IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 2, Ver. III (Mar - Apr.2015), PP 69-74
www.iosrjournals.org

Implementation of DDR SDRAM Controller using Verilog HDL

Mayuri Sanwatsarkar and Jagdish Nagar
Department of Electronics and Communication
Acropolis Institute of Technology & research, RGPV Bhopal, India

Abstract: *Now days, DDR SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory) has become the most popular class of memory used in computers due to its high speed, burst access and pipeline feature. For high speed applications like image/video processing, signal processing, networking etc. DDR SDRAM is widely used. The basic operations of DDR SDRAM controller are similar to that of SDR (Single Data Rate) SDRAM; however there is a difference in the circuit design; DDR simply use sophisticated circuit techniques to achieve high speed. To perform more operations per clock cycle DDR SDRAM uses double data rate architecture. DDR SDRAM (also known as DDR) transfers data on both the rising and falling edge of the clock. The DDR controller is designed with objective of proper commands for SDRAM initialization, read/write accesses, regular refresh operation, proper active and precharge command etc. DDR SDRAM controller is implemented using Verilog HDL and simulation and synthesis is done by using Modelsim and Xilinx ISE accordingly.*

Keywords: *DDR, SDRAM, burst access, pipeline, Verilog HDL.*

I. Introduction

Electronics is the field in which new developments are taken place every day. The sphere of electronics has become so vast, it has penetrates into our homes, places of work, in means of communication, in medical science as well as in defense applications. In homes and offices we use electronic devices like mobile phones, laptop, landline phone, T.V., desktop computers, microwave oven, washing machines etc. Most of the electronic devices need memory to store program and/ or data. In case of landline phones we need memory to store messages in case we are not at home or busy in some other work or in meeting. In case of mobile phones and laptop or in case of desktop computer we need memory to store data like contact numbers, photos, reminders, wallpapers, song, movies etc. So all of these devices need memory to store data whatever that can be. There are several types of memories are available. In some cases access to the memory is supported by processor’s in-built circuitry, but in some cases we need to design a sophisticated circuit to access the memory in order to achieve high speed. A microprocessor has several tasks to do at once, to make free the processor from the task like sending or receiving the data from the memory we need memory controller. The use of a memory controller provides the benefit of accessing the memory fast as it directly deal with the memory. Only once the processor sends the instruction to the controller according to that the controller generates the control signal and addresses to the memory.

In the applications where the processor has to transfer bulk data it may require a lot of time in just transferring the data from source to destination using program controlled data transfer or interrupt driven data transfer. The alternate way of transferring the bulk data is the use of direct memory access technique in which the data transfer takes place under the control of DDR SDRAM controller, after it is initialized by the processor. A DDR memory controller is designed to achieve the high speed bulk data transfer task much faster than the processor [7].

RAM: Every computer system needs a memory area in which it can store the data on which it is working. This memory area is almost invariably made up of Random Access Memory (RAM). Random access memories can be both read and written. They are called random access because addresses can be read in any order. A RAM can be written an infinite number of times. The microprocessors can read data from the RAM quickly, faster than the ROM. The RAM forgets its data if the power is turned OFF. Hence the RAM is called the volatile memory.

Usually RAMs are made up of cells consisting of MOSFETs. There is one cell each for a bit in the memory. Memory is made up of bits arranged in a two dimensional grid. In which memory cells are etched onto a silicon wafer in an array of columns (bit lines) and row (word lines). The intersection of a bit line and word line constitutes the address of the memory cell [2].

Two forms of RAM are Static RAM (SRAM) and Dynamic RAM (DRAM). A battery backed RAM is called Non Volatile RAM (NVRAM).

Static RAM: The Static RAM uses a flip flop to store a bit of binary information or data. The static means that once a processor cycle writes a bit in a cell, it remains unchanged until it is modified in the processor cycle or until the power switches off. There are four to six transistors per cell in a SRAM. An advantage of

SRAM is that a write to it is static as long as the power is on and it does not require any refreshing circuit since it does not uses a capacitor in each cell. So static RAM is fast but is expensive.

Dynamic RAM: Most bulk memory in modern systems is Dynamic RAM (DRAM). A DRAM uses tiny capacitor to store a bit of binary information. DRAM is very dense, but it requires that its values be refreshed periodically since the values inside the memory cells decay over time. An advantage of DRAM is that it stores a bit in less space than SRAM and is inexpensive. However it is slow since its memory locations need to be refreshed at regular intervals.

