

SDIO Simplified Specification Version 2.00

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	CCCR/SDIO Revision	SDIO bit 3	SDIO bit 2	SDIO bit 1	SDIO bit 0	CCCR bit 3	CCCR bit 2	CCCR bit 1	CCCR bit 0
0x01	SD Specification Revision	RFU	RFU	RFU	RFU	SD bit 3	SD bit 2	SD bit 1	SD bit 0
0x02	I/O Enable	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	RFU
0x03	I/O Ready	IOR7	IOR6	IOR5	IOR4	IOR3	IOR2	IOR1	RFU
0x04	Int Enable	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IENM
0x05	Int Pending	INT7	INT6	INT5	INT4	INT3	INT2	INT1	RFU
0x06	I/O Abort	RFU	RFU	RFU	RFU	RES	AS2	AS1	AS0
0x07	Bus Interface Control	CD Disable	SCSI	ECSI	RFU	RFU	RFU	Bus Width 1	Bus Width 0
0x08	Card Capability	4BLS	LSC	E4MI	S4MI	SBS	SRW	SMB	SDC
0x09-0x0B	Common CIS Pointer	Pointer to card's <i>common</i> Card Information Structure (CIS)							
0x0C	Bus Suspend	RFU	RFU	RFU	RFU	RFU	RFU	BR	BS
0x0D	Function Select	DF	RFU	RFU	RFU	FS3	FS2	FS1	FS0
0x0E	Exec Flags	EX7	EX6	EX5	EX4	EX3	EX2	EX1	EXM
0x0F	Ready Flags	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RFM
0x10-0x11	FN0 Block Size	I/O block size for Function 0							
0x12	Power Control	Reserved for Future Use (RFU)						EMPC	SMPC
0x13	High-Speed	RFU	RFU	RFU	RFU	RFU	RFU	EHS	SHS
0x14-0xEF	RFU	Reserved for Future Use (RFU)							
0xF0-0xFF	Reserved for Vendors	Area Reserved for Vendor Unique Registers							

Table 6-1 Card Common Control Registers (CCCR)

Field	Type	Description												
CCCRx	R/O	CCCR Format Version number. These 4 bits contain the version of the CCCR and FBR format that this card supports. Any change to the CCCR and/or the FBR structure shall cause a new version code to be assigned. The codes for the CCCR/FBR formats are as follows:												
		<table><tr><th>Value</th><th>CCCR/FBR Format Version</th></tr><tr><td>0x00</td><td>CCCR/FBR Version 1.00</td></tr><tr><td>0x01</td><td>CCCR/FBR Version 1.10</td></tr><tr><td>0x02</td><td>CCCR/FBR Version 1.20</td></tr><tr><td>0x03-0x0F</td><td>Reserved for Future Use</td></tr></table>	Value	CCCR/FBR Format Version	0x00	CCCR/FBR Version 1.00	0x01	CCCR/FBR Version 1.10	0x02	CCCR/FBR Version 1.20	0x03-0x0F	Reserved for Future Use		
		Value	CCCR/FBR Format Version											
		0x00	CCCR/FBR Version 1.00											
		0x01	CCCR/FBR Version 1.10											
		0x02	CCCR/FBR Version 1.20											
0x03-0x0F	Reserved for Future Use													
SDIOx	R/O	SDIO Specification Revision number. These 4 bits contain the version of the SDIO Specification that this card supports. The codes for the SDIO Specifications are as follows:												
		<table><tr><th>Value</th><th>SDIO Specification</th></tr><tr><td>0x00</td><td>SDIO Specification Version 1.00</td></tr><tr><td>0x01</td><td>SDIO Specification Version 1.10</td></tr><tr><td>0x02</td><td>SDIO Specification Version 1.20 (unreleased)</td></tr><tr><td>0x03</td><td>SDIO Specification Version 2.00</td></tr><tr><td>0x04-0x0F</td><td>Reserved for Future Use</td></tr></table>	Value	SDIO Specification	0x00	SDIO Specification Version 1.00	0x01	SDIO Specification Version 1.10	0x02	SDIO Specification Version 1.20 (unreleased)	0x03	SDIO Specification Version 2.00	0x04-0x0F	Reserved for Future Use
		Value	SDIO Specification											
		0x00	SDIO Specification Version 1.00											
		0x01	SDIO Specification Version 1.10											
		0x02	SDIO Specification Version 1.20 (unreleased)											
		0x03	SDIO Specification Version 2.00											
0x04-0x0F	Reserved for Future Use													

SDIO Simplified Specification Version 2.00

Field	Type	Description										
SDx	R/O	SD Format Version number. These 4 bits contain the version of the SD Physical Specification that this card supports. The codes for the SD Physical Specification are as follows: <table><tr><th>Value</th><th>SD Physical Specification</th></tr><tr><td>0x00</td><td>SD Physical Specification Version 1.01 (March 2000)</td></tr><tr><td>0x01</td><td>SD Physical Specification Version 1.10 (October 2004)</td></tr><tr><td>0x02</td><td>SD Physical Specification Version 2.00 (May 2006)</td></tr><tr><td>0x03-0x0F</td><td>Reserved for Future Use</td></tr></table>	Value	SD Physical Specification	0x00	SD Physical Specification Version 1.01 (March 2000)	0x01	SD Physical Specification Version 1.10 (October 2004)	0x02	SD Physical Specification Version 2.00 (May 2006)	0x03-0x0F	Reserved for Future Use
Value	SD Physical Specification											
0x00	SD Physical Specification Version 1.01 (March 2000)											
0x01	SD Physical Specification Version 1.10 (October 2004)											
0x02	SD Physical Specification Version 2.00 (May 2006)											
0x03-0x0F	Reserved for Future Use											
IOEx	R/W	Enable Function - If this bit is reset to 0, the function is disabled. If this bit is set to 1, the function is enabled to start its initialization. The completion of initialization is indicated in IORx. On power up or after a reset, the card shall reset this bit to 0. The host can also use IOEx as a per function reset for error recovery. The host sequence for a per function reset is to reset IOEx to 0, wait until IORx becomes 0 and then set IOEx to 1 again. If the error is not recovered by this sequence, SDIO reset should be used noting that the operation of all functions will be aborted. See section 11 for relation to Master Power Control and Power Select.										
IORx	R/O	I/O Function Ready- If this bit is reset to 0, the function is not ready to operate. This may be caused by the function being disabled or not ready due to internal causes such as a built-in self-test in progress. If this bit is set to 1, the function is ready to operate. The functions shall set this bit to 1 within the timeout value defined in the TPLFE_ENABLE_TIMEOUT_VAL tuple. On power up or after a reset, this bit shall be set to 0. For any function that is not implemented on an SDIO card, this bit shall always be 0.										
IENx	R/W	Interrupt Enable for function x. If this bit is cleared to 0, any interrupt from this function shall not be sent to the host. If this bit is set to 1, then this function's interrupt shall be sent to the host if the master Interrupt Enable (bit 0) is also set to 1.										
IENM	R/W	Interrupt Enable Master. If this bit is cleared to 0, no interrupts from this card shall be sent to the host. If this bit is set to 1, then any function's interrupt shall be sent to the host.										
INTx	R/O	Interrupt Pending for function x. If this bit is cleared to 0, this indicates that no interrupts are pending from this function. If this bit is set to 1, then this function has interrupt pending. Note that if the IENx or IENM bits are not set, the host cannot receive this pending interrupt.										
ASx	W/O	Abort Select In order to abort an I/O read or write and free the SD bus, the function that is currently transferring data must be addressed. These 3 bits define which function's transfer to stop. For example, the abort the transfer to function number 3, the value of 0x03 would be written to these bits using CMD52 only. If the abort is addressed to a suspended function, it does not affect current data transaction. Note that this is an abort, not a reset. The addressed function shall return to the CMD state and data transfer pending to that function shall be halted. This abort procedure does not work for SPI write operations. To abort an SPI write data transfer use the STOP_TRAN token as defined in section 7.3.3 of the SD Physical Specification. This form of abort applies only to the functions of an SDIO card. For the memory of a combo card, the abort methods defined in the SD Physical Specification shall be used to abort transfers to/from memory										
RES	W/O	I/O CARD RESET: Setting the RES to 1 shall cause all I/O functions in an SDIO or Combo card to perform a soft reset. Setting the RES to 1 does not affect the current card protocol selection(SD vs. SPI mode) and CD Disable. Setting of the RES bit shall only be performed using CMD52. When RES=1, the values of AS2-0 are don't-cares. The RES bit is auto cleared, so there is no need to rewrite a value of 0. This bit is write-only, any read returns an undetermined value. Memory in a combo card is not affected.										

