SPARTAN6 - DSP48A1

Project 1

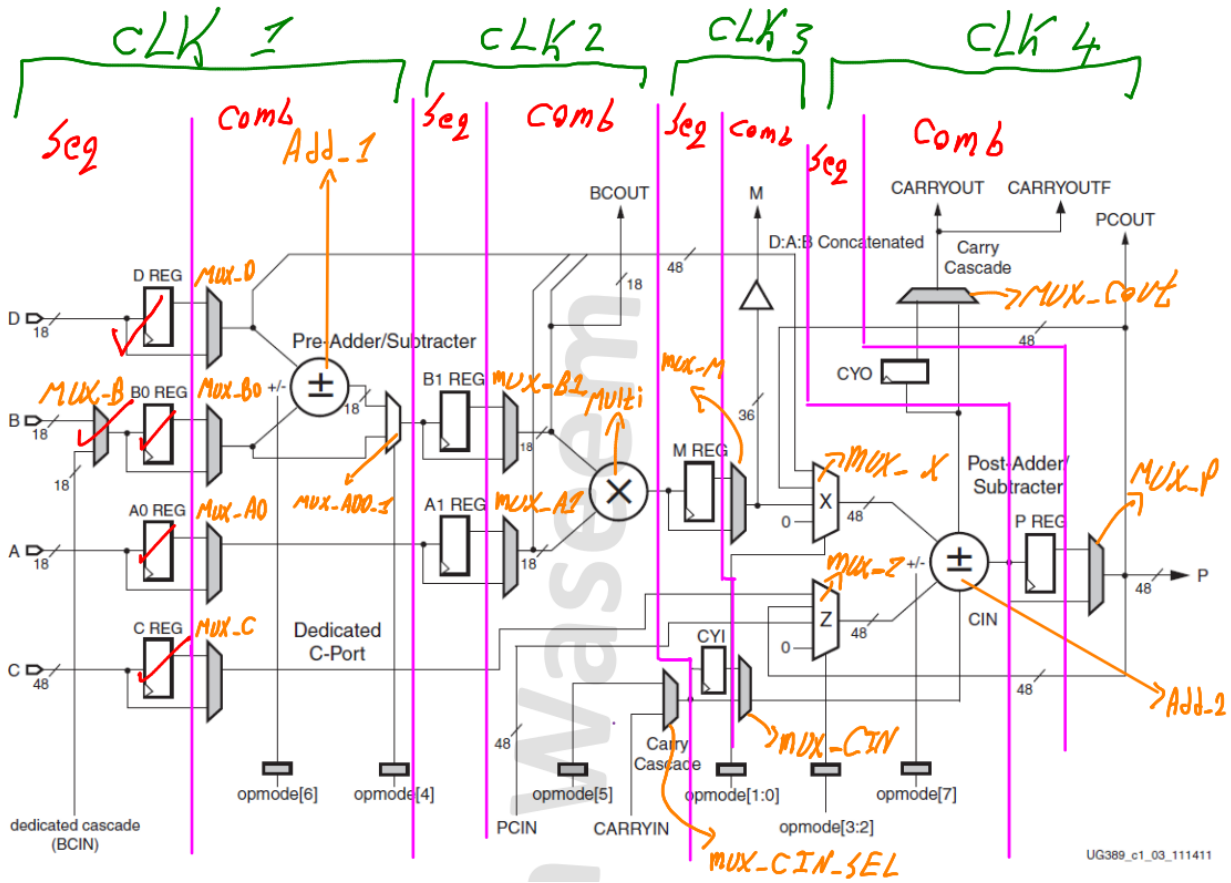
Under Supervision of:

ENG. Kareem Waseem

ENG. Mohamed Khaled

Submitted by: Ahmed Belal

- **Annotated Project Schematic:**



- **Main Module Code:**

```
module Project_1(clk,A,B,C,D,OPMODE,BCIN,CARRYIN,RSTA,RSTB,RSTM ,RSTP,RSTC,
RSTD ,RSTCARRYIN ,RSTOPMODE ,CEA ,CEB,CEM ,CEP ,CEC ,CED ,CECARRYIN,CEOPMODE,PCIN,
BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF);

parameter A0REG = 0 ;
parameter A1REG = 1 ;
parameter B0REG = 0 ;
parameter B1REG = 1 ;
parameter CREG = 1 ;
parameter DREG = 1 ;
parameter MREG = 1 ;
parameter PREG = 1 ;
parameter CARRYINREG = 1 ;
parameter CARRYOUTREG = 1 ;
parameter OPMODEREG = 1 ;
parameter CARRYINSEL = "OPMODE5" ;   // CARRYIN or OPMODE5
parameter B_INPUT = "DIRECT" ;   // DIRECT or CASCADE
parameter RSTTYPE = "SYNC" ;   // ASYNC or SYNC

input [17:0] A,B,D,BCIN ;
input [47:0] C,PCIN ;
input [7:0] OPMODE ;
input   CARRYIN,clk,CEA,CEB,CEC,CECARRYIN,
        CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,
        RSTM,RSTOPMODE,RSTP ;

output reg [47:0] P,PCOUT ;
output reg CARRYOUT,CARRYOUTF;
output reg [17:0] BCOUT;
output reg [35:0] M ;

wire [17:0] mux_B ;
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 ;
reg [47:0] C_reg,P_reg,mux_C,mux_P,mux_X,mux_Z ;
reg [35:0] M_reg,mux_M ;
reg [7:0] OPMODE_reg ;
reg CARRYIN_reg,CARRYOUT_reg,mux_CIN,mux_COUT,mux_CIN_SEL ;
reg [17:0] ADD_1 ;   // The output of Pre-Adder/Subtracter
reg [35:0] MULTI ;   // The output of Multiplication process
reg [48:0] ADD_2 ;   /* The output of Post-Adder/Subtracter but it is used
                        49 bit to take the ADD_2[48] to give it to CARYYOUT*/


assign mux_B =  (B_INPUT == "DIRECT") ? B :
                (B_INPUT == "CASCADE") ? BCIN : 18'b0 ;  // B first selection