Synchronous DRAM: The most dominant form of DRAM today, which uses clocks to improve the performance of DRAM, is Synchronous DRAM (SDRAM). Each cell is organized in rows and columns. SDRAM is arranged in banks of memory addressed by the row and column. The number of row and column address bits and number of bank defines the size of memory. SDRAM uses Row Address Select (RAS) and Column Address Select (CAS) signals to break the address into two parts, which selects the proper row and column in RAM array. Signal transitions are relative to the SDRAM clock that allows the internal operations of SDRAM to be pipelined. SDRAM uses a separate refresh control circuit to perform auto refresh operation. SDRAM has to be refreshed roughly once per millisecond. SDRAMs refresh part of the memory at a time instead of refreshing the entire memory at once. When a part of memory is being refreshed, it cannot be accessed until the refresh operation is completed. The memory refresh occurs over fairly few seconds so that each section is refreshed every few microseconds [6].

SDRAM also contains register that defines or control the mode in which SDRAM operates. SDRAM supports the burst length of 2, 4, and 8 that allows several sequential addresses to be accessed by sending only one address. Memories for PCs can be purchased either as Single In-line Memory Modules (SIMMs) or Double In-line Memory Modules (DIMMs). A SIMM or DIMM is a small circuit that fits into a standard memory socket. A DIMMS has two leads compared to the SIMM’s one. Memory chips are soldered to the circuit board to supply the desired memory [6]. SDRAM come in DIMMS.

On the basis of data transfer rate the SDRAM can be classified into two categories: one is Single Data Rate (SDR) and other is Double Data Rate (DDR) SDRAM. The SDR SDRAM transfers data only on the rising edge of the clock whereas DDR SDRAM transfers data on both the rising as well as on the falling edge of the clock due to which the data transfer rate gets double. Now a day, even faster version of DDR SDRAMs i.e., DDR2 and DDR3 are also available for use in system designing.

However most of the operations of SDR SDRAM and DDR SDRAM controller are same, but in case of DDR SDRAM controller we need to modify some features since it uses double data rate architecture. The DDR controller uses the commands such as NOP, REFRESH, READA, WRITEA, PRECHARGE, and Load_MODE command to control the operation of SDRAM.

II. Design of DDR SDRAM controller:

The basic functional block diagram of DDR SDRAM controller shown in fig.1 [3] consists of three modules:

i. The control interface module
ii. Command module or signal generation module and
iii. The data path module

1. Control interface module: The control interface module; is the primary module; consists of a finite state machine shown in fig.2 and a programmable 16 bit counter to perform auto refresh operation. The control interface module accepts commands from the processor, decodes them and sends the decoded REFRESH, NOP, READA, WRITEA, PRECHARGE, and Load_MODE command to command module. The DDR SDRAM uses a separate clock: CLK and CLK` (the crossing of CLK going high and CLK` going low is considered as the positive edge of the clock). The commands are registered only on the positive edge of the clock. READ and WRITE are the basic commands used to access the SDRAM. The DDR SDRAM supports the burst length of 2, 4, or 8 locations for programmable READ/WRITE operation. An access to the SDRAM starts at any assigned location and continues until the burst length is reached. The READ and WRITE operation start by sending the ACTIVATE command.

