SD Specifications

Part A2

SD Host Controller

Simplified Specification

Version 4.20

April 10, 2017

Technical Committee
SD Association

Revision History

Date	Version	Changes compared to previous issue
April 3, 2006	1.00	Host Controller Simplified Specification Version 1.00
February 8, 2007	2.00	Host Controller Simplified Specification Version 2.00
February 25, 2011	3.00	Host Controller Simplified Specification Version 3.00 (1) UHS-I Support (2) Shared Bus Support
April 10, 2017	4.20	Host Controller Simplified Specification Version 4.20 (1) UHS-II Support (2) ADMA3 Support (3) 64-bit System Addressing Support

Release of SD Simplified Specification/Addendum

The following conditions apply to the release of the SD Simplified Specification/Addendum by the SD Card Association. The Simplified Specification/Addendum is a subset of the complete version of SD Specification/Addendum that is owned by the SD Card Association.

Conditions for publication

Publisher and Copyright Holder:

SD Card Association
2400 Camino Ramon, Suite 375
San Ramon, CA 94583 USA
Telephone: +1 (925) 275-6615,
Fax: +1 (925) 886-4870
E-mail: office@sdcard.org

Notes:

This Simplified Specification/Addendum is provided on a non-confidential basis subject to the disclaimers below. Any implementation of the Simplified Specification/Addendum may require a license from the SD Card Association or other third parties.

Disclaimers:

Conventions Used in This Document

Naming Conventions

- Register names are shown in italic text such as *Present State*.
- Names of bits or fields within registers are in bold text such as **Buffer Write Enable**.
- Signal names are capitalized, bold and italic, followed by '#' if low active such as ***SDCD#.***
- Some terms are capitalized to distinguish their definition from their common English meaning. Words not capitalized have their common English meaning.
- Register names and the names of fields and bits in registers and headers are presented with the first letter capitalized and the remainder in lower case.

Numbers and Number Bases

- Hexadecimal numbers are written with a lower case "h" suffix, e.g., FFFFh and 80h.
- Binary numbers are written with a lower case "b" suffix (e.g., 10b).
- Binary numbers larger than four digits are written with a space dividing each group of four digits, as in 1000 0101 0010b.
- All other numbers are decimal.

Key Words

- May: Indicates flexibility of choice with no implied recommendation or requirement.
- Shall: Indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interchangeability and to claim conformance with the specification.
- Should: Indicates a strong recommendation but not a mandatory requirement. Designers should give strong consideration to such recommendations, but there is still a choice in implementation.

Special Terms

In this document, the following terms shall have special meaning:

- Host Controller — Refers to an SD Host Controller that complies with this Specification.
- Host Driver — Refers to the OS-specific driver for a Host Controller
- Card Driver — Refers to a driver for an SD/SDIO card or card function
- Host System (or System) — Refers to the entire system, such as a cellular phone, containing the Host Controller

Implementation Notes

- Some sections of this document provide guidance to Host Controller or Host Driver implementers. To distinguish non-mandatory guidance from other parts of the SD Host Specification, it will be shown as follows:

Implementation Note: This is an example of an implementation note.

Table of Contents

Table of Figures

Table of Tables

1. Overview of the SD Standard Host

The Secure Digital (SD) Host Standard Specification is the SD Association's (SDA) guideline for designing SD Host Controllers and related vendor products. Within the scope of the SD Associations adherence to this specification is not mandatory. It is the Host Controller vendor's responsibility to design products that comply with the SD Specification and where possible to use standard Host Drivers. OS vendor, IHVs and OEMs may require compliance according to their own policies so adherence is recommended.

1.1 Scope of the Standard SD Host

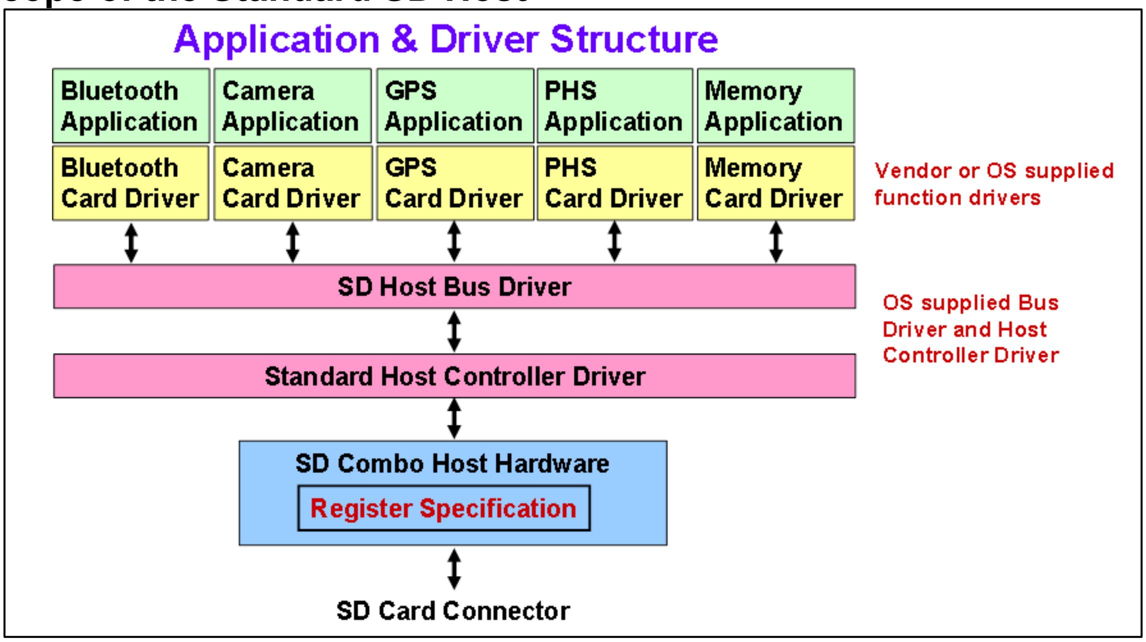


Figure 1-1 : Host Hardware and Driver Architecture

Defining a standard SD Host Controller is intended to promote increase of SD host products that can use SD memory cards and SDIO cards. Host Controller standardization enables Operating System (OS) Vendors to develop Host Driver (SD Host Bus Driver and Standard Host Controller Driver) that works with Host Controllers from any vendor.
Applications may in addition require the Card Drivers that supplied by card vendors or OS vendor. The Card Drivers communicate with the SD Host Bus Driver using a driver interface specified by the OS.

Implementation Note: This specification can be applied to any system bus interface. The interface between the Host Driver and its parent system driver (if any) is not defined by this specification.

1.2 Register Map

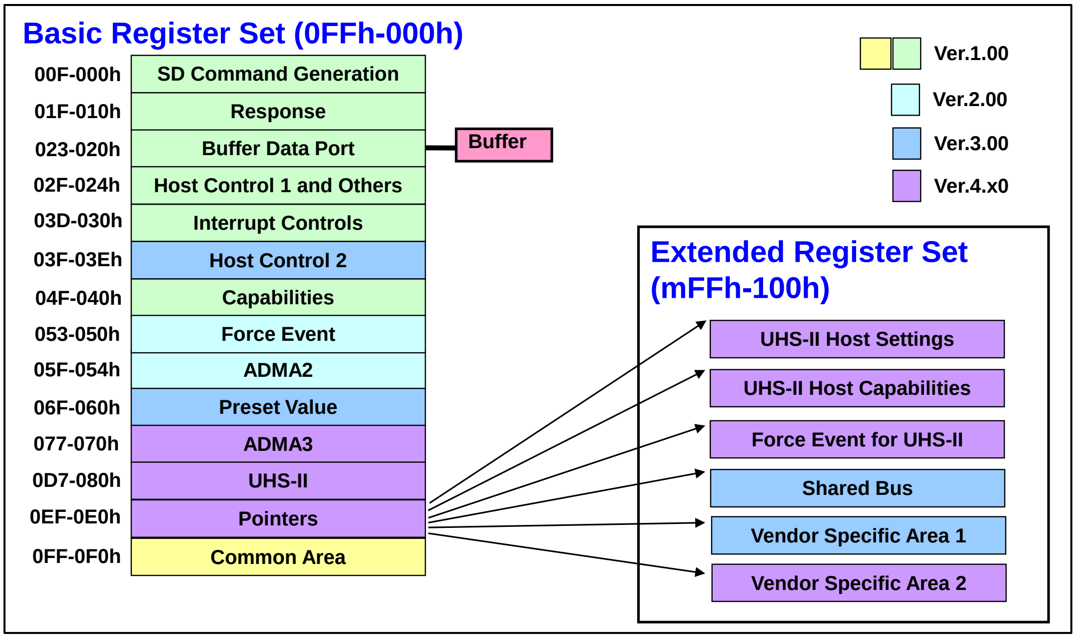


Figure 1-2 : Classification of the Host Controller Register Map

The standard register map is classified in 18 parts listed below. The Host Controller shall support byte, word and double word accesses to these registers. Reserved bits in all registers shall be fixed to zero. The Host Controller shall ignore writes to reserved bits; however, the Host Driver should write them as zero to ensure compatibility with possible future revisions to this Specification.

No	Register Name	Version					Comment
		1.00	2.00	3.00	4.00	4.10	
1	SD command generation	M	M	M	M	M	32-bit Block Count from Ver.4.10
2	Response	M	M	M	M	M	
3	Buffer Data port	M	M	M	M	M	
4	Host control 1 and Others	M	M	M	M	M	
5	Interrupt controls	M	M	M	M	M	Some fields will be added in later ver.
6	Capabilities	M	M	M	M	M	Some fields will be added in later ver.
7	Host Control 2	N/A	N/A	M	M	M	For UHS-I Support
8	Force Event	N/A	M	M	M	M	For test
9	ADMA2	N/A	O	M	M	M	
10	Preset Value	N/A	N/A	M	M	M	
11	ADMA3	N/A	N/A	N/A	N/A	O	ADMA3 from Ver.4.10
12	UHS-II	N/A	N/A	N/A	O	O	
13	Pointers	N/A	N/A	N/A	M	M	Mandatory if mFFh-100h is used
14	Common Area	M	M	M	M	M	
15	UHS-II Settings	N/A	N/A	N/A	O	O	Mandatory if UHS-II is supported
16	UHS-II Host Capabilities	N/A	N/A	N/A	O	O	Mandatory if UHS-II is supported
17	UHS-II Force Event	N/A	N/A	N/A	O	O	Mandatory if UHS-II is supported
18	Embedded Control	N/A	N/A	O	O	O	Moved to 1FFh-100h

M : Mandatory, O : Optional, N/A : Not Available

Table 1-1 : Supported Registers

1.3 Multiple Slot Support

One Standard Register Set is defined for each slot. If the Host Controller has two slots, two register sets is required. Each slot is controlled independently. This enables support for combinations of bus interface voltage, bus timing and SD Clock frequencies.

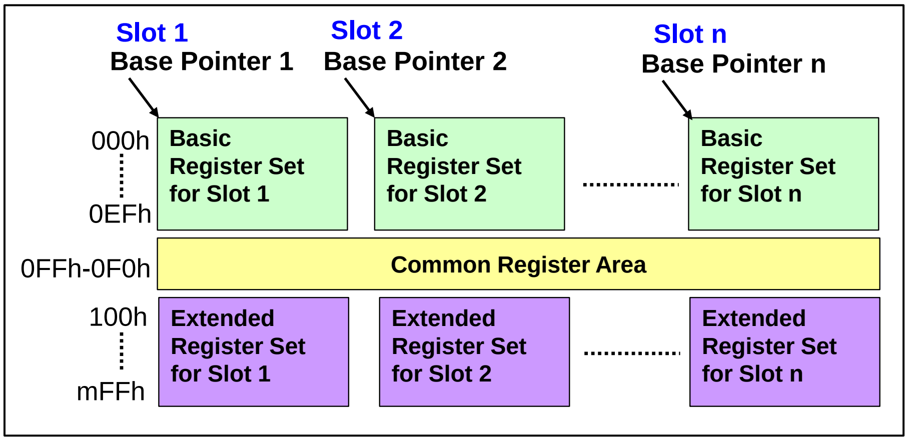


Figure 1-3 : Register Map for Multiple Slots Controller

Figure 1-3 shows the register map for a multiple slot Host Controller. The Host Driver shall determine the number of slots and base pointers to each slot's Basic Register Set using PCI Configuration register or vendor specific methods. Offsets from 0F0h to 0FFh are reserved for the Common register area that defines information for slot control and common status. The common register area is accessible from any slot's register set. This allows software to control each slot independently, since it has access to the *Slot Interrupt Status* register and the *Host Controller Version* register from each register set. From 100h to mFFh, Extended Register Set can be assigned. A parameter "m" denotes integer to determine size of the Extended Register Set.

1.4 Supporting DMA

The Host Controller provides a "programmed I/O" method for the Host Driver to transfer data using the *Buffer Data Port* register. Optionally, Host Controller implementers may support data transfer using DMA. The DMA algorithm defined in the SD Host Controller Standard Specification Version 1.00 is called SDMA (Single Operation DMA). Only one SD command transaction can be executed per an SDMA operation. Support of SDMA can be checked by the **SDMA Support** in the *Capabilities* register.

This specification defines a DMA transfer algorithm called ADMA (Advanced DMA). ADMA provides data transfer between system memory and SD card without interruption of CPU execution. Support of ADMA can be checked by the *Capabilities* register. Refer to Section 1.13 for more details about ADMA. When the term "DMA" is used in this document, it applies to both SDMA and ADMA.

Prior to using DMA, the Host Driver shall confirm that both the Host Controller and the system bus support it (PCI bus can support DMA). DMA shall support both single block and multiple-block transfers. Host Controller registers shall remain accessible for issuing non-DAT line commands during a DMA transfer execution (Not applicable to UHS-II mode). The result of a DMA transfer shall be the same regardless of the system bus data transfer method.

The Host Driver can stop and restart a DMA operation by the control bits in the *Block Gap Control* register. By setting **Stop At Block Gap Request**, a DMA operation can be stopped at block gap. By setting **Continue Request**, DMA operation can be restarted. Refer to the *Block Gap Control* register for more details. If an error occurs, DMA operation shall be stopped. Synchronous abort described in Section 3.8.3 should be used to abort DMA transfer instead of asynchronous abort to ensure cancellation of DMA does not affect system bus operation. Stop operating DMA by using **Stop At Block Gap Request** and then issue abort command to stop card data transfer. After that, Host Driver resets the Host Controller by using the **Software Reset For DAT Line** in the *Software Reset* register (SD mode) or the **Host SD-TRAN Reset** in the *UHS-II Software Reset* register (UHS-II mode).

1.5 SD Command Generation

1.5.1 SD Mode Command Generation

	SDMA Command	ADMA Command	CPU data Transfer	Non-DAT Transfer
32-bit Block Count / (SDMA System Address)	Note 1	Note 1,2,3	Note 1,2,3	No (Protected)
Block Size	Yes	Yes	Yes	No (Protected)
16-bit Block Count	Note 3	Note 3 Refer to 1.15.3	Note 3	No (Protected)
Argument	Yes	Yes	Yes	Yes
Transfer Mode	Yes	Yes	Yes	Note 4
Command	Yes	Yes	Yes	Yes

Note 1: If **Host Version 4 Enable** is set to 1 in the *Host Control 2* register, this register is used for *32-bit Block Count* instead of *SDMA System Address*. SDMA start address is moved to *ADMA System Address* register.

Note 2: Auto CMD23 is supported from Version 3.00 and setting of this register is set to the argument of CMD23 when Auto CMD23 is executed. If **Host Version 4 Enable** =0, Auto CMD23 cannot be used with SDMA. Host Controller Version 4.10 re-defines this register as *32-bit Block Count* so that all data transfer modes may use *32-bit Block Count* when **Host Version 4 Enable** =1.

Note 3: Version 2.00 or later uses *16-bit Block Count* register or ADMA2 total length to determine transfer length. Additionally, Version 4.10 may use *32-bit Block Count register,* which is selected when **Host Version 4 Enable** is set to 1 and **16-bit Block Count** is set to 0000h.

Note 4: If **Host Version 4 Enable** = 0, "No (Protected)". If **Host Version 4 Enable** = 1, "Yes".

Table 1-2 : Registers to Generate SD Command

Table 1-2 shows register settings (at offsets from 000h to 00Fh in the register set) necessary for three types of transactions: SDMA generated transfers, ADMA generated transfers, CPU data transfers (using "programmed I/O") and non-DAT transfers. When initiating a transaction, the Host Driver should program these registers sequentially from 000h to 00Fh. The beginning register offset may be calculated based on the type of transaction. The last written offset shall be always 00Fh because writing to the upper byte of the *Command* register shall trigger issuance of an SD command.

The Host Driver should not read the *SDMA System Address*, *Block Size* and *Block Count* registers during a data transaction unless the transfer is stopped because the value is changing and not stable. To prevent destruction of registers using data transfer when issuing command, the *32-bit Block Count, Block Size*, *16-bit Block Count* and *Transfer Mode* registers shall be write protected by the Host Controller while **Command Inhibit (DAT)** is set to 1 in the *Present State* register. (When Host Version 4 Enable =0, the *SDMA System Address* is not protected by this signal.) The Host Driver shall not write the *Argument* and *Command* registers while **Command Inhibit (CMD)** is set to 1.

1.5.2 UHS-II Mode Command Generation

UHS-II Command Packet is generated by setting following registers.

(1) UHS-II Block Size Register for DCMD
(2) UHS-II Block Count Register for DCMD
(3) UHS-II Command Packet Register for CCMD and DCMD
(4) UHS-II Transfer Mode Register for CCMD and DCMD
(5) UHS-II Command Register for CCMD and DCMD

Theses registers are correspondent to that of SD Mode registers. UHS-II command packet image is set to *UHS-II Command Packet* register. Host Controller does not analyze this register to issue a command packet. On writing to *UHS-II Command* register, Host Controller issues a command packet.

1.5.2.1 Command Issuing during CTS

When CCMD is issued at CTS of DCMD execution, the RES packet of CCMD is set to the *UHS-II Response* register. If CCMD is other than reset or abort command, DCMD execution continues. For example, this feature may be used for issuing CMD52 during CMD53 execution in case of SDIO. Refer to Section 1.17 for more details about response error check.

1.5.2.2 Support of TID Check

Host Controller Version 4.10 supports TID. Host Controller compares TID of command with TID of all received packets of the same transaction. If TID is not matched, TID Error is set to 1 in the *UHS-II Error Interrupt Status* register and the transaction shall be stopped.
In case of command issuing during CTS, it works if TID in DCMD and TID in CCMD during CTS are set to 0. It also works if different values other than 0 are set to them.

1.6 Suspend and Resume Mechanism (Version 3.00 or less)

Suspend/Resume function is not supported from Version 4.00.

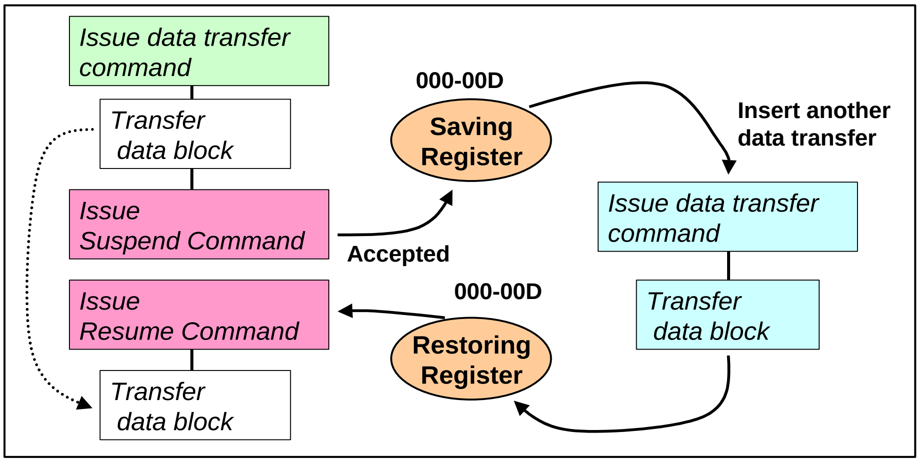


Figure 1-4 : Suspend and Resume Mechanism

Support for Suspend/Resume can be determined by checking **Suspend/Resume Support** in the *Capabilities* register. When the SD card accepts a suspend request, the Host Driver saves information in the first 14 bytes registers (that is, offsets 000h-00Dh) before issuing other SD commands. On resuming, the Host Driver restores these registers and then issues the Resume command to continue suspended operation.

The SDIO card sets the **DF** (Resume Data Flag) in the response to the Resume command. (Since the Suspend and Resume commands are CMD52 operations, the response data is actually the Function Select Register in the CCCR.) If **DF** is set to 0, it means the SDIO card cannot continue data transfer while suspended. This bit can be used to control data transfers and interrupt cycles. If the Resume Data Flag is set to 0, no more data is transferred and an interrupt cycle is started if the transaction being resumed is in 4-bit mode. If **DF** is set to 1, data transfers continue. The Suspend/Resume protocol is described in the SDIO Specification (SD Specifications Part E1).

Note: To use Suspend/Resume function, it is necessary that SDIO Card supports the Suspend and Resume commands and Read Wait control.

1.7 Buffer Control

The Host Controller has a data buffer for data transfer. The Host Driver accesses internal buffer through the 32-bit *Buffer Data Port* register. DMA also uses internal buffer to control data transfer between system memory and SD Card.

Buffer Size is determined by setting of block size. Data Transfer Size is determined by setting of block size and block count.
In SD mode, block size is set by *Block Size* register (Offset 004h) and block count is set by 16-bit *Block Count* register (Offset 006h). 16-bit block count limits the maximum data transfer length. The buffer size is determined by setting of the *Block Size* register.
In UHS-II mode, block size is set by *UHS-II Block Size* register (Offset 080h) and block count is set by 32-bit *UHS-II Block Count* register (Offset 084h). There is no limit of data transfer length by 32-bit block count. The buffer size is determined by setting of the Block Size register and **N_FCU** in the *UHS-II Settings* register as follows:
Buffer Size in UHS-II mode = **UHS-II Block Size** * **N_FCU** (Settings)

Followings Sections show some rules to access the buffer.

1.7.1 Control of Buffer Pointer (Non DMA)

Internally, the Host Controller maintains a pointer to control the data buffer. The pointer is not directly accessible by the Host Driver. Every time the *Buffer Data Port* register is accessed, the pointer is incremented depending on amount of data written to the buffer. In order to accommodate a variety of system busses, this pointer shall be implemented regardless of system bus width (8-bit, 16-bit, 32-bit or 64-bit system bus width can be supported). To specify control of the pointer, the Host Controller data buffer interface shall have the following characteristics:

(6) System Bus Width and Byte Enable Address
8-bit, 16-bit, 32-bit or 64-bit system bus is supported. To specify byte position for *Buffer Data Port* register (4 bytes), Byte Enable (***BE***[]) or Lower Address (***A***[]) is used. Table 1-3 shows the relation between lower address and byte enable depending on system bus width. The *Buffer Data Port* register can be accessed by ***BE***[3:0] for 64-bit system bus, which has ***BE***[7:0].

System Bus	A[02]	A[01]	A[00]	BE[3] D[31:24]	BE[2] D[23:16]	BE[1] D[15:08]	BE[0] D[07:00]
64-bit	No	No	No	Yes	Yes	Yes	Yes
32-bit	Yes	No	No	Yes	Yes	Yes	Yes
16-bit	Yes	Yes	No	No	No	Yes	Yes
8-bit	Yes	Yes	Yes	No	No	No	Yes*2

*1 "Yes" means the signal is used for control and "No" means the signal is not used.
*2 : BE[00] for 8-bit bus is always 1 therefore it may not be defined.

Table 1-3 : Relations between Address and Byte Enable

(7) Sequential and continuous access
The *Buffer Data Port* register shall be accessed by sequential and continuous manner. The buffer pointer is controlled by the Byte Enable patterns when accessing to the Buffer Data Port register. Therefore, Byte Enable patterns shall be sequential and continuous as well. The order of Byte Enable is according to little endian format. For example, ***BE***[1] is accessed, next access shall start form ***BE***[2]. Random or skipped access is not allowed.
Table 1-4 shows possible byte enables patterns that shall be supported by the Host Controller. However, if the system controller supports write merge, it may generate the other byte enable patterns. To avoid generating unsupported byte enable patterns for the 32-bit or 64-bit bus

system, the Host Driver is allowed to use word or double word access to the *Buffer Data Port* register except for the last access to every block data.

OK	BE[3:0]=0011b (2-byte) =>	BE[3:0]=1100b (2-byte) =>	BE[3:0]=0011b (2-byte)
OK	BE[3:0]=1100b (2-byte) =>	BE[3:0]=1111b (4-byte) =>	BE[3:0]=0011b (2-byte)
OK	BE[3:0]=1111b (4-byte) =>	BE[3:0]=1111b (4-byte) =>	BE[3:0]=1111b (4-byte)
Not OK	BE[3:0]=0011b (2-byte) =>	BE[3:0]=0011b (2-byte)	(Cannot skip BE[2],BE[3])
Not OK	BE[3:0]=0011b (2-byte) =>	BE[3:0]=1111b (4-byte)	(Cannot skip BE[2],BE[3])

Byte Enable		BE[3]	BE[2]	BE[1]	BE[0]
Data Bus		**D[31:24]**	**D[23:16]**	**D[15:08]**	**D[07:00]**
Access Type	**4-byte**	1	1	1	1
	2-byte	0	0	1	1
	2-byte	1	1	0	0
	1-byte	0	0	0	1
	1-byte	0	0	1	0
	1-byte	0	1	0	0
	1-byte	1	0	0	0

* 1 means BE is valid and 0 means BE is not valid.

Table 1-4 : Available Byte Enable Pattern for Buffer Data Port Register

(8) Buffer Control with Block Size
The buffer preserves data up to the block size specified by the *Block Size* register. Following definitions of controlling buffer enable the Host Driver to access the *Buffer Data Port* register repeatedly with 32-bit width regardless of block size.

In case of write operation, the buffer accumulates the data written through the *Buffer Data Port* register. When the buffer pointer reaches the block size, **Buffer Write Enable** in the *Present State* register changes 1 to 0. It means no more data can be written to the buffer. Excess data of the last write is ignored. For example, if just lower 2 bytes data can be written to the buffer and a 32-bit (4-byte) block of data is written to the *Buffer Data Port* register, the lower 2 bytes of data is written to the buffer and the upper 2 bytes is ignored. Every time **Buffer Write Enable** changes 0 to 1, it means a next block of data can be written to the buffer. A new blocks write shall always start from BE[00] position. After that, a block of data can be written to the buffer without checking **Buffer Write Enable**.

In case of read operation, every time **Buffer Read Enable** in the *Present State* register changes 0 to 1, a block of data can be read through the *Buffer Data Port* register. A new block read shall always start from BE[00] position. After that, a block of data can be read from the buffer without checking **Buffer Read Enable**. Excess data of the last read is ignored. For example, if just lower 2 bytes of data are left in the buffer and a 32-bit (4-byte) read is performed, the lower 2 bytes is valid but the upper 2 bytes is undefined. When the buffer pointer reaches block size, **Buffer Read Enable** changes 1 to 0. It means no more data can be read from the buffer.

Implementation Note:
Table 1-4 implies that the Host Driver should align register accesses on address boundaries matching the number of bytes in the access. That is, single byte accesses may be aligned on any offset within the register set; word (double byte) accesses should be aligned on two-byte offsets; and double-word (quad byte) accesses should be aligned on four-byte offsets. According to the feature (3), the Host Driver can always access *Buffer Data Port* register with double-word access.

1.7.2 Determining Buffer Block Length

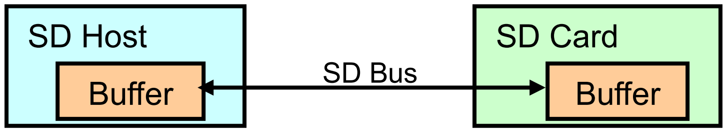


Figure 1-5 : Buffer Size Relation between Host and Card

To be able to transfer blocks of data at a burst, the relationship between Host Controller and SD card buffer sizes is important. The Host Driver shall use the same buffer size for both Host Controller and SD Card. The buffer size is determined by block size. If the maximum block size of Host Controller and SD Card is different (capability), the Host Driver shall use the smaller one as the maximum block size of block size register. The maximum Host Controller buffer size is defined by the **Max Block Length** field in the *Capabilities* register in SD mode and the **Host Maximum Block Length** field in the *UHS-II LINK/TRAN Capabilities* register.

Implementation Note: The card buffer size is described as maximum block length in the Card Specific Data (CSD) register for memory cards (READ_BL_LEN and WRITE_BL_LEN) and in the CCCR (Function 0) and FBR (Function 1-7) for SDIO cards. The Physical Layer Specification re-defines that the maximum block length is only used to calculate capacity of memory card. Even though it indicates larger than 512 bytes, block length shall be set to 512 byte for data transfer. This is because 512 bytes block length is required to keep compatibility with 512 bytes data boundary. The Host Controller shall have at least 512 bytes buffer and 512 bytes fixed block length is used for memory data transfer. UHS-II has a parameter of N_FCU, which indicates the number of blocks per flow control unit. Host Controller requires at least 512 bytes * N_FCU buffer size. In case of SDIO, buffer size is variable up to the maximum block length. If multiple functions SDIO card has different buffer size in each function, Card Driver should adjust buffer size depends on the maximum block length of each function.

1.7.3 Dividing Large Data Transfer

On transferring very large data, Card Driver should divide the data into small unit for avoiding an operation continues to hold SD Bus Interface long time. Small data unit access allows time-sharing operation and several applications may use a common SD Card at the same time.
Following the Speed Class write conditions (defined by the Physical Layer Specification) is the most efficient method to write data to SD Memory card.

Data transfer size of CMD53 is limited by the 9-bit block count field in the argument. Up to 511 blocks can be transferred per this command.

1.8 Relationship between Interrupt Control Registers

The Host Controller implements a number of interrupt sources. Interrupt sources can be enabled as interrupts or as system wakeup signals as shown in Figure 1-6. If the interrupt source's corresponding bit in the *Normal Interrupt Status Enable* or *Error Interrupt Status Enable* register is 1 and the interrupt becomes active, its active state is latched and made available to the Host Driver in the *Normal Interrupt Status* register or the *Error Interrupt Status* register. Interrupt Status shall be cleared when *Interrupt Status Enable* is cleared. (This is not expressed in the Figure 1-6.)

An interrupt source with its bit set in an interrupt status register shall assert a system interrupt signal if its corresponding bit is also set in the *Normal Interrupt Signal Enable* register or the *Error Interrupt Signal Enable* register. Once signaled, most interrupts are cleared by writing a 1 to the associated bit in the interrupt status register. Card interrupts, however, shall be cleared by the Card Driver. If the Card Interrupt is generated, the Host Driver may clear Card Interrupt Status Enable to disable card interrupts while the Card Driver is processing them. After all interrupt sources are cleared, the Host Driver sets it again to enable another card interrupt. Disabling the Card Interrupt Status Enable avoids generating multiple interrupts during processing interrupt service.

The *Wakeup Control* register enables **Card Interrupt**, **Card Insertion,** or **Card Removal** status changes to be configured to generate a system wakeup signal. These interrupts are enabled or masked independently of the *Normal Interrupt Signal Enable* register. The kind of wakeup event can be read from the *Normal Interrupt Status* register.
The interrupt signal and wakeup signal are logical ORed and shall be read from the *Slot Interrupt Status* register.

Implementation Note: The Host Driver is responsible for enabling wakeup signals and disabling interrupt signals when the Host System goes into its sleep mode, and for disabling wakeup signals and enabling interrupt signals when the Host System goes into run mode. The Host Driver should not enable both at the same time.

Implementation Note: The Host Systems may implement interrupt and wakeup signals in various ways. For example, the PCI bus supports ***PME#***, which can be asserted without PCI clock, then interrupts use ***INTx#*** and wakeups use ***PME#***. Alternatively, the system may use an ORed signal of interrupt and wakeup if the system bus supports one interrupt line to the Host Controller.

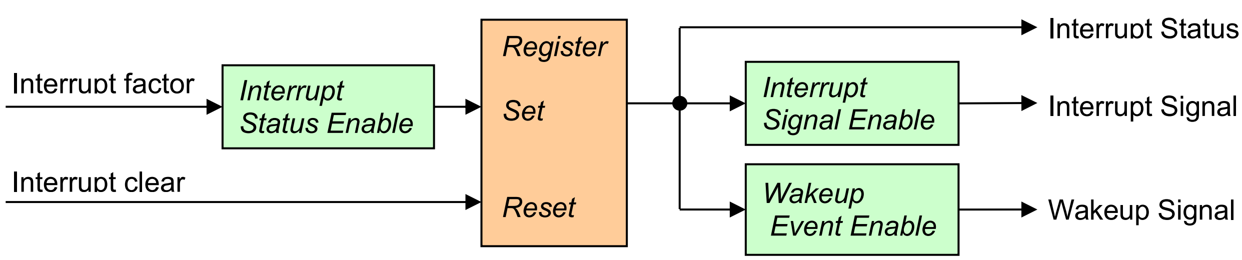


Figure 1-6 : Logical Relation for Interrupt Registers

Interrupt Status Enable	Interrupt Signal Enable	Wakeup Event Enable	Interrupt Status	Interrupt Signal
0 (Mask)	x (don't care)	x (don't care)	0 (Not exist)	0 (De-assert)
1 (Enable)	0 (Mask)	x (don't care)	x (don't care)	0 (De-assert)
1 (Enable)	1 (Enable)	x (don't care)	0 (Not exist)	0 (De-assert)
1 (Enable)	1 (Enable)	x (don't care)	1 (Exist)	1 (Assert)

Table 1-5 : Interrupt Signal Table

Interrupt Status Enable	Interrupt Signal Enable	Wakeup Event Enable	Interrupt Status	Wakeup Signal
0 (Mask)	x (don't care)	x (don't care)	0 (Not exist)	0 (De-assert)
1 (Enable)	x (don't care)	0 (Mask)	x (don't care)	0 (De-assert)
1 (Enable)	x (don't care)	1 (Enable)	0 (Not exist)	0 (De-assert)
1 (Enable)	x (don't care)	1 (Enable)	1 (Exist)	1 (Assert)

Table 1-6 : Wakeup Signal Table

Implementation Note: The Host Controller may implement asserted wakeup or interrupt signals as active high or active low.

1.9 HW Block Diagram and Timing Part

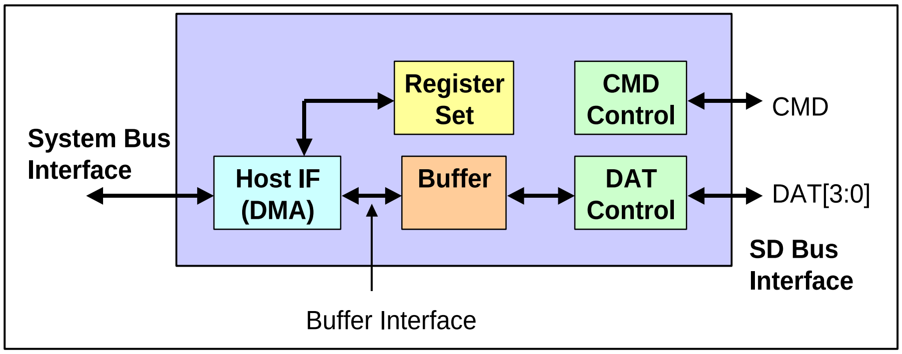


Figure 1-7 : Block Diagram of Host Controller

The Host Controller has two bus interfaces, the System Bus Interface and the SD Bus Interface. The Host Controller assumes that these interfaces are asynchronous (that is, are working on different clock frequencies). The Host Driver is on system bus time (because it is software executed by the Host Controller CPU, on its system clock). The SD card is on SD Bus time (that is, its operation is synchronized by ***SDCLK***). The Host Controller shall synchronize signals to communicate between these interfaces. Blocks of data shall be synchronized at the buffer module. All status registers shall be synchronized by the system clock and maintain synchronization during output to the system interface. Control registers, which trigger SD Bus transactions, shall be synchronized by ***SDCLK***. Therefore, there will be a timing delay when propagating signals between the two interfaces. This means the Host Driver cannot do real time control of the SD Bus and needs to rely on the Host Controller to control the SD Bus according to register settings.

The Buffer Interface enables internal read and write buffers (Refer to use of the **Buffer Read Enable** and **Buffer Write Enable** in the *Present State* register as described in Section 1.7 "Buffer Control"). The **Transfer Complete** interrupt status indicates completion of the read / writes transfer for both DMA and non-DMA transfers. However, the timing of data transfer completion is different between reads and writes. Read transfers shall be completed after all valid data have been transferred to the Host System and are ready for the Host Driver to access. Write transfers shall be completed after all valid data have been transferred to the SD card and the busy state is over. If all block data is written to buffer, Host Driver should ignore another Buffer Write Ready until Transfer Complete is generated.

Table 1-7 shows the relation between statuses and interrupts for data transfer.

Type of Data transfer	Buffer Status	Buffer Interrupt	Complete Interrupt
Write Transfer (Non DMA)	Buffer Write Enable	Buffer Write Ready	Transfer Complete
Write Transfer (DMA)	(Driver ignores)	(Driver ignores)	Transfer Complete
Read Transfer (Non DMA)	Buffer Read Enable	Buffer Read Ready	Transfer Complete
Read Transfer (DMA)	(Driver ignores)	(Driver ignores)	Transfer Complete

Table 1-7 : Summary of Register Status for Data Transfer

1.10 Power State Definition of SD Host Controller

Implementation Note: Table 1-8 defines controller power states, which are listed in increasing order of power consumption. The Host Controller should reduce the power consumption by using these conditions.

SD Card	Internal Clock *1	SD Power	SD Clock	SD Bus	Power State *2	Comment
No exist	Stop	OFF	Stop	-	P00	Host not used
	Oscillate	OFF	Stop	-	P01	No card
	Oscillate	ON	Stop	-	P02	Short transition state *3
	Oscillate	ON	Oscillate	-	P03	Short transition state *3
Exist	Stop	OFF	Stop	-	P10	Host not used
	Oscillate	OFF	Stop	-	P11	Low power mode
	Oscillate	ON	Stop	-	P12	Wakeup
	Oscillate	ON	Oscillate	Wait	P13	Ready to issue command
	Oscillate	ON	Oscillate	Access	P14	During transaction

Table 1-8 : Power State Definition

Implementation Note:
*1: Internal clock should be stopped when the Host System does not use the Host Controller.
*2: Power states are not actually implemented in Host Controller. This label is for reference.
*3: Short transition state: Temporary power states. The Host Controller automatically goes to P01 when it detects No Card.

The SD Clock shall not be supplied when card power is OFF.
States described in Table 1-8 can be determined by reading the corresponding register bits:

Internal clock oscillate/stop	**Internal Clock Enable** in the *Clock Control* register
SD Card : Exist/Not exist	**Card Inserted** in the *Present State* register
SD Power : ON/OFF	**SD Bus Power** in the *Power Control* register
SD Clock : oscillate/stop	**SD Clock Enable** in the *Clock Control* register
SD Bus : access/wait (idle)	**Command Inhibit (CMD)** and **Command Inhibit (DAT)** in the *Present State* register

1.11 Auto CMD12

Multiple block transfers for SD memory require CMD12 to stop the transactions. The Host Controller automatically issues CMD12 when the last block transfer is completed. This feature of the Host Controller is called Auto CMD12. The Host Driver should set **Auto CMD12 Enable** in the *Transfer Mode* register when issuing a multiple block transfer command. Auto CMD12 timing synchronization with the last data block shall be done by hardware in the Host Controller. Commands that do not use the DAT line can be issued during multiple block transfers. These commands are referred to using the notation CMD_wo_DAT.

In order to prevent DAT line commands and CMD_wo_DAT commands from conflicting, the Host Controller shall arbitrate the timing by which each command is issued on the SD Bus. Therefore, a command might not immediately be issued after the Host Driver writes to the *Command* register. The command may be issued before or after Auto CMD12, depending on the timing. To be able to distinguish the responses of DAT line and CMD_wo_DAT commands, the Auto CMD12 response can be determined from the upper four bytes of the *Response* register (at offset 01Ch in the standard register set).

If errors are detected related to Auto CMD12, the Host Controller shall issue an **Auto CMD Error** interrupt. The Host Driver can check the Auto CMD12 error status (Command Index/End bit/CRC/Timeout Error) by reading the *Auto CMD Error Status* register.

The Table 1-9 illustrates the relationship between Auto CMD12 errors and any CMD_wo_DAT commands that have been issued by the Host Driver.

Relation of the commands	Error Status	Comments
Auto CMD12 only	CMD_wo_DAT : Unrelated Auto CMD12 : Error	Only Auto CMD12 is issued, therefore Auto CMD12 is failed.
CMD_wo_DAT before Auto CMD12	CMD_wo_DAT : No Error Auto CMD12 : Error	CMD_wo_DAT successful, but Auto CMD12 failed.
CMD_wo_DAT before Auto CMD12	CMD_wo_DAT : Error Auto CMD12 : Not executed	CMD_wo_DAT is failed, therefore Auto CMD12 could not be issued.
Auto CMD12 before CMD_wo_DAT	CMD_wo_DAT : Not executed Auto CMD12 : Error	Auto CMD12 is failed, therefore CMD_wo_DAT could not be issued.

Table 1-9 : Relation between Auto CMD12 and CMD_wo_DAT

The Host Driver may determine which of these error cases has occurred by checking the *Auto CMD Error Status* register when an **Auto CMD Error** interrupt occurs. If the Auto CMD12 was not executed, the Host Driver needs to recover from the CMD_wo_DAT error and issue CMD12 to stop the multiple block transfer. If the CMD_wo_DAT was not executed, the Host Driver can issue it again after recovering from the Auto CMD12 error. The procedures for recovering from error interrupts and from Auto CMD12 errors are described in sections 3.10.1 and 3.10.2.

In UHS mode SDR104 (Refer to Section 2.2.25 *Host Control 2* register), Host Driver shall use Auto CMD23 to stop multiple block read / write operation instead of using Auto CMD12. In the other bus speed mode, if the card supports CMD23, Host Driver should use Auto CMD23 instead of using CMD12.

In UHS-II mode, Host Driver should use TLEN to stop multiple block read / write operation instead of using CMD12.

1.12 Controlling SDCLK

Table 1-10 shows how **SDCLK** is controlled by the **SD Bus Power** in the *Power Control* register and the **SD Clock Enable** in the *Clock Control* register.

The Clock Period of ***SDCLK*** is specified by the **SDCLK/RCLK Frequency Select** in the *Clock Control* register and the **Base Clock Frequency For SD Clock** in the *Capabilities* register. Because of the SD card may use both clock edges, the duty of SD clock should be average 50% (scattering within 45-55%) and the Period of High should be half of the Clock Period. The oscillation of ***SDCLK*** starts from driving specified Period of High. When ***SDCLK*** is stopped by the **SD Clock Enable**, the Host Controller shall stop ***SDCLK*** after driving Period of High to maintain clock duty. When ***SDCLK*** is stopped by the **SD Bus Power**, the Host Controller shall stop ***SDCLK*** immediately (drive Low) and **SD Clock Enable** should be cleared.

SD Bus Power (Note 1)	**SD Clock Enable** (Note 2)	**State of *SDCLK***
Change 0 to 1	0	Drive Low
	1	Start Clock with specified Period of High
Change 1 to 0	0	Drive Low
	1	Drive Low immediately
0	Don't Care	Drive Low
1	Change 0 to 1	Start Clock with specified Period of High
	Change 1 to 0	Maintains Period of High and then stops Clock and drive Low

Table 1-10 : Controlling SDCLK by the SD Bus Power and SD Clock Enable

Note 1: When the card state is changed from Debouncing to No Card, the Host Controller shall clear the **SD Bus Power**.
Note 2: When the the card state is changed from Card Inserted to Debouncing, the Host Controller shall clear the **SD Clock Enable** immediately.

If Host Controller supports shared bus in SD mode, each device is selected by clock output pins and specific clock control is required. Refer to *Embedded Control* register for more detail.

1.13 Advanced DMA

There are three types of DMAs: SDMA, ADMA2 and ADMA3. SDMA (Single Operation DMA) performs a read / write SD command operation. SDMA is suitable for short data transfer because SDMA requires address update at page boundary of system memory. **DMA Interrupt** generated at every page boundary disturbs CPU to reprogram the new system address. A long data transfer should use ADMA to avoid performance bottleneck by interruption at every page boundary. ADMA2 and ADMA3 adopt scatter gather DMA algorithm so that higher data transfer speed is available. The Host Driver can program a list of data transfers between system memory and SD card to the Descriptor Table. ADMA2 performs a read / write SD command operation at a time. ADMA3 can program multiple read / write SD commands operation in a Descriptor Table. ADMA3 is suitable to perform very large data transfer.

Support of 64-bit addressing is modified from Version 4.00. ADMA2 and ADMA3 64-bit Descriptor length is modified to 128 bits considering byte enable alignment. 32-bit or 64-bit addressing mode is selected at initialization that is determined by OS (Up to Version 3.00, 64-bit mode may be selected by Host Driver at each operation by **DMA Select** in *Host Control 1* register).

SDMA is also extended in Version 4.00 by supporting not only 64-bit addressing but also 32-bit block count in UHS-II mode.

1.13.1 ADMA Data Transfer between Host Controller and System Memory

1.13.1.1 Block Diagram of ADMA2

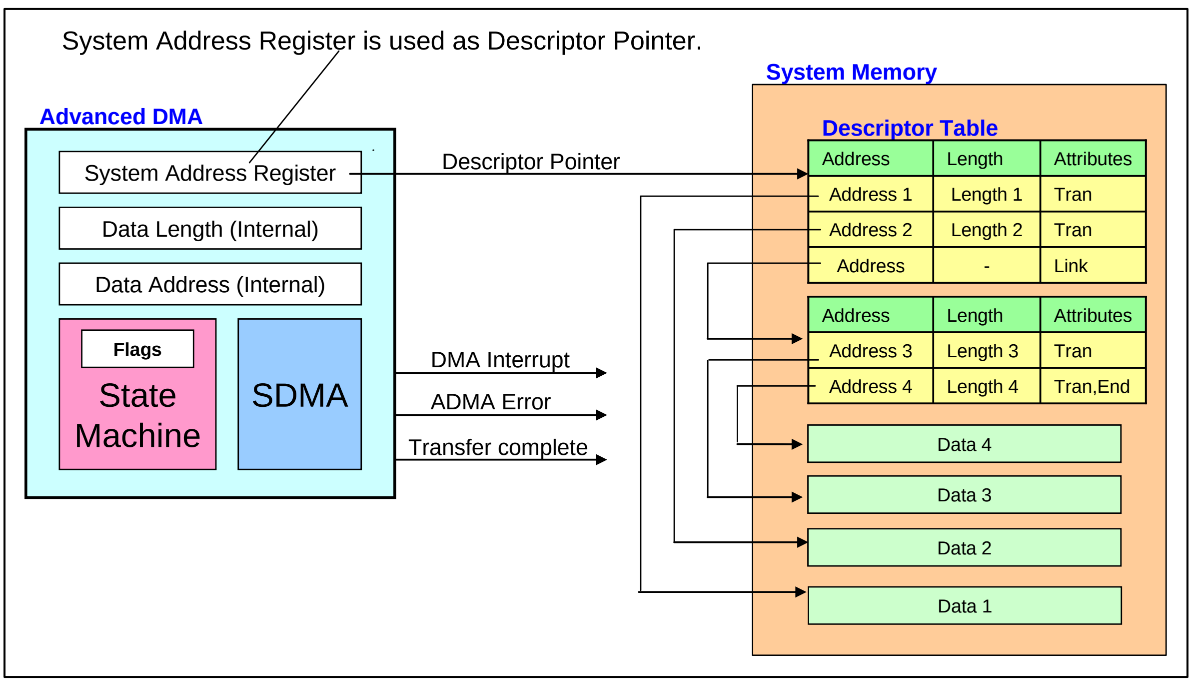


Figure 1-8 : Block Diagram of ADMA2

SDMA and ADMA2 handles data transfer between Host Controller and System Memory. Figure 1-8 shows block diagram of ADMA2. The Descriptor Table is created in system memory by the Host Driver. 32-bit Address Descriptor Table is used for the system with 32-bit addressing and 64-bit Address Descriptor Table is used for the system with 64-bit addressing. Each descriptor line (one executable unit) consists with address, length and attribute field. The attribute specifies operation of the descriptor line. ADMA2 includes SDMA, State Machine and registers circuits. ADMA2 does not use 32-bit *SDMA*

System Address register (offset 0) but uses the 64-bit *Advanced DMA System Address* register (offset 058h) for descriptor pointer. Writing Command register triggers off ADMA2 transfer. ADMA2 fetches one descriptor line and execute it. This procedure is repeated until end of descriptor is found (End=1 in attribute).

1.13.1.2 An Example of ADMA2 Programming

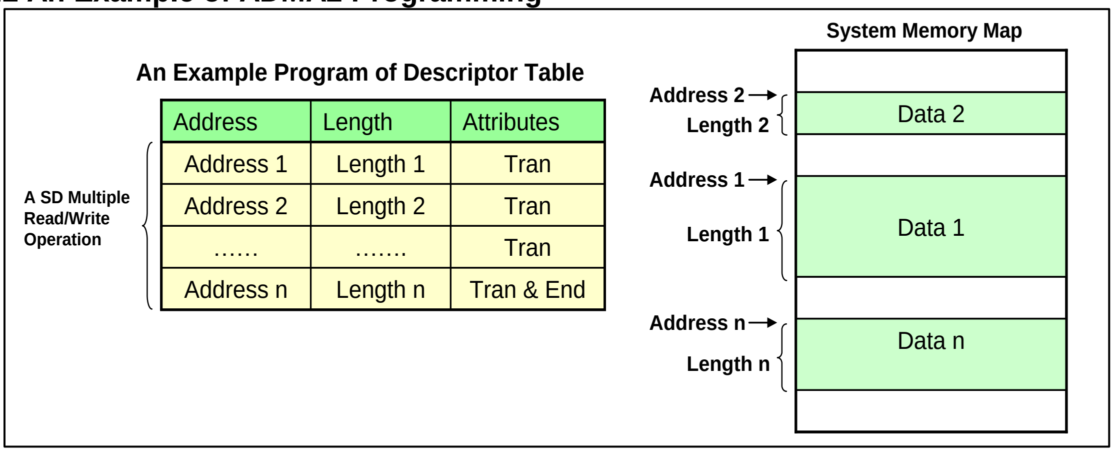


Figure 1-9 : An Example of ADMA2 Data Transfer

Figure 1-9 shows a typical ADMA2 descriptor program. The Host Driver describes the Descriptor Table with each slice is placed somewhere in contiguous system memory. The Host Driver describes the Descriptor Table with set of address, length and attributes. Each sliced data is transferred in turns as programmed in descriptor.

1.13.1.3 Data Address and Data Length Requirements

There are three requirements to program the descriptor.

(1) The minimum unit of address is 4 bytes.
(2) The maximum data length of each descriptor line is less than 64KB.
(3) Total Length = Length 1 + Length 2 + Length 3 + ... + Length n
                 = multiple of Block Size

4 bytes unit of address simplifies Byte Enable control on 32-bit (64-bit) data bus and it would be sufficient length to manage a minimum data unit on system memory for most Operating Systems regardless of 32-bit/64-bit addressing.
If total length of a descriptor were not multiple of block size, ADMA2 transfer might not be terminated. In this case, data timeout would occur and the transfer would be stopped by abort command. Therefore, total length should be multiple of block size.

1.13.2 General Descriptor Table Format

Figure 1-10 shows general format of Descriptor Table. One descriptor line consumes 64-bit (8-byte) for 32-bit addressing mode and 128-bit (16-byte) for 64-bit addressing mode. Attribute is used to control descriptor. Act0 is defined from Version 4.10 to extend descriptors for ADMA3. Act0=1 is assigned to ADMA3 Descriptor. Refer to Section 1.13.4 about ADMA3 Descriptor. Act0=0 is assigned to ADMA2 Descriptor. Refer to Section 1.13.3 about ADMA2 Descriptor.

128-bit Descriptor (64-bit Addressing) General Format

Reserved	64-bit Field	16-bit Field	10-bit Field	Attribute					
127 96	95 32	31 16	15 06	05	04	03	02	01	00
0000_0000h	xxxx_xxxx_xxxx_xxxxh	xxxxh	xxxxxxxxxxb	Act2	Act1	Act0	Int	End	Valid

64-bit Descriptor (32-bit Addressing) General Format

32-bit Field	16-bit Field	10-bit Field	Attribute					
63 32	31 16	15 06	05	04	03	02	01	00
xxxx_xxxxh	xxxxh	xxxxxxxxxxb	Act2	Act1	Act0	Int	End	Valid

Valid	**Valid=1 indicates this line of descriptor is effective. If Valid=0 generate ADMA Error Interrupt and stop ADMA.**
End	**End=1 indicates terminate transfer and generates Transfer Complete Interrupt when this line of transfer is completed.**
Int	**INT=1 generates DMA Interrupt when this line of transfer is completed.**

Act2	Act1	Act0	Operation
x	x	0	ADMA2 Descriptor (A2Des)
0	0	1	Command Descriptor for SD Mode
0	1	1	Command Descriptor for UHS-II Mode
1	0	1	Reserved
1	1	1	Integrated Descriptor

Act2	Act1	Act0	Symbol	Operation
0	0	0	Nop	No Operation
0	1	0	rsv	reserved
1	0	0	Tran	Transfer Data
1	1	0	Link	Link Descriptor

Figure 1-10 : General Descriptor Table Format

There are two kinds of descriptors for 64-bit addressing mode: 96-bit Descriptor and 128-bit Descriptor. Table 1-11 shows the 96-bit Descriptor format. 128-bit Descriptor is defined from Version 4.00. Host Controller Version 4.00 or later should support 128-bit Descriptor and support of 96-bit Descriptor is optional. **Host Version 4 Enable** in the *Host Control 2* register is used to select either of descriptors. Setting 0 selects 96-bit Descriptor and setting 1 selects 128-bit Descriptor.

Address Field	Length	Reserved	Attribute					
95 32	31 16	15 06	05	04	03	02	01	00
64-bit Address	16-bit Length	000000	Act2	Act1	0	Int	End	Valid

Table 1-11 : 64-bit Address Descriptor Table

Address registers are defined as 64-bit to support 64-bit addressing. 32-bit address is stored in the lower 32-bit of 64-bit address register. 64-bit and 96-bit Descriptor shall be aligned to 4-byte address boundary (Lower 2-bit of system address is always 0) and 128-bit Descriptor shall be aligned to 8-byte address boundary (Lower 3-bit of system address is always 0).

1.13.3 ADMA2

1.13.3.1 ADMA2 Descriptor Format

Figure shows ADMA2 Descriptor Table. Act0=0 is assigned to ADMA2 descriptor. Three action symbols are specified by combination of Act2 and Act1. "Nop" operation skips current descriptor line and fetches next one. "Tran" operation transfers data designated by address and length field. "Link" operation is used to connect separated two descriptors. The address field of link points to next Descriptor Table. The combination of Act2=0 and Act1=1 is reserved and defined the same operation as Nop. A future version of controller may use this field and redefine a new operation.

128-bit ADMA2 Descriptor Line (64-bit Addressing Mode)

Reserved	64-bit Address	16-bit Length	10-bit Length	Attribute					
127 96	95 32	31 16	15 06	05	04	03	02	01	00
0000_0000h	xxxx_xxxx_xxxx_xxxxh	xxxxh	xxxxxxxxxxb	Act2	Act1	Act0	Int	End	Valid

64-bit ADMA2 Descriptor Line (32-bit Addressing Mode)

32-bit Address	16-bit Length	10-bit Length	Attribute					
63 32	31 16	15 06	05	04	03	02	01	00
xxxx_xxxxh	xxxxh	xxxxxxxxxxb	Act2	Act1	Act0	Int	End	Valid
	Lower	Upper						

Data length is extended from Ver.4.10.
(1) 16-bit Data Length mode
(2) 26-bit Data Length mode

Valid	Indicates Validity of a Descriptor Line
End	End of Descriptor
Int	Force to generate ADMA Interrupt

Act2	Act1	Act0	Symbol	Comment	Operation
0	0	0	Nop	No Operation	Do not execute current line and go to next line.
0	1	0	rsv	reserved	(Same as Nop. Do not execute current line and go to next line.)
1	0	0	Tran	Transfer Data	Transfer data of one descriptor line
1	1	0	Link	Link Descriptor	Link to another descriptor

Figure 1-11 : ADMA2 Descriptor Table

"Int" in Attribute may be set in only ADMA2 Descriptor.

From Version 4.10, 26-bit Data Length mode is added to reduce the number of descriptor lines for large continuous data. Support of 26-bit Data Length is mandatory for Host Controller Version 4.10 or later. Table 1-12 shows the definition of 16-bit Data Length and Table 1-13 shows the definition of 26-bit Data Length. 16-bit Data Length is selected when **ADMA2 Length Mode** in the *Host Control 2* register is set to 0. 26-bit Data Length is selected when **ADMA2 Length Mode** is set to 1.

16-bit Length (D31-D16)	Value of Length
0000h	65536 bytes
0001h	1 byte
0002h	2 bytes
....	....
FFFFh	65535 bytes

Table 1-12 : ADMA2 16-bit Length Mode

26-bit Length (D15-D06, D31-D16)	Value of Length
000_0000h	64M bytes
000_0001h	1 byte
000_0002h	2 bytes
000_0003h	3 bytes
....	....
3FF_FFFFh	64M-1 bytes

Table 1-13 : ADMA2 26-bit Length Mode

1.13.3.2 ADMA2 States

Figure 1-12 shows state diagram of ADMA2. 4 states are defined; Fetch Descriptor state, Change Address state, Transfer Data state and Stop ADMA2 state. Operation of each state is explained in Table 1-14.

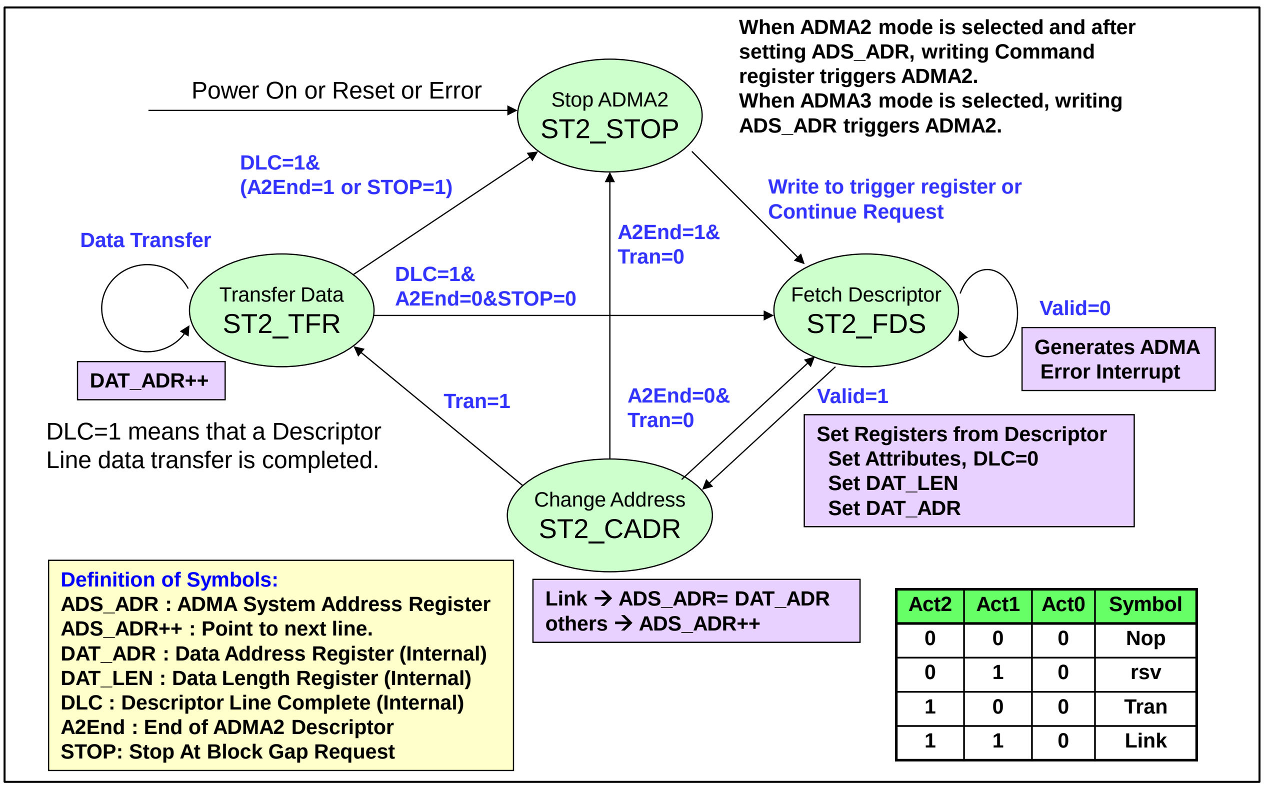


Figure 1-12 : State Diagram of the ADMA2

State Name	Operation
ST2_FDS (Fetch Descriptor)	ADMA2 fetches a descriptor line and set parameters in internal registers. Next go to ST2_CADR state.
ST2_CADR (Change Address)	Link operation loads another Descriptor address to *ADMA System Address* register. In other operations, *ADMA System Address* register is incremented to point next descriptor line. If End=0, go to ST2_TFR state. This is temporal state and immediately moves to next state.
ST2_TFR (Transfer Data)	Data transfer of one descriptor line is executed between system memory and Host Controller. If data transfer continues (End=0) go to ST2_FDS state. If data transfer completes, go to ST2_STOP state.
ST2_STOP (Stop ADMA2)	Start of ADMA2 After Power on, ADMA2 starts from this state. (1) If ADMA2 is selected, writing to Command register by Host Driver triggers start of ADMA2. (2) If ADMA3 is selected, writing to the ADMA System Address register by ADMA3 triggers start of ADMA2. Stop of ADMA2 (1) If ADMA2 is selected, any transition to ST2_STOP generates **Transfer Complete**. (2) If ADMA3 is selected, the timing of **Transfer Complete** depends on ADMA3 implementation (refer to Section 1.13.4.1).

