UE19CS206 – DIGITAL DESIGN & COMPUTER ORGANIZATION LABORATORY

AUGUST – DECEMBER 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DDCO Mini Project ISA Submission

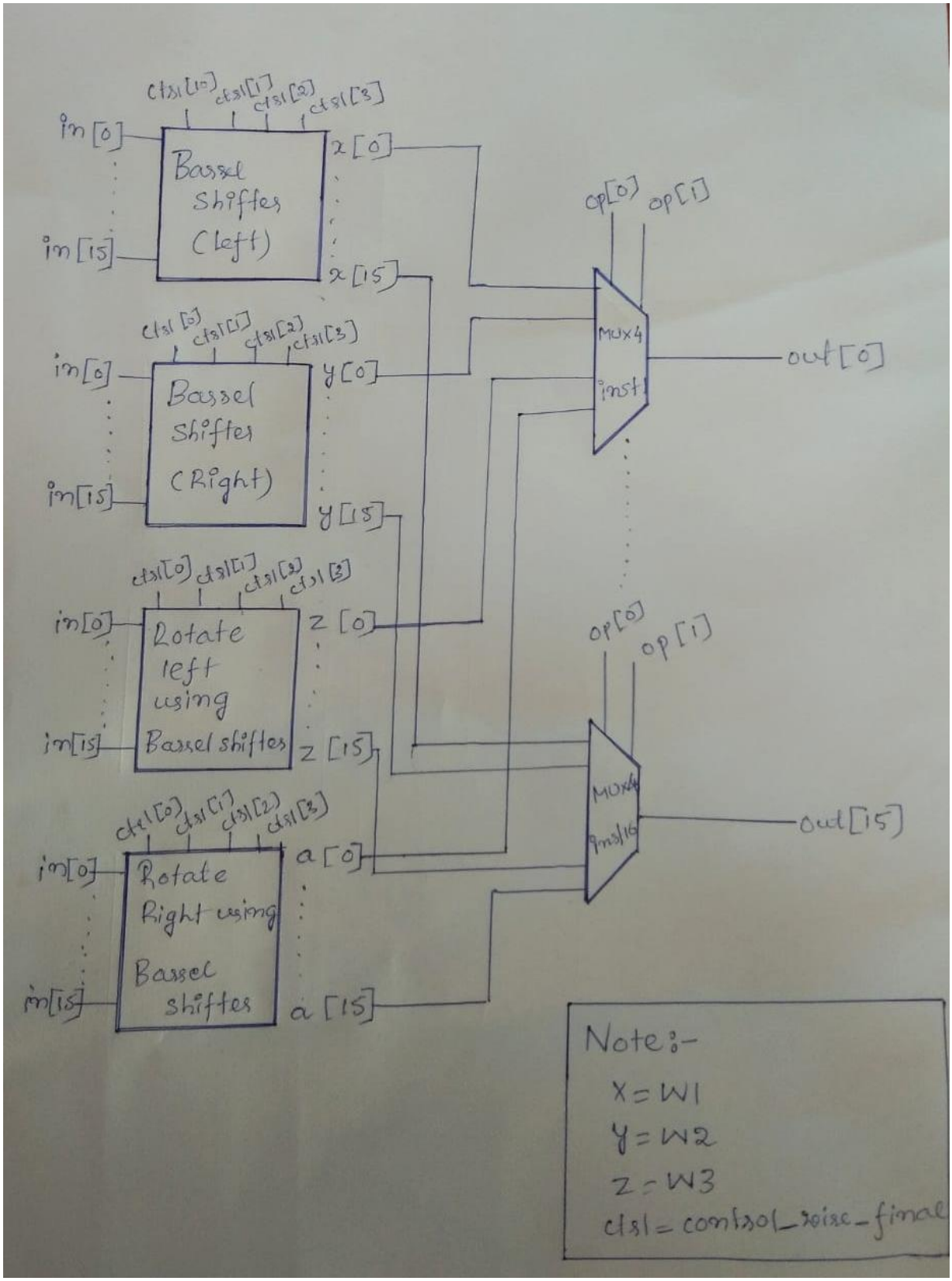
PROJECT TITLE: SHIFT-LEFT, SHIFT-RIGHT AND ROTATE OPERATIONS ON 16-BIT ALU USING MULTIPLEXER (PROBLEM 2)

SECTION: D

BATCH DETAILS:

Student Name	SRN
KISHAN	PES1UG19CS220
KISHAN MINNA NARASIMHA MURTHY	PES1UG19CS221
KK TARUNKUMAR	PES1UG19CS222
KOLLI SAI RITHWIK	PES1UG19CS223

CIRCUIT DIAGRAM:



ALGORITHM:

- The two inputs, the input to be shifted (in[15:0]) and the precise number of bits to be shifted (ctrl[3:0]), are passed through each of the four 16-bit barrel shifters.
- The left barrel shifter shifts the input in [15:0] by the specified number of ctrl [3:0] bits and gives the output w1[15:0].
