

EE533_Lab8_Report

1. Instruction Part

1.1 Purpose

- Bubble sort the packet payload data
 - Each component in payload is 64-bit wide.
 - First Line would be the array size.
 - Following elements are the array elements.

1.2 Definition

- Instruction opcode lookup table

| Instr | OP Code [31:26] |
|-------|-----------------|
| noop | 000000 |
| addi | 000001 |
| movi | 000010 |
| lw | 000011 |
| sw | 000100 |
| beq | 000101 |
| bgt | 000110 |
| blt | 000111 |
| j | 001000 |
| subi | 001001 |

- Instruction Table

| Addr | Label | Instr | OP Code [31:26] | Rs [25:21] | Rt [20:16] | Offset [15:0] |
|------|------------|-----------------|-----------------|------------|------------|---------------|
| 0 | | lw r1, r0(#3) | 000011 | 5'd0 | 5'd1 | 16'd3 |
| 1 | | movi r2, #0 | 000010 | 5'd0 | 5'd2 | 16'd0 |
| 2 | | movi r3, #1 | 000010 | 5'd0 | 5'd3 | 16'd1 |
| 3 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 4 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 5 | outer_loop | beq r1, r3, end | 000101 | 5'd3 | 5'd1 | 16'd27 |

| Addr | Label | Instr | OP Code [31:26] | Rs [25:21] | Rt [20:16] | Offset [15:0] |
|------|------------|-------------------------|--------------------|---------------|---------------|------------------|
| 6 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 7 | inner_loop | beq r3, r1, next_out | 000101 | 5'd1 | 5'd3 | 16'd21 |
| 8 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 9 | | lw r4, r2(#1) | 000011 | 5'd2 | 5'd4 | 16'd1 |
| 10 | | lw r5, r3(#1) | 000011 | 5'd3 | 5'd5 | 16'd1 |
| 11 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 12 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 13 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 14 | | blt r4, r5, no_swap | 000111 | 5'd5 | 5'd4 | 16'd18 |
| 15 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 16 | | sw r4, r3(#1) | 000100 | 5'd3 | 5'd4 | 16'd1 |
| 17 | | sw r5, r2(#1) | 000100 | 5'd2 | 5'd5 | 16'd1 |
| 18 | no_swap | addi r2, r2, #1 | 000001 | 5'd2 | 5'd2 | 16'd1 |
| 19 | | addi r3, r3, #1 | 000001 | 5'd3 | 5'd3 | 16'd1 |
| 20 | | j inner_loop | 001000 | 5'd0 | 5'd0 | 16'd7 |
| 21 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 22 | next_out | subi r1, r1, #1 | 001001 | 5'd1 | 5'd1 | 16'd1 |
| 23 | | movi r2, #0 | 000010 | 5'd0 | 5'd2 | 16'd0 |
| 24 | | movi r3, #1 | 000010 | 5'd0 | 5'd3 | 16'd1 |
| 25 | | j outer_loop | 001000 | 5'd0 | 5'd0 | 16'd5 |
| 26 | | noop | 000000 | 5'd0 | 5'd0 | 16'd0 |
| 27 | end | j end | 001000 | 5'd0 | 5'd0 | 16'd27 |

2. Packet Part

2.1 Packet Header Format Design

| VER | HLEN | Service | Total Length | Identification | Flags | Fragmentation Offset |
|------------------------|-------|----------|-----------------|-------------------|--------|----------------------|
| 4 bit | 4 bit | 8 bits | 16 bits | 16 bits | 3 bits | 13 bits |
| TTL | | Protocol | Header Checksum | Source IP Address | | |
| 8 bits | | 8 bits | 16 bits | 32 bits | | |
| Destination IP Address | | | | Option | | |
| 32 bits | | | | 32 bits | | |

2.2 Sample Initial Packet

| VER | HLEN | Service | Total Length | Identification | Flags | Fragmentation Offset |
|--|---------|-----------|---------------------|---|-----------|----------------------|
| 4 bit | 4 bit | 8 bits | 16 bits | 16 bits | 3 bits | 13 bits |
| IPv4 | 24 Byte | 0 | 64 Bytes | 1C 46 | (0, D, M) | 0 |
| 0100 | 0110 | 0000 0000 | 0000 0000 0100 0000 | 0001 1100 0100 0110 | 010 | 0 0000 0000 0000 |
| TTL | | Protocol | Header Checksum | Source IP Address | | |
| 8 bits | | 8 bits | 16 bits | 32 bits | | |
| 4 | | TCP | 2A6D | 10.0.13.3 (node 1) | | |
| 0000 0100 | | 0000 0110 | 0010 1010 0110 1101 | 0000 1010 0000 0000 0000 1101 0000 0011 | | |
| Destination IP Address | | | | Option | | |
| 32 bits | | | | 32 bits | | |
| 10.0.14.3 (node 2) | | | | 0 | | |
| 0000 1010 0000 0000 0000 1110 0000 0011 | | | | 0000 0000 0000 0000 0000 0000 0000 0000 | | |
| Payload | | | | | | |
| 64 bits | | | | | | |
| 4 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 | | | | | | |
| 4 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 | | | | | | |
| 3 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0011 | | | | | | |
| 2 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 | | | | | | |
| 1 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 | | | | | | |

2.3 Packet Expected to get after Processing Bubble Sort Code by Pipeline

| VER | HLEN | Service | Total Length | Identification | Flags | Fragmentation Offset |
|--|---------|-----------|---------------------|---|-----------|----------------------|
| 4 bit | 4 bit | 8 bits | 16 bits | 16 bits | 3 bits | 13 bits |
| IPv4 | 24 Byte | 0 | 64 Bytes | 1C 46 | (0, D, M) | 0 |
| 0100 | 0110 | 0000 0000 | 0000 0000 0100 0000 | 0001 1100 0100 0110 | 010 | 0 0000 0000 0000 |
| TTL | | Protocol | Header Checksum | Source IP Address | | |
| 8 bits | | 8 bits | 16 bits | 32 bits | | |
| 4 | | TCP | 2A6D | 10.0.13.3 (node 1) | | |
| 0000 0100 | | 0000 0110 | 0010 1010 0110 1101 | 0000 1010 0000 0000 0000 1101 0000 0011 | | |
| Destination IP Address | | | | Option | | |
| 32 bits | | | | 32 bits | | |
| 10.0.14.3 (node 2) | | | | 0 | | |
| 0000 1010 0000 0000 0000 1110 0000 0011 | | | | 0000 0000 0000 0000 0000 0000 0000 0000 | | |
| Payload | | | | | | |
| 64 bits | | | | | | |
| 4 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 | | | | | | |
| 1 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 | | | | | | |
| 2 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 | | | | | | |
| 3 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0011 | | | | | | |
| 4 | | | | | | |
| 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 | | | | | | |

3. Pipeline Updated Part

3.1 D_MEM Mode Definition

| Mode Name | Mode Code | Description |
|-----------------|-----------|--|
| FIFO_IN | 00 | BRAM working as FIFO and write in packet, WP <= WP + 1 |
| FIFO_OUT | 01 | BRAM working as FIFO and read out packet, RP <= RP + 1 |
| SRAM_PROCESSING | 10 | BRAM working as D_MEM in pipeline processor |

3.2 RP (as Head Address)

3.2.1 RP_Reg

- Verilog

```
timescale 1ns / 1ps

module RP_Reg
(
    input clk,
    input rst,
    input RP_en,
    input [7:0] RP_next,

    output reg [7:0] RP
);

    always @(posedge clk) begin
        if (rst) begin
            RP <= 0;
        end
        else if(RP_en) begin
            RP <= RP_next;
        end
    end
endmodule
```

