications. These registers are used to define address ranges for memory accesses to the Endpoint device. This register may be used to define a range of 32-bit addresses, or paired with the next adjacent register to define a 64-bit address range. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1's into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if this BAR is not configured. Otherwise, the number of 1's returned is based on the length of the BAR.

Bits	SW	Name	Description	Reset
0	R	Memory Space Indicator [MSI]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical Function	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O BARs.	0x0
2	R	Size [S0]	When the BAR is used to define a memory address range, this field indicates whether the address range is 32-bit or 64-bit (0 = 32-bit, 1 = 64 bit). For 64-bit address ranges, the value in BAR 1 is treated as a continuation of the base address in BAR 0. The value read in this field is determined by the setting of BAR Configuration Registers of the associated Physical	0x1

Bits	SW	Name	Description	Reset
3	R	Prefetchability [P0]	When the BAR is used to define a memory address range, this field declares whether data from the address range is prefetchable (0 = non- prefetchable, 1 = prefetchable). The value read in this field is determined by the setting of BAR Configuration Registers of the	0x0
7:4	R	Reserved	These bits are hardwired to 0	0x0
21:8	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in BAR Configuration Registers of the associated Physical Function.	0x0

17.6.4.12.15 VF Base Address Register 5

Propname：VF Base Address Register 5
Address：@0x238
Description：This is part of the set of six Virtual Function Base Address Registers defined by the SR-IOV Specifications. These registers are used to define address ranges for memory accesses to the Endpoint device. This register may be used to define a range of 32-bit addresses. During the initial configuration of the device, the configuration program determines the size of the address range defined by the BAR by writing a pattern of all 1's into the BAR, reading back from the BAR, and noting the position of the first 1 (the most significant) in the returned value. A value of 0 is returned by the core if this BAR is not configured. Otherwise, the number of 1's returned is based on the length of the BAR.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture setting of BAR Configuration Registers of the associated Physical Function. All other bits are not writeable, and are read as 0's.	0x0

17.6.4.12.16 VF Migration State Array Offset Register

Propname：VF Migration State Array Offset Register
Address：@0x23c
Description：Not implemented

Bits	SW	Name	Description	Reset
31:0	R	MSAOR	(no description)	0x0

17.6.4.12.17 Reserved

Propname：Reserved
Address：@0x240 + [0..12 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.4.13 i_TPH_cap_struct

17.6.4.13.1 TPH Requester Extended Capability Header Register

Propname：TPH Requester Extended Capability Header Register
Address：@0x274
Description：This location contains the PCI Express Extended Capability ID for Transaction Processing Hints (TPH) Requester Capability, its capability version, and the pointer to the next capability block.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the TPH Requester Capability.	0x0017
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified independently for each PF from the local management bus.	0x1
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h900

17.6.4.13.2 TPH Requester Capability Register

Propname：TPH Requester Capability Register
Address：@0x278
Description：This is a read-only register that specifies the capabilities associated with the implementation of the TPH in the device. All the fields in this register, except the reserved ones, can be modified from the local management bus.

Bits	SW	Name	Description	Reset
0	R	No ST Mode Supported [NSM]	When set to 1, indicates that this Function supports the 'No ST Mode' for the generation of TPH Steering Tags. In the No ST Mode, the device must use a Steering Tag value of 0 for all requests. This bit is hardwired to 1, as all TPH Requesters are required to support the No ST Mode of operation.	0x1

Bits	SW	Name	Description	Reset
1	R	Interrupt Vector Mode Supported [IVMS]	A setting of 1 indicates that the Function supports the Interrupt Vector Mode for TPH Steering Tag generation. In the Interrupt Vector Mode, Steering Tags are attached to MSI/MSI-X interrupt requests. The Steering Tag for each interrupt request is selected by the MSI/MSI-X interrupt vector number. This bit is set to 1 by default, but can be modified from the local management bus.	0x1
2	R	Device-Specific Mode Supported [DSMS]	A setting of 1 indicates that the Function supports the Device-Specific Mode for TPH Steering Tag generation. In this mode, the Steering Tags are supplied by the client for each request through the HAL master interface. The client typically chooses the Steering Tag values from the ST Table, but is not required to do so. This bit is set to 1 by default, but can be modified from the local management bus.	0x1
7:3	R	Reserved [R0]	Reserved	0x0
8	R	Extended TPH Requester Supported [ERS]	When set to 1, indicates that the Function is capable of generating requests with a TPH TLP Prefix.	0x0
10:9	R	ST Table Location [STL]	The setting of this field indicates if a Steering Tag Table is implemented for this Function, and its location if present. (00 = ST Table not present, 01 = ST Table in the TPH Requester Capability Structure, 10 = ST values stored in the MSI-X Table in client RAM, 11 = reserved.). This field can be modified from the local management bus.	0x1
15:11	R	Reserved [R1]	Reserved	0x0
26:16	R	ST Table Size [STS]	Specifies the number of entries in the Steering Tag Table (0 = 1 entry, 1 = 2 entries, and so on). Max limit is 64 entries when the ST Table is located in the TPH Requester Capability Structure, and 2048 entries when located in the MSI-X table. Each entry is 16 bits long. This field can be modified from the local management bus.	11'd7
31:27	R	Reserved [R2]	Reserved	0x0

17.6.4.13.3 TPH Requester Control Register

Propname：TPH Requester Control Register
Address：@0x27c
Description：This register can be used by the software to enable the TPH Requester Capability of the Function, and to select the mode of generation of Steering Tags.

Bits	SW	Name	Description	Reset
2:0	R/W	ST Mode [CSM]	This field selects the ST mode (000 = No Steering Tag Mode, 001 = Interrupt Vector Mode, 010 = Device-Specific Mode, other values are reserved). The TPH_ST_MODE output of the core reflects the setting of this register field. This field can also be written from the local management bus.	0x0
7:3	R	RSVD	RSVD	5'h00
9:8	R/W	TPH Requester Enable [CRE]	When set the Function is allowed to generate requests with Transaction Processing Hints. Defined Encodings are: 00b - Function operating as a Requester is not permitted to issue Requests with TPH or Extended TPH. 01b - Function operating as a Requester is permitted to issue Requests with TPH and is not permitted to issue Requests with Extended TPH. 10b - Reserved. 11b - Function operating as a Requester is permitted to issue Requests with TPH and Extended TPH.	0x00
31:10	R	Reserved [R10]	Reserved	0x0

17.6.4.13.4 TPH ST Table 0

Propname：TPH ST Table 0
Address：@0x280
Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus.

Bits	SW	Name	Description	Reset
7:0	R/W	ST Lower 0 [STL0]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST Upper 0 [STU0]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST Lower 1 [STL1]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST Upper 1 [STU1]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.4.13.5 TPH ST Table 1

Propname：TPH ST Table 1

Address：@0x284

Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus.

Bits	SW	Name	Description	Reset
7:0	R/W	ST Lower 0 [STL0]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST Upper 0 [STU0]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST Lower 1 [STL1]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST Upper 1 [STU1]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.4.13.6 TPH ST Table 2

Propname：TPH ST Table 2

Address：@0x288

Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus.

Bits	SW	Name	Description	Reset
7:0	R/W	ST Lower 0 [STL0]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST Upper 0 [STU0]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST Lower 1 [STL1]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST Upper 1 [STU1]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.4.13.7 TPH ST Table 3

Propname：TPH ST Table 3

Address：@0x28c

Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus.

Bits	SW	Name	Description	Reset
7:0	R/W	ST Lower 0 [STL0]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST Upper 0 [STU0]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST Lower 1 [STL1]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST Upper 1 [STU1]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.4.14 i_regf_L1_PM_cap_struct

17.6.4.14.1 L1 PM Substates Extended Capability Header Register

Propname：L1 PM Substates Extended Capability Header Register
Address：@0x900
Description：(no description)

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the L1 PM Substates Extended Capability Structure (001E hex).	0x01e
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified from the local management bus.	0x01
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h0

17.6.4.14.2 L1 PM Substates Capabilities Register

Propname：L1 PM Substates Capabilities Register
Address：@0x904
Description：(no description)

Bits	SW	Name	Description	Reset
0	R	PML1.2 Supported [L1PML12SUPP]	(no description)	0x1
1	R	PML1.1 Supported [L1PML11SUPP]	(no description)	0x1
2	R	ASPML1.2 Supported [L1ASPML12SUPP]	(no description)	0x1
3	R	ASPML1.1 Supported [L1ASPML11SUPP]	(no description)	0x1
4	R	L1 PML Supported [L1PMSUPP]	(no description)	0x1

Bits	SW	Name	Description	Reset
7:5	R	RSVD	RSVD	3'h0
15:8	R	Port Common Mode Restore Time [L1PrtCmMdRetrTime]	(no description)	8'hff
17:16	R	Port Power-On Time Scale [L1PrtPvrOnScale]	(no description)	0x0
18	R	RSVD	RSVD	1'h0
23:19	R	Port Power- On Time Value [R0]	(no description)	0x5
31:24	R	RSVD	RSVD	8'h00

17.6.4.14.3 L1 PM Substates Control 1 Register

Propname：L1 PM Substates Control 1 Register
Address：@0x908
Description：(no description)

Bits	SW	Name	Description	Reset
0	R/W	PML1.2 Enable [L1PML12EN]	(no description)	0x0
1	R/W	PML1.1 Enable [L1PML11EN]	(no description)	0x0
2	R/W	ASPML1.2 Enable [L1ASPML12E]	(no description)	0x0
3	R/W	ASPML1.1 Enable [L1ASPM11E]	(no description)	0x0
7:4	R	RSVD	RSVD	4'h0
15:8	R	Common Mode Restore Time [L1CmMdReStr	This field is reserved for EP.	0x0
25:16	R/W	LTR L1.2 Threshold Value [L1ThrshldVal]	(no description)	0x0
28:26	R	RSVD	RSVD	3'h0
31:29	R/W	LTR L1.2 Threshold Scale [L1ThrshldSc]	(no description)	0x0

17.6.4.14.4 L1 PM Substates Control 2 Register

Propname：L1 PM Substates Control 2 Register
Address：@0x90c

Description：(no description)

Bits	SW	Name	Description	Reset
1:0	R/W	T_POWER_ON Scale [L1PwrOnSc]	(no description)	0x0
2	R	RSVD	RSVD	1'h0
7:3	R/W	T_POWER_ON Value [L1PwrOnVal]	(no description)	0x5
31:8	R	RSVD	RSVD	24'h000000

17.6.5 Virtual Function Configuration Register Set Description

This version of the core supports a total of 8 Virtual Functions, which may be assigned among the 1 Physical Functions. The VFs occupy the address range 64 - 71 in the Function address space. The core automatically sets the offset and stride values for each Physical Functions based on the setting of the VF mode. The following sections describe the registers in detail below.

17.6.5.1 i_vf_pcie_base

17.6.5.1.1 Vendor ID and Device ID

Propname：Vendor ID and Device ID

Address：@0x0

Description： Hardwired to all 1's

Bits	SW	Name	Description	Reset
15:0	R	Vendor ID [VID]	This is the Vendor ID assigned by the PCI SIG to the manufacturer of the device The Vendor ID is set in the Vendor ID Register within the local management register block.	0xffff
31:16	R	Device ID [DID]	Device ID assigned by the manufacturer of the device. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be written independently for each Function from the local management bus.	0xffff

17.6.5.1.2 Command and Status Register

Propname：Command and Status Register

Address：@0x4

Description：This location contains the 16-bit Command Register and the 16-bit Status Register defined in PCI Specifications 3.0.

Bits	SW	Name	Description	Reset
0	R	IO-Space Enable [IOSE]	Reserved	0x0
1	R	Mem-Space Enable [MSE]	Reserved	0x0
2	R/W	Bus-Master Enable [BME]	Enables the device to issue memory requests from this Function. This field can be written from the local management bus.	0x0

Bits	SW	Name	Description	Reset
5:3	R	Reserved [R0]	Reserved	0x0
6	R	Parity Error Response Enable [PERE]	Reserved	0x0
7	R	Reserved [R1]	Reserved	0x0
8	R	SERR Enable [SE]	Reserved	0x0
9	R	Reserved [R2]	Reserved	0x0
10	R	INTx Message Disable [IMD]	Reserved	0x0
18:11	R	Reserved [R3]	Reserved	0x0
19	R	Interrupt Status [IS]	Reserved	0x0
20	R	Capabilities List [CL]	Indicates the presence of PCI Extended Capabilities registers. This bit is hardwired to 1.	0x1
23:21	R	Reserved [R4]	Reserved	0x0
24	R/WOCLR	Master Data Parity Error [MDPE]	When the Parity Error Response enable bit in the PCI Command Register of the associated Physical Function is set, the core sets this bit when it detects the following error conditions: (i) The core receives a Poisoned Completion TLP from the link in response to a request from this VF. (ii)The core sends out a poisoned write request on the link from this VF. (This bit remains 0 when the Parity Error Response enable bit in the PCI Command Register of the associated Physical Function is 0). This field can also be cleared from the local management bus by writing a 1 into this bit position. STICKY.	0x0
26:25	R	Reserved [R5]	Reserved	0x0
27	R/WOCLR	Signaled Target Abort [STA]	This bit is set when the core has sent a completion from this VF to the link with the Completer Abort status. This field can also be cleared from the local management bus by writing a 1 into this bit position. STICKY.	0x0

Bits	SW	Name	Description	Reset
28	R/WOCLR	Received Target Abort [RTA]	This bit is set when this Virtual Function has received a completion from the link with the Completer Abort status. This field can also be cleared from the local management bus by writing a 1 into this bit position. STICKY.	0x0
29	R/WOCLR	Received Master Abort [RMA]	This bit is set when this VF has received a completion from the link with the Unsupported Request status. This field can also be cleared from the local management bus by writing a 1 into this bit position. STICKY.	0x0
30	R/WOCLR	Signaled System Error [SSE]	If the SERR enable bit in the PCI Command Register of the associated Physical Function is 1, this bit is set when this VF has sent out a fatal or non-fatal error message on the link to the Root Complex. If the SERR enable bit is 0, this bit remains 0. This field can also be cleared from the local management bus by writing a 1 into this bit position. STICKY.	0x0
31	R/WOCLR	Detected Parity Error [DPE]	This bit is set when the core has received a Poisoned TLP targeted at this VF. The Parity Error Response enable bit (bit 6) in the PCI Command Register of the associated PF has no effect on the setting of this bit. STICKY.	0x0

17.6.5.1.3 Revision ID and Class Code Register

Propname：Revision ID and Class Code Register
Address：@0x8
Description：This register contains the Revision ID and Class Code associated with the device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
7:0	R	Revision ID [RID]	Assigned by the manufacturer of the device to identify the revision RO Setting of this field Denali PCIe Core Register Specification, PMC-Sierra Version 3.4 202 number of the device. This field reflects the setting of the corresponding register in the configuration space of the associated Physical Function.	8'h0
15:8	R	Programming Interface Byte [PIB]	Identifies the register set layout of the device. This field reflects the setting of the corresponding register in the configuration space of the associated Physical Function.	0x0

Bits	SW	Name	Description	Reset
23:16	R	Sub-Class Code [SCC]	Identifies a sub-category within the selected function. This field reflects the setting of the corresponding register in the configuration space of the associated Physical Function.	8'h0
31:24	R	Class Code [CC]	Identifies the function of the device. This field reflects the setting of the corresponding register in the configuration space of the associated Physical Function.	8'h0

17.6.5.1.4 BIST, Header Type, Latency Timer and Cache Line Size Registers

Propname：BIST, Header Type, Latency Timer and Cache Line Size Registers
Address：@0xc
Description：This location contains the BIST, header-type, Latency Timer and Cache Line Size Registers.

Bits	SW	Name	Description	Reset
7:0	R	Cache Line Size [CLS]	Reserved	0x0
15:8	R	Latency Timer [LT]	Reserved	0x0
22:16	R	Header Type [HT]	Reserved	0x0
23	R	Device Type [DT]	Identifies whether the device supports a single Function or multiple Functions. This bit is read as 0 when only Function 0 has been enabled in the Physical Function Configuration Register (in the local management block). Reserved for VFs	0x0
31:24	R	BIST Register [BR]	Reserved	0x0

17.6.5.1.5 Base Address Register 0

Propname：Base Address Register 0
Address：@0x10
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.6 Base Address Register 1

Propname：Base Address Register 1
Address：@0x14
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.7 Base Address Register 2

Propname：Base Address Register 2
Address：@0x18
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.8 Base Address Register 3

Propname：Base Address Register 3
Address：@0x1c
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.9 Base Address Register 4

Propname：Base Address Register 4
Address：@0x20
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.10 Base Address Register 5

Propname：Base Address Register 5
Address：@0x24
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.11 Reserved

Propname：Reserved
Address：@0x28
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.1.12 Subsystem Vendor ID and Subsystem ID Register

Propname：Subsystem Vendor ID and Subsystem ID Register
Address：@0x2c
Description：This register contains the Subsystem Vendor ID and Subsystem ID associated with the device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
15:0	R	Subsystem Vendor ID [SVID]	Specifies the Subsystem Vendor ID assigned by the PCI SIG to the manufacturer of the device. Its value comes from the Subsystem Vendor ID Register in the local management register block.	16'h17cd
31:16	R	Subsystem ID [SID]	Specifies the Subsystem ID assigned by the manufacturer of the device. This field reflects the setting of the corresponding register in the configuration space of the associated Physical Function.	16'h0

17.6.5.1.13 Expansion ROM Base Address Register

Propname：Expansion ROM Base Address Register
Address：@0x30
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.14 Capabilities Pointer

Propname：Capabilities Pointer
Address：@0x34
Description：This location contains the pointer to the first PCI Capability Structure. Its default value is defined in the RTL file reg_defaults.h.

Bits	SW	Name	Description	Reset
7:0	R	Capabilities Pointer [CP]	Contains pointer to the first PCI Capability Structure. This field is set by default to point to the Power Management Capability Structure. It can be modified by writing to VF 0 from the local management bus, and the setting is common across all VFs.	0x80
31:8	R	Reserved [R6]	Reserved	0x0

17.6.5.1.15 Reserved

Propname：Reserved
Address：@0x38
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.1.16 Interrupt Line and Interrupt Pin Register

Propname：Interrupt Line and Interrupt Pin Register
Address：@0x3c
Description：Not Implemented

Bits	SW	Name	Description	Reset
31:0	R	Not Implemented [NI]	(no description)	0x0

17.6.5.1.17 Reserved

Propname：Reserved
Address：@0x40 + [0..15 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.2 i_vf_power_mgmt_cap_struct

17.6.5.2.1 Power Management Capabilities Register

Propname：Power Management Capabilities Register
Address：@0x80
Description：This location contains the Power Management Capabilities Register, its Capability ID, and a pointer to the next capability. This version of the core supports the PCI power states D0, D1 and D3.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Identifies that the capability structure is for Power Management. This field is set by default to 01 hex. It can be re-written independently for each Function from the local management bus.	0x01
15:8	R	Capabilities Pointer [CP]	Contains pointer to the next PCI Capability Structure. The core sets it to the value defined in the RTL file reg_defaults.h. By default, this points to the MSI Capability Structure. This field can be re-written independently for each Function from the local management bus.	8'h90
18:16	R	Version ID [VID]	Indicates the version of the PCI Bus Power Management Specifications that the Function implements. This field is set by default to 011 (Version 1.2). It can be re-written independently for each Function from the local management bus.	0x03
19	R	PME Clock [PC]	Not applicable to PCI Express. This bit is hardwired to 0.	0x0
20	R	Reserved [R0]	Reserved	0x0
21	R	Device Specific Initialization Bit [DSI]	This bit, when set, indicates that the device requires additional configuration steps beyond setting up its PCI configuration space, to bring it to the D0active state from the D0uninitialized state. This bit is hardwired to 0.	0x0
24:22	R	Max Current Required from Aux Power Supply [MCRAPS]	Specifies the maximum current drawn by the device from the aux power source in the D3cold state. This field is not implemented in devices not supporting PME notification when in the D3cold state, and is therefore hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
25	R	D1 Support [D1S]	Set if the Function supports the D1 power state. This bit can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x1
26	R	D2 Support [D2S]	Set if the Function supports the D2 power state. Currently hardwired to 0.	0x0
27	R	PME Support for D0 State [PSD0S]	Indicates whether the Function is capable of sending PME messages when in the D0 state. This bit is set to 1 by default, but can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x01
28	R	PME Support for D1 State [PSD1S]	Indicates whether the Function is capable of sending PME messages when in the D1 state. This bit is set to 1 by default, but can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x1
29	R	PME Support for D2 State [PSD2S]	Indicates whether the Function is capable of sending PME messages when in the D2 state. This bit is hardwired to 0 because D2 state is not supported.	0x0
30	R	PME Support for D3(hot) State [PSDHS]	Indicates whether the Function is capable of sending PME messages when in the D3hot state. This bit is set to 1 by default, but can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x01
31	R	PME Support for D3(cold) State [PSDCS]	Indicates whether the Function is capable of sending PME messages when in the D3cold state. Because the device does not have aux power, this bit is hardwired to 0.	0x0

17.6.5.2.2 Power Management Control/Status Report

Propname：Power Management Control/Status Report
Address：@0x84
Description：This location contains the 16-bit Power Management Control/Status Register.

Bits	SW	Name	Description	Reset
1:0	R/W	Power State [PS]	Indicates the power state this Function is currently in. This field can be read by the software to monitor the current power state, or can be written to cause a transition to a new state. The valid settings are 00 (state D0), 01 (state D1) and 11 (state D3hot). The software should not write any other value into this field. This field can also be written from the local management bus independently for each VF Function.	0x0
2	R	Reserved [R4]	Reserved	0x0
3	R	No Soft Reset [NSR]	When this bit is set to 1, the Function will maintain all its state in the PM state D3hot. The software is not required to re-initialize the Function registers on the transition back to D0. This bit is set to 1 by default, but can be modified independently for each VF from the local management bus.	0x01
7:4	R	Reserved [R3]	Reserved	0x0
8	R/W	PME Enable [PE]	Setting this bit enables the notification of PME events from the associated Function. This bit can be set also by writing into this register from the local management bus.	0x0
14:9	R	Reserved [R2]	Reserved	0x0
15	R/WOCLR	PME Status [PMES]	When PME notification is enabled, writing a 1 into this bit position from the local management bus sets this bit and causes the core to send a PME message from the associated Function. When the Root Complex processes this message, it will turn off this bit by writing a 1 into this bit position though a Config Write. This bit can be set or cleared from the local management bus, by writing a 1 or 0, respectively. It can only be cleared from the configuration path (by writing a 1).	0x0
23:16	R	Reserved [R1]	Reserved	0x0
31:24	R	Data Register [DR]	This optional register is not implemented in the PCIe core. This field is hardwired to 0.	0x0

17.6.5.2.3 Reserved

Propname：Reserved
Address：@0x88 + [0..1 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.3 i_vf_MSI_cap_struct

17.6.5.3.1 MSI Control Register

Propname：MSI Control Register
Address：@0x90
Description：This register is used only when the core is configured to support Message Signaled Interrupts (MSIs). In addition to the MSI control bits, this location also contains the MSI Capability ID and the pointer to the next PCI Capability Structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Specifies that the capability structure is for MSI. Hardwired to 05 hex.	0x05
15:8	R	Capabilities Pointer [CP]	Pointer to the next PCI Capability Structure. The value read from this read-only field is the corresponding pointer in the MSI Capability Structure of the Physical Function this VF is attached to. The setting is common across all the Virtual Functions.	8'hb0
16	R/W	MSI Enable [ME]	Set by the configuration program to enable the MSI feature. This field can also be written from the local management bus.	0x0
19:17	R	Multiple Message Capable [MMC]	Encodes the number of distinct messages that the core is capable of generating for this Function (000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101= 32). Thus, this field defines the number of the interrupt vectors for this Function. The core allows up to 32 distinct messages, but the setting of this field must be based on the number of interrupt inputs of the core that are actually used by the client. For example, if the client logic uses 8 of the 32 distinct MSI interrupt inputs of the core for this Function, then the value of this field must be set to 011. This field can be written from the local management bus.	0x0
22:20	R/W	Multiple Message Enable [MME]	Encodes the number of distinct messages that the core is programmed to generate for this Function (000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101= 32). This setting must be based on the number of interrupt inputs of the core that are actually used	0x0
23	R	64-Bit Address Capable [AC64]	Set to 1 to indicate that the device is capable of generating 64-bit addresses for MSI messages.	0x1
24	R	MSI masking capable [MC]	can be modified using localmanagement interface	0x1

Bits	SW	Name	Description	Reset
31:25	R	Reserved [R0]	Reserved	0x0

17.6.5.3.2 MSI Message Low Address Register

Propname：MSI Message Low Address Register
Address：@0x94
Description：This register contains the first 32 bits of the address to be used in the MSI messages generated by the core for this Function. This address is taken as a 32-bit address if the value programmed in the MSI Message High Address Register is 0. Otherwise, this address is taken as the least significant 32 bits of the 64-bit address sent in MSI messages.

Bits	SW	Name	Description	Reset
1:0	R	Reserved [R1]	The two lower bits of the address are hardwired to 0 to align the address on a double-word boundary.	0x0
31:2	R/W	Message Address Low [MAL]	Lower bits of the address to be used in MSI messages. This field can also be written from the local management bus.	0x0

17.6.5.3.3 MSI Message High Address Register

Propname：MSI Message High Address Register
Address：@0x98
Description：This register contains the 32 most significant bits of the 64-bit address sent by the core in MSI messages. A value of all zeroes in the register is taken to mean that the core should use 32-bit addresses in the messages.

Bits	SW	Name	Description	Reset
31:0	R/W	Message Address High [MAH]	Contains bits 63:32 of the 64-bit address to be used in MSI Messages. A value of 0 specifies that 32-bit addresses are to be used in the messages. This field can also be written from the local management bus.	0x0

17.6.5.3.4 MSI Message Data Register

Propname：MSI Message Data Register
Address：@0x9c
Description：This register contains the write data to be used in the MSI messages to be generated for the associated PCI Function. When the number of distinct messages programmed in the MSI Control Register is 1, the 32-bit value from this register is used as the data value in the MSI packets generated by the core for this Function. If the number of distinct
messages is more than 1, the least significant bits of the programmed value are replaced with the encoded interrupt vector [31:0] of the specific message to generate the write data value for the message.

Bits	SW	Name	Description	Reset
15:0	R/W	Message Data [MD]	Message data to be used for this Function. This field can also be written from the local management bus.	0x0
31:16	R	Reserved [R2]	Hardwired to 0	0x0

17.6.5.3.5 MSI Mask Register

Propname：MSI Mask Register

Address：@0xa0
Description：This register contains the MSI mask bits, one for each of the interrupt levels.

Bits	SW	Name	Description	Reset
0	R/W	MSI Mask [MM]	Mask bits for MSI interrupts. The Multiple Message Capable field of the MSI Control Register specifies the number of distinct interrupts for the Function, which determines the number of valid mask bits. Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of the MSI Mask field also changes correspondingly	0x0
31:1	R	Reserved [R0]	Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of this field also changes correspondingly	0x0

17.6.5.3.6 MSI Pending Bits Register

Propname：MSI Pending Bits Register
Address：@0xa4
Description：This register contains the MSI pending interrupt bits, one for each of the interrupt levels. This field can be written from the local management APB bus.

Bits	SW	Name	Description	Reset
0	R	MSI Pending Bits [MP]	Pending bits for MSI interrupts. This register contains the MSI pending interrupt bits, one for each of the interrupt levels. This field can be written from the local management APB bus. The Multiple Message Capable field of the MSI Control Register specifies the number of distinct interrupts for the Function, which determines the number of valid pending bits. Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of the MSI Pending Bits field also changes correspondingly	0x0
31:1	R	Reserved [R0]	Please note that if the Multiple Message Capable field is changed from the local management APB bus, then the width of this field also changes correspondingly	0x0

17.6.5.3.7 Reserved

Propname：Reserved
Address：@0xa8 + [0..1 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.4 i_vf_MSIX_cap_struct

17.6.5.4.1 MSI-X Control Register

Propname：MSI-X Control Register
Address：@0xb0
Description：This register contains the MSI-X configuration bits, the Capability ID for MSI-X, and the pointer to the next PCI Capability structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Identifies that the capability structure is for MSI-X. This field is set by default to 11 hex. It can be rewritten independently for each Function from the local management bus.	0x11
15:8	R	Capabilities Pointer [CP]	Contains a pointer to the next PCI Capability Structure. The value read from this read-only field is the corresponding pointer in the MSI-X Capability Structure of the Physical Function this VF is attached to.	8'hc0
26:16	R	MSI-X Table Size [MSIXTS]	Specifies the size of the MSI-X Table, that is, the number of interrupt vectors defined for the Function. The programmed value is 1 minus the size of the table (that is, this field is set to 0 if the table size is 1.). It can be re-written independently for each Function from the local management bus.	11'h0
29:27	R	Reserved [R0]	Reserved	0x0
30	R/W	Function Mask [FM]	This bit serves as a global mask to all the interrupt conditions associated with this Function. When this bit is set, the core will not send out MSI messages from this Function. This field can also be written from the local management bus.	0x0
31	R/W	MSI-X Enable [MSIXE]	Set by the configuration program to enable the MSI-X feature. This field can also be written from the local management bus.	0x0

17.6.5.4.2 MSI-X Table Offset Register

Propname：MSI-X Table Offset Register
Address：@0xb4
Description：This register is used to specify the location of the MSI-X Table in memory. All of the 32 bits of this register can be re-written independently for each Virtual Function from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Indicator Register [BARI]	Identifies the BAR corresponding to the memory address range where the MSI-X Table is located (000 = BAR 0, 001 = BAR 1, ... , 101 = BAR 5).	3'd0

Bits	SW	Name	Description	Reset
31:3	R	Table Offset [TO]	Offset of the memory address where the MSI- X Table is located, relative to the selected BAR. The three least significant bits of the address are omitted, as the addresses are QWORD aligned.	29'h0

17.6.5.4.3 MSI-X Pending Interrupt Register

Propname：MSI-X Pending Interrupt Register
Address：@0xb8
Description：This register is used to specify the location of the MSI-X Pending Bit Array (PBA). The PBA is a structure in memory containing the pending interrupt bits. All the 32 bits of this register can be re-written independently for each Virtual Function from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Indicator Register [BARI]	Identifies the BAR corresponding to the memory address range where the PBA Structure is located (000 = BAR 0, 001 = BAR 1, ... , 101 = BAR 5). The value programmed must be the same as the BAR Indicator configured in the MSI-X TableOffset Register.Identifies the BAR corresponding to the memory address range where the PBA Structure is located (000 = BAR 0, 001 = BAR1, ... , 101 = BAR 5). The value programmed must be the same as the BAR Indicator configured in the MSI-X Table Offset Register.	3'd0
31:3	R	PBA Offset [PO]	Offset of the memory address where the PBA is located, relative to the selected BAR. The three least significant bits of the address are omitted, as the addresses are QWORD aligned.	29'h1

17.6.5.4.4 Reserved

Propname：Reserved
Address：@0xbc
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.5 i_vf_PCIE_cap_struct

17.6.5.5.1 PCI Express Capability List Register

Propname：PCI Express Capability List Register
Address：@0xc0
Description：This location identifies the PCI Express device type and its capabilities. It also contains the Capability ID for the PCI Express Structure and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Specifies Capability ID assigned by PCI SIG for this structure. This field is hardwired to 10 hex.	0x010
15:8	R	Next Capability Pointer [NCP]	Points to the next PCI capability structure. Set to 0 because this is the last capability structure.	0x0
19:16	R	Capability Version [CV]	Identifies the version number of the capability structure. The value depends on the value of the strap input PCIE_GENERATION_SEL If PCIE_GENERATION_SEL indicates Gen 2 or later generations, then the value is 2 else 1.	0x02
23:20	R	Device Type [DT]	Indicates the type of device implementing this Function. This field is hardwired to 0 in the EP mode.	0x0
24	R	Slot Status [SS]	Set to 1 when the link connected to a slot. Hardwired to 0.	0x0
29:25	R	Interrupt Message Number [IMN]	Identifies the MSI or MSI-X interrupt vector for the interrupt message generated corresponding to the status bits in the Slot Status Register, Root Status Register, or this capability structure. This field must be defined based on the chosen interrupt mode - MSI or MSI-X. This field is hardwired to 0.	0x0
30	R	TCS Routing Supported [TRS]	When set to 1, this bit indicates that the device supports routing of Trusted Configuration Requests. Not valid for Endpoints. Hardwired to 0.	0x0
31	R	Reserved [R0]	Reserved	0x0

17.6.5.5.2 PCI Express Device Capabilities Register

Propname：PCI Express Device Capabilities Register
Address：@0xc4
Description：This register advertises the capabilities of the PCI Express device encompassing this Function.

Bits	SW	Name	Description	Reset
2:0	R	Max Payload Size [MPS]	Specifies maximum payload size supported by the device. This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0	3'b001
4:3	R	Phantom Functions Supported [PFS]	This field is used to extend the tag field by combining unused Function bits with the tag bits. This field is hardwired to 00 to disable this feature.	0x0

Bits	SW	Name	Description	Reset
5	R	Extended Tag Field Supported [ETFS]	Set when device allows the tag field to be extended from 5 to 8 bits. This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0.	0x0
8:6	R	Acceptable L0S Latency [AL0SL]	Specifies acceptable latency that the Endpoint can tolerate while transitioning from L0S to L0. This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0.	0x4
11:9	R	Acceptable L1 Latency [AL1SL]	Specifies acceptable latency that the Endpoint can tolerate while transitioning from L1 to L0. This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0.	0x0
14:12	R	Reserved [R1]	Reserved	0x0
15	R	Role-Based Error Reporting [RBER]	This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0.	0x01
17:16	R	Reserved [R2]	Reserved	0x0
25:18	R	Captured Slot Power Limit Value [CSPLV]	This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0.	0x0
27:26	R	Captured Power Limit Scale [CPLS]	This field reflects the setting of the corresponding field in the PCIe Device Capability Register of PF 0.	0x0
28	R	FLR Capable [FLRC]	Set when device has Function-Level Reset capability. Hardwired to 1.	0x01
31:29	R	Reserved [R3]	Reserved	0x0

17.6.5.5.3 PCI Express Device Control and Status Register

Propname：PCI Express Device Control and Status Register
Address：@0xc8
Description：This register contains control and status bits associated with the device implementing this Function. All the read-write bits in this register can also be written from the local management bus. Likewise, bits designated as RW1C can also be cleared by writing a 1 from the local management bus.

Bits	SW	Name	Description	Reset
0	R	Enable Correctable Error Reporting [ECER]	Reserved	0x0

Bits	SW	Name	Description	Reset
1	R	Enable Non-Fatal Error Reporting [ENFER]	Reserved	0x0
2	R	Enable Fatal Error Reporting [EFER]	Reserved	0x0
3	R	Enable Unsupported Request Reporting [EURR]	Reserved	0x0
4	R	Enable Relaxed Ordering [ERO]	Reserved	0x0
7:5	R	Max Payload Size [MPS]	Reserved	0x0
8	R	Extended Tag Field Enable [ETFE]	Reserved	0x0
9	R	Enable Phantom Functions [EPF]	Reserved	0x0
10	R	Enable Aux Power [EAP]	Reserved	0x0
11	R	Enable No Snoop [EBS]	Reserved	0x0
14:12	R	Max Read Request Size [MRRS]	Reserved	0x0
15	R/W	Function-Level Reset [FLR]	Writing a 1 into this bit position generated a Function-Level Reset for the selected VF. This bit reads as 0.	0x0
16	R/WOCLR	Correctable Error Detected [CED]	Set to 1 by the core when it detects a correctable error, regardless of whether error reporting is enabled or not, and regardless of whether the error is masked	0x0
17	R/WOCLR	Non-Fatal Error Detected [NFER]	Set to 1 by the core when it detects a non-fatal error, regardless of whether error reporting is enabled or not, and regardless of whether the error is masked.	0x0
18	R/WOCLR	Fatal Error Detected [FED]	Set to 1 by the core when it detects a fatal error, regardless of whether error reporting is enabled or not, and regardless of whether the error is masked.	0x0
19	R/WOCLR	Unsupported Request Detected [URD]	Set to 1 by the core when it receives an unsupported request, regardless of whether its reporting is enabled or not.	0x0

Bits	SW	Name	Description	Reset
20	R	Aux Power Detected [APD]	Reserved	0x0
21	R	Transaction Pending [TP]	Indicates if any of the Non-Posted requests issued by the VF are still pending.	0x0
31:22	R	Reserved [R4]	Reserved	0x0

17.6.5.5.4 Link Capabilities Register

Propname：Link Capabilities Register
Address：@0xcc
Description：This register advertises the link-specific capabilities of the device incorporating the PCIe core. There are no writable bits at this location. A read to this address returns the Link Capability Register fields of Physical Function 0.