- The right barrel shifter shifts the input in [15:0] by the specified number of ctrl[3:0] bits and gives the output w2[15:0]
- Correspondingly, the rotate left and rotate right barrel shifter, left rotate and right rotate the input in [15:0] by the desired number of control bits ctrl [3:0] and give the outputs, w3[15:0] and a[15:0]
- Each of the bits of these output is passed through a 4:1 mux, which gives the desired output, out [15:0] based on the two-bit operation code op [1:0]
- The operation codes for different operations are as follows

Operation code	Operation
00	Left-shift
01	Right-shift
10	Rotate-left
11	Rotate-right

IMPLEMENTATION:

```
//2X1 Mux
module mux2X1(in0,in1,sel,out);
input in0,in1;
input sel;
output out;
assign out=(sel)?in1:in0;
endmodule

module mux4 (i0,i1,i2,i3, j0,j1,o);
 input i0,i1,i2,i3;
 input j0,j1;
 output o;
  wire t0, t1;
 mux2X1 mux2_0 (i0, i1, j0, t0);
 mux2X1 mux2_1 (i2, i3, j0, t1);
 mux2X1 mux2_2(t0,t1,j1,o);
endmodule

module barrel_shifter_16bit_left (input wire [15:0] in, input wire [3:0] control_wire_final, output wire [15:0] out);
        wire [0:15] w1,w2,w3;
        //8bit shift
        mux2X1 ins_1 (.in0(in[0]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[0]));
        mux2X1 ins_2 (.in0(in[1]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[1]));
        mux2X1 ins_3 (.in0(in[2]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[2]));
        mux2X1 ins_4 (.in0(in[3]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[3]));
        mux2X1 ins_5 (.in0(in[4]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[4]));
```

```
mux2X1 ins_6 (.in0(in[5]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[5]));
mux2X1 ins_7 (.in0(in[6]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[6]));
mux2X1 ins_8 (.in0(in[7]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[7]));
mux2X1 ins_9 (.in0(in[8]),.in1(in[0]),.sel(control_wire_final[3]),.out(w3[8]));
mux2X1 ins_10 (.in0(in[9]),.in1(in[1]),.sel(control_wire_final[3]),.out(w3[9]));
mux2X1 ins_11 (.in0(in[10]),.in1(in[2]),.sel(control_wire_final[3]),.out(w3[10]));
mux2X1 ins_12 (.in0(in[11]),.in1(in[3]),.sel(control_wire_final[3]),.out(w3[11]));
mux2X1 ins_13 (.in0(in[12]),.in1(in[4]),.sel(control_wire_final[3]),.out(w3[12]));
mux2X1 ins_14 (.in0(in[13]),.in1(in[5]),.sel(control_wire_final[3]),.out(w3[13]));
mux2X1 ins_15 (.in0(in[14]),.in1(in[6]),.sel(control_wire_final[3]),.out(w3[14]));
mux2X1 ins_16 (.in0(in[15]),.in1(in[7]),.sel(control_wire_final[3]),.out(w3[15]));

//4bit shift left
mux2X1 ins_17 (.in0(w3[0]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[0]));
mux2X1 ins_18(.in0(w3[1]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[1]));
mux2X1 ins_19 (.in0(w3[2]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[2]));
mux2X1 ins_20 (.in0(w3[3]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[3]));
mux2X1 ins_21 (.in0(w3[4]),.in1(w3[0]),.sel(control_wire_final[2]),.out(w1[4]));
mux2X1 ins_22 (.in0(w3[5]),.in1(w3[1]),.sel(control_wire_final[2]),.out(w1[5]));
mux2X1 ins_23 (.in0(w3[6]),.in1(w3[2]),.sel(control_wire_final[2]),.out(w1[6]));
mux2X1 ins_24 (.in0(w3[7]),.in1(w3[3]),.sel(control_wire_final[2]),.out(w1[7]));
mux2X1 ins_25 (.in0(w3[8]),.in1(w3[4]),.sel(control_wire_final[2]),.out(w1[8]));
mux2X1 ins_26 (.in0(w3[9]),.in1(w3[5]),.sel(control_wire_final[2]),.out(w1[9]));
mux2X1 ins_27 (.in0(w3[10]),.in1(w3[6]),.sel(control_wire_final[2]),.out(w1[10]));
mux2X1 ins_28 (.in0(w3[11]),.in1(w3[7]),.sel(control_wire_final[2]),.out(w1[11]));
mux2X1 ins_29 (.in0(w3[12]),.in1(w3[8]),.sel(control_wire_final[2]),.out(w1[12]));
mux2X1 ins_30 (.in0(w3[13]),.in1(w3[9]),.sel(control_wire_final[2]),.out(w1[13]));
mux2X1 ins_31 (.in0(w3[14]),.in1(w3[10]),.sel(control_wire_final[2]),.out(w1[14]));
```

```
mux2X1 ins_32 (.in0(w3[15]),.in1(w3[11]),.sel(control_wire_final[2]),.out(w1[15]));
//2 bit shift left
mux2X1 ins_33 (.in0(w1[0]),.in1(1'b0),.sel(control_wire_final[1]),.out(w2[0]));
