



SPARTAN6 - DSP48A1

Project 1



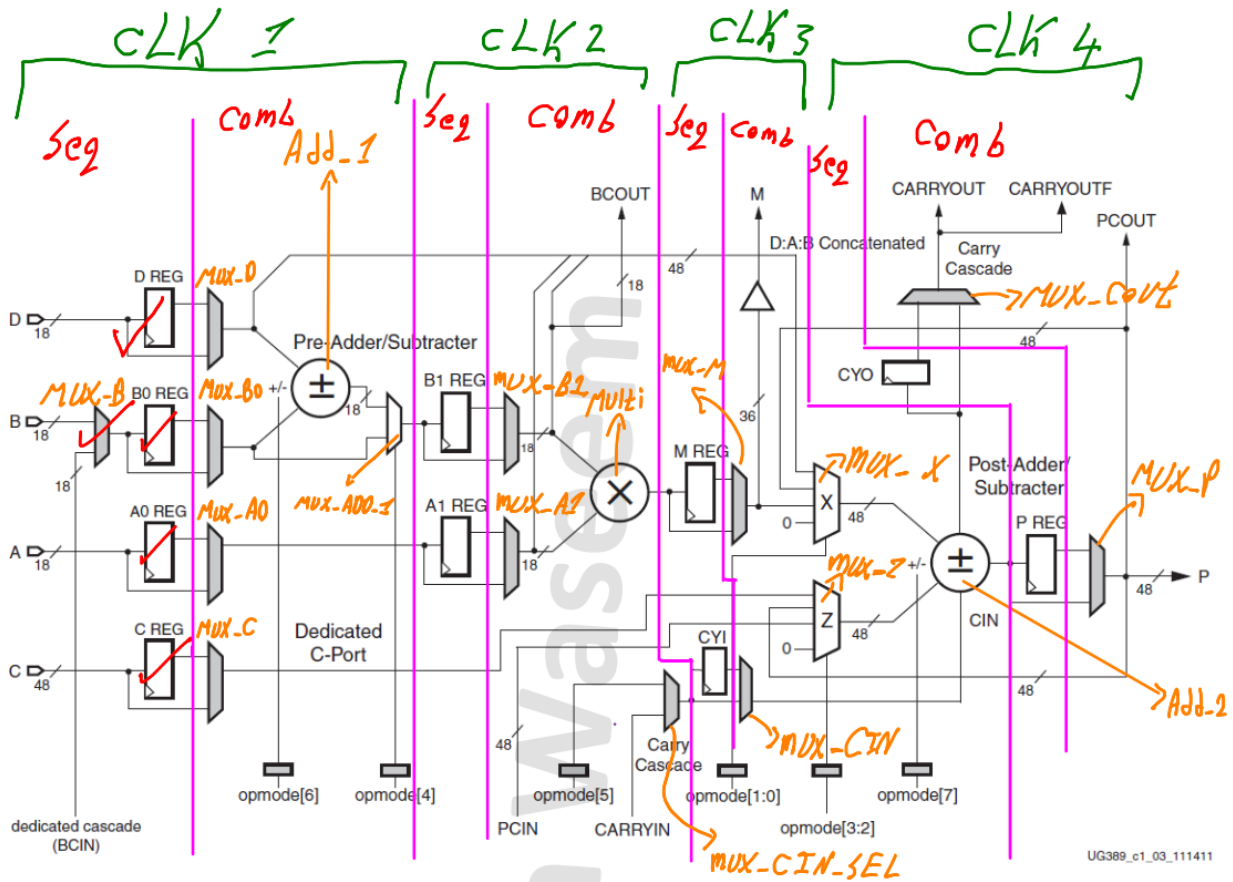
Under Supervision of:

ENG. Kareem Waseem

ENG. Mohamed Khaled

Submitted by: Ahmed Belal

- Annotated Project Schematic:



- Main Module Code:

```

1  module Project_1(clk,A,B,C,D,OPMODE,BCIN,CARRYIN,RSTA,RSTB,RSTM ,RSTP,RSTC,
2  RSTD ,RSTCARRYIN ,RSTOPMODE ,CEA ,CEB,CEM ,CEP ,CEC ,CED ,CECARRYIN,CEOPMODE,PCIN,
3  BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF);
4
5  parameter A0REG = 0 ;
6  parameter A1REG = 1 ;
7  parameter B0REG = 0 ;
8  parameter B1REG = 1 ;
9  parameter CREG = 1 ;
10 parameter DREG = 1 ;
11 parameter MREG = 1 ;
12 parameter PREG = 1 ;
13 parameter CARRYINREG = 1 ;
14 parameter CARRYOUTREG = 1 ;
15 parameter OPMODEREG = 1 ;
16 parameter CARRYINSEL = "OPMODE5" ; // CARRYIN or OPMODE5
17 parameter B_INPUT = "DIRECT" ; // DIRECT or CASCADE
18 parameter RSTTYPE = "SYNC" ; // ASYNC or SYNC
19
20 input [17:0] A,B,D,BCIN ;
21 input [47:0] C,PCIN ;
22 input [7:0] OPMODE ;
23 input CARRYIN,clk,CEA,CEB,CEC,CECARRYIN,
24 | CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,
25 | RSTM,RSTOPMODE,RSTP ;
26
27 output reg [47:0] P,PCOUT ;
28 output reg CARRYOUT,CARRYOUTF;
29 output reg [17:0] BCOUT;
30 output reg [35:0] M ;
31
32 wire [17:0] mux_B ;
33 reg [17:0] A0_reg,A1_reg,B0_reg,B1_reg,D_reg,mux_A0,mux_A1,mux_B0,mux_B1,mux_D,mux_ADD_1 ;
34 reg [47:0] C_reg,P_reg,mux_C,mux_P,mux_X,mux_Z ;
35 reg [35:0] M_reg,mux_M ;
36 reg [7:0] OPMODE_reg ;
37 reg CARRYIN_reg,CARRYOUT_reg,mux_CIN,mux_COUT,mux_CIN_SEL ;
38 reg [17:0] ADD_1 ; // The output of Pre-Adder/Subtractor
39 reg [35:0] MULTI ; // The output of Multiplication process
40 reg [48:0] ADD_2 ; /* The output of Post-Adder/Subtractor but it is used
41 | | | | | 49 bit to take the ADD_2[48] to give it to CARRYOUT*/
42
43
44 assign mux_B = (B_INPUT == "DIRECT") ? B :
45 | | | (B_INPUT == "CASCADE") ? BCIN : 18'b0 ; // B first selection
46
47

```

```

48 //FIRST Sequential Always Block of clk_1
49 generate
50 if (RSTTYPE == "ASYNC") begin
51     always @(posedge clk or posedge RSTA or posedge RSTB or posedge RSTC or posedge RSTD or posedge RSTOPMODE) begin
52         if (RSTA) A0_reg <= 0;
53         else if (CEA) A0_reg <= A;
54
55         if (RSTB) B0_reg <= 0;
56         else if (CEB) B0_reg <= mux_B;
57
58         if (RSTC) C_reg <= 0;
59         else if (CEC) C_reg <= C;
60
61         if (RSTD) D_reg <= 0;
62         else if (CED) D_reg <= D;
63
64         if (RSTOPMODE) OPMODE_reg <= 0;
65         else if (CEOPMODE) OPMODE_reg <= OPMODE;
66     end
67 end else if (RSTTYPE == "SYNC") begin
68     always @(posedge clk) begin
69         if (RSTA) A0_reg <= 0;
70         else if (CEA) A0_reg <= A;
71
72         if (RSTB) B0_reg <= 0;
73         else if (CEB) B0_reg <= mux_B;
74
75         if (RSTC) C_reg <= 0;
76         else if (CEC) C_reg <= C;
77
78         if (RSTD) D_reg <= 0;
79         else if (CED) D_reg <= D;
80
81         if (RSTOPMODE) OPMODE_reg <= 0;
82         else if (CEOPMODE) OPMODE_reg <= OPMODE;
83     end
84 end
85 endgenerate
86
87

