TH1520 Memory Interface
User Manual

Revision 1.0.0

Security Secret

Date 2023-08-26

平头哥（上海）半导体技术有限公司 T-HEAD (Shanghai) Semiconductor Co., LTD

Address: 5th Floor Number 2 Chuan He Road 55, Number 366 Shang Ke Road, Shanghai free trade area, China
Website: www.t-head.cn

平头哥（上海）半导体技术有限公司 T-HEAD (Shanghai) Semiconductor Co., LTD

地址： 中国（上海）自由贸易试验区上科路 366 号、川和路 55 弄 2 号 5 层
网址： www.t-head.cn

Revisions

Rev	Description	Author(s)	Date
V1.0.0	Initial version	T-Head	2023-08-26

Contents

Figures & Tables

List of Abbreviations

Abbreviations	Full Spelling	Chinese Explanation
ADMA	Advanced Direct Memory Access	先进的直接内存访问
CAM	Content Addressable Memory	内容可寻址存储器
DDR	Double Data Rate SDRAM	双倍速率同步动态随机存储器
DMA	Direct Memory Access	直接内存访问
eMMC	Embedded Multi Media Card	嵌入式多媒体卡
MSHC	Mobile Storage Host Controller	移动存储主控
QoS	Quality of Service	服务质量
SD Card	Secure Digital Card	安全数字卡
SDIO	Secure Digital Input and Output	安全数字输入输出

1 SRAMC

By default, the chip provides 1.5MB +0.5MB on-chip SRAM for software expansion. The last 0.5MB address space can be used only when the DSP reset is released and DSP is not working.

2 LPDDR4

2.1 Overview

This document describes the initialization steps and precautions when using the DDR subsystem. The DDR subsystem consists of a DDR controller (uMCTL2) and two independent PHYs, as shown in Figure & Table 2-1. The DDRC is responsible for the configuration of logic layers such as arbitration, QoS, and address mapping. The PHYs convert DFI logic signals into LPDDR4 storage control signals and contain PMU, which can independently complete external particle training and initialization.

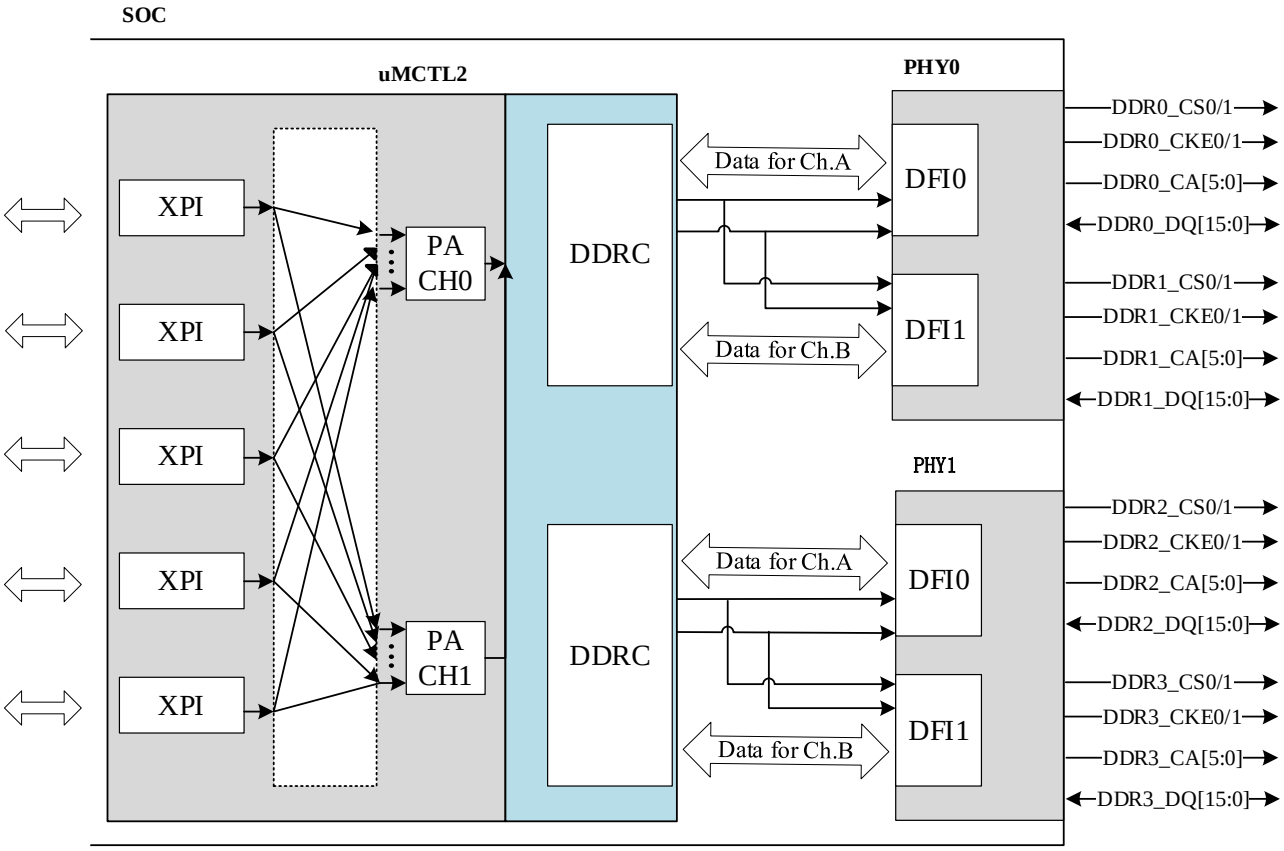

Figure & Table 2-1 DDR subsystem block diagram

The DDRC interface can receive 5 AXI inputs and supports the standard AMBA AXI4 protocol.
The PHY interface supports up to 64-bit LPDDR4 or LPDDR4x, and the maximum rate is 4267Mbps.

2.2 Main Features

The built-in DDRC (uMCTL2) and two independent PHYs realize the data access to the external memory LPDDR4 SDRAM by the master device such as CPU in the SoC system. The following features are supported:

- Supports the standard JEDE LPDDR4/LPDDR4x, the maximum bit rate is 4267Mbps.

- Supports software-triggered ZQ calibration and hardware-based ZQ calibration at fixed intervals.
- Supports dual-channel interleaved 64-bit mode.
- Supports single-channel 32-bit mode.
- Supports software-configured DQ/CA bit exchange.
- Supports periodic delay unit training to compensate for changes in voltage and temperature.
- PHY built-in independent training program:
  - The PHY master interface supports periodic training for LPDDR4.
  - CA bus training
  - Write/read training
  - Reference voltage training for read and write
- Testability features:
  - Address data channel supports internal loopback self-test
  - Self-test of delay unit
  - PLL lock test
  - ZQ calibration test

2.3 Interface

The DDR subsystem supports the standard JEDEC LPDDR4 protocol and is compatible with LPDDR4x. It supports 64-bit dual channel mode and 32-bit single channel mode. The specific modes are as follows:

- 64-bit LPDDR4/LPDDR4x, capacity 4GB/8GB

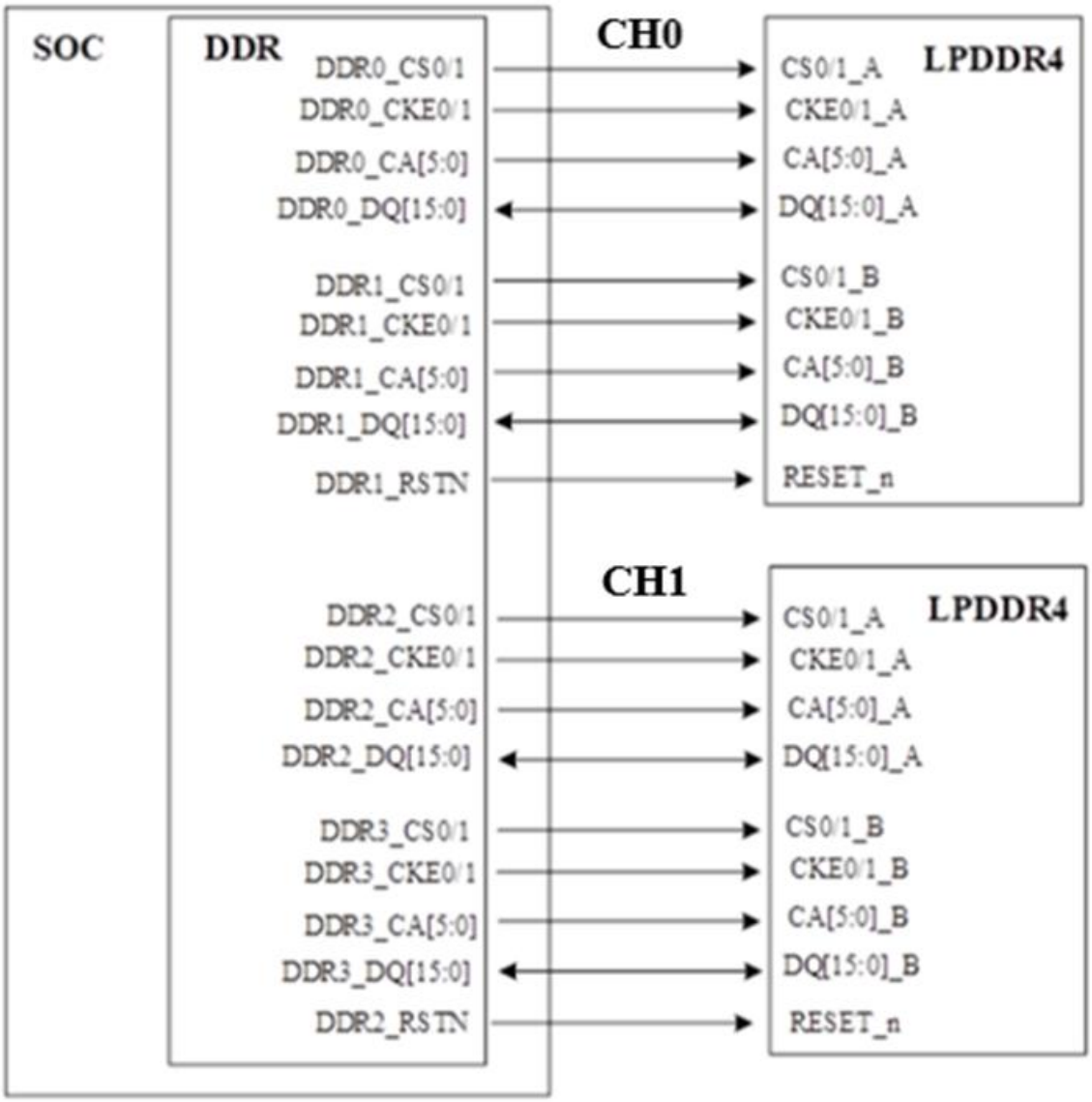

Figure & Table 2-2 64-bit dual channel connection diagram

- 32-bit LPDDR4/LPDDR4x, capacity 2GB/4GB

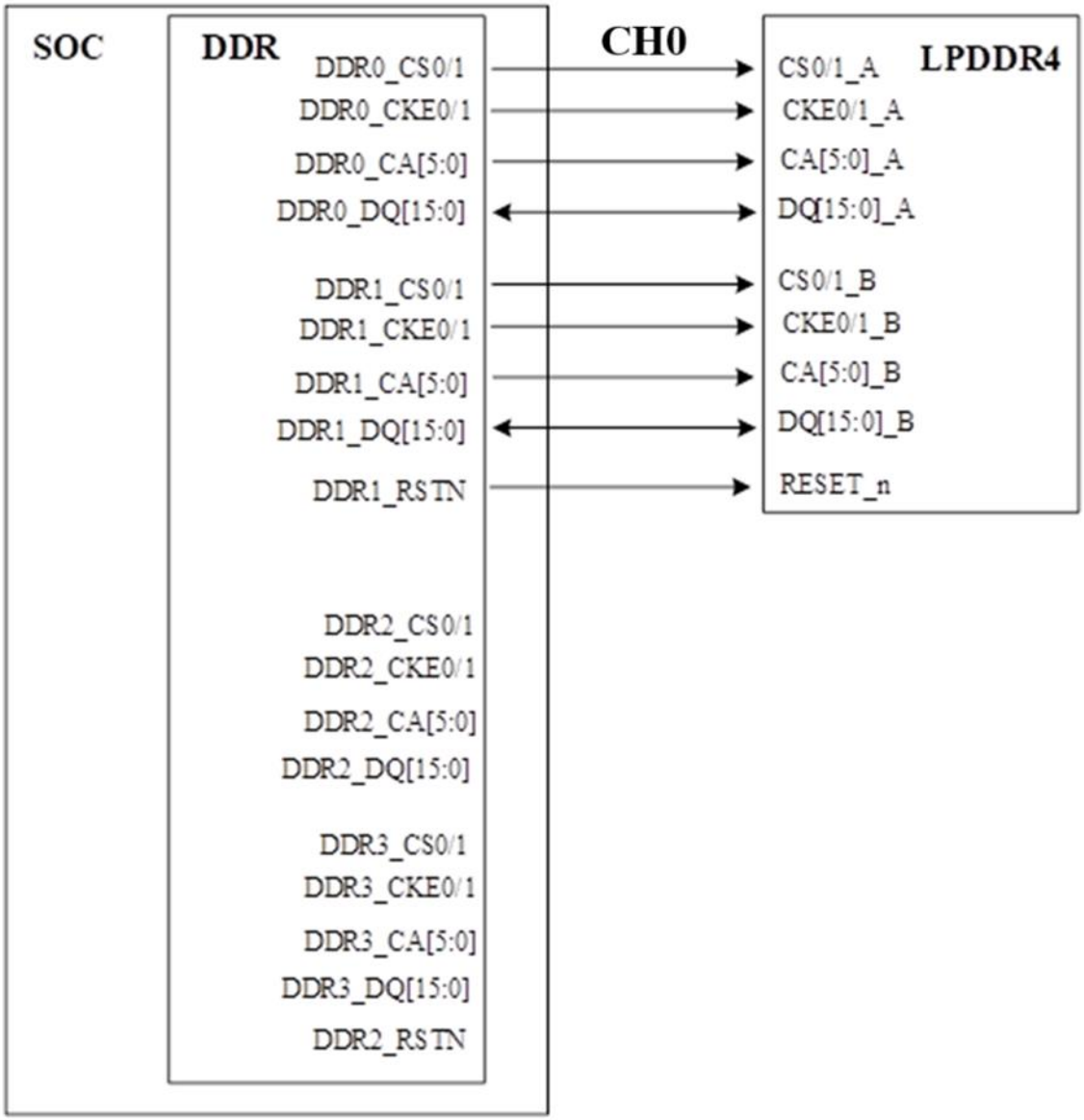

Figure & Table 2-3 32-bit single channel connection diagram

2.3.1 Pin Description

Figure & Table 2-4 Pin description table

Pin Name	Direction	Width	Description
DDR0_CKT DDR0_CKC	O	2	Differential clock of channel 0 lower 16 bits
DDR0_CA0~5	O	6	Command address bus of channel 0 lower 16 bits
DDR0_CS0~1	O	2	CS of channel 0 lower 16 bits
DDR0_CKE0~CKE1	O	2	Clock enable of channel 0 lower 16 bits

Pin Name	Direction	Width	Description
DDR0_DQSC0 DDR0_DQST0	IO	2	Bidirectional differential data select of channel 0 bit1~7
DDR0_DQSC1 DDR0_DQST1	IO	2	Bidirectional differential data select of channel 0 bit8~15
DDR0_DQ0~15	IO	16	Data channel of channel 0 lower 16 bits
DDR0_DMI0~1	IO	2	Data mask/inverse of channel 0 lower 16 bits
DDR1_CKT DDR1_CKC	O	2	Differential clock of channel 0 upper 16 bits
DDR1_CA0~5	O	6	Command address bus of channel 0 upper 16 bits
DDR1_CS0~1	O	2	CS of channel 0 upper 16 bits
DDR1_CKE0~CKE1	O	2	Clock enable of channel 0 upper 16 bits
DDR1_DQSC0 DDR1_DQST0	IO	2	Bidirectional differential data select of channel 0 bit16~23
DDR1_DQSC1 DDR1_DQST1	IO	2	Bidirectional differential data select of channel 0 bit24~31
DDR1_DQ0~15	IO	16	Data channel of channel 0 upper 16 bits
DDR1_DMI0~1	IO	2	Data mask/inverse of channel 0 upper 16 bits
DDR1_RSTN	O	1	Reset signal of channel 0, active low
DDR1_VREF	O	1	Reference voltage signal of channel 0, which can be used to monitor internal signals
DDR1_ZN	O	1	Reference resistance of channel 0, external 120ohm resistance, accuracy +/-1%
DDR2_CKT DDR2_CKC	O	2	Differential clock of channel 1 lower 16 bits
DDR2_CA0~5	O	6	Command address bus of channel 1 lower 16 bits
DDR2_CS0~1	O	2	CS of channel 1 lower 16 bits
DDR2_CKE0~CKE1	O	2	Clock enable of channel 1 lower 16 bits
DDR2_DQSC0 DDR2_DQST0	IO	2	Bidirectional differential data select of channel 1 bit0~7
DDR2_DQSC1 DDR2_DQST1	IO	2	Bidirectional differential data select of channel 1 bit8~15

Pin Name	Direction	Width	Description
DDR2_DQ0~15	IO	16	Data channel of channel 1 lower 16 bits
DDR2_DMI0~1	IO	2	Data mask/inverse of channel 1 lower 16 bits
DDR3_CKT DDR3_CKC	O	2	Differential clock of channel 1 upper 16 bits
DDR3_CA0~5	O	6	Command address bus of channel 1 upper 16 bits
DDR3_CS0~1	O	2	CS of channel 1 upper 16 bits
DDR3_CKE0~CKE1	O	2	Clock enable of channel 1 upper 16 bits
DDR3_DQSC0 DDR3_DQST0	IO	2	Bidirectional differential data select of channel 1 bit16~23
DDR3_DQSC1 DDR3_DQST1	IO	2	Bidirectional differential data select of channel 1 bit24~31
DDR3_DQ0~15	IO	16	Data channel of channel 1 upper 16 bits
DDR3_DMI0~1	IO	2	Data mask/inverse of channel 1 upper 16 bits
DDR2_RSTN	O	1	Reset signal of channel 1, active low
DDR2_VREF	O	1	Reference voltage signal of channel 1, which can be used to monitor internal signals
DDR2_ZN	O	1	Reference resistance of channel 1, external 120ohm resistance, accuracy +/-1%

2.3.2 Electrical Parameters

Refer to the JEDEC protocol description.

2.4 Function Description

2.4.1 Overview

2.4.1.1 uMCTL2

The uMCTL2 is responsible for converting the AXI protocol into the DFI protocol. It consists of the interleaver and two DDRCs, as shown in Figure & Table 2-5.

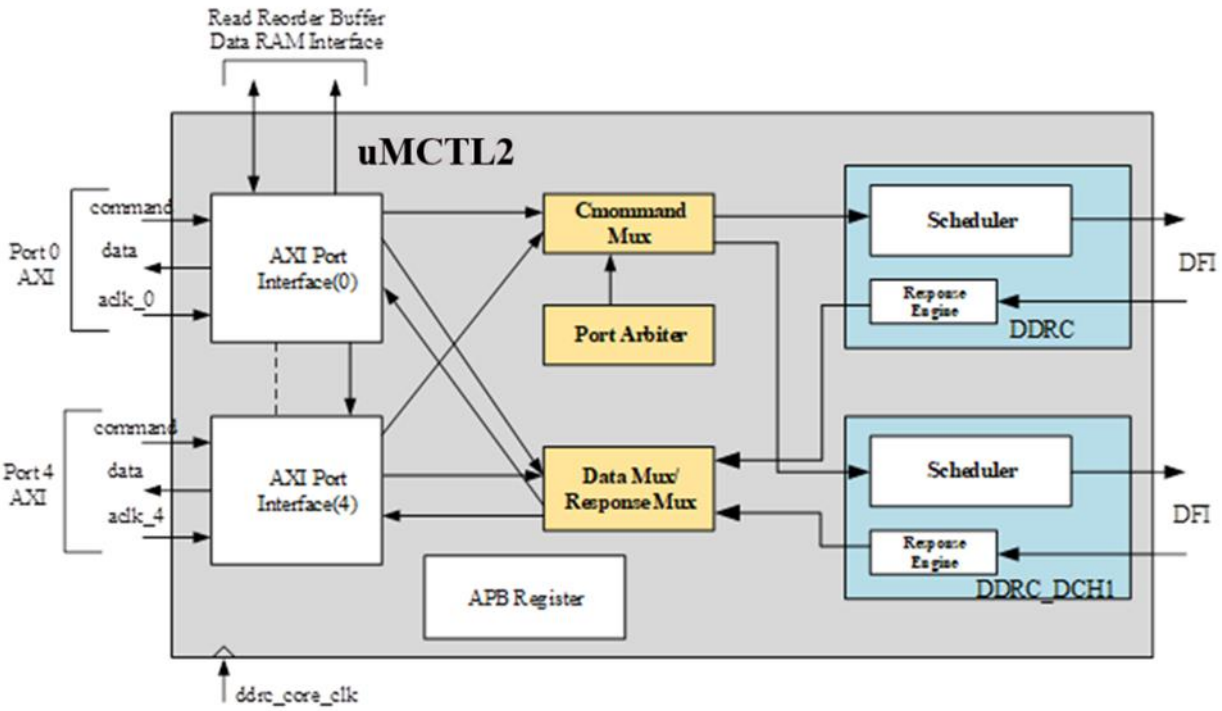

Figure & Table 2-5 Diagram of the uMCTL2

The uMCTL2 has the following main modules:

- AXI port interface: This module provides a standard AXI interface. It is responsible for AXI data sending and receiving, address processing and reordering of read data.
- Port arbiter: This module provides arbitration mechanism for each port.
- DDRC: Built-in Content Addressable Memory (CAM), used to store the received commands. Schedules according to row, column and bank addresses and converts the AXI protocol into the DFI protocol and sends to PHY.
- APB register: Used to configure registers for DDRC.

2.4.1.2 PHY

The PHY consists of a soft core PUB and a hardened module. The hardened module consists of four DBYTEs, six AXI4s and one Master. The structure is shown in Figure & Table 2-6.

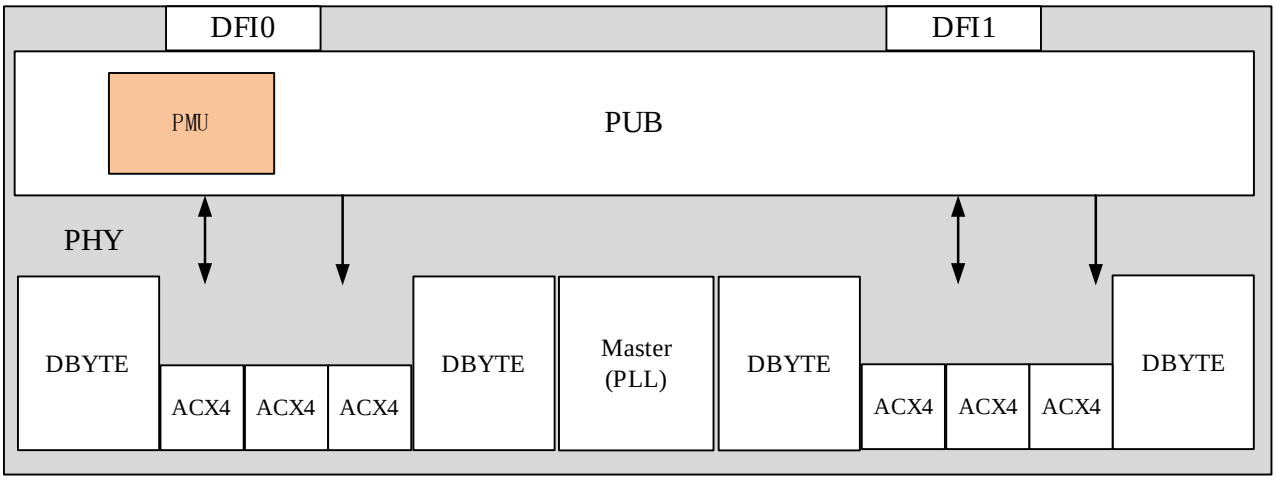

Figure & Table 2-6 Diagram of the PHY

- PUB: Built-in PMU, responsible for DDR training and initialization.
- DBYTE: Connects to the 1byte data interface of SDRAM.
- ACX4: Connects to the 4-way address/command interface of SDRAM.
- Master: Built-in PLL, providing clock source.

2.4.2 Address Mapping

2.4.2.1 Channel Interleaving

To improve schedule efficiency and system bandwidth performance, the DDR subsystem adopts a dual channel interleaved architecture. Users can implement interleaving with different granularities according to actual needs. The current system defaults to setting bit9 of the system address as the interleaving bit, that is, users can interleaving access different channels every 512B. In 32-bit single channel mode, users can access only channel 0 by configuring the address map register, and only the 32-bit DDR is connected externally.

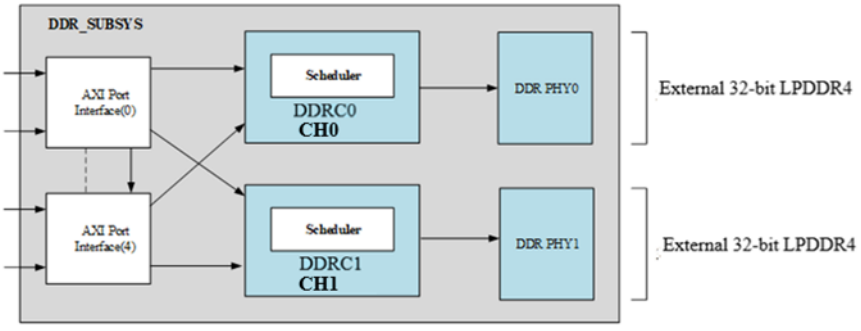

Figure & Table 2-7 Dual channel architecture

2.4.2.2 Address Mapping Configuration

Different mapping methods are implemented by configuring registers ADDRMAP0~11. The recommended groups of configurations are as follows:

Figure & Table 2-8 Address mapping description

Mode	Row Address Width	Column Address Width	Rank Address	Row Address	Channel Address	Bank Address	Column Address	
	16	10	1	[15:0]	1	[2:0]	[9:7]	[6:0]
8GB 64bit	16	10	[32]	[31:16]	[9]	[15:13]	[12:10]	[8:2]
4GB 32bit	16	10	[31]	[30:15]	-	[14:12]	[11:9]	[8:2]
4GB 64bit	16	10	-	[31:16]	[9]	[15:13]	[12:10]	[8:2]
2GB 32bit	16	10	-	[30:15]	-	[14:12]	[11:9]	[8:2]

2.4.3 Auto Refresh Mode

When RFSHCTL3.dis_auto_refresh is configured to a value other than 0, the DDRC will automatically generate refresh instructions periodically. In addition, the DDRC supports predictive refresh. When the refresh instruction is held more than one time, the controller can perform predictive refresh. That is, when the CAM does not have transaction of the corresponding rank/bank, the controller will automatically insert the corresponding refresh instruction. The RFSHCTL0.refresh_to_x1_x32 register determines how long the corresponding transaction is not in the CAM before inserting the predictive refresh. Every time a predictive refresh is performed, tREFI consumption count is reduced by one time, so it will buy time for the critical refresh and ensure that the number of predictive refreshes will not exceed that of refreshes that should be.

After setting RFSHCTL0.per_bank_refresh=1, per-bank refresh is enabled instead of all-bank refresh. After enabling per-bank refresh, you need to set tREFIpb and tRFCpb to the corresponding registers RFSHTMG.t_rfc_nom_x1_x32 and RFSHTMG.t_rfc_min. In this mode, the controller will schedule transactions to other banks according to the currently refreshed bank to improve potential performance. To improve the efficiency of scheduling, you need to set RFSHTMG.t_trfc_nom_x1_sel=1. LPDDR4 per-bank refresh can be performed in any bank order. When there is no transaction in the CAM or the current bank has been recharged, the controller will give priority to the per-bank refresh of the current bank. For related register description, refer to DWC_ddr_umctl2_databook.pdf.

2.4.4 Low Power Management

DDRC supports a variety of low-power designs. When PWRCTL.dfi_dram_clk_disable=1, the IO clock will be automatically turned off in the following scenarios to save power:

- Self-refresh
- In working mode, no access request

When PWCTL.selfref_en=1 and the system is in idle state, that is, there is no read/write memory access on the DDRC bus interface for a certain period, DDRC will trigger to enter self-refresh. When the system detects an access request, the DDRC will automatically exit self-refresh immediately and turn on the IO clock.

2.4.5 Periodic Training

With the change of voltage and temperature, the LPDDR4 SDRAM parameters tDQSCK and tDQS2DQ will be affected. The DDR subsystem will periodically perform re-training to compensate for the impact of temperature and voltage changes. For detailed principles, refer to dwc_ddrn_phy_training_firmware_application_note.a.2020.06.pdf Chapter 13. For the specs of specific parameters, refer to the respective manufacturer's standards. The following table shows the specs of Micron LPDDR4.

Figure & Table 2-9 Temperature and voltage parameters

Parameter	Name	min	Max	Unit
Read DQS output access from CK_t/CK_c	tDQSCK	1500	3500	ps
tDQSCK voltage variation	tDQSCK_VOLT	-	7	ps/mv
tDQSCK temperature variation	tDQSCK_TEMP	-	4	ps/°C
DQ-to-DQS offset	tDQS2DQ	200	800	ps
tDQS2DQ voltage variation	tDQS2DQ_VOLT	-	33	ps/50mv
tDQS2DQ temperature variation	tDQS2DQ_TEMP	-	0.6	ps/°C

The compensation interval is configured by the register PhyMstrTrainInterval of the PHY and the default is about 0.7s. For the description of this register, refer to dwc_lpddr4x_multiphy_pub_databook_2.41a.pdf.

2.5 Usage

2.5.1 Reset Timing

When performing initialization, the clock and reset relationship must be as shown in Figure & Table 2-10.

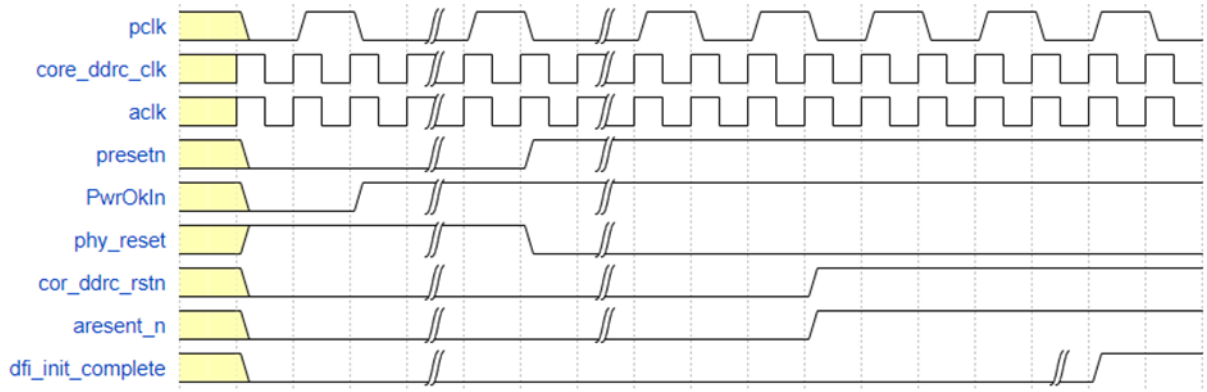

Figure & Table 2-10 Reset timing diagram

Reset within the DDR subsystem are register controllable.

1. Release PwrOKIn, ddr_sysreg.DDR_CFG0[6]=1.
2. Delay 1us.
3. Release APB reset (ddr_sysreg.DDR_CFG0[4]=1) and PHY reset (ddr_sysreg.DDR_CFG0[7]).
4. Configure the controller registers.
5. Release controller core reset (ddr_sysreg.DDR_CFG0[5]=1) and AXI port reset (ddr_sysreg.DDR_CFG0[13:8]=0x1f).
6. Initialize the PHY.
7. Execute DFI handshake, the controller enters mission mode.

2.5.2 Initialization Process

The DDR initialization flow chart is as follows:

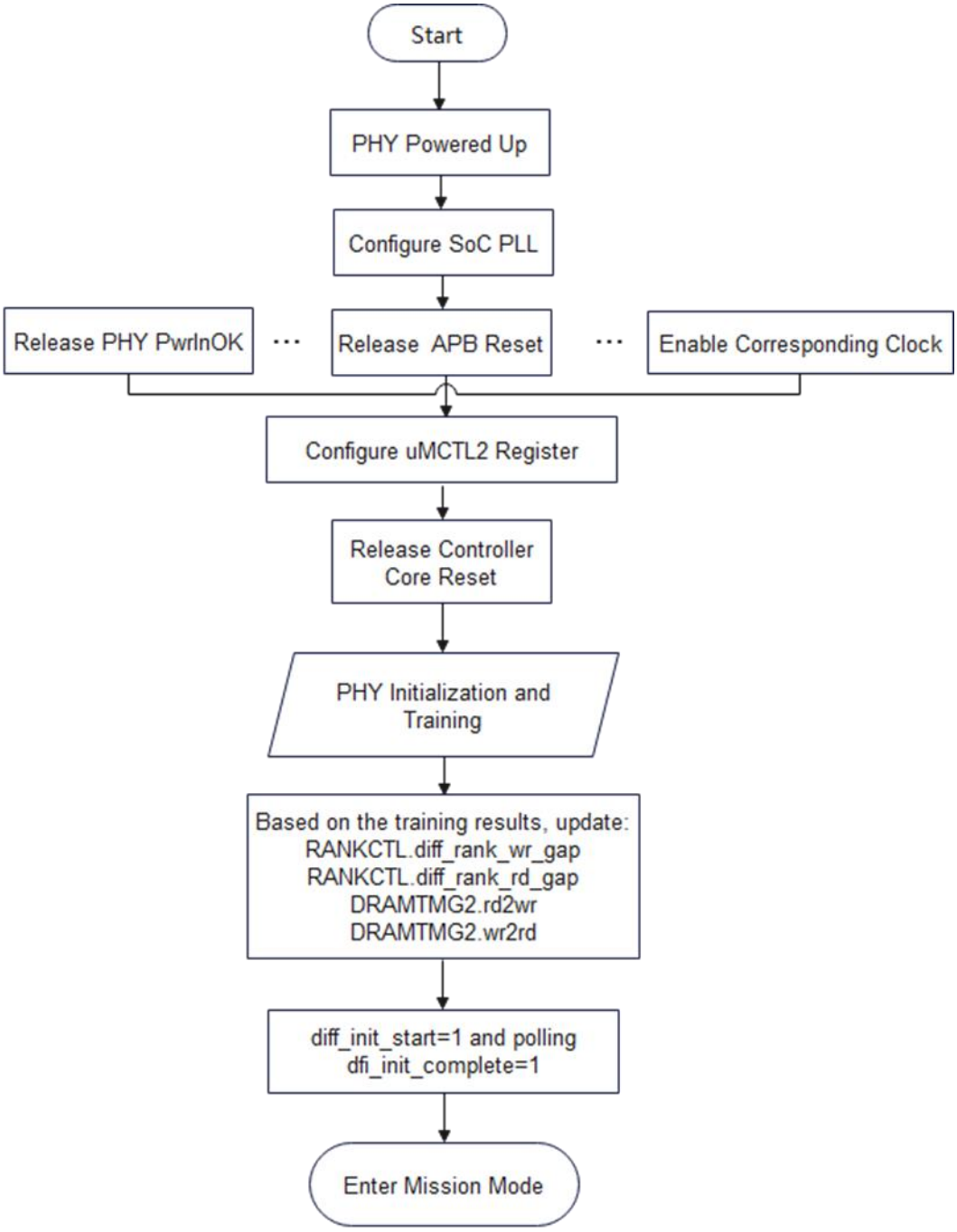

Figure & Table 2-11 DDR initialization flow chart

3 eMMC/SD

3.1 Overview

The DesignWare Cores DWC_mshc is a highly configurable and programmable, high performance Mobile Storage Host Controller (MSHC) with AXI as the bus interface for data transfer.

DWC_mshc provides a flexible bus interface that enables you to integrate DWC_mshc into embedded applications for System-on-a-Chip (SoC) designs. DWC_mshc has an AXI master interface that supports 64-bit address and data bus. Besides supporting non-DMA mode, DWC_mshc supports various DMA options such as SDMA, ADMA2, and ADMA3 as specified in the SD Host Controller Standard.

DWC_mshc primarily targets host-controller and card-reader applications. In such instances, the device memory card to which the DWC_mshc sends or receives data is typically a device for FLASH mass storage. The DWC_mshc is optimized for the following applications and systems:

- Portable electronic devices
- High-speed storage applications

Specifically, DWC_mshc is targeted at these devices:

- Mobile phones and laptops
- Digital cameras and camcorders
- Printer devices
- Embedded applications

3.2 Main Features

The main features of DWC_mshc are shown below.

- Supports SD memory and SD Input/Output (SDIO) digital interface protocol, and compliant with SD HCI specification.
- Uses the same SD-HCI register set for eMMC transfers.
- Supports eMMC protocols including eMMC 5.1.
- Supports SD-HCI host version 4 mode or less.
- Supports the following data transfer types for SD and eMMC modes:
  - CPU
  - SDMA
  - ADMA2
  - ADMA3
- Supports independent controller, slave interface and master interface clocks.
- Supports gating of controller base clock if host controller is inactive.
- Support context aware functional clock gates.
- Applications can gate the slave interface clock if host controller is inactive.
- Data buffering

  - 256 + 32 buffer depth
  - Automatic packing/unpacking of data to fit buffer width
- Interrupt outputs
  - Combined interrupt outputs
  - Supports interrupt enabling and masking.
- Supports Command Queuing Engine (CQE) and compliant with eMMC CQ HCI.
  - Programmable scheduler algorithm selection of task execution
  - Supports data prefetch for back-to-back write operations.
- Supports tuning:
  - SD/eMMC tuning using CMD19 (SD) or CMD21 (eMMC)
  - Mode 1 re-tuning - Host driver maintains the re-tune timer
  - Fully software driven tuning/re-tuning operations
  - Auto-tuning or mode 3 re-tuning

3.3 Interface

In the SoC system, there are 3 instances of DWC_mshc named eMMC/SDIO0/SDIO1.The data width of eMMC is 8bit with HS400 supported. And the data width of SDIO0 and SDIO1 are 4bit with SDR104 supported. The PADs of eMMC/SDIO0/SDIO1 are shown in Figure & Table 3-1.

Figure & Table 3-1 Pin description table

Pin Name	Direction	Width	Description
EMMC_CLK	O	1	CLK signal, 1.8V domain
EMMC_CMD	IO	1	CMD I/O signal, 1.8V domain
EMMC_DAT0	IO	1	DAT0 I/O signal, 1.8V domain
EMMC_DAT1	IO	1	DAT1 I/O signal, 1.8V domain
EMMC_DAT2	IO	1	DAT2 I/O signal, 1.8V domain
EMMC_DAT3	IO	1	DAT3 I/O signal,1.8V domain
EMMC_DAT4	IO	1	DAT4 I/O signal, 1.8V domain
EMMC_DAT5	IO	1	DAT5 I/O signal, 1.8V domain
EMMC_DAT6	IO	1	DAT6 I/O signal, 1.8V domain
EMMC_DAT7	IO	1	DAT7 I/O signal, 1.8V domain
EMMC_DS	I	1	This signal is generated by the device and only used in HS400 mode. The frequency of this signal follows the frequency of EMMC_CLK, 1.8V domain.
EMMC_RSTN	O	1	eMMC card reset signal output, 1.8V domain
SDIO0_CLK	O	1	CLK signal, 1.8V/3.3V domain

Pin Name	Direction	Width	Description
SDIO0_CMD	IO	1	CMD I/O signal, 1.8V/3.3V domain
SDIO0_DAT0	IO	1	DAT0 I/O signal, 1.8V/3.3V domain
SDIO0_DAT1	IO	1	DAT1 I/O signal, 1.8V/3.3V domain
SDIO0_DAT2	IO	1	DAT2 I/O signal, 1.8V/3.3V domain
SDIO0_DAT3	IO	1	DAT3 I/O signal, 1.8V/3.3V domain
SDIO0_DETN	I	1	Card detect signal When this is 0, it represents card is connected. When this signal goes from 0 to 1, card insertion interrupt is generated if enabled.
SDIO0_WPRTN	I	1	SD card write protect signal When this is 0, it represents card is write protected.
SDIO1_CLK	O	1	CLK signal, 1.8V/3.3V domain
SDIO1_CMD	IO	1	CMD I/O signal, 1.8V/3.3V domain
SDIO1_DAT0	IO	1	DAT0 I/O signal, 1.8V/3.3V domain
SDIO1_DAT1	IO	1	DAT1 I/O signal, 1.8V/3.3V domain
SDIO1_DAT2	IO	1	DAT2 I/O signal, 1.8V/3.3V domain
SDIO1_DAT3	IO	1	DAT3 I/O signal, 1.8V/3.3V domain
SDIO1_DETN	I	1	Card detect signal When this is 0, it represents card is connected. When this signal goes from 0 to 1, card insertion interrupt is generated if enabled.
SDIO1_WPRTN	I	1	SD card write protect signal When this is 0, it represents card is write protected.

3.4 Function Description

3.4.1 Overview of Architecture

This section describes the main functional building blocks of the DWC_mshc controller, represented in Figure & Table 3-2.

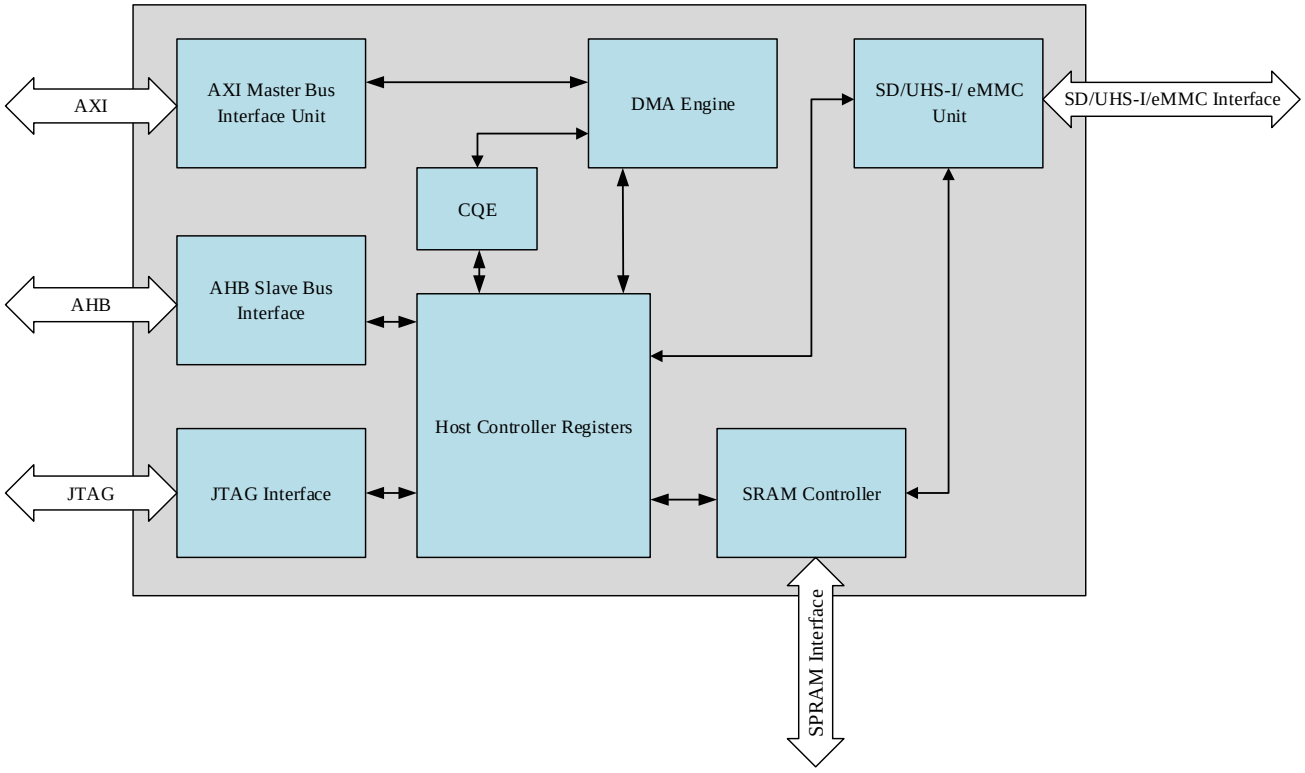

Figure & Table 3-2 System-level block diagram

Following are the various modules in the MSHC:

- Master Bus Interface Unit (MBIU)
  This MBIU implements the logic to transfer data on the AMBA Extensible Interface Bus (AXI). The AXI interface transfers data to and from the system memory through the AXI master bus interface.
- AHB Slave Bus Interface (SBI) module
  The SBI module implements the logic to primarily access the DWC_mshc registers by using an external AMBA high-performance (AHB) bus. This module supports only the little endian scheme for register access.
- JTAG Interface
  The JTAG interface is not configured for use.
- DMA Engine
  The DMA Engine unit handles data transfer between DWC_mshc and system memory.
- Host Controller Registers
  The host controller register unit comprises the standard SD host controller registers as specified in the SD Specifications Part A2 SD Host Controller Standard Specification Version 4.20a. It also includes Command Queuing registers compliance to JEDEC eMMC 5.1 HCI specification.
  These registers are implemented in three clock domains, namely AXI master bus interface clock (aclk), AHB slave bus interface clock (hclk), and controller base clock (bclk). The non-transfer related registers are implemented in the AHB slave clock domain, so that the controller base clock can be gated for power savings.
- SRAM Controller (Packet buffer interface)

The SRAM controller interfaces the packet buffer of the host and the transaction controller units (SD/UHS-I/eMMC unit).

- SD/UHS-I/eMMC Unit
  SD/UHS-I/eMMC unit manages the SD/eMMC interface protocols.
- CQE
  This module implements command queuing and includes the following:
  – Task scheduler with the ability to prioritize execution of tasks
  – Control logic for descriptor fetch
  – Control and sequence task submission and execution
  – Status polling
  – Timers and counter dedicated for CQE operation

3.4.2 Device and Card Interface

3.4.2.1 SD Card Interface

SD unit is responsible for managing SD interface protocols.
The key features of this unit are as follows:

- Generate DS/High-speed/UHS-I command and data packets.
- Generate CRC and check for command and data packets.
- Handle packet timeouts.
- Support 1-bit DAT and 4-bit DAT modes.
- Handle SDIO card interrupt.
- Supports Default Speed (DS), High-Speed (HS), SDR12, SDR25, SDR50 and SDR104 speed modes.

3.4.2.2 eMMC Card Interface

eMMC unit is responsible for managing eMMC interface protocols.
The key features of this unit are as follows:

- Generate eMMC bus command and data packets.
- Generate CRC and check for command and data packets.
- Handle packet timeouts.
- Support 1-bit DAT, 4-bit DAT, and 8-bit DAT modes.
- Support legacy, High Speed SDR, High Speed DDR, HS200, and HS400 modes.

3.4.3 Master Bus Interface

DWC_mshc utilizes the MBIU to initiate data traffic with the system memory. The system memory read/write transfer requests are initiated by the DMA of DWC_mshc through a native master interface, which is the Generic Master (GM) interface. Irrespective of the selected system bus protocol, the GM interface is used to interact between the DMA and the MBIU. DMA transfers utilize the read descriptors to transfer (write/read) data from system memory.

The GM interface operates on a packet- or descriptor-level transfer. That is, each request represents a descriptor or packet size transfer. In contrast, the AXI gaskets break the GM request into multiple AXI transfers as appropriate for the interface.
Wait cycles may appear on the master read or write data path. This happens if the RAM controller does not have enough bandwidth to serve both the Transaction Layer Unit (TLU) and the DMA. It can also occur during CQE as the DMA FIFOs are also utilized for message passing.
The AXI master interface is a gasket between the native GM and the AXI bus. DMA read and write accesses are initiated by the DWC_mshc controller through the AXI master interface to read descriptors, and read/write transfer data. The read and write channels are independent and can be active simultaneously. However, if a read access is dependent on a write access completing first, the controller does not issue the DMA read until it receives the DMA write response from AXI.

3.4.4 AHB Slave Bus Interface

The AHB slave interface supports the following features:
- Fully compliant with AMBA 2.0 AHB slave
- Single burst transfer
- 32-bit write or read transfers to DWC_mshc registers
- Transfer response: OKAY, ERROR, and RETRY responses

3.4.5 DMA Engine

The DMA Engine unit handles data transfer between DWC_mshc and the system memory. The key features of this unit are as follows:
- Support SDMA/ADMA2/ADMA3 modes.
- Fetch the descriptor and data. The same DMA engine is used to interleave data transfer and descriptor fetch. This enables new task descriptor fetches (for CMD44 and CMD45) while DMA is moving data during task execution (for CMD46 and CMD47).
- The AXI transaction ID 0 is used for moving data and AXI transaction ID 1 is used to fetch task descriptors.
- Pre-fetch data for back-to-back eMMC write commands.
- Write back the received data packets to the system memory.

3.4.6 SRAM Controller

The SRAM controller interfaces the packet buffer of the host and the transaction controller units (SD/UHS-I/eMMC). The buffer depth is configured as 256 + 32, where size of each location is equal to the AXI data width (64bits).
As write and read transfers to the cards do not occur simultaneously, a single shared buffer is used for read and write operations to save area. During the data transfer command handshake, the read/write bit of the command register is sampled and stored. This internal bit defines whether the DWC_mshc is in the read or write mode.

The packet buffer uses a single clock single-port RAM synchronous to the controller base clock (bclk).

3.4.7 Data Flow for Card Read/Write in DWC_mshc

Figure & Table 3-3 explains how data flows from the card interface (SD/SDIO/eMMC) to the AXI interface through packet buffer for card read transfer. Received data from the card interface is written into packet buffer. When one block of data (SD/SDIO/eMMC) is received, DMA starts transmitting that data to the system by reading it from packet buffer. For a card write transfer, data flows in the reverse direction. DMA writes data into packet buffer that is subsequently read by the card interface logic.

DMA and card interface logic can work simultaneously as read and write to packet buffer can be interleaved. For card read, DMA can send out previous block while card interface logic is receiving current block. Whereas for card write, DMA can write current block into packet buffer while card interface logic is sending out previous block.

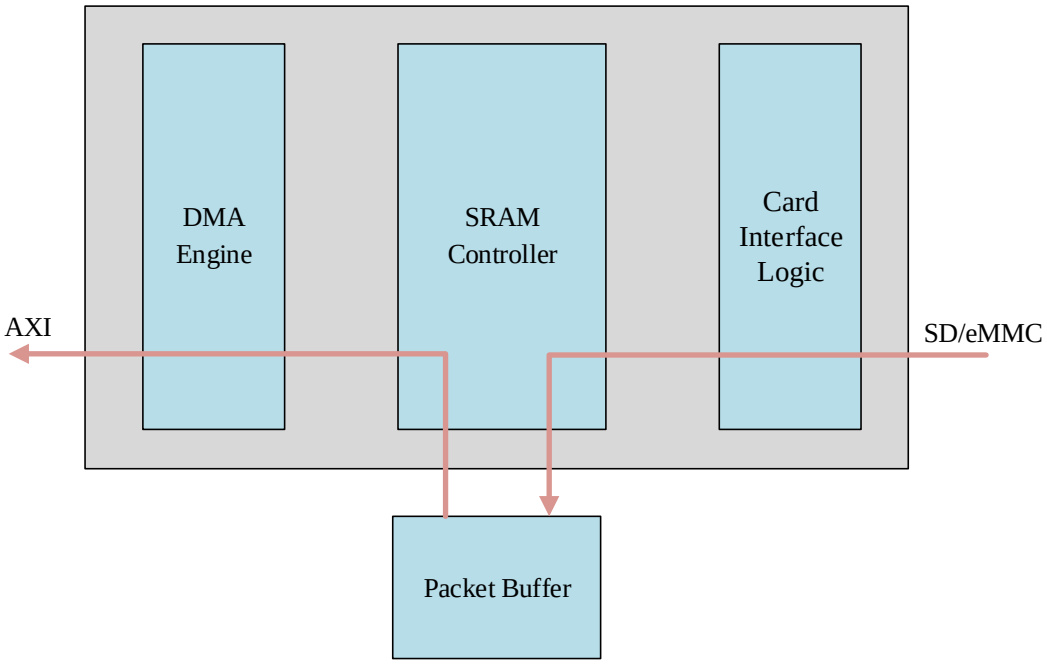

Figure & Table 3-3 Data flow for card read/write in DWC_mshc

3.4.8 Tuning in DWC_mshc

DWC_mshc is compatible with SD 4.2 and eMMC 5.1 protocols and both these protocols have speed modes in which the incoming data must be sampled on a clock with a programmable sampling. Therefore, a method called tuning is required to identify the correct phase where the data can be robustly sampled.

