S3F49DAX

MultiMediaCard Controllers

Datasheet

Revision 1.0

SAMSUNG ELECTRONICS

PRODUCT OVERVIEW

S3F49DAX 16/32-bit RISC microprocessor would be designed to provide a cost-effective, low power, small die size and high performance micro-controller solution for Low voltage MMC (MultiMediaCard). To reduce total system cost, S3F49DAX also provides the followings: 16K Byte SRAM, 48K Byte Internal NOR Flash memory, NAND Flash controller, interrupt controller, 2-channel DMA (Direct Memory Access), 3-channel Timer and Analog block including IVC (Internal Voltage Converter), POR (Power-On Reset) and RCOSC (RC Oscillator).

S3F49DAX microprocessor would be developed using an ARM7TDMI-S core, 0.18um CMOS standard cells housed in 64UELP package. And S3F49DAX adopts the new bus architecture, AMBA 2.0 (Advanced Micro-controller Bus Architecture).

The integrated on-chip functions that are described in this document include:

- Compliant with MMC (MultiMediaCard) system specification version 3.3
- Built in internal memory: 48K Byte NOR Flash for program, 16K Byte SRAM for data buffer and stack
- Integrated NAND Flash memory controller
- Built in ECC (Error Correction Code) Engine.
- Two-channel DMA (Direct Memory Access)
- Three-channel Timer and one-channel UART with interrupt-based operation
- Interrupt controller: 21 interrupt sources
- Power control: Normal, Slow and Standby mode
- Built in analog function block: POR (Power-On Reset), RCOSC (RC Oscillator) and IVC (Internal Voltage Converter)

FEATURES

Architecture

- Integrated system for Low Voltage MMC (MultiMediaCard).
- 16/32-bit RISC architecture and powerful instruction set with ARM7TDMI-S CPU core.
- Internal AMBA (Advanced Micro-controller Bus Architecture, AMBA 2.0)

ARM7TDMI-S

- Supports fixed little endian mode.
- 32/16-bit RISC architecture. (ARM V4T)
- 32-bit ARM instruction set for maximum performance and flexibility
- 16-bit Thumb instruction set for increased code density
- Coprocessor Interface / JTAG debug interface unit

MMC Interface

- Fully compliant with MultiMediaCard specification version 3.3
- Supports MMC (MultiMediaCard) mode and SPI mode.
- Supports command classes which are class 0, 2, 4, 5, 6, 7
- CRC (Cyclic Redundancy Check) bits generation and check

NAND Flash Memory Controller

- Supports variable capacity NAND Flash memory.
- Supports low voltage NAND Flash memory (Operating voltage range of NAND Flash: 1.65 V to 1.95 V)
- Supports up to Two NAND flash memories extension.

NOR Flash Memory Controller

- Built in 48K Byte NOR Flash memory.
- Supports two type programming mode: User programming mode, Tool programming mode
- Supports three type data protection: Read Protection, LD Protection, Hardware Protection
- Supports Byte Programming/Option Sector Programming/Chip Erase/Sector Erase/Verify Sequence.

SRAM Memory Controller

- Built in 16K Byte SRAM memory for FTL map table, ECC, Global variables, Stack & Heap area.

Interrupt Controller

- Supports normal or fast interrupt modes (IRQ, FIQ)
- Supports vectored interrupts (Hard-wired Interrupt)
- S/W programmable interrupt priority.

DMA (Direct Memory Access)

- Two dedicated DMA (Host I/F DMA, NAND Flash I/F DMA)

Timer

- One-channel 16-bit timer & 32-bit timer.
- Watch-dog Timer
- Interrupt-based operation (Support interval mode)

UART

- One-channel UART with interrupt-based operation
- Programmable baud rate.

DES / T-DES

- Built-in hardware DES
- Four keys for encryption and decryption
- Start / Stop control

Random Number Generator

- One 16-bit random number generator in accordance with FIPS140-2
- Start / Stop Control

Analog Function Block

- Built-in RCOSC (RC Oscillator): 18 MHz @ 1.8 V
- Built-in POR (Power-on Reset) and LVD (Low Voltage Detector)
- Built-in IVC: 3.3 V to 1.8 V

Operating Frequency

- 18 MHz ± 3 MHz (RCOSC Output Frequency)

Operating Voltage

- Internal Logic: 1.65 V to 1.95 V
- I/O: 1.65 V to 3.6 V
- NAND flash interface: 1.65 V to 3.6 V
- In case of high voltage MMC, You must use the interface power (2.7 V to 3.6 V).

