Implementation and Verification of Memory Controller Using System Verilog

Darshan V[1], Dr. Radha B.L[2]

[1]*Student, VLSI Design and Embedded Systems, Bangalore Institute of Technology, Bengaluru, India*

[2]*Associate Professor, Bangalore Institute of Technology, Bengaluru, India*

Abstract:-The performance of the memory is the main area of the computer system that needs to be improved. Memory controllers enable efficient data control between the processor and memory. As a result, numerous techniques are being used in modern computer systems by different researchers to enhance memory capacity. One of these initiatives was the introduction of Memory Controllers (MC), a reliable data control intermediary between the processor and memory. Any design path must include verification because it is completed before silicon growth. Thus, we connect ROM, FLASH, SSRAM, and SDRAM to create the memory controller. It is possible to electrically program the memory regulator. A constraint irregular confirmation environment that is inclusion operated and erasable ROM are intended for the planned storage regulator. The code is written in Verilog HDL, and Verilog is also used to complete the confirmation. The reproduction is completed, and inclusion results are obtained, using the QuestaSim 10.0b programming tool. Our method's confirmation climate achieves a utilitarian inclusion of 96.8%, a declaration success of 100%, and zero percent statement disappointments. Reproduction results show that the intended regulator performed well and complied with all framework requirements.

INTRODUCTION

The term "Memory Controller" or simply "Memory Controlling Device" refers to a circuitry that can control the flow of information from the PC's primary storage to the processor and back to it [1]. It might be built into a different chip called an IMC (Integrated Memory Controller). Storage regulators allow microchips to communicate with memory centres. Storage regulators' primary goal is to provide the best possible point of contact and convention between hosts and memories so that information may be handled productively, enhancing move speed, information reliability, and data maintenance. The memory centres and memory regulators can be advanced in order to work on this relationship as a solution to the memory bottleneck. To further expand the regulator structures and planning computations is the most well-known previous memory regulators-oriented arrangement.

Section of the thought overdue the arrangement is to dump less-level memory administration on or after processor as a host, opening up assets [1]. The devised memory regulator streamlines the highlights of the current memory regulators on the work and coordinates them in single conventional widespread memory regulator. This nonexclusive widespread memory regulator can supplant the current memory regulators because of its strong determinations that can be summed up in these focuses: (i) Two overwhelmed sorts of memory: DRAM and FLASH are upheld. (ii) Good control of power. (iii) Most of the major strong and significant highlights for any current memory regulator are upheld and intended to be alternatively empowered or incapacitated by the producer want.

The DDR controller is an extension of old traditional memory controller synchronous DRAM (SDRAM). The DDR controller is capable of transacting the data on both rising and falling edges of the clock cycles. Thereby, double the transfer of data rate in the memory device. The cost of DDR memory is low, because it is most commonly used in PC's, storage and buffers. The DDR SDRAM controller's capability of data widths of 16, 32 and 64 bits and its throughput increases by using pipelining command and bank management techniques. Changes in pipelining command and bank management techniques enhance DDRSDRAM memory module transfer data twice as speed as SDRAM memory.

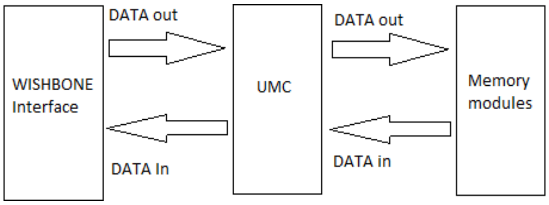
RELATED WORKS

The DDR controller is an extension of old traditional memory controller synchronous DRAM (SDRAM). The DDR controller is capable of transacting the data on both rising and falling edges of the clock cycles. Thereby, double the transfer of data rate in the memory device.

