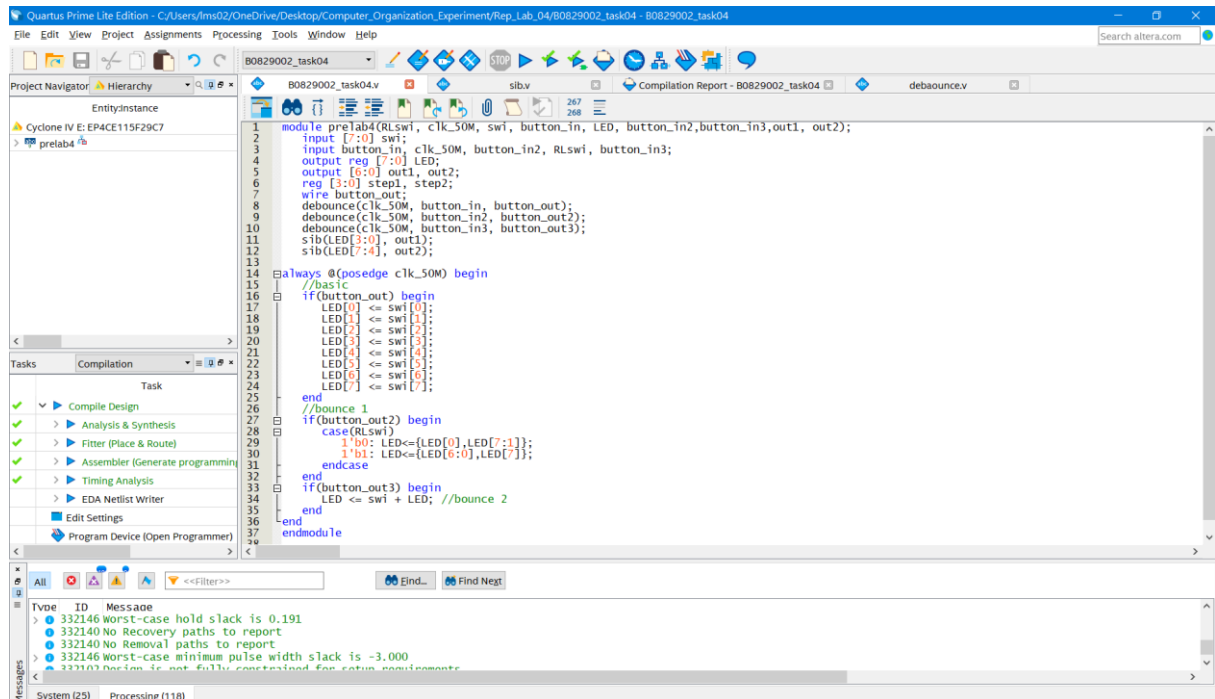


Lab4-Registers and shift Registers

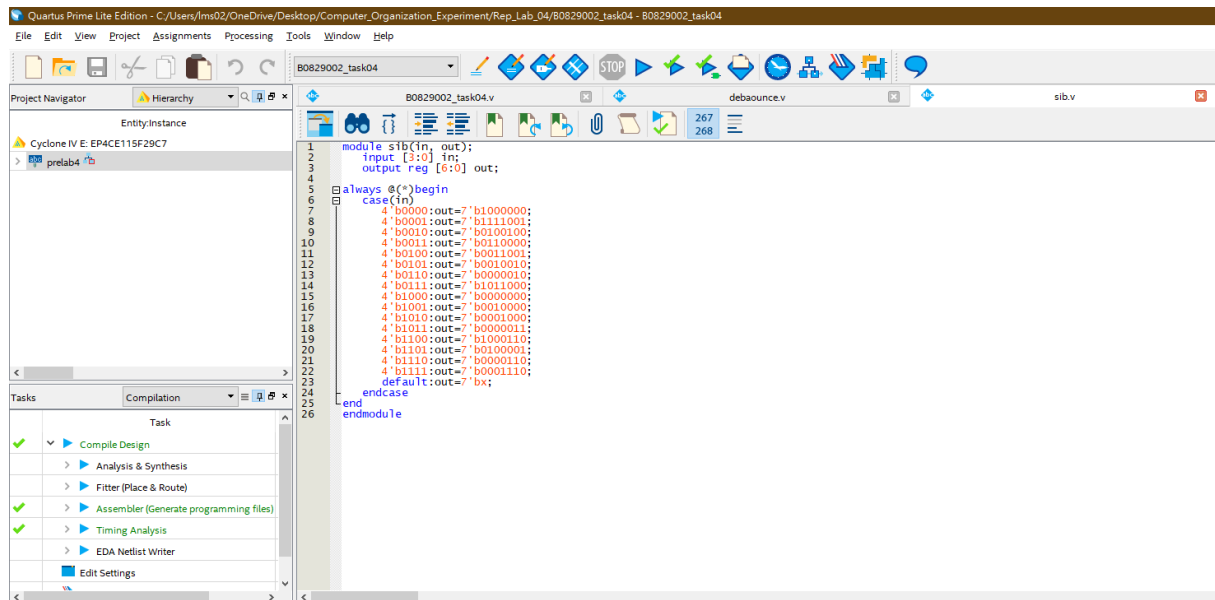
In this Task, we are going to make a register, shifter and adder module, furthermore, we also need to make the output signal into the 7-segment display decoder.

