International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 4, Issue 6, June-2018

Design and Implementation of Custom Logic for utilizing DDR 2 Memory Controller on FPGA for High Speed Data Storage Application

Rikin J Nayak[1]*, Jaiminkumar B Chavda[2], Kishan Patel[3]
[1]*Assistant Professor, E&C Department,* [2]*Assistant Professor, IT Department,* [3]*Assistant Professor, EE Department,*
[1,3] *Chandubhai S. Patel Institute of Technology, Charotar University of Science & Technology, Changa, Gujarat,*
[2]*L.D. Collge of Engineering, Ahmedabad*
* *rikinnayak.ec@charusat.ac.in*

Abstract— In the recent development semiconductor technology and availability of microprocessors with rich instruction sets, FPGAs and other such computing platforms, many complex processing tasks becomes more easy, faster and accurate. With rich processing capabilities, it requires to have platform where large amount of data can be buffered for faster processing by high performance computing platforms. For optimized High performance, computing platform requires compatible memory interface, which enables the processing unit (microprocessor / FPGA) to communicate for data readout and write back at relatively faster speed which matches the processing speed. For storage, many different storage options are available like SRAM, DDR RAM, SDCARD, Flash Memory etc. Each one them has their own throughput for reading-writing. In this paper, A DDR2 memory controller is custom configured for High-speed data storage using Xilinx ISE tool and Virtex 5 FPGA as a computing platform. Detailed results are discussed in paper.

Keywords— FPGA, DDR2, Memory Controller, High Speed Data, RS232 protocol.

I. INTRODUCTION

In the modern computing era, to address the end-user demands for diverse nature of applications, scalable computing platforms have became prime focus to meet custom requirements and specification. These includes issues to be addressed are like power consumption, computing capabilities using FPU and SIMD instruction sets, support for memory interfaces, and programming interface etc. With considering all diverse set of design requirements and constraints, system designer has to select specific device and design based on the application specific performance requirements. In the recent times, due development of advanced processors with rich instruction set and other computing systems along support of high-speed interfaces for data communication and storage, new range of applications has emerged, which is being on prime focus and are being developed and deployed. In the span of recent years, there has been increased clock frequency of the processor but the memory operating clock frequency has not increased at same rate as microprocessor This scenario provides the bottleneck for the range of applications which are data centric (The applications which are heavily dependent on the data read write transactions for processing data items for specific algorithm, where the computer architecture could not accelerate the performance beyond certain limits due to timing and synchronization). These bottlenecks serve the research opportunity for the developing designs for optimized performance. For storing data at the computing platform, various options of memory modules are available to provide suitable selection of memory. Based on the performance qualification criteria (memory throughput, size, data transfer unit, memory response time, etc) designer can suitable memory. For high speed storage on-chip memory serves as one of the best option as in case of processor and internal flash memory or RAM in case of FPGA, which could be Block memory or Distributed memory. However on-chip memories are not available in large size due to high cost (semiconductor design and fabrication). For large volume of data, COTS (commercially off the self) external memories available are SRAM, SDRAM, PROM, Compact flash, DDR Memory (DDR2, DDR3, DDR4, LPDDR, LPDDR2 [1] etc).

In case of Dynamic RAM requires refreshing due to capacitors for holding charge (state) as information, while Static RAM having flip-flops for holding the state information. This architecture makes Static RAM much faster than Dynamic RAM. However, due to flip-flops used in static RAM physical size of the RAM is larger than dynamic RAM and Static RAM are available in comparative low capacity, while Dynamic RAM comes with higher data storage capacity. For any digital system, to read-write the data to and from the storage remains one of the primary operations. With rapid increases in computational resources, and high speed processing, compatible high speed memory with large capacity becomes primary requisite. In this paper, the design and implementation of DDR2 memory controller on Virtex 5 platform for storing the data is discussed in brief along with preliminary results.