Table 1-14 : ADMA2 States

Implementation Note:
ADMA2 may be initialized when it is triggered by writing the ADMA System Address register.

1.13.3.3 Stop/Continue Function during ADMA2

"Stop/Continue" is a function to halt data transfer on the way at the block gap of SD bus and to restart data transfer. The **Stop At Block Gap Request** in the *Block Gap Control* register is used to halt ADMA2 data transfer and **Continue Request** in the *Block Gap Control* register is used to restart ADMA2 data transfer. While stopping ADMA2, any SD command cannot be issued if intending to continue ADMA2 operation (An abort command may be issued and ADMA2 shall be aborted accordingly).
The Host Controller stops read operation on SD bus by using Read Wait or stopping SD Clock (In case of Host Controller Version 1.00, the **Stop At Block Gap Request** can be used with the Read Wait).
Host Controller generates the **Transfer Complete** interrupt when data transfer halts and sets the **Block Gap Event** when data transfer is not completed yet together with the **Transfer Complete**. Setting the **Continue Request** restarts data transfer (**Block Gap Event**=0 means ADMA2 data transfer is completed and continue request is not required).
Section 3.12.3 and Section 3.12.4 define Stop/Continue timing for non-DMA. The timing of the **Transfer Complete** and the **Block Gap Event** may be different in case of DMA because interrupt timing depends on the relation between data transfer on SD bus and ADMA2 data transfer on system bus (system memory). There may be a difference in data length transferred on SD bus and system bus. In this case, buffer in the Host Controller holds untreated data.

Behavior of Stop/Continue function for ADMA2 has been defined with ADMA2 states in Figure 1-12. Host Controller Version 4.20 provides not only clarification of Figure 1-12 but also another implementation of Stop/Continue function with flexibility.

Figure 1-12 defines behavior of Stop/Continue function with state transition:

(a) On receiving Stop At Block Gap Request, ADMA2 halts data transfer by transition from ST2_TFR to in ST2_STOP after execution of a descriptor line is completed. Data transfer may halt at any block gap and at any descriptor line where ADMA2 can easily stop. This means that ADMA2 may control the timing of setting STOP symbol to 1 after the Stop At Block Gap Request is set to 1.

(b) On receiving Continue Request, ADMA2 restarts data transfer by transiting ST2_STOP to ST2_FDS.

Another simplified implementation is allowed instead of using STOP condition in Figure 1-12, that is, Stop/Continue function may be controlled in ST2_TFR without state transition. This means that ADMA2 may halt and continue during middle of a descriptor line.

1.13.3.4 ADMA Error Status Register

Error occurrence during ADMA2 transfer may stop ADMA2 operation and generate an **ADMA Error Interrupt**. The **ADMA Error State** field in the *ADMA Error Status* register holds state of ADMA2 stopped. The Host Driver can identify the error descriptor location by the following method: If ADMA stopped at ST_FDS state, the *ADMA System Address Register* points the error descriptor line. If ADMA stopped at ST_TFR or ST_STOP state, the *ADMA System Address* register points the next location of error descriptor line. By this reason, ADMA2 shall not stop at ST_CADR state.

1.13.4 ADMA3

ADMA3 enables host to program multiple of ADMA2 operations. In case of ADMA2, SD Command issuing is controlled by Host Driver by writing to Host Controller registers. ADMA3 uses Command Descriptor to issue an SD command. A multi-block data transfer between system memory and SD Card is programmed by using a pair of Command Descriptor and ADMA2 Descriptor. ADMA3 performs multiple of multi-block data transfer by using Integrated Descriptor. ADMA3 is optional and support of ADMA3 is indicated by **ADMA3 Support** in the *capabilities* register.

Figure 1-13 shows an example ADMA3 operation that three data blocks (Data A, Data B and Data C) are written to different area of SD Memory Card. Integrated Descriptor consists of pointers to Command Descriptors. Each of Command Descriptor is followed by ADMA2 Descriptor. The first descriptor pair is programed to transfer Data A. The second pair is to transfer Data B and the third pair is to transfer Data C.

Location of Integrated Descriptor is set to *ADMA3 Integrated Descriptor Address* register. ADMA3 fetches pointers one by one in the Integrated Descriptor and executes Descriptors designated by the pointer. ADMA3 sets contents of Command Descriptor to the Host Controller registers to issue an SD command and then executes ADMA2 Descriptor. The first operation transfers Data A from system memory to SD Memory Card. The second operation transfers Data B and the third operation transfers Data C. When execution of all descriptors pointed by the Integrated Descriptor is completed, ADMA3 generates Transfer Complete interrupt to inform Host Driver completion of ADMA3 operation.

There are two types of Command Descriptor: SD Command type and UHS-II Command type. SD Command type is set to register offsets 000h to 00Fh and UHS-II Command type is set to register offsets 080h to 09Fh.

Stop/Continue function is also effective to ADMA3 (Refer to Section 1.13.3.3 for ADMA2). The **Stop At Block Gap Request** is used to halt ADMA3 data transfer and the **Continue Request** is used to restart ADMA3 data transfer when **Block Gap Event** is set together with **Transfer Complete**.

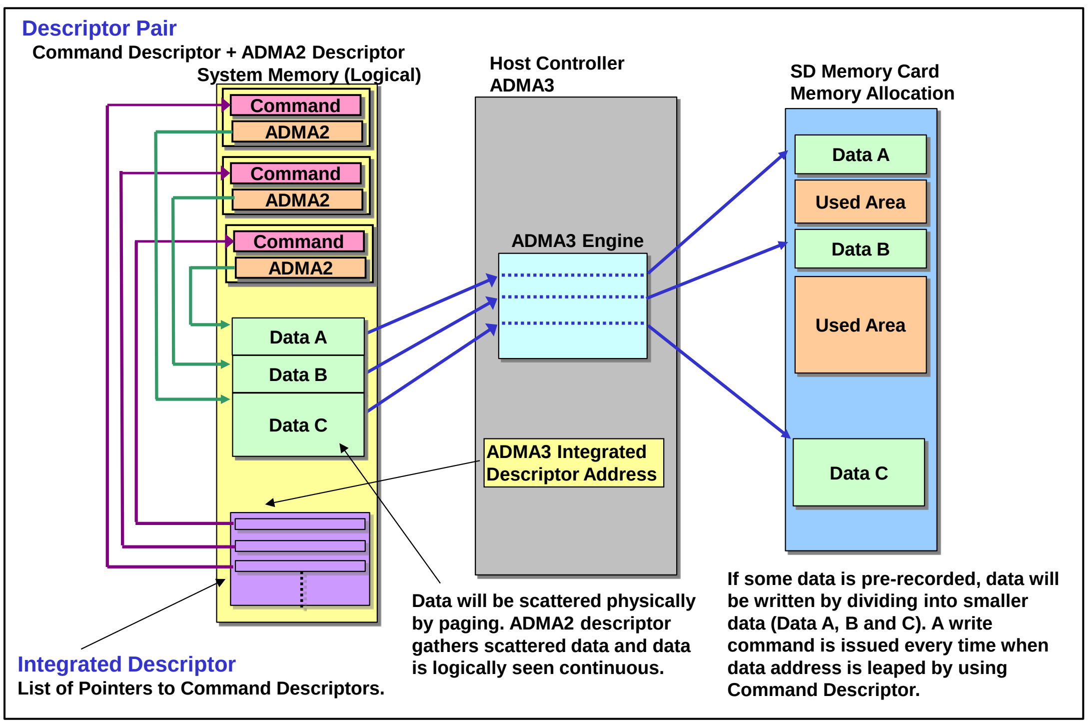


Figure 1-13 : Example of ADMA3 Operation

1.13.4.1 ADMA3 States

Figure 1-14 shows state diagram of ADMA3. Four states are defined; Fetch Descriptor state, Set Register state, Execute ADMA2 state and Stop ADMA3 state. Operation of each state is explained in Table 1-15.

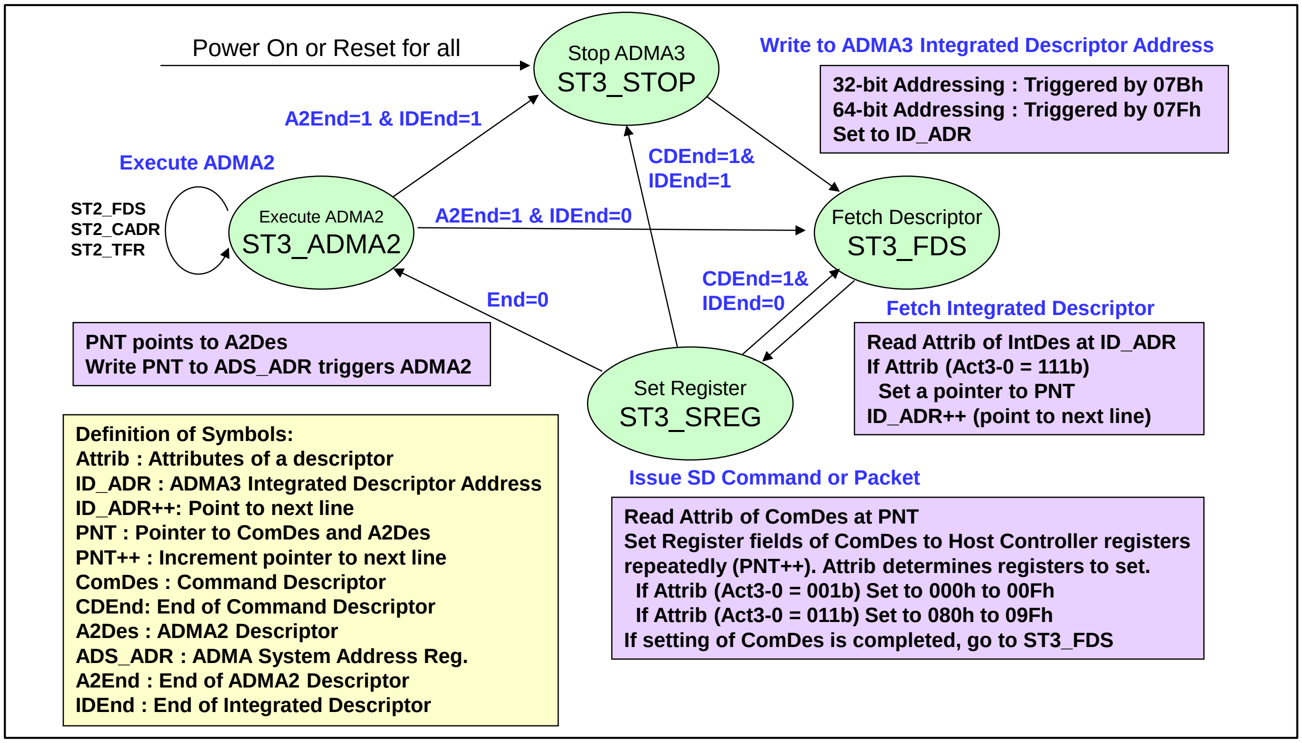


Figure 1-14 : State Diagram of the ADMA3

State Name	Operation
ST3_FDS (Fetch Descriptor)	ADMA3 fetches a pointer from Integrated Descriptor. The pointer is set to PNT. If the last pointer is fetched, IDEnd=1. Next go to ST3_SREG state.
ST3_SREG (Change Address)	ADMA3 fetches Command Descriptor designated by PNT. In case of SD Command type, contents of the descriptor are set to register offsets 000h to 00Fh. In case of UHS-II Command type, contents of the descriptor are set to register offsets 080h to 09Fh. PNT is incremented every time a descriptor line is read. When CDEnd is set to 1 in attribute, it enables to issue an SD command without data transfer (ex. insert CMD13 check). From Host Controller Version 4.20, ADMA3 is improved so that SD command without data transfer can be the last operation of ADMA3. (a) From Host Controller Version 4.20 If CDEnd is set to 1 in attribute, go to ST3_FDS when IDEnd=0 and go to ST3_STOP if IDEnd=1. (b) Prior to Host Controller Version 4.20 The last Command Descriptor shall be SD command with data transfer.
ST3_ADMA2 (Execute ADMA2)	Assuming PNT points to top of ADMA2 Descriptor and PNT is set to the *ADMA System Address* register and it starts execution of ADMA2. When ADMA2 execution completes (A2End=1), go to ST3_FDS if next pointer is there in the Integrated Descriptor (IDEnd=0). If the last pointer was fetched at ST3_FDS (IDEnd=1), go to ST3_STOP state.

ST3_STOP (Stop ADMA3)	Start of ADMA3 After Power on, ADMA3 starts from this state. Write to the ADMA3 Integrated Descriptor Address register triggers start of ADMA3. Stop of ADMA3 Any transition to ST3_STOP generates **Transfer Complete**. **Command Complete** is disabled by setting **Response Interrupt Disable** in the *(UHS-II) Transfer Mode* register during ADMA3. From Host Controller Version 4.20, conditions to ST3_STOP is modified by adding the transition from ST3_SREG to ST3_STOP with the condition of IDEnd=1 & CDEnd=1.

Table 1-15 : ADMA3 States

Implementation Note:
ADMA3 may be initialized when it is triggered by writing the ADMA3 Integrated Descriptor Address register.

1.13.4.2 Command Descriptor Format

Figure 1-15 shows Command Descriptor Format. 32-bit register data is set in bit 63-32 of each descriptor line. Command Descriptor types (SD Mode or UHS-II Mode) are distinguished by Attribute.
If Attribute indicates Command Descriptor for SD Mode (Act2-0=001b), 32-bit Register fields are written to Host Controller Registers from 000h to 00Fh. When 00Fh is written, an SD Command is issued. If Attribute indicates Command Descriptor for UHS-II Mode (Act2-0=011b), 32-bit Register fields are written to Host Controller Registers from 080h to 09Fh. When 09Fh is written, a UHS-II Command Packet is issued. Host Controller has a pointer to a descriptor line for Command Descriptor and ADMA2 Descriptor. The pointer is incremented after reading of every descriptor line. When the last line of Command Descriptor is read, the pointer is assumed to point top of ADMA2 Descriptor, which is placed just after Command Descriptor. Host Controller ignores "Int" of Attribute in this descriptor.

Command Descriptor (for 32-bit/64-bit Addressing Mode)

32-bit Register	Reserved	Reserved	Attribute					
63 32	31 16	15 06	05	04	03	02	01	00
xxxx_xxxxh	0000h	000000b	Act2	Act1	Act0	Int	End	Valid

(1) Command Descriptor for SD Mode

32-bit Register Fields are written to Host Controller Registers from 000h to 00Fh to issue a SD Command.

32-bit Register	Reserved	Attribute	
32-bit Block Count	all 0	001001b	→ Set to 003h-000h
(16-bit Block Count) + Block Size	all 0	001001b	→ Set to 007h-004h
Argument	all 0	001001b	→ Set to 00Bh-008h
Command + Transfer Mode	all 0	001011b	→ Set to 00Fh-00Ch

(2) Command Descriptor for UHS-II Mode

32-bit Register Fields are written to Host Controller Registers from 080h to 09Fh to issue a UHS-II Command Packet.

32-bit Register	Reserved	Attribute	
0000h + UHS-II Block Size	all 0	011001b	→ Set to 083h-080h
UHS-II Block Count	all 0	011001b	→ Set to 087h-084h
UHS-II Command Packet (5 lines)	all 0	011001b	→ Set to 09Bh-088h
UHS-II Command + UHS-II Transfer Mode	all 0	011011b	→ Set to 09Fh-09Ch

Figure 1-15 : Command Descriptor Format

Followings are programing requirements of Command Descriptor for SD Mode:

(1) Setting infinite data transfer (Block Count Enable=0) is not allowed, except for single block transfer (Multi / Single Block Select=0).

(2) As ADMA3 only supports 32-bit Block Count mode, Block Count Enable shall be set to 1, 16-bit Block Count shall be set to 0000h. Setting Stop Count (32-bit Block Count =0) is not allowed, except for single block transfer (Multi / Single Block Select=0).

(3) Setting no data transfer (Block Size=0) is not allowed, except for no data transfer command in Command Descriptor (CDEnd=1).

(4) In case of memory data transfer command, “Auto CMD Auto Select” of **Auto CMD Enable** in the *Transfer Mode* register shall be used so that Command Descriptor is independent to whether card support CMD23. For another command (including SDIO), “Auto Command Disabled” of **Auto CMD Enable** shall be used.

1.13.4.3 Integrated Descriptor Format

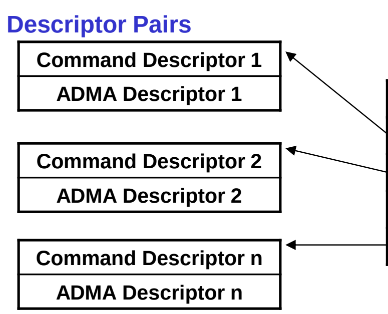
Figure 1-16 shows Integrated Descriptor Format. Multiple of pointers to Command Descriptors are set in the Integrated Descriptor. The pointer of 64-bit address is set to bit 95-32 in the 128-bit Integrated Descriptor. The pointer of 32-bit address is set to bit 63-32 in the 64-bit Integrated Descriptor. Host Controller ignores "Int" of Attribute in this descriptor.

128-bit Integrated Descriptor (64-bit Addressing)

Reserved	64-bit Pointer	Reserved	Reserved	Attribute					
127 96	95 32	31 16	15 06	05	04	03	02	01	00
0000_0000h	xxxx_xxxx_xxxx_xxxxh	0000h	000000b	Act2	Act1	Act0	Int	End	Valid

64-bit Integrated Descriptor (32-bit Addressing)

32-bit Pointer	Reserved	Reserved	Attribute					
63 32	31 16	15 06	05	04	03	02	01	00
xxxx_xxxxh	0000h	000000b	Act2	Act1	Act0	Int	End	Valid



Integrated Descriptor

Pointer Field	Reserved	Attribute
Pointer to Command Descriptor 1	all 0	111001b
Pointer to Command Descriptor 2	all 0	111001b
....	......	......
Pointer to Command Descriptor n	all 0	111011b

End=1 indicates the last line.

Figure 1-16 : Integrated Descriptor Format

ADMA3 Integrated Descriptor Address register (offset 07Fh - 078h) is used to designate the location of the Integrated Descriptor. Start of ADMA3 is triggered by writing to offset 07Bh in 32-bit addressing mode and writing to offset 07Fh in 64-bit addressing mode.

1.13.4.4 Response Error Check During ADMA3

During ADMA3 operation, Host Controller should perform response check to prevent performance loss of checking by Host Driver. When creating Command Descriptor, following 3 bits in the *(UHS-II) Transfer Mode* register are set as follows:

Response Interrupt Disable = 1
Response Error Check Enable = 1
Response Type R1 / R5 =0 if command is for memory or =1 if command is for SDIO.

In SD mode, if an error is detected in R1/R5, **Response Error Interrupt** is generated in the *Error Interrupt Status* register. ADMA3 is stopped and Host Driver can read the error response from the *Response* register.
In UHS-II mode, **RES Packet Error** Interrupt is generated in the *UHS-II Error Interrupt Status* register. ADMA3 is stopped and Host Driver can read the error response from *UHS-II Response* register.

1.14 Test Registers

The test registers are defined for testing purpose. When it is difficult to generate some interrupts intentionally, this feature can be used to generate these interrupts manually for driver debugging. The *Force Event* register to control the *Error Interrupt Status* and *Auto CMD Error Status* are defined for this purpose. Intentional control of card insertion and removal are also difficult. The **Card Detect Signal Selection** and **Card Detect Test Level** in the *Host Control* 1 register enable manual control of **Card Inserted** in the *Present State* register and generating interrupt of **Card Insertion** and **Card Removal** in the *Normal Interrupt Status* register. Support of the test registers is mandatory.
For UHS-II mode, *Force Event for UHS-II Error Interrupt Status* register is defined.

1.15 Block Count

Block Count is the parameter for SD Bus data transfer to determine total data length by multiplying Block Length. Data transfer length sets to SD Card shall be equivalent to data transfer length sets to ADMA2 and ADMA3 Descriptor.

1.15.1 Selection of 16-bit or 32-bit Block Count

According to increase of memory card capacity, larger data would be transferred to/from SD Memory Card. By this reason, Host Controller Version 4.10 extends block count from 16-bit to 32-bit for all operations in SD mode; CPU transfer, SDMA and ADMA. (In prior version, long data transfer can be supported by using Auto CMD23 or in UHS-II mode.) As SDMA may use *ADMA System Address* register to support 64-bit addressing, *SDMA System Address* register is re-assigned to *32-bit Block Count register.* Selection of block count registers either *16-bit Block Count* (offset 007h-006h) or *32-bit Block Count* (offset 003h-000h) is defined as follows, which allows mixed use of 16-bit Block Count or 32-bit Block Count.

(1) If **Host Version 4 Enable** in the *Host Control 2* register is set to 0 or *16-bit Block Count* register is set to non-zero, *16-bit Block Count* register is selected.
(2) If **Host Version 4 Enable** is set to 1 and *16-bit Block Count* register is set to 0000h, *32-bit Block Count* register is selected.

Use of block count is enabled by setting **Block Count Enable** in the *Transfer Mode* register.
Support of 32-bit block count is mandatory in Version 4.10 and Host Driver Version 4.10 shall use 32-bit block count (ADMA3 supports only 32-bit block count).
32-bit *UHS-II Block Count* register is always used in UHS-II mode.

1.15.2 Block Count for Auto CMD23

Set Block Count Command (CMD23) is defined by the Physical Layer Specification Version 3.00 for SD mode as optional. It provides timing free method to stop a multiple block operation. A block count is set in the argument of CMD23 to specify a transfer length of following CMD18 or CMD25.
Auto CMD23 is a feature that automatically issues a CMD23 before a CMD18 or CMD25 is sent. Objective of this function is to avoid performance deterioration during memory access by removing the interrupt service of CMD23. Offset 008h *Argument* register is used for CMD18 or CMD25 and offset 000h is assigned for *32-bit Block Count* register for CMD23.
In UHS-II mode, data length is set to TLEN instead of using CMD23.

1.15.3 Restriction of 16-bit Block Count

When *16-bit Block Count* register is used, Host Controller requires special management.
Old Host Driver might set **Block Count Enable** to 1. In this case, *16-bit Block Count* register limits the maximum of 65535 blocks transfer for ADMA
ADMA enables longer data transfer by disabling **Block Count Enable** as described in Table 1-16.

Instead of using Block Count register, total data length can be determined by sum of data length for each line in ADMA Descriptor. However, as SD Bus and ADMA are in different timing domain, Host Controller needs to control total data length of ADMA Descriptor on SD Bus timing domain instead of block count (Special management may be required to control **Read Transfer Active**, **Write Transfer Active** and **DAT line Active** in the *Present State* register). In read operation, several blocks may be read more than required on SD Bus domain.
Host Driver Version 4.00 needs to control **Block Count Enable** as described in Table 1-16.
Host Driver Version 4.10 may keep **Block Count Enable** to 1 by using 32-bit block count mode. ADMA3 is used in 32-bit block count mode.

Transfer Mode	Block Count Enable	Data Length
SDMA	1	Block Count Register Value
ADMA2	0	Total length of ADMA Descriptor

Table 1-16 : Host Controller Data Transfer Length

1.16 Sampling Clock Tuning

In UHS-I mode, the SD bus can be operating in high clock frequency mode and then the data window from the card on CMD and DAT[3:0] lines gets smaller. The position of the data window will vary depending on the card and host system implementation. Therefore, the Host Controller shall support a tuning circuit when SDR104 or SDR50 (if **Use Tuning for SDR50** is set to 1 in the *Capabilities* register) is supported by executing the tuning procedure defined by Figure 2-29, and adjusting the sampling clock. **Execute Tuning** and **Sampling Clock Select** in the *Host Control 2* register are used to control the tuning circuit.

1.17 Command Issuing During Data Transfer

SDIO Specification allows for using CMD52 during CMD53 operation. These two commands can operate independently. To make the function more general, CMD53 type command is called “main command” and CMD52 type command is called “sub command”. The sub command does not have data block or does not indicate busy, only communicates command - response sequence (expression of "CMD_wo_DAT" is used in Section 0 and Section 3.10.2).

1.17.1 Response Error Check Function

Prior to Version 4.10, when response error check function is not supported or disabled (**Response Error Check Enable** is set to 0), Host Driver may be able to distinguish which command generates response errors by using **Command Complete** interrupt. However, when response error check is used, Command Complete is not generated and then it will be difficult to distinguish main or sub command. Furthermore, if service of response error interrupt were delayed, Host Driver would issue a next command and then response error statuses might be over-written.

Host Controller Version 4.10 improves Response Error Check function so that main and sub command may use it. To prevent error statuses from overwriting by a next command, **Command Inhibit (CMD)** keeps 1 while any of response errors (explained later) is indicated, **Command Not Issued by Error** in the D27 of *Present State* register is set to 1 or **Command Not Issued by Auto CMD12 Error** in the D07 of *Auto CMD Error Status* register is set to 1 (Auto CMD12 is not supported in UHS-II mode).
The error statues above and **Command Inhibit (CMD)** are cleared by **Software Reset For CMD Line** in the *Software Reset* register.

1.17.2 Concept of How to Retry Command

Sub Command Flag is added in the D02 of *Command* register and *UHS-II Command* register. Host Driver manages how to use **Sub Command Flag** to distinguish main or sub command. Setting of **Sub Command Flag** is read through **Sub Command Status,** which is added in the D28 of *Preset State* register. **Sub Command Flag** is copied to **Sub Command Status** just before reading of *Present State* register.

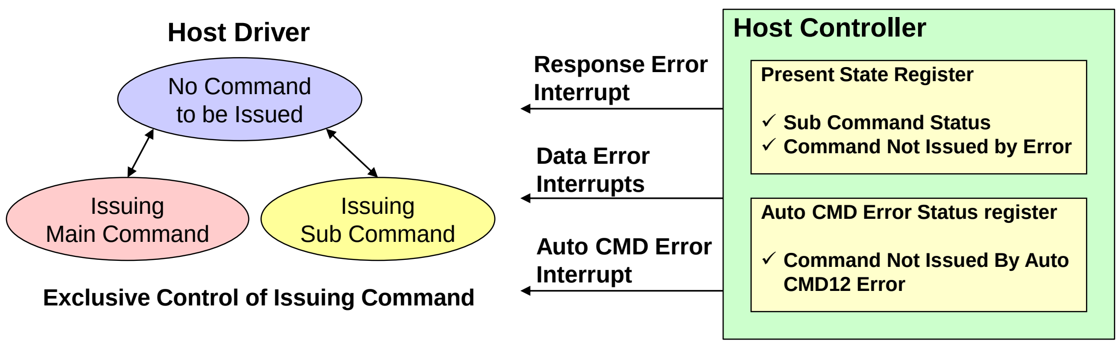


Figure 1-17 : Concept of How to Retry Command

Figure 1-17 shows a concept of how to retry command when an error occurs. This example implements two flags in Host Driver for exclusive control of issuing main or sub command. "Issuing Main Command" flag is set from Main Driver starts setting parameters to command registers and is cleared by the a write to the command register. In a similar way, "Issuing Sub Command" flag is set from Sub Driver starts setting parameters to command registers and is cleared by the a write to the command register. The flags are not set to 1 all together for exclusive control. When data transfer error interrupt is detected, an abort command will be issued to stop data transfer. In this case if the flag is set to 1, setting of command registers will be broken by issuing the abort command and then Host Driver needs to instruct Main or Sub Driver, which sets the flag, on the command retry.
Both the flags set to 0 means that Host Controller may have error statuses of command issuing. Host Controller sets **Command Not Issued by Error** or **Command Not Issued by Auto CMD12 Error** if a command is not issued after writing to the Command register or the UHS-II Command register by an error. Which command (main or sub) is not issued is determined by **Sub Command Status** in the *Present State* register.

Host Driver also needs to instruct Main or Sub Driver on the command retry in following four cases.

(1) Response Error Interrupt
The command is issued but Host Controller detects any response errors. As the command is issued, **Command Not Issued by Error** is set to 0 and Command **Not Issued by Auto CMD12 Error** is set to 0.

(2) Auto CMD Error Interrupt in SD Mode
If Command Not Issued by Auto CMD12 Error is set to 1, the command is not issued due to Auto CMD12 error.

(3) Data Transfer Error Interrupts in SD Mode
If **Command Not Issued by Error** is set to 1, the command is not issued due to errors of data transfer. The data errors are indicated as **ADMA Error**, **Tuning Error**, **Auto CMD Error**, **Data End Bit Error**, **Data CRC Error** and **Data Timeout Error** in the *Error Interrupt Status* register.

(4) Data Transfer Error Interrupts in UHS-II Mode
If **Command Not Issued by Error** is set to 1, the command is not issued due to errors of data transfer. The data errors are indicated as **Timeout for Deadlock**, **ADMA Error**, **EBSY Error**, **Unrecoverable Error, TID Error**, **Framing Error**, **CRC Error**, **Retry Expired** and **Header Error** in the *UHS-II Error Interrupt Status* register.

1.17.3 Response Error Statuses

In case of SD mode, following statuses in the *Error Interrupt Status* register are commonly used to indicate response errors of main or sub command.

D11: **Response Error**
D03: **Command Index Error**
D02: **Command End Bit Error**
D01: **Command CRC Error**
D00: **Command Timeout Error**

Moreover, Auto CMD12 might influence issuing of Sub Command. If **Command Not Issued By Auto CMD12 Error** in the *Auto CMD Error Status* register is set to 1, it means that Sub Command is not issued and Host Controller keeps **Command Inhibit (CMD)** setting to 1. In this case, **Sub Command Status** is not effective but a command not issued is a sub command.

In case of UHS-II mode, following statuses in the *UHS-II Error Interrupt Status* register are commonly used to indicate response errors of main or sub command.

D16: **Timeout for CMD_RES**
D05: **TID Error**
D04: **Framing Error**
D03: **CRC Error**
D01: **Res Packet Error**
D00: **Header Error**

Sub command can be issued during ADMA2 operation. In case of ADMA3 operation, **Stop At Block Gap Request** is used to insert sub command (Host Driver may use registers to issue a command while ADMA3 is stopping) and retrieves ADMA3 operation by the **Continue Request**.

To recover from response error, **Software Reset For CMD Line** is used to initialize command circuit (**Command Inhibit (CMD)** is set to 0) for not only SD mode but also UHS-II mode. It does not affect data transfer of main command.

1.17.4 Summary of Command Issuing During Data Transfer

If the card does not support Read Wait, there are two notes for implementation:

(1) The Host Driver cannot issue CMD_wo_DAT during read transfer because the Host Controller will stop SDCLK to wait read data transfer.

(2) On issuing Abort command, the Host Controller needs to provide SDCLK even it has been stopped to wait read data transfer and after issuing an abort command, the Host Controller should discard data stored in buffers.

Data transfer direction	Read Wait	Command can be Issued during data transfers	Notes
Read	Not Supported	Abort	Any command cannot be issued while clock has been stopped to halt read transfer (as internal buffers could overflow at a read transfer if the clock were started for issuing). By providing clock, abort command can be issued to abort read transfer. The Host Controller stops all data circuits and discards any data sent by the card.
	Supported	Abort CMD_wo_DAT	"Read Wait" control gives the host and card the ability to stop read transfer without stopping clock when internals buffers are full. It allows the Host Controller to issue commands without using DAT line.
Write	Don't care	Abort CMD_wo_DAT	Read Wait is not used for write operations. The Host Controller provides SDCLK throughout write operation and is fully in control of sending data at the appropriate time by checking busy of card.

Table 1-17 : Summary of Command Issuing During Data Transfer

2. SD Host Standard Register

2.1 Summary of register set

2.1.1 SD Host Control Register Map

Table 2-1 summarizes the standard SD Host Controller register set. The Host Driver needs to determine the base address of the register set by a Host System specific method. The register set is 256 bytes in size. For multiple slot controllers, one register set is assigned per each slot, but the registers at offsets 0F0h-0FFh are assigned as a common area. These registers contain the same values for each slot's register set.

Offset	15-08 bit	07-00 bit	Offset	15-08 bit	07-00 bit
002h	32-bit Block Count (High) SDMA System Address (High)		000h	32-bit Block Count (Low) SDMA System Address (Low)	
006h	16-bit Block Count		004h	Block Size	
00Ah	Argument (High)		008h	Argument (Low)	
00Eh	Command		00Ch	Transfer Mode	
012h	Response1		010h	Response0	
016h	Response3		014h	Response2	
01Ah	Response5		018h	Response4	
01Eh	Response7		01Ch	Response6	
022h	Buffer Data Port1		020h	Buffer Data Port0	
026h	Present State		024h	Present State	
02Ah	Wakeup Control	Block Gap Control	028h	Power Control	Host Control 1
02Eh	Software Reset	Timeout Control	02Ch	Clock Control	
032h	Error Interrupt Status		030h	Normal Interrupt Status	
036h	Error Interrupt Status Enable		034h	Normal Interrupt Status Enable	
03Ah	Error Interrupt Signal Enable		038h	Normal Interrupt Signal Enable	
03Eh	Host Control 2		03Ch	Auto CMD Error Status	
042h	Capabilities		040h	Capabilities	
046h	Capabilities		044h	Capabilities	
04Ah	Maximum Current Capabilities		048h	Maximum Current Capabilities	
04Eh	Maximum Current Capabilities (Reserved)		04Ch	Maximum Current Capabilities (Reserved)	
052h	Force Event for Error Interrupt Status		050h	Force Event for Auto CMD Error Status	
056h	---		054h	---	ADMA Error Status
05Ah	ADMA System Address [31:16]		058h	ADMA System Address [15:00]	
05Eh	ADMA System Address [63:48]		05Ch	ADMA System Address [47:32]	
062h	Preset Value		060h	Preset Value	
066h	Preset Value		064h	Preset Value	
06Ah	Preset Value		068h	Preset Value	
06Eh	Preset Value		06Ch	Preset Value	
072h	---		070h	---	
076h	---		074h	Preset Value for UHS-II	
07Ah	ADMA3 ID Address [31:16]		078h	ADMA3 ID Address [15:00]	
07Eh	ADMA3 ID Address [63:48]		07Ch	ADMA3 ID Address [47:32]	

082h	---	080h	UHS-II Block Size [15:0]
086h	UHS-II Block Count [31:16]	084h	UHS-II Block Count [15:0]
08Ah	UHS-II Command Packet (Byte 3, 2)	088h	UHS-II Command Packet (Byte 1, 0)
08Eh	UHS-II Command Packet (Byte 7, 6)	08Ch	UHS-II Command Packet (Byte 5, 4)
092h	UHS-II Command Packet (Byte 11, 10)	090h	UHS-II Command Packet (Byte 9, 8)
096h	UHS-II Command Packet (Byte 15, 14)	094h	UHS-II Command Packet (Byte 13, 12)
09Ah	UHS-II Command Packet (Byte 19, 18)	098h	UHS-II Command Packet (Byte 17, 16)
09Eh	UHS-II Command [15:00]	09Ch	UHS-II Transfer Mode [15:0]
0A2h	UHS-II Response (Byte 3, 2)	0A0h	UHS-II Response (Byte 1, 0)
0A6h	UHS-II Response (Byte 7, 6)	0A4h	UHS-II Response (Byte 5, 4)
0AAh	UHS-II Response (Byte 11, 10)	0A8h	UHS-II Response (Byte 9, 8)
0AEh	UHS-II Response (Byte 15, 14)	0ACh	UHS-II Response (Byte 13, 12)
0B2h	UHS-II Response (Byte 19, 18)	0B0h	UHS-II Response (Byte 17, 16)
0B6h	---	0B4h	--- | UHS-II MSG Select
0BAh	UHS-II MSG [31:16]	0B8h	UHS-II MSG [15:0]
0BEh	UHS-II Dev. Int. Code | UHS-II Device Select	0BCh	UHS-II Device Interrupt Status [15:0]
0C2h	UHS-II Timer Control	0C0h	UHS-II Software Reset
0C6h	UHS-II Error Interrupt Status [31:16]	0C4h	UHS-II Error Interrupt Status [15:0]
0CAh	UHS-II Error Interrupt Status Enable [31:16]	0C8h	UHS-II Error Interrupt Status Enable [15:0]
0CEh	UHS-II Error Interrupt Signal Enable [31:16]	0CCh	UHS-II Error Interrupt Signal Enable [15:0]
0D2h	Reserved	0D0h	Reserved
0D6h	Reserved	0D4h	Reserved
	---		---
0E2h	Pointer for UHS-II Host Capabilities	0E0h	Pointer for UHS-II Settings
0E6h	Pointer for Embedded Control	0E4h	Pointer for UHS-II Test
0EAh	Reserved: Pointer for Specific Control	0E8h	Pointer for Vendor Specific Area
0EEh	---	0ECh	---
0F2h	---	0F0h	---
---	---	---	---
0FEh	Host Controller Version	0FCh	Slot Interrupt Status

Table 2-1 : SD Host Controller Register Map (0FFh – 000h)

SD Host Controller Register Map (mFFh–100h, "m" is integer) is defined as re-locatable so that each register may be extended in future. This feature provides flexibility to assign registers. The start location of each register group is designated by pointers described in 0EF-0E0h. Device manufacturer may assign registers any location in mFFh-100h and Host Driver needs to support this feature to access these registers.

Table 2-2 shows an example assignment of registers in 1FFh–100h. The each register location is not restricted by this table. In Version 3.00, 1FFh–100h was assigned to Vendor Specific Area. Even if a Host Controller use this area for Vendor Specific register area (assuming it does not consume large area), still it can be maintained in Version 4.00.

Shared Bus Control register is renamed *Embedded Control* register and moved to 1FFh–100h.

Offset	15-08 bit	07-00 bit	Offset	15-08 bit	07-00 bit
102h	UHS-II Settings (General) [31:16]		100h	UHS-II Settings (General) [15:0]	
106h	UHS-II Settings (PHY) [31:16]		104h	UHS-II Settings (PHY) [15:0]	
10Ah	UHS-II Settings (LINK/TRAN) [31:16]		108h	UHS-II Settings (LINK/TRAN) [15:0]	
10Eh	UHS-II Settings (LINK/TRAN) [63:48]		10Ch	UHS-II Settings (LINK/TRAN) [47:32]	
112h	UHS-II Host Capabilities (General) [31:16]		110h	UHS-II Host Capabilities (General) [15:0]	
116H	UHS-II Host Capabilities (PHY)[31:16]		114h	UHS-II Host Capabilities (PHY) [15:0]	
11Ah	UHS-II Host Capabilities (LINK/TRAN) [31:16]		118h	UHS-II Host Capabilities (LINK/TRAN) [15:0]	
11Eh	UHS-II Host Capabilities (LINK/TRAN) [63:48]		11Ch	UHS-II Host Capabilities (LINK/TRAN) [47:32]	
122h	Force Event for UHS-II Error Int. Status[31:16]		120h	Force Event for UHS-II Error Int. Status [15:0]	
126h	Embedded Control (High)		124h	Embedded Control (Low)	
12Ah	---		128h	---	
	---			---	
1FEh	---		1FCh	---	

Table 2-2 : SD Host Controller Register Map (1FFh – 100h)

2.1.2 Configuration Register Types

Configuration register fields are assigned one of the attributes described below:

Register Attribute	Description
RO	Read-only register: Register bits are read-only and cannot be altered by software or any reset operation. Writes to these bits are ignored.
ROC	Read-only status: These bits are initialized to zero at reset. Writes to these bits are ignored.
RW	Read-Write register: Register bits are read-write and may be either set or cleared by software to the desired state.
RW1C	Read-only status, Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect.
RWAC	Read-Write, automatic clear register: The Host Driver requests a Host Controller operation by setting the bit. The Host Controllers shall clear the bit automatically when the operation of complete. Writing a 0 to RWAC bits has no effect.
HwInit	Hardware Initialized: Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. Bits are read-only after initialization, and writes to these bits are ignored.
Rsvd	Reserved. These bits are initialized to zero, and writes to them are ignored.
WO	Write-only register. It is not physically implemented register. Rather, it is an address at which registers can be written.

Table 2-3 : Register (and Register Bit-Field) Types

Implementation Note: If the Host Driver writes to RO, ROC, HwInit and Rsvd bits, the Host Driver should write these bits as zero to avoid possible compatibility problems with future versions of this specification.

2.1.3 Register Initial Values

The Host Controller shall set all registers to their initial values at power-on reset. Initial values of the register are defined as follows. All other registers' default value shall be all bits set to zero.
Value of the Capabilities register, *Maximum Current Capabilities* register and *UHS-II Host Capabilities* depends on the Host Controller. Value of the Host Controller Version register depends on the Host Controller.

2.1.4 Reserved Bits of Register

"Reserved" means the bit can be defined for future use and is currently set to 0. These bits should be written as zero.

2.1.5 Register Categories

Registers are classified into following three categories:

Cat.A: Registers used for only SD 4-bit Bus Interface Mode
Cat.B: Registers used for only UHS-II Bus Interface Mode
Cat.C: Registers used for SD 4-bit and UHS-II Bus Interface Mode

In case of Cat.C, usage condition may be described at the title of each field. If no condition is described, it means the field is used in any bus mode.

2.2 Host Controller Interface Register

2.2.1 32-bit Block Count / (SDMA System Address) Register (Cat.A Offset 000h)

D31	D00
SDMA System Address / Argument 2	

Figure 2-1 : 32-bit Block Count / (SDMA System Address) Register

Location	Attrib	Register Field Explanation
31-00	RW	**32-bit Block Count (SDMA System Address)** When **Host Version 4 Enable** is set to 0 in the *Host Control 2* register, SDMA uses this register as system address in only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When **Host Version 4 Enable** is set to 1, SDMA uses *ADMA System Address* register (05Fh-058h) instead of using this register to support both 32-bit and 64-bit addressing. This register is re-assigned to 32-bit Block Count and then SDMA may use Auto CMD23. (1) **SDMA System Address (Host Version 4 Enable** = 0) This register contains the system memory address for an SDMA transfer in 32-bit addressing mode. When the Host Controller stops an SDMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Reading this register during SDMA transfers may return an invalid value. The Host Driver shall initialize this register before starting an SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the **SDMA Buffer Boundary** in the *Block Size* register. The Host Controller generates **DMA Interrupt** to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by setting **Continue Request** in the *Block Gap Control* register, the Host Controller shall start at the next contiguous address stored here in the *SDMA System Address* register. ADMA does not use this register. (2) **32-bit Block Count (Host Version 4 Enable** = 1) Host Controller Version 4.10 re-defines this register as *32-bit Block Count* (Refer to Section 1.15 for more details about block count extension). In version 4.00, this register may be used as 32-bit block count only for Auto CMD23 to set the argument of the CMD23 while executing Auto CMD23. FFFF_FFFFh: 4G - 1 block ...: ... 0000_0002h: 2 blocks 0000_0001h: 1 block 0000_0000h: Stop Count

		The Host Controller would decrement the block count of this register every block transfer and data transfer stops when the count reaches zero. This register should be accessed only when no transaction is executing. Reading this register during data transfers may return invalid value.

Table 2-4 : SDMA System Address / Argument 2 Register

2.2.2 Block Size Register (Cat.A Offset 004h)

This register is used to configure the number of bytes in a data block.

D15	D14 D10	D11 D00
Rsvd	SDMA Buffer Boundary	Transfer Block Size

Figure 2-2 : Block Size Register

Location	Attrib	Register Field Explanation
15	Rsvd	**Reserved**
14-12	RW	**SDMA Buffer Boundary** The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, *SDMA System Address* register shall be updated at every system memory boundary during SDMA transfer. These bits specify the size of contiguous buffer in the system memory. The SDMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the **DMA Interrupt** to request the Host Driver to update the *SDMA System Address* register. At the end of transfer, the Host Controller may issue or may not issue **DMA Interrupt**. In particular, **DMA Interrupt** shall not be issued after **Transfer Complete Interrupt** is issued. In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The SDMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12. These bits shall be supported when the **SDMA Support** in the *Capabilities* register is set to 1 and this function is active when the **DMA Enable** in the *Transfer Mode* register is set to 1. ADMA does not use this register. 000b — 4K bytes (Detects A11 carry out) 001b — 8K bytes (Detects A12 carry out) 010b — 16K Bytes (Detects A13 carry out) 011b — 32K Bytes (Detects A14 carry out) 100b — 64K bytes (Detects A15 carry out) 101b — 128K Bytes (Detects A16 carry out) 110b — 256K Bytes (Detects A17 carry out) 111b — 512K Bytes (Detects A18 carry out)

<table>
<tr><td>11-00</td><td>RW</td><td>Transfer Block Size
This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes (Refer to Implementation Note in Section 1.7.2). It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored.

<table>
<tr><td>0800h</td><td>2048 Bytes</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0200h</td><td>512 Bytes</td></tr>
<tr><td>01FFh</td><td>511 Bytes</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0004h</td><td>4 Bytes</td></tr>
<tr><td>0003h</td><td>3 Bytes</td></tr>
<tr><td>0002h</td><td>2 Bytes</td></tr>
<tr><td>0001h</td><td>1 Byte</td></tr>
<tr><td>0000h</td><td>No data transfer</td></tr>
</table>
</td></tr>
</table>

Table 2-5 : Block Size Register

2.2.3 16-bit Block Count Register (Cat.A Offset 006h)

This register is used to configure the number of data blocks.

D15	D00
Blocks Count For Current Transfer	

Figure 2-3 : 16-bit Block Count Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-00</td><td>RW</td><td>16-bit Block Count
Host Controller Version 4.10 extends block count to 32-bit (Refer to Section 1.15). Selection of either 16-bit Block Count register or 32-bit Block Count register is defined as follows:
(1) If Host Version 4 Enable in the Host Control 2 register is set to 0 or 16-bit Block Count register is set to non-zero, 16-bit Block Count register is selected.
(2) If Host Version 4 Enable is set to 1 and 16-bit Block Count register is set to zero, 32-bit Block Count register is selected.

Use of 16-bit/32-bit Block Count register is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers.
The Host Driver shall set this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks is transferred.

This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.

<table><tr><td>FFFFh</td><td>65535 blocks</td></tr><tr><td>...</td><td>...</td></tr><tr><td>0002h</td><td>2 blocks</td></tr><tr><td>0001h</td><td>1 block</td></tr><tr><td>0000h</td><td>Stop Count</td></tr></table></td></tr>
</table>

Table 2-6 : 16-bit Block Count Register

2.2.4 Argument Register (Cat.A Offset 008h)

This register contains the SD Command Argument.

D31	D00
Command Argument	

Figure 2-4 : Argument Register

Location	Attrib	Register Field Explanation
31-00	RW	**Command Argument** The SD command argument is specified as bit39-8 of Command-Format in the Physical Layer Specification.

Table 2-7 : Argument Register

2.2.5 Transfer Mode Register (Cat.A Offset 00Ch)

This register is used to control the operation of data transfers. The Host Driver shall set this register before issuing a command which transfers data (Refer to **Data Present Select** in the *Command* register), or before issuing a Resume command. The Host Driver shall save the value of this register when the data transfer is suspended (using a Suspend command) and restore it before issuing a Resume command. To prevent data loss, the Host Controller shall implement write protection for this register during data transactions. Writes to this register shall be ignored when the Command Inhibit (DAT) in the *Present State* register is 1.

D15 - D09	D08	D07	D06	D05	D04	D03 - D02	D01	D00
Reserved	Response Interrupt Disable	Response Error Check Enable	Response Type R1/R5	Multi / Single Block Select	Data Transfer Direction Select	Auto CMD Enable	Block Count Enable	DMA Enable

Figure 2-5 : Transfer Mode Register

Location	Attrib	Register Field Explanation
15-09	Rsvd	**Reserved**
08	R/W	**Response Interrupt Disable** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, sets this bit to 0, and waits **Command Complete** Interrupt and then checks the response register. If Host Controller checks response error, sets this bit to 1 and sets **Response Error Check Enable** to 1. **Command Complete** Interrupt is disabled by this bit regardless of **Command Complete Signal Enable.** 0: Response Interrupt is enabled 1: Response Interrupt is disabled
07	R/W	**Response Error Check Enable** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, this bit is set to 0 and **Response Interrupt Disable** is set to 0. If Host Controller checks response error, sets this bit to 1 and sets **Response Interrupt Disable** to 1. **Response Type R1/R5** selects either R1 or R5 response type. If an error is detected, **Response Error** Interrupt is generated in the *Error Interrupt Status* register. 0: Response Error Check is disabled 1: Response Error Check is enabled

<table>
<tr><td>06</td><td>R/W</td><td>Response Type R1/R5
When response error check is enabled, this bit selects either R1 or R5 response types. Two types of response checks are supported: R1 for memory and R5 for SDIO.

Error Statuses Checked in R1
Bit31 OUT_OF_RANGE
Bit30 ADDRESS_ERROR
Bit29 BLOCK_LEN_ERROR
Bit26 WP_VIOLATION
Bit25 CARD_IS_LOCKED
Bit23 COM_CRC_ERROR
Bit21 CARD_ECC_FAILED
Bit20 CC_ERROR
Bit19 ERROR

Response Flags Checked in R5
Bit07 COM_CRC_ERROR
Bit03 ERROR
Bit01 FUNCTION_NUMBER
Bit00 OUT_OF_RANGE

<table><tr><td>0</td><td>R1 (Memory)</td></tr><tr><td>1</td><td>R5 (SDIO)</td></tr></table></td></tr>
<tr><td>05</td><td>RW</td><td>Multi / Single Block Select
This bit is set when issuing multiple-block transfer commands using DAT line. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (Refer to Table 2-9)

<table><tr><td>1</td><td>Multiple Block</td></tr><tr><td>0</td><td>Single Block</td></tr></table></td></tr>
<tr><td>04</td><td>RW</td><td>Data Transfer Direction Select
This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands.

<table><tr><td>1</td><td>Read (Card to Host)</td></tr><tr><td>0</td><td>Write (Host to Card)</td></tr></table></td></tr>
</table>

<table>
<tr><td>03-02</td><td>Rsvd</td><td>

Auto CMD Enable
This field determines use of auto command functions.

00b	Auto Command Disabled
01b	Auto CMD12 Enable
10b	Auto CMD23 Enable
11b	Auto CMD Auto Select

When a multiple-block read/write command that does not have data length information is issued, a setting of this field selects a method to stop the read/write operation that will be invoked by the read/write command. Auto CMD12 is defined from Version 1.00, Auto CMD23 is added from Version 3.00 and Auto CMD Auto Select is added from Version 4.10. This field is set to 00b for the other commands (single read/write commands, multiple-block read/write commands that have data length information, commands other than read/write).

(1) Auto CMD12 Enable
When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the *Auto CMD Error Status* register.
The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands defined in the Part 3 File Security specification do not require CMD12.
When **Host Version 4 Enable** =0, CMD12 is issued when **16-bit Block Count** is expired.
When **Host Version 4 Enable** =1, CMD12 is issued when **16-bit Block Count** or **32-bit Block Count** is expired.

(2) Auto CMD23 Enable
When this bit field is set to 10b, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command register. The Host Controller Version 3.00 and later shall support this function. The following conditions are required to use the Auto CMD23.

- Auto CMD23 Supported (Host Controller Version is 3.00 or later)
- A memory card that supports CMD23 (SCR[33]=1)
- If DMA is used, it shall be ADMA.
- Only when CMD18 or CMD25 is issued
(Note, the Host Controller does not check command index.)

Auto CMD23 can be used with or without ADMA. By writing the *Command* register, the Host Controller issues a CMD23 first and then issues a command specified by the **Command Index** in *Command* register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the *Auto CMD Error Status* register.
32-bit block count value for CMD23 is set to *32-bit Block Count (SDMA System Address)* register.

(3) Auto CMD Auto Select (Version 4.10)
As CMD23 is optional for SD Memory Card except UHS104 Card, If card supports CMD23, Auto CMD23 should be used instead of Auto CMD12. Host Controller Version 4.10 defines this "Auto CMD Auto Select" mode. Selection of Auto CMD depends on setting of **CMD23 Enable** in the *Host Control 2* register, which indicates whether card supports CMD23. If **CMD23 Enable** =1, Auto CMD23 is used and if **CMD23 Enable** =0, Auto CMD12 is used. In case of Version 4.10 or later, use of Auto CMD Auto Select is recommended rather than use of Auto CMD12 Enable or Auto CMD23 Enable.

</td></tr>
</table>

01	RW	**Block Count Enable** This bit is used to enable the *Block Count* register, which is only relevant for multiple block transfers. When this bit is 0, the *Block Count* register is disabled, which is useful in executing an infinite transfer. (Refer to Table 2-9) 1 : Enable 0 : Disable
00	RW	**DMA Enable** This bit enables DMA functionality as described in section 1.4. DMA can be enabled only if it is supported as indicated in the *Capabilities* register. One of the DMA modes can be selected by **DMA Select** in the *Host Control 1* register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of *Command* register (00Fh). 1 : DMA Data transfer 0 : No data transfer or Non DMA data transfer

Table 2-8 : Transfer Mode Register

Table 2-9 shows the summary of how register settings determine types of data transfer.

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

Table 2-9 : Determination of Transfer Type

2.2.6 Command Register (Cat.A Offset 00Eh)

The Host Driver shall check the **Command Inhibit (DAT)** bit and **Command Inhibit (CMD)** bit in the *Present State* register before writing to this register (except while data transfer is being stopped by **Stop At Block Gap Request**). Writing to the upper byte of this register triggers SD command generation. The Host Driver has the responsibility to write this register because the Host Controller does not protect for writing when **Command Inhibit (CMD)** is set.

Host Controller prior to Version 4.20 is capable to issue an abort command according to Section 3.8. Host Controller from Version 4.20 is capable to issue further any command without using DAT line (in SD mode) and any UHS-II command regardless of **Command Type** in this register including CMD52 during data transfer, which is defined by the SDIO Specification Version 4.10. Host Driver shall manage SD commands can be issued depends on card protocol specification (e.g., UHS-II mode, SDIO).

Even SD Clock has been stopped in SD mode to halt read operation by the Stop At Block Gap Request, Host Controller may provide SD Clock to issue an abort command and data circuits including DMA should be still stopped.

D15 D14	D13 D08	D07 D06	D05	D04	D03	D02	D01 D00
Rsvd	Command Index	Command Type	Data Present Select	Command Index Check Enable	Command CRC Check Enable	Sub Command Flag	Response Type Select

Figure 2-6 : Command Register

Location	Attrib	Register Field Explanation
15-14	Rsvd	**Reserved**
13-08	RW	**Command Index** These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the Physical Layer Specification and SDIO Card Specification.

<table>
<tr><td>07-06</td><td>RW</td><td>Command Type
There are three types of special commands: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands.

(1) Suspend Command
If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command, which uses the DAT line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting Continue Request in the Block Gap Control register. (Refer to 3.12.1 Suspend Sequence)
(2) Resume Command
The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to Figure 1-4 in section 1.6 for the register map.) The Host Controller shall check for busy before starting write transfers.
(3) Abort Command
If this command is set when executing a read transfer, the Host Controller may discard read data (stop reading data to the buffer). If this command is set when executing a write transfer, the Host Controller shall stop driving the DAT line. After issuing the Abort command, the Host Driver should issue a software reset to discard data in the Host Controller buffer. (Refer to 3.8 Abort Transaction)

<table>
<tr><td>11b</td><td>Abort</td><td>CMD12, CMD52 for writing "I/O Abort" in CCCR</td></tr>
<tr><td>10b</td><td>Resume</td><td>CMD52 for writing "Function Select" in CCCR</td></tr>
<tr><td>01b</td><td>Suspend</td><td>CMD52 for writing "Bus Suspend" in CCCR</td></tr>
<tr><td>00b</td><td>Normal</td><td>Other commands</td></tr>
</table></td></tr>
<tr><td>05</td><td>RW</td><td>Data Present Select
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following:

(1) Commands using only CMD line (ex. CMD52).
(2) Commands with no data transfer but using busy signal on DAT[0] line (R1b or R5b ex. CMD38)
(3) Resume command

<table>
<tr><td>1</td><td>Data Present</td></tr>
<tr><td>0</td><td>No Data Present</td></tr>
</table></td></tr>
<tr><td>04</td><td>RW</td><td>Command Index Check Enable
If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked.

<table>
<tr><td>1</td><td>Enable</td></tr>
<tr><td>0</td><td>Disable</td></tr>
</table></td></tr>
</table>

03	RW	**Command CRC Check Enable** If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The position of CRC field is determined according to the length of the response. (Refer to definition in D01-00 and Table 2-11 below.) 1 — Enable 0 — Disable
02	R/W	**Sub Command Flag** This bit is added from Version 4.10 to distinguish a main command or sub command (Refer to Section 1.17). When issuing a main command, this bit is set to 0 and when issuing a sub command, this bit is set to 1. Setting of this bit is checked by **Sub Command Status** in the *Present State* register. Host Driver manages whether main or sub command. Host Controller does not refer to this bit to issue a command. 1 — Sub Command 0 — Main Command
01-00	RW	**Response Type Select** 00 — No Response 01 — Response Length 136 10 — Response Length 48 11 — Response Length 48 check Busy after response

Table 2-10 : Command Register

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6, R7
11	1	1	R1b, R5b

Table 2-11 : Relation between Parameters and the Name of Response Type

These bits determine Response types.
Note: In the SDIO specification, response type notation of R5b is not defined. R5 includes R5b in the SDIO specification. However, R5b is defined in this specification to specify the Host Controller shall check busy after receiving response. For example, usually CMD52 is used as R5 but I/O abort command shall be used as R5b.

Implementation Note: the CRC field for R3 and R4 is expected to be all "1" bits. The CRC check should be disabled for these response types.

2.2.7 Response Register (Cat.C Offset 010h)

This register is used to store responses from SD cards.

Offset 010h	D31	D00
	Command Response 0 – 31	
Offset 014h	D31	D00
	Command Response 32 – 63	
Offset 018h	D31	D00
	Command Response 64 – 95	
Offset 01Ch	D31	D00
	Command Response 96 – 127	

Figure 2-7 : Response Register

Location	Attrib	Register Field Explanation
127-00	ROC	**Command Response** The Table 2-13 describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the *Response* register.

Table 2-12 : Response Register

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc.	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc.	R [39:8]	REP [31:0]

Table 2-13 : Response Bit Definition for Each Response Type.

The Response Field indicates bit positions of "Responses" defined in the Physical Layer Specification.

The Table 2-13 shows that most responses with a length of 48 (R[47:0]) have 32 bits of the response data (R[39:8]) stored in the *Response* register at REP[31:0]. Responses of type R1b (Auto CMD12 responses) and R1 (Auto CMD23 response) have response data bits R[39:8] stored in the *Response* register at REP[127:96]. Responses with length 136 (R[135:0]) have 120 bits of the response data (R[127:8]) stored in the *Response* register at REP[119:0].
To be able to read the response status efficiently, the Host Controller only stores part of the response data in the *Response* register. This enables the Host Driver to read 32 bits of response data efficiently in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the Host Controller (as specified by the **Command Index Check Enable** and the **Command CRC Check Enable** bits in the *Command* register) and generate an error interrupt if an error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the Host Controller shall check R[47:1], and if the response length is 136 the Host Controller shall check R[119:1].

Since the Host Controller may transfer a multiple blocks of data through DAT line with executing a CMD_wo_DAT command concurrently, the Host Controller stores the Auto CMD12 response in the upper bits (REP[127:96]) of the *Response* register. The CMD_wo_DAT response is stored in REP[31:0].

This allows the Host Controller to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa.

While executing Auto CMD23, the response of CMD23 is saved to REP [127:96] and the response of multiple-block read and write command is save to REP [31:0]. The response error of CMD23 is indicated in the *Auto CMD Error Status* register.

When the Host Controller modifies part of the *Response* register, as shown in the Table 2-13, it shall preserve the unmodified bits.

In UHS-II mode, the response of CM-TRAN abort CCMD (4-byte) is stored in offset 13h-10h and the response of SD-TRAN abort CCMD (8-byte) is stored in offset 1Fh-18h

2.2.8 Buffer Data Port Register (Cat.C Offset 020h)

32-bit data port register to access internal buffer.

D31	D00
Buffer Data	

Figure 2-8 : Buffer Data Port Register

Buffer can be accessed through 32-bit *Data Port* register.

Location	Attrib	Register Field Explanation
31-00	RW	**Buffer Data** The Host Controller buffer can be accessed through this 32-bit *Data Port* register. Refer to Section 1.7

Table 2-14 : Buffer Data Port Register

2.2.9 Present State Register (Cat.C Offset 024h)

The Host Driver can get status of the Host Controller from this 32-bit read only register.

D31	D30	D29	D28	D27	D26	D25	D24	D23 - D20	D19	D18	D17	D16
UHS-II IF Detection	Lane Synchronization	In Dormant State	Sub Command Status	Command Not Issued by Error	Rsvd	Host Regulator Voltage Stable	CMD Line Signal Level	DAT[3:0] Line Signal Level	Write Protect Switch Pin Level	Card Detect Pin Level	Card State Stable	Card Inserted

D15 - D12	D11	D10	D09	D08	D07 - D04	D03	D02	D01	D00
Rsvd	Buffer Read Enable	Buffer Write Enable	Read Transfer Active	Write Transfer Active	DAT[7:4] Line Signal Level	Re-Tuning Request	DAT Line Active	Command Inhibit (DAT)	Command Inhibit (CMD)

Figure 2-9 : Present State Register

Location	Attrib	Register Field Explanation
31	RO	**UHS-II IF Detection (UHS-II Only)** This status indicates whether a card supports UHS-II IF. This status is enabled by setting **UHS-II Interface Enable** to 1 in the *Host Control 2* register. UHS-II interface initialization is activated by setting **SD Clock Enable** in the *Clock Control* register. Host Controller drives STB.L on D0 lane from EIDL state and waits for receiving STB.L on D1 lane. This bit is set to 1 if STB.L is detected on D1 lane. Host Controller shall compensate latency from setting **SD Clock Enable** to output STB.L on D0 lane when reading this status (Refer to Figure 3-36 about details of this method). This bit may be read any time after setting **SD Clock Enable** for faster UHS-II IF detection but Host Driver shall check this status at least 200us period from setting **SD Clock Enable** until detecting UHS-II IF. After UHS-II IF is detected, this bit is cleared by when EIDL is detected on D0 lane, **UHS-II Interface Enable** is set to 0 or Host full reset is executed. 1 : UHS-II IF is detected 0 : UHS-II IF is not detected Refer to Section 3.13.2 for more details about sequence of detecting UHS-II IF and checking PHY Initialization Completion.

