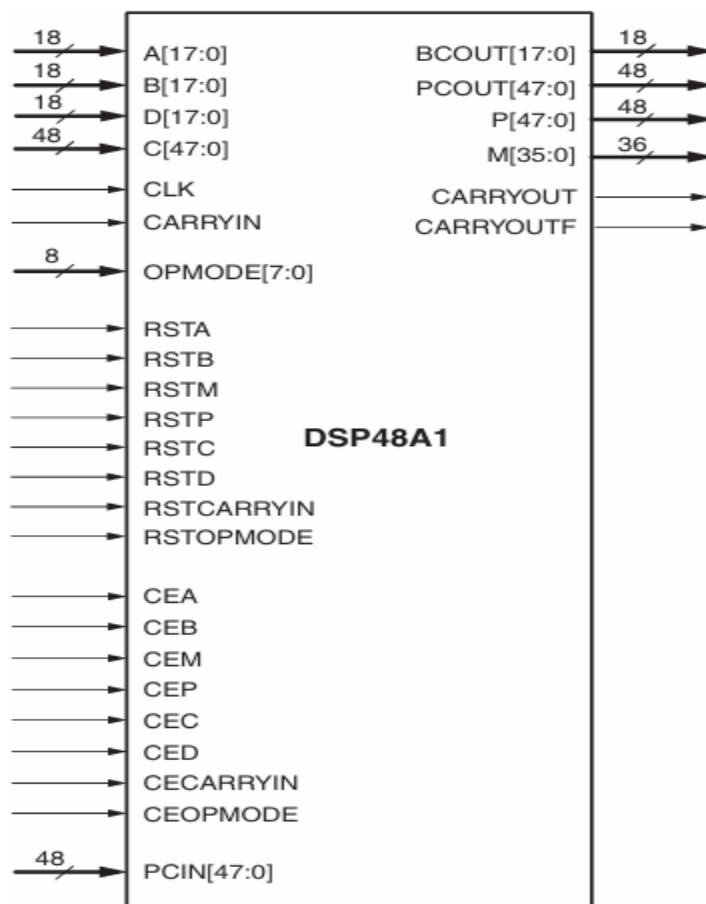
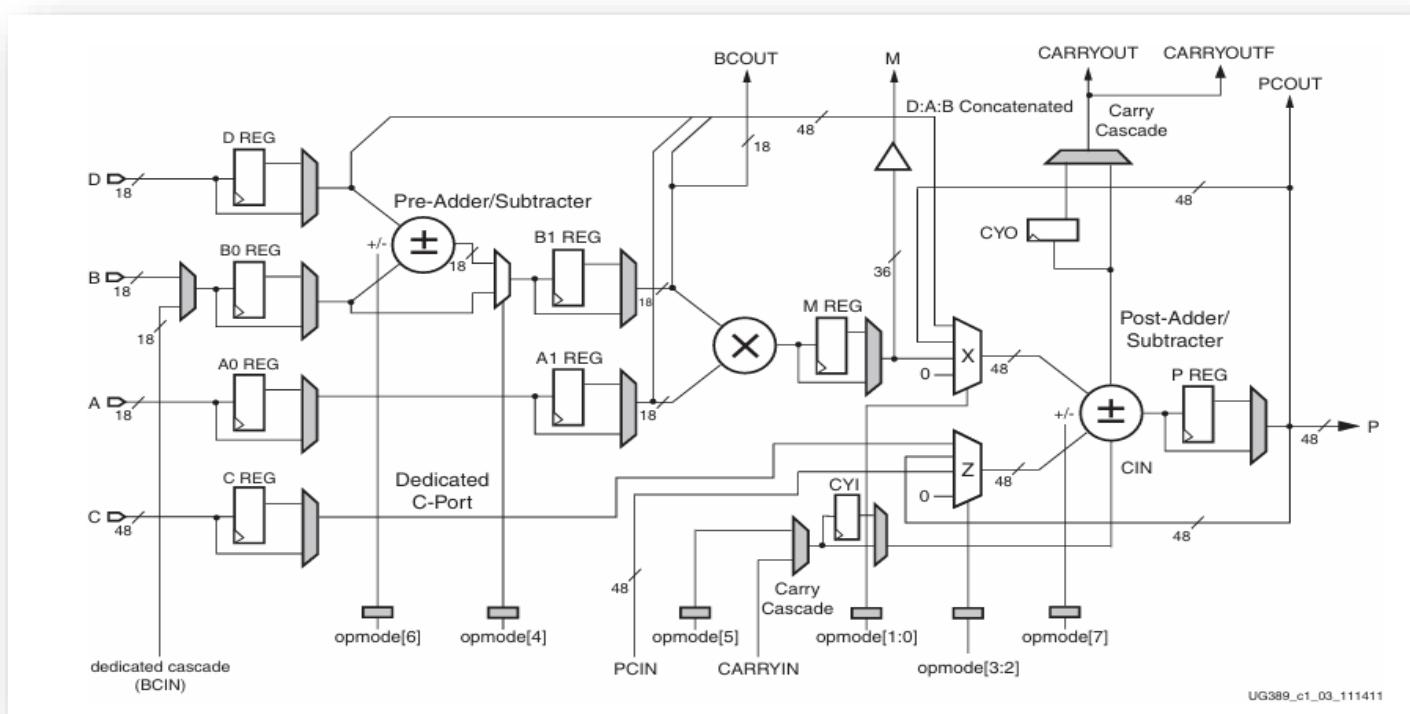


Spartan6 - DSP48A1



CODE

```

DSP48A1.v > DSP48A1
1 module DSP48A1 (A,B,C,D,BCIN,CARRYIN,M,P,CARRYOUT,CARRYOUTF,CLK,OPMODE,CEA,CEB,CEC,CECARRYIN,CED,CEM
2 ,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP,BCOUT,PCIN,PCOUT);
3 parameter A0REG=0; parameter A1REG=1; parameter B0REG=0; parameter B1REG=1; parameter CREG=1;
4 parameter DREG=1; parameter MREG=1; parameter PREG=1; parameter CAARYINREG=1; parameter CAARYOUTREG=1;
5 parameter OPMODEREG=1; parameter CARRYINSEL="OPMODE5"; parameter B_INPUT = "DIRECT"; parameter RSTTYPE = "SYNC" ;
6 input [17:0] A,B,D,BCIN;
7 input [47:0] C,PCIN;
8 input [7:0] OPMODE;
9 input CARRYIN,CLK,CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP;
10 input RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP;
11 output [17:0] BCOUT ;
12 output [47:0] P,PCOUT;
13 output [35:0] M;
14 output CARRYOUT,CARRYOUTF;
15 wire [17:0] MUXED_B,B0_S,A0_S,D_S,A1_S,PRE_OUT,B1_INPUT,B1_S;
16 wire [47:0] C_S,MUX_X_IN3,MUX_X_OUT,MUX_Z_OUT,POST_OUT,M_TO_MUX;
17 wire [7:0] OPMODE_S;
18 wire [35:0] MULT_OUT,M_S;
19 wire CARRYIN_S1,CIN,POST_COUT;
20 assign BCOUT=B1_S;
21 assign M=M_S;
22 assign PCOUT=P;
23 assign CARRYOUTF=CARRYOUT;
24 assign MUX_X_IN3={D_S[11:0],A1_S,B1_S};
25 assign M_TO_MUX={12'h000,M_S};
26 assign MUXED_B=(B_INPUT=="DIRECT")?B:(B_INPUT=="CASCADE")?BCIN:0;
27 assign CARRYIN_S1=(CARRYINSEL=="OPMODE5")?OPMODE_S[5]:((CARRYINSEL=="CARRYIN")?CARRYIN:0);
28 assign B1_INPUT=(OPMODE_S[4])?PRE_OUT:B0_S;
29 assign MULT_OUT=A1_S*B1_S;
30 assign PRE_OUT=(OPMODE_S[6])?D_S-B0_S:B0_S+D_S;
31 assign {POST_COUT,POST_OUT}=(OPMODE_S[7])? MUX_Z_OUT-(MUX_X_OUT+CIN):MUX_X_OUT+MUX_Z_OUT+CIN;

32 MUX_REG_INPUT #( .WIDTH(8), .MODE(RSTTYPE) ) OPMODE_SEL(OPMODE,OPMODEREG,OPMODE_S,CLK,RSTOPMODE,CEOPMODE);
33 MUX_REG_INPUT #( .WIDTH(18), .MODE(RSTTYPE) ) B0_SEL(MUXED_B,B0REG,B0_S,CLK,RSTB,CEB);
34 MUX_REG_INPUT #( .WIDTH(18), .MODE(RSTTYPE) ) A0_SEL(A,A0REG,A0_S,CLK,RSTA,CEA);
35 MUX_REG_INPUT #( .WIDTH(18), .MODE(RSTTYPE) ) A1_SEL(A0_S,A1REG,A1_S,CLK,RSTA,CEA);
36 MUX_REG_INPUT #( .WIDTH(18), .MODE(RSTTYPE) ) D_SEL(D,DREG,D_S,CLK,RSTD,CED);
37 MUX_REG_INPUT #( .WIDTH(48), .MODE(RSTTYPE) ) C_SEL(C,CREG,C_S,CLK,RSTC,CEC);
38 MUX_REG_INPUT #( .WIDTH(18), .MODE(RSTTYPE) ) B1_SEL(B1_INPUT,B1REG,B1_S,CLK,RSTB,CEB);
39 MUX_REG_INPUT #( .WIDTH(36), .MODE(RSTTYPE) ) M_SEL(MULT_OUT,MREG,M_S,CLK,RSTM,CEM);
40 MUX4_1 #( .WIDTH(48) ) MUX_X(M_TO_MUX,PCOUT,MUX_X_IN3,OPMODE_S[1:0],MUX_X_OUT);
41 MUX4_1 #( .WIDTH(48) ) MUX_Z(PCIN,PCOUT,C_S,OPMODE_S[3:2],MUX_Z_OUT);
42 MUX_REG_INPUT #( .MODE(RSTTYPE) ) CARRYIN_SEL(CARRYIN_S1,CAARYINREG,CIN,CLK,RSTCARRYIN,CECARRYIN);
43 MUX_REG_INPUT #( .WIDTH(48), .MODE(RSTTYPE) ) P_SEL(POST_OUT,PREG,P,CLK,RSTP,CEP);
44 MUX_REG_INPUT #( .MODE(RSTTYPE) ) CARRYOUT_SEL(POST_COUT,CAARYOUTREG,CARRYOUT,CLK,RSTCARRYIN,CECARRYIN);
45 endmodule //DSP48A1