- Testbench

[illegible]

```

module RP_Reg_tb;

    // Inputs
    reg clk;
    reg rst;
    reg RP_en;
    reg [7:0] RP_next;

    // Outputs
    wire [7:0] RP;

    // Instantiate the Unit Under Test (UUT)
    RP_Reg uut (
        .clk(clk),
        .rst(rst),
        .RP_en(RP_en),
        .RP_next(RP_next),
        .RP(RP)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 1;
        rst = 1;
        RP_en = 0;
        RP_next = 0;

        // wait 100 ns for global reset to finish
        @(posedge clk);
        rst = 0;

        // Add stimulus here
        @(posedge clk);
        RP_en = 1;
        RP_next = 8'd0;

        @(posedge clk);
        RP_en = 1;
        RP_next = 8'd1;

        @(posedge clk);
        RP_en = 1;
        RP_next = 8'd2;

        @(posedge clk);
        RP_en = 1;
        RP_next = 8'd3;

        @(posedge clk);
        RP_en = 0;
        RP_next = 8'd4;

        @(posedge clk);

```

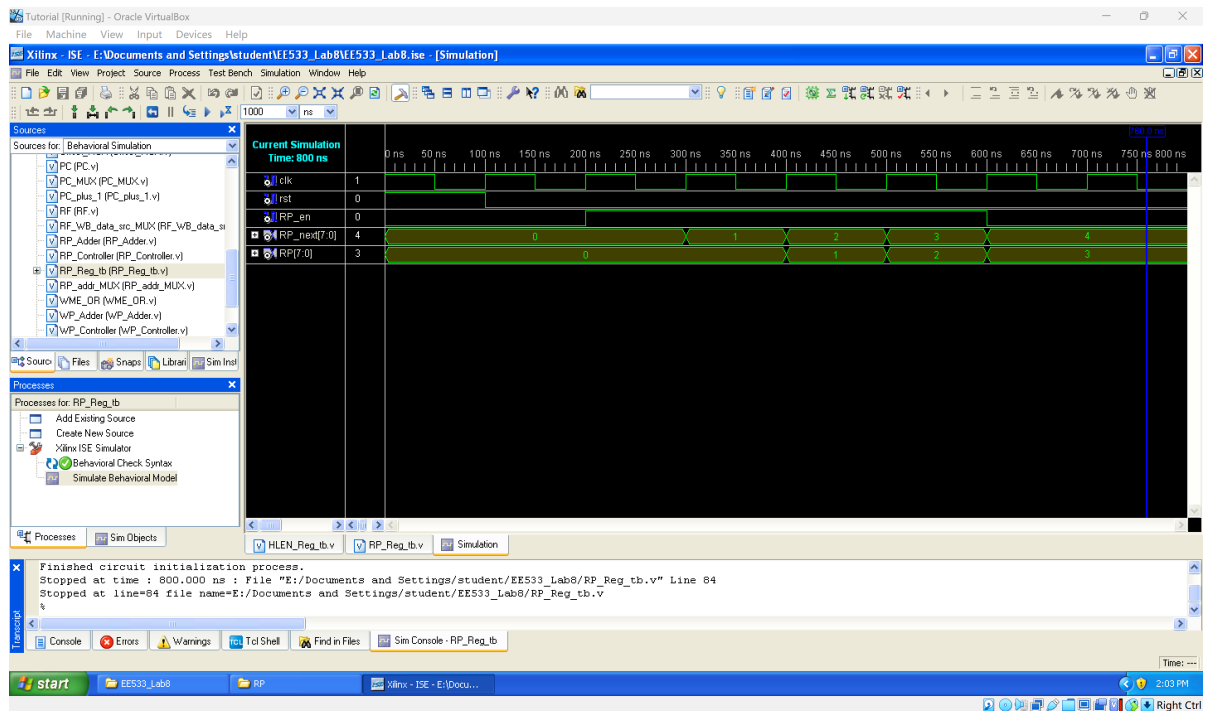
```
RP_en = 0;
RP_next = 8'd4;

@(posedge clk);
$stop;

end

endmodule
```

- Waveform



3.2.2 RP_Adder

- Verilog

```
timescale 1ns / 1ps

module RP_Adder
(
    input [7:0] RP,
    output [7:0] RP_next
);

    assign RP_next = RP + 1;

endmodule
```

- Testbench

```
timescale 1ns / 1ps

////////////////////////////////////
// Company:
```

```

// Engineer:
//
// Create Date:    14:04:08 03/08/2025
// Design Name:    RP_Adder
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/RP_Adder_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: RP_Adder
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module RP_Adder_tb;

    // Inputs
    reg [7:0] RP;

    // Outputs
    wire [7:0] RP_next;

    // Instantiate the Unit Under Test (UUT)
    RP_Adder uut (
        .RP(RP),
        .RP_next(RP_next)
    );

    initial begin
        // Initialize Inputs
        RP = 0;

        // wait 100 ns for global reset to finish
        #100;
        RP = 8'd1;

        // Add stimulus here
        #100;
        RP = 8'd2;

        #100;
        RP = 8'd3;

        #100;
        RP = 8'd9;

        #100;
        RP = 8'd1;

        #100;

```

```

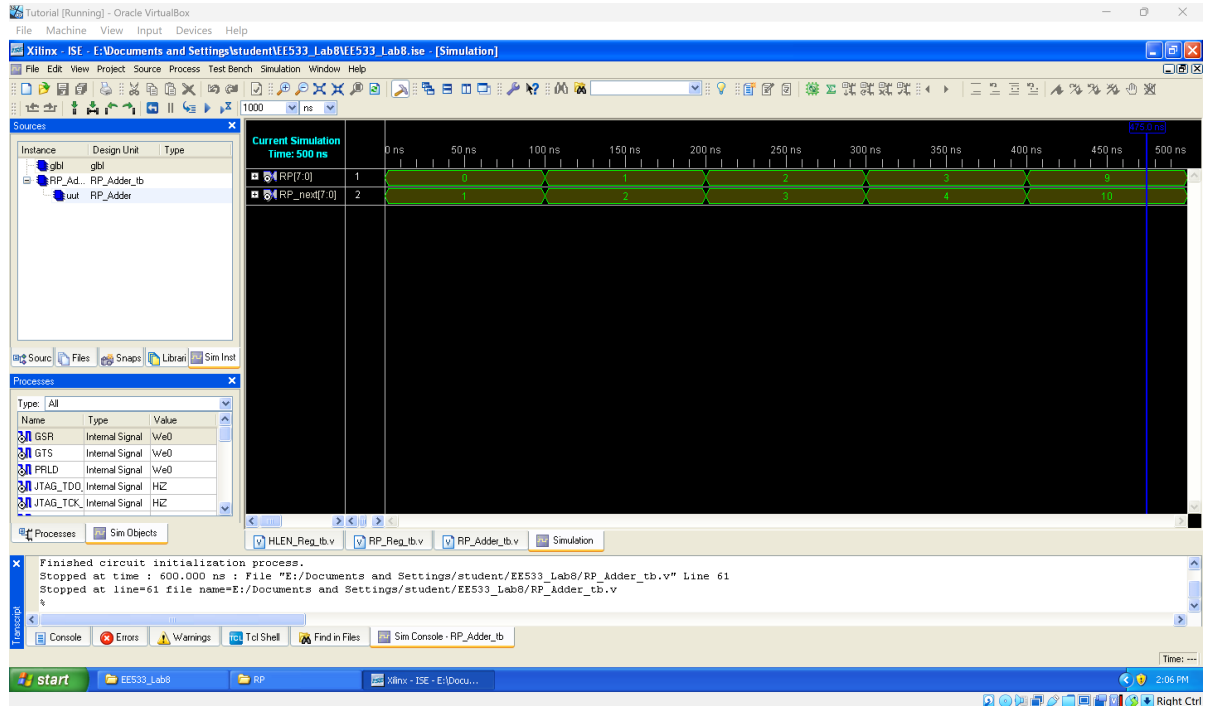
$stop;

end

endmodule

```

- Waveform



3.2.3 RP_Controller

- Verilog

```

`timescale 1ns / 1ps

module RP_Controller
(
    input [1:0] mode_code,

    output RP_en
);

    assign RP_en = (mode_code == 2'b01) ? 1 : 0;

endmodule

