**Task 1**

File: lab01part01.v

Code:

module lab01part01(input [17:0]SW, output [17:0]LEDR);

assign LEDR = SW; //connect each switch to light

endmodule

module testBenchT01;

reg [17:0]sw;

wire[17:0]ld;

lab01part01 testingLab1Task1(sw, ld);

initial begin

#100;

sw = 18'b000000000000000000;

#100;

sw = 18'b000000000000000001;

#100;

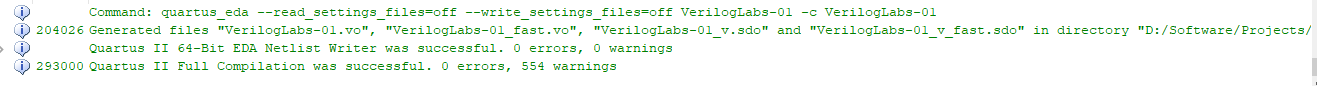
sw = 18'b000000000000001111;

#100;

end

endmodule

Output:



**Task 2**

File: lab01part02.v, base\_8xMUX2to1

Code:

module lab01part02(

input [17:0] SW,

output [7:0] LEDG,

output [17:0] LEDR

);

base\_8xMUX2to1(SW[17], SW[7:0], SW[15:8], LEDG[7:0]);

lab01part01(SW, LEDR); //assigns switches to LEDs

endmodule

///…………………………………………………………………………………………………………………………………………………

module base\_8xMUX2to1 ( input s,

input [7:0] X, Y,

output [7:0] M);

assign M[7] = ((~s & X[7])|(s & Y[7]));

assign M[6] = ((~s & X[6])|(s & Y[6]));

assign M[5] = ((~s & X[5])|(s & Y[5]));

assign M[4] = ((~s & X[4])|(s & Y[4]));

assign M[3] = ((~s & X[3])|(s & Y[3]));

assign M[2] = ((~s & X[2])|(s & Y[2]));

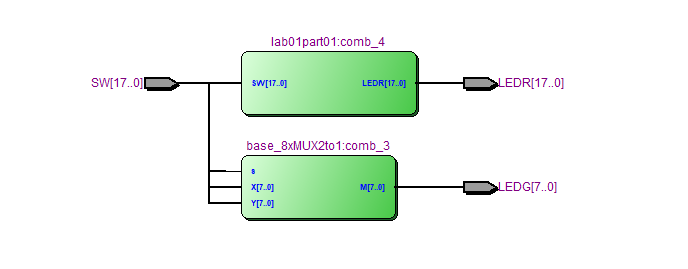
assign M[1] = ((~s & X[1])|(s & Y[1]));

assign M[0] = ((~s & X[0])|(s & Y[0]));

//assign M = ((~s & X) | (s & Y));

endmodule

Output:



**Task 03:**

Filename: lab01part03.v, base\_3xMUX5to1.v

Code:

MAIN

module lab01part03(

input [17:0]SW,

output [17:0]LEDR,

output [2:0]LEDG

);

base\_3xMUX5to1(SW[17:15], SW[14:12], SW[11:9], SW[8:6], SW[5:3], SW[2:0], LEDG[2:0]);

lab01part01(SW, LEDR);

endmodule

3BIT 5x1 MUX

module base\_3xMUX5to1(

input [2:0] S,

input [2:0] U, V, W, X, Y,

output reg [2:0] M

);

always@(\*)

begin

case(S)

3'b000: begin M = U; end

3'b001: begin M = V; end

3'b010: begin M = W; end

3'b011: begin M = X; end

3'b100: begin M = Y; end

3'b101: begin M = Y; end

3'b110: begin M = Y; end

3'b111: begin M = Y; end

endcase

end

endmodule

**Task 04**

Filename: base\_7SegmentDec0.v, lab01task04.v, lab01task04table.csv

Code:

MAIN

module lab01part04(

input [2:0]SW,

output [6:0]HEX0

);

base\_7SegmentDec0(SW[2:0], HEX0[6:0]);

endmodule

TESTBENCH

//MODELSIM TestBench

module testBencht01p04;

reg [2:0]sw;

wire[6:0]hx;

lab01part04 instancA(sw, hx);

initial begin

#100; sw = 3'b000;

#100; sw = 3'b001;

#100; sw = 3'b010;

#100; sw = 3'b011;

#100; sw = 3'b100;

#100; sw = 3'b101;

#100; sw = 3'b110;

#100; sw = 3'b111;

end

endmodule

7 SEGMENT DECODER MODULE

module base\_7SegmentDec0(

input [2:0]in,

output [6:0]out

);

assign out[0] = ~( (~in[0] & ~in[1] & in[2]) | //001

(~in[0] & in[1] & in[2])); //011

assign out[1] = ~( (~in[0] & ~in[1] & ~in[2]) | //000

(~in[0] & in[1] & in[2])); //011

assign out[2] = ~( (~in[0] & ~in[1] & ~in[2]) | //000

(~in[0] & in[1] & in[2])); //011

assign out[3] = ~( (~in[0] & ~in[1] & in[2]) | //001

(~in[0] & in[1] & ~in[2]) | //010

(~in[0] & in[1] & in[2])); //011

assign out[4] = ~( (~in[0] & ~in[1] & ~in[2]) | //000

(~in[0] & ~in[1] & in[2]) | //001

(~in[0] & in[1] & ~in[2]) | //010

(~in[0] & in[1] & in[2])); //011

assign out[5] = ~( (~in[0] & ~in[1] & ~in[2]) | //000

(~in[0] & ~in[1] & in[2]) | //001

(~in[0] & in[1] & ~in[2]) | //010

(~in[0] & in[1] & in[2])); //011

assign out[6] =~( (~in[0] & ~in[1] & ~in[2]) | //000

(~in[0] & !in[1] & in[2])); //011

endmodule

Output:



**Task 05**

Filename: lab01task05.v, base\_3xMUX5to1.v, base\_7SegmentDec0.v

FPGA: Completed.

Code:

MAIN

module lab01part05(

input [2:0]SW,

output [6:0]HEX0, HEX1, HEX2, HEX3, HEX4

);

wire [2:0]M;

reg [2:0]N,O,P,R;

reg [2:0]v,w,x,y;

base\_3xMUX5to1 characterSelect(SW[2:0], 3'b000, 3'b001, 3'b010, 3'b011, 3'b100, M[2:0]);

always@(\*)

begin

case(M)

3'b000: begin v=0; w=0; x=0; y=0; end

3'b001: begin v=0; w=0; x=0; y=4; end

3'b010: begin v=0; w=0; x=4; y=4; end

3'b011: begin v=0; w=4; x=4; y=4; end

3'b100: begin v=4; w=4; x=4; y=4; end

3'b101: begin v=4; w=4; x=4; y=0; end

3'b110: begin v=4; w=4; x=0; y=0; end

3'b111: begin v=4; w=0; x=0; y=0; end

endcase

R = M+3'b000-v;

N = M+3'b001-w;

O = M+3'b010-x;

P = M+3'b011-y;

end

base\_7SegmentDec0 displayHex0(R[2:0], HEX3[6:0]);

base\_7SegmentDec0 displayHex1(N[2:0], HEX2[6:0]);

base\_7SegmentDec0 displayHex2(O[2:0], HEX1[6:0]);

base\_7SegmentDec0 displayHex3(P[2:0], HEX0[6:0]);

endmodule

TESTBENCH

module testBenchL01T05;

reg [2:0]sw;

wire [6:0]h0, h1, h2, h3, h4;

lab01part05 lab1Verify5(sw, h0, h1, h2, h3, h4);

initial begin

#100; sw=3'b000;

#100; sw=3'b001;

#100; sw=3'b010;

#100; sw=3'b011;

#100; sw=3'b100;

#100; sw=3'b101;

#100; sw=3'b110;

#100; sw=3'b111;

end

endmodule