```

REG_MUX

```
@ MUX_REG_INPUT.v > ...
1  module MUX_REG_INPUT (in,sel,out,CLK,rst,enable);
2  parameter WIDTH = 1 ;
3  parameter MODE="SYNC";
4  input [WIDTH-1:0]in;
5  input sel,rst,CLK,enable;
6  output [WIDTH-1:0] out ;
7  generate
8    if (MODE=="SYNC") sync_model #( .WIDTH(WIDTH)) SS(in,sel,out,CLK,rst,enable);
9    else Async_model #( .WIDTH(WIDTH)) AA(in,sel,out,CLK,rst,enable);
10   endgenerate
11  endmodule //MUX_REG_INPUT
12  module sync_model(in,sel,out,CLK,rst,enable);
13  parameter WIDTH = 1 ;
14  input [WIDTH-1:0]in;
15  input sel,rst,CLK,enable;
16  output [WIDTH-1:0] out ;
17  reg [WIDTH-1:0] out_reg ;
18  assign out =(sel)?out_reg:in;
19  always @ (posedge CLK) begin
20    if(rst) out_reg<=0;
21    else if (enable) begin
22      out_reg<=in;
23    end
24  end
25  endmodule
26  module Async_model(in,sel,out,CLK,rst,enable);
27  parameter WIDTH = 1 ;
28  input [WIDTH-1:0]in;
29  input sel,rst,CLK,enable;
30  output [WIDTH-1:0] out ;
31  reg [WIDTH-1:0] out_reg ;
32  assign out =(sel)?out_reg:in;
33  always @(posedge CLK or posedge rst) begin
34    if(rst) out_reg<=0;
35    else if (enable) begin
36      out_reg<=in;
37    end
38  end
39  endmodule
```

4_TO_1_MUX

```
@ MUX4_1.v > ...
1  module MUX4_1 (in0,in1,in2,sel,out);
2  parameter WIDTH=48;
3  input [WIDTH-1:0] in0,in1,in2 ;
4  input [1:0]sel;
5  output reg [WIDTH-1:0] out ;
6  always @(*) begin
7    case (sel)
8      2'b00:out=0;
9      2'b01:out=in0;
10     2'b10:out=in1;
11     2'b11:out=in2;
12     endcase
13  end
14  endmodule //MUX4_1
```

do_file

```
≡ DSP48A1.do
1 vlib work
2 vlog DSP48A1.v DSP48A1_Tb.v
3 vsim -voptargs=+acc work.DSP48A1_Tb
4 add wave *
5 run -all
```

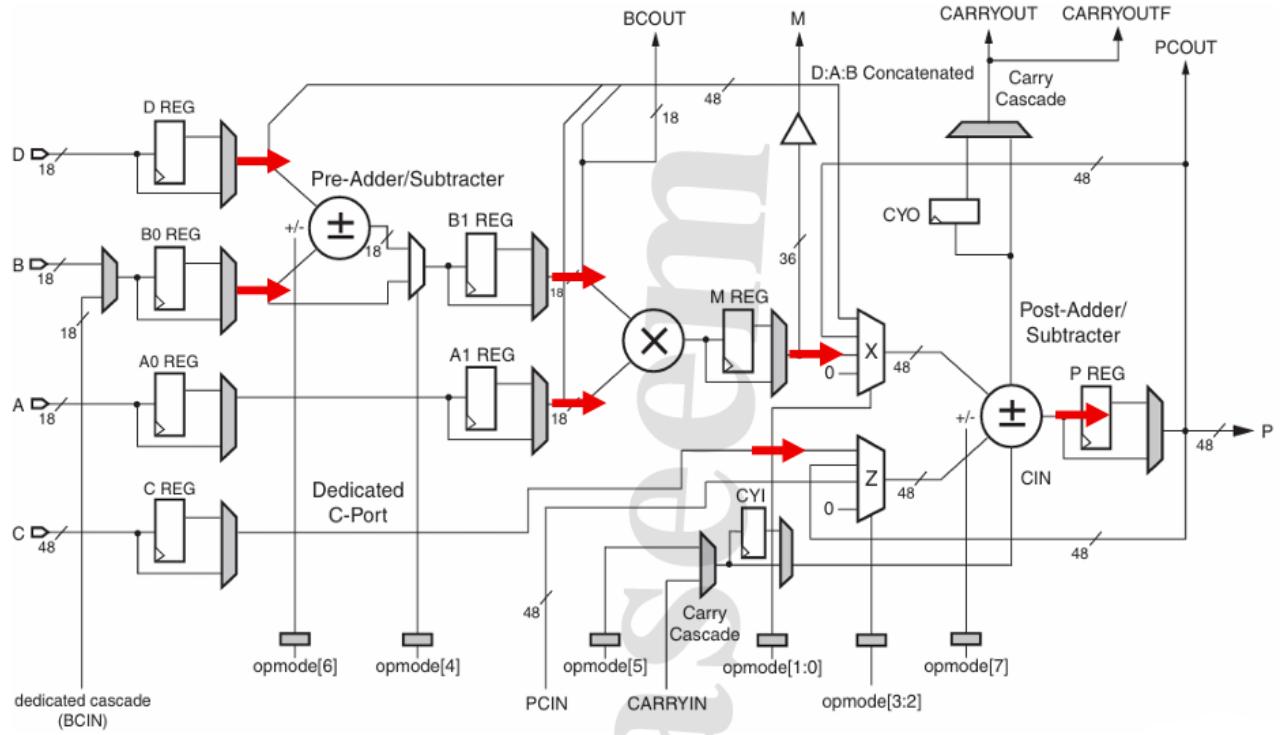
RST_tb

```
⌚ DSP48A1_Tb.v > ...
1 module DSP48A1_Tb ();
2 reg CLK,CARRYIN,CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP;
3 reg RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP;
4 reg [17:0]A,B,D,BCIN;
5 reg [47:0]C,PCIN;
6 reg [7:0]OPMODE;
7 wire CARRYOUT,CARRYOUTF;
8 wire [47:0]P,PCOUT;
9 wire [35:0] M;
10 wire [17:0] BCOUT ;
11 DSP48A1 TEST(A,B,C,D,BCIN,CARRYIN,M,P,CARRYOUT,CARRYOUTF,CLK,OPMODE,CEA,CEB,CEC,CECARRYIN
12 ,CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP,BCOUT,PCIN,PCOUT);
13 initial begin
14     CLK=0;
15     forever begin
16         #1 CLK=~CLK;
17     end
18 end
19 initial begin
20     {RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP}=8'b11111111;
21     A=$random; B=$random; C=$random; D=$random; BCIN=$random; CARRYIN=$random;
22     OPMODE=$random; CEA=$random; CEB=$random; CEC=$random; CECARRYIN=$random;
23     CED=$random; CEM=$random; CEOPMODE=$random; CEP=$random; PCIN=$random;
24     repeat(2)@(negedge CLK); //// to be clearly
25     if({P,M,CARRYOUT,CARRYOUTF,BCOUT,PCOUT}!=0)begin
26         $display("ERROR IN RST FUNCTION");
27         $stop;
28     end
```

RST_WAVE_PART

	Msgs						
/DSP48A1_Tb/P	48h000000000000	000000000000	000000000015e	000000000096	000000000032	000000000000	feffffe0bb2
/DSP48A1_Tb/PCOUT	48h000000000000	000000000000	000000000015e	000000000096	000000000032	000000000000	feffffe0bb2
/DSP48A1_Tb/CARRYOUT	1h0						
/DSP48A1_Tb/CARRYOUTF	1h0						
/DSP48A1_Tb/BCOUT	18h0000	00000	0000a	0000f		00023	0000a
/DSP48A1_Tb/M	36h0000000000	00000000		00000008	00000012c	0000002bc	00000008
/DSP48A1_Tb/OPMODE	8b01100101	01101101	11011101		00010000	000001010	10100111
/DSP48A1_Tb/CEA	1h0						
/DSP48A1_Tb/CEB	1h1						
/DSP48A1_Tb/CEC	1h1						
/DSP48A1_Tb/CED	1h1						
/DSP48A1_Tb/CEM	1h1						
/DSP48A1_Tb/CEP	1h1						
/DSP48A1_Tb/CCARRYIN	1h0						
/DSP48A1_Tb/CEOPMODE	1h0						
/DSP48A1_Tb/RSTA	1h1						
/DSP48A1_Tb/RSTB	1h1						
/DSP48A1_Tb/RSTC	1h1						
/DSP48A1_Tb/RSTCARR...	1h1						
/DSP48A1_Tb/RSTD	1h1						
/DSP48A1_Tb/RSTM	1h1						
/DSP48A1_Tb/RSTOPMODE	1h1						
/DSP48A1_Tb/RSTP	1h1						
/DSP48A1_Tb/A	18h13524	13524	00014				00005
/DSP48A1_Tb/B	18h15e81	15e81	0000a				00006
/DSP48A1_Tb/C	48hffff8484d609	ffff8484d609	000000000015e				
/DSP48A1_Tb/D	18h05663	05663	00019				
/DSP48A1_Tb/BCIN	18h17b0d	17b0d	3845		27277	069f2	24ec5
/DSP48A1_Tb/CCARRYIN	1h1						
/DSP48A1_Tb/PCIN	48hffffe33724e6	fffffe33724e6	000072aff7e5		000047ccb8f	fffff4007ae8	000000000b8
/DSP48A1_Tb/CLK	1h1						

