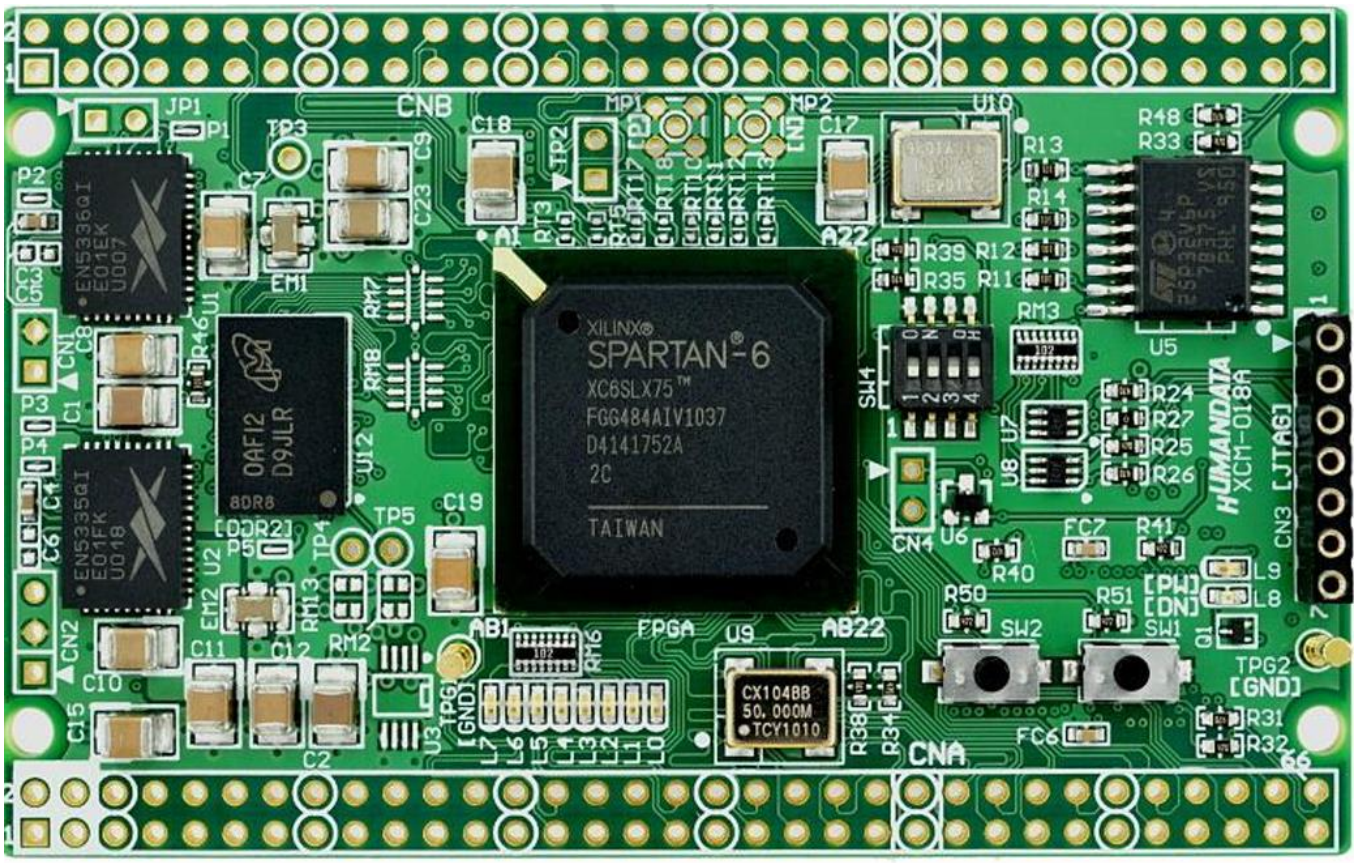
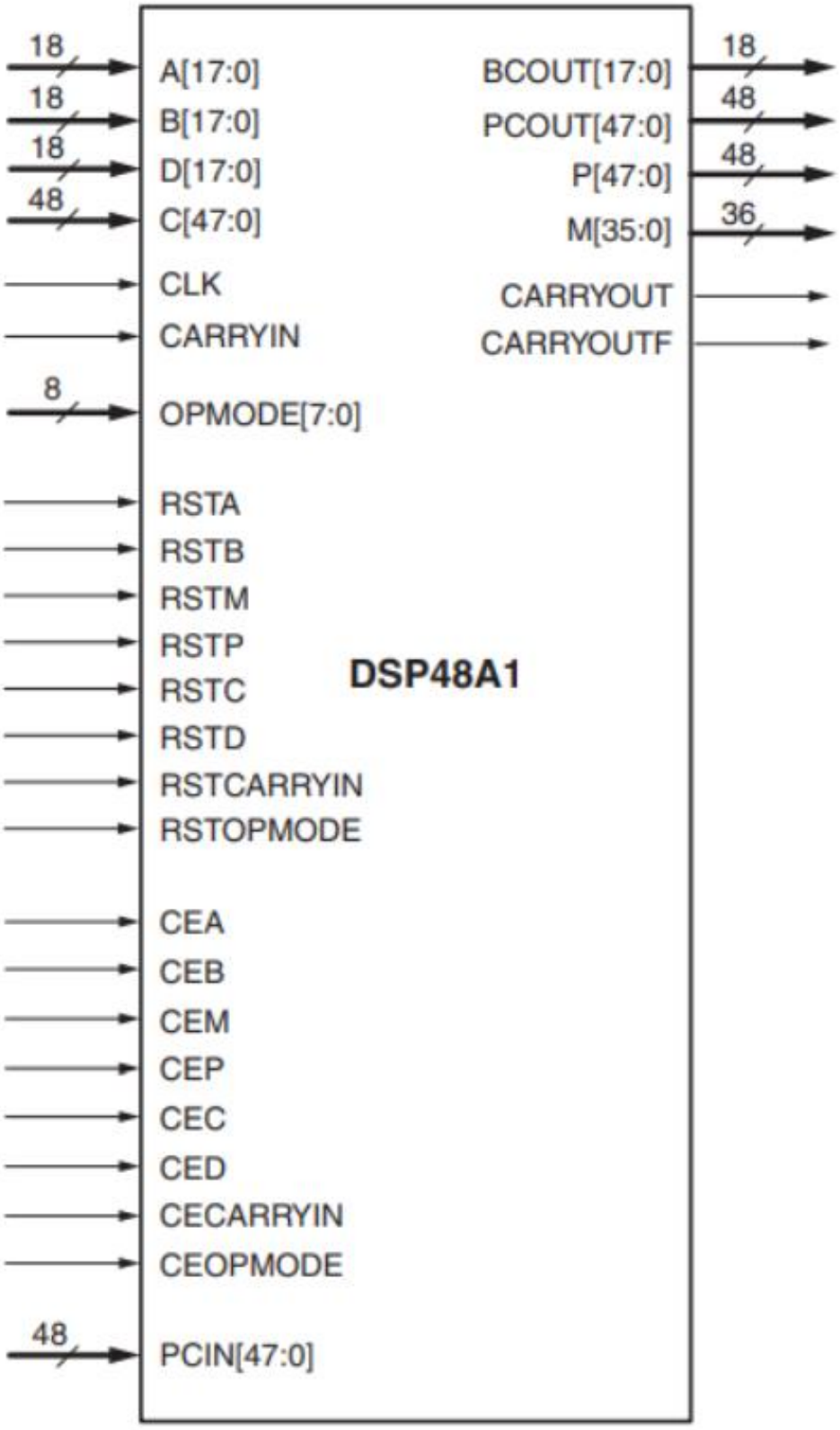
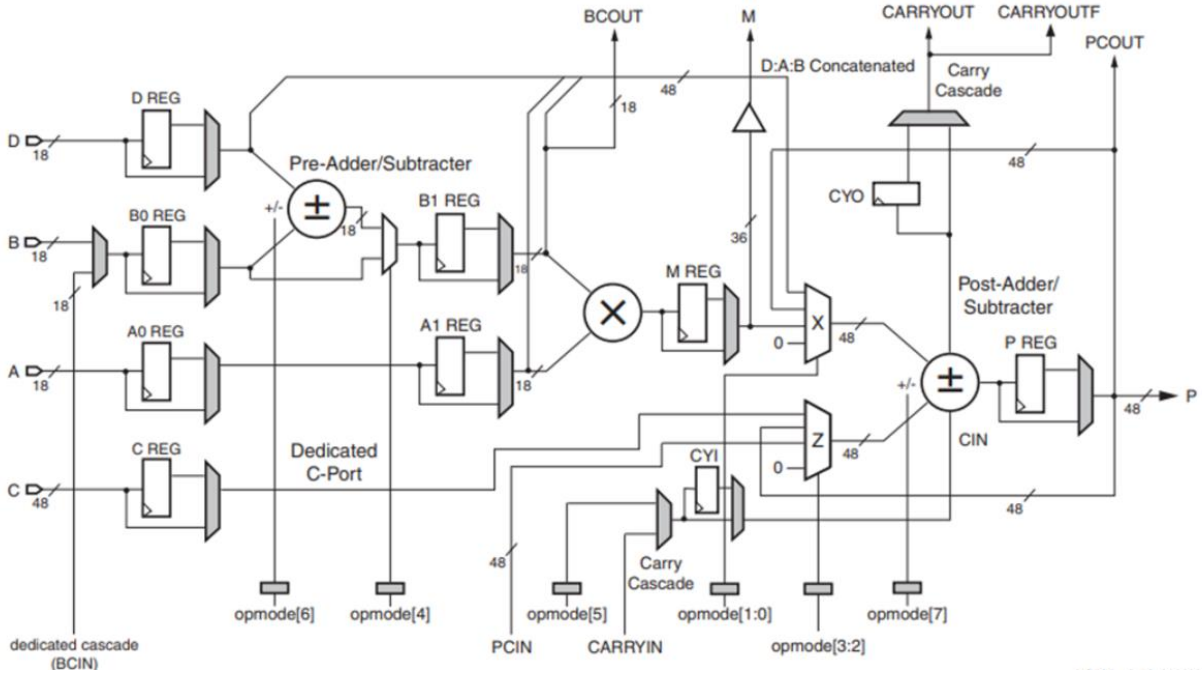
NAME:SHERIF AHMED ABDELFATAH SAYED

