Customer Training Workshop
Traveo™ II SDHC Host Controller

Q4 2020

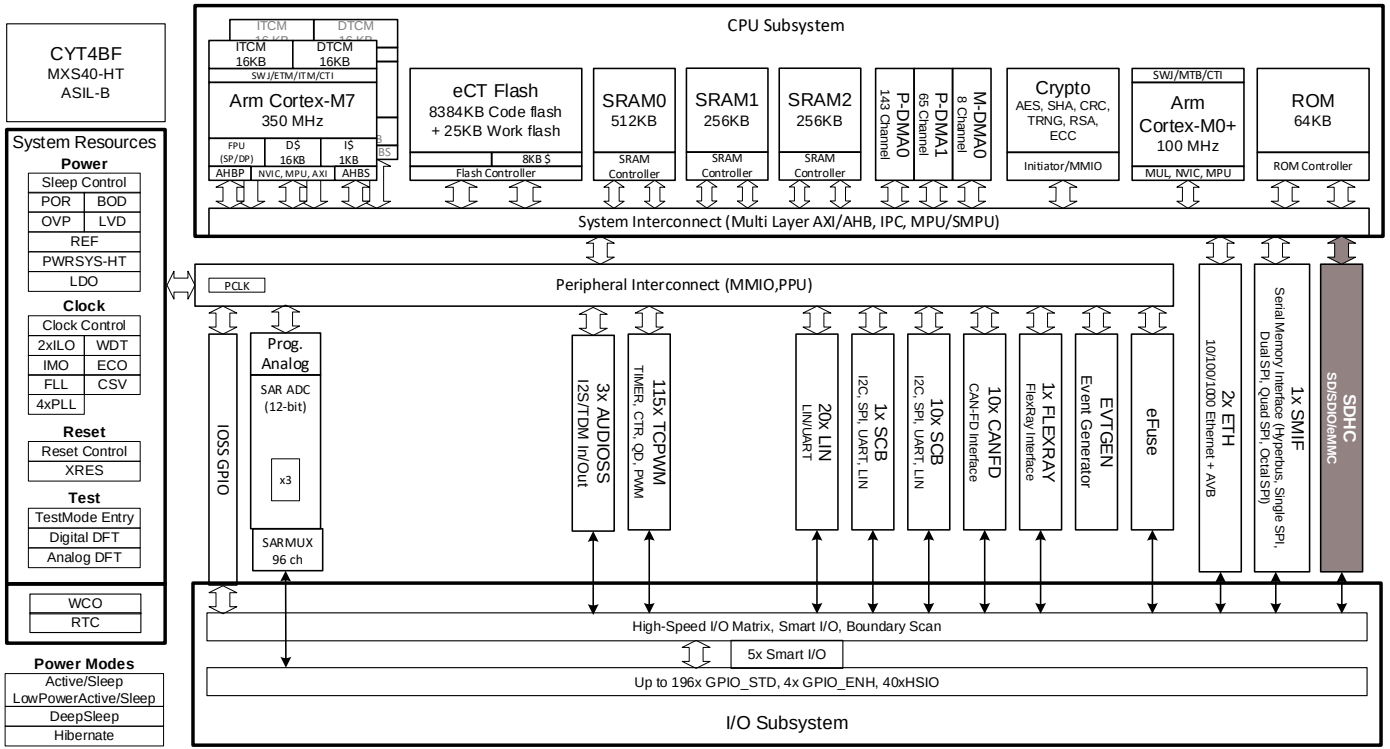
Target Products

› Target product list for this training material

Family Category	Series	Code Flash Memory Size
Traveo™ II Automotive Body Controller High	CYT3BB/4BB	Up to 4160KB
Traveo II Automotive Body Controller High	CYT4BF	Up to 8384KB

Introduction to Traveo II Body Controller High

› SDHC Host Controller is located in the Peripheral blocks



Hint Bar

Review TRM chapter 33 for additional details

SDHC Host Controller Overview

- Overview
  - SDHC Host Controller enables interface with:
    - Secure Digital (SD) cards
    - Secure Digital Input Output (SDIO) cards or WiFi products (e.g. CYW4343W)
    - Embedded Multimedia Card (eMMC)–based memory devices
- Features
  - SD 6.0, SDIO 4.10, and eMMC 5.1 standards
  - Host Controller Interface (HCI) 4.2
  - eMMC/SD/SDIO Interface for 3.3 V
  - Three DMA modes
  - I/O interfaces
    - Card detection
    - Mechanical write protection