Following are the tuning and re-tuning modes supported in DWC_mshc:

- Tuning flow based on CMD21 (eMMC) or CMD19 (SD)
- Mode 1 (software assisted) re-tuning flow
- Mode 3 (hardware managed) re-tuning (that is, auto-tuning)

- Software driven tuning flow

As shown in Figure & Table 3-4, the output of DWC_mshc signals tuning_cclk_sel and tuning_cclk_sel_update are driven by the internal tuning engine and must be used for phase control of the DelayLine. These outputs are used by tuning engine to change the incoming phase of cclk_rx clock so that it can find the phase at which incoming data on sd_dat_in port can be robustly sampled.

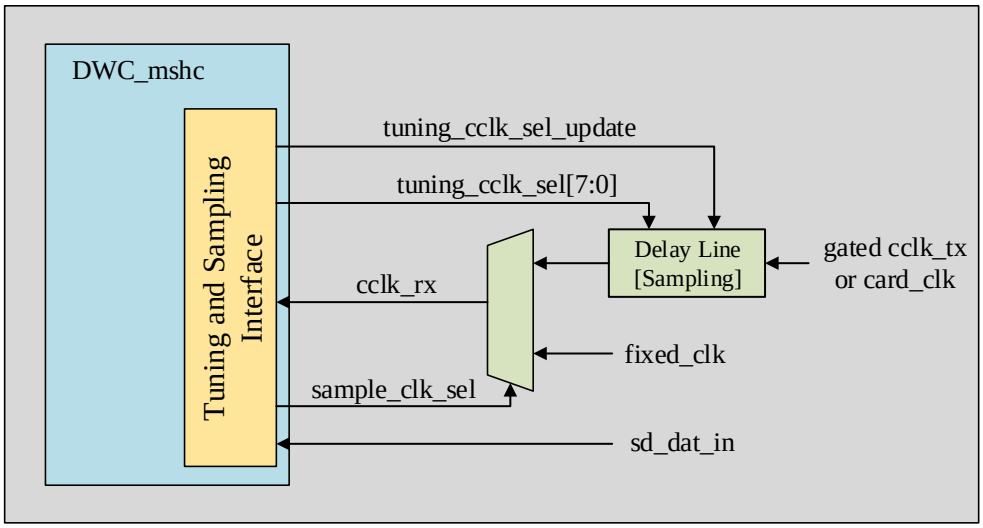

Figure & Table 3-4 Tuning

The high-speed SDR modes that need tuning are eMMC HS200 and SD SDR104. The tuning sequence for SD or eMMC is a software-assisted sequence in which the hardware controls the DelayLine and calculates the phase for robust sampling, and the software determines the necessity for tuning and starting the tuning sequence.

3.4.8.1 Mode 1 Re-Tuning

DWC_mshc implements mode 1 re-tuning as per the SD HCI specification and is available for use even in the eMMC mode of operation. Mode 1 re-tuning is similar to the tuning sequence, and a distinction between the two is made using HOST_CTRL2_R.SAMPLE_CLK_SEL register field value. The SD HCI specification recommends that software must invoke re-tuning sequence for every 4 MB of data transferred or as per the re-tuning timer maintained by the software.

During re-tuning, the tuning engine tests the validity of the sampling window, using the information gathered in the initial tuning sequence. If the window is unaltered, re-tuning sequence is quick (total CMD21/CMD19 iterations equal to number of Pass-steps +1). In case of any drifts, re-tuning may result in more CMD21/CMD19 iterations (equal to total steps of DelayLine).

Following are the tuning schema available for mode 1 re-tuning:

- Threshold Based Selection Tuning Schema

  This mode allows the tuning engine to select the first complete sampling window that meets the threshold criteria defined. In this mode, a complete sampling window is preferred over a partial sampling window. Tuning stops when a complete sampling window meets the threshold criteria. Tuning also stops when all the taps are parsed.
- Backward Compatible Tuning Schema

This mode allows the tuning engine to function as it was in 1.50a and earlier releases of the DWC_mshc. In this mode, a complete sampling window is preferred over a partial sampling window. Tuning stops when either a complete sampling window is found or when all taps are parsed. There is no minimum threshold on the width of sampling window.

- Largest Sampling Window Tuning Schema
  This mode allows the tuning algorithm to move through all the taps of delay line to identify the largest sampling window available. In this mode, tuning stops only when all the taps are parsed. The algorithm picks the largest sampling window.

3.4.8.2 Software Tuning

DWC_mshc implements a fully software-controlled tuning sequence called the software tuning. The software can set the AT_CTRL_R. SW_TUNE_EN bit to take control of tuning_cclk_sel, tuning_cclk_sel_update and sample_clk_sel signals which form the tuning and data sampling interface. When AT_CTRL_R. SW_TUNE_EN is set to 1, the software can write into the AT_STAT_R. CENTER_PH_CODE bit to change the value of tuning_cclk_sel. While using software tuning, software must calculate the center phase and periodic re-tuning of the sampling clock.

3.4.8.3 Auto-Tuning or Mode 3 Re-Tuning

Auto-tuning is a hardware-managed re-tuning feature that complies with the SD HCI Mode3 re-tuning procedure. Auto-tuning removes the need for the host software to re-tune the sampling clock for every 4 MB of data transfer (SD HCI). A tuning sampling clock is recommended in both SD and eMMC modes to ensure ease in timing closure and for robust operation while operating at high SDR speeds, such as SDR104 and HS200.

Following are some functionalities of the auto-tuning feature:

- Requires tuning based on CMD19 (SD) or CMD21 (eMMC) for initialization.
  Once initialized, the auto-tuning engine can detect drifts and calculate corrections when data transfers are in progress.
- Does not depend on tuning commands of SD/eMMC for drift detection or corrections.
  This is necessary as auto-tuning happens in the background during regular data/CMD transfers.
- Detects drifts for all active data lines of the controller.
- Provides a software selectable correction interval.
  Corrections can be applied either at block boundary or between data transfers.
- Handles drifts
  Drift monitoring window is hardware initialized. Software can program the AT_CTRL_R register to alter the initialized values, if required. If auto-tuning fails to correct a detected drift, the tuning-error interrupt is set.

Following is a procedure for an auto-tuning operation:

1. Software must initiate tuning sequence based on CMD19 (for SD) or CMD21 (for eMMC) for initializing the Auto-tuning engine.
2. After initialization, the auto-tuning engine is activated by default for all data read transfers.

3. Auto-tuning engine uses a DM DelayLine to generate drift_cclk_rx.
4. The drift_cclk_rx is generated using the same source as cclk_rx; however, its phase is controlled by the autotune_cclk_sel outputs of the auto-tuning engine.
5. Auto-tuning engine uses the data sampled on drift_cclk_rx to check for drifts on the data lines. A drift is detected if following data errors occur:
   - Data CRC error
   - Start bit error
   - End bit error

   These drift checks are active only when READ transfers are in progress in HS200/SDR104 modes.
6. On drift detection, auto-tuning engine sweeps through DM DelayLine phases until it finds a sampling window without drifts.
7. The new sampling window range allows the auto-tuning engine to calculate a correction for the sampling phase.
8. The calculated correction is applied on the sampling DelayLine using tuning_cclk_sel outputs at the software-selected correction interval.

According to the SD specification, the timer load value is a capability that is a configuration value and not a programmable value. This value must be maintained by the software and not by the hardware. The software must maintain the re-tuning timer and must start this timer at the end of a data transfer and reset it when a new data transfer starts. Software must invoke the CMD-based tuning procedure when this timer expires.

3.4.8.4 Interrupts

In DWC_mshc, interrupts are generated based on various events.
There are two interrupt outputs provided by DWC_mshc:

- intr
- wakeup_intr

The interrupt signal must be used as interrupt for different events during active mode. During standby mode, wakeup_intr must be used to identify any wakeup event, such as card removal or insertion, or an SDIO card interrupt. The interrupts are of level type, that is, the interrupt remains asserted (high) until it is cleared by the host or the software.
The interrupt status registers indicate the events that caused the interrupt generation. Each event can be prevented from asserting the interrupt on the interrupt signal by setting the corresponding mask bits.
A bit in interrupt status register is set only if the corresponding interrupt status enable is set and interrupt event is observed. This bit set in interrupt status register asserts interrupt only if corresponding bit in interrupt signal enable is set.
Host driver is responsible for enabling wakeup signals and disabling interrupt signals when the host system enters its standby mode, and for disabling wakeup signals and enabling interrupt signals when host system goes into active mode. The host driver must not enable both at the same time. Interrupt signals are enabled using interrupt signal enable and wakeup signals are enabled using wakeup event enable.

3.4.8.5 Error Detection

Figure & Table 3-5 lists error types and categories into which they can be grouped for SD/eMMC mode.

Figure & Table 3-5 Error types and categories for SD/eMMC mode

Error Type	Categories
Command errors	■ Command timeout error ■ Command CRC error ■ Command end bit error ■ Command index error ■ Command conflict error ■ Response error
Auto command errors	■ Command not issued by auto CMD12 error ■ Auto command timeout error ■ Auto command CRC error ■ Auto command end bit error ■ Auto command index error ■ Auto command conflict error ■ Auto command response error
Data errors	■ Data timeout error ■ Data CRC error ■ Data end bit error ■ ADMA error ■ Tuning error

3.5 Usage

Following is the programming sequence required before starting the data transfer with the card:

1. Start providing hclk to DWC_mshc and apply its reset hresetn. This clock is required for card detection logic and for accessing most of the registers in DWC_mshc.
2. Check if the card is already inserted by reading Present State Register (PSTATE_REG). This step is only required for a removable card. If the card is already inserted (eMMC), then go to Step 4.
3. Wait for the card insertion interrupt. This step is only required for removable card. Card detection sequence is described in section “Card Detection”.
4. Setup basic settings for DWC_mshc as described in section “Host Controller Setup Sequence”. It involves settings such as the bus power voltage level for the card, timeout counter value, setting clock generation parameters. Different setup sequences for SD card (“SD Interface”) and eMMC card (“eMMC Device”) are provided. This step can also be executed immediately after Step 1.

5. Enable input clocks (other than hclk) that are available in DWC_mshc. Internal clocks (aclk, bclk, tmclk and cqetmclk) are enabled followed by the card clocks (cclk_tx/cclk_rx) as discussed in the programming sequence in section "Host Controller Clock Setup Sequence". At this stage, DWC_mshc is ready to communicate with a card.
6. Identify the type of card (SD or eMMC card) that is connected to DWC_mshc as explained in the programming sequence in section "SD Card Interface Detection". If DWC_mshc is connected to an eMMC card, then programming sequence described in section "eMMC Card Interface Setup" must be executed. This is a simplified sequence especially for an eMMC card as the type of card is always known. Both the programming sequences provide power and clock to the card.
7. After powering up the card, the card is initialized. Appropriate sequence as shown in Figure & Table 3-6must to be executed based on type of the connected card.
8. As both host controller and card are initialized, the host can start sending commands to the card to perform the data transfer. Control and data commands are two types of commands that can be issued to the card. The control command is used to read or write any register in the card. The data command is used for writing or reading data to or from the card. Figure & Table 3-7 lists the control and data command sequences for SD and eMMC cards.

Figure & Table 3-6 Card initialization sequence for different cards

Type of Card	Card Initialization Sequences
SD	SD Card Initialization and Identification ("SD Card Initialization and Identification")
eMMC	Initializing and Identifying an eMMC Card ("Initializing and Identifying an eMMC Device")

Figure & Table 3-7 Command sequence for different cards

Type of Card	Command Sequence	
SD/eMMC	Control CMD	Issuing CMD without Data Transfer ("Issuing CMD without Data Transfer")
	Data CMD	Issuing CMD with Data Transfer ("Issuing CMD with Data Transfer")

3.5.1 Card Detection

Figure & Table 3-8 shows the sequence for detecting a card. The procedure outlined in Figure & Table 3-8 applies to both SD and SDIO cards. This programming sequence is not required for an eMMC device as it is non-removable.

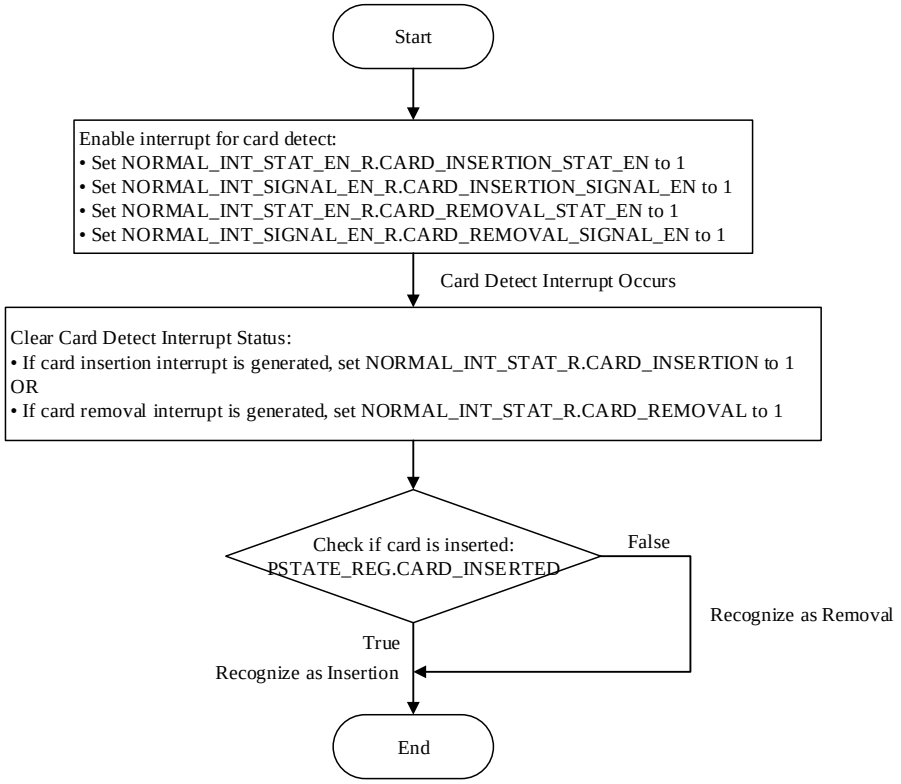

Figure & Table 3-8 Card detection

3.5.2 Host Controller Setup Sequence

This section discusses the host controller setup sequence for SD interface and an eMMC device.

3.5.2.1 SD Interface

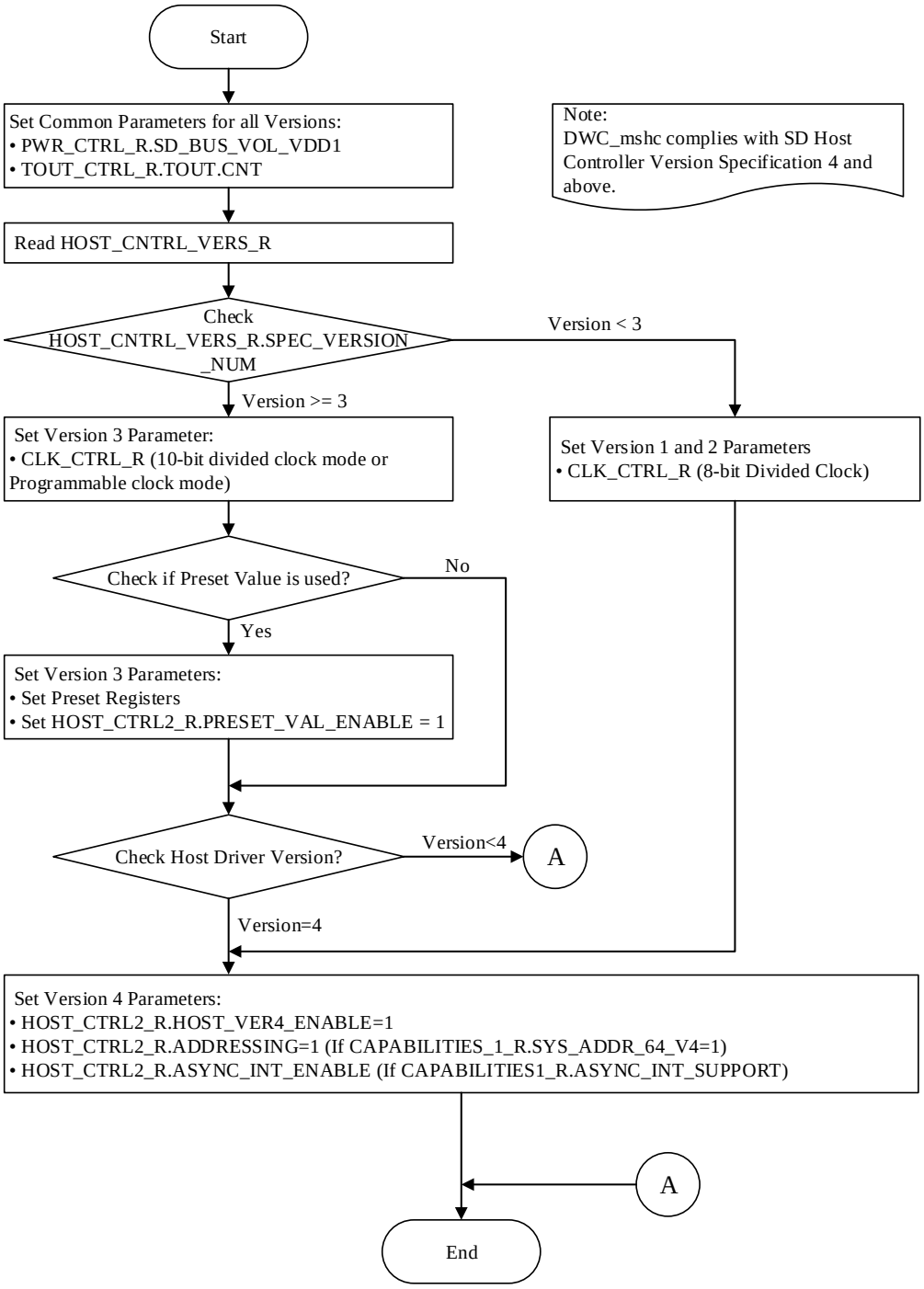

Figure & Table 3-9 Host controller setup sequence for SD

3.5.2.2 eMMC Device

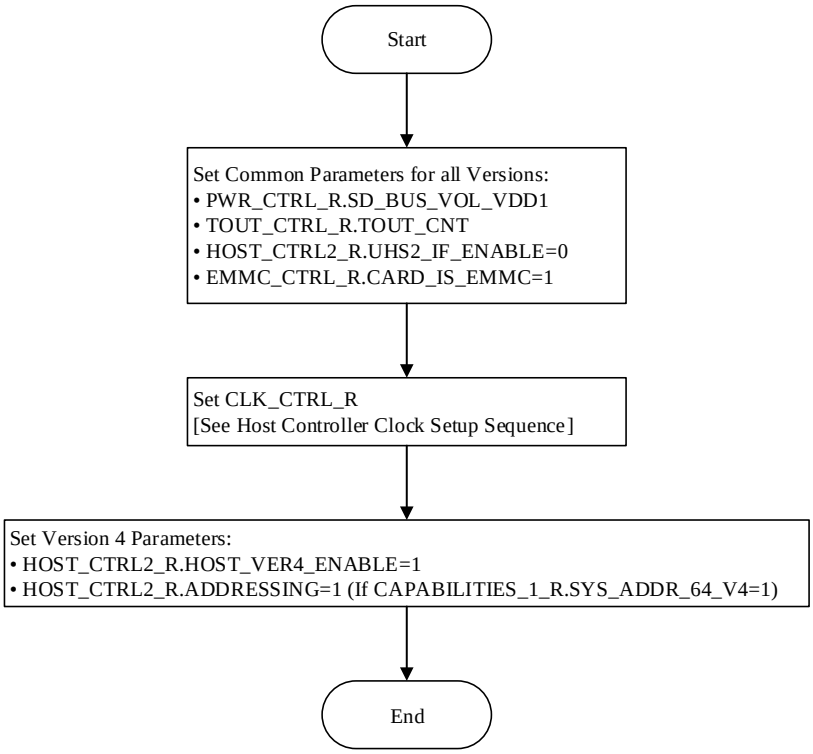

Figure & Table 3-10 Host controller setup sequence for eMMC interface

3.5.3 Clock Control

This section discusses the programming sequence for setting up internal clocks and card clock provided to DWC_mshc and for supplying and setting up a clock provided to the card, and for changing the card clock frequency.

3.5.3.1 Host Controller Clock Setup Sequence

Figure & Table 3-11 shows the sequence for setting up clocks for DWC_mshc. The DWC_mshc requires number of clock signals from the system. All the input clocks are asynchronous to each other. Based on controls available for enabling clocks required for DWC_mshc, they are categorized into two types, namely internal clock and card clock.

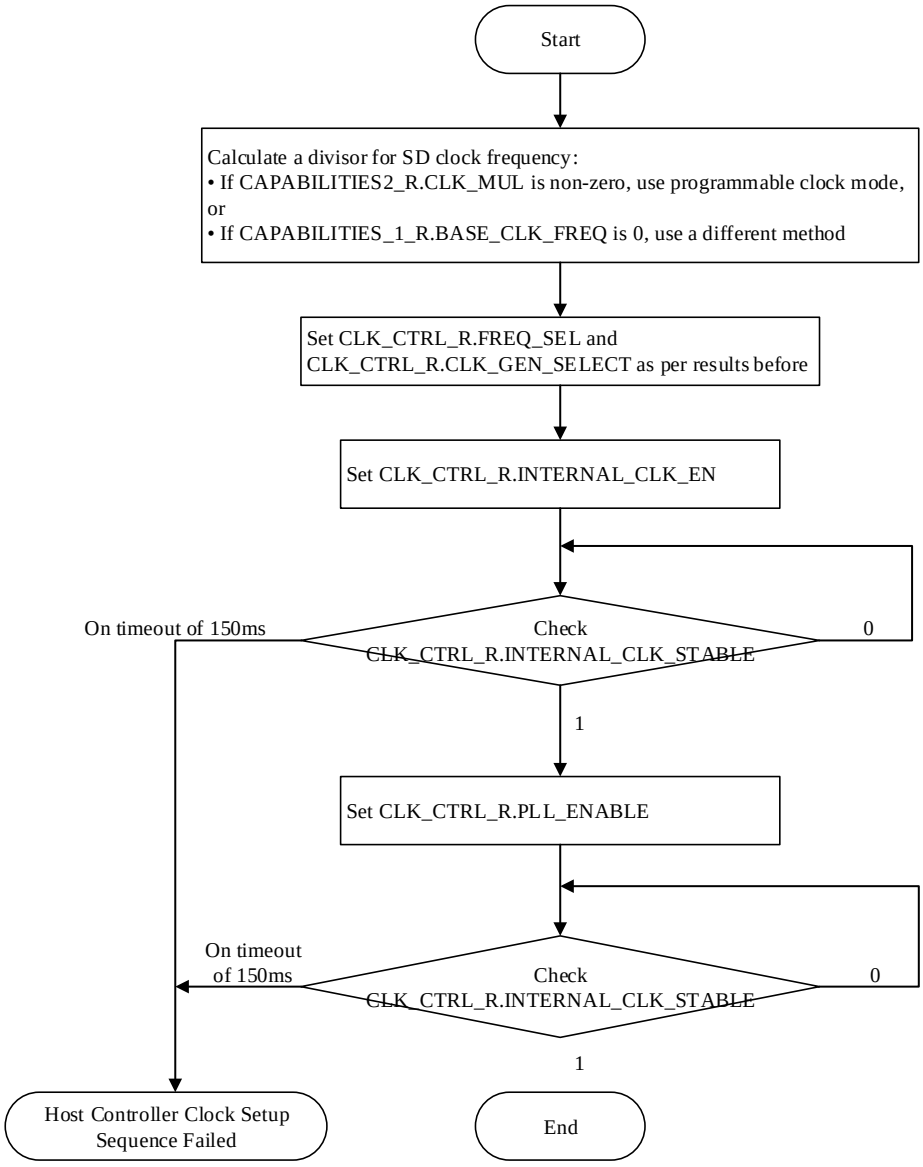

Figure & Table 3-11 Host controller clock setup sequence

3.5.3.2 Card Clock Supply and Stop Sequence

Figure & Table 3-12 shows the flow chart for stopping the clock to card.

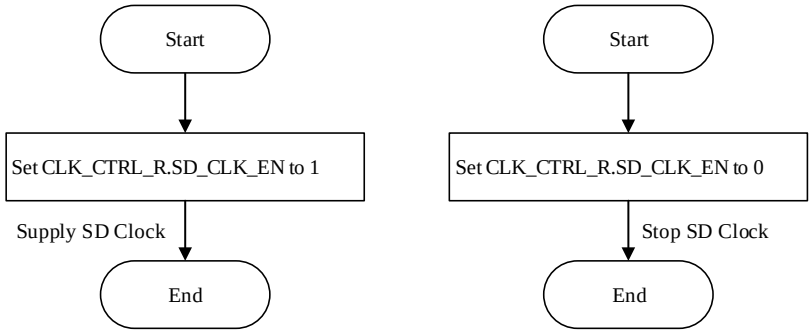

Figure & Table 3-12 Card clock supply and stop sequence

3.5.3.3 SD Clock Frequency Change Sequence

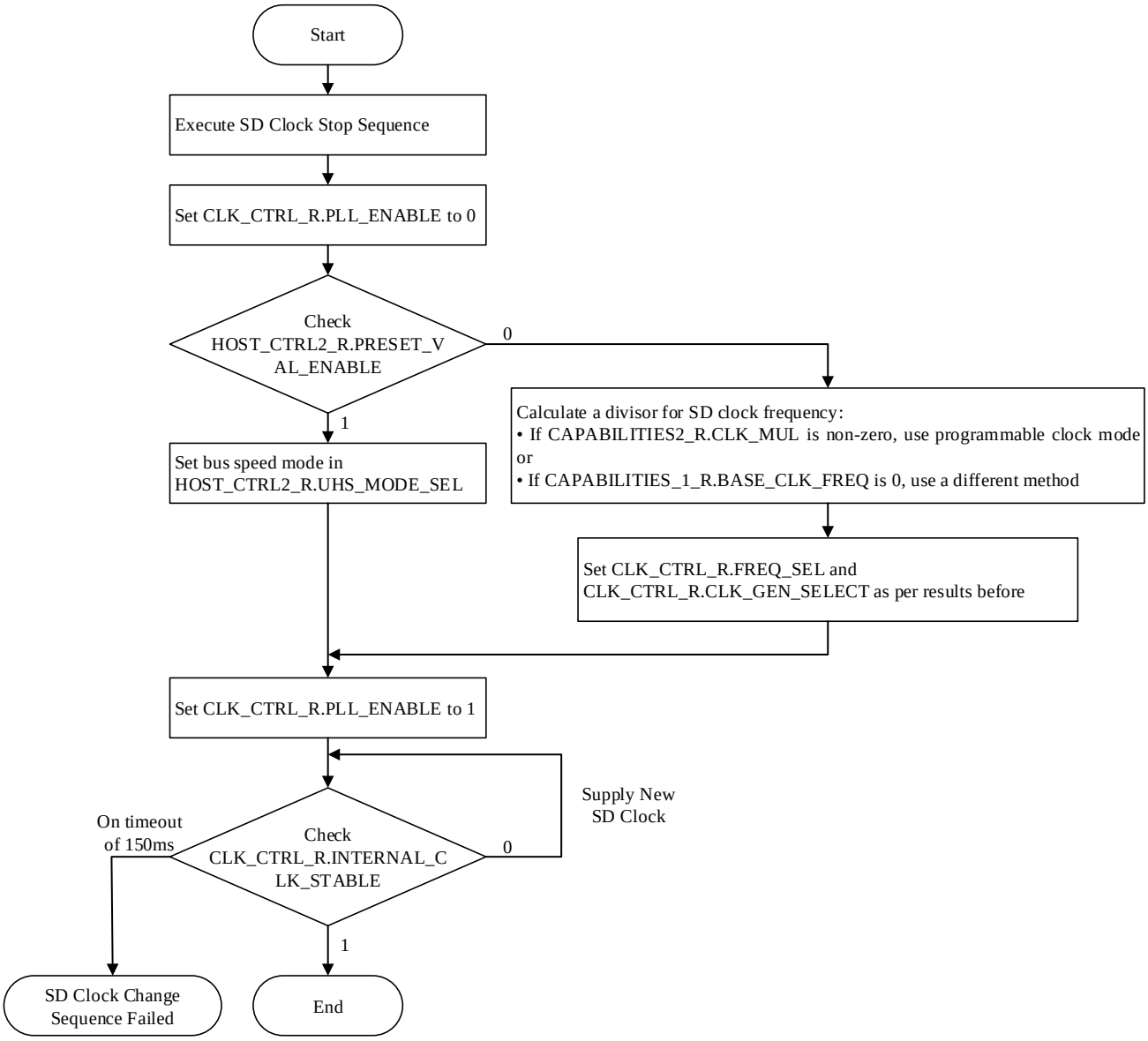

Figure & Table 3-13 SD clock frequency change sequence

3.5.4 Card Interface Setup Sequence

This section discusses the programming sequence for setting up the card interface for an SD and an eMMC card.

3.5.4.1 SD Card Interface Detection

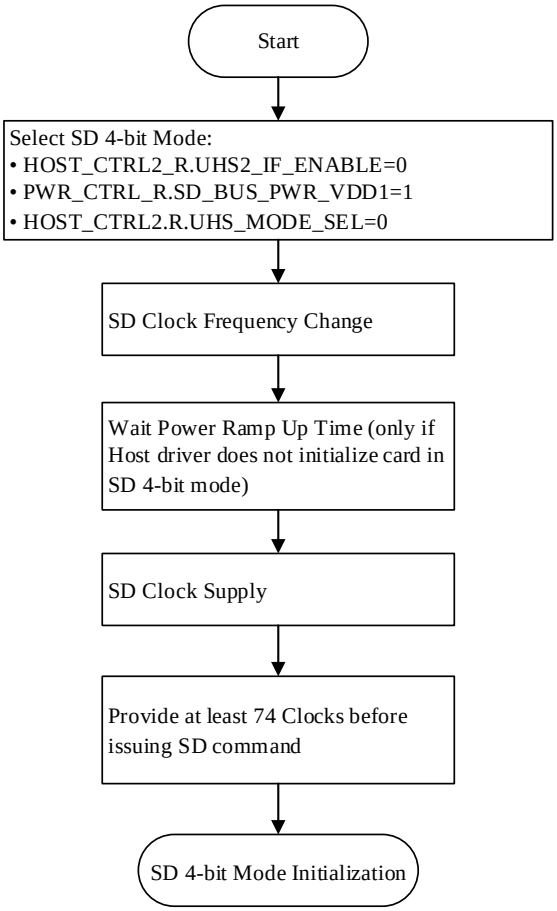

Figure & Table 3-14 SD card interface detection sequence

3.5.4.2 eMMC Card Interface Setup

Figure & Table 3-15 shows the programming sequence to set up an eMMC device.

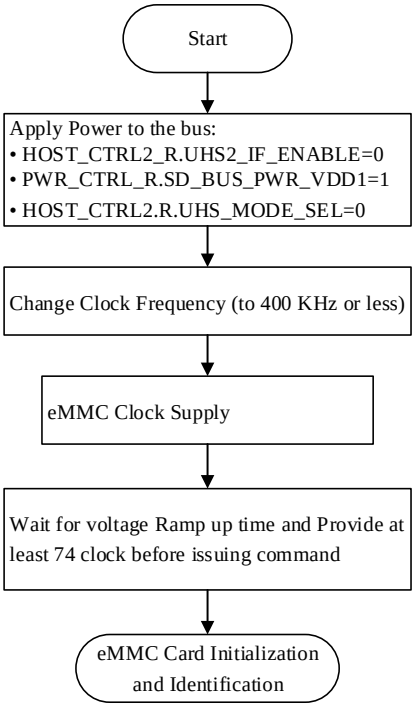

Figure & Table 3-15 eMMC card setup sequence

3.5.5 Tuning Sequences

This section discusses the tuning sequences for software tuning for SD and eMMC modes.

3.5.5.1 Tuning Flow Sequence

Figure & Table 3-16 shows how to perform mode1 re-tuning, which is the process to start tuning when the HOST_CTRL2_R.SAMPLE_CLK_SEL register bit is already set to 1. While using a tuning reset before the start of a new tuning, use a dedicated HOST_CTRL2_R write to first reset tuning, followed by a second HOST_CTRL2_R write to start tuning. A CMD19/CMD21 is used to start the tuning procedure. CMD19/CMD21 is similar to any read command. Therefore, before writing CMD19/CMD21 into the CMD_R, the settings of BLOCKSIZE_R, BLOCKCOUNT_R, and XFERMODE_R registers must be proper.

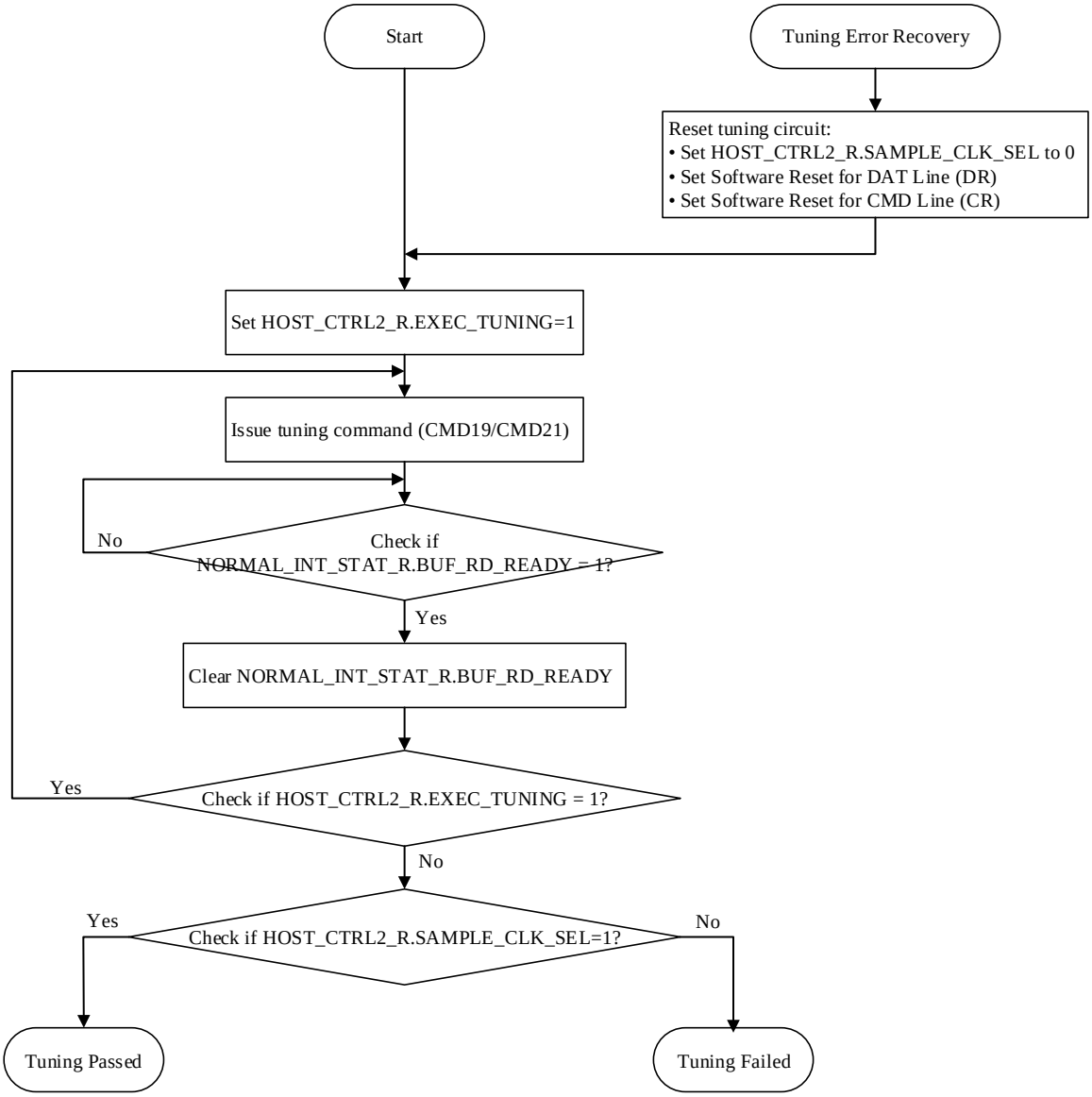

Figure & Table 3-16 Tuning sequence

3.5.5.2 Mode 1 Re-Tuning Flow

Figure & Table 3-17 shows the Mode 1 re-tuning programming sequence.

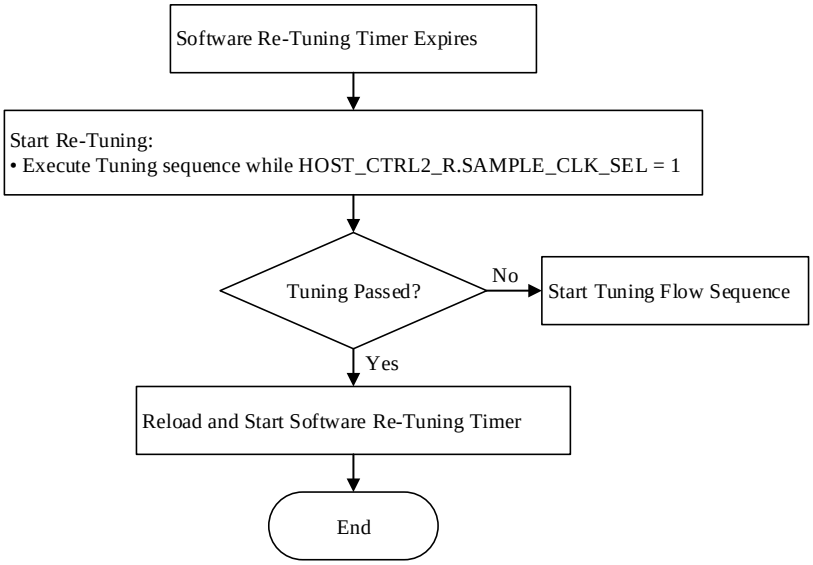

Figure & Table 3-17 Mode 1 re-tuning flow sequence

3.5.5.3 Mode 3 Re-Tuning/Auto-Tuning Flow Sequence

The auto-tuning feature is enabled by default in DWC_mshc when the speed mode is switched to SDR104 or HS200. If there is a need to change any of its parameters or to disable it, The AT_CTRL_R register can be used to enable or disable this feature. AT_CTRL_R can be programmed only when sample_cclk_sel is set to 0 before starting tuning.
You must not use mode-1 re-tuning when auto-tuning (mode3) is enabled.

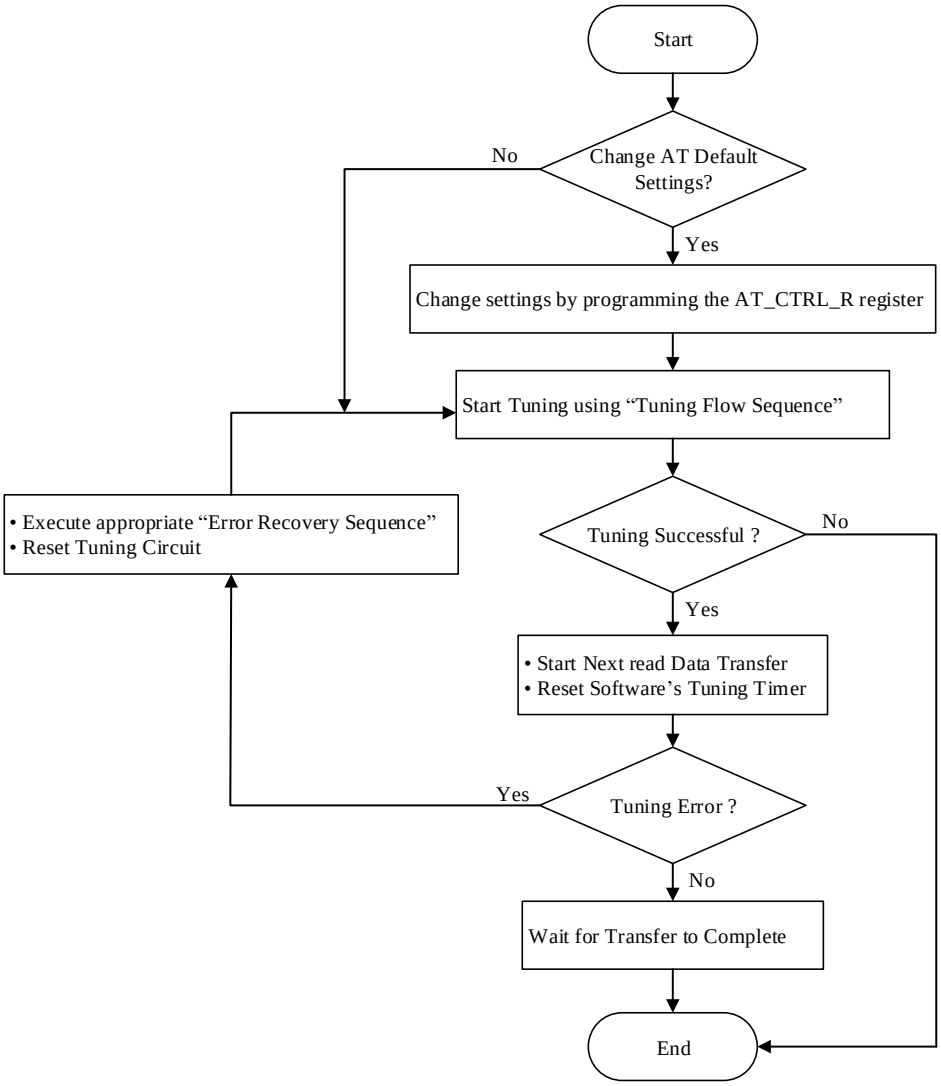

Figure & Table 3-18 Auto-tuning sequence

3.5.6 SD/SDIO Transaction Mode

This section includes the sequences to generate and control different types of SD/SDIO transactions that can be classified as transactions without data transfer using the DAT line and transactions with data transfer using the DAT line. This section also discusses the sequence for tuning the flow, controlling the bus power, changing the bus width and speed mode, setting the timeout on a DAT line, initializing and identifying a SD card, and the aborting sequence.

3.5.6.1 SD Card Initialization and Identification

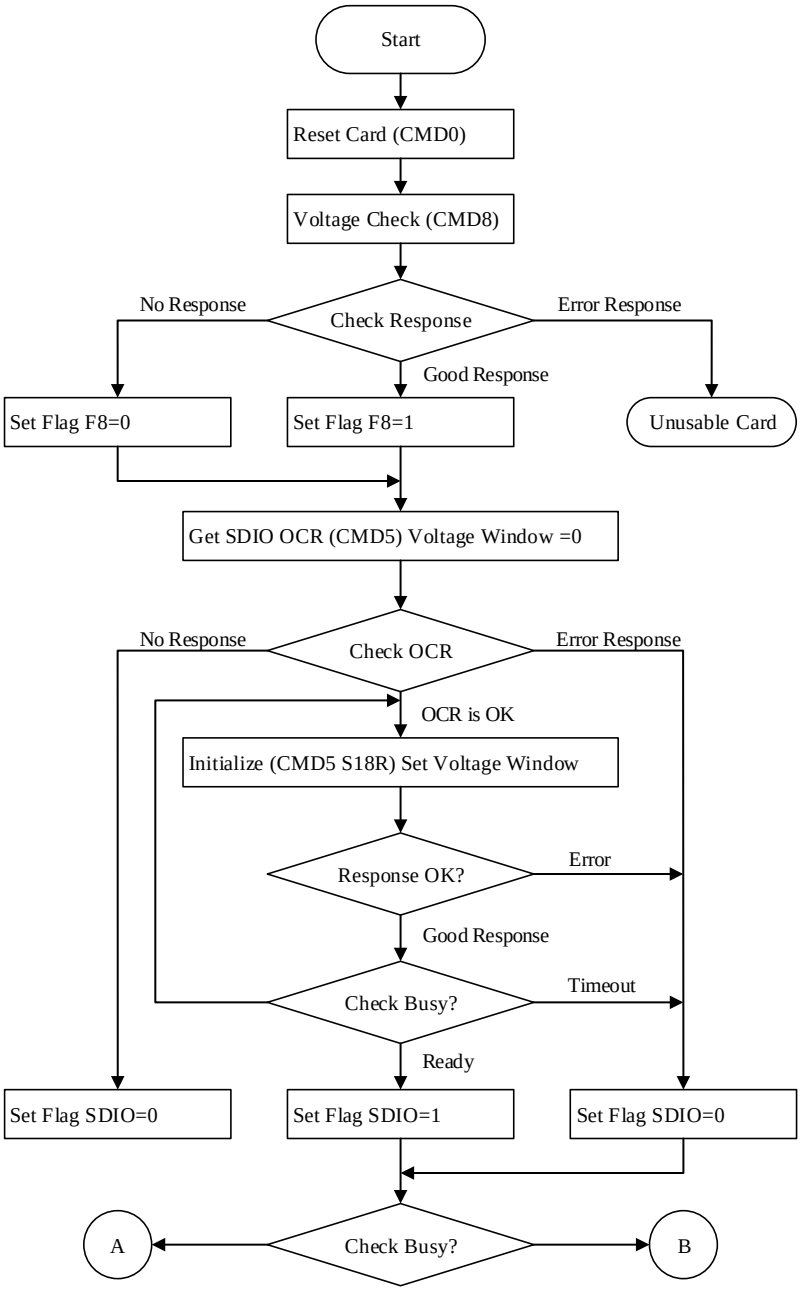

Figure & Table 3-19 SD card initialization and identification

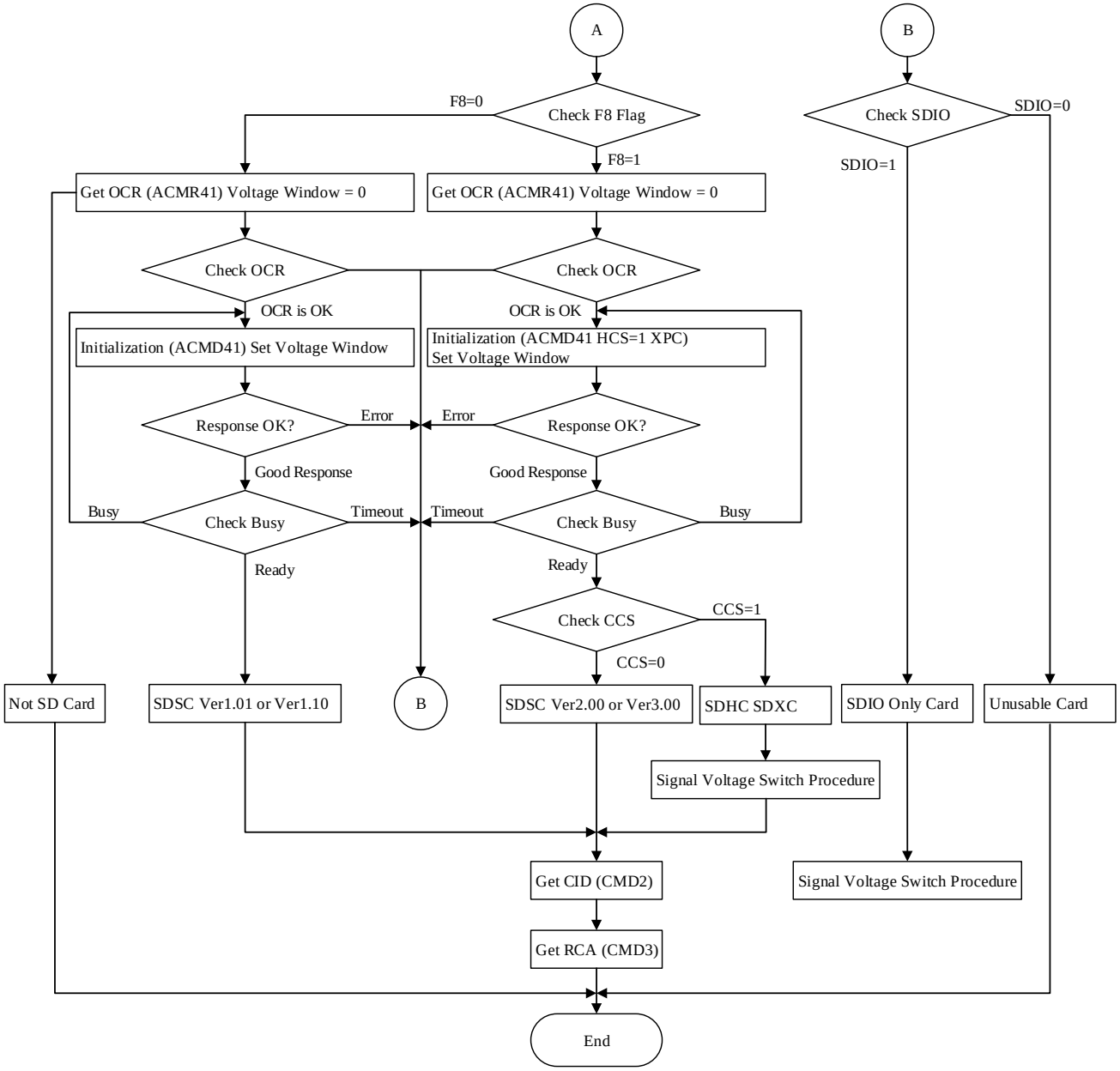

Figure & Table 3-20 SD card initialization and identification (continued)

3.5.6.2 Changing SD Bus Width

Figure & Table 3-21 shows the sequence for changing bit mode on SD Bus.

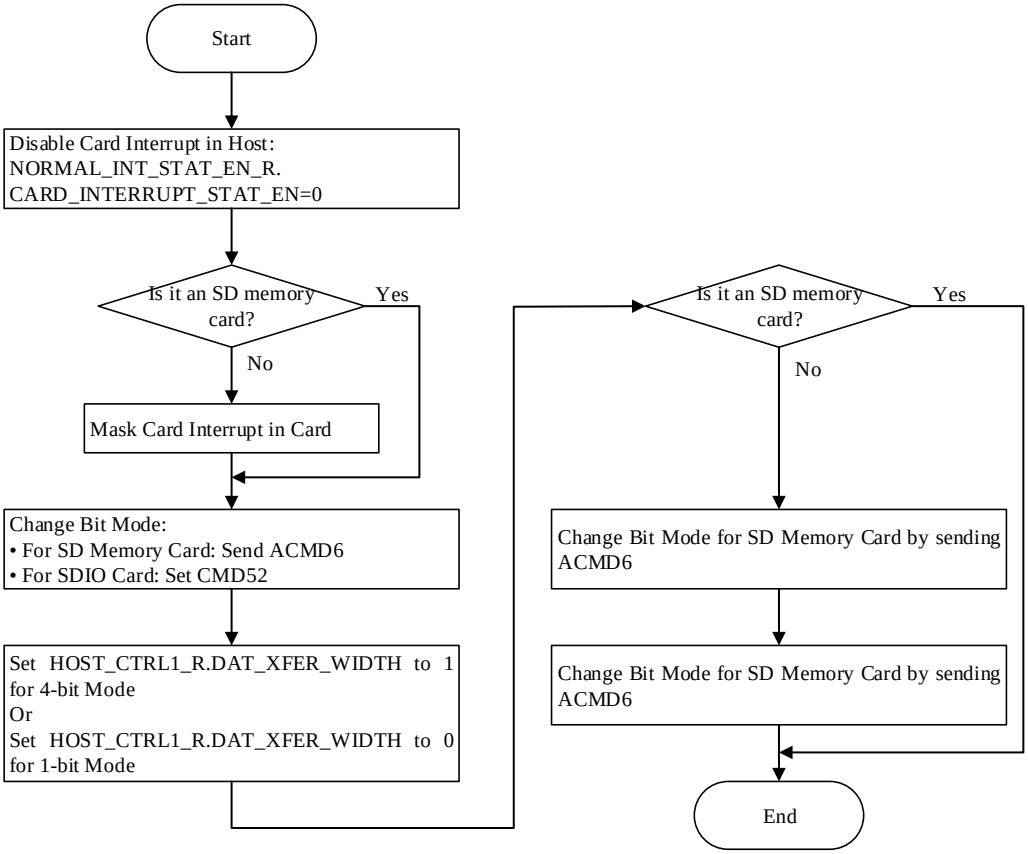

Figure & Table 3-21 Change bus width sequence

3.5.6.3 SD Bus Power Control

Figure & Table 3-22 shows the sequence for controlling the SD Bus Power.

