IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 3, Issue 6, January 2015)

Design of Control Interface Module for DDR SDRAM Controller using Verilog HDL

Mayuri Sanwatsarkar[1] and Jagdish Nagar[2]
Acropolis Institute of Technology & Research, Indore (India)
[1]mayuri.sanwatsarkar@yahoo.com
[2]jagdishnagar@acropolis.in

***Abstract*—An embedded system needs memory for two purposes either to store the program or to store its data. Embedded systems are the electronic systems that contain a microprocessor or a microcontroller, but we do not think of them as computer – the computer is hidden or embedded in the system. As microprocessors have become cheaper and smaller, more and more products have microprocessor embedded in them to make the devices smart. Products such as VCRs, digital watches, elevators, automobile engine, laser printer, industrial control equipments, scientific and medical instruments etc. are driven by these microprocessors and their software. A microprocessor has two essential units to perform the overall system operation: one is control interface unit and other is execution unit. All the electronic devices consist of a control interface module embedded in them to control the operation of that device. The control interface module accepts the commands from processor, decode the commands and passing the request to the peripheral devices and the peripheral devices then perform the operation. In this work we have designed a control interface module for DDR SDRAM (dual data rate Synchronous DRAM) controller using Verilog HDL and simulation and synthesis is done by using Modelsim and Xilinx ISE accordingly.**

***Keywords*—Embedded system, embedded, control interface module, DDR, SDRAM.**

I. INTRODUCTION

These days, we find that the field of electronics has spread so vast. It has become the most important part of our life, as it has made our life easier and convenient, we can't think of our life without it. It has penetrated into our homes, our places of work and our means of communication from one place to another. The electronic devices and gadgets are also used in almost all industries for quality control and automation. It also plays an important role in defence applications. These days, we have aeroplanes, ships, anti-aircraft guns and guided missiles are completely controlled by electronics.

Electronics is the field in which rapid developments are taking place every day. The developments which have taken place in last 35 years may be understood from the fact that today's microcomputer is 1000 times faster, has a large memory, thousand times more reliable, consumes less volume, consumes negligible power and cost is also very less as compared to the first large electronic computer. Computers are used in scientific calculation, at homes and in offices for processing the data, in educational institutes and in many other areas. Generality is the most striking property of digital computer. It follows the sequence of instruction called program which operates on given data. The user can change the program and/or data according to the need. And to store this program and data we need memory [9]. A general purpose computer is the best known example of a digital system. A digital system mainly consists of two parts, the data path circuit and the control circuit. The data path circuit is used to store and manipulate data and it also transfer data between the peripheral devices. Data path circuit comprises of building blocks such as shift registers, counters, multiplexers, adders, decoders and so on. The control circuit controls the operation of data path circuit.

As microprocessors are getting faster every year, it is also necessary to improve memory architecture in order to enhance the overall system performance. Several types of memories are there, but DDR SDRAM is most commonly used in various embedded application. The DDR SDRAM supports burst access, high speed and pipelining features, due to this DDR Synchronous DRAM (SDRAM) has become more popular memory of choice in design. DDR SDRAM uses double data rate architecture to achieve high speed data transfers [11]. DDR SDRAM transfers data on both the rising edge and the falling edge of the clock. DDR SDRAM employs a differential clock (CLK) and data strobe signal (DQS) to achieve high data transfer. DQS is synchronized with the clock, and input/ output data (DQ) is synchronized with both the rising and falling edge of the clock [11].

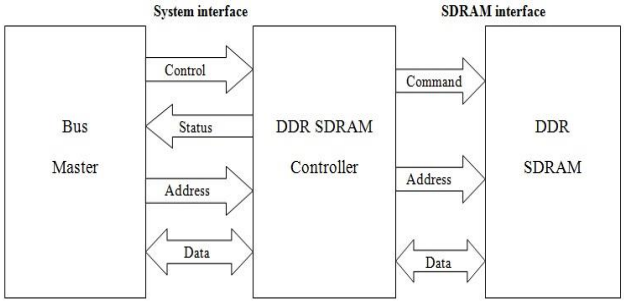


Fig.1 DDR SDRAM Controller system interface

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 3, Issue 6, January 2015)

For some applications which require the processor; the access to DDR SDRAM is supported by processor's in-built devices. But, for high-speed applications, we must need to design a controller to provide the proper commands for accessing SDRAM and for memory refresh. This DDR SDRAM controller design is located between the processor or the bus master and SDRAM; provides direct access to memory hence reduces the efforts and improves the speed; as shown in fig.1 [6].

