ISSN: 2366-1313

MEMORY CONTROLLER FOR DDR SDRAM AND IT'S VERIFICATION USING VERILOG FOR DSP PROCESSORS

[1]P.Chenchu Navya Lalithya,[2]K.Dhanumjaya
[1]PG Scholar,[2]Professor,Dept. of E.C.E
[1,2]Audisankara College of Engineering and Technology(Autonomous),Gudur,
SPSR Nellore(dt.) A.P

ABSTRACT

Synchronous DRAM (SDRAM) has become a mainstream memory of choice in design due to its speed, burst access and pipeline features. For high-end applications using processors, the interface to the SDRAM is supported by the processor's built-in peripheral module. However, for other applications, the system designer must design a controller to provide proper commands for SDRAM initialization, read/write accesses and memory refresh. DDR SDRAM uses double data rate architecture to achieve high-speed data transfers. DDR SDRAM (referred to as DDR) transfers data on both the rising and falling edge of the clock. This DDR controller is typically implemented in a system between the DDR and the Processor. In this paper, the implementation has been done in Verilog by using Xilinx ISE 14.5

INTRODUCTION

1.1 Overview

Image processing data pipelines require a large amount of data to be buffered. Memory is the place where integrated circuits (ICs) hold current data for the pipelines. Several types of memory could be used for this purpose, yet each has their pros and cons. Static RAMs (SRAMs) are fast and have reduced power consumption, but their low density - several transistors are required to store a single bit of data - increases their cost and limits their use.

On the other hand, Dynamic RAMs (DRAMs) have slower performance and higher power consumption, but they are denser and cheaper than SRAMs. These characteristics make them a better choice for the main memory in digital imaging systems. There are several types of DRAM. Previous types of DRAM include Fast Page Mode DRAM (FPM DRAM) and Extended Data Out DRAM (EDO DRAM) that are asynchronous. These are quite slow for current systems. Synchronous DRAM (SDRAM), Double-Data Rate (DDR)

SDRAM, and RAMBus DRAM (RDRAM) are currently mainstream in the PC industry

SDRAM is synchronous, meaning that the device directly depends on the clock speed driving the device. With SDRAMs, latency is significantly reduced when compared to previous DRAM technologies. Inputs and outputs are simplified in terms of signal interfacing. SDRAM is still used but is being quickly replaced by DDR SDRAM.

RDRAM narrows the memory bus to 16-bit, and uses a faster bus operation (up to 800MHz). Although the bandwidth is incremented respect to SDRAM, it has a big drawback: it is a proprietary technology.

DDR SDRAM is a natural evolution of SDRAM. Data transactions are done in both edges of the clock, thus doubling the raw bandwidth of the data path. Performance is improved as with RDRAM but costs are lower. Due to widespread adoption by the PC industry and improved long-term availability over SDRAM, it makes a good choice for imaging applications

With the advent of modern process technologies (13µm and beyond), FPGAs are being considered for high-volume applications. FPGAs have evolved in terms of complexity and density and they are now not very far from ASIC performance and capacity. Therefore they can now be used in demanding applications that were not possible before. For instance, most of Xilinx and Altera FPGAs include enough features to implement a high-speed memory controller design. FPGAs are now supporting the features needed to interface a DDR SDRAM. FPGA I/Os are now compatible with the SSTL-II electrical interface, I/Os include DDR registers, and phase locked loops (PLLs) are flexible enough to allow different clock frequencies and meet data capture timings. In some Altera families, hard-macro delay chains for data strobes have been added, a feature reserved to special ASIC cores [1].

The FPGA based intellectual propriety (IP) cores market offers memory controllers designed to make the most out of these features. In conclusion, FPGAs have improved enough to implement DDR interfacing both for products and prototypes. Modern FPGA families, combined with IP cores, provide features for achieving a bandwidth comparable to ASIC designs. The goal and novelty of this paper is to compile the design techniques to implement high-speed memory controllers. In this paper we summarize the features to be taken into account when choosing an FPGA for high-speed DDR SDRAM interfacing, we depict a memory controller data path architecture

that can be implemented with Altera and Xilinx FPGAs and we explain the design rules to meet timing requirements. We show timing measurements of a Virtex-II FPGA interfacing a Samsung K4H560838C DDR SDRAM device at 60MHz. We perform the read data capture timing analysis for a Stratix FPGA interfacing a Samsung M470L6524BT0 DDR SDRAM device at 80MHz.