Operating Temperature

- Regular specification: –25 °C to +85 °C

Package Type

- 64-pin UELP

BLOCK DIAGRAM

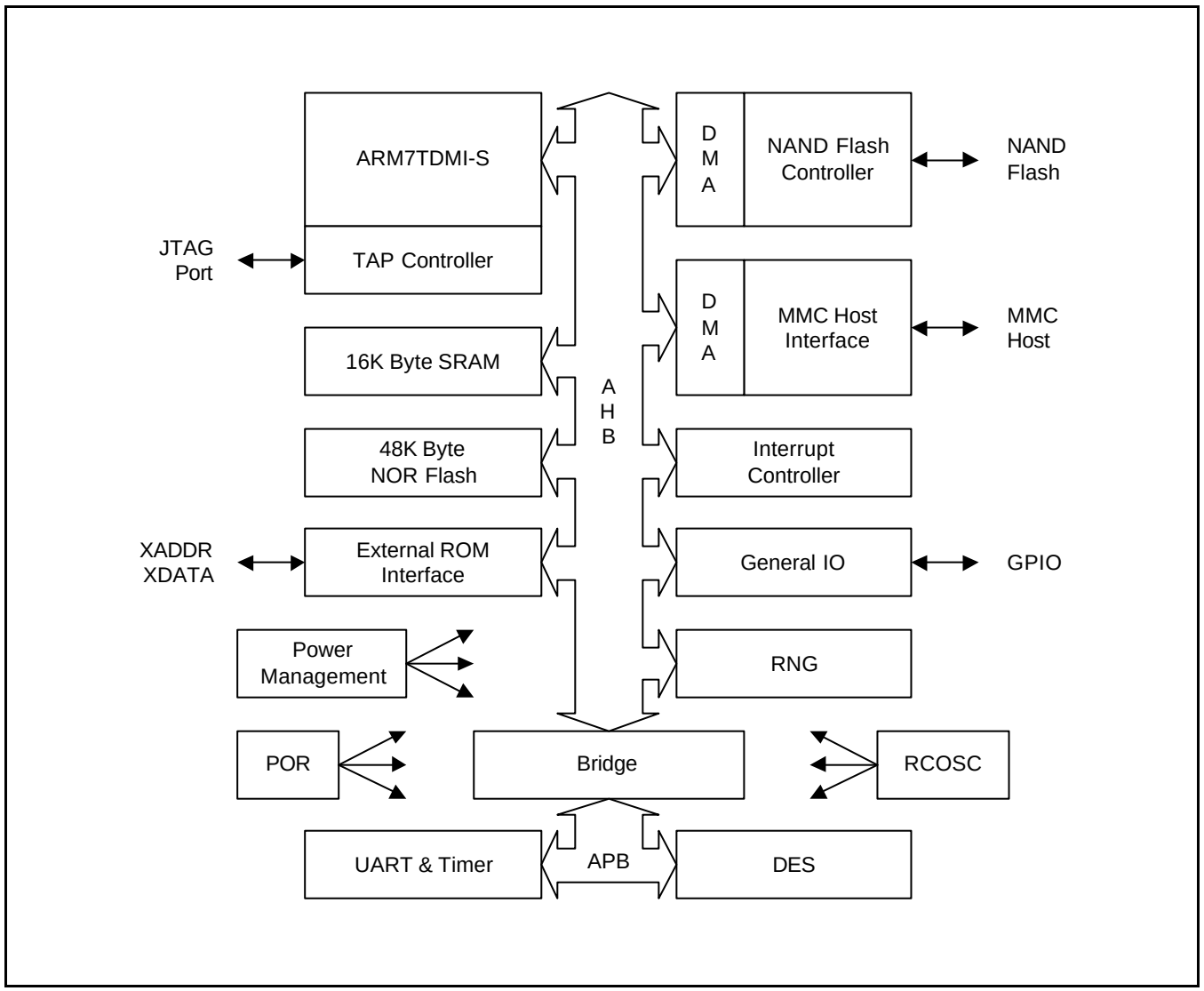

Figure 1. S3C49DAX Block Diagram

PIN ASSIGNMENTS

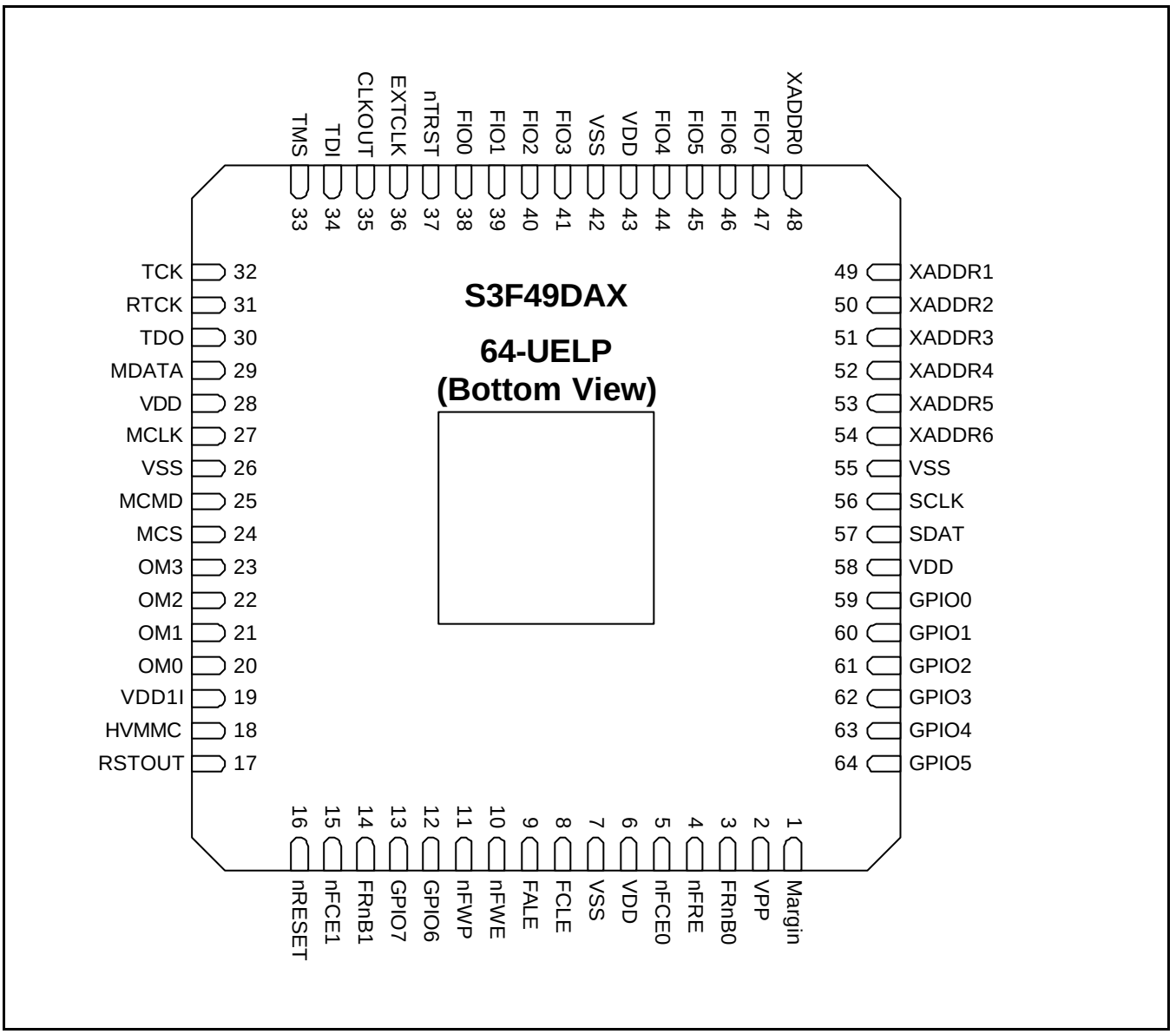

Figure 2. S3F49DAX Pin Assignments

Table 1. 64-Pin ELP Pin Assignment

Pin No	Pin Name	Function	I/O State	I/O Type
1	Margin	Internal NOR flash cell margin test	I	phia
2	VPP	Internal NOR flash program power	I	vpp12
3	FRnB0	NAND flash memory ready/busy signal 0	I	phicu10
4	nFRE	NAND flash memory read enable signal	O	phmbct
5	nFCE0	NAND flash memory chip enable signal 0	O	phmbct
6	VDD	MMC power	P	vdd3io
7	VSS	MMC ground	G	vss3io
8	FCLE	NAND flash memory command latch enable signal	O	phmbct
9	FALE	NAND flash memory address latch enable signal	O	phmbct
10	nFWE	NAND flash memory write enable signal	O	phmbct
11	nFWP	NAND flash memory write protect signal	O	phmbct
12	GPIO6/XDATA6	General IO port 6/External ROM data 6	I/O	phbcu60t
13	GPIO7/XDATA7	General IO port 7/External ROM data 7	I/O	phbcu60t
14	FRnB1	NAND flash memory ready/busy signal 1	I	phicu10
15	nFCE1	NAND flash memory chip enable signal 1	O	phmbct
16	nRESET	Global reset input for the S3F49DAX	I	phicu60
17	RSTOUT	Internal POR monitoring output	O	phob4
18	HVMMC	High voltage MMC select	O	phic
19	IVCOUT	Internal V_{DD} (1.8V)	P	vdd1i
20	OM0	Operation mode select 0	I	phic
21	OM1	Operation mode select 1	I	phic
22	OM2	Operation mode select 2	I	phic
23	OM3	Operation mode select 3	I	phic
24	MCS	MMC card select signal for SPI mode	I	phmicu60
25	MCMD	MMC command signal	I/O	phmbcu10t
26	VSS	MMC ground	G	vss3io
27	MCLK	MMC clock signal	I	phmic
28	VDD	MMC power	P	vdd3io
29	MDATA	MMC data signal	I/O	phmbcu60t
30	TDO	TAP controller data output	O	phob
31	RTCK	Returned TCK	O	phob
32	TDI	TAP controller data input	I	phicu60

Table 1. 64-Pin ELP Pin Assignment (Continued)

Pin No	Pin Name	Function	I/O State	I/O Type
33	TMS	TAP controller mode select	I	phicu60
34	TDI	TAP controller data input	I	phicu60
35	CLKOUT	Internal VCO clock monitoring output	O	phob
36	EXTCLK	External clock input	I	phic
37	nTRST	TAP controller reset	I	phicu60
38	FIO0	NAND flash memory IO signal 0	I/O	phmbct
39	FIO1	NAND flash memory IO signal 1	I/O	phmbct
40	FIO2	NAND flash memory IO signal 2	I/O	phmbct
41	FIO3	NAND flash memory IO signal 3	I/O	phmbct
42	VSS	MMC ground	G	vss3io
43	VDD	MMC power	P	vdd3io
44	FIO4	NAND flash memory IO signal 4	I/O	phmbct
45	FIO5	NAND flash memory IO signal 5	I/O	phmbct
46	FIO6	NAND flash memory IO signal 6	I/O	phmbct
47	FIO7	NAND flash memory IO signal 7	I/O	phmbct
48	XADDR0	External ROM address 0 out for test	I/O	phob
49	XADDR1	External ROM address 1 out for test	I/O	phob
50	XADDR2	External ROM address 2 out for test	I/O	phob
51	XADDR3	External ROM address 3 out for test	I/O	phob
52	XADDR4	External ROM address 4 out for test	I/O	phob
53	XADDR5	External ROM address 5 out for test	I/O	phob
54	XADDR6	External ROM address 6 out for test	I/O	phob
55	VSS	MMC ground	G	vss3io
56	SCLK	Serial clock input for NOR Flash programming	I	phicu60
57	SDAT	Serial data signal for NOR Flash programming	I/O	phbcu60t
58	VDD	MMC power	P	vdd3io
59	GPIO0/XDATA0	General IO port 0/External ROM data 0	I/O	phbcu60t
60	GPIO1/XDATA1	General IO port 1/External ROM data 1	I/O	phbcu60t
61	GPIO2/XDATA2	General IO port 2/External ROM data 2	I/O	phbcu60t
62	GPIO3/XDATA3	General IO port 3/External ROM data 3	I/O	phbcu60t
63	GPIO4/XDATA4	General IO port 4/External ROM data 4	I/O	phbcu60t
64	GPIO5/XDATA5	General IO port 5/External ROM data 5	I/O	phbcu60t

Table 2. I/O Type Description

I/O Type	Description
phia	High voltage analog input buffer
phic	High voltage CMOS input buffer
phicu60	High voltage CMOS input buffer with pull-up resistor (60 kΩ)
phicu10	High voltage CMOS input buffer with pull-up resistor (10 kΩ)
phmic	High voltage CMOS MMC input buffer
phob	High voltage normal output buffer, Io = 4 mA
phmbct	High voltage CMOS MMC input buffer and tri-stat output buffer
phmbc60t	High voltage CMOS MMC input buffer with pull-up resistor (60 kΩ) and tri-stat output buffer
phmbc10t	High voltage CMOS MMC input buffer with pull-up resistor (10 kΩ) and tri-stat output buffer
vdd3io	3.3 V V_{DD}
vss3io	Ground
vdd1i	1.8 V V_{DD} (Internal Regulator Output)

SIGNAL DESCRIPTIONS

Table 3. S3F49DAX Signal Description

Signal	I/O	Description
NAND Flash Interface		
FIO [7:0]	I/O	**Data Input/Output:** These pins are use to input data during read operations, and to output command, address and data.
nFCE [1:0]	O	**Chip Enable:** The nFCE output is the device selection control.
FCLE	O	**Command Latch Enable:** The FCLE output controls the activating path for commands to the NAND Flash.
FALE	O	**Address Latch Enable:** The FALE output controls the activating path for address to NAND Flash
nFRE	O	**Read Enable:** The nFRE output controls the serial data-out from NAND Flash, and when active fetches the data onto the I/O bus.
nFWE	O	**Write Enable:** The nFWE output controls writes to I/O port
nFWP	O	**Write Protect:** The nFWP signal provides inadvertent write/erase protection during power transitions.
FRnB[1:0]	I	**Ready/Busy:** These pins indicate the status of the NAND Flash operation. When low, it indicates that a program, erase or random read operation is in process.
MMC Interface		
MCS	I	**MMC Card Select:** Host to card chip select signal
MCMD	I/O	**MMC Command:** Host to/from card Command/Response signal
MCLK	I	**MMC Serial Clock:** Host to Card clock signal
MDATA	I/O	**MMC Data:** Host to/from card data signal
Clock & Reset		
nRESET	I	**External System Reset:** This signal suspends any operation in progress.
RSTOUT	O	**Internal POR Output:** This pin is used to check the Power-On Reset detect voltage.
EXTCLK	I	**External Clock Source:** External clock input for test mode
CLKOUT	O	**VCO Output:** This pin is used to check the Internal VCO Characteristics
Special Interface		
OM[3:0]	I	These pins determine status of product for manufacture test. 0000 = Normal operation mode
HVMMC	I	This pin is used to select high voltage MMC
GPIO [7:0]	IO	**General IO port**
XADDR[6:0]	O	**External ROM Address:** These pins are used to check manufacture test.
XDATA	I/O	**External ROM data:** These pins are used to check manufacture test.

Table 3. S3F49DAX Signal Description (Continuous)

Signal	I/O	Description
NOR Flash Interface		
SCLK	I	**Serial CLOCK:** This pin is used Tool program for internal Flash (Write speed: Max 250 kHz, Read speed: Max 3 MHz)
SDAT	IO	**Serial DATA:** This pin is used Tool program for internal Flash (Output when reading, Input when writing.) Input & push-pull output port can be assigned.
VPP	P	Flash cell writing power supply pin for tool program mode. The function of entering flash writing Mode.
Margin	I	This pin can used the flash cell margin test
JTAG Interface		
nTRST	I	This pin (TAP Controller Reset) can reset the TAP controller at power-up. If the debugger is not used, this pin should be "Low" level or low active pulse should be applied before CPU running. For example, nRESET signal can be tied with this pin
TMS	I	This pin (TAP Controller Mode Select) can control the sequence of the state diagram of TAP controller
TCK	I	This pin (TAP Controller Clock) can provide the clock input for the JTAG logic
RTCK	O	This pin is the returned TCK
TDI	I	This pin (TAP Controller Data Input) is the serial input for JTAG port.
TDO	O	This Pin (TAP Controller Data Output) is the serial data output for JTAG port.
Power		
IVCOUT	P	**IVC Out:** This pin output the internal IVC Output voltage (1.8 V)
VDD	P	Interface power supply
VSS	P	Interface ground

1. MULTIMEDIACARD FUNCTION DESCTION

The MultiMediaCard system defines two communication protocols: MMC and SPI mode. All communication between host and card is controlled by the host. The card automatically selects the mode of the reset command and will operate all command to be in the same communication mode.

MMC MODE

The basic MultiMediaCard concept is based on transferring data via a minimal number of signals. The communication signals are:

- **CLK:** with each cycle of this signal an one bit transfer on the command and data lines is done. The frequency may vary between zero and the maximum clock frequency.
- **CMD:** is a bi-directional command channel used for card initialization and data transfer commands. The CMD signal has two operation modes: open-drain for initialization mode and push-pull for fast command transfer. Commands are sent from the MultiMediaCard bus master to the card and responses from the cards to the host.
- **DAT:** is a bi-directional data channel. The DAT signal operates in push-pull mode. Only one card or the host is driving this signal at a time.