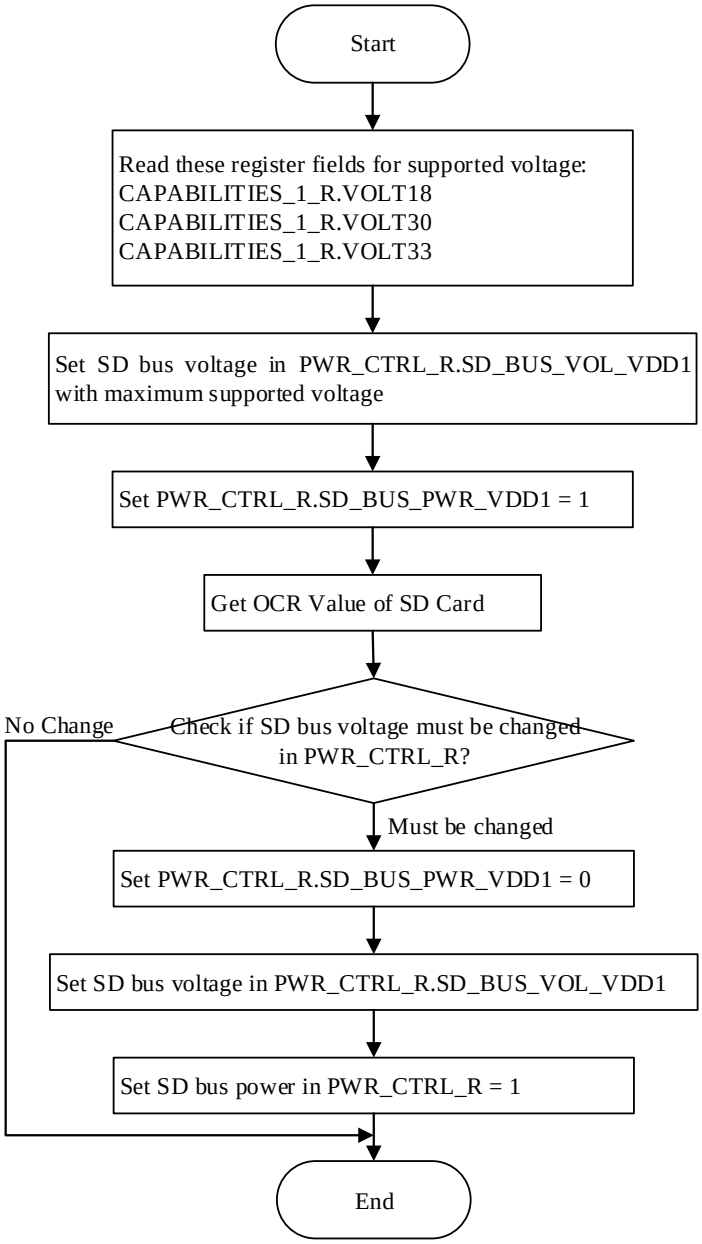

Figure & Table 3-22 SD bus power control sequence

3.5.6.4 Issuing CMD without Data Transfer

Figure & Table 3-23 shows the sequence on how to issue and complete an SD command. An SD command may or may not use a data line (DAT) (which is used for busy signaling). Figure & Table 3-23 shows the procedure to issue commands for both scenarios.

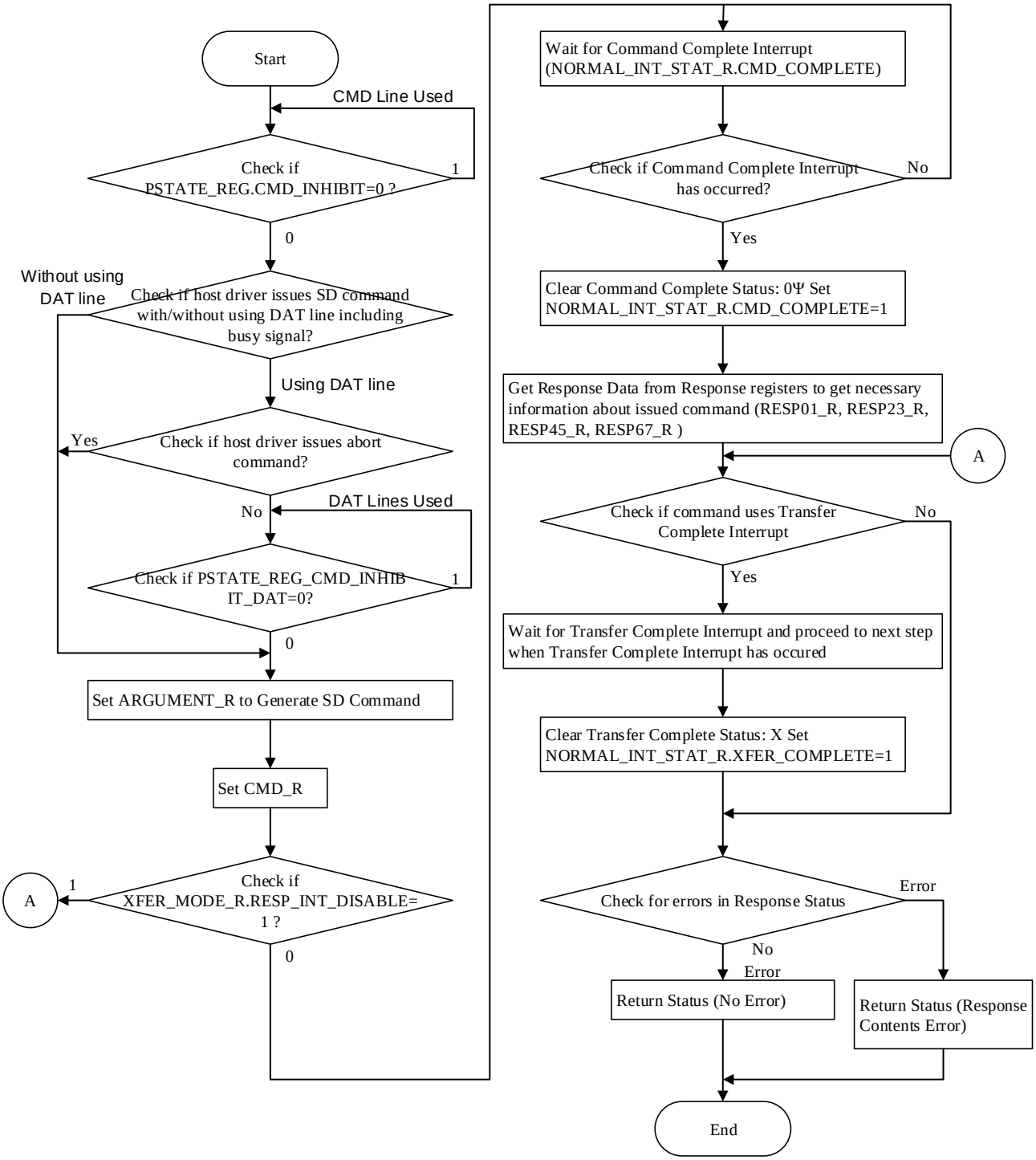

Figure & Table 3-23 SD command issue and complete

3.5.6.5 Issuing CMD with Data Transfer

This section includes the sequence to issues a command with data transfer without using DMA and by using SDMA, ADMA2, and ADMA3 modes.

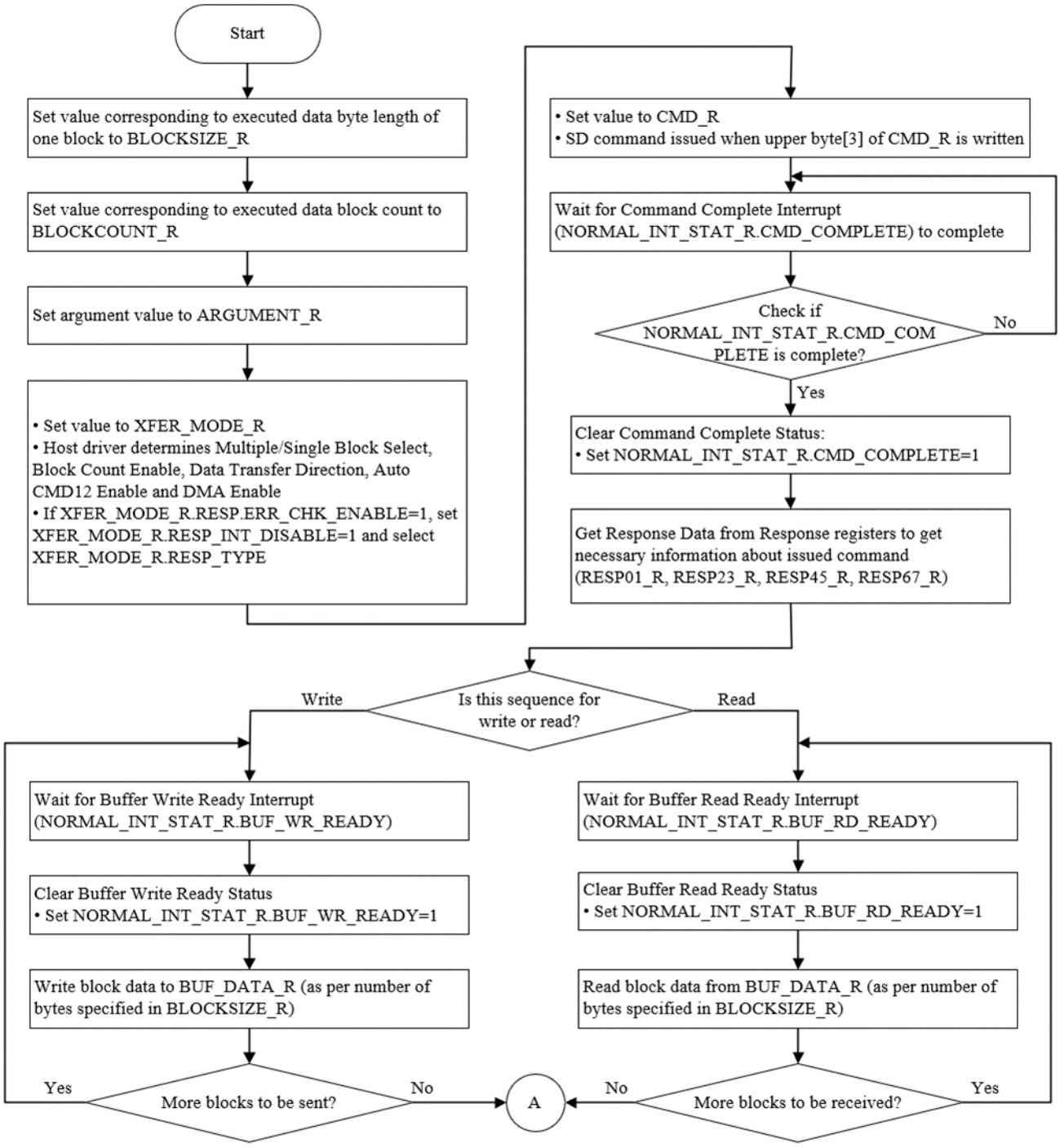

Figure & Table 3-24 Transaction control with data transfer (not using DMA)

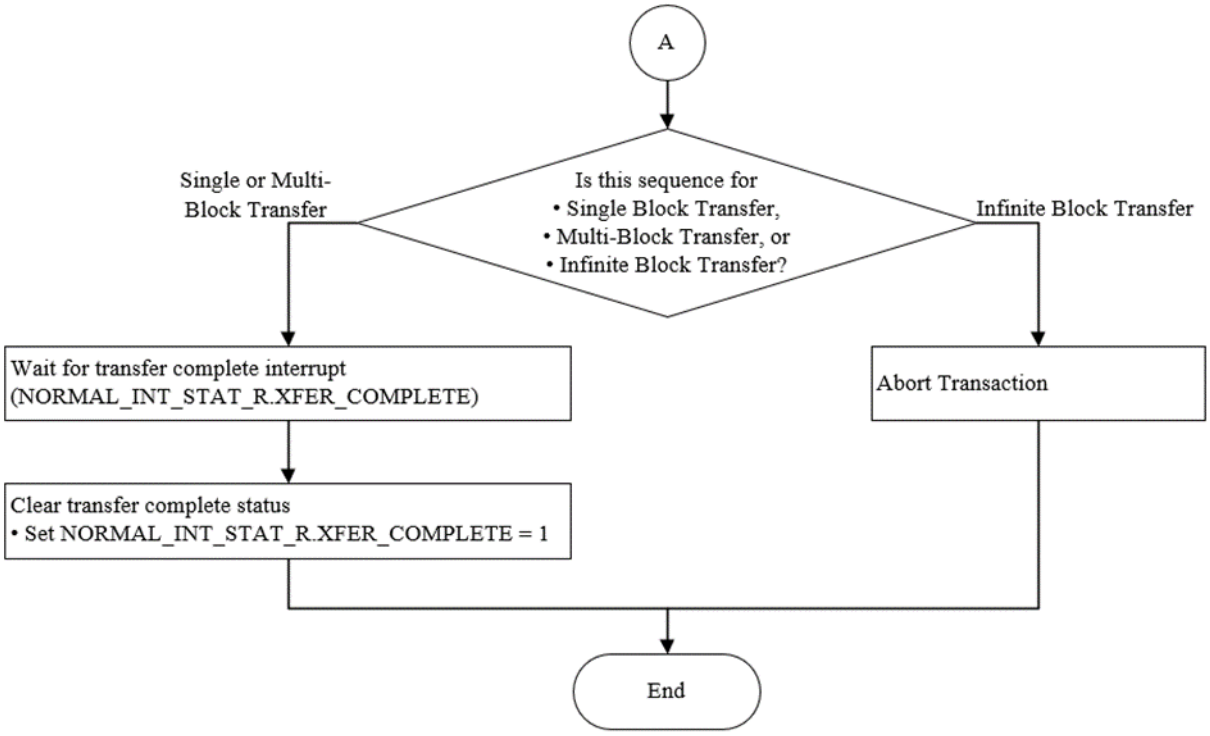

Figure & Table 3-25 Transaction control with data transfer (not using DMA, continued)

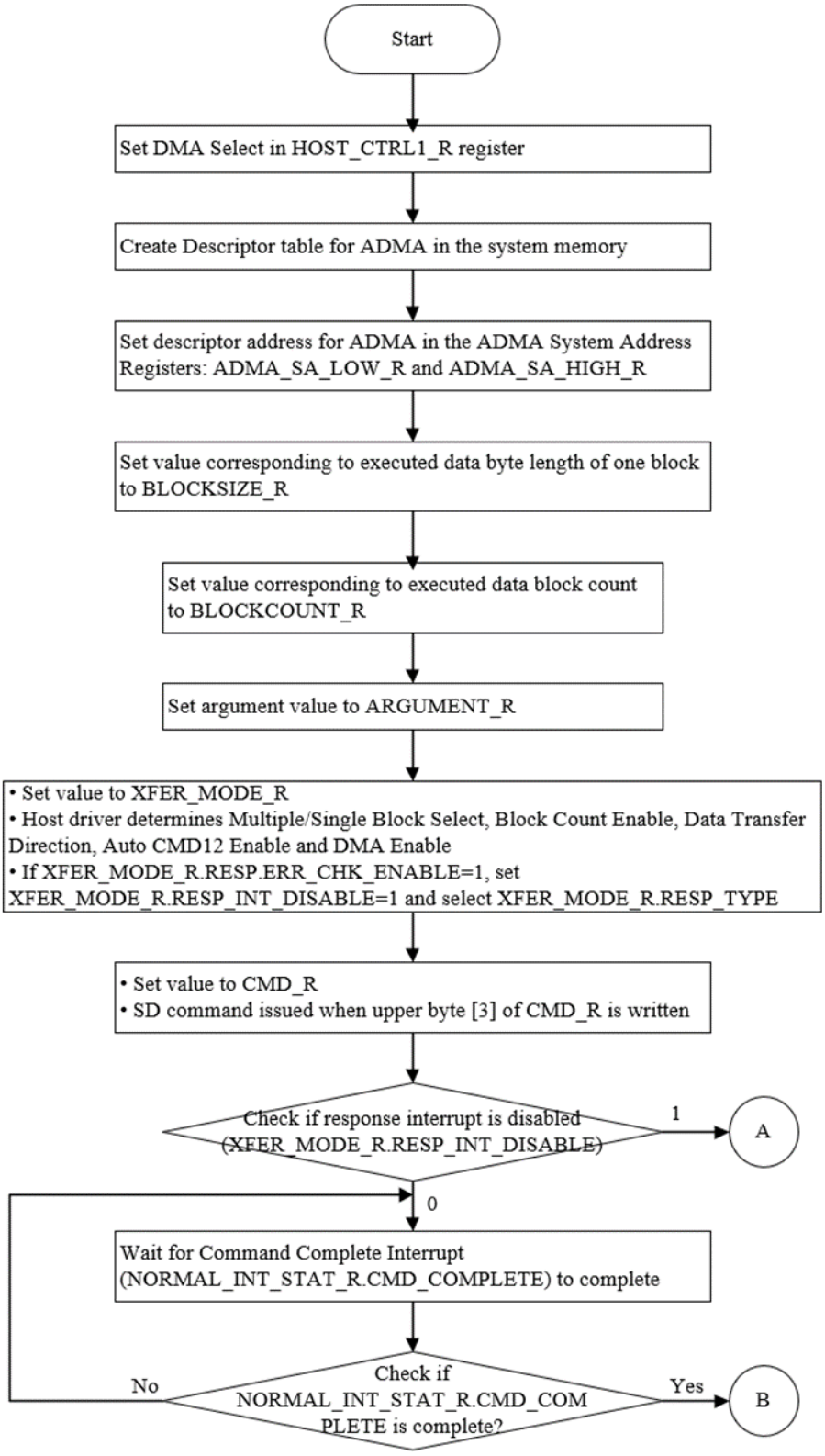

Figure & Table 3-26 Transaction control with data transfer (using ADMA2)

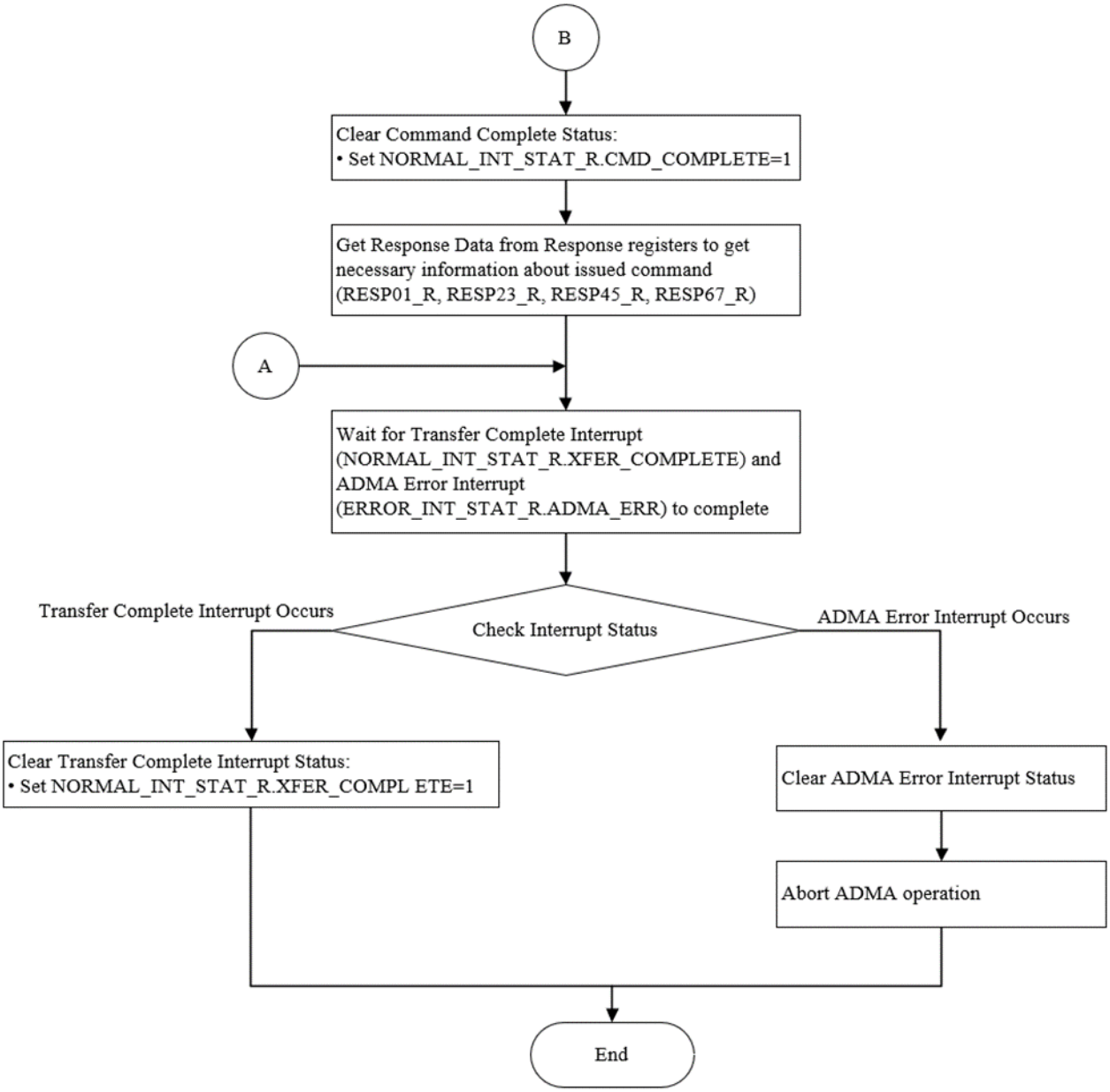

Figure & Table 3-27 Transaction control with data transfer (using ADMA2, continued)

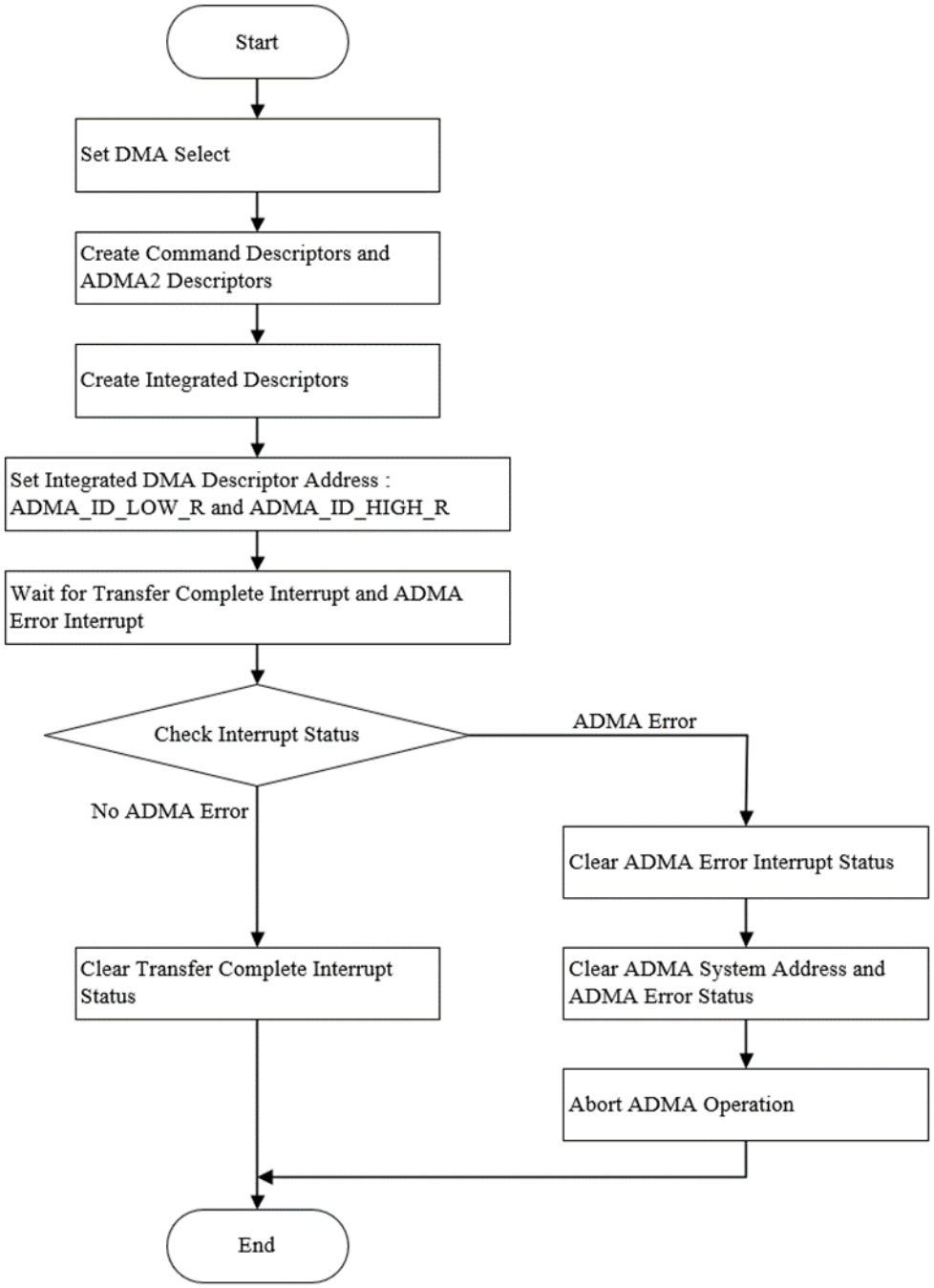

Figure & Table 3-28 Transaction control with data transfer (using ADMA3)

3.5.6.6 UHS-I Signal Voltage Switch Procedure

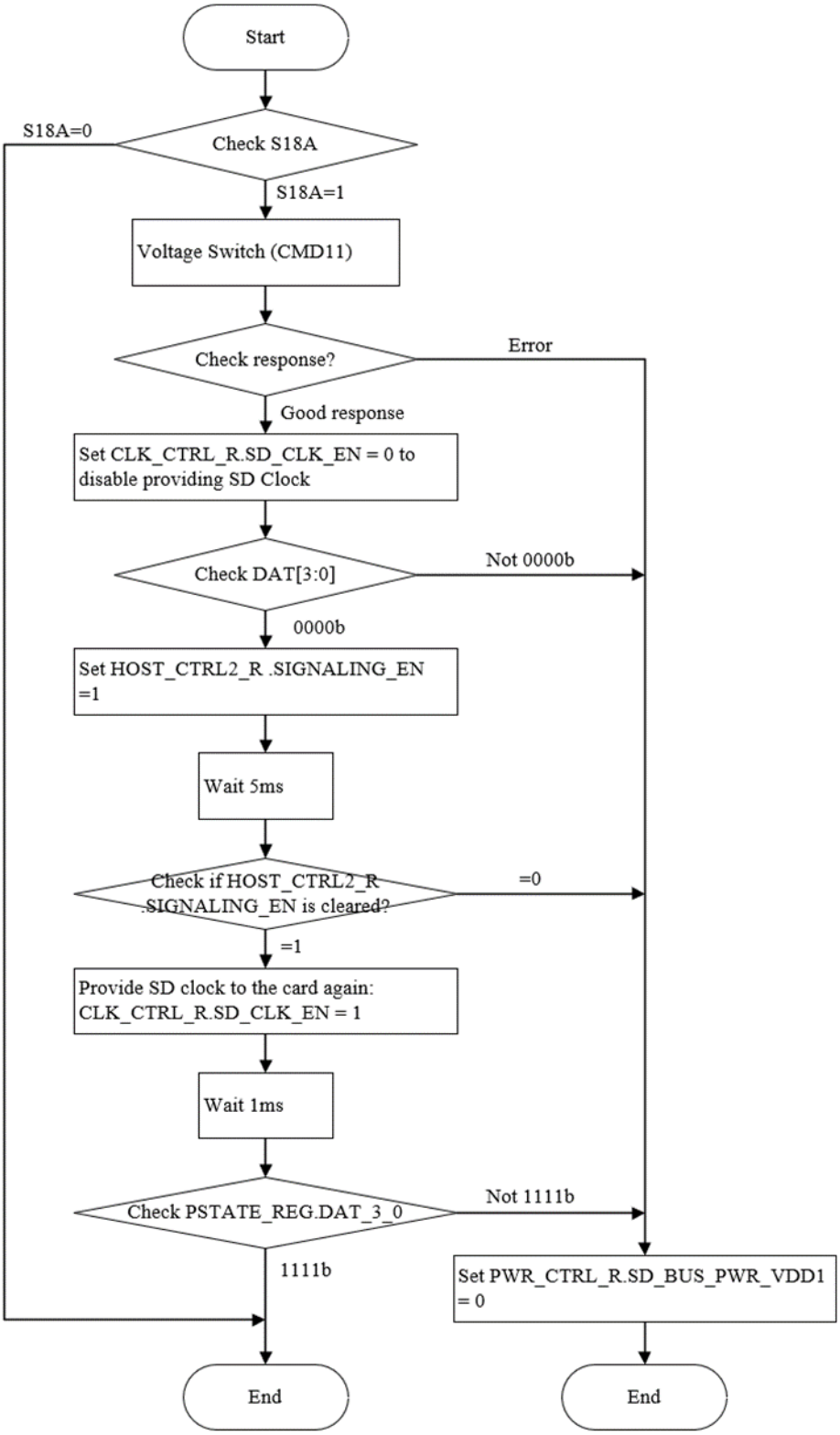

Figure & Table 3-29 Signal voltage switch procedure

3.5.6.7 Changing SD Bus Speed Mode

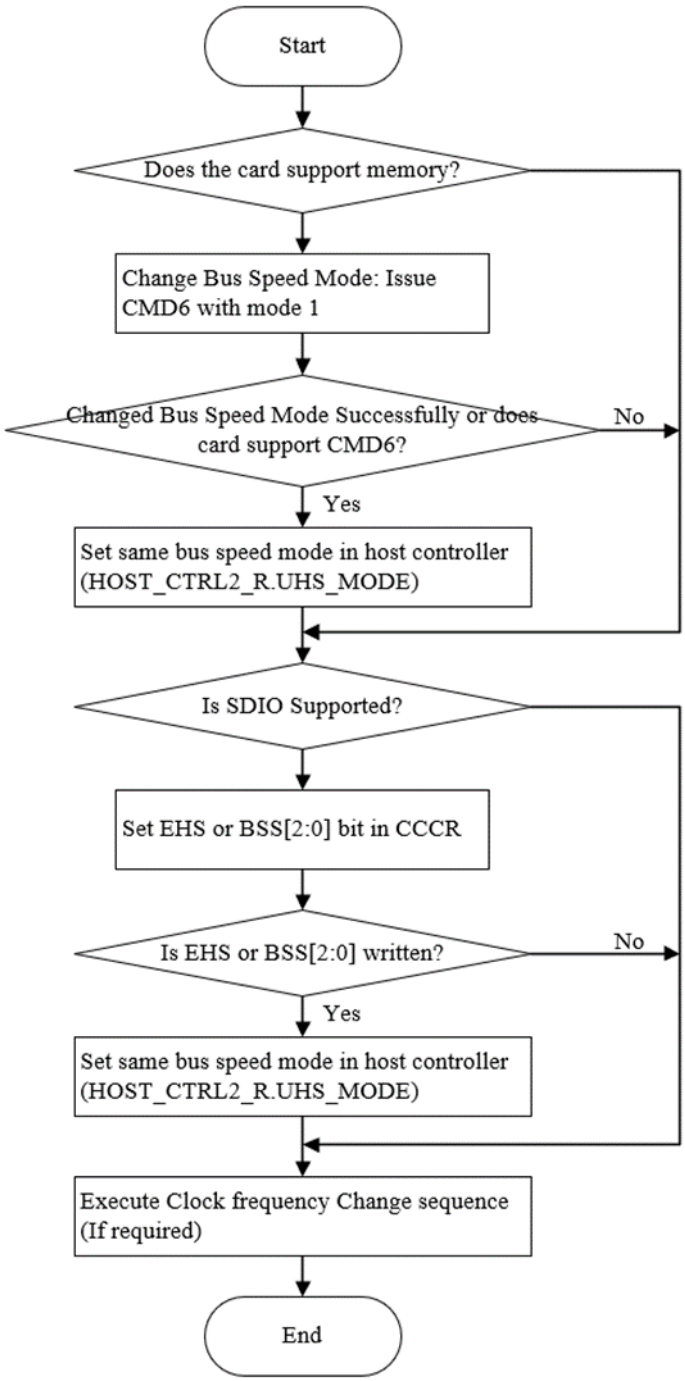

Figure & Table 3-30 Changing SD bus speed mode

3.5.6.8 SDIO Card Interrupt

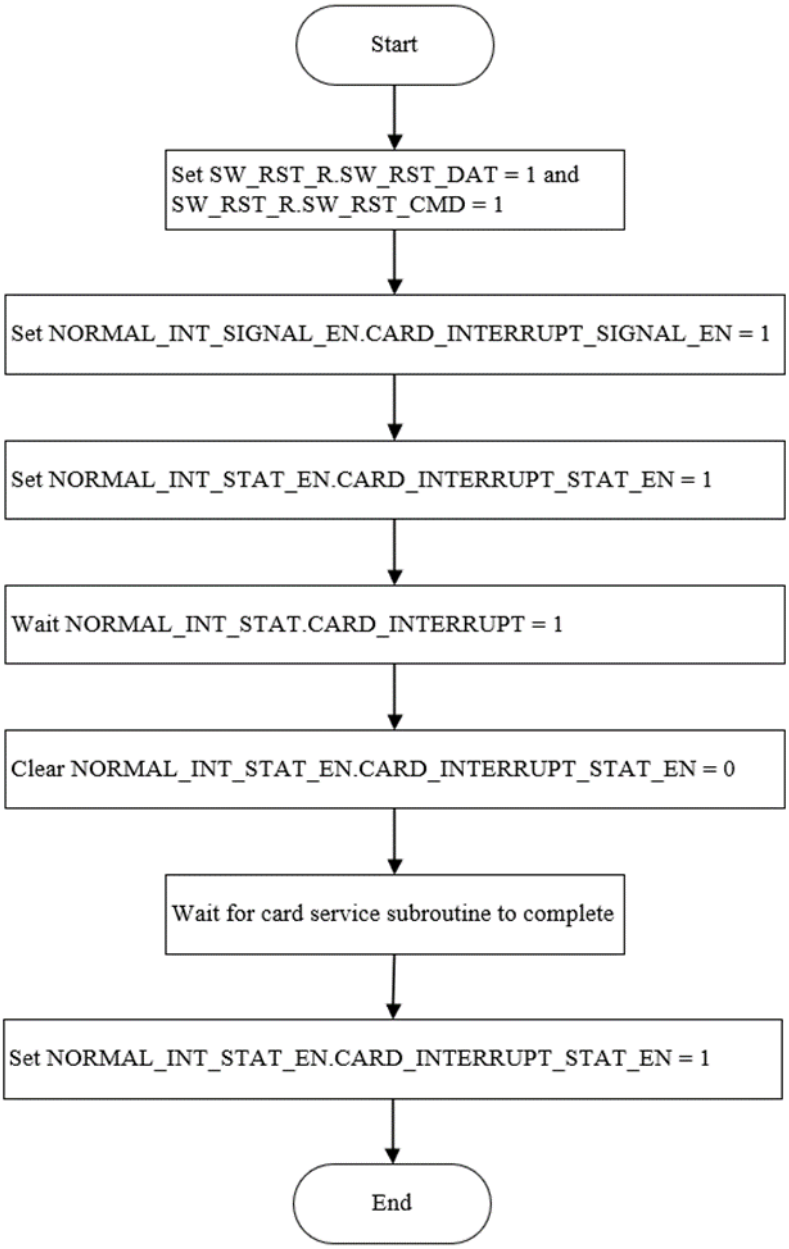

Figure & Table 3-31 SDIO card interrupt sequence

3.5.7 eMMC Transaction Mode

This section discusses the programming sequence for setting the host controller for an eMMC interface, the power sequence, initializing and identify the card, issuing a command with/without data transfer using or not using a DAT line, switching to various speed modes and to change the data bus width. This section also discusses how to program, initiate, and abort different boot speed modes.

3.5.7.1 Initializing and Identifying an eMMC Device

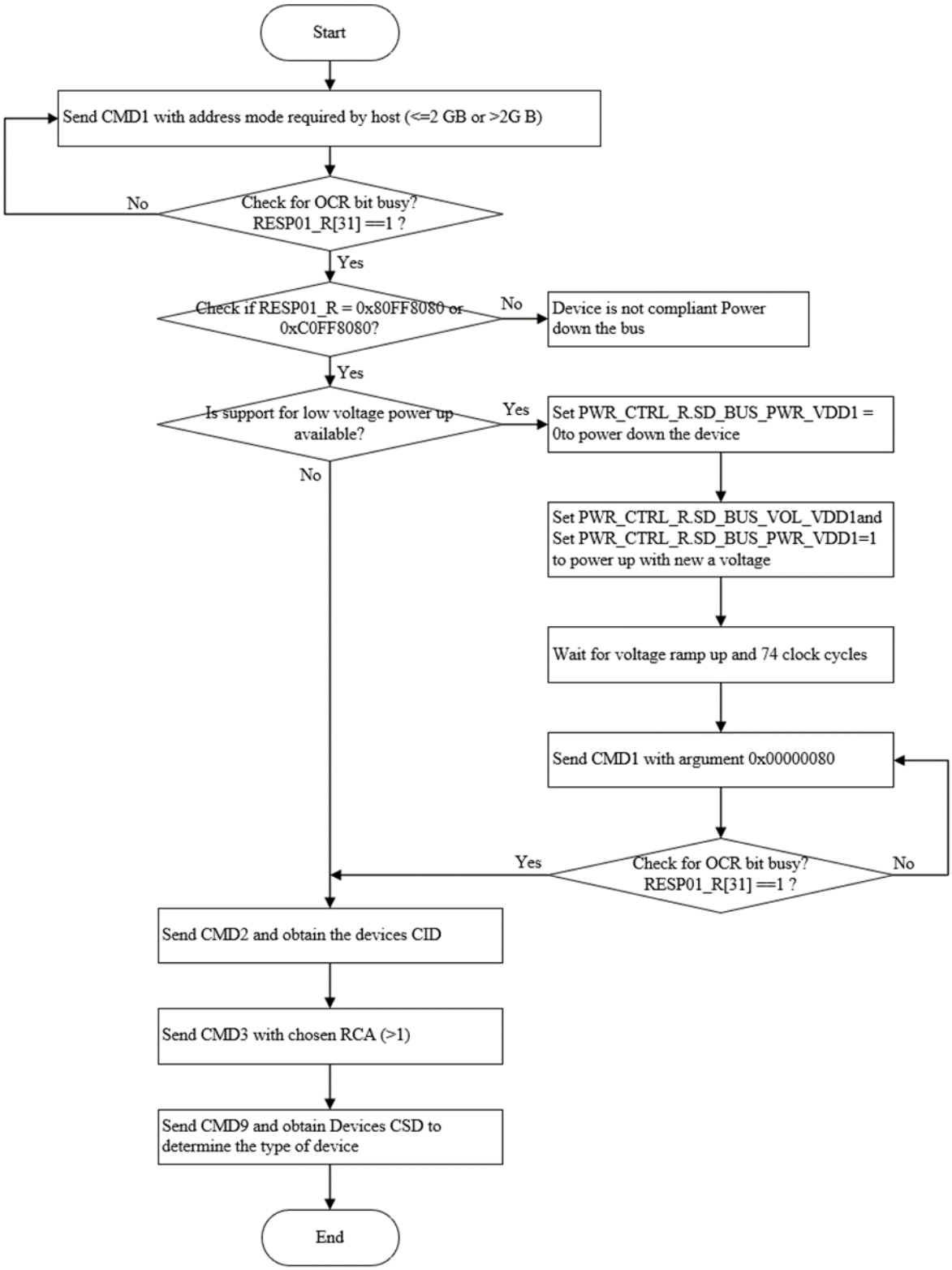

Figure & Table 3-32 Card initialization and identification programming sequence

3.5.7.2 Issue CMD without Data Transfer for an eMMC Device

The programming sequence for issuing a command without data transfer for an eMMC device is similar to the sequence mentioned in section "Issuing CMD without Data Transfer".

3.5.7.3 Issue CMD with Data Transfer for an eMMC Device

The programming sequence for issuing a command with data transfer for an eMMC device is similar to the programming sequence mentioned in section "Issuing CMD with Data Transfer". Data transfer can be done with or without using DMA. The supported DMA modes are SDMA, ADMA2, and ADMA3.

3.5.7.4 Switch to Various Speed Modes in an eMMC Device

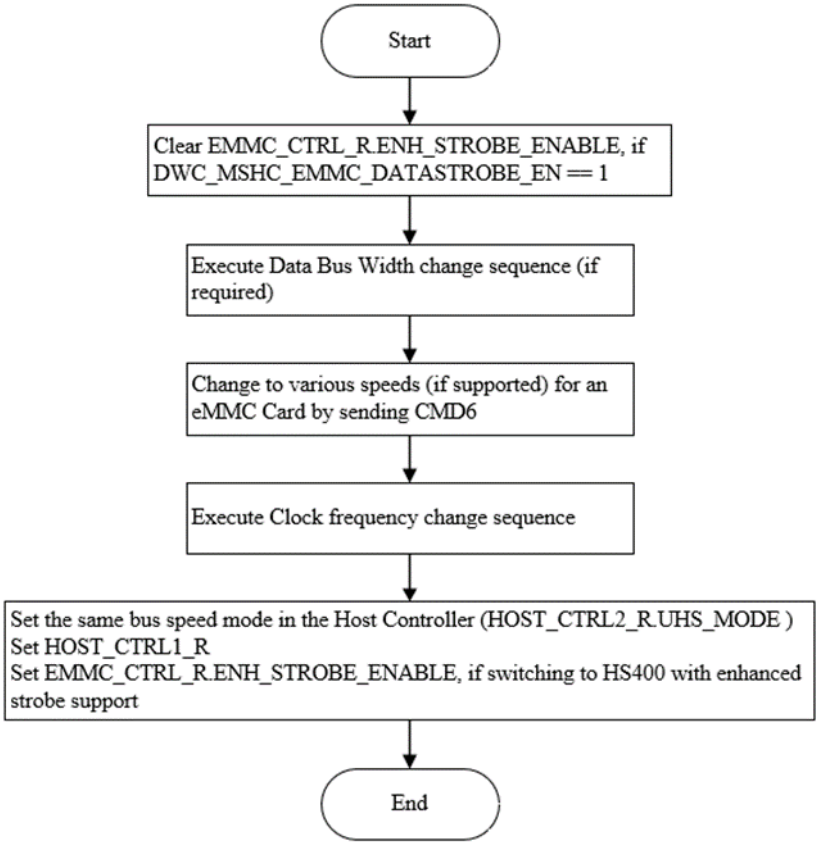

Figure & Table 3-33 Programming sequence to switch to various speed modes in an eMMC device

3.5.7.5 Changing the Data Bus Width for an eMMC Device

Figure & Table 3-34 shows the programming sequence to change the data bus width for an eMMC device. In Figure & Table 3-34, note that CMD6 is effective only during transfer state.

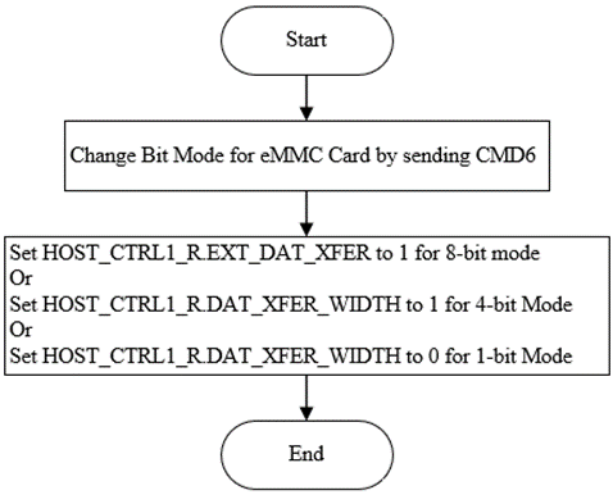

Figure & Table 3-34 Programming sequence to change data bus width for an eMMC device

3.5.7.6 Command Queuing

Figure & Table 3-35 shows the programming sequence to initialize the command queuing engine.

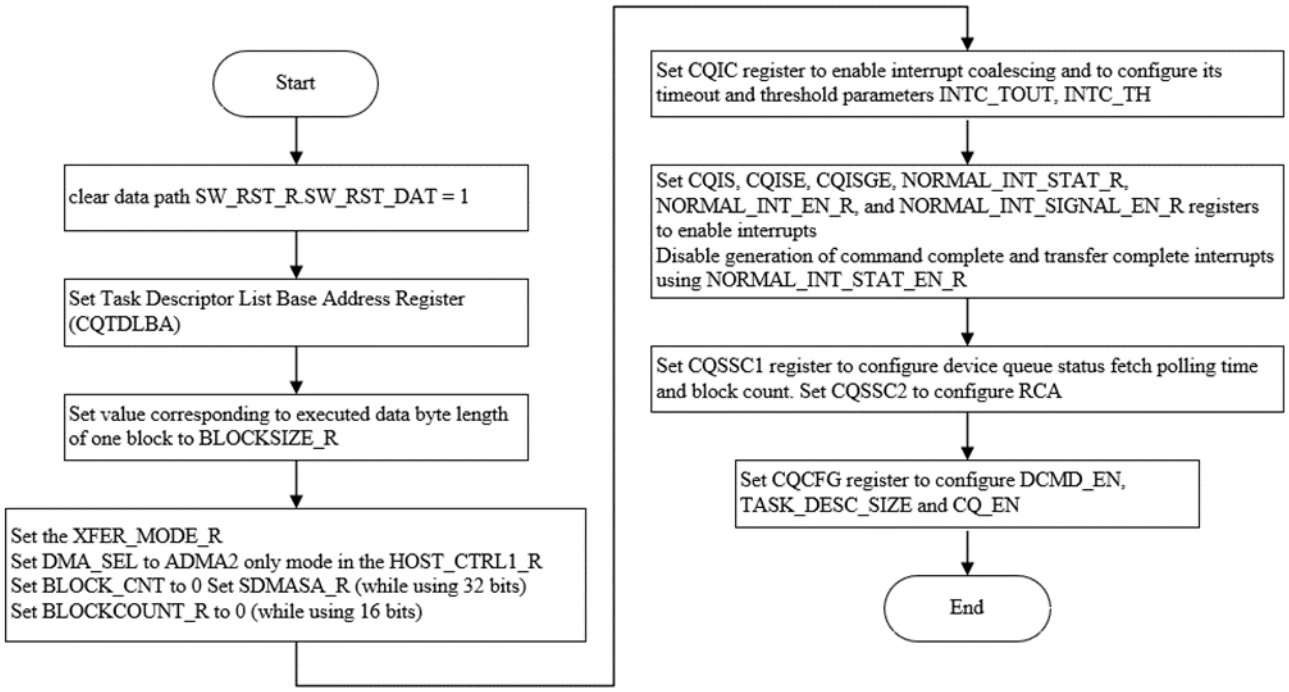

Figure & Table 3-35 Initializing the command queuing engine

Figure & Table 3-36 shows the sequence to submit and complete a task in a command queuing engine.

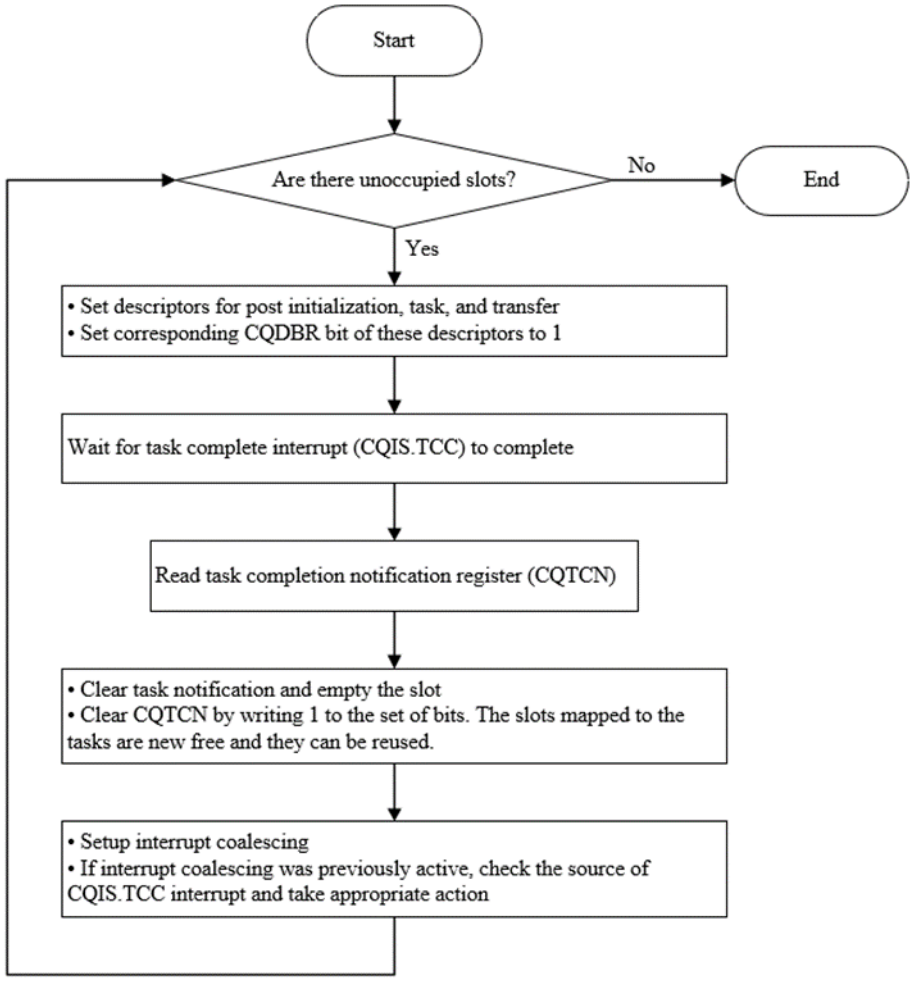

Figure & Table 3-36 Submitting and completing a task

3.5.7.7 Boot Programming Sequences

Figure & Table 3-37 shows the programming sequence to prepare for a boot. This sequence is common for both the mandatory and alternate modes.

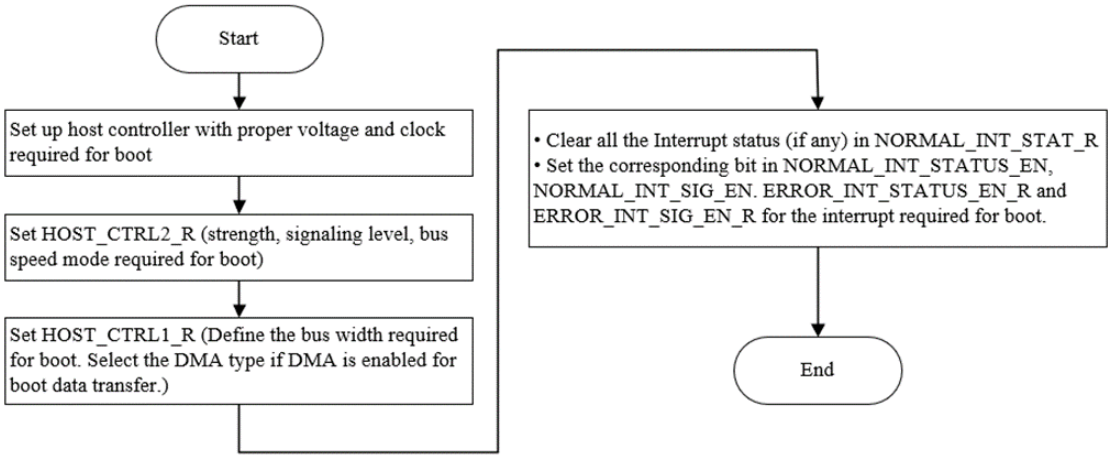

Figure & Table 3-37 Programming sequence to prepare for a boot

Figure & Table 3-38 shows the programming flow to initiate a mandatory boot in non-DMA mode.

