Design of SDRAM Memor y Controller using VHDL

Abhishek Kumar
Lovely Professional University

Nisha Sarwade
Veermata Jijabai Technological Institute

International Journal of Computer Science and Application Issue 2010 ISSN 0974-0767

Design of SDRAM Memory Controller using VHDL

Abhishek Kumar, Yogesh. E. Wankhede, Kirti Shinde P., Nisha Sarwade

Abstract- **Today's high-performance CPUs demand high-speed memory. Conventional DRAM technology cannot support the data rates that today's CPUs require. As the bus speed gets faster than 50 MHz, new memory devices are required. Synchronous DRAM (SDRAM) is the new memory for high-speed CPUs. DDR SDRAM is similar in function to the regular SDRAM but doubles the bandwidth of the memory by transferring data on both edges of the clock cycles SDRAM is the most preferable memory for storing large amount of data storage. SDRAM stands for synchronous DRAM in this case all the I/O and control signal of memory is synchronize to clock. In this paper we design a kind of SDRAM controller. SDARM controller provides a synchronous command interface to the SDRAM memory along with several control signals.**

Keywords- **SDARM; Memory read-write;memory control action**

I. INTRODUCTION

With the development of semiconductor industry the speed and capacity of memory device enlarges increasingly from RAM to DRAM and SDRAM. DDR SDARM is an enhancement in traditionally synchronous DRAM. It supports data transfer on both edge of each clock cycle, effectively doubling the data transfer throughput of the memory device. It is widely used on PC for its low cost. Of all DRAMs manufactured today, approximately 70% are used in desktop and notebook PCs, where they are used to provide two different functions: main storage and buffers. The operating frequency of CPU is much higher than memory thus there is a gap exit between CPU and memory. Most PCs are offered with L2 cache to bridge the processor-memory performance gap [2]. This makes the speed of the DRAM memory used for main storage an important but secondary consideration to price. However as multitasking increases with large programs, the frequency of accesses to the L2 cache decreases, this degrades overall system performance, since the processor must wait for the DRAM to supply the requested data. To recover lost system performance, larger L2 cache or faster DRAM main memory are required. Therefore SDRAM is the right main memory choice to increase the system performance.

The features of the SDRAM are [1, 2]

1) Fully synchronous
2) Dual banks
3) Programmable CAS latency: 1, 2, 3 clock cycles
4) Programmable burst length: 1, 2, 4, 8
5) Programmable wrap sequence: sequential or interleave
6) Multiple burst read with single write option
7) Automatic and controlled precharge command
8) Data mask for read-write control
9) Auto refresh or self-refresh

Conventional DRAM is controlled asynchronously. The system must insert wait states (latency cycles to allow the DRAM to catch up with the CPU) to meet the specifications of the conventional DRAMs. Timing depends on the speed of DRAM device used, and is independent of the bus speed. With SDRAM, the performance can be increased by up to 2.7 times that of the fast page mode device (conventional, asynchronous DRAM).

SDRAM provides for programmable READ or WRITE burst lengths of 2, 4, or 8 locations. An auto precharge function may be enabled to provide a self timed row precharge that is initiated at the end of the burst access this model has implemented in RTL by VHDL. The focus of this work is to implement behavioral model of SDRAM. The Top level model is as shown in Fig.1. The core contains mainly two parts, AHB Slave and DDR SDRAM controller.

II. ARCHITECTURE OF SDRAM CONTROLLER

Fig.1 shows the block diagram of memory controller it consist of three component command module, signal generation module and data path module.