Compared to SDR SDRAM, DDR SDRAM have the burst length of 2, 4and 8,thusone address is required to access sequential address. There are two types of memories that can be purchased, namely Single In-line Memory Modules (SIMMs) and Double In-line Memory Modules (DIMMs). When in SIMM or DIMM both are small circuit that is capable to fit into common standard memory socket available in PC's. In DIMMs has two leads compared to the SIMMs has only one lead. They are soldered to the motherboard to supply desired memory. SDRAM memory usually comes in DIMMs. DIMMs are classified based on the data transfer rate, such has one is Single Data Rate (SDR) and other called Double Data Rate (DDR).SDR SDRAM transaction of data only on the rising edge of the clock cycle whereas DDRSDRAM transaction of data on both the rising as well as on the falling edge of the clock cycles, At present, other faster versions of available DDR SDRAMs, areDDR2,DDR3 andDDR4.

METHODOLOGY

In the proposed system, the general UMC module is explained, along with its interfaces with WISHBONE modules, memory controllers, and module blocks. We may check the UMC's functionality by looking at read and write data matching in the wishbone block. The UMC has successfully been verified if both varieties of data matching are available.



SDRAM, SSRAM, and SYNC memory types are all contained in memory modules. The UMC and these memories are connected through a memory interface. The top file of the system Verilog module contains all of these memory designs that have been expressed in Verilog code. Verilog top modules are utilized during the verification system.

With cutting-edge feature support, UMC has its unique architecture design. WISHBONE and the memory module are separated by UMC. WISHBONE interface and memory interface connect these blocks to the UMC. Among both WISHBONE as well as the memory module is Universal Memory Controller. Wishbone is a portable interface that is employed in semiconductor IP cores with the goal of accelerating system reusability as well as resolving architectural integration issues.

Wishbone module

An open-source hardware computer bus called the Wishbone Bus was created to facilitate communication between integrated circuit components. The objective is to enable connections between various cores inside a chip. A lot of ideas in the Open Cores project make use of the Wishbone Bus.

Wishbone serves as a logic bus by design. The bus topology and electrical data are not specified. Instead, signals, clock cycles, and high and low levels are used to describe the requirements. Wishbone is intended as a "logic bus". It does not specify electrical information or the bus topology. Instead, the specification is written in terms of "signals", clock cycles, and high and low levels.

Universal Memory Controller (UMC):

A digital circuit called the memory controller controls the data flow to and from the main memory. It is more commonly mentioned to as Combined Memory Organizer when a memory controller is integrated into another chip, such as being placed on the same die or as a core component of a microprocessor, rather than when it is present as a separate chip (IMC).

The Memory Controller becomes prominent due to following features

- Eight selector chips, each of which can be customized.
- Transferring and terminating bursts.
- RMW (Read Modify Write) cycles are supported.
- Dynamic bus scaling for asynchronous device reading.
- Default boot categorization sustenance.

- Provisions power miserable manner.
- Up to 8*64MB memory extent.

The UMC is situated between memory modules and wishbones. By wishbone interface, memory interface, and the UMC, these blocks are connected.

Memory Modules:
A printed circuit board with memory integrated circuits installed on it is called a memory module. The universal memory controller has been validated to work with a variety of memory, those are:
1) Flash Memory
2) Synchronous Dynamic Random Access Memory
Flash Memory: This section provides a general overview of the several types of flash memory, beginning with NOR flash memory, NAND flash memory, also the uses for each of them.

NOR Flash Memory: NOR Midway through the 1980s, flash memory was developed, and by the conclusion of the decade, it was launched to replace EPROMs. The first-generation products actually resembled erasable EPROMs due to their need for an external 12 V supply for programming and erasing, the fact that they could only perform bulk erases (erasing all memory content at once), and the fact that an external machine was needed to control the laborious erase process.

NAND Flash Memory: The memory cell structure of NAND Flash is essentially the same as that of NOR, but it has a completely different array organization and a distinct programming technique (Fig. 2.3). The memory array is set up using a NAND design, which involves connecting a number of cells in series between the bit line contact and ground. With NOR, which instead requires a ground line and a bit line contact every two cells, the density can be increased, but the speed is significantly reduced. This effectively prevents the use of this technology for random access memories and limits it to serial memories only because every cell must be read through a number of other cells, greatly reducing read current.

