Spartan6-DSP48A1

by Karim Mohamed

Q1:

1) RTL code:

- **Top Module:**

```
module DSP48A1 #(
    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 [7:0] opmode,
    input rsta,rstb,rstm,rstp,rstc,rstd,rstcarryin,rstopmode,
    cea,ceb,cem,cep,cec,ced,cecarryin,ceopmode,clk,carryin,
    output [17:0] bcout,
    output [47:0] pcout,p,
    output [35:0] m,
    output carryout,carryoutf
);

    //opmode
    wire [7:0] opmode_mux,opmode_reg;
    dff  #(.WIDTH(8),.RSTTYPE(RSTTYPE))
OPMODE_REG(opmode,opmode_reg,clk,ceopmode,rstopmode);
    mux2 #(.WIDTH(8)) op_mux(opmode_reg,opmode,OPMODEREG,opmode_mux);

    //d block
    wire [17:0] d_reg,d_mux;
    dff  #(.WIDTH(18),.RSTTYPE(RSTTYPE)) D_REG (d,d_reg,clk,ced,rstd);
```

`

```
    mux2 #(.WIDTH(18)) m1(d_reg,d,DREG,d_mux);

    //b block
    wire [17:0] b0_reg,bin_mux,b0_mux;
    assign bin_mux = (B_INPUT == "DIRECT")? b : (B_INPUT =="CASCADE")? bcin : 0;
    dff  #(.WIDTH(18),.RSTTYPE(RSTTYPE)) B0_REG (bin_mux,b0_reg,clk,ceb,rstb);
    mux2 #(.WIDTH(18)) m2(b0_reg,bin_mux,B0REG,b0_mux);

    //a block
    wire [17:0] a0_reg,a0_mux;
    dff  #(.WIDTH(18),.RSTTYPE(RSTTYPE)) A0_REG (a,a0_reg,clk,cea,rsta);
    mux2 #(.WIDTH(18)) m3(a0_reg,a,A0REG,a0_mux);

    //c block
    wire [47:0] c_reg,c_mux;
    dff  #(.WIDTH(48),.RSTTYPE(RSTTYPE)) C_REG (c,c_reg,clk,cec,rstc);
    mux2 #(.WIDTH(48)) m4(c_reg,c,CREG,c_mux);

    //pre_adder/subtractor and b1 regester
    wire [17:0] padd_sub,padd_sub_mux,b1_reg,b1_mux,padd,psub;
    assign padd = d_mux + b0_mux;
    assign psub = d_mux - b0_mux;
    assign padd_sub = (opmode_mux[6])? psub : padd;
    mux2 #(.WIDTH(18)) m5(padd_sub,b0_mux,opmode_mux[4],padd_sub_mux);
    dff  #(.WIDTH(18),.RSTTYPE(RSTTYPE)) B1_REG (padd_sub_mux,b1_reg,clk,ceb,rstb);
    mux2 #(.WIDTH(18)) m6(b1_reg,padd_sub_mux,B1REG,b1_mux);
    assign bcout = b1_mux;

    //a1 regester
    wire [17:0] a1_reg,a1_mux;
    dff  #(.WIDTH(18),.RSTTYPE(RSTTYPE)) A1_REG (a0_mux,a1_reg,clk,cea,rsta);
    mux2 #(.WIDTH(18)) m7(a1_reg,a0_mux,A1REG,a1_mux);

    //multiplication block
    wire [35:0] m_res,m_reg,m_mux;
    assign m_res = a1_mux*b1_mux;
    dff  #(.WIDTH(36),.RSTTYPE(RSTTYPE)) M_REG (m_res,m_reg,clk,cem,rstm);
    mux2 #(.WIDTH(36)) m8(m_reg,m_res,MREG,m_mux);
    assign m = (~(~m_mux));

    //mux x
    wire [47:0] x_mux;
    mux4 #(.WIDTH(48))
m9(48'b0,{12'b0,m_mux},p,{d_mux[11:0],a[17:0],b[17:0]},opmode_mux[1:0],x_mux);

    //mux z
    wire [47:0] z_mux;
    mux4 #(.WIDTH(48)) m10(48'b0,pcin,p,c_mux,opmode_mux[3:2],z_mux);

    //carry in
    wire cin0_mux,cin_reg,cin1_mux;
    assign cin0_mux = (CARRYINSEL == "OPMODE5")? opmode_mux[5] : (CARRYINSEL ==
"CARRYIN")? carryin : 0;
```

```
    dff #(.WIDTH(1),.RSTTYPE(RSTTYPE)) CYI(cin0_mux,cin_reg,clk,cecarryin,rstcarryin);
    mux2 #(.WIDTH(1)) m11(cin_reg,cin0_mux,CARRYINREG,cin1_mux);

    //post adder
    wire [47:0] post_add_sub,post_add_sub_reg,post_add_sub_mux,post_add,post_sub;
    wire cout,cout_add,cout_sub;
    assign {cout_add,post_add} = z_mux+x_mux+{47'b0,cin1_mux};
    assign {cout_sub,post_sub} = z_mux-x_mux-{47'b0,cin1_mux};
    assign {cout,post_add_sub}= (opmode_mux[7])? {cout_sub,post_sub} :
{cout_add,post_add};
    dff #(.WIDTH(48),.RSTTYPE(RSTTYPE)) P_REG(post_add_sub,post_add_sub_reg,clk,cep,rstp);
    mux2 #(.WIDTH(48)) m12(post_add_sub_reg,post_add_sub,PREG,post_add_sub_mux);
    assign p     = post_add_sub_mux;
    assign pcout = post_add_sub_mux;

    //carryout mux
    wire cout_reg,cout_mux;
    dff #(.WIDTH(1),.RSTTYPE(RSTTYPE)) CYO(cout,cout_reg,clk,cecarryin,rstcarryin);
    mux2 #(.WIDTH(1)) m13(cout_reg,cout,CARRYOUTREG,cout_mux);
    assign carryout  = cout_mux;
    assign carryoutf = cout_mux;
endmodule
```

- **DFF Module:**

```
module dff(d,q,clk,enable,rst);
    parameter WIDTH = 18;
    parameter RSTTYPE = "SYNC";
    input [WIDTH-1:0] d;
    input clk,enable,rst;
    output reg [WIDTH-1:0] q;
    generate
        if(RSTTYPE == "SYNC")
            always @(posedge clk) begin
                if(enable)
                    q <= (rst)? 0:d;
            end
        else if(RSTTYPE == "ASYNC")
            always @(posedge clk or posedge rst) begin
                if(rst)
                    q <= 0;
                else if (enable)
                    q <= d;
            end
    endgenerate
endmodule
```

`

- **MUX2 Module:**

```
module mux2 (a,b,sel,out);
    parameter WIDTH = 18 ;
    input [WIDTH-1:0] a,b;
    input sel;
    output [WIDTH-1:0] out;
    assign out = (sel)? a:b;
endmodule
```

- **MUX4 Module:**

```
module mux4(a,b,c,d,sel,out);
    parameter WIDTH = 18 ;
    input [WIDTH-1:0] a,b,c,d;
    input [1:0] sel;
    output reg [WIDTH-1:0] out;
    always @(*) begin
        case(sel)
            0: out = a;
            1: out = b;
            2: out = c;
            3: out = d;
        endcase
    end
endmodule
```

Note: DFF, MUX2, and MUX4 were tested in previous assignments, so I skipped testing their control signals.