```

```
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
```

```
88   //FIRST Combinational Always Block of clk_1
89   always @(*) begin
90
91       // mux_A0
92       if(A0REG)
93           mux_A0 = A0_reg ;
94       else
95           mux_A0 = A ;
96
97       // mux_B0
98       if(B0REG)
99           mux_B0 = B0_reg ;
100      else
101          mux_B0 = mux_B ;
102
103      // mux_C
104      if(CREG)
105          mux_C = C_reg ;
106      else
107          mux_C = C ;
108
109      // mux_D
110      if(DREG)
111          mux_D = D_reg ;
112      else
113          mux_D = D ;
114
115      // Pre-Adder/Subtracter Output (ADD_1)
116      if (OPMODE_reg[6])
117          ADD_1 = mux_D - mux_B0 ;
118      else
119          ADD_1 = mux_D + mux_B0 ;
120
121      // Mux of OPMODE[4] that choose bet. Pre-Adder/Subtracter OR Mux_B0
122      if (OPMODE_reg[4])
123          mux_ADD_1 = ADD_1 ;
124      else
125          mux_ADD_1 = mux_B0 ;
126
127  end
128
129
```

```
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
```

```
//THIRD Sequential Always Block of clk_3
generate
if (RSTTYPE == "ASYNC") begin
    always @(posedge clk or posedge RSTM or posedge RSTCARRYIN) begin
        if (RSTM) M_reg <= 0;
        else if (CEM) M_reg <= MULTI;

        if (RSTCARRYIN) CARRYIN_reg <= 0;
        else if (CECARRYIN) CARRYIN_reg <= mux_CIN_SEL;
    end
end else if (RSTTYPE == "SYNC") begin
    always @(posedge clk) begin
        if (RSTM) M_reg <= 0;
        else if (CEM) M_reg <= MULTI;

        if (RSTCARRYIN) CARRYIN_reg <= 0;
        else if (CECARRYIN) CARRYIN_reg <= mux_CIN_SEL;
    end
end
endgenerate


//THIRD Combinational Always Block of clk_3
always @(*) begin

    // mux_M
    if (MREG)
        mux_M = M_reg ;
    else
        mux_M = MULTI ;

    // mux_CIN
    if (CARRYINREG)
        mux_CIN = CARRYIN_reg ;
    else
        mux_CIN = mux_CIN_SEL ;

    // M Output
    M = mux_M ;

    // mux_X
    case (OPMODE_reg[1:0])
        2'b00: mux_X = 0 ;
        2'b01: mux_X = mux_M ;
        2'b10: mux_X = mux_P ;
        2'b11: mux_X = {mux_D[11:0] , mux_A1[17:0] , mux_B1[17:0]} ;
    endcase

    // mux_Z
    case (OPMODE_reg[3:2])
        2'b00: mux_Z = 0 ;
        2'b01: mux_Z = PCIN ;
        2'b10: mux_Z = mux_P ;
        2'b11: mux_Z = mux_C ;
    endcase

    // Post-Adder/Subtracter Output (ADD_2)
    if (OPMODE_reg[7])
        ADD_2 = mux_Z - (mux_X + mux_CIN) ;
    else
        ADD_2 = mux_Z + mux_X + mux_CIN ;

end


```

```
//FORTH Sequential Always Block of clk_4
generate
if (RSTTYPE == "ASYNC") begin
    always @(posedge clk or posedge RSTP or posedge RSTCARRYIN) begin
        if (RSTP) P_reg <= 0;
        else if (CEP) P_reg <= ADD_2[47:0];

        if (RSTCARRYIN) CARRYOUT_reg <= 0;
        else if (CECARRYIN) CARRYOUT_reg <= ADD_2[48];
    end
end else if (RSTTYPE == "SYNC") begin
    always @(posedge clk) begin
        if (RSTP) P_reg <= 0;
        else if (CEP) P_reg <= ADD_2[47:0];

        if (RSTCARRYIN) CARRYOUT_reg <= 0;
        else if (CECARRYIN) CARRYOUT_reg <= ADD_2[48];
    end
end
endgenerate


//FORTH Combinational Always Block of clk_4
always @(*) begin

    // Carry Cascade Output Mux
    if (CARRYOUTREG)
        mux_COUT = CARRYOUT_reg ;
    else
        mux_COUT = ADD_2[48] ;

    // CARRYOUT & CARRYOUTF Output
    CARRYOUT = mux_COUT ;
    CARRYOUTF = mux_COUT ;

    // mux_P
    if(PREG)
        mux_P = P_reg ;
    else
        mux_P = ADD_2[47:0] ;

    // P & PCOUT Output
    P = mux_P ;
    PCOUT = mux_P ;

end

endmodule
```

- **Test Bench Code:**

```
module Project_1_tb();


reg [17:0] A,B,D,BCIN ;
reg [47:0] C,PCIN ;
reg [7:0] OPMODE ;
reg   CARRYIN,clk,CEA,CEB,CEC,CECARRYIN,
      CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,
      RSTM,RSTOPMODE,RSTP ;

wire [47:0] P,PCOUT ;
wire CARRYOUT,CARRYOUTF;
wire [17:0] BCOUT;
wire [35:0] M ;

Project_1 #(
    .A0REG(0),
    .A1REG(1),
    .B0REG(0),
    .B1REG(1),
    .CREG(1),
    .DREG(1),
    .MREG(1),
    .PREG(1),
    .CARRYINREG(1),
    .CARRYOUTREG(1),
    .OPMODEREG(1),
    .CARRYINSEL("OPMODE5"),
    .B_INPUT("DIRECT"),
    .RSTTYPE("SYNC")
)tb (  clk,A,B,C,D,OPMODE,BCIN,CARRYIN,RSTA,RSTB,RSTM,RSTP,RSTC,
       RSTD,RSTCARRYIN,RSTOPMODE,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,
       CEOPMODE,PCIN,BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF
    );



initial begin
    clk = 0 ;
    forever begin
        #1 clk = ~clk ;
    end
end

initial begin