mux2X1 ins_34 (.in0(w1[1]),.in1(1'b0),.sel(control_wire_final[1]),.out(w2[1]));
mux2X1 ins_35 (.in0(w1[2]),.in1(w1[0]),.sel(control_wire_final[1]),.out(w2[2]));
mux2X1 ins_36 (.in0(w1[3]),.in1(w1[1]),.sel(control_wire_final[1]),.out(w2[3]));
mux2X1 ins_37 (.in0(w1[4]),.in1(w1[2]),.sel(control_wire_final[1]),.out(w2[4]));
mux2X1 ins_38 (.in0(w1[5]),.in1(w1[3]),.sel(control_wire_final[1]),.out(w2[5]));
mux2X1 ins_39 (.in0(w1[6]),.in1(w1[4]),.sel(control_wire_final[1]),.out(w2[6]));
mux2X1 ins_40 (.in0(w1[7]),.in1(w1[5]),.sel(control_wire_final[1]),.out(w2[7]));
mux2X1 ins_41 (.in0(w1[8]),.in1(w1[6]),.sel(control_wire_final[1]),.out(w2[8]));
mux2X1 ins_42 (.in0(w1[9]),.in1(w1[7]),.sel(control_wire_final[1]),.out(w2[9]));
mux2X1 ins_43 (.in0(w1[10]),.in1(w1[8]),.sel(control_wire_final[1]),.out(w2[10]));
mux2X1 ins_44 (.in0(w1[11]),.in1(w1[9]),.sel(control_wire_final[1]),.out(w2[11]));
mux2X1 ins_45 (.in0(w1[12]),.in1(w1[10]),.sel(control_wire_final[1]),.out(w2[12]));
mux2X1 ins_46 (.in0(w1[13]),.in1(w1[11]),.sel(control_wire_final[1]),.out(w2[13]));
mux2X1 ins_47 (.in0(w1[14]),.in1(w1[12]),.sel(control_wire_final[1]),.out(w2[14]));
mux2X1 ins_48 (.in0(w1[15]),.in1(w1[13]),.sel(control_wire_final[1]),.out(w2[15]));
//1 bit shift left
mux2X1 ins_49 (.in0(w2[0]),.in1(1'b0),.sel(control_wire_final[0]),.out(out[0]));
mux2X1 ins_50 (.in0(w2[1]),.in1(w2[0]),.sel(control_wire_final[0]),.out(out[1]));
mux2X1 ins_51 (.in0(w2[2]),.in1(w2[1]),.sel(control_wire_final[0]),.out(out[2]));
mux2X1 ins_52 (.in0(w2[3]),.in1(w2[2]),.sel(control_wire_final[0]),.out(out[3]));
mux2X1 ins_53 (.in0(w2[4]),.in1(w2[3]),.sel(control_wire_final[0]),.out(out[4]));
mux2X1 ins_54 (.in0(w2[5]),.in1(w2[4]),.sel(control_wire_final[0]),.out(out[5]));
mux2X1 ins_55 (.in0(w2[6]),.in1(w2[5]),.sel(control_wire_final[0]),.out(out[6]));
mux2X1 ins_56 (.in0(w2[7]),.in1(w2[6]),.sel(control_wire_final[0]),.out(out[7]));
mux2X1 ins_57 (.in0(w2[8]),.in1(w2[7]),.sel(control_wire_final[0]),.out(out[8]));
```

```
        mux2X1 ins_58 (.in0(w2[9]),.in1(w2[8]),.sel(control_wire_final[0]),.out(out[9]));
        mux2X1 ins_59 (.in0(w2[10]),.in1(w2[9]),.sel(control_wire_final[0]),.out(out[10]));
        mux2X1 ins_60 (.in0(w2[11]),.in1(w2[10]),.sel(control_wire_final[0]),.out(out[11]));
        mux2X1 ins_61 (.in0(w2[12]),.in1(w2[11]),.sel(control_wire_final[0]),.out(out[12]));
        mux2X1 ins_62 (.in0(w2[13]),.in1(w2[12]),.sel(control_wire_final[0]),.out(out[13]));
        mux2X1 ins_63 (.in0(w2[14]),.in1(w2[13]),.sel(control_wire_final[0]),.out(out[14]));
        mux2X1 ins_64 (.in0(w2[15]),.in1(w2[14]),.sel(control_wire_final[0]),.out(out[15]));
endmodule

module barrel_shifter_16bit_right (input wire [15:0] in, input wire [3:0] control_wire_final, output
wire [15:0]
out);
        wire [15:0] w1,w2,w3;
        //8bit shift right
        mux2X1 ins_1 (.in0(in[15]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[15]));
        mux2X1 ins_2 (.in0(in[14]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[14]));
        mux2X1 ins_3 (.in0(in[13]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[13]));
        mux2X1 ins_4 (.in0(in[12]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[12]));
        mux2X1 ins_5 (.in0(in[11]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[11]));
        mux2X1 ins_6 (.in0(in[10]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[10]));
        mux2X1 ins_7 (.in0(in[9]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[9]));
        mux2X1 ins_8 (.in0(in[8]),.in1(1'b0),.sel(control_wire_final[3]),.out(w3[8]));
        mux2X1 ins_9 (.in0(in[7]),.in1(in[15]),.sel(control_wire_final[3]),.out(w3[7]));
        mux2X1 ins_10 (.in0(in[6]),.in1(in[14]),.sel(control_wire_final[3]),.out(w3[6]));
        mux2X1 ins_11 (.in0(in[5]),.in1(in[13]),.sel(control_wire_final[3]),.out(w3[5]));
        mux2X1 ins_12 (.in0(in[4]),.in1(in[12]),.sel(control_wire_final[3]),.out(w3[4]));
        mux2X1 ins_13 (.in0(in[3]),.in1(in[11]),.sel(control_wire_final[3]),.out(w3[3]));