```

```

128 //FIRST Combinational Always Block of clk_1
129 always @(*) begin
130
131     // mux_A0
132     if(A0REG)
133         mux_A0 = A0_reg ;
134     else
135         mux_A0 = A ;
136
137     // mux_B0
138     if(B0REG)
139         mux_B0 = B0_reg ;
140     else
141         mux_B0 = mux_B ;
142
143     // mux_C
144     if(CREG)
145         mux_C = C_reg ;
146     else
147         mux_C = C ;
148
149     // mux_D
150     if(DREG)
151         mux_D = D_reg ;
152     else
153         mux_D = D ;
154
155     // Pre-Adder/Subtractor Output (ADD_1)
156     if (OPMODE_reg[6])
157         ADD_1 = mux_D - mux_B0 ;
158     else
159         ADD_1 = mux_D + mux_B0 ;
160
161     // Mux of OPMODE[4] that choose bet. Pre-Adder/Subtractor OR Mux_B0
162     if (OPMODE_reg[4])
163         mux_ADD_1 = ADD_1 ;
164     else
165         mux_ADD_1 = mux_B0 ;
166
167 end
168
169
170

```

```

130 //SECOND Sequential Always Block of clk_2
131 generate
132 if (RSTTYPE == "ASYNC") begin
133     always @(posedge clk or posedge RSTA or posedge RSTB) begin
134         if (RSTA) A1_reg <= 0;
135         else if (CEA) A1_reg <= mux_A0;
136
137         if (RSTB) B1_reg <= 0;
138         else if (CEB) B1_reg <= mux_ADD_1;
139     end
140 end else if (RSTTYPE == "SYNC") begin
141     always @(posedge clk) begin
142         if (RSTA) A1_reg <= 0;
143         else if (CEA) A1_reg <= mux_A0;
144
145         if (RSTB) B1_reg <= 0;
146         else if (CEB) B1_reg <= mux_ADD_1;
147     end
148 end
149 endgenerate
150
151
152 //SECOND Combinational Always Block of clk_2
153 always @(*) begin
154
155     // mux_A1
156     if (A1REG)
157         mux_A1 = A1_reg ;
158     else
159         mux_A1 = mux_A0 ;
160
161     // mux_B1
162     if (B1REG)
163         mux_B1 = B1_reg ;
164     else
165         mux_B1 = mux_ADD_1 ;
166
167     // BCOUT from mux_B1
168     BCOUT = mux_B1 ;
169
170     // Multiplication Process
171     MULTI = mux_A1 * mux_B1 ;
172
173     // Carry Cascade Input Mux
174     if (CARRYINSEL == "OPMODE5")
175         mux_CIN_SEL = OPMODE_reg[5] ;
176     else if (CARRYINSEL == "CARRYIN")
177         mux_CIN_SEL = CARRYIN ;
178     else
179         mux_CIN_SEL = 0 ;
180
181 end
182
183

```

```

184 //THIRD Sequential Always Block of clk_3
185 generate
186 if (RSTTYPE == "ASYNC") begin
187     always @(posedge clk or posedge RSTM or posedge RSTCARRYIN) begin
188         if (RSTM) M_reg <= 0;
189         else if (CEM) M_reg <= MULTI;
190
191         if (RSTCARRYIN) CARRYIN_reg <= 0;
192         else if (CECARRYIN) CARRYIN_reg <= mux_CIN_SEL;
193     end
194 end else if (RSTTYPE == "SYNC") begin
195     always @(posedge clk) begin
196         if (RSTM) M_reg <= 0;
197         else if (CEM) M_reg <= MULTI;
198
199         if (RSTCARRYIN) CARRYIN_reg <= 0;
200         else if (CECARRYIN) CARRYIN_reg <= mux_CIN_SEL;
201     end
202 end
203 endgenerate
204
205
206 //THIRD Combinational Always Block of clk_3
207 always @(*) begin
208
209     // mux_M
210     if (MREG)
211         mux_M = M_reg ;
212     else
213         mux_M = MULTI ;
214
215     // mux_CIN
216     if (CARRYINREG)
217         mux_CIN = CARRYIN_reg ;
218     else
219         mux_CIN = mux_CIN_SEL ;
220
221     // M Output
222     M = mux_M ;
223
224     // mux_X
225     case (OPMODE_reg[1:0])
226         2'b00: mux_X = 0 ;
227         2'b01: mux_X = mux_M ;
228         2'b10: mux_X = mux_P ;
229         2'b11: mux_X = {mux_D[11:0] , mux_A1[17:0] , mux_B1[17:0]} ;
230     endcase
231
232     // mux_Z
233     case (OPMODE_reg[3:2])
234         2'b00: mux_Z = 0 ;
235         2'b01: mux_Z = PCIN ;
236         2'b10: mux_Z = mux_P ;
237         2'b11: mux_Z = mux_C ;
238     endcase
239
240     // Post-Adder/Subtractor Output (ADD_2)
241     if (OPMODE_reg[7])
242         ADD_2 = mux_Z - (mux_X + mux_CIN) ;
243     else
244         ADD_2 = mux_Z + mux_X + mux_CIN ;
245
246 end
247
248

```

```

249 //FORTH Sequential Always Block of clk_4
250 generate
251 if (RSTTYPE == "ASYNC") begin
252     always @(posedge clk or posedge RSTP or posedge RSTCARRYIN) begin
253         if (RSTP) P_reg <= 0;
254         else if (CEP) P_reg <= ADD_2[47:0];
255
256         if (RSTCARRYIN) CARRYOUT_reg <= 0;
257         else if (CECARRYIN) CARRYOUT_reg <= ADD_2[48];
258     end
259 end else if (RSTTYPE == "SYNC") begin
260     always @(posedge clk) begin
261         if (RSTP) P_reg <= 0;
262         else if (CEP) P_reg <= ADD_2[47:0];
263
264         if (RSTCARRYIN) CARRYOUT_reg <= 0;
265         else if (CECARRYIN) CARRYOUT_reg <= ADD_2[48];
266     end
267 end
268 endgenerate
269
270
271 //FORTH Combinational Always Block of clk_4
272 always @(*) begin
273
274     // Carry Cascade Output Mux
275     if (CARRYOUTREG)
276         mux_COUT = CARRYOUT_reg ;
277     else
278         mux_COUT = ADD_2[48] ;
279
280     // CARRYOUT & CARRYOUTF Output
281     CARRYOUT = mux_COUT ;
282     CARRYOUTF = mux_COUT ;
283
284     // mux_P
285     if(PREG)
286         mux_P = P_reg ;
287     else
288         mux_P = ADD_2[47:0] ;
289
290     // P & PCOUT Output
291     P = mux_P ;
292     PCOUT = mux_P ;
293
294 end
295
296 endmodule

```


- Test Bench Code:

```

1  module Project_1_tb();
2
3
4  reg [17:0] A,B,D,BCIN ;
5  reg [47:0] C,PCIN ;
6  reg [7:0] OPMODE ;
7  reg CARRYIN,clk,CEA,CEB,CEC,CECARRYIN,
8      CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,
9      RSTM,RSTOPMODE,RSTP ;
10
11 wire [47:0] P,PCOUT ;
12 wire CARRYOUT,CARRYOUTF;
13 wire [17:0] BCOUT;
14 wire [35:0] M ;
15
16 Project_1 #(
17     .A0REG(0),
18     .A1REG(1),
19     .B0REG(0),
20     .B1REG(1),
21     .CREG(1),
22     .DREG(1),
23     .MREG(1),
24     .PREG(1),
25     .CARRYINREG(1),
26     .CARRYOUTREG(1),
27     .OPMODEREG(1),
28     .CARRYINSEL("OPMODE5"),
29     .B_INPUT("DIRECT"),
30     .RSTTYPE("SYNC")
31 )tb ( clk,A,B,C,D,OPMODE,BCIN,CARRYIN,RSTA,RSTB,RSTM,RSTP,RSTC,
32      RSTD,RSTCARRYIN,RSTOPMODE,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,
33      CEOPMODE,PCIN,BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF
34 );
35
36
37
38 initial begin
39     clk = 0 ;
40     forever begin
41         #1 clk = ~clk ;
42     end
43 end
44
45 initial begin
46