```

- Testbench

```

`timescale 1ns / 1ps

/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:07:53 03/08/2025
// Design Name:    RP_Controller

```



```

// Module Name:    E:/Documents and Settings/student/EE533_Lab8/RP_Controller_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: RP_Controller
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module RP_Controller_tb;

    // Inputs
    reg [1:0] mode_code;

    // Outputs
    wire RP_en;

    // Instantiate the Unit Under Test (UUT)
    RP_Controller uut (
        .mode_code(mode_code),
        .RP_en(RP_en)
    );

    initial begin
        // Initialize Inputs
        mode_code = 2'b00;

        // wait 100 ns for global reset to finish
        #100;

        // Add stimulus here
        #100;
        mode_code = 2'b00;

        #100;
        mode_code = 2'b01;

        #100;
        mode_code = 2'b10;

        #100;
        mode_code = 2'b11;

        #100;

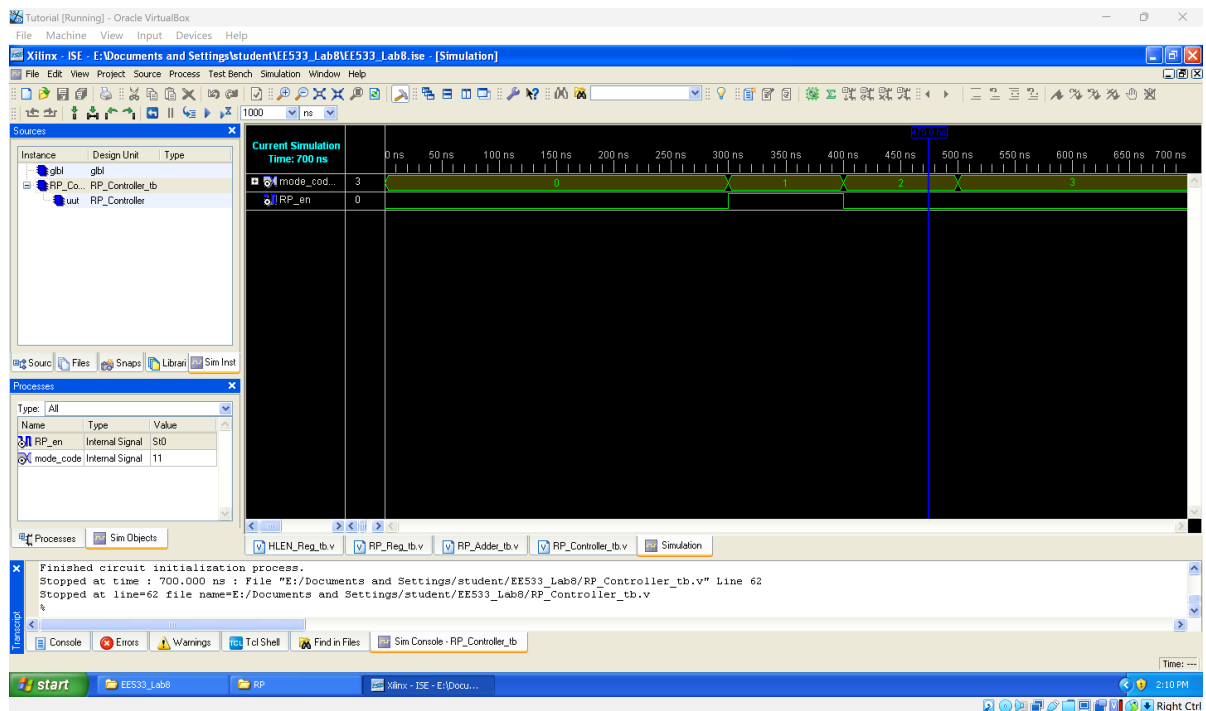
        #100;
        $stop;

    end

```

```
endmodule
```

- Waveform



3.2.4 RP_addr_MUX

- Verilog

```
`timescale 1ns / 1ps

module RP_addr_MUX
(
    input RP_ctr1,

    input [7:0] SRAM_addr,
    input [7:0] RP,

    output [7:0] D_raddr
);

    assign D_raddr = RP_ctr1 ? RP : SRAM_addr;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:13:34 03/08/2025
// Design Name:    RP_addr_MUX
```

```

// Module Name:    E:/Documents and Settings/student/EE533_Lab8/RP_addr_MUX_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: RP_addr_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module RP_addr_MUX_tb;

    // Inputs
    reg RP_ctrl;
    reg [7:0] SRAM_addr;
    reg [7:0] RP;

    // Outputs
    wire [7:0] D_raddr;

    // Instantiate the Unit Under Test (UUT)
    RP_addr_MUX uut (
        .RP_ctrl(RP_ctrl),
        .SRAM_addr(SRAM_addr),
        .RP(RP),
        .D_raddr(D_raddr)
    );

    initial begin
        // Initialize Inputs
        RP_ctrl = 0;
        SRAM_addr = 0;
        RP = 0;

        // wait 100 ns for global reset to finish
        #100;

        // Add stimulus here
        #100;
        RP_ctrl = 0;
        SRAM_addr = 8'd3;
        RP = 8'd5;

        #100;
        RP_ctrl = 1;
        SRAM_addr = 8'd9;
        RP = 8'd7;

        #100;
        RP_ctrl = 0;
    end

```

```

SRAM_addr = 8'd2;
RP = 8'd7;

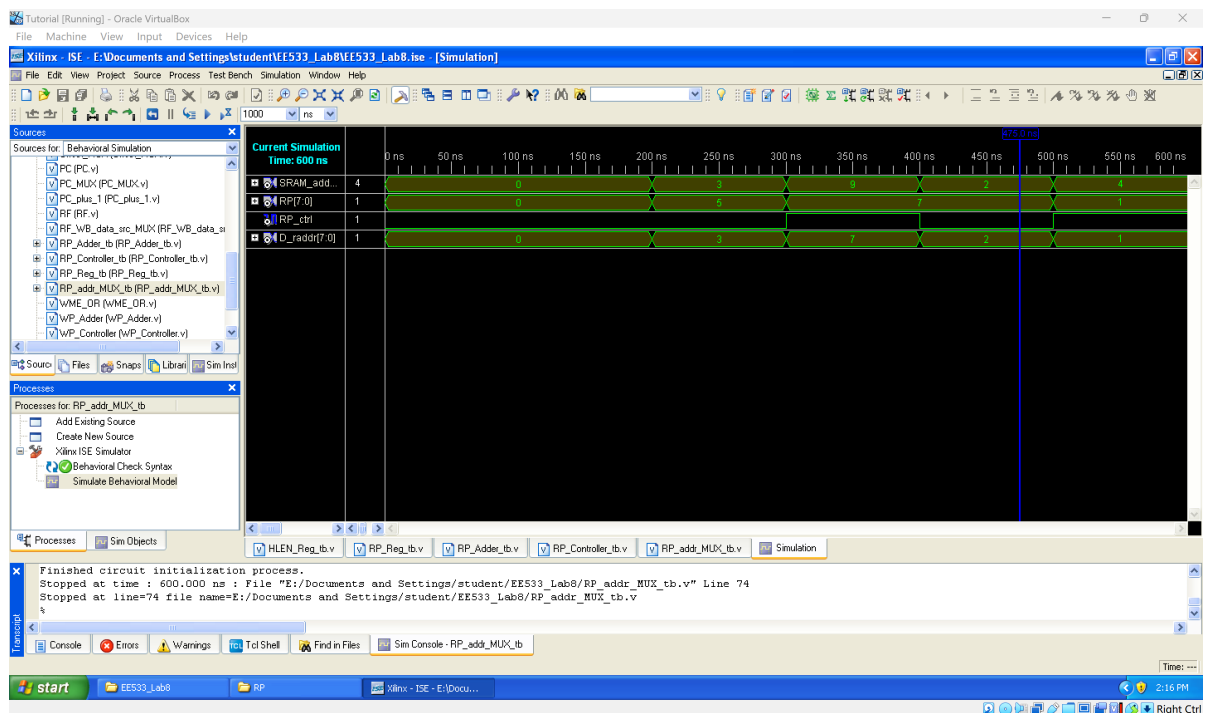
#100;
RP_ctrl = 1;
SRAM_addr = 8'd4;
RP = 8'd1;