Code:



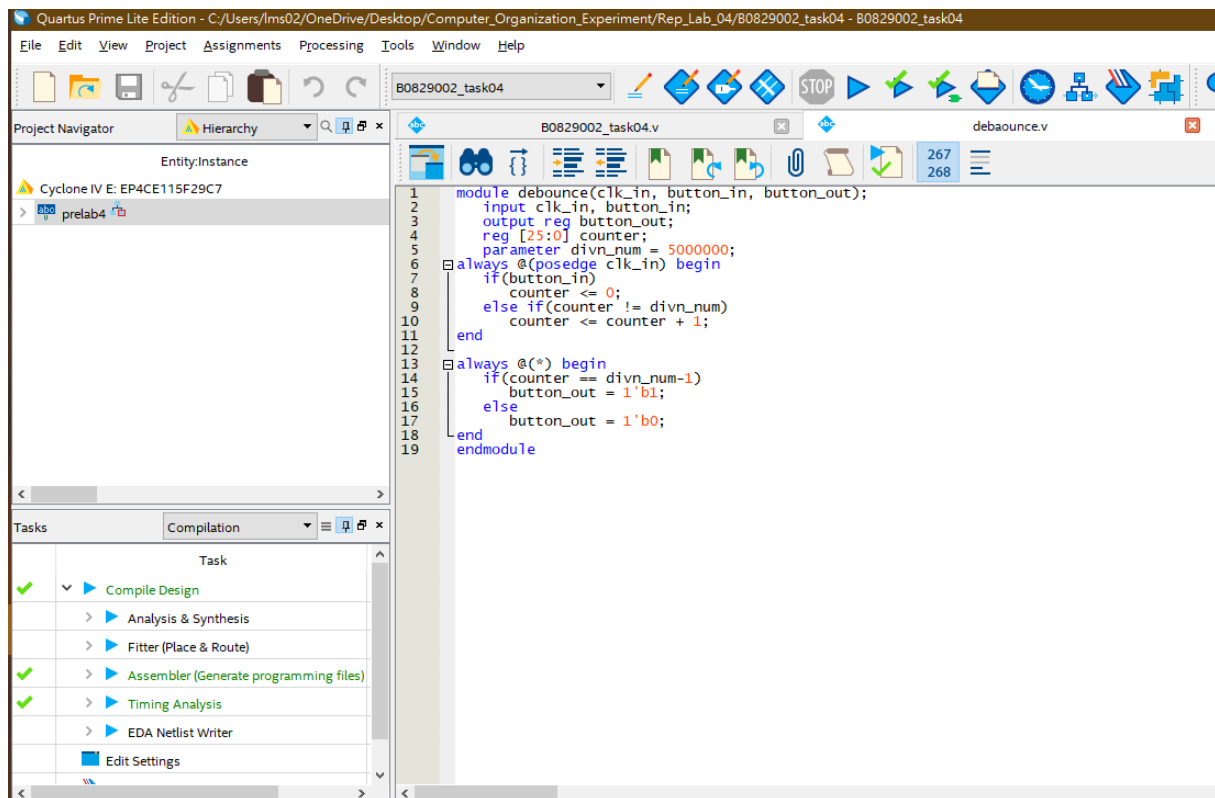
```
1 module prelab4(RLswi, clk_50M, swi, button_in, LED, button_in2, button_in3, out1, out2);
2   input [7:0] swi;
3   input button_in, clk_50M, button_in2, RLswi, button_in3;
4   output reg [7:0] LED;
5   output [6:0] out1, out2;
6   reg [3:0] step1, step2;
7   wire button_out;
8   debounce(clk_50M, button_in, button_out);
9   debounce(clk_50M, button_in2, button_out2);
10  s1b(LED[3:0], out1);
11  s1b(LED[7:4], out2);
12
13  always @(posedge clk_50M) begin
14    //basic
15    if(button_out) begin
16      LED[0] <= swi[0];
17      LED[1] <= swi[1];
18      LED[2] <= swi[2];
19      LED[3] <= swi[3];
20      LED[4] <= swi[4];
21      LED[5] <= swi[5];
22      LED[6] <= swi[6];
23      LED[7] <= swi[7];
24    end
25    //bounce 1
26    if(button_out2) begin
27      case(RLswi)
28        1'b0: LED<={LED[0],LED[7:1]};
29        1'b1: LED<={LED[6:0],LED[7]};
30      endcase
31    end
32    if(button_out3) begin
33      LED <= swi + LED; //bounce 2
34    end
35  end
36 endmodule
```

