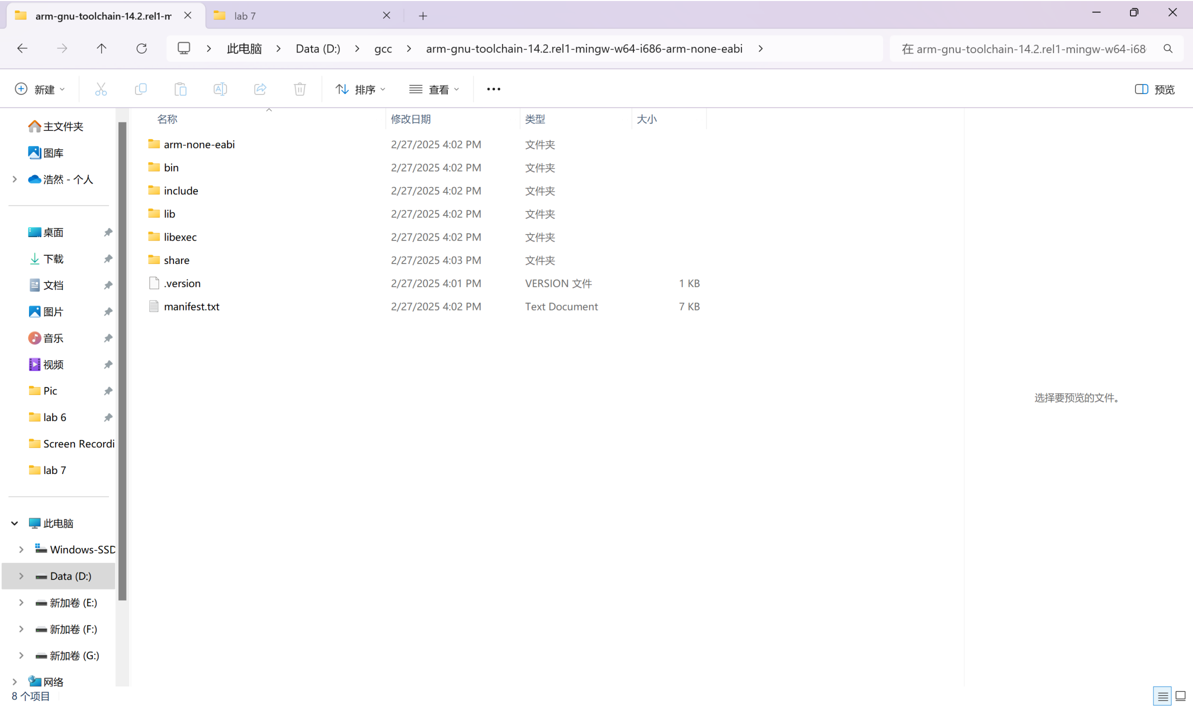
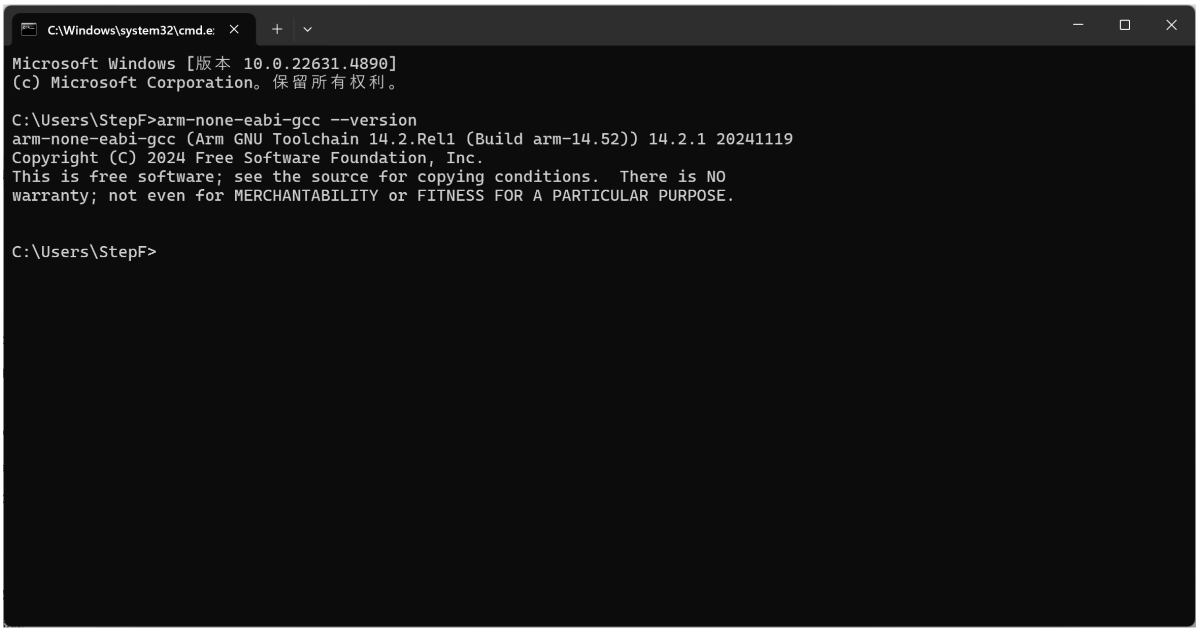
EE533_Lab7_Report

1. Find what Instruction We Need to Implement

1.1 ARM C/C++ Cross-complier Installation

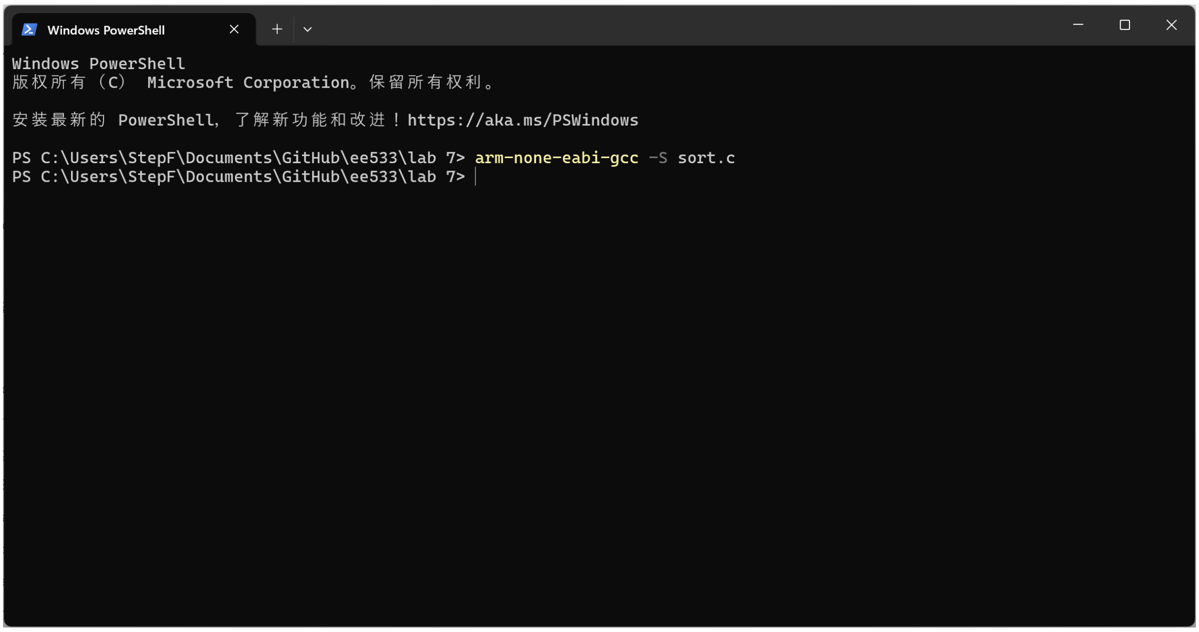


名称	修改日期	类型	大小
arm-none-eabi	2/27/2025 4:02 PM	文件夹	
bin	2/27/2025 4:02 PM	文件夹	
include	2/27/2025 4:02 PM	文件夹	
lib	2/27/2025 4:02 PM	文件夹	
libexec	2/27/2025 4:02 PM	文件夹	
share	2/27/2025 4:03 PM	文件夹	
.version	2/27/2025 4:01 PM	VERSION 文件	1 KB
manifest.txt	2/27/2025 4:02 PM	Text Document	7 KB



```
Microsoft Windows [版本 10.0.22631.4890]
(c) Microsoft Corporation。保留所有权利。

C:\Users\StepF>arm-none-eabi-gcc --version
arm-none-eabi-gcc (Arm GNU Toolchain 14.2.Rel1 (Build arm-14.52)) 14.2.1 20241119
Copyright (C) 2024 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\StepF>
```

1.2 Assembly Code for Bubble Sort



```
Windows PowerShell
版权所有 (C)  Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\Users\StepF\Documents\GitHub\ee533\lab 7> arm-none-eabi-gcc -S sort.c
PS C:\Users\StepF\Documents\GitHub\ee533\lab 7>
```

- sort.c

```
int main() {
    int array[10] = {323, 123, -455, 2, 98, 125, 10, 65, -56, 0};
    int i, j, swap;
    for (i = 0 ; i < 10; i++) {
        for (j = i+1 ; j < 10 ; j++) {
            if (array[j] < array[i]) {
                swap = array[j];
                array[j] = array[i];
                array[i] = swap;
            }
        }
    }
}
```

- sort.s

```
    .cpu arm7tdmi
    .arch armv4t
    .fpu softvfp
    .eabi_attribute 20, 1
    .eabi_attribute 21, 1
    .eabi_attribute 23, 3
    .eabi_attribute 24, 1
    .eabi_attribute 25, 1
    .eabi_attribute 26, 1
    .eabi_attribute 30, 6
    .eabi_attribute 34, 0
    .eabi_attribute 18, 4
    .file   "sort.c"
    .text
    .section    .rodata
    .align  2
.LC0:
    .word   323
    .word   123
    .word   -455
    .word   2
    .word   98
    .word   125
    .word   10
    .word   65
    .word   -56
    .word   0
    .text
    .align  2
    .global main
    .syntax unified
    .arm
    .type   main, %function
main:
    @ Function supports interworking.
    @ args = 0, pretend = 0, frame = 56
    @ frame_needed = 1, uses_anonymous_args = 0
    push    {fp, lr}
    add fp, sp, #4
    sub sp, sp, #56
    ldr r3, .L8
    sub ip, fp, #56
    mov lr, r3
    ldmia   lr!, {r0, r1, r2, r3}
    stmia   ip!, {r0, r1, r2, r3}
    ldmia   lr!, {r0, r1, r2, r3}
    stmia   ip!, {r0, r1, r2, r3}
    ldm lr, {r0, r1}
    stm ip, {r0, r1}
    mov r3, #0
    str r3, [fp, #-8]
    b   .L2
.L6:
    ldr r3, [fp, #-8]
    add r3, r3, #1
    str r3, [fp, #-12]
    b   .L3
.L5:
    ldr r3, [fp, #-12]
    lsl r3, r3, #2
    sub r3, r3, #4
    add r3, r3, fp
    ldr r2, [r3, #-52]
    ldr r3, [fp, #-8]
    lsl r3, r3, #2
    sub r3, r3, #4
    add r3, r3, fp
    ldr r3, [r3, #-52]
    cmp r2, r3
    bge .L4
    ldr r3, [fp, #-12]
    lsl r3, r3, #2
    sub r3, r3, #4
    add r3, r3, fp
    ldr r3, [r3, #-52]
    str r3, [fp, #-16]
    ldr r3, [fp, #-8]
    lsl r3, r3, #2
    sub r3, r3, #4
    add r3, r3, fp
    ldr r2, [r3, #-52]
    ldr r3, [fp, #-12]
    lsl r3, r3, #2
    sub r3, r3, #4
    add r3, r3, fp
    str r2, [r3, #-52]
    ldr r3, [fp, #-8]
    lsl r3, r3, #2
    sub r3, r3, #4
    add r3, r3, fp
    ldr r2, [fp, #-16]
    str r2, [r3, #-52]
.L4:
    ldr r3, [fp, #-12]
    add r3, r3, #1
    str r3, [fp, #-12]
.L3:
    ldr r3, [fp, #-12]
    cmp r3, #9
    ble .L5
    ldr r3, [fp, #-8]
    add r3, r3, #1
    str r3, [fp, #-8]
.L2:
    ldr r3, [fp, #-8]
    cmp r3, #9
    ble .L6
    mov r3, #0
    mov r0, r3
    sub sp, fp, #4
    @ sp needed
    pop {fp, lr}
    bx  lr
.L9:
    .align  2
.L8:
    .word   .LC0
    .size   main, .-main
    .ident  "GCC: (Arm GNU Toolchain 14.2.Rel1 (Build arm-14.52)) 14.2.1 20241119"
```

- Included Instructions' Description and Reference

Instruction	Explaniation	Reference
add	Add	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/general-data-processing-instructions/add--adc--sub--sbc--and-rsb
sub	Subtract	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/general-data-processing-instructions/add--adc--sub--sbc--and-rsb
mov	Move	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/general-data-processing-instructions/mov-and-mvn
ldm	Load Multiple registers, increment after	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/ldm-and-stm
ldmia	LDMIA is a synonym for LDM	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/ldm-and-stm
stm	Store Multiple registers, increment after	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/ldm-and-stm
stmia	STMIA is a synonym for STM	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/ldm-and-stm
ldr	Load Register with word	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/ldr-and-str--register-offset
str	Store Register word	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/ldr-and-str--register-offset
lsl	Logical Shift Left	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/general-data-processing-instructions/asr--lsl--lsr--ror--and-rrx?lang=en
cmp	Compare	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/general-data-processing-instructions/cmp-and-cmn
push	Push registers onto stack	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/push-and-pop
pop	Pop registers from stack	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/memory-access-instructions/push-and-pop

Instruction	Explaniation	Reference
b	Branch	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/branch-and-control-instructions/b--bl--bx--and-blx
bge	Branch if greater than or equal	https://developer.arm.com/documentation/100076/0100/A64-Instruction-Set-Reference/A64-General-Instructions/B-cond/Condition-code-suffixes-and-related-flags
ble	Branch if less or equal	https://developer.arm.com/documentation/100076/0100/A64-Instruction-Set-Reference/A64-General-Instructions/B-cond/Condition-code-suffixes-and-related-flags
bx	Indirect branch	https://developer.arm.com/documentation/dui0552/a/the-cortex-m3-instruction-set/branch-and-control-instructions/b--bl--bx--and-blx

1.3 List of instruction needs to be implemented

1.3.1 Instruction Format Definition

- Instruction Format definition

OP_CODE	Rs	Rt	Rd	Shift	Funct
I_MEM[31:26]	I_MEM[25:21]	I_MEM[20:16]	I_MEM[15:11]	I_MEM[10:6]	I_MEM[5:0]

- OP_Code look up table

Instruction	OP_Code	Description
add	000000	Add
sub	000001	Subtract
mov	000010	Move
ldr	000011	Load Register with word
str	000100	Store Register word
ldm	000101	Load Multiple registers, increment after
stm	000110	Store Multiple registers, increment after
ldmia	000111	LDMIA is a synonym for LDM
stmia	001000	STMIA is a synonym for STM
lsl	001001	Logical Shift Left
cmp	001010	Compare

Instruction	OP_Code	Description
push	001011	Push registers onto stack
pop	001100	Pop registers from stack
b	001101	Branch
bge	001110	Branch if greater than or equal
ble	001111	Branch if less or equal
bx	010000	Indirect branchs

1.3.2 Special Registers in ARM ISA

Register Name	Actual Register Location	Description
FP	R11	Frame Pointer
IP	R12	Intra Procedural Call
SP	R13	Stack Pointer
LR	R14	Link Register

2. Pipelined Processor on NetFPGA

2.1 ARM Instruction Generated

```
.data
array:  .dword  323, 123, -455, 2, 98, 125, 10, 65, -56, 0
N:      .dword  10

.text
.global _start
_start:
    ldr r4, =array       @ r4 = base address of array
    ldr r5, =N           @ r5 = address of N
    ldr r5, [r5]         @ r5 = N (size of array)
    sub r5, r5, #1       @ r5 = N-1 (outer loop limit)

outer_loop:
    mov r6, #0           @ i = 0

inner_loop:
    sub r7, r6, r5       @ if i >= N-1, exit inner loop
    bge outer_continue

    @ Load array[i] and array[i+1]
    mov r8, r6, LSL #2   @ r8 = i * 4 (word offset)
    add r9, r4, r8       @ r9 = address of array[i]
    ldr r10, [r9]        @ r10 = array[i]
    ldr r11, [r9, #4]    @ r11 = array[i+1]
    sub r12, r10, r11    @ r12 = array[i] - array[i+1]
    ble no_swap          @ If array[i] <= array[i+1], no swap

    str r11, [r9]        @ array[i] = array[i+1]
    str r10, [r9, #4]    @ array[i+1] = array[i]

no_swap:
    add r6, r6, #1       @ i++
    b inner_loop

outer_continue:
    sub r5, r5, #1       @ Reduce loop limit (N-1, N-2, ...)
    bgt outer_loop       @ If still positive, loop again

end:
    b end                @ Infinite loop (halt)
```

- Assembly Code

```
#0  ldr r4, =array
#1  ldr r5, =N
#2  ldr r5, [r5]
#3  sub r5, r5, #1       @ outer_continue
#4  mov r6, #0           @ outer_loop
#5  mov r8, r6
#6  lsl r8, r8, #2
#7  add r9, r4, r8
#8  ldr r10, [r9]
#9  ldr r11, [r9, #8]
#10 sub r12, r10, r11
#11 ble no_swap
#12 str r11, [r9]
#13 str r10, [r9, #8]
#14 add r6, r6, #1        @ no_swap
#15 sub r7, r6, r5        @ inner_loop
#16 bge outer_continue
#17 b inner_loop
#18 b end                 @ end
```

