Data Archival to SD Card Via Hardware Description Language

Omar Elkeelany, *Senior Member, IEEE*, and Vivekanand S. Todakar

***Abstract*—The main objective of this letter is to present the design of an efficient, real-time data archival system to a secure digital flash memory card via reconfigurable hardware. The data access from the SD card is implemented completely using Verilog and hence, there is no use of any microcontroller or on-chip general purpose processors. And since the complete design is a single-purpose system, no extra hardware is required. 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. We show how the bidirectional access takes place correctly and the data integrity has been verified using cyclic redundancy code in both field-programmable gate array (FPGA) chip and the SD card controller.**

***Index Terms*—Flash memory read/write, secured digital (SD) protocol, Verilog HDL.**

I. Introduction and Related Work

NONVOLATILE 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 [1]. 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 [1]. Also the stored data in flash memory can be easily transferred to a personal computer whenever needed since it is designed for portable use.

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 [2]. 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 main flash memory controller is the one which is present on the memory chip itself. This controller is designed to handle bit errors, bad blocks, maintain the high data access speed, etc. The flash memory access controller is the one which communicates with flash memory controller. An improvement on the flash memory controller designed in [3] 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. This is why we chose a reconfigurable approach to design a custom flash memory controller.

A software based flash access technique is used in many applications and one among them is for designing a storage system for two-way cable modem. In [4] the author clearly mentions that a Godson-1 CPU was used and flash access processes were scheduled by the way of interrupts instead of continuous queries. But the software access is less efficient as compared to hardware based systems. Another technique introduced by Vince *et al.* [5] and later improved by Abdallah *et al.* [1] using system-on-a-chip (SOC) technology to achieve the same. The main drawback in [5] is the use of a processor on the field-programmable gate array (FPGA) chip and the use of a high level language to program the flash controller. The high level languages, such as C are slower in execution as compared to Verilog and VHDL. The problem with the SOC technology based on general purpose processors (GPP) is that embeds a processor on a chip. Such complex system requires more hardware, power and hence high cost. Also the systems in [1], [5] did not perform in real-time.

Another related work is in [6] (but for MMC card not SD) [6] uses a synchronous peripheral interface (SPI) bus. This design uses Freescale DSP 56858 platform, but with an I/O speed lower than real-time. An SD card based SPI controller is studied in [7]. In [7], the interface between AT89C51SNDC microcontroller and SD card as well as the program flows chart are given. In the related work of [1], [4]–[7], flash memory access is 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. Obviously, this wastes processor's time, and consumes more power.

II. 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. A design choice made in the proposed system is to use detachable flash memory devices to allow data analysis at remote locations.

Manuscript received July 13, 2011; accepted September 07, 2011. Date of publication September 19, 2011; date of current version December 21, 2011. This manuscript was recommended for publication by G. Qu.

The authors are with the Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN 38505 USA (e-mail: OElkeelany@tntech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LES.2011.2168804

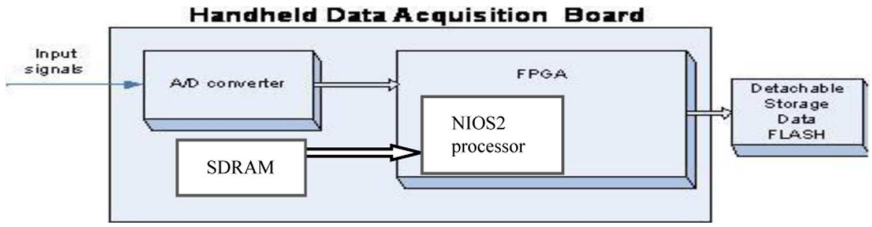

Fig. 1. Processor on chip flash memory controller [1].