```

```

47 //2.1. Verify Reset Operation
48 RSTA = 1 ;
49 RSTB = 1 ;
50 RSTC = 1 ;
51 RSTD = 1 ;
52 RSTCARRYIN = 1 ;
53 RSTOPMODE = 1 ;
54 RSTM = 1 ;
55 RSTP = 1 ;
56
57 CEA = 0 ;
58 CEB = 0 ;
59 CEC = 0 ;
60 CED = 0 ;
61 CEP = 0 ;
62 CEM = 0 ;
63 CECARRYIN = 0 ;
64 CEOPMODE = 0 ;
65
66 A = $random ;
67 B = $random ;
68 C = $random ;
69 D = $random ;
70 CARRYIN = $random ;
71 OPMODE = $random ;
72 PCIN = $random ;
73 BCIN = $random ;
74
75 @(negedge clk) ;
76
77 if (M != 0 || P != 0 || CARRYOUT != 0 || CARRYOUTF != 0 || BCOUT != 0 || PCOUT != 0) begin
78     $display ("error in 2.1. Verify Reset Operation") ;
79     $stop ;
80 end
81
82
83

```

```

84 //2.2. Verify DSP Path 1
85 RSTA = 0 ;
86 RSTB = 0 ;
87 RSTC = 0 ;
88 RSTD = 0 ;
89 RSTCARRYIN = 0 ;
90 RSTOPMODE = 0 ;
91 RSTM = 0 ;
92 RSTP = 0 ;
93
94 CEA = 1 ;
95 CEB = 1 ;
96 CEC = 1 ;
97 CED = 1 ;
98 CEP = 1 ;
99 CEM = 1 ;
100 CECARRYIN = 1 ;
101 CEOPMODE = 1 ;
102
103 OPMODE = 8'b11011101 ;
104 A = 20 ;
105 B = 10 ;
106 C = 350 ;
107 D = 25 ;
108 BCIN = $random ;
109 PCIN = $random ;
110 CARRYIN = $random ;
111
112 repeat(4) @(negedge clk) ;
113
114 if (M != 36'h12c || P != 48'h32 || CARRYOUT != 0 || CARRYOUTF != 0 || BCOUT != 18'hf || PCOUT != 48'h32) begin
115     $display ("error in 2.2. Verify DSP Path 1") ;
116     $stop ;
117 end
118
119
120

```

```

121 //2.3. Verify DSP Path 2
122 OPMODE = 8'b00010000 ;
123 A = 20 ;
124 B = 10 ;
125 C = 350 ;
126 D = 25 ;
127 BCIN = $random ;
128 PCIN = $random ;
129 CARRYIN = $random ;
130
131 repeat(3) @(negedge clk) ;
132
133 if (M != 36'h2bc || P != 0 || CARRYOUT != 0 || CARRYOUTF != 0 || BCOUT != 18'h23 || PCOUT != 0) begin
134     $display ("error in 2.3. Verify DSP Path 2") ;
135     $stop ;
136 end
137
138
139

```

```

140 //2.4. Verify DSP Path 3
141 OPMODE = 8'b00001010 ;
142 A = 20 ;
143 B = 10 ;
144 C = 350 ;
145 D = 25 ;
146 BCIN = $random ;
147 PCIN = $random ;
148 CARRYIN = $random ;
149 repeat(3) @(negedge clk) ;
150
151
152 ✓ if (M != 36'hc8 || P != 0 || CARRYOUT != 0 || CARRYOUTF != 0 || BCOUT != 18'ha || PCOUT != 0) begin
153     $display ("error in 2.4. Verify DSP Path 3") ;
154     $stop ;
155 end
156
157
158
159 //2.5. Verify DSP Path 4
160 OPMODE = 8'b10100111 ;
161 A = 5 ;
162 B = 6 ;
163 C = 350 ;
164 D = 25 ;
165 BCIN = $random ;
166 PCIN = 3000 ;
167 CARRYIN = $random ;
168
169 repeat(3) @(negedge clk) ;
170
171 ✓ if (M != 36'h1e || P != 48'hfe6ffec0bb1 || CARRYOUT != 1 || CARRYOUTF != 1 || BCOUT != 18'h6 || PCOUT != 48'hfe6ffec0bb1) begin
172     $display ("error in 2.5. Verify DSP Path 4") ;
173     $stop ;
174 end
175
176 $stop;
177
178 end
179
180 endmodule

```

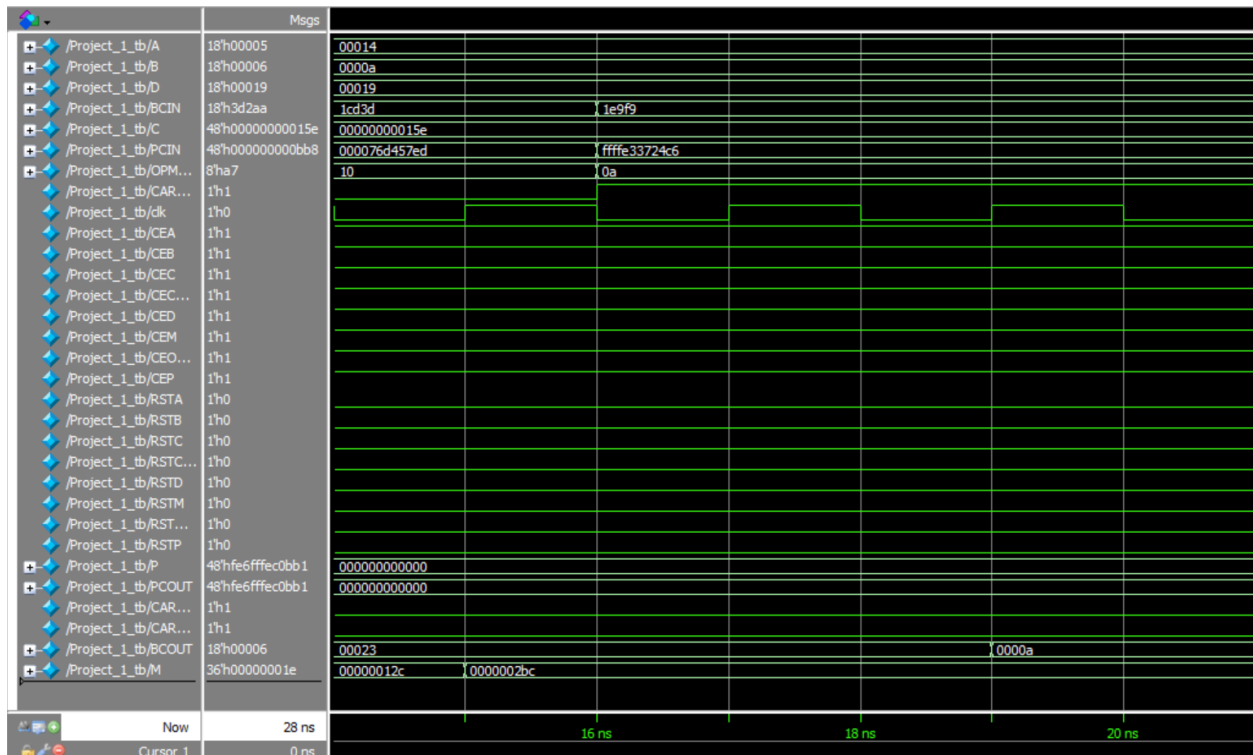
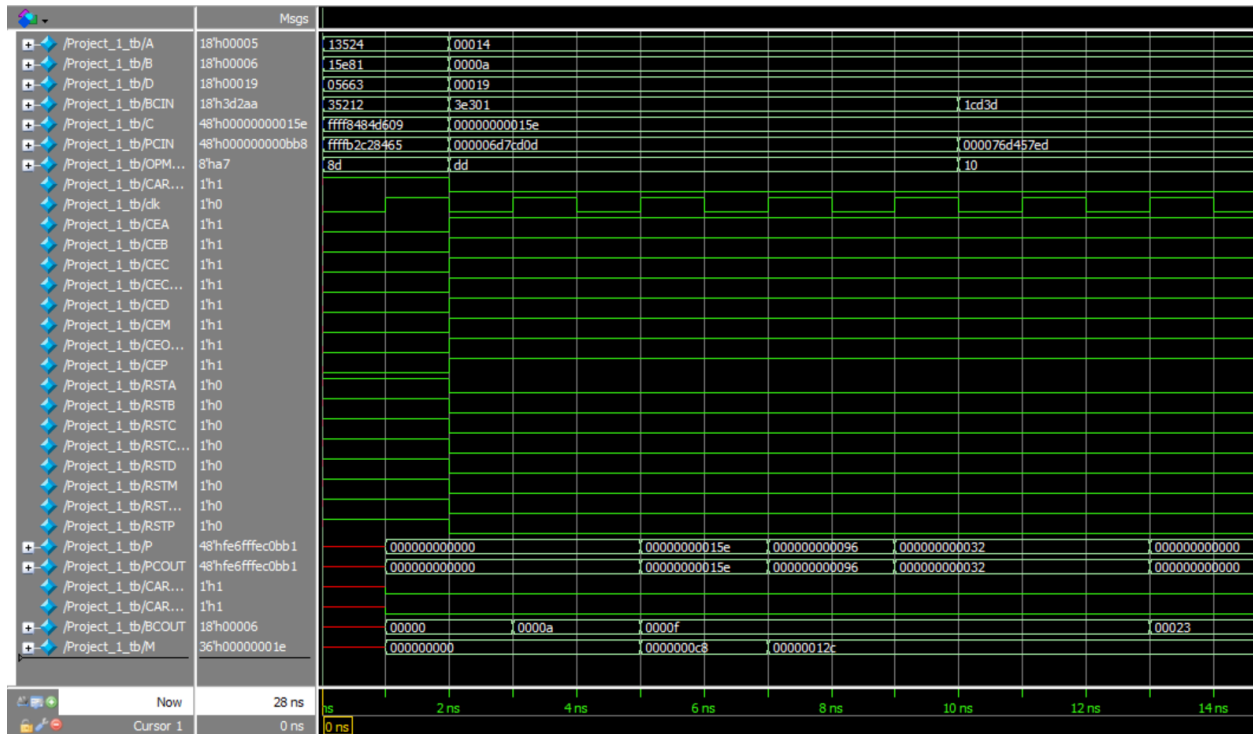
- Do File:

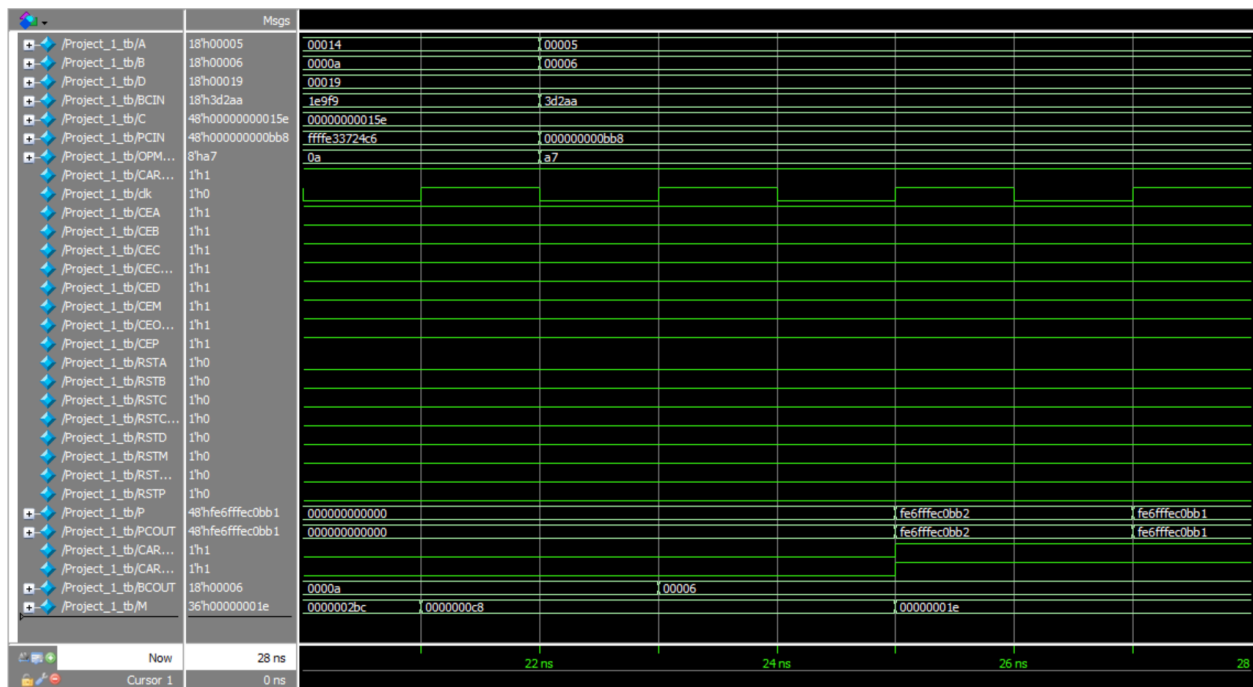
```

vlib work
vlog Project_1.v Project_1_tb.v
vsim -voptargs=+acc work.Project_1_tb
add wave *
run -all
#quit -sim

```

- Questa Sim Snippets:





- Only timing constraint & Debug core constraint:

```

6  ## Clock signal
7  set_property -dict {PACKAGE_PIN W5 IOSTANDARD LVCMOS33} [get_ports clk]
8  create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [get_ports clk]

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_BUF]]
172 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe0]
173 set_property port_width 18 [get_debug_ports u_ila_0/probe0]
174 connect_debug_port u_ila_0/probe0 [get_nets [list {B_IBUF[0]} {B_IBUF[1]} {B_IBUF[2]} {B_IBUF[3]} {B_IBUF[4]} {B_IBUF[5]} {B_IBUF[6]} {B_IBUF[7]} {B_IBUF[8]} {B_IBUF[9]} {B_IBUF[10]} {B_IBUF[11]} {B_IBUF[12]} {B_IBUF[13]} {B_IBUF[14]} {B_IBUF[15]} {B_IBUF[16]} {B_IBUF[17]}]]
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 {D_IBUF[0]} {D_IBUF[1]} {D_IBUF[2]} {D_IBUF[3]} {D_IBUF[4]} {D_IBUF[5]} {D_IBUF[6]} {D_IBUF[7]} {D_IBUF[8]} {D_IBUF[9]} {D_IBUF[10]} {D_IBUF[11]} {D_IBUF[12]} {D_IBUF[13]} {D_IBUF[14]} {D_IBUF[15]} {D_IBUF[16]} {D_IBUF[17]}]]
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 18 [get_debug_ports u_ila_0/probe2]
182 connect_debug_port u_ila_0/probe2 [get_nets [list {BCOUT_OBUF[0]} {BCOUT_OBUF[1]} {BCOUT_OBUF[2]} {BCOUT_OBUF[3]} {BCOUT_OBUF[4]} {BCOUT_OBUF[5]} {BCOUT_OBUF[6]} {BCOUT_OBUF[7]} {BCOUT_OBUF[8]} {BCOUT_OBUF[9]} {BCOUT_OBUF[10]} {BCOUT_OBUF[11]} {BCOUT_OBUF[12]} {BCOUT_OBUF[13]} {BCOUT_OBUF[14]} {BCOUT_OBUF[15]} {BCOUT_OBUF[16]} {BCOUT_OBUF[17]}]]
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 {C_IBUF[0]} {C_IBUF[1]} {C_IBUF[2]} {C_IBUF[3]} {C_IBUF[4]} {C_IBUF[5]} {C_IBUF[6]} {C_IBUF[7]} {C_IBUF[8]} {C_IBUF[9]} {C_IBUF[10]} {C_IBUF[11]} {C_IBUF[12]} {C_IBUF[13]} {C_IBUF[14]} {C_IBUF[15]} {C_IBUF[16]} {C_IBUF[17]} {C_IBUF[18]} {C_IBUF[19]} {C_IBUF[20]} {C_IBUF[21]} {C_IBUF[22]} {C_IBUF[23]} {C_IBUF[24]} {C_IBUF[25]} {C_IBUF[26]} {C_IBUF[27]} {C_IBUF[28]} {C_IBUF[29]} {C_IBUF[30]} {C_IBUF[31]} {C_IBUF[32]} {C_IBUF[33]} {C_IBUF[34]} {C_IBUF[35]} {C_IBUF[36]} {C_IBUF[37]} {C_IBUF[38]} {C_IBUF[39]} {C_IBUF[40]} {C_IBUF[41]} {C_IBUF[42]} {C_IBUF[43]} {C_IBUF[44]} {C_IBUF[45]} {C_IBUF[46]} {C_IBUF[47]}]]
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 18 [get_debug_ports u_ila_0/probe4]
190 connect_debug_port u_ila_0/probe4 [get_nets [list {A_IBUF[0]} {A_IBUF[1]} {A_IBUF[2]} {A_IBUF[3]} {A_IBUF[4]} {A_IBUF[5]} {A_IBUF[6]} {A_IBUF[7]} {A_IBUF[8]} {A_IBUF[9]} {A_IBUF[10]} {A_IBUF[11]} {A_IBUF[12]} {A_IBUF[13]} {A_IBUF[14]} {A_IBUF[15]} {A_IBUF[16]} {A_IBUF[17]}]]
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 36 [get_debug_ports u_ila_0/probe5]
194 connect_debug_port u_ila_0/probe5 [get_nets [list {M_OBUF[0]} {M_OBUF[1]} {M_OBUF[2]} {M_OBUF[3]} {M_OBUF[4]} {M_OBUF[5]} {M_OBUF[6]} {M_OBUF[7]} {M_OBUF[8]} {M_OBUF[9]} {M_OBUF[10]} {M_OBUF[11]} {M_OBUF[12]} {M_OBUF[13]} {M_OBUF[14]} {M_OBUF[15]} {M_OBUF[16]} {M_OBUF[17]} {M_OBUF[18]} {M_OBUF[19]} {M_OBUF[20]} {M_OBUF[21]} {M_OBUF[22]} {M_OBUF[23]} {M_OBUF[24]} {M_OBUF[25]} {M_OBUF[26]} {M_OBUF[27]} {M_OBUF[28]} {M_OBUF[29]} {M_OBUF[30]} {M_OBUF[31]} {M_OBUF[32]} {M_OBUF[33]} {M_OBUF[34]} {M_OBUF[35]}]]
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 48 [get_debug_ports u_ila_0/probe6]
198 connect_debug_port u_ila_0/probe6 [get_nets [list {P_OBUF[0]} {P_OBUF[1]} {P_OBUF[2]} {P_OBUF[3]} {P_OBUF[4]} {P_OBUF[5]} {P_OBUF[6]} {P_OBUF[7]} {P_OBUF[8]} {P_OBUF[9]} {P_OBUF[10]} {P_OBUF[11]} {P_OBUF[12]} {P_OBUF[13]} {P_OBUF[14]} {P_OBUF[15]} {P_OBUF[16]} {P_OBUF[17]} {P_OBUF[18]} {P_OBUF[19]} {P_OBUF[20]} {P_OBUF[21]} {P_OBUF[22]} {P_OBUF[23]} {P_OBUF[24]} {P_OBUF[25]} {P_OBUF[26]} {P_OBUF[27]} {P_OBUF[28]} {P_OBUF[29]} {P_OBUF[30]} {P_OBUF[31]} {P_OBUF[32]} {P_OBUF[33]} {P_OBUF[34]} {P_OBUF[35]} {P_OBUF[36]} {P_OBUF[37]} {P_OBUF[38]} {P_OBUF[39]} {P_OBUF[40]} {P_OBUF[41]} {P_OBUF[42]} {P_OBUF[43]} {P_OBUF[44]} {P_OBUF[45]} {P_OBUF[46]} {P_OBUF[47]}]]
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 8 [get_debug_ports u_ila_0/probe7]
202 connect_debug_port u_ila_0/probe7 [get_nets [list {OPMODE_IBUF[0]} {OPMODE_IBUF[1]} {OPMODE_IBUF[2]} {OPMODE_IBUF[3]} {OPMODE_IBUF[4]} {OPMODE_IBUF[5]} {OPMODE_IBUF[6]} {OPMODE_IBUF[7]}]]
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 48 [get_debug_ports u_ila_0/probe8]
206 connect_debug_port u_ila_0/probe8 [get_nets [list {PCIN_IBUF[0]} {PCIN_IBUF[1]} {PCIN_IBUF[2]} {PCIN_IBUF[3]} {PCIN_IBUF[4]} {PCIN_IBUF[5]} {PCIN_IBUF[6]} {PCIN_IBUF[7]} {PCIN_IBUF[8]} {PCIN_IBUF[9]} {PCIN_IBUF[10]} {PCIN_IBUF[11]} {PCIN_IBUF[12]} {PCIN_IBUF[13]} {PCIN_IBUF[14]} {PCIN_IBUF[15]} {PCIN_IBUF[16]} {PCIN_IBUF[17]} {PCIN_IBUF[18]} {PCIN_IBUF[19]} {PCIN_IBUF[20]} {PCIN_IBUF[21]} {PCIN_IBUF[22]} {PCIN_IBUF[23]} {PCIN_IBUF[24]} {PCIN_IBUF[25]} {PCIN_IBUF[26]} {PCIN_IBUF[27]} {PCIN_IBUF[28]} {PCIN_IBUF[29]} {PCIN_IBUF[30]} {PCIN_IBUF[31]} {PCIN_IBUF[32]} {PCIN_IBUF[33]} {PCIN_IBUF[34]} {PCIN_IBUF[35]} {PCIN_IBUF[36]} {PCIN_IBUF[37]} {PCIN_IBUF[38]} {PCIN_IBUF[39]} {PCIN_IBUF[40]} {PCIN_IBUF[41]} {PCIN_IBUF[42]} {PCIN_IBUF[43]} {PCIN_IBUF[44]} {PCIN_IBUF[45]} {PCIN_IBUF[46]} {PCIN_IBUF[47]}]]

```

```

206 connect_debug_port u_ila_0/probe8 [get_nets [list {PCIN_IBUF[0]} {PCIN_IBUF[1]} {PCIN_IBUF[2]} {PCIN_IBUF[3]} {PCIN_IBUF[4]} {PCIN_IBUF[5]}]
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 CARRYOUTF_OBUF]]
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 CEA_IBUF]]
215 create_debug_port u_ila_0 probe
216 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe11]
217 set_property port_width 1 [get_debug_ports u_ila_0/probe11]
218 connect_debug_port u_ila_0/probe11 [get_nets [list CEB_IBUF]]
219 create_debug_port u_ila_0 probe
220 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe12]
221 set_property port_width 1 [get_debug_ports u_ila_0/probe12]
222 connect_debug_port u_ila_0/probe12 [get_nets [list CEC_IBUF]]
223 create_debug_port u_ila_0 probe
224 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe13]
225 set_property port_width 1 [get_debug_ports u_ila_0/probe13]
226 connect_debug_port u_ila_0/probe13 [get_nets [list CECARRYIN_IBUF]]
227 create_debug_port u_ila_0 probe
228 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe14]
229 set_property port_width 1 [get_debug_ports u_ila_0/probe14]
230 connect_debug_port u_ila_0/probe14 [get_nets [list CED_IBUF]]
231 create_debug_port u_ila_0 probe
232 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe15]
233 set_property port_width 1 [get_debug_ports u_ila_0/probe15]
234 connect_debug_port u_ila_0/probe15 [get_nets [list CEM_IBUF]]
235 create_debug_port u_ila_0 probe
236 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe16]
237 set_property port_width 1 [get_debug_ports u_ila_0/probe16]
238 connect_debug_port u_ila_0/probe16 [get_nets [list CEOPMODE_IBUF]]
239 create_debug_port u_ila_0 probe
240 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe17]
241 set_property port_width 1 [get_debug_ports u_ila_0/probe17]
242 connect_debug_port u_ila_0/probe17 [get_nets [list CEP_IBUF]]
243 create_debug_port u_ila_0 probe
244 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe18]
245 set_property port_width 1 [get_debug_ports u_ila_0/probe18]
246 connect_debug_port u_ila_0/probe18 [get_nets [list clk_IBUF]]
247 create_debug_port u_ila_0 probe
248 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe19]
249 set_property port_width 1 [get_debug_ports u_ila_0/probe19]
250 connect_debug_port u_ila_0/probe19 [get_nets [list RSTA_IBUF]]