SDRAM Controller Features:

You can choose from a range of SDRAM devices to organize the SDRAM memory. The same type of SDRAM must be used for all connections to the same chip pick. The different SDRAM devices that are supported are shown in the table below.

SSRAM Memory:
SSRAMs can be set up as devices with 8, 16, or 32 bits. There is no distinction made by the memory controller between the various SSRAM groups. Always set the BW field in the CSC registry to 32. On SSRAM accesses, the Memory controller won't do dynamic bus scaling.

Memory Organization:
The various memory organizations that the Memory Controller core supports are discussed in this section.

SDRAM Memory Organization
You could choose from range of SDRAM devices to organize the SDRAM memory. The same type of SDRAMS must be utilized for all connections to the same chip select.
For storing data that is utilized to control electronics projects, memory is a major element of microcontrollers. Individually, memory has been broken down into a number of segments, each of which has a specialized kind of entry that helps to store information. RAM memory as well as ROM memory are two various forms of memories that are quite often both available as well as utilized in predictable manner.

MEMORY TIMING CONFIGURATION

Each chip select's Timing Select Register (TMSn) controls the memory timing. This register's significance varies depending on the memory type.

SDRAM Timing Configuration
The SDRAM can be used in one of two modes: One of two options is to keep the current column open, and the other is to close it after the Read or Write operation. It will depend on the overall architecture whether mode may be appropriate for a particular application as both offer benefits and drawbacks. Keep a dispute open is the first strategy. In this scenario, accesses to the same row after the initial Active command don't require expensive activation cycles. The drawback is that pre-charge cycles must be executed more than once when accessing a different row. This mode will offer improved overall performance in linear access scenarios when it is

expected that the same row will be touched more than once. Such accesses are shown in the following figure.

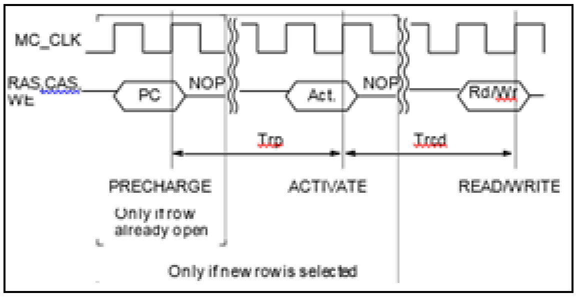


Fig.1 Keep Row Open Initial SDRAM Access

The second approach is to quickly close the row following a Read or Write action. Here, before performing the Read or Write operation, each new access must first run the Activate command.

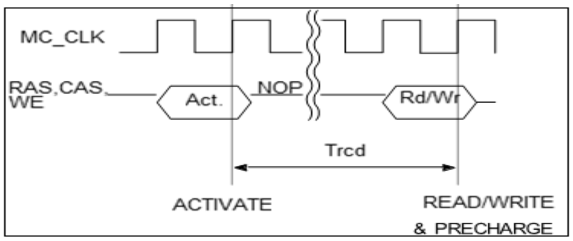


Fig.2 Close Row SDRAM Access

Single word and burst transfers are supported in both of the aforementioned modes. Categories based on how well the TMSn register is designed, a 1, 4, nor full page burst may be utilized. A command called Burst Terminate (BT) is sent when a transfer is too tiny to fit within the programmed burst size. Figure below demonstrates Burst read/write transfers.