```

```
end
```

```
endmodule
```

- Waveform



3.3 WP (as Tail Address)

3.3.1 WP_Reg

- Verilog

```

`timescale 1ns / 1ps

module WP_Reg
(
    input clk,
    input rst,
    input WP_en,
    input [7:0] WP_next,

    output reg [7:0] WP
);

always @(posedge clk) begin

```

```

        if (rst) begin
            WP <= 0;
        end
        else if(WP_en) begin
            WP <= WP_next;
        end
    end
endmodule

```

- Testbench

```

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:18:15 03/08/2025
// Design Name:    WP_Reg
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/WP_Reg_tb.v
// Project Name:    EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: WP_Reg
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module WP_Reg_tb;

    // Inputs
    reg clk;
    reg rst;
    reg WP_en;
    reg [7:0] WP_next;

    // Outputs
    wire [7:0] WP;

    // Instantiate the Unit Under Test (UUT)
    WP_Reg uut (
        .clk(clk),
        .rst(rst),
        .WP_en(WP_en),
        .WP_next(WP_next),
        .WP(WP)
    );

```

```

always #50 clk = ~clk;

initial begin
    // Initialize Inputs
    clk = 1;
    rst = 1;
    WP_en = 0;
    WP_next = 0;

    // wait 100 ns for global reset to finish
    @(posedge clk);
    rst = 0;

    // Add stimulus here
    @(posedge clk);
    WP_en = 1;
    WP_next = 8'd1;

    @(posedge clk);
    WP_en = 1;
    WP_next = 8'd2;

    @(posedge clk);
    WP_en = 1;
    WP_next = 8'd8;

    @(posedge clk);
    WP_en = 0;
    WP_next = 8'd3;

    @(posedge clk);
    WP_en = 0;
    WP_next = 8'd2;

    @(posedge clk);
    WP_en = 1;
    WP_next = 8'd9;

    @(posedge clk);

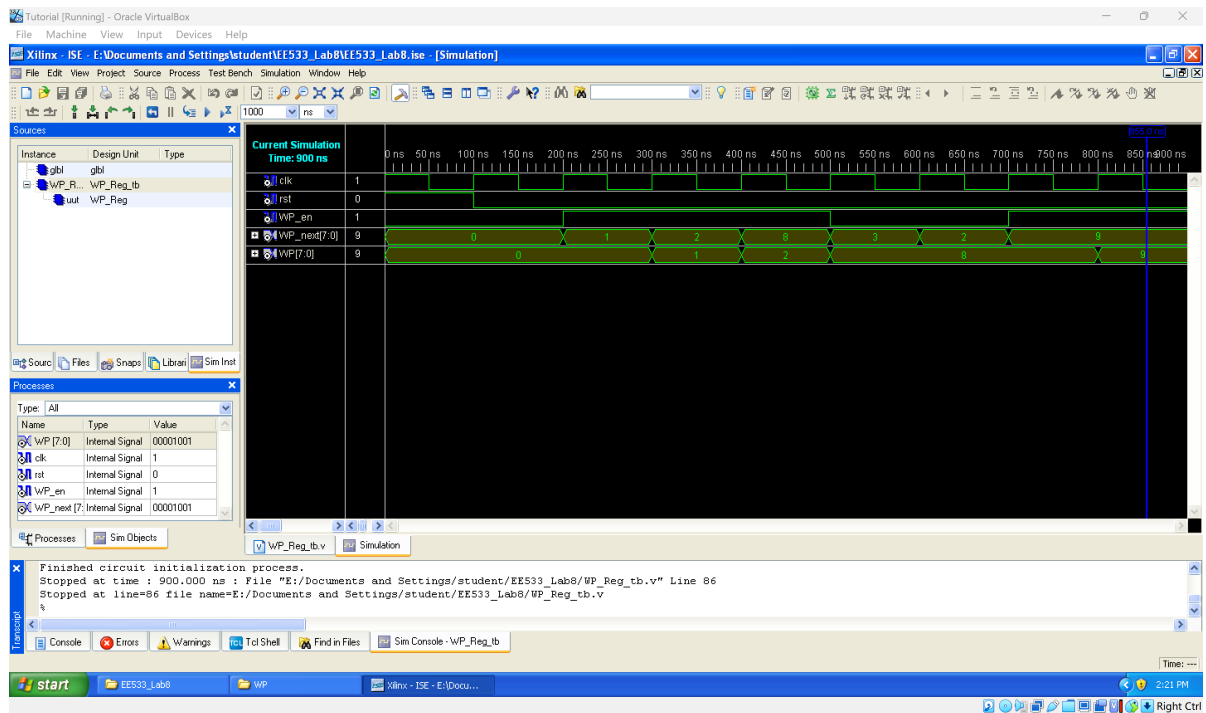
    @(posedge clk);
    $stop;

end

endmodule