SDIO Simplified Specification Version 2.00

Field	Type	Description
Bus Width 1:0	R/W	Defines the data bus width ('00'=1-bit or '10'=4-bit bus) to be used for data transfer. All Full-Speed SDIO cards support both 1 and 4-bit bus. A Low-Speed SDIO card's support of 4-bit bus is optional. On reset or power-on these bits are cleared to 00.
CD Disable	R/W	Connect[0]/Disconnect[1] the 10K-90K ohm pull-up resistor on CD/DAT[3] (pin 1) of the card. The pull-up may be used for card detection. This bit is cleared to 0 on power-on (connected). Its state is not affected by a reset command.
SCSI	R/O	Support Continuous SPI interrupt. This read-only bit is set to indicate that this SDIO card supports the assertion of interrupts in the SPI mode at any time, irrespective of the status of the card select (CS) line. If this bit is zero, then this SDIO card can only assert the interrupt line in the SPI mode when the CS line is asserted. This bit signals the capability of all functions in the SDIO card.
ECSI	R/W	Enable Continuous SPI Interrupt. If the SCSI bit is set, then this R/W bit is used to allow the SDIO card to assert the interrupt line in the SPI mode at any time, irrespective of the state of the CS line. This bit is cleared to zero on reset or power-up. If the SCSI bit is clear, this bit shall be read only and set to zero. This bit controls the assertion of interrupts in the SPI mode for all functions in the SDIO card.
SDC	R/O	Card Supports Direct Commands during data transfer. This bit applies only to the SD modes, it does not apply to SPI mode. This flag bit reports the SDIO card's ability to execute CMD52 while data transfer is in progress. If this bit is set, all I/O functions shall accept and execute the CMD52 while data transfer is underway on the DAT[x] lines. Also, any memory in a combo card shall allow the CMD52 to execute while it is transferring data. Since the CMD52 does not use the DAT[x] lines, it is possible to execute while data transfer to a different address on the card is underway. CMD52 is described in 5.1. In any case, SD or SPI mode, if an error occurs during data transfer the SDIO card shall accept CMD52 to allow I/O abort and reset regardless of this bit value. If the card supports suspend/resume then it shall also support this bit.
SMB	R/O	Card Supports Multi-Block. This flag bit reports the SDIO card's ability to execute the IO_RW_EXTENDED command (CMD53) in the block mode. If this bit is set, all I/O functions (0-7) shall accept and execute CMD53 with the optional block mode bit set. The IO_RW_EXTENDED command is described in 5.3
SRW	R/O	Card Supports Read Wait. This bit applies only to the SD modes, it does not apply to SPI mode. This flag bit reports the SDIO card's ability to support the Read Wait Control (RWC) operation. If set, all functions on the card are able to accept the wait signal on DAT[2]. RWC operation is described in section 6.5. Any card that supports Suspend/Resume shall also support Read Wait
SBS	R/O	Card supports Suspend/Resume. This bit applies only to the SD modes, it does not apply to SPI mode. This flag bit reports the SDIO card's ability to Suspend and Resume operations at the request of the host. If this bit is set, all functions except 0 shall accept a request to suspend operations and resume under host control. Suspend/Resume operation is described in 6.4. If this bit is 0, registers (0x0C-0x0F) shall not be supported.
S4MI	R/O	Supports interrupt between blocks of data in 4-bit SD mode. This flag bit reports the SDIO card's ability to generate interrupts during a 4-bit multi-block data transfer. If this bit is 0, then the SDIO card is not able to signal an interrupt during a multi-block data transfer in 4-bit mode. In this case, the interrupt is not signaled until after the data transfer is complete. If this bit is 1, then the SDIO card is able to signal an interrupt between blocks while data transfer is in progress. This operation is described in 8.1.4 Note, even if a card does not support the interrupt during 4-bit block transfer (S4MI=0), the card may signal interrupts during all other Interrupt Periods while the interrupt is enabled (IENx=1).

SDIO Simplified Specification Version 2.00

Field	Type	Description
E4MI	R/W	Enable interrupt between blocks of data in 4-bit SD mode. Enable the multi-block IRQ during 4-bit transfer for the SDIO card. When this bit is 0, the card shall not signal interrupts during a 4-bit multi-block data transfer. If this bit is 1, the card shall generate interrupts during 4 bit multi-block data transfers as described in 8.1.4 If this SDIO card does not support 4 bit multi-block IRQs (S4MI=0), then this bit shall be R/O and always read as 0. This bit shall be cleared to 0 by any reset
LSC	R/O	Card is a Low-Speed card. If this bit is set, it indicates that the SDIO card is a Low-Speed card (see 2.1). If this bit is clear, the SDIO card is a Full-Speed card.
4BLS	R/O	4-bit support for Low-Speed cards. If the SDIO card is a Low-Speed card (LSC=1) and it supports 4-bit data transfer, then this bit shall be set. If the card is not Low-Speed or if the card does not support 4-bit transfer, then this bit shall be zero.
Pointer to card's common CIS	R/O	This 3-byte pointer points to the start of the card's common CIS. The common CIS contains information relation to the entire card. The card common CIS shall be located within the CIS space of function 0 (0x001000- 0x017FFF) as described in section 6.11. A card common CIS is mandatory for all SDIO cards. This pointer is stored in little-endian format (LSB first).
BS	R/O	Bus status: If this bit is set to 1, then the currently addressed function (selected by FSx or by the function number in an I/O command) is currently executing a command which transfers data on the DAT[x] line(s). If this bit is 0, then the addressed function is not using the data bus. This bit is used by the host to determine which function of a multi-function or combo card is currently performing data transfer. Note that this bit is a part of the optional Suspend/Resume protocol. If the card does not support Suspend/Resume, this bit shall be read as 0. Any access to the CIA may not be suspended, so in this case, BS shall always be set to 1, irrespective of the host setting BR to 1.
BR	R/W	Bus Release Request/Status: This bit is used to request that the addressed function (selected by FSx or by the function number in CMD53 or Memory commands using DAT line) release the Data lines and suspend operation. If the host sets this bit to 1, the addressed function shall temporarily halt data transfer on the DAT[x] lines and suspend the command that is in process. The BR bit shall remain set to 1 until the release is complete. If the card can never accept the suspend request while executing transactions, the card shall return response with BR cleared to 0 and BS set to 1. This indicates that the suspend request is cancelled by the card and thus the host should not issue a cancel suspend command. The followings are the cases where the card can cancel a suspend request: - Transaction addressed to function 0. - The card knows the transfer will terminate soon. - The card knows the transfer is timing critical (i.e. If suspended, the transfer cannot proceed). A Multi function card that indicates SBS=1, but contains a function that does not support suspend/resume. Once the function is in suspend, it shall signal the host by clearing the BS and BR bits. The host can monitor the status of the suspend request by reading the BR bit. If it is set, the suspend request is still in progress. A pending suspend request can be cancelled by the host by writing 0 to the BR bit. The Standard Host Specification defines following suspend sequence: If the suspend request is not accepted, the host retries with a cancel suspend request command. Even if the card received a cancel suspend command, it should accept suspend if possible. If the card does not accept suspend, the host considers the function to have never suspend. However, the host should monitor the BR, BS and EXx bits to confirm that the suspend request was cancelled rather than granted. If SBS=0, this bit shall be R/O and read as 0.

SDIO Simplified Specification Version 2.00

Field	Type	Description										
FSx	R/W	Select Function bits 3:0 These four bits are used to select a function number (0-7) or the memory of a combo card (8) for Suspend/Resume. There are 2 means to write the value of FSx. First, an I/O writes to the register in the CCCR and second, a new I/O command causes the FSx to be set to the function number in that command. The value of FSx shall remain until overwritten. If a function or memory is currently suspended, the writing of it's number to FSx shall re-start (resume) the data transfer operation When reading FSx, the value returned shall be the number of the currently addressed function. Note that when reading FSx, if the Bus Status is 0 (BS=0), the FSx value is undefined. The FSx bits are coded as follows: <table><tr><th>FSx</th><th>Current Transaction</th></tr><tr><td>0000</td><td>Transaction of function 0 (CIA)</td></tr><tr><td>0001-0111</td><td>Transaction to functions 1-7</td></tr><tr><td>1000</td><td>Transaction of memory in combo card</td></tr><tr><td>1001-1111</td><td>Not defined, reserved for future use</td></tr></table> If SBS=0 these bits shall be R/O.	FSx	Current Transaction	0000	Transaction of function 0 (CIA)	0001-0111	Transaction to functions 1-7	1000	Transaction of memory in combo card	1001-1111	Not defined, reserved for future use
FSx	Current Transaction											
0000	Transaction of function 0 (CIA)											
0001-0111	Transaction to functions 1-7											
1000	Transaction of memory in combo card											
1001-1111	Not defined, reserved for future use											
DF	R/O	Resume Data Flag: A data transaction is resumed by writing its number to FSx. Once the transaction is resumed, the DF indicates if more data will be transferred. If DF is cleared to 0, then no additional data will be transferred after the function or memory is resumed. If DF is set to 1, then there is more data to transfer that will begin after the function or memory is resumed. The DF flag can be used to control the interrupt cycle in 4-bit mode. If DF=1, there is more data to transfer after restoring the function. In this case, the interrupt cycle should be disabled. If DF=0, the function or memory was suspended at end of data transfer (during busy). In this case, no data transfer shall begin after resume so the host can detect a start interrupt cycle after restore. When resuming, if the suspended function cannot continue data transfer the card shall return DF=0 to abort the transfer.										
EXx	R/O	Execution Flag bits 7:0 These bits are used by the host to determine the current execution status of all functions (1-7) and memory (0). The bit is set to 1 for each function or memory that is currently executing a command. The EXx bits tell the host that a function or memory is currently executing a command so no additional command should be issued to that function/memory. These bits are only defined if SBS=1. This bit is set if the function is active (either currently executing or suspended). If SBS=0 these bits shall be read as zero.										
RFx	R/O	Ready Flag bits 7:0 These bits tell the host the read or write busy status for functions (1-7) and memory (0). If a function or memory is executing a write transaction, an RFx bit cleared to 0 indicates the function/memory is busy and not ready to accept more data. If the RFx bit is set to 1, then the function/memory can accept write data. If a function/memory is executing a read command, if the RFx bit is cleared to 0, it indicates that read data is NOT available. If the bit is set to 1, it indicates that read data is ready to be transferred. These bits are only defined if SBS=1. Setting a bit to 1 indicates the function is ready to accept the resume command. There are two conditions where the function will set the bit to 1. One is when the function (executing or suspended) is ready to continue data transfer. The other is when the suspended function cannot continue data transfer. If SBS=0 these bits shall be read as zero.										
FN0 Block Size	R/W	This 16-bit register sets the block size for I/O block operations for Function 0 only. If this card does not support I/O block operations (SMB=0), then this register becomes read-only and shall always read 0x0000. The maximum block size is 2048 (0x0800) and the minimum is 1. At power-up or reset, this register shall be initially loaded with a value of 0x0000. The host is responsible for setting the appropriate value for the block size supported by function 0. This pointer is stored in little-endian format (LSB first).										

SDIO Simplified Specification Version 2.00

Field	Type	Description
SMPC	R/O	Support Master Power Control These bits tell the host if the card supports Master Power Control. SMPC=0 : The total card current is less than 200mA, even if all functions are active (IOEx=1). EMPC,SPS and EPS shall be zero. SMPC=1 :The total card current may exceed 200mA. EMPC, SPS and EPS are available.
EMPC	R/W	Enable Master Power Control EMPC=0(default): The total card current shall be less than 200mA. The card automatically switches the mode of function(s) to lower current or does not allow some functions to become enabled, regardless of the value of EPS, so that total card current is 200mA or less. (The card manufacturer determines which functions operate and their modes to guarantee this limit.) EMPC=1: The total card current may exceed 200mA and SPS and EPS are available. The host uses SPS, EPS in FBR and IOEx to enable higher current function modes based on the host's ability to supply the necessary current.
SHS	R/O	Support High-Speed This flag bit reports the card's ability to operate in High-Speed mode SHS=0: The card does not support High-Speed mode SHS=1: The card supports High-Speed mode. The host enables High-Speed mode via the EHS bit. See section 12 for details on switching between default and High-Speed mode.
EHS	R/W	Enable High-Speed EHS=0 (default): The card operates in default timing mode with a clock rate up to 25MHz. EHS=1 : High-Speed Mode The card operates in High-Speed timing mode with a clock rate up to 50MHz. See section 12 for details on switching between default and High-Speed mode. When SHS is set to 0, writing to this bit is ignored and always indicates 0.
RFU	R/O	Any bit defined as Reserved for Future Use (RFU) shall be read-only and shall be read as 0.
Reserved for Vendors	R/W	These 16 registers are reserved for the maker of the I/O card to be used for any operations that are defined by and specific to any vendor unique operation. Information about the use of these optional registers needs to be obtained from the SDIO card maker. Reading and/or writing these registers without understanding the vendor's definitions may cause unexpected behavior or even damage to the card.