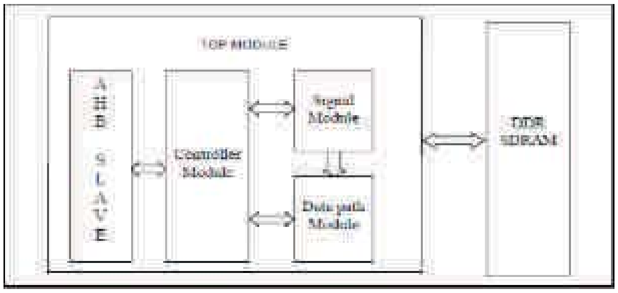


Fig.1 Top Module

Command module consists of sub module initialization and command generation. Initialization module initializes the memory before use it. Initialization start with high on input INIT_START. It also provides the control information for command and signal generator module. Command module accepts the command like read, write, active, refresh and precharge from user and generate internal signal CSTATE. All the output of initialization and command module are internally coded in the form of ISTATE and CSTATE respectively. Depending on the value of ISTATE and CSTATE signal generator generates the value at the output. Signal WR_EN to enable write and read operation in data path module, to activate the data transfer between CPU and memory at data transfer states [1].

A memory controller regulates the data transfer between CPU and memory. All the command and address are the input of controller and it generates the necessary control signal for addressing the memory. SDRAM is organized in bank, row and column architecture. Bank row and column address are multiplexed into a single address bus “ADDRESS”. The entire control signals are synchronized with clock.

III. BASIC FUNCTIONALITY [3]

The SDRAM is a high-speed CMOS, dynamic random-access memory. It is internally configured as a dual bank DRAM. The DDR SDRAM uses double data rate architecture to achieve high-speed operation. The double data rate architecture is essentially 2n prefetch architecture with an interface designed to transfer two data words per clock cycle at the I/O pins. A single read or write access for the DDR SDRAM effectively consists of a single 2*n*-bit wide, one clock- cycle data transfer at the internal DRAM core and two corresponding *n*-bit wide, one-half clock-cycle data transfers at the I/O pins. A bidirectional data strobe (DQS) is transmitted externally, along with data, for use in data capture at the receiver. DQS is a strobe transmitted by the DDR SDRAM during READs and by the memory controller during WRITEs. DQS is edge-aligned with data for READ and center-aligned with data for WRITE.

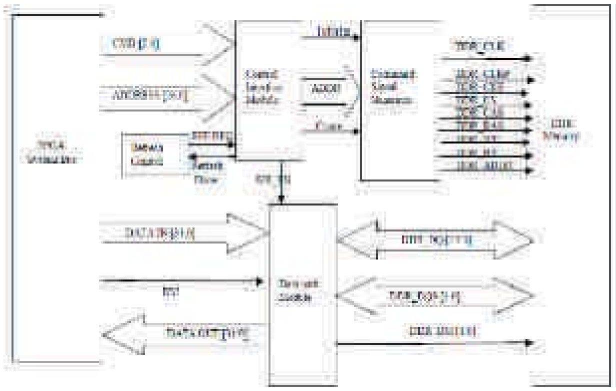


Figure 2. Block Diagram of SDRAM controller

+Accesses begin with the registration of an Active Command, which is then followed by a Read or Write command. The address bits registered coincident with the Active command are used to select the bank and row to be accessed (BA0-BA2 select the bank; A26-A13 select the row). The address bits registered coincident with the Read or Write command are used to select the starting column location for the burst access and to determine if the auto precharge command is to be issued.

Fig.2 shows the block diagram of SDRAM controller. Input contains the necessary signal for bus system and output side contains the signal for addressing the memory. At the input a multiplexed 27 bit addresses bus. RD_WR indicates type of access read or write, for read it would be '1'. A three bit command input indicates the input command defined by user. 'Burst' input indicates whether the input is single data transfer or burst data transfer. With 14 rows bit address and 2 bit bank address we can address 512 MB SDRAM. An active command moves row from the memory array into row buffer, thereby opening the row. Once the row has been opened any number of Read- Write command issued to transfer data on into and out of the row. A precharge command closes a row, restores it back to the memory array and precharges the bank for next row activation.