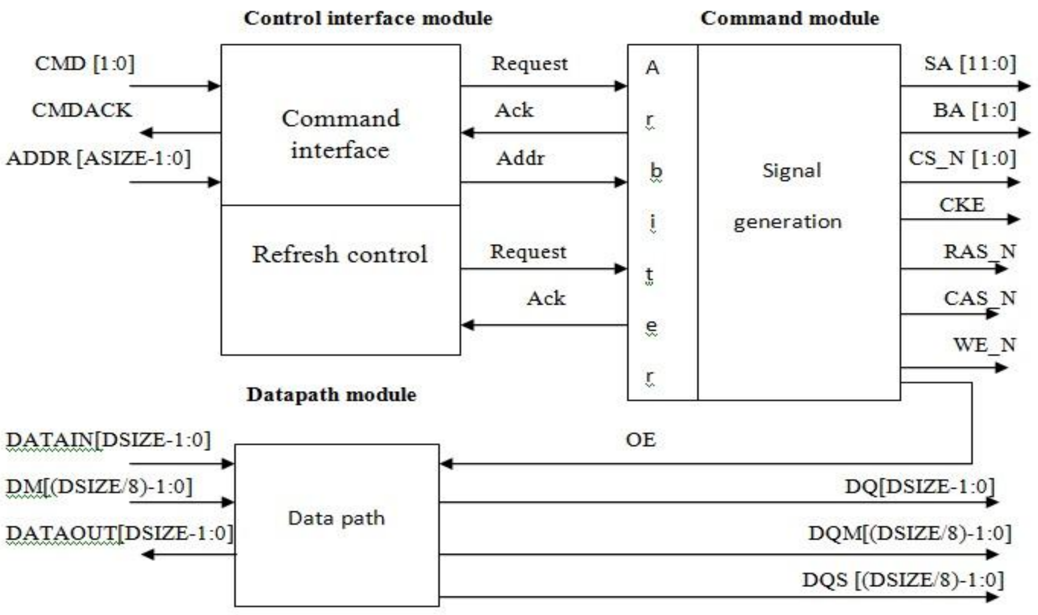


Fig.1 Block diagram of DDR SDRAM Controller

The controller consists of a finite state machine shown in fig.2 [5]. The initial state of the controller is ideal state. In the next state controller can be in either PRECHARGE, REFRESH, LOAD_MODE, or ACTIVATE state depend on the request from the processor. The dotted lines in the fig.2 show an automatic sequence. The READ and WRITE operation is followed by the ACT command. The active command is used to open a row in a particular bank and PRECHARGE command is used to close an open bank if the open bank is other than that we want to access. When a WRITE command is detected, the controller will first go to the ACT state and then to the WRITE state. The WRITE operation will continue until the burstlength is reached or the burst terminate is inserted[x]. During the READ operation, the initial address is registered. During the READ operation the data will arrive 1-3 clock cycles later on the data bus and this delay is due the time required to read the internal DRAM. This time delay is known as CAS latency. On completion of the burst READ and WRITE operation the controller will go back to the IDLE state.

