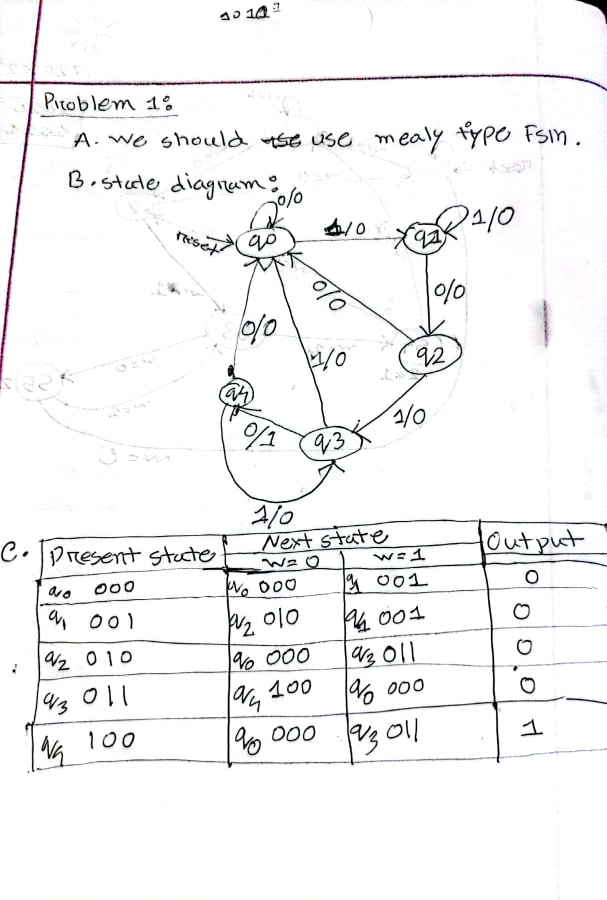
Task 1:



Code:

module task\_1(clk,resetn,w,z);

input clk,resetn,w;

output reg z;

reg y, Y;

parameter [3:1] q0=3'b000,

q1=3'b001,

q2=3'b010,

q3=3'b011,

q4=3'b100;

always@(w,y)

case (y)

q0: if (w)

begin

z = 0;

Y = q1;

end

else

begin

z = 0;

Y = q0;

end

q1: if (w)

begin

z = 0;

Y = q1;

end

else

begin

z = 0;

Y = q2;

end

q2: if (w)

begin

z = 0;

Y = q3;

end

else

begin

z = 0;

Y = q0;

end

q3: if (w)

begin

z = 0;

Y = q0;

end

else

begin

z = 1;

Y = q4;

end

q4: if (w)

begin

z = 0;

Y = q3;

end

else

begin

z = 0;

Y = q0;