Project 1 (DSP48A1)

The DSP48A1 is a dedicated Digital Signal Processing (DSP) block in the Xilinx Spartan-6 FPGA family. It is a high-performance, low-power building block used for implementing arithmetic-intensive operations like multiply-accumulate (MAC), multiply-add (MAD), and more complex digital filters or signal processing pipelines.




18
A[17:0]
18
B[17:0]
18
D[17:0]
48
C[47:0]
CLK
CARRYIN
8
OPMODE[7:0]
RSTA
RSTB
RSTM
RSTP
RSTC
RSTD
RSTCARRYIN
RSTOPMODE
CEA
CEB
CEM
CEP
CEC
CED
CECARRYIN
CEOPMODE
48
PCIN[47:0]
DSP48A1
BCOUT[17:0]
18
PCOUT[47:0]
48
P[47:0]
48
M[35:0]
36
CARRYOUT
CARRYOUTF



1 Code:

```
//rst type (sync or async) (defsult sunc)

module ff_forinst (clk,rst,ce,d,q);
parameter width =18 ;
parameter REG=1;   //lw reg y3dy 3la ff lw msh kda a2lb combinational
parameter RSTTYPE = "sync";
input clk,rst,ce;
//ce enable of clk
input [width-1:0] d;
output reg [width-1:0] q;

generate
    if(REG)begin
    if(RSTTYPE == "sync") begin
      always @(posedge clk ) begin

        if(rst) //Reset Input Ports: All the resets are active high reset.
        q<=0;
        else if(ce)
        q<=d;
      end
    end
    else if(RSTTYPE == "async")begin
```

```
         always @(posedge clk or posedge rst ) begin

      if(rst)
        q<=0;
        else if(ce)
        q<=d;
         end
    end
    end
    else if (!REG) begin
         always @(* ) begin
            q=d;
         end
    end

endgenerate
endmodule
```

main module:

```
module DSP48A1( A,B,D,BCIN, C,clk,CARRYIN, PCIN,OPMODE,rstA
,rstB,rstM,rstP,rstC,rstD,rstCARRYIN,rstSTOPMODE, CEA
,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE,BCOUT,P,PCOUT,M,CARRYOUT, CARRYOUTF);
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,rstSTOPMODE;
input CEA ,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE;
input clk ,CARRYIN;
output [17:0]BCOUT;
output [47:0]P,PCOUT;
output [35:0]M;
```

```
output CARRYOUT;
output CARRYOUTF;

wire [17:0] D_OUT,A0_OUT,A1_OUT,B0_OUT,B1_OUT;
reg  [17:0] B_IN;
wire [47:0] C_OUT,D_A_B_CONC;
reg [47:0] post_add_sub,X_OUT,Z_OUT;
reg [17:0] pre_add_sub;
reg [17:0] B1_IN;
reg [35:0] M_IN;
wire [35:0] M_OUT;
wire [7:0]OPMODE_OUT;
reg CYI_IN;
reg CYO_IN;
wire CIN;

always @(*) begin
    if(B_INPUT == "direct")
    B_IN=B;
    else  if(B_INPUT == "CASCADE")
      B_IN=BCIN;
      else
         B_IN=0;
end
 ff_forinst #(.width(18),.REG(A0REG),.RSTTYPE(RSTTYPE)) a0_reg(clk,rstA,CEA,A,A0_OUT);
//a
  ff_forinst #(.width(18),.REG(B0REG),.RSTTYPE(RSTTYPE)) b0_reg (clk,rstB,CEB,
B_IN,B0_OUT); //b
 ff_forinst #(.width(48),.REG(CREG),.RSTTYPE(RSTTYPE)) c0_reg(clk,rstC,CEC,C,C_OUT);//c
 ff_forinst #(.width(18),.REG(DREG),.RSTTYPE(RSTTYPE)) d0_reg(clk,rstD,CED,D,D_OUT);//d
  ff_forinst #(.width(8),.REG(OPMODEREG),.RSTTYPE(RSTTYPE))
op_mode(clk,rstSTOPMODE,CEOPMODE,OPMODE,OPMODE_OUT);//opmode



always @(*) begin
    case(OPMODE_OUT[6])

0:pre_add_sub=D_OUT+B0_OUT;
1:pre_add_sub=D_OUT-B0_OUT;
    endcase
end

always @(*) begin
    case(OPMODE_OUT[4])

0:B1_IN= B0_OUT;
1:B1_IN=pre_add_sub;
    endcase
```

```
end
ff_forinst #(.width(18),.REG(A1REG),.RSTTYPE(RSTTYPE))
a1_reg(clk,rstA,CEA,A0_OUT,A1_OUT);
  ff_forinst #(.width(18),.REG(B1REG),.RSTTYPE(RSTTYPE)) b1_reg (clk,rstB,CEB,
B1_IN,B1_OUT);
  assign BCOUT=B1_OUT;
always @(*) begin
    M_IN=A1_OUT *B1_OUT;
end
 ff_forinst #(.width(36),.REG(MREG),.RSTTYPE(RSTTYPE)) m_reg(clk,rstM,CEM, M_IN,M_OUT);
 assign M=M_OUT;

always@(*)begin
if(CARRYINSEL=="OPMODE5")
CYI_IN=OPMODE_OUT[5];
else if(CARRYINSEL=="CARRYIN")
CYI_IN=CARRYIN;
else
CYI_IN=0;
end
ff_forinst #(.width(1),.REG(CARRYINREG),.RSTTYPE(RSTTYPE))
cyi_in(clk,rstCARRYIN,CECARRYIN, CYI_IN,CIN);
assign D_A_B_CONC={D_OUT[11:0],A1_OUT[17:0],B1_OUT[17:0]};
//X
always @(*) begin
case(OPMODE_OUT[1:0])
2'b00:X_OUT=0;
2'b01:X_OUT=M_OUT;
2'b10:X_OUT=P;
2'b11:X_OUT=D_A_B_CONC;
endcase
end
//Z
always @(*) begin
case(OPMODE_OUT[3:2])
2'b00:Z_OUT=0;
2'b01:Z_OUT=PCIN ;
2'b10:Z_OUT=P;
2'b11:Z_OUT=C_OUT;

endcase
end

always @(*) begin
    case(OPMODE_OUT[7])
//mtnsash cout_post_add_sub
0:{ CYO_IN,post_add_sub}=X_OUT+Z_OUT+CIN;
1:{ CYO_IN,post_add_sub}=Z_OUT-(X_OUT+CIN);
    endcase
```

```
end

//CYO_IN = HWA AL CARRY OUT BTA3Y MN AL post_add_sub

ff_forinst #(.width(1),.REG(CARRYOUTREG),.RSTTYPE(RSTTYPE))
CYOO(clk,rstCARRYIN,CECARRYIN, CYO_IN, CARRYOUT);
assign  CARRYOUTF= CARRYOUT;

ff_forinst #(.width(48),.REG(PREG),.RSTTYPE(RSTTYPE)) FINAL(clk,rstP,CEP, post_add_sub,
P);
assign  PCOUT= P;


endmodule
```