DDR SDRAM controller system: The block diagram of DDR SDRAM controller is shown in fig.2 [6]. DDR SDRAM controller consists of three modules: the control interface module, the command module or signal generator, and data path module. Control interface module; is the primary module; contains finite state machines and a counter used to perform refresh operation. The command module generates the command signals and address required for SDRAM. The datapath module performs the data latching and dispatching between the bus master and DDR SDRAM [11].

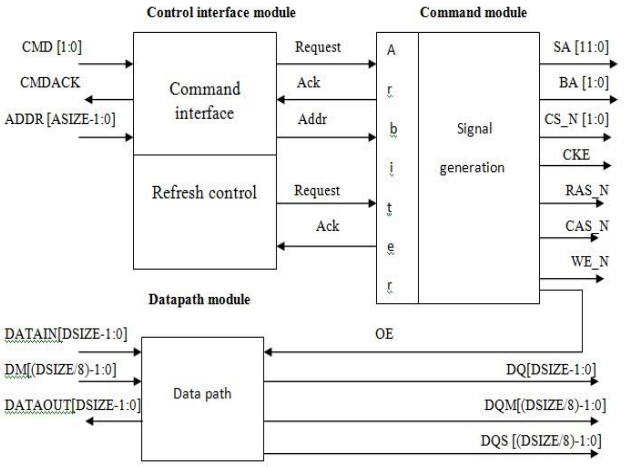


Fig.2 Block diagram of DDR SDRAM controller

Control interface module: A control interface module accepts the commands from embedded system and decoding the command and provides the request to the command module. The control interface module accepts commands from the host decode them and passing the decoded REFRESH, NOP, PRECHARGE, READ/WRITE and LOAD_MODE commands with corresponding ADDR to signal generation module or command module.

The LOAD_MODE defines the mode in which the SDRAM operates. The LOAD_REG1 command is decoded and used to load the value stored at ADDR of REG1and LOAD_REG2 command is decoded and used to load the register REG2 with the values from ADDR.

Since SDRAM needs to be refreshed to prevent its data. So control interface module uses a down counter (16-bit) to generate periodic refresh commands to the SDRAM. The value of 16- bit down counter is set by the values from REG2 which counts from 15 and reaches down to zero. As the counter reaches to zero the REFRESH_REQ output is asserted again and it remains asserted til any acknowledgement is issued from the command module. The 16-bit value of REG2 defines the time between REFRESH commands which the controller issues and this value is set by the equation int(REQ_PERIOD/ CLK_PERIOD) [6].

Some bus commands are used to control the operations of SDRAM. These bus commands are formed by the combinations of the WEN, RASN, and CASN signals. Suppose, for instance, on any clock cycle if all the signals are high then associated command is NOP (No Operation). A NOP is also indicated when chip select is not asserted. The standard bus commands are shown in table 1 [6].

SDRAM banks should be opened before addresses can be written to it or read from it. The row and bank to be accessed can be opened by the ACT command. It is necessary to close a bank and re-open it if the row or bank to be accessed is different from the row that is currently opened. Closing of a row/ bank is done by the precharge (PCH) command.

