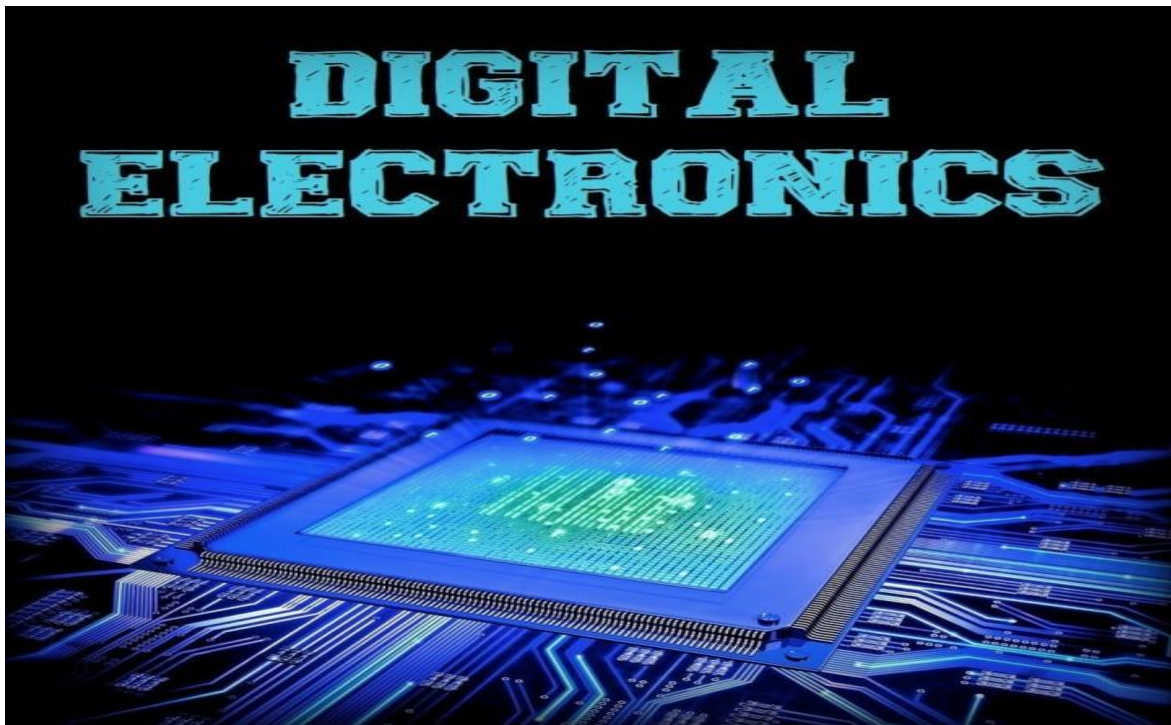


Abanob Evram

Spartan6-DSP48A1



[Q1]

The code of reg_mux module:

```
1 module reg_mux(data,rst,ce,clk,out);
2     parameter WIDTH=18;
3     parameter RSTTYPE = "SYNC";//SYNC or ASYNC
4     parameter REGISTER_ENABLE=1;//1 or 0
5     input [WIDTH-1:0] data;
6     input clk,rst,ce;
7     output [WIDTH-1:0] out;
8     reg [WIDTH-1:0] out_sync,out_async;
9
10    always @(posedge clk ) begin
11        if (RSTTYPE!="ASYNC") begin //default SYNC
12            if(rst)
13                out_sync<=0;
14            else if(ce)
15                out_sync<=data;
16        end
17    end
18    always @(posedge clk or posedge rst) begin
19        if (RSTTYPE=="ASYNC") begin
20            if(rst)
21                out_async<=0;
22            else if(ce)
23                out_async<=data;
24        end
25    end
26    assign out =(REGISTER_ENABLE&&RSTTYPE=="ASYNC"?out_async:(REGISTER_ENABLE&&RSTTYPE!="ASYNC"?out_sync:data );
27 endmodule
```