end

endcase

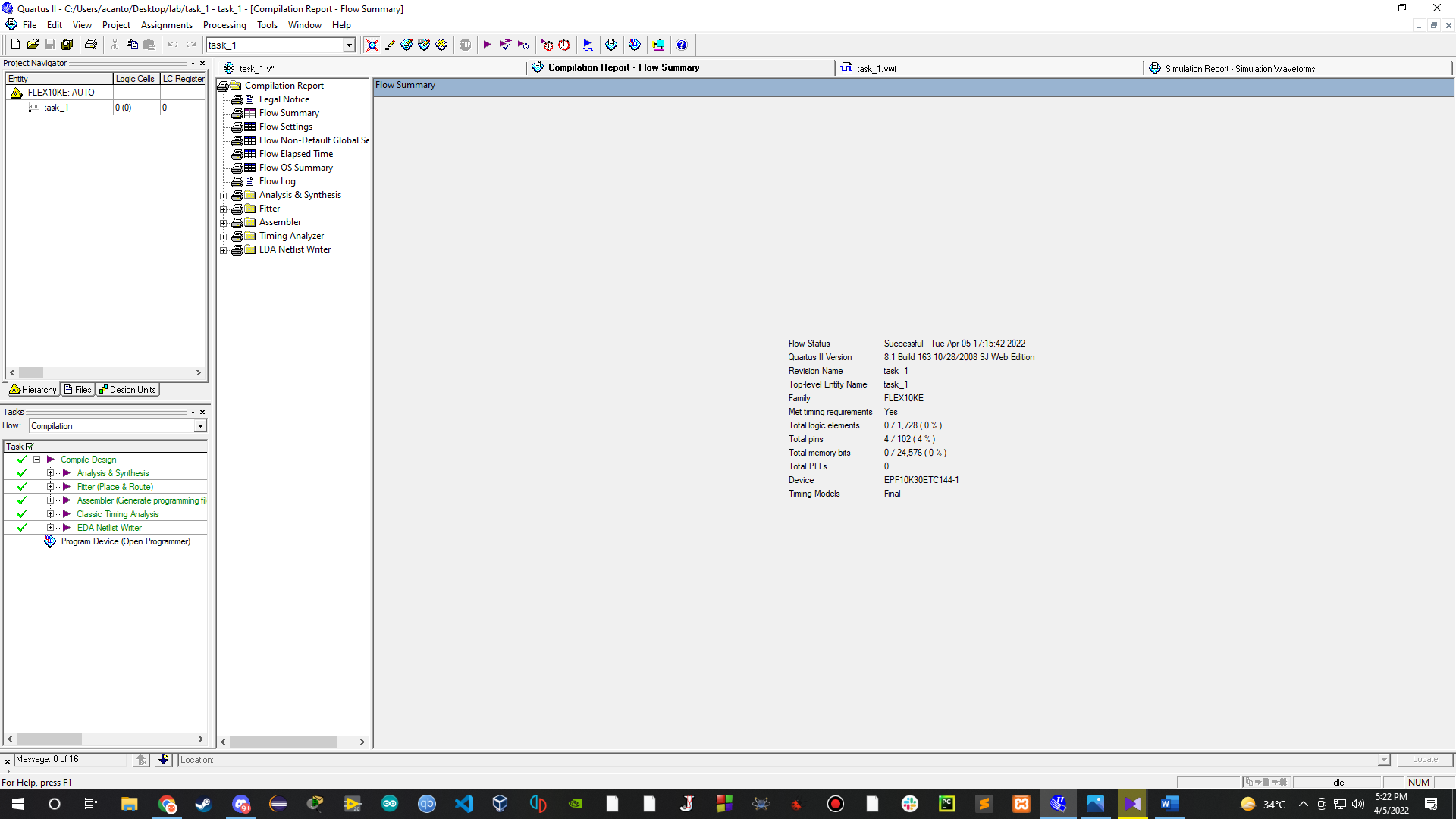
always @ (negedge resetn, posedge clk)

if (resetn == 0) y <= q0;

