Team name: 32bit

Names:

Mahmoud mohamed alsayd

Omar Khaled elsaid

Abdelkader saad

Code :

```
module project_dsp (A , B , C , D , BCIN , CARRYIN , M , P , CARRYOUT , CARRYOUTF , CLK , OPMODE , CEA , CEB , CEC , CECARRYIN , CED , CEM , CEOPMODE , CEP , RSTA , RSTB , RSTC , RSTCARRYIN ,

parameter A0REG = 0;
parameter A1REG = 1;
parameter B0REG = 0;
parameter B1REG = 1;
parameter CREG = 1;
parameter DREG = 1;
parameter MREG = 1;
parameter PREG = 1;
parameter CARRYINREG =1;
parameter CARRYOUTREG =1;
parameter OPMODEREG =1;
parameter CARRYINSEL = "OPMODE5" ;
parameter B_INPUT = "DIRECT" ;
parameter RSTTYPE = "SYNC" ;

input [17:0] A , B , D , BCIN ;
input [47:0] C , PCIN ;
input CARRYIN , CLK , CEA , CEB , CEC ,CECARRYIN , CED , CEM , CEOPMODE , CEP , RSTA , RSTB , RSTC , RSTCARRYIN , RSTD , RSTM , RSTOPMODE , RSTP  ;
output [35:0] M ;
output CARRYOUTF , CARRYOUT ;
input [7:0] OPMODE ;
output [17:0] BCOUT  ;
output [47:0] PCOUT,P ;


wire [17:0] D_from_mux  , B0_from_mux , A0_from_mux , Pre_Add_Sub , Pre_Add_Sub_from_mux , A1_from_mux, B1_from_mux ;
wire [47:0] C_from_mux , concatenated , post_Add_Sub , P_from_mux ;
wire[35:0] mult_out , mult_out_from_mux ;
reg [17:0] data_b;
wire CYI_from_mux , cout , CYO_from_mux;
reg carryin;
wire [7:0] OPMODE_from_mux;
reg [47:0] X_out ,  Z_out;

//op regs
ff_and_mux #(.ENable_ff(OPMODEREG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(8)) op (OPMODE ,CLK, CEOPMODE , RSTOPMODE , OPMODE_from_mux );


always @(*) begin
case (B_INPUT)
"DIRECT" : data_b = B ;
"CASCADE" : data_b = BCIN ;
default : data_b = 0;
endcase
end
//stg 1
ff_and_mux #(.ENable_ff(DREG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(18)) m1 (D ,CLK, CED , RSTD , D_from_mux );
```

```
endcase
end
//stg 1
ff_and_mux #(.ENable_ff(DREG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(18)) m1 (D ,CLK, CED , RSTD , D_from_mux );
ff_and_mux # (.ENable_ff(B0REG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(18)) m2 (data_b ,CLK, CEB , RSTB , B0_from_mux);
ff_and_mux # (.ENable_ff(A0REG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(18)) m3 (A ,CLK, CEA , RSTA , A0_from_mux);
ff_and_mux # (.ENable_ff(CREG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(48)) m4 (C ,CLK, CEC , RSTC ,C_from_mux);

//stg 2
 add_sub #(.WIDTH_2(18),.FULLADDER("OFF")) pre (D_from_mux , B0_from_mux , Pre_Add_Sub , OPMODE_from_mux[6]);
 assign Pre_Add_Sub_from_mux = (OPMODE_from_mux[4]) ? Pre_Add_Sub : B0_from_mux ;

//stg 3
ff_and_mux #(.ENable_ff(B1REG),.SYN_OR_ASN(RSTTYPE),.WIDTH(18)) m5 (Pre_Add_Sub_from_mux , CLK , CEB , RSTB ,B1_from_mux );
ff_and_mux # (.ENable_ff(A1REG) ,.SYN_OR_ASN(RSTTYPE) ,.WIDTH(18)) m6 (A0_from_mux ,CLK, CEA , RSTA , A1_from_mux);

 //stg 4
 assign mult_out = A1_from_mux * B1_from_mux;
 //wires
 assign concatenated = {D_from_mux[11:0] , A1_from_mux[17:0] , B1_from_mux[17:0]} ;
 assign BCOUT = B1_from_mux ;

//stg 5
ff_and_mux #(.ENable_ff(MREG),.SYN_OR_ASN(RSTTYPE),.WIDTH(36)) m7 (mult_out , CLK , CEM , RSTM , mult_out_from_mux);
assign  M = mult_out_from_mux ;
always @(*) begin
    case (CARRYINSEL)
     "CARRYIN" : carryin=CARRYIN;
     "OPMODE5" : carryin=OPMODE_from_mux[5];
     default : carryin=0;

    endcase
end
ff_and_mux # (.ENable_ff(CARRYINREG),.SYN_OR_ASN(RSTTYPE),.WIDTH(1)) m8 (carryin , CLK , CECARRYIN , RSTCARRYIN , CYI_from_mux);

//stg 6

always @(*) begin
    case (OPMODE_from_mux[1:0])
    2'b00 : X_out = 0;
    2'b01 : X_out = mult_out_from_mux ;
    2'b10 : X_out = PCOUT ;
    2'b11 : X_out = concatenated ;

    endcase
end
always @(*) begin
    case (OPMODE_from_mux[3:2])
    2'b00 : Z_out = 0;
    2'b01 : Z_out = PCIN ;
    2'b10 : Z_out = PCOUT ;
    2'b11 : Z_out = C_from_mux ;

    endcase
end


//stg 7
 add_sub #(.WIDTH_2(48),.FULLADDER("ON")) post (Z_out,X_out,post_Add_Sub,OPMODE_from_mux[7], CYI_from_mux , cout );
 ff_and_mux #(.ENable_ff(CARRYOUTREG),.SYN_OR_ASN(RSTTYPE),.WIDTH(1)) m9 (cout , CLK, CECARRYIN , RSTCARRYIN , CYO_from_mux  );
assign CARRYOUT = CYO_from_mux ;
assign CARRYOUTF = CYO_from_mux ;
//stg 8
 ff_and_mux #(.ENable_ff(PREG),.SYN_OR_ASN(RSTTYPE),.WIDTH(48)) m10 (post_Add_Sub , CLK, CEP , RSTP , P_from_mux );
assign PCOUT = P_from_mux;
assign P = P_from_mux;

endmodule
```

