International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

DDR-SDRAM CONTROLLER ASIC DESIGNFORHIGH-SPEED INTERFACING WITH PROCESSING MODULE

Yashwant Singh[1], Sunil Saha[2]

[1]M.Tech. Scholar, Gyan Ganga Institute of Engineering and Sciences, Jabalpur, India
[2]Assistant Professor, Gyan Ganga Institute of Engineering and Sciences, Jabalpur, India

A B S T R A C T

Modern real-time embedded system must support multiple concurrently running applications. Double Data Rate Synchronous DRAM (DDR SDRAM) became mainstream choice in designing memories due to its burst access, speed and pipeline features. Synchronous dynamic access memory is designed to support DDR transferring. To achieve the correctness of different applications and system work as to be intended, the memory controller must be configured with pipelined design for multiple operations without delay.

Keywords: *SDRAM, Double Data Rate, FPGA, RTS, Memory Controller, VHDL*

1. INTRODUCTION

In recent years, SDRAM, Synchronous Dynamic Random Access Memory, from the initial SDRAM DDR to the most recent SDRAM SDR, have gone through several generations of upgrades. Due to its excellent performance and cheap price, SDRAM has been applied in many fields, such as general computing, graphic technology and embedded system. Its performance has been continuously improved from three dimensions, including capacity, frequency and bandwidth. On the other hand, the frequency of processor as the master of SDRAM is higher. And the number of processing cores becomes more and more, which can even reach thousands. However, the system composed of SDRAM processor and its performance often falls sharply. The direct cause of this phenomenon is the interface between processor and SDRAM controller, which performs too lower. Some of necessary functions of modern DRAM architectures produce relatively long access latencies [1]. The use of modern conventional architectures in real-time systems (RTS) requires complex analysis and suffers from high resource over-allocation needed to cover uncertainties stemming from employed speculative, average case optimizations. The design of time predictable RTS optimized architectures that allow easy timing analysis and tight timing guaranties is an active research topic. A real-time system (RTS) must perform its operation within a predefined time. For hard-RTS the delayed operation could cause serious consequences and must be avoided at all cost. Such systems must be rigorously analyzed and proved to always act on time. To ensure the proper timing in all conditions, the timing analysis considers worst possible case. If the worst-case performance is satisfactory, the system will to also operate properly under other conditions. The Synchronous Dynamic Random Access Memory (SDRAM) is widely used external memory because of its attractive combination of high capacity, low cost and competitive performance. However variable SDRAM access latencies pose some challenges for its effective use in RTS. The goal of this thesis is to explore the options for predictable SDRAM controller for use in the FPGA platform.

Figure 1 presents the context of RTS memory controller design. First of all the knowledge of DRAM technology is needed because it is the source of the limit- tins. Secondly, the controller should interact well with the analysis framework, because it derives the performance guaranties for the whole system. Finally, the good performance guaranties of a memory controller are only possible by tuning it to the execution platform and programming model. Only by understanding the interaction of these three domains of knowledge the efficient controller is possible.

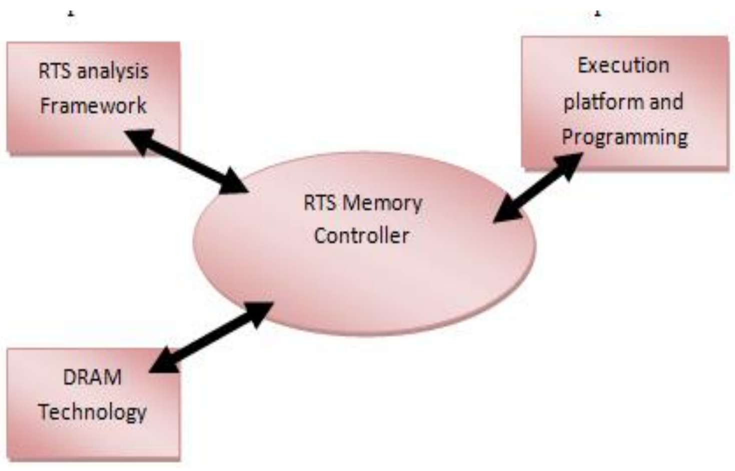

Figure 1: the context of the memory controller for the Real-Time System

DPRAM TECHNOLOGY: In this work, the focus is on Synchronous DRAM (SDRAM) which has been the most prevalent volatile on-chip memory for more than a decade. It is called synchronous to distinguish it from the asynchronous DRAM interfaces that were dominant at the time the standard was created. The standard is prepared by JEDEC Solid State Technology Association and several generations of the standards have evolved over the years.