<table>
<tr><td>30</td><td>RO</td><td>Lane Synchronization (UHS-II only)
This status indicates whether lane is synchronized in UHS-II mode. This status is enabled by setting UHS-II Interface Enable to 1 in the Host Control 2 register. On detecting UHS-II Interface (D31=1), Host Controller provides SYN LSS on D0 lane and waits for receiving SYN LSS on D1 lane. If SYN LSS is detected on D1 lane, Host Controller provides LIDL LSS on D0 lane and waits for receiving LIDL LSS on D1 lane.

In case of Version 4.00, this bit indicates completion of Device PHY Initialization by detecting LIDL LSS on D1 lane.
From Version 4.10, Host Controller may implement a specific PHY verification method and PHY Initialization Failure can be indicated by keeping this bit to 0 even LIDL LSS is detected on D1 lane. Host Driver detects PHY Initialization Failure by timeout.
This bit is cleared by when D0 lane is set to EIDL, UHS-II Interface Enable is set to 0 or executes Host full reset.

<table><tr><td>1</td><td>UHS-II PHY is initialized</td></tr><tr><td>0</td><td>UHS-II PHY is not initialized</td></tr></table>
Refer to Section 3.13.2 for more details about checking PHY Initialization Completion.</td></tr>
<tr><td>29</td><td>RO</td><td>In Dormant State (UHS-II only)
This status indicates whether UHS-II lanes enter Dormant state. This function is enabled by setting UHS-II Interface Enable to 1 in the Host Control 2 register. On issuing GO_DORMAT_STATE command, "Go Dormant Command (111b)" is set to Command Type in the UHS-II Command register. This command type acts as a trigger to enter lanes into dormant state. Host Controller provides STB.H and EIDL on D0 lane and waits for receiving STB.H and EIDL on D1 lane. This bit is set to 1 after the time of T_DMT_ENTRY (750 RCLK cycle) or more from detecting EIDL on D1 lane.

<table><tr><td>1</td><td>In DORMANT state</td></tr><tr><td>0</td><td>Not in DORMANT state</td></tr></table>
RCLK may be stopped in dormant state, by setting SD Clock Enable to 0 in the Clock Control register while In Dormant State bit is set to 1. On writing Clock Control register with setting SD Clock Enable to 1, Host Controller wakes lanes to exit Dormant State and In Dormant State is set to 0.

In case of the card enters Hibernate Mode (RCLK is stopped), Host Driver may turn off VDD1 by clearing SD Bus Power for VDD1 bit in the Power Control register. Host Controller shall turn off VDD1 after stopping RCLK.
This bit is cleared by when Host Controller drives STB.L on D0 lane, UHS-II Interface Enable is set to 0 or executes Host full reset.</td></tr>
</table>

<table>
<tr><td>28</td><td>RO</td><td>Sub Command Status
The Command register and Response register are commonly used for main command and sub command. This status is used to distinguish which response error statuses, main command or sub command, indicated in the Error Interrupt Status register or in the UHS-II Error Interrupt Status register. Refer to Section 1.17 about details of response error statuses. Just before reading of this register, the Sub Command Flag of the Command register or the UHS-II Command register is copied to this status. This status is effective when not only Response Error interrupt is generated but also data error interrupt is generated with Command Not Issued by Error (D27 of this register) or Auto CMD Error interrupt is generated with Command Not Issued by Error by Auto CMD12 in the Auto CMD Error Status register.
<table><tr><td>1</td><td>Sub Command Status</td></tr><tr><td>0</td><td>Main Command Status</td></tr></table></td></tr>
<tr><td>27</td><td>RO</td><td>Command Not Issued by Error
Setting of this status indicates that a command cannot be issued due to an error except Auto CMD12 error. (Equivalent error status by Auto CMD12 error is defined as Command Not Issued By Auto CMD12 Error in the Auto CMD Error Status register.) This status is set to 1 when Host Controller cannot issue a command after setting Command register or UHS-II Command register. Refer to Section 3.10 about 2L-HD error case in UHS-II mode.
Sub Command Status (D28) indicates which command is not issued (main or sub).
<table><tr><td>1</td><td>Command cannot be issued</td></tr><tr><td>0</td><td>No error for issuing a command</td></tr></table></td></tr>
<tr><td>26</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

25	RO	**Host Regulator Voltage Stable** This status is added from Version 4.10 and is used to check whether host regulator voltage is stable for switching signal voltage of UHS-I mode. 1: Host Regulator Voltage is stable 0: Host Regulator Voltage is not stable Support of this function is checked by reading this status after that **Software Reset For All** in the *Software Reset* register is cleared by the Host Controller in initialization. Setting of this status means that the Host Controller supports this function. This status may be related to **1.8V Signaling Enable** in the *Host Control 2* register. Changing **1.8V Signaling Enable** causes unstable of host regulator voltage for I/O cell. Then once this status is set to 0 and retrieved to 1 when host regulator voltage is stable again. When executing power cycle, Host Driver also executes **Software Reset For All** and it clears 1.8V Signaling Enable to go back signal voltage to 3.3V. If this status is not supported, Host Driver should take more than 5ms for stable time of host voltage regulator from changing **1.8V Signaling Enable**. Specific Host Driver may use a specific time, which is provided by Host System, instead of using 5ms.
24	RO	**CMD Line Signal Level (SD Mode only)** This status is used to check the ***CMD*** line level to recover from errors, and for debugging.
23-20	RO	**DAT[3:0] Line Signal Level (SD Mode only)** This status is used to check the ***DAT*** line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from ***DAT***[0]. D23: DAT[3] D22: DAT[2] D21: DAT[1] D20: DAT[0]
19	RO	**Write Protect Switch Pin Level** The Write Protect Switch is supported for memory and combo cards. This bit reflects the ***SDWP#*** pin. 1: Write enabled (***SDWP#***=1) 0: Write protected (***SDWP#***=0)
18	RO	**Card Detect Pin Level** This bit reflects the inverse value of the ***SDCD***# pin. Debouncing is not performed on this bit. This bit may be valid when **Card State Stable** is set to 1, but it is not guaranteed because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. 1: Card present (***SDCD#***=0) 0: No card present (***SDCD#***=1)

17	RO	**Card State Stable** This bit is used for testing. If it is 0, the **Card Detect Pin Level** is not stable. If this bit is set to 1, it means the **Card Detect Pin Level** is stable. No Card state can be detected by this bit is set to 1 and **Card Inserted** is set to 0. The **Software Reset For All** in the *Software Reset* register shall not affect this bit. 1 — No Card or Inserted 0 — Reset or Debouncing
16	RO	**Card Inserted** This bit indicates whether a card has been inserted. The Host Controller shall debounce this signal so that the Host Driver will not need to wait for it to stabilize. Changing from 0 to 1 generates a **Card Insertion** interrupt in the *Normal Interrupt Status* register and changing from 1 to 0 generates a **Card Removal** interrupt in the *Normal Interrupt Status* register. The **Software Reset For All** in the *Software Reset* register shall not affect this bit. If a card is removed while its power is on and its clock is oscillating, the Host Controller shall clear **SD Bus Power** in the *Power Control* register (Refer to Section 2.2.12) and **SD Clock Enable** in the *Clock Control* register (Refer to Section 2.2.15). When this bit is changed from 1 to 0, the Host Controller shall immediately stop driving ***CMD*** and ***DAT[3:0]*** (tri-state). In addition, the Host Driver should clear the Host Controller by the **Software Reset For All** in *Software Reset* register. The card detect is active regardless of the **SD Bus Power**. 1 — Card Inserted 0 — Reset or Debouncing or No Card

Table 2-15 : Present State Register (Part 1)

Figure 2-10 shows the state definitions of hardware that handles "Debouncing".

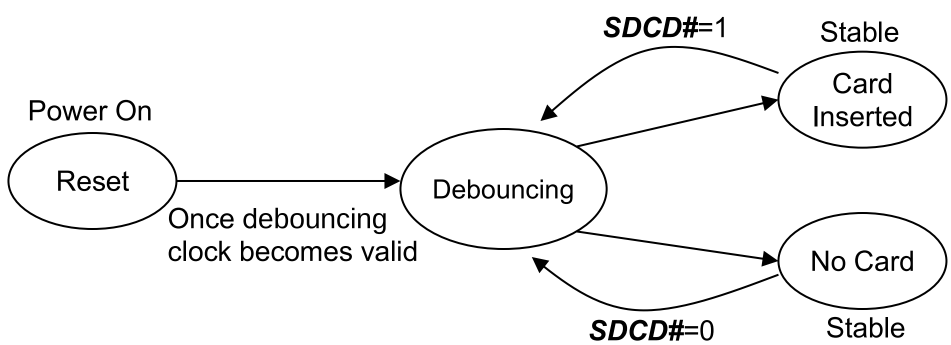


Figure 2-10 : Card Detect State

> Implementation Note: The Host Controller starts in "Reset" state at power on and changes to the "Debouncing" state once the debouncing clock is valid. In the "Debouncing" state, if the Host Controller detects that the signal (***SDCD#***) is stable during the debounce period, the state shall change to "Card Inserted" or "No Card". If the card is removed while in the "Card Inserted" state, it will immediately change to the "Debouncing" state. Since the card detect signal is then not stable, the Host Controller will change to the "Debouncing" state.

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-12</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>11</td><td>ROC</td><td>Buffer Read Enable
This status is used for non-DMA read transfers.
The Host Controller may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when block data is ready in the buffer and generates the Buffer Read Ready interrupt.

<table><tr><td>1</td><td>Read enable</td></tr><tr><td>0</td><td>Read disable</td></tr></table></td></tr>
<tr><td>10</td><td>ROC</td><td>Buffer Write Enable
This status is used for non-DMA write transfers.
The Host Controller can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when top of block data can be written to the buffer and generates the Buffer Write Ready interrupt. The Host Controller should neither set Buffer Write Enable nor generate Buffer Write Ready Interrupt after the last block data is written to the Buffer Data Port register.

<table><tr><td>1</td><td>Write enable</td></tr><tr><td>0</td><td>Write disable</td></tr></table></td></tr>
<tr><td>09</td><td>ROC</td><td>Read Transfer Active (SD Mode only)
This status is used for detecting completion of a read transfer. Refer to Section 3.12.3 for sequence details.

This bit is set to 1 for either of the following conditions:
(1) After the end bit of the read command.
(2) When read operation is restarted by writing a 1 to Continue Request in the Block Gap Control register.

This bit is cleared to 0 for either of the following conditions::
(1) When the last data block as specified by block length is transferred to the System.
(2) In case of ADMA2, end of read operation is designated by Descriptor Table.
(3) When all valid data blocks in the Host Controller have been transferred to the System and no current block transfers are being sent as a result of the Stop At Block Gap Request being set to 1.

A Transfer Complete interrupt is generated when this bit changes to 0.

<table><tr><td>1</td><td>Transferring data</td></tr><tr><td>0</td><td>No valid data</td></tr></table></td></tr>
<tr><td>08</td><td>ROC</td><td>Write Transfer Active (SD Mode only)
This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the Host Controller. Refer to Section 3.12.4 for more details on the sequence of events.</td></tr>
</table>

<table>
<tr><td></td><td></td><td>This bit is set in either of the following cases:
(1) After the end bit of the write command.
(2) When write operation is restarted by writing a 1 to Continue Request in the Block Gap Control register.
This bit is cleared in either of the following cases:
(1) After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) In case of ADMA2, transfer count is designated by Descriptor Table.
(2) After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request.

During a write transaction, a Block Gap Event interrupt is generated when this bit is changed to 0, as the result of the Stop At Block Gap Request being set. This status is useful for the Host Driver in determining non-DAT line commands can be issued during write busy.

<table><tr><td>1</td><td>Transferring data</td></tr><tr><td>0</td><td>No valid data</td></tr></table></td></tr>
<tr><td>07-04</td><td>RO</td><td>DAT[7:4] Line Signal Level (Embedded only)
This status is used to check the DAT line level to recover from errors, and for debugging.
.
<table><tr><td>D07</td><td>DAT[7]</td></tr><tr><td>D06</td><td>DAT[6]</td></tr><tr><td>D05</td><td>DAT[5]</td></tr><tr><td>D04</td><td>DAT[4]</td></tr></table></td></tr>
<tr><td>03</td><td>ROC</td><td>Re-Tuning Request (UHS-I Only)
Host Controller may request Host Driver to execute re-tuning sequence by setting this bit when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data.
This bit is cleared when a command is issued with setting Execute Tuning in the Host Control 2 register.
Changing of this bit from 0 to 1 generates Re-Tuning Event. Refer to Normal Interrupt Status registers for more detail.
This bit is not set to 1 if Sampling Clock Select in the Host Control 2 register is set to 0 (using fixed sampling clock). Refer to Re-Tuning Modes in the Capabilities register for more detail.

<table><tr><td>1</td><td>Sampling clock needs re-tuning</td></tr><tr><td>0</td><td>Fixed or well-tuned sampling clock</td></tr></table></td></tr>
<tr><td>02</td><td>ROC</td><td>DAT Line Active (SD Mode only)
This bit indicates whether one of the DAT line on SD Bus is in use.

(a) In the case of read transactions
This status indicates whether a read transfer is executing on the SD Bus. Changing this value from 1 to 0 generates a Block Gap Event interrupt in the Normal Interrupt Status register, as the result of the Stop At Block Gap Request being set. Refer to Section 3.12.3 for details on timing.

This bit shall be set in either of the following cases:
(1) After the end bit of the read command.</td></tr>
</table>

		(2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer. This bit shall be cleared in either of the following cases: (1) When the end bit of the last data block is sent from the SD Bus to the Host Controller. In case of ADMA2, the last block is designated by the last transfer of Descriptor Table. (2) When a read transfer is stopped at the block gap initiated by a **Stop At Block Gap Request.** The Host Controller shall stop read operation at the start of the interrupt cycle of the next block gap by driving Read Wait or stopping SD clock. If the Read Wait signal is already driven (due to data buffer cannot receive data), the Host Controller can continue to stop read operation by driving the Read Wait signal. It is necessary to support Read Wait in order to use Suspend/Resume function. (b) In the case of write transactions This status indicates that a write transfer is executing on the SD Bus. Changing this value from 1 to 0 generate a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. Refer to Section 3.12.4 for sequence details. This bit shall be set in either of the following cases: (1) After the end bit of the write command. (2) When writing to 1 to **Continue Request** in the *Block Gap Control* register to continue a write transfer. This bit shall be cleared in either of the following cases: (1) When the SD card releases write busy of the last data block. If SD card does not drive busy signal for 8 SD Clocks, the Host Controller shall consider the card drive "Not Busy". In case of ADMA2, the last block is designated by the last transfer of Descriptor Table. (2) When the SD card releases write busy prior to waiting for write transfer as a result of a **Stop At Block Gap Request**. (c) Command pairing with response-with-busy This status indicates whether a command indicates busy (e.g., erase command for memory) is executing on the SD Bus. This bit is set after the end bit of the command and is cleared when busy after the response is de-asserted. Changing this bit from 1 to 0 generate a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. Refer Figure 2-11 to Figure 2-13. 1 : DAT Line Active 0 : DAT Line Inactive
01	ROC	**Command Inhibit (DAT) (SD Mode only)** Setting this status to 1 indicates that Host Controller is currently in a state, which cannot issue a command using DAT line. While data transfer is being stopped by **Stop At Block Gap Request**, Host Driver shall not issue any command using DAT line (except an abort command) regardless of this status. This status bit is generated if either the **DAT Line Active** or the **Read Transfer Active** is set to 1. If this bit is 0, it indicates the Host Controller can issue the next SD Command. Commands with busy signal belong to **Command Inhibit (DAT)** (ex. R1b, R5b type). Changing from 1 to 0 generates a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. Note: The SD Host Driver can save registers in the range of 000-00Dh for a

		suspend transaction after this bit has changed from 1 to 0. 1 : Cannot issue command which uses the ***DAT*** line 0 : Can issue command which uses the ***DAT*** line
00	ROC	**Command Inhibit (CMD)** Setting this status to 1 indicates that Host Controller is currently in a state, which cannot issue a command using only CMD line or a UHS-II command. While data transfer is being stopped by **Stop At Block Gap Request**, this status is indicated to 0. **(1) SD Mode** If this bit is 0, it indicates the ***CMD*** line is not in use and the Host Controller can issue an SD Command using the ***CMD*** line. This bit is set immediately after the *Command* register (00Fh) is written. This bit is cleared when the command response is received. Auto CMD12 and Auto CMD23 consist of two responses. In this case, this bit is not cleared by the response of CMD12 or CMD23 but cleared by the response of a read/write command. Status issuing Auto CMD12 is not read from this bit. Therefore, if a command is issued during Auto CMD12 operation, Host Controller shall manage to issue two commands: CMD12 and a command set by *Command* register. Even if the **Command Inhibit (DAT)** is set to 1, commands using only the ***CMD*** line can be issued if this bit is 0. Changing from 1 to 0 generates a **Command Complete Interrupt** in the *Normal Interrupt Status* register. If the Host Controller cannot issue the command because of a command conflict error (Refer to **Command CRC Error** in Section 2.2.19) or because of **Command Not Issued By Auto CMD12 Error** (Refer to Section 2.2.24), this bit shall remain 1 and the **Command Complete** is not set. **(2) UHS-II Mode** This bit is 0 means that a command packet can be issued by the Host Controller. While this bit is set to 1, which means the Host Controller is not ready to issue a next command, Host Driver shall not write the registers from *UHS-II Block Size* (Offset 080h) to the *UHS-II Command* (Offset 09Eh). Changing from 1 to 0 generates a **Command Complete Interrupt** in the *Normal Interrupt Status* register. 1 : Host Controller is not ready to issue a command 0 : Host Controller is ready to issue a command Version 4.10 adds a new control to prevent error statuses from overwriting by receipt of a next command. This status keeps indicating 1 while any of response error statuses is set to 1 (as described in Section 1.17), **Command Not Issued by Error** in this register is set to 1 or **Command Not Issued by Auto CMD12 Error** in the *Auto CMD Error Status* register is set to 1. **Software Reset For CMD Line** is used to clear the error statuses above and this status.

Table 2-16 : Present State Register (Part 2)

Implementation Note:
The Host Driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the **DAT** lines are busy during data transfer. These commands can be issued when **Command Inhibit (CMD)** is set to zero. Other commands shall be issued when **Command Inhibit (DAT)** is set to zero.

Possible changes to the Physical Layer Specification may add other commands to this list in the future.

Implementation Note: Some fields defined in the Present State register change values asynchronous to the system clock. The System reads these statuses through the System Bus Interface and it may require data stable period during bus cycle. The Host Controller should sample and hold values during reads from this register according to the timing required by the System Bus Interface specification.

Figure 2-11Figure 2-11 to Figure 2-13 shows the timing of setting and clearing the **Command Inhibit (DAT)** and the **Command Inhibit (CMD)**.

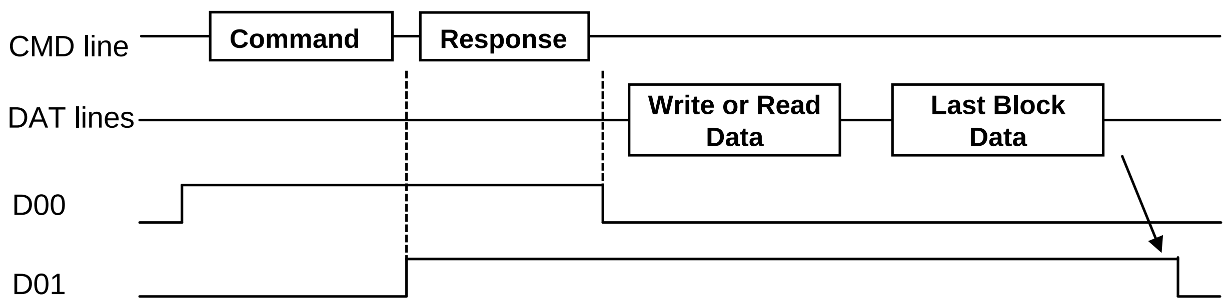


Figure 2-11 : Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with Data Transfer

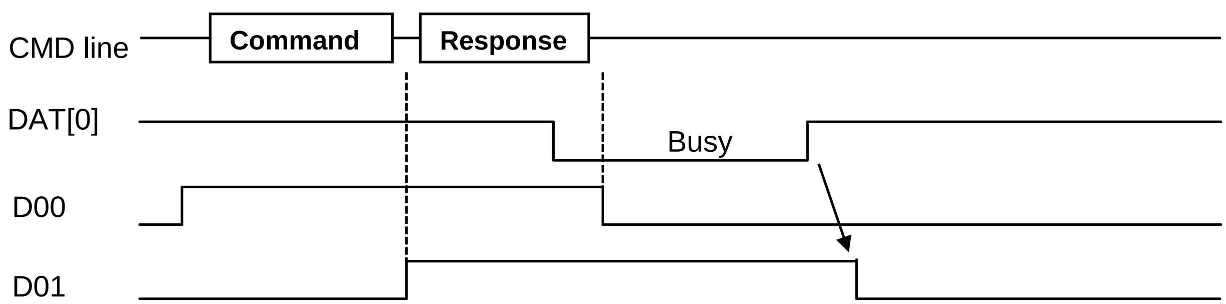


Figure 2-12 : Timing of Command Inhibit (DAT) for the Case of Response with Busy

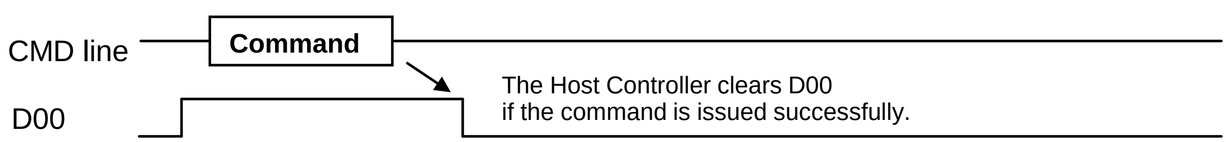


Figure 2-13 : Timing of Command Inhibit (CMD) for the Case of No Response Command

2.2.11 Host Control 1 Register (Cat.C Offset 028h)

D07	D06	D05	D04 D03	D02	D01	D00
Card Detect Signal Selection	Card Detect Test Level	Extended Data Transfer Width	DMA Select	High Speed Enable	Data Transfer Width	LED Control

Figure 2-14 : Host Control 1 Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>07</td><td>RW</td><td>Card Detect Signal Selection
This bit selects source for the card detection.

1: The Card Detect Test Level is selected (for test purpose)
0: SDCD# is selected (for normal use)

When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch.
The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.</td></tr>
<tr><td>06</td><td>RW</td><td>Card Detect Test Level
This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not.

1: Card Inserted
0: No Card</td></tr>
<tr><td>05</td><td>RW</td><td>Extended Data Transfer Width (Embedded and SD Mode only)
This bit controls 8-bit bus width mode for embedded device. Support of this function is indicated in 8-bit Support for Embedded Device in the Capabilities register. If a device supports 8-bit bus mode, this bit may be set to 1. If this bit is 0, bus width is controlled by Data Transfer Width in the Host Control 1 register.
This bit is not effective when multiple devices are installed on a bus slot (Slot Type is set to 10b in the Capabilities register). In this case, each device bus width is controlled by Bus Width Preset field in the Embedded Control register.

1: 8-bit Bus Width
0: Bus Width is Selected by Data Transfer Width</td></tr>
<tr><td>04-03</td><td>RW</td><td>DMA Select
This field is used to select DMA type. The Host Driver shall check support of DMA modes by referring the Capabilities register. Selected DMA is enabled by DMA Enable of the Transfer Mode register in SD mode and DMA Enable of UHS-II Transfer Mode register in UHS-II mode.

(1) Up to Version 3.00
When Host Version 4 Enable is set to 0, setting of this field is compatible to Host Controller Version 3.00.
SDMA is initiated by writing to the Command register when this field is set to 00b and the SDMA System Address register (32-bit) is used. SDMA does not</td></tr>
</table>

<table>
<tr><td></td><td></td><td>
support 64-bit addressing.
ADMA2 is initiated by writing to the Command register when this field is set to 10b or 11b. Lower 32-bit of the ADMA System Address register is used when this field is set to 10b and 64-bit of the ADMA System Address register is used when this field is set to 11b. Support of 64-bit System Addressing is indicated by 64-bit System Address Support for V3 in the Capabilities register. 64-bit ADMA2 uses 96-bit Descriptor.
<table>
<tr><td>00</td><td>SDMA is selected</td></tr>
<tr><td>01</td><td>Reserved (New assignment is not allowed)</td></tr>
<tr><td>10</td><td>32-bit Address ADMA2 is selected</td></tr>
<tr><td>11</td><td>64-bit Address ADMA2 is selected (Optional)</td></tr>
</table>
(2) Version 4.00 or later
When Host Version 4 Enable is set to 1, setting of this field is changed as follows.
SDMA is initiated by Host Driver writes to the Command register when this field is set to 00b.
ADMA2 is initiated by Host Driver writes to the Command register when this field is set to 10b or 11b and by ADMA3 sets to the ADMA System Address register when this field is set to 11b.
ADMA3 is initiated by Host Driver writes to the ADMA3 ID Address register when this field is set to 11b.
<table>
<tr><td>00</td><td>SDMA is selected</td></tr>
<tr><td>01</td><td>Not Used (New assignment is not allowed)</td></tr>
<tr><td>10</td><td>ADMA2 is selected (ADMA3 is not supported or disabled)</td></tr>
<tr><td>11</td><td>ADMA2 or ADMA3 is selected</td></tr>
</table>
Support of 64-bit DMA and 128-bit Descriptor is indicated by 64-bit System Address Support for V4 in the Capabilities register. If the support bit is set to 1, all supported DMAs (depends on Support, ADMA2 Support and ADMA3 Support) shall support 64-bit addressing. 64-bit Addressing in the Host Controller 2 register selects either 32-bit or 64-bit system addressing of DMAs.
</td></tr>
<tr><td>02</td><td>RW</td><td>
High Speed Enable (SD Mode only)
This bit is optional. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock (up to 50MHz).

If Preset Value Enable in the Host Control 2 register is set to 1, Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitches. After setting this field, the Host Driver sets SD Clock Enable again.
This bit is not effective in UHS-II mode.
<table>
<tr><td>1</td><td>High Speed mode</td></tr>
<tr><td>0</td><td>Normal Speed mode</td></tr>
</table>
</td></tr>
<tr><td>01</td><td>RW</td><td>
Data Transfer Width (SD Mode only)
This bit selects the data width of the Host Controller. The Host Driver shall set it to
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>match the data width of the SD card.
This bit is not effective in UHS-II mode.

<table><tr><td>1</td><td>4-bit mode</td></tr><tr><td>0</td><td>1-bit mode</td></tr></table></td></tr>
<tr><td>00</td><td>RW</td><td>LED Control
This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction.

<table><tr><td>1</td><td>LED on</td></tr><tr><td>0</td><td>LED off</td></tr></table></td></tr>
</table>

Table 2-17 : Host Control 1 Register

2.2.12 Power Control Register (Cat.C Offset 029h)

D07 D05	D04	D03 D01	D00
SD Bus Voltage Select for VDD2	SD Bus Power for VDD2	SD Bus Voltage Select for VDD1	SD Bus Power for VDD1

Figure 2-15 : Power Control Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>07-05</td><td>RW</td><td>SD Bus Voltage Select for VDD2 (UHS-II Only)
This field determines supply voltage range to VDD2. This field can be set to 101b if 1.8V VDD2 Support in the Capabilities register is set to 1.

<table>
<tr><td>111b</td><td>Not used</td></tr>
<tr><td>110b</td><td>Not used</td></tr>
<tr><td>101b</td><td>1.8V</td></tr>
<tr><td>100b</td><td>Reserved for 1.2V</td></tr>
<tr><td>011b – 001b</td><td>Reserved</td></tr>
<tr><td>000b</td><td>VDD2 Not Supported</td></tr>
</table></td></tr>
<tr><td>04</td><td>RW</td><td>SD Bus Power for VDD2 (UHS-II Only)
Setting this bit enables providing VDD2.
<table>
<tr><td>1</td><td>Power on</td></tr>
<tr><td>0</td><td>Power off</td></tr>
</table></td></tr>
<tr><td>03-01</td><td>RW</td><td>SD Bus Voltage Select for VDD1
By setting these bits, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver shall check the Voltage Support bits in the Capabilities register. If an unsupported voltage is selected, the Host System shall not supply SD Bus voltage.

<table>
<tr><td>111b</td><td>3.3V (Typ.)</td></tr>
<tr><td>110b</td><td>3.0V (Typ.)</td></tr>
<tr><td>101b</td><td>1.8V (Typ.) for Embedded</td></tr>
<tr><td>100b – 000b</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW</td><td>SD Bus Power for VDD1
Before setting this bit, the SD Host Driver shall set SD Bus Voltage Select. If the Host Controller detects the No Card state, this bit shall be cleared.
If this bit is cleared, the Host Controller should immediately stop driving CMD and DAT[3:0] (tri-state), and drive SDCLK to low level (Refer to Section 2.2.15). If card is connected to Host Controller, Host Controller shall set these lines to low before stopping to supply VDD1.
In UHS-II mode, before clearing this bit, Host Driver shall clear SD Clock Enable and before stopping to supply VDD1, Host Controller shall set DAT[2] to low if DAT[2] is used as out-of band interrupt.

<table>
<tr><td>1</td><td>Power on</td></tr>
<tr><td>0</td><td>Power off</td></tr>
</table></td></tr>
</table>

Table 2-18 : Power Control Register

Implementation Note: The Host Driver has responsibility to supply SD Bus voltage by **SD Bus Power**, according to SD card OCR and supply voltage capabilities depend on the Host System. If the Host Driver selects an unsupported voltage in the **SD Bus Voltage Select** field, the Host Controller may ignore writes to SD Bus Power and keep its value at zero.

Implementation Note: The Host System shall not supply SD Bus power when **SD Bus Power** is set to 0 and can supply SD Bus power when **SD Bus Power** is set to 1 depending on the system conditions (ex. Left of the battery).

2.2.13 Block Gap Control Register (Cat.C Offset 02Ah)

D07 — D04	D03	D02	D01	D00
Rsvd	Interrupt At Block Gap	Read Wait Control	Continue Request	Stop At Block Gap Request

Figure 2-16 : Block Gap Control Register

Location	Attrib	Register Field Explanation
07-04	Rsvd	**Reserved**
03	RW	**Interrupt At Block Gap (SD Mode only)** This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. 1 : Enabled 0 : Disabled
02	RW	**Read Wait Control (SD Mode only)** The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the ***DAT***[2] line. Otherwise, the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise ***DAT*** line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. In UHS-II mode, Read Wait is disabled and ***DAT***[2] line is used for Interrupt Signal from UHS-II Card. 1 : Enable Read Wait Control 0 : Disable Read Wait Control
01	RWAC	**Continue Request** This bit is used to restart data transfer, which has been halted using the **Stop At Block Gap Request**. Setting the **Stop At Block Gap Request** to 0 and this bit to 1 restarts the data transfer. While the **Stop At Block Gap Request** is set to 1, any write to this bit is ignored. The Host Controller automatically clears this bit when the data transfer is restarted. In read operation, this bit is cleared in response to changing the **DAT Line Active** 0 to 1 (refer to Figure 3-30). In write operation, this bit is cleared in response to changing the **Write Transfer Active** 0 to 1 (refer to Figure 3-33). In all cases (Non DMA, ADMA2 and ADMA3), when the **Block Gap Event** is set to 1, data transfer is restarted by setting this bit (**Block Gap Event**=0 means data transfer is completed and continue request is not required).

<table>
<tr><td></td><td></td><td><table><tr><td>1</td><td>Restart</td></tr><tr><td>0</td><td>Not affect</td></tr></table></td></tr>
<tr><td>00</td><td>RW</td><td>Stop At Block Gap Request
This bit is used to control Stop/Continue function. Setting this bit halts data transfer at any block gap (as soon as possible) for non-DMA, SDMA and ADMA transfers. The Host Driver shall leave this bit set to 1 until the Transfer Complete is set to 1 which indicates either halt or completion of data transfer. Halting data transfer is distinguished by setting the Block Gap Event together with the Transfer Complete. Host Driver shall wait for Transfer Complete before attempting to restart the data transfer by setting the Continue Request.

Followings are notes for Stop/Continue function:
(1) While the Stop At Block Gap Request and the Continue Request are set to 0, data transfer does not restart. Simultaneous setting the Stop At Block Gap Request to 0 and the Continue Request to 1 restarts data transfer.
(2) When Host Controller version is 1.00, Stop/Continue function can be used if the card supports Read Wait Control. When Host Controller Version is 2.00 or later, Stop/Continue function can be used regardless of supporting Read Wait Control so that the Host Controller shall stop read data transfer by using Read Wait or stopping SD clock.
(3) Host Controller disables (Host Driver ignores) timeout interrupts while the Stop At Block Gap is set.
(4) In case of write transfers in which the Host Driver writes data to the Buffer Data Port register, the Host Driver shall set this bit after all block data is written. If this bit is set to 1, the Host Driver shall not write data to Buffer Data Port register.
(5) The timing of Stop/Continue function related to Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the Present State register is described in Section 3.12.3 and Section 3.12.4.
(6) Abort Transaction (Section 3.8) utilizes Stop/Continue function. Suspend/Resume (Section 3.12) utilized Stop/Continue function but Suspend/Resume function was not supported from Version 4.00.
(7) In case of UHS-II, data transfer can halt at the boundary of DATA Burst (Flow Control basis). Host Controller waits for sending Flow Control MSG until Continue Request is set to 1.
.<table><tr><td>1</td><td>Stop</td></tr><tr><td>0</td><td>Transfer</td></tr></table></td></tr>
</table>

Table 2-19 : Block Gap Control Register

2.2.14 Wakeup Control Register (Cat.C Offset 02Bh)

This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver shall maintain voltage on the SD Bus, by setting **SD Bus Power** to 1 in the *Power Control* register, when wakeup event via Card Interrupt is desired.

D07 D03	D02	D01	D00
Rsvd	Wakeup Event Enable On SD Card Removal	Wakeup Event Enable On SD Card Insertion	Wakeup Event Enable On SD Card Interrupt

Figure 2-17 : Wakeup Control Register

Location	Attrib	Register Field Explanation
07-03	Rsvd	**Reserved**
02	RW	**Wakeup Event Enable On SD Card Removal** This bit enables wakeup event via **Card Removal** assertion in the *Normal Interrupt Status* register. **FN_WUS** (Wake Up Support) in CIS does not affect this bit. 1 Enable 0 Disable
01	RW	**Wakeup Event Enable On SD Card Insertion** This bit enables wakeup event via **Card Insertion** assertion in the *Normal Interrupt Status* register. **FN_WUS** (Wake Up Support) in CIS does not affect this bit. 1 Enable 0 Disable
00	RW	**Wakeup Event Enable On Card Interrupt** This bit enables wakeup event via **Card Interrupt** assertion in the *Normal Interrupt Status* register. This bit can be set to 1 if **FN_WUS** (Wake Up Support) in CIS is set to 1. 1 Enable 0 Disable

Table 2-20 : Wakeup Control Register

2.2.15 Clock Control Register (Cat.C Offset 02Ch)

At the initialization of the Host Controller, the Host Driver shall set the **SDCLK/RCLK Frequency Select** according to the *Capabilities* register. This register controls **SDCLK** in SD Mode and **RCLK** in UHS-II mode.

D15 – D08	D07 – D06	D05	D04	D03	D02	D01	D00
SDCLK/RCLK Frequency Select	Upper Bits of SDCLK/RCLK Frequency Select	Clock Generator Select	Reserved	PLL Enable	SD Clock Enable	Internal Clock Stable	Internal Clock Enable

Figure 2-18 : Clock Control Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-08</td><td>RW</td><td>SDCLK/RCLK Frequency Select
This register is used to select the frequency of SDCLK pin. The definition of this field is dependent on the Host Controller Version.

(1) 8-bit Divided Clock Mode
This mode is supported by the Host Controller Version 1.00 and 2.00. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD Clock in the Capabilities register. Only the following settings are allowed.

<table>
<tr><td>80h</td><td>base clock divided by 256</td></tr>
<tr><td>40h</td><td>base clock divided by 128</td></tr>
<tr><td>20h</td><td>base clock divided by 64</td></tr>
<tr><td>10h</td><td>base clock divided by 32</td></tr>
<tr><td>08h</td><td>base clock divided by 16</td></tr>
<tr><td>04h</td><td>base clock divided by 8</td></tr>
<tr><td>02h</td><td>base clock divided by 4</td></tr>
<tr><td>01h</td><td>base clock divided by 2</td></tr>
<tr><td>00h</td><td>Base clock (10MHz-63MHz)</td></tr>
</table>

Setting 00h specifies the highest frequency of the SD Clock. When setting multiple bits, the most significant bit is used as the divisor but it should not be set. The three default divider values can be calculated by the frequency that is defined by the Base Clock Frequency For SD Clock in the Capabilities register.
400KHz divider value
25MHz divider value
50MHz divider value
According to the Physical Layer Specification, the maximum SD Clock frequency is 25 MHz in normal speed mode and 50MHz in high speed mode, and shall never exceed this limit.
The frequency of SDCLK is set by the following formula:

Clock Frequency = (Base Clock) / divisor</td></tr>
</table>

Thus, choose the smallest possible divisor which results in a clock frequency that is less than or equal to the target frequency.
For example, if the **Base Clock Frequency For SD Clock** in the *Capabilities* register has the value 33MHz, and the target frequency is 25MHz, then choosing the divisor value of 01h will yield 16.5MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400KHz, the divisor value of 40h yields the optimal clock value of 258KHz.

(2) 10-bit Divided Clock Mode
Host Controller Version 3.00 or later supports this mandatory mode instead of the 8-bit Divided Clock Mode. The length of divider is extended to10 bits and all divider values shall be supported.

3FFh	1/2046 Divided Clock
......	....
N	1/2N Divided Clock (Duty 50%)
......	....
002h	1/4 Divided Clock
001h	1/2 Divided Clock
000h	Base Clock (10MHz-255MHz)

(3) Programmable Clock Mode
Host Controller Version 3.00 or later supports this mode as optional. A non-zero value set to **Clock Multiplier** in the *Capabilities* register indicates support of this clock mode. The multiplier enables the Host System to select a finer grain SD clock frequency. It is not necessary to support all frequency generation specified by this field because programmable clock generator is vendor specific and dependent on the implementation. Therefore, this mode is used with *Preset Value* registers. The Host Controller vendor provides possible settings and the Host System vendor sets appropriate values to the *Preset Value* registers.

3FFh	Base Clock * M / 1024
......	....
N - 1	Base Clock * M / N
......	....
002h	Base Clock * M / 3
001h	Base Clock * M / 2
000h	Base Clock * M

This field depends on setting of **Preset Value Enable** in the *Host Control 2* register.
If **Preset Value Enable = 0,** this field is set by Host Driver.
If the **Preset Value Enable = 1,** this field is automatically set to a value specified in one of *Preset Value* registers.

07-06	ROC or RW	**Upper Bits of SDCLK/RCLK Frequency Select** Host Controller Version 1.00 and 2.00 do not support these bits and they are treated as 00b fixed value (ROC). Host Controller Version 3.00 shall support these bits to expand **SDCLK/RCLK Frequency Select** to 10-bit. Bit 07-06 is assigned to bit 09-08 of clock divider in **SDCLK/RCLK Frequency Select.**
05	RW or ROC	**Clock Generator Select** Host Controller Version 3.00 supports this bit. This bit is used to select the clock generator mode in **SDCLK/RCLK Frequency Select.** If the Programmable Clock Mode is supported (setting non-zero value to **Clock Multiplier** in the *Capabilities* register), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read. This bit depends on the setting of **Preset Value Enable** in the *Host Control 2* register. If the **Preset Value Enable = 0,** this bit is set by Host Driver. If the **Preset Value Enable = 1,** this bit is automatically set to a value specified in one of *Preset Value* registers. 1: Programmable Clock Mode 0: Divided Clock Mode
04		**Reserved**
03	RW	**PLL Enable** This bit is added from Version 4.10 for Host Controller using PLL. This feature allows Host Controller to initialize clock generator in two steps: (a) stabling input clock of PLL with **Internal Clock Enable** and (b) stabling PLL with **PLL Enable.** Host Controller can configure to minimize output latency from **SD Clock Enable**. For example, start sending symbols on D0 lane with setting **SD Clock Enable**. There are two modes to keep Host Drivers compatibility. In both modes, PLL Locked timing is indicated by **Internal Clock Stable**. (1) When **Host Version 4 Enable** =0 (Host Driver Version 3, which does not support this bit) or this bit is not implemented, **Internal Clock Enable** (or **SD Clock Enable)** may activate PLL (exit low power mode and start locking clock). (2) When **Host Version 4 Enable** =1 (Host Driver Version 4), **Internal Clock Enable** is set before setting this bit and then setting this bit may activate PLL (exit low power mode and start locking clock). 1: PLL is enabled 0: PLL is in low power mode

<table>
<tr><td>02</td><td>RW</td><td>SD Clock Enable
The Host Controller shall stop providing SDCLK or RCLK when writing this bit to 0. SDCLK/RCLK Frequency Select can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the Card Inserted in the Present State register is cleared, this bit shall be cleared.

<table><tr><td>1</td><td>Enable providing SDCLK or RCLK</td></tr><tr><td>0</td><td>Disable providing SDCLK or RCLK</td></tr></table>
(1) SD Mode
This is the case when UHS-II Interface Enable is set to 0 in the Host Control 2 register. By setting this bit to 1, SDCLK is provided on pin number 5 (CLK). Refer to Section 1.12 Controlling SDCLK.
When PLL is used to generate clock, PLL is enabled by PLL Enable (if supported) or by SD Clock Enable (if PLL Enable is not supported).
When PLL is enabled by PLL Enable, the clock synchronization is checked by Internal Clock Stable.

(2) UHS-II Mode
This is the case when UHS-II Interface Enable is set to 1 in the Host Control 2 register. By setting this bit to 1, RCLK is provided on pin number 7 and 8 (DAT0 and DAT1). Internal clock shall be stable before providing RCLK. PLL is enabled by PLL Enable and the clock synchronization is checked by Internal Clock Stable in this register. After PLL is locked, RCLK is provided to devices by setting SD Clock Enable.
If in dormant sate, writing to this register with setting this bit acts as a trigger to exit Dormant state even if this bit is already set to 1. Host Controller changes Lane State in turn: EIDL, SYN and LIDL. Refer to UHS-II IF Detection and Lane Synchronization in the Present State register for more details.
If this bit is set to 0, Host Controller drives DIF-PD on both RCLK differential lines.

If card uses DAT[2] as out-of-band interrupt and does not generate interrupt in Dormant state, card interrupt should be disabled before clearing this bit (setting IENx to 0 in CCCR).</td></tr>
</table>

<table>
<tr><td>01</td><td>ROC</td><td>Internal Clock Stable
As PLL Enable is added from Version 4.10, this status is expanded to check two cases. Host Driver Version 4.10 checks clock stability by this status twice after Internal Clock Enable is set and after PLL Enable is set. Refer to Figure 3-3.

(1) Internal Clock Stable (when PLL Enable = 0 or not supported)
This bit is set to 1 when internal clock is stable after writing to Internal Clock Enable in this register to 1.

(2) PLL Clock Stable (when PLL Enable = 1)
Host Controller that supports PLL Enable sets this status to 0 once when PLL Enable is changed 0 to 1 and then this status is set to 1 when PLL is locked. (PLL uses an internal clock in stable as a reference clock, which is enabled by Internal Clock Enable). After this bit is set to 1, Host Driver may set SD Clock Enable.

<table><tr><td>1</td><td>Ready</td></tr><tr><td>0</td><td>Not Ready</td></tr></table></td></tr>
<tr><td>00</td><td>RW</td><td>Internal Clock Enable
This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.

<table><tr><td>1</td><td>Oscillate</td></tr><tr><td>0</td><td>Stop</td></tr></table></td></tr>
</table>

Table 2-21 : Clock Control Register

2.2.16 Timeout Control Register (Cat.A Offset 02Eh)

At the initialization of the Host Controller, the Host Driver shall set the **Data Timeout Counter Value** according to the *Capabilities* register.

D07 D04	D03 D00
Rsvd	Data Timeout Counter Value

Figure 2-19 : Timeout Control Register

Location	Attrib	Register Field Explanation
07-04	Rsvd	**Reserved**
03-00	RW	**Data Timeout Counter Value** This value determines the interval by which DAT line timeouts are detected. For more information about timeout generation, refer to the **Data Timeout Error** in the *Error Interrupt Status* register. Timeout clock frequency will be generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the **Data Timeout Error Status Enable** (in the *Error Interrupt Status Enable* register) 1111b: Reserved 1110b: TMCLK x 2^{27} ....: .... 0001b: TMCLK x 2^{14} 0000b: TMCLK x 2^{13}

Table 2-22 : Timeout Control Register

Implementation Note: The Physical Layer Specification Version 3.0x defines that SDXC card may indicate 500ms busy. Then Host Driver may need to change timeout value for SDXC. It is also possible to set more than 500ms timeout regardless of card capacities.

2.2.17 Software Reset Register (Cat.C Offset 02Fh)

A reset pulse is generated when writing 1 to each bit of this register. After completing the reset, the Host Controller shall clear each bit. Because it takes some time to complete software reset, the SD Host Driver shall confirm that these bits are 0.

D07 D03	D02	D01	D00
Rsvd	Software Reset For DAT Line	Software Reset For CMD Line	Software Reset For All

Figure 2-20 : Software Reset Register

Location	Attrib	Register Field Explanation
07-03	Rsvd	**Reserved**
02	RWAC	**Software Reset For DAT Line (SD Mode only)** Only part of data circuit is reset. DMA circuit is also reset. The following registers and bits are cleared by this bit: *Buffer Data Port* register Buffer is cleared and initialized. *Present State* register **Buffer Read Enable** **Buffer Write Enable** **Read Transfer Active** **Write Transfer Active** **DAT Line Active** **Command Inhibit (DAT)** *Block Gap Control* register **Continue Request** **Stop At Block Gap Request** *Normal Interrupt Status* register **Buffer Read Ready** **Buffer Write Ready** **DMA Interrupt** **Block Gap Event** **Transfer Complete** 1: Reset 0: Work

01	RWAC	**Software Reset For CMD Line** Only part of command circuit is reset to be able to issue a command. From Version 4.10, this bit is also used to initialize UHS-II command circuit. This reset is effective only command issuing circuit (including response error statuses related to **Command Inhibit (CMD)** control) and does not affect data transfer circuit. Host Controller can continue data transfer even this reset is executed during handling of sub command response errors. The following registers and bits are cleared by this bit: *Present State* register **Command Inhibit (CMD)** *Normal Interrupt Status* register **Command Complete** *Error Interrupt Status (from Version 4.10)* Response error statuses related to **Command Inhibit (CMD)** 1 – Reset 0 – Work
00	RWAC	**Software Reset For All** This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the Host Driver shall set this bit to 1 to reset the Host Controller. The Host Controller shall reset this bit to 0 when *Capabilities* registers are valid and the Host Driver can read them. Additional use of **Software Reset For All** may not affect the value of the C*apabilities* registers. If this bit is set to 1, the Host Driver should issue reset command and reinitialize the SD card. 1 – Reset 0 – Work

Table 2-23 : Software Reset Register

2.2.18 Normal Interrupt Status Register (Cat.C Offset 030h)

The *Normal Interrupt Status Enable* affects reads of this register, but *Normal Interrupt Signal Enable* does not affect these reads. An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. Writing 1 to a bit of RW1C attribute clears it; writing 0 keeps the bit unchanged. Writing 1 to a bit of ROC attribute keeps the bit unchanged. More than one status can be cleared with a single register write. The **Card Interrupt** is cleared when the card stops asserting the interrupt; that is, when the Card Driver services the interrupt condition.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Error Interrupt	Rsvd	FX Event	Re-Tuning Event	INT_C	INT_B	INT_A	Card Interrupt	Card Removal	Card Insertion	Buffer Read Ready	Buffer Write Ready	DMA Interrupt	Block Gap Event	Transfer Complete	Command Complete

Figure 2-21 : Normal Interrupt Status Register

Location	Attrib	Register Field Explanation
15	ROC	**Error Interrupt** This status is set to 1 when any of the bits is set in the *Error Interrupt Status* register and in the *UHS-II Error Interrupt Status* register so that the Host Driver can efficiently test for an error by checking this bit first. This bit is read only. Standard Host Driver Requirements To simplify error check sequence, the Standard Host Driver should be implemented as follows: In SD mode, the Standard Host Driver sets 0 to the *UHS-II Error Interrupt Status Enable* register so that the driver may check the *Error Interrupt Status* register alone when this status (**Error Interrupt**) is set to 1. In UHS-II mode, the Standard Host Driver sets 0 to the *Error Interrupt Status Enable* register so that the driver may check the *UHS-II Error Interrupt Status* register alone when this status (**Error Interrupt**) is set to 1. 1 : Error 0 : No Error
14	Rsvd	**Reserved**
13	ROC	**FX Event** This status is added from Version 4.10. Bit06 of response data will be stored in the R[14] of the *Response* register. This interrupt may be used with response check function. In this case, this status is set when R[14] of *Response* register is set to 1 and **Response Type R1 / R5** is set to 0 in the T*ransfer Mode* register or *UHS-II Transfer Mode* register. If response check is disabled, this status is set when R[14] of *Response* register is set to 1. Host Driver needs to screen **FX Event** interrupt by checking response type is R1. 1 : FX_EVENT is detected 0 : No Event
12	ROC	**Re-Tuning Event (UHS-I only)**

<table>
<tr><td></td><td></td><td>This status is set if Re-Tuning Request in the Present State register changes from 0 to 1.
Host Controller requests Host Driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning.
In UHS-II mode, this bit is not effective.

<table><tr><td>1</td><td>Re-Tuning should be performed</td></tr><tr><td>0</td><td>Re-Tuning is not required</td></tr></table></td></tr>
<tr><td>11</td><td>ROC</td><td>INT_C (Embedded)
This status is set if INT_C is enabled and INT_C# pin is in low level. Writing this bit to 1 does not clear this bit. It is cleared by resetting the INT_C interrupt factor. Refer to the Embedded Control register.

<table><tr><td>1</td><td>INT_C is detected</td></tr><tr><td>0</td><td>No interrupt is detected</td></tr></table></td></tr>
<tr><td>10</td><td>ROC</td><td>INT_B (Embedded)
This status is set if INT_B is enabled and INT_B# pin is in low level. Writing this bit to 1 does not clear this bit. It is cleared by resetting the INT_B interrupt factor. Refer to the Embedded Control register.

<table><tr><td>1</td><td>INT_B is detected</td></tr><tr><td>0</td><td>No interrupt is detected</td></tr></table></td></tr>
<tr><td>09</td><td>ROC</td><td>INT_A (Embedded)
This status is set if INT_A is enabled and INT_A# pin is in low level. Writing this bit to 1 does not clear this bit. It is cleared by resetting the INT_A interrupt factor. Refer to the Embedded Control register.

<table><tr><td>1</td><td>INT_A is detected</td></tr><tr><td>0</td><td>No interrupt is detected</td></tr></table></td></tr>
<tr><td>08</td><td>ROC</td><td>Card Interrupt
When this status has been set and the Host Driver needs to start this interrupt service, Card Interrupt Status Enable in the Normal Interrupt Status Enable register may be set to 0 in order to clear the card interrupt statuses latched in the Host Controller and to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set Card Interrupt Status Enable to 1 and start sampling the interrupt signal again.
Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor.

(1) DAT[1] Interrupt Input in SD Mode
In 1-bit mode, the Host Controller shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. Interrupt detected by DAT[1] is supported when there is a card per slot. In case of UHS-I mode, switching time of Interrupt Period is relaxed for 2 clock cycles. Then Host Controller needs to delay start of interrupt sampling at least 2 clocks for sampling interrupt while Interrupt Period is stable.</td></tr>
</table>

		(2) DAT[2] Interrupt Input in UHS-II Mode When **Card Inserted** in the *Present State* register and **SD Bus Power for VDD1** in the *Power Control* register are set to 1, Host Controller configures DAT[2] as Interrupt Input and enables pull-up of DAT[2]. DAT[2] interrupt is asynchronous to RCLK, low level sensitive and 3.3V signal level. DAT[2] interrupt is masked by setting **Card Interrupt Status Enable** to 0 in the *Normal Interrupt* register. When either **Card Inserted** or **SD Bus Power for VDD1** is set to 0, Host Controller sets DAT[2] to low. Only point-to-point connection is allowed between Host and Card. (3) INT MSG in UHS-II Mode INT MSG is enabled by setting **INT MSG Enable** in the *UHS-II Device Select* register. DAT[2] and INT MSG interrupt sources are ORed and indicated to **Card Interrupt**. If any bit in the *UHS-II Device Interrupt Status* register is set to 1, INT MSG interrupt is generated. INT MSG interrupt is cleared by writing a correspondent bit to 1 in the *UHS-II Device Interrupt Status* register. Masking DAT[2] interrupt also disables INT MSG interrupt due to **Card Interrupt Status Enable** is set to 0. SDIO Version 4.00 does not support INT MSG. 1 : Generate Card Interrupt 0 : No Card Interrupt
07	RW1C	**Card Removal** This status is set if the **Card Inserted** in the *Present State* register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the **Card Inserted** in the *Present State* register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated. 1 : Card removed 0 : Card state stable or Debouncing
06	RW1C	**Card Insertion** This status is set if the **Card Inserted** in the *Present State* register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the **Card Inserted** in the *Present State* register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated. 1 : Card inserted 0 : Card state stable or Debouncing
05	RW1C	**Buffer Read Ready** This status is set if the **Buffer Read Enable** changes from 0 to 1. Refer to the **Buffer Read Enable** in the *Present State* register. While performing tuning procedure (**Execute Tuning** is set to 1), **Buffer Read Ready** is set to 1 for every CMD19 execution. In UHS-II mode, this bit is set at FC (Flow Control) unit basis. 1 : Ready to read buffer 0 : Not ready to read buffer

<table>
<tr><td>04</td><td>RW1C</td><td>Buffer Write Ready
This status is set if the Buffer Write Enable changes from 0 to 1. Refer to the Buffer Write Enable in the Present State register.
In UHS-II mode, this bit is set at FC (Flow Control) unit basis.
<table><tr><td>1</td><td>Ready to write buffer</td></tr><tr><td>0</td><td>Not ready to write buffer</td></tr></table></td></tr>
<tr><td>03</td><td>RW1C</td><td>DMA Interrupt
This status is set if the Host Controller detects the SDMA buffer boundary during transfer. Refer to the SDMA Buffer Boundary in the Block Size register.
Other DMA interrupt factors may be added in the future.
In case of ADMA, by setting Int field in the descriptor table, Host Controller generates this interrupt. Suppose that it is used for debugging.
This interrupt shall not be generated after the Transfer Complete.
<table><tr><td>1</td><td>DMA Interrupt is generated</td></tr><tr><td>0</td><td>No DMA Interrupt</td></tr></table></td></tr>
<tr><td>02</td><td>RW1C</td><td>Block Gap Event
This status is checked with generation of Transfer Complete interrupt by setting the Stop At Block Gap Request in the Block Gap Control register. Host determines whether the transaction is completed or can continue:
=1: The transaction is stopped on the way and can be restarted by using Continue Request in the Block Gap Control register
=0: The transaction is already completed and host is not required to set Continue Request

Timing of this status on non-DMA data transfer
(1) In the case of non-DMA Read Transaction
This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing. The Read Wait shall be supported in order to use this function. Refer to Section 3.12.3 about the detail timing.
(2) In the case of non-DMA Write Transaction
This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). Refer to Section 3.12.4 for more details on the sequence of events.

Timing of this status on DMA data transfer
This status shall be valid by generating Transfer Complete and the timing depends on bus timing and DMA implementation.
<table><tr><td>1</td><td>Transaction stopped at block gap (not completed)</td></tr><tr><td>0</td><td>No Block Gap Event</td></tr></table></td></tr>
<tr><td>01</td><td>RW1C</td><td>Transfer Complete
This bit indicates stop of transaction on three cases:
(1) Completion of data transfer
(2) Completion of a command pairing with response-with-busy (R1b, R5b)
(3) Stop of data transfer by setting Stop At Block Gap Request in the Block Gap Control register</td></tr>
</table>

		Following explanation about the timing of generating this status is for the case of non-DMA operations. SD Bus transaction timing (busy or data block) determines the timing of this status. In case of DMA operation, timing of this status depends on DMA implementation. **(1) SD Mode** (a) In the case of a Read Transaction This bit is set at the falling edge of **Read Transfer Active Status**. This interrupt is generated in two cases. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data transfer has stopped at the block gap by setting the **Stop At Block Gap Request** in the *Block Gap Control* register. Refer to Section 3.12.3 for more details as an example of non-DMA. (b) In the case of a Write Transaction This bit is set at the falling edge of the **DAT Line Active Status**. This interrupt is generated in two cases. The first is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting **Stop At Block Gap Request**. Refer to Section 3.12.4 for more details as an example of non-DMA. (c) In the case of a command pairing with response-with-busy This bit is set when busy is de-asserted. Refer to **DAT Line Active** and **Command Inhibit (DAT)** in the *Present State* register. (d) In UHS-I mode While performing tuning procedure (**Execute Tuning** is set to 1), **Transfer Complete** is not set to 1**.** (2) **UHS-II Mode** This interrupt is generated in following two cases: (a) EBSY Completion (for EBSY supported commands) When **EBSY Wait** in the *UHS-II Transfer Mode* register is set to 1, this bit is set when EBSY packet has been received, and all valid data have been sent to system memory in case of read operation. (b) Stop/Continue during DCMD Data Transfer When **Stop At Block Gap Request** in the *Block Gap Control* register is set to 1 and data transfer is stopped at the Flow Control. Following is for both SD mode and UHS-II mode. The table below shows that **Transfer Complete** has higher priority than **Data Timeout Error**. If both bits are set to 1 together, suppose execution of a command is completed. **Relation between Transfer Complete and Data Timeout Error** <table><tr><th>Transfer Complete</th><th>Data Timeout Error</th><th>Meaning of the status</th></tr><tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr><tr><td>0</td><td>1</td><td>Timeout occur during transfer</td></tr><tr><td>1</td><td>Don't Care</td><td>Command Execution complete</td></tr></table> <table><tr><td>1</td><td>Command execution is completed</td></tr><tr><td>0</td><td>Not complete</td></tr></table>

<table>
<tr><td>00</td><td>RW1C</td><td>Command Complete
(1) SD Mode
This bit is set when get the end bit of a response except the case of Auto CMD12 and Auto CMD23. Command Complete is not generated by the response of CMD12 or CMD23 but generated by the response of a read/write command.
Refer to Command Inhibit (CMD) in the Present State register for how to control this bit.

The table below shows that Command Timeout Error has higher priority than Command Complete. If both bits are set to 1, it can be considered that the response was not received correctly.

<table>
<tr><th>Command Complete</th><th>Command Timeout Error</th><th>Meaning of the status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>Don't Care</td><td>1</td><td>Response not received within 64 SDCLK cycles.</td></tr>
<tr><td>1</td><td>0</td><td>Response received</td></tr>
</table>

Version 4.00 defines response check function for R1 and R5. If Response Interrupt Disable in the Transfer Mode register is set to 1, generation of this interrupt is prohibited regardless of Command Complete Signal Enable.

(2) UHS-II Mode
If Response Interrupt Disable is set to 0 in the UHS-II Transfer Mode register, this interrupt is generated when response packet is received.
If Response Interrupt Disable is set to 1 in the UHS-II Transfer Mode register, generation of this interrupt is prohibited regardless of Command Complete Signal Enable.

<table>
<tr><td>1</td><td>Command complete</td></tr>
<tr><td>0</td><td>No command complete</td></tr>
</table>
</td></tr>
</table>

Table 2-24 : Normal Interrupt Status Register

2.2.19 Error Interrupt Status Register (Cat.C Offset 032h)

Signals defined in this register can be enabled by the *Error Interrupt Status Enable* register, but not by the *Error Interrupt Signal Enable* register. The interrupt is generated when the *Error Interrupt Signal Enable* is enabled and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 keeps the bit unchanged. More than one status can be cleared at the one register write.

D15 D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error Status	Response Error	Tuning Error	ADMA Error	Auto CMD Error	Current limit Error	Data End Bit Error	Data CRC Error	Data Timeout Error	Command Index Error	Command End Bit Error	Command CRC Error	Command Timeout Error

Figure 2-22 : Error Interrupt Status Register

Location	Attrib	Register Field Explanation
15-12	RW1C	**Vendor Specific Error Status** Additional status bits can be defined in this register by the vendor.
11	RW1C	**Response Error (SD Mode only)** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If **Response Error Check Enable** is set to 1 in the *Transfer Mode* register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1. 1: Error 0: No Error
10	RW1C	**Tuning Error (UHS-I only)** This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure (Occurrence of an error during tuning procedure is indicated by **Sampling Clock Select** in the *Host Control 2* register). By detecting **Tuning Error**, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, **Sampling Clock Select** shall be set to 0 before executing tuning procedure (Refer to Figure 2-29). The **Tuning Error** is higher priority than the other error interrupts generated during data transfer. By detecting **Tuning Error**, the Host Driver should discard data transferred by a current read/write command and retry data transfer after the Host Controller retrieved from tuning circuit error. 1: Error 0: No Error

<table>
<tr><td>09</td><td>RW1C</td><td>ADMA Error
This bit is set when the Host Controller detects errors during ADMA based data transfer in SD mode and UHS-II mode. The state of the ADMA at an error occurrence is saved in the ADMA Error Status register.
Host Driver can obtain information of ADMA error from the ADMA System Address register and the ADMA Error Status register.