II. DDR MEMORY AND ITS ADVANCEMENT

With advances in semiconductor fabrication technologies have resulted into achievement of memory modules which have faster speed, high memory throughput, low power consumption and, very small size to large size memories for commercial markets. In dual data rate (DDR) memory, two words of data are transferred per one clock cycle.

In DDR1 is a first generation DDR RAM which allows double words to be transferred on rising edge and falling edge. In first generation DDR, prefetch buffer is of 2 bit, which is double of SDRAM. Its transferred rate is around 266 - 400 MT/s [6]. DDR2 is the second generation DDR memory. DDR2 is having prefetch buffer of 4-bit, which means its transferred rate is double than the DDR1. Its transferred speed around 533 - 800 MT/s. DDR3 is third generation DDR memory having prefetch buffer is of 8-bit and its transfer speed is around 1066-1600 MT/s. DDR4's transfer rate is 2133-3200 MT/s. Operating voltage requirement for DDR is 2.5V, DDR2 is 1.8V, DDR3 is 1.5V and DDR4 is 1.2V. DDR memory has fast speed; however with increasing clock rate has resulted in increased power consumption, so it requires balancing as per the application. Author in [5] have discussed about streaming memory controller for utilize external memory on a network for increasing the speed of processing.

III. DDR2 ARCHITECTURE

In DDR 2 SDRAM, it consists of different banks, where each bank contains number of rows and columns as shown in Fig 1 [2].

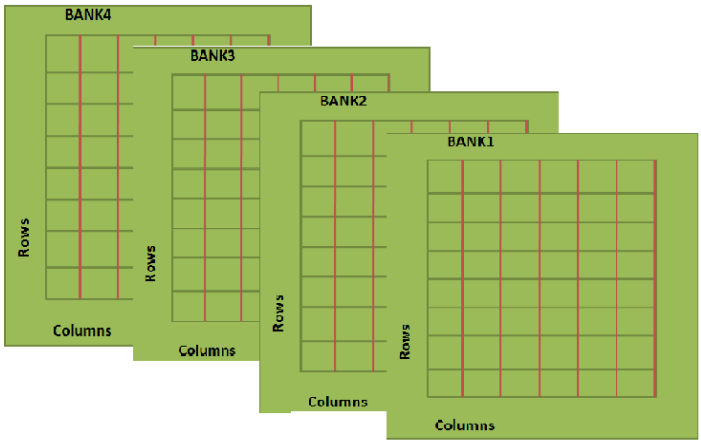

Figure 1 SDRAM structure

In Dual in line Module (DIMM) more than one module are there [3]. In SDRAM interface address lines, data lines and control lines are there. For writing/reading operation to/from DDR, appropriate row and column address is selected. For DDR2 with 4N Prefetch buffer for each address there will be 4 data words will be written into the DDR. Structure of the Bank, Row and Columns are as below.

Based on module density and word size, numbers of ROW, COLUMN and BANK SIZE are required. For example DDR2 with 128 MB size and with 64 bit word size its memory is organized as

BANK: BA0, BA1 (4) **ROW**: A0-A12 (8K) **COLUMN**: A0 - A8 (512)

For each bank, row and column DDR will have space of 64 bit so total amount of memory can be found by No of Bank pair (4) x Size of Row (2^13) x Size of Column (2 ^ 9) x Word Length (64) = 1073741824 bit = 128 MB. For accessing particular space /page in memory appropriate BANK, ROW and COLUMN is get selected and a combination of all three values generates the physical location of memory address.

IV. DDR Address generator

In DDR2 memory controller user generates address after selecting fixed bank, row and column. In MIG, common address FIFO is used for both read and writes data. Address FIFO is having memory address in terms of ROW address, COLUMN address, BANK address and dynamic commands, which user will generate for specific operations like AUTO REFRESH, PRE-CHARGE, WRITE and READ. With address logic, RAS (Row Address Strobe) and CAS (Column Address Strobe) signals are used for the latching row and column address respectively. These signals are also used to initiate and terminate read and write operation. RAS is active low signal is it required to remain low until RAS is no longer needed. The minimum amount of time of RAS must be remain active is called tRAS , and minimum amount of time of RAS must be inactive is called RAS precharge time tCP. In case of CAS, minimum amount of time CAS must be remain active is called tCAS , and minimum amount of time of CAS must be inactive is called CAS precharge time tCP [4]. Status of the RAS and CAS determines row and column address on address bus.