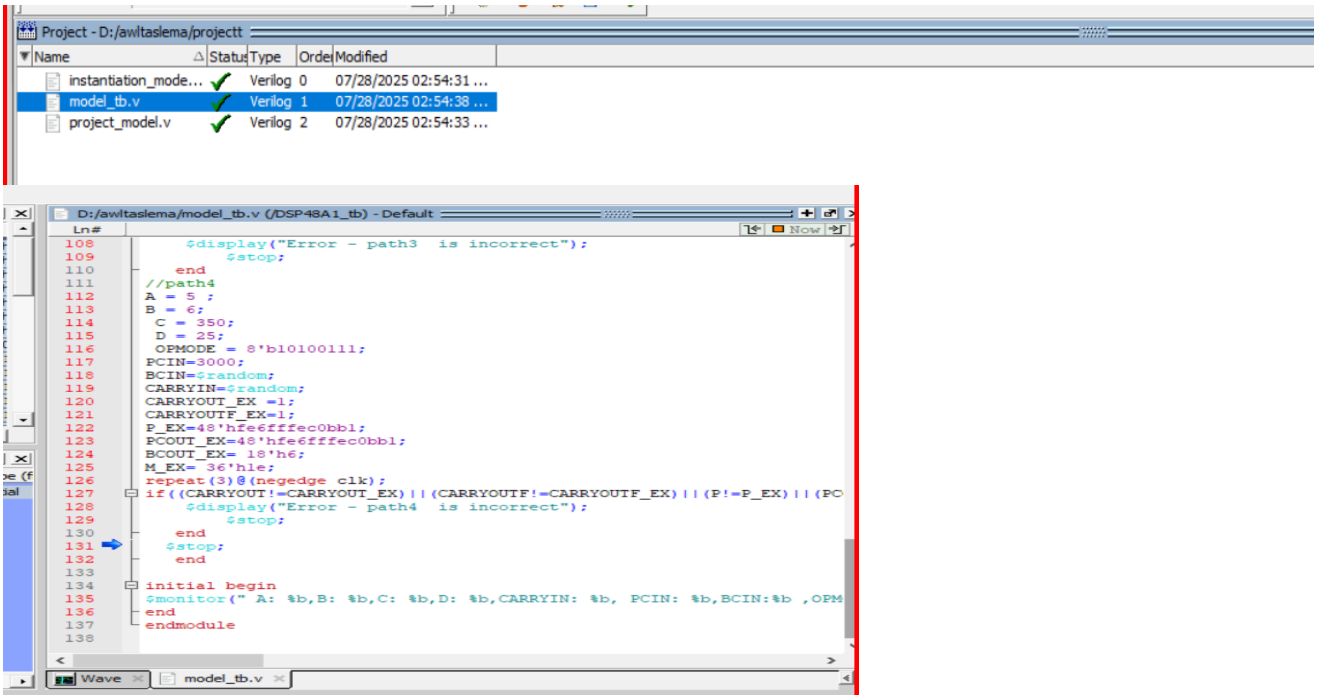
testbench

```
module DSP48A1_tb();

reg
clk,CARRYIN,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP,CEA,CEB,CEC,CECARRYIN,CE
D,CEM,CEOPMODE,CEP;
reg [17:0] A,B,D,BCIN;
reg [47:0] C,PCIN;
reg [7:0] OPMODE;
wire  CARRYOUT,CARRYOUTF;
wire   [47:0] P,PCOUT;
wire   [17:0] BCOUT;
wire   [35:0] M;
reg   CARRYOUT_EX,CARRYOUTF_EX;
reg    [47:0] P_EX,PCOUT_EX;
reg    [17:0] BCOUT_EX;
reg    [35:0] M_EX;
integer i;
//
parameter A0REG=0;
DSP48A1 m1 ( A,B,D,BCIN, C,clk,CARRYIN,
PCIN,OPMODE,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE, CEA
,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE,BCOUT,P,PCOUT,M,CARRYOUT, CARRYOUTF) ;

initial begin
    clk=0;
    forever
        #1 clk=~clk;
    end
        initial begin
    {RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP}=8'b11111111;
```

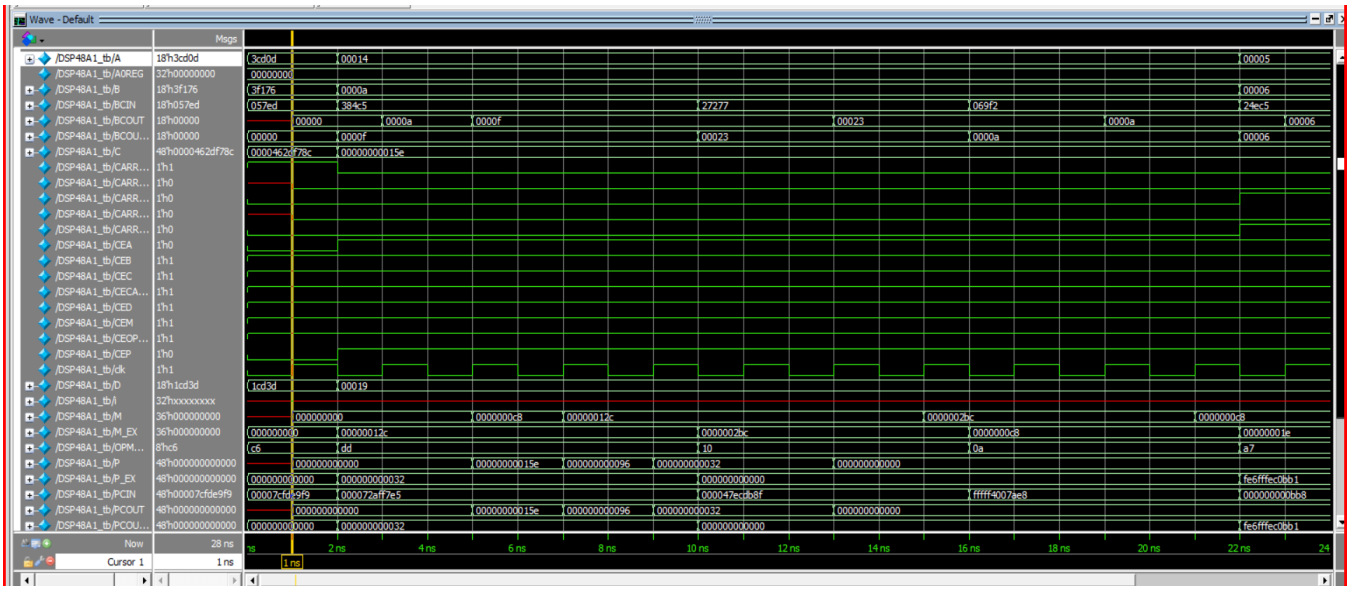
```
CEA=$random;
CEB=$random;
CEC=$random;
CECARRYIN=$random;
CED=$random;
CEM=$random;
CEOPMODE=$random;
CEP=$random;
CARRYIN=$random;
A=$random;
B=$random;
D=$random;
BCIN=$random;
C=$random;
PCIN=$random;
 OPMODE=$random;
  {CARRYOUT_EX,CARRYOUTF_EX,P_EX,PCOUT_EX,BCOUT_EX,M_EX}=0;
 @(negedge clk);
 if((CARRYOUT!=CARRYOUT_EX)||(CARRYOUTF!=CARRYOUTF_EX)||(P!=P_EX)||(PCOUT!=PCOUT_EX)||(
BCOUT!=BCOUT_EX)||(M!=M_EX))begin
    $display("Error - reset  is incorrect");
        $stop;
   end
 //path1
 {RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP}=0;
{CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP}=8'b11111111;
A = 20 ;
B = 10;
 C = 350;
 D = 25;
  OPMODE = 8'b11011101;
  BCIN=$random;
CARRYIN=$random;
PCIN=$random;
CARRYOUT_EX =0;
CARRYOUTF_EX=0;
P_EX=48'h32;
PCOUT_EX=48'h32;
BCOUT_EX= 18'hf;
M_EX= 36'h12c;
repeat(4)@(negedge clk);
if((CARRYOUT!=CARRYOUT_EX)||(CARRYOUTF!=CARRYOUTF_EX)||(P!=P_EX)||(PCOUT!=PCOUT_EX)||(B
COUT!=BCOUT_EX)||(M!=M_EX))begin
    $display("Error - path1  is incorrect");
        $stop;
   end
   //path2
A = 20 ;
B = 10;
```

```
 C = 350;
 D = 25;
 OPMODE = 8'b00010000;
BCIN=$random;
CARRYIN=$random;
PCIN=$random;
CARRYOUT_EX =0;
CARRYOUTF_EX=0;
P_EX=48'h0;
PCOUT_EX=48'h0;
BCOUT_EX= 18'h23;
M_EX= 36'h2bc;
repeat(3)@(negedge clk);
if((CARRYOUT!=CARRYOUT_EX)||(CARRYOUTF!=CARRYOUTF_EX)||(P!=P_EX)||(PCOUT!=PCOUT_EX)||(B
COUT!=BCOUT_EX)||(M!=M_EX))begin
    $display("Error - path2  is incorrect");
        $stop;
   end
   //path3
A = 20 ;
B = 10;
 C = 350;
 D = 25;
 OPMODE = 8'b00001010;
BCIN=$random;
CARRYIN=$random;
PCIN=$random;
CARRYOUT_EX =0;
CARRYOUTF_EX=0;
P_EX=48'h0;
PCOUT_EX=48'h0;
BCOUT_EX= 18'ha;
M_EX= 36'hc8;
repeat(3)@(negedge clk);
if((CARRYOUT!=CARRYOUT_EX)||(CARRYOUTF!=CARRYOUTF_EX)||(P!=P_EX)||(PCOUT!=PCOUT_EX)||(B
COUT!=BCOUT_EX)||(M!=M_EX))begin
    $display("Error - path3  is incorrect");
        $stop;
   end
//path4
A = 5 ;
B = 6;
 C = 350;
 D = 25;
 OPMODE = 8'b10100111;
PCIN=3000;
BCIN=$random;
CARRYIN=$random;
CARRYOUT_EX =1;
```