1) Register Definition

Fig.3 shows scheme of address bus multiplexing. A 27 bit multiplexed address bus consists of address of bank row and column.

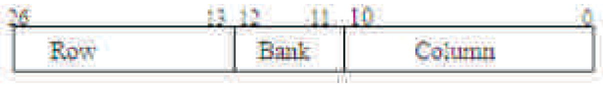


Figure.3 Address bus multiplexing

SDARM is organized as a set of rank that consists of independent memory bank. 2 bit bank can address maximum 4 no. of bank. Each memory bank consists of independent 2D memory cell.

2) Bank Address Bit Configuration:

Bank address bit BA0 and BA1 used to select the MR and EMR register. Both bits BA0 and BA1 must be decoded for Mode/Extended Mode Register Set (MRS/EMRS) Commands. Users must initialize Mode Registers.

Table1: Mode Selection of SDRAM[5]

BA1	BA0	Mode Register
0	0	MR
0	1	EMR
1	0	EMR2
1	1	EMR3

3) Mode Register

The mode register stores the data for controlling the various operating modes of SDRAM. It controls CAS latency, burst length, burst sequence, test mode, DLL reset, WR and various vendor specific options to make SDRAM useful for various applications. The Mode register is used to define the specific mode of operation of the DDR2 SDRAM. Fig.4 shows the Mode register features used by the controller. Bank Addresses BA1 and BA0 select the Mode registers[5].

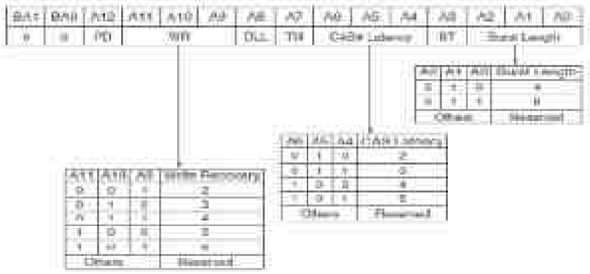
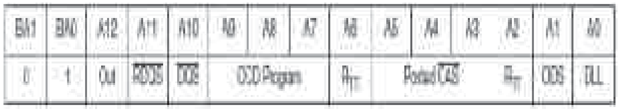


Figure 4. Programming of mode registers [5]

4) Extended Mode Register

In addition to the functions controlled by the Mode register, the Extended Mode register controls these functions: DLL enable/disable; output drive strength (ODS); on-die termination; posted CAS additive latency (AL); off-chip driver impedance calibration (OCD); differential DQS enable/disable; RDQS enable/disable; OCD enable/disable and output disable/enable as shoen in fig5.



BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	Out	RDQS	DQS	OCD Program			R_{TT}	Posted CAS			R_{TT}	ODS	DLL

Figure 5. Programming of extended mode registers [2,5]

5) Extended Mode Register 2 (EMR2)

Bank Address bits are set to 10 (BA1 is set High, and BA0 is set Low). The address bits are all set to Low.

6) Extended Mode Register 3 (EMR3)

Bank Address bits are set to 11 (BA1 and BA0 are set High). Address bits are all set to Low, as in EMR2.
Extended mode register 2 and 3 are kept reserved by JEDEC (Joint Electron Device Engineering Council)[5].

IV. DIFFERENT MODULE

A. Initialization Module

Prior to normal operation, the SDRAM must be initialized. SDRAMs must be powered up and initialized in a predefined manner. The PHY layer of SDRAM executes a JEDEC-compliant initialization sequence for memory. Fig6. Shows the memory initialization sequence executed for the physical layer during initialization of memory certain bit values programmed to the Mode register (MR) and Extended Mode register (EMRS), such as for the burst length, CAS latency, and additive latency, are configurable in the design and determined by the value of top-level HDL parameters. During the initialization the mode register and extended mode register are initialized with user defined value of burst length, CL latency, AL latency and other parameter.

