

Design and Implementation of a 16-bit Memory Buffer Using Microwind

Rethika. S¹, Prithika.P², Kavya.M³, Yogesh.S⁴, Dr. S. Bhavani⁵

^{1,2,3,4}students, department Electronics Engineering – VLSI design and technology
Sri Shakthi institute of engineering and technology

⁵Guide, Professor & head of the department ECE, Sri Shakthi institute of engineering and technology

Abstract—This project presents the detailed design and implementation of a 16-bit Memory Buffer using CMOS logic in the Microwind environment. The memory buffer is a temporary storage unit used in digital systems to enable smooth data transfer between modules. This project includes schematic design, layout creation, circuit simulation, power and delay analysis, and validation of read/write operations. The design uses a 4-to-16 line decoder for address selection and CMOS-based latch elements for high reliability and low power consumption.

I. INTRODUCTION

A memory buffer is an essential component in VLSI digital systems used for temporary data storage. It allows digital systems to overcome timing differences between sender and receiver circuits. The aim of this project is to design a 16-bit memory buffer using CMOS logic and validate it through Microwind simulations. Microwind is a VLSI CAD tool that enables layout-level design and device-level analysis. The project demonstrates transistor-level implementation and discusses optimization for area, power, and performance.

II. OBJECTIVES

The primary objectives of the project are:

- To design a 16-bit memory buffer using CMOS logic gates.
- To implement address selection using a 4-to-16 decoder.
- To develop a complete layout using Microwind.
- To simulate read/write operations and verify correctness.

- To analyze power, delay, and area metrics.
- To understand trade-offs in VLSI memory design

III. LITERATURE

Survey Previous literature focuses on various memory design techniques including CMOS-based SRAM cells, register files, FIFO buffers, and temporary data storage systems. Earlier designs used large transistor counts and suffered from high static power consumption. With the development of tools like Microwind and DSCH, accurate transistor-level layout and simulation are possible. Research trends include size optimization, leakage reduction, and noise immunity improvements. This project builds upon prior studies by scaling the architecture to a 16-bit buffer with possible efficient CMOS design.

IV. METHODOLOGY

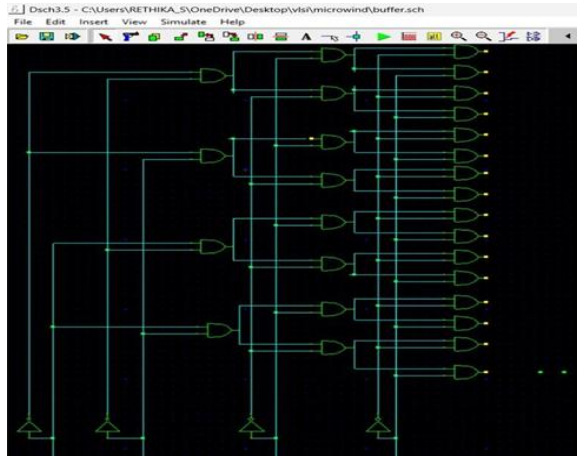
The methodology followed in the project consists of

1. Requirement analysis and function definition.
2. Schematic design using DSCH
3. Integration of decoder and buffer cells.
4. Layout design in Microwind using CMOS layers.
5. Design-rule check (DRC) and error correction.
6. Simulation of read/write operations, delay, and power.

Schematic Diagram :

The schematic includes:

- A 4-to-16 line decoder for selecting one of the sixteen memory locations.
- CMOS transmission gates and inverters for storage.
- Input, output, and control signals.
- Write enable (WE) and Read enable (RE) controls.