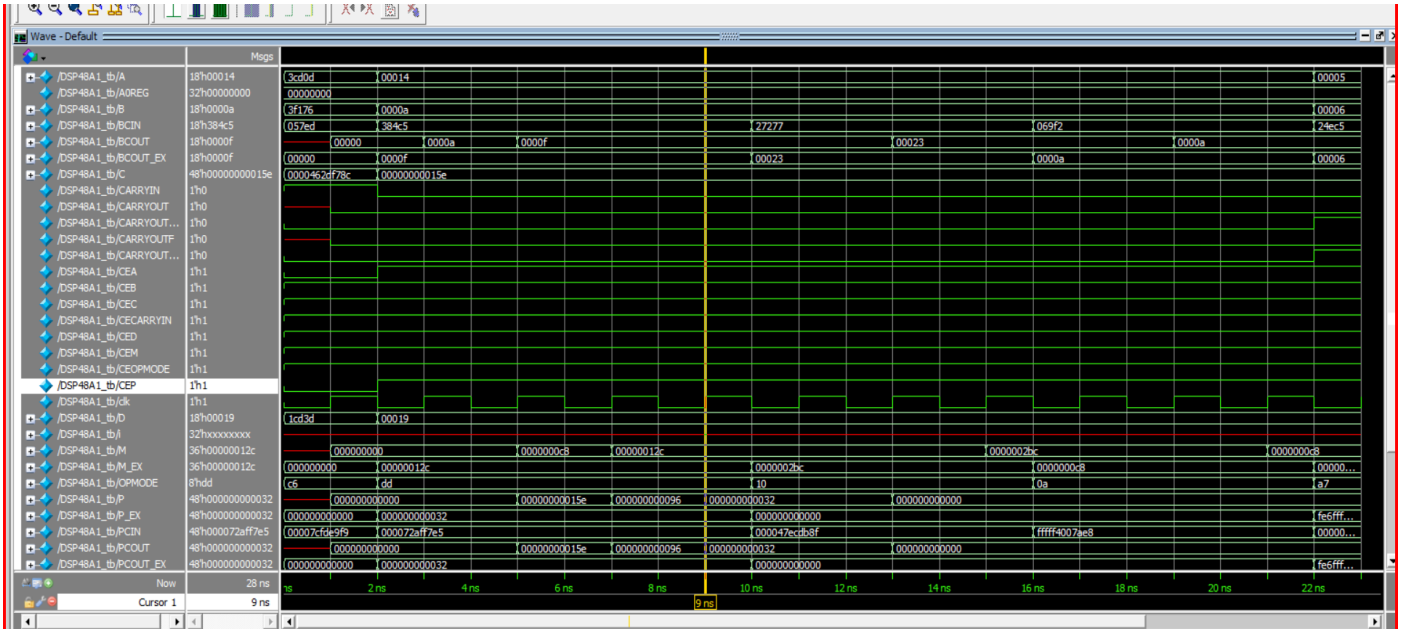
```
CARRYOUTF_EX=1;
P_EX=48'hfe6fffec0bb1;
PCOUT_EX=48'hfe6fffec0bb1;
BCOUT_EX= 18'h6;
M_EX= 36'h1e;
repeat(3)@(negedge clk);
if((CARRYOUT!=CARRYOUT_EX)|||(CARRYOUTF!=CARRYOUTF_EX)|||(P!=P_EX)|||(PCOUT!=PCOUT_EX)|||(B
COUT!=BCOUT_EX)|||(M!=M_EX))begin
    $display("Error - path4  is incorrect");
        $stop;
   end
  $stop;
   end

initial begin
$monitor(" A: %b,B: %b,C: %b,D: %b,CARRYIN: %b, PCIN: %b,BCIN:%b ,OPMODE:
%b,{CARRYOUT,CARRYOUTF,P,PCOUT,BCOUT,M}=%b,{CARRYOUT_EX,CARRYOUTF_EX,P_EX,PCOUT_EX,BCOU
T_EX,M_EX}=%b",A,B,C,D,CARRYIN,PCIN,BCIN,OPMODE,{CARRYOUT,CARRYOUTF,P,PCOUT,BCOUT,M},{C
ARRYOUT_EX,CARRYOUTF_EX,P_EX,PCOUT_EX,BCOUT_EX,M_EX});
end
endmodule
```