Initialization process provides the necessary signal (ISTATE) to the signal generator. Initialization starts with *IDLE* followed by *NOP* and *precharge all* signal. EMRS are initialized with specific bank address value.

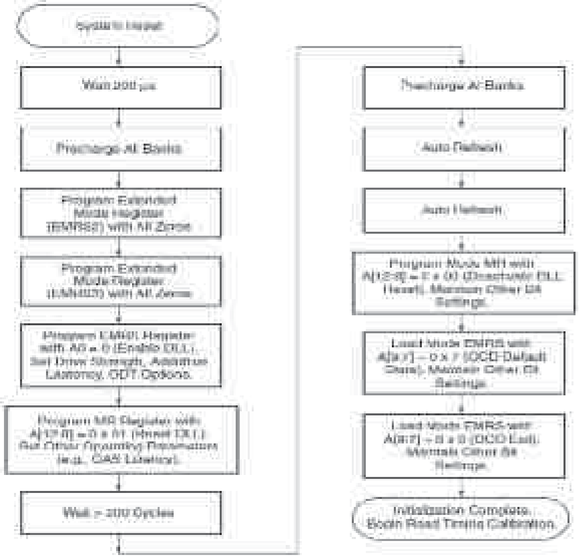

System Reset
Wait 200 µs
Precharge All Banks
Program Extended Mode Register (EMRS2) with All Zeros
Program Extended Mode Register (EMRS3) with All Zeros
Program EMR1 Register with A0 = 0 (Enable DLL), Set Drive Strength, Additive Latency, ODT Options.
Program MR Register with A[12:8] = 0 x 01 (Reset DLL), Set Other Operating Parameters (e.g., CAS Latency).
Wait > 200 Cycles
Precharge All Banks
Auto Refresh
Auto Refresh
Program Mode MR with A[12:8] = 0 x 00 (Deactivate DLL Reset), Maintain Other Bit Settings.
Load Mode EMRS with A[9:7] = 0 x 7 (OCD Default State), Maintain Other Bit Settings.
Load Mode EMRS with A[9:7] = 0 x 0 (OCD Exit), Maintain Other Bit Settings.
Initialization Complete. Begin Read Timing Calibration.

Figure 6. Initialization sequence of SDARM [4]

Mode register is initialized with value "0342" to reset DLL and programmed the other value like CAS latency. Onwards all bank precharged and generates two refresh signal. Again it loads mode register with value "0242" to reset DLL. EMR is loaded with value "2380" for default calibration and "2000" to *ocd exit*. Once the MR and EMR are initializing with certain value it will remain same till end. To change the values have to re-initialize again[1]. After initialization completed it generates INIT_DONE signal to the command module, it shows that memory is initialized and it is ready for reading or writing.

Fig.7 shows the simulated result of initialization. All the simulation has been performed using Xilinx simulation tool. Initialization sequence (ISTATE) follows as 0-1-2-3-4-5-6-7-8-9-A-B-C-D which is the coded value of Idle, Nop,PrecharfeAll,EMR2,EMR3,EMR,MR,PrechargeAll, Refresh1,Refresh2,Reset DLL,OCD Default, OCD Exit, Ready. After Ready state signal INIT_DONE indicate initialization complete.

