OMAP5912 Multimedia Processor Multimedia Card (MMC/SD/SDIO) Interface Reference Guide

Literature Number: SPRU765A
March 2004

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Preface

Read This First

About This Manual

This document describes the multimedia card (MMC) interface of the OMAP5912 multimedia processor.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

The following documents describe the OMAP5910 device and related peripherals. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

OMAP5912 Multimedia Processor Device Overview and Architecture Reference Guide (literature number SPRU748) introduces the setup, components, and features of the OMAP5912 multimedia processor and provides a high-level view of the device architecture.

OMAP5912 Multimedia Processor OMAP 3.2 Subsystem Reference Guide (literature number SPRU749) introduces and briefly defines the main features of the OMAP3.2 subsystem of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor DSP Sybsystem Reference Guide (literature number SPRU750) describes the OMAP5912 multimedia processor DSP subsystem. The digital signal processor (DSP) subsystem is built around a core processor and peripherals that interface with: 1) The ARM926EJS via the microprocessor unit interface (MPUI); 2) Various standard memories via the external memory interface (EMIF); 3) Various system peripherals via the TI peripheral bus (TIPB) bridge.

OMAP5912 Multimedia Processor Clocks Reference Guide (literature number SPRU751) describes the clocking mechanisms of the OMAP5912 multimedia processor. In OMAP5912, various clocks are created from special components such as the digital phase locked loop (DPLL) and the analog phase-locked loop (APLL).

OMAP5912 Multimedia Processor Initialization Reference Guide (literature number SPRU752) describes the reset architecture, the configuration, the initialization, and the boot ROM of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Power Management Reference Guide (literature number SPRU753) describes power management in the OMAP5912 multimedia processor. The ultralow-power device (ULPD) generates and manages clocks and reset signals to OMAP3.2 and to some peripherals. It controls chip-level power-down modes and handles chip-level wake-up events. In deep sleep mode, this module is still active to monitor wake-up events.This book describes the ULPD module and outline architecture.

OMAP5912 Multimedia Processor Security Features Reference Guide (literature number SPRU754) describes the security features of teh OMAP5912 multimedia processor. The OMAP5912 security scheme relies on the OMAP3.2 secure mode. The distributed security on the OMAP3.2 platform is a Texas Instruments solution to address m-commerce and security issues within a mobile phone environment. The OMAP3.2 secure mode was developed to bring hardware robustness to the overall OMAP5912 security scheme.

OMAP5912 Multimedia Processor Direct Memory Access (DMA) Support Reference Guide (literature number SPRU755) describes the direct memory access support of the OMAP5912 multimedia processor. The OMAP5912 processor has three DMAs:

- The system DMA is embedded in OMAP3.2. It handles DMA transfers associated with MPU and shared peripherals.
- The DSP DMA is embedded in OMAP3.2. It handles DMA transfers associated with DSP peripherals.
- The generic distributed DMA (GDD) is an OMAP5912 resource attached to the SSI peripheral. It handles only DMA transfers associated with the SSI peripheral.

OMAP5912 Multimedia Processor Memory Interfaces Reference Guide (literature number SPRU756) describes the memory interfaces of the OMAP5912 multimedia processor.

- SDRAM (external memory interface fast, or EMIFF)
- Asynchronous and synchronous burst memory (external memory interface slow, or EMIFS)
- NAND flash (hardware controller or software controller)
- CompactFlash on EMIFS interface
- Internal static RAM

OMAP5912 Multimedia Processor Interrupts Reference Guide (literature number SPRU757) describes the interrupts of the OMAP5912 multimedia processor. Three level 2 interrupt controllers are used in OMAP5912:

- One MPU level 2 interrupt handler (also referred to as MPU interrupt level 2) is implemented outside of OMAP3.2 and can handle 128 interrupts.
- One DSP level 2 interrupt handler (also referred to as DSP interrupt level 2.1) is instantiated outside of OMAP3.2 and can handle 64 interrupts.
- One OMAP3.2 DSP level 2 interrupt handler (referenced as DSP interrupt level 2.0) can handle 16 interrupts.

OMAP5912 Multimedia Processor Peripheral Interconnects Reference Guide (literature number SPRU758) describes various periperal interconnects of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Timers Reference Guide (literature number SPRU759) describes various timers of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Serial Interfaces Reference Guide (literature number SPRU760) describes the serial interfaces of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Universal Serial Bus (USB) Reference Guide (literature number SPRU761) describes the universal serial bus (USB) host on the OMAP5912 multimedia processor. The OMAP5912 processor provides several varieties of USB functionality. Flexible multiplexing of signals from the OMAP5912 USB host controller, the OMAP5912 USB function controller, and other OMAP5912 peripherals allow a wide variety of system-level USB capabilities. Many of the OMAP5912 pins can be used for USB-related signals or for signals from other OMAP5912 peripherals. The OMAP5912 top-level pin multiplexing

controls each pin individually to select one of several possible internal pin signal interconnections. When these shared pins are programmed for use as USB signals, the OMAP5912 USB signal multiplexing selects how the signals associated with the three OMAP5912 USB host ports and the OMAP5912 USB function controller can be brought out to OMAP5912 pins.

OMAP5912 Multimedia Processor Multi-channel Buffered Serial Ports (McBSPs) Reference Guide (literature number SPRU762) describes the three multi-channel buffered serial ports (McBSPs) available on the OMAP5912 device. The OMAP5912 device provides multiple high-speed multichannel buffered serial ports (McBSPs) that allow direct interface to codecs and other devices in a system.

OMAP5912 Multimedia Processor Camera Interface Reference Guide (literature number SPRU763) describes two camera inerfaces implemented in the OMAP5912 multimedia processor: compact serial camera port and camera parallel interface.

OMAP5912 Multimedia Processor Display Interface Reference Guide (literature number SPRU764) describes the display interface of the OMAP5912 multimedia processor.

- LCD module
- LCD data conversion module
- LED pulse generator
- Display interface

OMAP5912 Multimedia Processor Multimedia Card (MMC/SD/SDIO) (literature number SPRU765) describes the multimedia card (MMC) interface of the OMAP5912 multimedia processor. The multimedia card/secure data/secure digital IO (MMC/SD/SDIO) host controller provides an interface between a local host, such as a microprocessor unit (MPU) or digital signal processor (DSP), and either an MMC or SD memory card, plus up to four serial flash cards. The host controller handles MMC/SD/SDIO or serial port interface (SPI) transactions with minimal local host intervention.

OMAP5912 Multimedia Processor Keyboard Interface Reference Guide (literature number SPRU766) describes the keyboard interface of the OMAP5912 multimedia processor. The MPUIO module enables direct I/O communication between the MPU (through the public TIPB) and external devices. Two types of I/O can be used: specific I/Os dedicated for 8 x 8 keyboard connection, and general-purpose I/Os.

OMAP5912 Multimedia Processor General-Purpose Interface Reference Guide (literature number SPRU767) describes the general-purpose in-

terface of the OMAP5912 multimedia processor. There are four GPIO modules in the OMAP5912. Each GPIO peripheral controls 16 dedicated pins configurable either as input or output for general purposes. Each pin has an independent control direction set by a programmable register. The two−edge control registers configure events (rising edge, falling edge, or both edges) on an input pin to trigger interrupts or wake−up requests (depending on the system mode). In addition, an interrupt mask register masks out specified pins. Finally, the GPIO peripherals provide the set and clear capabilities on the data output registers and the interrupt mask registers. After detection, all event sources are merged and a single synchronous interrupt (per module) is generated in active mode, whereas a unique wake−up line is issued in idle mode. Eight data output lines of the GPIO3 are ORed together to generate a global output line at the OMAP5912 boundary. This global output line can be used in conjunction with the SSI to provide a CMT−APE interface to the OMAP5912.

OMAP5912 Multimedia Processor VLYNQ Serial Communications Interface Reference Guide (literature number SPRU768) describes the VLYNQ of the OMAP5912 multimedia processor.

VLYNQ is a serial communications interface that enables the extension of an internal bus segment to one or more external physical devices. The external devices are mapped into local, physical address space and appear as if they are on the internal bus of the OMAP 5912. The external devices must also have a VLYNQ interface. The VLYNQ module serializes bus transactions in one device, transfers the serialized data between devices via a VLYNQ port, and de-serializes the transaction in the external device.

OMAP5912 includes one VLYNQ module connected on OCPT2 target port and OCPI initiator port. These connections are configured via a static switch, which selects either SSI or VLYNQ module. This switch, forbids the simultaneous use of GDD/SSI and VLYNQ. The switch is controlled by the VLYNQ_EN bit in the OMAP5912 configuration control register (CONF_5912_CTRL).

OMAP5912 Multimedia Processor Pinout Reference Guide (literature number SPRU769) provides the pinout of the OMAP5912 multimedia processor. After power-up reset, the user can change the configuration of the default interfaces. If another interface is available on top of the default, it is possible to enable a new interface for each ball by setting the corresponding 3-bit field of the associated FUNC_MUX_CTRL register. It is also possible to configure on-chip pullup/pulldown. This document

also describes the various power domains so that the user can apply the different interfaces seamlessly with external components.

OMAP5912 Multimedia Processor Window Tracer (WT) Reference Guide (literature number SPRU770) describes the window tracer module used to capture the memory transactions from four interfaces: EMIFF, EMIFS, OCP-T1, and OCP-T2. This module is located in the OMAP3.2 traffic controller (TC).

OMAP5912 Multimedia Processor Real-Time Clock Reference Guide (literature number SPRUxxx) describes the real-time clock of the OMAP5912 multimedia processor. The real-time clock (RTC) block is an embedded real-time clock module directly accessible from the TIPB bus interface.

Trademarks

Contents

Figures

Tables

Notes

Multimedia Card (MMC/SD/SDIO) Interface

This document describes the multimedia card (MMC) interface of the OMAP5912 multimedia processor.

1 MMC Overview

The multimedia card/secure data/secure digital IO (MMC/SD/SDIO) host controller provides an interface between a local host, such as a microprocessor unit (MPU) or digital signal processor (DSP), and either an MMC or SD memory card, plus up to four serial flash cards. The host controller handles MMC/SD/SDIO or serial port interface (SPI) transactions with minimal local host intervention. Figure 2 provides an overview of the system.

The host controller supports the following combination of external devices:

- One or more MMC memory cards sharing the same bus, plus up to four devices with 8-bit SPI protocol interface (serial flash memories)
- One SD memory card or SDIO card, plus up to four devices with 8-bit SPI protocol interface

Other combinations, such as two SD cards or one MMC card plus one SD card, are not supported through a single controller.

The application interface manages transaction semantics. The MMC/SD/SDIO host controller handles the MMC/SD protocol at the transmission level, including data packing, adding the cyclic redundancy check (CRC) and start/end bits, and checking for syntactical correctness. It also supports SD mode wide-bus width.

The application interface can send every MMC/SD/SDIO command and either poll for the status of the adapter, wait for an interrupt request, which is sent back in case of exceptions, or warn of the end of the operation. The application interface reads card responses and flag registers, and masks interrupt sources individually. These operations are performed by reading and writing control registers. The MMC/SD/SDIO module also supports two direct memory access (DMA) channels.

Figure 1 shows the OMAP5912 processor with the MMC area highlighted.

Figure 1. OMAP5912 MMC Area

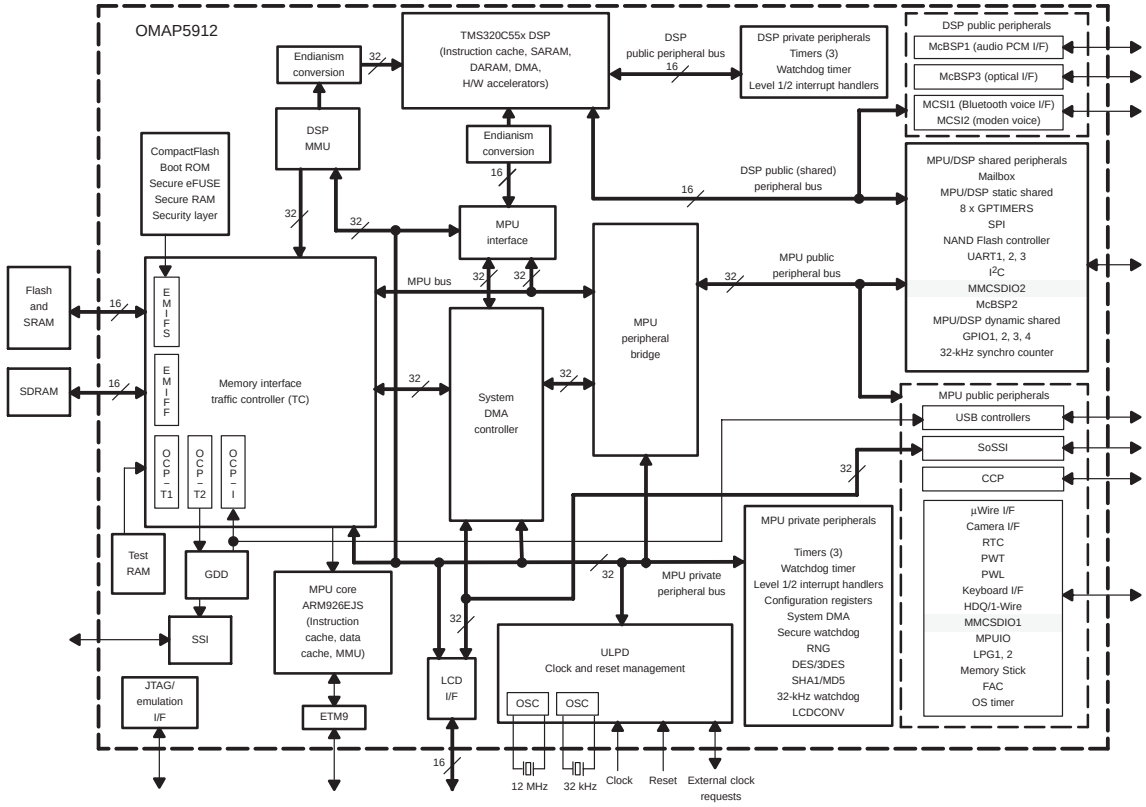
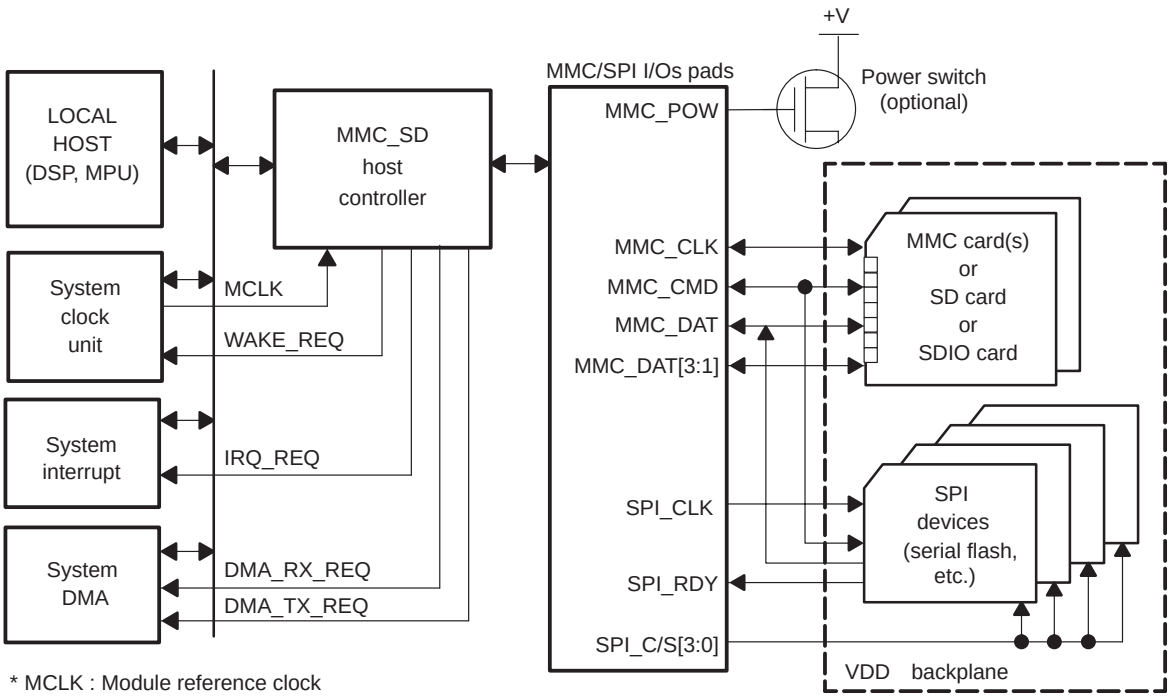


Figure 2. MMC/SD/SDIO System Overview



1.1 MMC/SD/SDIO Host Controller Features

The main features of the controller are:

- ❏ Full compliance with MMC command/response sets as defined in *The Multimedia Card-System Specification*, MMCA Technical Committee, Version 3.1, June 2001
- ❏ Full compliance with SD command/response sets as defined in *SD Memory Card Specification−Part 1, Physical Layer Specification*, SD Group, Version 1.0, March 2000, and *Supplementary Notes−Part 1, Physical Layer Specification*, SD Group, June 2000
- ❏ Full compliance with SDIO command/response sets and interrupt/read-wait mode as defined in *SDIO Card Specification Part E1*, SDIO Working Group, Version 1.0, October 2001
- ❏ Flexible architecture that allows support for new command structure
- ❏ Separate SPI interface with four CS. Provides support for up to four serial flash devices
- ❏ Built-in 64-byte FIFO for buffered read or write

- 16-bit-wide access bus to maximize bus throughput
- Low-power design
- Wide interrupt capability
- Programmable clock generation
- Two DMA channels
- Big- /little-endian mode for data

Known limitations:

- No built-in hardware support for error correction codes (ECC)

1.2 MMC/SD Host Controller Signal Pads

The signal pads listed in Table 1 describe the physical interface between the driving IC (the transceiver) and the target MMC/SD memory cards, SDIO device, or serial flash memories.