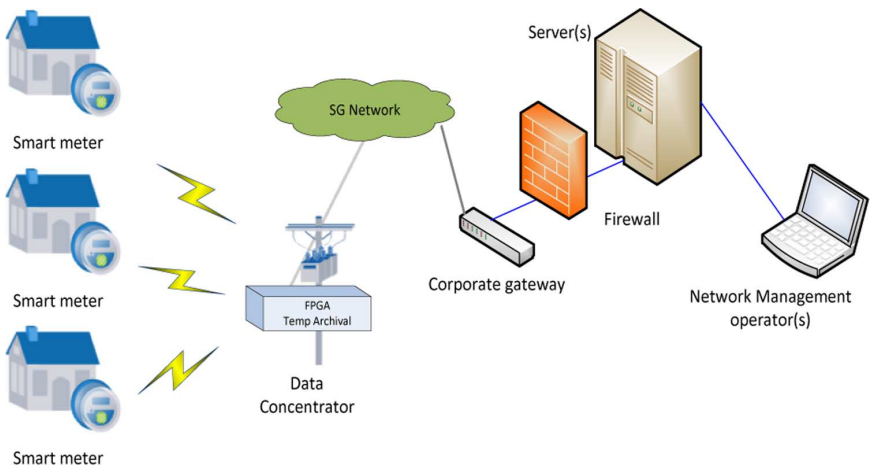

Fig. 2. Proposed FPGA based data concentrator.

A. Multiple Clock Synchronization

One serious challenge of GPP architectures can be seen from examining technical details of an SD card manufacturer. The challenge is that SD card manufacturers (such as SanDisk) specify two different clock frequencies; initialization mode (100–400 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 [1], the SD card cannot be accessed at a frequency more than 25 MHz and, hence, if dsPIC 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. In a microcontroller design implementation using ATmega32 microcontroller [8], the authors 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 low clock frequency of the processor, which yields an access time for a single block (of 512 bytes) of 4.15 ms.

Such available GPP based 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. Both of these approaches waste processing power and power consumption. On the other hand, if a FPGA is used to implement such requirement, one can design a precise finite state machine (FSM) with two independent clocks. Another aspect is that FPGA chips are reconfigurable. This is a crucial aspect in hardware, since it might be necessary to modify the functionality in the future without replacing the hardware already deployed.

B. Design Flexibility

However, designing a custom hardware via FSM will tend to lack the flexibility offered by GPP approaches mostly used in industry. This lack of flexibility can be reduced via the use of reconfigurable chip technologies.

III. Proposed System Design

A typical PC based network management for the smart grid (SG) utilizes a computer and 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.

In an earlier work of the authors, a Processor-on-FPGA-chip Handheld Data Acquisition System [1] was developed. As shown in Fig. 1, a Nios II processor was placed on the FPGA chip to carry out the data acquisition and archival processes [1]. The on-chip processor and an external SDRAM, are not required in the proposed system (in this letter) as shown in Fig. 2. 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.

In the proposed system, the flash memory controller will be designed to copy the collected data temporary stored in the FPGA internal buffers. It is advantageous to use the FPGA's 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.

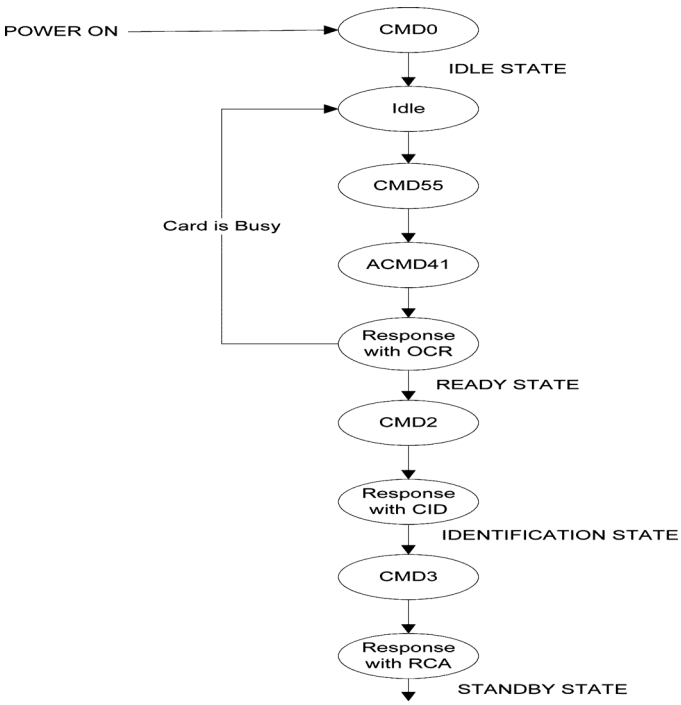

Fig. 3. FSM for SD card identification.