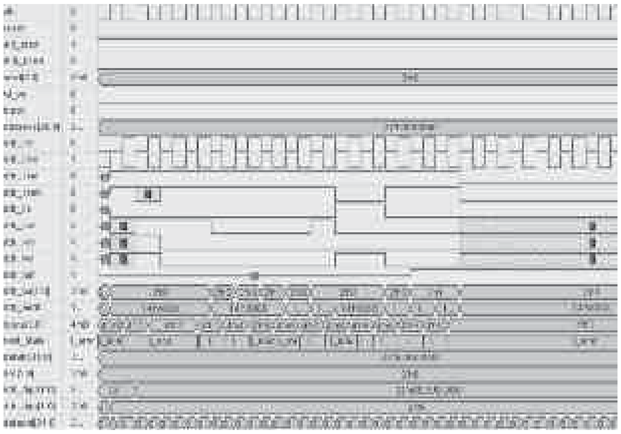


Figure.7 Waveform of initialization cycle

B. Command Module

In order to improve the whole system's performance, a SDRAM controller has been designed to make sure the SDRAM normally and efficiently working. Command module accepts the command from user and generates the internally coded signal CSTATE and WR_EN. Command module is based on the finite machine based operation; all the states are synchronizes with a single clock. CSTATE indicates the different state. WR_EN signal is used to enable read and write control circuitry. WR_EN is high for writing in memory low for during reading from memory, data transfer takes place in either direction depending on the RD_WR signal. If RD_WR is high read occurs and data read from memory to processor, while RD_WR ids low write occurs and data write from processor to memory. SDRAM can transfer data in burst. If burst Input at the command is low means a burst of 4 otherwise 8. In this design a burst of 8 is tested. During burst transfer no other command are accepted except NOP. During burst transfer WR_EN & RD_EN signal will became high and it enable the write and read data path respectively. After completion of data transfer a precharge signal is issued to close the bank [2, 4].

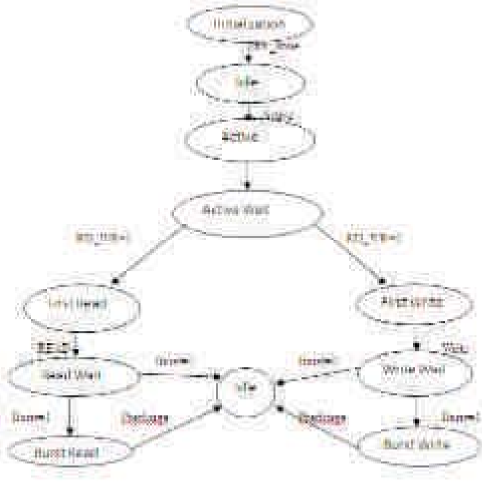


Figure 8. State machine of command module

The Finite State Machine of command generation module is shown in fig8[3,4]

1) *Initialization*: The memory device intilized. In this state the SDRAM will be precharged, refreshed and all row are in the closed status after the intilization. The next state is IDLE state.

2) Idle: In this state the system will wait until a memory access is initiated. On registration of ACTIVE command bank and row address is sent to the signal generation module. Control module must wait on this state to opening the row. Next state is active state.

3) Active: In this state the system will come in active mode and it issued bank address and row address only NOP command is allowed in this state. The next state will be active wait.

4) Active Wait: in this state RAS (row activation strobe) command will be sent to the SDRAM. Next state will be first read or first write depending on the value RD_WR.

5) First Read: In this state the CAS (column activation strobe) command will be sent to the SDRAM. WR_EN signal o high .*Read_wait* is the next state.

6) Raed Wait: In this state the SDRAM Mode Register will be set to a burst length of 4 or 8 depending on the burst input. If burst is '1' next state will be burst read else Idle.

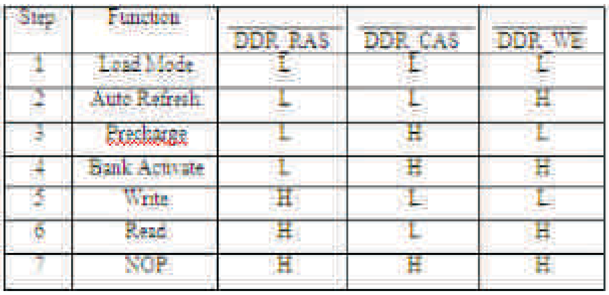
7) First Write: In this state the CAS command will be sent to the SDRAM. WR_EN signal, write_*wait* is the next state.

8) Write Wait: In this state the SDRAM Mode Register will be set to a burst length of 4 or 8 depending on the burst input. If burst is '1' next state will be burst write else Idle.

