Analysis of RAM with High Speed Processor using VHDL

Mohit Ahirwar

Assistant Professor Braj bihari Soni ,NRI Institute information science & Technology, Bhopal, Madhya Pradesh, India

Abstract

As we know the leakage power is very important and these is increasing when we going from one technology to another technology generation. CMOS technology improved the level of performance which providing good characteristics such as low cost, low discharge, low power dissipation, high density, low leakage current in the circuits. DRAM used as a cache memory due to the volatile in nature. So that we need to refresh the circuitry and which providing the problem of the leakage current and power consumption. Which can be overcome by proceeding the implementation of DRAM with self-voltage level controllable technique (SVLC). This can be done by using DSCH and Microwind software. This technique help to reduce the leakage current at the lowest point.

Keywords - SCVL, DRAM, leakage current, power dissipation.

I. INTRODUCTION

A processor is the most important integrated circuitry (IC) in computer. It is used for interpreting most computer commands. It is an electrical component that performs operation on external data sources such as memory and data stream. Typically taking the form of microprocessors, it can be implemented on integrated circuit chips.

A processor, also known as a central processing unit (CPU), is the primary component of a computer that performs most of the processing inside the computer. It acts as the "brain" of the computer, executing instructions from computer programs.

The microprocessor, which evolved from the inventions of the transistor and the integrated circuit, is one of the icons of the present information age. The pervasiveness of the microprocessor in this age goes far beyond the wildest imagination at the time of the first microprocessor. From the fastest computers to the simplest toys, the microprocessor is found everywhere. The microprocessor today has over 1 billion transistors on some of the most powerful devices. A microprocessor is a tiny, enormously powerful high speed electronic brain etched on a single silicon semiconductor chip which contains the basic logic, storage and arithmetic functions of a computer. It thinks for the computer and, like a traffic cop, coordinates its operations. It receives and decodes instructions from input devices like keyboards, disks then sends them over a bus system consisting of microscopic etched conductive "wiring" to be processed by its arithmetic calculator and logic unit. The results are temporarily stored in memory cells and released in a timed sequence through the bus system to output devices such as CRT Screens, networks, or printers. In some cases, the terms 'CPU' and 'microprocessor' are used interchangeably to denote the same device.

VLSI PROCESSOR IN COMPUTER

Processor or Central Processing Unit (CPU) refers to logic circuitry that responds to and processes basic instructions that drive computers. This integrated electronic circuit performs calculations that run computers. These circuits are found in electronic devices. They receive input in the form of program instructions and execute calculations for providing with which the user will interact. For any operation on a computer, the processor must interpret the operating system. A processor consists of arithmetical logic and a control unit (CU) that measures capability in terms of: Maximum number of bits/instructions Relative clock speed. The ability to process instruction at a given time

II. COMPONENTS OF PROCESSOR

A processor has four components: a floating point unit (FPU), an arithmetic logic unit (ALU), registers, and cache memories.

1. Arithmetic Logic Unit (ALU)

ALU is the main component in a processor that performs various arithmetic and logic operations. It is an integrated circuit within the CPU/GPU, due to which it is also known as an integer unit (IU). This is the last component that performs calculations in the processor.

2. Floating-Point Unit (FPU)

It is part of the computer system used for carrying out operations on floating-point numbers. These operations include square root, multiplication, division, subtraction, and addition. It can perform transcendental functions such as trigonometric and exponential functions; however, it may not be accurate.

VLSI PROCESSORS WORK

Every processor is constituted of one or more individual processing units called 'cores'. Every core processes instructions from a single computing task at a particular speed which is also known as 'clock speed'. This speed is measured in gigahertz (GHz). In modern computers, there are multiple processor cores since increasing clock speed is a technically difficult task after a point. All these processor cores work together for processing instructions and completing multiple tasks at the same time. Within the processor, there are four key elements including arithmetic logic unit (ALU), floating point unit (FPU), registers and cache memories. ALU carries basic and advanced arithmetic and logic operations. These results are sent to registers that also store instructions. Cache are the small and fast memories that store data copies for frequent use. To process instructions, processors use an instruction pipeline.