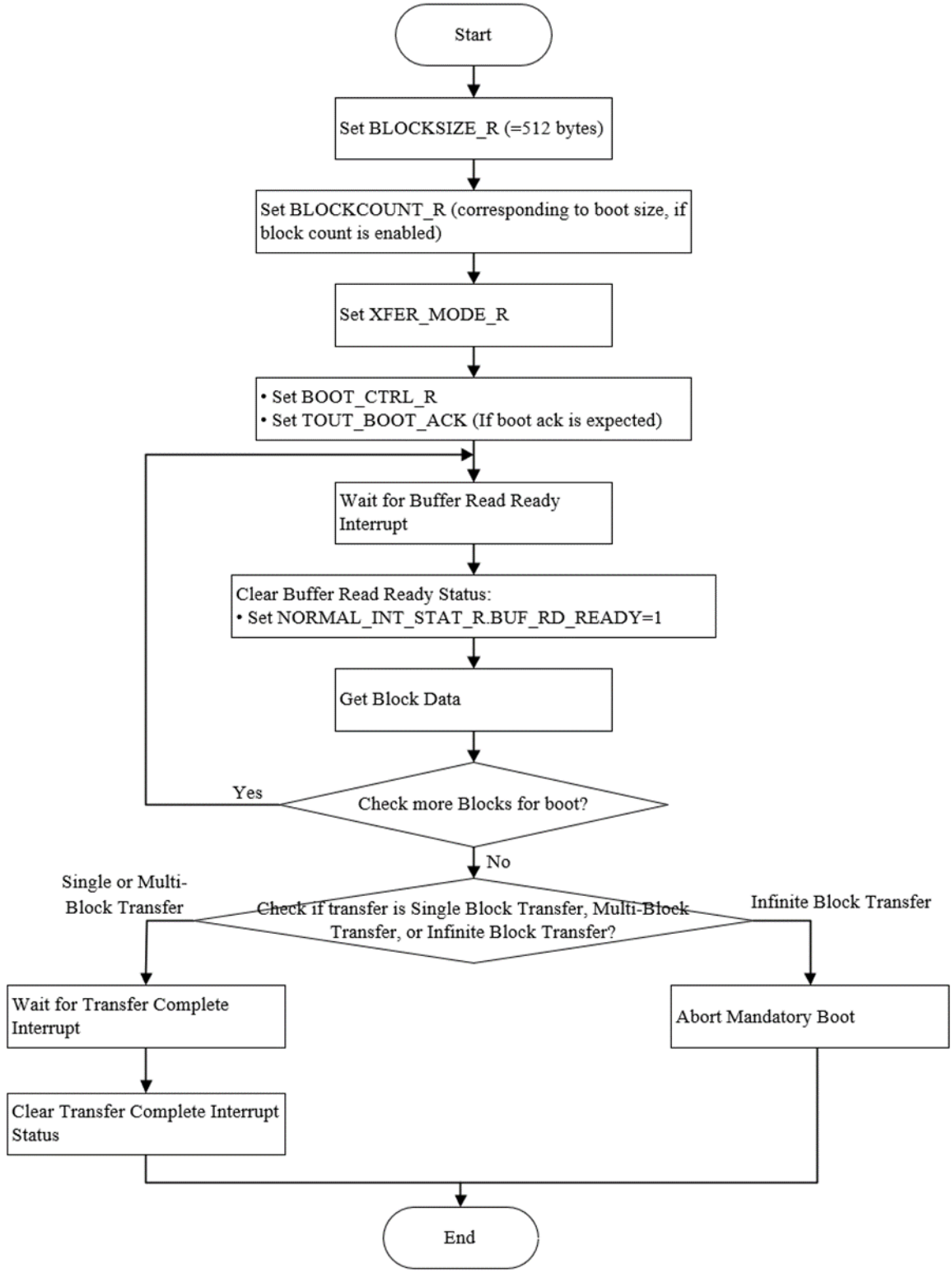

Figure & Table 3-38 Programming sequence for initiating a mandatory boot in non-DMA mode

Alternate Boot is a command-based boot operation and can be initiated by issuing CMD0 with the 0xFFFFFFFA argument. For the programming sequence for initiating an alternate boot, refer to "Issue CMD with Data Transfer for an eMMC Device" with the following exceptions:

- If boot acknowledgment is expected, BOOT_CTRL_R shall be programmed to set the timeout counter for boot acknowledgment before setting the CMD_R register.
- Response check cannot be performed as CMD0 has no response.

When the boot is initiated, device state is assumed to be in pre-boot state and 74 clock cycles has expired after power is stable before issuing CMD0.

3.5.8 SD/eMMC PHY Configuration

3.5.8.1 PHY Initial Power-up and Reset

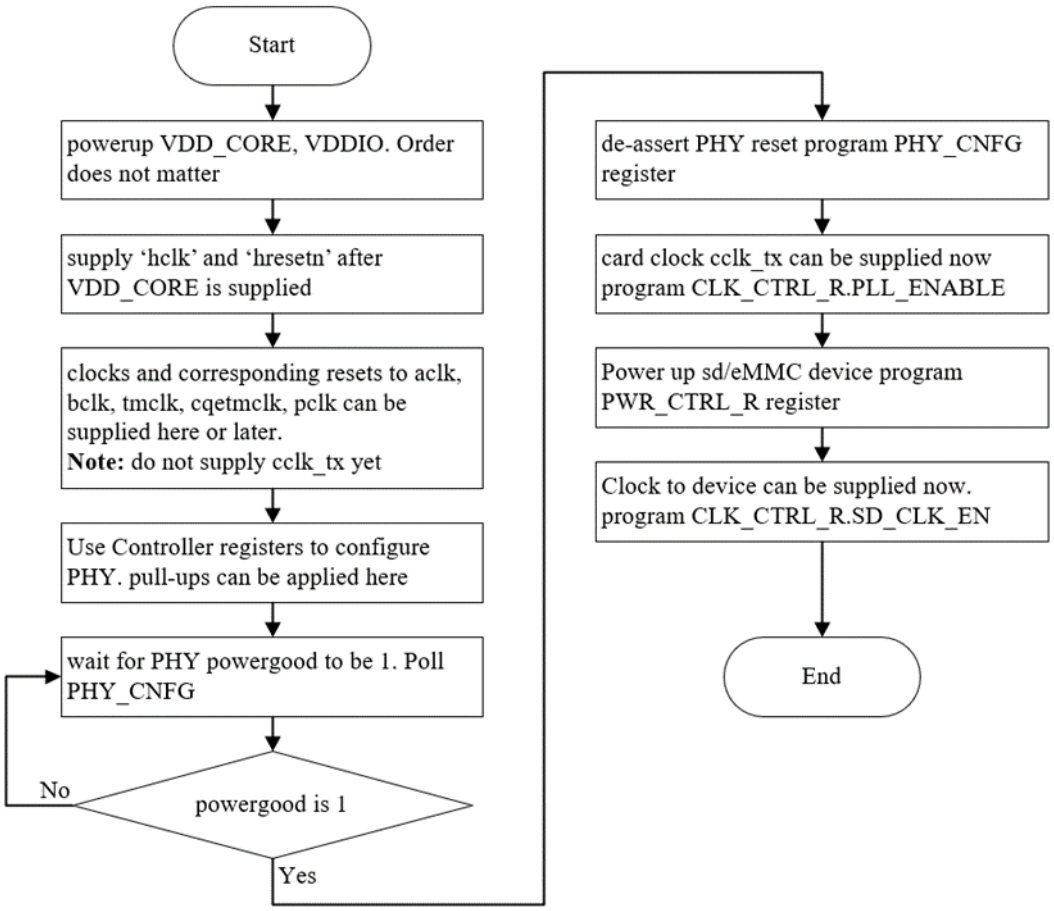

Figure & Table 3-39 PHY power-up and reset programming sequence

3.5.8.2 PAD Configuration Sequence

Use the recommended PAD configuration sequence when changing speed mode.

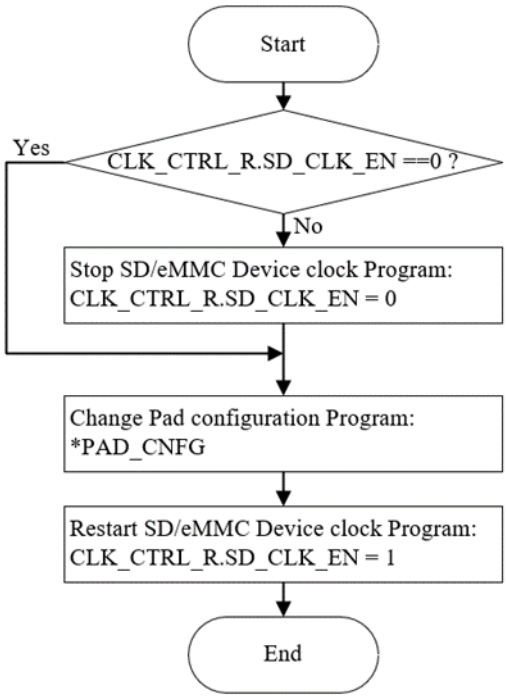

Figure & Table 3-40 PAD configuration sequence

Figure & Table 3-41 Recommended PAD settings for SD PHY in 1.8V mode

3.3V/1.8V PHY 1.8V Mode	TXSLEW_CTRL_N*	TXSLEW_CTRL_P*	WEAKPULL_EN	RXSEL	PAD_SN*	PAD_SP*
LEGACY	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS SDR	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS DDR	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS 200	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0	0x1	0x8	0x9
HS 400	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0 STB: 0x2	0x1	0x8	0x9

3.3V/1.8V PHY 1.8V Mode	TXSLEW_CTRL_N*	TXSLEW_CTRL_P*	WEAKPULL_EN	RXSEL	PAD_SN*	PAD_SP*
HS400 ENHANCED	0x1	0x3	DAT, CMD, RST: 0x1 CLK: 0x0 STB: 0x2	0x1	0x8	0x9

3.6 Registers

Figure & Table 3-42 Possible read and write behaviors

Read (or Write) Behavior	Description
RC	A read clears this register field.
RS	A read sets this register field.
RM	A read modifies the contents of this register field.
Wo	You can only write to this register once field.
W1C	A write of 1 clears this register field.
W1S	A write of 1 sets this register field.
W1T	A write of 1 toggles this register field.
W0C	A write of 0 clears this register field.
W0S	A write of 0 sets this register field.
W0T	A write of 0 toggles this register field.
WC	Any write clears this register field.
WS	Any write sets this register field.
WM	Any write toggles this register field.
no Read Behavior attribute	You cannot read this register. It is Write-Only.
no Write Behavior attribute	You cannot write to this register. It is Read-Only.

Figure & Table 3-43 Memory access examples

Memory Access	Description
R	Read-only register field.
W	Write-only register field.
R/W	Read/write register field.
R/W1C	You can read this register field. Writing 1 clears it.

RC/W1C	Reading this register field clears it. Writing 1 clears it.
R/Wo	You can read this register field. You can only write to it once.

3.6.1 DWC_mshc_map/DWC_mshc_block Registers

3.6.1.1 SDMASA_R

- Name: SDMA System Address Register
- Description: This register is used to configure a 32-bit block count or an SDMA system address based on the Host Version 4 Enable bit in the host control 2 register. This register is applicable for both SD and eMMC modes.
- Size: 32 bits
- Offset: 0x0

Figure & Table 3-44 Fields for register: SDMASA_R

Bits	Name	Access	Description
31:0	BLOCKCNT_SDMASA	R/W	32-bit Block Count (SDMA System Address) ■ SDMA System Address (Host Version 4 Enable = 0): This register contains the system memory address for an SDMA transfer in the 32-bit addressing mode. When the Host Controller stops an SDMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only if no transaction is executing. Reading this register during data transfers may return an invalid value. ■ 32-bit Block Count (Host Version 4 Enable = 1): From the Host Controller Version 4.10 specification, this register is redefined as 32-bit Block Count. The Host Controller decrements the block count of this register for every block transfer and the data transfer stops when the count reaches zero. This register must be accessed when no transaction is executing. Reading this register during data transfers may return invalid value. Following are the values for BLOCKCNT_SDMASA: ■ 0xFFFF_FFFF: 4G - 1 Block ■ ... ■ 0x0000_0002: 2 Blocks ■ 0x0000_0001: 1 Block ■ 0x0000_0000: Stop Count Note: ■ For Host Version 4 Enable = 0, the Host driver does not program the system address in this register while operating in ADMA mode. The system address must be programmed in the ADMA System Address

Bits	Name	Access	Description
			register. ■ For Host Version 4 Enable = 0, the Host driver programs a non-zero 32-bit block count value in this register when Auto CMD23 is enabled for non-DMA and ADMA modes. Auto CMD23 cannot be used with SDMA. ■ This register must be programmed with a non-zero value for data transfer if the 32-bit Block count register is used instead of the 16-bit Block count register. Value After Reset: 0x0 Exists: Always Volatile: true

3.6.1.2 BLOCKSIZE_R

- Name: Block Size Register
- Description: This register is used to configure an SDMA buffer boundary and the number of bytes in a data block. This register is applicable for both SD and eMMC modes.
- Size: 16 bits
- Offset: 0x4

Figure & Table 3-45 Fields for register: BLOCKSIZE_R

Bits	Name	Access	Description
15	RSVD_BLOCKSIZE15	R	This bit of the BLOCKSIZE_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
14:12	SDMA_BUF_BDARY	R/W	SDMA Buffer Boundary These bits specify the size of contiguous buffer in system memory. The SDMA transfer waits at every boundary specified by these fields and the Host Controller generates the DMA interrupt to request the Host Driver to update the SDMA System Address register. Values: ■ 0x0 (BYTES_4K): 4K bytes SDMA buffer boundary ■ 0x1 (BYTES_8K): 8K bytes SDMA buffer boundary ■ 0x2 (BYTES_16K): 16K bytes SDMA buffer boundary ■ 0x3 (BYTES_32K): 32K bytes SDMA buffer boundary ■ 0x4 (BYTES_64K): 64K bytes SDMA buffer boundary ■ 0x5 (BYTES_128K): 128K bytes SDMA buffer

Bits	Name	Access	Description
			boundary ■ 0x6 (BYTES_256K): 256K bytes SDMA buffer boundary ■ 0x7 (BYTES_512K): 512K bytes SDMA buffer boundary Value After Reset: 0x0 Exists: Always
11:0	XFER_BLOCK_SIZE	R/W	Transfer Block Size These bits specify the block size of data transfers. In case of memory, it is set to 512 bytes. It can be accessed only if no transaction is executing. Read operations during transfers may return an invalid value, and write operations are ignored. Following are the values for XFER_BLOCK_SIZE: ■ 0x1: 1 byte ■ 0x2: 2 bytes ■ 0x3: 3 bytes ■ ..... ■ 0x1FF: 511 bytes ■ 0x200: 512 bytes ■ ..... ■ 0x800: 2048 bytes Note: This register must be programmed with a non-zero value for data transfer. Value After Reset: 0x0 Exists: Always

3.6.1.3 BLOCKCOUNT_R

- Name: 16-bit Block Count Register
- Description: This register is used to configure the number of data blocks. This register is applicable for both SD and eMMC modes.
- Size: 16 bits
- Offset: 0x6

Figure & Table 3-46 Fields for register: BLOCKCOUNT_R

Bits	Name	Access	Description
15:0	BLOCK_CNT	R/W	16-bit Block Count ■ If the Host Version 4 Enable bit is set 0 or the 16-bit Block Count register is set to non-zero, the 16-bit Block Count register is selected. ■ If the Host Version 4 Enable bit is set 1 and the 16-bit Block Count register is set to zero, the 32-bit Block Count register is selected. Following are the values for BLOCK_CNT: ■ 0x0: Stop Count ■ 0x1: 1 Block ■ 0x2: 2 Blocks ■ ... - ... ■ 0xFFFF: 65535 Blocks Note: For Host Version 4 Enable = 0, this register must be set to 0000h before programming the 32-bit block count register when Auto CMD23 is enabled for non-DMA and ADMA modes. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.4 ARGUMENT_R

- Name: Argument Register
- Description: This register is used to configure the SD/eMMC command argument.
- Size: 32 bits
- Offset: 0x8

Figure & Table 3-47 Fields for register: ARGUMENT_R

Bits	Name	Access	Description
31:0	ARGUMENT	R/W	Command Argument These bits specify the SD/eMMC command argument that is specified in bits 39-8 of the Command format. Value After Reset: 0x0 Exists: Always

3.6.1.5 XFER_MODE_R

- Name: Transfer Mode Register

- Description: This register is used to control the operation of data transfers for an SD/eMMC mode. The Host driver sets this register before issuing a command that transfers data.
- Size: 16 bits
- Offset: 0xc

Figure & Table 3-48 Fields for register: XFER_MODE_R

Bits	Name	Access	Description
15:9	RSVD	R	These bits of the XFER_MODE_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
8	RESP_INT_DISABLE	R/W	Response Interrupt Disable The Host Controller supports response check function to avoid overhead of response error check by the Host driver. Response types of only R1 and R5 can be checked by the Controller. If Host Driver checks the response error, set this bit to 0 and wait for Command Complete Interrupt and then check the response register. If the Host Controller checks the response error, set this bit to 1 and set the Response Error Check Enable bit to 1. The Command Complete Interrupt is disabled by this bit regardless of the Command Complete Signal Enable. Note: During tuning (when the Execute Tuning bit in the Host Control2 register is set), the Command Complete Interrupt is not generated irrespective of the Response Interrupt Disable setting. Values: ■ 0x0 (ENABLED): Response Interrupt is enabled. ■ 0x1 (DISABLED): Response Interrupt is disabled. Value After Reset: 0x0 Exists: Always
7	RESP_ERR_CHK_ENABLE	R/W	Response Error Check Enable The Host Controller supports response check function to avoid overhead of response error check by Host driver. Response types of only R1 and R5 can be checked by the Controller. If the Host Controller checks the response error, set this bit to 1 and set Response Interrupt Disable to 1. If an error is detected, the Response Error interrupt is generated in the Error Interrupt Status register.

Bits	Name	Access	Description
			Note: ■ Response error check must not be enabled for any response type other than R1 and R5. ■ Response check must not be enabled for the tuning command. Values: ■ 0x0 (DISABLED): Response Error Check is disabled. ■ 0x1 (ENABLED): Response Error Check is enabled. Value After Reset: 0x0 Exists: Always
6	RESP_TYPE	R/W	Response Type R1/R5 This bit selects either R1 or R5 as a response type when the Response Error Check is selected. Error statuses checked in R1: ■ OUT_OF_RANGE ■ ADDRESS_ERROR ■ BLOCK_LEN_ERROR ■ WP_VIOLATION ■ CARD_IS_LOCKED ■ COM_CRC_ERROR ■ CARD_ECC_FAILED ■ CC_ERROR ■ ERROR Response Flags checked in R5: ■ COM_CRC_ERROR ■ ERROR ■ FUNCTION_NUMBER ■ OUT_OF_RANGE Values: ■ 0x0 (RESP_R1): R1 (memory) ■ 0x1 (RESP_R5): R5 (SDIO) Value After Reset: 0x0 Exists: Always
5	MULTI_BLK_SEL	R/W	Multi/Single Block Select This bit is set when issuing multiple-block transfer commands using the DAT line. If this bit is set to 0, it is

Bits	Name	Access	Description
			not necessary to set the Block Count register. Values: ■ 0x0 (SINGLE): Single block ■ 0x1 (MULTI): Multiple block Value After Reset: 0x0 Exists: Always
4	DATA_XFER_DIR	R/W	Data Transfer Direction Select This bit defines the direction of DAT line data transfers. This bit is set to 1 by the Host Driver to transfer data from the SD/eMMC card to the Host Controller and it is set to 0 for all other commands. Values: ■ 0x1 (READ): Read (card to host) ■ 0x0 (WRITE): Write (host to card) Value After Reset: 0x0 Exists: Always
3:2	AUTO_CMD_ENABLE	R/W	Auto Command Enable This field determines use of Auto Command functions. Note: In SDIO, this field must be set as 00b (Auto Command Disabled). Values: ■ 0x0 (AUTO_CMD_DISABLED): Auto command disabled. ■ 0x1 (AUTO_CMD12_ENABLED): Auto CMD12 enable. ■ 0x2 (AUTO_CMD23_ENABLED): Auto CMD23 enable. ■ 0x3 (AUTO_CMD_AUTO_SEL): Auto CMD auto select. Value After Reset: 0x0 Exists: Always
1	BLOCK_COUNT_ENABLE	R/W	Block Count Enable This bit is used to enable the Block Count register, which is relevant for multiple block transfers. If this bit is set to 0, the Block Count register is disabled, which is useful in executing an infinite transfer. The Host Driver must set this bit to 0 when ADMA is used. Values: ■ 0x1 (ENABLED): Enable ■ 0x0 (DISABLED): Disable

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
0	DMA_ENABLE	R/W	DMA Enable This bit enables the DMA functionality. If this bit is set to 1, a DMA operation begins when the Host Driver writes to the Command register. You can select one of the DMA modes by using DMA Select in the Host Control 1 register. Values: ■ 0x1 (ENABLED): DMA data transfer ■ 0x0 (DISABLED): No data transfer or non-DMA data transfer Value After Reset: 0x0 Exists: Always

3.6.1.6 CMD_R

- Name: Command Register
- Description: This register is used to provide the information related to a command and a response packet. This register is applicable for an SD/eMMC mode.
- Size: 16 bits
- Offset: 0xe

Figure & Table 3-49 Fields for register: CMD_R

Bits	Name	Access	Description
15:14	RSVD	R	These bits of the CMD_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
13:8	CMD_INDEX	R/W	Command Index These bits are set to the command number that is specified in bits 45-40 of the Command Format. Value After Reset: 0x0 Exists: Always
7:6	CMD_TYPE	R/W	Command Type These bits indicate the command type. Note: While issuing Abort CMD using CMD12/CMD52 or reset CMD using CMD0/CMD52, CMD_TYPE field shall be set to 0x3.

Bits	Name	Access	Description
			Values: ■ 0x3 (ABORT_CMD): Abort ■ 0x2 (RESUME_CMD): Resume ■ 0x1 (SUSPEND_CMD): Suspend ■ 0x0 (NORMAL_CMD): Normal Value After Reset: 0x0 Exists: Always
5	DATA_PRESENT_SEL	R/W	Data Present Select This bit is set to 1 to indicate that data is present and that the data is transferred using the DAT line. This bit is set to 0 in the following instances: ■ Command using the CMD line ■ Command with no data transfer but using busy signal on the DAT[0] line ■ Resume command Values: ■ 0x0 (NO_DATA): No data present. ■ 0x1 (DATA): Data present. Value After Reset: 0x0 Exists: Always
4	CMD_IDX_CHK_ENABLE	R/W	Command Index Check Enable This bit enables the Host Controller to check the index field in the response to verify if it has the same value as the command index. If the value is not the same, it is reported as a Command Index error. Note: ■ Index Check enable must be set to 0 for the command with no response, R2 response, R3 response and R4 response. ■ For the tuning command, this bit must always be set to enable the index check. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable Value After Reset: 0x0 Exists: Always
3	CMD_CRC_CHK_ENABLE	R/W	Command CRC Check Enable

Bits	Name	Access	Description
			This bit enables the Host Controller to check the CRC field in the response. If an error is detected, it is reported as a Command CRC error. Note: ■ CRC Check enable must be set to 0 for the command with no response, R3 response, and R4 response. ■ For the tuning command, this bit must always be set to 1 to enable the CRC check. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable Value After Reset: 0x0 Exists: Always
2	SUB_CMD_FLAG	R/W	Sub Command Flag This bit distinguishes between a main command and a sub command. Values: ■ 0x0 (MAIN): Main command ■ 0x1 (SUB): Sub command Value After Reset: 0x0 Exists: Always
1:0	RESP_TYPE_SELECT	R/W	Response Type Select This bit indicates the type of response expected from the card. Values: ■ 0x0 (NO_RESP): No response ■ 0x1 (RESP_LEN_136): Response length 136 ■ 0x2 (RESP_LEN_48): Response length 48 ■ 0x3 (RESP_LEN_48B): Response length 48; Check Busy after response Value After Reset: 0x0 Exists: Always

3.6.1.7 RESP01_R

- Name: Response Register 0/1
- Description: This register stores bits 39-08 of the response field for an SD/eMMC mode. In UHS-II mode, this register stores the response of the TRANS_ABORT CCMD. The response for an

SD/eMMC command can be a maximum of 128 bits. These 128 bits are segregated into four 32-bit registers: RESP01_R, RESP23_R, RESP45_R and RESP67_R.

- Size: 32 bits
- Offset: 0x10

Figure & Table 3-50 Fields for register: RESP01_R

Bits	Name	Access	Description
31:0	RESP01	R	Command Response These bits reflect 39-8 bits of SD/eMMC Response Field. In UHS-II mode, it stores the response of the TRANS_ABORT CCMD. Note: For Auto CMD, the 32-bit response (bits 39-8 of the Response Field) is updated in the RESP67_R register. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.8 RESP23_R

- Name: Response Register 2/3
- Description: This register stores bits 71-40 of the Response Field for an SD/eMMC mode. This register is used to store the response from the cards. The response can be a maximum of 128 bits.
  These 128 bits are segregated into four 32-bit registers: RESP01_R, RESP23_R, RESP45_R and RESP67_R. In UHS-II mode, this register is reserved.
- Size: 32 bits
- Offset: 0x14

Figure & Table 3-51 Fields for register: RESP23_R

Bits	Name	Access	Description
31:0	RESP23	R	Command Response These bits reflect 71-40 bits of the SD/eMMC response field. In UHS-II mode, it is reserved. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.9 RESP45_R

- Name: Response Register 4/5

- Description: This register stores bits 103-72 of the response field for an SD/eMMC mode. In UHS-II mode, this register is used to store the lower 4-byte CMD12 response. The response for SD/eMMC command can be a maximum of 128 bits. These 128 bits are segregated into four 32-bit registers: RESP01_R, RESP23_R, RESP45_R and RESP67_R.
- Size: 32 bits
- Offset: 0x18

Figure & Table 3-52 Fields for register: RESP45_R

Bits	Name	Access	Description
31:0	RESP45	R	Command Response These bits reflect bits 103-72 of the response field. In UHSII mode, it stores the lower 4-byte CMD12 response. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.10 RESP67_R

- Name: Response Register 6/7
- Description: This register stores bits 135-104 of the response field for an SD/eMMC mode. In UHSII mode, this register stores the upper 4-byte CMD12 response. The SD/eMMC response can be a maximum of 128 bits. These 128 bits are segregated into four 32-bit registers: RESP01_R, RESP23_R, RESP45_R and RESP67_R.
- Size: 32 bits
- Offset: 0x1c

Figure & Table 3-53 Fields for register: RESP67_R

Bits	Name	Access	Description
31:0	RESP67	R	Command Response These bits reflect bits 135-104 of SD/eMMC response field. In UHS-II mode, it stores the upper 4-byte CMD12 response. Note: For Auto CMD, this register also reflects the 32-bit response (bits 39-8 of the Response Field). Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.11 BUF_DATA_R

- Name: Buffer Data Port Register

- Description: This register is used to access the packet buffer. This register is applicable for an SD/eMMC/UHS-II mode.
- Size: 32 bits
- Offset: 0x20

Figure & Table 3-54 Fields for register: BUF_DATA_R

Bits	Name	Access	Description
31:0	BUF_DATA	R/W	Buffer Data These bits enable access to the Host Controller packet buffer. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.12 PSTATE_REG

- Name: Present State Register
- Description: This register indicates the present status of the host controller. This register is applicable for an SD/eMMC/UHS-II mode.
- Size: 32 bits
- Offset: 0x24

Figure & Table 3-55 Fields for register: PSTATE_REG

Bits	Name	Access	Description
31	UHS2_IF_DETECT	R	UHS-II Interface Detection This bit indicates whether a card supports the UHS-II interface. For SD/eMMC mode, this bit always returns 0. Values: ■ 0x0 (FALSE): UHS-II interface is not detected. ■ 0x1 (TRUE): UHS-II interface is detected. Value After Reset: 0x0 Exists: Always Volatile: True
30	LANE_SYNC	R	Lane Synchronization This bit indicates whether a lane is synchronized in the UHSII mode. For SD/eMMC mode, this bit always returns 0. Values: ■ 0x0 (FALSE): UHS-II PHY is not initialized. ■ 0x1 (TRUE): UHS-II PHY is initialized.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always Volatile: True
29	IN_DORMANT_ST	R	In Dormant Status This bit indicates whether UHS-II lanes enter dormant state in the UHS-II mode. For SD/eMMC mode, this bit always returns 0. Values: ■ 0x0 (FALSE): Not in dormant state ■ 0x1 (TRUE): In dormant state Value After Reset: =mshc_present_state_in_dormant_reset_val Exists: Always Volatile: True
28	SUB_CMD_STAT	R	Sub Command Status This bit is used to distinguish between a main command and a sub command status. Values: ■ 0x0 (FALSE): Main command status ■ 0x1 (TRUE): Sub command status Value After Reset: 0x0 Exists: Always Volatile: True
27	CMD_ISSUE_ERR	R	Command Not Issued by Error This bit is set if a command cannot be issued after setting the command register due to an error except the Auto CMD12 error. Values: ■ 0x0 (FALSE): No error for issuing a command. ■ 0x1 (TRUE): Command cannot be issued. Value After Reset: 0x0 Exists: Always Volatile: True
26	RSVD_26	R	This bit of the PRESENT_ST_R register is reserved bits. It always returns 0. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always Volatile: True
25	HOST_REG_VOL	R	Host Regulator Voltage Stable This bit is used to check whether the host regulator voltage is stable for switching the voltage of UHS-I mode. This bit reflects the synchronized value of the host_reg_vol_stable signal. Values: ■ 0x0 (FALSE): Host regulator voltage is not stable. ■ 0x1 (TRUE): Host regulator voltage is stable. Value After Reset: 0x0 Exists: Always Volatile: True
24	CMD_LINE_LVL	R	Command-Line Signal Level This bit is used to check the CMD line level to recover from errors and for debugging. These bits reflect the value of the sd_cmd_in signal. This bit is irrelevant for an UHS-II mode and always returns 0. Value After Reset: 0x0 Exists: Always Volatile: True
23:20	DAT_3_0	R	DAT[3:0] Line Signal Level This bit is used to check the DAT line level to recover from errors and for debugging. These bits reflect the value of the sd_dat_in (lower nibble) signal. These bits are irrelevant for the UHS-II mode and always returns 0. Value After Reset: 0x0 Exists: Always Volatile: True
19	WR_PROTECT_SW_LVL	R	Write Protect Switch Pin Level This bit is supported only for memory and combo cards. This bit reflects the synchronized value of the card_write_prot signal. Values: ■ 0x0 (FALSE): Write protected. ■ 0x1 (TRUE): Write enabled.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always Volatile: True
18	CARD_DETECT_PIN_LEVEL	R	Card Detect Pin Level This bit reflects the inverse synchronized value of the card_detect_n signal. Values: ■ 0x0 (FALSE): No card present. ■ 0x1 (TRUE): Card present. Value After Reset: 0x0 Exists: Always Volatile: True
17	CARD_STABLE	R	Card Stable This bit indicates the stability of the Card Detect Pin Level. A card is not detected if this bit is set to 1 and the value of the CARD_INSERTED bit is 0. Values: ■ 0x0 (FALSE): Reset or Debouncing ■ 0x1 (TRUE): No card or inserted Value After Reset: 0x0 Exists: Always Volatile: True
16	CARD_INSERTED	R	Card Inserted This bit indicates whether a card has been inserted. The host controller debounces this signal so that host driver need not wait for it to stabilize. Values: ■ 0x0 (FALSE): Reset, Debouncing, or No card ■ 0x1 (TRUE): Card inserted Value After Reset: 0x0 Exists: Always Volatile: True
15:12	RSVD_15_12	R	These bits of the PRESENT_STAT_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
			Volatile: True
11	BUF_RD_ENABLE	R	Buffer Read Enable This bit is used for non-DMA transfers. This bit is set if valid data exists in the Host buffer. Values: ■ 0x0 (DISABLED): Read disable. ■ 0x1 (ENABLED): Read enable. Value After Reset: 0x0 Exists: Always Volatile: True
10	BUF_WR_ENABLE	R	Buffer Write Enable This bit is used for non-DMA transfers. This bit is set if space is available for writing data. Values: ■ 0x0 (DISABLED): Write disable. ■ 0x1 (ENABLED): Write enable. Value After Reset: 0x0 Exists: Always Volatile: True
9	RD_XFER_ACTIVE	R	Read Transfer Active This bit indicates whether a read transfer is active for SD/eMMC mode. This bit irrelevant for the UHS-II mode and always returns 0. Values: ■ 0x0 (INACTIVE): No valid data ■ 0x1 (ACTIVE): Transferring data Value After Reset: 0x0 Exists: Always Volatile: True
8	WR_XFER_ACTIVE	R	Write Transfer Active This status indicates whether a write transfer is active for SD/eMMC mode. It is irrelevant for UHS-II mode and always returns 0. Values: ■ 0x0 (INACTIVE): No valid data ■ 0x1 (ACTIVE): Transferring data

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always Volatile: True
7:4	DAT_7_4	R	DAT[7:4] Line Signal Level This bit is used to check the DAT line level to recover from errors and for debugging. These bits reflect the value of the sd_dat_in (upper nibble) signal. These bits are irrelevant for the UHS-II mode and always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
3	RE_TUNE_REQ	R	Re-Tuning Request DWC_mshc does not generate retuning request. The software must maintain the Retuning timer. Value After Reset: 0x0 Exists: Always Volatile: True
2	DAT_LINE_ACTIVE	R	DAT Line Active (SD/eMMC Mode only) This bit indicates whether one of the DAT lines on the SD/eMMC bus is in use. This bit is irrelevant for the UHS-II mode. In the case of read transactions, this bit indicates whether a read transfer is executing on the SD/eMMC bus. In the case of write transactions, this bit indicates whether a write transfer is executing on the SD/eMMC bus. For a command with busy, this status indicates whether the command executing busy is executing on an SD or eMMC bus. Values: ■ 0x0 (INACTIVE): DAT line inactive ■ 0x1 (ACTIVE): DAT line active Value After Reset: 0x0 Exists: Always Volatile: True
1	CMD_INHIBIT_DAT	R	Command Inhibit (DAT) This bit is applicable for SD/eMMC mode and is

Bits	Name	Access	Description
			generated if either DAT line active or Read transfer active is set to 1. If this bit is set to 0, it indicates that the Host Controller can issue subsequent SD/eMMC commands. For the UHS-II mode, this bit is irrelevant and always returns 0. Values: ■ 0x0 (READY): Can issue command which used DAT line. ■ 0x1 (NOT_READY): Cannot issue command which used DAT line. Value After Reset: 0x0 Exists: Always Volatile: True
0	CMD_INHIBIT	R	Command Inhibit (CMD) This bit indicates the following: ■ SD/eMMC mode: If this bit is set to 0, it indicates that the CMD line is not in use and the host controller can issue an SD/eMMC command using the CMD line. This bit is set when the command register is written. This bit is cleared when the command response is received. This bit is not cleared by the response of auto CMD12/23 but cleared by the response of read/write command. ■ UHS-II mode: If this bit is set to 0, it indicates that a command packet can be issued by the host controller. Values: ■ 0x0 (READY): Host controller is ready to issue a command. ■ 0x1 (NOT_READY): Host controller is not ready to issue a command. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.13 HOST_CTRL1_R

- Name: Host Control 1 Register
- Description: This register is used to control the operation of the host controller. This register is applicable for an SD/eMMC/UHS-II mode.
- Size: 8 bits

- Offset: 0x28

Figure & Table 3-56 Fields for register: HOST_CTRL1_R

Bits	Name	Access	Description
7	CARD_DETECT_SIG_SEL	R/W	Card Detect Signal Selection This bit selects a source for card detection. When the source for the card detection is switched, the interrupt must be disabled during the switching period. Values: ■ 0x1 (CARD_DT_TEST_LEVEL): Card Detect Test Level is selected. (for test purpose) ■ 0x0 (SDCD_PIN): SDCD# (card_detect_n signal) is selected. (for normal use) Value After Reset: 0x0 Exists: Always
6	CARD_DETECT_TEST_LVL	R/W	Card Detect Test Level This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates whether a card inserted or not. Values: ■ 0x1 (CARD_INSERTED): Card inserted ■ 0x0 (No_CARD): No card Value After Reset: 0x0 Exists: Always
5	EXT_DAT_XFER	R/W	Extended Data Transfer Width This bit controls 8-bit bus width mode of embedded device. This bit is not applicable for UHS-II mode. Values: ■ 0x1 (EIGHT_BIT): 8-bit bus width ■ 0x0 (DEFAULT): Bus width is selected by the data transfer width. Value After Reset: 0x0 Exists: Always
4:3	DMA_SEL	R/W	DMA Select This field is used to select the DMA type. When Host Version 4 Enable is 1 in host control 2 register: ■ 0x0: SDMA is selected. ■ 0x1: Reserved.

Bits	Name	Access	Description
			■ 0x2: ADMA2 is selected. ■ 0x3: ADMA2 or ADMA3 is selected. When Host Version 4 Enable is 0 in Host Control 2 register: ■ 0x0: SDMA is selected. ■ 0x1: Reserved. ■ 0x2: 32-bit Address ADMA2 is selected. ■ 0x3: 64-bit Address ADMA2 is selected. Values: ■ 0x0 (SDMA): SDMA is selected. ■ 0x1 (RSVD_BIT): Reserved. ■ 0x2 (ADMA2): ADMA2 is selected. ■ 0x3 (ADMA2_3): ADMA2 or ADMA3 is selected. Value After Reset: 0x0 Exists: Always
2	HIGH_SPEED_EN	R/W	High Speed Enable (SD/eMMC Mode only) In SD/eMMC mode, this bit is used to determine the selection of preset value for High Speed mode. Before setting this bit, the Host Driver checks the High Speed Support in the Capabilities register. Note: DWC_MSHC always outputs the sd_cmd_out and sd_dat_out lines at the rising edge of cclk_tx clock irrespective of this bit. Please refer to the section "Connecting the Clock IO interface" in the Mobile Storage Host Controller user guide on clocking requirement for an SD/eMMC card. In UHS-II mode, this bit is irrelevant. Values: ■ 0x1 (HIGH_SPEED): High speed mode ■ 0x0 (NORMAL_SPEED): Normal speed mode Value After Reset: 0x0 Exists: Always
1	DAT_XFER_WIDTH	R/W	Data Transfer Width For SD/eMMC mode, this bit selects the data transfer width of the host controller. The host driver sets it to match the data width of the SD/eMMC card. In UHS-II mode, this bit is irrelevant. Values:

Bits	Name	Access	Description
			■ 0x1 (FOUR_BIT): 4-bit mode ■ 0x0 (ONE_BIT): 1-bit mode Value After Reset: 0x0 Exists: Always
0	LED_CTRL	R/W	LED Control This bit is used to caution the user not to remove the card while the SD card is being accessed. The value is reflected on the led_control signal. Values: ■ 0x0 (OFF): LED off. ■ 0x1 (ON): LED on. Value After Reset: 0x0 Exists: Always

3.6.1.14 PWR_CTRL_R

- Name: Power control register
- Description: This register is used to control the bus power for the card. This register is applicable for an SD, eMMC, and UHS-II modes.
- Size: 8 bits
- Offset: 0x29

Figure & Table 3-57 Fields for register: PWR_CTRL_R

Bits	Name	Access	Description
7:5	SD_BUS_VOL_VDD2	R/W	SD Bus Voltage Select for VDD2. This bit determines supply voltage range to VDD2 of UHS-II card. This bit can be set to 0x5 if 1.8V VDD2 Support in the Capabilities register is set to 1. This is irrelevant for SD/eMMC card. Values: ■ 0x7 (NOT_USED7): Not used ■ 0x6 (NOT_USED6): Not used ■ 0x5 (V_1_8): 1.8V ■ 0x4 (V_1_2): Reserved for 1.2V ■ 0x3 (RSVD3): Reserved ■ 0x2 (RSVD2): Reserved ■ 0x1 (RSVD1): Reserved ■ 0x0 (NO_VDD2): VDD2 not supported.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
4	SD_BUS_PWR_VDD2	R/W	SD Bus Power for VDD2 This bit enables VDD2 power of UHS-II card. This setting is available on the sd_vdd2_on output of DWC_mshc so that it can be used to control the VDD2 power supply of the card. This is irrelevant for SD/eMMC card. Values: ■ 0x0 (OFF): Power off. ■ 0x1 (ON): Power on. Value After Reset: 0x0 Exists: Always
3:1	SD_BUS_VOL_VDD1	R/W	SD Bus Voltage Select for VDD1/eMMC Bus Voltage Select for VDD These bits enable the host driver to select the voltage level for an SD/eMMC card. Before setting this register, the host driver checks the Voltage Support bits in the Capabilities register. If an unsupported voltage is selected, the host system does not supply the SD bus voltage. The value set in this field is available on the DWC_mshc output signal (sd_vdd1_sel), which is used by the voltage switching circuitry. SD Bus Voltage Select options: ■ 0x7: 3.3V (Typical) ■ 0x6: 3.0V (Typical) ■ 0x5: 1.8V (Typical) for embedded ■ 0x4: 0x0 - Reserved eMMC Bus Voltage Select options: ■ 0x7: 3.3V (Typical) ■ 0x6: 1.8V (Typical) ■ 0x5: 1.2V (Typical) ■ 0x4: 0x0 - Reserved Values: ■ 0x7 (V_3_3): 3.3V (Typ.) ■ 0x6 (V_3_0): 3.0V (Typ.) ■ 0x5 (V_1_8): 1.8V (Typ.) for embedded ■ 0x4 (RSVD4): Reserved

Bits	Name	Access	Description
			■ 0x3 (RSVD3): Reserved ■ 0x2 (RSVD2): Reserved ■ 0x1 (RSVD1): Reserved ■ 0x0 (RSVD0): Reserved Value After Reset: 0x0 Exists: Always
0	SD_BUS_PWR_VDD1	R/W	SD Bus Power for VDD1 This bit enables VDD1 power of the card. This setting is available on the sd_vdd1_on output of DWC_mshc so that it can be used to control the VDD1 power supply of the card. Before setting this bit, the SD host driver sets the SD Bus Voltage Select bit. If the host controller detects a No Card state, this bit is cleared. In SD mode, if this bit is cleared, the host controller stops the SD clock by clearing the SD_CLK_IN bit in the CLK_CTRL_R register. In UHS-II mode, before clearing this bit, the host driver clears the SD_CLK_IN bit in the CLK_CTRL_R register. Values: ■ 0x0 (OFF): Power off. ■ 0x1 (ON): Power on. Value After Reset: 0x0 Exists: Always

3.6.1.15 BGAP_CTRL_R

- Name: Block Gap Control Register
- Description: This register is used by the host driver to control any operation related to Block Gap. This register is applicable for an SD/eMMC/UHS-II mode.
- Size: 8 bits
- Offset: 0x2a

Figure & Table 3-58 Fields for register: BGAP_CTRL_R

Bits	Name	Access	Description
7:4	RSVD_7_4	R	These bits of the Block Gap Control register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
3	INT_AT_BGAP	R/W	Interrupt At Block Gap

Bits	Name	Access	Description
			This bit is valid only in the 4-bit mode of an SDIO card and is used to select a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. In UHS-II mode, this bit is disabled. Values: ■ 0x0 (DISABLE): Disabled ■ 0x1 (ENABLE): Enabled Value After Reset: 0x0 Exists: Always
2	RD_WAIT_CTRL	R/W	Read Wait Control This bit is used to enable the read wait protocol to stop read data using DAT[2] line if the card supports read wait. Otherwise, the Host Controller has to stop the card clock to hold the read data. In UHS-II mode, Read Wait is disabled. Values: ■ 0x0 (DISABLE): Disable Read Wait Control. ■ 0x1 (ENABLE): Enable Read Wait Control. Value After Reset: 0x0 Exists: Always
1	CONTINUE_REQ	R/W	Continue Request This bit is used to restart the transaction, which was stopped using the Stop At Block Gap Request. The Host Controller automatically clears this bit when the transaction restarts. If stop at block gap request is set to 1, any write to this bit is ignored. Values: ■ 0x0 (NO_AFFECT): No effect ■ 0x1 (RESTART): Restart Value After Reset: 0x0 Exists: Always Volatile: True
0	STOP_BG_REQ	R/W	Stop At Block Gap Request This bit is used to stop executing read and write transactions at the next block gap for non-DMA, SDMA, and ADMA transfers. Values:

Bits	Name	Access	Description
			■ 0x0 (XFER): Transfer ■ 0x1 (STOP): Stop Value After Reset: 0x0 Exists: Always

3.6.1.16 WUP_CTRL_R

- Name: Wakeup Control Register
- Description: This register is mandatory for the Host Controller, but the wakeup functionality depends on the Host Controller system hardware and software. The Host Driver maintains voltage on the SD Bus by setting the SD Bus Power to 1 in the Power Control Register, while a wakeup event through the Card Interrupt is desired.
- Size: 8 bits
- Offset: 0x2b

Figure & Table 3-59 Fields for register: WUP_CTRL_R

Bits	Name	Access	Description
7:3	RSVD_7_3	R	These bits of Wakeup Control register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
2	CARD_REMOVAL	R/W	Wakeup Event Enable on SD Card Removal This bit enables wakeup event through Card Removal assertion in the Normal Interrupt Status register. For the SDIO card, Wake Up Support (FN_WUS) in the Card Information Structure (CIS) register does not affect this bit. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable Value After Reset: 0x0 Exists: Always
1	CARD_INSERT	R/W	Wakeup Event Enable on SD Card Insertion This bit enables wakeup event through Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit. Values: ■ 0x0 (DISABLED): Disable

Bits	Name	Access	Description
			■ 0x1 (ENABLED): Enable Value After Reset: 0x0 Exists: Always
0	CARD_INT	R/W	Wakeup Event Enable on Card Interrupt This bit enables wakeup event through a Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable Value After Reset: 0x0 Exists: Always

3.6.1.17 CLK_CTRL_R

- Name: Clock Control Register
- Description: This register controls SDCLK (card clock) in an SD/eMMC mode and RCLK in UHS-II mode. This register is applicable for an SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x2c

Figure & Table 3-60 Fields for register: CLK_CTRL_R

Bits	Name	Access	Description
15:8	FREQ_SEL	R/W	SDCLK/RCLK Frequency Select These bits are used to select the frequency of the SDCLK signal. These bits depend on setting of Preset Value Enable in the Host Control 2 register. If Preset Value Enable = 0, these bits are set by the host driver. If Preset Value Enable = 1, these bits are automatically set to a value specified in one of the Preset Value register. The value is reflected on the lower 8-bit of the card_clk_freq_sel signal. 10-bit Divided Clock Mode ■ 0x3FF: 1/2046 divided clock ■ ..... ■ N: 1/2N divided clock ■ ..... ■ 0x002: 1/4 divided clock

Bits	Name	Access	Description
			■ 0x001: 1/2 divided clock ■ 0x000: Base clock (10MHz - 255MHz) Programmable Clock Mode: Enables the host system to select a fine grain SD clock frequency: ■ 0x3FF: Base clock * M /1024 ■ ..... ■ N-1: Base clock * M /N ■ ..... ■ 0x002: Base clock * M /3 ■ 0x001: Base clock * M /2 ■ 0x000: Base clock * M Value After Reset: 0x0 Exists: Always Volatile: True
7:6	UPPER_FREQ_SEL	R/W	These bits specify the upper 2 bits of 10-bit SDCLK/RCLK Frequency Select control. The value is reflected on the upper 2 bits of the card_clk_freq_sel signal. Value After Reset: 0x0 Exists: Always Volatile: True
5	CLK_GEN_SELECT	R/W	Clock Generator Select This bit is used to select the clock generator mode in SDCLK/RCLK Frequency Select. If Preset Value Enable = 0, this bit is set by the host driver. If Preset Value Enable = 1, this bit is automatically set to a value specified in one of the Preset Value registers. The value is reflected on the card_clk_gen_sel signal. Values: ■ 0x0 (FALSE): Divided clock mode ■ 0x1 (TRUE): Programmable clock mode Value After Reset: 0x0 Exists: Always Volatile: True
4	RSVD_4	R	This bit of the CLK_CTRL_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
3	PLL_ENABLE	R/W	PLL Enable This bit is used to activate the PLL (applicable when Host Version 4 Enable = 1). When Host Version 4 Enable = 0, INTERNAL_CLK_EN bit may be used to activate PLL. The value is reflected on the card_clk_en signal. Note: If this bit is not used to to activate the PLL when Host Version 4 Enable = 1, it is recommended to set this bit to 1. Values: ■ 0x0 (FALSE): PLL is in low power mode. ■ 0x1 (TRUE): PLL is enabled. Value After Reset: 0x0 Exists: Always
2	SD_CLK_EN	R/W	SD/eMMC Clock Enable This bit stops the SDCLK or RCLK when set to 0. The SDCLK/RCLK Frequency Select bit can be changed when this bit is set to 0. The value is reflected on the clk2card_on pin. Values: ■ 0x0 (FALSE): Disable providing SDCLK/RCLK. ■ 0x1 (TRUE): Enable providing SDCLK/RCLK. Value After Reset: 0x0 Exists: Always
1	INTERNAL_CLK_STABLE	R	Internal Clock Stable This bit enables the host driver to check the clock stability twice after the Internal Clock Enable bit is set and after the PLL Enable bit is set. This bit reflects the synchronized value of the intclk_stable signal after the Internal Clock Enable bit is set to 1 and also reflects the synchronized value of the card_clk_stable signal after the PLL Enable bit is set to 1. Values: ■ 0x0 (FALSE): Not ready ■ 0x1 (TRUE): Ready Value After Reset: 0x0 Exists: Always Volatile: True
0	INTERNAL_CLK_EN	R/W	Internal Clock Enable

Bits	Name	Access	Description
			This bit is set to 0 when the host driver is not using the host controller or the host controller awaits a wakeup interrupt. The host controller must stop its internal clock to enter a very low power state. However, registers can still be read and written to. The value is reflected on the intclk_en signal. Note: If this bit is not used to control the internal clock (base clock and master clock), it is recommended to set this bit to 1. Values: ■ 0x0 (FALSE): Stop ■ 0x1 (TRUE): Oscillate Value After Reset: 0x0 Exists: Always

3.6.1.18 TOUT_CTRL_R

- Name: Timeout Control Register
- Description: This register is used to set the Data Timeout Counter value for an SD/eMMC mode according to the timer clock defined by the Capabilities register, while initializing the host controller.
- Size: 8 bits
- Offset: 0x2e

Figure & Table 3-61 Fields for register: TOUT_CTRL_R

Bits	Name	Access	Description
7:4	RSVD_7_4	R	These bits of the Timeout Control register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
3:0	TOUT_CNT	R/W	Data Timeout Counter Value This value determines the interval by which DAT line timeouts are detected. The timeout clock frequency is generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt Status Enable register). The values for these bits are: ■ 0xF : Reserved ■ 0xE : TMCLK x 2^27

Bits	Name	Access	Description
			■ ..... ■ 0x1 : TMCLK x 2^14 ■ 0x0 : TMCLK x 2^13 Note: During a boot operating in eMMC mode, an application must configure the boot data timeout value (approximately 1 second) in this bit. Value After Reset: 0x0 Exists: Always

3.6.1.19 SW_RST_R

- Name: Software Reset Register
- Description: This register is used to generate a reset pulse by writing 1 to each bit of this register. After completing the reset, the host controller clears each bit. As it takes some time to complete a software reset, the host driver confirms that these bits are 0. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 8 bits
- Offset: 0x2f

Figure & Table 3-62 Fields for register: SW_RST_R

Bits	Name	Access	Description
7:3	RSVD_7_3	R	These bits of the SW_RST_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
2	SW_RST_DAT	R/W	Software Reset For DAT line This bit is used in SD/eMMC mode and it resets only a part of the data circuit and the DMA circuit is also reset. The following registers and bits are cleared by this bit: ■ Buffer Data Port register – Buffer is cleared and initialized. ■ Present State register – Buffer Read Enable – Buffer Write Enable – Read Transfer Active – Write Transfer Active – DAT Line Active – Command Inhibit (DAT)

Bits	Name	Access	Description
			■ Block Gap Control register – Continue Request – Stop At Block Gap Request ■ Normal Interrupt Status register – Buffer Read Ready – Buffer Write Ready – DMA Interrupt – Block Gap Event – Transfer Complete In UHS-II mode, this bit shall be set to 0. Values: ■ 0x0 (FALSE): Work ■ 0x1 (TRUE): Reset Value After Reset: 0x0 Exists: Always Volatile: True
1	SW_RST_CMD	R/W	Software Reset For CMD line This bit resets only a part of the command circuit is able to issue a command. This bit is also used to initialize a UHS-II command circuit. This reset is effective only for a command issuing circuit (including response error statuses related to Command Inhibit (CMD) control) and does not affect the data transfer circuit. Host controller can continue data transfer even after this reset is executed while handling subcommand-response errors. The following registers and bits are cleared by this bit: ■ Present State register: Command Inhibit (CMD) bit ■ Normal Interrupt Status register: Command Complete bit ■ Error Interrupt Status: Response error statuses related to Command Inhibit (CMD) bit Values: ■ 0x0 (FALSE): Work ■ 0x1 (TRUE): Reset Value After Reset: 0x0 Exists: Always Volatile: True

Bits	Name	Access	Description
0	SW_RST_ALL	R/W	Software Reset for All This reset affects the entire host controller except for the card detection circuit. During its initialization, the host driver sets this bit to 1 to reset the host controller. All registers are reset except the capabilities register. If this bit is set to 1, the host driver must issue reset command and reinitialize the card. Values: ■ 0x0 (FALSE): Work ■ 0x1 (TRUE): Reset Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.20 NORMAL_INT_STAT_R

- Name: Normal Interrupt Status Register
- Description: This register reflects the status of the normal interrupt. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x30

Figure & Table 3-63 Fields for register: NORMAL_INT_STAT_R

Bits	Name	Access	Description
15	ERR_INTERRUPT	R	Error Interrupt If any of the bits in the Error Interrupt Status register are set, then this bit is set. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
14	CQE_EVENT	R/W1C	Command Queuing Event This status is set if Command Queuing/Crypto related event has occurred in eMMC/SD mode. Read CQHCI's CQIS/CRNQIS register for more details. In UHS-II Mode, this bit is irrelevant. Values:

Bits	Name	Access	Description
			■ 0x0 (FALSE): No event ■ 0x1 (TRUE): Command Queuing Event is detected. Value After Reset: 0x0 Exists: Always Volatile: True
13	FX_EVENT	R	FX Event This status is set when R [14] of response register is set to 1 and Response Type R1/R5 is set to 0 in Transfer Mode register. This interrupt is used with response check function. Values: ■ 0x0 (FALSE): No event ■ 0x1 (TRUE): FX Event is detected. Value After Reset: 0x0 Exists: Always Volatile: True
12	RE_TUNE_EVENT	R	Re-tuning Event This bit is set if the Re-Tuning Request changes from 0 to 1. Re-Tuning request is not supported. Value After Reset: 0x0 Exists: Always Volatile: True
11	INT_C	R	INT_C (Embedded) This bit is set if INT_C is enabled and if INT_C# pin is in low level. The INT_C# pin is not supported. Value After Reset: 0x0 Exists: Always Volatile: True
10	INT_B	R	INT_B (Embedded) This bit is set if INT_B is enabled and if INT_B# pin is in low level. The INT_B# pin is not supported. Value After Reset: 0x0 Exists: Always Volatile: True
9	INT_A	R	INT_A (embedded) This bit is set if INT_A is enabled and if INT_A# pin is in

Bits	Name	Access	Description
			low level. The INT_A# pin is not supported. Value After Reset: 0x0 Exists: Always Volatile: True
8	CARD_INTERRUPT	R	Card Interrupt This bit reflects the synchronized value of: ■ DAT [1] interrupt input for SD mode ■ DAT [2] interrupt input for UHS-II mode Values: ■ 0x0 (FALSE): No card interrupt ■ 0x1 (TRUE): Generate card interrupt. Value After Reset: 0x0 Exists: Always Volatile: True
7	CARD_REMOVAL	R/W1C	Card Removal This bit is set if the Card Inserted in the Present State register changes from 1 to 0. Values: ■ 0x0 (FALSE): Card state stable or Debouncing ■ 0x1 (TRUE): Card removed Value After Reset: 0x0 Exists: Always Volatile: True
6	CARD_INSERTION	R/W1C	Card Insertion This bit is set if the Card Inserted in the Present State register changes from 0 to 1. Values: ■ 0x0 (FALSE): Card state stable or Debouncing ■ 0x1 (TRUE): Card inserted Value After Reset: 0x0 Exists: Always Volatile: True
5	BUF_RD_READY	R/W1C	Buffer Read Ready This bit is set if the Buffer Read Enable changes from 0 to 1.