        mux2X1 ins_14 (.in0(in[2]),.in1(in[10]),.sel(control_wire_final[3]),.out(w3[2]));
        mux2X1 ins_15 (.in0(in[1]),.in1(in[9]),.sel(control_wire_final[3]),.out(w3[1]));
```

```
mux2X1 ins_16 (.in0(in[0]),.in1(in[8]),.sel(control_wire_final[3]),.out(w3[0]));
//4bit shift right
mux2X1 ins_17 (.in0(w3[15]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[15]));
mux2X1 ins_18 (.in0(w3[14]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[14]));
mux2X1 ins_19 (.in0(w3[13]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[13]));
mux2X1 ins_20 (.in0(w3[12]),.in1(1'b0),.sel(control_wire_final[2]),.out(w1[12]));
mux2X1 ins_21 (.in0(w3[11]),.in1(w3[15]),.sel(control_wire_final[2]),.out(w1[11]));
mux2X1 ins_22 (.in0(w3[10]),.in1(w3[14]),.sel(control_wire_final[2]),.out(w1[10]));
mux2X1 ins_23 (.in0(w3[9]),.in1(w3[13]),.sel(control_wire_final[2]),.out(w1[9]));
mux2X1 ins_24 (.in0(w3[8]),.in1(w3[12]),.sel(control_wire_final[2]),.out(w1[8]));
mux2X1 ins_25 (.in0(w3[7]),.in1(w3[11]),.sel(control_wire_final[2]),.out(w1[7]));
mux2X1 ins_26 (.in0(w3[6]),.in1(w3[10]),.sel(control_wire_final[2]),.out(w1[6]));
mux2X1 ins_27 (.in0(w3[5]),.in1(w3[9]),.sel(control_wire_final[2]),.out(w1[5]));
mux2X1 ins_28 (.in0(w3[4]),.in1(w3[8]),.sel(control_wire_final[2]),.out(w1[4]));
mux2X1 ins_29 (.in0(w3[3]),.in1(in[7]),.sel(control_wire_final[2]),.out(w1[3]));
mux2X1 ins_30 (.in0(w3[2]),.in1(in[6]),.sel(control_wire_final[2]),.out(w1[2]));
mux2X1 ins_31 (.in0(w3[1]),.in1(in[5]),.sel(control_wire_final[2]),.out(w1[1]));
mux2X1 ins_32 (.in0(w3[0]),.in1(in[4]),.sel(control_wire_final[2]),.out(w1[0]));
//2 bit shift right
mux2X1 ins_33 (.in0(w1[15]),.in1(1'b0),.sel(control_wire_final[1]),.out(w2[15]));
mux2X1 ins_34 (.in0(w1[14]),.in1(1'b0),.sel(control_wire_final[1]),.out(w2[14]));
mux2X1 ins_35 (.in0(w1[13]),.in1(w1[15]),.sel(control_wire_final[1]),.out(w2[13]));
mux2X1 ins_36 (.in0(w1[12]),.in1(w1[14]),.sel(control_wire_final[1]),.out(w2[12]));
mux2X1 ins_37 (.in0(w1[11]),.in1(w1[13]),.sel(control_wire_final[1]),.out(w2[11]));
mux2X1 ins_38 (.in0(w1[10]),.in1(w1[12]),.sel(control_wire_final[1]),.out(w2[10]));
mux2X1 ins_39 (.in0(w1[9]),.in1(w1[11]),.sel(control_wire_final[1]),.out(w2[9]));
mux2X1 ins_40 (.in0(w1[8]),.in1(w1[10]),.sel(control_wire_final[1]),.out(w2[8]));
mux2X1 ins_41 (.in0(w1[7]),.in1(w1[9]),.sel(control_wire_final[1]),.out(w2[7]));
```

```
	mux2X1 ins_42 (.in0(w1[6]),.in1(w1[8]),.sel(control_wire_final[1]),.out(w2[6]));
	mux2X1 ins_43 (.in0(w1[5]),.in1(w1[7]),.sel(control_wire_final[1]),.out(w2[5]));
	mux2X1 ins_44 (.in0(w1[4]),.in1(w1[6]),.sel(control_wire_final[1]),.out(w2[4]));
	mux2X1 ins_45 (.in0(w1[3]),.in1(w1[5]),.sel(control_wire_final[1]),.out(w2[3]));
	mux2X1 ins_46 (.in0(w1[2]),.in1(w1[4]),.sel(control_wire_final[1]),.out(w2[2]));
	mux2X1 ins_47 (.in0(w1[1]),.in1(w1[3]),.sel(control_wire_final[1]),.out(w2[1]));
	mux2X1 ins_48 (.in0(w1[0]),.in1(w1[2]),.sel(control_wire_final[1]),.out(w2[0]));
	//1 bit shift right
	mux2X1 ins_49 (.in0(w2[15]),.in1(1'b0),.sel(control_wire_final[0]),.out(out[15]));
	mux2X1 ins_50 (.in0(w2[14]),.in1(w2[15]),.sel(control_wire_final[0]),.out(out[14]));
	mux2X1 ins_51 (.in0(w2[13]),.in1(w2[14]),.sel(control_wire_final[0]),.out(out[13]));
	mux2X1 ins_52 (.in0(w2[12]),.in1(w2[13]),.sel(control_wire_final[0]),.out(out[12]));
	mux2X1 ins_53 (.in0(w2[11]),.in1(w2[12]),.sel(control_wire_final[0]),.out(out[11]));
	mux2X1 ins_54 (.in0(w2[10]),.in1(w2[11]),.sel(control_wire_final[0]),.out(out[10]));
	mux2X1 ins_55 (.in0(w2[9]),.in1(w2[10]),.sel(control_wire_final[0]),.out(out[9]));
	mux2X1 ins_56 (.in0(w2[8]),.in1(w2[9]),.sel(control_wire_final[0]),.out(out[8]));
	mux2X1 ins_57 (.in0(w2[7]),.in1(w2[8]),.sel(control_wire_final[0]),.out(out[7]));
	mux2X1 ins_58 (.in0(w2[6]),.in1(w2[7]),.sel(control_wire_final[0]),.out(out[6]));
	mux2X1 ins_59 (.in0(w2[5]),.in1(w2[6]),.sel(control_wire_final[0]),.out(out[5]));
	mux2X1 ins_60 (.in0(w2[4]),.in1(w2[5]),.sel(control_wire_final[0]),.out(out[4]));