```

- Waveform



3.3.2 WP_Adder

- Verilog

```
`timescale 1ns / 1ps

module WP_Adder
(
    input [7:0] WP,
    output [7:0] WP_next
);

    assign WP_next = WP + 1;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:22:37 03/08/2025
// Design Name:    WP_Adder
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/WP_Adder_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: WP_Adder
//
```

```

// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module WP_Adder_tb;

    // Inputs
    reg [7:0] WP;

    // Outputs
    wire [7:0] WP_next;

    // Instantiate the Unit Under Test (UUT)
    WP_Adder uut (
        .WP(WP),
        .WP_next(WP_next)
    );

    initial begin
        // Initialize Inputs
        WP = 0;

        // wait 100 ns for global reset to finish
        #100;
        WP = 8'd1;

        // Add stimulus here
        #100;
        WP = 8'd2;

        #100;
        WP = 8'd5;

        #100;
        WP = 8'd9;

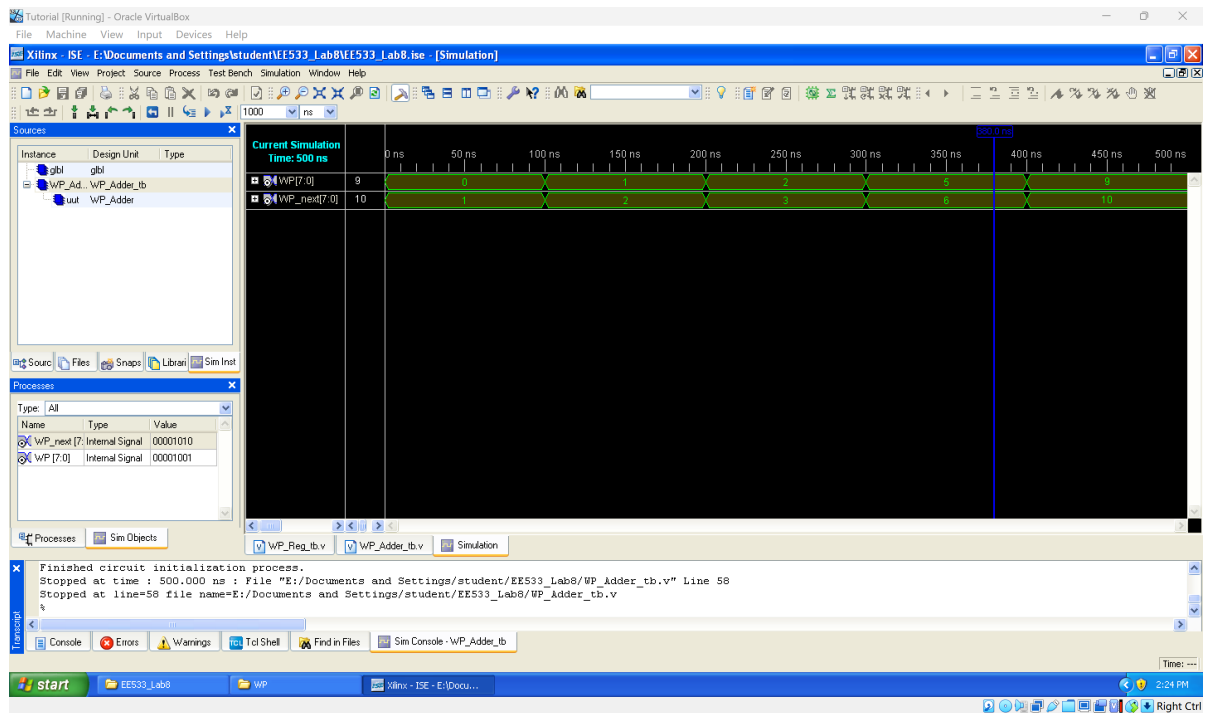
        #100;
        $stop;

    end

endmodule

```

- Waveform



3.3.3 WP_Controller

- Verilog

```
`timescale 1ns / 1ps

module WP_Controller
(
    input [1:0] mode_code,

    output WP_en
);

    assign WP_en = (mode_code == 2'b00) ? 1 : 0;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:25:30 03/08/2025
// Design Name:    WP_Controller
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/WP_Controller_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: WP_Controller
//
```

```

// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module WP_Controller_tb;

    // Inputs
    reg [1:0] mode_code;

    // Outputs
    wire WP_en;

    // Instantiate the Unit Under Test (UUT)
    WP_Controller uut (
        .mode_code(mode_code),
        .WP_en(WP_en)
    );

    initial begin
        // Initialize Inputs
        mode_code = 2'b00;

        // wait 100 ns for global reset to finish
        #100;
        mode_code = 2'b01;

        // Add stimulus here
        #100;
        mode_code = 2'b10;

        #100;
        mode_code = 2'b11;

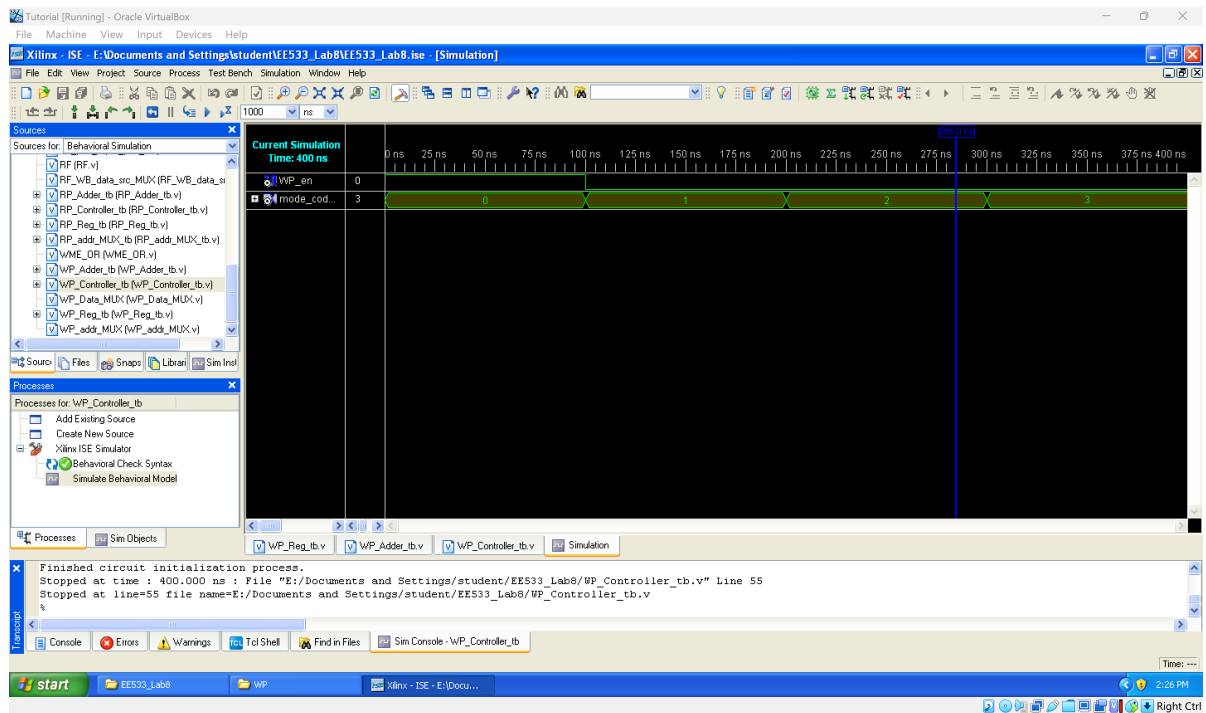
        #100;
        $stop;

    end

endmodule

```

- Waveform



3.3.4 WP_addr_MUX

- Verilog

```
`timescale 1ns / 1ps

module WP_addr_MUX
(
    input WP_ctrl,

    input [7:0] SRAM_addr,
    input [7:0] WP,

    output [7:0] D_waddr
);

    assign D_waddr = WP_ctrl ? WP : SRAM_addr;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:27:52 03/08/2025
// Design Name:    WP_addr_MUX
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/WP_addr_MUX_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
```

```

//
// Verilog Test Fixture created by ISE for module: WP_addr_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module WP_addr_MUX_tb;

    // Inputs
    reg WP_ctrl;
    reg [7:0] SRAM_addr;
    reg [7:0] WP;

    // Outputs
    wire [7:0] D_waddr;

    // Instantiate the Unit Under Test (UUT)
    WP_addr_MUX uut (
        .WP_ctrl(WP_ctrl),
        .SRAM_addr(SRAM_addr),
        .WP(WP),
        .D_waddr(D_waddr)
    );

    initial begin
        // Initialize Inputs
        WP_ctrl = 0;
        SRAM_addr = 0;
        WP = 0;

        // Wait 100 ns for global reset to finish
        #100;
        WP_ctrl = 0;
        SRAM_addr = 8'd1;
        WP = 8'd2;

        // Add stimulus here
        #100;
        WP_ctrl = 1;
        SRAM_addr = 8'd6;
        WP = 8'd4;

        #100;
        WP_ctrl = 0;
        SRAM_addr = 8'd9;
        WP = 8'd5;

        #100;
        WP_ctrl = 1;
        SRAM_addr = 8'd4;
        WP = 8'd1;
    end