Single data rate (SDR) SDRAM (Synchronous Dynamic Random Access Memory) drives/latches the data and command information on the rising edge of the synchronous clock. DDR SDRAM is a type of SDRAM that inherits technologies from SDR SDRAM and realizes faster operation and lower power consumption. DDR SDRAM most commonly used in various embedded application like signal processing, networking, image/video processing etc which require cheap and fast memory. DDR SDRAM was the initial development of SDRAM memory technology , to improve its performance. DDR SDRAM utilises various techniques including very tight timing controls , to increase the data transfer rates by almost a fact of two[1][2]. The very tight timing requirements often require the use of phase locked loops and self-calibration techniques to ensure the timing is sufficiently accurate

DDR SDRAM achieves a data transfer rate that is twice the clock frequency by employing 2-n bit prefetch architecture [3]. In this architecture, 2n bits of data are transferred from the memory cell array to the I/O buffer every clock. Data transferred to the I/O buffer is output of n bits at a time , for every half clock (both rising and falling edges of the clock (CK). The proposed memory controller and CoreDDRis used to interface memory to rest of the embedded system design. CoreDDR provides a high-performance interface to double data rate (DDR). CoreDDR accepts read and write commands using the simple local bus interface and translates this request to the command sequences required by SDRAM devices. It also performs all initialization and refresh functions using memory controller. This Memory controller design has been implemented in Verilog. The focus of this work is to design and implement DDR SDRAM controller which provides memory interface between the DDR SDRAM memory module and main embedded system

LITERATURE SURVEY

2.1 Survey on Storage Architectures

ISSN: 2366-1313

The Literature survey started with understanding of the various types of storage architectures for very high volume of data, and continued with survey of storage elements that can store very huge data which are of the order of Terabytes or Zettabytes. Then it was necessary to select a memory controller which controls the various functions of the memory element

Sang-Woo Jun_, Ming Liu et.al[1], has implemented a novel high-performance storage architecture, which is named as BlueDBM (Blue Database Machine). This architecture can be used for scalable distributed flash and this aims at achieving high performance, high capacity storage element which allow random access to the flash. Through this design throughput achieved is high as large number of flash chips are shared which has low latency. The authors have designed flash controllers which manages the chip to chip network with the help of Ethernet. The controller is placed between the storage element i.e., flash and the host this leads to hardware acceleration so latency gets directly reduced. The interface used for data transfer is PCIe. The designed storage architecture was implemented for Big Data applications where a communication link was established directly between nodes through flash controllers. The flash controllers accelerates the system. The Blue DBM consists of nodes which has storage resources and acceleration to access data is through reconfigurable fabric. Each node is connected through inter-FPGAnetwork which improves the overall performance. The makers have developed a little scale model whose system data transmission scales straightforwardly with the quantity of hubs, and where normal dormancy for client programming to get to glimmer store is under 70µs including 3.5µs of system overhead.

Duncan.G. Elliott et.al [5] have developed an architecture called Computational RAM. It is a processer in memory architecture via bandwidth is strongly made use in the internal memory . In the computational RAM design's customizing model, a host CPU can read and keep in touch with any memory area amid an outside memory cycle. Amid a work cycle, all handling components execute the same basic guideline also, alternatively get to the same memory balance inside their private memory allotments. By the day's end, computational RAM is a SIMD processor with circled, nonshared, reliably tended to memory. A vehicle energizes interprocessor correspondence and is useful for

ISSN: 2366-1313

combinational operations. Moreover, a straight interconnect that widens to two estimations at the terminations of lines is important for 1D likewise, 2D nearest neighbor operations. Computational RAM (furthermore suggested as C•RAM) which can work either as a standard memory chip or as a SIMD (single-rule stream, diverse data stream) PC. Right when used as a memory, computational RAM is forceful with customary DRAM to the extent access time, packaging, and cost.