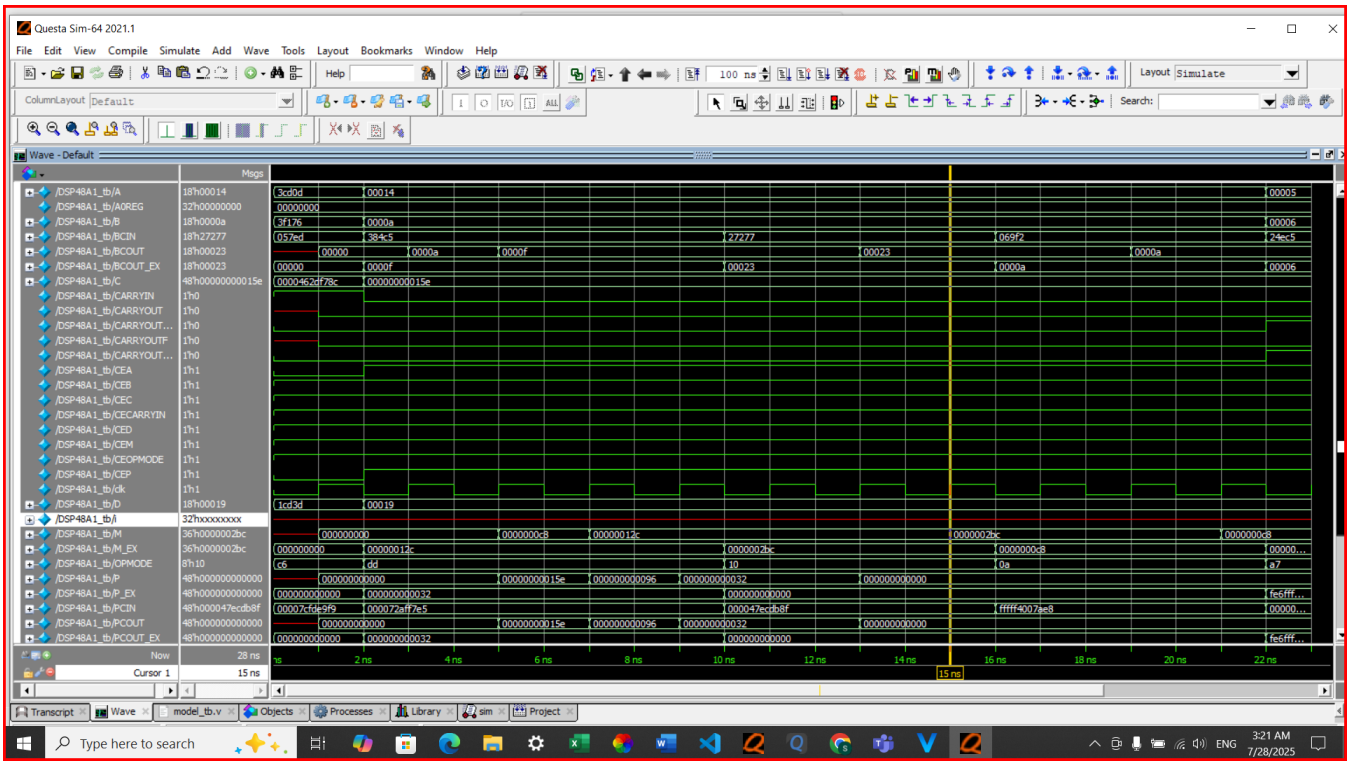
First verify reset



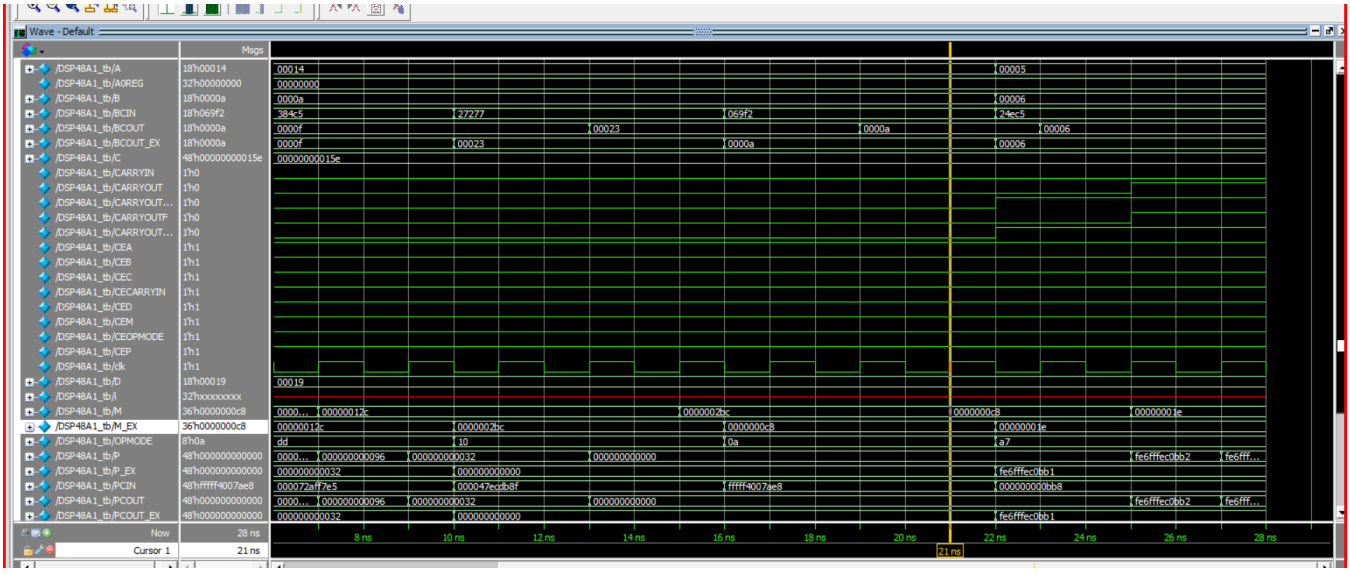
Verify Path1



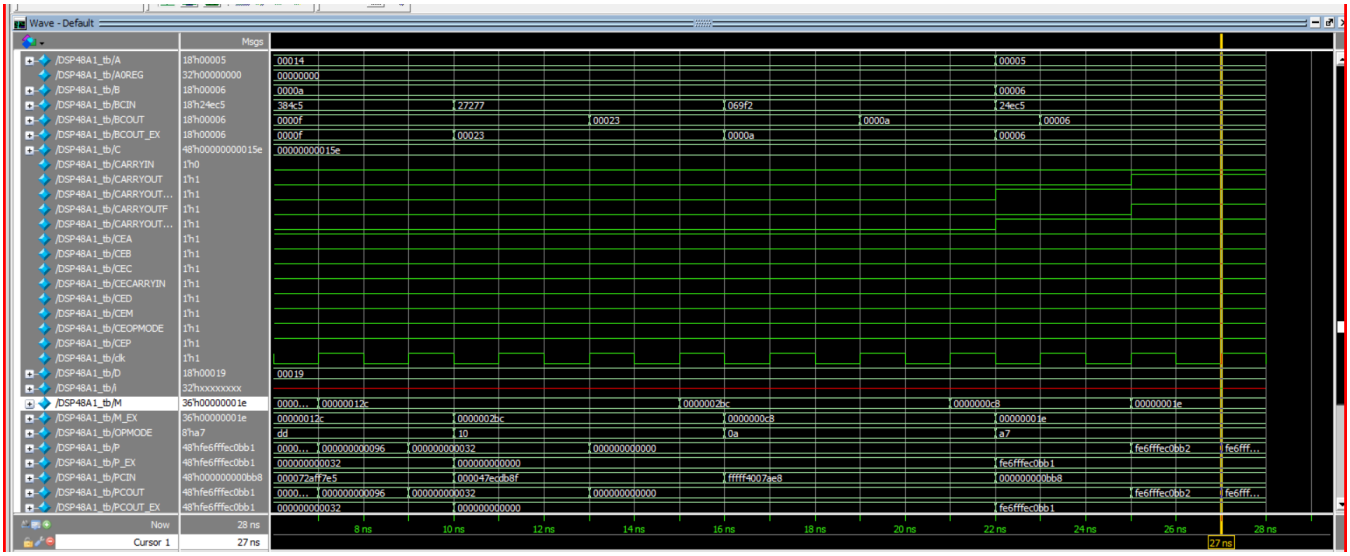
Verify path 2



Verify path 3

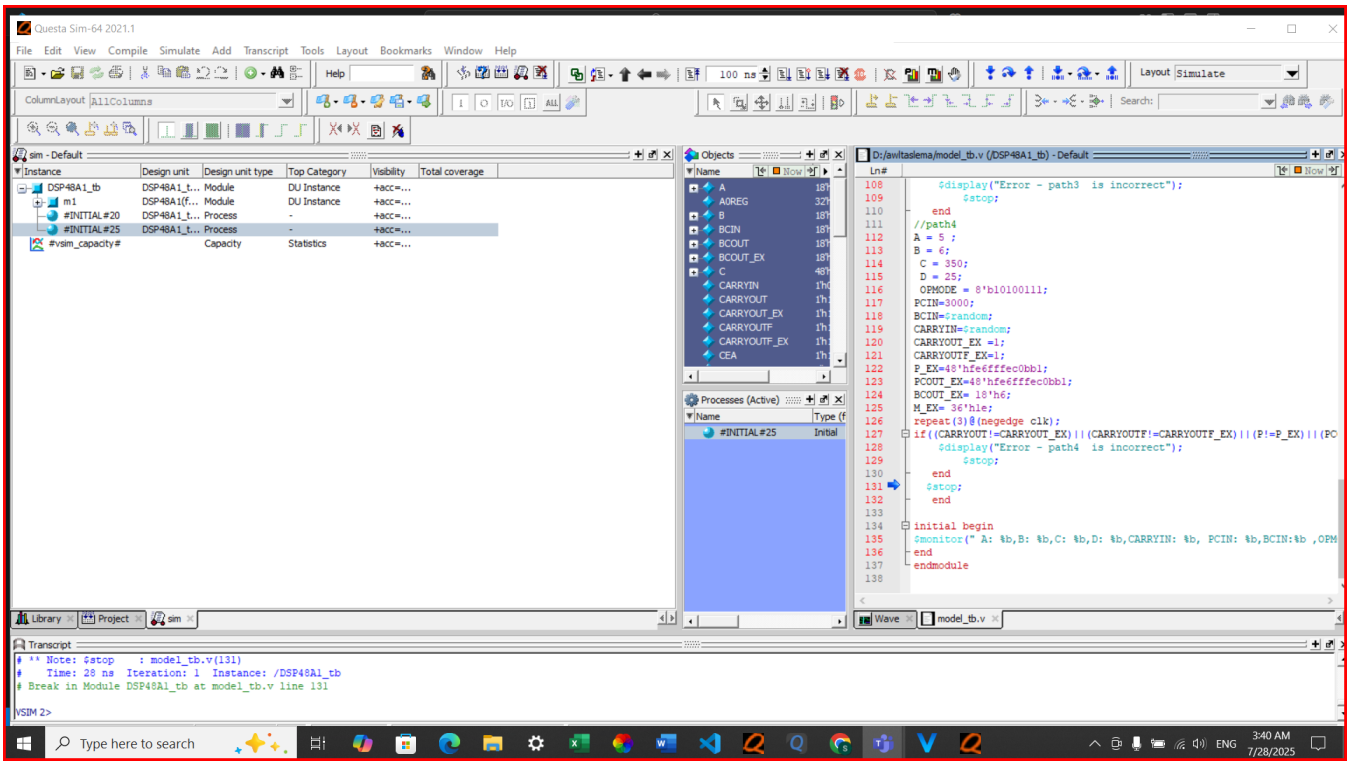


Verify path 4

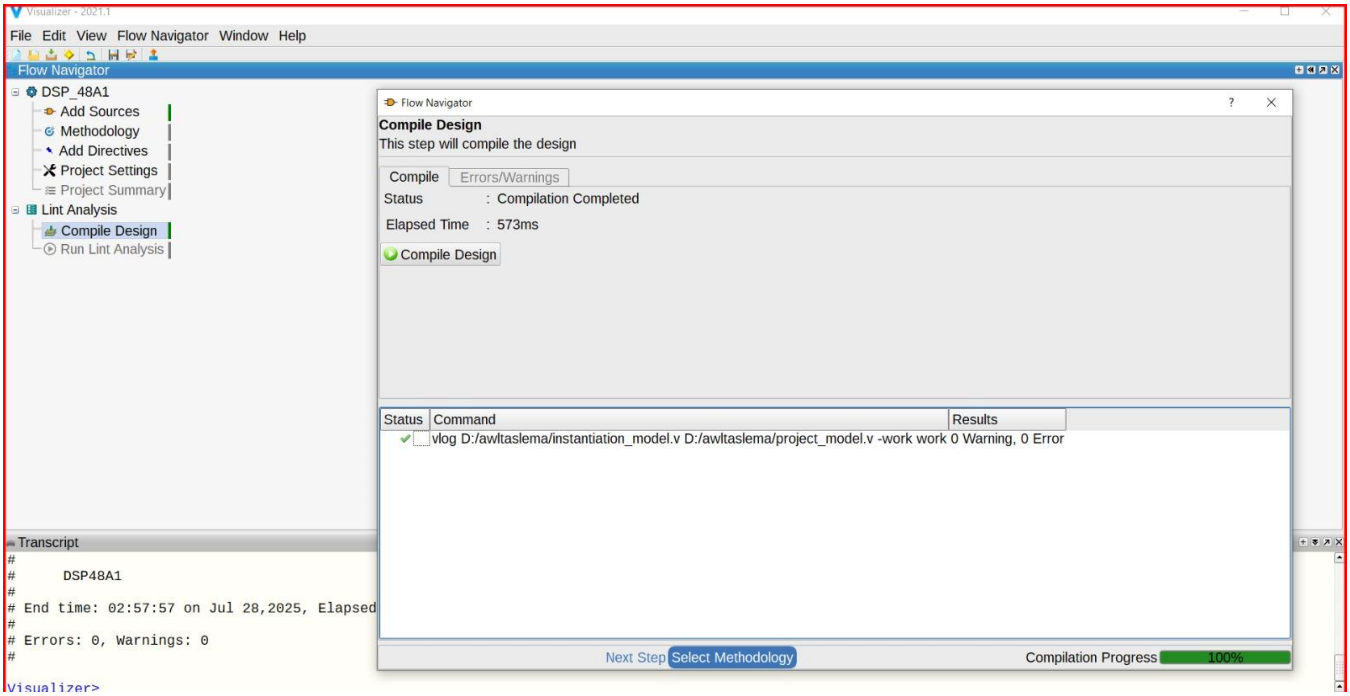


Dofile :

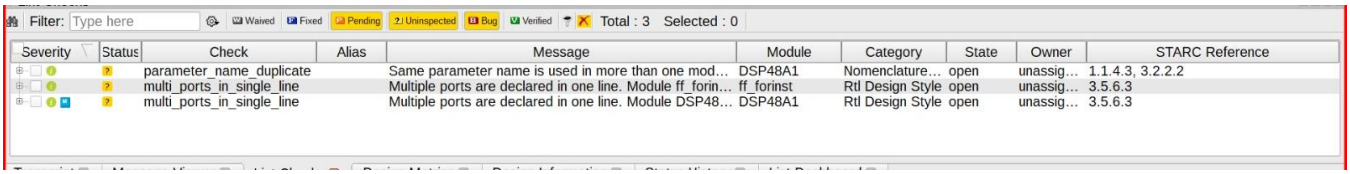
```
vlib work
vlog  instantiation_model.v project_model.v model_tb.v
vsim -voptargs=+acc work.DSP48A1_tb
add wave *
run -all
#quit -sim
```

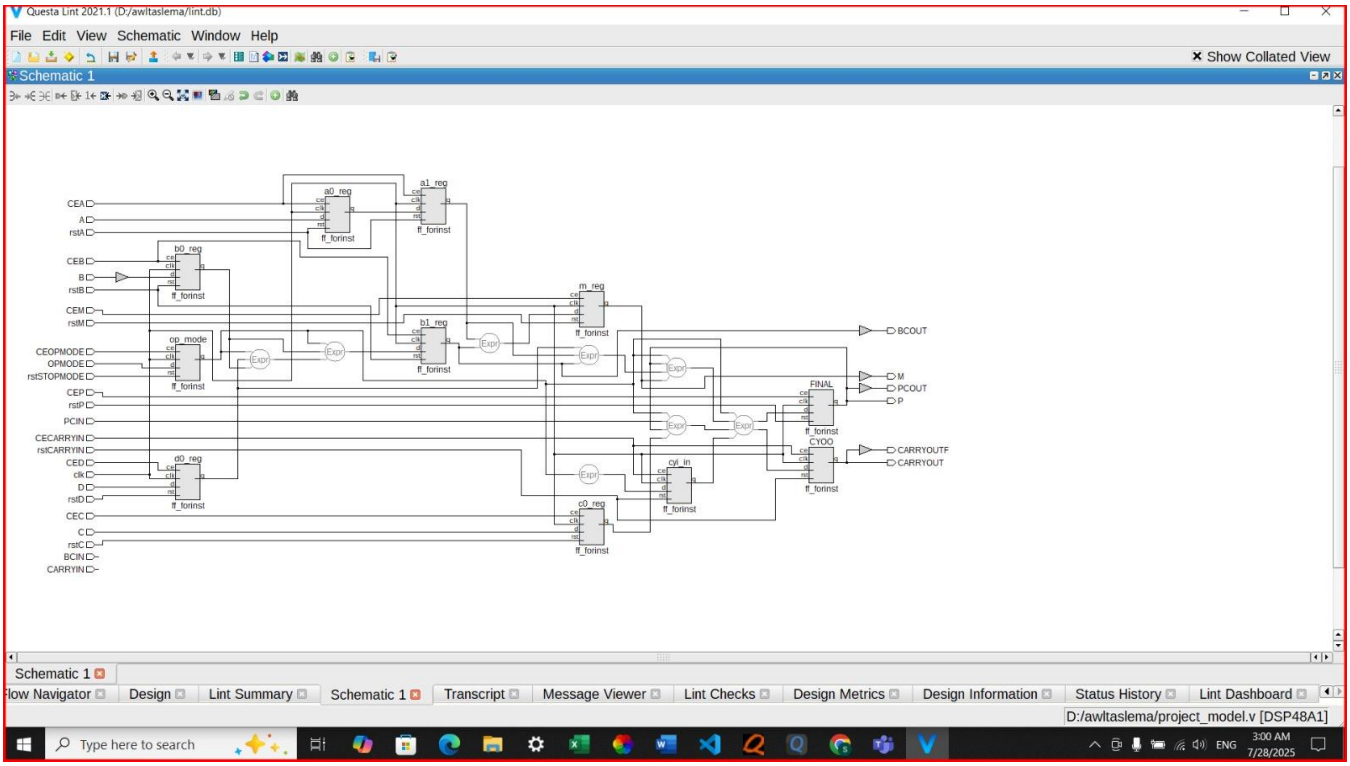

Questa Sim-64 2021.1