This typically involves stages such as fetch (retrieving the instruction from memory), decode (determining what the instruction does), execute (performing the actual operation), and write-back (storing the result). To optimize execution, processors employ branch prediction techniques, attempting to guess the outcome of conditional instructions before they're fully processed. Processors interact with a larger memory hierarchy beyond their internal components.

This starts with multiple levels of cache (L1, L2, L3), then extends to main RAM, and even storage devices. Each level in this hierarchy offers a trade-off between speed and capacity, with faster but smaller memories closest to the processor. In modern computers, multiple processor cores work together to handle various tasks simultaneously. This is complemented by multi-threading capabilities, allowing each core to handle multiple threads of execution at once, further improving performance and efficiency.

III. PROCESSOR SYSTEM KEY ELEMENTS

A **processor system** refers to a collection of components designed to process data, execute instructions, and manage operations within a computer or embedded system. It typically consists of the following key elements:

1. **Processor (CPU)**: The central processing unit (CPU) is the core of the system. It executes instructions, performs arithmetic and logic operations, and controls the flow of data between components.

2. **Memory**:

○ **RAM (Random Access Memory)**: Provides temporary storage for data and instructions that the processor needs to access quickly during operation.

○ **ROM (Read-Only Memory)**: Stores firmware or essential instructions that do not change during regular use.

3. **Input/ Output (I/O) Interfaces**: Facilitates communication between the processor system and external devices, such as keyboards, displays, sensors, or storage devices.

4. **Storage**: Non-volatile storage like hard drives, SSDs, or flash memory, used for long-term data retention.

5. **Bus System**: A communication pathway for transferring data and signals between the processor, memory, and peripherals.

6. **Control Unit**: A part of the processor responsible for interpreting instructions and orchestrating the operation of the processor system.

7. **Clock and Timing System**: Ensures synchronization of operations by providing a consistent timing reference.

8. **Power Supply**: Supplies and regulates the power needed to operate the processor system.

IV. DYNAMIC RANDOM-ACCESS MEMORY (DYNAMIC RAM OR DRAM)

Dynamic random-access memory (**dynamic RAM** or **DRAM**) is a type of random-access semiconductor memory that stores each bit of data in a memory cell, usually consisting of a tiny capacitor and a transistor, both typically based on metal–oxide–semiconductor (MOS) technology. While most DRAM memory cell designs use a capacitor and transistor, some only use two transistors. In the designs where a capacitor is used, the capacitor can either be charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and 1. The electric charge on the capacitors gradually leaks away; without intervention the data on the capacitor would soon be lost. To prevent this, DRAM requires an external memory refresh circuit which periodically rewrites the data in the capacitors, restoring them to their original charge. This refresh process is the defining characteristic of dynamic random-access memory, in contrast to static random-access memory (SRAM) which does not require data to be refreshed. Unlike flash memory, DRAM is volatile memory (vs. non-volatile memory), since it loses its data quickly when power is removed. However, DRAM does exhibit limited data remanence.

DRAM typically takes the form of an integrated circuit chip, which can consist of dozens to billions of DRAM memory cells. DRAM chips are widely used in digital electronics where low-cost and high-capacity computer memory is required. One of the largest applications for DRAM is the *main memory* (colloquially called the RAM) in modern computers and graphics cards (where the main memory is called the *graphics memory*). It is also used in many portable devices and video game consoles. In contrast, SRAM, which is faster and more expensive than DRAM, is typically used where speed is of greater concern than cost and size, such as the cache memories in processors.

The need to refresh DRAM demands more complicated circuitry and timing than SRAM. This complexity is offset by the structural simplicity of DRAM memory cells: only one transistor and a capacitor are required per bit, compared to four or six transistors in SRAM. This allows DRAM to reach very high densities with a simultaneous reduction in cost per bit. Refreshing the data consumes power, causing a variety of techniques to be used to manage the overall power consumption. For this reason, DRAM usually needs to operate with a memory controller; the memory controller needs to know DRAM parameters, especially memory timings, to initialize DRAMs, which may be different depending on different DRAM manufacturers and part numbers.