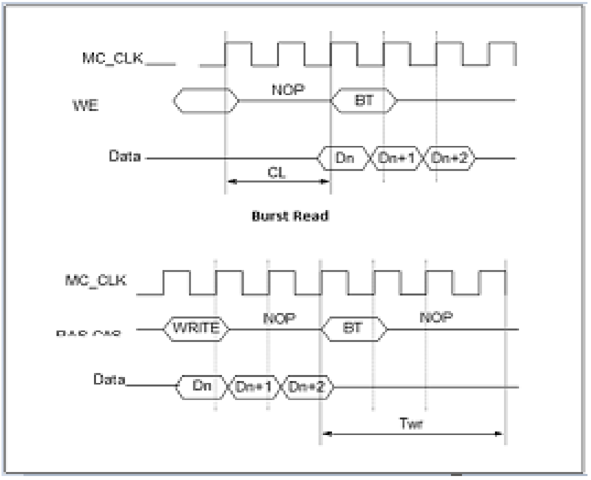


Fig.3 Burst Transfers

The memory controller would also provide auto synchronization cycles. The periodicity of the Automatic Refreshing counter is felt compelled by the Trf variable in the TMS register.

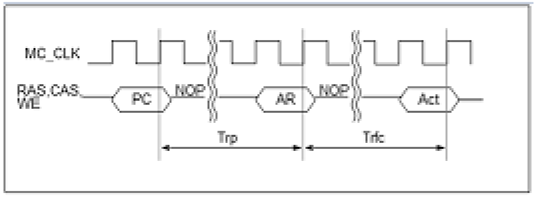


Fig.4 Auto Refresh

VERIFICATION ENVIRONMENT FORMEMORY CONTROLLER

The UVM Testbench normally creates instances of the UVM Test class as well as the Design under Test module as well as sets up the connections among respectively. The UVM Testbench can be built once as well as utilised to execute a variety of assessment tools also because UVM Test is dynamically constructed at run-time.

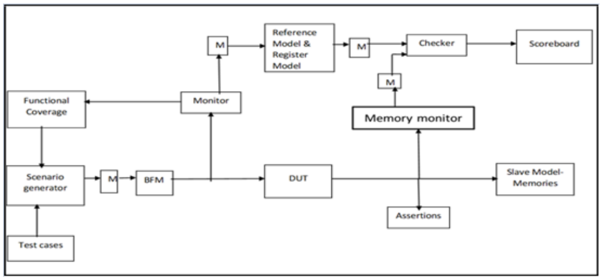


Fig.5 Authentication Environment for Memory Controller

Consequence generator: The device is stimulated by the scenario generator in ways that are intended to train engaging behaviors. Blocks communicate with one another using mailboxes.

Driver (BFM): The driver transforms pin squirms on a bus otherwise pin-level crossing point starting transaction-level inducements.

Monitor: The driver's companion is the monitor. In order to create transactions, it perceives pointers on a bus otherwise pin-level crossing point.

Slave Model Memories: SRAM, SDRAM, FLASH, and ROM are examples of peripheral memory devices.

Assertions: This directory contains the binding files for assertion inspections.

Functional Coverage: It keeps track of which functionalities have already been performed whenever monitoring behavior on monitors.

Scoreboard: All of this includes a reference model as well as equates how the Design under Test performs to the reference [10]. A "scoreboard breakdown" is any disagreement amongst the Design under Test as well as the benchmark. Fault of the scoreboard messes a problem that needs to be considered at, with either Design under Test or even with the scoreboard.

RESULTS AND DISCUSSION

The Fig. 8 shows the waveforms that were recorded after each memory module connected to the chip underwent composition-perused activity. All of the chip selections in the aforementioned test cases were related to the same kind of memories. Figure 8 shows the compose and browse activities of a memory following the completion of the compose cycles once all chip selections have been linked to various types of memories. Additionally, it can be inferred from the figures below that there has been no information debasement because the information that was created (mc data o) and the information that was read (mc data i) from a specific location are identical.