Bits	SW	Name	Description	Reset
3:0	R	Maximum Link Speed [MLS]	Indicates the maximum speed supported by the link. (2.5 GT/s, 5 GT/s per lane). This field is hardwired to 0001 (2.5GT/s) when the strap input PCIE_GENERATION_SEL is set to 0, to 0010 (5GT/s) when the strap is set to 1.	0x2
9:4	R	Maximum Link Width [MLW]	Indicates the maximum number of lanes supported by the device. This field is hardwired based on the setting of the LANE_COUNT_IN strap input.	6'd4
11:10	R	Active State Power Management [ASPM]	Indicates the level of ASPM support provided by the device. This field can be re-written independently for each Function from the local management bus. When SRIS is enabled in local management register bit, L0s capability is not supported and is forced low.	2'b11
14:12	R	L0S Exit Latency [L0SEL]	Specifies the time required for the device to transition from L0S to L0. This parameter is dependent on the Physical Layer implementation. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x2
17:15	R	L1 Exit Latency [L1EL]	Specifies the exit latency from L1 state. This parameter is dependent on the Physical Layer implementation. It is set by default to the value define in reg_defaults.h. It can be re-written independently for each Function from the local management bus.	0x3

Bits	SW	Name	Description	Reset
18	R	Clock Power Management [CPM]	Indicates that the device supports removal of referenc clocks. It is set by default to the value of the define in reg_defaults.h. It can be re-written independently for each function from the local management bus.	0x0
19	R	Surprise Down Error Reporting Capability [SDERC]	Indicates the capability of the device to report a Surprise Down error condition. This bit is hardwired to 0, as this version of the core does not support the feature.	0x0
20	R	Data Link Layer Active Reporting Capability [DLLARC]	Set to 1 if the device is capable of reporting that the DL Control and Management State Machine has reached the DL Active state. This bit is hardwired to 0, as this version of the core does not support the feature.	0x0
21	R	Link Bandwidth Notification Capability [LBNC]	A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. Reserved for Endpoint.	0x0
22	R	ASPM Optionality Compliance [AOC]	Setting this bit indicates that the device supports the ASPM Optionality feature. It can be turned off by writing a 0 to this bit position through the local management bus.	0x1
23	R	Reserved [R5]	Reserved	0x0
31:24	R	Port Number [PN]	Specifies the port number assigned to the PCI Express link connected to this device.	8'h0

17.6.5.5.5 Reserved

Propname：Reserve
Address：@0xd0 + [0..4 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.5.6 PCI Express Device Capabilities Register 2

Propname：PCI Express Device Capabilities Register 2
Address：@0xe4
Description：This register is not implemented for Virtual Functions. A read to this address returns the Device Capabilities 2 Register fields of Physical Function 0.

Bits	SW	Name	Description	Reset
3:0	R	Completion Timeout Ranges [CTR]	Specifies the Completion Timeout values supported by the device. This field is set by default to 0010 (10 ms - 250 ms). The actual timeout values are in two programmable local management registers, which allow the timeout settings of the two sub-ranges within Range B to be programmed independently.	0x02
4	R	Completion Timeout Disable Supported [CTDS]	A 1 in this field indicates that the associated Function supports the capability to turn off its Completion timeout. This bit is set to 1 by default, but can be re-written independently for each Function from the local management bus.	0x01
5	R	ARI forwarding support [AFS]	ARI forwarding supported.	0x0
6	R	OP routing supported [OPRS]	Atomic OP routing supported.	0x0
7	R	32-Bit Atomic Op Completer Supported [BAOCS32]	Hardwired to 0.	0x0
8	R	64-Bit Atomic Op Completer Supported [BAOCS64]	Hardwired to 0.	0x0
9	R	128-Bit CAS Atomic Op Completer Supported [BAOCS128]	Hardwired to 0.	0x0
10	R	Reserved [R12]	Reserved	0x0
11	R	LTR Mechanism Supported [LMS]	A 1 in this bit position indicates that the Function supports the Latency Tolerance Reporting (LTR) Capability. This bit is set to 1 by default, but can be turned off for all Physical Functions by writing into PF 0.	0x01

Bits	SW	Name	Description	Reset
13:12	R	TPH Completer Supported [TCS]	These bits, when set, indicate that the Function is capable of serving as a completer for requests with Transaction Processing Hints (TPH). It can be turned off for all Physical Functions by writing into PF 0. Defined Encodings are: 00b TPH and Extended TPH Completer not supported. 01b TPH Completer supported; Extended TPH Completer not supported. 10b Reserved. 11b Both TPH and Extended TPH Completer supported.	0x01
17:14	R	Reserved [R13]	Reserved	0x0
19:18	R	OBFF Supported [OPFFS]	A 1 in this bit position indicates that the Function supports the Optimized Buffer Flush/Fill (OBFF) capability using message signaling.	0x1
20	R	Extended Format Field Supported [EXFS]	Indicates that the Function supports the 3-bit definition of the Fmt field in the TLP header. This bit is hardwired to 1 for all Physical Functions.	0x0
21	R	End-End TLP Prefix Supported [EEPS]	Indicates whether the Function supports End-End TLP Prefixes. A 1 in this field indicates that the Function supports receiving TLPs containing End- End TLP Prefixes.	0x0
23:22	R	Max End-End TLP Prefixes [MEEP]	Indicates the maximum number of End-End TLP Prefixes supported by the Function. The supported values are: 01b 1 End-End TLP Prefix 10b 2 End- End TLP Prefixes	0x0
31:24	R	Reserved [R14]	Reserved	0x0

17.6.5.5.7 Reserved

Propname：Reserved
Address：@0xe8 + [0..5 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.6 i_vf_AER_cap_struct

17.6.5.6.1 Advanced Error Reporting (AER) Enhanced Capability Header Register

Propname：Advanced Error Reporting (AER) Enhanced Capability Header Register
Address：@0x100
Description：This is the first register in the PCI Express Advanced Error Reporting Capability Structure of a Virtual Function. This register contains the PCI Express Extended Capability ID, the capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the PCI Express AER Extended Capability Structure (0001 hex).	0x01
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field reflects the setting of the corresponding field in the AER Enhanced Capability Header Register of PF 0.	4'h2
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h140

17.6.5.6.2 Uncorrectable Error Status Register

Propname Uncorrectable Error Status Register Address @0x104
Description This register provides the status of the various uncorrectable errors detected by the PCI Express core. Software may clear any error bit by writing a 1 into the corresponding bit position. The states of the bits in the Uncorrectable Error Mask Register have no effect on the status bits of this register. The setting of an uncorrectable error status bit causes the core to generate an ERR_FATAL message if the corresponding severity bit of the Uncorrectable Error Severity Register is 1. If the severity bit is 0, however, there are two separate ways the error could be processed:(i)In certain cases, the uncorrectable error is treated as an Advisory Non-Fatal Error. These cases are treated as similar to correctable errors, causing the core to generate an ERR_COR message instead of an ERR_NONFATL message. For details on these special cases, refer to Section 6.2.3.2.4 of the PCI Express Base Specifications, Version 1.1. (ii) In all other cases, the core sends an ERR_NONFATAL message when the error is detected. In all cases, the sending of the error message can be suppressed by setting the bit corresponding to the error type in the Uncorrectable Error Mask Register. For errors that are not Function-specific, the error status bus is set in the registers belonging to all the Functions associated with the link, but only a single message is generated for the entire link. In the case of certain errors detected by the Transaction Layer, the associated TLP header is logged in the Header Log Registers. All the RW1C bits can also be cleared from the local management bus by writing a 1 into the bit position.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R0]	Reserved	0x0
4	R	Data Link Protocol Error Status [DLPER]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
11:5	R	Reserved [R1]	Reserved	0x0
12	R/WOCLR	Poisoned TLP Status [PTS]	This bit is set when the core receives a poisoned TLP from the link, targeted at this VF. This error is Function-specific. This error is considered non- fatal by default. The error is reported by sending an ERR_NONFATAL message. The header of the received TLP with error is logged in the Header Log Registers associated with the VF. STICKY.	0x0

Bits	SW	Name	Description	Reset
13	R	Flow Control Protocol Error Status [FCPES]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
14	R/WOCLR	Completion Timeout Status [CTS]	This bit is set when the completion timer associated with an outstanding request times out. This error is Function-specific. This error is considered non-fatal by default. STICKY.	0x0
15	R/WOCLR	Completer Abort Status [CAS]	This bit is set when the core has returned the Completer Abort (CA) status to a request received from the link. This error is Function-specific. The header of the received request that caused the error is logged in the Header Log Registers. STICKY.	0x0
16	R/WOCLR	Unexpected Completion Status [UCS]	This bit is set when the core has received an unexpected Completion packet from the link. This error is not Function-specific. STICKY.	0x0
17	R	Receiver Overflow Status [Rcvr_Overflow _Status]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
18	R	Malformed TLP Status [Malformed_TL_Status]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
19	R	ECRC Error Status [ECRC_Err_Stat us]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
20	R/WOCLR	Unsupported Request Error Status [URES]	This bit is set when the core has received a request from the link that it does not support. This error is not Function-specific. This error is considered non-fatal by default. In the special case described in Sections 6.2.3.2.4.1 of the PCI Express Specifications, the error is reported by sending an ERR_COR message. In all other cases, the error is reported by sending an ERR_NONFATAL message. The header of the received request that caused the error is logged in the Header Log Registers. STICKY.	0x0
21	R	Reserved [R2]	Reserved	0x0
22	R	Uncorrectable Internal Error Status [Uncorr_Int_Err_status]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
31:23	R	Reserved [R3]	Reserved	0x0

17.6.5.6.3 Uncorrectable Error Mask Register

Propname：Uncorrectable Error Mask Register
Address：@0x108

Description：This register is not implemented for Virtual Functions. The setting of the mask bits in the Uncorrectable Error Mask Register of the Physical Function apply to all associated VFs.

Bits	SW	Name	Description	Reset
31:0	R	Reserved [R4]	(no description)	0x0

17.6.5.6.4 Uncorrectable Error Severity Register

Propname：Uncorrectable Error Severity Register
Address：@0x10c
Description：This register is not implemented for Virtual Functions. The settings of the severity bits in the Uncorrectable Error Severity Register of the Physical Function apply to all associated VFs.

Bits	SW	Name	Description	Reset
31:0	R	Reserved [R8]	(no description)	0x0

17.6.5.6.5 Correctable Error Status Register

Propname：Correctable Error Status Register
Address：@0x110
Description：This register provides the status of the various correctable errors detected by the PCI Express core. Software may clear any error bit by writing a 1 into the corresponding bit position. The states of the bits in the Correctable Error Mask Register have no effect on the status bits of this register. The setting of a correctable error status bit causes the core to generate an ERR_COR error message to the Root Complex if the error is not masked in the Correctable Error Mask Register. For errors that are not Function-specific, the error status bus is set in the registers belonging to all the Functions associated with the link, but only a single message is generated for the entire link. Header logging of received TLPs does not apply to correctable errors. All the RW1C bits can also be cleared from the local management bus by writing a 1 into the bit position.

Bits	SW	Name	Description	Reset
0	R	Receiver Error Status [RES]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
5:1	R	Reserved [R12]	Reserved	0x0
6	R	Bad TP Status [BTPS]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
7	R	Bad DLLP Status [BDS]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
8	R	Replay Number Rollover Status [RNRS]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
11:9	R	Reserved [R13]	Reserved	0x0
12	R	Replay Timer Timeout Status [RTTS]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
13	R/WOCLR	Advisory Non- Fatal Error Status [ANFES]	This bit is set when an uncorrectable error occurs, which is determined to belong to one of the special cases described in Section 6.2.3.2.4 of the PCI Express 2.0 Specifications. This causes the core to generate an ERR_COR message in place of an ERR_NONFATAL message. STICKY.	0x0
14	R	Corrected Internal Error Status [CIES]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
15	R/WOCLR	Header Log Overflow Status [HLOS]	This bit is set on a Header Log Register overflow, that is, when the header could not be logged in the Header Log Register because it is occupied by a previous header. STICKY.	0x0
31:16	R	Reserved [R14]	Reserved	0x0

17.6.5.6.6 Correctable Error Mask Register

Propname：Correctable Error Mask Register
Address：@0x114
Description：The mask bits in this register control the reporting of correctable errors. For each error type in the Correctable Error Status Register, there is a corresponding bit in this register to mask its reporting. When a mask bit is set, the occurrence of the error is not reported (by asserting the CORRECTABLE_ERROR_OUT output).

Bits	SW	Name	Description	Reset
0	R	Receiver Error Mask [REM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
5:1	R	Reserved [R15]	Reserved	0x0
6	R	Bad TLP Mask [BTM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
7	R	Bad DLLP Mask [BDM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
8	R	Replay Number Rollover Mask [RNRM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
11:9	R	Reserved [R16]	Reserved	0x0
12	R	Replay Timer Timeout Mask [RTTM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
13	R	Advisory Non-Fatal Error Mask [ANFEM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
14	R	Corrected Internal Error Mask [CIEM]	This bit is not implemented for Virtual Functions. Hardwired to 0.	0x0
15	R	Header Log Overflow Mask [HLOM]	This bit, when set, masks the generation of error messages in response to a Header Log register overflow. STICKY. Header logs are shared across Vfs hence this field is reserved. This field is reserved since Header log sharing is selected for this configuration.	0x0
31:16	R	Reserved [R17]	(no description)	0x0

17.6.5.6.7 Advanced Error Capabilities and Control Register

Propname：Advanced Error Capabilities and Control Register
Address：@0x118
Description：This location contains a pointer to the first error that is reported in the Uncorrectable Error Status Register.

Bits	SW	Name	Description	Reset
4:0	R	First Error Pointer [FER]	This is a 5-bit pointer to the bit position in the Uncorrectable Error Status Register corresponding to the error that was detected first. When there are multiple bits set in the Uncorrectable Error Status Register, this field informs the software which error was observed first. To prevent the field from being overwritten before software was able to read it, this field is not updated while the status bit pointed by it in the Uncorrectable Error Status Register remains set. After the software clears this status bit, a subsequent error condition that sets any bit in the Uncorrectable Error Status Register will update the First Error Pointer. Any uncorrectable error type, including the special cases where the error is reported using an ERR_COR message, will set the First Error Pointer (assuming the software has reset the error pointed by it in the Uncorrectable Error Status Register). STICKY.	0x0

Bits	SW	Name	Description	Reset
5	R	ECRC Generation Capability [EGC]	This read-only bit indicates to the software that the device is capable of generating ECRC in packets transmitted on the link. This bit is hardwired to 0. The setting of the corresponding bit in the Advanced Error Capabilities and Control Register of PF 0 applies to all Virtual Functions.	0x0
6	R	Enable ECRC Generation [EEG]	Enables the ECRC generation on the transmit side of the core. This bit is hardwired to 0. The setting of the corresponding bit in the Advanced Error Capabilities and Control Register of PF0 applies to all Virtual Functions.	0x0
7	R	ECRC Check Capability [ECCAP]	This read-only bit indicates to the software that the device is capable of checking ECRC in packets received from the link. This bit is hardwired to0. This setting of the corresponding bit in the Advanced Error Capabilities and Control Register of PF 0 applies to all Virtual Functions.	0x0
8	R	Enable ECRC Check [ECC]	Setting this bit enables ECRC checking on the receive side of the core. This bit is hardwired to 0. The setting of the corresponding bit in the Advanced Error Capabilities and Control Register of PF 0 applies to all Virtual Functions.	0x0
9	R	Multiple Header Recording Capable [MHRC]	This bit is set when the Function has the capability to log more than one error header in its Header Log Registers. It is hardwired to 0.	0x0
10	R	Multiple Header Recording Enable [MHRE]	Setting this bit enables the Function to log multiple error headers in its Header Log Registers. It is hardwired to 0	0x0
31:11	R	Reserved [R18]	Reserved	0x0

17.6.5.6.8 Header Log Register 0

Propname Header Log Register 0 Address @0x11c

Description This is the first of a set of four registers used to capture the header of a TLP received by the core from the link upon detection of an uncorrectable error. Each Virtual Function has an independent set of Header Log Registers. When multiple bits are set in the Uncorrectable Error Status Register, the captured header corresponds to the error that was detected first, that is, the error pointed by the First Error Pointer, of the associated VF. To prevent the captured header from being over-written before the software is able to read it, this register is not updated while the status bit pointed by the First Error Pointer in the Uncorrectable Error Status Register remains set. After the software clears this status bit, a subsequent error condition that sets any bit in the Uncorrectable Error Status Register will

also cause the Header Log Registers to be updated. The double words of the TLP header are stored in the Header Log Registers with their bytes transposed. That is the byte containing the Type/Format fields of the header is stored at bit positions 31:24 of the Header Log Register 0.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 0 [HD0]	First DWORD of captured TLP header STICKY.	0x0

17.6.5.6.9 Header Log Register 1

Propname：Header Log Register 1
Address：@0x120
Description：This location contains the second Dword of the captured header of a TLP received from the link The bytes are stored in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 1 [HD1]	Second DWORD of captured TLP header STICKY.	0x0

17.6.5.6.10 Header Log Register 2

Propname：Header Log Register 2
Address：@0x124
Description：This location contains the third Dword of the captured header of a TLP received from the link The bytes are stored in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 2 [HD2]	Third DWORD of captured TLP header STICKY.	0x0

17.6.5.6.11 Header Log Register 3

Propname：Header Log Register 3
Address：@0x128
Description：If the captured TLP header is 4 Dwords long, this location contains the last Dword of the captured header of a TLP received from the link. If the captured header is a 3-Dword header, this register is unused. The bytes of the Dword are stored in this register in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header DWORD 3 [HD3]	Fourth DWORD of captured TLP header STICKY.	0x0

17.6.5.6.12 Reserved

Propname：Reserved
Address：@0x12c + [0..2 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.7 i_vf_ARI_cap_struct

17.6.5.7.1 ARI Extended Capability Header Register

Propname：ARI Extended Capability Header Register
Address：@0x140
Description：This register is used to enable the Alternate Routing ID interpretation. This register contains the PCI Express Extended Capability ID, the capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PCCID]	This field is hardwired to the Capability ID assigned by PCI-SIG to the ARI Extended Capability (000E hex).	0x0E
19:16	R	Capability Version [CV]	Specifies the SIG-assigned value for the version of the capability structure. This field is taken from the setting of the corresponding field in the ARI Extended Capability Header Register of PF 0.	0x01
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h274

17.6.5.7.2 ARI Capability Register and ARI Control Register

Propname：ARI Capability Register and ARI Control Register
Address：@0x144
Description：This location contains the ARI Capability Register and the ARI Control Register. All the fields in this register are hardwired to 0.

Bits	SW	Name	Description	Reset
31:0	R	Reserved [R13]	Reserved	0x0

17.6.5.7.3 Reserved

Propname：Reserved
Address：@0x148 + [0..74 * 0x4]
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [RSVD]	Reserved	0x0

17.6.5.8 i_vf_TPH_cap_struct

17.6.5.8.1 TPH Requester Enhanced Capability Header Register

Propname：TPH Requester Enhanced Capability Header Register
Address：@0x274
Description：This register contains the PCI Express Extended Capability ID for Transaction Processing Hints (TPH) Requester Capability, its capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the TPH Requester Capability.	0x0017
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified for all VFs by writing into this register field of Physical Function 0 from the local management bus.	0x1

Bits	SW	Name	Description	Reset
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h0

17.6.5.8.2 TPH Requester Capability Register

Propname TPH Requester Capability Register
Address @0x278
Description This is a read-only register that specifies the capabilities associated with the implementation of the TPH in the device. All fields in this register, except the reserved ones, can be modified from the local management bus.

Bits	SW	Name	Description	Reset
0	R	No ST Mode Supported [NSTM]	When set to 1, indicates that this Function supports the 'No ST Mode' for the generation of TPH Steering Tags. In the No ST Mode, the device must use a Steering Tag value of 0 for all requests. This bit is hardwired to 1, as all TPH Requesters are required to support the No ST Mode of operation.	0x1
1	R	Interrupt Vector Mode Supported [IVMS]	A setting of 1 indicates that the Function supports the Interrupt Vector Mode for TPH Steering Tag generation. In the Interrupt Vector Mode, Steering Tags are attached to MSI/MSI-X interrupt requests. The Steering Tag for each interrupt request is selected by the MSI/MSI-X interrupt vector number. This bit is set to 1 by default, but can be modified from the local management bus.	0x1
2	R	Device-Specific Mode Supported [DSMS]	A setting of 1 indicates that the Function supports the Device-Specific Mode for TPH Steering Tag generation. In this mode, the Steering Tags are supplied by the client for each request through the HAL master interface. The client typically chooses the Steering Tag values from the ST Table, but is not required to do so. This bit is set to 1 by default, but can be modified from the local management bus.	0x1
7:3	R	Reserved [R0]	Reserved	0x0
8	R	Extended TPH Requester Supported [ERS]	When set to 1, indicates that the Function is capable of generating requests with a TPH TLP Prefix.	0x0

Bits	SW	Name	Description	Reset
10:9	R	ST Table Location [STTL]	The setting of this field indicates if a Steering Tag Table is implemented for this Function, and its location if present. (00 = ST Table not present, 01 = ST Table in the TPH Requester Capability Structure, 10 = ST values stored in the MSI-X Table in client RAM, 11 = reserved.). This field can be modified from the local management bus.	0x1
15:11	R	Reserved [R1]	Reserved	0x0
26:16	R	ST Table Size [STTS]	Specifies the number of entries in the Steering Tag Table (0 = 1 entry, 1 = 2 entries, and so on). Max limit is 64 entries when the ST Table is located in the TPH Requester Capability Structure, and 2048 entries when located in the MSI-X table. Each entry is 16 bits long. This field can be modified from the local management bus.	11'd7
31:27	R	Reserved [R2]	Reserved	0x0

17.6.5.8.3 TPH Requester Control Register

Propname：TPH Requester Control Register
Address：@0x27c
Description：This register can be used by the software to enable the TPH Request capability of the Function, and to select the mode of generation of Steering Tags.

Bits	SW	Name	Description	Reset
2:0	R/W	ST Mode [STM]	This field selects the ST mode (000 = No Steering Tag Mode, 001 = Interrupt Vector Mode, 010 = Device-Specific Mode, other values are reserved). The VF_TPH_ST_MODE output of the core reflects the setting of this register field (bits 3:0 for VF 0 and so on). This field can also be written from the local management bus.	0x0
7:3	R	RSVD	RSVD	5'h00
9:8	R/W	TPH Requester Enable [TRE]	When set the Function is allowed to generate requests with Transaction Processing Hints. Defined Encodings are: 00b Function operating as a Requester is not permitted to issue Requests with TPH or Extended TPH. 01b Function operating as a Requester is permitted to issue Requests with TPH and is not permitted to issue Requests with Extended TPH. 10b Reserved. 11b Function operating as a Requester is permitted to issue Requests with TPH and Extended TPH.	0x00

Bits	SW	Name	Description	Reset
31:10	R	Reserved [R10]	Reserved	0x0

17.6.5.8.4 TPH ST Table 0

Propname：TPH ST Table 0
Address：0x280
Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus. The format of each register is shown below.

Bits	SW	Name	Description	Reset
7:0	R/W	ST 0 Lower [ST0L]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST 0 Upper [ST0U]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST 1 Lower [ST1L]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST 1 Upper [ST1U]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.5.8.5 TPH ST Table 1

Propname：TPH ST Table 1
Address：0x284
Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus. The format of each register is shown below.

Bits	SW	Name	Description	Reset
7:0	R/W	ST 0 Lower [ST0L]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST 0 Upper [ST0U]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST 1 Lower [ST1L]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST 1 Upper [ST1U]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.5.8.6 TPH ST Table 2

Propname：TPH ST Table 2

Address：0x288
Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus. The format of each register is shown below.

Bits	SW	Name	Description	Reset
7:0	R/W	ST 0 Lower [ST0L]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST 0 Upper [ST0U]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST 1 Lower [ST1L]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST 1 Upper [ST1U]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.5.8.7 TPH ST Table 3

Propname：TPH ST Table 3
Address：0x280
Description：This table stores the Steering Tags for the TPH Capability. This table has 8 entries per Function, each 16 bits long. Two of these entries occupy each 32-bit word of the table. Each of the entries can be read/written through the link (by a Configuration transaction) or through the local management bus. The format of each register is shown below.

Bits	SW	Name	Description	Reset
7:0	R/W	ST 0 Lower [ST0L]	Lower 8 bits of the first Steering Tag. This is the 8- bit Steering Tag sent out in requests.	0x0
15:8	R	ST 0 Upper [ST0U]	This field is used for the upper 8 bits of the first Steering Tag when Extended TPH Requester support is enabled.	0x0
23:16	R/W	ST 1 Lower [ST1L]	Lower 8 bits of the second Steering Tag. This is the 8-bit Steering Tag sent out in requests.	0x0
31:24	R	ST 1 Upper [ST1U]	This field is used for the upper 8 bits of the second Steering Tag when Extended TPH Requester support is enabled.	0x0

17.6.6 Root Port Configuration Register Set Description

17.6.6.1 i_rc_pcie_base

17.6.6.1.1 Vendor ID and Device ID

Propname：Vendor ID and Device ID
Address：@0x0
Description：16-bit Vendor ID register and 16-bit Device ID register.

Bits	SW	Name	Description	Reset
15:0	R	Vendor ID [VID]	This is the Vendor ID assigned by PCI SIG to the manufacturer of the device. The Vendor ID is set in the Vendor ID Register within the local management register block.	16'h17cd
31:16	R	Device ID [DID]	Device ID assigned by the manufacturer of the device. On power-up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	16'h100

17.6.6.1.2 Command and Status Register

Propname：Command and Status Register
Address：@0x4
Description：16-bit Command Register and 16-bit Status Register.

Bits	SW	Name	Description	Reset
0	R/W	IO-Space Enable [ISE]	Enables IO accesses through the core for this PCI Function.	0x0
1	R/W	Mem-Space Enable [MSE]	Enables memory accesses through the core for this PCI Function.	0x0
2	R/W	Bus-Master Enable [BE]	Enables the device to issue memory and I/O requests from this Function.	0x0
5:3	R	Reserved [R0]	Reserved	0x0
6	R/W	Parity Error Response Enable [PERE]	When this bit is 1, the core sets the Master Data Parity Error status bit when it detects the following error conditions: (i) The core receives a poisoned completion from the link in response to a request. (ii) The core sends out a poisoned write request on the link (this may be because an underflow occurred during the packet transfer at the host interface of the core.). When this bit is 0, the Master Data Parity Error status bit is never set.	0x0
7	R	Reserved [R1]	Reserved	0x0
8	R/W	SERR Enable [SE]	Enables the reporting of fatal and non-fatal errors detected by the core to the Root Complex.	0x0
9	R	Reserved [R2]	Reserved	0x0

Bits	SW	Name	Description	Reset
10	R/W	INTx Message Disabled [IMD]	Enables or disables the transmission of INTx Assert and De-assert messages from the core. The setting of this bit has no effect on the operation of the core in the RC mode.	0x0
15:11	R	Reserved [R3]	Reserved	0x0
18:16	R	Reserved [R4]	Reserved	0x0
19	R	Interrupt Status [IS]	This bit is valid only when the core is configured to support legacy interrupts. Indicates that the core has a pending interrupt, that is, the core has sent an Assert_INTx message but has not transmitted a corresponding Deassert_INTx message.	0x0
20	R	Capabilities List [CL]	Indicates the presence of PCI Extended Capabilities registers. This bit is hardwired to 1.	0x1
23:21	R	Reserved [R5]	Reserved	0x0
24	R/WOCLR	Master Data Parity Error [MDPE]	When the Parity Error Response enable bit is 1, the core sets this bit when it detects the following error conditions: (i) The core receives a poisoned request from the link. (ii) The core has sent a Poisoned Completion downstream to the link This bit remains 0 when the Parity Error Response enable bit is 0. This field can be forced to 1 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
26:25	R	Reserved [R6]	Reserved	0x0
27	R/WOCLR	Signaled Target Abort [STA]	This bit is set when the core has sent a completion to the link with the Completer Abort status. This field can also be cleared from the local management APB bus by writing a 1 into this bit position. This field can be forced to 1 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0

Bits	SW	Name	Description	Reset
28	R/WOCLR	Received Target Abort [RTA]	This bit is set when the core has received a completion from the link with the Completer Abort status. This field can also be cleared from the local management APB bus by writing a 1 into this bit position. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
29	R/WOCLR	Received Master Abort [RMA]	This bit is set when the core has received a completion from the link with the Unsupported Request status. This field can also be cleared from the local management APB bus by writing a 1 into this bit position This field can be forced to 1 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
30	R/WOCLR	Signaled System Error [SSE]	The core sets this bit (i)On receiving an error message from the link, if SERR-Enable in PCI Command Register is 1 and SERR-Enable in the Bridge Control Register is also 1. (ii)On any internal Fatal/Non-Fatal error detected, if SERR-Enable in PCI Command Register is 1. This field can also be cleared from the local management APB bus by writing a 1 into this bit position. This field can be forced to 1 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
31	R/WOCLR	Detected Parity Error [DPE]	This bit is set when the core has received a poisoned TLP. The Parity Error Response enable bit (bit 6) has no effect on the setting of this bit. This field can also be cleared from the local management bus APB by writing a 1 into this bit position. This field can be forced to 1 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0

17.6.6.1.3 Revision ID and Class Code Register

Propname：Revision ID and Class Code Register
Address：@0x8
Description：This register contains the Revision ID and Class Code associated with the device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
7:0	R	Revision ID [RID]	Assigned by the manufacturer of the device to identify the revision number of the device. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	8'h0
15:8	R	Programming Interface Byte [PIB]	Identifies the register set layout of the device. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
23:16	R	Sub-Class Code [SCC]	Identifies a sub-category within the selected function. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	8'h0
31:24	R	Class Code [CC]	Identifies the function of the device. On power- up, the core sets it to the value defined in the RTL file reg_defaults.h. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	8'h0

17.6.6.1.4 BIST, Header Type, Latency Timer and Cache Line Size Registers

Propname：BIST, Header Type, Latency Timer and Cache Line Size Registers
Address：@0xc
Description：This location contains the BIST, header-type, Latency Timer and Cache Line Size Registers.

Bits	SW	Name	Description	Reset
7:0	R/W	Cache Line Size [CLS]	Cache Line Size Register defined in PCI Specifications 3.0. This field can be read or written, both from the link and from the local management bus, but its value is not used.	0x0
15:8	R	Latency Timer [LT]	This is an unused field and is hardwired to 0.	0x0
22:16	R	Header Type [HT]	Identifies format of header. This field is hardwired to 1.	0x1
23	R	Device Type [DT]	Identifies whether the device supports a single Function or multiple Functions. Hardwired to zero	0x0
31:24	R	BIST Register [BR]	BIST control register. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0

17.6.6.1.5 Root Complex Base Address Register 0

Propname：Root Complex Base Address Register 0

Address：@0x10
Description：This is the Base Address Register 0 in the Type-1 Config Space. It can be configured as a 32-bit memory BAR, a 32-bit IO BAR, or can be paired with RC BAR 1 to form a 64-bit memory BAR. The parameters of this BAR are configured in the local management register Root Complex BAR Configuration Register.

Bits	SW	Name	Description	Reset
0	R	BAR Type [MSI0]	Specifies whether this BAR defines a memory address range or an I/O address range (0 = memory, 1 = I/O). The value read in this field is determined by the setting of Root Complex BAR Configuration Register.	0x0
1	R	Reserved [R7]	This bit is hardwired to 0 for both memory and I/O BARs.	0x0
2	R	Size [S0]	For memory BAR: This bit reads as 0 when BAR 0 is configured as a 32-bit BAR, and as 1 when configured as a 64-bit BAR. For IO BAR: This is bit 3 of the base address. The value read in this field is determined by the setting of Root Complex BAR Configuration Register.	0x1
3	R	Prefetchability [P0]	For memory BAR: This bit reads as 1 when BAR 0 is configured as a prefetchable BAR, and as 0 when configured as a non-prefetchable BAR. For IO BAR: This is bit 3 of the base address. The value read in this field is determined by the setting of Root Complex BAR Configuration Register.	0x0
21:4	R	Base Address - RO part [BAMR0]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in Root Complex BAR Configuration Register. All other bits are not writeable, and are read as 0's.	0x0
31:22	R/W	Base Address - RW part [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in Root Complex BAR Configuration Register. All other bits are not writeable, and are read as 0's.	0x0

17.6.6.1.6 Root Complex Base Address Register 1

Propname：Root Complex Base Address Register 1
Address：@0x14
Description：This is the Base Address Register 1 in the Type-1 Config Space. It can be configured as a 32-bit memory BAR, a 32-bit IO BAR, or can be paired with RC BAR 0 to form a 64-bit memory BAR. The parameters of this BAR are configured in the local management register Root Complex BAR Configuration Register.

Bits	SW	Name	Description	Reset
31:0	R/W	Base Address [BAMRW]	This field defines the base address of the memory address range. The number of implemented bits in this field determines the BAR aperture configured in Root Complex BAR Configuration Register. All other bits are not writeable, and are read as 0's.	0x0

17.6.6.1.7 Primary Bus Number, Secondary Bus Number, Subordinate Bus Number, Secondary Latency Timer

Propname：Primary Bus Number, Secondary Bus Number, Subordinate Bus Number, Secondary Latency Timer
Address：@0x18
Description：This location contains the 8-bit fields: Primary Bus Number, Secondary Bus Number, Subordinate Bus Number, Secondary Latency Timer.

Bits	SW	Name	Description	Reset
7:0	R/W	Primary Bus Number [PBN]	This field can be read and written from the local management bus, but its value is not used within the core.	0x0
15:8	R/W	Secondary Bus Number [SBN]	This field can be read and written from the local management bus, but its value is not used within the core.	0x0
23:16	R/W	Subordinate Bus Number [SUBN]	This field can be read and written from the local management bus, but its value is not used within the core.	0x0
31:24	R	Secondary Latency Timer [SLTN]	This field is not implemented.	0x0

17.6.6.1.8 IO Base, IO Limit, Secondary Status Register

Propname：IO Base, IO Limit, Secondary Status Register
Address：@0x1c
Description：This location contains the 8-bit IO Base Register, the 8-bit IO Limit Register and the 16-bit Secondary Status Registers.

Bits	SW	Name	Description	Reset
0	R	Type1 cfg IO bar size [IOBS1]	value set in Type1 cfg IO bar size(bit 20 of RC BAR CONFIG register). If type1 cfg IO bar enable bit(bit 19 in RC BAR CONFIG register) is not set, then this field will be hard coded to 0.	0x0
3:1	R	Reserved [R1]	Reserved	0x0
7:4	R	IO Base Register [IBR]	This field can be read and written from the local management bus if IO BAR is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	0x0

Bits	SW	Name	Description	Reset
8	R	Type1 cfg IO bar size [IOBS2]	value set in Type1 cfg IO bar size(bit 20 of RC BAR CONFIG register).If type1 cfg IObar enable bit(bit 19 in RC BAR CONFIG register) is not set, then this field will be hard coded to 0.	0x0
11:9	R	Reserved [R2]	Reserved	0x0
15:12	R	IO Limit Register [ILR]	This field can be read and written from the local management bus if IO BAR is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	0x0
23:16	R	Reserved [R3]	Reserved	0x0
24	R/WOCLR	Master Data Parity Error [MPE]	The core does not set this bit by itself. This bit can be cleared by writing a 1 into this bit position from the local management APB bus. This field can be forced to 1 or 0 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write. Note that this bit can be set only when the Parity Error Response Enable bit is set in the Bridge Control Register	0x0
26:25	R	Reserved [R4]	Reserved	0x0
27	R/WOCLR	Signaled Target Abort [STA]	The core does not set this bit by itself. This bit can be cleared by writing a 1 into this bit position from the local management APB bus. This field can be forced to 1 or 0 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
28	R/WOCLR	Recieved Target Abort [RTA]	The core does not set this bit by itself. This bit can be cleared by writing a 1 into this bit position from the local management APB bus. This field can be forced to 1 or 0 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
29	R/WOCLR	Received Master Abort [RMA]	The core does not set this bit by itself. This bit can be cleared by writing a 1 into this bit position from the local management APB bus. This field can be forced to 1 or 0 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0

Bits	SW	Name	Description	Reset
30	R/WOCLR	Received System Error [RSE]	The core does not set this bit by itself. This bit can be cleared by writing a 1 into this bit position from the local management APB bus. This field can be forced to 1 or 0 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
31	R/WOCLR	Detected Parity Error [DPE]	The core does not set this bit by itself. This bit can be cleared by writing a 1 into this bit position from the local management APB bus. This field can be forced to 1 or 0 from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0

17.6.6.1.9 Memory Base, Memory Limit

Propname：Memory Base, Memory Limit
Address：@0x20
Description：This location contains the 16-bit Memory Base Register and the 16-bit Memory Limit Register

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R1]	Reserved	0x0
15:4	R/W	Memory Base Register [MBR]	This field can be read and written from the local management APB bus, but its value is not used within the core.	12'h0
19:16	R	Reserved [R2]	Reserved	0x0
31:20	R/W	Memory Limit Register [MLR]	This field can be read and written from the local management APB bus, but its value is not used within the core.	12'h0

17.6.6.1.10 Prefetchable Memory Base, Prefetchable Memory Limit

Propname：Prefetchable Memory Base, Prefetchable Memory Limit
Address：@0x24
Description：This location contains the Prefetchable Memory Base Register and the Prefetchable Memory Limit Register. This register is enabled by programming the Root Complex BAR configuration register in the Local Management space

Bits	SW	Name	Description	Reset
15:0	R	Prefetchable Memory Base Register [PMBR]	This field can be read and written from the local management APB bus if prefetchable memory is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	16'h0
31:16	R	Prefetchable Memory Limit Register [PMLR]	This field can be read and written from the local management APB bus if prefetchable memory is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	16'h0

17.6.6.1.11 Prefetchable Base Upper

Propname：Prefetchable Base Upper
Address：@0x28
Description：This location contains the upper 32 bits of the Prefetchable Base Register. This register is enabled by programming the Root Complex BAR configuration register in the Local Management space.

Bits	SW	Name	Description	Reset
31:0	R	Prefetchable Base Register Upper [PBRU]	This field can be read and written from the local management APB bus if 64bit prefetchable memory is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	32'h0

17.6.6.1.12 Prefetchable Limit Upper

Propname：Prefetchable Limit Upper
Address：@0x2c
Description：This location contains the upper 32 bits of the Prefetchable Limit Register. This register is enabled by programming the Root Complex BAR configuration register in the Local Management space.