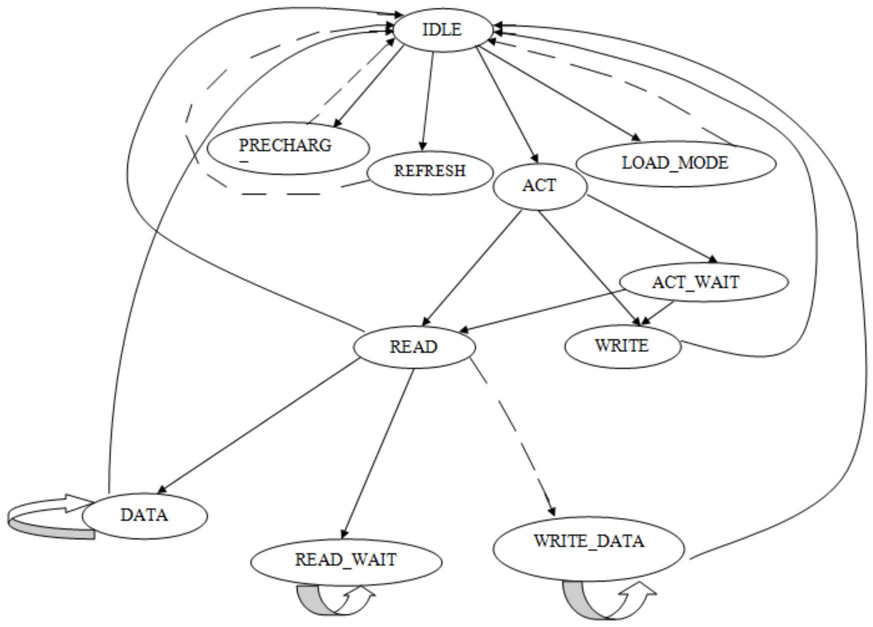


Fig.2 Finite state machine diagram

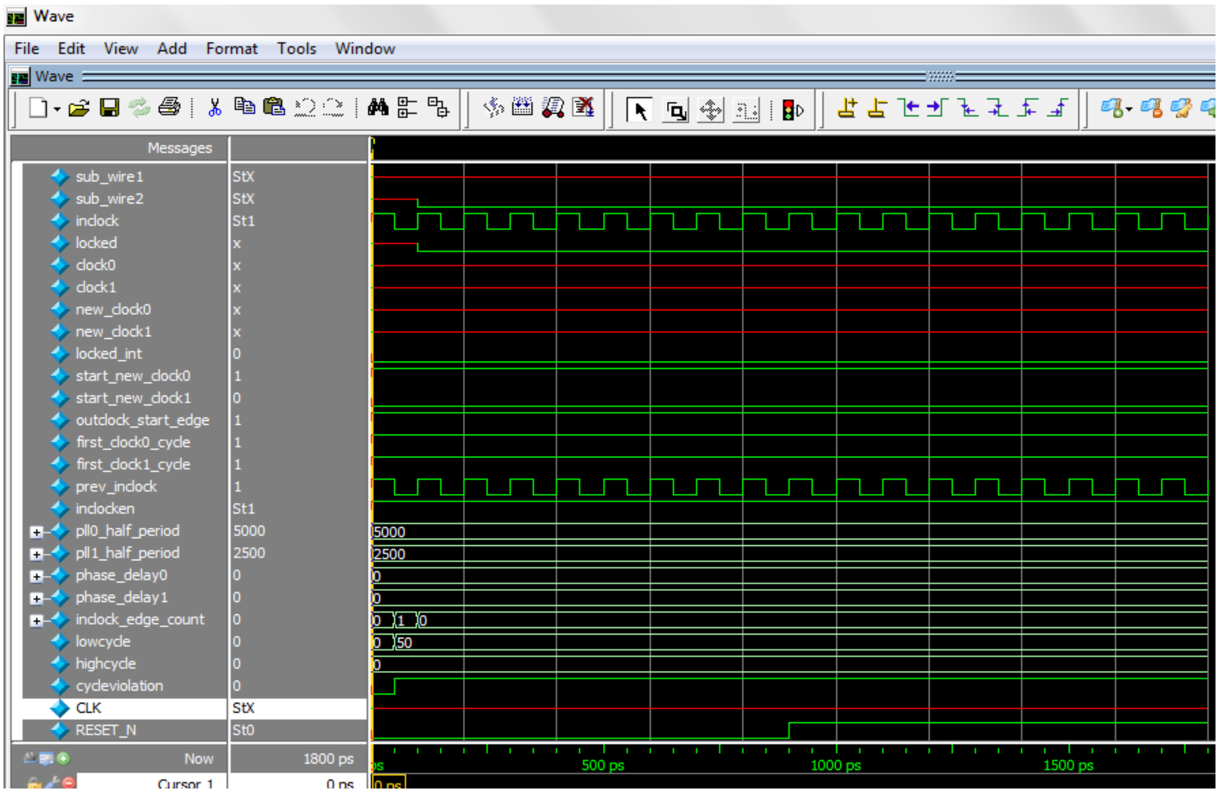
2. Command module: The command module consists of an arbiter, multiplexer and three shift registers. An arbiter is used in case if multiple requests are arriving from different- different devices. If a high- priority and low- priority device both need to transmit data at the same time then the priority is given to high priority device. The low priority device will not be able to transmit data until the high priority device has sent all its data. In this work high priority is given to the REFRESH control operation. If the REFRESH operation is going on and the command from the host arrive then the controller will first continue the refresh operation and hold the command requests by not asserting the command acknowledge (CMDACK). Once the refresh operation is finished the command module will start performing the requested commands from the host. The shift registers are used to maintain the timing between the commands issued to the SDRAM. In addition to this the command module contains a multiplexer which is used to multiplex the address. The row address is multiplexed to SDRAM during the ACTIVATE (RAS) command. The column portion is multiplexed to SDRAM address outputs during a READ (WRITE) command [3].

3. Data path module: The data path module acts as an interface between the processor and the SDRAM. The data module only performs latching and dispatching of the data between the processor/host and the SDRAM. The data path module contains a tristate buffer which is controlled by the OE signal generated by the command module. The data to be written is received on the DATAIN pin during the WRITE operation and during the READA operation data is provided on the DATAOUT pin.