Kermin Fleming et.al.[6] presents Customarily, equipment outlines divided over numerous FPGAs have had low execution because of the wastefulness of keeping up by every cycle timing in between discrete FPGAs. In this thesis, the authors have presented a system by which complex plans might be productively furthermore, naturally parceled among numerous FPGAs utilizing expressly modified inertness obtuse connections. They portray the programmed amalgamation of a range proficient, elite system for directing these between FPGA joins. By mapping a different arrangement of vast research models onto a different FPGA stage, we illustrate that our device acquires critical additions in outline attainability, accumulation time, and even divider clock execution. There are two issues in combining a between FPGA interchanges system for idleness uncaring connections: execution and correctness. Partitioning plans at inertness obtuse FIFOs permits us to transport just expressly enqueued information. Be that as it may, to accomplish elite a system must be capable adventure the pipeline parallelism intrinsic in the parceled design. However, because numerous connections can cross between FPGAs, there is a need to multiplex the physical connections between the FPGAs. Here the authors have introduced a dialect expansion and compiler that influences inactivity heartless outline to deliver elite executions traversing various FPGAs. Our dialect and compiler license us to construct bigger exploration models, enhance arrangement time, and, now and again, pick up execution over single FPGA executions. The compiler performs best in parceling computerized signal handling applications. These applications for the most part highlight high data transfer capacity, calculation necessities. Thus they are stronger to the inertness presented in chip-to-chip correspondence and have the potential for super-direct execution increments when scaling to systems with numerous FPGAs. Applications with bigger measures of input, similar to processor

ISSN: 2366-1313

models, may encounter execution debasements in respect to a single FPGA because of dormancy. Be that as it may, these applications still advantage from enhanced access to assets, outline scaling, and decreased arrange times.

PROPOSED METHOD:

Double Data Rate Synchronous Dynamic Random-Access Memory is the original form of DDR SDRAM. It is just like SDRAM except that is has higher bandwidth, meaning greater speed. Maximum transfer rate to L2 cache is approximately 1,064 MPs (for DDR SDRAM 133 MHZ). DDR RAM is clock doubled version of SDRAM, which is replacing SDRAM during 2001- 2002. It allows transactions on both the rising and falling edges of the clock cycle.The Control Interface module of SDRAM approaches summons and related memory addresses from the host and translating the charges and passing the look to the Command module. The Command module acknowledges summons and address from the Control Interface module for creating the correct orders to the SDRAM. The Data Path module handles the information way activities amid WRITE and READ orders. The best level module additionally instantiates two PPL's that are utilized as a part of CLOCK_LOCKS.DDR SDRAM Control Interface Module The control Interface Module deciphers and registers charges from the host and tracks the decoded NOP, WRITEA, READA, REFRESH, PRECHARGE, and LOAD_MODE summons, alongside ADDR, to the Command Module. The LOAD_REG1 and LOAD_REG2 summons are decoded and utilized intramural to load the REG1 and REG2 registers with the qualities from ADDR.

The REFRESH_REQ yield is cited with the counter achieves zero and remains cited until the point that the Command Module recognizes the demand. The recognize from the Command Module attaches the down counter to be reloaded with REG2 and the procedure rehashes. REG2 is a16-bit esteem that speaks to the period between REFRESH summons that the controller issues. The esteem is set by the condition int (REF_PERIOD/CLK_PERIOD).DDR SDRAM Command Module The summon module gets decoded orders from the Control Interface Module, alongside invigorate demands from the revive control rationale and effectuates the suitable orders to the SDRAM. The module includes a basic

ISSN: 2366-1313

referee that referees between the orders from the host interface and the invigorate demands from the revive control rationale.DDR SDRAM Data Path Module One of the most difficult aspects of DDR SDRAM controller design is to transmit and capture data at double data rate. This module transmits data to the memories. The basic function of data path module is storing the write data and calculates the value for read data path.The Data Path Module issues the SDRAM information interface to the host. Host information is acknowledged on port DATAIN for WRITEA charges and information is given to the host on port DATAOUT amid READA commands. The information path width into the controller is twice the information way width to the DDR SDRAM gadgets. The information path module's DATAIN and DATAOUT ports are settled at 32bits and the DQ port is settled at 16bits. To construct data paths bigger than 32bits, Data Path Modules can be fell to expand the information transport width for increases of32bits.

