

iCE65 as SDIO Controller (SD mode)

Overview

The Secure Digital(SD) Memory Card is a flash based memory, which is designed to meet security, capacity and performance requirements of various audio and video consumer electronic devices like mobile phones, digital cameras, digital recorders etc.. This design example describes the implementation of SDIO Controller in SD 4-bit mode using iCE FPGAs.

Description

SD Card protocol consists of an exchange of command and/or response tokens between the host and the card. Data transfers occurs in packets. A packet consists of a data block and CRC check bits. Data transfer can occur in single block or multi-block. There are three bus protocols defined for SD Cards, i.e., 4 line SD mode, 1 line SD mode and the 2 line SPI mode. The default mode is SD mode but the card can be configured to use a slower SPI mode. Protocol selection is done during the first reset command after power up. The SD Card powers up in SD mode. To switch the card to SPI mode, the CS signal must be asserted while the host issues a reset command to the card. Once a particular protocol is configured, it cannot be changed while power is applied. The only way to switch between protocol modes is to do a power cycle.

The SD 4-bit protocol uses seven pins. They are:

- CS : Active low chip select from host to card
- CLK : Host to card clock signal
- CMD : Host to card and vice versa(command signal)
- DAT[3:0] : Host to card and vice versa(data signal)

SD Cards can operate in the frequency range of 0-25MHz.

Messages in the SD protocol consists of command, response and data blocks. In SD configuration, all the data transaction between the master and the slave is started and terminated by the master. All transmissions on the CMD/DAT pins are done with MSB first. Commands and responses are sent on the CMD pin while data transfer is done on the DAT pins. CRC is used to provide bus transfer protection in the SD mode. The SD mode supports single and multi block read/write operations.

Implementation

The SDIO controller implementation is based on SD card specification version 2.0(Non-licensed version). Figure 1 shows the overall interface of SDIO controller to both the host processor and the SD Memory card.

The controller FSM block monitors host side transaction requests to the SD card through CPU interface logic. Depending on the host processor request, appropriate commands are sent to the SD card. After receiving the right response, the controller either sends the address and reads/writes data for single/multiple block read/write operation. The design has two built in data FIFOs to store both the transmit and receive data. The controller checks the status of FIFO before carrying out a transaction. The master FSM is the heart of the SD card controller which makes sure that correct transactions take place in sequence . Figure 2 shows a simplified diagram of the state machine.

1. Power-on brings the SDIO controller to power-up-initialization state. The SD mode is selected through a sequence of initialization commands.

2. Host uses read, write, register_addr and data_bus pins to configure SD card controller to desired data transfer width and operation mode.

3. {read_write, register_addr} signals configures read/write operations from/to host CPU as follows:

- when "011", writes data from data_bus to fifo module
- when "111", reads fifo contents to data_bus
- when "001", writes data bus contents to configuration register
- when "110", reads status register contents to data_bus.

4. There are two FIFO modules, one each for writing and reading operations with respect to CPU. CPU addresses the FIFOs either for reading or writing the data by giving the appropriate read, write, register_addr signals.

Table 1 lists the configuration register bit mapping.

5. Configuration register bits program the SDIO controller operation, data block length and number of data blocks as follows:

- STOP R/W operation when configuration register(0) is '1',
- config_reg(10 downto 1) decides the data block length(Maximum 16).
- config_reg(14 downto 11) decides the number of data blocks for R/W operations. Maximum number of words per read/write block is 256 (512 Bytes).

6. Combination of {read, write,config_reg(0)} signals from host CPU issues following commands to the SD card controller:

- "10" - Read single/multiple block(depending on number of blocks)
- "00" - Write single/multiple block(depending on number of blocks)
- "01" - Stops read/write operations

6. Interrupt from SDIO controller – SDIO controller generates a High on CPU_interrupt pin under the following conditions: write fifo empty, read fifo full, Initialization done, SD read/write completion.

7. Status register contents provide following SDIO controller status : {NONE, NONE, write fifo full, write fifo empty, read fifo full, read fifo empty, initialization done, read/write acknowledge}.

Table 2 lists the status register bit mapping.

Table 3 lists the pin description of SDIO controller. Table 4 summarizes the post P&R resource utilization summary of this Design Example when implemented using iCE65 FPGA.

