Semiconductor Optoelectronics
ISSN: 1001-5868

MEMORY CONTROLLER WITH AN AHB PROTOCOL

Ms. Anupama Shetter [1], Dr. L Basavaraj[2] Dr. Prakash Kuravatti [3], Mr. Chandra Shekar P[4] [4]Ms. Swapna H[5]

[1-4]Department of ECE, ATME College of, Engineering
Mysore, India,
[5]Department of EEE, ATME College of, Engineering
Mysore, India

Abstract:
The multiprocessing-based Soc architecture is commonly used in the latest electronic devices such as smartphones, tablets, and smart wristwatches with large memory size. The major bottleneck of the computer system to improve the overall performance, is the memory performance. A circuit can manage the flow of data from the computer's main memory to the processor and back to it, is a Memory Controller. An effective control of the data between processor and memory can be done using memory controller. This can be integrated into another chip or as an integral part of a microprocessor which is called Integrated Memory controller (IMC). The design of memory controller is getting optimizes day by day. The memory controller architecture supports SRAM, SDRAM, FLASH, ROM and any synchronous or asynchronous memory devices. Implementation of the design through the synthesis and physical design flow which form a part of ASIC Design Flow. A memory controller with an AHB (Advanced High-Performance Bus) protocol is a critical component of modern System-on-Chip (SoC) designs. It provides efficient and high-speed communication between the processor and the memory devices, enabling the processor to read from and write to the memory devices at a faster speed while maintaining the reliability and consistency of the data transfers. Verification plays a role in any design flow as it is completed before silicon development. We used Verilog HDL for RTL and test bench purpose. Verified the all functionalities of the design. Synthesized the design, in which it converted into gate level which is called as netlist. Generated the layout by using this net list and verified its function.

Index Terms— AHB, SRAM, MAR, SDRAM, FLASH, ROM, Verilog HDL, RTL

1. INTRODUCTION

The memory controller is a digital circuit that manages the flow of data going to and from the computer's main memory. A memory controller can be a separate chip or integrated into another chip, such as being placed on the same die or as an integral part of a microprocessor, it is usually called an integrated memory controller (IMC).It is an important part of a computer system that controls the memory and is responsible for data exchange between memory and the CPU

1.1 Overview

The use of RF (radio frequency) technology in ambulance rescue systems can significantly improve the speed and efficiency of emergency medical services. The system works by utilizing RF signals to enable communication between ambulances, hospitals, and emergency response teams. The sensors used the car determine whether the accident is occurred are not. As soon any SMS is received to the police station and hospital the rescue process will be initiated. The ambulance uses the RF technology to indicate the traffic signal that the ambulance is coming on the wave, the range of the RF module use is around 100m. The concept is to green the traffic signal in the path of ambulance automatically so that the ambulance can reach the spot in the shortest time and to reach the hospital in time to save human life.

Memory Address Register (MAR) is the address register which is used to store the address of the memory location where the operation is being performed. Memory Data Register (MDR) is the data register which is used to store the data on which the operation is being performed. Memory read operation transfers the desired word to address lines and activates the read control line. Memory writes operation transfers the address of the desired word to the address lines, transfers the data bits to be stored in memory to the data input lines. Then it activates the write control line.

Advanced High-Performance Bus (AHB) is an on-chip communication bus protocol developed by ARM. It is a part of the Advanced Microcontroller Bus Architecture specifications. AMBA AHB-Lite addresses the requirements of high-performance synthesizable designs. It is a bus interface that supports a single bus manager and provides high-bandwidth operation. AHB-Lite implements the features required for high-performance, high clock frequency systems including burst transfers, single-clock edge operation, non-tristate implementation and wide data bus configurations.

A memory controller with an AHB (Advanced High-Performance Bus) protocol is a vital component in modern System-on-Chip (SoC) designs. It serves as an interface between the processor and memory devices, facilitating efficient and high-speed data transfers. The AHB protocol, known for its high performance and versatility, enables seamless communication and reliable access to various memory types, such as SRAM, SDRAM, and Flash. The memory controller with an AHB protocol plays a crucial role in managing the data flow between the processor and memory devices, ensuring optimized performance, low latency, and consistent data integrity. By leveraging the capabilities of the AHB protocol, the memory controller enhances the overall system efficiency and enables smooth execution of memory operations.