2.2 MIPS Instruction Format

2.2.1 Pesudo Instruction

```
lw r0, array_addr
lw r1, array_size
subi r1, r1, #1
mov r2, #0
outer_loop:
beq r2, r1, end
addi r3, r2, #1
inner_loop:
bgt r3, r1, next_out
lw r4, r0(r2)
lw r5, r0(r3)
```

```
blt r4, r5, no_swap
sw r4, r0(r3)
sw r5, r0(r2)
no_swap:
addi r3, r3, #1
j inner_loop
next_out:
addi r2, r2, #1
j outer_loop
end:
j end
```

2.2.2 Real MIPS Instruction

- With manually introduced NOOP, we can avoid data dependency problem and early branch flush problem.

```
movi r1, #9
outer_loop:
noop
noop
beq r2, r1, end
noop
addi r3, r2, #1
inner_loop:
noop
noop
bgt r3, r1, next_out
noop
lw r4, r2(#0)
lw r5, r3(#0)
noop
noop
blt r4, r5, no_swap
noop
sw r4, r3(#0)
sw r5, r2(#0)
no_swap:
addi r3, r3, #1
j inner_loop
noop
next_out:
addi r2, r2, #1
j outer_loop
noop
end:
j end
```

- Instruction OP Code

Instr	OP Code [31:26]
noop	000000

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

- Instruction Table

Addr	Label	Instr	OP Code [31:26]	Rs [25:21]	Rt [20:16]	Offset [15:0]
0		movi r1, #9	000010	5'd0	5'd1	16'd9
1	outer_loop	noop	000000	5'd0	5'd0	16'd0
2		noop	000000	5'd0	5'd0	16'd0
3		beq r2, r1, end	000101	5'd1	5'd2	16'd24
4		noop	000000	5'd0	5'd0	16'd0
5		addi r3, r2, #1	000001	5'd2	5'd3	16'd1
6	inner_loop	noop	000000	5'd0	5'd0	16'd0
7		noop	000000	5'd0	5'd0	16'd0
8		bgt r3, r1, next_out	000110	5'd1	5'd3	16'd21
9		noop	000000	5'd0	5'd0	16'd0
10		lw r4, r2(#0)	000011	5'd2	5'd4	16'd0
11		lw r5, r3(#0)	000011	5'd3	5'd5	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

Addr	Label	Instr	OP Code [31:26]	Rs [25:21]	Rt [20:16]	Offset [15:0]
16		sw r4, r3(#0)	000100	5'd3	5'd4	16'd0
17		sw r5, r2(#0)	000100	5'd2	5'd5	16'd0
18	no_swap	addi r3, r3, #1	000001	5'd3	5'd3	16'd1
19		j inner_loop	001000	5'd0	5'd0	16'd6
20		noop	000000	5'd0	5'd0	16'd0
21	next_out	addi r2, r2, #1	000001	5'd2	5'd2	16'd1
22		j outer_loop	001000	5'd0	5'd0	16'd1
23		noop	000000	5'd0	5'd0	16'd0
24	end	j end	001000	5'd0	5'd0	16'd24

3. 5-Stage Pipeline Elements

3.1 IF Stage

3.1.1 PC

- Verilog

```
`timescale 1ns / 1ps

module PC
(
    input clk,
    input rst,
    input [63:0] PC_next,

    output reg [63:0] PC
);

    always @(posedge clk) begin
        if (rst)
            PC <= 64'b0;
        else
            PC <= PC_next;
    end

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

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

```
// Engineer:
//
// Create Date:   15:17:32 03/01/2025
// Design Name:   PC
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/PC_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: PC
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module PC_tb;

    // Inputs
    reg clk;
    reg rst;
    reg [63:0] PC_next;

    // Outputs
    wire [63:0] PC;

    // Instantiate the Unit Under Test (UUT)
    PC uut (
        .clk(clk),
        .rst(rst),
        .PC_next(PC_next),
        .PC(PC)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 1;
        rst = 1;
        PC_next = 0;

        // Wait 100 ns for global reset to finish
        #100;
        rst = 0;

        // Add stimulus here
        @(posedge clk);
        PC_next = 64'd1;

        @(posedge clk);
        PC_next = 64'd2;
        @(posedge clk);
        PC_next = 64'd3;

        @(posedge clk);
        PC_next = 64'd4;

        @(posedge clk);
        $stop;

    end

endmodule
```

- Waveform

屏幕截图 2025-03-01 152033

3.1.2 PC+1

- Verilog

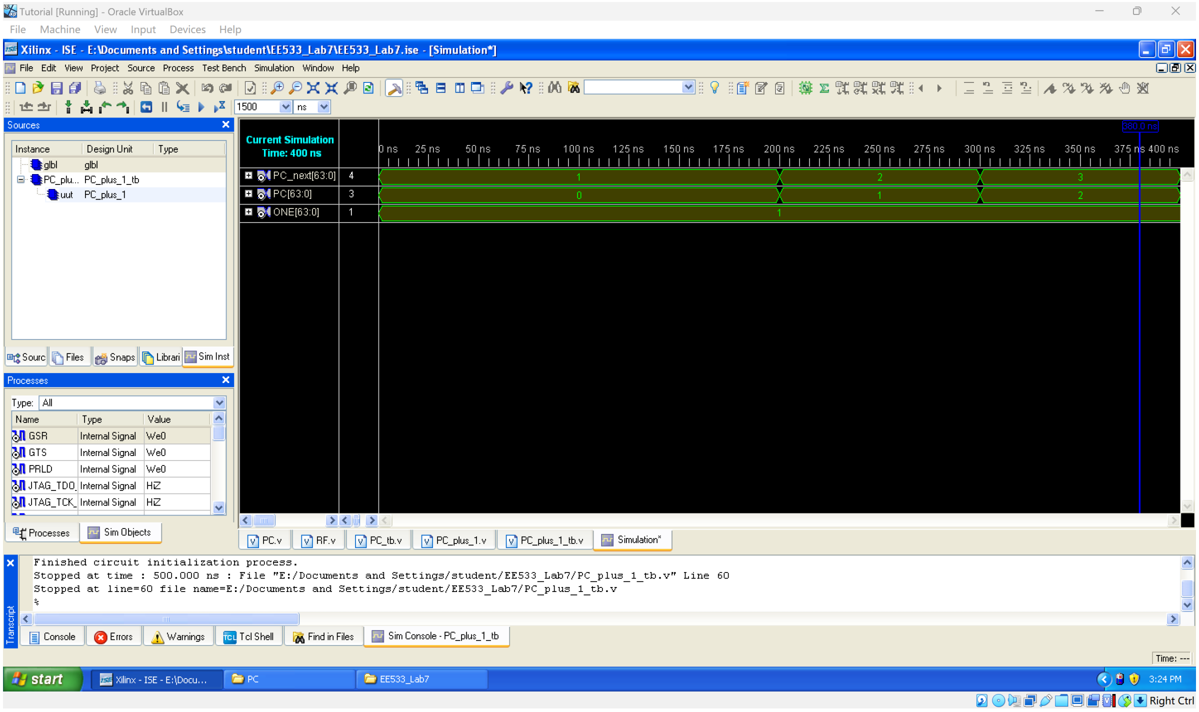
```
`timescale 1ns / 1ps

module PC_plus_1
(
    input [63:0] PC,
    input [63:0] ONE,

    output [63:0] PC_next
);

    assign PC_next = PC + ONE;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   15:22:37 03/01/2025
// Design Name:   PC_plus_1
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/PC_plus_1_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: PC_plus_1
//
// Dependencies:
```

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

module PC_plus_1_tb;

    // Inputs
    reg [63:0] PC;
    reg [63:0] ONE;

    // Outputs
    wire [63:0] PC_next;

    // Instantiate the Unit Under Test (UUT)
    PC_plus_1 uut (
        .PC(PC),
        .ONE(ONE),
        .PC_next(PC_next)
    );

    initial begin
        // Initialize Inputs
        PC = 0;
        ONE = 1;

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

        // Add stimulus here
        #100;
        PC = 1;

        #100;
        PC = 2;

        #100;
        PC = 3;

        #100;
        $stop;

    end

endmodule
```

- Waveform


Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation*]
Current Simulation
Time: 400 ns
PC_next[63:0] 4
PC[63:0] 3
ONE[63:0] 1
Sim Console - PC_plus_1_tb

```
Finished circuit initialization process.
Stopped at time : 500.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/PC_plus_1_tb.v" Line 60
Stopped at line=60 file name=E:/Documents and Settings/student/EE533_Lab7/PC_plus_1_tb.v
%
```

3.1.3 PC_MUX

- Verilog

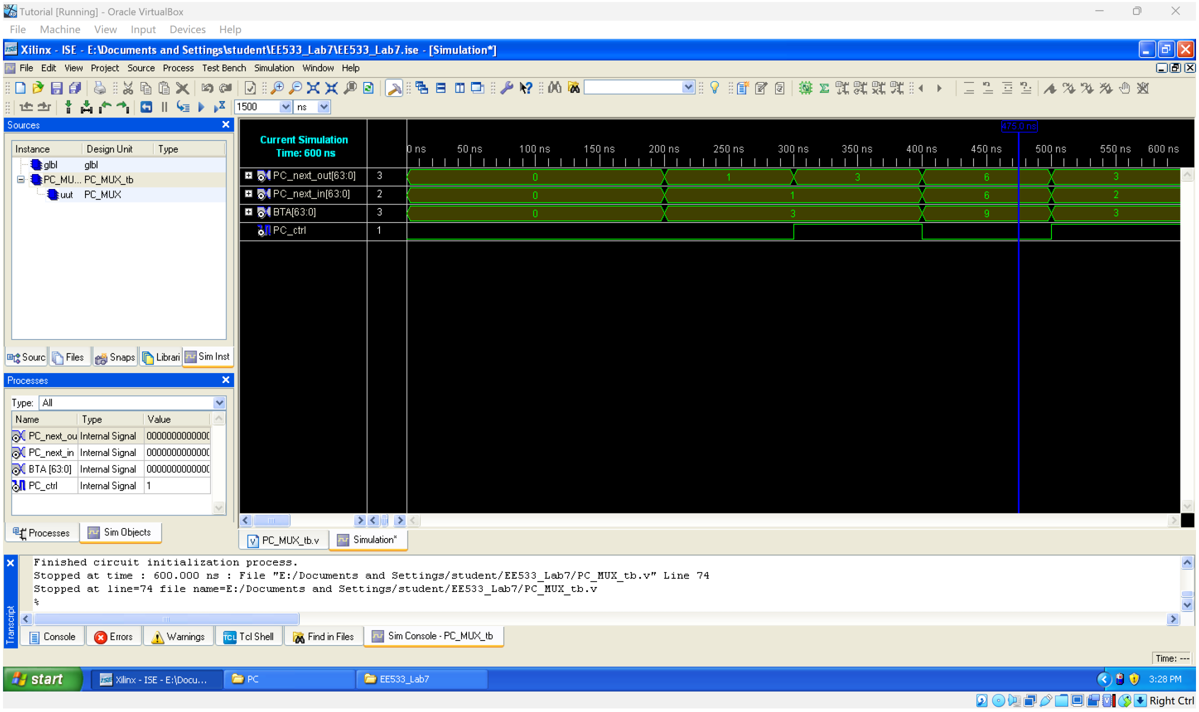
```
`timescale 1ns / 1ps

module PC_MUX
(
    input [63:0] PC_next_in,
    input [63:0] BTA,
    input PC_ctrl,

    output [63:0] PC_next_out
);

    assign PC_next_out = PC_ctrl == 1? BTA : PC_next_in;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   15:26:36 03/01/2025
// Design Name:   PC_MUX
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/PC_MUX_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
```

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

module PC_MUX_tb;

    // Inputs
    reg [63:0] PC_next_in;
    reg [63:0] BTA;
    reg PC_ctrl;

    // Outputs
    wire [63:0] PC_next_out;

    // Instantiate the Unit Under Test (UUT)
    PC_MUX uut (
        .PC_next_in(PC_next_in),
        .BTA(BTA),
        .PC_ctrl(PC_ctrl),
        .PC_next_out(PC_next_out)
    );

    initial begin
        // Initialize Inputs
        PC_next_in = 0;
        BTA = 0;
        PC_ctrl = 0;

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

        // Add stimulus here
        #100;
        PC_next_in = 1;
        BTA = 3;
        PC_ctrl = 0;

        #100;
        PC_next_in = 1;
        BTA = 3;
        PC_ctrl = 1;

        #100;
        PC_next_in = 6;
        BTA = 9;
        PC_ctrl = 0;

        #100;
        PC_next_in = 2;
        BTA = 3;
        PC_ctrl = 1;

        #100;
        $stop;

    end

endmodule
```

- Waveform



Tutorial [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation*]
File Edit View Project Source Process Test Bench Simulation Window Help
1500 ns
Sources
Instance Design Unit Type
glbl glbl
PC_MU... PC_MUX_tb
uut PC_MUX
Sourc Files Snaps Librari Sim Inst
Processes
Type: All
Processes Sim Objects
Current Simulation Time: 600 ns
PC_next_out[63:0] 3
PC_next_in[63:0] 2
BTA[63:0] 3
PC_ctrl 1
0 ns 50 ns 100 ns 150 ns 200 ns 250 ns 300 ns 350 ns 400 ns 450 ns 500 ns 550 ns 600 ns
0 1 3 6 3
0 1 6 2
0 3 9 3
PC_MUX_tb.v Simulation*
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - PC_MUX_tb
Time: ---
start Xilinx - ISE - E:\Docu... PC EE533_Lab7
3:28 PM
Right Ctrl

Name	Type	Value
PC_next_ou	Internal Signal	000000000000(
PC_next_in	Internal Signal	000000000000(
BTA [63:0]	Internal Signal	000000000000(
PC_ctrl	Internal Signal	1

```
Finished circuit initialization process.
Stopped at time : 600.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/PC_MUX_tb.v" Line 74
Stopped at line=74 file name=E:/Documents and Settings/student/EE533_Lab7/PC_MUX_tb.v
%
```

3.1.4 I_MEM

- Verilog

- Vector File

- Memory Initialization File

- Testbench

- Waveform

3.2 ID Stage

3.2.1 RegFIle

- Verilog

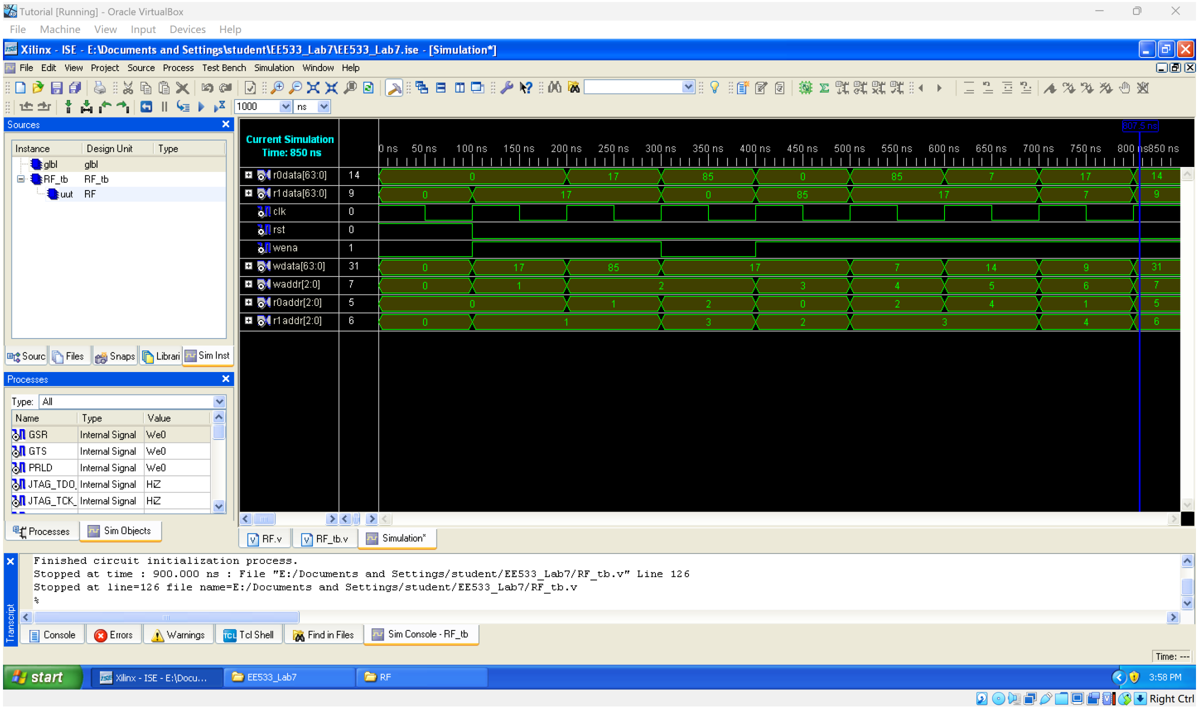
```verilog
`timescale 1ns / 1ps

module RF
(
    input clk,
    input rst,
    input wena,
    input [63:0] wdata,
    input [2:0] waddr,
    input [2:0] r0addr,
    input [2:0] r1addr,

    output reg [63:0] r0data,
    output reg [63:0] r1data
);

    reg [63:0] RF [7:0];

    integer i;

    always @(posedge clk) begin
        if (rst == 1)
        begin
            for (i = 0; i < 8; i = i + 1) begin
                RF[i] <= 64'b0;
            end
        end
        else if (wena == 1)
        begin
            RF[waddr] <= wdata;
        end
    end

    always @(*) begin
        r0data = ((waddr == r0addr) && wena) ? RF[waddr] : RF[r0addr];
        r1data = ((waddr == r1addr) && wena) ? RF[waddr] : RF[r1addr];
    end


endmodule
```

- Testbench

```verilog
`timescale 1ns / 1ps

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

```
//
// Create Date:   15:53:18 03/01/2025
// Design Name:   RF
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/RF_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: RF
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module RF_tb;

    // Inputs
    reg clk;
    reg rst;
    reg wena;
    reg [63:0] wdata;
    reg [2:0] waddr;
    reg [2:0] r0addr;
    reg [2:0] r1addr;

    // Outputs
    wire [63:0] r0data;
    wire [63:0] r1data;

    // Instantiate the Unit Under Test (UUT)
    RF uut (
        .clk(clk),
        .rst(rst),
        .wena(wena),
        .wdata(wdata),
        .waddr(waddr),
        .r0addr(r0addr),
        .r1addr(r1addr),
        .r0data(r0data),
        .r1data(r1data)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 1;
        rst = 1;
        wena = 0;
        wdata = 0;
        waddr = 0;
        r0addr = 0;
        r1addr = 0;

        // Wait 100 ns for global reset to finish
        #100;
        rst = 0;

        // Add stimulus here
        wena = 1;
        waddr = 3'd1;
        wdata = 64'd17;
        r0addr = 3'b000;
        r1addr = 3'b001;
        #100;

        wena = 1;
        waddr = 3'd2;
        wdata = 64'd85;
        r0addr = 3'd1;
        r1addr = 3'd1;
        #100;

        wena = 0;
        waddr = 3'd2;
        wdata = 64'd17;
        r0addr = 3'd2;
        r1addr = 3'd3;
        #100;

        wena = 1;
        waddr = 3'd3;
        wdata = 64'd17;
        r0addr = 3'd0;
        r1addr = 3'd2;
        #100;

        wena = 1;
        waddr = 3'd4;
        wdata = 64'd7;
        r0addr = 3'd2;
        r1addr = 3'd3;
        #100;

        wena = 1;
        waddr = 3'd5;
        wdata = 64'd14;
        r0addr = 3'd4;
        r1addr = 3'd3;
        #100;

        wena = 1;
        waddr = 3'd6;
        wdata = 64'd9;
        r0addr = 3'd1;
        r1addr = 3'd4;
        #100;
        wena = 1;
        waddr = 3'd7;
        wdata = 64'd31;
        r0addr = 3'd5;
        r1addr = 3'd6;
        #100;

        $stop;
    end

endmodule
```