Characteristics of DRAM:

- Volatile: Data is retained only as long as the memory module is powered; it is erased when power is lost.
- Dynamic: The capacitors holding data leak charge over time, requiring periodic refreshing (every few milliseconds) to maintain stored data.
- Cost-Effective: Cheaper to produce than other types of RAM, such as Static RAM (SRAM), but slower.
- High Density: Offers high storage capacity in a compact form factor.

Working Principle:

- Data Storage: Each bit of data is stored in a tiny capacitor.
- Charge/Discharge: A charged capacitor represents a binary 1, and a discharged capacitor represents a binary 0.

- Refreshing: Since capacitors lose charge over time, DRAM requires constant refreshing to replenish the charge and retain data.

Types of DRAM:

- SDRAM (Synchronous DRAM): Operates in sync with the system clock, improving efficiency and speed.
- DDR (Double Data Rate) SDRAM:
- DDR1, DDR2, DDR3, DDR4, and DDR5 are generations of DDR DRAM, with each offering improved speed, efficiency, and power consumption over its predecessor.
- Mobile DRAM (LPDDR): Designed for mobile devices with low power consumption and high efficiency.
- Graphics DRAM (GDDR): Used in GPUs for handling intensive graphics processing tasks.

Advantages of DRAM:

- High Capacity: Can store large amounts of data compared to SRAM.
- Low Cost: More economical to manufacture than SRAM.
- Widely Used: Essential in computers, laptops, smartphones, and other electronic devices.

Disadvantages of DRAM:

- Slower than SRAM: Requires refreshing, which adds latency compared to static memory.
- Volatile: Loses data when power is removed.
- Energy Intensive: Needs constant refreshing, consuming more power than non-volatile storage.

RESEARCH MOTIVATION

The traditional design of SRAM (Static RAM) using CMOS technology represents severe performance degradation due to its higher power dissipation and leakage current. Thus, aNano-scaled device named FinFET is introduced for designing SRAM since it has three dimensional (3D) design of the gate. FinFET has been used to improve the overall performance, which includes efficiency, power, and area. Moreover, FinFET has been chosen as a transistor of choice because it is not affected by SCEs. In this work, we have reviewed various FinFET based SRAM cells, performance metrics and the comparison over different technologies.

V. PROPOSED METHODOLOGY SDRAM

The DDR SDRAM Controller (referred to as the memory controller) is designed to be able to support a various number of AHB slave interfaces. To achieve this, the control path and data path has been separated. The core memory controller handles the control path and the data path is incorporated into each AHB interface of the memory controller. An internal arbitrator decides which AHB interface that has access to the data path and the core memory controller. For the initialization of the memory controller there is also an APB interface. Through the APB interface EMRS and MRS registers are set up as well as the size and organization of the DDR SDRAM being used. The memory controller can handle a various number of DDR SDRAM chips and the maximum number of supported chips are decided upon synthesis.

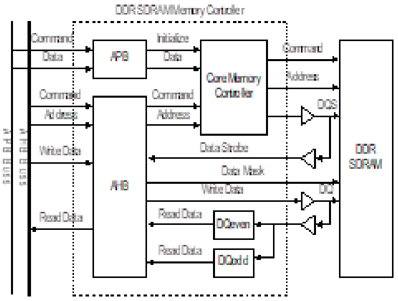


Fig.1 Schematic view of the memory controller with one AHB Interface

When an AHB interface has been granted access to the data bus and the core memory controller the AHB interface can tell the core memory controller to do either a read or write operation. The AHB interface also tells how long the burst is going to be. The core memory controller then handles the activation of rows and if necessary splits the burst into more than one command and issues them to the DDR SDRAM. The core memory controller then signals to the AHB interface when data has to be sampled from or presented on the data bus depending on if it is a read or write operation that have been requested. The core memory controller handles all the timings that are involved when a command can or has to be issued to the DDR SDRAM.

