Project_1

DSP48A1 :

- Design
- Testbench
- Elaboration
- Synthesis
- Implementation

Design :

- ASYNC_rst
- SYNC_rst

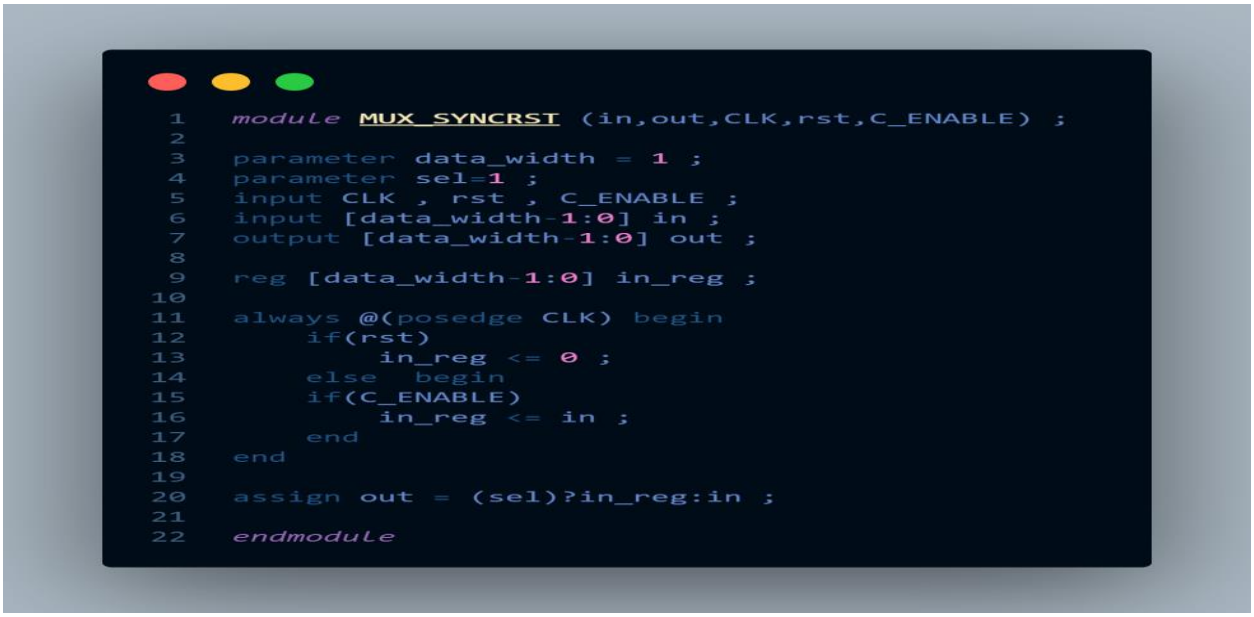
ASYNC_rst :

```verilog
module MUX_ASYNCRST (in,out,CLK,rst_a,C_ENABLE) ;

parameter data_width = 1 ;
parameter sel=1 ;
input CLK , rst_a , C_ENABLE ;
input [data_width-1:0] in ;
output [data_width-1:0] out ;

reg [data_width-1:0] in_reg ;

always @(posedge CLK or posedge rst_a) begin
    if(rst_a)
        in_reg <= 0 ;
    else  begin
    if(C_ENABLE)
        in_reg <= in ;
    end
end

assign out = (sel)?in_reg:in ;

endmodule
```

```verilog

module POST_ADD_SUB_ASYNCrst ( X , Z  , CARRYIN , opmode_5 ,  opmode_7 , CARRYOUT , CARRYOUTF , P ,
                               CLK , rst_a  , C_ENABLE ) ;

parameter  POST_ADD_SUB_width = 48 ;
parameter  CARRYINREG         = 1 ;
parameter  CARRYOUTREG        = 1 ;
parameter  CARRYINSEL         = "OPMODE5" ;

input  CLK , rst_a , C_ENABLE ;
input  [POST_ADD_SUB_width-1:0] X , Z   ;
input  CARRYIN   , opmode_5 , opmode_7  ;

output CARRYOUT , CARRYOUTF ;
output reg [POST_ADD_SUB_width-1:0] P  ;

// internal signals
wire  CIN_POST_ADD_SUB   ;
reg   COUT_POST_ADD_SUB  ;
wire  CARRY_CASCADE ;

MUX_ASYNCRST #( .data_width(1) , .sel(CARRYINREG)) CYI ( .in(CARRY_CASCADE) , .out(CIN_POST_ADD_SUB) , .CLK(CLK) , .rst_a(rst_a) ,.C_ENABLE(C_ENABLE)) ;

MUX_ASYNCRST #( .data_width(1) , .sel(CARRYOUTREG)) CYO ( .in(COUT_POST_ADD_SUB) , .out(CARRYOUT) , .CLK(CLK) , .rst_a(rst_a),.C_ENABLE(C_ENABLE)) ;


assign CARRY_CASCADE  = (CARRYINSEL=="OPMODE5")? opmode_5 : (CARRYINSEL=="CARRYIN")? CARRYIN : 0 ;


always @( X , Z , CIN_POST_ADD_SUB , opmode_7  ) begin
    if (~opmode_7)                                  // 0 > ADD
        {COUT_POST_ADD_SUB , P } = X + Z + CIN_POST_ADD_SUB ;
    else
        {COUT_POST_ADD_SUB , P } = Z-(X+CIN_POST_ADD_SUB) ;
end

assign CARRYOUTF = CARRYOUT ;

endmodule
```

```
1
2  // rst >> ASYNC ;
3
4  module DSP48A1_ASYNC ( A , B , C , D , BCIN , PCIN , opmode , CLK , CARRYIN ,
5                         CEA  , CEB , CEC   , CED  , CECARRYIN  , CEOPMODE  , CEM  , CEP  ,
6                         RSTA , RSTB , RSTC , RSTD , RSTCARRYIN , RSTOPMODE , RSTM , RSTP ,
7                         M , P , PCOUT , CARRYOUT , CARRYOUTF , BCOUT) ;
8  parameter A0REG       = 0  ;
9  parameter B0REG       = 0  ;
10 parameter A1REG       = 1  ;
11 parameter B1REG       = 1  ;
12 parameter CREG        = 1  ;
13 parameter DREG        = 1  ;
14 parameter MREG        = 1  ;
15 parameter PREG        = 1  ;
16 parameter CARRYINREG  = 1  ;
17 parameter CARRYOUTREG = 1  ;
18 parameter OPMODEREG   = 1  ;
19 parameter CARRYINSEL  = "OPMODE5" ;
20 parameter B_INPUT     = "DIRECT"  ;
21 parameter RSTTYPE     = "SYNC"    ;
22
23 input [17:0] A , B , BCIN , D ;
24 input [47:0] C , PCIN ;
25 input [7:0] opmode ;
26 input CLK , CARRYIN ;
27 input CEA , CEB , CEC , CED , CECARRYIN , CEOPMODE , CEM , CEP ;
28 input RSTA , RSTB , RSTC , RSTD , RSTCARRYIN , RSTOPMODE , RSTM , RSTP ;
29
30 output [35:0] M ;
31 output [47:0] P , PCOUT ;
32 output CARRYOUT , CARRYOUTF ;
33 output [17:0] BCOUT ;
34 reg    [7:0] opmodereg ;
35
36 wire   [17:0] B0_SEL ;  // out of mux that sel between B , BCIN ;
37
38 // any signal that is output of MUX_ASYNCRST module will take suffix (_IN)
39 wire [17:0] A0_IN , A1_IN , B0_IN , D_IN , B1_IN;
40 wire [47:0] C_IN ;
41 wire [17:0] pre_ADD_SUB_out ;
42 wire [17:0] B1 ;
43 wire [47:0] DAB ;
44 wire [35:0] multi_out , M ;
45 wire [47:0] P , PCOUT , POST_OUT ;
46 reg  [47:0] X , Z  ;
47  // generate opmode_reg ;
48 always @(posedge CLK or  posedge RSTOPMODE ) begin
49     if (RSTOPMODE)
50         opmodereg <= 8'b0 ;
51     else begin
52         if(CEOPMODE)
53             opmodereg <= opmode ;
54     end
55 end
56 assign B0_SEL = (B_INPUT == "DIRECT")? B : BCIN ;
57
58 //A_0
59 MUX_ASYNCRST #(.data_width(18),.sel(A0REG)) INS_A0  (.in(A),.out(A0_IN),.CLK(CLK),.rst_a(RSTA),.C_ENABLE(CEA)) ;
60
61 //A_1
62 MUX_ASYNCRST #(.data_width(18),.sel(A1REG)) INS_A1  (.in(A0_IN),.out(A1_IN),.CLK(CLK),.rst_a(RSTA),.C_ENABLE(CEA)) ;
63
64 //B_0
65 MUX_ASYNCRST #(.data_width(18),.sel(B0REG)) INS_B0  (.in(B0_SEL),.out(B0_IN),.CLK(CLK),.rst_a(RSTB),.C_ENABLE(CEB)) ;
66
67 //C
68 MUX_ASYNCRST #(.data_width(48),.sel(CREG)) INS_C  (.in(C),.out(C_IN),.CLK(CLK),.rst_a(RSTC),.C_ENABLE(CEC)) ;
69
70 //D
71 MUX_ASYNCRST #(.data_width(18),.sel(DREG)) INS_D (.in(D),.out(D_IN),.CLK(CLK),.rst_a(RSTD),.C_ENABLE(CED)) ;
72
73 // PRE-ADD_SUB ON B , D  ;
74 assign pre_ADD_SUB_out = (opmodereg[6])? (D_IN - B0_IN) : (D_IN + B0_IN) ;
75
76 // mux
77 assign B1 = (opmodereg[4])? pre_ADD_SUB_out : B0_IN ;
78
79 // B_1
80 MUX_ASYNCRST #(.data_width(18),.sel(B1REG)) INS_B1  (.in(B1),.out(B1_IN),.CLK(CLK),.rst_a(RSTB),.C_ENABLE(CEB)) ;
81
82 assign BCOUT = B1_IN ;
83
84 assign DAB = { D_IN[11:0] , A1_IN[17:0] , B1_IN[17:0] } ;
85
86 assign multi_out = A1_IN * B1_IN ;
87
88 //multiout ins
89 MUX_ASYNCRST #(.data_width(36),.sel(MREG)) multiplier (.in(multi_out),.out(M),.CLK(CLK),.rst_a(RSTM),.C_ENABLE(CEM)) ;
90
91
92 always @(*) begin
93     case (opmodereg[1:0])
94     0 : X = 0   ;
95     1 : X = M   ;
96     2 : X = P   ;
97     3 : X = DAB ;
98     endcase
99     case(opmodereg[3:2])
100        0 : Z = 0    ;
101        1 : Z = PCIN ;
102        2 : Z = P    ;
103        3 : Z = C_IN ;
104    endcase
105 end
106
107 POST_ADD_SUB_ASYNCrst #(.CARRYINREG(CARRYINREG),.CARRYOUTREG(CARRYOUTREG),.CARRYINSEL(CARRYINSEL))
108                     POST_ADD_SUB (.X(X) , .Z(Z)  , .CARRYIN(CARRYIN) , .opmode_5(opmodereg[5]) , .opmode_7(opmodereg[7]) ,
109                                   .CARRYOUT(CARRYOUT) , .CARRYOUTF(CARRYOUTF) , .P(POST_OUT) ,.CLK(CLK) , .rst_a(RSTCARRYIN),.C_ENABLE(CECARRYIN)) ;
110
111 MUX_ASYNCRST #(.data_width(48),.sel(PREG)) INS_P  (.in(POST_OUT),.out(P),.CLK(CLK),.rst_a(RSTP),.C_ENABLE(CEP)) ;
112
113 assign PCOUT = P ;
114
115 endmodule
```

Sync_rst:



```verilog
module MUX_SYNCRST (in,out,CLK,rst,C_ENABLE) ;

parameter data_width = 1 ;
parameter sel=1 ;
input CLK , rst , C_ENABLE ;
input [data_width-1:0] in ;
output [data_width-1:0] out ;

reg [data_width-1:0] in_reg ;

always @(posedge CLK) begin
     if(rst)
          in_reg <= 0 ;
     else  begin
     if(C_ENABLE)
          in_reg <= in ;
     end
end

assign out = (sel)?in_reg:in ;

endmodule
```

```verilog

module POST_ADD_SUB_SYNCrst ( X , Z  , CARRYIN , opmode_5 ,  opmode_7 , CARRYOUT , CARRYOUTF , P ,
                              CLK , rst , C_ENABLE ) ;

parameter  POST_ADD_SUB_width = 48 ;
parameter  CARRYINREG         = 1 ;
parameter  CARRYOUTREG        = 1 ;
parameter  CARRYINSEL         = "OPMODE5" ;

input  CLK , rst , C_ENABLE ;
input  [POST_ADD_SUB_width-1:0] X , Z   ;
input  CARRYIN   , opmode_5 , opmode_7  ;

output CARRYOUT , CARRYOUTF ;
output reg [POST_ADD_SUB_width-1:0] P  ;

// internal signals
wire  CIN_POST_ADD_SUB   ;
reg   COUT_POST_ADD_SUB  ;
wire  CARRY_CASCADE ;

MUX_SYNCRST #( .data_width(1) , .sel(CARRYOUTREG)) CYI ( .in(CARRY_CASCADE) , .out(CIN_POST_ADD_SUB) , .CLK(CLK) , .rst(rst),.C_ENABLE(C_ENABLE)) ;

MUX_SYNCRST #( .data_width(1) , .sel(CARRYOUTREG)) CYO ( .in(COUT_POST_ADD_SUB) , .out(CARRYOUT) , .CLK(CLK) , .rst(rst),.C_ENABLE(C_ENABLE)) ;


assign CARRY_CASCADE  = (CARRYINSEL=="OPMODE5")? opmode_5 : (CARRYINSEL=="CARRYIN")? CARRYIN : 0 ;


always @( X , Z , CIN_POST_ADD_SUB , opmode_7  ) begin
    if (~opmode_7)                                  // 0 > ADD
        {COUT_POST_ADD_SUB , P } = X + Z + CIN_POST_ADD_SUB ;
    else
        {COUT_POST_ADD_SUB , P } = Z-(X+CIN_POST_ADD_SUB) ;
end

assign CARRYOUTF = CARRYOUT ;

endmodule
```

```verilog

// rst >> ASYNC ;

module DSP48A1_SYNC ( A , B , C , D , BCIN , PCIN , opmode , CLK , CARRYIN ,
                      CEA  , CEB , CEC   , CED  , CECARRYIN  , CEOPMODE  , CEM  , CEP  ,
                      RSTA , RSTB , RSTC , RSTD , RSTCARRYIN , RSTOPMODE , RSTM , RSTP ,
                      M , P , PCOUT , CARRYOUT , CARRYOUTF , BCOUT) ;
parameter A0REG       = 0  ;
parameter B0REG       = 0  ;
parameter A1REG       = 1  ;
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" ;
parameter B_INPUT     = "DIRECT"  ;
parameter RSTTYPE     = "SYNC"    ;

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

output [35:0] M ;
output [47:0] P , PCOUT ;
output CARRYOUT , CARRYOUTF ;
output [17:0] BCOUT ;
reg    [7:0] opmodereg ;

wire   [17:0] B0_SEL ;  // out of mux that sel between B , BCIN ;

// any signal that is output of MUX_ASYNCRST module will take suffix (_IN)
wire [17:0] A0_IN , A1_IN , B0_IN , D_IN , B1_IN;
wire [47:0] C_IN ;
wire [17:0] pre_ADD_SUB_out ;
wire [17:0] B1 ;
wire [47:0] DAB ;
wire [35:0] multi_out , M ;
wire [47:0] P , PCOUT , POST_OUT ;
reg  [47:0] X , Z  ;
 // generate opmode_reg ;
always @(posedge CLK or  posedge RSTOPMODE ) begin
    if (RSTOPMODE)
        opmodereg <= 8'b0 ;
    else begin
        if(CEOPMODE)
          opmodereg <= opmode ;
    end
end
assign B0_SEL = (B_INPUT == "DIRECT")? B : BCIN ;

//A_0
MUX_SYNCRST #(.data_width(18),.sel(A0REG)) INS_A0  (.in(A),.out(A0_IN),.CLK(CLK),.rst(RSTA),.C_ENABLE(CEA)) ;

//A_1
MUX_SYNCRST #(.data_width(18),.sel(A1REG)) INS_A1  (.in(A0_IN),.out(A1_IN),.CLK(CLK),.rst(RSTA),.C_ENABLE(CEA)) ;

//B_0
MUX_SYNCRST #(.data_width(18),.sel(B0REG)) INS_B0  (.in(B0_SEL),.out(B0_IN),.CLK(CLK),.rst(RSTB),.C_ENABLE(CEB)) ;

//C
MUX_SYNCRST #(.data_width(48),.sel(CREG)) INS_C  (.in(C),.out(C_IN),.CLK(CLK),.rst(RSTC),.C_ENABLE(CEC)) ;

//D
MUX_SYNCRST #(.data_width(18),.sel(DREG)) INS_D (.in(D),.out(D_IN),.CLK(CLK),.rst(RSTD),.C_ENABLE(CED)) ;

// PRE-ADD_SUB ON B , D  ;
assign pre_ADD_SUB_out = (opmodereg[6])? (D_IN - B0_IN) : (D_IN + B0_IN) ;

// mux
assign B1 = (opmodereg[4])? pre_ADD_SUB_out : B0_IN ;

// B_1
MUX_SYNCRST #(.data_width(18),.sel(B1REG)) INS_B1  (.in(B1),.out(B1_IN),.CLK(CLK),.rst(RSTB),.C_ENABLE(CEB)) ;

assign BCOUT = B1_IN ;

assign DAB = { D_IN[11:0] , A1_IN[17:0] , B1_IN[17:0] } ;

assign multi_out = A1_IN * B1_IN ;

//multiout ins
MUX_SYNCRST #(.data_width(36),.sel(MREG)) multiplier (.in(multi_out),.out(M),.CLK(CLK),.rst(RSTM),.C_ENABLE(CEM)) ;


always @(*) begin
    case (opmodereg[1:0])
    0 : X = 0   ;
    1 : X = M   ;
    2 : X = P   ;
    3 : X = DAB ;
    endcase
    case(opmodereg[3:2])
        0 : Z = 0    ;
        1 : Z = PCIN ;
        2 : Z = P    ;
        3 : Z = C_IN ;
    endcase
end

POST_ADD_SUB_SYNCrst #(.CARRYINREG(CARRYINREG),.CARRYOUTREG(CARRYOUTREG),.CARRYINSEL(CARRYINSEL))
                     POST_ADD_SUB (.X(X) , .Z(Z)  , .CARRYIN(CARRYIN) , .opmode_5(opmodereg[5]) , .opmode_7(opmodereg[7]) ,
                                   .CARRYOUT(CARRYOUT) , .CARRYOUTF(CARRYOUTF) , .P(POST_OUT) ,.CLK(CLK) , .rst(RSTCARRYIN),.C_ENABLE(CECARRYIN)) ;

MUX_SYNCRST #(.data_width(48),.sel(PREG)) INS_P  (.in(POST_OUT),.out(P),.CLK(CLK),.rst(RSTP),.C_ENABLE(CEP)) ;

assign PCOUT = P ;

endmodule
```

DSP_DESIGN_CODE

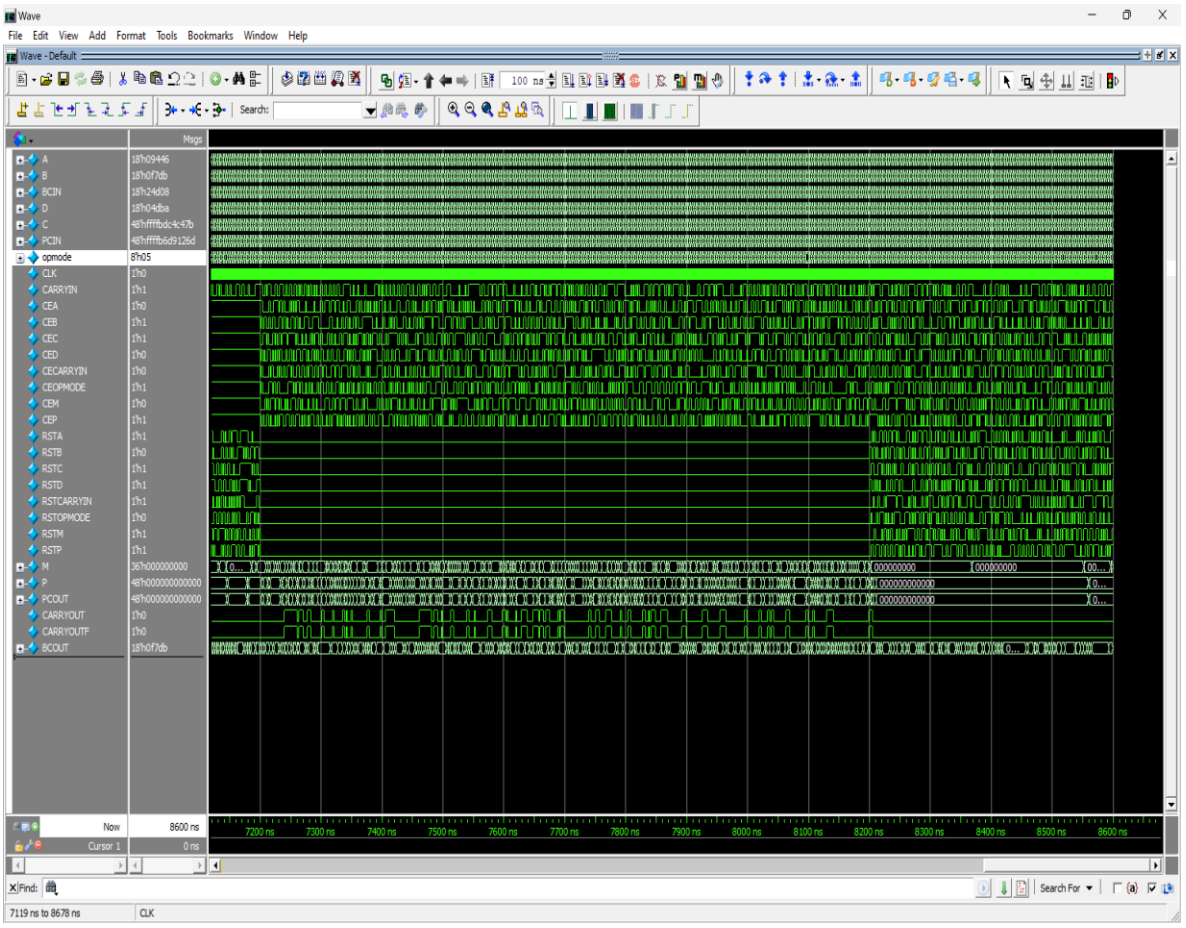
```
module DSP48A1 ( A , B , C , D , BCIN , PCIN , opmode , CLK , CARRYIN ,
                 CEA  , CEB , CEC   , CED  , CECARRYIN  , CEOPMODE  , CEM  , CEP  ,
                 RSTA , RSTB , RSTC , RSTD , RSTCARRYIN , RSTOPMODE , RSTM , RSTP ,
                 M , P , PCOUT , CARRYOUT , CARRYOUTF , BCOUT) ;
parameter A0REG       = 0  ;
parameter B0REG       = 0  ;
parameter A1REG       = 1  ;
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" ;
parameter B_INPUT     = "DIRECT"  ;
parameter RSTTYPE     = "SYNC"    ;

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

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

generate
    case(RSTTYPE)
        "ASYNC" : begin
            DSP48A1_ASYNC  #(.A0REG(A0REG),.B0REG(B0REG),.A1REG(A1REG),.B1REG(B1REG),.CREG(CREG),.DREG(DREG),.MREG(MREG),
                             .PREG(PREG),.CARRYINREG(CARRYINREG),.CARRYOUTREG(CARRYOUTREG),.OPMODEREG(OPMODEREG),
                             .CARRYINSEL(CARRYINSEL),.B_INPUT(B_INPUT),.RSTTYPE(RSTTYPE)  )
            DSP_ASYNC  ( .A(A) , .B(B) , .C(C) , .D(D) , .BCIN(BCIN) , .PCIN(PCIN) ,
                                                 .opmode(opmode) , .CLK(CLK) , .CARRYIN(CARRYIN) , .CEA(CEA) ,
                                                 .CEB(CEB) , .CEC(CEC) , .CED(CED) , .CECARRYIN(CECARRYIN)  ,
                                                 .CEOPMODE(CEOPMODE) , .CEM(CEM) ,.CEP (CEP) , .RSTA(RSTA) ,
                                                 .RSTB(RSTB) , .RSTC(RSTC) , .RSTD(RSTD) ,.RSTCARRYIN(RSTCARRYIN) ,
                                                 .RSTOPMODE(RSTOPMODE) , .RSTM(RSTM) , .RSTP(RSTP) , .M(M) , .P(P) ,
                                                 .PCOUT(PCOUT) , .CARRYOUT(CARRYOUT) , .CARRYOUTF(CARRYOUTF) ,
                                                 .BCOUT(BCOUT)) ;
        end

        "SYNC" : begin
            DSP48A1_SYNC #(.A0REG(A0REG),.B0REG(B0REG),.A1REG(A1REG),.B1REG(B1REG),.CREG(CREG),.DREG(DREG),.MREG(MREG),
                           .PREG(PREG),.CARRYINREG(CARRYINREG),.CARRYOUTREG(CARRYOUTREG),.OPMODEREG(OPMODEREG),
                           .CARRYINSEL(CARRYINSEL),.B_INPUT(B_INPUT),.RSTTYPE(RSTTYPE)  )
            DSP_ASYNC  ( .A(A) , .B(B) , .C(C) , .D(D) , .BCIN(BCIN) , .PCIN(PCIN) ,
                                                 .opmode(opmode) , .CLK(CLK) , .CARRYIN(CARRYIN) , .CEA(CEA) ,
                                                 .CEB(CEB) , .CEC(CEC) , .CED(CED) , .CECARRYIN(CECARRYIN)  ,
                                                 .CEOPMODE(CEOPMODE) , .CEM(CEM) ,.CEP (CEP) , .RSTA(RSTA) ,
                                                 .RSTB(RSTB) , .RSTC(RSTC) , .RSTD(RSTD) ,.RSTCARRYIN(RSTCARRYIN) ,
                                                 .RSTOPMODE(RSTOPMODE) , .RSTM(RSTM) , .RSTP(RSTP) , .M(M) , .P(P) ,
                                                 .PCOUT(PCOUT) , .CARRYOUT(CARRYOUT) , .CARRYOUTF(CARRYOUTF) ,
                                                 .BCOUT(BCOUT)) ;
        end
    endcase
endgenerate
endmodule
```

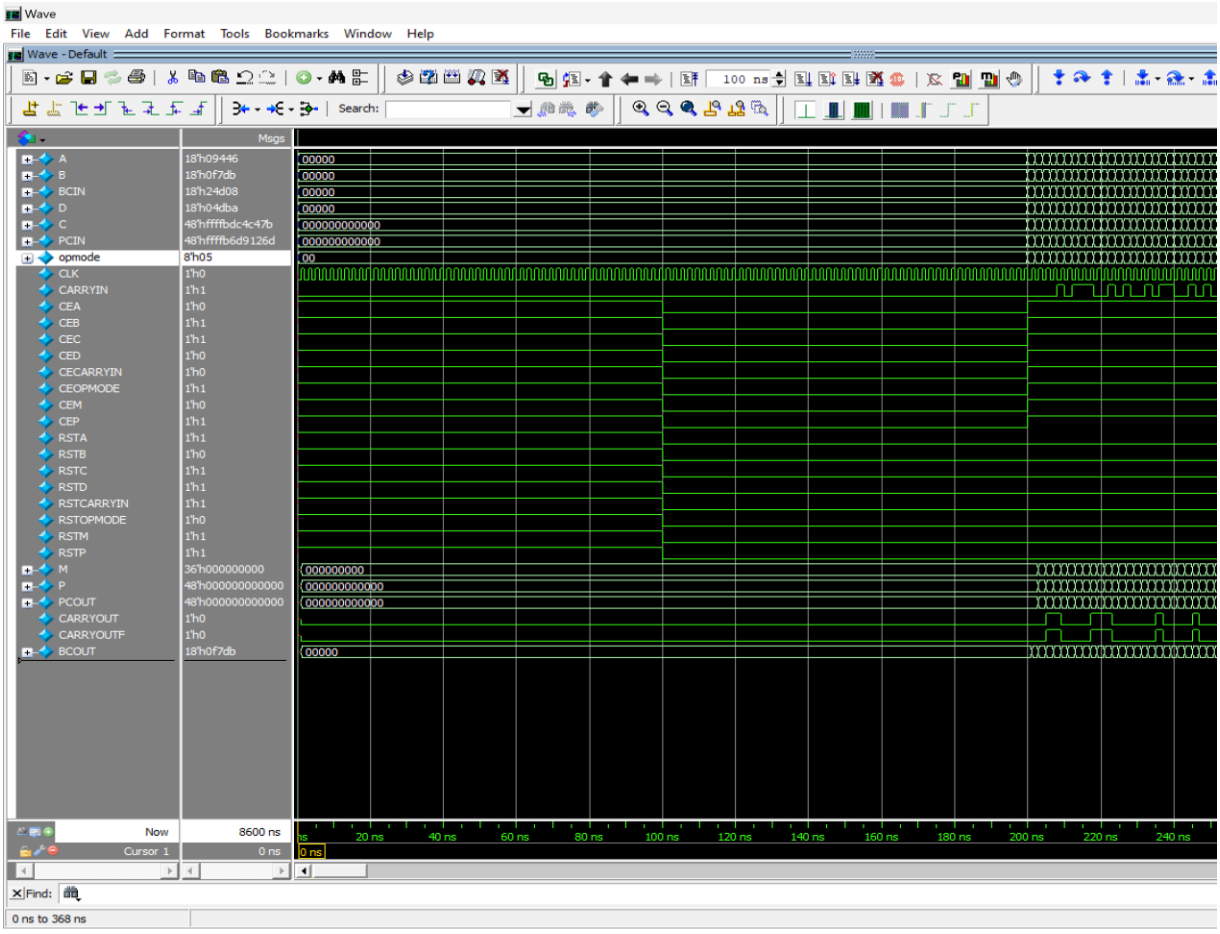
- TB :
- Code
- Do_file
- Simulation

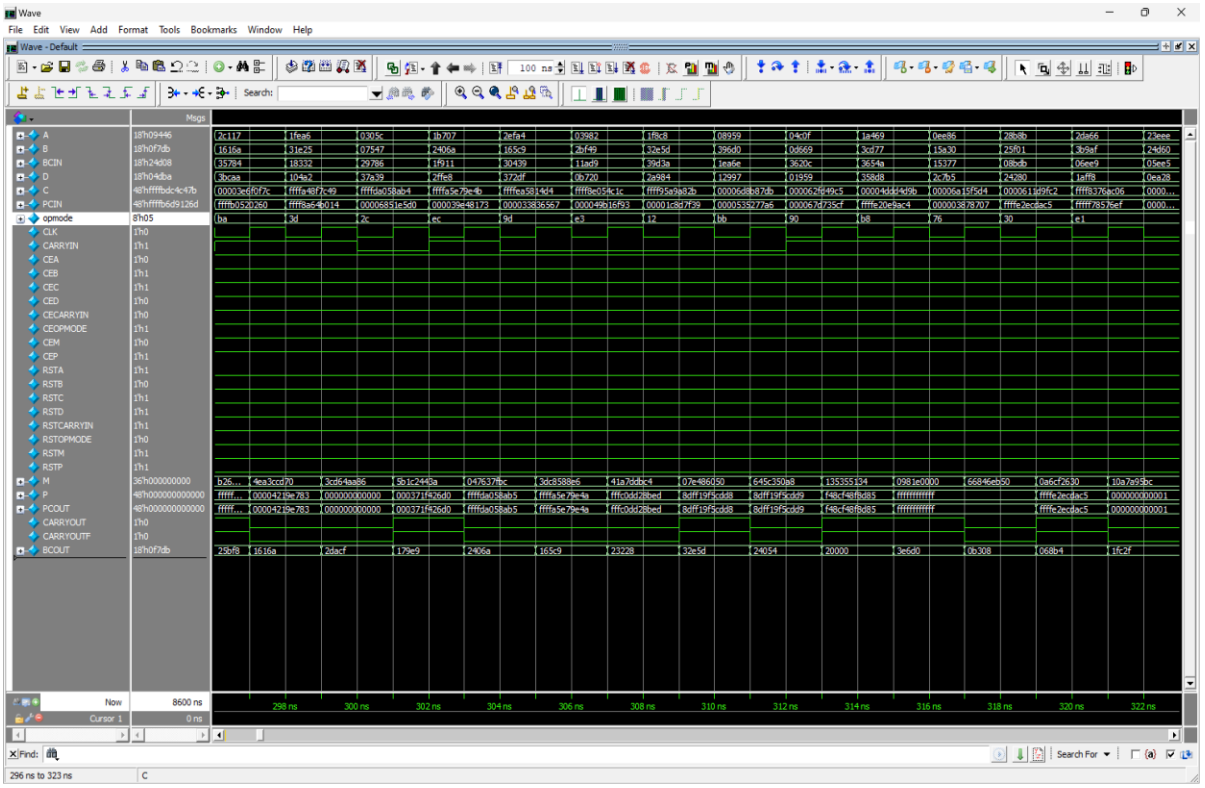
```
module DSP48A1_tb ;

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

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

DSP48A1 DUT ( .A(A) , .B(B)     , .C(C) , .D(D) , .BCIN(BCIN) , .PCIN(PCIN) , .opmode(opmode) ,
                    .CLK(CLK)   , .CARRYIN(CARRYIN) ,.CEA(CEA)   , .CEB(CEB) , .CEC(CEC)        ,
                    .CED(CED)   , .CECARRYIN(CECARRYIN)   , .CEOPMODE(CEOPMODE)   , .CEM(CEM)    ,
                    .CEP(CEP)   , .RSTA(RSTA) , .RSTB(RSTB) , .RSTC(RSTC) , .RSTD(RSTD)        ,
                    .RSTCARRYIN(RSTCARRYIN)   , .RSTOPMODE(RSTOPMODE) , .RSTM (RSTM)           ,
                    .RSTP(RSTP) , .M(M) , .P(P) , .PCOUT(PCOUT) , .CARRYOUT(CARRYOUT)          ,
                    .CARRYOUTF(CARRYOUTF) , .BCOUT(BCOUT)) ;

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

initial begin
    repeat(50) begin
        RSTA        =  1 ;
        RSTB        =  1 ;
        RSTC        =  1 ;
        RSTD        =  1 ;
        RSTCARRYIN  =  1 ;
        RSTOPMODE   =  1 ;
        RSTM        =  1 ;
        RSTP        =  1 ;
        CEA         = 1 ;
        CEB         = 1 ;
        CEC         = 1 ;
        CED         = 1 ;
        CECARRYIN   = 1 ;
        CEOPMODE    = 1 ;
        CEM         = 1 ;
        CEP         = 1 ;
        A       = 0 ;
        B       = 0 ;
        BCIN    = 0 ;
        D       = 0 ;
        C       = 0 ;
        PCIN    = 0 ;
        opmode  = 0 ;
        CARRYIN = 0 ;
        @(negedge CLK) ;
    end
    repeat(50) begin
        RSTA        =  0 ;
        RSTB        =  0 ;
        RSTC        =  0 ;
        RSTD        =  0 ;
        RSTCARRYIN  =  0 ;
        RSTOPMODE   =  0 ;
        RSTM        =  0 ;
        RSTP        =  0 ;
        CEA         = 0 ;
        CEB         = 0 ;
        CEC         = 0 ;
        CED         = 0 ;
        CECARRYIN   = 0 ;
        CEOPMODE    = 0 ;
        CEM         = 0 ;
        CEP         = 0 ;
        @(negedge CLK) ;
    end
        RSTA        =  0 ;
        RSTB        =  0 ;
        RSTC        =  0 ;
        RSTD        =  0 ;
        RSTCARRYIN  =  0 ;
        RSTOPMODE   =  0 ;
        RSTM        =  0 ;
        RSTP        =  0 ;
        CEA         = 1 ;
        CEB         = 1 ;
        CEC         = 1 ;
        CED         = 1 ;
        CECARRYIN   = 1 ;
        CEOPMODE    = 1 ;
        CEM         = 1 ;
        CEP         = 1 ;
    repeat (3000) begin
```

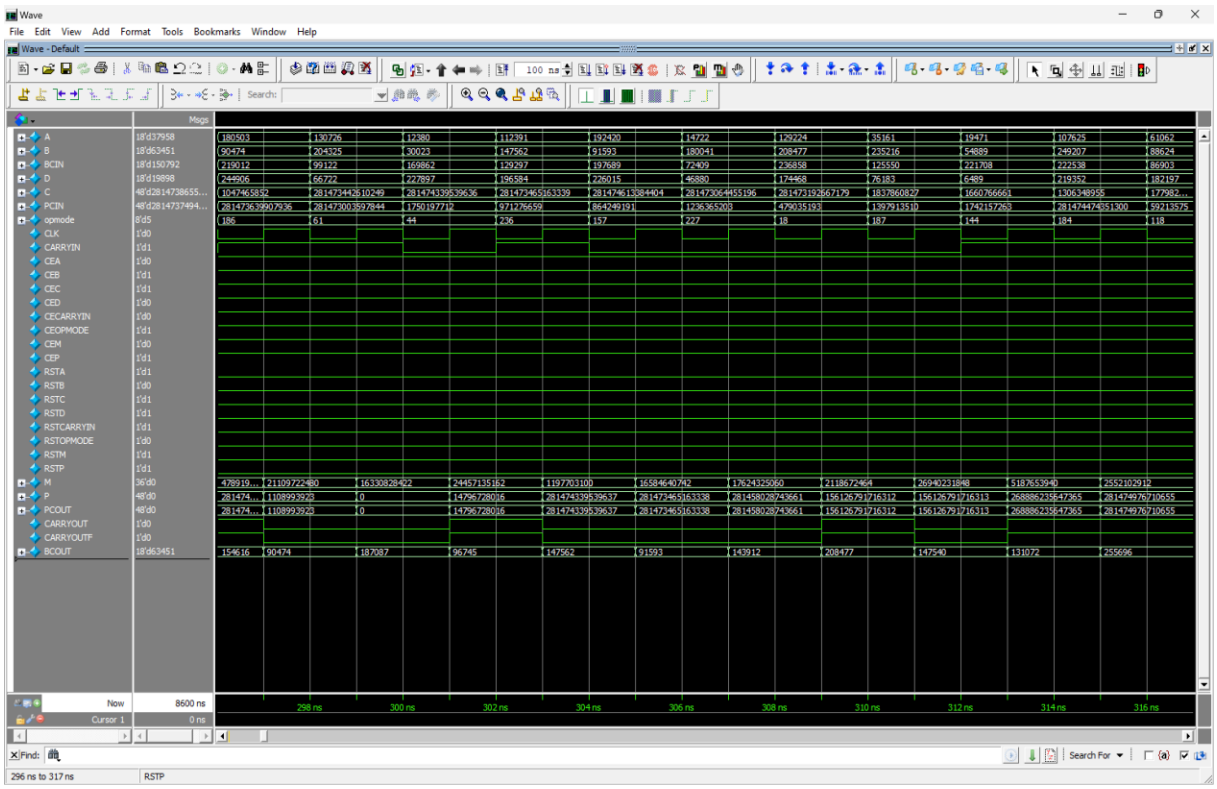
```
        CEP       = 1 ;
    repeat (3000) begin
        A       = $random ;
        B       = $random ;
        BCIN    = $random ;
        D       = $random ;
        C       = $random ;
        PCIN    = $random ;
        opmode  = $random ;
        CARRYIN = $random ;
        @(negedge CLK) ;
    end
    repeat (500) begin
        A       = $random ;
        B       = $random ;
        BCIN    = $random ;
        D       = $random ;
        C       = $random ;
        PCIN    = $random ;
        opmode  = $random ;
        CARRYIN = $random ;
        RSTA        =  $random ;
        RSTB        =  $random ;
        RSTC        =  $random ;
        RSTD        =  $random ;
        RSTCARRYIN  =  $random ;
        RSTOPMODE   =  $random ;
        RSTM        =  $random ;
        RSTP        =  $random ;
        @(negedge CLK) ;
    end
    repeat (500) begin
        A       = $random ;
        B       = $random ;
        BCIN    = $random ;
        D       = $random ;
        C       = $random ;
        PCIN    = $random ;
        opmode  = $random ;
        CARRYIN = $random ;
        RSTA          =  0    ;
        RSTB          =  0    ;
        RSTC          =  0    ;
        RSTD          =  0    ;
        RSTCARRYIN    =  0    ;
        RSTOPMODE     =  0    ;
        RSTM          =  0    ;
        RSTP          =  0    ;
        CEA       = $random ;
        CEB       = $random ;
        CEC       = $random ;
        CED       = $random ;
        CECARRYIN = $random ;
        CEOPMODE  = $random ;
        CEM       = $random ;
        CEP       = $random ;
        @(negedge CLK) ;
    end
    repeat (200) begin
        A       = $random ;
        B       = $random ;
        BCIN    = $random ;
        D       = $random ;
        C       = $random ;
        PCIN    = $random ;
        opmode  = $random ;
        CARRYIN = $random ;
        RSTA       =  $random  ;
        RSTB       =  $random  ;
        RSTC       =  $random  ;
        RSTD       =  $random  ;
        RSTCARRYIN =  $random  ;
        RSTOPMODE  =  $random  ;
        RSTM       =  $random  ;
        RSTP       =  $random  ;
        CEA       = $random ;
        CEB       = $random ;
        CEC       = $random ;
        CED       = $random ;
        CECARRYIN = $random ;
        CEOPMODE  = $random ;
        CEM       = $random ;
        CEP       = $random ;
        @(negedge CLK) ;
    end
    $stop ;
end
endmodule
```

```
quit -sim

vlib work

vlog DSP_COMPLETE_PROJECT.v  DSP_tb.v  DSP_ASYNC_complete_design.v MUX_ASYNCrst_MODULE.v POST_ADD_SUB_ASYNCrst_MODULE.v  DSP_SYNC_complete_design.v  MUX_SYNCrst_MODULE.v  POST_ADD_SUB_SYNCrst_MODULE.v

vsim -voptargs=+acc DSP48A1_tb

add wave *

run -all
```

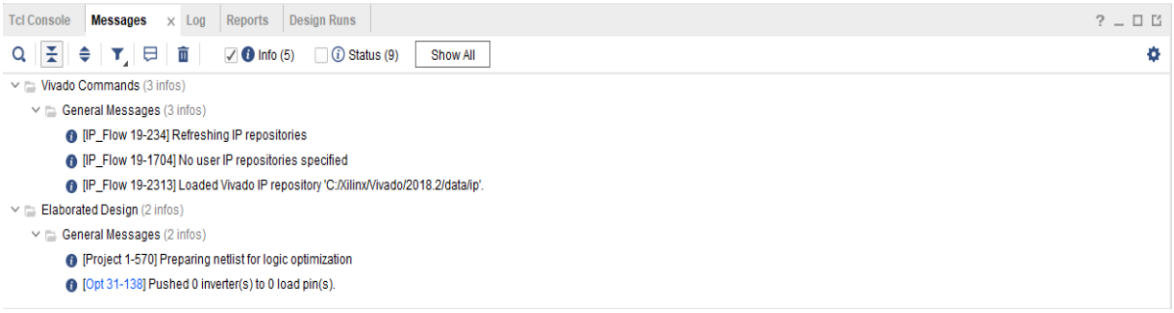


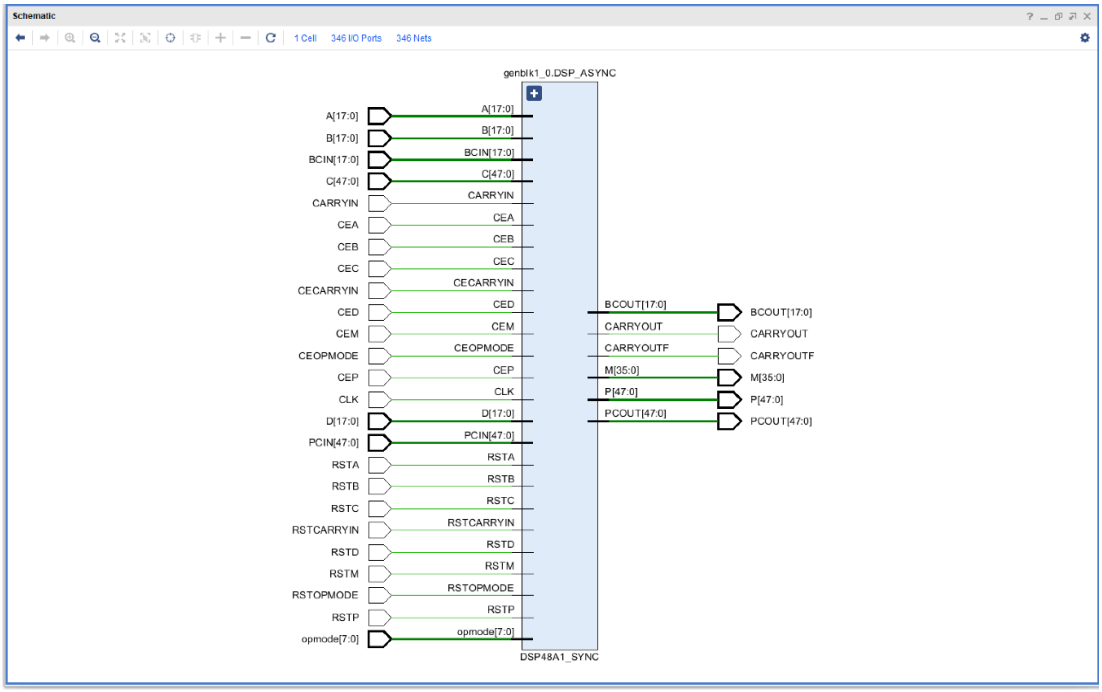

Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
A 18'h09446
B 18'h0f7db
BCIN 18'h24d08
D 18'h04dba
C 48'hffffbdc4c47b
PCIN 48'hffffb6d9126d
opmode 8'h05
CLK 1'h0
CARRYIN 1'h1
CEA 1'h0
CEB 1'h1
CEC 1'h1
CED 1'h0
CECARRYIN 1'h0
CEOPMODE 1'h1
CEM 1'h0
CEP 1'h1
RSTA 1'h1
RSTB 1'h0
RSTC 1'h1
RSTD 1'h1
RSTCARRYIN 1'h1
RSTOPMODE 1'h0
RSTM 1'h1
RSTP 1'h1
M 36'h000000000
P 48'h000000000000
PCOUT 48'h000000000000
CARRYOUT 1'h0
CARRYOUTF 1'h0
BCOUT 18'h0f7db
Now 8600 ns
Cursor 1 0 ns
20 ns 40 ns 60 ns 80 ns 100 ns 120 ns 140 ns 160 ns 180 ns 200 ns 220 ns 240 ns
Find:
0 ns to 368 ns

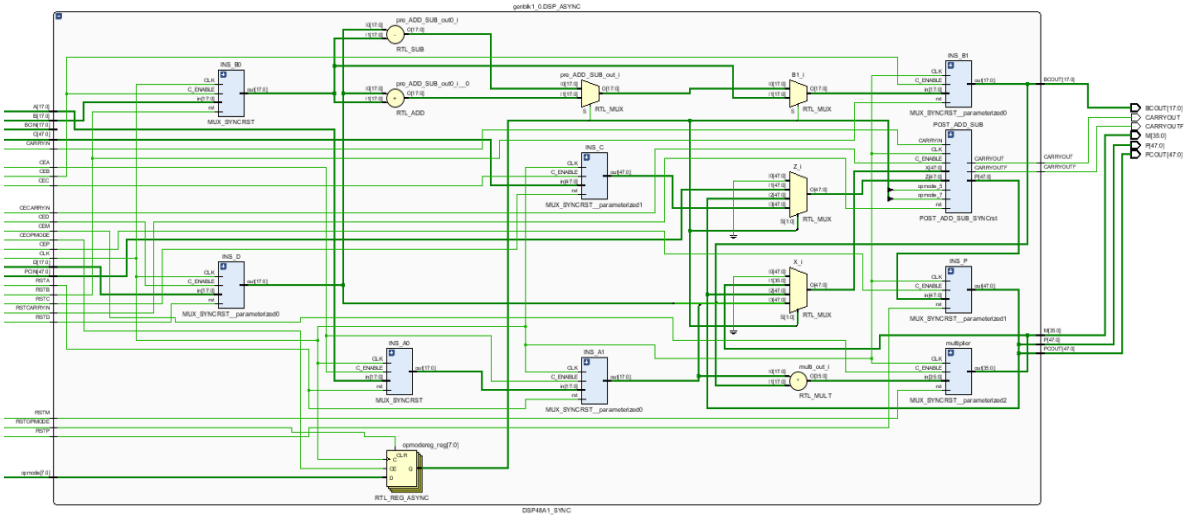

Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
Now 8600 ns
Cursor 1 0 ns
298 ns 300 ns 302 ns 304 ns 306 ns 308 ns 310 ns 312 ns 314 ns 316 ns 318 ns 320 ns 322 ns
Find:
Search For
296 ns to 323 ns

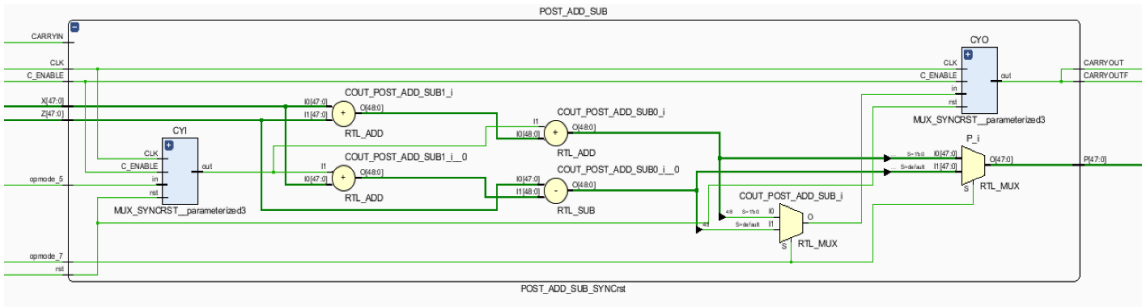

Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
A
B
BCIN
D
C
PCIN
opmode
CLK
CARRYIN
CEA
CEB
CEC
CED
CECARRYIN
CEOPMODE
CEM
CEP
RSTA
RSTB
RSTC
RSTD
RSTCARRYIN
RSTOPMODE
RSTM
RSTP
M
P
PCOUT
CARRYOUT
CARRYOUTF
BCOUT
Now
8600 ns
Cursor 1
0 ns
298 ns
300 ns
302 ns
304 ns
306 ns
308 ns
310 ns
312 ns
314 ns
316 ns
Find:
Search For
296 ns to 317 ns
RSTP

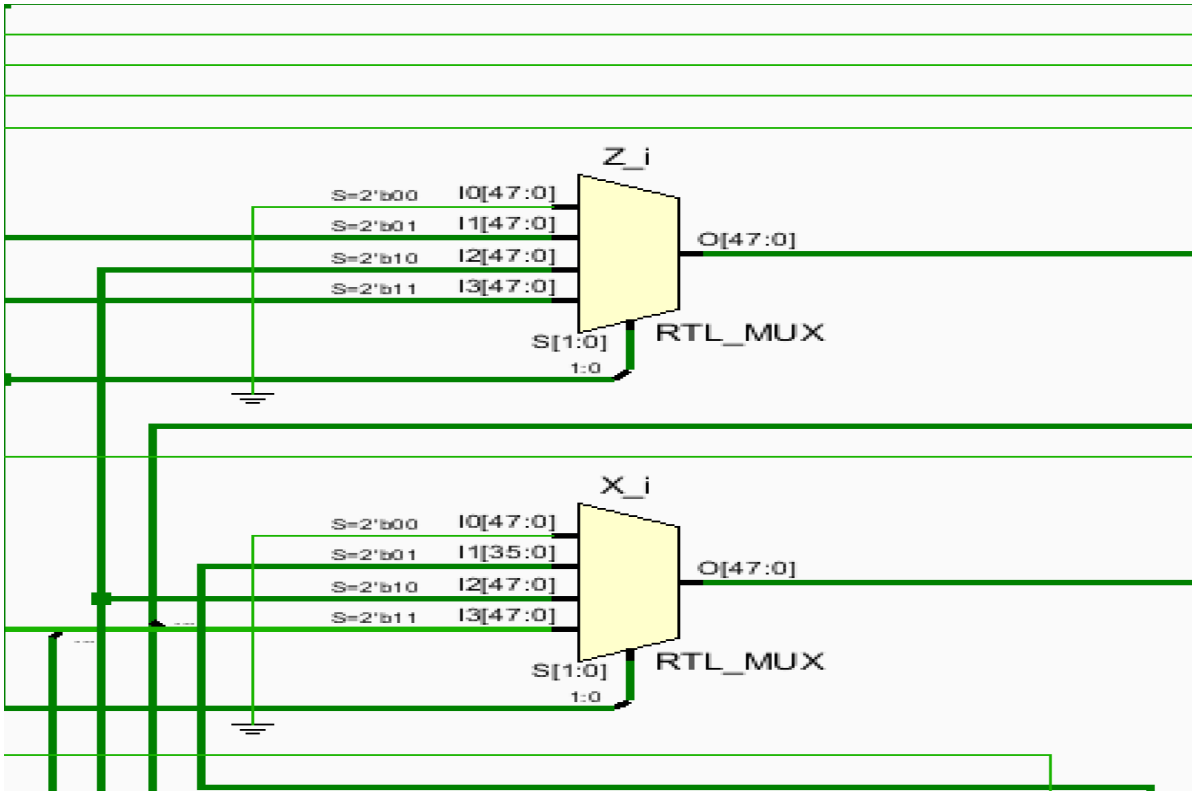
- Elaboration :
  - Message
  - Schematic

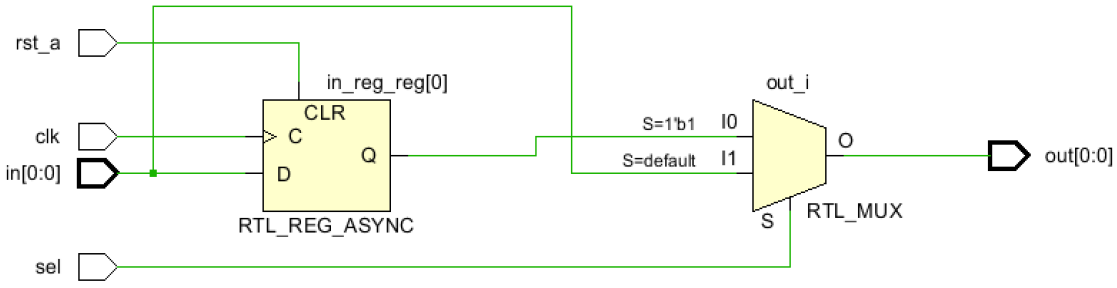




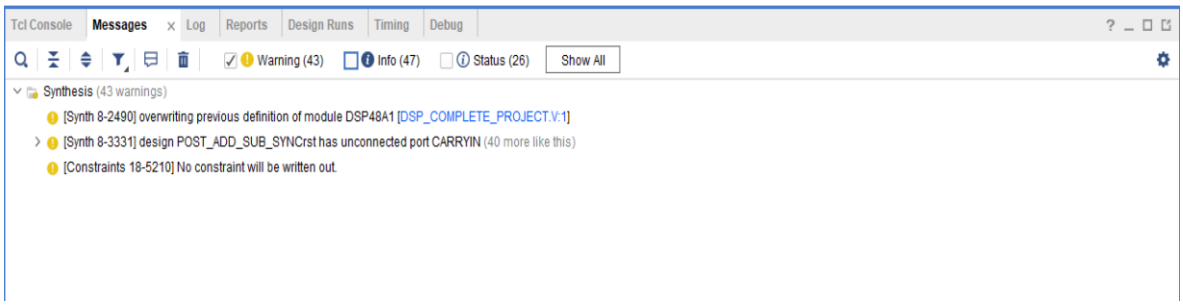

genblk1_0.DSP_ASYNC
pre_ADD_SUB_out0_i
RTL_SUB
pre_ADD_SUB_out0_i__0
RTL_ADD
pre_ADD_SUB_out_i
RTL_MUX
B1_i
INS_B1
MUX_SYNCRST__parameterized0
POST_ADD_SUB
POST_ADD_SUB_SYNCnst
INS_B0
MUX_SYNCRST
INS_C
MUX_SYNCRST__parameterized1
Z_i
INS_D
MUX_SYNCRST__parameterized0
X_i
INS_P
INS_A0
INS_A1
mult_out_i
RTL_MULT
multiplier
MUX_SYNCRST__parameterized2
opmodereg_reg[7:0]
RTL_REG_ASYNC
BCOUT[17:0]
CARRYOUT
CARRYOUTF
M[35:0]
P[47:0]
PCOUT[47:0]
DSP48A1_SYNC

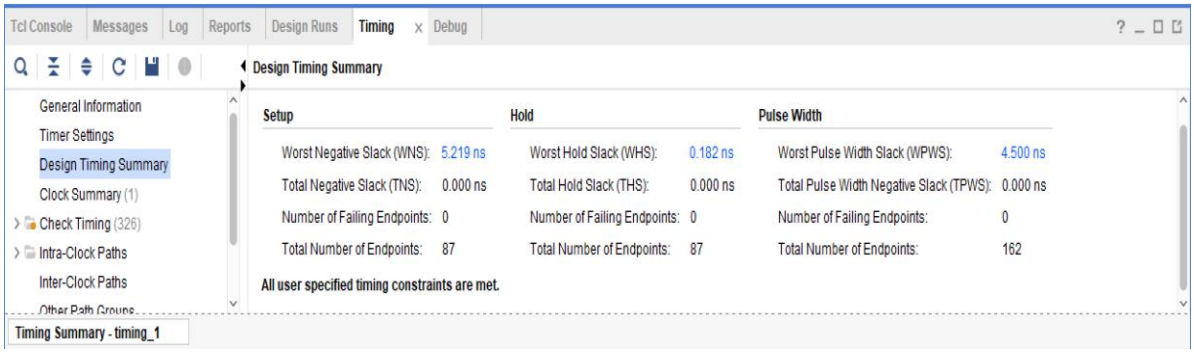

POST_ADD_SUB
CARRYIN
CLK
C_ENABLE
X[47:0]
Z[47:0]
opmode_5
opmode_7
rst
CYI
MUX_SYNCRST__parameterized3
COUT_POST_ADD_SUB1_i
RTL_ADD
COUT_POST_ADD_SUB1_i__0
RTL_ADD
COUT_POST_ADD_SUB0_i
RTL_ADD
COUT_POST_ADD_SUB0_i__0
RTL_SUB
COUT_POST_ADD_SUB_i
RTL_MUX
CYO
MUX_SYNCRST__parameterized3
P_i
RTL_MUX
CARRYOUT
CARRYOUTF
P[47:0]
POST_ADD_SUB_SYNCnst

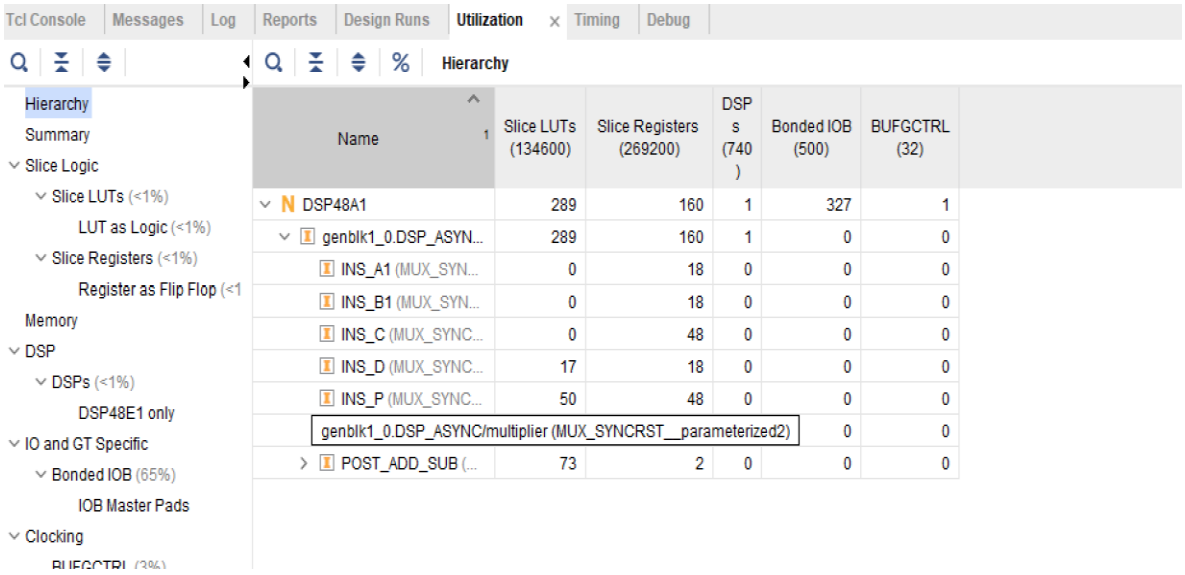

Z_i
S=2'b00
S=2'b01
S=2'b10
S=2'b11
I0[47:0]
I1[47:0]
I2[47:0]
I3[47:0]
O[47:0]
S[1:0]
1:0
RTL_MUX
X_i
S=2'b00
S=2'b01
S=2'b10
S=2'b11
I0[47:0]
I1[35:0]
I2[47:0]
I3[47:0]
O[47:0]
S[1:0]
1:0
RTL_MUX

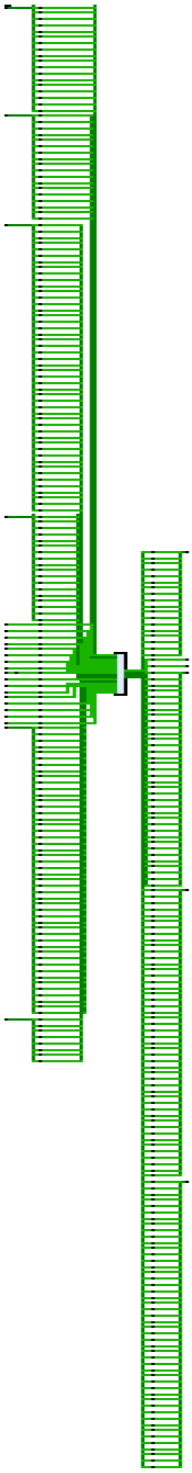

rst_a
clk
in[0:0]
sel
in_reg_reg[0]
CLR
C
D
Q
RTL_REG_ASYNC
out_i
S=1'b1
S=default
I0
I1
O
S
RTL_MUX
out[0:0]

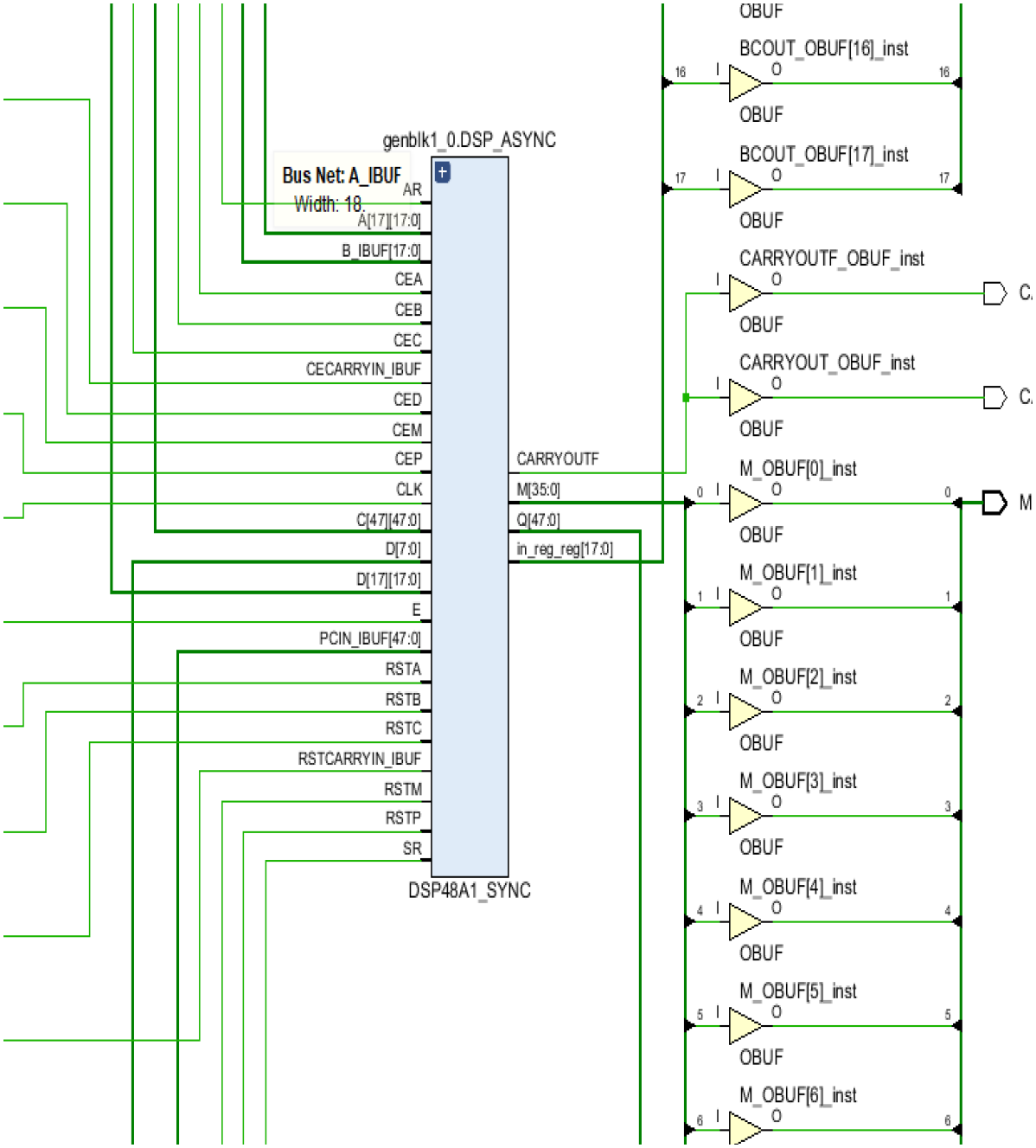
- Synthesis
  - -message
  - -schematic
  - -time report
  - -utilization report

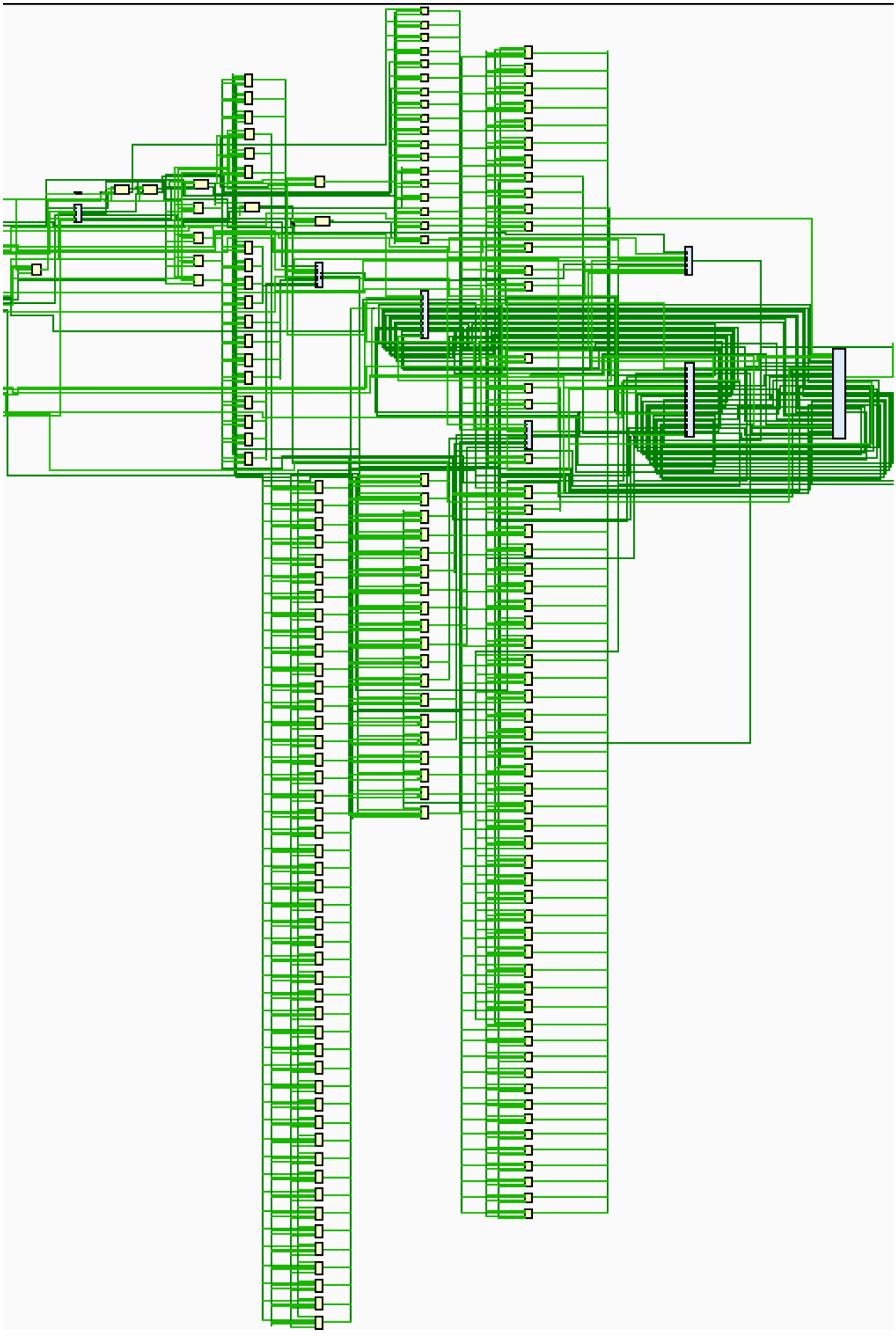

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Debug
Warning (43)
Info (47)
Status (26)
Show All
Synthesis (43 warnings)
[Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP_COMPLETE_PROJECT.V:1]
[Synth 8-3331] design POST_ADD_SUB_SYNCrst has unconnected port CARRYIN (40 more like this)
[Constraints 18-5210] No constraint will be written out.

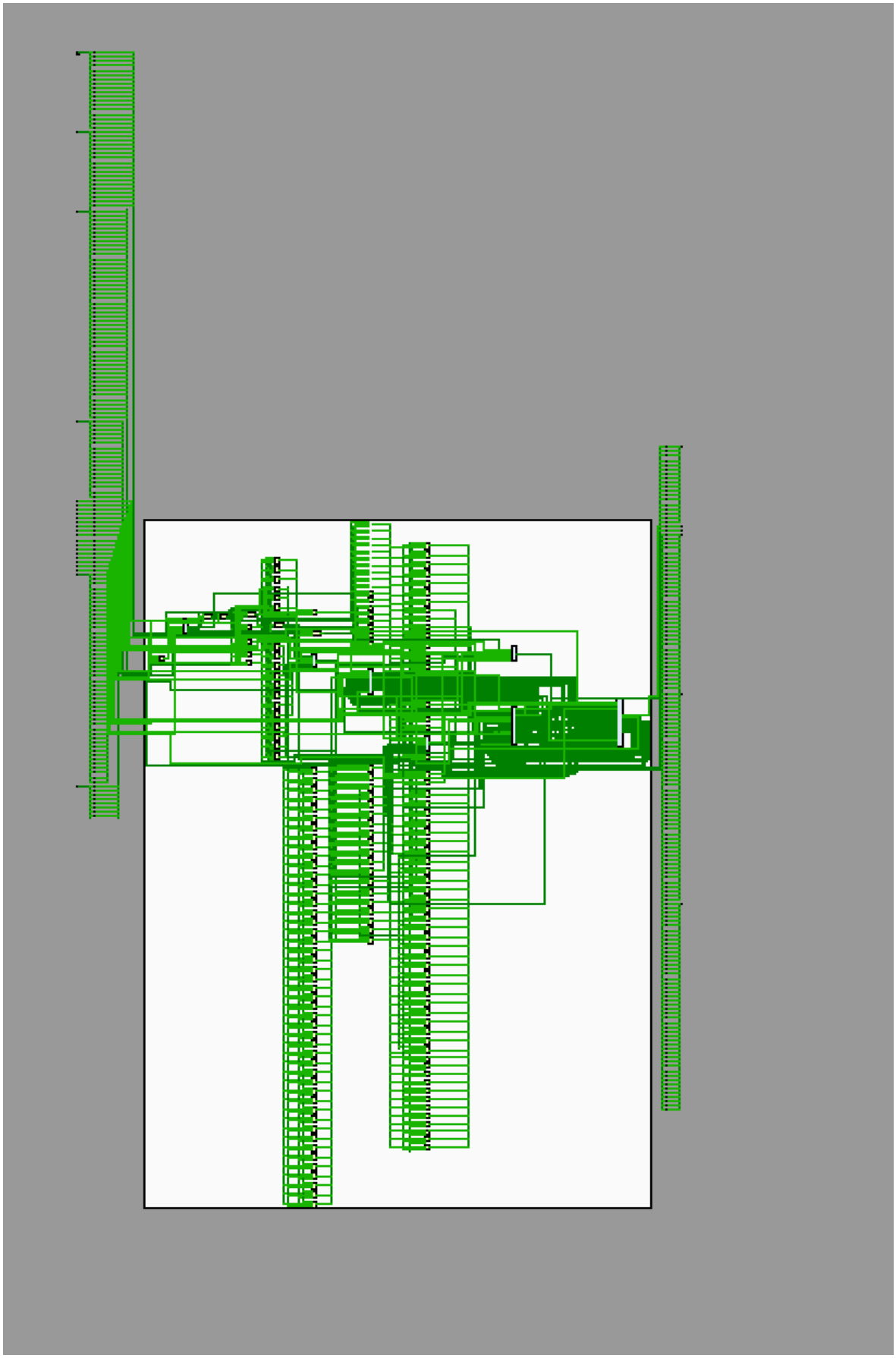

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Debug
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (326)
Intra-Clock Paths
Inter-Clock Paths
Setup
Worst Negative Slack (WNS): 5.219 ns
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 87
Hold
Worst Hold Slack (WHS): 0.182 ns
Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 87
Pulse Width
Worst Pulse Width Slack (WPWS): 4.500 ns
Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 162
All user specified timing constraints are met.
Timing Summary - timing_1


Tcl Console
Messages
Log
Reports
Design Runs
Utilization
Timing
Debug
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%)
Hierarchy
Name
Slice LUTs (134600)
Slice Registers (269200)
DSPs (740)
Bonded IOB (500)
BUFGCTRL (32)
DSP48A1 289 160 1 327 1
genblk1_0.DSP_ASYN... 289 160 1 0 0
INS_A1 (MUX_SYN... 0 18 0 0 0
INS_B1 (MUX_SYN... 0 18 0 0 0
INS_C (MUX_SYNC... 0 48 0 0 0
INS_D (MUX_SYNC... 17 18 0 0 0
INS_P (MUX_SYNC... 50 48 0 0 0
genblk1_0.DSP_ASYNC/multiplier (MUX_SYNCRST__parameterized2) 0 0
POST_ADD_SUB (... 73 2 0 0 0

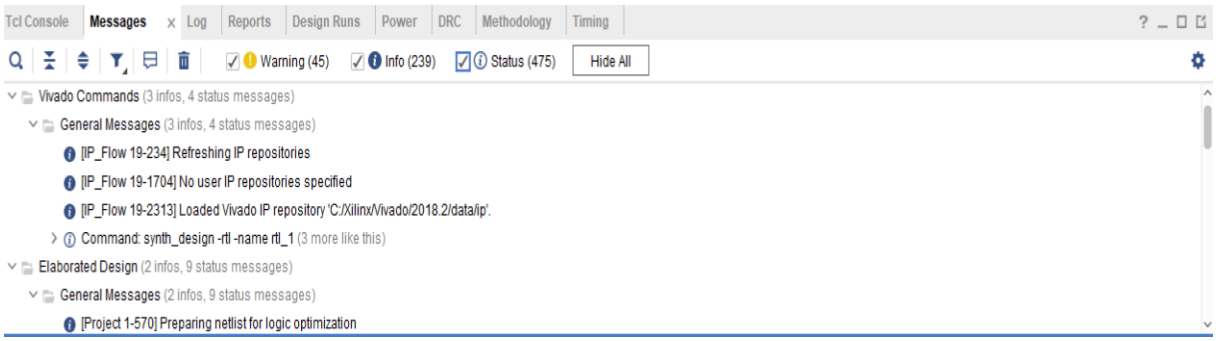


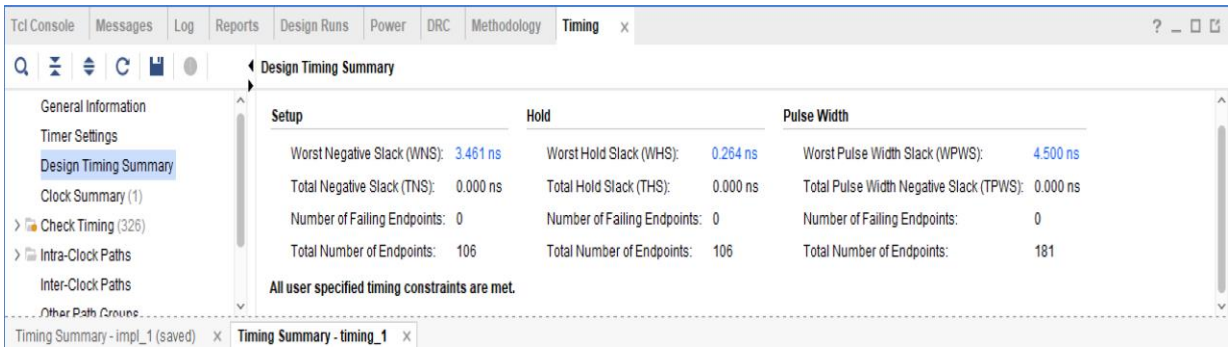

genblk1_0.DSP_ASYNC
Bus Net: A_IBUF
Width: 18.
AR
A[17][17:0]
B_IBUF[17:0]
CEA
CEB
CEC
CECARRYIN_IBUF
CED
CEM
CEP
CLK
C[47][47:0]
D[7:0]
D[17][17:0]
E
PCIN_IBUF[47:0]
RSTA
RSTB
RSTC
RSTCARRYIN_IBUF
RSTM
RSTP
SR
DSP48A1_SYNC
CARRYOUTF
M[35:0]
Q[47:0]
in_reg_reg[17:0]
OBUF
BCOUT_OBUF[16]_inst
BCOUT_OBUF[17]_inst
CARRYOUTF_OBUF_inst
CARRYOUT_OBUF_inst
M_OBUF[0]_inst
M_OBUF[1]_inst
M_OBUF[2]_inst
M_OBUF[3]_inst
M_OBUF[4]_inst
M_OBUF[5]_inst
M_OBUF[6]_inst