This protocol offers a range of features that simplify the design process and ensure effective memory access. It includes mechanisms for address decoding, arbitration, and timing control, enabling efficient coordination between the processor and memory devices. The scalable nature of the AHB protocol allows for its implementation in various applications, making it a popular choice among designers and developers.

The memory controller with an AHB protocol finds widespread application in diverse fields, including embedded systems, networking, multimedia, and automotive systems. In embedded

systems, it facilitates seamless storage and retrieval of data. In networking systems, the memory controller manages data transfers between the network processor and memory devices. Similarly, in multimedia and automotive systems, it ensures efficient communication between the processor and memory devices for tasks like multimedia processing and automotive control units.

A memory controller with an AHB protocol is a vital component in SoC designs. It enables high-performance data transfers, efficient memory access, scalability, and simplified design. With its broad application scope, the memory controller with an AHB protocol serves as a critical interface to facilitate seamless communication between processors and memory devices, ensuring optimal system performance and reliability.

1.2 Existing System

Memory controllers connect different types of memory to the processor bus. On power-up a memory controller is configured in hardware to allow certain memory devices to be active. These memory devices allow the initialization code to be executed. Some memory devices must be set up by software; for example, when using SRAM, you first have to set up the memory timings and refresh rate before it can be accessed.

The AHB (Advanced High-Performance Bus) protocol is a point-point specification, not a bus specification. Therefore, it describes only the signals and timing between interfaces. The previous diagram shows that each AHB manager interface is connected to a single AHB subordinate interface. AHB bus is a high-performance, point-to-point, manager-slave parallel bus used to connect on-chip peripheral circuits (or IP blocks) to processor cores.

Memory controllers with an AHB protocol are widely used in the industry, and many semiconductor companies offer such controllers as part of their system-on-chip (SoC) solutions. These memory controllers are often designed to meet the specific requirements of various applications and memory types. The specific features and capabilities of existing memory controllers with an AHB protocol may vary depending on the manufacturer and the target application. Some companies provide memory controllers that support multiple memory types, including SRAM, SDRAM, and Flash, while others may focus on specific memory technologies.

These memory controllers typically incorporate the necessary functionality to handle address decoding, arbitration, and timing control to ensure efficient memory access and data transfers. They are designed to comply with the AHB protocol specifications, ensuring compatibility and interoperability with other components in the system.

In addition to the basic functionalities, memory controllers with an AHB protocol may also include advanced features such as error correction codes (ECC), power management capabilities, multi-port support, and bus bridges for connecting to other bus protocols like AXI (Advanced eXtensible Interface).

2. LITERATURE REVIEW

Reddy et al. [1] deals with implementation of interface for efficient SOC. It accepts the Read

/ write commands from AXI and convert into DDR3 access protocol is an open standard on chip interconnect specification for the connection & management of functional blocks in SOC. In this project, an interface between a master (processor/user) & slave (DDR3 memory) was designed. This interface will transfer the data from master to slave & vice versa. This interface implemented is called as DDR3 Controller, which is specially targeted for the DDR3 memory Apart from the designing DDR3 controller, The communication is achieved between the master & the slave using various read / write commands of AXI protocol. Certain novel features of AXI protocol like variable length burst (from 1 to 16 data transfer per burst) & fixed wrapping burst are included in the design to achieve the data transfer. Our design has been implemented with respect to latency reduction and improvement in various performance parameters and the design is simulated Modelsim Synthesized on Xilinx successfully

Krishna D. Shalini et al. [2] "AXI Bus Compatible DDR3 Memory Controller for SoC". This paper discusses the overall architecture of AMBA AXI design core along with its advantage with DDR3 memory controller and operation of its individual sub blocks. It takes care of the DDR3 initialization and various timing requirements of the DDR3 memory. The memory controller works as an intelligent bridge between the AXI host and DDR3 memory. Our design has been implemented with respect to latency reduction and improvement in various performance parameters and the design is simulated on Modelsim and synthesized on Xilinx successfully. Presently, Communication between processors and memories is often a major bottleneck, making the design of the memory controller a critical task in determining overall system- level performance. The AXI compliant DDR3 Controller permits access of DDR3 memory through AXI Bus interface. It describes the controller and the data capture technique for high-performance DDR3 interfaces. The memory controller, in addition to collecting the requests from the masters and forwarding them to the memory, is tasked with re-ordering these requests in order to optimize specific system characteristics, such as latencies, power consumption and throughput.