```

```
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
```

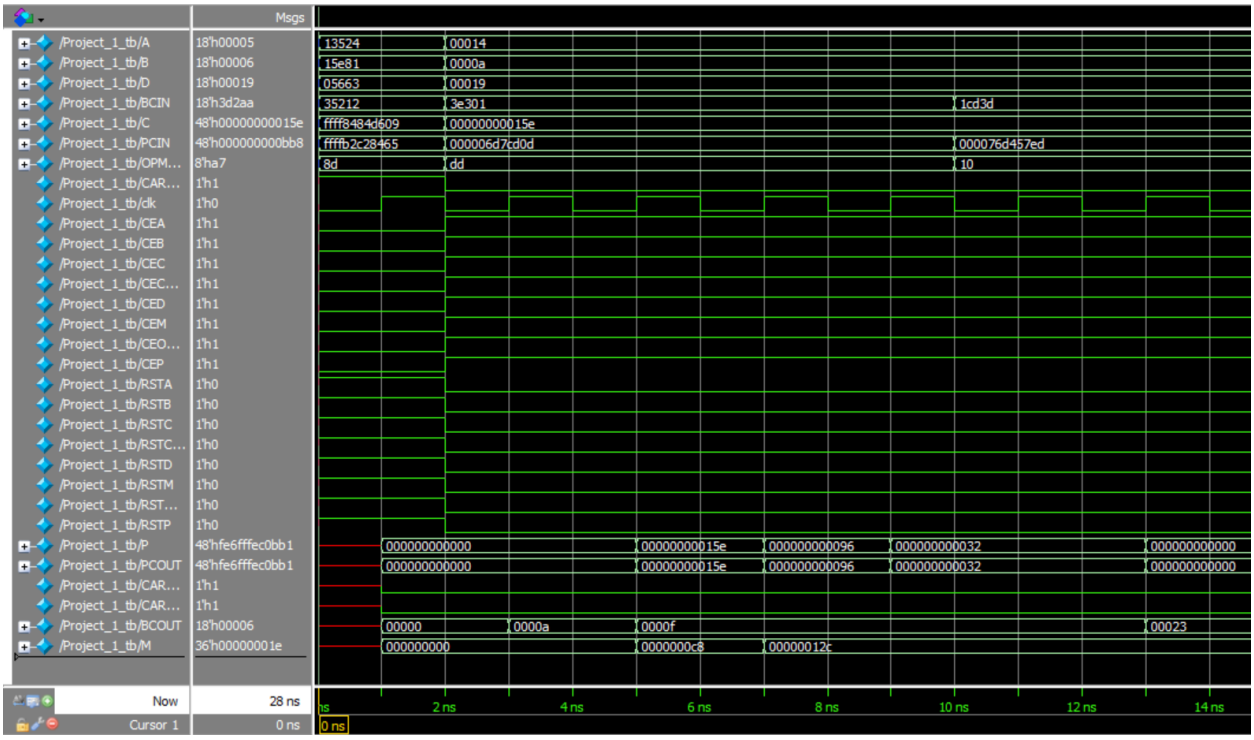
```
121  //2.3. Verify DSP Path 2
122  OPMODE = 8'b00010000 ;
123  A = 20 ;
124  B = 10 ;
125  C = 350 ;
126  D = 25 ;
127  BCIN = $random ;
128  PCIN = $random ;
129  CARRYIN = $random ;
130
131  repeat(3) @(negedge clk) ;
132
133  if (M != 36'h2bc || P != 0 || CARRYOUT != 0 || CARRYOUTF != 0 || BCOUT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

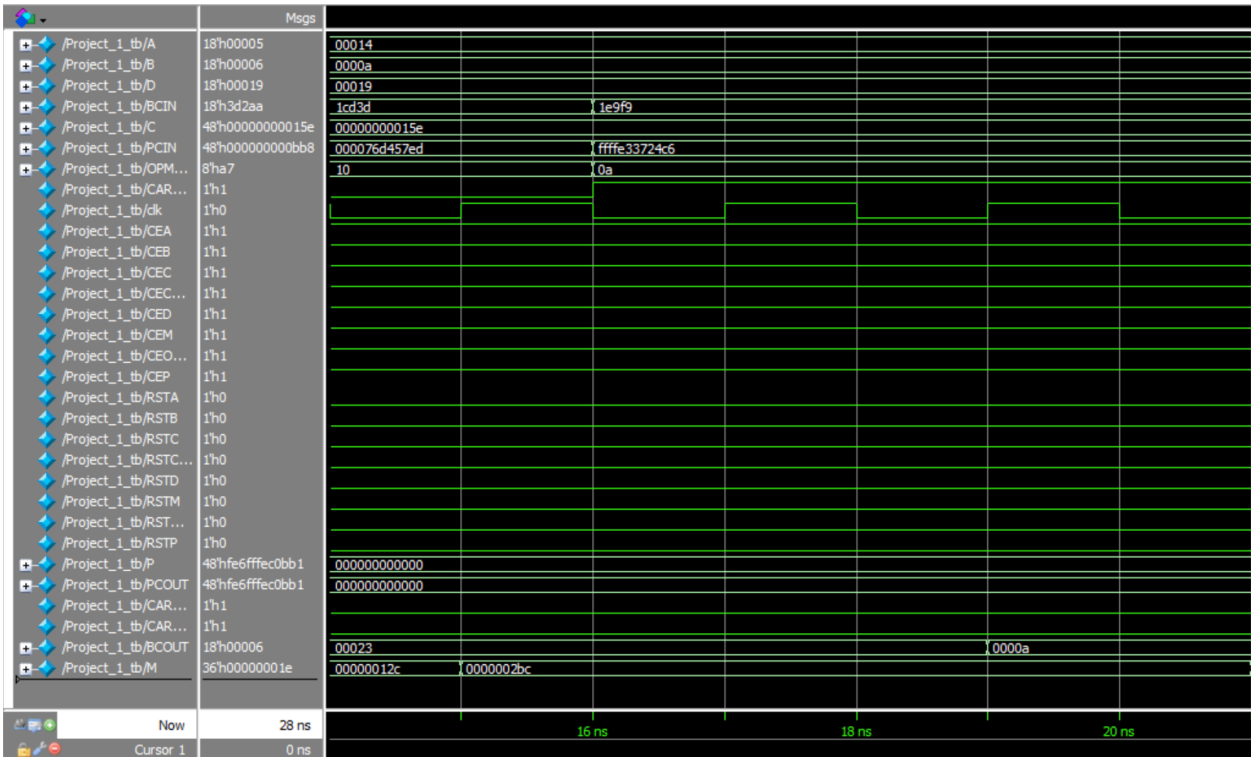
```
140   //2.4. Verify DSP Path 3
141   OPMODE = 8'b00001010 ;
142   A = 20 ;
143   B = 10 ;
144   C = 350 ;
145   D = 25 ;
146   BCIN = $random ;
147   PCIN = $random ;
148   CARRYIN = $random ;
149
150   repeat(3) @(negedge clk) ;
151
152   if (M != 36'hc8 || P != 0 || CARRYOUT != 0 || CARRYOUTF != 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'hfe6fffec0bb1 || CARRYOUT != 1 || CARRYOUTF != 1 || BCOUT != 18'h6 || PCOUT != 48'hfe6fffec0bb1) begin
172       $display ("error in 2.5. Verify DSP Path 4") ;
173       $stop ;
174   end
175
176   $stop;
177
178   end
179
180   endmodule
```

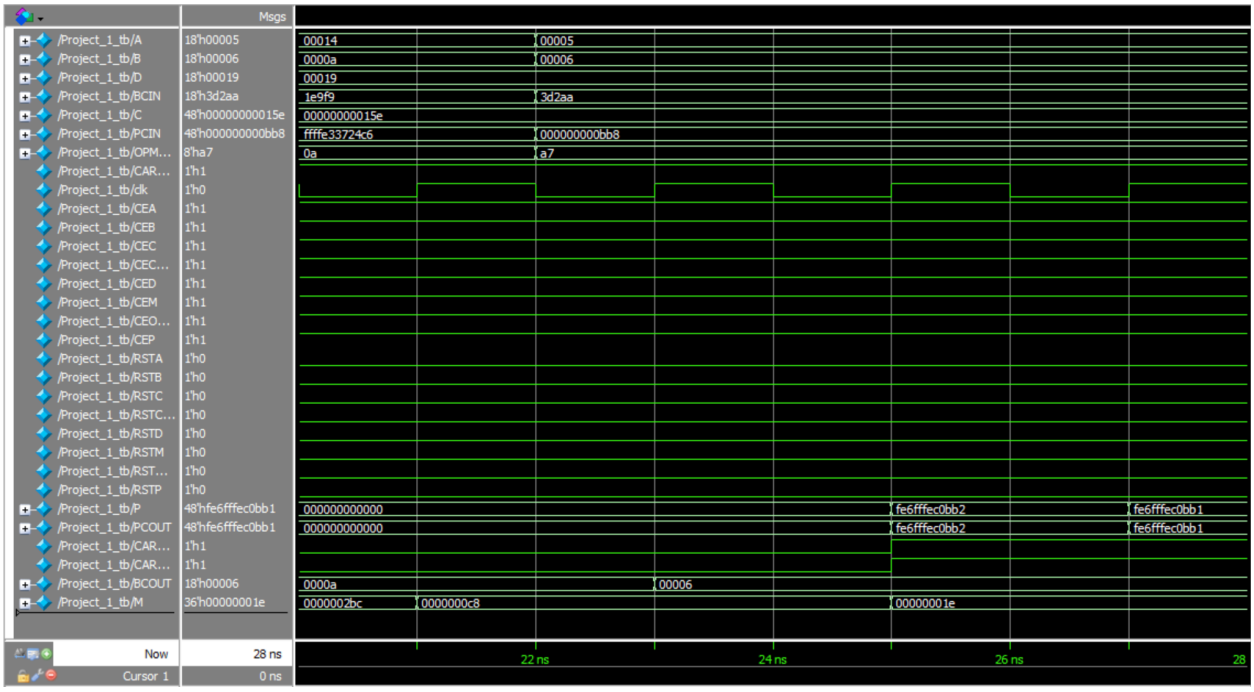
- **Do File:**