V. Implementation

For testing DDR controller VIRTEX 5 XUPV 5 board is used as a hardware platform and Xilinx ISE is used for designing MIG Controller. Our design is based on [7]. Fig 2 shows block diagram of the system. Here, digital data pattern is generated on 12 differential LVDS channels of Spartan 6 FPGA board as data source. Spartan 6 FPGA also generates Clock and Strobe (line start indication) signal on LVDS channel. Here, source synchronous communication system is realized for test experiment. Detail scheme for the implementation is shown in fig 3. The scheme is designed as per our requirement of acquiring data from CMOS image sensor having 12 channel output. In this experiment, instead of data from detector, Spartan 6 FPGA is generating data on 12 channels. Digital data is sampled at each falling edge of the clock signal from source. After receiving 10 bit on each 12 channel, all 12 channel data rearrange as a array of 12 x 10 = 120 bits. Such two successive data frame forms 240 bits and storing that data at internal FIFO buffer on FPGA with FIFO of size 256x1024, along with 16bit dummy data.

Here DDR2 memory each column is of 64 bit word length. So with 2n prefatch it can write 256 bits per clock cycle. As one location of internal FIFO is of 256bits , 240 bits are of data along with remaining 16 bits are padded as “0” value. In DDR2 MIG, common address generator is generates address for both read and write transaction of DDR2. In DDR2, due to 2n pre-fetch, each time address gets incremented by 4 for writing/reading operation 4 words simultaneously.

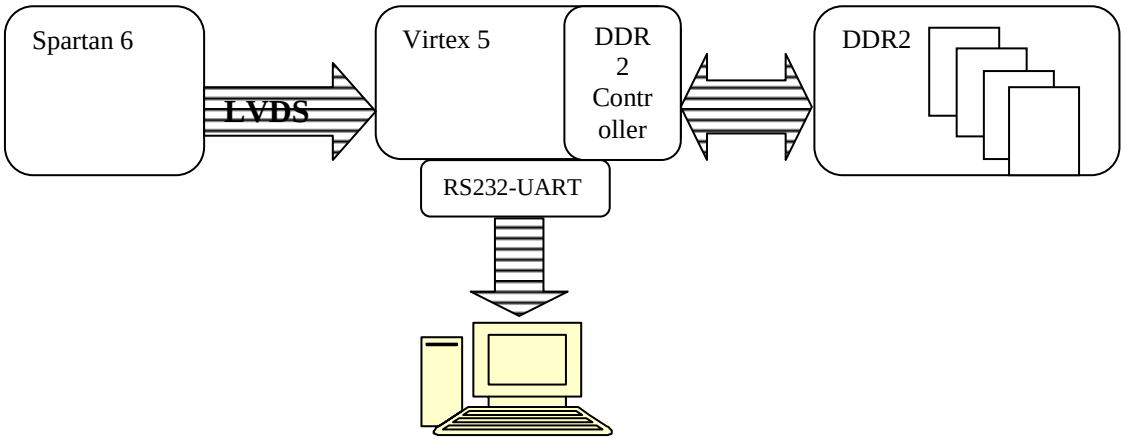

Figure 2 Experimental Setup: Data Generation to Data Readout with Buffering at DDR2