- Waveform



```
Finished circuit initialization process.
Stopped at time : 900.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/RF_tb.v" Line 126
Stopped at line=126 file name=E:/Documents and Settings/student/EE533_Lab7/RF_tb.v
%
```

3.2.2 Control_Unit

- Verilog

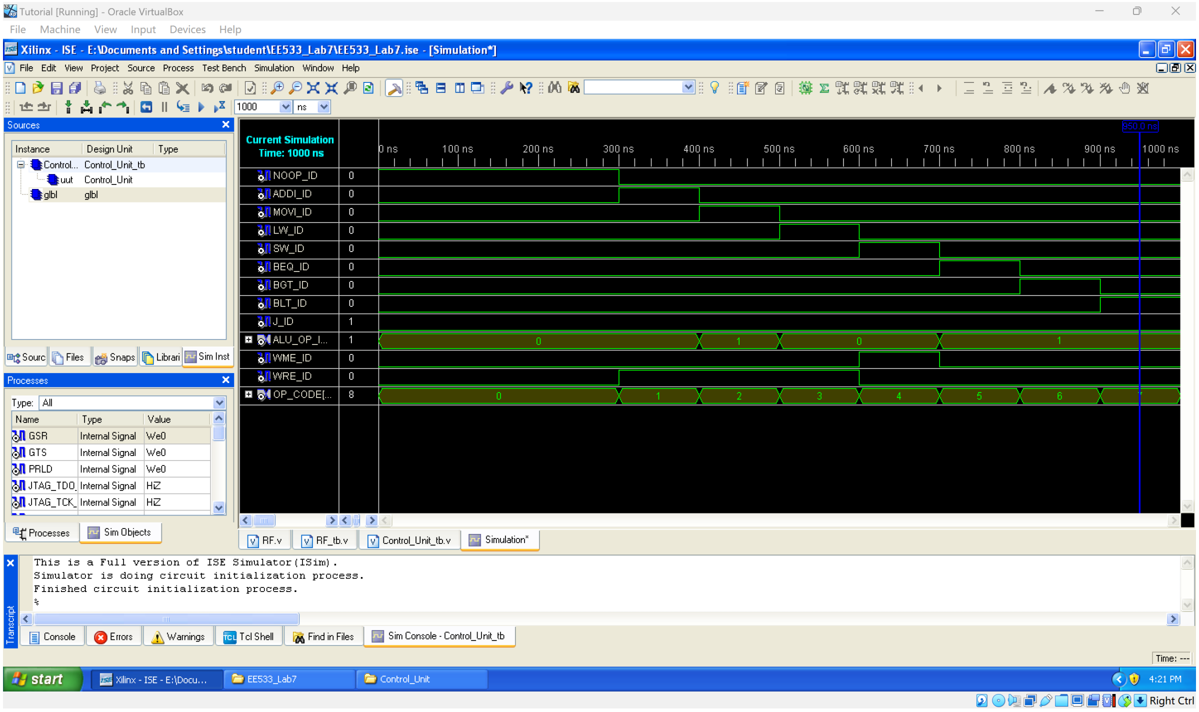
```
`timescale 1ns / 1ps

module Control_Unit
(
    input [5:0] OP_CODE,

    output NOOP_ID,
    output ADDI_ID,
    output MOVI_ID,
    output LW_ID,
    output SW_ID,
    output BEQ_ID,
    output BGT_ID,
    output BLT_ID,
    output J_ID,

    output [3:0] ALU_OP_ID,
    output WME_ID,
    output WRE_ID
);

    assign NOOP_ID = (OP_CODE == 6'd0) ? 1 : 0;
    assign ADDI_ID = (OP_CODE == 6'd1) ? 1 : 0;
    assign MOVI_ID = (OP_CODE == 6'd2) ? 1 : 0;
    assign LW_ID = (OP_CODE == 6'd3) ? 1 : 0;
    assign SW_ID = (OP_CODE == 6'd4) ? 1 : 0;
    assign BEQ_ID = (OP_CODE == 6'd5) ? 1 : 0;
    assign BGT_ID = (OP_CODE == 6'd6) ? 1 : 0;
    assign BLT_ID = (OP_CODE == 6'd7) ? 1 : 0;
    assign J_ID = (OP_CODE == 6'd8) ? 1 : 0;

    assign ALU_OP_ID = ((OP_CODE == 6'd0) || (OP_CODE == 6'd1) || (OP_CODE == 6'd3) || (OP_CODE == 6'd4)) ? 4'd0 : 4'd1;
    assign WME_ID = (OP_CODE == 6'd4) ? 1 : 0;
    assign WRE_ID = ((OP_CODE == 6'd1) || (OP_CODE == 6'd2) || (OP_CODE == 6'd3)) ? 1 : 0;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   16:16:45 03/01/2025
// Design Name:   Control_Unit
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/Control_Unit_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: Control_Unit
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module Control_Unit_tb;

    // Inputs
    reg [5:0] OP_CODE;

    // Outputs
    wire NOOP_ID;
```

```verilog
    wire ADDI_ID;
    wire MOVI_ID;
    wire LW_ID;
    wire SW_ID;
    wire BEQ_ID;
    wire BGT_ID;
    wire BLT_ID;
    wire J_ID;
    wire [3:0] ALU_OP_ID;
    wire WME_ID;
    wire WRE_ID;

    // Instantiate the Unit Under Test (UUT)
    Control_Unit uut (
        .OP_CODE(OP_CODE),
        .NOOP_ID(NOOP_ID),
        .ADDI_ID(ADDI_ID),
        .MOVI_ID(MOVI_ID),
        .LW_ID(LW_ID),
        .SW_ID(SW_ID),
        .BEQ_ID(BEQ_ID),
        .BGT_ID(BGT_ID),
        .BLT_ID(BLT_ID),
        .J_ID(J_ID),
        .ALU_OP_ID(ALU_OP_ID),
        .WME_ID(WME_ID),
        .WRE_ID(WRE_ID)
    );

    initial begin
        // Initialize Inputs
        OP_CODE = 0;

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

        // Add stimulus here
        #100;
        OP_CODE = 0;

        #100;
        OP_CODE = 1;

        #100;
        OP_CODE = 2;

        #100;
        OP_CODE = 3;

        #100;
        OP_CODE = 4;

        #100;
        OP_CODE = 5;

        #100;
        OP_CODE = 6;

        #100;
        OP_CODE = 7;

        #100;
        OP_CODE = 8;

        #100;
        $stop;

    end

endmodule
```

- Waveform



Tutorial [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation*]
File Edit View Project Source Process Test Bench Simulation Window Help
1000 ns
Sources
Sourc Files Snaps Librari Sim Inst
Processes
Type: All
Processes Sim Objects
Current Simulation Time: 1000 ns
0 ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns
950.0 ns
RF.v RF_tb.v Control_Unit_tb.v Simulation*
Console Errors Warnings Tcl Shell Find in Files Sim Console - Control_Unit_tb
Transcript
Time: ---
start Xilinx - ISE - E:\Docu... EE533_Lab7 Control_Unit
4:21 PM
Right Ctrl

Instance	Design Unit	Type
Control...	Control_Unit_tb	
uut	Control_Unit	
glbl	glbl	

Name	Type	Value
GSR	Internal Signal	We0
GTS	Internal Signal	We0
PRLD	Internal Signal	We0
JTAG_TDO_	Internal Signal	HiZ
JTAG_TCK_	Internal Signal	HiZ

NOOP_ID	0
ADDI_ID	0
MOVI_ID	0
LW_ID	0
SW_ID	0
BEQ_ID	0
BGT_ID	0
BLT_ID	0
J_ID	1
ALU_OP_I...	1
WME_ID	0
WRE_ID	0
OP_CODE[...	8

```
This is a Full version of ISE Simulator(ISim).
Simulator is doing circuit initialization process.
Finished circuit initialization process.
%
```

3.2.3 Branch_Detection_Unit

- Verilog

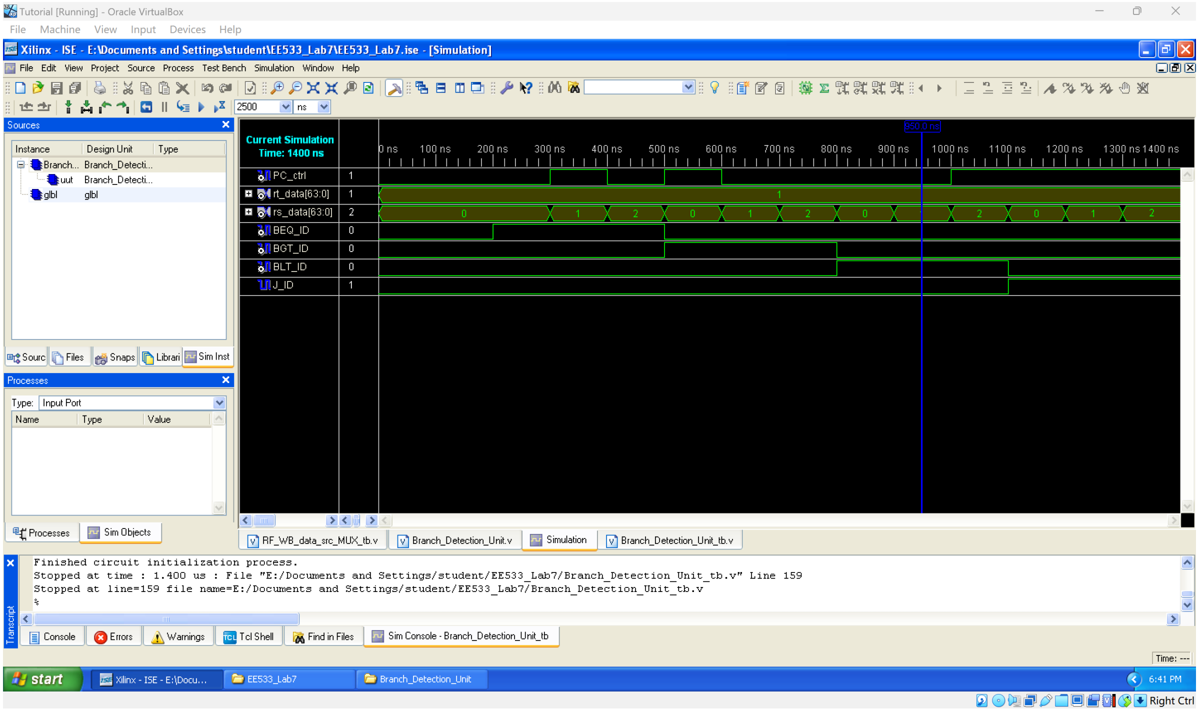
```
`timescale 1ns / 1ps

module Branch_Detection_Unit
(
    input [63:0] rs_data,
    input [63:0] rt_data,

    input BEQ_ID,
    input BGT_ID,
    input BLT_ID,
    input J_ID,

    output PC_ctrl
);
```

```
    assign PC_ctrl =
        ((BEQ_ID == 1) && (rt_data == rs_data)) ||
        ((BGT_ID == 1) && (rt_data > rs_data)) ||
        ((BLT_ID == 1) && (rt_data < rs_data)) ||
        (J_ID == 1)
        ? 1 : 0;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   16:29:58 03/01/2025
// Design Name:   Branch_Detection_Unit
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/Branch_Detection_Unit_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: Branch_Detection_Unit
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module Branch_Detection_Unit_tb;

    // Inputs
    reg [63:0] rs_data;
    reg [63:0] rt_data;
    reg BEQ_ID;
    reg BGT_ID;
    reg BLT_ID;
    reg J_ID;

    // Outputs
    wire PC_ctrl;

    // Instantiate the Unit Under Test (UUT)
    Branch_Detection_Unit uut (
        .rs_data(rs_data),
        .rt_data(rt_data),
        .BEQ_ID(BEQ_ID),
        .BGT_ID(BGT_ID),
        .BLT_ID(BLT_ID),
        .J_ID(J_ID),
        .PC_ctrl(PC_ctrl)
    );

    initial begin
        // Initialize Inputs
        rt_data = 64'd1;
        rs_data = 64'd0;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 0;

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

        // Add stimulus here
        #100;
        rt_data = 64'd1;
        rs_data = 64'd0;
        BEQ_ID = 1;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd1;
        BEQ_ID = 1;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd2;
        BEQ_ID = 1;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd0;
        BEQ_ID = 0;
        BGT_ID = 1;
        BLT_ID = 0;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd1;
        BEQ_ID = 0;
        BGT_ID = 1;
        BLT_ID = 0;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd2;
        BEQ_ID = 0;
        BGT_ID = 1;
        BLT_ID = 0;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd0;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 1;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd1;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 1;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd2;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 1;
        J_ID = 0;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd0;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 1;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd1;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 1;

        #100;
        rt_data = 64'd1;
        rs_data = 64'd2;
        BEQ_ID = 0;
        BGT_ID = 0;
        BLT_ID = 0;
        J_ID = 1;

        #100;
        $stop;

    end

endmodule
```

- Waveform



Tutorial [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
2500 ns
Sources
Instance Design Unit Type
Branch... Branch_Detecti...
uut Branch_Detecti...
glbl glbl
Sourc Files Snaps Librari Sim Inst
Processes
Type: Input Port
Name Type Value
Processes Sim Objects
Current Simulation Time: 1400 ns
950.0 ns
0 ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns 1100 ns 1200 ns 1300 ns 1400 ns
PC_ctrl 1
rt_data[63:0] 1
rs_data[63:0] 2
BEQ_ID 0
BGT_ID 0
BLT_ID 0
J_ID 1
1
0 1 2 0 1 2 0 1 2 0 1 2
RF_WB_data_src_MUX_tb.v Branch_Detection_Unit.v Simulation Branch_Detection_Unit_tb.v
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - Branch_Detection_Unit_tb
Time: ---
start Xilinx - ISE - E:\Docu... EE533_Lab7 Branch_Detection_Unit
6:41 PM
Right Ctrl

```
Finished circuit initialization process.
Stopped at time : 1.400 us : File "E:/Documents and Settings/student/EE533_Lab7/Branch_Detection_Unit_tb.v" Line 159
Stopped at line=159 file name=E:/Documents and Settings/student/EE533_Lab7/Branch_Detection_Unit_tb.v
%
```