Naomi et al. [3] "High Speed Data Transactions for Memory Controller based on AXI4 interface protocol SoC". In 2021 Recently, the latency resulted from the communication between the multi-core processor IP cores and the other such memory controller and peripherals on the same die SoC becomes a critical issue especially when all the core processors using advanced interface protocol such AXI4 and makes write and read transactions in different burst mode capability. The multi-processor cores in SoC which have high burst data transactions can play a critical role while accessing the shared resources such as the off-chip memory. These processor cores can starve other processor cores that have fewer burst data transactions while accessing the same shared resources. The starving issue of other processor cores leads to degrade the entire system performance of the SoC. However, the arbiter architecture in the SoC design plays the best solution to manage different processor core requests and granting one of them to access the shared resources according to different scheduling algorithms. In this paper, we have designed AXI interconnect, which includes arbiter architecture to connect four processor cores represented by the AXI masters and the off-chip memory represented by the salve.

The proposed model of this work improved the existing work by 34.4% as shown in the

simulation results of figures and 72.5%, and 104.8% as shown in the simulation results of Table1 and 2 respectively. The SoC design architecture is modeled in System Verilog HDL; simulation and synthesis are done by using the Vivado tool and FPGA ZYNQ-7 ZC702 Evaluation Board (xc7z020clg484-1).

3. MOTIVATION

The motivation behind developing a memory controller with an AHB (Advanced High-Performance Bus) protocol lies in the need for efficient and high-speed communication between the processor and memory devices in System-on-Chip (SoC) designs. The AHB protocol addresses the challenges associated with data transfers, reliability, and performance in complex embedded systems. Here are some key motivations for using a memory controller with an AHB protocol:

1. High-Performance Data Transfers: The AHB protocol is designed to provide high-performance data transfers between the processor and memory devices. It offers optimized data transfer rates, reducing latency and improving overall system performance.

2. Efficient Memory Access: The AHB protocol supports efficient memory access by allowing multiple masters to access the memory simultaneously. This parallel access capability reduces contention and improves overall memory utilization, resulting in faster and more efficient memory operations.

3. Standardization and Interoperability: The AHB protocol is an industry-standard bus protocol widely adopted in SoC designs. Its standardized nature ensures compatibility and interoperability between different components from various vendors. This simplifies system integration and promotes reuse of intellectual property (IP) cores.

4. Scalability and Flexibility: The AHB protocol is designed to be scalable and adaptable to different memory types, sizes, and system configurations. It accommodates a wide range of memory devices, including SRAM, SDRAM, Flash, and more, making it suitable for diverse applications and memory requirements.

5. Simplified Design and Development: Memory controllers with an AHB protocol offer features such as address decoding, arbitration, and timing control, which simplify the design and development process. These built-in functionalities reduce the effort required to implement a memory controller, thus accelerating the overall development cycle.

6. Enhanced System Reliability: The AHB protocol incorporates mechanisms for error detection and correction, ensuring the integrity and reliability of data transfers. Error correction codes (ECC) can be used to detect and correct errors, improving the system's overall robustness.

7. Performance Optimization: The AHB protocol provides features such as burst transfers, pipelining, and efficient bus protocols, enabling optimized performance for memory access. These features minimize the impact of memory latency and maximize data throughput, resulting in improved overall system performance.

To overcome the problem of the integrated memory controller inside the CPU is that it has poor memory adaptability and flexibility.

The motivation behind using a memory controller with an AHB protocol is to enable efficient, high-speed, and reliable communication between the processor and memory devices in SOC designs. The AHB protocol addresses the challenges of data transfer, memory access, standardization, scalability, and system performance, making it a popular choice in the industry.

4. PROBLEM STATEMENT

As per the survey stated above, this project consists of an implementation of a memory controller with an AHB interface. In this regard, the problem statement may be defined as "the design and implementation of an AHB based memory controller".

5. METHODOLOGY

This chapter consists of proposed block diagram, methodology of the design, Software requirement, advantages, disadvantages and applications.

5.1 Proposed system

This section consists of overview, AHB manager, AHB subordinate, SRAM controller, SRAM and result of the design.

