THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Data Concentration and Archival to SD Card in Verilog Language

Dinesh K. P.
M.Tech Student, VLSI Design and Embedded Systems
BGS Institute of Technology, BG Nagar, Mandya, India
Anandaraju M. B.
Professor and HOD, Department of ECE
BGS Institute of Technology, BG Nagar, Mandya, India

Abstract:
Smart Grid (SG) is the next generation's power grid system. Delivering control, monitoring and management data to grid elements often requires the efficient archival of acquired information. The main objective of this research is to design an experimental platform for efficient, on-chip, real-time data concentrator for accessing data from a Secure Digital flash memory card using the SD bus protocol. All the hardware design is done using Verilog hardware descriptive language and implemented in Field Programmable Gate Array (FPGA). The design has four independent modules for the required different operations on the SD memory card. These four modules are for single block write, multiple block write, single block read, and multiple block read operations. A temporary data is either stored internally in an array of registers or externally in the Synchronous RAM for the analysis purposes. The design is implemented on the Altera's Cyclone II EP2C35F672C6 FPGA chip using 1GB SanDisk SD card and has shown a maximum data concentration rate of 25 Mbps.

***Key words**: SD Protocol, Flash memory read/write, Verilog HDL.*

1. Introduction

NON-VOLATILE memory (such as flash memory) devices are most widely used in data storage applications. Flash memories are used whenever a shock resistance is the key requirement [2]. In addition, Flash memory is used because of its compact, fast access, low-power consumption, and multiple write characteristics, as compared to mechanical magnetic or optical media, which do not have all of these desired features [2]. Also the stored data in Flash memory can be easily transferred to a personal computer whenever needed since it is designed for portable use. It is also worth mentioning that storing sensitive data to SD cards can be done due to inherent security features of the SD card.

The SanDisk SD Card, for example, provides features in multiple memory sizes. It includes an on-card intelligent controller which manages interface protocols, security algorithms for copyright protection, data storage and retrieval, as well as error correction code, defect handling and diagnostics, power management and clock control. All the technical details about the Secured Digital (SD) card are presented in [3]. The main flash memory controller is one which is present on the memory chip itself. This controller is basically designed to handle the bit errors, the bad blocks, maintaining the high data accessing speed, flash memory management, etc. The flash memory access controller is one which communicates with flash memory controller. An improvement on the flash memory controller designed in [4] gives a higher rate of data access. Such designs definitely improve the performance of a flash memory access controller. But this method cannot be adapted to every existing flash memory device. To do so the flash memory controller design has to be reconfigurable.

The MMC card is similar to the SD card. A design to read and write into MMC card is implemented by [10] using an SPI bus. This design introduces the interface example using Free scale DSP 56858 platform, but at the same time the I/O speed is lower than using a controller. In order to make use of a microcontroller to read or write secure digital card, according to synchronous peripheral interface protocol and secure digital card Synchronous Peripheral Interface (SPI) mode, the communication between synchronous peripheral interface of single chip microcomputer and secure digital card is studied in [11].

2. SD Card Controller Challenges

In traditional systems, Flash memories are written by a host personal computer, through a permanent interface (i.e. soldered chips on circuit boards). Such use of Flash memory devices is common in embedded systems to store configuration information. One design choice made in the proposed system is to use detachable Flash memory devices. Sampled data must be written to the Flash memory device to allow further remote analysis and interpretation. This introduces technical design challenges. Particularly, Flash memory must be written by a customized hardware not a personal computer. Hence, the proposed Flash memory controller must address the following design requirements.

2.1. High Data Rate

A Detachable Flash memory should have a small and robust physical interface. This limits the maximum number of data and control pins in the interface. Consequently, this affects the writing speed, as serial data communication must be employed. We use high clock frequency in serial mode to overcome this challenge. Timing and clock signals between the Flash memory controller and the data acquisition units must be properly matched. We use phase locked loop, and clock dividers to achieve this matching.

2.2. Data Integrity

Flash memory must be initialized before the storing process. As Flash memory cards are detachable, card detection is another challenge. The storing process cannot be done unless the Flash memory is attached. Continuous monitoring of the Flash memory during both the reading process and the writing process is done to ensure that the data is passed to the Flash memory. We also use Error check code to check the valid arrival and storage of the data into the Flash device.

2.3. Data Quality

The witting process is limited by the access time of the Flash memory device (table 1). High quality data should be received fully (without dropped blocks), in order, and in time. There is no need to store data which is incomplete or out of order. One solution we used is to use internal buffer. We assume no data compression, and

fixed data arrival rate. We make sure that the processing rate is faster than the arrival rate. In future work, we will consider variable data arrival rate (e.g. compressed data), and will evaluate the optimal buffer depth, such that no blocks are dropped.