The main module code:

```
1 module Spartan6_DSP48A1
2   #(
3     parameter A0REG = 0, // if 1(register) ,0(no register)
4     A1REG = 1,
5     B0REG = 0,
6     B1REG = 1,
7     CREG = 1,
8     DREG = 1,
9     MREG = 1,
10    PREG = 1,
11    CARRYINREG = 1,
12    CARRYOUTREG = 1,
13    OPMODEREG = 1,
14    CARRYINSEL = "OPMODE5", //CARRYIN or OPMODE5 else out=0
15    B_INPUT = "DIRECT", //DIRECT or CASCADE
16    RSTTYPE = "SYNC" //SYNC or ASYNC
17  )
18  (
19    input [17:0] A,B,D,BCIN,
20    input [47:0] C,PCIN,
21    input [7:0] OPMODE,
22    input CLK,CARRYIN,
23    input RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPMODE,
24    input CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPMODE,
25    output [17:0] BCOUT,
26    output [47:0] PCOUT,P,
27    output [35:0] M,
28    output CARRYOUT, CARRYOUTF
29  );
30  wire [17:0] A0_m, A1_m, B0_m,B1_m,D_m,B0_in,B1_in,pre_adder_out;
31  wire [47:0] C_m,Post_adder_out;
32  wire [35:0] M_m,mult_out;
33  wire [7:0] OPMODE_m;
34  wire CYI_in,CYI_m,CYO_in;
35  reg [47:0] X_m,Z_m;
36
37  reg_mux #(18,RSTTYPE,A0REG) A0_REG(A,RSTA,CEA,CLK,A0_m);
38  reg_mux #(18,RSTTYPE,A1REG) A1_REG(A0_m,RSTA,CEA,CLK,A1_m);
39  reg_mux #(18,RSTTYPE,B0REG) B0_REG(B0_in,RSTB,CEB,CLK,B0_m);
40  reg_mux #(18,RSTTYPE,B1REG) B1_REG(B1_in,RSTB,CEB,CLK,B1_m);
41  reg_mux #(48,RSTTYPE,CREG) C_REG(C,RSTC,CEC,CLK,C_m);
42  reg_mux #(18,RSTTYPE,DREG) D_REG(D,RSTD,CED,CLK,D_m);
43  reg_mux #(36,RSTTYPE,MREG) M_REG(mult_out,RSTM,CEM,CLK,M_m);
44  reg_mux #(48,RSTTYPE,PREG) P_REG(Post_adder_out,RSTP,CEP,CLK,P);
45  reg_mux #(1,RSTTYPE,CARRYINREG) CYI_REG(CYI_in,RSTCARRYIN,CECARRYIN,CLK,CYI_m);
46  reg_mux #(1,RSTTYPE,CARRYOUTREG) CYO_REG(CYO_in,RSTCARRYIN,CECARRYIN,CLK,CARRYOUT);
47  reg_mux #(8,RSTTYPE,OPMODEREG) OPMODE_REG(OPMODE,RSTOPMODE,CEOPMODE,CLK,OPMODE_m);
48
49  assign CYI_in=(CARRYINSEL=="OPMODE5")?OPMODE_m[5]:(CARRYINSEL=="CARRYIN")?CARRYIN:0;
50
51  assign B0_in=(B_INPUT=="DIRECT")?B:(B_INPUT=="CASCADE")?BCIN:0;
52
```

```

40 reg_mux #(18,RSTTYPE,B1REG) B1_REG(B1_in,RSTB,CEB,CLK,B1_m);
41 reg_mux #(48,RSTTYPE,CREG) C_REG(C,RSTC,CEC,CLK,C_m);
42 reg_mux #(18,RSTTYPE,DREG) D_REG(D,RSTD,CED,CLK,D_m);
43 reg_mux #(36,RSTTYPE,MREG) M_REG(mult_out,RSTM,CEM,CLK,M_m);
44 reg_mux #(48,RSTTYPE,PREG) P_REG(Post_adder_out,RSTP,CEP,CLK,P);
45 reg_mux #(1,RSTTYPE,CARRYINREG) CYI_REG(CYI_in,RSTCARRYIN,CECARRYIN,CLK,CYI_m);
46 reg_mux #(1,RSTTYPE,CARRYOUTREG) CYO_REG(CYO_in,RSTCARRYIN,CECARRYIN,CLK,CARRYOUT);
47 reg_mux #(8,RSTTYPE,OPMODEREG) OPMODE_REG(OPMODE,RSTOPMODE,CEOPMODE,CLK,OPMODE_m);
48
49 assign CYI_in=(CARRYINSEL=="OPMODE5")?OPMODE_m[5]:(CARRYINSEL=="CARRYIN")?CARRYIN:0;
50
51 assign B0_in=(B_INPUT=="DIRECT")?B:(B_INPUT=="CASCADE")?BCIN:0;
52
53 assign pre_adder_out=(OPMODE_m[6]==0)?(D_m+B0_m):(D_m-B0_m);
54
55 assign B1_in=(OPMODE_m[4]==1)?pre_adder_out:B0_m ;
56
57 assign BCOUT = B1_m;
58
59 assign mult_out=A1_m*B1_m;
60
61 assign M = M_m;
62
63
64 always @(*) begin
65     case(OPMODE_m[1:0])
66         0:X_m=0;
67         1:X_m=M_m;
68         2:X_m=P;
69         3:X_m={D_m[11:0],A1_m[17:0],B1_m[17:0]};
70     endcase
71 end
72 always @(*) begin
73     case(OPMODE_m[3:2])
74         0:Z_m=0;
75         1:Z_m=PCIN;
76         2:Z_m=P;
77         3:Z_m=C_m;
78     endcase
79 end
80 assign {CYO_in,Post_adder_out}=(OPMODE_m[7]==0)?(X_m+Z_m+CYI_m):(Z_m-(X_m+CYI_m));
81 assign CARRYOUTF = CARRYOUT ;
82 assign PCOUT = P ;
83 endmodule