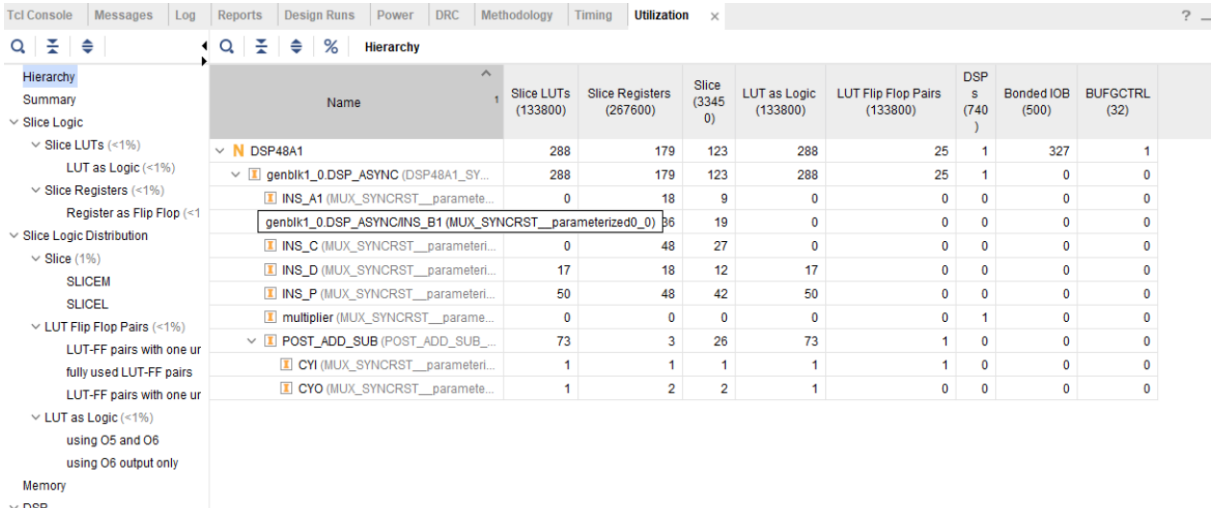


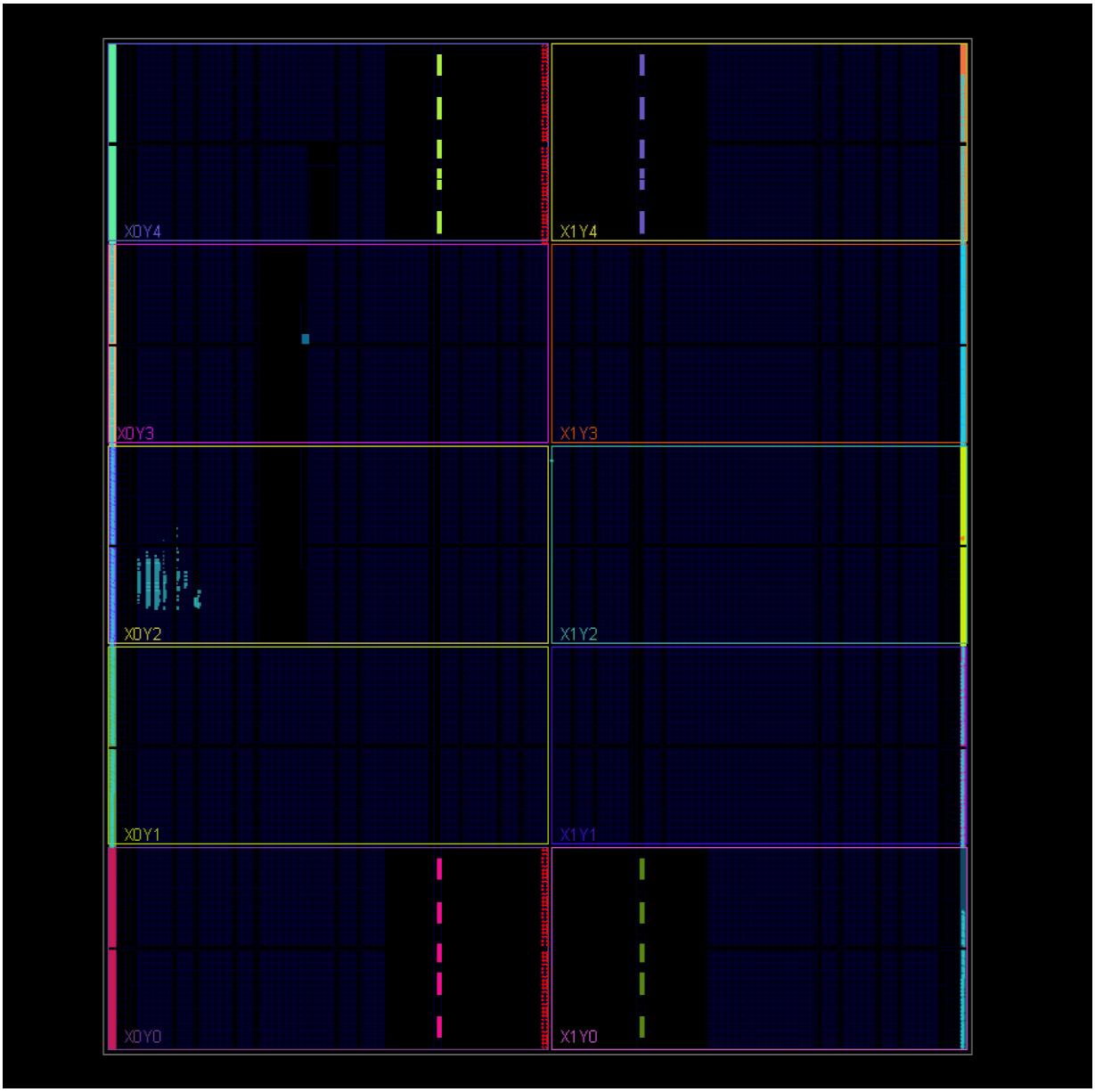


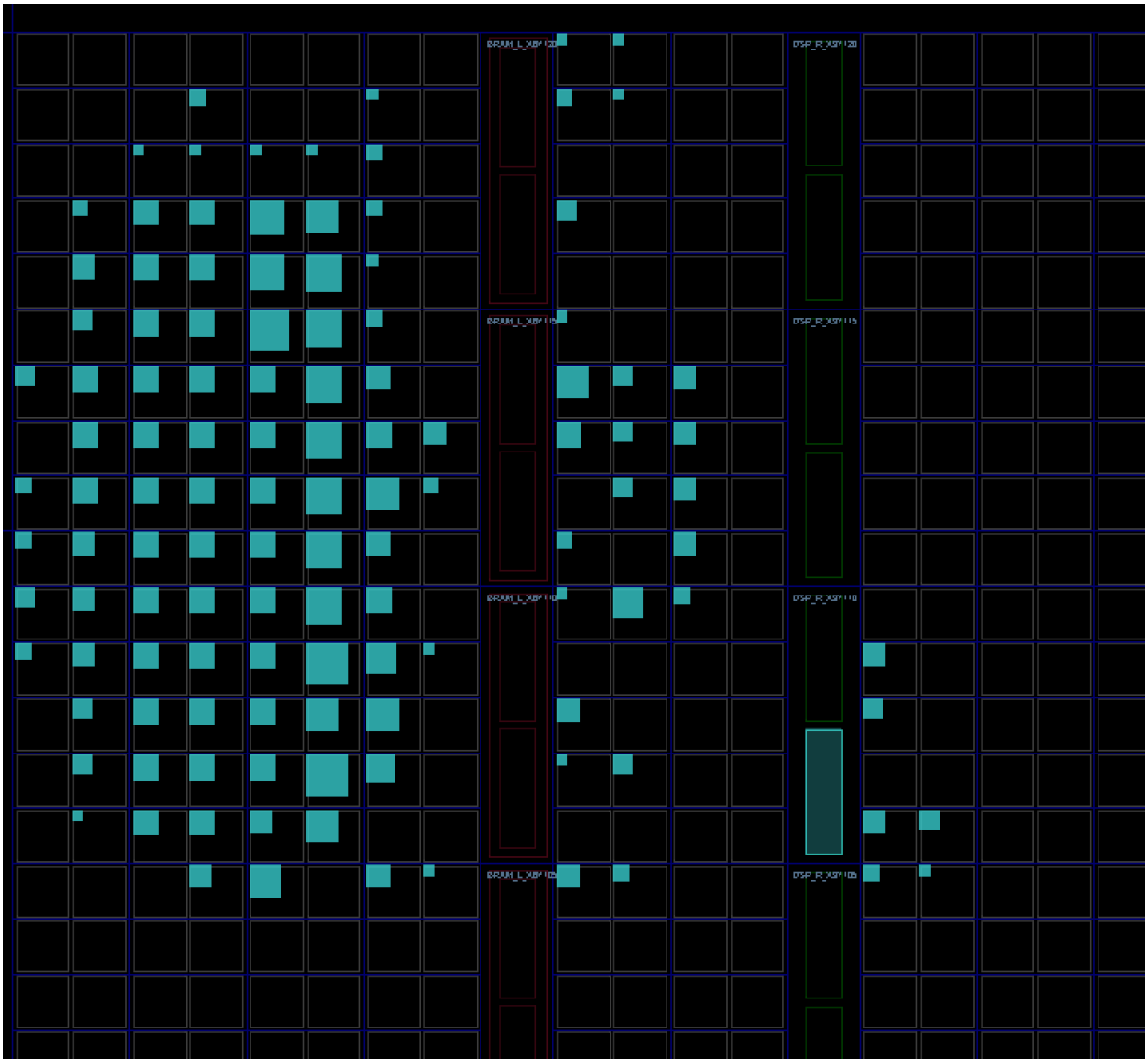
- Implementation
  - -message
  - -time report
  - -utilization report
  - -device

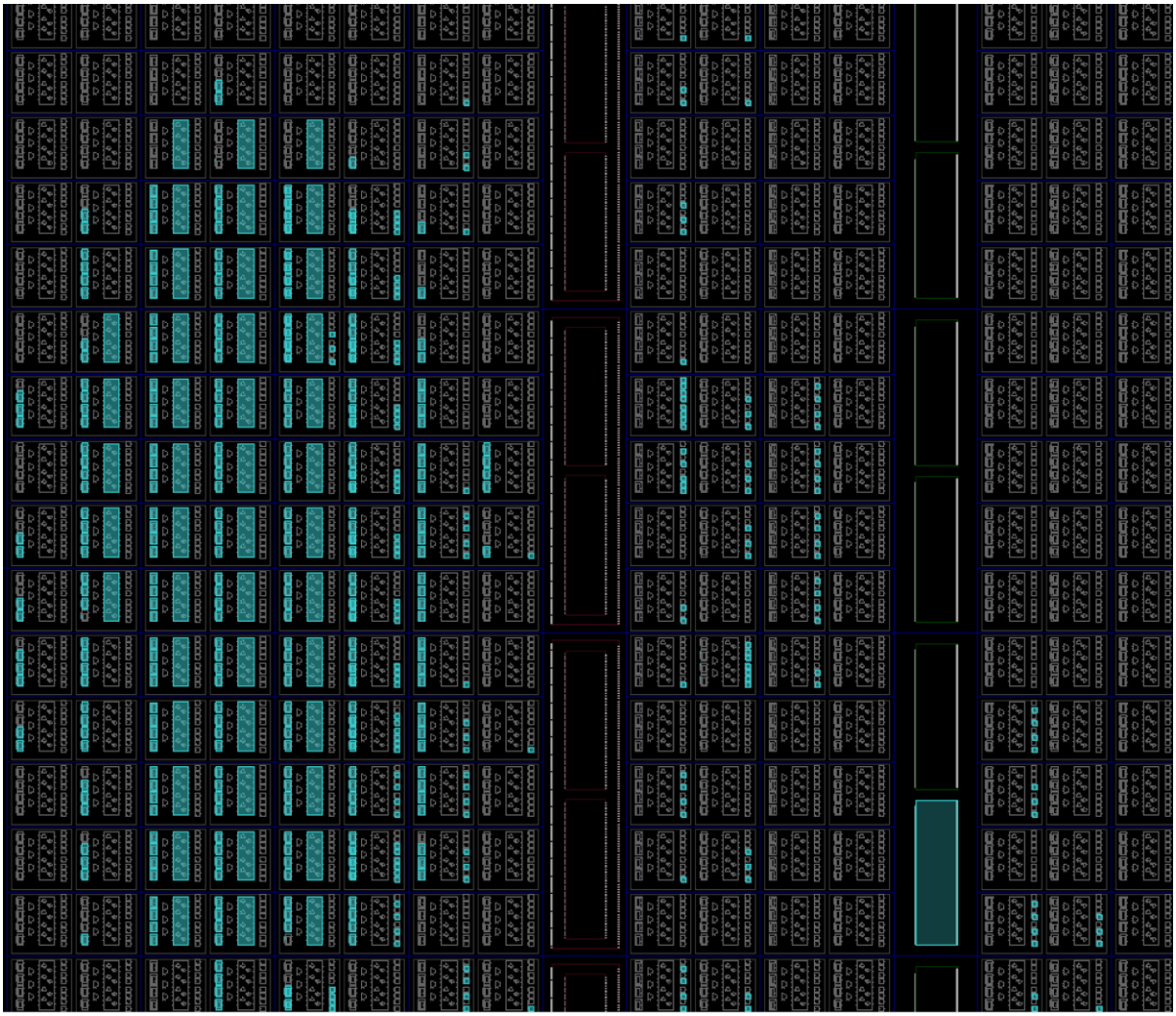

Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Warning (45)
Info (239)
Status (475)
Hide All
Vivado Commands (3 infos, 4 status messages)
General Messages (3 infos, 4 status messages)
[IP_Flow 19-234] Refreshing IP repositories
[IP_Flow 19-1704] No user IP repositories specified
[IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2018.2/data/ip'.
Command: synth_design -rtl -name rtl_1 (3 more like this)
Elaborated Design (2 infos, 9 status messages)
General Messages (2 infos, 9 status messages)
[Project 1-570] Preparing netlist for logic optimization


Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (326)
Intra-Clock Paths
Inter-Clock Paths
Setup
Worst Negative Slack (WNS): 3.461 ns
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 106
Hold
Worst Hold Slack (WHS): 0.264 ns
Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 106
Pulse Width
Worst Pulse Width Slack (WPWS): 4.500 ns
Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 181
All user specified timing constraints are met.
Timing Summary - impl_1 (saved)
Timing Summary - timing_1


Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Utilization
Hierarchy
Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Flip Flop (<1
Slice Logic Distribution
Slice (1%)
SLICEM
SLICEL
LUT Flip Flop Pairs (<1%)
LUT-FF pairs with one ur
fully used LUT-FF pairs
LUT-FF pairs with one ur
LUT as Logic (<1%)
using O5 and O6
using O6 output only
Memory
Name
Slice LUTs (133800)
Slice Registers (267600)
Slice (33450)
LUT as Logic (133800)
LUT Flip Flop Pairs (133800)
DSPs (740)
Bonded IOB (500)
BUFGCTRL (32)
DSP48A1 288 179 123 288 25 1 327 1
genblk1_0.DSP_ASYNC (DSP48A1_SY... 288 179 123 288 25 1 0 0
INS_A1 (MUX_SYNCRST__paramete... 0 18 9 0 0 0 0 0
genblk1_0.DSP_ASYNC/INS_B1 (MUX_SYNCRST__parameterized0_0) 36 19 0 0 0 0 0
INS_C (MUX_SYNCRST__parameteri... 0 48 27 0 0 0 0 0
INS_D (MUX_SYNCRST__parameteri... 17 18 12 17 0 0 0 0
INS_P (MUX_SYNCRST__parameteri... 50 48 42 50 0 0 0 0
multiplier (MUX_SYNCRST__parame... 0 0 0 0 0 1 0 0
POST_ADD_SUB (POST_ADD_SUB_... 73 3 26 73 1 0 0 0
CYI (MUX_SYNCRST__parameteri... 1 1 1 1 1 0 0 0
CYO (MUX_SYNCRST__paramete... 1 2 2 1 0 0 0 0


X0Y4
X1Y4
X0Y3
X1Y3
X0Y2
X1Y2
X0Y1
X1Y1
X0Y0
X1Y0





Thx ;