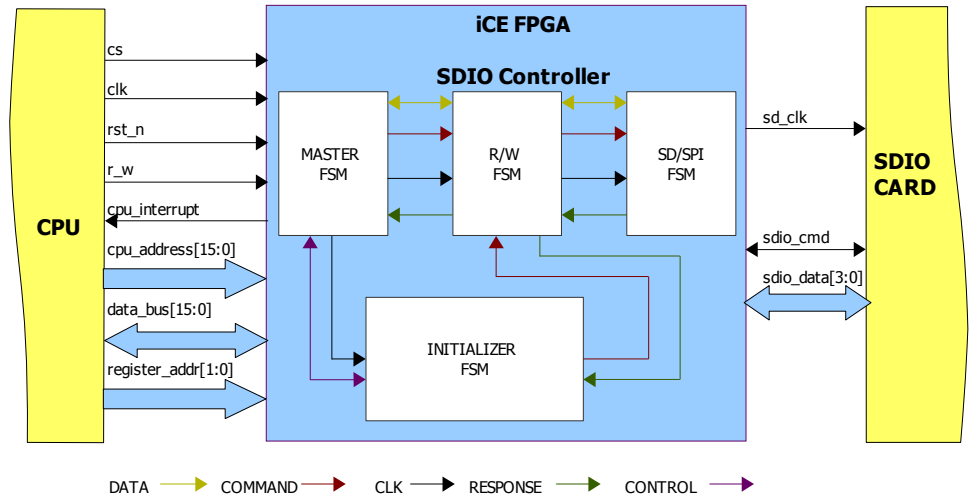


Fig 1: SDIO Controller block diagram

CFG15	CFG14	CFG13	CFG12	CFG11	CFG10	CFG9	CFG8	CFG7	CFG6	CFG5	CFG4	CFG3	CFG2	CFG1	CFG0
RW_SIZE7	RW_SIZE6	RW_SIZE5	RW_SIZE4	RW_SIZE3	RW_SIZE2	RW_SIZE1	RW_SIZE0	RDSTART	WRSTART	NUMBLK2	NUMBLK1	NUMBLK0	INTRCLR	INTREN	STOP

Table 1: Configuration Register

STS15-7	STS6	STS5	STS4	STS3	STS2	STS1	STS0
X	CRCERR	WREEMPTY	WRFULL	RDEEMPTY	RDFULL	INITDONE	RW_ACK

Table 2: Status Register

Pin	Direction	Description	Comments
cs	Input	SD card select. Active low.	Any IO Bank/Voltage Level
clk	Input	System Clock	Global Clock Signal. Any IO Bank/Voltage Level
rst_n	Input	System Reset, Active Low	Global reset signal. Any IO Bank/Voltage Level
r_w	Input	Read- Active High, Write – Active Low	Any IO Bank/Voltage Level
cpu_address[15:0]	Input	Host Address Bus	Any IO Bank/Voltage Level
cpu_interrupt	Output	Asserts high when generates an interrupt to host	Any IO Bank/Voltage Level
data_bus[15:0]	Inout	Host Data Bus	Any IO Bank/Voltage Level
register_addr[1:0]	Input	Register Select	Any IO Bank/Voltage Level
sdio_clk	Output	SD card clock	Any IO Bank/Voltage Level
sdio_cmd	Inout	SD card command and response pin	Any IO Bank/Voltage 3.3V, with Pull-up.
sdio_data[3:0]	Inout	SD card data in and out pins	Any IO Bank/Voltage 3.3V, with Pull-up

Table 3: Pin Description

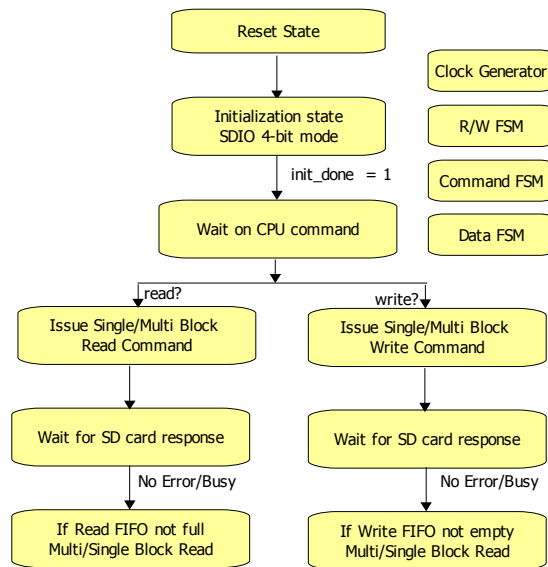


Fig 2: Simplified State Diagram

Device	Logic Cells	IO Cells
iCE65L04-UCB284	1175	45

Table 4: Resource Utilization

Conclusion

This design example demonstrates the implementation of a SDIO controller using iCE FPGAs. The low power, high density and speed properties of SD cards are well complemented by iCE FPGA's very low power and high speed capabilities. This makes iCE FPGAs an obvious choice for SDIO controller in all low power and portable applications like feature-rich cellphones, smartphones, digital cameras.