3.2.4 Offset_Extend

- Verilog

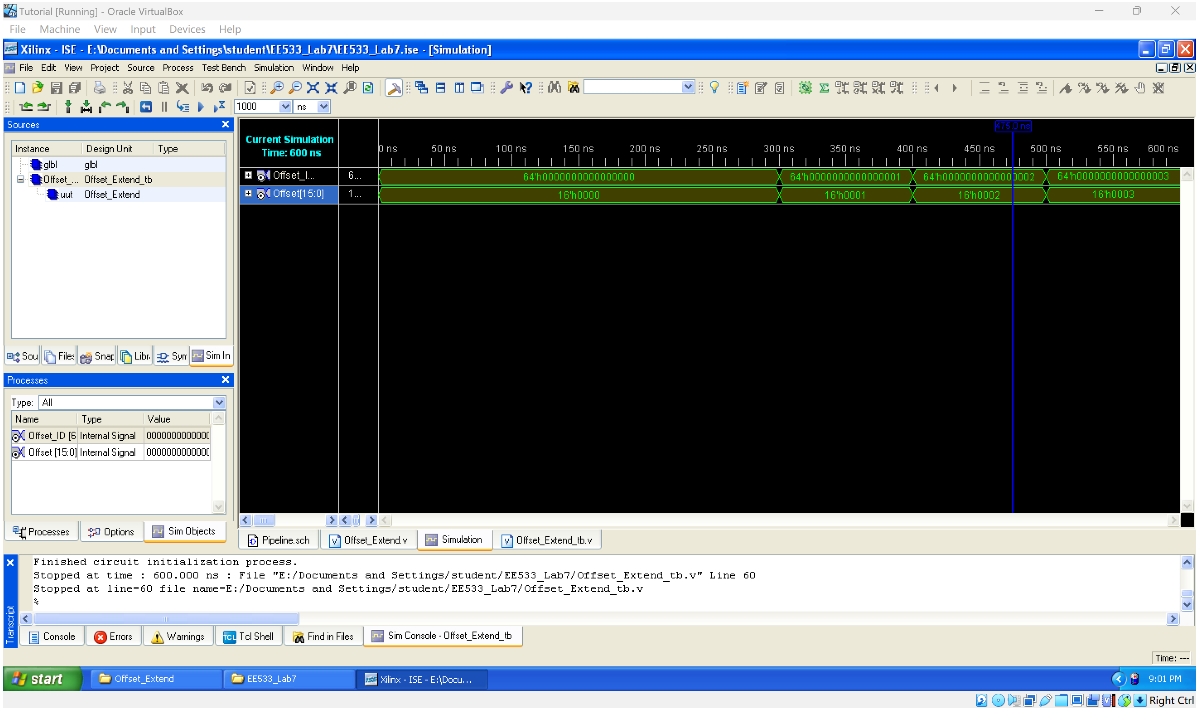
```
`timescale 1ns / 1ps

module Offset_Extend
(
    input [15:0] Offset,

    output [63:0] Offset_ID
);

    assign Offset_ID[15:0]  = Offset;
    assign Offset_ID[63:16]  = 48'b0;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

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

```
// Engineer:
//
// Create Date:   16:40:50 03/01/2025
// Design Name:   Offset_Extend
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/Offset_Extend_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: Offset_Extend
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module Offset_Extend_tb;

    // Inputs
    reg [15:0] Offset;

    // Outputs
    wire [63:0] Offset_ID;

    // Instantiate the Unit Under Test (UUT)
    Offset_Extend uut (
        .Offset(Offset),
        .Offset_ID(Offset_ID)
    );

    initial begin
        // Initialize Inputs
        Offset = 0;

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

        // Add stimulus here
        #100;
        Offset = 16'd0;

        #100;
        Offset = 16'd1;

        #100;
        Offset = 16'd2;

        #100;
        Offset = 16'd3;

        #100;
        $stop;
    end

endmodule
```

- Waveform



```
Finished circuit initialization process.
Stopped at time : 600.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/Offset_Extend_tb.v" Line 60
Stopped at line=60 file name=E:/Documents and Settings/student/EE533_Lab7/Offset_Extend_tb.v
%
```

3.3 EX Stage

3.3.1 ALU

- OP_CODE lookup table

aluctrl	Operation	A	B	Expected Output
4'b0000	ADD	1	3	4
4'b0001	SUB	4	2	2
4'b0010	AND	5	7	5
4'b0011	OR	8	3	11
4'b0100	XNOR	13	3	-15
4'b0101	Compare	5	5	1
4'b0110	Left Shift	4	2	16
4'b0111	Right Shift	256	4	16
4'b1000	Substring Compare	15	3	1
4'b1001	Shift-then-Compare	15	3	0

- Verilog

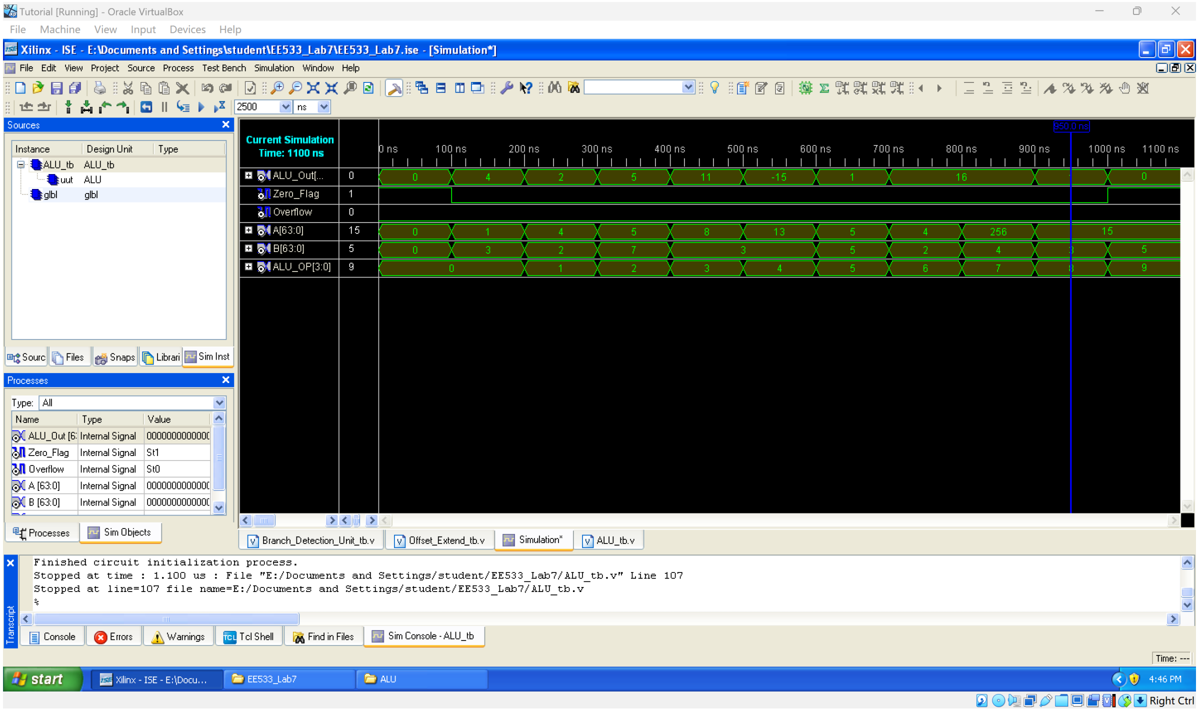
```verilog
`timescale 1ns / 1ps

module ALU (
    input  [63:0] A,
    input  [63:0] B,
    input  [3:0]  ALU_OP,
    output reg [63:0] ALU_Out,
    output reg Zero_Flag,
    output reg Overflow
);

    always @(*) begin
        case (ALU_OP)
            4'b0000: begin // Addition
                {Overflow, ALU_Out} = A + B;
            end
            4'b0001: begin // Subtraction
                {Overflow, ALU_Out} = A - B;
            end
            4'b0010: ALU_Out = A & B;              // Bitwise AND
            4'b0011: ALU_Out = A | B;              // Bitwise OR
            4'b0100: ALU_Out = A ^~ B;             // Bitwise XNOR
            4'b0101: ALU_Out = (A == B) ? 64'b1 : 64'b0; // Compare (Equality)
            4'b0110: ALU_Out = A << B[5:0];        // Logical Left Shift
            4'b0111: ALU_Out = A >> B[5:0];        // Logical Right Shift
            4'b1000: ALU_Out = substring_match(A, B); // Substring Compare
            4'b1001: ALU_Out = shift_then_compare(A, B); // Shift-then-Compare
            default: ALU_Out = 64'b0;
        endcase

        // Zero Flag
        Zero_Flag = (ALU_Out == 64'b0) ? 1'b1 : 1'b0;

    end

    // Function to check if B is a substring of A
    function [63:0] substring_match;
        input [63:0] A, B;
        integer i;
        begin
            substring_match = 64'b0;
            for (i = 0; i < 64; i = i + 1) begin
                if ((A >> i) & B == B) begin
                    substring_match = 64'b1;
                end
            end
        end
    endfunction

    // Function to shift A and then compare with B
    function [63:0] shift_then_compare;
        input [63:0] A, B;
        integer i;
        begin
            shift_then_compare = 64'b0;
            for (i = 0; i < 64; i = i + 1) begin
```

```
                if ((A >> i) == B) begin
                    shift_then_compare = 64'b1;
                end
            end
        end
    endfunction
endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   16:44:30 03/01/2025
// Design Name:   ALU
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/ALU_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: ALU
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module ALU_tb;

    // Inputs
    reg [63:0] A;
    reg [63:0] B;
    reg [3:0] ALU_OP;

    // Outputs
    wire [63:0] ALU_Out;
    wire Zero_Flag;
    wire Overflow;

    // Instantiate the Unit Under Test (UUT)
    ALU uut (
        .A(A),
        .B(B),
        .ALU_OP(ALU_OP),
        .ALU_Out(ALU_Out),
        .Zero_Flag(Zero_Flag),
        .Overflow(Overflow)
    );
    initial begin
        // Initialize Inputs
        A = 0;
        B = 0;
        ALU_OP = 0;

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

        // Add stimulus here
        A = 64'd1;
        B = 64'd3;
        ALU_OP = 4'b0000;
        #100;

        A = 64'd4;
        B = 64'd2;
        ALU_OP = 4'b0001;
        #100;

        A = 64'd5;
        B = 64'd7;
        ALU_OP = 4'b0010;
        #100;

        A = 64'd8;
        B = 64'd3;
        ALU_OP = 4'b0011;
        #100;

        A = 64'd13;
        B = 64'd3;
        ALU_OP = 4'b0100;
        #100;

        A = 64'd5;
        B = 64'd5;
        ALU_OP = 4'b0101;
        #100;

        A = 64'd4;
        B = 64'd2;
        ALU_OP = 4'b0110;
        #100;

        A = 64'd256;
        B = 64'd4;
        ALU_OP = 4'b0111;
        #100;

        A = 64'd15;
        B = 64'd3;
        ALU_OP = 4'b1000;
        #100;
        A = 64'd15;
        B = 64'd5;
        ALU_OP = 4'b1001;
        #100;

        $stop;

    end

endmodule
```

- Waveform


Tutorial [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation*]
File Edit View Project Source Process Test Bench Simulation Window Help
2500 ns
Sources
Instance Design Unit Type
ALU_tb ALU_tb
uut ALU
glbl glbl
Sourc Files Snaps Librari Sim Inst
Processes
Type: All
Name Type Value
ALU_Out [6: Internal Signal 000000000000(
Zero_Flag Internal Signal St1
Overflow Internal Signal St0
A [63:0] Internal Signal 000000000000(
B [63:0] Internal Signal 000000000000(
Processes Sim Objects
Current Simulation Time: 1100 ns
950.0 ns
0 ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns 1100 ns
ALU_Out[... 0 | 0 4 2 5 11 -15 1 16 0
Zero_Flag 1
Overflow 0
A[63:0] 15 | 0 1 4 5 8 13 5 4 256 15
B[63:0] 5 | 0 3 2 7 3 5 2 4 5
ALU_OP[3:0] 9 | 0 1 2 3 4 5 6 7 9
Branch_Detection_Unit_tb.v Offset_Extend_tb.v Simulation* ALU_tb.v
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - ALU_tb
Time: ---
start Xilinx - ISE - E:\Docu... EE533_Lab7 ALU
4:46 PM
Right Ctrl

```
Finished circuit initialization process.
Stopped at time : 1.100 us : File "E:/Documents and Settings/student/EE533_Lab7/ALU_tb.v" Line 107
Stopped at line=107 file name=E:/Documents and Settings/student/EE533_Lab7/ALU_tb.v
%
```

3.3.2 ALU_src_MUX

- Verilog

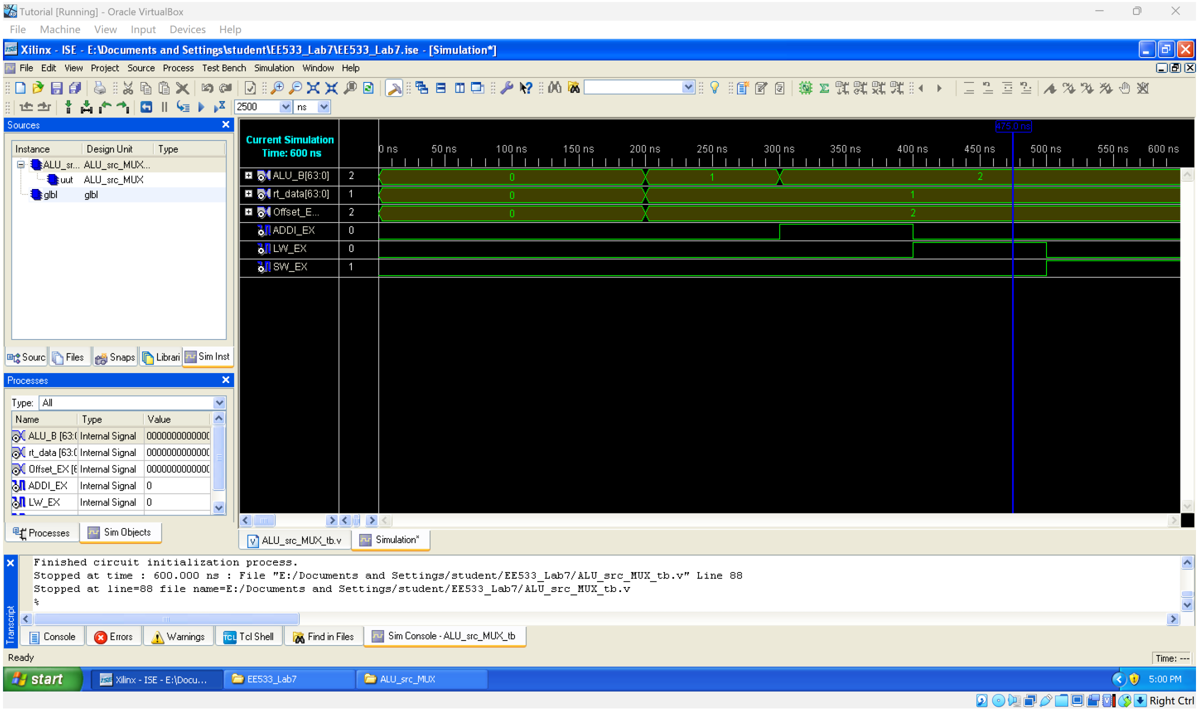
```
`timescale 1ns / 1ps

module ALU_src_MUX
(
    input [63:0] rt_data,
    input [63:0] Offset_EX,
    input ADDI_EX,
    input LW_EX,
    input SW_EX,

    output [63:0] ALU_B
);

    wire ALU_src_ctrl;

    assign ALU_src_ctrl = ADDI_EX || LW_EX || SW_EX;

    assign ALU_B = (ALU_src_ctrl == 1) ? Offset_EX : rt_data;
```

```
endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   16:58:15 03/01/2025
// Design Name:   ALU_src_MUX
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/ALU_src_MUX_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: ALU_src_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module ALU_src_MUX_tb;

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

    // Outputs
    wire [63:0] ALU_B;

    // Instantiate the Unit Under Test (UUT)
    ALU_src_MUX uut (
        .rt_data(rt_data),
        .Offset_EX(Offset_EX),
        .ADDI_EX(ADDI_EX),
        .LW_EX(LW_EX),
        .SW_EX(SW_EX),
        .ALU_B(ALU_B)
    );

    initial begin
        // Initialize Inputs
        rt_data = 0;
        Offset_EX = 0;
        ADDI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;

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

        // Add stimulus here
        #100;
        rt_data = 64'd1;
        Offset_EX = 64'd2;
        ADDI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;

        #100;
        rt_data = 64'd1;
        Offset_EX = 64'd2;
        ADDI_EX = 1;
        LW_EX = 0;
        SW_EX = 0;

        #100;
        rt_data = 64'd1;
        Offset_EX = 64'd2;
        ADDI_EX = 0;
        LW_EX = 1;
        SW_EX = 0;

        #100;
        rt_data = 64'd1;
        Offset_EX = 64'd2;
        ADDI_EX = 0;
        LW_EX = 0;
        SW_EX = 1;

        #100;
        $stop;

    end

endmodule
```

- Waveform



Name	Type	Value
ALU_B [63:(	Internal Signal	000000000000(
rt_data [63:(	Internal Signal	000000000000(
Offset_EX [(	Internal Signal	000000000000(
ADDI_EX	Internal Signal	0
LW_EX	Internal Signal	0