2.4. Dual Clock Frequencies

One serious drawback of GPP architectures can be seen from examining technical details of an SD card manufacturer [1]. The challenge is that SD card manufacturers (such as SanDisk) specify two different clock frequencies; initialization mode (100 KHz – 400 KHz) and data transfer mode (25 MHz) for handling the card. Implementing these two different clock signals on a GPP microcontroller is not an easy task and the ultimate solution has to sacrifice the performance. As per [12] the SD card cannot be accessed at a frequency more than 25 MHz and, hence, if ds PIC microcontroller is used to interface SD card, the clock signal has to be reduced to half that of the maximum frequency the microcontroller can support. Another microcontroller design implementation using ATmega32 microcontroller [13] has dropped the operating clock frequency for the processor to 4 MHz frequency so that executed instructions can initialize the card according to specified low rate. Thus, instructions for data transfer will also operate at the very low clock frequency of the processor, which yields an access time for a single block (of 512 bytes) of 4.15 ms. such available systems, are completely instruction based and single clocked. In other words, the user has no access to clock period and uses software instruction delays to operate at a slower rate, or slows down the processor clock frequency by clock dividers.

3. Proposed System Design

Figure 1 shows a typical model for PC based network management for SG. It basically needs a computer for each network management aspect. This computer will be connected to the corporate network with its network interface card (NIC). Data will be stored in the PC (in Flash or hard drive media). This requires data communication to the PC that can be done serially via USB cable or on parallel cables. The cost and size of such

system can be prohibitive to many applications, like SG. Figure 2 shows the Processor on FPGA chip Handheld

Data Acquisition System. As shown in figures, components of the computer based data acquisition system were custom designed in [2]. Figure 2 clearly shows that the system in [2] has on chip processor and an external SDRAM, which are not required in the proposed system as shown in figure 3. Apart from being real time the proposed system greatly enhances the performance as it archives concentrated data locally before it is transmitted periodically to the corporate network for further analysis and management operations.

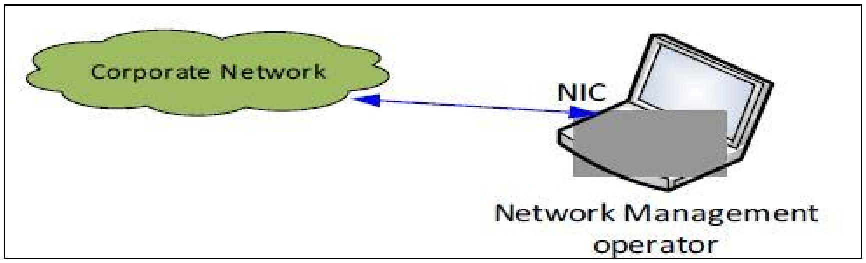

Figure 1: PC based network management

The Flash memory controller will be designed to copy the collected data temporary stored in the FPGA internal buffers. The Flash memory controller will be designed in the FPGA to write data directly to the Flash memory card, with a minimal interface. It is advantageous to use the FPGA internal memory cells for buffering because their access time is much smaller than that of external memories. The Flash memory controller will follow an error detection algorithm in the data writing process to ensure the integrity of the stored data. Overall, the FPGA will record acquired data after filling its internal storage buffers in non-real time mode. In order to optimize the acquire-and-write processes, the Flash memory controller can use single or multiple block writing mechanisms. In this research only single block mechanism is tested. Single block writing can be done to ensure data integrity, by using an integrity check value (Cyclic redundancy check) at the end of each block. However, if multiple blocks are to be written sequentially to the Flash memory, the total write time can be reduced significantly by use of a more sophisticated write mechanism. In the multiple blocks write mechanism, the total write time decreases.