9) Burst Read/burst Write: In this state burst data transfer takes place. WR_EN became high to enable the read –write path. A precharge command is issued to close the bank. Next state always is idle.

C. Signal Generator

This module accept the command from initialization (Istate) and command module (Cstate) and activate the DDR_CAS, DDR_RAS, DDR_WE signal at the output upon the detection of initialization and command state. fig.9 shows the generated signal. DDR_CKE signal always remain high to enable the data transfer between controller and memory. DDR_CS always remain low to enable the memory device.



Step	Function	DDR_RAS	DDR_CAS	DDR_WE
1	Load Mode	L	L	L
2	Auto Refresh	L	L	H
3	Precharge	L	H	L
4	Bank Activate	L	H	H
5	Write	H	L	L
6	Read	H	L	H
7	NOP	H	H	H

Figure 9 Signal generation

D. DATAPATH

The data flow design between the SDRAM and the system interface. Fig10 shows the diagram of controller's datapath. The module in this reference design interfaces between the SDRAM with 16-bit bidirectional data bus and the bus master with 32 bit data bus. The data path module performs the data latching and dispatching of the data between the processor and SDRAM. This module accepts a data of 32 bit and convert into two 16 bit DDR data and issue the data at both rising and falling edge of clock.

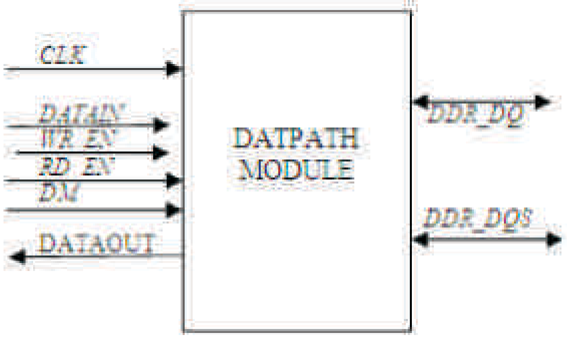


Figure 10 data path

DATAPATH functions with enable signal WR_EN. During writing a 32 bit data SDR data is converted into 16 bit DDR data, while reading 16 bit DDR data is converted into 32 bit SDR data. Read is always burst oriented, DQS signal will increases on each byte of read or write access. On rising and falling edge of clock we can access byte of data which doubles the bandwidth and system performance.

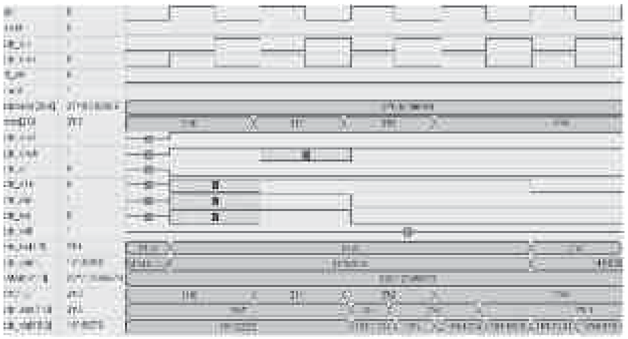


Figure 11 waveform of write cycle

Fig.11 shows the simulation result of write cycle. Controller is initialized at first. Then it sends command 2 which is Burst Write timing and at this time, data given to the input port DATAIN is “12345678”. Controller writes data at the port DDR_DQ is “1234” at the rising edge of clock and “5678” at the falling edge of clock at the address of bank '1' and row address “005B”. On writing each byte signal ddr_dqs will increase by one. The simulation result demonstrates the correctness of the design.