In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.

<table><tr><td>1</td><td>Error</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>08</td><td>RW1C</td><td>Auto CMD Error (SD Mode only)
Auto CMD12 and Auto CMD23 use this error status. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared with clearing of this bit (another implementation is also allowed).

<table><tr><td>1</td><td>Error</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>07</td><td>RW1C</td><td>Current Limit Error
By setting the SD Bus Power bit in the Power Control register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the Host Controller is not supplying power to SD card due to some failure. Reading 0 means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. If the Host Controller does not support this function, this bit shall always be set to 0.

Because this register may not be referred during UHS-II data transfer, the Host Driver should check this status during UHS-II Card initialization. Refer to Standard Host Driver Requirements, which is described in the Error Interrupt of the Normal Interrupt Status register.

<table><tr><td>1</td><td>Power fail</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>06</td><td>RW1C</td><td>Data End Bit Error (SD Mode only)
This bit is set to 1 when detecting 0 at the end bit position on the DAT line: either read data or the CRC Status.

<table><tr><td>1</td><td>Error</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
</table>

<table>
<tr><td>05</td><td>RW1C</td><td>Data CRC Error (SD Mode only)
This bit is set to 1 when detecting a CRC error in data transfer through the DAT line by checking CRC data transferred with read data or by detecting the Write CRC status having a value of other than "010".

<table><tr><td>1</td><td>Error</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>04</td><td>RW1C</td><td>Data Timeout Error (SD Mode only)
This bit is set when detecting one of following timeout conditions.
(1) Busy timeout for R1b,R5b type
(2) Busy timeout after Write CRC status
(3) Write CRC Status timeout
(4) Read Data timeout.

<table><tr><td>1</td><td>Time out</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>03</td><td>RW1C</td><td>Command Index Error (SD Mode only)
This bit is set if a Command Index error occurs in the command response.

<table><tr><td>1</td><td>Error</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>02</td><td>RW1C</td><td>Command End Bit Error (SD Mode only)
This bit is set when detecting that the end bit of a command response is 0.

<table><tr><td>1</td><td>End Bit Error Generated</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>01</td><td>RW1C</td><td>Command CRC Error (SD Mode only)
Command CRC Error is generated in two cases.
If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response.
The Host Controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the Host Controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SD clock edge, then the Host Controller shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict (Refer to Table 2-26).

<table><tr><td>1</td><td>CRC Error Generated.</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
<tr><td>00</td><td>RW1C</td><td>Command Timeout Error (SD Mode only)
This bit is set only if no response is returned within 64 SD clock cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, in which case Command CRC Error shall also be set as shown in Table 2-26, this bit shall be set without waiting for 64 SD clock cycles because the command will be aborted by the Host Controller.

<table><tr><td>1</td><td>Time out</td></tr><tr><td>0</td><td>No Error</td></tr></table></td></tr>
</table>

Table 2-25 : Error Interrupt Status Register

The relation between **Command CRC Error** and **Command Timeout Error** is shown in Table 2-26.

Command CRC Error	**Command Timeout Error**	**Kinds of error**
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	***CMD*** line conflict

Table 2-26 : The Relation between Command CRC Error and Command Timeout Error

2.2.20 Normal Interrupt Status Enable Register (Cat.C Offset 034h)

Setting to 1 enables Interrupt Status.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Fixed to 0	Rsvd	FX Event Status Enable	Re-Tuning Event Status Enable	INT_C Status Enable	INT_B Status Enable	INT_A Status Enable	Card Interrupt Status Enable	Card Removal Status Enable	Card Insertion Status Enable	Buffer Read Ready Status Enable	Buffer Write Ready Status Enable	DMA Interrupt Status Enable	Block Gap Event Status Enable	Transfer Complete Status Enable	Command Complete Status Enable

Figure 2-23 : Normal Interrupt Status Enable Register

Location	Attrib	Register Field Explanation
15	RO	**Fixed to 0** The Host Driver shall control error interrupts using the *Error Interrupt Status Enable* register.
14	Rsvd	**Reserved**
13	RW	**FX Event Status Enable** This bit is added from Version 4.10. 1 Enabled 0 Masked
12	RW	**Re-Tuning Event Status Enable (UHS-I only)** 1 Enabled 0 Masked
11	RW	**INT_C Status Enable (Embedded)** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System. The Host Driver may clear this bit before servicing the **INT_C** and may set this bit again after all interrupt requests to INT_C pin are cleared to prevent inadvertent interrupts. 1 Enabled 0 Masked
10	RW	**INT_B Status Enable (Embedded)** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System. The Host Driver may clear this bit before servicing the **INT_B** and may set this bit again after all interrupt requests to INT_B pin are cleared to prevent inadvertent interrupts. 1 Enabled 0 Masked
09	RW	**INT_A Status Enable (Embedded)** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System. The Host Driver may clear this bit before servicing the **INT_A** and may set this bit again after all interrupt requests to INT_A pin are cleared to prevent inadvertent interrupts. 1 Enabled 0 Masked

08	RW	**Card Interrupt Status Enable** If this bit is set to 0, the Host Controller shall clear interrupt request to the System. The **Card Interrupt** detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver may clear the **Card Interrupt Status Enable** before servicing the **Card Interrupt** and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. By setting this bit to 0, interrupt input should be masked by implementation so that the interrupt Input is not affected by external signal in any state (ex. floating). 1 – Enabled 0 – Masked
07	RW	**Card Removal Status Enable** 1 – Enabled 0 – Masked
06	RW	**Card Insertion Status Enable** 1 – Enabled 0 – Masked
05	RW	**Buffer Read Ready Status Enable** 1 – Enabled 0 – Masked
04	RW	**Buffer Write Ready Status Enable** 1 – Enabled 0 – Masked
03	RW	**DMA Interrupt Status Enable** 1 – Enabled 0 – Masked
02	RW	**Block Gap Event Status Enable** 1 – Enabled 0 – Masked
01	RW	**Transfer Complete Status Enable** 1 – Enabled 0 – Masked
00	RW	**Command Complete Status Enable** 1 – Enabled 0 – Masked

Table 2-27 : Normal Interrupt Status Enable Register

Implementation Note:
The Host Controller may sample the card interrupt signal during interrupt period and may hold its value in the flip-flop. If the **Card Interrupt Status Enable** is set to 0, the Host Controller shall clear all internal signals regarding Card Interrupt.

2.2.21 Error Interrupt Status Enable Register (Cat.C Offset 036h)

Setting to 1 enables Interrupt Status.

D15 - D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error Status Enable	Response Error Status Enable	Tuning Error Status Enable	ADMA Error Status Enable	Auto CMD Error Status Enable	Current Limit Error Status Enable	Data End Bit Error Status Enable	Data CRC Error Status Enable	Data Timeout Error Status Enable	Command Index Error Status Enable	Command End Bit Error Status Enable	Command CRC Error Status Enable	Command Timeout Error Status Enable

Figure 2-24 : Error Interrupt Status Enable Register

Location	Attrib	Register Field Explanation
15-12	RW	**Vendor Specific Error Status Enable** 1: Enabled 0: Masked
11	RW	**Response Error Status Enable (SD Mode only)** 1: Enabled 0: Masked
10	RW	**Tuning Error Status Enable (UHS-I only)** 1: Enabled 0: Masked
09	RW	**ADMA Error Status Enable** 1: Enabled 0: Masked
08	RW	**Auto CMD Error Status Enable (SD Mode only)** 1: Enabled 0: Masked
07	RW	**Current Limit Error Status Enable** 1: Enabled 0: Masked
06	RW	**Data End Bit Error Status Enable (SD Mode only)** 1: Enabled 0: Masked
05	RW	**Data CRC Error Status Enable (SD Mode only)** 1: Enabled 0: Masked
04	RW	**Data Timeout Error Status Enable (SD Mode only)** 1: Enabled 0: Masked

03	RW	**Command Index Error Status Enable (SD Mode only)** 1 Enabled 0 Masked
02	RW	**Command End Bit Error Status Enable (SD Mode only)** 1 Enabled 0 Masked
01	RW	**Command CRC Error Status Enable (SD Mode only)** 1 Enabled 0 Masked
00	RW	**Command Timeout Error Status Enable (SD Mode only)** 1 Enabled 0 Masked

Table 2-28 : Error Interrupt Status Enable Register

Implementation Note: To detect CMD line conflict, the Host Driver must set both **Command Timeout Error Status Enable** and **Command CRC Error Status Enable** to 1.

2.2.22 Normal Interrupt Signal Enable Register (Cat.C Offset 038h)

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all may share the same 1-bit interrupt line. Setting 1 to any bit of this register enables interrupt generation.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Fixed to 0	Rsvd	FX Event Signal Enable	Re-Tuning Event Signal Enable	INT_C Signal Enable	INT_B Signal Enable	INT_A Signal Enable	Card Interrupt Signal Enable	Card Removal Signal Enable	Card Insertion Signal Enable	Buffer Read Ready Signal Enable	Buffer Write Ready Signal Enable	DMA Interrupt Signal Enable	Block Gap Event Signal Enable	Transfer Complete Signal Enable	Command Complete Signal Enable

Figure 2-25 : Normal Interrupt Signal Enable Register

Location	Attrib	Register Field Explanation
15	RO	**Fixed to 0** The Host Driver shall control error interrupts using the *Error Interrupt Signal Enable* register.
14	Rsvd	**Reserved**
13	RW	**FX Event Signal Enable** This bit is added from Version 4.10. 1 Enabled 0 Masked
12	RW	**Re-Tuning Event Signal Enable (UHS-I only)**
11	RW	**INT_C Signal Enable (Embedded)** 1 Enabled 0 Masked
10	RW	**INT_B Signal Enable (Embedded)** 1 Enabled 0 Masked
09	RW	**INT_A Signal Enable (Embedded)** 1 Enabled 0 Masked
08	RW	**Card Interrupt Signal Enable** 1 Enabled 0 Masked
07	RW	**Card Removal Signal Enable** 1 Enabled 0 Masked

06	RW	**Card Insertion Signal Enable** 1 Enabled 0 Masked
05	RW	**Buffer Read Ready Signal Enable** 1 Enabled 0 Masked
04	RW	**Buffer Write Ready Signal Enable** 1 Enabled 0 Masked
03	RW	**DMA Interrupt Signal Enable** 1 Enabled 0 Masked
02	RW	**Block Gap Event Signal Enable** 1 Enabled 0 Masked
01	RW	**Transfer Complete Signal Enable** 1 Enabled 0 Masked
00	RW	**Command Complete Signal Enable** 1 Enabled 0 Masked

Table 2-29 : Normal Interrupt Signal Enable Register

2.2.23 Error Interrupt Signal Enable Register (Cat.C Offset 03Ah)

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits all share the same 1-bit interrupt line. Setting 1 to any bit of this register enables interrupt generation.

D15 - D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error Signal	Response Error Signal Enable	Tuning Error Signal Enable	ADMA Error Signal Enable	Auto CMD Error Signal Enable	Current Limit Error Signal Enable	Data End Bit Error Signal Enable	Data CRC Error Signal Enable	Data Timeout Error Signal Enable	Command Index Error Signal Enable	Command End Bit Error Signal Enable	Command CRC Error Signal Enable	Command Timeout Error Signal Enable

Figure 2-26 : Error Interrupt Signal Enable Register

Location	Attrib	Register Field Explanation
15-12	RW	**Vendor Specific Error Signal Enable** 1: Enabled 0: Masked
11	RW	**Response Error Signal Enable (SD Mode only)** 1: Enabled 0: Masked
10	RW	**Tuning Error Signal Enable (UHS-I only)** 1: Enabled 0: Masked
09	RW	**ADMA Error Signal Enable** 1: Enabled 0: Masked
08	RW	**Auto CMD Error Signal Enable (SD Mode only)** 1: Enabled 0: Masked
07	RW	**Current Limit Error Signal Enable** 1: Enabled 0: Masked
06	RW	**Data End Bit Error Signal Enable (SD Mode only)** 1: Enabled 0: Masked
05	RW	**Data CRC Error Signal Enable (SD Mode only)** 1: Enabled 0: Masked

04	RW	**Data Timeout Error Signal Enable (SD Mode only)** 1 Enabled 0 Masked
03	RW	**Command Index Error Signal Enable (SD Mode only)** 1 Enabled 0 Masked
02	RW	**Command End Bit Error Signal Enable (SD Mode only)** 1 Enabled 0 Masked
01	RW	**Command CRC Error Signal Enable (SD Mode only)** 1 Enabled 0 Masked
00	RW	**Command Timeout Error Signal Enable (SD Mode only)** 1 Enabled 0 Masked

Table 2-30 : Error Interrupt Signal Enable Register

2.2.24 Auto CMD Error Status Register (Cat.A Offset 03Ch)

This register is used to indicate CMD12 response error of Auto CMD12 and CMD23 response error of Auto CMD23. The Host driver can determine what kind of Auto CMD12 / CMD23 errors occur by this register. Auto CMD23 errors are indicated in bit 04-01.This register is valid only when the **Auto CMD Error** is set.

D15-D08	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd	Command Not Issued by Auto CMD12 Error	Rsvd	Auto CMD Response Error	Auto CMD Index Error	Auto CMD End Bit Error	Auto CMD CRC Error	Auto CMD Timeout Error	Auto CMD12 not executed

Figure 2-27 : Auto CMD Error Status Register

Location	Attrib	Register Field Explanation
15-08	Rsvd	**Reserved**
07	ROC	**Command Not Issued By Auto CMD12 Error** Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. This bit is set to 0 when **Auto CMD Error** is generated by Auto CMD23. 1: Not Issued 0: No error
06	Rsvd	**Reserved**
05	ROC	**Auto CMD Response Error** This bit is set when **Response Error Check Enable** in the *Transfer Mode* register is set to 1 and an error is detected in R1 response of either Auto CMD12 or Auto CMD23. This status should be ignored if any bit of D00 to D04 is set to 1. 1: Error 0: No error
04	ROC	**Auto CMD Index Error** This bit is set if the Command Index error occurs in response to a command. 1: Error 0: No error
03	ROC	**Auto CMD End Bit Error** This bit is set when detecting that the end bit of command response is 0. 1: End Bit Error Generated 0: No error
02	ROC	**Auto CMD CRC Error** This bit is set when detecting a CRC error in the command response. 1: CRC Error Generated 0: No error

<table>
<tr><td>01</td><td>ROC</td><td>Auto CMD Timeout Error
This bit is set if no response is returned within 64 SDCLK cycles from the end bit of command.
If this bit is set to1, the other error status bits (D04-D02) are meaningless.

<table><tr><td>1</td><td>Time out</td></tr><tr><td>0</td><td>No error</td></tr></table></td></tr>
<tr><td>00</td><td>ROC</td><td>Auto CMD12 Not Executed
If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, error status bits (D04-D01) are meaningless.
This bit is set to 0 when Auto CMD Error is generated by Auto CMD23.

<table><tr><td>1</td><td>Not executed</td></tr><tr><td>0</td><td>Executed</td></tr></table></td></tr>
</table>

Table 2-31 : Auto CMD Error Status Register

The relation between Auto CMD CRC Error and Auto CMD Timeout Error is shown in Table 2-32.

Auto CMD CRC Error	Auto CMD Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Table 2-32 : The Relation between CRC Error and Timeout Error for Auto CMD

The timing of changing *Auto CMD Error Status* can be classified in three scenarios:

(1) When the Host Controller is going to issue Auto CMD12
Set D00 to 1 if Auto CMD12 cannot be issued due to an error in the previous command.
Set D00 to 0 if Auto CMD12 is issued.
(2) At the end bit of an Auto CMD12 response
Check received responses by checking the error bits D01, D02, D03 and D04.
Set to 1 if error is detected.
Set to 0 if error is not detected.
(3) Before reading the Auto CMD Error Status bit D07
Set D07 to 1 if there is a command cannot be issued
Set D07 to 0 if there is no command to issue

Timing of generating the **Auto CMD Error** and writing to the *Command* register are asynchronous. Then D07 shall be sampled when driver never writing to the *Command* register. So just before reading the *Auto CMD Error Status* register is good timing to set the D07 status bit.

An **Auto CMD Error** Interrupt is generated when one of the error bits D00 to D05 is set to 1. The **Command Not Issued By Auto CMD12 Error** does not make any effect on interrupt because it is set when any bit of D01 to D04 is set to 1.

2.2.25 Host Control 2 Register (Cat.C Offset 03Eh)

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05-04	D03	D02-D00
Preset Value Enable	Asynchronous Interrupt Enable	64-bit Addressing	Host Version 4 Enable	CMD23 Enable	ADMA2 Length Mode	Reserved	UHS-II Interface Enable	Sampling Clock Select	Execute Tuning	Driver Strength Select	1.8V Signaling Enable	UHS Mode Select

Figure 2-28 : Host Control 2 Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15</td><td>RW</td><td>Preset Value Enable
Host Controller Version 3.00 supports this bit.
As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set, automatic SDCLK frequency generation and driver strength selection is performed without considering system specific conditions. This bit enables the functions defined in the Preset Value registers.

<table><tr><td>1</td><td>Automatic Selection by Preset Value are Enabled</td></tr><tr><td>0</td><td>SDCLK and Driver Strength are controlled by Host Driver</td></tr></table>
If this bit is set to 0, SDCLK/RCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Driver.
If this bit is set to 1, SDCLK/RCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Controller as specified in the Preset Value registers.</td></tr>
<tr><td>14</td><td>RW</td><td>Asynchronous Interrupt Enable
This bit can be set to 1 if a card supports asynchronous interrupts and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode (and zero is set to Interrupt Pin Select in the Embedded Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver the Card Interrupt to the host when it is asserted by the Card.

<table><tr><td>1</td><td>Enabled</td></tr><tr><td>0</td><td>Disabled</td></tr></table></td></tr>
</table>

<table>
<tr><td>13</td><td>RW</td><td>64-bit Addressing
This field is effective when Host Version 4 Enable is set to 1.
Host Controller selects either of 32-bit or 64-bit addressing modes to access system memory. OS installed in a host system determines which addressing mode is used either 32-bit or 64-bit. Host Driver sets this bit depends on addressing mode of installed OS. Refer to 64-bit System Address Support in the Capabilities register.

1 | 64 bits addressing
0 | 32 bits addressing</td></tr>
<tr><td>12</td><td>RW</td><td>Host Version 4 Enable
This bit selects either Version 3.00 compatible mode or Version 4 mode.
In Version 4.00, support of 64-bit System Addressing is modified. All DMAs support 64-bit System Addressing. UHS-II supported Host Driver shall enable this bit.
Version 4.10 supports 32-bit Block Count for all operations.

Functions of following fields are modified.
(1) SDMA Address
SDMA uses ADMA System Address register (05Fh-058h) instead of SDMA System Address register (Offset 003-000h)
(2) ADMA2 / ADMA3 Selection
ADMA3 is selected by DMA Select in the Host Control 1 register.
(3) 64-bit ADMA Descriptor Size
128-bit descriptor is used instead of 96-bit descriptor when 64-bit Addressing is set to 1.
(4) Selection of 32-bit / 64-bit System Addressing
Either 32-bit or 64-bit system addressing is selected by 64-bit Addressing bit in this register instead of DMA Select in the Host Control 1 register.
(5) 32-bit Block Count
SDMA System Address register (003h-000h) is modified to 32-bit Block Count register.

1 | Version 4 Mode
0 | Version 3.00 Compatible Mode</td></tr>
<tr><td>11</td><td>RW</td><td>CMD23 Enable
In memory card initialization, Host Driver Version 4.10 checks whether card supports CMD23 by checking a bit SCR[33]. If the card supports CMD23 (SCR[33]=1), this bit is set to 1. This bit is used to select Auto CMD23 or Auto CMD12 for ADMA3 data transfer. Refer to Auto CMD Enable in the Transfer Mode register.</td></tr>
<tr><td>10</td><td>RW</td><td>ADMA2 Length Mode
This bit selects one of ADMA2 Length Modes either 16-bit or 26-bit.

1 | 26-bit Data Length Mode
0 | 16-bit Data Length Mode</td></tr>
<tr><td>09</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

08	RW	**UHS-II Interface Enable** This bit is used to enable UHS-II Interface. Before trying to start UHS-II initialization, this bit shall be set to 1. Before trying to start SD mode initialization, this bit shall be set to 0. This bit is used to enable **UHS-II IF Detection**, **Lane Synchronization** and **In Dormant State** in the *Present State* register, and to select clock source of either SD mode or UHS-II mode. Host Controller shall not leave unused SD 4-bit Interface lines (CLK, CMD and DAT[3:2]) floating in UHS-II mode by using pull-up or driving to low. When DAT[2] is used as interrupt input in UHS-II mode, DAT[2] of Host Controller is set to input and then DAT[2] of SDIO card is set to output to avoid conflict. 1: UHS-II Interface Enabled 0: 4-bit SD Interface Enabled
07	RW	**Sampling Clock Select (UHS-I only)** This bit controls a tuning procedure (refer to Figure 2-29) and indicates a kind of clocks for sampling **CMD** and **DAT**: a fixed clock (as default) or a tuned clock. While tuning is not executing (**Execute Tuning**=0), writing with clearing this bit forces selecting the fixed clock and resets a tuning circuit. On starting the tuning procedure when **Execute Tuning** is set from 0 to 1, this bit controls behavior of the tuning circuit: =0: Reset the tuning circuit at the start of tuning and it will take time to complete tuning due to the first time tuning from a reset state =1: Use of previous tuning result enables the tuning circuit to complete Re-Tuning in a short time when tuned clock has been selected During **Execute Tuning**=1, reading of this bit is meaningless. Reading of this bit during **Execute Tuning**=0 includes two meanings; which sampling clock has been selected and whether tuning has succeeded or failed: =1: The tuned clock is being used for sampling as tuning has been completed successfully. Re-Tuning is possible in this case. =0: The fixed clock is being used for sampling. If after the completion of tuning, it indicates that tuning has failed. Change of this bit is not allowed while the Host Controller is receiving response or a read data block. 1: During **Execute Tuning**=0: Writing 1 does not affect the tuning circuit Reading 1 means the tuned clock is used to sample data When setting **Execute Tuning** 0 to 1: Keeping 1 for Re-Tuning 0: During **Execute Tuning**=0: Writing 0 resets the tuning circuit Reading 0 means a fixed clock is used to sample data or failure of tuning When setting **Execute Tuning** 0 to 1: Writing 0 for the First Time Ttuning

<table>
<tr><td>06</td><td>RWAC</td><td>Execute Tuning (UHS-I only)
This bit is set to 1 to start the tuning procedure (refer to Figure 2-29) and is automatically cleared by tuning completion. With this bit, Sampling Clock Select is used to control tuning method and to indicate the result of tuning. Execution of the tuning procedure can be aborted by writing this bit to 0.

<table><tr><td>1</td><td>Execute Tuning</td></tr><tr><td>0</td><td>Not Tuned or Tuning Completed</td></tr></table></td></tr>
<tr><td>05-04</td><td>RW</td><td>Driver Strength Select (UHS-I only)
Host Controller output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set depends on Driver Type A, C and D support bits in the Capabilities register.

This bit depends on setting of Preset Value Enable.
If Preset Value Enable = 0, this field is set by Host Driver.
If Preset Value Enable = 1, this field is automatically set by a value specified in the one of Preset Value registers.

<table><tr><td>00b</td><td>Driver Type B is Selected (Default)</td></tr><tr><td>01b</td><td>Driver Type A is Selected</td></tr><tr><td>10b</td><td>Driver Type C is Selected</td></tr><tr><td>11b</td><td>Driver Type D is Selected</td></tr></table></td></tr>
<tr><td>03</td><td>RW</td><td>1.8V Signaling Enable (UHS-I only)
This bit controls voltage regulator for I/O cell. 3.3V is supplied to the card regardless of signaling voltage.
Setting this bit (from 0 to 1) starts changing signal voltage from 3.3V to 1.8V. Host Controller clears this bit if switching to 1.8V signaling fails, and should try to use the card in 3.3V signaling.
Clearing this bit (from 1 to 0) starts changing signal voltage from 1.8V to 3.3V. Refer to Host Regulator Voltage Stable in the Present State register how to detect stability of host regulator voltage.

Host Driver can set this bit to 1 when Host Controller supports 1.8V signaling (One of support bits is set to 1: SDR50, SDR104 or DDR50 in the Capabilities register) and the card or device supports UHS-I (S18A=1. Refer to Bus Signal Voltage Switch Sequence in the Physical Layer Specification Version 3.0x).

<table><tr><td>1</td><td>1.8V Signaling</td></tr><tr><td>0</td><td>3.3V Signaling</td></tr></table></td></tr>
</table>

<table>
<tr><td>02-00</td><td>RW</td><td>UHS Mode Select
This field is used to select one of UHS-I modes or UHS-II mode. In case of UHS-I mode, this field is effective when 1.8V Signaling Enable is set to 1. In case of UHS-II mode, 1.8V Signaling Enable shall be set to 0. Setting of this field selects one of preset values in UHS-I mode or UHS-II mode.

If Preset Value Enable in the Host Control 2 register is set to 1, Host Controller sets SDCLK/RCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select according to Preset Value registers. In this case, one of preset value registers is selected by this field. Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitch. After setting this field, Host Driver sets SD Clock Enable again.

<table><tr><td>000b</td><td>SDR12</td></tr><tr><td>001b</td><td>SDR25</td></tr><tr><td>010b</td><td>SDR50</td></tr><tr><td>011b</td><td>SDR104</td></tr><tr><td>100b</td><td>DDR50</td></tr><tr><td>101b</td><td>Reserved</td></tr><tr><td>110b</td><td>Reserved</td></tr><tr><td>111b</td><td>UHS-II</td></tr></table>
When SDR50, SDR104 or DDR50 is selected for SDIO card, interrupt detection at the block gap shall not be used. Read Wait timing is changed for these modes. Refer to the SDIO Specification Version 3.00 for more detail.</td></tr>
</table>

Table 2-33 : Host Control 2 Register

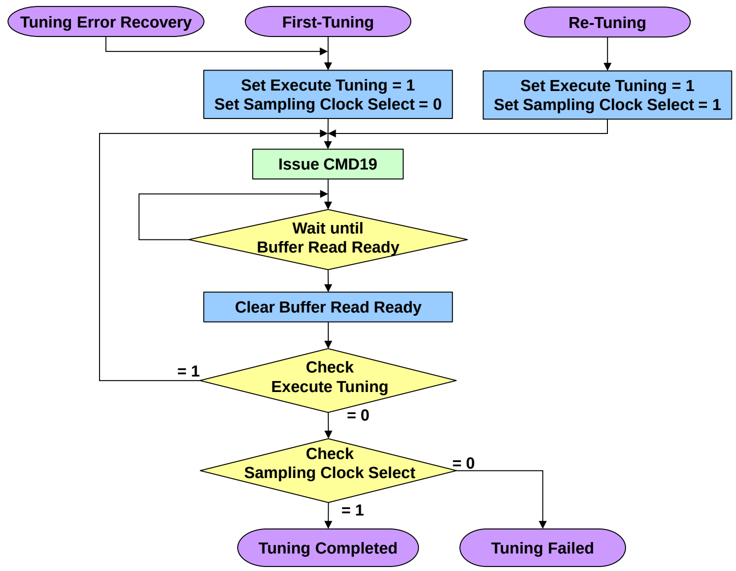


Figure 2-29 : Sampling Clock Tuning Procedure (UHS-I only)

Figure 2-29 shows the tuning procedure to adjust the sampling clock. In the default, lower frequency operation, a fixed sampling clock is used to receive signals on ***CMD*** and ***DAT[3:0]***. Before using the SDR104 or SDR50 (if **Use Tuning for SDR50** is set to 1 in the *Capabilities* register) modes, the Host Driver shall execute the tuning procedure at the initialization sequence regardless of **Re-Tuning Modes** state in the *Capabilities* register.
The Host Driver requests the Host Controller to start the tuning sequence by setting **Execute Tuning** to 1. The Host Driver issues CMD19 repeatedly until Host Controller resets **Execute Tuning** to 0. The Host Controller then resets **Execute Tuning** when the tuning is completed or the tuning fails. The Host Driver may abort this loop by writing 0 to **Execute Tuning** register when CMD19 tuning timeout (default 150ms) occurs. In this case, a fixed sampling clock should be used (**Sampling Clock Select** = 0).
The **Sampling Clock Select** is valid after **Execute Tuning** has changed from 1 to 0. Setting **Sampling Clock Select** to 1 indicates that the tuning procedure has completed successfully. Setting **Sampling Clock Select** to 0 indicates that the tuning procedure has failed.
While the tuning sequence is being performed, the Host Controller does not generate interrupts (including **Command Complete**) except **Buffer Read Ready** and CMD19 response errors are not indicated.

Writing **Sampling Clock Select** to 0 forces the Host Controller to use a fixed sampling clock and resets the tuning circuit of the Host Controller. If tuning is started after the reset of tuning circuit (First-Tuning entry in Figure 2-29), it will take time to complete tuning sequence. It is possible to execute Re-Tuning (Re-Tuning entry in Figure 2-29) when tuned clock has been used (**Sampling Clock Select**=1), that utilizes previous tuning result by keeping **Sampling Clock Select** to 1 at the start of re-tuning to shorten re-tuning time than the first tuning time.
When receiving **Tuning Error** interrupt, the Host Driver needs to reset the tuning circuit by clearing the **Sampling Clock Select** to 0 and then execute the tuning procedure (Tuning Error Recovery entry in Figure 2-29 that is equivalent to the First-Tuning).

The re-tuning timing is specified by two methods: **Re-Tuning Request** generated by Host Controller and expiration of a re-tuning timer prepared by Host Driver. When Host Driver receives either interrupts, the tuning procedure defined by Figure 2-29 is inserted before issuing any command. Refer to **Re-Tuning Request** in the *Present State* register and **Re-Tuning Modes** in the *Capabilities* register for more detail.

Implementation Note for Host Driver:

(1) Tuning timeout management
Host Driver may change a tuning timeout value depends on the Host controller implementation but the initilal value of tunung timeout should be set to 150ms. If the tuning timeout happens, the Host Driver may increase the tunig timeout value.

(2) Reloading the timeout value after the first CMD19 execution
During tuning period, Host Controller issues CMD19 repeatedly. The execution time in SD card side may be different between the first CMD19 and other following CMD19s. Some SD cards may take longer time (e.g. 100ms) to execute the first CMD19 because SD cards need to prepare the data to be returned. Considering such behavior, the host driver may reload the timeout value after the first CMD19 execution to reduce the possibility of tuning timeout occurrence.

2.2.26 Capabilities Register (Cat.C Offset 040h)

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to **Software Reset For All** in the *Software Reset* register for loading from flash memory and completion timing control.

D63-D62	D61	D60	D59	D58-D56	D55-D48	D47-D46	D45	D44	D43-D40	D39	D38	D37	D36	D35	D34	D33	D32
Rsvd	Rsvd for future VDD2	1.8V VDD2 Support	ADMA3 Support	Rsvd	Clock Multiplier	Re-Tuning Modes	Use Tuning for SDR50	Rsvd	Timer Count for Re-Tuning	Rsvd	Driver Type D Support	Driver Type C Support	Driver Type A Support	UHS-II Support	DDR50 Support	SDR104 Support	SDR50 Support

D31 D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17-16	D15-D08	D07	D06	D05-D00
Slot Type	Asynchronous Interrupt Support	64-bit System Address Support for V3	64-bit System Address Support for V4	Voltage Support 1.8V	Voltage Support 3.0V	Voltage Support 3.3V	Suspend/Resume Support	SDMA Support	High Speed Support	Rsvd	ADMA2 Support	8-bit Support for Embedded Device	Max Block Length	Base Clock Frequency For SD Clock	Timeout Clock Unit	Rsvd	Timeout Clock Frequency

Figure 2-30 : Capabilities Register

Location	Attrib	Register Field Explanation
63-62	Rsvd	**Reserved**
61	HwInit	**Reserved for future VDD2** Set this bit to 0.
60	HwInit	**1.8V VDD2 Support** This bit indicates that support of VDD2 on the Host System. 0b: 1.8V VDD2 is not supported 1b: 1.8V VDD2 is supported
59	HwInit	**ADMA3 Support** This bit indicates that support of ADMA3 on Host Controller. 0b: ADMA3 is not supported 1b: ADMA3 is supported
58-56	Rsvd	**Reserved**
55-48	HwInit	**Clock Multiplier** This field indicates clock multiplier value of programmable clock generator. Refer to *Clock Control* register. Setting 00h means that Host Controller does not support programmable clock generator. 00h: Clock Multiplier is Not Supported 01h: Clock Multiplier M = 2 02h: Clock Multiplier M = 3 .....: .... FFh: Clock Multiplier M = 256

<table>
<tr><td>47-46</td><td>HwInit</td><td>

Re-Tuning Modes (UHS-I only)
This field selects re-tuning method and limits the maximum data length.

<table>
<tr><th>Bit47-46</th><th>Re-Tuning Mode</th><th>Re-Tuning Method</th><th>Data Length</th></tr>
<tr><td>00b</td><td>Mode 1</td><td>Timer</td><td>4MB (Max.)</td></tr>
<tr><td>01b</td><td>Mode 2</td><td>Timer and Re-Tuning Request</td><td>4MB (Max.)</td></tr>
<tr><td>10b</td><td>Mode 3</td><td>Auto Re-Tuning (for transfer)
Timer and Re-Tuning Request</td><td>Any</td></tr>
<tr><td>11b</td><td colspan="3">Reserved</td></tr>
</table>

There are two re-tuning timings: **Re-Tuning Request** controlled by the Host Controller and expiration of a Re-Tuning Timer controlled by the Host Driver. By receiving either timing, the Host Driver executes the re-tuning procedure just before a next command issue.

The maximum data length per read/write command is restricted so that re-tuning procedures can be inserted during data transfers.

(1) Re-Tuning Mode 1
The Host Controller does not have any internal logic to detect when the re-tuning needs to be performed. In this case, the Host Driver should maintain all re-tuning timings by using a Re-Tuning Timer. To enable inserting the re-tuning procedure during data transfers, the data length per read/write command shall be limited up to 4MB.

(2) Re-Tuning Mode 2
The Host Controller has the capability to indicate the re-tuning timing by **Re-Tuning Request** during data transfers. Then the data length per read/write command shall be limited up to 4MB.
During non-data transfer, re-tuning timing is determined by either **Re-Tuning Request** or Re-Tuning Timer. If **Re-Tuning Request** is used, Re-Tuning Timer should be disabled.

(3) Re-Tuning Mode 3
The Host Controller has the capability to take care of the re-tuning during data transfer (Auto Re-Tuning). **Re-Tuning Request** shall not be generated during data transfers and there is no limitation to data length per read/write command.
During non-data transfer, re-tuning timing is determined by either **Re-Tuning Request** or Re-Tuning Timer. If **Re-Tuning Request** is used, Re-Tuning Timer should be disabled.

Re-Tuning Timer Control Example for Re-Tuning Mode 1
The initial value of re-tuning timer is provided by **Timer Count for Re-Tuning** field in this register. The timer starts counting by loading the initial value. When the timer expires, the Host Driver marks an expiration flag. On receiving a command request, the Host driver checks the expiration flag. If the expiration flag is set, then the Host Driver should perform the re-tuning procedure before issuing a command. If the expiration flag is not set, then the Host Driver issues a command without performing the re-tuning procedure. Every time the

</td></tr>
</table>

		re-tuning procedure is performed, the timer loads the new initial value and the expiration flag is cleared. Re-Tuning Timer Control Example for Re-Tuning Mode 2 and Mode 3 The timer control is almost the same as Re-Tuning Mode 1 except the timer loads the new initial value after data transfer (when receiving **Transfer Complete**). In case of Mode 3, **Timer Count for Re-Tuning** is set either smaller value: Tuning effective time after re-tuning procedure or after data transfer. If a Host System goes into power down mode, the Host Driver should stop the re-tuning timer and set the expiration flag to 1 when the Host System resumes from power down mode.
45	HwInit	**Use Tuning for SDR50 (UHS-I only)** If this bit is set to 1, this Host Controller requires tuning to operate SDR50. (Tuning is always required to operate SDR104.) 1: SDR50 requires tuning 0: SDR50 does not require tuning
44	Rsvd	**Reserved**
43-40	HwInit	**Timer Count for Re-Tuning (UHS-I only)** This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 0 disables Re-Tuning Timer. 0h: Re-Tuning Timer disabled 1h: 1 seconds 2h: 2 seconds 3h: 4 seconds 4h: 8 seconds .....: .... n: $2^{(n-1)}$ seconds .....: .... Bh: 1024 seconds Eh - Ch: Reserved Fh: Get information from other source
39	Rsvd	**Reserved**
38	HwInit	**Driver Type D Support (UHS-I only)** This bit indicates support of Driver Type D for 1.8 Signaling. 1: Driver Type D is Supported 0: Driver Type D is Not Supported
37	HwInit	**Driver Type C Support (UHS-I only)** This bit indicates support of Driver Type C for 1.8 Signaling. 1: Driver Type C is Supported 0: Driver Type C is Not Supported

<table>
<tr><td>36</td><td>HwInit</td><td>Driver Type A Support (UHS-I only)
This bit indicates support of Driver Type A for 1.8 Signaling.<table><tr><td>1</td><td>Driver Type A is Supported</td></tr><tr><td>0</td><td>Driver Type A is Not Supported</td></tr></table></td></tr>
<tr><td>35</td><td>HwInit</td><td>UHS-II Support (UHS-II only)
This bit indicates whether Host Controller supports UHS-II. If this bit is set to 1, 1.8V VDD2 Support shall be set to 1 (Host System shall support VDD2 power supply).<table><tr><td>1</td><td>UHS-II is Supported</td></tr><tr><td>0</td><td>UHS-II is Not Supported</td></tr></table></td></tr>
<tr><td>34</td><td>HwInit</td><td>DDR50 Support (UHS-I only)<table><tr><td>1</td><td>DDR50 is Supported</td></tr><tr><td>0</td><td>DDR50 is Not Supported</td></tr></table></td></tr>
<tr><td>33</td><td>HwInit</td><td>SDR104 Support (UHS-I only)
SDR104 requires tuning.<table><tr><td>1</td><td>SDR104 is Supported</td></tr><tr><td>0</td><td>SDR104 is Not Supported</td></tr></table></td></tr>
<tr><td>32</td><td>HwInit</td><td>SDR50 Support (UHS-I only)
If SDR104 is supported, this bit shall be set to 1. Bit 45 indicates whether SDR50 requires tuning or not.<table><tr><td>1</td><td>SDR50 is Supported</td></tr><tr><td>0</td><td>SDR50 is Not Supported</td></tr></table></td></tr>
<tr><td>31-30</td><td>HwInit</td><td>Slot Type
This field indicates usage of a slot by a specific Host System. (A Host Controller register set is defined per slot.) Embedded Slot for One Device (01b) means that only one non-removable device is connected to an SD bus slot. Shared Bus Slot (10b) can be set if Host Controller supports Embedded Control register. UHS-II Embedded (11b) means that embedded devices are connected by UHS-II Interface.

The Standard Host Driver controls only a removable card or one embedded device connected to an SD bus slot. If a slot is configured for shared bus (10b) or UHS-II Multiple Embedded Devices (11b), the Standard Host Driver cannot be used. A specific Host Driver is required developed by a Host System.
Refer to Card Interrupt about interrupt signals.<table><tr><td>00b</td><td>Removable Card Slot</td></tr><tr><td>01b</td><td>Embedded Slot for One Device</td></tr><tr><td>10b</td><td>Shared Bus Slot (SD Mode)</td></tr><tr><td>11b</td><td>UHS-II Multiple Embedded Devices</td></tr></table></td></tr>
<tr><td>29</td><td>HwInit</td><td>Asynchronous Interrupt Support (SD Mode only)
Refer to SDIO Specification Version 4.00 about asynchronous interrupt.<table><tr><td>1</td><td>Asynchronous Interrupt Supported</td></tr><tr><td>0</td><td>Asynchronous Interrupt Not Supported</td></tr></table></td></tr>
</table>

<table>
<tr><td>28</td><td>HwInit</td><td>64-bit System Address Support for V3
Meaning of this bit is different depends on Versions (Refer to Table 2-35 for more details). Host Controller Version 3.00 and Ver4.10 use this bit as 64-bit System Address support for V3 mode. Host Controller Version 4.00 uses this bit as 64-bit System Address support for both V3 and V4 modes.

SDMA cannot be used in 64-bit Addressing in Version 3 mode.
If this bit is set to 1, 64-bit ADMA2 with using 96-bit Descriptor may be enabled as follows:
In case of Host Controller Version 3, 64-bit ADMA2 is enabled by DMA Select =11b in the Host Control 1 register. In case of Host Controller Version 4, 64-bit ADMA2 for Version 3 is enabled by setting Host Version 4 Enable =0 and DMA Select = 11b.

<table><tr><td>1</td><td>64-bit System Address for V3 is Supported</td></tr><tr><td>0</td><td>64-bit System Address for V3 is not Supported</td></tr></table></td></tr>
<tr><td>27</td><td>HwInit</td><td>64-bit System Address Support for V4
This bit is added from Version 4.10. Setting 1 to this bit indicates that the Host Controller supports 64-bit System Addressing of Version 4 mode (Refer to Table 2-35 for the summary of 64-bit system address support)..

When this bit is set to 1, full or a part of 64-bit address should be used to decode Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller Registers is effective regardless of setting to 64bit Addressing in Host Control 2.

If this bit is set to 1, 64-bit DMA Addressing for Version 4 is enabled by setting Host Version 4 Enable =1, 64-bit Addressing =1 in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit Descriptor.

<table><tr><td>1</td><td>64-bit System Address for V4 is Supported</td></tr><tr><td>0</td><td>64-bit System Address for V4 is not Supported</td></tr></table></td></tr>
<tr><td>26</td><td>HwInit</td><td>Voltage Support 1.8V
Embedded system can use 1.8V power supply.

<table><tr><td>1</td><td>1.8V Supported</td></tr><tr><td>0</td><td>1.8V Not Supported</td></tr></table></td></tr>
<tr><td>25</td><td>HwInit</td><td>Voltage Support 3.0V

<table><tr><td>1</td><td>3.0V Supported</td></tr><tr><td>0</td><td>3.0V Not Supported</td></tr></table></td></tr>
<tr><td>24</td><td>HwInit</td><td>Voltage Support 3.3V

<table><tr><td>1</td><td>3.3V Supported</td></tr><tr><td>0</td><td>3.3V Not Supported</td></tr></table></td></tr>
</table>

Table 2-34 : Capabilities Register (Part 1)

If a slot is for removable card (Slot Type = 00b), Host System can set Voltage Support 3.3V or 3.0V. Host Driver selects 3.3V in default. If 3.3V is not supported, 3.0V is selected.
If a slot is for embedded device (Slot Type = 01b or 11b), Host System can set one of Voltage Support

bits for host interface voltage (VDDH).

Host Controller	**Version 3.00**	**Version 4.00**	**Version 4.10 or later**
D28 (from Version 2.00)	for V3	for V3 and V4	for V3
D27 (from Version 4.10)	Not Defined	Not Defined	for V4
Register Decode	32-bit or 64-bit (up to implementation)		If D27=1, 64-bit
SDMA	Not supported	Supported when **Host Version 4 Enable** =1	
ADMA2 (96-bit Descriptor)	**DMA Select**=11b	Selected by **Host Version 4 Enable** =0	
ADMA2 (128-bit Descriptor)	Not Defined	Selected by **Host Version 4 Enable** =1	

Table 2-35 : 64-bit System Address Support depends on Versions

As the specification of 64-bit System Address Support has been changed, capabilities of 64-bit functions are different depends on versions. Table 2-35 shows summary of 64-bit System Address Support.
Definition of D28 is different depends on Versions. 96-bit Descriptor was defined by Version 2 but notation V3 is used including V2. Version 4.10 divides 64-bit System Address Support into V3 mode (D28) and V4 mode (D27) so that V3 mode can be optional. Migrate to V4 is recommended. From Host Controller Version 4.00, either V3 mode or V4 mode is selected by **Host Version 4 Enable** in the *Host Control 2* register. V3 mode can be used if **64-bit System Address Support for V3** is set to 1. V4 mode can be used if **64-bit System Address Support for V4** is set to 1.

Prior to Version 4.10, address length of Host Controller registers decoding is not defined and whether 32-bit or 64-bit address is used to decode Host Controller registers is up to implementation. If Host Controller decodes 32-bit system address in default, the Host Controller Registers shall be placed in 32-bit addressing space.

When D27=1, Host Controller Version 4.10 or later should use full or a part of 64-bit address to decode Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller Registers is effective regardless of setting to **64bit Addressing** in *Host Control 2.* How to decode register also should follow a system bus specification or a motherboard specification.

From Version 4.00, 64-bit System Addressing of DMA is enabled by setting **64-bit Addressing** in the *Host Control 2.* 64-bit SDMA is not supported in V3 mode and is supported in V4 mode. There are two Descriptor types for ADMA2 96-bit (V3) or 128-bit (V4). Support of 96-bit Descriptor is optional for Host Controller Version 4.10. If D28=0, 96-bit Descriptor is not supported.

Location	Attrib	Register Field Explanation
23	HwInit	**Suspend/Resume Support** This bit indicates whether the Host Controller supports Suspend/Resume function. If this bit is 0, the Host Driver shall not issue either Suspend or Resume commands because the Suspend/Resume mechanism (Refer to Section 1.6) is not supported. 1 Supported 0 Not Supported
22	HwInit	**SDMA Support** This bit indicates whether the Host Controller is capable of using SDMA to transfer data between system memory and the Host Controller directly. Version 4.10 Host Controller shall support SDMA if ADMA2 is supported. 1 SDMA Supported 0 SDMA not Supported

21	HwInit	**High Speed Support** This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz. 1 – High Speed Supported 0 – High Speed not Supported
20	HwInit	**Reserved (New assignment is not allowed)** This bit is reserved for backward compatibility with prior specifications. If set, the Host Controller is indicating that it supports legacy ADMA1 mode. Host drivers are not required to support this mode.
19	HwInit	**ADMA2 Support** This bit indicates whether the Host Controller is capable of using ADMA2. Version 4.10 Host Controller shall support ADMA2 if ADMA3 is supported. 1 – ADMA2 Supported 0 – ADMA2 not Supported
18	HwInit	**8-bit Support for Embedded Device (Embedded)** This bit indicates whether the Host Controller is capable of using 8-bit bus width mode. This bit is not effective when **Slot Type** is set to 10b. In this case, refer to **Bus Width Preset** in the *Embedded Control* resister. 1 – 8-bit Bus Width Supported 0 – 8-bit Bus Width not Supported
17-16	HwInit	**Max Block Length** This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. It is noted that transfer block length shall be always 512 bytes for SD Memory Cards regardless this field. 00 – 512(byte) 01 – 1024 10 – 2048 11 – Reserved

<table>
<tr><td>15-08</td><td>HwInit</td><td>Base Clock Frequency For SD Clock
This value indicates the base (maximum) clock frequency for the SD Clock. Definition of this field depends on Host Controller Version.

(1) 6-bit Base Clock Frequency
This mode is supported by the Host Controller Version 1.00 and 2.00. Upper 2 bits are not effective and always 0.
Unit values are 1MHz. The supported clock range is 10MHz to 63MHz.

<table>
<tr><td>11xx xxxxb</td><td>Not supported</td></tr>
<tr><td>0011 1111b</td><td>63MHz</td></tr>
<tr><td>.........</td><td>...........</td></tr>
<tr><td>0000 0010b</td><td>2MHz</td></tr>
<tr><td>0000 0001b</td><td>1MHz</td></tr>
<tr><td>0000 0000b</td><td>Get information via another method</td></tr>
</table>
(2) 8-bit Base Clock Frequency
This mode is supported by the Host Controller Version 3.00.
Unit values are 1MHz. The supported clock range is 10MHz to 255MHz.

<table>
<tr><td>FFh</td><td>255MHz</td></tr>
<tr><td>.........</td><td>...........</td></tr>
<tr><td>02h</td><td>2MHz</td></tr>
<tr><td>01h</td><td>1MHz</td></tr>
<tr><td>00h</td><td>Get information via another method</td></tr>
</table>
If the real frequency is 16.5MHz, the lager value shall be set 0001 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the SDCLK/RCLK Frequency Select in the Clock Control register.) and it shall not exceed upper limit of the SD Clock frequency.
If these bits are all 0, the Host System has to get information via another method.</td></tr>
<tr><td>07</td><td>HwInit</td><td>Timeout Clock Unit
This bit shows the unit of base clock frequency used to detect Data Timeout Error.

<table>
<tr><td>0</td><td>KHz</td></tr>
<tr><td>1</td><td>MHz</td></tr>
</table></td></tr>
<tr><td>06</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>05-00</td><td>HwInit</td><td>Timeout Clock Frequency
This bit shows the base clock frequency used to detect Data Timeout Error.
The Timeout Clock Unit defines the unit of this field's value.
Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz
Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz

<table>
<tr><td>Not 0</td><td>1KHz to 63KHz or 1MHz to 63MHz</td></tr>
<tr><td>00 0000b</td><td>Get information via another method</td></tr>
</table></td></tr>
</table>

Table 2-36 : Capabilities Register (Part 2)

2.2.27 Maximum Current Capabilities Register (Cat.C Offset 048h)

These registers indicate maximum current capability for each voltage. The value is meaningful if **Voltage Support** is set in the *Capabilities* register. If this information is supplied by the Host System via another method, all *Maximum Current Capabilities* register shall be 0.

D63 D40			D39 D32
Rsvd			Maximum Current for 1.8V VDD2
D31 D24	D23 D16	D15 D08	D07 D00
Rsvd	Maximum Current for 1.8V	Maximum Current for 3.0V	Maximum Current for 3.3V

Figure 2-31 : Maximum Current Capabilities Register

Location	Attrib	Register Field Explanation
63-56	Rsvd	**Reserved**
55-48	Rsvd	**Reserved**
47-40	Rsvd	**Reserved**
39-32	Rsvd	**Maximum Current for 1.8V VDD2**
31-24	Rsvd	**Reserved**
23-16	HwInit	**Maximum Current for 1.8V VDD1**
15-08	HwInit	**Maximum Current for 3.0V VDD1**
07-00	HwInit	**Maximum Current for 3.3V VDD1**

Table 2-37 : Maximum Current Capabilities Register

This register measures current in 4mA steps. Each voltage level's current support is described using the Table 2-38.

Register Value	Current Value
0	Get information via another method
1	4mA
2	8mA
3	12mA
....	....
255	1020mA

Table 2-38 : Maximum Current Value Definition

SDXC card supported Host Driver needs to check this register to determine **XPC** value in the argument of ACMD41. If a Host System can afford more than 150mA, Host Driver set **XPC** to 1. If a Host System can afford less than 150mA, Host Driver set **XPC** to 0. Refer to the Physical Layer Specification Version 3.0x for more detail of **XPC**.

2.2.28 Force Event Register for Auto CMD Error Status (Cat.A Offset 050h)

The *Force Event* register is not a physically implemented register, or rather used for setting any bit of interrupt status, which is difficult to set intentionally. This register simplifies test of the *Auto CMD Error Status* register, which corresponds to **Auto CMD Error** interrupt in the *Interrupt Error Status* register.

Writing 1: Set correspondent bit of the *Auto CMD Error Status* register
Writing 0: no effect

D15 - D08	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd	Force Event for Command Not Issued by Auto CMD12 Error	Rsvd	Force Event for Auto CMD Response Error	Force Event for Auto CMD Index Error	Force Event for Auto CMD End Bit Error	Force Event for Auto CMD CRC Error	Force Event for Auto CMD Timeout Error	Force Event for Auto CMD12 not executed

Figure 2-32 : Force Event Register for Auto CMD Error Status

Location	Attrib	Register Field Explanation
15-08	Rsvd	**Reserved**
07	WO	**Force Event for Command Not Issued By Auto CMD12 Error** 1: Command Not Issued By Auto CMD12 Error Status is set 0: Not Affected
06	Rsvd	**Reserved**
05		**Force Event for Auto CMD Response Error** 1: Auto CMD Response Error Status is set 0: Not Affected
04	WO	**Force Event for Auto CMD Index Error** 1: Auto CMD Index Error Status is set 0: Not Affected
03	WO	**Force Event for Auto CMD End Bit Error** 1: Auto CMD End Bit Error Status is set 0: Not Affected
02	WO	**Force Event for Auto CMD CRC Error** 1: Auto CMD CRC Error Status is set 0: Not Affected
01	WO	**Force Event for Auto CMD Timeout Error** 1: Auto CMD Timeout Error Status is set 0: Not Affected
00	WO	**Force Event for Auto CMD12 Not Executed** 1: Auto CMD12 Not Executed Status is set 0: Not Affected

Table 2-39 : Force Event Register for Auto CMD Error Status

2.2.29 Force Event Register for Error Interrupt Status (Cat.A Offset 052h)

The *Force Event* register is not a physically implemented register, or rather used for setting any bit of interrupt status, which is difficult to set intentionally. This register simplifies test of the *Error Interrupt Status* register.

Writing 1: Set correspondent bit of the *Error Interrupt Status* register
Writing 0: no effect

In order to generate interrupt signal, the correspondent bit shall be set in the *Error Interrupt Status Enable* register and *Error Interrupt Signal Enable* register.

D15 D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Force Event for Vendor Specific Error Status	Force Event for Response Error	Force Event for Tuning Error	Force Event for ADMA Error	Force Event for Auto CMD Error	Force Event for Current limit Error	Force Event for Data End Bit Error	Force Event for Data CRC Error	Force Event for Data Timeout Error	Force Event for Command Index Error	Force Event for Command End Bit Error	Force Event for Command CRC Error	Force Event for Command Timeout Error

Figure 2-33 : Force Event Register for Error Interrupt Status

Location	Attrib	Register Field Explanation
15-12	WO	**Force Event for Vendor Specific Error Status** Additional status bits can be defined in this register by the vendor. 1: Vendor Specific Error Status is set 0: Not Affected
11	WO	**Force Event for Response Error** 1: Response Error Status is set 0: Not Affected
10	WO	**Force Event for Tuning Error** 1: Tuning Error Status is set 0: Not Affected
09	WO	**Force Event for ADMA Error** 1: ADMA Error Status is set 0: Not Affected
08	WO	**Force Event for Auto CMD Error** 1: Auto CMD Error Status is set 0: Not Affected
07	WO	**Force Event for Current Limit Error** 1: Current Limit Error Status is set 0: Not Affected
06	WO	**Force Event for Data End Bit Error** 1: Data End Bit Error Status is set 0: Not Affected

05	WO	**Force Event for Data CRC Error** 1 : CRC Error Status is set 0 : Not Affected
04	WO	**Force Event for Data Timeout Error** 1 : Timeout Error Status is set 0 : Not Affected
03	WO	**Force Event for Command Index Error** 1 : Command Index Error Status is set 0 : Not Affected
02	WO	**Force Event for Command End Bit Error** 1 : Command End Bit Error Status is set 0 : Not Affected
01	WO	**Force Event for Command CRC Error** 1 : Command CRC Error Status is set 0 : Not Affected
00	WO	**Force Event for Command Timeout Error** 1 : Command Timeout Error Status is set 0 : Not Affected

Table 2-40 : Force Event for Error Interrupt Status Register

2.2.30 ADMA Error Status Register (Cat.C Offset 054h)

When **ADMA Error** Interrupt is occurred, the **ADMA Error States** field in this register holds the ADMA state and the *ADMA System Address* register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address
ST_FDS: Current location set in the ADMA System Address register is the error descriptor address
ST_CADR: This state is never set because do not generate ADMA error in this state.
ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address

In case of write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller.
The Host Controller generates the **ADMA Error** Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the "Valid" bit is not set in the error descriptor.

D07 D03	D02	D01 D00
Rsvd	ADMA Length Mismatch Error	ADMA Error States

Figure 2-34 : ADMA Error Status Register

Location	Attrib	Register Field Explanation
07-03	Rsvd	**Reserved**
02	ROC	**ADMA Length Mismatch Error** This error occurs in the following 2 cases. (1) While **Block Count Enable** being set, the total data length specified by the Descriptor table is different from that specified by the *Block Count* and *Block Length*. (2) Total data length cannot be divided by the block length. 1 : Error 0 : No Error
01-00	ROC	**ADMA Error State** This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. D01 – D00 / ADMA Error State when error is occurred / Contents of SYS_SDR register 00 / ST_STOP (Stop DMA) / Points next of the error descriptor 01 / ST_FDS (Fetch Descriptor) / Points the error descriptor 10 / Never set this state / (Not used) 11 / ST_TFR (Transfer Data) / Points the next of the error descriptor

Table 2-41 : ADMA Error Status Register

2.2.31 ADMA System Address Register (Cat.C Offset 05Fh-058h)

This register contains the physical Descriptor address used for ADMA data transfer.

D63	D00
ADMA System Address	

Figure 2-35 : ADMA System Address Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>63-00</td><td>RW</td><td>ADMA System Address
For 32-bit addressing
The 32-bit addressing Host Driver uses lower 32-bit of this register (upper 32-bit should be set to 0) to point to top of first descriptor on the system memory and shall program Descriptor Tables on 32-bit boundary due to DMA2/3 ignores lower 2-bit of this register and assumes it to be 00b.

For 64-bit addressing
The 64-bit addressing Host Driver uses this 64-bit register to point to top of first descriptor on the system memory and shall program Descriptor Tables on 64-bit boundary due to DMA2/3 ignores lower 3-bit of this register and assumes it to be 000b.

(1) SDMA
If Host Version 4 Enable is set to 1, SDMA use this register to indicate System Address of data location instead of using SDMA System Address register (Offset 003-000h). SDMA can be used in 32-bit and 64-bit addressing from Version 4.00.

(2) ADMA2
This register holds byte address of executing command of the Descriptor table.
At the start of ADMA2, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.

(3) ADMA3
This register is set by ADMA3. Host Driver is not necessary to set this register. The ADMA3 increments address of this register, which points to next line, when every time fetching a Descriptor line. When Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.

<table><tr><td>Register Value</td><td>Addressing Mode</td></tr><tr><td>00000000_xxxxxxxxh</td><td>32-bit System Address</td></tr><tr><td>xxxxxxxx_xxxxxxxxh</td><td>64-bit System Address</td></tr></table></td></tr>
</table>

Table 2-42 : ADMA System Address Register

2.2.32 Preset Value Registers (Cat.C Offset 074-060h)

Offset	Preset Value Registers	Signal Voltage
060h	Preset Value for Initialization	3.3V or 1.8V
062h	Preset Value for Default Speed	3.3V
064h	Preset Value for High Speed	3.3V
066h	Preset Value for SDR12	1.8V
068h	Preset Value for SDR25	1.8V
06Ah	Preset Value for SDR50	1.8V
06Ch	Preset Value for SDR104	1.8V
06Eh	Preset Value for DDR50	1.8V
070h	Reserved	
072h	Reserved	
074h	Preset Value for UHS-II	No switch

Table 2-43 : Preset Value Registers

Table 2-43 shows a set of preset values per card or device. One of the *Preset Value* registers (06Eh - 062h) is effective based on the Selected Bus Speed Mode. Table 2-44 defines the conditions to select one of *Preset Value* registers and Figure 2-36 defines fields of a *Preset Value* register. When **Preset Value Enable** in the *Host Control 2* register is set to 1, **SDCLK/RCLK Frequency Select and Clock Generator Select** in the *Clock Control* register, and **Driver Strength Select** in the *Host Control 2* register are automatically set based on the Selected Bus Speed Mode. This means the Host Driver needs not set these fields when preset is enabled. A *Preset Value for Initialization* (060h) is not selected by bus speed mode. Before starting the initialization sequence, the Host Driver needs to set a clock preset value to **SDCLK/RCLK Frequency Select** in the *Clock Control* register. **Preset Value Enable** can be set after initialization completed.

Selected Bus Speed Mode	1.8V Signaling Enable (Host Control 2)	High Speed Enable (Host Control 1)	UHS-I Mode Selection (Host Control 2)
Default Speed	0	0	don't care
High Speed	0	1	don't care
SDR12	1	don't care	000b
SDR25	1	don't care	001b
SDR50	1	don't care	010b
SDR104	1	don't care	011b
DDR50	1	don't care	100b
Reserved	Not determined	don't care	101b
Reserved	Not determined	don't care	110b
UHS-II	0	don't care	111b

Table 2-44 : Preset Value Register Select Condition

D15 - D14	D13 - D11	D10	D09 D00
Driver Strength Select Value	Reserved	Clock Generator Select Value	SDCLK/RCLK Frequency Select Value

Figure 2-36 : Fields of One Preset Value Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-14</td><td>HwInit</td><td>Driver Strength Select Value (UHS-I only)
Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling.
<table><tr><td>11b</td><td>Driver Type D is Selected</td></tr><tr><td>10b</td><td>Driver Type C is Selected</td></tr><tr><td>01b</td><td>Driver Type A is Selected</td></tr><tr><td>00b</td><td>Driver Type B is Selected</td></tr></table></td></tr>
<tr><td>13-11</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>10</td><td>HwInit</td><td>Clock Generator Select Value
This bit is effective when Host Controller supports programmable clock generator.
<table><tr><td>1</td><td>Programmable Clock Generator</td></tr><tr><td>0</td><td>Host Controller Ver2.00 Compatible Clock Generator</td></tr></table></td></tr>
<tr><td>09-00</td><td>HwInit</td><td>SDCLK/RCLK Frequency Select Value
10-bit preset value to set SDCLK/RCLK Frequency Select in the Clock Control register is described by a host system.</td></tr>
</table>

Table 2-45 : Fields of One Preset Value Register

When Host Controller supports shared bus, a set of *Preset Value* registers for each device is required and the registers location are duplicated to the offset 06Fh-060h. A set of *Preset Value* registers can be accessible by selecting **Clock Pin Select** in the *Embedded Control* register.

Implementation Notes for Standard Host Driver
As "Driver Strength" and "Programmable Clock Mode" settings are Host Controller and Host System dependent parameters, it is difficult for Standard Host Driver to calculate those preset values. Basically, Preset Value Registers are set by HwInit. However, if Preset Value Registers are set to 0, Standard Host Driver should calculate clock frequencies at least for card initialization, Default Speed and High Speed modes using Divided Clock mode by referring *Capabilities* register. (Host Driver may use *Preset* Value Registers or direct setting to *Clock Control* register and *Host Control 2* register.) If preset values for UHS modes are set to 0, Standard Host Driver should initialize card with Default Speed or High Speed mode.