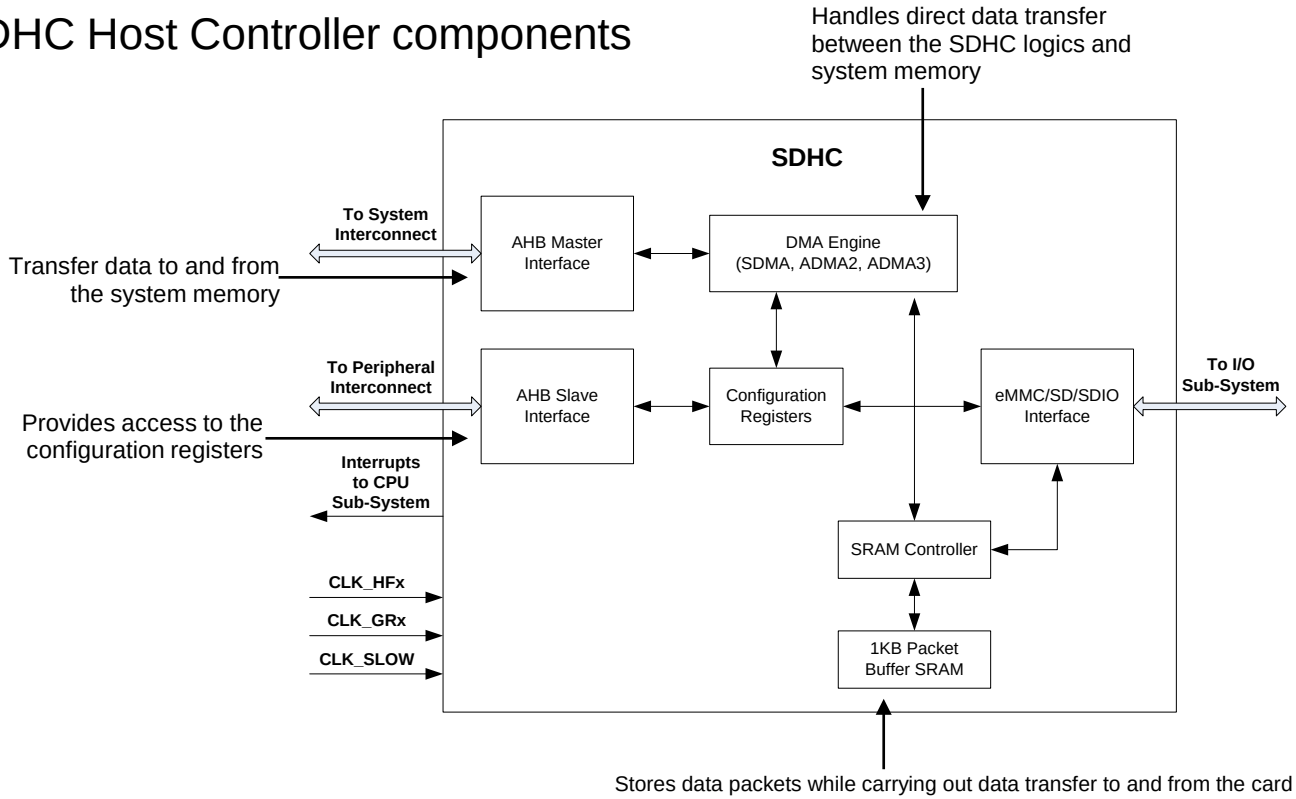
Hint Bar

Review the following documents for additional details

- **Datasheet**
- **TRM chapter 33**
- **SD Specifications Part 1 Physical Layer Specification Version 6.00**
- **SD Specifications Part A2 SD Host Controller Standard Specification Version 4.20**
- **SD Specifications Part E1 SDIO Specifications Version 4.10**
- **Embedded Multi-Media Card (eMMC) Electrical Standard 5.1**

SDHC Host Controller Block Diagram

SDHC Host Controller components

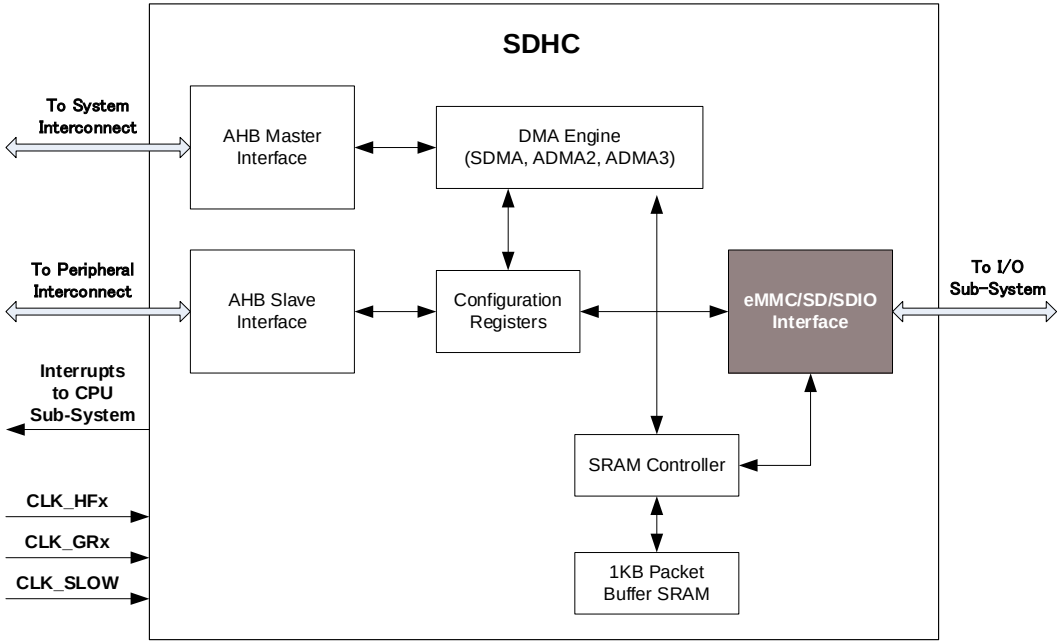


Hint Bar

Review TRM section 33.2 for additional details

SDHC Host Controller Block Diagram

- SDHC Host Controller components
  - eMMC/SD/SDIO Bus Protocol
  - eMMC/SD/SDIO Interface

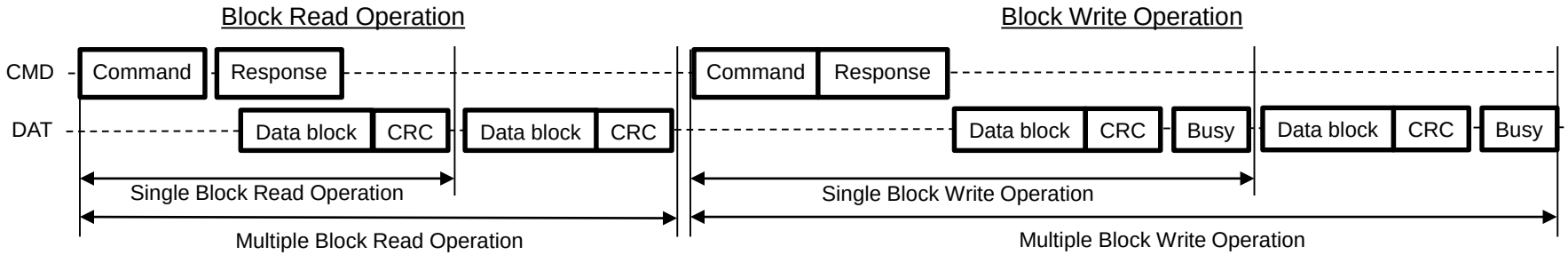


Hint Bar

Review TRM section 33.1 for additional details

eMMC/SD/SDIO Bus Protocol

- Each message is represented by one of the following tokens
  - Command: Command is a token that is sent from the host to the card for operation start
  - Response: Response is a token that is sent from the card to the host as an answer to a previously received command
  - Data: Data is transferred via the data line
  - Busy: Block write operation uses a simple busy signaling of the write operation on the DAT0 line



