FPGA Implementation of a SD Card Controller using SPI communication

Conference Paper · June 2020

DOI: 10.1109/ECAI50035.2020.9223238

Dumitrel Catalin Costache
Polytechnic University of Bucharest
3 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Lucian Andrei Perisoara
Polytechnic University of Bucharest
50 PUBLICATIONS **166** CITATIONS
SEE PROFILE

FPGA Implementation of a SD Card Controller using SPI communication

Dumitrel Cătălin Costache
Faculty of Electronics, Telecommunications and Information Technology
University Politehnica of Bucharest
Romania
costache.c22@gmail.com

Lucian Andrei Perişoară
Faculty of Electronics, Telecommunications and Information Technology
University Politehnica of Bucharest
Romania
lucian.perisoara@upb.ro

Adriana Florescu
Faculty of Electronics, Telecommunications and Information Technology
University Politehnica of Bucharest
Romania
adriana.florescu@upb.ro

***Abstract*—The Field Programmable Gate Array (FPGA) devices are being used more and more, especially in applications that require real-time data processing, for example in communication systems. To meet the memory requirements for FPGA systems, the onboard memory can be used, but it cannot be expanded easily by adding some cards. More useful are Secure Digital (SD) cards that are detachable and can be replaced easily. In this paper, we have designed and implemented a FPGA controller that allows writing and reading SD cards using the SPI protocol. The main objective is to provide a storage solution for FPGAs, to store large files on large capacity, cheap, portable and easy to use storage devices. Using the SPI protocol, the system can access multiple SD cards with a minimum usage of data lines from FPGA. The hardware design and implementation are developed on a Xilinx AC701 platform with Artix-7 FPGA, using VHDL language. The FPGA controller was tested with more SD cards and the results shown that the system can run without errors.**

Keywords—FPGA, Artix7, SD Card, SPI, SPI Mode

I. Introduction

Secure Digital (SD) cards are proprietary non-volatile memory cards based on a new generation of semiconductor electronic devices, mainly developed for data storage in portable or stationary applications. The major advantages of SD cards are small size, high storage capacity, fast data bit rates, high power efficiency, great flexibility and good security. Along with USB memory sticks, the SD cards are most used portable storage devices in digital systems based on microprocessors, microcontrollers, Digital Signal Processors (DSP) or Field Programmable Gate Arrays (FPGA) chips, in different applications, like digital cameras, laptops, mobile phones and embedded systems [1], [2], [3].

The answer to the needs and requirements of users was the establishment of SD card Association (SDA), with the aim of promoting and creating SD card standards, which define the card types, speed classes, electrical interface, and communication protocols. The SD cards support three communication protocols: 1-bit SD mode, 4-bit SD mode and Serial Peripheral Interface (SPI) mode [1], [2].

The FPGAs are semiconductor electronic devices that are built from Configurable Logic Blocks (CLBs) which can be interconnected by programming language. Unlike Application Specific Integrated Circuits (ASIC), FPGAs can be reprogrammed to perform certain tasks in different applications like audio/video processing, wired and wireless communications, data storage, test and measurement equipment, automotive electronic control units, Industrial, Scientific and Medical (ISM) equipments [4].

Different applications of reading large data files from SD cards using FPGAs were designed and implemented for audio [5] and image [3] processing, based on the structure of 1-bit SD transfer mode. Also, in [6] was implemented a FPGA system that read the images stored on a SD card and display them on a classical TV monitor. For low power applications, a FPGA card controller for SD mode is proposed in [7], which combines a bus interface unit and a card interface unit, the controller being shut down if no data or command is transferred for a long time. For Wireless Sensor Networks (WSN), a microcontroller based sensor node was developed in [8], provided with a microSD card for locally storing of large amounts of environmental data.

Because the SPI interface supports communication between more devices connected on the same bus, the main objective of our work is to develop a writing/reading transmission protocol based on SPI interface, as a basis for a single FPGA controller for multiple SD cards.

