International Journal of Advanced and Innovative Research (2278-7844) / Volume 7 Issue 2

# DDR-SDRAM Controller ASIC Design for High Speed Interfacing

*Zeba khan, GGITS, Jabalpur*
*Dr. Vinod Kapse, GGITS, Jabalpur*

**Abstract:** The goal of this work is to develop DRAM controller between Main Processor and the main memory for fast interfacing of the data and this is achieved with the help of a new Super Harvard type of interfacing parallel interfacing for the data, program data and instructions, also the proposed work used four stage pipelining to achieve high throughput and high speed interfacing. Vertex Corse grain FPGA has been used for the design of the work hence the area can be minimized also the mix modeling architecture is been used. The architecture is designed in Xilinx EDA using Verilog HDL and verification of the design is been done of ISE. The result in terms of speed and area are found better.

**Keywords:** SDRAM, DDR, FPGA, RTS

## I-INTRODUCTION

The SDRAM has a synchronous interface and is operated by a predefined set of commands. Because only the bits of the active row are directly accessible, In general case the read/write operations involve a sequence of SDRAM com- minds, also called transactions. Three command signals (RAS#, CAS# and WE#) allow representing 8 dif- ferment commands.

Synchronous Dynamic Random-Access Memory, This is the first SDRAM standard and is now referred as Single Data Rate (SDR) to distinguish it from later Double Data Rate (DDR) standards. Single command and/or data word is transferred in one clock cycle. Unfortunately the standards document does not seem to be publicly accessible, so a datasheets of specific chips were used when preparing this section. The memory device might have 2 or 4 banks. The first devices were supporting the clock frequencies of 66 to 100 MHz but more recent 64 Mb.

## II-PROPOSED DESIGN

The main function of DDR SDRAM is to double the bandwidth of the memory by transferring data (either read operation or write operation) twice per cycle on both the falling and raising edges of the clock signal. The designed DDR Controller generates the control signals as synchronous command interface between the DRAM Memory and other modules. Fast interfacing of the data and this is achieved with the help of a new Super Harvard type of interfacing parallel interfacing for the data, program data and instructions, also the proposed work used four stage pipelining to achieve high throughput and high speed interfacing. Vertex Corse grain FPGA has been used for the design of the work hence the area can be minimized also the mix modeling architecture is been used. The architecture is designed in Xilinx EDA using Verilog HDL and verification of the design is been done of ISE. The result in terms of speed and area are found better.

**Choice of FPGA:** Various FPGA’s are available now a day’s which give us good hands on research work in the field of ASIC designing. And as we knew a hard core ASIC will gives us a better throughput over the software based library routine if our application is specific. If our application is defined then there is nothing to trade off, for better performance FPGA based IP (Intellectual Property) is batter choice. Hybrid FPGA is an FPGA with a Hybrid Interconnect Structure

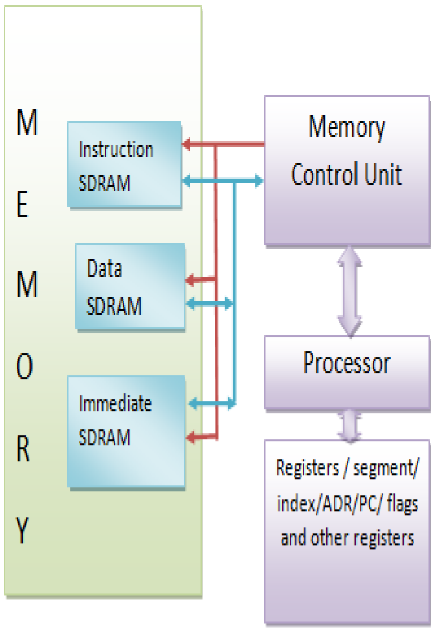


Figure 1: Block diagram of proposed controller architecture

Figure 1 shown below is the outlook architecture of proposed work; here it can easily observe that there are total three isolated memory first instruction SDRAM second data SDRAM and third immediate data SDRAM. As known from the super Harvard architecture each memory has its separate data and address line. Proposed Memory controller work also has separate data and address line for each memory. The instruction SDRAM is there to store OPCODE of instructions; the data SDRAM is there for storing temporary data generated during execution of program. And the immediate SDRAM memory is there for storing direct data of Program if any