sim - Default
Objects
Processes (Active)
D:/awltaslema/model_tb.v (/DSP48A1_tb) - Default
Transcript
** Note: $stop : model_tb.v(131)
Time: 28 ns Iteration: 1 Instance: /DSP48A1_tb
Break in Module DSP48A1_tb at model_tb.v line 131
VSIM 2>
Type here to search
3:40 AM
7/28/2025

Questa_lint



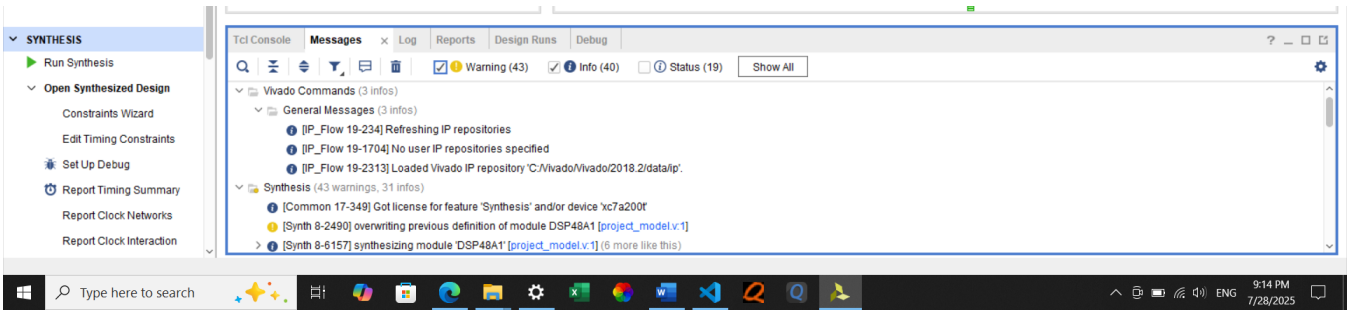
The info because module DSP48A1(A,B,D,BCIN,…….
So multiports….
Second info because defining paramter reset in insstantiation module and top module which also not error but just warning
Zero error

