

Project (1): “DSP48A1”

Team Name: Silicon Maestro

Team Member(solo): Amr Ayman Mohamed Abdo

1. RTL Code

```
registered.v x DSP48A1.v x DSP48A1_tb.v x run_dff.do x DSP.xdc x
1 module DSP48A1 #(
2     parameter A0REG = 0,
3     parameter A1REG = 1,
4     parameter B0REG = 0,
5     parameter B1REG = 1,
6     parameter CREG = 1,
7     parameter DREG = 1,
8     parameter MREG = 1,
9     parameter PREG = 1,
10    parameter CARRYINREG = 1,
11    parameter CARRYOUTREG = 1,
12    parameter OPMODEREG = 1,
13    parameter CARRYINSEL = "OPMODE5",
14    parameter B_INPUT = "DIRECT",
15    parameter RSTTYPE = "SYNC"
16 ) (
17     input [17:0] A, B, D,
18     input [47:0] C,
19     input [7:0] OPMODE,
20     input [17:0] BCIN,
21     input [47:0] PCIN,
22     input CLK, CARRYIN,
23     input RSTA, RSTB, RSTC, RSTCARRYIN, RSTD, RSTM, RSTOPMODE, RSTP,
24     input CEA, CEB, CEC, CECARRYIN, CED, CEM, CEOPMODE, CEP,
25     output [17:0] BCOUT,
26     output [47:0] PCOUT, P,
27     output [35:0] M,
28     output CARRYOUT,
29     output CARRYOUTF
30 );
31
32 wire [17:0] A0_reg, B0_reg, D_reg, A1_reg, B1_reg;
33 wire [47:0] C_reg;
34 wire [35:0] M_reg;
35 wire [7:0] OPMODE_reg;
36 wire [17:0] BINPUT;
37 wire [35:0] M_mult;
38 wire [17:0] B1_mux;
39 wire [47:0] X, Z;
40 wire [47:0] X_POST_Z;
41 wire CIN, COUT, CARRYIN_mux;
42 wire [47:0] M_to_mux;
43 wire [17:0] Pre_AS;
44 wire [47:0] DAB_concat;
45 genvar i;
46
47 assign BINPUT = (B_INPUT == "DIRECT")? B : (B_INPUT == "CASCADE")? BCIN : 0;
48 assign CARRYIN_mux = (CARRYINSEL == "OPMODE5")? OPMODE_reg[5] : (CARRYINSEL == "CARRYIN")? CARRYIN : 0;
49 assign Pre_AS = (OPMODE_reg[6])? D_reg - B0_reg : D_reg + B0_reg;
50 assign B1_mux = (OPMODE_reg[4])? Pre_AS : B0_reg;
51 assign M_mult = A1_reg * B1_reg;
52 assign M_to_mux = {12'b0, M_reg};
53 assign DAB_concat = {D[11:0], A1_reg, B1_reg};
54 assign X = (OPMODE_reg[1:0] == 0)? 0 : (OPMODE_reg[1:0] == 1)? M_to_mux : (OPMODE_reg[1:0] == 2)? P : {D[11:0], A1_reg, B1_reg};
55 assign Z = (OPMODE_reg[3:2] == 0)? 0 : (OPMODE_reg[3:2] == 1)? PCIN : (OPMODE_reg[3:2] == 2)? P : C_reg;
56 assign {COUT, X_POST_Z} = (OPMODE_reg[7])? Z - (X+CIN) : Z+X+CIN;
57 assign CARRYOUTF = CARRYOUT;
58 assign BCOUT = B1_reg;
59 assign PCOUT = P;
60
61 generate
62     for (i = 0; i < 36; i = i + 1)
63     begin : buffer
64         buf mbuf[M[i], M_reg[i]];
65     end
66 endgenerate
```

```

67
68     registered #(.REG(A0REG),
69                 .INWIDTH(18),
70                 .RSTTYPE(RSTTYPE)) A0_REG
71                 (.ce(CEA), .rst(RSTA), .clk(CLK), .in(A), .out(A0_reg));
72
73     registered #(.REG(B0REG),
74                 .INWIDTH(18),
75                 .RSTTYPE(RSTTYPE)) B0_REG
76                 (.ce(CEB), .rst(RSTB), .clk(CLK), .in(BINPUT), .out(B0_reg));
77
78     registered #(.REG(CREG),
79                 .INWIDTH(48),
80                 .RSTTYPE(RSTTYPE)) C_REG
81                 (.ce(CEC), .rst(RSTC), .clk(CLK), .in(C), .out(C_reg));
82
83     registered #(.REG(DREG),
84                 .INWIDTH(18),
85                 .RSTTYPE(RSTTYPE)) D_REG
86                 (.ce(CED), .rst(RSTD), .clk(CLK), .in(D), .out(D_reg));
87
88     registered #(.REG(MREG),
89                 .INWIDTH(36),
90                 .RSTTYPE(RSTTYPE)) M_REG
91                 (.ce(CEM), .rst(RSTM), .clk(CLK), .in(M_mult), .out(M_reg));
92
93     registered #(.REG(OPMODEREG),
94                 .INWIDTH(8),
95                 .RSTTYPE(RSTTYPE)) OPMODE_REG
96                 (.ce(CEOPMODE), .rst(RSTOPMODE), .clk(CLK), .in(OPMODE), .out(OPMODE_reg));
97
98     registered #(.REG(PREG),
99                 .INWIDTH(48),
100                 .RSTTYPE(RSTTYPE)) P_REG
101                 (.ce(CEP), .rst(RSTP), .clk(CLK), .in(X_POST_Z), .out(P));
102
103     registered #(.REG(A1REG),
104                 .INWIDTH(18),
105                 .RSTTYPE(RSTTYPE)) A1_REG
106                 (.ce(CEA), .rst(RSTA), .clk(CLK), .in(A0_reg), .out(A1_reg));
107
108     registered #(.REG(B1REG),
109                 .INWIDTH(18),
110                 .RSTTYPE(RSTTYPE)) B1_REG
111                 (.ce(CEB), .rst(RSTB), .clk(CLK), .in(B1_mux), .out(B1_reg));
112
113     registered #(.REG(CARRYINREG),
114                 .INWIDTH(1),
115                 .RSTTYPE(RSTTYPE)) CYI
116                 (.ce(CECARRYIN), .rst(RSTCARRYIN), .clk(CLK), .in(CARRYIN_mux), .out(CIN));
117
118     registered #(.REG(CARRYOUTREG),
119                 .INWIDTH(1),
120                 .RSTTYPE(RSTTYPE)) CYO
121                 (.ce(1), .rst(0), .clk(CLK), .in(COUT), .out(CARRYOUT));
122
123 endmodule