2.2.33 ADMA3 Integrated Descriptor Address (Cat.C Offset 07F-078h)

D63	D00
Integrated DMA Descriptor Address	

Figure 2-37 : ADMA3 Integrated Descriptor Address Register

Location	Attrib	Register Field Explanation
63-00	RW	**ADMA3 Integrated Descriptor Address** The start address of Integrated DMA Descriptor is set to this register. Writing to a specific address starts ADMA3 depends on 32-bit/64-bit addressing. The ADMA3 fetches one Descriptor Address and increments this field to indicate the next Descriptor address. For 32-bit addressing The 32-bit addressing Host Driver uses lower 32-bit of this register (upper 32-bit should be set to 0) to point to top of the Integrated DMA Descriptor on the system memory and shall program Descriptor Tables on 32-bit boundary due to ADMA3 ignores lower 2-bit of this register and assumes it to be 00b. Writing to 07Bh starts ADMA3 data transfer. For 64-bit addressing The 64-bit addressing Host Driver uses this 64-bit register to point to top of the Integrated DMA Descriptor on the system memory and shall program Descriptor Tables on 64-bit boundary due to ADMA3 ignores lower 3-bit of this register and assumes it to be 000b. Writing to 07Fh starts ADMA3 data transfer. Register Value: Addressing Mode 00000000_xxxxxxxxh: 32-bit System Address xxxxxxxx_xxxxxxxxh: 64-bit System Address

Register Value	Addressing Mode
00000000_xxxxxxxxh	32-bit System Address
xxxxxxxx_xxxxxxxxh	64-bit System Address

Table 2-46 : Integrated DMA Descriptor Address Register

2.3 UHS-II Registers in 000-0FFh

All UHS-II registers are Category B.

2.3.1 UHS-II Block Size (Cat.B Offset 081-080h)

D15	D14 D12	D11 D00
Reserved	UHS-II SDMA Buffer Boundary	UHS-II Block Size

Figure 2-38 : UHS-II Block Size Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>14-12</td><td></td><td>UHS-II SDMA Buffer Boundary (SDMA only)
When system memory is managed by paging, SDMA data transfer is performed in unit of paging. A page size of system memory management is set to this field. Host Controller generates the DMA Interrupt at the page boundary and requests the Host Driver to update the ADMA System Address register. SDMA waits until the ADMA System Address register is written.
At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. In particular, DMA Interrupt shall not be issued after Transfer Complete Interrupt is issued.
These bits shall be supported when the SDMA Support in the Capabilities register is set to 1 and this function is active when the DMA Enable in the UHS-II Transfer Mode register is set to 1. ADMA does not use this field.

<table>
<tr><td>000b</td><td>4K bytes</td><td>(Detects A11 carry out)</td></tr>
<tr><td>001b</td><td>8K bytes</td><td>(Detects A12 carry out)</td></tr>
<tr><td>010b</td><td>16K Bytes</td><td>(Detects A13 carry out)</td></tr>
<tr><td>011b</td><td>32K Bytes</td><td>(Detects A14 carry out)</td></tr>
<tr><td>100b</td><td>64K bytes</td><td>(Detects A15 carry out)</td></tr>
<tr><td>101b</td><td>128K Bytes</td><td>(Detects A16 carry out)</td></tr>
<tr><td>110b</td><td>256K Bytes</td><td>(Detects A17 carry out)</td></tr>
<tr><td>111b</td><td>512K Bytes</td><td>(Detects A18 carry out)</td></tr>
</table></td></tr>
<tr><td>11-00</td><td>RW</td><td>UHS-II Block Size
This register specifies the block size of data packet. SD Memory Card uses a fixed block size of 512 bytes. Variable block size may be used for SDIO. The maximum value is 2048 Bytes because CRC16 covers up to 2048 bytes.
This register is effective when Data Present is set to 1 in UHS-II Command register.

<table>
<tr><td>0000h</td><td>No data transfer</td></tr>
<tr><td>0001h</td><td>1 Byte</td></tr>
<tr><td>0002h</td><td>2 Bytes</td></tr>
<tr><td>0003h</td><td>3 Bytes</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>01FFh</td><td>511 Bytes</td></tr>
<tr><td>0200h</td><td>512 Bytes</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0800h</td><td>2048 Bytes</td></tr>
</table></td></tr>
</table>

Table 2-47 : UHS-II Block Size Register

2.3.2 UHS-II Block Count (Offset 087-084h)

D31	D00
UHS-II Blocks Count	

Figure 2-39 : UHS-II Block Count Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>31-00</td><td>RW</td><td>UHS-II Blocks Count
This register is effective when Data Present is set to 1 in UHS-II Command register and is enabled when Block Count Enable is set to 1 and Block / Byte Mode is set to 0 in the UHS-II Transfer Mode register. Data transfer stops when the count reaches zero. Setting the block count to 0 results in no data blocks is transferred.

This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.

<table><tr><td>00000000h</td><td>Stop Count</td></tr><tr><td>00000001h</td><td>1 block</td></tr><tr><td>00000002h</td><td>2 blocks</td></tr><tr><td>......</td><td>......</td></tr><tr><td>FFFFFFFFh</td><td>4G blocks -1</td></tr></table></td></tr>
</table>

Table 2-48 : Block Count Register

2.3.3 UHS-II Command Packet (Cat.B Offset 09B-088h)

UHS-II Command Packet image is set to this register. The maximum length is 20 bytes. The command length varies depends on a Command Packet type. The length is specified by the *UHS-II Command* register. Refer to the UHS-II Addendum for more details about a Command Packet image.

Offset	Command Packet Registers
088h	Command Packet Byte 0
089h	Command Packet Byte 1
08Ah	Command Packet Byte 2
......	....
09Bh	Command Packet Byte 19

Table 2-49 : UHS-II Command Packet Register

2.3.1 UHS-II Transfer Mode (Cat.B Offset 09D-09Ch)

On issuing a Command Packet, a Command Packet image is set to *UHS-II Command Packet* register but Host Controller does not analyze the setting of *UHS-II Command Packet* register. Instead, Host Controller refers setting of this register to issue a Command Packet to make the control easy. Setting of these registers shall be correspondent.

D15	D14	D13 D09	D08	D07	D06	D05	D04	D03 D02	D01	D00
Half / Full Select	EBSY Wait	Reserved	Response Interrupt Disable	Response Error Check Enable	Response Type R1/R5	Block / Byte Mode	Data transfer Direction	Reserved	Block Count Enable	DMA Enable

Figure 2-40 : UHS-II Transfer Mode Register

Location	Attrib	Register Field Explanation
15	R/W	**Half / Full Select** Host Driver determines use of 2 lanes half-duplex mode. 0 : Full Duplex Mode 1 : 2 Lane Half Duplex Mode
14	R/W	**EBSY Wait** This bit is set when issuing a command which is accompanied by EBSY packet to indicate end of command execution. Busy is expected for CCMD with R1b/R5b type and DCMD with data transfer. If this bit is set to 1, Host Controller waits receiving of EBSY packet and on receiving EBSY packet, **Transfer Complete** in the *Normal Interrupt Status* register is set to 1 to indicate end of busy. If an error is indicated in EBSY packet (ex. Memory Error), **EBSY Error** in the *UHS-II Error Interrupt Status* register is set to 1. Setting of **EBSY Error** also sets **Error Interrupt** to 1 in the *Normal Interrupt Status* register. **Error Interrupt** and **Transfer Complete** shall be set together. 0 : Issue a command without busy 1 : Wait EBSY
13-09	Rsvd	**Reserved**
08	R/W	**Response Interrupt Disable** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, sets this bit to 0 and waits **Command Complete** Interrupt and then check the response register. If Host Controller checks response error, sets this bit to 1 and sets **Response Error Check Enable** to 1. **Command Complete** Interrupt is disabled by this bit regardless of **Command Complete Signal Enable.** 0 : Response Interrupt is enabled 1 : Response Interrupt is disabled

07	R/W	**Response Error Check Enable** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, this bit is set to 0 and **Response Interrupt Disable** is set to 0. If Host Controller checks response error, sets this bit to 1 and sets **Response Interrupt Disable** to 1. **Response Type R1/R5** selects either R1 or R5 response type. If an error is detected, **RES Packet Error** Interrupt is generated in the *UHS-II Error Interrupt Status* register. 0: Response Error Check is disabled 1: Response Error Check is enabled
06	R/W	**Response Type R1/R5** When response error check is enabled, this bit selects either R1 or R5 response types. Two types of response checks are supported: R1 for memory and R5 for SDIO. Error Statuses Checked in R1 Bit31 OUT_OF_RANGE Bit30 ADDRESS_ERROR Bit29 BLOCK_LEN_ERROR Bit26 WP_VIOLATION Bit25 CARD_IS_LOCKED Bit23 COM_CRC_ERROR Bit21 CARD_ECC_FAILED Bit20 CC_ERROR Bit19 ERROR Response Flags Checked in R5 Bit07 COM_CRC_ERROR Bit03 ERROR Bit01 FUNCTION_NUMBER Bit00 OUT_OF_RANGE 0: R1 (Memory) 1: R5 (SDIO)
05	R/W	**Block / Byte Mode** This bit specifies whether data transfer is in byte mode or block mode when **Data Present** is set to 1. This bit is effective to a command with data transfer. 0: Block Mode 1: Byte Mode
04	R/W	**Data Transfer Direction** This bit specifies direction of data transfer when **Data Present** is set to 1. This bit is effective to a command with data transfer. 0: Read (Card to Host) 1: Write (Host to Card)
03-02	Rsvd	**Reserved**

<table>
<tr><td>01</td><td>R/W</td><td>Block Count Enable
This bit specifies whether data transfer uses UHS-II Block Count register. If this bit is set to 1, data transfer is terminated by Block Count. Setting to UHS-II Block Count register shall be equivalent to TLEN in UHS-II Command Packet register.

<table><tr><td>0</td><td>Block Count Disabled</td></tr><tr><td>1</td><td>Block Count Enabled</td></tr></table></td></tr>
<tr><td>00</td><td>R/W</td><td>DMA Enable
This bit selects whether DMA is used or not and is effective to a command with data transfer. One of DMA types is selected by DMA Select in the Host Control 1 register.

<table><tr><td>0</td><td>DMA is disabled</td></tr><tr><td>1</td><td>DMA is enabled</td></tr></table></td></tr>
</table>

Table 2-50 : UHS-II Transfer Mode Register

2.3.2 UHS-II Command (Cat.B Offset 09F-9Eh)

Writing to upper byte of this register acts as a trigger to issue the Command Packet.

D15 D13	D12 D08	D07 D06	D05	D04 D03	D02	D01 D00
Reserved	UHS-II Command Packet Length	Command Type	Data Present	Reserved	Sub Command Flag	Reserved

Figure 2-41 : UHS-II Command Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-13</td><td>Rsvd</td><td>Reserved (000b)</td></tr>
<tr><td>12-08</td><td>R/W</td><td>UHS-II Command Packet Length
A command packet length, which is set in the UHS-II Command Packet register, is set to this register.
<table>
<tr><td>00011b - 00000b</td><td>3-0 Bytes (Not used)</td></tr>
<tr><td>00100b</td><td>4 Bytes</td></tr>
<tr><td>.......</td><td>......</td></tr>
<tr><td>10100b</td><td>20 Bytes</td></tr>
<tr><td>11111b – 10101b</td><td></td></tr>
</table></td></tr>
<tr><td>07-06</td><td>R/W</td><td>Command Type
This field is used to distinguish a specific command like abort command. If this field is set to 00b, the UHS-II RES Packet is stored in UHS-II Response register (0B3h-0A0h). To avoid overwriting the UHS-II Response register, when this filed is set to 01b, the RES Packet (4 bytes length) of TRANS_ABORT CCMD is stored in the Response register (013h-010h) and when this filed is set to 10b, the RES Packet (8 bytes length) of memory or SDIO abort command (CMD12 or SDIO Abort command) is stored in the Response register (01Fh-018h). When this filed is set to 11b to issue GO_DORMANT_STATE CCMD or FULL_RESET CCMD, the response of the command is saved to 0B3h-0A0h (same location as Command Type 00b) and then Host Controller controls lane to go into dormant state.
After issuing TRANS_ABORT CCMD, Host Driver should use Host Full Reset to re-initialize Host Controller. After issuing CMD12 or SDIO Abort, Host Driver should use Host SD-TRAN Reset to discard data in the Host Controller buffer. (Refer to 3.8 Abort Transaction) If any of abort command is issued while data transfer is being stopped by Block At Gap Request, data circuits including DMA are still being stopped.
<table>
<tr><td>00b</td><td>Normal Command</td></tr>
<tr><td>01b</td><td>TRANS_ABORT CCMD</td></tr>
<tr><td>10b</td><td>CMD12 or SDIO Abort command</td></tr>
<tr><td>11b</td><td>Go Dormant Command</td></tr>
</table></td></tr>
<tr><td>05</td><td>R/W</td><td>Data Present
This bit specifies whether the command is accompanied by data packet.
<table>
<tr><td>0</td><td>No Data Present</td></tr>
<tr><td>1</td><td>Data Present</td></tr>
</table></td></tr>
<tr><td>04-03</td><td>Rsvd</td><td>Reserved (00000b)</td></tr>
</table>

02	R/W	**Sub Command Flag** This bit is added from Version 4.10 to distinguish a main command or sub command (Refer to Section 1.17). When issuing a main command, this bit is set to 0 and when issuing a sub command, this bit is set to 1. Setting of this bit is checked by **Sub Command Status** in the *Present State* register. 0: Sub Command 1: Main Command
01-00	Rsvd	**Reserved (00000b)**

Table 2-51 : UHS-II Command Register

2.3.3 UHS-II Response (Cat.B Offset 0B3-0A0h)

Host Controller saves received UHS-II RES Packet image to this register except the response of an abort command, which is specified by setting 01b or 10b to **Command Type** in the *UHS-II Command* register. The maximum response length is 20 bytes.

Offset	Response Packet Registers
0A0h	Response Packet Byte 0
0A1h	Response Packet Byte 1
0A2h	Response Packet Byte 2
......	....
0B3h	Response Packet Byte 19

Table 2-52 : UHS-II Response Register

2.3.4 UHS-II MSG Select (Cat.B Offset 0B4h)

D07 — D02	D01 — D00
Reserved (000000b)	UHS-II MSG Select

Figure 2-42 : UHS-II MSG Select Register

Location	Attrib	Register Field Explanation
07-02	Rsvd	**Reserved (000000b)**
01-00	R/W	**UHS-II MSG Select** Host Controller holds 4 MSG packets in FIFO buffer. One of 4 MSGs can be read from the *UHS-II MSG* register (0BB-0B8h) by setting this register. (Assumed for debug usage.) 00b: The latest MSG 01b: One MSG before 10b: Two MSGs before 11b: Three MSGs before

Table 2-53 : UHS-II MSG Select Register

2.3.5 UHS-II MSG Register (Cat.B Offset 0BB-0B8h)

Host Controller holds four MSG packets, which are received from card, in FIFO buffer. Then Host Controller stores FCRDY, STAT and EBSY for write operation and stores FCREQ and EBSY for read operation. One of four MSGs (length is 4 bytes) can be read from this register by setting *UHS-II MSG*

Select register. Usually two duplicate MSG packets are sent from UHS-II card. One of these two MSG packets, which Host Controller recognizes as valid, is stored in the *UHS-II MSG* register.

D31	D00
UHS-II MSG	

Figure 2-43 : UHS-II MSG Register

2.3.6 UHS-II Device Interrupt Status (Cat.B Offset 0BD-0BCh)

D15	D00
UHS-II Device Interrupt Status	

Figure 2-44 : UHS-II Device Interrupt Status Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-00</td><td>RW1C</td><td>UHS-II Device Interrupt Status
This register shows receipt of INT MSG from which device and is effective when INT MSG Enable is set to 1 in the UHS-II Device Select register. On receiving INT MSG from a device, Host Controller saves the INT MSG to UHS-II Device Interrupt Code register. A bit of this register, which is correspondent to Device ID, is set to 1 and generate Card Interrupt in Normal Interrupt Status register. Writing a bit to 1 clears the status bit (interrupt is treated) and writing a bit to 0 keeps the status value (interrupt is untreated).
If INT MSG Enable is set to 0, this register is cleared to 0 and Host Controller ignores receipt of INT MSG.

Effective bit range of this register is determined by Number of Devices Supported in the UHS-II General Capabilities register. If N devices are supported, bits 1 to N are effective. Then Device ID is supposed to be assigned from 1 sequentially at the UHS-II Initialization. A bit of unsupported Device ID in this register shall be indicated to 0.

<table>
<tr><td>D00</td><td>Not used (Reserved)</td></tr>
<tr><td>D01</td><td>Setting 1 means INT MSG is received from Device ID 1</td></tr>
<tr><td>D02</td><td>Setting 1 means INT MSG is received from Device ID 2</td></tr>
<tr><td>...</td><td>......</td></tr>
<tr><td>D15</td><td>Setting 1 means INT MSG is received from Device ID 15</td></tr>
</table></td></tr>
</table>

Table 2-54 : UHS-II Device Interrupt Status Register

2.3.7 UHS-II Device Select (Offset 0BEh)

D07	D06 D04	D03 D00
INT MSG Enable	Reserved (000b)	UHS-II Device Select

Figure 2-45 : UHS-II Device Select Register

Location	Attrib	Register Field Explanation
07	R/W	**INT MSG Enable (Optional)** This bit enables receipt of INT MSG. If this bit is set to 1, receipt of INT MSG is informed by **Card Interrupt** in the *Normal Interrupt Status* register. If this bit is set to 0, Host Controller ignores receipt of INT MSG and may not set the *UHS-II Device Interrupt Code r*egister. Support of INT MSG Interrupt is optional. If trying to set this bit to 1 but still this bit is read 0, INT MSG Interrupt is not supported by the Host Controller. In this case, *UHS-II Device Interrupt Status* register always shall be read 0 and UHS-II Device Interrupt Code register may not be implemented. 0h: Disabled 1h: Enabled
06-04	Rsvd	**Reserved (000000b)**
03-00	R/W	**UHS-II Device Select** Host Controller holds an INT MSG packet per device. One of INT MSGs (up to 15) can be selected by this field and read from the *UHS-II Device Interrupt Code* register (0BFh). This field is effective when **INT MSG Enable** is set to 1. The number of devices implemented in the Host Controller is indicated by **Number of Devices Supported** in the *UHS-II General Capabilities* register. 0h: Unselected (Default) 1h: INT MSG of Device ID 1 is selected 2h: INT MSG of Device ID 2 is selected .....: .... Fh: INT MSG of Device ID 15 is selected

Table 2-55 : UHS-II Device Select Register

2.3.8 UHS-II Device Interrupt Code (Cat.B Offset 0BFh)

This register is effective when **INT MSG Enable** is set to 1 in the *UHS-II Device Select* register. Host Controller holds an INT MSG packet per device. One of INT MSGs (Code length is 1 byte) up to 15 can be read from this register by selecting **UHS-II Device Select**. The number of the registers to hold INT MSGs is determined by **Number of Devices Supported** in the *UHS-II General Capabilities* register. Device ID is supposed to be assigned from 1 sequentially at the UHS-II Initialization.

D07 D00
Device Interrupt Code

Figure 2-46 : UHS-II Device Interrupt Code Register

2.3.9 UHS-II Software Reset (Cat.B Offset 0C1-0C0h)

D15 … D02	D01	D00
Reserved	Host SD-TRAN Reset	Host Full Reset

Figure 2-47 : UHS-II Software Reset Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15-02</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>01</td><td>RWAC</td><td>Host SD-TRAN Reset
Host Driver set this bit to 1 to reset SD-TRAN layer when an abort command or CMD0 is issued to Device or data transfer error occurs. This bit is cleared automatically at completion of SD-TRAN reset. If CMD0 is issued, SD-TRAN Initialization sequence from CMD8 is required to use UHS-II mode. Assuming that bus power is maintained and CM-TRAN Initialization is not required.
<table><tr><td>0h</td><td>Not Affected</td></tr><tr><td>1h</td><td>Reset SD-TRAN</td></tr></table>
Host Controller requires to do followings:
(1) SD Clock Enable is maintained (Continue to provide RCLK).
(2) All setting register is maintained.
(3) Internal sequencers are reset to be able to issue a command.
(4) All Interrupt Status, Status Enable and Signal Enable are cleared.
(5) Data transfer is terminated and data in buffer is discarded.</td></tr>
<tr><td>00</td><td>RWAC</td><td>Host Full Reset
On issuing FULL_RESET CCMD, Host Driver set this bit to 1 to reset Host Controller. This bit is cleared automatically at completion of Host Controller reset. Initialization sequence from PHY Initialization is required to use UHS-II mode. Bus power is supposed to be maintained.
<table><tr><td>0h</td><td>Not Affected</td></tr><tr><td>1h</td><td>Reset Host Controller</td></tr></table>
Host Controller requires to do followings:
(1) SD Clock Enable is cleared (Internal Clock is still synchronized).
(2) All setting register is cleared.
(3) Internal sequencers are reset to just after power on.
(4) All Interrupt Status, Status Enable and Signal Enable are cleared.</td></tr>
</table>

Table 2-56 : UHS-II Software Reset Register

2.3.10 UHS-II Timer Control (Cat.B Offset 0C3-0C2h)

D15 D08	D07 D04	D03 D00
Rsvd	Timeout Counter Value for Deadlock	Timeout Counter Value for CMD_RES

Figure 2-48 : UHS-II Timeout Control Register

Location	Attrib	Register Field Explanation
15-08	Rsvd	**Reserved**
07-04	RW	**Timeout Counter Value for Deadlock** This value determines the deadlock period while host expecting to receive a packet (1 second). Timeout clock frequency will be generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the **Timeout for Deadlock** (in the *UHS-II Error Interrupt Status Enable* register) 1111b: Reserved 1110b: TMCLK x 2^{27} ....: .... 0001b: TMCLK x 2^{14} 0000b: TMCLK x 2^{13}
03-00	RW	**Timeout Counter Value for CMD_RES** This value determines the interval between command packet and response packet (5ms). Timeout clock frequency will be generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the **Timeout for CMD_RES** (in the *UHS-II Error Interrupt Status Enable* register) 1111b: Reserved 1110b: TMCLK x 2^{27} ....: .... 0001b: TMCLK x 2^{14} 0000b: TMCLK x 2^{13}

Table 2-57 : UHS-II Timeout Control Register

2.3.11 UHS-II Error Interrupt Status (Cat.B Offset 0C7-0C4h)

When any of these fields is set to 1, **Error Interrupt** in the *Normal Interrupt Status* register is set to 1.
As described in the UHS-II Addendum, duplicate MSG packets are sent from/to UHS-II card. If either of these packets is recognized as correct one, Host Controller does not assert error interrupt while keeping on data transfer.

D31-27	D26-18	D17	D16	D15	D14-09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error	Reserved	Timeout for Deadlock	Timeout for CMD_RES	ADMA Error	Reserved	EBSY Error	Unrecoverable Error	Reserved	TID Error	Framing Error	CRC Error	Retry Expired	Res Packet Error	Header Error

Figure 2-49 : UHS-II Error Interrupt Status Register

Location	Attrib	Register Field Explanation
31-27	RW1C	**Vendor Specific Error** Vendor may use this field for vendor specific error status. 0h: Interrupt is not generated 1h: Vendor Specific Error
26-18	Rsvd	**Reserved**
17	RW1C	**Timeout for Deadlock** Setting of this bit means that deadlock timeout occurs. Host expects to receive a packet but not received in a specified timeout (1 second). Timeout value is determined by the setting of **Timeout Counter Value for Deadlock** in *UHS-II Timer Control* register. 0h: Interrupt is not generated 1h: Deadlock Error
16	RW1C	**Timeout for CMD_RES** Setting of this bit means that RES Packet timeout occurs. Host expects to receive RES packet but not received in a specified timeout (5ms). Timeout value is determined by the setting of **Timeout Counter Value for CMD_RES** in *UHS-II Timer Control* register. 0h: Interrupt is not generated 1h: RES Packet Timeout Error
15	RW1C	**ADMA Error** Setting of this bit means that ADMA Error occurs in UHS-II mode. The same status is indicated to the **ADMA Error** in the *Error Interrupt Status* register. Host Driver can obtain information of ADMA error from the *ADMA System Address* register and the *ADMA Error Status* register. 0h: Interrupt is not generated 1h: ADMA Error
14-09	Rsvd	**Reserved**
08	RW1C	**EBSY Error** On receiving EBSY packet, if the packet indicates an error, this bit is set to 1. Setting of this bit also sets **Error Interrupt** and **Transfer Completer** together in the *Normal Interrupt Status* register. This error check is effective for a command with setting **EBSY Wait** in the *UHS-II Transfer Mode* register. 0h: Interrupt is not generated 1h: EBSY Error (Backend Error)
07	RW1C	**Unrecoverable Error** Setting of this bit means that Unrecoverable Error is set in a packet from a device. 0h: Interrupt is not generated 1h: Device Unrecoverable Error
06	Rsvd	**Reserved**

05	RW1C	**TID Error** Setting of this bit means that TID Error occurs. 0h: Interrupt is not generated 1h: TID Error
04	RW1C	**Framing Error** Setting of this bit means that Framing Error occurs during a packet receiving. 0h: Interrupt is not generated 1h: Framing Error
03	RW1C	**CRC Error** Setting of this bit means that CRC Error occurs during a packet receiving. 0h: Interrupt is not generated 1h: CRC Error
02	RW1C	**Retry Expired** Setting of this bit means that Retry Counter Expired Error occurs during data transfer. If this bit is set, either **Framing Error** or **CRC Error** in this register shall be set. 0h: Interrupt is not generated 1h: Retry Expired Error
01	RW1C	**RES Packet Error** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If **Response Error Check Enable** is set to1 in the *UHS-II Transfer Mode* register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1. 0h: Interrupt is not generated 1h: RES Packet Error
00	RW1C	**Header Error** Setting of this bit means that Header Error occurs in a received packet. 0h: Interrupt is not generated 1h: Header Error

Table 2-58 : UHS-II Error Interrupt Status Register

2.3.12 UHS-II Error Interrupt Status Enable (Cat.B Offset 0CB-0C8h)

D31-27	D26-18	D17	D16	D15	D14-09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error	Reserved	Timeout for Deadlock	Timeout for CMD_RES	ADMA Error	Reserved	EBSY Error	Unrecoverable Error	Reserved	TID Error	Framing Error	CRC Error	Retry Expired	Res Packet Error	Header Error

Figure 2-50 : UHS-II Error Interrupt Status Enable Register

Location	Attrib	Register Field Explanation
31-27	RW	**Vendor Specific Error** Setting of each bit to 1 enables setting of **Vendor Specific Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
26-18	Rsvd	**Reserved**
17	RW	**Timeout for Deadlock** Setting this bit to 1 enables setting of **Timeout for Deadlock** bit in the *UHS-II Error Interrupt Status* register 0h: Status is Disabled 1h: Status is Enabled
16	RW	**Timeout for CMD_RES** Setting this bit to 1 enables setting of **Timeout for CMD_RES** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
15	RW	**ADMA Error** Setting this bit to 1 enables setting of **ADMA Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
14-09	Rsvd	**Reserved**
08	RW	**EBSY Error** Setting this bit to 1 enables setting of **EBSY Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
07	RW	**Unrecoverable Error** Setting this bit to 1 enables setting of **Unrecoverable Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
06	Rsvd	**Reserved**

05	RW	**TID Error** Setting this bit to 1 enables setting of **TID Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
04	RW	**Framing Error** Setting this bit to 1 enables setting of **Framing Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
03	RW	**CRC Error** Setting this bit to 1 enables setting of **CRC Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
02	RW	**Retry Expired** Setting this bit to 1 enables setting of **Retry Expired** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
01	RW	**RES Packet Error** Setting this bit to 1 enables setting of **RES Packet Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled
00	RW	**Header Error** Setting this bit to 1 enables setting of **Header Error** bit in the *UHS-II Error Interrupt Status* register. 0h: Status is Disabled 1h: Status is Enabled

Table 2-59 : UHS-II Error Interrupt Status Enable Register

2.3.13 UHS-II Error Interrupt Signal Enable (Cat.B Offset 0CF-0CCh)

D31-27	D26-18	D17	D16	D15	D14-09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error	Reserved	Timeout for Deadlock	Timeout for CMD_RES	ADMA Error	Reserved	EBSY Error	Unrecoverable Error	Reserved	TID Error	Framing Error	CRC Error	Retry Expired	Res Packet Error	Header Error

Figure 2-51 : UHS-II Error Interrupt Signal Enable Register

Location	Attrib	Register Field Explanation
31-27	RW	**Vendor Specific Error** Setting of a bit to 1 in this field enables generating interrupt signal when correspondent bit of **Vendor Specific Error** is set in the *UHS-II Error Interrupt Status* register. 0h: Interrupt Signal is Disabled 1h: Interrupt Signal is Enabled
26-18	Rsvd	**Reserved**
17	RW	**Timeout for Deadlock** Setting this bit to 1 enables generating interrupt signal when **Timeout for Deadlock** bit is set in the *UHS-II Error Interrupt Status* register. 0h: Interrupt Signal is Disabled 1h: Interrupt Signal is Enabled
16	RW	**Timeout for CMD_RES** Setting this bit to 1 enables generating interrupt signal when **Timeout for CMD_RES** bit is set in the *UHS-II Error Interrupt Status* register. 0h: Interrupt Signal is Disabled 1h: Interrupt Signal is Enabled
15	RW	**ADMA Error** Setting this bit to 1 enables generating interrupt signal when **ADMA Error** bit is set in the *UHS-II Error Interrupt Status* register. 0h: Interrupt Signal is Disabled 1h: Interrupt Signal is Enabled
14-09	Rsvd	**Reserved**
08	RW	**EBSY Error** Setting this bit to 1 enables generating interrupt signal when **EBSY Error** bit is set in the *UHS-II Error Interrupt Status* register. 0h: Interrupt Signal is Disabled 1h: Interrupt Signal is Enabled
07	RW	**Unrecoverable Error** Setting this bit to 1 enables generating interrupt signal when **Unrecoverable Error** bit is set in the *UHS-II Error Interrupt Status* register. 0h: Interrupt Signal is Disabled 1h: Interrupt Signal is Enabled

<table>
<tr><td>06</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>05</td><td>RW</td><td>TID Error
Setting this bit to 1 enables generating interrupt signal when TID Error bit is set in the UHS-II Error Interrupt Status register.<table><tr><td>0h</td><td>Interrupt Signal is Disabled</td></tr><tr><td>1h</td><td>Interrupt Signal is Enabled</td></tr></table></td></tr>
<tr><td>04</td><td>RW</td><td>Framing Error
Setting this bit to 1 enables generating interrupt signal when Framing Error bit is set in the UHS-II Error Interrupt Status register.<table><tr><td>0h</td><td>Interrupt Signal is Disabled</td></tr><tr><td>1h</td><td>Interrupt Signal is Enabled</td></tr></table></td></tr>
<tr><td>03</td><td>RW</td><td>CRC Error
Setting this bit to 1 enables generating interrupt signal when CRC Error bit is set in the UHS-II Error Interrupt Status register.<table><tr><td>0h</td><td>Interrupt Signal is Disabled</td></tr><tr><td>1h</td><td>Interrupt Signal is Enabled</td></tr></table></td></tr>
<tr><td>02</td><td>RW</td><td>Retry Expired
Setting this bit to 1 enables generating interrupt signal when Retry Expired bit is set in the UHS-II Error Interrupt Status register.<table><tr><td>0h</td><td>Interrupt Signal is Disabled</td></tr><tr><td>1h</td><td>Interrupt Signal is Enabled</td></tr></table></td></tr>
<tr><td>01</td><td>RW</td><td>RES Packet Error
Setting this bit to 1 enables generating interrupt signal when RES Packet Error bit is set in the UHS-II Error Interrupt Status register.<table><tr><td>0h</td><td>Interrupt Signal is Disabled</td></tr><tr><td>1h</td><td>Interrupt Signal is Enabled</td></tr></table></td></tr>
<tr><td>00</td><td>RW</td><td>Header Error
Setting this bit to 1 enables generating interrupt signal when Header Error bit is set in the UHS-II Error Interrupt Status register.<table><tr><td>0h</td><td>Interrupt Signal is Disabled</td></tr><tr><td>1h</td><td>Interrupt Signal is Enabled</td></tr></table></td></tr>
</table>

Table 2-60 : UHS-II Error Interrupt Signal Enable Register

2.3.14 Pointer Registers to mFFh-100h Area

Area of offset mFFh-100h (m=1, 2, 3, ... E, F) is defined as re-locatable area. The locations of following register sets are pointed by offset address. Pointer Registers are listed up in Table 2-61 and format of Pointer Registers is indicated in Figure 2-52. Pointers shall be larger than or equal to 100h.

Offset	Attrib	Pointer Name
0E1-0E0h	HwInit	Pointer for UHS-II Settings
0E3-0E2h	HwInit	Pointer for UHS-II Host Capabilities
0E5-0E4h	HwInit	Pointer for UHS-II Test
0E7-0E6h	HwInit	Pointer for Embedded Control
0E9-0E8h	HwInit	Pointer for Vendor Specific Area
0EB-0EAh	HwInit	Reserved: Pointer for Specific Control
0ED-0ECh	HwInit	Reserved
0EF-0EEh	HwInit	Reserved

Table 2-61 : Pointer Registers for mFF-100h Area

D15 D12	D11 D00
Reserved (all 0)	Register Offset Address

Figure 2-52 : Register format of Pointer Register

The Pointer for Vendor Specific Area may be used to extend vendor specific functions.
0EB-0EAh is reserved for a pointer for specific control, which will be clarified in a later version.

2.3.15 Slot Interrupt Status Register (Cat.C Offset 0FCh)

D15 D08	D07 D00
Rsvd	Interrupt Signal For Each Slot

Figure 2-53 : Slot Interrupt Status Register

Location	Attrib	Register Field Explanation
15-08	Rsvd	**Reserved**
07-00	ROC	**Interrupt Signal For Each Slot** These status bits indicate the logical OR of Interrupt Signal and Wakeup Signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can know which interrupt is generated by reading these status bits. By a power on reset or by setting **Software Reset For All**, the interrupt signal shall be de-asserted and this status shall read 00h. Bit 00 — Slot 1 Bit 01 — Slot 2 Bit 02 — Slot 3 .... — .... Bit 07 — Slot 8

Table 2-62 : Slot Interrupt Status Register

2.3.16 Host Controller Version Register (Cat.C Offset 0FEh)

D15 D08	D07 D00
Vendor Version Number	Specification Version Number

Figure 2-54 : Host Controller Version Register

Location	Attrib	Register Field Explanation
15-08	HwInit	**Vendor Version Number** This status is reserved for the vendor version number. The Host Driver should not use this status.
07-00	HwInit	**Specification Version Number** This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version. 00h: SD Host Controller Specification Version 1.00 01h: SD Host Controller Specification Version 2.00 02h: SD Host Controller Specification Version 3.00 03h: SD Host Controller Specification Version 4.00 04h: SD Host Controller Specification Version 4.10 05h: SD Host Controller Specification Version 4.20 others: Reserved

Table 2-63 : Host Controller Version

2.4 UHS-II Registers in 100-1FFh

This area is defined as re-locatable area. Pointer of each entry is specified by pointers in 0E0-0EFh. Following registers can be allocated anywhere and any order in 1FF-100h area.

2.4.1 UHS-II Settings (Cat.B 16 Bytes)

There are three types of UHS-II Settings registers. Start address of General Settings is pointed by *Pointer for UHS-II Setting* register.

Length	UHS-II Setting Registers
4 bytes	General Settings
4 bytes	PHY Settings
8 bytes	LINK / TRAN Settings

Table 2-64 : UHS-II Settings Registers

2.4.1.1 UHS-II General Settings (4 Bytes)

D31 D12	D11 D08	D07 D01	D00
Reserved	Number of Lanes and Functionalities	Reserved	Power Mode

Figure 2-55 : UHS-II General Settings Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>31-12</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>11-08</td><td>RW</td><td>Number of Lanes and Functionalities
The lane configuration of a Host System is set to this field depends on the capability among Host Controller and connected devices. 2 Lanes FD mode is mandatory and the others modes are optional.
<table>
<tr><td>0000b</td><td>2 Lanes FD or 2L-HD</td></tr>
<tr><td>0001b</td><td>Not Used</td></tr>
<tr><td>0010b</td><td>3 Lanes 2D1U-FD (Embedded)</td></tr>
<tr><td>0011b</td><td>3 Lanes 1D2U-FD (Embedded)</td></tr>
<tr><td>0100b</td><td>4 Lanes 2D2U-FD (Embedded)</td></tr>
<tr><td>others</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>07-01</td><td>RW</td><td>Reserved</td></tr>
<tr><td>00</td><td>RW</td><td>Power Mode
This field determines either Fast Mode or Low Power Mode. Host and all devices connected to the host shall be set to the same mode.
<table>
<tr><td>0</td><td>Fast Mode</td></tr>
<tr><td>1</td><td>Low Power Mode</td></tr>
</table></td></tr>
</table>

Table 2-65 : UHS-II General Settings Register

2.4.1.2 UHS-II PHY Settings (4 Bytes)

D31 D24	D23 D20	D19 D16	D15	D14 D08	D07 D06	D05 D00
Reserved	Host N_LSS_DIR	Host N_LSS_SYN	Hibernate Enable	Reserved	Speed Range	Reserved

Figure 2-56 : UHS-II PHY Settings Register

Location	Attrib	Register Field Explanation
31-24	Rsvd	**Reserved**
23-20	RW	**Host N_LSS_DIR** The largest value of N_LSS_DIR capabilities among the Host Controller and connected devices is set to this field. 0h: 8 x16 LSS 1h: 8 x 1 LSS 2h: 8 x 2 LSS 3h: 8 x 3 LSS ......: ...... Fh: 8 x 15 LSS
19-16	RW	**Host N_LSS_SYN** The largest value of N_LSS_SYN capabilities among the Host Controller and connected devices is set to this field. 0h: 4 x16 LSS 1h: 4 x 1 LSS 2h: 4 x 2 LSS 3h: 4 x 3 LSS ......: ...... Fh: 4 x 15 LSS
15	RW	**Hibernate Enable** After checking card capability of Hibernate mode, if all devices support Hibernate mode, this bit may be set. This bit determines whether Host remains in Dormant state or goes to Hibernate state. In Hibernate mode, VDD1 Power may be off. 0: Hibernate Disabled 1: Hibernate Enabled
14-08	Rsvd	**Reserved**
07-06	RW	**Speed Range** PLL Multiplier is selected by this field. Change of PLL Multiplier is not effective immediately and is applied from exiting Dormant State. 00b: Range A (Default) 01b: Range B 10b: Reserved 11b: Reserved
05-00	RW	**Reserved**

Table 2-66 : UHS-II PHY Settings Register

2.4.1.3 UHS-II LINK/TRAN Settings (8 Bytes)

D63 D40	D39 D32	D31 D18	D17 D16	D15 D08	D07 D00
Reserved	Host N_DATA_GAP	Reserved	Retry Count	Host N_FCU	Reserved

Figure 2-57 : UHS-II LINK/TRAN Settings Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>63-40</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>39-32</td><td>RW</td><td>Host N_DATA_GAP
The largest value of N_DATA_GAP capabilities among the Host Controller and connected card and devices is set to this field.
<table><tr><td>00h</td><td>No Gap</td></tr><tr><td>01h</td><td>1 LSS</td></tr><tr><td>02h</td><td>2 LSS</td></tr><tr><td>03h</td><td>3 LSS</td></tr><tr><td>......</td><td>......</td></tr><tr><td>FFh</td><td>255 LSS</td></tr></table></td></tr>
<tr><td>31-18</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>17-16</td><td>RW</td><td>Retry Count
DATA Burst retry count is set to this field.
<table><tr><td>00b</td><td>Retry Disabled</td></tr><tr><td>01b</td><td>1 time</td></tr><tr><td>10b</td><td>2 times</td></tr><tr><td>11b</td><td>3 times</td></tr></table></td></tr>
<tr><td>15-08</td><td>RW</td><td>Host N_FCU
Host Driver sets the number of blocks in Data Burst (Flow Control) to this field. The value shall be smaller than or equal to N_FCU capabilities among the Host Controller and connected card and devices. Setting 1 to 4 blocks is recommended considering buffer size.
<table><tr><td>00h</td><td>256 Blocks</td></tr><tr><td>01h</td><td>1 Block</td></tr><tr><td>02h</td><td>2 Blocks</td></tr><tr><td>03h</td><td>3 Blocks</td></tr><tr><td>......</td><td>......</td></tr><tr><td>FFh</td><td>255 Blocks</td></tr></table></td></tr>
<tr><td>07-00</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

Table 2-67 : UHS-II LINK/TRAN Settings Register

2.4.2 UHS-II Host Capabilities (Cat.B 16 Bytes)

There are three types of *UHS-II Host Capabilities* registers. Start address of General Capabilities is pointed by *Pointer for UHS-II Host Capabilities* register.

Length	UHS-II Host Capabilities Registers
4 bytes	General Capabilities
4 bytes	PHY Capabilities
8 bytes	LINK / TRAN Capabilities

Table 2-68 : UHS-II Host Capabilities Registers

2.4.2.1 UHS-II General Capabilities (4 Bytes)

D31 D24	D23 D22	D21 D18	D17 D16	D15	D14	D13 D08	D07 D04	D03 D00
Reserved	Bus Topology	Number of Devices Supported	Removable / Embedded	Boot Code Loading	64-bit Addressing	Number of Lanes and Functionalities	GAP	DAP

Figure 2-58 : UHS-II General Capabilities Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>31-24</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>23-22</td><td>HwInit</td><td>Bus Topology
This field indicates one of bus topologies configured by a Host System.
<table><tr><td>00b</td><td>P2P Connection</td></tr><tr><td>01b</td><td>Ring Connection</td></tr><tr><td>10b</td><td>HUB Connection</td></tr><tr><td>11b</td><td>HUB is connected in Ring</td></tr></table></td></tr>
<tr><td>21-18</td><td>HwInit</td><td>Number of Devices Supported
This field indicates the maximum number of devices supported by the Host Controller.
<table><tr><td>0h</td><td>Not used</td></tr><tr><td>1h</td><td>1 Devices</td></tr><tr><td>2h</td><td>2 Devices</td></tr><tr><td>.....</td><td>.......</td></tr><tr><td>Fh</td><td>15 Devices</td></tr></table></td></tr>
<tr><td>17-16</td><td>HwInit</td><td>Removable / Embedded
This field indicates device type configured by a Host System.
<table><tr><td>00b</td><td>Removable Card (P2P)</td></tr><tr><td>01b</td><td>Embedded Devices</td></tr><tr><td>10b</td><td>Embedded Devices + Removable Card</td></tr><tr><td>11b</td><td>Reserved</td></tr></table></td></tr>
<tr><td>15</td><td>Rsvd</td><td>Boot Code Loading
This field indicates whether Host Controller tries to boot system in UHS-II mode. If this bit is set to 1, BSYN LSS is send at the PHY Initialization.
(How to execute Boot Code Loading will be described later version.)
<table><tr><td>0</td><td>No Boot Code Loading</td></tr><tr><td>1</td><td>Execute Boot Code Loading</td></tr></table></td></tr>
</table>

14	HwInit	**64-bit Addressing** This field indicates support of 64-bit addressing by the Host Controller. 00b: 32-bit Addressing is supported 01b: 32-bit and 64-bit Addressing is supported
13-08	HwInit	**Number of Lanes and Functionalities** This field indicates support of lanes by the Host Controller. 0 mean not supported and 1 means supported. D08: 2L-HD D09: 2D1U-FD D10: 1D2U-FD D11: 2D2U-FD D12: Reserved D13: Reserved
07-04	HwInit	**GAP (Group Allocation Power)** This field indicates the maximum capability of host power supply for a group configured by a Host System. This field is used to set the argument of DEVICE_INIT CCMD. 0h: Not used 1h: 360 mW 2h: 720 mW .....: .... Fh: 360 x15 mW
03-00	HwInit	**DAP (Device Allocation Power)** This field indicates the maximum capability of host power supply for a device configured by a Host System. This field is used to set the argument of DEVICE_INIT CCMD. 0h: 360 mW (Default) 1h: 360 mW 2h: 720 mW .....: .... Fh: 360 x15 mW

Table 2-69 : UHS-II General Capabilities Register

2.4.2.2 UHS-II PHY Capabilities (4 Bytes)

D31 D24	D23 D20	D19 D16	D15 D08	D07 D06	D05 D00

Reserved	Host N_LSS_DIR	Host N_LSS_SYN	Reserved	Speed Range	PHY Revision

Figure 2-59 : UHS-II PHY Capabilities Register

Location	Attrib	Register Field Explanation
31-24	Rsvd	**Reserved**
23-20	HwInit	**Host N_LSS_DIR** This field indicates the minimum N_LSS_DIR required by the Host Controller. 0h: 8 x16 LSS 1h: 8 x 1 LSS 2h: 8 x 2 LSS 3h: 8 x 3 LSS ......: ...... Fh: 8 x 15 LSS
19-16	HwInit	**Host N_LSS_SYN** This field indicates the minimum N_LSS_SYN required by the Host Controller. 0h: 4 x16 LSS 1h: 4 x 1 LSS 2h: 4 x 2 LSS 3h: 4 x 3 LSS ......: ...... Fh: 4 x 15 LSS
15-08	Rsvd	**Reserved**
07-06	HwInit	**Speed Range** This field indicates supported Speed Range by the Host Controller. 00b: Range A (Default) 01b: Range A and Range B 10b: Reserved 11b: Reserved
05-00	HwInit	**PHY Revision** This field indicates PHY Revision number. 000000b: The first revision others: Reserved

Table 2-70 : UHS-II PHY Capabilities Register

2.4.2.3 UHS-II LINK/TRAN Capabilities (8 Bytes)

D63 D40	D39 D32	D31 D20	D19	D18 D16	D15 D08	D07 D06	D05 D00

Reserved	Host N_DATA_GAP	Host Maximum Block Length	Reserved	Host Device Type	Host N_FCU	Reserved	LINK Revision

Figure 2-60 : UHS-II LINK/TRAN Capabilities Register

Location	Attrib	Register Field Explanation
63-40	Rsvd	**Reserved**
39-32	HwInit	**Host N_DATA_GAP** This field indicates the minimum number of data gap (DIDL) supported by the Host Controller. 00h | No Gap 01h | 1 LSS 02h | 2 LSS 03h | 3 LSS ...... | ...... FFh | 255 LSS
31-20	HwInit	**Host Maximum Block Length** This field indicates the maximum block length of the Host Controller. 000h | Not Used 001h | 1 byte 002h | 2 bytes ...... | ...... 200h | 512 bytes ...... | ...... 800h | 2048 bytes 801h-FFFh | Not Used
19	Rsvd	**Reserved**
18-16	HwInit	**Host Device Type** This field shall be fixed to 000b. 000b | Host Controller others | Not Used
15-08	HwInit	**Host N_FCU** This field indicates the maximum number of blocks supported in a flow control unit by the Host Controller. This value is determined by supported buffer size. 00h | 256 Blocks 01h | 1 Block 02h | 2 Blocks 03h | 3 Blocks ...... | ...... FFh | 255 Blocks
07-06	Rsvd	**Reserved**

05-00	HwInit	**LINK Revision** This field indicates LINK Revision number. 000000b — The first revision others — Reserved

Table 2-71 : UHS-II LINK/TRAN Capabilities Register

2.4.3 UHS-II Test Register (Cat.B 4 Bytes)

2.4.3.1 Force Event for UHS-II Error Interrupt Status

D31 D27	D26 D18	D17	D16	D15	D14 D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Vendor Specific Error	Reserved	Timeout for Deadlock	Timeout for CMD_RES	ADMA Error	Reserved	EBSY Error	Unrecoverable Error	Reserved	TID Error	Framing Error	CRC Error	Retry Expired	Res Packet Error	Header Error

Figure 2-61 : Force Event for UHS-II Error Interrupt Status Register

Location	Attrib	Register Field Explanation
31-27	WO	**Force Event for Vendor Specific Error** 0h — Not Affected 1h — Vendor Specific Error Status is set
26-18	Rsvd	**Reserved**
17	WO	**Force Event for Timeout for Deadlock** Setting this bit forces the Host Controller to set **Timeout for Deadlock** in the *UHS-II Error Interrupt Status* register. 0h — Not Affected 1h — Timeout for Deadlock Error Status is set
16	WO	**Force Event for Timeout for CMD_RES** Setting this bit forces the Host Controller to set **Timeout for CMD_RES** in the *UHS-II Error Interrupt Status* register. 0h — Not Affected 1h — Timeout for CMD_RES Status is set
15	WO	**Force Event for ADMA Error** Setting this bit forces the Host Controller to set **ADMA Error** in the *UHS-II Error Interrupt Status* register. 0h — Not Affected 1h — ADMA Error Status is set
14-09	Rsvd	**Reserved**
08	WO	**Force Event for EBSY Error** Setting this bit forces the Host Controller to set **EBSY Error** in the *UHS-II Error Interrupt Status* register. 0h — Not Affected 1h — EBSY Error Status is set

07	WO	**Force Event for Unrecoverable Error** Setting this bit forces the Host Controller to set **Unrecoverable Error** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: Unrecoverable Error Status is set
06	Rsvd	**Reserved**
05	WO	**Force Event for TID Error** Setting this bit forces the Host Controller to set **TID Error** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: TID Error Status is set
04	WO	**Force Event for Framing Error** Setting this bit forces the Host Controller to set **Framing Error** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: Framing Error Status is set
03	WO	**Force Event for CRC Error** Setting this bit forces the Host Controller to set **CRC Error** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: CRC Error Status is set
02	WO	**Force Event for Retry Expired** Setting this bit forces the Host Controller to set **Retry Expired** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: Retry Expired Error Status is set
01	WO	**Force Event for RES Packet Error** Setting this bit forces the Host Controller to set **RES Packet Error** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: RES Packet Error Status is set
00	WO	**Force Event for Header Error** Setting this bit forces the Host Controller to set **Header Error** in the *UHS-II Error Interrupt Status* register. 0h: Not Affected 1h: Header Error Status is set

Table 2-72 : Force Event for UHS-II Error Interrupt Status Register

2.4.4 Embedded Control Register (Cat.C 4 Bytes)

This register is optional. Controlling the embedded devices is beyond the scope of the Standard Host Driver because embedded configuration depends on a host system and then the embedded devices should be controlled by a specific driver of a host system.

D31	D30 D24	D23	D22 D20	D19	D18 D16	D15	D14 D08	D07 D06	D05 D04	D03	D02 D00
Rsvd	Back-End Power Control	Rsvd	Interrupt Pin Select	Rsvd	Clock Pin Select	Rsvd	Bus Width Preset	Rsvd	Number of Interrupt Input Pins	Rsvd	Number of Clock Pins

Figure 2-62 : Embedded Control Register

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>31</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>30-24</td><td>RW</td><td>Back-End Power Control (SD Mode)
Each bit of this field controls back-end power supply for an embedded device. Host interface voltage (VDDH) is not controlled by this field. The number of devices supported is specified by Number of Clock Pins and a maximum of 7 devices can be controlled.
<table>
<tr><td>D24</td><td>Back-end Power Control for Device 1</td></tr>
<tr><td>D25</td><td>Back-end Power Control for Device 2</td></tr>
<tr><td>D26</td><td>Back-end Power Control for Device 3</td></tr>
<tr><td>D27</td><td>Back-end Power Control for Device 4</td></tr>
<tr><td>D28</td><td>Back-end Power Control for Device 5</td></tr>
<tr><td>D29</td><td>Back-end Power Control for Device 6</td></tr>
<tr><td>D30</td><td>Back-end Power Control for Device 7</td></tr>
</table>
The function of each bit is defined as follows:
<table>
<tr><td>0</td><td>Back-end Power is Off</td></tr>
<tr><td>1</td><td>Back-end Power is Supplied</td></tr>
</table>
Back-End power control is effective for embedded memory devices in the Sleep State that support the Sleep command (CMD14) to reduce power consumption and embedded SDIO devices when IOEx is set to 0.</td></tr>
<tr><td>23</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>22-20</td><td>RW</td><td>Interrupt Pin Select
Interrupt pin inputs are enabled by this field. Enable of unsupported interrupt pin is meaningless.
<table>
<tr><td>000b</td><td>INT_A, INT_B and INT_C are disabled</td></tr>
<tr><td>xx1b</td><td>INT_A is Enabled</td></tr>
<tr><td>x1xb</td><td>INT_B is Enabled</td></tr>
<tr><td>1xxb</td><td>INT_C is Enabled</td></tr>
</table></td></tr>
<tr><td>19</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

<table>
<tr><td>18-16</td><td>RW</td><td>Clock Pin Select (SD Mode)
One of clock pin outputs is selected by this field. Select of unsupported clock pin is meaningless. Refer to Figure 2-63 for the timing of clock outputs.

<table><tr><td>000b</td><td>Clock Pins are Disabled</td></tr><tr><td>001b</td><td>CLK[1] is Selected</td></tr><tr><td>010b</td><td>CLK[2] is Selected</td></tr><tr><td>......</td><td>..........</td></tr><tr><td>111b</td><td>CLK[7] is Selected</td></tr></table></td></tr>
<tr><td>15</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>14-08</td><td>HwInit</td><td>Bus Width Preset (SD Mode)
Shared bus supports mixing of 4-bit and 8-bit bus width devices.
Each bit of this field specifies the bus width for each embedded device. The number of devices supported is specified by Number of Clock Pins and a maximum of 7 devices are supported.
This field is effective when multiple devices are connected to a shared bus (Slot Type is set to 10b in the Capabilities register). In the other case, Extended Data Transfer Width in the Host Control 1 register is used to select 8-bit bus width. As use of 1-bit mode is not intended for shared bus, Data Transfer Width in the Host Control 1 register should be set to 1.

<table><tr><td>D08</td><td>Bus Width Preset for Device 1</td></tr><tr><td>D09</td><td>Bus Width Preset for Device 2</td></tr><tr><td>D10</td><td>Bus Width Preset for Device 3</td></tr><tr><td>D11</td><td>Bus Width Preset for Device 4</td></tr><tr><td>D12</td><td>Bus Width Preset for Device 5</td></tr><tr><td>D13</td><td>Bus Width Preset for Device 6</td></tr><tr><td>D14</td><td>Bus Width Preset for Device 7</td></tr></table>
The function of each bit is defined as follows:<table><tr><td>0</td><td>4-bit bus width mode (Data Transfer Width = 1)</td></tr><tr><td>1</td><td>8-bit bus width mode</td></tr></table></td></tr>
<tr><td>07-06</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>05-04</td><td>HwInit</td><td>Number of Interrupt Input Pins
This field indicates support of interrupt input pins for embedded system.
Three asynchronous interrupt pins are defined, INT_A#, INT_B# and INT_C#. Which interrupt pin is used is determined by the system. Each one is driven by open drain and then wired or connection is possible.

<table><tr><td>00b</td><td>Interrupt Input Pin is Not Supported</td></tr><tr><td>01b</td><td>INTA is Supported</td></tr><tr><td>10b</td><td>INTA and INTB are Supported</td></tr><tr><td>11b</td><td>INTA, INTB and INTC are Supported</td></tr></table></td></tr>
<tr><td>03</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

02-00	HwInit	**Number of Clock Pins (SD Mode)** This field indicates support of clock pins to select one of devices for shared bus system. Up to 7 clock pins can be supported. Shared bus is supported by specific system. Then Standard Host Driver does not support control of these clock pins. 000b: Shared bus is not supported 001b: 1 SDCLK pin is supported 010b: 2 SDCLK pins are supported .....: .... 111b: 7 SDCLK pins are supported

Table 2-73 : Embedded Control Register

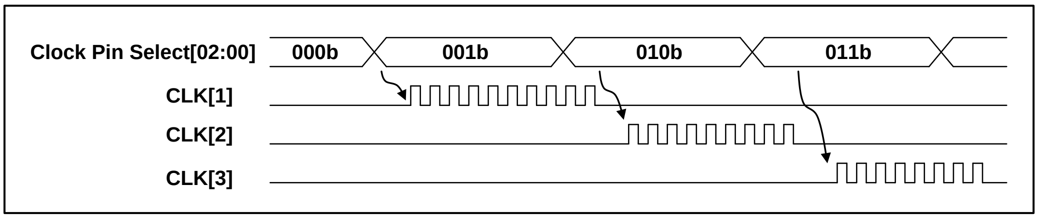


Figure 2-63 : An Example Timing of Selecting Clock Pin

Figure 2-63 shows an example timing of selecting clock pin. When **Clock Pin Select** is set to 000b, no clocks are generated from ***CLK[7:0]*** pins. When **Clock Pin Select** is set to 001b, Host Controller provides clock to only ***CLK[1]***. This means the device 1 is selected. By setting 010b to **Clock Pin Select,** device 2 is selected. ***CLK[1]*** is stopped and then ***CLK[2]*** is generated. By setting 011b to **Clock Pin Select,** device 3 is selected. ***CLK[2]*** is stopped and then ***CLK[3]*** is generated. Clock outputs shall not be overlapped and no glitch shall be included. Clock frequency and output buffer strength are changed during clock stop interval.

Implementation Notes:
Lock-Reset pin is defined by eSD Version 2.10 to improve security level for updating boot loader and system codes. The Host System supported Lock-Reset needs to consider following points.
Driving Lock-Reset to high during the device power off is not allowed to prevent protection diode of Lock-Reset input buffer from destroying. Lock-Rest signal can be driven to high after power is supplied to the device. Host interface voltage (VDDH) should be always supplied to the device. Then Host System should implement so that VDDH is not controlled by SD Bus Power bit in the Power Control register.

In case of eSD, sleep mode should be used instead of eSD device power off for saving power. Memory voltage VDDF can be off when the device is in Sleep State.
In case of SDIO, embedded SDIO can support back-end function power pin. The back-end function power can be off while IOEx=0.

3. SEQUENCE

This section defines basic sequence flow chart divided into several sub sequences.

"Wait for interrupts" is used in the flow chart. This means the Host Driver waits until specified interrupts are asserted. If already asserted, then fall through that step in the flow chart. Timeout checking shall be always required to detect no interrupt generated but this is not described in the flow chart.

This specification uses the double box like Figure 3-1 (e.g., the step (1) in Figure 3-5 and the step (5) in Figure 3-26). It means that the other flows, which already are shown, shall be referred.

Figure 3-1 : Double Box Notation

3.1 SD Card Detection

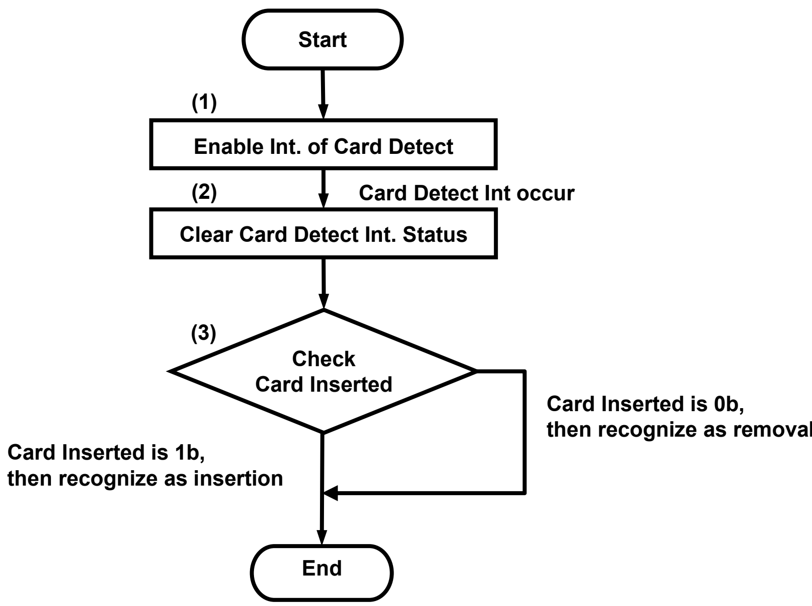


Figure 3-2: SD Card Detect Sequence

The flow chart for detecting an SD card is shown in Figure 3-2. Each step is executed as follows:

To enable interrupt for card detection, write 1 to the following bits:

Card Insertion Status Enable in the *Normal Interrupt Status Enable* register
Card Insertion Signal Enable in the *Normal Interrupt Signal Enable* register
Card Removal Status Enable in the *Normal Interrupt Status Enable* register
Card Removal Signal Enable in the *Normal Interrupt Signal Enable* register

(1) When the Host Driver detects the card insertion or removal, clear its interrupt statuses. If **Card Insertion** interrupt is generated, write 1 to **Card Insertion** in the *Normal Interrupt Status* register. If **Card Removal** interrupt is generated, write 1 to **Card Removal** in the *Normal Interrupt Status* register.
(2) Check **Card Inserted** in the *Present State* register. In the case where **Card Inserted** is 1, the Host Driver can supply the power and the clock to the SD card. In the case where **Card Inserted** is 0, the other executing processes of the Host Driver shall be immediately closed.

If miniSD adaptor is used for standard SD slot and miniSD card is inserted or extracted from the adaptor, card detect interrupt may not be generated. When host does not receive response to any commands, miniSD card is extracted or in idle state, the host should try to re-initialize the card. In this case, all card information shall be re-loaded.

