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

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>31-00</td><td>RW</td><td><b>32-bit Block Count (SDMA System Address)</b>
When <b>Host Version 4 Enable</b> is set to 0 in the <i>Host Control 2</i> register, SDMA uses this register as system address in only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA.
When <b>Host Version 4 Enable</b> is set to 1, SDMA uses <i>ADMA System Address</i> 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) <b>SDMA System Address (Host Version 4 Enable</b> = 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 <b>SDMA Buffer Boundary</b> in the <i>Block Size</i> register. The Host Controller generates <b>DMA Interrupt</b> 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 <b>Continue Request</b> in the <i>Block Gap Control</i> register, the Host Controller shall start at the next contiguous address stored here in the <i>SDMA System Address</i> register.
ADMA does not use this register.

(2) <b>32-bit Block Count (Host Version 4 Enable</b> = 1)
Host Controller Version 4.10 re-defines this register as <i>32-bit Block Count</i> (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.
<table>
<tr><td>FFFF_FFFFh</td><td>4G - 1 block</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0000_0002h</td><td>2 blocks</td></tr>
<tr><td>0000_0001h</td><td>1 block</td></tr>
<tr><td>0000_0000h</td><td>Stop Count</td></tr>
</table>
</td></tr>
</table>

		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

<table>
<tr><th>Location</th><th>Attrib</th><th>Register Field Explanation</th></tr>
<tr><td>15</td><td>Rsvd</td><td><b>Reserved</b></td></tr>
<tr><td>14-12</td><td>RW</td><td><b>SDMA Buffer Boundary</b>
The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, <i>SDMA System Address</i> 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 <b>DMA Interrupt</b> to request the Host Driver to update the <i>SDMA System Address</i> register. At the end of transfer, the Host Controller may issue or may not issue <b>DMA Interrupt</b>. In particular, <b>DMA Interrupt</b> shall not be issued after <b>Transfer Complete Interrupt</b> 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 <b>SDMA Support</b> in the <i>Capabilities</i> register is set to 1 and this function is active when the <b>DMA Enable</b> in the <i>Transfer Mode</i> register is set to 1. ADMA does not use this register.
<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>
</table>

<table>
<tr>
<td>11-00</td>
<td>RW</td>
<td><b>Transfer Block Size</b>
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>
<b>16-bit Block Count</b>
Host Controller Version 4.10 extends block count to 32-bit (Refer to Section 1.15). Selection of either <i>16-bit Block Count</i> register or <i>32-bit Block Count</i> register is defined as follows:
(1) If <b>Host Version 4 Enable</b> in the <i>Host Control 2</i> register is set to 0 or <i>16-bit Block Count</i> register is set to non-zero, <i>16-bit Block Count</i> register is selected.
(2) If <b>Host Version 4 Enable</b> is set to 1 and <i>16-bit Block Count</i> register is set to zero, <i>32-bit Block Count</i> register is selected.

<i>Use of 16-bit/32-bit Block Count</i> register is enabled when <b>Block Count Enable</b> in the <i>Transfer Mode</i> 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

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	RW	**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) 1 Multiple Block 0 Single Block
04	RW	**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. 1 Read (Card to Host) 0 Write (Host to Card)

<table>
<tr>
<td>03-02</td>
<td>Rsvd</td>
<td>**Auto CMD Enable**
This field determines use of auto command functions.
<table>
<tr><td>00b</td><td>Auto Command Disabled</td></tr>
<tr><td>01b</td><td>Auto CMD12 Enable</td></tr>
<tr><td>10b</td><td>Auto CMD23 Enable</td></tr>
<tr><td>11b</td><td>Auto CMD Auto Select</td></tr>
</table>

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.