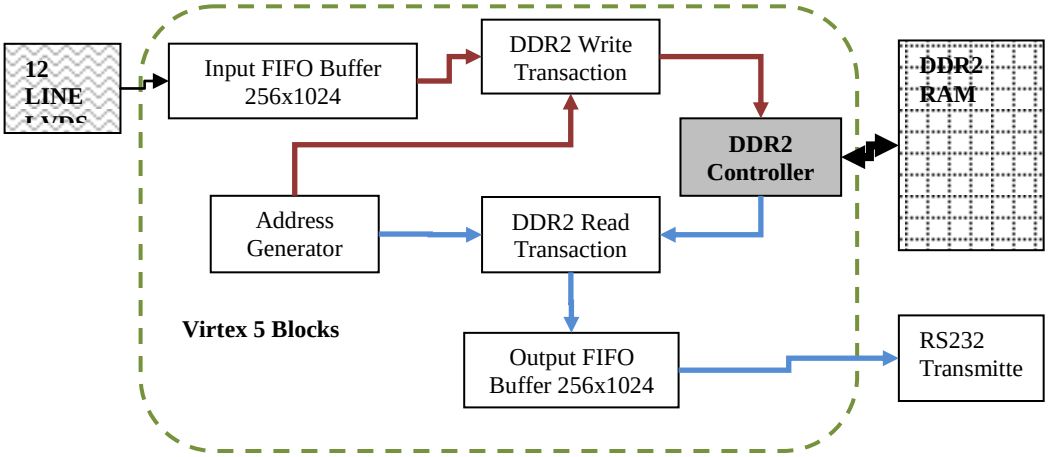

Figure 3 Internal DDR Controller scheme

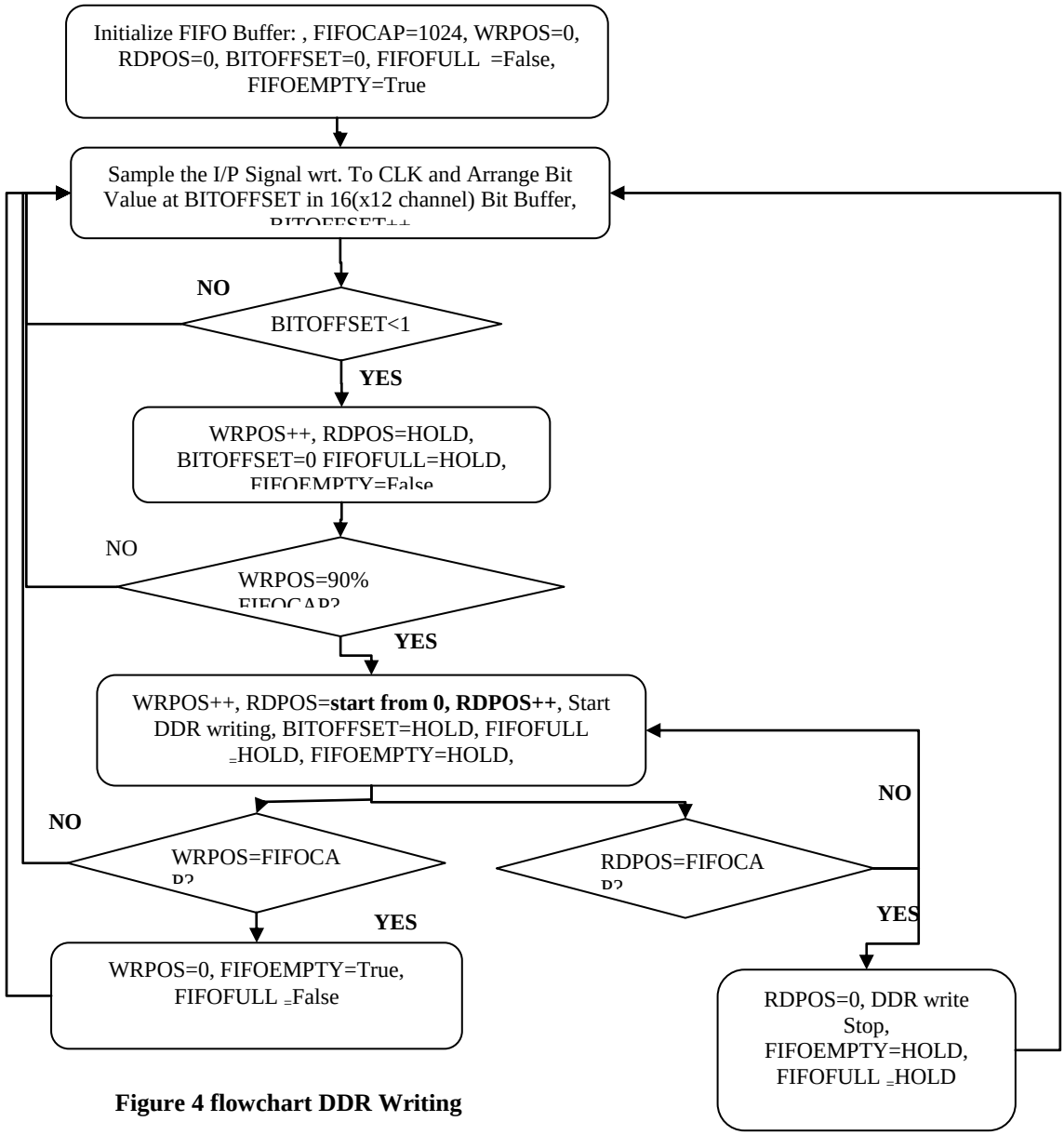

Figure 4 flowchart DDR Writing

In Fig 4, it shows flowchart for the system. Here, developed system is acquiring data from CMOS detector simulator developed for generating multichannel (12 channels) data, line start (STROBE) and CLOCK on FPGA. Since, FIFO size is of 256x1024. Initially WRPOS (write position of address) and RDPOS (Read position of address) is 0.Initially FIFO Full signal is 0 and FIFO empty signal is 1. After strobe will become high data sampling start and after 20 bit received from each channels, FPGA will rearrange those bits in to 1 dimension array of 240 bits + 16 bits (zeros) = 256 bits. After 16 bits receive from each channel it will write the word of 256 bits in to input FIFO at WRPOS and increments WRPOS by one.

These rearranging and writing will continuously repeat until WRPOS in FIFO reaches of 90 % of total capacity. As FIFO gets filled 90% capacity, now Read FIFO is initiated and it starts data writing using generated address in DDR2. When RDPOS reaches up to FIFOCAP (FIFO capacity) DDR Write gets finished writing