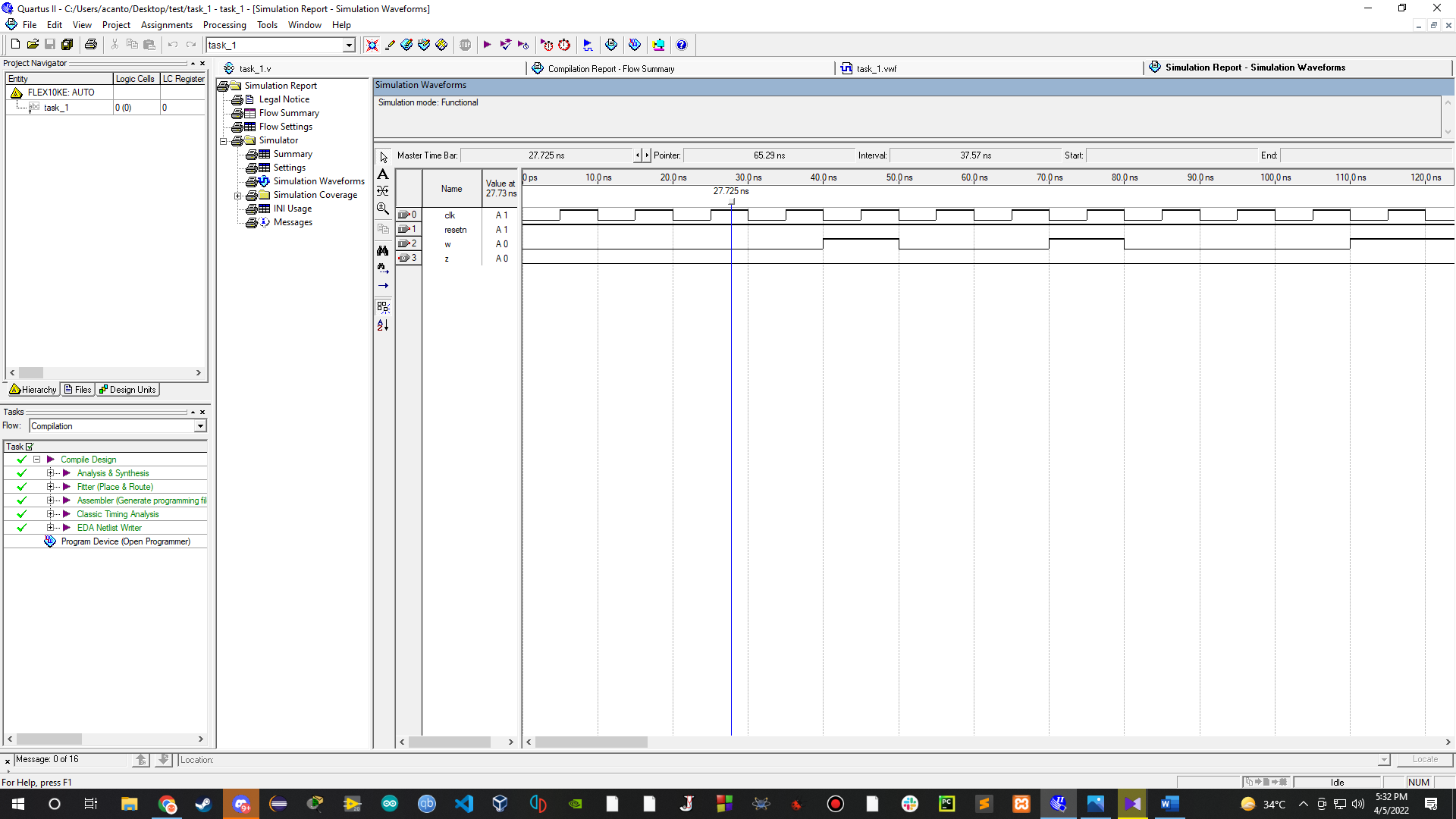
else y<=Y;