Bits	Name	Access	Description
			Values: ■ 0x0 (FALSE): Not ready to read buffer. ■ 0x1 (TRUE): Ready to read buffer. Value After Reset: 0x0 Exists: Always Volatile: True
4	BUF_WR_READY	R/W1C	Buffer Write Ready This bit is set if the Buffer Write Enable changes from 0 to 1. Values: ■ 0x0 (FALSE): Not ready to write buffer. ■ 0x1 (TRUE): Ready to write buffer. Value After Reset: 0x0 Exists: Always Volatile: True
3	DMA_INTERRUPT	R/W1C	DMA Interrupt This bit is set if the host controller detects the SDMA Buffer Boundary during transfer. In case of ADMA, by setting the Int field in the descriptor table, the host controller generates this interrupt. This interrupt is not generated after a transfer Complete. Values: ■ 0x0 (FALSE): No DMA interrupt ■ 0x1 (TRUE): DMA interrupt is generated. Value After Reset: 0x0 Exists: Always Volatile: True
2	BGAP_EVENT	R/W1C	Block Gap Event This bit is set when both read/write transaction is stopped at block gap due to a Stop at Block Gap Request. Values: ■ 0x0 (FALSE): No Block Gap Event ■ 0x1 (TRUE): Transaction stopped at block gap. Value After Reset: 0x0 Exists: Always Volatile: True

Bits	Name	Access	Description
1	XFER_COMPLETE	R/W1C	Transfer Complete This bit is set when a read/write transfer and a command with status busy is completed. Values: ■ 0x0 (FALSE): Not complete ■ 0x1 (TRUE): Command execution is completed. Value After Reset: 0x0 Exists: Always Volatile: True
0	CMD_COMPLETE	R/W1C	Command Complete In SD/eMMC Mode, this bit is set when the end bit of a response except for Auto CMD12 and Auto CMD23. In UHSII Mode, this bit is set when response packet is received. This interrupt is not generated when the Response Interrupt Disable in Transfer Mode Register is set to 1. Values: ■ 0x0 (FALSE): No command complete ■ 0x1 (TRUE): Command complete. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.21 ERROR_INT_STAT_R

- Name: Error Interrupt Status Register
- Description: This register enables an interrupt when the Error Interrupt Status Enable is enabled and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 retains the bit unchanged. Signals defined in this register can be enabled by the Error Interrupt Status Enable register, but not by the Error Interrupt Signal Enable register. More than one status can be cleared with a single register write. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x32

Figure & Table 3-64 Fields for register: ERROR_INT_STAT_R

Bits	Name	Access	Description
15	VENDOR_ERR3	R/W1C	This bit (VENDOR_ERR3) of the ERROR_INT_STAT_R register is reserved. It always returns 0.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always Volatile: True
14	VENDOR_ERR2	R/W1C	This bit (VENDOR_ERR2) of the ERROR_INT_STAT_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always Volatile: True
13	VENDOR_ERR1	R/W1C	This bit (VENDOR_ERR1) of the ERROR_INT_STAT_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always Volatile: True
12	BOOT_ACK_ERR	R/W1C	Boot Acknowledgement Error This bit is set when there is a timeout for boot acknowledgement or when detecting boot ack status having a value other than 010. This is applicable only when boot acknowledgement is expected in eMMC mode. In SD/UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
11	RESP_ERR	R/W1C	Response Error Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by host driver during DMA execution. If Response Error Check Enable is set to 1 in the Transfer Mode register, host controller checks R1 or R5 response. If an error is detected in a response, this bit is set to 1. This is applicable in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error

Bits	Name	Access	Description
			■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
10	TUNING_ERR	R/W1C	Tuning Error This bit is set when an unrecoverable error is detected in a tuning circuit except during the tuning procedure (occurrence of an error during tuning procedure is indicated by Sampling Clock Select in the Host Control 2 register). By detecting Tuning Error, host driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock Select is set to 0 before executing tuning procedure. The Tuning Error has higher priority than the other error interrupts generated during data transfer. By detecting Tuning Error, the host driver must discard data transferred by a current read/write command and retry data transfer after the host controller retrieved from the tuning circuit error. This is applicable in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
9	ADMA_ERR	R/W1C	ADMA Error This bit is set when the host controller detects error during ADMA-based data transfer. The error could be due to the following reasons: ■ Error response received from system bus (master I/F) ■ ADMA3, ADMA2 descriptors invalid ■ CQE task or transfer descriptors invalid When the error occurs, the state of the ADMA is saved in the ADMA Error Status register. In eMMC CQE mode: The host controller generates this interrupt when it detects an invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status

Bits	Name	Access	Description
			indicates that an error has occurred in ST_FDS state. The host driver may find that Valid bit is not set at the error descriptor. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
8	AUTO_CMD_ERR	R/W1C	Auto CMD Error This error status is used by Auto CMD12 and Auto CMD23 in SD/eMMC mode. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared by clearing of this bit. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
7	CUR_LMT_ERR	R/W1C	Current Limit Error By setting the SD Bus Power bit in the Power Control register, the host controller is requested to supply power for the SD bus. If the host controller 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. A reading of 1 for this bit means that the host controller is not supplying power to the SD card due to some failure. A reading of 0 for this bit means that the host controller is supplying power and no error has occurred. The host controller may require some sampling time to detect the current limit. DWC_mshc host controller does not support this function, this bit is always set to 0. Values: ■ 0x0 (FALSE): No error

Bits	Name	Access	Description
			■ 0x1 (TRUE): Power fail. Value After Reset: 0x0 Exists: Always Volatile: True
6	DATA_END_BIT_ERR	R/W1C	Data End Bit Error This error occurs in SD/eMMC mode either when detecting 0 at the end bit position of read data that uses the DAT line or at the end bit position of the CRC status. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
5	DATA_CRC_ERR	R/W1C	Data CRC Error This error occurs in SD/eMMC mode when detecting CRC error when transferring read data which uses the DAT line, when detecting the Write CRC status having a value other than 010 or when writing CRC status timeout. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
4	DATA_TOUT_ERR	R/W1C	Data Timeout Error This bit is set in SD/eMMC mode when detecting one of the following timeout conditions: ■ Busy timeout for R1b, R5b type ■ Busy timeout after Write CRC status ■ Write CRC status timeout ■ Read data timeout In UHS-II mode, this bit is irrelevant.

Bits	Name	Access	Description
			Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Time out Value After Reset: 0x0 Exists: Always Volatile: True
3	CMD_IDX_ERR	R/W1C	Command Index Error This bit is set if a Command Index error occurs in the command response in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always Volatile: True
2	CMD_END_BIT_ERR	R/W1C	Command End Bit Error This bit is set when detecting that the end bit of a command response is 0 in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): End bit error generated. Value After Reset: 0x0 Exists: Always Volatile: True
1	CMD_CRC_ERR	R/W1C	Command CRC Error Command CRC Error is generated in SD/eMMC mode for following two cases. ■ If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response. ■ The host controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the host controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next

Bits	Name	Access	Description
			SD clock edge, then the host controller aborts the command (stop driving CMD line) and set this bit to 1. The Command Timeout Error is also set to 1 to distinguish a CMD line conflict. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): CRC error generated. Value After Reset: 0x0 Exists: Always Volatile: True
0	CMD_TOUT_ERR	R/W1C	Command Timeout Error In SD/eMMC Mode, this bit is set only if no response is returned within 64 SD clock cycles from the end bit of the command. If the host controller detects a CMD line conflict, along with Command CRC Error bit, this bit is set to 1, without waiting for 64 SD/eMMC card clock cycles. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Time out Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.22 NORMAL_INT_STAT_EN_R

- Name: Normal Interrupt Status Enable Register
- Description: This register enables the interrupt status for Normal Interrupt Status register (NORMAL_INT_STAT_R) when NORMAL_INT_STAT_R is set to 1. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x34

Figure & Table 3-65 Fields for register: NORMAL_INT_STAT_EN_R

Bits	Name	Access	Description
15	RSVD_15	R	This bit of the NORMAL_INT_STAT_EN_R register is reserved. It always returns 0.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
14	CQE_EVENT_STAT_EN	R/W	CQE Event Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
13	FX_EVENT_STAT_EN	R/W	FX Event Status Enable This bit is added from Version 4.10. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
12	RE_TUNE_EVENT_STAT_EN	R/W	Re-Tuning Event (UHS-I only) Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
11	INT_C_STAT_EN	R/W	INT_C (Embedded) Status Enable If this bit is set to 0, the host controller clears the interrupt request to the system. The host driver may clear this bit before servicing the INT_C and may set this bit again after all interrupt requests to INT_C pin are cleared to prevent inadvertent interrupts. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
10	INT_B_STAT_EN	R/W	INT_B (Embedded) Status Enable If this bit is set to 0, the host controller clears the interrupt request to the system. The host driver may clear this bit before servicing the INT_B and may set this

Bits	Name	Access	Description
			bit again after all interrupt requests to INT_B pin are cleared to prevent inadvertent interrupts. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
9	INT_A_STAT_EN	R/W	INT_A (Embedded) Status Enable If this bit is set to 0, the host controller clears the interrupt request to the system. The host driver may clear this bit before servicing the INT_A and may set this bit again after all interrupt requests to INT_A pin are cleared to prevent inadvertent interrupts. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
8	CARD_INTERRUPT_STAT_EN	R/W	Card Interrupt Status Enable If this bit is set to 0, the host controller clears the interrupt request to the system. The card interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The host driver may clear the Card Interrupt Status Enable before servicing the card interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. By setting this bit to 0, interrupt input must be masked by implementation so that the interrupt input is not affected by external signal in any state (for example, floating). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
7	CARD_REMOVAL_STAT_EN	R/W	Card Removal Status Enable Values:

Bits	Name	Access	Description
			■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
6	CARD_INSERTION_STAT_EN	R/W	Card Insertion Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
5	BUF_RD_READY_STAT_EN	R/W	Buffer Read Ready Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
4	BUF_WR_READY_STAT_EN	R/W	Buffer Write Ready Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
3	DMA_INTERRUPT_STAT_EN	R/W	DMA Interrupt Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
2	BGAP_EVENT_STAT_EN	R/W	Block Gap Event Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
1	XFER_COMPLETE_STAT_EN	R/W	Transfer Complete Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
0	CMD_COMPLETE_STAT_EN	R/W	Command Complete Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

3.6.1.23 ERROR_INT_STAT_EN_R

- Name: Error Interrupt Status Enable Register
- Description: This register sets the interrupt status for Error Interrupt Status register (ERROR_INT_STAT_R), when ERROR_INT_STAT_EN_R is set to 1. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x36

Figure & Table 3-66 Fields for register: ERROR_INT_STAT_EN_R

Bits	Name	Access	Description
15	VENDOR_ERR_STAT_EN3	R/W	The 15th bit of Error Interrupt Status Enable register is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
14	VENDOR_ERR_STAT_EN2	R/W	The 14th bit of Error Interrupt Status Enable register is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
13	VENDOR_ERR_STAT_EN1	R/W	The 13th bit of Error Interrupt Status Enable register is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
12	BOOT_ACK_ERR_STAT_EN	R/W	Boot Acknowledgment Error (eMMC mode only) Setting this bit to 1 enables setting of Boot Acknowledgment Error in Error Interrupt Status register (ERROR_INT_STAT_R). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
11	RESP_ERR_STAT_EN	R/W	Response Error Status Enable (SD mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
10	TUNING_ERR_STAT_EN	R/W	Tuning Error Status Enable (UHS-I mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
9	ADMA_ERR_STAT_EN	R/W	ADMA Error Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
8	AUTO_CMD_ERR_STAT_EN	R/W	Auto CMD Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
7	CUR_LMT_ERR_STAT_EN	R/W	Current Limit Error Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
6	DATA_END_BIT_ERR_STAT_EN	R/W	Data End Bit Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
5	DATA_CRC_ERR_STAT_EN	R/W	Data CRC Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
4	DATA_TOUT_ERR_STAT_EN	R/W	Data Timeout Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
3	CMD_IDX_ERR_STAT_EN	R/W	Command Index Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked

Bits	Name	Access	Description
			■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
2	CMD_END_BIT_ERR_STAT_EN	R/W	Command End Bit Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
1	CMD_CRC_ERR_STAT_EN	R/W	Command CRC Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
0	CMD_TOUT_ERR_STAT_EN	R/W	Command Timeout Error Status Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

3.6.1.24 NORMAL_INT_SIGNAL_EN_R

- Name: Normal Interrupt Signal Enable Register
- Description: This register is used to select the interrupt status that is indicated to the host system as the interrupt. All these status bits share the same 1-bit interrupt line. Setting any of these bits to 1, enables interrupt generation. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x38

Figure & Table 3-67 Fields for register: NORMAL_INT_SIGNAL_EN_R

Bits	Name	Access	Description
15	RSVD_15	R	This bit of the NORMAL_INT_STAT_R register is reserved.

Bits	Name	Access	Description
			It always returns 0. Value After Reset: 0x0 Exists: Always
14	CQE_EVENT_SIGNAL_EN	R/W	Command Queuing Engine Event Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
13	FX_EVENT_SIGNAL_EN	R/W	FX Event Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
12	RE_TUNE_EVENT_SIGNAL_EN	R/W	Re-Tuning Event (UHS-I only) Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
11	INT_C_SIGNAL_EN	R/W	INT_C (Embedded) Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
10	INT_B_SIGNAL_EN	R/W	INT_B (Embedded) Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
9	INT_A_SIGNAL_EN	R/W	INT_A (Embedded) Signal Enable

Bits	Name	Access	Description
			Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
8	CARD_INTERRUPT_SIGNAL_EN	R/W	Card Interrupt Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
7	CARD_REMOVAL_SIGNAL_EN	R/W	Card Removal Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
6	CARD_INSERTION_SIGNAL_EN	R/W	Card Insertion Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
5	BUF_RD_READY_SIGNAL_EN	R/W	Buffer Read Ready Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
4	BUF_WR_READY_SIGNAL_EN	R/W	Buffer Write Ready Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
3	DMA_INTERRUPT_SIGNAL_EN	R/W	DMA Interrupt Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
2	BGAP_EVENT_SIGNAL_EN	R/W	Block Gap Event Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
1	XFER_COMPLETE_SIGNAL_EN	R/W	Transfer Complete Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
0	CMD_COMPLETE_SIGNAL_EN	R/W	Command Complete Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

3.6.1.25 ERROR_INT_SIGNAL_EN_R

- Name: Error Interrupt Signal Enable Register
- Description: This register is used to select the interrupt status that is notified to the host system as an interrupt. All these status bits share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x3a

Figure & Table 3-68 Fields for register: ERROR_INT_SIGNAL_EN_R

Bits	Name	Access	Description
15	VENDOR_ERR_SIGNAL_EN3	R/W	The 16th bit of Error Interrupt Signal Enable is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
14	VENDOR_ERR_SIGNAL_EN2	R/W	The 15th bit of Error Interrupt Signal Enable is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
13	VENDOR_ERR_SIGNAL_EN1	R/W	The 14th bit of Error Interrupt Signal Enable is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
12	BOOT_ACK_ERR_SIGNAL_EN	R/W	Boot Acknowledgment Error (eMMC mode only) Setting this bit to 1 enables generating interrupt signal when Boot Acknowledgement Error in Error Interrupt Status register is set. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
11	RESP_ERR_SIGNAL_EN	R/W	Response Error Signal Enable (SD mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
10	TUNING_ERR_SIGNAL_EN	R/W	Tuning Error Signal Enable (UHS-I mode only)

Bits	Name	Access	Description
			Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
9	ADMA_ERR_SIGNAL_EN	R/W	ADMA Error Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
8	AUTO_CMD_ERR_SIGNAL_EN	R/W	Auto CMD Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
7	CUR_LMT_ERR_SIGNAL_EN	R/W	Current Limit Error Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
6	DATA_END_BIT_ERR_SIGNAL_EN	R/W	Data End Bit Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
5	DATA_CRC_ERR_SIGNAL_EN	R/W	Data CRC Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
4	DATA_TOUT_ERR_SIGNAL_EN	R/W	Data Timeout Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
3	CMD_IDX_ERR_SIGNAL_EN	R/W	Command Index Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error Value After Reset: 0x0 Exists: Always
2	CMD_END_BIT_ERR_SIGNAL_EN	R/W	Command End Bit Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
1	CMD_CRC_ERR_SIGNAL_EN	R/W	Command CRC Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
0	CMD_TOUT_ERR_SIGNAL_EN	R/W	Command Timeout Error Signal Enable (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always

3.6.1.26 AUTO_CMD_STAT_R

- Name: Auto CMD Status Register
- Description: This register is used to indicate the CMD12 response error of Auto CMD12, and the CMD23 response error of Auto CMD23. The host driver can determine the kind of Auto CMD12/CMD23 errors that can occur in this register. Auto CMD23 errors are indicated in bit 04-01. This register is valid only when Auto CMD Error is set. This register is applicable for SD/eMMC mode.
- Size: 16 bits
- Offset: 0x3c

Figure & Table 3-69 Fields for register: AUTO_CMD_STAT_R

Bits	Name	Access	Description
15:8	RSVD_15_8	R	These bits of the AUTO_CMD_STAT_R register are reserved bits. They always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
7	CMD_NOT_ISSUED_AUTO_CMD12	R	Command Not Issued By Auto CMD12 Error If this bit is set to 1, 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. Values: ■ 0x1 (TRUE): Not issued ■ 0x0 (FALSE): No Error Value After Reset: 0x0 Exists: Always Volatile: True
6	RSVD_6	R	This bit of the AUTO_CMD_STAR_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always Volatile: True
5	AUTO_CMD_RESP_ERR	R	Auto CMD Response Error This bit is set when Response Error Check Enable in the Transfer Mode register is set to 1 and an error is detected in R1 response of either Auto CMD12 or CMD13. This status is ignored if any bit between D00 to

Bits	Name	Access	Description
			D04 is set to 1. Values: ■ 0x1 (TRUE): Error ■ 0x0 (FALSE): No Error Value After Reset: 0x0 Exists: Always Volatile: True
4	AUTO_CMD_IDX_ERR	R	Auto CMD Index Error This bit is set if the command index error occurs in response to a command. Values: ■ 0x1 (TRUE): Error ■ 0x0 (FALSE): No Error Value After Reset: 0x0 Exists: Always Volatile: True
3	AUTO_CMD_EBIT_ERR	R	Auto CMD End Bit Error This bit is set when detecting that the end bit of command response is 0. Values: ■ 0x1 (TRUE): End Bit Error generated. ■ 0x0 (FALSE): No error Value After Reset: 0x0 Exists: Always Volatile: True
2	AUTO_CMD_CRC_ERR	R	Auto CMD CRC Error This bit is set when detecting a CRC error in the command response. Values: ■ 0x1 (TRUE): CRC Error generated. ■ 0x0 (FALSE): No error Value After Reset: 0x0 Exists: Always Volatile: True
1	AUTO_CMD_TOUT_ERR	R	Auto CMD Timeout Error This bit is set if no response is returned with 64 SDCLK

Bits	Name	Access	Description
			cycles from the end bit of the command. If this bit is set to 1, error status bits (D04-D01) are meaningless. Values: ■ 0x1 (TRUE): Time out ■ 0x0 (FALSE): No error Value After Reset: 0x0 Exists: Always Volatile: True
0	AUTO_CMD12_NOT_EXEC	R	Auto CMD12 Not Executed If multiple memory block data transfer is not started due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12. Setting this bit to 1 means that the host controller cannot issue Auto CMD12 to stop multiple memory block data transfer, due to some error. If this bit is set to 1, error status bits (D04-D01) are meaningless. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. Values: ■ 0x1 (TRUE): Not executed ■ 0x0 (FALSE): Executed Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.27 HOST_CTRL2_R

- Name: Host Control 2 Register
- Description: This register is used to control how the host controller operates. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x3e

Figure & Table 3-70 Fields for register: HOST_CTRL2_R

Bits	Name	Access	Description
15	PRESET_VAL_ENABLE	R/W	Preset Value Enable This bit enables automatic selection of SDCLK frequency and Driver strength Preset Value registers. When Preset Value Enable is set, SDCLK frequency generation (Frequency Select and Clock Generator Select) and the driver strength selection are performed by the controller. These values are selected from set of Preset Value registers based on selected speed mode. Note: For more information, see the FAQ on Preset Register in the DWC_mshc Databook. Values: ■ 0x0 (FALSE): SDCLK and Driver Strength are controlled by host driver. ■ 0x1 (TRUE): Automatic Selection by Preset Value is enabled. Value After Reset: 0x0 Exists: Always
14	ASYNC_INT_ENABLE	R/W	Asynchronous Interrupt Enable This bit can be set if a card supports asynchronous interrupts and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Values: ■ 0x0 (FALSE): Disabled ■ 0x1 (TRUE): Enabled Value After Reset: 0x0 Exists: Always
13	ADDRESSING	R/W	64-bit Addressing This bit is effective when Host Version 4 Enable is set to 1. Values: ■ 0x0 (FALSE): 32 bits addressing ■ 0x1 (TRUE): 64 bits addressing Value After Reset: 0x0 Exists: Always
12	HOST_VER4_ENABLE	R/W	Host Version 4 Enable This bit selects either Version 3.00 compatible mode or Version 4 mode.

Bits	Name	Access	Description
			Functions of following fields are modified for Host Version 4 mode: ■ SDMA Address: SDMA uses ADMA System Address (05Fh-058h) instead of SDMA System Address register (003h-000h). ■ ADMA2/ADMA3 selection: ADMA3 is selected by DMA select in Host Control 1 register. ■ 64-bit ADMA Descriptor Size: 128-bit descriptor is used instead of 96-bit descriptor when 64-bit Addressing is set to 1. ■ Selection of 32-bit/64-bit System Addressing: Either 32-bit or 64-bit system addressing is selected by 64-bit Addressing bit in this register. ■ 32-bit Block Count: SDMA System Address register (003h-000h) is modified to 32-bit Block Count register. Note: It is recommended not to program ADMA3 Integrated Descriptor Address registers, UHS-II registers and Command Queuing registers (if applicable) while operating in Host version less than 4 mode (Host Version 4 Enable = 0). Values: ■ 0x0 (FALSE): Version 3.00 compatible mode ■ 0x1 (TRUE): Version 4 mode Value After Reset: 0x0 Exists: Always
11	CMD23_ENABLE	R/W	CMD23 Enable If the card supports CMD23, this bit is set to 1. This bit is used to select Auto CMD23 or Auto CMD12 for ADMA3 data transfer. Values: ■ 0x0 (FALSE): Auto CMD23 is disabled. ■ 0x1 (TRUE): Auto CMD23 is enabled. Value After Reset: 0x0 Exists: Always
10	ADMA2_LEN_MODE	R/W	ADMA2 Length Mode This bit selects ADMA2 Length mode to be either 16-bit or 26-bit. Values:

Bits	Name	Access	Description
			■ 0x0 (FALSE): 16-bit data length mode ■ 0x1 (TRUE): 26-bit data length mode Value After Reset: 0x0 Exists: Always
9	RSVD_9	R	This bit of the HOST_CTRL2_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
8	UHS2_IF_ENABLE	R/W	UHS-II Interface Enable This bit is used to enable the UHS-II Interface. The value is reflected on the uhs2_if_en pin. Values: ■ 0x0 (FALSE): SD/eMMC interface disabled. ■ 0x1 (TRUE): UHS-II interface enabled. Value After Reset: 0x0 Exists: Always
7	SAMPLE_CLK_SEL	R/W	Sampling Clock Select This bit is used by the host controller to select the sampling clock in SD/eMMC mode to receive CMD and DAT. This bit is set by the tuning procedure and is valid after the completion of tuning (when Execute Tuning is cleared). Setting this bit to 1 means that tuning is completed successfully and setting this bit to 0 means that tuning has failed. The value is reflected on the sample_cclk_sel pin. This bit is irrelevant in UHS-II mode. Values: ■ 0x0 (FALSE): Fixed clock is used to sample data. ■ 0x1 (TRUE): Tuned clock is used to sample data. Value After Reset: 0x0 Exists: Always Volatile: True
6	EXEC_TUNING	R/W	Execute Tuning This bit is set to 1 to start the tuning procedure in UHSI/eMMC speed modes and this bit is automatically cleared when tuning procedure is completed. This bit is irrelevant in UHS-II mode. Values:

Bits	Name	Access	Description
			■ 0x0 (FALSE): Not tuned or Tuning completed ■ 0x1 (TRUE): Execute tuning Value After Reset: 0x0 Exists: Always Volatile: True
5:4	DRV_STRENGTH_SEL	R/W	Driver Strength Select This bit is used to select the host controller output driver in 1.8V signaling UHS-I/eMMC speed modes. The bit depends on setting of Preset Value Enable. The value is reflected on the uhs1_drv_sth pin. This bit is irrelevant in UHS-II mode. Values: ■ 0x0 (TYPEB): Driver TYPE B is selected. ■ 0x1 (TYPEA): Driver TYPE A is selected. ■ 0x2 (TYPEC): Driver TYPE C is selected. ■ 0x3 (TYPED): Driver TYPE D is selected. Value After Reset: 0x0 Exists: Always Volatile: True
3	SIGNALING_EN	R/W	1.8V Signaling Enable This bit controls voltage regulator for I/O cell in UHS-I/eMMC speed mode. Setting this bit from 0 to 1 starts changing the signal voltage from 3.3V to 1.8V. Host controller clears this bit if switching to 1.8 signaling fails. The value is reflected on the uhs1_swvolt_en pin. This bit shall be set to 0 in UHS-II mode Note: This bit must be set for all UHS-I speed modes (SDR12/SDR25/SDR50/SDR104/DDR50). Values: ■ 0x0 (V_3_3): 3.3V signalling ■ 0x1 (V_1_8): 1.8V signalling Value After Reset: 0x0 Exists: Always Volatile: True
2:0	UHS_MODE_SEL	R/W	UHS Mode/eMMC Speed Mode Select These bits are used to select UHS mode in SD mode of operation. In eMMC mode, these bits are used to select eMMC speed mode.

Bits	Name	Access	Description
			UHS Mode (SD/UHS-II mode only) ■ 0x0: SDR12 ■ 0x1: SDR25 ■ 0x2: SDR50 ■ 0x3: SDR104 ■ 0x4: DDR50 ■ 0x5: Reserved ■ 0x6: Reserved ■ 0x7: UHS-II eMMC Speed Mode (eMMC mode only) ■ 0x0: Legacy ■ 0x1: High speed SDR ■ 0x2: Reserved ■ 0x3: HS200 ■ 0x4: High speed DDR ■ 0x5: Reserved ■ 0x6: Reserved ■ 0x7: HS400 Values: ■ 0x0 (SDR12): SDR12/Legacy ■ 0x1 (SDR25): SDR25/High speed SDR ■ 0x2 (SDR50): SDR50 ■ 0x3 (SDR104): SDR104/HS200 ■ 0x4 (DDR50): DDR50/High speed DDR ■ 0x5 (RSVD5): Reserved ■ 0x6 (RSVD6): Reserved ■ 0x7 (UHS2): UHS-II/HS400 Value After Reset: 0x0 Exists: Always

3.6.1.28 CAPABILITIES1_R

- Name: Capabilities 1 Register - 0 to 31
- Description: This register provides the host driver with information specific to the host controller implementation. The host controller may implement these values as fixed or loaded from the flash memory during power on initialization. Capabilities register is segregated into

two 32-bit registers: CAPABILITIES1_R and CAPABILITIES2_R. The CAPABILITIES1_R register is the lower part of Capabilities register.

- Size: 32 bits
- Offset: 0x40

Figure & Table 3-71 Fields for register: CAPABILITIES1_R

Bits	Name	Access	Description
31:30	SLOT_TYPE_R	R	Slot Type These bits indicate usage of a slot by a specific host system. Values: ■ 0x0 (REMOVABLE_SLOT): Removable card slot ■ 0x1 (EMBEDDED_SLOT): Embedded slot for one device ■ 0x2 (SHARED_SLOT): Shared bus slot (SD mode) ■ 0x3 (UHS2_EMBEDDED_SLOT): UHS-II multiple embedded devices Value After Reset: 0x0 Exists: Always
29	ASYNC_INT_SUPPORT	R	Asynchronous Interrupt Support (SD mode only) Values: ■ 0x0 (FALSE): Asynchronous interrupt is not supported. ■ 0x1 (TRUE): Asynchronous interrupt is supported. Value After Reset: 0X1 Exists: Always
28	SYS_ADDR_64_V3	R	64-bit System Address Support for V3 This bit sets the host controller to support 64-bit system addressing of V3 mode. SDMA cannot be used in 64-bit addressing in Version 3 mode. If this bit is set to 1, 64-bit ADMA2 with using 96-bit descriptor can be enabled by setting Host Version 4 Enable (HOST_VER4_ENABLE = 0) and DMA select (DMA_SEL = 11b). Values: ■ 0x0 (FALSE): 64-bit system address for V3 is not supported. ■ 0x1 (TRUE): 64-bit system address for V3 is

Bits	Name	Access	Description
			supported. Value After Reset: 0X1 Exists: Always
27	SYS_ADDR_64_V4	R	64-bit System Address Support for V4 This bit sets the host controller to support 64-bit system addressing of V4 mode. When this bit is set to 1, full or part of 64-bit address must be used to decode the Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller registers is effective regardless of setting to 64-bit addressing in Host Control 2. If this bit is set to 1, 64-bit DMA Addressing for version 4 is enabled by setting Host Version 4 Enable (HOST_VER4_ENABLE = 1) and by setting 64-bit Addressing (ADDRESSING =1) in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit descriptor. Values: ■ 0x0 (FALSE): 64-bit system address for V4 is not supported. ■ 0x1 (TRUE): 64-bit system address for V4 is supported. Value After Reset: 0X1 Exists: Always
26	VOLT_18	R	Voltage Support for 1.8V Values: ■ 0x0 (FALSE): 1.8V is not supported. ■ 0x1 (TRUE): 1.8V is supported. Value After Reset: DWC_MSHC_VOLT18_VDD1_SUPPORT Exists: Always
25	VOLT_30	R	Voltage Support for SD 3.0V or Embedded 1.2V Values: ■ 0x0 (FALSE): SD 3.0V or Embedded 1.2V is not supported. ■ 0x1 (TRUE): SD 3.0V or Embedded is supported. Value After Reset: DWC_MSHC_VOLT30_VDD1_SUPPORT

Bits	Name	Access	Description
			Exists: Always
24	VOLT_33	R	Voltage Support for 3.3V Values: ■ 0x0 (FALSE): 3.3V is not supported. ■ 0x1 (TRUE): 3.3V is supported. Value After Reset: DWC_MSHC_VOLT33_VDD1_SUPPORT Exists: Always
23	SUS_RES_SUPPORT	R	Suspense/Resume Support This bit indicates whether the host controller supports Suspend/Resume functionality. If this bit is 0, the host driver does not issue either Suspend or Resume commands because the Suspend and Resume mechanism is not supported. Values: ■ 0x0 (FALSE): Not supported ■ 0x1 (TRUE): Supported Value After Reset: 0X0 Exists: Always
22	SDMA_SUPPORT	R	SDMA Support This bit indicates whether the host controller is capable of using SDMA to transfer data between the system memory and the host controller directly. Values: ■ 0x0 (FALSE): SDMA is not supported. ■ 0x1 (TRUE): SDMA is supported. Value After Reset: 0X1 Exists: Always
21	HIGH_SPEED_SUPPORT	R	High Speed Support This bit indicates whether the host controller and the host system supports High Speed mode and they can supply the SD Clock frequency from 25MHz to 50MHz. Values: ■ 0x0 (FALSE): High Speed is not supported. ■ 0x1 (TRUE): High Speed is supported. Value After Reset: 0X1

Bits	Name	Access	Description
			Exists: Always
20	RSVD_20	R	This bit of the CAPABILITIES1_R is a reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
19	ADMA2_SUPPORT	R	ADMA2 Support This bit indicates whether the host controller is capable of using ADMA2. Values: ■ 0x0 (FALSE): ADMA2 is not supported. ■ 0x1 (TRUE): ADMA2 is supported. Value After Reset: 0x1 Exists: Always
18	Embedded_8_BIT	R	8-bit Support for Embedded Device This bit indicates whether the host controller is capable of using an 8-bit bus width mode. This bit is not effective when the Slot Type is set to 10b. Values: ■ 0x0 (FALSE): 8-bit Bus Width is not supported. ■ 0x1 (TRUE): 8-bit Bus Width is supported. Value After Reset: 0x1 Exists: Always
17:16	MAX_BLK_LEN	R	Maximum Block Length This bit indicates the maximum block size that the host driver can read and write to the buffer in the host controller. The buffer transfers this block size without wait cycles. The transfer block length is always 512 bytes for the SD Memory irrespective of this bit. Values: ■ 0x0 (ZERO): 512 Byte ■ 0x1 (ONE): 1024 Byte ■ 0x2 (TWO): 2048 Byte ■ 0x3 (THREE): Reserved Value After Reset: 0x1 Exists: Always
15:8	BASE_CLK_FREQ	R	■ Base Clock Frequency for SD clock These bits indicate the base (maximum) clock

Bits	Name	Access	Description
			frequency for the SD Clock. The definition of these bits depend on the Host Controller Version. 6-Bit Base Clock Frequency: This mode is supported by the Host Controller version 1.00 and 2.00. The upper 2 bits are not effective and are always 0. The unit values are 1MHz. The supported clock range is 10MHz to 63MHz. – 0x00: Get information through another method. – 0x01: 1MHz – 0x02: 2MHz – ..... – 0x3F: 63MHz – 0x40-0xFF: Not supported ■ 8-Bit Base Clock Frequency: This mode is supported by the Host Controller version 3.00. The unit values are 1MHz. The supported clock range is 10MHz to 255MHz. – 0x00: Get information through another method. – 0x01: 1MHz – 0x02: 2MHz – ..... – 0xFF: 255MHz If the frequency is 16.5MHz, the larger value is set to 0001001b (17MHz) because the host driver uses this value to calculate the clock divider value and it does not exceed the upper limit of the SD Clock frequency. If these bits are all 0, the host system has to get information using a different method. Value After Reset: 0xC8 Exists: Always
7	TOUT_CLK_UNIT	R	Timeout Clock Unit This bit shows the unit of base clock frequency used to detect Data Timeout Error. Values: ■ 0x0 (KHZ): KHz ■ 0x1 (MHZ): MHz Value After Reset: 0x1 Exists: Always
6	RSVD_6	R	This bit of the CAPABILITIES1_R register is reserved. It always returns 0. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
5:0	TOUT_CLK_FREQ	R	Timeout Clock Frequency This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock unit defines the unit of timeout clock frequency. It can be KHz or MHz. ■ 0x00: Get information through another method. ■ 0x01: 1KHz/1MHz ■ 0x02: 2KHz/2MHz ■ 0x03: 3KHz/3MHz ■ ..... ■ 0x3F: 63KHz/63MHz Value After Reset: 0x1 Exists: Always

3.6.1.29 CAPABILITIES2_R

- Name: Capabilities Register - 32 to 63
- Description: This register provides the host driver with information specific to the host controller implementation. The host controller may implement these values as fixed or as loaded from flash memory during power on initialization. Capabilities register is segregated into two 32-bit registers, namely CAPABILITIES1_R and CAPABILITIES2_R. The CAPABILITIES2_R register is upper part of Capabilities register.
- Size: 32 bits
- Offset: 0x44

Figure & Table 3-72 Fields for register: CAPABILITIES2_R

Bits	Name	Access	Description
31:30	RSVD_62_63	R	These bits (RSVD_62_63) of the CAPABILITIES2_R register are reserved bits. They always return 0. Value After Reset: 0x0 Exists: Always
29	RSVD_61	R	This bit (RSVD_61) of the CAPABILITIES2_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
28	VDD2_18V_SUPPORT	R	1.8V VDD2 Support This bit indicates support of VDD2 for the host system. Values:

Bits	Name	Access	Description
			■ 0x0 (FALSE): 1.8V VDD2 is not supported. ■ 0x1 (TRUE): 1.8V VDD2 is supported. Value After Reset: 0x0 Exists: Always
27	ADMA3_SUPPORT	R	ADMA3 Support This bit indicates whether the host controller is capable of using ADMA3. Values: ■ 0x0 (FALSE): ADMA3 is not supported. ■ 0x1 (TRUE): ADMA3 is supported. Value After Reset: 0x1 Exists: Always
26:24	RSVD_56_58	R	These bits (RSVD_56_58) of the CAPABILITIES2_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
23:16	CLK_MUL	R	Clock Multiplier These bits indicate the clock multiplier of the programmable clock generator. Setting these bits to 0 means that the host controller does not support a programmable clock generator. ■ 0x0: Clock Multiplier is not supported. ■ 0x1: Clock Multiplier M = 2 ■ 0x2: Clock Multiplier M = 3 ■ ..... ■ 0xFF: Clock Multiplier M = 256 Value After Reset: 0x0 Exists: Always
15:14	RE_TUNING_MODES	R	Re-Tuning Modes (UHS-I only) These bits select the re-tuning method and limit the maximum data length. Values: ■ 0x0 (MODE1): Timer ■ 0x1 (MODE2): Timer and Re-Tuning Request (Not supported) ■ 0x2 (MODE3): Auto Re-Tuning (for transfer)

Bits	Name	Access	Description
			■ 0x3 (RSVD_MODE): Reserved Value After Reset: 0x2 Exists: Always
13	USE_TUNING_SDR50	R	Use Tuning for SDR50 (UHS-I only) Values: ■ 0x0 (ZERO): SDR50 does not require tuning. ■ 0x1 (ONE): SDR50 requires tuning. Value After Reset: 0x0 Exists: Always
12	RSVD_44	R	This bit (RSVD_44) of the CAPABILITIES2_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
11:8	RETUNE_CNT	R	Timer Count for Re-Tuning (UHS-I only) ■ 0x0: Re-Tuning Timer disabled ■ 0x1: 1 seconds ■ 0x2: 2 seconds ■ 0x3: 4 seconds ■ …….. ■ 0xB: 1024 seconds ■ 0xC: Reserved ■ 0xD: Reserved ■ 0xE: Reserved ■ 0xF: Get information from other sources. Value After Reset: 0x1 Exists: Always
7	RSVD_39	R	This bit (RSVD_39) of the CAPABILITIES2_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
6	DRV_TYPED	R	Driver Type D Support (UHS-I only) This bit indicates support of Driver Type D for 1.8 Signaling. Values: ■ 0x0 (FALSE): Driver Type D is not supported.

Bits	Name	Access	Description
			■ 0x1 (TRUE): Driver Type D is supported. Value After Reset: 0x1 Exists: Always
5	DRV_TYPEC	R	Driver Type C Support (UHS-I only) This bit indicates support of Driver Type C for 1.8 Signaling. Values: ■ 0x0 (FALSE): Driver Type C is not supported. ■ 0x1 (TRUE): Driver Type C is supported. Value After Reset: 0x1 Exists: Always
4	DRV_TYPEA	R	Driver Type A Support (UHS-I only) This bit indicates support of Driver Type A for 1.8 Signaling. Values: ■ 0x0 (FALSE): Driver Type A is not supported. ■ 0x1 (TRUE): Driver Type A is supported. Value After Reset: 0x1 Exists: Always
3	UHS2_SUPPORT	R	UHS-II Support (UHS-II only) This bit indicates whether host controller supports UHS-II. Values: ■ 0x0 (FALSE): UHS-II is not supported. ■ 0x1 (TRUE): UHS-II is supported. Value After Reset: 0x0 Exists: Always
2	DDR50_SUPPORT	R	DDR50 Support (UHS-I only) Values: ■ 0x0 (FALSE): DDR50 is not supported. ■ 0x1 (TRUE): DDR50 is supported. Value After Reset: 0x1 Exists: Always
1	SDR104_SUPPORT	R	SDR104 Support (UHS-I only) This bit means that SDR104 requires tuning.

Bits	Name	Access	Description
			Values: ■ 0x0 (FALSE): SDR104 is not supported. ■ 0x1 (TRUE): SDR104 is supported. Value After Reset: 0x1 Exists: Always
0	SDR50_SUPPORT	R	SDR50 Support (UHS-I only) This bit indicates that SDR50 is supported. The bit 13 (USE_TUNING_SDR50) indicates whether SDR50 requires tuning or not. Values: ■ 0x0 (FALSE): SDR50 is not supported. ■ 0x1 (TRUE): SDR50 is supported. Value After Reset: 0x1 Exists: Always

3.6.1.30 CURR_CAPABILITIES1_R

- Name: Maximum Current Capabilities Register - 0 to 31
- Description: This register indicates the maximum current capability for each voltage, for VDD1. The value is meaningful if the Voltage Support is set in the Capabilities register. If this information is supplied by the host system through another method, all the Maximum Current Capabilities registers are set to 0.
- Size: 32 bits
- Offset: 0x48

Figure & Table 3-73 Fields for register: CURR_CAPABILITIES1_R

Bits	Name	Access	Description
31:24	RSVD_31_24	R	These bits of the CURR_CAPABILITIES1_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
23:16	MAX_CUR_18V	R	Maximum Current for 1.8V This bit specifies the Maximum Current for 1.8V VDD1 power supply for the card. ■ 0: Get information through another method. ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA

Bits	Name	Access	Description
			■ ..... ■ 255: 1020mA Value After Reset: 0x19 Exists: Always
15:8	MAX_CUR_30V	R	Maximum Current for 3.0V This bit specifies the Maximum Current for 3.0V VDD1 power supply for the card. ■ 0: Get information through another method. ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA ■ ..... ■ 255: 1020mA Value After Reset: 0x19 Exists: Always
7:0	MAX_CUR_33V	R	Maximum Current for 3.3V This bit specifies the Maximum Current for 3.3V VDD1 power supply for the card. ■ 0: Get information through another method. ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA ■ ..... ■ 255: 1020mA Value After Reset: 0x19 Exists: Always

3.6.1.31 CURR_CAPABILITIES2_R

- Name: Maximum Current Capabilities Register - 32 to 63
- Description: This register indicates the maximum current capability for each voltage (for VDD2). The value is meaningful if Voltage Support is set in the Capabilities register. If this information is supplied by the host system through another method, all the Maximum Current Capabilities registers are set to 0.
- Size: 32 bits
- Offset: 0x4c

Figure & Table 3-74 Fields for register: CURR_CAPABILITIES2_R

Bits	Name	Access	Description
31:8	RSVD_63_40	R	These bits of the CURR_CAPABILITIES2_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
7:0	MAX_CUR_VDD2_18V	R	Maximum Current for 1.8V VDD2 This bit specifies the Maximum Current for 1.8V VDD2 power supply for the UHS-II card. ■ 0: Get information through another method. ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA ■ ..... ■ 255: 1020mA Value After Reset: 0x0 Exists: Always

3.6.1.32 FORCE_AUTO_CMD_STAT_R

- Name: Force Event Register for Auto CMD Error Status register
- Description: The register is not a physically implemented but is an address at which the Auto CMD Error Status register can be written. This register is applicable for SD/eMMC mode.
  - 1: Sets each bit of the Auto CMD Error Status register.
  - 0: No effect
- Size: 16 bits
- Offset: 0x50

Figure & Table 3-75 Fields for register: FORCE_AUTO_CMD_STAT_R

Bits	Name	Access	Description
15:8	RSVD_15_8	R	These bits of the FORCE_AUTO_CMD_STAT_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
7	FORCE_CMD_NOT_ISSUED_AUTO_CMD12	W	Force Event for Command Not Issued By Auto CMD12 Error Values: ■ 0x1 (TRUE): Command Not Issued By Auto CMD12 Error Status is set.

Bits	Name	Access	Description
			■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always
6	RSVD_6	R	This bit of the FORCE_AUTO_CMD_STAT_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
5	FORCE_AUTO_CMD_RESP_ER R	W	Force Event for Auto CMD Response Error Values: ■ 0x1 (TRUE): Auto CMD Response Error Status is set. ■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always
4	FORCE_AUTO_CMD_IDX_ERR	W	Force Event for Auto CMD Index Error Values: ■ 0x1 (TRUE): Auto CMD Index Error Status is set. ■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always
3	FORCE_AUTO_CMD_EBIT_ERR	W	Force Event for Auto CMD End Bit Error Values: ■ 0x1 (TRUE): Auto CMD End Bit Error Status is set. ■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always
2	FORCE_AUTO_CMD_CRC_ERR	W	Force Event for Auto CMD CRC Error Values: ■ 0x1 (TRUE): Auto CMD CRC Error Status is set. ■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always
1	FORCE_AUTO_CMD_TOUT_ER R	W	Force Event for Auto CMD Timeout Error Values: ■ 0x1 (TRUE): Auto CMD Timeout Error Status is set.

Bits	Name	Access	Description
			■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always
0	FORCE_AUTO_CMD12_NOT_EXEC	W	Force Event for Auto CMD12 Not Executed Values: ■ 0x1 (TRUE): Auto CMD12 Not Executed Status is set. ■ 0x0 (FALSE): Not affected. Value After Reset: 0x0 Exists: Always

3.6.1.33 FORCE_ERROR_INT_STAT_R

- Name: Force Event Register for Error Interrupt Status
- Description: This register is not physically implemented but is an address at which the Error Interrupt Status register can be written. The effect of a write to this address is reflected in the Error Interrupt Status register if the corresponding bit of the Error Interrupt Status Enable register is set. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 16 bits
- Offset: 0x52

Figure & Table 3-76 Fields for register: FORCE_ERROR_INT_STAT_R

Bits	Name	Access	Description
15	FORCE_VENDOR_ERR3	W	This bit (FORCE_VENDOR_ERR3) of the FORCE_ERROR_INT_STAT_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
14	FORCE_VENDOR_ERR2	W	This bit (FORCE_VENDOR_ERR2) of the FORCE_ERROR_INT_STAT_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
13	FORCE_VENDOR_ERR1	W	This bit (FORCE_VENDOR_ERR1) of the FORCE_ERROR_INT_STAT_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
12	FORCE_BOOT_ACK_ERR	W	Force Event for Boot Ack error

Bits	Name	Access	Description
			Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Boot ack Error Status is set. Value After Reset: 0x0 Exists: Always
11	FORCE_RESP_ERR	W	Force Event for Response Error (SD mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Response Error Status is set. Value After Reset: 0x0 Exists: Always
10	FORCE_TUNING_ERR	W	Force Event for Tuning Error (UHS-I Mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Tuning Error Status is set. Value After Reset: 0x0 Exists: Always
9	FORCE_ADMA_ERR	W	Force Event for ADMA Error Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): ADMA Error Status is set. Value After Reset: 0x0 Exists: Always
8	FORCE_AUTO_CMD_ERR	W	Force Event for Auto CMD Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Auto CMD Error Status is set. Value After Reset: 0x0 Exists: Always
7	FORCE_CUR_LMT_ERR	W	Force Event for Current Limit Error Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Current Limit Error Status is set. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
6	FORCE_DATA_END_BIT_ERR	W	Force Event for Data End Bit Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Data End Bit Error Status is set. Value After Reset: 0x0 Exists: Always
5	FORCE_DATA_CRC_ERR	W	Force Event for Data CRC Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Data CRC Error Status is set. Value After Reset: 0x0 Exists: Always
4	FORCE_DATA_TOUT_ERR	W	Force Event for Data Timeout Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Data Timeout Error Status is set. Value After Reset: 0x0 Exists: Always
3	FORCE_CMD_IDX_ERR	W	Force Event for Command Index Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Command Index Error Status is set. Value After Reset: 0x0 Exists: Always
2	FORCE_CMD_END_BIT_ERR	W	Force Event for Command End Bit Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Command End Bit Error Status is set. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
1	FORCE_CMD_CRC_ERR	W	Force Event for Command CRC Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Command CRC Error Status is set. Value After Reset: 0x0 Exists: Always
0	FORCE_CMD_TOUT_ERR	W	Force Event for Command Timeout Error (SD/eMMC mode only) Values: ■ 0x0 (FALSE): Not affected. ■ 0x1 (TRUE): Command Timeout Error Status is set. Value After Reset: 0x0 Exists: Always

3.6.1.34 ADMA_ERR_STAT_R

- Name: ADMA Error Status Register
- Description: This register stores the ADMA state during an ADMA error. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 8 bits
- Offset: 0x54

Figure & Table 3-77 Fields for register: ADMA_ERR_STAT_R

Bits	Name	Access	Description
7:3	RSVD_7_3	R	These bits of the ADMA_ERR_STAT_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
2	ADMA_LEN_ERR	R	ADMA Length Mismatch Error States This error occurs in the following instances: ■ While the Block Count Enable is being set, the total data length specified by the descriptor table is different from that specified by the Block Count and Block Length. ■ When the total data length cannot be divided by the block length.