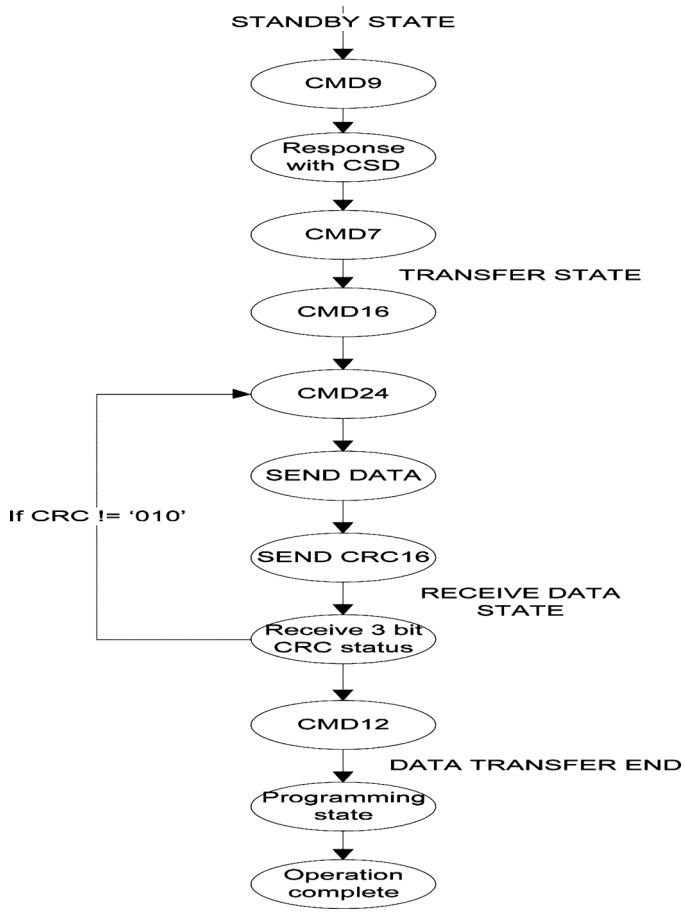

Fig. 4. FSM for single block write.

In brief, the FPGA will record acquired data after filling its internal storage buffers in nonreal 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.

A complete Hardware architecture is developed using Verilog Hardware description language (HDL). All the critical and noncritical tasks are implemented in hardware only. There is no use of any embedded C programming language. To achieve our goal a single-block 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 and retransmitted.

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 FSM for SD card identification is shown in Fig. 3. 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.

Fig. 4 shows the FSM designed in this research to write one block of data to the SD card. Here the host selects one card for data write operation by CMD7. Using CMD16 host sets the valid block length for 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 no 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.

Multiple-block operations differ from single-block operations in the fact that a single command can be used followed by a bulk of data transmissions. This can improve data throughput.

IV. Experimental Results

The design is implemented on the Altera's Cyclone II EP2C35F672C6 FPGA chip using a SanDisk SD card. The closest existing comparable system takes approximately 60 s to write 5000 blocks of data in the same SD card, where in each block is of 512 bytes. And as per the testing and verification results the proposed system utilizes the full bandwidth provided by the SD card technology and it can write the same 5000 blocks of data in 0.85 seconds. Similarly a read operation

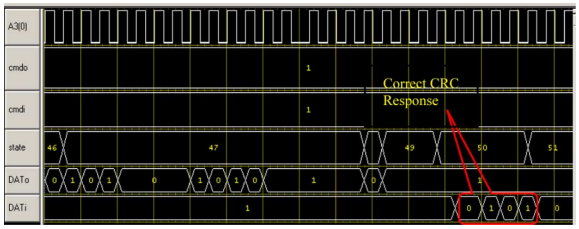

Fig. 5. CRC16 for data block and CRC status response.

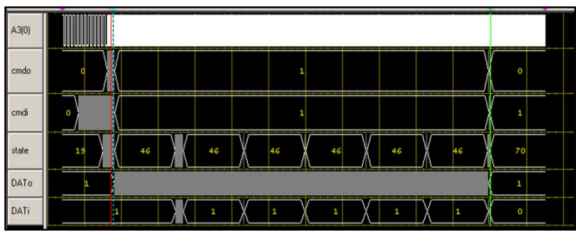

Fig. 6. Timings of multiple-block write.