TABLE 1
SDRAM bus commands

Command	RASN	CASN	WEN
No Operation (NOP)	1	1	1
ACTIVE (ACT)	0	1	1
READ (RD)	1	0	1
WRITE	1	0	0
Burst terminate (BT)	1	1	0
Precharge (PCH)	0	1	0
Auto Refresh (ARF)	0	0	1
Load mode register (LMR)	0	0	0

READ (RD) and WRITE (WR) are the basic commands needed to access SDRAM. During the WRITE command, column address and data is registered. During the READ command, the initial address is registered. During the RD operation, the data appears 1-3 clock cycles later on the data bus. This time delay is known as CAS latency (CL) and this delay is due to the time needed to read the DRAM and to enter the data on bus. Speed of the SDRAM and frequency of clock are the things on which CAS latency depends. The READ or WRITE operation will continue until the burst length is reached or burst terminate (BT) command is issued [6]. DDR SDRAM comes in CAS 2 and CAS 2.5 ratings,

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 3, Issue 6, January 2015)

with CAS 2 costing more and performing better and it supports the burst lengths of 2, 4, or 8 data cycles.

DDR SDRAM controller command interface: DDR SDRAM controller uses several control registers provides a synchronous interface to SDRAM. The commands used to control SDRAM are listed in table 2 with detailed description in the following sections [6].

The controller registers the commands on the rising edge of the clock. The controller acknowledges the command by assert CMDACK for one clock cycle. During WRITE operation, the user starts writing data on the DATAIN ports. During the READ operation the user will receive data on DATAOUT port.

TABLE 2
Interface commands

Command	Value	Description
NOP	000b	No Operation
READA	001b	SDRAM Read with auto precharge.
WRITEA	010b	SDRAM write with auto precharge.
REFRESH	011b	SDRAM auto refresh.
PRECHARGE	100b	SDRAM precharge all banks.
LOAD_MODE	101b	SDRAM load mode register.
LOAD_REG1	110b	Load controller configuration register.
LOAD_REG2	111b	Load controller refresh period register.

Functional description:

NOP command: The No Operation (NOP) is a command used to instruct the controller to perform a NOP in the following clock cycle. Operations that are already in progress are not affected by the NOP command.

READA command: The READA command is used to instruct the controller to perform burst read operation. The controller will issue an Active (ACT) command on the SDRAM followed by the READA command. The burst data will appear on DATAOUT port (RC+CL+4) clock cycles later as the controller has asserted command acknowledge (CMDACK), where CL is the integer portion of the actual CAS latency (i.e. if CAS latency is 2.5 then CL=2) [6]. The burst read operation continues until the burst length is reached or the burst terminate command is inserted.

WRITEA command: The WRITEA command is an instruction to controller to perform a burst write operation. The controller issues an ACTIVATE command followed by a WRITEA command to SDRAM. The host can start writing data after it has acknowledges the WRITE command.

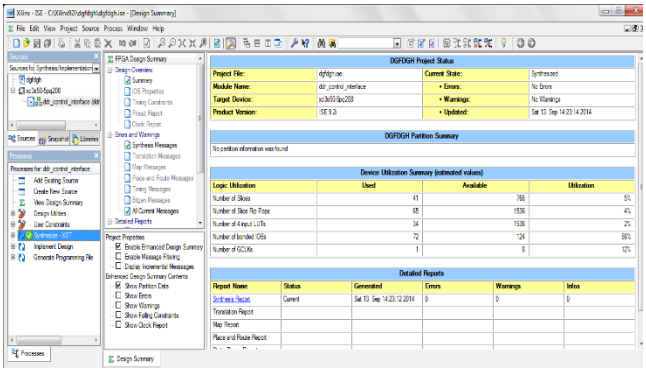
REFRESH command: SDRAM is a volatile memory and to retain its data it needs to get refresh periodically. The REFRESH command is used to instruct the controller to perform an Auto REFRESH operation. The controller will acknowledge the command with CMDACK. The auto refresh command is periodically issued to avoid data retention.

PRECHARGE command: The precharge (PCH) command is used to instruct the controller to perform a PRECHARGE command. The precharge command is used to close an open row or a bank. The controller acknowledges the command by sending CMDACK for one clock cycle.

LOAD_MODE Register command: The Load mode register command is used to define the specific mode of the SDRAM mode register. The register stores the burst type, CAS latency, burst length and burst write mode [6]. There are two control registers REG1 and REG2. REG2 is used to provide an auto refresh operation.