Table 4. MMC/SPI Interface Pin Configuration

Pin	MultiMediaCard Mode			SPI Mode		
No	Name	Type1	Description	Name	Type	Description
1	RSV	NC	Reserved	CS	I	Chip select (Active Low)
2	CMD	I/O/PP/OD	Command/Response	DI	I/PP	Data In
3	VSS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS2	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT	I/O/PP	Data	DO	O/PP	Data Out

NOTE: S: Power supply; I: Input; O: Output; PP: Push-pull; OD: Open-drain; NC: Not connected (or Logical high)

SPI MODE

The SPI mode consists of a secondary, optional communication protocol which is offered by Flash-based MultiMediaCards. This mode is a subset of the MultiMediaCard protocol, designed to communicate with a SPI channel, commonly found in Motorola's (and lately a few other vendors') microcontrollers. The interface is selected during the first reset command after power up (CMD0) and cannot be changed once the part is powered on.

The SPI standard defines the physical link only, and not the complete data transfer protocol. The MultiMediaCard SPI implementation uses a subset of the MultiMediaCard protocol and command set. It is intended to be used by systems which require a small number of cards (typically one) and have lower data transfer rates (compared to MultiMediaCard protocol based systems). From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to minimum.

The MultiMediaCard SPI interface is compatible with SPI hosts available on the market. As in any other SPI device, the MultiMediaCard SPI channel consists of the following four signals:

- **CS:** Host to card Chip Select signal.
- **CLK:** Host to card clock signal
- **DataIn:** Host to card data signal.
- **DataOut:** Card to host data signal.

MULTIMEDIACARD PROTOCOL

MMC MODE

After a power-on reset, the card is initialized by MultiMediaCard bus protocol using the command channel. MultiMediaCard bus protocol is represented by one of the following tokens:

- **Command:** A command is a token, which starts an operation. A command is sent from the host either to a single card (addressed command) or to all connected cards (broadcast command). A command is transferred serially on the CMD line.
- **Response:** A response is a token, which is sent from an addressed card, or (synchronously) from all connected cards, to the host as an answer to a previously received command. A response is transferred serially on the CMD line.
- **Data:** Data can be transferred from the card to the host or vice versa. Data is transferred via the data line.

Card addressing is implemented using a session address assigned during the initialization phase, by the bus controller to all currently connected cards. Individual cards are identified by their CID number. This method requires that every card will have an unique CID number. To ensure uniqueness of CIDs the CID register contains 24 bits (MID and OID fields - see Chapter 5), which are defined by the MMCA. Every card manufacturers is required to apply for an unique MID (and optionally OID) number.

MultiMediaCard bus data transfers are composed of these tokens. One data transfer is a bus operation. There are different types of operations. Addressed operations always contain a command and a response token. In addition, some operations have a data token, the others transfer their information directly within the command or response structure. In this case no data token is present in an operation. The bits on the DAT and the CMD lines are transferred synchronous to the host clock.

Two types of data transfer commands are defined:

- **Sequential commands:** These commands initiate a continuous data stream, they are terminated only when a stop command follows on the CMD line. This mode reduces the command overhead to an absolute minimum.
- **Block-oriented commands:** These commands send a data block succeeded by CRC bits. Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the CMD line similarly to the sequential read.

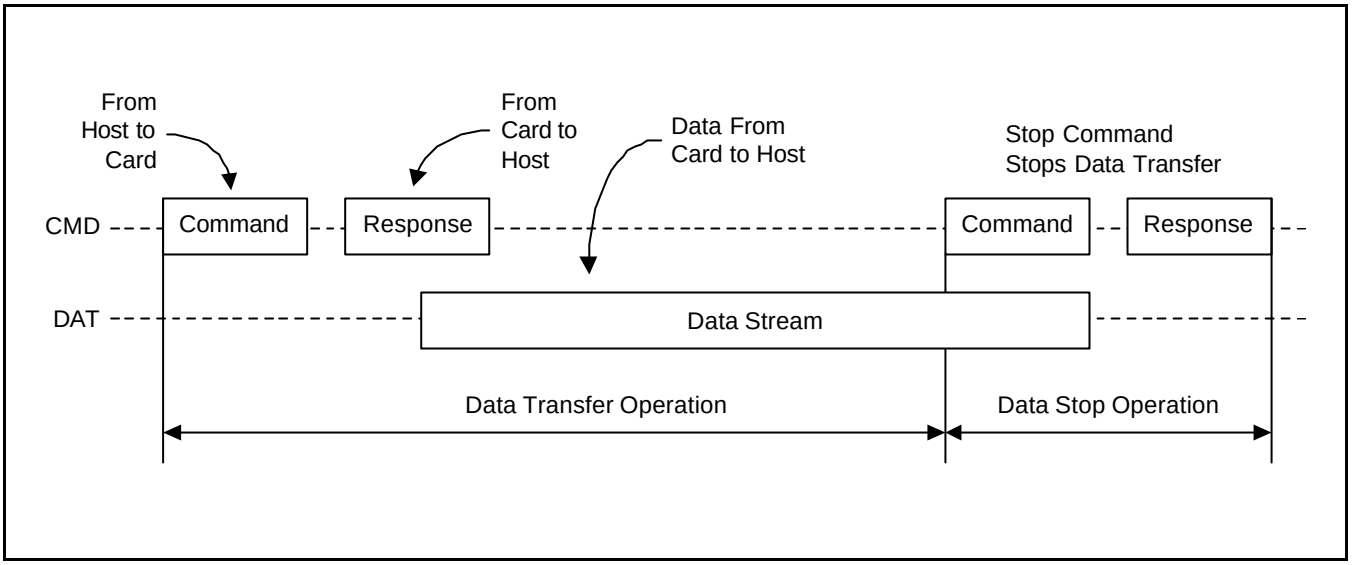

Figure 3. Sequential Read Operations

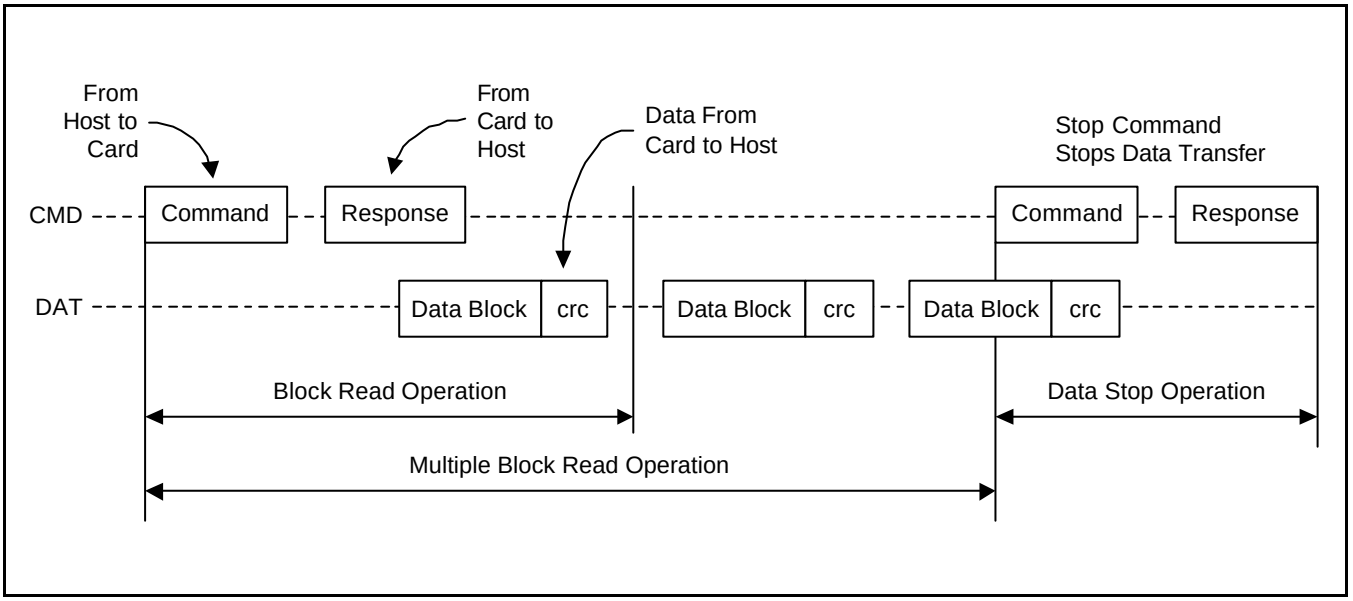

Figure 4. (Multiple) Block Read Operations

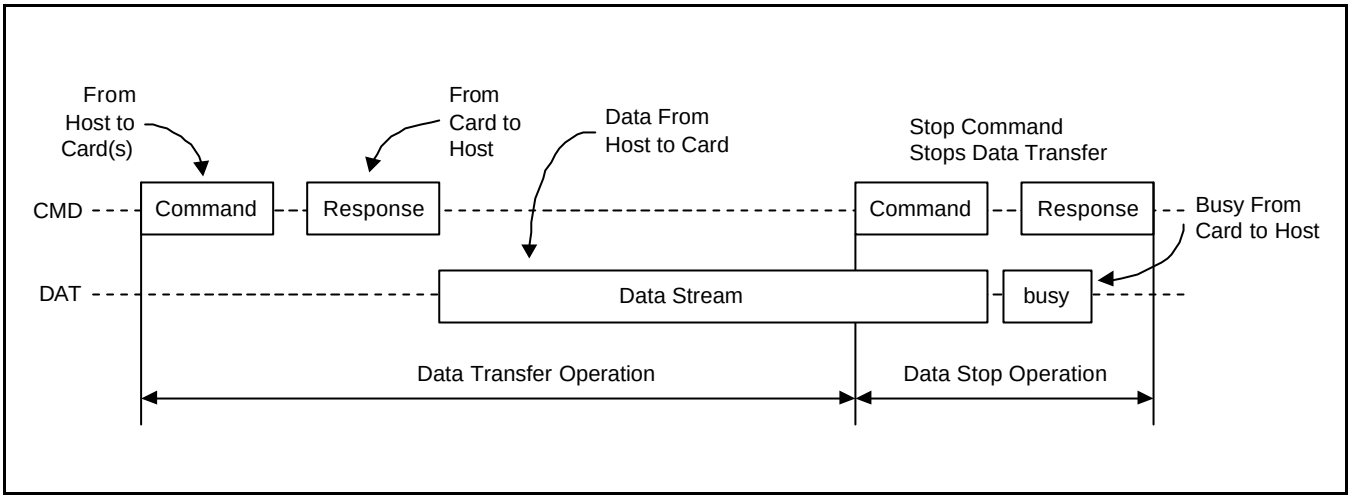

Figure 5. Sequential Write Operations

The block write operation uses a simple busy signalling of the write operation duration on the data (DAT) line. (see Figure6).

