

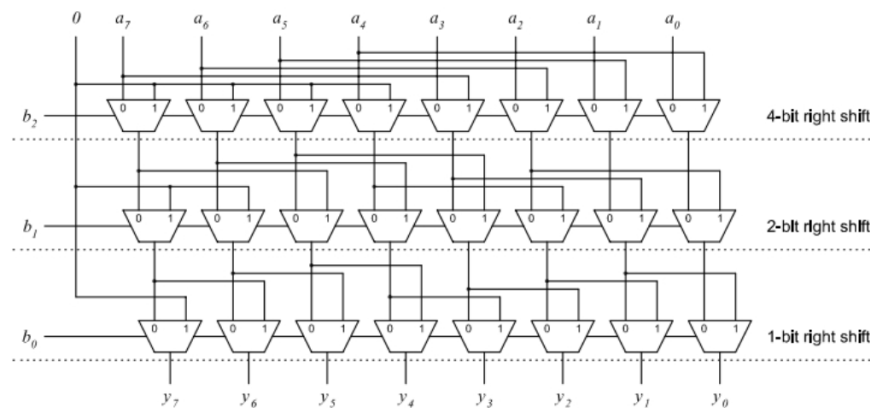


DAY-100

#100DAYSOFRTL

Aim:- Implementation of BARREL SHIFTERS.

1)8-Bit Right Barrel Shifter.



RTL CODE:-

```
//DATE:-09/04/2024
//DAY-100
//IMPLEMENTATION OF 8-BIT LOGICAL LEFT BARREL SHIFTER
module Day_100(input [7:0] A, input [2:0] ctrl, output [7:0] Y);
wire [7:0] w1,w2;
//4-Bit Shift Right
mux_2x1 M1(A[4],A[0],ctrl[2],w1[0]);
mux_2x1 M2(A[5],A[1],ctrl[2],w1[1]);
mux_2x1 M3(A[6],A[2],ctrl[2],w1[2]);
mux_2x1 M4(A[7],A[3],ctrl[2],w1[3]);
mux_2x1 M5(1'b0,A[4],ctrl[2],w1[4]);
mux_2x1 M6(1'b0,A[5],ctrl[2],w1[5]);
mux_2x1 M7(1'b0,A[6],ctrl[2],w1[6]);
mux_2x1 M8(1'b0,A[7],ctrl[2],w1[7]);
//2-Bit Shift Right
mux_2x1 M9(w1[2],w1[0],ctrl[1],w2[0]);
mux_2x1 M10(w1[3],w1[1],ctrl[1],w2[1]);
mux_2x1 M11(w1[4],w1[2],ctrl[1],w2[2]);
mux_2x1 M12(w1[5],w1[3],ctrl[1],w2[3]);
mux_2x1 M13(w1[6],w1[4],ctrl[1],w2[4]);
mux_2x1 M14(w1[7],w1[5],ctrl[1],w2[5]);
mux_2x1 M15(1'b0,w1[6],ctrl[1],w2[6]);
mux_2x1 M16(1'b0,w1[7],ctrl[1],w2[7]);
//1 Bit Shift Right
mux_2x1 M17(w2[1],w2[0],ctrl[0],Y[0]);
mux_2x1 M18(w2[2],w2[1],ctrl[0],Y[1]);
mux_2x1 M19(w2[3],w2[2],ctrl[0],Y[2]);
mux_2x1 M20(w2[4],w2[3],ctrl[0],Y[3]);
mux_2x1 M21(w2[5],w2[4],ctrl[0],Y[4]);
mux_2x1 M22(w2[6],w2[5],ctrl[0],Y[5]);
mux_2x1 M23(w2[7],w2[6],ctrl[0],Y[6]);
mux_2x1 M24(1'b0,w2[7],ctrl[0],Y[7]);
endmodule
```

```

1 //2x1 Multiplexer Module Declaration
2 module mux_2x1(input A,B,Sel,y);
3 assign y=(Sel)? A:B;
4 endmodule