Ffandmux:

```
module ff_and_mux (inp , clk, clk_enable , rst , out );

parameter ENable_ff = 1;
parameter SYN_OR_ASN = "SYNC" ;
parameter WIDTH = 18 ;


input [WIDTH -1 : 0] inp ;
input clk ,clk_enable, rst ;
output [WIDTH -1 : 0] out ;
reg [WIDTH-1 : 0] out_ff;

generate
    if (SYN_OR_ASN == "SYNC")begin
    always @(posedge clk ) begin
        if (rst) begin
            out_ff <= 0;
        end
        else if (clk_enable) begin
            out_ff <= inp;
        end
    end
end
 else if (SYN_OR_ASN == "ASYNC")begin
    always @(posedge clk or posedge rst) begin
        if (rst) begin
            out_ff <= 0;

        end
        else if (clk_enable) begin
            out_ff <= inp ;
        end
    end
 end

endgenerate

assign out = (ENable_ff == 1)? out_ff : inp ;
endmodule


```

Adder and subtractor:

```
module add_sub (in1,in2,out,OPMODE,cin,cout);
 parameter FULLADDER="ON";
 parameter WIDTH_2=18;
 input OPMODE,cin;
 input [WIDTH_2-1:0] in1,in2;
 output reg [WIDTH_2-1:0] out;
 output reg cout ;
 generate
 if (FULLADDER=="ON") begin
          always@(*) begin

     if(OPMODE)
     {cout,out}=in1-(in2+cin);
     else
     {cout,out}=in1+in2+cin;

     end
 end

     else if (FULLADDER=="OFF") begin
 always@(*) begin

     if(OPMODE)
     out=in1-in2;
     else
     out=in1+in2;

    end
 end
 endgenerate
 endmodule

```

Verify :

```
module project_dsp_tb ();
parameter A0REG = 0;
parameter A1REG = 1;
parameter B0REG = 0;
parameter B1REG = 1;
parameter CREG = 1;
parameter DREG = 1;
parameter MREG = 1;
parameter PREG = 1;
parameter CARRYINREG =1;
parameter CARRYOUTREG =1;
parameter OPMODEREG =1;
parameter CARRYINSEL = "OPMODE5" ;
parameter B_INPUT = "DIRECT" ;
parameter RSTTYPE = "SYNC" ;

reg [17:0] A , B , D , BCIN ;
reg [47:0] C , PCIN ;
reg CARRYIN , CLK , CEA , CEB , CEC ,CECARRYIN , CED , CEM , CEOPMODE , CEP , RSTA , RSTB , RSTC , RSTCARRYIN , RSTD , RSTM , RSTOPMODE , RSTP  ;
reg [7:0] OPMODE ;

wire [35:0] M ;
wire CARRYOUTF , CARRYOUT ;
wire [17:0] BCOUT  ;
wire [47:0] PCOUT,P ;

project_dsp dut (A , B , C , D , BCIN , CARRYIN , M , P , CARRYOUT , CARRYOUTF , CLK , OPMODE , CEA , CEB , CEC , CECARRYIN , CED , CEM , CEOPMODE , CEP , RSTA , RSTB , RSTC , RSTCARRYIN , RST


initial begin

    CLK=0;
    forever
    #1 CLK=~CLK;

end



initial begin

RSTA =1;
RSTB =1;
RSTC =1;
RSTCARRYIN =1;
RSTD =1;
RSTM =1;
RSTOPMODE =1;
RSTP =1;
```



```
    RSTA =1;
    RSTB =1;
    RSTC =1;
    RSTCARRYIN =1;
    RSTD =1;
    RSTM =1;
    RSTOPMODE =1;
    RSTP =1;
    A=0;
    B=0 ;
    C=0 ;
    D =0;
    BCIN =0;
    CARRYIN =0;
    OPMODE =8'b1011_1101;
    CEA =1;
    CEB =1;
    CEC =1;
    CECARRYIN =1;
    CED =1;
    CEM =1;
    CEOPMODE=1;
    CEP=1;
    PCIN =0;
    #50;
    RSTA =0;
    RSTB =0;
    RSTC =0;
    RSTCARRYIN =0;
    RSTD =0;
    RSTM =0;
    RSTOPMODE =0;
    RSTP =0;
    A=18'd4;
    B=18'd10 ;
    C=48'd150 ;
    D =18'd15;
    BCIN =0;
    CARRYIN =0;
    OPMODE =8'b1001_1101;
    CEA =1;
    CEB =1;
    CEC =1;
    CECARRYIN =1;
    CED =1;
    CEM =1;
    CEOPMODE=1;
```

```
    CECARRYIN =1;
    CED =1;
    CEM =1;
    CEOPMODE=1;
    CEP=1;
    PCIN =0;
    #50;
    OPMODE=8'b0101_1010;
    #50;
    A=0;
    D=0;
    OPMODE=8'b0010_0011;

    #20 $stop;









    end
    endmodule
```