5.1.1 Overview

The above figure 3.1 is the general block diagram of the memory controller with an AHB protocol. It consists of memory controller, memory processor, data signals, control signals and address signals

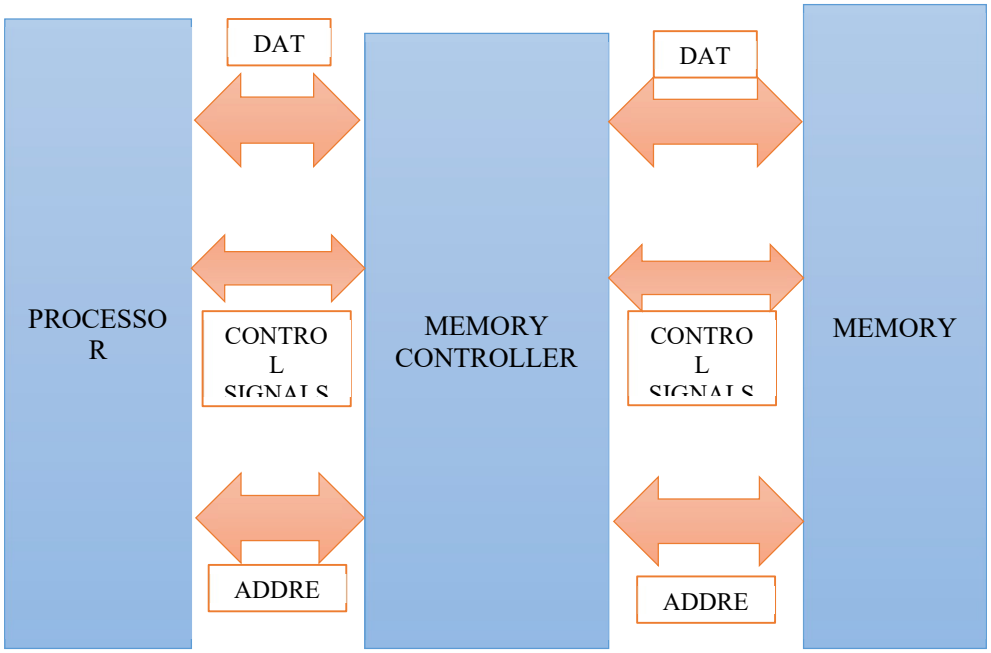

Figure 1: Memory controller with **AHB protocol.**

5.1.2 AHB manager

AHB manager is the part of the on-chip protocol of the AHB protocol. It converts the processor signals to memory signals. It is controlled by the processor and AHB subordinate. The address and control signals fed from the processor to AHB manager. The control, address and data signals fed to the subordinate. Some signals are fed from subordinate to check whether the subordinate is ready to transfer address, control and data signals.

The AHB manager consists of signals which have its own function. The signals can categorize as global signals, control signals, address signals, data signals and transfer response signal. The global signals are clock and reset. Clock times all bus transfers. All signal's timings are related to the rising edge of clock. Reset is the signal in which it clears the data present in the bus and system. This is the only active low signal.

5.1.3 AHB subordinate

An AHB-Lite slave responds to transfers initiated by managers in the system. The slave signals back to the manager the success, failure or waiting of the data transfer. The input signals are fed from the manager signals are HADDR, HWRITE, HBURST, HWDATA, HTRANS, HPROT AND HSIZE. The subordinate output signals are fed to manager and also SRAM controller. HRDATA is the data signal from the subordinate. During read operations, the read data bus transfers data from the selected slave.

The transfer response signals are HREADY AND HRESP. HREADY signal describes; When HIGH, the HREADY signal indicates that a transfer has finished on the bus. This signal can be driven LOW to extend a transfer. HRESP signal describes; When LOW, the HRESP signal indicates that the transfer status is OKAY. When HIGH, the HRESP signal indicates that the transfer status is ERROR.

5.1.4 SRAM controller

SRAM is static random-access memory that uses latching circuitry to store each bit. It will hold its data permanently in presence of power. SRAM controller is the device to control the operation of read and write operation of data in specific location to memory devices. It controls the data, address and control signals from processor and memory devices.