from FIFO. This process of filling the FIFO, and writing to DDR continues till the data is received. After DDR Write finish fixed volume of data, DDR Controller initiates DDR Read and transfer the read data over UART-RS232. Fig 5 shows data received from UART.

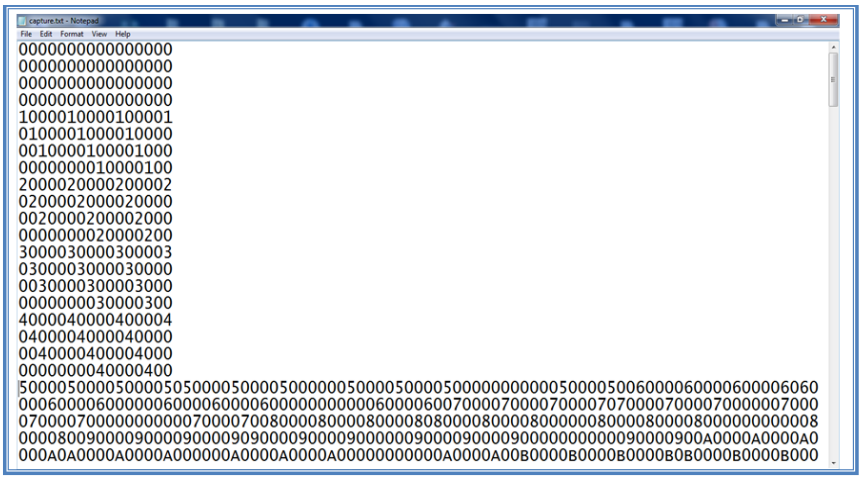

Figure 5 UART received data after DDR Read

VII. Verification

After writing fixed data size, the DDR2 Write process stops and the DDR_READ_START signal is generates (goes HIGH). Here, the data read is stored in intermediate FIFO_UART and send to UART at slow speed. Every time the FIFO_UART get emptied, the DDR_READ_START is issued until the all DDR data is getting read. Here, captured data from the UART is stored in HEX_DATA file on computer, which is parsed to a custom C program for formatting and verification of contents of HEX file. Data received on REALTERM terminal software, and software generates .txt file for the received data**.** In **Fig 6, s**hows data verification scheme. Each of the data is gets arranged according to particular location (64BIT Data). For simplicity each data values and location values are same, so one can easily check for possible errors.