Do file :

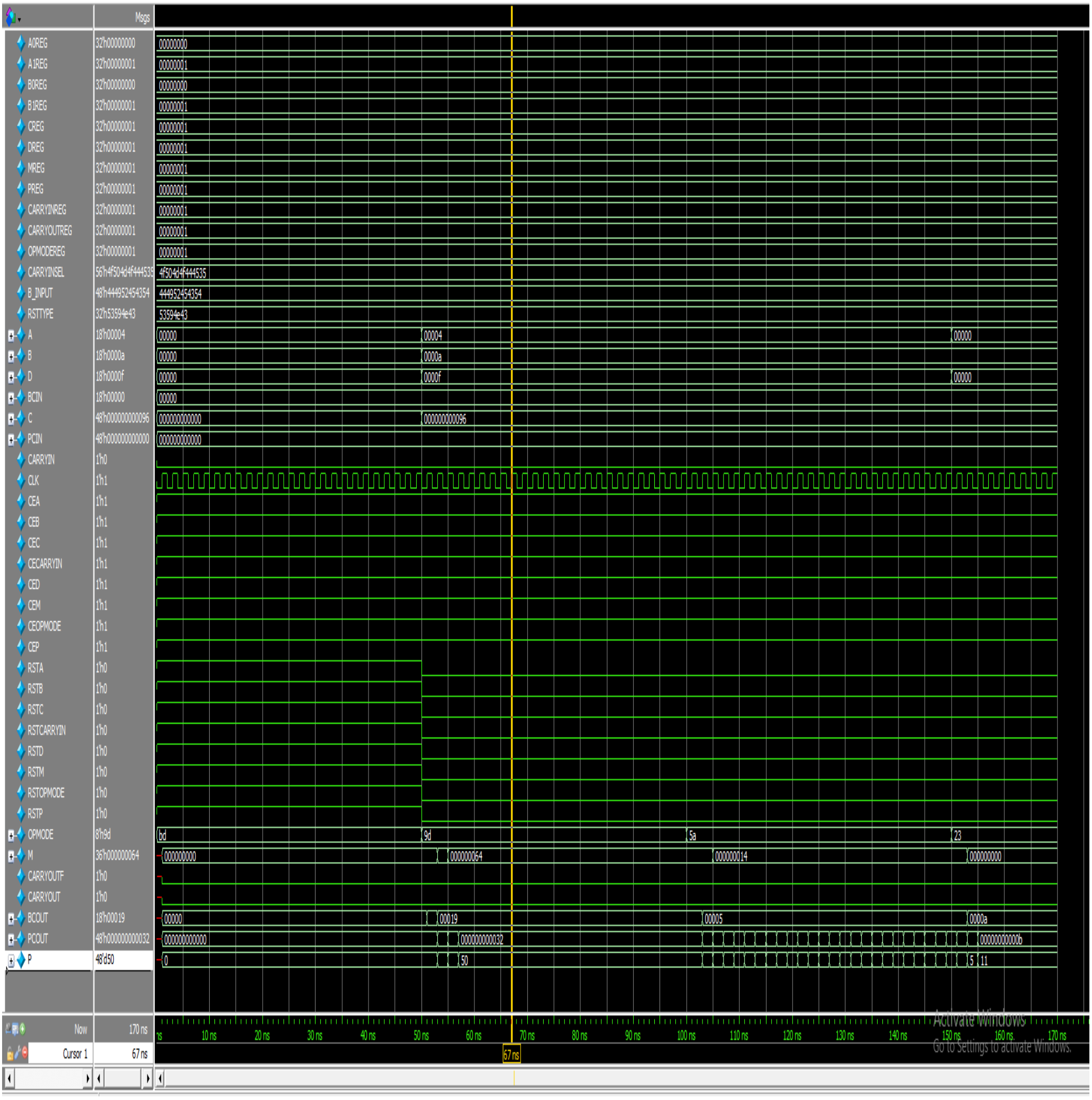
```
vlib work

vlog add_sub.v ffandmux.v project_1.v project__tb.v

vsim -voptargs=+acc work.project_dsp_tb
add wave *

run -all
```

Tcl file :