PATH1



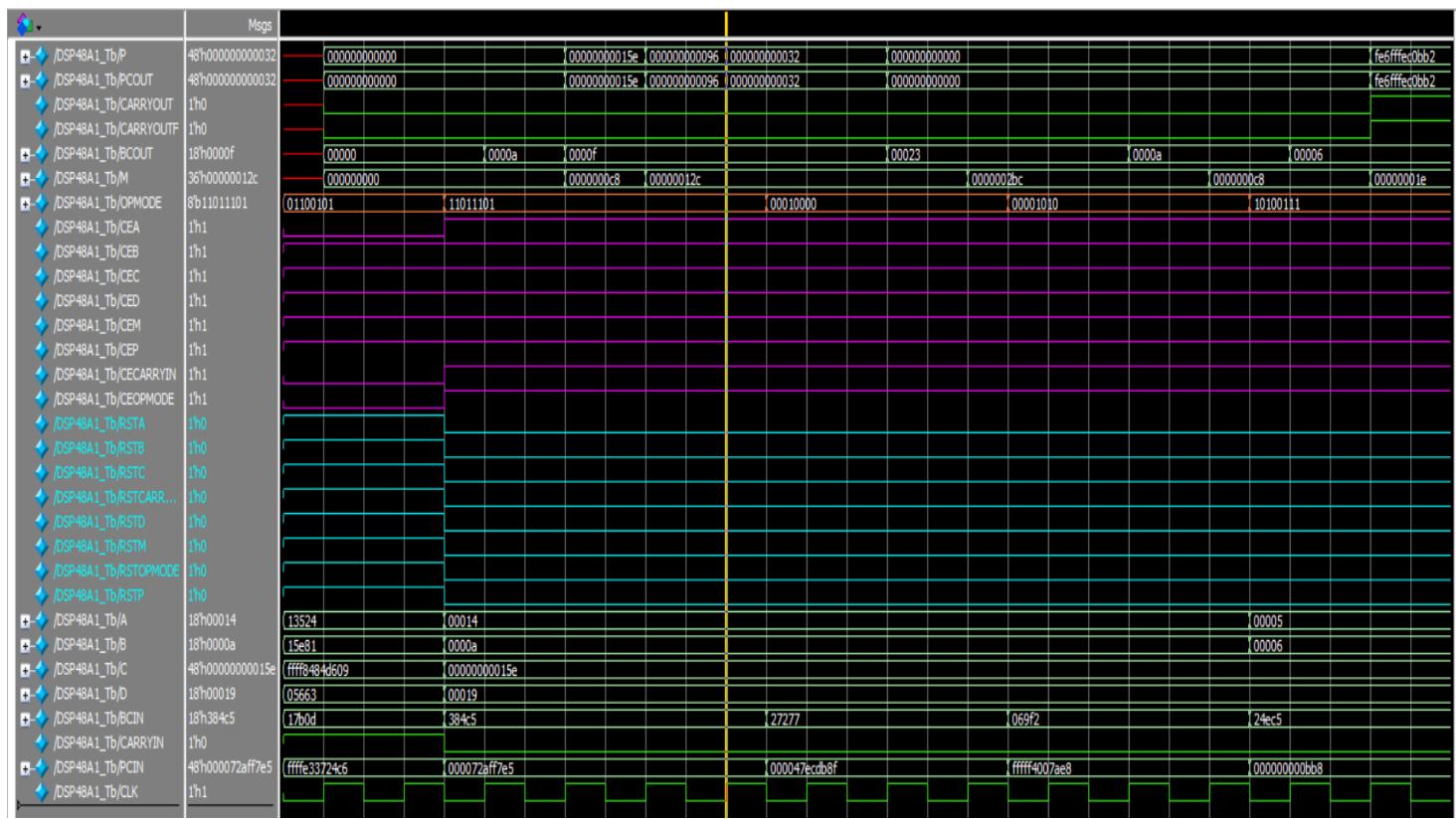
PATH1_tb

```

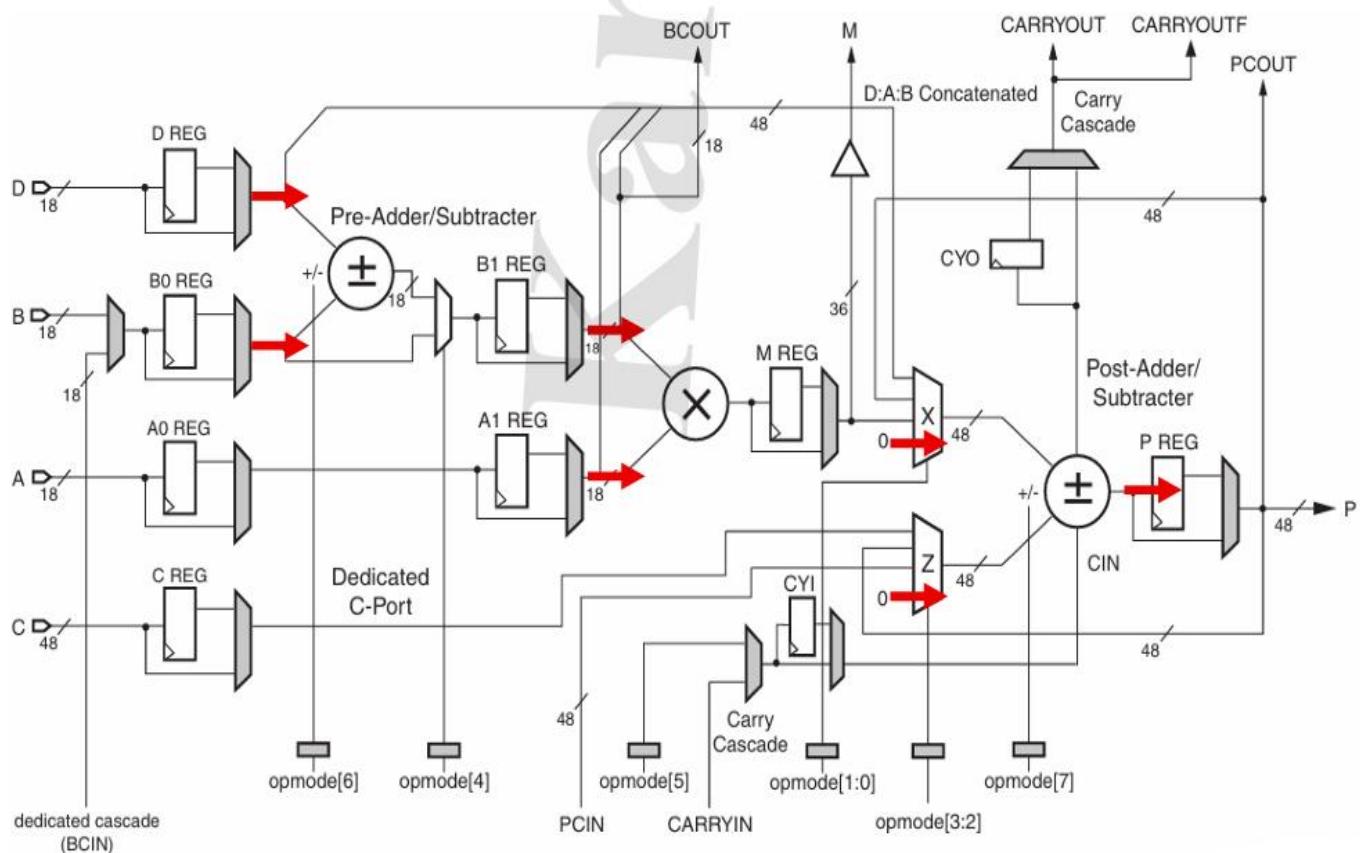
//////PATH1
{RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP}=0;
{CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP}=8'b11111111;
A=20; B=10; C=350; D=25;
BCIN=$random; CARRYIN=$random; PCIN=$random;
OPMODE=8'b11011101;
repeat(4)@(negedge CLK);
if(BCOUT != 'hf || M != 'h12c || P != 'h32 || PCOUT != 'h32 || CARRYOUT != 0 || CARRYOUTF != 0)begin
$display("ERROR IN PATH1");
$stop;
end

