IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 5, Issue 6, Ver. I (Nov -Dec. 2015), PP 07-12
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

FPGA based QDR-II+ SRAM Controller

Abhilash Chadhar, R. Nandakumar
National Institute of Electronics and Information Technology Calicut, Kerala, India
National Institute of Electronics and Information Technology Calicut, Kerala, India

Abstract: *The hike in internet usage globally and the ascending number of its users have ensued to high demands for high speed data communication systems enriched with faster processors and high speed interfaces to peripheral components. With the advent to QDR SRAMs these requirements of this evolving system are fulfilled. The QDR SRAM enables the system to work on four times the bandwidth when compared to other SRAM architectures. For proper functioning of these SRAMs we introduce a design of QDR II+ controller which serves the purpose of translating memory requests issued by the hardware to which the RAM has been interfaced, into data and control signals for the SRAM while complying to the timing requirements imposed by the memory configurations.*

Keywords: *FPGA, QDR II+ memory, IP core, SoC, Verilog*

I. Introduction

With the rapid increase in the processor speed, system designers have used better speed management techniques such as use of interleaving, burst mode and other high speed memory systems for accessing memory. For fulfilling the need of high speed data management SRAM serve as a most likely option [5]. There as some concrete reasons that justifies the use of SRAM in a system. These include speed; cost of memory, volatility and features. Even the fastest DRAMs require approximately six to ten processor clock cycles to access the first data. This results in mismatch of speeds of the processor and the DRAM which in turn results in slower system with improper resource management. In such cases SRAMs emerges to be a better option which can operate at speeds of 550 MHz and even beyond. They even provide access times and cycle times equal to clock cycles used by the microprocessors. Thus these ultra fast SRAMs prove as better and effective alternatives over its other RAM variants. The density of SRAM is not as higher as DRAM because number of transistors required in for one SRAM cell is large as compared to DRAM cell hence in today's scenario (in terms of chip density) where 4-Gbits DRAMs are available the largest SRAMs are 144-Mbits [14]. DRAM cells need to be refreshed for proper operation while SRAM cells need not be refreshed due to its volatile nature. Hence they are available for reading and writing cent percent of the time.

II. Architecture of the SRAM

QDR-II+ SRAM memory device offers two architecture variants [2], [11], [12]; those are 2 word burst and 4-word burst. Only 4-word burst architecture is discussed here. 4- Word memory device architecture comprises of 4 array of 512×18 SRAM. Unlike DDR SRAM, this memory device has two data ports .Address port is of 19-bits and Data ports of 72-bits. Control Signals $\overline{RPS}$, $\overline{WPS}$, $\overline{BWS}$ are available with this memory device. $\overline{RPS}$ is active low and asserted

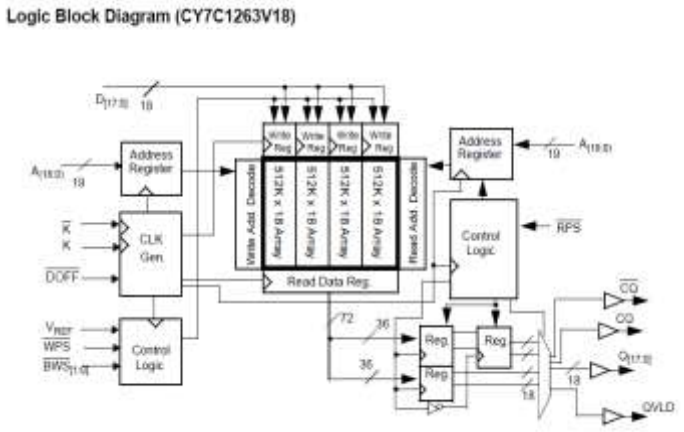


Fig. 1 the logic block diagram or the architectural view of QDR + SRAM. It has been taken from CYPRESS CY7C1263v18 datasheet.[11]

when the read operation is required. $\overline{WPS}$ is an active low control signal and it is asserted when write operation has to be performed. $\overline{BWS}$ is provided a 0 (Active Low) in order to write the data into memory and if it will be 1 the data will not be written into memory at the target address. $\overline{DOFF}$, when it is given 0 then DLL present inside the memory device will get off and start to perform like QDR-1 SRAM memory. K and $\overline{K}$ are input clock signal and complement of one another. CQ and $\overline{CQ}$ are two echo clocks and synchronized with Input Clock K and $\overline{K}$ respectively.