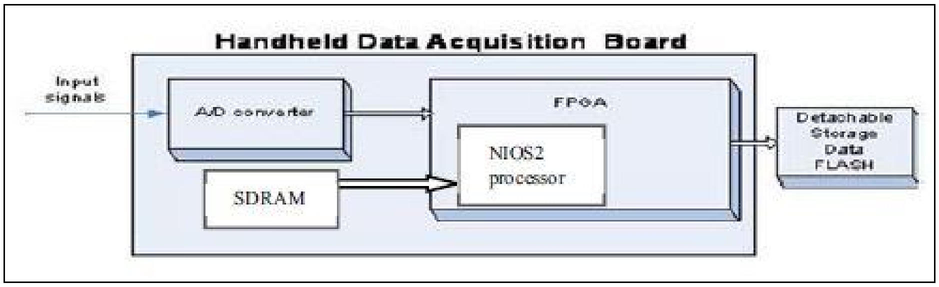
Handheld Data Acquisition Board
Input signals
A/D converter
FPGA
NIOS2 processor
SDRAM
Detachable Storage Data FLASH

Figure 2: Processor on chip Flash memory Controller [2]

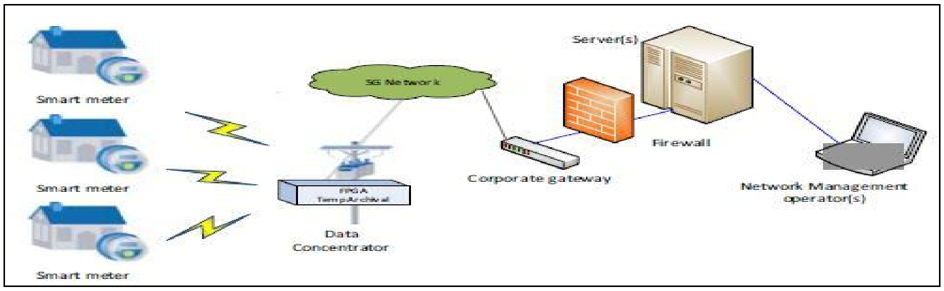

Figure 3: Proposed FPGA based Data Concentrator for Bidirectional Data Archival

A complete Hardware architecture is developed using Verilog Hardware description language (HDL). All the critical and non-critical tasks are implemented in hardware only. There is no use of any embedded C programming language. To achieve our goal a single block n write approach is used initially. A single block data write into the Flash memory device involves the write of 512-bytes of data. A 16-bit CRC is appended to each data block. The Flash memory device controller checks the CRC and if the CRC is the correct CRC then the data is written. But if any failure happens, the data is discarded. Then host tries to send the same data again starting with another single block write command. Any SD card can be used in two modes at any given time. It can be either in card identification mode or data transfer mode. The designed Finite state machine for SD card identification is shown in figure 4. Wherein the host resets the card initially and then validates operating voltage range by asking to publish Operating Condition Register (OCR) data, identifies the card using Card Identification Register (CID) and asks to publish Relative Card Address (RCA). All the data communication in the card identification mode uses only the command line (CMD) on SD card. Once the RCA is published the card completes its initialization and further communication for commands to data write and data read is done using RCA.

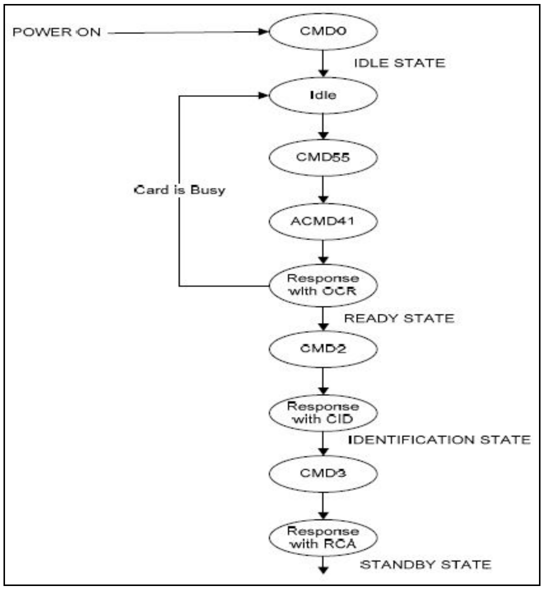

Figure 4: FSM for SD card identification

Block oriented data transfer. The sequence starts with a single block write command CMD24 that determines the start address. The card performs the CRC check for each data line at the end of each received data block prior to write operation. The data is suffixed with CRC check bits to allow the card to check for transmission errors. The card sends back the CRC check result as CRC status on the DAT0 line. CRC status ‘010’ indicates non-erroneous transmission, ‘101’ indicates transmission error and ‘111’ for flash programming errors. This 3 bit CRC status response output is always two clocks after the end of data. The data flow is terminated by a stop transmission command CMD12.

4. Experimental Results

The design implementation results are verified on the Tectonics TLA 611 Logic Analyzer and the results for various commands are shown below. In figure 5, the A3(0) signal indicates the clock signal on CLK of SD card which is 100 KHz, ‘cmdo’ and ‘DATo’ are the lines on which data is sent from the host to SD card, ‘cmdi’ and ‘DATi’ are the lines on which data is received from the SD card. The 48 bit response for CMD3 which consists of Relative Card Address is received on ‘cmdi’ line and is found to be 0x8000.