The transceiver provides a dc-level adaptation function between the controller core and the target devices. It can be integrated either on-chip with the controller or implemented off-chip (system-dependent issue).

Table 1. Signal Pads

Name	Type	Pull-Up	Reset Value	Description
MMC.CLK	Out	–	0	MMC/SD/SDIO card CLK signal. Only active during active command to MMC/SD/SDIO card using MMC or SPI protocols.
MMC.CMD/SPI.DO (SPI_SO)	In/out	Yes	Input	MMC/SD card CMD signal in MMC/SD mode. SPI serial out signal in SPI modes (output—goes to serial in of target device(s)).
MMC.DAT0 (SPI_SI)	In/out	Yes	Input	MMC card DAT signal or SD/SDIO card DAT[0] signal in MMC/SD mode. SPI serial in signal in SPI modes (input—comes from serial out of target device(s)).
MMC.DAT1 (SDIO_IRQ)†	In/out	Yes	Input	SD/SDIO card DAT[1] signal. Interrupt (IRQ) for SDIO card (SD and SPI protocol).
MMC.DAT2 (SDIO_RW)†	In/out	Yes	Input	SD/SDIO card DAT[2] signal. Read wait (RW) for SDIO card.
MMC.DAT3 (MMC_CS, SD_CD)¶	In/out	Yes	Input	SD/SDIO card DAT[3] signal. Chip-select (CS) for MMC/SD/SDIO cards using SPI protocol. Chip detect (CD) for SD/SDIO cards.
SPI.CLK‡	Out	–	0	SPI CLK signal. Only active in SPI mode during active SPI transfers, except when MMC_CLK is selected.
SPI_C/Sn[3:0]‡	Out	–	b1111	Four SPI chip-select signals. Active low. Only active in SPI mode during active SPI transfers
SPI.RDY‡	In	Yes	Input	SPI ready/busy signal. When low, denotes a busy condition. Only active in SPI mode during active SPI transfers.
MMC_POW§	Out	–	0	MMC/SD cards on/off power supply control. When high, denotes power-on condition.

† Optional signals. Only needed for SD/SDIO cards.
‡ Optional signals. Only needed for devices with SPI interfaces (serial flash, additional MMC, SD, or SDIO cards).
§ Optional signal. Only needed if power supply (VDD) of cards or other SPI devices are to be switched on and off in the application.
¶ Optional signals. Only needed for SD/SDIO cards or for MMC card operated in SPI mode.

1.3 MMC.CLK, SPI.CLK Signals ac Characteristics

The core internally gates the MMC or SPI clock signals to be active only during a valid transaction to the selected target device (memory cards or serial flash). The duty cycle of the clock depends on the clock division factor and the polarity setting.

Figure 3. MMC.CLK/SPI.CLK Signals ac Characteristics

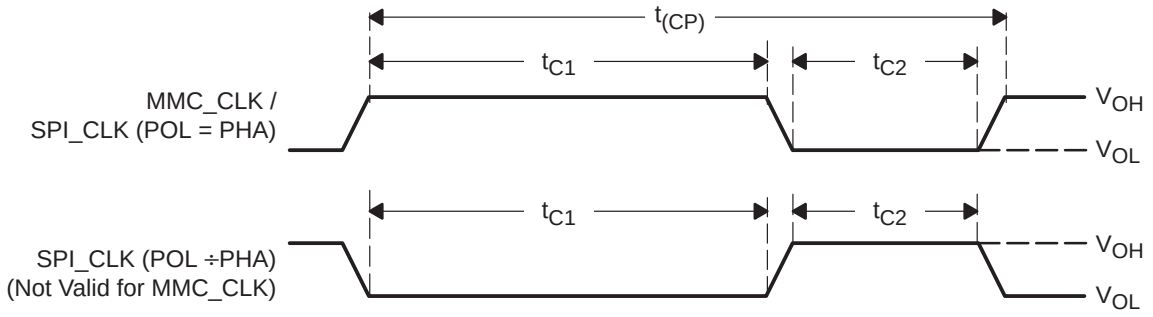


Table 2. MMC.CLK/SPI.CLK Signals ac Parameters

Parameter	Description	Min	Max	Unit
t_{CP}	Clock period	40	–	ns
t_{C1}	MMC mode: Clock high time	–	–	ns
	SPI mode: Clock high time (POL = PHA), clock low time (POL ≠ PHA)			
t_{C2}	MMC mode: Clock low time	–	–	ns
	SPI mode: Clock low time (POL = PHA), clock high time (POL ≠ PHA)			

The clock period and the high and low times specified in Table 2, as well as all timings in subsequent pages, are controller capabilities.

The real clock period must be computed as a function of the system reference clock and adjusted to the target device used in the application. All derived timings must be checked against selected target device specifications.

For example:

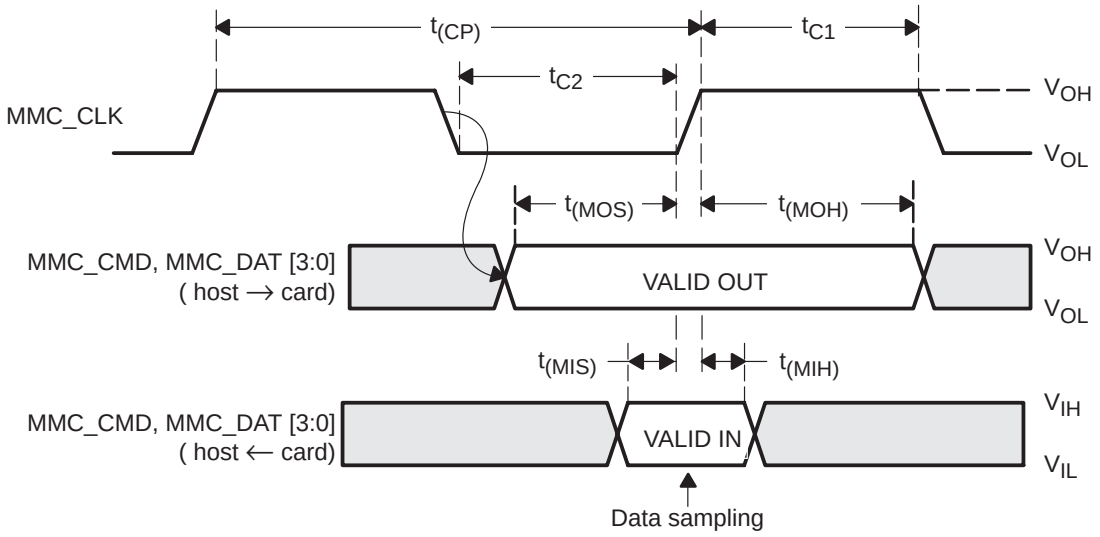
- ❏ MMC card: max 20 MHz (min 50 ns)
- ❏ SD card: max 25 MHz (min 40 ns)
- ❏ SPI serial flash: max 13 MHz (min 77 ns)

1.4 MMC/SD/SDIO Modes—Interface Signal ac Characteristics

Figure 4 depicts the ac characteristics of the interface signals when the interface is configured for MMC/SD/SDIO operation.

SPI-specific output signals (SPI_C/Sn[3:0], SPI.CLK) are held to their inactive state, and SPI-specific input signals are don't care (SPI.RDY).

Figure 4. MMC/SD/SDIO ac Characteristics



Data is sampled on the rising edge of the clock.

Table 3. MMC/SD/SDIO ac Parameters

Parameter	Description	Min	Max	Unit
t_{MOS}	Data output setup to rising edge of clock	t_{C2}–5	–	ns
t_{MOH}	Data output hold to rising edge of clock	t_{C1}–5	–	ns
t_{MIS}	Data input setup to rising edge of clock	4	–	ns
t_{MIH}	Data input hold to rising edge of clock	4	–	ns

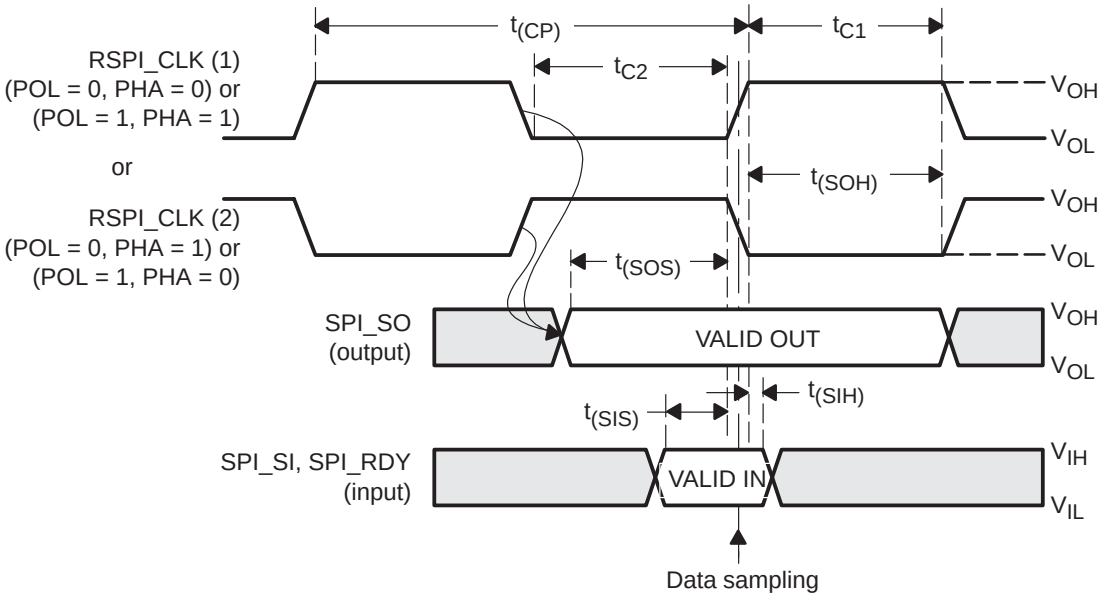
1.5 SPI Mode—Interface Signal ac Characteristics

Figure 5 depicts the ac characteristics of the interface signals when the interface is configured for SPI operation.

MMC-specific input/output signals (MMC.DAT3) are held to their inactive state.

The SPI interface is master only.

Figure 5. SPI ac Characteristics



Data is sampled on the rising or falling edge of the clock, depending on the polarity/phase setting. See Table 4.

Table 4. SPI ac Parameters

Parameter	Description	Min	Max	Unit
t_{SOS}	Data output set up to rising edge of clock (1) or falling edge of clock (2)	t_{C2}–5	–	ns
t_{SOH}	Data output hold to rising edge of clock (1) or falling edge of clock (2)	t_{C1}–5	–	ns
t_{SIS}	Data input set up to rising edge of clock (1) or falling edge of clock (2)	16	–	ns
t_{SIH}	Data input hold to rising edge of clock (1) or falling edge of clock (2)	0	–	ns

2 MMC Registers

Table 5 lists the MMC registers. Table 6 through Table 36 describe the registers bits.

Table 5. MMC Registers

Base Address = FFFB 7800 and 0xFFFB 7C00			
Register	**Description**	**R/W**	**Offset**
MMC.CMD/SPI.DO	MMC command		0x00
MMC_ARGL	MMC argument low		0x04
MMC_ARGH	MMC argument high		0x08
MMC_CON	MMC module configuration		0x0C
MMC_STAT	MMC module status		0x10
MMC_IE	MMC system interrupt enable		0x14
MMC_CTO	MMC command time-out		0x18
MMC_DTO	MMC data read time-out		0x1C
MMC_DATA	MMC data access		0x20
MMC_BLEN	MMC block length		0x24
MMC_NBLK	MMC number of blocks		0x28
MMC_BUF	MMC buffer configuration		0x2C
MMC_SPI	MMC SPI configuration		0x30
MMC_SDIO	MMC SDIO mode configuration		0x34
MMC_SYST	MMC system test		0x38
MMC_REV	MMC module revision		0x3C
MMC_RSP0	MMC/SD command response 0		0x40
MMC_RSP1	MMC/SD command response 1		0x44
MMC_RSP2	MMC/SD command response 2		0x48
MMC_RSP3	MMC/SD command response 3		0x4C
MMC_RSP4	MMC/SD command response 4		0x50
MMC_RSP5	MMC/SD command response 5		0x54
MMC_RSP6	MMC/SD command response 6		0x58
MMC_RSP7	MMC/SD command response 7		0x5C
MMC_IOSR	MMC SDIO suspend/resume control		0x60
MMC_SYSC	MMC system control		0x64
MMC_SYSS	MMC system status		0x68
Reserved	Reserved		0x6C−7C

Table 6. MMC Command Register (MMC_CMD)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x00		
Bit	**Name**	**Description**
15	DDIR	Data direction [write, read]
14	SHR	Stream command or broadcast host response [normal, stream/host]
13:12	TYPE	Command type [bc, bcr, ac, adtc]
11	BUSY	Command with busy response
10:8	RSP	Command responses [no response, R1/R1b, R2, R3, R4, R5, R6, reserved]
7	INAB	Send initialization stream/data abort command
6	ODTO	Card open drain mode/extended command time-out mode
5:0	INDX	Command index [CMD0, …, CMD63]

A write to this register sends a command to the card.

If the local host accesses this register byte-wise, the card receives the command only after a write access to the least significant (LS) byte (bits 7:0). Therefore, the most significant (MS) byte must always be written first in a byte-accessed situation.

A read has no effect except to return the last command that was sent.

Note: Writes

A write into this register with MMC_CMD[TYPE] = adtc resets the FIFO. Writes with other type of values (bc, bcr, ac) do not affect the FIFO contents. Hence, data must be written into the FIFO after sending a single or multiple block-write command.

MMC.CMD/SPI.DO[15] Data Direction (DDIR)

This bit specifies whether the data transfer is a read or a write, and is valid only if the command type is adtc.

This bit has the same polarity as the RD/WR argument bit 0 for a GEN_CMD command (CMD56):

- 0: Data write
- 1: Data read

Value after reset is low.

MMC.CMD/SPI.DO[14] Stream Command or Broadcast Host Response (SHR)

MMC card only.

The SD card does not support stream operation or host-generated response. This bit must be set to 1 in two cases:

- Associated with adtc command type, if the command is a stream data transfer (read or write). Stream read is a class 1 command (CMD11: READ_DAT_UNTIL_STOP). Stream write is a class 3 command (CMD20: WRITE_DAT_UNTIL_STOP).
- Associated with a bc command type, the host generates a 48-bit response instead of a command. It can be used to terminate the interrupt mode by generating a CMD40 response by the core (see Section 4.3, *Interrupt Mode* in *The Multimedia Card-System Specification*). In order for the host response to be generated in open drain mode, MMC_CMD[ODTO] must be set to 1.

This bit is valid only if the command type is adtc or bc:

- 0: Normal mode
- 1: Stream mode (MMC_CMD[TYPE] = adtc), host response (MMC_CMD[TYPE] = bc)

Value after reset is low.

MMC.CMD/SPI.DO[13:12] Command Type (TYPE)

Encoded bits that define the type of command passed by the core to the MMC/SD memory card (see Section 4.7.1, *Command Types*, *The Multimedia Card−System Specification*, or *SD Memory Card Specification−Part 1 Physical*
Layer Specification and *Supplementary Notes−Part 1 Physical Layer Specification).*

- 00: Broadcast commands (bc), no response
- 01: Broadcast commands with response (bcr)
- 10: Addressed commands (ac), no data transfer
- 11: Addressed data transfer commands (adtc) and reset of the FIFO

Value after reset is low (both bits).

MMC.CMD/SPI.DO[11] Command With Busy Response (BUSY)

This bit must be set to 1 if the response to the command sent is type R1b (R1 + busy):

- 0: Response without busy (R1, R2, R3, R4, R5, R6)
- 1: Response with busy (R1b)

Value after reset is low.