The objective of this design is to implement a controller in FPGA which allows all the operations in QDRII+ SRAM. The design also contains a test module which contains data generators in the form of LFSRs (Linear feedback shift register), a comparator and an address counter.

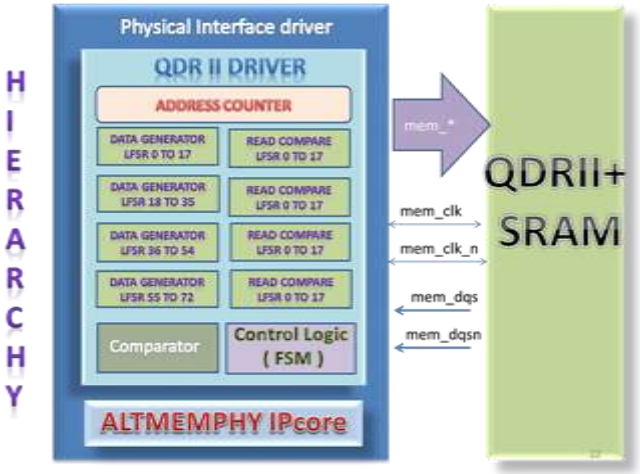


Fig. 2 The figure shows the hierarchy of the modules used in the program along with signals interacting with the target device.

III. Principle Of Operation

The design has been produced as per the standards of QDRII+ SRAMs which is provided in the table 1.1 Memory controller assigns the signals to QDRII+ SRAM based on the inputs from the system which are memory requests. In our design the ALTMEMPHY IP core serves as the interface between the user logic and the memory controller as well as between the memory controller and the QDR II+ SRAM [6],[12]. A diagrammatic illustration of the top level scheme containing these interfaces has been clearly in the fig 1.2.

The principle behind the design is based on the timing characteristics of the SRAM device which gives the information of the control, data path and acknowledgment signals. A FSM which appropriately changes the states providing signals at required time acts as the heart of the controller. There are five states which properly describe the working of the controller. The FSM is shown in fig 2 containing all those states with corresponding inputs and outputs. The machine is a mealy model.

Initially the controller is at reset state, also it returns to reset state whenever reset is provided. At the positive edge of the phy_clk it latches the write and read port select signals, these signals are active low. In our design the controller is embedded with testbench which first writes to 100 locations and then reads them back in order to compare whether the data is correctly written or not. Hence initially address counter is reset so that it could start counting 100 addresses then the controller initiates writing when it senses wps_n as a low. It latches the lower data word of D at the same instant. At the falling edge of the phy_clk the address is latched along with upper data word on D. This completes the write cycle.

The quad data rate is achieved by operating two ports at dual data rate. This is the reason why throughput is enhanced in QDR SRAMs. The controller asserts read and write control signals at different ports hence the time required for bus turnaround is reduced because same bus is not used for write and read. This results in increase of bandwidth in QDR SRAMs as compared to DDR SRAMs or ordinary SRAMs. Data flow into the SRAM is through the write port and out of it is through the read port. These devices multiplex the address inputs to minimize the number of address pins required.

When the controller is in write state it latches 18-bit word at every rising edge of phyclk, another 18-bit word is latched at the negative edge of the clock. This process continues for one more cycle where a total of 72 bits are obtained, this makes the ports busy for two clock cycles hence the write operation cannot be accessed two consecutive cycles. This is the reason why there is an intermediate state. When the address counter reaches 100 addresses it resets the address counter and lfsr. Now if rps_n is active it will start read operation else it will wait at the intermediate state. In the read state 4 array accesses of 512x18 bits are performed in a burst of four

subsequent 18 bit data words. On the following two falling edges of phy_clk after detection of rps_n active leads to driving of lower 18 bit data on to corresponding Q, at the next subsequent rising edges of phy_clk the data is driven out to Q until all four 18bit data words are given to Q.