```



```
1  module registered #(
2      parameter REG = 0,
3      parameter INWIDTH = 18,
4      parameter RSTTYPE = "SYNC"
5  )(
6      input ce, rst, clk,
7      input [INWIDTH-1:0] in,
8      output [INWIDTH-1:0] out
9  );
10
11     reg [INWIDTH-1:0] registered;
12
13     assign out = (REG)? registered : in;
14
15     generate
16         if(RSTTYPE == "ASYNC") begin
17             always @(posedge clk or posedge rst) begin
18                 if(rst)
19                     registered <= 0;
20                 else if(ce)
21                     registered <= in;
22             end
23         end
24
25         else if(RSTTYPE == "SYNC") begin
26             always @(posedge clk) begin
27                 if(ce) begin
28                     if(rst)
29                         registered <= 0;
30                     else
31                         registered <= in;
32                 end
33             end
34         end
35     endgenerate
36
37 endmodule
```

2. Testbench code

```
registered.v x DSP48A1.v x DSP48A1_tb.v x run_dff.do x DSP.xdc x
1 module DSP48A1_tb();
2     reg [17:0] A, B, D;
3     reg [47:0] C;
4     reg [7:0] OPMODE;
5     reg [17:0] BCIN;
6     reg [47:0] PCIN;
7     reg CLK, CARRYIN;
8     reg RSTA, RSTB, RSTC, RSTCARRYIN, RSTD, RSTM, RSTOPMODE, RSTP;
9     reg CEA, CEB, CEC, CECARRYIN, CED, CEM, CEOPMODE, CEP;
10    wire [17:0] BCOUT;
11    wire [47:0] PCOUT, P;
12    wire [35:0] M;
13    wire CARRYOUT;
14    wire CARRYOUTF;
15
16    //Inputs being operated on (forced inputs in the first clock cycle)
17    reg [17:0] A_op, B_op, D_op, BCIN_op, BCOUT_op;
18    reg [47:0] C_op, PCIN_op;
19    reg [35:0] M_op;
20    reg [7:0] OPMODE_op;
21    reg CARRYIN_op;
22    reg [47:0] DAB_concat_op;
23
24    //Expected outputs
25    reg [17:0] BCOUT_exp;
26    reg [35:0] M_exp;
27    reg [48:0] COUTandP_exp;
28    reg [48:0] X_POST_Z;
29    reg [47:0] P_exp;
30    reg CARRYOUT_exp;
31
32    //Instantiating the DUT module
33    DSP48A1 DUT(A, B, D, C, OPMODE, BCIN, PCIN, CLK, CARRYIN, RSTA, RSTB, RSTC, RSTCARRYIN, RSTD, RSTM, RSTOPMODE,
34               RSTP, CEA, CEB, CEC, CECARRYIN, CED, CEM, CEOPMODE, CEP, BCOUT, PCOUT, P, M, CARRYOUT, CARRYOUTF);
35
36    //Implementing the clock
37    initial begin
38        CLK = 0;
39        forever
40            #10 CLK = ~CLK;
41    end
42
43    //Generating stimulus and comparing outputs and their corresponding expected outputs
44    initial begin
45        //Initializing the module by resetting everything
46        {RSTA, RSTB, RSTC, RSTCARRYIN, RSTD, RSTM, RSTOPMODE, RSTP} = 8'hFF;
47        {CEA, CEB, CEC, CECARRYIN, CED, CEM, CEOPMODE, CEP} = 0;
48        repeat(2) @(negedge CLK);
49
50        //Starting testing
51        {RSTA, RSTB, RSTC, RSTCARRYIN, RSTD, RSTM, RSTOPMODE, RSTP} = 0;
52        {CEA, CEB, CEC, CECARRYIN, CED, CEM, CEOPMODE, CEP} = 8'hFF;
53
54        //Test case(1): the execution passes through 4 pipelines (4 clock cycles), where OPMODE[1:0] = 2'b01
55        repeat(5000) begin
56            A = $random;
57            B = $random;
58            C = $random;
59            D = $random;
60            PCIN = $random;
61            BCIN = $random;
62            CARRYIN = $random;
63            OPMODE = 8'b1111_1101;
64        end
    end
end
```

```

65 //operating inputs are the forced inputs in the first clock cycle to be operated on in the upcoming cycles
66 A_op = A;
67 B_op = B;
68 C_op = C;
69 D_op = D;
70 PCIN_op = PCIN;
71 BCIN_op = BCIN;
72 CARRYIN_op = CARRYIN;
73 OPMODE_op = OPMODE;
74
75 @(posedge CLK);
76 BCOUT_exp = D_op - B_op;
77 repeat(2) @(negedge CLK);
78 if(BCOUT != BCOUT_exp) begin
79     $display("Error - Incorrect BCOUT, BCOUT_tb = %d, BCOUT_expected = %d", BCOUT, BCOUT_exp);
80     $stop;
81 end
82
83 BCOUT_op = BCOUT;
84
85 @(posedge CLK);
86 M_exp = BCOUT_op * A_op;
87
88 @(negedge CLK);
89 A = $random;
90 B = $random;
91 C = $random;
92 D = $random;
93 PCIN = $random;
94 BCIN = $random;
95 CARRYIN = $random;
96 OPMODE = $random;
97
98 if(M != M_exp) begin
99     $display("Error - Incorrect M, M_tb = %d, M_expected = %d", M, M_exp);
100     $stop;
101 end
102
103 M_op = M;
104
105 @(posedge CLK);
106 COUTandP_exp = C_op-(M_op+OPMODE_op[5]);
107 P_exp = COUTandP_exp[47:0];
108 CARRYOUT_exp = COUTandP_exp[48];
109
110 @(negedge CLK);
111 A = $random;
112 B = $random;
113 C = $random;
114 D = $random;
115 PCIN = $random;
116 BCIN = $random;
117 CARRYIN = $random;
118 OPMODE = $random;
119
120 if(P != P_exp) begin
121     $display("Error - Incorrect P, P_tb = %d, P_exp = %d", P, P_exp);
122     $stop;
123 end
124 if(CARRYOUT != CARRYOUT_exp) begin
125     $display("Error - Incorrect P, CARRYOUT_tb = %d, CARRYOUT_exp = %d", CARRYOUT, CARRYOUT_exp);
126     $stop;
127 end
128 end
129 $display("Test case (1) BCOUT after 2 clock cycles, M after 3 clock cycles, P after 4 clock cycles(X = M_reg)\n");