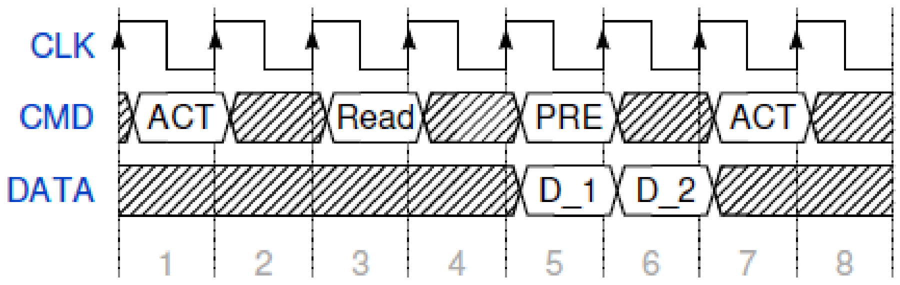

Figure 2: SDRAM (SDR) command sequence for burst read oftwo words.

The closed page policy used, i.e. the bank is recharged after use.

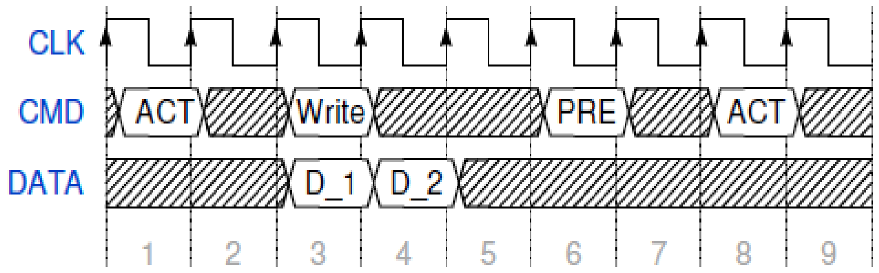

Figure 3: SDRAM (SDR) command sequence for burst write of two words.

Because the single Read and Write commands can only perform a transfer with the active row. In general case extra commands will need to be issued to perform a read or write operation. The Figure 2 and Figure 3 show the transactions for general two-word transfer operations.

In the examples the Recharge command is issued explicitly to show its place in the transaction. The action could also be scheduled for automatic execution by enabling the auto-recharge during Read /Write command.

SHARC: The Super Harvard Architecture Single-Chip Computer also known as SHARC is a high performance floating point (FPI) and fixed point DSP from Analog Devices. SHARC is quite common in a variety of signal processing applications starting from single CPU guided artillery shells to 2000CPU over the horizon radar processing super computers. The actual design dates to about Jan-feb 1994.

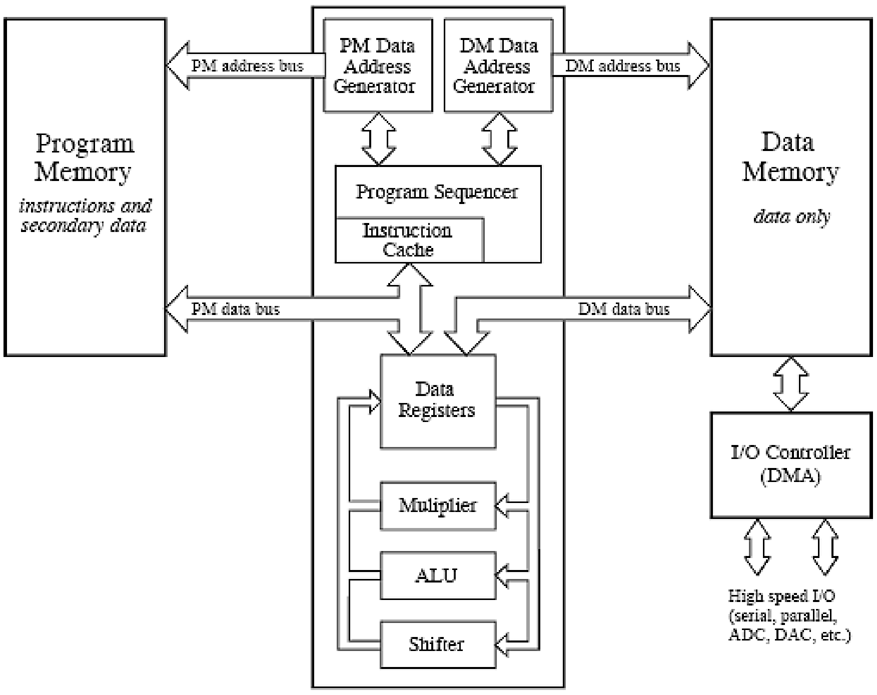

Figure 4: Super Harvard Architecture

SHARC processors were used because it provides good floating point number performance. SHARC processors are generally intended to have a significant number of serial links to other type of SHARC processors nearby; it can be used as a low-cost alternative of SMP. The SHARC use Super Harvard architecture with word-addressed VLIW processor; it does not use 8-bit or 16-bit values because each address is used to point to a complete 32-bit word, not just a 8bit number (byte). It is thus neither big-endian nor little-endian, although a compiler may use any convention if it implements 64-bit data or some different way to pack many 8-bit or 16-bit values into a single 32-bit word. Analog Devices chose this approachto avoid the issue by using a 32-bit char for their C compiler.