Table 6-2 CCCR bit Definitions

6.10 Function Basic Registers (FBR)

In addition to the CCCR, each supported I/O function has a 256-byte area used to allow the host to quickly determine the abilities and requirements of each function, enable power selection for each function and to enable software loading. The address of this area is from 0x00n00 to 0x00nFF where n is the function number (0x1 to 0x7). This per-function area is structured as follows:

Address	7	6	5	4	3	2	1	0
0x100	Function 1 CSA enable	Function 1 supports CSA	RFU	RFU	Function 1 Standard SDIO Function interface code			
0x101	Function 1 Extended standard SDIO Function interface code							
0x102	RFU	RFU	RFU	RFU	RFU	RFU	EPS	SPS
0x103-0x108	Reserved for Future Use (RFU)							
0x109-0x10B	Pointer to Function 1 Card Information Structure (CIS)							
0x10C-0x10E	Pointer to Function 1 Code Storage Area (CSA)							
0x10F	Data access window to Function 1 Code Storage Area (CSA)							
0x110-0x111	I/O block size for Function 1							
0x112-0x1FF	Reserved for Future Use							
0x200-0x7FF	Function 2 to 7 Function Basic Information Registers (FBR)							
0x800-0xFFFF	Reserved for Future Use							

Table 6-3 Function Basic Information Registers (FBR)

The Individual bits and fields in the FBA are defined below in Table 6-4.

Field	Type	Description
SDIO Standard Function interface code	R/O	The SDIO Standard Function code identifies those I/O functions, which implement the recommended standard interface as defined in a separate Application Specification. A complete and current list of assigned standard codes shall be maintained and published in any addendums to this specification. The codes assigned to those standard interfaces at the time this specification was published are: 0x0 No SDIO standard interface supported by this function 0x1 This function supports the SDIO Standard UART 0x2 This function supports the SDIO Type-A for Bluetooth standard interface 0x3 This function supports the SDIO Type-B for Bluetooth standard interface 0x4 This function supports the SDIO GPS standard interface 0x5 This function supports the SDIO Camera standard interface 0x6 This function supports the SDIO PHS standard interface 0x7 This function supports the SDIO WLAN interface 0x8 This function supports the Embedded SDIO-ATA standard interface (Embedded SDIO-ATA shall be implemented only on devices following the "Embedded SDIO Specification"). 0x9-0x0E Not assigned, reserved for future use 0xF This function supports an SDIO standard interface number greater than 0xE. In this case, the value in byte 0x101 identifies the standard SDIO interfaces type.
Function Supports CSA	R/O	If this function supports and contains a Code Storage Area (CSA), this bit shall be set to 1. If this function does not support a CSA, this bit shall be cleared to 0. CSA enable is controlled by bit 7 of register 0xn00.

SDIO Simplified Specification Version 2.00

Field	Type	Description
Function CSA Enable	R/W	This bit controls access to the Code Storage Area for this function. If this bit is cleared to 0, then any read or write access to the CSA shall be blocked. If this bit is set to 1, then access to the CSA is allowed. This bit is cleared to 0 upon reset. If this function does not support CSA (0xn00 bit 6=0), then this bit shall be R/O and always read as 0.
Extended SDIO Standard Function interface code	R/O	This is the extension of the SDIO Standard Function interface code. If the SDIO Standard Function interface code is greater than 0xE, then this byte shall contain the code and the standard code (0x100 bits 3-0) shall contain a value of 0xF. If the standard code is less than 0xF, then this byte shall be 0x00.
SPS (Support Power Selection)	R/O	This bit indicates if the function has Power Selection. SPS=0 :This function has no Power Selection. EPS shall be zero. SPS=1 :This function has 2 power modes which are selected by EPS.
EPS (Enable Power Selection)	R/W	EPS=0(default): The function operates in Higher Current Mode The maximum current for the function shall be given in TPLFE_HP_MAX_PWR_3.3V EPS=1: The function works in Lower Current Mode The maximum current for the function shall be given in TPLFE_LP_MAX_PWR_3.3V This bit shall be reset when IOEx=0.
Address pointer to Function CIS	R/O	These three bytes make up a 24-bit pointer (only the lower 17 bits are used) to the start of the Card Information Structure (CIS) that is associated with each function. The CIS is defined in section 6.11. A CIS is mandatory for each function on an SDIO card. This pointer is stored in little-endian format (LSB first). This register points to the End of Chain tuple if the function is not supported on the card.
Address pointer to Function CSA	R/W	These three bytes make up a 24-bit pointer to the desired byte in the CSA to read or write. After any read or write to the CSA access window register, this pointer shall be automatically incremented by 1. If this function does not support CSA (0xn00 bit 6=0), then these 24 bits shall be R/O and always read as 0x000000. This pointer is stored in little-endian format (LSB first).
Data access window to CSA	R/W	Any read or write to this address when the CSA is enabled (0xn00 bit 7=1), shall pass data to/from the byte addressed by the CSA address pointer. If this function does not support CSA (0xn00 bit 6=0), then these 8 bits shall be R/O and always read as 0x00.
Function 1-7 I/O Block Size	R/W	This 16-bit register sets the block size for I/O block operations for each function (1-7). If this card does not support I/O block operations (SMB=0), then this register becomes read-only and shall always read 0x0000. The maximum block size is 2048 (0x0800) and the minimum is 1 (0x0001). At power-up or reset, this register shall be initially loaded with a value of 0x0000. The host is responsible for setting the appropriate value for the block size supported by each function. This pointer is stored in little-endian format (LSB first).

Table 6-4 FBR bit and field definitions

6.11 Card Information Structure (CIS)

The Card Information Structure provides more complete information about the card and the individual functions. The CIS is the common area to read information about all I/O functions that exist in a card. The design is based on the PC Card16 design standardized by PCMCIA. All cards that support I/O shall have a common CIS and a CIS for each function. The CIS is accessed by reads to a fixed area as shown in Table 6-5. This one area serves the card as a Common CIS and also as the storage area for each function. The common area and each function have a pointer to the start of its CIS within this memory space.

Address	7	6	5	4	3	2	1	0
0x0001000 - 0x017FFF	Card Common Card Information Structure (CIS) area for card common and all functions							
0x018000- 0x01FFFF	Reserved for Future Use							

Table 6-5 Card Information Structure (CIS) and reserved area of CIA

The valid tuples (storage structures) from the PCMCIA specification and new tuples created for SDIO are defined in section 16.7.

6.12 Multiple Function SDIO Cards

Multiple Function SDIO Cards shall have a separate set of Configuration registers for each function on the card. Multiple Function SDIO Cards shall use a combination of a CIS common to all functions on the card and a separate function-specific CIS specific to each function on the card. The common CIS describes features that are common to all functions on the card. Each function-specific CIS describes features specific to a particular function on the SDIO Card. Functions are numbered sequentially beginning with 1. The CMD5 response indicates the total number of functions, which includes 'dummy' functions. The host shall iterate through the CIS entries based on the CMD5 response.

The ERROR status flag of an R5 response is type "E R X", (see section 5.2.1) and can indicate an error in the previous command. Since the host software needs a method to determine which function detected the error, a Multiple Function SDIO cards shall only return the R5 ERROR status flag in the subsequent command issued to the same function.

6.13 Setting Block Size with CMD53

The host sets the block size for a function's multiple block transfers by writing to the 16-bit Function I/O Block Size register in the FBR (see Table 6-4). The host shall not write this register using CMD53 with Block Mode set to 1. If the card detects an invalid block size before executing CMD53 with Block Mode set to 1, it shall indicate an OUT_OF_RANGE error in the current response and shall not perform data transfer. This will also stop the interrupt period (see section 8.1.3)

6.14 Bus State Diagram

Figure 6-2 shows the Bus State Diagram for an SDIO card. It shows the bus states and their relations to SDIO commands and Suspend/Resume.