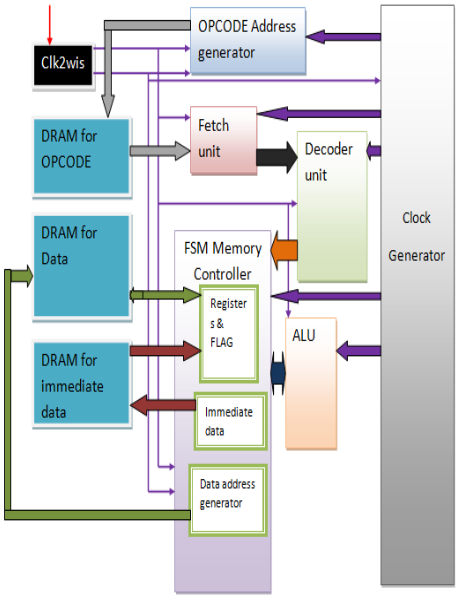


Figure 2: Elaborated Architecture of Proposed Processor Module

Proposed concept to have an isolated memory for the immediate data in Super Harvard architecture with three level SDRAM memory. With the proposed Controller architecture it was possible to execute CISC instruction set with RISC feature that is every instruction in one clock cycle. It helps to achieve pipeline efficiently. It is been achieved with proficient design of Decoder and execution module of proposed work. Because proposed concept fetch module does not requires to wait for decoder module, it keep fetching the OPCODE at each clock without taking care about decoder signal, decoder signal is been dedicated to execution unit only.

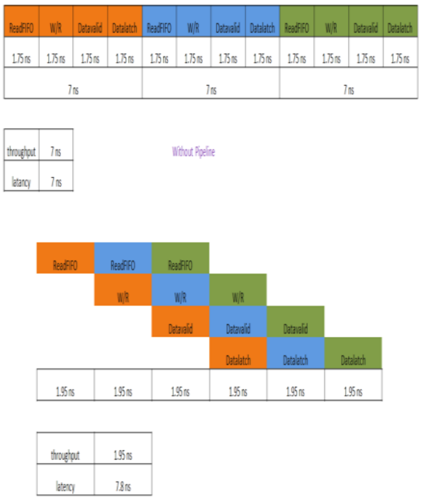


Figure 3 Tian Jin[1] pipeline architecture for high throughput

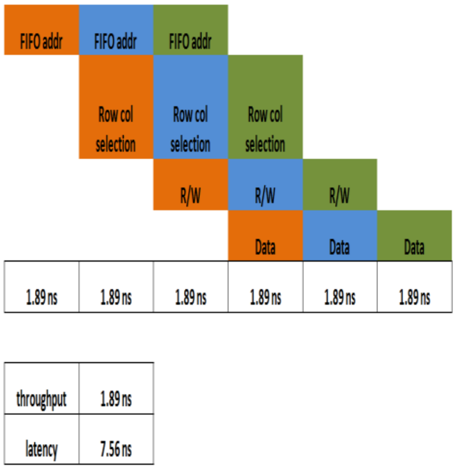
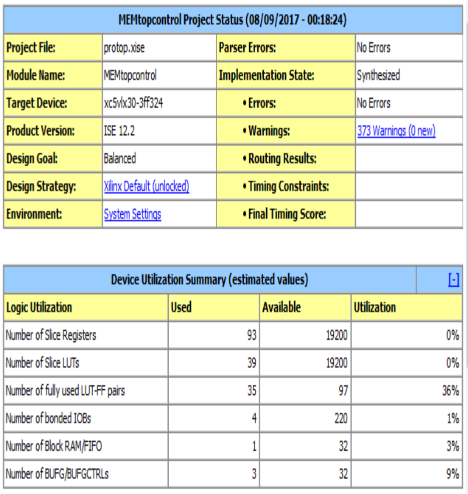


Figure 4 Proposed pipeline architecture for high throughput

## III-RESULTS



| MEMtopcontrol Project Status (08/09/2017 - 00:18:24) | | | |
|---|---|---|---|
| Project File: | protop.xise | Parser Errors: | No Errors |
| Module Name: | MEMtopcontrol | Implementation State: | Synthesized |
| Target Device: | xc5vlx30-3ff324 | • Errors: | No Errors |
| Product Version: | ISE 12.2 | • Warnings: | 373 Warnings (0 new) |
| Design Goal: | Balanced | • Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | System Settings | • Final Timing Score: | |

| Device Utilization Summary (estimated values) | | | [-] |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slice Registers | 93 | 19200 | 0% |
| Number of Slice LUTs | 39 | 19200 | 0% |
| Number of fully used LUT-FF pairs | 35 | 97 | 36% |
| Number of bonded IOBs | 4 | 220 | 1% |
| Number of Block RAM/FIFO | 1 | 32 | 3% |
| Number of BUFG/BUFGCTRLs | 3 | 32 | 9% |

Figure 5 the synthesis summary

| Design | Fmax (MHz) | Slices | LUT |
|---|---|---|---|
| Xilinx | 251 Mhz | 97 | 72 |
| Edgar Lakis et al [2] | 202Mhz | 101 | |
| Satish reddy et al [3] | | 104 | |
| Tian Jin et al [1] | 349 Mhz | | 127 |
| Ours | 374 Mhz | 93 | 74 |

Table 1: comparison of Synthesis results for evaluated SDR controllers.