```

```

130
131 //Test case(2): the execution passes through 3 pipelines (3 clock cycles), where OPMODE[1:0] = 2'b11
132 repeat(5000) begin
133     A = $random;
134     B = $random;
135     C = $random;
136     D = $random;
137     PCIN = $random;
138     BCIN = $random;
139     CARRYIN = $random;
140     OPMODE = 8'b1111_1111;
141
142     //operating inputs are the forced inputs in the first clock cycle to be operated on in the upcoming cycles
143     A_op = A;
144     B_op = B;
145     C_op = C;
146     D_op = D;
147     PCIN_op = PCIN;
148     BCIN_op = BCIN;
149     CARRYIN_op = CARRYIN;
150     OPMODE_op = OPMODE;
151
152     @(posedge CLK);
153     BCOUT_exp = D_op - B_op;
154     repeat(2) @(negedge CLK);
155     if(BCOUT != BCOUT_exp) begin
156         $display("Error - Incorrect BCOUT, BCOUT_tb = %d, BCOUT_expected = %d", BCOUT, BCOUT_exp);
157         $stop;
158     end
159
160     BCOUT_op = BCOUT;
161     DAB_concat_op = {D_op[11:0], A_op, BCOUT_op};
162
163     @(posedge CLK);
164     M_exp = BCOUT_op * A_op;
165     COUTandP_exp = C_op - (DAB_concat_op + OPMODE_op[5]);
166     P_exp = COUTandP_exp[47:0];
167     CARRYOUT_exp = COUTandP_exp[48];
168
169     @(negedge CLK);
170     A = $random;
171     B = $random;
172     C = $random;
173     D = $random;
174     PCIN = $random;
175     BCIN = $random;
176     CARRYIN = $random;
177     OPMODE = $random;
178
179     if(M != M_exp) begin
180         $display("Error - Incorrect M, M_tb = %d, M_expected = %d", M, M_exp);
181         $stop;
182     end
183     if(P != P_exp) begin
184         $display("Error - Incorrect P, P_tb = %d, P_exp = %d", P, P_exp);
185         $stop;
186     end
187     if(CARRYOUT != CARRYOUT_exp) begin
188         $display("Error - Incorrect P, CARRYOUT_tb = %d, CARRYOUT_exp = %d", CARRYOUT, CARRYOUT_exp);
189         $stop;
190     end
191 end
192 $display("Test case (2) BCOUT after 2 clock cycles, M after 3 clock cycles, P after 3 clock cycles(X = D:A:B)\n");
193

```

```

194 //Test case(3): P gets an initial value after the first 3 cycles and then gets updated every clock cycle(X = P)
195 repeat(5000) begin
196     A = $random;
197     B = $random;
198     C = $random;
199     D = $random;
200     PCIN = $random;
201     BCIN = $random;
202     CARRYIN = $random;
203     OPMODE = 8'b1111_1110;
204
205     //operating inputs are the forced inputs in the first clock cycle to be operated on in the upcoming cycles
206     A_op = A;
207     B_op = B;
208     C_op = C;
209     D_op = D;
210     PCIN_op = PCIN;
211     BCIN_op = BCIN;
212     CARRYIN_op = CARRYIN;
213     OPMODE_op = OPMODE;
214
215     @(posedge CLK);
216     BCOU_exp = D_op - B_op;
217     repeat(2) @(negedge CLK);
218     if(BCOUT != BCOU_exp) begin
219         $display("Error - Incorrect BCOU, BCOU_tb = %d, BCOU_expected = %d", BCOU, BCOU_exp);
220         $stop;
221     end
222
223     BCOU_op = BCOU;
224     DAB_concat_op = {D_op[11:0], A_op, BCOU_op};
225     X_POST_Z = C_op-(P+OPMODE_op[5]);
226
227     @(posedge CLK);
228     M_exp = BCOU_op * A_op;
229     COUTandP_exp = X_POST_Z;
230     P_exp = COUTandP_exp[47:0];
231     CARRYOUT_exp = COUTandP_exp[48];
232
233
234     @(negedge CLK);
235     A = $random;
236     B = $random;
237     C = $random;
238     D = $random;
239     PCIN = $random;
240     BCIN = $random;
241     CARRYIN = $random;
242     OPMODE = $random;
243
244     if(M != M_exp) begin
245         $display("Error - Incorrect M, M_tb = %d, M_expected = %d", M, M_exp);
246         $stop;
247     end
248     if(P != P_exp) begin
249         $display("Error - Incorrect P, P_tb = %d, P_exp = %d", P, P_exp);
250         $stop;
251     end
252     if(CARRYOUT != CARRYOUT_exp) begin
253         $display("Error - Incorrect P, CARRYOUT_tb = %d, CARRYOUT_exp = %d", CARRYOUT, CARRYOUT_exp);
254         $stop;
255     end
256 end
257 $display("Test case (3) P gets an initial value after the first 3 cycles and then gets updated every clock cycle(X = P)\n");
258
259 $display("DSP48A1 is successfully verified");
260 $stop;
261 end
262 endmodule

```

3. Do file

◀▶ registered.v × DSP48A1.v × DSP48A1_tb.v × run_DSP48A1.do ×

```

1 vlib work
2 vlog registered.v DSP48A1.v DSP48A1_tb.v
3 vsim -voptargs=+acc work.DSP48A1_tb
4 add wave *
5 run -all
6 #quit -sim

```


[illegible]

The screenshot displays a logic analyzer interface. On the left, a list of signals is shown, including A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, and various control signals like CS, CE, and CLK. The signals are organized into groups, with some signals having a 'Name' column and others having a 'Value' column. The main area shows a multi-channel waveform display. The top of the waveform shows the signal names, and the bottom shows the time scale. The time scale is set to 1000ns, and the cursor is at 1500ns. The waveform shows digital signals over time, with a time scale of 1000ns and a cursor at 1500ns. The signals are color-coded: A (blue), B (green), C (red), D (purple), E (brown), F (pink), G (gray), H (olive), I (teal), J (light blue), K (light green), L (light purple), M (light brown), N (light pink), O (light gray), P (light olive), Q (light teal), R (light blue), S (light green), T (light purple), U (light brown), V (light pink), W (light gray), X (light olive), Y (light teal), Z (light blue).