```
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:**






Msgs
/Project_1_tb/A
18'h00005
00014
00005
/Project_1_tb/B
18'h00006
0000a
00006
/Project_1_tb/D
18'h00019
00019
/Project_1_tb/BCIN
18'h3d2aa
1e9f9
3d2aa
/Project_1_tb/C
48'h00000000015e
00000000015e
/Project_1_tb/PCIN
48'h000000000bb8
ffffe33724c6
000000000bb8
/Project_1_tb/OPM...
8'ha7
0a
a7
/Project_1_tb/CAR...
1'h1
/Project_1_tb/clk
1'h0
/Project_1_tb/CEA
1'h1
/Project_1_tb/CEB
1'h1
/Project_1_tb/CEC
1'h1
/Project_1_tb/CEC...
1'h1
/Project_1_tb/CED
1'h1
/Project_1_tb/CEM
1'h1
/Project_1_tb/CEO...
1'h1
/Project_1_tb/CEP
1'h1
/Project_1_tb/RSTA
1'h0
/Project_1_tb/RSTB
1'h0
/Project_1_tb/RSTC
1'h0
/Project_1_tb/RSTC...
1'h0
/Project_1_tb/RSTD
1'h0
/Project_1_tb/RSTM
1'h0
/Project_1_tb/RST...
1'h0
/Project_1_tb/RSTP
1'h0
/Project_1_tb/P
48'hfe6fffec0bb1
000000000000
fe6fffec0bb2
fe6fffec0bb1
/Project_1_tb/PCOUT
48'hfe6fffec0bb1
000000000000
fe6fffec0bb2
fe6fffec0bb1
/Project_1_tb/CAR...
1'h1
/Project_1_tb/CAR...
1'h1
/Project_1_tb/BCOUT
18'h00006
0000a
00006
/Project_1_tb/M
36'h00000001e
0000002bc
0000000c8
00000001e
Now
28 ns
Cursor 1
0 ns
22 ns
24 ns
26 ns
28