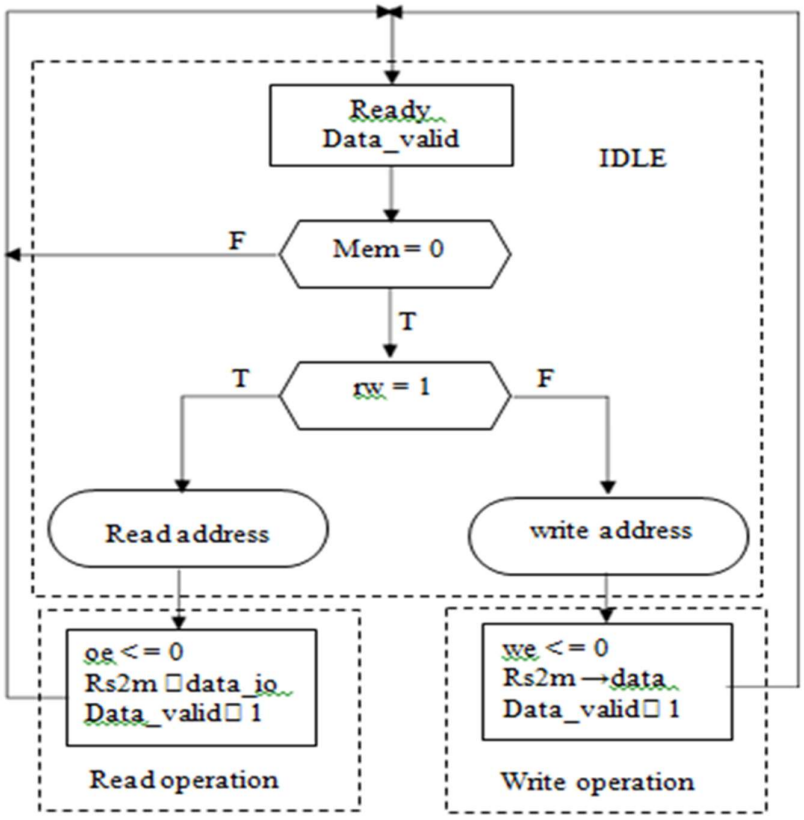

Figure 2: SRAM controller flow chart

5.1.5 SRAM

SRAM is static random-access memory that uses latching circuitry to store each bit. It will hold its data permanently in presence of power. From the above figure 3.6, the control signals are chip enable, output enable and write enable. Chip enable signal in which turns on the system or turns off. Output enable signal whether the output should be transferred Memory controller with an AHB protocol or not to transfer the data and address. Write enable decides to perform read or write operation of the system. The inputs are fed from the SRAM controller and outputs fed to SRAM controller.

5.1.6 Simulation of the design

The top module in which all sub blocks like AHB manager, AHB subordinate, SRAM controller and SRAM are integrated as shown in figure 3.7. AHB manager and AHB subordinate is the protocol. AHB manager is interfaced with input/output of the processor and AHB subordinate is interfaced with input/output to SRAM controller. Then SRAM controller signals are passed to SRAM to perform read or write operations. The read/write operations perform based on little endian principle.