eMMC/SD/SDIO Interface

Data Bus and Speed Modes

	eMMC	SD	SDIO
Data Bus	1-bit 4-bit 8-bit	1-bit 4-bit	
Max. Speed Modes	Legacy: 26 MBps @ 26 MHz, 8-bit High Speed SDR: 52 MBps @ 52 MHz, 8-bit High Speed DDR: 104 MBps @ 52 MHz, 8-bit	Default Speed (DS): 12.5 MBps @ 25 MHz, 4-bit High Speed (HS): 25.0 MBps @ 50 MHz, 4-bit	

I/O Interface

Signal	Function	eMMC	SD	SDIO
SDHC_CLK_CARD	Clock output	Yes	Yes	Yes
SDHC_CARD_CMD	Command (bi-directional)	Yes	Yes	Yes
SDHC_CARD_DAT_3TO0_[3:0]	Data (bi-directional)	Yes	Yes	Yes
SDHC_CARD_DAT_7TO4_[7:4]	Data (bi-directional)	Yes	-	-
SDHC_CARD_DETECT_N[1]	Card detect signal input, Active low	-	Yes	-
SDHC_CARD_MECH_WRITE_PROT[1]	Mechanical write protect signal input, Active low	-	Yes	-
SDHC_CARD_IF_PWR_EN	Card interface power enable output	Yes	Yes	Yes

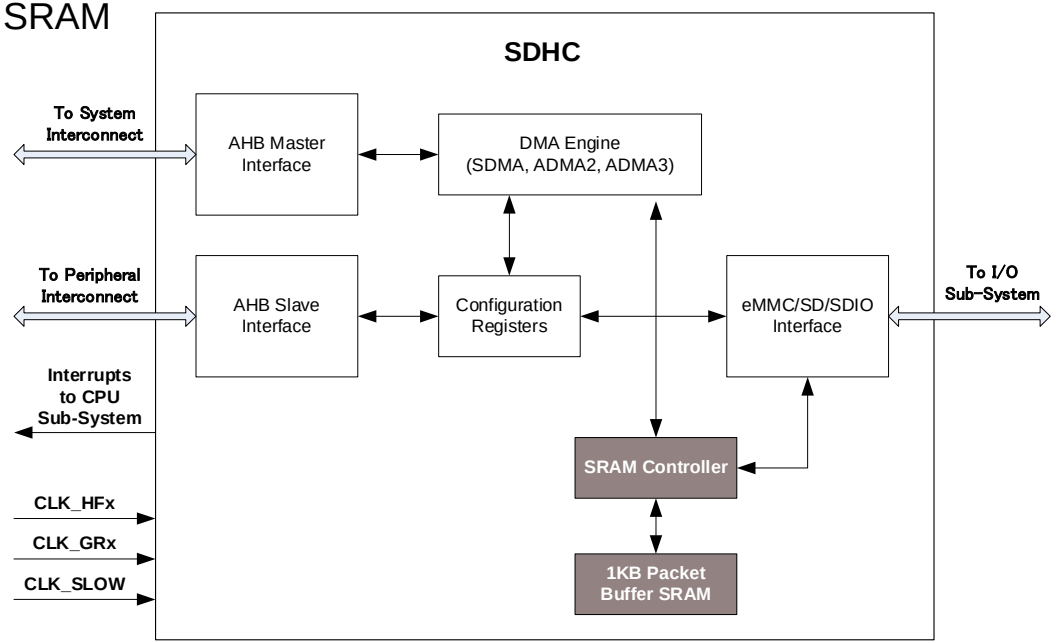
Hint Bar

Review TRM section 33.1 and 33.7, and Register TRM for additional details

[1] SDHC_CARD_DETECT_N and SDHC_CARD_MECH_WRITE_PROT should be connected to ground if an eMMC or an embedded SDIO device is connected.

SDHC Host Controller Block Diagram

- SDHC Host Controller components
  - SRAM
    - SRAM Controller
    - Packet Buffer SRAM



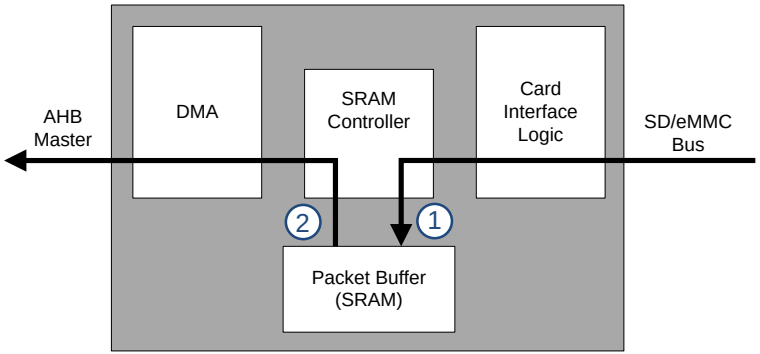
Hint Bar

Review TRM section 33.8 for additional details

Packet Buffer SRAM

- Packet buffer SRAM stores the data packets while carrying out data transfer to and from the card
  - SRAM size is 1KB to support buffering of two 512-byte blocks
    - Read Transfer:

Data Flow in Read Transfer



1. Received data from the card interface is written into packet buffer
2. When one block of data is received, DMA starts transmitting that data to the system by reading it from the packet buffer.