The screenshot shows the Quartus Prime Lite Edition interface. The Project Navigator on the left shows the hierarchy for 'Cyclone IV E: EP4CE115F29C7' with 'prelab4' selected. The main editor displays the VHDL code for the 'prelab4' module. The Messages window at the bottom shows system and processing messages.

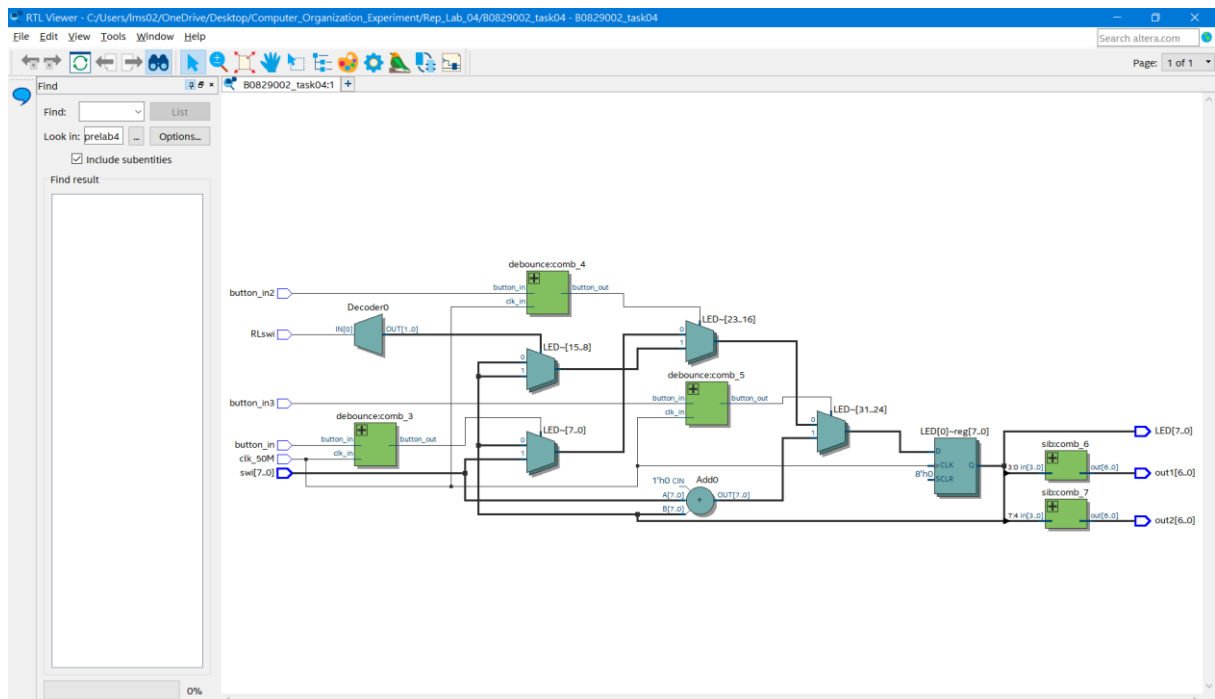


```
1 module s1b(in, out);
2   input [3:0] in;
3   output reg [6:0] out;
4
5   always @(*)begin
6     case(in)
7       4'b0000:out<=7'b1000000;
8       4'b0001:out<=7'b1111001;
9       4'b0010:out<=7'b0100100;
10      4'b0011:out<=7'b0110000;
11      4'b0100:out<=7'b0011001;
12      4'b0101:out<=7'b0010010;
13      4'b0110:out<=7'b0000010;
14      4'b0111:out<=7'b1011000;
15      4'b1000:out<=7'b0000000;
16      4'b1001:out<=7'b0010000;
17      4'b1010:out<=7'b0001000;
18      4'b1011:out<=7'b0000011;
19      4'b1100:out<=7'b1000110;
20      4'b1101:out<=7'b0100001;
21      4'b1110:out<=7'b0000110;
22      4'b1111:out<=7'b0001110;
23      default:out<=7'bx;
24     endcase
25   end
26 endmodule
```

The screenshot shows the Quartus Prime Lite Edition interface. The Project Navigator on the left shows the hierarchy for 'Cyclone IV E: EP4CE115F29C7' with 'prelab4' selected. The main editor displays the VHDL code for the 's1b' module. The Messages window at the bottom shows system and processing messages.