VI. RESULT AND SIMULATION

SOFTWARE USED

1. MICROWIND

Microwind is a tool for designing and simulating circuits at layout level. The tool features full editing facilities (copy, cut, past, duplicate, move), various views (MOS characteristics, 2D cross section, 3D process viewer), and an analog simulator.

Microwind is a CMOS circuit editor and simulation tool, for logic and layout-level design, running on Microsoft Windows. It has been developed since 1998 through several versions, and is available as a freeware (lite version [5]) for educational purpose, and sold to universities through Ni2designs, India. Microwind allows students to draw the masks of the circuit layout, use a set of simulation properties (clocks, pulses, DC voltage sources,) to build a test scenario, and validate the current/voltage relationships through built-in analog simulation.

The approach consists of a step-by-step illustration of the most important relationships between the layout and its performance. We concentrate on, static and dynamic characteristics and the MOS model parameters that are considered most significant for educational purposes.

The evolution of integrated circuit (IC) fabrication techniques is a unique fact in the history of modern industry. The improvements in terms of speed, density and cost have keep constant for more than 30 years. By the end of 2000, “System-on-Chips” with about 100,000,000 transistors will be fabricated on a single piece of silicon no larger than 2x2 cm. In this chapter, we present some information illustrating the technology scale down.

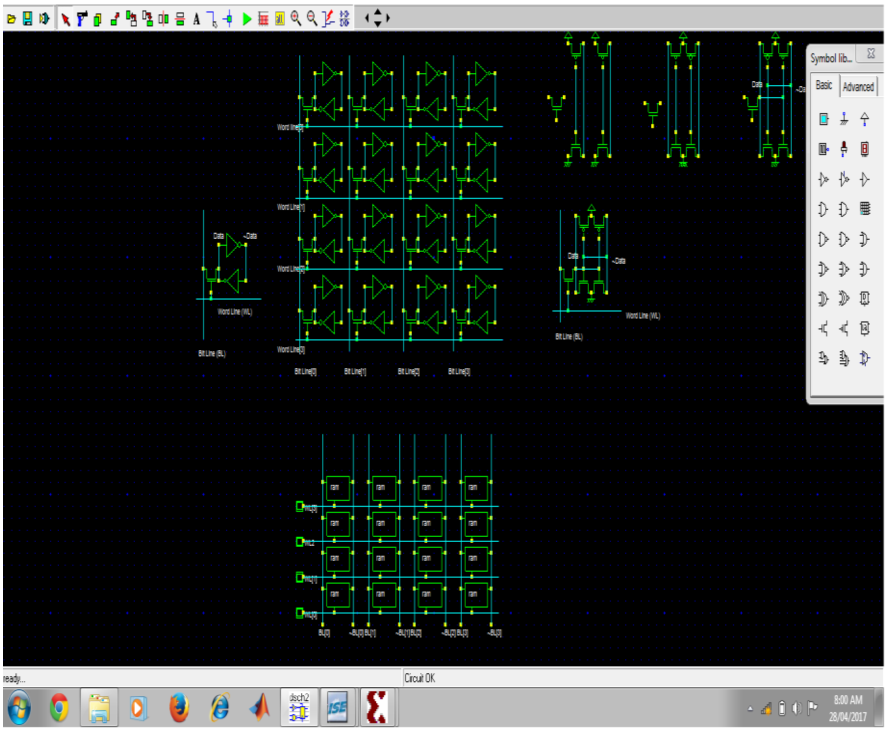


Fig.2 6Timing SDRAM design

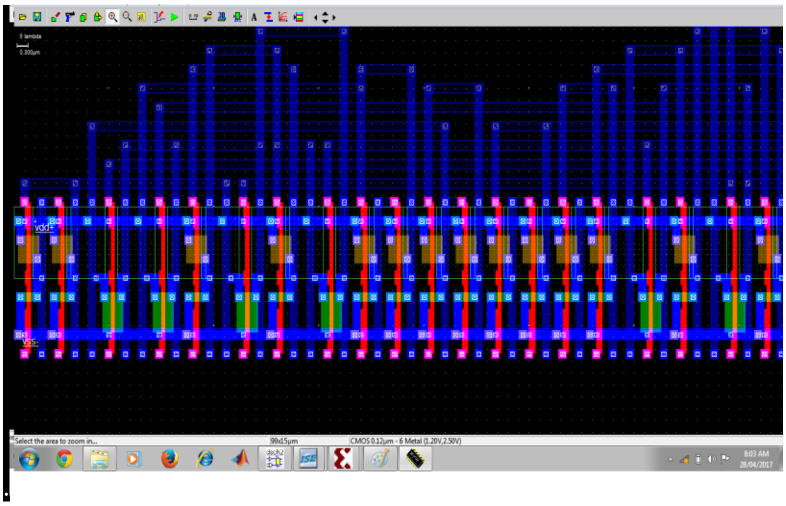

vdd
vss
Select the area to zoom in...
99x15μm
CMOS 0.12μm - 6 Metal (1.20V,2.50V)
8:03 AM
28/04/2017

Fig.3 6Timing SDRAM LAYOUT design.

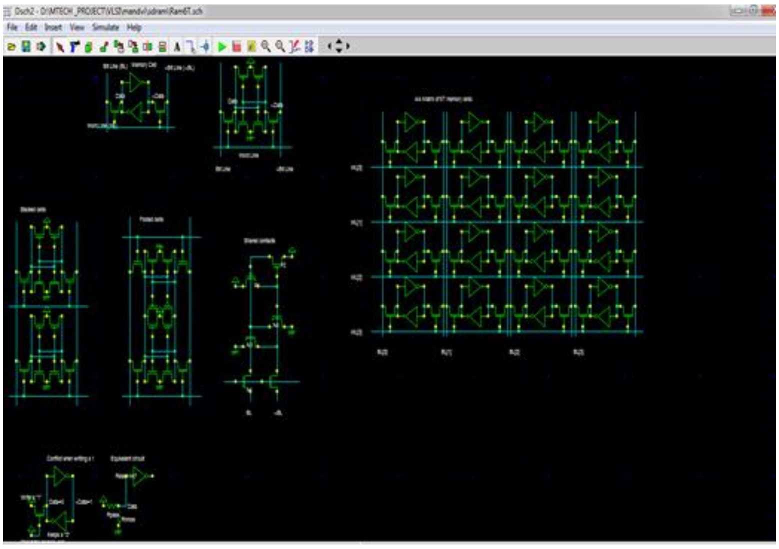

Dsch2 - D:\MTECH_PROJECT\VLSI\mandvi\sdram\Ram6T.sch
File Edit Insert View Simulate Help

Fig.4.Multi DDR SDRAM design.

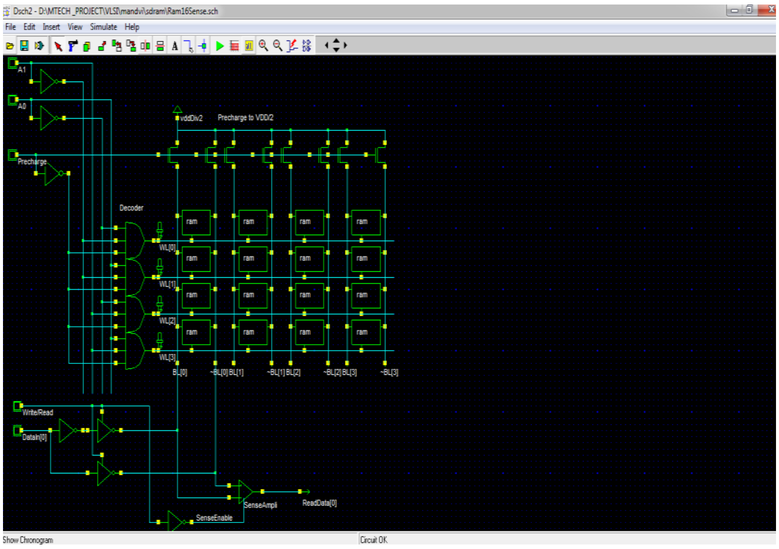


Fig.5. 16 Sense Schematic design.