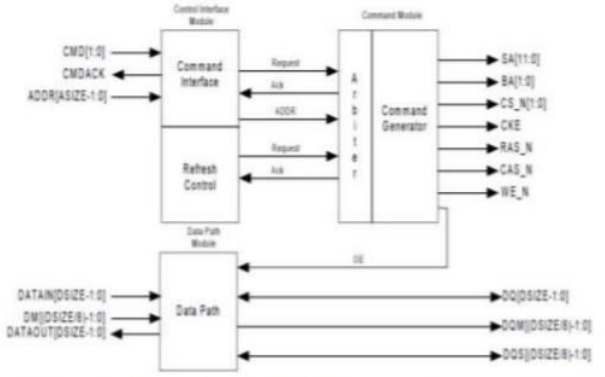
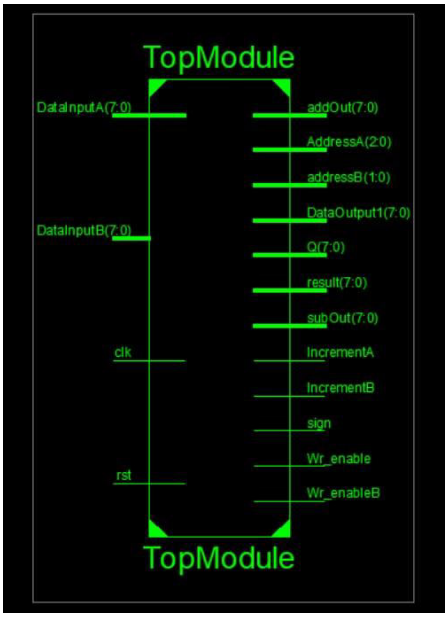
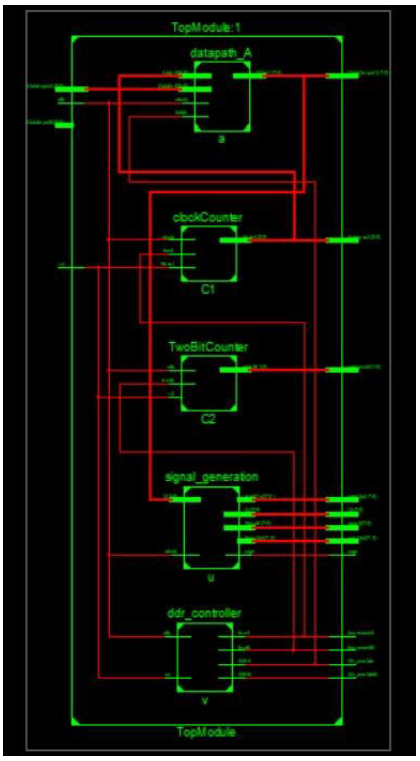


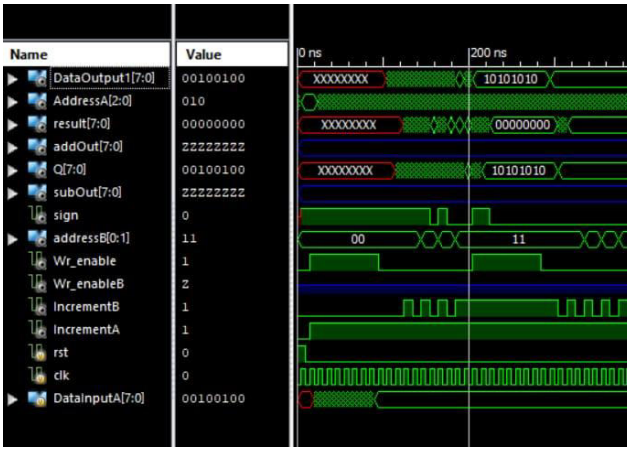
Figure 1: Functional Block Diagram of proposed DDR SDRAM Memory Controller core.