Bits	Name	Access	Description
			Values: ■ 0x0 (NO_ERR): No error ■ 0x1 (ERROR): Error Value After Reset: 0x0 Exists: Always Volatile: True
1:0	ADMA_ERR_STATES	R	ADMA Error States These bits indicate the state of ADMA when an error occurs during ADMA data transfer. Values: ■ 0x0 (ST_STOP): Stop DMA - SYS_ADR register points to a location next to the error descriptor. ■ 0x1 (ST_FDS): Fetch Descriptor - SYS_ADR register points to the error descriptor. ■ 0x2 (UNUSED): Never set this state. ■ 0x3 (ST_TFR): Transfer Data - SYS_ADR register points to a location next to the error descriptor. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.35 ADMA_SA_LOW_R

- Name: ADMA System Address Register - Low
- Description: This register holds the lower 32-bit system address for DMA transfer. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 32 bits
- Offset: 0x58

Figure & Table 3-78 Fields for register: ADMA_SA_LOW_R

Bits	Name	Access	Description
31:0	ADMA_SA_LOW	R/W	ADMA System Address These bits indicate the lower 32 bits of the ADMA system address. ■ SDMA: If Host Version 4 Enable is set to 1, this register stores the system address of the data location. ■ ADMA2: This register stores the byte address of the executing command of the descriptor table.

Bits	Name	Access	Description
			■ ADMA3: This register is set by ADMA3. ADMA2 increments the address of this register that points to the next line, every time a descriptor line is fetched. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.36 ADMA_SA_HIGH_R

- Name: ADMA System Address Register - High
- Description: This register holds the upper 32-bit system address for the DMA transfer. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 32 bits
- Offset: 0x5c

Figure & Table 3-79 Fields for register: ADMA_SA_HIGH_R

Bits	Name	Access	Description
31:0	ADMA_SA_HIGH	R/W	ADMA System Address These bits indicate the upper 32 bits of the ADMA system address. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.37 PRESET_INIT_R

- Name: Preset Value for Initialization
- Description: This register defines Preset Value for Initialization in SD/eMMC mode.
- Size: 16 bits
- Offset: 0x60

Figure & Table 3-80 Fields for register: PRESET_INIT_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate that the driver strength is supported by 1.8V signaling bus speed modes. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected.

Bits	Name	Access	Description
			■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0x0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_INIT_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when the host controller supports a programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0X0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a host system. Value After Reset: 250 Exists: Always

3.6.1.38 PRESET_DS_R

- Name: Preset Value for Default Speed
- Description: This register defines Preset Value for Default Speed mode in SD mode.
- Size: 16 bits
- Offset: 0x62

Figure & Table 3-81 Fields for register: PRESET_DS_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate the driver strength value supported by 1.8V signaling bus speed modes. This field is meaningless for the default speed mode as it uses 3.3V signaling.

Bits	Name	Access	Description
			Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0x0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_DS_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when host controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System. Value After Reset: 4 Exists: Always

3.6.1.39 PRESET_HS_R

- Name: Preset Value for High Speed
- Description: This register defines Preset Value for High Speed mode in SD mode.
- Size: 16 bits
- Offset: 0x64

Figure & Table 3-82 Fields for register: PRESET_HS_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate the driver strength value supported

Bits	Name	Access	Description
			by 1.8V signaling bus speed modes. This field is meaningless for high speed mode as it uses 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0x0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_HS_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when host controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a host system. Value After Reset: 2 Exists: Always

3.6.1.40 PRESET_SDR12_R

- Name: Preset Value for SDR12
- Description: This register defines Preset Value for SDR12 and Legacy speed mode in SD and eMMC mode respectively.
- Size: 16 bits
- Offset: 0x66

Figure & Table 3-83 Fields for register: PRESET_SDR12_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate the driver strength value supported for the SDR12 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_SDR12_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when host controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a host system. Value After Reset: 4 Exists: Always

3.6.1.41 PRESET_SDR25_R

- Name: Preset Value for SDR25
- Description: This register defines Preset Value for SDR25 and High Speed SDR speed mode in SD and eMMC mode respectively.
- Size: 16 bits

- Offset: 0x68

Figure & Table 3-84 Fields for register: PRESET_SDR25_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate the driver strength value supported for the SDR25 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_SDR25_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System. Value After Reset: 2 Exists: Always

3.6.1.42 PRESET_SDR50_R

- Name: Preset Value for SDR50
- Description: This register defines Preset Value for SDR50 speed mode in SD mode.
- Size: 16 bits

- Offset: 0x6a

Figure & Table 3-85 Fields for register: PRESET_SDR50_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate driver strength value supported for SDR50 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_SDR50_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when host controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a host system. Value After Reset: 1 Exists: Always

3.6.1.43 PRESET_SDR104_R

- Name: Preset Value for SDR104
- Description: This register defines Preset Value for SDR104 and HS200 speed modes in SD and eMMC modes, respectively.

- Size: 16 bits
- Offset: 0x6c

Figure & Table 3-86 Fields for register: PRESET_SDR104_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate driver strength value supported for SDR104 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_SDR104_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when host controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value These bits specify a 10-bit preset value that must be set in the SDCLK/RCLK Frequency Select field of the Clock Control register described by a host system. Value After Reset: 0 Exists: Always

3.6.1.44 PRESET_DDR50_R

- Name: Preset Value for DDR50

- Description: This register defines the Preset Value for DDR50 and High Speed DDR speed modes in SD and eMMC modes, respectively.
- Size: 16 bits
- Offset: 0x6e

Figure & Table 3-87 Fields for register: PRESET_DDR50_R

Bits	Name	Access	Description
15:14	DRV_SEL_VAL	R	Driver Strength Select Value These bits indicate driver strength value supported for DDR50 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected. ■ 0x1 (TYPEA): Driver Type A is selected. ■ 0x2 (TYPEC): Driver Type C is selected. ■ 0x3 (TYPED): Driver Type D is selected. Value After Reset: 0 Exists: Always
13:11	RSVD_13_11	R	These bits of the PRESET_DDR50_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CLK_GEN_SEL_VAL	R	Clock Generator Select Value This bit is effective when host controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator Value After Reset: 0 Exists: Always
9:0	FREQ_SEL_VAL	R	SDCLK/RCLK Frequency Select Value These bits specify a 10-bit preset value that must be set in the SDCLK/RCLK Frequency Select field of the Clock Control register, as described by a host system. Value After Reset: 2 Exists: Always

3.6.1.45 ADMA_ID_LOW_R

- Name: ADMA3 Integrated Descriptor Address Register - Low
- Description: This register holds the lower 32-bit Integrated Descriptor address. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 32 bits
- Offset: 0x78

Figure & Table 3-88 Fields for register: ADMA_ID_LOW_R

Bits	Name	Access	Description
31:0	ADMA_ID_LOW	R/W	ADMA Integrated Descriptor Address These bits indicate the lower 32 bits of the ADMA Integrated Descriptor address. The start address of Integrated Descriptor is set to these register bits. The ADMA3 fetches one Descriptor Address and increments these bits to indicate the next descriptor address. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.46 ADMA_ID_HIGH_R

- Name: ADMA3 Integrated Descriptor Address Register - High
- Description: This register holds the upper 32-bit Integrated Descriptor address. This register is applicable for SD/eMMC/UHS-II mode.
- Size: 32 bits
- Offset: 0x7c

Figure & Table 3-89 Fields for register: ADMA_ID_HIGH_R

Bits	Name	Access	Description
31:0	ADMA_ID_HIGH	R/W	ADMA Integrated Descriptor Address These bits indicate the upper 32 bits of the ADMA Integrated Descriptor address. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.47 SLOT_INTR_STATUS_R

- Name: Slot Interrupt Status Register
- Description: This register indicates the interrupt status of each slot.
- Size: 16 bits

- Offset: 0xfc

Figure & Table 3-90 Fields for register: SLOT_INTR_STATUS_R

Bits	Name	Access	Description
15:8	RESERVED_15_8	R	These bits of the SLOT_INTR_STATUS_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
7:0	INTR_SLOT	R	Interrupt signal for each Slot These status bits indicate the logical OR of interrupt signal and wakeup signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the host driver can identify the interrupt that is generated by reading these bits. By a power on reset or by setting Software Reset For All bit, the interrupt signals are de-asserted and this status reads 00h. ■ Bit 00: Slot 1 ■ Bit 01: Slot 2 ■ Bit 02: Slot 3 ■ ..... ■ ..... ■ Bit 07: Slot 8 Note: MSHC host controller support single card slot. This register shall always return 0. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.1.48 HOST_CNTRL_VERS_R

- Name: Host Controller Version
- Description: This register is used to indicate the Host Controller Version number.
- Size: 16 bits
- Offset: 0xfe

Figure & Table 3-91 Fields for register: HOST_CNTRL_VERS_R

Bits	Name	Access	Description
15:8	VENDOR_VERSION_NUM	R	Vendor Version Number This field is reserved for the vendor version number.

Bits	Name	Access	Description
			Host driver must not use this status. Value After Reset: 0 Exists: Always
7:0	SPEC_VERSION_NUM	R	Specification Version Number These bits indicate the host controller specification version. The upper and lower 4 bits indicate the version. Values 0x06-0xFF are reserved. Values: ■ 0x0 (VER_1_00): SD Host Controller Specification Version 1.00 ■ 0x1 (VER_2_00): SD Host Controller Specification Version 2.00 ■ 0x2 (VER_3_00): SD Host Controller Specification Version 3.00 ■ 0x3 (VER_4_00): SD Host Controller Specification Version 4.00 ■ 0x4 (VER_4_10): SD Host Controller Specification Version 4.10 ■ 0x5 (VER_4_20): SD Host Controller Specification Version 4.20 Value After Reset: 5 Exists: Always

3.6.2 DWC_mshc_map/DWC_mshc_embedded_control_block Registers

3.6.2.1 EMBEDDED_CTRL_R

- Name: Embedded Control Register
- Description: This register controls the embedded device. When the host controller is connected to a removable device, this register is not used.
- Size: 32 bits
- Offset: 0xf6c

Figure & Table 3-92 Fields for Register: EMBEDDED_CTRL_R

Bits	Name	Access	Description
31	RSVD_31	R	This bit (RSVD_31) of the EMBEDDED_CTRL_R register is reserved. It always returns 0. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
30:24	BACK_END_PWR_CTRL	R/W	Back-End Power Control (SD mode) Each bit of this field controls back-end power supply for an embedded device. ■ 0: Back-End Power is off. ■ 1: Back-End Power is supplied. D24: Back-End Power for Device 1 D25: Back-End Power for Device 2 D26: Back-End Power for Device 3 D27: Back-End Power for Device 4 D28: Back-End Power for Device 5 D29: Back-End Power for Device 6 D30: Back-End Power for Device 7 Value After Reset: 0x0 Exists: Always
23	RSVD_23	R	This bit (RSVD_23) of the EMBEDDED_CTRL_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
22:20	INT_PIN_SEL	R/W	Interrupt Pin Select These bits enable the interrupt pin inputs. ■ 000: Interrupts (INT_A, INT_B, INT_C) are disabled. ■ xx1: INT_A is enabled. ■ x1x: INT_B is enabled. ■ 1xx: INT_C is enabled. Value After Reset: 0x0 Exists: Always
19	RSVD_19	R	This bit (RSVD_19) of the EMBEDDED_CTRL_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
18:16	CLK_PIN_SEL	R/W	Clock Pin Select (SD Mode) This bit is selected by one of clock pin outputs. ■ 0x0: Clock pins are disabled. ■ 0x1: CLK [1] is selected.

Bits	Name	Access	Description
			■ 0x2: CLK [2] is selected. ■ . . ■ . . ■ . . ■ 0x7: CLK [7] is selected. Value After Reset: 0x0 Exists: Always
15	RSVD_15	R	This bit (RSVD_15) of the EMBEDDED_CTRL_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
14:8	BUS_WIDTH_PRESET	R	Bus Width Preset (SD mode) Each bit of this field specifies the bus width for each embedded device. The shared bus supports mixing of 4-bit and 8-bit bus width devices. ■ D08: Bus Width Preset for Device 1 ■ D09: Bus Width Preset for Device 2 ■ D10: Bus Width Preset for Device 3 ■ D11: Bus Width Preset for Device 4 ■ D12: Bus Width Preset for Device 5 ■ D13: Bus Width Preset for Device 6 ■ D14: Bus Width Preset for Device 7 Function of each bit is defined as follows: ■ 0: 4-bit bus width mode ■ 1: 8-bit bus width mode Value After Reset: 0x0 Exists: Always
7:6	RSVD_7_6	R	These bits (RSVD_7_6) of the EMBEDDED_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
5:4	NUM_INT_PIN	R	Number of Interrupt Input Pins This field indicates support of interrupt input pins for an embedded system. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
3	RSVD_3	R	This bit (RSVD_3) of the EMBEDDED_CTRL_R register is reserved. It always returns 0. Value After Reset: 0x0 Exists: Always
2:0	NUM_CLK_PIN	R	Number of Clock Pins (SD mode) This field indicates support of clock pins to select one of devices for shared bus system. Up to 7 clock pins can be supported. ■ 0x0: Shared bus is not supported. ■ 0x1: 1 SDCLK is supported. ■ 0x2 - 2 SDCLK is supported. ■ . . ■ . . ■ . . ■ 0x7: 7 SDCLK is supported. Value After Reset: 0x0 Exists: Always

3.6.3 DWC_mshc_map/DWC_mshc_vendor2_block Registers

3.6.3.1 CQVER

- Name: Command Queuing Version Register
- Description: This register provides information about the version of the eMMC command queueing standard, which is implemented by the CQE in BCD format.
- Size: 32 bits
- Offset: 0x180

Figure & Table 3-93 Fields for register: CQVER

Bits	Name	Access	Description
31:12	EMMMC_VER_RSVD	R	These bits of the CQVER register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
11:8	EMMC_VER_MAJOR	R	This bit indicates the eMMC major version (1st digit left of decimal point) in BCD format. Value After Reset: 0x5 Exists: Always

Bits	Name	Access	Description
7:4	EMMC_VER_MINOR	R	This bit indicates the eMMC minor version (1st digit right of decimal point) in BCD format. Value After Reset: 0x1 Exists: Always
3:0	EMMC_VER_SUFFIX	R	This bit indicates the eMMC version suffix (2nd digit right of decimal point) in BCD format. Value After Reset: 0x0 Exists: Always

3.6.3.2 CQCAP

- Name: Command Queuing Capabilities Register
- Description: This register indicates the capabilities of the command queuing engine.
- Size: 32 bits
- Offset: 0x180 + 0x4

Figure & Table 3-94 Fields for Register: CQCAP

Bits	Name	Access	Description
31:29	CQCCAP_RSVD3	R	These bits [31:29] of the CQCAP register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
28	CRYPTO_SUPPORT	R	Crypto Support This bit indicates whether the Host controller supports cryptographic operations. Values: ■ 0x0 (FALSE): Crypto is not supported. ■ 0x1 (TRUE): Crypto is supported. Value After Reset: 0 Exists: Always
27:16	CQCCAP_RSVD2	R	These bits [27:16] of the CQCAP register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
15:12	ITCFMUL	R	Internal Timer Clock Frequency Multiplier (ITCFMUL) This field indicates the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details.

Bits	Name	Access	Description
			Values 0x5 to 0xF are reserved. Values: ■ 0x0 (CLK_1KHz): 1KHz clock ■ 0x1 (CLK_10KHz): 10KHz clock ■ 0x2 (CLK_100KHz): 100KHz clock ■ 0x3 (CLK_1MHz): 1MHz clock ■ 0x4 (CLK_10MHz): 10MHz clock Value After Reset: 3 Exists: Always
11:10	CQCCAP_RSVD1	R	These bits of the CQCAP register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
9:0	ITCFVAL	R	Internal Timer Clock Frequency Value (ITCFVAL) This field scales the frequency of the timer clock provided by ITCFMUL. The final clock frequency of actual timer clock is calculated as ITCFVAL* ITCFMUL. Value After Reset: 0 Exists: Always

3.6.3.3 CQCFG

- Name: Command Queuing Configuration Register
- Description: This register controls CQE behavior affecting the general operation of command queuing engine.
- Size: 32 bits
- Offset: 0x180 + 0x8

Figure & Table 3-95 Fields for register: CQCFG

Bits	Name	Access	Description
31:13	CQCCFG_RSVD3	R	These bits (CQCCFG_RSVD3) of the CQCFG register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
12	DCMD_EN	R/W	This bit indicates to the hardware whether the task descriptor in slot #31 of the TDL is a data transfer descriptor or a direct-command descriptor. CQE uses this bit when a task is issued in slot #31, to determine

Bits	Name	Access	Description
			how to decode the task descriptor. Values: ■ 0x1 (SLOT31_DCMD_ENABLE): Task descriptor in slot #31 is a DCMD task descriptor. ■ 0x0 (SLOT31_DCMD_DISABLE): Task descriptor in slot #31 is a data transfer task descriptor. Value After Reset: 0x0 Exists: Yes
11:9	CQCCFG_RSVD2	R	These bits (CQCCFG_RSVD2) of the CQCFG register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
8	TASK_DESC_SIZE	R/W	Bit Value Description This bit indicates the size of task descriptor used in host memory. This bit can only be configured when Command Queuing Enable bit is 0 (Command queuing is disabled). Values: ■ 0x1 (TASK_DESC_128b): Task descriptor size is 128 bits. ■ 0x0 (TASK_DESC_64b): Task descriptor size is 64 bits. Value After Reset: 0x0 Exists: Yes
7:2	CQCCFG_RSVD1	R	These bits (CQCCFG_RSVD1) of the CQCFG register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
1	Reserved	R	
0	CQ_EN	R/W	Enable command queuing engine (CQE) When CQE is disabled, the software controls the eMMC bus using the registers between the addresses 0x000 to 0x1FF. Before the software writes to this bit, the software verifies that the eMMC host controller is in idle state and there are no ongoing commands or data transfers. When software wants to exit command queuing mode, it clears all previous tasks (if any) before setting this bit to 0.

Bits	Name	Access	Description
			Values: ■ 0x1 (CQE_ENABLE): Enable command queuing. ■ 0x0 (CQE_DISABLE): Disable command queuing. Value After Reset: 0x0 Exists: Yes Testable: Restore

3.6.3.4 CQCTL

- Name: Command Queuing Control Register
- Description: This register controls CQE behavior affecting the general operation of command queuing module or simultaneous operation of multiple tasks.
- Size: 32 bits
- Offset: 0x180 + 0xc

Figure & Table 3-96 Fields for register: CQCTL

Bits	Name	Access	Description
31:9	CQCTL_RSVD2	R	These bits (CQCTL_RSVD2) of the CQCTL register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
8	CLR_ALL_TASKS	R/W	Clear all tasks This bit can only be written when the controller is halted. This bit does not clear tasks in the device. The software has to use the CMDQ_TASK_MGMT command to clear device's queue. Values: ■ 0x1 (CLEAR_ALL_TASKS): Clears all the tasks in the controller. ■ 0x0 (NO_EFFECT): Programming 0 has no effect. Value After Reset: 0x0 Exists: Always Volatile: True
7:1	CQCTL_RSVD1	R	These bits (CQCTL_RSVD1) of the CQCTL register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
			Volatile: True
0	HALT	R/W	Halt request and resume Values: ■ 0x1 (HALT_CQE): Software writes 1 to this bit when it wants to acquire software control over the eMMC bus and to disable CQE from issuing command on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT). When the software writes 1, CQE completes the ongoing task (if any in progress). After the task is completed and the CQE is in idle state, CQE does not issue new commands and indicates to the software by setting this bit to 1. The software can poll on this bit until it is set to 1 and only then send commands on the eMMC bus. ■ 0x0 (RESUME_CQE): Software writes 0 to this bit to exit from the halt state and resume CQE activity. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.5 CQIS

- Name: Command Queuing Interrupt Status Register
- Description: This register indicates pending interrupts that require service. Each bit in this register is asserted in response to a specific event, only if the respective bit is set in the CQISE register.
- Size: 32 bits
- Offset: 0x180 + 0x10

Figure & Table 3-97 Fields for register: CQIS

Bits	Name	Access	Description
31:6	CQIS_RSVD1	R	These bits of the CQIS register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always Volatile: True
5	Reserved	R	
4	Reserved	R	

Bits	Name	Access	Description
3	TCL	R/W1C	Task cleared interrupt This status bit is asserted (if CQISE.TCL_STE=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (by writing CQTCLR) or clearing of all tasks (by writing CQCTL). A value of 1 clears this status bit. Values: ■ 0x1 (SET): TCL interrupt is set. ■ 0x0 (NOTSET): TCL interrupt is not set. Value After Reset: 0x0 Exists: Always Volatile: True
2	RED	R/W1C	Response error detected interrupt This status bit is asserted (if CQISE.RED_STE=1) when a response is received with an error bit set in the device status field. Configure the CQRMEM register to identify device status bit fields that may trigger an interrupt and that are masked. A value of 1 clears this status bit. Values: ■ 0x1 (SET): RED interrupt is set. ■ 0x0 (NOTSET): RED interrupt is not set. Value After Reset: 0x0 Exists: Always Volatile: True
1	TCC	R/W1C	Task complete interrupt This status bit is asserted (if CQISE.TCC_STE=1) when at least one of the following conditions are met: ■ A task is completed and the INT bit is set in its task descriptor. ■ Interrupt caused by interrupt coalescing logic due to timeout. ■ Interrupt coalescing logic reached the configured threshold. ■ A value of 1 clears this status bit. Values: ■ 0x1 (SET): TCC interrupt is set. ■ 0x0 (NOTSET): TCC interrupt is not set. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always Volatile: True
0	HAC	R/W1C	Halt complete interrupt This status bit is asserted (only if CQISE.HAC_STE=1) when halt bit in the CQCTL register transitions from 0 to 1 indicating that the host controller has completed its current ongoing task and has entered halt state. A value of 1 clears this status bit. Values: ■ 0x1 (SET): HAC interrupt is set. ■ 0x0 (NOTSET): HAC interrupt is not set. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.6 CQISE

- Name: Command Queuing Interrupt Status Enable Register
- Description: This register enables and disables the reporting of the corresponding interrupt to host software in the CQIS register. When a bit is set (1) and the corresponding interrupt condition is active, then the bit in CQIS is asserted. Interrupt sources that are disabled (when '0') are not indicated in the CQIS register. This register is bit-index matched to the CQIS register.
- Size: 32 bits
- Offset: 0x180 + 0x14

Figure & Table 3-98 Fields for register: CQISE

Bits	Name	Access	Description
31:6	CQISTE_RSVD1	R	These bits of the CQISE register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
5	Reserved	R	
4	Reserved	R	
3	TCL_STE	R/W	Task cleared interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.TCL is set when its interrupt condition is active. ■ 0x0 (INT_STS_DISABLE): CQIS.TCL is disabled.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
2	RED_STE	R/W	Response error detected interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.RED is set when its interrupt condition is active. ■ 0x0 (INT_STS_DISABLE): CQIS.RED is disabled. Value After Reset: 0x0 Exists: Always
1	TCC_STE	R/W	Task complete interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.TCC is set when its interrupt condition is active. ■ 0x0 (INT_STS_DISABLE): CQIS.TCC is disabled. Value After Reset: 0x0 Exists: Always
0	HAC_STE	R/W	Halt complete interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.HAC is set when its interrupt condition is active. ■ 0x0 (INT_STS_DISABLE): CQIS.HAC is disabled. Value After Reset: 0x0 Exists: Always

3.6.3.7 CQISGE

- Name: Command Queuing Interrupt Signal Enable Register
- Description: This register enables and disables the generation of interrupts to host software. When a bit is set and the corresponding bit in CQIS is set, then an interrupt is generated. Interrupt sources that are disabled are still indicated in the CQIS register. This register is bit-index matched to the CQIS register.
- Size: 32
- Offset: 0x180 + 0x18bits

Figure & Table 3-99 Fields for register: CQISGE

Bits	Name	Access	Description
31:6	CQISGE_RSVD1	R	These bits of the CQISGE register are reserved. They always return 0.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
5	Reserved	R	
4	Reserved	R	
3	TCL_SGE	R/W	Task cleared interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.TCL interrupt signal generation is active. ■ 0x0 (INT_SIG_DISABLE): CQIS.TCL interrupt signal generation is disabled. Value After Reset: 0x0 Exists: Always
2	RED_SGE	R/W	Response error detected interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.RED interrupt signal generation is active. ■ 0x0 (INT_SIG_DISABLE): CQIS.RED interrupt signal generation is disabled. Value After Reset: 0x0 Exists: Always
1	TCC_SGE	R/W	Task complete interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.TCC interrupt signal generation is active. ■ 0x0 (INT_SIG_DISABLE): CQIS.TCC interrupt signal generation is disabled. Value After Reset: 0x0 Exists: Always
0	HAC_SGE	R/W	Halt complete interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.HAC interrupt signal generation is active. ■ 0x0 (INT_SIG_DISABLE): CQIS.HAC interrupt signal generation is disabled. Value After Reset: 0x0 Exists: Always

3.6.3.8 CQIC

- Name: Command Queuing Interrupt Coalescing Register
- Description: This register controls and configures interrupt coalescing feature.
- Size: 32 bits
- Offset: 0x180 + 0x1c

Figure & Table 3-100 Fields for register: CQIC

Bits	Name	Access	Description
31	INTC_EN	R/W	Interrupt Coalescing Enable Bit Values: ■ 0x1 (ENABLE_INT_COALESCING): Interrupt coalescing mechanism is active. Interrupts are counted and timed, and coalesced interrupts are generated. ■ 0x0 (DISABLE_INT_COALESCING): Interrupt coalescing mechanism is disabled (default). Value After Reset: 0x0 Exists: Always
30:21	CQIC_RSVD3	R	These bits (CQIC_RSVD3) of the CQIC register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
20	INTC_STAT	R	Interrupt Coalescing Status Bit This bit indicates to the software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (that is, this is set if and only if INTC counter > 0). Values: ■ 0x1 (INTC_ATLEAST1_COMP): At least one INT0 task completion has been counted (INTC counter > 0). ■ 0x0 (INTC_NO_TASK_COMP): INT0 Task completions have not occurred since last counter reset (INTC counter == 0). Value After Reset: 0x0 Exists: Always
19:17	CQIC_RSVD2	R	These bits (CQIC_RSVD2) of the CQIC register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
16	INTC_RST	W	Counter and Timer Reset When host driver writes 1, the interrupt coalescing timer and counter are reset. Values: ■ 0x1 (ASSERT_INTC_RESET): Interrupt coalescing timer and counter are reset. ■ 0x0 (NO_EFFECT): No effect Value After Reset: 0x0 Exists: Always
15	INTC_TH_WEN	W	Interrupt Coalescing Counter Threshold Write Enable When software writes 1 to this bit, the value INTC_TH is updated with the contents written on the same cycle. Values: ■ 0x1 (WEN_SET): Sets INTC_TH_WEN. ■ 0x0 (WEN_CLR): Clears INTC_TH_WEN. Value After Reset: 0x0 Exists: Always
14:13	CQIC_RSVD1	R	These bits (CQIC_RSVD1) of the CQIC register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
12:8	INTC_TH	W	Interrupt Coalescing Counter Threshold field Software uses this field to configure the number of task completions (only tasks with INT=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 by CQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in INTC_TH, and generates interrupt. ■ 0x0: Interrupt coalescing feature disabled. ■ 0x1: Interrupt coalescing interrupt generated after 1 task when INT=0 completes. ■ 0x2: Interrupt coalescing interrupt generated after 2 tasks when INT=0 completes. ■ …….. ■ 0x1f: Interrupt coalescing interrupt generated after

Bits	Name	Access	Description
			31 tasks when INT=0 completes. To write to this field, the INTC_TH_WEN bit must be set during the same write operation. Value After Reset: 0x0 Exists: Always Testable: Untestable
7	TOUT_VAL_WEN	W	When software writes 1 to this bit, the value TOUT_VAL is updated with the contents written on the same cycle. Values: ■ 0x1 (WEN_SET): Sets TOUT_VAL_WEN. ■ 0x0 (WEN_CLR): Clears TOUT_VAL_WEN. Value After Reset: 0x0 Exists: Always
6:0	TOUT_VAL	R/W	Interrupt Coalescing Timeout Value 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 the first 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. ■ 0x0: Timer is disabled. Timeout-based interrupt is not generated. ■ 0x1: Timeout on 01x1024 cycles of timer clock frequency. ■ 0x2: Timeout on 02x1024 cycles of timer clock frequency. ■ …….. ■ 0x7f: Timeout on 127x1024 cycles of timer clock frequency. In order to write to this field, the TOUT_VAL_WEN bit must be set at the same write operation. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
			Testable: Untestable

3.6.3.9 CQTDLBA

- Name: Command Queuing Task Descriptor List Base Address Register
- Description: This register is used for configuring the lower 32 bits of the byte address of the head of the task descriptor list in the host memory.
- Size: 32 bits
- Offset: 0x180 + 0x20

Figure & Table 3-101 Fields for register: CQTDLBA

Bits	Name	Access	Description
31:0	TDLBA	R/W	This register stores the LSB 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 the host driver. This address is set on 1KB boundary. The lower 10 bits of this register are set to 0 by the software and are ignored by CQE. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.10 CQTDLBAU

- Name: Command Queuing Task Descriptor List Base Address Upper Register
- Description: This register is used for configuring the upper 32 bits of the byte address of the head of the task descriptor list in the host memory.
- Size: 32 bits
- Offset: 0x180 + 0x24

Figure & Table 3-102 Fields for register: CQTDLBAU

Bits	Name	Access	Description
31:0	TDLBAU	R/W	This register stores the MSB 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 address is set on 1KB boundary. The lower 10 bits of this register are set to 0 by the software and are ignored by CQE. This register is reserved when using 32-bit addressing mode. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.11 CQTDBR

- Name: Command Queuing DoorBell Register
- Description: Using this register, software triggers CQE to process a new task.
- Size: 32 bits
- Offset: 0x180 + 0x28

Figure & Table 3-103 Fields for register: CQTDBR

Bits	Name	Access	Description
31:0	DBR	R/W	The software configures 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. Writing 0 by the software does not have any impact on the hardware, and does not change the value of the register bit. CQE always processes tasks 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 task execution is completed (with success or

Bits	Name	Access	Description
			error). ■ The task is cleared using CQTCLR register. ■ All tasks are cleared using CQCTL register. ■ 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 processes 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 is based on said processing order. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.12 CQTCN

- Name: Command Queuing Task Clear Notification Register
- Description: This register is used by CQE to notify software about completed tasks.
- Size: 32 bits
- Offset: 0x180 + 0x2c

Figure & Table 3-104 Fields for register: CQTCN

Bits	Name	Access	Description
31:0	TCN	R/W1C	Task Completion Notification Each of the 32 bits are bit mapped to the 32 tasks. ■ Bit-N(1): Task-N has completed execution (with success or errors). ■ Bit-N(0): Task-N has not completed, could be pending or not submitted. On task completion, software may read this register to know tasks that have completed. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.13 CQDQS

- Name: Device Queue Status Register
- Description: This register stores the most recent value of the device's queue status.
- Size: 32 bits
- Offset: 0x180 + 0x30

Figure & Table 3-105 Fields for register: CQDQS

Bits	Name	Access	Description
31:0	DQS	R	Device Queue Status Each of the 32 bits are bit mapped to the 32 tasks. ■ Bit-N (1): Device has marked task N as ready for execution. ■ Bit-N (0): Task-N is not ready for execution. This task could be pending in device or not submitted. Host controller updates this register with response of the Device Queue Status command. Value After Reset: 0x0 Exists: Always

3.6.3.14 CQDPT

- Name: Device Pending Tasks Register
- Description: This register maintains the list of tasks that are queued into device and are awaiting execution completion.
- Size: 32 bits
- Offset: 0x180 + 0x34

Figure & Table 3-106 Fields for register: CQDPT

Bits	Name	Access	Description
31:0	DPT	R	Device-Pending Tasks Each of the 32 bits are bit mapped to the 32 tasks. ■ Bit-N (1): Task-N has been successfully queued into the device and is awaiting execution. ■ Bit-N (0): Task-N is not yet queued. 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 has not been executed. The controller sets this bit after receiving a successful response for CMD45. CQE clears this bit after the task has completed execution.

Bits	Name	Access	Description
			Software reads this register in the task-discard procedure to determine if the task is queued in the device. Value After Reset: 0x0 Exists: Always

3.6.3.15 CQTCLR

- Name: Command Queuing DoorBell Register
- Description: This register is used for removing an outstanding task in the CQE. The register must be used only when CQE is in halt state.
- Size: 32 bits
- Offset: 0x180 + 0x38

Figure & Table 3-107 Fields for register: CQTCLR

Bits	Name	Access	Description
31:0	TCLR	R/W	Writing 1 to bit n of this register orders CQE to clear a task that the 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 the clear operation is complete. Software must poll on the CQTCLR until it is cleared to verify that a clear operation was done. Value After Reset: 0x0 Exists: Always Volatile: True

3.6.3.16 CQSSC1

- Name: CQ Send Status Configuration 1 Register
- Description: This register is used for removing an outstanding task in the CQE. The register controls when SEND_QUEUE_STATUS commands are sent.
- Size: 32 bits
- Offset: 0x180 + 0x40

Figure & Table 3-108 Fields for register: CQSSC1

Bits	Name	Access	Description
31:20	RSVD_20_31	R	These bits of the CQSSC1 register are reserved. They always return 0.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
19:16	SQSCMD_BLK_CNT	R/W	This field indicates when SQS CMD is sent while data transfer is in progress. A value of 'n' indicates that CQE sends status command on the CMD line, during the transfer of data block BLOCK_CNTn, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. ■ 0x0: SEND_QUEUE_STATUS (CMD13) command is not sent during the transaction. Instead, it is sent only when the data lines are idle. ■ 0x1: SEND_QUEUE_STATUS command is to be sent during the last block of the transaction. ■ 0x2: SEND_QUEUE_STATUS command when last 2 blocks are pending. ■ 0x3: SEND_QUEUE_STATUS command when last 3 blocks are pending. ■ …….. ■ 0xf: SEND_QUEUE_STATUS command when last 15 blocks are pending. Should be programmed only when CQCFG.CQ_EN is 0. Value After Reset: 0x1 Exists: Always
15:0	SQSCMD_IDLE_TMR	R/W	This field configures the polling period to be used 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 indicates 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 register. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). For example, a CQCAP field value of 0 indicates a 19.2MHz clock frequency (period = 52.08ns). If the setting in CQSSC1.CIT is 1000h, the calculated polling period is 4096*52.08ns= 213.33ns. Should be programmed only when CQCFG.CQ_EN is 0.

Bits	Name	Access	Description
			Value After Reset: 0x1000 Exists: Always

3.6.3.17 CQSSC2

- Name: CQ Send Status Configuration 2 Register
- Description: The register is used for configuring the RCA field in SEND_QUEUE_STATUS command argument.
- Size: 32 bits
- Offset: 0x180 + 0x44

Figure & Table 3-109 Fields for register: CQSSC2

Bits	Name	Access	Description
31:16	RSVD_16_31	R	These bits of the CQSSC2 register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
15:0	SQSCMD_RCA	R/W	This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_STATUS (CMD13) command argument. CQE copies this field to bits[31:16] of the argument when transmitting SEND_QUEUE_STATUS (CMD13) command. Value After Reset: 0x0 Exists: Always

3.6.3.18 CQCRDCT

- Name: Command Response for Direct Command Register
- Description: This register stores the response of last executed DCMD.
- Size: 32 bits
- Offset: 0x180 + 0x48

Figure & Table 3-110 Fields for register: CQCRDCT

Bits	Name	Access	Description
31:0	DCMD_RESP	R	This register contains the response of the command generated by the last direct command (DCMD) task that was sent. Contents of this register are valid only after bit 31 of CQTDBR register is cleared by the controller. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always

3.6.3.19 CQRMEM

- Name: Command Response Mode Error Mask Register
- Description: This register controls the generation of response error detect (RED) interrupt. Only the bits enabled here can contribute to RED.
- Size: 32 bits
- Offset: 0x180 + 0x50

Figure & Table 3-111 Fields for register: CQRMEM

Bits	Name	Access	Description
31:0	RESP_ERR_MASK	R/W	The bits of this field are bit mapped to the device response. This bit is used as an interrupt mask on the device status filed that is received in R1/R1b responses. ■ 1: When a R1/R1b response is received, with a 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 that they are ignored by this logic. Value After Reset: 0xfdf9a080 Exists: Always

3.6.3.20 CQTERRI

- Name: CQ Task Error Information Register
- Description: This register is updated by CQE when an error occurs on data or command related to a task activity. When such an error is detected by CQE or indicated by the eMMC controller, CQE stores the following in the CQTERRI register: task IDs and indices of commands that were executed on the command line and data lines when the error occurred. Software must use this information in the error recovery procedure.
- Size: 32 bits
- Offset: 0x180 + 0x54

Figure & Table 3-112 Fields for register: CQTERRI

Bits	Name	Access	Description
31	TRANS_ERR_FIELDS_VALID	R	This bit is updated when an error is detected while a

Bits	Name	Access	Description
			data transfer transaction was in progress. Values: ■ 0x1 (SET): Data transfer related error detected. Check contents of TRANS_ERR_TASKID and TRANS_ERR_CMD_INDX fields. ■ 0x0 (NOT_SET): Ignore contents of TRANS_ERR_TASKID and TRANS_ERR_CMD_INDX. Value After Reset: 0x0 Exists: Always
30:29	RSVD_30_29	R	These bits (RSVD_30_29) of the CQTERRI register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
28:24	TRANS_ERR_TASKID	R	This field captures the ID of the task that was executed and whose data transfer has errors. Value After Reset: 0x0 Exists: Always
23:22	RSVD_23_22	R	These bits (RSVD_23_22) of the CQTERRI register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
21:16	TRANS_ERR_CMD_INDX	R	This field captures the index of the command that was executed and whose data transfer has errors. Value After Reset: 0x0 Exists: Always
15	RESP_ERR_FIELDS_VALID	R	This bit is updated when an error is detected while a command transaction was in progress. Values: ■ 0x1 (SET): Response-related error is detected. Check contents of RESP_ERR_TASKID and RESP_ERR_CMD_INDX fields. ■ 0x0 (NOT_SET): Ignore contents of RESP_ERR_TASKID and RESP_ERR_CMD_INDX. Value After Reset: 0x0 Exists: Always
14:13	RSVD_13_14	R	These bits (RSVD_13_14) of the CQTERRI register are reserved. They always return 0.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
12:8	RESP_ERR_TASKID	R	This field captures the ID of the task which was executed on the command line when the error occurred. Value After Reset: 0x0 Exists: Always
7:6	RSVD_6_7	R	These bits (RSVD_6_7) of the CQTERRI register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
5:0	RESP_ERR_CMD_INDX	R	This field captures the index of the command that was executed on the command line when the error occurred. Value After Reset: 0x0 Exists: Always

3.6.3.21 CQCRI

- Name: CQ Command Response Index
- Description: This register stores the index of the last received command response.
- Size: 32 bits
- Offset: 0x180 + 0x58

Figure & Table 3-113 Fields for register: CQCRI

Bits	Name	Access	Description
31:6	RSVD_31_6	R	These bits of the CQCRI register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
5:0	CMD_RESP_INDX	R	Last Command Response index This field stores the index of the last received command response. Controller updates the value every time a command response is received. Value After Reset: 0x0 Exists: Always

3.6.3.22 CQCRA

- Name: CQ Command Response Argument Register
- Description: This register stores the argument of the last received command response.
- Size: 32 bits
- Offset: 0x180 + 0x5c

Figure & Table 3-114 Fields for register: CQCRA

Bits	Name	Access	Description
31:0	CMD_RESP_ARG	R	Last Command Response argument This field stores the argument of the last received command response. Controller updates the value every time a command response is received. Value After Reset: 0x0 Exists: Always

3.6.4 DWC_mshc_map/DWC_mshc_phy_block Registers

3.6.4.1 PHY_CNFG

- Name: SD/eMMC PHY General Configuration
- Description: SD/eMMC PHY general configuration register
- Size: 32 bits
- Offset: 0x300 + 0x0

Figure & Table 3-115 Fields for register: PHY_CNFG

Bits	Name	Access	Description
31:24			Reserved Field: Yes
23:20	PAD_SN	R/W	NMOS TX drive strength control. Common configuration for all SD/eMMC PADs. Value After Reset: 0x0 Exists: Always
19:16	PAD_SP	R/W	PMOS TX drive strength control. Common configuration for all SD/eMMC PADs. Value After Reset: 0x0 Exists: Always
15:2			Reserved Field: Yes
1	PHY_PWRGOOD	R	Phy's Power Good status is captured here. Ensure this is 1 before stating transactions. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always Reset Mask: 0x0
0	PHY_RSTN	R/W	Active-Low reset control for PHY, write 0 to reset PHY, Write 1 to deassert reset. Value After Reset: 0x0 Exists: Always

3.6.4.2 CMDPAD_CNFG

- Name: SD/eMMC PHY CMD/RESP PAD Setting
- Description: SD/eMMC PHY's command/response PAD settings are controlled here.
- Size: 16 bits
- Offset: 0x300 + 0x4

Figure & Table 3-116 Fields for register: CMDPAD_CNFG

Bits	Name	Access	Description
15:13			Reserved Field: Yes
12:9	TXSLEW_CTRL_N	R/W	Slew control for N-Type CMD PAD's TX Value After Reset: 0x2 Exists: Always
8:5	TXSLEW_CTRL_P	R/W	Slew control for P-Type CMD PAD's TX Value After Reset: 0x2 Exists: Always
4:3	WEAKPULL_EN	R/W	Pull-up/Pull-down enable control for CMD PAD Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled. ■ 0x1 (PULLUP): Weak pull up enabled. ■ 0x2 (PULLDOWN): Weak pull down enabled. ■ 0x3 (ILLEGAL): Should not be used. Value After Reset: 0x0 Exists: Always
2:0	RXSEL	R/W	Receiver type select for PAD. Controls the RXSEL value of SD/eMMC PHY CMD PAD. Value After Reset: 0x0 Exists: Always

3.6.4.3 DATPAD_CNFG

- Name: SD/eMMC PHY Data PAD Setting
- Description: SD/eMMC PHY's data PAD settings are controlled here. Common settings for all data PADs
- Size: 16 bits
- Offset: 0x300 + 0x6

Figure & Table 3-117 Fields for register: DATPAD_CNFG

Bits	Name	Access	Description
15:13			Reserved Field: Yes
12:9	TXSLEW_CTRL_N	R/W	Slew control for N-Type DATA PAD's TX Value After Reset: 0x2 Exists: Always
8:5	TXSLEW_CTRL_P	R/W	Slew control for P-Type DATA PAD's TX Value After Reset: 0x2 Exists: Always
4:3	WEAKPULL_EN	R/W	Pull-up/Pull-down enable control for DATA PADs Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled. ■ 0x1 (PULLUP): Weak pull up enabled. ■ 0x2 (PULLDOWN): Weak pull down enabled. ■ 0x3 (ILLEGAL): Should not be used. Value After Reset: 0x0 Exists: Always
2:0	RXSEL	R/W	Receiver type select for PAD. Controls the RXSEL value of SD/eMMC PHY data PADs. Value After Reset: 0x0 Exists: Always

3.6.4.4 CLKPAD_CNFG

- Name: SD/eMMC PHY Clock PAD Setting
- Description: SD/eMMC PHY's clock PAD settings are controlled here.
- Size: 16 bits
- Offset: 0x300 + 0x8

Figure & Table 3-118 Fields for register: CLKPAD_CNFG

Bits	Name	Access	Description
15:13			Reserved Field: Yes
12:9	TXSLEW_CTRL_N	R/W	Slew control for N-Type CLK PAD's TX Value After Reset: 0x2 Exists: Always
8:5	TXSLEW_CTRL_P	R/W	Slew control for P-Type CLK PAD's TX Value After Reset: 0x2 Exists: Always
4:3	WEAKPULL_EN	R/W	Pull-up/Pull-down enable control for clock PAD Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled. ■ 0x1 (PULLUP): Weak pull up enabled. ■ 0x2 (PULLDOWN): Weak pull down enabled. ■ 0x3 (ILLEGAL): Should not be used. Value After Reset: 0x0 Exists: Always
2:0	RXSEL	R/W	Receiver type select for PAD. Controls the RXSEL value of SD/eMMC PHY clock PAD. Value After Reset: 0x0 Exists: Always

3.6.4.5 STBPAD_CNFG

- Name: SD/eMMC PHY Strobe PAD Setting
- Description: SD/eMMC PHY's strobe PAD settings are controlled here.
- Size: 16 bits
- Offset: 0x300 + 0xA

Figure & Table 3-119 Fields for register: STBPAD_CNFG

Bits	Name	Access	Description
15:13			Reserved Field: Yes
12:9	TXSLEW_CTRL_N	R/W	Slew control for N-Type Strobe PAD's TX Value After Reset: 0x2 Exists: Always
8:5	TXSLEW_CTRL_P	R/W	Slew control for P-Type Strobe PAD's TX

Bits	Name	Access	Description
			Value After Reset: 0x2 Exists: Always
4:3	WEAKPULL_EN	R/W	Pull-up/Pull-down enable control for strobe PAD Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled. ■ 0x1 (PULLUP): Weak pull up enabled. ■ 0x2 (PULLDOWN): Weak pull down enabled. ■ 0x3 (ILLEGAL): Should not be used. Value After Reset: 0x0 Exists: Always
2:0	RXSEL	R/W	Receiver type select for PAD. Controls the RXSEL value of SD/eMMC PHY strobe PAD. Value After Reset: 0x0 Exists: Always

3.6.4.6 RSTNPAD_CNFG

- Name: SD/eMMC PHY RSTN PAD Setting
- Description: SD/eMMC PHY's RSTN PAD settings are controlled here.
- Size: 16 bits
- Offset: 0x300 + 0xC

Figure & Table 3-120 Fields for register: RSTNPAD_CNFG

Bits	Name	Access	Description
15:13			Reserved Field: Yes
12:9	TXSLEW_CTRL_N	R/W	Slew control for N-Type RST_N PAD's TX Value After Reset: 0x2 Exists: Always
8:5	TXSLEW_CTRL_P	R/W	Slew control for P-Type RST_N PAD's TX Value After Reset: 0x2 Exists: Always
4:3	WEAKPULL_EN	R/W	Pull-up/Pull-down enable control for RST_N PAD(s) Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled. ■ 0x1 (PULLUP): Weak pull up enabled.