```

TESTBENCH:-

```

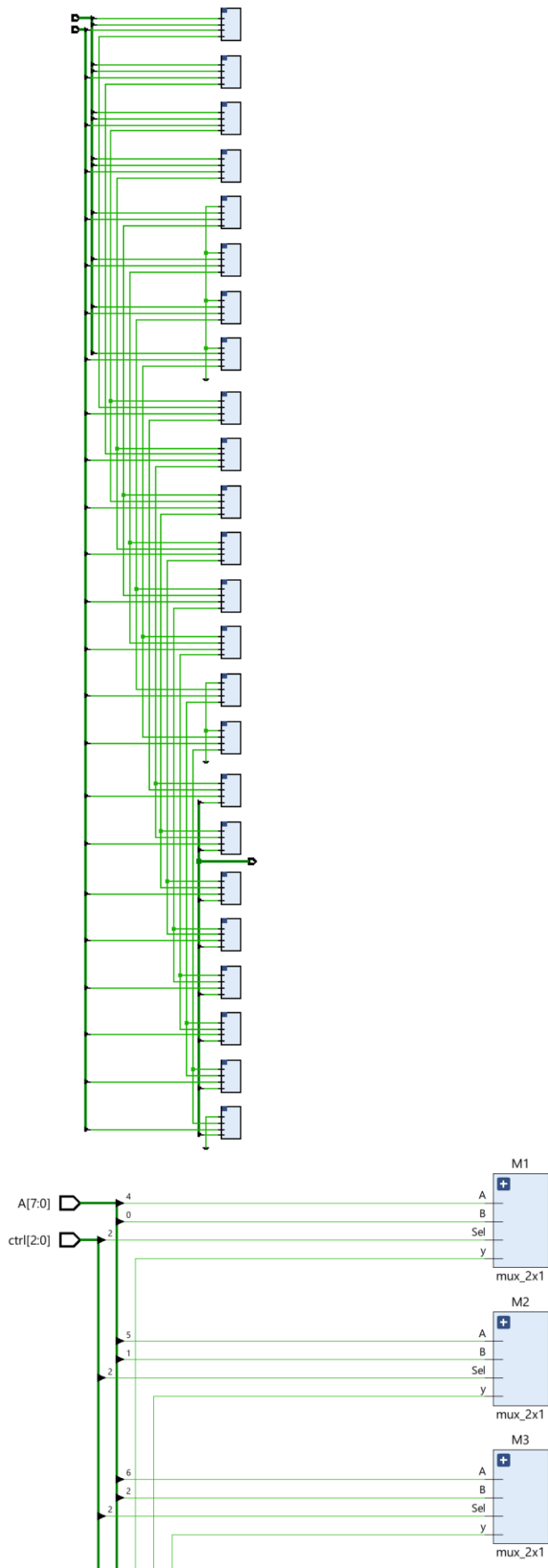
1 module Day_100_tb();
2 reg [7:0] A;
3 reg [2:0] ctrl;
4 wire [7:0] Y;
5 Day_100 uut(A,ctrl,Y);
6 initial begin
7     A=8'd1;
8     ctrl=3'd0;
9     #10;
10    A=8'd4;
11    ctrl=3'd1;
12    #10;;
13    A=8'd8;
14    ctrl=3'd2;
15    #10;
16    A=8'd16;
17    ctrl=3'd3;
18    #10;
19    A=8'd128;
20    ctrl=3'd4;
21    #10;
22    A=8'd32;
23    ctrl=3'd5;
24    #10;
25    A=8'd64;
26    ctrl=3'd6;
27    #10;
28    A=8'd128;
29    ctrl=3'd7;#10;
30    $finish();
31 end
32 endmodule

```

WAVEFORMS:-

Name	Value	0.000 ns	10.000 ns	20.000 ns	30.000 ns	40.000 ns	50.000 ns	60.000 ns	70.000 ns
> A[7:0]	128	1	4	8	16	128	32	64	128
> ctrl[2:0]	7	0	1	2	3	4	5	6	7
> Y[7:0]	1	1	2	8	1	1	1	1	1

SCHEMATIC:-



2)16-BIT RIGHT AND LEFT ROTATOR BARREL SHIFTER USING CONTROL INPUT.