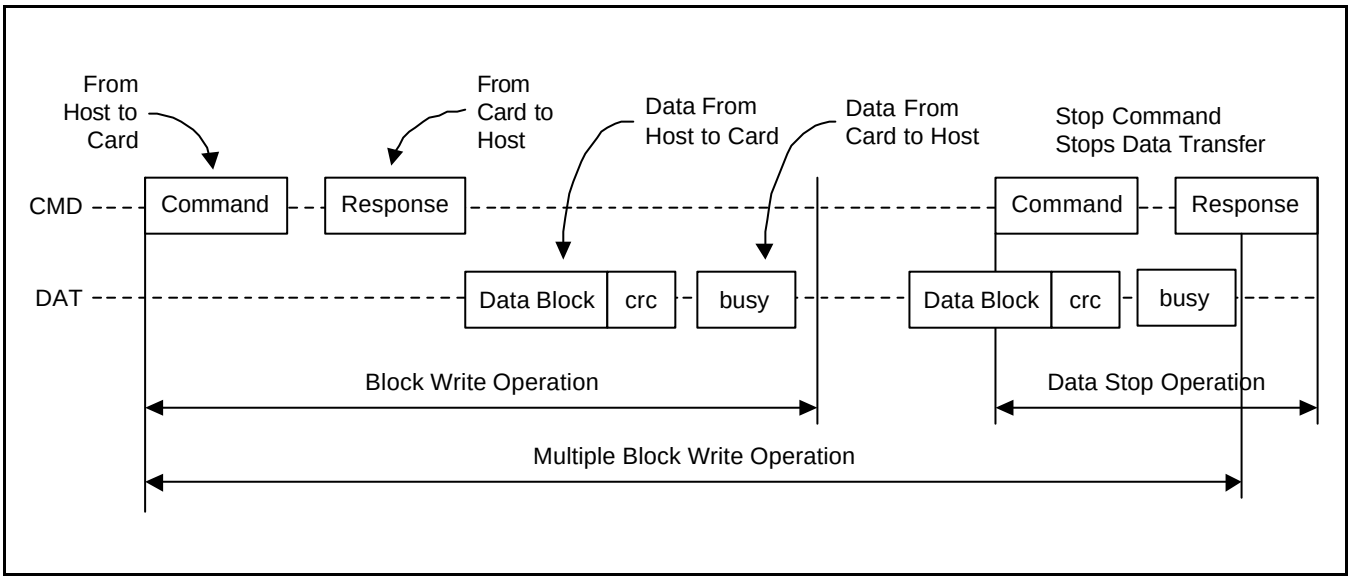

Figure 6. (Multiple) Block Write Operations

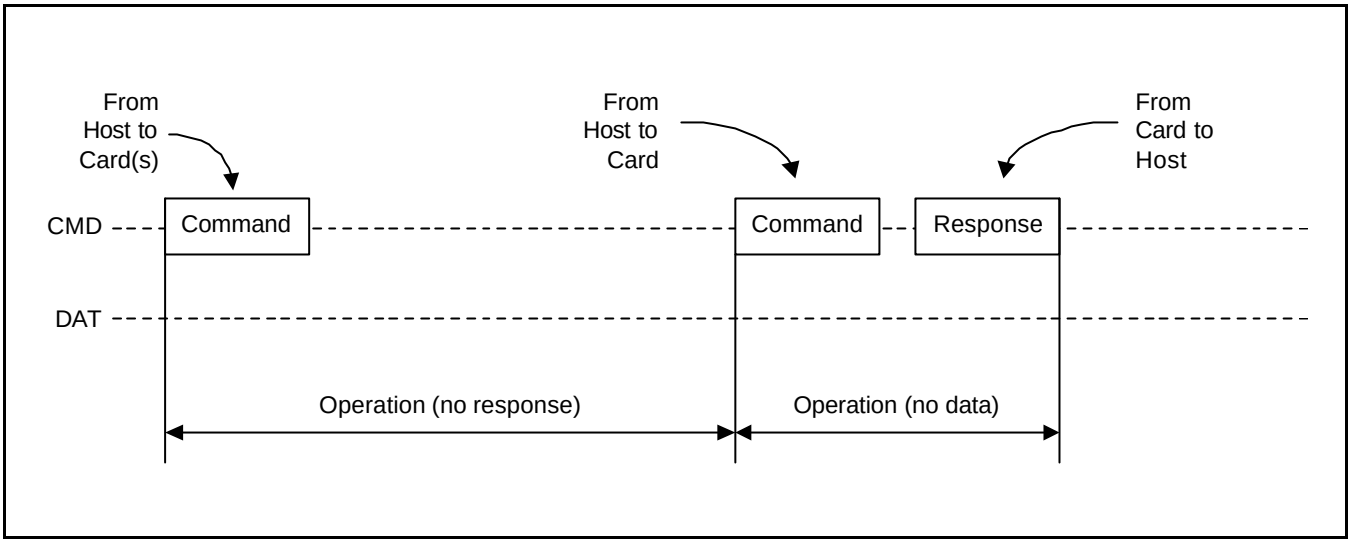

Figure 7. “No Response” and “No Data” Operations

Command tokens have the following coding scheme:

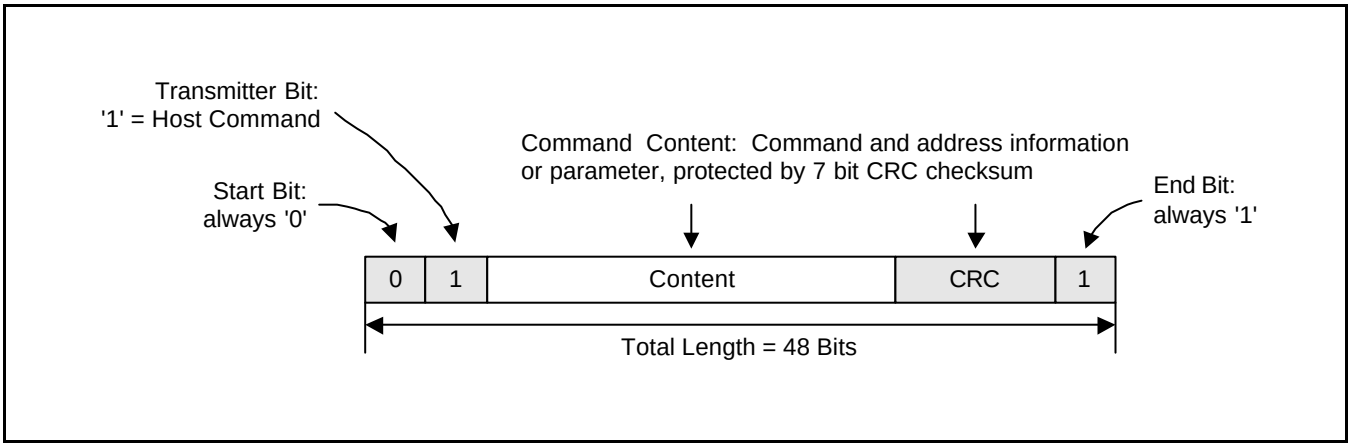

Figure 8. Command Token Format

Each command token is preceded by a start bit (‘0’) and succeeded by an end bit (‘1’). The total length is 48 bits. Each token is protected by CRC bits so that transmission errors can be detected and the operation may be repeated.

Response tokens have five coding schemes depending on their content. The token length is either 48 or 136 bits. The detailed commands and response definition is given in next sections. The CRC protection algorithm for block data is a 16 bits CCITT polynomial.

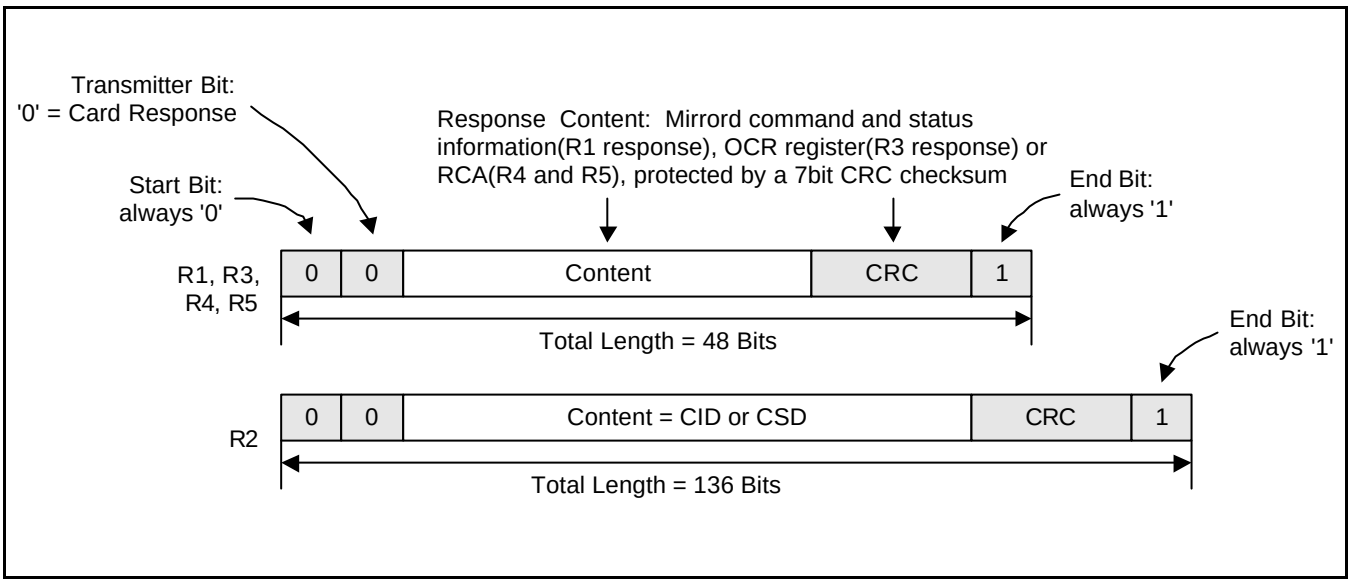

Figure 9. Response Token Format

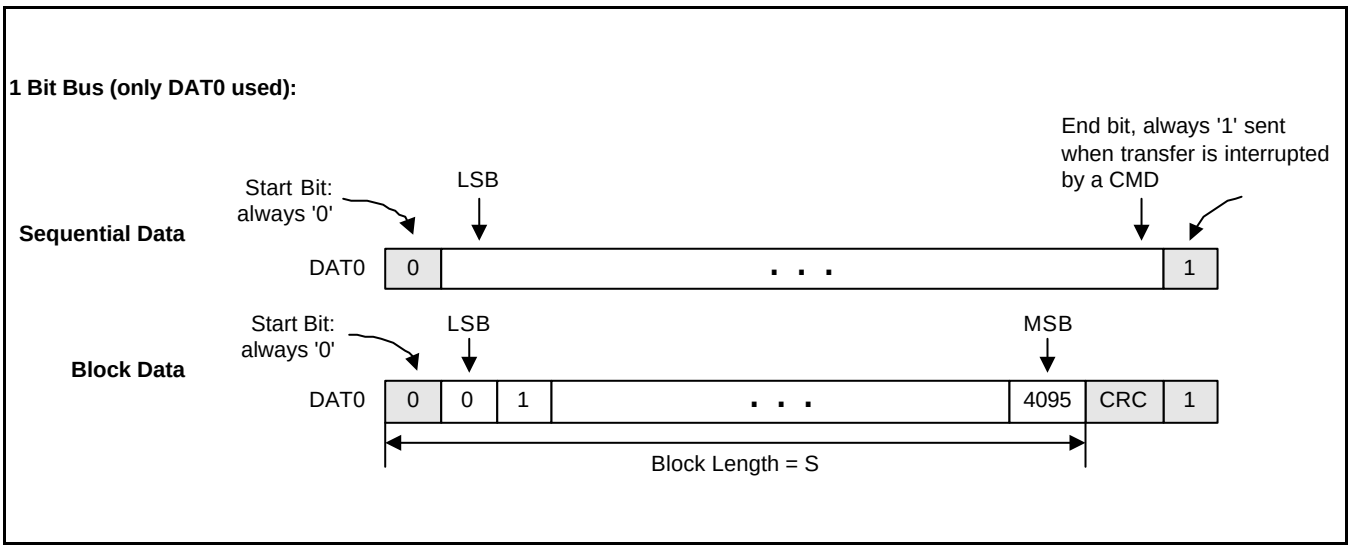

Figure 10. Data Token Format

SPI MODE

While the MultiMediaCard channel is based on command and data bit streams, which are initiated by a start bit and terminated by a stop bit, the SPI channel is byte oriented. Every command or data block is built of 8-bit bytes and is byte aligned to the CS signal (i.e. the length is a multiple of 8 clock cycles).