VIII. Conclusion

Designing the controller configuration of DDR2 Memory Controller for the custom application provides the optimized performance. A custom configuration leads towards efficient usage of DDR2 memory for real time data centric applications likes video processing, cryptography and other where the stream of data is expected. In case of DDR memory increased in clock frequency results in increased power consumption, so as a design engineer, custom scheme of arrangement for data for efficient storage and retrieval leaves the opportunities for power aware computing. To optimize the performance of DDR read/write operation with dynamic input , dynamic DDR controller in which one can schedule the request run time is designed where DDR write, Read initiated and control by the external input. Here, during the experiment, 200 Mb of data from external source to DDR 2 memory and verified through transmit the same data (after DDR read) on UART-RS232. Data verification is carried out by reading the written text file using C and also verifying them by comparing data with corresponding address of the data.

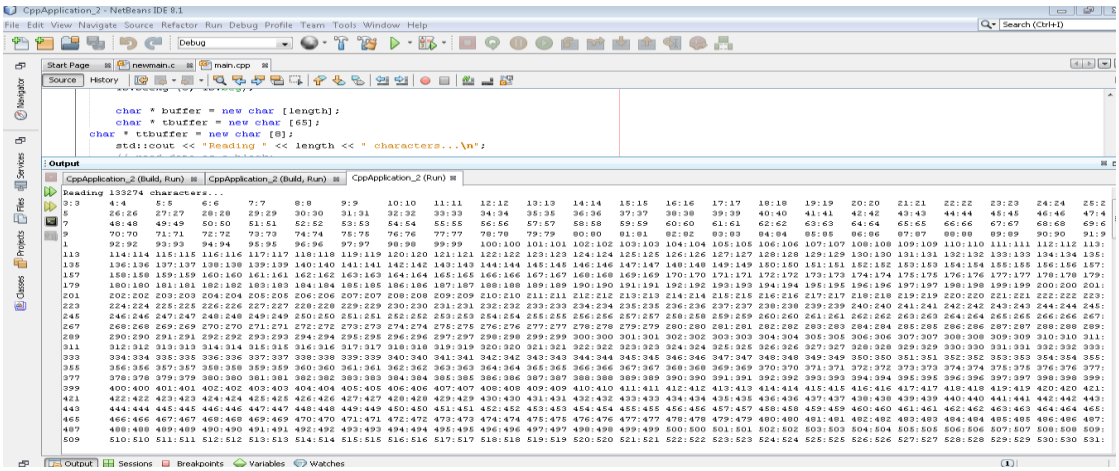

Figure 6 UART received Verified using C programming

References

1. An Improved and Simplified Interface. Protocol for DDR Memory. James Gay. Senior Member of the Technical Staff, Freescale Semiconductor, Inc. White Paper
2. B. Jacob et al., Memory systems: cache, DRAM, disk. Morgan Kaufmann Pub, 2007.
3. Li, Yonghui, Benny Akesson, and Kees Goossens. "Dynamic command scheduling for real-time memory controllers." Real-Time Systems (ECRTS), 2014 26th Euromicro Conference on. IEEE, 2014.
4. Understanding DRAM Operation Applications Note IBM.
5. Burchardt, E. Hekstra-Nowacka and A. Chauhan, "A real-time streaming memory controller," *Design, Automation and Test in Europe*, 2005, pp. 20-25 Vol. 3. doi: 10.1109/DATE.2005.34
6. www.transcend-info.com
7. UG086 Memory Interface Solutions User Guide