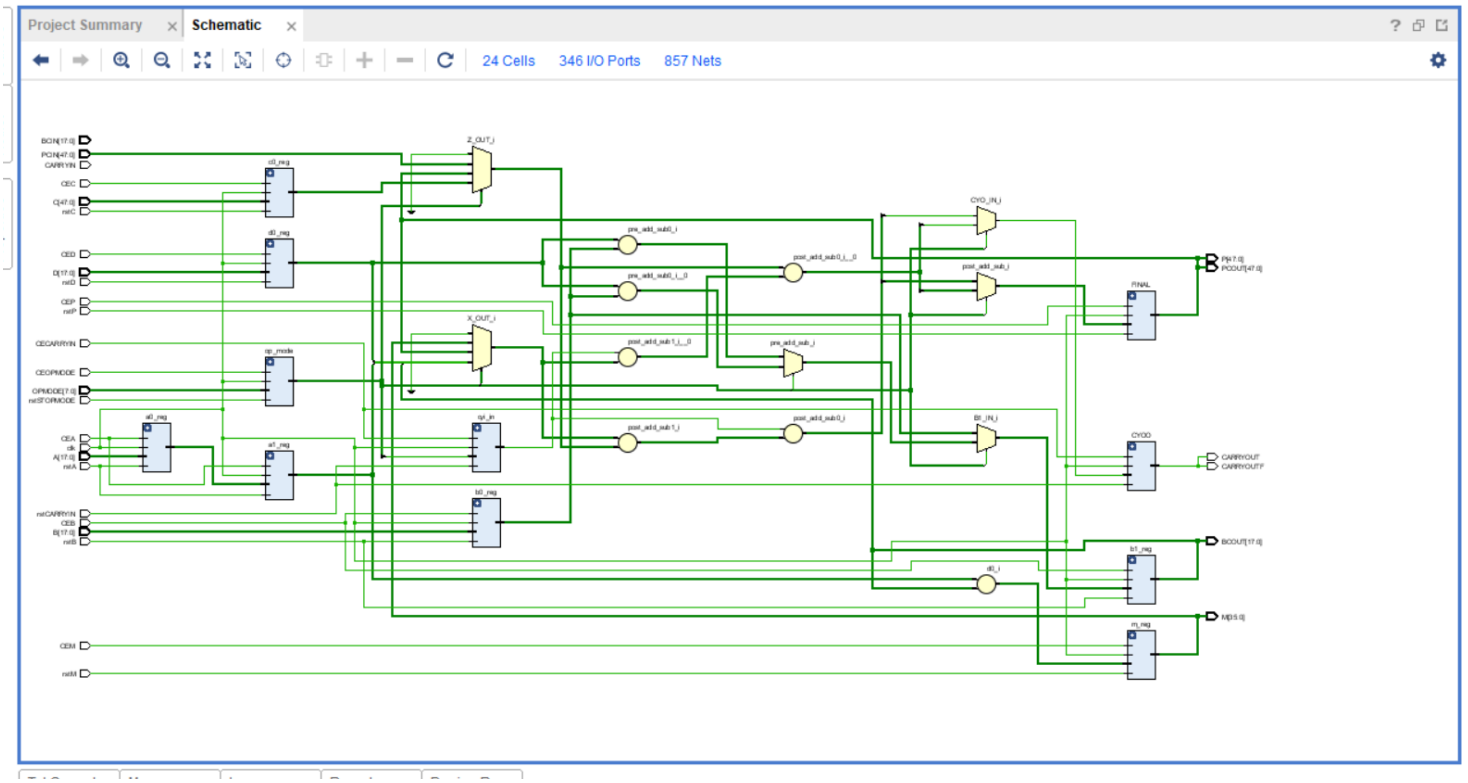




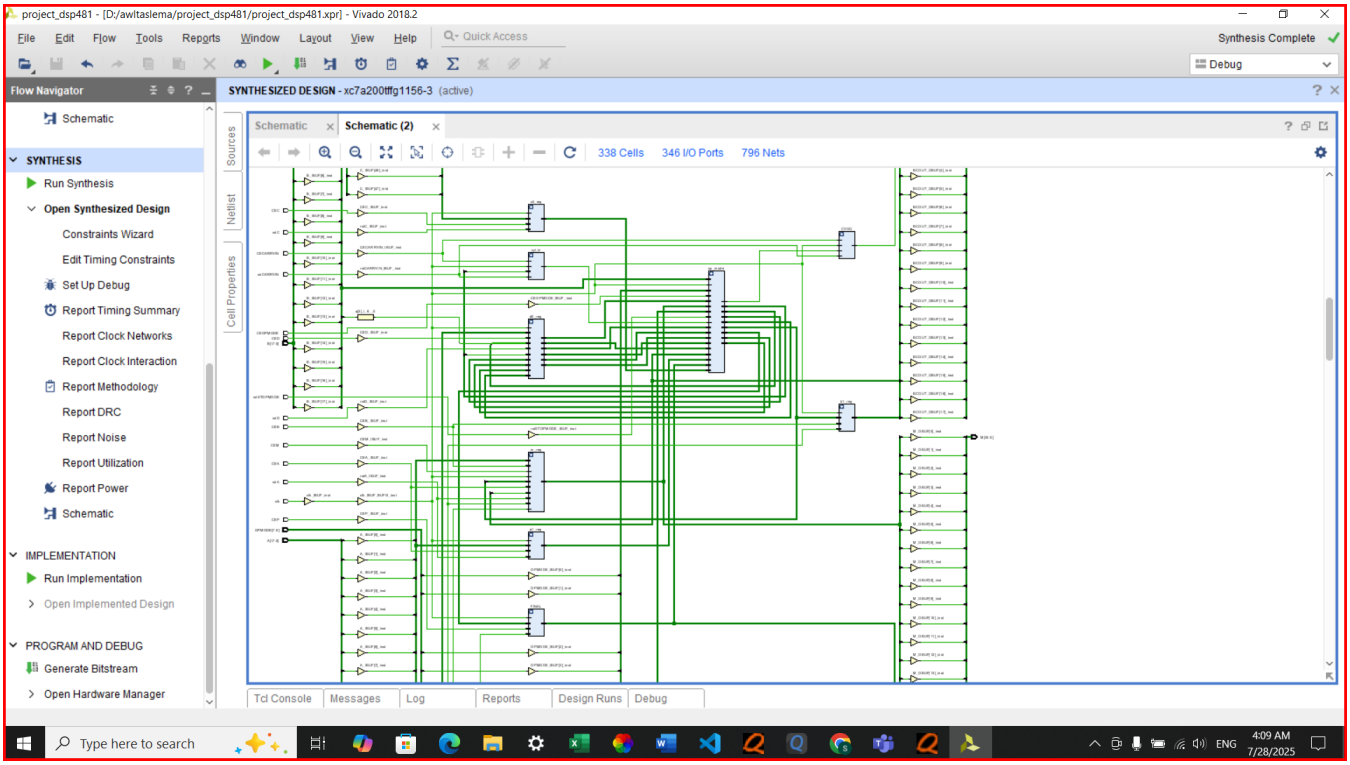
Vivado

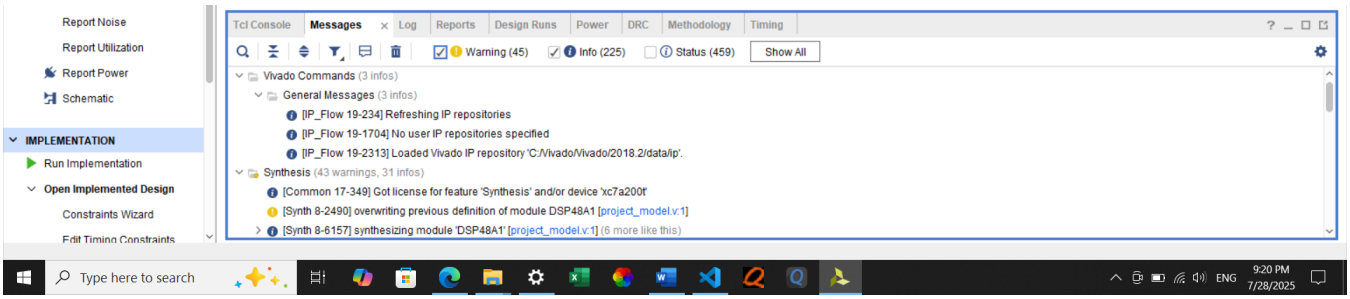
No critical warning



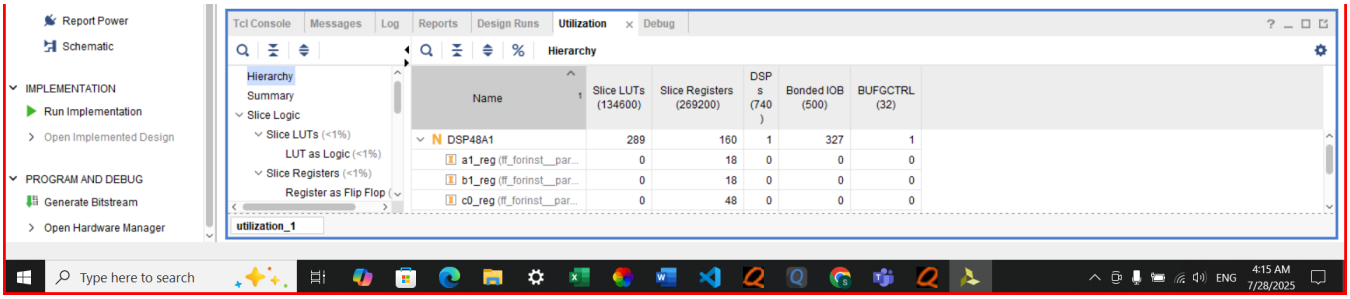


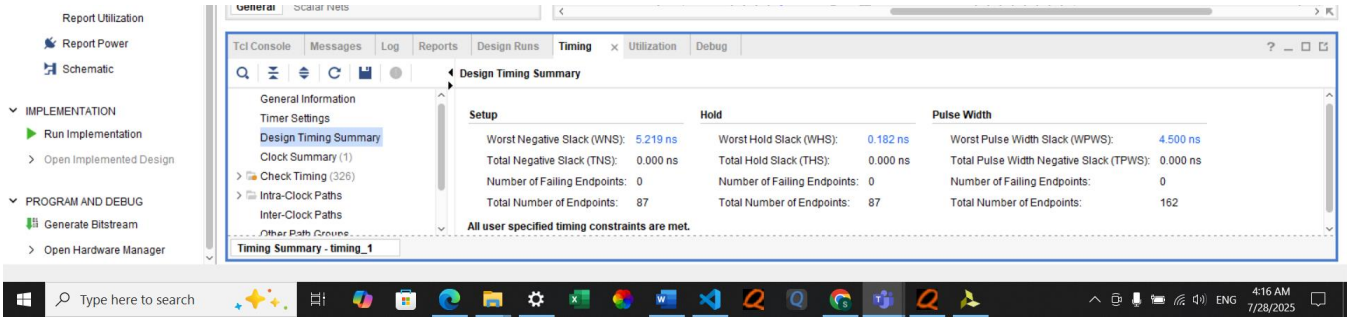
After synthesis : fy buffers kter bsbb an fy outputs take 48 bits f shaklo kber kda



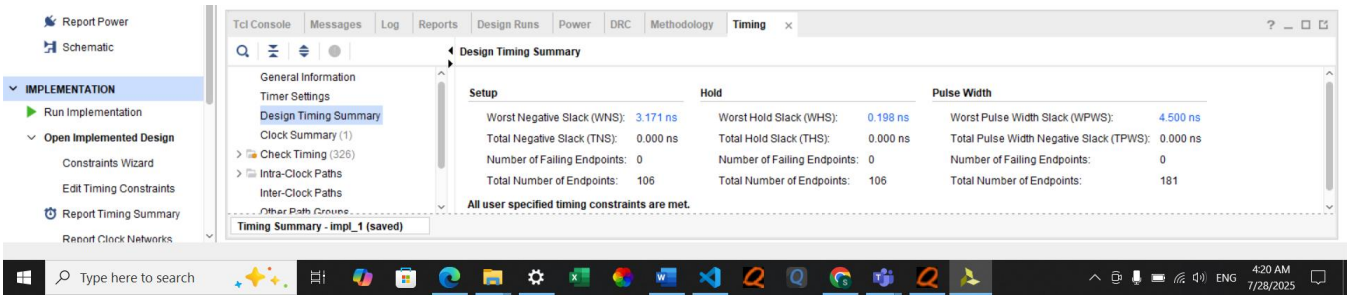


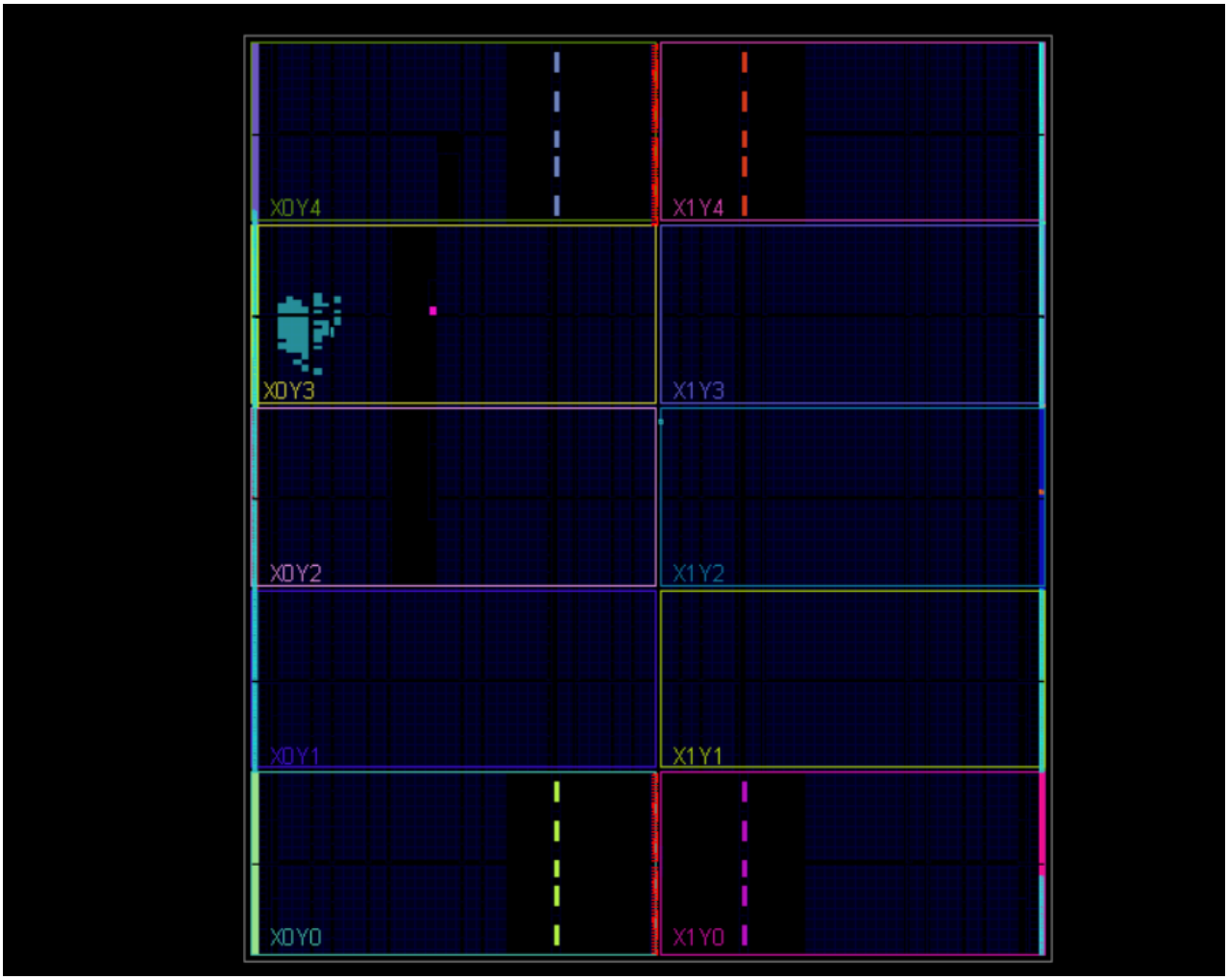
Report utilization





NOW: RUN_IMPLEMENTATION





Constrain 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 -add -name sys_clk_pin -period 10.00 -waveform {0 5} [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]

#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]
```