MMC.CMD/SPI.DO[10:8] Command Response (RSP)

Encoded bits that define the response for the command that the core passes to the MMC/SD memory card (see Section 4.9, *Responses* in *The Multimedia Card-System Specification*, or *SD Memory Card Specifications−Part 1 Physical Layer Specification* and *Supplementary Notes−Part 1 Physical Layer Specification*).

- 000: No response
- 001: R1/R1b (normal response command)
- 010: R2 (CID, CSD registers)
- 011: R3 (OCR register)
- 100: R4 (fast I/O—MMC card only)
- 101: R5 (interrupt request—MMC card only/IO_RW_DIRECT—SDIO card only)
- 110: R6 (published RCA response—SD card only)
- 111: Reserved

Value after reset is low (3 bits).

MMC.CMD/SPI.DO[7] Send Initialization Stream/Data Abort Command (INAB)

This bit must only be set in two particular cases:

- When the card is idle, to send an initialization sequence. This option simplifies the acquisition of new cards. An initialization sequence consists of setting the CMD line to 1 during 80 clock cycles (see Section 6.3, *Power-Up* in *The Multimedia Card-System Specification*, or Section 6.4, *SD Memory Card Specifications−Part 1 Physical Layer Specification* and

Supplementary Notes−Part 1 Physical Layer Specification SD Memory Card Specifications−Part 1 Physical Layer Specification and *Supplementary Notes−Part 1 Physical Layer Specification*). In this mode, no command is sent to the card and no response is expected.

- When the card is in the data transfer stage, to stop or abort an ongoing data transfer. The card is said to be in such state when the previous command was of type adtc and has not yet completed (MMC_STAT[BRS] = 0). A stop or aborted data command:
  - Freezes the MMC_BLEN[BLEN] and MMC_NBLK[NBLK] values according to the last valid byte written to or read from the card
  - Sets the MMC_STAT[BRS] status bit as follows:
    - 0: No action
    - 1: Initialization (80 clock cycles)/data abort command

Value after reset is low.

MMC.CMD/SPI.DO[6] Card Open Drain Mode/Extended Command Time-Out (ODTO)

This bit has a dual function, depending upon the value set in the MMC_SDIO[XDTS] bit.

- Open drain control function (MMC_SDIO[XDTS] = 0—MMC card only

This bit must be set to 1 if the MMC card bus is operating in open-drain mode during the response phase to the command sent. Typically, it occurs during card identification mode when the card is in idle, ready, or ident state. It is also necessary to set this bit to 1 for a broadcast host response (see MMC.CMD[SHR]). This bit must be set for MMC card commands 1, 2, 3, and 40.

For the SD card, this bit must always be kept low because SD cards do not have open drain capability.

- 0: Push-pull
- 1: Open drain

- Extended command time-out function (MMC_SDIO[XDTS] = 1)—SDIO card only.

This bit must be set to 1 if the SDIO command response requires a long time-out (typically an IO_RW_DIRECT CMD52). When set, the command time-out is set to the data time-out value (see MMC_DTO[DTO]). When clear, the normal command time-out applies (see MMC_CTO[CTO]).

- 0: Command time-out equals CTO
- 1: Command time-out equals DTO

Value after reset is low.

MMC.CMD/SPI.DO[5:0] Command Index (INDX)

Binary encoded value from 0 to 63 specifying the command number sent to the card.

- 000000: CMD0
- 000001: CMD1

  …
- 111111: CMD63

Value after reset is low (all 6 bits).

Table 7. System Argument Low Register (MMC_ARGL)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x04		
Bit	**Name**	**Description**
15:0	ARGL	Command argument bits [15:0]

Value after reset is low (all 16 bits).

Table 8. System Argument High Register (MMC_ARGH)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x08		
Bit	**Name**	**Description**
15:0	ARGH	Command argument bits [31:16]

Value after reset is low (all 16 bits).

Table 9 lists the module configuration characteristics.

These two 16-bit registers (Table 7 and Table 8) specify the 32-bit argument value that is passed with the command. These registers must be initialized before sending the command to the card (write action into the MMC_CMD register). The only exception making a write unnecessary is a command index specifying stuff bits in arguments.

This section describes the module configuration register.

Table 9. Module Configuration Register (MMC_CON)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x0C		
Bit	**Name**	**Description**
15	DW	Data bus width
14	–	Reserved
13:12	MODE	Operating mode select [MMC/SD, SPI, SYSTEST]
11	POWER_UP	Power-up control
10	BE	Big-endian mode [little, big]
9:0	CLKD	Clock divider [disabled, 1:1023]

Note: Active Transfer Phase

This register must never be written during an active transfer phase. Changing it may result in unpredictable behavior.

MMC_CON[15] Bus Width During Data Phase (DW)

SD mode only.

This bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the bus width supported by the SD card.

- 0: 1-bit data width (DAT[0] used)
- 1: 4-bit data width (DAT[3:0] used, SD card only)

Value after reset is low.

This bit must always be set to 0 for MMC cards or during SPI transfer. Failing to set this bit correctly results in unpredictable behavior.

MMC_CON[13:12] Mode Select (MODE)

These two bits select among MMC/SD, SPI, and SYSTEST modes.

- In MMC/SD mode, transfers to the MMC/SD/SDIO card follow the MMC protocol. The MMC clock is enabled and the SPI clock is disabled. MMC/SD transfers are operated under the control of the MMC.CMD register.

- In SPI mode, transfers to as many as four SPI controlled devices (serial flash, MMC/SD/SDIO cards) are supported. SPI transfers are operated under the control of the MMC_SPI register.
- In SYSTEST mode, the signal pins are configured as general-purpose input/output, and the 64-byte FIFO is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output, or in-out). The YSTEST mode is operated under the control of the MMC_SYST register.
  - 00: MMC/SD mode (MMC/SD/SDIO cards using MMC/SD protocol)
  - 01: SPI mode (for serial flash or others SPI slave devices)
  - 10: SYSTEST mode
  - 11: Reserved

Value after reset is low (both bits).

MMC_CON[11] Power-Up Control (POW)

This bit must be set to 1 before any valid transaction to either MMC/SD or SPI memory cards.

When 1, the card is considered powered-up and the controller core is enabled.

When 0, the card is considered powered-down (system dependent), and the controller core logic is in pseudo-reset state. This is, the MMC_STAT flags and the FIFO pointers are reset, any access to MMC_DATA[DATA] has no effect, a write into the MMC.CMD register is ignored, and a setting of MMC_SPI[STR] to 1 is ignored.

This bit directly controls the MMC_POW signal (if implemented as device pin).

- 0: Powered-down/pseudo-reset state.
- 1: Powered-up/normal operation mode.

Value after reset is low.

MMC_CON[10] Big Endian (BE)

When this bit is 0 (default), the FIFO is accessed in little endian format. In transmit mode, the LS byte (MMC_DATA[7:0]) is transmitted first, and the MS byte (MMC_DATA[15:8]) is transmitted to the card in second position. Conversely, in receive mode, the first or odd byte received (1, 3, 5, …) is stored in the LS byte position, and the second or even byte received is stored in the MS byte position.

When the LH sets this bit to a 1, the FIFO is accessed in big endian format. In transmit mode, the MS byte (MMC_DATA[15:8]) is transmitted first and the LS byte (MMC_DATA[7:0]) is transmitted to the card in second position. Conversely, in receive mode, the first or odd byte received (1, 3, 5, …) is stored in the MS byte position, and the second or even byte received is stored in the LS byte position.

- 0: Little-endian mode
- 1: Big-endian mode

Value after reset is low.

MMC_CON[9:0] Clock Divider (CLKD)

These bits define the ratio between a reference clock frequency (system dependent) and the output clock frequency on the CLK pin of either the memory card (MMC or SD) or other 8-bit mode SPI-controlled device.

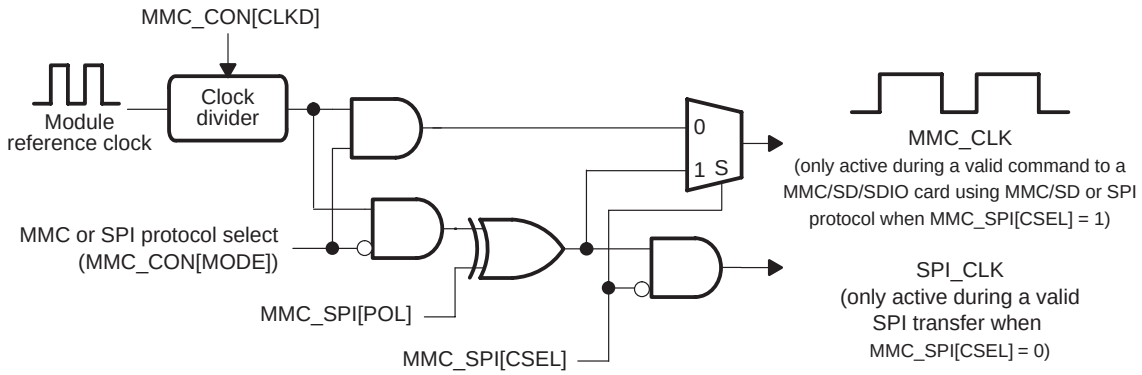
The division factor is equal to the binary encoded decimal value for values between 1 and 1023. A value of 0 disables the clock.

- 0x00: Clock disabled
- 0x01: Ref clock/1

  ....
- 0x3FF: Ref clock/1023

Value after reset is low (all 10 bits).

Figure 6. Clock Control



Notes: 1) During the identification phase, the maximum frequency on the MMC CLK line is 400 kHz (see Section 6.7, *Bus Timing* in *The Multimedia Card-System Specification*, or Section 6.8, *SD Memory Card Specifications−Part 1, Physical Layer Specification* or *Supplementary Notes−Part 1, Physical Layer Specification*). A value of 50 must be set into the frequency ratio register if the reference clock frequency to the MMC/SD host controller is 20 MHz.

2) During the data transfer phase, the maximum frequency is 20 MHz for MMC card and 25 MHz for SD cards.

3) The duty cycle of the generated MMC.CLK and SPI.CLK signals depends on the clock divider value (MMC_CON[CLKD]) and on the polarity setting (MMC_SPI[POL]) in SPI mode only. The low and high times approximate values can be computed using rules set in Table 10.

4) In MMC/SD mode, the idle value of MMC.CLK signal is low. In SPI mode, the idle value of either the MMC.CLK (MMC_SPI[CSEL] = 1) or the SPI.CLK (MMC_SPI[CSEL] = 0) is a function of the polarity setting (low if MMC_SPI[POL] = 0, high if MMC_SPI[POL] = 1).

Table 10. MMC.CLK/SPI.CLK High/Low Time Computation

MMC_CON[CLKD]	MMC.CLK/SPI.CLK High Time	MMC.CLK/SPI.CLK Low Time
1	REF_CLK_HIGH_TIME	REF_CLK_LOW_TIME
Even ≥ 2	REF_CLK_PER (CLKD/2)	REF_CLK_PER (CLKD/2)
Odd ≥ 3 (POL = PHA)	REF_CLK_PER (TRUNC[CLKD/2] + 1)	REF_CLK_PER (TRUNC[CLKD/2])
Odd ≥ 3 (POL ≠ PHA)	REF_CLK_PER (TRUNC[CLKD/2])	REF_CLK_PER (TRUNC[CLKD/2] + 1)

Notes:

1) REF_CLK_PER is the reference clock period (in ns) to the module (end-system dependent).

2) TRUNC is the truncate function to an integer number (round down).

Example 1:

Module reference clock = 48 MHz (20.83 ns); target is the MMC card.

1) a) (MMC_CON[CLKD] = 3 (because the MMC card is 20 MHz max)

2) b) MMC_CLK period = 62.5 ns (> 50 ns OK)

3) c) Ideal MMC_CLK high time = 41.66 ns (>>10 ns OK)

4) d) Ideal MMC_CLK low time = 20.83 ns (>>10 ns OK)

Example 2:

Module reference clock = 60 MHz (16.67 ns); target is the 13-MHz serial flash requiring polarity 1 programming.

1) a) (MMC_CON[CLKD] = 5 (because 13-MHz serial flash)
2) b) SPI_CLK period = 83.33 ns (> 77 ns OK)
3) c) Ideal SPI_CLK high time = 33.3 ns (<35 ns FAIL, use (MMC_CON[CLKD] = 6)
4) d) Ideal SPI_CLK low time = 50 ns (>>35 ns OK)
5) e) (MMC_CON[CLKD] = 6 (because clock high time min 35 ns)
6) f) SPI_CLK period = 100 ns (> 77 ns OK)
7) g) Ideal SPI_CLK high time = 50 ns (>>35 ns OK)
8) h) Ideal SPI_CLK low time = 50 ns (>>35 ns OK)

Table 11. Module Status Register (MMC_STAT)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x10		
Bit	**Name**	**Description**
15	−	Reserved
14	CERR	Card status error in response
13	CIRQ	MMC card IRQ received (following CMD40) or SDIO card interrupt
12	OCRB	Operation condition register (OCR) busy (following CMD1 or ACMD41)
11	AE	Buffer almost empty
10	AF	Buffer almost full
9	CRW	Card read wait
8	CCRC	Command CRC error
7	CTO	Command response time-out (no response)
6	DCRC	Data CRC error
5	DTO	Data response time-out (no response)
4	EOFB	Card exit busy state
3	BRS	Block received/sent
2	CB	Card enter busy state
1	CD	Card detect on DAT3
0	EOC	End of command phase

The following is common to all bits:

- ❑ The local host can clear a set bit location only by writing a 1 into that bit location. Writing 0 has no effect.
- ❑ When the core sets a bit location to 1, the local host receives an interrupt signal if the interrupt was enabled.

MMC_STAT[14] Card Status Error (CERR)

Table 12. Card Status Error (CERR)

Response Type	Card Status Bits With Error	Response Register Significant Bits	Comments
R1 (MMC, SD, SDIO)	31−26, 24−16,	MMC_RSP7[15−10, 8−0]†	Bit 4 if MMC_SDIO[C14E] = 1 (SDIO)
	4[2] 3[2], 2[2] (opt)	MMC_RSP6[4, 3, 2]‡	Bit 3 if MMC_SDIO[C13E] = 1 (SD/app spec)
			Bit 2 if MMC_SDIO[C12E] = 1 (app spec)
R6 (SD, SDIO)	15−13, 3	MMC_RSP6[15−13, 3]	These correspond to 23, 22, 19, 3 card status errors (SDIO card does not generate error on bit 3 force 0—superset)
R5 (SDIO)	7, 6, 3, 1, 0[3]	MMC_RSP6[15,14, 11, 9, 8]	[3] Only if MMC_SDIO[15] = 1

† These 15 bits can all generate errors (SDIO spec specifies 31, 23−22, 19 while others are 0—superset).
‡ These 3 bits can also generate errors if enabled.

MMC/SD mode only.

The core automatically sets this bit when there is at least one error in a response of type R1, R1b, R6, or R5 if enabled. Only bits referenced as type E (error) set a card status error.

The error handler must parse the response registers to understand the source of the error.

Other responses (type R2, R3, or R4) do not trigger a card status error.

This bit has no meaning and always reads 0 in SPI or SYSTEST modes.

- 0: No action or no error
- 1: Error occurred

Value after reset is low.

MMC_STAT[13] Card IRQ (CIRQ)

MMC card only:

The core automatically sets this bit when a card is in interrupt mode and exits Wait_IRQ state (irq) by asserting a low level on the CMD line (cards are in open-drain mode). Only Class 9 MMC cards can be put into interrupt mode when in standby state using a GO_IRQ_STATE (CMD40) command.

SDIO card only:

The core automatically sets this bit when a SDIO card has signaled an interrupt on DAT1 line and if the MMC_SDIO[IRQE] bit was set to 1. The interrupt condition is detected in either 1-bit or 4-bit transfer mode and for either MMC/SD or SPI operation mode. SD memory cards do not support interrupt mode.

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- ❏ 0: No action or idle
- ❏ 1: Card exits IRQ state (MMC card), card interrupt detected (SDIO card).

Value after reset is low.

MMC_STAT[12] OCR Busy (OCRB)

MMC/SD mode only:

The core automatically sets this bit after a SEND_OP_COND (CMD1) or a SD_APP_OP_COND (ACMD1) command when one or more cards have not yet completed power-up. When this bit is set, the CMD1/ACMD1 command must be repeated until the card stops responding with a busy condition (a low value on bit 31 of OCR register indicates a busy condition) (see Section 6.3, *Power-Up*, in *The MultiMediaCard−System Specification* or Section 6.4, *SD Memory Card Specifications−Part 1, Physical Layer Specification* or *Supplementary Notes−Part 1, Physical Layer Specification*).

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- ❏ 0: No action or card powered up
- ❏ 1: OCR busy

Value after reset is low.

MMC_STAT[11] Buffer Almost Empty (AE)

The core automatically sets this bit during a write operation to the card (see class 4 block-oriented write command in Section 4.7.3, *Command Classes,* in *The MultiMediaCard−System Specification−Part 1, Physical Layer Specification, SD Group June 2000*) when the level equals or is below the threshold value (in 16-bit words) set in MMC_BUF[AEL]. It indicates that the memory card has emptied the buffer to the specified level and that the local host is able to write more data into the buffer.