```



```

251 create_debug_port u_ila_0 probe
252 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe20]
253 set_property port_width 1 [get_debug_ports u_ila_0/probe20]
254 connect_debug_port u_ila_0/probe20 [get_nets [list RSTB_IBUF]]
255 create_debug_port u_ila_0 probe
256 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe21]
257 set_property port_width 1 [get_debug_ports u_ila_0/probe21]
258 connect_debug_port u_ila_0/probe21 [get_nets [list RSTC_IBUF]]
259 create_debug_port u_ila_0 probe
260 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe22]
261 set_property port_width 1 [get_debug_ports u_ila_0/probe22]
262 connect_debug_port u_ila_0/probe22 [get_nets [list RSTCARRYIN_IBUF]]
263 create_debug_port u_ila_0 probe
264 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe23]
265 set_property port_width 1 [get_debug_ports u_ila_0/probe23]
266 connect_debug_port u_ila_0/probe23 [get_nets [list RSTD_IBUF]]
267 create_debug_port u_ila_0 probe
268 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe24]
269 set_property port_width 1 [get_debug_ports u_ila_0/probe24]
270 connect_debug_port u_ila_0/probe24 [get_nets [list RSTM_IBUF]]
271 create_debug_port u_ila_0 probe
272 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe25]
273 set_property port_width 1 [get_debug_ports u_ila_0/probe25]
274 connect_debug_port u_ila_0/probe25 [get_nets [list RSTOPMODE_IBUF]]
275 create_debug_port u_ila_0 probe
276 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe26]
277 set_property port_width 1 [get_debug_ports u_ila_0/probe26]
278 connect_debug_port u_ila_0/probe26 [get_nets [list RSTP_IBUF]]
279 set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
280 set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
281 set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
282 connect_debug_port dbg_hub/clk [get_nets clk_IBUF_BUFG]

```

➤ **It is important to note that:**

Now I will show the schematic from VIVADO, and when I compared it to the schematic given in the project, I found a difference where the:

Inputs (BCIN, CARRYIN) and Registers (A0REG, B0REG) are not present in the synthesized schematic.

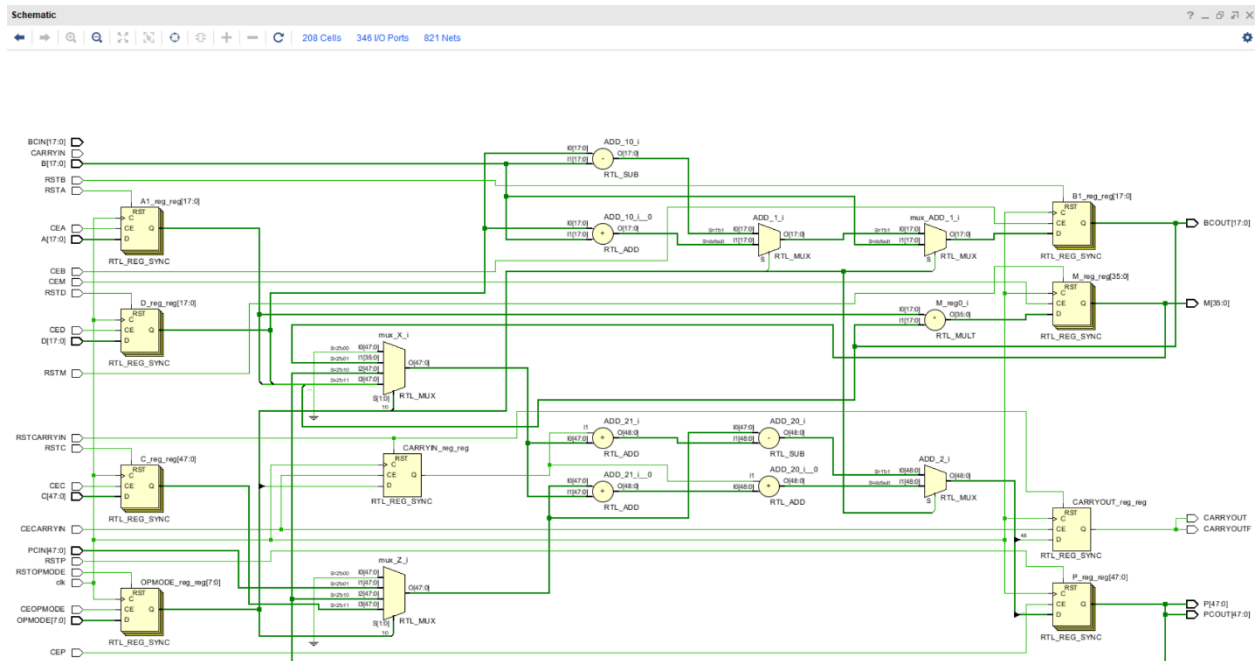
This is due to the fact that the parameters controlling these components were set in a way that **disables their activation** during synthesis.

Specifically, the following parameters were responsible for this behavior:

- B_INPUT = "DIRECT" disables the use of BCIN, so the input is optimized away.
- CARRYINSEL = "OPMODE5" causes the design to rely on OPMODE[5] instead of CARRYIN, making the CARRYIN input unused.
- A0REG = 0 and B0REG = 0 bypass the first stage registers for A and B, resulting in A0_reg and B0_reg not being instantiated or connected.

Since these elements are not functionally required in the current parameter configuration, **VIVADO removes them during synthesis optimization**. This is expected behavior and confirms that the design is functioning correctly according to the defined parameters.