Similar to the MultiMediaCard protocol, the SPI messages consist of command, response and data-block tokens (see Chapter 3 for a detailed description). All communication between host and cards is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low.

The response behavior in the SPI mode differs from the MultiMediaCard mode in the following three aspects:

- The selected card always responds to the command.
- Additional (8, 16 & 40 bit) response structures are used
- When the card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than by a time-out, as in the MultiMediaCard mode.

Only single and multiple block read/write operations are supported in SPI mode (sequential mode is not supported). In addition to the command response, every data block sent to the card during write operations will be responded to with a special data response token. A data block may be as big as one card write block and as small as a single byte. Partial block read/write operations are enabled by card options specified in the CSD register.

Data Read

The SPI mode supports single and multiple block read operations. The main difference between SPI and MultiMediaCard modes is that the data and the response are both transmitted to the host on the DataOut signal (refer to Figure11 and Figure12). Therefore the card response to the STOP_COMMAND may cut-short and replace the last data block.

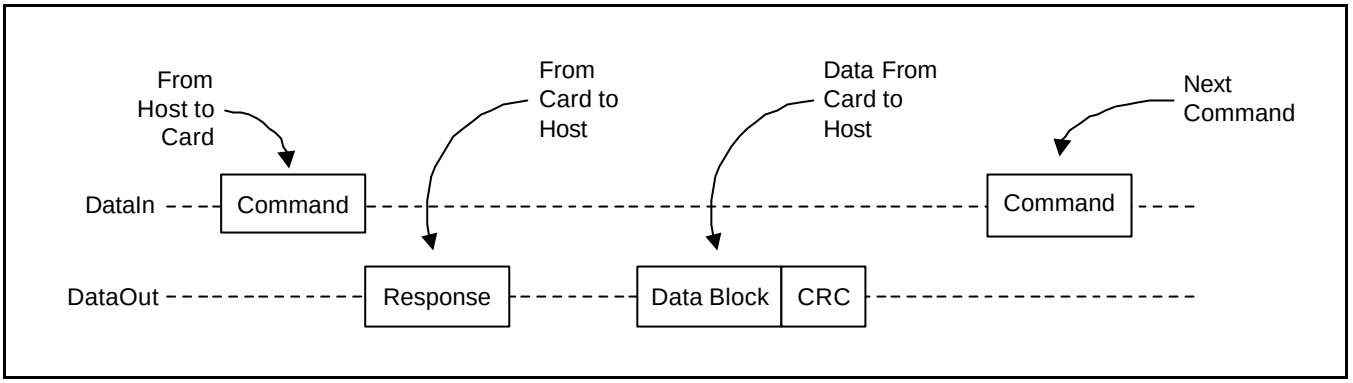

Figure 11. Single Block Read Operation

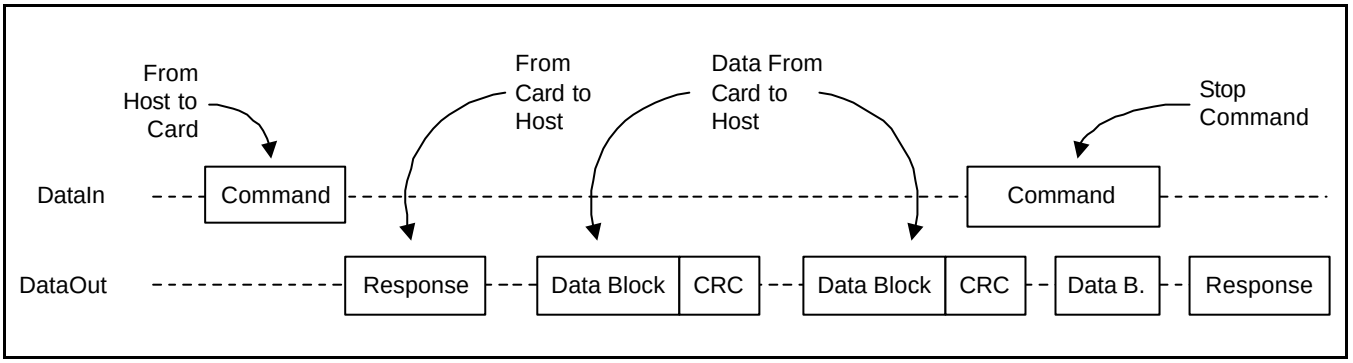

Figure 12. Multiple Block Read Operation

In case of a data retrieval error (e.g. out of range, address misalignment, internal error, etc.), the card will not transmit any data. Instead (as opposed to MultiMediaCard mode where the card times out), a special data error token will be sent to the host. Figure13 shows a single block read operation, which terminates with an error token rather than a data block.

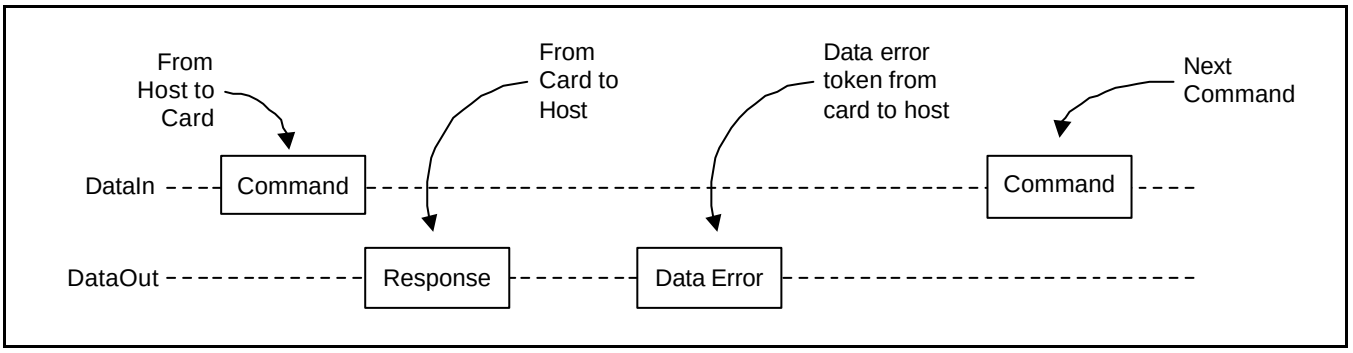

Figure 13. Read Operation – Data Error

Multiple block read operation can be terminated the same way, the error token replacing a data block anywhere in the sequence. The host must than abort the operation by sending the stop transmission command.

If the host sends a stop transmission command after the card transmitted the last block of a multiple block read with a pre-defined number of blocks, it will be responded to as an illegal command.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment error condition at the beginning of the first misaligned block (ADDRESS_ERROR error bit will is set in the data error token).

Data Write

The SPI mode supports single block and Multiple block write commands. Upon reception of a valid write command (CMD24 or CMD25), the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix, block length and start address restrictions are (with the exception of the CSD parameter WRITE_BL_PARTIAL controlling the partial block write option) identical to the read operation (see Figure14). If a CRC error is detected it will be reported in the data-response token and the data block will not be programmed.

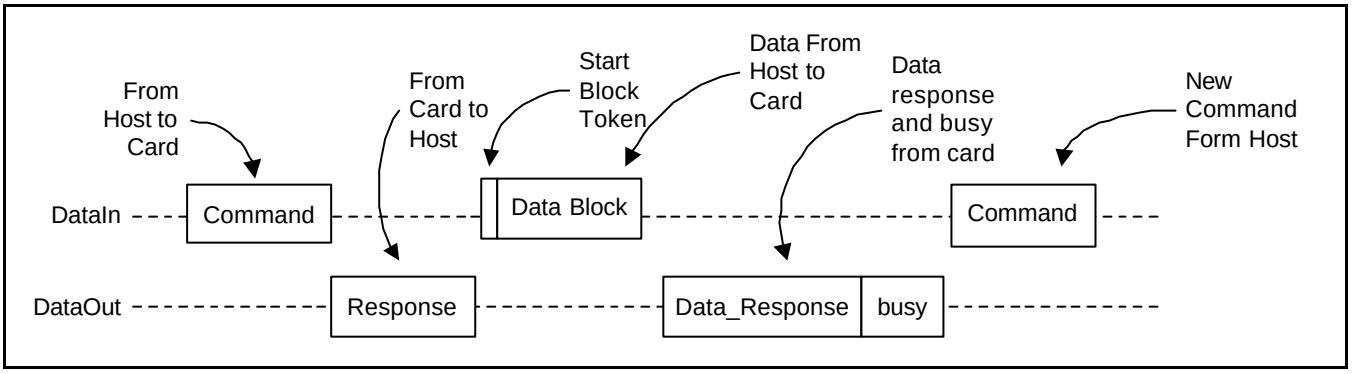

Figure 14. Single Block Write Operation

Every data block has a prefix of ‘Start Block’ token (one byte).

After a data block has been received, the card will respond with a data-response token. If the data block has been received without errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the DataOut line low).

In Multiple Block write operation the stop transmission will be done by sending ‘Stop Tran’ token instead of ‘Start Block’ token at the beginning of the next block.

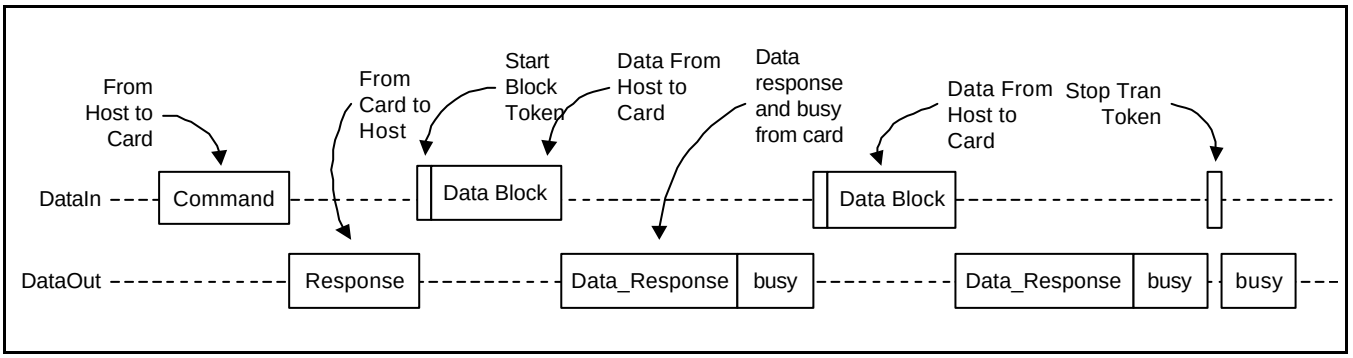

Figure 15. Multiple Block Write Operation

COMMANDS

Command Types

There are four kinds of commands defined to control the MultiMediaCard:

- Broadcast commands (bc), no response.
- Broadcast commands with response (bcr) response from all cards simultaneously.
- Addressed (point-to-point) commands (ac) no data transfer on DAT.
- Addressed (point-to-point) data transfer commands (adtc) data transfer on DAT

All commands and responses are sent over the CMD line of the MultiMediaCard bus. The command transmission always starts with the left bit of the bit-string corresponding to the command codeword.