```

```

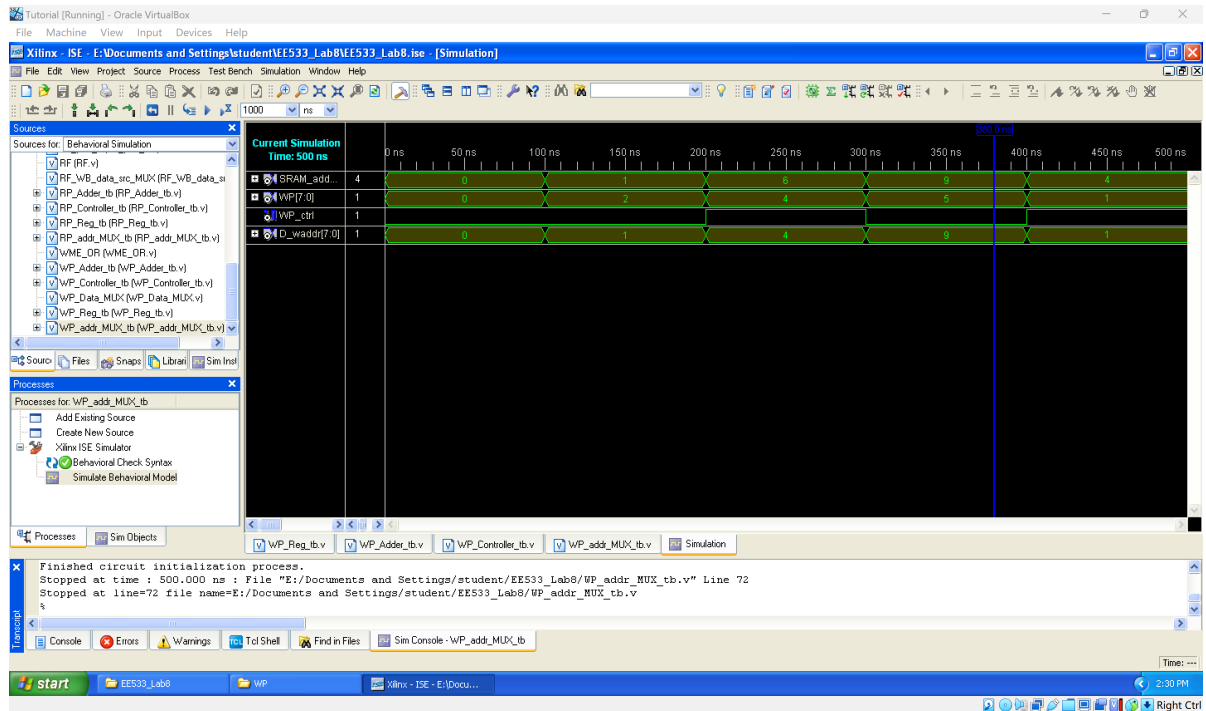
#100;
$stop;

end

endmodule

```

- Waveform



3.3.5 WP_Data_MUX

- Verilog

```

`timescale 1ns / 1ps

module WP_Data_MUX
(
    input WP_ctrl,

    input [63:0] SRAM_Din,
    input [63:0] FIFO_Din,

    output [63:0] D_MEM_Din
);

    assign D_MEM_Din = WP_ctrl ? FIFO_Din : SRAM_Din;

endmodule

```

- Testbench

```

`timescale 1ns / 1ps

////////////////////////////////////

```

```

// Company:
// Engineer:
//
// Create Date:    14:31:20 03/08/2025
// Design Name:    WP_Data_MUX
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/WP_Data_MUX_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: WP_Data_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module WP_Data_MUX_tb;

    // Inputs
    reg WP_ctrl;
    reg [63:0] SRAM_Din;
    reg [63:0] FIFO_Din;

    // Outputs
    wire [63:0] D_MEM_Din;

    // Instantiate the Unit Under Test (UUT)
    WP_Data_MUX uut (
        .WP_ctrl(WP_ctrl),
        .SRAM_Din(SRAM_Din),
        .FIFO_Din(FIFO_Din),
        .D_MEM_Din(D_MEM_Din)
    );

    initial begin
        // Initialize Inputs
        WP_ctrl = 0;
        SRAM_Din = 0;
        FIFO_Din = 0;

        // wait 100 ns for global reset to finish
        #100;
        WP_ctrl = 1;
        SRAM_Din = 64'd2;
        FIFO_Din = 64'd9;

        // Add stimulus here
        #100;
        WP_ctrl = 0;
        SRAM_Din = 64'd2;
        FIFO_Din = 64'd9;
    end

```

```

#100;
WP_ctrl = 1;
SRAM_Din = 64'd7;
FIFO_Din = 64'd6;

#100;
WP_ctrl = 0;
SRAM_Din = 64'd7;
FIFO_Din = 64'd6;

#100;
WP_ctrl = 1;
SRAM_Din = 64'd3;
FIFO_Din = 64'd34;

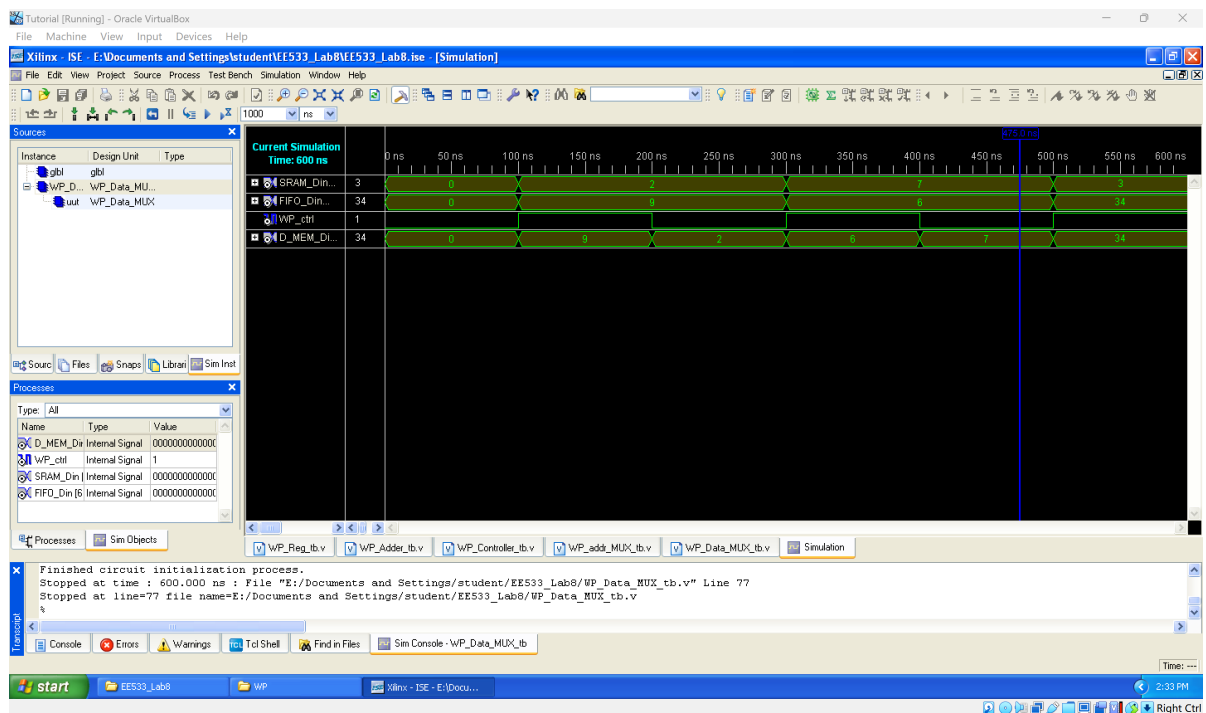
#100;
$stop;