```

The testbench code:

```

1 module DSP_tb();
2   localparam A0REG = 0; // if 1(register) ,0(no register)
3   localparam A1REG = 1;
4   localparam B0REG = 0;
5   localparam B1REG = 1;
6   localparam CREG = 1;
7   localparam DREG = 1;
8   localparam MREG = 1;
9   localparam PREG = 1;
10  localparam CARRYINREG = 1;
11  localparam CARRYOUTREG = 1;
12  localparam OPMODEREG = 1;
13  localparam CARRYINSEL = "OPMODE5"; //CARRYIN or OPMODE5 else out=0
14  localparam B_INPUT = "DIRECT"; //DIRECT or CASCADE
15  localparam RSTTYPE = "SYNC"; //SYNC or ASYNC
16
17  reg [17:0] A,B,D,BCIN;
18  reg [47:0] C,PCIN;
19  reg [7:0] OPMODE;
20  reg CLK,CARRYIN;
21  reg RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE;
22  reg CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE;
23  wire [17:0] Bcout;
24  wire [47:0] Pcout,P;
25  wire [35:0] M;
26  wire CARRYOUT,CARRYOUTF;
27
28  Spartan6_DSP48A1 #(A0REG,A1REG,B0REG,B1REG,CREG,DREG,MREG,PREG,CARRYINREG,CARRYOUTREG,OPMODEREG,CARRYINSEL,B_INPUT,RSTTYPE) dut(
29    A,B,D,BCIN,C,PCIN,OPMODE,CLK,CARRYIN,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE,Bcout,Pcout,P,M,CARRYOUT,CARRYOUTF);
30  initial begin
31    CLK=0;
32    forever
33      #1 CLK=~CLK;
34  end
35  integer i;
36  initial begin
37    // Initialize and reset signals
38    RSTA = 1; RSTB = 1; RSTM = 1; RSTP = 1; RSTC = 1; RSTD = 1; RSTCARRYIN = 1; RSTOPMODE = 1;
39    CEA = 1; CEB = 1; CEM = 1; CEP = 1; CEC = 1; CED = 1; CECARRYIN = 1; CEOPMODE = 1;
40    A = 0; B = 0; C = 0; D = 0; CARRYIN = 0; BCIN = 0; PCIN = 0;
41    OPMODE = 8'b00000000;
42    repeat(5)
43      @(negedge CLK);
44
45    // put reset signal =1
46    RSTA = 0; RSTB = 0; RSTM = 0; RSTP = 0; RSTC = 0; RSTD = 0; RSTCARRYIN = 0; RSTOPMODE = 0;
47    $display("Case : 1");
48    //B*A+C+opmode[5] =(15*2)+10+1=41
49    OPMODE=8'b01101101;
50    A=15;B=2;C=10;
51    repeat(5)
52      @(negedge CLK);