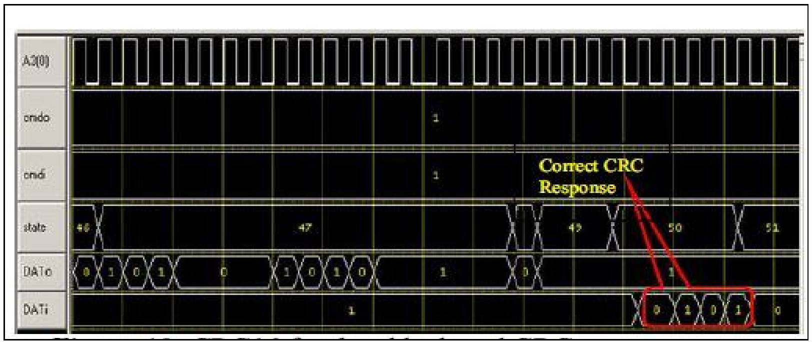

Figure 5: CRC16 for data block and CRC status response

5. Conclusion

A complete hardware based bidirectional access to the SD card design was implemented successfully without the need of any microcontroller or general purpose microprocessors. The design includes four modules single block write, multiple block write, single block read and multiple block read operation. Also, two other

Modules are designed, in which the SRAM is interfaced to the SD memory card. These two modules are single block write and multiple block write from the SRAM to the SD card. The data in the SD card is stored in the form of Windows wav file format. The single and multiple write operation implementations were verified using TLA611 Logic Analyzer, hexcmp2 software and also tracing the wav file from the SD card using MATLAB wave read function. Also the SRAM data storage mechanism is not same as that of the SD card, but the design can be altered as per the applications requirement. Overall a write operation for 5000 blocks can be done in 1 second and the same 5000 blocks can be read from the SD card in 1.051 seconds. The maximum achieved bandwidth utilization for write operation achieved is 91.65%. Finally, the designed FPGA based data concentrator can serve real time operations with a rate up to 25 Mb/s.

6. References

1. Mohammed Abdullah, Omar Elkeelany, and Ali Alouani, “An Efficient Embedded System Design for capturing and Storing Analog Data,” Journal of Engineering and Applied Sciences 2 (8), pp. 1290-1296, 2007.
2. Mohammed Abdallah, and Omar Elkeelany, “Simultaneous Multi-channel Data Acquisition and Storing System,” ICC,pp.233-236, 2009 International Conference on Computing, Engineering and Information.
3. SanDisk SD card product manual, available: http://www.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.p df, accessed April 2010.
4. C. Lin and L. Dung. “A NAND Flash Memory Controller for SD/MMC Flash Memory Card,” IEEE Transactions on Magnetics, Vol. 43, 2007.
5. C. Li, Q. Wang, and L. Wang. “A high efficient Flash storage system for Two way Cable Modem,” in Proceedings of IEEE International Workshop on Computer and Information Technology, pp. 551-556, 2008.
6. O. Elkeelany and G. Vince. "Portable Analog Data Capture Using Custom Processing," in Proceedings of The IEEE 39th Southeastern Symposium on System Theory, pp. 120-123, 2007.
7. M. Abdallah and O. Elkeelany. “Simultaneous Multi-channel Data Acquisition and Storing System,” in proceedings of International Conference on Computing, Engineering and Information ICC’09, pp.233-236, 2009.
8. J. Kim, J. Hong, E. Cha, and T. Lee. “Development of ECG and BCG Measuring System on Moving wheelchair using CDMA Network,” in Proceedings of 6th International Conference on Information Technology Applications in Biomedicine, pp. 179-181, 2007.
9. M. Abdallah, O. Elkeelany, and A. Alouani. “An Efficient Embedded System Design for capturing and Storing Analog Data,” Journal of Engineering and Applied Sciences 2 (8), pp. 1290-1296, 2007.
10. D. Young, C. Zhen-Cheng, and S. Han. “Read/write SD Card based on Microcontroller MSP430,” Journal of Biomedical Engineering Research, 2004.

11. Y. Ming-Ji, L. Yuan, and S. Ji-Yu. “Interface Design between Secured Digital Card and Single Chip Microcomputer,” Journal Information Technology, 2008.
12. V. Todakar and O. Elkeelany. “On-chip Hardware Mechanism for storing data in SD card,” in Proceedings International Conference on Embedded systems (ICES), 2010