- **Only timing constraint & Debug core constraint:**

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

```
create_debug_core u_ila_0 ila
set_property ALL_PROBE_SAME_MU true [get_debug_cores u_ila_0]
set_property ALL_PROBE_SAME_MU_CNT 1 [get_debug_cores u_ila_0]
set_property C_ADV_TRIGGER false [get_debug_cores u_ila_0]
set_property C_DATA_DEPTH 1024 [get_debug_cores u_ila_0]
set_property C_EN_STRG_QUAL false [get_debug_cores u_ila_0]
set_property C_INPUT_PIPE_STAGES 0 [get_debug_cores u_ila_0]
set_property C_TRIGIN_EN false [get_debug_cores u_ila_0]
set_property C_TRIGOUT_EN false [get_debug_cores u_ila_0]
set_property port_width 1 [get_debug_ports u_ila_0/clk]
connect_debug_port u_ila_0/clk [get_nets [list clk_IBUF_BUFG]]
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe0]
set_property port_width 18 [get_debug_ports u_ila_0/probe0]
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[
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe1]
set_property port_width 18 [get_debug_ports u_ila_0/probe1]
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[
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe2]
set_property port_width 18 [get_debug_ports u_ila_0/probe2]
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_O
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe3]
set_property port_width 48 [get_debug_ports u_ila_0/probe3]
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[
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe4]
set_property port_width 18 [get_debug_ports u_ila_0/probe4]
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[
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe5]
set_property port_width 36 [get_debug_ports u_ila_0/probe5]
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[
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe6]
set_property port_width 48 [get_debug_ports u_ila_0/probe6]
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[
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe7]
set_property port_width 8 [get_debug_ports u_ila_0/probe7]
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]} {O
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe8]
set_property port_width 48 [get_debug_ports u_ila_0/probe8]
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]}
```

```
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]}
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe9]
set_property port_width 1 [get_debug_ports u_ila_0/probe9]
connect_debug_port u_ila_0/probe9 [get_nets [list CARRYOUTF_OBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe10]
set_property port_width 1 [get_debug_ports u_ila_0/probe10]
connect_debug_port u_ila_0/probe10 [get_nets [list CEA_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe11]
set_property port_width 1 [get_debug_ports u_ila_0/probe11]
connect_debug_port u_ila_0/probe11 [get_nets [list CEB_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe12]
set_property port_width 1 [get_debug_ports u_ila_0/probe12]
connect_debug_port u_ila_0/probe12 [get_nets [list CEC_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe13]
set_property port_width 1 [get_debug_ports u_ila_0/probe13]
connect_debug_port u_ila_0/probe13 [get_nets [list CECARRYIN_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe14]
set_property port_width 1 [get_debug_ports u_ila_0/probe14]
connect_debug_port u_ila_0/probe14 [get_nets [list CED_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe15]
set_property port_width 1 [get_debug_ports u_ila_0/probe15]
connect_debug_port u_ila_0/probe15 [get_nets [list CEM_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe16]
set_property port_width 1 [get_debug_ports u_ila_0/probe16]
connect_debug_port u_ila_0/probe16 [get_nets [list CEOPMODE_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe17]
set_property port_width 1 [get_debug_ports u_ila_0/probe17]
connect_debug_port u_ila_0/probe17 [get_nets [list CEP_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe18]
set_property port_width 1 [get_debug_ports u_ila_0/probe18]
connect_debug_port u_ila_0/probe18 [get_nets [list clk_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe19]
set_property port_width 1 [get_debug_ports u_ila_0/probe19]
connect_debug_port u_ila_0/probe19 [get_nets [list RSTA_IBUF]]
```

```
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe20]
set_property port_width 1 [get_debug_ports u_ila_0/probe20]
connect_debug_port u_ila_0/probe20 [get_nets [list RSTB_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe21]
set_property port_width 1 [get_debug_ports u_ila_0/probe21]
connect_debug_port u_ila_0/probe21 [get_nets [list RSTC_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe22]
set_property port_width 1 [get_debug_ports u_ila_0/probe22]
connect_debug_port u_ila_0/probe22 [get_nets [list RSTCARRYIN_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe23]
set_property port_width 1 [get_debug_ports u_ila_0/probe23]
connect_debug_port u_ila_0/probe23 [get_nets [list RSTD_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe24]
set_property port_width 1 [get_debug_ports u_ila_0/probe24]
connect_debug_port u_ila_0/probe24 [get_nets [list RSTM_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe25]
set_property port_width 1 [get_debug_ports u_ila_0/probe25]
connect_debug_port u_ila_0/probe25 [get_nets [list RSTOPMODE_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe26]
set_property port_width 1 [get_debug_ports u_ila_0/probe26]
connect_debug_port u_ila_0/probe26 [get_nets [list RSTP_IBUF]]
set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
connect_debug_port dbg_hub/clk [get_nets clk_IBUF_BUFG]
```

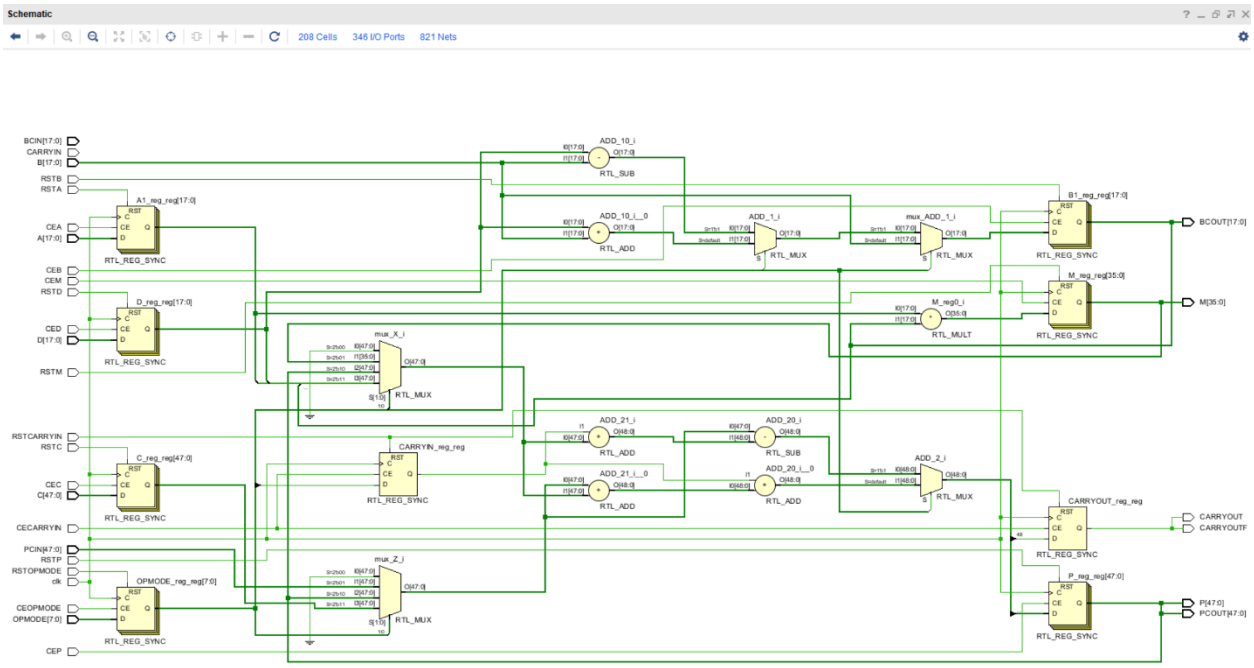
➢ It is important to note that:

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.

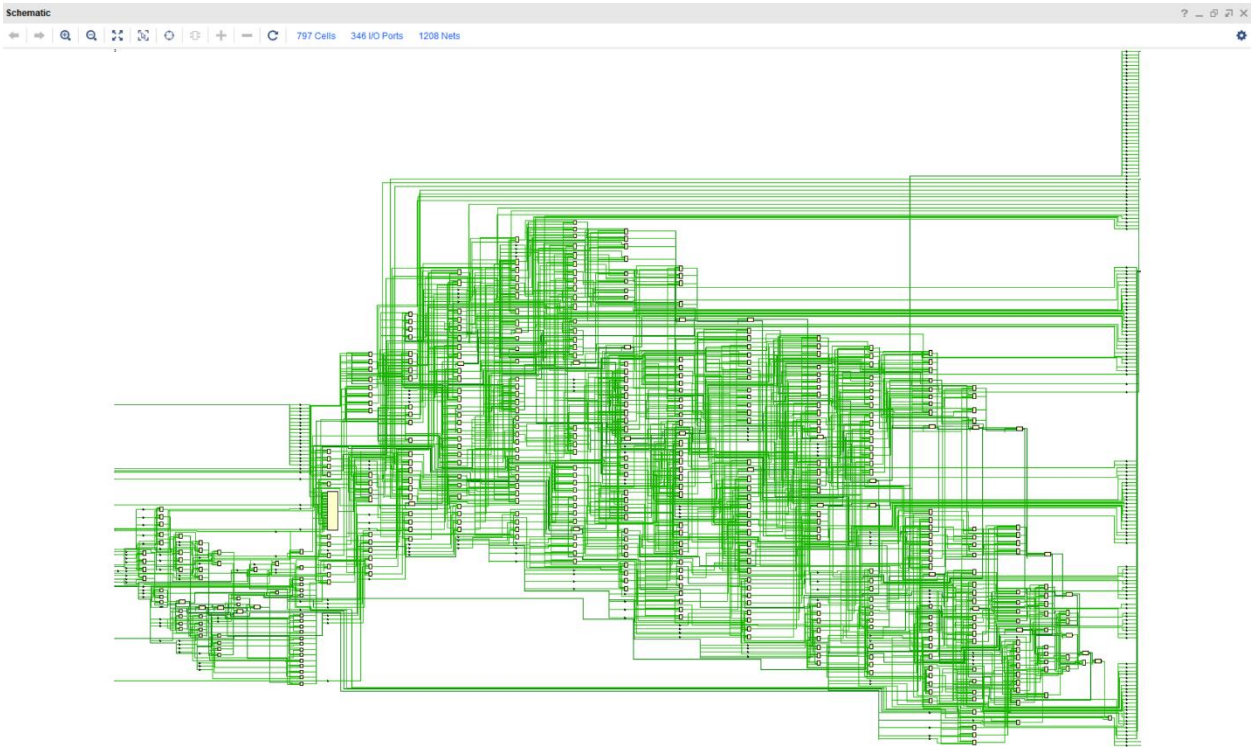
Since these elements are not functionally required in the current parameter configuration, **VIVADO removes them during synthesis optimization**. This is expected behavior and confirms that the design is functioning correctly according to the defined parameters.