```

```

49 OPMODE=8'b01101101;
50 A=15;B=2;C=10;
51 repeat(5)
52 @(negedge CLK);
53
54 $display("Case : 2");
55 //C-((D-B)*A)=1000-((13-3)*10)=900
56 OPMODE=8'b11011101;
57 D=13;B=3;A=10;C=1000;
58 repeat(5)
59 @(negedge CLK);
60
61 $display("Case : 3");
62 //X_m=P,Z_m=P:x+z=900+900=1800
63 OPMODE=8'b01011010;
64 repeat(2)
65 @(negedge CLK);
66
67 $display("Case : 4");
68 //P=PCIN=12345;
69 OPMODE=8'b01010100;
70 PCIN=12345;
71 repeat(2)
72 @(negedge CLK);
73
74 $display("Case : 5");
75 //(D+B)*A=(3+2)*5=25
76 OPMODE=8'b00010001;
77 D=3;B=2;A=5;
78 repeat(5)
79 @(negedge CLK);
80
81 $display("Case : 6");
82 //Output equal the concatenated numbe
83 // =10_1010_1010_0101_0101_1010_1010_0101_0101_1010_1010_0101_0101
84 OPMODE = 8'b00010011;
85 D=18'b1010_1010_1010_0101_0101;
86 B=0;
87 A=18'b1010_1010_1010_0101_0101;
88 repeat(5)
89 @(negedge CLK);
90 $display("P=%b",P);
91
92 $display("Case : 7");
93 //Note the output from subtracter will be N_num so the out will be big num
94 OPMODE=8'b11011111;
95 B=47;
96 A=33;
97 C=47;
98 D=30;
99 PCIN=7;
100 repeat(5)

```

```

85     D=18'b1010_1010_1010_0101_0101;
86     B=0;
87     A=18'b1010_1010_1010_0101_0101;
88     repeat(5)
89     @(negedge CLK);
90     $display("P=%b",P);
91
92     $display("Case : 7");
93     //Note the output from subtracter will be N_num so the out will be big num
94     OPMODE=8'b11011111;
95     B=47;
96     A=33;
97     C=47;
98     D=30;
99     PCIN=7;
100    repeat(5)
101    @(negedge CLK);
102
103    $display("PCIN+M_m+CIN,M_m=(D+B)*A ,so P=PCIN+(D+B)*A+CIN");
104    OPMODE = 8'b00110101;
105    for(i=0;i<5;i=i+1) begin
106        $display("iteration : %d",i);
107        A=$urandom_range(1,50);
108        B=$urandom_range(1,50);
109        D=$urandom_range(1,50);
110        PCIN=$urandom_range(1,50);
111        repeat(5)
112        @(negedge CLK);
113    end
114
115    $display("Randomize cases");
116    for(i=0;i<1000;i=i+1) begin
117        A = $urandom_range(1, 50);
118        B = $urandom_range(1, 50);
119        D = $urandom_range(1, 50);
120        C = $urandom_range(1, 50);
121        PCIN = $urandom_range(1, 50);
122        OPMODE = $random;
123        repeat(5)
124        @(negedge CLK);
125    end
126
127
128    $stop;
129 end
130
131 initial begin
132     $monitor("A=%d, B=%d, C=%d, D=%d, CARRYIN=%d ,PCIN=%d , OPMODE=%b, P=%d,BCOUT=%d ,M=%d ,CARRYOUT=%d", A, B, C, D,CARRYIN ,PCIN , OPMODE, P,BCOUT ,M , CARRYOUT);
133 end
134 endmodule

```

The do file code:

```

1  vlib work
2
3  vlog Spartan6_DSP48A1.v reg_mux.v DSP_tb.v
4
5  vsim -voptargs=+acc DSP_tb
6
7  add wave *
8
9  run -all
10
11 #quit -sim

