RTL Code:

```
module D_mux_SYNC (D, clk, clk_en, rst, out);

parameter PARAM_REG = 1;
parameter WIDTH  = 18;
input [WIDTH-1:0] D;
input clk, clk_en, rst;
output reg [WIDTH-1:0] out;
reg [WIDTH-1:0] D_reg;

always @ (posedge clk) begin
    if (rst) begin
        D_reg <= 0;
    end
    else begin
        if (clk_en)
        D_reg <= D;
    end
end
always @(*) begin

    if (PARAM_REG) begin
        out = D_reg;
    end
    else begin
        out = D;
    end
end
endmodule
```

```
module D_mux_ASYNC (D, clk, clk_en, rst, out);

parameter PARAM_REG = 0;
parameter WIDTH  = 18;
input [WIDTH-1:0] D;
input clk, clk_en, rst;
output reg [WIDTH-1:0] out;
```

```
reg [WIDTH-1:0] D_reg;


always @ (posedge clk or posedge rst) begin
    if (rst) begin
        D_reg <= 0;
    end
    else begin
        if (clk_en)
        D_reg <= D;
    end
end
always @(*) begin

    if (PARAM_REG) begin
        out = D_reg;
    end
    else begin
        out = D;
    end
end
endmodule
```

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

parameter WIDTH_A = 18;
parameter WIDTH_C = 48;
parameter WIDTH_OP = 8;
parameter WIDTH_M = 36;
parameter WIDTH_CARRY =  1;
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 [WIDTH_A-1:0] A, B, D, BCIN;
input [WIDTH_C-1:0] C, PCIN;
input [WIDTH_OP-1:0] OPMODE;
input CLK, CARRYIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN,
RSTOPMODE, CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, COPMODE;
output [17:0] BCOUT;
output [47:0] P;
output [47:0] PCOUT;
output [35:0] M;
output CARRYOUT;
output CARRYOUTF;



wire [17:0] OUT_D, OUT_A0, OUT_A1, OUT_B0, OUT_B1;
wire [47:0] OUT_C;
wire [7:0] OUT_OPMODE;
wire [35:0] OUT_M;
wire OUT_CARRYIN;
reg [17:0] out_adder1, input_B1;
reg [35:0] multiplier_out;
reg [47:0] out_mux_x, out_mux_z;
reg [48:0] out_adder2;

generate
    if (RSTTYPE == "SYNC") begin
        D_mux_SYNC #(DREG, WIDTH_A) D (D, CLK, CED, RSTD, OUT_D);
        D_mux_SYNC #(A0REG, WIDTH_A) A0 (A, CLK, CEA, RSTA, OUT_A0);
        D_mux_SYNC #(CREG, WIDTH_C) C (C, CLK, CEC, RSTC, OUT_C);
        D_mux_SYNC #(OPMODEREG, WIDTH_OP) OP (OPMODE, CLK, COPMOD,
RSTOPMODE, OUT_OPMODE);
```

```
        if (B_INPUT == "DIRECT") begin
            D_mux_SYNC #(B0REG, WIDTH_A) B0 (B, CLK, CEB, RSTB, OUT_B0);
        end
        else begin
            D_mux_SYNC #(B0REG, WIDTH_A) B0 (BCIN, CLK, CEB, RSTB,
OUT_B0);
        end
    end
    else begin
        D_mux_ASYNC #(DREG, WIDTH_A) D (D, CLK, CED, RSTD, OUT_D);
        D_mux_ASYNC #(A0REG, WIDTH_A) A0 (A, CLK, CEA, RSTA, OUT_A0);
        D_mux_ASYNC #(CREG, WIDTH_C) C (C, CLK, CEC, RSTC, OUT_C);
        D_mux_ASYNC #(OPMODEREG, WIDTH_OP) OP (OPMODE, CLK, COPMOD,
RSTOPMODE, OUT_OPMODE);