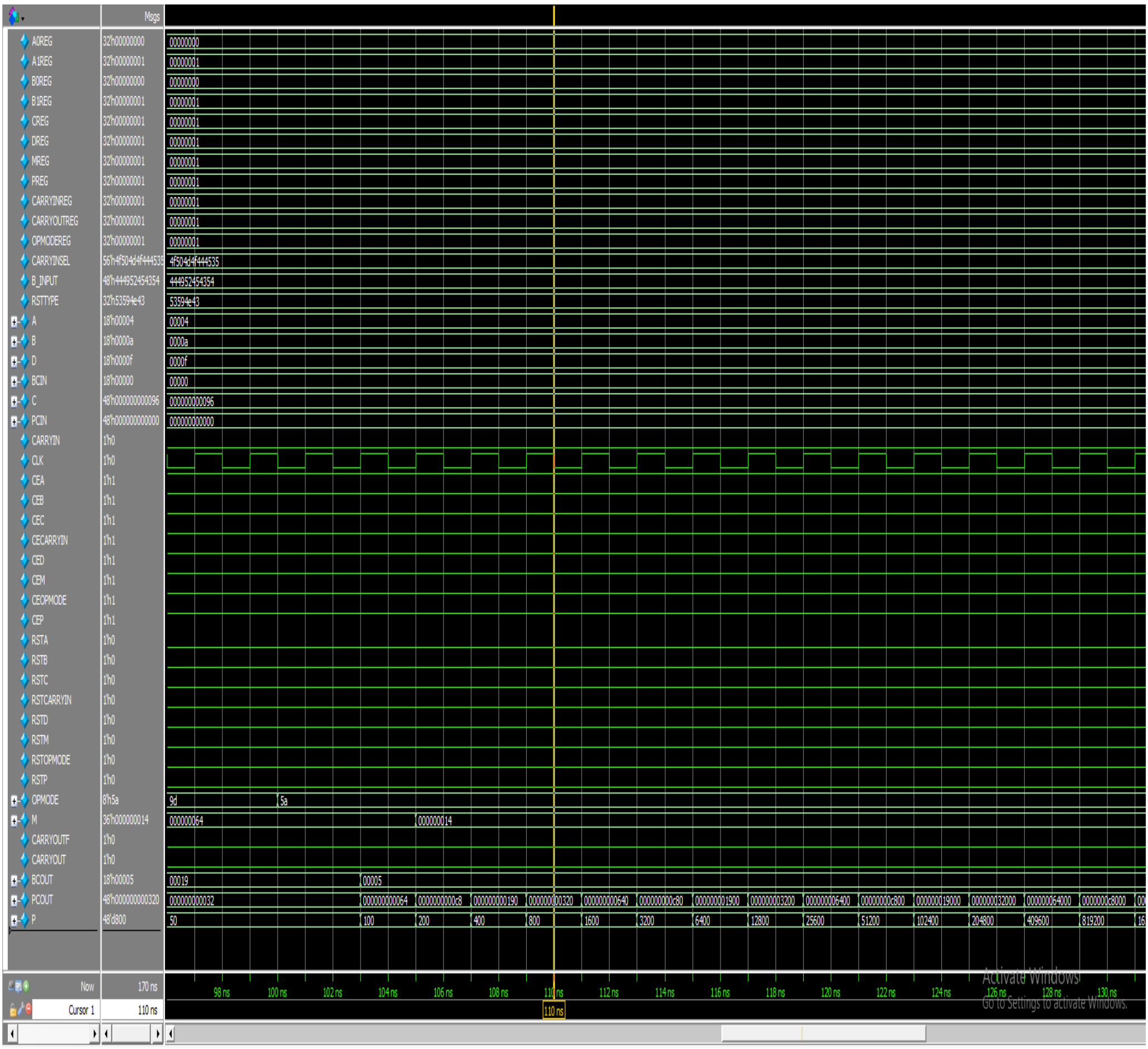
```
create_project project_DSP D:/kwtraining/final_project_1/code -part xc7a200tffg1156-1 -force

add_files project_1.v add_sub.v ffandmux.v

synth_design -rtl -top project_dsp > elab.log
# report_timing_summary -delay_type min_max -report_unconstrained -check_timing_verbose -max_paths 10 -input_pins -routable_nets -name timing_1>timimg rpt

write_schematic elaborated_schematic.pdf -format pdf -force

launch_runs synth_1 > synth.log

wait_on_run synth_1
open_run synth_1

write_schematic synthesized_schematic.pdf -format pdf -force

write_verilog -force DSP_netlist.v

launch_runs impl_1 > impl.log

wait_on_run impl_1
open_run impl_1

open_hw

connect_hw_server
```

Questasim:

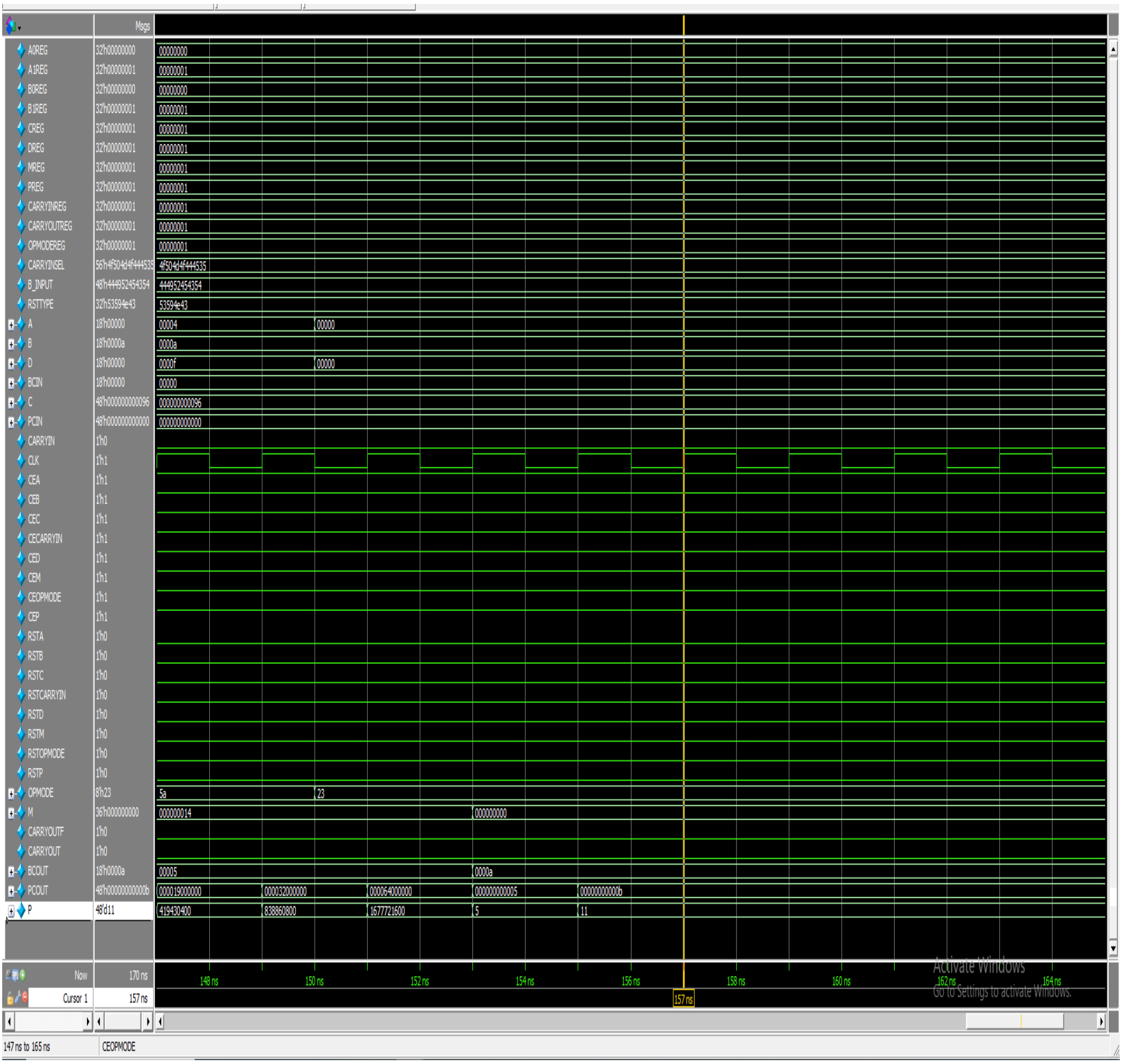
snippet for test and (P) RESULT WILL EQUAL 50 due to values that i put to inputs as you see:



by zoom in as you see i test the accumulation p in both mux X and Z so number will be double:

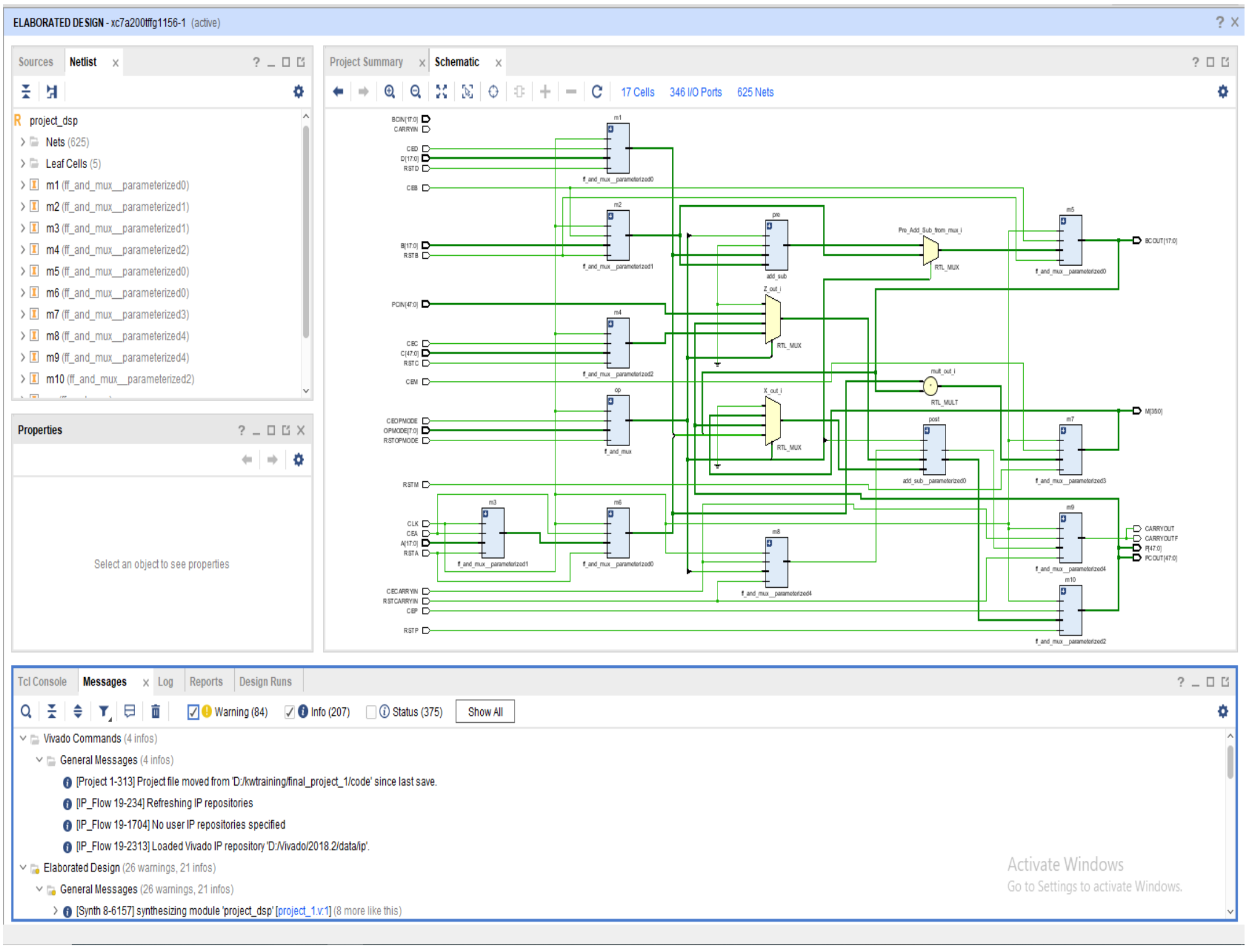


here i test concatenated and carry in =1 by opmode so result equal to 11:

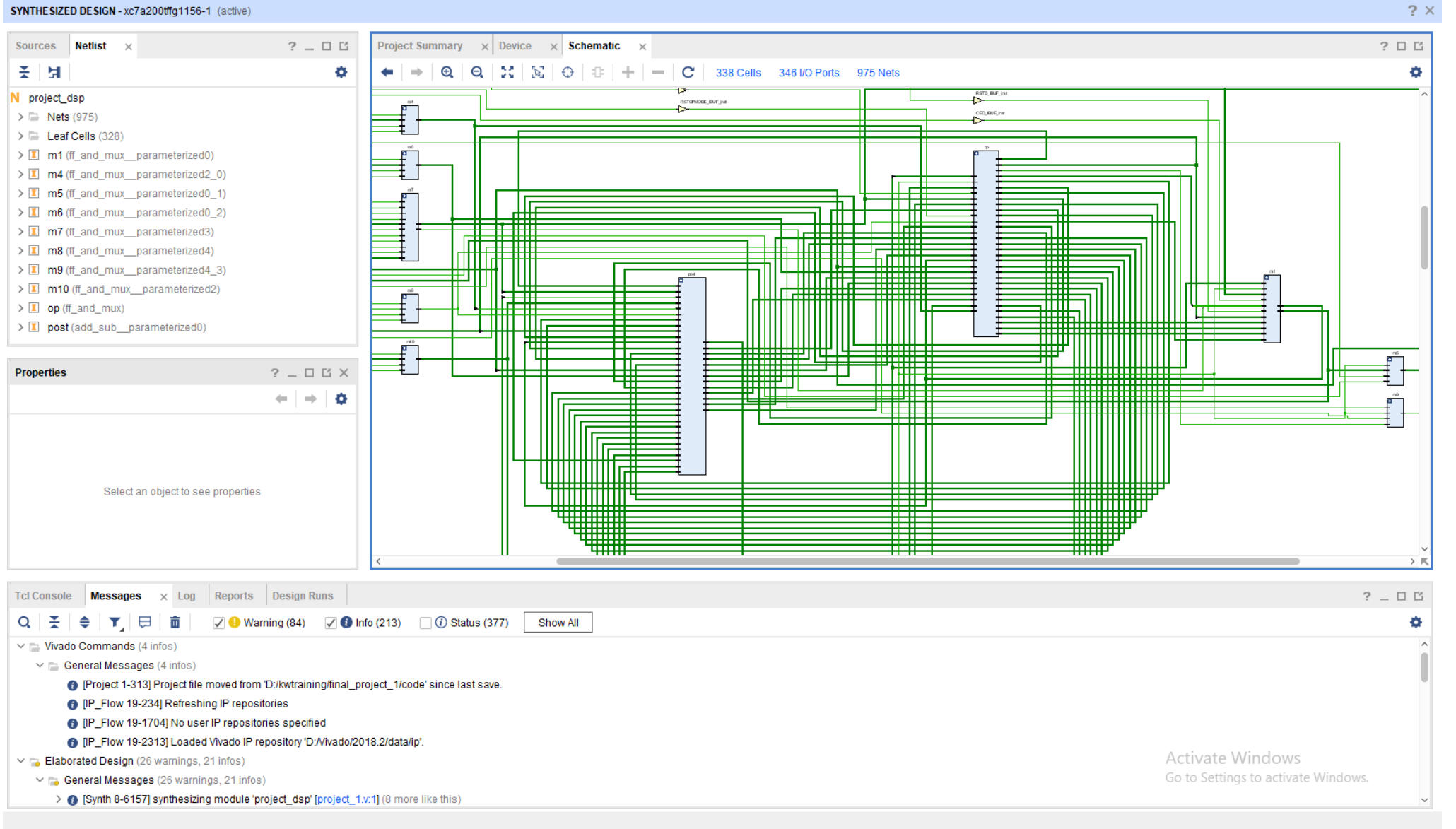


Vivado :