The paper is organized as it follows. Section II presents the main system features for SD Cards, while Section III describes the Xilinx AC701 development platform for Artix-7 FPGA. The design methods and implementation results for SD card controller are presented in Section IV. Finally, Section V concludes the paper.

II. Main Characteristics of SD Cards

The SD card standard specifies following card types depending on memory capacity: Standard Capacity SD (SDSC) cards for up to 2 GB, High Capacity SD (SDHC) cards for 2 – 32 GB, Extended Capacity SD (SDXC) cards for 32 GB – 2 TB, Ultra Capacity SD (SDUC) cards for more than 2 GB and up to 128 TB. Other features of SD cards are: different form factors (standard SD, miniSD, microSD), card detection for insertion or removal, write protection using mechanical switch, copyright protection mechanism, file protection and encryption system, maximum bit rates from 25 Mbps (default mode) up to 7.88 Gbps (PCIe Gen3 mode), communication bus with 4 parallel data lines, UHS-II differential lines or PCI Express (PCIe) differential lines [1].

A. Electrical interface

The electrical interface of an SD card is based on a 9-pin socket, designed to operate with low voltages, see Table I. All SD memory cards use initially a 3.3 V signaling on Default Speed mode. Using some programming commands, SDHC and SDXC cards can be switched to low voltage operation at 1.8 V. In this system, the development kit that we used, provides a voltage of 3.3V to power the SD cards.

978-1-7281-6843-2/20/$31.00 ©2020 IEEE

TABLE I. SD CARD PINOUT AND COMMUNICATION MODES

Pin #	SD Mode		SPI Mode	
	Name	Description	Name	Description
1	CD/DAT3	Card Detect / Data Bit 3	CS	Chip Select
2	CMD	Command / Response	DI	Data In
3	VSS	Ground	VSS	Ground
4	VDD	Supply voltage	VDD	Supply voltage
5	CLK	Clock	SCLK	Clock
6	VSS	Ground	VSS	Ground
7	DAT0	Data Bit 0	DO	Data Out
8	DAT1	Data Bit 1	RSV	Reserved
9	DAT2	Data Bit 2	RSV	Reserved

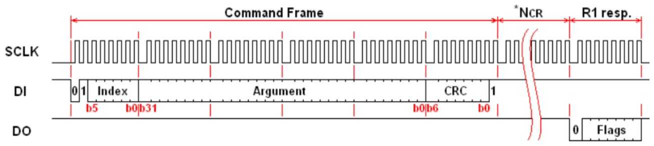

Fig. 1. A command frame transmitted from host to SD card.

TABLE II. BASIC COMMANDS SET FOR SPI MODE

Command Index	Argument	Response	Description
CMD0	None(0)	R1	Software reset.
CMD1	None(0)	R1	Start initialization process
ACMD41	*1	R1	Start initialization process
CMD8	*2	R7	Check voltage range (only for SD v2)
CMD9	None(0)	R1	Read CSD register
CMD10	None(0)	R1	Read CID register
CMD12	None(0)	R1b	Stop to read data
CMD16	Block length[31:0]	R1	Change read/write block size
CMD17	Address[31:0]	R1	Read a block
CMD18	Address[31:0]	R1	Read multiple blocks
CMD23	No. of blocks[15:0]	R1	Number of transferred blocks with CMD18/CMD25
ACMD23	No. of blocks[22:0]	R1	Number of pre-erased blocks with CMD25
CMD24	Address[31:0]	R1	Write a block
CMD25	Address[31:0]	R1	Write multiple blocks
CMD55	None(0)	R1	Leading command of ACMD<n> command
CMD58	None(0)	R3	Read OCR

*1: Rsv(0) [31], HCS[30], Rsv(0) [29:0]
*2: Rsv(0) [31:12], Supply Voltage(1) [11:8], Check Pattern (0xAA) [7:0]

B. *Transmission modes*

The SD card supports three transmission modes [1]:

- 1-bit SD mode is the standard serial protocol, with bidirectional one command (CMD) and one data (DAT3) lines. The data bit rate is up to 25 Mbps;
- 4-bit SD mode, with the same lines as 1-bit SD mode, but with extra 3 data lines (DAT0, DAT1, DAT2) for 4-bit parallel transmission. Although the bit rate is up to 100 Mbps, time synchronization makes this mode more difficult to be implemented;
- SPI mode, with unidirectional serial Data Input (DI) and serial Data Output (DO) lines. The transmission bit rate is up to 25 Mbps.