Bits	SW	Name	Description	Reset
31:0	R	Prefetchable Limit Register Upper [PLRU]	This field can be read and written from the local management APB bus if 64bit prefetchable memory is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	32'h0

17.6.6.1.13 IO Base Upper, IO Limit Upper

Propname：IO Base Upper, IO Limit Upper
Address：@0x30
Description：This location contains the upper 16 bits of the IO Base and IO Limit Registers

Bits	SW	Name	Description	Reset
15:0	R	IO Base Register Upper [IBRU]	This field can be read and written from the local management bus if 32bit IO BAR is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	0x0
31:16	R	IO Limit Register Upper [ILR]	This field can be read and written from the local management bus if 32bit IO BAR is enabled in the Root Complex BAR configuration register, else it is hardwired to zero. Its value is not used within the core.	0x0

17.6.6.1.14 Capabilities Pointer

Propname：Capabilities Pointer
Address：@0x34
Description：This location contains the pointer to the first PCI Capabilities Structure.

Bits	SW	Name	Description	Reset
7:0	R	Capabilities Pointer [CP]	Contains pointer to the first PCI Capability Structure. This field is set by default to the value defined in the RTL file reg_defaults.h. It can be re-written independently for every Function from the local management APB bus.	0x80
31:8	R	Reserved [R15]	Reserved	0x0

17.6.6.1.15 Reserved

Propname：Reserved
Address：@0x38
Description：Reserved

Bits	SW	Name	Description	Reset
31:0	R	Reserved [rsvd]	Reserved	0x0

17.6.6.1.16 Interrupt Line, Interrupt Pin Register and Bridge Control Register

Propname：Interrupt Line, Interrupt Pin Register and Bridge Control Register
Address：@0x3c
Description：This location contains the Interrupt Line Register, the Interrupt Pin Register, and the Bridge Control Register

Bits	SW	Name	Description	Reset
7:0	R/W	Interrupt Line Register [ILR]	This field can be read and written from the local management bus, but its value is not used within the core. The given reset value is for PF0.	8'hff
10:8	R	Interrupt Pin Register [IPR]	Identifies the interrupt input (A, B, C, D) to which this Functions interrupt output is connected to (01= INTA, 02 = INTB, 03 = INTC, 04 = INTD). The assignment of interrupt inputs to Functions is fixed when the core is configured. This field can be re-written independently for each Function from the local management bus. Default values - PF0: 01 (INTA), PF1: 02 (INTB).	0x01
15:11	R	Reserved [R5]	Reserved	0x0
16	R/W	Parity Error Response Enable [PERE]	This field can be read and written from the local management APB bus. It is used only to enable the Master Data Parity Error bit in the Secondary Status Register.	0x0
17	R/W	Bridge Control SERR Enable [BCSE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0

Bits	SW	Name	Description	Reset
18	R/W	ISA Enable [ISAE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0
19	R/W	VGA Enable [VGAE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0
20	R/W	VGA 16 DEcode [VGA16D]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0
21	R	Reserved [R21]	Reserved	0x0
22	R/W	Bridge Control Register Secondary Bus Reset [BCRSBR]	This field can be read and written from the local management APB bus. When set, it initiates a hot reset on the link.	0x0
31:23	R	Reserved [R23]	Reserved	0x0

17.6.6.1.17 Power Management Capabilities Register

Propname：Power Management Capabilities Register
Address：@0x80
Description：This location contains the Power Management Capabilities Register, its Capability ID, and a pointer to the next capability. In the RC mode, the settings of the fields of this register have no effect on the operation of the core

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Identifies that the capability structure is for Power Management. This field is set by default to 01 hex. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x01
15:8	R	Capabilities Pointer [CP]	Contains pointer to the next PCI Capability Structure. The core sets it to the value defined in the RTL file reg_defaults.h. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	8'h90
18:16	R	Version ID [VID]	Indicates the version of the PCI Bus Power Management Specifications that the Function implements. This field is set by default to 011 (Version 1.2). This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x3
19	R	PME Clock [PC]	Not applicable to PCI Express. This bit is hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
20	R	Reserved [R0]	Reserved	0x0
21	R	Device Specific Initialization Bit [DSI]	This bit, when set, indicates that the device requires additional configuration steps beyond setting up its PCI configuration space, to bring it to the D0 active state from the D0 uninitialized state. This bit is hardwired to 0.	0x0
24:22	R	Max Current Required from Aux Power Supply [MCRAPS]	Specifies the maximum current drawn by the device from the aux power source in the D3cold state. This field is not implemented in devices not supporting PME notification when in the D3cold state, and is therefore hardwired to 0.	0x0
25	R	D1 Support [D1S]	Set if the Function supports the D1 power state. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1
26	R	D2 Support [D2S]	Set if the Function supports the D2 power state. Currently hardwired to 0.	0x0
27	R	PME Support for D0 State [PSD0S]	Indicates whether the Function is capable of sending PME messages when in the D0 state. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x01
28	R	PME Support for D1 State [PSD1S]	Indicates whether the Function is capable of sending PME messages when in the D1 state. This bit is set to 1 by default, but can be modified from the local management bus by writing into Function 0. All other Functions assume the value set in Function 0s Power Management Capabilities Register.	0x1
29	R	PME Support for D2 State [PSD2S]	Indicates whether the Function is capable of sending PME messages when in the D2 state. This bit is hardwired to 0 because D2 state is not supported.	0x0
30	R	PME Support for D3(hot) State [PSDHS]	Indicates whether the Function is capable of sending PME messages when in the D3hot state. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x01

Bits	SW	Name	Description	Reset
31	R	PME Support for D3(cold) State [PSDCS]	Indicates whether the Function is capable of sending PME messages when in the D3cold state. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1

17.6.6.1.18 Power Management Control/Status Report

Propname Power Management Control/Status Report
Address：@0x84
Description：This location contains the Power Management Control/Status and Data Registers.

Bits	SW	Name	Description	Reset
1:0	R/W	Power State [PS]	This field can also be read or written from the local management APBbus.	0x0
2	R	Reserved [R4]	Reserved	0x0
3	R	No Soft Reset [NSR]	This bit is set to 1 by default. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x01
7:4	R	Reserved [R3]	Reserved	0x0
8	R/W	PME Enable [PE]	This bit can be set or cleared from the local management APB bus, by writing a 1 or 0, respectively.	0x0
14:9	R	Reserved [R2]	Reserved	0x0
15	R/WOCLR	PME Status [PMES]	This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
23:16	R	Reserved [R1]	Reserved	0x0
31:24	R	Data Register [DR]	This optional register is not implemented in the PCIe core. This field is hardwired to 0.	0x0

17.6.6.1.19 MSI Control Register

Propname：MSI Control Register
Address：@0x90
Description：This register is used only when the core is configured to support Message Signaled Interrupts (MSIs). In addition to the MSI control bits, this location also contains the Capability ID for MSI and the pointer to the next PCI Capability Structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID1]	Specifies that the capability structure is for MSI. Hardwired to 05 hex.	0x05
15:8	R	Capabilities Pointer [CP1]	Pointer to the next PCI Capability Structure. This can be modified from the local management bus. This field can be written from the local management bus.	8'hb0

Bits	SW	Name	Description	Reset
16	R/W	MSI Enable [ME]	Set by the configuration program to enable the MSI feature. This field can also be written from the local management bus.	0x0
19:17	R	Multiple Message Capable [MMC]	Encodes the number of distinct messages that the core is capable of generating for this Function (000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101= 32). Thus, this field defines the number of the interrupt vectors for this Function. The core allows up to 32 distinct messages, but the setting of this field must be based on the number of interrupt inputs of the core that are actually used by the client. For example, if the client logic uses 8 of the 32 distinct MSI interrupt inputs of the core for this Function, then the value of this field must be set to 011. This field can be written from the local management bus.	0x0
22:20	R/W	Multiple Message Enable [MME]	Encodes the number of distinct messages that the core is programmed to generate for this Function (000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101 = 32). This setting must be based on the number of interrupt inputs of the core that are actually used by this Function. This field can be written from the local management bus.	0x0
23	R	64-Bit Address Capable [BAC64]	Set to 1 to indicate that the device is capable of generating 64-bit addresses for MSI messages. Can be modified using local management interface	0x01
24	R	MSI masking capable [MC]	can be modified using local management interface	0x01
31:25	R	Reserved [R0]	Reserved	0x0

17.6.6.1.20 MSI Message Low Address Register

Propname: MSI Message Low Address Register
Address: @0x94
Description: This register contains the first 32 bits of the address to be used in the MSI messages generated by the core for this Function. This address is taken as a 32-bit address if the value programmed in the MSI Message High Address Register is 0. Otherwise, this address is taken as the least significant 32 bits of the 64-bit address sent in MSI messages.

Bits	SW	Name	Description	Reset
1:0	R	Reserved [R1]	The two lower bits of the address are hardwired to 0 to align the address on a double-word boundary.	0x0
31:2	R/W	Message Address Low [MAL]	Lower bits of the address to be used in MSI messages. This field can also be written from the local management bus.	0x0

17.6.6.1.21 MSI Message High Address Register

Propname：MSI Message High Address Register
Address：@0x98
Description：This register contains the most significant 32 bits of the 64-bit address sent by the core in MSI messages. A value of all zeroes in this register is taken to mean that the core should use 32-bit addresses in the messages.

Bits	SW	Name	Description	Reset
31:0	R/W	Message Address High [MAH]	Contains bits 63:32 of the 64-bit address to be used in MSI Messages. A value of 0 specifies that 32-bit addresses are to be used in the messages. This field can also be written from the local management bus.	0x0

17.6.6.1.22 MSI Message Data Register

Propname：MSI Message Data Register
Address：@0x9c
Description：This register contains the write data to be used in the MSI messages to be generated for the associated PCI Function. When the number of distinct messages programmed in the MSI Control Register is 1, the 32-bit value from this register is used as the data value in the MSI packets generated by the core for this Function. If the number of distinct messages is more than 1, the least significant bits of the programmed value are replaced with the encoded interrupt vector [31:0] of the specific message to generate the write data value for the message.

Bits	SW	Name	Description	Reset
15:0	R/W	Message Data [MD]	Message data to be used for this Function. This field can also be written from the local management bus.	0x0
31:16	R	Reserved [R2]	Hardwired to 0	0x0

17.6.6.1.23 MSI Mask Register

Propname：MSI Mask Register
Address：@0xa0
Description：This register contains the MSI mask bits, one for each of the interrupt levels.

Bits	SW	Name	Description	Reset
0	R/W	MSI Mask [MM]	Mask bits for MSI interrupts. The Multiple Message Capable field of the MSI Control Register specifies the number of distinct interrupts for the Function, which determines the number of valid mask bits.	0x0
31:1	R	RSVD	RSVD	31'h00000000

17.6.6.1.24 MSI Pending Bits Register

Propname：MSI Pending Bits Register
Address：@0xa4
Description：This register contains the MSI pending interrupt bits, one for each of the interrupt levels.

Bits	SW	Name	Description	Reset
0	R	MSI Pending Bits [MP]	Pending bits for MSI interrupts. This field can be written from the APB interface to refelct the current pending status.	0x0

Bits	SW	Name	Description	Reset
31:1	R	RSVD	RSVD	31'h00000000

17.6.6.1.25 MSI-X Control Register

Propname：MSI-X Control Register
Address：@0xb0
Description：This register contains the MSI-X configuration bits, the Capability ID for MSI-X and the pointer to the next PCI Capability Structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Identifies that the capability structure is for MSI-X. This field is set by default to 11 hex. It can be rewritten independently for each Function from the local management bus.	0x11
15:8	R	Capabilities Pointer [CP]	Contains pointer to the next PCI Capability Structure. This is set to point to the PCI Express Capability Structure at 30 hex. This can be rewritten independently for each Function from the local management bus.	8'hc0
26:16	R	MSI-X Table Size [MSIXTS]	Specifies the size of the MSI-X Table, that is, the number of interrupt vectors defined for the Function. The programmed value is 1 minus the size of the table (that is, this field is set to 0 if the table size is 1.). It can be re- written independently for each Function from the local management bus.	11'h0
29:27	R	Reserved [R0]	Reserved	0x0
30	R/W	Function Mask [FM]	This bit serves as a global mask to all the interrupt conditions associated with this Function. When this bit is set, the core will not send out MSI-X messages from this Function. This field can also be written from the local management bus.	0x0
31	R/W	MSI-X Enable [MSIXE]	Set by the configuration program to enable the MSI-X feature. This field can also be written from the local management bus.	0x0

17.6.6.1.26 MSI-X Table Offset Register

Propname：MSI-X Table Offset Register
Address：@0xb4
Description：This register is used to specify the location of the MSI-X Table in memory. All the 32 bits of this register can be re-written independently for each Function from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Indicator Register [BARI]	Identifies the BAR corresponding to the memory address range where the MSI-X Table is located (000 = BAR 0, 001 = BAR 1, ... , 101 = BAR 5).	3'd0
31:3	R	Table Offset [TO]	Offset of the memory address where the MSI- X Table is located, relative to the selected BAR. The three least significant bits of the address are omitted, as the addresses are QWORD aligned.	29'h0

17.6.6.1.27 MSI-X Pending Interrupt Register

Propname：MSI-X Pending Interrupt Register

Address：@0xb8

Description：This register is used to specify the location of the MSI-X Pending Bit Array (PBA). The PBA is a structure in memory containing the pending interrupt bits. All the 32 bits of this register can be rewritten independently for each Function from the local management bus.

Bits	SW	Name	Description	Reset
2:0	R	BAR Indicator Register [BARI1]	Identifies the BAR corresponding to the memory address range where the PBA Structure is located (000 = BAR 0, 001 = BAR 1, ... , 101 = BAR 5). The value programmed must be the same as the BAR Indicator configured in the MSI-X Table Offset Register. Identifies the BAR corresponding to the memory address range where the PBA Structure is located (000 = BAR 0, 001 = BAR1, ... , 101 = BAR 5). The value programmed must be the same as the BAR Indicator configured in the MSI-X Table Offset Register.	3'd0
31:3	R	PBA Offset [PBAO]	Offset of the memory address where the PBA is located, relative to the selected BAR. The three least significant bits of the address are omitted, as the addresses are QWORD aligned.	29'h1

17.6.6.1.28 PCI Express Capability List Register

Propname：PCI Express Capability List Register

Address @0xc0

Description：This location identifies the PCI Express device type and its capabilities. It also contains the Capability ID for the PCI Express Structure and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
7:0	R	Capability ID [CID]	Specifies Capability ID assigned by PCI SIG for this structure. This field is hardwired to 10 hex.	0x010

Bits	SW	Name	Description	Reset
15:8	R	Next Capability Pointer [NCP]	Points to the next PCI capability structure. Set to 0 because this is the last capability structure.	0x0
19:16	R	Capability Version [PCV]	Identifies the version number of the capability structure. The value depends on the value of the strap input PCIE_GENERATION_SEL If PCIE_GENERATION_SEL indicates Gen 2 or later generations, then the value is 2 else 1. Can be modified using local management interface after asserting input signal MGMT_TYPE1_CONFIG_REG_ACCESS high.	0x2
23:20	R	Device Type [DT]	Indicates the type of device implementing this Function. This field is hardwired to 4 in the RP mode.	0x4
24	R	Slot Implemented [SI]	When Set, this bit indicates that the Link associated with this Port is connected to a slot	1'b1
29:25	R	Interrupt Message Number [IMN]	Identifies the MSI or MSI-X interrupt vector for the interrupt message generated corresponding to the status bits in the Slot Status Register, Root Status Register, or this capability structure. This field must be defined based on the chosen interrupt mode - MSI or MSI-X. This field is hardwired to 0.	0x0
30	R	TCS Routing Supported [TRS]	When set to 1, this bit indicates that the device supports routing of Trusted Configuration Requests. Not valid for Endpoints. Hardwired to 0.	0x0
31	R	Reserved [R0]	Reserved	0x0

17.6.6.1.29 PCI Express Device Capabilities Register

Propname：PCI Express Device Capabilities Register
Address：@0xc4
Description：This register advertises the capabilities of the PCI Express device.

Bits	SW	Name	Description	Reset
2:0	R	Max Payload Size [MP]	Specifies maximum payload size supported by the device. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	3'b001
4:3	R	Phantom Functions Supported [PFS]	This field is used to extend the tag field by combining unused Function bits with the tag bits. This field is hardwired to 00 to disable this feature.	0x0

Bits	SW	Name	Description	Reset
5	R	Extended Tag Field Supported [ETFS]	hard coded to zero .	0x0
8:6	R	Acceptable L0S Latency [AL0L]	Specifies acceptable latency that the Endpoint can tolerate while transitioning from L0S to L0. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
11:9	R	Acceptable L1 Latency [AL1L]	Specifies acceptable latency that the Endpoint can tolerate while transitioning from L1 to L0. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
14:12	R	Reserved [R3]	Reserved	0x0
15	R	Role-Based Error Reporting [RER]	Enables role-based error reporting. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	1'b1
17:16	R	Reserved [R4]	Reserved	0x0
25:18	R	Captured Slot Power Limit Value [CSP]	Specifies upper limit on power supplied by slot. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
27:26	R	Captured Power Limit Scale [CPLS]	Specifies the scale used by Slot Power Limit Value. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
28	R	Function level reset capability [FLRC]	A value of 1b indicates the Function supports the optional Function Level Reset mechanism	0x0
31:29	R	Reserved [R5]	Reserved	0x0

PCI Express Device Control and Status Register
Propname：PCI Express Device Control and Status Register
Address：@0xc8
Description：This register contains control and status bits associated with the device.

Bits	SW	Name	Description	Reset
0	R/W	Enable Correctable Error Reporting [ECER]	This bit is not used by the core in Root Port mode.	0x0

Bits	SW	Name	Description	Reset
1	R/W	Enable Non-Fatal Error Reporting [ENFER]	This bit is not used by the core in Root Port mode.	0x0
2	R/W	Enable Fatal Error Reporting [EFER]	This bit is not used by the core in Root Port mode.	0x0
3	R/W	Enable Unsupported Request Reporting [EURR]	Enables the sending of error messages by the core on receiving unsupported requests.	0x0
4	R/W	Enable Relaxed Ordering [ERO]	When set, this bit indicates that the device is allowed to set the Relaxed Ordering bit in the Attributes field of transactions initiated from it. when the transactions do not require Strong Ordering.	0x1
7:5	R/W	Max Payload Size [MP]	Specifies the maximum TLP payload size configured. The device must be able to receive a TLP of this maximum size, and should not generate TLP's larger than this value. Software must set this field based on the maximum payload size in the Device Capabilities Register, and the capability of the other side.	0x0
8	R	Extended Tag Enable [ETE]	extended tag not enabled. Hence hard coded to zero .	0x0
9	R	phantum functions enable [PFE]	Hardwired to 0	0x0
10	R	aux power PM enable [APPME]	Hardwired to 0	0x0
11	R/W	Enable no snoop [ENS]	If this bit is Set, the Function is permitted to Set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency.	0x1
14:12	R/W	Max Read Request Size [MRR]	Specifies the maximum size allowed in read requests generated by the device.	3'b010
15	R	Reserved [R7]	Hardwired to 0.	0x0
16	R/WOCLR	Correctable Error Detected [CED]	Set to 1 by the core when it detects a correctable error, regardless of whether the error is masked.	0x0
17	R/WOCLR	Non-Fatal Error Detected [NFED]	Set to 1 by the core when it detects a non-fatal error, regardless of whether the error is masked.	0x0

Bits	SW	Name	Description	Reset
18	R/WOCLR	Fatal Error Detected [FED]	Set to 1 by the core when it detects a fatal error, regardless of whether the error is masked.	0x0
19	R/WOCLR	Unsupported Request Detected [URD]	Set to 1 by the core when it receives an unsupported request.	0x0
20	R	Aux Power Detected [APD]	Set when auxiliary power is detected by the device. This is an unused field.	0x0
21	R	Transaction Pending [TP]	Indicates if any of the Non-Posted requests issued by the RC are still pending.	0x0
31:22	R	Reserved [R8]	(no description)	0x0

17.6.6.1.30 Link Capabilities Register

Propname：Link Capabilities Register
Address：@0xcc
Description：This register advertises the link-specific capabilities of the device incorporating the PCIe core.

Bits	SW	Name	Description	Reset
3:0	R	Max Link Speed [MLS]	Indicates the speeds supported by the link (2.5 GT/s, 5 GT/s per lane). This field is hardwired to 0001 (2.5GT/s) when the strap input PCIE_GENERATION_SEL is set to 0, to 0010 (5GT/s) when the strap is set to 1.	0x2
9:4	R	Max Link Width [MLW]	Indicates the maximum number of lanes supported by the device. This field is hardwired based on the setting of the LANE_COUNT_IN strap input.	0x4
11:10	R	Active State Power Management [ASPM]	Indicates the level of ASPM support provided by the device. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x3
14:12	R	L0S Exit Latency [L0EL]	Specifies the time required for the device to transition from L0S to L0. This parameter is dependent on the Physical Layer implementation. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x2
17:15	R	L1 Exit Latency [L1EL]	Specifies the exit latency from L1 state. This parameter is dependent on the Physical Layer implementation. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x3

Bits	SW	Name	Description	Reset
18	R	Clock Power Management [CPM]	Indicates that the device supports removal of reference clocks. Not supported in this version of the core. Hardwired to 0.	0x0
19	R	Surprise Down Error Reporting Capability [SERC]	Indicates the capability of the device to report a Surprise Down error condition. This bit is hardwired to 0, as this version of the core does not support the feature.	0x0
20	R	Data Link Layer Active Reporting Capability [DARC]	Set to 1 if the device is capable of reporting that the DL Control and Management State Machine has reached the DL_Active state. This bit is hardwired to 0, as this version of the core does not support the feature.	0x0
21	R	Link Bandwidth Notification Capability [LBNC]	A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1
22	R	ASPM Optionality Compliance [ASPMOC]	A 1 in this position indicates the device supports the ASPM Optionality feature. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1
23	R	Reserved [R9]	Reserved	0x0
31:24	R	Port Number [PN]	Specifies the port number assigned to the PCI Express link connected to this device. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0

17.6.6.1.31 Link Control and Status Register

Propname: Link Control and Status Register
Address: @0xd0
Description: This register contains control and status bits specific to the PCI Express link.

Bits	SW	Name	Description	Reset
1:0	R/W	Active State Power Management Control [ASPMC]	Controls the level of ASPM support on the PCI Express link associated with the function. The valid setting are 00: ASPM disabled 01: L0s entry enabled, L1 disabled 10: L1 entry enabled, L0s disabled 11: Both L0s and L1 enabled. Note that these Control bits can be enabled only if the corresponding ASPM Support bit is 1.	0x0

Bits	SW	Name	Description	Reset
2	R	Reserved [R10]	Reserved	0x0
3	R	Read Completion Boundary [RCB]	Indicates the Read Completion Boundary of the Root Port (0 = 64 bytes, 1 = 128 bytes). This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
4	R/W	Link Disable [LD]	Writing a 1 to this bit position causes the LTSSM to go to the Disable Link state. The LTSSM stays in the Disable Link state while this bit is set.	0x0
5	R	Retrain Link [RL]	Setting this bit to 1 causes the LTSSM to initiate link training. This bit always reads as 0.	0x0
6	R/W	Common Clock Configuration [CCC]	A value of 0 indicates that the reference clock of this device is asynchronous to that of the upstream device. A value of 1 indicates that the reference clock is common.	0x0
7	R/W	Extended Synch [ES]	Set to 1 to extend the sequence of ordered sets transmitted while exiting from the L0S state.	0x0
8	R	Enable Clock Power Management [ECPM]	This field is hardwired to 0 when the core is in the RC mode.	0x0
9	R	Reserved [R9]	Reserved	0x0
10	R/W	Link Bandwidth Management Interrupt Enable [LBMIE]	When Set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been Set. This enables an interrupt to be generated through PHY_INTERRUPT_OUT if triggered. Hardwired to 0 if Link Bandwidth Notification Capability is 0.	0x0
11	R/W	Link Autonomous Bandwidth Interrupt Enable [LABIE]	When Set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been Set. This enables an interrupt to be generated through PHY_INTERRUPT_OUT if triggered. Hardwired to 0 if Link Bandwidth Notification Capability is 0.	0x0
15:12	R	Reserved [R11]	Reserved	0x0
19:16	R	Negotiated Link Speed [NLS]	Negotiated link speed of the device. The only supported speed ids are 2.5 GT/s per lane (0001),5 GT/s per lane (0010) .	0x1

Bits	SW	Name	Description	Reset
25:20	R	Negotiated Link Width [NLW]	Set at the end of link training to the actual link width negotiated between the two sides.	0x4
26	R	Reserved [R12]	Reserved	0x0
27	R	Link Training Status [LTS]	This bit is set to 1 when the LTSSM is in the Recovery or Configuration states, or if a 1 has been written to the Retrain Link bit but the link training has not yet begun.	0x0
28	R	Slot Clock Configuration [SCC]	Indicates that the device uses the reference clock provided by the connector. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
29	R	Data Link Layer Active [DA]	Indicates the status of the Data Link Layer. Set to 1 when the DL Control and Management State Machine has reached the DL_Active state. This bit is hardwired to 0 in this version of the core.	0x0
30	R/WOCLR	Link Bandwidth Management Status [LBMS]	This bit is Set by hardware to indicate that either link training has completed following write to retrain link bit, or when HW has changed link speed or width to attempt to correct unreliable link operation. This triggers an interrupt to be generated through PHY_INTERRUPT_OUT if enabled. Hardwired to 0 if Link Bandwidth Notification Capability is 0.	0x0
31	R/WOCLR	Link Autonomous Bandwidth Status [LABS]	This bit is Set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This triggers an interrupt to be generated through PHY_INTERRUPT_OUT if enabled. Hardwired to 0 if Link Bandwidth Notification Capability is 0.	0x0

17.6.6.1.32 Slot Capability Register

Propname：Slot Capability Register
Address：@0xd4
Description：The Slot Capabilities register identifies PCI Express slot specific capabilities.

Bits	SW	Name	Description	Reset
0	R/W	Attention Button Present [ABPRSNT]	When Set, this bit indicates that an Attention Button for this slot is electrically controlled by the chassis.	1'b0

Bits	SW	Name	Description	Reset
1	R/W	Power Controller Present [PCP]	When Set, this bit indicates that a software programmable Power Controller is implemented for this slot/adapter (depending on form factor).	1'b0
2	R/W	MRL Sensor Present [MRLSP]	When Set, this bit indicates that an MRL Sensor is implemented on the chassis for this slot.	1'b0
3	R/W	Attention Indicator Present [AIP]	When Set, this bit indicates that an Attention Indicator is electrically controlled by the chassis.	1'b0
4	R/W	Power Indicator Present [PIP]	When Set, this bit indicates that a Power Indicator is electrically controlled by the chassis for this slot.	1'b0
5	R/W	Hot-Plug Surprise [HPS]	When Set, this bit indicates that an adapter present in this slot might be removed from the system without any prior notification. This is a form factor specific capability. This bit is an indication to the operating system to allow for such removal without impacting continued software operation.	1'b0
6	R/W	Hot-Plug Capable [HPC]	When Set, this bit indicates that this slot is capable of supporting hot-plug operations.	1'b0
14:7	R/W	Slot Power Limit Value [SPLV]	In combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot (see Section 6.9) or by other means to the adapter. Power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field except when the Slot Power Limit Scale field equals 00b (1.0x) and Slot Power Limit Value exceeds EFh, the following alternative encodings are used: F0h = 250 W Slot Power Limit F1h = 275 W Slot Power Limit F2h = 300 W Slot Power Limit F3h to FFh= Reserved for Slot Power Limit values above 300 W This register must be implemented if the Slot Implemented bit is Set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. The default value prior to hardware/ firmware initialization is 0000 0000b.	8'b00000000

Bits	SW	Name	Description	Reset
16:15	R/W	Slot Power Limit Scale [SPLS]	Specifies the scale used for the Slot Power Limit Value. Range of Values: 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x This register must be implemented if the Slot Implemented bit is Set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. The default value prior to hardware/firmware initialization is 00b.	2'b00
17	R/W	Electromechanical Interlock Present [EIP]	When Set, this bit indicates that an Electromechanical Interlock is implemented on the chassis for this slot.	1'b0
18	R/W	No Command Completed Support [NCCS]	When Set, this bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller. This bit is only permitted to be Set if the hot-plug capable Port is able to accept writes to all fields of the Slot Control register without delay between successive writes.	1'b0
31:19	R/W	Physical Slot Number [PSN]	This field indicates the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is unique within the chassis, regardless of the form factor associated with the slot. This field must be initialized to zero for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port.	13'd0

17.6.6.1.33 Slot Control and Status Register

Propname：Slot Control and Status Register

Address：@0xd8

Description：This register contains control bits specific to PCI Express slot parameters and status bits specific to the PCI Express Slot. All the read-write bits in this register can also be written from the local management APB bus.

Bits	SW	Name	Description	Reset
0	R/W	Attention Button Pressed Enable [ABPE]	When Set to 1b, this bit enables software notification on an attention button pressed event. If the Attention Button Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	1'b0

Bits	SW	Name	Description	Reset
1	R/W	Power Fault Detected Enable [PFDE]	When Set, this bit enables software notification on a power fault event If a Power Controller that supports power fault detection is not implemented, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	1'b0
2	R/W	MRL Sensor Changed Enable [MSCE]	When Set, this bit enables software notification on a MRL sensor changed event If the MRL Sensor Present bit in the Slot Capabilities register is Clear, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	1'b0
3	R/W	Presence Detect Changed Enable [PDCE]	When Set, this bit enables software notification on a presence detect changed event. If the Hot-Plug Capable bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	1'b0
4	R/W	Command Completed Interrupt Enable [CCIE]	If Command Completed notification is supported (if the No Command Completed Support bit in the Slot Capabilities register is 0b), when Set, this bit enables software notification when a hot-plug command is completed by the Hot-Plug Controller. If Command Completed notification is not supported, this bit must be hardwired to 0b. Default value of this bit is 0b.	1'b0
5	R/W	Hot-Plug Interrupt Enable [HPIE]	When Set, this bit enables generation of an interrupt on enabled hot-plug events. If the Hot Plug Capable bit in the Slot Capabilities register is Clear, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	1'b0
7:6	R/W	Attention Indicator Control [AIC]	If an Attention Indicator is implemented, writes to this field set the Attention Indicator to the written state. Reads of this field must reflect the value from the latest write, Defined encodings are: 00b Reserved 01b On 10b Blink 11b Off	2'b11

Bits	SW	Name	Description	Reset
9:8	R/W	Power Indicator Control [PIC]	If a Power Indicator is implemented, writes to this field set the Power Indicator to the written state. Reads of this field must reflect the value from the latest write, Defined encodings are: 00b Reserved 01b On 10b Blink 11b Off	2'b11
10	R/W	Power Controller Control [PCC]	If a Power Controller is implemented, this bit when written sets the power state of the slot per the defined encodings. Reads of this bit must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write, if required to, without waiting for the previous command to complete in which case the read value is undefined. The defined encodings are: 0b Power On 1b Power Off	1'b1
11	R/W	Electromechanical Interlock Control [EMIC]	If an Electromechanical Interlock is implemented, a write of 1b to this bit causes the state of the interlock to toggle. A write of 0b to this bit has no effect. A read of this bit always returns a 0b.	1'b0
12	R/W	Data Link Layer State Changed Enable [DLLSCE]	If the Data Link Layer Link Active Reporting capability is 1b, this bit enables software notification when Data Link Layer Link Active bit is changed. If the Data Link Layer Link Active Reporting Capable bit is 0b, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	1'b0
15:13	R	Reserved [RSCS1]	Reserved	0x0
16	R/WOCLR	Attention Button Pressed [ABPRSD]	If an Attention Button is implemented, this bit is Set when the attention button is pressed. If an Attention Button is not supported, this bit must not be Set.	0x0

Bits	SW	Name	Description	Reset
17	R/WOCLR	Power Fault Detected [PFD]	If a Power Controller that supports power fault detection is implemented, this bit is Set when the Power Controller detects a power fault at this slot. Note that, depending on hardware capability, it is possible that a power fault can be detected at any time, independent of the Power Controller Control setting or the occupancy of the slot. If power fault detection is not supported, this bit must not be Set.	0x0
18	R/WOCLR	MRL Sensor Changed [MRLSC]	If an MRL sensor is implemented, this bit is Set when a MRL Sensor state change is detected. If an MRL sensor is not implemented, this bit must not be Set.	0x0
19	R/WOCLR	Presence Detect Changed [PDC]	This bit is set when the value reported in the Presence Detect State bit is changed.	0x0
20	R/WOCLR	Command Completed [CMDCMPL]	If Command Completed notification is supported (if the No Command Completed Support bit in the Slot Capabilities register is 0b), this bit is Set when a hot-plug command has completed and the Hot-Plug Controller is ready to accept a subsequent command. The Command Completed status bit is Set as an indication to host software that the Hot- Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete. If Command Completed notification is not supported, this bit must be hardwired to 0b.	0x0
21	R	MRL Sensor State [MRLSS]	This bit reports the status of the MRL sensor if implemented. Defined encodings are: 0b MRL Closed 1b MRL Open	0x0

Bits	SW	Name	Description	Reset
22	R	Presence Detect State [PDS]	This bit indicates the presence of an adapter in the slot, reflected by the logical "OR" of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot's corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism. Defined encodings are: 0b Slot Empty 1b Card Present in slot.	0x0
23	R	Electromechanical Interlock Status [EMIS]	If an Electromechanical Interlock is implemented, this bit indicates the status of the Electromechanical Interlock. Defined encodings are: 0b Electromechanical Interlock Disengaged 1b Electromechanical Interlock Engaged	0x0
24	R/WOCLR	Data Link Layer State Changed [DLLSC]	This bit is Set when the value reported in the Data Link Layer Link Active bit of the Link Status register is changed. In response to a Data Link Layer State Changed event, software must read the Data Link Layer Link Active bit of the Link Status register to determine if the Link is active before initiating configuration cycles to the hot plugged device.	0x0
31:25	R	Reserved [RSCS2]	(no description)	0x0

17.6.6.1.34 Root Control and Capability Register

Propname: Root Control and Capability Register
Address: @0xdc
Description: This register controls and identifies PCI Express Root Complex specific parameters.

Bits	SW	Name	Description	Reset
0	R/W	System Error on Correctable Error Enable [SECEE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0

Bits	SW	Name	Description	Reset
1	R/W	System Error on Non-Fatal Error Enable [SENFEE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0
2	R/W	PME Interrupt Enable [PMEIE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0
3	R/W	CRS Software Visibility Enable [CRSSVE]	This field can be read and written from the local management APB bus, but its value is not used within the core.	0x0
31:4	R	Reserved [R28]	Reserved	0x0

17.6.6.1.35 Root Status Register

Propname：Root Status Register
Address：@0xe0
Description：This register controls and identifies PCI Express Root Complex specific parameters.

Bits	SW	Name	Description	Reset
15:0	R	PME Requester ID [PMERID]	This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
16	R/WOCLR	PME Status [PMES]	This field is not set by the core but can be cleared by writing a 1 from the local management APB bus. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
17	R	PME Pending [PMEP]	This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x0
31:18	R	Reserved [R18]	Reserved	0x0

17.6.6.1.36 PCI Express Device Capabilities 2 Register

Propname：PCI Express Device Capabilities 2 Register
Address：@0xe4
Description：This register advertises the capabilities of the PCI Express device.

Bits	SW	Name	Description	Reset
3:0	R	Completion Timeout Ranges [CTR]	Specifies the Completion Timeout values supported by the device. This field is set by default to 0010 (10 ms - 250 ms), but can be modified from the local management APB bus. The actual timeout values are in two programmable local management registers, which allow the timeout settings of the two sub-ranges within Range B to be programmed independently. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	4'b0010
4	R	Completion Timeout Disable Supported [CTDS]	A 1 in this field indicates that the associated Function supports the capability to turn off its Completion timeout. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write	1'b1
5	R	ARI Forwarding Supported [AFS]	A 1 in this bit indicates that the device is able to forward TLPs with function number greater than 8. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1
6	R	Atomic OP routing supported [AOPRS]	Applicable only to Switch Upstream Ports, Switch Downstream Ports, and Root Ports; must be 0b for other Function types. This bit must be set to 1b if the Port supports this optional capability. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1
7	R	32-Bit Atomic Op Completer Supported [ACS32]	Hardwired to 0.	0x0
8	R	64-bit Atomic Op Completer Supported [ACS64]	Hardwired to 0.	0x0
9	R	128-bit CAS Atomic Op Completer Supported [ACS128]	Hardwired to 0.	0x0

Bits	SW	Name	Description	Reset
10	R	Reserved [R14]	Reserved	0x0
11	R	LTR mechanism supported [LMS]	A value of 1b indicates support for the optional Latency Tolerance Reporting (LTR) mechanism. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write.	0x1
12	R	TPH Completer Supported [TPHC]	These bits, when set, indicate that the Function is capable of serving as a completer for requests with Transaction Processing Hints (TPH). It can be turned off for all Physical Functions by writing into PF 0. Defined Encodings are: 00b TPH and Extended TPH Completer not supported. 01b TPH Completer supported; Extended TPH Completer not supported. 10b Reserved. 11b Both TPH and Extended TPH Completer supported.	0x01
13	R	RSVD	RSVD	1'h0
17:14	R	Reserved [R15]	Reserved	0x0
19:18	R	OBFF Supported [OBFF]	A 1 in this bit position indicates that the Function supports the Optimized Buffer Flush/Fill (OBFF) capability using message signaling. This field can be written from the APB bus by setting [21] bit high of the pcie_mgmt_APB_ADDR during a local management register write	0x1
20	R	Extended Format Field Supported [EXFS]	Indicates that the Function supports the 3-bit definition of the Fmt field in the TLP header. This bit is hardwired to 1 for all Physical Functions.	0x0
21	R	End-End TLP Prefix Supported [EEPS]	hard coded to zero.	0x0
23:22	R	Max End-End TLP Prefixes [MEEP]	hard coded to zero.	0x0
31:24	R	Reserved [R16]	Reserved	0x0

17.6.6.1.37 PCI Express Device Control and Status 2 Register

Propname：PCI Express Device Control and Status 2 Register
Address：@0xe8
Description：This register contains control and status bits associated with the device.