It can be observe that proposed design is working better than available work. Xilinx design is clearly optimized for speed, so the high frequency is not surprising. The design is pipelined and the control logic is distributed always depending on just few bits. The critical path is on some wide multiplexer used to initialize a delay counter according the configuration register.

## IV-CONCLUSIONS

The open source SDR SDRAM controller has been created. Its initial integration into two RTS platforms (FPGA and JOP) was performed and tested. The different options of memory access scheduling for the FPGA plat- form have been investigated. The analysis included estimates of their RTS efficiency and the hardware implementation feasibility. For hard-RTS, the round robin (RR) does not have advantages over time division. Multiplexing (TDM), whereas WCET bounds can be made tighter with TDM. The static priority (SP) arbiters like CCSP and PBS are not scalable for WCET analysis because the least priority requester will suffer from latency proportional to the total bandwidth allocation of other requesters. The memory access timing analysis performed at WCET level suffers from fundamental limitations in reducing memory bandwidth over-allocation. The local worst case required bandwidth has to be allocated for the whole task's execution period.

## REFERENCES

[1] Tian Jin, Wenxin Li, Xiangyu Hu, A Tow-Level Buffered SDRAM Controller,2016 3rd International Conference on Information Science and Control Engineering, 978-1-5090-2534-3 /16 -2016 IEEE, DOI 10.1109/ICISCE.2016.37

[2] SATISH REDDY N, GANESH CHOKKAKULA, BHUMARAPU DEVENDRA, ASIC Implementation of High Speed Pipelined DDR SDRAM Controller, ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India, ISBN No.978-1-4799-3834-6/14/2014 IEEE

[3] Edgar Lakis, Martin Schoeber,An SDRAM Controller for Real-Time Systems, In Proc. IEEE International Workshop on Application of Reliable Computing and Communication, pages 29–34, Dec. 2015.

[4] Deepali S h a r m a, S hruti bhargava, M a h e n d r a Vucha, Design and VLSI Implementation of DDR SDRAM Controller for High Speed Applications, Deepali Sharma et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1625-1632

[5] D. Vanden Bout,, application note on XSA SDRAM Controller, September 5, 2002 (Version 1.1)

[6] Benny Åkesson, An introduction to SDRAM and memory controllers, Philips

[7] Shabana Aqueel and Kavita Khare, Design and FPGA Implementation of DDR3 SDRAM Controller for High Performance, International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 4, August 2011, DOI : 10.5121/ijcsit.2011.3408 101

[8] Altera. SDRAM Controller Core, Quartus II Handbook Version 9.1 Volume 5: Embedded Peripherals, v9.1 edition, November 2009.

[9] Altera Corporation. SDR SDRAM Controller White Paper, ver. 1.1 edition, August 2002

[10] JEDEC. Synchronous Dynamic Random Access Memory (SDRAM). JEDEC Solid State Technology Association, JESD21-C, June 1994.

[11] XILINX, Inc. LogiCORE IP Multi-Port Memory Controller(V6.02.a). Data Sheet, Xilinx, Inc., September, 2010.

[12]Priyanka Bibay, Anil Kumar Sahu, Vivek Kumar Chandra "Design and Implementation of DDR SDRAM Controller using Verilog", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064. [13]Amit Bakshi, Sudhanshu Shekhar Pandey, Tribikram Pradhan, Ratnadip Dey, "ASIC Implementation of DDR SDRAM Memory Controller", 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology.

[14] Marco Paolieri, Eduardo Qui˜nones, Francisco J. Cazorla, and Mateo Valero. An analyzable memory controller for hard real-time CMPs. Embedded Systems Letters, 1(4):86–90, 2009.

[15] Jan Reineke, Isaac Liu, Hiren D. Patel, Sungjun Kim, and Edward A. Lee. PRET DRAM controller: bank privatization for predictability and temporal isolation. In Robert P. Dick and Jan Madsen, editors, Proceedings of the 9th International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS 2011, part of ESWeek '11 Seventh Embedded Systems Week, Taipei, Taiwan, 9-14 October, 2011, pages 99–108. ACM, 2011.