3.2 SD Clock Control

3.2.1 Internal Clock Setup Sequence

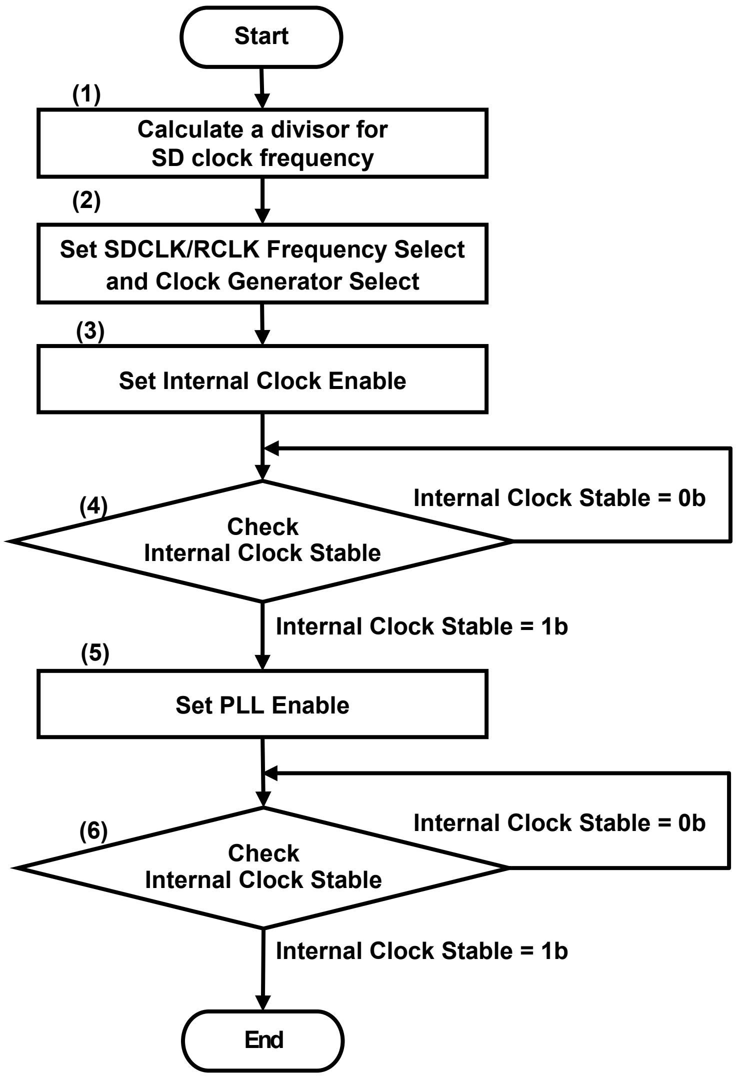


Figure 3-3: Internal Clock Setup Sequence

The sequence for supplying SD Clock to an SD card is described in Figure 3-3. From Version 4.10, **PLL Enable** is added. This sequence is also applicable to prior versions which do not support **PLL Enable**.

(1) Calculate a divisor to determine SD Clock frequency for legacy IF or RCLK frequency for UHS-II IF by reading **Base Clock Frequency For SD Clock** and **Clock Multiplier** in the *Capabilities* register. If non-zero value is set to **Clock Multiplier**, Programmable Clock Mode can be used. If **Base Clock Frequency For SD Clock** is 00 0000b, the Host System shall provide this information to the Host Driver by another method.

(2) Set **SDCLK/RCLK Frequency Select** and **Clock Generator Select** in the *Clock Control* register in accordance with the calculated result of step (1).
If **Preset Value Enable** is set in the *Host Control 2* register, these bits are set by Host Controller

automatically as specified in the Preset Value register.
(3) Set **Internal Clock Enable** in the *Clock Control* register.
(4) Check **Internal Clock Stable** in the *Clock Control* register. Repeat this step until this status is 1. Clock will be stable in shorter time but timeout of this loop is defined as 150ms.
(5) Set **PLL Enable** in the *Clock Control* register. This step does not affect Host Controllers which do not support **PLL Enable**.
(6) If **PLL Enable** is supported, PLL locked may be checked by this status. (If **PLL Enable** is not supported, this status is supposed to indicate 1 by step (3)). Clock will be stable in shorter time but timeout of this loop is defined as 150ms.

3.2.2 SD Clock Supply and Stop Sequence

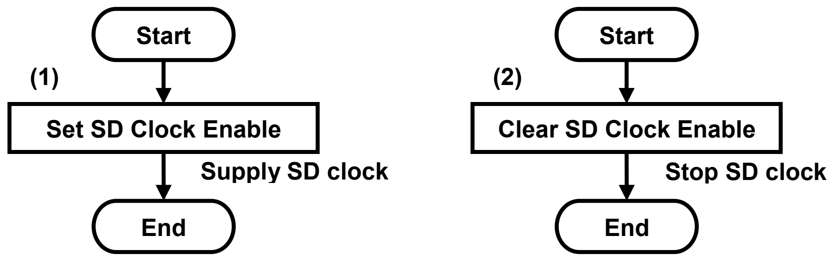


Figure 3-4: SD Clock Supply and Stop Sequence

The flow chart for stopping the SD Clock is shown in Figure 3-4.

(1) Set **SD Clock Enable** in the *Clock Control* register to 1. Then, the Host Controller starts supplying the SD Clock.
(2) Set **SD Clock Enable** in the *Clock Control* register to 0. Then, the Host Controller stops supplying the SD Clock. Internal Clock is still oscillating.

The Host Driver shall not clear **SD Clock Enable** while an SD transaction is executing on the SD Bus -- namely, while either **Command Inhibit (DAT)** or **Command Inhibit (CMD)** in the *Present State* register is set to 1.
During read operation in SD mode, Host Controller may stop data transfer through stopping ***SDCLK*** regardless of **SD Clock Enable**. Stopping clock while SD Bus is not in use (e.g., data block gap) is recommended.
SDCLK/***RCLK*** output is controlled with **SD Clock Enable**. Host Controller should be implemented so that clock output latency is small. It can be achieved by stabling oscillation of internal clocks of the Host Controller before enabling output of ***SDCLK***/***RCLK*** according to the setup sequence described in Figure 3-3.

3.2.3 SD Clock Frequency Change Sequence

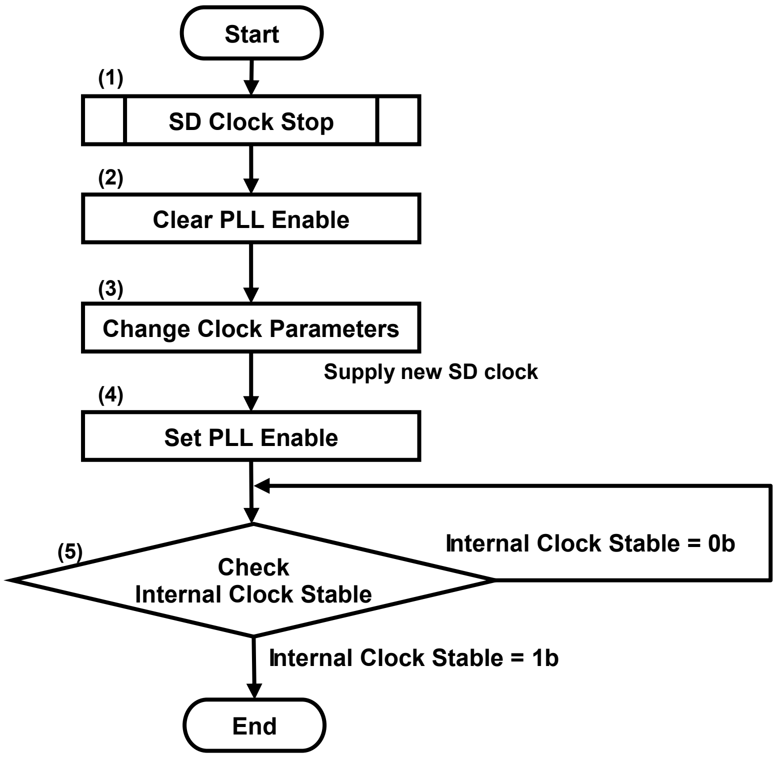


Figure 3-5: SD Clock Change Sequence

The sequence for changing SD Clock frequency is shown in Figure 3-5.

(1) Execute the SD Clock Stop Sequence. If SD Clock supply to card is already stopped, skip this step.
(2) Clear PLL Enable to 0. If **PLL Enable** is not supported, this step has no effect and may skip.
(3) When **Preset Value Enable** in the *Host Control 2* register is set to 0, Host Driver changes clock parameters in the Clock Control register. When **Preset Value Enable** is set to 1, preset clock is selected according to Bus Speed Mode as described in Table 2-44.
(4) Set PLL Enable to 1. If **PLL Enable** is not supported, this step has no effect and may skip.
(5) Wait until **Internal Clock Stable** is set to 1. Clock will be stable in shorter time but timeout of this loop is defined as 150ms. SD Clock Supply Sequence is required to provide clock to device.

3.3 SD Bus Power Control

The sequence for controlling the SD Bus Power is described in Figure 3-6.

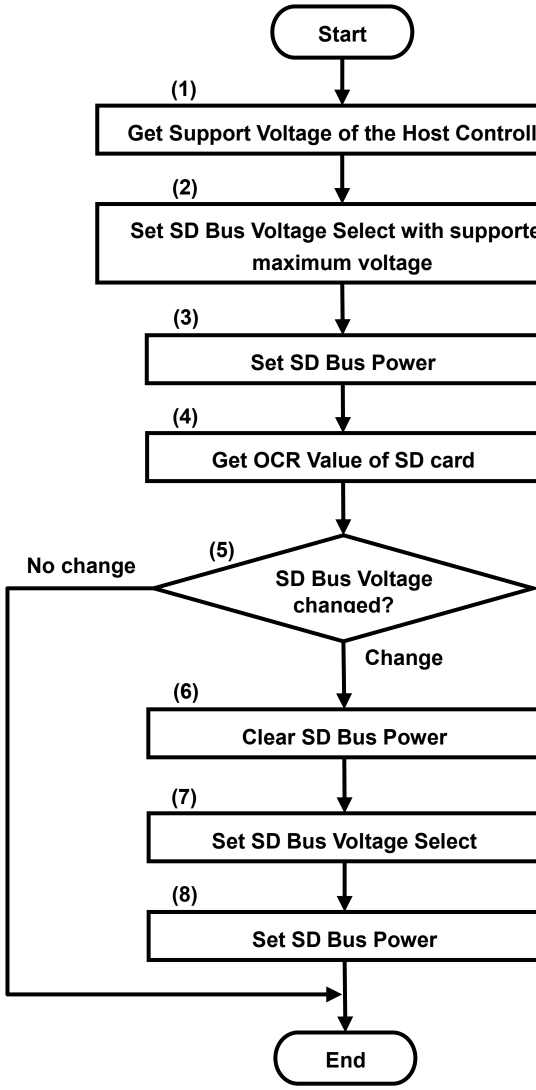


Figure 3-6: SD Bus Power Control Sequence

(1) By reading the *Capabilities* register, get the support voltage of the Host Controller.
(2) Set **SD Bus Voltage Select** in the *Power Control* register with maximum voltage that the Host Controller supports.
(3) Set **SD Bus Power** in the *Power Control* register to 1.
(4) Get the OCR value of all function internal of SD card.
(5) Judge whether SD Bus voltage needs to be changed or not. In case where SD Bus voltage needs to be changed, go to step (6). In case where SD Bus voltage does not need to be changed, go to 'End'.
(6) Set **SD Bus Power** in the *Power Control* register to 0 for clearing this bit. The card requires voltage rising from 0 volt to detect it correctly. The Host Driver shall clear **SD Bus Power** before changing voltage by setting **SD Bus Voltage Select**.
(7) Set **SD Bus Voltage Select** in the *Power Control* register.
(8) Set **SD Bus Power** in the *Power Control* register to 1.

Note:
Step (2) and step (3) can be executed at same time. Also, step (7) and step (8) can be executed at same time.

3.4 Changing Bus Width

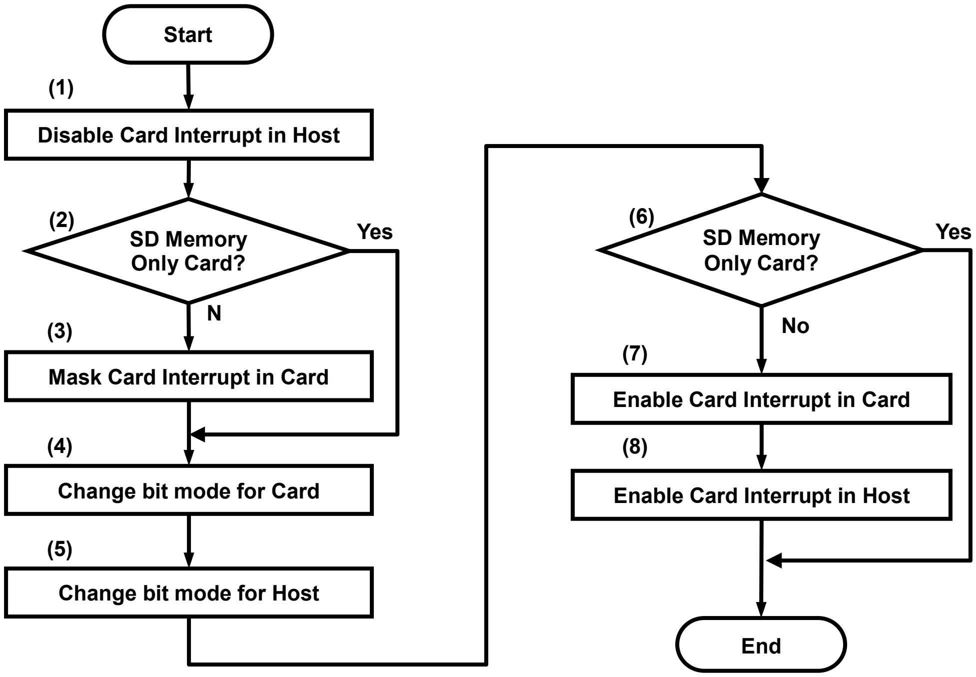


Figure 3-7: Change Bus Width Sequence

The sequence for changing bit mode on SD Bus is shown in Figure 3-7.

(1) Set **Card Interrupt Status Enable** in the *Normal Interrupt Status Enable* register to 0 for masking incorrect interrupts that may occur while changing the bus width.
(2) In case of SD memory only card, go to step (4). In case of other card, go to step (3).
(3) Set "**IENM**" of the CCCR in an SDIO or SD combo card to 0 by CMD52. Please refer to Section 3.7.1 for how to generate CMD52.
(4) Change the bus width mode for an SD card. SD Memory Card bus width is changed by ACMD6 and SDIO card bus width is changed by setting **Bus Width** of *Bus Interface Control* register in CCCR.
(5) In case of changing to 4-bit mode, set **Data Transfer Width** to 1 in the *Host Control 1* register. In another case (1-bit mode), set this bit to 0.
(6) In case of SD memory only card, go to the 'End'. In case of other card, go to step (7).
(7) Set "**IENM**" of the CCCR in an SDIO or SD combo card to 1 by CMD52.
(8) Set **Card Interrupt Status Enable** in the *Normal Interrupt Status Enable* register to 1.

Note that if the card is locked, bus width cannot be changed. Unlock the card is required before changing bus width.

3.5 Timeout Setting on DAT Line

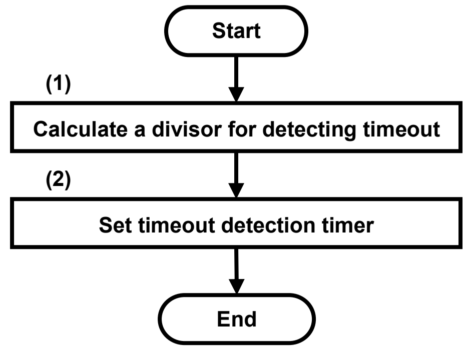


Figure 3-8: Timeout Setting Sequence

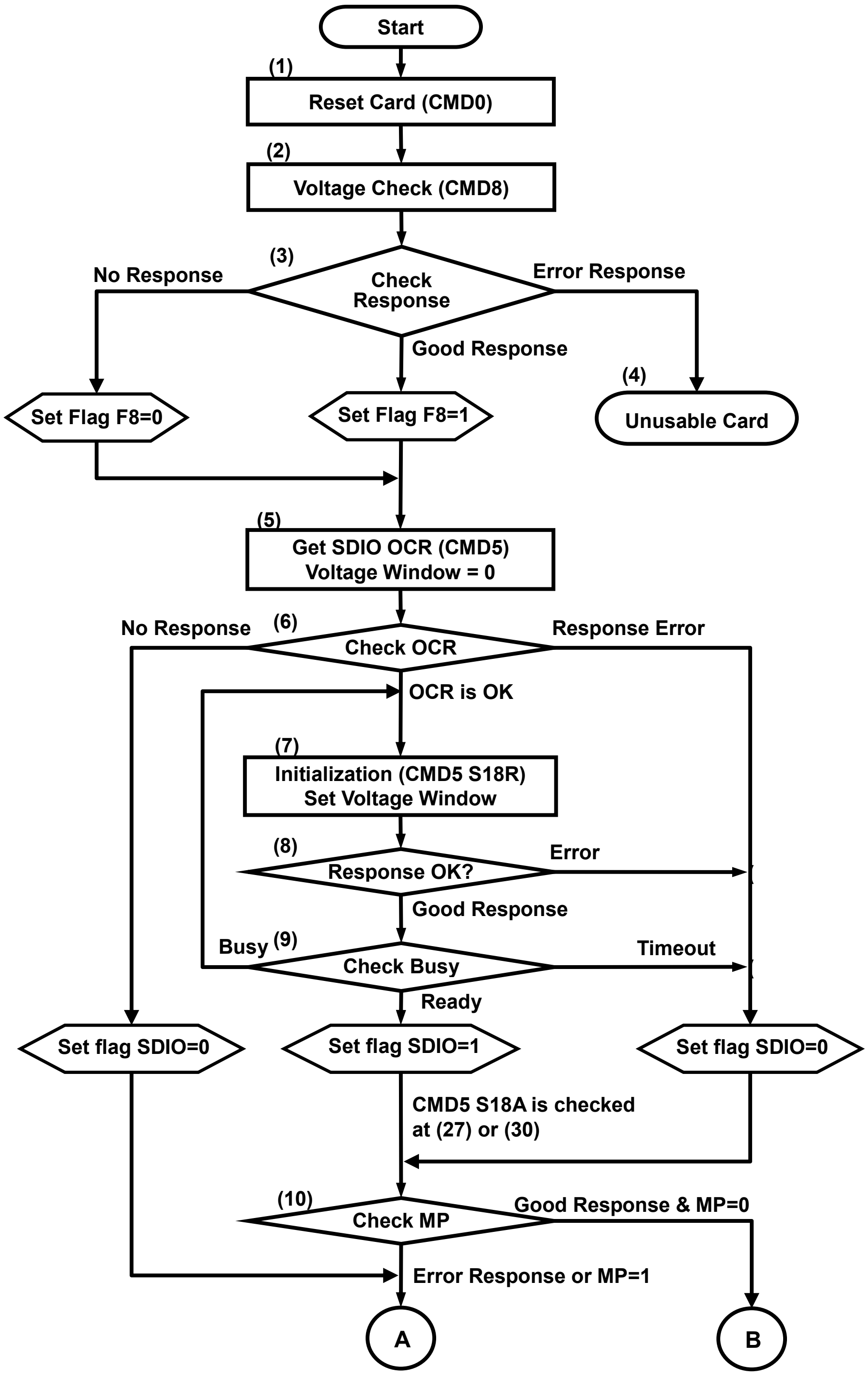
In order to detect timeout errors on DAT line, the Host Driver shall execute the following two steps before any SD transaction. For more information regarding SD transactions, refer to Section 3.7.2

(1) Calculate a divisor to detect timeout errors by reading **Timeout Clock Frequency** and **Timeout Clock Unit** in the *Capabilities* register. If **Timeout Clock Frequency** is 00 0000b, the Host System shall provide this information to the Host Driver by another method.

(2) Set **Data Timeout Counter Value** in the *Timeout Control* register in accordance with the value from step (1) above.

3.6 Card Initialization and Identification (for SD I/F)

Figure 3-9 shows initialization and card identification sequence for the Standard Capacity SD Memory Card (SDSC), the High Capacity SD Memory Card (SDHC) and the Extended Capacity SD Memory Card (SDXC) that was based on the Physical Layer Version 3.01. Refer to the latest sequence; Figure 3-19 to Figure 3-21 and Figure 4-6 in the Physical Layer Version 4.10, and Figure 3-2, Figure 3-10 and Figure 3-11 in the SDIO Version 4.10.



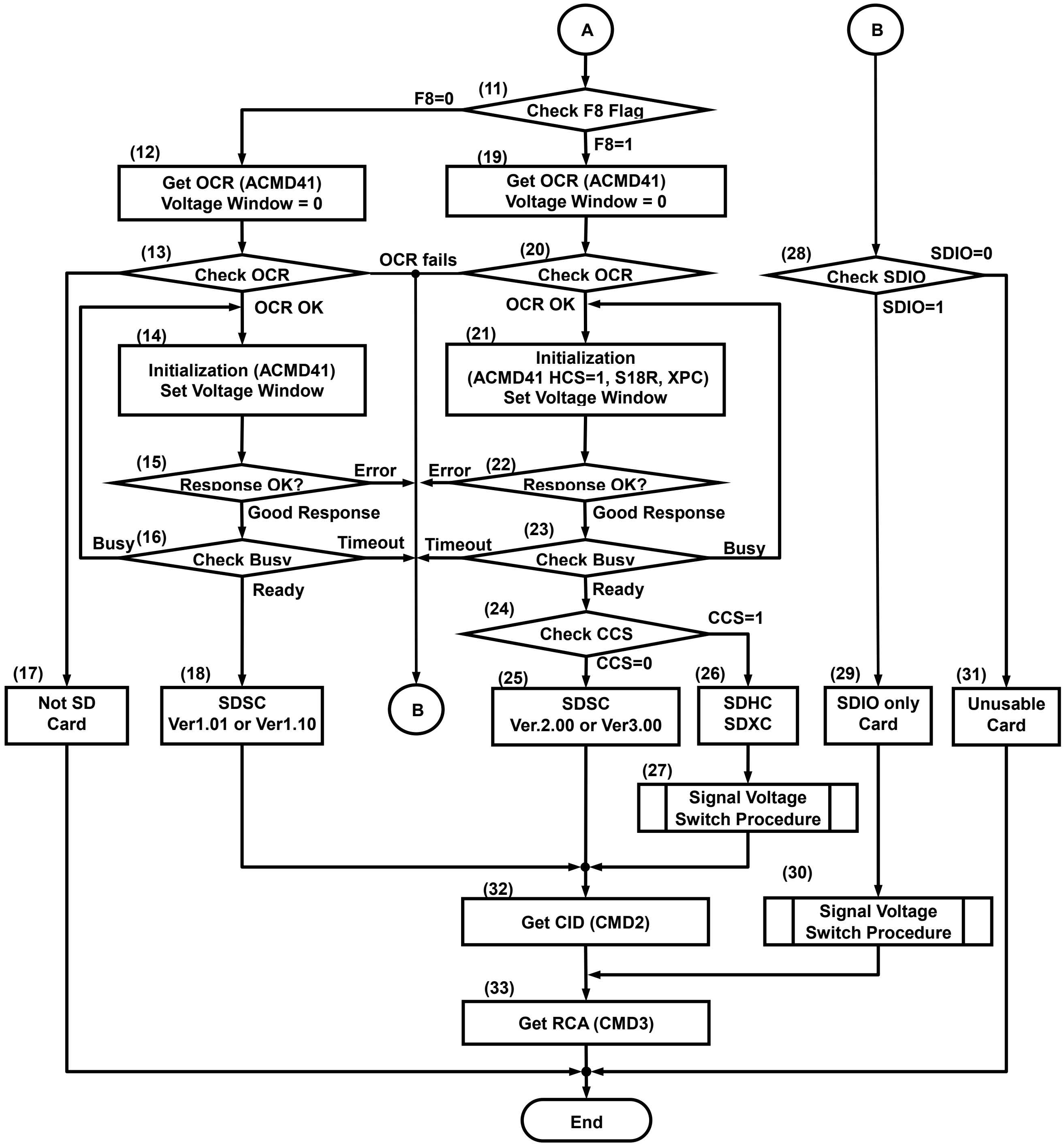


Figure 3-9 : Card Initialization and Identification

(1) SD Bus mode is selected by CMD0 (Keep Pin 1 to high during CMD0 execution).
(2) New CMD8 shall be issued after CMD0 to support High Capacity SD Memory Card.
(3) Voltage check command enables the Hosts to support future low voltage specification. However, at this time, only one voltage is defined. Legacy cards and Not SD cards do not respond to CMD8. In this case, set F8 to 0 (F8 is CMD8 valid flag used in step (11)) and go to Step (5).

Only Version 2.00 or higher cards can respond to CMD8. The host needs to check whether CRC of the response is valid and whether VHS and check pattern in the argument are equal to VCA and check pattern in the response. Passing all these checks results in CMD8 response OK. In this case, set F8 to 1 and go to step (5). If one of the checks is failed, go to step (4).

(4) Initialization is stopped by CMD8 fails. The Host Driver should retry step (1) to (3) one more time. (This is not described in the figure).

(5) SDIO OCR is available by issuing CMD5 with setting voltage window (bit 23 to 0) in the argument to 0. SDIO initialization is not started.

(6) No response means the card does not have SDIO function. Set SDIO flag to 0 and go to step (11). If the card responds to CMD5 and the response is OK, go to step (7). If the response is error, set SDIO flag to 0 and go to step (10). SDIO flag indicates whether SDIO functions are initialized or not.

(7) The SDIO portion starts initialization by CMD5 with setting the supply voltage to the voltage window. UHS-I supported host sets S18R to1. If the supplied voltage is not matched with voltage window of card, the card goes into inactive state and does not return the response.

(8) If no response or error response is receive, set SDIO flag to 0 and go to step (10). If good response is received, go to step (9).

(9) Check busy status in the response. If busy is released, set SDIO flag to 1 and go to step (10). Repeat from step (7) while busy is indicated. Detecting timeout of 1 second exits the loop. In this case, set SDIO flag to 0 and go to step (10).

(10) Good response in this step means that all responses received at (6) and (8) are valid. When response is good, MP (memory present) flag in the response can be checked. If the response valid and MP=0, go to step (28). Otherwise, go to step (11).

(11) Check F8 flag set in step (3). If CMD8 is executed correctly (F8=1), go to step (19). Otherwise, go to step (12).

(12) OCR is available by issuing ACMD41 with the voltage window (bit 23 to 0) in the argument is set to 0. Memory initialization is not started. The response of CMD55 (ACMD41) may indicate illegal command error due to some SD cards do not recognized CMD8. The Host Driver should ignore this error or issue CMD0 before ACMD41 to clear this error status.

(13) If response of CMD55 is not received, the card is not SD cards and goes to (17). If the card responds to CMD55, it may also respond to CMD41. If the responses of ACMD41 are OK, go to Step (14). Otherwise, go to step (28). Locked card can be detected by the card status in the response of CMD55.

(14) The memory portion starts initialization by Issuing ACMD41 with setting the supply voltage to the voltage window. If the supplied voltage is not matched with voltage window of card, the card goes into inactive state and does not return the response.

(15) If no response or error response is received, go to step (28). If good response is received, go to step (16).

(16) Check busy status in the response. If busy is released, go to step (18). Repeat from step (14) while busy is indicated. The interval of ACMD41 shall be less than 50ms. Detecting timeout of 1 second exits the loop and go to step (28).

(17) The host recognizes that the card is not SD memory card and quits SD card initialization.

(18) The host recognizes that the card is Version 1.xx Standard Capacity SD Memory Card. Go to Step (30).

(19) OCR is available by issuing ACMD41 with setting the voltage window (bit 23 to 0) in the argument is set to 0. Memory initialization is not started. Setting of HCS does not affect this operation.

(20) If the card responds to CMD55, it may also respond to CMD41. If the responses of ACMD41 are OK, go to Step (21). Otherwise, go to step (28). Locked card can be detected by the card status in the response of CMD55.

(21) The memory portion starts initialization by Issuing ACMD41 with setting the supply voltage to the voltage window. UHS-I supported host sets S18R to1. If the host can supply more than 150mA, XPC is set to 1. HCS in the argument is set to 1, which indicates supporting High Capacity

Memory Card. If the supplied voltage is not matched with voltage window of card, the card goes into inactive state and does not return the response.

(22) If no response or error response is received, go to step (28). If good response is received, go to step (23).
(23) Check busy status in the response. If busy is released, go to step (24). Repeat from step (21) while busy is indicated. The interval of ACMD41 shall be less than 50ms. Detecting timeout of 1 second exits the loop and go to step (28).
(24) CCS in the response is valid after busy is released. If CCS = 0, it indicates the Standard Capacity SD Memory Card and go to step (25). If CCS = 1, it indicates the High Capacity SD Memory Card or Extended Capacity Memory Card and go to Step (26).
(25) The host recognizes that the card is Ver2.00 or Ver3.00 Standard Capacity SD Memory Card. Optimal functions defined in Version 2.00 or higher are available. Go to Step (32).
(26) The host recognizes that the card is the High Capacity SD Memory Card or Extended Capacity Memory Card.
(27) Perform the signal voltage switch procedure and go to step (32).
(28) Check SDIO flag. If SDIO=1, go to step (28). Otherwise, go to step (31).
(29) The host recognizes that the card is SDIO only card and go to step (30).
(30) Perform the signal voltage switch procedure and go to step (33).
(31) The host recognizes that the card is unusable.
(32) In case of memory card, CMD2 is issued to get CID and Go to Step (31).
(33) CMD3 is issued to get RCA. If the RCA number is 0, the Host should issue CMD3 again.

3.6.1 Signal Voltage Switch Procedure (for UHS-I)

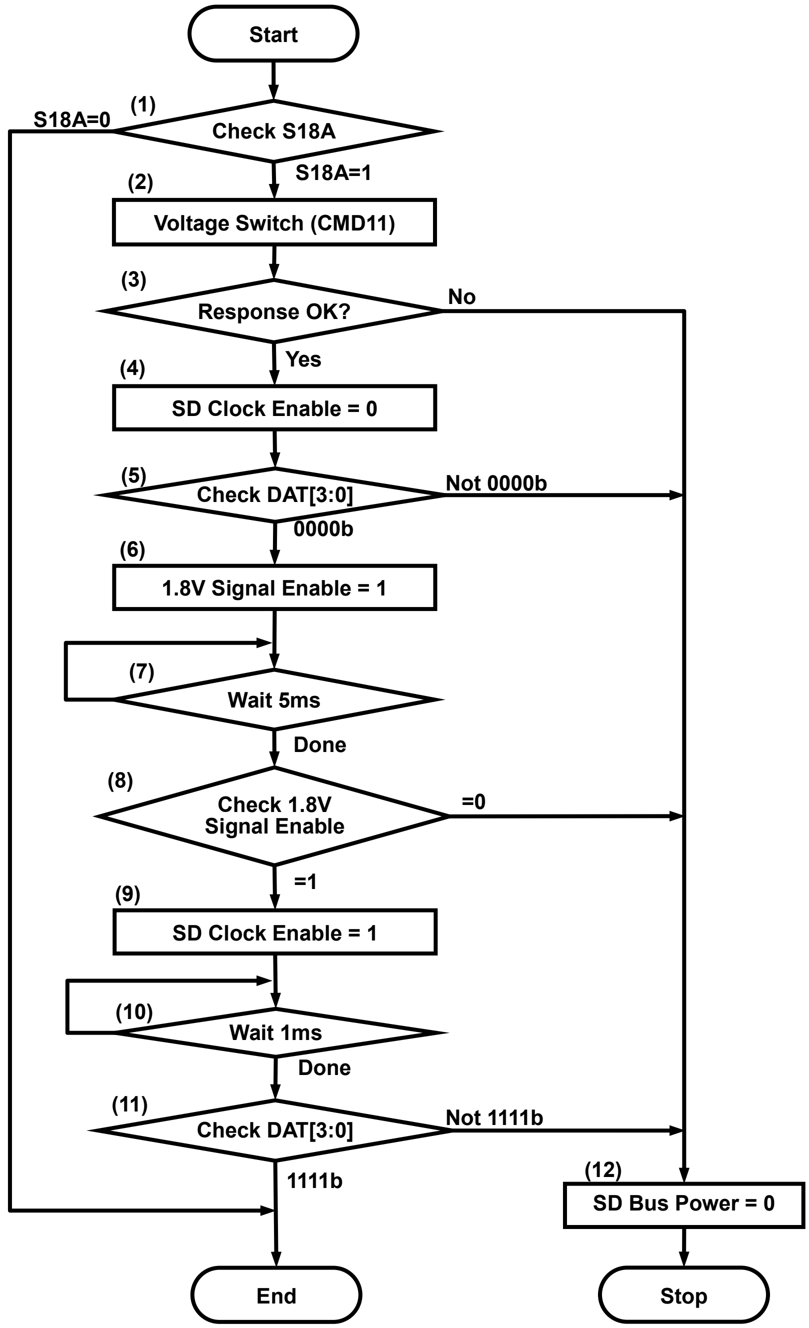


Figure 3-10 : Signal Voltage Switch Procedure

(1) If S18A of CMD5 or S18A of ACMD41 is set to 1, signal voltage switch is performed according to the following steps. Otherwise, exits from this procedure.
(2) Issue CMD11.
(3) Check response and if an error is detected, go to step (12)
(4) Stop providing SD clock to the card.
(5) Check DAT[3:0] level. If the level is 0000b, the card is ready to start voltage switch sequence. Otherwise, go to (12) to quit the sequence.
(6) Set **1.8V Signal Enable** in the *Host Control 2* register.
(7) Wait 5ms. 1.8V voltage regulator shall be stable within this period.
(8) If **1.8V Signal Enable** is cleared by Host Controller, go to step (12).
(9) Provide SD Clock to the card again.
(10) Wait 1ms.
(11) Check DAT[3:0] level. If the level is 1111b, switch to 1.8V signal level is completed successfully. Otherwise, go to (12).
(12) If an error occurs during voltage switch procedure, stop providing the power to the card. In this case, Host Driver should retry initialization procedure by setting S18R to 0 at step (7) and (21) in Figure 3-9.

3.7 SD Transaction Generation

This section describes the sequences how to generate and control various kinds of SD transactions. SD transactions are classified into three cases:

(1) Transactions that do not use the DAT line
(2) Transactions that use the DAT line only for the busy signal
(3) Transactions that use the DAT line for transferring data

In this specification, the first and the second case's transactions are classified as "Transaction Control without Data Transfer using DAT Line" and the third case's transaction is classified as "Transaction Control with Data Transfer using DAT Line".
Refer to the latest SD Physical Layer Specification and SDIO Specification for more detail about SD commands specification.

3.7.1 Transaction Control without Data Transfer Using DAT Line

In this section, the sequence for how to issue SD Command and how to complete SD Command is explained. Figure 3-11 shows the sequence to issue an SD Command and Figure 3-12 shows the sequence to finalize an SD Command.

3.7.1.1 The Sequence to Issue an SD Command

The sequence to issue the SD Command is detailed below.

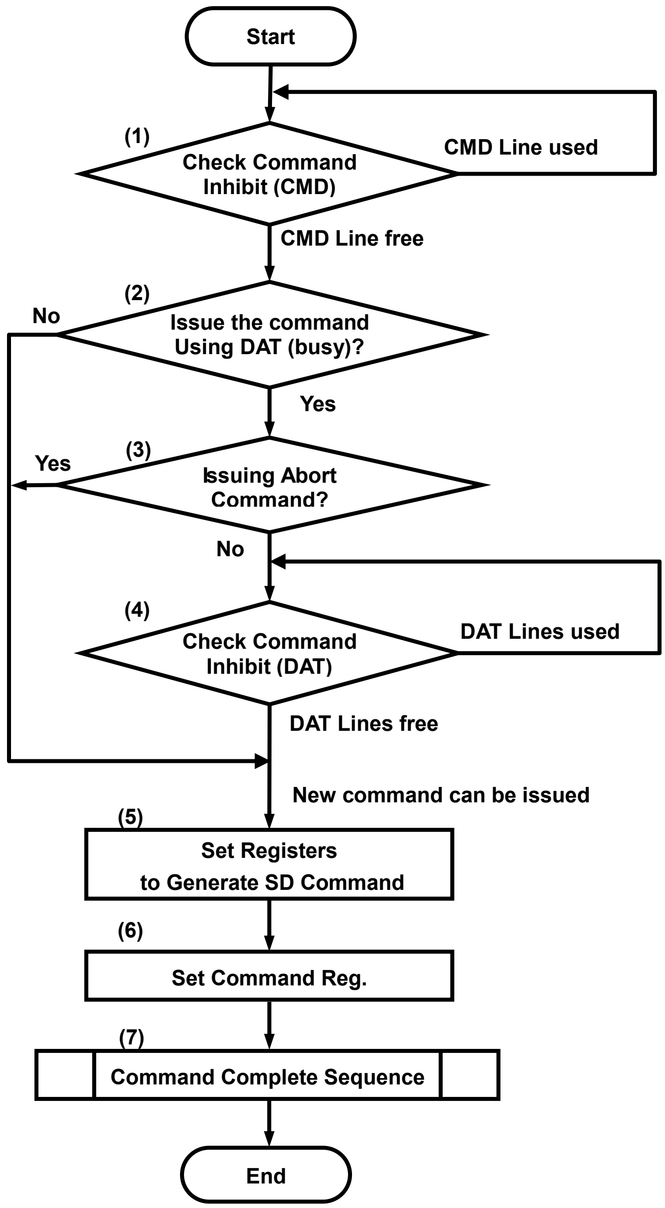


Figure 3-11: SD Command Issue Sequence

(1) Check **Command Inhibit (CMD)** in the *Present State* register. Repeat this step until **Command Inhibit (CMD)** is 0. That is, when **Command Inhibit (CMD)** is 1, the Host Driver shall not issue an SD Command.
(2) If the Host Driver issues an SD Command using DAT lines including busy signal, go to step (3). If without using DAT lines including busy signal, go to step (5).
(3) If the Host Driver is issuing an abort command, go to step (5). In the case of non-abort command, go to step (4).
(4) Check **Command Inhibit (DAT)** in the *Present State* register. Repeat this step until **Command Inhibit (DAT)** is set to 0.
(5) Set registers as described in Table 1-2 except Command register.
(6) Set the *Command* register.
    Note: Writing the upper byte [3] in the *Command* register causes the Host Controller to issue an SD command to the SD card.
(7) Perform Command Completion Sequence in accordance with 3.7.1.2.

3.7.1.2 The Sequence to Finalize a Command

Figure 3-12 shows the sequence to finalize an SD Command when response check is disabled. There is a possibility that some errors (Command Index/End bit/CRC/Timeout Error) occur during this sequence. If response check is enabled, error is indicated by Response Error Interrupt

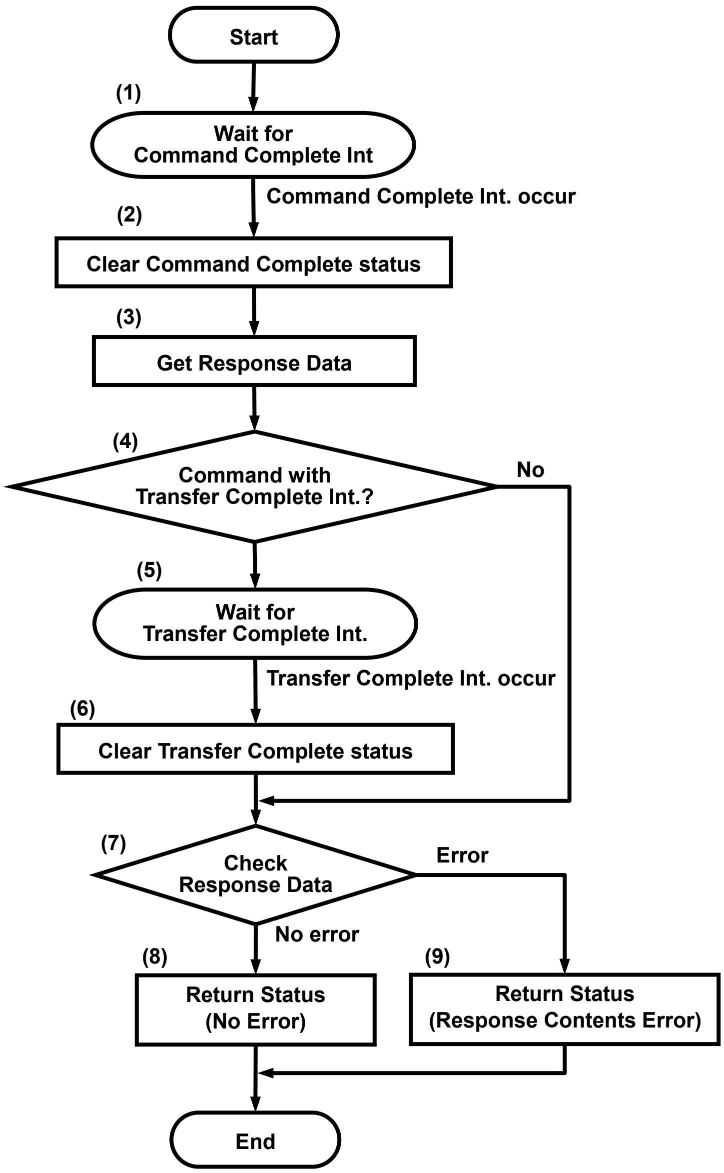


Figure 3-12: Command Complete Sequence

(1) If **Response Interrupt Disable** in the *Transfer Mode* register is set to 1 (response check is enabled), go to stop (4) else wait for the **Command Complete** Interrupt. If the **Command Complete** Interrupt has occurred, go to step (2).
(2) Write 1 to **Command Complete** in the *Normal Interrupt Status* register to clear this bit.
(3) Read the *Response* register and get necessary information of the issued command.
(4) Judge whether the command uses the **Transfer Complete** Interrupt or not. If it uses **Transfer Complete**, go to step (5). If not, go to step (7).
(5) Wait for the **Transfer Complete** Interrupt. If the **Transfer Complete** Interrupt has occurred, go to step (6).
(6) Write 1 to **Transfer Complete** in the *Normal Interrupt Status* register to clear this bit.
(7) Check for errors in Response Data. If there is no error, go to step (8). If there is an error, go to step (9).
(8) Return Status of "No Error".
(9) Return Status of "Response Contents Error".

Note1: While waiting for the **Transfer Complete** interrupt, the Host Driver shall only issue commands that do not use the busy signal.
Note2: The Host Driver shall judge the Auto CMD12 complete by monitoring Transfer Complete.
Note3: When the last block of un-protected area is read using memory multiple block read command (CMD18), OUT_OF_RANGE error may occur even if the sequence is correct. The Host Driver should ignore it. This error will appear in the response of Auto CMD12 or in the response of the next memory command.

3.7.2 Transaction Control with Data Transfer Using DAT Line

Depending on whether DMA (optional) is used or not, there are two execution methods. The sequence not using DMA is shown in Figure 3-13 and the sequence using DMA is shown in Figure 3-14.

In addition, the sequences for SD transfers are classified into following three kinds according to how the number of blocks is specified:

(1) Single Block Transfer:
The number of blocks is specified to the Host Controller before the transfer. The number of blocks specified is always one.
(2) Multiple Block Transfer:
The number of blocks is specified to the Host Controller before the transfer. The number of blocks specified shall be one or more.
(3) Infinite Block Transfer:
The number of blocks is not specified to the Host Controller before the transfer. This transfer is continued until an abort transaction is executed. This abort transaction is performed by CMD12 in the case of an SD memory card and by CMD52 in the case of an SDIO card.

3.7.2.1 Not using DMA

The sequence for not using DMA is shown below.

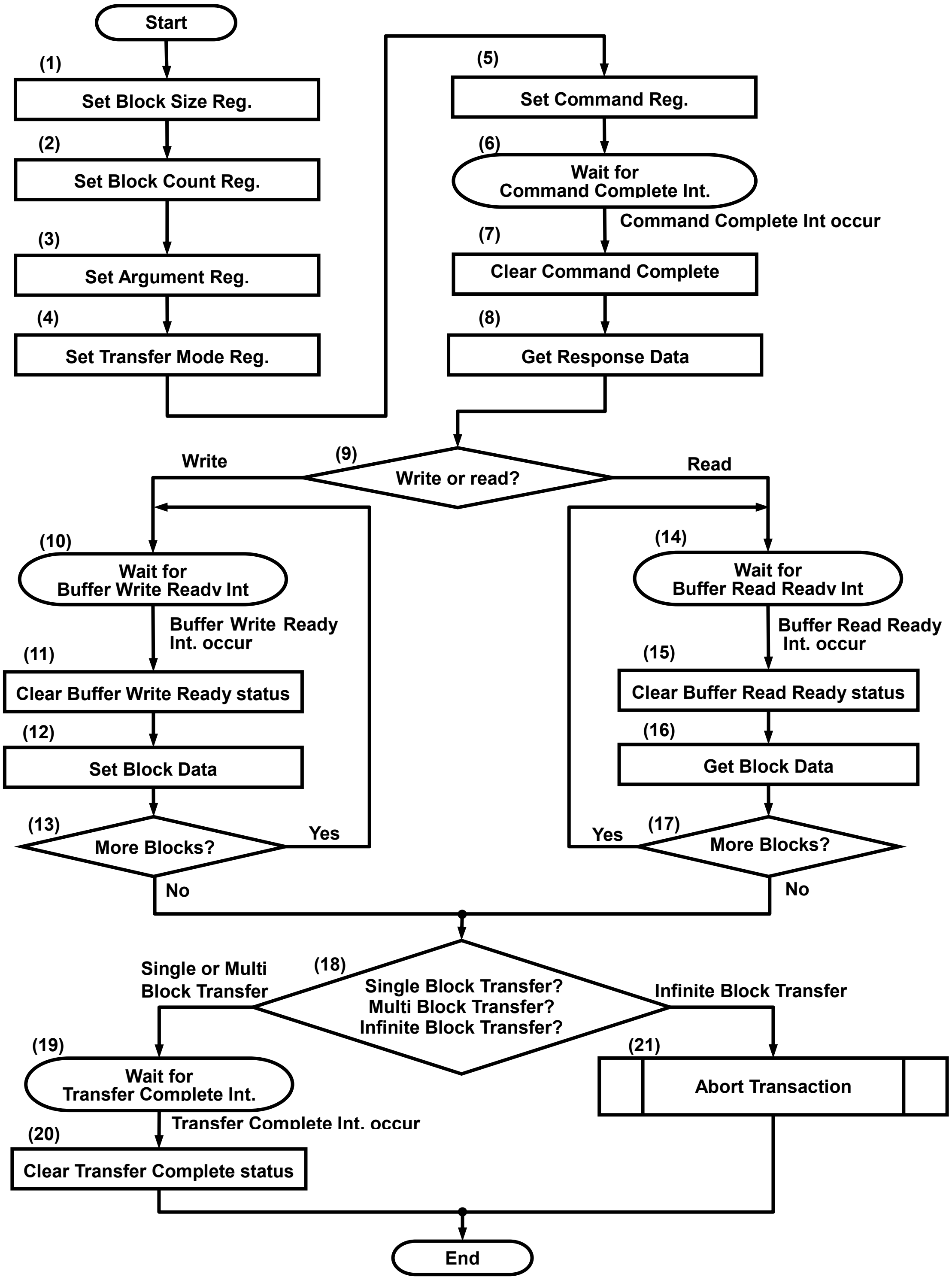


Figure 3-13: Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)

(1) Set the value corresponding to the executed data byte length of one block to *Block Size* register.
(2) Set the value corresponding to the executed data block count to *Block Count* register in accordance with Table 2-9. Refer to Section 1.15 for more details.
(3) Set the argument value to *Argument* register.
(4) Set the value to the *Transfer Mode* register. The Host Driver determines **Multi / Single Block Select, Block Count Enable**, **Data Transfer Direction**, **Auto CMD12 Enable** and **DMA Enable**. **Multi / Single Block Select** and **Block Count Enable** are determined according to Table 2-9.
If response check is enabled (**Response Error Check Enable** =1), set **Response Interrupt Disable** to 1 and select **Response Type R1 / R5**.
(5) Set the value to *Command* register.
Note: When writing the upper byte [3] of *Command* register, SD command is issued.
(6) If response check is enabled, go to stop (9) else wait for the **Command Complete** Interrupt.
(7) Write 1 to the **Command Complete** in the *Normal Interrupt Status* register for clearing this bit.
(8) Read *Response* register and get necessary information of the issued command.
(9) In the case where this sequence is for write to a card, go to step (10). In case of read from a card, go to step (14).
(10) Then wait for **Buffer Write Ready** Interrupt.
(11) Write 1 to the **Buffer Write Ready** in the *Normal Interrupt Status* register for clearing this bit.
(12) Write block data (in according to the number of bytes specified at the step (1)) to *Buffer Data Port* register.
(13) Repeat until all blocks are sent and then go to step (18).
(14) Then wait for the **Buffer Read Ready** Interrupt.
(15) Write 1 to the **Buffer Read Ready** in the *Normal Interrupt Status* register for clearing this bit.
(16) Read block data (in according to the number of bytes specified at the step (1)) from the *Buffer Data Port* register.
(17) Repeat until all blocks are received and then go to step (18).
(18) If this sequence is for Single or Multiple Block Transfer, go to step (19). In case of Infinite Block Transfer, go to step (21).
(19) Wait for **Transfer Complete** Interrupt.
(20) Write 1 to the **Transfer Complete** in the *Normal Interrupt Status* register for clearing this bit.
(21) Perform the sequence for Abort Transaction in accordance with Section 3.8.
Note: Step (1) and Step (2) can be executed at same time. Step (4) and Step (5) can be executed at same time.

3.7.2.2 Using SDMA

The sequence for using SDMA is shown below.

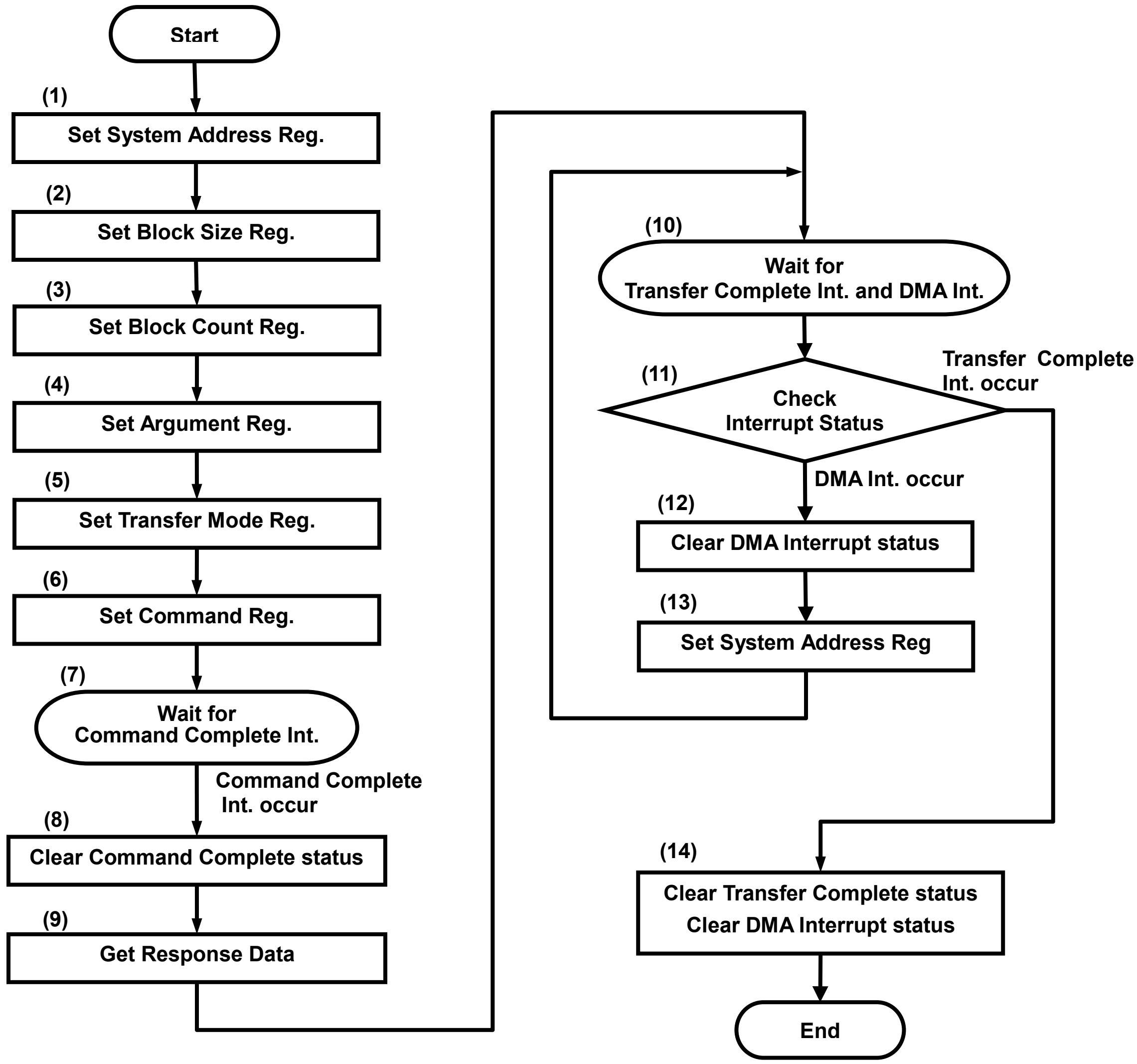


Figure 3-14: Transaction Control with Data Transfer Using DAT Line Sequence (Using SDMA)

(1) Data location of system memory is set to the *SDMA System Address* register if **Host Version 4 Enable** = 0 or set to *ADMA System Address* register if **Host Version 4 Enable** = 1.
(2) Set the value corresponding to the executed data byte length of one block in the *Block Size* register.
(3) Set the value corresponding to the executed data block count in the *Block Count* register in accordance with Table 2-9. Refer to Section 1.15 for more details.
(4) Set the argument value to the *Argument* register.
(5) Set the value to the *Transfer Mode* register. The Host Driver determines **Multi / Single Block Select, Block Count Enable**, **Data Transfer Direction**, **Auto CMD12 Enable** and **DMA Enable**. **Multi / Single Block Select** and **Block Count Enable** are determined according to Table 2-9.
If response check is enabled (Response Error Check Enable =1), set Response Interrupt Disable to 1 and select Response Type R1 / R5.
(6) Set the value to the *Command* register.
Note: When writing to the upper byte [3] of the *Command* register, the SD command is issued and SDMA is started.
(7) If response check is enabled, go to stop (10) else wait for the **Command Complete** Interrupt.
(8) Write 1 to the **Command Complete** in the *Normal Interrupt Status* register to clear this bit.
(9) Read *Response* register and get necessary information of the issued command.
(10) Wait for the **Transfer Complete** Interrupt and **DMA Interrupt**.
(11) If **Transfer Complete** is set to 1, go to Step (14) else if **DMA Interrupt** is set to 1, go to Step (12).**Transfer Complete** is higher priority than **DMA Interrupt**.
(12) Write 1 to the **DMA Interrupt** in the *Normal Interrupt Status* register to clear this bit.
(13) Set the next system address of the next data position to the *System Address* register and go to Step (10).
(14) Write 1 to the **Transfer Complete** and **DMA Interrupt** in the *Normal Interrupt Status* register to clear this bit.

Note: Step (2) and Step (3) can be executed simultaneously. Step (5) and Step (6) can also be executed simultaneously.

3.7.2.3 Using ADMA

The sequence for using ADMA is shown below.

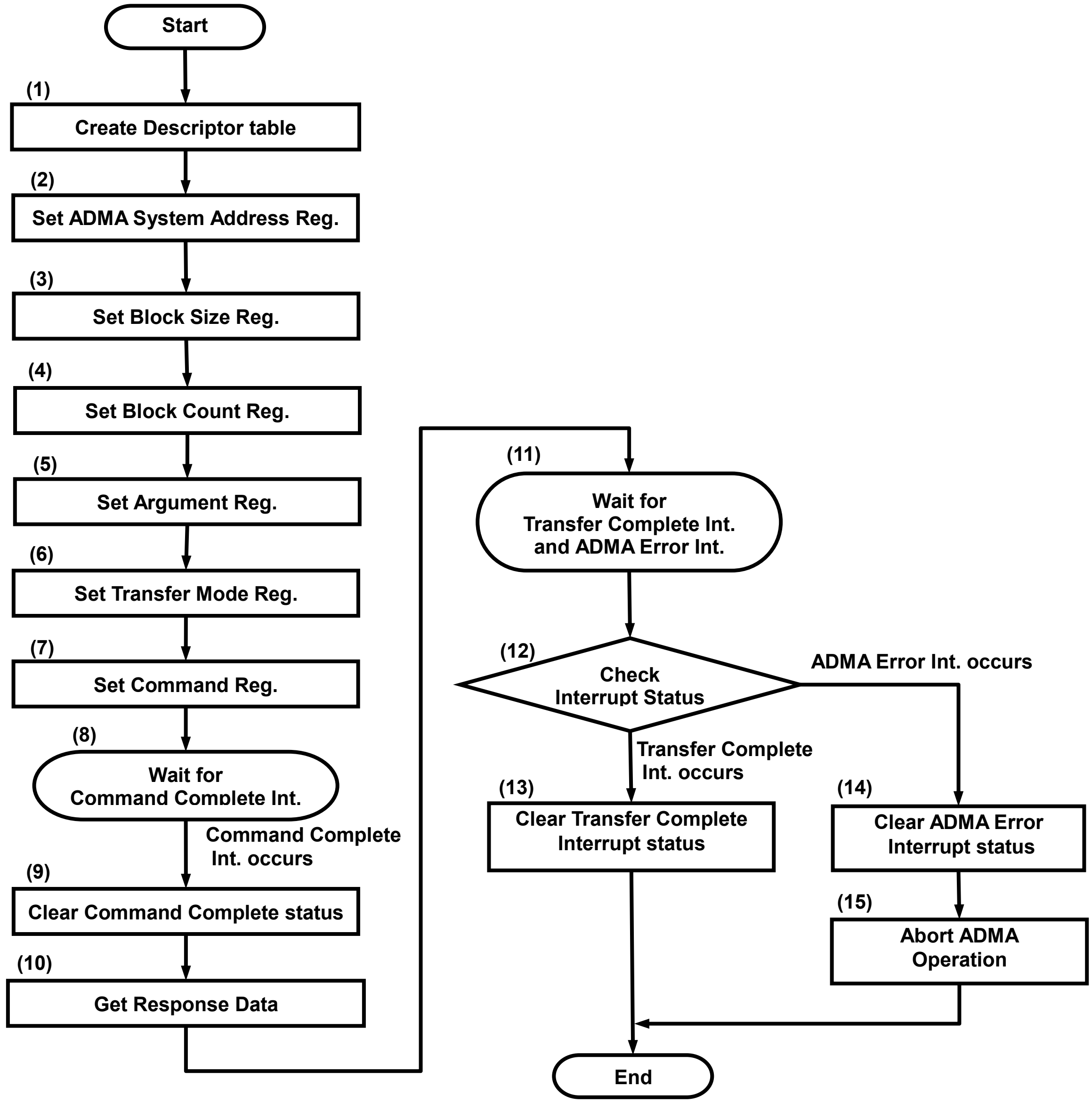


Figure 3-15: Transaction Control with Data Transfer Using DAT Line Sequence (Using ADMA)

(1) Create Descriptor table for ADMA in the *system memory*
(2) Set the Descriptor address for ADMA in the ADMA *System Address* register.
(3) Set the value corresponding to the executed data byte length of one block in the *Block Size* register.
(4) Set the value corresponding to the executed data block count in the *Block Count* register in accordance with Table 2-9. Refer to Section 1.15 for more details.
(5) Set the argument value to the *Argument* register.
(6) Set the value to the *Transfer Mode* register. The Host Driver determines **Multi / Single Block Select, Block Count Enable**, **Data Transfer Direction**, **Auto CMD12 Enable** and **DMA Enable**. **Multi / Single Block Select** and **Block Count Enable** are determined according to Table 2-9.
If response check is enabled (Response Error Check Enable =1), set Response Interrupt Disable to 1 and select Response Type R1 / R5.
(7) Set the value to the *Command* register.
Note: When writing to the upper byte [3] of the *Command* register, the SD command is issued and DMA is started.
(8) If response check is enabled, go to stop (11) else wait for the **Command Complete** Interrupt.
(9) Write 1 to the **Command Complete** in the *Normal Interrupt Status* register to clear this bit.
(10) Read *Response* register and get necessary information of the issued command.
(11) Wait for the **Transfer Complete** Interrupt and **ADMA Error Interrupt**.
(12) If **Transfer Complete** is set to 1, go to Step (13) else if **ADMA Error Interrupt** is set to 1, go to Step (14).
(13) Write 1 to the **Transfer Complete Status** in the *Normal Interrupt Status* register to clear this bit.
(14) Write 1 to the **ADMA Error Interrupt Status** in the *Error Interrupt Status* register to clear this bit.
(15) Abort ADMA operation. SD card operation should be stopped by issuing abort command. If necessary, the Host Driver checks *ADMA Error Status* register to detect why **ADMA error** is generated.

Note: Step (3) and Step (4) can be executed simultaneously. Step (6) and Step (7) can also be executed simultaneously.

3.8 Abort Transaction

An abort transaction is performed by issuing CMD12 for an SD memory card and by issuing CMD52 for an SDIO card. There are two cases where the Host Driver needs to do an Abort Transaction. The first case is when the Host Driver stops Infinite Block Transfers. The second case is when the Host Driver stops transfers while a Multiple Block Transfer is executing.

There are two ways to issue an Abort Command. The first is an asynchronous abort. The second is a synchronous abort. In an asynchronous abort sequence, the Host Driver can issue an Abort Command at any time unless **Command Inhibit (CMD)** in the *Present State* register is set to 1. In a synchronous abort, the Host Driver shall issue an Abort Command after the data transfer stopped by using **Stop At Block Gap Request** in the *Block Gap Control* register.