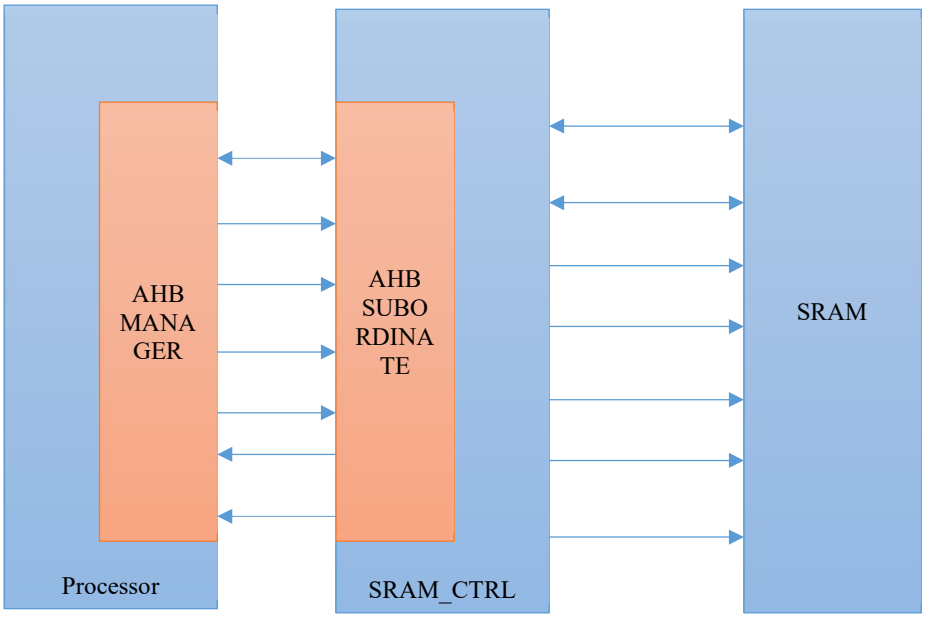

Figure 3: Connectivity of memory controller and AHB protocol.

6. RESULT & ANALYZING

After integrating the memory blocks and AHB protocol, it works on the basis of the three phases. The three phases are Idle, address and data phase.

Idle phase:

All the enable signals are enabled. In Idle transfer type indicates that no data transfer is required. A manager uses an IDLE transfer when it does not want to perform a data transfer. Hence all the signals are enabled and initially set to default values.

Address phase:

In this phase processor sends the address of specific location to read or write the data. In AHB manager address is fetched and it locates the data to be read or need to be written in the SRAM. Once AHB subordinate receives the address of any location, it provides the data present in the location or allows to write into the location. Until the address is not fetched by SRAM controller, then the read or write operation cannot be performed.

Data phase:

Data phase comprises of two operations, they are read and write operation. And also, it has size of the transfer to store or read the data like 8-bits (Byte), 16-bits (Half-word) and so on in each operation. When the system is set to default state i.e., idle state. Then reset operation is performed to clear the data in the system. When the data is cleared, the write operation is performed to load the data into memory after fetching address location. Then write response is set to high that indicates write operation is done. If the write response is low indicates the write operation is not done. The write operation is load into SRAM on the basis of size of the transfer.

After this read operation is performed to retrieve the data from the SRAM from the required address location. The size of the transfer is same for the read operation.

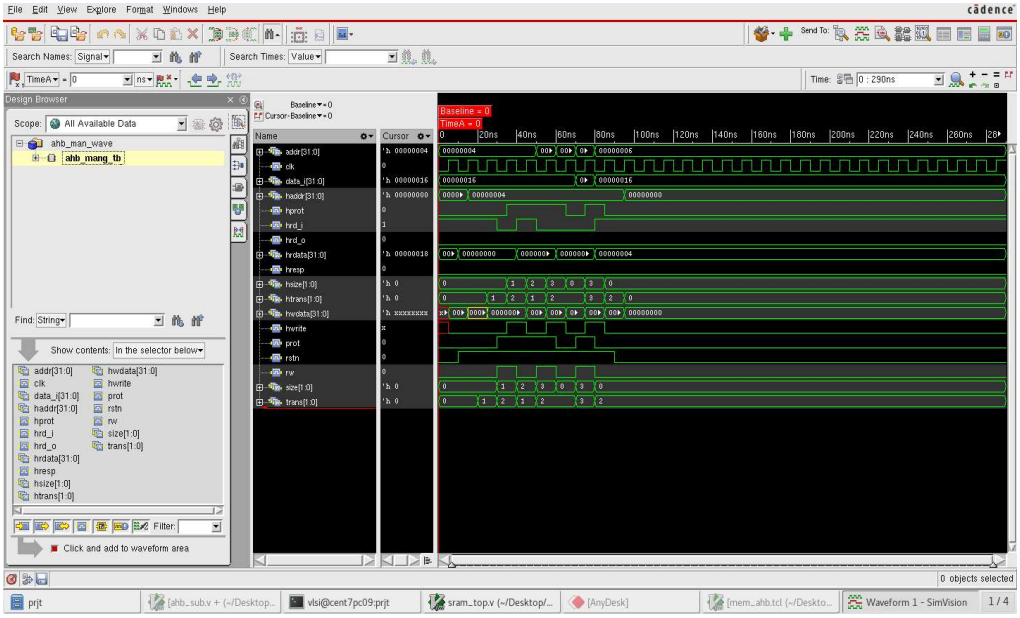

Figure 4: Test cases result of an AHB protocol

The above figure 4 is the test cases and simulated waveform of an AHB protocol. It comprises of the idle, address and data phases in which data comprises of the read and write operation. These operations are controlled by the processor. Controlled signals are passed to subordinate which is interfaced with the SRAM controller