Conclusion and Result: The Control interface module designed for DDR controller is developed by Verilog and simulated on Modelsim 6.5b and synthesized on Xilinx ISE 9.2i. DDR SDRAM is a volatile memory device. When the power is removed, all contents and operating configurations are lost. In this project we have concentrated on implementation of only one module (control interface module) out of the three modules. In future we will design the other two modules i.e., signal generation module and data path module and interface these three modules to design the controller. This memory controller can be used in FPGA base embedded system design to achieve high speed. The advantages of this controller are synchronizes the data transfer and burst data transfer. The core is developed by using Verilog HDL [4].

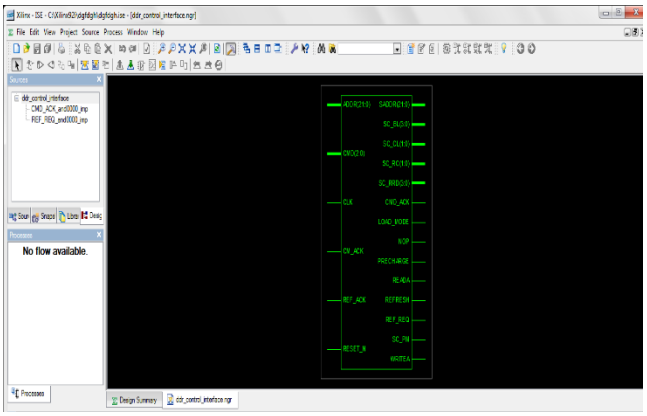
Synthesis Report:

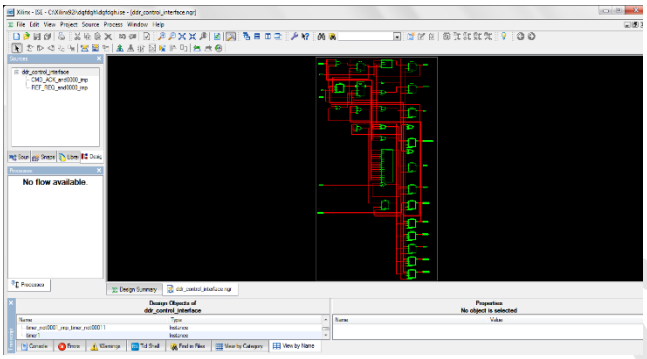


IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

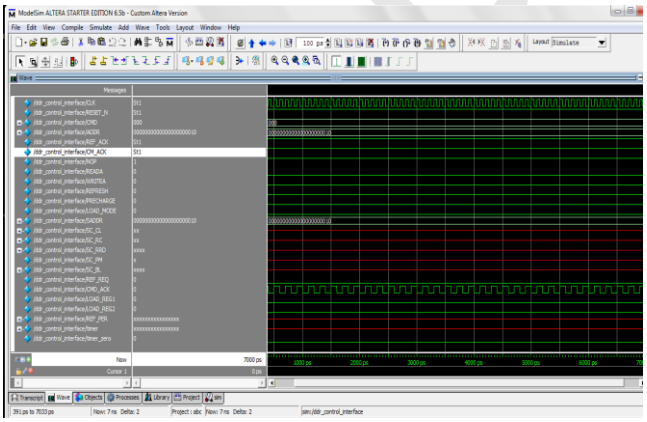
International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 3, Issue 6, January 2015)