Bits	SW	Name	Description	Reset
3:0	R/W	Completion Timeout Value [CTV]	Specifies the Completion Timeout value for the device. Allowable values are 0101 (sub-range 1) and 0110 (sub-range 2). The corresponding timeout values are stored in the local management register's Completion Timeout Interval Registers 0 and 1, respectively.	0x0
4	R/W	Completion Timeout Disable [CTD]	Setting this bit disables the Completion Timeout in the device.	0x0
5	R/W	ARI Forwarding Enable [AFE]	A 1 in this filed indicates that the port treats fields 7:0 of the ID as function number while converting a Type 1 config packet to type 0 config packet.	0x0
6	R	Atomic Op Requester Enable [AORE]	This bit must be set to enable the generation of Atomic Op Requests. If the client logic attempts to send an Atomic Op when this bit is not set, logic in the core will nullify the TLP on its way to the link.	0x0
7	R	Reserved [R18]	Reserved	0x0
8	R	IDO Request Enable [IRE]	When this bit is 1, the RC is allowed to set the ID- based Ordering (IDO) Attribute bit in the requests it generates.	0x0
9	R	IDO Completion Enable [ICE]	When this bit is 1, the RC is allowed to set the ID-based Ordering (IDO) Attribute bit in the Completions it generates.	0x0
10	R/W	LTR Mechanism Enable [LTRME]	This must be set to 1 to enable the Latency Tolerance Reporting Mechanism. This bit is implemented only in PF 0. Its default value is 1, but can be modified from the local management bus. This bit is read-only in PF 1.	0x0
12:11	R	Reserved [R19]	Reserved	0x0
14:13	R/W	OBFF Enable [OBFFE]	Enables the Optimized Buffer Flush/Fill (OBFF) capability in the device. Valid settings are 00 (disabled), 01 (Variation A), and 10 (Variation B).	0x0
31:15	R	Reserved [R20]	(no description)	0x0

17.6.6.1.38 Link Capabilities Register 2

Propname：Link Capabilities Register 2
Address：@0xec
Description：This register advertises the supported link speeds of the core.

Bits	SW	Name	Description	Reset
0	R	RSVD	RSVD	1'h0
2:1	R	Supported Link Speeds Vector [SLSV]	This field indicates the supported link speeds of the core. For each bit, a value of 1 indicates that the corresponding link speed is supported, while a value of 0 indicates that the corresponding speed is not supported. This field is RsvdP for Gen1, Gen2 configurations.	0x0
31:3	R	RSVD	RSVD	29'h00000000

17.6.6.1.39 Link Control and Status 2 Register

Propname：Link Control and Status 2 Register
Address：@0xf0
Description：This register contains control and status bits specific to the PCI Express link.

Bits	SW	Name	Description	Reset
3:0	R/W	Target Link Speed [TLS]	This field sets the target speed when the software forces the link into Compliance mode by setting the Enter Compliance bit in this register (0001= 2.5 GT/s, 0010 = 5 GT/s, 0100 = 8 GT/s). The default value of this field is 0001 (2.5 GT/s) when the PCIE_GENERATION_SEL[1:0] strap pins of the core are set to 0, 0010 (5 GT/s) when the strap is set to 1. STICKY.	
4	R/W	Enter Compliance [EC]	This bit is used to force the Endpoint device to enter the Compliance mode. Software sets this bit to 1 and initiates a hot reset to force the device into the Compliance mode. The target speed for the Compliance mode is determined by the Target Link Speed field of this register. STICKY.	0x0
5	R/W	Hardware Autonomous Speed Disable [HASD]	When this bit is set, the LTSSM is prevented from changing the operating speed of the link, other than reducing the speed to correct unreliable operation of the link. STICKY	0x0
6	R/W	Selectable De-Emphasis [SD]	This bit selects the de-emphasis level when the core is operating at 5 GT/s (0 = -6 dB, 1 = -3.5 dB).	0x0

Bits	SW	Name	Description	Reset
9:7	R/W	Transmit Margin [TM]	This field is intended for debug and compliance testing purposes only. It controls the non- deemphasized voltage level at the transmitter outputs. Its encodings are: 000 = Normal operating range, 001 = 800 - 1200 mV for full swing and 400 - 700 mV for half swing, 010 - 111 = See PCI Express Base Specification 2.0. This field is reset to 0 when th LTSSM enters the Polling.Configuration substate during link training. STICKY.	0x0
10	R/W	Enter Modified Compliance [EMC]	This field is intended for debug and compliance testing purposes only. If this bit is set to 1, the device will transmit the Modified Compliance Pattern when the LTSSM enters the Polling.Compliance substate. STICKY	0x0
11	R/W	Compliance SOS [CS]	When this bit is set to 1, the device will transmit SKP ordered sets between compliance patterns. STICKY	0x0
15:12	R/W	Compliance De-Emphasis [CD]	This bit sets the de-emphasis level (for 5 GT/s operation) or the Transmitter Preset level (for 8 GT/s operation) when the LTSSM enters the Polling.Compliance state because of software setting the Enter Compliance bit in this register. It is used only when the link is running at 5 GT/s or 8 GT/s. At 5 GT/s, the only valid setting are 0 (-6dB) and 1 (-3.5 dB). STICKY	0x0
16	R	Current De-Emphasis Level [CDEL]	This status bit indicates the current operating de- emphasis level of the transmitter (0 = -6dB, 1 = -3.5dB).	0x1
21:17	R	Reserved [R20]	Reserved	0x0
31:22	R	Reserved [R19]	Reserved	0x0

17.6.6.1.40 Advanced Error Reporting (AER) Enhanced Capability Header Register

Propname：Advanced Error Reporting (AER) Enhanced Capability Header Register
Address：@0x100
Description：This is the first register in the PCI Express Advanced Error Reporting Capability Structure. This register contains the PCI Express Extended Capability ID, the capability version, and the pointer to the next capability structure.

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the PCI Express AER Extended Capability Structure (0001 hex).	16'h1

Bits	SW	Name	Description	Reset
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 4'h2.	0x2
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h274

17.6.6.1.41 Uncorrectable Error Status Register

Propname：Uncorrectable Error Status Register

Address：@0x104

Description：This register provides the status of the various uncorrectable errors detected by the PCI Express core. Software may clear any error bit by writing a 1 into the corresponding bit position. The states of the bits in the Uncorrectable Error Mask Register have no effect on the status bits of this register. In the case of certain errors detected by the Transaction Layer, the associated TLP header is logged in the Header Log Registers.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R25]	(no description)	0x0
4	R/WOCLR	Data Link Protocol Error Status [DLPE]	This bit is set when the core receives an Ack or Nak DLLP whose sequence does not correspond to that of an unacknowledged TLP or that of the last acknowledged TLP (for details, refer to the PCI Express Base Specifications).	0x0
11:5	R	Reserved [R26]	Reserved	0x0
12	R/WOCLR	Poisoned TLP Status [PT]	This bit is set when the core receives a poisoned TLP from the link. This error is considered non-fatal by default. The header of the received TLP with error is logged in the Header Log Registers.	0x0
13	R/WOCLR	Flow Control Protocol Error Status [FCPE]	This bit is set when certain violations of the flow control protocol are detected by the core.	0x0
14	R/WOCLR	Completion Timeout Status [CT]	This bit is set when the completion timer associated with an outstanding request times out. This error is considered non-fatal by default.	0x0
15	R/WOCLR	Completer Abort Status [CA]	This bit is set when the core has returned the Completer Abort (CA) status to a request received from the link. This error is considered non-fatal by default, except for the special cases outlined in PCI Express Base Specification 2.0. The header of the received request that caused the error is logged in the Header Log Registers.	0x0

Bits	SW	Name	Description	Reset
16	R/WOCLR	Unexpected Completion Status [UC]	This bit is set when the core has received an unexpected Completion packet from the link.	0x0
17	R/WOCLR	Receiver Overflow Status [RO]	This bit is set when the core receives a TLP in violation of the receive credit currently available.	0x0
18	R/WOCLR	Malformed TLP Status [MT]	This bit is set when the core receives a malformed TLP from the link. This error is considered fatal by default. The header of the received TLP with error is logged in the Header Log Registers.	0x0
19	R/WOCLR	ECRC Error Status [EE]	This bit is set when the core has detected an ECRC error in a received TLP.	0x0
20	R/WOCLR	Unsupported Request Error Status [URE]	This bit is set when the core has received a request from the link that it does not support. This error is not Function-specific. This error is considered non-fatal by default, except for the special case outlined in PCI Express Base Specification 2.0. The header of the received request that caused the error is logged in the Header Log Registers.	0x0
21	R	Reserved [R27]	Reserved	0x0
22	R/WOCLR	Uncorrectable Internal Error Status [UIE]	This bit is set when the core has detected an internal uncorrectable error (HAL parity error or an uncorrectable ECC error while reading from any of the RAMs). This bit is also set in response to the client signaling an internal error through the input UNCORRECTABLE_ERROR_IN. This error is considered fatal by default.	0x0
31:23	R	Reserved [R28]	Reserved	0x0

17.6.6.1.42 Uncorrectable Error Mask Register

Propname：Uncorrectable Error Mask Register
Address：@0x108
Description The mask bits in this register control the reporting of uncorrectable errors. For each error type in the Uncorrectable Error Status Register, there is a corresponding bit in this register to mask its reporting. Setting the mask bit has the following effects: (1) The occurrence of the error does not cause activation of the FATAL_ERROR_OUT or NON_FATAL_ERROR_OUT output of the core, depending on the severity of the error. (2) The header of the TLP in which the error was detected is not logged in the Header Log Registers. (3) The First Error Pointer in the Advanced Error Capabilities and Control Register is not updated on the detection of the error. The individual bits of the mask register are described below.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R29]	Reserved	0x0

Bits	SW	Name	Description	Reset
4	R/W	Data Link Protocol Error Mask [DLPER]	This bit is set to mask the reporting of Data Link Protocol Errors. STICKY.	0x0
11:5	R	Reserved [R30]	Reserved	0x0
12	R/W	Poisoned TLP Mask [PTM]	This bit is set to mask the reporting of a Poisoned TLP. STICKY.	0x0
13	R/W	Flow Control Protocol Error Mask [FCPER]	This bit is set to mask the reporting of Flow Control Protocol Errors. STICKY.	0x0
14	R/W	Completion Timeout Mask [CTM]	This bit is set to mask the reporting of Completion Timeouts. STICKY.	0x0
15	R/W	Completer Abort Mask [CAM]	This bit is set to mask the reporting of the core sending a Completer Abort. STICKY.	0x0
16	R/W	Unexpected Completion Mask [UCM]	This bit is set to mask the reporting of unexpected Completions received by the core. STICKY.	0x0
17	R/W	Receiver Overflow Mask [ROM]	This bit is set to mask the reporting of violations of receive credit. STICKY.	0x0
18	R/W	Malformed TLP Mask [MTM]	This bit is set to mask the reporting of malformed TLPs received from the link. STICKY.	0x0
19	R/W	ECRC Error Mask [EEM]	This bit is set to mask the reporting of ECRC errors. STICKY.	0x0
20	R/W	Unsupported Request Error Mask [UREM]	This bit is set to mask the reporting of unexpected requests received from the link. STICKY.	0x0
21	R	Reserved [R31]	Reserved	0x0
22	R/W	Uncorrectable Internal Error Mask [UIEM]	This bit is set to mask the reporting of internal errors. STICKY.	0x1
31:23	R	Reserved [R32]	Reserved	0x0

17.6.6.1.43 Uncorrectable Error Severity Register

Propname：Uncorrectable Error Severity Register
Address：@0x10c
Description：The setting of this register determines whether an uncorrectable error is reported as a fatal error or non- fatal, that is, whether the FATAL_ERROR_OUT or NON_FATAL_ERROR_OUT output of the core is activated. If a severity bit of this register is 0, the corresponding error is reported by the core by asserting NON_FATAL_ERROR_OUT. Otherwise, it is reported by asserting FATAL_ERROR_OUT.

Bits	SW	Name	Description	Reset
3:0	R	Reserved [R33]	Reserved	0x0
4	R/W	Data Link Protocol Error Severity [DLPES]	Severity of Data Link Protocol Errors (0 = Non- Fatal, 1 = Fatal). STICKY.	1'b1

Bits	SW	Name	Description	Reset
5	R	surprise down error severity [SDES]	surprise down error severity. This field is hard coded to 1.	1'b1
11:6	R	Reserved [R35]	(no description)	0x0
12	R/W	Poisoned TLP Severity [PTS]	Severity of a Poisoned TLP error (0 = Non-Fatal, 1= Fatal). STICKY.	0x0
13	R/W	Flow Control Protocol Error Severity [FCPES]	Severity of a Flow Control Protocol Error (0 = Non-Fatal, 1 = Fatal). STICKY.	1'b1
14	R/W	Completion Timeout Severity [CTS]	Severity of Completion Timeouts (0 = Non-Fatal, 1= Fatal). STICKY.	0x0
15	R/W	Completer Abort Severity [CAS]	Severity of sending a Completer Abort (0 = Non- Fatal, 1 = Fatal). STICKY.	0x0
16	R/W	Unexpected Completion Severity [UCS]	Severity of unexpected Completions received by the core (0 = Non-Fatal, 1 = Fatal). STICKY.	0x0
17	R/W	Receiver Overflow Severity [ROS]	Severity of receive credit violations (0 = Non- Fatal, 1 = Fatal). STICKY.	1'b1
18	R/W	Malformed TLP Severity [MTS]	Severity of malformed TLPs received from the link (0 = Non-Fatal, 1 = Fatal). STICKY.	1'b1
19	R/W	ECRC Error Severity [EES]	Severity of ECRC errors (0 = Non-Fatal, 1 = Fatal). STICKY.	0x0
20	R/W	Unsupported Request Error Severity [URES]	Severity of unexpected requests received from the link (0 = Non-Fatal, 1 = Fatal). STICKY.	0x0
21	R	Reserved [R36]	Reserved	0x0
22	R/W	Uncorrectable Internal Error Severity [Uncorr_Intrnal_Err_Svrty]	Severity of internal errors (0 = Non-Fatal, 1 =Fatal).	1'b1
31:23	R	Reserved [R37]	(no description)	0x0

17.6.6.1.44 Correctable Error Status Register

Propname：Correctable Error Status Register
Address：@0x110
Description：This register provides the status of the various correctable errors detected by the PCI Express core. Software may clear any error bit by writing a 1 into the corresponding bit position. The states of the bits in the Correctable Error Mask Register have no effect on the status bits of this register. The setting of a correctable error status bit causes the core to assert the CORRECTABLE_ERROR_OUT output if the error is not masked in the Correctable Error Mask Register. Header logging of received TLPs does not apply to correctable errors.

Bits	SW	Name	Description	Reset
0	R/WOCLR	Receiver Error Status [RES]	This bit is set when an error is detected in the receive side of the Physical Layer of the core (e.g. an 8b10b decode error).	0x0

Bits	SW	Name	Description	Reset
5:1	R	Reserved [R37]	Reserved	0x0
6	R/WOCLR	Bad TP Status [BTS]	This bit is set when an error is detected in a received TLP by the Data Link Layer of the core the conditions causing this error are (1) an LCRC error, (2) the packet terminates with EDB symbol, but its LCRC field does not equal the inverted value of the calculated CRC.	0x0
7	R/WOCLR	Bad DLLP Status [BDS]	This bit is set when an LCRC error is detected in a received DLLP, and no errors were detected by the Physical Layer.	0x0
8	R/WOCLR	Replay Number Rollover Status [RNRS]	This bit is set when the replay count rolls over after three re-transmissions of a TLP at the Data Link Layer of the core.	0x0
11:9	R	Reserved [R38]	Reserved	0x0
12	R/WOCLR	Replay Timer Timeout Status [RTTS]	This bit is set when the replay timer in the Data Link Layer of the core times out, causing the core to re-transmit a TLP.	0x0
13	R/WOCLR	Advisory Non- Fatal Error Status [ANES]	This bit is set when an uncorrectable error occurs, which is determined to belong to one of the special cases described in the PCI Express Base Specification 2.0. This causes the core to assert the CORRECTABLE_ERROR_OUT output in place of NON_FATAL_ERROR_OUT.	0x0
14	R/WOCLR	Corrected Internal Error Status [CIES]	This bit is set when the core has detected an internal correctable error condition (a correctable ECC error while reading from any of the RAMs). This bit is also set in response to the client signaling an internal error through the input CORRECTABLE_ERROR_IN.	0x0
15	R/WOCLR	Header Log Overflow Status [HLOS]	This bit is set on a Header Log Register overflow, that is, when the header could not be logged in the Header Log Register because it is occupied by a previous header.	0x0
31:16	R	Reserved [R39]	Reserved	0x0

17.6.6.1.45 Correctable Error Mask Register

Propname：Correctable Error Mask Register
Address：@0x114
Description：The mask bits in this register control the reporting of correctable errors. For

each error type in the Correctable Error Status Register, there is a corresponding bit in this register to mask its reporting. When a mask bit is set the occurrence of the error is not reported (by asserting the CORRECTABLE_ERROR_OUT output).

Bits	SW	Name	Description	Reset
0	R/W	Receiver Error Mask [REM]	This bit, when set, masks the reporting of Physical Layer errors. STICKY.	0x0
5:1	R	Reserved [R40]	Reserved	0x0
6	R/W	Bad TP Mask [BTM]	This bit, when set, masks the reporting of an error in response to a 'Bad TLP' received. STICKY.	0x0
7	R/W	Bad DLLP Mask [BDM]	This bit, when set, masks the reporting of an error in response to a 'Bad DLLP' received. STICKY.	0x0
8	R/W	Replay Number Rollover Mask [RNRM]	This bit, when set, masks the reporting of an error in response to a Replay Number Rollover event. STICKY.	0x0
11:9	R	Reserved [R41]	Reserved	0x0
12	R/W	Replay Timer Timeout Mask [RTTM]	This bit, when set, masks the reporting of an error in response to a Replay Timer timeout event. STICKY.	0x0
13	R/W	Advisory Non- Fatal Error Mask [ANEM]	This bit, when set, masks the reporting of an error in response to an uncorrectable error occurrence, which is determined to belong to one of the special cases in the PCI Express Base Specification 2.0. STICKY.	0x1
14	R/W	Corrected Internal Error Mask [CIEM]	This bit, when set, masks the reporting of an error in response to a corrected internal error condition. STICKY.	1'b1
15	R/W	Header Log Overflow Mask [HLOM]	This bit, when set, masks the reporting of an error in response to a Header Log register overflow. STICKY.	1'b1
31:16	R	Reserved [R42]	Reserved	0x0

17.6.6.1.46 Advanced Error Capabilities and Control Register

Propname：Advanced Error Capabilities and Control Register
Address：@0x118
Description：This register contains a pointer to the first error that is reported in the Uncorrectable Error Status Register, and bits to enable ECRC generation and checking.

Bits	SW	Name	Description	Reset
4:0	R	First Error Pointer [FEP]	This is a 5-bit pointer to the bit position in the Uncorrectable Error Status Register corresponding to the error that was detected first. When there are multiple bits set in the Uncorrectable Error Status Register, this field informs the software which error was observed first. To prevent the field from being overwritten before the software is able to read it, this field is not updated while the status bit it points to in the Uncorrectable Error Status Register remains set. After the software clears this status bit, a subsequent error condition that sets any bit in the Uncorrectable Error Status Register will update the First Error Pointer. Any uncorrectable error type, including the special cases where the error is reported using an ERR_COR message, will set the First Error Pointer (assuming the software has reset the error pointed by it in the Uncorrectable Error Status Register). STICKY.	0x0
5	R	ECRC Generation Capability [EGC]	This read-only bit indicates to the software that the device is capable of generating ECRC in packets transmitted on the link.	0x1
6	R/W	Enable ECRC Generation [EEG]	Setting this bit enables the ECRC generation on the transmit side of the core. This bit is writable from the local management bus. STICKY.	0x0
7	R	ECRC Check Capability [ECC]	This read-only bit indicates to the software that the device is capable of checking ECRC in packets received from the link.	0x1
8	R/W	Enable ECRC Check [EEC]	Setting this bit enables ECRC checking on the receive side of the core. This bit is writable from the local management bus. STICKY.	0x0
9	R	Multiple Header Recording Capable [MHRC]	This bit is set when the RC has the capability to log more than one error header in its Header Log Registers. It is hardwired to 0.	0x0
10	R	Multiple Header Recording Enable [MHRE]	Setting this bit enables the RC to log multiple error headers in its Header Log Registers. It is hardwired to 0.	0x0
31:11	R	Reserved [R43]	Reserved	0x0

17.6.6.1.47 Header Log Register 0

Propname：Header Log Register 0
Address：@0x11c
Description：This is the first of a set of four registers used to capture the header of a TLP received by the core from the link upon detection of an uncorrectable error. When multiple bits are set in the Uncorrectable Error Status Register, the captured header corresponds the error that was detected first, that is, the error pointed by the First Error Pointer. To prevent the captured header from being over-written before software is able to read it, this register is not updated while the status bit pointed to by the First Error Pointer in the Uncorrectable Error Status Register remains set. After the software clears this status bit, a subsequent error condition that sets any bit in the Uncorrectable Error Status Register will also cause the Header Log Registers to be updated. The double words of the TLP header are stored in the Header Log Registers with their bytes transposed. That is, the byte containing the Type/Format fields of the header is stored at bit positions 31:24 of the Header Log Register 0.

Bits	SW	Name	Description	Reset
31:0	R	Header Dword 0 [HD0]	First Dword of captured TLP header. STICKY.	0x0

17.6.6.1.48 Header Log Register 1

Propname：Header Log Register 1
Address：@0x120
Description：This register contains the second Dword of the captured TLP header. The bytes are stored in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header Dword 1 [HD1]	Second Dword of captured TLP header. STICKY.	0x0

17.6.6.1.49 Header Log Register 2

Propname：Header Log Register 2
Address：@0x124
Description：This register contains the third Dword of the captured TLP header. The bytes are stored in transposed order.

Bits	SW	Name	Description	Reset
31:0	R	Header Dword 2 [HD2]	Third Dword of captured TLP header. STICKY.	0x0

17.6.6.1.50 Header Log Register 3

Propname：Header Log Register 3
Address：@0x128
Description：If the captured TLP header is 4 Dwords long, this register contains its fourth Dword. If the captured header is a 3-Dword header, this register is unused. The bytes of the Dword are stored in this register in transposed order

Bits	SW	Name	Description	Reset
31:0	R	Header Dword 3 [HD3]	Fourth Dword of captured TLP header. STICKY.	0x0

17.6.6.1.51 Root Error Command Register

Propname：Root Error Command Register
Address：@0x12c
Description：This register contains bits that control how the RC responds to errors reported by remote devices.

Bits	SW	Name	Description	Reset
0	R/W	Correctable Error Reporting Enable [CERE]	If this bit is set, the core will active its CORRECTABLE_ERROR_OUT output in response to an error message received from the link.	0x0
1	R/W	Non-Fatal Error Reporting Enable [NFERE]	If this bit is set, the core will active its NON_FATAL_ERROR_OUT output in response to an error message received from the link.	0x0
2	R/W	Fatal Error Reporting Enable [FERE]	If this bit is set, the core will active its FATAL_ERROR_OUT output in response to an error message received from the link.	0x0
31:3	R	Reserved [R44]	Reserved	0x0

17.6.6.1.52 Root Error Status Register

Propname：Root Error Status Register
Address：@0x130
Description：This register contains status information on error messages received from the link (that is, errors reported by remote devices attached to this Root Complex).

Bits	SW	Name	Description	Reset
0	R/WOCLR	Correctable Error Message Received [ECR]	This bit is set when the RC receives a Correctable error message from the link. STICKY	0x0
1	R/WOCLR	Multiple Correctable Error Messages Received [MECR]	This bit is set when the RC receives a Correctable error message from the link, if the ERR_COR received bit is already set. STICKY	0x0
2	R/WOCLR	Fatal/Non-Fatal Error Message Received [EFNR]	This bit is set when the RC receives either a Fatal or Non-Fatal error message from the link. STICKY	0x0
3	R/WOCLR	Multiple Fatal/ Non-Fatal Error Messages Received [MEFNR]	This bit is set when the RC receives either a Fatal or Non-Fatal error message from the link, and the ERR_FATAL/NONFATAL Received bit is already set. STICKY	0x0
4	R/WOCLR	First Uncorrectable Fatal [FUF]	This bit, when set, indicates that the first Uncorrectable error message received was for a Fatal error. STICKY	0x0
5	R/WOCLR	Non-Fatal Error Messages Received [NEMR]	This bit, when set, indicates that the RC has received one or more Non-Fatal error messages from the link. STICKY	0x0

Bits	SW	Name	Description	Reset
6	R/WOCLR	Fatal Error Messages Received [FEMR]	This bit, when set, indicates that the RC has received one or more Fatal error messages from the link. STICKY	0x0
31:7	R	Reserved [R45]	Reserved	0x0

17.6.6.1.53 Error Source Identification Register

Propname：Error Source Identification Register
Address：@0x134
Description：This register stores the Register IDs extracted from error messages received by the Root Complex from the link.

Bits	SW	Name	Description	Reset
15:0	R	Correctable Error Message Source ID [ECSI]	This field captures and stores the Requester ID from an ERR_COR message received by the RC, if the ERR_COR bit was not set at the time the message was received. STICKY	0x0
31:16	R	Fatal/Non-Fatal Error Message Source ID [EFNSI]	This field captures and stores the Requester ID from an ERR_FATAL or ERROR_NONFATAL message received by the RC, if the ERR_FATAL or NONFATAL Received bit was not set at the time the message was received. STICKY	0x0

17.6.6.2 i_regf_L1_PM_cap_struct

17.6.6.2.1 L1 PM Substates Extended Capability Header Register

Propname：L1 PM Substates Extended Capability Header Register
Address：@0x900
Description：(no description)

Bits	SW	Name	Description	Reset
15:0	R	PCI Express Extended Capability ID [PECID]	This field is hardwired to the Capability ID assigned by PCI SIG to the L1 PM Substates Extended Capability Structure (001E hex).	0x01e
19:16	R	Capability Version [CV]	Specifies the SIG assigned value for the version of the capability structure. This field is set by default to 1, but can be modified from the local management bus.	0x01
31:20	R	Next Capability Offset [NCO]	Indicates offset to the next PCI Express capability structure. The default next pointer value is dynamic and is dependent on whether the strap or LMI bits are set.	12'h0

17.6.6.2.2 L1 PM Substates Capabilities Register

Propname：L1 PM Substates Capabilities Register
Address：@0x904
Description：(no description)

Bits	SW	Name	Description	Reset
2	R	ASPML1.2Supported [L1ASPML12SUPP]	(no description)	0x1
3	R	ASPML1.1Supported [L1ASPML11SUPP]	(no description)	0x1
4	R	L1 PML Supported [L1PMSUPP]	(no description)	0x1
7:5	R	RSVD	RSVD	3'h0
15:8	R	Port Common Mode Restore Time[L1PrtCmMdReStrTime]	(no description)	8'hff
17:16	R	Port Power-On Time Scale [L1PrtPvrOnScale]	(no description)	0x0
18	R	RSVD	RSVD	1'h0
23:19	R	Port Power- On Time Value [R0]	(no description)	0x5
31:24	R	RSVD	RSVD	8'h00

17.6.6.2.3 L1 PM Substates Control 1 Register

Propname：L1 PM Substates Control 1 Register
Address：@0x908
Description：(no description)

Bits	SW	Name	Description	Reset
0	R/W	PML1.2 Enable [L1PML12EN]	(no description)	0x0
1	R/W	PML1.1 Enable [L1PML11EN]	(no description)	0x0
2	R/W	ASPML1.2 Enable [L1ASPML12EN]	(no description)	0x0
3	R/W	ASPML1.1 Enable [L1ASPML11EN]	(no description)	0x0
7:4	R	RSVD	RSVD	4'h0
15:8	R/W	Common Mode Restore Time [L1CmMdReStrTime]	(no description)	0x0
25:16	R/W	LTR L1.2 Threshold Value [L1ThrshldVal]	(no description)	0x0
28:26	R	RSVD	RSVD	3'h0
31:29	R/W	LTR L1.2 Threshold Scale [L1ThrshldSc]	(no description)	0x0

17.6.6.2.4 L1 PM Substates Control 2 Register

Propname：L1 PM Substates Control 2 Register
Address：@0x90c
Description：(no description)

Bits	SW	Name	Description	Reset
1:0	R/W	T_POWER_ON Scale [L1PwrOnSc]	(no description)	0x0
2	R	RSVD	RSVD	1'h0
7:3	R/W	T_POWER_ON Value [L1PwrOnVal]	(no description)	0x5
31:8	R	RSVD	RSVD	24'h000000

17.6.7 Local Management Registers Description

The local management registers are used to configure various operational parameters

associated with the core, and for a local processor to monitor its status. These registers are accessible only from the local management bus. The Local Management registers are listed in the sections below.

17.6.7.1 i_regf_lm_pcie_base

17.6.7.1.1 Physical Layer Configuration Register 0

Propname：Physical Layer Configuration Register 0

Address：@0x0

Description：This register contains the configured parameters at the Physical Layer of the link, and status information from the Physical Layer.

Bits	SW	Name	Description	Reset
0	R	Link Status [LS]	Current state of link (1 = link training complete, 0 = link training not complete).	
2:1	R	Negotiated Lane Count [NLC]	Lane count negotiated with other side during link training (00 = x1, 01 = x2, 10 = x4, 11 = x8).	0x0 0x2
4:3	R	Negotiated Speed [NS]	Current operating speed of link (00 = 2.5G, 01 = 5G, 10 = 8G).	0x0
5	R	Link Training Direction [LTD]	The state of this bit indicates whether the core completed link training as an upstream port or a downstream port (0 = upstream, 1 = downstream). Default value depends on CORE_TYPE strap pin.	0x1
6	R/W	Phy Error Reporting [APER]	If set to 0, the core will only report those errors that caused a TLP or DLLP to be dropped because of a detected phy error, TLP framing error or DLLP framing error. When set to 1, the core will report all detected phy errors regardless of whether a TLP or DLLP was dropped but does not include TLP nor DLLP framing errors. Detected phy errors include:- received errors indicated on PIPE RxStatus interface, and TLP or DLLP framing errors depending on the programmed value of this bit.	0x0
7	R/W	Tx Swing Setting [TSS]	This bit drives the PIPE_TX_SWING output of the core.	0x0
15:8	R	Received FTS Count for 2.5 GT/s speed [RFC]	FTS count received from the other side during link training for use at the 2.5 GT/s link speed. The core transmits this many FTS sequences while exiting the L0S state, when operating at the 2.5 GT/s speed.	0x0
23:16	R	Received Link ID [RLID]	Link ID received from other side during link training.	0x0
29:24	R	LTSSM State [LTSSM]	Current state of the LTSSM. The encoding of the states is given in Appendix C.	0x0

Bits	SW	Name	Description	Reset
30	R	Remote Linkwidth Upconfigure Capability Status [R0]	A 1 in this field indicates that the remote node advertised Link Width Up configure Capability in the training sequences in the Configuration. Complete state when the link came up. A 0 indicates that the remote node did not set the Link Up configure bit.	0x0
31	R/W	Master Loopback Enable [MLE]	When the core is operating as a Root Port, setting this to 1 causes the LTSSM to initiate a loopback and become the loopback master. This bit is not used in the EndPoint Mode.	0x0

17.6.7.1.2 Physical Layer Configuration Register 1

Propname：Physical Layer Configuration Register 1
Address：@0x4
Description：This register contains additional configured parameters at the Physical Layer of the link, and command bits for various Physical Layer functions.

Bits	SW	Name	Description	Reset
7:0	R/W	Transmitted Link ID [TLI]	Link ID transmitted by the device in training sequences in the Root Port mode.	0x0
15:8	R/W	Transmitted FTS Count at 2.5 GT/s Speed [TFC1]	FTS count transmitted by the core in TS1/TS2 sequences during link training. This value must be set based on the time needed by the receiver on the other side to acquire sync while exiting from L0S state.	0x80
23:16	R/W	Transmitted FTS Count at 5 GT/s Speed [TFC2]	FTS count transmitted by the core in TS1/TS2 sequences during link training. This value must be set based on the time needed by the receiver to acquire sync while exiting from L0S state.	0x80
31:24	R/W	Transmitted FTS Count at 8 GT/s Speed [TFC3]	FTS count transmitted by the core in TS1/TS2 sequences during link training. This value must be set based on the time needed by the receiver to acquire sync while exiting from L0S state.	0x40

17.6.7.1.3 Data Link Layer Timer Configuration Register

Propname：Data Link Layer Timer Configuration Register
Address：@0x8
Description：This register defines the replay timeout values used by the DL receive and transmit sides of the link. It can be read or written via the local management APB bus.

Bits	SW	Name	Description	Reset
8:0	R/W	Transmit-Side Replay Timeout Adjustment [TSRT]	Additional transmit-side replay timer timeout interval. This 9-bit value is added as a signed 2's complement number to the internal replay timer timeout value computed by the core based on the PCI Express Specifications. This enables the user to make minor adjustments to the spec-defined replay timer settings. Its value is in multiples of 4ns (maximum = +1020 ns, minimum = -1024 ns).	0x0
15:9	R	Reserved [R9]	Reserved	0x0
24:16	R/W	Receive-Side ACK-NAK Replay Timeout Adjustment [RSART]	Additional receive side ACK-NAK timer timeout interval. This 9-bit value is added as a signed 2's complement number to the internal ACK-NAK timer timeout value computed by the core based on the PCI Express Specifications. This enables the user to make minor adjustments to the spec-defined replay timer settings. Its value is in multiples of 4 ns (maximum = +1020 ns, minimum = -1024 ns).	0x0
31:25	R	Reserved [R25]	Reserved	0x0

17.6.7.1.4 Receive Credit Limit Register 0 VC0

Propname：Receive Credit Limit Register 0 VC0

Address：@0xc

Description：This register contains the initial credit limits advertised by the core during th DL initialization. If the fields of this register are modified, the link must be re-trained to re-initialize the DL for the modified settings to take effect.

Bits	SW	Name	Description	Reset
11:0	R/W	Posted Payload Credit VC0 [PPC]	Posted payload credit limit advertised by the core for VC 0 (in units of 4 Dwords).	0xe0
19:12	R/W	Posted Header Credit VC0 [PHC]	Posted header credit limit advertised by the core for VC 0 (in number of packets).	0x20
31:20	R/W	Non-Posted Payload Credit VC0 [NPPC]	Non-Posted payload credit limit advertised by the core for VC 0 (in units of 4 Dwords).	0x20

17.6.7.1.5 Receive Credit Limit Register 1 VC0

Propname：Receive Credit Limit Register 1 VC0

Address：@0x10

Description：This register contains the initial credit limits advertised by the core during the DL initialization. If the fields of this register are modified, the link must be re-trained to re-initialize the DL for the modified settings to take effect.

Bits	SW	Name	Description	Reset
7:0	R/W	Non-Posted Header Credit Limit VC0 [NPHCL]	Non-Posted header credit limit advertised by the core for VC 0 (in number of packets).	0x20
19:8	R/W	Completion Payload Credit VC0 [CPC]	Completion payload credit limit advertised by the core for VC 0 (in units of 4 Dwords).	0x0
23:20	R	Reserved [R2]	Reserved	0x0
31:24	R/W	Completion Header Credit VC0 [CHC]	Completion header credit limit advertised by the core for VC 0 (in number of packets).	0x0

17.6.7.1.6 Transmit Credit Limit Register 0 VC0

Propname：Transmit Credit Limit Register 0 VC0
Address：@0x14
Description：This register contains the initial credit limits received from the opposite node during the DL initialization. It is a read-only register.

Bits	SW	Name	Description	Reset
11:0	R	Posted Payload Credit VC0 [PPC]	Posted payload credit limit received by the core for this link (in units of 4 Dwords).	0x0
19:12	R	Posted Header Credit VC0 [PHC]	Posted header credit limit received by the core for this link (in number of packets).	0x0
31:20	R	Non-Posted Payload Credit VC0 [NPPC]	Non-Posted payload credit limit received by the core for Link 0 (in units of 4 Dwords).	0x0

17.6.7.1.7 Transmit Credit Limit Register 1 VC0

Propname：Transmit Credit Limit Register 1 VC0
Address：@0x18
Description：This register contains the initial credit limits received from the opposite node during the DL initialization. It is a read-only register.

Bits	SW	Name	Description	Reset
7:0	R	Non-Posted Header Credit VC0 [NPHC]	Non-Posted header credit limit received by the core for VC 0 (in number of packets).	0x0
19:8	R	Completion Payload Credit VC0 [CPC]	Completion payload credit limit received by the core for VC 0 (in units of 4 Dwords).	0x0
23:20	R	Reserved [R3]	Reserved	0x0

Bits	SW	Name	Description	Reset
31:24	R	Completion Header Credit VC0 [CHC]	Completion header credit limit received by the core for VC 0 (in number of packets).	0x0

17.6.7.1.8 Transmit Credit Update Interval Configuration Register 0

Propname：Transmit Credit Update Interval Configuration Register 0
Address：@0x1c
Description：This register contains parameters that control how frequently the core sends a credit update to the opposite node.