If the DMA transmit mode is enabled, this bit is never set. Instead, a DMA TX request is generated to the system's main DMA controller.

Note: DMA TX Request

The almost-empty status bit and DMA TX request are generated under the same conditions. This bit is set initially when a new block write command is sent to the card. Once the bit is set, the core internally masks a new set condition until the local host has performed MMC_BUF[AEL] 16-bit word write access(es) to the FIFO.

- 0: No action or buffer is equal or above almost-empty level
- 1: Buffer almost empty

Value after reset is low.

MMC_STAT[10] Buffer Almost Full (AF)

The core automatically sets this bit during a read operation to the card (see class 2 block-oriented read commands in Section 4.7.3, *Command Classes,* in *The MultiMediaCard−System Specification−Part 1, Physical Layer Specification, SD Group June 2000*) when the level equals or is above the threshold value (in 16-bit words) set in MMC_BUF[AFL]. It indicates that the memory card has filled out the buffer to the specified level and that the local host needs to empty the buffer by reading it.

If the DMA-receive mode is enabled, this bit is never set. Instead, a DMA RX request is generated to the system's main DMA controller.

Note: DMA RX Request

The almost full status bit and DMA RX request are generated under the same conditions. Once set, the core internally masks a new set condition till the local host has performed MMC_BUF[AFL] 16-bit word read access(es) from the FIFO.

- 0: No action or buffer is below or equal almost full level
- 1: Buffer almost full

Value after reset is low.

MMC_STAT[9] Card Read Wait (CRW)

SDIO card only.

The core automatically sets this bit when an SDIO card has entered read wait. It indicates that the previous read multiple transfer (CMD53) has been temporarily stalled and that a new command without data stage (such as CMD52) can be sent to the SDIO card.

This bit is set on the condition that the core has requested a wait to the card (MMC_SDIO[RW] = 1) and the read wait mode is enabled (MMC_SDIO[RWE] = 1). The read wait condition is detected in either 1- or 4-bit transfer mode.

- ❏ 0: No action
- ❏ 1: SDIO card in read wait

Value after reset is low.

MMC_STAT[8] Command CRC Error (CCRC)

MMC/SD mode only.

The core automatically sets this bit if there is a CRC7 error in the command response (bits 7:1 of response). CRC7 is checked for all command response types (R1 through R6) with the exception of type R3, and conditionally for type R4 if MMC_SDIO[DCR4] = 1.

In SPI or SYSTEST modes, this bit has no meaning and always reads 0.

- ❏ 0: No action or no CRC7 error
- ❏ 1: CRC7 error

Value after reset is low.

MMC_STAT[7] Command Time-Out Error (CTO)

MMC/SD mode only.

The core automatically sets this bit if the card does not respond to any command requiring a response within the specified number of command time-out clock cycles set in MMC_CTO[CTO].

Note: Command Time-Out

If this bit is set after a command time-out, clearing this bit automatically stops the MMC clock and forces the controller FSM to its default state.

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- ❏ 0: No action or no command time-out
- ❏ 1: Command time-out

Value after reset is low.

MMC_STAT[6] Data CRC Error (DCRC)

MMC/SD mode only.

The core automatically sets this bit if there is a CRC16 error in the data phase response following a block read command (single or multiple), or if a 3-bit CRC status differs from a positive 010 token during a block-write command (single or multiple). A token error can be either a data transmission error 101, or a no CRC response 111 in the case of a programming error (SD card only). Every block of the CRC is checked in a multiple-block transfer.

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- ❏ 0: No action or no CRC error
- ❏ 1: CRC16 error (read), 3-bit CRC token error (write)

Value after reset is low.

MMC_STAT[5] Data Time-Out Error (DTO)

The core automatically sets this bit if the card does not respond within the specified number of data time-out clock cycles (DTO) set in MMC_DTO[DTO].

This bit is also set in SPI mode if the RDY/BUSY signal remains asserted in busy condition for MMC_DTO[DTO] consecutive clock cycles.

Note: Data Time-Out

If this bit is set after a data time-out, clearing this bit automatically stops the MMC or SPI clock and forces the controller FSM to its default state.

This bit has no meaning and always reads 0 in SYSTEST mode.

- ❏ 0: No action or no data time-out
- ❏ 1: Data time-out

Value after reset is low.

MMC_STAT[4] Card Exit Busy State (EOFB)

MMC/SD mode only.

The core automatically sets this bit when the addressed card releases the DAT line from its busy state (low level = busy). This bit can only be set during a programming phase (write operation) to an MMC or SD memory card.

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- 0: No action
- 1: Data line released/exit busy state

Value after reset is low.

MMC_STAT[3] Block Received/Sent (BRS)

The core automatically sets this bit at the end of a block transfer (read or write).

In MMC or SD mode, this bit is set when the block transfer completes without error. If a CRC error occurs, this bit is not set. Instead, a data CRC error is set to 1. For either multiple block or stream transfer, this bit is set only once after the last successful block transfer (when MMC_NBLK[NBLK] decrements down to 0) or until interrupted by a stop command.

In SPI mode, this bit is set when either the read or write command completes *(MMC_BLEN[BLEN] decrements down to 0).*

The difference between a DMA and a non-DMA receive operation are:

- In non-DMA RX mode, this bit is set after the last byte has been received in the FIFO. At this stage, the FIFO is not empty and must be read by the LH until it is emptied before sending a new command.
- In DMA RX mode, this bit is set after both the last byte has been received and the FIFO is empty.

This bit has no meaning and always reads 0 in SYSTEST mode.

- 0: No action
- 1: Block received/sent

Value after reset is low.

MMC_STAT[2] Card Enter Busy State (CB)

MMC/SD mode only.

The core automatically sets this bit when the addressed card asserts the DAT line to a low level during a programming phase (write operation) to an MMC or SD memory card. For an MMC card only, users can optionally use this interrupt to deselect the card, which continues to program, and select another card.

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- 0: No action
- 1: Data line asserted low/card busy

Value after reset is low.

MMC_STAT[1] Card Detected on DAT3 (CD)

MMC/SD mode only.

The core automatically sets this bit after it has detected a card-detect condition on the DAT3 line and if MMC_SDIO[CDE] = 1.

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- 0: No action
- 1: Card-detect event

Value after reset is low.

MMC_STAT[0] End of Command (EOC)

MMC/SD mode only.

The core automatically sets this bit at the end of a successful command/response sequence or at the end of a command without response. This bit is not set in case of a card status error (MMC_STAT[CERR] = 1), command CRC error (MMC_STAT[CCRC] = 1), or command time-out (MMC_STAT[CTO] = 1).

This bit has no meaning and always reads 0 in SPI or SYSTEST mode.

- 0: No action
- 1: End of command/response sequence

Value after reset is low.

Table 13 lists the characteristics of the system interrupt enable register.

Table 13. System Interrupt Enable Register (MMC_IE)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x14		
Bit	**Name**	**Description**
15	-	Reserved
14	CERR	Card status error interrupt enable
13	CIRQ	Card IRQ interrupt enable
12	OCRB	OCR busy interrupt enable
11	AE	Buffer almost empty interrupt enable
10	AF	Buffer almost full interrupt enable
9	CRW	Card read wait enable
8	CCRC	Command CRC error interrupt enable
7	CTO	Command response time-out interrupt enable
6	DCRC	Data CRC error interrupt enable
5	DTO	Data response time-out interrupt enable
4	EOFB	Card exit busy state interrupt enable
3	BRS	Block received/sent interrupt enable
2	CB	Card enter busy state interrupt enable
1	CD	Card-detect interrupt enable
0	EOC	End of command interrupt enable

Common to all bits:

When the local host sets a bit location to 1, it is notified of an interrupt if the core asserted the corresponding bit location in MMC_STAT register to 1.

If set to 0, the interrupt is masked and the local host is not signaled.

- ❏ 0: Interrupt disabled
- ❏ 1: Interrupt enabled

Value after reset is low (all bits).

Table 14. Command Time-Out Register(MMC_CTO)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x18		
Bit	**Name**	**Description**
15:8	–	Reserved
7:0	CTO	MMC command time-out value

This 16-bit register specifies the maximum number of clock cycles before a command time-out condition occurs.

MMC_CTO[7:0] Command Time-Out Value (CTO)

MMC/SD mode only.

The local host sets this field based on N_{CR} clock cycles. The MMC and SD cards specify N_{CR} to be between 2 and 64 clock cycles.

If the card does not respond within the specified number of cycles, command time-out is set to 1 in the MMC_STAT[CTO] register bit.

For MMC card-interrupt mode support, this time-out is disabled when the command passes with an R5 response (CMD40) if register bit MMC_SDIO[C5E] = 0.

For the SDIO card, this time-out value is not valid for command passes with MMC_CMD[ODTO] = 1 if MMC_SDIO[XDTS] = 1; MMC_DTO[DTO] applies instead.

- 0x00: Command time-out disabled
- 0x01: 1 clock cycle

  …
- 0xFF: 255 clock cycles (2^8–1)

Value after reset is low (all 8 bits).

Table 15 lists the characteristics of the data read time-out register.

Table 15. Data Read Time-Out Register (MMC_DTO)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x1C		
Bit	**Name**	**Description**
15:0	DTO	Data read time-out

This 16-bit register specifies the maximum number of clock cycles before a data time-out condition occurs.

MMC_DTO[15:0] Data Time-Out Value (DTO)

In MMC/SD mode, the local host must set this field based on N_{AC} clock cycles. N_{AC} is computed from the parameters TAAC and NSAC and the operating clock frequency. TAAC and NSAC are CSD card parameters and are obtained by reading the response register after successful execution of a SEND_CSD command (CMD9).

If the card does not respond within the specified number of cycles, data time-out is set to 1 in the MMC_STAT[DTO] register bit.

The effective number of clock cycles for time-out value are to be multiplied by 1024 if MMC_SDIO[DPE] = 1 and by 1 if MMC_SDIO[DPE] = 0.

For the SDIO card, this time-out value is also valid for command without data stage passes with MMC_CMD[ODTO] = 1 and MMC_SDIO[XDTS] = 1.

In SPI mode, a data time-out condition is also generated if the RDY/BUSY signal is asserted low (BUSY) for DTO consecutive clock cycles.

Table 16. Clock Cycles for Time-out Value

MMC_DTO[DTO]	MMC_SDIO[DPE] = 0	MMC_SDIO[DPE] = 1	Units
0x0000	No time-out	No time-out	
0x0001	1	1024	MMC clock cycles
0x0002	2	2048	“
…	…	…	“
0xFFFF	65535 (2^{16}–1)	67107840 (2^{26}–2^{10})	“

Value after reset is low (all 16 bits).

Table 17. Data Access Register (MMC_DATA)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x20		
Bit	**Name**	**Description**
15:0	DATA	Transmit/receive FIFO data

This register is the entry point for the local host to read data from, or write data into, the FIFO buffer. The FIFO size is 32x16 bits (64 bytes). Bytes within a word are stored and read in little endian format (MMC_CON[BE] = 0) or big endian format (MMC_CON[BE] = 1).

If the local host accesses this register byte-wise, the MS byte (bits [15:8]) must always be written/read first. A byte access to the LS byte without a prior write into the MS byte results in the MS byte being filled with 0x00.

MMC_DATA[15:0] Transmit/Receive FIFO Data Value (DATA)

In MMC/SD mode, this register contains either the data packet associated with block transfer (read or write), the CID contents for a PROGRAM_CID (CMD26) command, or the CSD contents for a PROGRAM_CSD (CMD27) command.

Because the block length is passed as an argument, the local host can perform only 16-bit access (read or write) into the buffer even if the block length is not an even number. In case of an odd number of bytes to read, the upper byte of the last access always reads as 0x00. Conversely, for an odd number of bytes to write, the upper byte must be filled with 0x00 for the last data value.

In SPI mode, the register contains both the command (opcode and address for a serial flash) and the data.

In SYSTEST mode, the FIFO behaves as a stack accessible only by the local host (push and pop operations). In this mode, the set FIFO threshold values are active, as are the associated interrupts and DMA if enabled. This special mode can be used for system test purposes.

Value after reset is low (all 16 bits).

Note: Read/Write Access

A read access when the buffer is empty returns the previous read data value. A write access when the buffer is full is ignored. In both events, the FIFO pointers are not updated and a remote access error (hardware error) is generated (access qualifier). No remote error is generated when the local host performs a 16-bit access if the buffer contains a single byte. A debugger read access (SUSPEND operation) returns the current read data value but does not update the FIFO pointers or generate an access error.

Figure 7 shows the behavior of the data access to/from the FIFO as a function of the MMC_CON[BE] bit value.

Figure 7. Little/Big Endian Mode FIFO Access

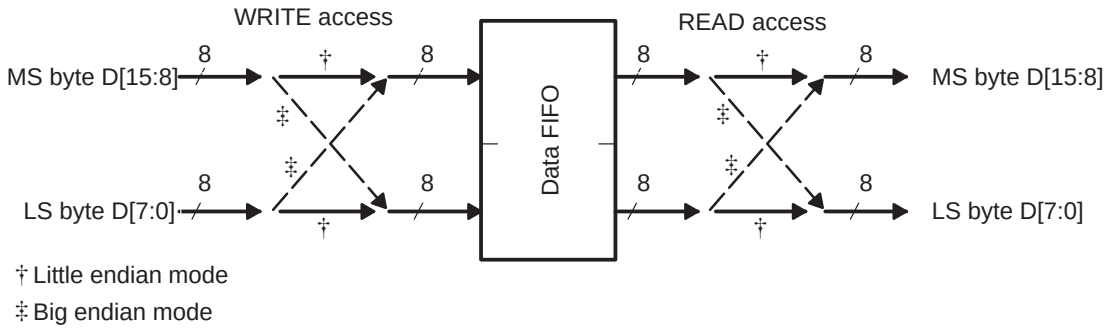


Table 18. Block Length Register (MMC_BLEN)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x24		
Bit	**Name**	**Description**
15:11	−	Reserved
10:0	BLEN	Block length value

This register (Table 18) configures the core for the number of bytes to read or write. It must be initialized at least once before starting an MMC, SD, or SPI block data transfer (read or write).

MMC_BLEN[10:0] Block Length (BLEN)

A write into this register initializes an 11-bit counter that decrements by 1 after each byte is transferred. A read of register returns the number of bytes remaining to be transferred. When the counter reaches 0 and the last byte transfer completes, the core automatically reloads the block length to its programmed value, except in the case of an aborted/stopped data transfer.

In MMC/SD mode, this 11-bit value specifies the data block length. This value must be set with max $2^{READ\_BL\_LEN}-1$ for a block read or max $2^{WRITE\_BL\_LEN}-1$ for a block write, respectively.

READ_BL_LEN and WRITE_BL_LEN are CSD register settings of the card returned in an R2 response following a SEND_CSD command (CMD9).

In SPI mode and for a read transaction, the block length must be initialized with the exact byte count to read −1, excluding the opcode and address arguments.

Opcode and address arguments passed to the SPI device must be written into the FIFO buffer before starting the SPI transfer. BLEN starts to decrement as soon as the buffer contents have been shifted out to the SPI device. The buffer then starts to be filled with the received data from the SPI device.

In SPI mode and for a write transaction, the block length must be initialized with the exact byte count to write −1, including the number of bytes needed to pass the opcode and address arguments.

It is recommended to have opcodes and address that are passed to the SPI module written into the FIFO buffer before starting the SPI transfer. This allows the DMA write operation to access only the data portion. The block length starts to decrement for every byte shifted out to the SPI device.

- 0x000: 1 byte

  …

- 0x7FF: 2048 bytes

Value after reset is low (all 11 bits).

Table 19. Number of Blocks Register (MMC_NBLK)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x28		
Bit	**Name**	**Description**
15:11	−	Reserved
10:0	NBLK	Number of blocks value

This register (Table 19) configures the number of blocks for a multiple block data transfer (read or write) operation for MMC/SD cards. This register is not used for SPI transfers.

MMC_NBLK[10:0] Number of Blocks (NBLK)

MMC/SD mode only.

In MMC/SD mode, this 11-bit value specifies the number of blocks for a multiple block data transfer (read or write). This value must be set with the number of blocks −1. Note that each block is of size MMC_BLEN[BLEN].

This register must be programmed before any multiple block data transfer. A write into this register initializes an 11-bit counter that decrements by 1 after each block transfer. A read of this register returns the number of blocks remaining to be transferred to the card. When the counter reaches 0, the transfer stops after the last transfer completes.

For stream or multiple block transfers, a block received/sent interrupt is generated only once after the last successful transfer when MMC_NBLK[NBLK] reaches 0. In stream mode, once the block receive/sent interrupt is received, MMC_NBLK[NBLK] can be reprogrammed to continue the transfer from the point it was interrupted.