RTL:



Pin planer:

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Clock Preservation
LED[1]	Output	PIN_F19	7	B7_N0	PIN_F19	2.5 V		8mA (default)	2 (default)		
LED[0]	Output	PIN_G19	7	B7_N2	PIN_G19	2.5 V		8mA (default)	2 (default)		
RLsw[6]	Input	PIN_Y23	5	B5_N2	PIN_Y23	2.5 V		8mA (default)	2 (default)		
button_in	Input	PIN_M23	6	B6_N2	PIN_M23	2.5 V		8mA (default)	2 (default)		
button_in2	Input	PIN_M21	6	B6_N1	PIN_M21	2.5 V		8mA (default)	2 (default)		
button_in3	Input	PIN_N21	6	B6_N2	PIN_N21	2.5 V		8mA (default)	2 (default)		
clk_50M	Input	PIN_Y2	2	B2_N0	PIN_Y2	2.5 V		8mA (default)	2 (default)		
out1[6]	Output	PIN_H22	6	B6_N0	PIN_H22	2.5 V		8mA (default)	2 (default)		
out1[5]	Output	PIN_J22	6	B6_N0	PIN_J22	2.5 V		8mA (default)	2 (default)		
out1[4]	Output	PIN_L25	6	B6_N1	PIN_L25	2.5 V		8mA (default)	2 (default)		
out1[3]	Output	PIN_L26	6	B6_N1	PIN_L26	2.5 V		8mA (default)	2 (default)		
out1[2]	Output	PIN_E17	7	B7_N2	PIN_E17	2.5 V		8mA (default)	2 (default)		
out1[1]	Output	PIN_F22	7	B7_N0	PIN_F22	2.5 V		8mA (default)	2 (default)		
out1[0]	Output	PIN_G18	7	B7_N2	PIN_G18	2.5 V		8mA (default)	2 (default)		
out2[6]	Output	PIN_U24	5	B5_N0	PIN_U24	2.5 V		8mA (default)	2 (default)		
out2[5]	Output	PIN_U23	5	B5_N1	PIN_U23	2.5 V		8mA (default)	2 (default)		
out2[4]	Output	PIN_W25	5	B5_N1	PIN_W25	2.5 V		8mA (default)	2 (default)		
out2[3]	Output	PIN_W22	5	B5_N0	PIN_W22	2.5 V		8mA (default)	2 (default)		
out2[2]	Output	PIN_W21	5	B5_N1	PIN_W21	2.5 V		8mA (default)	2 (default)		
out2[1]	Output	PIN_Y22	5	B5_N0	PIN_Y22	2.5 V		8mA (default)	2 (default)		
out2[0]	Output	PIN_M24	6	B6_N2	PIN_M24	2.5 V		8mA (default)	2 (default)		
sw1[7]	Input	PIN_AB26	5	B5_N1	PIN_AB26	2.5 V		8mA (default)	2 (default)		
sw1[6]	Input	PIN_AD26	5	B5_N2	PIN_AD26	2.5 V		8mA (default)	2 (default)		
sw1[5]	Input	PIN_AC26	5	B5_N2	PIN_AC26	2.5 V		8mA (default)	2 (default)		
sw1[4]	Input	PIN_AB27	5	B5_N1	PIN_AB27	2.5 V		8mA (default)	2 (default)		
sw1[3]	Input	PIN_AD27	5	B5_N2	PIN_AD27	2.5 V		8mA (default)	2 (default)		
sw1[2]	Input	PIN_AC27	5	B5_N2	PIN_AC27	2.5 V		8mA (default)	2 (default)		
sw1[1]	Input	PIN_AC28	5	B5_N2	PIN_AC28	2.5 V		8mA (default)	2 (default)		
sw1[0]	Input	PIN_AB28	5	B5_N1	PIN_AB28	2.5 V		8mA (default)	2 (default)		

Discussion and difficulties:

In my opinion, the most difficult part is to know how to use and code the v file for 7-segment display decoder because when I coding this file, I had some problems from the input to this file and output from this file. In the beginning, I have no idea why I use reg to input to this v file from main.v and output from this v file and to main.v will have some problems to the undeclare variables. Thus, I solve a lot of time to check in which need reg and in which does not need. Another difficulty I thought was the making a adder.

Because of storage of the result signal, we also need to use register to store the value from the addition operation. Here we can use the reg from the last problem to solve this problem easily, just do some easy addition and assign the value into the reg from 7-segment display decoder file. After that we can get an addition module.

About the shifter is using the feature of register, input the value to next flipflop and the last one rotate to the beginning of flipflop.