C. *SPI communication protocol*

The SPI standard defines only the physical layer and not the entire communication protocol. The SPI mode for SD cards uses a command subset of the SD mode protocol. Four different modes are defined for SPI protocol, depending on phase and polarity of SCK signal, SPI mode 0 being defined for SD cards. The SPI command set for generic read/write operations and card initialization is presented in Table II [1].

After power on, the SD card will enter automatically in 1-bit SD mode. The host will send a reset command CMD0. During this, if the CS signal line is asserted in Low state, the SD card will recognize that the SPI mode is needed. So, the card will choose the SPI mode and will send the R1 response frame. If CS signal remains High, the card will not respond at all and will remain in the SD mode. Also, the host may send another command to switch from 1-bit to the 4-bit SD mode, if the card supports. The transmission mode cannot be changed until the SD card is no longer powered.

After initialization of SPI mode, the SD card is ready for data transfer, the read/write operations, see Fig. 1. If the DO signal is set to High, then the card is ready for reception. After a command frame is sent from the host to the card on the DI line, a response frame is sent back from the card to the host on the DO line. Because the transmission is controlled by the CLK signal of the host, the last one must continuously read data, send 0xFF and get the received data, until a valid response is obtained. During the read transfer, the DI signal must be in High state. Within the command response time N_{CR}, which must be less than 8 bytes, a response frame must be received by the host. Before sending a command, the CS signal must change from High state to Low state and must remain in Low during the entire transmission. In SPI mode, the CRC field is optional, not being checked by the SD card.

A command frame of 48 bits begins with a start bit “0” and the direction bit “1”, which indicates the direction of transmission, from host to device, see Fig. 1. The following 6 bits represent the command index from Table II. Some commands need an argument coded on 32 bits. All command frames are error protected by a 7-bit Cyclic Redundancy Check (CRC) code and are terminated by the end bit “1”.

III. FPGA DEVELOPMENT BOARD

A. *Xilinx AC701 development board for Artix-7 FPGA*

The AC701 Evaluation Kit is a ready-to-use development platform which contains all basic hardware and software components, design tools, IP cores, and reference designs to quickly prototype different high-performance serial connectivity applications, see Fig. 2.

Fig. 2. Xilinx AC701 development board for Artix-7 FPGA.

The AC701 development board provides usual characteristics which are common for many embedded development platforms, including the Artix-7 XC7A200T-2FBG676C FPGA (215360 logic cells, 740 DSP slices, 13140 memory, 500 Input/Output pins), MicroBlaze 32 bit RISC processor, 1 GB DDR3 SODIMM memory, SD card slot, PCI Express interface, a Gigabit Ethernet interface, UART interface, general purpose Input/Output ports [9].

For the development of a FPGA controller for the SD card, we need to use following blocks from the block diagram of Xilinx AC701 development board: the SD Card Interface, the Quad SPI Flash Memory and the USB to UART Bridge, see Fig. 3.