```

PATH1_WAVE_PART



PATH2



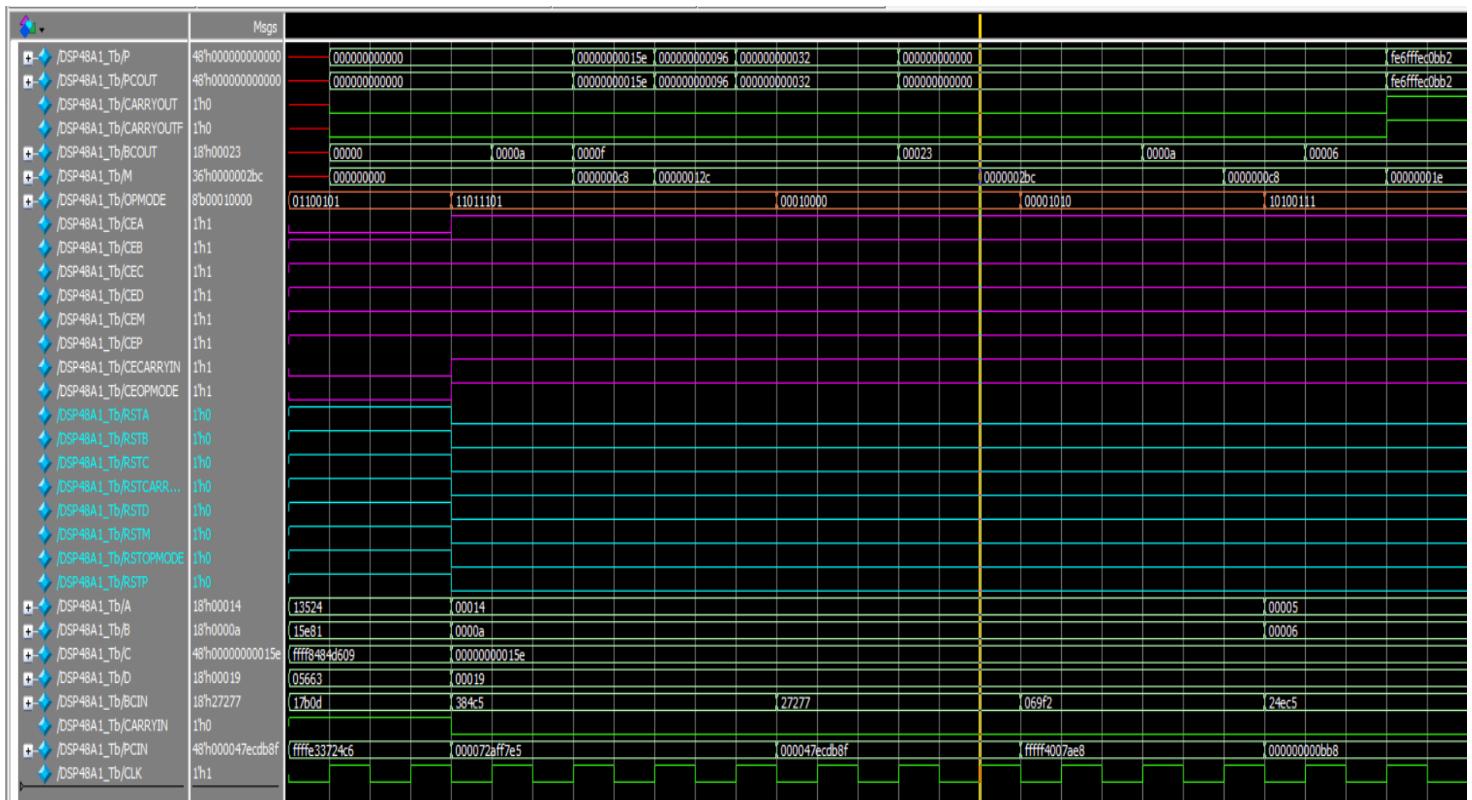
PATH2_tb

```

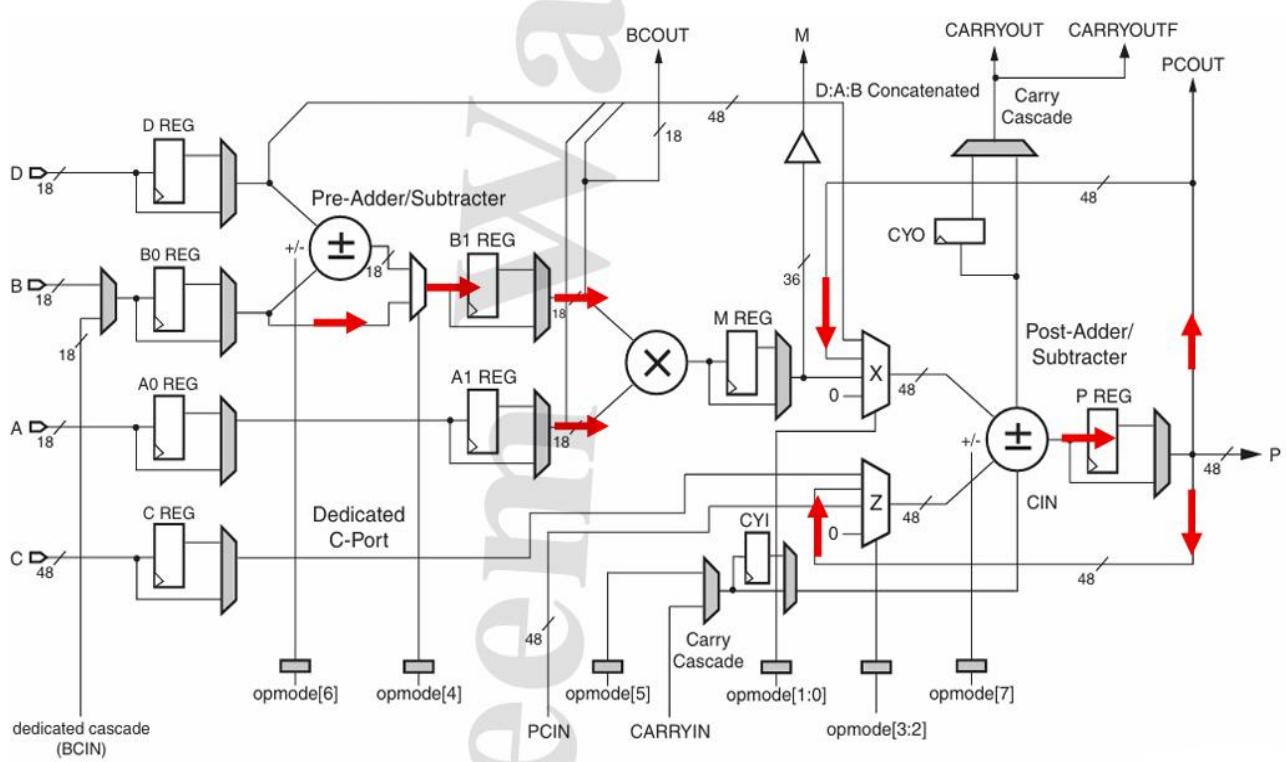
/// Path 2
A=20; B=10; C=350; D=25;
BCIN=$random; CARRYIN=$random; PCIN=$random;
OPMODE=8'b00010000;
repeat(3)@(negedge CLK);
if(BCOUT!='h23 || M!='h2bc || P!=0 || PCOUT!=0 || CARRYOUT!=0 || CARRYOUTF!=0)begin
    $display("ERROR IN PATH2");
    $stop;
end