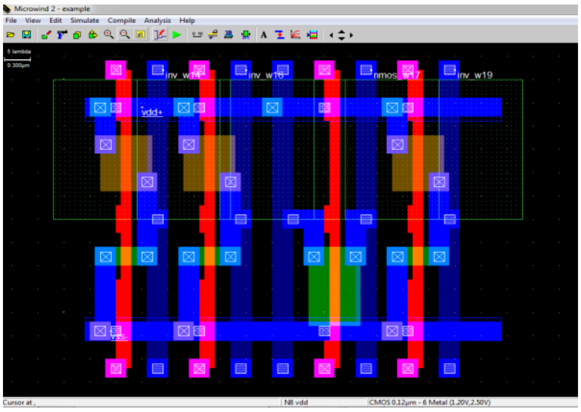

Microwind 2 - example
File View Edit Simulate Compile Analysis Help
inv_w14
inv_w16
nmos_w17
inv_w19
vdd+
Cursor at
N8 vdd
CMOS 0.12µm - 6 Metal (1.20V,2.50V)

Fig.6. Layout design 16 sense

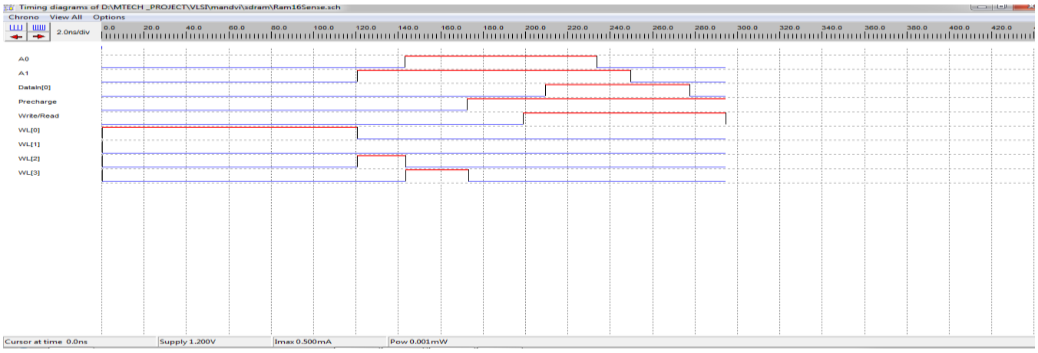


Fig.7.Timing waveform16 sense.

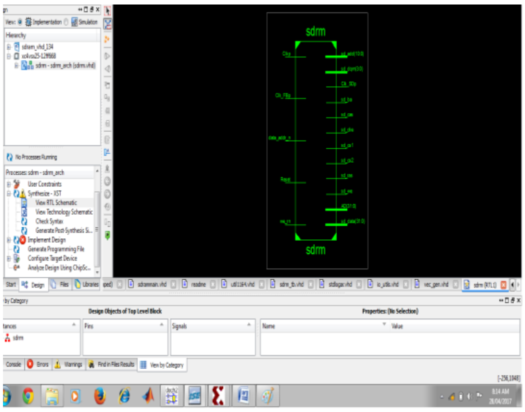


Fig.8. SDRAM Xillinx Design.

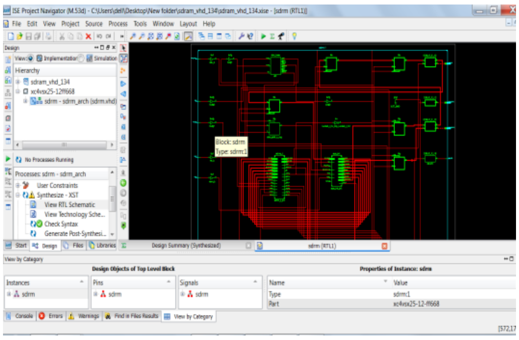

File Edit View Project Source Process Tools Window Layout Help
Hierarchy
No Processes Running
Synthesize - XST
View RTL Schematic
View Technology Schematic
Check Syntax
Block: sdrm
Type: sdrm1
Design Objects of Top Level Block
Properties of Instance: sdrm
Instances
Pins
Signals
Name
Value
sdrm
Type
sdrm1
Part