2) Testbench code:

```
module DSP48A1_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 [7:0] opmode;
```

`

```
reg
    rsta,
    rstb,
    rstm,
    rstp,
    rstc,
    rstd,
    rstcarryin,
    rstopmode,
    cea,
    ceb,
    cem,
    cep,
    ccec,
    ced,
    cecarryin,
    ceopmode,
    clk,
    carryin;
wire [17:0] bcout_dut;
wire [47:0] pcout_dut, p_dut;
wire [35:0] m_dut;
wire carryout_dut, carryoutf_dut;

//module instantiation
DSP48A1 #(
    .A0REG(A0REG),
    .A1REG(A1REG),
    .B0REG(B0REG),
    .B1REG(B1REG),
    .CREG(CREG),
    .DREG(DREG),
    .MREG(MREG),
    .PREG(PREG),
    .CARRYINREG(CARRYINREG),
    .CARRYOUTREG(CARRYOUTREG),
    .OPMODEREG(OPMODEREG),
    .CARRYINSEL(CARRYINSEL),
    .B_INPUT(B_INPUT),
    .RSTTYPE(RSTTYPE)
) DUT (
    .a(a),
    .b(b),
    .c(c),
    .d(d),
    .pcin(pcin),
    .bcin(bcin),
    .opmode(opmode),
    .rsta(rsta),
    .rstb(rstb),
    .rstc(rstc),
    .rstd(rstd),
    .rstm(rstm),
```

```
    .rstp(rstp),
    .rstcarryin(rstcarryin),
    .rstopmode(rstopmode),
    .cea(cea),
    .ceb(ceb),
    .cem(cem),
    .cep(cep),
    .ccec(ccec),
    .ced(ced),
    .cecarryin(cecarryin),
    .ceopmode(ceopmode),
    .clk(clk),
    .carryin(carryin),
    .bcout(bcout_dut),
    .pcout(pcout_dut),
    .p(p_dut),
    .m(m_dut),
    .carryout(carryout_dut),
    .carryoutf(carryoutf_dut)
);
reg [47:0] x, z, post_add_sub_exp, post_add, post_sub;
reg [35:0] m_exp;
reg [17:0] bcout_exp;
reg cout_exp;
reg [47:0] pout_temp;

initial begin
  clk = 0;
  forever #1 clk = ~clk;
end

//test generator
initial begin
  cea        = 1;
  ceb        = 1;
  cem        = 1;
  cep        = 1;
  ccec       = 1;
  ced        = 1;
  cecarryin  = 1;
  ceopmode   = 1;
  rsta       = 1;
  rstb       = 1;
  rstc       = 1;
  rstd       = 1;
  rstm       = 1;
  rstp       = 1;
  rstcarryin = 1;
  rstopmode  = 1;
  @(negedge clk);
  rsta       = 0;
  rstb       = 0;
  rstc       = 0;
```

```
    rstd       = 0;
    rstm       = 0;
    rstp       = 0;
    rstcarryin = 0;
    rstopmode  = 0;
    repeat (1000) begin
      a       = $random;
      b       = $random;
      c       = $random;
      d       = $random;
      carryin = $random;  // opmode[5] should take its turn as CARRYINSEL = "OPMODE5"
      pcin    = $random;
      opmode  = $random;
      if (opmode[3:2] == 2) opmode[3:2] = 3;
      if (opmode[1:0] == 2) opmode[1:0] = 3;
      bcout_exp = (opmode[6]) ? (d - b) : (d + b);
      bcout_exp = (opmode[4]) ? bcout_exp : b;
      m_exp = a * bcout_exp;
      case (opmode[1:0])
        2'b00: x = 48'b0;
        2'b01: x = m_exp;
        2'b10: x = p_dut;
        2'b11: x = {d[11:0], a[17:0], b[17:0]};
      endcase
      case (opmode[3:2])
        2'b00: z = 48'b0;
        2'b01: z = pcin;
        2'b10: z = p_dut;
        2'b11: z = c;
      endcase
      post_add = z + x + {47'b0, opmode[5]};
      post_sub = z - x - {47'b0, opmode[5]};
      {cout_exp,post_add_sub_exp} = (opmode[7])? z-x-{47'b0,opmode[5]} :
z+x+{47'b0,opmode[5]};
      repeat (4) @(negedge clk);
      if (bcout_dut != bcout_exp) begin
          $display("Error in bcout value");
          $stop;
      end
      if (m_dut != m_exp) begin
          $display("Error in the multiplier output");
          $stop;
      end
      if (p_dut != post_add_sub_exp || pcout_dut != p_dut) begin
          $display("Error in the p output");
          $stop;
      end
      if (carryout_dut != cout_exp || carryout_dut != carryoutf_dut) begin
          $display("Error in the carryout output");
          $stop;
      end
      end
    $display("the DSP passed the opmodes 0,1,3");
```

```
    rsta        = 1;
    rstb        = 1;
    rstc        = 1;
    rstd        = 1;
    rstm        = 1;
    rstp        = 1;
    rstcarryin  = 1;
    rstopmode   = 1;
    @(negedge clk);
    rsta        = 0;
    rstb        = 0;
    rstc        = 0;
    rstd        = 0;
    rstm        = 0;
    rstp        = 0;
    rstcarryin  = 0;
    rstopmode   = 0;
    c           = 50;
    opmode[1:0] = 0;
    opmode[3:2] = 3;
    opmode[5]   = 0;
    opmode[7]   = 0;
    repeat (2) @(negedge clk);
    if (p_dut != 50) begin
      $display("Error in opmode 3");
      $stop;
    end
    pout_temp   = p_dut;
    opmode[1:0] = 2;
    opmode[3:2] = 2;
    repeat (2) @(negedge clk);
    $display("the DSP passed the tests");
    $stop;
  end
endmodule
```