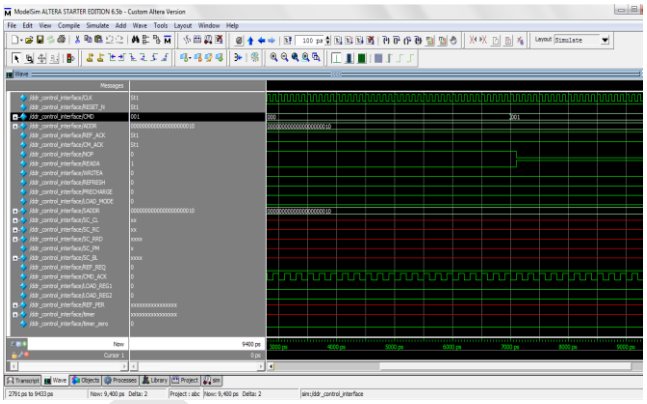


Simulation Result: The Control interface module designed for DDR controller is developed by Verilog and simulated on Modelsim 6.5b and synthesized on Xilinx ISE 9.2i. Simulation results for NOP, Read (RD), Write (WR), REFRESH, Precharge, LOAD_MODE, LOAD_REG1 and LOAD_REG2 cycles is shown in following fig. below.

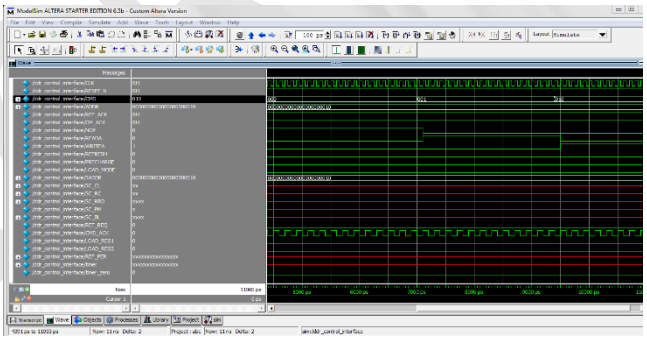
a) NOP Operation:



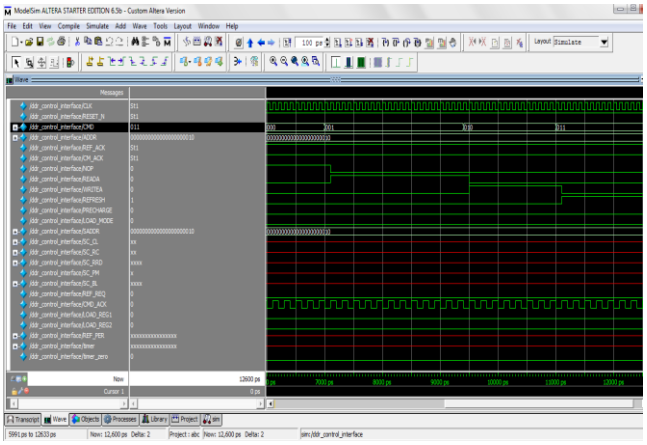
b) READA operation:



c) WRITEA operation:



d) Refresh operation:

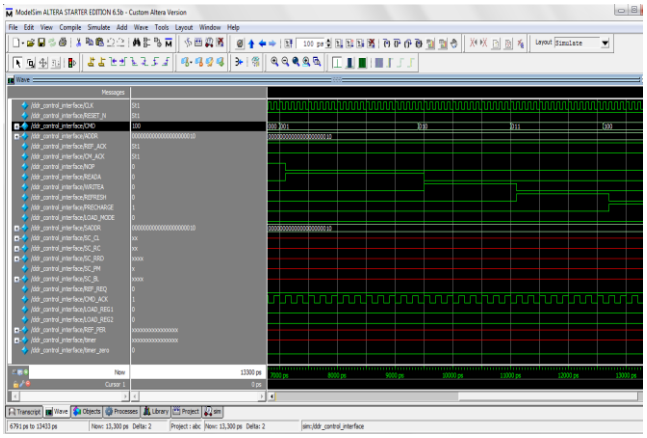


IJDACR
ISSN: 2319-4863

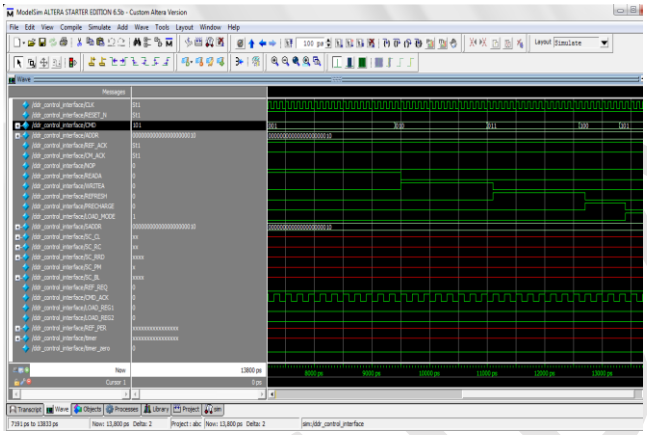
IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 3, Issue 6, January 2015)