5. Constraint File

```
registered.v      x | DSP48A1.v      x | DSP48A1_tb.v      x | run_DSP48A1.do      x | DSP.xdc      x
1  ## Configuration options, can be used for all designs
2  set_property CONFIG_VOLTAGE 3.3 [current_design]
3  set_property CFBV5 VCC0 [current_design]
4
5  ## SPI configuration mode options for QSPI boot, can be used for all designs
6  set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
7  set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
8
9  ## Clock signal
10 set_property -dict {PACKAGE_PIN AL30 IOSTANDARD LVCMOS33} [get_ports CLK]
11 create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [get_ports CLK]
12
13 ##I/O Ports
14 set_property IOSTANDARD LVCMOS33 [get_ports *]
15 set_property PACKAGE_PIN AG7 [get_ports {D[6]}]
16 set_property PACKAGE_PIN AB4 [get_ports {B[3]}]
17 set_property PACKAGE_PIN V4 [get_ports {B[10]}]
18 set_property PACKAGE_PIN W28 [get_ports {M[28]}]
19 set_property PACKAGE_PIN N2 [get_ports {C[29]}]
20 set_property PACKAGE_PIN U7 [get_ports {C[6]}]
21 set_property PACKAGE_PIN Y25 [get_ports {M[25]}]
22 set_property PACKAGE_PIN AG5 [get_ports {D[12]}]
23 set_property PACKAGE_PIN V26 [get_ports {M[32]}]
24 set_property PACKAGE_PIN AH28 [get_ports {P[25]}]
25 set_property PACKAGE_PIN AG34 [get_ports {P[47]}]
26 set_property PACKAGE_PIN AE32 [get_ports {P[38]}]
27 set_property PACKAGE_PIN V34 [get_ports {M[17]}]
28 set_property PACKAGE_PIN AF33 [get_ports {P[41]}]
29 set_property PACKAGE_PIN L25 [get_ports {PCIN[41]}]
30 set_property PACKAGE_PIN AB32 [get_ports {M[7]}]
31 set_property PACKAGE_PIN AD29 [get_ports {P[31]}]
32 set_property PACKAGE_PIN AB31 [get_ports {M[8]}]
33 set_property PACKAGE_PIN V24 [get_ports {M[35]}]
34 set_property PACKAGE_PIN W10 [get_ports {BCOUT[16]}]
35 set_property PACKAGE_PIN W8 [get_ports {BCOUT[11]}]
36 set_property PACKAGE_PIN M6 [get_ports {BCOUT[4]}]
37 set_property PACKAGE_PIN AD25 [get_ports {C[45]}]
38 set_property PACKAGE_PIN M5 [get_ports {C[34]}]
39 set_property PACKAGE_PIN K28 [get_ports {PCIN[27]}]
40 set_property PACKAGE_PIN H24 [get_ports {PCIN[37]}]
41 set_property PACKAGE_PIN H27 [get_ports {PCIN[36]}]
42 set_property PACKAGE_PIN J29 [get_ports {PCIN[24]}]
43 set_property PACKAGE_PIN G29 [get_ports {PCIN[20]}]
44 set_property PACKAGE_PIN H32 [get_ports {PCIN[12]}]
45 set_property PACKAGE_PIN G34 [get_ports {PCIN[7]}]
46 set_property PACKAGE_PIN M34 [get_ports {PCOUT[3]}]
47 set_property PACKAGE_PIN M26 [get_ports {PCIN[0]}]
48 set_property PACKAGE_PIN T24 [get_ports {PCOUT[47]}]
49 set_property PACKAGE_PIN N28 [get_ports {PCOUT[31]}]
50 set_property PACKAGE_PIN N34 [get_ports {PCOUT[4]}]
51 set_property PACKAGE_PIN R28 [get_ports {PCOUT[29]}]
52 set_property PACKAGE_PIN P30 [get_ports {PCOUT[21]}]
53 set_property PACKAGE_PIN AD23 [get_ports CARRYOUTF]
54 set_property PACKAGE_PIN AD1 [get_ports RSTD]
55 set_property PACKAGE_PIN AJ3 [get_ports CECARRYIN]
56 set_property PACKAGE_PIN AF4 [get_ports CEC]
57 set_property PACKAGE_PIN AH1 [get_ports RSTM]
58 set_property PACKAGE_PIN AC11 [get_ports RSTP]
59 set_property PACKAGE_PIN AF2 [get_ports CEP]
60 set_property PACKAGE_PIN R5 [get_ports {C[23]}]
61 set_property PACKAGE_PIN R11 [get_ports {BCOUT[6]}]
62 set_property PACKAGE_PIN AB2 [get_ports {B[2]}]
63 set_property PACKAGE_PIN AC2 [get_ports {A[16]}]
64 set_property PACKAGE_PIN W26 [get_ports {M[30]}]
65 set_property PACKAGE_PIN T3 [get_ports {C[16]}]
66 set_property PACKAGE_PIN Y27 [get_ports {M[24]}]
67 set_property PACKAGE_PIN P1 [get_ports {C[31]}]
```



```
135 set_property PACKAGE_PIN U32 [get_ports {PCOUT[11]}]
136 set_property PACKAGE_PIN AF3 [get_ports {CEOPMODE}]
137 set_property PACKAGE_PIN Y5 [get_ports {B[7]}]
138 set_property PACKAGE_PIN AC3 [get_ports {A[13]}]
139 set_property PACKAGE_PIN AA7 [get_ports {A[11]}]
140 set_property PACKAGE_PIN AC7 [get_ports {A[10]}]
141 set_property PACKAGE_PIN AD10 [get_ports {B[17]}]
142 set_property PACKAGE_PIN AE10 [get_ports {B[16]}]
143 set_property PACKAGE_PIN AA5 [get_ports {B[6]}]
144 set_property PACKAGE_PIN W9 [get_ports {BCOUT[12]}]
145 set_property PACKAGE_PIN P6 [get_ports {C[38]}]
146 set_property PACKAGE_PIN R2 [get_ports {C[19]}]
147 set_property PACKAGE_PIN Y6 [get_ports {D[4]}]
148 set_property PACKAGE_PIN W31 [get_ports {M[13]}]
149 set_property PACKAGE_PIN AA25 [get_ports {OPMODE[3]}]
150 set_property PACKAGE_PIN AB24 [get_ports {OPMODE[2]}]
151 set_property PACKAGE_PIN W33 [get_ports {M[20]}]
152 set_property PACKAGE_PIN Y33 [get_ports {M[15]}]
153 set_property PACKAGE_PIN AF30 [get_ports {P[28]}]
154 set_property PACKAGE_PIN G25 [get_ports {PCIN[45]}]
155 set_property PACKAGE_PIN J28 [get_ports {PCIN[26]}]
156 set_property PACKAGE_PIN AC34 [get_ports {M[5]}]
157 set_property PACKAGE_PIN AA24 [get_ports {OPMODE[4]}]
158 set_property PACKAGE_PIN W1 [get_ports {D[3]}]
159 set_property PACKAGE_PIN U4 [get_ports {C[13]}]
160 set_property PACKAGE_PIN T9 [get_ports {C[1]}]
161 set_property PACKAGE_PIN V28 [get_ports {M[34]}]
162 set_property PACKAGE_PIN T8 [get_ports {C[4]}]
163 set_property PACKAGE_PIN M11 [get_ports {C[42]}]
164 set_property PACKAGE_PIN AF27 [get_ports {P[14]}]
165 set_property PACKAGE_PIN J23 [get_ports {PCIN[47]}]
166 set_property PACKAGE_PIN AA28 [get_ports {P[1]}]