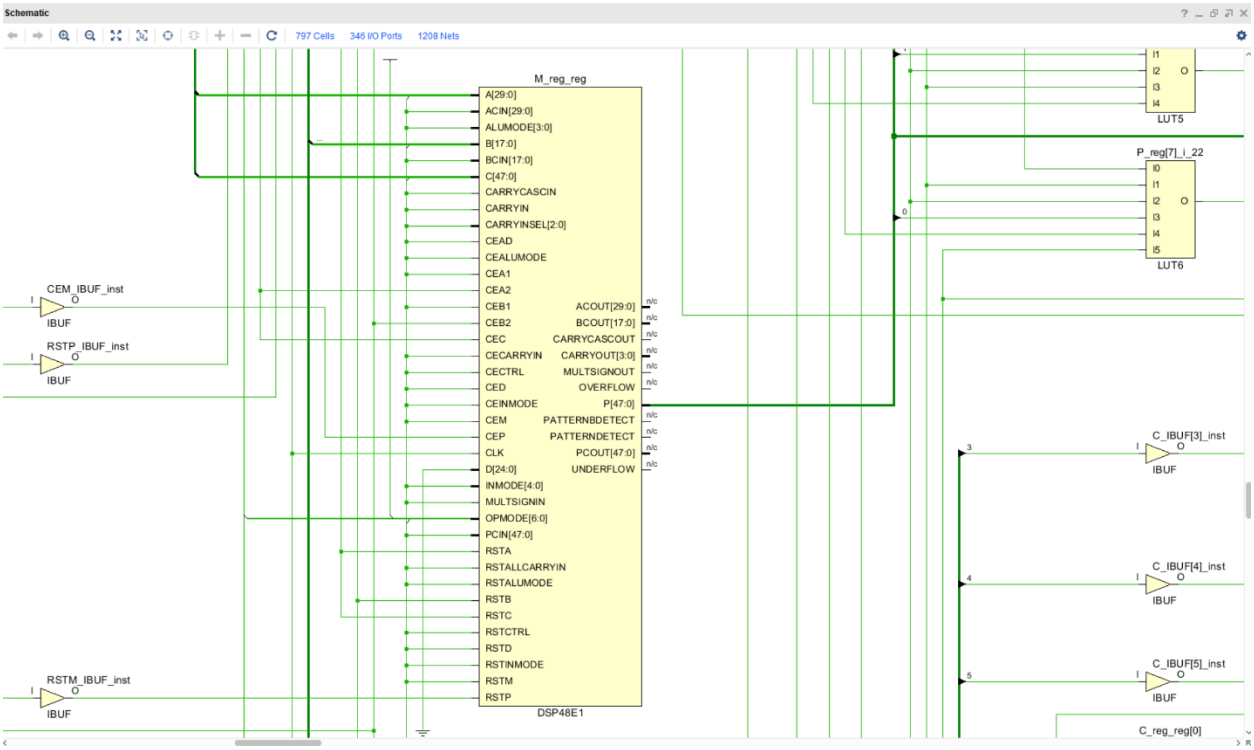
- **Schematic after Elaboration:**



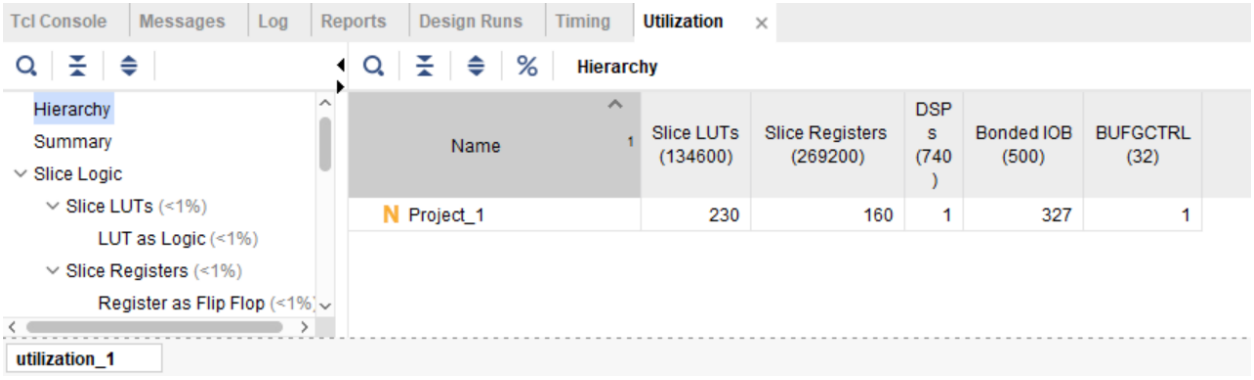
- **Messages after Elaboration:**

- **Schematic After Synthesis:**

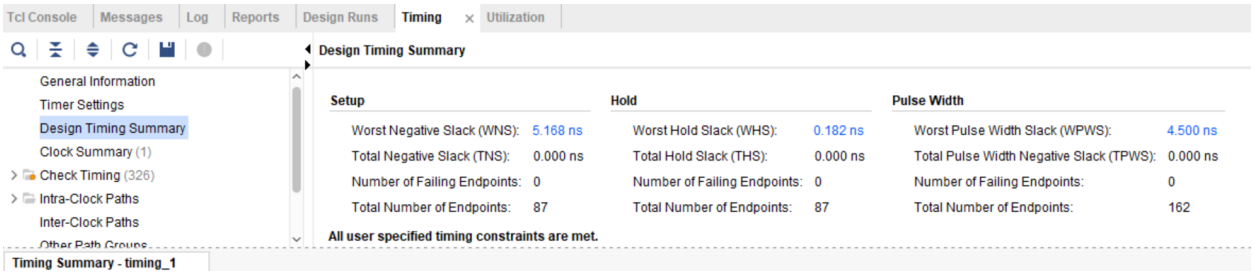




- **Utilization after Synthesis:**

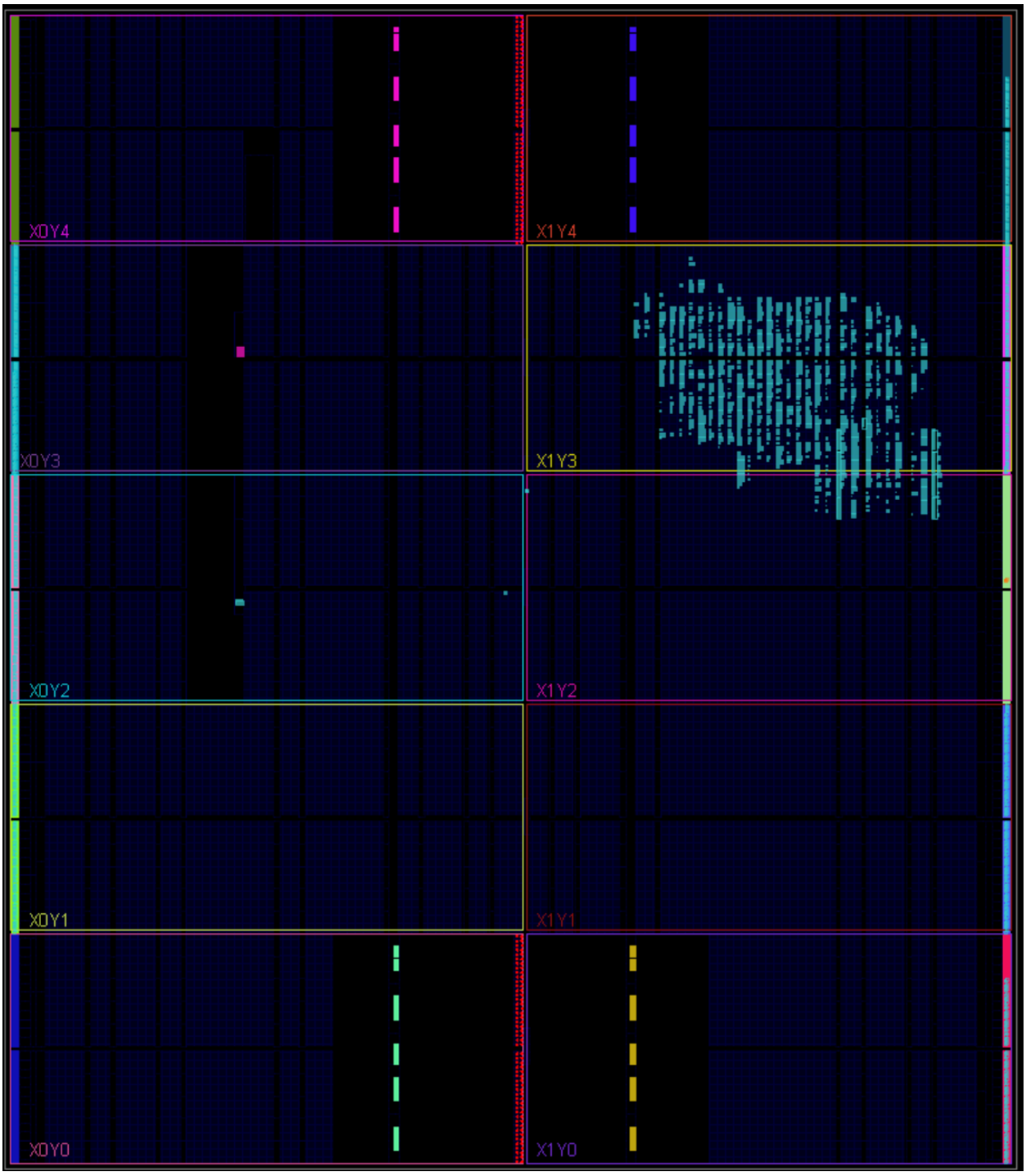


- **Timing after Synthesis:**

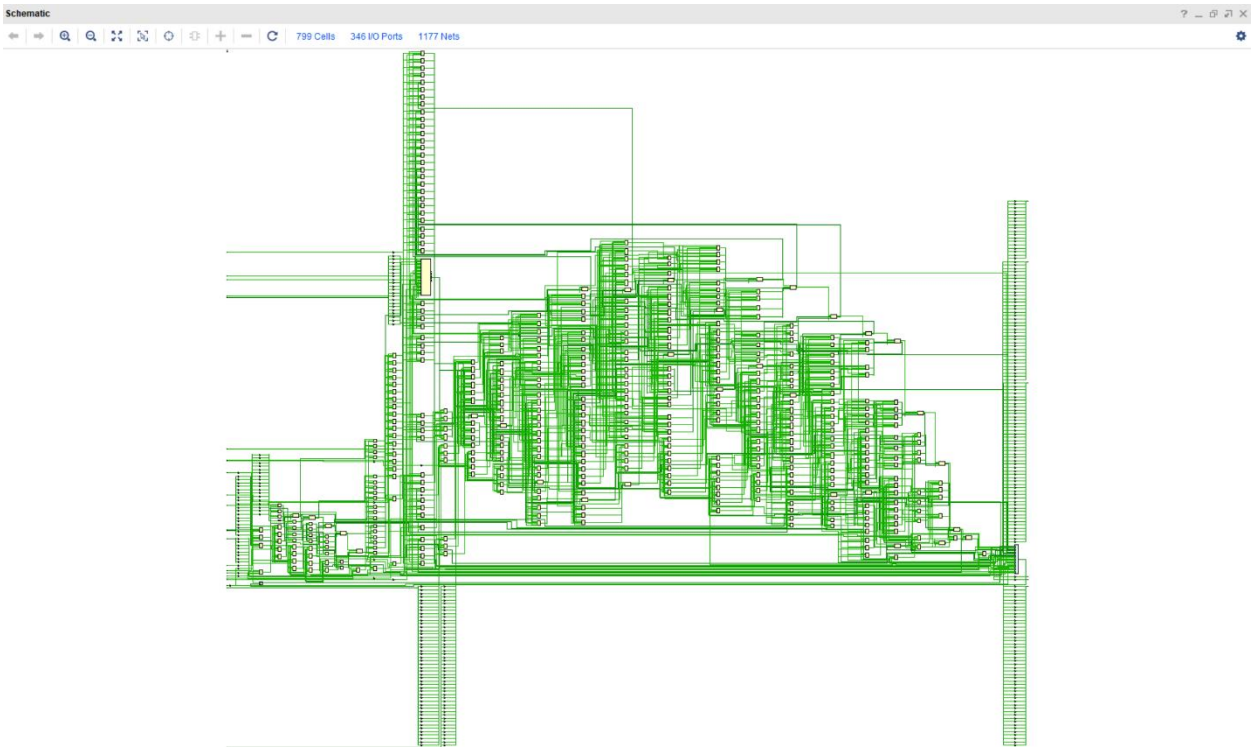


- **Messages after Synthesis:**

- **Device after Implementation:**



- **Schematic after Implementation:**

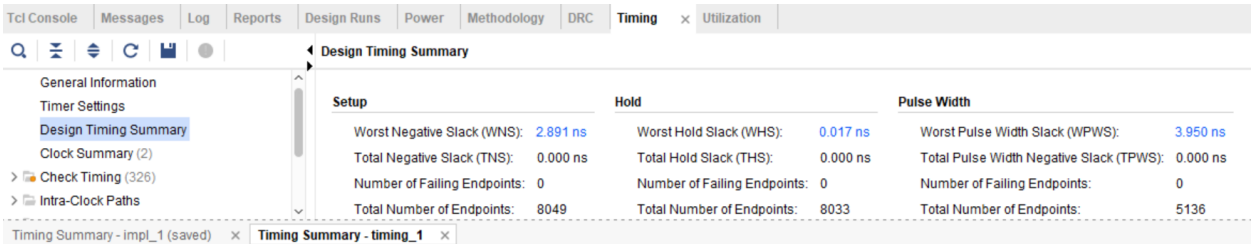


- **Utilization after Implementation:**

Hierarchy

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)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)	BSCANE2 (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:**



- **Messages after Implementation:**

- **Quest Lint Checking:**