Command Format

All commands have a fixed code length of 48 bits, needing a transmission time of 2.4usec @ 20 MHz

Bit Position	47	46	[45:40]	[39:8]	[7:1]	[0]
Width (bits)	1	1	6	32	7	1
Value	“0”	“1”	“x”	“x”	“x”	“1”
Description	Start bit	Transmission bit	Command index	Argument	CRC7	End bit

A command always starts with a start bit (always ‘0’), followed by the bit indicating the direction of transmission (host = ‘1’). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by ‘x’ in the table above indicates this variable is dependent on the command. All commands are protected by a CRC (see Chapter 8.2 for the definition of CRC7). Every command codeword is terminated by the end bit (always ‘1’). All commands and their arguments are listed in Table 5, Table 13.

Detailed Command Description

The following tables define in detail all MultiMediaCard bus commands. The responses R1-R5 are defined in Response section. The registers CID, CSD and DSR are described in Register section.

Table 5. Basic Commands (class 0) and Read Stream Commands (class 1)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD0	bc	[31:0] stuff bits	–	GO_IDLE_STATE	Resets all cards to idle state
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3	ac	[31:16] RCA [15:0] stuff bits	R1	SET_RELATIVE_ ADDR	Assigns relative address to the card
CMD4	bc	[31:16] DSR [15:0] stuff bits	–	SET_DSR	Programs the DSR of all cards
CMD5	Reserved				
CMD6	Reserved				
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/DISELECT_ CARD	Command toggles a card between the standby and transfer states or between the programming and disconnect states. In both cases the card is selected by its own relative address and gets deselected by any other address; address 0 deselects all.
CMD8	Reserved				
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification data (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DATA_UNTIL _STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1 or R1b	STOP_ TRANSMISSION	Terminates a read/write stream/multiple block operation. When CMD12 is used to terminate a read transaction the card will respond with R1. when it is used to stop a write transaction the card will respond with R1b.
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] stuff bits	–	GO_INACTIVE_ STATE	Set the card to inactive state in order to protect the card stack against communication breakdowns

Table 6. Block Oriented Read Commands (class 2)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD16	ac	[31:0] block address	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read/write). Default block length is specified in the CSD
CMD17	adtc	[31:0] block address	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] block address	R1	READ_MULTIPLE_ BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command or the requested number of data block transmitted.
CMD19	Reserved				

Table 7. Stream Write Commands (class 3)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD20	adtc	[31:0] block address	R1	WRITE_DAT_ UNTIL_STOP	Writes data stream from the host, starting at the given address. Until a STOP_SRANSMISSION follows.
CMD21	Reserved				
CMD22	Reserved				

Table 8. Block Oriented Write Commands (class 4)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD23	ac	[31:16] set to 0 [15:0] number of block	R1	SET_BLOCK_ COUNT	Defines the number of blocks which are going to be transfer in the immediately succeeding multiple block read or write commands.
CMD24	adtc	[31:0] block address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] block address	R1	WRITE_MULTIPLE _BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows or the requested number of block received.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

Table 9. Block Oriented Write Protection Commands (class 6)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD28	ac	[31:0] block address	R1	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] block address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_ PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits
CMD31	Reserved				

Table 10. Erase Commands (class 5)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD32	Reserved: These command indexes cannot be used in order to maintain backwards compatibility with older versions of the MultiMediaCards.				
CMD33					
CMD34					
CMD35	ac	[31:0] block address	R1	ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] block address	R1	ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase
CMD37	Reserved: These command indexes cannot be used in order to maintain backwards compatibility with older versions of the MultiMediaCards.				
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erases all previously selected write blocks

Table 11. I/O Mode Commands (class 9, S3F49DAX cannot support the this command class)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD39	ac	[31:16] RCA [15] Register write flag [14:8] Register address [7:0] Register data	R4	FAST_IO	Used to write and read 8 bit (register) data fields. The command addresses a card and a register and provides the data for writing if the write flag is set. The R4 response contains data read from the addressed register. This command accesses application dependent registers which are not defined in the MultiMediaCard standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				

Table 12. Lock Card (class 7)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/ unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43 – CMD54	Reserved				

Table 13. Application Specific Commands (class 8, S3F49DAX cannot support the this command class)

CMD Index	Type	Argument	Resp.	Abbreviation	Command Description
CMD55	ac	[31:16] RCA [15:0]stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command
CMD56	adtc	[31:1] stuff bits [0] RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose/application specific commands. The size of the data block shall be set by the SEL_BLOCK_LEN command.
CMD57	Reserved				
CMD58					
CMD59					
CMD60	Reserved for manufacture				
CMD61					
CMD62					
CMD63					

RESPONSES

All responses are sent via the command line CMD. The response transmission always starts with the left bit of the bit-string corresponding to the response codeword. The code length depends on the response type.

A response always starts with a start bit (always '0'), followed by the bit indicating the direction of transmission (card = '0'). A value denoted by 'x' in the tables below indicates a variable entry. All responses except for the type R3 (see below) are protected by a CRC. Every command codeword is terminated by the end bit (always '1').

There are five types of responses. Their formats are defined as follows:

- **R1** (normal response command): code length 48 bit. The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits. The card status is described in next section.

Bit Position	47	46	[45:40]	[39:8]	[7:1]	[0]
Width (bits)	1	1	6	32	7	1
Value	"0"	"0"	"x"	"x"	"x"	"1"
Description	Start bit	Transmission bit	Command index	Argument	CRC7	End bit

- **R1b** is identical to R1 with an optional busy signal transmitted on the data line. The card may become busy after receiving these commands based on its state prior to the command reception.
- **R2** (CID, CSD register): code length 136 bits. The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127...1] of the CID and CSD are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response.

Bit Position	135	134	[133:128]	[127:1]	[0]
Width (bits)	1	1	6	127	1
Value	"0"	"0"	"111111"	"x"	"1"
Description	Start bit	Transmission bit	Reserved	CID or CSD register incl. Internal CRC7	End bit

- **R3** (OCR register): code length 48 bits. The contents of the OCR register is sent as a response to CMD1. The level coding is as follows: restricted voltage windows = LOW, card busy = LOW.

Bit Position	47	46	[45:40]	[39:8]	[7:1]	[0]
Width (bits)	1	1	6	32	7	1
Value	"0"	"0"	"111111"	"x"	"1111111"	"1"
Description	Start bit	Transmission bit	Reserved	OCR Register	Reserved	End bit

- **R4** (Fast I/O): code length 48 bits. The argument field contains the RCA of the addressed card, the register address to be read out or written to, and its contents.

Bit Position	47	46	[45:40]	[39:8] Argument field			[7:1]	[0]
Width (bits)	1	1	6	16	8	8	7	1
Value	“0”	“1”	“100111”	“x”	“x”	“x”	“x”	“1”
Description	Start bit	Transmission bit	CMD39	RCA[31:16]	Register Address	Read Register Contents [7:0]	CRC7	End bit

- **R5** (Interrupt request): code length 48 bits. If the response is generated by the host, the RCA field in the argument shall be 0x0.

Bit Position	47	46	[45:40]	[39:8] Argument field		[7:1]	[0]
Width (bits)	1	1	6	16	16	7	1
Value	“0”	“1”	“100111”	“x”	“x”	“x”	“1”
Description	Start bit	Transmission bit	CMD39	RCA [31:16] of winning card or of the host	[15:0] Not defined. May be used for IRQ data	CRC7	End bit

CARD STATUS

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card’s status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command. The semantics of this register is according to the CSD entry SPEC_VERS (see Chapter 5.3), indicating the version of the response formats (possibly used for later extensions).

Table 16 defines the different entries of the status. The type and clear condition fields in the table are abbreviated as follows:

Types:

- E: Error bit.
- S: Status bit.
- R: Detected and set for the actual command response.
- X: Detected and set during command execution. The host must poll the card by issuing the status command in order to read these bits.

Card Condition:

- A: According to the card current state.
- B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).
- C: Clear by read

Table 14. Card Status

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	ER	0 = No error 1 = Error	The command's argument was out of the allowed range for this card	C
30	ADDRESS_ERROR	ERX	0 = No error 1 = Error	A misaligned address, which did not match the block length, was used in the command.	C
29	BLOCK_LEN_ERROR	ER	0 = No error 1 = Error	The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.	C
28	ERASE_SEQ_ERROR	ER	0 = No error 1 = Error	An error in the sequence of erase commands occurred.	C
27	ERASE_PARA	EX	0 = No error 1 = Error	An invalid selection of erase groups for erase occurred.	C
26	WP_VIOLATION	ERX	0 = Not protected 1 = Protected	Attempt to program a write protected block.	C
25	CARD_IS_LOCKED	SX	0 = Card unlocked 1 = Card locked	When set, signals that the card is locked by the host	A
24	LOCK_UNLOCK_ FAILED	ERX	0 = No error 1 = Error	Set when a sequence or password error has been detected in lock/ unlock card command or if there was an attempt to access a locked card	C
23	COM_CRC_ERROR	ER	0 = No error 1 = Error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	ER	0 = No error 1 = Error	Command not legal for the card state	B
21	CARD_ECC_FAILED	EX	0 = No error 1 = Error	Card initial ECC was applied but failed to correct the data	C
20	CC_ERROR	ERX	0 = No error 1 = Error	Internal card controller error	C
19	ERROR	ERX	0 = No error 1 = Error	A general or an unknown error occurred during the operation.	C
18	UNDERRUN	EX	0 = No error 1 = Error	The card could not sustain data transfer in stream read mode.	C
17	OVERRUN	EX	0 = No error 1 = Error	The card could not sustain programming in stream write mode.	C

Table 14. Card Status

Bits	Identifier	Type	Value	Description	Clear Condition
16	CID/CSD_ OVERWRITE	ERX	0 = No error 1 = Error	Card be either one of the following error: - The CID register has been already written and cannot be overwritten. - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.	C
15	WP_ERASE_SKIP	SX	0 = Not protected 1 = Protected	Only partial address space was erased due to existing write protected blocks.	C
14	CARD_ECC_ DISABLED	SX	0 = Enabled 1 = Disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	SR	0 = Cleared 1 = Set	An erase sequence was cleared before executing because an out of erase sequence command was received.	C
12:9	CURRENT_STATE	SX	0 = idle 1 = ready 2 = ident 3 = standby 4 = tran 5 = data 6 = rcv 7 = prg 8 = dis 9-15 = reserved	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15	B
8	READY_FOR_DAT A	SX	0 = Not ready 1 = Ready	Corresponds to buffer empty signaling on the bus	A
7:6	Reserved				
5	APP_CMD	SR	0 = Disabled 1 = Enabled	The card will expect ACMD, or indication that the command has been interpreted as ACMD	C
4	Reserved				
3, 2	Reserved for application specific commands				
1, 0	Reserved for manufacture test mode				

2. CARD REGISTERS

Within the card interface five registers are defined: OCR, CID, CSD, RCA and DSR. These can be accessed only by corresponding commands (above section). The OCR, CID and CSD registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters.