Bits	SW	Name	Description	Reset
15:0	R/W	Minimum Posted Update Interval [MPUI]	Minimum credit update interval for posted transactions. The core follows this minimum interval between issuing posted credit updates on the link. This is to limit the bandwidth use of credit updates. If new credit becomes available in the receive FIFO since the last update was sent, the core will issue a new update only after this interval has elapsed since the last update. The value is in units of 4 ns. This field is re-written by the internal logic when the negotiated link width or link speed changes, to correspond to the default values defined in defines.h. The user may override this default value by writing into this register field. The value written will be lost on a change in the negotiated link width/speed.	16'd16
31:16	R/W	Minimum Non-Posted Update Interval [MNUI]	Minimum credit update interval for non-posted transactions. The core follows this minimum interval between issuing posted credit updates on the link. This is to limit the bandwidth use of credit updates. If new credit becomes available in the receive FIFO since the last update was sent, the core will issue a new update only after this interval has elapsed since the last update. The value is in units of 4 ns. This field is re-written by the internal logic when the negotiated link width or link speed changes, to correspond to the default values defined in defines.h. The user may override this default value by writing into this register field. The value written will be lost on a change in the negotiated link width/speed.	16'd16

17.6.7.1.9 Transmit Credit Update Interval Configuration Register 1

Propname：Transmit Credit Update Interval Configuration Register 1
Address：@0x20

Description：This register contains parameters that control how frequently the core sends a credit update to the opposite node.

Bits	SW	Name	Description	Reset
15:0	R/W	Minimum Completion Update Interval [CUI]	has elapsed since the last update. The value is in units of 4 ns. This parameter is not used when the Completion credit is infinity.	
31:16	R/W	Maximum Update Interval [MUI]	Maximum credit update interval for all transactions. If no new credit has become available since the last update, the core will repeat the last update after this interval. This is to recover from any losses of credit update packets. The value is in units of 4 ns. This field could be re-written by the internal logic when the negotiated link width or link speed changes, to correspond to the default values defined in defines.h. The user may override this default value by writing into this register field. The value written will be lost on a change in the negotiated link width/speed.	16'd938

17.6.7.1.10 L0S Timeout Limit Register

Propname：L0S Timeout Limit Register
Address：@0x24
Description：This register defines the timeout value for transitioning to the L0S power state. If the transmit side has been idle for this interval, the core will transmit the idle sequence on the link and transition the state of the link to L0S.

Bits	SW	Name	Description	Reset
15:0	R/W	L0S Timeout [LT]	Contains the timeout value (in units of 4ns) for transitioning to the L0S power state. Setting this parameter to 0 permanently disables the transition to the L0S power state.	0x02EE
31:16	R	Reserved [R4]	Reserved	0x0

17.6.7.1.11 Transmit TLP Count Register

Propname：Transmit TLP Count Register
Address：@0x28
Description：This register contains the number of Transaction-Layer packets transmitted by the core on the link since the register was last reset. This counter saturates on reaching a count of all 1's. Writing any value to this register causes it to be reset to 0.

Bits	SW	Name	Description	Reset
31:0	R/WOCLR	Transmit TLP Count [TTC]	Count of TLPs transmitted	0x0

17.6.7.1.12 Transmit TLP Payload Dword Count Register

Propname：Transmit TLP Payload Dword Count Register
Address：@0x2c
Description：This register contains the aggregate number of payload double-words

transmitted in Transaction-Layer Packets by the core on the link since the register was last reset. This counter saturates on reaching a count of all 1's. Writing any value to this register causes it to be reset to 0.

Bits	SW	Name	Description	Reset
31:0	R/WOCLR	Transmit TLP Payload Byte Count [TTPBC]	Count of TLPs payload Dwords transmitted	0x0

17.6.7.1.13 Receive TLP Count Register

Propname：Receive TLP Count Register
Address：@0x30
Description：This register contains the number of Transaction-Layer packets received by the core from the link since the register was last reset. This counter saturates on reaching a count of all 1's. Writing any value to this register causes it to be reset to 0.

Bits	SW	Name	Description	Reset
31:0	R/WOCLR	Receive TLP Count [RTC]	Count of TLPs received	0x0

17.6.7.1.14 Receive TLP Payload Dword Count Register

Propname：Receive TLP Payload Dword Count Register
Address：@0x34
Description：This register contains the aggregate number of payload double-words received in Transaction-Layer packets by the core from the link since the register was last reset. This counter saturates on reaching a count of all 1's. Writing any value to this register causes it to be reset to 0

Bits	SW	Name	Description	Reset
31:0	R/WOCLR	Receive TLP Payload Byte Count [RTPDC]	Count of TLP payload Dwords received	0x0

17.6.7.1.15 Completion Timeout Limit Register 0

Propname：Completion Timeout Limit Register 0
Address：@0x38
Description：This register contains the timeout value used to detect a completion timeout event for a request originated by the core from it master interface, when sub-range 1 is programmed in the Device Control 2 Register.

Bits	SW	Name	Description	Reset
23:0	R/W	Completion Timeout Limit [CTL]	Timeout limit for completion timers (in 4 ns cycles).	0xBEBC20
31:24	R	Reserved [R5]	Reserved	0x0

17.6.7.1.16 Completion Timeout Limit Register 1

Propname：Completion Timeout Limit Register 1
Address：@0x3c
Description：This register contains the timeout value used to detect a completion timeout event for a request originated by the core from its master interface, when sub-range 2 is programmed in the Device Control 2 Register.

Bits	SW	Name	Description	Reset
27:0	R/W	Completion Timeout Limit [CTL]	Timeout limit for completion timers (in 4 ns cycles).	28'd50_000_000
31:28	R	Reserved [R6]	Reserved	0x0

17.6.7.1.17 L1 State Re-Entry Delay Register

Propname：L1 State Re-Entry Delay Register
Address @0x40
Description This register specifies the time the core will wait before it re-enters the L1 state if its link partner transitions the link to L0 while all the Functions of the core are in D3 power state. The core will change the power state of the link from L0 to L1 if no activity is detected both on the transmit and receive sides before this interval, while all Functions are in D3 state and the link is in L0. Setting this register to 0 disables re-entry to L1 state if the link partner returns the link to L0 from L1 when all the Functions of the core are in D3 state. This register controls only the re-entry to L1. The initial transition to L1 always occurs when all of the Functions of the core are set to the D3 state.

Bits	SW	Name	Description	Reset
31:0	R/W	L1 Re-Entry Delay [L1RD]	Delay to re-enter L1 after no activity (in units of 4 ns).	0x0

17.6.7.1.18 Vendor ID Register

Propname：Vendor ID Register
Address：@0x44
Description：This register contains the Vendor ID and Subsystem Vendor ID that the device advertises during its enumeration of the PCI configuration space.

Bits	SW	Name	Description	Reset
15:0	R/W	Vendor ID [VID]	Vendor ID	16'h17cd
31:16	R/W	Subsystem Vendor ID [SVID]	Subsystem Vendor ID	16'h17cd

17.6.7.1.19 ASPM L1 Entry Timeout Delay Register

Propname：ASPM L1 Entry Timeout Delay Register
Address：@0x48
Description：This register defines the timeout value for transitioning to the L1 power state under Active State Power management. If the transmit side has been idle for this interval, the core will initiate a transition of its link to the L1 power state.

Bits	SW	Name	Description	Reset
19:0	R/W	L1 Timeout [L1T]	Contains the timeout value (in units of 4 ns) for transitioning to the L1 power state. Setting it to 0 permanently disables the transition to the L1 power state.	20'd1500
31:20	R	Reserved [R7]	Reserved	0x0

17.6.7.1.20 PME TurnOff Ack Delay Register

Propname：PME TurnOff Ack Delay Register
Address：@0x4c
Description：Defines the time interval between the core receiving a PME_Turn_Off message from the link and generating an ack for it.

Bits	SW	Name	Description	Reset
15:0	R/W	PME Turnoff Ack Delay [PTOAD]	Time in microseconds between the core receiving a PME_TurnOff message TLP and the core sending a PME_TO_Ack response to it. This field must be set to a non-zero value in order for the core to send a response. Setting this field to 0 suppresses the core's response to PME_TurnOff message, so that the client may transmit the PME_TO_Ack message through the master interface.	0x64
31:16	R	Reserved [R7]	Reserved	0x0

17.6.7.1.21 Linkwidth Control Register

Propname：Linkwidth Control Register
Address：@0x50
Description：When the core is configured as a Root Complex, this register can be used to retrain the link to a different width, without bringing the link down. This register is not to be used in the EndPoint mode.

Bits	SW	Name	Description	Reset
3:0	R/W	Target Lane Map [TLM]	This field contains the bitmap of the lanes to be included in forming the link during the re-training. If the target lane map includes lanes that were inactive when retraining is initiated, then both the core and its link partner must support the LinkWidth Upconfigure Capability to be able to activate those lanes. The user can check if the remote node has this capability by reading the Remote Link Upconfigure Capability Status bit in Physical Layer Configuration Register 0 after the link first came up.	4'b1111
15:4	R	Reserved [R0]	Reserved	0x0

Bits	SW	Name	Description	Reset
16	R/W	Retrain Link [RL]	Writing a 1 into this field results in the core re- training the link to change its width. When setting this bit to 1, the software must also set the target lane-map field to indicate the lanes it desires to be part of the link. The core will attempt to form a link with this set of lanes. The link formed at the end of the retraining may include all of these lanes (if both nodes agree on them during re-training), or the largest subset that both sides were able to activate. This bit is cleared by the internal logic of the core after the re-training has been completed and link has reached the L0 state. Software must wait for the bit to be clear before setting it again to change the link width.	0x0
31:17	R	Reserved [R1]	Reserved	0x0

SRIS Control Register
Propname：SRIS Control Register
Address：@0x74
Description：This register contains control bits to enable the SRIS operation in the PHY Layer

Bits	SW	Name	Description	Reset
0	R/W	SRIS Enable [SRISE]	Setting this bit enables SRIS mode in the PHY layer. This bit should be before link training begins by holding the LINK_TRAINING_ENABLE input to 1'b0.	0x0
31:1	R	Reserved [R31]	Reserved	0x0

17.6.7.1.22 Shadow register header log 0

Propname：Shadow register header log 0
Address：@0x100
Description：(no description)

Bits	SW	Name	Description	Reset
31:0	R/W	Shadow header log 0 [SHDW_HDR_LOG_0]	The value here will be reflected in the target function's header log register when f/w sets any bit In the shadow error register. If the header log is already set in the function's AER space, the value here may not get written and a header log overflow bit would get set. This register holds [31:0] value of the TLP header.	0x0

17.6.7.1.23 Shadow register header log 1

Propname：Shadow register header log 1
Address：@0x104

Description：(no description)

Bits	SW	Name	Description	Reset
31:0	R/W	Shadow header log 1 [SHDW_HDR_LOG_1]	The value here will be reflected in the target function's header log register when f/w sets any bit In the shadow error register. If the header log is already set in the function's AER space, the value here may not get written and a header log overflow bit would get set. This register holds [63:32] value of the TLP header.	0x0

17.6.7.1.24 Shadow register header log 2

Propname：Shadow register header log 2
Address：@0x108
Description：(no description)

Bits	SW	Name	Description	Reset
31:0	R/W	Shadow header log 2 [SHDW_HDR_LOG_2]	The value here will be reflected in the target function's header log register when f/w sets any bit In the shadow error register. If the header log is already set in the function's AER space, the value here may not get written and a header log overflow bit would get set. This register holds [95:64] value of the TLP header.	0x0

17.6.7.1.25 Shadow register header log 3

Propname：Shadow register header log 3
Address：@0x10c
Description：(no description)

Bits	SW	Name	Description	Reset
31:0	R/W	Shadow header log 3 [SHDW_HDR_LOG_3]	The value here will be reflected in the target function's header log register when f/w sets any bit In the shadow error register. If the header log is already set in the function's AER space, the value here may not get written and a header log overflow bit would get set. This register holds [127:96] value of the TLP header.	0x0

17.6.7.1.26 Shadow register function number

Propname：Shadow register function number.
Address：@0x110
Description：(no description)

Bits	SW	Name	Description	Reset
7:0	R/W	Shadow register target function number [SHDW_FUNC_NUM	The value here will be the target function number when f/w sets any bit in the shadow error register.	0x0

Bits	SW	Name	Description	Reset
31:8	R	Reserved [R0]	Reserved	0x0

17.6.7.1.27 Shadow Register UR Error

Propname：Shadow Register UR Error

Address：@0x114

Description：Shadow register to create UR error via local f/w. Please make sure this register is written to last, after writing to all the header log and function number registers. A write to this register with any bits set, will internally create a single cycle pulse with the corresponding error type and the header log will reflect the value written in the shadow header log registers.

Bits	SW	Name	Description	Reset
0	W	Posted UR Error [P_UR_ERR]	If this bit is set, the corresponding posted UR error bits will be set in the AER and device status registers of the target function.	0x0
1	W	Non Posted Error [NP_UR_ERR]	If this bit is set, the corresponding non-posted UR error bits will be set in the AER and device status registers of the target function.	0x0
31:2	R	Reserved [R0]	Reserved	0x0

17.6.7.1.28 Negotiated Lane Map Register

Propname：Negotiated Lane Map Register

Address：@0x200

Description：This register contains a map of the active lanes used by the core to form the link during link training. It also contains a bit to indicate whether the core reversed the lane number on its lanes during link training.

Bits	SW	Name	Description	Reset
3:0	R	Negotiated Lane Map [NLM]	Bit i of this field is set to 1 at the end of link training if Lane i is part of the PCIe link. The value of this field is valid only when the link is in L0 or L0s states.	0x0
15:4	R	Reserved [R70]	Reserved	0x0
16	R	Lane Reversal Status [LRS]	This bit set by the core at the end of link training if the LTSSM had to reverse the lane numbers to form the link.	0x0
31:17	R	Reserved [R71]	Reserved	0x0

17.6.7.1.29 Receive FTS Count Register

Propname：Receive FTS Count Register

Address：@0x204

Description：This register contains the FTS count values received from the link partner during link training for use at the 5 GT/s. These values determine the number of Fast Training Sequences transmitted by the core when it exits the L0s link power state.

Bits	SW	Name	Description	Reset
7:0	R	Received FTS Count for 5GT/s Speed [RFC5S]	FTS count received from the other side during link training for use at the 5 GT/s link speed. The core transmits this many FTS sequences while exiting the L0S state, when operating at the 5 GT/s speed.	0x0
15:8	R	Reserved [R7]	Reserved	0x0
31:16	R	Reserved [R72]	Reserved	0x0

17.6.7.1.30 Debug Mux Control Register

Propname：Debug Mux Control Register
Address：@0x208
Description：(no description)

Bits	SW	Name	Description	Reset
3:0	R/W	Mux Select [MS]	Bits 3:2 select the module and bits 1:0 select the group of signals within the module that are driven on the debug bus. The assignments of signals on the debug outputs of the core are given in Appendix B.	0x0
8:4	R	Reserved [R8]	(no description)	0x0
9	R	Reserved [R99]	Reserved	0x0
10	R	Reserved [R1010]	Reserved	0x0
11	R/W	Disable Client TX MUX arbitration [R1111]	When this bit is 1, Disable Client TX MUX Completion and PNP request arbitration, logic added to prevent PNP requests from starving when completions are present	0x0
12	R	Reserved [R1212]	(no description)	0x0
13	R	Reserved [R1313]	(no description)	0x0
14	R/W	Disable Set Slot Power Limit Message [DSSPLM]	Disable sending Set Slot Power Limit Message if the Slot Capability register is configured	0x0
15	R/W	Force Disable Scrambling [FDS]	Disable Scrambling/Descrambling in Gen1/Gen2.	0x0
16	R/W	Enable AXI Bridge Write Priority [AWRPRI]	When this bit is 1, the AXI bridge places a write request on the HAL Master interface in preference over a read request if both AXI write and AXI read requests are available to be asserted on the same clock cycle.	0x0
17	R	Reserved [R8B]	(no description)	0x0
19:18	R	Reserved [R1918]	(no description)	0x0

Bits	SW	Name	Description	Reset
20	R/W	Disable checking of invalid message codes [DCIVMC]	When this bit is 1, the core will not check for invalid message codes. This bit should normally set to 0, as the invalid message code checking is mandatory in the PCIe 3.0 specifications.	0x0
22:21	R	Reserved [R21]	Reserved	0x0
23	R/W	Disable Link Upconfigure Capability [DLUC]	The user may set this bit to turn off the link upconfigure capability of the core. Setting this bit prevents the core from advertising the link upconfigure capability in training sequences transmitted in the Configuration.Complete state.	0x0
24	R/W	Enable Fast Link Training [EFLT]	This bit is provided to shorten the link training time to facilitate fast simulation of the design, especially at the gate level. Enabling this bit has the following effects: 1. The 1ms, 2ms, 12ms, 24ms, 32ms and 48ms timeout intervals in the LTSSM are shortened by a factor of 500. 2. In the Polling.Active state of the LTSSM, only 16 training sequences are required to be transmitted (Instead of 1024) to make the transition to the Configuration state. This bit should not be set during normal operation of the core.	0x0
25	R/W	Enable Slot Power Capture [ESPC]	When this bit is set to 1, the core will capture the Slot Power Limit Value and Slot Power Limit Scale parameters from a Set_Slot_Power_Limit message received in the Device Capabilities Register. When this bit is 0, the capture is disabled. This bit is valid only when the core is configured as an EndPoint. It has no effect when the core is a Root Complex.	0x0
26	R	Reserved [R26]	Reserved	0x0
27	R	Reserved [R27]	Reserved	0x0
28	R/W	Disable Electrical Idle Infer in L0 State [DEI]	Setting this bit to 1 disables the inferring of electrical idle in the L0 state. Electrical idle is inferred when no flow control updates and no SKP sequences are received within an interval of 128 us. This bit should not be set during normal operation, but is useful for testing.	0x0

Bits	SW	Name	Description	Reset
29	R/W	Disable Flow Control Update Timeout [DFCUT]	When this bit is 0, the core will time out and re-train the link when no Flow Control Update DLLPs are received from the link within an interval of 128 us. Setting this bit to 1 disables this timeout. When the advertised receive credit of the link partner is infinity for the header and payload of all credit types, this timeout is always suppressed. The setting of this bit has no effect in this case. This bit should not be set during normal operation, but is useful for testing.	0x0
30	R/W	Disable Ordering Checks [DOC]	Setting this bit to 1 disables the ordering check in the core between Completions and Posted requests received from the link.	0x0
31	R/W	Enable Function-Specific Reporting of Type-1 Configuration Accesses [EFSRTCA]	Setting this bit to 0 causes all the enabled Functions to report an error when a Type-1 configuration access is received by the core, targeted at any Function. Setting it to 1 limits the error reporting to the type-0 Function whose number matches with the Function number specified in the request. If the Function number in the request refers to an unimplemented or disabled Function, all enabled Functions report the error regardless of the setting of this bit.	0x01

17.6.7.1.31 Local Error and Status Register

Propname：Local Error and Status Register
Address：@0x20c
Description：This register contains the status of the various events, errors and abnormal conditions in the core. Any of the status bits can be reset by writing a 1 into the bit position. This register does not capture any errors signaled by remote devices using PCIe error messages when the core is operating in the RC mode. Unless masked by the setting of the Local Interrupt Mask Register, the occurrence of any of these conditions causes the core to activate the LOCAL_INTERRUPT output.

Bits	SW	Name	Description	Reset
0	R/WOCLR	PNP RX FIFO Parity Error [PRFPE]	Parity error detected while reading from the PNP Receive FIFO RAM.	0x0
1	R/WOCLR	Completion RX FIFO Parity Error [CRFPE]	Parity error detected while reading from the Completion Receive FIFO RAM.	0x0
2	R/WOCLR	Replay RAM Parity Error [RRPE]	Parity error detected while reading from Replay Buffer RAM.	0x0

Bits	SW	Name	Description	Reset
3	R/WOCLR	PNP RX FIFO Overflow [PRFO]	Overflow occurred in the PNP Receive FIFO.	0x0
4	R/WOCLR	Completion RX FIFO Overflow [CRFO]	Overflow occurred in the Completion Receive FIFO.	0x0
5	R/WOCLR	Replay Timeout [RT]	Replay timer timed out	0x0
6	R/WOCLR	Replay Timer Rollover [RTR]	Replay timer rolled over after 4 transmissions of the same TLP.	0x0
7	R/WOCLR	Phy Error [PE]	Phy error detected on receive side.	0x0
8	R/WOCLR	Malformed TLP Received [MTR]	Malformed TLP received from the link.	0x0
9	R/WOCLR	Unexpected Completion Received [UCR]	Unexpected Completion received from the link.	0x0
10	R/WOCLR	Flow Control Error [FCE]	An error was observed in the flow control advertisements from the other side.	0x0
11	R/WOCLR	Completion Timeout [CT]	A request timed out waiting for completion.	0x0
12	R	Reserved [R12]	Reserved	0x0
16:13	R	Reserved [R13]	Reserved	0x0
17	R	Reserved [R17]	Reserved	0x0
18	R/WOCLR	Unmapped TC [UTC]	Unmapped TC error	0x0
19	R/WOCLR	MSI Mask Value Change [MMVC]	This status bit is set whenever the MSI mask register value in the MSI capability register changes value in ANY of the functions in the controller	0x0
20	R	Reserved [R22]	Reserved	0x0
31:21	R	Reserved [R9]	Reserved	0x0

17.6.7.1.32 Local Interrupt Mask Register

Propname：Local Interrupt Mask Register
Address：@0x210
Description：This register contains a mask bit for each interrupting condition. Setting the bit to 1 prevents the corresponding condition in the Local Error Status Register from activating the LOCAL_INTERRUPT output.

Bits	SW	Name	Description	Reset
0	R/W	PNP RX FIFO Parity Error [PRFPE]	Parity error detected while reading from the PNP Receive FIFO RAM.	1'b1
1	R/W	Completion RX FIFO Parity Error [CRFPE]	Parity error detected while reading from the Completion Receive FIFO RAM.	1'b1
2	R/W	Replay RAM Parity Error [RRPE]	Parity error detected while reading from Replay Buffer RAM.	1'b1
3	R/W	PNP RX FIFO Overflow [PRFO]	Overflow occurred in the PNP Receive FIFO.	1'b1
4	R/W	Completion RX FIFO Overflow [CRFO]	Overflow occurred in the Completion Receive FIFO.	1'b1
5	R/W	Replay Timeout [RT]	Replay timer timed out	1'b1
6	R/W	Replay Timer Rollover [RTR]	Replay timer rolled over after 4 transmissions of the same TLP.	1'b1
7	R/W	Phy Error [PE]	Phy error detected on receive side.	1'b1
8	R/W	Malformed TLP Received [MTR]	Malformed TLP received from the link.	1'b1
9	R/W	Unexpected Completion Received [UCR]	Unexpected Completion received from the link.	1'b1
10	R/W	Flow Control Error [FCE]	An error was observed in the flow control advertisements from the other side.	1'b1
11	R/W	Completion Timeout [CT]	A request timed out waiting for completion.	1'b1
12	R	Reserved [R12]	Reserved	1'b0
16:13	R	Reserved [R13]	Reserved	0x0
17	R	Reserved [R17]	Reserved	0x0
18	R/W	Unmapped TC [UTC]	Unmapped TC error	0x1
19	R/W	MSI Mask Value Change [MMVC]	MSI mask register value in the MSI capability register changes value in ANY of the functions in the controller	0x1

Bits	SW	Name	Description	Reset
20	R	Reserved [R45]	Reserved	1'b0
31:21	R	Reserved [R9]	Reserved	0x0

17.6.7.1.33 LCRC Error Count Register

Propname：LCRC Error Count Register Address @0x214

Description：This register contains the count of the number of TLPs received by the core with LCRC errors in them. This is a 16-bit saturating counter that can be reset to 0 by writing all 1's into it.

Bits	SW	Name	Description	Reset
15:0	R/WOCLR	LCRC Eror Count [LEC]	Number of TLPs received with LCRC errors.	0x0
31:16	R	Reserved [R11]	Reserved	0x0

17.6.7.1.34 ECC Correctable Error Count Register

Propname ECC Correctable Error Count Register Address @0x218

Description This register contains the count of the number of ECC errors detected and corrected during reads from the three external RAMs.

Bits	SW	Name	Description	Reset
7:0	R/WOCLR	PNP FIFO RAM Correctable Error Count [PFRCER]	Number of correctable errors detected while reading from the PNP FIFO RAM. This is an 8-bit saturating counter that can be cleared by writing all 1's into it.	0x0
15:8	R/WOCLR	SC FIFO RAM Correctable Error Count [SFRCER]	Number of correctable errors detected while reading from the SC FIFO RAM. This is an 8-bit saturating counter that can be cleared by writing all 1's into it.	0x0
23:16	R/WOCLR	Replay RAM Correctable Error Count [RRCER]	Number of correctable errors detected while reading from the Replay Buffer RAM. This is an 8- bit saturating counter that can be cleared by writing all 1's into it.	0x0
31:24	R/WOCLR	TPH ST RAM Correctable Error Count [R12]	Number of correctable errors detected while reading from the TPH Steering Tag RAM. This is an 8-bit saturating counter that can be cleared by writing all 1s into it.	0x0

17.6.7.1.35 LTR Snoop/No-Snoop Latency Register

Propname：LTR Snoop/No-Snoop Latency Register

Address：@0x21c

Description：This register contains the Snoop and No-Snoop Latency parameters used by the core when sending Latency Tolerance Reporting (LTR) Message. When the core is configured in the EndPoint mode, client software can program these fields to the desired latency settings and then set the Send LTR Message bit in the LTR Message Generation Control Register to send an LTR message to the Root Complex. The fields in this register should not be changed when the Send LTR Message bit in the LTR Message Generation

Control Register is 1, which indicates that an LTR message is pending to be transmitted.

Bits	SW	Name	Description	Reset
9:0	R/W	No-Snoop Latency Value [NSLV]	The client software must program this field with the value to be sent in the No-Snoop Latency Value field of the LTR message.	0x0
12:10	R/W	No-Snoop Latency Scale [NSLS]	The client software must program this field with the value to be sent in the No-Snoop Latency Scale field of the LTR message.	0x0
14:13	R	Reserved [R12]	(no description)	0x0
15	R/W	No-Snoop Latency Requirement [NSLR]	The client software must set this bit to 1 to set the No-Snoop Latency Requirement bit in the LTR message to be sent.	0x0
25:16	R/W	Snoop Latency Value [SLV]	The client software must program this field with the value to be sent in the Snoop Latency Value field of the LTR message.	0x0
28:26	R/W	Snoop Latency Scale [SLS]	The client software must program this field with the value to be sent in the Snoop Latency Scale field of the LTR message.	0x0
30:29	R	Reserved [R13]	Reserved	0x0
31	R/W	Snoop Latency [SL]	The client software must set this bit to 1 to set the Snoop Latency Requirement bit in the LTR message to be sent.	0x0

17.6.7.1.36 LTR Message Generation Control Register

Propname：LTR Message Generation Control Register

Address：@0x220

Description：This register contains fields for the generation of Latency Tolerance Reporting (LTR) Messages. This register is to be used only when the core is configured in the EndPoint mode.

Bits	SW	Name	Description	Reset
9:0	R/W	Minimum LTR Interval [MLI]	This field specifies the minimum spacing between LTR messages transmitted by the core in units of microseconds. The PCI Express Specifications recommend sending no more than two LTR messages within a 500 microsecond interval. The core will wait for the minimum delay specified by this field after sending an LTR message, before transmitting a new LTR message.	0xFA

Bits	SW	Name	Description	Reset
10	R	Send LTR Message [SLM]	Setting this bit causes the core to transmit an LTR message with the parameters specified in the LTR Snoop/No-Snoop Latency Register (Section 8.4.2.9). This bit is cleared by the core on transmitting the LTR message, and stays set until then. Client software must read this register and verify that this bit is 0 before setting it again to send a new message. This field becomes writable when LTR mechanism is enabled in device control-2 register.	0x0
11	R/W	Transmit Message on LTR Mechanism Enable Transition [TMLMET]	When this bit is set to 1, the core will automatically transmit an LTR message whenever the LTR Mechanism Enable bit in the Device Control 2 Register changes from 0 to 1, with the parameters specified in the LTR Snoop/No-Snoop Latency Register. When this bit is 1, the core will also transmit an LTR message whenever the LTR Mechanism Enable bit is cleared, if the following conditions are both true: 1. The core sent at least one LTR message since the LTR Mechanism Enable bit was last set. 2. The most recent LTR message transmitted by the core had as least one of the Requirement bits set. The core will set the Requirement bits in this LTR message to 0. When this bit 11 is 0, the core will not, by itself, send any LTR messages in response to state changes of the LTR Mechanism Enable bit. Client logic may monitor the state of the LTR_MECHANISM_ ENABLE output of the core and transmit LTR messages through the master interface, in response to its state changes.	0x1

Bits	SW	Name	Description	Reset
12	R/W	Transmit Message on Function Power State Change [TMFPSC]	When this bit is set to 1, the core will automatically transmit an LTR message when all the Functions in the core have transitioned to a non-D0 power state, provided that the following conditions are both true: 1. The core sent at least one LTR message since the Data Link layer last transitioned from down to up state. 2. The most recent LTR message transmitted by the core had as least one of the Requirement bits set. The core will set the Requirement bits in this LTR message to 0. When this bit 12 is 0, the core will not, by itself, send any LTR messages in response to Function Power State changes. Client logic may monitor the FUNCTION_POWER_STATE outputs of the core and transmit LTR messages through the master interface, in response to changes in their states.	0x1
31:13	R	RSVD	RSVD	19'h00000

17.6.7.1.37 PME Service Timeout Delay Register

Propname：PME Service Timeout Delay Register
Address：@0x224
Description：This register stores the timeout delay parameter for the service timeout mechanism associated with the generation of PM_PME messages. In the EndPoint mode, the core will retransmit a PM_PME message after the expiration of this delay, if the Root Complex did not clear the PME Status bit in the Power Management Control and Status Register. This register is not used when the core is configured as Root Complex.

Bits	SW	Name	Description	Reset
19:0	R/W	PME Service Timeout Delay [PSTD]	Specifies the timeout delay for retransmission of PM_PME messages. The value is in units of microseconds. The actual time elapsed has a +1 microseconds tolerance from the value programmed.	0x186A0
20	R/W	Disable PME message on PM Status [DPMOPS]	When this bit is set, core will not automatically send a PME message, when PM Status bit in PMCSR register is set	0x0
31:21	R	Reserved [R21]	Reserved	0x0

17.6.7.1.38 Root Port Requestor ID Register

Propname：Root Port Requestor ID Register
Address：@0x228
Description：When the core is configured as Root Complex, this ID will be used for all internally generated messages.

Bits	SW	Name	Description	Reset
15:0	R/W	Root Port Requestor ID [RPRI]	RID (bus, device and function numbers) for all TLPs internally generated by Root Port	0x0
31:16	R	Reserved [R0]	Reserved	0x0

17.6.7.1.39 End Point Bus and Device Number Register

Propname：End Point Bus and Device Number Register
Address：@0x22c
Description：When the core is configured as End Point, this register holds the Bus and Device number captured for Function 0

Bits	SW	Name	Description	Reset
4:0	R	Device Number [EPDN]	Device Number captured by Function 0 in End Point mode	0x0
7:5	R	Reserved [R5]	Reserved	0x0
15:8	R	Bus Number [EPBN]	Bus Number captured by Function 0 in End Point mode	0x0
31:16	R	Reserved [R16]	Reserved	0x0

17.6.7.1.40 Physical Function BAR Configuration Register 0

Propname：Physical Function BAR Configuration Register 0
Address：@0x240
Description：This register specifies the configuration of the BARs associated with the Physical Function 0

Bits	SW	Name	Description	Reset
4:0	R/W	BAR 0 Aperture [BAR0A]	Specifies the aperture of the 32-bit BAR 0 or 64bit BAR0-1. For 32-bit BAR 0, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512B, 00011 = 1 KB, 00100 = 2 KB, 00101 = 4 KB,00110 = 8 KB, 00111 = 16 KB, 01000 = 32 KB,01001 = 64 KB, 01010 = 128 KB, 01011 = 256KB, 01100 = 512 KB, 01101 = 1 MB, 01110 =2 MB, 01111 = 4 MB, 10000 = 8 MB, 10001 =16 MB, 10010 = 32 MB, 10011 = 64 MB, 10100= 128 MB, 10101 = 256 MB, 10110 = 512 MB,10111 = 1 GB, 11000 = 2 GB For 64-bit BAR0-1, the valid encodings are: 00000 = 128 B, 00001 =256 B, 00010 = 512 B, 00011 = 1 KB, 00100 =2 KB, 00101 = 4 KB, 00110 = 8 KB, 00111 = 16KB, 01000 = 32 KB, 01001 = 64 KB, 01010 = 128 KB, 01011 = 256 KB, 01100 = 512 KB, 01101 =1 MB, 01110 = 2 MB, 01111 = 4 MB, 10000 = 8MB, 10001 = 16 MB, 10010 = 32 MB, 10011 = 64MB, 10100 = 128 MB, 10101 = 256 MB, 10110= 512 MB, 10111 = 1 GB, 11000 = 2 GB, 11001= 4 GB, 11010 = 8 GB, 11011 = 16 GB, 11100 =32 GB, 11101 = 64 GB, 11110 = 128 GB, 11111 =256 GB	5'hf
7:5	R/W	BAR 0 Control [BAR0C]	Specifies the configuration of BAR0. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	3'h6

Bits	SW	Name	Description	Reset
12:8	R/W	BAR 1 Aperture [BAR1A]	Specifies the aperture of the BAR 1 when it is configured as a 32-bit BAR. For 32-bit BAR 1, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512 B, 00011 = 1 KB, 00100 =2 KB, 00101 = 4 KB, 00110 = 8 KB, 00111 = 16KB, 01000 = 32 KB, 01001 = 64 KB, 01010 = 128KB, 01011 = 256 KB, 01100 = 512 KB, 01101 =1 MB, 01110 = 2 MB, 01111 = 4 MB, 10000 = 8MB, 10001 = 16 MB, 10010 = 32 MB, 10011 = 64MB, 10100 = 128 MB, 10101 = 256 MB, 10110 =512 MB, 10111 = 1 GB, 11000 = 2 GB	5'hf
15:13	R/W	BAR 1 Control [BAR1C]	Specifies the configuration of BAR1. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	3'h4
20:16	R/W	BAR 2 Aperture [BAR2A]	Specifies the aperture of the 32-bit BAR 2 or 64bit BAR2-3. For 32-bit BAR 2, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512B, 00011 = 1 KB, 00100 = 2 KB, 00101 = 4 KB,00110 = 8 KB, 00111 = 16 KB, 01000 = 32 KB,01001 = 64 KB, 01010 = 128 KB, 01011 = 256KB, 01100 = 512 KB, 01101 = 1 MB, 01110 =2 MB, 01111 = 4 MB, 10000 = 8 MB, 10001 =16 MB, 10010 = 32 MB, 10011 = 64 MB, 10100= 128 MB, 10101 = 256 MB, 10110 = 512 MB,10111 = 1 GB, 11000 = 2 GB For 64-bit BAR2-3, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512 B, 00011 = 1 KB, 00100 =2 KB, 00101 = 4 KB, 00110 = 8 KB, 00111 = 16KB, 01000 = 32 KB, 01001 = 64 KB, 01010 = 128 KB, 01011 = 256 KB, 01100 = 512 KB, 01101 =1 MB, 01110 = 2 MB, 01111 = 4 MB, 10000 = 8MB, 10001 = 16 MB, 10010 = 32 MB, 10011 = 64MB, 10100 = 128 MB, 10101 = 256 MB, 10110= 512 MB, 10111 = 1 GB, 11000 = 2 GB, 11001= 4 GB, 11010 = 8 GB, 11011 = 16 GB, 11100 =32 GB, 11101 = 64 GB, 11110 = 128 GB, 11111 =256 GB	5'hf

Bits	SW	Name	Description	Reset
23:21	R/W	BAR 2 Control [BAR2C]	Specifies the configuration of BAR2. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	3'h6
28:24	R/W	BAR 3 Aperture [BAR3A]	Specifies the aperture of the BAR 3 when it is configured as a 32-bit BAR. For 32-bit BAR 3, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512 B, 00011 = 1 KB, 00100 =2 KB, 00101 = 4 KB, 00110 = 8 KB, 00111 = 16KB, 01000 = 32 KB, 01001 = 64 KB, 01010 = 128KB, 01011 = 256 KB, 01100 = 512 KB, 01101 =1 MB, 01110 = 2 MB, 01111 = 4 MB, 10000 = 8MB, 10001 = 16 MB, 10010 = 32 MB, 10011 = 64MB, 10100 = 128 MB, 10101 = 256 MB, 10110 =512 MB, 10111 = 1 GB, 11000 = 2 GB	5'hf
31:29	R/W	BAR 3 Control [BAR3C]	Specifies the configuration of BAR3. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	3'h4

17.6.7.1.41 Physical Function BAR Configuration Register 1

Propname：Physical Function BAR Configuration Register 1
Address：@0x244
Description：This register specifies the configuration of the BARs associated with the Physical Function.

Bits	SW	Name	Description	Reset
4:0	R/W	BAR 4 Aperture [BAR4A]	Specifies the aperture of the 32-bit BAR 4 or 64bit BAR4-5. For 32-bit BAR 4, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512B, 00011 = 1 KB, 00100 = 2 KB, 00101 = 4 KB,00110 = 8 KB, 00111 = 16 KB, 01000 = 32 KB,01001 = 64 KB, 01010 = 128 KB, 01011 = 256KB, 01100 = 512 KB, 01101 = 1 MB, 01110 =2 MB, 01111 = 4 MB, 10000 = 8 MB, 10001 =16 MB, 10010 = 32 MB, 10011 = 64 MB, 10100= 128 MB, 10101 = 256 MB, 10110 = 512 MB,10111 = 1 GB, 11000 = 2 GB For64-bit BAR4-5, the valid encodings are: 00000 = 128 B, 00001 =256 B, 00010 = 512 B, 00011 = 1 KB, 00100 =2 KB, 00101 = 4 KB, 00110 = 8 KB, 00111 = 16KB, 01000 = 32 KB, 01001 = 64 KB, 01010 = 128KB, 01011 = 256 KB, 01100 = 512 KB, 01101 =1 MB, 01110 = 2 MB, 01111 = 4 MB, 10000 = 8MB, 10001 = 16 MB, 10010 = 32 MB, 10011 = 64MB, 10100 = 128 MB, 10101 = 256 MB, 10110= 512 MB, 10111 = 1 GB, 11000 = 2 GB, 11001= 4 GB, 11010 = 8 GB, 11011 = 16 GB, 11100 =32 GB, 11101 = 64 GB, 11110 = 128 GB, 11111 = 256 GB	
7:5	R/W	BAR 4 Control [BAR4C]	Specifies the configuration of BAR4. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	3'h6

Bits	SW	Name	Description	Reset
12:8	R/W	BAR 5 Aperture [BAR5A]	Specifies the aperture of the BAR 5 when it is configured as a 32-bit BAR. For 32-bit BAR 5, the valid encodings are: 00000 = 128 B, 00001 = 256 B, 00010 = 512 B, 00011 = 1 KB, 00100 =2 KB, 00101 = 4 KB, 00110 = 8 KB, 00111 = 16KB, 01000 = 32 KB, 01001 = 64 KB, 01010 = 128KB, 01011 = 256 KB, 01100 = 512 KB, 01101 =1 MB, 01110 = 2 MB, 01111 = 4 MB, 10000 = 8MB, 10001 = 16 MB, 10010 = 32 MB, 10011 = 64MB, 10100 = 128 MB, 10101 = 256 MB, 10110 =512 MB, 10111 = 1 GB, 11000 = 2 GB	5'hf
15:13	R/W	BAR 5 Control [BAR5C]	Specifies the configuration of BAR5. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	3'h4
23:16	R	Reserved [R16]	Reserved	0x0
30:24	R	Reserved [R24]	Reserved	0x0
31	R/W	Enable Resizable BAR Capability [ERBC]	Setting this bit to 1 enables the Resizable BAR Capability in the PCI Express Configuration Space of the associated Function. When the Resizable BAR Capability is enabled, the apertures of the memory BARs of the corresponding Function are no longer selected by the fields in this register, but by the setting of the registers in the Resizable BAR Capability Structure.	0x0

17.6.7.1.42 Virtual Function BAR Configuration Register 0

Propname：Virtual Function BAR Configuration Register 0
Address：@0x280
Description：This register specifies the configuration of the VF BARs associated with the Physical Function 0

Bits	SW	Name	Description	Reset
4:0	R/W	VF BAR 0 Aperture [VFBAR0A]	Specifies the aperture of the 32-bit VF BAR 0 or 64bit VF BAR0-1. The encodings are: 00000 = 128 Bytes, 0001 = 256 Bytes, 0010 = 512 Bytes, 0011 = 1 Kbytes, 00100 = 2 Kbytes, 00101 = 4Kbytes, 00110 = 8 Kbytes, 00111 = 16 Kbytes, 01000 = 32 Kbytes, 01001 = 64 Kbytes, 01010= 128 Kbytes, 01011 = 256 Kbytes, 01100 = 512 Kbytes, 01101 = 1 Mbyte, 01110 = 2 Mbytes, 01111 = 4 Mbytes, 10000 = 8 Mbytes, 10001 = 16 Mbytes, 10010 = 32 Mbytes, 10011 = 64 Mbytes, 10100 = 128 Mbytes, 10101 = 256 Mbytes,10110 = 512 Mbytes, 10111 = 1 Gbyte, 11000 =2 Gbytes, 11001 = 4 Gbytes, 11010 = 8 Gbytes,11011 = 16 Gbytes, 11100 = 32 Gbytes, 11101= 64 Gbytes, 11110 = 128 Gbytes, 11111 = 256 Gbytes	5'hf
7:5	R/W	VF BAR 0 Control [VFBAR0C]	Specifies the configuration of VF BAR0. The various encodings are: 000: Disabled 001-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	3'h6
12:8	R/W	VF BAR 1 Aperture [VFBAR1A]	Specifies the aperture of the VF BAR 1 when it is configured as a 32-bit BAR. The encodings are: 00000 = 128 Bytes, 0001 = 256 Bytes, 0010 =512 Bytes, 0011 = 1 Kbytes, 00100 = 2 Kbytes, 00101 = 4 Kbytes, 00110 = 8 Kbytes, 00111 = 16Kbytes, 01000 = 32 Kbytes, 01001 = 64 Kbytes,01010 = 128 Kbytes, 01011 = 256 Kbytes, 01100 = 512 Kbytes, 01101 = 1 Mbyte, 01110 = 2 Mbytes, 01111 = 4 Mbytes, 10000 = 8 Mbytes, 10001 = 16Mbytes, 10010 = 32 Mbytes, 10011 = 64 Mbytes,10100 = 128 Mbytes, 10101 = 256 Mbytes,10110= 512 Mbytes, 10111 = 1 Gbyte, 11000 = 2 Gbytes	5'hf
15:13	R/W	VF BAR 1 Control [VFBAR1C]	Specifies the configuration of VF BAR1. The various encodings are: 000: Disabled 001-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	3'h4

Bits	SW	Name	Description	Reset
20:16	R/W	VF BAR 2 Aperture [VFBAR2A]	Specifies the aperture of the 32-bit VF BAR 2 or 64bit VF BAR2-3. The encodings are: 00000 = 128 Bytes, 0001 = 256 Bytes, 0010 = 512 Bytes, 0011 = 1 Kbytes, 00100 = 2 Kbytes, 00101 = 4Kbytes, 00110 = 8 Kbytes, 00111 = 16 Kbytes,01000 = 32 Kbytes, 01001 = 64 Kbytes, 01010= 128 Kbytes, 01011 = 256 Kbytes, 01100 = 512Kbytes, 01101 = 1 Mbyte, 01110 = 2 Mbytes, 01111 = 4 Mbytes, 10000 = 8 Mbytes, 10001 = 16 Mbytes, 10010 = 32 Mbytes, 10011 = 64 Mbytes, 10100 = 128 Mbytes, 10101 = 256 Mbytes, 10110 = 512 Mbytes, 10111 = 1 Gbyte, 11000 =2 Gbytes, 11001 = 4 Gbytes, 11010 = 8 Gbytes, 11011 = 16 Gbytes, 11100 = 32 Gbytes, 11101= 64 Gbytes, 11110 = 128 Gbytes, 11111 = 256 Gbytes	5'hf
23:21	R/W	VF BAR 2 Control [VFBAR2C]	Specifies the configuration of VF BAR2. The various encodings are: 000: Disabled 001-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	3'h6
28:24	R/W	VF BAR 3 Aperture [VFBAR3A]	Specifies the aperture of the VF BAR 3 when it is configured as a 32-bit BAR. The encodings are: 00000 = 128 Bytes, 0001 = 256 Bytes, 0010 =512 Bytes, 0011 = 1 Kbytes, 00100 = 2 Kbytes, 00101 = 4 Kbytes, 00110 = 8 Kbytes, 00111 = 16Kbytes, 01000 = 32 Kbytes, 01001 = 64 Kbytes, 01010 = 128 Kbytes, 01011 = 256 Kbytes, 01100 = 512 Kbytes, 01101 = 1 Mbyte, 01110 = 2 Mbytes, 01111 = 4 Mbytes, 10000 = 8 Mbytes, 10001 = 16 Mbytes, 10010 = 32 Mbytes, 10011 = 64 Mbytes, 10100 = 128 Mbytes, 10101 = 256 Mbytes, 10110 = 512 Mbytes, 10111 = 1 Gbyte, 11000 = 2 Gbytes	5'hf
31:29	R/W	VF BAR 3 Control [VFBAR3C]	Specifies the configuration of VF BAR3. The various encodings are: 000: Disabled 001-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	3'h4

17.6.7.1.43 Virtual Function BAR Configuration Register 1

Propname：Virtual Function BAR Configuration Register 1
Address：@0x284
Description：This register specifies the configuration of the VF BARs associated with the Physical Function.

Bits	SW	Name	Description	Reset
4:0	R/W	VF BAR 4 Aperture [VFBAR4A]	Specifies the aperture of the 32-bit VF BAR 4 or 64bit VF BAR4-5. The encodings are: 00000 = 128 Bytes, 0001 = 256 Bytes, 0010 = 512 Bytes, 0011 = 1 Kbytes, 00100 = 2 Kbytes, 00101 = 4Kbytes, 00110 = 8 Kbytes, 00111 = 16 Kbytes, 01000 = 32 Kbytes, 01001 = 64 Kbytes, 01010= 128 Kbytes, 01011 = 256 Kbytes, 01100 = 512 Kbytes, 01101 = 1 Mbyte, 01110 = 2 Mbytes, 01111 = 4 Mbytes, 10000 = 8 Mbytes, 10001 = 16 Mbytes, 10010 = 32 Mbytes, 10011 = 64 Mbytes,10100 = 128 Mbytes, 10101 = 256 Mbytes,10110 = 512 Mbytes, 10111 = 1 Gbyte, 11000 = 2 Gbytes, 11001 = 4 Gbytes, 11010 = 8 Gbytes, 11011 = 16 Gbytes, 11100 = 32 Gbytes, 11101 = 64 Gbytes, 11110 = 128 Gbytes, 11111 = 256 Gbytes	5'hf
7:5	R/W	VF BAR 4 Control [VFBAR4C]	Specifies the configuration of VF BAR4. The various encodings are: 000: Disabled 001-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	3'h6
12:8	R/W	VF BAR 5 Aperture [VFBAR5A]	Specifies the aperture of the VF BAR 5 when it is configured as a 32-bit BAR. The encodings are: 00000 = 128 Bytes, 0001 = 256 Bytes, 0010 = 512 Bytes, 0011 = 1 Kbytes, 00100 = 2 Kbytes, 00101 = 4 Kbytes, 00110 = 8 Kbytes, 00111 = 16 Kbytes, 01000 = 32 Kbytes, 01001 = 64 Kbytes, 01010 = 128 Kbytes, 01011 = 256 Kbytes, 01100 = 512 Kbytes, 01101 = 1 Mbyte, 01110 = 2 Mbytes, 01111 = 4 Mbytes, 10000 = 8 Mbytes, 10001 = 16 Mbytes, 10010 = 32 Mbytes, 10011 = 64 Mbytes, 10100 = 128 Mbytes, 10101 = 256 Mbytes, 10110 = 512 Mbytes, 10111 = 1 Gbyte, 11000 = 2 Gbytes	5'hf

Bits	SW	Name	Description	Reset
15:13	R/W	VF BAR 5 Control [VFBAR5C]	Specifies the configuration of VF BAR5. The various encodings are: 000: Disabled 001-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	3'h4
31:16	R	Reserved [R16]	Reserved	0x0

17.6.7.1.44 Physical Function Configuration Register

Propname：Physical Function Configuration Register
Address：@0x2c0
Description：This register contains the enable bits for all the Functions implemented by the core. Resetting the enable bit of a Function disables the Function from responding to configuration requests.

Bits	SW	Name	Description	Reset
0	R	Function 0 Enable [F0E]	Enable for Function 0. This bit is hardwired to 1.	0x01
31:1	R	Reserved [R]	Reserved	0x0

17.6.7.1.45 Root Complex BAR Configuration Register

Propname：Root Complex BAR Configuration Register
Address：@0x300
Description：The root complex side of the core contains two memory BARs that can be used for address-range checking of incoming requests from devices connected to it. The fields in this register determine the configuration of these BARs.

Bits	SW	Name	Description	Reset
5:0	R/W	RC BAR 0 Aperture [RCBAR0A]	This field specifies the aperture of the RC BAR 0. The encodings are: 0000 = 4, 00001 =8B, ... 01_1111 = 8G, ...10_0100 = 256G.	0x14
8:6	R/W	RC BAR 0 control [RCBAR0C]	Specifies the configuration of RC BAR0. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110: 64bit memory BAR, non prefetchable 111: 64bit memory BAR, prefetchable	0x6
13:9	R/W	RC BAR 1 Aperture [RCBAR1A]	This field specifies the aperture of the RC BAR 1. The encodings are: 0000 = 4, 00001 =8B, ... 1_1101 = 2G	0x14
16:14	R/W	RC BAR 1 control [RCBAR1C]	Specifies the configuration of RC BAR1. The various encodings are: 000: Disabled 001: 32bit IO BAR 010-011: Reserved 100: 32bit memory BAR, non prefetchable 101: 32bit memory BAR, prefetchable 110-111: Reserved	0x4

Bits	SW	Name	Description	Reset
17	R/W	Type1 cfg prefetchable mem bar enable [RCBARPME]	Enable for Prefetchable memory base and limit registers in type1 config space	0x0
18	R/W	Type1 cfg prefetchable mem bar size [RCBARPMS]	Width of Prefetchable Memory Base and Limit registers in type1 config space. 0=32 bits, 1=64bits	0x0
19	R/W	Type1 cfg IO bar enable [RCBARPIE]	Enable for IO Base and Limit registers in type1 config space	0x0
20	R/W	Type1 cfg IO bar size [RCBARPIS]	Width of IO Base and Limit registers in type1 config space. 0=32 bits, 1=64bits	0x0
30:21	R	Reserved [R10]	Reserved	0x0
31	R/W	RC BAR Check Enable [RCBCE]	This bit must be set to 1 to enable BAR checking in the RC mode. When this bit is set to 0, the core will forward all incoming memory requests to the client logic without checking their address ranges.	0x0

17.6.8 Address Translation Registers Description

17.6.8.1 ATR Configuration Register Address Map

Address	Register Group	Register
0x000	Region 0 Outbound Config Registers	ob_addr0
0x004		ob_addr1
0x008		ob_desc0
0x00C		ob_desc1
0x010		ob_desc2
0x014		ob_desc3
0x018	Unused	
0x01C	Unused	
0x020	Region 1 Outbound Config Registers	
0x038		
	Unused	
0x040	Region 2 Outbound Config Registers	
0x058		
:		
0x400	Region 32 Outbound Config Registers	
0x418	Please see the description below to know how to select the regions.	
	Unused	
0x800	BAR0 RP Inbound Address Translation Registers	ib_addr0
0x804		ib_addr1
0x808	BAR1 RP Inbound Address Translation Registers	ib_addr0
0x80C		ib_addr1
0x810	Inbound Address Translation	ib_addr0

Address	Register Group	Register
0x814	Registers RP - No BAR Match	ib_addr1
0x824	Link_down_indication_bit	
0x828	BAR 0 Function 0 EP Inbound Address	ib_addr0
0x82c	BAR 0 Function 0 EP Inbound Address	ib_addr1
0x830	BAR 1 Function 0 EP Inbound Address	ib_addr0
0x834	BAR 1 Function 0 EP Inbound Address	ib_addr1
0x838	BAR 2 Function 0 EP Inbound Address	ib_addr0
0x83c	BAR 2 Function 0 EP Inbound Address	ib_addr1
0x840	BAR 3 Function 0 EP Inbound Address	ib_addr0
0x844	BAR 3 Function 0 EP Inbound Address	ib_addr1
0x848	BAR 4 Function 0 EP Inbound Address	ib_addr0
0x84c	BAR 4 Function 0 EP Inbound Address	ib_addr1
0x850	BAR 5 Function 0 EP Inbound Address	ib_addr0
0x854	BAR 5 Function 0 EP Inbound Address	ib_addr1
0x858	ROM BAR Function 0 EP Inbound Address	ib_addr0
0x85c	ROM BAR Function 0 EP Inbound Address	ib_addr1

The AXI wrapper performs simple address decoding and response multiplexing to map a single APB port to its configuration registers, the DMA APB port and the PCIe core APB port. Accesses with PADDR [22:21] values of 2'b10 are passed to the AXI wrapper registers and accesses with PADDR [22:21] values of 2'b11 are passed to the DMA. All other accesses are passed to the PCIe core

17.6.8.2 Outbound ATR Register Descriptions

17.6.8.2.1 Outbound Region Address 0

Propname：Outbound Region Address 0
Address：@0x0
Description：Provides bits 31:0 of the OB Region address (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
5:0	R/W	Number_bits [5:0] [num_bits]	Number of bits of the addres sthat are valid	6'h00
7:6	R	Reserved [rsvd]	Bits 7 and 6 are reserved	2'b00
31:8	R/W	Address bits [31:8] [data]	Lower 32-bits of Address Register for region N	24'h00000000

17.6.8.2.2 Outbound Region Address 1

Propname：Outbound Region Address 1
Address：@0x4

Description：Provides bits 63:21 of the OB Region address (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R/W	Address bits [63:32] [data]	Upper 32-bits of Address Register for region N	32'h00000000

17.6.8.2.3 Outbound Region Descriptor 0

Propname：Outbound Region Descriptor 0
Address：@0x8
Description：Provides bits 31:0 of the OB Region descriptor (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R/W	Descriptor bits [31:0] [data]	Lowest 32-bits of Address Register for region N	32'h00000000

17.6.8.2.4 Outbound Region Descriptor 1

Propname：Outbound Region Descriptor 1
Address：@0xc
Description：Provides bits 63:21 of the OB Region descriptor (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R/W	Descriptor bits [63:32] [data]	Lower middle 32-bits of Address Register for region N	32'h00000000

17.6.8.2.5 Outbound Region Descriptor 2

Propname：Outbound Region Descriptor 2
Address：@0x10
Description：Provides bits 95:64 of the OB Region descriptor (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
15:0	R/W	Descriptor bits [95:64] [data]	Upper middle 32-bits of Address Register for region N	16'h00000000
31:16	R	Descriptor bits [95:64] [rsvd]	Upper middle 32-bits of Address Register for region N	16'h00000000

17.6.8.2.6 Outbound Region Descriptor 3

Propname：Outbound Region Descriptor 3
Address：@0x14
Description：Provides bits 127:96 of the OB Region descriptor (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R	Descriptor bits [127:96] [data]	Upmost 32-bits of Address Register for region N	32'h00000000

17.6.8.3 RP Inbound ATR Register Description

17.6.8.3.1 RP Inbound BAR Address Translation 0

Propname：RP Inbound BAR Address Translation 0

Address：@0x0
Description：Provides bits 31:8 of the IB AXI address (AWADDR/ARADDR) and the number of PCIE address bits passed through

Bits	SW	Name	Description	Reset
5:0	R/W	Number_bits [5:0] [num_bits]	Number of bits - 1 of the PCIE address passed through	6'h00
7:6	R	Reserved [rsvd0]	Bits 7 and 6 are reserved	2'b00
31:8	R/W	Address bits [31:8] [data]	Bits [31:8] of Address Register for BAR N	24'h00000000

17.6.8.3.2 RP Inbound BAR Address Translation 1
Propname：RP Inbound BAR Address Translation 1
Address：@0x4
Description：Provides bits 63:32 of the IB AXI address (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R/W	Address bits [63:32] [data]	Bits [63:32] of Address Register for BAR N	32'h00000000

17.6.8.4 Link down indication bit
17.6.8.4.1 Link down indication bit
Propname：Link down indication bit
Address：@0x24
Description：(no description)

Bits	SW	Name	Description	Reset
0	R/W	Link down indication bit [clear_link_down_bit]	This bit will be set when link down reset comes. client should clear this bit before issueing new traffic	1'h0
31:1	R	RSVD	RSVD	31'h00000000

17.6.8.5 EP Inbound ATR Register Description
17.6.8.5.1 EP Inbound BAR Address Translation 0
Propname：EP Inbound BAR Address Translation 0
Address：@0x0
Description：Provides bits 31:0 of the IB AXI address (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R/W	Address bits [31:0] [data]	Bits [31:0] of Address Register for BAR N	32'h00000000

17.6.8.5.2 EP Inbound BAR Address Translation 1
Propname：EP Inbound BAR Address Translation 1
Address：@0x4
Description：Provides bits 63:32 of the IB AXI address (AWADDR/ARADDR)

Bits	SW	Name	Description	Reset
31:0	R/W	Address bits [63:32] [data]	Bits [63:32] of Address Register for BAR N	32'h00000000

17.6.9 PCIe DMA registers Description

17.6.9.1 PCIe DMA Channel 0 Registers

17.6.9.1.1 PCIe DMA Channel 0 Control Register

Propname：PCIe DMA Channel 0 Control Register
Address：@0x0
Description：Enables software to control the UDMA Channel

Bits	SW	Name	Description	Reset
0	R/W	Go command bit [go]	Kicks off the uDMA channel controller to fetch valid Outbound or Inbound linked list	1'b0
1	R/W	Inbound or outbound select [ob_not_ib]	Determines the direction of the DMA transfer	1'b0
31:2	R	Reserved [reserve_30]	Reserved for future use	30'd0

17.6.9.1.2 PCIe DMA Channel 0 Start Pointer Lower Register

Propname PCIe DMA Channel 0 Start Pointer Lower Register
Address @0x4
Description Address of the first Linked List descriptor in system memory

Bits	SW	Name	Description	Reset
31:0	R/W	Start pointer Lower DWORD [ptr]	Lower 32-bits Pointer Address Registers	32'h00000000

17.6.9.1.3 PCIe DMA Channel 0 Start Pointer Upper Register

Propname PCIe DMA Channel 0 Start Pointer Upper Register
Address @0x8
Description Address of the first Linked List descriptor in system memory

Bits	SW	Name	Description	Reset
31:0	R/W	Start Pointer Upper DWORD [ptr]	Upper 32-bits Pointer Address Registers	32'h00000000

17.6.9.1.4 PCIe DMA Channel 0 Attribute Lower Register

Propname：PCIe DMA Channel 0 Attribute Lower Register
Address：@0xc
Description：Attribute data used by the UDMA channel when fetching and returning link list descriptors

Bits	SW	Name	Description	Reset
31:0	R/W	Descriptor Attributes Lower DWORD [attr]	Lower 32-bits Attribute Values used when fetching and returning link list descriptors	32'h00000000

17.6.9.1.5 PCIe DMA Channel 0 Attribute Upper Register

Propname：PCIe DMA Channel 0 Attribute Upper Register

Address：@0x10
Description：Attribute data used by the UDMA channel when fetching and returning link list descriptors

Bits	SW	Name	Description	Reset
31:0	R/W	Descriptor Attributes Upper DWORD [attr]	Upper 32-bit Attribute Values used when fetching and returning link list descriptors	32'h00000000

17.6.9.2 PCIe DMA Channel 1 Registers

17.6.9.2.1 PCIe DMA Channel 1 Control Register

Propname：PCIe DMA Channel 1 Control Register
Address：@0x14
Description：Enables software to control the UDMA Channel

Bits	SW	Name	Description	Reset
0	R/W	Go command bit [go]	Kicks off the uDMA channel controller to fetch valid Outbound or Inbound linked list	1'b0
1	R/W	Inbound or outbound select [ob_not_ib]	Determines the direction of the DMA transfer	1'b0
31:2	R	Reserved [reserve_30]	Reserved for future use	30'd0

17.6.9.2.2 PCIe DMA Channel 1 Start Pointer Lower Register

Propname：PCIe DMA Channel 1 Start Pointer Lower Register
Address：@0x18
Description：Address of the first Linked List descriptor in system memory

Bits	SW	Name	Description	Reset
31:0	R/W	Start pointer Lower DWORD [ptr]	Lower 32-bits Pointer Address Registers	32'h00000000

17.6.9.2.3 PCIe DMA Channel 1 Start Pointer Upper Register

Propname：PCIe DMA Channel 1 Start Pointer Upper Register
Address：@0x1c
Description：Address of the first Linked List descriptor in system memory

Bits	SW	Name	Description	Reset
31:0	R/W	Start Pointer Upper DWORD [ptr]	Upper 32-bits Pointer Address Registers	32'h00000000

17.6.9.2.4 PCIe DMA Channel 1 Attribute Lower Register

Propname：PCIe DMA Channel 1 Attribute Lower Register
Address：@0x20
Description：Attribute data used by the UDMA channel when fetching and returning link list descriptors

Bits	SW	Name	Description	Reset
31:0	R/W	Descriptor Attributes Lower DWORD [attr]	Lower 32-bits Attribute Values used when fetching and returning link list descriptors	32'h00000000

17.6.9.2.5 PCIe DMA Channel 1 Attribute Upper Register

Propname：PCIe DMA Channel 1 Attribute Upper Register
Address：@0x24
Description：Attribute data used by the UDMA channel when fetching and returning link list descriptors

Bits	SW	Name	Description	Reset
31:0	R/W	Descriptor Attributes Upper DWORD [attr]	Upper 32-bit Attribute Values used when fetching and returning link list descriptors	32'h00000000

17.6.9.3 PCIe DMA Common Registers

17.6.9.3.1 PCIe DMA Interrupt Register

Propname：PCIe DMA Interrupt Register
Address：@0xa0
Description：Each channel has 2 associated interrupts: an error and a done. When the appropriate channel interrupt enables and disable bit is asserted (1) and de-asserted (0) respectively, an interrupt is allowed to assert. There is also a further interrupt enable on a descriptor that must also be enabled to generate the done interrupt. The error interrupt shall assert when an error condition is detected The done interrupt shall assert when the channel completes the inbound or outbound transfer descriptor

Bits	SW	Name	Description	Reset
0	R/WOCLR	Channel 0 Done Interrupt [ch0_done_int]	Channel 0 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
1	R/WOCLR	Channel 1 Done Interrupt [ch1_done_int]	Channel 1 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
2	R/WOCLR	Channel 2 Done Interrupt [ch2_done_int]	Channel 2 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
3	R/WOCLR	Channel 3 Done Interrupt [ch3_done_int]	Channel 3 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
4	R/WOCLR	Channel 4 Done Interrupt [ch4_done_int]	Channel 4 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
5	R/WOCLR	Channel 5 Done Interrupt [ch5_done_int]	Channel 5 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
6	R/WOCLR	Channel 6 Done Interrupt [ch6_done_int]	Channel 6 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
7	R/WOCLR	Channel 7 Done Interrupt [ch7_done_int]	Channel 7 Done Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0

Bits	SW	Name	Description	Reset
8	R/WOCLR	Channel 0 Error Interrupt [ch0_error_int]	Channel 0 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
9	R/WOCLR	Channel 1 Error Interrupt [ch1_error_int]	Channel 1 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
10	R/WOCLR	Channel 2 Error Interrupt [ch2_error_int]	Channel 2 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
11	R/WOCLR	Channel 3 Error Interrupt [ch3_error_int]	Channel 3 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
12	R/WOCLR	Channel 4 Error Interrupt [ch4_error_int]	Channel 4 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
13	R/WOCLR	Channel 5 Error Interrupt [ch5_error_int]	Channel 5 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
14	R/WOCLR	Channel 6 Error Interrupt [ch6_error_int]	Channel 6 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
15	R/WOCLR	Channel 7 Error Interrupt [ch7_error_int]	Channel 7 Error Interrupt RegisterInterrupt, Sticky (individual bits)	1'b0
31:16	R	Reserved [reserve_16]		16'd0

17.6.9.3.2 PCIe DMA Interrupt Enable Register

Propname PCIe DMA Interrupt Enable Register
Address：@0xa4
Description：To enable interrupt generation for a particular channel, write 1 to the appropriate interrupt enable bit.

Bits	SW	Name	Description	Reset
0	R/WOCLR	Channel 0 Done Enable Interrupt [ch0_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
1	R/WOCLR	Channel 1 Done Enable Interrupt [ch1_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
2	R/WOCLR	Channel 2 Done Enable Interrupt [ch2_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
3	R/WOCLR	Channel 3 Done Enable Interrupt [ch3_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
4	R/WOCLR	Channel 4 Done Enable Interrupt [ch4_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
5	R/WOCLR	Channel 5 Done Enable Interrupt [ch5_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
6	R/WOCLR	Channel 6 Done Enable Interrupt [ch6_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1

Bits	SW	Name	Description	Reset
7	R/WOCLR	Channel 7 Done Enable Interrupt [ch7_done_ena]	Assert to 1 to enable done interrupts to be generated	1'b1
8	R/WOCLR	Channel 0 Error Enable Interrupt [ch0_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
9	R/WOCLR	Channel 1 Error Enable Interrupt [ch1_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
10	R/WOCLR	Channel 2 Error Enable Interrupt [ch2_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
11	R/WOCLR	Channel 3 Error Enable Interrupt [ch3_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
12	R/WOCLR	Channel 4 Error Enable Interrupt [ch4_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
13	R/WOCLR	Channel 5 Error Enable Interrupt [ch5_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
14	R/WOCLR	Channel 6 Error Enable Interrupt [ch6_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
15	R/WOCLR	Channel 7 Error Enable Interrupt [ch7_error_ena]	Assert to 1 to enable error interrupts to be generated	1'b1
31:16	R	Reserved [reserve_16]		16'd0

17.6.9.3.3 PCIe DMA Interrupt Disable Register

Propname：PCIe DMA Interrupt Disable Register
Address：@0xa8
Description：To disable interrupt generation for a particular channel, write 1 to the appropriate interrupt disable bit.

Bits	SW	Name	Description	Reset
0	R/WOSET	Channel 0 Done Disable Interrupt [ch0_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
1	R/WOSET	Channel 1 Done Disable Interrupt [ch1_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
2	R/WOSET	Channel 2 Done Disable Interrupt [ch2_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
3	R/WOSET	Channel 3 Done Disable Interrupt [ch3_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1

Bits	SW	Name	Description	Reset
4	R/WOSET	Channel 4 Done Disable Interrupt [ch4_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
5	R/WOSET	Channel 5 Done Disable Interrupt [ch5_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
6	R/WOSET	Channel 6 Done Disable Interrupt [ch6_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
7	R/WOSET	Channel 7 Done Disable Interrupt [ch7_done_dis]	Assert to 1 to disable done interrupts to be generated	1'b1
8	R/WOSET	Channel 0 Error Disable Interrupt [ch0_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
9	R/WOSET	Channel 1 Error Disable Interrupt [ch1_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
10	R/WOSET	Channel 2 Error Disable Interrupt [ch2_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
11	R/WOSET	Channel 3 Error Disable Interrupt [ch3_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
12	R/WOSET	Channel 4 Error Disable Interrupt [ch4_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
13	R/WOSET	Channel 5 Error Disable Interrupt [ch5_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
14	R/WOSET	Channel 6 Error Disable Interrupt [ch6_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
15	R/WOSET	Channel 7 Error Disable Interrupt [ch7_error_dis]	Assert to 1 to disable error interrupts to be generated	1'b1
31:16	R	Reserved [reserve_16]		16'd0

17.6.9.3.4 PCIe DMA Inbound Buffer Uncorrected ECC Errors

Propname：PCIe DMA Inbound Buffer Uncorrected ECC Errors
Address：@0xac
Description：Total number of Inbound Buffer Uncorrected ECC Errors

Bits	SW	Name	Description	Reset
15:0	R	ECC Error Reg [total]	ECC Error Detection Register	16'd0
31:16	R	reserved_16 [reserve_16]	Reserved for future use	16'd0

17.6.9.3.5 PCIe DMA Inbound Buffer corrected ECC Errors

Propname：PCIe DMA Inbound Buffer corrected ECC Errors
Address：@0xb0
Description：Total number of Inbound Buffer corrected ECC Errors

Bits	SW	Name	Description	Reset
15:0	R	ECC Error Reg [total]	ECC Error Detection Register	16'd0
31:16	R	reserved_16 [reserve_16]	Reserved for future use	16'd0

17.6.9.3.6 PCIe DMA Outbound Buffer Uncorrected ECC Errors

Propname：PCIe DMA Outbound Buffer Uncorrected ECC Errors
Address：@0xb4
Description：Total number of Outbound Buffer Uncorrected ECC Errors

Bits	SW	Name	Description	Reset
15:0	R	ECC Error Reg [total]	ECC Error Detection Register	16'd0
31:16	R	reserved_16 [reserve_16]	Reserved for future use	16'd0

17.6.9.3.7 PCIe DMA Outbound Buffer corrected ECC Errors

Propname：PCIe DMA Outbound Buffer corrected ECC Errors
Address：@0xb8
Description：Total number of Outbound Buffer corrected ECC Errors

Bits	SW	Name	Description	Reset
15:0	R	ECC Error Reg [total]	ECC Error Detection Register	16'd0
31:16	R	reserved_16 [reserve_16]	Reserved for future use	16'd0

17.6.9.3.8 PCIe DMA Capability and Version Register

Propname：PCIe DMA Capability and Version Register
Address：@0xf8
Description：Allows software to read the version numbers of the uDMA IP Block

Bits	SW	Name	Description	Reset
7:0	R	min_ver	Minor Version No	8'd1
15:8	R	maj_ver	Major Version No	8'd0
31:16	R	reserved_16 [reserve_16]	Reserved for future use	16'd0

17.6.9.3.9 PCIe DMA Configuration Register

Propname：PCIe DMA Configuration Register
Address：@0xfc
Description：Enables software to read various configration parameters

Bits	SW	Name	Description	Reset
3:0	R	num_channels	Number of uDMA Channels	4'd2
7:4	R	num_partitions	Number of DPRAM Partitions	4'd2
11:8	R	partition_size	Size of each Partition	4'd5
12	R	sys_aw_gt_32	Sys Addr Width > 32-bits	1'd0
13	R	sys_tw_gt_32	Sys Attr Width > 32-bits	1'd0

Bits	SW	Name	Description	Reset
14	R	ext_aw_gt_32	Ext Addr Width > 32-bits	1'd1
15	R	ext_tw_gt_32	Ext Attr Width > 32-bits	1'd1
31:16	R	Reserved [reserve_16]	Reserved for future use	16'd0

17.6.10 PCIe PIPE PHY registers Description

17.6.10.1 Test Control Register

Manual configuration for analogue circuit parameter adjustment and internal test control is allowed. It is implemented with a 5-bit address bus (TEST_ADDR[4:0]) and a 4-bit data bus (TEST_I[3:0]), plus an asynchronous stroke(TEST_WRITE). When data for the selected register is provided, it is latched internally when strobe(TEST_WRITE) is set to high. TEST_ADDR_5 is don't care when register writing

Address TEST_ADDR [4:0]	Bit Description (default value)			
	TEST_I[3]	TEST_I[2]	TEST_I[1]	TEST_I[0]
00000	lb_test_mode (4'b000)			
00001	JITTER_AMP[3:0] (4'b0000)			
00010	JITTER_FREQ[5:2] (4'b0000)			
00011	JITTER_AMP[5:4] (2'b00)		JITTER_FREQ[1:0] (2'b00)	
00100	DESKEW_TRIM (4'b0000)			
00101	EQ_BITS_L (2'b01)		EQ_BITS_S (2'b01)	
00110	EQ_TRIM[5:4] (2'b11)		DESKEW_AUTO_DIS (1'b0)	EQ_AUTO_DIS (1'b0)
00111	EQ_TRIM[3:0] (4'b1111)			
01000	SPLIT_CP_DIS (1'b0)	TXPLL_TRIM (3'b101)		
01001	DOUBLE_PI_CURRENT (1'b0)	DOUBLE_PI_CMP_CURRENT (1'b0)	TXPLL_TRIM_LBW (2'b10)	
01010	TX_SWING_COMP (4'b1000)			
01011	TX_TERMINATION_TRIM (4'b1000)			
01100	BANDGAP_TRIM (2'b00)		DESKEW_MAGNITUDE (2'b00)	
01101	BYPASS_3RD_ORDER (1'b0)	TURBO_DETECT (1'b0)	CDR_DIRECT_TRIM (2'b01)	
01110	CDR_CPF_TRIM (2'b10)		Reserved	SLEW_ASSIST_DIS (1'b0)
01111	CDR_CPD_TRIM (2'b01)		Reserved	
10000	separate_rate (1'b0)	INVERT_PLL_CLK (1'b0)	TEST_CLK_SEL (2'b00)	
10001	SSCG_DISABLE (1'b1)	SSC_REDUCE (1'b0)	SSC_OFFSET (2'b00)	
10010	SEL_PLL_100M (1'b0)	GATE_100M (1'b1)	Reserved	
10011	aid_idle_det (1'b0)	IDLE_DET_TRIM (1'b0)	l1ss_custom (1'b1)	keep_rx_term_on (1'b0)
10100	pwdn_write (1'b0)	HERSHEY (1'b0)	PWON_PLL (1'b1)	PWON_IRF (1'b1)
10101	Reserved			
10110	Reserved			
10111	RX_TERMINATION_TRIM (4'b0101)			
11000	A_HIGH_RX_DET_TH (1'b1)	A_INVERT_CDR_CLK (1'b0)	a_cdr_fl_en (1'b0)	a_cdr_loop_manual (1'b0)
11001	B_HIGH_RX_DET_TH (1'b1)	B_INVERT_CDR_CLK (1'b0)	b_cdr_fl_en (1'b0)	b_cdr_loop_manual (1'b0)

Address TEST_ADDR [4:0]	**Bit Description** (default value)			
	TEST_I[3]	TEST_I[2]	TEST_I[1]	TEST_I[0]
11010	C_HIGH_RX_DET_TH (1'b1)	C_INVERT_CDR_CLK (1'b0)	c_cdr_fl_en (1'b0)	c_cdr_loop_manual (1'b0)
11011	D_HIGH_RX_DET_TH (1'b1)	D_INVERT_CDR_CLK (1'b0)	d_cdr_fl_en (1'b0)	d_cdr_loop_manual (1'b0)
11100	A_PWON_TX_BUF (1'b1)	A_PWON_IDLE_DET (1'b1)	A_PWON_RX_BUF (1'b1)	A_PWON_CDR (1'b1)
11101	B_PWON_TX_BUF (1'b1)	B_PWON_IDLE_DET (1'b1)	B_PWON_RX_BUF (1'b1)	B_PWON_CDR (1'b1)
11110	C_PWON_TX_BUF (1'b1)	C_PWON_IDLE_DET (1'b1)	C_PWON_RX_BUF (1'b1)	C_PWON_CDR (1'b1)
11111	D_PWON_TX_BUF (1'b1)	D_PWON_IDLE_DET (1'b1)	D_PWON_RX_BUF (1'b1)	D_PWON_CDR (1'b1)

- lb_test_mode: selects the mode of loopback BIST (values not listed below are reserved)

Mode		**Description** (run time measured in 5Gbps)
Bit[3]	Bit[2:0]	
1'b0	3'b000	Normal operation and not self loopback
1'b1	3'b000	External BIST(about 280us), requires external connection from TxP/TxN to RxP/RxN
1'b0	3'b001	Internal BIST(about 280us) without self jitter
1'b1	3'b001	Internal BIST(about 280us) with pre-defined self jitter freq of 50MHz and jitter amp of 40
1'b0	3'b100	Fast receiver jitter tolerance test with 10ms each jitter point, automatic test jitter at frequency of 5, 10, 20, 50MHz, with jitter amp started from 6'd6 and increased by 6'd2 at each step.
1'b1	3'b100	Fast receiver jitter tolerance test with 10ms each jitter point, automatic test jitter(reduced) at frequency of 5, 10, 20, 50MHz, with jitter amp started from 6'd6 and increased by 6'd2 at each step.
1'b0	3'b110	Medium receiver jitter tolerance test with 100ms each jitter point, automatic test jitter at frequency of 5, 10, 15, 20, 30, 40, 50, 60MHz, with jitter amp started from 6'd6 and increased by 6'd1 at each step.
1'b1	3'b110	Medium receiver jitter tolerance test with 100ms each jitter point, automatic test jitter(reduced) at frequency of 5, 10, 15, 20, 30, 40, 50, 60MHz, with jitter amp started from 6'd6 and increased by 6'd1 at each step
1'b0	3'b010	Slow receiver jitter tolerance test with 1s each jitter point, automatic test jitter at frequency of (1+N*4)MHz, with jitter amp started from 6'd0 and increased by 6'd1 at each step.
1'b1	3'b010	Slow receiver jitter tolerance test with 1s each jitter point, automatic test jitter(reduced) at frequency of (1+N*4)MHz, with jitter amp started from 6'd0 and increased by 6'd1 at each step.
1'b0	3'b011	Manual jitter tolerance test of about 6s. Jitter freq and jitter amp are controlled by JITTER_FREQ and JITTER_AMP(register address 5'b00001 ~ 5'b00011).
1'b1	3'b011	Manual jitter(reduced) tolerance test of about 6s. Jitter freq and jitter amp are controlled by JITTER_FREQ and JITTER_AMP(register address 5'b00001 ~ 5'b00011).