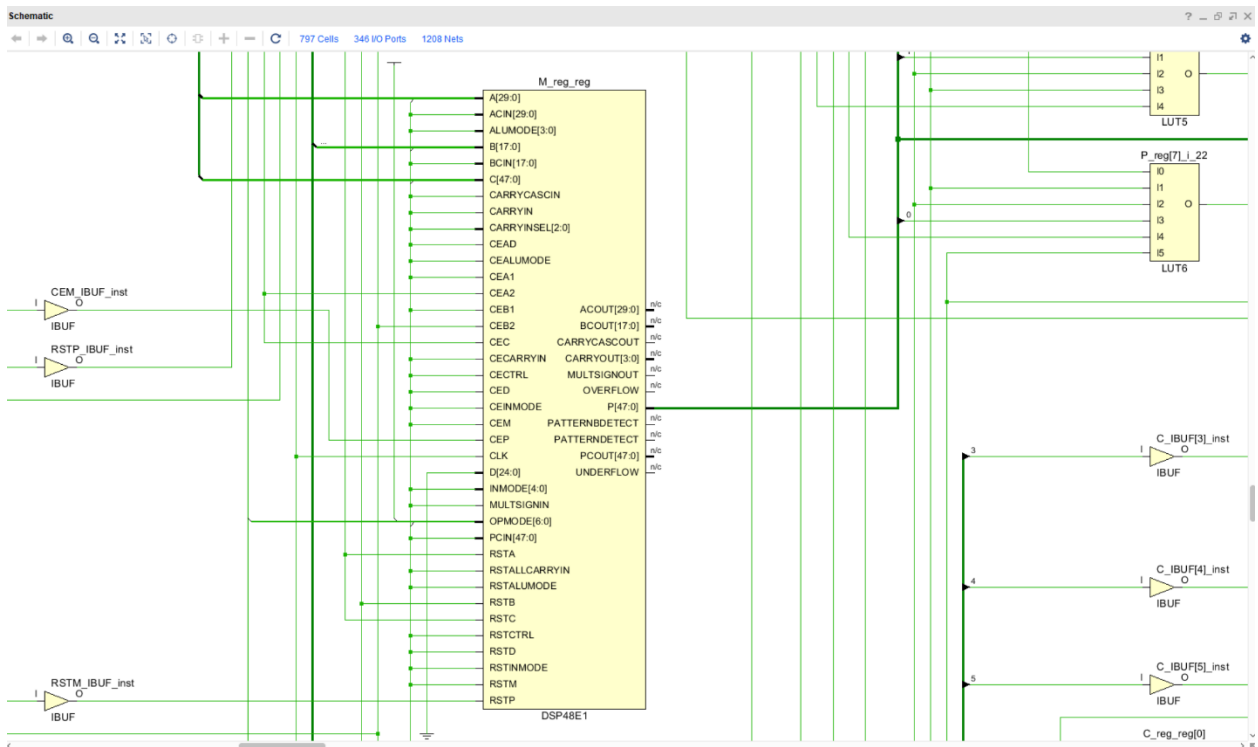
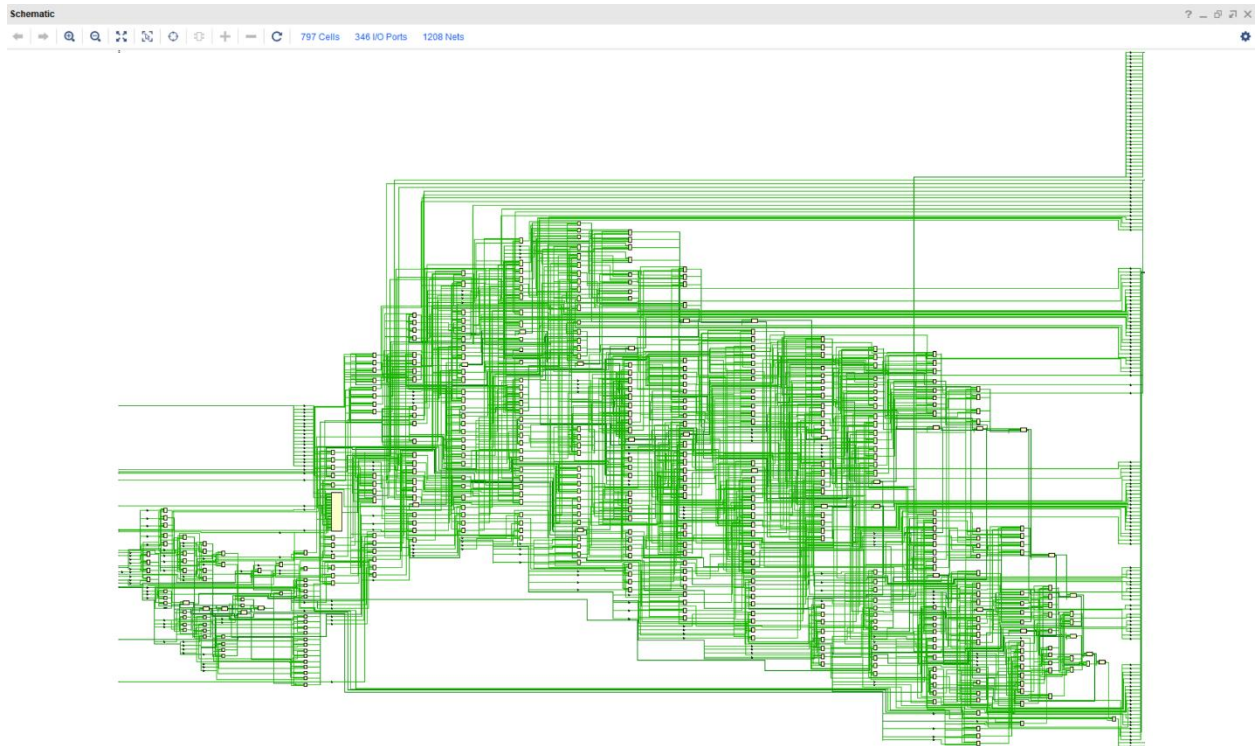
- **Schematic after Elaboration:**



- **Messages after Elaboration:**



• Schematic After Synthesis:



- Utilization after Synthesis:

Tcl Console	Messages	Log	Reports	Design Runs	Timing	Utilization					
Hierarchy											
Hierarchy											
Summary											
v Slice Logic											
v Slice LUTs (<1%)											
LUT as Logic (<1%)											
v Slice Registers (<1%)											
Register as Flip Flop (<1%)											
utilization_1											

Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
Project_1	230	160	1	327	1

- Timing after Synthesis:

Tcl Console	Messages	Log	Reports	Design Runs	Timing	Utilization					
Design Timing Summary											
General Information											
Timer Settings											
Design Timing Summary											
Clock Summary (1)											
v Check Timing (326)											
v Intra-Clock Paths											
Inter-Clock Paths											
Other Path Groups											
Timing Summary - timing_1											

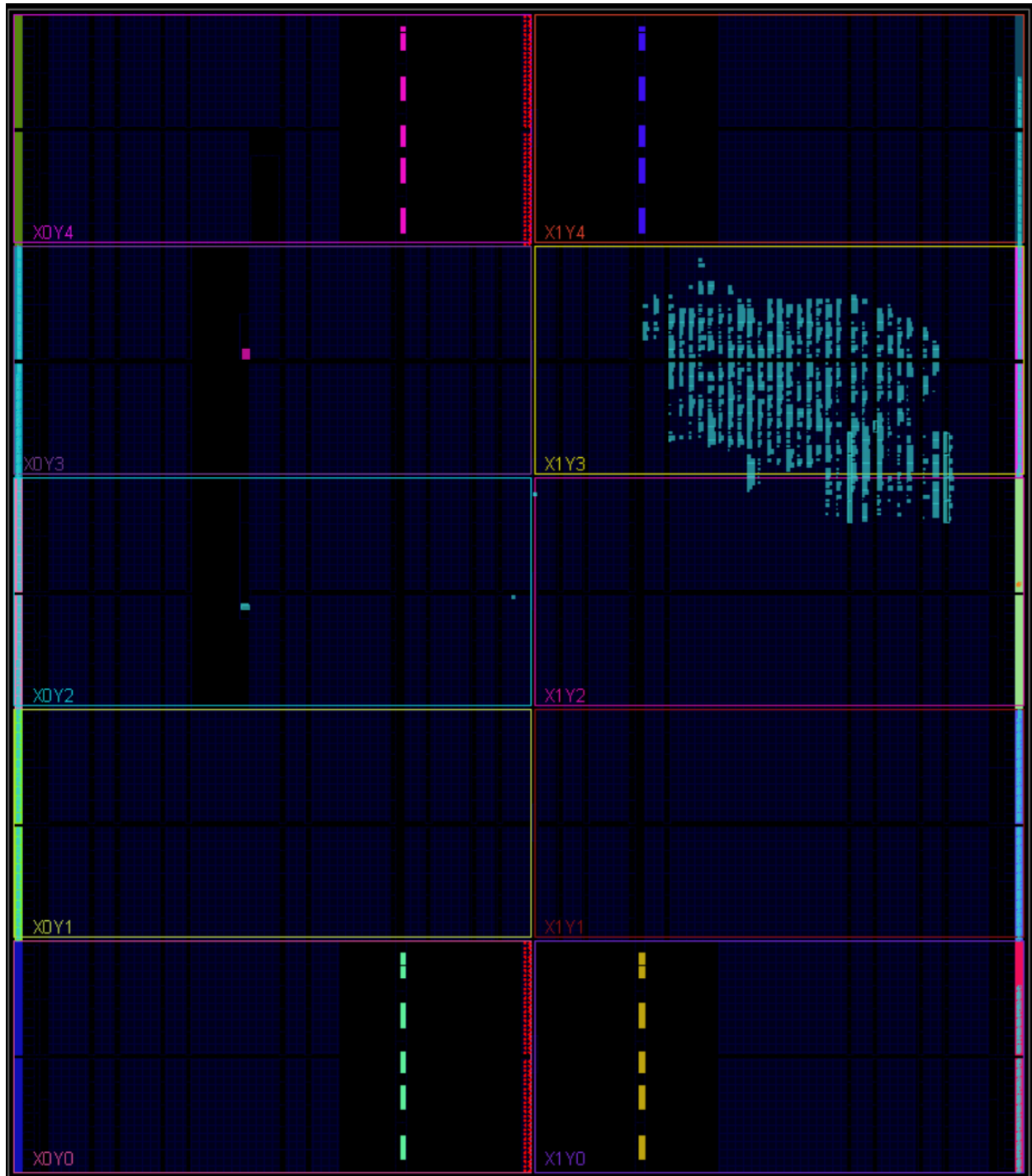
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.168 ns	Worst Hold Slack (WHS): 0.182 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 87	Total Number of Endpoints: 87	Total Number of Endpoints: 162