	mux2X1 ins_61 (.in0(w2[3]),.in1(w2[4]),.sel(control_wire_final[0]),.out(out[3]));
	mux2X1 ins_62 (.in0(w2[2]),.in1(w2[3]),.sel(control_wire_final[0]),.out(out[2]));
	mux2X1 ins_63 (.in0(w2[1]),.in1(w2[2]),.sel(control_wire_final[0]),.out(out[1]));
	mux2X1 ins_64 (.in0(w2[0]),.in1(w2[1]),.sel(control_wire_final[0]),.out(out[0]));
endmodule

module rotator_16bit_left (input wire [15:0] in,input wire [3:0] control_wire_final,output wire [15:0]
out);
	wire [0:15] w1,w2,w3;
```

```
//8bit rotate
mux2X1 ins_1 (.in0(in[0]),.in1(in[8]),.sel(control_wire_final[3]),.out(w3[0]));
mux2X1 ins_2 (.in0(in[1]),.in1(in[9]),.sel(control_wire_final[3]),.out(w3[1]));
mux2X1 ins_3 (.in0(in[2]),.in1(in[10]),.sel(control_wire_final[3]),.out(w3[2]));
mux2X1 ins_4 (.in0(in[3]),.in1(in[11]),.sel(control_wire_final[3]),.out(w3[3]));
mux2X1 ins_5 (.in0(in[4]),.in1(in[12]),.sel(control_wire_final[3]),.out(w3[4]));
mux2X1 ins_6 (.in0(in[5]),.in1(in[13]),.sel(control_wire_final[3]),.out(w3[5]));
mux2X1 ins_7 (.in0(in[6]),.in1(in[14]),.sel(control_wire_final[3]),.out(w3[6]));
mux2X1 ins_8 (.in0(in[7]),.in1(in[15]),.sel(control_wire_final[3]),.out(w3[7]));
mux2X1 ins_9 (.in0(in[8]),.in1(in[16]),.sel(control_wire_final[3]),.out(w3[8]));
mux2X1 ins_10 (.in0(in[9]),.in1(in[0]),.sel(control_wire_final[3]),.out(w3[9]));
mux2X1 ins_11 (.in0(in[10]),.in1(in[1]),.sel(control_wire_final[3]),.out(w3[10]));
mux2X1 ins_12 (.in0(in[11]),.in1(in[3]),.sel(control_wire_final[3]),.out(w3[11]));
mux2X1 ins_13 (.in0(in[12]),.in1(in[4]),.sel(control_wire_final[3]),.out(w3[12]));
mux2X1 ins_14 (.in0(in[13]),.in1(in[5]),.sel(control_wire_final[3]),.out(w3[13]));
mux2X1 ins_15 (.in0(in[14]),.in1(in[6]),.sel(control_wire_final[3]),.out(w3[14]));
mux2X1 ins_16 (.in0(in[15]),.in1(in[7]),.sel(control_wire_final[3]),.out(w3[15]));
//4bit rotate
mux2X1 ins_17 (.in0(w3[0]),.in1(w3[12]),.sel(control_wire_final[2]),.out(w1[0]));
mux2X1 ins_18 (.in0(w3[1]),.in1(w3[13]),.sel(control_wire_final[2]),.out(w1[1]));
mux2X1 ins_19 (.in0(w3[2]),.in1(w3[14]),.sel(control_wire_final[2]),.out(w1[2]));
mux2X1 ins_20 (.in0(w3[3]),.in1(w3[15]),.sel(control_wire_final[2]),.out(w1[3]));
mux2X1 ins_21 (.in0(w3[4]),.in1(w3[0]),.sel(control_wire_final[2]),.out(w1[4]));
mux2X1 ins_22 (.in0(w3[5]),.in1(w3[1]),.sel(control_wire_final[2]),.out(w1[5]));
mux2X1 ins_23 (.in0(w3[6]),.in1(w3[2]),.sel(control_wire_final[2]),.out(w1[6]));
mux2X1 ins_24 (.in0(w3[7]),.in1(w3[3]),.sel(control_wire_final[2]),.out(w1[7]));
mux2X1 ins_25 (.in0(w3[8]),.in1(w3[4]),.sel(control_wire_final[2]),.out(w1[8]));
mux2X1 ins_26 (.in0(w3[9]),.in1(w3[5]),.sel(control_wire_final[2]),.out(w1[9]));
```

```
mux2X1 ins_27 (.in0(w3[10]),.in1(w3[6]),.sel(control_wire_final[2]),.out(w1[10]));
mux2X1 ins_28 (.in0(w3[11]),.in1(w3[7]),.sel(control_wire_final[2]),.out(w1[11]));
mux2X1 ins_29 (.in0(w3[12]),.in1(w3[8]),.sel(control_wire_final[2]),.out(w1[12]));
mux2X1 ins_30 (.in0(w3[13]),.in1(w3[9]),.sel(control_wire_final[2]),.out(w1[13]));
mux2X1 ins_31 (.in0(w3[14]),.in1(w3[10]),.sel(control_wire_final[2]),.out(w1[14]));
mux2X1 ins_32 (.in0(w3[15]),.in1(w3[11]),.sel(control_wire_final[2]),.out(w1[15]));
//2 bit rotate
mux2X1 ins_33 (.in0(w1[0]),.in1(w1[14]),.sel(control_wire_final[1]),.out(w2[0]));
mux2X1 ins_34 (.in0(w1[1]),.in1(w1[15]),.sel(control_wire_final[1]),.out(w2[1]));
mux2X1 ins_35 (.in0(w1[2]),.in1(w1[0]),.sel(control_wire_final[1]),.out(w2[2]));
mux2X1 ins_36 (.in0(w1[3]),.in1(w1[1]),.sel(control_wire_final[1]),.out(w2[3]));
mux2X1 ins_37 (.in0(w1[4]),.in1(w1[2]),.sel(control_wire_final[1]),.out(w2[4]));
mux2X1 ins_38 (.in0(w1[5]),.in1(w1[3]),.sel(control_wire_final[1]),.out(w2[5]));
mux2X1 ins_39 (.in0(w1[6]),.in1(w1[4]),.sel(control_wire_final[1]),.out(w2[6]));
mux2X1 ins_40 (.in0(w1[7]),.in1(w1[5]),.sel(control_wire_final[1]),.out(w2[7]));
mux2X1 ins_41 (.in0(w1[8]),.in1(w1[6]),.sel(control_wire_final[1]),.out(w2[8]));
mux2X1 ins_42 (.in0(w1[9]),.in1(w1[7]),.sel(control_wire_final[1]),.out(w2[9]));