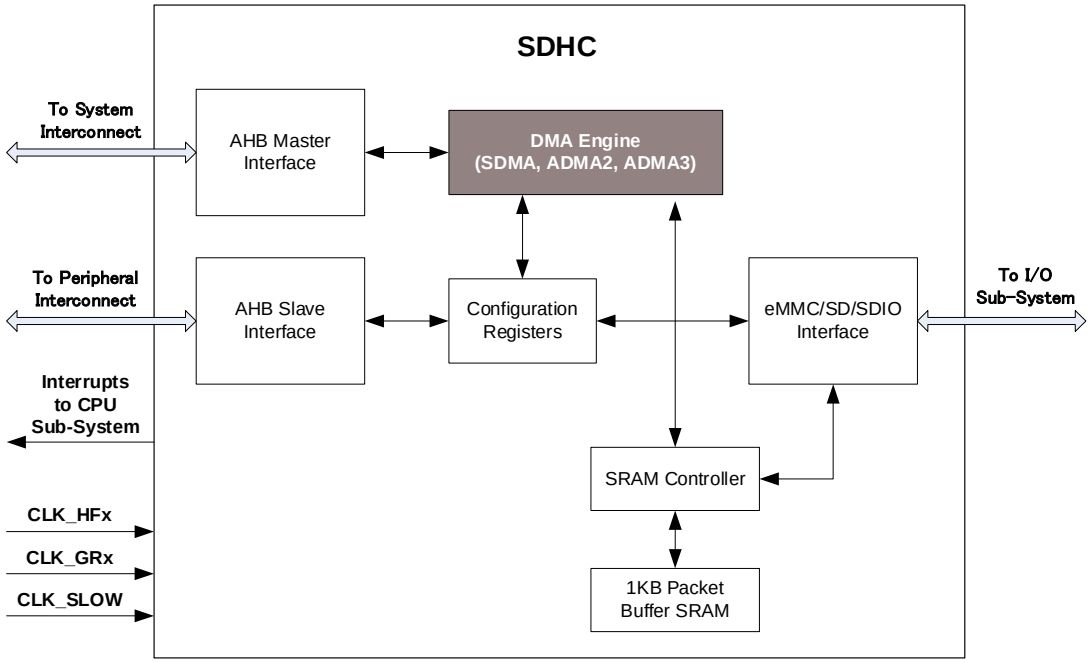
    - Write Transfer: DMA writes data into a packet buffer that is subsequently read by the card interface

Hint Bar

Review TRM section 33.8 and Register TRM for additional details

SDHC Host Controller Block Diagram

- SDHC Host Controller components
  - DMA Engine
    - SDMA
    - ADMA2
    - ADMA3



Hint Bar

Review TRM section 33.9 for additional details

Single Operation DMA (SDMA)

Advanced DMA (ADMA)

DMA Engine

- DMA engine handles data transfer between SDHC and system memory
- Features
  - Supports SDMA, ADMA2, and ADMA3 modes

Modes	Description	Max. Transfer Size
SDMA	SDMA transfers a data boundary by a read/write command	512KB
ADMA2	ADMA2 transfers multiple data boundaries by a read/write command	4GB
ADMA3	ADMA3 performs multiple ADMA2 operations	4EB

  - DMA engine enables new task descriptor fetches[1] while DMA is moving data during task execution[2]

Hint Bar

Review TRM section 33.9 for additional details

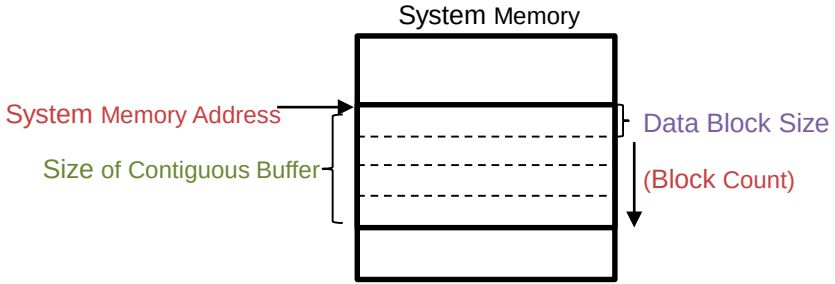
Single Operation DMA (SDMA)

Advanced DMA (ADMA)

[1] For CMD44 and CMD45
[2] For CMD46 and CMD47

SDMA Operation

- SDMA transfers a data boundary by a read/write command
  - Writing to the command register[1] triggers SDMA transfer
  - Configuration registers for transfer
    - System Memory Address or Block Count[2]:
      - SDMA uses ADMA System Address Register[3] when Block Count is specified[4]
    - Data Block Size[5]:
      - 1 byte to 2048 bytes
    - Size of Contiguous Buffer[6]:
      - 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, 512 KB



[1] CMD_R
[2] SDMASA_R.BLOCKCNT_SDMASA
[3] ADMA_SA_LOW_R
[4] When HOST_CTRL2_R.HOST_VER4_ENABLE = 1
[5] BLOCKSIZE_R.XFER_BLOCK_SIZE
[6] BLOCKSIZE_R.SDMA_BUF_BUF_BDARY

ADMA2 Operation

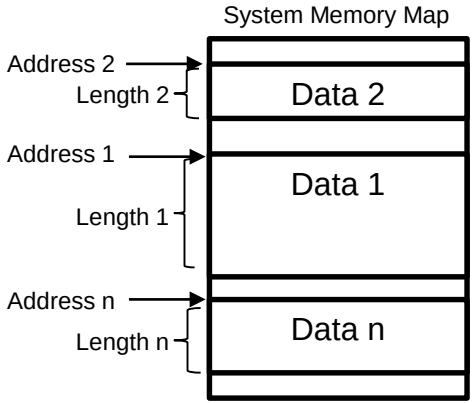
- ADMA2 transfers multiple data boundaries by a read/write command
  - Procedure
    1. Setting ADMA System Memory Address[1] for Descriptor Table start address
    2. Writing to the command register[2] triggers ADMA2 transfer
    3. ADMA2 fetches one descriptor line and executes it in the Descriptor Table
       - This is repeated until the end of the descriptor is found[3]
  - Data Address and Length
    - Minimum unit of address: 4 bytes
    - Maximum data length of each descriptor line: less than 64KB