Fig.9. RTL logic design

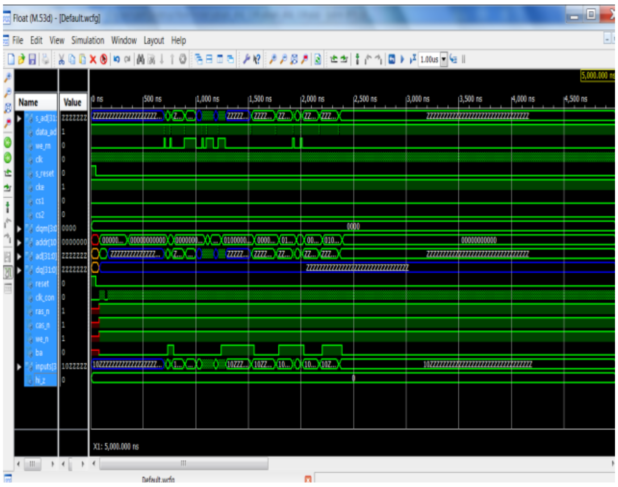
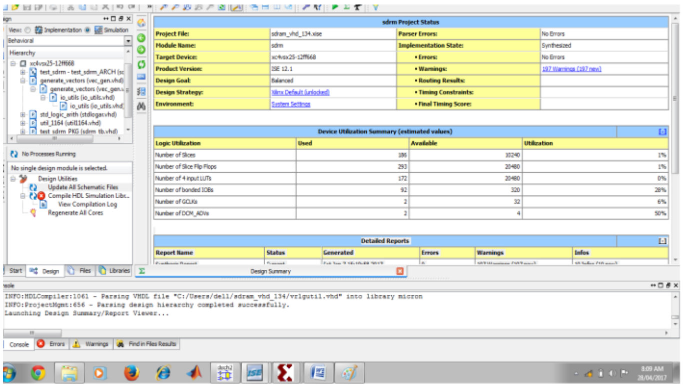


Fig.10. Timing Waveform.



sdrm Project Status			
Project File:	sdram_vhd_134.xise	Parser Errors:	No Errors
Module Name:	sdrm	Implementation State:	Synthesized
Target Device:	xc4vsx25-12ff668	• Errors:	No Errors
Product Version:	ISE 12.1	• Warnings:	197 Warnings (197 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	186	10240	1%
Number of Slice Flip Flops	293	20480	1%
Number of 4 input LUTs	172	20480	0%
Number of bonded IOBs	92	320	28%
Number of GCLKs	2	32	6%
Number of DCM_ADVs	2	4	50%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos

```
INFO:HDLCompiler:1061 - Parsing VHDL file "C:/Users/dell/sdram_vhd_134/vrlgutil.vhd" into library micron
INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
```

Fig.11. Enhancement report.

VII. CONCLUSION AND FUTURE SCOPE

CONCLUSION

In this work an efficient fully functional DDR SDRAM controller is designed. The controller generates different types of timing and control signals, which synchronises the timing and control the flow of operation. The memory system operates at double the frequency of processor, without affecting the performance. Thus we can reduce the data bus size. The drawback of this controller is complex schematic with large number of buffers in the circuit increases the amount of delay.

In the proposed work, Control Registers are designed according to SDRAM specification. From the results obtained, it is concluded that master successfully writes the timing parameter data into the control registers and also successfully reads the response from the CSR. When CSR register is updated with new value corresponding status register bit is successfully set to logic one indicating the DDR core that CSR register is updated with the new value or else reset to zero. Status register is successfully updated with the status on each read/write success/failure. The results also show that the DDR core is successfully reading the timing parameter from CSR for the command generation and loading this value into the counter logic. The improvement that can be incorporated in this work is error detection and correction of message using CRC (Cyclic Redundancy code) and status register.