167 set_property PACKAGE_PIN K25 [get_ports {PCIN[44]}]
168 set_property PACKAGE_PIN G27 [get_ports {PCIN[35]}]
169 set_property PACKAGE_PIN H29 [get_ports {PCIN[23]}]
170 set_property PACKAGE_PIN L33 [get_ports {PCIN[2]}]
171 set_property PACKAGE_PIN N33 [get_ports {PCOUT[7]}]
172 set_property PACKAGE_PIN R33 [get_ports {PCOUT[5]}]
173 set_property PACKAGE_PIN R26 [get_ports {PCOUT[46]}]
174 set_property PACKAGE_PIN U25 [get_ports {PCOUT[42]}]
175 set_property PACKAGE_PIN AG2 [get_ports {CEM}]
176 set_property PACKAGE_PIN AH2 [get_ports {RSTC}]
177 set_property PACKAGE_PIN P9 [get_ports {C[40]}]
178 set_property PACKAGE_PIN AA2 [get_ports {A[17]}]
179 set_property PACKAGE_PIN AB6 [get_ports {A[7]}]
180 set_property PACKAGE_PIN AB10 [get_ports {A[2]}]
181 set_property PACKAGE_PIN AF10 [get_ports {B[15]}]
182 set_property PACKAGE_PIN W3 [get_ports {B[11]}]
183 set_property PACKAGE_PIN AB5 [get_ports {B[4]}]
184 set_property PACKAGE_PIN AA3 [get_ports {B[0]}]
185 set_property PACKAGE_PIN N9 [get_ports {BCOUT[3]}]
186 set_property PACKAGE_PIN N6 [get_ports {C[37]}]
187 set_property PACKAGE_PIN U2 [get_ports {C[18]}]
188 set_property PACKAGE_PIN U5 [get_ports {C[14]}]
189 set_property PACKAGE_PIN T2 [get_ports {C[15]}]
190 set_property PACKAGE_PIN U11 [get_ports {C[0]}]
191 set_property PACKAGE_PIN V29 [get_ports {M[33]}]
192 set_property PACKAGE_PIN R7 [get_ports {C[11]}]
193 set_property PACKAGE_PIN R10 [get_ports {C[10]}]
194 set_property PACKAGE_PIN Y28 [get_ports {M[23]}]
195 set_property PACKAGE_PIN AD33 [get_ports {P[46]}]
196 set_property PACKAGE_PIN AE26 [get_ports {P[20]}]
197 set_property PACKAGE_PIN V31 [get_ports {M[22]}]
198 set_property PACKAGE_PIN AC32 [get_ports {M[1]}]
199 set_property PACKAGE_PIN AG31 [get_ports {P[30]}]
200 set_property PACKAGE_PIN AG29 [get_ports {P[27]}]
201 set_property PACKAGE_PIN AC29 [get_ports {P[3]}]
```

```
202 set_property PACKAGE_PIN M25 [get_ports {PCIN[42]}]
203 set_property PACKAGE_PIN AG30 [get_ports {P[26]}]
204 set_property PACKAGE_PIN U10 [get_ports {C[8]}]
205 set_property PACKAGE_PIN Y11 [get_ports {BCOUT[17]}]
206 set_property PACKAGE_PIN G31 [get_ports {PCIN[15]}]
207 set_property PACKAGE_PIN AH27 [get_ports {P[16]}]
208 set_property PACKAGE_PIN G26 [get_ports {PCIN[33]}]
209 set_property PACKAGE_PIN L27 [get_ports {PCIN[32]}]
210 set_property PACKAGE_PIN K33 [get_ports {PCIN[10]}]
211 set_property PACKAGE_PIN K32 [get_ports {PCIN[5]}]
212 set_property PACKAGE_PIN P26 [get_ports {PCOUT[45]}]
213 set_property PACKAGE_PIN R31 [get_ports {PCOUT[16]}]
214 set_property PACKAGE_PIN M31 [get_ports {PCOUT[17]}]
215 set_property PACKAGE_PIN U34 [get_ports {PCOUT[2]}]
216 set_property PACKAGE_PIN M27 [get_ports {PCOUT[43]}]
217 set_property PACKAGE_PIN P24 [get_ports {PCOUT[40]}]
218 set_property PACKAGE_PIN N24 [get_ports {PCOUT[39]}]
219 set_property PACKAGE_PIN N27 [get_ports {PCOUT[32]}]
220 set_property PACKAGE_PIN AE1 [get_ports RSTCARRYIN]
221 set_property PACKAGE_PIN T4 [get_ports {C[21]}]
222 set_property PACKAGE_PIN AC6 [get_ports {A[9]}]
223 set_property PACKAGE_PIN AC8 [get_ports {A[5]}]
224 set_property PACKAGE_PIN AE5 [get_ports {D[11]}]
225 set_property PACKAGE_PIN Y1 [get_ports {D[2]}]
226 set_property PACKAGE_PIN P4 [get_ports {C[28]}]
227 set_property PACKAGE_PIN W30 [get_ports {M[14]}]
228 set_property PACKAGE_PIN W24 [get_ports {OPMODE[0]}]
229 set_property PACKAGE_PIN AC26 [get_ports {P[19]}]
230 set_property PACKAGE_PIN Y30 [get_ports {M[12]}]
231 set_property PACKAGE_PIN AC33 [get_ports {M[6]}]
232 set_property PACKAGE_PIN AA30 [get_ports {M[10]}]
233 set_property PACKAGE_PIN AF23 [get_ports {P[10]}]
234 set_property PACKAGE_PIN L28 [get_ports {PCIN[28]}]
235 set_property PACKAGE_PIN AE28 [get_ports {P[23]}]
236 set_property PACKAGE_PIN Y31 [get_ports {M[11]}]
237 set_property PACKAGE_PIN V2 [get_ports {D[1]}]
238 set_property PACKAGE_PIN T10 [get_ports {C[7]}]
239 set_property PACKAGE_PIN U9 [get_ports {C[2]}]
240 set_property PACKAGE_PIN AH4 [get_ports {D[17]}]
241 set_property PACKAGE_PIN V6 [get_ports {BCOUT[9]}]
242 set_property PACKAGE_PIN Y8 [get_ports {BCOUT[8]}]
243 set_property PACKAGE_PIN M9 [get_ports {BCOUT[2]}]
244 set_property PACKAGE_PIN H33 [get_ports {PCIN[8]}]
245 set_property PACKAGE_PIN AG24 [get_ports {P[9]}]
246 set_property PACKAGE_PIN J25 [get_ports {PCIN[43]}]
247 set_property PACKAGE_PIN N29 [get_ports {PCOUT[28]}]
248 set_property PACKAGE_PIN N32 [get_ports {PCOUT[8]}]
249 set_property PACKAGE_PIN M32 [get_ports {PCOUT[13]}]
250 set_property PACKAGE_PIN J24 [get_ports {PCIN[38]}]
251 set_property PACKAGE_PIN K27 [get_ports {PCIN[31]}]
252 set_property PACKAGE_PIN J30 [get_ports {PCIN[21]}]
253 set_property PACKAGE_PIN N26 [get_ports {PCOUT[44]}]
254 set_property PACKAGE_PIN T25 [get_ports {PCOUT[41]}]
255 set_property PACKAGE_PIN U27 [get_ports {PCOUT[37]}]
256 set_property PACKAGE_PIN T29 [get_ports {PCOUT[25]}]
257 set_property PACKAGE_PIN AF24 [get_ports {C[43]}]
258 set_property PACKAGE_PIN Y10 [get_ports {BCOUT[15]}]
259 set_property PACKAGE_PIN AA10 [get_ports {A[4]}]
260 set_property PACKAGE_PIN AA8 [get_ports {A[12]}]
261 set_property PACKAGE_PIN AC9 [get_ports {A[6]}]
262 set_property PACKAGE_PIN N1 [get_ports {C[36]}]
263 set_property PACKAGE_PIN R1 [get_ports {C[32]}]