Descriptor Table

ADMA System Memory Address

Multiple Read/Write Operation

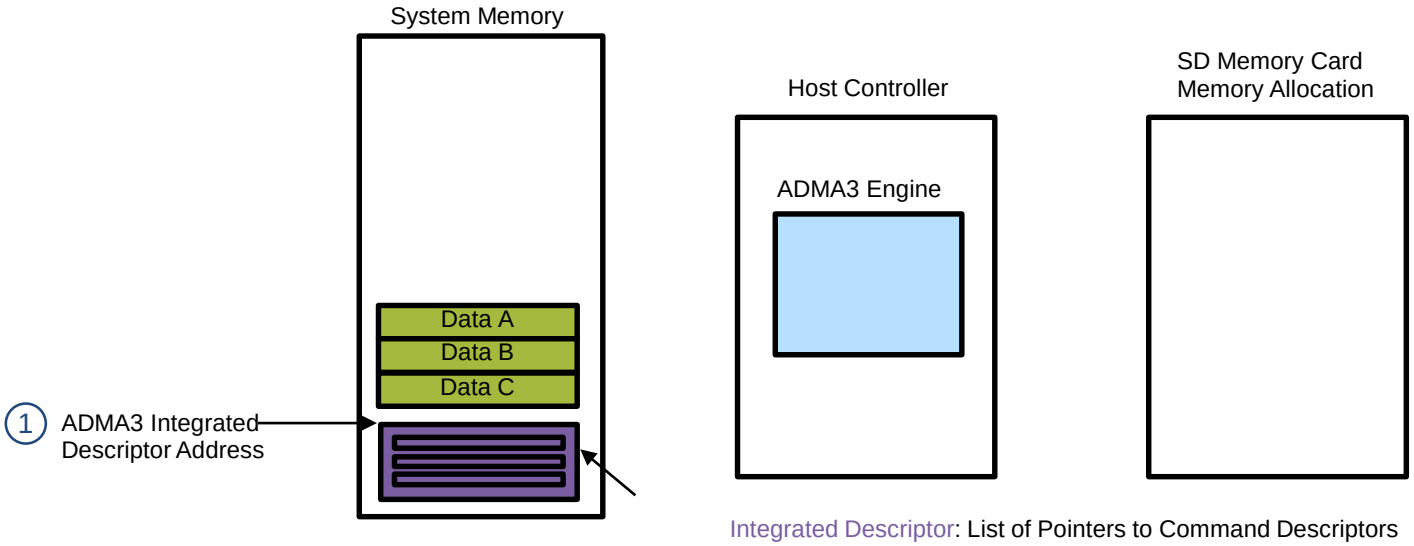
Address	Length	Attributes
Address 1	Length 1	Transfer
Address 2	Length 2	Transfer
…	…	…
Address n	Length n	Transfer & End



[1] ADMA_SA_LOW_R
[2] CMD_R
[3] End = 1 in attribute

ADMA3 Operation: Writing to Integrated Address Register

- ADMA3 performs multiple ADMA2 operations. The following slides describe each of the operations.
- Procedure:
  1. Writing to the ADMA3 Integrated Address register[1] triggers ADMA3