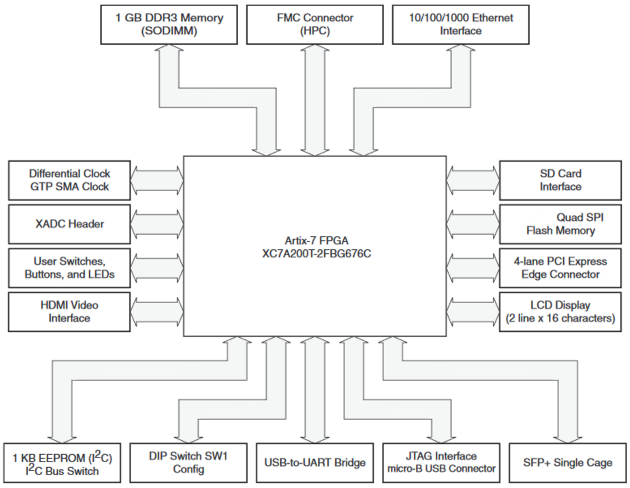
1 GB DDR3 Memory (SODIMM)
FMC Connector (HPC)
10/100/1000 Ethernet Interface
Differential Clock GTP SMA Clock
XADC Header
User Switches, Buttons, and LEDs
HDMI Video Interface
Artix-7 FPGA XC7A200T-2FBG676C
SD Card Interface
Quad SPI Flash Memory
4-lane PCI Express Edge Connector
LCD Display (2 line x 16 characters)
1 KB EEPROM (I2C) I2C Bus Switch
DIP Switch SW1 Config
USB-to-UART Bridge
JTAG Interface micro-B USB Connector
SFP+ Single Cage

Fig. 3. The block diagram of Xilinx AC701 development board [9].

B. *SD card interface*

The evaluation board AC701 includes a Secure Digital Input / Output (SDIO) connector to provide access to general purpose SD memory cards and peripherals, see Fig. 4. The SD card slot is designed to support up to 50 MHz cards. The SD signals are connected to FPGA I/O bank 14, with the supply voltage VCC set to 3.3 V [9].

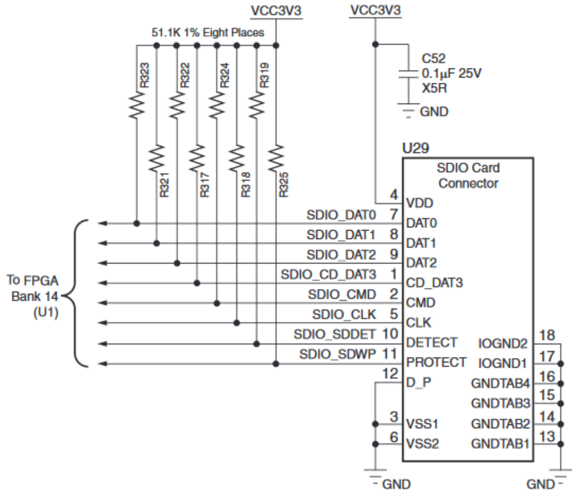

Fig. 4. The SD card interface of Xilinx AC701 development board.

C. *Quad SPI Flash Memory*

The JTAG interface allows the user to download the bitstreams on the FPGA board. To do this, the host computer must have installed the Xilinx software tools.

To store the configuration, the AC701 development board is equipped with a non-volatile 256Mb memory, which uses the Quad SPI communication interface. The FPGA board requires to use a Quad SPI memory because it is necessary to store the bitstream in order to be able to reload the configuration if the board is restarted.

IV. Design and Implementation

The FPGA prototyping for the SD card controller is carried out with the Very High-Speed Integrated Circuit Hardware Description Language (VHDL) using the Vivado Design Suite from Xilinx.

A. *FPGA controller structure*

The FPGA block diagram of the proposed SD card controller with SPI interface is shown in Fig. 5. The main advantage of using the SPI transmission mode instead of SD mode is the possibility to connect more SD cards on the same communication bus, thus the memory capacity is increased linearly with the number of cards used. In SPI mode, the clock (SCLK), Data Input (DI) and Data Output (DO) lines are common for all SD cards, and the chip selection (CS1-*n*) and chip detection (CD1-*n*) lines are dedicated for each SD card.

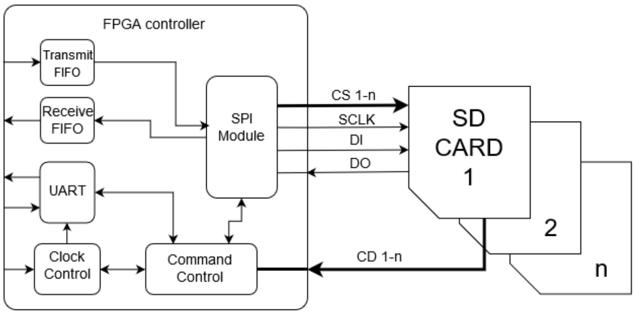

Fig. 5. FPGA block diagram for the SD card controller.