mux2X1 ins_43 (.in0(w1[10]),.in1(w1[8]),.sel(control_wire_final[1]),.out(w2[10]));
mux2X1 ins_44 (.in0(w1[11]),.in1(w1[9]),.sel(control_wire_final[1]),.out(w2[11]));
mux2X1 ins_45 (.in0(w1[12]),.in1(w1[10]),.sel(control_wire_final[1]),.out(w2[12]));
mux2X1 ins_46 (.in0(w1[13]),.in1(w1[11]),.sel(control_wire_final[1]),.out(w2[13]));
mux2X1 ins_47 (.in0(w1[14]),.in1(w1[12]),.sel(control_wire_final[1]),.out(w2[14]));
mux2X1 ins_48 (.in0(w1[15]),.in1(w1[13]),.sel(control_wire_final[1]),.out(w2[15]));
//1 bit rotate
mux2X1 ins_49 (.in0(w2[0]),.in1(w2[15]),.sel(control_wire_final[0]),.out(out[0]));
mux2X1 ins_50 (.in0(w2[1]),.in1(w2[0]),.sel(control_wire_final[0]),.out(out[1]));
mux2X1 ins_51 (.in0(w2[2]),.in1(w2[1]),.sel(control_wire_final[0]),.out(out[2]));
mux2X1 ins_52 (.in0(w2[3]),.in1(w2[2]),.sel(control_wire_final[0]),.out(out[3]));
```

```
        mux2X1 ins_53 (.in0(w2[4]),.in1(w2[3]),.sel(control_wire_final[0]),.out(out[4]));
        mux2X1 ins_54 (.in0(w2[5]),.in1(w2[4]),.sel(control_wire_final[0]),.out(out[5]));
        mux2X1 ins_55 (.in0(w2[6]),.in1(w2[5]),.sel(control_wire_final[0]),.out(out[6]));
        mux2X1 ins_56 (.in0(w2[7]),.in1(w2[6]),.sel(control_wire_final[0]),.out(out[7]));
        mux2X1 ins_57 (.in0(w2[8]),.in1(w2[7]),.sel(control_wire_final[0]),.out(out[8]));
        mux2X1 ins_58 (.in0(w2[9]),.in1(w2[8]),.sel(control_wire_final[0]),.out(out[9]));
        mux2X1 ins_59 (.in0(w2[10]),.in1(w2[9]),.sel(control_wire_final[0]),.out(out[10]));
        mux2X1 ins_60 (.in0(w2[11]),.in1(w2[10]),.sel(control_wire_final[0]),.out(out[11]));
        mux2X1 ins_61 (.in0(w2[12]),.in1(w2[11]),.sel(control_wire_final[0]),.out(out[12]));
        mux2X1 ins_62 (.in0(w2[13]),.in1(w2[12]),.sel(control_wire_final[0]),.out(out[13]));
        mux2X1 ins_63 (.in0(w2[14]),.in1(w2[13]),.sel(control_wire_final[0]),.out(out[14]));
        mux2X1 ins_64 (.in0(w2[15]),.in1(w2[14]),.sel(control_wire_final[0]),.out(out[15]));
endmodule

module rotator_16bit_right (input wire [15:0] in,input wire [3:0] control_wire_final,output wire [15:0]
out);
        wire [15:0] w1,w2,w3;
        //8bit shift right
        mux2X1 ins_1 (.in0(in[15]),.in1(in[7]),.sel(control_wire_final[3]),.out(w3[15]));
        mux2X1 ins_2 (.in0(in[14]),.in1(in[6]),.sel(control_wire_final[3]),.out(w3[14]));
        mux2X1 ins_3 (.in0(in[13]),.in1(in[5]),.sel(control_wire_final[3]),.out(w3[13]));
        mux2X1 ins_4 (.in0(in[12]),.in1(in[4]),.sel(control_wire_final[3]),.out(w3[12]));
        mux2X1 ins_5 (.in0(in[11]),.in1(in[3]),.sel(control_wire_final[3]),.out(w3[11]));
        mux2X1 ins_6 (.in0(in[10]),.in1(in[2]),.sel(control_wire_final[3]),.out(w3[10]));
        mux2X1 ins_7 (.in0(in[9]),.in1(in[1]),.sel(control_wire_final[3]),.out(w3[9]));
        mux2X1 ins_8 (.in0(in[8]),.in1(in[0]),.sel(control_wire_final[3]),.out(w3[8]));
        mux2X1 ins_9 (.in0(in[7]),.in1(in[15]),.sel(control_wire_final[3]),.out(w3[7]));
        mux2X1 ins_10 (.in0(in[6]),.in1(in[14]),.sel(control_wire_final[3]),.out(w3[6]));
        mux2X1 ins_11 (.in0(in[5]),.in1(in[13]),.sel(control_wire_final[3]),.out(w3[5]));
```

```
mux2X1 ins_12 (.in0(in[4]),.in1(in[12]),.sel(control_wire_final[3]),.out(w3[4]));
mux2X1 ins_13 (.in0(in[3]),.in1(in[11]),.sel(control_wire_final[3]),.out(w3[3]));
mux2X1 ins_14 (.in0(in[2]),.in1(in[10]),.sel(control_wire_final[3]),.out(w3[2]));
mux2X1 ins_15 (.in0(in[1]),.in1(in[9]),.sel(control_wire_final[3]),.out(w3[1]));
mux2X1 ins_16 (.in0(in[0]),.in1(in[8]),.sel(control_wire_final[3]),.out(w3[0]));
//4bit shift right
mux2X1 ins_17 (.in0(w3[15]),.in1(w3[3]),.sel(control_wire_final[2]),.out(w1[15]));
mux2X1 ins_18 (.in0(w3[14]),.in1(w3[2]),.sel(control_wire_final[2]),.out(w1[14]));
mux2X1 ins_19 (.in0(w3[13]),.in1(w3[1]),.sel(control_wire_final[2]),.out(w1[13]));
mux2X1 ins_20 (.in0(w3[12]),.in1(w3[0]),.sel(control_wire_final[2]),.out(w1[12]));
mux2X1 ins_21 (.in0(w3[11]),.in1(w3[15]),.sel(control_wire_final[2]),.out(w1[11]));
mux2X1 ins_22 (.in0(w3[10]),.in1(w3[14]),.sel(control_wire_final[2]),.out(w1[10]));