264 set_property PACKAGE_PIN T5 [get_ports {C[22]}]
265 set_property PACKAGE_PIN W29 [get_ports {M[27]}]
266 set_property PACKAGE_PIN Y26 [get_ports {M[29]}]
267 set_property PACKAGE_PIN P10 [get_ports {C[9]}]
268 set_property PACKAGE_PIN AE3 [get_ports {D[14]}]
```

```
268 set_property PACKAGE_PIN AE3 [get_ports {D[14]}]
269 set_property PACKAGE_PIN P3 [get_ports {C[27]}]
270 set_property PACKAGE_PIN AB30 [get_ports {M[9]}]
271 set_property PACKAGE_PIN AC31 [get_ports {M[2]}]
272 set_property PACKAGE_PIN AB27 [get_ports {OPMODE[7]}]
273 set_property PACKAGE_PIN AB29 [get_ports {OPMODE[5]}]
274 set_property PACKAGE_PIN K30 [get_ports {PCIN[22]}]
275 set_property PACKAGE_PIN J34 [get_ports {PCIN[9]}]
276 set_property PACKAGE_PIN AH33 [get_ports {P[44]}]
277 set_property PACKAGE_PIN AA32 [get_ports {M[4]}]
278 set_property PACKAGE_PIN AG6 [get_ports {D[13]}]
279 set_property PACKAGE_PIN V9 [get_ports {BCOUT[14]}]
280 set_property PACKAGE_PIN P8 [get_ports {C[39]}]
281 set_property PACKAGE_PIN Y7 [get_ports {BCOUT[7]}]
282 set_property PACKAGE_PIN AE27 [get_ports {P[15]}]
283 set_property PACKAGE_PIN AH26 [get_ports {P[12]}]
284 set_property PACKAGE_PIN R30 [get_ports {PCOUT[22]}]
285 set_property PACKAGE_PIN AE23 [get_ports {P[11]}]
286 set_property PACKAGE_PIN U24 [get_ports {PCIN[39]}]
287 set_property PACKAGE_PIN K26 [get_ports {PCIN[30]}]
288 set_property PACKAGE_PIN H31 [get_ports {PCIN[16]}]
289 set_property PACKAGE_PIN U26 [get_ports {PCOUT[38]}]
290 set_property PACKAGE_PIN R25 [get_ports {PCOUT[36]}]
291 set_property PACKAGE_PIN P25 [get_ports {PCOUT[35]}]
292 set_property PACKAGE_PIN U29 [get_ports {PCOUT[26]}]
293 set_property PACKAGE_PIN U30 [get_ports {PCOUT[20]}]
294 set_property PACKAGE_PIN AG1 [get_ports RSTOPMODE]
295 set_property PACKAGE_PIN AE2 [get_ports RSTA]
296 set_property PACKAGE_PIN AH3 [get_ports CED]
297 set_property PACKAGE_PIN AD5 [get_ports CEA]
298 set_property PACKAGE_PIN AB9 [get_ports {A[1]}]
299 set_property PACKAGE_PIN AB7 [get_ports {A[8]}]
300 set_property PACKAGE_PIN AA9 [get_ports {A[3]}]
301 set_property PACKAGE_PIN AB11 [get_ports {A[0]}]
302 set_property PACKAGE_PIN W4 [get_ports {B[9]}]
303 set_property PACKAGE_PIN M1 [get_ports {C[35]}]
304 set_property PACKAGE_PIN W25 [get_ports {M[26]}]
305 set_property PACKAGE_PIN P5 [get_ports {C[26]}]
306 set_property PACKAGE_PIN AA33 [get_ports {M[3]}]
307 set_property PACKAGE_PIN AG32 [get_ports {P[40]}]
308 set_property PACKAGE_PIN AD30 [get_ports {P[34]}]
309 set_property PACKAGE_PIN AD34 [get_ports {P[45]}]
310 set_property PACKAGE_PIN AE31 [get_ports {P[35]}]
311 set_property PACKAGE_PIN AF32 [get_ports {P[37]}]
312 set_property PACKAGE_PIN AC24 [get_ports {P[7]}]
313 set_property PACKAGE_PIN J33 [get_ports {PCIN[4]}]
314 set_property PACKAGE_PIN AF28 [get_ports {P[22]}]
315 set_property PACKAGE_PIN AH34 [get_ports {P[43]}]
316 set_property PACKAGE_PIN V1 [get_ports {D[0]}]
317 set_property PACKAGE_PIN U1 [get_ports {C[17]}]
318 set_property PACKAGE_PIN AF7 [get_ports {D[7]}]
319 set_property PACKAGE_PIN W6 [get_ports {D[5]}]
320 set_property PACKAGE_PIN AD3 [get_ports {D[15]}]
321 set_property PACKAGE_PIN N4 [get_ports {C[25]}]
322 set_property PACKAGE_PIN AG25 [get_ports {C[46]}]
323 set_property PACKAGE_PIN M7 [get_ports {BCOUT[5]}]
324 set_property PACKAGE_PIN M10 [get_ports {C[41]}]
325 set_property PACKAGE_PIN AB34 [get_ports {P[5]}]
326 set_property PACKAGE_PIN AD24 [get_ports {P[6]}]
327 set_property PACKAGE_PIN G24 [get_ports {PCIN[46]}]
328 set_property PACKAGE_PIN N31 [get_ports {PCOUT[14]}]
329 set_property PACKAGE_PIN U31 [get_ports {PCOUT[12]}]
330 set_property PACKAGE_PIN T32 [get_ports {PCOUT[40]}]
331 set_property PACKAGE_PIN P33 [get_ports {PCOUT[0]}]
332 set_property PACKAGE_PIN J26 [get_ports {PCIN[29]}]
333 set_property PACKAGE_PIN L30 [get_ports {PCIN[13]}]
334 set_property PACKAGE_PIN G32 [get_ports {PCIN[11]}]
335 set_property PACKAGE_PIN L34 [get_ports {PCIN[1]}]
336 set_property PACKAGE_PIN T28 [get_ports {PCOUT[30]}]
337 set_property PACKAGE_PIN P31 [get_ports {PCOUT[15]}]
338 set_property PACKAGE_PIN AF34 [get_ports CARRYOUT]
339 set_property PACKAGE_PIN AG4 [get_ports CEB]
340 set_property PACKAGE_PIN A31 [get_ports RSTB]
341 set_property PACKAGE_PIN AG9 [get_ports {BCIN[4]}]
342 set_property PACKAGE_PIN AH7 [get_ports {BCIN[13]}]
343 set_property PACKAGE_PIN AF8 [get_ports {BCIN[10]}]
344 set_property PACKAGE_PIN AG10 [get_ports {BCIN[5]}]
345 set_property PACKAGE_PIN AF12 [get_ports {BCIN[7]}]
346 set_property PACKAGE_PIN AH11 [get_ports {BCIN[0]}]
347 set_property PACKAGE_PIN AD9 [get_ports {BCIN[17]}]
348 set_property PACKAGE_PIN AD8 [get_ports {BCIN[16]}]
349 set_property PACKAGE_PIN AE7 [get_ports {BCIN[14]}]
350 set_property PACKAGE_PIN AH6 [get_ports {BCIN[12]}]
351 set_property PACKAGE_PIN AE8 [get_ports {BCIN[15]}]
352 set_property PACKAGE_PIN AG12 [get_ports {BCIN[6]}]
353 set_property PACKAGE_PIN AD11 [get_ports {BCIN[3]}]
354 set_property PACKAGE_PIN AD4 [get_ports CARRYIN]
355 set_property PACKAGE_PIN AF9 [get_ports {BCIN[11]}]
356 set_property PACKAGE_PIN AE11 [get_ports {BCIN[2]}]
357 set_property PACKAGE_PIN AH8 [get_ports {BCIN[8]}]
358 set_property PACKAGE_PIN AH9 [get_ports {BCIN[9]}]
359 set_property PACKAGE_PIN AG11 [get_ports {BCIN[1]}]
360
```