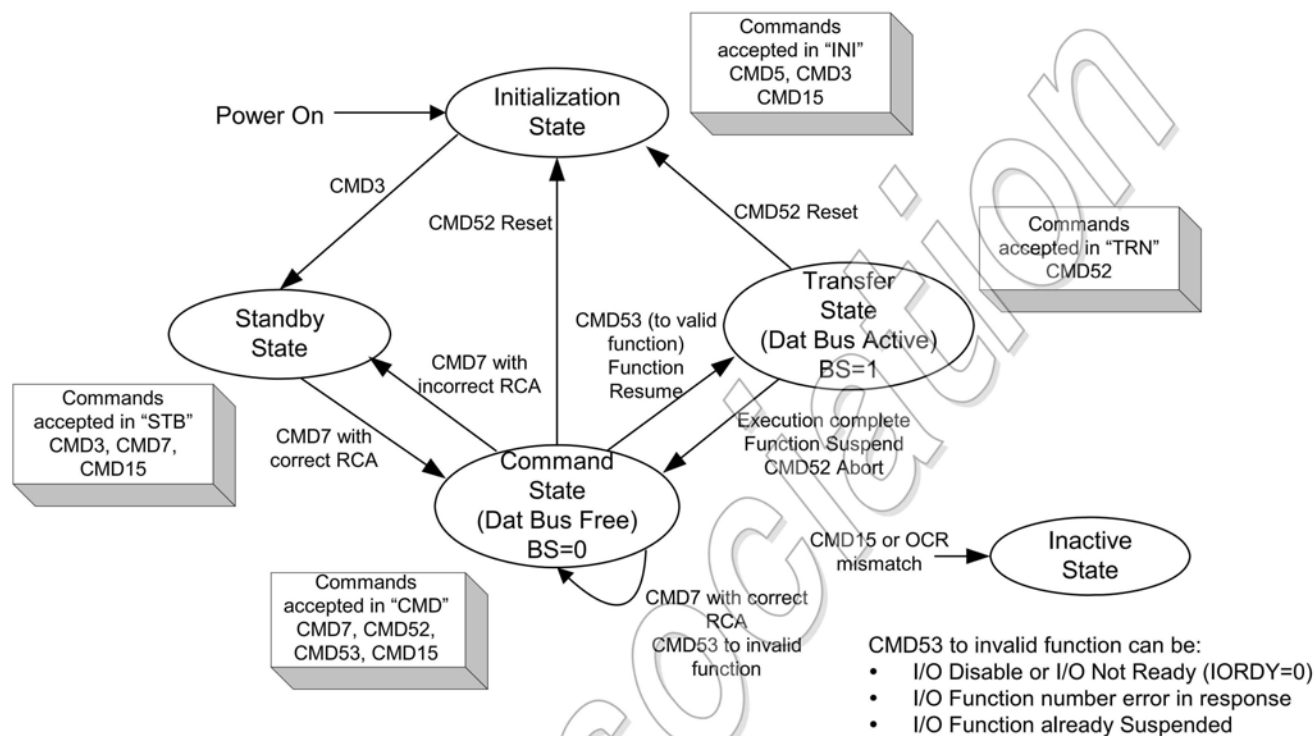


Figure 6-2 State Diagram for Bus State Machine

7. Embedded I/O Code Storage Area (CSA)

In order to support the concept of "Plug-and-Play" for SDIO cards, each function contained in a card may need to contain a block of memory for the storage of drivers and/or applications. In addition, since the same SDIO card may be used on multiple different host platforms, several different versions of the code may be needed for each function. One option is to store these programs in a standard SD Memory section of a combo card. Alternately, a standard access means to load the code is contained in the optional Code Storage Area (CSA). The CSA is a separate 16MB memory area that is accessed using the CSA address pointer and the CSA window register contained in the FBR registers. Note that each function may have its own CSA to support it. The CSA data can be read only or R/W. The actual storage method for the CSA is not a part of this specification and left to the implementers.

7.1 CSA Access

In order for the host to access a function's CSA, it first shall determine if that function supports a CSA. The host reads the FBR register at address 0x00n00 where n is the function number (0x1 to 0x7). If bit 6=1, then the function supports a CSA and the host enables access by writing bit 7=1. The next step is for the host to load the 24 bit address to start reading or writing. This is accomplished by writing the 24 bits (A23-0) to registers 0x00n0C to 0x00n0E where n is the function number (0x1 to 0x7). Once the start address is written, data can be read or written by accessing register 0x00n0F, the CSA data window register. If more than 1 byte needs to be read or written, an extended I/O command (byte or block) can be performed with an OP code of 0 (fixed address). The address pointer shall be automatically incremented with each access to the window register, so the access will be to sequential addresses within the CSA. Once the operation is complete, the address of the NEXT operation shall be held in the 24 bit address register for the host to read.

7.2 CSA Data Format

The data stored in the CSA shall be structured using the FAT12/FAT16 format. This format is defined in the ISO specification: *ISO/IEC9293:1994 Information technology - Volume and file structure of disk cartridges for information interchange*. This specification is also the basis for the SD memory cards. The information on the SD memory implementation can be found in the SDA publication: *Part 2 FILE SYSTEM SPECIFICATION Version 2.00 May 9, 2006*. The actual layout of files within the CSA is undefined by this specification.

The use of the CSA for program or data storage for different host types requires that the SDIO card manufacturer load the programs and data in a file format that may be recognized by the host. An example of this would be the use of a specific file name saved within a specific subdirectory that is recognized and executed by a particular host operating system. Such formats are specific and sometimes proprietary to different host implementations and operating systems.

The rest of this chapter is not included in the Simplified Specification.

8. SDIO Interrupts

In order to allow the SDIO card to interrupt the host, an interrupt function is added to a pin on the SD interface. Pin number 8, which is used as DAT[1] when operating in the 4-bit SD mode, is used to signal the card's interrupt to the host. The use of interrupt is optional for each card or function within a card. The SDIO interrupt is "level sensitive", that is, the interrupt line shall be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period (see 8.1.2). Once the host has serviced the interrupt, it is cleared via some function unique I/O operation. All hosts shall provide pull-up resistors on all data lines DAT[3:0] as described in section 6 of the SD Physical Specification.

8.1 Interrupt Timing

The operation of the interrupt pin is different between the SPI mode and the SD mode. The operation of the interrupt pin is defined as follows:

8.1.1 SPI and SD 1-bit Mode Interrupts

In the SPI and 1-bit SD mode, Pin 8 is dedicated to the interrupt function. Thus, in the SPI and SD 1-bit modes there are no timing constraints on interrupts. A card in the SPI or 1-bit SD mode signals an interrupt to the host at any time by asserting pin 8 low. The host detects this pending interrupt using a level sensitive input. The host is responsible for clearing the interrupt. If the SDIO card is operating in the SPI mode, the interrupt from the card may not be asserted if the card is not selected. (CS=0). The exception to this requirement occurs only if the card is both capable of interrupting when not selected (the SCSI bit in the CCCR = 1), and has that feature turned on (the ECSI bit = 1). In this case, the card may assert the interrupt irrespective of the state of the CS line. For more information, see Table 6-1.

8.1.2 SD 4-bit Mode

Since Pin 8 is shared between the IRQ and DAT[1] use in the 4-bit SD mode, an interrupt shall only be sent by the card and recognized by the host during a specific time. The time that a low on Pin 8 shall be recognized as an interrupt is defined as the **Interrupt Period**.

An SDIO host shall only sample the level on Pin 8 (DAT[1]/IRQ) into the interrupt detector during the Interrupt Period. At all other times, the host interrupt controller shall ignore the level on Pin 8. Note that the Interrupt Period is applicable for both memory and I/O operations

The definition of the Interrupt Period is different for operations with single block and multiple block data transfer.

8.1.3 Interrupt Period Definition

This section is not included in the Simplified Specification.

8.1.4 Interrupt Period at the Data Block Gap in 4-bit SD Mode (Optional)

This section is not included in the Simplified Specification.

8.1.5 Inhibited Interrupts (Removed Section)

This section is not included in the Simplified Specification.

8.1.6 End of Interrupt Cycles

This section is not included in the Simplified Specification.

8.1.7 Terminated Data Transfer Interrupt Cycle

This section is not included in the Simplified Specification.

8.1.8 Interrupt Clear Timing

Since the SDIO card uses level sensitive interrupts, the host shall clear pending interrupts with an I/O read or write to some function unique area. In some host implementations, the sending of a CMD52 to the card is handled by host adapter hardware while the host CPU can execute other operations. This condition may allow an interrupt that has already been handled to re-interrupt the host if the timing of the interrupt clear is not controlled. To prevent this condition, Any SDIO card that implements interrupts shall follow some required timing with respect to removing the interrupt from the DAT[1] line after the write to the function unique area that clears the interrupt. The clearing of the interrupt can be caused by an I/O write in a function unique method, or by a function unique I/O read. An example of clearing an interrupt using an I/O read would be a function where the reading of a data register may automatically clear the data ready interrupt.

The rest of this section is not included in the Simplified Specification.

9. SDIO Suspend/Resume Operation

The procedure used to perform the Suspend/Resume operation on the SD bus is:

1. The host determines which function is currently using the DAT[3:0] line(s).
2. The host requests the lower priority or slower transaction to suspend.
3. The host checks for the transaction suspension to complete.
4. The host begins the higher priority transaction.
5. The host waits for the completion of the higher priority transaction.
6. The host restores the suspended transaction.

If the current transaction can accept suspend and the card receives a suspend command during Read Wait, it shall accept the suspend request.

The rest of this chapter is not included in the Simplified Specification.

10. SDIO Read Wait Operation

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The read Wait operation allows a host to signal a card that is executing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SDIO card. To determine if a card supports the Read Wait protocol, the host shall test SRW capability bit in the Card Capability byte of the CCCR (see Table 6-1). The timing for Read Wait is based on the Interrupt Period that is defined in section 8.1

If a card does not support the Read Wait protocol, the only means a host has to stall (not abort) data in the middle of a read multiple command is to control the SDCLK. Read wait support is mandatory for the card to support suspend/resume.

The rest of this chapter is not included in the Simplified Specification.

11. Power Control

11.1 Power Control Overview

The concept of high-power SDIO cards was introduced in Version 1.10 of the SDIO Specification. Power Control supports following two features:

- **High-Power Support**
SDIO cards created prior to Version 1.10 of the SDIO Specification were limited to a maximum current of 200mA at any time, irrespective of the number or types of functions supported. With the creation of wireless communication devices in the SDIO form factor, a need was seen to provide more current to accommodate the higher power requirements of some SDIO cards. Since backward compatibility is a primary concern for any changes made to this specification, a method was chosen to prevent a high-power card from drawing excessive amounts of current from hosts designed to only support the SDIO 1.0 cards. Master Power Control allows standard and high-power cards to be inserted into any host without causing excessive current damage to the host. It is important to note that there exists the possibility of trying to use a card that requires high-power in a standard power host and having that card fail to operate. Master Power Control is supported on a per card basis and available to the host in the CCCR. A high-power card may have a mix of both high and standard power functions.
- **Power Selection Support**
Not all hosts can supply enough current for all SDIO cards. A host may choose to use the SDIO card in a lower power mode to increase operation time. Power Selection enables the host to switch the card to a lower power mode. It is important to note that there exists the possibility of trying to use a standard power card in a host that does not have enough power to meet the card's requirement. In this case the card will fail to operate. Cards supporting Power Selection will enable the widest range of host support. Power Selection is supported on a per function basis and available to the host in the FBR.

11.2 Power Control support for SDIO Cards

11.2.1 Master Power Control

SDIO version 1.10 cards indicate their support for the new power control functions with the SMPC (Support Master Power Control) bit in the CCCR (See section 6.9). Hosts enable the card's power control functions with the EMPC (Enable Master Power Control) bit.

SMPC can be set to 0 if the card maximum current is less than 200mA and Power Selection (see section 11.2.2) is not supported. A SDIO version 1.10 card which has SMPC set to 1 shall maintain backward compatibility when EMPC is set to 0. A SDIO version 1.0 host may not be aware of EMPC, which will remain 0 (its default state).