The read access is similar to write operation contains four 18 bit words which take two cycles to read. Hence read requests also cannot be made on 2 consecutive cycles. This is the reason why there is an intermediate state. When the address counter reaches 100 addresses it resets the address counter and lfsr. Now if rps_n is active it will start read operation else it will wait at the intermediate state. In the read state 4 array accesses of 512x18 bits are performed in a burst of four subsequent 18 bit data words. On the following two falling edges of phy_clk after detection of rps_n active leads to driving of lower 18 bit data on to corresponding Q, at the next subsequent rising edges of phy_clk the data is driven out to Q until all four 18bit data words are given to Q.

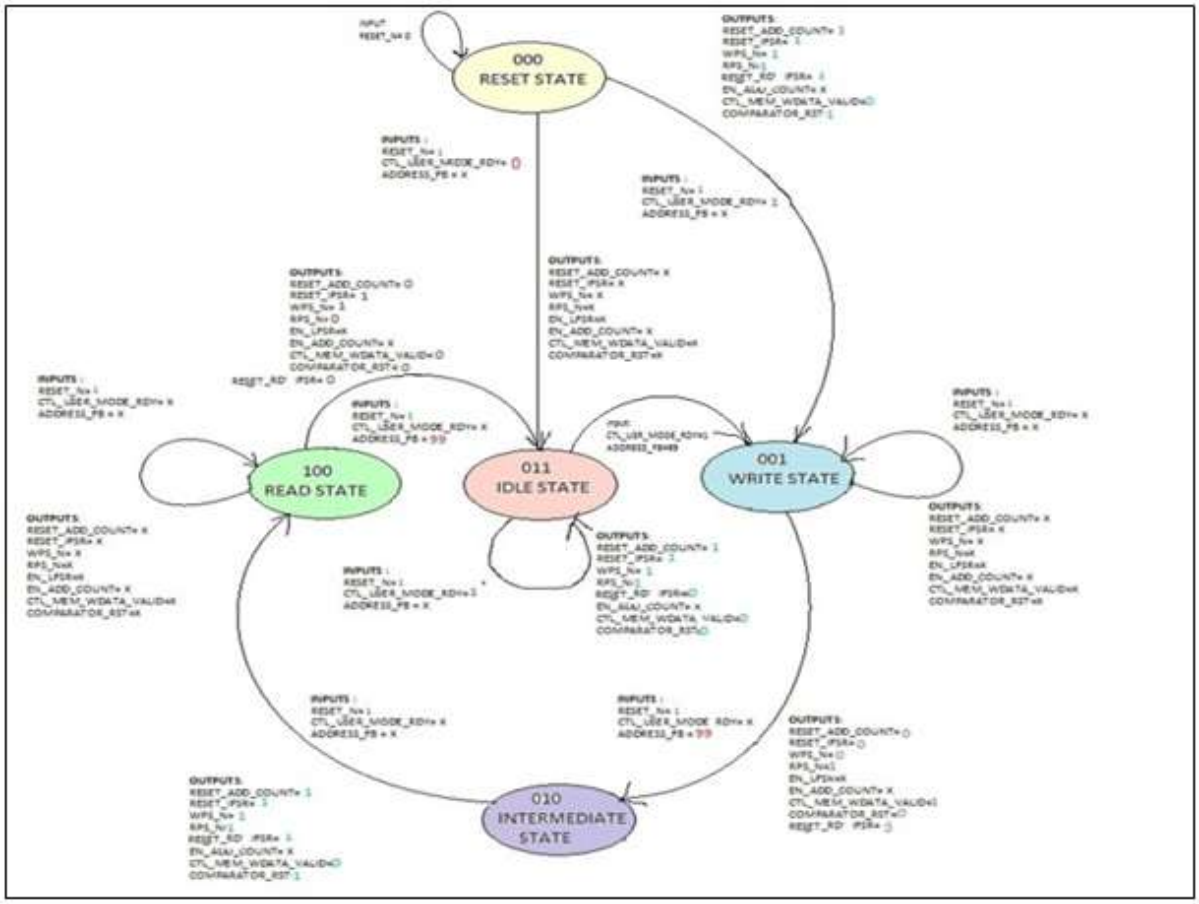


Fig 3 . Control Logic FSM

The read access is similar to write operation contains four 18 bit words which take two cycles to read. Hence read requests also cannot be made on 2 consecutive cycles. After the completion of read operation it goes to idle state from where it could again start writing as per the command issued.

IV. Core I/O Diagram (Fig 4)

The core contains the following:

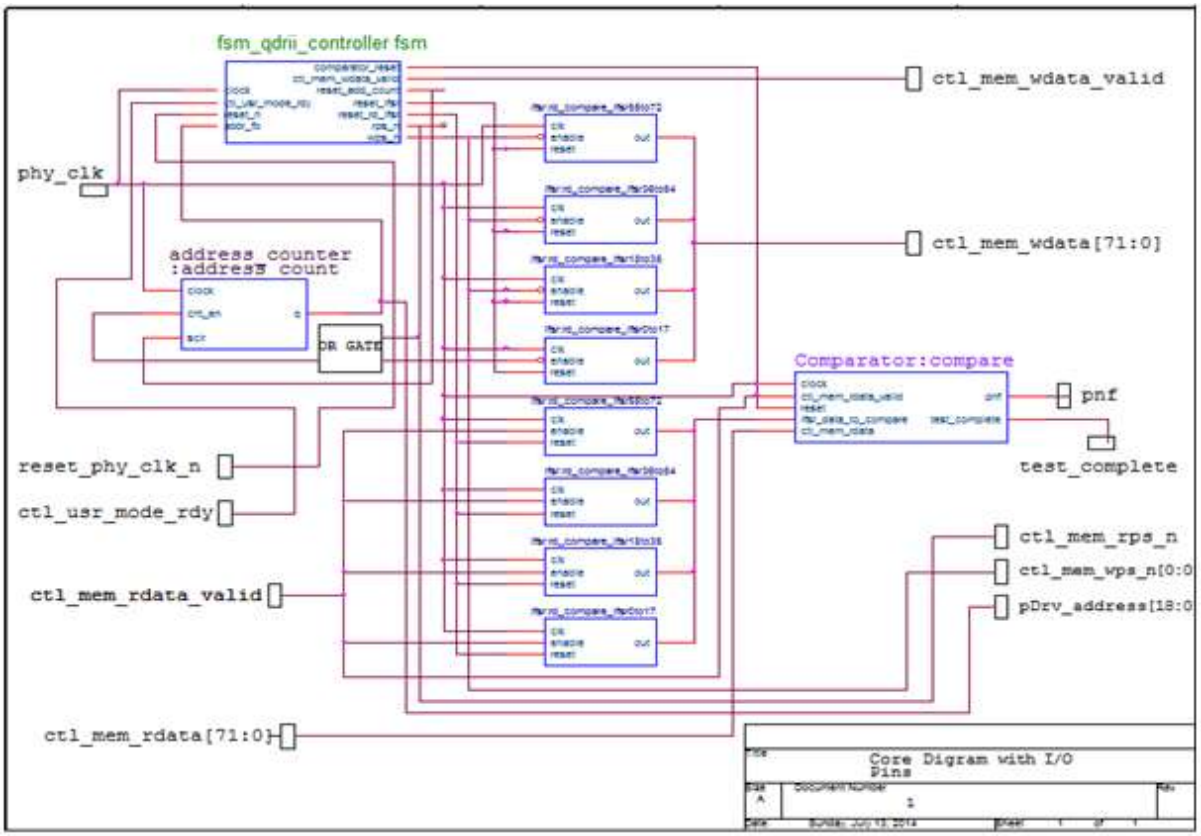
LFSRs (Linear Feedback Shift Registers)

Each one is of 18 bit. Total 8 LFSRs are used. Four are used to for generation of data to be written, rest of the four are used for read comparison.