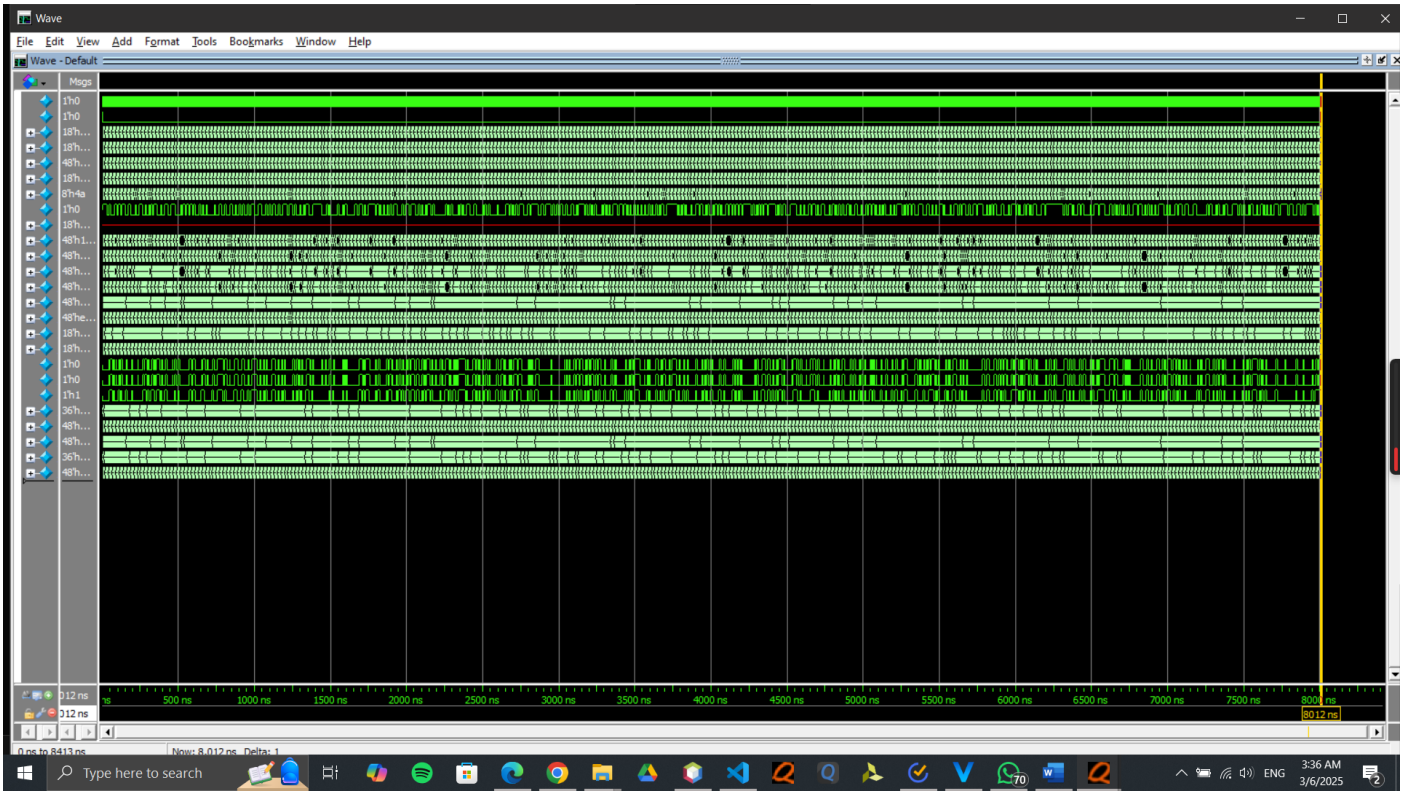
`

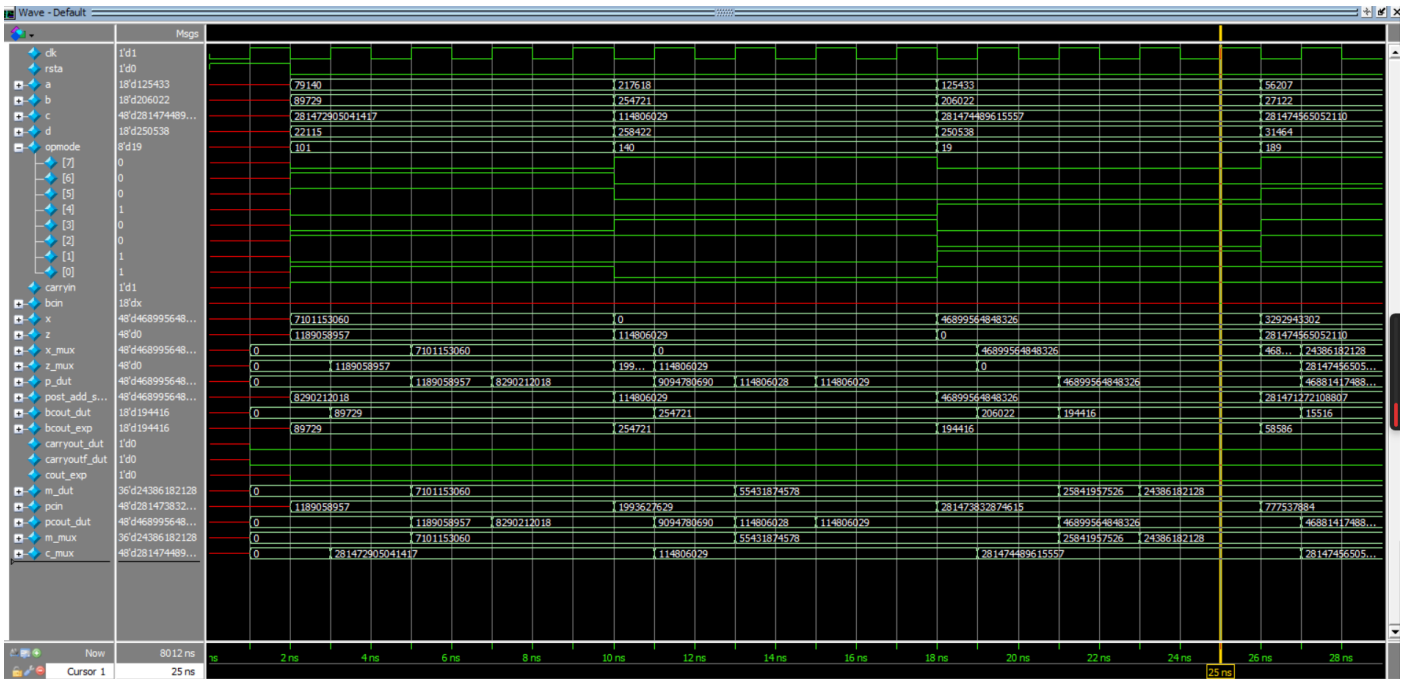
3) DO file:

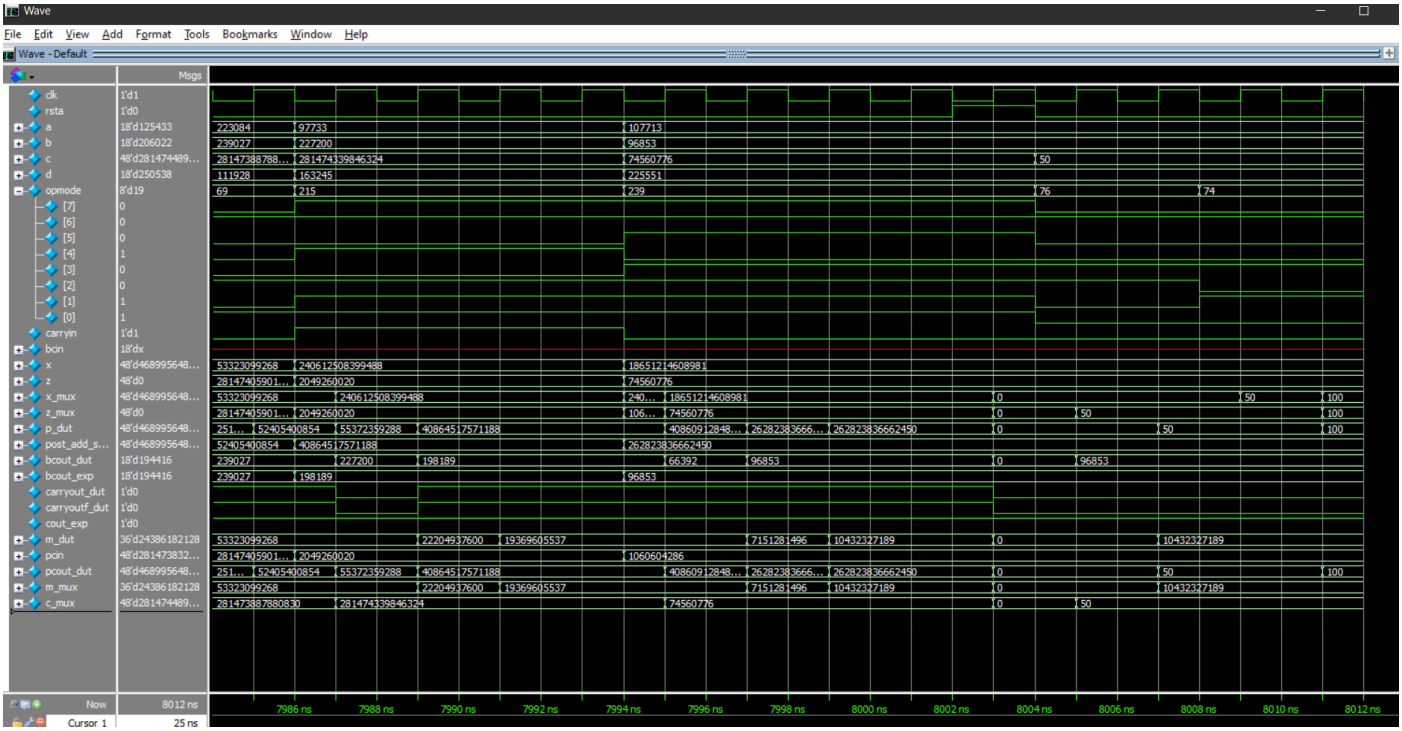
```
vlib work

vlog design.v testbench.v DFF.v mux2.v mux4.v

vsim -voptargs=+acc work.DSP48A1_tb

add wave -position insertpoint  \
sim:/DSP48A1_tb/clk \
sim:/DSP48A1_tb/rsta \
sim:/DSP48A1_tb/a \
sim:/DSP48A1_tb/b \
sim:/DSP48A1_tb/c \
sim:/DSP48A1_tb/d \
sim:/DSP48A1_tb/opmode \
sim:/DSP48A1_tb/carryin \
sim:/DSP48A1_tb/bcin \
sim:/DSP48A1_tb/x \
sim:/DSP48A1_tb/z \
sim:/DSP48A1_tb/DUT/x_mux \
sim:/DSP48A1_tb/DUT/z_mux \
sim:/DSP48A1_tb/p_dut \
sim:/DSP48A1_tb/post_add_sub_exp \
sim:/DSP48A1_tb/bcout_dut \
sim:/DSP48A1_tb/bcout_exp \
sim:/DSP48A1_tb/carryout_dut \
sim:/DSP48A1_tb/carryoutf_dut \
sim:/DSP48A1_tb/cout_exp \
sim:/DSP48A1_tb/m_dut \
sim:/DSP48A1_tb/pcin \
sim:/DSP48A1_tb/pcout_dut \
sim:/DSP48A1_tb/DUT/m_mux  \
sim:/DSP48A1_tb/DUT/c_mux

run -all
```

`

4) Questasim Snippets:







5) Constraint File:

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level
signal names in the project

## Clock signal
set_property -dict {PACKAGE_PIN W5 IOSTANDARD LVCMOS33} [get_ports clk]
create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [get_ports clk]


## Switches
# set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {sw[0]}]
# set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {sw[1]}]
# set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {sw[2]}]
# set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {sw[3]}]
#set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {sw[4]}]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {sw[5]}]
#set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports {sw[6]}]
#set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {sw[7]}]
#set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports {sw[8]}]
#set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports {sw[9]}]
#set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports {sw[10]}]
#set_property -dict { PACKAGE_PIN R3    IOSTANDARD LVCMOS33 } [get_ports {sw[11]}]
#set_property -dict { PACKAGE_PIN W2    IOSTANDARD LVCMOS33 } [get_ports {sw[12]}]
#set_property -dict { PACKAGE_PIN U1    IOSTANDARD LVCMOS33 } [get_ports {sw[13]}]
#set_property -dict { PACKAGE_PIN T1    IOSTANDARD LVCMOS33 } [get_ports {sw[14]}]
#set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports {sw[15]}]
```

```
## LEDs
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {led[0]}]
#set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports {led[1]}]
#set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports {led[2]}]
#set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports {led[3]}]
#set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports {led[4]}]
#set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 } [get_ports {led[5]}]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
#set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
#set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
#set_property -dict { PACKAGE_PIN W3    IOSTANDARD LVCMOS33 } [get_ports {led[10]}]
#set_property -dict { PACKAGE_PIN U3    IOSTANDARD LVCMOS33 } [get_ports {led[11]}]
#set_property -dict { PACKAGE_PIN P3    IOSTANDARD LVCMOS33 } [get_ports {led[12]}]
#set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports {led[13]}]
#set_property -dict { PACKAGE_PIN P1    IOSTANDARD LVCMOS33 } [get_ports {led[14]}]
#set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports {led[15]}]


##7 Segment Display
#set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
#set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
#set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
#set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
#set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
#set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
#set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]

#set_property -dict { PACKAGE_PIN V7   IOSTANDARD LVCMOS33 } [get_ports dp]

#set_property -dict { PACKAGE_PIN U2   IOSTANDARD LVCMOS33 } [get_ports {an[0]}]
#set_property -dict { PACKAGE_PIN U4   IOSTANDARD LVCMOS33 } [get_ports {an[1]}]
#set_property -dict { PACKAGE_PIN V4   IOSTANDARD LVCMOS33 } [get_ports {an[2]}]
#set_property -dict { PACKAGE_PIN W4   IOSTANDARD LVCMOS33 } [get_ports {an[3]}]


##Buttons
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports rst_n]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports btnU]
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports btnL]
#set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports btnR]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports btnD]


##Pmod Header JA
#set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports {JA[0]}];#Sch name
= JA1
#set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports {JA[1]}];#Sch name
= JA2
#set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports {JA[2]}];#Sch name
= JA3
#set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports {JA[3]}];#Sch name
= JA4
```

`

```
#set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports {JA[4]}];#Sch name
= JA7
#set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports {JA[5]}];#Sch name
= JA8
#set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports {JA[6]}];#Sch name
= JA9
#set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports {JA[7]}];#Sch name
= JA10