In the case where EMPC is set to 0, the card total current shall not exceed 200mA. Functions that exceed 200mA shall not set IORx to 1 and TPLFE_OP_MAX_PWR shall be set to 0. If a multi-function card's total power exceeds 200mA the card shall not set all IORx to 1, even if all IOEx are set to 1. Some of the functions' IORx can be set to 1 as long as the card's total current is less than 200mA. If the host tries to enable a function (IOEx =1) that will cause the card's total current to exceed 200mA, the card shall disable (IORx=0) one or more functions to keep the card's total current less than 200mA. Which functions are enabled depends on the design of the card vendor.

In the case where EMPC is set to 1, the card current can exceed 200mA, up to a maximum of 500mA. Card should be designed, where possible, to not require the maximum current, thus functioning in as many hosts as possible with sufficient power.

SDIO Simplified Specification Version 2.00**11.2.2 Power Selection**

Power Selection defines two power modes for a function: Lower Current Mode and Higher Current Mode. A card implementing Power Selection gives the host the choice between these two power modes. These modes can be used for functions, such as a radio, which can operate in full performance (Higher Current Mode) or reduced performance (Lower Current Mode). A card's support of Power Selection is indicated with the SPS bit (Support Power Selection) in the FBR. The host enables Power Selection with the EPS bit (Enable Power Selection) in the FBR.

11.2.3 High-Power Tuples

Six new tuples are defined in version 1.10 for each function tuple.

- Average power required when Master Power Control is not enabled (TPLFE_SP_AVG_PWR_3.3V)
- Peak power required when Master Power Control is not enabled (TPLFE_SP_MAX_PWR_3.3V)
- Average power required in Higher Current Mode (TPLFE_HP_AVG_PWR_3.3V)
- Peak power required in Higher Current Mode (TPLFE_HP_MAX_PWR_3.3V)
- Average power required in Lower Current Mode (TPLFE_LP_AVG_PWR_3.3V)
- Peak power required in Lower Current Mode (TPLFE_LP_MAX_PWR_3.3V)

These values are 16 bits long with a 1mA/step resolution. This allows a value of 0 to 65,535 mA to be used. Current varies depending on the voltage. These tuples are defined in 3.1-3.5V or 2.7-3.6V range. When a new voltage range is added in future specification, another six tuples will be added.

Table 11-1 shows which tuples a host shall refer to depending on the host version and the settings of EMPC and EPS bits.

Host	EMPC	EPS	Reference TPLs	Comments
Ver 1.0	Don't care	Don't care	TPLFE_OP_MIN_PWR TPLFE_OP_AVG_PWR TPLFE_OP_MAX_PWR	
Ver 1.1	0	Don't care	TPLFE_SP_AVG_PWR_3.3V TPLFE_SP_MAX_PWR_3.3V	Same as TPLFE_OP_AVG_PWR Same as TPLFE_OP_MAX_PWR
Ver 1.1	1	0	TPLFE_HP_AVG_PWR_3.3V TPLFE_HP_MAX_PWR_3.3V	Non zero value is required when SMPC=1
Ver 1.1	1	1	TPLFE_LP_AVG_PWR_3.3V TPLFE_LP_MAX_PWR_3.3V	Non zero value is required when SPS=1

Table 11-1 Reference Tuples by Master Power Control and Power Select

A version 1.10 SDIO card shall implement the six new tuples. A card which has SMPC set to 1, shall set a non-zero value to TPLFE_HP_AVG_PWR_3.3V and TPLFE_HP_MAX_PWR_3.3V. A card which has SPS set to 1, shall set a non-zero value to TPLFE_LP_AVG_PWR_3.3V and TPLFE_LP_MAX_PWR_3.3V. TPLFE_SP_AVG_PWR_3.3V and TPLFE_SP_MAX_PWR_3.3V should be set to the same value as TPLFE_OP_AVG_PWR and TPLFE_OP_MAX_PWR respectively.

11.3 Power Control Support for the SDIO Host**11.3.1 Version 1.10 Host**

The following are requirements for a version 1.10 host:

- The host shall recognize new Power Control registers and tuples as defined in SDIO Specification Version 1.10.
 - Power Control bits (SMPC, EMPC) in the CCCR
 - Power Selection bits (SPS, EPS) in the FBR
 - High-Power Tuples
- The host shall know its own power supply ability

- The host shall have the ability to manage power by calculating maximum current shown in the tuples and control EMPC, EPS and IOEx not to exceed total current that the host can supply. If the host does not have enough power to use the card, the host shall not enable the card.

11.3.2 Power Control Operation

A host reads the SMPC to see if the card supports Power Control additions. If so, the host checks SPS to see the card's support of Power Selection. If SMPC is set to 1, a version 1.10 host should set EMPC to 1.

Power Selection provides the host with the choice of two power modes for the function. A host that can supply enough power to the card does not need to use Power Selection.

If SMPC is set to 1 and SPS is set to 1, the host can utilize Power Selection by setting EMPC to 1. If EPS is set to 0 (default mode of the card), Higher Current Mode is selected. If EPS is set to 1, Lower Current Mode is selected. For example, a wireless function may offer a power mode with reduced transmission range and corresponding reduced current requirements. As the host knows how much current it can supply to the card, it reads the appropriate tuples and decides whether or not to enable each function (with IOEX), and if so in what Current Mode. The host shall verify that the total current required by the card shall not exceed the current of host can supply. A version 1.10 host should read new tuples as shown in Table 11-1.

12. High-Speed Mode

High-Speed mode increases the bus clock rate to 50MHz and the SD bus throughput from 12.5MB/sec to 25MB/sec. For information on High-Speed mode for SD memory cards see Part 1 *Physical Layer Specification Version 2.00*, sections 4.3.10, 4.3.11 and 6.8. SDIO and combo cards may also support High-Speed mode.

12.1 SDIO High-Speed Mode

SDIO version 1.20 cards indicate their support for High-Speed mode with the SHS (Support High-Speed) bit in the CCCR (See section 6.9). Hosts switch between default and High-Speed mode with the EHS (Enable High-Speed) bit in the CCCR by CMD52. Following a command to set or clear EHS, cards shall switch speed mode within 8 clocks after the end bit of the corresponding response.

When switching from default to High-Speed mode the host can try to set EHS without first checking SHS. The host issues CMD52 in RAW mode, setting EHS to one, and after getting the response of CMD52 the host checks SHS and EHS. If SHS=0 or EHS=0, the command will be ignored and the card is still in default mode. If SHS=1 and EHS=1, the card is in High-Speed Mode.

12.2 Switching Bus Speed Mode in a Combo Card

A combo card that supports High-Speed shall support it for both memory and IO. Two bus speed switch commands are defined; SD memory command (CMD6) and SDIO command (EHS in CCCR is changed using CMD52).

A part of this section is not included in the Simplified Specification.

When one bus speed switch commands is executed successfully, the card switches the card bus speed mode. If two bus speed switch commands are executed in turn (to the same bus speed mode), only the first successful command is effective to switch bus speed mode.

The host needs to check success of executing bus speed switch command and then the host can switch the host bus speed mode to the same one.

Success of switching bus speed mode is determined by checking receipt of a good response and the result of switching bus speed mode is the same as the switch requested.

The status of current bus speed mode is read by bus speed switch commands. For example, when bus speed mode is switched by CMD6, the result can be read from EHS. If switching by RAW mode of CMD52 has failed, there are two kinds of responses. One is no response with illegal command error. The other is that CMD52 is accepted and the status of RAW mode indicates EHS is not changed.

A reset of either the memory or IO portion of a combo card will also reset both portions to default speed mode. Within 8 clocks after response of a reset by CMD52 (write to RES in CCCR) or CMD0 the card shall change the speed mode to default speed mode.

Note that when changing the bus speed the host bus driver should treat a bus speed change request from any driver as an atomic operation. The host should mask interrupts and not issue any command to the card until the bus speed change is complete.

If a combo card supports the Lock/unlock function, a locked card cannot change bus speed mode. A locked card indicates an illegal command error to a bus speed switch command. The host needs to unlock the card by CMD42 before changing bus speed. It also implies that a host should not change bus speed mode during initialization before managing a locked card.

13. SDIO Physical Properties

This chapter is not included in the Simplified Specification.

13.1 SDIO Form Factors

The SDIO definition encompasses different form factors:

- Full-Size SDIO — compatible with host sockets designed for SD memory cards
- miniSDIO — compatible with host sockets designed for miniSD memory cards

13.2 Full-Size SDIO

The SDIO card is compatible with host sockets designed for SD memory cards. In addition, the SDIO cards can be extended to allow for external connectors, antennas etc. With the exception of the write protect switch, all SDIO cards shall meet the mechanical specifications described in the SD Physical Specification for that portion of the card that is not extended. The WP switch is not supported by hosts for SDIO only cards.

The rest of this section is not included in the Simplified Specification.

13.3 miniSDIO

The miniSDIO card is compatible with host sockets designed for miniSD memory cards. In addition, the miniSDIO cards can be extended to allow for external connectors, antennas etc. All miniSDIO cards shall meet the mechanical specifications described in *miniSD Card Addendum Version 2.00*, Part 1 Physical Layer Specification Version 2.00 for that portion of the card that is not extended.

When a miniSDIO card is inserted into a miniSD to SD memory card adaptor, and plugged-into an SDIO host slot, it shall appear to the host as an SDIO card.

The rest of this section is not included in the Simplified Specification.

14. SDIO Power

14.1 SDIO Card Initialization Voltages

SDIO Version 2.00 eliminate the voltage range 2.0-2.7V for basic communication because the SD Physical Specification Version 2.00 eliminates it. SDIO cards follow the same voltage and current requirements as SD memory cards. This means that an SDIO or combo card shall be used at an operating voltage range of 2.7 to 3.6V. The hosts shall not supply the voltage range 2.0-2.7V for basic communication to the SDIO Version 2.00 cards.

14.2 SDIO Power Consumption

The SDIO cards are intended to operate in mobile devices that have limited power sources available. Because the host's battery life may be significantly reduced if the SDIO card draws excessive power, a primary goal of SDIO designers should be low power. By reducing power consumption to a minimum, Host battery life and consumer satisfaction will be enhanced. The following power data represents the maximum that a SDIO card may draw. It is important for designers to note that a low power host may reject any SDIO card that identifies

SDIO Simplified Specification Version 2.00

itself as drawing more power than the host is willing to supply, thus lower power cards may have a competitive advantage in the market.

The rest of this chapter is not included in the Simplified Specification.

SD Association

15. Inrush Current Limiting

This chapter is not included in the Simplified Specification.