TABLE I
ACHIEVED

Operation	Time taken
SD Card Initialization	63.88 ms
Single block Write (1 block)	318.88 µs
Multiple Block Write Operation (6 concentrated blocks)	1.02274 ms
Single block Read (1 block)	481 µs
Multiple Block Read (3 blocks)	816 µs

of 5000 blocks from the same SD card is done in 1.051 s. The evaluated maximum data concentration rate is 25 Mbps.

The design implementation results are verified on a logic analyzer and the results for various commands are shown in Figs. 5 and 6. 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 CRC check code is sent after two clock cycles at the end of data transfer which is as shown in Fig. 6. Once a proper CRC status is received the host sends next block of data to be written on the card or sends stop transmission command CMD12 to terminate the data flow. After the completion of data write the SD card is tested for its contents using MATLAB and found to have the same number of values and data as the authors intended to write.

After all timing simulations were verified, successful read and write operations were verified by examining the SD card contents and comparing them to a known set of inputs. Table I shows a summary of evaluations results for SD card Initialization, single and multiple block write and single and multiple block read.

Since the designed system is hardware based, and is implemented on a different chip other than the surrounding environment, (such as data acquisition chip), it is possible for the data host such as data acquisition chip) to operate concurrently with the designed chip. It is well known that only when the acquisition (i.e., data arrival rate) and archival (i.e., data service rate) operate constantly, real-time operations can be provided.

Since the block size was 512 bytes, it is possible for the designed FPGA data concentrator to acquire six blocks of data at a rate R of 25 Mb/s.

$$R = \frac{6\ (\text{blocks}) * 512\ \text{bytes} * 8\ \text{bits each byte}}{1.02274^{-3}\text{seconds}}.$$

V. Conclusion and Future Work

A bidirectional hardware based core for the SD card design was implemented in reconfigurable FPGA based data concentrator chip. Overall a write operation for 5000 blocks can be done in 1 s and the same 5000 blocks can be read from the SD card in 1.051 s. In an earlier work of the authors, the same block size needed 60 s. This difference is mainly due to the use of custom finite state machine design as compared to software based solutions found in related work. Simulation results show real-time data rates up to 25 Mb/s can be reached.

This letter presents a real-time design of data archival to the SD card that can be used in data concentrator applications for remote sites such as in SG.

Scalability issues such as the impact of larger size SD cards, and SD cards from various manufacturers will be addressed in future work.

References

[1] M. Abdallah and O. Elkeelany, "Simultaneous multi-channel data acquisition and storing system," in *Proc. Int. Conf. Comput., Eng. Inform. (ICC'09)*, 2009, pp. 233–236.

[2] SanDisk SD Card Product Manual [Online]. Available: http://www.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf Apr. 2010, available, accessed

[3] C. Lin and L. Dung, "A NAND flash memory controller for SD/MMC flash memory card," *IEEE Trans. Magnet.*, vol. 43, no. 2, pt. 2, Feb. 2007.

[4] C. Li, Q. Wang, and L. Wang, "A high efficient flash storage system for two way cable modem," in *Proc. IEEE Int. Workshop Comput. Inform. Technol.*, Sydney, Australia, 2008, pp. 551–556.

[5] O. Elkeelany and G. Vince, "Portable analog data capture using custom processing," in *Proc. IEEE 39th Southeastern Symp. Syst. Theory*, Macon, GA, 2007, pp. 120–123.

[6] D. Young, C. Zheng-Cheng, and S. Han, "Read/write SD card based on microcontroller MSP430," *J. Biomed. Eng. Res.*, 2004.

[7] Y. Ming-Ji, L. Yuan, and S. J. Yu, "Interface design between secured digital card and single chip microcomputer," *J. Inform. Technol.*, 2008.

[8] H. Kuan and Y. Lu, "Nonvolatile semiconductor memory revolutionizing information storage," *IEEE Nanotechnol. Mag.*, pp. 4–9, Dec. 2009.