Bits	Name	Access	Description
			■ 0x2 (PULLDOWN): Weak pull down enabled. ■ 0x3 (ILLEGAL): Should not be used. Value After Reset: 0x0 Exists: Always
2:0	RXSEL	R/W	Receiver type select for PAD. Controls the RXSEL value of SD/eMMC PHY RST_N PAD(s). Value After Reset: 0x0 Exists: Always

3.6.4.7 PADTEST_CNFG

- Name: SD/eMMC PHY PAD Test Interface Setting
- Description: PAD test path and direction control
- Size: 16 bits
- Offset: 0x300 + 0xE

Figure & Table 3-121 Fields for register: PADTEST_CNFG

Bits	Name	Access	Description
15:10			Reserved Field: Yes
9:4	TEST_OE	R/W	Test interface OE control. Drive's PHY's itest_oe inputs. Value After Reset: 0x0 Exists: Always
3:1	RSVD_1	R	RSVD1 field is reserved. Value After Reset: 0x0 Exists: Always
0	TESTMODE_EN	R/W	Enables test mode interface for all PADS. Functional interface is disabled. Values: ■ 0x0 (PAD_FUNCMODE): PAD's functional mode I/F is active. ■ 0x1 (PAD_TESTMODE): PAD's test mode interface is active. Value After Reset: 0x0 Exists: Always

3.6.4.8 PADTEST_OUT

- Name: SD/eMMC PHY PAD Test Data out Value

- Description: PAD test path data out, drives itest_a input of SD/eMMC PHY.
- Size: 16 bits
- Offset: 0x300 + 0x10

Figure & Table 3-122 Fields for register: PADTEST_OUT

Bits	Name	Access	Description
15:6			Reserved Field: Yes
5:0	TESTDATA_OUT	R/W	Data written here is reflected on corresponding itest_a. Individual bits are mapped to corresponding itest_a inputs of the PHY. Value After Reset: 0x0 Exists: Always

3.6.4.9 PADTEST_IN

- Name: SD/eMMC PHY PAD Test Data In Value
- Description: PAD test path data in, reflects value of otest_y output of SD/eMMC PHY.
- Size: 16 bits
- Offset: 0x300 + 0x12

Figure & Table 3-123 Fields for register: PADTEST_IN

Bits	Name	Access	Description
15:6			Reserved Field: Yes
5:0	TESTDATA_IN	R	Individual bits here capture data available on corresponding otest_y output. Should be used for DC test and low frequency data patterns. Value After Reset: 0x0 Exists: Always

3.6.4.10 PRBS_CNFG

- Name: Controller PRBS Configuration Register
- Description: Register to configure PRBS engine
- Size: 16 bits
- Offset: 0x300 + 0x18

Figure & Table 3-124 Fields for register: PRBS_CNFG

Bits	Name	Access	Description
15:0	INIT_SEED	R/W	Value programmed here is used as SEED for PRBS engine. Value After Reset: 0xffff Exists: Always

3.6.4.11 PHYLPBK_CNFG

- Name: Loopback Configuration Register
- Description: Register to setup loopback mode
- Size: 8 bits
- Offset: 0x300 + 0x1A

Figure & Table 3-125 Fields for register: PHYLPBK_CNFG

Bits	Name	Access	Description
7:2			Reserved Field: Yes
1	OUT_EN_PHYLPBK_MODE	R/W	CMD/DATA output enable in PHY loopback mode Values: ■ 0x1 (OUTPUT_EN): CMD/DATA output is enabled in PHY loopback mode. ■ 0x0 (OUTPUT_DIS): CMD/DATA output is disabled in PHY loopback mode. Value After Reset: 0x0 Exists: Always
0	PHYLPBK_EN	R/W	PHY local loopback mode enable Values: ■ 0x1 (PHYLPBK_MODE_EN): Controller is now in PHY loopback mode. ■ 0x0 (PHYLPBK_MODE_DIS): Controller is not in PHY loopback mode. Value After Reset: 0x0 Exists: Always

3.6.4.12 COMMDL_CNFG

- Name: Common DelayLine Configuration Settings Register
- Description: Configuration register to settings common to all DelayLines used in PHY
- Size: 8 bits
- Offset: 0x300 + 0x1C

Figure & Table 3-126 Fields for register: COMMDL_CNFG

Bits	Name	Access	Description
7:2			Reserved Field: Yes
1	DLOUT_EN	R/W	When '1' DL outputs can be sampled on PADs. Drives idlout_en for all PADs. Values:

Bits	Name	Access	Description
			■ 0x1 (DL_OUTEN): DelayLine outputs on PAD enabled. ■ 0x0 (DL_OUTDIS): DelayLine outputs on PAD disabled. Value After Reset: 0x0 Exists: Always
0	DLSTEP_SEL	R/W	DelayLine's per step delay selection. Drives PHY's idl_step input. Value After Reset: 0x0 Exists: Always

3.6.4.13 SDCLKDL_CNFG

- Name: SD/eMMC DelayLine Settings
- Description: Settings for SD/eMMC clock DelayLine
- Size: 8 bits
- Offset: 0x300 + 0x1D

Figure & Table 3-127 Fields for register: SDCLKDL_CNFG

Bits	Name	Access	Description
7:5			Reserved Field: Yes
4	UPDATE_DC	R/W	Prepares DealyLine for code update when '1'. It's recommended that this bit is 1 when SDCLKDL_DC is being written. Ensure this is '0' when not updating code. Note: Turn off card clock using CLK_CTRL_R.SD_CLK_EN before programing this field. Values: ■ 0x1 (BYPASSMODE): Output of DelayLine is DelayLine output active. ■ 0x0 (DLMODE): DelayLine output is enabled. Value After Reset: 0x0 Exists: Always
3:2	INPSEL_CNFG	R/W	Drives SD/eMMC clock DelayLine's configuration input. Value here selects the input source to DelayLine. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
1	BYPASS_EN	R/W	Drives SD/eMMC clock DelayLine's bypassen input. Values: ■ 0x1 (BYPASSMODE): Delay line is bypass mode. ■ 0x0 (DLMODE): Delay line active mode Value After Reset: 0x0 Exists: Always
0	EXTDLY_EN	R/W	Drives SD/eMMC clock DelayLine's extdlyen input. Values: ■ 0x1 (EXTDL_MODE): Delay line works with extended delay range setting. ■ 0x0 (DEF_MODE): Delay line defaut range setting Value After Reset: 0x0 Exists: Always

3.6.4.14 SDCLKDL_DC

- Name: SD/eMMC DelayLine Delay Code Setting
- Description: SD/eMMC clock DelayLine delay code value
- Size: 8 bits
- Offset: 0x300 + 0x1E

Figure & Table 3-128 Fields for Register: SDCLKDL_DC

Bits	Name	Access	Description
7			Reserved Field: Yes
6:0	CCKDL_DC	R/W	Drives SD/eMMC CLK DelayLine's Delay Code input. Value here Selects the number of active stages in the card clock delay line. Note: Turn off card clock using CLK_CTRL_R.SD_CLK_EN before programing this field. Value After Reset: 0x0 Exists: Always

3.6.4.15 SMPLDL_CNFG

- Name: SD/eMMC cclk_rx DelayLine Settings
- Description: SD/eMMC cclk_rx DelayLine configuration settings
- Size: 8 bits
- Offset: 0x300 + 0x20

Figure & Table 3-129 Fields for register: SMPLDL_CNFG

Bits	Name	Access	Description
7:5			Reserved Field: Yes
4	INPSEL_OVERRIDE	R/W	PHY's Sampling delay line configuration is controlled by controller using sample_cclk_sel, this signal overrides sample_cclk_sel so that INPSEL_CFG field directly controls PHY's configuration input. ■ 0x0: Controller logic drives sampling delay line configuration. ■ 0x1: SMPLDL_CNFG.INPSEL_CNFG drives sampling delay line configuration. Value After Reset: 0x0 Exists: Always
3:2	INPSEL_CNFG	R/W	Drives CCLK_RX DelayLine's configuration input. Value here selects the input source to DelayLine. Value After Reset: 0x3 Exists: Always
1	BYPASS_EN	R/W	Drives CCLK_RX DelayLine's bypassen input. Values: ■ 0x1 (BYPASSMODE): DelayLine is bypass mode. ■ 0x0 (DLMODE): Delay line active mode Value After Reset: 0x1 Exists: Always
0	EXTDLY_EN	R/W	Drives CCLK_RX DelayLine's extdlyen input Values: ■ 0x1 (EXTDL_MODE): DelayLine works with extended delay range setting. ■ 0x0 (DEF_MODE): Delay line defaut range setting Value After Reset: 0x0 Exists: Always

3.6.4.16 ATDL_CNFG

- Name: SD/eMMC drift_cclk_rx DelayLine Configuration Settings
- Description: SD/eMMC drift_cclk_rx DelayLine configuration settings
- Size: 8 bits
- Offset: 0x300 + 0x21

Figure & Table 3-130 Fields for register: ATDL_CNFG

Bits	Name	Access	Description
7:4			Reserved Field: Yes
3:2	INPSEL_CNFG	R/W	Drives drift_cclk_rx DelayLine's configuration input. Value here selects the input source to DelayLine. Value After Reset: 0x0 Exists: Always
1	BYPASS_EN	R/W	Drives drift_cclk_rx DelayLine's bypassen input. Values: ■ 0x1 (BYPASSMODE): Delay line is bypass mode. ■ 0x0 (DLMODE): Delay line active mode Value After Reset: 0x0 Exists: Always
0	EXTDLY_EN	R/W	Drives drift_cclk_rx DelayLine's extdlyen input. Values: ■ 0x1 (EXTDL_MODE): Delay line works with extended delay range setting. ■ 0x0 (DEF_MODE): Delay line defaut range setting Value After Reset: 0x0 Exists: Always

3.6.4.17 DLL_CTRL

- Name: SD/eMMC PHY DLL Control Setting
- Description: SD/eMMC PHY's DLL control settings register
- Size: 8 bits
- Offset: 0x300 + 0x24

Figure & Table 3-131 Fields for register: DLL_CTRL

Bits	Name	Access	Description
7:3	RSVD_3_7	R	These bits of the register are reserved. Value After Reset: 0x0 Exists: Always
2	SLV_SWDC_UPDATE	R/W	Corresponding output drives PHY's DLL slave's dc update input. This is used to turn off Slave Delay line's output when changing its delay code using DLL_OFFST register. Values:

Bits	Name	Access	Description
			■ 0x1 (DL_OUT_OFF): Update in progress. ■ 0x0 (DL_OUT_ON): Update completed. Value After Reset: 0x0 Exists: Always
1	OFFST_EN	R/W	Enables offset mode of PHY when DLL is enabled. When DLL is disabled, this allows direct control of delay generated by DLL's slave. Values: ■ 0x1 (OFFSTEN): Offset value is valid. ■ 0x0 (OFFSTDIS): offset value is invalid. Value After Reset: 0x0 Exists: Always
0	DLL_EN	R/W	Enable's DLL when '1'. Values: ■ 0x1 (DLLENABLE): PHY DLL is enabled. ■ 0x0 (DLLDISABLE): PHY DLL is disabled. Value After Reset: 0x0 Exists: Always

3.6.4.18 DLL_CNFG1

- Name: DLL Configuration Register 1
- Description: SD/eMMC PHY DLL configuration register 1
- Size: 8 bits
- Offset: 0x300 + 0x25

Figure & Table 3-132 Fields for register: DLL_CNFG1

Bits	Name	Access	Description
7:6			Reserved Field: Yes
5:4	SLVDLY	R/W	Sets the value of DLL slave's update delay input islv_update_dly. Value After Reset: 0x0 Exists: Always
3			Reserved Field: Yes
2:0	WAITCYCLE	R/W	Sets the value of DLL's wait cycle input. Value After Reset: 0x0 Exists: Always

3.6.4.19 DLL_CNFG2

- Name: DLL Configuration Register 2
- Description: SD/eMMC PHY DLL configuration register 2
- Size: 8 bits
- Offset: 0x300 + 0x26

Figure & Table 3-133 Fields for register: DLL_CNFG2

Bits	Name	Access	Description
7			Reserved Field: Yes
6:0	JUMPSTEP	R/W	Sets the value of DLL's jump step input. Value After Reset: 0x0 Exists: Always

3.6.4.20 DLLDL_CNFG

- Name: DLL Configuration Register 2
- Description: SD/eMMC PHY DLL master & slave DL configuration settings
- Size: 8 bits
- Offset: 0x300 + 0x28

Figure & Table 3-134 Fields for register: DLLDL_CNFG

Bits	Name	Access	Description
7	SLV_BYPASS	R/W	Bypass enable control for slave DL Value After Reset: 0x0 Exists: Always
6:5	SLV_INPSEL	R/W	Clock source select for slave DL Value After Reset: 0x0 Exists: Always
4	SLV_EXTDLYEN	R/W	Enable extended delay mode for slave Value After Reset: 0x0 Exists: Always
3	MST_BYPASS	R/W	Bypass enable control for master DL Value After Reset: 0x0 Exists: Always
2:1	MST_INPSEL	R/W	Clock source select for master DL Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
0	MST_EXTDLYEN	R/W	Enable extended delay mode for master Value After Reset: 0x0 Exists: Always

3.6.4.21 DLL_OFFST

- Name: DLL Offset Setting Register
- Description: SD/eMMC PHY DLL offset value settings
- Size: 8 bits
- Offset: 0x300 + 0x29

Figure & Table 3-135 Fields for register: DLL_OFFST

Bits	Name	Access	Description
7			Reserved Field: Yes
6:0	OFFST	R/W	Sets the value of DLL's offset input. Value After Reset: 0x0 Exists: Always

3.6.4.22 DLLMST_TSTDC

- Name: DLL Master Test Code Setting Register
- Description: SD/eMMC PHY DLL master testing delay code register
- Size: 8 bits
- Offset: 0x300 + 0x2A

Figure & Table 3-136 Fields for register: DLLMST_TSTDC

Bits	Name	Access	Description
7			Reserved Field: Yes
6:0	MSTTST_DC	R/W	Sets the value of DLL's master test code input when DLL is disabled. Value After Reset: 0x0 Exists: Always

3.6.4.23 DLLLBT_CNFG

- Name: DLL LBT Setting Register
- Description: SD/eMMC PHY DLL low bandwidth timer configuration register
- Size: 16 bits
- Offset: 0x300 + 0x2C

Figure & Table 3-137 Fields for register: DLLLBT_CNFG

Bits	Name	Access	Description
15:0	LBT_LOADVAL	R/W	Sets the value of DLL's olbt_loadval input. Controls the lbt timer's timeout value at which DLL runs a revalidation cycle. Value After Reset: 0x0 Exists: Always

3.6.4.24 DLL_STATUS

- Name: DLL Status Register
- Description: SD/eMMC PHY DLL status register
- Size: 8 bits
- Offset: 0x300 + 0x2E

Figure & Table 3-138 Fields for register: DLL_STATUS

Bits	Name	Access	Description
7:2			Reserved Field: Yes
1	ERROR_STS	R	Captures the value of DLL's lock error status information. Value is valid only when LOCK_STS is set. ■ IF LOCK_STS =1 and ERR_STS = 0 then DLL is locked and no errors are generated. ■ IF LOCK_STS =1 and ERR_STS = 1 then DLL is locked to default and has errors. Transactions at this phase can fail. Values: ■ 0x1 (DLL_ERROR): DLL is locked and ready. ■ 0x0 (DLL_LOCK_OKAY): DLL has not locked. Value After Reset: 0x0 Exists: Always
0	LOCK_STS	R	Captures the value of DLL's lock status information. Values: ■ 0x1 (DLL_IS_LOCKED): DLL is locked and ready. ■ 0x0 (DLL_NOT_LOCKED): DLL has not locked. Value After Reset: 0x0 Exists: Always

3.6.4.25 DLLDBG_MLKDC

- Name: DLL Master Lock Code Debug Register

- Description: SD/eMMC PHY DLL's master lock code status
- Size: 8 bits
- Offset: 0x300 + 0x30

Figure & Table 3-139 Fields for register: DLLDBG_MLKDC

Bits	Name	Access	Description
7			Reserved Field: Yes
6:0	MSTLKDC	R	Captures the value Delay Code to which DLL's master has locked. Value After Reset: 0x0 Exists: Always

3.6.4.26 DLLDBG_SLKDC

- Name: DLL Master Slave Code Debug Register
- Description: SD/eMMC PHY DLL's slave lock code status
- Size: 8 bits
- Offset: 0x300 + 0x32

Figure & Table 3-140 Fields for register: DLLDBG_SLKDC

Bits	Name	Access	Description
7			Reserved Field: Yes
6:0	SLVLKDC	R	Captures the value Delay Code to which DLL's slave has locked. Value After Reset: 0x0 Exists: Always

3.6.5 DWC_mshc_map/DWC_mshc_vendor1_block Registers

3.6.5.1 MSHC_VER_ID_R

- Name: MSHC Version
- Description: This register reflects the current release number of DWC_mshc.
- Size: 32 bits
- Offset: 0x500

Figure & Table 3-141 Fields for register: MSHC_VER_ID_R

Bits	Name	Access	Description
31:0	MSHC_VER_ID	R	Current release number This field indicates the Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite current release number that is read by an application.

Bits	Name	Access	Description
			An application reading this register in conjunction with the MSHC_VER_TYPE_R register, gathers details of the current release. Value After Reset: "SNPS_RSVDPARAM_5" Exists: Always

3.6.5.2 MSHC_VER_TYPE_R

- Name: MSHC Version Type
- Description: This register reflects the current release type of DWC_mshc.
- Size: 32 bits
- Offset: 0x500 + 0x4

Figure & Table 3-142 Fields for Register: MSHC_VER_TYPE_R

Bits	Name	Access	Description
31:0	MSHC_VER_TYPE	R	Current release type This field indicates the Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite current release type that is read by an application. An application reading this register in conjunction with the MSHC_VER_ID_R register, gathers details of the current release. Value After Reset: "SNPS_RSVDPARAM_6" Exists: Always

3.6.5.3 MSHC_CTRL_R

- Name: MSHC Control Register
- Description: This register is used to control the operation of MSHC host controller.
- Size: 8 bits
- Offset: 0x500 + 0x8

Figure & Table 3-143 Fields for Register: MSHC_CTRL_R

Bits	Name	Access	Description
7	Reserved	R	
6	Reserved	R	
5			Reserved Field: Yes
4	SW_CG_DIS	R/W	Internal clock gating disable control This bit must be used to disable module's internal clock gating when required. When disabled, clocks are not

Bits	Name	Access	Description
			gated. Clocks to the core (except hclk) must be stopped when programming this bit. Values: ■ 0x0 (ENABLE): Internal clock gates are active and clock gating is controlled internally. ■ 0x1 (DISABLE): Internal clock gating is disabled, clocks are not gated internally. Value After Reset: 0x0 Exists: Yes
3:1	RSVD1	R	These bits (RSVD1) of the MSHC_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
0	CMD_CONFLICT_CHECK	R/W	Command conflict check This bit enables command conflict check. Note: DWC_mshc controller monitors the CMD line whenever a command is issued and checks whether the value driven on sd_cmd_out matches the value on sd_cmd_in at next subsequent edge of cclk_tx to determine command conflict error. This bit is cleared only if the feedback delay (including IO PAD delay) is more than (t_card_clk_period - t_setup), where t_setup is the setup time of a flop in DWC_mshc. The I/O PAD delay is consistent across CMD and DATA lines, and it is within the value: (2*t_card_clk_period - t_setup). Values: ■ 0x0 (DISABLE_CMD_CONFLICT_CHK): Disable command conflict check. ■ 0x1 (CMD_CONFLICT_CHK_LAT1): Check for command conflict after 1 card clock cycle. Value After Reset: 0x1 Exists: Yes

3.6.5.4 MBIU_CTRL_R

- Name: MBIU Control Register
- Description: This register is used to select the valid burst types that the AHB master bus interface can generate. When more than one bit is set the master selects the burst it prefers among those that are enabled in this register.
- Size: 8 bits

- Offset: 0x500 + 0x10

Figure & Table 3-144 Fields for register: MBIU_CTRL_R

Bits	Name	Access	Description
7:4	RSVD	R	Reserved field Value After Reset: 0x0 Exists: Always
3	BURST_INCR16_EN	R/W	INCR16 Burst Controls generation of INCR16 transfers on master interface. Values: ■ 0x0 (FALSE): AHB INCR16 burst type is not generated on master I/F. ■ 0x1 (TRUE): AHB INCR16 burst type can be generated on master I/F. Value After Reset: 1 Exists: Yes
2	BURST_INCR8_EN	R/W	INCR8 Burst Controls generation of INCR8 transfers on master interface. Values: ■ 0x0 (FALSE): AHB INCR8 burst type is not generated on master I/F. ■ 0x1 (TRUE): AHB INCR8 burst type can be generated on master I/F. Value After Reset: 1 Exists: Yes
1	BURST_INCR4_EN	R/W	INCR4 Burst Controls generation of INCR4 transfers on master interface. Values: ■ 0x0 (FALSE): AHB INCR4 burst type is not generated on master I/F. ■ 0x1 (TRUE): AHB INCR4 burst type can be generated on master I/F. Value After Reset: 1 Exists: Yes
0	UNDEFL_INCR_EN	R/W	Undefined INCR Burst Controls generation of undefined length INCR transfer

Bits	Name	Access	Description
			on master interface. Values: ■ 0x0 (FALSE): Undefined INCR type burst is the least preferred burst on AHB master I/F. ■ 0x1 (TRUE): Undefined INCR type burst is the most preferred burst on AHB master I/F. Value After Reset: DWC_MSHC_MBIU_UNDEFLBURSTEN Exists: Yes

3.6.5.5 EMMC_CTRL_R

- Name: eMMC Control Register
- Description: This register is used to control the eMMC operation.
- Size: 16 bits
- Offset: 0x500 + 0x2c

Figure & Table 3-145 Fields for register: EMMC_CTRL_R

Bits	Name	Access	Description
15:11	RSVD	R	These bits (RSVD) of the EMMC_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
10	CQE_PREFETCH_DISABLE	R/W	Enable or Disable CQE's PREFETCH feature. This field allows software to disable CQE's data prefetch feature when set to 1. Values: ■ 0x0 (PREFETCH_ENABLE): CQE can prefetch data for sucessive WRITE transfers and pipeline sucessive READ transfers. ■ 0x1 (PREFETCH_DISABLE): Prefetch for WRITE and Pipeline for READ are disabled. Value After Reset: 0x0 Exists: Yes
9	CQE_ALGO_SEL	R/W	Scheduler algorithm selected for execution This bit selects the algorithm used for selecting one of the many ready tasks for execution. Values: ■ 0x0 (PRI_REORDER_PLUS_FCFS): Priority based

Bits	Name	Access	Description
			reordering with FCFS to resolve equal priority tasks ■ 0x1 (FCFS_ONLY): First come First serve, in the order of DBR rings Value After Reset: 0x0 Exists: Yes
8	ENH_STROBE_ENABLE	R/W	Enhanced Strobe Enable This bit instructs DWC_mshc to sample the CMD line using data strobe for HS400 mode. Values: ■ 0x1 (ENH_STB_FOR_CMD): CMD line is sampled using data strobe for HS400 mode. ■ 0x0 (NO_STB_FOR_CMD): CMD line is sampled using cclk_rx for HS400 mode. Value After Reset: 0x0 Exists: Yes
7:4			Reserved Field: Yes
3	EMMC_RST_N_OE	R/W	Output Enable Control for eMMC Device Reset Signal PAD Control This field drived sd_rst_n_oe output of DWC_mshc. Values: ■ 0x1 (ENABLE): sd_rst_n_oe is 1. ■ 0x0 (DISABLE): sd_rst_n_oe is 0. Value After Reset: 0x1 Exists: Yes
2	EMMC_RST_N	R/W	eMMC Device Reset Signal Control This register field controls the sd_rst_n output of DWC_mshc. Values: ■ 0x1 (RST_DEASSERT): Reset to eMMC device is deasserted. ■ 0x0 (RST_ASSERT): Reset to eMMC device is asserted (active low). Value After Reset: 0x1 Exists: Yes
1	DISABLE_DATA_CRC_CHK	R/W	Disable Data CRC Check This bit controls masking of CRC16 error for Card Write in eMMC mode. This is useful in bus testing (CMD19) for

Bits	Name	Access	Description
			an eMMC device. In bus testing, an eMMC card does not send CRC status for a block, which may generate CRC error. This CRC error can be masked using this bit during bus testing. Values: ■ 0x1 (DISABLE): Data CRC check is disabled. ■ 0x0 (ENABLE): Data CRC check is enabled. Value After Reset: 0x0 Exists: Yes
0	CARD_IS_EMMC	R/W	eMMC Card Present This bit indicates the type of card connected. An application program this bit based on the card connected to MSHC. Values: ■ .0x1 (EMMC_CARD): Card connected to MSHC is an eMMC card. ■ .0x0 (NON_EMMC_CARD): Card connected to MSHC is a non-eMMC card. Value After Reset: 0x0 Exists: Yes

3.6.5.6 BOOT_CTRL_R

- Name: eMMC Boot Control Register
- Description: This register is used to control the eMMC boot operation.
- Size: 16 bits
- Offset: 0x500 + 0x2e

Figure & Table 3-146 Fields for register: BOOT_CTRL_R

Bits	Name	Access	Description
15:12	BOOT_TOUT_CNT	R/W	Boot Ack Timeout Counter Value This value determines the interval by which boot ack timeout (50 ms) is detected when boot ack is expected during boot operation. ■ 0xF: Reserved ■ 0xE: TMCLK x 2^27 ■ .. - ..... ■ 0x1: TMCLK x 2^14 ■ 0x0: TMCLK x 2^13

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
11:9	RSVD_11_9	R	These bits (RSVD_11_9) of the BOOT_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
8	BOOT_ACK_ENABLE	R/W	Boot Acknowledge Enable When this bit set, DWC_mshc checks for boot acknowledge start pattern of 0-1-0 during boot operation. This bit is applicable for both mandatory and alternate boot mode. Values: ■ 0x1 (TRUE): Boot Ack enable. ■ 0x0 (FALSE): Boot Ack disable. Value After Reset: 0x0 Exists: Always
7	VALIDATE_BOOT	W	Validate Mandatory Boot Enable This bit is used to validate the MAN_BOOT_EN bit. Values: ■ 0x1 (TRUE): Validate Mandatory Boot Enable bit. ■ 0x0 (FALSE): Ignore Mandatory Boot Enable bit. Value After Reset: 0x0 Exists: Always
6:1	RSVD_6_1	R	These bits (RSVD_6_1) of the BOOT_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
0	MAN_BOOT_EN	R/W	Mandatory Boot Enable This bit is used to initiate the mandatory boot operation. The application sets this bit along with VALIDATE_BOOT bit. Writing 0 is ignored. The DWC_mshc clears this bit after the boot transfer is completed or terminated. Values: ■ 0x1 (MAN_BOOT_EN): Mandatory boot enabled. ■ 0x0 (MAN_BOOT_DIS): Mandatory boot disabled. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always Testable: Read-only

3.6.5.7 AT_CTRL_R

- Name: Tuning and Auto-tuning Control Register
- Description: This register controls some aspects of tuning and auto-tuning features. Do not program this register when HOST_CTRL2_R.SAMPLE_CLK_SEL is 1.
- Size: 32 bits
- Offset: 0x500 + 0x40

Figure & Table 3-147 Fields for register: AT_CTRL_R

Bits	Name	Access	Description
30:24	SWIN_TH_VAL	R/W	Sampling Window Threshold Value Setting The maximum value that can be set here depends on the length of delayline used for tuning. A delayLine with 128 taps can use values from 0x0 to 0x7F. This field is valid only when SWIN_TH_EN is 1. Should be programmed only when SAMPLE_CLK_SEL is 0. ■ 0x0: Threshold values is 0x1, windows of length 1 tap and above can be selected as sampling window. ■ 0x1: Threshold values is 0x1, windows of length 2 taps and above can be selected as sampling window. ■ 0x2: Threshold values is 0x1, windows of length 3 taps and above can be selected as sampling window. ■ ..... ■ 0x7F: Threshold values is 0x1, windows of length 128 taps and above can be selected as sampling window. Value After Reset: 3
23:21			Reserved Field: Yes
20:19	POST_CHANGE_DLY	R/W	Time taken for phase switching and stable clock output Specifies the maximum time (in terms of cclk cycles) that the delay line can take to switch its output phase after a change in tuning_cclk_sel or autotuning_cclk_sel. Values:

Bits	Name	Access	Description
			■ 0x0 (LATENCY_LT_1): Less than 1-cycle latency ■ 0x1 (LATENCY_LT_2): Less than 2-cycle latency ■ 0x2 (LATENCY_LT_3): Less than 3-cycle latency ■ 0x3 (LATENCY_LT_4): Less than 4-cycle latency Value After Reset: 0x0
18:17	PRE_CHANGE_DLY	R/W	Maximum latency specification between cclk_tx and cclk_rx. Values: ■ 0x0 (LATENCY_LT_1): Less than 1-cycle latency ■ 0x1 (LATENCY_LT_2): Less than 2-cycle latency ■ 0x2 (LATENCY_LT_3): Less than 3-cycle latency ■ 0x3 (LATENCY_LT_4): Less than 4-cycle latency Value After Reset: 0x0
16	TUNE_CLK_STOP_EN	R/W	Clock stopping control for tuning and auto-tuning circuit. When enabled, clock gate control output of DWC_mshc (clk2card_on) is pulled low before changing phase select codes on tuning_cclk_sel and autotuning_cclk_sel. This effectively stops the device/card clock, cclk_rx and also drift_cclk_rx. Changing phase code when clocks are stopped ensures glitch free phase switching. Set this bit to 0 if the PHY or delayline can guarantee glitch free switching. Values: ■ 0x1 (ENABLE_CLK_STOPPING): Clocks stopped during phase code change. ■ 0x0 (DISABLE_CLK_STOPPING): Clocks not stopped. PHY ensures glitch free phase switching. Value After Reset: 0x0
15:12	RSDV3	R	These bits (RSVD3) of the AT_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
11:8	WIN_EDGE_SEL	R/W	This field sets the phase for left and right edges for drift monitoring. [Left edge offset + Right edge offset] must not be less than total taps of delay line. ■ 0x0: User selection disabled. Tuning calculated

Bits	Name	Access	Description
			edges are used. ■ 0x1: Right edge phase is center + 2 stages, Left edge phase is center - 2 stages. ■ 0x2: Right edge phase is center + 3 stages, Left edge phase is center - 3 stages. ■ ... ■ 0xF: Right edge phase is center + 16 stages, Left edge phase is center - 16 stages. Value After Reset: 0x0 Exists: Yes
7:5	RSDV2	R	These bits (RSVD2) of the AT_CTRL_R register are reserved. They always return 0. Value After Reset: 0x0 Exists: Always
4	SW_TUNE_EN	R/W	This fields enables software-managed tuning flow. Values: ■ 0x1 (SW_TUNING_ENABLE): Software-managed tuning enabled. AT_STAT_R. CENTER_PH_CODE Field is now writable. ■ 0x0 (SW_TUNING_DISABLE): Software-managed tuning disabled. Value After Reset: 0x0 Exists: Always
3	RPT_TUNE_ERR	R/W	Framing errors are not generated when executing tuning. This debug bit allows users to report these errors. Values: ■ 0x1 (DEBUG_ERRORS): Debug mode for reporting framing errors ■ 0x0 (ERRORS_DISABLED): Default mode whereas per SD-HCI no errors are reported. Value After Reset: 0x0 Exists: Always
2	SWIN_TH_EN	R/W	Sampling window threshold enable Selects the tuning mode field should be programmed only when SAMPLE_CLK_SEL is 0. Values: ■ 0x1 (THRESHOLD_MODE): Tuning engine selects the

Bits	Name	Access	Description
			first complete sampling window that meets the threshold set by SWIN_TH_VAL field. ■ 0x0 (LARGEST_WIN_MODE): Tuning engine sweeps all taps and settles at the largest window. Value After Reset: 1
1	CI_SEL	R/W	Selects the interval when the corrected center phase select code can be driven on tuning_cclk_sel output. Values: ■ 0x0 (WHEN_IN_BLK_GAP): Driven in block gap interval. ■ 0x1 (WHEN_IN_IDLE): Driven at the end of the transfer. Value After Reset: 0x0 Exists: Yes
0	AT_EN	R/W	Setting this bit enables auto-tuning engine. This bit is enabled by default when core is configured with mode3 retuning support. Clear this bit to 0 when core is configured to have mode3 re-tuning but software wishes to disable mode3 retuning. This field should be programmed only when CLK_CTRL_R.SD_CLK_EN is 0. Values: ■ 0x1 (AT_ENABLE): Auto-tuning is enabled. ■ 0x0 (AT_DISABLE): Auto-tuning is disabled. Value After Reset: 1 Exists: Yes

3.6.5.8 AT_STAT_R

- Name: Tuning and Auto-tuning Status Register
- Description: Register to read the Center, Left and Right codes used by tuning and auto-tuning engines. Center code field is also used for software managed tuning.
- Size: 32 bits
- Offset: 0x500 + 0x44

Figure & Table 3-148 Fields for register: AT_STAT_R

Bits	Name	Access	Description
31:24	RSDV1	R	These bits of the AT_STAT_R register are reserved. They always return 0.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
23:16	L_EDGE_PH_CODE	R	Left Edge Phase code. Reading this field returns the phase code value used by Auto-tuning engine to sample data on Left edge of sampling window. Value After Reset: 0x0 Exists: Always
15:8	R_EDGE_PH_CODE	R	Right Edge Phase code. Reading this field returns the phase code value used by auto-tuning engine to sample data on Right edge of sampling window. Value After Reset: 0x0 Exists: Always
7:0	CENTER_PH_CODE	R/W	Centered Phase code. Reading this field returns the current value on tuning_cclk_sel output. Setting AT_CTRL_R. SW_TUNE_EN enables software to write to this field and its contents are reflected on tuning_cclk_sel. Value After Reset: 6 Exists: Always Testable: Read-only

4 QSPI

4.1 Overview

The DW_apb_ssi is a configurable, synthesizable, and programmable component that is a full-duplex master serial interface. The host processor accesses data, control, and status information on the DW_apb_ssi through the APB interface. The DW_apb_ssi may also interface with a DMA controller using an optional set of DMA signals, which can be selected at configuration time.
As described in more detail later, the DW_apb_ssi can be configured as a serial master. The DW_apb_ssi can connect to any serial-slave peripheral device using Motorola Serial Peripheral Interface (SPI).

4.2 Main Features

DW_apb_ssi has the following features:

- APB interface
- APB3 protocol support
- Scalable APB data bus width – Supports APB data bus widths of 32 bits.
- Serial-master – Enables serial communication serial-slave peripheral devices.
- Configurable and programmable Quad SPI (QSPI) support in master mode – Configure DW_apb_ssi to support QSPI mode, and then program DW_apb_ssi for QSPI transfers.
- Data mask support - Enables the DW_apb_ssi to selectively update the bytes in the device. This feature is applicable only in enhanced SPI modes.
- eXecute In Place (XIP) support - Enables the DW_apb_ssi master to behave as a memory mapped I/O and fetches the data from the device based on the APB read request. This feature is applicable only in enhanced SPI modes.
- DMA controller interface – Enables the DW_apb_ssi to interface to a DMA controller over the bus using a handshaking interface for transfer requests.
- Independent masking of interrupts – Master collision, transmit FIFO overflow, transmit FIFO empty, receive FIFO full, receive FIFO underflow, and receive FIFO overflow interrupts can all be masked independently.
- Multi-master contention detection – Informs the processor of multiple serial master accesses on the serial bus.
- Programmable delay on the sample time of the received serial data bit (rxd), when configured in master mode; enables programmable control of routing delays resulting in higher serial data-bit rates.
- Number of slave select outputs – When operating as a serial master, 4 serial slave-select output signals can be generated.
- Combined interrupt line – Bring one combined interrupt line from the DW_apb_ssi to the interrupt controller.

- Programmable features:
  - Serial interface operation - Choice of Motorola SPI.
  - Clock bit-rate - Dynamic control of the serial bit rate of the data transfer; used in only serial-master mode of operation.
  - Data item size (4 to 32 bits) - Item size of each data transfer under the control of the programmer.
- Configurable features:
  - Interrupt polarity - This configuration option selects the active level of the output interrupt line.
  - Serial clock polarity - This configuration option selects the serial-clock polarity of the SPI format directly after reset.
  - Serial clock phase - This configuration option selects the serial-clock phase of the SPI format directly after reset.

4.3 Interface

Pin Name	Direction	Width	Description
QSPI0_CSN0	O	1	Slave select output of QSPI0. The active polarity is logic 0.
QSPI0_CSN1	O	1	Slave select output of QSPI0. The active polarity is logic 0.
QSPI0_D0_MOSI	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device.
QSPI0_D1_MISO	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device.
QSPI0_D2_WP	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device. The Write Protect (WP) pin can be used to prevent the Status Register of the device from being written.
QSPI0_D3_HOLD	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device. The HOLD pin allows the device to be paused while it is actively selected.
QSPI0_SCLK	O	1	The SPI Serial Clock Output pin provides the timing for serial input and output operations.
QSPI1_CSN0	O	1	Slave select output of QSPI1. The active polarity is logic 0.
QSPI1_D0_MOSI	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device.
QSPI1_D1_MISO	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device.

Pin Name	Direction	Width	Description
QSPI1_D2_WP	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device. The WP pin can be used to prevent the Status Register of the device from being written.
QSPI1_D3_HOLD	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device. The HOLD pin allows the device to be paused while it is actively selected.
QSPI1_SCLK	O	1	The SPI Serial Clock Output pin provides the timing for serial input and output operations.
SPI_CSN	O	1	Slave select output of SPI. The active polarity is logic 0.
SPI_MISO	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device.
SPI_MOSI	IO	1	The bidirectional IO pins to serially write instructions, addresses or data to the device.
SPI_SCLK	O	1	The SPI Serial Clock Output pin provides the timing for serial input and output operations.

4.4 Function Description

4.4.1 Data Flow

In order for the DW_apb_ssi to connect to a serial-slave peripheral device, the peripheral must have a Motorola SPI. This is 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 the slave select signal or the first edge of the serial clock. The slave select line is held high when the DW_apb_ssi is idle or disabled.

You can program the FRF (frame format) bit field in the Control Register 0 (CTRLR0) to select which protocol is used.

The serial protocols supported by the DW_apb_ssi allow for serial slaves to be selected or addressed using either hardware or software. When implemented in hardware, serial slaves are selected under the control of dedicated hardware select lines. The number of select lines generated from the serial master is equal to the number of serial slaves present on the bus. The serial-master device asserts the select line of the target serial slave before data transfer begins. This architecture is illustrated in Figure & Table 4-1.

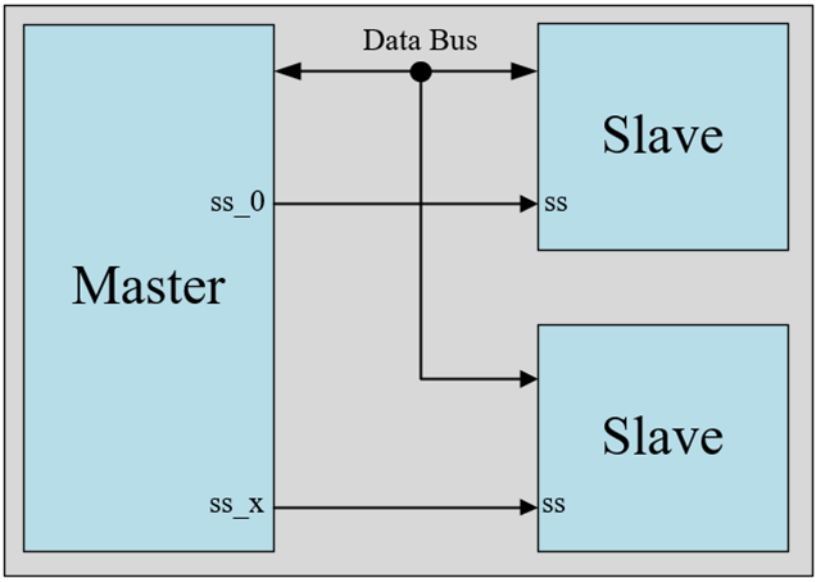

Figure & Table 4-1 Slave selection

4.4.2 Clock Ratios

When DW_apb_ssi is configured as a master device, the maximum frequency of the bit-rate clock (sclk_out) is one-half the frequency of ssi_clk. This allows the shift control logic to capture data on one clock edge of sclk_out and propagate data on the opposite edge. The sclk_out line toggles only when an active transfer is in progress. At all other times it is held in an inactive state.
The frequency of sclk_out can be derived from the following equation:
Fsclk_out = Fssi_clk/SCKDV
SCKDV is a bit field in the programmable register BAUDR, holding any even value in the range 0 to 65,534. If SCKDV is 0, then sclk_out is disabled.

4.4.3 Transmit and Receive FIFO Buffers

The transmit and receive FIFO buffers are cleared when the DW_apb_ssi is disabled (SSI_EN = 0) or when it is reset (presetn).
The transmit FIFO is loaded by APB write commands to the DW_apb_ssi data register (DR). Data are popped (removed) from the transmit FIFO by the shift control logic into the transmit shift register. The transmit FIFO generates a FIFO empty interrupt request (ssi_txe_intr) when the number of entries in the FIFO is less than or equal to the FIFO threshold value. The threshold value, set through the programmable register TXFTLR, determines the level of FIFO entries at which an interrupt is generated. The threshold value allows you to provide early indication to the processor that the transmit FIFO is nearly empty. A transmit FIFO overflow interrupt (ssi_txo_intr) is generated if you attempt to write data into an already full transmit FIFO.
Data are popped from the receive FIFO by APB read commands to the DW_apb_ssi DR. The receive FIFO is loaded from the receive shift register by the shift control logic. The receive FIFO generates a FIFO-full interrupt request (ssi_rxf_intr) when the number of entries in the FIFO is greater than or equal to the FIFO threshold value plus 1. The threshold value, set through programmable register

RXFTLR, determines the level of FIFO entries at which an interrupt is generated. The threshold value allows you to provide early indication to the processor that the receive FIFO is nearly full. A receive FIFO overrun interrupt (ssi_rxo_intr) is generated when the receive shift logic attempts to load data into a completely full receive FIFO. However, this newly received data are lost. A receive FIFO underflow interrupt (ssi_rxu_intr) is generated if you attempt to read from an empty receive FIFO. This alerts the processor that the read data are invalid.

Figure & Table 4-2 Transmit FIFO Threshold (TFT) decode values

TFT Value (Bin)	Description
0000_0000	ssi_txe_intr is asserted when 0 data entries are present in transmit FIFO.
0000_0001	ssi_txe_intr is asserted when 1 or less data entry is present in transmit FIFO.
0000_0010	ssi_txe_intr is asserted when 2 or less data entries are present in transmit FIFO.
0000_0011	ssi_txe_intr is asserted when 3 or less data entries are present in transmit FIFO.
...	...
...	...
1111_1100	ssi_txe_intr is asserted when 252 or less data entries are present in transmit FIFO.
1111_1101	ssi_txe_intr is asserted when 253 or less data entries are present in transmit FIFO.
1111_1110	ssi_txe_intr is asserted when 254 or less data entries are present in transmit FIFO.
1111_1111	ssi_txe_intr is asserted when 255 or less data entries are present in transmit FIFO.

Figure & Table 4-3 Receive FIFO Threshold (RFT) decode values

RFT Value	Description
0000_0000	ssi_rxf_intr is asserted when 1 or more data entry is present in receive FIFO.
0000_0001	ssi_rxf_intr is asserted when 2 or more data entries are present in receive FIFO.
0000_0010	ssi_rxf_intr is asserted when 3 or more data entries are present in receive FIFO.
0000_0011	ssi_rxf_intr is asserted when 4 or more data entries are present in receive FIFO.
...	...
...	...
1111_1100	ssi_rxf_intr is asserted when 253 or more data entries are present in receive FIFO.
1111_1101	ssi_rxf_intr is asserted when 254 or more data entries are present in receive FIFO.
1111_1110	ssi_rxf_intr is asserted when 255 or more data entries are present in receive FIFO.
1111_1111	ssi_rxf_intr is asserted when 256 data entries are present in receive FIFO.

4.4.4 SSI Interrupts

The DW_apb_ssi interrupts are described as follows:

- Transmit FIFO Empty Interrupt (ssi_txe_intr) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (ssi_txo_intr) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until you read the transmit FIFO overflow interrupt clear register (TXOICR).
- Receive FIFO Full Interrupt (ssi_rxf_intr) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (ssi_rxo_intr) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data are discarded. This interrupt remains set until you read the receive FIFO overflow interrupt clear register (RXOICR).
- Receive FIFO Underflow Interrupt (ssi_rxu_intr) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until you read the receive FIFO underflow interrupt clear register (RXUICR).
- Multi-Master Contention Interrupt (ssi_mst_intr) – Present only when the DW_apb_ssi component is configured as a serial-master device. The interrupt is set when another serial master on the serial bus selects the DW_apb_ssi master as a serial-slave device and is actively transferring data. This informs the processor of possible contention on the serial bus. This interrupt remains set until you read the multi-master interrupt clear register (MSTICR).
- Combined Interrupt Request (ssi_intr) – OR'ed result of all the above interrupt requests after masking. To mask this interrupt signal, you must mask all other DW_apb_ssi interrupt requests.

4.4.5 Transfer Modes

- When TMOD = 2'b00, both transmit and receive logic are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.
- When TMOD = 2'b01, the receive data are invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target

device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

- When TMOD = 2'b10, the transmit data are invalid. When configured as a slave, the transmit FIFO is never popped in Receive Only mode. The txd output remains at a constant logic level during the transmission. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.
- When TMOD = 2'b11, the transmit data is used to transmit an opcode and/or an address to the EEPROM device. Typically, this takes three data frames (8-bit opcode followed by 8-bit upper address and 8-bit lower address). During the transmission of the opcode and address, no data is captured by the receive logic (as long as the DW_apb_ssi master is transmitting data on its txd line, data on the rxd line is ignored). The DW_apb_ssi master continues to transmit data until the transmit FIFO is empty. Therefore, you should ONLY have enough data frames in the transmit FIFO to supply the opcode and address to the EEPROM. If more data frames are in the transmit FIFO than are needed, then read data is lost. When the transmit FIFO becomes empty (all control information has been sent), data on the receive line (rxd) is valid and is stored in the receive FIFO; the txd output is held at a constant logic level. The serial transfer continues until the number of data frames received by the DW_apb_ssi master matches the value of the NDF field in the CTRLR1 register + 1.

For transmit and receive transfers (transfer mode field [9:8] of the Control Register 0 = 2'b00), data transmitted from the DW_apb_ssi to the external serial device is written into the transmit FIFO. Data received from the external serial device into the DW_apb_ssi is pushed into the receive FIFO.

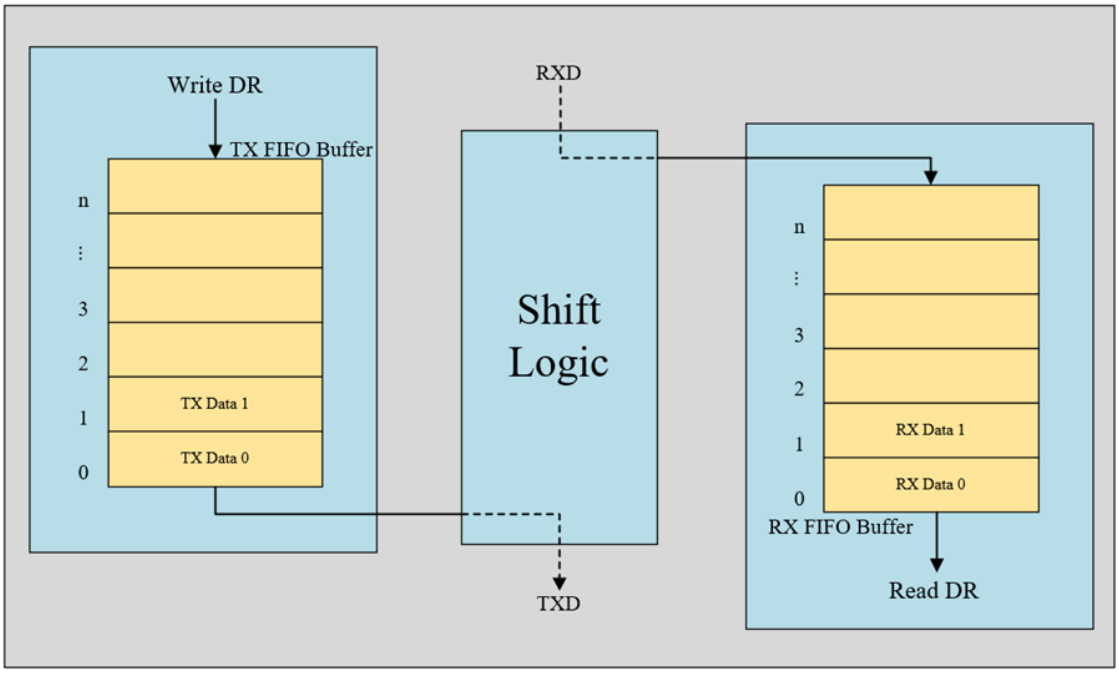

Figure & Table 4-4 FIFO status for transmit & receive SPI transfers

For transmit only transfers (transfer mode field [9:8] of the Control Register 0 = 2'b01), data transmitted from the DW_apb_ssi to the external serial device is written into the transmit FIFO. As the data received from the external serial device is deemed invalid, it is not stored in the DW_apb_ssi receive FIFO.

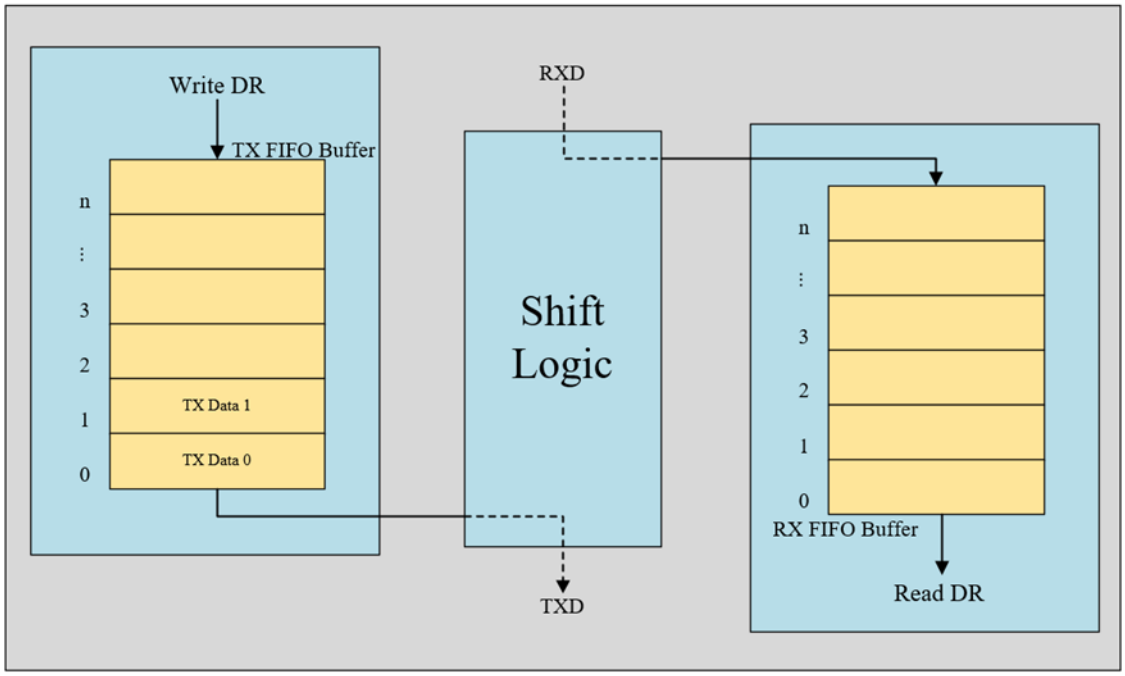

Figure & Table 4-5 FIFO status for transmit only SPI transfers

For receive only transfers (transfer mode field [9:8] of the Control Register 0 = 2'b10), data transmitted from the DW_apb_ssi to the external serial device is invalid, so a single dummy word is written into the transmit FIFO to begin the serial transfer. The txd output from the DW_apb_ssi is held at a constant logic level for the duration of the serial transfer. Data received from the external serial device into the DW_apb_ssi is pushed into the receive FIFO.

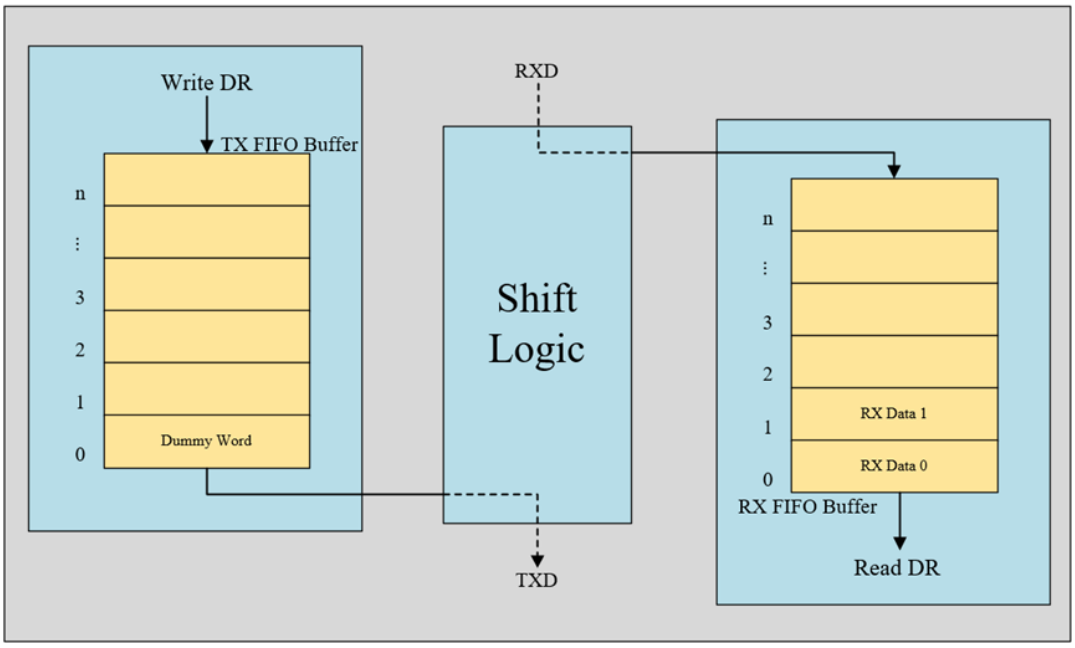

Figure & Table 4-6 FIFO status for receive only SPI transfers

For eeprom_read transfers (transfer mode field [9:8] of the Control Register 0 = 2'b11), opcode and/or EEPROM address are written into the transmit FIFO. During transmission of these control frames, received data is not captured by the DW_apb_ssi master. After the control frames have been transmitted, receive data from the EEPROM is stored into the receive FIFO.

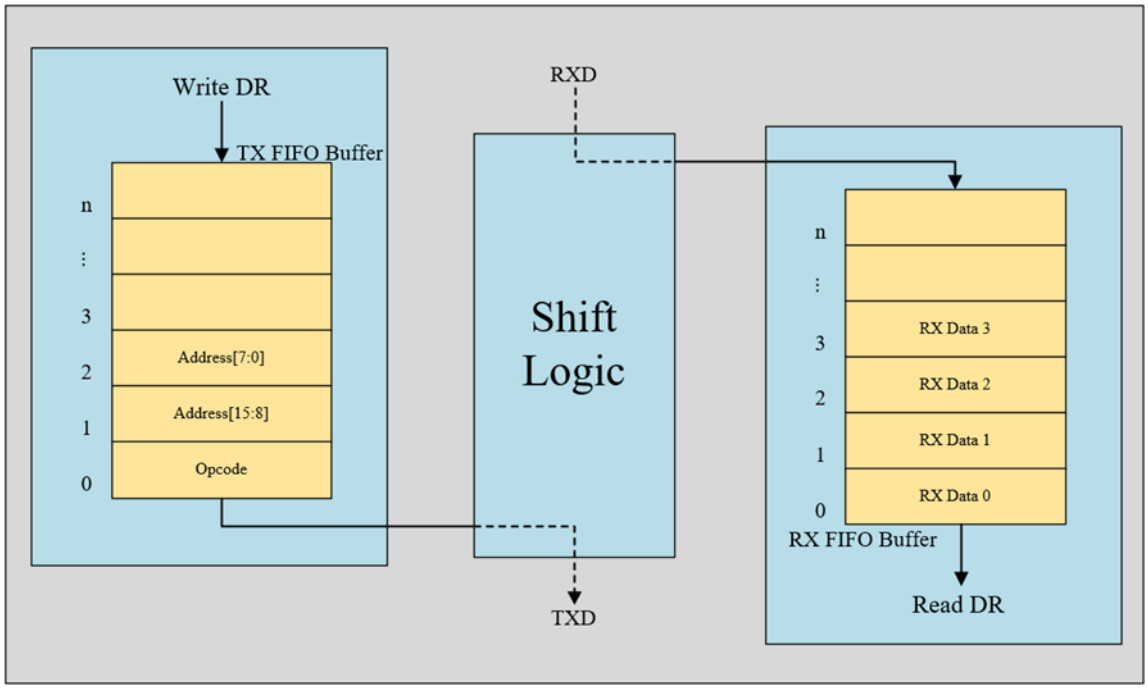

Figure & Table 4-7 FIFO status for EEPROM read transfer mode

4.4.6 RXD Sample Delay

When the DW_apb_ssi is configured as a master, additional logic can be included in the design in order to delay the default sample time of the rxd signal. This additional logic can help to increase the maximum achievable frequency on the serial bus.