```
Finished circuit initialization process.
Stopped at time : 600.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/ALU_src_MUX_tb.v" Line 88
Stopped at line=88 file name=E:/Documents and Settings/student/EE533_Lab7/ALU_src_MUX_tb.v
%
```

3.4 MEM Stage

3.4.1 D_MEM

- Verilog

- Vector File

- Memory Initialization File

- Testbench

- Waveform

3.4.2 D_addr_src_MUX

- Verilog

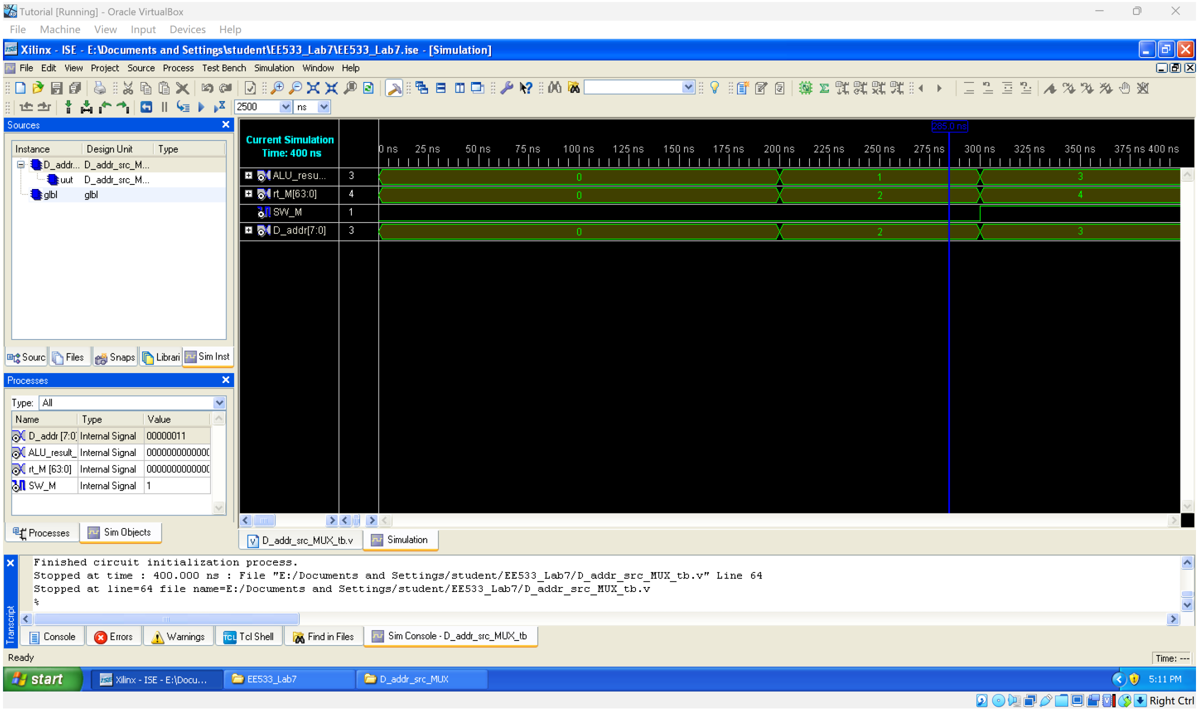
```
`timescale 1ns / 1ps

module D_addr_src_MUX
(
    input [63:0] ALU_result_M,
    input [4:0] rt_M,
    input SW_M,

    output [7:0] D_addr
);
```

```
    assign D_addr[4:0] = (SW_M == 1) ? ALU_result_M[4:0] : rt_M[4:0];
    assign D_addr[7:5] = 0;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   17:09:35 03/01/2025
// Design Name:   D_addr_src_MUX
// Module Name:   E:/Documents and
Settings/student/EE533_Lab7/D_addr_src_MUX_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: D_addr_src_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module D_addr_src_MUX_tb;

    // Inputs
    reg [63:0] ALU_result_M;
    reg [4:0] rt_M;
    reg SW_M;

    // Outputs
    wire [7:0] D_addr;

    // Instantiate the Unit Under Test (UUT)
    D_addr_src_MUX uut (
        .ALU_result_M(ALU_result_M),
        .rt_M(rt_M),
        .SW_M(SW_M),
        .D_addr(D_addr)
    );

    initial begin
        // Initialize Inputs
        ALU_result_M = 0;
        rt_M = 0;
        SW_M = 0;
        // Wait 100 ns for global reset to finish
        #100;

        // Add stimulus here
        #100;
        ALU_result_M = 64'd1;
        rt_M = 5'd2;
        SW_M = 0;

        #100;
        ALU_result_M = 64'd3;
        rt_M = 5'd4;
        SW_M = 1;

        #100;
        $stop;

    end

endmodule
```

- Waveform


Tutorial [Running] - Oracle VirtualBox
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation]
Current Simulation Time: 400 ns
0 ns 25 ns 50 ns 75 ns 100 ns 125 ns 150 ns 175 ns 200 ns 225 ns 250 ns 275 ns 300 ns 325 ns 350 ns 375 ns 400 ns
ALU_resu... 3 0 1 3
rt_M[63:0] 4 0 2 4
SW_M 1
D_addr[7:0] 3 0 2 3
D_addr... D_addr_src_M...
uut D_addr_src_M...
glbl glbl
D_addr_src_MUX_tb.v
Simulation
Console
Errors
Warnings
Tcl Shell
Find in Files
Sim Console - D_addr_src_MUX_tb
Ready

Type: All

Name	Type	Value
D_addr [7:0	Internal Signal	00000011
ALU_result_	Internal Signal	000000000000
rt_M [63:0]	Internal Signal	000000000000
SW_M	Internal Signal	1

```
Finished circuit initialization process.
Stopped at time : 400.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/D_addr_src_MUX_tb.v" Line 64
Stopped at line=64 file name=E:/Documents and Settings/student/EE533_Lab7/D_addr_src_MUX_tb.v
%
```

3.5 WB Stage

3.5.1 RF_WB_data_src_MUX

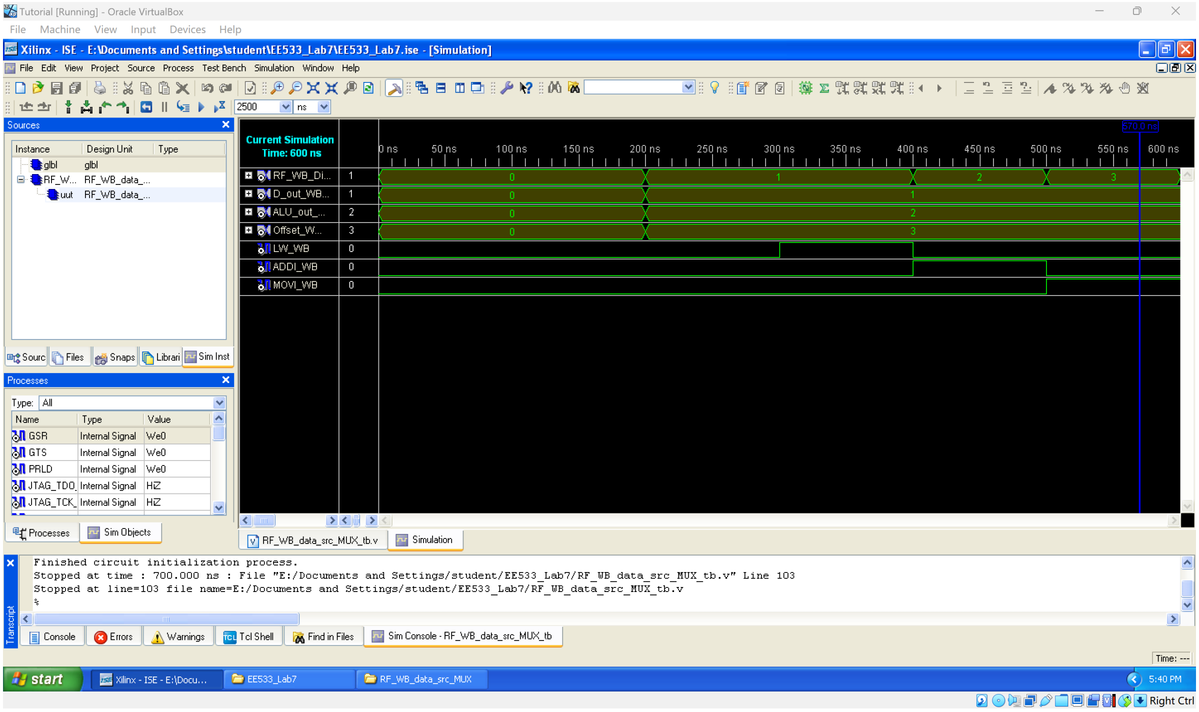
- Verilog

```
`timescale 1ns / 1ps

module RF_WB_data_src_MUX
(
    input [63:0] D_out_WB,
    input [63:0] ALU_out_WB,
    input [63:0] Offset_WB,

    input LW_WB,
    input ADDI_WB,
    input MOVI_WB,

    output [63:0] RF_WB_Din
);

    wire [63:0] temp;

    assign temp = (~LW_WB && ADDI_WB) ? ALU_out_WB : D_out_WB;
    assign RF_WB_Din = (MOVI_WB && ~LW_WB && ~ADDI_WB) ? Offset_WB : temp;

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   17:24:15 03/01/2025
// Design Name:   RF_WB_data_src_MUX
// Module Name:   E:/Documents and
Settings/student/EE533_Lab7/RF_WB_data_src_MUX_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: RF_WB_data_src_MUX
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module RF_WB_data_src_MUX_tb;

    // Inputs
    reg [63:0] D_out_WB;
    reg [63:0] ALU_out_WB;
    reg [63:0] Offset_WB;
    reg LW_WB;
    reg ADDI_WB;
    reg MOVI_WB;

    // Outputs
    wire [63:0] RF_WB_Din;
    // Instantiate the Unit Under Test (UUT)
    RF_WB_data_src_MUX uut (
        .D_out_WB(D_out_WB),
        .ALU_out_WB(ALU_out_WB),
        .Offset_WB(Offset_WB),
        .LW_WB(LW_WB),
        .ADDI_WB(ADDI_WB),
        .MOVI_WB(MOVI_WB),
        .RF_WB_Din(RF_WB_Din)
    );

    initial begin
        // Initialize Inputs
        D_out_WB = 0;
        ALU_out_WB = 0;
        Offset_WB = 0;
        LW_WB = 0;
        ADDI_WB = 0;
        MOVI_WB = 0;

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

        // Add stimulus here
        #100;
        D_out_WB = 64'd1;
        ALU_out_WB = 64'd2;
        Offset_WB = 64'd3;
        LW_WB = 0;
        ADDI_WB = 0;
        MOVI_WB = 0;

        #100;
        D_out_WB = 64'd1;
        ALU_out_WB = 64'd2;
        Offset_WB = 64'd3;
        LW_WB = 1;
        ADDI_WB = 0;
        MOVI_WB = 0;

        #100;
        D_out_WB = 64'd1;
        ALU_out_WB = 64'd2;
        Offset_WB = 64'd3;
        LW_WB = 0;
        ADDI_WB = 1;
        MOVI_WB = 0;

        #100;
        D_out_WB = 64'd1;
        ALU_out_WB = 64'd2;
        Offset_WB = 64'd3;
        LW_WB = 0;
        ADDI_WB = 0;
        MOVI_WB = 1;
        #100;
        D_out_WB = 64'd1;
        ALU_out_WB = 64'd2;
        Offset_WB = 64'd3;
        LW_WB = 0;
        ADDI_WB = 0;
        MOVI_WB = 0;

        #100;
        $stop;

    end

endmodule
```

- Waveform



```
Finished circuit initialization process.
Stopped at time : 700.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/RF_WB_data_src_MUX_tb.v" Line 103
Stopped at line=103 file name=E:/Documents and Settings/student/EE533_Lab7/RF_WB_data_src_MUX_tb.v
%
```

3.6 Stage Reg

3.6.1 IF_ID_Reg

- Verilog

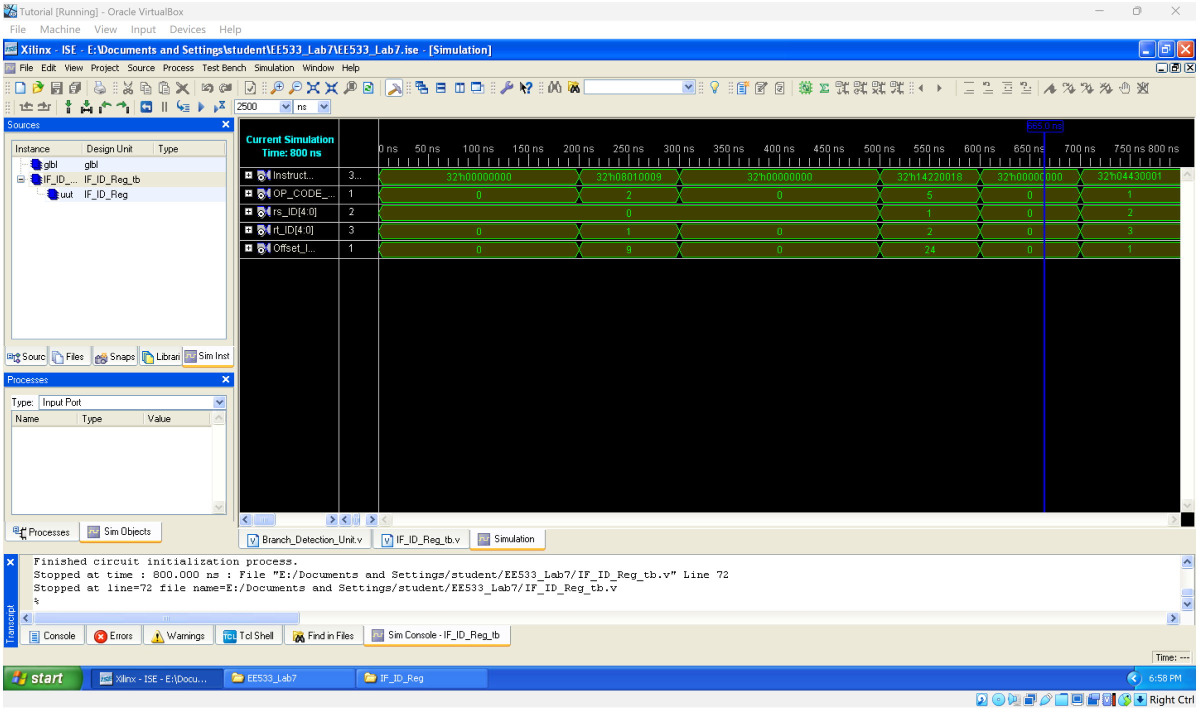
```
`timescale 1ns / 1ps

module IF_ID_Reg
(
    input [31:0] Instruction,

    output [5:0] OP_CODE_ID,
    output [4:0] rs_ID,
    output [4:0] rt_ID,
    output [15:0] Offset_ID
);
```

```
    assign OP_CODE_ID = Instruction[31:26];
    assign rs_ID = Instruction[25:21];
    assign rt_ID = Instruction[20:16];
    assign Offset_ID = Instruction[15:0];

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   18:48:10 03/01/2025
// Design Name:   IF_ID_Reg
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/IF_ID_Reg_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: IF_ID_Reg
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module IF_ID_Reg_tb;

    // Inputs
    reg [31:0] Instruction;

    // Outputs
    wire [5:0] OP_CODE_ID;
    wire [4:0] rs_ID;
    wire [4:0] rt_ID;
    wire [15:0] Offset_ID;

    // Instantiate the Unit Under Test (UUT)
    IF_ID_Reg uut (
        .Instruction(Instruction),
        .OP_CODE_ID(OP_CODE_ID),
        .rs_ID(rs_ID),
        .rt_ID(rt_ID),
        .Offset_ID(Offset_ID)
    );

    initial begin
        // Initialize Inputs
        Instruction = 0;

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

        // Add stimulus here
        #100;
        Instruction = 32'b00001000000000010000000000001001;

        #100;
        Instruction = 32'b0;

        #100;
        Instruction = 32'b0;

        #100;
        Instruction = 32'b00010100001000100000000000011000;

        #100;
        Instruction = 32'b0;

        #100;
        Instruction = 32'b00000100010000110000000000000001;

        #100;
        $stop;

    end

endmodule
```

- Waveform


Tutorial [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
2500 ns
Sources
Instance Design Unit Type
glbl glbl
IF_ID_... IF_ID_Reg_tb
uut IF_ID_Reg
Sourc Files Snaps Librari Sim Inst
Processes
Type: Input Port
Name Type Value
Processes Sim Objects
Current Simulation Time: 800 ns
665.0 ns
0 ns 50 ns 100 ns 150 ns 200 ns 250 ns 300 ns 350 ns 400 ns 450 ns 500 ns 550 ns 600 ns 650 ns 700 ns 750 ns 800 ns
Instruct... 3... 32'h00000000 32'h08010009 32'h00000000 32'h14220018 32'h00000000 32'h04430001
OP_CODE_... 1 0 2 0 5 0 1
rs_ID[4:0] 2 0 1 0 2
rt_ID[4:0] 3 0 1 0 2 0 3
Offset_I... 1 0 9 0 24 0 1
Branch_Detection_Unit.v IF_ID_Reg_tb.v Simulation
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - IF_ID_Reg_tb
Time: ---
start Xilinx - ISE - E:\Docu... EE533_Lab7 IF_ID_Reg
6:58 PM
Right Ctrl

```
Finished circuit initialization process.
Stopped at time : 800.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/IF_ID_Reg_tb.v" Line 72
Stopped at line=72 file name=E:/Documents and Settings/student/EE533_Lab7/IF_ID_Reg_tb.v
%
```

3.6.2 ID_EX_Reg

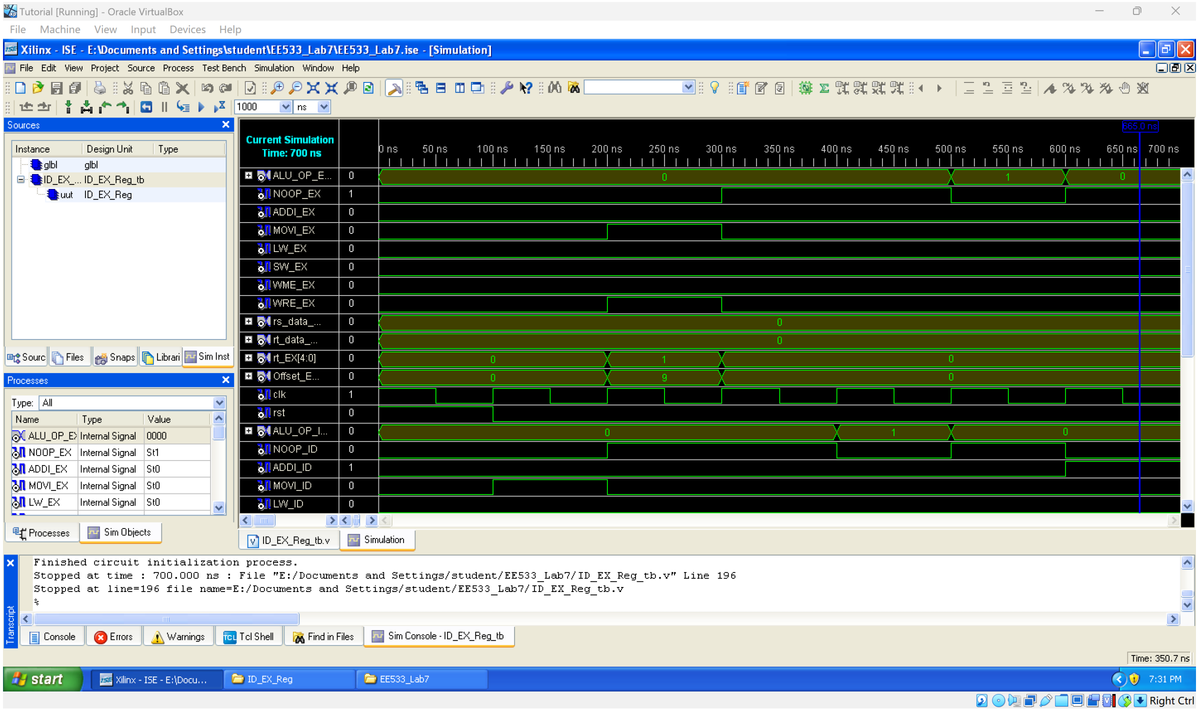
- Verilog