Disclaimer

SiliconBlue is providing this document, design example, or information "as is." SiliconBlue does not ensure that this implementation or information is void of any claims of infringement. As a reader / implementer, you are responsible for obtaining any rights you may require for your implementation. SiliconBlue also does not warranty the fitness of this design example to be readily implemented in any specific context. It is your sole responsibility to validate this design example and its correctness while implementing it for your requirement.

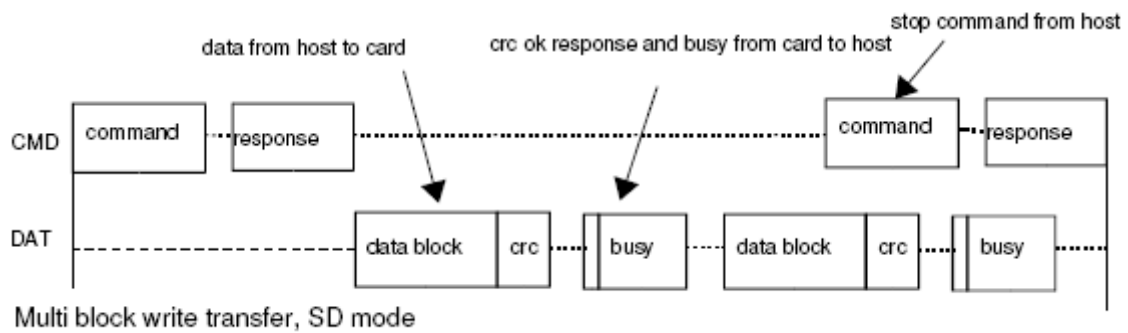
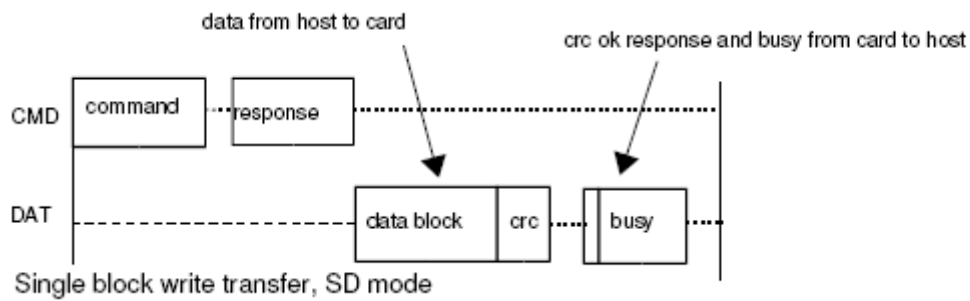
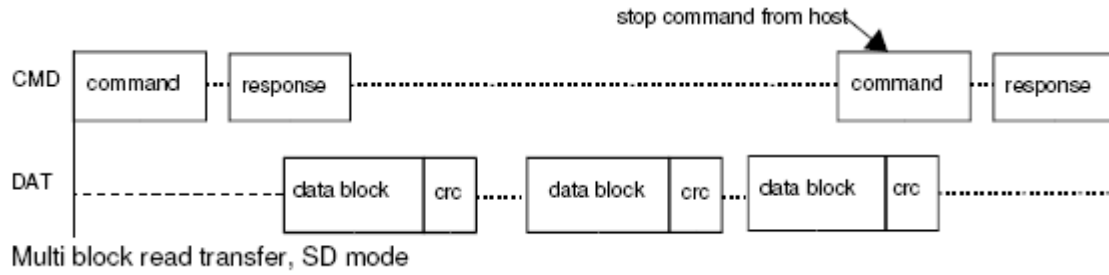
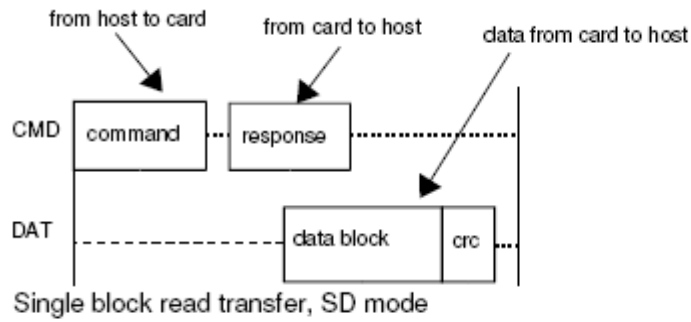


Fig 3: SD Card single/multi block read/write sequence