##Pmod Header JB
#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports {JB[0]}];#Sch
name = JB1
#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports {JB[1]}];#Sch
name = JB2
#set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVCMOS33 } [get_ports {JB[2]}];#Sch
name = JB3
#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports {JB[3]}];#Sch
name = JB4
#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports {JB[4]}];#Sch
name = JB7
#set_property -dict { PACKAGE_PIN A17   IOSTANDARD LVCMOS33 } [get_ports {JB[5]}];#Sch
name = JB8
#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports {JB[6]}];#Sch
name = JB9
#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports {JB[7]}];#Sch
name = JB10

##Pmod Header JC
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports {JC[0]}];#Sch
name = JC1
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports {JC[1]}];#Sch
name = JC2
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports {JC[2]}];#Sch
name = JC3
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports {JC[3]}];#Sch
name = JC4
#set_property -dict { PACKAGE_PIN L17   IOSTANDARD LVCMOS33 } [get_ports {JC[4]}];#Sch
name = JC7
#set_property -dict { PACKAGE_PIN M19   IOSTANDARD LVCMOS33 } [get_ports {JC[5]}];#Sch
name = JC8
#set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports {JC[6]}];#Sch
name = JC9
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports {JC[7]}];#Sch
name = JC10

##Pmod Header JXADC
#set_property -dict { PACKAGE_PIN J3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[0]}];#Sch
name = XA1_P
#set_property -dict { PACKAGE_PIN L3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[1]}];#Sch
name = XA2_P
#set_property -dict { PACKAGE_PIN M2   IOSTANDARD LVCMOS33 } [get_ports {JXADC[2]}];#Sch
name = XA3_P
```

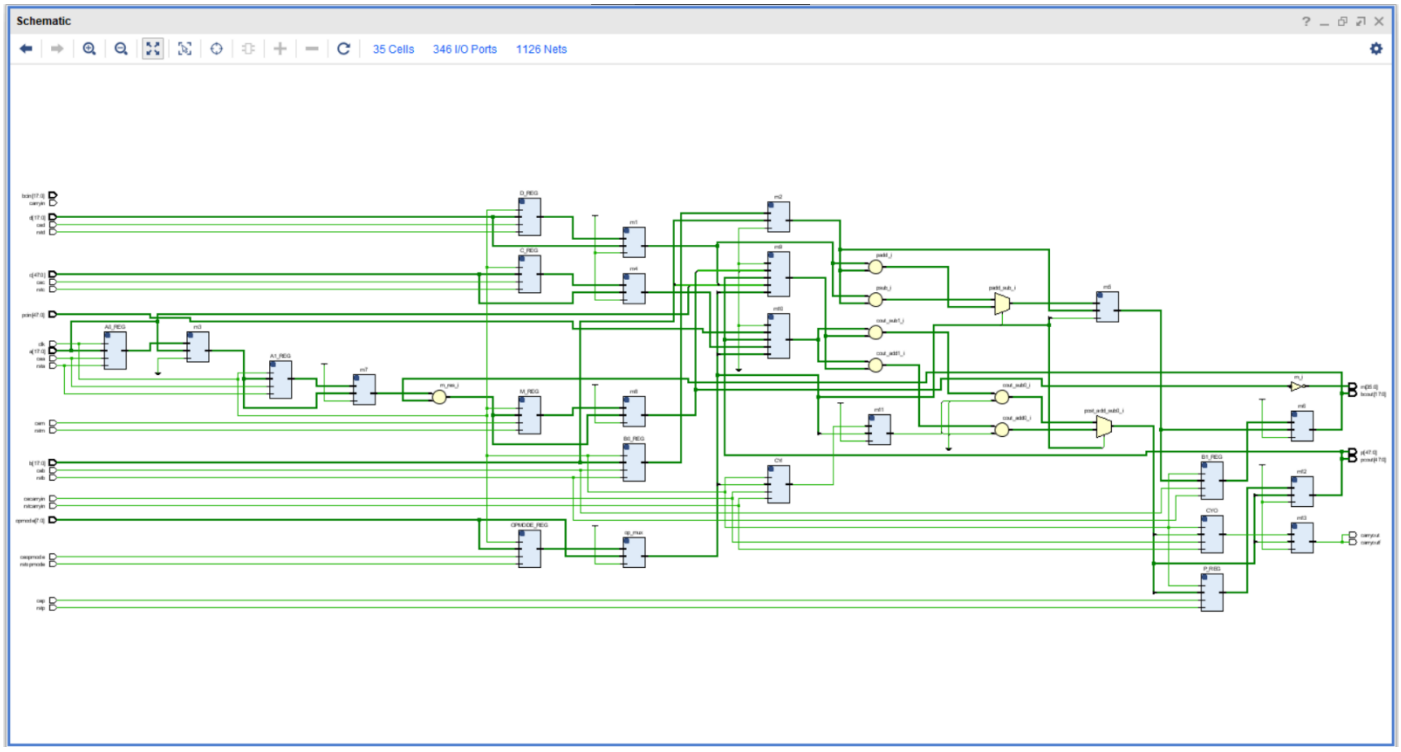
`

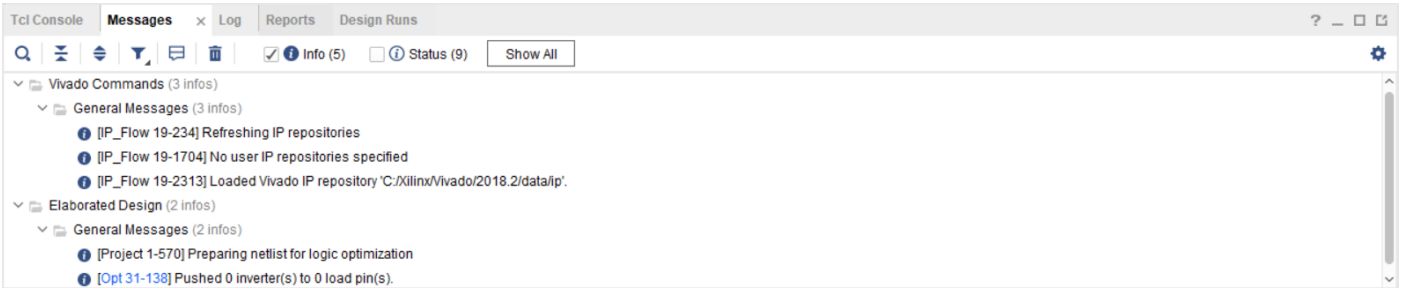
```
#set_property -dict { PACKAGE_PIN N2   IOSTANDARD LVCMOS33 } [get_ports {JXADC[3]}];#Sch
name = XA4_P
#set_property -dict { PACKAGE_PIN K3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[4]}];#Sch
name = XA1_N
#set_property -dict { PACKAGE_PIN M3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[5]}];#Sch
name = XA2_N
#set_property -dict { PACKAGE_PIN M1   IOSTANDARD LVCMOS33 } [get_ports {JXADC[6]}];#Sch
name = XA3_N
#set_property -dict { PACKAGE_PIN N1   IOSTANDARD LVCMOS33 } [get_ports {JXADC[7]}];#Sch
name = XA4_N