        if (B_INPUT == "DIRECT") begin
            D_mux_ASYNC #(B0REG, WIDTH_A) B0 (B, CLK, CEB, RSTB, OUT_B0);
        end
        else begin
            D_mux_ASYNC #(B0REG, WIDTH_A) B0 (BCIN, CLK, CEB, RSTB,
OUT_B0);
        end
    end
endgenerate

always @(*) begin
    if (OUT_OPMODE[6]) begin
        out_adder1 = OUT_D - OUT_B0;
    end
    else begin
        out_adder1 = OUT_D + OUT_B0;
    end

    if (OUT_OPMODE[4]) begin
        input_B1 = out_adder1;
    end
    else begin
        input_B1 = OUT_B0;
```

```
    end
end

generate

    if (RSTTYPE == "SYNC") begin
        D_mux_SYNC #(B1REG, WIDTH_A) B1 (input_B1, CLK, CEB, RSTB,
OUT_B1);
    end
    else begin
        D_mux_ASYNC #(B1REG, WIDTH_A) B1 (input_B1, CLK, CEB, RSTB,
OUT_B1);
    end


    if (RSTTYPE == "SYNC") begin
        D_mux_SYNC #(A1REG, WIDTH_A) A1 (OUT_A0, CLK, CEA, RSTA, OUT_A1);
    end
    else begin
        D_mux_ASYNC #(A1REG, WIDTH_A) A1 (OUT_A0, CLK, CEA, RSTA, OUT_A1);
    end
endgenerate

assign BCOUT = OUT_B1;

always @(*) begin
    multiplier_out = OUT_B1 * OUT_A1;
end

generate
    if (RSTTYPE == "SYNC") begin
        D_mux_SYNC #(MREG, WIDTH_M) M (multiplier_out, CLK, CEM, RSTM,
OUT_M);
    end
    else begin
        D_mux_ASYNC #(MREG, WIDTH_M) M (multiplier_out, CLK, CEM, RSTM,
OUT_M);
    end
endgenerate
```

```
assign M = OUT_M;

generate
    if ( CARRYINSEL == "OPMODE5") begin
        if (RSTTYPE == "SYNC") begin
             D_mux_SYNC #(CARRYINREG, WIDTH_CARRY) CARRYIN (OPMODE[5],
CLK, CECARRYIN, RSTCARRYIN, OUT_CARRYIN);
        end
        else begin
            D_mux_ASYNC #(CARRYINREG, WIDTH_CARRY) CARRYIN (OPMODE[5],
CLK, CECARRYIN, RSTCARRYIN, OUT_CARRYIN);
        end
    end
    else begin
         if (RSTTYPE == "SYNC") begin
             D_mux_SYNC #(CARRYINREG, WIDTH_CARRY) CARRYIN (CARRYIN, CLK,
CECARRYIN, RSTCARRYIN, OUT_CARRYIN);
        end
        else begin
            D_mux_ASYNC #(CARRYINREG, WIDTH_CARRY) CARRYIN (CARRYIN, CLK,
CECARRYIN, RSTCARRYIN, OUT_CARRYIN);
        end
    end
endgenerate

always @(*) begin
    if (OUT_OPMODE[1:0] == 0) begin
        out_mux_x = 0;
    end
    else if (OUT_OPMODE[1:0] == 1) begin
        out_mux_x = OUT_M;
    end
    else if (OUT_OPMODE[1:0] == 2) begin
        out_mux_x = P;
    end
    else begin
         out_mux_x = {OUT_D[11:0], OUT_A0, OUT_B0};
    end

    if (OUT_OPMODE[3:2] == 0) begin
```

```
        out_mux_z = 0;
    end
    else if (OUT_OPMODE[3:2] == 1) begin
        out_mux_z = PCIN;
    end
    else if (OUT_OPMODE[3:2] == 2) begin
        out_mux_z = P;
    end
    else begin
        out_mux_z = OUT_C;
    end
end