SD Association

16. CIS Formats

16.1 CIS Reference Document

The CIS used by SDIO is based directly upon the metaformat specification used by PCMCIA and Compact Flash. The user of this specification is directed to:

PC Card Standard
Volume 4
Metaformat Specification

Published by:

PCMCIA (Personal Computer Memory Card International Association)
2635 North First Street, Suite 209
 San Jose, CA 95134 USA
+1-408-433-2273
+1-408-433-9558 (Fax)

16.2 Basic Tuple Format and Tuple Chain Structure

The Card Information Structure is one or more chains (or linked lists) of data blocks or tuples. The basic format of tuples is shown in Table 16-1.

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE Tuple code: CISTPL_XXX							
0x01	TPL_LINK Offset to next tuple in chain. This is the number of bytes in the tuple body. (n)							
0x02.. (n+2)	The tuple body. (n bytes)							

Table 16-1 Basic Tuple Format

Byte 0 of each tuple contains a tuple code. A tuple code of 0xFF is a special mark indicating that there are no more tuples in the chain. There are 2 tuples with only a tuple code, the CISTPL_NULL and the CISTPL_END (see Table 16-2). These tuples do not have any additional bytes. For all other tuples, byte 1 of each tuple contains a link to the next tuple in the chain. If the link field is 0, then the tuple body is empty. If the link field contains 0xFF, then this tuple is the last tuple in its chain. There are two ways of marking the end of a tuple chain for SDIO cards: a tuple code of 0xFF, or a tuple link of 0xFF. The use of an FFH link value is allowed in SDIO cards, but it is recommended to use the End of Chain tuple. System software shall use the link field to validate tuples. No SDIO card tuple can be longer than 257 bytes: 1 byte *TPL_CODE* + 1 byte *TPL_LINK* + FFH byte tuple body (and this 257 byte tuple ends the chain). Some tuples provide a termination or stop byte that marks the end of the tuple. In this case, the tuple can effectively be shorter than the value implied by its link field. However, software shall not scan beyond the implied length of the tuple, even if a termination byte has not been seen.

16.3 Byte Order Within Tuples

Within tuples, all multi-byte numeric data shall be recorded in little-endian order. That is, the least-significant byte of a data item shall be stored in the first byte of a given field. Within tuples, all character data shall be stored in the natural order. That is, the first character of the field shall be stored in the first byte of the field. Fixed-length character fields shall be padded with null characters, if necessary.

16.4 Tuple Version

With the introduction of SDIO Specification Version 1.10, a different format for the tuple information is possible based on the changes made by each specification revision. These changes could be added fields in a tuple or entirely new tuples. In order to maintain backward compatibility, new data fields are added *after* existing fields in order to maximize backward compatibility. It is the responsibility of the host program that is scanning the tuple chain (sometimes called “walking the tuples”) to first determine the SDIO specification version that the card was designed to. The program shall be designed to anticipate that SDIO cards will be encountered that are built to a later version of the specification. This requires the program to ignore the additional data existing in SDIO cards built to a later specification. Specifically, the program shall first read the SDIO specification version that the card was designed to meet from the SDIOx field in the CCCR area. If the version is less than or equal to the version of the tuple scan program, then the program shall know and properly decode all tuple information. If the card version is greater than the scan program’s version, the scan program may need to ignore additional information fields or tuples. The unknown tuples can be ignored by simply skipping those tuples with unrecognized codes. Skipping is accomplished by using the TPL_LINK field (always the second byte) to jump over the unknown tuple. In a similar manner, the additional data fields in tuples should be ignored using the link field. For example, if a scan program is expecting 0x15 bytes of data and the TPL_LINK field indicates a size of 0x19 bytes, the scan program should ignore and skip over the last 4 bytes of data.

16.5 SDIO Card Metaformat

Unlike the PCMCIA card, the SDIO card has multiple CIS areas. There is a common CIS for the entire card and a CIS assigned to each function. Because of the multiple CIS areas, the SDIO card does not need to support the *CISTPL_LONGLINK_MFC* tuple or the *CISTPL_LINKTARGET* as described in section 2.3.6 of the PCMCIA spec. Table 16-2 lists the tuple codes supported by SDIO cards. The type field indicates if a tuple is Optional (O), Mandatory (M), Recommended (R), or not applicable (n/a) for the common (function 0) tuple and for each function (1-7) supported by the card. For more details on each tuple, see the PCMCIA metaformat specification section referenced.

Code	Name	Description	PCMCIA Reference	Type Common	Type Function
0x00	CISTPL_NULL	Null tuple	3.1.9	O	O
0x10	CISTPL_CHECKSUM	Checksum control	3.1.1	R	R
0x15	CISTPL_VERS_1	Level 1 version/product-information	3.2.10	O	O
0x16	CISTPL_ALTSTR	The Alternate Language String Tuple	3.2.1	O	O
0x20	CISTPL_MANFID	Manufacturer Identification String Tuple	3.2.9	M	O
0x21	CISTPL_FUNCID	Function Identification Tuple	3.2.7	n/a	M
0x22	CISTPL_FUNCE	Function Extensions	3.2.6	n/a	M
0x80-0x8F		Vendor Unique Tuples	None	O	O
0x91	CISTPL_SDIO_STD	Additional information for functions built to support application specifications for standard SDIO functions.	6.1.2	n/a	M ¹ :Standard SDIO Function O ¹ :non-Standard SDIO Function
0x92	CISTPL_SDIO_EXT	Reserved for future use with SDIO devices.	6.1.3	n/a	n/a
0xFF	CISTPL_END	The End-of-chain Tuple	3.1.2	M	M

Table 16-2 Tuples Supported by SDIO Cards

SDIO Simplified Specification Version 2.00

Note 1: the use of CISTPL_SDIO_STD is mandatory for all functions that claim to support a SDIO standard interface specification (see 1.3). If the function does not support a standard SDIO interface, this tuple should be used with a value of 0.

16.6 CISTPL_MANFID: Manufacturer Identification String Tuple

The manufacturer identification tuple contains information about the manufacturer of a SDIO Card. Two types of information are provided: the SDIO Card's manufacturer and a manufacturer card number. This tuple shall be present in the card common CIS.

This should also be present in each function's CIS. This allows a function to override the card common manufacturer information so the driver can take advantage of unique features.

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_MANFID (20H)							
0x01	TPL_LINK Link to next tuple (at least 4)							
0x02-0x03	TPLMID_MANF SDIO Card manufacturer code							
0x04-0x05	TPLMID_CARD manufacturer information (Part Number and/or Revision)							

Table 16-3 CISTPL_MANFID: Manufacturer Identification Tuple

The *TPLMID_MANF* field identifies the SDIO Card's manufacturer. New codes are assigned by both PCMCIA and JEIDA. The first 256 identifiers (0x0000 through 0x00FF) are reserved for manufacturers who have JEDEC IDs assigned by JEDEC Publication 106. Manufacturers with JEDEC IDs may use their eight-bit JEDEC manufacturer code as the least significant eight bits of their SDIO Card manufacturer code. In this case, the most significant eight bits shall be zero (0). For example, if a JEDEC manufacturer code is 89H, their SDIO Card manufacturer code is 0x0089. If a SDIO card manufacturer does not currently have a *TPLMID_MANF* assigned, one can be obtained at little or no cost from the PCMCIA. The *TPLMID_CARD* field is reserved for the use of the SDIO Card's manufacturer. It is anticipated that the field will be used to store card identifier and revision information.

16.7 SDIO Specific Extensions

SDIO cards use two to four tuples to provide additional information about the card (common) and each function. The first is the Function ID tuple. The changes for SDIO are detailed in the next sections.

16.7.1 CISTPL_FUNCID: Function Identification Tuple

To identify an SDIO card, the CISTPL_FUNCID tuple shall exist in all CIS areas. This means there shall be a CISTPL_FUNCID in the common CIS space chain and one in each function's CIS space chain. The format of this tuple is shown in

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_FUNCID (0x21)							
0x01	TPL_LINK Link to next tuple (0x02)							
0x02	TPLFID_FUNCTION Card function code (0x0C)							
0x03	TPLFID_SYSINIT System initialization bit mask. (Not used, set to 0x00)							

Table 16-4 CISTPL_FUNCID Tuple

The function identification tuple contains information about the functionality provided by an SDIO Card. Information is also provided to enable system utilities to decide if the SDIO Card should be configured during system initialization. Since additional function specific information is available, one or more function extension tuples follow this tuple. The *TPLFID_FUNCTION* field contains an identifier assigned by PCMCIA (0x0C) to identify the SDIO device class.

SDIO Simplified Specification Version 2.00**16.7.2 CISTPL_FUNCCE: Function Extension Tuple**

The CISTPL_FUNCCE tuple provides standard information about the card (common) and each individual function. There shall be one CISTPL_FUNCCE in each function's CIS immediately following the CISTPL_FUNCID tuple. The format of the CISTPL_FUNCCE is shown in Table 16-5.

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_FUNCCE (0x22)							
0x01	TPL_LINK Link to next tuple (see following sections)							
0x02	TPLFE_TYPE Type of extended data (see following sections)							
0x03-n	TPLFE_DATA Function information (see following sections)							

Table 16-5 CISTPL_FUNCCE Tuple General Structure

There are two versions of the CISTPL_FUNCCE tuple, one for the common CIS (function 0) and a version used by the individual function's CIS (1-7). Both types are described below.

16.7.3 CISTPL_FUNCCE Tuple for Function 0 (common)

This version of the CISTPL_FUNCCE tuple gives the host common information about the card. There shall be only one of these tuples, located in the CIS for function 0 following the CISTPL_FUNCID. The format of this tuple is shown in Table 16-6.

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_FUNCCE (0x22)							
0x01	TPL_LINK Link to next tuple (0x04)							
0x02	TPLFE_TYPE Type of extended data (0x00)							
0x03-0x04	TPLFE_FNO_BLK_SIZE							
0x05	TPLFE_MAX_TRAN_SPEED							

Table 16-6 TPLFID_FUNCTION Tuple for Function 0 (common)

The fields in this tuple have the following definition:

Field	Description
TPLFE_FNO_BLK_SIZE	This is both the maximum block size and byte count that function 0 can support. A value of zero is not valid and shall not be used.
TPLFE_MAX_TRAN_SPEED	This byte indicates the maximum transfer rate per one data line during data transfer. This value applies to all functions in the SDIO card. This value shall be 25 Mb/Sec (0x32) for all Full-Speed SDIO cards. The minimum value for Low-Speed SDIO cards shall be 400 Kb/Sec (0x48). The format is identical to the TRAN_SPEED value stored in the CSD of SD memory cards. The maximum data transfer rate is coded according to the following method: Bits 2:0 contain the transfer rate unit coded as follows: 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/s, 4... 7=reserved Bits 6:3 contain the time value codes as follows: 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.5, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.0, C=5.5, D=6.0, E=7.0, F=8.0 Bit 7 is reserved and shall be zero A Combo Card shall support 25MHz clock (0x32)

Table 16-7 TPLFID_FUNCTION Field Descriptions for Function 0 (common)

SDIO Simplified Specification Version 2.00**16.7.4 CISTPL_FUNCCE Tuple for Function 1-7**

This version of the CISTPL_FUNCCE tuple gives the host common information about each individual function on a per-function basis. There shall be one of these tuples, located in the CIS for each function following the CISTPL_FUNCID. The format of this tuple is shown in Table 16-8.