OCR Register (Operation Condition Register)

The 32-bit operation conditions register stores the VDD voltage profile of the card. In addition, this register includes a status information bit. This status bit is set if the card power up procedure has been finished. The register is optional and can be read only. The OCR register shall be implemented by the cards, which do not support the full operating voltage range of the MultiMediaCard bus. The operation voltage range of S3F49DAX is 2.7V to 3.6V, and the initial value of OCR is “0x00FF8000”.

The supported voltage range is coded as shown in Table 15. A voltage range is not supported if the corresponding bit value is set to LOW. As long as the card is busy, the corresponding bit (31) is set to LOW.

Table 15. OCR Register Definitions

OCR Bit	Supply Voltage Range Window	OCR Bit	Supply Voltage Range Window
Bit [6:0]	Reserved	Bit 16	2.8 – 2.9
Bit 7	1.65 – 1.95	Bit 17	2.9 – 3.0
Bit 8	2.0 – 2.1	Bit 18	3.0 – 3.1
Bit 9	2.1 – 2.2	Bit 19	3.1 – 3.2
Bit 10	2.2 – 2.3	Bit 20	3.2 – 3.3
Bit 11	2.3 – 2.4	Bit 21	3.3 – 3.4
Bit 12	2.4 – 2.5	Bit 22	3.4 – 3.5
Bit 13	2.5 – 2.6	Bit 23	3.5 – 3.6
Bit 14	2.6 – 2.7	Bit [30:24]	Reserved
Bit 15	2.7 – 2.8	Bit 31	Card power up status bit (busy) (NOTE)

NOTE: This bit is set to LOW if the card has not finished the power up routine.

CID Register (Card Identification Register)

The Card IDentification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase (MultiMediaCard protocol). Every individual flash or I/O card shall have an unique identification number. Every type of MultiMediaCard ROM cards (defined by content) shall have an unique identification number.

The structure of the CID register is defined in the following paragraphs:

- **MID:** An 8 bit binary number that identifies the card manufacturer. The MID number is controlled, defined and allocated to a MultiMediaCard manufacturer by the MMCA. This procedure is established to ensure uniqueness of the CID register.
- **OID:** A 16 bit binary number that identifies the card OEM and/or the card contents (when used as a distribution media either on ROM or FLASH cards). The OID number is controlled, defined and allocated to a MultiMediaCard manufacturer by the MMCA. This procedure is established to ensure uniqueness of the CID register.
- **PNM:** The product name is a string, 6 ASCII characters long.
- **PRV:** The product revision is composed of two Binary Coded Decimal (BCD) digits, four bits each, representing an “n.m” revision number. The “n” is the most significant nibble and “m” is the least significant nibble. As an example, the PRV binary value field for product revision “6.2” will be: 0110 0010
- **PSN:** A 32 bits unsigned binary integer.
- **MDT:** The manufacturing date is composed of two hexadecimal digits, four bits each, representing a two digits date code m/y;
  The “m” field, most significant nibble, is the month code. 1 = January.
  The “y” field, least significant nibble, is the year code. 0 = 1997.
  As an example, the binary value of the MDT field for production date “April 2000” will be: 0100 0011
- **CRC:** CRC7 checksum (7 bits).

Table 16. CID Register

Name	Field	Width	CID-Slice
Manufacturer ID	MID	8	[127:120]
OEM/Application ID	OID	16	[119:104]
Product Name	PNM	48	[103:56]
Product Revision	PRV	8	[55:48]
Product Serial Number	PSN	32	[47:16]
Manufacturing Date	MDT	8	[15:8]
CRC7	CRC	7	[7:1]
Not used, always “1”	–	1	[0]

CSD Register (Card Specific Data Register)

The Card-Specific Data register provides information on how to access the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used etc. The programmable part of the register (entries marked by W or E, see below) can be changed by CMD27. The type of the entries in the table below is coded as follows: R = readable, W = writable once, E = erasable (multiple writable).

Table 17. The CSD Field

Name	Field	Width	Cell Type	CSD-Slice
CSD Structure	CSD_STRUCTURE	2	R	[127:126]
System Specification Version	SPEC_VER	4	R	[125:122]
Reserved	–	2	R	[121:120]
Data Read Access Time 1	TAAC	8	R	[119:112]
Data Read Access Timer 2 in CLK cycles	NSAC	8	R	[111:104]
Max. Data Transfer Rate	TRAN_SPEED	8	R	[103:96]
Card Command Classes	CCC	12	R	[95:84]
Max. Read Data Block Length	READ_BL_LEN	4	R	[83:80]
Partial Blocks For Read Allowed	READ_BL_PARTIAL	1	R	[79]
Write Block Misalignment	WRITE_BLK_MISALIGN	1	R	[78]
Read Block Misalignment	READ_BLK_MISALIGN	1	R	[77]
DSR implemented	DSR_IMP	1	R	[76]
Reserved	–	2	R	[75:74]
Device Size	C_SIZE	12	R	[73:62]
Max. Read Current @ VDD min	VDD_R_CURR_MIN	3	R	[61:59]
Max. Read Current @ VDD max	VDD_R_CURR_MAX	3	R	[58:56]
Max. Write Current @ VDD min	VDD_W_CURR_MIN	3	R	[55:53]
Max. Write Current @ VDD max	VDD_W_CURR_MAX	3	R	[52:50]
Device Size Multiplier	C_SIZE_MULT	3	R	[49:47]
Erase Sector Size	SECTOR_SIZE	5	R	[46:42]
Erase Group Size	ERASE_GRP_SIZE	5	R	[41:37]
Write Protect Group Size	WP_GRP_SIZE	5	R	[36:32]
Write Protect Group Enable	WP_GRP_ENABLE	1	R	[31]
Manufacturer Default ECC	DEFAULT_ECC	2	R	[30:29]
Write Speed Factor	R2W_FACTOR	3	R	[28:26]
Max. Write Data Block Length	WRITE_BL_LEN	4	R	[25:22]
Partial Blocks for Write Allowed	WRITE_BL_PARTIAL	1	R	[21]
Reserved	–	5	R	[20:16]

Table 17. The CSD Field (Continuous)

Name	Field	Width	Cell Type	CSD-Slice
File Format Group	FILE_FORMAT_GRP	1	R/W	[15]
Copy Flag (OTP)	COPY	1	R/W	[14]
Permanent Write Protection	PERM_WRITE_PROTECT	1	R/W	[13]
Temporary Write Protection	TMP_WRITE_PROTECT	1	R/W	[12]
File Format	FILE_FORMAT	2	R/W	[11:10]
ECC Code	ECC	2	R/W/E	[9:8]
CRC	CRC	7	R/W/E	[7:1]
Not used, always “1”	–	1	–	[0]

The following sections describe the CSD fields and the relevant data types. If not explicitly defined otherwise, all bit-strings are interpreted as binary coded numbers starting with the left bit first.

· **CSD_STRUCTURE:** Describes the version number of the CSD structure.

CSD_STRUCTURE	CSD Structure Version	Valid for System Specification Version
0	CSD version No. 1.0	Version 1.0 – 1.2
1	CSD version No. 1.1	Version 1.4 – 2.2
2	CSD version No. 1.2	Version 3.1
3	Reserved	Reserved

· **SPEC_VER:** Defines the MultiMediaCard System Specification version supported by the card.

SPEC_VER	System Specification Version Number
0	Version 1.0 – 1.2
1	Version 1.4
2	Version 2.0 – 2.2
3	Version 3.1
4–15	Reserved

· **TAAC:** Defines the asynchronous part of the data access time.

TAAC	Description
2:0	Time Unit: 000 = 1nsec 001 = 10nsec 010 = 100nsec 011 = 1usec, 100 = 10usec 101 = 100usec 110 = 1msec 111 = 10msec
6:3	Time Value: 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
7	Reserved

- **NSAC:** Defines the typical case for the clock dependent factor of the data access time. The unit for NSAC is 100 clock cycles. Therefore, the maximal value for the clock dependent part of the data access time is 25.5k clock cycles. The total access time N_{AC} is calculated based on TAAC and NSAC. It has to be computed by the host for the actual clock rate. The read access time should be interpreted as a typical delay for the first data bit of a data block or stream.

- **TRAN_SPEED:** The following table defines the maximum data transfer rate TRAN_SPEED

TRAN_SPEED	Description
2:0	Transfer Rate Unit: 000 = 100Kbit/s 001 = 1Mbit/s 010 = 10Mbit/s 011 = 10Mbit/s Other = Reserved
6:3	Time Value: 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
7	Reserved

- **CCC:** The MultiMediaCard command set is divided into subsets (command classes). The card command class register CCC defines which command classes are supported by this card. A value of '1' in a CCC bit means that the corresponding command class is supported.

Bit Position	Description	Bit Position	Description	Bit Position	Description
0	Class 0	4	Class 4	8	Class 8
1	Class 1	5	Class 5	9	Class 9
2	Class 2	6	Class 6	10	Class 10
3	Class 3	7	Class 7	11	Class 11

- **READ_BL_LEN:** The data block length is computed as 2READ_BL_LEN. The block length might therefore be in the range 1,2,4...2048 bytes.

Bit Value	Description	Bit Value	Description	Bit Value	Description
0	2^0 = 1 byte	4	2^4 = 16 bytes	8	2^8 = 256 bytes
1	2^1 = 2 bytes	5	2^5 = 32 bytes	9	2^9 = 512 bytes
2	2^2 = 4 bytes	6	2^6 = 64 bytes	10	2^{10} = 1024 bytes
3	2^3 = 8 bytes	7	2^7 = 128 bytes	11	2^{11} = 2048 bytes
15:12	Reserved				

- **READ_BL_PARTIAL:** Defines whether partial block sizes can be used in block read commands.

  READ_BL_PARTIAL=0 means that only the READ_BL_LEN block size can be used for block oriented data transfers.

  READ_BL_PARTIAL=1 means that smaller blocks can be used as well. The minimum block size will be equal to minimum addressable unit (one byte)

- **WRITE_BLK_MISALIGN:** Defines if the data block to be written by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in WRITE_BL_LEN.

  WRITE_BLK_MISALIGN=0 signals that crossing physical block boundaries is invalid.

  WRITE_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

- **READ_BLK_MISALIGN:** Defines if the data block to be read by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in READ_BL_LEN.

  READ_BLK_MISALIGN=0 signals that crossing physical block boundaries is invalid.