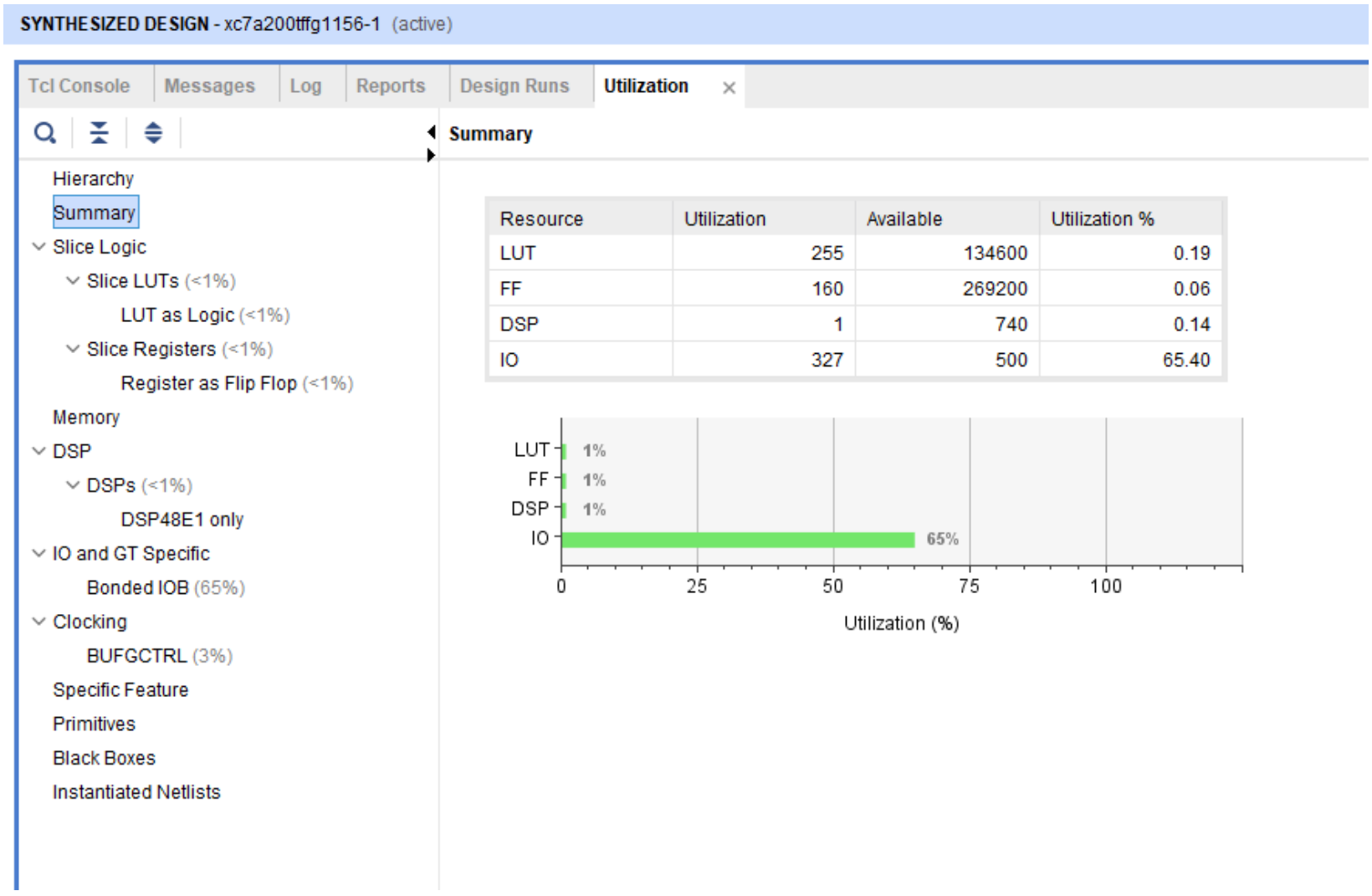
Schematic after elaborated without error :



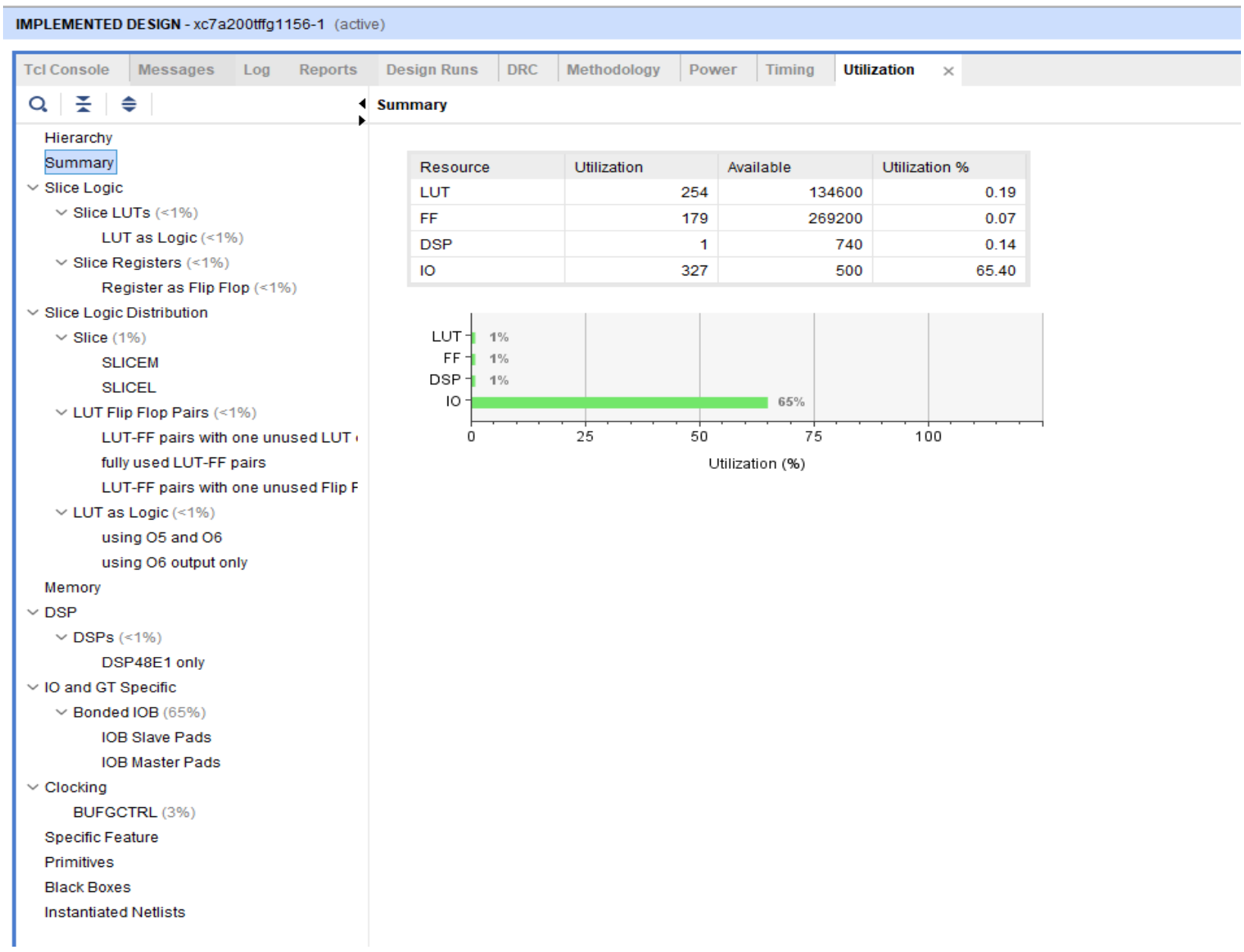
schematic after synthysis with no error :