Note: Value Requirement

This value must be 0x000 for a single block transfer. In stream mode, the minimum allowable number of blocks is 2. Finally, if the transfer is interrupted by a STOP_TRANSMISSION command (CMD12) before the counter reaches 0, reprogram this register before starting any new single- or multiple-block data transfer.

- 0x000: 1 block

  …

- 0x7FF: 2048 blocks

Value after reset is low (all 11 bits).

Table 20. Buffer Configuration Register (MMC_BUF)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x2C		
Bit	**Name**	**Description**
15	RXDE	Receive DMA channel enable
14:13	–	Reserved
12:8	AFL	Buffer almost full level
7	TXDE	Transmit DMA channel enable
6:5	–	Reserved
4:0	AEL	Buffer almost-empty level

This register configures the buffer threshold level of the 32 16-bit-word FIFO entries and enables DMA transfers.

MMC_BUF[15] Receive DMA Channel Enable (RXDE)

When this bit is set to 1, the receive DMA channel is enabled and the core forces the MMC_STAT[AF] status bit to 0 regardless of the buffer almost-full level setting (MMC_BUF[AFL]). See *Buffer Almost Full (AFL)*; for more detail on DMA operations, see Section 4, *DMA Operations*.

- 0: Receive DMA channel disabled
- 1: Receive DMA channel enabled

Value after reset is low.

MMC_BUF[12:8] Buffer Almost Full Level (AFL)

This register holds the programmable almost-full level value used to determine the almost-full buffer condition. If users want an interrupt or a DMA read request to be issued during a read operation when the data buffer holds *at least* n 16-bit words, the buffer MMC_BUF[AFL] must be set with n−1.

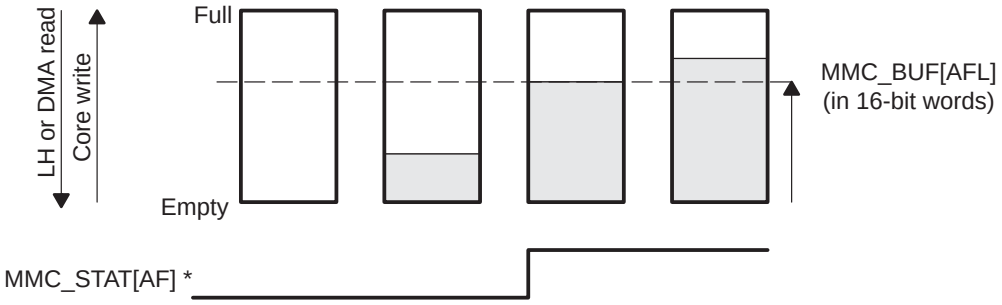
- 0x00: 1 16-bit word (2 bytes)

  …

- 0x1E: 31 16-bit words (62 bytes)
- 0x1F: 32 16-bit words (64 bytes)

Value after reset is low (all 5 bits).

Figure 8. Buffer Almost Full Level (AFL)



* Non-DMA mode only. In DMA mode, the DMA TX request is asserted to its active level under identical conditions.

MMC_BUF[7] Transmit DMA Channel Enable (TXDE)

When this bit is set to 1, the transmit DMA channel is enabled and the core forces the MMC_STAT[AE] status bit to 0 regardless of the buffer almost-empty level (MMC_BUF[AEL]). See *Buffer Almost Empty (AEL)*; for more detail on DMA operations, see Section 4, *DMA Operations*.

- 0: Receive DMA channel disabled
- 1: Receive DMA channel enabled

Value after reset is low.

MMC_BUF[4:0] Buffer Almost Empty Level (AEL)

This register holds the programmable almost-empty level value used to determine an almost-empty buffer condition. If users want an interrupt or a DMA write request to be issued during a write operation when the data buffer is able to receive n words of 16 bits, then MMC_BUF[AEL] must be set with n−1.

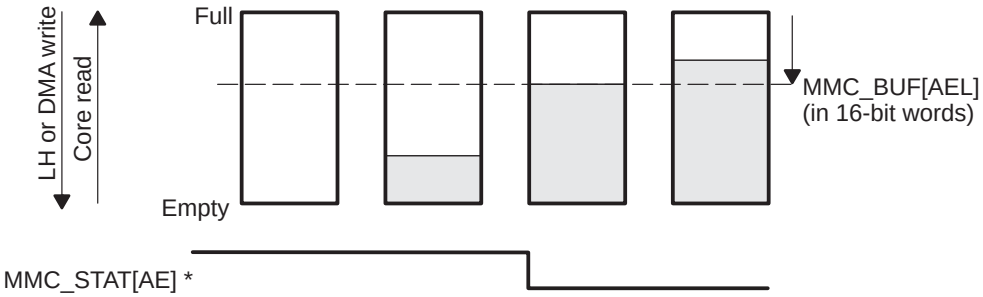
- 0x00: 1 16-bit word (2 bytes)

  …

- 0x1E: 31 16-bit words (62 bytes)
- 0x1F: 32 16-bit words (64 bytes)

Value after reset is low (all 5 bits).

Figure 9. Figure 1: Buffer Almost-Empty Level (AEL)



* Non−DMA mode only. In DMA mode, the DMA RX request is asserted to its active level under identical conditions.

Table 21. SPI Configuration Register (MMC_SPI)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x30		
Bit	**Name**	**Description**
15	STR	Start SPI transfer
14	WNR	Write/not read
13	SODV	SPI_SO serial out pin default value
12	CSTR	SPI transfer controlled start
11:10	TCSH	Chip-select hold time control
9:8	TCSS	Chip-select setup time control
7	CSEL	Card socket connector select
6	−	Reserved
5:4	CS	Chip-select control
3	CSM	Chip-select mode
2	CSD	Chip-select disable

Table 21. SPI Configuration Register (MMC_SPI) (Continued)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x30		
Bit	**Name**	**Description**
1	PHA	Phase control
0	POL	Polarity control

This register is used to configure the SPI interface and start an SPI transfer if the SPI mode has been enabled.

MMC_SPI[15] Start SPI Transfer (STR)

Set only bit.

This bit can be set only if MMC_SPI[CSTR] = 0. This bit always reads as 0. A write to 0 has no effect.

When the local host sets the bit to 1, an SPI transfer automatically starts. Users must initialize MMC_BLEN[BLEN] before starting an SPI transfer.

The SPI transfer automatically stops when the size programmed in MMC_BLEN[BLEN] decrements to 0 (in read and in write).

- 0: No action
- 1: SPI transfer starts (condition: MMC_SPI[CSTR] = 0)

Value after reset is low.

MMC_SPI[14] Write /Not Read (WNR)

This bit instructs the 11-bit block length counter (MMC_BLEN[BLEN]) to decrement either on byte read when MMC_SPI[WNR] = 0 or on byte write when MMC_SPI[WNR] = 1.

- 0: Decrement on byte received
- 1: Decrement on byte sent

Value after reset is low.

MMC_SPI[13] Serial-Out Default Value (SODV)

This bit determines the default value for the serial-out signal before SPI start and during the data phase of a SPI read transfer. This bit must be set to 1 for MMC/SD/SDIO cards using SPI protocol.

- 0: SPI_SO default value = 0
- 1: SPI_SO default value = 1

Value after reset is low.

MMC_SPI[12] SPI Transfer Controlled Start (CSTR)

When this bit is set, the FIFO is disabled and a write in MMC_DATA register automatically triggers an SPI transfer of one byte or one 16-bit word, depending on the local host write access. Because the FIFO is disabled, the local host must wait for the SPI transfer completion signaled by MMC_STAT[AF] = 1 before attempting to write again in the MMC_DATA register.

- 0: No action
- 1: A write in MMC_DATA starts an SPI transfer.

Value after reset is low.

MMC_SPI[11:10] Chip-Select Hold Time Control (TCSH)

This 2-bit field defines the number of interface clock cycles that the core waits after the last serial clock edge before asserting the chip-select signal to its inactive high level.

- 00: 0.5 clock cycle
- 01: 1.5 clock cycles
- 10: 2.5 clock cycles
- 11: 3.5 clock cycles

Value after reset is low (2 bits).

MMC_SPI[9:8] Chip-Select Setup Time Control (TCSS)

This 2-bit field defines the number of interface clock cycles that the core waits after asserting the chip-select signal to its active-low level before asserting the first serial clock edge.

- 00: 1 clock cycle
- 01: 2 clock cycles
- 10: 3 clock cycles
- 11: 4 clock cycles

Value after reset is low (2 bits).

Figure 10. SPI Mode C/S Timing Controls (POL = 0)

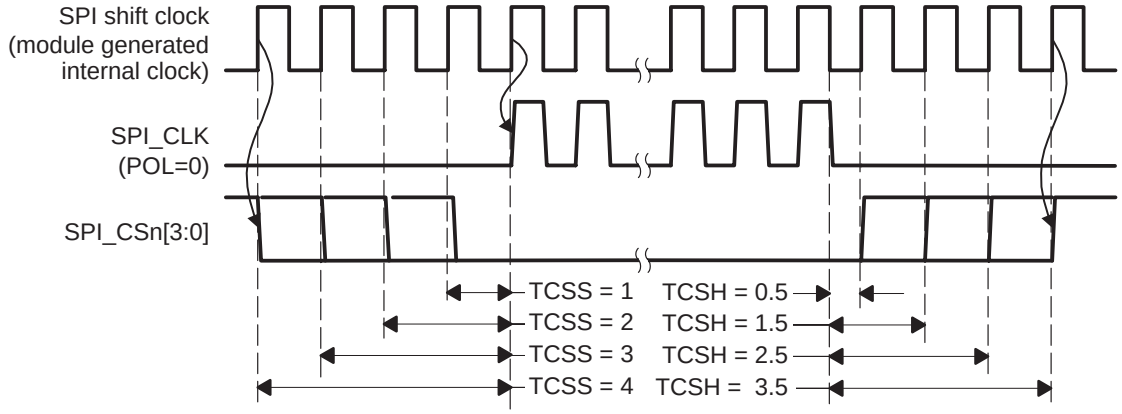
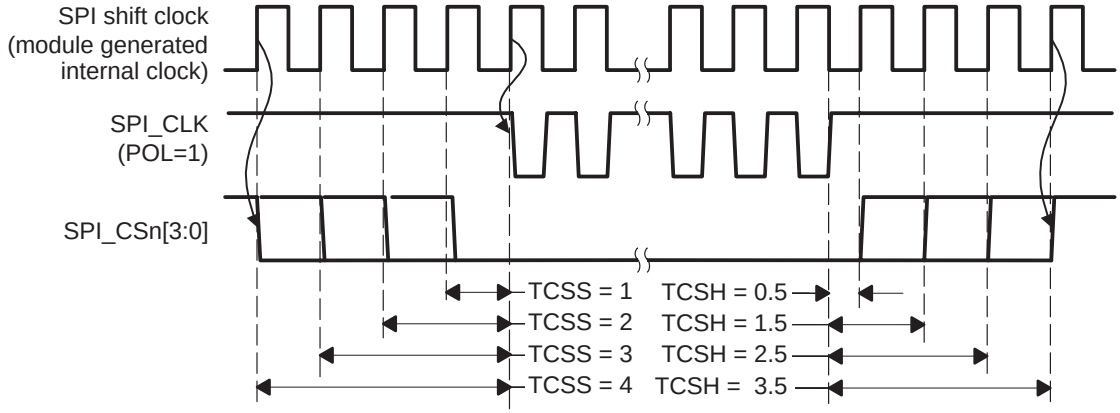


Figure 11. SPI Mode C/S Timing Controls (POL = 1)



MMC_SPI[7] Card Socket Connector Select (CSEL)

When this bit is set, the core outputs the serial clock on the MMC.CLK signal instead of the SPI.CLK signal. It also outputs the chip-select 0 on the MMC.DAT3 signal instead of the SPI_C/Sn[0] signal. This can be used to communicate with MMC/SD/SDIO cards using the SPI protocol on the default MMC/SD card connector.

- ❏ 0: Use the SPI.CLK and MMC_C/Sn[0] signals during SPI transfer (MMC.CLK inactive low and MMC.DAT3 inactive high though pull-up).
- ❏ 1: Use the MMC.CLK and MMC.DAT3 signals during SPI transfer (SPI.CLK inactive low and SPI_C/Sn[0] inactive high).

Value after reset is low.

MMC_SPI[5:4] Chip-Select Control (CS)

Encoded value that selects the device being targeted for SPI transfer.

- 00: C/S 0
- 01: C/S 1
- 10: C/S 2
- 11: C/S 3

Value after reset is low (2 bits).

MMC_SPI[3] Chip-Select Mode (CSM)

When this bit is set to 0 and enabled (MMC_SPI[CSD] = 0), the selected chip-select signal pin is brought active (low) only when the SPI transfer starts, and returns automatically to its inactive state (high) when the SPI transfer completes.

When this bit is set to 1, automatic control of the chip-select signal is disabled, and the signal pin is then manually controlled by the chip-select disable register bit (MMC_SPI[CSD]). This mode provides support for a complex SPI transfer scheme that requires chip-select to be kept active during the entire transfer (for example, MMC card write with busy condition).

- 0: Automatic mode
- 1: Manual mode (controlled by CSD)

Value after reset is low.

MMC_SPI[2] Chip-Select Disable (CSD)

When this bit is set to 0, the selected chip-select signal is asserted to its active (low) state either automatically when MMC_SPI[CSM] = 0, or manually when MMC_SPI[CSM] = 1.

When this bit is set to 1, the selected chip-select signal is forced to its inactive (high) state. This bit can be used to send dummy clocks with chip-select inactive to an MMC or SD card.

- 0: Selected chip-select conditionally asserted (low). See Table 22.
- 1: Selected chip-select deasserted (high)

Value after reset is low.

Table 22. Chip-Select Control (SPI Mode)

CSM	CSD	Selected CS	Comment
0	0	High → Low → High	Automatic mode: CS asserted active (low) during SPI transfer
0	1	High	Automatic mode: CS forced inactive (high)
1	0	Low	Manual mode: CS asserted active (low)
1	1	High	Manual mode: CS asserted inactive (high)

MMC_SPI[1] Clock Phase (PHA)

The clock polarity and clock phase bits select four different clocking schemes for the serial clock pin (SPI.CLK or MMC.CLK). The clock phase bit (MMC_SPI[PHA]) selects a half-cycle delay for the clock.

When the clock phase = 0:

- MSB data is ready one-half cycle of the serial clock before the SPI clock starts.
- Data is shifted in during reception on the first edge transition of the serial clock.
- Data is shifted out in transmission on the second edge transition of the serial clock.

When the clock phase = 1:

- Data is shifted out in transmission on the first edge transition of the serial clock.
- Data is shifted in during reception on the second edge transition of the serial clock.
  - 0: Phase 0
  - 1: Phase 1

Value after reset is low.

MMC_SPI[0] Clock Polarity (POL)

The clock polarity bit (MMC_SPI[POL]) selects the active edge of the clock, either rising or falling.

When this bit is 0, the idle value of the serial clock signal (SPI.CLK or MMC.CLK) is low, and the rising edge of the clock is active.

When this bit is 1, the idle value of the serial clock signal is high, and and the falling edge of the clock is active.

- 0: Rising edge active
- 1: Falling edge active

Value after reset is low.

Table 23. SDIO Mode Configuration Register (MMC_SDIO)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x34		
Bit	**Name**	**Description**
15	C5E	Card status error on bit 5 of response 1 enable
14	C14E	Card status error on bit 4 of response 1 enable
13	C13E	Card status error on bit 3 of response 1 enable
12	C12E	Card status error on bit 2 of response 1 enable
11	D3PS	DAT3 polarity control
10	D3ES	DAT3 edge/level detection mode
9	CDWE	Card-detect wake request enable
8	IWE	Interrupt wake request enable
7	DCR4	Disable CRC7 check in R4 response
6	XDTS	Extended data time-out mode select (default OD)
5	DPE	Data time-out prescaler enable
4	RW	Assert read wait condition to the card
3	−	Reserved
2	CDE	Card-detect mode enable
1	RWE	SDIO read wait mode enable
0	IRQE	SDIO interrupt mode enable

This register configures the MMC/SD interface for SDIO operation and the card-detection mechanism on DAT3. If the card-detection mechanism is based on a mechanical switch, its control/sense is the responsibility of another system module.

MMC_SDIO[15] Card Status Error on R5 Enable (C5E)

This bit must be set to 1 for SDIO cards only.

If this bit is set to 1, the R5 response generates card status errors and enables the command time-out. See SDIO CMD52. See Section 5.1, IO_RW_DIRECT command (CMD52), *SDIO Card Specification*.

By default, when this bit is set to 0, the R5 response does not generate errors or disable the command time-out for MMC interrupt mode support.

- ❏ 0: MMC default mode
- ❏ 1: Card status errors on response R5 enabled (SDIO card only)

Value after reset is low.

MMC_SDIO[14] Card Status Error on Bit 4 of Response R1 Enable (C14E)

This bit must be set to 1 for SDIO cards only.