RESULTS:



ISSN: 2366-1313





CONCLUSION:

An efficient fully functional DDR SDRAM controller is designed and verified by using Verilog HDL. The controller generates different types of timing and control signals, which synchronizes the timing and control the flow of operation. The memory system operates at double the frequency of processor, without affecting the performance. Thus, we can reduce the data bus size. The drawback of this controller is complex schematic with large number of buffers in the circuit increases the amount of delay.

REFERENCES

[1] R. C. Baumann, “Soft errors in advanced computer systems,” IEEE Design Test. Comput., vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[2] N. P. Rao and M. P. Desai, “A detailed characterization of errors in logic circuits due to single-event transients,” in Proc. Euromicro Conf. Digit. Syst. Design, Funchal, Portugal, 2015, pp. 714–721.

[3] B. Narasimham et al., “Characterization of digital single event transient pulse-widths in 130-nm and 90-nm CMOS technologies,” IEEE Trans. Nucl. Sci., vol. 54, no. 6, pp. 2506–2511, Dec. 2007.

[4] M. P. Baze and S. P. Buchner, “Attenuation of single event induced pulses in CMOS combinational logic,” IEEE Trans. Nucl. Sci., vol. 44, no. 6, pp. 2217–2223, Dec. 1997.

[5] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, "Comparison of error rates in combinational and sequential logic," IEEE Trans. Nucl. Sci., vol. 44, no. 6, pp. 2209–2216, Dec. 1997.

[6] J. Benedetto et al., "Heavy ion-induced digital single-event transients in deep submicron processes," IEEE Trans. Nucl. Sci., vol. 51, no. 6, pp. 3480–3485, Dec. 2004.

[7] M. A. Bajura et al., "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 935–945, Aug. 2007.

[8] T. Bonnoit, M. Nicolaidis, and N.-E.Zergainoh, "Using error correcting codes without speed penalty in embedded memories: Algorithm, implementation and case study," J. Electron. Test., vol. 29, no. 3, pp. 383–400, Jun. 2013.

[9] P. Papavramidou and M. Nicolaidis, "Test algorithms for ECC-based memory repair in ultimate CMOS and post-CMOS," IEEE Trans. Comput., vol. 65, no. 7, pp. 2284–2298, Jul. 2015.

[10] R. Vemu, A. Jas, J. A. Abraham, S. Patil, and R. Galivanche, "A lowcost concurrent error detection technique for processor control logic," in Proc. DATE, Munich, Germany, 2008, pp. 897–902.

[11] S. Ghosh, N. A. Touba, and S. Basu, "Synthesis of low power CED circuits based on parity codes," in Proc. 23rd IEEE VLSI Test Symp., May 2005, pp. 315–320.

[12] V. Sapozhnikov, V. Sapozhnikov, D. Efanov, and A. Blyudov, "On the synthesis of unidirectional combinational circuits detecting all single faults," in Proc. EWDTS, 2014, pp. 1–8.

[13] C. A. Lisboa, F. L. Kastensmidt, E. H. Neto, G. Wirht, and L. Carro, "Using built-in sensors to cope with long duration transient faults in future technologies," in Proc. ITC, 2007, pp. 1–10.

[14] R. P. Bastos, F. S. Torres, G. Di Natale, M. Flottes, and B. Rouzeyre, "Novel transient-fault detection circuit featuring enhanced bulk builtin current sensor with low-power sleep-mode," J. Microelectron. Rel., vol. 52, nos. 9–10, pp. 1781–1786, 2012.

[15] M. Nicolaidis, "Double-sampling design paradigm—A compendium of architectures," IEEE Trans. Device Mater. Rel., vol. 15, no. 1, pp. 10–23, Mar. 2015.

[16] D. Ernst et al., "Razor: A low-power pipeline based on circuit-level timing speculation," in Proc. MICRO, 2003, pp. 7–18.

[17] S. Das et al., "RazorII: In situ error detection and correction for PVT and SER

ISSN: 2366-1313

tolerance," IEEE J. Solid-State Circuits, vol. 44, no. 1, pp. 32–48, Jan. 2009.

[18] M. R. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 8, pp. 1163–1176, Aug. 2013.