mux2X1 ins_23 (.in0(w3[9]),.in1(w3[13]),.sel(control_wire_final[2]),.out(w1[9]));
mux2X1 ins_24 (.in0(w3[8]),.in1(w3[12]),.sel(control_wire_final[2]),.out(w1[8]));
mux2X1 ins_25 (.in0(w3[7]),.in1(w3[11]),.sel(control_wire_final[2]),.out(w1[7]));
mux2X1 ins_26 (.in0(w3[6]),.in1(w3[10]),.sel(control_wire_final[2]),.out(w1[6]));
mux2X1 ins_27 (.in0(w3[5]),.in1(w3[9]),.sel(control_wire_final[2]),.out(w1[5]));
mux2X1 ins_28 (.in0(w3[4]),.in1(w3[8]),.sel(control_wire_final[2]),.out(w1[4]));
mux2X1 ins_29 (.in0(w3[3]),.in1(in[7]),.sel(control_wire_final[2]),.out(w1[3]));
mux2X1 ins_30 (.in0(w3[2]),.in1(in[6]),.sel(control_wire_final[2]),.out(w1[2]));
mux2X1 ins_31 (.in0(w3[1]),.in1(in[5]),.sel(control_wire_final[2]),.out(w1[1]));
mux2X1 ins_32 (.in0(w3[0]),.in1(in[4]),.sel(control_wire_final[2]),.out(w1[0]));
//2 bit shift right
mux2X1 ins_33 (.in0(w1[15]),.in1(w1[1]),.sel(control_wire_final[1]),.out(w2[15]));
mux2X1 ins_34 (.in0(w1[14]),.in1(w1[0]),.sel(control_wire_final[1]),.out(w2[14]));
mux2X1 ins_35 (.in0(w1[13]),.in1(w1[15]),.sel(control_wire_final[1]),.out(w2[13]));
mux2X1 ins_36 (.in0(w1[12]),.in1(w1[14]),.sel(control_wire_final[1]),.out(w2[12]));
mux2X1 ins_37 (.in0(w1[11]),.in1(w1[13]),.sel(control_wire_final[1]),.out(w2[11]));
```

```
mux2X1 ins_38 (.in0(w1[10]),.in1(w1[12]),.sel(control_wire_final[1]),.out(w2[10]));
mux2X1 ins_39 (.in0(w1[9]),.in1(w1[11]),.sel(control_wire_final[1]),.out(w2[9]));
mux2X1 ins_40 (.in0(w1[8]),.in1(w1[10]),.sel(control_wire_final[1]),.out(w2[8]));
mux2X1 ins_41 (.in0(w1[7]),.in1(w1[9]),.sel(control_wire_final[1]),.out(w2[7]));
mux2X1 ins_42 (.in0(w1[6]),.in1(w1[8]),.sel(control_wire_final[1]),.out(w2[6]));
mux2X1 ins_43 (.in0(w1[5]),.in1(w1[7]),.sel(control_wire_final[1]),.out(w2[5]));
mux2X1 ins_44 (.in0(w1[4]),.in1(w1[6]),.sel(control_wire_final[1]),.out(w2[4]));
mux2X1 ins_45 (.in0(w1[3]),.in1(w1[5]),.sel(control_wire_final[1]),.out(w2[3]));
mux2X1 ins_46 (.in0(w1[2]),.in1(w1[4]),.sel(control_wire_final[1]),.out(w2[2]));
mux2X1 ins_47 (.in0(w1[1]),.in1(w1[3]),.sel(control_wire_final[1]),.out(w2[1]));
mux2X1 ins_48 (.in0(w1[0]),.in1(w1[2]),.sel(control_wire_final[1]),.out(w2[0]));
//1 bit shift right
mux2X1 ins_49 (.in0(w2[15]),.in1(w2[0]),.sel(control_wire_final[0]),.out(out[15]));
mux2X1 ins_50 (.in0(w2[14]),.in1(w2[15]),.sel(control_wire_final[0]),.out(out[14]));
mux2X1 ins_51 (.in0(w2[13]),.in1(w2[14]),.sel(control_wire_final[0]),.out(out[13]));
mux2X1 ins_52 (.in0(w2[12]),.in1(w2[13]),.sel(control_wire_final[0]),.out(out[12]));
mux2X1 ins_53 (.in0(w2[11]),.in1(w2[12]),.sel(control_wire_final[0]),.out(out[11]));
mux2X1 ins_54 (.in0(w2[10]),.in1(w2[11]),.sel(control_wire_final[0]),.out(out[10]));
mux2X1 ins_55 (.in0(w2[9]),.in1(w2[10]),.sel(control_wire_final[0]),.out(out[9]));
mux2X1 ins_56 (.in0(w2[8]),.in1(w2[9]),.sel(control_wire_final[0]),.out(out[8]));
mux2X1 ins_57 (.in0(w2[7]),.in1(w2[8]),.sel(control_wire_final[0]),.out(out[7]));
mux2X1 ins_58 (.in0(w2[6]),.in1(w2[7]),.sel(control_wire_final[0]),.out(out[6]));
mux2X1 ins_59 (.in0(w2[5]),.in1(w2[6]),.sel(control_wire_final[0]),.out(out[5]));
mux2X1 ins_60 (.in0(w2[4]),.in1(w2[5]),.sel(control_wire_final[0]),.out(out[4]));
mux2X1 ins_61 (.in0(w2[3]),.in1(w2[4]),.sel(control_wire_final[0]),.out(out[3]));
mux2X1 ins_62 (.in0(w2[2]),.in1(w2[3]),.sel(control_wire_final[0]),.out(out[2]));
mux2X1 ins_63 (.in0(w2[1]),.in1(w2[2]),.sel(control_wire_final[0]),.out(out[1]));
mux2X1 ins_64 (.in0(w2[0]),.in1(w2[1]),.sel(control_wire_final[0]),.out(out[0]));
```

```
endmodule


module alu(input wire [15:0] in,input wire [3:0] control_wire_final,input wire [1:0] op, output wire [15:0] out);

        wire [15:0] w1,w2,w3,a;

        barrel_shifter_16bit_left bs_left(in, control_wire_final, w1);

        barrel_shifter_16bit_right bs_right(in, control_wire_final, w2);

        rotator_16bit_left rotate_l(in, control_wire_final, w3);

        rotator_16bit_right rotate_r(in, control_wire_final,a);