Fig 12 shows the RTL schematic of top module. All the design is simulated and tested in vertex5 Xilinx ISE simulator.

```
HDL Synthesis report of the core as shown below
Macro Statistics
# FSMs                                  : 2
# Registers                             : 44
 Flip-Flops                             : 44
# Latches                               : 14
 1-bit latch                            : 1
 16-bit latch                           : 7
2-bit latch                             : 1
32-bit latch                            : 1
8-bit latch                             : 4
Minimum period                          : 2.653ns
(Maximum Frequency                      : 376.932MHz)
Minimum input arrival time before clock: 3.333ns
Maximum output required time after clock: 3.597ns
```

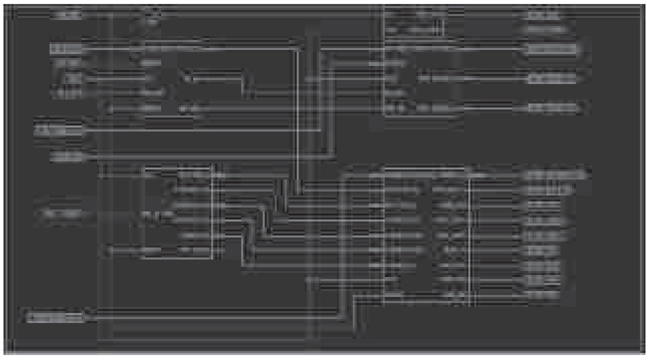


Figure12. RTL Schematic of Top module

```
-------------------------------------
Selected Device: 5vlx30ff324-3
-------------------------------------
Slice Logic Utilization:
Number of Slice Registers        :186 out of 19200    0%
Number of Slice LUTs             :174 out of 19200    0%
Number used as Logic             :174 out of 19200    0%

Slice Logic Distribution:
Number of Bit Slices used        :208
Number with an unused Flip Flop:     22   out of   208   10%
Number with an unused LUT        :34   out of   208   16%
Number of fully used Bit Slices  :152 out of   208   73%

IO Utilization:
Number of IOs                    :161
```

Number of bonded IOBs	:159 out of 220	72%
IOB Flip Flops/Latches	:34	

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs	: 3 out of 32	9%

V. CONCLUSION

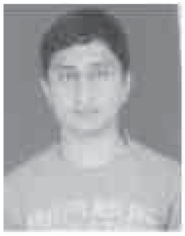
In this paper a fully functional SDRAM controller is designed, with structure proven to be efficient. The presented controller has two main control schemes. Command generation and signal generation, simultaneously these modules provide signal to the data path for data transfer. Control module and data module can be utilized free of charge. Memory system operates at double the frequency of processor, without affecting the performance we can reduce the data bus size. The disadvantages of this controller at half the frequency using a more complicated scheme and large number of buffer leads to a noticeable increase in delay. This paper describes the possible implementations of an SDRAM controller using a XILINX Vertex 5 FPFA device.

REFERENCES

1. High-Performance DDR2 SDRAM Interface in Virtex-5 Devices XAPP858 (v2.1) May 8, 2008
2. LatticeSC/M DDR/DDR2 SDRAM Memory interface User's Guide,TechnicalNoteTN1099, July 2008
3. Implementing a Synchronous DRAM Controller in Cypress CPLDs, Cypress Semiconductor Corporation CA 95134 408-943-2600 July 15, 1999
4. VHDL for Programmable Logic, Kevin S kahill, Cypress Semiconductor
5. JEDEC Solid State Technology Association. DDR2 SDRAM Specification.
6. Double Data Rate (DDR) SDRAM Specification, JEDEC STANDARD JESD79E, May 2005

Abhishek kumar
M.Tech. (Student)
V.J.T.I. (Mumbai)
abkvjti@gmail.com

Kirti Shinde P
M.Tech. (Student)
V.J.T.I.(Mumbai)
kirti.shine@gmail.com



Yogesh E. Wankhede
M.Tech. (Student)
V.J.T.I. (Mumbai)
yogesh17.wankhede@gamil.com

Dr. Nisha Sarwade
Professor
V.J.T.I.(Mumbai)
nishasarvade@vjti.org.in