- JITTER_FREQ[5:2]: Also used as address of auto jitter tolerance test result(i.e. max_passed_jitter) which shows the max passed jitter amplitude at different jitter frequencies configured by JITTER_FREQ[5:2].

Result Address (JITTER_FREQ[5:2])	Jitter Frequency (MHz)		
	FAST mode	MEDIUM mode	SLOW mode
4'b0000	-	-	1
4'b0001	5	5	5
4'b0010	10	10	9
4'b0011	-	15	13
4'b0100	-	-	17
4'b0101	20	20	21
4'b0110	-	-	25
4'b0111	-	30	29
4'b1000	-	-	33
4'b1001	-	-	37
4'b1010	-	40	41
4'b1011	-	-	45
4'b1100	50	50	49
4'b1101	-	-	53
4'b1110	-	-	57
4'b1111	-	60	61

- SPLIT_CP_DIS: The High BandWidth Charge Pump will be off when PLL locked if this bit is low, and thus only TXPLL_TRIM_LBW take effect. When this bit is high, both CPs work all the time.
- TXPLL_TRIM: This value (from '000' to '111') controls the total CP(HBW+LBW) current to be 1.33uA, 2.67uA, 4uA, 5.33uA, 8uA, 16uA, 24uA and 32uA.
- TXPLL_TRIM_LBW: 0uA, 0.67uA, 1.33uA, 2.67uA current for Low BandWidth Charge Pump. Shall not be larger than the total value chosen by TXPLL_TRIM.
- BYPASS_3RD_ORDER: Bypass the 3rd order filter in PLL.
- DOUBLE_PI_CURRENT: Doubles the current in the phase interpolater.
- DOUBLE_PI_CMP_CURRENT: Doubles the current in phase interpolater comparator.
- TX_SWING_COMP: This value should be set to (64-56*0.9/VDD09) depending on actual voltage of VDD09.
- TX_TERMINATION_TRIM: Adjusts resistor termination at TX side.
- DESKEW_MAGNITUDE: Adjusts the magnitude in deskew circuit. Default 00 is 2x, 01 is 1x, 11 is 3x and 10 turns it off.
- DESKEW_AUTO_DIS: Disables the automatic DESKEW adjusting algorithm and make use of the value that set manually to DESKEW_TRIM.
- EQ_AUTO_DIS: Disables the automatic EQ adjusting algorithm and make use of the value that set manually to EQ_TRIM [5:0].
- EQ_BITS_L: Indicates (VALUE+2) bits are required to qualify as a long bit run in the auto equalization algorithm.
- EQ_BITS_S: Indicates (VALUE+1) bit run is required before a short bit for it to qualify as a short bit run in the auto equalization algorithm.
- BANDGAP_TRIM: Adjusts bandgap regulator current.
- CDR_DIRECT_TRIM: Adjusts the amount of direct control in CDR.
- CDR_CPF_TRIM: Adjusts the charge pump current in frequency loop.
- CDR_CPD_TRIM: Adjusts the charge pump current in data loop.
- TEST_CLK_SEL: Selects which lane of cdr clock is probed from cdr_test_clk (TEST_O[0] when TEST_ADDR[5:0] is set to 6'b010000) with the frequency of RATE/80. 00 for lane A, 01 for lane B, 10 for lane C and 11 for lane D.
- INVERT_PLL_CLK: Inverts polarity of PLL clock at input from analog to PCS.
- separate_rate: Forces PLL VCO to always run at 5Gbps with additional divider, while only CDR rate is controlled by RATE_I. When set low, both PLL and CDR run at RATE_I.
- SSCG_DISABLE: Disables SSC modulation. Shall only be enabled when all the conditions are met: 1) in systems with common clocked architecture, 2) accuracy of REF_CLK frequency is within +/-300ppm, 3) the PHY is in the device acting as Root Complex, 4) 100MHz differential output clock is enabled and used for reference clock of

EndPoint, 5) PLL is selected as the source of the differential 100MHz output clock.
- SSC_REDUCE: Reduces SSC swing from 0~-4500ppm to -2000ppm~-4500ppm.
- SSC_OFFSET: Adjusts SSC offset: 01 down 100ppm, 10 down 200ppm, 11 up 100ppm.
- SEL_PLL_100M: 100MHz differential clock is generated by PLL when this bit is high, otherwise it is from input port REF_CLK_100M. Note: the 100MHz clock from PLL would not be active until it has locked.
- GATE_100M: The differential 100MHz output clock is gated asynchronously (high impedance) when this bit is high. By reset, this bit is high and the LVDS clock output is off regardless of which source the 100MHz clock is from. When this bit is low and PLL is selected as source of the 100MHz output but has not locked, LVDS clock output will keep constant and would NOT be off automatically, i.e. it is not high impedance and will draw current.
- keep_rx_term_on: Always keep RX_TERMINATION on even if lane is turned off.
- IDLE_DET_TRIM: Adjusts hysteresis of idle pulse detector, by which the threshold is changed from 75mV~175mV to 20mV~80mV.
- aid_idle_det: Enables a set of PCS logic to aid idle detection in case problem occurred in analog idle detection circuit.
- pwdn_write: When set high, A/B/C/D_PWON_* and PWON_* values would be used to power on/off individual blocks. By default, those powers are controlled by PCS logic automatically.
- RX_TERMINATION_TRIM: Adjusts resistor termination at RX side.
- A/B/C/D_HIGH_RX_DET_TH: Use a high threshold for RX detection circuit.
- a/b/c/d_cdr_loop_manual: Select whether corresponding cdr loop is selected manually or controlled automatically by PCS logic.
- a/b/c/d_cdr_fl_en: When corresponding cdr_loop_manual is high, these registers force CDR to be in frequency loop when they are high while be in data loop when they are low.
- A/B/C/D_INVERT_CDR_CLK: Invert polarity of each CDR clock at corresponding input from analog to PCS.

17.6.10.2 Probe for Debugging

For debugging purpose, a group of internal signals are probed via four dedicated output bus(TEST_O[3:0]) according to the 6-bit address bus(TEST_ADDR[5:0]). See table below.

Address Bus TEST_ADDR [5:0]	**Probing Signals – TEST_O[3:0]**			
000000	BIST_FINISH	BIST_RESULT	BIST_TIMEOUT	BIST_RUNNING
000001	4'b0000			
000010	4'b0000			
000011	TX_BEACON_O	TX_DETECT_RX_O	lp_rx_en	INTERNAL_LOOP
000100	DESKEW_TRIM			
000101	EQ_BITS_L		EQ_BITS_S	
000110	EQ_TRIM[5:4]		DESKEW_AUTO_DIS	EQ_AUTO_DIS
000111	EQ_TRIM[3:0]			
001000	SPLIT_CP_DIS	TXPLL_TRIM		
001001	DOUBLE_PI_CURRENT	DOUBLE_PI_CMP_CURRENT	TXPLL_TRIM_LBW	
001010	TX_SWING_COM			
001011	TX_TERMINATION_TRIM			
001100	BANDGAP_TRIM		DESKEW_MAGNITUDE	
001101	Reserved	TURBO_DETECT	CDR_DIRECT_TRIM	
001110	CDR_CPF_TRIM		Reserved	Reserved
001111	CDR_CPD_TRIM		Reserved	Reserved
010000	pll_test_clk	CLK_PI_FB	TXPLL_LOCK	cdr_test_clk

Address Bus TEST_ADDR [5:0]	**Probing Signals – TEST_O[3:0]**			
010001	SSCG_DISABLE	SSC_REDUCE	SSC_OFFSET	
010010	SEL_PLL_100M	GATE_100M	Reserved	Reserved
010011	1'b0	IDLE_DET_TRIM	2'b00	
010100	1'b0	Reserved	PWON_PLL_O	PWON_IRF_O
010101	Reserved			
010110	Reserved			
010111	RX_TERMINATION_TRIM			
011000	A_HIGH_RX_DET_TH	1'b0	A_CDR_FL_EN_O	A_CDR_FRQ_LOCK
011001	B_HIGH_RX_DET_TH	1'b0	B_CDR_FL_EN_O	B_CDR_FRQ_LOCK
011010	C_HIGH_RX_DET_TH	1'b0	C_CDR_FL_EN_O	C_CDR_FRQ_LOCK
011011	D_HIGH_RX_DET_TH	1'b0	D_CDR_FL_EN_O	D_CDR_FRQ_LOCK
011100	A_PWON_TX_BUF_O	A_PWON_IDLE_D ET_O	A_PWON_RX_BUF_O	A_PWON_CDR_O
011101	B_PWON_TX_BUF_O	B_PWON_IDLE_D ET_O	B_PWON_RX_BUF_O	B_PWON_CDR_O
011110	C_PWON_TX_BUF_O	C_PWON_IDLE_D ET_O	C_PWON_RX_BUF_O	C_PWON_CDR_O
011111	D_PWON_TX_BUF_O	D_PWON_IDLE_D ET_O	D_PWON_RX_BUF_O	D_PWON_CDR_O
100000	BIST_FINISH	a_bist_result	a_bist_timeout	1'b0
100001	BIST_FINISH	b_bist_result	b_bist_timeout	1'b0
100010	BIST_FINISH	c_bist_result	c_bist_timeout	1'b0
100011	BIST_FINISH	d_bist_result	d_bist_timeout	1'b0
100100	A_TX_ELECIDLE_O	A_RX_DETECTED_I	A_DESKEW_SERIAL	A_EQ_SERIAL
100101	B_TX_ELECIDLE_O	B_RX_DETECTED_I	B_DESKEW_SERIAL	B_EQ_SERIAL
100110	C_TX_ELECIDLE_O	C_RX_DETECTED_I	C_DESKEW_SERIAL	C_EQ_SERIAL
100111	D_TX_ELECIDLE_O	D_RX_DETECTED_I	D_DESKEW_SERI AL	D_EQ_SERIAL
101000	a_max_passed_jitter[5:2]			
101001	b_max_passed_jitter[5:2]			
101010	c_max_passed_jitter[5:2]			
101011	d_max_passed_jitter[5:2]			
101100	a_brst_sym		a_max_passed_jitter[1:0]	
101101	b_brst_sym		b_max_passed_jitter[1:0]	
101110	c_brst_sym		c_max_passed_jitter[1:0]	
101111	d_brst_sym		d_max_passed_jitter[1:0]	
110000	A_RX_ELECIDLE_O	A_RX_IDLE_I	1'b0	A_RX_TERMINATI ON_O
110001	B_RX_ELECIDLE_O	B_RX_IDLE_I	1'b0	B_RX_TERMINATI ON_O

Address Bus TEST_ADDR [5:0]	Probing Signals – TEST_O[3:0]			
110010	C_RX_ELECIDLE_O	C_RX_IDLE_I	1'b0	C_RX_TERMINATI ON_O
110011	D_RX_ELECIDLE_O	D_RX_IDLE_I	1'b0	D_RX_TERMINATI ON_O
110100	A_RX_VALID_O	a_comma_det	a_comma_receive d	a_cdr_data_valid
110101	B_RX_VALID_O	b_comma_det	b_comma_receive d	b_cdr_data_valid
110110	C_RX_VALID_O	c_comma_det	c_comma_received	c_cdr_data_valid
110111	D_RX_VALID_O	d_comma_det	d_comma_receive d	d_cdr_data_valid
111000	a_disparity_err		a_invalid_code	
111001	b_disparity_err		b_invalid_code	
111010	c_disparity_err		c_invalid_code	
111011	d_disparity_err		d_invalid_code	
111100	a_eb_of	a_eb_uf	a_skp_removed	a_skp_inserted
111101	b_eb_of	b_eb_uf	b_skp_removed	b_skp_inserted
111110	c_eb_of	c_eb_uf	c_skp_removed	c_skp_inserted
111111	d_eb_of	d_eb_uf	d_skp_removed	d_skp_inserted

- BIST_FINISH: Indicates one BIST operation finished, and BIST_RESULT is ready for read.
- BIST_RESULT: A high value indicates BIST passed, while low means fail.
- *_bist_result: BIST result for each lane.
- BIST_TIMEOUT: Indicates if there is timeout occurred during the BIST.
- *_bist_timeout: BIST timeout indication of each lane.
- BIST_RUNNING: Shows the BIST is in progress.
- INTERNAL_LOOP: Shows the BIST is one that loops serialized stream to AFE input by enabling internal connection.
- lp_rx_en: Standard RX loopback enabled.
- TX_DETECT_RX_O: Controls TX buffer to take an RX detection operation
- *_RX_DETECTED_I: Receiver detection result from analog for each lane. During simulation, only when the other side of the connection has pulldown resistor(either added internally by the other device or directly in testbench on differential lines) to its receiver, can rx detection get a device presence result. It could be implemented by either the verilog 'pulldown' syntax or the same way as in our receiver model(refer to description of *_RX_TERMINATION_O in this section).
- *_TX_ELECIDLE_O: Sets corresponding transmitter into idle state.
- TX_BEACON_O: When TX_ELECIDLE_O of corresponding lane is low, that lane will transmit serialized data out if TX_BEACON_O is low and send BEACON if it is high. When TX_ELECIDLE_O is high, TX_BEACON_O is ignored by corresponding lane.
- cdr_test_clk: The internal clocks of each CDR can be viewed via cdr_test_clk. The lane is selected by register bit TEST_CLK_SEL (default is 00) as shown in table below.

Table 17-24 CDR Test CLK

TEST_CLK_SEL	cdr_test_clk (RATE/80)
00	CDR Clock of lane A divided by 4
01	CDR Clock of lane B divided by 4
10	CDR Clock of lane C divided by 4
11	CDR Clock of lane D divided by 4

- TXPLL_LOCK: Indicates the transmitter PLL has locked.
- CLK_PI_FB: 100MHz feedback clock in PI module probed.
- pll_test_clk: Probes internal clock of PLL at the frequency of RATE/40.

- *_CDR_FRQ_LOCK: Indicate the corresponding CDR has locked.
- *_CDR_FL_EN_O: When high, the corresponding CDR is set to frequency loop, while in data loop otherwise.
- *_DESKEW_SERIAL: About 94MHz signals for monitoring the state of the dynamic deskew circuit in corresponding lane, which should be muxed to probe out for test purpose.
- *_EQ_SERIAL: About 94MHz signals for monitoring the state of the dynamic equalization circuit in corresponding lane, which should be muxed to probe out for test purpose.
- *_max_passed_jitter: The max jitter amplitude, with jitter frequency set into register JITTER_FREQ[5:2], that passed during the self jitter tolerance BIST for each lane.
- *_brst_sym: Indicate BRST ordered set detected by corresponding lane, which is used to initialize every BIST point.
- *_RX_TERMINATION_O: Turn on the receiver termination of corresponding lane. In our verilog model for simulation, RX_N and RX_P have weak pulldown enabled by RX_TERMINATION to emulate the real resistor in design as below:
  input RX_N; input RX_P;
  input RX_TERMINATION;
  wire (strong1, weak0) RX_N =! RX_TERMINATION; wire (strong1, weak0) RX_P = !RX_TERMINATION;
- *_RX_IDLE_I: Indicate idle (no pulse longer than 600ps within 2 REF_CLK cycles) detected on differential receiver lines of corresponding lane.
- *_RX_ELECIDLE_O: Same signals as top output signals *_RX_ELECIDLE_O to MAC layer.
- *_cdr_data_valid: Indicate the data from corresponding CDR become valid for checking, since CDR has been set into data loop for about 1.3us. Note: this restricts that the Number of Fast Training Sequences(N_FTS) that link layer advertise during link training should at least be 210.
- *_comma_received: Indicate a comma symbol is detected in corresponding lane.
- *_comma_det: Indicate the symbol boundary for corresponding lane has been established.
- *_RX_VALID_O: Same signals as top output signals *_RX_VALID_O to MAC layer.
- *_disparity_err: Indicate corresponding decoder detect disparity error.
- *_invalid_code: Indicate corresponding decoder detect invalid code.
- *_skp_inserted: Indicate one SKP symbol inserted into data stream of corresponding lane.
- *_skp_removed: Indicate one SKP symbol removed from data stream of corresponding lane.
- *_eb_uf: Indicate underflow occurred in the elastic buffer of corresponding lane.
- *_eb_of: Indicate overflow occurred in the elastic buffer of corresponding lane.

17.7 PCIe Appendix

17.7.1 Appendix A. Debug Data Bus Signal Assignments

Table 17-25 Debug Bus Signals: Physical Layer

MUX_SELECT [3:2]	MUX_SELECT [1:0]	SIGNAL NAME	DEBUG BUS OUTPUT
00 (phy)	00 (LTSSM)	lane_count_negotiated[1:0]	DEBUG_DATA_OUT[0]
		link_up	DEBUG_DATA_OUT[1]
		phy_status	DEBUG_DATA_OUT[2]
		ltssm_state	DEBUG_DATA_OUT[10:4]
		tx_state	DEBUG_DATA_OUT[15:11]
	01 (LTSSM)	rcvr_detect_status[7:0]	DEBUG_DATA_OUT[7:0]

MUX_SELECT [3:2]	MUX_SELECT [1:0]	SIGNAL NAME	DEBUG BUS OUTPUT
		ts1_received[0]	DEBUG_DATA_OUT[8]
		ts2_received[0]	DEBUG_DATA_OUT[9]
		rx_lane_no_reversed[0]	DEBUG_DATA_OUT[10]
		tx_lane_no_reversed	DEBUG_DATA_OUT[11]
		de_scrambling_on	DEBUG_DATA_OUT[12]
		scrambling_on	DEBUG_DATA_OUT[13]
		link_tx_inactivity_timeout	DEBUG_DATA_OUT[14]
		timeout_12ms	DEBUG_DATA_OUT[15]

Table 17-26 Debug Bus Signals: Data Link Layer

MUX_SELECT [3:2]	MUX_SELECT [1:0]	SIGNAL NAME	DEBUG BUS OUTPUT
01 (Data Link Layer)	00 (TX side)	in_data_valid	DEBUG_DATA_OUT[15]
		in_sop	DEBUG_DATA_OUT[14]
		in_eop	DEBUG_DATA_OUT[13]
		in_error	DEBUG_DATA_OUT[12]
		out_tlp_data_valid	DEBUG_DATA_OUT[11]
		out_tlp_sop	DEBUG_DATA_OUT[10]
		out_tlp_eop	DEBUG_DATA_OUT[9]
		out_tlp_error	DEBUG_DATA_OUT[8]
		mgmt_dl_protocol_error	DEBUG_DATA_OUT[7]
		mgmt_replay_timer_ timeout_event	DEBUG_DATA_OUT[6]
		mgmt_replay_num_rollover_ status	DEBUG_DATA_OUT[5]
		mgmt_replay_buffer_ram_parity _error	DEBUG_DATA_OUT[4]
		start_replay	DEBUG_DATA_OUT[3]
		link_state[2:0]	DEBUG_DATA_OUT[2:0]
	01 (RX side)	out_data_valid	DEBUG_DATA_OUT[15]
		out_sop	DEBUG_DATA_OUT[14]
		out_eop	DEBUG_DATA_OUT[13]
		out_error	DEBUG_DATA_OUT[12]
		tlp_rcvd	DEBUG_DATA_OUT[11]
		in_tlp_error	DEBUG_DATA_OUT[10]
		in_tlp_nullified	DEBUG_DATA_OUT[9]
		crc32_error	DEBUG_DATA_OUT[8]
		in_dllp0_data_valid	DEBUG_DATA_OUT[7]
		in_dllp1_data_valid	DEBUG_DATA_OUT[6]
		ack_rcvd	DEBUG_DATA_OUT[5]
		ack_type	DEBUG_DATA_OUT[4]
		mgmt_dllp_error	DEBUG_DATA_OUT[3]
		mgmt_tlp_error	DEBUG_DATA_OUT[2]
		rx_state[1:0]	DEBUG_DATA_OUT[1:0]

Table 17-27 Debug Bus Signals: Transaction Layer, Transmit Side

MUX_SELECT [3:2]	MUX_SELECT [1:0]	SIGNAL NAME	DEBUG BUS OUTPUT
10 (Transaction	00 (TX side)	client_tx_pnp_valid	DEBUG_DATA_OUT[15]
		client_tx_pnp_type	DEBUG_DATA_OUT[14]

MUX_SELECT [3:2]	MUX_SELECT [1:0]	SIGNAL NAME	DEBUG BUS OUTPUT
Layer)		client_tx_pnp_sop	DEBUG_DATA_OUT[13]
		client_tx_pnp_eop	DEBUG_DATA_OUT[12]
		client_tx_sc_valid	DEBUG_DATA_OUT[11]
		client_tx_sc_sop	DEBUG_DATA_OUT[10]
		client_tx_sc_eop	DEBUG_DATA_OUT[9]
		local_compl_data_valid	DEBUG_DATA_OUT[8]
		local_msg_data_valid	DEBUG_DATA_OUT[7]
		link_tx_data_valid	DEBUG_DATA_OUT[6]
		posted_eligible	DEBUG_DATA_OUT[5]
		posted_accepted	DEBUG_DATA_OUT[4]
		non_posted_eligible	DEBUG_DATA_OUT[3]
		non_posted_accepted	DEBUG_DATA_OUT[2]
		compl_eligible	DEBUG_DATA_OUT[1]
		compl_accepted	DEBUG_DATA_OUT[0]

Table 17-28 Debug Bus Signals: Transaction Layer, Receive Side

MUX_SELECT [3:0]	MUX_SELECT [1:0]	SIGNAL NAME	DEBUG BUS OUTPUT
10 (Transaction Layer)	01(RX side)	pnp_receive_fifo_overflow	DEBUG_DATA_OUT[15]
		sc_receive_fifo_overflow	DEBUG_DATA_OUT[14]
		malformed_tlp_received	DEBUG_DATA_OUT[13]
		ecrc_error_detected	DEBUG_DATA_OUT[12]
		pnp_fifo_ram_parity_error	DEBUG_DATA_OUT[11]
		sc_fifo_ram_parity_error	DEBUG_DATA_OUT[10]
		ecrc_check_module_out_ dword_count[9:0]	DEBUG_DATA_OUT[9:0]
	10(RX side)	client_rx_pnp_valid	DEBUG_DATA_OUT[15]
		client_rx_pnp_type	DEBUG_DATA_OUT[14]
		client_rx_pnp_sop	DEBUG_DATA_OUT[13]
		client_rx_pnp_eop	DEBUG_DATA_OUT[12]
		client_rx_pnp_ready	DEBUG_DATA_OUT[11]
		client_rx_pnp_error	DEBUG_DATA_OUT[10]
		client_rx_sc_valid	DEBUG_DATA_OUT[9]
		client_rx_sc_sop	DEBUG_DATA_OUT[8]
		client_rx_sc_eop	DEBUG_DATA_OUT[7]
		client_rx_sc_ready	DEBUG_DATA_OUT[6]
		client_rx_sc_error	DEBUG_DATA_OUT[5]
		cfg_req_data_valid	DEBUG_DATA_OUT[4]
		cfg_req_ack	DEBUG_DATA_OUT[3]
		link_rx_data_valid	DEBUG_DATA_OUT[2]
		link_rx_sop	DEBUG_DATA_OUT[1]
		link_rx_eop	DEBUG_DATA_OUT[0]

17.7.2 Appendix B. LTSSM State Encoding

The following table provides the encoding of the LTSSM states on the LTSSM_STATE output of the core, as well the state read from the Physical Layer Configuration Register 0.

Table 17-29 LTSSM State Encoding

LTSSM State Name	Value (hex)
Detect.Quiet	00
Detect.Active	01
Polling.Active	02
Polling.Compliance	03
Polling.Configuration	04
Configuration.Linkwidth.Start	05
Configuration.Linkwidth.Accept	06
Configuration.Lanenum.Accept	07
Configuration.Lanenum.Wait	08
Configuration.Complete	09
Configuration.Idle	0A
Recovery.RcvrLock	0B
Recovery.Speed	0C
Recovery.RcvrCfg	0D
Recovery.Idle	0E
L0	10
Rx_L0s.Entry	11
Rx_L0s.Idle	12
Rx_L0s.FTS	13
Tx_L0s.Entry	14
Tx_L0s.Idle	15
Tx_L0s.FTS	16
L1.Entry	17
L1.Idle	18
L2.Idle	19
L2.TransmitWake	1A
Disabled	20
Loopback.Entry (Master)	21
Loopback.Active (Master)	22
Loopback.Exit (Master)	23
Loopback.Entry (Slave)	24
Loopback.Active (Slave)	25
Loopback.Exit (Slave)	26
Hot Reset	27
Recovery.Equalization, Phase 0	28
Recovery.Equalization, Phase 1	29
Recovery.Equalization, Phase 2	2A
Recovery.Equalization, Phase 3	2B

17.7.3 Appendix C. PERFORMANCE DATA OUT

Table 17-30 PERFORMANCE_DATA_OUT

Bit	Description
17	DLLP Nack packet comes to the Core.
16	DLLP Ack packet comes to the Core.
15	TLP Message write packet comes to the Core.
14	TLP Config write packet comes to the Core.
13	TLP Config read packet comes to the Core.
12	TLP I/O write packet comes to the Core.
11	TLP I/O read packet comes to the Core.
10	TLP memory write packet comes to the Core.
9	TLP memory read packet comes to the Core.
8	DLLP Nack packet goes from the Core.
7	DLLP Ack packet goes from the Core.
6	TLP Message write packet goes from the Core.
5	TLP Config write packet goes from the Core.
4	TLP Config read packet goes from the Core. When used as EP. This bit should be fixed 0
3	TLP I/O write packet goes from the Core.
2	TLP I/O read packet goes from the Core.
1	TLP memory write packet goes from the Core.
0	TLP memory read packet goes from the Core.

17.7.4 Appendix D. Programming the SRIOV Registers

The VF Function Numbers begin immediately after the PF Function Numbers. There are no gaps in the Function Number allocation between any PFs and VFs. The VF Stride is fixed at value 0x1. Hence all VFs that belong to the same PF are allocated successive function numbers.

The Function Numbers allocated upon reset is described in the table below:

Table 17-31 VF Function Number allocation

Routing ID	Description
0	PF0
1 to 8	PF0_VF1 to PF0_VF8

For a PCIe Controller with multiple PFs, the Total Number of VFs allocated to each PF can be changed by programming the Total VF Count[15:0] register in the PFs' SRIOV Extended Capabilities. When the Total VF Count field is modified, the VF function number allocation changes so as to ensure that all VFs are allocated successive function numbers. The First VF Offset field then needs to be re-programmed by the local firmware to reflect the new VF function number allocation as described below.

- PF0 First VF Offset[15:0] = Total Number of PFs
- PF1 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) - 1
- PF2 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) + (PF1 Total VF Count[15:0]) - 2
- PF3 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) + (PF1

Total VF Count[15:0]) + (PF2 Total VF Count[15:0]) - 3

- PF4 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) + (PF1 Total VF Count[15:0]) + (PF2 Total VF Count[15:0]) + (PF3 Total VF Count[15:0]) - 4

- PF5 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) + (PF1 Total VF Count[15:0]) + (PF2 Total VF Count[15:0]) + (PF3 Total VF Count[15:0]) + (PF4 Total VF Count[15:0]) - 5

- PF6 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) + (PF1 Total VF Count[15:0]) + (PF2 Total VF Count[15:0]) + (PF3 Total VF Count[15:0]) + (PF4 Total VF Count[15:0]) + (PF5 Total VF Count[15:0]) - 6

- PF7 First VF Offset[15:0] = Total Number of PFs + (PF0 Total VF Count[15:0]) + (PF1 Total VF Count[15:0]) + (PF2 Total VF Count[15:0]) + (PF3 Total VF Count[15:0]) + (PF4 Total VF Count[15:0]) + (PF5 Total VF Count[15:0]) + (PF6 Total VF Count[15:0]) - 7

Chapter 18 TypeC PHY

18.1 Overview

TypeC PHY is a combination of USB3.0 SuperSpeed PHY and DisplayPort Transmit PHY. It translates the protocol between PIPE Interface/DisplayPort Interface and differential lanes depending on the configurations of normal or flipped orientation, DP or USB lanes. This PHY will be a part to construct a fully feature TypeC subsystem.

TypeC PHY supports the following features:

- Support USB3.0 (SuperSpeed only)
- Support DisplayPort 1.3 (RBR, HBR and HBR2 data rates only)
- Support DisplayPort AUX channel
- Support USB TypeC and DisplayPort Alt Mode
- Support DisplayPort Alt Mode on TypeC A, B, C, D, E and F pin assignments
- Support Normal and Flipped orientation

18.2 Block Diagram

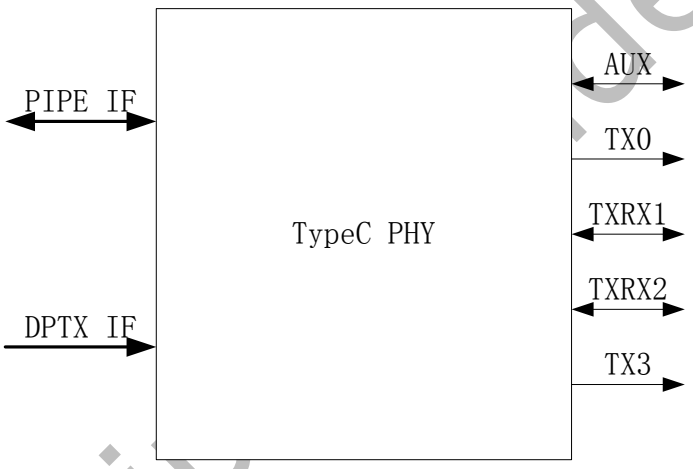


Fig. 18-1TypC PHY Block Diagram

18.3 Function Description

The USB Type-C PHY is designed to support the USB3 and DP applications. The PHY basically has two main components the PMA and the PCS. The PCS is optimized for USB3 protocol, whereas the PMA is designed to support multi-protocols. USB3 operates in SuperSpeed mode and the DP can operate at RBR, HBR and HBR2 data rates.

18.4 Register Description

18.4.1 High Level Address Mapping

Offset Bit index	Internal address region
17:16	Internal address space. • 2'b00: PMA macro common module registers. • 2'b01: PMA macro transmitter module and AUX TX registers. • 2'b10: PMA macro receiver module and AUX RX registers. • 2'b11: PHY level registers. Note: For PMA/AUX register space, see the PMA/AUX specification for addressing details.
15:14	When the PHY level register space is selected, these bits select PHY level or AUX macro registers. • 2'b00: PHY level registers • 2'b01, 2'b10 and 2'b11: Reserved
13	When the PHY level register space is selected, this bit selects PHY level PCS or PMA registers. Set to 0 when AUX register space is selected. • 1'b0: PHY level top registers. • 1'b1: PHY level PMA registers.
12	When the PHY level register space is selected, this bit selects PHY level common or lane registers. Set to 0 when AUX register space is selected. • 1'b0: PHY level common registers. • 1'b1: PHY level lane registers.
11	When the PHY level register space is selected, this bit is reserved, and shall be set to 0.
10:8	When the PHY level lane register space (PHY top or PMA level) is selected, and lane registers are selected, these bits specify the lane being addressed. • 3'b000: Lane 0 • 3'b001: Lane 1 • 3'b010: Lane 2 • 3'b011: Lane 3 • 3'b100 - 3'b111: Reserved When PHY level common registers are selected, these bits are reserved and shall be set to 3'b000.
5:0	When the PHY macro top level register space is selected, these bits specify up to 64 addressable registers for a given lane or common.