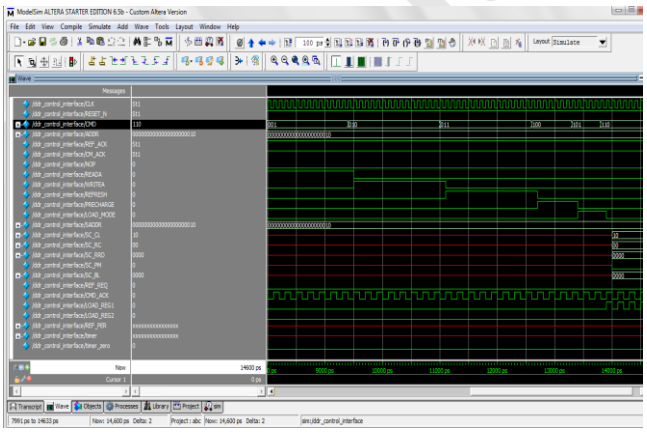
e) Precharge Operation:



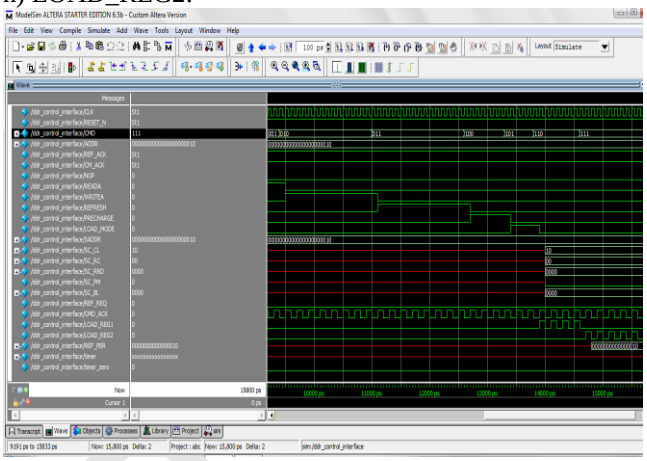
f) LOAD_MODE Operation:



g) LOAD_REG1:



h) LOAD_REG2:



References

[1] Zhang Bo, Zhang Gang "Design of Dual- Processor sharing DRAM controller" Second International Workshop on Computer Science and Engineering 2009

[2] DDR SDRAM memory Datasheet of 512 Mb_DDR_x4x8x16 from Micron Technology, Inc.

[3] Samir Palnitkar, Pearson 2nd edition "Verilog HDL, A Guide to digital Design and Synthesis".

[4] User's Manual E0234E50 (Ver.5.0) from Elpida Memory, Inc. 2002-2007.

[5] ALTERA, DDR SDRAM Controller White paper, Ver1.1,2002, 8.

[6] http://www.latticesemi.com/products/intellectualproperty/referencedesigns/ddrsdramcontroller.cfm

[7] Double Data Rate (DDR) SDRAM Specification, JEDEC STANDARD, JESD79E, May 2005.

[8] M. Morris Mano, PEARSON Education "Digital Logic and Computer Design.

[9] Sonali AshokKumar Patel, Santosh Krishna M. "Design of DDR SDRAM Controller for Embedded system" Journal of Information, Knowledge and Research in Electronics and Communication Engineering, Nov12 to Oct13, volume-02, Issue-02.

[10] Deepali Sharma, Shruti Bhargava, Mahendra Vucha "Design and VLSI Implementation of DDR SDRAM controller For High Speed Applications" IJCSIT, Vol.2 (4), 2011, 1625-1632.