endmodule

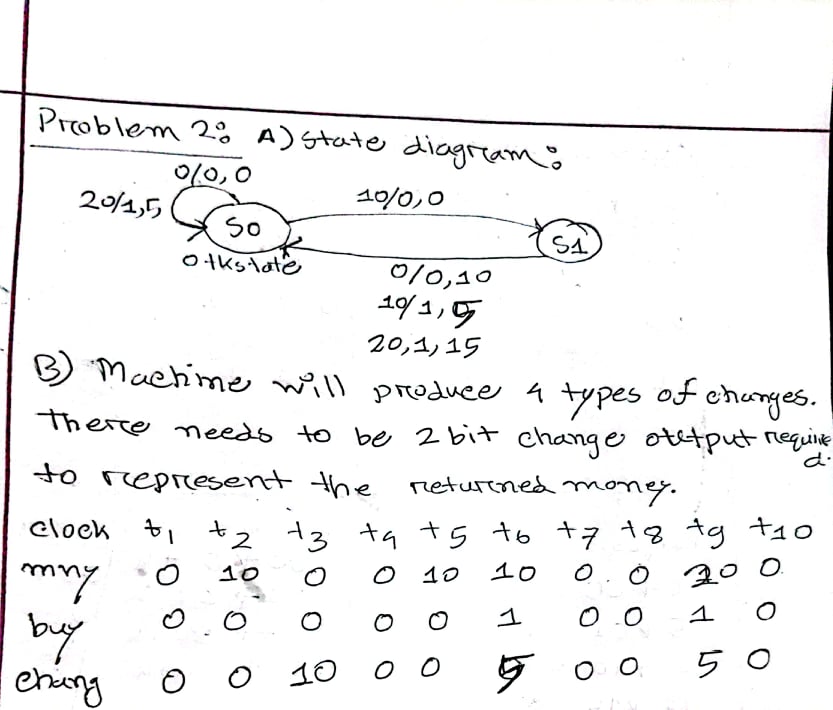
compilation report:

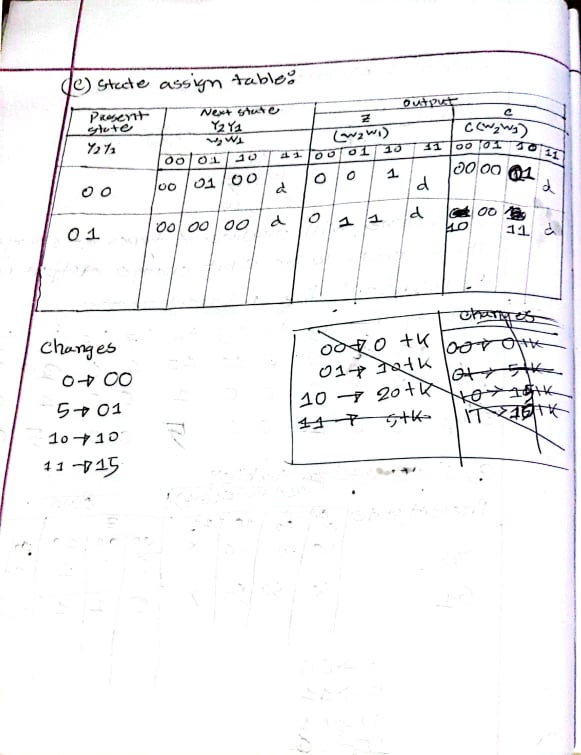


simulation report:



Task 2:





Code:

module task\_2(clock, reset, cash\_in, purchase, present\_state, next\_state, cash\_return);

input clock, reset;

input [1:0] cash\_in;

output reg purchase;

output reg [1:0] cash\_return, present\_state, next\_state;

parameter state0= 2'b00,

state1= 2'b01,

n = 15,

R0= 2'b00,

R5= 2'b01,

R10= 2'b10,

R15= 2'b11;

always@(posedge clock)

begin

if(reset==1)

begin

present\_state = state0;

next\_state = state0;

end

else

begin

present\_state = next\_state;

case(present\_state)

state0: if(cash\_in == 2'b00)

begin

next\_state = state0;

purchase =0;

cash\_return = R0;

end

else if(cash\_in == 2'b01)

begin

next\_state = state1;

purchase = 0;

cash\_return = R0;

end

else if(cash\_in == 2'b10)

begin

next\_state = state0;

purchase = 1;

cash\_return = R5;

end

state1: if(cash\_in == 2'b00)

begin

next\_state = state0;

purchase =0;

cash\_return = R10;

end

else if(cash\_in == 2'b01)

begin

next\_state=state0;

purchase = 1;

cash\_return = R5;

end

else if(cash\_in == 2'b10)

begin

next\_state=state0;

purchase=1;

cash\_return = R15;