##VGA Connector
#set_property -dict { PACKAGE_PIN G19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[0]}]
#set_property -dict { PACKAGE_PIN H19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[1]}]
#set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[2]}]
#set_property -dict { PACKAGE_PIN N19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[3]}]
#set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[0]}]
#set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[1]}]
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[2]}]
#set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[3]}]
#set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[0]}]
#set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[1]}]
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[2]}]
#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[3]}]
#set_property -dict { PACKAGE_PIN P19   IOSTANDARD LVCMOS33 } [get_ports Hsync]
#set_property -dict { PACKAGE_PIN R19   IOSTANDARD LVCMOS33 } [get_ports Vsync]


##USB-RS232 Interface
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports RsRx]
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports RsTx]


##USB HID (PS/2)
#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports
PS2Clk]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports
PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the
##STARTUPE2 primitive.
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[0]}]
#set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[1]}]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[2]}]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[3]}]
#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports QspiCSn]


## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
```

`

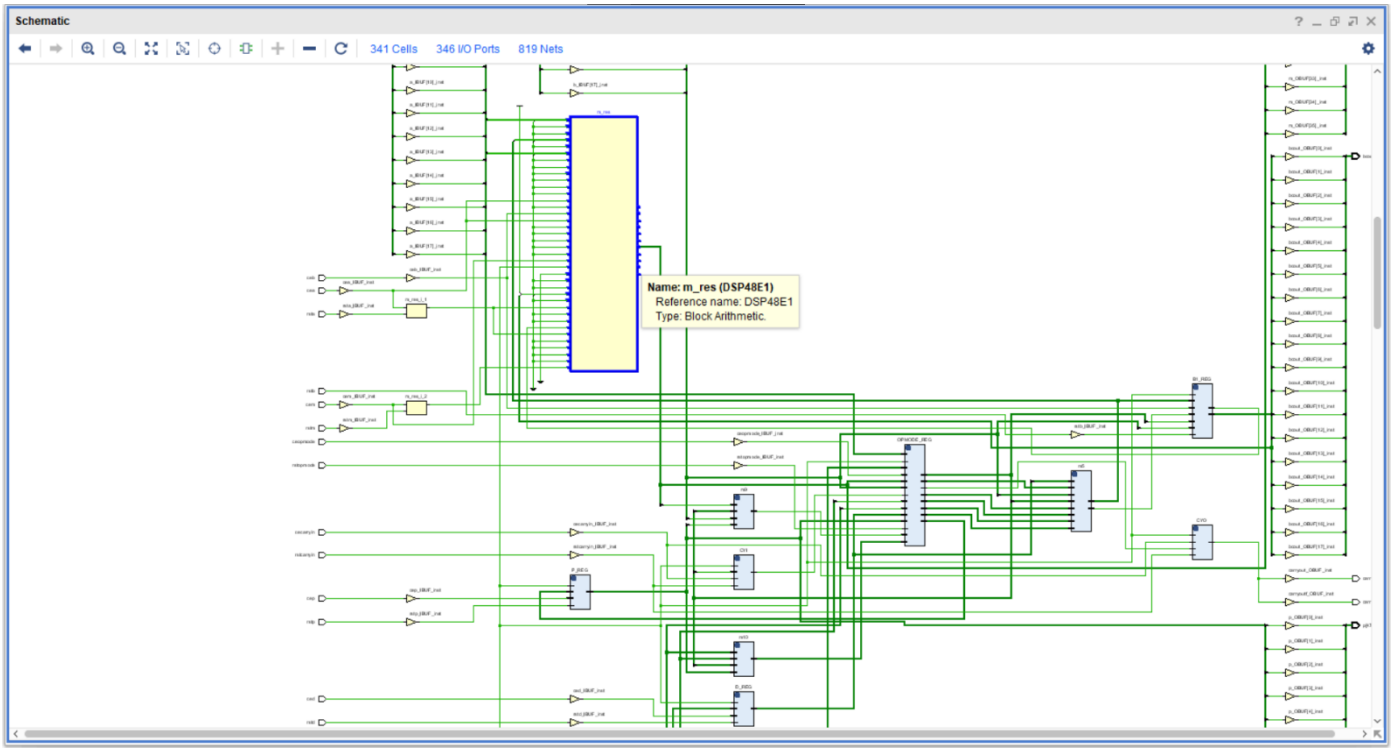
```
set_property CFGBVS VCCO [current_design]