```

end

endmodule

- Waveform



3.4 HLEN & Offset Updated Part

3.4.1 HLEN_Reg

- Verilog

```

`timescale 1ns / 1ps

module HLEN_Reg
(
    input clk,

```

```

    input rst,
    input HLEN_Reg_write_en,
    input [63:0] HLEN_in,

    output reg [63:0] HLEN_out
);

    reg [63:0] HLEN;

    always @(posedge clk) begin
        if (rst) begin
            HLEN <= 0;
        end
        else if (HLEN_Reg_write_en) begin
            HLEN <= HLEN_in;
        end
    end

    always @(*) begin
        HLEN_out = HLEN;
    end

endmodule

```

- Testbench

```

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    13:29:22 03/08/2025
// Design Name:    HELN_Reg
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/HLEN_Reg_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: HELN_Reg
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module HLEN_Reg_tb;

    // Inputs
    reg clk;
    reg rst;

```



```

reg HLEN_Reg_write_en;
reg [63:0] HLEN_in;

// Outputs
wire [63:0] HLEN_out;

// Instantiate the Unit Under Test (UUT)
HLEN_Reg uut (
    .clk(clk),
    .rst(rst),
    .HLEN_Reg_write_en(HLEN_Reg_write_en),
    .HLEN_in(HLEN_in),
    .HLEN_out(HLEN_out)
);

always #50 clk = ~clk;

initial begin
    // Initialize Inputs
    clk = 1;
    rst = 1;
    HLEN_Reg_write_en = 0;
    HLEN_in = 0;

    // wait 100 ns for global reset to finish
    @(posedge clk);
    rst = 0;

    // Add stimulus here
    @(posedge clk);
    HLEN_Reg_write_en = 1;
    HLEN_in = 64'd24;

    @(posedge clk);
    HLEN_Reg_write_en = 0;
    HLEN_in = 64'd2;

    @(posedge clk);
    HLEN_Reg_write_en = 1;
    HLEN_in = 64'd3;

    @(posedge clk);
    HLEN_Reg_write_en = 0;
    HLEN_in = 64'd31;

    @(posedge clk);

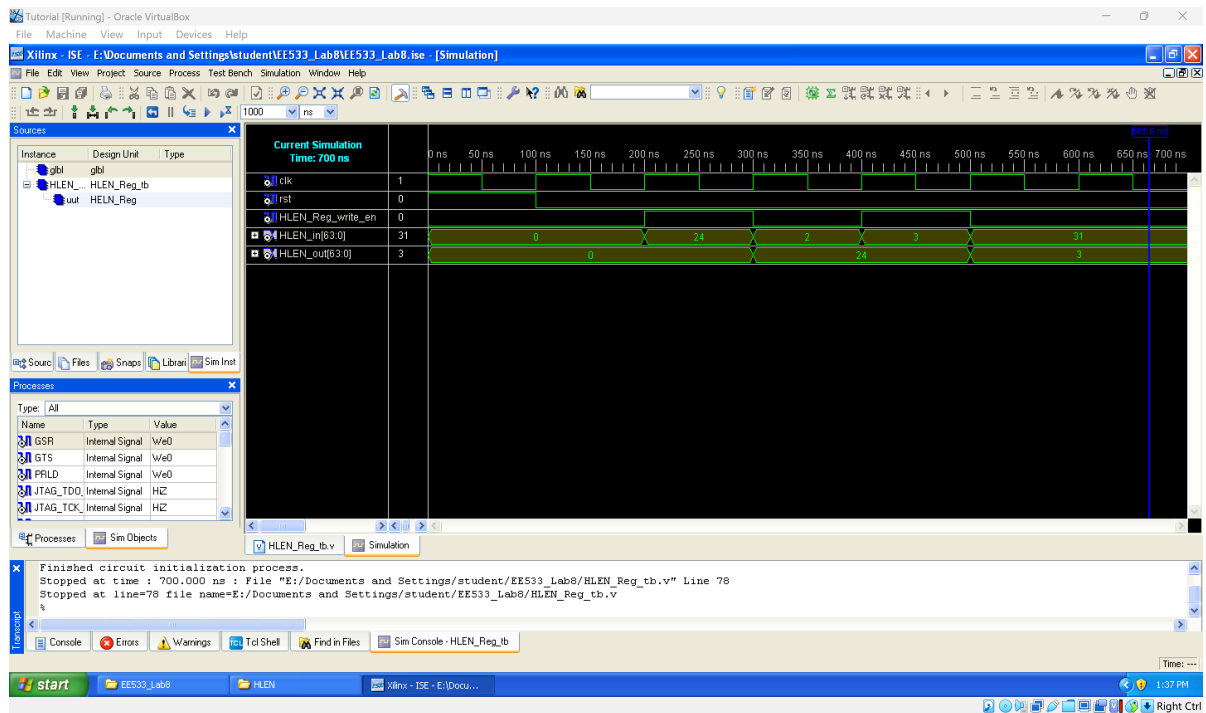
    @(posedge clk);
    $stop;

end

endmodule

```

- Waveform



3.4.2 HLEN_Offset_Adder

- Verilog

```
`timescale 1ns / 1ps

module HLEN_Offset_Adder
(
    input [63:0] offset,
    input [63:0] HLEN,

    output [63:0] result
);

    assign result = offset + HLEN;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:34:57 03/08/2025
// Design Name:    HLEN_Offset_Adder
// Module Name:    E:/Documents and
Settings/student/EE533_Lab8/HLEN_Offset_Adder_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
```

```

// Verilog Test Fixture created by ISE for module: HLEN_Offset_Adder
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module HLEN_Offset_Adder_tb;

    // Inputs
    reg [63:0] Offset;
    reg [63:0] HLEN;

    // Outputs
    wire [63:0] result;

    // Instantiate the Unit Under Test (UUT)
    HLEN_Offset_Adder uut (
        .Offset(Offset),
        .HLEN(HLEN),
        .result(result)
    );

    initial begin
        // Initialize Inputs
        Offset = 0;
        HLEN = 0;

        // wait 100 ns for global reset to finish
        #100;

        // Add stimulus here
        #100;
        Offset = 64'd1;
        HLEN = 64'd3;

        #100;
        Offset = 64'd7;
        HLEN = 64'd3;

        #100;
        Offset = 64'd9;
        HLEN = 64'd3;

        #100;
        Offset = 64'd23;
        HLEN = 64'd3;

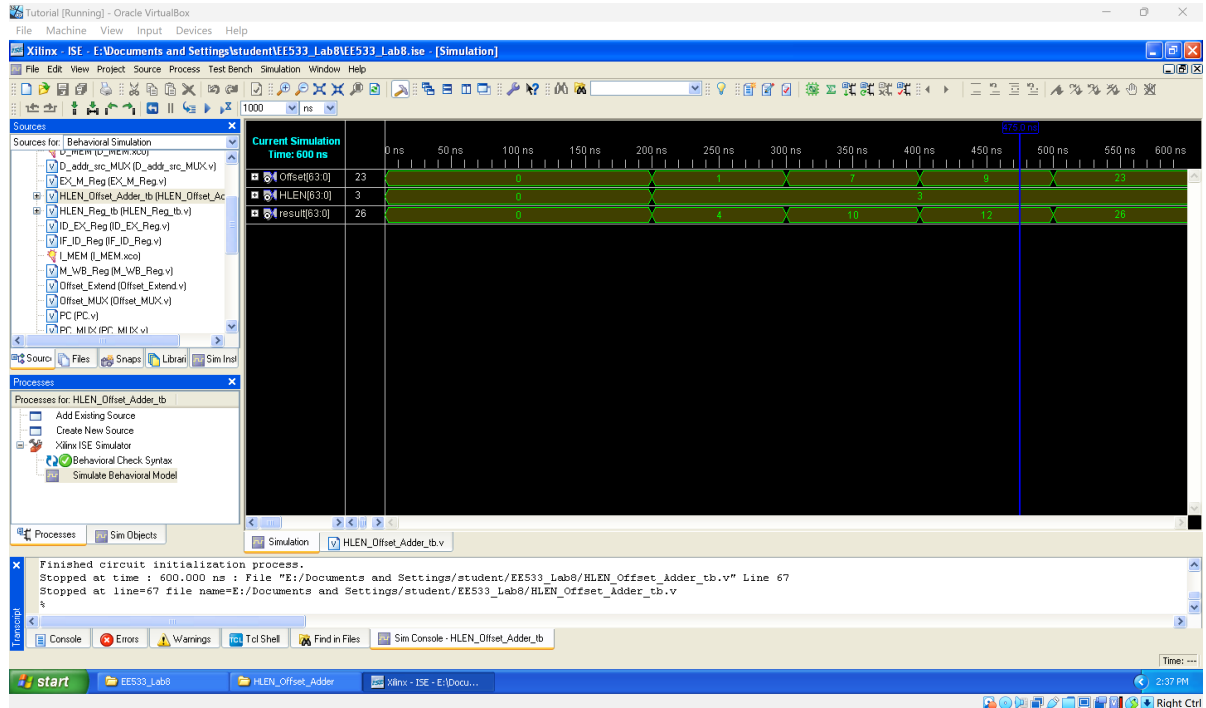
        #100;
        $stop;

    end