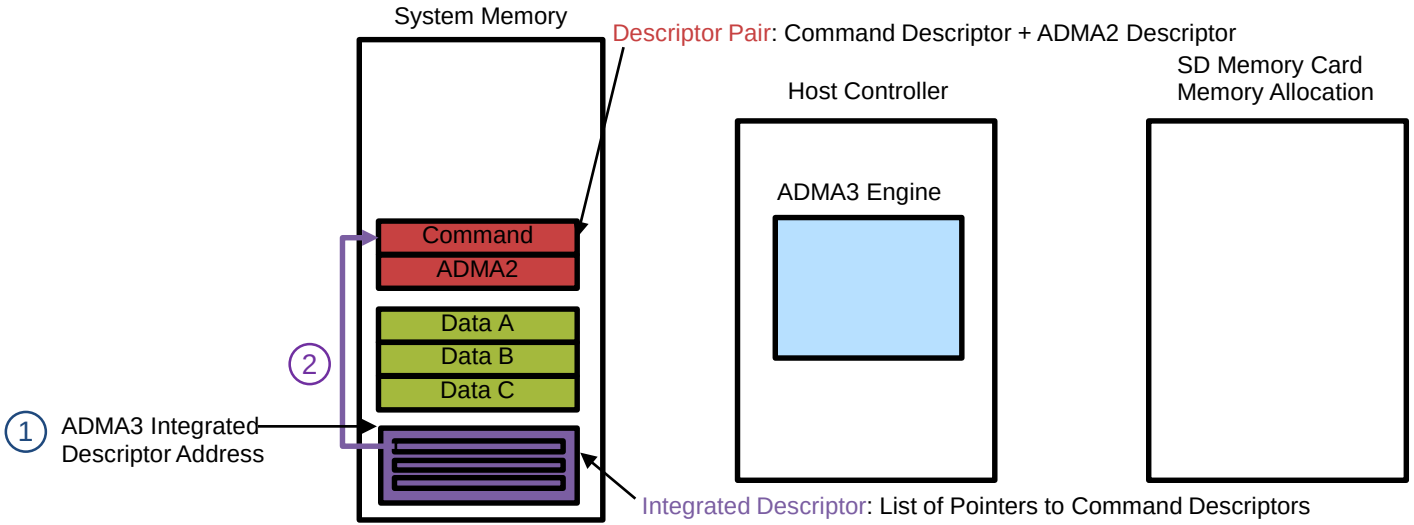


Integrated Descriptor: List of Pointers to Command Descriptors

[1] ADMA_ID_LOW_R

ADMA3 Operation: Fetching Pointer in Descriptor

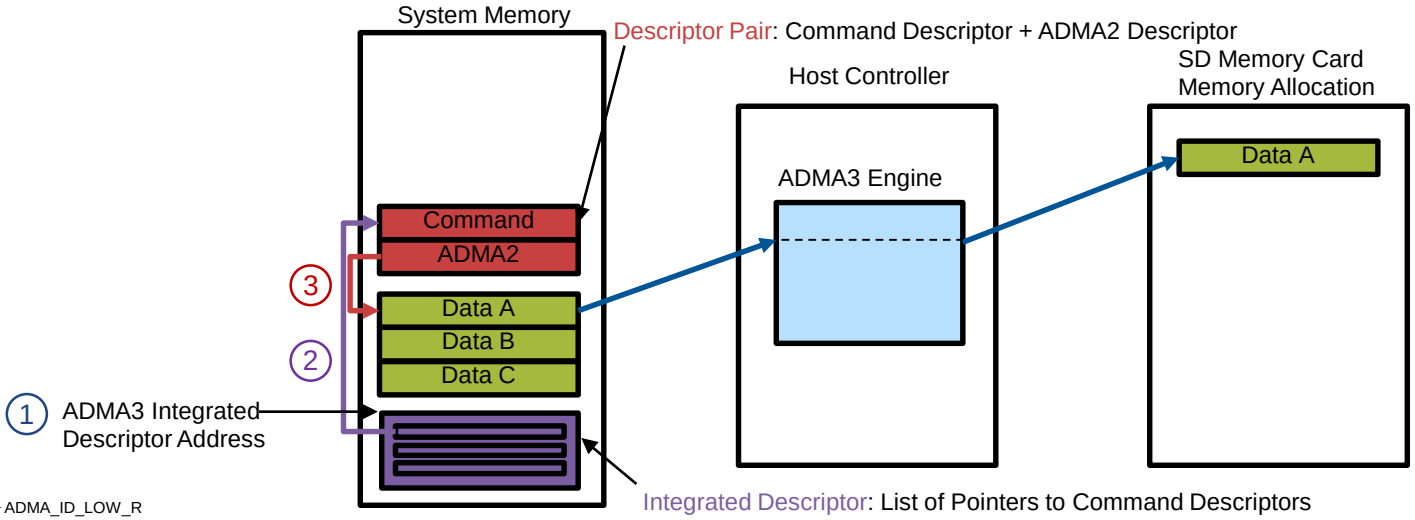
- Procedure:
  1. Writing to the ADMA3 Integrated Address register[1] triggers ADMA3
  2. ADMA3 fetches pointers one by one in the Integrated Descriptor and executes descriptors designated by the pointer



[1] ADMA_ID_LOW_R

ADMA3 Operation: Issuing a Command

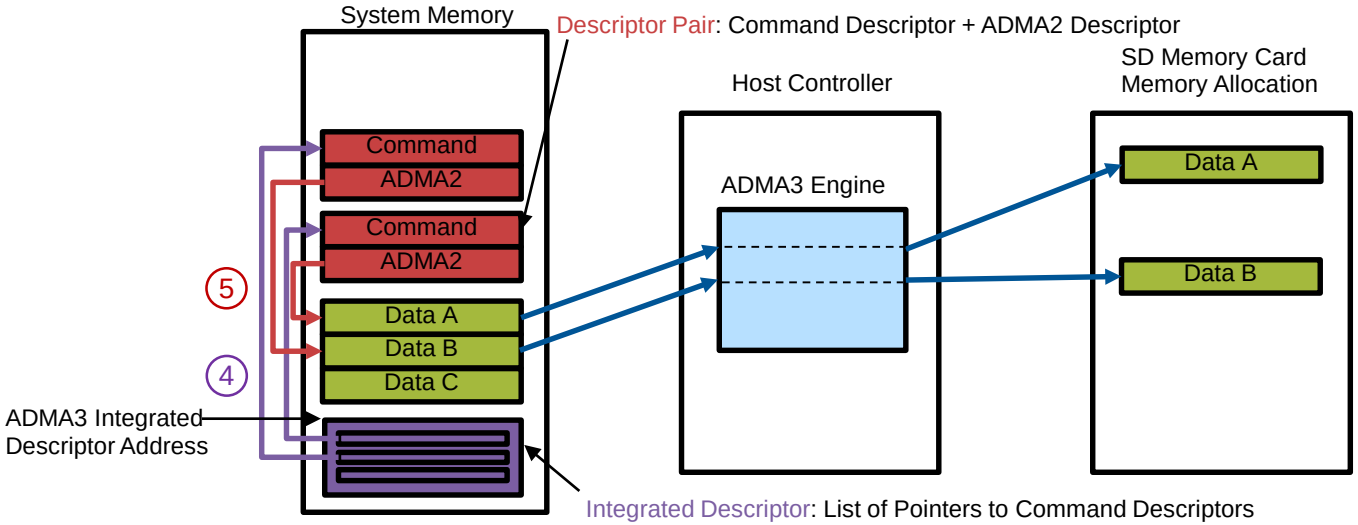
- Procedure:
  1. Writing to the ADMA3 Integrated Address register[1] triggers ADMA3
  2. ADMA3 fetches pointers one by one in the Integrated Descriptor and executes Descriptors designated by the pointer
  3. ADMA3 writes the Command Descriptor contents to the Host Controller registers to issue a command and then executes the ADMA2 Descriptor