## SPI configuration mode options for QSPI boot, can be used for all designs
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
```

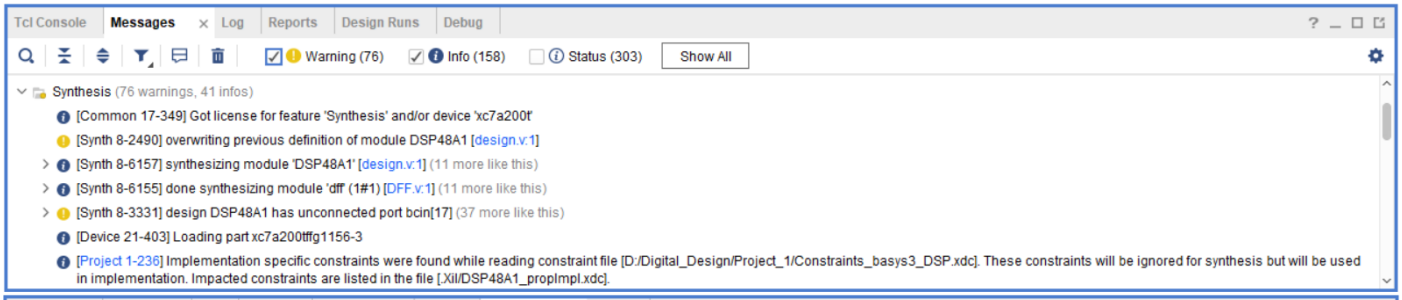
6) Elaboration:

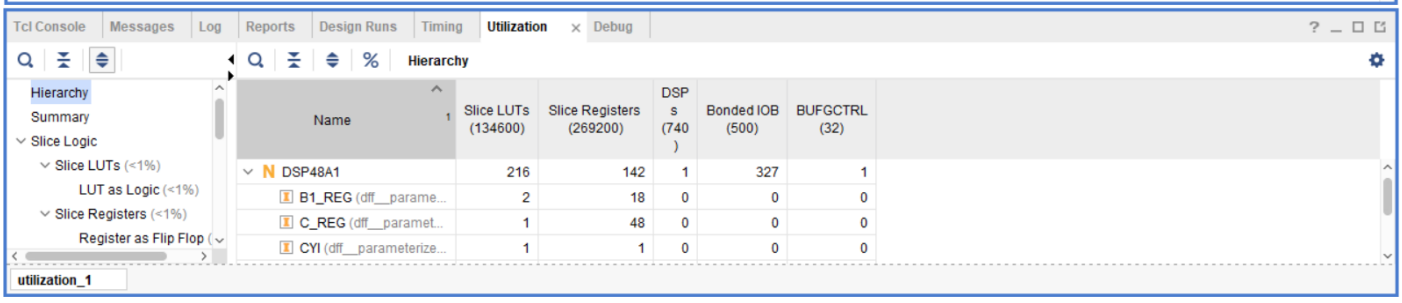


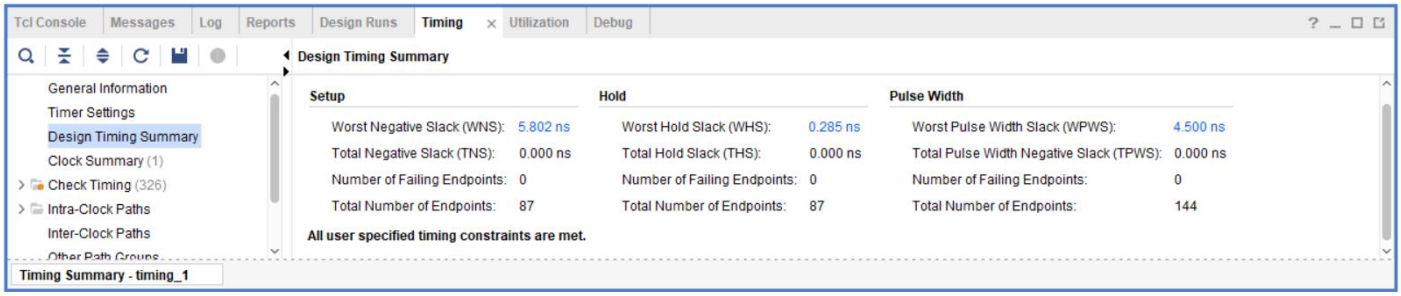


7) Synthesis:

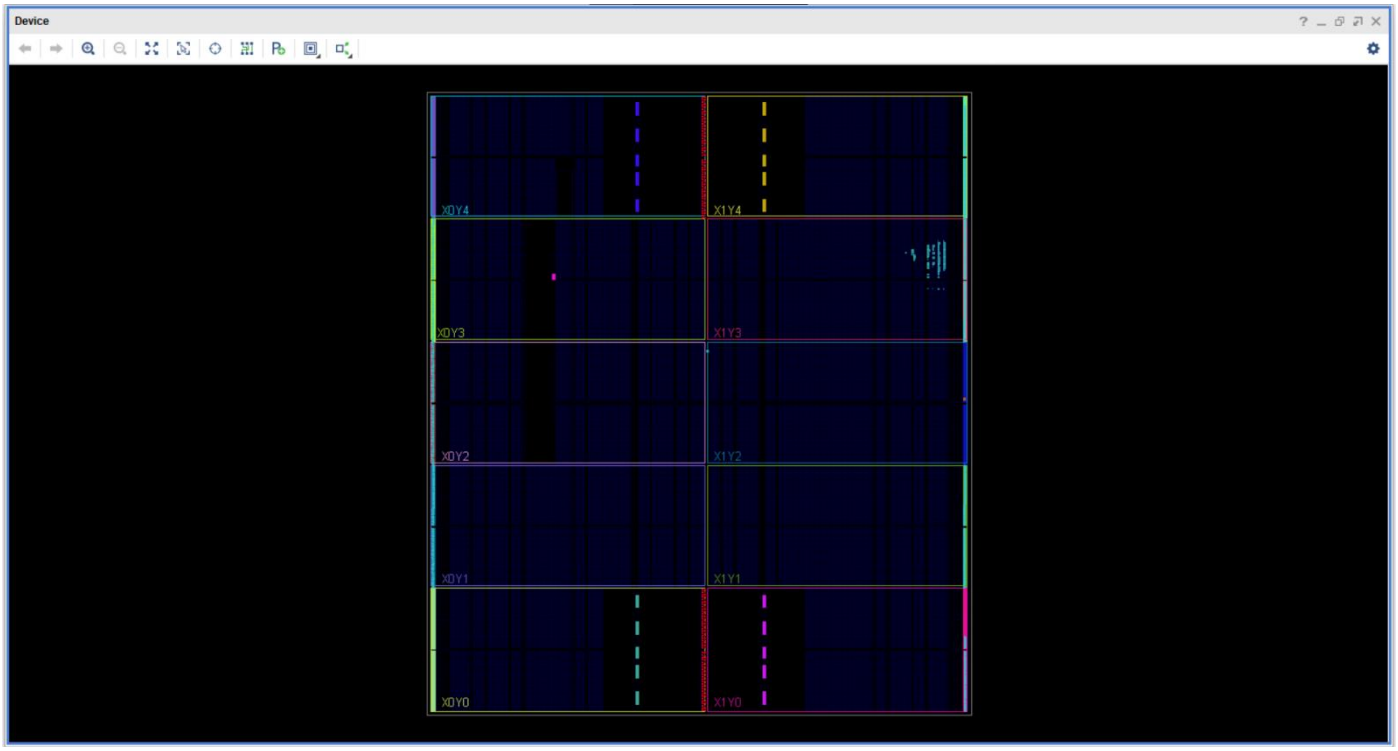

Schematic
341 Cells
346 I/O Ports
819 Nets
Name: m_res (DSP48E1)
Reference name: DSP48E1
Type: Block Arithmetic.

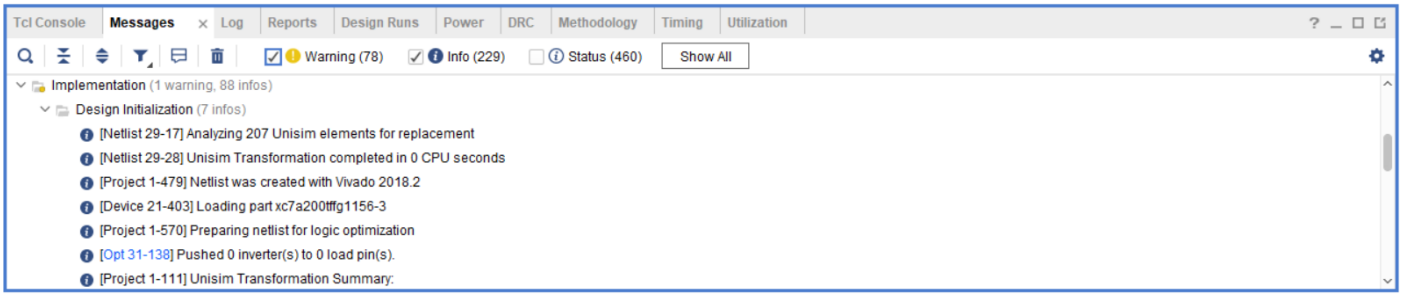

Tcl Console
Messages
Log
Reports
Design Runs
Debug
Warning (76)
Info (158)
Status (303)
Show All
Synthesis (76 warnings, 41 infos)
[Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a200t'
[Synth 8-2490] overwriting previous definition of module DSP48A1 [design.v:1]
[Synth 8-6157] synthesizing module 'DSP48A1' [design.v:1] (11 more like this)
[Synth 8-6155] done synthesizing module 'dff' (1#1) [DFF.v:1] (11 more like this)
[Synth 8-3331] design DSP48A1 has unconnected port bcin[17] (37 more like this)
[Device 21-403] Loading part xc7a200tffg1156-3
[Project 1-236] Implementation specific constraints were found while reading constraint file [D:/Digital_Design/Project_1/Constraints_basys3_DSP.xdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [.Xil/DSP48A1_propImpl.xdc].