3.8.1 Abort Command Sequence

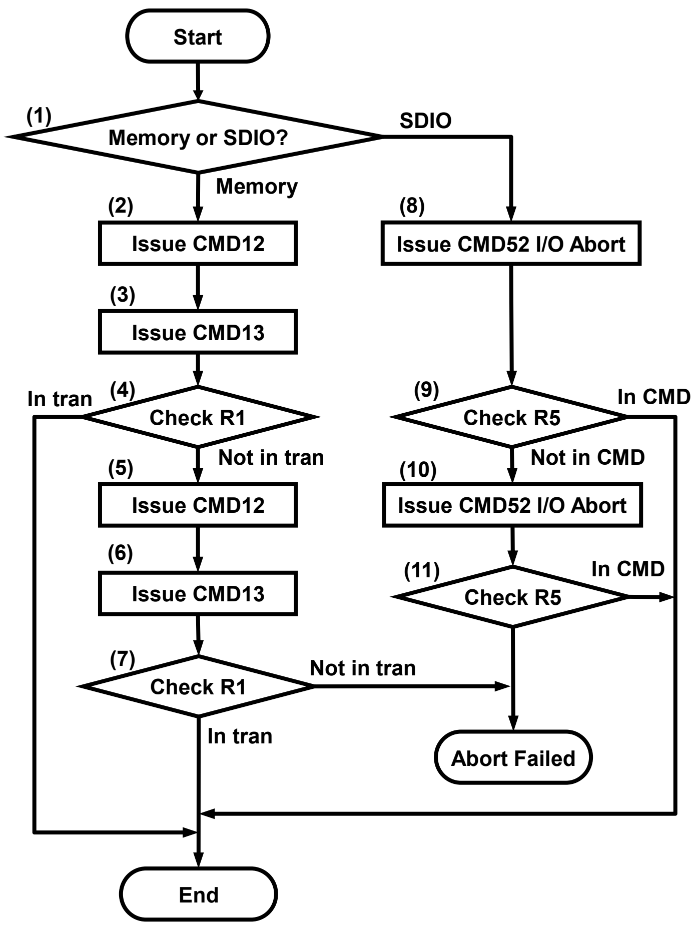


Figure 3-16 : Abort Command Sequence

(1) Check whether device is memory or SDIO. Steps (2) to (7) are for memory abort and Steps (8) to (11) are for I/O abort.
(2) Issue CMD12 to abort memory card. If card is already in tran state, CMD12 is not accepted and Host Driver needs to ignore illegal command error in next R1.
(3) Issue CMD13 to check card state after completion of CMD12.

(4) Check R1 and if card is in tran state, abort succeeds.
(5) If card is not in tran state, retries to issue CMD12 one more time.
(6) Issue CMD13 to check card state after completion of CMD12.
(7) Check R1 and if card is in tran state, retry abort succeeds. Otherwise, abort fails.
(8) Issue CMD52 I/O Abort with RAW (read after write) mode to write CCCR 06h.
(9) Check R5 and if SDIO card is in CMD state, abort succeeds.
(10) If SDIO card is not in CMD state, retries to issue CMD52 I/O Abort with RAW one more time.
(11) Check R5 and if SDIO card is in CMD state, retry abort succeeds. Otherwise, abort fails.

3.8.2 Asynchronous Abort

The sequence for Asynchronous Abort is shown in Figure 3-17.

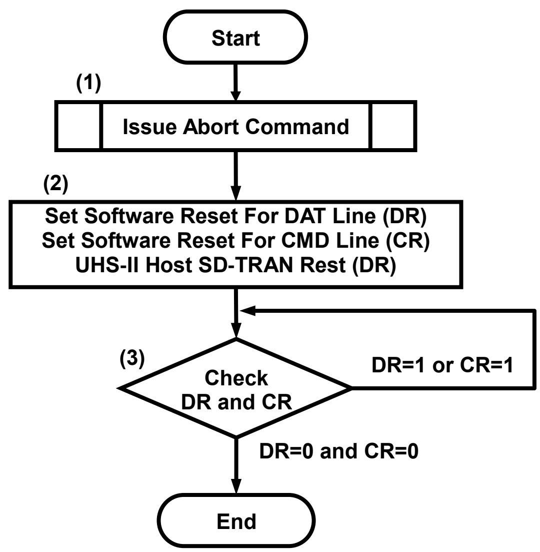


Figure 3-17: Asynchronous Abort Sequence

(1) Issue an Abort Command in accordance with Section 3.8.1.
(2) To discard data in the host controller buffer, set **Software Reset For DAT Line** to 1 in the *Software Reset* register in SD mode or set **Host SD-TRAN Reset** to 1 in the *UHS-II Software Reset* register in UHS-II mode.
In both modes, if the abort command of step (1) is completed successfully, the command circuit reset by **Software Reset For CMD Line** in the *Software Reset* register may not be needed.
(3) Wait completion of all software resets executed in step (2) until the bits which have been set to 1 in step (2) are cleared to 0.

3.8.3 Synchronous Abort

The sequence for Synchronous Abort is shown in Figure 3-18

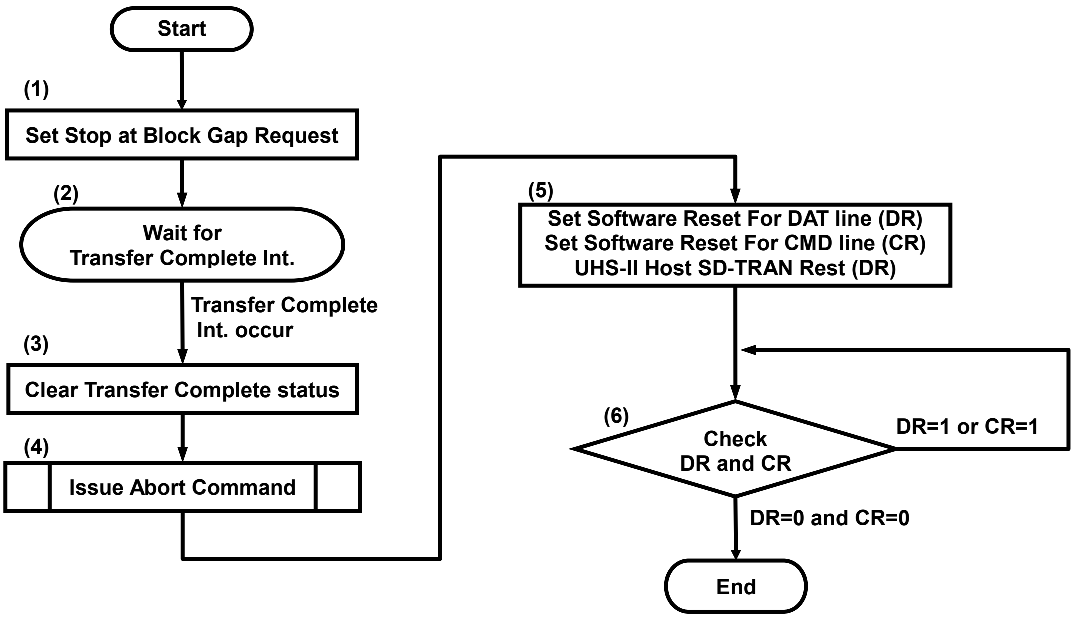


Figure 3-18: Synchronous Abort Sequence

(1) Set the **Stop At Block Gap Request** in the *Block Gap Control* register to 1 to stop SD transactions.
(2) Wait for the **Transfer Complete** Interrupt.
(3) Set the **Transfer Complete** to 1 in the *Normal Interrupt Status* register to clear this bit.
(4) Issue the Abort Command in accordance with Section 3.8.1.
If SD Clock has been stopped to halt read operation, Host Controller provides SD Clock to be able to issue abort command but data circuits including DMA are still stopped.
(5) To discard data in the host controller buffer, set **Software Reset For DAT Line** to 1 in the *Software Reset* register in SD mode or set **Host SD-TRAN Reset** to 1 in the *UHS-II Software Reset* register in UHS-II mode.
In both modes, if the abort command of step (4) is completed successfully, the command circuit reset by **Software Reset For CMD Line** in the *Software Reset* register may not be needed.
(6) Wait completion of all software resets executed in step (5) until the bits which have been set to 1 in step (5) are cleared to 0.

3.8.4 Reset Command

Host Driver should use a reset command when a communication between host and card is not recovered through the abort transaction. Before issuing a reset command, execute **Software Reset For DAT Line** (in SD mode), **Host SD-TRAN Reset** (UHS-II mode) and **Software Reset For CMD Line** (both modes) to reset the command and data circuits of the Host Controller.

3.9 Changing Bus Speed Mode

This section describes the sequence for switching the bus speed mode: Default Speed, High Speed mode and UHS-I mode. The switch command (CMD6) is used to change one of bus speed modes of the SD Memory Card. The **EHS** bit (SDIO Version 2.00) or **BSS[2:0]** bits (SDIO Version 3.00) in the CCCR register is used to change the bus speed mode for SDIO card. In case of Combo card, either of the switch method changes both memory and IO bus speed mode. This means the first switch is effective. Refer to the Physical Layer Specification Version 3.0x and the SDIO Specification Version 3.00 for more information about switching bus speed. Figure 3-19 shows the sequence for switching bus speed mode for Combo Card. Note that if the card is locked, bus width cannot be changed. Unlock the card is required before changing bus width.

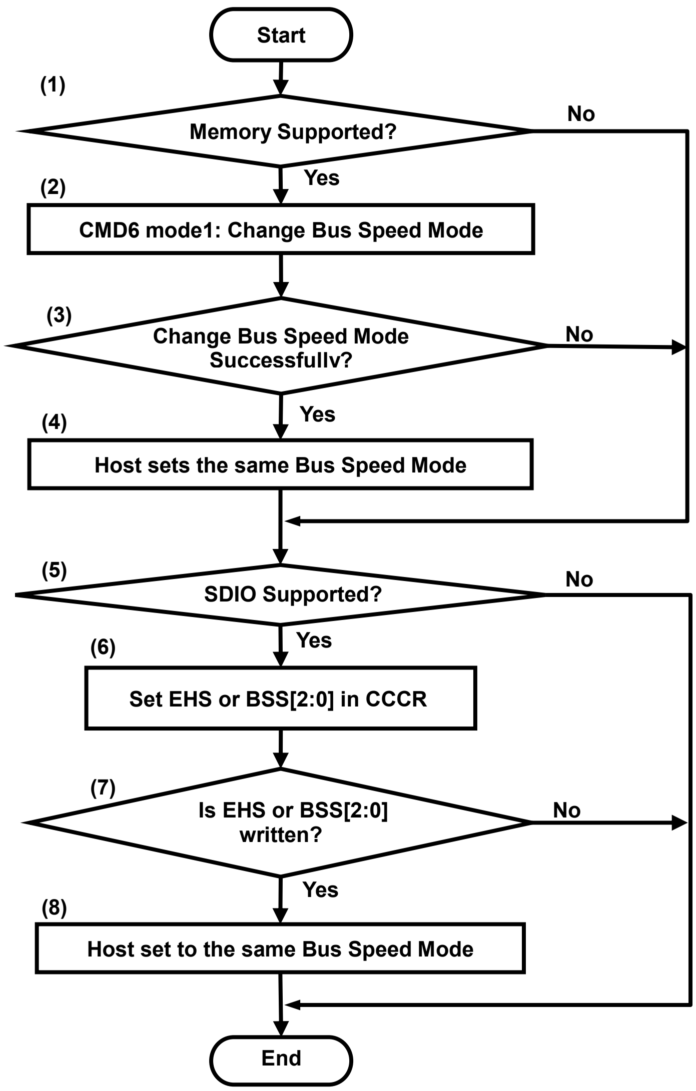


Figure 3-19 : Bus Speed Mode Setting for Combo Card

(1) The Host Driver checks if the card supports memory. If not supported, go to (5).
(2) Issue CMD6 with mode 1 to change one of bus speed modes (e.g., Default Speed mode, High Speed mode or UHS-I mode).
(3) Check the response of CMD6. If the card does not supports CMD6 (no response) or bus speed is not changed successfully, go to step (5). In this case, the card is in Default Speed mode.
(4) The Host Driver changes the Host Controller bus speed mode to the same mode.
(5) The Host Driver checks if the card supports SDIO. If not supported, go to the end.
(6) Issue CMD52 to write **EHS** bit or **BSS**[2:0] bits in CCCR to change bus speed mode. (The same bus speed mode of (2) shall be set.)
(7) If **EHS** or **BSS**[2:0] are not changed successfully, go to the end.
(8) The Host Driver changes the Host Controller bus speed to the same mode. In case of Combo card, bus speed is already changed at step (4) and this step does not affect changing bus speed.

3.10 Error Recovery

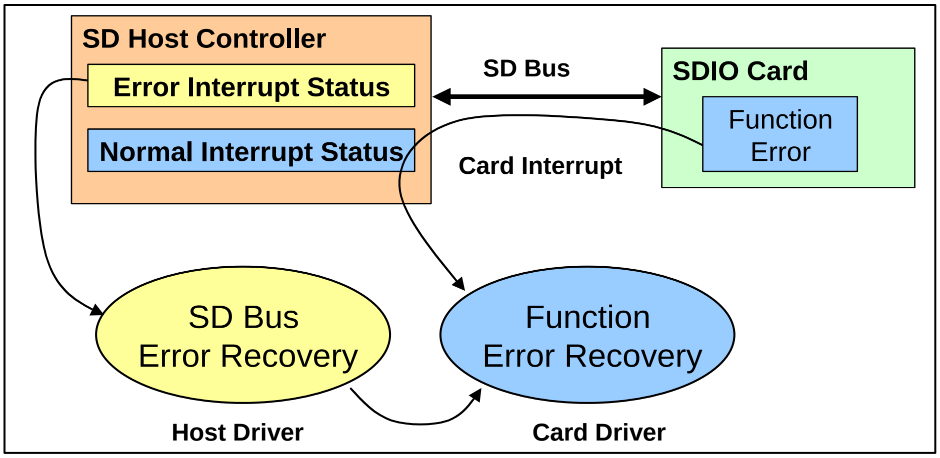


Figure 3-20 : Error Report and Recovery

Figure 3-20 shows concept of error report and its recovery. The Host Controller has two interrupt status registers. If an error occurs in the SD Bus transaction, one of the bits is set in the *Error Interrupt Status* register. If the function errors occur in the SDIO card, the card interrupt informs these function errors and the **Card interrupt** is set in the *Normal Interrupt Status* register. (The **Card Interrupt** is used to inform not only error statuses but also normal information. For example, to inform function ready.) The Card Driver shall do function error recovery because the Host Driver does not know how to control the function. In the case that function error occurs due to SD Bus error, SD Bus error recovery is required before function error recovery. Abort command is used to recover SD Bus, and then the Host Driver should save error statuses related to SD Bus errors before issuing abort command and transfer these statuses to the Card Driver. These statuses may be used to recover function error. Following explanations are related to SD Bus error recovery. This specification does not specify the function error recovery.

When an error occurs during data transfer in 2L-HD UHS-II mode, there will be the case that Host Controller cannot drive D0 lane in input mode due to DIR LSS for retrieving lane direction is not detected. In this case, Host Driver cannot issue abort command for recovery. Then if DIR LSS is not detected, Host Controller sets **Timeout For Deadlock** in the *UHS-II Error Interrupt Status* register. Host Driver should execute power cycle if **Timeout For Deadlock** is detected to recover from this error. Furthermore, if this type of error is detected several times in 2L-HD, Host Driver should use FD mode rather than using 2L-HD.

> Implementation Note:
> If the Card Driver cannot recover the function errors, the Host Driver should try following methods.
> (1) Using **IOEx** for SDIO card
> **IOEx** may be used as the reset per function basis. Sequence is as follows:
> Clear **IOEx**=0 and wait until **IORx**=0 and then set **IOEx**=1 again. SDIO may be recovered when IORx=1.
> (2) Using reset command for memory and SDIO card
> Re-initialization sequence is required.
> (3) Off and on power supply for the SD Bus
> The card may be recovered by the power on reset. Re-initialization sequence is required.

The two cases where the Host Driver needs the "Error Recovery" sequence are classified as follows:

(1) Error Interrupt Recovery:
If error interrupt is indicated by the *Error Interrupt Status* register, the Host Driver shall apply this sequence.

(2) Auto CMD12 Error Recovery:
If there are errors in Auto CMD12, the Host Driver shall apply this sequence. In terms of Return Status, Auto CMD12 Error Recovery is classified into four cases. It is shown in Figure 3-21. If error occurs during memory write transfer, strongly recommend using ACMD22 and then in the following recovery sequence, retry to send remaining blocks not written.

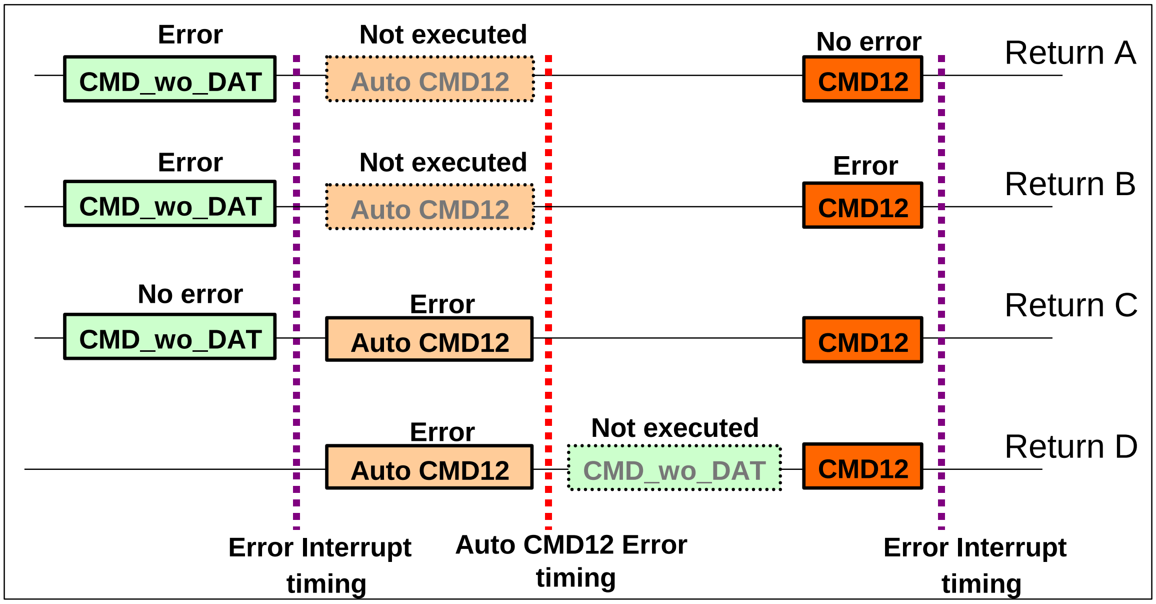


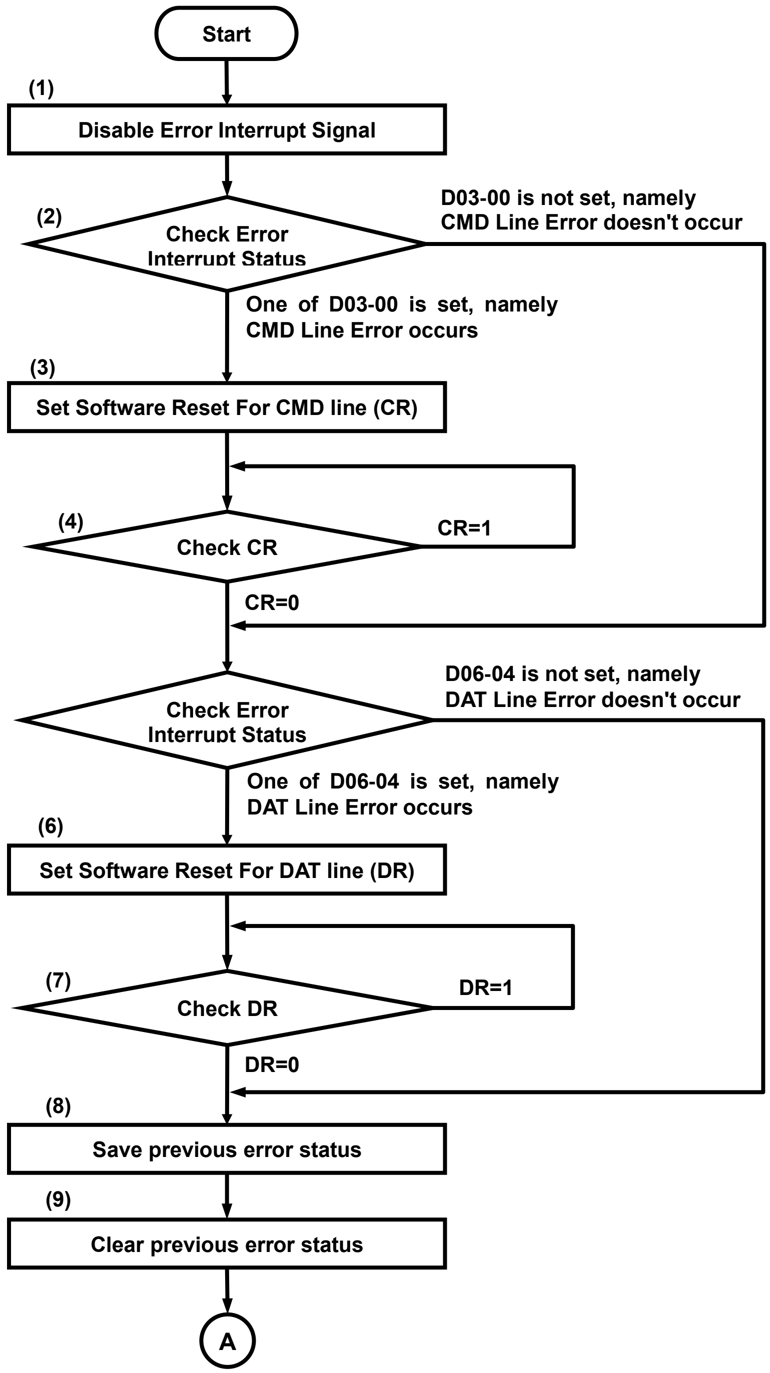
Figure 3-21: Return Status of Auto CMD12 Error Recovery

Implementation Note: Abort command is used to recover from SD Bus error. SDIO transaction abort using CMD52 returns response but in the case of memory transaction abort using CMD12, response returns depending on the memory card state. If no response returns after issue CMD12, the Host Driver should check card state using CMD13. If the state is "tran" in the CURRENT_STATE, consider CMD12 is successful.

Implementation Note: The following sequence is one possible error recovery flow. There may be another methods, sometimes using interrupts or polling. It can be possible to use another flows, based on Host System requirements. In these error recovery sequences, return statuses for the next sequence. When the Host Controller cannot issue the next command due to SD Bus error, the error recovery sequences return "Non-recoverable" status. In this case, the Host System may cut off power to the SD Bus and then power on SD Bus and initialize both the Host Controller and the SD card again.

3.10.1 Error Interrupt Recovery

The sequence for Error Interrupt Recovery is shown in Figure 3-22.



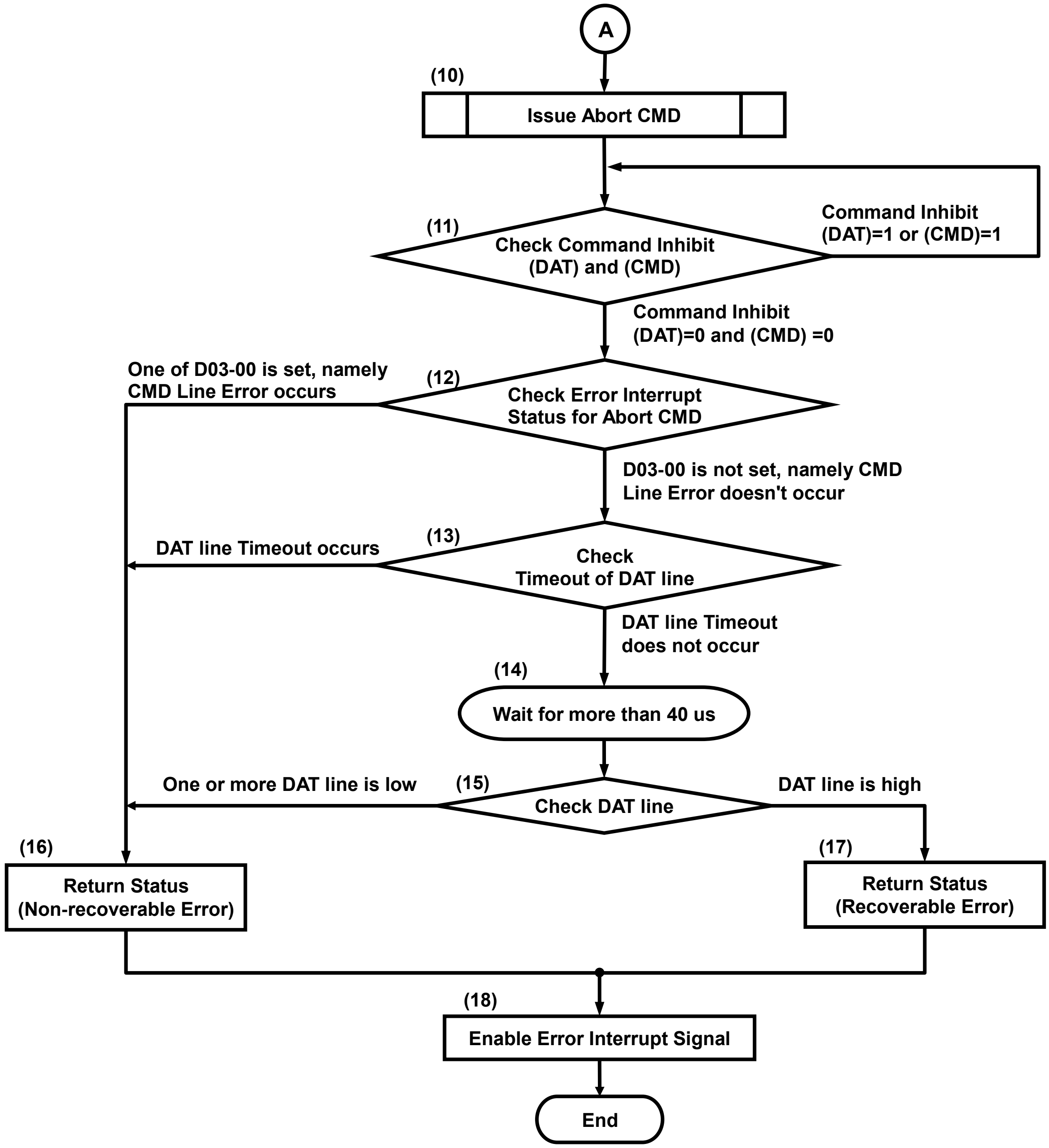


Figure 3-22: Error Interrupt Recovery Sequence

(1) Disable the Error Interrupt Signal.
(2) Check bits D03-00 in the *Error Interrupt Status* register. If one of these bits (D03-00) is set to 1, go to step (3). If none are set to 1 (all are 0), go to step (5).
(3) Set **Software Reset For CMD Line** to 1 in the *Software Reset* register for software reset of the CMD line.
(4) Check **Software Reset For CMD Line** in the *Software Reset* register. If **Software Reset For CMD Line** is 0, go to step (5). If it is 1, go to step (4).
(5) Check bits D06-04 in the *Error Interrupt Status* register. If one of these bits (D06-04) is set to 1, go to step (6). If none are set to 1 (all are 0), go to step (8).
(6) Set **Software Reset For DAT Line** to 1 in the *Software Reset* register for software reset of the DAT line.
(7) Check **Software Reset For DAT Line** in the *Software Reset* register. If **Software Reset For DAT Line** is 0, go to step (8). If it is 1, go to step (7).
(8) Save previous error status.
(9) Clear previous error status with setting them to 1.
(10) Issue Abort Command in accordance with Section 3.8.1.
(11) Check **Command Inhibit (DAT)** and **Command Inhibit (CMD)** in the *Present State* register. Repeat this step until both **Command Inhibit (DAT)** and **Command Inhibit (CMD)** are set to 0.
(12) Check bits D03-00 in the *Error Interrupt Status* register for Abort Command. If one of these bits is set to 1, go to step (16). If none of these bits are set to 1 (all are 0), go to step (13).
(13) Check **Data Timeout Error** in the *Error Interrupt Status* register. If this bit is set to 1, go to step (16). If it is 0, go to step (14).
(14) Wait for more than 40 us.
(15) By monitoring the **DAT [3:0] Line Signal Level** in the *Present State* register, judge whether the level of the DAT line is low or not. If one or more DAT lines are low, go to step (16). If the DAT lines are high, go to step (17).
(16) Return Status of "Non-recoverable Error".
(17) Return Status of "Recoverable Error".
(18) Enable the Error Interrupt Signal.

3.10.2 Auto CMD12 Error Recovery

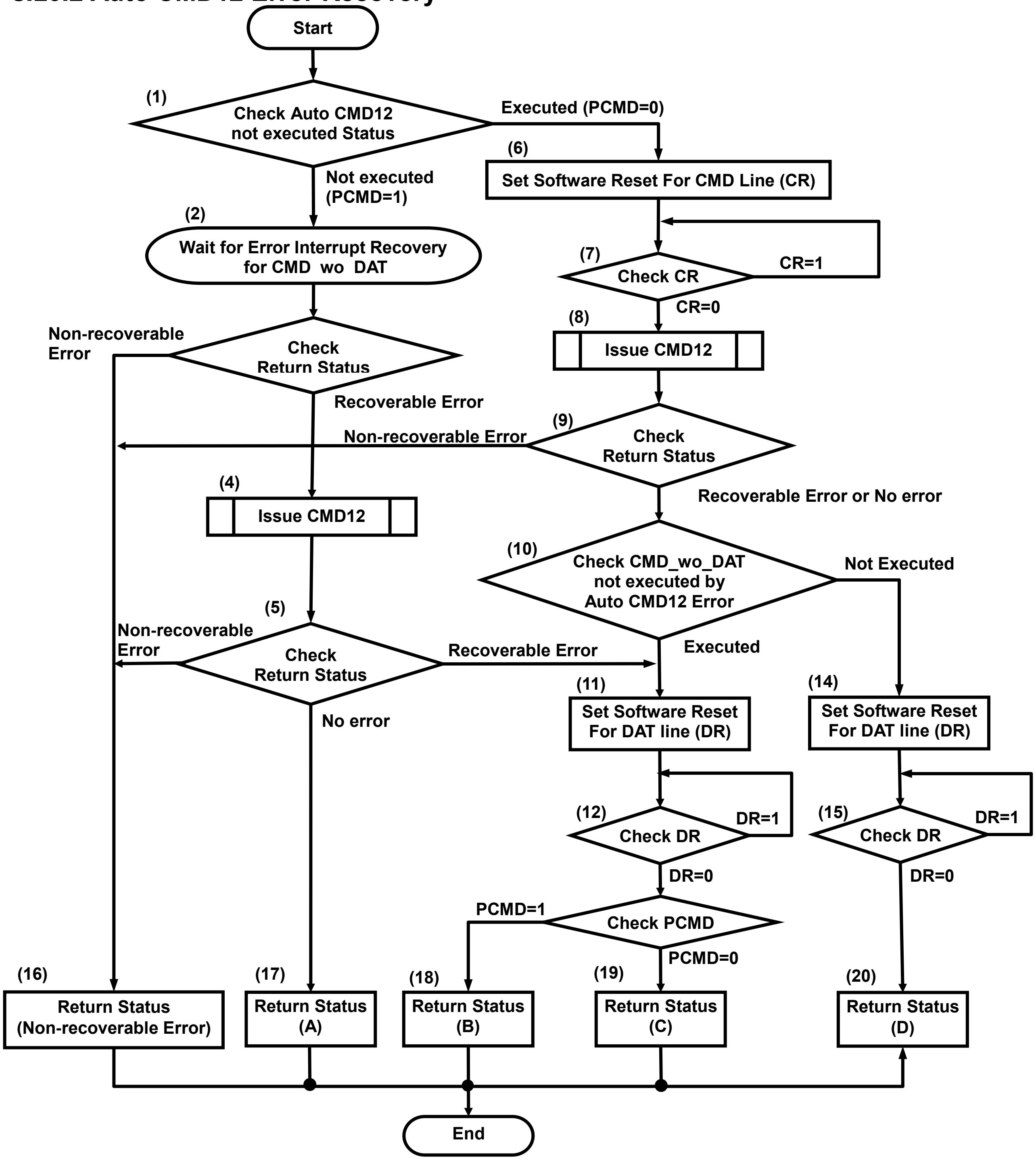


Figure 3-23 : Auto CMD12 Error Recovery Sequence

The sequence for Auto CMD12 Error Recovery is shown in Figure 3-23. Following four cases A to D shall be covered.

A: An error occurred in CMD_wo_DAT, but not in the SD memory transfer
B: An error occurred in CMD_wo_DAT, and also occurred in the SD memory transfer
C: An error did not occur in CMD_wo_DAT, but an error occurred in the SD memory transfer
D: CMD_wo_DAT was not issued, and an error occurred in the SD memory transfer

(1) Check **Auto CMD12 Not Executed** in the *Auto CMD Error Status* register. If this bit is set to 1, go to step (2). If this bit is set to 0, go to step (6). In addition, the Host Driver shall define **PCMD** flag, which changes to 1 if **Auto CMD12 Not Executed** is set to 1.
(2) Wait for Error Interrupt Recovery for CMD_wo_DAT.
(3) Check "Return Status". In the case of "Non-recoverable Error", go to step (16). In the case of "Recoverable Error", go to step (4).
(4) Issue CMD12 in accordance with Section 3.7.1
(5) If the ***CMD*** line errors occur for the CMD12 (One of D03-00 is set in the *Error Interrupt Status* register), "Return Status" is "Non-recoverable Error" and go to step (16). If not ***CMD*** line error and busy timeout error occur (D04 is set in the *Error Interrupt Status* register), "Return Status" is "Recoverable Error" and go to step (11). Otherwise, "Return Status" is "No error" and go to step (17).
(6) Set **Software Reset For CMD Line** to 1 in the *Software Reset* register for software reset of the CMD line.
(7) Check **Software Reset For CMD Line** in the *Software Reset* register. If **Software Reset For CMD Line** is 0, go to step (8). If it is 1, go to step (7).
(8) Issue CMD12 according to Section 3.7.1. Acceptance of CMD12 depends on the state of the card. CMD12 may make the card to return to tran state. If the card is already in tran state, the card does not response to CMD12.
(9) Check "Return Status" for CMD12. If "Return Status" returns "Non-recoverable Error", go to step (16). In the case of "Recoverable Error" or "No error", go to step (10).
(10) Check the **Command Not Issued By Auto CMD12 Error** in the *Auto CMD Error Status* register. If this bit is 0, go to step (11). If it is 1, go to step (14).
(11) Set **Software Reset For DAT Line** to 1 in the *Software Reset* register for software reset of the DAT line.
(12) Check **Software Reset For DAT Line** in the *Software Reset* register. If **Software Reset For DAT Line** is 0, go to step (13). If it is 1, go to step (12).
(13) Check the **PCMD** flag. If **PCMD** is 1, go to step (18). If it is 0, go to step (19).
(14) Set **Software Reset For DAT Line** to 1 in the *Software Reset* register for software reset of the DAT line.
(15) Check **Software Reset For DAT Line** in the *Software Reset* register. If **Software Reset For DAT Line** is 0, go to step (20). If it is 1, go to step (15).
(16) Return Status of "Non-recoverable Error".
(17) Return Status that an error has occurred in CMD_wo_DAT, but not in the SD memory transfer.
(18) Return Status that an error has occurred in both CMD_wo_DAT, and the SD memory transfer.
(19) Return Status that an error has not occurred in CMD_wo_DAT, but has occurred in the SD memory transfer.
(20) Return Status that CMD_wo_DAT has not been issued, and an error has occurred in the SD memory transfer.

3.11 Wakeup Control (Optional)

After the Host System goes into standby mode, the Host System can resume from standby via a wakeup event initiated by one of the following three events:

(1) Interrupt from an SD card:
If an SD card interrupt occurs, the Host System can resume from standby mode. If the Host System uses this wakeup factor, SD Bus power shall be kept on.
(2) Insertion of SD card:
If an SD card is inserted, the Host System can resume from standby mode.
(3) Removal of SD card:
If an SD card is removed, the Host System can resume from standby mode.

The sequence for preparing wakeup before the Host System goes into standby mode is shown in Figure 3-24.

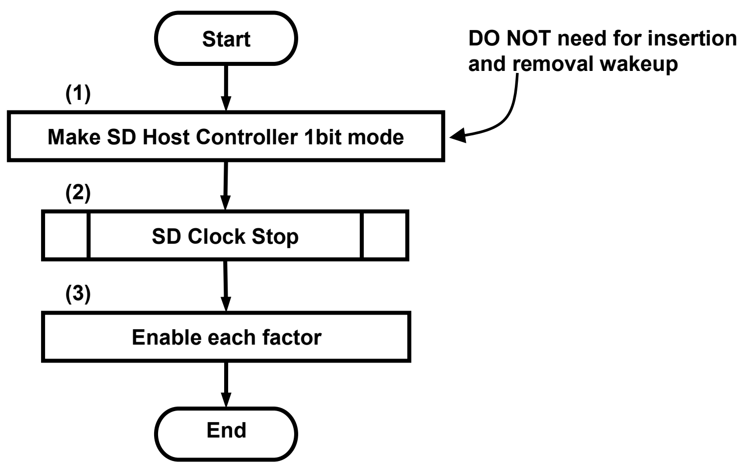


Figure 3-24: Wakeup Control before Standby Mode

(1) Set **Data Transfer Width** to 0 in the *Host Control 1* register.
(2) Execute SD Clock Stop Sequence as described Section 3.2.2.
(3) Clear the *Normal Interrupt Status* register and the *Normal Interrupt Signal Enable* register, and then set the enable bits of each wakeup event factor to 1 in the *Wakeup Control* register and set the bits of *Normal Interrupt Status Enable* register to use wakeup.

The sequence for wakeup once in standby mode is shown in Figure 3-25.

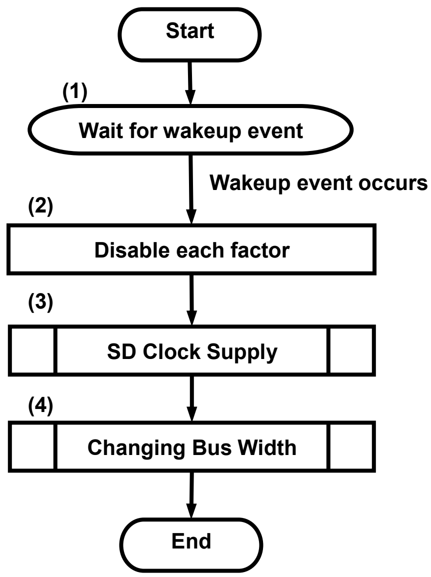


Figure 3-25: Wakeup from Standby

(1) Wait for wakeup event.
(2) Set the enable bits of each wakeup event factor to 0 in the *Wakeup Control* register and then clear event statuses in the *Normal Interrupt Status* register. If necessary, set the *Normal Interrupt Signal Enable* register.
(3) Execute SD Clock Supply Sequence as described Section 3.2.2.
(4) Set the SD Bus width in accordance with Section 3.4.

3.12 Suspend/Resume (Optional, Not Supported from Version 4.00)

If an SD card supports suspend and resume functionality, then the Host Controller can initiate suspend and resume. It is necessary for both the Host Controller and the SD card to support the function of "**Read Wait**". ADMA operation does not support this function.

3.12.1 Suspend Sequence

The sequence for suspend is shown in Figure 3-26.

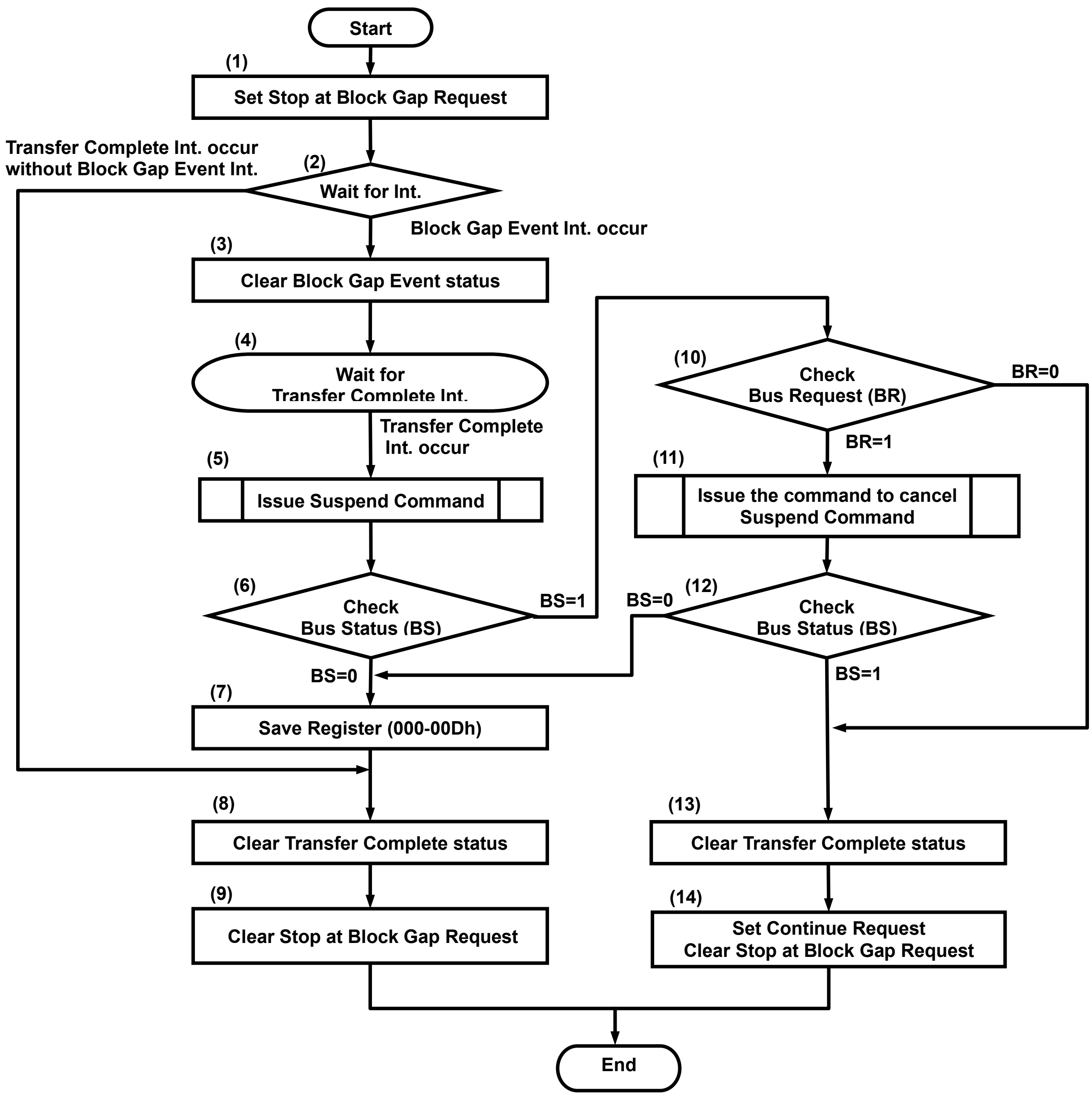


Figure 3-26 : The Sequence for Suspend

(1) Set **Stop At Block Gap Request** to 1 in the *Block Gap Control* register to stop the SD transaction.
(2) Wait for an Interrupt. If **Block Gap Event** is set to 0 and **Transfer Complete** is set to 1 in the *Normal Interrupt Status* register, go to step (8). If **Block Gap Event** is set to 1, go to step (3).
(3) Set **Block Gap Event** to 1 in the *Normal Interrupt Status* register to clear this bit.
(4) Wait for the **Transfer Complete** Interrupt.
(5) Issue the Suspend Command in accordance with Section 3.7.1.
(6) Check the **BS** value of the response data. If **BS** is 0, go to step (7). If **BS** is 1, go to step (10).
(7) Save the register (000h-00Dh).
(8) Set Transfer Complete to 1 in the *Normal Interrupt Status* register to clear this bit.
(9) Set **Stop At Block Gap Request** to 0 in the Block Gap Control register to clear this bit.
(10) Check the **BR** value of the response data. If **BR** is 1, go to step (11). If **BR** is 0, go to step (13).
(11) Issues the command to cancel the previous suspend command in accordance with Section 3.7.1 Transaction Control without **Data Transfer Using DAT Line**.
(12) Check the **BS** value of the response data. If BS is 0, go to step (7). If **BS** is 1, go to step (13).
(13) Set **Transfer Complete** to 1 in the *Normal Interrupt Status* register to clear this bit.
(14) Set **Continue Request** to 1 in the Block Gap Control register to continue the transaction. At the same time, write 0 to **Stop At Block Gap Request** to clear this bit.

The Table 3-1 shows conditions to be able to use Suspend / Resume function.

Conditions			**Suspend/Resume Function**	
Host Suspend/Resume Support	**Card Suspend/Resume Support**	**Card Read Wait Support**	**Write Suspend/Resume**	**Read Suspend/Resume**
Not supported	Don't care	Don't care	Cannot be used	Cannot be used
Supported	Not supported	Don't care	Cannot be used	Cannot be used
Supported	Supported	Not supported	Can be used	Cannot be used
Supported	Supported	Supported	Can be used	Can be used

Table 3-1 Suspend/Resume Condition

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether the Host Controller issues a Suspend command or the SD card accepts the Suspend command.

(1) If the Host Driver does not issue a Suspend command, the **Continue Request** shall be used to restart the transfer.
(2) If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command shall be used to restart the transfer.
(3) If the Host Driver issues a Suspend command and the SD card does not accept it, the **Continue Request** shall be used to restart the transfer.

3.12.2 Resume Sequence

The sequence for resume is shown in Figure 3-27.

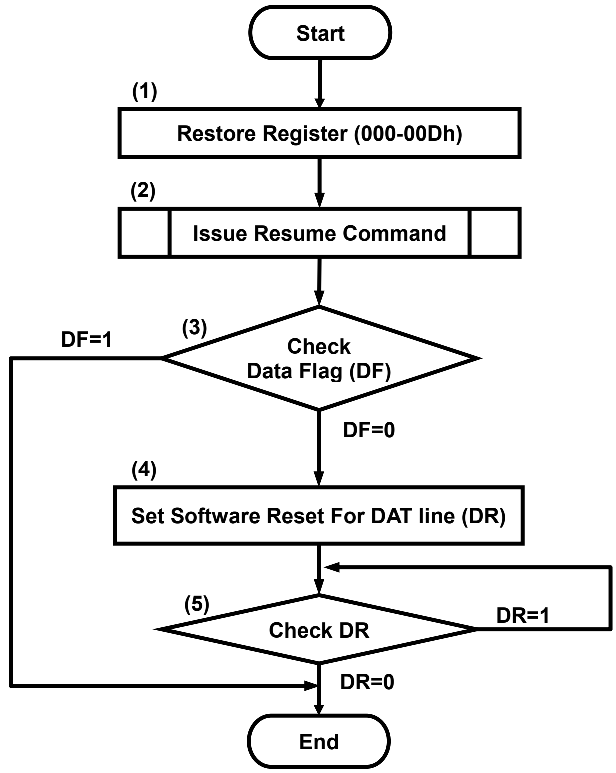


Figure 3-27 : The Sequence for Resume

(1) Restore the register (000h-00Dh).
(2) Issue the Resume Command in accordance with Section 3.7.1.
(3) Check the **DF** value of the response data. If **DF** is 0, go to step (4). If **DF** is 1, go to 'End'.
(4) Set **Software Reset For DAT Line** to 1 in the *Software Reset* register for software reset of the DAT line.
(5) Check **Software Reset For DAT Line** in the *Software Reset* register. If **Software Reset For DAT Line** is 0, go to 'End'. If it is 1, go to step (5).

3.12.3 Stop At Block Gap / Continue Timing for Read Transaction

Figure 3-28 to Figure 3-33 show the timing of **Stop At Block Gap Request** and **Continue Request** for non-DMA transfer. The **Transfer Complete** interrupt is always generated by setting **Stop At Block Gap Request** where data transfer is stopped. However, generation of the **Block Gap Event** interrupt is dependent on whether the last data block is sent or not. **Block Gap Event** is not generated If all data blocks are transferred (the last block is transferred). It is not necessary to enable **Block Gap Event** interrupt. The status can be checked when transfer complete interrupt is detected. It is not necessary to use **Continue Request** if **Block Gap Event** status is not set because there is no further data to be transferred.
Read Wait was required to issue a command during suspend. As support of Suspend/Resume is not required from Version 4.00, currently Read Wait is not the requirement, or rather; **Stop At Block Gap Request** and **Continue Request** shall be supported. If Read Wait is not supported, Host Controller uses SD Clock to stop read data transfer at the block gap by stopping SD Clock. Read Wait timing is described in figures of Section 3.12.3 and Section 3.12.4 but Read Wait =1 can be interpreted as the timing where SD Clock is stopped.

Implementation Note: Read Wait, **DAT Line Active** and **Read Transfer Active** are set and cleared by the Host Controller. **Stop At Block Gap Request** is set and cleared by the Host Driver. **Continue Request** is set by the Host Driver and be cleared by the Host Controller. **Block Gap Event** and **Transfer Complete** are set by the Host Controller and are cleared by the Host Driver.

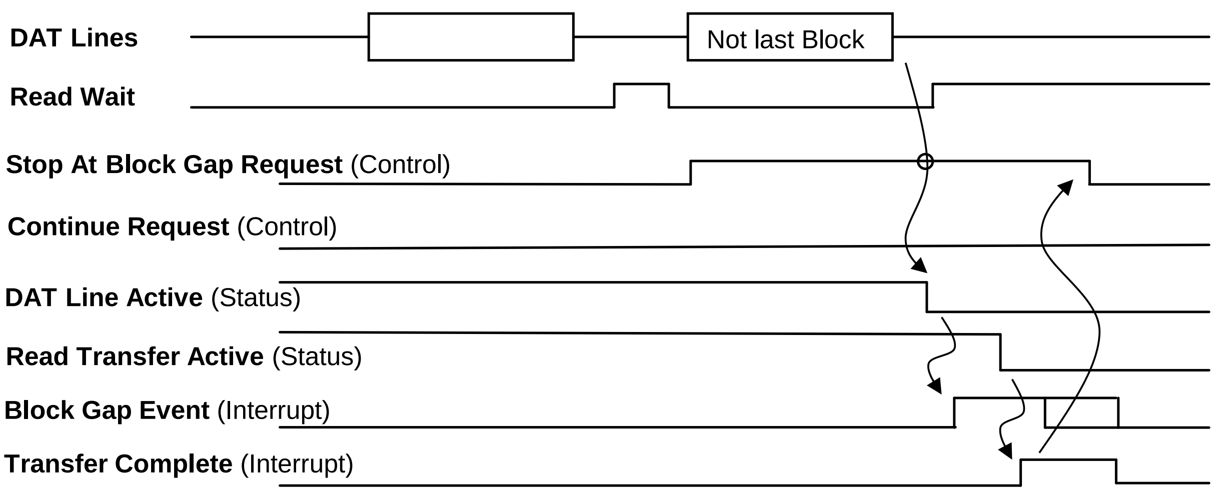


Figure 3-28 : Wait Read Transfer by Stop At Block Gap Request

Execution Steps of the Stop At Block Gap Request:

(1) Clear **DAT Line Active** status and generate the **Block Gap Event** Interrupt after completion of a data block transfer on SD Bus which is not the last data block. Data read in Host Controller is still transferring to system memory.
(2) Clear the **Read Transfer Active** status when data transfer to system memory is ready to stop and generate the **Transfer Complete** Interrupt when execution of Stop At Block Gap Request is completed.
(3) On accepting **Transfer Complete** Interrupt, Host Driver clears the **Stop At Block Gap Request.**

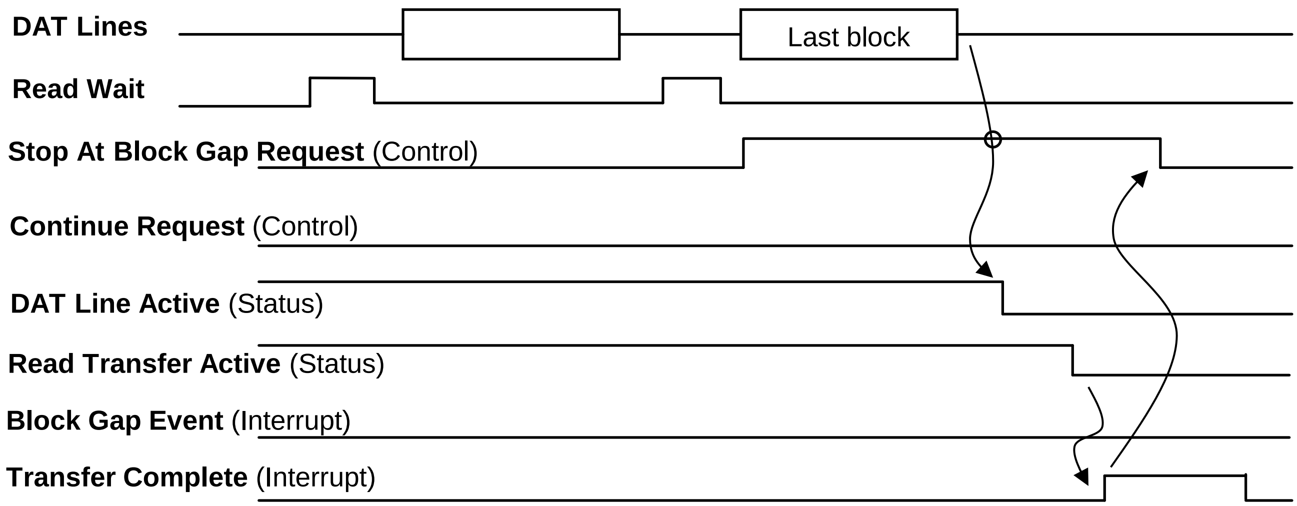


Figure 3-29 : Stop At Block Gap Request is Not Accepted at the Last Block of the Read Transfer

If data transfer cannot be stopped before the last data block transfer, the Host Controller cannot accept the **Stop At Block Gap Request** and stops the transaction normally. The **Block Gap Event** Interrupt is not generated. On accepting the **Transfer Complete** Interrupt, Host Driver clears the **Stop At Block Gap Request**.

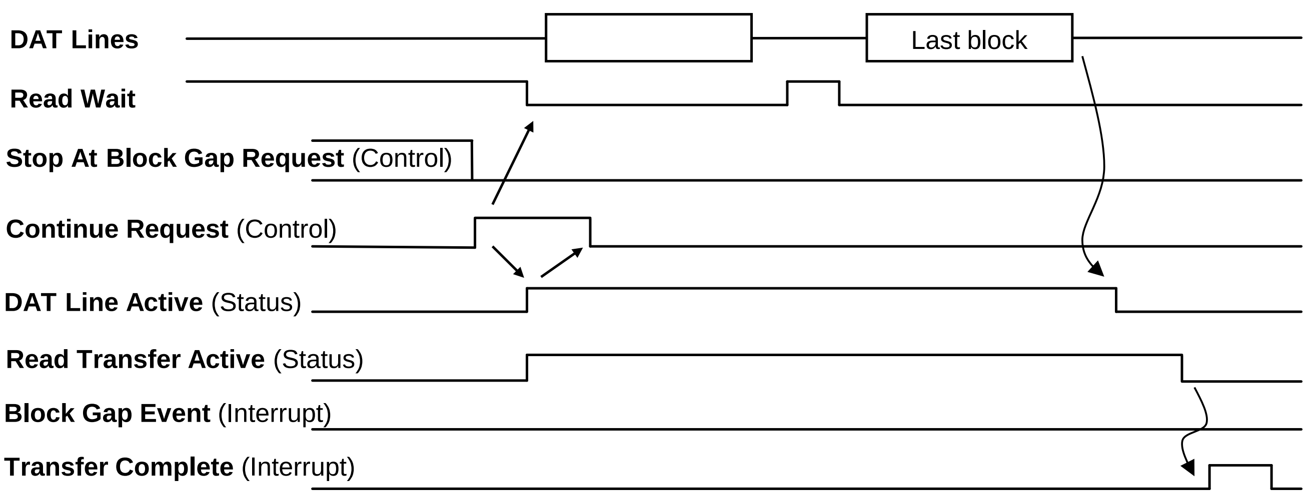


Figure 3-30 : Continue Read Transfer by Continue Request

To restart a stopped data transfer, Host Driver sets the **Continue Request** to 1. (The **Stop At Block Gap Request** shall be set to 0.)
On accepting the **Continue Request**, Host Controller executes followings:

(1) Release Read Wait
(2) Set the **DAT Line Active** status and the **Read Transfer Active** status
(3) The **Continue Request** is automatically cleared by (2).

After the last data block is transferred on SD Bus, Host Controller executes followings:

(1) Clear the **DAT Line Active** status and do not generate the **Block Gap Event** Interrupt.
(2) After all valid data has been read (No valid read data remains in the Host Controller), clear the

Read Transfer Active status and generate the **Transfer Complete** Interrupt.

3.12.4 Stop At Block Gap / Continue Timing for Write Transaction

Implementation Note:
DAT Line Active and **Write Transfer Active** are set and cleared by the Host Controller.
Stop At Block Gap Request is set and cleared by the Host Driver.
Continue Request is set by the Host Driver and is cleared by the Host Controller.
Block Gap Event and **Transfer Complete** are set by the Host Controller and are cleared by the Host Driver.

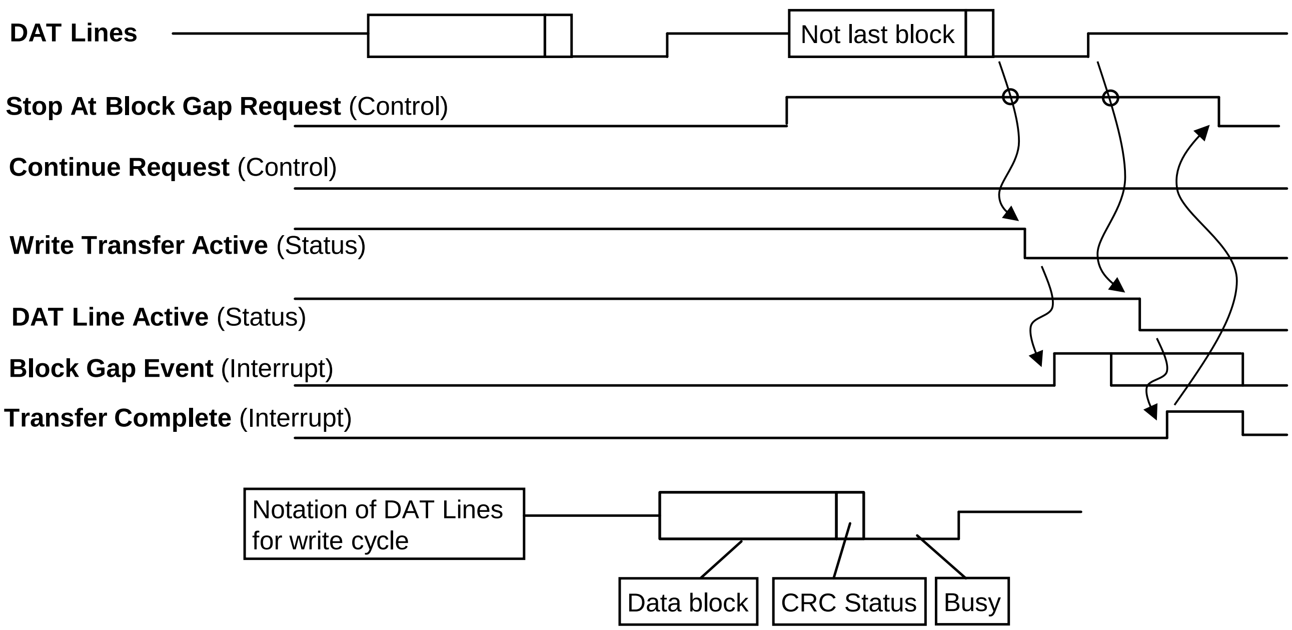


Figure 3-31 : Wait Write Transfer by Stop At Block Gap Request

Execution Steps of the **Stop At Block Gap Request**

(1) Clear the **Write Transfer Active** Status and generate the **Block Gap Event** Interrupt after completion of a data block transfer on SD Bus which is not the last data block.
(2) After the busy signal is released, clear the **DAT Line Active** status and generate the **Transfer Complete** Interrupt.
(3) On accepting the **Transfer Complete** Interrupt, Host Driver clears the **Stop At Block Gap Request**

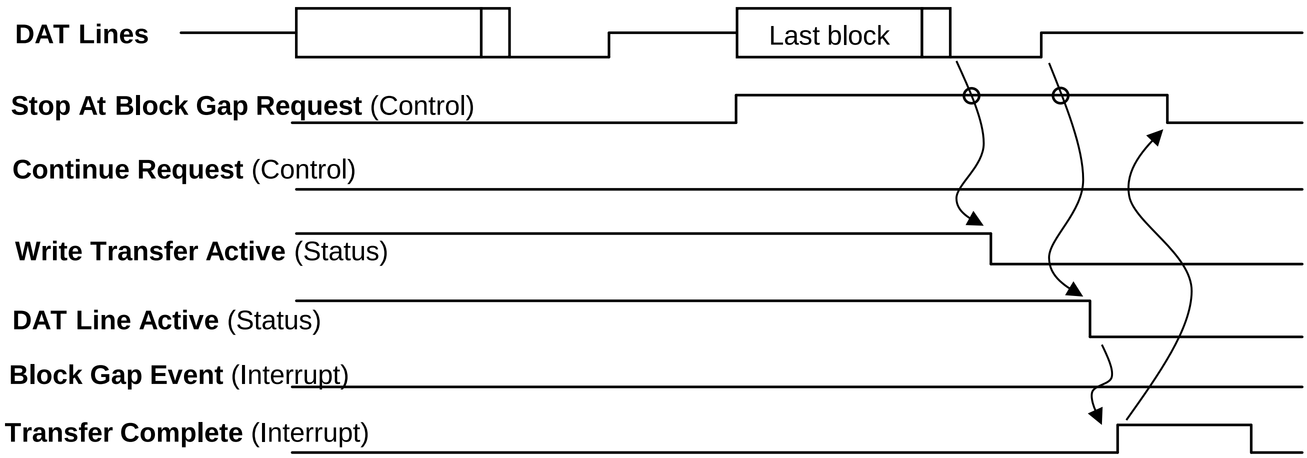


Figure 3-32 : Stop At Block Gap Request is Not Accepted at the Last Block of the Write Transfer

If data transfer cannot be stopped before the last data block transfer, the Host Controller cannot accept the **Stop At Block Gap Request** and terminates the transaction normally. The **Block Gap Event** Interrupt is not generated. On accepting the **Transfer Complete** Interrupt, Host Driver clears the **Stop At Block Gap Request**.