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_FUNCCE (0x22)							
0x01	TPL_LINK Link to next tuple (0x2A)							
0x02	TPLFE_TYPE Type of extended data (0x01)							
0x03	TPLFE_FUNCTION_INFO							
0x04	TPLFE_STD_IO_REV							
0x05-0x08	TPLFE_CARD_PSN							
0x09-0x0C	TPLFE_CSA_SIZE							
0x0D	TPLFE_CSA_PROPERTY							
0x0E-0x0F	TPLFE_MAX_BLK_SIZE							
0x10-0x13	TPLFE_OCR							
0x14	TPLFE_OP_MIN_PWR							
0x15	TPLFE_OP_AVG_PWR							
0x16	TPLFE_OP_MAX_PWR							
0x17	TPLFE_SB_MIN_PWR							
0x18	TPLFE_SB_AVG_PWR							
0x19	TPLFE_SB_MAX_PWR							
0x1A-0x1B	TPLFE_MIN_BW							
0x1C-0x1D	TPLFE_OPT_BW							
0x1E-0x1F	TPLFE_ENABLE_TIMEOUT_VAL							
0x20-0x21	TPLFE_SP_AVG_PWR_3.3V							
0x22-0x23	TPLFE_SP_MAX_PWR_3.3V							
0x24-0x25	TPLFE_HP_AVG_PWR_3.3V							
0x26-0x27	TPLFE_HP_MAX_PWR_3.3V							
0x28-0x29	TPLFE_LP_AVG_PWR_3.3V							
0x2A-0x2B	TPLFE_LP_MAX_PWR_3.3V							

Table 16-8 TPLFID_FUNCTION Tuple for Function 1-7

The fields in this tuple have the following definition:

Field	Description
TPLFE_FUNCTION_INFO	Bit significant information about the Function The bits are defined in Table 16-10
TPLFE_STD_IO_REV	This 8-bit value contains the version level of the Application Specification for Standard SDIO Functions that this function supports. The format is x.y where x is the major version (4-bits) and y is the minor version level. For example if the version is 2.4 the value would be 0x24. If this function does not support an SDIO standard function, the value shall be 0x00.
TPLFE_CARD_PSN	The Product Serial Number is a 32 bit unsigned binary integer. Support of a serial number is optional, if there is no serial number, this field shall be 0x00000000 While a unique serial number is not required for all devices, it is strongly recommended that SDIO card vendors place a unique serial number in this field to assist the operating systems in differentiating multiple cards of the same type. Also note that some Application Specifications for Standard SDIO Functions require a unique serial number in this field. The individual Application Specification will indicate if support of this field is mandatory.

SDIO Simplified Specification Version 2.00

TPLFE_CSA_SIZE	Size of the CSA space available for this function in bytes
TPLFE_CSA_PROPERTY	This byte contains flags identifying properties of this function's CSA. The bits are defined in Table 16-11
TPLFE_MAX_BLK_SIZE	This is both the maximum block size and byte count that this function can support. A value of zero is not valid and shall not be used.
TPLFE_OCR	This is the OCR value for this function. The format is identical to the 32-bit OCR format used by SD memory devices. For more details, see section 5.1 of the SD Physical Specification.
TPLFE_OP_MIN_PWR	This is the minimum current, in mA, required by this function when operating. This value is valid for all voltages supported by this function. If the required current exceeds 200mA, this value shall be zero.
TPLFE_OP_AVG_PWR	This is the average current, in mA, required by this function when operating. This value is valid for all voltages supported by this function. If the required current exceeds 200mA, this value shall be zero
TPLFE_OP_MAX_PWR	This is the maximum (peak) current, in mA, required by this function when operating. This value is valid for all voltages supported by this function. If the required current exceeds 200mA, this value shall be zero
TPLFE_SB_MIN_PWR	This is the minimum current, in mA, required by this function when in the standby condition. If this function does not support standby, this value shall be 0x00. The method to place this function in the standby state and the capabilities it has while in standby are vendor defined. This value is valid for all voltages supported by this function. Note that this value is valid only for standard power SDIO cards or high-power cards when EMPC is 0. With an 8-bit field, the range is from 0 to 254 mA. A value of 255 (0xFF) is used to indicate a value of 255mA or greater.
TPLFE_SB_AVG_PWR	This is the average current, in mA, required by this function when in the standby condition. If this function does not support standby, this value shall be 0x00. The method to place this function in the standby state and the capabilities it has while in standby are vendor defined. This value is valid for all voltages supported by this function. Note that this value is valid only for standard power SDIO devices cards or high-power cards when EMPC is 0. With an 8-bit field, the range is from 0 to 254 mA. A value of 255 (0xFF) is used to indicate a value of 255mA or greater.
TPLFE_SB_MAX_PWR	This is the maximum current, in mA, required by this function when in the standby condition. If this function does not support standby, this value shall be 0x00. The method to place this function in the standby state and the capabilities it has while in standby are vendor defined. This value is valid for all voltages supported by this function. Note that this value is valid only for standard power SDIO devices cards or high-power cards when EMPC is 0. With an 8-bit field, the range is from 0 to 254 mA. A value of 255 (0xFF) is used to indicate a value of 255mA or greater.
TPLFE_MIN_BW	This is the minimum data transfer bandwidth, in KB/sec, needed by this function to successfully operate. If this function has no minimum necessary bandwidth, these bytes shall be 0x0000.
TPLFE_OPT_BW	This is the data transfer bandwidth, in KB/sec, needed by this function to function at an optimum level. If this function has no optimum bandwidth, these bytes shall be 0x0000.
TPLFE_ENABLE_TIMEOUT_VAL	(Added in SDIO Rev 1.1) This 16-bit value indicates the function's required time-out value for coming ready after being enabled. This per-function value indicates the time a host should wait from asserting IOEx until expecting the card to indicate ready by asserting IORx. Different SDIO functions take different amounts of time to become ready after being enabled due to different internal initialization requirements. The required time-out limit is in 10mS steps, allowing a range of 0-655.35

SDIO Simplified Specification Version 2.00

	seconds. If the cards required no time-out, this field shall be set to 0x0000.
TPLFE_SP_AVG_PWR_3.3V	This value is the same as that of TPLFE_OP_AVG_PWR.
TPLFE_SP_MAX_PWR_3.3V	This value is the same as that of TPLFE_OP_MAX_PWR.
TPLFE_HP_AVG_PWR_3.3V	This 16-bit value indicates the average current, in mA, required by this function when operating in the Higher Current Mode (EMPC=1 & EPS=0). This value indicates a current range from 1 to 65535 mA.
TPLFE_HP_MAX_PWR_3.3V	This 16-bit value indicates the peak current, in mA, required by this function when operating in the Higher Current Mode. This value indicates a current range from 1 to 65535 mA.
TPLFE_LP_AVG_PWR_3.3V	This 16-bit value indicates the average current, in mA, required by this function when operating in the Lower Current Mode (EMPC=1 & EPS=1). This value indicates a current range from 1 to 65535 mA.
TPLFE_LP_MAX_PWR_3.3V	This 16-bit value indicates the maximum (peak) current, in mA, required by this function when operating in the Lower current mode (EMPC=1 & EPS=1). This value indicates a current range from 1 to 65535 mA.

Table 16-9 TPLFID_FUNCTION Field Descriptions for Functions 1-7

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	FN_WUS

Table 16-10 TPLFE_FUNCTION_INFO Definition

The FN_WUS (Wake Up Support) bit signals the function's support of wake-up when the function is placed into a low power state and the SDCLK is stopped. The intent is to allow a SDIO card to be placed into a low power condition and still signal the host of a wake-up event. If this bit is set to 1, then this function may be placed into a low power state using some function standard or vendor defined method and the SDCLK can be stopped with the power remaining applied to the card. Irrespective of the state of this bit, the host may stop the SDCLK as allowed by the SD Physical Specification.

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	CSA_NF	CSA_WP

Table 16-11 TPLFE_CSA_PROPERTY Definition

The CSA_WP bit is used to write protect the CSA for this function, or identifies it as read-only. If this bit is set to 1, then the CSA of this function is either write protected or read-only. If this bit is clear, then the host may write the CSA. Setting of this bit for R/W devices is handled in a vendor-defined method.

The CSA_NF bit is used to indicate to the host that the CSA area should not be reformatted. If the bit is set to 1, the host may not format the CSA file structure. If this bit is cleared to 0, then the host may reformat the file system of the CSA.

SDIO Simplified Specification Version 2.00**16.7.5 CISTPL_SDIO_STD: Function is a Standard SDIO Function**

This tuple code (0x91) has been reserved for use by SDIO devices that conform to the application specifications for standard SDIO functions, as defined in those separate specifications. The exact format for this tuple can be found in those specifications. The basic format only is provided here and the reader is directed to the appropriate Application Specification for Standard SDIO Functions for more complete details:

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_SDIO_STD (91h)							
0x01	TPL_LINK Link to next tuple ($2 \leq n \leq 255$) n is the number of bytes in the tuple body							
0x02	TPLSDIO_STD_ID							
0x03	TPLSDIO_STD_TYPE							
0x04 ... (n+1)	TPLSDIO_STD_DATA If n = 2, TPLSDIO_STD_DATA does not exist.							

Table 16-12 CISTPL_SDIO_STD: Tuple Reserved for SDIO Cards

The fields in this tuple have the following definition:

Field	Description
TPLSDIO_STD_ID	This 8-bit code identifies the Standard SDIO Function type for which this tuple supplies additional information. The available codes can be found in the I/O device interface code entry of the FBR (see 6.10)
TPLSDIO_STD_TYPE	This 8-bit value identifies the format and type of data contained within the body of this tuple. If this value is 0x00, then only 1 standard data structure has been defined for this Standard SDIO Function. If this value is non-zero, then this byte identifies which tuple data format is being used for the data. This code is defined in the Application Specification for Standard SDIO Functions.
TPLSDIO_STD_DATA	The data component of this tuple (n-2 bytes). The format of this data structure is defined in the Application Specification for Standard SDIO Functions.