always @(*) begin
    if (OUT_OPMODE[7]) begin
        out_adder2 = out_mux_z - (out_mux_x + OUT_CARRYIN);
    end
    else begin
        out_adder2 = out_mux_z + out_mux_x;
    end
end

generate
    if (RSTTYPE == "SYNC") begin
            D_mux_SYNC #(CARRYOUTREG, WIDTH_CARRY) CARRYOUT
(out_adder2[48], CLK, CECARRYIN, RSTCARRYIN, CARRYOUT);
        end
        else begin
            D_mux_ASYNC #(CARRYOUTREG, WIDTH_CARRY) CARRYOUT
(out_adder2[48], CLK, CECARRYIN, RSTCARRYIN, CARRYOUT);
        end

    if (RSTTYPE == "SYNC") begin
         D_mux_SYNC #(PREG, WIDTH_C) P (out_adder2[47:0], CLK, CEP, RSTP,
P);
    end
    else begin
        D_mux_ASYNC #(PREG, WIDTH_C) P (out_adder2[47:0], CLK, CEP, RSTP,
P);
    end
```

```
endgenerate

assign CARRYOUTF = CARRYOUT;
assign PCOUT = P;
endmodule
```

Testbench:

```
module test_DSP48A1 ();
parameter WIDTH_A = 18;
parameter WIDTH_C = 48;
parameter WIDTH_OP = 8;
parameter WIDTH_M = 36;
parameter WIDTH_CARRY =  1;
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";
reg [WIDTH_A - 1:0] A, B, D, BCIN;
reg [WIDTH_C - 1:0] C, PCIN;
reg [WIDTH_OP - 1:0] OPMODE;
reg CLK, CARRYIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN,
RSTOPMODE, CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, COPMODE;
wire [17:0] BCOUT;
wire [47:0] PCOUT, P;
wire [35:0] M;
wire CARRYOUT, CARRYOUTF;
```

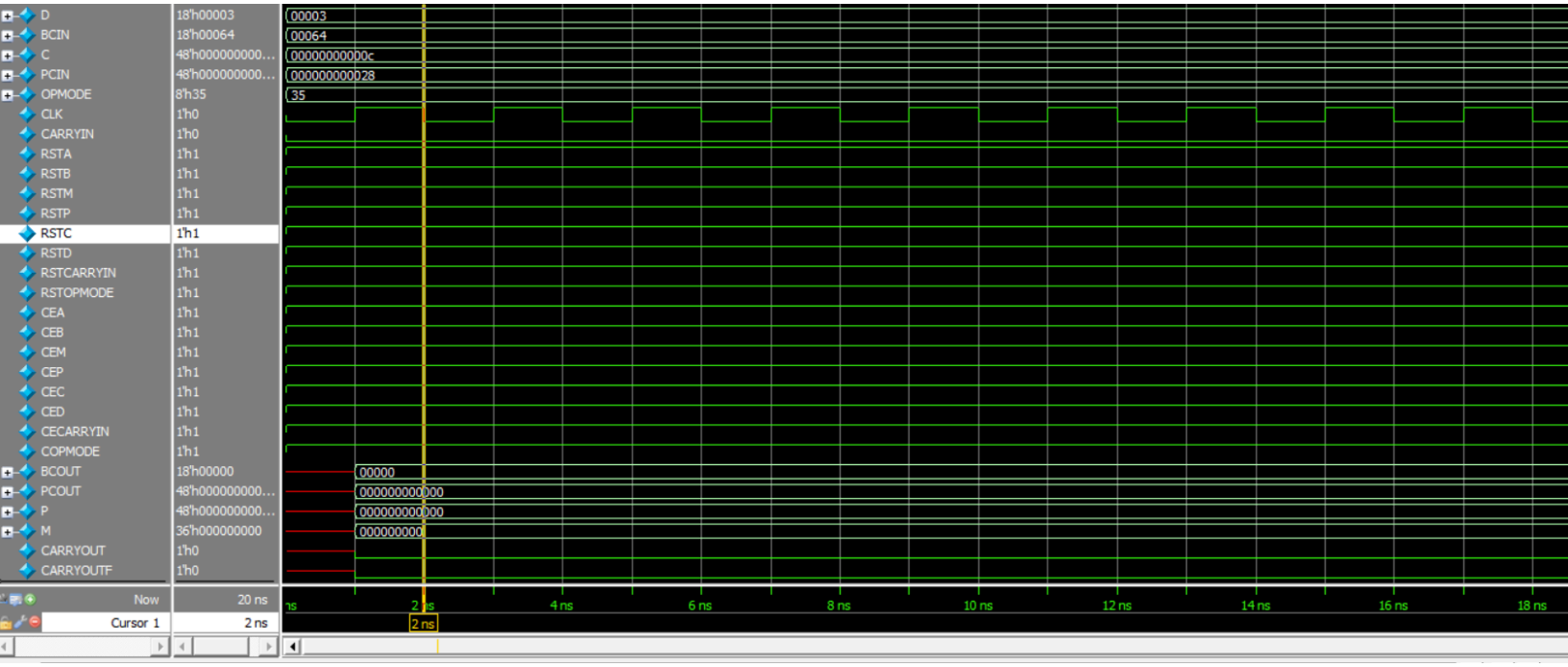
```
DSP48A1 #(WIDTH_A, WIDTH_C, WIDTH_OP, WIDTH_M, WIDTH_CARRY, A0REG, B0REG,
A1REG, B1REG, CREG, DREG, MREG, PREG, CARRYINREG, CARRYOUTREG, OPMODEREG,
CARRYINSEL, B_INPUT, RSTTYPE) DUT1 (A, B, C, D, CLK, CARRYIN, OPMODE,
BCIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPMODE, CEA, CEB,
CEM, CEP, CEC, CED, CECARRYIN, COPMODE, PCIN, BCOUT, PCOUT, P, M,
CARRYOUT, CARRYOUTF);