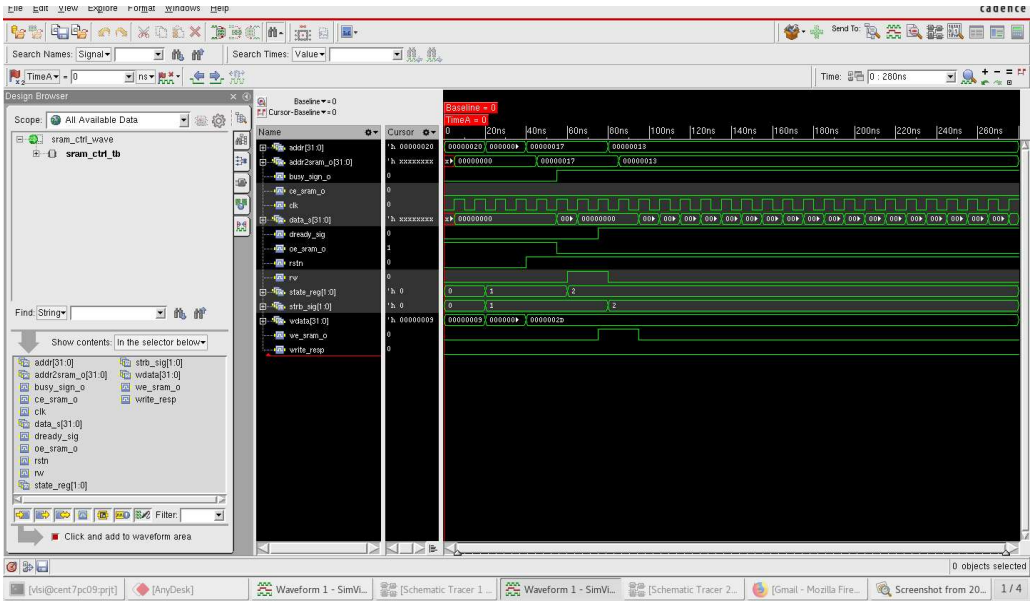

Figure 5: Simulated waveform of SRAM controller

From the above figure 5 is the test cases and simulated waveform of the SRAM controller. It

has three states in which idle, read and write operation. These states are tested and verified.

7. CONCLUSION

The Design work of this project was started by surveying existing memory controllers. The work is designed so that the controller works in three different phases namely idle phase, read and write phase. The memory controller block comprises SRAM controller and SRAM sub-blocks. SRAM is used to store data while SRAM controller is designed to manage the flow of data between the processor and the memory. The SRAM controller is designed and verified for three test cases. For communication with other on-chip blocks, the memory controller is interfaced with the AHB protocol. Before the interface AHB protocol was tested for its functionality. A single manager and subordinate were designed and verified. The memory controller with AHB protocol was then verified. This was done by carrying out the simulation. Post simulation the design's synthesis was carried out using Cadence Genus and Cadence gsclib45. As per the objective of the project the memory controller with an AHB interface has been designed and synthesized.

REFERENCES

[1] Reddy, Y. Nageswara, and A. Suman Kumar Reddy. "Implementation of Interface between AXI Protocol and DDR3 Memory for SoC".

[2] Krishna, D. Shalini, Syed Kareem Saheb, and M. V. S. Prasad. "AXI BUS COMPATIBLE DDR3 MEMORY CONTROLLER FOR SoC".

[3] Noami, A., Alahdal, A., Kumar, B. P., Chandrasekhar, P., & Safi, N. (2021, February). "High speed data transactions for memory controller based on AXI4 interface protocol SOC". In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-7). IEEE.

[4] Kante, S., Kakarla, H. K., & Yadlapati, A. (2016). "Design and Verification of AMBA AHB-Lite protocol using Verilog HDL". International Journal of Engineering and Technology, EISSN, (0975-4024), 734-741.

[5] Lambe, P. T., & Kulkarni, M. (2017). "Functional Verification of AMBA AHB LITE Interconnect using System Verilog". International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 6(7).

[6] Joe, N., & Brite, A. (2013). "A Novel Design Approach to an AMBA AHB Compliant Memory Controller".International Journal of Advances in Engineering & Technology, 6(5), 2053.