16.7.6 CISTPL_SDIO_EXT: Tuple Reserved for SDIO Cards

This tuple code (0x92) has been reserved for future use by SDIO cards. The actual format of the data has not been established at this time. The basic format of this tuple is:

Byte	7	6	5	4	3	2	1	0
0x00	TPL_CODE CISTPL_SDIO_EXT (92h)							
0x01	TPL_LINK Link to next tuple ($0 \leq n \leq 255$) n is the number of bytes in the tuple body							
0x02 ... (n+1)	TPLSDIO_EXT_DATA The data component of this tuple (n bytes) If n=0, TPLSDIO_EXT_DATA does not exist.							

Table 16-13 CISTPL_SDIO_EXT: Tuple Reserved for SDIO Cards

Appendix A

(Normative)

A.1 SD and SPI Command List

Table A-14 and Table A-15 show the commands that are supported by SD memory and SDIO devices in both SPI and SD modes. If a command is not identified as either mandatory or optional, then it is not supported by that device.

Supported Commands	Abbreviation	SDMEM System	SDIO System	Comments
CMD0	GO_IDLE_STATE	Mandatory	Mandatory	Used to change from SD to SPI mode
CMD2	ALL_SEND_CID	Mandatory		CID not supported by SDIO
CMD3	SEND_RELATIVE_ADDR	Mandatory	Mandatory	
CMD4	SET_DSR	Optional		DSR not supported by SDIO
CMD5	IO_SEND_OP_COND		Mandatory	
CMD6	SWITCH_FUNC	Mandatory ¹	Mandatory ¹	Added in Part 1 v1.10
CMD7	SELECT/DESELECT_CARD	Mandatory	Mandatory	
CMD9	SEND_CSD	Mandatory		CSD not supported by SDIO
CMD10	SEND_CID	Mandatory		CID not supported by SDIO
CMD12	STOP_TRANSMISSION	Mandatory		
CMD13	SEND_STATUS	Mandatory		Card Status includes only SDMEM information
CMD15	GO_INACTIVE_STATE	Mandatory	Mandatory	
CMD16	SET_BLOCKLEN	Mandatory		
CMD17	READ_SINGLE_BLOCK	Mandatory		
CMD18	READ_MULTIPLE_BLOCK	Mandatory		
CMD24	WRITE_BLOCK	Mandatory		
CMD25	WRITE_MULTIPLE_BLOCK	Mandatory		
CMD27	PROGRAM_CSD	Mandatory		CSD not supported by SDIO
CMD28	SET_WRITE_PROT	Optional		
CMD29	CLR_WRITE_PROT	Optional		
CMD30	SEND_WRITE_PROT	Optional		
CMD32	ERASE_WR_BLK_START	Mandatory		
CMD33	ERASE_WR_BLK_END	Mandatory		
CMD38	ERASE	Mandatory		
CMD42	LOCK_UNLOCK	Optional		
CMD52	IO_RW_DIRECT		Mandatory	
CMD53	IO_RW_EXTENDED		Mandatory	Block mode is optional
CMD55	APP_CMD	Mandatory		
CMD56	GEN_CMD	Mandatory		
ACMD6	SET_BUS_WIDTH	Mandatory		
ACMD13	SD_STATUS	Mandatory		
ACMD22	SEND_NUM_WR_BLOCKS	Mandatory		
ACMD23	SET_WR_BLK_ERASE_COUNT	Mandatory		
ACMD41	SD_APP_OP_COND	Mandatory		
ACMD42	SET_CLR_CARD_DETECT	Mandatory		
ACMD51	SEND_SCR	Mandatory		SCR not supported by SDIO

Table A-14 SD Mode Command List

¹ For Part 1 v1.10 or higher Memory or Combo Cards

SDIO Simplified Specification Version 2.00

Supported Commands	Abbreviation	SDMEM System	SDIO System	Comments
CMD0	GO_IDLE_STATE	Mandatory	Mandatory	Used to change from SD to SPI mode
CMD1	SEND_OP_COND	Mandatory		
CMD5	IO_SEND_OP_COND		Mandatory	
CMD6	SWITCH_FUNC	Mandatory ¹	Mandatory ¹	Added in Part 1 v1.10
CMD9	SEND_CSD	Mandatory		CSD not supported by SDIO
CMD10	SEND_CID	Mandatory		CID not supported by SDIO
CMD12	STOP_TRANSMISSION	Mandatory		
CMD13	SEND_STATUS	Mandatory		Card Status includes only SDMEM information.
CMD16	SET_BLOCKLEN	Mandatory		
CMD17	READ_SINGLE_BLOCK	Mandatory		
CMD18	READ_MULTIPLE_BLOCK	Mandatory		
CMD24	WRITE_BLOCK	Mandatory		
CMD25	WRITE_MULTIPLE_BLOCK	Mandatory		
CMD27	PROGRAM_CSD	Mandatory		CSD not supported by SDIO.
CMD28	SET_WRITE_PROT	Optional		
CMD29	CLR_WRITE_PROT	Optional		
CMD30	SEND_WRITE_PROT	Optional		
CMD32	ERASE_WR_BLK_START	Mandatory		
CMD33	ERASE_WR_BLK_END	Mandatory		
CMD38	ERASE	Mandatory		
CMD42	LOCK_UNLOCK	Optional		
CMD52	IO_RW_DIRECT		Mandatory	
CMD53	IO_RW_EXTENDED		Mandatory	Block mode is optional
CMD55	APP_CMD	Mandatory		
CMD56	GEN_CMD	Mandatory		
CMD58	READ_OCR	Mandatory		
CMD59	CRC_ON_OFF	Mandatory	Mandatory	
ACMD13	SD_STATUS	Mandatory		
ACMD22	SEND_NUM_WR_BLOCKS	Mandatory		
ACMD23	SET_WR_BLK_ERASE_COUNT	Mandatory		
ACMD41	SD_APP_OP_COND	Mandatory		
ACMD42	SET_CLR_CARD_DETECT	Mandatory		
ACMD51	SEND_SCR	Mandatory		SCR includes only SD-MEM information.

Table A-15 SPI Mode Command List¹ For Part 1 v1.10 or higher Memory or Combo Cards

Appendix B

(Normative)

B.1 Normative References

The following documents are referenced by this specification.

This specification can apply to any released versions of below SD Specifications after Version 2.00.

- 1) SD Specifications Part 1 PHYSICAL LAYER SPECIFICATION
Version 2.00 May 9, 2006
- 2) SD Specifications Part 1 miniSD Card Addendum
Version 2.00 January 30, 2007
- 3) SD Specifications Part 2 FILE SYSTEM SPECIFICATION
Version 2.00 May 9, 2006
- 4) ISO/IEC9293:1994
Information technology - Volume and file structure of disk cartridges for information interchange
- 5) PC CARD STANDARD Release 8.1 2002
Volume 4 Metaformat Specification

Appendix C

C.1 Abbreviations and Terms

3C	The original 3 companies that established the SDA and the SD card specification
ACMD	Application Command An additional set of commands created to support SD specific requirements.
Block	A number of bytes, basic data transfer unit
CCCR	Common Card Control Registers
CD	Connect/Disconnect
CIA	Common Information Area
CID	Card IDentification number register
CIS	Card Information Structure
CLK	Clock signal
CMD	Command line or SD bus command (if extended CMDXX)
Combo Card	A card that includes both SDIO and SD memory
CRC	Cyclic Redundancy Check
CS	Chip or Card Select
CSA	Code Storage Area
CSD	Card Specific Data register
DAT[x]	Data line where x is in the range of 0 to 3
DSR	Driver Stage Register
ECC	Error Correction Code
Embedded SDIO	I/O or combo devices that are embedded in a host device that utilize the SDIO electrical and command interface, but are not intended to be removed.
EMI	Electro-Magnetic Interference
ESC	External Signal Contacts
ESD	Electro-Static Discharge
FAT	File Allocation Table
FBR	Function Basic Registers
FIFO	First In-First Out buffer
Flash	A type of multiple time programmable non volatile memory
Full-Size SDIO Card	A SDIO card with physical dimensions based on SD Physical Specification Version 1.01
Function	An I/O device contained within an SDIO card
High-Power SDIO	A SDIO device that requires up to 500 mA rather than the standard 200 mA allowed for a Standard-Power SDIO device
hi-Z	A three-state driver in the high-impedance state
Interrupt	A signal from the SDIO device to the host signaling the need for attention
Interrupt Period	The times that a card may generate an interrupt signal on the SD bus
Legacy Slot	SD Slot that supports only the SD Physical Specification Version 1.01

SDIO Simplified Specification Version 2.00

LOW, HIGH	Binary interface states with defined assignment to a voltage level
MBIO	Multi-Block I/O
miniSDIO Card	A SDIO card based on the miniSD card form factor.
MMC	MultiMedia Card
MSB, LSB	The Most Significant Bit or Least Significant Bit
OCR	Operation Conditions Register
PCMCIA	Personal Computer Memory Card International Association
PnP	Plug and Play a means to identify an SDIO device and optionally load applications and/or drivers without user intervention
R/O	Read Only
R/W	Read or Write
RAW	Read After Write
RCA	Relative Card Address register
Resume	Re-starting the temporarily halted data transfer
RFU	Reserved for Future Use. Normally Read-Only and set to 0
ROM	Read Only Memory
RWC	Read Wait Control
SCR	SD Configuration Register
SDA	SD Association
SDCLK	SD clock signal
SDIO	SD Input/Output
SDIO aware	A host designed to support the signals and protocol of SDIO devices
SPI	Serial Peripheral Interface
Standard-Power SDIO	A SDIO device that operates with 200 mA or less of current at all times.
Stuff bit(s)	Filling bit(s) to ensure fixed length frames for commands and responses.
Suspend	Temporarily halting the transfer of data
TBD	To Be Determined (in the future)
Three-state driver	A driver stage which has three output driver states: HIGH, LOW and high impedance (which means that the interface does not have any influence on the interface level)
Tuple	Data blocks in a linked list or chain format
VDD	+ Power supply
VSS	Power supply ground
W/O	Write Only
WP	Write Protect

Appendix D

(Informative)

Appendix D is not included in the Simplified Specification.

SD Association