Round trip routing delays on the sclk_out signal from the master and the rxd signal from the slave can mean that the timing of the rxd signal—as seen by the master—has moved away from the normal sampling time. Figure & Table 4-8 illustrates this situation.

The Slave uses the sclk_out signal from the master as a strobe in order to drive rxd signal data onto the serial bus. Routing and sampling delays on the sclk_out signal by the slave device can mean that the rxd bit has not stabilized to the correct value before the master samples the rxd signal. Figure & Table 4-8 shows an example of how a routing delay on the rxd signal can result in an incorrect rxd value at the default time when the master samples the port.

Without the RXD Sample Delay logic, you would have to increase the baud-rate for the transfer in order to ensure that the setup times on the rxd signal are within range; this results in reducing the frequency of the serial interface.

When the RXD Sample Delay logic is included, you can dynamically program a delay value in order to move the sampling time of the rxd signal equal to a number of ssi_clk cycles from the default.

The sample delay logic has a resolution of one ssi_clk cycle. Software can “train” the serial bus by coding a loop that continually reads from the slave and increments the master's RXD Sample Delay value until the correct data is received by the master.

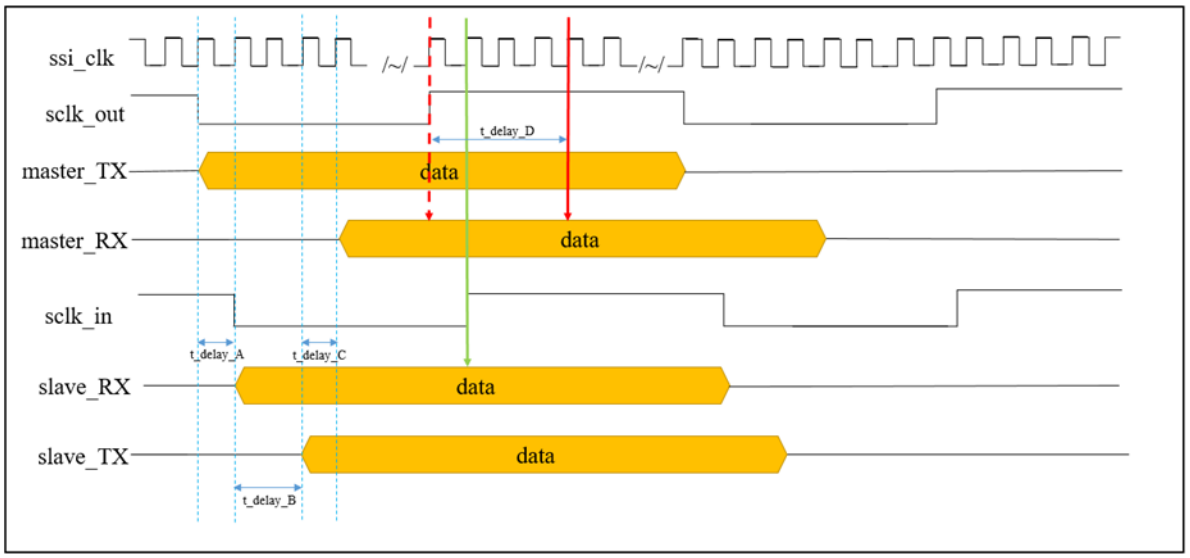

Figure & Table 4-8 Effects of round-trip routing delays on sclk_out signal

4.4.7 Data Transfer

Data transfers are started by the serial-master device. When the DW_apb_ssi is enabled (SSI_EN=1), at least one valid data entry is present in the transmit FIFO and a serial-slave device is selected. When actively transferring data, the busy flag (BUSY) in the status register (SR) is set. You must wait until the busy flag is cleared before attempting a new serial transfer.

When the transfer mode is “transmit and receive” or “transmit only” (TMOD = 2'b00 or TMOD = 2'b01, respectively), transfers are terminated by the shift control logic when the transmit FIFO is empty. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data have been transmitted. The transmit FIFO threshold level (TXFTLR) can be used to early interrupt (ssi_txe_intr) the processor indicating that the transmit FIFO buffer is nearly empty. When a DMA is used for APB accesses, the transmit data level (DMATDLR) can be used to early request (dma_tx_req) the DMA controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. You may also write a block of data (at least two FIFO entries) into the transmit FIFO before enabling a serial slave. This ensures that serial transmission does not begin until the number of data-frames that make up the continuous transfer are present in the transmit FIFO.

When the transfer mode is “receive only” (TMOD = 2'b10), a serial transfer is started by writing one“dummy” data word into the transmit FIFO when a serial slave is selected. The txd output from the DW_apb_ssi is held at a constant logic level for the duration of the serial transfer. The transmit FIFO is popped only once at the beginning and may remain empty for the duration of the serial transfer. The end of the serial transfer is controlled by the “number of data frames” (NDF) field in CTRLR1.

If, for example, you want to receive 24 data frames from a serial-slave peripheral, you should program the NDF field with the value 23; the receive logic terminates the serial transfer when the number of frames received is equal to the NDF value + 1. This transfer mode increases the bandwidth of the APB bus as the transmit FIFO never needs to be serviced during the transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow.
When the transfer mode is "eeprom_read" (TMOD = 2'b11), a serial transfer is started by writing the opcode and/or address into the transmit FIFO when a serial slave (EEPROM) is selected. The opcode and address are transmitted to the EEPROM device, after which read data is received from the EEPROM device and stored in the receive FIFO. The end of the serial transfer is controlled by the NDF field in CTRLR1.

4.4.8 XIP Mode Support in SPI Mode

The XIP mode enables transfer of SPI data directly through the APB interface without writing the data register of DW_apb_ssi. XIP mode can be enabled in DW_apb_ssi by selecting the SSI_XIP_EN configuration parameter, which includes an extra sideband signal xip_en, on the APB interface. This signal indicates whether APB transfers are register read-write or XIP reads. If the xip_en signal is driven to 1, DW_apb_ssi expects only read request on the APB interface. This request is translated to SPI read on the serial interface and soon after the data is received, the data is returned to the APB interface in the same transaction.

4.4.9 DMA Controller Interface

The DW_apb_ssi has optional built-in DMA capability which can be selected at configuration time; it 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. While the DW_apb_ssi DMA operation is designed in a generic way to fit any DMA controller as easily as possible, it is designed to work seamlessly, and best used, with the DesignWare DMA Controller, the DW_axi_dmac.
When the DW_apb_ssi interfaces to the DW_axi_dmac, the DW_axi_dmac is always a flow controller; that is, it controls the block size. This must be programmed by software in the DW_axi_dmac.
The DW_apb_ssi uses two DMA channels, one for the transmit data and one for the receive data.
The DW_apb_ssi has these DMA registers:

- DMACR – Control register to enable DMA operation.
- DMATDLR – Register to set the transmit FIFO level at which a DMA request is made.
- DMARDLR – Register to set the receive FIFO level at which a DMA request is made.

The DW_apb_ssi uses the following handshaking signals to interface with the DMA controller:

- dma_tx_req
- dma_tx_single
- dma_tx_ack
- dma_rx_req
- dma_rx_single

- dma_rx_ack

To enable the DMA controller interface on the DW_apb_ssi, you must write the DMA Control Register (DMACR). Writing a 1 into the TDMAE bit field of DMACR register enables the DW_apb_ssi transmit handshaking interface. Writing a 1 into the RDMAE bit field of the DMACR register enables the DW_apb_ssi receive handshaking interface.

Figure & Table 4-9 DMA Transmit Data Level (DMATDL) decode value

DMATDL Value	Description
0000_0000	dma_tx_req is asserted when 0 data entries are present in the transmit FIFO.
0000_0001	dma_tx_req is asserted when 1 or less data entry is present in the transmit FIFO.
0000_0010	dma_tx_req is asserted when 2 or less data entries are present in the transmit FIFO.
0000_0011	dma_tx_req is asserted when 3 or less data entries are present in the transmit FIFO.
...	...
...	...
1111_1100	dma_tx_req is asserted when 252 or less data entries are present in the transmit FIFO.
1111_1101	dma_tx_req is asserted when 253 or less data entries are present in the transmit FIFO.
1111_1110	dma_tx_req is asserted when 254 or less data entries are present in the transmit FIFO.
1111_1111	dma_tx_req is asserted when 255 or less data entries are present in the transmit FIFO

Figure & Table 4-10 DMA Receive Data Level (DMARDL) decode value

DMARDL Value	Description
0000_0000	dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.
0000_0001	dma_rx_req is asserted when 2 or more data entries are present in the receive FIFO.
0000_0010	dma_rx_req is asserted when 3 or more data entries are present in the receive FIFO.
0000_0011	dma_rx_req is asserted when 4 or more valid data entries are present in the receive FIFO.
...	...
...	...
1111_1100	dma_rx_req is asserted when 253 or more data entries are present in the receive FIFO.
1111_1101	dma_rx_req is asserted when 254 or more data entries are present in the receive FIFO.
1111_1110	dma_rx_req is asserted when 255 or more data entries are present in the receive FIFO.
1111_1111	dma_rx_req is asserted when 256 data entries are present in the receive FIFO.

4.4.9.1 Transmit Watermark Level and Transmit FIFO Underflow

During DW_apb_ssi serial transfers, transmit FIFO requests are made to the DW_axi_dmac whenever the number of entries in the transmit FIFO is less than or equal to the DMA Transmit Data Level Register (DMATDLR) value; this is known as the watermark level. The DW_axi_dmac responds by writing a burst of data to the transmit FIFO buffer, of length CTLx.DEST_MSIZE.

Data should be fetched from the DMA often enough for the transmit FIFO to perform serial transfers continuously; that is, when the FIFO begins to empty another DMA request should be triggered. Otherwise the FIFO runs out of data (underflow). To prevent this condition, you must set the watermark level correctly.

Consider the example where the assumption is made:

DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - SSI.DMATDLR

Here the number of data items to be transferred in a DMA burst is equal to the empty space in the Transmit FIFO. Consider two different watermark level settings.

Case 1: DMATDLR = 2

- Transmit FIFO watermark level = SSI.DMATDLR = 2
- DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - SSI.DMATDLR = 6
- SSI transmit FIFO_DEPTH = 8
- DMA.CTLx.BLOCK_TS = 30

Therefore, the number of burst transactions needed equals the block size divided by the number of data items per burst:

DMA.CTLx.BLOCK_TS/DMA.CTLx.DEST_MSIZE = 30/6 = 5

The number of burst transactions in the DMA block transfer is 5. But the watermark level, SSI.DMATDLR, is quite low. Therefore, the probability of an SSI underflow is high where the SSI serial transmit line needs to transmit data, but where there is no data left in the transmit FIFO. This occurs because the DMA has no time to service the DMA request before the transmit FIFO becomes empty.

Case 2: DMATDLR = 6

- Transmit FIFO watermark level = SSI.DMATDLR = 6
- DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - SSI.DMATDLR = 2
- SSI transmit FIFO_DEPTH = 8
- DMA.CTLx.BLOCK_TS = 30

Number of burst transactions in Block:

DMA.CTLx.BLOCK_TS/DMA.CTLx.DEST_MSIZE = 30/2 = 15

In this block transfer, there are 15 destination burst transactions in a DMA block transfer. But the watermark level, SSI.DMATDLR, is high. Therefore, the probability of an SSI underflow is low because the DMA controller has plenty of time to service the destination burst transaction request before the SSI transmit FIFO becomes empty.

Thus, the second case has a lower probability of underflow at the expense of more burst transactions per block. This provides a potentially greater amount of AMBA bursts per block and worse bus utilization than the former case.

Therefore, the goal in choosing a watermark level is to minimize the number of transactions per block, while at the same time keeping the probability of an underflow condition to an acceptable level. In practice, this is a function of the ratio of the rate at which the SSI transmits data to the rate at which the DMA can respond to destination burst requests.
For example, promoting the channel to the highest priority channel in the DMA, and promoting the DMA master interface to the highest priority master in the AMBA layer, increases the rate at which the DMA controller can respond to burst transaction requests. This in turn allows you to decrease the watermark level, which improves bus utilization without compromising the probability of an underflow occurring.
For optimal operation, DMA.CTLx.DEST_MSIZE should be set at the FIFO level that triggers a transmit DMA request; that is:
DMA.CTLx.DEST_MSIZE = SSI.FIFO_DEPTH - SSI.DMATDLR
Adhering to this equation reduces the number of DMA bursts needed for a block transfer, and this in turn improves AMBA bus utilization.

4.4.9.2 Receive Watermark Level and Receive FIFO Overflow

During DW_apb_ssi serial transfers, receive FIFO requests are made to the DW_axi_dmac whenever the number of entries in the receive FIFO is at or above the DMA Receive Data Level Register; that is, DMARDLR+1. This is known as the watermark level. The DW_axi_dmac responds by fetching a burst of data from the receive FIFO buffer of length CTLx.SRC_MSIZE.
Data should be fetched by the DMA often enough for the receive FIFO to accept serial transfers continuously; that is, when the FIFO begins to fill, another DMA transfer is requested. Otherwise, the FIFO fills with data (overflow). To prevent this condition, you must correctly set the watermark level.
Similar to choosing the transmit watermark level described earlier, the receive watermark level, DMARDLR+1, should be set to minimize the probability of overflow. It is a tradeoff between the number of DMA burst transactions required per block versus the probability of an overflow occurring.
For optimal operation, DMA.CTLx.SRC_MSIZE should be set at the watermark level; that is:
DMA.CTLx.SRC_MSIZE = SSI.DMARDLR + 1
Adhering to this equation reduces the number of DMA bursts in a block transfer, and this in turn can improve AMBA bus utilization.

4.4.10 Reset Signals

The DW_apb_ssi includes the following reset signals, each dedicated to its own clock domain:

- presetn – Resets logic in the pclk clock domain.
- ssi_rst_n – Resets logic in ssi_clk clock domain.

The recommended procedure for resetting the DW_apb_ssi is as follows:

1. Assert the ssi_rst_n and presetn signal. The sequence of asserting these two signals and their timing relationships with ssi_clk and pclk are not important.

2. De-assert the ssi_rst_n signal synchronously with ssi_clk.
3. De-assert the presetn signal synchronously with pclk.

4.5 Usage

Figure & Table 4-11 shows a typical software flow for starting a DW_apb_ssi master SPI serial transfer. The diagram also shows the hardware flow inside the serial-master component.

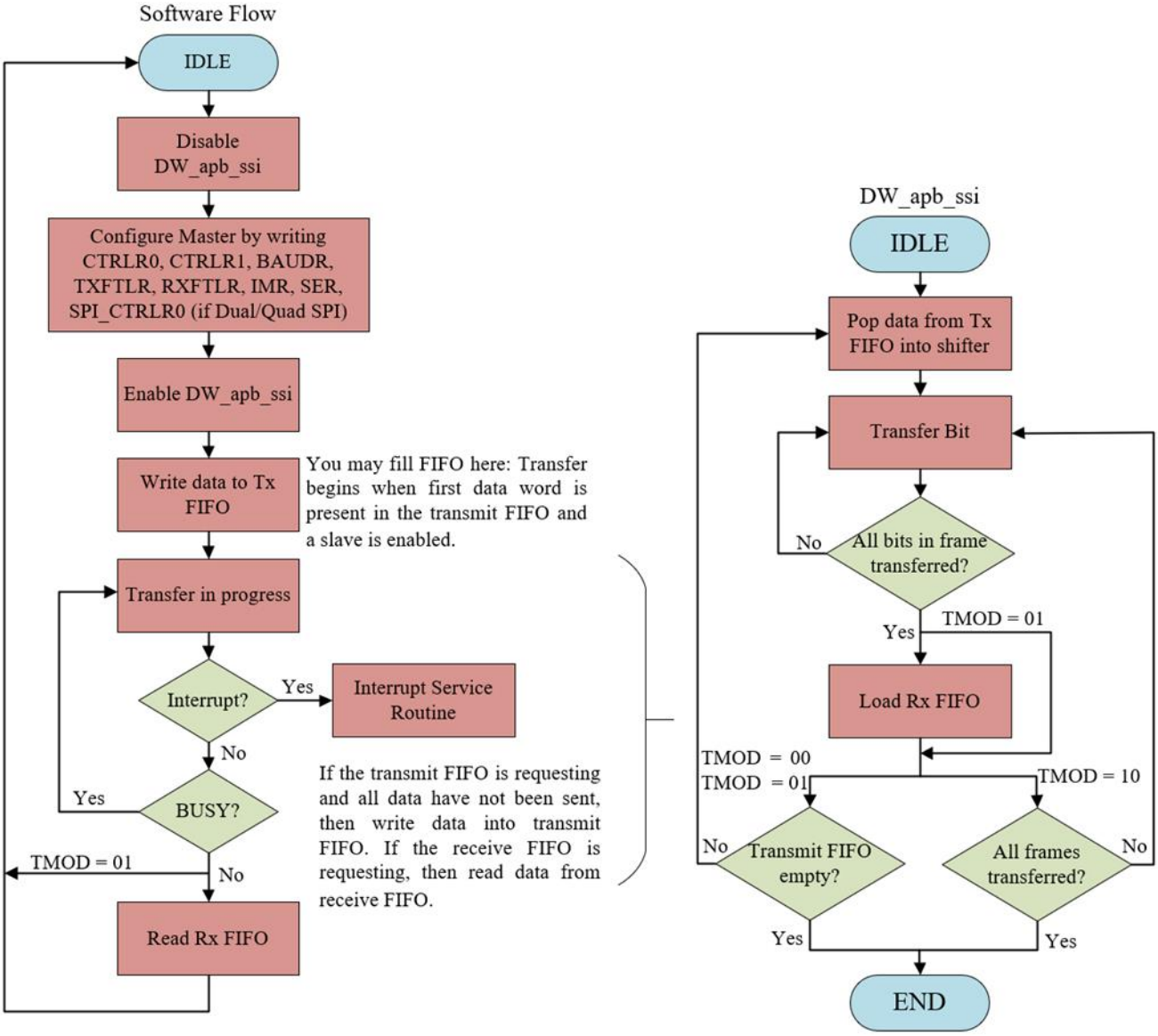

Figure & Table 4-11 DW_apb_ssi master SPI transfer flow

4.6 Registers

Figure & Table 4-12 Possible read and write behaviors

Read (or Write) Behavior	Description
RC	A read clears this register field.

Read (or Write) Behavior	Description
RS	A read sets this register field.
RM	A read modifies the contents of this register field.
Wo	You can only write to this register once field.
W1C	A write of 1 clears this register field.
W1S	A write of 1 sets this register field.
W1T	A write of 1 toggles this register field.
W0C	A write of 0 clears this register field.
W0S	A write of 0 sets this register field.
W0T	A write of 0 toggles this register field.
WC	Any write clears this register field.
WS	Any write sets this register field.
WM	Any write toggles this register field.
no Read Behavior attribute	You cannot read this register. It is Write-Only.
no Write Behavior attribute	You cannot write to this register. It is Read-Only.

Figure & Table 4-13 Memory access examples

Memory Access	Description
R	Read-only register field
W	Write-only register field
R/W	Read/write register field
R/W1C	You can read this register field. Writing 1 clears it.
RC/W1C	Reading this register field clears it. Writing 1 clears it.
R/Wo	You can read this register field. You can only write to it once.

4.6.1 ssi_memory_map/ssi_address_block Registers

4.6.1.1 CTRLR0

- Name: Control Register 0
- Description: This register controls the serial data transfer. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits

- Offset: 0x0
- Exists: Always

Figure & Table 4-14 Fields for register: CTRLR0

Bits	Name	Access	Description
31:25	RSVD_CTRLR0	R	SSTE Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
24	SSTE	R/W	Slave Select Toggle Enable When operating in SPI mode with clock phase (SCPH) set to 0, this register controls the behavior of the slave select line (ss_*_n) between data frames. If this register field is set to 1. the ss_*_n line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while ss_*_n is high; if this register field is set to 0, the ss_*_n will stay low and sclk will run continuously for the duration of the transfer. Value After Reset: 1 Exists: Always
23	RSVD_CTRLR0_23	R	CTRLR0_23 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
22:21	SPI_FRF	R(SPI); R/W(QSPI)	SPI Frame Format Selects data frame format for Transmitting/Receiving the data Bits only valid when SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode. When SSI_SPI_MODE is configured for "Dual Mode", 10/11 combination is reserved. When SSI_SPI_MODE is configured for "Quad Mode", 11 combination is reserved. Values: ■ 0x0 (STD_SPI_FRF): Standard SPI frame format ■ 0x1 (DUAL_SPI_FRF): Dual SPI frame format ■ 0x2 (QUAD_SPI_FRF): Quad SPI frame format ■ 0x3 (OCTAL_SPI_FRF): Octal SPI frame format Value After Reset: 0x0 Exists: Always
20:16	DFS_32	R/W	Data Frame Size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode.

Bits	Name	Access	Description
			These bits are only valid when SSI_MAX_XFER_SIZE is configured to 32. When the data frame size is programmed to be less than 32 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zeropadded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. Note: When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00. DFS value should be multiple of 2 if SPI_FRF = 0x01, DFS value should be multiple of 4 if SPI_FRF = 0x10, DFS value should be multiple of 8 if SPI_FRF = 0x11. Values: ■ 0x3 (FRAME_04BITS): 4-bit serial data transfer ■ 0x4 (FRAME_05BITS): 5-bit serial data transfer ■ 0x5 (FRAME_06BITS): 6-bit serial data transfer ■ 0x6 (FRAME_07BITS): 7-bit serial data transfer ■ 0x7 (FRAME_08BITS): 8-bit serial data transfer ■ 0x8 (FRAME_09BITS): 9-bit serial data transfer ■ 0x9 (FRAME_10BITS): 10-bit serial data transfer ■ 0xa (FRAME_11BITS): 11-bit serial data transfer ■ 0xb (FRAME_12BITS): 12-bit serial data transfer ■ 0xc (FRAME_13BITS): 13-bit serial data transfer ■ 0xd (FRAME_14BITS): 14-bit serial data transfer ■ 0xe (FRAME_15BITS): 15-bit serial data transfer ■ 0xf (FRAME_16BITS): 16-bit serial data transfer ■ 0x10 (FRAME_17BITS): 17-bit serial data transfer ■ 0x11 (FRAME_18BITS): 18-bit serial data transfer ■ 0x12 (FRAME_19BITS): 19-bit serial data transfer ■ 0x13 (FRAME_20BITS): 20-bit serial data transfer ■ 0x14 (FRAME_21BITS): 21-bit serial data transfer ■ 0x15 (FRAME_22BITS): 22-bit serial data transfer ■ 0x16 (FRAME_23BITS): 23-bit serial data transfer ■ 0x17 (FRAME_24BITS): 24-bit serial data transfer ■ 0x18 (FRAME_25BITS): 25-bit serial data transfer ■ 0x19 (FRAME_26BITS): 26-bit serial data transfer

Bits	Name	Access	Description
			■ 0x1a (FRAME_27BITS): 27-bit serial data transfer ■ 0x1b (FRAME_28BITS): 28-bit serial data transfer ■ 0x1c (FRAME_29BITS): 29-bit serial data transfer ■ 0x1d (FRAME_30BITS): 30-bit serial data transfer ■ 0x1e (FRAME_31BITS): 31-bit serial data transfer ■ 0x1f (FRAME_32BITS): 32-bit serial data transfer Value After Reset: 0x7 Exists: Always
15:12	CFS	R/W	Control Frame Size. Selects the length of the control word for the Microwire frame format. Values: ■ 0x0 (SIZE_01_BIT): 1-bit control word ■ 0x1 (SIZE_02_BIT): 2-bit control word ■ 0x2 (SIZE_03_BIT): 3-bit control word ■ 0x3 (SIZE_04_BIT): 4-bit control word ■ 0x4 (SIZE_05_BIT): 5-bit control word ■ 0x5 (SIZE_06_BIT): 6-bit control word ■ 0x6 (SIZE_07_BIT): 7-bit control word ■ 0x7 (SIZE_08_BIT): 8-bit control word ■ 0x8 (SIZE_09_BIT): 9-bit control word ■ 0x9 (SIZE_10_BIT): 10-bit control word ■ 0xa (SIZE_11_BIT): 11-bit control word ■ 0xb (SIZE_12_BIT): 12-bit control word ■ 0xc (SIZE_13_BIT): 13-bit control word ■ 0xd (SIZE_14_BIT): 14-bit control word ■ 0xe (SIZE_15_BIT): 15-bit control word ■ 0xf (SIZE_16_BIT): 16-bit control word Value After Reset: 0x0 Exists: Always
11	SRL	R/W	Shift Register Loop Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serial-slave and serial-master modes. When the DW_apb_ssi is configured as a slave in loopback mode, the ss_in_n and ssi_clk signals must be

Bits	Name	Access	Description
			provided by an external source. In this mode, the slave cannot generate these signals because there is nothing to which to loop back. Values: ■ 0x1 (TESTING_MODE): Test mode: Tx & Rx shift reg connected ■ 0x0 (NORMAL_MODE): Normal mode operation Value After Reset: 0x0 Exists: Always
10	SLV_OE	-	Slave Output Enable. Relevant only when the DW_apb_ssi is configured as a serial-slave device. When configured as a serial master, this bit field has no functionality. This bit enables or disables the setting of the ssi_oe_n output from the DW_apb_ssi serial slave. When SLV_OE = 1, the ssi_oe_n output can never be active. When the ssi_oe_n output controls the tri-state buffer on the txd output from the slave, a high impedance state is always present on the slave txd output when SLV_OE = 1. This is useful when the master transmits in broadcast mode (master transmits data to all slave devices). Only one slave may respond with data on the master rxd line. This bit is enabled after reset and must be disabled by software (when broadcast mode is used), if you do not want this device to respond with data. Values: ■ 0x1 (DISABLED): Slave Output is disabled. ■ 0x0 (ENABLED): Slave Output is enabled. Value After Reset: 0x0 Exists: No
9:8	TMOD	R/W	Transfer Mode Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer.

Bits	Name	Access	Description
			In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. This transfer mode is only valid when the DW_apb_ssi is configured as master device. 00- Transmit & Receive 01- Transmit Only 10- Receive Only 11- EEPROM Read When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00. There are only two valid combinations: 10 - Read 01 – Write Values: ■ 0x0 (TX_AND_RX): Transmit & Receive ■ 0x1 (TX_ONLY): Transmit Only or Write ■ 0x2 (RX_ONLY): Receive Only or Read ■ 0x3 (EEPROM_READ): EEPROM Read Value After Reset: 0x0 Exists: Always
7	SCPOL	R/W	Serial Clock Polarity Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the DW_apb_ssi master is not actively transferring data on the serial bus. Values: ■ 0x0 (SCLK_LOW): Inactive state of serial clock is low. ■ 0x1 (SCLK_HIGH): Inactive state of serial clock is high. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
6	SCPH	R/W	Serial Clock Phase Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. Values: ■ 0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit. ■ 0x1 (SCPH_START): Serial clock toggles at start of first data bit. Value After Reset: 0x0 Exists: Always
5:4	FRF	R/W	Frame Format Selects which serial protocol transfers the data. Values: ■ 0x0 (MOTOROLA_SPI): Motorolla SPI frame format ■ 0x1 (TEXAS_SSP): Texas Instruments SSP frame format ■ 0x2 (NS_MICROWIRE): National Microwire frame format ■ 0x3 (RESERVED): Reserved value Value After Reset: 0x0 Exists: Always
3:0	DFS	R	Data Frame Size This register field is only valid when SSI_MAX_XFER_SIZE is configured to 16. If SSI_MAX_XFER_SIZE is configured to 32, then writing to this field will not have any effect. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into

Bits	Name	Access	Description
			the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. Note: When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00. DFS value should be multiple of 2 if SPI_FRF = 01, DFS value should be multiple of 4 if SPI_FRF = 10, DFS value should be multiple of 8 if SPI_FRF = 11. Values: ■ 0x3 (FRAME_04BITS): 4-bit serial data transfer ■ 0x4 (FRAME_05BITS): 5-bit serial data transfer ■ 0x5 (FRAME_06BITS): 6-bit serial data transfer ■ 0x6 (FRAME_07BITS): 7-bit serial data transfer ■ 0x7 (FRAME_08BITS): 8-bit serial data transfer ■ 0x8 (FRAME_09BITS): 9-bit serial data transfer ■ 0x9 (FRAME_10BITS): 10-bit serial data transfer ■ 0xa (FRAME_11BITS): 11-bit serial data transfer ■ 0xb (FRAME_12BITS): 12-bit serial data transfer ■ 0xc (FRAME_13BITS): 13-bit serial data transfer ■ 0xd (FRAME_14BITS): 14-bit serial data transfer ■ 0xe (FRAME_15BITS): 15-bit serial data transfer ■ 0xf (FRAME_16BITS): 16-bit serial data transfer Value After Reset: 0x0 Exists: Always

4.6.1.2 CTRLR1

- Name: Control Register 1
- Description: This register exists only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. Control register 1 controls the end of serial transfers when in receive-only mode. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits
- Offset: 0x4
- Exists: Yes

Figure & Table 4-15 Fields for register: CTRLR1

Bits	Name	Access	Description
31:16	RSVD_CTRLR1	R	CTRLR1 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
15:0	NDF	R/W	Number of Data Frames When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the DW_apb_ssi. The DW_apb_ssi 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. When the DW_apb_ssi is configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the DW_apb_ssi is configured as a serial slave. Value After Reset: 0x0 Exists: Always

4.6.1.3 SSIENR

- Name: SSI Enable Register
- Description: This register enables and disables the DW_apb_ssi.
- Size: 32 bits
- Offset: 0x8
- Exists: Always

Figure & Table 4-16 Fields for register: SSIENR

Bits	Name	Access	Description
31:1	RSVD_SSIENR	R	SSIENR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
0	SSI_EN	R/W	SSI Enable Enables and disables all DW_apb_ssi operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the DW_apb_ssi control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk,

Bits	Name	Access	Description
			thus saving power consumption in the system. Values: ■ 0x0 (DISABLE): Disables serial transfer. ■ 0x1 (ENABLED): Enables serial transfer. Value After Reset: 0x0 Exists: Always

4.6.1.4 SER

- Name: Slave Enable Register
- Description: This register is valid only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. The register enables the individual slave select output lines from the DW_apb_ssi master. Up to 16 slave-select output pins are available on the DW_apb_ssi master. Register bits can be set or cleared when SSI_EN=0. If SSI_EN=1, then register bits can be set (to delay the slave select assertion while TX FIFO is getting filled) but cannot be cleared.
- Size: 32 bits
- Offset: 0x10
- Exists: Yes

Figure & Table 4-17 Fields for register: SER

Bits	Name	Access	Description
31:4	RSVD_SER	R	SER Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
3:0	SER	R/W	Slave Select Enable Flag Each bit in this register corresponds to a slave select line (ss_x_n) from the DW_apb_ssi master. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set. Values:

Bits	Name	Access	Description
			■ 0x0 (NOT_SELECTED): No slave selected. ■ 0x1 (SELECTED): Slave is selected. Value After Reset: 0x0 Exists: Always

4.6.1.5 BAUDR

- Name: Baud Rate Select
- Description: This register is valid only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. The register derives the frequency of the serial clock that regulates the data transfer. The 16-bit field in this register defines the ssi_clk divider value. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits
- Offset: 0x14
- Exists: Yes

Figure & Table 4-18 Fields for register: BAUDR

Bits	Name	Access	Description
31:16	RSVD_BAUDR	R	BAUDR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
15:0	SCKDV	R/W	SSI Clock Divider 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 = Fssi_clk/SCKDV where SCKDV is any even value between 2 and 65534. For example: for Fssi_clk = 3.6864MHz and SCKDV =2, Fsclk_out = 3.6864/2 = 1.8432MHz Value After Reset: 0x0 Exists: Always

4.6.1.6 TXFTLR

- Name: Transmit FIFO Threshold Level

- Description: This register controls the threshold value for the transmit FIFO memory. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits
- Offset: 0x18
- Exists: Always

Figure & Table 4-19 Fields for register: TXFTLR

Bits	Name	Access	Description
31:8	RSVD_TXFTLR	R	TXFTLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
7:0	TFT	R/W	Transmit FIFO Threshold Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. For information on the Transmit FIFO Threshold values, see the "Master SPI and SSP Serial Transfers" in the DW_apb_ssi Databook. ssi_txe_intr is asserted when TFT or less data entries are present in transmit FIFO. Value After Reset: 0x0 Exists: Always

4.6.1.7 RXFTLR

- Name: Receive FIFO Threshold Level
- Description: This register controls the threshold value for the receive FIFO memory. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits
- Offset: 0x1c
- Exists: Always

Figure & Table 4-20 Fields for register: RXFTLR

Bits	Name	Access	Description
31:8	RSVD_RXFTLR	R	RXFTLR Reserved bits - Read Only Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
7:0	RFT	R/W	Receive FIFO Threshold Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered. For information on the Receive FIFO Threshold values, see the "Master SPI and SSP Serial Transfers" in the DW_apb_ssi Databook. ssi_rxf_intr is asserted when RFT or more data entries are present in receive FIFO. Value After Reset: 0x0 Exists: Always

4.6.1.8 TXFLR

- Name: Transmit FIFO Level Register
- Description: This register contains the number of valid data entries in the transmit FIFO memory.
- Size: 32 bits
- Offset: 0x20
- Exists: Always

Figure & Table 4-21 Fields for register: TXFLR

Bits	Name	Access	Description
31:9	RSVD_TXFLR	R	TXFLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
8:0	TXTFL	R	Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.9 RXFLR

- Name: Receive FIFO Level Register
- Description: This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.
- Size: 32 bits
- Offset: 0x24
- Exists: Always

Figure & Table 4-22 Fields for register: RXFLR

Bits	Name	Access	Description
31:9	RSVD_RXFLR	R	RXFLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: True
8:0	RXTFL	R	Receive FIFO Level Contains the number of valid data entries in the receive FIFO. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.10 SR

- Name: Status Register
- Description: This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time. None of the bits in this register request an interrupt.
- Size: 32 bits
- Offset: 0x28
- Exists: Always

Figure & Table 4-23 Fields for register: SR

Bits	Name	Access	Description
31:7	RSVD_SR	R	SR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: True
6	DCOL	R	Data Collision Error Relevant only when the DW_apb_ssi is configured as a

Bits	Name	Access	Description
			master device. This bit will be set if ss_in_n input is asserted by other master, when the DW_apb_ssi master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. Values: ■ 0x0 (NO_ERROR_CONDITION): No error ■ 0x1 (TX_COLLISION_ERROR): Transmit data collision error Value After Reset: 0x0 Exists: Yes Volatile: True
5	TXE	R	Transmission Error Set if the transmit FIFO is empty when a transfer is started. This bit can be set only when the DW_apb_ssi is configured as a slave device. Data from the previous transmission is resent on the txd line. This bit is cleared when read. Values: ■ 0x0 (NO_ERROR): No error ■ 0x1 (TX_ERROR): Transmission error Value After Reset: 0x0 Exists: No Volatile: True
4	RFF	R	Receive FIFO Full When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. Values: ■ 0x0 (NOT_FULL): Receive FIFO is not full. ■ 0x1 (FULL): Receive FIFO is full. Value After Reset: 0x0 Exists: Always Volatile: True
3	RFNE	R	Receive FIFO Not Empty Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the

Bits	Name	Access	Description
			receive FIFO. Values: ■ 0x0 (EMPTY): Receive FIFO is empty. ■ 0x1 (NOT_EMPTY): Receive FIFO is not empty. Value After Reset: 0x0 Exists: Always Volatile: True
2	TFE	R	Transmit FIFO Empty When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. Values: ■ 0x0 (NOT_EMPTY): Transmit FIFO is not empty. ■ 0x1 (EMPTY): Transmit FIFO is empty. Value After Reset: 0x1 Exists: Always Volatile: True
1	TFNF	R	Transmit FIFO Not Full Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. Values: ■ 0x0 (FULL): Transmit FIFO is full. ■ 0x1 (NOT_FULL): Transmit FIFO is not full. Value After Reset: 0x1 Exists: Always Volatile: True
0	BUSY	R	SSI Busy Flag When set, indicates that a serial transfer is in progress; when cleared indicates that the DW_apb_ssi is idle or disabled. Values: ■ 0x0 (INACTIVE): DW_apb_ssi is idle or disabled. ■ 0x1 (ACTIVE): DW_apb_ssi is actively transferring data. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
			Volatile: True

4.6.1.11 IMR

- Name: Interrupt Mask Register
- Description: This read/write reigster masks or enables all interrupts generated by the DW_apb_ssi. When the DW_apb_ssi is configured as a slave device, the MSTIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.
- Size: 32 bits
- Offset: 0x2c
- Exists: Always

Figure & Table 4-24 Fields for register: IMR

Bits	Name	Access	Description
31:6	RSVD_IMR	R	IMR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
5	MSTIM	R/W	Multi-Master Contention Interrupt Mask. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. Values: ■ 0x0 (MASKED): ssi_mst_intr interrupt is masked. ■ 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked. Value After Reset: 1 Exists: Yes
4	RXFIM	R/W	Receive FIFO Full Interrupt Mask Values: ■ 0x0 (MASKED): ssi_rxf_intr interrupt is masked. ■ 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked. Value After Reset: 0x1 Exists: Always
3	RXOIM	R/W	Receive FIFO Overflow Interrupt Mask Values: ■ 0x0 (MASKED): ssi_rxo_intr interrupt is masked. ■ 0x1 (UNMASKED): ssi_rxo_intr interrupt is not

Bits	Name	Access	Description
			masked. Value After Reset: 0x1 Exists: Always
2	RXUIM	R/W	Receive FIFO Underflow Interrupt Mask Values: ■ 0x0 (MASKED): ssi_rxu_intr interrupt is masked. ■ 0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked. Value After Reset: 0x1 Exists: Always
1	TXOIM	R/W	Transmit FIFO Overflow Interrupt Mask Values: ■ 0x0 (MASKED): ssi_txo_intr interrupt is masked. ■ 0x1 (UNMASKED): ssi_txo_intr interrupt is not masked. Value After Reset: 0x1 Exists: Always
0	TXEIM	R/W	Transmit FIFO Empty Interrupt Mask Values: ■ 0x0 (MASKED): ssi_txe_intr interrupt is masked. ■ 0x1 (UNMASKED): ssi_txe_intr interrupt is not masked. Value After Reset: 0x1 Exists: Always

4.6.1.12 ISR

- Name: Interrupt Status Register
- Description: This register reports the status of the DW_apb_ssi interrupts after they have been masked.
- Size: 32 bits
- Offset: 0x30
- Exists: Always

Figure & Table 4-25 Fields for register: ISR

Bits	Name	Access	Description
31:6	RSVD_ISR	R	ISR Reserved bits - Read Only

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always Volatile: True
5	MSTIS	R	Multi-Master Contention Interrupt Status. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. Values: ■ 0x0 (INACTIVE): ssi_mst_intr interrupt not active after masking. ■ 0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking. Value After Reset: 0x0 Exists: Yes Volatile: True
4	RXFIS	R	Receive FIFO Full Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking. ■ 0x1 (ACTIVE): ssi_rxf_intr interrupt is full after masking. Value After Reset: 0x0 Exists: Always Volatile: True
3	RXOIS	R	Receive FIFO Overflow Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking. ■ 0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking. Value After Reset: 0x0 Exists: Always Volatile: True
2	RXUIS	R	Receive FIFO Underflow Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking. ■ 0x1 (ACTIVE): ssi_rxu_intr interrupt is active after

Bits	Name	Access	Description
			masking. Value After Reset: 0x0 Exists: Always Volatile: True
1	TXOIS	R	Transmit FIFO Overflow Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking. ■ 0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking. Value After Reset: 0x0 Exists: Always Volatile: True
0	TXEIS	R	Transmit FIFO Empty Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_txe_intr interrupt is not active after masking. ■ 0x1 (ACTIVE): ssi_txe_intr interrupt is active after masking. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.13 RISR

- Name: Raw Interrupt Status Register
- Description: This read-only register reports the status of the DW_apb_ssi interrupts prior to masking.
- Size: 32 bits
- Offset: 0x34
- Exists: Always

Figure & Table 4-26 Fields for register: RISR

Bits	Name	Access	Description
31:6	RSVD_RISR	R	RISR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true

Bits	Name	Access	Description
5	MSTIR	R	Multi-Master Contention Raw Interrupt Status. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. Values: ■ 0x0 (INACTIVE): ssi_mst_intr interrupt is not active prior to masking. ■ 0x1 (ACTIVE): ssi_mst_intr interrupt is active prior masking. Value After Reset: 0x0 Exists: Yes Volatile: True
4	RXFIR	R	Receive FIFO Full Raw Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active prior to masking. ■ 0x1 (ACTIVE): ssi_rxf_intr interrupt is active prior to masking. Value After Reset: 0x0 Exists: Always Volatile: True
3	RXOIR	R	Receive FIFO Overflow Raw Interrupt Status Values: ■ 0x1 (ACTIVE): ssi_rxo_intr interrupt is not active prior to masking. ■ 0x0 (INACTIVE): ssi_rxo_intr interrupt is active prior masking. Value After Reset: 0x0 Exists: Always Volatile: True
2	RXUIR	R	Receive FIFO Underflow Raw Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active prior to masking. ■ 0x1 (ACTIVE): ssi_rxu_intr interrupt is active prior to masking. Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
			Volatile: True
1	TXOIR	R	Transmit FIFO Overflow Raw Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior to masking. ■ 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior masking. Value After Reset: 0x0 Exists: Always Volatile: True
0	TXEIR	R	Transmit FIFO Empty Raw Interrupt Status Values: ■ 0x0 (INACTIVE): ssi_txe_intr interrupt is not active prior to masking. ■ 0x1 (ACTIVE): ssi_txe_intr interrupt is active prior masking. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.14 TXOICR

- Name: Transmit FIFO Overflow Interrupt Clear Registers
- Description: Transmit FIFO Overflow Interrupt Clear Register
- Size: 32 bits
- Offset: 0x38
- Exists: Always

Figure & Table 4-27 Fields for register: TXOICR

Bits	Name	Access	Description
31:1	RSVD_TXOICR	R	TXOICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: True
0	TXOICR	R	Clear Transmit FIFO Overflow Interrupt This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.15 RXOICR

- Name: Receive FIFO Overflow Interrupt Clear Register
- Description: Receive FIFO Overflow Interrupt Clear Register
- Size: 32 bits
- Offset: 0x3c
- Exists: Always

Figure & Table 4-28 Fields for register: RXOICR

Bits	Name	Access	Description
31:1	RSVD_RXOICR	R	RXOICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: True
0	RXOICR	R	Clear Receive FIFO Overflow Interrupt This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.16 RXUICR

- Name: Receive FIFO Underflow Interrupt Clear Register
- Description: Receive FIFO Underflow Interrupt Clear Register
- Size: 32 bits
- Offset: 0x40
- Exists: Always

Figure & Table 4-29 Fields for register: RXUICR

Bits	Name	Access	Description
31:1	RSVD_RXUICR	R	RXUICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always

Bits	Name	Access	Description
			Volatile: true
0	RXUICR	R	Clear Receive FIFO Underflow Interrupt This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.17 MSTICR

- Name: Multi-Master Interrupt Clear Register
- Description: Multi-Master Interrupt Clear Register
- Size: 32 bits
- Offset: 0x44
- Exists: Always

Figure & Table 4-30 Fields for register: MSTICR

Bits	Name	Access	Description
31:1	RSVD_MSTICR	R	MSTICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: True
0	MSTICR	R	Clear Multi-Master Contention Interrupt This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.18 ICR

- Name: Interrupt Clear Register
- Description: Interrupt Clear Register
- Size: 32 bits
- Offset: 0x48
- Exists: Always

Figure & Table 4-31 Fields for register: ICR

Bits	Name	Access	Description
31:1	RSVD_ICR	R	ICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: True
0	ICR	R	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.19 DMACR

- Name: DMA Control Register
- Description: This register is only valid when DW_apb_ssi is configured with a set of DMA Controller interface signals (SSI_HAS_DMA = 1). When DW_apb_ssi is not configured for DMA operation, this register will not exist and writing to the register's address will have no effect; reading from this register address will return zero. The register is used to enable the DMA Controller interface operation.
- Size: 32 bits
- Offset: 0x4c
- Exists: Yes

Figure & Table 4-32 Fields for register: DMACR

Bits	Name	Access	Description
31:2	RSVD_DMACR	R	DMACR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
1	TDMAE	R/W	Transmit DMA Enable This bit enables/disables the transmit FIFO DMA channel. Values: ■ 0x0 (DISABLE): Transmit DMA disabled. ■ 0x1 (ENABLED): Transmit DMA enabled. Value After Reset: 0x0

Bits	Name	Access	Description
			Exists: Always
0	RDMAE	R/W	Receive DMA Enable This bit enables/disables the receive FIFO DMA channel. Values: ■ 0x0 (DISABLE): Receive DMA disabled. ■ 0x1 (ENABLED): Receive DMA enabled. Value After Reset: 0x0 Exists: Always

4.6.1.20 DMATDLR

- Name: DMA Transmit Data Level
- Description: This register is only valid when the DW_apb_ssi is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When DW_apb_ssi is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero.
- Size: 32 bits
- Offset: 0x50
- Exists: Yes

Figure & Table 4-33 Fields for register: DMATDLR

Bits	Name	Access	Description
31:8	RSVD_DMATDLR	R	DMATDLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
7:0	DMATDL	R/W	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 TDMAE = 1. For information on the DMATDL decode values, see the "Slave SPI and SSP Serial Transfers" section in the DW_apb_ssi Databook. dma_tx_req is asserted when DMATDL or less data entries are present in the transmit FIFO Value After Reset: 0x0 Exists: Always

4.6.1.21 DMARDLR

- Name: DMA Receive Data Level
- Description: This register is only valid when DW_apb_ssi is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When DW_apb_ssi is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero.
- Size: 32 bits
- Offset: 0x54
- Exists: Yes

Figure & Table 4-34 Fields for register: DMARDLR

Bits	Name	Access	Description
31:8	RSVD_DMARDLR	R	DMARDLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
7:0	DMARDL	R/W	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 RDMAE=1. For information on the DMARDL decode values, see the "Slave SPI and SSP Serial Transfers" section in the DW_apb_ssi Databook. dma_rx_req is asserted when DMARDL or more valid data entries are present in the receive FIFO. Value After Reset: 0x0 Exists: Always

4.6.1.22 IDR

- Name: Identification Register
- Description: This register contains the peripherals identification code, which is written into the register at configuration time using CoreConsultant.
- Size: 32 bits
- Offset: 0x58
- Exists: Always

Figure & Table 4-35 Fields for register: IDR

Bits	Name	Access	Description
31:0	IDCODE	R	Identification code

Bits	Name	Access	Description
			The register contains the peripheral's identification code, which is written into the register at configuration time using CoreConsultant. Value After Reset: 0xffff_ffff Exists: Always

4.6.1.23 SSI_VERSION_ID

- Name: coreKit version ID Register
- Description: This read-only register stores the specific DW_apb_ssi component version.
- Size: 32 bits
- Offset: 0x5c
- Exists: Always

Figure & Table 4-36 Fields for register: SSI_VERSION_ID

Bits	Name	Access	Description
31:0	SSI_COMP_VERSION	R	Contains the hex representation of the Synopsys component version. Consists of ASCII value for each number in the version, followed by *. For example, 32_30_31_2A represents the version 2.01*. Value After Reset: 0x3430_322a Exists: Always

4.6.1.24 DRx (for x = 0; x <= 35)

- Name: Data Register x
- Description: The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. NOTE: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable.
- Size: 32 bits
- Offset: 0x60
- Exists: Always

Figure & Table 4-37 Fields for register: DRx (for x = 0; x <= 35)

Bits	Name	Access	Description
31:0	DR	R/W	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Value After Reset: 0x0 Exists: Always Volatile: True

4.6.1.25 RX_SAMPLE_DLY

- Name: RX Sample Delay Register
- Description: This register is only valid when the DW_apb_ssi is configured with rxd sample delay logic (SSI_HAS_RX_SAMPLE_DELAY==1). When the DW_apb_ssi is not configured with rxd sample delay logic, this register will not exist and writing to its address location will have no effect; reading from its address will return zero. This register controls the number of ssi_clk cycles that are delayed (from the default sample time) before the actual sample of the rxd input occurs. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits
- Offset: 0xf0
- Exists: Yes

Figure & Table 4-38 Fields for register: RX_SAMPLE_DLY

Bits	Name	Access	Description
31:8	RSVD_RX_SAMPLE_DLY	R	SAMPLE_DLY Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
7:0	RSD	R/W	Rxd Sample Delay This register is used to delay the sample of the rxd input port. Each value represents a single ssi_clk delay on the sample of rxd. Note: If this register is programmed with a value that exceeds the depth of the internal shift registers (SSI_RX_DLY_SR_DEPTH), zero delay will be applied to the rxd sample. Value After Reset: 0x0 Exists: Always

4.6.1.26 SPI_CTRLR0

- Name: SPI Control Register
- Description: This register is valid only when SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode. This register is used to control the serial data transfer in SPI mode of operation. The register is only relevant when SPI_FRF is set to either 01 or 10 or 11. It is not possible to write to this register when the DW_apb_ssi is enabled (SSI_EN=1). The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.
- Size: 32 bits
- Offset: 0xf4
- Exists: QSPI Only

Figure & Table 4-39 Fields for register: SPI_CTRLR0

Bits	Name	Access	Description
31:19	RSVD_SPI_CTRLR0	R	SPI_CTRLR0 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
18	SPI_RXDS_EN	R	Read data strobe enable bit Once this bit is set to 1, DW_apb_ssi will use Read data strobe (rxds) to capture read data in DDR mode. Value After Reset: 0x0 Exists: Always
17	INST_DDR_EN	R	Instruction DDR enable bit This will enable Dual-data rate transfer for Instruction phase. Value After Reset: 0x0 Exists: Always
16	SPI_DDR_EN	R	SPI DDR enable bit This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI. Value After Reset: 0x0 Exists: Always
15:11	WAIT_CYCLES	R/W	Wait cycles Number of wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. This value is specified as number of SPI clock cycles. For information on the WAIT_CYCLES decode value, see "Read Operation in Enhanced SPI Modes" section in the DW_apb_ssi Databook.

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
10	RSVD_SPI_CTRLR0_10	R	CTRLR0_10 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
9:8	INST_L	R/W	Instruction Length Dual/Quad/Octal mode instruction length in bits. Values: ■ 0x0 (INST_L_0): 0-bit (No Instruction) ■ 0x1 (INST_L_1): 4-bit instruction ■ 0x2 (INST_L_2): 8-bit instruction ■ 0x3 (INST_L_3): 16-bit instruction Value After Reset: 0x2 Exists: Always
7:6	RSVD_SPI_CTRLR0_6_7	R	CTRLR0_6_7 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
5:2	ADDR_L	R/W	Address Length This bit defines Length of Address to be transmitted. Only after this much bits are programmed into the FIFO, the transfer can begin. For information on the ADDR_Ldecode value, see "Read Operation in Enhanced SPI Modes" section in the DW_apb_ssi Databook. Values: ■ 0x0 (ADDR_L_0): 0-bit address width ■ 0x1 (ADDR_L_1): 4-bit address width ■ 0x2 (ADDR_L_2): 8-bit address width ■ 0x3 (ADDR_L_3): 12-bit address width ■ 0x4 (ADDR_L_4): 16-bit address width ■ 0x5 (ADDR_L_5): 20-bit address width ■ 0x6 (ADDR_L_6): 24-bit address width ■ 0x7 (ADDR_L_7): 28-bit address width ■ 0x8 (ADDR_L_8): 32-bit address width ■ 0x9 (ADDR_L_9): 36-bit address width ■ 0xa (ADDR_L_10): 40-bit address width

Bits	Name	Access	Description
			■ 0xb (ADDR_L_11): 44-bit address width ■ 0xc (ADDR_L_12): 48-bit address width ■ 0xd (ADDR_L_13): 52-bit address width ■ 0xe (ADDR_L_14): 56-bit address width ■ 0xf (ADDR_L_15): 60-bit address width Value After Reset: 0x0 Exists: Always
1:0	TRANS_TYPE	R/W	Address and instruction transfer format Selects whether DW_apb_ssi will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRLR0.SPI_FRF field. 00: Instruction and address will be sent in Standard SPI Mode. 01: Instruction will be sent in Standard SPI Mode and address will be sent in the mode specified by CTRLR0.SPI_FRF. 10: Both instruction and address will be sent in the mode specified by SPI_FRF. 11: Reserved. Value After Reset: 0x0 Exists: Always