```

endmodule

- Waveform



3.4.3 Offset_MUX

- Verilog

```
`timescale 1ns / 1ps

module offset_MUX
(
    input LW_EX,
    input SW_EX,

    input [63:0] HLEN_Offset_Adder_result,
    input [63:0] offset_EX,

    output [63:0] ALU_src_MUX_in
);

    wire offset_MUX_ctrl;

    assign offset_MUX_ctrl = LW_EX || SW_EX;

    assign ALU_src_MUX_in = offset_MUX_ctrl ? HLEN_Offset_Adder_result :
offset_EX;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////
```

```

// Company:
// Engineer:
//
// Create Date:    14:39:47 03/08/2025
// Design Name:    Offset_MUX
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/Offset_MUX_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: Offset_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////

module Offset_MUX_tb;

    // Inputs
    reg LW_EX;
    reg SW_EX;
    reg [63:0] HLEN_Offset_Adder_result;
    reg [63:0] Offset_EX;

    // Outputs
    wire [63:0] ALU_src_MUX_in;

    // Instantiate the Unit Under Test (UUT)
    Offset_MUX uut (
        .LW_EX(LW_EX),
        .SW_EX(SW_EX),
        .HLEN_Offset_Adder_result(HLEN_Offset_Adder_result),
        .Offset_EX(Offset_EX),
        .ALU_src_MUX_in(ALU_src_MUX_in)
    );

    initial begin
        // Initialize Inputs
        LW_EX = 0;
        SW_EX = 0;
        HLEN_Offset_Adder_result = 0;
        Offset_EX = 0;

        // wait 100 ns for global reset to finish
        #100;
        LW_EX = 0;
        SW_EX = 0;
        HLEN_Offset_Adder_result = 64'd2;
        Offset_EX = 64'd7;

        // Add stimulus here

```

```

#100;

LW_EX = 0;
SW_EX = 1;
HLEN_Offset_Adder_result = 64'd3;
offset_EX = 64'd8;

#100;

LW_EX = 1;
SW_EX = 0;
HLEN_Offset_Adder_result = 64'd4;
offset_EX = 64'd9;

#100;

LW_EX = 1;
SW_EX = 1;
HLEN_Offset_Adder_result = 64'd5;
offset_EX = 64'd10;

#100;

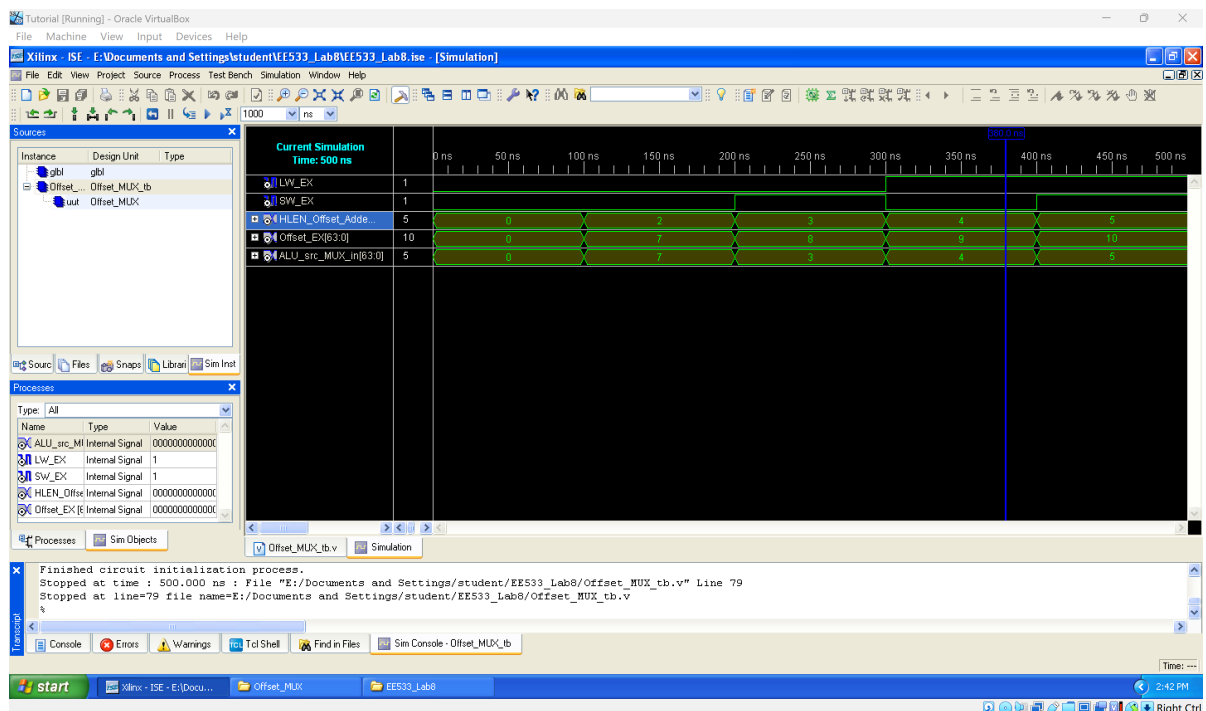
$stop;

```

end

endmodule

- Waveform



3.5 WME_OR

- Verilog

```

`timescale 1ns / 1ps

module WME_OR
(
    input WME_EX,
    input WP_en,

    output WME
);

    assign WME = WME_EX || WP_en;

endmodule

```

- Testbench

```

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:43:27 03/08/2025
// Design Name:    WME_OR
// Module Name:    E:/Documents and Settings/student/EE533_Lab8/WME_OR_tb.v
// Project Name:   EE533_Lab8
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: WME_OR
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module WME_OR_tb;

    // Inputs
    reg WME_EX;
    reg WP_en;

    // Outputs
    wire WME;

    // Instantiate the Unit Under Test (UUT)
    WME_OR uut (
        .WME_EX(WME_EX),
        .WP_en(WP_en),
        .WME(WME)
    );

```

```

);

initial begin
    // Initialize Inputs
    WME_EX = 0;
    WP_en = 0;

    // wait 100 ns for global reset to finish
    #100;
    WME_EX = 0;
    WP_en = 0;

    // Add stimulus here
    #100;
    WME_EX = 0;
    WP_en = 1;

    #100;
    WME_EX = 1;
    WP_en = 0;

    #100;
    WME_EX = 1;
    WP_en = 1;

    #100;
    $stop;

end

endmodule

```

- Waveform

