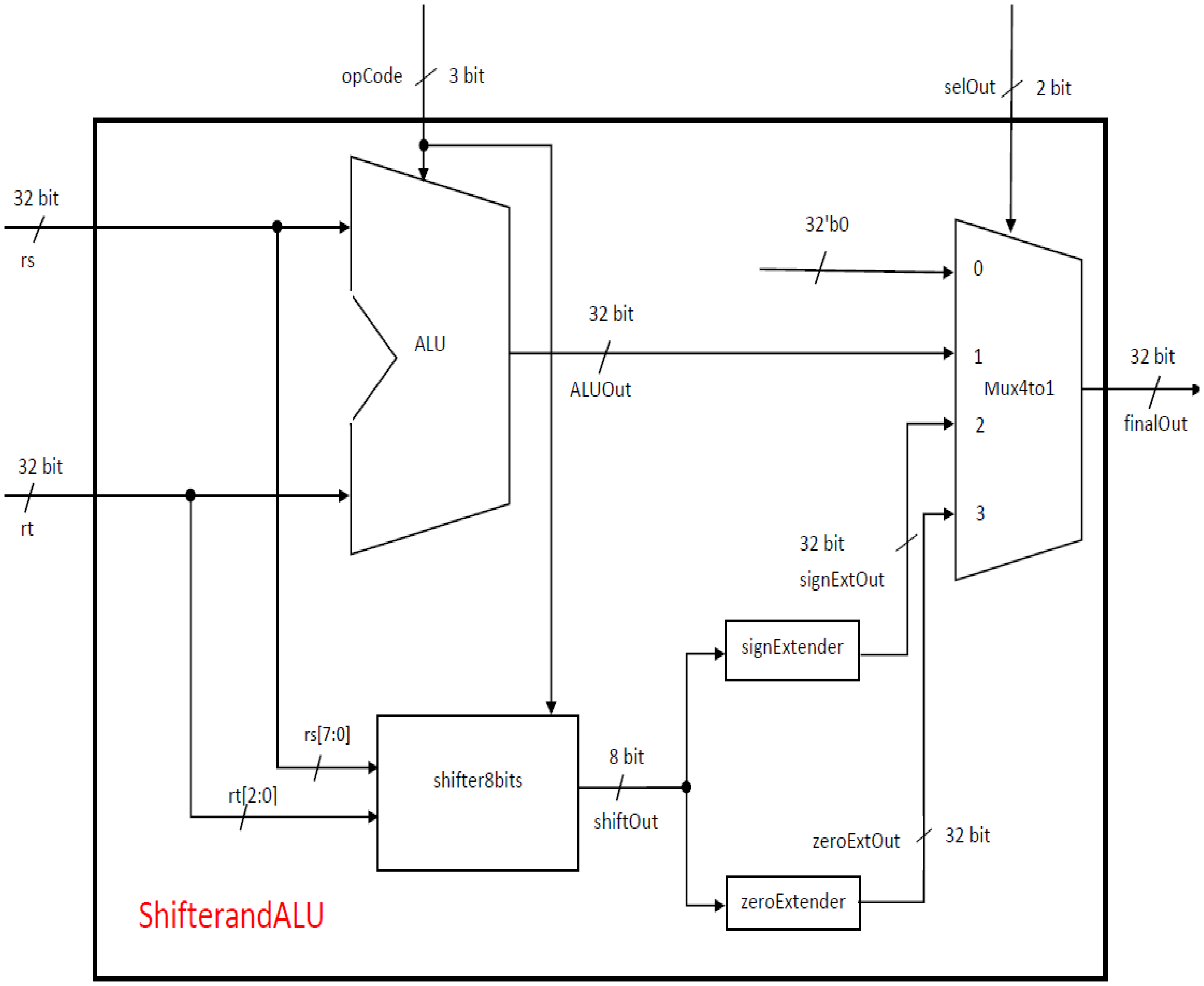
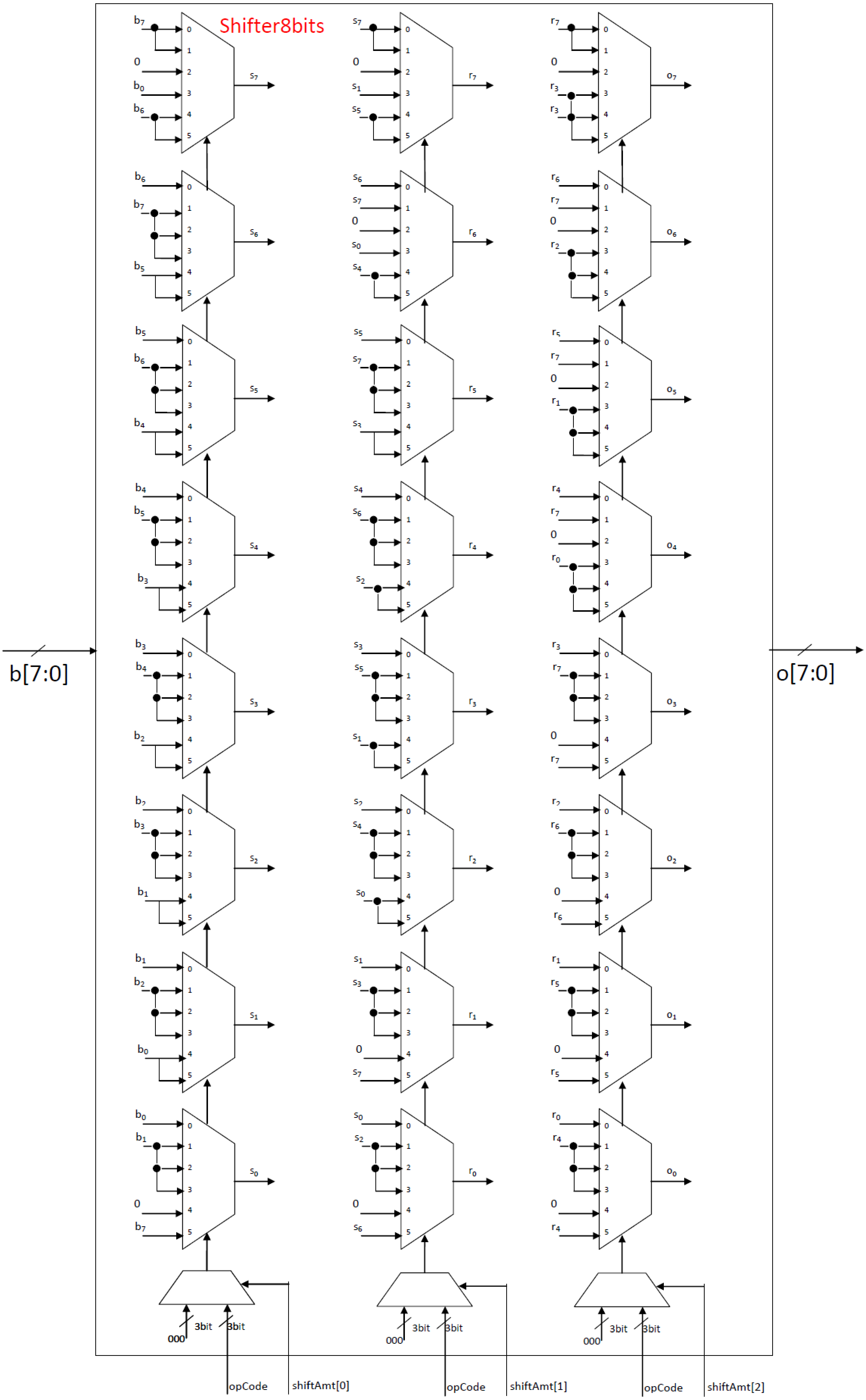
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI K. K. BIRLA Goa Campus
First Semester 2014-2015
CS F342 Computer Organization and Architecture
Lab – 2, 21th August 2014

