



Faculty Of Engineering Ain Shams University
Electronics and Communications Department

Project (1)

Spartan6 - DSP48A1

Prepared by

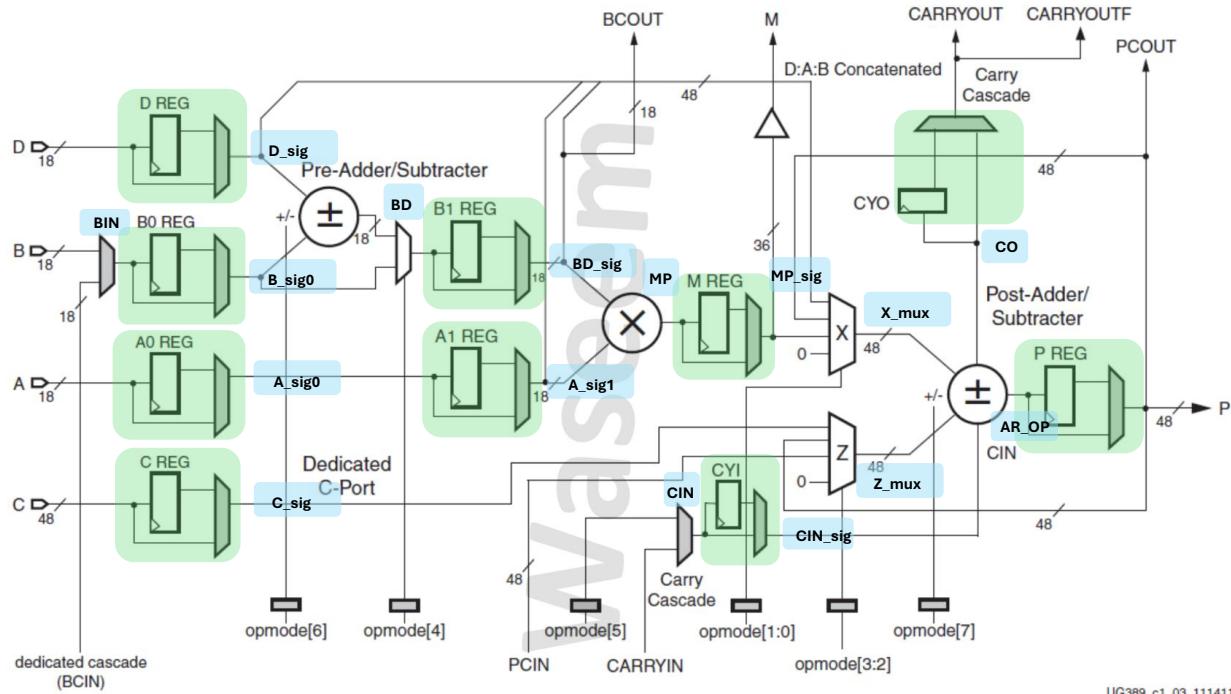
Omar Mohamed Hussein Mostafa

Table of Contents

.....	1
Summary	4
Minor Blocks.....	5
Verilog RTL code.....	5
DSP main RTL code:	6
Testbench code:	8
Do file	10
Questasim waveforms	11
Reset check	11
Path 1 check	12
Path 2 check	13
Path 3 check	14
Path 4 check	15
Lint check	16
Constrain file	16
Synthesis tool results	19
Elaboration	19
schematic	19
Messages	20
Synthesis	20
Schematic.....	20
Message.....	21
Utilities report.....	21
Timing report	22
Implementation	22

Schematic.....	22
Messages	23
Utilities report.....	23
Timing report	24
Debug cores	24

Summary



The green boxes represent repeated blocks in our design and will therefore be instantiated as separate modules. The blue boxes represent internal signals, making it easier to treat the DSP design like a puzzle, where we simply connect the signals between the modules!

Minor Blocks

The green box is our parameterized **seq_comb** module.

Verilog RTL code

```
● ○ ●
1  module seq_comb #(parameter WIDTH=18,parameter RSTTYPE="SYNC" )(in,out,clk,en,rst,REG);
2
3  input [WIDTH-1:0] in;
4  input REG,en,rst,clk;
5  output wire[WIDTH-1:0]out;
6  reg [WIDTH-1:0] S_seq,A_seq; //A stands for async
7  generate
8      if(RSTTYPE=="SYNC") begin
9          always @(posedge clk) begin
10              if(rst) begin
11                  S_seq<='b0;
12              end
13              else begin
14                  if(en) begin
15                      S_seq<=in;
16                  end
17              end
18          end
19          assign out=(REG)? S_seq:in;
20      end
21      else if (RSTTYPE=="ASYNC") begin
22          assign out=(REG)? A_seq:in;
23          always @(posedge clk or posedge rst) begin
24              if(rst) begin
25                  A_seq<='b0;
26              end
27              else begin
28                  if(en) begin
29                      A_seq<=in;
30                  end
31              end
32          end
33      end
34  endgenerate
35 endmodule
```

DSP main RTL code:

```
1 module DSP#(
2     parameter A0REG=0,parameter A1REG=1,parameter B0REG=0, parameter B1REG=1,
3     parameter CREG=1,parameter DREG=1,parameter MREG=1,parameter PREG=1,parameter CARRYINREG=1,
4     parameter CARRYOUTREG=1,parameter OPMODEREG=1,parameter CARRYINSEL= "OPMODE5",
5     parameter B_INPUT="DIRECT",parameter RSTTYPE="SYNC")
6 (
7     A,B,C,D,CARRYIN,M,P,CARRYOUT,CARRYOUTF,CLK,OPMODE,CEA,CEB,CEC,CED,CECARRYIN,CEM,CEOPMODE,CEP,
8     RSTA,RSTB,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,RSTP,RSTM,BCOUT,BCIN,PCIN,PCOUT
9 );
10
11 // I/P ports
12 input [17:0]A,B,BCIN,D;
13 input [47:0]C,PCIN;
14 input [7:0]OPMODE;
15 input RSTA,RSTB,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,RSTP,RSTM,CEA,CEB,CEC,CED,CECARRYIN,CEM,CEOPMODE,CEP,CLK,CARRYIN;
16 output [17:0]BCOUT;
17 output[47:0] PCOUT,P;
18 output CARRYOUT,CARRYOUTF;
19 output [35:0]M;
20
21 //internal signals
22 wire [7:0]OPMODE_reg;
23 wire [17:0]D_sig,B_sig0,A_sig0,A_sig1,BD_sig;
24 wire[47:0]C_sig,P_sig;
25 wire[35:0]MP_sig;
26 wire CIN_sig,CO;
27 reg[47:0] X_mux,Z_mux;
28 reg[48:0]AR_OP;
29 reg[35:0]MP;
30 reg [17:0]BD;
31
32 //instantiate the sequential/combinational blocks
33 // #(parameter WIDTH=18,parameter RSTTYPE="SYNC",parameter REG=1)(in,out,clk,en,rst);
34 seq_comb #(18,"SYNC") A0_REG(A,A_sig0,CLK,CEA,RSTA,A0REG);
35 seq_comb #(18,"SYNC") A1_REG(A,A_sig0,A_sig1,CLK,CEA,RSTA,A1REG);
36 seq_comb #(48,"SYNC") C_REG(C,C_sig,CLK,CEC,RSTC,CREG);
37 seq_comb #(18,"SYNC") D_REG(D,D_sig,CLK,CED,RSTD,DREG);
38 seq_comb #(8,"SYNC") OP_MODE_REG(OPMODE,OPMODE_reg,CLK,CEOPMODE,RSTOPMODE,OPMODEREG);
39 seq_comb #(18,"SYNC") B1_REG(BD,BD_sig,CLK,CEB,RSTB,B1REG);
40 seq_comb #(48,"SYNC") P_REG(P_sig,P,CLK,CEP,RSTP,PREG);
41 seq_comb #(1,"SYNC") CYO(CO,CARRYOUT,CLK,CECARRYIN,RSTCARRYIN,CARRYINREG);
42 seq_comb #(36,"SYNC") MP_REG(MP,MP_sig,CLK,CEM,RSTM,MREG);
43
44 //instantiation blocks that depend on a parameters
45 generate
46     if (B_INPUT == "DIRECT") begin
47         seq_comb #(18, "SYNC") B0_REG(B, B_sig0, CLK, CEB, RSTB,B0REG);
48     end
49     else if (B_INPUT == "CASCADE") begin
50         seq_comb #(18, "SYNC") B0_REG(BCIN, B_sig0, CLK, CEB, RSTB,B0REG);
51     end
52     else begin
53         wire [17:0] zero_wire = 18'b0;
54         seq_comb #(18, "SYNC") B0_REG(zero_wire, B_sig0, CLK, CEB, RSTB,B0REG);
55     end
56 endgenerate
57 generate
58     if (CARRYINSEL == "OPMODE5") begin
59         seq_comb #(1, "SYNC") CARRY_IN(OPMODE_reg[5], CIN_sig, CLK, CECARRYIN, RSTCARRYIN,CARRYINREG);
60     end
61     else if (CARRYINSEL == "CARRYIN") begin
62         seq_comb #(1, "SYNC") CARRY_IN(CARRYIN, CIN_sig, CLK, CECARRYIN, RSTCARRYIN,CARRYINREG);
63     end
64     else begin
65         wire zero_wire = 1'b0;
66         seq_comb #(1, "SYNC") CARRY_IN(zero_wire,CIN_sig, CLK, CECARRYIN, RSTCARRYIN,CARRYINREG);
67     end
68 end
```

```
1 //assign statements
2 assign PCOUT = P;
3 assign BCOUT = BD_sig;
4 assign M = MP_sig;
5 assign P_sig=AR_OP[47:0];
6 assign CARRYOUT=CARRYOUT;
7 assign CO = AR_OP[48];
8
9
10 always @(*) begin
11     case(OPMODE_reg[4])
12         1'b0: BD=B_sig0;
13         1'b1:begin
14             case(OPMODE_reg[6])
15                 1'b1: BD=D_sig-B_sig0;
16                 1'b0: BD=D_sig+B_sig0;
17             endcase
18         end
19     endcase
20     MP = BD_sig*A_sig1;
21     case (OPMODE_reg[1:0])
22         2'b00: X_mux='b0;
23         2'b01: X_mux={12'b0,MP_sig};
24         2'b10: X_mux=P;
25         2'b11: X_mux={D[11:0], A[17:0],B[17:0]};
26     endcase
27     case(OPMODE_reg[3:2])
28         2'b00: Z_mux='b0;
29         2'b01: Z_mux=PCIN;
30         2'b10: Z_mux=P;
31         2'b11: Z_mux={30'b0,C_sig};
32     endcase
33     case (OPMODE_reg[7])
34         2'b0: begin
35             AR_OP=Z_mux+X_mux+CIN_sig;
36         end
37         2'b1: begin
38             AR_OP=Z_mux-(X_mux+CIN_sig);
39         end
40     endcase
41 end
42 endmodule
```

Testbench code:

```
1  module DSP_tb();
2
3  reg [17:0]A,B,BCIN,D;
4  reg [47:0]C,PCIN;
5  reg [7:0]OPMODE;
6  reg RSTA,RSTB,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,RSTP,RSTM,CEA,CEB,CEC,CED,CECARRYIN,CEM,CEOPMODE,CEP,CLK,CARRYIN;
7  wire [17:0]BCOUT;
8  wire[47:0] PCOUT_P;
9  wire CARRYOUT,CARRYOUTF;
10 wire [35:0]M;
11
12 DSP #(0,1,0,1,1,1,1,1,1,1,1,"OPMODE5","DIRECT","SYNC") DUT(
13     A,B,C,D,CARRYIN,M,P,CARRYOUT,CARRYOUTF,CLK,OPMODE,CEA,CEB,CEC,CED,CECARRYIN,CEM,CEOPMODE,CEP,
14     RSTA,RSTB,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,RSTP,RSTM,BCOUT,BCIN,PCIN,PCOUT
15 );
16
17 initial begin
18     CLK=0;
19     forever begin
20         #10 CLK=~CLK;
21     end
22 end
23
24 initial begin
25 //RESET check
26     RSTA=1;
27     RSTB=1;
28     RSTC=1;
29     RSTD=1;
30     RSTCARRYIN=1;
31     RSTOPMODE=1;
32     RSTP=1;
33     RSTM=1;
34     repeat(10) begin
35         A=$random;
36         B=$random;
37         BCIN=$random;
38         D=$random;
39         C=$random;
40         PCIN=$random;
41         OPMODE=$random;
42         CEA=$random;
43         CEB=$random;
44         CEC=$random;
45         CED=$random;
46         CECARRYIN=$random;
47         CEM=$random;
48         CEOPMODE=$random;
49         CEP=$random;
50         CARRYIN=$random;
51
52         @(negedge CLK);
53         if(M!=0 && CARRYOUT!=0 && CARRYOUTF!=0 && P!=0 && PCOUT!=0 && BCOUT!=0) begin
54             $display("Error in RESET");
55             $stop;
56         end
57     end

```

```

1 //Deactivate RESET
2     RSTA=0;
3     RSTB=0;
4     RTC=0;
5     RSTD=0;
6     RSTCARRYIN=0;
7     RSTOPMODE=0;
8     RSTP=0;
9     RSTM=0;
10 //Activate enable signals
11    CEA=1;
12    CEB=1;
13    CEC=1;
14    CED=1;
15    CECARRYIN=1;
16    CEM=1;
17    CEOPMODE=1;
18    CEP=1;
19
20 //verify path 1
21    OPMODE=8'b11011101; //multiplier product,using the C port,use pre adder/sub ,carry with 0,using subtraction
22    A = 20;
23    B = 10;
24    C = 350; //we expect 350-((25-10)*20)=50
25    D = 25;
26    repeat(10)begin
27        BCIN=$random;
28        PCIN=$random;
29        CARRYIN=$random;
30        repeat(4) @(posedge CLK);
31        if(BCOUT!='hf && M!='h12c && P!='h32 && PCOUT!='h32 && CARRYOUT!=0 && CARRYOUTF!=0) begin
32            $display("Error in PATH 1");
33            $stop;
34        end
35    end
36
37 //verify path 2
38    OPMODE=8'b00010000; //using the zeros for the post add/sub,use pre adder/sub ,carry with 0,using addition
39    A = 20;
40    B = 10;
41    C = 350;
42    D = 25;
43    repeat(10)begin
44        BCIN=$random;
45        PCIN=$random;
46        CARRYIN=$random;
47        repeat(3) @(posedge CLK);
48        if(BCOUT=='h23 && M!='h2bc && P!='h0 && PCOUT=='h0 && CARRYOUT!=0 && CARRYOUTF!=0) begin
49            $display("Error in PATH 2");
50            $stop;
51        end
52    end
53
54 //verify path 3
55    OPMODE=8'b00001010; //using p for X and Z MUX for the post add/sub,use B directly,carry with 0,using addition, no pre add/sub
56    A = 20;
57    B = 10;
58    C = 350;
59    D = 25;
60    repeat(10)begin
61        BCIN=$random;
62        PCIN=$random;
63        CARRYIN=$random;
64        repeat(3) @(posedge CLK);
65        if(BCOUT=='ha && M!='hc8 && P!=PCOUT && CARRYOUT!=CARRYOUTF) begin//recheck///////////
66            $display("Error in PATH 3");

```

```
1
2 //verify path 4
3 OPMODE=8'b10100111; //concatenated DBA,PCIN,no pre add/sub, subtracting conc DAB and PCIN,
4 A=5;
5 B=6;
6 C=350;
7 D=25;
8 PCIN=3000;
9 repeat(10)begin
10     BCIN=$random;
11     CARRYIN=$random;
12     repeat(3) @(negedge CLK);
13     if(BCOUT!='16 && M!='11e && P!='hfe6fffec0bb1 && PCOUT!='hfe6fffec0bb1 && CARRYOUT!=1 && CARRYOUTF!=1) begin
14         $display("Error in PATH 4");
15         $stop;
16     end
17 end
18 $stop;
19 end
20 endmodule
```

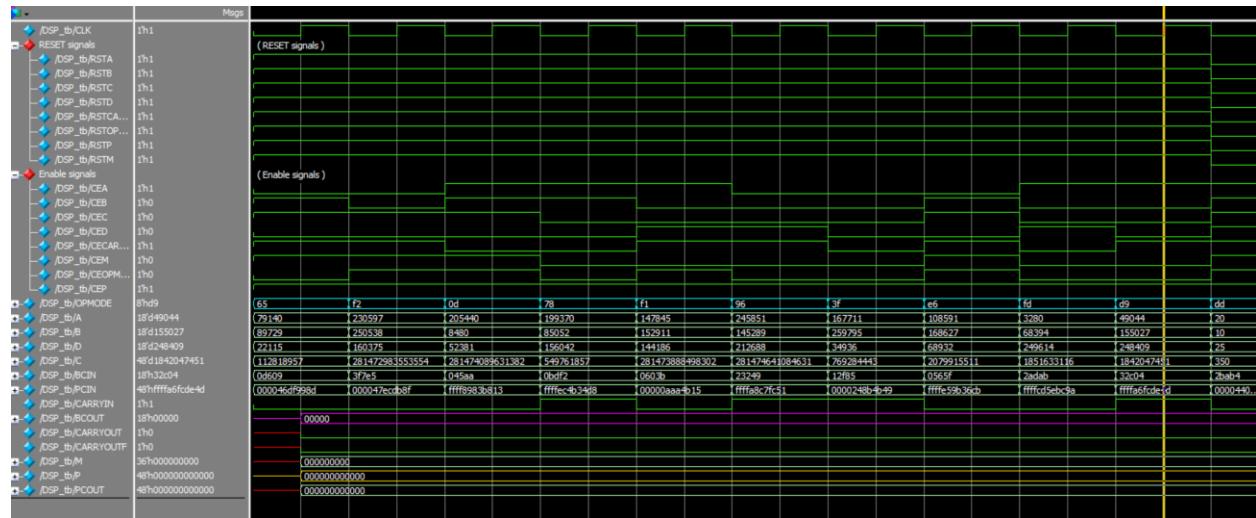
Do file

```
1 vlib work
2 vlog DSP_op.v DSP_tb.v seq_comb_op.v
3 vsim -voptargs+=acc work.DSP_tb
4 add wave *
5 run -all
6 #quit -sim
```

Questasim waveforms

Reset check

Here all the reset signals are activated and all the output signals became zeros.



After we had checked for the functionality of the RESET signal , we will deactivate it and make the enable activated to check the remaining paths.

Also we have grouped all the reset and enable signals for clear vision.

Path 1 check

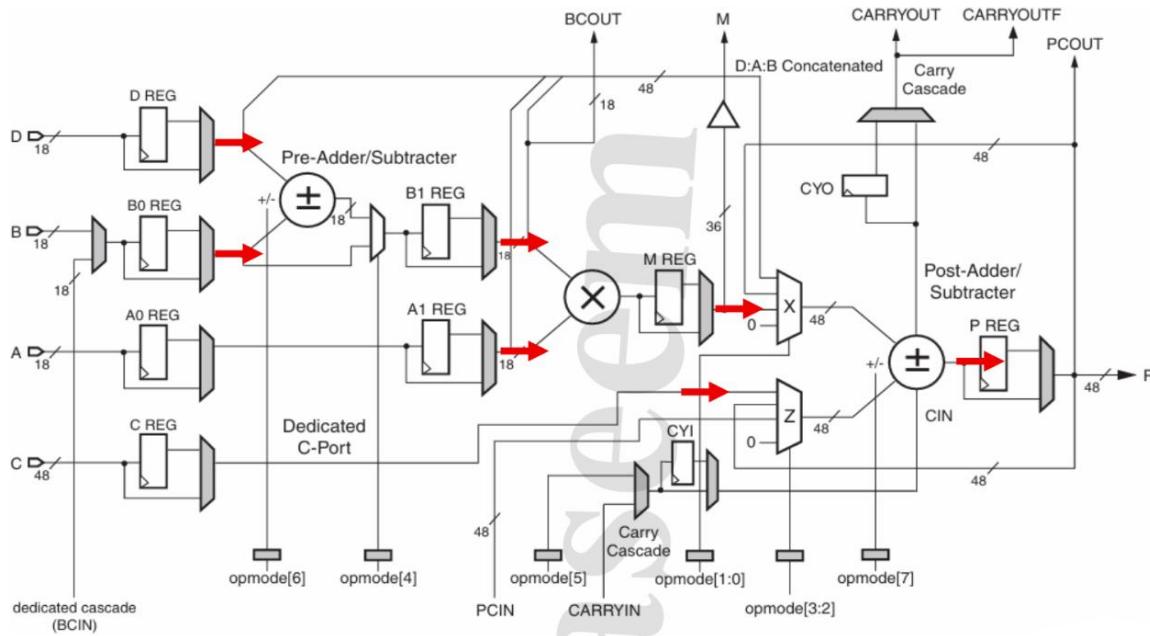
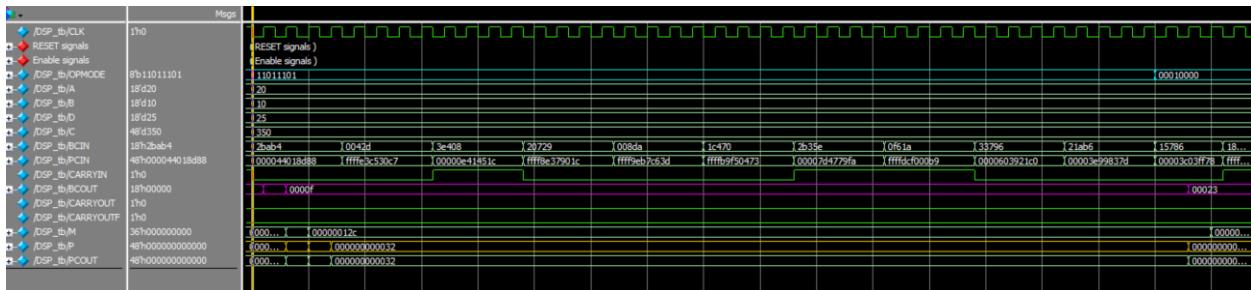


Figure 1: Path 1 Data Flow



For the input shown, the output sticks to the expected values although the remaining inputs are changing.

Path 2 check

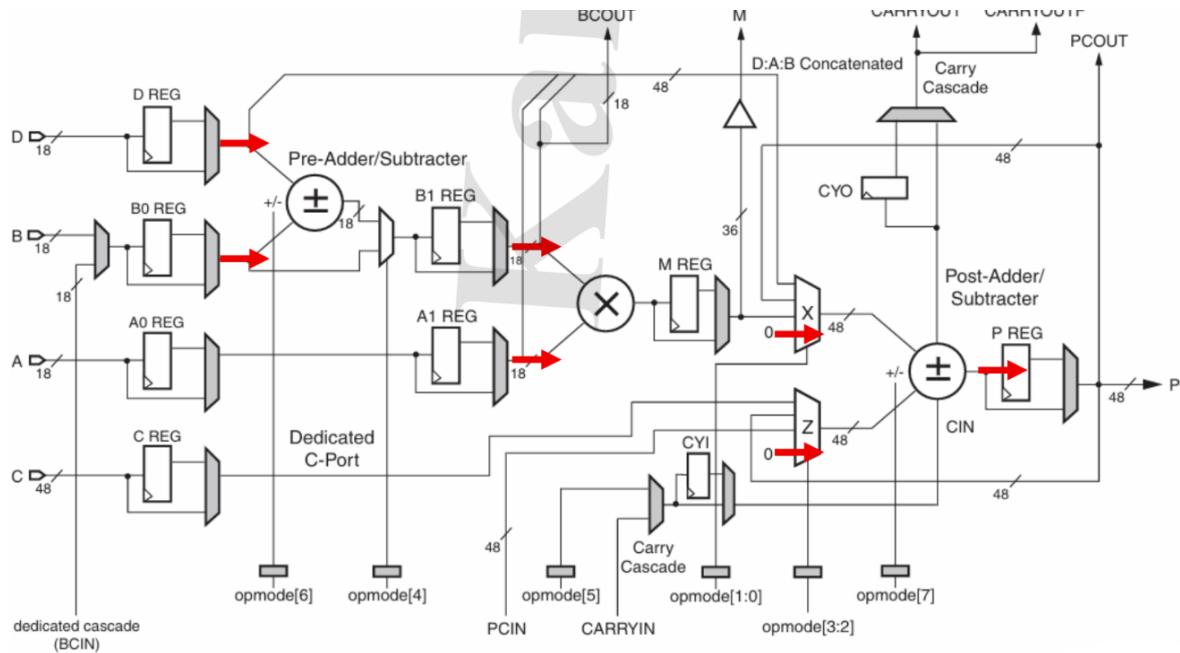
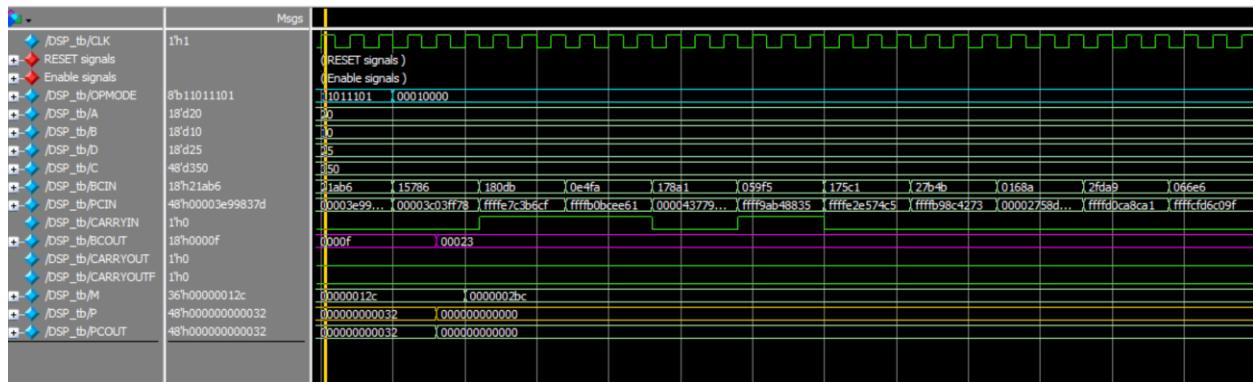
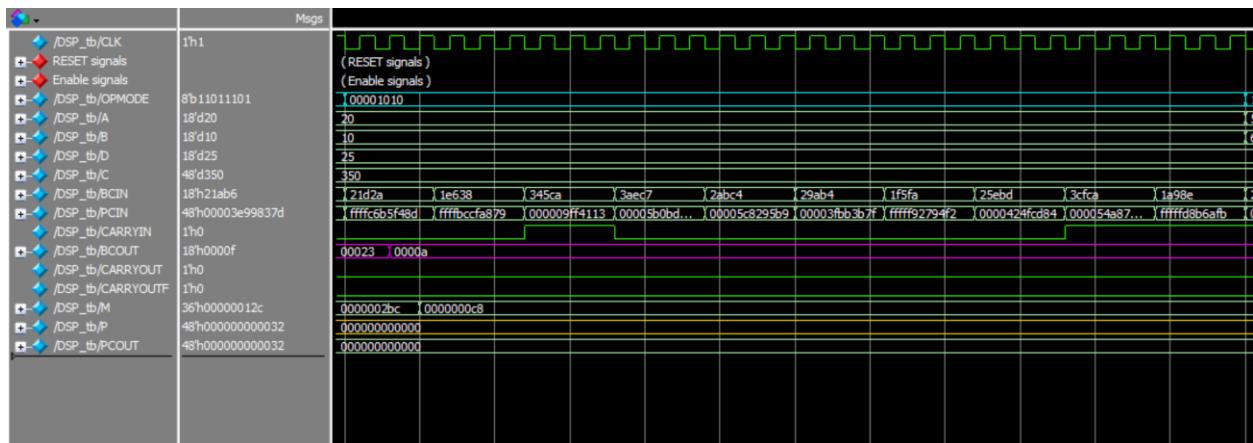
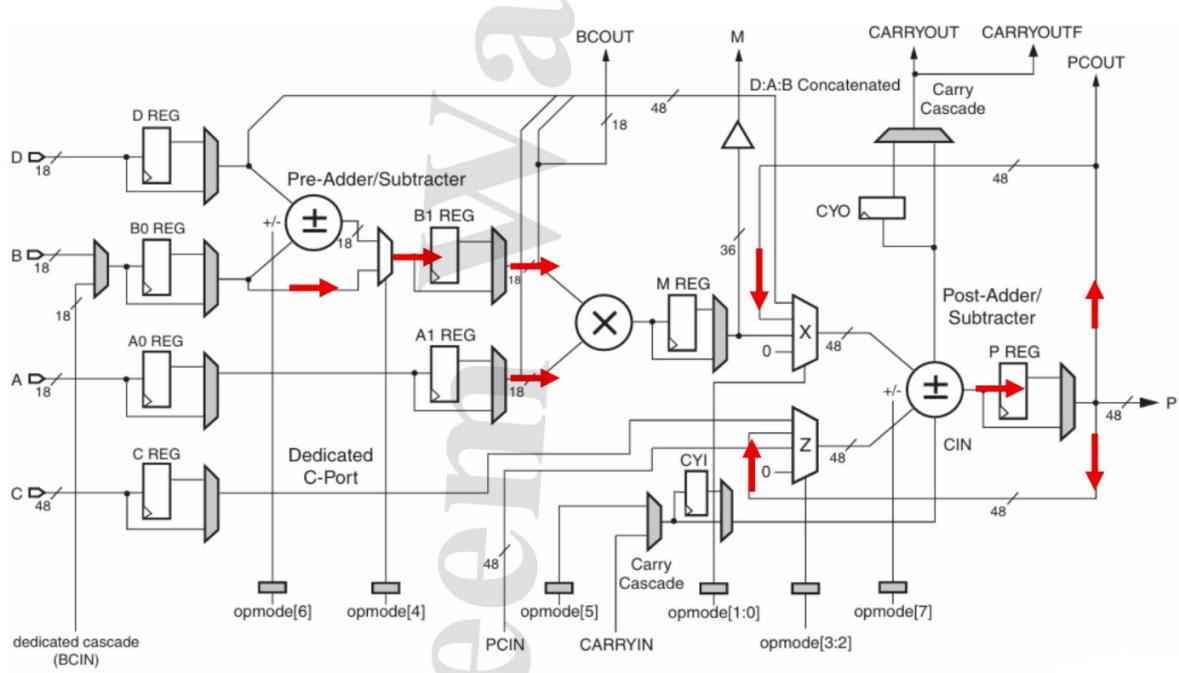


Figure 2: Path 2 Data Flow



For the input shown, the output sticks to the expected values although the remaining inputs are changing.

Path 3 check



Path 4 check

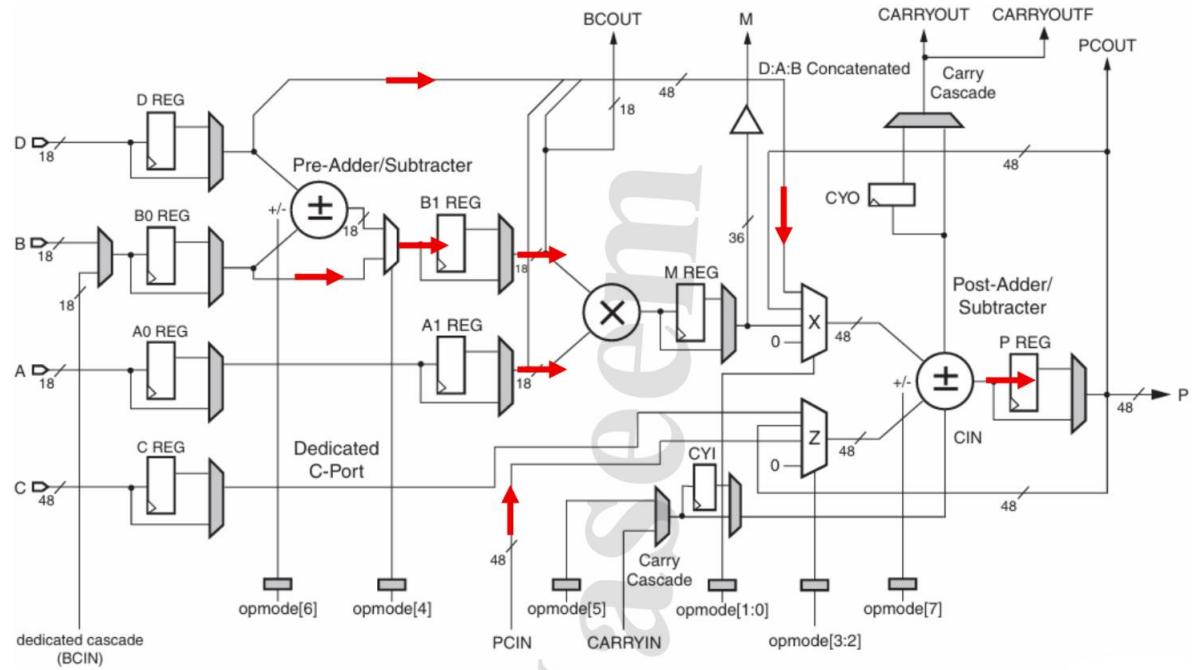
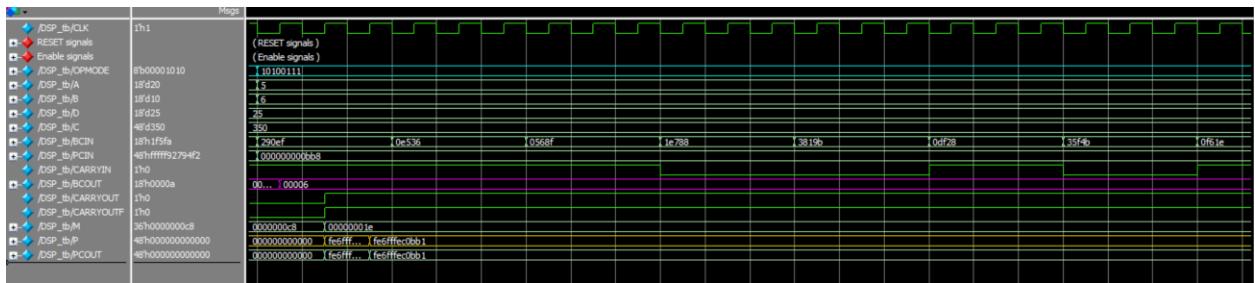


Figure 4: Path 4 Data Flow



Lint check

The screenshot shows the Integrity software interface with the 'Lint Checks' tab selected. The window displays a table of errors with columns for Severity, Status, Check, Alias, and Message. There are three errors listed:

Severity	Status	Check	Alias	Message
Warning	Pending	parameter_...		Same parameter name is used in more than one module. Parameter RSTTYPE, Total count 2, First module: Module DSP, File D:/Digital_Diploma/Project/DSP.v, Line 5, Second module: Module seq_comb, File D:/Digital_Diploma/Project/seq_comb.v, Line 12.
Warning	Pending	multi_ports...		Multiple ports are declared in one line. Module DSP, File D:/Digital_Diploma/Project/DSP.v, Line 12.
Warning	Pending	multi_ports...		Multiple ports are declared in one line. Module seq_comb, File D:/Digital_Diploma/Project/seq_comb.v, Line 4.

Below the table, a message bar indicates 'Total : 3 Selected : 0'. The bottom status bar shows the path 'D:/Digital_Diploma/Project/DSP.v [DSP]'.

Constrain file

The screenshot shows the Integrity software interface with the 'Constraints Editor' window open. The window displays a text-based constraint file with numerous lines of code defining pin properties for a Basys3 rev B board. The code includes sections for clock signals, switches, LEDs, and segment displays, using the 'set_property' command to map pins to specific I/O standards (IOSTANDARD LVCMS33) and port ranges (e.g., sw[0] to sw[15] for switches, led[0] to led[15] for LEDs).

```
1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6 ## Clock signal
7 set_property -dict { PACKAGE_PIN W5 IOSTANDARD LVCMS33 } [get_ports CLK]
8 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports CLK]
9
10
11 ## Switches
12 #set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMS33 } [get_ports {sw[0]}]
13 #set_property -dict { PACKAGE_PIN V16 IOSTANDARD LVCMS33 } [get_ports {sw[1]}]
14 #set_property -dict { PACKAGE_PIN W16 IOSTANDARD LVCMS33 } [get_ports {sw[2]}]
15 #set_property -dict { PACKAGE_PIN W17 IOSTANDARD LVCMS33 } [get_ports {sw[3]}]
16 #set_property -dict { PACKAGE_PIN W15 IOSTANDARD LVCMS33 } [get_ports {sw[4]}]
17 #set_property -dict { PACKAGE_PIN V15 IOSTANDARD LVCMS33 } [get_ports {sw[5]}]
18 #set_property -dict { PACKAGE_PIN W14 IOSTANDARD LVCMS33 } [get_ports {sw[6]}]
19 #set_property -dict { PACKAGE_PIN W13 IOSTANDARD LVCMS33 } [get_ports {sw[7]}]
20 #set_property -dict { PACKAGE_PIN V2 IOSTANDARD LVCMS33 } [get_ports {sw[8]}]
21 #set_property -dict { PACKAGE_PIN T3 IOSTANDARD LVCMS33 } [get_ports {sw[9]}]
22 #set_property -dict { PACKAGE_PIN T2 IOSTANDARD LVCMS33 } [get_ports {sw[10]}]
23 #set_property -dict { PACKAGE_PIN R3 IOSTANDARD LVCMS33 } [get_ports {sw[11]}]
24 #set_property -dict { PACKAGE_PIN W2 IOSTANDARD LVCMS33 } [get_ports {sw[12]}]
25 #set_property -dict { PACKAGE_PIN U1 IOSTANDARD LVCMS33 } [get_ports {sw[13]}]
26 #set_property -dict { PACKAGE_PIN T1 IOSTANDARD LVCMS33 } [get_ports {sw[14]}]
27 #set_property -dict { PACKAGE_PIN R2 IOSTANDARD LVCMS33 } [get_ports {sw[15]}]
28
29
30 ## LEDs
31 #set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMS33 } [get_ports {led[0]}]
32 #set_property -dict { PACKAGE_PIN E19 IOSTANDARD LVCMS33 } [get_ports {led[1]}]
33 #set_property -dict { PACKAGE_PIN V19 IOSTANDARD LVCMS33 } [get_ports {led[2]}]
34 #set_property -dict { PACKAGE_PIN W18 IOSTANDARD LVCMS33 } [get_ports {led[3]}]
35 #set_property -dict { PACKAGE_PIN U15 IOSTANDARD LVCMS33 } [get_ports {led[4]}]
36 #set_property -dict { PACKAGE_PIN U14 IOSTANDARD LVCMS33 } [get_ports {led[5]}]
37 #set_property -dict { PACKAGE_PIN P14 IOSTANDARD LVCMS33 } [get_ports {led[6]}]
38 #set_property -dict { PACKAGE_PIN V14 IOSTANDARD LVCMS33 } [get_ports {led[7]}]
39 #set_property -dict { PACKAGE_PIN V13 IOSTANDARD LVCMS33 } [get_ports {led[8]}]
40 #set_property -dict { PACKAGE_PIN V3 IOSTANDARD LVCMS33 } [get_ports {led[9]}]
41 #set_property -dict { PACKAGE_PIN W3 IOSTANDARD LVCMS33 } [get_ports {led[10]}]
42 #set_property -dict { PACKAGE_PIN U3 IOSTANDARD LVCMS33 } [get_ports {led[11]}]
43 #set_property -dict { PACKAGE_PIN P3 IOSTANDARD LVCMS33 } [get_ports {led[12]}]
44 #set_property -dict { PACKAGE_PIN N3 IOSTANDARD LVCMS33 } [get_ports {led[13]}]
45 #set_property -dict { PACKAGE_PIN P1 IOSTANDARD LVCMS33 } [get_ports {led[14]}]
46 #set_property -dict { PACKAGE_PIN L1 IOSTANDARD LVCMS33 } [get_ports {led[15]}]
47
48
49 ##7 Segment Display
50 #set_property -dict { PACKAGE_PIN W7 IOSTANDARD LVCMS33 } [get_ports {seg[0]}]
51 #set_property -dict { PACKAGE_PIN W6 IOSTANDARD LVCMS33 } [get_ports {seg[1]}]
52 #set_property -dict { PACKAGE_PIN U8 IOSTANDARD LVCMS33 } [get_ports {seg[2]}]
53 #set_property -dict { PACKAGE_PIN V8 IOSTANDARD LVCMS33 } [get_ports {seg[3]}]
54 #set_property -dict { PACKAGE_PIN U5 IOSTANDARD LVCMS33 } [get_ports {seg[4]}]
55 #set_property -dict { PACKAGE_PIN V5 IOSTANDARD LVCMS33 } [get_ports {seg[5]}]
56 #set_property -dict { PACKAGE_PIN U7 IOSTANDARD LVCMS33 } [get_ports {seg[6]}]
```

```

1
2  #set_property -dict { PACKAGE_PIN V7    IOSTANDARD LVC MOS33 } [get_ports dp]
3
4  #set_property -dict { PACKAGE_PIN U2    IOSTANDARD LVC MOS33 } [get_ports {an[0]}]
5  #set_property -dict { PACKAGE_PIN U4    IOSTANDARD LVC MOS33 } [get_ports {an[1]}]
6  #set_property -dict { PACKAGE_PIN V4    IOSTANDARD LVC MOS33 } [get_ports {an[2]}]
7  #set_property -dict { PACKAGE_PIN W4    IOSTANDARD LVC MOS33 } [get_ports {an[3]}]
8
9
10 ##Buttons
11 #set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVC MOS33 } [get_ports rst]
12 #set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVC MOS33 } [get_ports btnU]
13 #set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVC MOS33 } [get_ports btnL]
14 #set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVC MOS33 } [get_ports btnR]
15 #set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVC MOS33 } [get_ports btnD]
16
17
18 ##Pmod Header JA
19 #set_property -dict { PACKAGE_PIN J1    IOSTANDARD LVC MOS33 } [get_ports {JA[0]}];#Sch name = JA1
20 #set_property -dict { PACKAGE_PIN L2    IOSTANDARD LVC MOS33 } [get_ports {JA[1]}];#Sch name = JA2
21 #set_property -dict { PACKAGE_PIN J2    IOSTANDARD LVC MOS33 } [get_ports {JA[2]}];#Sch name = JA3
22 #set_property -dict { PACKAGE_PIN G2    IOSTANDARD LVC MOS33 } [get_ports {JA[3]}];#Sch name = JA4
23 #set_property -dict { PACKAGE_PIN H1    IOSTANDARD LVC MOS33 } [get_ports {JA[4]}];#Sch name = JA7
24 #set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVC MOS33 } [get_ports {JA[5]}];#Sch name = JA8
25 #set_property -dict { PACKAGE_PIN H2    IOSTANDARD LVC MOS33 } [get_ports {JA[6]}];#Sch name = JA9
26 #set_property -dict { PACKAGE_PIN G3    IOSTANDARD LVC MOS33 } [get_ports {JA[7]}];#Sch name = JA10
27
28 ##Pmod Header JB
29 #set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVC MOS33 } [get_ports {JB[0]}];#Sch name = JB1
30 #set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVC MOS33 } [get_ports {JB[1]}];#Sch name = JB2
31 #set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVC MOS33 } [get_ports {JB[2]}];#Sch name = JB3
32 #set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVC MOS33 } [get_ports {JB[3]}];#Sch name = JB4
33 #set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVC MOS33 } [get_ports {JB[4]}];#Sch name = JB7
34 #set_property -dict { PACKAGE_PIN A17   IOSTANDARD LVC MOS33 } [get_ports {JB[5]}];#Sch name = JB8
35 #set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVC MOS33 } [get_ports {JB[6]}];#Sch name = JB9
36 #set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVC MOS33 } [get_ports {JB[7]}];#Sch name = JB10
37
38 ##Pmod Header JC
39 #set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVC MOS33 } [get_ports {JC[0]}];#Sch name = JC1
40 #set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVC MOS33 } [get_ports {JC[1]}];#Sch name = JC2
41 #set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVC MOS33 } [get_ports {JC[2]}];#Sch name = JC3
42 #set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVC MOS33 } [get_ports {JC[3]}];#Sch name = JC4
43 #set_property -dict { PACKAGE_PIN L17   IOSTANDARD LVC MOS33 } [get_ports {JC[4]}];#Sch name = JC7
44 #set_property -dict { PACKAGE_PIN M19   IOSTANDARD LVC MOS33 } [get_ports {JC[5]}];#Sch name = JC8
45 #set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVC MOS33 } [get_ports {JC[6]}];#Sch name = JC9
46 #set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVC MOS33 } [get_ports {JC[7]}];#Sch name = JC10
47
48 ##Pmod Header JXADC
49 #set_property -dict { PACKAGE_PIN J3    IOSTANDARD LVC MOS33 } [get_ports {JXADC[0]}];#Sch name = XA1_P
50 #set_property -dict { PACKAGE_PIN L3    IOSTANDARD LVC MOS33 } [get_ports {JXADC[1]}];#Sch name = XA2_P
51 #set_property -dict { PACKAGE_PIN M2    IOSTANDARD LVC MOS33 } [get_ports {JXADC[2]}];#Sch name = XA3_P
52 #set_property -dict { PACKAGE_PIN N2    IOSTANDARD LVC MOS33 } [get_ports {JXADC[3]}];#Sch name = XA4_P
53 #set_property -dict { PACKAGE_PIN K3    IOSTANDARD LVC MOS33 } [get_ports {JXADC[4]}];#Sch name = XA1_N
54 #set_property -dict { PACKAGE_PIN M3    IOSTANDARD LVC MOS33 } [get_ports {JXADC[5]}];#Sch name = XA2_N
55 #set_property -dict { PACKAGE_PIN M1    IOSTANDARD LVC MOS33 } [get_ports {JXADC[6]}];#Sch name = XA3_N
56 #set_property -dict { PACKAGE_PIN N1    IOSTANDARD LVC MOS33 } [get_ports {JXADC[7]}];#Sch name = XA4_N
57
58

```

```

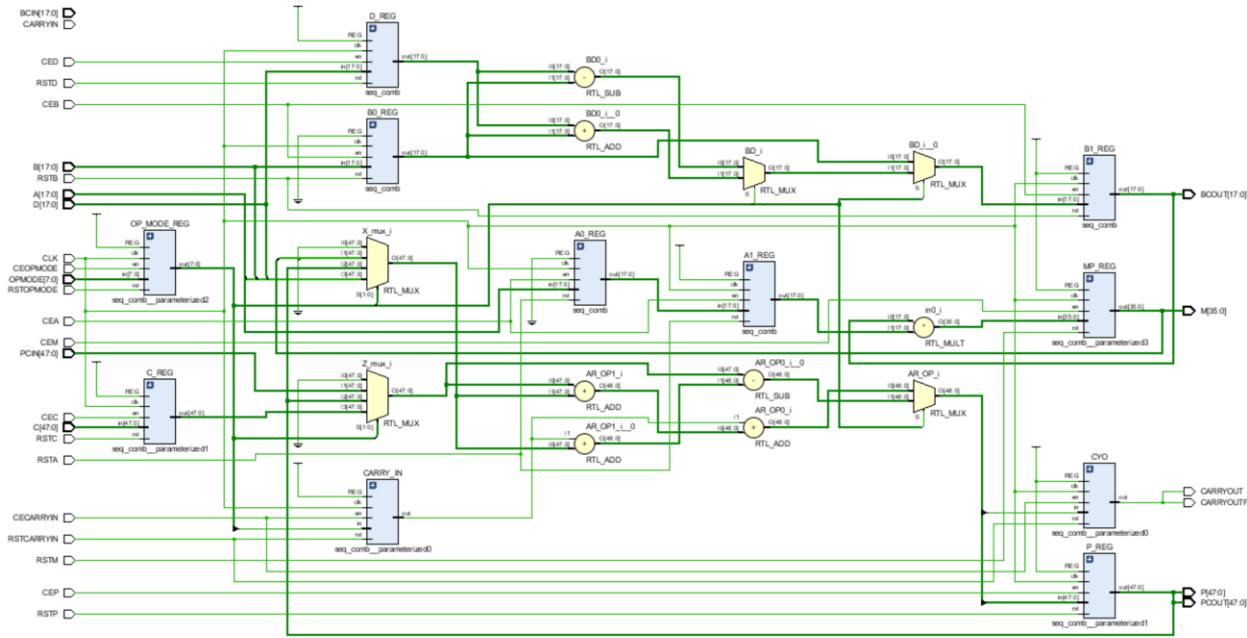
1
2 ##VGA Connector
3 #set_property -dict { PACKAGE_PIN G19     IOSTANDARD LVC莫斯33 } [get_ports {vgaRed[0]}]
4 #set_property -dict { PACKAGE_PIN H19     IOSTANDARD LVC莫斯33 } [get_ports {vgaRed[1]}]
5 #set_property -dict { PACKAGE_PIN J19     IOSTANDARD LVC莫斯33 } [get_ports {vgaRed[2]}]
6 #set_property -dict { PACKAGE_PIN N19     IOSTANDARD LVC莫斯33 } [get_ports {vgaRed[3]}]
7 #set_property -dict { PACKAGE_PIN N18     IOSTANDARD LVC莫斯33 } [get_ports {vgaBlue[0]}]
8 #set_property -dict { PACKAGE_PIN L18    IOSTANDARD LVC莫斯33 } [get_ports {vgaBlue[1]}]
9 #set_property -dict { PACKAGE_PIN K18    IOSTANDARD LVC莫斯33 } [get_ports {vgaBlue[2]}]
10 #set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVC莫斯33 } [get_ports {vgaBlue[3]}]
11 #set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVC莫斯33 } [get_ports {vgaGreen[0]}]
12 #set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVC莫斯33 } [get_ports {vgaGreen[1]}]
13 #set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVC莫斯33 } [get_ports {vgaGreen[2]}]
14 #set_property -dict { PACKAGE_PIN D17  IOSTANDARD LVC莫斯33 } [get_ports {vgaGreen[3]}]
15 #set_property -dict { PACKAGE_PIN P19  IOSTANDARD LVC莫斯33 } [get_ports Hsync]
16 #set_property -dict { PACKAGE_PIN R19  IOSTANDARD LVC莫斯33 } [get_ports Vsync]
17
18
19 ##USB-RS232 Interface
20 #set_property -dict { PACKAGE_PIN B18  IOSTANDARD LVC莫斯33 } [get_ports RsRx]
21 #set_property -dict { PACKAGE_PIN A18  IOSTANDARD LVC莫斯33 } [get_ports RsTx]
22
23
24 ##USB HID (PS/2)
25 #set_property -dict { PACKAGE_PIN C17  IOSTANDARD LVC莫斯33  PULLUP true } [get_ports PS2Clk]
26 #set_property -dict { PACKAGE_PIN B17  IOSTANDARD LVC莫斯33  PULLUP true } [get_ports PS2Data]
27
28
29 ##Quad SPI Flash
30 ##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the
31 ##STARTUPE2 primitive.
32 #set_property -dict { PACKAGE_PIN D18  IOSTANDARD LVC莫斯33 } [get_ports {QspiDB[0]}]
33 #set_property -dict { PACKAGE_PIN D19  IOSTANDARD LVC莫斯33 } [get_ports {QspiDB[1]}]
34 #set_property -dict { PACKAGE_PIN G18  IOSTANDARD LVC莫斯33 } [get_ports {QspiDB[2]}]
35 #set_property -dict { PACKAGE_PIN F18  IOSTANDARD LVC莫斯33 } [get_ports {QspiDB[3]}]
36 #set_property -dict { PACKAGE_PIN K19  IOSTANDARD LVC莫斯33 } [get_ports QspiCSn]
37
38
39 ## Configuration options, can be used for all designs
40 set_property CONFIG_VOLTAGE 3.3 [current_design]
41 set_property CFGBVS VCCO [current_design]
42
43 ## SPI configuration mode options for QSPI boot, can be used for all designs
44 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
45 set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
46 set_property CONFIG_MODE SPIx4 [current_design]

```

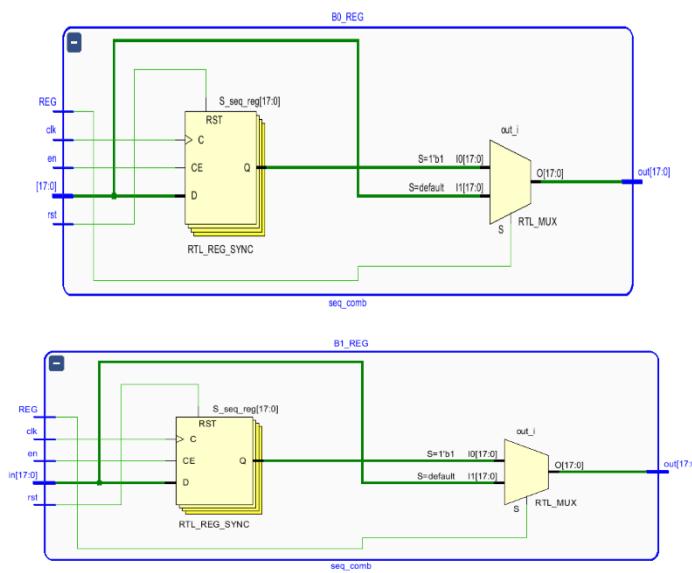
Synthesis tool results

Elaboration

schematic



Zooming inside B0REG, B1REG to check the seq_comb is instantiated right.

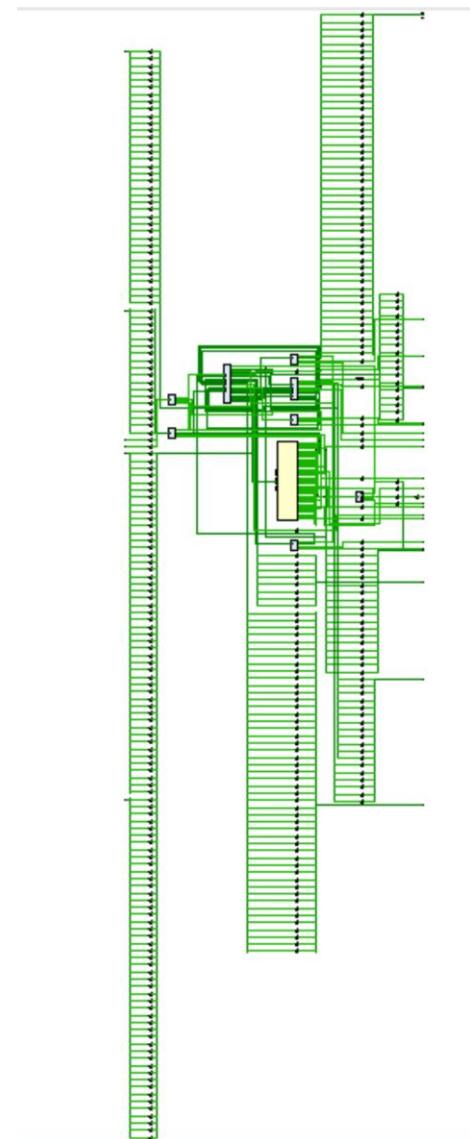


Messages

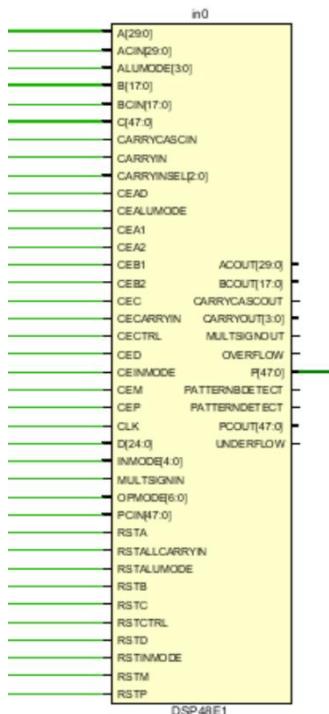
- ➊ [Device 21-403] Loading part xc7a200tffg1156-3
- ➋ [Project 1-570] Preparing netlist for logic optimization
- ➌ Processing XDC Constraints (6 more like this)
- ➍ [Project 1-236] Implementation specific constraints were found while reading constraint file [D:/Digital_Diploma/Project/DSP_cons.xdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [XII/DSP_propimpl.xdc].
Resolution: To avoid this warning, move constraints listed in [XII/DSP_propimpl.xdc] to another XDC file and exclude this new file from synthesis with the used_in_synthesis property (File Properties dialog in GUI) and re-run elaboration/synthesis.
- ➎ [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
- ➏ [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

Synthesis

Schematic



Although it may not appear remarkable at first, a closer examination of the yellow block reveals the presence of our DSP!



Message

Tcl Console Messages Log Reports Design Runs Timing Utilization Debug

Q X Hide All

Manage Message Suppression

Warning (57) Info (60) Status (22)

- Parsing XDC File
- [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
- [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.
- [Timing 38-35] Done setting XDC timing constraints.
- [Timing 38-91] UpdateTimingParams: Speed grade: -3, Delay Type: min_max.
- [Timing 38-191] Multithreading enabled for timing update using a maximum of 2 CPUs

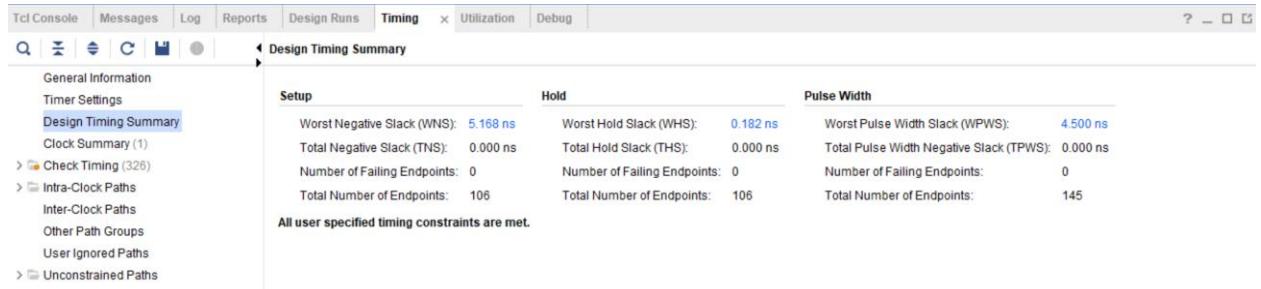
Utilities report

Tcl Console Messages Log Reports Design Runs Utilization Debug

Q X % Hierarchy

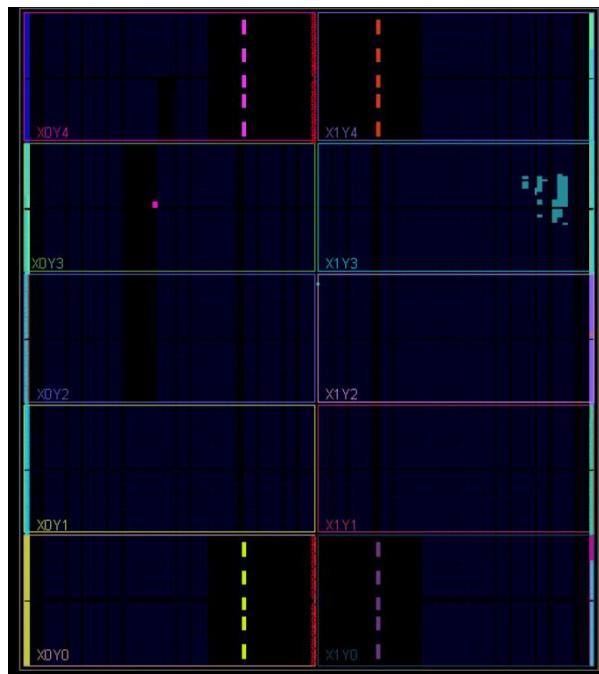
Name	Slice LUTs (134600)	Slice Registers (269200)	DSP s (740)	Bonded IOB (500)	BUFGCTRL (32)
N DSP	230	143	1	327	1
A1_REG (seq_comb)	0	1	0	0	0
B1_REG (seq_comb_0)	0	18	0	0	0
C_REG (seq_comb_...)	0	48	0	0	0
CARRY_IN (seq_comb...)	1	1	0	0	0
CY0 (seq_comb_par...)	0	1	0	0	0
D_REG (seq_comb_2)	0	18	0	0	0
OP_MODE_REG (seq...)	228	8	0	0	0
P_REG (seq_comb...)	0	48	0	0	0

Timing report

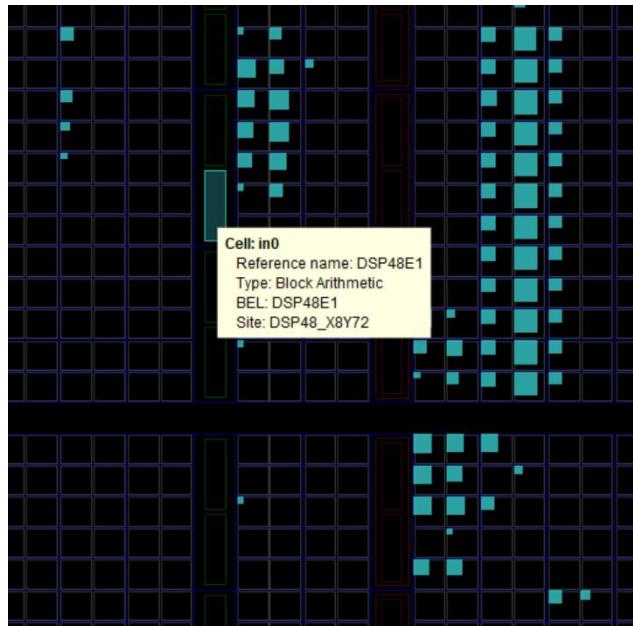


Implementation

Schematic



Zooming at the DSP block



Messages

A screenshot of the 'Messages' tab in the Tcl Console. It displays several log messages: '[Device 21-403] Loading part xc200ffg1156-3', '[Project 1-570] Preparing netlist for logic optimization', and '[Timing 38-478] Restoring timing data from binary archive.' There are checkboxes for 'Warning', 'Info', and 'Status' messages, all of which are checked.

Utilities report

A screenshot of the 'Utilization' report. On the left, there is a hierarchical tree view showing categories like 'Slice Logic', 'Slice Registers', 'Slice Logic Distribution', and 'LUT Flip Flop Pairs'. The main table lists resources for different components. The columns include: Name, Slice LUT's (133800), Slice Registers (267600), Slice (33450), LUT as Logic (133800), LUT Flip Flop Pairs (133800), DSPs (740), Bonded IOB (500), and BUFGCTRL (32). The 'DSP' row is expanded, showing sub-components: A1_REG, B1_REG, C_REG, CARRY_IN, CYO, D_REG, OP_MODE_REG, and P_REG, each with their respective resource counts.

Timing report

Tcl Console Messages Log Reports Design Runs Power DRC Methodology **Timing** Utilization

Design Timing Summary

General Information

Timer Settings

Design Timing Summary

Clock Summary (1)

> Check Timing (326)

> Intra-Clock Paths

Inter-Clock Paths

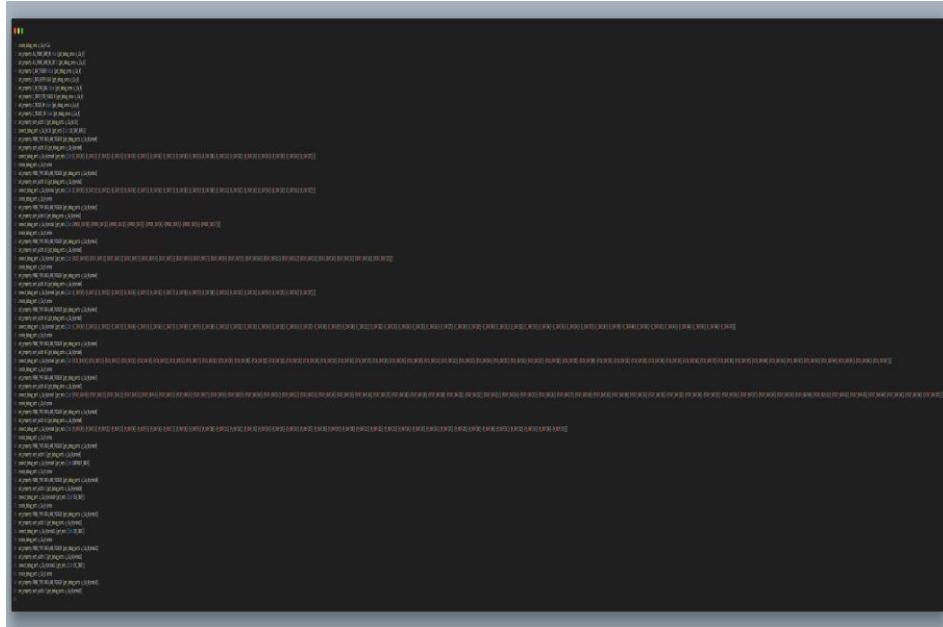
Other Path Groups

Setup Hold Pulse Width

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 3.962 ns	Worst Hold Slack (WHS): 0.273 ns	Worst Pulse Width Negative Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWNS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 107	Total Number of Endpoints: 107	Total Number of Endpoints: 146

All user specified timing constraints are met.

Debug cores



```
1
2 connect_debug_port u_ila_0/probe13 [get_nets [list CECARRYIN_IBUF]]
3 create_debug_port u_ila_0 probe
4 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe14]
5 set_property port_width 1 [get_debug_ports u_ila_0/probe14]
6 connect_debug_port u_ila_0/probe14 [get_nets [list CED_IBUF]]
7 create_debug_port u_ila_0 probe
8 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe15]
9 set_property port_width 1 [get_debug_ports u_ila_0/probe15]
10 connect_debug_port u_ila_0/probe15 [get_nets [list CEM_IBUF]]
11 create_debug_port u_ila_0 probe
12 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe16]
13 set_property port_width 1 [get_debug_ports u_ila_0/probe16]
14 connect_debug_port u_ila_0/probe16 [get_nets [list CEOPMODE_IBUF]]
15 create_debug_port u_ila_0 probe
16 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe17]
17 set_property port_width 1 [get_debug_ports u_ila_0/probe17]
18 connect_debug_port u_ila_0/probe17 [get_nets [list CEP_IBUF]]
19 create_debug_port u_ila_0 probe
20 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe18]
21 set_property port_width 1 [get_debug_ports u_ila_0/probe18]
22 connect_debug_port u_ila_0/probe18 [get_nets [list CLK_IBUF]]
23 create_debug_port u_ila_0 probe
24 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe19]
25 set_property port_width 1 [get_debug_ports u_ila_0/probe19]
26 connect_debug_port u_ila_0/probe19 [get_nets [list RSTA_IBUF]]
27 create_debug_port u_ila_0 probe
28 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe20]
29 set_property port_width 1 [get_debug_ports u_ila_0/probe20]
30 connect_debug_port u_ila_0/probe20 [get_nets [list RSTB_IBUF]]
31 create_debug_port u_ila_0 probe
32 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe21]
33 set_property port_width 1 [get_debug_ports u_ila_0/probe21]
34 connect_debug_port u_ila_0/probe21 [get_nets [list RSTC_IBUF]]
35 create_debug_port u_ila_0 probe
36 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe22]
37 set_property port_width 1 [get_debug_ports u_ila_0/probe22]
38 connect_debug_port u_ila_0/probe22 [get_nets [list RSTCARRYIN_IBUF]]
39 create_debug_port u_ila_0 probe
40 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe23]
41 set_property port_width 1 [get_debug_ports u_ila_0/probe23]
42 connect_debug_port u_ila_0/probe23 [get_nets [list RSTD_IBUF]]
43 create_debug_port u_ila_0 probe
44 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe24]
45 set_property port_width 1 [get_debug_ports u_ila_0/probe24]
46 connect_debug_port u_ila_0/probe24 [get_nets [list RSTM_IBUF]]
47 create_debug_port u_ila_0 probe
48 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe25]
49 set_property port_width 1 [get_debug_ports u_ila_0/probe25]
50 connect_debug_port u_ila_0/probe25 [get_nets [list RSTOPMODE_IBUF]]
51 create_debug_port u_ila_0 probe
52 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe26]
53 set_property port_width 1 [get_debug_ports u_ila_0/probe26]
54 connect_debug_port u_ila_0/probe26 [get_nets [list RSTP_IBUF]]
55 set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
56 set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
57 set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
58 connect_debug_port dbg_hub/clk [get_nets CLK_IBUF_BUFG]
59
```