2. LITERATURE WORK

There are several previous works with the goal of creating SDRAM controller optimized for RTS. Altera Corporation [9] described a problem with refresh incorporation into WCET of the task without considering each transfer. They showed that even though the WCET augmented with the possible refresh interference is safe, the actual WCET path of the program might be different. This does not seem to be a problem in practice, if well behaved architecture without timing anomalies is used. Bhatt and Mueller describe an approach of grouping the refresh operations to- gather and executing them in separate special task [13]. This helps eliminate refresh interference uncertainty, because the refresh interference can be handled by the schedule ability analysis. And the refresh operation does not have to be incorporated into WCMAT of each memory operation which is pessimistic. UN- fortunately the authors did not mention that there is strong limitation on burst refresh in memories of later generations of SDRAM. Pitter et al has in part of his thesis performed some evaluation of arbitration schemes for RTS chip multiprocessor [9].

Edgar Lakis et al [3] Real-time systems need time-predictable components to support static WCET analysis. In this paper they presented an SDRAM controller that provides constant and well-known access time to the SDRAM memory. The controller is placed into open source and has been tested with two real-time platforms: the Java processor JOP and the time-predictable CMP platform T-CREST. As future work they consider to explore the possibilities of performing memory access analysis at schedulability level. The precision and feasibility of this approach will depend on the modeling of the task’s memory demand.

Satish reddy et al [2] In this paper an efficient and fully functional DDR SDRAM controller is designed. The controller generates different types of timing and control signals, which synchronizes the timing and control the flow of operation. Data path module is designed as it takes the data once per clock cycle and gives the data to the user at the rate twice per the clock cycle. Address latch is designed to generate the address location in the memory accordingly row and column basis, so that it can operate at double data rate. In normal DRAM, the data transfer (either read or write) is occurs from single location of memory, but in pipelined version of DDRSDRAM controller, it can do multiple operations from different memory locations simultaneously such as read from one location and write to another location. In pipelined DDRSDRAM controller, for a single row and multiple columns, the data is transferred at a rate faster than normal SDRAM controller, and 28.57% of improvement is achieved in the performance of memory accessing. In future this design can be implemented forDDR2/DDR3 also. This is one of the efficient designs for pipeline the memory accessing and multiple operations without including the much delay.

Tian Jin et al [1] Tow-Level Buffered SDRAM enhances the efficiency of reading and writing the peripheral SDRAM in a certain extent.
The Comparison test could instruct that the proposed design will improve the efficiency of 103.8% compared with the official SDRAM controller of XILINX. This design promotes the throughput capacity of interface between processor and peripheral SDRAM. To eliminate external memory bottlenecks, the basic performance gauges of the whole system will rack up some impressive gains.

Table 1 Literature work

Author/ Journal	Work	Outcome
Edgar Lakis et al / IEEE 2015	They design a SDRAM Controller for Real-Time Systems with statically determine worst-case execution times (WCET)	101 slices used of Vertex FPGA
Satish reddy et al/ IEEE 2014	They develop an ASIC for High-Speed Pipelined DDR SDRAM Controller	104 slices used of Vertex FPGA
Tian Jin et al / IEEE 2016	They work on A Tow-Level Buffered SDRAM Controller for fast access in RTS	127 LUT used of V Spartan FPGA used, 349 Mhz maximum speed achieved

The use of modern, conventional architectures in real-time systems (RTS) requires complex analysis and suffers from high resource over-allocation, needed to cover uncertainties stemming from employed speculative, average-case optimizations. The designs of time-predictable, current computers use DRAM as a cost-effective main memory. However, these DRAM chips have timing requirements that depend on former accesses and also need to be refreshed to retain their content. Standard memory controllers for DRAM memories are optimized to provide maximum bandwidth or throughput at the cost of variable latency for individual memory accesses.

Tian Jin et al [1] develop a two level SDRAM which works well and two level allow fast interfacing but their architecture does not use time compensation when there is miss and whenever miss appears time to interface increases then normal condition which make their method slower in that situations. Satish reddy et al [2] uses pipeline DDR RAM and implemented their work on FPGA but pipelining increases latency timing and they did not resolve the branch penalty which may cause wrong data access. Also, their work did not resolve any RAM fault.Edgar Lakis et al [3] develop a new interfacing of SDRAM for real time interfacing but they use DDR3 for interfacing which is itself faster and can interface in real time conditions also they did not use FSM for controlling the SDRAM which may cause slow response when non-sequential interfacing needed.

In this work we present an SDRAM controller for real-time systems. The controller is optimized for the worst case and constant latency to provide a base of the memory hierarchy for time-predictable systems.

3. CONCLUSIONS

This paper describes the work related to the SDRAM memory controller which is popularly used in microprocessor for fast data interfacing different research works and their limitation has been discussed and one can conclude that a design of memory controller which can deal with SHARC architecture in real time will reduces the limitations with significant amount.

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 DDRSDRAM 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 Workshopon 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 DDRSDRAM 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 SDRAMand memory controllers, Philips

[7] Shabana Aqueel and Kavita Khare, Design and FPGA Implementation of DDR3SDRAM 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 9.1Volume 5: Embedded Peripherals, v9.1 edition, November 2009.

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