Design and implement the following using Verilog HDL in **ModelSim software** with the following specifications.



Operations table:

Opcode	ALU operation	Shifter Operation
000	ZERO	NO SHIFT
001	ADD	Arithmetic shift right
010	SUB	Logical shift right
011	AND	Circular shift right
100	OR	Logical shift left
101	NEGATE[rs]	Circular shift left

Shifter8bits
b_7
0
b_0
b_6
s_7
b_6
b_7
b_5
s_6
b_5
b_6
b_4
s_5
b_4
b_5
b_3
s_4
b_3
b_4
b_2
s_3
b_2
b_3
b_1
s_2
b_1
b_2
b_0
s_1
b_0
b_1
0
b_7
s_0
s_7
0
s_1
s_5
r_7
s_6
s_7
0
s_0
s_4
r_6
s_5
s_7
s_3
r_5
s_4
s_6
s_2
r_4
s_3
s_5
s_1
r_3
s_2
s_4
s_0
r_2
s_1
s_3
0
s_7
r_1
s_0
s_2
0
s_6
r_0
r_7
0
r_3
r_3
o_7
r_6
r_7
0
r_2
o_6
r_5
r_7
0
r_1
o_5
r_4
r_7
0
r_0
o_4
r_3
r_7
0
r_7
o_3
r_2
r_6
0
r_6
o_2
r_1
r_5
0
r_5
o_1
r_0
r_4
0
r_4
o_0
0
1
2
3
4
5
b[7:0]
o[7:0]
000
3bit
3bit
opCode
shiftAmt[0]
000
3bit
3bit
opCode
shiftAmt[1]
000
3bit
3bit
opCode
shiftAmt[2]

Modules:

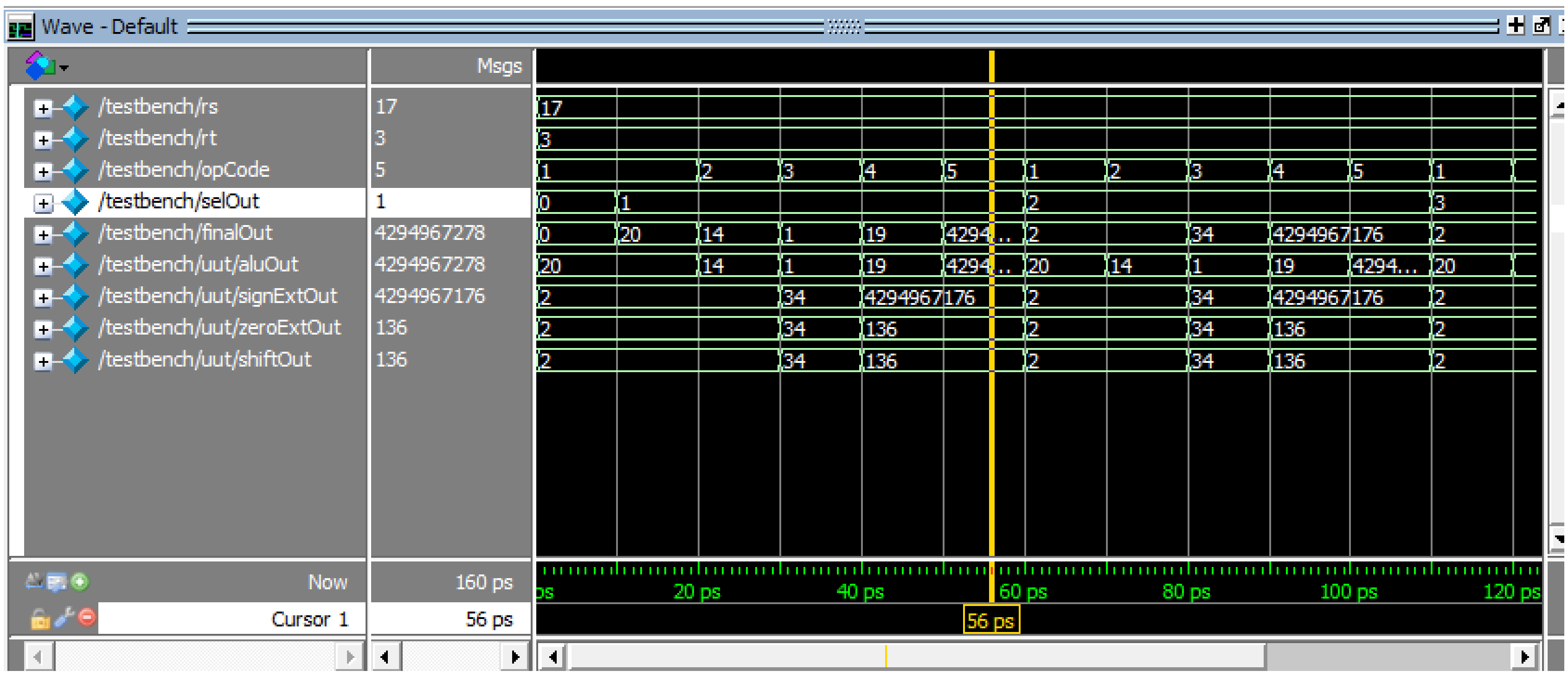
1. **module alu(input [31:0] rs, input [31:0] rt, input [2:0] opCode, output reg [31:0] aluOut);**
2. **module mux2to1(input [2:0] in0, input [2:0] in1, input sel, output reg [2:0] muxOut);**
3. **module mux8to1(input in0, input in1, input in2, input in3, input in4, input in5, input [2:0] sel, output reg muxOut);**
4. **module shifter8bits(input [7:0] rs, input [2:0] opCode, input [2:0] shiftAmt, output [7:0] shiftOut);**
5. **module signExtender(input [7:0] shiftOut, output reg [31:0] signExtOut);**
6. **module zeroExtender(input [7:0] shiftOut, output reg [31:0] zeroExtOut);**
7. **module mux4to1(input [31:0] zeroInput, input [31:0] ALUOut, input [31:0] signExtOut, input [31:0] zeroExtOut,input [1:0] selOut, output reg [31:0] finalOut);**
8. **module shifterAndAlu(input [31:0] rs, input [31:0] rt, input [2:0] opCode, input [1:0] selOut, output [31:0] finalOut);**
9. **module testBench();**

```
//inputs
reg [31:0] rs, rt;
reg [2:0] opCode;
reg [1:0] selOut;
//outputs
wire [31:0] finalOut;

shifterAndAlu uut( rs, rt, opCode, selOut, finalOut);

initial
begin rs=32'd17; rt=32'd3; opCode=3'd1; selOut=2'd0;
$monitor("rs = %d, rt = %d, opCode = %d, selOut = %d, finalOut = %d", rs,rt,opCode,selOut,finalOut);
#10    selOut=2'd1;
#10    selOut=2'd1; opCode=3'd2;
#10    selOut=2'd1; opCode=3'd3;
#10    selOut=2'd1; opCode=3'd4;
#10    selOut=2'd1; opCode=3'd5;
#10    selOut=2'd2; opCode=3'd1;
#10    selOut=2'd2; opCode=3'd2;
#10    selOut=2'd2; opCode=3'd3;
#10    selOut=2'd2; opCode=3'd4;
#10    selOut=2'd2; opCode=3'd5;
#10    selOut=2'd3; opCode=3'd1;
#10    selOut=2'd3; opCode=3'd2;
#10    selOut=2'd3; opCode=3'd3;
#10    selOut=2'd3; opCode=3'd4;
#10    selOut=2'd3; opCode=3'd5;
#10 $finish;
end
```

endmodule



Wave - Default

	Msgs
/testbench/rs	17
/testbench/rt	3
/testbench/opCode	5
/testbench/selOut	1
/testbench/finalOut	4294967278
/testbench/uut/aluOut	4294967278
/testbench/uut/signExtOut	4294967176
/testbench/uut/zeroExtOut	136
/testbench/uut/shiftOut	136
Now	160 ps
Cursor 1	56 ps