```
`timescale 1ns / 1ps

module ID_EX_Reg
(
    input clk,
    input rst,

    input [3:0] ALU_OP_ID,
    input NOOP_ID,
    input ADDI_ID,
    input MOVI_ID,
    input LW_ID,
    input SW_ID,

    input WME_ID,
    input WRE_ID,
    input [63:0] rs_data_ID,
    input [63:0] rt_data_ID,
    input [4:0] rt_ID,
    input [63:0] Offset_ID,

    output reg [3:0] ALU_OP_EX,
    output reg NOOP_EX,
    output reg ADDI_EX,
    output reg MOVI_EX,
    output reg LW_EX,
    output reg SW_EX,

    output reg WME_EX,
    output reg WRE_EX,
    output reg [63:0] rs_data_EX,
    output reg [63:0] rt_data_EX,
    output reg [4:0] rt_EX,
    output reg [63:0] Offset_EX
);

    always @(posedge clk) begin
        if (rst) begin
            ALU_OP_EX <= 0;
            NOOP_EX <= 0;
            ADDI_EX <= 0;
            MOVI_EX <= 0;
            LW_EX <= 0;
            SW_EX <= 0;
            WME_EX <= 0;
            WRE_EX <= 0;
            rs_data_EX <= 0;
            rt_data_EX <= 0;
            rt_EX <= 0;
            Offset_EX <= 0;
        end
        else begin
            ALU_OP_EX <= ALU_OP_ID;
            NOOP_EX <= NOOP_ID;
            ADDI_EX <= ADDI_ID;
            MOVI_EX <= MOVI_ID;
            LW_EX <= LW_ID;
            SW_EX <= SW_ID;
            WME_EX <= WME_ID;
            WRE_EX <= WRE_ID;
            rs_data_EX <= rs_data_ID;
            rt_data_EX <= rt_data_ID;
            rt_EX <= rt_ID;
            Offset_EX <= Offset_ID;
        end
    end

endmodule
```

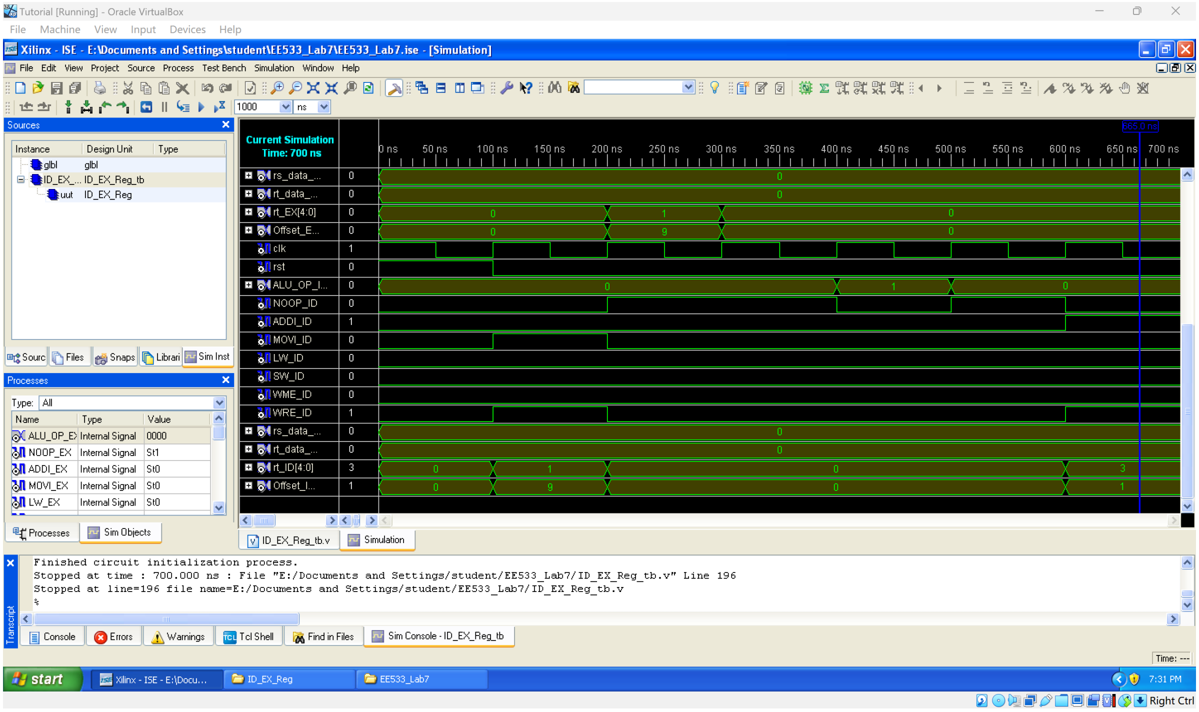
- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   19:13:54 03/01/2025
// Design Name:   ID_EX_Reg
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/ID_EX_Reg_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: ID_EX_Reg
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module ID_EX_Reg_tb;

    // Inputs
    reg clk;
    reg rst;
    reg [3:0] ALU_OP_ID;
    reg NOOP_ID;
    reg ADDI_ID;
    reg MOVI_ID;
    reg LW_ID;
    reg SW_ID;
    reg WME_ID;
    reg WRE_ID;
    reg [63:0] rs_data_ID;
    reg [63:0] rt_data_ID;
    reg [4:0] rt_ID;
    reg [63:0] Offset_ID;

    // Outputs
    wire [3:0] ALU_OP_EX;
    wire NOOP_EX;
    wire ADDI_EX;
    wire MOVI_EX;
    wire LW_EX;
    wire SW_EX;
    wire WME_EX;
    wire WRE_EX;
    wire [63:0] rs_data_EX;
    wire [63:0] rt_data_EX;
    wire [4:0] rt_EX;
    wire [63:0] Offset_EX;

    // Instantiate the Unit Under Test (UUT)
    ID_EX_Reg uut (
        .clk(clk),
        .rst(rst),
        .ALU_OP_ID(ALU_OP_ID),
        .NOOP_ID(NOOP_ID),
        .ADDI_ID(ADDI_ID),
        .MOVI_ID(MOVI_ID),
        .LW_ID(LW_ID),
        .SW_ID(SW_ID),
        .WME_ID(WME_ID),
        .WRE_ID(WRE_ID),
        .rs_data_ID(rs_data_ID),
        .rt_data_ID(rt_data_ID),
        .rt_ID(rt_ID),
        .Offset_ID(Offset_ID),
        .ALU_OP_EX(ALU_OP_EX),
        .NOOP_EX(NOOP_EX),
        .ADDI_EX(ADDI_EX),
        .MOVI_EX(MOVI_EX),
        .LW_EX(LW_EX),
        .SW_EX(SW_EX),
        .WME_EX(WME_EX),
        .WRE_EX(WRE_EX),
        .rs_data_EX(rs_data_EX),
        .rt_data_EX(rt_data_EX),
        .rt_EX(rt_EX),
        .Offset_EX(Offset_EX)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 1;
        rst = 1;
        ALU_OP_ID = 0;
        NOOP_ID = 0;
        ADDI_ID = 0;
        MOVI_ID = 0;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 0;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 0;
        Offset_ID = 0;

        // Wait 100 ns for global reset to finish
        #100;
        rst = 0;

        // Add stimulus here
        @(posedge clk);
        ALU_OP_ID = 4'd0;
        NOOP_ID = 0;
        ADDI_ID = 0;
        MOVI_ID = 1;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 1;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 5'd1;
        Offset_ID = 16'd9;

        @(posedge clk);
        ALU_OP_ID = 4'd0;
        NOOP_ID = 1;
        ADDI_ID = 0;
        MOVI_ID = 0;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 0;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 5'd0;
        Offset_ID = 16'd0;

        @(posedge clk);
        ALU_OP_ID = 4'd0;
        NOOP_ID = 1;
        ADDI_ID = 0;
        MOVI_ID = 0;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 0;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 5'd0;
        Offset_ID = 16'd0;

        @(posedge clk);
        ALU_OP_ID = 4'd1;
        NOOP_ID = 0;
        ADDI_ID = 0;
        MOVI_ID = 0;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 0;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 5'd0;
        Offset_ID = 16'd0;

        @(posedge clk);
        ALU_OP_ID = 4'd0;
        NOOP_ID = 1;
        ADDI_ID = 0;
        MOVI_ID = 0;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 0;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 5'd0;
        Offset_ID = 16'd0;

        @(posedge clk);
        ALU_OP_ID = 4'd0;
        NOOP_ID = 0;
        ADDI_ID = 1;
        MOVI_ID = 0;
        LW_ID = 0;
        SW_ID = 0;
        WME_ID = 0;
        WRE_ID = 1;
        rs_data_ID = 0;
        rt_data_ID = 0;
        rt_ID = 5'd3;
        Offset_ID = 16'd1;

        @(posedge clk);
        $stop;

    end

endmodule
```

- Waveform



```
Finished circuit initialization process.
Stopped at time : 700.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/ID_EX_Reg_tb.v" Line 196
Stopped at line=196 file name=E:/Documents and Settings/student/EE533_Lab7/ID_EX_Reg_tb.v
%
```



```
Finished circuit initialization process.
Stopped at time : 700.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/ID_EX_Reg_tb.v" Line 196
Stopped at line=196 file name=E:/Documents and Settings/student/EE533_Lab7/ID_EX_Reg_tb.v
%
```

3.6.3 EX_M_Reg

- Verilog

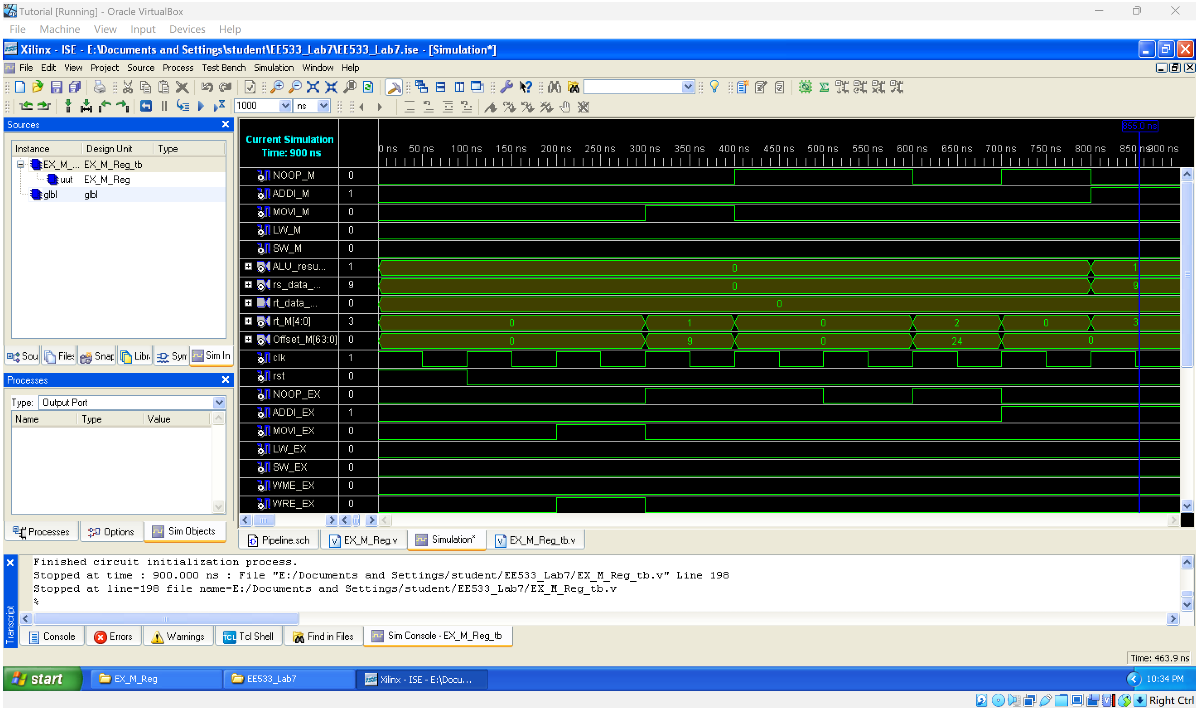
```
`timescale 1ns / 1ps

module EX_M_Reg
(
    input clk,
    input rst,

    input NOOP_EX,
    input ADDI_EX,
    input MOVI_EX,
    input LW_EX,
    input SW_EX,
    input WME_EX,
    input WRE_EX,
    input [63:0] ALU_result_EX,
    input [63:0] rs_data_EX,
    input [63:0] rt_data_EX,
    input [4:0] rt_EX,
    input [63:0] Offset_EX,

    output reg NOOP_M,
    output reg ADDI_M,
    output reg MOVI_M,
    output reg LW_M,
    output reg SW_M,

    output reg WME_M,
    output reg WRE_M,
    output reg [63:0] ALU_result_M,
    output reg [63:0] rs_data_M,
    output reg [63:0] rt_data_M,
    output reg [4:0] rt_M,
    output reg [63:0] Offset_M
);

    always @(posedge clk) begin
        if (rst) begin
            NOOP_M <= 0;
            ADDI_M <= 0;
            MOVI_M <= 0;
            LW_M <= 0;
            SW_M <= 0;

            WME_M <= 0;
            WRE_M <= 0;
            ALU_result_M <= 0;
            rs_data_M <= 0;
            rt_data_M <= 0;
            rt_M <= 0;
            Offset_M <= 0;
        end
        else begin
            NOOP_M <= NOOP_EX;
            ADDI_M <= ADDI_EX;
            MOVI_M <= MOVI_EX;
            LW_M <= LW_EX;
            SW_M <= SW_EX;

            WME_M <= WME_EX;
            WRE_M <= WRE_EX;
            ALU_result_M <= ALU_result_EX;
            rs_data_M <= rs_data_EX;
            rt_data_M <= rt_data_EX;
            rt_M <= rt_EX;
            Offset_M <= Offset_EX;
        end
    end
```

```
endmodule
```

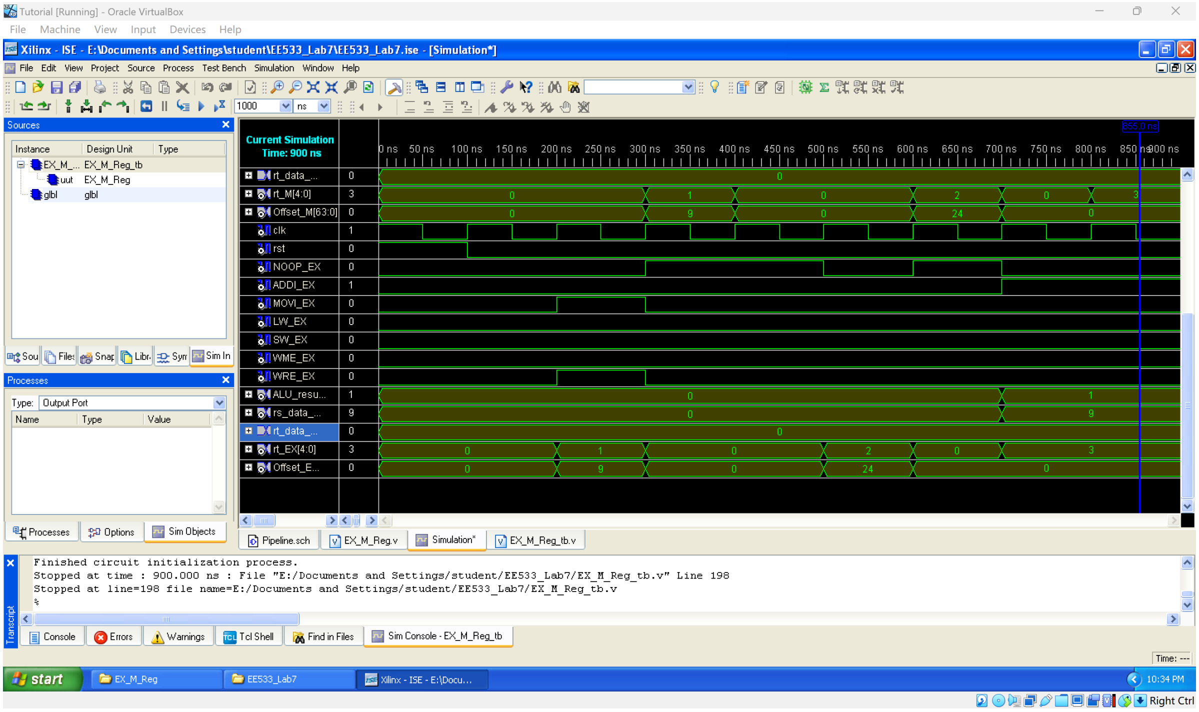
- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   19:34:10 03/01/2025
// Design Name:   EX_M_Reg
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/EX_M_Reg_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: EX_M_Reg
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module EX_M_Reg_tb;

    // Inputs
    reg clk;
    reg rst;
    reg NOOP_EX;
    reg ADDI_EX;
    reg MOVI_EX;
    reg LW_EX;
    reg SW_EX;
    reg WME_EX;
    reg WRE_EX;
    reg [63:0] ALU_result_EX;
    reg [63:0] rs_data_EX;
    reg [63:0] rt_data_EX;
    reg [4:0] rt_EX;
    reg [63:0] Offset_EX;

    // Outputs
    wire NOOP_M;
    wire ADDI_M;
    wire MOVI_M;
    wire LW_M;
    wire SW_M;
    wire WME_M;
    wire WRE_M;
    wire [63:0] ALU_result_M;
    wire [63:0] rs_data_M;
    wire [63:0] rt_data_M;
    wire [4:0] rt_M;
    wire [63:0] Offset_M;

    // Instantiate the Unit Under Test (UUT)
    EX_M_Reg uut (
        .clk(clk),
        .rst(rst),
        .NOOP_EX(NOOP_EX),
        .ADDI_EX(ADDI_EX),
        .MOVI_EX(MOVI_EX),
        .LW_EX(LW_EX),
        .SW_EX(SW_EX),
        .WME_EX(WME_EX),
        .WRE_EX(WRE_EX),
        .ALU_result_EX(ALU_result_EX),
        .rs_data_EX(rs_data_EX),
        .rt_data_EX(rt_data_EX),
        .rt_EX(rt_EX),
        .Offset_EX(Offset_EX),
        .NOOP_M(NOOP_M),
        .ADDI_M(ADDI_M),
        .MOVI_M(MOVI_M),
        .LW_M(LW_M),
        .SW_M(SW_M),
        .WME_M(WME_M),
        .WRE_M(WRE_M),
        .ALU_result_M(ALU_result_M),
        .rs_data_M(rs_data_M),
        .rt_data_M(rt_data_M),
        .rt_M(rt_M),
        .Offset_M(Offset_M)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 1;
        rst = 1;
        NOOP_EX = 0;
        ADDI_EX = 0;
        MOVI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 0;
        ALU_result_EX = 0;
        rs_data_EX = 0;
        rt_data_EX = 0;
        rt_EX = 0;
        Offset_EX = 0;

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

        // Add stimulus here
        @(posedge clk);
        NOOP_EX = 0;
        ADDI_EX = 0;
        MOVI_EX = 1;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 1;
        ALU_result_EX = 0;
        rs_data_EX = 0;
        rt_data_EX = 0;
        rt_EX = 5'd1;
        Offset_EX = 16'd9;

        @(posedge clk);
        NOOP_EX = 1;
        ADDI_EX = 0;
        MOVI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 0;
        ALU_result_EX = 0;
        rs_data_EX = 0;
        rt_data_EX = 0;
        rt_EX = 5'd0;
        Offset_EX = 16'd0;

        @(posedge clk);
        NOOP_EX = 1;
        ADDI_EX = 0;
        MOVI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 0;
        ALU_result_EX = 0;
        rs_data_EX = 0;
        rt_data_EX = 0;
        rt_EX = 5'd0;
        Offset_EX = 16'd0;

        @(posedge clk);
        NOOP_EX = 0;
        ADDI_EX = 0;
        MOVI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 0;
        ALU_result_EX = 0;
        rs_data_EX = 0;
        rt_data_EX = 0;
        rt_EX = 5'd2;
        Offset_EX = 16'd24;

        @(posedge clk);
        NOOP_EX = 1;
        ADDI_EX = 0;
        MOVI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 0;
        ALU_result_EX = 0;
        rs_data_EX = 0;
        rt_data_EX = 0;
        rt_EX = 5'd0;
        Offset_EX = 16'd0;

        @(posedge clk);
        NOOP_EX = 0;
        ADDI_EX = 1;
        MOVI_EX = 0;
        LW_EX = 0;
        SW_EX = 0;
        WME_EX = 0;
        WRE_EX = 0;
        ALU_result_EX = 64'd1;
        rs_data_EX = 64'd9;
        rt_data_EX = 0;
        rt_EX = 5'd3;
        Offset_EX = 16'd0;

        @(posedge clk);

        @(posedge clk);
        $stop;

    end

endmodule
```

- Waveform



```
Finished circuit initialization process.
Stopped at time : 900.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/EX_M_Reg_tb.v" Line 198
Stopped at line=198 file name=E:/Documents and Settings/student/EE533_Lab7/EX_M_Reg_tb.v
%
```



```
Finished circuit initialization process.
Stopped at time : 900.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/EX_M_Reg_tb.v" Line 198
Stopped at line=198 file name=E:/Documents and Settings/student/EE533_Lab7/EX_M_Reg_tb.v
%
```

3.6.4 M_WB_Reg

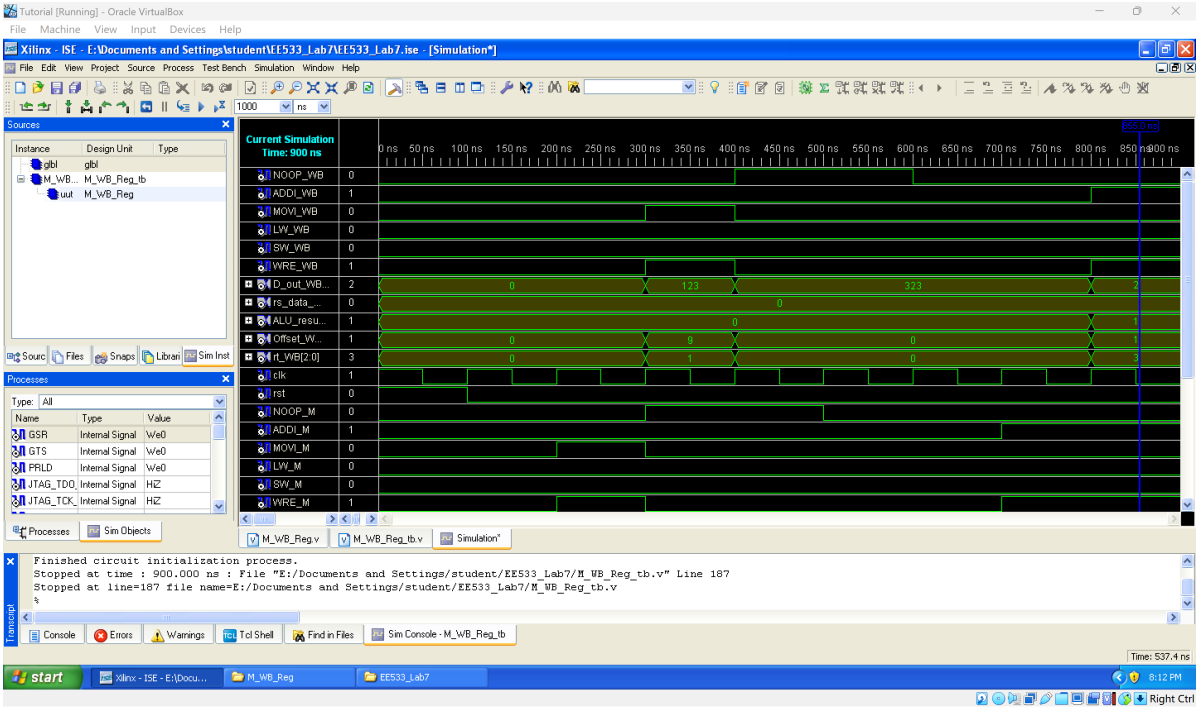
- Verilog

```
`timescale 1ns / 1ps

module M_WB_Reg
(
    input clk,
    input rst,

    input NOOP_M,
    input ADDI_M,
    input MOVI_M,
    input LW_M,
    input SW_M,
    input WRE_M,
    input [63:0] D_out_M,
    input [63:0] rs_data_M,
    input [63:0] ALU_result_M,
    input [63:0] Offset_M,
    input [4:0] rt_M,

    output reg NOOP_WB,
    output reg ADDI_WB,
    output reg MOVI_WB,
    output reg LW_WB,
    output reg SW_WB,

    output reg WRE_WB,
    output reg [63:0] D_out_WB,
    output reg [63:0] rs_data_WB,
    output reg [63:0] ALU_result_WB,
    output reg [63:0] Offset_WB,
    output reg [2:0] rt_WB
);

    always @(posedge clk) begin
        if (rst) begin
            NOOP_WB <= 0;
            ADDI_WB <= 0;
            MOVI_WB <= 0;
            LW_WB <= 0;
            SW_WB <= 0;
            WRE_WB <= 0;
            D_out_WB <= 0;
            rs_data_WB <= 0;
            ALU_result_WB <= 0;
            Offset_WB <= 0;
            rt_WB <= 0;
        end
        else begin
            NOOP_WB <= NOOP_M;
            ADDI_WB <= ADDI_M;
            MOVI_WB <= MOVI_M;
            LW_WB <= LW_M;
            SW_WB <= SW_M;
            WRE_WB <= WRE_M;
            D_out_WB <= D_out_M;
            rs_data_WB <= rs_data_M;
            ALU_result_WB <= ALU_result_M;
            Offset_WB <= Offset_M;
            rt_WB <= rt_M[2:0];
        end
    end

endmodule
```

- Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   19:46:24 03/01/2025
// Design Name:   M_WB_Reg
// Module Name:   E:/Documents and Settings/student/EE533_Lab7/M_WB_Reg_tb.v
// Project Name:  EE533_Lab7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: M_WB_Reg
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module M_WB_Reg_tb;

    // Inputs
    reg clk;
    reg rst;
    reg NOOP_M;
    reg ADDI_M;
    reg MOVI_M;
    reg LW_M;
    reg SW_M;
    reg WRE_M;
    reg [63:0] D_out_M;
    reg [63:0] rs_data_M;
    reg [63:0] ALU_result_M;
    reg [63:0] Offset_M;
    reg [4:0] rt_M;

    // Outputs
    wire NOOP_WB;
    wire ADDI_WB;
    wire MOVI_WB;
    wire LW_WB;
    wire SW_WB;
    wire WRE_WB;
    wire [63:0] D_out_WB;
    wire [63:0] rs_data_WB;
    wire [63:0] ALU_result_WB;
    wire [63:0] Offset_WB;
    wire [2:0] rt_WB;

    // Instantiate the Unit Under Test (UUT)
    M_WB_Reg uut (
        .clk(clk),
        .rst(rst),
        .NOOP_M(NOOP_M),
        .ADDI_M(ADDI_M),
        .MOVI_M(MOVI_M),
        .LW_M(LW_M),
        .SW_M(SW_M),
        .WRE_M(WRE_M),
        .D_out_M(D_out_M),
        .rs_data_M(rs_data_M),
        .ALU_result_M(ALU_result_M),
        .Offset_M(Offset_M),
        .rt_M(rt_M),
        .NOOP_WB(NOOP_WB),
        .ADDI_WB(ADDI_WB),
        .MOVI_WB(MOVI_WB),
        .LW_WB(LW_WB),
        .SW_WB(SW_WB),
        .WRE_WB(WRE_WB),
        .D_out_WB(D_out_WB),
        .rs_data_WB(rs_data_WB),
        .ALU_result_WB(ALU_result_WB),
        .Offset_WB(Offset_WB),
        .rt_WB(rt_WB)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 1;
        rst = 1;
        NOOP_M = 0;
        ADDI_M = 0;
        MOVI_M = 0;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 0;
        D_out_M = 0;
        rs_data_M = 0;
        ALU_result_M = 0;
        Offset_M = 0;
        rt_M = 0;

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

        // Add stimulus here
        @(posedge clk);
        NOOP_M = 0;
        ADDI_M = 0;
        MOVI_M = 1;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 1;
        D_out_M = 64'd123;
        rs_data_M = 64'd0;
        ALU_result_M = 0;
        Offset_M = 64'd9;
        rt_M = 5'd1;

        @(posedge clk);
        NOOP_M = 1;
        ADDI_M = 0;
        MOVI_M = 0;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 0;
        D_out_M = 64'd323;
        rs_data_M = 64'd0;
        ALU_result_M = 0;
        Offset_M = 64'd0;
        rt_M = 5'd0;

        @(posedge clk);
        NOOP_M = 1;
        ADDI_M = 0;
        MOVI_M = 0;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 0;
        D_out_M = 64'd323;
        rs_data_M = 64'd0;
        ALU_result_M = 0;
        Offset_M = 64'd0;
        rt_M = 5'd0;

        @(posedge clk);
        NOOP_M = 0;
        ADDI_M = 0;
        MOVI_M = 0;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 0;
        D_out_M = 64'd323;
        rs_data_M = 64'd0;
        ALU_result_M = 0;
        Offset_M = 64'd0;
        rt_M = 5'd0;

        @(posedge clk);
        NOOP_M = 0;
        ADDI_M = 0;
        MOVI_M = 0;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 0;
        D_out_M = 64'd323;
        rs_data_M = 64'd0;
        ALU_result_M = 0;
        Offset_M = 64'd0;
        rt_M = 5'd0;

        @(posedge clk);
        NOOP_M = 0;
        ADDI_M = 1;
        MOVI_M = 0;
        LW_M = 0;
        SW_M = 0;
        WRE_M = 1;
        D_out_M = 64'd2;
        rs_data_M = 64'd0;
        ALU_result_M = 64'd1;
        Offset_M = 64'd1;
        rt_M = 5'd3;

        @(posedge clk);

        @(posedge clk);
        $stop;

    end

endmodule
```

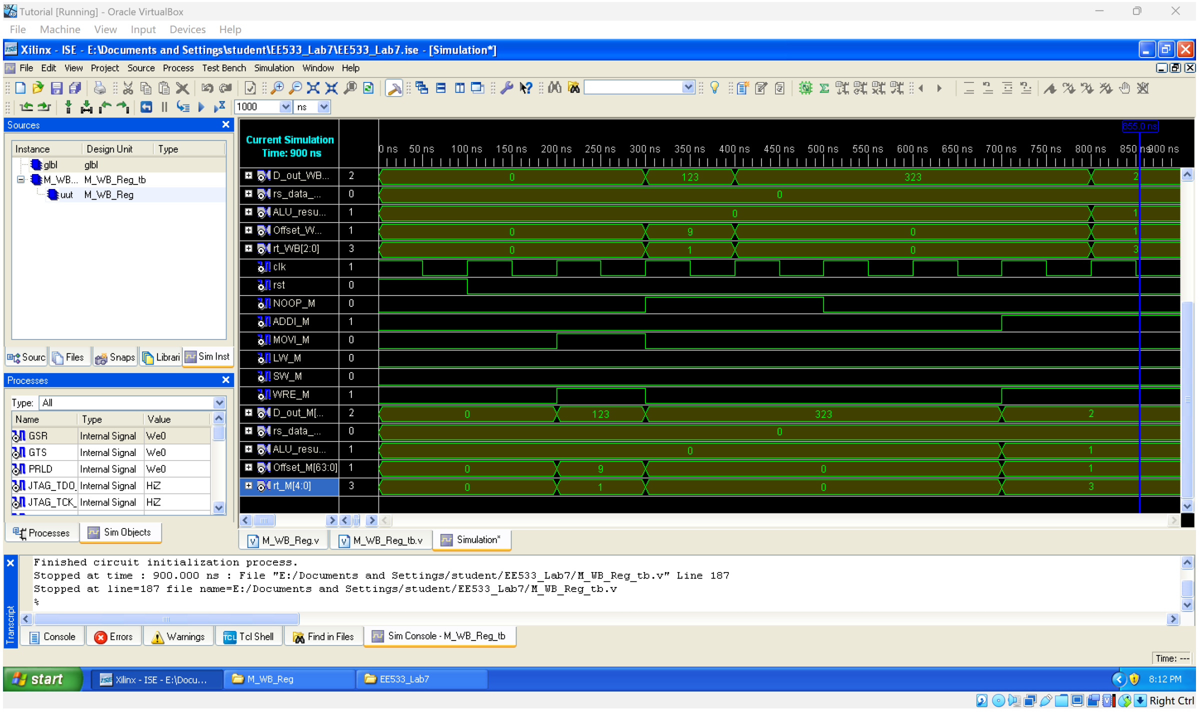
- Waveform


Tutorial [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Xilinx - ISE - E:\Documents and Settings\student\EE533_Lab7\EE533_Lab7.ise - [Simulation*]
File Edit View Project Source Process Test Bench Simulation Window Help
Sources
Instance Design Unit Type
glbl glbl
M_WB... M_WB_Reg_tb
uut M_WB_Reg
Sourc Files Snaps Librari Sim Inst
Processes
Type: All
Current Simulation Time: 900 ns
0 ns 50 ns 100 ns 150 ns 200 ns 250 ns 300 ns 350 ns 400 ns 450 ns 500 ns 550 ns 600 ns 650 ns 700 ns 750 ns 800 ns 850 ns 900 ns
0 123 323 2
0 9 0 1
0 1 0 3
Processes Sim Objects
M_WB_Reg.v M_WB_Reg_tb.v Simulation*
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - M_WB_Reg_tb
Time: 537.4 ns
start Xilinx - ISE - E:\Docu... M_WB_Reg EE533_Lab7
8:12 PM
Right Ctrl

Name	Type	Value
GSR	Internal Signal	We0
GTS	Internal Signal	We0
PRLD	Internal Signal	We0
JTAG_TDO_	Internal Signal	HiZ
JTAG_TCK_	Internal Signal	HiZ