        mux4 inst1(w1[0],w2[0],w3[0], a[0], op[0],op[1],out[0]);

        mux4 inst2(w1[1],w2[1],w3[1], a[1], op[0],op[1],out[1]);

        mux4 inst3(w1[2],w2[2],w3[2], a[2], op[0],op[1],out[2]);

        mux4 inst4(w1[3],w2[3],w3[3], a[3], op[0],op[1],out[3]);

        mux4 inst5(w1[4],w2[4],w3[4], a[4], op[0],op[1],out[4]);

        mux4 inst6(w1[5],w2[5],w3[5], a[5], op[0],op[1],out[5]);

        mux4 inst7(w1[6],w2[6],w3[6], a[6], op[0],op[1],out[6]);

        mux4 inst8(w1[7],w2[7],w3[7], a[7], op[0],op[1],out[7]);

        mux4 inst9(w1[8],w2[8],w3[8], a[8], op[0],op[1],out[8]);

        mux4 inst10(w1[9],w2[9],w3[9], a[9], op[0],op[1],out[9]);

        mux4 inst11(w1[10],w2[10],w3[10], a[10], op[0],op[1],out[10]);

        mux4 inst12(w1[11],w2[11],w3[11], a[11], op[0],op[1],out[11]);

        mux4 inst13(w1[12],w2[12],w3[12], a[12], op[0],op[1],out[12]);

        mux4 inst14(w1[13],w2[13],w3[13], a[13], op[0],op[1],out[13]);

        mux4 inst15(w1[14],w2[14],w3[14], a[14], op[0],op[1],out[14]);

        mux4 inst16(w1[15],w2[15],w3[15], a[15], op[0],op[1],out[15]);

endmodule
```

TESTBENCH:

```
`timescale 1 ns / 100 ps

`define TESTVECS 16

module tb;

 reg clk, reset;

 reg [1:0] op; reg [15:0] in;

 reg [3:0]ctrl;

 wire [15:0] out;

 reg [21:0] test_vecs [0:(`TESTVECS-1)];

 integer i;

 initial begin $dumpfile("tb_shift_rotate_16bit.vcd"); $dumpvars(0,tb); end

 initial begin reset = 1'b1; #12.5 reset = 1'b0; end

 initial clk = 1'b0; always #5 clk =~ clk;

 initial begin

 test_vecs[0][21:20] = 2'b00; test_vecs[0][19:16] = 4'b0001;test_vecs[0][15:0] = 16'h75ab;//left shift
by 1 bits

 test_vecs[1][21:20] = 2'b00; test_vecs[1][19:16] = 4'b0010;test_vecs[1][15:0] = 16'h75ab;//left shift
by 2 bits

 test_vecs[2][21:20] = 2'b00; test_vecs[2][19:16] = 4'b0100;test_vecs[2][15:0] = 16'h75ab;//" by 4 bits

 test_vecs[3][21:20] = 2'b00; test_vecs[3][19:16] = 4'b1000;test_vecs[3][15:0] = 16'h75ab;//" by 8 bits

 test_vecs[4][21:20] = 2'b01; test_vecs[4][19:16] = 4'b0001;test_vecs[4][15:0] = 16'h75ab;//right shift
by 1 bit

 test_vecs[5][21:20] = 2'b01; test_vecs[5][19:16] = 4'b0010;test_vecs[5][15:0] = 16'h75ab;// "by 2 bits

 test_vecs[6][21:20] = 2'b01; test_vecs[6][19:16] = 4'b0100;test_vecs[6][15:0] = 16'h75ab;//" by 4 bits

 test_vecs[7][21:20] = 2'b01; test_vecs[7][19:16] = 4'b1000;test_vecs[7][15:0] = 16'h75ab;//" by 8 bits

 test_vecs[8][21:20] = 2'b10; test_vecs[8][19:16] = 4'b0001;test_vecs[8][15:0] = 16'h75ab;//rotate left
by 1 bit

 test_vecs[9][21:20] = 2'b10; test_vecs[9][19:16] = 4'b0010;test_vecs[9][15:0] = 16'h75ab;//rotate by
2 bits
```

```
 test_vecs[10][21:20] = 2'b10; test_vecs[10][19:16] = 4'b0100;test_vecs[10][15:0] = 16'h75ab;// by 4 bits

 test_vecs[11][21:20] = 2'b10; test_vecs[11][19:16] = 4'b1000;test_vecs[11][15:0] = 16'h75ab;//"by 8 bits

 test_vecs[12][21:20] = 2'b11; test_vecs[12][19:16] = 4'b0001;test_vecs[12][15:0] = 16'h75ab;//rotate right by 1 bit

 test_vecs[13][21:20] = 2'b11; test_vecs[13][19:16] = 4'b0010;test_vecs[13][15:0] = 16'h75ab;//" by 2 bits

 test_vecs[14][21:20] = 2'b11; test_vecs[14][19:16] = 4'b0100;test_vecs[14][15:0] = 16'h75ab;//" by 4 bits

 test_vecs[15][21:20] = 2'b11; test_vecs[15][19:16] = 4'b1000;test_vecs[15][15:0] = 16'h75ab;//" by 8 bits

 end

 initial {in,ctrl,op} = 0;

 alu alu_0 (in,ctrl,op,out);

 initial begin

 #6 for(i=0;i<`TESTVECS;i=i+1)

 begin #10 {op,ctrl,in}=test_vecs[i]; end

 #100 $finish;

 end

endmodule
```

OUTPUT:

TERMINAL COMMANDS:

```
tarun@codebind: ~/Desktop/Year2/Sem3/DDCO Project
tarun@codebind:~/Desktop/Year2/Sem3/DDCO Project$ iverilog -o aout shift_rotate_16bit.v tb_shift_rotate_16bit.v
tarun@codebind:~/Desktop/Year2/Sem3/DDCO Project$ vvp aout
VCD info: dumpfile tb_shift_rotate_16bit.vcd opened for output.
tarun@codebind:~/Desktop/Year2/Sem3/DDCO Project$ gtkwave tb_shift_rotate_16bit.vcd

GTKWave Analyzer v3.3.103 (w)1999-2019 BSI

[0] start time.
[266000] end time.
^C
tarun@codebind:~/Desktop/Year2/Sem3/DDCO Project$
```

GTKWAVE OUTPUT: