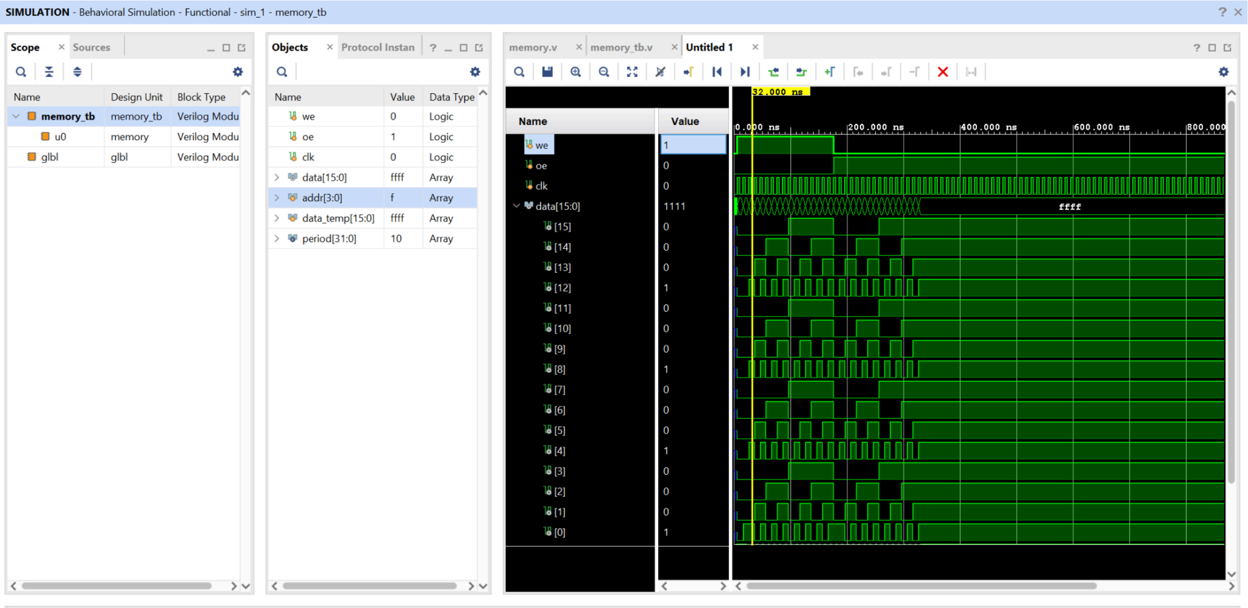
1) A copy of the test bench used to verify the functionality of part 1 of this lab. The test bench code should include useful comments (5 pts).

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/17/2025 10:05:58 AM
// Design Name:
// Module Name: memory_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module memory_tb( );
reg we, oe, clk;
wire [15:0] data;
reg [3:0] addr;
reg [15:0] data_temp;
parameter period = 10;
memory u0 (.we(we), .oe(oe), .clk(clk), .data(data), .addr(addr));

assign data = we?data_temp:16'hZZZZ;  //If we are not writing set data to a high impedance
always #(period/2) clk = ~clk;
initial
begin    clk = 0;    oe = 0;    we = 0; addr = 4'h0; data_temp = 16'h0000; //set things up so we
can save values
    #5 we = 1'b1;
    @(posedge clk)
    #1              addr = 4'h0; data_temp = 16'h0001;   // wait 1 ns before writing
    #period     addr = 4'h1; data_temp = 16'h1111;        // writing to memory same with the lines
below
    #period     addr = 4'h2; data_temp = 16'h2222;
    #period     addr = 4'h3; data_temp = 16'h3333;
    #period     addr = 4'h4; data_temp = 16'h4444;
    #period     addr = 4'h5; data_temp = 16'h5555;
    #period     addr = 4'h6; data_temp = 16'h6666;
    #period     addr = 4'h7; data_temp = 16'h7777;
    #period     addr = 4'h8; data_temp = 16'h8888;
    #period     addr = 4'h9; data_temp = 16'h9999;
    #period     addr = 4'hA; data_temp = 16'hAAAA;
    #period     addr = 4'hB; data_temp = 16'hBBBB;
    #period     addr = 4'hC; data_temp = 16'hCCCC;
    #period     addr = 4'hD; data_temp = 16'hDDDD;
    #period     addr = 4'hE; data_temp = 16'hEEEE;
    #period     addr = 4'hF; data_temp = 16'hFFFF;

    #period     addr = 4'h0; we= 1'b0; oe = 1'b1;      // reading from memory same as the lines
below though we only have to set things up once
    #period     addr = 4'h1;
    #period     addr = 4'h2;
    #period     addr = 4'h3;
    #period     addr = 4'h4;
    #period     addr = 4'h5;
    #period     addr = 4'h6;
    #period     addr = 4'h7;
    #period     addr = 4'h8;
    #period     addr = 4'h9;
    #period     addr = 4'hA;
    #period     addr = 4'hB;
    #period     addr = 4'hC;
    #period     addr = 4'hD;
    #period     addr = 4'hE;
    #period     addr = 4'hF;


    #1000 $finish;
end
endmodule
```

2) A description of the new tools, technologies, or concepts used to implement this lab (5 pts).

This week we learned to use memory which was a core part of this lab. We learned that we were able to set up memory like 16x16 SRAM. We had to use SRAM because we can use flip flops with it and we don't have to set up a bunch of refresh systems that DRAM would require. The 16x16 means there are 16 addresses and each address contains 16 bits. WE then learned to write and load data from the memory

3) Include Vivado screen shots of simulations, the elaborated schematic of the entire system, and other meaningful information to document your work (5 pts).



SIMULATION - Behavioral Simulation - Functional - sim_1 - memory_tb

Scope | Sources

Name	Design Unit	Block Type
memory_tb	memory_tb	Verilog Modu
u0	memory	Verilog Modu
glbl	glbl	Verilog Modu

Objects | Protocol Instan

Name	Value	Data Type
we	0	Logic
oe	1	Logic
clk	0	Logic
data[15:0]	ffff	Array
addr[3:0]	f	Array
data_temp[15:0]	ffff	Array
period[31:0]	10	Array

memory.v | memory_tb.v | Untitled 1

Name	Value
we	1
oe	0
clk	0
data[15:0]	1111
[15]	0
[14]	0
[13]	0
[12]	1
[11]	0
[10]	0
[9]	0
[8]	1
[7]	0
[6]	0
[5]	0
[4]	1
[3]	0
[2]	0
[1]	0
[0]	1

Figure 1: test bench simulation of memory

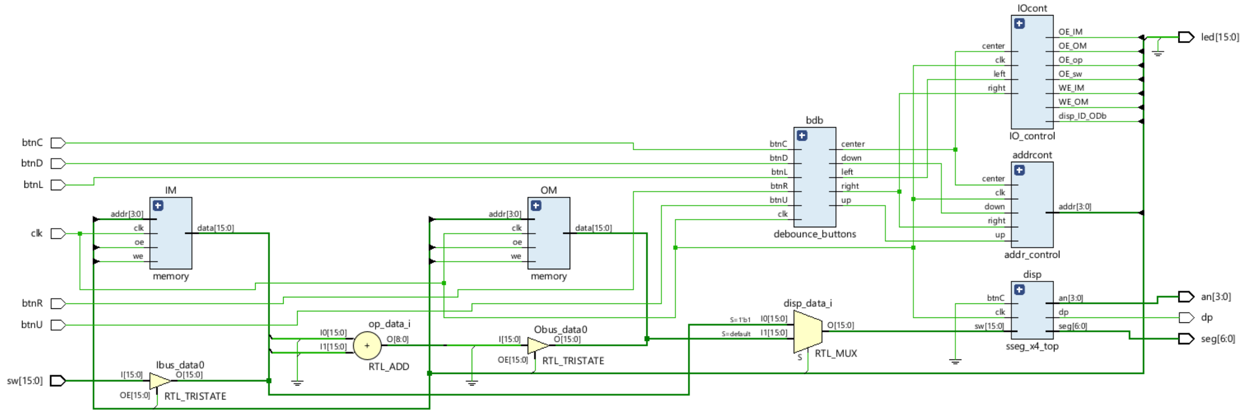


Figure 2: Elaborated design of final code

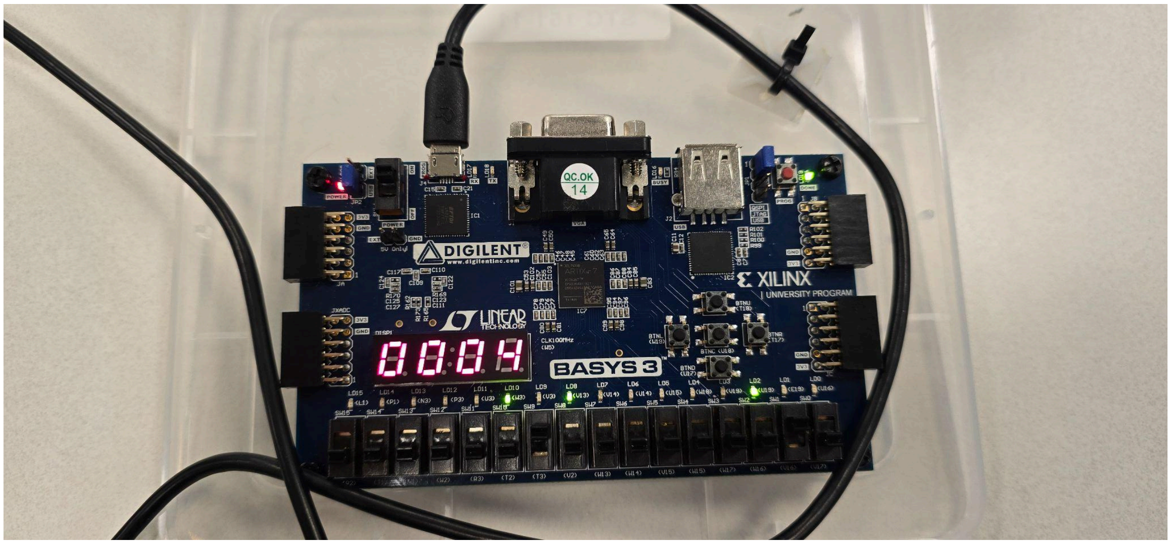


Figure 3: final functionality of the code

4) In your conclusion statement, please discuss your results, the method of testing, and include the level of functionality of the lab (10 pts).

This lab felt like a lot at first because we were using LEDs, display, switches, all the buttons and memory to top it all off. After diving into the code I was able to see how and more importantly why this all worked. I gained a thorough understanding of how memory works and how we can not only set it up but write and read from it as well.