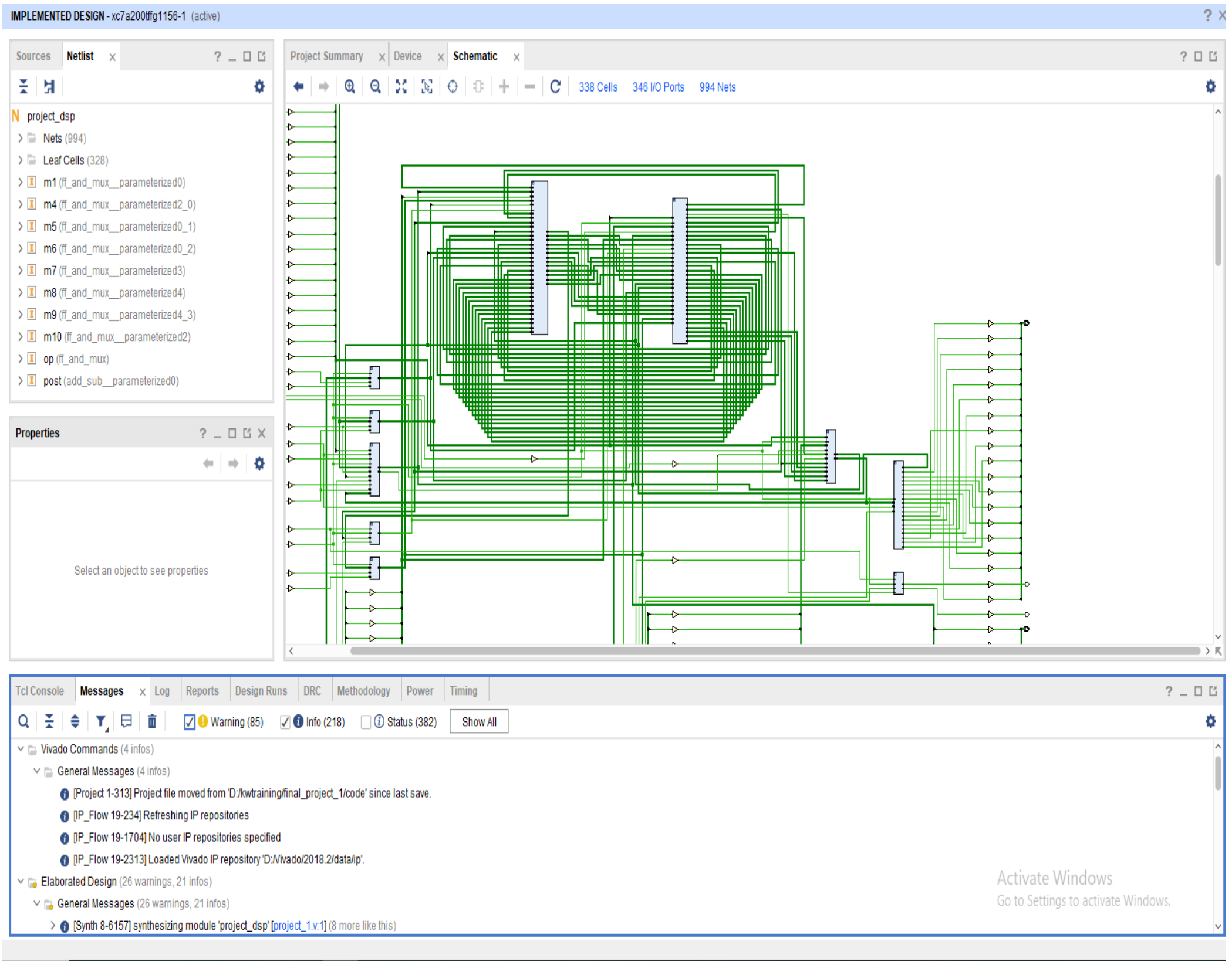
utllization after synthysis :



utullization after implementation:



implementaion with no errors and its schematic :



Device :