When this bit is set to 1, a card status error is generated if bit 4 of the status is 1 for an R1 or R1b response.

- ❏ 0: Error on bit 4 masked
- ❏ 1: Card status errors on bit 4 of response R1 enabled (SDIO card only)

Value after reset is low.

MMC_SDIO[13] Card Status Error on Bit 3 of Response R1 Enable (C13E)

This bit must be set to 1 for SD cards only or for an application-specific command that generates an error.

When this bit is set to 1, a card status error is generated if bit 3 of the status is 1 for an R1 or R1b response.

- ❏ 0: Error on bit 3 masked
- ❏ 1: Card status errors on bit 3 of response R1 enabled (SD card or application specific only)

Value after reset is low.

MMC_SDIO[12] Card Status Error on Bit 2 of Response R1 Enable (C12E)

This bit must be set to 1 for an application-specific command that generates an error.

When this bit is set to 1, a card status error is generated if bit 2 of the status is 1 for an R1 or R1b response.

- ❏ 0: Error on bit 2 masked
- ❏ 1: Card status errors on bit 2 of response R1 enabled (application specific only)

Value after reset is low.

MMC_SDIO[11] DAT3 Polarity Select (D3PS)

This bit determines the active edge/level condition for DAT3 signal.

- ❏ 0: Not inverted
- ❏ 1: Inverted

Value after reset is low.

MMC_SDIO[10] DAT3 Polarity Select (D3ES)

This bit determines whether DAT3 is an edge- or level-sensitive signal.

- ❏ 0: Level sensitive
- ❏ 1: Edge sensitive

Value after reset is low.

MMC_SDIO[9] Card-Detect Wake-Request Enable (CDWE)

SD/SDIO card only.

This bit must be set to allow a valid card-detect condition to assert the WAKE_REQ signal active.

- ❏ 0: Masked
- ❏ 1: Unmasked

Value after reset is low.

MMC_SDIO[8] Interrupt Wake Request Enable (IWE)

SDIO card only.

This bit must be set to allow a valid SDIO interrupt condition to assert the WAKE_REQ signal active.

- ❏ 0: Masked
- ❏ 1: Unmasked

Value after reset is low.

MMC_SDIO[7] Disable CRC7 Check on R4 Response (DCR4)

SDIO card only.

This bit must be set to disable the CRC7 check on R4 response. This prevents the MMC_STAT[CCRC] from being set following an IO_SEND_OP_COND command (see *SDIO Card Specification Part E1*).

- ❏ 0: CRC7 check enabled on R4 response
- ❏ 1: CRC7 check disabled on R4 response

Value after reset is low.

MMC_SDIO[6] Extended Data Time-Out Mode Select (XDTS)

This bit determines the default action of the MMC_CMD[ODTO] register bit when passing the command to the card (see section).

The local host must set this bit to 1 after the card has been identified as an SDIO card in order to support the IO_RW_DIRECT command with long time-out values.

This bit can remain clear in all other cases.

- ❏ 0: Open-drain control mode
- ❏ 1: Time-out control mode

Value after reset is low.

MMC_SDIO[5] Data Time-Out Prescaler Enable (DPE)

When the local host sets this bit to 1, the data time-out value (MMC_DTO[DTO]) is x1024 the number of MMC.CLK cycles.

- ❏ 0: x1 (prescaler off)
- ❏ 1: x1024 (prescaler on)

Value after reset is low.

MMC_SDIO[4] Assert Read Wait Condition (RW)

SDIO card only.

The local host can set this bit to stall a read multiple data transfer (CMD53) temporarily. After setting this bit, the host must wait until the wait becomes effective (MMC_STAT[CRW] = 1) before sending a new command without data phase (such as CMD52) to the card.

After completion of CMD52, the local host can resume the read multiple data transfer by clearing this bit.

- 0: No action or read wait released
- 1: Read wait requested (if MMC_SDIO[RWE] = 1)

Value after reset is low.

MMC_SDIO[2] Card-Detect Enable (CDE)

SD or SDIO card only.

The local host must set this bit to 1 to permit the card detection operation.

When this bit is set, the core sets the MMC_STAT[CD] register bit to 1 if a valid card-detection condition occurs on the DAT3 line. This happens when a selectable level/edge sensitive event occurs on DAT3 and the card is in idle state. The MMC_SDIO[D3ES] register bit controls edge/level, and the MMC_SDIO[D3PS] register bit controls polarity. This event also asynchronously asserts the WAKE_REQ signal event to warn the system to wake up its clocks if needed. The MMC_SDIO[CDWE] register bit masks the WAKE_REQ signal for this event.

- 0: Card-detect mode disabled
- 1: Card-detect mode enabled

Value after reset is low.

MMC_SDIO[1] SDIO Read Wait Mode Enable (RWE)

SDIO card only.

The local host must set this bit to 1 to permit the read wait operation.

When this bit is set to 0, the read wait operation is disabled, and setting a 1 in the MMC_SDIO[RW] register bit results in no action.

- 0: Read wait mode disabled
- 1: Read wait mode enabled

Value after reset is low.

MMC_SDIO[0] SDIO Interrupt Mode Enable (IRQE)

SDIO card only.

The local host must set this bit to 1 to permit the interrupt operation.

When this bit is set, the core sets the MMC_STAT[CIRQ] register bit to 1 if an interrupt condition is detected on the DAT1 line. A detected interrupt condition when the card is in idle state also asynchronously asserts the WAKE_REQ signal event to warn the system to wake up its clocks if needed. The MMC_SDIO[IWE] register bit masks the WAKE_REQ signal for this event.

- ❏ 0: Interrupt mode disabled
- ❏ 1: Interrupt mode enabled

Value after reset is low.

Table 24. System Test Register (MMC_SYST)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x38		
Bit	**Name**	**Description**
15	WAKD	WAKE_REQ output signal data value (internal signal to system)
14	SSB	Set status bits
13	RDYD	Ready/busy input signal data value
12	DDIR	DAT[3:0] signals direction
11	D3D	DAT3 input/output signal data value
10	D2D	DAT2 input/output signal data value
9	D1D	DAT1 input/output signal data value
8	D0D	DAT0/SI input/output signal data value
7	CDIR	CMD/SO signal direction
6	CDAT	CMD/SO input/output signal data value
5	MCKD	MMC clock output signal data value
4	SCKD	SPI clock output signal data value
3	CS3D	C/S3 output signal data value
2	CS2D	C/S2 output signal data value
1	CS1D	C/S1 output signal data value
0	CS0D	C/S0 output signal data value

This register controls the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode.

MMC_SYST[15] WAKE_REQ Data (WAKD)

The WAKE_REQ signal is driven high or low according to the value written into this register bit.

Value after reset is low.

MMC_SYST[14] Set Status Bits (SSB)

Writing 1 into this bit sets to 1 all status bits contained in the MMC_STAT register.

Writing 0 into this bit does not clear already set status bits; only writing 1 into a set status bit can clear it (see MMC_STAT operation). This bit must be cleared before attempting to clear a status bit.

- ❏ 0: No action
- ❏ 1: Set all status bits

Value after reset is low.

MMC_SYST[13] Ready/Busy Data (RDYD)

This read-only bit returns the value of the signal on the input pad (high or low).

- ❏ 0: Ready/busy low
- ❏ 1: Ready/busy high

Value after reset is high.

MMC_SYST[12] Direction (DDIR)

When set, this bit places all of the in/out MMC_DAT[3:0] pins in output mode.

- ❏ 0: Input
- ❏ 1: Output

Value after reset is low.

MMC_SYST[11:8] Data (DnD)

If MMC_SYST[DDIR] = 0 (input mode direction), these bits return the value on the corresponding MMC.DAT pins (high or low). A write into these bits has no effect.

If MMC_SYST[DDIR] = 1 (output mode direction), the MMC.DAT pins are driven high or low according to the value written into these register bits.

Value after reset is low (all 4 bits).

MMC_SYST[7] CMD Direction (CDIR)

When set, this bit places the in/out MMC.CMD/SPI.DO pin in output mode.

- 0: Input
- 1: Output

Value after reset is low.

MMC_SYST[6] CMD Data (CDAT)

If MMC_SYST[CDIR] = 0 (input mode direction), these bits return the value on the MMC.CMD/SPI.DO pin (high or low). A write into this bit has no effect.

If MMC_SYST[CDIR] = 1 (output mode direction), the MMC_CMD pin is driven high or low according to the value written into this register bit.

Value after reset is low.

MMC_SYST[5] MMC.CLK Data (MCKD)

The MMC.CLK pin is driven high or low according to the value written into this register bit.

Value after reset is low.

MMC_SYST[4] SPI.CLK Data (SCKD)

The SPI.CLK pin is driven high or low according to the value written into this register bit.

Value after reset is low.

MMC_SYST[3:0] CSData (CSnD)

The CS[3:0] pins are driven high or low according to the values written into these register bits.

Value after reset is low (all 4 bits).

Table 25. Module Revision Register (MMC_REV)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x3C		
Bit	**Name**	**Description**
15:8	–	Reserved
7:0	REV	Module revision number

This read-only register contains the revision number of the module. A write to this register has no effect.

MMC_REV[7:0] Module Revision Number (REV)

This 8-bit field indicates the revision number of the RTL for this module. This value is fixed by hardware.

- ❏ The 4 LSBs indicate a minor revision.
- ❏ The 4 MSBs indicate a major revision.
- ❏ 0x30: Revision 3.0
- ❏ 0x31: Revision 3.1

A reset has no effect on the value returned.

Notes: 1) MMC/SD HOST controller without SDIO support is revision 1.x or 2.x (WMU_020_1 spec)
2) MMC/SD/SDIO HOST controller is revision 3.x.

Table 26. MMC/SD Command Response Register 0 (MMC_RSP0)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x40		
Bit	**Name**	**Description**
15:0	RSP0	CMD response (R2[15:0])

This 16-bit register holds bit positions [15:0] for a 128-bit response of type R2.

MMC_RSP0 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 27. MMC/SD Command Response Register 1 (MMC_RSP1)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x44		
Bit	**Name**	**Description**
15:0	RSP1	CMD response (R2[31:16])

This 16-bit register holds bit positions [31:16] for a 128-bit response of type R2.

MMC_RSP1 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 28. MMC/SD Command Response Register 2 (MMC_RSP2)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x48		
Bit	**Name**	**Description**
15:0	RSP2	CMD response (R2[47:32])

This 16-bit register holds bit positions [47:32] for a 128-bit response of type R2.

MMC_RSP2 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 29. MMC/SD Command Response Register 3 (MMC_RSP3)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x4C		
Bit	**Name**	**Description**
15:0	RSP3	CMD response (R2[63:48])

This 16-bit register holds bit positions [63:48] for a 128-bit response of type R2.

MMC_RSP3 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 30. MMC/SD Command Response Register 4 (MMC_RSP4)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x50		
Bit	**Name**	**Description**
15:0	RSP4	CMD response (R2[79:64])

This 16-bit register holds bit positions [79:64] for a 128-bit response of type R2.

MMC_RSP4 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 31. MMC/SD Command Response Register 5 (MMC_RSP5)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x54		
Bit	**Name**	**Description**
15:0	RSP5	CMD response (R2[95:80])

This 16-bit register holds bit positions [95:80] for a response of type R2.

MMC_RSP5 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 32. MMC/SD Command Response Register 6 (MMC_RSP6)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x58		
Bit	**Name**	**Description**
15:0	RSP6	CMD response (R2[111:96], R1/R1b/R3/R4/R5/R6[23:8])

This 16-bit register holds bit positions [111:96] for a 128-bit response of type R2 and bit positions [23:8] for a 32-bit response of types R1/R1b/R3/R4/R5/R6.

MMC_RSP6 is also reset after a write into the MMC_CMD register (command sent).

Table 33. MMC/SD Command Response Register 7 (MMC_RSP7)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x5C		
Bit	**Name**	**Description**
15:0	RSP7	CMD response (R2[127:112], R1/R1b/R3/R4/R5/R6[39:24])

This 16-bit register holds bit positions [127:112] for a 128-bit response of type R2 and bit positions [39:24] for a 32-bit response of types R1/R1b/R3/R4/R5/R6.

MMC_RSP7 is also reset after a write into the MMC.CMD/SPI.DO register (command sent).

Table 34. SDIO Suspend/Resume Control Register (MMC_IOSR)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x60		
Bit	**Name**	**Description**
15:4	–	Reserved
3	STOP	Stop core data operation request (after card grants suspend)
2	SAVE	Save FIFO contents of suspended function
1	RESU	Next SD command is a resume request
0	SUSP	Next SD command is a suspend request

This register implements the necessary controls for SDIO suspend/resume operations.

MMC_IOSR[3] Stop Core Data Operation Request (STOP)

SDIO cards only

The local host can set this bit to 1 only when the SDIO card function has been suspended. When this bit is set, the core immediately stops signaling the data transfer request (either DMA request or interrupt) to the system if no request is pending, or immediately after it has been serviced if a request is pending. Once the bit is set, the core automatically clears it to signal that FIFO

operations are effectively suspended. This happens when all data requests have been disabled and serviced. The local host must poll this bit until cleared to 0 before attempting to save the FIFO contents.

A write to 0 has no action.

- 0: No action, or FIFO operations suspended
- 1: Stop core data operation request

Value after reset is low.

MMC_IOSR[2] Save FIFO Contents of Suspended Function (SAVE)

SDIO cards only.

This bit must be set to 1 when the core acknowledges the stop data operation request MMC_IOSR[STOP] = 0. When this bit is set, the FIFO is placed in read mode, and the local host downloads the FIFO contents of the suspended function until totally empty (signaled by MMC_STAT[AF] = 0).

In this mode, the actual threshold value set in MMC_BUFF[AFL] is a don't care. MMC_STAT[AF] equals 0 only when the FIFO is totally empty.

- 0: No action
- 1: Save action (SDIO card only)

Value after reset is low.

MMC_IOSR[1] Next SD Command Is a RESUME Request (RESU)

SDIO cards only.

This bit must be set to 1 when the next command passed to the card is a resume request for a suspended function.

- 0: No action
- 1: Next command is resume (SDIO card only).

MMC_IOSR[0] Next SD Command Is a SUSPEND Request (SUSP)

SDIO cards only.

This bit must be set to 1 when the next command passed to the card is a suspend request of the current active function.

When this bit is set for a multiple block read or write operation, the core automatically stops its data transfer operation at the end of the current active block. It also stops requesting new data from the local host or system DMA for a new block write.

- 0: No action
- 1: Next command is suspend (SDIO card only).

Value after reset is low.

Table 35 lists the characteristics of the system control register.

Table 35. System Control Register(1.1MMC_SYSC)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x64		
Bit	**Name**	**Description**
15:2	–	Reserved
1	SRST	Software reset
0	–	Reserved

MMC_SYSC[1] Software Reset (SRST)

When this bit is set to 1, the entire module is reset as for the hardware reset. The core automatically sets this bit to 0, and it is only reset by the hardware reset. It always returns 0 during reads.

The module can be also partially reset using the MMC_CON[POW] bit.

- 0: No action
- 1: Module is reset.

Value after reset is low.

Table 36. System Status Register(MMC_SYSS)

Base Address = 0xFFFB 7800 and 0xFFFB 7C00, Offset Address = 0x68		
Bit	**Name**	**Description**
15:1	–	Reserved
0	RSTD	Reset done

MMC_SISS[0] Reset Done Status (RSTD)

This read-only bit indicates the state of the reset in case of hardware reset, global software reset MMC_SYSC[SRST], or partial software reset MMC_CON[POW]).

The module must receive all of its clocks before it can grant a reset completed status.

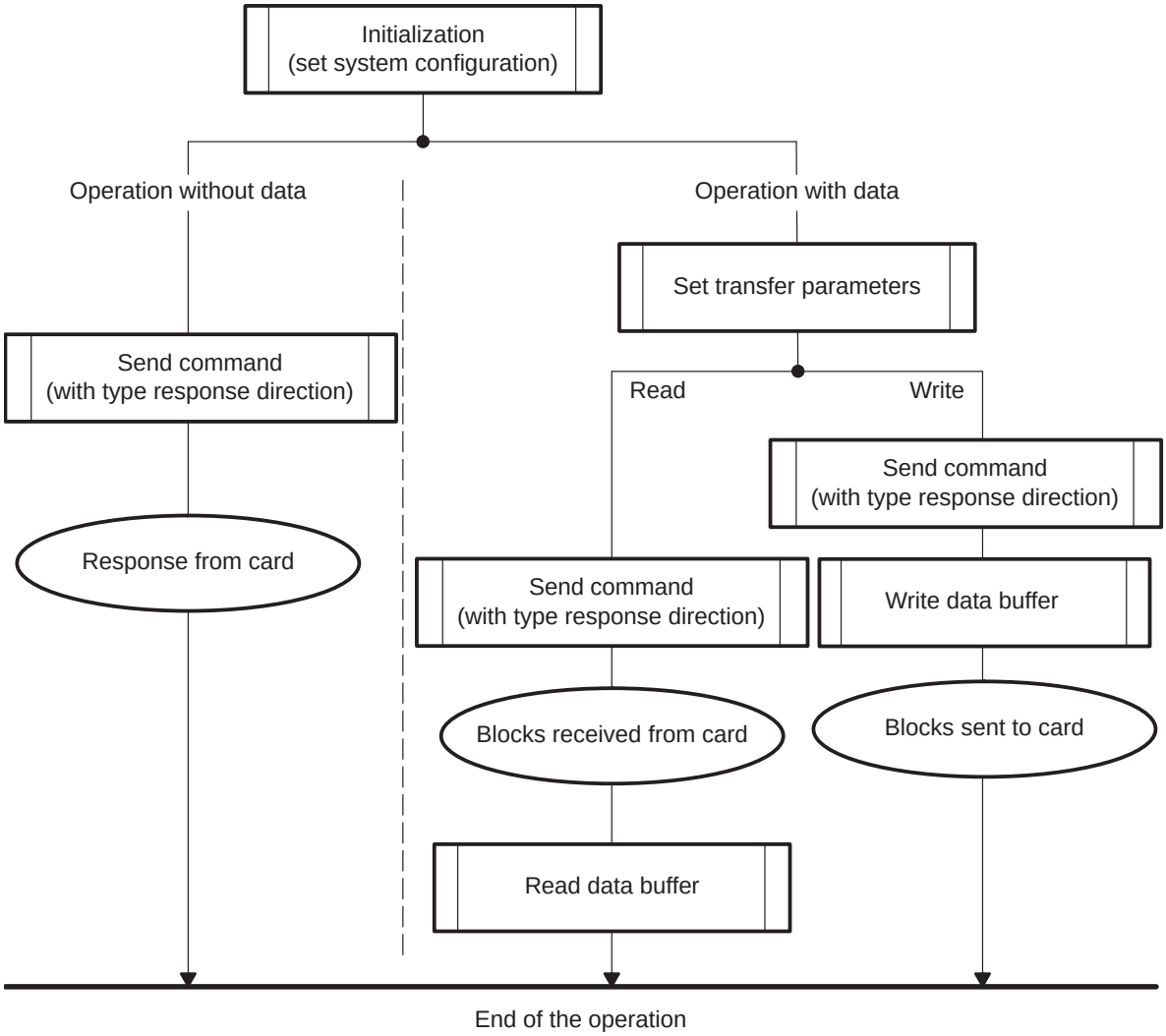
- ❏ 0: Internal module reset in ongoing or partially held in reset
- ❏ 1: Reset completed

Value after reset is low.

3 MMC Command Flow

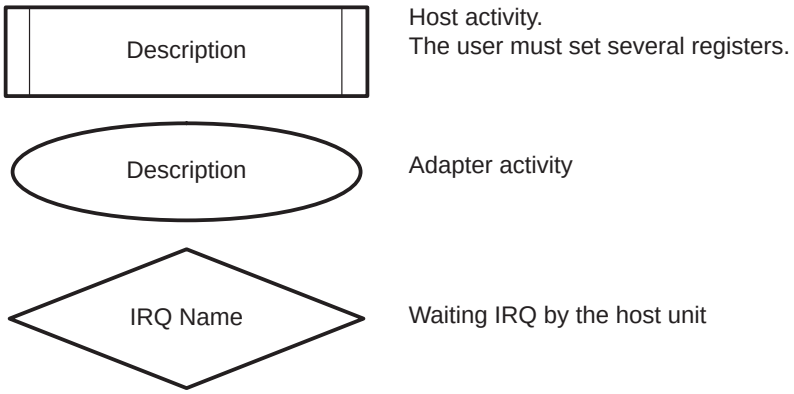
To properly drive the MMC/SD/SDIO correctly for the execution of a command, the host must process as shown in Figure 12 and Figure 13.

Figure 12. General Command Flow



In all modes (MMC, SD, SPI), an initialization phase is necessary at the beginning. After the initialization, the MMC/SD adapter works differently depending on whether the host sends a command without data or with data, and also according to the type, the index of the response, and the direction.

Figure 13. Flow Conventions



3.1 Basic Operations

Figure 14 depicts initialization.

Figure 14. Initialization

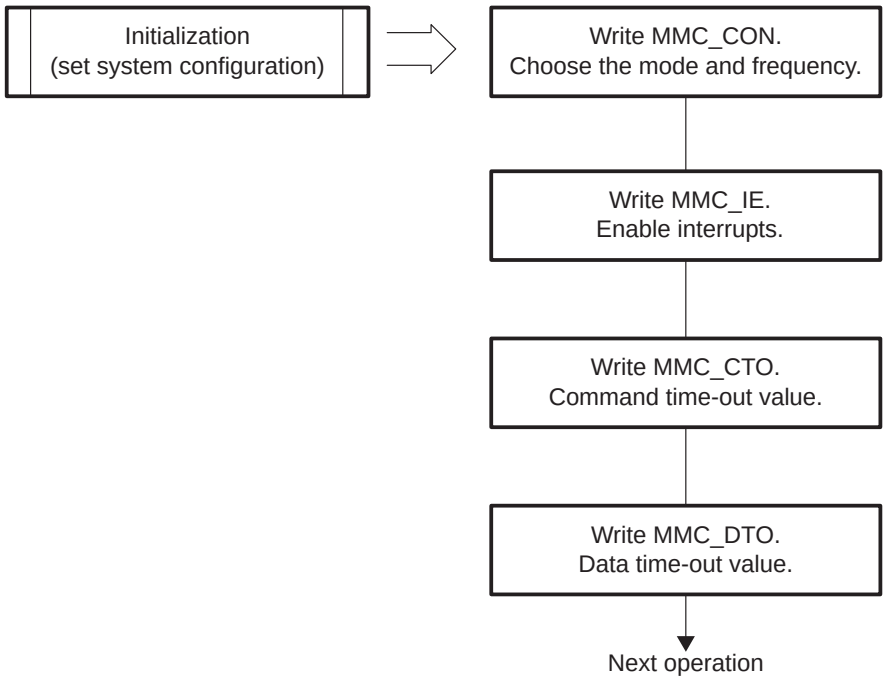


Figure 15 depicts the command transfer.

Figure 15. Command Transfer

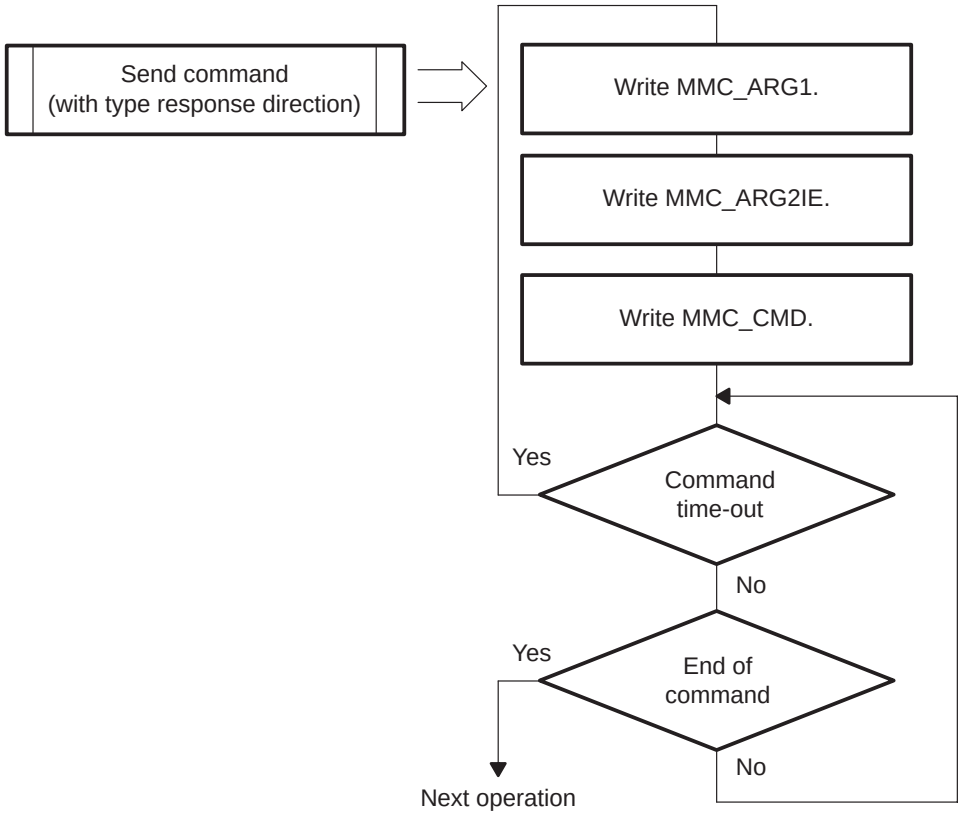
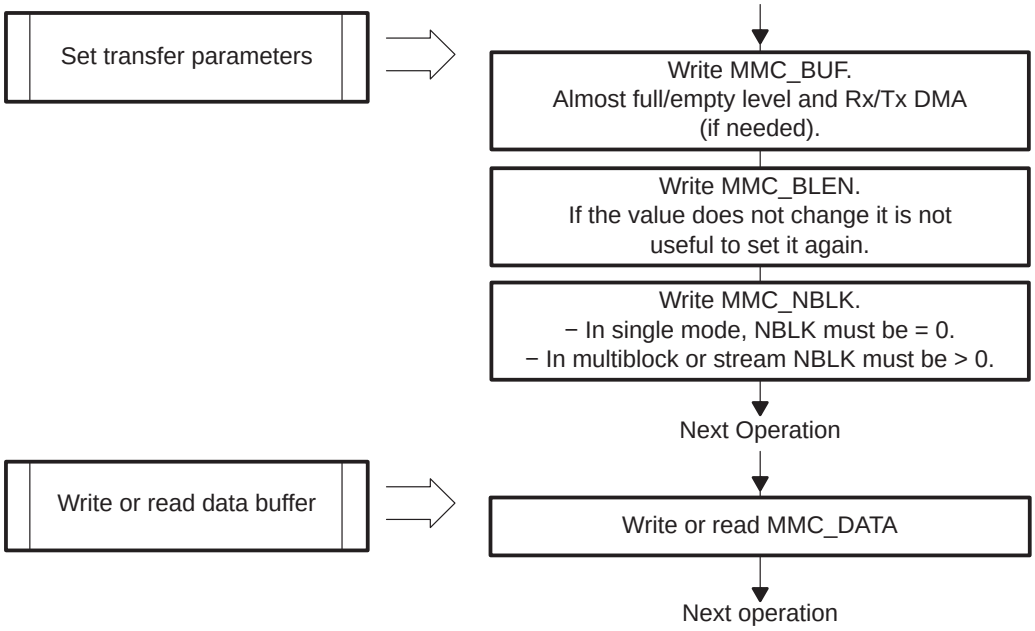


Figure 16 illustrates the transfer of data.

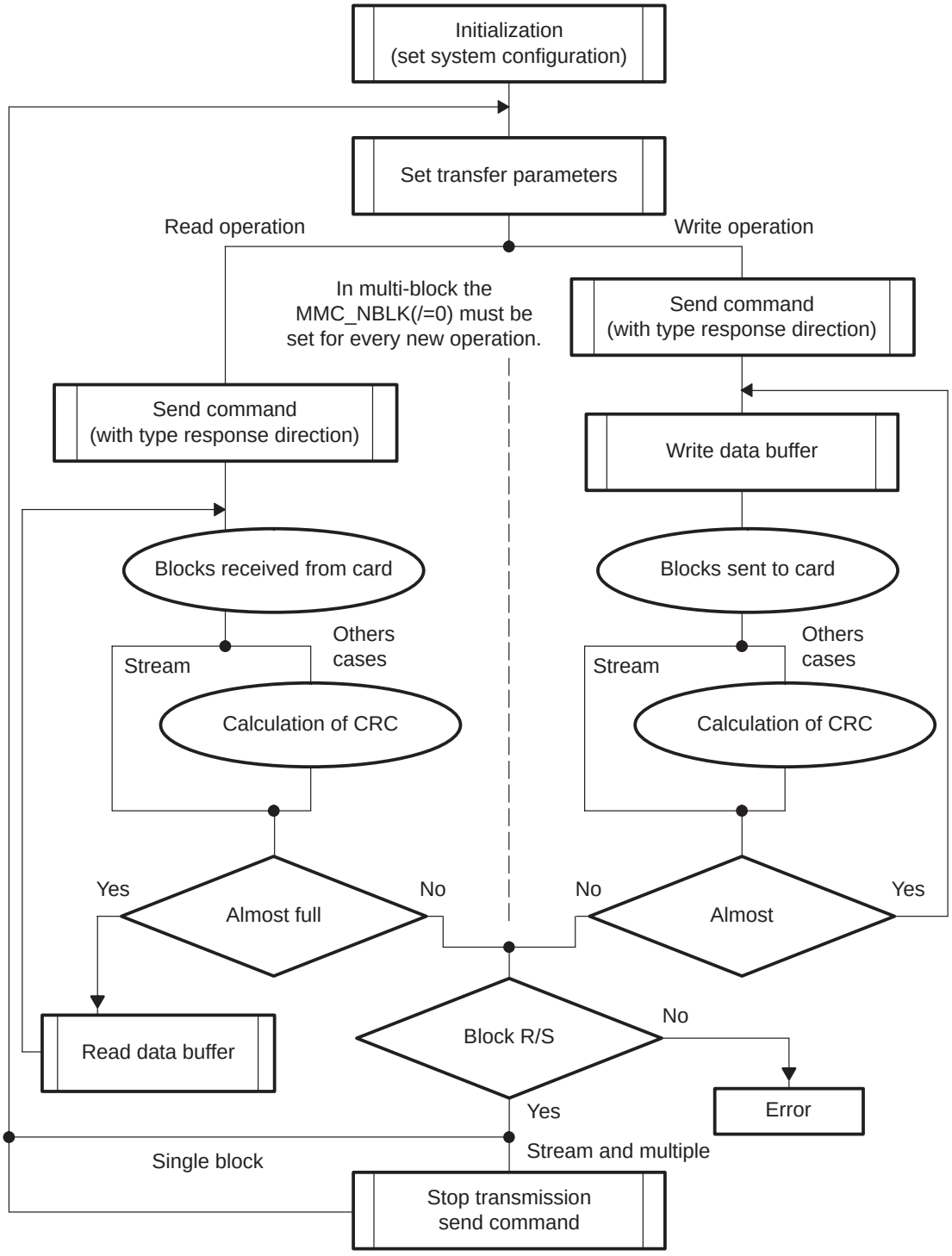
Figure 16. Data Transfer



Mode MMC/SD (MMC_CON[MODE] = 00) is selected for this example.

Figure 17 illustrates the transfer of data in MMC/SD mode.