```


The constraint file:

```
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/clock]
171 connect_debug_port u_ila_0/clock [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]}]]
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 48 [get_debug_ports u_ila_0/probe1]
178 connect_debug_port u_ila_0/probe1 [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]}]]
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]}]]
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 18 [get_debug_ports u_ila_0/probe3]
186 connect_debug_port u_ila_0/probe3 [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]}]]
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 {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]}]]
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 8 [get_debug_ports u_ila_0/probe6]
198 connect_debug_port u_ila_0/probe6 [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]}]]
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 48 [get_debug_ports u_ila_0/probe7]
202 connect_debug_port u_ila_0/probe7 [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]}]]
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 {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]}]]
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]]
```



```

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

```

```

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/clock [get_nets CLK_IBUF_BUFG]
283

```

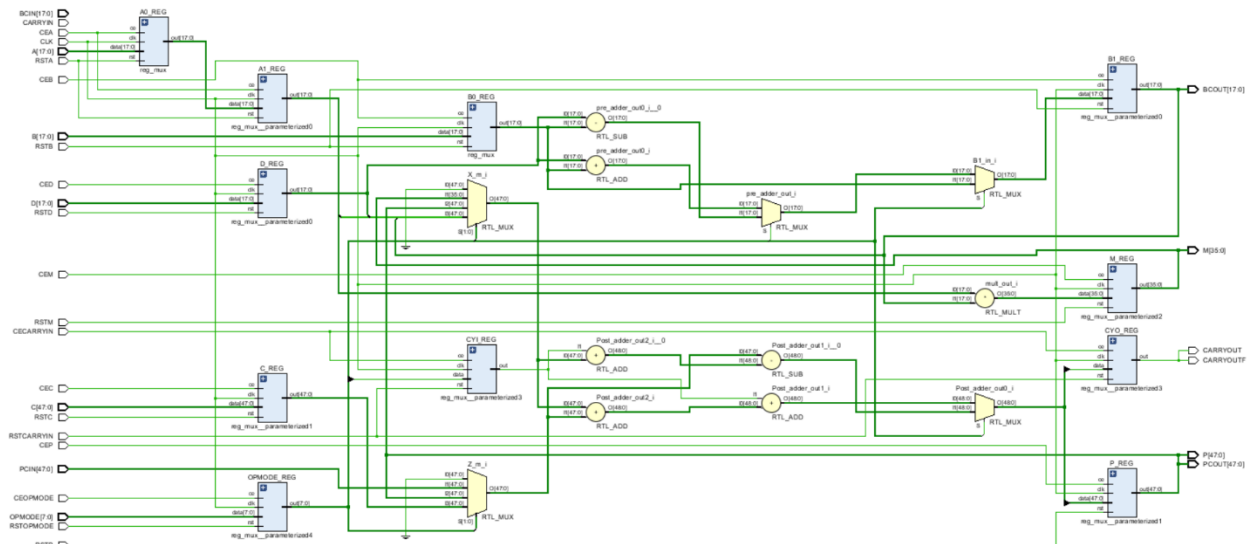
1-Elaboration (messages_tab)

Tcl Console Messages x Log Reports Design Runs

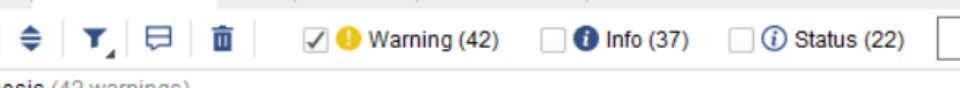
☐ Info (5)
 ☐ Status (9)
 Show All

- v Vivado Commands (3 infos)
 - v General Messages (3 infos)
 - [IP_Flow 19-234] Refreshing IP repositories
 - [IP_Flow 19-1704] No user IP repositories specified
 - [IP_Flow 19-2313] Loaded Vivado IP repository 'D:/Vivado/Vivado/2018.2/data/ip'.
- v Elaborated Design (2 infos)
 - v General Messages (2 infos)
 - [Project 1-570] Preparing netlist for logic optimization
 - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

2-Elaboration(Schematic)



3-synthesis(messages_tab)



Tcl Console Messages x Log Reports Design Runs

Search Zoom Sort Filter Chat Delete

☒ Warning (42) ☐ Info (37) ☐ Status (22) Show All

▼ Synthesis (42 warnings)

- > [Synth 8-3331] design reg_mux has unconnected port rst (40 more like this)
- [Constraints 18-5210] No constraint will be written out.