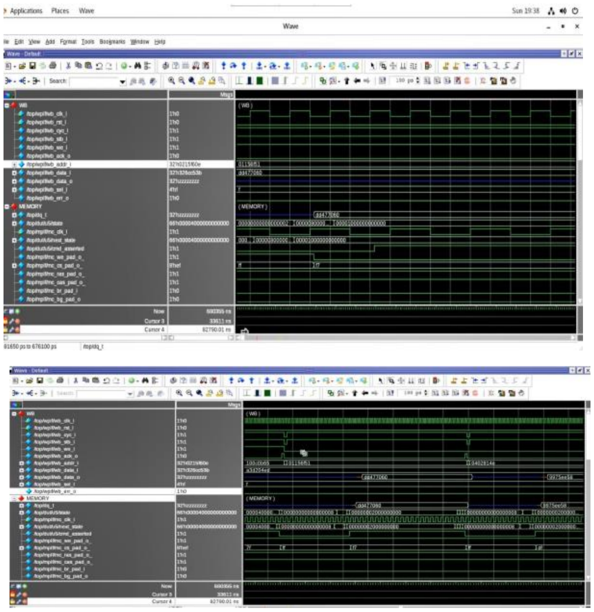


Fig. Simulation Results

CONCLUSION

SRAM, SDRAM, FLASH, and ROM are just a few examples of the synchronous and asynchronous memory devices that can work with the memory interface architecture. In order to test a memory controller's capabilities, a coverage driven constraint random-based model environment that is re-scalable, re-configurable, and reusable is provided. Questa Sim 10.0b was the simulation tool used in this experiment. It has been proven that this way of memory controller verification is speedier than the drawn-out directed test methodology.

REFERENCE

[1] Ashutosh Kumar Singh, Digitech Solution, Indore, India, Anurag Shrivastava, “A system Verilog approach for verification of memory controller”. Department of Electronics & Communication, Priyatam Institute of Technology and Management, Indore, India, 2011.

[2] Y. Savaria, Department of Electrical and Computer Engineering, Concordia University, Verification of universal Memory controller based on UVM, Department of Electrical and Computer Engineering, Concordia University, Montréal, Canada, Montreal, Canada. 2005.

[3] M. Bertola, Groupe de recherche en microélectronique, École Polytechnique de Montréal, A low area direct memory access controller architecture for a RIS-V based low power Microcontroller , QUE, Canada, G. Bois, Groupe de recherche en microélectronique, École Polytechnique de Montréal, Montreal, QUE, Canada, 2003.

[4] B. R Shivakumar, ,LDeeksha, Design of Reconfigurable Cache Memory using Verilog HDL NMAM Institute of Technolgy, Karkala, India, , India, 2019.

[5] CN, Yang Ben, Hu YueLi, Verification of Universal Memory Controller, Shanghai University, Shanghai, CN, 2011.

[6] M.D. Nguyen, M. Thalmaier, M.Wedler, D. Stoffel, and W. Kunz. A system Verilog approach for verification of memory controller. In Proc. Forum on Specification & Design Langliages(FDLJ. Sophia Antipolis. France. September 2009.

[7] Minh D. Nguyen, Max Thalmaier, Markus Wedler, Jorg Bormann, Dominik Stoffel. and Wolfgang Kunz.Controller Architecture for Memory BIST Algoritrhms. IEEE Transactions on Computer-Aided Design, 27(11):2068- 2082, November 2008.

[8] Chris spear, system Verilog for verification, New York: springer, 2006. Rath A.W, Esen.V and Ecker.W ,"A novel NAND Flash memory controller compatible with asynchronous and source synchronous data types" Publication Year: 2014 Page(s): 806 811.

[9] Mulani, Level Verification Using System Verilog Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on 16-18 Dec.2009 pp.378- 380.

[10] Pockrandt, M. Herber, P and Glesner, S, Verification of memory controller Using Deterministic Test Methodolgy, 2011 9th IEEE Symposium on 13-14 Oct.2011 pp.66

[11] 11.2nd Generation Intel® Core™ Processor Family Desktop, Intel® Pentium® Processor Family Desktop, and Intel® Celeron® Processor Family Desktop Datasheet, Volume 1, June 2013.

[12] Rudolf Usselmann, "Scalable approach for flash storage", Open Cores Revision.1.7 January 21, 2019.