RTL CODE:-

```
///DATE:-09/04/2024
///DAY-100
///IMPLEMENTATION OF 16-BIT RIGHT AND LEFT ROTATOR BARREL SHIFTER USING Control input
module Day_100(input [15:0] A, [3:0] ctrl,input control,
output reg [15:0] Y);
○ always@(*) begin
○ if(control) begin
○ case(ctrl)
○ 4'd0:Y=A;
○ 4'd1:Y={A[0],A[15:1]}; ///control=1; Right
○ 4'd2:Y={A[1:0],A[15:2]}; ///control=0; left
○ 4'd3:Y={A[2:0],A[15:3]};
○ 4'd4:Y={A[3:0],A[15:4]};
○ 4'd5:Y={A[4:0],A[15:5]};
○ 4'd6:Y={A[5:0],A[15:6]};
○ 4'd7:Y={A[6:0],A[15:7]};
○ 4'd8:Y={A[7:0],A[15:8]};
○ 4'd9:Y={A[8:0],A[15:9]};
○ 4'd10:Y={A[9:0],A[15:10]};
○ 4'd11:Y={A[10:0],A[15:11]};
○ 4'd12:Y={A[11:0],A[15:12]};
○ 4'd13:Y={A[12:0],A[15:13]};
○ 4'd14:Y={A[13:0],A[15:14]};
○ 4'd15:Y={A[14:0],A[15]};
○ default: Y=0;
○ endcase
○ end

○ else begin
○ case(ctrl)
○ 4'd0:Y=A;
○ 4'd1:Y={A[14:0],A[15]};
○ 4'd2:Y={A[13:0],A[15:14]};
○ 4'd3:Y={A[12:0],A[15:13]};
○ 4'd4:Y={A[11:0],A[15:12]};
○ 4'd5:Y={A[10:0],A[15:11]};
○ 4'd6:Y={A[9:0],A[15:10]};
○ 4'd7:Y={A[8:0],A[15:9]};
○ 4'd8:Y={A[7:0],A[15:8]};
○ 4'd9:Y={A[6:0],A[15:7]};
○ 4'd10:Y={A[5:0],A[15:6]};
○ 4'd11:Y={A[4:0],A[15:5]};
○ 4'd12:Y={A[3:0],A[15:4]};
○ 4'd13:Y={A[2:0],A[15:3]};
○ 4'd14:Y={A[1:0],A[15:2]};
○ 4'd15:Y={A[0],A[15:1]};
○ default: Y=0;
○ endcase
○ end
○ end
○ endmodule
```

TESTBENCH:-

```

module Day_100_tb();
    reg [15:0] A;
    reg [3:0] ctrl;
    reg control;
    wire [15:0] Y;
    Day_100 uut(A,ctrl,control,Y);
    initial begin
        for(integer i=0; i<10; i++) begin
            A=$random();
            ctrl=$random();
            control=$random();
            #10;
            $display("A=%b,ctrl=%b,control=%b,Y=%b",A,ctrl,control,Y);
            #10;
        end
    end
    initial begin
        #200;
        $finish();
    end
endmodule

```

WAVEFORMS:-

Name	Value	0.000 ns	20.000 ns	40.000 ns	60.000 ns	80.000 ns	100.000 ns	120.000 ns	140.000 ns	160.000 ns
> A[15:0]	0010010011	0011010100100100	0101011001100011	1000010001100101	1100110100001101	0101011111101101	0010010011000110	1111011111100101	1101101110001111	...
> ctrl[3:0]	0101	0001	1101	0010	0110	1100	0101	0111	0010	...
control	0									
> Y[15:0]	1001100011	0001101010010010	1011001100011010	0110000100011001	0011011100110100	0111111011010101	1001100011000100	1111001011111011	0110111000111111	...

SCHEMATIC:-