COMPARATOR

It is used to compare whether the read-back data is same as written data in the memory. It gives output pass or fail based on comparison.

Control logic (FSM)
Most essential part of the control that issues control signals based on the FSM. The FSM has 5 states: RESET, READ, WRITE, INTERMEDIATE (state between read and write) and IDLE state



Address Counter
It gives 19 bit data output and counts up to 524287. It is used to generate address to which data is to be written or from where it has to be read.
Most of the pins have been dealt with earlier. Remaining are: CTL_USR_MODE_RDY which is a high after the resynchronization phase is over after this signal only the user data can be written or read from the RAM.
CTL_MEM_RDATA_VALID is assigned high when the controller is in read mode else it is in low condition.

Resource utilization:

Parameter	Result
Combinational ALUTs	908 / 113,600 (1 %)
Memory ALUTs	72 / 56,800 (< 1 %)
Dedicated logic registers	4,041 / 113,600 (4 %)
Total registers	4177
Total pins	75 / 744 (10 %)
Total block memory bits	528,384 / 5,630,976 (9 %)
DSP block 18-bit elements	0 / 384 (0 %)
Total PLLs	1 / 8 (13 %)
Total DLLs	1 / 4 (25 %)

Power Analysis Result:

Parameter	Result
Family	Stratix III
Device	EP3SL150F1152C2
Total Thermal Power Dissipation	1434.64 mW
Core Dynamic Thermal Power Dissipation	107.57 mW
Core Static Thermal Power Dissipation	598.50 mW
I/O Thermal Power Dissipation	728.58 mW