The Clock Control module provides a clock signal with a frequency of 156.250 MHz, necessary for the operation of all modules, and a clock signal, with an adjustable frequency between 0 to 25 MHz, necessary for data transmission.

The SPI module ensures the transmission of data and commands to the SD card. On signal lines DI and DO, the data are transmitted unidirectionally, one byte at a time, until all data are transmitted.

The Command Control module assures all the functions of the SPI communication protocol.

Because the size of a memory location (one sector) from SD card is fixed at 512 bytes, it was necessary to add two FIFO buffers for temporary storage of data that will be transmitted, respectively received.

B. *SD card initialization*

The Finite State Machine (FSM) for SD card initialization has the following functional diagram.

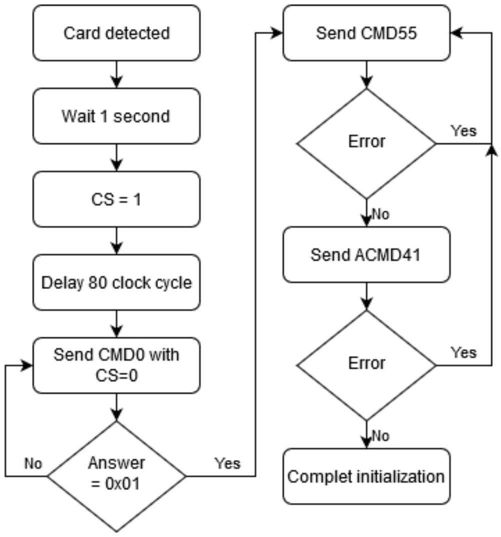

Fig. 6. FSM functional diagram for SD card initialization.

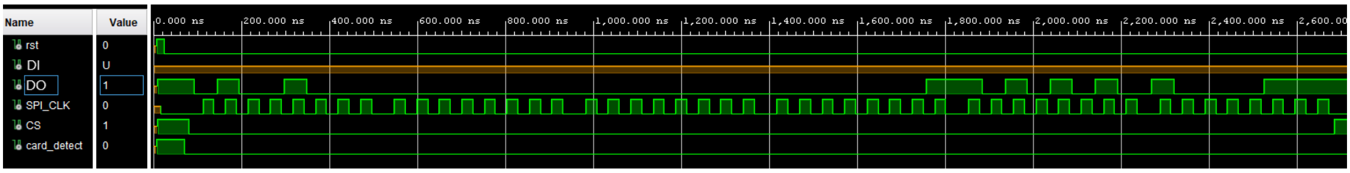

Fig. 7. CMD8 transmission simulation.

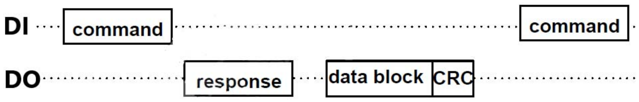

Fig. 8. Single block read operation.

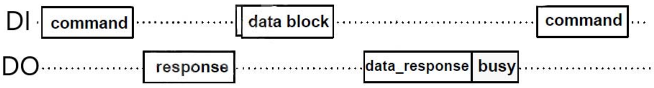

Fig. 9. Single block write operation.

After the SD card is detected using the CD signal, it is powered on, and the control logic initialize the card. According to the SD card protocol, SPI clock rate is set between 100 kHz and 400 kHz. The DI and CS signals are set to High and 80 clock pulses are applied on SCLK line.

The SD card will enter in its native operating mode and to reset it the CMD0 command with CS asserted on Low is firstly sent. To detect the card status, the host continuously send the CMD55 and ACMD41 commands, until it receives the correct answer. When the initialization has been done, the host increases the clock frequency to 25 MHz.

A simulation for CMD8 transmission is shown in Fig. 7. The CS signal is asserted on Low when the transmission is started, and it is changed back to High when the transmission or reception of all bytes is completed.