All user specified timing constraints are met.

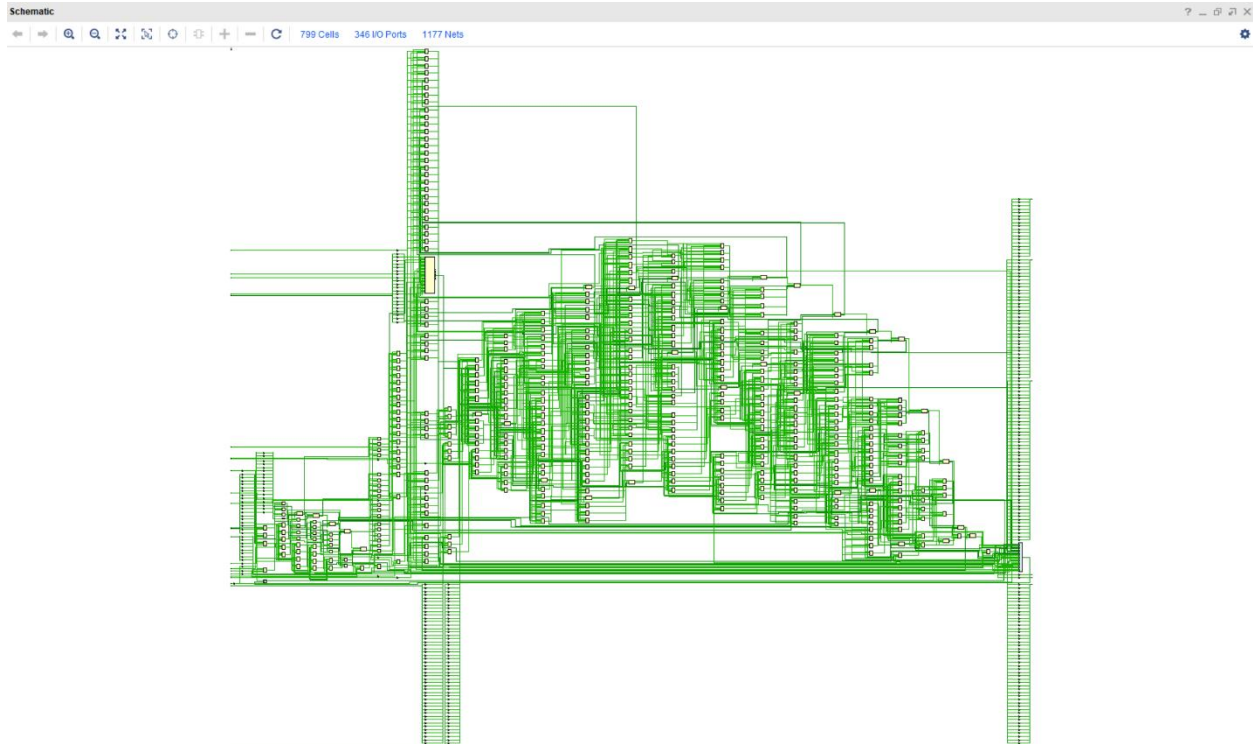
- Messages after Synthesis:

Tcl Console	Messages	Log	Reports	Design Runs	Debug					
Warning (44)										
Info (34)										
Status (20)										
Show All										
Vivado Commands (3 infos)										
Synthesis (44 warnings, 22 infos)										
Synthesized Design (9 infos)										

- Device after Implementation:



- Schematic after Implementation:



- Utilization after Implementation:

Name	Slice LUTs (133800)	Slice Registers (267600)	F7 Muxes (66900)	F8 Muxes (33450)	Slice (33450)	LUT as Logic (133800)	LUT as Memory (46200)	LUT Flip Flop Pairs (133800)	Block RAM Tile (365)	DSP s (740)	Bonded IOB (500)	BUFGCTRL (32)	BSCANEN2 (4)
Project_1	2652	4225	100	7	1510	2178	474	1554	8	1	327	2	1
dbg_hub (dbg_hub)	475	727	0	0	259	451	24	306	0	0	0	1	1
u_ila_0 (u_ila_0)	1947	3338	100	7	1180	1497	450	1196	8	0	0	0	0

- Timing after Implementation:

Design Timing Summary			
Setup	Hold	Pulse Width	
Worst Negative Slack (WNS): 2.891 ns	Worst Hold Slack (WHS): 0.017 ns	Worst Pulse Width Slack (WPWS): 3.950 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: 8049	Total Number of Endpoints: 8033	Total Number of Endpoints: 5136	

- Messages after Implementation:

The screenshot shows the Vivado Messages window with the following tabs: Tcl Console, Messages, Log, Reports, Design Runs, Power, Methodology, DRC, Timing, and Utilization. The Messages tab is active, displaying a list of messages categorized by type and count:

- > Vivado Commands (3 infos)
- > Synthesis (44 warnings, 22 infos)
- > Implementation (2 warnings, 102 infos)
- > Implemented Design (2 warnings, 11 infos)

At the top, there are filters for Warning (50), Info (240), and Status (490), along with a 'Show All' button.

- Quest Lint Checking:

The screenshot shows the Quest Lint Checking interface. The main window is divided into two panes. The left pane displays the source code for 'Project_1', which includes various parameters and inputs. The right pane shows the 'Lint Summary' table, which provides a quick overview of the linting results.

Name	Count
Open(uninspected, ...	5 (21)
Info	5 (21)

Below the code editor, there is a 'Lint Checks' section with a table of detailed linting results. The table has columns for Severity, Status, Check, Message, Module, Category, State, Owner, and STARC Reference.

Severity	Status	Check	Message	Module	Category	State	Owner	STARC Reference
Warning	Fixed	condition_const	Condition expression is a constant. Module ...	Project_1	Rtl Design ...	open	unass...	
Warning	Fixed	condition_const	Condition expression is a constant. Module ...	Project_1	Rtl Design ...	open	unass...	
Warning	Fixed	condition_const	Condition expression is a constant. Module ...	Project_1	Rtl Design ...	open	unass...	
Warning	Fixed	always_signal_assig...	Always block has more signal assignments ...	Project_1	Rtl Design ...	open	unass... 2.6.1.3	
Warning	Fixed	multi_ports_in_singl...	Multiple ports are declared in one line. Mod...	Project_1	Rtl Design ...	open	unass... 3.5.6.3	

At the bottom, there is a 'Lint Checks' summary bar showing the total number of checks and their status. The bottom status bar indicates the current project is '...v [Project_1]'.