# DAY 13
# MEMORY ELEMENTS - III

October 30, 2023

# SRAM Controller

### Test Procedures

1. Initialization:
   Reset (rstn) is asserted low, and all control signals are initialized. The clock is set to 0.

2. Reset De-assertion:
   After 1 time unit, rstn is de-asserted to initiate the operation of the SRAM.

3. Test Case 1 - Writing to SRAM:
   Write 24 to address 28.

4. Test Case 2 - Reading from SRAM:
   Read from address 28, expecting to retrieve 24.

5. Test Case 3 - Writing to SRAM:
   Write 30 to address 28.

6. Test Case 4 - Reading from SRAM:
   Read from address 28, expecting to retrieve 24 (since previous write should not have taken effect due to the lack of a write enable signal).

7. Test Case 5 - Reading from SRAM:
   Read from address 28, expecting to retrieve 24.

8. Termination:
   After 5 time units, the simulation is finished.

# SRAM Controller Module

```
 module sram_tb;

reg clk,rstn;
reg CS_b, WE_b, OE_b;
reg [10:0] addr;
wire [15:0] data_read;
reg [15:0] data_write;

wire [15:0] data;

assign data = ((CS_b == 0) && (WE_b == 0) && (OE_b == 1)) ? data_write :
'bz;
assign data_read = ((CS_b == 0) && (WE_b == 1) && (OE_b == 0)) ? data :
'bz;



sram INST0 (.clk(clk),.rstn(rstn),.CS_b(CS_b),.WE_b(WE_b),.OE_b(OE_b),
            .addr(addr),.data(data_read));

always begin
                #1 clk = ~clk;
        end

        initial begin
                $dumpfile("sram_sim.vcd");
                $dumpvars(0,sram_tb);
                CS_b = 1;
                WE_b = 0;
                data_write =0;
                OE_b =0;
                addr=0;

                clk = 0;
                rstn = 0;
                #1
                rstn = 1;

                // TestCase: Writing to SRAM.
                #1
```

```
                OE_b = 1;
                CS_b = 0;
                WE_b = 0;
                data_write = 24;
                addr = 28;

                // // TestCase: Reading the sram.
                #2
                OE_b = 0;
                CS_b = 0;
                WE_b = 1;
                data_write = 26;
                addr = 28;

                // TestCase: Writing to memory location 28;
                #1
                OE_b = 1;
                CS_b = 0;
                WE_b = 0;
                data_write = 30;
                addr = 28;

                // TestCase: Memory location 28 should be the same as before.
                #1
                OE_b = 0;
                CS_b = 0;
                WE_b = 1;
                data_write = 40;
                addr = 28;

                // TestCase: Memory location 28 should be the same as before.
                #1
                OE_b = 0;
                CS_b = 0;
                WE_b = 1;
                data_write = 40;
                addr = 28;

                #5
                $finish;
        end
endmodule
```

# Simulation

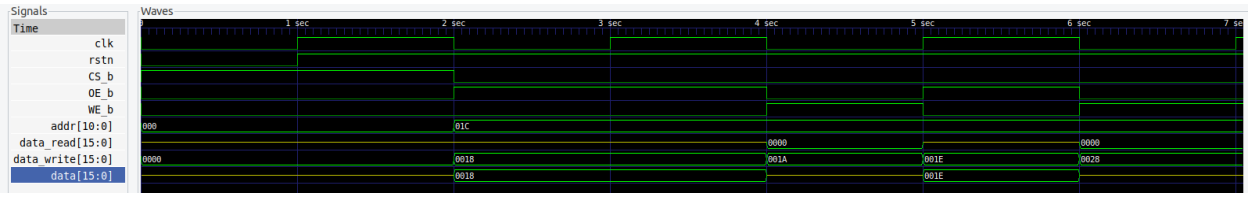


Figure 1: Simulation Results