```

PATH2_WAVE_PART



PATH3



PATH3_tb

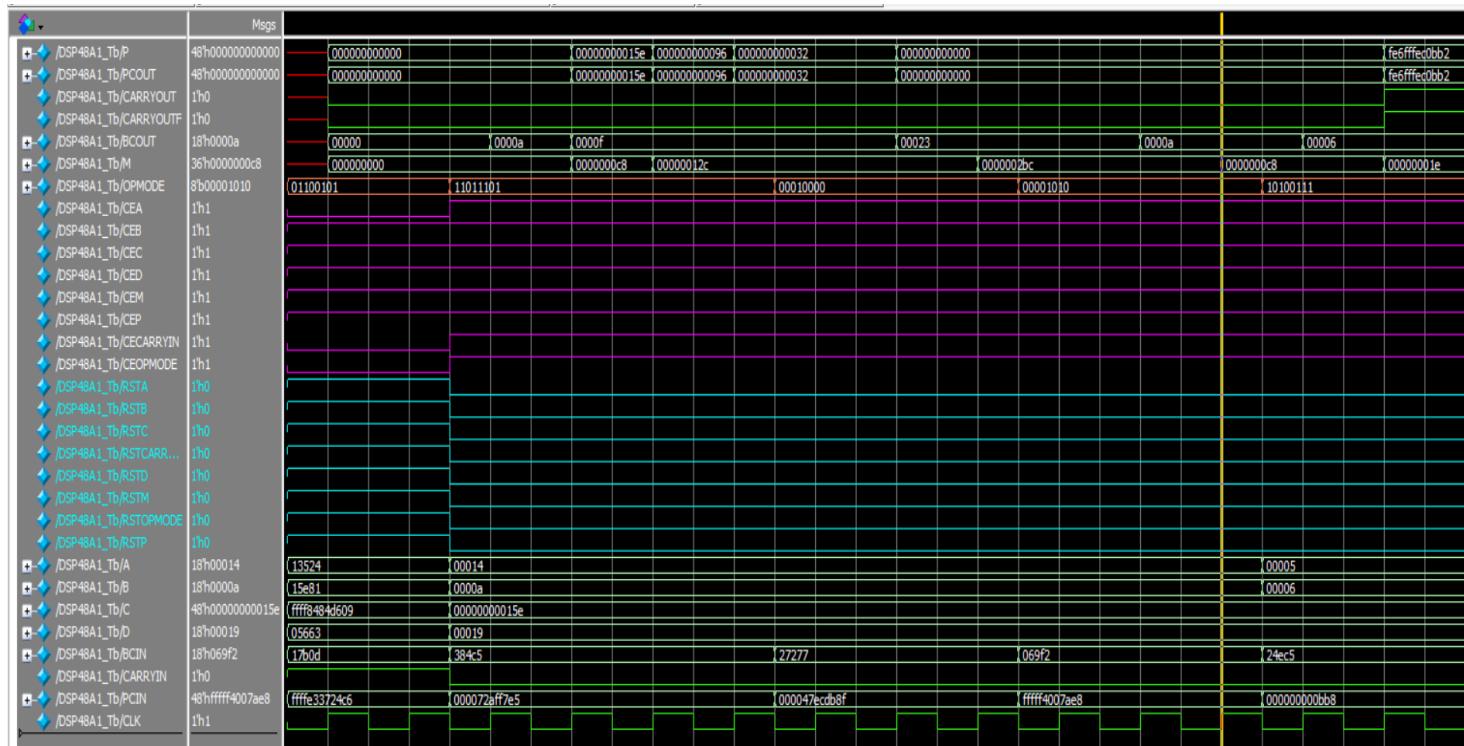
```

/// path 3

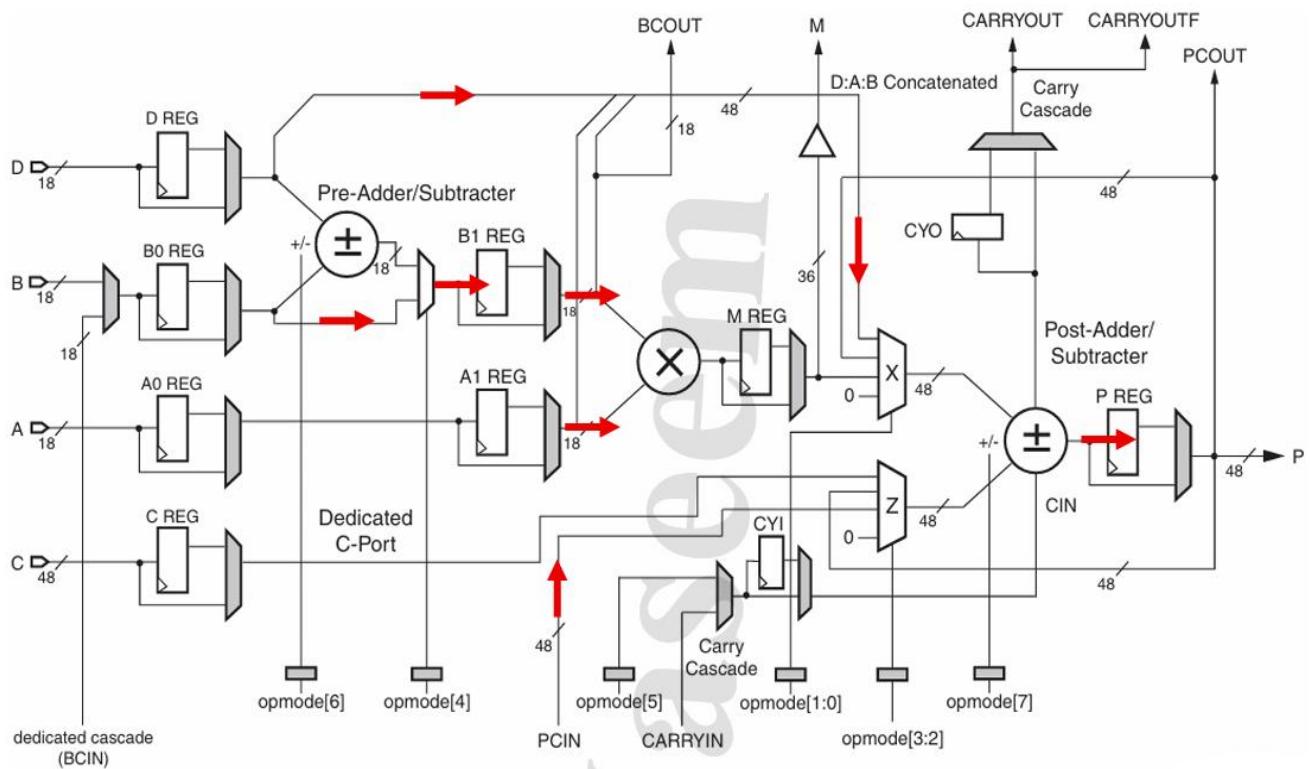
A=20; B=10; C=350; D=25;
BCIN=$random; CARRYIN=$random; PCIN=$random;
OPMODE=8'b00001010;
repeat(3)@(negedge CLK);
if(BCOUT!='ha || M=='hc8 || P!=0 || PCOUT!=0 || CARRYOUT!=0 || CARRYOUTF!=0)begin
    $display("ERROR IN PATH3");
    $stop;
end