C. Data read

In SPI mode, the available commands are CMD17 for single block read operation and CMD18 for multiple blocks read operation. If the read command is accepted, the card will respond with a confirmation message followed by the data block, see Fig. 8. A correctly transmitted data block is finished with the 16-bit CCITT CRC code, which in SPI mode can be deactivated with the command CMD59.

The data read from SD card are written in Receive FIFO buffer and then the system must wait until the user reads the data from buffer. When Receive FIFO buffer is empty, another read command is performed.

When using the multiple read operation, the SD card will transfer a 16-bit CRC suffix after each data block. To stop the reading operation, the host must send the CMD 12 to the SD card.

If the SD card detects an error when reading the data, it will send an error frame instead of a data block.

D. Data write

To write data on SD card, firstly, the user needs to write data to Transmit FIFO buffer. If the data are present, the host will send a write command to the card, CMD24 for single block write operation or CMD25 for multiple blocks write operation, see Fig. 9. If the write command is accepted, the SD card will respond with a confirmation message and will wait for the host to send a data block.

After each data block is received correctly, the SD card sends a confirmation response to the host. If the data block has been received without errors, the card will store it. When the Transmit FIFO buffer is empty, the user can add new data to be transmitted to the card.

The functional testing of the system was done using SD cards from several manufacturers. For communication between computer and FPGA, the UART module was used. For data reading, a randomly generated binary file stored on SD card was read by the FPGA controller, and further sent to the computer. For data writing, a binary file was received by the FPGA from the computer and written on the SD card. Finally, we have compared the files resulted after each operation with the original file and all files were identically.

The SD card controller core was developed in VHDL Xilinx for Artix-7 FPGA based development board, consuming 414 Look-Up Tables (LUT), 283 Flip-Flops (FF), 13 I/O pins and 1 global buffer (BUFG).

V. Conclusion

In this paper, we have proposed a FPGA controller that allows writing and reading SD cards using the SPI protocol. The SPI transmission mode is a must in order to have a storage system easily scalable. Adding an additional number of SD cards involves sharing the same clock and data signals and different chip-select signals for each card. The proposed architecture involves accessing each card in turn but allows the availability of a large amount of memory.

References

[1] SD Card Association, "SD Part 1, Physical Layer Simplified Specification, Version 7.10", 25 March 2020.

[2] SD Card Association, "SD Part E1, SDIO Simplified Specification, Version 3.00", 25 July 2018.

[3] Zhenlin Lu, Jingjiao Li, and Yao Zhang, "The reading/writing SD card system based on FPGA," First International Conference on Pervasive Computing, Signal Processing and Applications, 2010.

[4] Omar Elkeelany, Vivekanand S. Todakar, "Data Archival to SD Card Via Hardware Description Language," IEEE Embedded Systems Letters, Vol. 3, No. 4, pp. 105-108, Dec. 2011.

[5] Hong-Wei Liang, Jian-Ai Li, Ling-Ling Kan, "Implementation of SD Card Music Player Using Altera DE2-70," 2011 International Conference on Multimedia and Signal Processing, pp. 150-153, 2011.

[6] Yansi Yang, Yingyun Yang, Lipi Niu, Huabing Wang and Bo Liu, "Hardware System Design of SD Card Reader and Image Processor Based on FPGA," International Conference on Information and Automation, pp. 577-580, Shenzhen, China, June 2011.

[7] Pan Zhou, Teng Wang, Xin'an Wang, Yinhui Wang, "Hardware Implementation of a Low Power SD Card Controller", IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Guilin, China, pp. 158-161, 5-8 Aug. 2014.

[8] Dragoş Ioan Săcăleanu, Lucian Andrei Perişoară, Vasile Lăzărescu, Rodica Stoian, „A New Multipurpose Wireless Sensor Node for Data Acquisition Systems", 6th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, Vol. 6, No. 2, pp. 35-38, 23-25 Oct. 2014.

[9] Xilinx Inc., "AC701 Evaluation Board for the Artix-7 FPGA User Guide UG952 (v1.4)", 6 Aug. 2019.