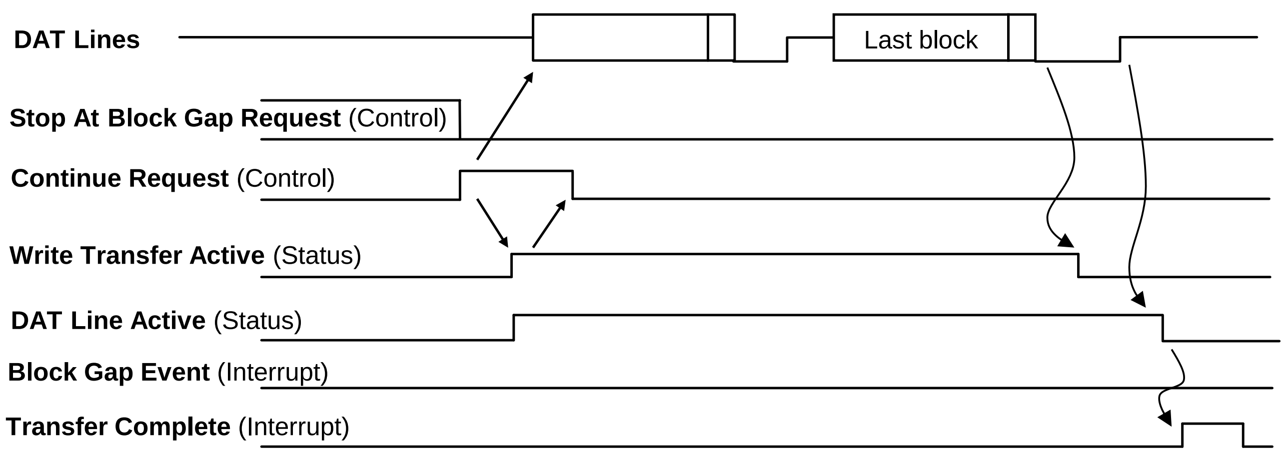


Figure 3-33 : Continue Write Transfer by Continue Request

To restart a stopped data transfer, set the **Continue Request** to 1. (**Stop At Block Gap Request** shall be set to 0.)

On accepting the **Continue Request,** Host Controller executes followings:
(1) Set the **DAT Line Active** status and the **Write Transfer Active** Status
(2) The **Continue Request** is automatically cleared by (1).

After the last data block is transferred on SD Bus, Host Controller executes followings:
(1) After all valid data has been written (No valid write data remains in the Host Controller), clear the **Write Transfer Active** Status, and do not generate the **Block Gap Event** Interrupt
(2) After the busy signal is released, clear the **DAT Line Active** status and generates the **Transfer Complete** Interrupt.

3.13 UHS-II Operation

In this section, the sequence to issue UHS-II command, to receive Response and MSG packet is described.

3.13.1 Host Controller Setup Sequence

This setup sequence includes Version 3.00 and 4.00 features. Register parameters set in this procedure keep values regardless of Bus Speed Mode changes.

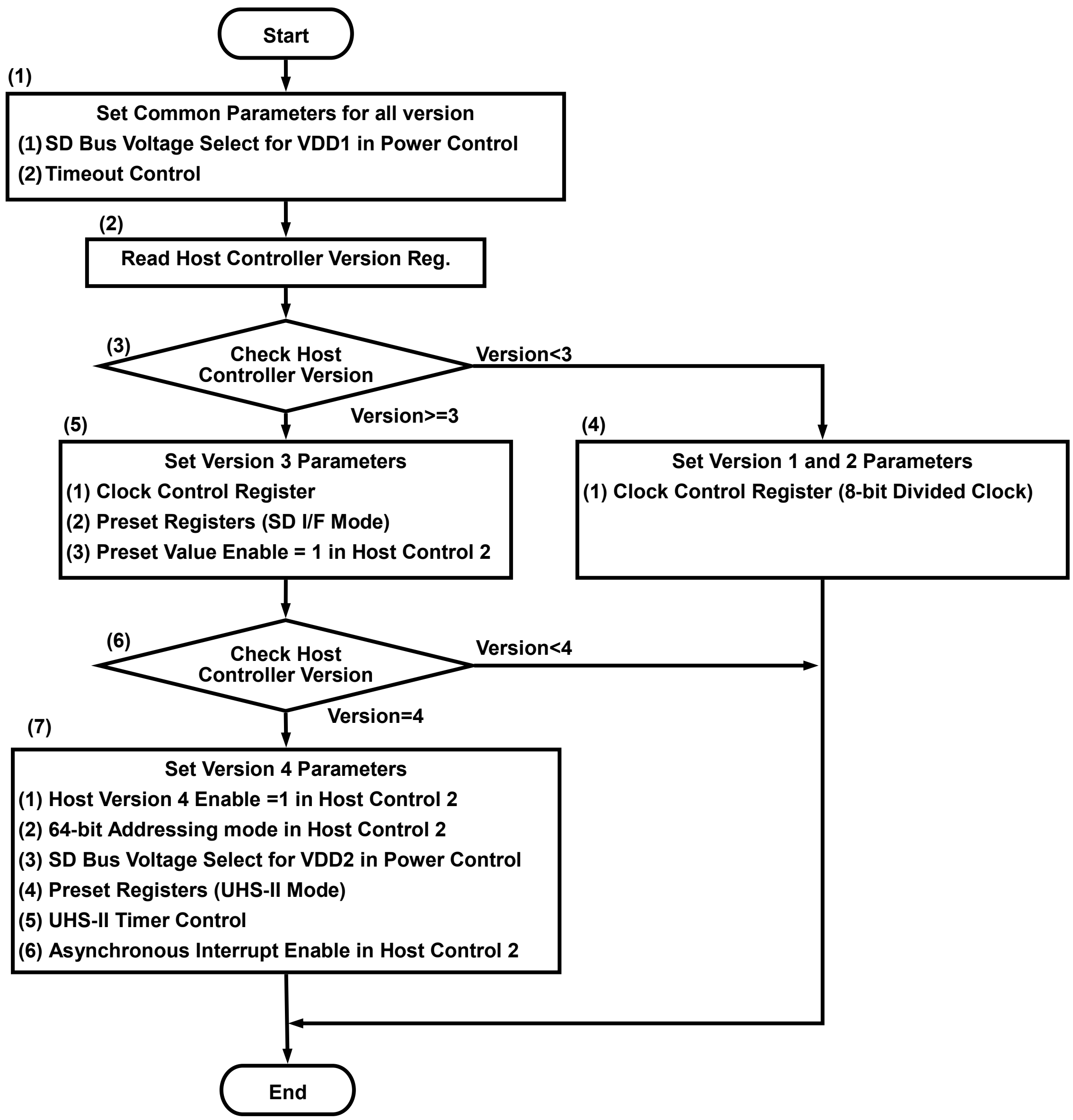


Figure 3-34 : Host Controller Setup Sequence

(1) Set parameters for all Host Controller versions. Set **SD Bus Voltage Select for** VDD1 in the *Power Control* register and **Data Timeout Counter Value** in the *Timeout Control* register.
(2) Read the *Host Controller Version* register and check **Specification Version Number**.
(3) If Specification Version number is less than version 3, go to step (4), if it is version 3 or later go to step (5).
(4) Set *Clock Control* register using 8-bit Divided Clock mode.
(5) Set Version 3 parameters. *Clock Control* register is sets in 10-bit Divided Clock Mode or Programmable Clock Mode. If **Clock Multiplier** in the *Capabilities* register is not zero, Programmable Clock Mode should be used. If Preset Value is used, set Preset Values of SD I/F Modes in the *Preset Value* register and set **Preset Value Enable** to 1 in the *Host Control 2* register.
(6) If Specification Version number is version 4, go to step (7), if it is less than version 4, exits.
(7) Set Version 4 parameters. Set **Host Version 4 Enable** to 1. If the **64-bit System Address Support** in the *Capabilities* register is set to 1, set **64-bit Addressing** to 1 in the *Host Control 2* register. If **UHS-II Support** is set to 1 and **1.8V VDD2 Support** is set to 1 in the *Capabilities* register, set **SD Bus Voltage Select for VDD2** to 1.8V mode in the *Power Control* register, set preset value for UHS-II Mode to *Preset Value* register and set **Timeout Counter Value for CMD_RES** and **Timeout Counter Value for Deadlock** in the *UHS-II Timeout Control* register based on **Timeout Clock Frequency** and **Timeout Clock Unit** in the *Capabilities* register. If **Asynchronous Interrupt Support** in the *Capabilities* register is set to 1, set **Asynchronous Interrupt Enable** is set to 1 in the *Host Control 2* register.

3.13.2 Card Interface Detection Sequence

This procedure is invoked by the Card Insertion interrupt. It is assumed that timings of this flow chart are based on P2P connection between host and removable card.

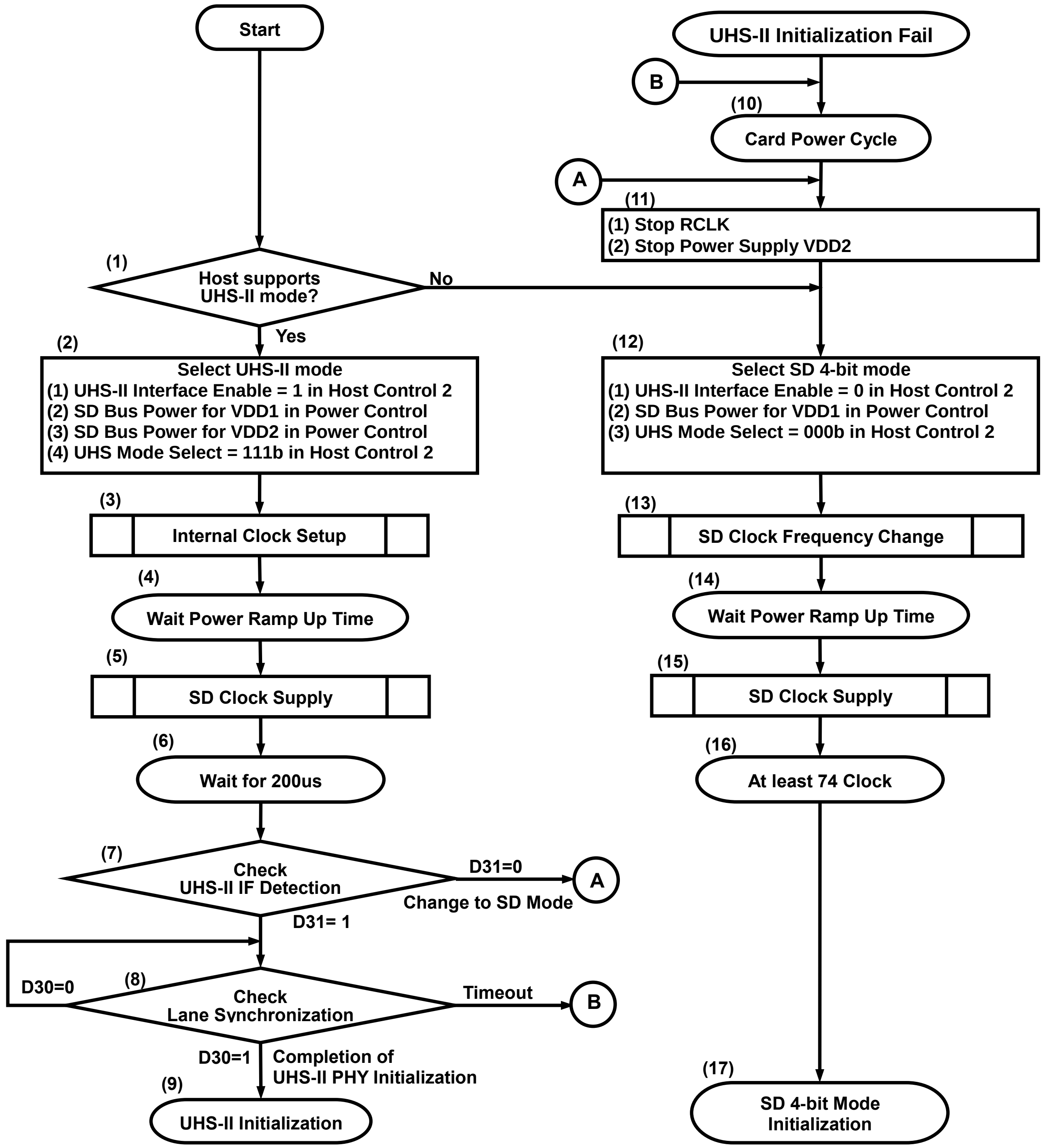


Figure 3-35 : Card Interface Detection Sequence

(1) If Host Supports UHS-II, go to step (2) else go to step (12).
(2) Try to initialize a card to UHS-II mode. Set **UHS-II Interface Enable** to 1 in the *Host Control 2* register, set **SD Bus Power for VDD1** and **SD Bus Power for VDD2** to 1 in the *Power Control* register, and set **UHS Mode Select** to 111b in the *Host Control 2* register.
(3) Execute Internal Clock Setup Sequence as described Section 3.2.1.
(4) Wait power ramp up time. It is dependent on a Host System.
(5) Execute SD Clock Supply Sequence as described Section 3.2.2. Host Controller should provide STB.L to D0 lane immediately.
(6) Wait 200us to detect support of UHS-II mode.
(7) If **UHS-II IF Detection** (D31) of the *Present State* register is set to 0, go to step (11) to start SD mode Initialization. **UHS-II IF Detection** (D31) is set to 1, go to step (8) to check completion of PHY Initialization.
(8) Check **Lane Synchronization** (D30) of the *Present State* register. If D30=1, UHS-II PHY Initialization is completed and go to step (9). D30=0 means that PHY is under initializing. The timeout while waiting D30=0 is defined as 150ms (Tactivate + Tlidl_lidl). If timeout is detected, which means PHY initialization failure occurs, go to step (10) to try to start SD mode initialization.
(9) Perform UHS-II initialization. Refer to Section 0 UHS-II Card Initialization and Section 3.13.4 UHS-II Setting Register Setup Sequence.
(10) If Host Driver gives up UHS-II Initialization due to an error occurs during UHS-II Initialization, Host Driver executes power cycle before starting SD mode initialization.
(11) If changing form UHS-II setup to SD Mode setup, stop supplying RCLK and VDD2.
(12) Try to initialize a card to SD 4-bit mode. Set **UHS-II Interface Enable** to 0 in the *Host Control 2* register, set **SD Bus Power for VDD1** to 1 in the *Power Control* register, and set **UHS Mode Select** to 000b in the *Host Control 2* register.
(13) Execute SD Clock Frequency Change Sequence as described Section 3.2.3.
(14) Wait power ramp up time. It is dependent on a Host System. If Host Driver initialize the card in SD 4-bit mode due to STB.L is not detected at step (8), it is not necessary to execute power cycle and then it is not necessary to wait power ramp up time in this step.
(15) Execute SD Clock Supply Sequence as described Section 3.2.2.
(16) Provide at least 74 clocks before issuing SD Command (CMD0).
(17) Perform SD 4-bit mode initialization. Refer to Section 3.6 Card Initialization and Identification (for SD I/F).

Implementation Guideline

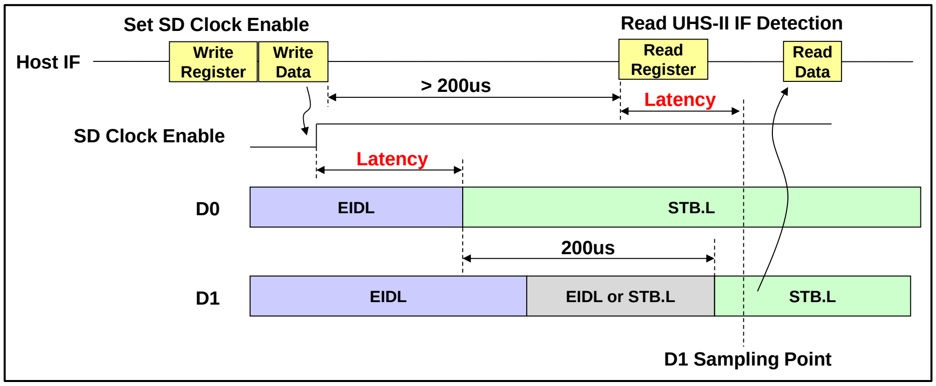


Figure 3-36 : Latency Compensation for Reading UHS-II IF Detection

Figure 3-36 shows timing of latency compensation. UHS-II initialization is started by setting **SD Clock Enable** and 200us later, Host Driver checks **UHS-II IF Detection**. There is latency from setting **SD Clock Enable** to Host Controller provides STB.L to D0 lane depends on implementation. Then when Host Controller receives read operation of **UHS-II IF Detection** bit in the *Present State* register (Step (8) of Figure 3-35), sampling timing of D1 lane shall be shifted during the register read operation to compensate the latency.
To minimize the latency, Host Controller completes clock synchronization before **SD Clock Enable** is set.

Checking receipt of correct SYN and LIDL symbols on D1 lane is a necessary condition for verifying PHY Initialization but it will not be sufficient. Then adding the other checkpoints of PHY Initialization is up to implementation. Host Controller may indicate PHY Initialization Frailer by own criteria.

3.13.3 UHS-II Card Initialization

UHS-II Card initialization flow chart for host is described in Figure 6-24 of the UHS-II Addendum. **DAP** (Device Allocated Power) is dependent on Host System capability, which is set to **DAP** in the *UHS-II Host Capabilities* register. By allowing higher power consumption during UHS-II initialization, initialization time may shorten. **GAP** (Group Allocated Power) is used for multiple UHS-II embedded devices initialization. GAP indicates maximum power supply capability of host and set to *GAP* in the *UHS-II Host Capabilities* register. Multiple devices can be initialized at the same time in the range of GAP. Host Driver set DAP and GAP in the argument of DEVICE_INIT command and repeat issuing DEVICE_INIT command until CF (Completion Flag) is set to 1 in the DEVICE_INIT command received from the card or device.

3.13.4 UHS-II Settings Register Setup Sequence

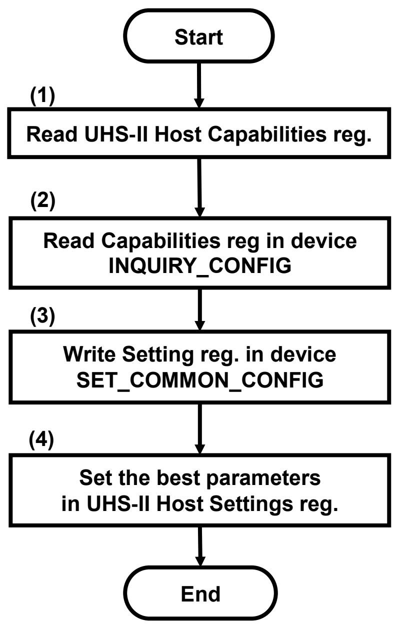


Figure 3-37 : UHS-II Settings Register Setup Sequence

(1) Read *UHS-II Host Generic, PHY and LINK/TRAN Capabilities* register and check capability in host system. Host capabilities are initial values set in the INQUIRY_CONFIG command.
(2) Read Generic, *PHY and LINK/TRAN Capabilities* register in device and check capability in device. Received INQUIRY_CONFIG packet indicates the best parameters that are acceptable for the host and all devices.
(3) The determined parameters are also set to device settings register of all devices by issuing SET_COMMON_CONFIG command.
(4) The determined parameters are set to the *UHS-II Host Settings register.*

3.13.5 UHS-II CCMD Packet Issuing

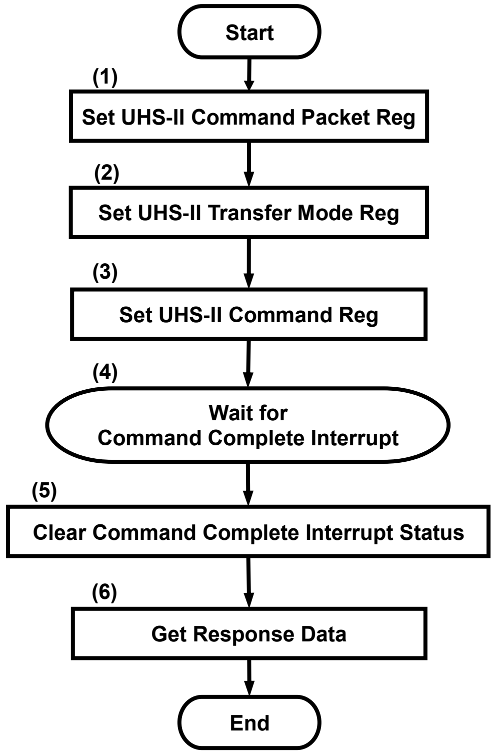


Figure 3-38 : UHS-II CCMD Packet Issuing

(1) Set Header, Argument and Payload in the *UHS-II Command Packet* register.
(2) Set required parameters in the *UHS-II Transfer Mode* register.
(3) Set packet length in the *UHS-II Command* register. Once packet length is programmed, a UHS-II command will be issued.
(4) Wait for **Command Complete** Interrupt in the *Normal Interrupt Status register*.
In case of Broadcast CMD, the issued command will be stored in the *UHS-II Response* register and **Command Complete** Interrupt will be asserted.
(5) Write 1 to **Command Complete** in the *Normal Interrupt Status* register to clear this bit.
(6) If necessary, read the *UHS-II Response* register and get necessary information of the issued command.

3.13.6 UHS-II DCMD Packet Issuing

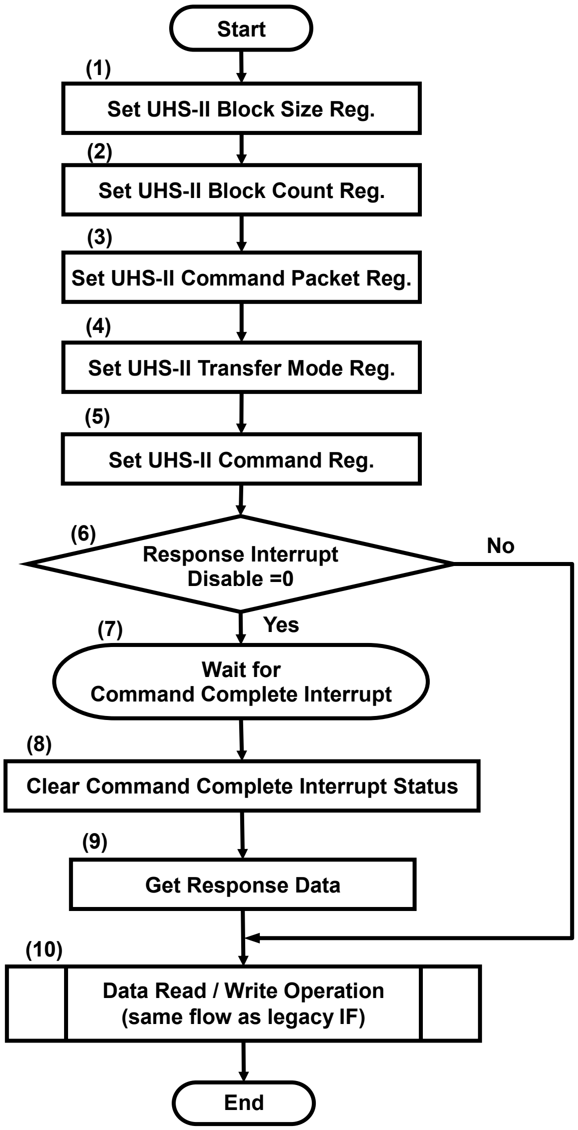


Figure 3-39 : UHS-II DCMD Packet Issuing

(1) et block size in the *UHS-II Block Size* register.
(2) Set block count in the *UHS-II Block Count* register.
(3) Set Header, Argument and Payload in the *Command Packet* register.
(4) Set required parameters in the *UHS-II Transfer Mode* register.
(5) Set packet length in the *UHS-II Command* register. Once packet length is programmed, a UHS-II command will be issued.
(6) If **Response Interrupt Disable**=1 in the *UHS-II Transfer Mode* register, go to step (10). In this case, **Response Error Check Enable** in the *Command Mode* register is set to 1 and **Response Error** Interrupt is enabled.
(7) Wait for **Command Complete** Interrupt.
(8) Write 1 to **Command Complete** in the *Normal Interrupt Status* register to clear this bit.
(9) If necessary, read the *UHS-II Response* register and get necessary information of the issued command.

(10) The following sequence is same as the sequence of after **Command Complete** in non-DMA, SDMA and ADMA operation.

3.13.7 Data Transfer Using ADMA3

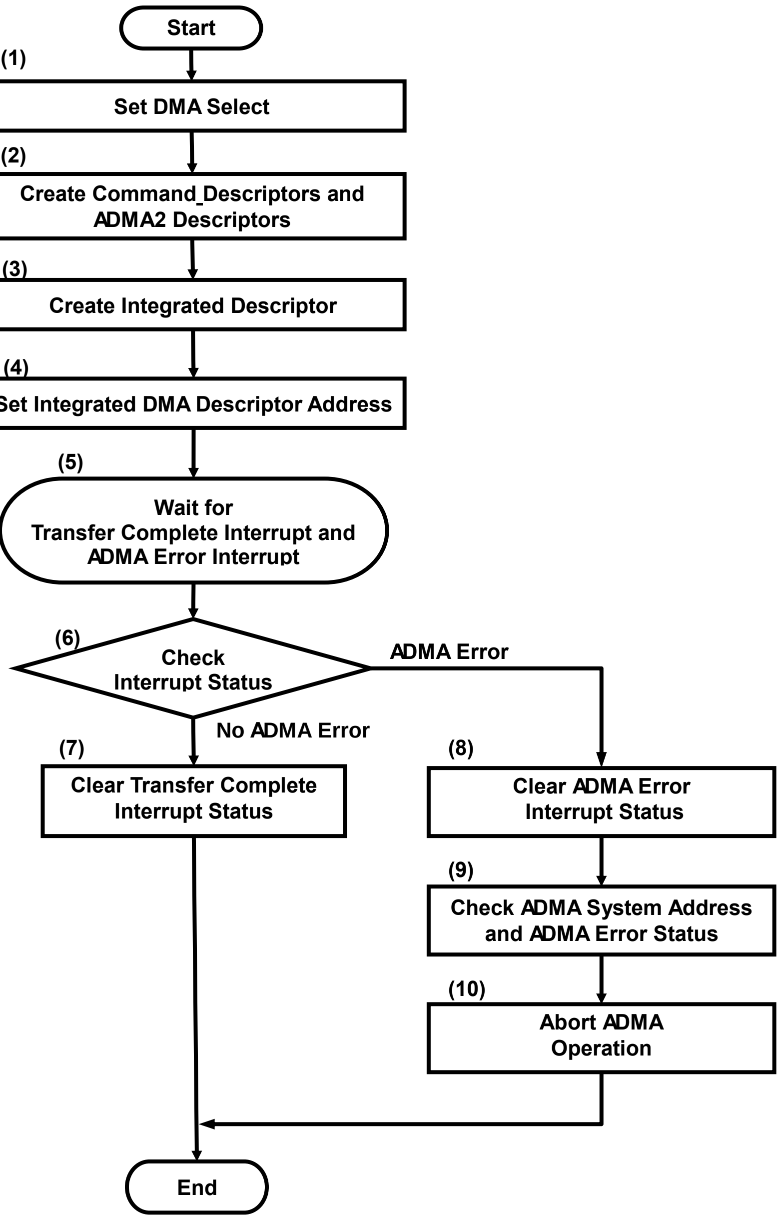


Figure 3-40 : Data Transfer Using ADMA3

(1) Set **DMA Select** field in the *Host Control 1* register to 11 to enable ADMA3.
(2) Create Command Descriptors and ADMA2 Descriptors in the system memory.
(3) Create Integrated Descriptor for ADMA3 in the *system memory.*
(4) Set the *Integrated Descriptor Address* register. Writing to this register starts ADMA3 data transfer.
(5) Wait for the **Transfer Complete** Interrupt and **ADMA Error Interrupt**.
(6) If **Transfer Complete** is set to 1, go to Step (7) else if **ADMA Error Interrupt** is set to 1, go to Step (8).
(7) Write 1 to the **Transfer Complete Status** in the *Normal Interrupt Status* register to clear this bit.
(8) Read *ADMA System Address* register and *ADMA Error Status* register to confirm in which descriptor ADMA error is occurred.
(9) Write 1 to the **ADMA Error Interrupt Status** in the *Error Interrupt Status* register to clear this bit.
(10) Abort ADMA operation. SD card operation should be stopped by issuing abort command. If necessary, Host Driver checks *ADMA Error Status* register to detect why **ADMA error** is asserted.

3.13.8 Entering Dormant or Hibernate Mode

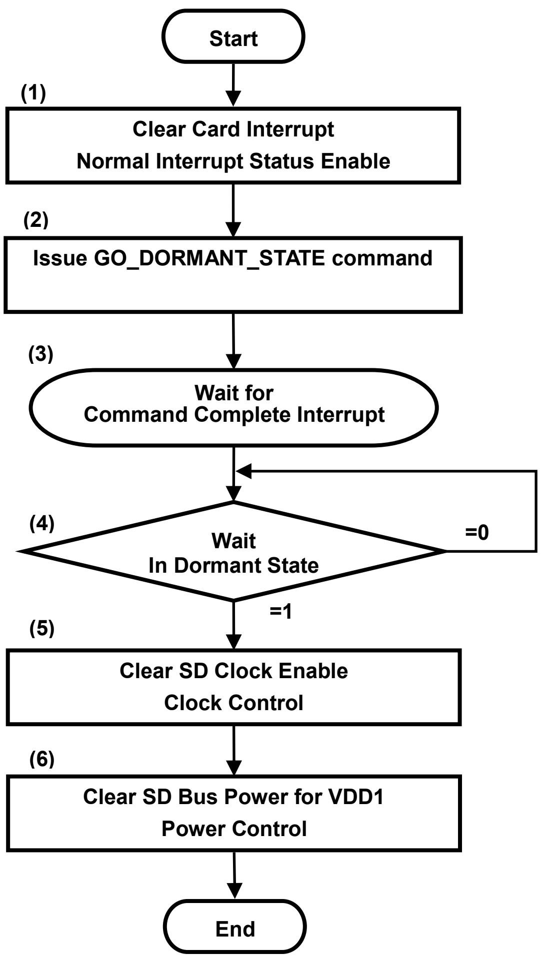


Figure 3-41 : Entering Dormant or Hibernate Mode

(1) Clear **Card Interrupt** in the *Normal Interrupt Status Enable* register if Card Interrupt is using.
(2) Issue GO_DORMANT_STATE command.
(3) Wait **Command Complete** interrupt.
(4) Wait until **In Dormant State** bit is set to 1 in the *Present State* register by polling. If **In Dormant State** bit is set to 1, go to step (5). Exit polling loop by timeout.
(5) RCLK may be stopped in dormant state by clearing **SD Clock Enable** in the *Clock Control* register. Stopping RCLK means that Host Controller drives ***RCLK*** differential line to DIF-PD. If Card Interrupt is enabled, **IENx** in CCCR of the Card shall be cleared before clearing **SD Clock Enable**.
(6) If the command of step (2) is set to Hibernate mode, clear **SD Bus Power for VDD1** in the *Power Control* register. If not in Hibernate mode, keep supplying VDD1.

3.13.9 SD-TRAN Reset Issuing Sequence

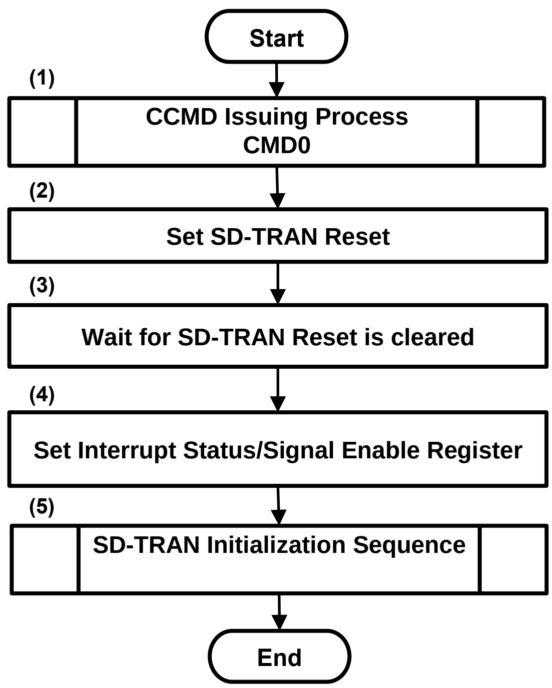


Figure 3-42 : SD-TRAN Reset Issuing Sequence

(1) CM0 (SD-TRAN Reset) is issued by using “CMD Issuing Process”.
(2) Set **Host SD-TRAN Reset** in the *UHS-II Software Reset* register.
(3) Wait for **Host SD-TRAN Reset** is cleared by Host Controller
(4) Set *Interrupt Status* / *Interrupt Signal Enable* register
(5) Execute “SD-TRAN Initialization Sequence” as described in Figure 3-20 of the Part 1 Physical Layer Specification Version 4.10

3.13.10 Host Full Reset Issuing Sequence

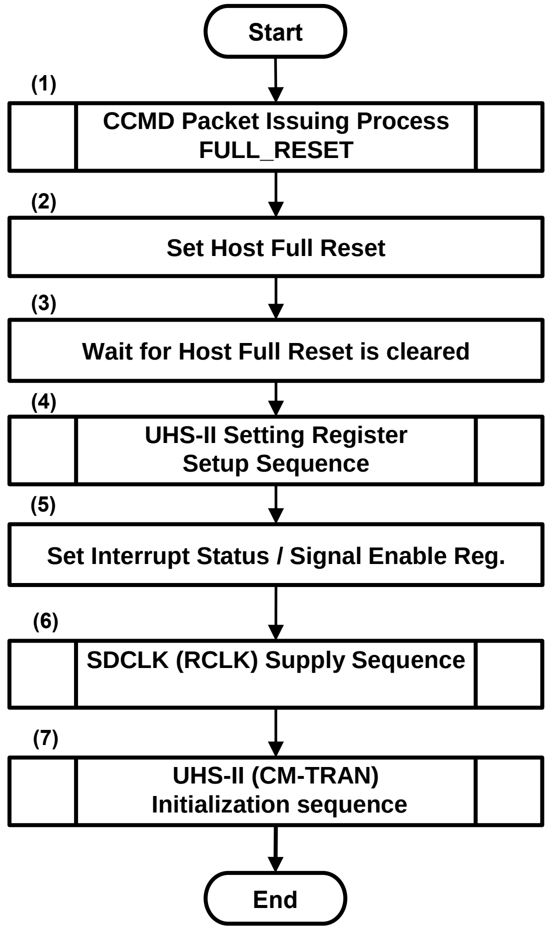


Figure 3-43 : Host Full Reset Issuing Sequence

(1) FULL_RESET CCMD is issued by using "CCMD Issuing Process".
(2) Set **Host Full Reset** in the *UHS-II Software Reset* register. **SD Bus Power** shall be kept 1 regardless of Full Reset.
(3) Wait for **Host Full Reset** is cleared by Host Controller
(4) Execute UHS-II Setting Register Setup Sequence.
(5) Set *Interrupt Status* / *Interrupt Signal Enable* register
(6) Execute SDCLK (RCLK) Supply Sequence
(7) Execute CM-TRAN Initialization Sequence, which is started from PHY Initialization, as described in the Figure 3-20 of Part 1 Physical Layer Specification Version 4.10.

Appendix A (Normative) : Reference

A.1 Reference

This specification refers extensively to any released version of the following SD specifications and the related Supplementary Notes.

SD Specifications Part 1 Physical Layer Specification Version 4.10 or later

SD Specifications Part 1 UHS-II Addendum Version 1.00 or later

SD Specifications Part 1 eSD Addendum Version 2.10 or later

SD Specifications Part 2 File System Specification Version 3.00 or later

SD Specifications Part 3 File Security Specification Version 3.00 or later

SD Specifications Part E1 SDIO Card Specification Version 4.00 or later

PCI Bus Power Management Interface Specification Revision 1.1 December 1998

PCI Local Bus Specification Revision 2.3 March 2002

Appendix B (Normative) : Special Terms

B.1 Abbreviations and Terms

ACPI	Advanced Configuration and Power Interface: PCI bus supports ACPI.
ADMA	Advanced DMA: This term stands for ADMA1 and ADMA2.
ADMA1	ADMA Version 1: 4KByte boundary base ADMA
ADMA2	ADMA Version 2: Without 4KByte boundary limitation. (Recommended to support)
API	Application Program Interface
Auto CMD12	Host Controller function to issue CMD12 to stop multiple-block operation.
Auto CMD23	Host Controller function to issue CMD23 to stop multiple-block operation.
Block Gap	Period between blocks of data
Block	a number of bytes, basic data transfer unit
Busy	Busy signal: SD card drives busy on DAT[0] line.
CCCR	Card Common Control Register: One of registers defined in SDIO card.
CCS	Card Capacity Status: A field name in the response of ACMD41.
CDCLK	Card Detect Clock: a clock for detecting SD card
CID	Card Identification number register
CMD	SD bus command line
CMDXX	SD commands: XX indicates one or two digit decimal command number.
CMD_wo_DAT	Commands without using DAT line
CRC	Cyclic Redundancy Check
CSD	Card Specific Data register
DAT	SD bus 4-bit Data line: It is also expressed by DAT[3:0]
Descriptor Table	Sequence of ADMA Programs created on system memory.
DMA	Direct Memory Access: This term stands for SDMA, ADMA1 and ADMA2.
GPS	Global Positioning System
HCS	Host Capacity Support: A field name in the argument of ACMD41.
HW	Hardware
Int.	Interrupt: SD card drives interrupt on DAT[1] line.
LED	Light Emitting Diode
OCR	Operation Conditions Register
OS	Operating System
Page Size	Unit of system memory management. Most host system adopts 4KB page size.
PCI	Peripheral Component Interconnect
PHS	Personal Handyphone System
PME	Power Management Enable
Resume	Restore and restart a suspended function. It is defined in SDIO spec.
RCA	Relative Card Address Register: RCA is received from CMD3.
SDCD#	Card Detect Signal: a signal, which is active in a level of low, for detecting SD card.
SDCLK	SD bus clock line: Host supplies clock to card through this line.
SDMA	Single Operation DMA defined in the Host Controller Specification Ver1.00.
SDR50	One of UHS-I modes up to 50MB/sec bus speed.
SDR104	One of UHS-I modes up to 104MB/sec bus speed.
SDWP	a signal, which is active in a level of high, for detecting SD card to be protected writing
Suspend	Stop and save a function to be able to resume. It is defined in SDIO spec.
TMCLK	a clock for detecting a timeout on DAT line
UHS-I	Ultra High Speed Version 1

Appendix C : PCI Configuration Register

As regards PCI bus interface, the Host Driver requires some information in the PCI Configuration registers to identify the SD Host Controller. It is specified in Appendix A of this specification.

C.1 Register Maps

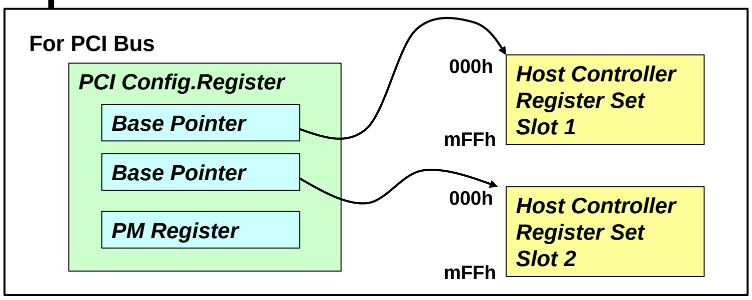


Figure C- 1 : Register Set for PCI Device (Example for 2 slots)

The PCI Configuration register is a special register to support Plug & Play and ACPI power management. The PCI Configuration registers for the PCI Based SD Host Controller is defined as appendix A.

Multiple slots can be supported through the use of multiple Base Addresses within a single PCI Function. Each of these Base Addresses is configured through the *Base Address* registers at offsets 10h to 24h in the PCI Configuration Space Header. The PCI Specification allows a PCI Function to have up to six Base Addresses. As such, a PCI Based SD Host Controller can support up to a total of six SD Slots.

A PCI Based SD Host Controller shall configure the *Base Address* register of each supported SD Slot such that it is a memory base address with at least 256 bytes allocated. This allows for enough memory address space in each Base Address to access all of the registers defined in this specification. Each set of the SD registers shall be implemented in a separate Base Address.

The values of Power Management register specified in the PCI Configuration registers should refer to the current and consumption values for all slots combined. It shall be read the total amount of power used by the PCI Based SD Host Controller, whether it has a single slot or multiple slots.

If Host Controller requires vendor specific register area, additional 256bytes area is added after the standard register area as shown Figure C- 2.

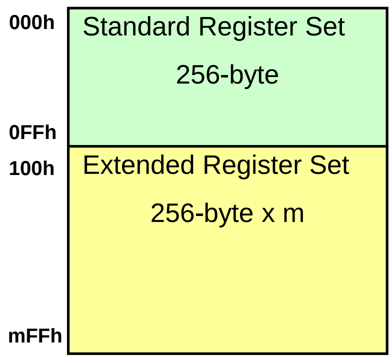


Figure C- 2 : Vendor Specific Register Area Extension

C.2 SD Controller Configuration Register MAP

<table>
<tr><th>31</th><th>23</th><th>15</th><th>07 00</th><th>Port</th></tr>
<tr><td colspan="2">Device ID</td><td colspan="2">Vendor ID</td><td>00h</td></tr>
<tr><td colspan="2">Status</td><td colspan="2">Command</td><td>04h</td></tr>
<tr><td colspan="3">Class Code</td><td>Revision ID</td><td>08h</td></tr>
<tr><td></td><td>Header Type</td><td></td><td></td><td>0Ch</td></tr>
<tr><td colspan="4">Base Address(es)</td><td>10-27h</td></tr>
<tr><td></td><td></td><td></td><td></td><td>28-2Bh</td></tr>
<tr><td colspan="2">Subsystem Device ID</td><td colspan="2">Subsystem Vendor ID</td><td>2Ch</td></tr>
<tr><td></td><td></td><td></td><td></td><td>30h</td></tr>
<tr><td></td><td></td><td></td><td>Capability Pointer</td><td>34h</td></tr>
<tr><td></td><td></td><td></td><td></td><td>38h</td></tr>
<tr><td></td><td></td><td>Interrupt Pin</td><td>Interrupt Line</td><td>3Ch</td></tr>
<tr><td></td><td></td><td></td><td>Slot Information</td><td>40h</td></tr>
<tr><td></td><td></td><td></td><td></td><td>44-7Fh</td></tr>
<tr><td colspan="2">Power Management Capabilities (PMC)</td><td>Next Item Ptr</td><td>Capability ID</td><td>80h</td></tr>
<tr><td>Data</td><td>PMCSR PCI to PCI Bridge Support (PMCSR_BSE)</td><td colspan="2">Power Management Control/Status (PMCSR)</td><td>84h</td></tr>
<tr><td></td><td></td><td></td><td></td><td>88-FFh</td></tr>
</table>

Table C- 1 : PCI Configuration Register for Standard SD Host Controller

PCI configuration space is divided into 3 areas.

00h - 3Fh : Registers defined in the PCI Bus Interface Specification
40h - 7Fh : Register area reserved for the SD Host Specification
Slot Information assigns to 40h and 41h-7Fh is reserved for future.
80h - FFh : Register area reserved for vendor unique registers

> Implementation Note:
> The Host Controller should place Power Management registers anywhere in the vendor unique register area. The offset address of Power Management register is set by *Capability Pointer* register. The Table C- 1 shows the case of offset being 80h (registers from 80h to 87h).

C.3 PCI Configuration Register

This section defines PCI Configuration registers that are specific for the PCI Based SD Host Controller. Refer to PCI Specification Version 2.3 for other standard PCI Configuration registers.

C.3.1 Class Code Register (Offset 09h)

D31 D24	D23 D16	D15 D08
Basic Class	Sub Class	Interface Code

Figure C- 3 : PCI Config. Class Code Register

Location	Attrib	
31-24 (0Bh)	RO	**Basic Class** 08h: General Peripheral
23-16 (0Ah)	RO	**Sub Class** 05h: for SD Host Controller
15-08 (09h)	RO	**Interface Code** 00h: Standard Host not supported DMA 01h: Standard Host supported DMA 02h: Vendor unique SD Host Controller

Table C- 2 : PCI Config. Class Code Register

C.3.2 Base Address Register (Offset 10h)

Maximum six base addresses can be supported, In case of multiple functions controller; these registers are used to point location not only for the SD Host Controller register sets but also for other functions. Refer to C.3.3 Slot Information Register to identify which base address is used for the SD Host Controller.

D31 D08	D07 D01	D00
Base Address	00 0000	Space Indicator

Figure C- 4 : PCI Config. Base Address Register for 256Byte Register Map

Location	Attrib	
31-08	RW	**Base Address** The SD Host Controller register set is mapped on a memory space of 256bytes starting from this base address.
07-01	RO	Fixed to 00 0000b.
00	RO	**Space Indicator** Set to 0 if mapped to the memory space.

Table C- 3 : PCI Config. Base Address Register for 256Byte Register Map

D31 D09	D08 D01	D00
Base Address	000 0000	Space Indicator

Figure C- 5 : PCI Config. Base Address Register for 512Byte Register Map

Location	Attrib	
31-09	RW	**Base Address** The SD Host Controller register set is mapped on a memory space of 512bytes starting from this base address. Refer to Figure C- 2.
08-01	RO	Fixed to 000 0000b.
00	RO	**Space Indicator** Set to 0 if mapped to the memory space.

Table C- 4 : PCI Config. Base Address Register for 512Byte Register Map

Implementation Note:
Multiple slot support Host Controller use Base Address registers at offsets 10h to 24h in the PCI Configuration register. Format of all Base Address registers are the same as this register. Not used Base Address registers shall be zero with RO type.

Offset 10h:	Slot1
Offset 14h:	Slot2
Offset 18h:	Slot3
Offset 1Ch:	Slot4
Offset 20h:	Slot5
Offset 24h:	Slot6

C.3.3 Slot Information Register (Offset 40h)

D07	D06 D04	D03	D02 D00
Reserved	Number of slots	Reserved	First Base Address Register Number

Figure C- 6 : PCI Config. Slot Information Register

Location	Attrib	
07	Rsvd	**Reserved**
06-04	RO	**Number Of Slots** These statuses indicate the number of slots the Host Controller supports. In the case of single function, maximum 6 slots can be assigned. 000b: 1 slot 001b: 2 slot 010b: 3 slot 011b: 4 slot 100b: 5 slot 101b: 6 slot
03	Rsvd	Reserved
02-00	RO	**First Base Address Register Number** Up to 6 Base Address can be specified in single configuration. These bits indicate first Base Address register number assigned for SD Host Controller register set. In the case of single function and multiple register sets, contiguous base addresses are used. **Number Of Slot** specifies number of base address. 000b: Base Address 10h (BAR0) 001b: Base Address 14h (BAR1) 010b: Base Address 18h (BAR2) 011b: Base Address 1Ch (BAR3) 100b: Base Address 20h (BAR4) 101b: Base Address 24h (BAR5)

Table C- 5 : PCI Config. Slot Information Register

C.4 The Relation between Device State, Power and Clock

C.4.1 Power Management in SD Mode

Table C- 6 shows Power Management policies in SD Mode when an SD card is inserted.

State	Card Power	SD Clock	Bus Mode	SD Bus Action
D0	On	On	4 or 1 bit 1)	Any SD transaction or Interrupt
D1	On	On	4 or 1 bit 1)	Long busy indication or Interrupt
D2	On	Off	4 or 1 bit 2)	Long busy indication or Interrupt
D3 hot	On or Off 3)	Off	4 or 1 bit 2)	Interrupt only
D3 cold	On or Off 3)	Off	4 or 1 bit 2)	Interrupt only

Table C- 6 : The Relation between Device State, Power and Clock

(1) Setting to 4-bit mode is recommended
(2) If **Asynchronous Interrupt Enable** is set to 1, 4-bit mode can be used
(3) If PME is supported in the D3 state, card power shall be supplied

The relations between card power supply and Device states are shown below:
In the D0 state, while an SD card is inserted or not rejected, card power shall be supplied by setting the **SD Bus Power** in the *Power Control* register. In the D1 or D2 states, the Host Driver shall keep the preceding power supply state. In the D3 state, if the Host System supports card interrupt wakeup, the card power shall keep on. In all states, when the SD card is removed after the card power is supplied, the Host Controller shall shut off the card power and clear the **SD Bus Power** in *Power Control* register automatically.
The relations between the SD Clock and Device states are shown below:
In the D0 state, while the SD card is inserted or not rejected, the SD Clock shall be supplied by setting the **SD Clock Enable** in the *Clock Control* register. In the D1 state, the Host Driver shall keep the state of the SD Clock while in the D0 state. In the D2 and D3 states, the Host Controller shall stop the SD Clock regardless of the **SD Clock Enable**. If a card supports wakeup and **Asynchronous Interrupt Enable** in the *Host Control 2* register is set to 0, the SD Bus mode shall be changed to 1-bit mode just before transferring from the D0 state. In all states, when the SD card is removed after the card power has been supplied, the Host Controller shall stop the SD Clock and clear the **SD Clock Enable** automatically.

In case a slot is for embedded device, following rule is applied.
In D2 and D3 state, when a device supports wakeup, **Interrupt Pin Select** in the *Embedded Control* register is set to 0 and **Asynchronous Interrupt Enable** in the *Host Control 2* register is set to 0, a driver needs to change to 1-bit mode. Otherwise, it is possible to keep 4-bit mode.
In D3 state, VDDH should be supplied regardless of supporting PME.
In D3 cold state, back-end power should be stopped by a driver.

C.4.2 Power Management in UHS-II Mode

Table C- 7 shows Power Management policies in UHS-II Mode.

State	*VDD1*	*VDD2*	*RCLK*	UHS-II Card State
D0	On	On	On	Fast Mode
D1	On	On	On	Low Power Mode
D2	On	On	Off	Dormant
D3 hot	Off	On	Off	Hibernate
D3 cold	Off	Off	Off	Power off

Table C- 7 : The Relation between Device State, Power and Clock

On changing the device state in PCI configuration register, Host Controller generates interrupt to make Host Driver to set one of the device states of Card and Host Controller.
VDD1 is controlled by **SD Bus Power for VDD1,** ***VDD2*** is controlled by **SD Bus Power for VDD2** in the *Power Control* register and ***RCLK*** is controlled by **SD Clock Enable** in the *Clock Control* register.
Card is in Fast Mode by setting **Power Mode** to 0 in the *UHS-II General Settings* register and in Low Power Mode by setting **Power Mode** to 1. Dormant or Hibernate is designated by GO_DORMANT_STATE command.

C.4.3 Internal Clock Control

In D0, D1 and D2 states, internal clock is active. If PME is not used in D3 state, internal clock may be stopped by clearing Internal Clock Enable=0 and PLL Enable=0 in the Clock Control register. Once internal clock is stopped, Internal Clock Setup Sequence (Section 3.2.1) is executed to recover from D3.

C.5 Generate PME Interrupt by the Wakeup Events

PME interrupt is generated by rising edge of three interrupt statuses that gated by *Wakeup Event Enable (*Refer to Section 1.8). Writing 1 to the **PME Status** clears its status.

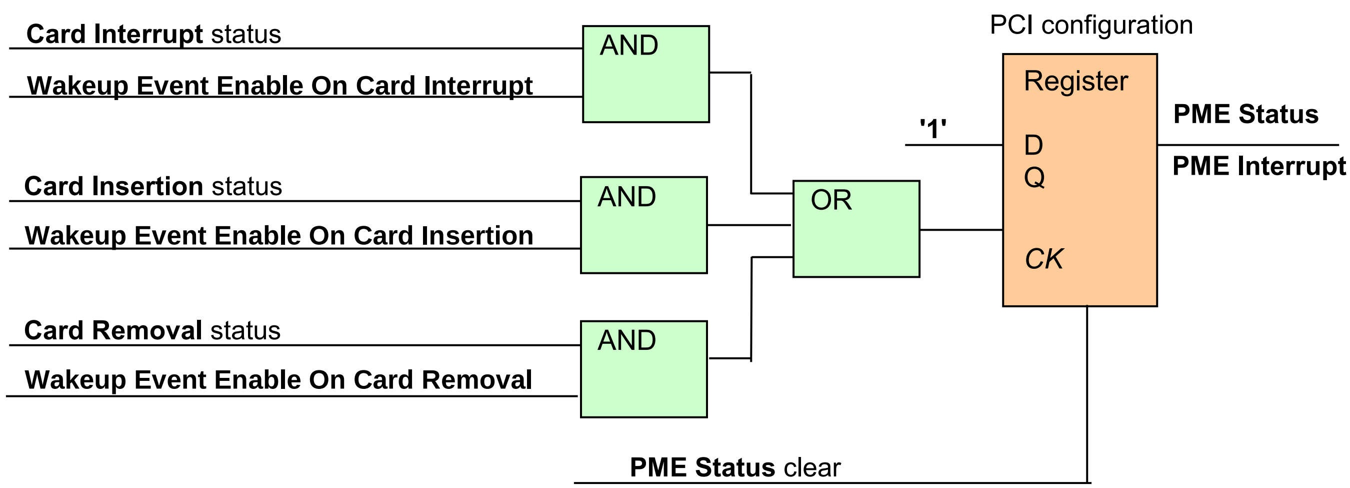


Figure C- 7 : Condition to Generate PME Interrupt

Appendix D : Shared Bus Supported Host Controller

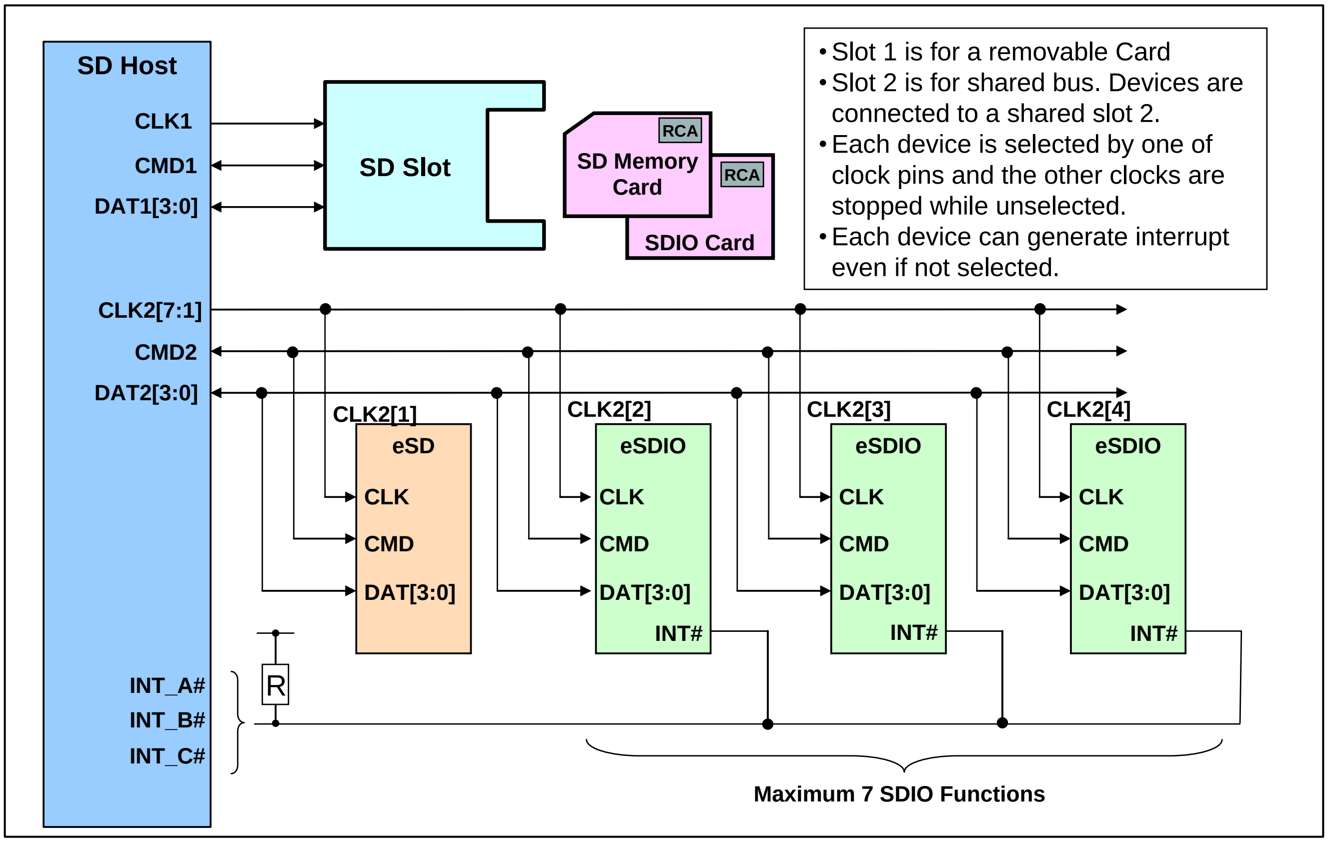


Figure D- 1 : Example Configuration Supporting Card Slot and Shared Bus

Figure D- 1 shows an example configuration of a system supporting a card slot and shared bus. A card slot bus and shared bus are separated so that removal card is not influence on the devices on shared bus. SD Bus signals except clock signal are connected together on the shared bus. Each device is selected by individual clock pins. An example timing of clock signals is shown in Figure 2-63. Stopping clock for unselected device enables system to reduce power consumption of devices. Even if SD bus clock is stopped, an SDIO device can generate interrupt by INT# pin to request a service to Host System. INT is asynchronous interrupt, low active, open drain and then can be wired-or with interrupt pin of another device. Pull-up resister is required for INT# signal.

Appendix E : UHS-II Implementation Notes

E.1 UHS-II Packet Header Check

Conditions of UHS-II packet header Check for Host is almost equivalent to that for Device as described in Table 5-14 of UHS-II Addendum. Host should bypass any of packets whose DID is not equal to 0 except packets setting unused IDs. In addition, as the Host Controller of this version is supposed to be only an initiator of UHS-II command packets, such implementation is allowed that Host does not bypass the packets. This Appendix shows an example of host implementation for Header Check and Unnecessary Packet Elimination.

E.1.1 An Example of Packet Header Check by Host

Host Controller initiates a transaction by issuing a command packet and checks header of received packets that is related to the command transaction. On issuing a command, Host Controller saves DID and TID of the command and compares header of received packets to host (DID=0) with DID and TID saved. IDs mismatch is indicated as Header Error.

TYP	Kind of Packet	Action
0: CCMD	Returned Broadcast CCMD (Normal)	If SID=0, received packet is valid.
	P2P CCMD to Host (Error)	If SID=non-zero, ignore and not output
1: DCMD	DCMD to Host (Error)	ignore and not output
2: RES	Response	Compared IDs (Note 1)
3: DATA	Read Data	Compared IDs (Note 1)
7: MSG	FCREQ, FCRDY, STAT, EBSY	Compared IDs (Note 1)
Others	Undefined	ignore and not output

Note 1: Compared SID/TID of the received packet with DID/TID of the command. If IDs are not matched, Header Error is indicated.

Table E - 1 : An Example of Packet Header Check by Host

E.1.2 An Example of Unnecessary Packet Elimination

The Table 5-14 of the UHS-II Addendum Version 1.00 describes the requirement of header check. Host has a responsibility to remove unused packets to prevent infinite packet loop in ring connection.
Host Controller Version 4.10 supports the case that host is only a command initiator in ring topology. Then unused received packet is defined as follows:

(1) In case of a received packet is sent to host (DID=0), ignored packet in Table E - 1 is unused packet.
(2) A received packet sent to other than host (DID=non-zero) is unused packet.

E.2 CCMD Read Transaction during CTS

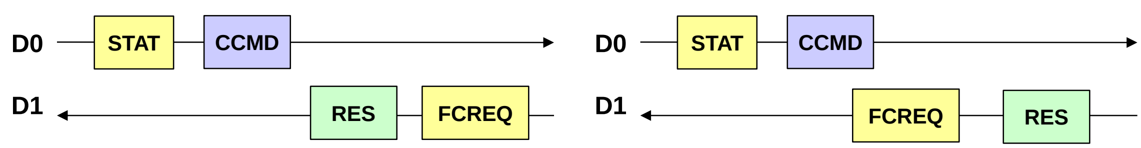
There are three notes for CCMD Read Transaction during CTS for Host Controller implementation. Figure E - 1 shows three cases.

When CCMD is issued after STAT, Host Controller needs to consider that there are two cases in order on receiving packets of RES and FCREQ as shown in case 1.

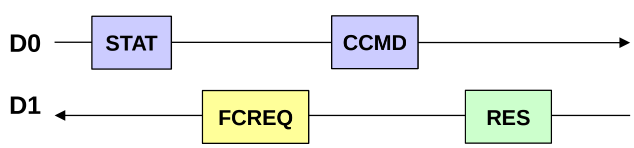
Host Controller may issue CCMD after receiving FCREQ as described in case 2 to simplify packet receiving control.

Host Controller will receive any mixture of symbols (LIDL, EIDL and DIDL) between CCMD and RES as shown in case 3.

1. If Host issues CCMD after STAT, there are two cases in ordering of RES and FCREQ



2. Host may issue CCMD after receiving FCREQ to ensure CCMD is followed by RES.



3. In case of CCMD is (TRANS_ABORT, CMD12 or FULL_RESET), any mixture of LIDL, EIDL and DIDL may be sent by Device between CCMD and RES.

Figure E - 1 : CCMD Read Transaction during CTS