4-synthesis(Utilization_report)

Tcl Console	Messages	Log	Reports	Design Runs	Utilization	Debug
Hierarchy						
Hierarchy						
Summary						
▼ Slice Logic						
▼ Slice LUTs (<1%)						
LUT as Logic (<1%)						
▼ Slice Registers (<1%)						
Register as Flip Flop (<1%)						
Memory						
▼ DSP						
▼ DSPs (<1%)						
DSP48E1 only						
▼ IO and GT Specific						
▼ Bonded IOB (65%)						
IOB Master Pads						
▼ Clocking						
BUFGCTRL (3%)						

Name	Slice LUTs (134600)	Slice Registers (269200)	DSP s (740)	Bonded IOB (500)	BUFGCTRL (32)
▼ N Spartan6_DSP48A1	218	160	1	327	1
A1_REG (reg_mux_p...	0	18	0	0	0
B1_REG (reg_mux_p...	0	18	0	0	0
C_REG (reg_mux_pa...	0	48	0	0	0
CYI_REG (reg_mux_...	2	1	0	0	0
CYO_REG (reg_mux_...	0	1	0	0	0
D_REG (reg_mux_pa...	0	18	0	0	0
M_REG (reg_mux_pa...	0	0	1	0	0
OPMODE_REG (reg_...	216	8	0	0	0
P_REG (reg_mux_pa...	0	48	0	0	0

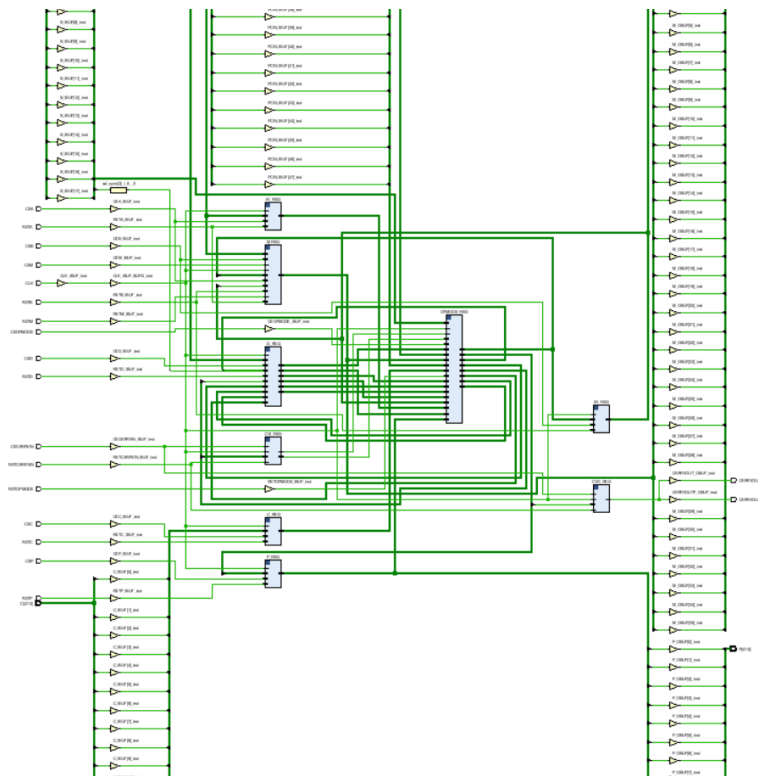
5-synthesis(Timing_report)

Tcl Console	Messages	Log	Reports	Design Runs	Timing	Utilization	Debug
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							
User Ignored Paths							
Unconstrained Paths							

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.516 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.