Figure 17. Data Transfer in MMC/SD Mode


Initialization
(set system configuration)
Set transfer parameters
Read operation
Write operation
In multi-block the
MMC_NBLK(/=0) must be
set for every new operation.
Send command
(with type response direction)
Send command
(with type response direction)
Write data buffer
Blocks received from card
Blocks sent to card
Others
cases
Stream
Calculation of CRC
Others
cases
Stream
Calculation of CRC
Yes
Almost full
No
No
Almost
Yes
Read data buffer
Block R/S
No
Error
Yes
Single block
Stream and multiple
Stop transmission
send command

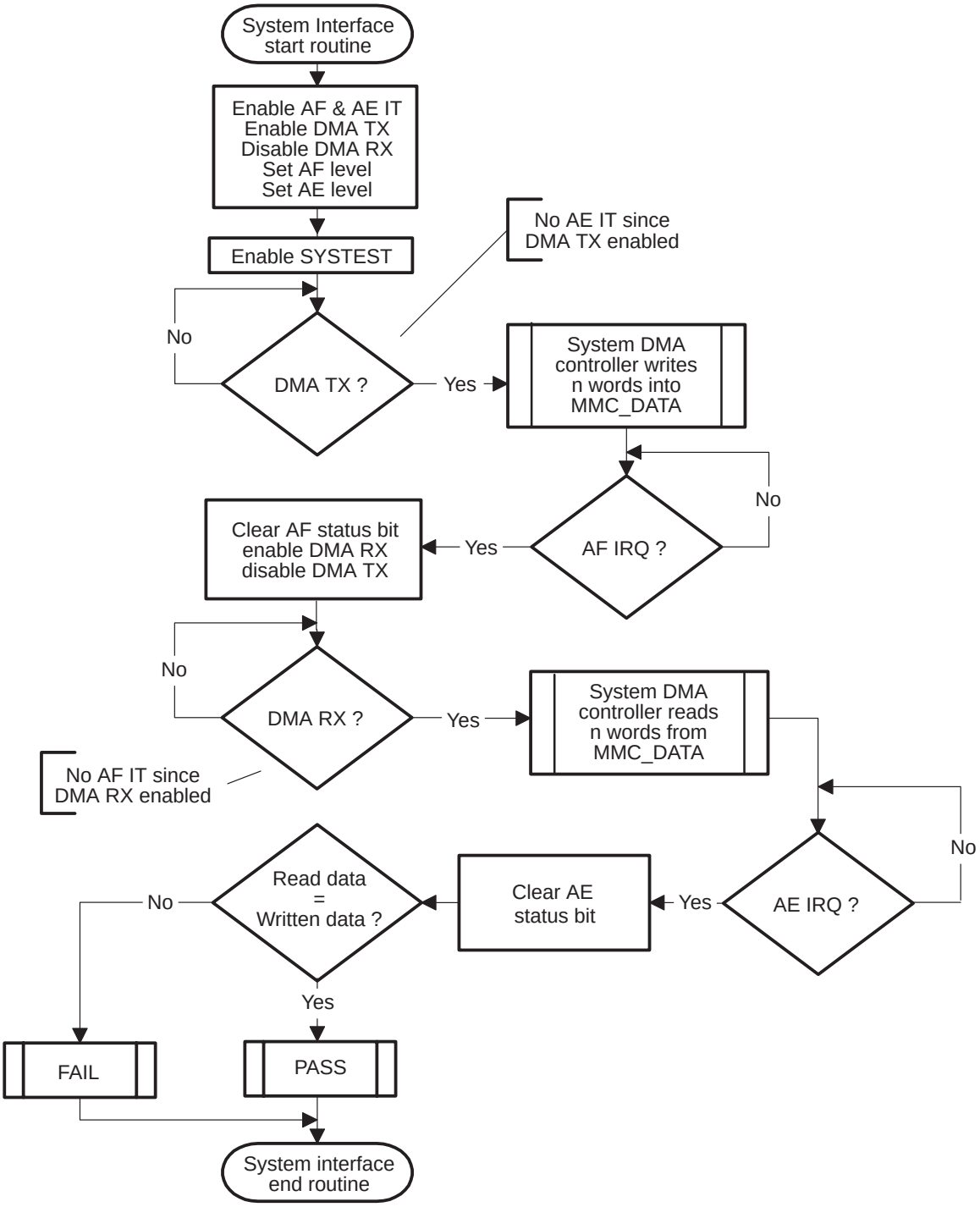
3.2 System Test Mode

The system test (SYSTEST) mode implemented in the MMC/SD host controller easily checks the correctness of the system interconnect either internally to the peripheral bus, central DMA, and interrupt handler, or externally to device I/O pads.

I/O verification is not depicted here but can be performed in SYSTEST mode by toggling the outputs and capturing the logic state of the inputs.

Figure 18 depicts a scenario to validate data DMA transfers and interrupt assertions in a one-pass flow. All data transfers are performed under DMA control.

Figure 18. System Interface Test Flow



3.3 SPI Mode

In write operations, the block length register (MMC_BLEN[BLEN]) is loaded with the data transfer dimension, opcode, and address.

In read operations, the block length register is only loaded with the data transfer dimension.

4 DMA Operations

4.1 MMC DMA Receive Mode

In a DMA block read operation (single or multiple), the DMA RX request signal is asserted to its active level when the FIFO level becomes greater than or equal to the threshold value (in 16-bit words) set in MMC_BUF[AFL]. The DMA RX request is deasserted to its inactive level when the system DMA has read one single word from the FIFO.

Because the request lasts one 16-bit word read cycle, it is recommended that the threshold value in MMC_BUF[AFL] (expressed in 16-bit words) be equal to the DMA burst access size (n). If the system DMA does not support more than one 16-bit word read access, MMC_BUF[AFL] must be set to 0.

New DMA requests are internally masked until the system DMA has performed exactly n reads.

Note: Block Size

Because each DMA transfer is of equal size, the block size of the transfer must be a multiple of the DMA read access size.

Summary:

- DMA transfer size = n ≤ FIFO size (max 32 16-bit words)
- MMC_BUF[AFL] = n − 1 (FIFO threshold level)
- n = Submultiple of block size

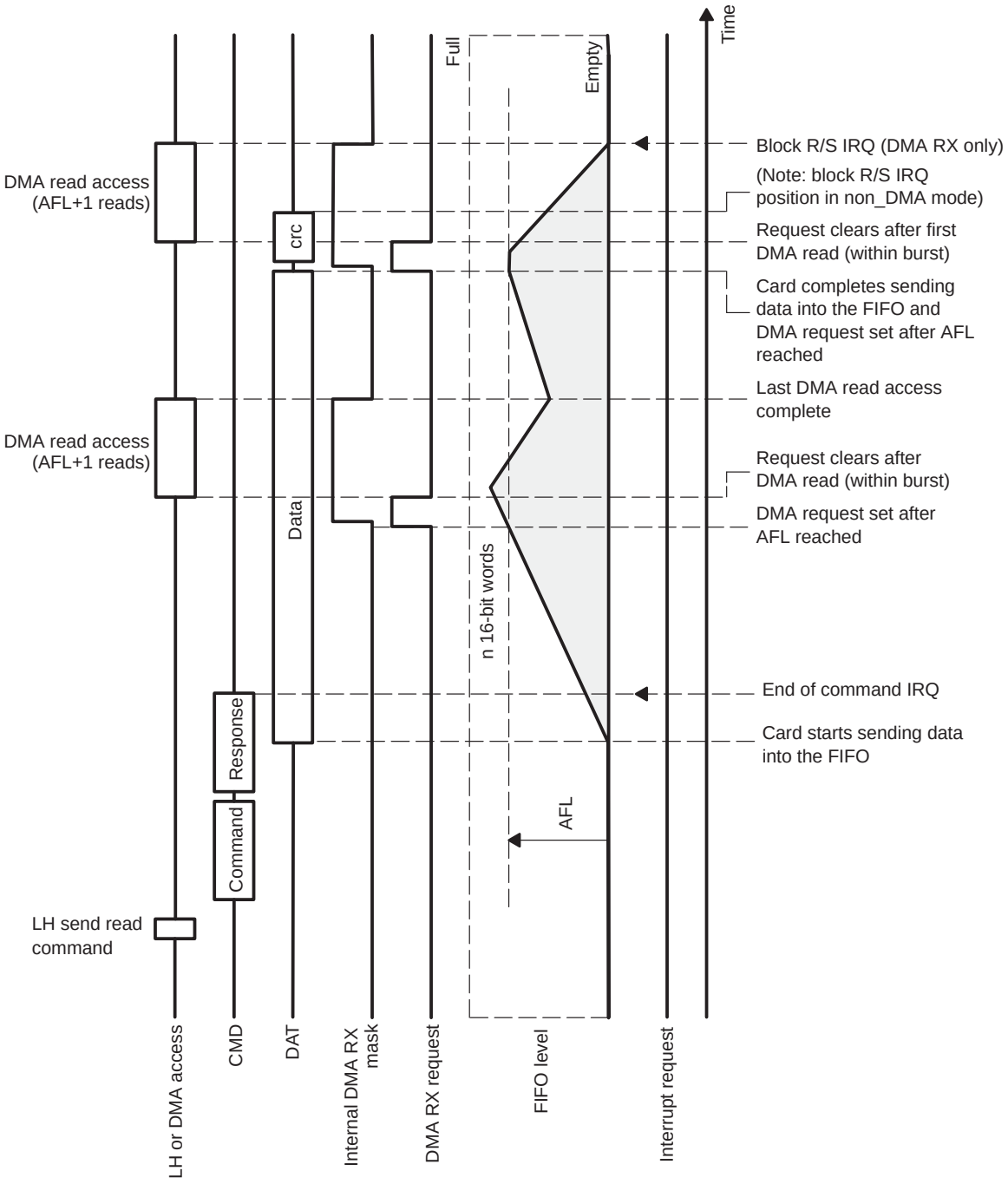
Example: Multiple block read of 7 blocks of 256 bytes each.

The DMA transfer size n can be set to 4 16-bit words (8 bytes) and MMC_BUF[AFL] = 0x3. The read transfer operation completes after 224 system DMA RX requests.

The receive FIFO never overflows. If the FIFO becomes full, the MMC.CLK signal is momentarily stopped until the system DMA or the local host performs a read access, which starts emptying the FIFO.

Figure 19 shows a typical DMA read operation to an MMC card.

Figure 19. MMC Mode DMA RX Transfer



Signals are represented in a symbolic form (a high level for the DMA mask or the request signals denotes an ON condition).

4.2 MMC DMA Transmit Mode

In a DMA block write operation (single or multiple), the DMA TX request signal is asserted to its active level when the FIFO level becomes less than or equal to the threshold value (in 16-bit words) set in MMC_BUF[AEL] after the block write command has been set (write action into MMC.CMD/SPI.DO). The DMA TX request is deasserted to its inactive level when the system DMA has written one single word into the FIFO.

Because the request lasts one 16-bit word write cycle, it is recommended that the threshold value in MMC_BUF[AEL] (expressed in 16-bit words) be equal to the DMA burst size (n). If the system DMA does not support more than one 16-bit word write access, MMC_BUF[AEL] must be set to 0.

New DMA requests are internally masked until the system DMA has performed exactly n writes.

Note: Block Size

Because each DMA transfer is of equal size, it is necessary to have the block size of the transfer be a multiple of the DMA write access size.

Summary:

- DMA transfer size = n ≤ FIFO size (max 32 16-bit words)
- MMC_BUF[AEL] = n − 1 (FIFO threshold level)
- n = Submultiple of block size

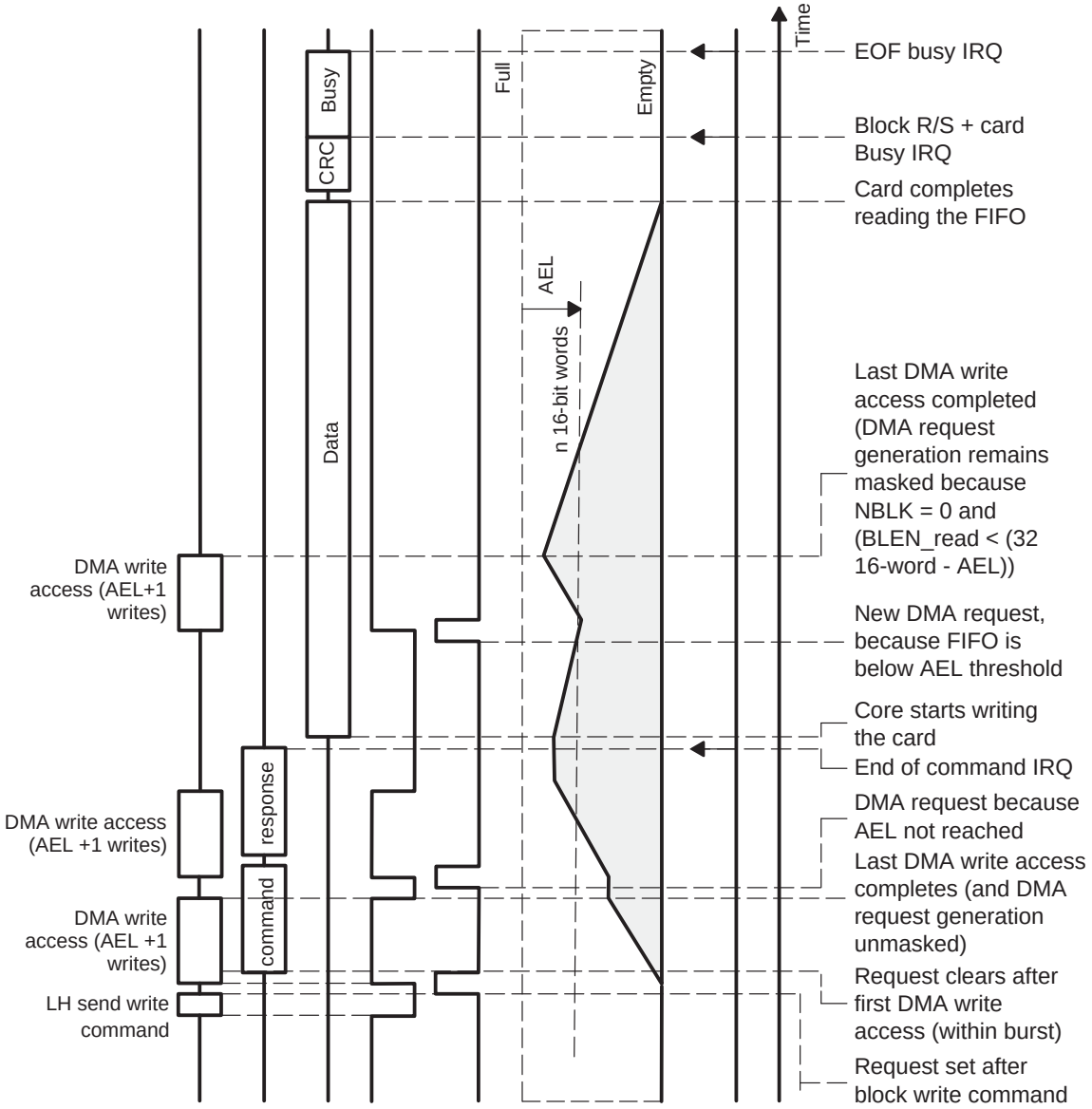
Example: Multiple block write of 10 blocks of 512 bytes each

The DMA transfer size n can be set to 16 16-bit words (32 bytes) and MMC_BUF[AEL] = 0xF. The write transfer operation completes after 160 system DMA TX requests.

The transmit FIFO never underflows. If the FIFO is emptied, the MMC.CLK clock signal is momentarily stopped until the system DMA or the local host performs a write access, which starts filling the FIFO.

Figure 20 shows a typical DMA write operation to an MMC card.

Figure 20. MMC Mode DMA TX Transfer



Signals are represented in a symbolic form (a high level for the DMA mask or the request signals denotes an ON condition).

4.3 SDIO Suspend/Resume

Suspend/resume only apply to multiple-block read or write operations.

To suspend a function, the local host must:

1) Save MMC_CMD[DDIR] value.
2) Set MMC_IOSR[SUSP] to 1.
3) Send the suspend command (CMD52) and check the SDIO card status for acknowledgment. Repeat the command as long as necessary.
4) Set MMC_IOSR[STOP] to 1 and read MMC_IOSR[STOP] until 0.
5) Disable DMA channel if enabled.
6) Save MMC_NBLK register.
7) Set MMC_IOSR[SAVE] bit to 1 and empty the FIFO by reading MMC_DATA register until empty (signaled by MMC_STAT[AF] = 0.
8) Clear MMC_IOSR[SUSP] and MMC_IOSR[SAVE] bits to 0.

A new command can be sent to another function at this point.

To resume the suspended function, the local host must:

1) Restore the FIFO contents by writing into MMC_DATA register.
2) Restore MMC_NBLK register with the saved value.
3) Reenable DMA channel if in DMA mode.
4) Set MMC_IOSR[RESU] bit to 1.
5) Send the resume command (CMD52) with MMC_CMD[DDIR] set according to the suspended function (needed to resume the function as a multiple-block read or as a multiple-block write)
6) Clear MMC_IOSR[RESU] bit to 0.

Table 37 contains information for programming the CMD register.

Table 37. Programming Aid for CMD Register (MMC_CMD)

	DDIR	SHR	TYPE	BUSY	Resp	Hex. Value (MSB of MMC.CMD/ SPI.DO)
bc: no resp	0	0	00	0	000	0x00
bcr: R3	0	0	01	0	011	0x13
bcr – R6	0	0	01	0	110	0x16
bcr: R2	0	0	01	0	010	0x12
bcr: R5	0	0	01	0	101	0x15
ac: R1	0	0	10	0	001	0x21
ac: R1b	0	0	10	1	001	0x29
ac: R2	0	0	10	0	010	0x22
ac: no resp	0	0	10	0	000	0x20
ac: R4	0	0	10	0	100	0x24
adtc: R1	1	0	11	0	001	0xB1
(no stream)	0					0x31
adtc: R1b	1	0	11	1	001	0xB9
(no stream)	0					0x39
adtc: R1	1	1	11	0	001	0xF1
(stream)	0					0x71
adtc: R1b	1	1	11	1	001	0xF9
(stream)	0					0x79
Host response	0	1	00	0	000	0x40

4.4 Programming Model Incompatibility

Software developed for the previous WMU_020_1 MMC/SD host controller version is compatible with this version except for the listed changes:

- MMC_CMD[INAB]: The previous core, send 80-clock initialization sequence followed by a command. This core only sends 80 clocks without any commands.

- MMC_CON[MODE]: The previous core defines an SPI mode 2 operation for SPI operated MMC/SD/SDIO cards on top of the SPI mode 1 operation for serial flash card. This is now replaced by a single SPI operation with a new SPI control bit MMC_SPI[CSEL].
- MMC_CMD[SHR]: The previous core did not send the host response in open-drain mode. The new core does, so it must be performed at the proper device identification speed.
- MMC_BUF[AFL]: The previous core defines a reset value of 0x1F. The reset value is 0x00 for this core.

Index