Verilog Code :

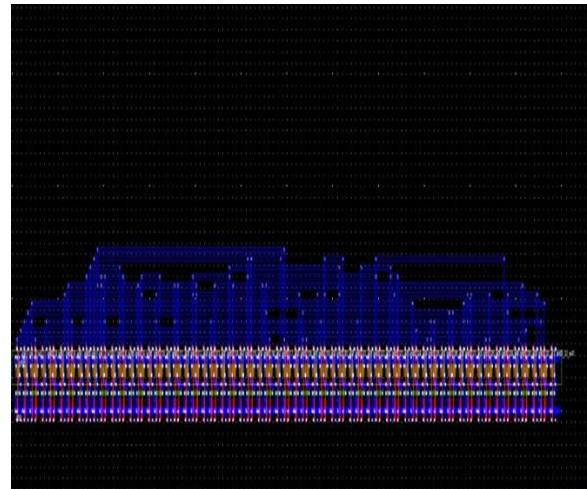
Sample core structure (expand as required):

```
module memory_buffer( input clk, we, re,
input [3:0] addr, input [15:0] data_in, output reg
[15:0] data_out );
reg [15:0] mem [0:15];
always @(posedge clk) begin if(we) mem[addr] <=
data_in; if(re) data_out <= mem[addr]; end
endmodule
```

Layout Design :

The CMOS layout is designed in Microwind using:

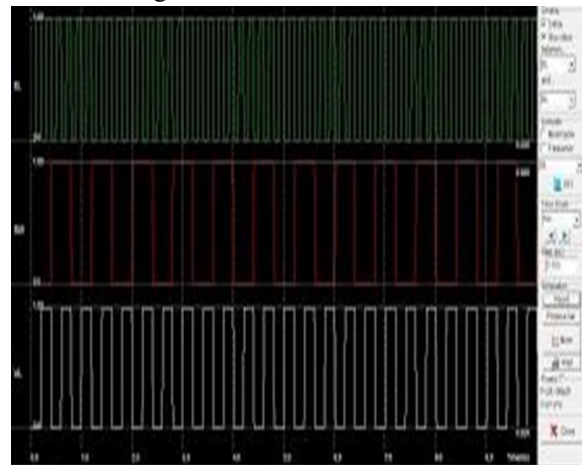
- Active layer (diffusion)
 - Poly-silicon gates
 - Metal layers for interconnections
 - Vias and contacts for connectivity
 - N-well and P-well regions for CMOS devices
- the layout is optimized for:
- Minimum area
 - Reduced parasitics
 - Better device matching
 - Clean routing and spacing



Simulation Results:

The simulation includes:

- Read operation waveform
- Write operation waveform
- Timing delay analysis
- Propagation delay calculation
- Static and dynamic power dissipation the results confirm correct memory behavior with stable read and write cycles under cmos switching conditions



Existing System

Existing systems include:

- 16-bit register files implemented in FPGA.
- FIFO memory buffers for serial communication.
- Basic SRAM-based 8-bit memory modules.

These designs focus on fast access rather than transistor-level optimization.

Proposed System:

The proposed 16-bit memory buffer uses CMOS latch elements and an efficient decode to balance area, speed, and power consumption

Key improvements:

- Lower leakage current
- Faster switching
- Reduced layout area
- High noise immunity due to CMOS logic

Applications:

Applications include:

- Microprocessors and CPUs
- Embedded system controllers
- Communication systems (UART, SPI, I2C)
- Digital buffering and synchronization units
- Data transfer pipeline

Advantages:

Advantages of the design:

- Low static power dissipation
- High noise immunity
- Fast read/write access
- Compact layout footprint
- Scalable architecture Future Enhancement :

Possible enhancements:

- Scaling to 32-bit or 64-bit memory buffers
- Implementing dual-port access
- Adding error correction code (ECC)
- Using FinFET/modern CMOS processes
- Adding pipeline buffering

V. CONCLUSION

This project successfully demonstrates the complete transistor-level design of a 16-bit memory buffer using CMOS logic in Microwind. The system is validated through schematic testing, layout design, and simulation. The design shows good area efficiency, reduced power consumption, and fast operation, making it

REFERENCES

- [1] Neil H.E. Weste, CMOS VLSI Design.
- [2] Microwind C DSCH User Manual.
- [3] Rabaey, Digital Integrated Circuits
- [4] IEEE papers on memory buffer optimization suitable for integration in advanced digital circuits.