[1] ADMA_ID_LOW_R

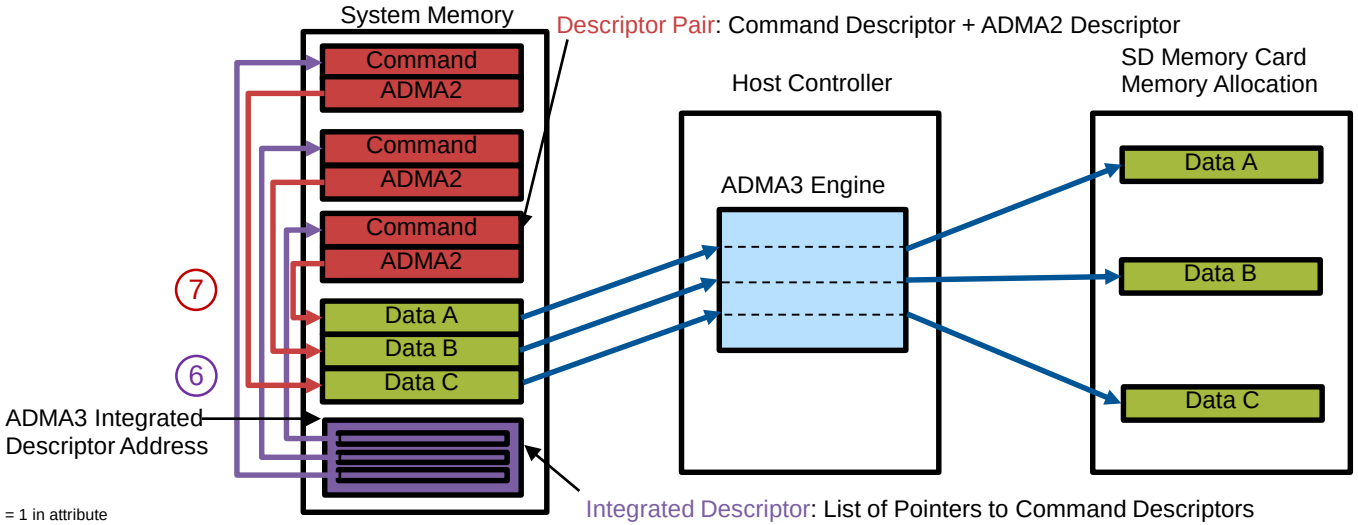
ADMA3 Operation: Executing Next Descriptor

- Procedure:
  - ④ ADMA3 fetches the next pointer in the Integrated Descriptor and executes descriptors designated by the pointer
  - ⑤ ADMA3 writes the next Command Descriptor contents to the Host Controller registers to issue a command and then executes the ADMA2 descriptor



ADMA3 Operation: Executing until the end of Descriptor

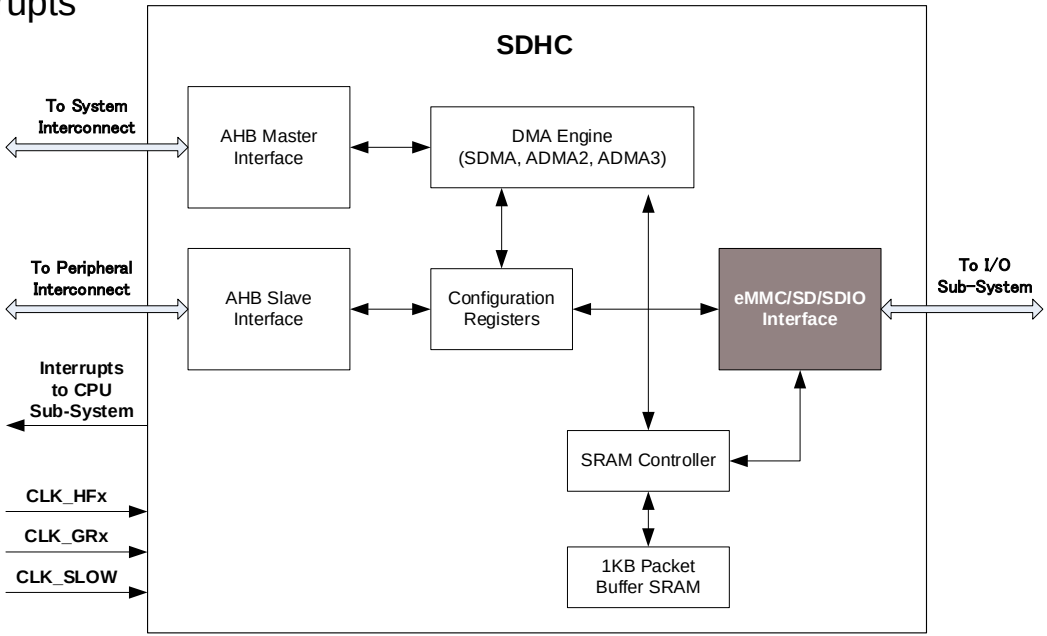
- Procedure:
  - ⑥ ADMA3 fetches the next pointer in the Integrated Descriptor and executes descriptors designated by the pointer until the end of the Command Descriptor[2]
  - ⑦ ADMA3 writes the end of the Command Descriptor contents to the Host Controller registers to issue a command and then executes the ADMA2 descriptor



[2] CDEnd = 1 in attribute

SDHC Host Controller Block Diagram

- SDHC Host Controller components
  - Interrupts
    - Wakeup Interrupts
    - General Interrupts



Hint Bar

Review TRM section 33.6 for additional details

Wakeup Interrupts (only for SD/SDIO mode)

- Wakeup SDHC Host Controller from Active/Sleep mode

Wakeup Interrupt	Set Condition
SD Card Insertion	Card Insertion
SD Card Removal	Card Removal
SDIO Card Interrupt	DAT[1] = 1

Hint Bar

Review TRM section 33.6 and Register TRM for additional details

Wakeup interrupts cannot wake up from DeepSleep and Hibernate mode

In DeepSleep mode, only CTL register is retained

General Interrupts (for eMMC/SD/SDIO mode)

- Triggered on all other events, in either normal conditions or error conditions

Normal Interrupt	Set Condition
Command Complete	Command Complete
Transfer Complete	Command execution is completed
Block Gap Event	Transaction stopped at block gap
DMA Interrupt	DMA Interrupt is generated
Buffer Write Ready	Ready to write buffer
Buffer Read Ready	Ready to read buffer
Card Insertion (SD mode only)	Card Inserted
Card Removal (SD mode only)	Card Removed
Card Interrupt (SDIO mode only)	DAT[1] = 1
FX Event	R[14] = 1 (Response register) and Response Type R1/R5 = 0 (Transfer Mode register)
Command Queuing Engine Event	Command Queuing related event has occurred (depends on Command Queuing Interrupt Status register)