NOOP_WB	0
ADDI_WB	1
MOVI_WB	0
LW_WB	0
SW_WB	0
WRE_WB	1
D_out_WB...	2
rs_data_...	0
ALU_resu...	1
Offset_W...	1
rt_WB[2:0]	3
clk	1
rst	0
NOOP_M	0
ADDI_M	1
MOVI_M	0
LW_M	0
SW_M	0
WRE_M	1

```
Finished circuit initialization process.
Stopped at time : 900.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/M_WB_Reg_tb.v" Line 187
Stopped at line=187 file name=E:/Documents and Settings/student/EE533_Lab7/M_WB_Reg_tb.v
%
```



Name	Type	Value
GSR	Internal Signal	We0
GTS	Internal Signal	We0
PRLD	Internal Signal	We0
JTAG_TDO_	Internal Signal	HiZ
JTAG_TCK_	Internal Signal	HiZ

```
Finished circuit initialization process.
Stopped at time : 900.000 ns : File "E:/Documents and Settings/student/EE533_Lab7/M_WB_Reg_tb.v" Line 187
Stopped at line=187 file name=E:/Documents and Settings/student/EE533_Lab7/M_WB_Reg_tb.v
%
```

4. 5-Stage Pipeline Generated

4.1 Schematic

4.2 Verilog

```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : Pipeline.vf
// /___/   /\     Timestamp : 03/01/2025 22:50:30
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w "E:/Documents and Settings/student/EE533_Lab7/Pipeline.sch" Pipeline.vf
//Design Name: Pipeline
//Device: virtex2p
//Purpose:
//    This verilog netlist is translated from an ECS schematic.It can be
//    synthesized and simulated, but it should not be modified.
//
`timescale 1ns / 1ps

module Pipeline(clk,
                Instr_IN,
                Instr_W_en,
                 I_W_Addr,
                 rst);

     input clk;
     input [31:0] Instr_IN;
     input Instr_W_en;
     input [8:0] I_W_Addr;
     input rst;

    wire ADDI_EX;
    wire ADDI_ID;
    wire ADDI_M;
    wire ADDI_WB;
    wire [63:0] ALU_B;
    wire [3:0] ALU_OP_EX;
    wire [3:0] ALU_OP_ID;
    wire [63:0] ALU_result_EX;
    wire [63:0] ALU_result_M;
    wire [63:0] ALU_result_WB;
    wire BEQ_ID;
    wire BGT_ID;
    wire BLT_ID;
    wire [63:0] D_out_WB;
    wire [31:0] Instruction;
    wire J_ID;
    wire LW_EX;
    wire LW_ID;
    wire LW_M;
    wire LW_WB;
    wire MOVI_EX;
    wire MOVI_ID;
    wire MOVI_M;
    wire MOVI_WB;
    wire NOOP_EX;
    wire NOOP_ID;
    wire NOOP_M;
    wire [15:0] offset;
    wire [63:0] offset_EX;
    wire [63:0] offset_ID;
    wire [63:0] offset_M;
    wire [63:0] offset_WB;
    wire [63:0] ONE;
    wire [5:0] OP_CODE_ID;
    wire [63:0] PC;
    wire [63:0] PC_next;
    wire [63:0] PC_plus_one;
    wire [63:0] RF_WB_Din;
    wire [63:0] rs_data_EX;
    wire [63:0] rs_data_ID;
    wire [63:0] rs_data_M;
    wire [4:0] rs_ID;
    wire [63:0] rt_data_EX;
    wire [63:0] rt_data_ID;
    wire [63:0] rt_data_M;
    wire [4:0] rt_EX;
    wire [4:0] rt_ID;
    wire SW_EX;
    wire SW_ID;
    wire SW_M;
    wire WME_EX;
    wire WME_ID;
    wire WME_M;
    wire WRE_EX;
    wire WRE_ID;
    wire WRE_M;
    wire WRE_WB;
    wire XLXN_22;
    wire [4:0] XLXN_93;
    wire [7:0] XLXN_96;
    wire [63:0] XLXN_100;
    wire [2:0] XLXN_114;

    PC XLXI_1 (.clk(clk),
               .PC_next(PC_next[63:0]),
               .rst(rst),
               .PC(PC[63:0]));
    PC_MUX XLXI_3 (.BTA(Offset_ID[63:0]),
                   .PC_ctrl(XLXN_22),
                   .PC_next_in(PC_plus_one[63:0]),
                   .PC_next_out(PC_next[63:0]));
    I_MEM XLXI_4 (.addra(PC[8:0]),
                  .addrb(I_W_Addr[8:0]),
                  .clka(clk),
                  .clkb(clk),
                  .dinb(Instr_IN[31:0]),
                  .web(Instr_W_en),
                  .douta(Instruction[31:0]));
    IF_ID_Reg XLXI_5 (.Instruction(Instruction[31:0]),
                      .Offset_ID(Offset[15:0]),
                      .OP_CODE_ID(OP_CODE_ID[5:0]),
                      .rs_ID(rs_ID[4:0]),
                      .rt_ID(rt_ID[4:0]));
    RF XLXI_6 (.clk(clk),
               .rst(rst),
               .r0addr(rs_ID[2:0]),
               .r1addr(rt_ID[2:0]),
               .waddr(XLXN_114[2:0]),
               .wdata(RF_WB_Din[63:0]),
               .wena(WRE_WB),
               .r0data(rs_data_ID[63:0]),
               .r1data(rt_data_ID[63:0]));
    Control_Unit XLXI_7 (.OP_CODE(OP_CODE_ID[5:0]),
                         .ADDI_ID(ADDI_ID),
                         .ALU_OP_ID(ALU_OP_ID[3:0]),
                         .BEQ_ID(BEQ_ID),
                         .BGT_ID(BGT_ID),
                         .BLT_ID(BLT_ID),
                         .J_ID(J_ID),
                         .LW_ID(LW_ID),
                         .MOVI_ID(MOVI_ID),
                         .NOOP_ID(NOOP_ID),
                         .SW_ID(SW_ID),
                         .WME_ID(WME_ID),
                         .WRE_ID(WRE_ID));
    Offset_Extend XLXI_9 (.Offset(Offset[15:0]),
                          .Offset_ID(Offset_ID[63:0]));
    ID_EX_Reg XLXI_10 (.ADDI_ID(ADDI_ID),
                       .ALU_OP_ID(ALU_OP_ID[3:0]),
                       .clk(clk),
                       .LW_ID(LW_ID),
                       .MOVI_ID(MOVI_ID),
                       .NOOP_ID(NOOP_ID),
                       .Offset_ID(Offset_ID[63:0]),
                       .rst(rst),
                       .rs_data_ID(rs_data_ID[63:0]),
                       .rt_data_ID(rt_data_ID[63:0]),
                       .rt_ID(rt_ID[4:0]),
                       .SW_ID(SW_ID),
                       .WME_ID(WME_ID),
                       .WRE_ID(WRE_ID),
                       .ADDI_EX(ADDI_EX),
                       .ALU_OP_EX(ALU_OP_EX[3:0]),
                       .LW_EX(LW_EX),
                       .MOVI_EX(MOVI_EX),
                       .NOOP_EX(NOOP_EX),
                       .Offset_EX(Offset_EX[63:0]),
                       .rs_data_EX(rs_data_EX[63:0]),
                       .rt_data_EX(rt_data_EX[63:0]),
                       .rt_EX(rt_EX[4:0]),
                       .SW_EX(SW_EX),
                       .WME_EX(WME_EX),
                       .WRE_EX(WRE_EX));
    ALU XLXI_11 (.A(rs_data_EX[63:0]),
                 .ALU_OP(ALU_OP_EX[3:0]),
                 .B(ALU_B[63:0]),
                 .ALU_Out(ALU_result_EX[63:0]),
                 .Overflow(),
                 .Zero_Flag());
    ALU_src_MUX XLXI_12 (.ADDI_EX(ADDI_EX),
                         .LW_EX(LW_EX),
                         .Offset_EX(Offset_EX[63:0]),
                         .rt_data(rt_data_EX[63:0]),
                         .SW_EX(SW_EX),
                         .ALU_B(ALU_B[63:0]));
    EX_M_Reg XLXI_13 (.ADDI_EX(ADDI_EX),
                      .ALU_result_EX(ALU_result_EX[63:0]),
                      .clk(clk),
                      .LW_EX(LW_EX),
                      .MOVI_EX(MOVI_EX),
                      .NOOP_EX(NOOP_EX),
                      .Offset_EX(Offset_EX[63:0]),
                      .rst(rst),
                      .rs_data_EX(rs_data_EX[63:0]),
                      .rt_data_EX(rt_data_EX[63:0]),
                      .rt_EX(rt_EX[4:0]),
                      .SW_EX(SW_EX),
                      .WME_EX(WME_EX),
                      .WRE_EX(WRE_EX),
                      .ADDI_M(ADDI_M),
                      .ALU_result_M(ALU_result_M[63:0]),
                      .LW_M(LW_M),
                      .MOVI_M(MOVI_M),
                      .NOOP_M(NOOP_M),
                      .Offset_M(Offset_M[63:0]),
                      .rs_data_M(rs_data_M[63:0]),
                      .rt_data_M(rt_data_M[63:0]),
                      .rt_M(XLXN_93[4:0]),
                      .SW_M(SW_M),
                      .WME_M(WME_M),
                      .WRE_M(WRE_M));
    D_addr_src_MUX XLXI_14 (.ALU_result_M(ALU_result_M[63:0]),
                            .rt_M(XLXN_93[4:0]),
                            .SW_M(SW_M),
                            .D_addr(XLXN_96[7:0]));
    D_MEM XLXI_15 (.addra(XLXN_96[7:0]),
                   .addrb(XLXN_96[7:0]),
                   .clka(clk),
                   .clkb(clk),
                   .dina(rt_data_M[63:0]),
                   .wea(WME_M),
                   .doutb(XLXN_100[63:0]));
    M_WB_Reg XLXI_16 (.ADDI_M(ADDI_M),
                      .ALU_result_M(ALU_result_M[63:0]),
                      .clk(clk),
                      .D_out_M(XLXN_100[63:0]),
                      .LW_M(LW_M),
                      .MOVI_M(MOVI_M),
                      .NOOP_M(NOOP_M),
                      .Offset_M(Offset_M[63:0]),
                      .rst(rst),
                      .rs_data_M(rs_data_M[63:0]),
                      .rt_M(XLXN_93[4:0]),
                      .SW_M(SW_M),
                      .WRE_M(WRE_M),
                      .ADDI_WB(ADDI_WB),
                      .ALU_result_WB(ALU_result_WB[63:0]),
                      .D_out_WB(D_out_WB[63:0]),
                      .LW_WB(LW_WB),
                      .MOVI_WB(MOVI_WB),
                      .NOOP_WB(),
                      .Offset_WB(Offset_WB[63:0]),
                      .rs_data_WB(),
                      .rt_WB(XLXN_114[2:0]),
                      .SW_WB(),
                      .WRE_WB(WRE_WB));
    RF_WB_data_src_MUX XLXI_17 (.ADDI_WB(ADDI_WB),
                                .ALU_out_WB(ALU_result_WB[63:0]),
                                .D_out_WB(D_out_WB[63:0]),
                                .LW_WB(LW_WB),
                                .MOVI_WB(MOVI_WB),
                                .Offset_WB(Offset_WB[63:0]),
                                .RF_WB_Din(RF_WB_Din[63:0]));
    PC_plus_1 XLXI_18 (.ONE(ONE[63:0]),
                       .PC(PC[63:0]),
                       .PC_next(PC_plus_one[63:0]));
    VCC XLXI_19 (.P(ONE[0]));
    Branch_Detection_Unit XLXI_21 (.BEQ_ID(BEQ_ID),
                                   .BGT_ID(BGT_ID),
                                   .BLT_ID(BLT_ID),
                                   .J_ID(J_ID),
                                   .rs_data(rs_data_ID[63:0]),
                                   .rt_data(rt_data_ID[63:0]),
                                   .PC_ctrl(XLXN_22));
 endmodule
```

4.3 Testbench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   23:28:48 03/01/2025
// Design Name:   Pipeline
// Module Name:   E:/Documents and
Settings/student/EE533_Lab7/EE533_Lab_7/Pipeline_tb.v
// Project Name:  EE533_Lab_7
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: Pipeline
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module Pipeline_tb;

    // Inputs
    reg clk;
    reg [31:0] Instr_IN;
    reg Instr_W_en;
    reg [8:0] I_W_Addr;
    reg rst;

    // Outputs
    wire ADDI_EX;
    wire ADDI_ID;
    wire ADDI_M;
    wire ADDI_WB;
    wire [63:0] ALU_B;
    wire [3:0] ALU_OP_EX;
    wire [3:0] ALU_OP_ID;
```

```verilog
    wire [63:0] ALU_result_EX;
    wire [63:0] ALU_result_M;
    wire [63:0] ALU_result_WB;
    wire BEQ_ID;
    wire BGT_ID;
    wire BLT_ID;
    wire [7:0] D_MEM_addr;
    wire [63:0] D_out_M;
    wire [63:0] D_out_WB;
    wire [31:0] Instruction;
    wire J_ID;
    wire LW_EX;
    wire LW_ID;
    wire LW_M;
    wire LW_WB;
    wire MOVI_EX;
    wire MOVI_ID;
    wire MOVI_M;
    wire MOVI_WB;
    wire NOOP_EX;
    wire NOOP_ID;
    wire NOOP_M;
    wire NOOP_WB;
    wire [63:0] Offset_EX;
    wire [63:0] Offset_ID;
    wire [63:0] Offset_M;
    wire [63:0] Offset_WB;
    wire [5:0] OP_CODE_ID;
    wire [63:0] PC;
    wire PC_ctrl;
    wire [63:0] PC_next;
    wire [63:0] PC_plus_one;
    wire [63:0] RF_WB_Din;
    wire [63:0] rs_data_EX;
    wire [63:0] rs_data_ID;
    wire [63:0] rs_data_M;
    wire [63:0] rs_data_WB;
    wire [4:0] rs_ID;
    wire [63:0] rt_data_EX;
    wire [63:0] rt_data_ID;
    wire [63:0] rt_data_M;
    wire [4:0] rt_EX;
    wire [4:0] rt_ID;
    wire [4:0] rt_M;
    wire [2:0] rt_WB;
    wire SW_EX;
    wire SW_ID;
    wire SW_M;
    wire SW_WB;
    wire WME_EX;
    wire WME_ID;
    wire WME_M;
    wire WRE_EX;
    wire WRE_ID;
    wire WRE_M;
    wire WRE_WB;
    // Instantiate the Unit Under Test (UUT)
    Pipeline uut (
        .clk(clk),
        .Instr_IN(Instr_IN),
        .Instr_W_en(Instr_W_en),
        .I_W_Addr(I_W_Addr),
        .rst(rst),
        .ADDI_EX(ADDI_EX),
        .ADDI_ID(ADDI_ID),
        .ADDI_M(ADDI_M),
        .ADDI_WB(ADDI_WB),
        .ALU_B(ALU_B),
        .ALU_OP_EX(ALU_OP_EX),
        .ALU_OP_ID(ALU_OP_ID),
        .ALU_result_EX(ALU_result_EX),
        .ALU_result_M(ALU_result_M),
        .ALU_result_WB(ALU_result_WB),
        .BEQ_ID(BEQ_ID),
        .BGT_ID(BGT_ID),
        .BLT_ID(BLT_ID),
        .D_MEM_addr(D_MEM_addr),
        .D_out_M(D_out_M),
        .D_out_WB(D_out_WB),
        .Instruction(Instruction),
        .J_ID(J_ID),
        .LW_EX(LW_EX),
        .LW_ID(LW_ID),
        .LW_M(LW_M),
        .LW_WB(LW_WB),
        .MOVI_EX(MOVI_EX),
        .MOVI_ID(MOVI_ID),
        .MOVI_M(MOVI_M),
        .MOVI_WB(MOVI_WB),
        .NOOP_EX(NOOP_EX),
        .NOOP_ID(NOOP_ID),
        .NOOP_M(NOOP_M),
        .NOOP_WB(NOOP_WB),
        .Offset_EX(Offset_EX),
        .Offset_ID(Offset_ID),
        .Offset_M(Offset_M),
        .Offset_WB(Offset_WB),
        .OP_CODE_ID(OP_CODE_ID),
        .PC(PC),
        .PC_ctrl(PC_ctrl),
        .PC_next(PC_next),
        .PC_plus_one(PC_plus_one),
        .RF_WB_Din(RF_WB_Din),
        .rs_data_EX(rs_data_EX),
        .rs_data_ID(rs_data_ID),
        .rs_data_M(rs_data_M),
        .rs_data_WB(rs_data_WB),
        .rs_ID(rs_ID),
        .rt_data_EX(rt_data_EX),
        .rt_data_ID(rt_data_ID),
        .rt_data_M(rt_data_M),
        .rt_EX(rt_EX),
        .rt_ID(rt_ID),
        .rt_M(rt_M),
        .rt_WB(rt_WB),
        .SW_EX(SW_EX),
        .SW_ID(SW_ID),
        .SW_M(SW_M),
        .SW_WB(SW_WB),
        .WME_EX(WME_EX),
        .WME_ID(WME_ID),
        .WME_M(WME_M),
        .WRE_EX(WRE_EX),
        .WRE_ID(WRE_ID),
        .WRE_M(WRE_M),
        .WRE_WB(WRE_WB)
    );

    always #50 clk = ~clk;

    initial begin
        // Initialize Inputs
        clk = 0;
        Instr_IN = 0;
        Instr_W_en = 0;
        I_W_Addr = 0;
        rst = 1;

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

        // Add stimulus here
        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);
        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);

        @(posedge clk);
        $stop;

    end

endmodule
```