18.4.2 Low Level Address Mapping

Address	Register name
PHY top common registers	
18'b11_00_0_0_0_000_000000_00	PMA lane configuration register
18'b11_00_0_0_0_000_000001_00	PIPE common control1 register

Address	Register name
18'b11_00_0_0_0_000_000010_00	PIPE common control2 register
18'b11_00_0_0_0_000_000011_00	PIPE comma lock configuration1 register
18'b11_00_0_0_0_000_000100_00	PIPE comma lock configuration2 register
18'b11_00_0_0_0_000_000101_00	PIPE receiver detect inhibit register
18'b11_00_0_0_0_000_001000_00	PHY DP mode control register
18'b11_00_0_0_0_000_001001_00	PHY DP clock control register
18'b11_00_0_0_0_000_001111_00	PHY status register
18'b11_00_0_0_0_000_010000_00	PHY common control signal isolation register
PHY top lane registers (nnn = lane number: 3'b000 = lane 0, 3'b001 = lane 1, ..., 3'b11 = lane 3, others are reserved)	
18'b11_00_0_1_0_000_000000_00	USB loopback slave BER count register
18'b11_00_0_1_0_nnn_001000_00	PHY DP TX Control
18'b11_00_0_1_0_000_010000_00	PIPE TX control signal isolation register
18'b11_00_0_1_0_000_010001_00	PIPE TX deemphasis isolation register
18'b11_00_0_1_0_000_010010_00	PIPE TX data low isolation register
18'b11_00_0_1_0_000_010011_00	PIPE TX data high isolation register
18'b11_00_0_1_0_000_010100_00	PIPE RX control signal isolation register
18'b11_00_0_1_0_000_010101_00	PIPE link control isolation register
18'b11_00_0_1_0_000_010110_00	PIPE RX data low isolation register
18'b11_00_0_1_0_000_010111_00	PIPE RX data high isolation register
18'b11_00_0_1_0_nnn_011100_00	DP TX data low isolation register
18'b11_00_0_1_0_nnn_011101_00	DP TX data high isolation register
PHY PMA common registers	
18'b11_00_1_0_0_000_000000_00	PMA common control1 register
18'b11_00_1_0_0_000_000001_00	PMA common control2 register
18'b11_00_1_0_0_000_000010_00	PMA SSM current state register
18'b11_00_1_0_0_000_000011_00	PMA PLL0 State Machine current state register
18'b11_00_1_0_0_000_000100_00	PMA PLL1 State Machine current state register
18'b11_00_1_0_0_000_010000_00	PHY common control signal isolation register
18'b11_00_1_0_0_000_010001_00	PMA PLL control0 isolation register
18'b11_00_1_0_0_000_010010_00	PMA PLL control1 isolation register
18'b11_00_1_0_0_000_011111_00	Isolation control register
PHY PMA lane registers (nnn = lane number: 3'b000 = lane 0, 3'b001 = lane 1, ..., 3'b011 = lane 3, others are reserved)	

Address	Register name
18'b11_00_1_1_0_nnn_000000_00	PMA transceiver control register
18'b11_00_1_1_0_nnn_000001_00	PMA loopback control register
18'b11_00_1_1_0_nnn_000010_00	PMA PI position register
18'b11_00_1_1_0_nnn_000100_00	PMA Transceiver PSM current state low register
18'b11_00_1_1_0_nnn_000101_00	PMA Transceiver PSM current state high register
18'b11_00_1_1_0_nnn_010000_00	PMA transceiver control isolation register
18'b11_00_1_1_0_nnn_010001_00	PMA TX configuration isolation register
18'b11_00_1_1_0_nnn_010010_00	PMA link mode isolation register
18'b11_00_1_1_0_nnn_010011_00	PMA link power state control isolation register
18'b11_00_1_1_0_nnn_010100_00	PMA transmit low data isolation register
18'b11_00_1_1_0_nnn_010101_00	PMA transmit high data isolation register
18'b11_00_1_1_0_nnn_010110_00	PMA receive low data isolation register
18'b11_00_1_1_0_nnn_010111_00	PMA receive high data isolation register

18.4.3 Register Detail Description

PHY_PMA_LANE_CFG
Address: Operational Base + offset (18'b11_00_0_0_0_000_000000_00)
PMA lane configuration register

Bits	Type	Reset	Description
15:14	R/W	2'd1	PHY DP lane selection - when PHY DP I/F is selected for PMA lane 3, this field selects which PHY DP lane drives PMA lane 3. (Only valid with bit [12] == 1.) When configured for PHY DP, the same PHY DP lane can be mapped to only 1 PMA lane. This mapping is applicable for a normal connector orientation. The logic automatically adjusts the lane mapping for a flipped connector orientation.
13	R/O	1'b0	Reserved
12	R/W	1'b1	PMA lane 3 interface select - selects between PIPE PCS and PHY DP to drive PMA lane 3. 0 = PIPE PCS; 1 = PHY DP I/F.
11:10	R/W	2'd0	PHY DP lane selection - when PHY DP I/F is selected for PMA lane 2, this field selects which PHY DP lane drives PMA lane 2. (Only valid with bit [8] == 1.) When configured for DP, the same PHY DP lane can be mapped to only 1 PMA lane. This mapping is applicable for a normal connector orientation. The logic automatically adjusts the lane mapping for a flipped connector orientation.
9	R/O	1'b0	Reserved

Bits	Type	Reset	Description
8	R/W	1'b1	PMA lane 2 interface select - selects between PIPE PCS and PHY DP to drive PMA lane 2. 0 = PIPE PCS; 1 = PHY DP I/F.
7:6	R/W	2'd0	PHY DP lane selection - when PHY DP I/F is selected for PMA lane 1, this field selects which PHY DP lane drives PMA lane 1. (Only valid with bit [4] == 1.) When configured for DP, the same PHY DP lane can be mapped to only 1 PMA lane. This mapping is applicable for a normal connector orientation. The logic automatically adjusts the lane mapping for a flipped connector orientation.
5	R/O	1'b0	Reserved
4	R/W	1'b0	PMA lane 1 interface select - selects between PIPE PCS and PHY DP interface to drive PMA lane 1. 0 = PIPE PCS; 1 = PHY DP I/F.
3:2	R/W	2'd0	PHY DP lane selection - when PHY DP I/F is selected for PMA lane 0, this field selects which PHY DP lane drives PMA lane 0. (Only valid with bit [0] == 1.) When configured for DP, the same PHY DP lane can be mapped to only 1 PMA lane. This mapping is applicable for a normal connector orientation. The logic automatically adjusts the lane mapping for a flipped connector orientation.
1	R/O	1'b0	Reserved
0	R/W	1'b0	PMA lane 0 interface select - selects between PIPE PCS and PHY DP to drive PMA lane 0. 0 = PIPE PCS; 1 = PHY DP I/F. (Note: PHY DP I/F refers to PHY PMA transmit data interface, i.e. phy_pma_tx_data_ln_<>/phy_pma_tx_data_clk_in, and the internal DP configuration and control registers.

PHY_PIPE_CMN_CTRL1

Address: Operational Base + offset (18'b11_00_0_0_0_000_000001_00)

PIPE common control1 register

Bits	Type	Reset	Description
15:9	R/O	7'd0	Reserved

Bits	Type	Reset	Description
8	R/W	1'b1	**Comma realign**: This field controls the comma alignment state machine to re-align to new bit position without going to loss of sync state. The requirement of the new bit position should meet the number of COMMAs as per Symbol unlock count register definition. When new bit position is identified the comma alignment state machine remains in sync state with the alignment now locked to the new bit position. This field needs to be programmed during the PHY initialization routine before training sequences are received. The effect here is that pipe_rx_valid is not de-asserted upon re-alignment. When this bit is 0, pipe_rx_valid will be de-asserted upon loss of COMMA lock and subsequent re-alignment.
7:5	R/O	3'd0	Reserved
4	R/W	1'b1	**TX electrical idle pre release** : When this bit is set, the TX electrical idle release to the PMA is advanced 1 cycle to allow the adjustment of the datapath timing.
3	R/O	1'b0	Reserved
2	R/W	1'b0	**USB PIPE3 Compatibility Mode enable** : When this bit is set to 1, USB PIPE3 compatibility mode is enabled. In this mode, when operating in nominal empty Elasticity Buffer mode, when the EB buffer goes empty, instead of de-asserting PIPE RxDataValid, a USB SKIP OS is inserted into the data stream. This is the behavior as defined in PIPE version 3. When this bit is low, PIPE RxDataValid is de-asserted when the EB buffer goes empty, as recommended by PIPE version 4.
1	R/W	1'b0	**USB Loopback Slave Error Count disable** : When this bit is set to 1, the error count insertion for USB loopback slave is disabled, such that the error count is not inserted into the BCNT OS.
0	R/W	1'b1	**USB Elasticity Buffer Re-align enable** : When this bit is set to 1, when RX for a USB link is started, the elasticity buffer is re-aligned to its idle point upon seeing the 3 consective COMMAs in the same relative bit position (first instance only). This will re-align the elasticity buffer (i.e. CTC FIFO) after receiving the TSEQ data which contains no SKIP OSs.

PHY_PIPE_CMN_CTRL2
Address: Operational Base + offset (18'b11_00_0_0_0_000_000010_00)
PIPE common control2 register

Bits	Type	Reset	Description
15:12	R/W	4'd11	**USB SuperSpeed TX LFPS Stretch** : Minimum number of data rate clock cycles in which PMA tx_lfps_en signal is asserted. Number of data rate clock cycles must be > 1 PMA RefClk cycle.
11:0	R/O	12'd0	Reserved

PHY_PIPE_COM_LOCK_CFG1
Address: Operational Base + offset (18'b11_00_0_0_0_000_000011_00)
PIPE comma lock configuration1 register

Bits	Type	Reset	Description
15:12	R/W	4'h4	**Symbol unlock count:** The number of COMMA symbols that need to be seen in the wrong bit position before the comma alignment state machine will transition to RESYNC or LOS state
11:0	R/W	12'h400	**comma full lock count:** The number of COMMA symbols that need to be seen in the same bit position for the comma alignment state machine to lock. The field is used for initial reset lock.

PHY_PIPE_COM_LOCK_CFG2
Address: Operational Base + offset (18'b11_00_0_0_0_000_000100_00)
PIPE comma lock configuration2 register

Bits	Type	Reset	Description
15:8	R/O	8'd0	Reserved
7:0	R/W	8'h20	**comma lock count:** The number of COMMA symbols that needs to be seen in the same bit position for the comma state machine to lock. This field is used while the PCS is transitioning back to the P0 power state.

PHY_PIPE_RCV_DET_INH
Address: Operational Base + offset (18'b11_00_0_0_0_000_000101_00)
PIPE receiver detect inhibit register

Bits	Type	Reset	Description
15:0	R/W	16'h3d09	**Receiver Detect Inhibit Counter Load Value:** Counter load value to delay receiver detection request to PMA until PMA common mode is within the required range. The timer starts once the PHY signals ready by de-assertion of pipe_phystatus. If receiver detect request is received while timer has not expired, the PCS will wait until the timer expires before signaling the request to the PMA. Load value is specified in multiples of 128 ns with a default value of 2 ms. Note: Under normal operation the effect of this timer is transparent since the USB controller's LTSSM state machine will wait 12 ms in the eSS.Inactive.Quiet state before performing requesting a receiver detect operation.

PHY_USB_BER_CNT
Address: Operational Base + offset (18'b11_00_0_1_0_000_000000_00)
USB loopback slave BER count isolation register

Bits	Type	Reset	Description
15:8	R/O	8'd0	Reserved
7:0	R/O	8'd0	Current value of USB 30 loopback lsave Bit Error Count from the PCS. (Not re-synchronized to apb_pclk)

PHY_DP_MODE_CTL
Address: Operational Base + offset (18'b11_00_0_0_0_000_001000_00)
DP Mode Control register

Bits	Type	Reset	Description
15:12	R/W	4'b1100	PHY DP lane disable - 0 = enable associated PHY DP lane; 1 = disable/powerdown the associated PHY DP lane. This field is used to disable PHY lanes when not used. For example, for VESA DP Alt Mode pin assignments A, C and E, this field would be used to disable unused lanes for 1 or 2 lane DP configura- tions. Additionally, any PMA lanes mapped to the PHY DP lane will be disabled and powered down. ·[12] = PHY DP lane 0 ·[13] = PHY DP lane 1 ·[14] = PHY DP lane 2 ·[15] = PHY DP lane 3
11:9	R/O	3'd0	Reserved

Bits	Type	Reset	Description
8	R/W	2'b0	DP link reset - Reset for DP link. 0 = reset asserted, 1 = reset de-asserted. Clearing this bit places all of the DP configured PMA lanes into reset even when phy_reset_n is de-asserted. It is used to change the DP configuration (i.e. number of active lanes) when a PHY reset is not possible (i.e. due to simultaneous USB operation).
7:4	R/O	4'b0000	PHY DP Power State Acknowledgement - power state acknowledgement for PHY DP lanes. (Re-synchro- nized to APB clock.) After re-synchronization to APB clock, this is the AND of xcvr_power_state_ack[3:0] for each enabled HPHY DP lane.
3:0	R/W	4'b0000	PHY DP Power State - power state for PHY DP lanes. Direct mapping to the PMA's A0 to A3 power states (A0 = 0b0001, A1 = 0b0010, A2 = 0b0100 and A3 = 0b1000). (Re-synchronized to PSM clock.) Automatically, cleared to 0x0, upon [7:4] == [3:0], i.e. upon the completion of the power state change. This eliminates the need to write this register back to 0x0. The initial power state is set to A2. Note: The PMA has 2 other defined power states, A4 and A5. These will not be used by HPHY DP and are thus not provided in this register. Drives xcvr_power_state_req_ln_<>[3:0] (after re-synchronization to the PSM clock) for each enabled PHY DP lane.

PHY_DP_CLK_CTL
Address: Operational Base + offset (18'b11_00_0_0_0_000_001001_00)
DP Clock Control register

Bits	Type	Reset	Description
15:12	R/O	4'd1	DP PLL data rate 1 clock divider value. Divider value for the PLL clock to generate phy_pma_char_clk_out. (HBR2 = 1, RBR/HBR = 2.)
11:8	R/W	4'd2	DP PLL data rate 0 clock divider value. Divider value for the PLL clock to generate phy_pma_tx_data_clk_out. (HBR2 = 2, RBR/HBR = 4.)
7:4	R/O	4'd0	Reserved

Bits	Type	Reset	Description
3	R/O	1'b0	DP PLL clock enable acknowledge - Indicates whether DP PLL's data rate and full rate clocks are active/ enabled. 1 = clocks enabled/active, 0 = clocks disabled/gated.
2	R/W	1'b1	DP PLL clock enable - Clock enable for DP PLL's data rate and full rate clocks out of PMA. 1 = enable PLL data rate and full rate clocks, 0 = gate PLL data rate and full rate clocks
1	R/O	1'b0	DP PLL ready - DP PLL's ready indication for DP high speed clocks. 1 = PLL ready, 0 = PLL not ready.
0	R/W	1'b1	DP PLL enable - DP PLL's enable for DP high speed clocks. 1 = enabled, 0 = disabled.

PHY_STS
Address: Operational Base + offset (18'b11_00_0_0_0_000_001111_00)
PHY status register

Bits	Type	Reset	Description
15	RC	1'b0	**PHY APB access timeout**: When set, an APB read/write request to PHY registers failed (i.e. timed out). When set, this bit is cleared upon read.
14:0	R/O	15'd0	Reserved

PHY_DP_TX_CTL
Address: Operational Base + offset (18'b11_00_0_1_0_nnn_001000_00)
DP Lane Configuration register

Bits	Type	Reset	Description
15:6	R/O	10'd0	Reserved
5:4	R/W	2'd0	TX Voltage Level - Dxrives tx_vmargin PMA input for the mapped PMA lane (for functional and isolation modes). This field is used to set the DP Voltage Swing Level (0b00 = Level 0, 0b01 = Level 1, 0b10 = Level 2 and 0b11 = Level 3). TBD if Voltage Swing Level 3 supported.
3:2	R/O	2'd0	Reserved
1:0	R/W	2'd0	TX Deemphasis setting - Drives tx_deemphasis PMA input for the mapped PMA lane . This field is used to set the DP Pre-emphasis Level (0b00 = Level 0, 0b01 = Level 1, 0b10 = Level 2 and 0b11 = Level 3). TBD if Pre-emphasis Level 3 supported.

PHY_PMA_CMN_CTRL1
Address: Operational Base + offset (18'b11_00_1_0_0_000_000000_00)
PMA common control1 register

Bits	Type	Reset	Description
15:14	R/W	2'b00	Drives cmn_ref_clk_ana_div PMA input
13:12	R/W	2'b00	Drives cmn_ref_clk_dig_div PMA input
11:10	R/W	2'b00	Drives cmn_psm_clk_dig_div PMA input
9:7	R/O	3'd000	Reserved
6:4	R/W	3'b000	Drives cmn_ref_clk_sel PMA input
3	R/W	1'b0	Drives cmn_ref_clk_rcv_out_en PMA input
2	R/O	1'b0	Current value of cmn_macro_suspend_ack PMA output
1	R/O	1'b0	Current value of cmn_refclk_active PMA output
0	R/O	1'b0	Current value of cmn_ready pin PMA output

PHY_PMA_ISO_PLL_CTRL0
Address: Operational Base + offset (18'b11_00_1_0_0_000_010001_00)
PMA PLL control0 isolation register

Bits	Type	Reset	Description
15:4	R/O	12'd0	Reserved
3	R/W	1'd1	Drives cmn_pll1_clk_datart_en PMA input when in PMA isolation mode
2	R/W	1'd1	Drives cmn_pll0_clk_datart_en PMA input when in PMA isolation mode
1	R/W	1'd1	Drives cmn_pll1_en PMA input when in PMA isolation mode
0	R/W	1'd1	Drives cmn_pll0_en PMA input when in PMA isolation mode

PHY_PMA_ISO_PLL_CTRL1
Address: Operational Base + offset (18'b11_00_1_0_0_000_010010_00)
PMA PLL control1 isolation register

Bits	Type	Reset	Description
15:12	R/W	4'b0001	Drives cmn_pll1_clk_datart1_div PMA input when in PMA isolation mode
11:8	R/W	4'b0001	Drives cmn_pll1_clk_datart0_div PMA input when in PMA isolation mode
7:4	R/W	4'b0010	Drives cmn_pll0_clk_datart1_div PMA input when in PMA isolation mode
3:0	R/W	4'b0010	Drives cmn_pll0_clk_datart0_div PMA input when in PMA isolation mode

PHY_ISOLATION_CTRL
Address: Operational Base + offset (18'b11_00_1_0_0_000_011111_00)
Isolation control register

Bits	Type	Reset	Description
15	R/W	1'b0	PHY/PMA isolation enable (isolation_en) - When set, enables isolation (PHY or PMA).
14	R/W	1'b0	PHY/PMA common isolation enable (cmn_isolation_en) - When in PHY Macro Isolation Mode, the PHY common isolation register(s) are selected. When in PMA Isolation Mode, the PMA common isolation reg- ister(s) are selected.
13	R/O	1'b0	Reserved
12	R/W	1'b0	PHY/PMA isolation mode select (isolation_mode_sel) - When isolation_en is set, this bit selects between PHY isolation and PMA isolation mode. 0 = PHY isolation mode; 1 = PMA isolation mode.
11:8	R/O	4'd0	Reserved
7:0	R/W	8'd0	PHY/PMA lane isolation enable (ln_isolation_en) - When in PHY Macro Isolation Mode, the selected PHY lane(s) isolation registers are selected. When in PMA Isolation Mode, the selected PMA lane(s) isolation registers are selected.

PHY_PMA_ISO_XCVR_CTRL
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010000_00)
PMA Isolation Tansceiver control register

Bits	Type	Reset	Description
15	R/O	1'b0	Current value of xcvr_pll_clk_en_ack PMA output for the associated lane.
14	R/W	1'b0	Drives xcvr_pll_clk_en PMA input for the associated lane when in PMA isolation mode.
13	R/W	1'b0	Drives tx_lfps_en PMA input for the associated lane when in PMA isolation mode.
12	R/W	1'b0	Drives tx_elec_idle PMA input for the associated lane when in PMA isolation mode.
11	R/O	1'b0	Current value of xcvr_psm_ready PMA output for the associated lane.
10	R/O	1'b0	Current value of tx_rcv_detected PMA output for the associated lane.
9	R/O	1'b0	Current value of tx_rcv_detect_done PMA ouptut for the associated lane.
8	R/W	1'b0	Drives tx_rcv_detect_en PMA input for the associated lane when in PMA isolation mode.

Bits	Type	Reset	Description
7	R/W	1'b0	Drives rx_eq_training_data_valid_ln_{nnnn} PMA input for the associated lane when in PMA isolation mode. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.
6	R/W	1'b0	Drives rx_eq_training PMA input for the associated lane when in PMA isolation mode. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.
5	R/W	1'b0	Drives xcvr_link_reset_n PMA input for the associated lane when in PMA isolation mode.
4	R/W	1'b0	Drives xcvr_lane_suspend PMA input for the associated lane when in PMA isolation mode.
3	R/O	1'b0	Current value of rx_lfps_detect PMA output for the assocaited lane. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.
2	R/O	1'b0	Current value of rx_signal_detect PMA output for the associated lane. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.
1	R/W	1'b0	Drives rx_termination PMA input for the associated lane when in PMA isolation mode. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.
0	R/W	1'b0	Drives xcvr_lane_en PMA input for the associated lane when in PMA isolation mode.

PHY_PMA_ISO_TX_CFG
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010001_00)
PMA TX configuration register

Bits	Type	Reset	Description
15:14	R/O	2'd0	Reserved
13:12	R/W	2'd0	Drives tx_deemphasis PMA input for the associated lane when in PMA isolation mode
11:9	R/O	3'd0	Reserved
8	R/W	1'b0	Drives tx_low_power_swing_en PMA input for the associated lane.
7:3	R/O	5'd0	Reserved
2:0	R/W	3'd0	Drives tx_vmargin PMA input for the associated lane.

PHY_PMA_ISO_LINK_MODE
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010010_00)
PMA Isolation mode control register

Bits	Type	Reset	Description
15	R/W	1'b1	tx_reset_n PMA input for the associated lane when in PMA isolation mode.
14	R/W	1'b1	rx_reset_n PMA input for the associated lane when in PMA isolation mode. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.
13	R/O	1'b0	Reserved
12	R/W	1'b0	Drives the tx_high_z PMA input for the associated lane when in PMA isolation mode
11:6	R/O	6'd0	Reserved
5:4	R/W	2'd0	Drives xcvr_standard_mode PMA input for the associated lane when in PMA isolation mode.
3	R/O	1'b0	Reserved
2:0	R/W	3'd0	Drives xcvr_data_width PMA input for the associated lane when in PMA isolation mode.

PHY_PMA_ISO_PWRST_CTRL
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010011_00)
PMA Isolation power state control register

Bits	Type	Reset	Description
15	R/W	1'b0	Drives rx_sig_det_en_ext_} PMA input for the associated lane when in PMA isolation mode. Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved. (Used for PCIe)
14	R/W	1'b0	Drives tx_cmn_mode_en_ext PMA input for the associated lanewhen in PMA isolation mode. (Used for PCIe)
13:8	R/O	6'h00	Current value of xcvr_power_state_ack PMA output for the associated lane.
7:6	R/O	2'b00	Reserved
5:0	R/W	6'h00	Drives xcvr_power_state_req PMA input for the associated lane when in PMA isolation mode.

PHY_PMA_ISO_TX_DATA_LO
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010100_00)
PMA transmit low data isolation register

Bits	Type	Reset	Description
15:0	R/W	16'h0000	Drives tx_td[15:0] PMA input for the associated lane when in PMA isolation mode. (Not re-synchronized to apb_pclk).

PHY_PMA_ISO_TX_DATA_HI
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010101_00)
PMA transmit high data isolation register

Bits	Type	Reset	Description
15:4	R/O	12'h000	Reserved
3:0	R/W	4'h0	Drives tx_td[19:16] PMA input for the associated lane when in PMA isolation mode. (Not re-synchronized to apb_pclk).

PHY_PMA_ISO_RX_DATA_LO
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010110_00)
PMA receive low data isolation register

Bits	Type	Reset	Description
15:0	R/O	16'h0000	Current value of rx_rd[15:0] PMA output for the associated lane. (Not re-synchronized to apb_pclk). Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.

PHY_PMA_ISO_RX_DATA_HI
Address: Operational Base + offset (18'b11_00_1_1_0_nnn_010111_00)
PMA receive high data isolation register

Bits	Type	Reset	Description
15:4	R/O	12'h000	Reserved
3:0	R/O	4'h0	Drives rx_rd[19:16] PMA input for the associated lane. (Not re-synchronized to apb_pclk). Valid for PMA lanes 2 and 3 only. For PMA lanes 0 and 3, reserved.

18.5 Interface Description

Table 18-1 TypeC PHY Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
tx_p_ln_0	O	IO_UPHY0_TCPHY_TXP0	TypeC PHY0 differential lane
tx_m_ln_0	O	IO_UPHY0_TCPHY_TXM0	TypeC PHY0 differential lane

Module Pin	Direction	Pad Name	IOMUX Setting
tx_rx_p_ln_1	I/O	IO_UPHY0_TCPHY_TXRXP1	TypeC PHY0 differential lane
tx_rx_m_ln_1	I/O	IO_UPHY0_TCPHY_TXRXM1	TypeC PHY0 differential lane
tx_rx_p_ln_2	I/O	IO_UPHY0_TCPHY_TXRXP2	TypeC PHY0 differential lane
tx_rx_m_ln_2	I/O	IO_UPHY0_TCPHY_TXRXM2	TypeC PHY0 differential lane
tx_p_ln_3	O	IO_UPHY0_TCPHY_TXP3	TypeC PHY0 differential lane
tx_m_ln_3	O	IO_UPHY0_TCPHY_TXM3	TypeC PHY0 differential lane
aux_p	I/O	IO_UPHY0_TCPHY_AUXP	TypeC PHY0 differential lane
aux_m	I/O	IO_UPHY0_TCPHY_AUXM	TypeC PHY0 differential lane
aux_p_pd_pu	O	IO_UPHY0_TCPHY_AUXP_PD_PU	TypeC PHY0 aux_p external pull up/down resistor polarity reversal pad
aux_m_pu_pd	O	IO_UPHY0_TCPHY_AUXM_PU_PD	TypeC PHY0 aux_m external pull up/down resistor polarity reversal pad
rext	I	IO_UPHY0_TCPHY_REXT	TypeC PHY0 external calibration resistor
tx_p_ln_0	O	IO_UPHY1_TCPHY_TXP0	TypeC PHY1 differential lane
tx_m_ln_0	O	IO_UPHY1_TCPHY_TXM0	TypeC PHY1 differential lane
tx_rx_p_ln_1	I/O	IO_UPHY1_TCPHY_TXRXP1	TypeC PHY1 differential lane
tx_rx_m_ln_1	I/O	IO_UPHY1_TCPHY_TXRXM1	TypeC PHY1 differential lane
tx_rx_p_ln_2	I/O	IO_UPHY1_TCPHY_TXRXP2	TypeC PHY1 differential lane
tx_rx_m_ln_2	I/O	IO_UPHY1_TCPHY_TXRXM2	TypeC PHY1 differential lane
tx_p_ln_3	O	IO_UPHY1_TCPHY_TXP3	TypeC PHY1 differential lane
tx_m_ln_3	O	IO_UPHY1_TCPHY_TXM3	TypeC PHY1 differential lane
aux_p	I/O	IO_UPHY1_TCPHY_AUXP	TypeC PHY1 differential lane

Module Pin	Direction	Pad Name	IOMUX Setting
aux_m	I/O	IO_UPHY1_TCPHY_AUXM	TypeC PHY1 differential lane
aux_p_pd_pu	O	IO_UPHY1_TCPHY_AUXP_PD_PU	TypeC PHY1 aux_p external pull up/down resistor polarity reversal pad
aux_m_pu_pd	O	IO_UPHY1_TCPHY_AUXM_PU_PD	TypeC PHY1 aux_m external pull up/down resistor polarity reversal pad
rext	I	IO_UPHY1_TCPHY_REXT	TypeC PHY1 external calibration resistor

18.6 Application Notes

18.6.1 Start-up Sequence and PHY Disable

1. USB Start-up Sequence:

a. Select external PSM clock (see Chapter GRF)
b. Set select TypeC PHY0 or TypeC PHY1 used for DPTX
c. Reset whole TypeC PHY, assert apb_preset_n, pipe_reset_n and phy_reset_n (see Chapter CRU)
d. Release apb_preset_n
e. Configurate TypeC PHY normal or flipped orientation
f. Configurate PHY and PMA for the selected mode of operation
g. Release phy_reset_n
h. Wait for CMN ready indication (assertion) by polling bit 0 of PHY_PMA_CNN_CTRL1 of PHY
i. Release pipe_reset_n
j. Wait for the de-assertion of pipe_phy_status, then TypeC PHY for USB operation is ready.

Note: It must hold whole USB3.0 OTG controller in resetting to hold pipe power state in P2 before initializing TypeC PHY.

2. DisplayPort Start-up Sequence:

a. Step a~f is same as USB Start-up Sequence
b. Set Register PHY_DP_MODE_CTL[8] to 1 and PHY_DP_MODE_CTL[3:0] to 0x4
c. Release phy_reset_n before or after step b
d. Wait for DP lanes for ready (PHY_DP_MODE_CTL[6] assertion), then TypeC PHY for DP operation is ready.

Note: Only after the phy_data_clk feeding back from DPTX is enabled, TypeC PHY for DP operation may be ready.

3. PHY Disable for Low Power:

a. Assert pipe_reset_n to put USB part of TypeC PHY into lowest power state when USB function is not used.
b. Clear Register PHY_DP_MODE_CTL[8] to 0 to put DP part of TypeC PHY into lowest power state when DP function is not used.

18.6.2 DP Data Rate and/or Lane Configuration Change

The procedure for changing the PHY's power state is:

a. Write PHY_DP_MODE_CTL[3:0] with 0'b1000. (Place the PHY lanes in the A3 power state.)
b. Wait for PHY_DP_MODE_CTL[7:4] == 0'b1000.
c. Clear PHY_DP_CLK_CTL[2]. Gate the PLL clocks from PMA.
d. Wait for PHY_DP_CLK_CTL[3] == 0.

e. Clear PHY_DP_CLK_CTL[0]. Disable the PLL.
f. Wait for PHY_DP_CLK_CTL[1] == 0.
g. Re-configure PHY registers for the new data rate (as defined in the programmer phy's guide) and/or DP lane configuration (enable/disable DP lanes)
h. Set PHY_DP_CLK_CTL[0]. Enable the PLL.
i. Wait for PHY_DP_CLK_CTL[1] == 1.
j. Set PHY_DP_CLK_CTL[2]. Enable PMA PLL clocks.
k. Wait for PHY_DP_CLK_CTL[3] == 1.
l. Write PHY_DP_MODE_CTL[3:0] with 0'b0100 (A2 power state). The PMA must go through the A2 power state upon a data rate change.
m. Wait for PHY_DP_MODE_CTL[7:4] == 0'b0100.
n. As required, change the PHY power state to A0.

18.6.3 Lane Mapping

For the TypeC PHY, the 4 lanes are mapping to the USB TypeC receptacle pins as follows:

Table 18-2 TypeC Recptacle Pins Mapping

PHY Lanes/Module Pins	TypeC Receptacle Pins
Lane0 (tx_p/m_ln_0)	TX1+/TX1- (pins A2/A3)
Lane1 (tx_rx_p/m_ln_1)	RX1+/RX1- (pins B11/B10)
Lane2 (tx_rx_p/m_ln_2)	RX2+/RX2- (pins A11/A10)
Lane3 (tx_p/m_ln_3)	TX2+/TX2- (pins B2/B3)

USB and DP lanes mapping to TypeC PHY lanes for each of pin assignment options (normal and flipped connector orientation) described in the VESA DisplayPort Alt Mode on USB TypeC Standard as follows:

Table 18-3 TypeC PHY Lanes Mapping for Normal Orientation

PHY Lanes	A	B	C	D	E	F
0	DP lane 1	USB TX	DP lane 2	USB TX	DP lane 2	USB TX
1	DP lane 3	USB RX	DP lane 3	USB RX	DP lane 3	USB RX
2	DP lane 2	DP lane 1	DP lane 0	DP lane 0	DP lane 0	DP lane 1
3	DP lane 0	DP lane 0	DP lane 1	DP lane 1	DP lane 1	DP lane 0

Table 18-4 TypeC PHY Lanes Mapping for Flipped Orientation

PHY Lanes	A	B	C	D	E	F
0	DP lane 0	DP lane 0	DP lane 1	DP lane 1	DP lane 1	DP lane 1
1	DP lane 2	DP lane 1	DP lane 0	DP lane 0	DP lane 0	DP lane 0
2	DP lane 3	USB RX	DP lane 3	USB RX	DP lane 3	USB RX
3	DP lane 1	USB TX	DP lane 2	USB TX	DP lane 2	USB TX

The PHY_PMA_LANE_CFG register is used to select whether a PMA lane is mapped for USB or PHY DP. The default configuration is for PHY lanes 0 and 1 to be mapped for USB (PHY lane 0 -> USB TX and PHY lane 1 -> USB RX) and PHY lanes 2 and 3 to be mapped for DP (PHY lane 2 -> DP lane 0 and PHY lane 3 -> DP lane 1). This maps to VESA DP Alt Mode pin assignments D and F. The PHY_PMA_LANE_CFG register is configured based on a normal connector orientation. Logic in the PHY automatically handles the flipped connector case based on the setting of orientation of TypeC PHY.

For pin assignments A, B, C and E, PHY_PMA_LANE_CFG must be updated as described below prior to de-asserting phy_reset_n:
Pin assignment A: PHY_PMA_LANE_CFG = 0x19D5
Pin assignment B: PHY_PMA_LANE_CFG = 0x1500
Pin assignment C: PHY_PMA_LANE_CFG = 0x9D15
Pin assignment E: PHY_PMA_LANE_CFG = 0x9D15
In cases where fewer than the configured number of DP lanes are being used PHY_DP_MODE_CTL[15:12] must be set to disable and power-down the unused PHY DP lanes (and their mapped PMA lanes). Set the bit ([15:12]) associated with each DP PHY lane(s) to be disabled.

Chapter 19 SAR-ADC

19.1 Overview

The ADC is a 6-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 1MSPS with 13MHz A/D converter clock.

19.2 Block Diagram

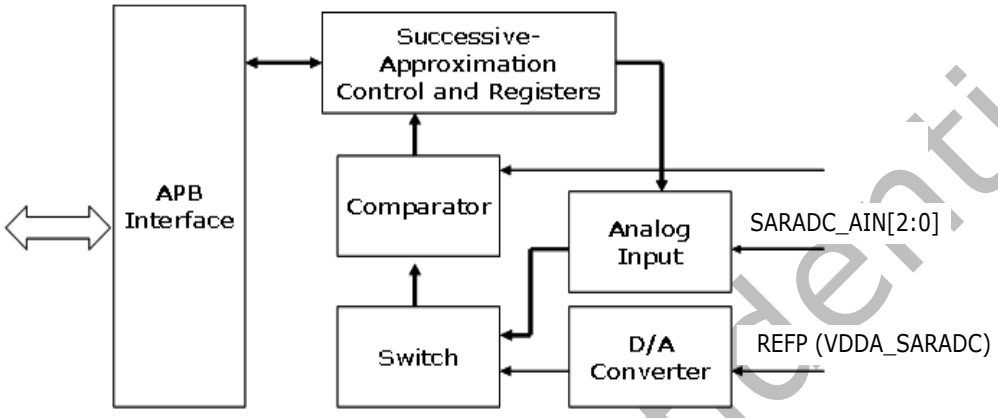


Fig. 19-1 RK3399SAR-ADC block diagram

Successive-Approximate Register and Control Logic Block
This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.
Comparator Block
This block compares the analog input SARADC_AIN[2:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

19.3 Function Description

19.3.1 APB Interface

In RK3399, SAR-ADC works at single-sample operation mode.
This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC. The detailed timing diagram will be shown in the following.

19.4 Register description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.

Name	Offset	Size	Reset Value	Description
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

19.4.2 Detail Register Description

SARADC_DATA
Address: Operational Base + offset (0x0000)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS
Address: Operational Base + offset (0x0004)
The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress

SARADC_CTRL
Address: Operational Base + offset (0x0008)
The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable. 0: Disable 1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down; 1: ADC power up and reset. start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

Bit	Attr	Reset Value	Description
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 000 : Input source 0 (SARADC_AIN[0]) 001 : Input source 1 (SARADC_AIN[1]) 010 : Input source 2 (SARADC_AIN[2]) 011 : Input source 3 (SARADC_AIN[3]) 100 : Input source 4 (SARADC_AIN[4]) 101 : Input source 5 (SARADC_AIN[5]) Others : Reserved

SARADC_DLY_PU_SOC
Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	DLY_PU_SOC delay between power up and start command The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

19.5 Timing Diagram

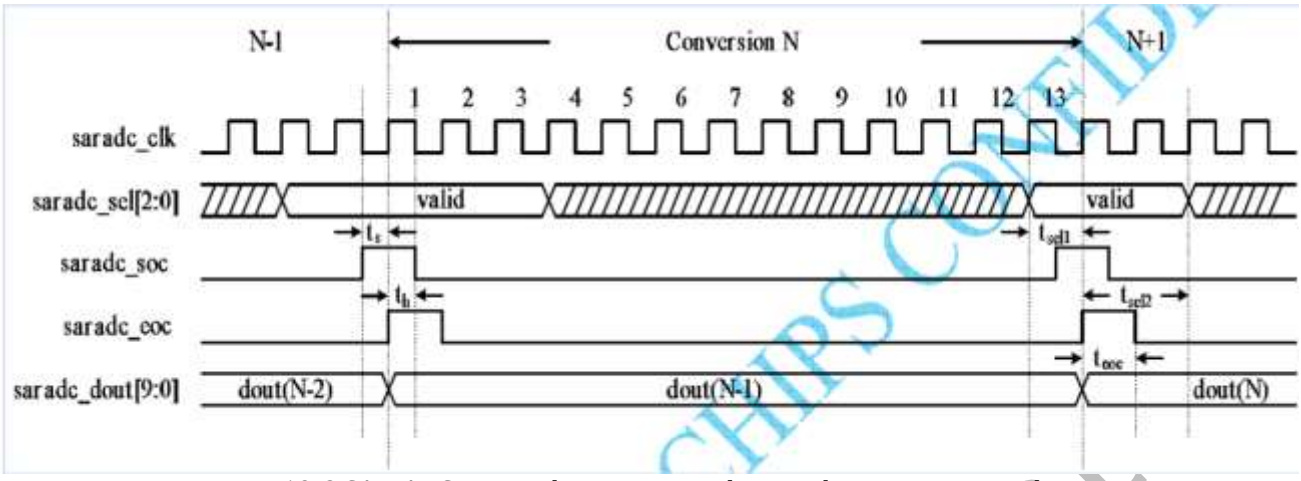


Fig. 19-2 SAR-ADC timing diagram in single-sample conversion mode

The following table has shows the detailed value for timing parameters in the above diagram.

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Condition						
Analog Supply	AVDD		1.62	1.8	1.98	V
Digital Supply	VDD		0.81	0.90	0.99	V
Junction Temperature	T_J		-40		125	℃
SARADC Performance						
Resolution				10		bit
Effective Number of Bit	ENOB			9		bit
Differential Nonlinearity	DNL		-1		+1	LSB
Integral Nonlinearity	INL		-2		+2	LSB
Input Voltage Range	V_{IN}		0		1	AVDD
Input Capacitance	C_{IN}			10		pF
Sampling Rate	f_S				1	MS/s
Spurious Free Dynamic Range	SFDR	f_S=1MS/s f_{OUT}=1.17KHz		61		dB
Signal to Noise and Harmonic Ratio	SNDR			56		dB
Timing Characteristic						
Clock Frequency	f_{CLK}				13	MHz
Clock Period	t_{CLK}		75			ns
Clock Duty Cycle			45		55	%
Conversion Time			13			t_{CLK}
Setup Time of soc signal	t_s			0.5		t_{CLK}
Hold Time of soc signal	t_h			0.5		t_{CLK}
Time Interval between Transition of sel[2:0] and Rising Edge of 1st clock	t_{sel1}		1			t_{CLK}
Time Interval between Transition of sel[2:0] and Rising Edge of 1st clock	t_{sel2}		2			t_{CLK}
High Level Time of eoc signal	t_{eoc}		1			t_{CLK}
Power Consumption						
Analog Supply Current	I_{AVDD}	f_S=1MS/s		450		uA
		Power Down		1		uA
Digital Supply Current	I_{VDD}	f_S=1MS/s		50		uA
		Power Down		1		uA

Fig. 19-3 RK3399 SAR-ADC timing parameters list

19.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down adc converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed
- Read the conversion result from SARADC_DATA[9:0]
- Note: The A/D converter was designed to operate at maximum 1MHZ.