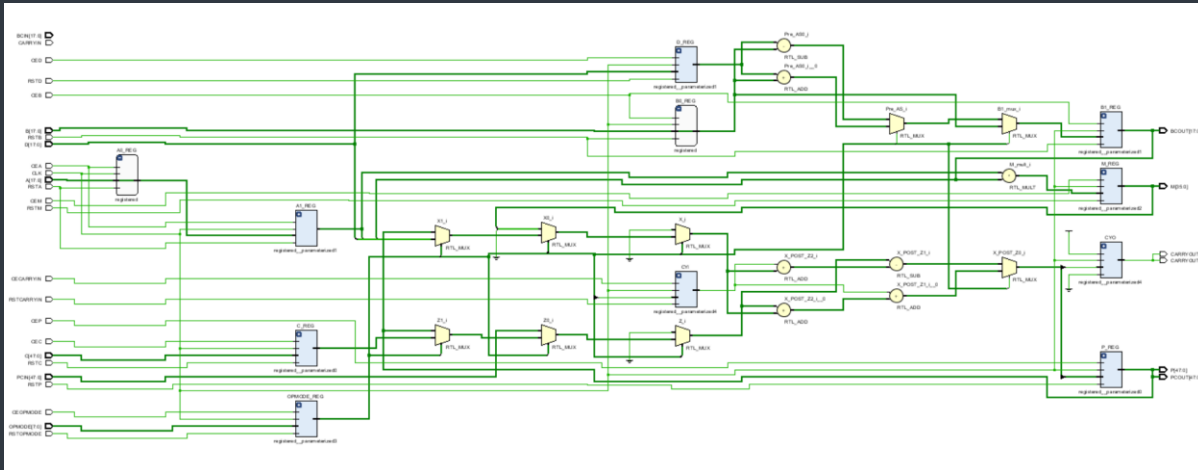

6. Elaboration

i. "Messages" tab

- a. [Synth 8-2490] is because I have chosen the module name similar to another module in the FPGA (DSP48A1)
- b. [Synth 8-3331] the unconnected ports are because based on the parameter B_INPUT B is passed into the module while BCIN is held unconnected. Subsequently BCIN [17:0] is held unconnected. Same for CARRYIN and A0_reg and B0_reg.



ii. Schematic snippets

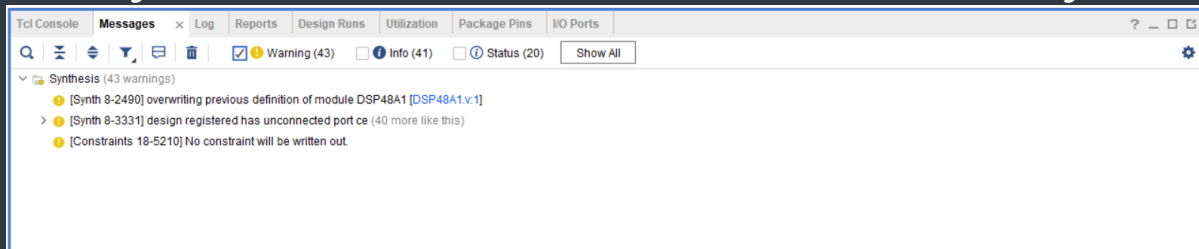


7. Synthesis

i. "Messages" tab

[Constraints 18-5210] This message is found in Vivado versions 2018.2 through 2019.2.

This message has been removed from the 2020.1 version of Vivado as it was found to be confusing.