6-synthesis(Schematic)



7-implmmentation(messages_tab)

Tcl Console Messages x Log Reports Design Runs Power DRC Methodology Timing

Warning (46) Info (244) Status (479) Show All

- > Synthesis (42 warnings)
- > Implementation (2 warnings)
 - > Route Design (2 warnings)
 - > DRC (1 warning)
 - > Pin Planning (1 warning)
 - [Timing 38-436] There are set_bus_skew constraint(s) in this design. Please run report_bus_skew to ensure that bus skew requirements are met.

8-Implementation(Utilization_report)

Name	^	1	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)	BUFCTRL (32)	BSCANE2 (4)
▼ Spartan6_DSP48A1			2690	4225	97	12	1484	2216	474	1569	8	1	327	2	1
A1_REG (reg_mux_p...			0	18	0	0	7	0	0	0	0	0	0	0	0
B1_REG (reg_mux_p...			0	18	0	0	6	0	0	0	0	0	0	0	0
C_REG (reg_mux_pa...			0	48	0	0	15	0	0	0	0	0	0	0	0
CYL_REG (reg_mux_...			2	1	0	0	2	2	0	1	0	0	0	0	0
CYO_REG (reg_mux_...			0	1	0	0	1	0	0	0	0	0	0	0	0
D_REG (reg_mux_pa...			0	18	0	0	10	0	0	0	0	0	0	0	0
> dbg_hub (dbg_hub)			476	727	0	0	254	452	24	301	0	0	0	1	1
M_REG (reg_mux_pa...			0	0	0	0	0	0	0	0	0	1	0	0	0
OPMODE_REG (reg_...			216	8	0	0	65	216	0	0	0	0	0	0	0
P_REG (reg_mux_pa...			0	48	0	0	12	0	0	0	0	0	0	0	0
> u_ila_0 (u_ila_0)			1996	3338	97	12	1168	1546	450	1211	8	0	0	0	0

9-Implementation(Timing_report)

Reports

Design Runs

Power

DRC

Methodology

Timing

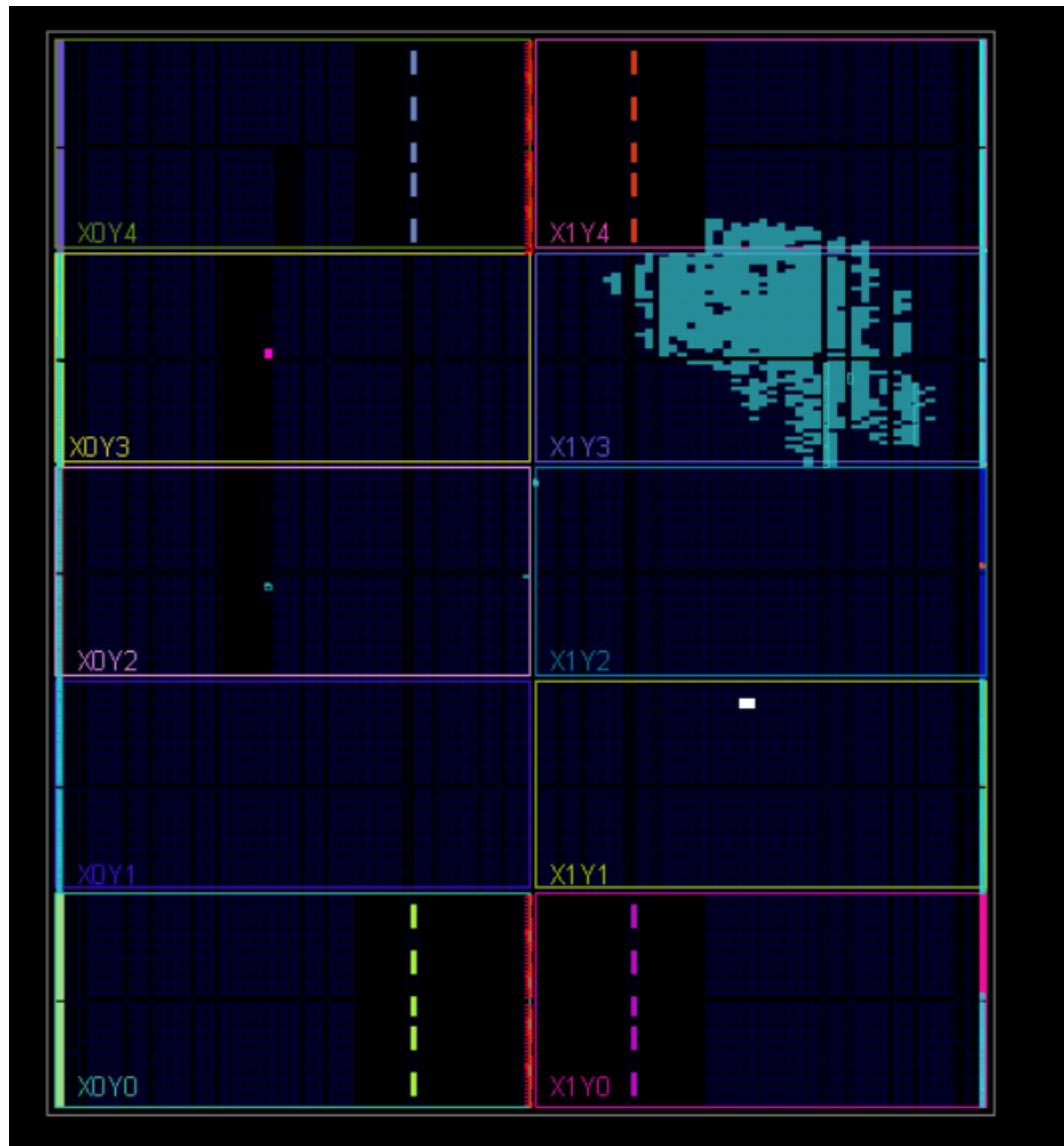
?