```

PATH3_WAVE_PART



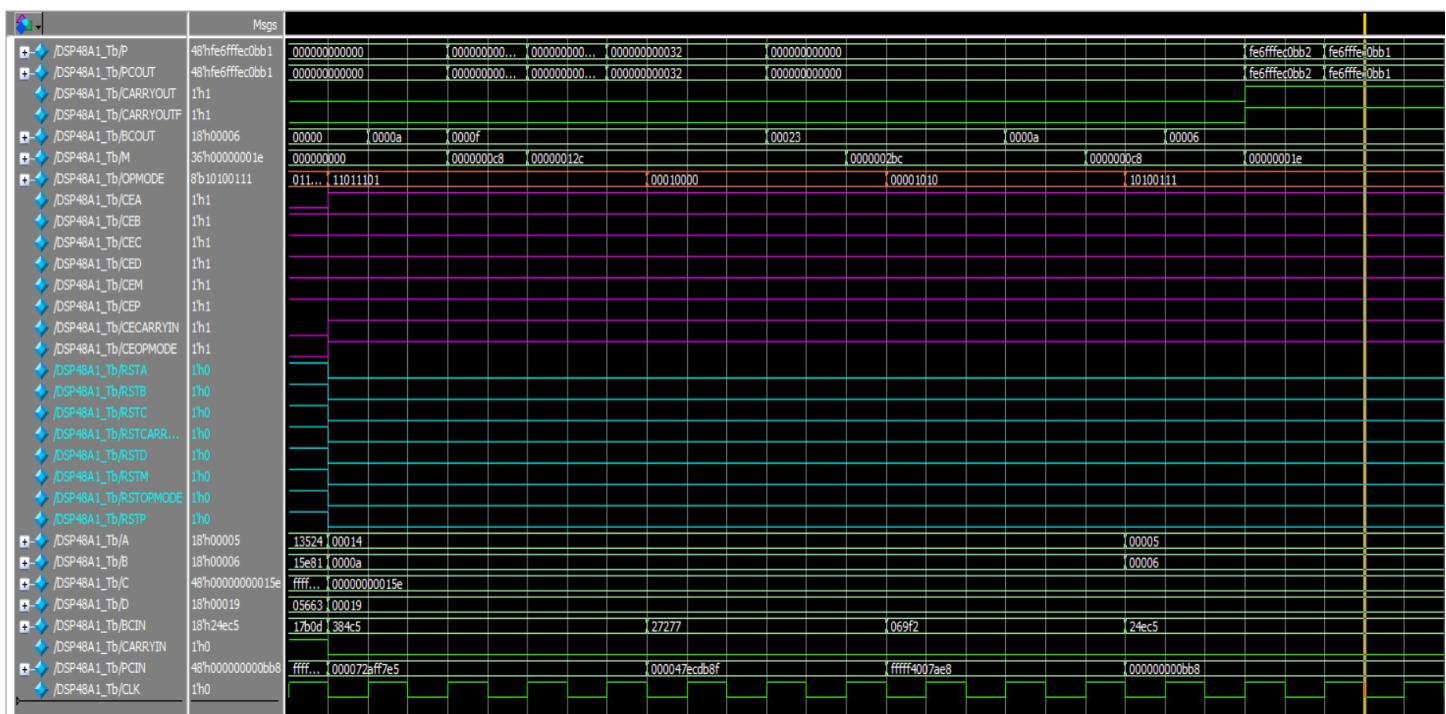
PATH4



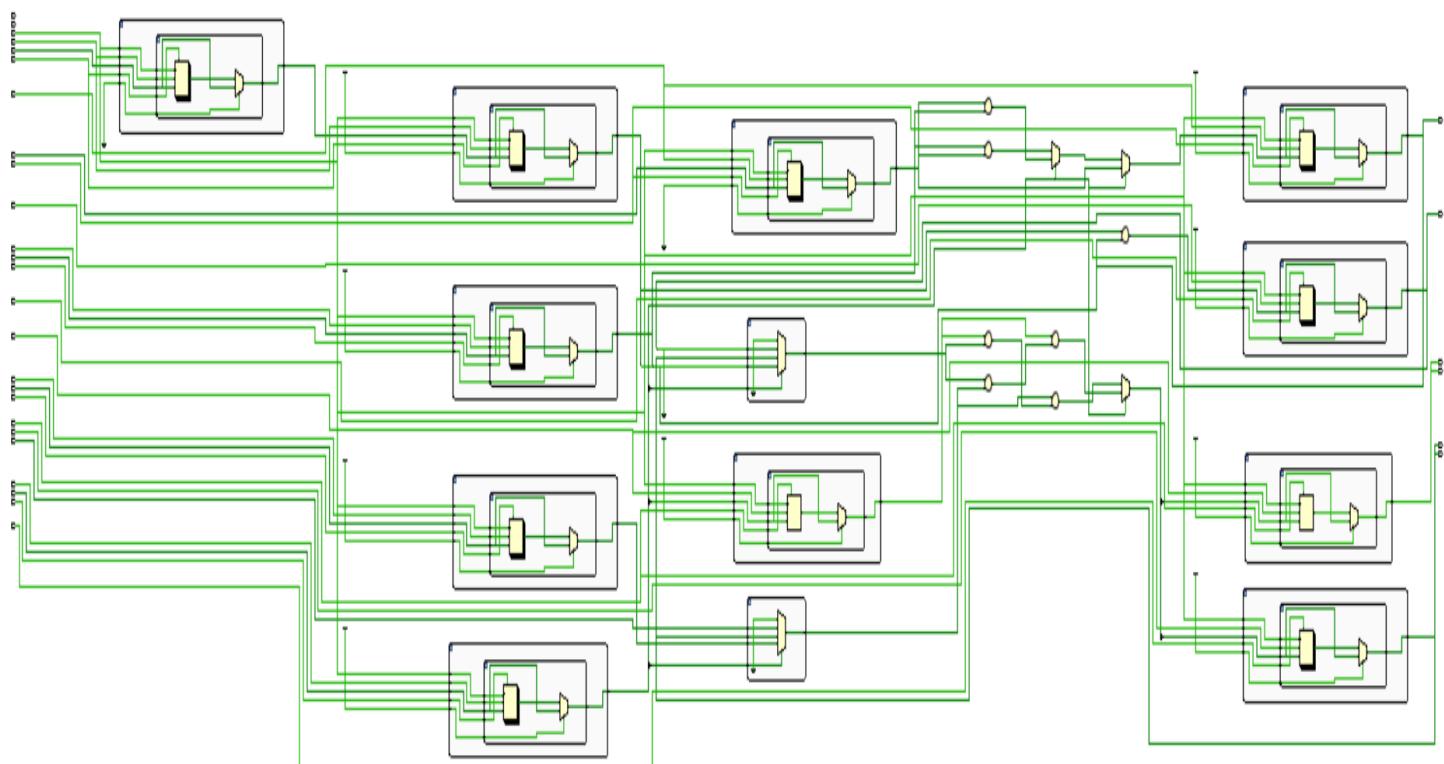
PATH4_tb

```
//// PATH4
A=5; B=6; C=350; D=25;
BCIN=$random; CARRYIN=$random; PCIN=3000;
OPMODE=8'b10100111;
repeat(3)@(negedge CLK);
if(BCOUT!='h6 || M!='h1e || P!='hfe6ffffec0bb1||PCOUT!='hfe6ffffec0bb1||CARRYOUT!=1||CARRYOUTF!=1)begin
    $display("ERROR IN PATH4");
    $stop;
end
@(negedge CLK); // to make last out obvious
$stop;
end
endmodule
```

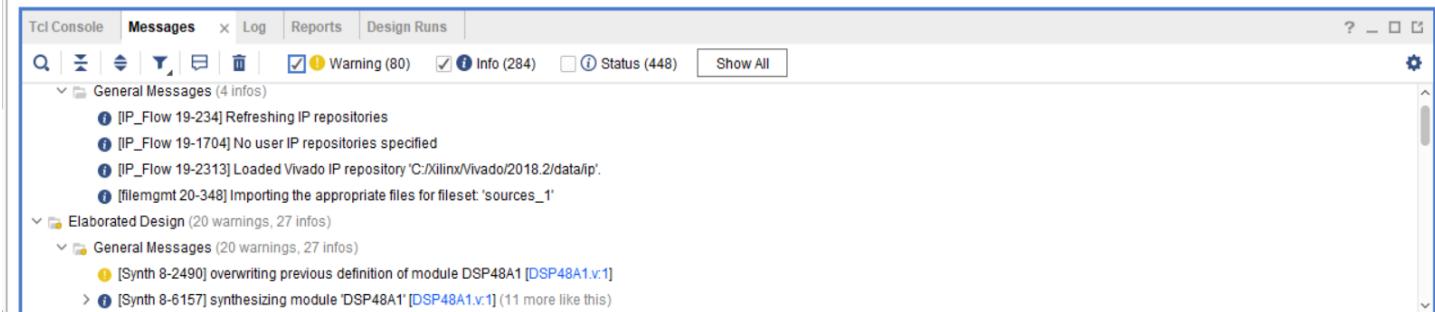
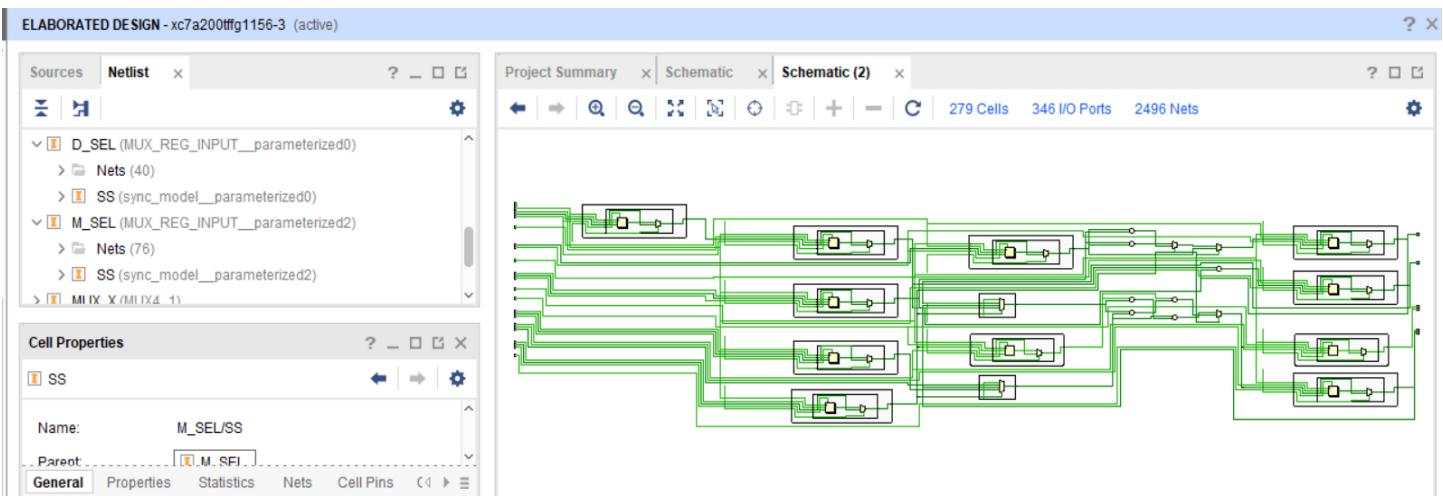
PATH4_WAVE_PART



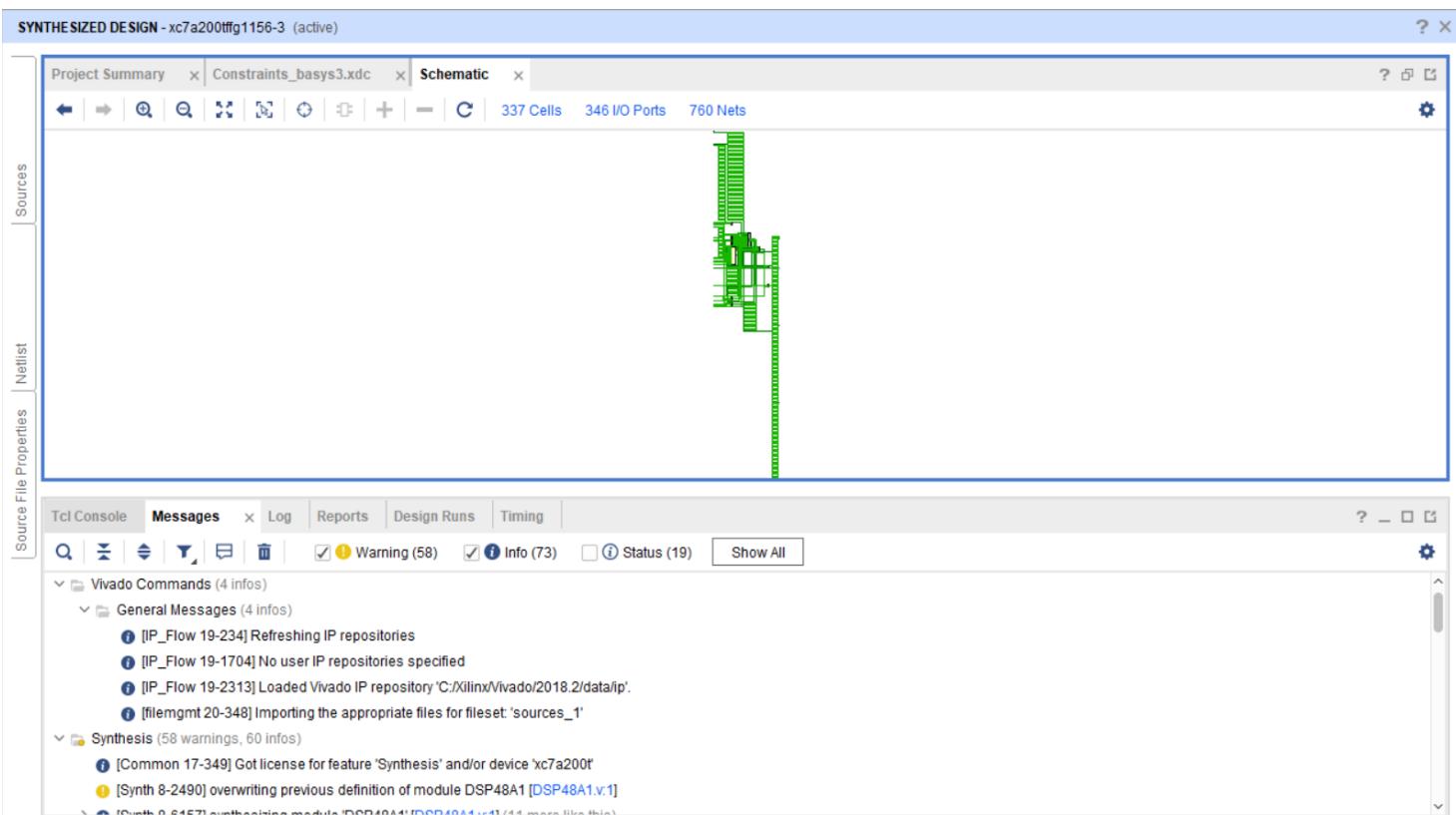
RTL_SCHEMATIC



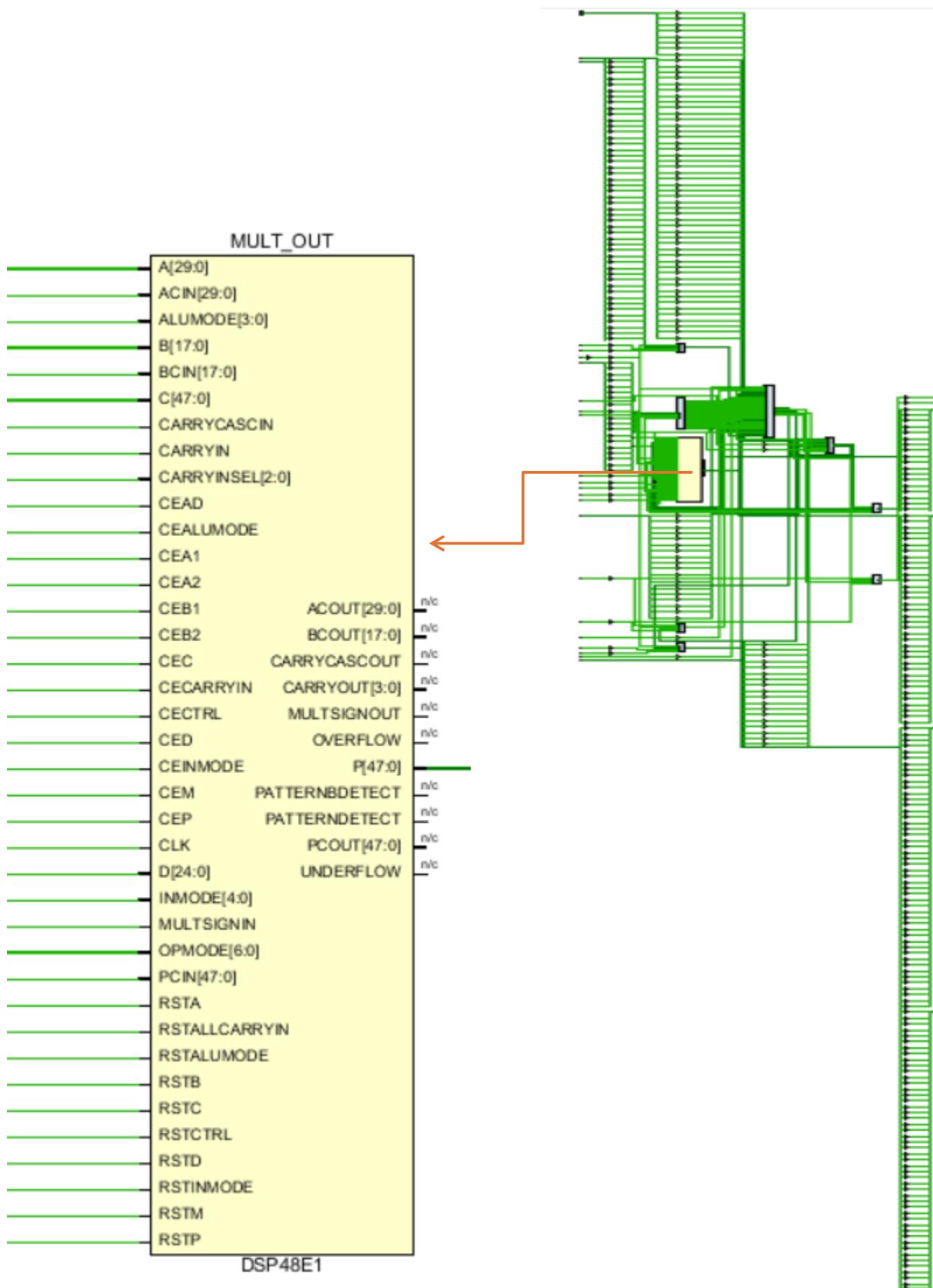
RTL_NO_ERRORS



SYNTHESIZED_NO_ERRORS



SYNTHESIZED_SCHEMATIC



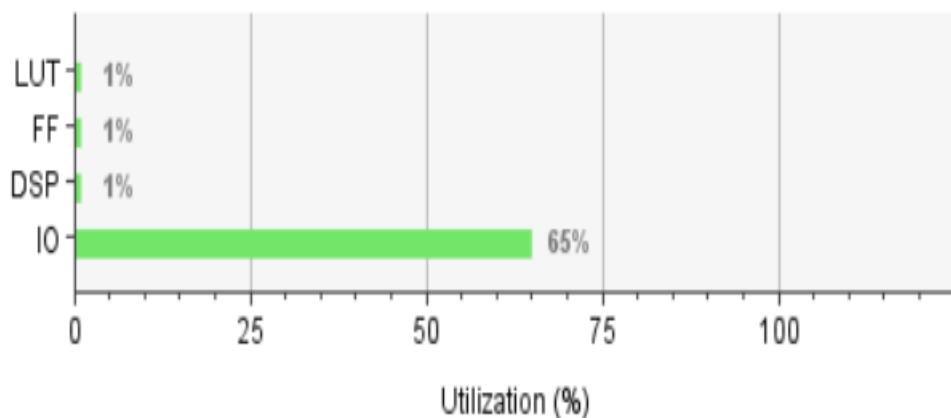
SYNTHESIZED_TIME_REPORT

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.925 ns	Worst Hold Slack (WHS): 0.182 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 87	Total Number of Endpoints: 87	Total Number of Endpoints: 162

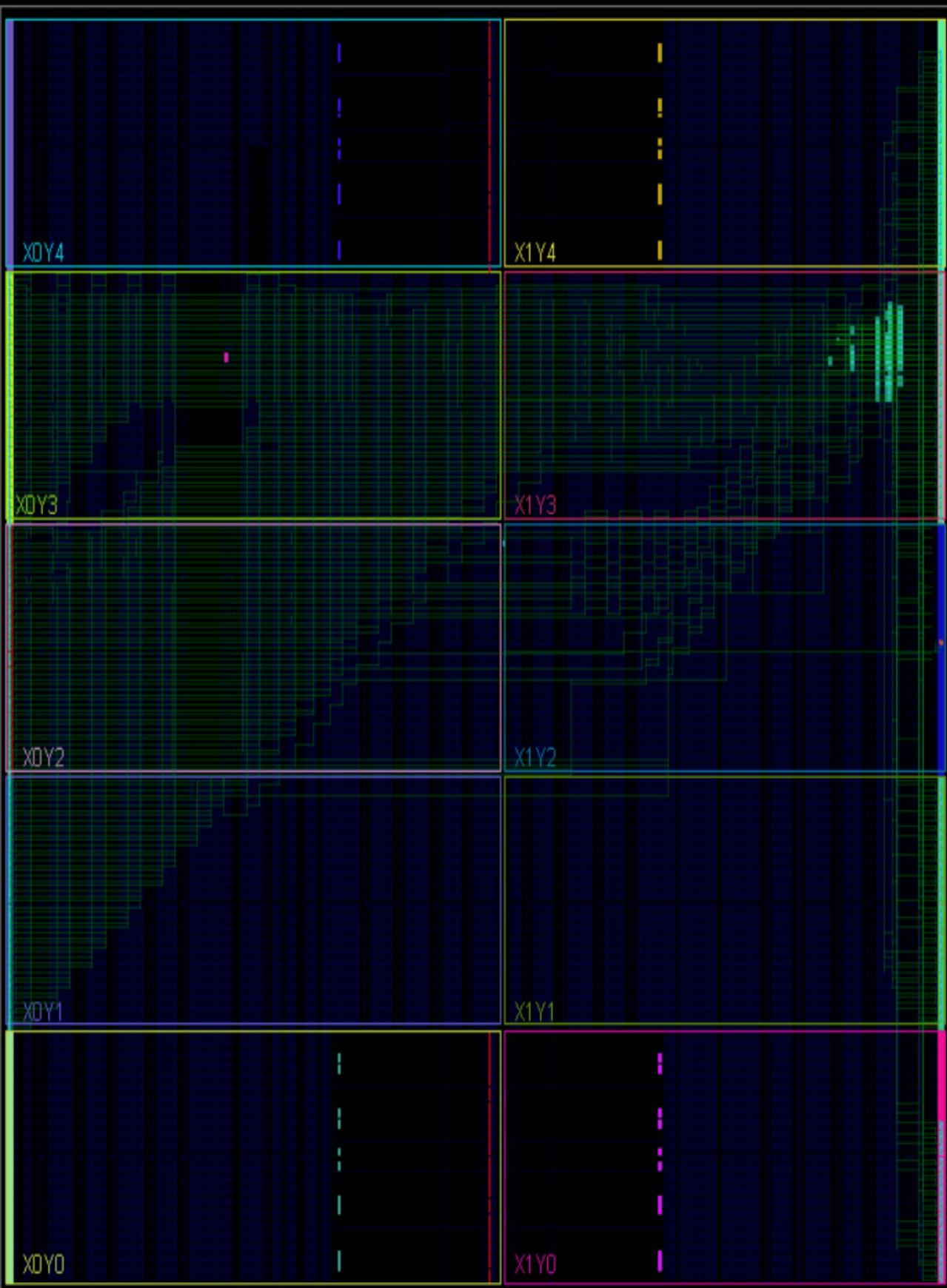
All user specified timing constraints are met.