V. Hardware Test Results

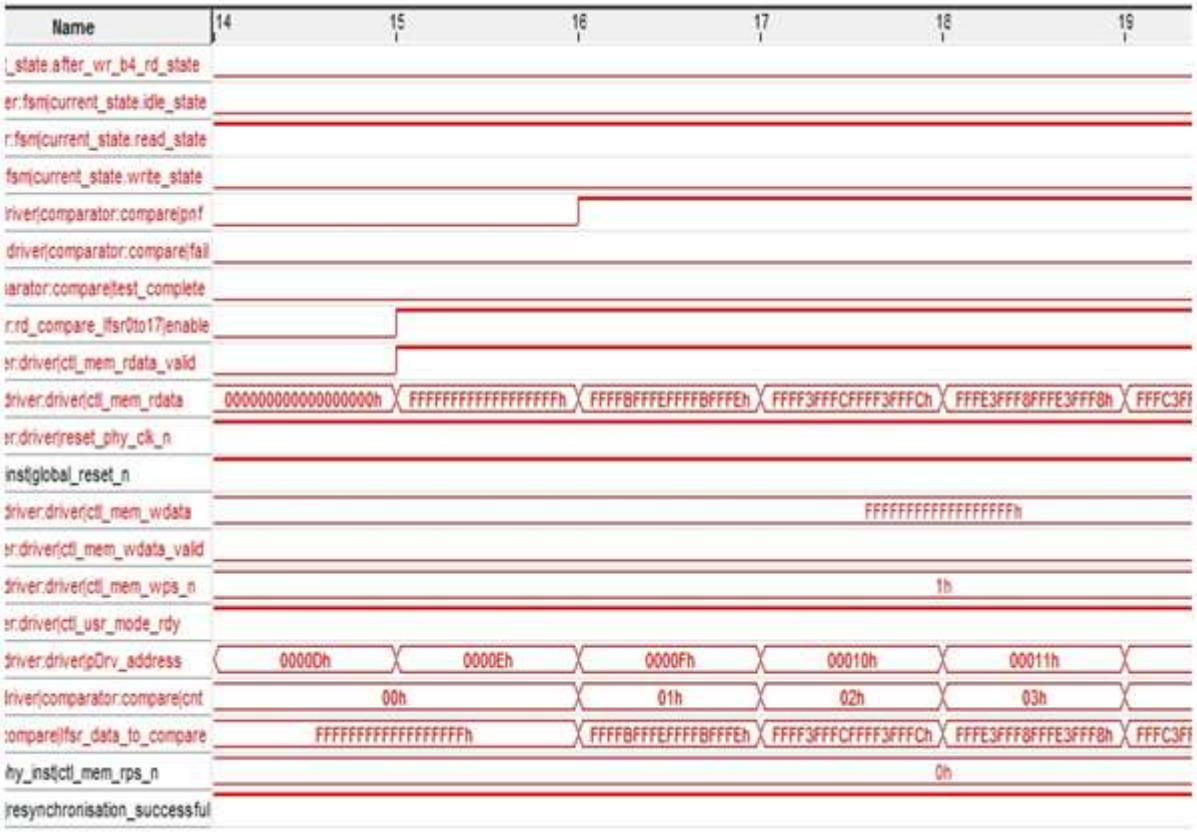


Fig.6. The above signal Tap output illustrates read state of the controller where comparison is being performed of the earlier written data into the the memory with currently read data through a comparator module which is a part of the test-bench .Wherever the read data is same as LFSR data to be compared the pnf(pass and fail) signal becomes high

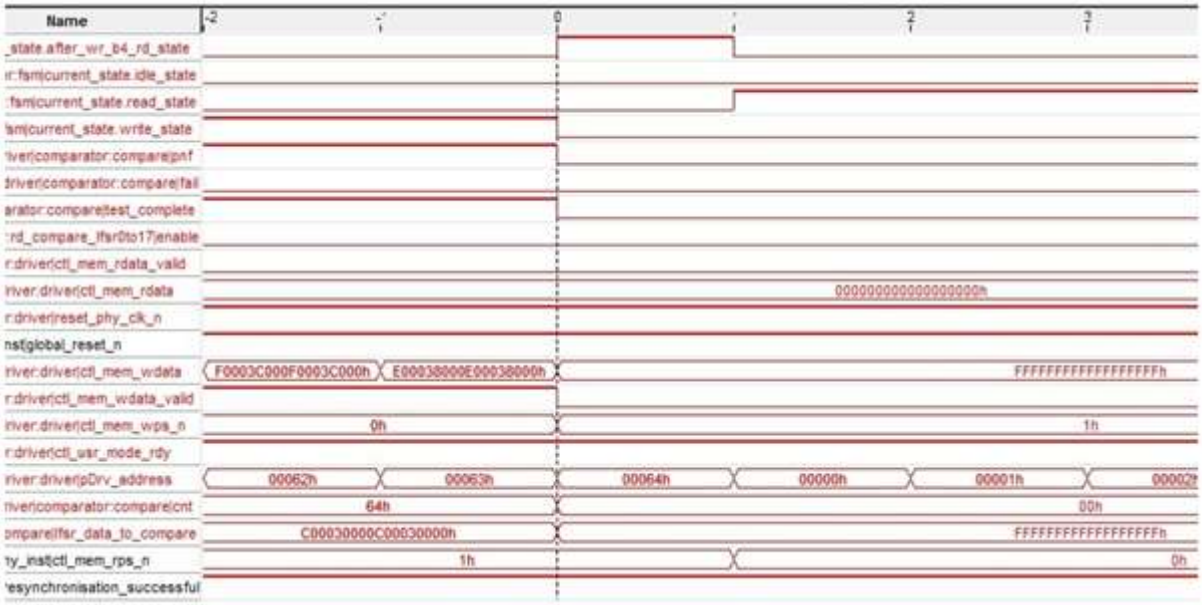


Fig. 5. The signal Tap output shows that the controller is switching from write state to read state. The write port select and read port select changing accordingly

VI. Potential Applictaions

It is used along with DSP processor memories, cache memories, routers, ATM switches, packet memories, look-up tables etc.