III. Simulation and Synthesis Result

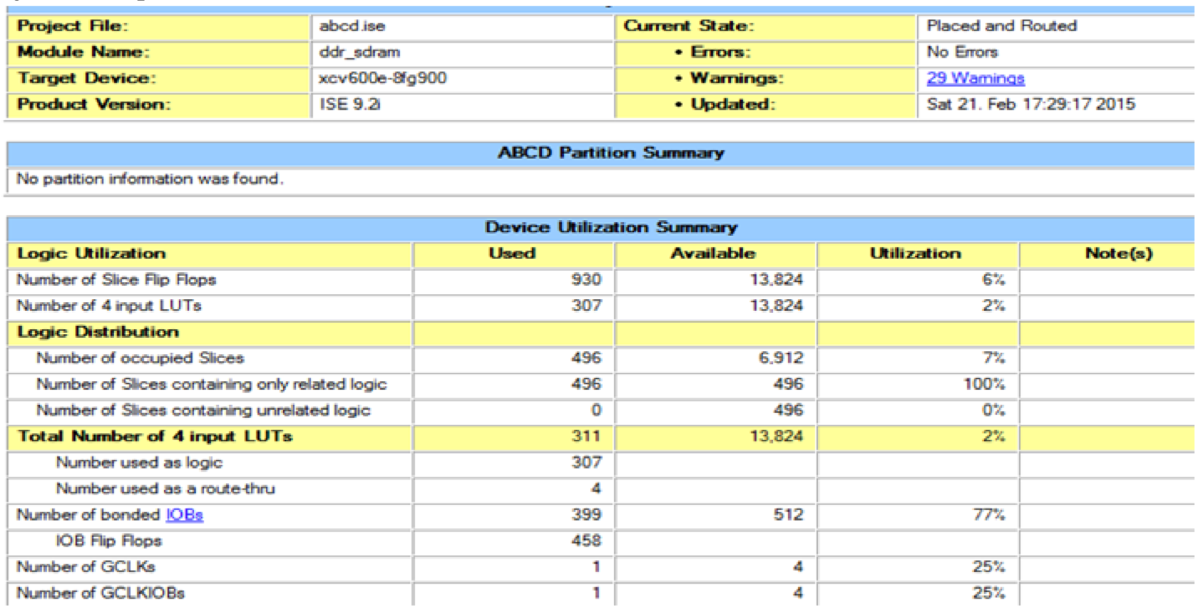
The design is simulated and synthesized on Modelsim 6.5b and Xilinx ISE 9.2i accordingly. The simulation and synthesis results are shown in the following fig.

Simulation result:



Messages	
sub_wire1	StX
sub_wire2	StX
inclock	St1
locked	x
clock0	x
clock1	x
new_clock0	x
new_clock1	x
locked_int	0
start_new_clock0	1
start_new_clock1	0
outclock_start_edge	1
first_clock0_cycle	1
first_clock1_cycle	1
prev_inclock	1
inclocken	St1
pll0_half_period	5000
pll1_half_period	2500
phase_delay0	0
phase_delay1	0
inclock_edge_count	0
lowcycle	0
highcycle	0
cycleviolation	0
CLK	StX
RESET_N	St0
Now	1800 ps
Cursor 1	0 ps

Synthesis report:

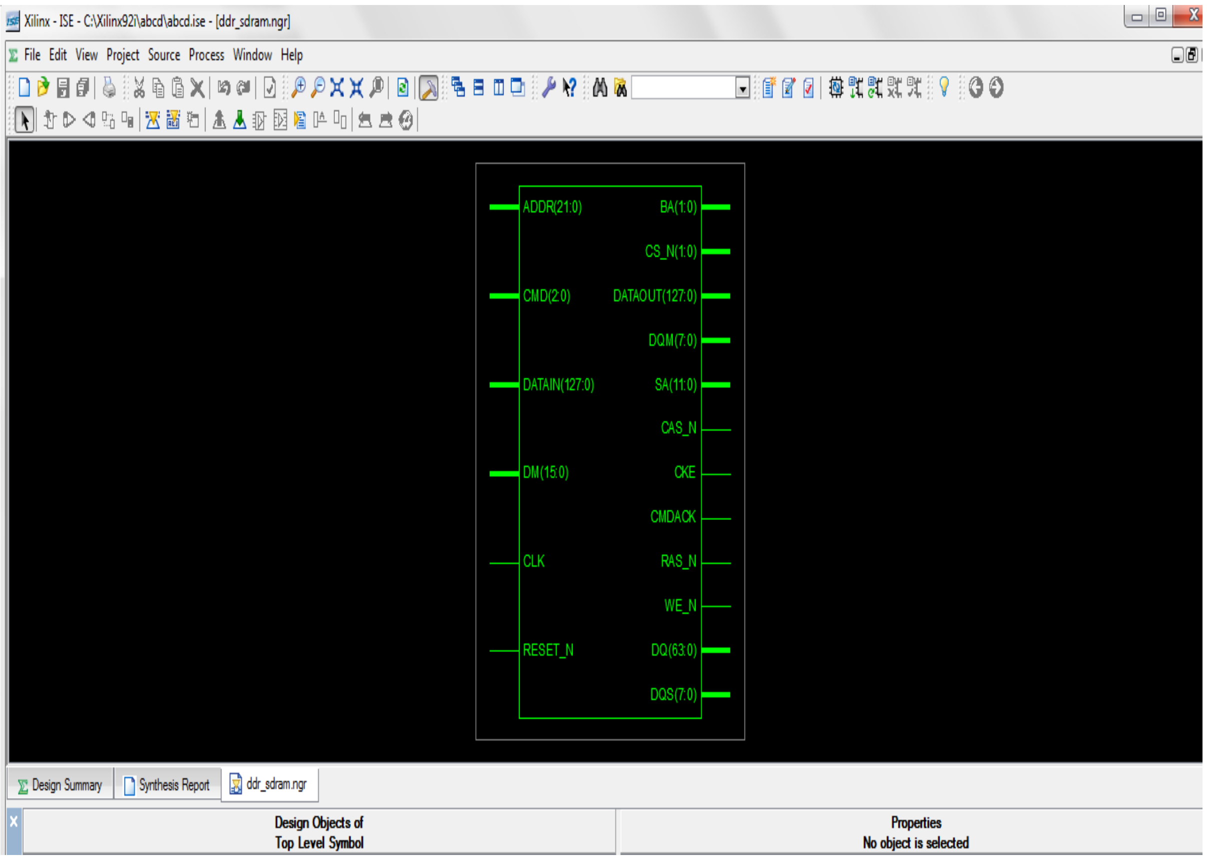


Project File:	abcd.ise	**Current State:**	Placed and Routed
Module Name:	ddr_sdram	• **Errors:**	No Errors
Target Device:	xcv600e-8fg900	• **Warnings:**	29 Warnings
Product Version:	ISE 9.2i	• **Updated:**	Sat 21. Feb 17:29:17 2015

ABCD Partition Summary
No partition information was found.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	930	13,824	6%	
Number of 4 input LUTs	307	13,824	2%	
Logic Distribution				
Number of occupied Slices	496	6,912	7%	
Number of Slices containing only related logic	496	496	100%	
Number of Slices containing unrelated logic	0	496	0%	
Total Number of 4 input LUTs	311	13,824	2%	
Number used as logic	307			
Number used as a route-thru	4			
Number of bonded IOBs	399	512	77%	
IOB Flip Flops	458			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	

RTL Schematic:



IV. Conclusion

DDR SDRAM controller has been implemented using Verilog HDL. High, speed, pipeline and burst access features makes it most popular form of memory used in system designing. Use of DDR SDRAM has two advantages that it reduces the system cost and increases the data transfer throughput. The synchronous interface provides the operation to be pipelined and arranged in queue as they requests. This DDR SDRAM controller is used in embedded system in which high speed is of great concern

References

[1]. Sonali, AshokKumar Patel, SantoshKrishna M., Design of DDR SDRAM controller for embedded system, Journal of Information, Knowledge and Research in Electronics and Communication Engineering, Nov 12 to oct13, volume-02, Issue-02.
[2]. Deepali Sharma, Shruti Bhargava, Mahendra Vucha, Design and VLSI Implementation of DDR SDRAM Controller for high speed applications, International Journal of computer Science and Information Technologies, Vol.2 (4), 2011.
[3]. ALTERA, DDR SDRAM Controller White Paper, Ver1.1, 2002, 8.
[4]. www.xilinx.com
[5]. http://www.latticesemi.com/products/intellectualproperty/referencedesign/ddrsdramcontroller.cfm.
[6]. Wayne Wolf, Computer as Components, Principles of Embedded Computing System Design.
[7]. A K Ray, K M Bhurchandi, Advanced Microprocessors and Peripherals, Architecture, Programming and Interfacing.