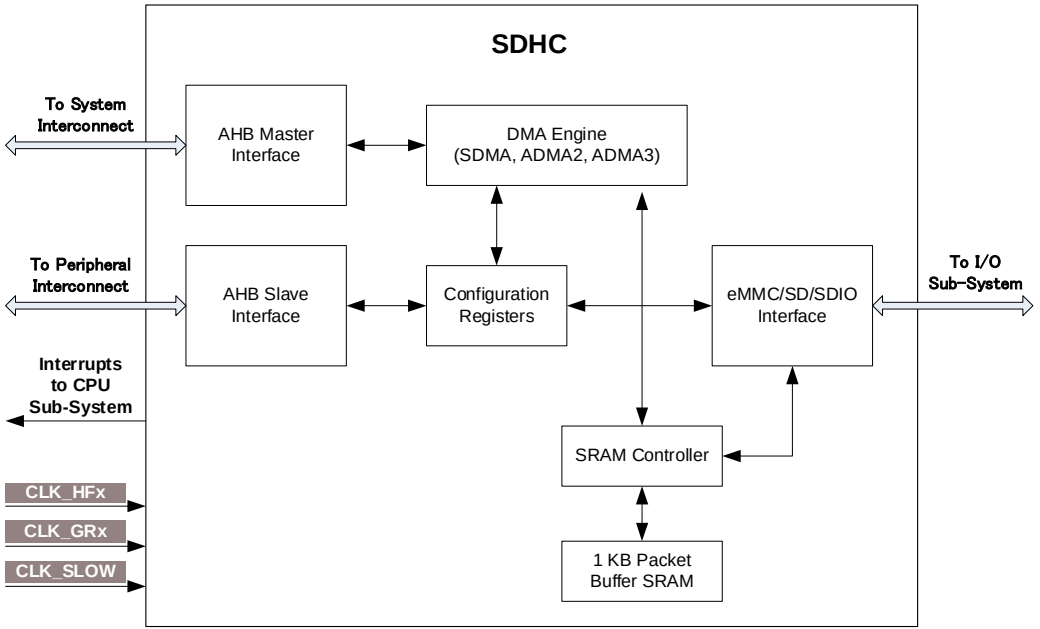
General Interrupts (for eMMC/SD/SDIO mode)

- Triggered on all other events, in either normal conditions or error conditions

Error Interrupt	Set Condition
Command Timeout Error	No response is returned within 64 SD clock cycles from the end bit of the Command
Command CRC Error	Command CRC error occurred
Command End Bit Error	Detected that the end bit of a command response is 0
Command Index Error	Command Index error occurred in the command response
Data Timeout Error	Detected one of the following timeout conditions: - Busy timeout for R1b, R5b type - Busy timeout after Write CRC status - Write CRC Status timeout - Read Data timeout
Data CRC Error	Detected CRC error when transferring read data, which uses the DAT line, when detecting the Write CRC status having a value of other than 010 or when write CRC status timeout occurs.
Data End Bit Error	Detected 0 at the end bit position of read data that uses the DAT line or at the end bit position of the CRC status.
Current Limit Error	Power fail occurred
Auto CMD Error	Detected that any of the bits D00 to D05 in Auto CMD Error Status register have changed from 0 to 1.
ADMA Error	ADMA error occurred
Response Error (SD/SDIO mode only)	An error is detected in a response

SDHC Host Controller Block Diagram

- SDHC Host Controller components
  - Clock
    - CLK_HFx[1]
    - CLK_GRx[2]
    - CLK_SLOW



Hint Bar

Review TRM section 33.3 for additional details

[1] It is connected to CLK_HF6 in CYT4BF. For other devices, refer to the respective device datasheet.
[2] It is connected to CLK_GR4 in CYT4BF. For other devices, refer to the respective device datasheet.

Clocks in SDHC Host Controller Block

› The following clocks are used in the SDHC Host Controller block:

Source	SDHC Host Controller Clock	Function
CLK_SLOW	Core SDHC Clock	Used for core SDHC functions including the packet buffer SRAM; it is sourced from the slow clock (CLK_SLOW); it must be ≥ AHB slave clock.
	AHB Master Interface Clock	Used by the AHB master interface; it is sourced from the slow clock (CLK_SLOW); it must be ≥ AHB slave clock.
CLK_GRx	AHB Slave Interface Clock	Used by the AHB slave interface; it is clocked by the PERI group clock (CLK_GRx); it must be ≥ CLK_CARD.
CLK_HFx	Base Clock / Card Clock	Used for sourcing the SD/eMMC interface clock (CLK_CARD); it is derived from CLK_HFx; it must be set to 100 MHz to be compatible with the Capabilities register[1].
	Timer Clock	Used for command and data timeout functions; it is derived from CLK_HFx.

CLK_SLOW ≥ CLK_GRx ≥ CLK_CARD

Hint Bar

Review TRM section 33.3 and Register TRM for additional details

[1] This register provides the Host Driver with information specific to the Host Controller

Appendix

Unsupported Features

- SD/SDIO operation in UHS-II mode
- Command queuing engine (CQE)
- eMMC boot operation in dual data rate mode
- Read wait operation by DAT[2] signaling in an SDIO card
- Suspend/resume operation in an SDIO card
- Interrupt input pins for embedded SD systems
- SPI protocol mode of operation
- SD UHS-I mode using 1.8-V signal voltage: SDR, SDR25, SDR50, and DDR50

Part of your life. Part of tomorrow.

Revision History

Revision	ECN	Submission Date	Description of Change
**	6401030	12/04/2018	Initial release
*A	6606574	06/28/2019	Added note descriptions in all pages. Updated page 3, 5, 6, 8, 9, 11, 15 to 20, 21, 23, 24.
*B	7044761	10/29/2020	Updated page 2.