Design Timing Summary

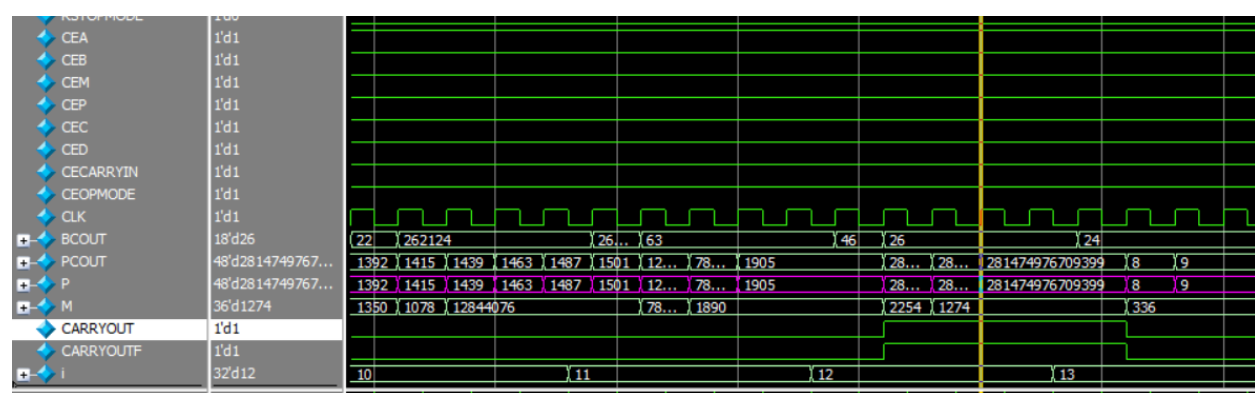
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 2.796 ns	Worst Hold Slack (WHS): 0.049 ns	Worst Pulse Width Slack (WPWS): 3.950 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 8058	Total Number of Endpoints: 8042	Total Number of Endpoints: 5136

Timing Summary - timing_1

10-Implmentation(device)



Signal	Width	Value
CEA	1'd1	
CEB	1'd1	
CEM	1'd1	
CEP	1'd1	
CEC	1'd1	
CED	1'd1	
CECARRYIN	1'd1	
CEOPMODE	1'd1	
CLK	1'd1	
BCOUT	18'd2	
PCOUT	48'd10	
P	48'd10	
M	36'd30	
CARRYOUT	1'd0	
CARRYOUTF	1'd0	
I	32'dx	



16