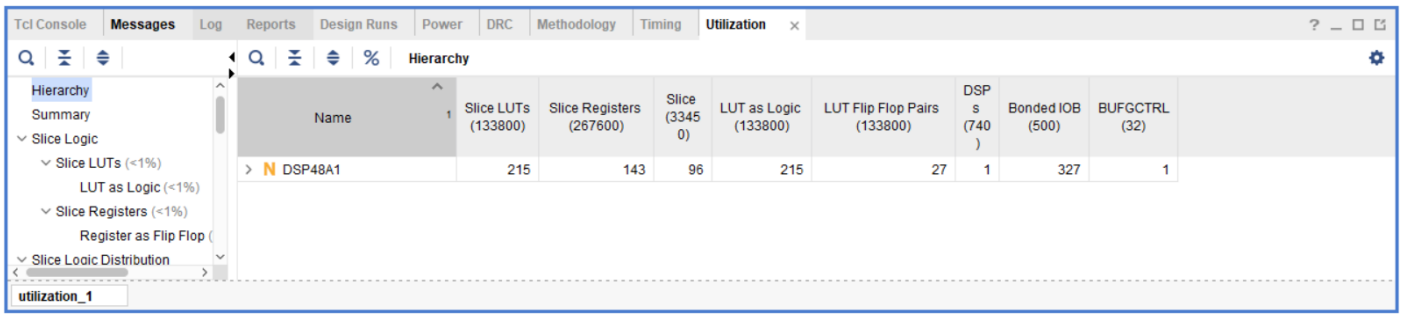

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Utilization
Debug
Hierarchy
Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Flip Flop
Name
Slice LUTs (134600)
Slice Registers (269200)
DSPs (740)
Bonded IOB (500)
BUFGCTRL (32)
DSP48A1 216 142 1 327 1
B1_REG (dff__parame... 2 18 0 0 0
C_REG (dff__paramet... 1 48 0 0 0
CYI (dff__parameterize... 1 1 0 0 0
utilization_1

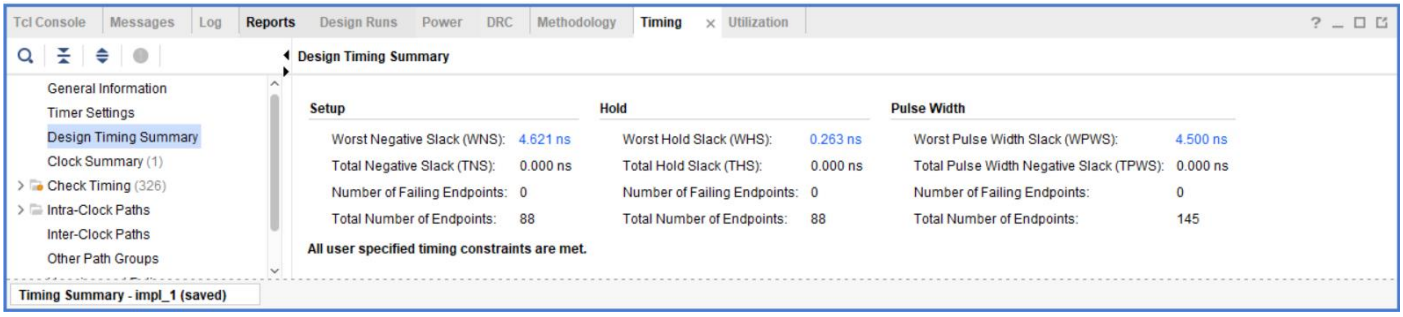

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Utilization
Debug
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (326)
Intra-Clock Paths
Inter-Clock Paths
Setup
Worst Negative Slack (WNS): 5.802 ns
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 87
Hold
Worst Hold Slack (WHS): 0.285 ns
Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 87
Pulse Width
Worst Pulse Width Slack (WPWS): 4.500 ns
Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 144
All user specified timing constraints are met.
Timing Summary - timing_1

8) Implementation:

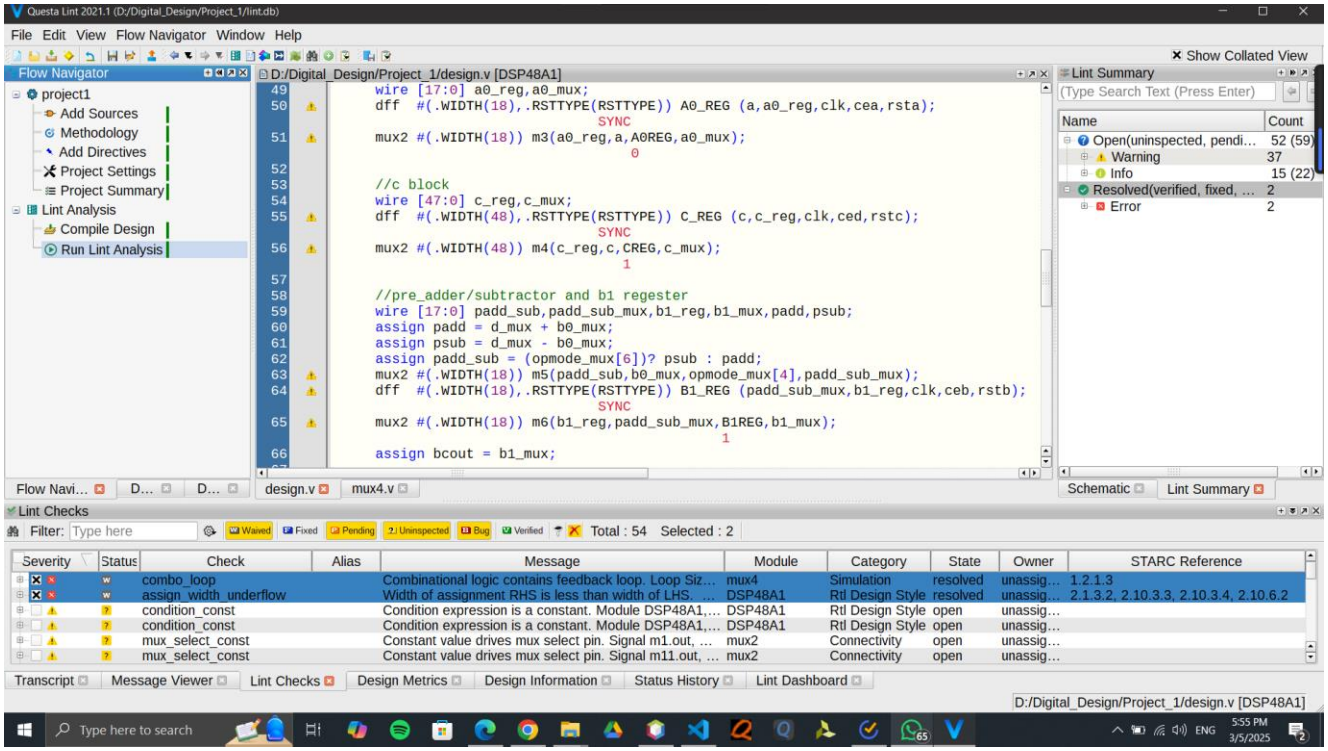








9) Linting:



Note: these errors by the linting tools doesn't affect on the design but they are intentional.