end

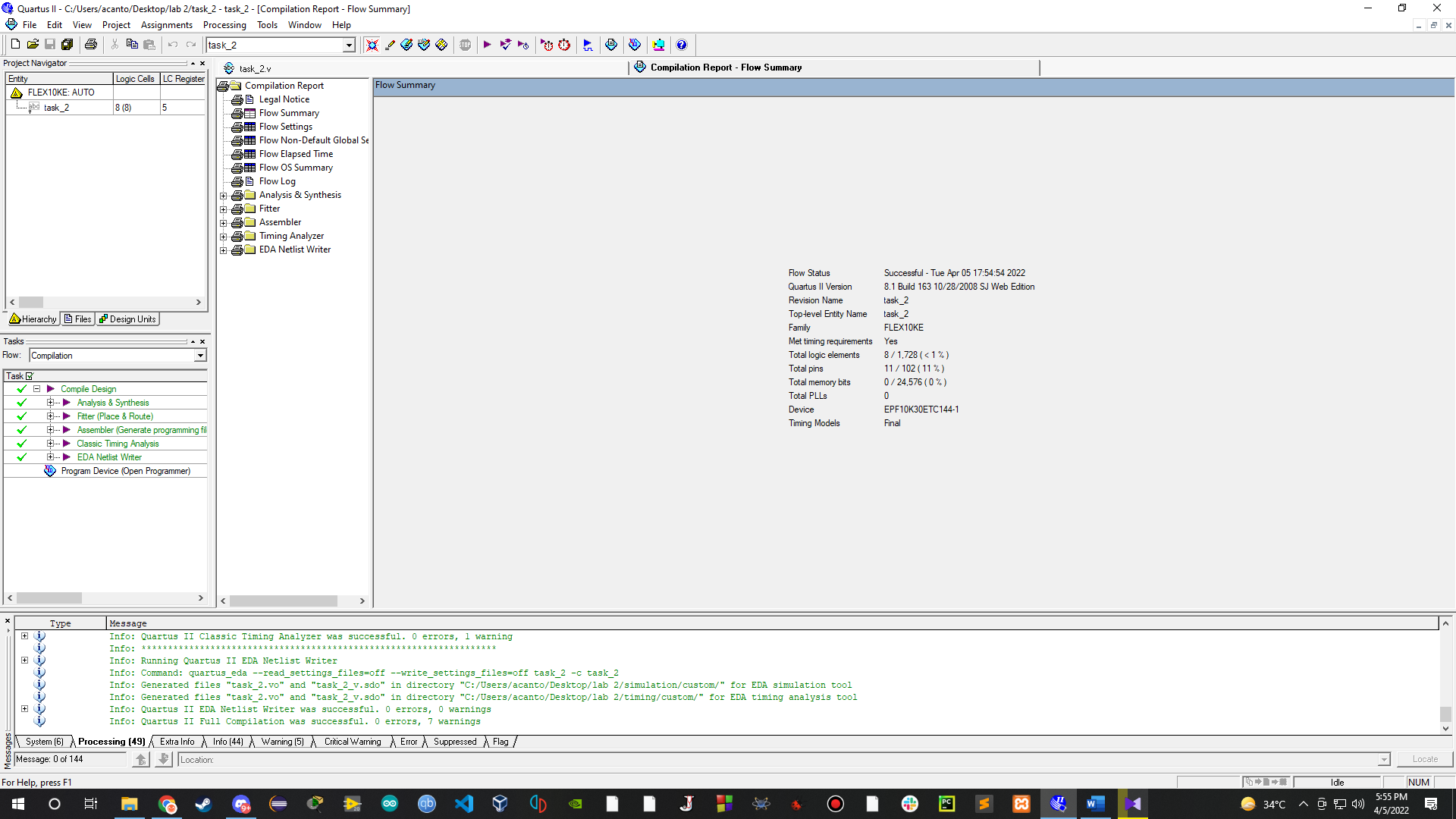
endcase

end

end

endmodule

compilation report:



simulation report:

