

Digital IC design
Project 1
Spartan6 – DSP48A1

Made by: Ahmed Emad Mohamed

Part I: RTL Code

The main module is composed of several sub-modules, which are connected using the instantiation method.

1) Register followed by Multiplexer

```
1  module DFF_with_MUX(D,clk,rst,CLK_EN,out);
2
3      parameter WIDTH = 18;
4      parameter RSTTYPE = "SYNC";
5      parameter sel = 0;
6      input clk,rst,CLK_EN;
7      input [WIDTH - 1: 0]D;
8      output [WIDTH - 1: 0]out;
9      reg [WIDTH - 1: 0]Q;
10
11     generate
12         if (RSTTYPE == "ASYNC") begin
13             always @ (posedge clk or posedge rst) begin
14                 if (rst) begin
15                     Q <= 0;
16                 end else if (CLK_EN) begin
17                     Q <= D;
18                 end
19             end
20         end else begin
21             always @ (posedge clk) begin
22                 if (rst) begin
23                     Q <= 0;
24                 end else if (CLK_EN) begin
25                     Q <= D;
26                 end
27             end
28         end
29         assign out = (sel == 1)? Q : D;
30
31     endgenerate
32
33 endmodule
```

2) Multiplexers (4:1) & (2:1)

```
1 module mux_4_1(in0,in1,in2,in3,sel,out);
2
3     parameter WIDTH = 48;
4     input [WIDTH -1:0]in0;
5     input [WIDTH -1:0]in1;
6     input [WIDTH -1:0]in2;
7     input [WIDTH -1:0]in3;
8     input [1:0]sel;
9     output reg [WIDTH -1:0]out;
10
11    always @(*) begin
12        case (sel)
13            2'b00: out = in0;
14            2'b01: out = in1;
15            2'b10: out = in2;
16            2'b11: out = in3;
17            default: out = 0;
18        endcase
19    end
20 endmodule
```

```
1 module mux_2_1(in0,in1,sel,out);
2
3     parameter WIDTH = 18;
4     input [WIDTH -1:0]in0;
5     input [WIDTH -1:0]in1;
6     input sel;
7     output [WIDTH -1:0]out;
8
9     assign out = (sel == 1)? in1 : in0;
10
11 endmodule
```

3) Pre & Post- Adders-Subtracters

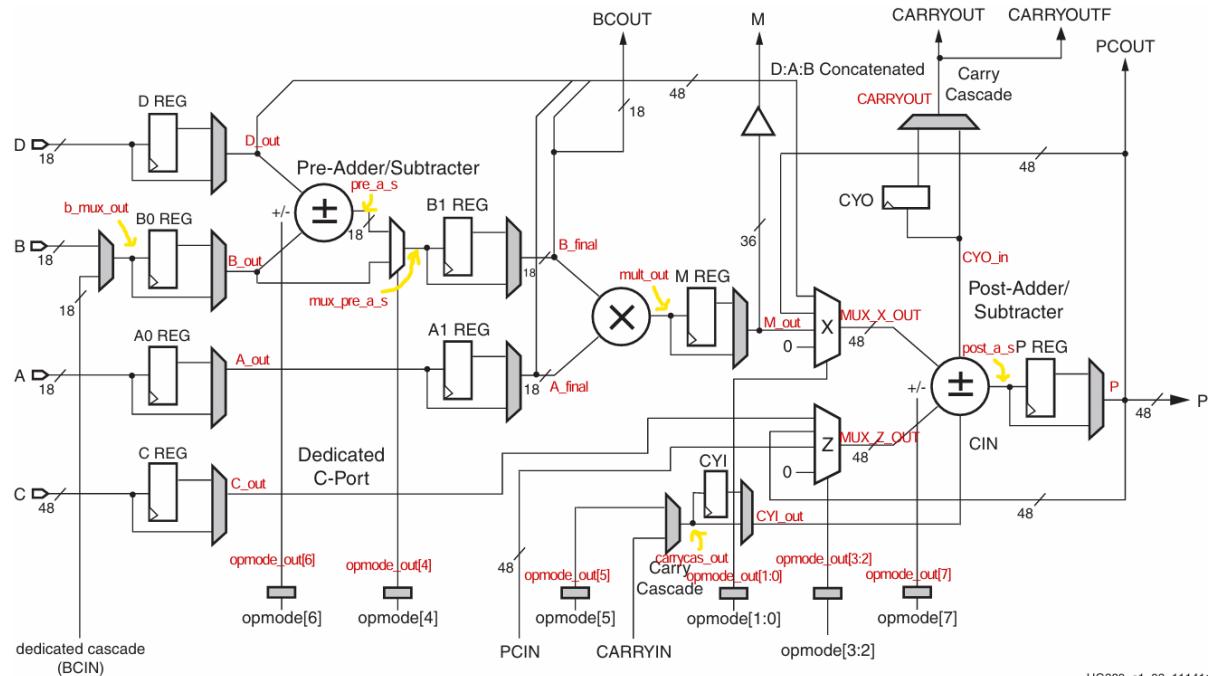
```
1  ✓ module PRE_ADD_SUB(in0,in1,sel,out);
2
3      parameter WIDTH = 18;
4      input [WIDTH -1:0]in0;
5      input [WIDTH -1:0]in1;
6      input sel;
7      output reg [WIDTH -1:0]out;
8
9      ✓ always @(*) begin
10     ✓     if (sel) begin
11         |     out = in0 - in1;
12     ✓     end else begin
13         |     out = in0 + in1;
14     end
15   end
16
17 endmodule
```

```
1  ✓ module POST_ADD_SUB(in0,in1,in2,sel,out,CARRYOUT);
2
3      parameter WIDTH = 48;
4      input [WIDTH -1:0]in0;
5      input [WIDTH -1:0]in1;
6      input sel,in2;
7      output reg [WIDTH -1:0]out;
8      output reg CARRYOUT;
9
10     ✓ always @(*) begin
11     ✓     if (sel) begin
12         |     {CARRYOUT,out} = in0 - (in1 + in2);
13     ✓     end else begin
14         |     {CARRYOUT,out} = in0 + in1 + in2;
15     end
16   end
17
18 endmodule
```

4) Multiplier

```
1 module MULT(in0,in1,out);
2
3     parameter WIDTH = 18;
4     localparam WIDTH_OUT = 2 * WIDTH;
5     input [WIDTH -1:0]in0;
6     input [WIDTH -1:0]in1;
7     output [WIDTH_OUT -1:0]out;
8
9     assign out = in0 * in1;
10
11 endmodule
```

5) Some indices used in the top module coding



UG389 c1 03 11141

6) DSP48A1 Top Module

```

1  module DSP48A1(A,B,D,C,clk,CARRYIN,OPMODE,BCIN,
2  RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,
3  CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE,PCIN,
4  BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF
5  );
6
7  parameter A0REG = 0; parameter A1REG = 1;
8  parameter B0REG = 0; parameter B1REG = 1;
9  parameter CREG = 1; parameter DREG = 1;
10 parameter MREG = 1; parameter PREG = 1;
11 parameter CARRYINREG = 1; parameter CARRYOUTREG = 1;
12 parameter OPMODEREG = 1; parameter CARRYINSEL = "OPMODE5";
13 parameter B_INPUT = "DIRECT"; parameter RSTTYPE = "SYNC";
14
15 localparam CARRYINSEL_condition = (CARRYINSEL == "OPMODE5")? 1'b1:1'b0;
16 localparam B_INPUT_condition = (B_INPUT == "BCIN")? 1'b1:1'b0;
17
18 input [17:0]A,B,D,BCIN;
19 input [47:0]C,PCIN;
20 input [7:0]OPMODE;
21 input clk,CARRYIN,
22 RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,
23 CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE;
24 output [17:0]BCOUT;
25 output [47:0]PCOUT,P;
26 output [35:0]M;
27 output CARRYOUT,CARRYOUTF;
28
29 wire [17:0]A_out,B_out,D_out,b_mux_out,pre_a_s,mux_pre_a_s,B_final,A_final;
30 wire [47:0]C_out,MUX_X_OUT,MUX_Z_OUT,M_before_MUX;
31 wire [47:0] post_a_s;
32 wire [35:0]mult_out,M_out;
33 wire [7:0]opmode_out;

```

```

35 //////////////////////////////////////////////////////////////////
36 // OPMODE REGISTERS
37 // DFF_with_MUX(D,clk,rst,CLK_EN,out)
38 //////////////////////////////////////////////////////////////////
39
40 DFF_with_MUX #( .WIDTH(8) , .RSTTYPE(RSTTYPE) , .sel(OPMODEREG) )
41 OPMODE_value(OPMODE,clk,RSTOPMODE,CEOPMODE,opmode_out);
42
43 //////////////////////////////////////////////////////////////////
44 // FIRST STAGE
45 // DFF_with_MUX(D,clk,rst,CLK_EN,out) -- mux_2_1(in0,in1,sel,out)
46 //////////////////////////////////////////////////////////////////
47 DFF_with_MUX #( .WIDTH(18) , .RSTTYPE(RSTTYPE) , .sel(DREG) ) D_REG(D,clk,RSTD,CED,D_out);
48 mux_2_1 #( .WIDTH(18) ) MUX_B(B,BCIN,B_INPUT_condition,b_mux_out);
49 DFF_with_MUX #( .WIDTH(18) , .RSTTYPE(RSTTYPE) , .sel(B0REG) ) B0_REG(b_mux_out,clk,RSTB,CEB,B_out);
50 DFF_with_MUX #( .WIDTH(18) , .RSTTYPE(RSTTYPE) , .sel(A0REG) ) A0_REG(A,clk,RSTA,CEA,A_out);
51 DFF_with_MUX #( .WIDTH(48) , .RSTTYPE(RSTTYPE) , .sel(CREG) ) C_REG(C,clk,RSTC,CEC,C_out);
52
53 //////////////////////////////////////////////////////////////////
54 // PRE_ADDER_SUBTRACTER & SECOND STAGE & Multiplier
55 // PRE_ADD_SUB(in0,in1,sel,out) -- mux_2_1(in0,in1,sel,out) -- MULT(in0,in1,out)
56 //////////////////////////////////////////////////////////////////
57 PRE_ADD_SUB #( .WIDTH(18) ) pre_add_sub(D_out,B_out,opmode_out[6],pre_a_s);
58 mux_2_1 #( .WIDTH(18) ) bypass_mux(B_out,pre_a_s,opmode_out[4],mux_pre_a_s);
59 DFF_with_MUX #( .WIDTH(18) , .RSTTYPE(RSTTYPE) , .sel(B1REG) ) B1_REG(mux_pre_a_s,clk,RSTB,CEB,B_final);
60 assign BCOUT = B_final;
61 DFF_with_MUX #( .WIDTH(18) , .RSTTYPE(RSTTYPE) , .sel(A1REG) ) A1_REG(A_out,clk,RSTA,CEA,A_final);
62 MULT #( .WIDTH(18) ) multi(B_final,A_final,mult_out);
63

```

```

64 //////////////////////////////////////////////////////////////////
65 // THIRD STAGE
66 // mux_4_1(in0,in1,in2,in3,sel,out) -- carry_cascade_MUX(in0,in1,sel,out)
67 //////////////////////////////////////////////////////////////////
68
69 DFF_with_MUX #( .WIDTH(36) , .RSTTYPE(RSTTYPE) , .sel(MREG) ) M_REG(mult_out,clk,RSTM,CEM,M_out);
70 assign M = M_out;
71 mux_4_1 #( .WIDTH(48) ) MUX_X(48'd0,{12'd0,M_out},P,[D_out[11:0],A_final,B_final],opmode_out[1:0],MUX_X_OUT);
72 mux_2_1 #( .WIDTH(1) ) CARRYCAS_MUX(CARRYIN,opmode_out[5],CARRYINSEL,carrycas_out);
73 DFF_with_MUX #( .WIDTH(1) , .RSTTYPE(RSTTYPE) , .sel(CARRYINREG) ) CYI(carrycas_out,clk,RSTCARRYIN,CECARRYIN,CYI_out);
74 mux_4_1 #( .WIDTH(48) ) MUX_Z(48'd0,PCIN,P,C_out,opmode_out[3:2],MUX_Z_OUT);
75
76 //////////////////////////////////////////////////////////////////
77 // FOURTH (Last) STAGE & POST_ADDER_SUBTRACTER
78 // POST_ADD_SUB(in0,in1,in2,sel,out)
79 //////////////////////////////////////////////////////////////////
80
81 POST_ADD_SUB #( .WIDTH(48) ) post_add_sub(MUX_Z_OUT,MUX_X_OUT,CYI_out,opmode_out[7],post_a_s,CYO_in);
82 DFF_with_MUX #( .WIDTH(1) , .RSTTYPE(RSTTYPE) , .sel(CARRYOUTREG) ) CYO(CYO_in,clk,RSTCARRYIN,CECARRYIN,CARRYOUT);
83 assign CARRYOUTF = CARRYOUT;
84 DFF_with_MUX #( .WIDTH(48) , .RSTTYPE(RSTTYPE) , .sel(PREG) ) P_REG(post_a_s,clk,RSTP,CEP,P);
85 assign PCOUT = P;
86
87 endmodule

```

Part II: Testbench Code

```

1  module DSP48A1_tb();
2
3
4      reg [17:0]A_tb,B_tb,D_tb,BCIN_tb,BCOUT_expected;
5      reg [47:0]C_tb,PCIN_tb,PCOUT_expected,P_expected;
6      reg [7:0]OPMODE_tb;
7      reg [35:0]M_expected;
8      reg clk,CARRYIN_tb,RSTA_tb,RSTB_tb,RSTM_tb,RSTP_tb,RSTC_tb,
9      RSTD_tb,RSTCARRYIN_tb,RSTOPMODE_tb,CEA_tb,CEB_tb,CEM_tb,CEP_tb,CEC_tb,
10     CED_tb,CECARRYIN_tb,CEOPMODE_tb,CARRYOUT_expected,CARRYOUTF_expected;
11     wire [17:0]BCOUT;
12     wire [47:0]PCOUT,P;
13     wire [35:0]M;
14     wire CARRYOUT,CARRYOUTF;
15
16     DSP48A1 #(A0REG(0),A1REG(1),B0REG(0),B1REG(1),CREG(1),DREG(1),
17     .MREG(1),.PREG(1),.CARRYINREG(1),.CARRYOUTREG(1),.OPMODEREG(1),
18     .CARRYINSEL("OPMODE5"),.B_INPUT("DIRECT"),.RSTTYPE("SYNC")) TEST(A_tb,B_tb,D_tb,c_tb,clk,CARRYIN_tb,OPMODE_tb,BCIN_tb,
19     RSTA_tb,RSTB_tb,RSTM_tb,RSTP_tb,RSTC_tb,RSTD_tb,RSTCARRYIN_tb,
20     RSTOPMODE_tb,CEA_tb,CEB_tb,CEM_tb,CEP_tb,CEC_tb,CED_tb,CECARRYIN_tb,
21     CEOPMODE_tb,PCIN_tb,BCOUT,PCOUT,P,
22     M,CARRYOUT,CARRYOUTF);
23
24 ///////////////////////////////////////////////////////////////////
25 // 2.1 Verify Reset Operation
26 ///////////////////////////////////////////////////////////////////
27
28     initial begin
29         clk =0;
30         forever begin
31             #1 clk = ~clk;
32         end
33     end

```

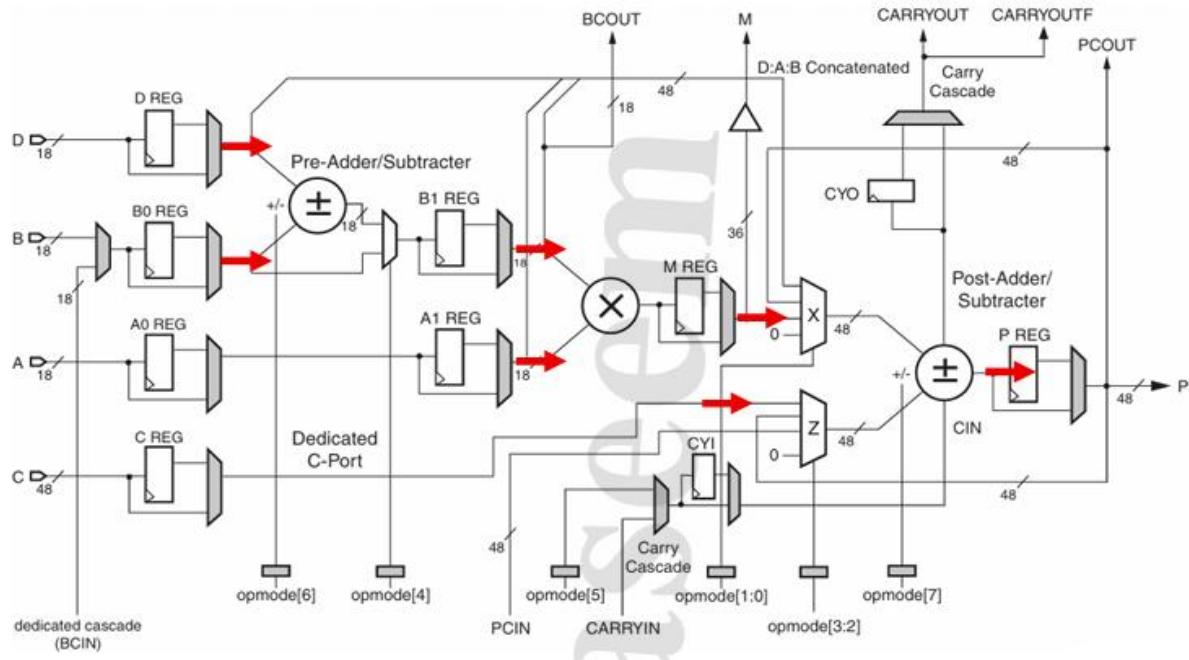
```

35     initial begin
36         RSTA_tb = 1;
37         RSTB_tb = 1;
38         RSTM_tb = 1;
39         RSTP_tb = 1;
40         RSTC_tb = 1;
41         RSTD_tb = 1;
42         RSTCARRYIN_tb = 1;
43         RSTOPMODE_tb = 1;
44         A_tb = $random;
45         B_tb = $random;
46         D_tb = $random;
47         C_tb = $random;
48         CARRYIN_tb = $random;
49         OPMODE_tb = $random;
50         BCIN_tb = $random;
51         CEA_tb = $random;
52         CEB_tb = $random;
53         CEM_tb = $random;
54         CEP_tb = $random;
55         CEC_tb = $random;
56         CED_tb = $random;
57         CECARRYIN_tb = $random;
58         CEOPMODE_tb = $random;
59         PCIN_tb = $random;
60         @(negedge clk);
61         if (BCOUT || PCOUT || P || M || CARRYOUT || CARRYOUTF) begin
62             $display("ERROR, Reset Verification is incorrect");
63             $stop;
64         end

```

```
66      RSTA_tb = 0;  
67      RSTB_tb = 0;  
68      RSTM_tb = 0;  
69      RSTP_tb = 0;  
70      RSTC_tb = 0;  
71      RSTD_tb = 0;  
72      RSTCARRYIN_tb = 0;  
73      RSTOPMODE_tb = 0;  
74      CEA_tb = 1;  
75      CEB_tb = 1;  
76      CEM_tb = 1;  
77      CEP_tb = 1;  
78      CEC_tb = 1;  
79      CED_tb = 1;  
80      CECARRYIN_tb = 1;  
81      CEOPMODE_tb = 1;
```

1) Path 1 test

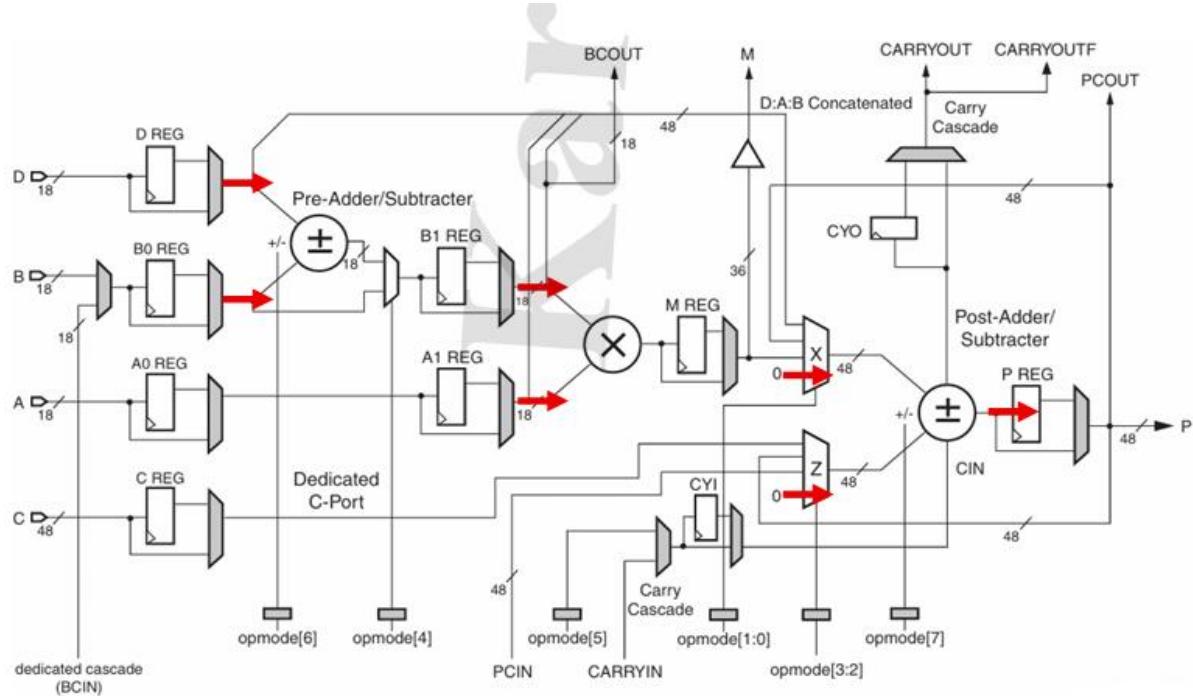


```

83 //////////////////////////////////////////////////////////////////
84 // 2.2 Verify DSP path 1
85 //////////////////////////////////////////////////////////////////
86
87 repeat(2) @(negedge clk);
88 A_tb = 20 ;
89 B_tb = 10 ;
90 D_tb = 25 ;
91 C_tb = 350 ;
92 OPMODE_tb = 8'b11011101;
93 CARRYIN_tb = $random;
94 BCIN_tb = $random;
95 PCIN_tb = $random;
96 BCOUT_expected = 18'hf;
97 M_expected = 36'h12c;
98 P_expected = 48'h32;
99 PCOUT_expected = 48'h32;
100 CARRYOUT_expected = 0;
101 CARRYOUTF_expected = 0;
102 repeat(4) @(negedge clk);
103 if ((BCOUT_expected != BCOUT) || (M_expected != M) || (P_expected != P) || (PCOUT_expected != PCOUT) ||
104 (CARRYOUT_expected != CARRYOUT) || (CARRYOUTF_expected != CARRYOUTF) ) begin
105 $display("ERROR, PATH1 verification is incorrect");
106 $stop;
107 end

```

2) Path 2 test

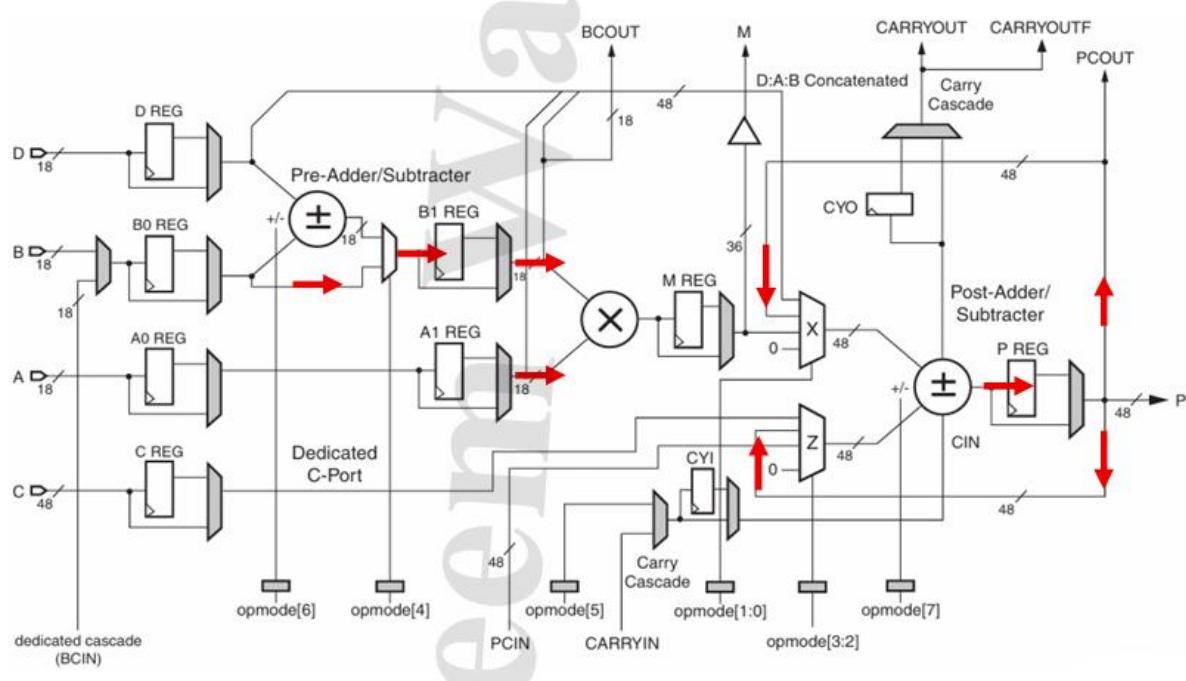


```

109 //////////////////////////////////////////////////////////////////
110 // 2.3 Verify DSP path 2
111 //////////////////////////////////////////////////////////////////
112
113     @(negedge clk);
114     A_tb = 20 ;
115     B_tb = 10 ;
116     D_tb = 25 ;
117     C_tb = 350 ;
118     OPMODE_tb = 8'b00010000;
119     CARRYIN_tb = $random;
120     BCIN_tb = $random;
121     PCIN_tb = $random;
122     BCOUT_expected = 18'h23;
123     M_expected = 36'h2bc;
124     P_expected = 0;
125     PCOUT_expected = 0;
126     CARRYOUT_expected = 0;
127     CARRYOUTF_expected = 0;
128     repeat(3) @(negedge clk);
129     if ((BCOUT_expected != BCOUT) || (M_expected != M) || (P_expected != P) || (PCOUT_expected != PCOUT) ||
130         (CARRYOUT_expected != CARRYOUT) || (CARRYOUTF_expected != CARRYOUTF) ) begin
131         $display("ERROR, PATH2 verification is incorrect");
132         $stop;
133     end

```

3) Path 3 test

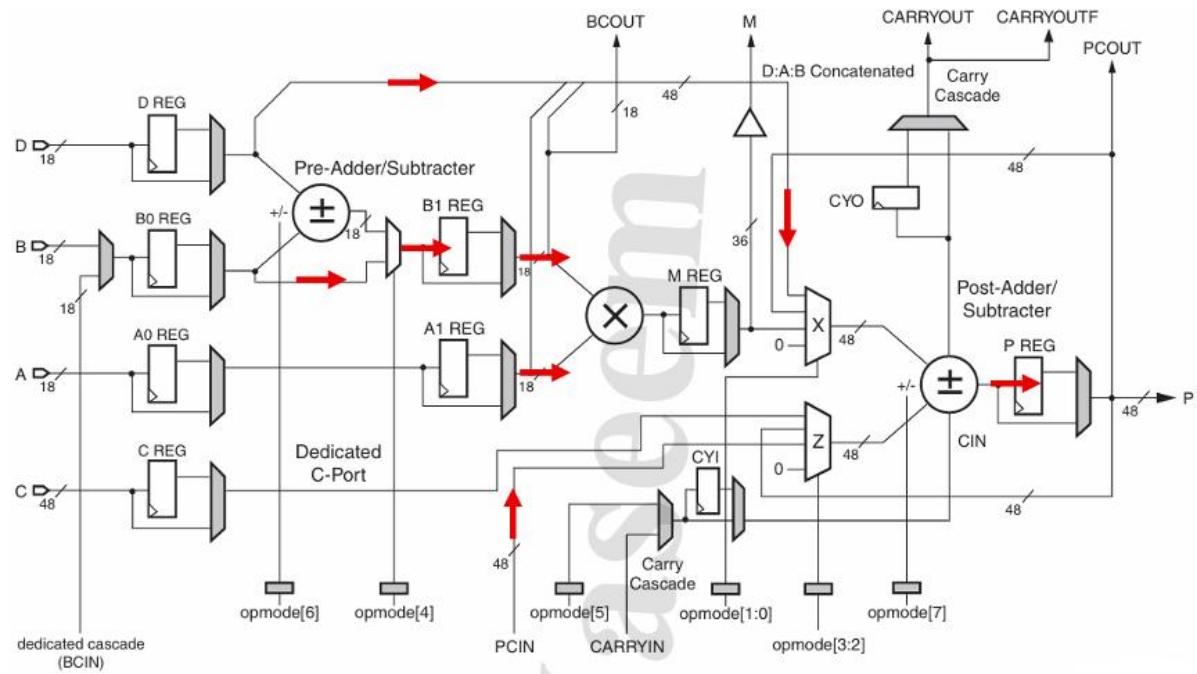


```

135 //////////////////////////////////////////////////////////////////
136 // 2.4 Verify DSP path 3
137 //////////////////////////////////////////////////////////////////
138
139     A_tb = 20 ;
140     B_tb = 10 ;
141     D_tb = 25 ;
142     C_tb = 350 ;
143     OPMODE_tb = 8'b00001010;
144     CARRYIN_tb = $random;
145     BCIN_tb = $random;
146     PCIN_tb = $random;
147     BCOUT_expected = 18'ha;
148     M_expected = 36'hc8;
149     P_expected = 0;
150     PCOUT_expected = 0;
151     CARRYOUT_expected = 0;
152     CARRYOUTF_expected = 0;
153     repeat(3) @(negedge clk);
154     if ((BCOUT_expected != BCOUT) || (M_expected != M) || (P_expected != P) || (PCOUT_expected != PCOUT) ||
155     (CARRYOUT_expected != CARRYOUT) || (CARRYOUTF_expected != CARRYOUTF) ) begin
156         $display("ERROR, PATH3 verification is incorrect");
157         $stop;
158     end

```

4) Path 4 test



```

160 //////////////////////////////////////////////////////////////////
161 // 2.5 Verify DSP path 4
162 //////////////////////////////////////////////////////////////////
163
164     A_tb = 5 ;
165     B_tb = 6 ;
166     D_tb = 25 ;
167     C_tb = 350 ;
168     OPMODE_tb = 8'b10100111;
169     CARRYIN_tb = $random;
170     BCIN_tb = $random;
171     PCIN_tb = 3000;
172     BCOUT_expected = 18'h6;
173     M_expected = 36'h1e;
174     P_expected = 48'hfe6ffffec0bb1;
175     PCOUT_expected = 48'hfe6ffffec0bb1;
176     CARRYOUT_expected = 1;
177     CARRYOUTF_expected = 1;
178     repeat(3) @(negedge clk);
179     if ((BCOUT_expected != BCOUT) || (M_expected != M) || (P_expected != P) || (PCOUT_expected != PCOUT) ||
180     (CARRYOUT_expected != CARRYOUT) || (CARRYOUTF_expected != CARRYOUTF) ) begin
181         $display("ERROR, PATH4 verification is incorrect");
182         $stop;
183     end
184     $stop;
185 end
186
187 endmodule

```

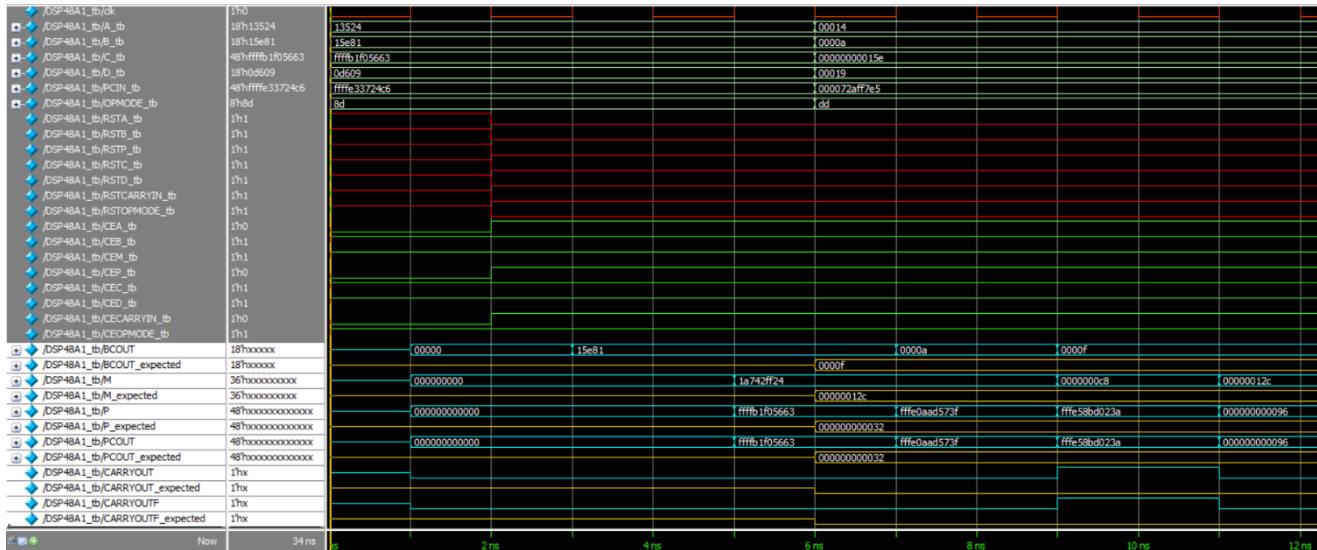
Part III: DO file

```
1  vlib work
2
3  vlog D_FF_mux.v
4  vlog multiplier.v
5  vlog mux21.v
6  vlog mux41.v
7  vlog post_adder_subtracter.v
8  vlog pre_adder_subtracter.v
9  vlog Spartan6.v
10 vlog Spartan6_tb.v
11
12 vsim -voptargs=+acc work.DSP48A1_tb
13
14 add wave *
15 run -all
16 #quit -sim
```

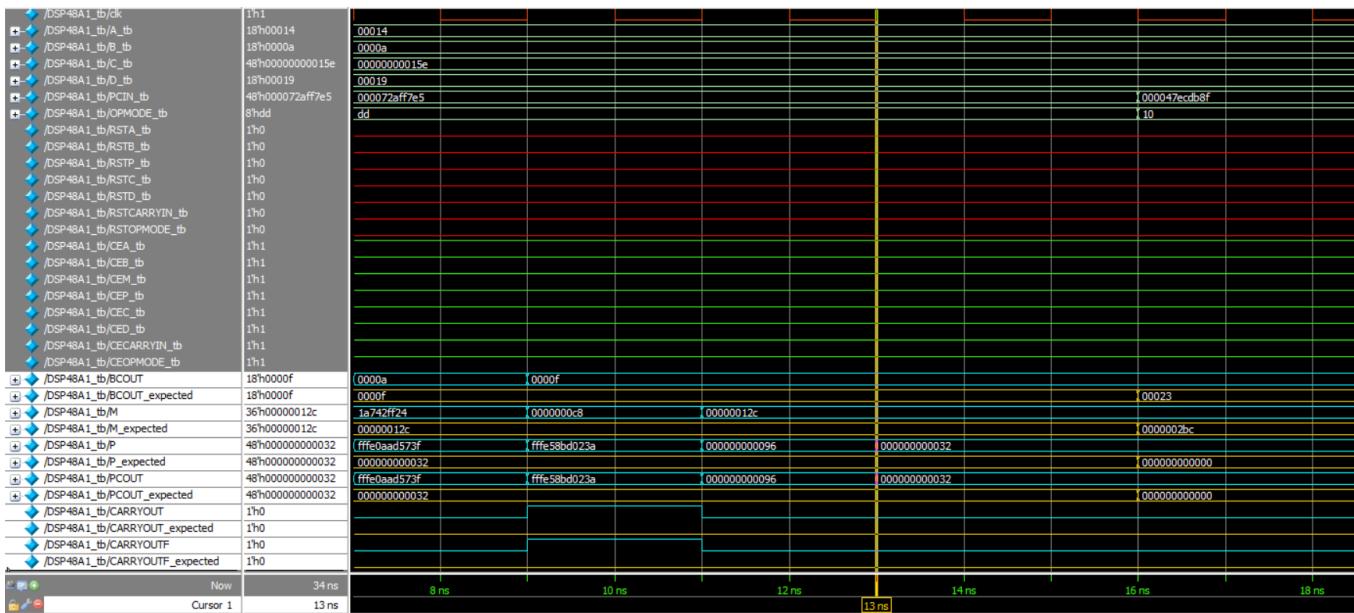
Part IV: QuestaSim Snippets

The highlighted Waves are both the output and the expected values

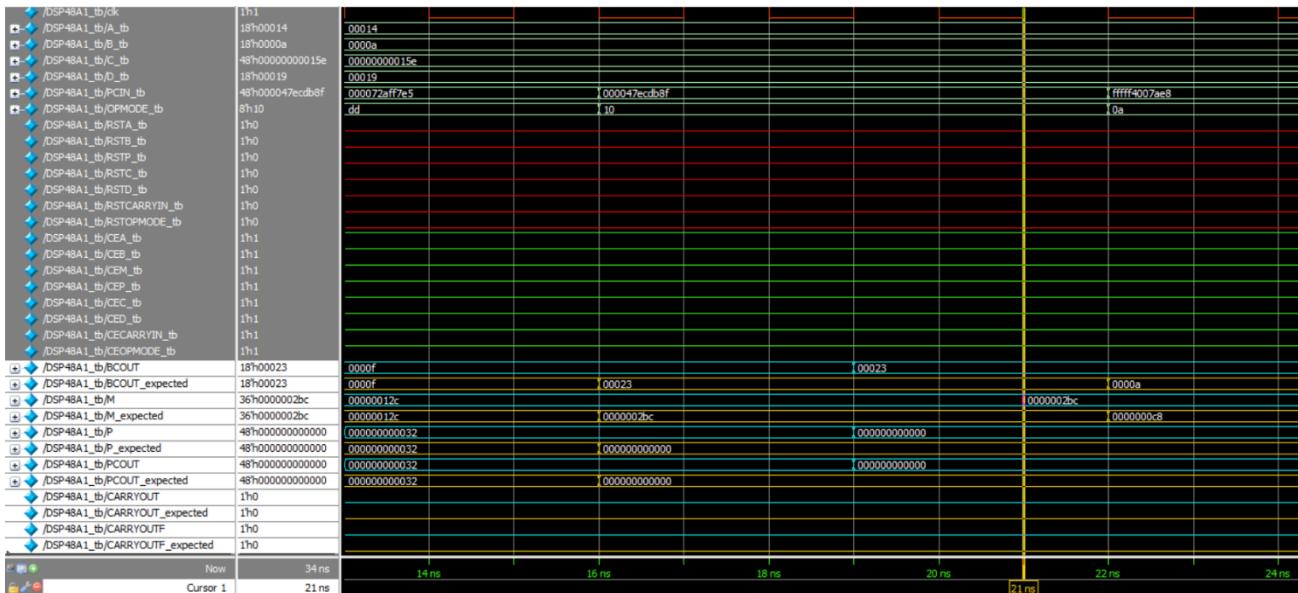
1) reset test waveform



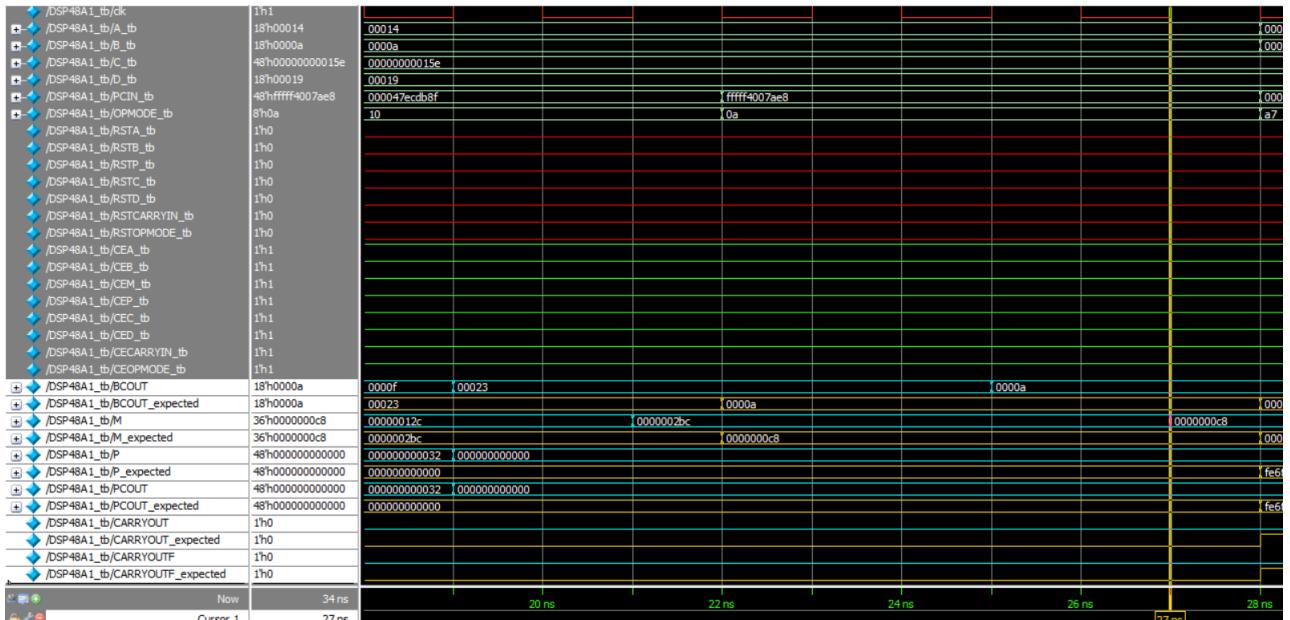
2) Path 1 Waveform



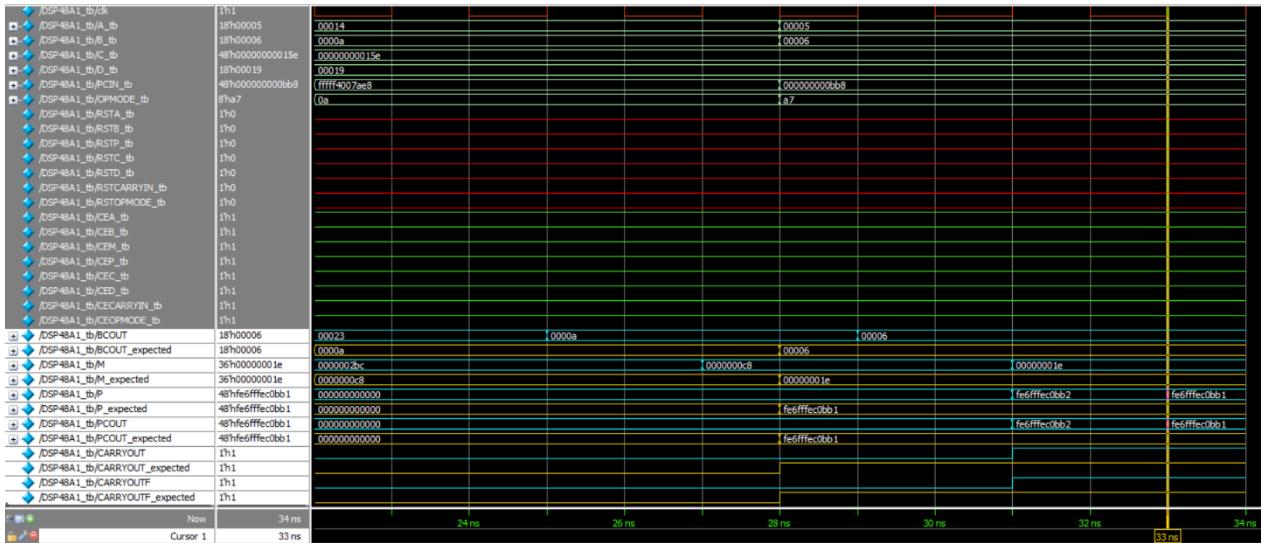
3) Path 2 Waveform



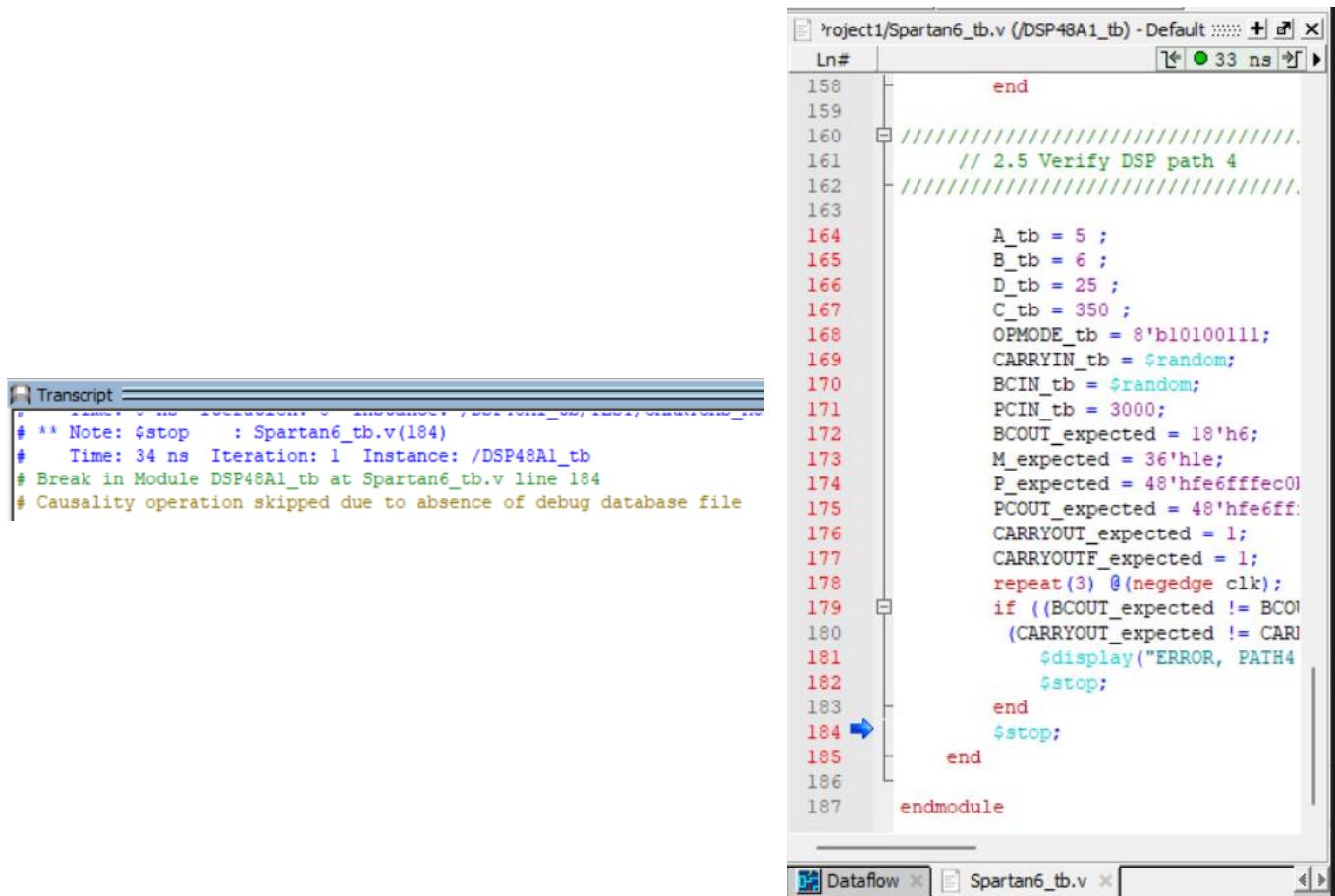
4) Path 3 Waveform



5) Path 4 Waveform



The simulation stopped regularly at the end of the code without detecting any error



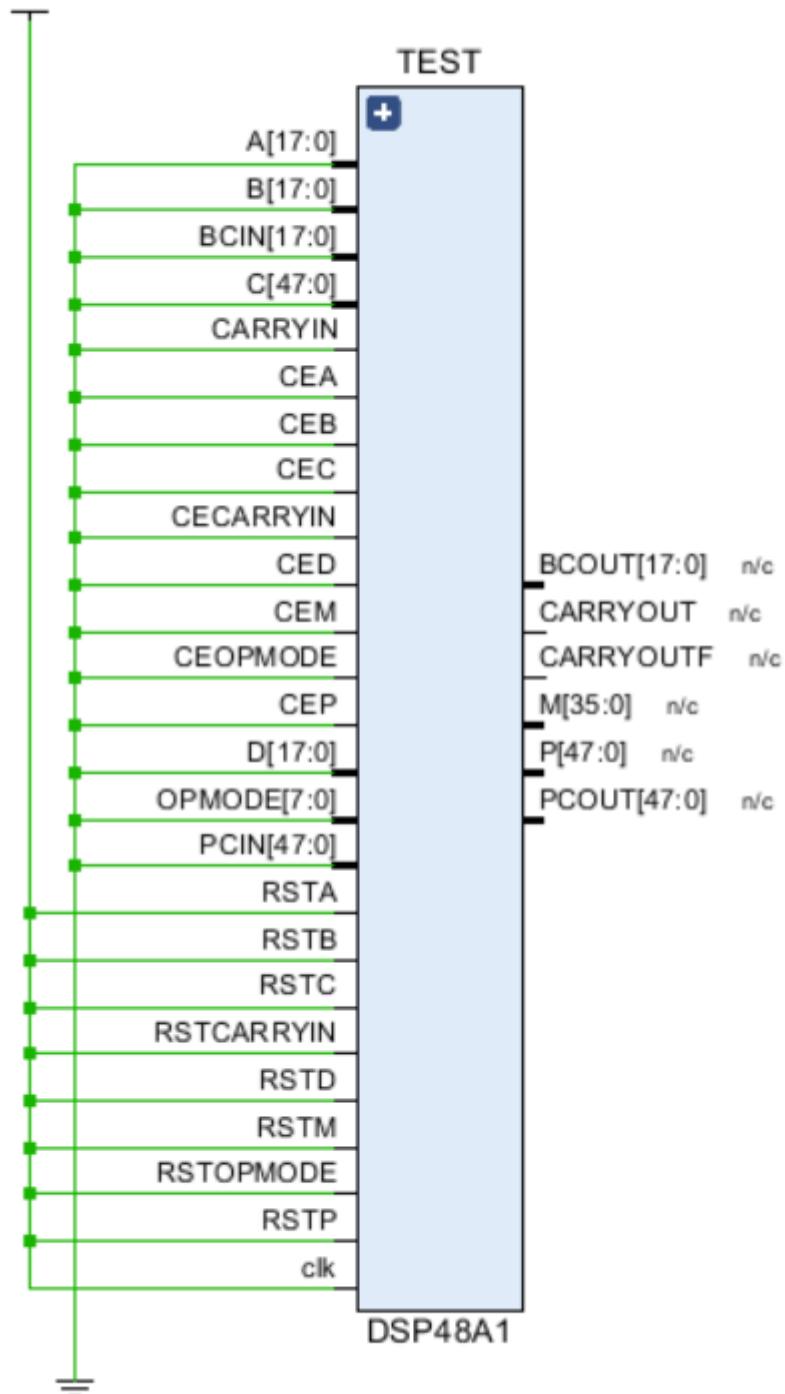
Part V: Constraint file

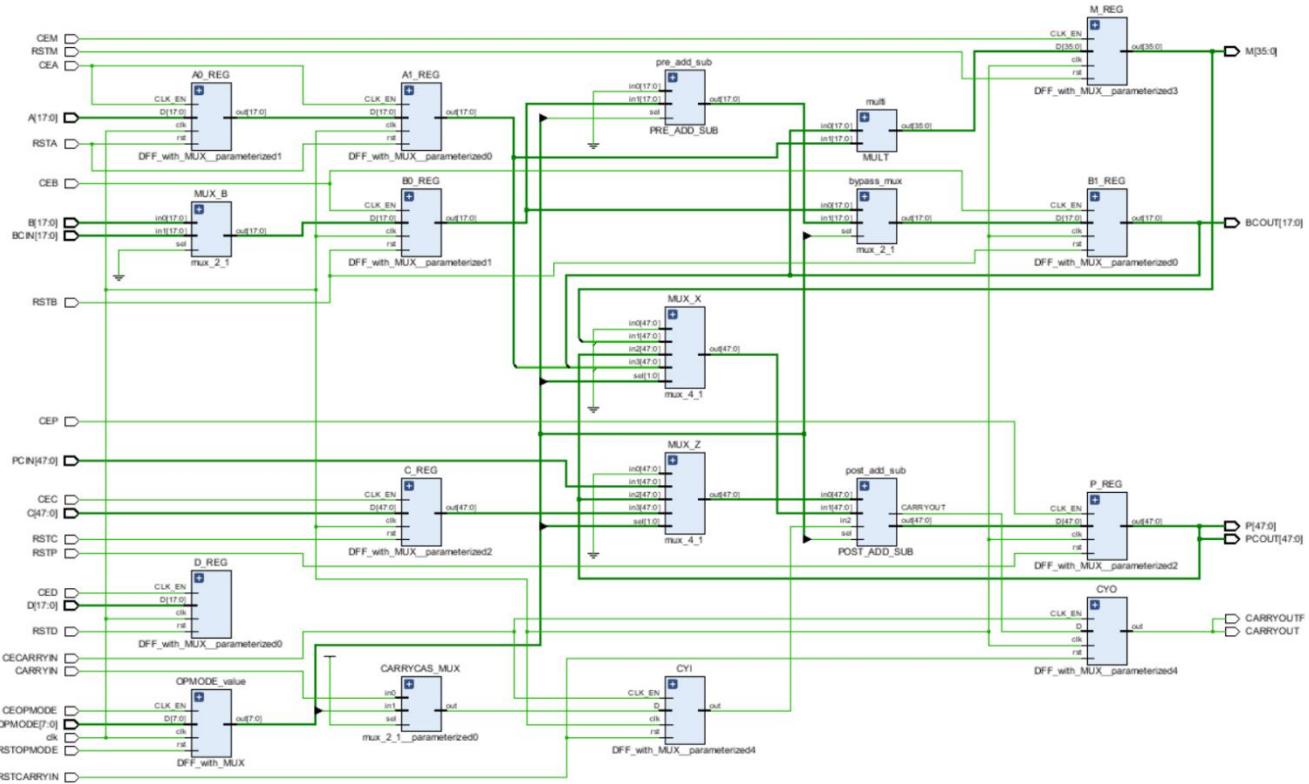
```
6 ## Clock signal
7 set_property -dict { PACKAGE_PIN W5    IOSTANDARD LVCMS33 } [get_ports clk]
8 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]

152 ## Configuration options, can be used for all designs
153 set_property CONFIG_VOLTAGE 3.3 [current_design]
154 set_property CFGBVS VCCO [current_design]
155
156 ## SPI configuration mode options for QSPI boot, can be used for all designs
157 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
158 set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
159 set_property CONFIG_MODE SPIx4 [current_design]
160
161 create_debug_core u_ila_0 ila
162 set_property ALL_PROBE_SAME_MU true [get_debug_cores u_ila_0]
163 set_property ALL_PROBE_SAME_MU_CNT 1 [get_debug_cores u_ila_0]
164 set_property C_ADV_TRIGGER false [get_debug_cores u_ila_0]
165 set_property C_DATA_DEPTH 1024 [get_debug_cores u_ila_0]
166 set_property C_EN_STRG_QUAL false [get_debug_cores u_ila_0]
167 set_property C_INPUT_PIPE_STAGES 0 [get_debug_cores u_ila_0]
168 set_property C_TRIGIN_EN false [get_debug_cores u_ila_0]
169 set_property C_TRIGOUT_EN false [get_debug_cores u_ila_0]
170 set_property port_width 1 [get_debug_ports u_ila_0/clk]
171 connect_debug_port u_ila_0/clk [get_nets [list CLK_IBUF_BUFG]]
172 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe0]
173 set_property port_width 1 [get_debug_ports u_ila_0/probe0]
174 connect_debug_port u_ila_0/probe0 [get_nets [list {Post_Adder_Block/PCIN_IBUF[0]}]]
175 create_debug_port u_ila_0 probe
176 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe1]
177 set_property port_width 18 [get_debug_ports u_ila_0/probe1]
178 connect_debug_port u_ila_0/probe1 [get_nets [list {Pre_Adder_MUX/B_IBUF[0]}] {Pre_Adder_...}]
179 create_debug_port u_ila_0 probe
180 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe2]
181 set_property port_width 48 [get_debug_ports u_ila_0/probe2]
182 connect_debug_port u_ila_0/probe2 [get_nets [list {Z_MUX/PCIN_IBUF[0]}] {Z_MUX/PCIN_IBUF...}]
183 create_debug_port u_ila_0 probe
184 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe3]
185 set_property port_width 48 [get_debug_ports u_ila_0/probe3]
186 connect_debug_port u_ila_0/probe3 [get_nets [list {OPMODE_Block/PCIN_IBUF[0]}] {OPMODE_B...}]
187 create_debug_port u_ila_0 probe
188 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe4]
189 set_property port_width 17 [get_debug_ports u_ila_0/probe4]
190 connect_debug_port u_ila_0/probe4 [get_nets [list {OPMODE_Block/B_IBUF[0]}] {OPMODE_B...}]
191 create_debug_port u_ila_0 probe
192 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe5]
193 set_property port_width 1 [get_debug_ports u_ila_0/probe5]
194 connect_debug_port u_ila_0/probe5 [get_nets [list CY0_Block/CARRYOUTF_OBUF]]
195 create_debug_port u_ila_0 probe
196 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe6]
197 set_property port_width 1 [get_debug_ports u_ila_0/probe6]
198 connect_debug_port u_ila_0/probe6 [get_nets [list CYI_Block/CECARRYIN_IBUF]]
199 create_debug_port u_ila_0 probe
200 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe7]
201 set_property port_width 1 [get_debug_ports u_ila_0/probe7]
202 connect_debug_port u_ila_0/probe7 [get_nets [list CY0_Block/CECARRYIN_IBUF]]
203 create_debug_port u_ila_0 probe
204 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe8]
205 set_property port_width 1 [get_debug_ports u_ila_0/probe8]
206 connect_debug_port u_ila_0/probe8 [get_nets [list M_Block/CEM_IBUF]]
207 create_debug_port u_ila_0 probe
208 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe9]
209 set_property port_width 1 [get_debug_ports u_ila_0/probe9]
210 connect_debug_port u_ila_0/probe9 [get_nets [list CY0_Block/RSTCARRYIN_IBUF]]
211 create_debug_port u_ila_0 probe
212 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe10]
213 set_property port_width 1 [get_debug_ports u_ila_0/probe10]
214 connect_debug_port u_ila_0/probe10 [get_nets [list M_Block/RSTM_IBUF]]
215 set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
216 set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
217 set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
218 connect_debug_port dbg_hub/clk [get_nets CLK_IBUF_BUFG]
```

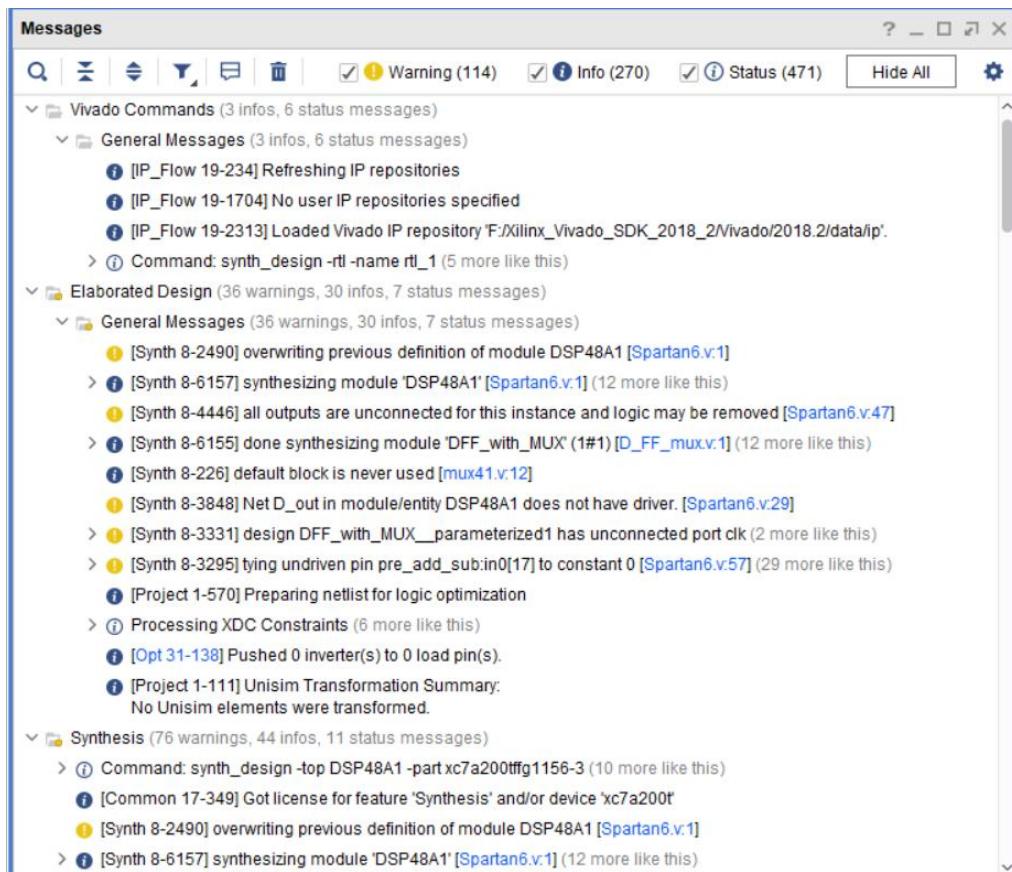
Part VI: Elaboration

1) Schematic Snippets



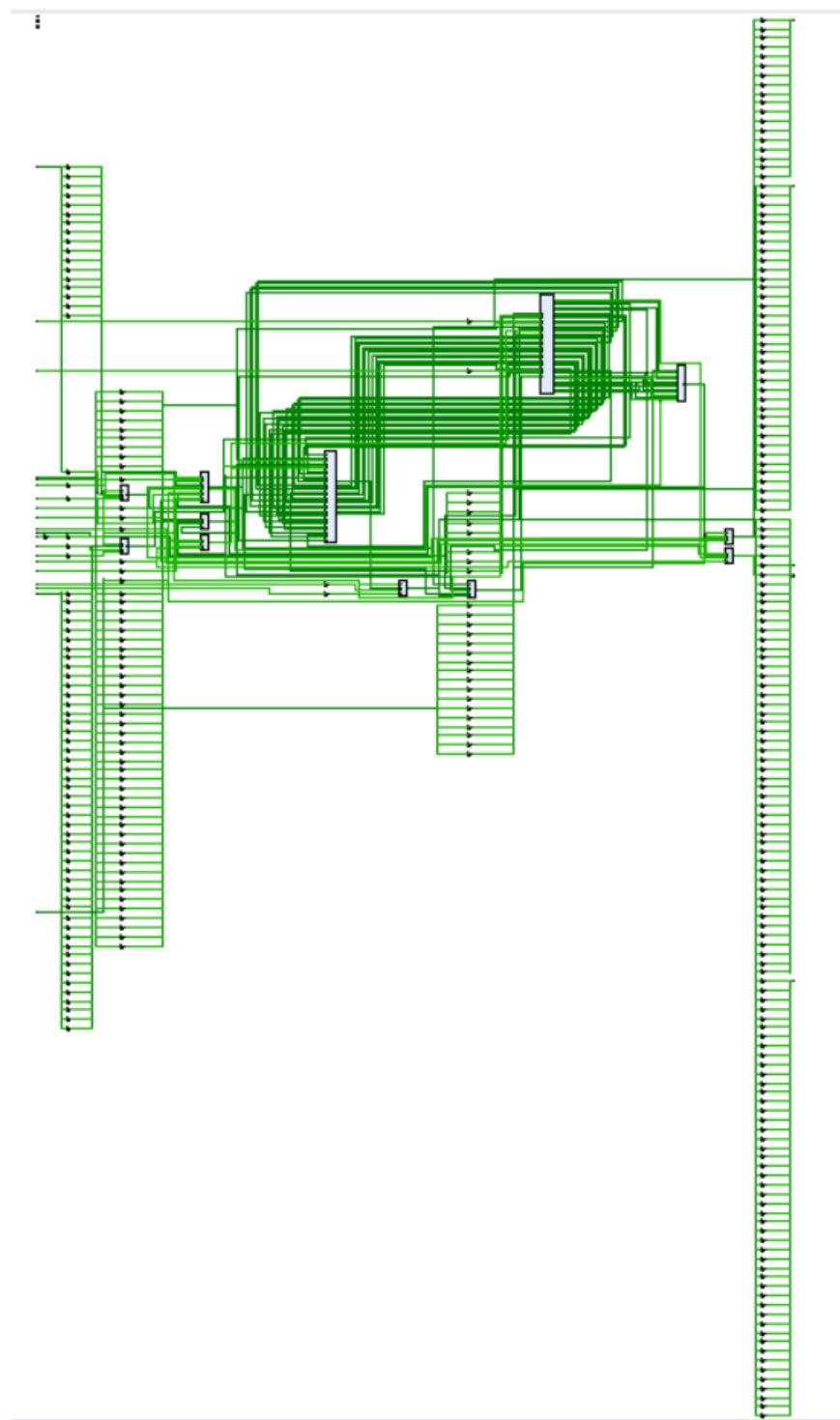


2) Messages tab

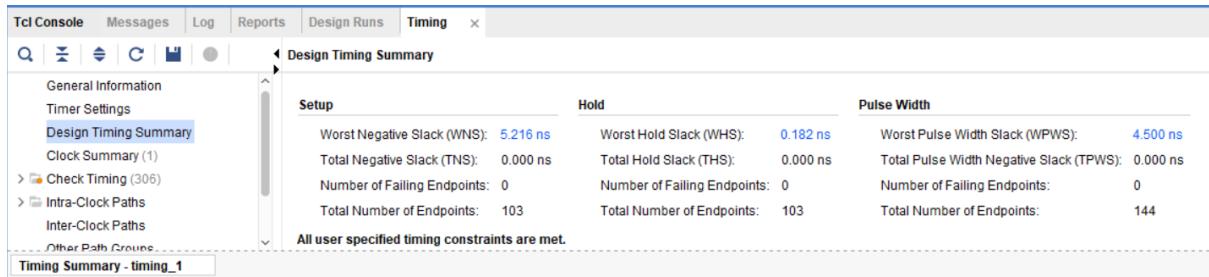


Part VII: Synthesis

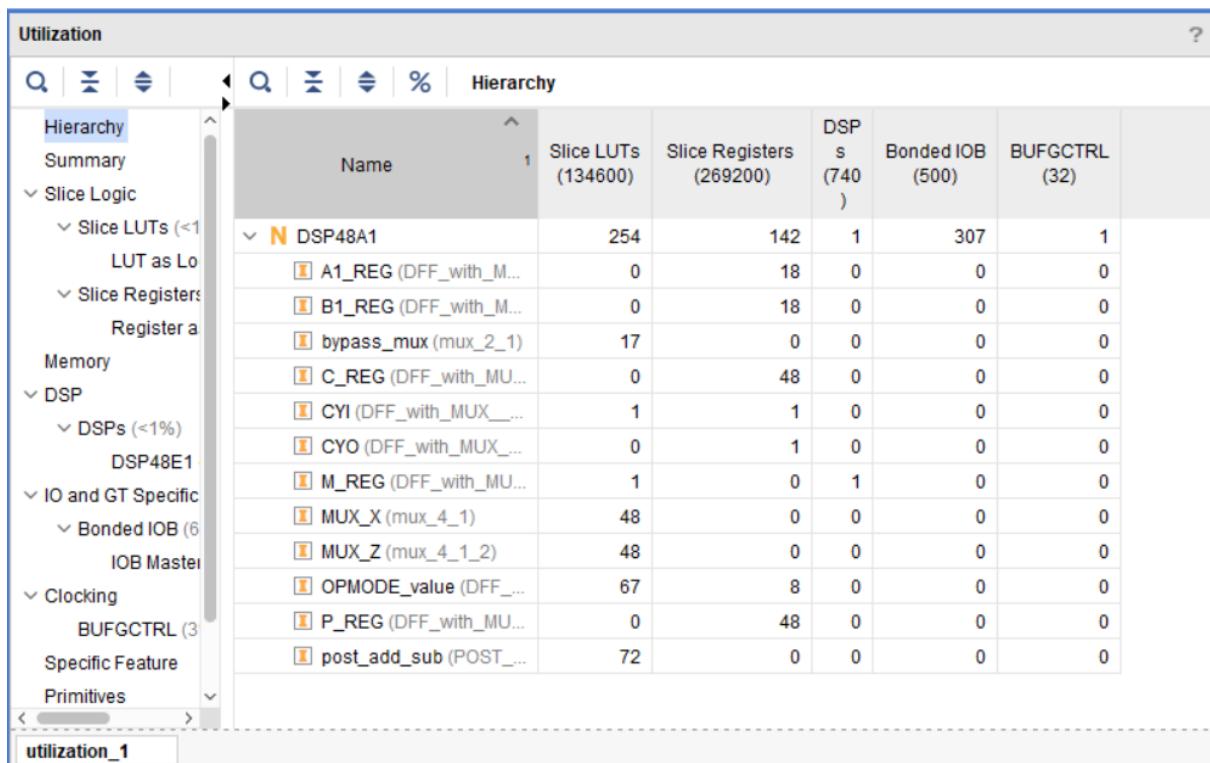
1) Schematic Snippets



2) Timing Report



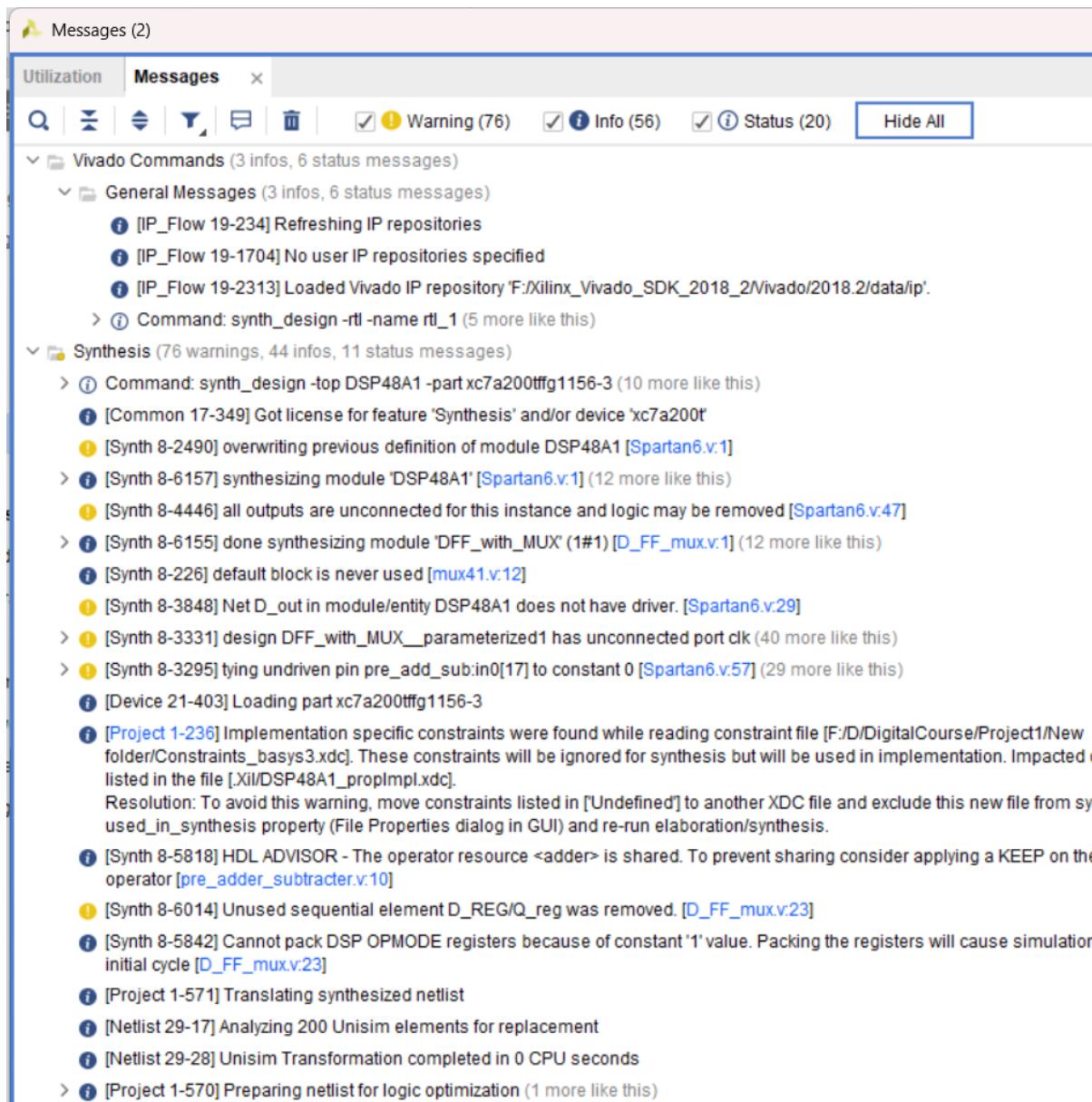
3) Utilization Report



The screenshot shows the Xilinx Vivado Design Suite interface with the 'Utilization' tab selected. The left sidebar shows a hierarchical tree view of utilized components. The main table provides detailed utilization data for each component:

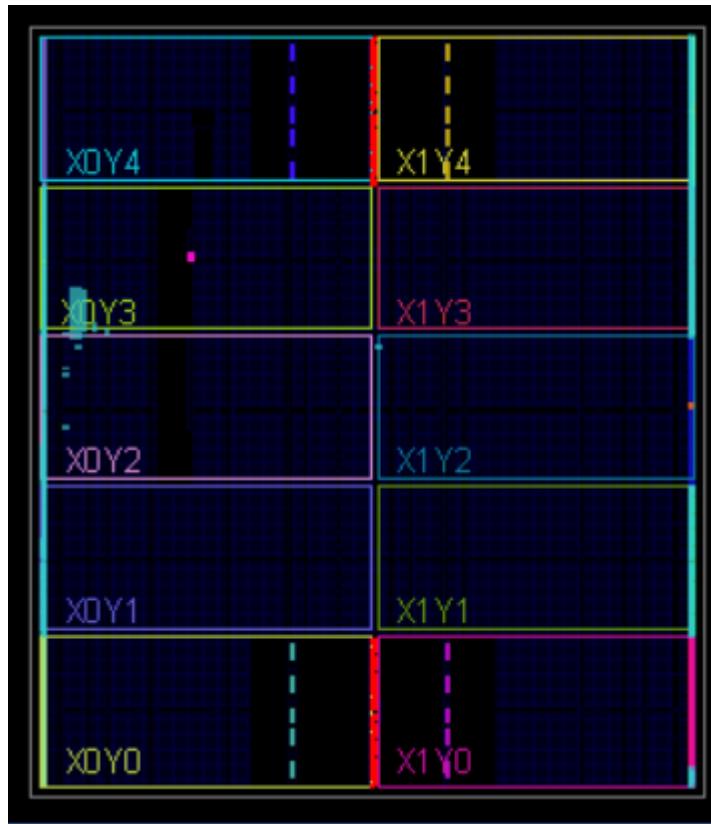
Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
N DSP48A1	254	142	1	307	1
A1_REG (DFF_with_MUX)	0	18	0	0	0
B1_REG (DFF_with_MUX)	0	18	0	0	0
bypass_mux (mux_2_1)	17	0	0	0	0
C_REG (DFF_with_MUX)	0	48	0	0	0
CYI (DFF_with_MUX)	1	1	0	0	0
CYO (DFF_with_MUX)	0	1	0	0	0
M_REG (DFF_with_MUX)	1	0	1	0	0
MUX_X (mux_4_1)	48	0	0	0	0
MUX_Z (mux_4_1_2)	48	0	0	0	0
OPMODE_value (DFF)	67	8	0	0	0
P_REG (DFF_with_MUX)	0	48	0	0	0
post_add_sub (POST)	72	0	0	0	0

4) Messages tab

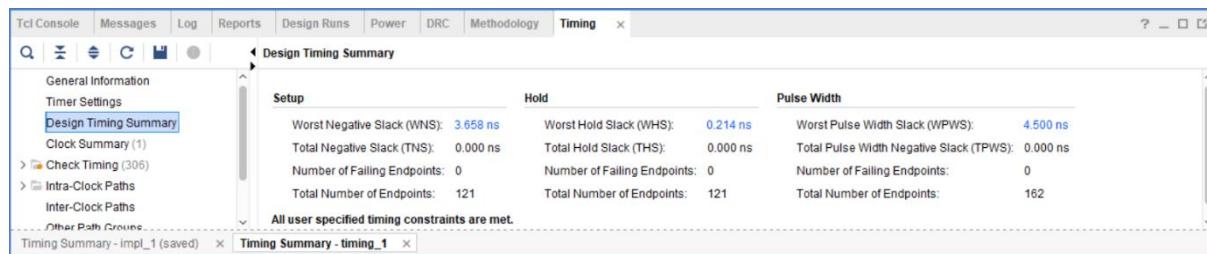


Part VIII: Implementation

1) Device Snippet



2) Timing Report



3) Utilization Report

Utilization

The Utilization report shows the following resource usage for the DSP48A1 module:

Name	Slice LUTs (133800)	Slice Registers (267600)	Slice (33450)	LUT as Logic (133800)	LUT Flip Flop Pairs (133800)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP48A1	253	160	99	253	25	1	307	1
A1_REG (DFF_with_MUX)	0	18	7	0	0	0	0	0
B1_REG (DFF_with_MUX)	0	35	10	0	0	0	0	0
bypass_mux (mux_2_1)	17	0	10	17	0	0	0	0
C_REG (DFF_with_MUX)	0	48	17	0	0	0	0	0
CYI (DFF_with_MUX)	1	1	1	1	1	0	0	0
CYO (DFF_with_MUX)	0	2	1	0	0	0	0	0
M_REG (DFF_with_MUX)	0	0	0	0	0	1	0	0
MUX_X (mux_4_1)	48	0	13	48	0	0	0	0
MUX_Z (mux_4_1_2)	48	0	16	48	0	0	0	0
OPMODE_value (DFF)	67	8	48	67	0	0	0	0
P_REG (DFF_with_MUX)	0	48	12	0	0	0	0	0
post_add_sub (POST)	72	0	24	72	0	0	0	0

4) Messages tab

Messages (2)

The Messages tab displays the following log entries:

Category	Message
Vivado Commands	[IP_Flow 19-234] Refreshing IP repositories
Vivado Commands	[IP_Flow 19-1704] No user IP repositories specified
Vivado Commands	[IP_Flow 19-2313] Loaded Vivado IP repository F:/Xilinx_Vivado_SDK_2018_2/Vivado/2018.2/data/ip'.
Synthesis	Command: synth_design -rtl_name rtl_1 (5 more like this)
Synthesis	[Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a200tffg1156-3' (10 more like this)
Synthesis	[Synth 8-2490] overwriting previous definition of module DSP48A1 [Spartan6.v:1]
Synthesis	[Synth 8-6157] synthesizing module 'DSP48A1' [Spartan6.v:1] (12 more like this)
Synthesis	[Synth 8-4446] all outputs are unconnected for this instance and logic may be removed [Spartan6.v:47]
Synthesis	[Synth 8-6155] done synthesizing module 'DFF_with_MUX' (#1) [D_FF_mux.v:1] (12 more like this)
Synthesis	[Synth 8-226] default block is never used [mux41.v:12]
Synthesis	[Synth 8-3848] Net D_out in module/entity DSP48A1 does not have driver. [Spartan6.v:29]
Synthesis	[Synth 8-3331] design DFF_with_MUX_parameterized1 has unconnected port clk (40 more like this)
Synthesis	[Synth 8-3295] tying undriven pin pre_add_sub:in0[17] to constant 0 [Spartan6.v:57] (29 more like this)
Synthesis	[Device 21-403] Loading part xc7a200tffg1156-3
Synthesis	[Project 1-236] Implementation specific constraints were found while reading constraint file [F:/DigitalCourse/Project1/New folder/Constraints_basys3.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]. Resolution: To avoid this warning, move constraints listed in [Undefined] 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.
Synthesis	[Synth 8-5818] HDL ADVISOR - The operator resource <adder> is shared. To prevent sharing consider applying a KEEP on the output of the operator [pre_adder_subtractor.v:10]
Synthesis	[Synth 8-6014] Unused sequential element D_REG/Q_reg was removed. [D_FF_mux.v:23]
Synthesis	[Synth 8-5842] Cannot pack DSP OPMODE registers because of constant '1' value. Packing the registers will cause simulation mismatch at initial cycle [D_FF_mux.v:23]
Synthesis	[Project 1-571] Translating synthesized netlist
Netlist	Analyzing 200 Unisim elements for replacement
Netlist	Unisim Transformation completed in 0 CPU seconds
Netlist	[Project 1-570] Preparing netlist for logic optimization (1 more like this)

Part IV: Linting

1) Before using Waivers

Lint Summary

Name	Count
Open(unexpected, pending)	9 (15)
Warning	2
Info	7 (13)
condition const	2
parameter_name_du...	2
multi_ports_in_sing...	3 (9)

Lint Checks

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference
w	w	mux_select_const		Constant value drives mux select pin. Signal CARRYC...	mux_2_1	Connectivity	open	unassis...	
w	w	mux_select_const		Constant value drives mux select pin. Signal MUX_B...	mux_2_1	Connectivity	open	unassis...	
w	w	condition_const		Condition expression is a constant. Module DFF_with...	DFF_with_MUX	Rtl Design Style	open	unassis...	
w	w	condition_const		Condition expression is a constant. Module DSP48A1...	DSP48A1	Rtl Design Style	open	unassis...	
w	w	parameter_name_du...		Same parameter name is used in more than one mod...	DFF_with_MUX	Nomenclature...	open	unassis...	1.1.4.3, 3.2.2.2
w	w	parameter_name_du...		Same parameter name is used in more than one mod...	DFF_with_MUX	Nomenclature...	open	unassis...	1.1.4.3, 3.2.2.2
w	w	multi_ports_in_sing...		Multiple ports are declared in one line. Module DFF...	DFF_with_MUX	Rtl Design Style	open	unassis...	3.5.6.3
w	w	multi_ports_in_sing...		Multiple ports are declared in one line. Module DSP48...	DSP48A1	Rtl Design Style	open	unassis...	3.5.6.3
w	w	multi_ports_in_sing...		Multiple ports are declared in one line. Module POST...	POST_ADD...	Rtl Design Style	open	unassis...	3.5.6.3

These warnings are because of using attributes controlling the selectors of the multiplexers

2) After adding Waivers

Lint Summary

Name	Count
Resolved(verified, fixed, ...)	9
Warning	2
Info	7
condition const	2
parameter_name_du...	2
multi_ports_in_sing...	3

Lint Checks

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference
w	w	mux_select_const		Constant value drives mux select pin. Signal CARRYC...	mux_2_1	Connectivity	resolved	unassis...	
w	w	mux_select_const		Constant value drives mux select pin. Signal MUX_B...	mux_2_1	Connectivity	resolved	unassis...	
w	w	condition_const		Condition expression is a constant. Module DFF_with...	DFF_with_MUX	Rtl Design Style	resolved	unassis...	
w	w	condition_const		Condition expression is a constant. Module DSP48A1...	DSP48A1	Rtl Design Style	resolved	unassis...	
w	w	parameter_name_du...		Same parameter name is used in more than one mod...	DFF_with_MUX	Nomenclature...	resolved	unassis...	1.1.4.3, 3.2.2.2
w	w	parameter_name_du...		Same parameter name is used in more than one mod...	DFF_with_MUX	Nomenclature...	resolved	unassis...	1.1.4.3, 3.2.2.2
w	w	multi_ports_in_sing...		Multiple ports are declared in one line. Module DFF...	DFF_with_MUX	Rtl Design Style	resolved	unassis...	3.5.6.3
w	w	multi_ports_in_sing...		Multiple ports are declared in one line. Module DSP48...	DSP48A1	Rtl Design Style	resolved	unassis...	3.5.6.3
w	w	multi_ports_in_sing...		Multiple ports are declared in one line. Module POST...	POST_ADD...	Rtl Design Style	resolved	unassis...	3.5.6.3

3) Linting