ii. Utilization report

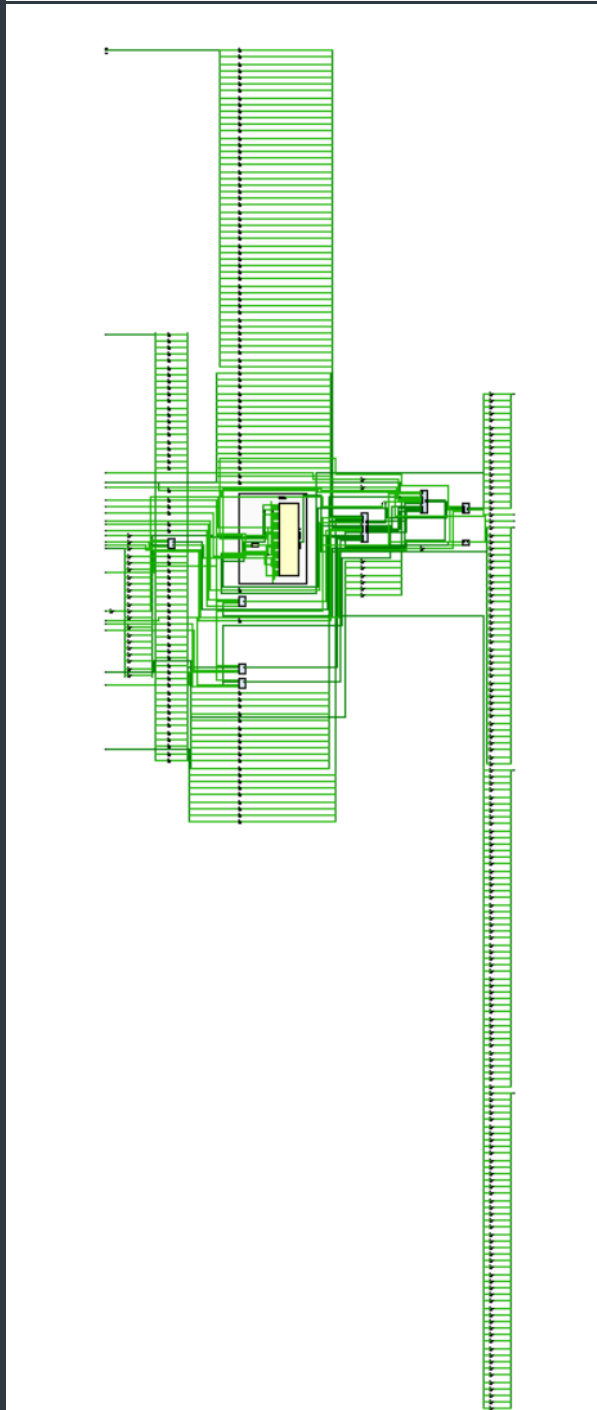
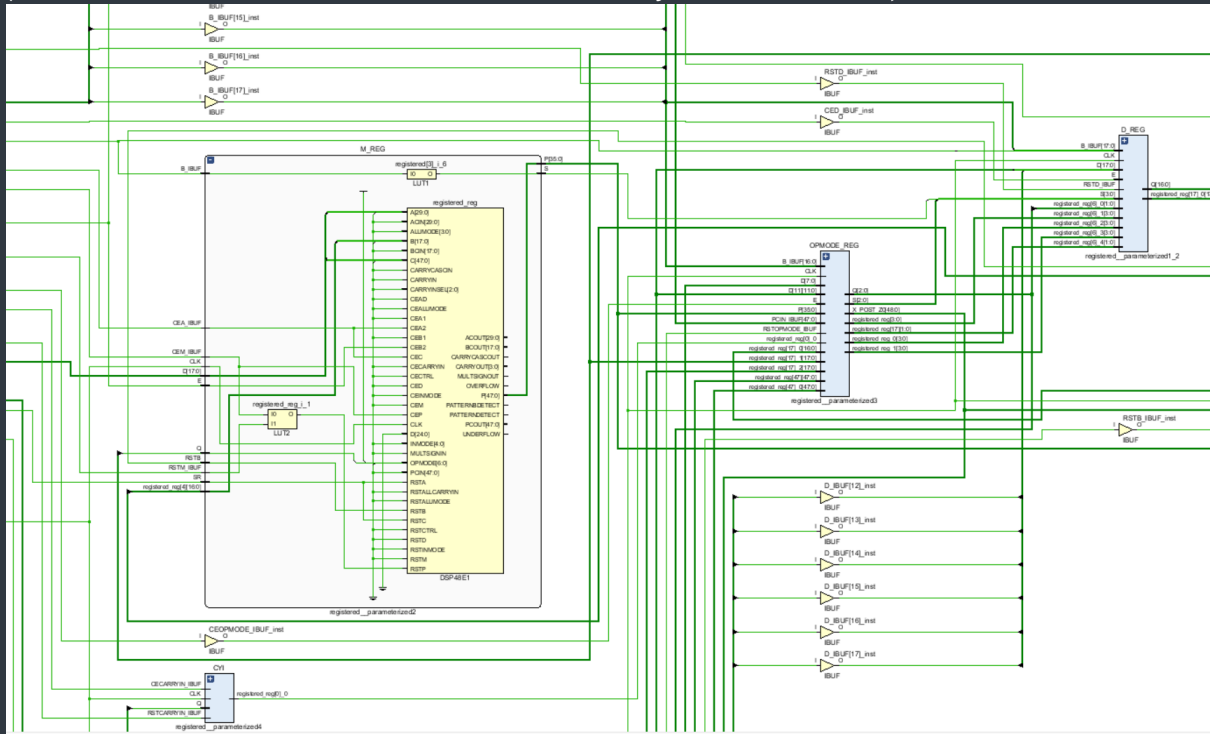
Name	Slice LUTs (134600)	Slice Registers (269200)	DSP s (740)	Bonded IOB (500)	BUFCTRL (32)
D_REG (registered_p_...)	19	18	0	0	0
M_REG (registered_p_...)	2	0	1	0	0
OPMODE_REG (regist...)	211	8	0	0	0
P_REG (registered_p_...)	1	48	0	0	0

iii. Timing report

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.205 ns	Worst Hold Slack (WHS): 0.287 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.

iv. Schematic snippets
(DSP block with other cells from the FPGA should be in the synthesized schematic)



8. Implementation

i. "Messages" tab

Tcl ConsoleMessages xLogReportsDesign RunsPowerDRCMethodologyTiming

Q

Warning (43)

Info (226)

Status (459)

Show All

Synthesis (43 warnings)

- [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v.1]
- [Synth 8-3331] design registered has unconnected port ce (40 more like this)
- [Constraints 18-5210] No constraint will be written out.

ii. Utilization report

Tcl ConsoleMessagesLogReportsDesign RunsPowerDRCMethodologyTimingUtilization x

Q

Hierarchy

Hierarchy

Summary

Slice Logic

- Slice LUTs (<1%)
 - LUT as Logic (<1%)
- Slice Registers (<1%)
 - Register as Flip Flop

Name	Slice LUTs (133800)	Slice Registers (267600)	Slice (3345 0)	LUT as Logic (133800)	LUT Flip Flop Pairs (133800)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP48A1	236	179	107	236	51	1	327	1
A1_REG (registered_...	1	18	6	1	0	0	0	0
B1_REG (registered_...	1	36	18	1	0	0	0	0
C_REG (registered_p...	1	48	23	1	0	0	0	0

iii. Timing report

Tcl ConsoleMessagesLogReportsDesign RunsPowerDRCMethodologyTiming xUtilization

Q

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
Worst Negative Slack (WNS): 3.796 ns	Worst Hold Slack (WHS): 0.261 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.

iv. Device snippets

DSP Slice