SYNTHESIZED_UTI_REPORT

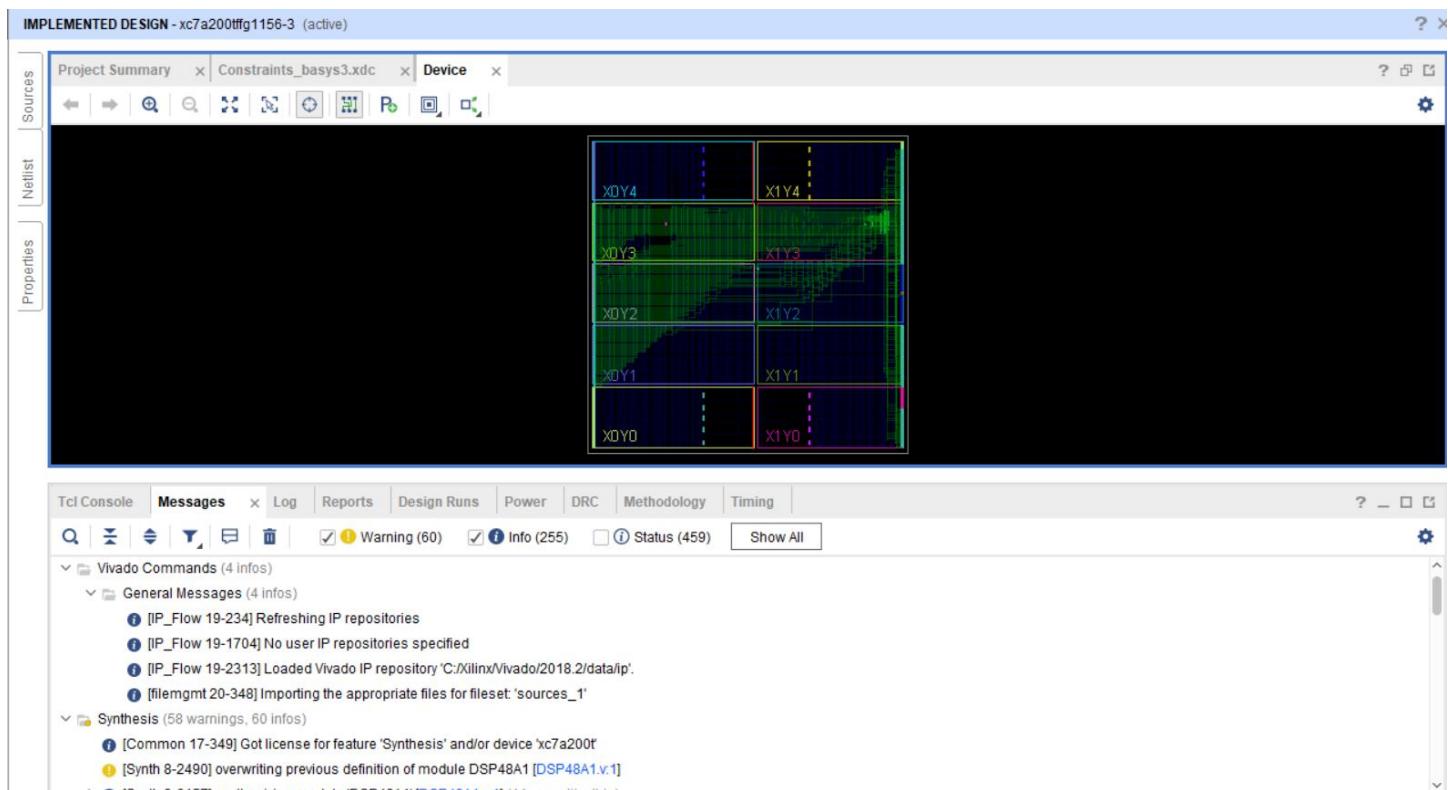
Resource	Utilization	Available	Utilization %
LUT	232	134600	0.17
FF	160	269200	0.06
DSP	1	740	0.14
IO	327	500	65.40



DEVICE



IMP_NO_ERRORS



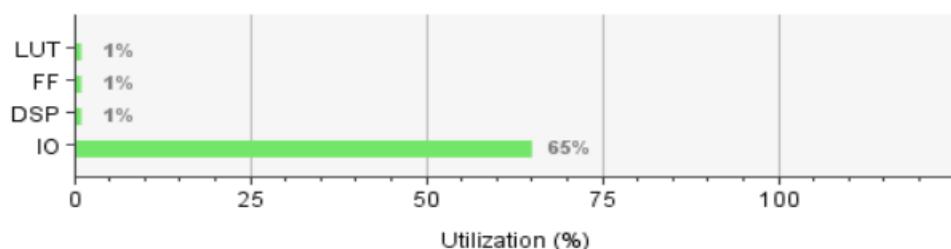
IMP_TIME_REPORT

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 4.603 ns	Worst Hold Slack (WHS): 0.215 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 106	Total Number of Endpoints: 106	Total Number of Endpoints: 181

All user specified timing constraints are met.

IMP_UTI_REPORT

Resource	Utilization	Available	Utilization %
LUT	232	133800	0.17
FF	179	267600	0.07
DSP	1	740	0.14
IO	327	500	65.40



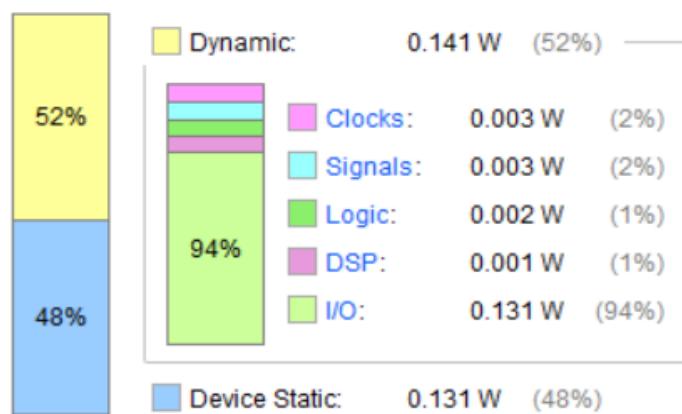
IMP_POWER_REPORT

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 0.272 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 25.4°C
Thermal Margin: 74.6°C (50.8 W)
Effective TJA: 1.5°C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Qlint_Snippet

```

C:/Users/LENOVO/OneDrive/Desktop/ROU/test/DSP48A1.v [DSP48A1]
1 module DSP48A1(A,B,C,D,CARRYIN,M,P,CARRYOUT,CARRYOUTF,CLK,OPMODE,CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP
2   ,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTPMODE,RSTP,BOUT,PCIN,PCOUT,BCIN);
3 parameter A0REG = 0;
4 parameter A1REG = 1;
5 parameter B0REG = 0;
6 parameter B1REG = 1;
7 parameter CREG = 1;
8 parameter DREG = 1;
9 parameter MREG = 1;
10 parameter PREG = 1;
11 parameter OPMODEREG = 1;
12 parameter CARRYINREG = 1;
13 parameter CARRYOUTREG =1;
14 parameter CARRYINSEL ="OPMODE5", // "OPMODE5" or "CARRYIN"
15 parameter B_INPUT = "DIRECT"; // "DIRECT" or "CASCADE"
16 parameter RSTTYPE = "SYNC"; // "SYNC" or "ASYNC"
17 input [17:0] A,B,D,BCIN;
18 input [47:0] PCIN;
19 input [47:0] C;
20 input [7:0]OPMODE;
21 input CARRYIN,CLK,CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP;
22 input RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTPMODE,RSTP;
23 output [17:0] BOUT;
24 output [35:0] M.
bb.v DSP48A1.v
Lint Checks
Filter: Type here Waived Fixed Pending Uninspected Bug Verified Total: 0 Selected: 0
Severity Status Check Alias Message Module Category State Owner STARC Reference
Transcript Message Viewer Lint Checks Design Metrics Design Information

```

 **Contact**

- **Email:** mohamedmouse066@gmail.com
- **Linkedin:** [Press Here](#)

Project Sources

- **GitHub:** [Press Here](#)