FUTURE SCOPE

The **future scope of SDRAM (Synchronous Dynamic Random Access Memory) design** focuses on addressing evolving demands in technology, including higher performance, lower power consumption, and advanced features. Here are key areas of future scope for SDRAM design:

1. Higher Speeds and Capacities

- **Increased Bandwidth**: Future SDRAM designs will focus on achieving higher data transfer rates to meet the needs of high-performance applications like gaming, AI, and real-time processing.

- **Higher Density**: Larger capacities will be crucial for handling massive datasets in fields such as big data analytics, AI, and scientific computing.

2. Power Efficiency

- **Low Power SDRAM**: Optimizing SDRAM for mobile devices, IoT devices, and energy-efficient servers by reducing power consumption without compromising performance.

- **Advanced Power States**: Introducing more granular power-saving modes for idle or low-usage scenarios.

Reference

1. Husain, S., & Gupta, N. (2023). Enhancing fault tolerance in QCA-based RAM cells: A USE clock-driven approach with a novel majority voter. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 6, 100352.

2. Jasemi, M., Hessabi, S., & Bagherzadeh, N. (2020). Reliable and energy efficient MLC STT-RAM buffer for CNN accelerators. Computers & Electrical Engineering, 86, 106698.

3. Ruiz, G. A., & Michell, J. A. (2011). An efficient VLSI processor chip for variable block size integer motion estimation in H. 264/AVC. Signal Processing: Image Communication, 26(6), 289-303.

4. Abutaleb, M. M. (2013). Design and simulation of novel TLG–SET based RAM cell designs. Microelectronics Journal, 44(6), 504-510.

5. Aziza, H., Perez, A., & Portal, J. M. (2018). Resistive RAMs as analog trimming elements. Solid-State Electronics, 142, 52-55.

6.Khosroshahy, M. B., Moaiyeri, M. H., Navi, K., & Bagherzadeh, N. (2017). An energy and cost efficient majority-based RAM cell in quantum-dot cellular automata. Results in physics, 7, 3543-3551.

7. Abdullah-Al-Shafi, M., & Ziaur, R. (2019). Analysis and modeling of sequential circuits in QCA nano computing: RAM and SISO register study. Solid State Electronics Letters, 1(2), 73-83.

8. Dorożyński, P., Czarnul, P., Malinowski, A., Czuryło, K., Dorau, Ł., Maciejewski, M., & Skowron, P. (2016). Checkpointing of parallel mpi applications using mpi one-sided api with support for byte-addressable non-volatile ram. Procedia Computer Science, 80, 30-40.

9. John, P. K. (2017). BIST architecture for multiple RAMs in SoC. Procedia computer science, 115, 159-165.

10. Qian, J., Mo, F., Ding, H., Zhou, Z., Zhao, L., & Zhai, Z. (2021). An improved algorithm for accelerating reconfiguration of VLSI array. Integration, 79, 124-132.

11. Mehbodniya, A., Kumar, R., Bedi, P., Mohanty, S. N., Tripathi, R., & Geetha, A. (2022). VLSI implementation using fully connected neural networks for energy consumption over neurons. Sustainable Energy Technologies and Assessments, 52, 102058.

12. Sravani, M. M., & Durai, S. A. (2021). Attacks on cryptosystems implemented via VLSI: A review. Journal of Information Security and Applications, 60, 102861.

13.Kim, M., & So, J. (2023). VLSI design and FPGA implementation of state-of-charge and state-of-health estimation for electric vehicle battery management systems. Journal of Energy Storage, 73, 108876.

14.Palchaudhuri, A., & Dhar, A. S. (2021). Speed-area optimized VLSI architecture of multi-bit cellular automaton cell based random number generator on FPGA with testable logic support. Journal of Parallel and Distributed Computing, 151, 13-23.

15.Tan, W. M., & Jarvis, S. A. (2016). Heuristic solutions to the target identifiability problem in directional sensor networks. Journal of Network and Computer Applications, 65, 84-102.

Author- Mohit Ahirwar - He is currently M.Tech research student and he is pushing M.Tech from department of Electronics and communication, NRI Institute information science & Technology Bhopal. He is highly interested about for Electronic and VLSI domain.