  READ_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

- **DSR_IMP:** Defines if the configurable driver stage is integrated on the card. If set, a driver stage register (DSR) must be implemented also. S3F49DAX have no DSR implementation.

  DSR_IMP=0 no DSR implementation

  DSR_IMP=1 DSR implementation

- **C-SIZE:** This parameter is used to compute the card capacity. The memory capacity of the card is computed from the entries C_SIZE, C_SIZE_MULT and READ_BL_LEN as follows:

  *memory capacity = BLOCKNR * BLOCK_LEN*

  where

  *BLOCKNR = (C_SIZE+1) * MULT*
  MULT = 2C_SIZE_MULT+2 (C_SIZE_MULT < 8)
  BLOCK_LEN = 2READ_BL_LEN , (READ_BL_LEN < 12)

  Therefore, the maximal capacity which can be coded is 4096*512*2048 = 4 GBytes. Example: A 4 Mbyte card with BLOCK_LEN = 512 can be coded by C_SIZE_MULT = 0 and C_SIZE = 2047.

- **VDD_R_CURR_MIN, VDD_W_CURR_MIN:** The maximum values for read and write currents at the minimal power supply V_{DD} are coded as follows:

VDD_R_CURR_MIN VDD_W_CURR_MIN	Code for Current Consumption @ V_{DD}
2:0	000 = 0.5mA, 001 = 1mA, 010 = 5mA, 011 = 10mA, 100 = 25mA, 101 = 35mA, 110 = 60mA, 111 = 100mA

- **VDD_R_CURR_MAX, VDD_W_CURR_MAX:** The maximum values for read and write currents at the maximal power supply V_{DD} coded as follows:

VDD_R_CURR_MAX VDD_W_CURR_MAX	Code for Current Consumption @ V_{DD}
2:0	000 = 0.5mA, 001 = 1mA, 010 = 5mA, 011 = 10mA, 100 = 25mA, 101 = 35mA, 110 = 60mA, 111 = 100mA

- **C_SIZE_MULT:** This parameter is used for coding a factor MULT for computing the total device size (see 'C_SIZE'). The fac tor MULT is defined as 2C_SIZE_MULT+2.

Bit Value	Description	Bit Value	Description
0	$2^4 = 4$	4	$2^6 = 64$
1	$2^3 = 8$	5	$2^7 = 128$
2	$2^4 = 16$	6	$2^8 = 256$
3	$2^5 = 32$	7	$2^9 = 512$

- **ERASE_GRP_SIZE:** The contents of this register is a 5 bit binary coded value, used to calculate the size of the erasable unit of the card. The size of the erase unit (also referred to as erase group) is determined by the ERASE_GRP_SIZE and the ERASE_GRP_MULT entries of the CSD, using the following equation:

  size of erasable unit = (ERASE_GRP_SIZE + 1) ´ (ERASE_GRP_MULT + 1)

  This size is given as minimum number of write blocks that can be erased in a single erase command.

- **RASE_GRP_MULT:** A 5 bit binary coded value used for calculating the size of the erasable unit of the card.

- **P_GRP_SIZE:** The size of a write protected group. The contents of this register is a 5 bit binary coded value, defining the number of erase groups which can be write protected. The actual size is computed by increasing this number by one. A value of zero means 1 erase group, 31 means 32 erase groups.

- **P_GRP_ENABLE:** A value of '0' means no group write protection possible.

- **EFAULT_ECC:** Set by the card manufacturer. It defines the ECC code, which is recommended for use. The field definition is the same as for the ECC field described later.

- **R2W_FACTOR:** Defines the typical block program time as a multiple of the read access time. The following table defines the field format.

R2W_FACTOR Value	Multiple of Read Access Time
0	1
1	2 (write half as fast as read)
2	4
3	8
4	16
5	32
6, 7	Reserved

- **WRITE_BL_LEN:** Block length for write operations. See READ_BL_LEN for field coding.

- **WRITE_BL_PARTIAL:** Defines whether partial block sizes can be used in block write commands.

  WRITE_BL_PARTIAL='0' means that only the WRITE_BL_LEN block size can be used for block oriented data write.

  WRITE_BL_PARTIAL='1' means that smaller blocks can be used as well. The minimum block size is one byte.

- **FILE_FORMAT_GRP:** Indicates the selected group of file formats. This field is read-only for ROM. The usage of this field is shown in FILE_FORMAT description.

- **COPY:** Defines if the contents is original (= '0') or has been copied (='1'). The COPY bit for OTP and MTP devices, sold to end consumers, is set to '1' which identifies the card contents as a copy. The COPY bit is an one time programmable bit.

- **PERM_WRITE_PROTECT:** Permanently protects the whole card content against overwriting or erasing (all write and erase commands for this card are permanently disabled). The default value is '0', i.e. not permanently write protected.

- **TMP_WRITE_PROTECT:** Temporarily protects the whole card content from being overwritten or erased (all write and erase commands for this card are temporarily disabled). This bit can be set and reset. The default value is '0', i.e. not write protected.

- **CONTENTS_PROT_APP:** This field in the CSD indicates whether the content protection application is supported. S3F49DAX cannot support the contents protection application.

- **FILE_FORMAT:** Indicates the file format on the card. This field is read-only for ROM. The following formats are defined:

FILE_FORMAT_GRP	FILE_FORMAT	Type
0	0	Hard disk-like file system with partition table
0	1	DOS FAT (floppy-like) with boot sector only (no partition table)
0	2	Universal File Format
0	3	Others / Unknown
1	0, 1, 2, 3	Reserved

A more detailed description is given in MultiMediaCard File System Specification.

- **ECC:** Defines the ECC code that was used for storing data on the card. This field is used by the host (or application) to decode the user data. The following table defines the field format:

ECC	ECC_TYPE	Maximum Number of Correctable Bits Per Block
0	None (default)	None
1	BHC (542, 512)	3
2, 3	Reserved	–

- **CRC:** The CRC field carries the check sum for the CSD contents. The checksum has to be recalculated by the host for any CSD modification. The default corresponds to the initial CSD contents.

RCA Register (Relative Card Address Register)

The writable 16-bit relative card address register carries the card address assigned by the host during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The default value of the RCA register is 0x0001. The value 0x0000 is reserved to set all cards into the Stand-by State with CMD7.

DSR Register (Driver Stage Register)

S3F49DAX is not implement

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Ratings	Unit
V_{DD}	Supply voltage	–0.3 to 3.8	V
V_{IN}	Input voltage	–0.3 to V_{DD} + 0.3	V
I_{IN}	DC input current	–10	UA
T_{STG}	Storage temperature	–40 to 125	°C

RECOMMANDED OPERATING CONDITIONS

Symbol	Parameter	Ratings	Unit
V_{DD}	Supply voltage	1.65 to 1.95 / 2.7 to 3.6	V
T_{OPR}	Operating temperature	–25 to 85	°C

DC ELECTRICAL CHARACTERISTICS

Table 18. DC Electrical Characteristics

(V_{DD} = 1.65 V to 1.95 V, 2.7 V to 3.6 V, T_A = –25 to 85 °C)

Symbol	Parameters	Conditions	Min	Type	Max	Unit
V_{IH}	High level input voltage	V_{DD} = 2.7 V to 3.6 V	0.625×V_{DD}		V_{DD} + 0.3	V
		V_{DD} = 1.65 V to 1.95 V	0.7 × V_{DD}		V_{DD} + 0.3	V
V_{IL}	Low level input voltage		V_{SS} – 0.3		0.25 × V_{DD}	V
			V_{SS} – 0.3		0.30 × V_{DD}	V
I_{IH}	High level input current		–10		10	uA
I_{IL}	Low level input current	Pull-up Resistor: 10K	–1000		10	uA
		Pull-up Resistor: 60K	–300		10	uA
			–10		10	uA
V_{OH}	High level output voltage	I_{OH} = –100 uA V_{DD} = 2.7 V to 3.6 V	0.75 × V_{DD}			V
		I_{OH} = –100 uA V_{DD} = 1.65 V to 1.95 V	V_{DD} – 0.2			V
V_{OL}	Low level output voltage	I_{OH} = 100 uA V_{DD} = 2.7 V to 3.6 V			0.125 × V_{DD}	V
		I_{OH} = 100 uA V_{DD} = 1.65 V to 1.95 V			0.2	V
I_{CC}	Operating Current	Read/Write/Erase			20	mA
I_{STOP}	Standby Current				100	uA
C_{IN}	Input capacitance (NOTE)	Any input and bi-directional buffers			4	pF
C_{OUT}	Output capacitance (NOTE)	Any output buffer			4	pF

NOTE: This value excludes package parasitic

AC CHARACTERISTICS

Table 19. System Clock Timing

Symbol	Parameter	Min	Typ	Max	Unit
T_C	Clock cycle time	54	55	63	ns
Tlpd	Clock low pulse duration	0.4Tc	0.5Tc	0.6Tc	ns
Thpd	Clock high pulse duration	0.4Tc	0.5Tc	0.6Tc	ns

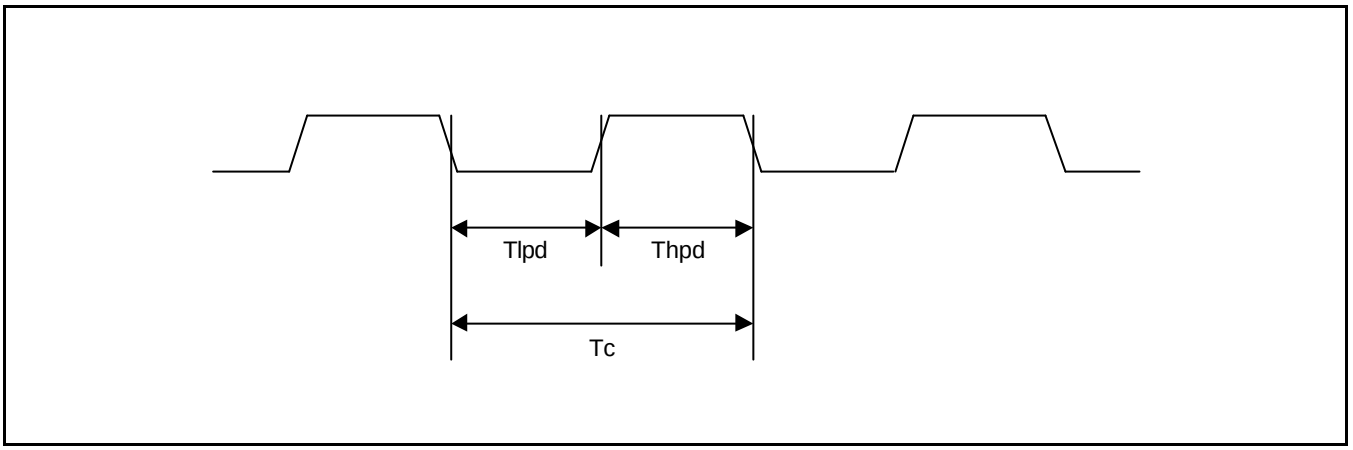

Table 20. POR (Power On Reset) Detection Level

Symbol	Parameter	Min	Typ	Max	Unit
POR	POR detection level	1.1	1.3	1.55	V

MECHANICAL DATA