VII. Conclusion And Future Work

As QDR II + SRAMs are playing a significant role in determining performance of the internet hence it is very important to consider impact of a fast controller in present scenario. The design presented by us is a simplistic model that serves the basic requirements of a memory controller. Hence it is incapable to detect if sometimes the memory gives incorrect results. As our future plan, work could be done in improving the data correction ability of the controller using various algorithms. These techniques will be useful in applications where correct retrieval of data is very important like the space applications. Another area is that, present controller is device specific. It could be made as a universal QDRII+ SRAM controller for all the devices or at least the same family of devices.

References

[1]. Altera Corporation, *AN 461: Design Guidelines for Implementing QDRII+ and QDRII SRAM Interfaces in Stratix III and Stratix IV Devices*. February 2010.
[2]. Altera Corporation, *AN 326: Interfacing QDRII+ & QDRII with Stratix II, Stratix II GX, Stratix, & Stratix GX Devices*.
[3]. Cypress Semiconductor Corporation.*CY7C1302V25 9-Mb Pipelined SRAM with QDR™ Architecture*. May 2008.
[4]. Steven M. Rubin, *Computer Aids for VLSI Design*, Addison-Wesley, 1987.
[5]. Jack Truong. (March 31st, 2005). *Evolution of Network Memory.* [Online]. Available: http://www.jedex.org/
[6]. Markus Ringhofer, "*Design and Implementation of a Memory Controller for Real-Time Applications*," Institut fur Technische Informatik ¨ Technische Universit¨at Graz
[7]. P Sundararajan. *High Performance Computing Using FPGAs*. Xilinx White Paper: FPGAs, 2010 - china.zylinks.com
[8]. M. Morris Mano, *Digital Design*, Prentice-Hall, 1984.
[9]. Granberg, *Handbook of Digital Techniques for High-Speed Design*.. Low price edition.Patparganj, Delhi,India. Pearson Education India,ch 13 and ch 16.
[10]. Altera Corporation, *AN 133: QDR SRAM Controller Reference Design*. December 2000, ver. 1.0
[11]. Altera Corporation. *AN 349: QDR SRAM Controller Reference Design for Stratix & Stratix GX Devices*. May 2004, ver. 1.0
[12]. Altera Corporation. *ALTERA External Memory PHY Interface (ALTMEMPHY) Mega function User Guide*. January 2010
[13]. Simulating DRAM controllers for future system architecture exploration.
[14]. W. Wolf, A. Jerraya, and G. Martin, "Multiprocessor system-on-chip (mpsoc) technology," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 27, no. 10, pp. 1701–1713, Oct. 2008.
[15]. M. W. Naouar, A. A. Naassani, E. Monmasson, and I. Slama-Belkhodja, ''FPGA-based predictive current for synchronous machine speed drive,'' IEEE Trans. Power Electron., vol. 23, no. 4, pp. 2115–2126, July 2008.
[16]. Cypress Perform . *36-Mbit QDR® II SRAM Four-Word Burst Architecture* [Online].Available:http://www.cypress.com/?rID=47422
[17]. Cypress Perform. *144-Mbit QDR® II SRAM Two-Word Burst Architecture* [Online]. Available: http://www.cypress.com/?docID=48535
[18]. Cypress Semiconductor Corporation (Jayasree N / Pritesh M.), *PLL Considerations in QDR® -II/II+/DDR-II/II+SRAMs*.
[19]. QDR Consortium. *QDR Product Family* [Online]. Available: http://www.qdrconsortium.org/qdr-product-family.htm
[20]. Anuj Chakrapani. *QDR SRAM and RLDRAM: A Comparative Analysis.* Network Systems: Design Line, Cypress Semiconductor Corp, 2010.