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

initial begin
    RSTA=1;
    RSTB=1;
    RSTM=1;
    RSTP=1;
    RSTC=1;
    RSTD=1;
    RSTCARRYIN=1;
    RSTOPMODE=1;
    A=10;
    B=15;
    C=12;
    D=3;
    CARRYIN=0;
    OPMODE=8'b00110101;
    BCIN=100;
    CEA=1;
    CEB=1;
    CEM=1;
    CEP=1;
    CEC=1;
    CED=1;
    CECARRYIN=1;
    COPMODE=0;
    PCIN=40;
```

```
    repeat(10) begin
        @(negedge CLK);
    end

    RSTA=0;
    RSTB=0;
    RSTM=0;
    RSTP=0;
    RSTC=0;
    RSTD=0;
    RSTCARRYIN=0;
    RSTOPMODE=0;
    COPMODE=1;



    repeat(10) begin
        @(negedge CLK);
    end

    OPMODE=8'b00111110;

    repeat(10) begin
        @(negedge CLK);
    end
    $stop;
end

endmodule
```

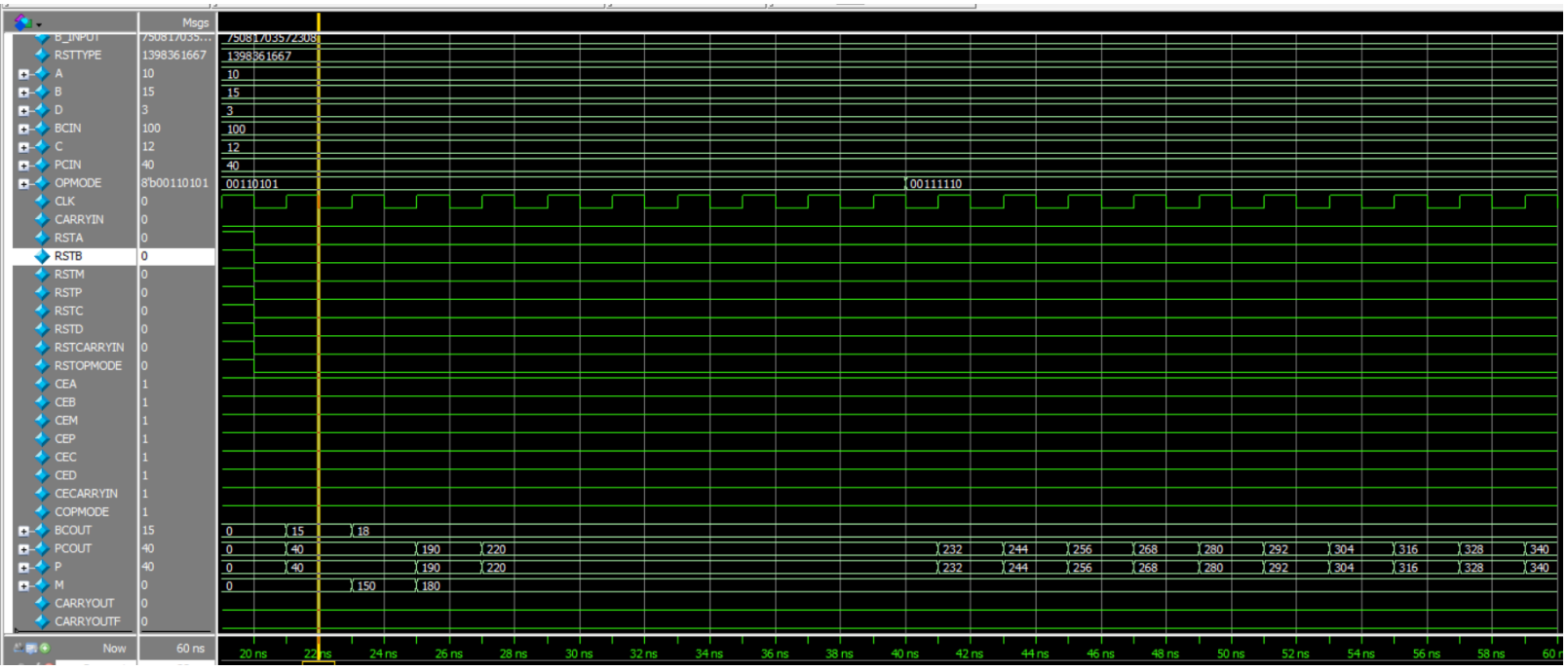
Do file:

```
vlib work
vlog D_mux_SYNC.v D_mux_ASYNC.v DSP48A1.v
vsim -voptargs=+acc work.test_DSP48A1
add wave *
run -all
#quit -sim
```

Reset:



2 Cases:



Constraint File:

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to
the top level signal names in the project

## 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]


## Switches
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports
{sw[0]}]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports
{sw[1]}]
#set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports
{sw[2]}]
#set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports
{sw[3]}]
#set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports
{sw[4]}]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports
{sw[5]}]
#set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports
{sw[6]}]
#set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports
{sw[7]}]
#set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports
{sw[8]}]
#set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports
{sw[9]}]
#set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports
{sw[10]}]
#set_property -dict { PACKAGE_PIN R3    IOSTANDARD LVCMOS33 } [get_ports
{sw[11]}]
```

```
#set_property -dict { PACKAGE_PIN W2    IOSTANDARD LVCMOS33 } [get_ports
{sw[12]}]
#set_property -dict { PACKAGE_PIN U1    IOSTANDARD LVCMOS33 } [get_ports
{sw[13]}]
#set_property -dict { PACKAGE_PIN T1    IOSTANDARD LVCMOS33 } [get_ports
{sw[14]}]
#set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports
{sw[15]}]


## LEDs
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports
{led[0]}]
#set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports
{led[1]}]
#set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports
{led[2]}]
#set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports
{led[3]}]
#set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports
{led[4]}]
#set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 } [get_ports
{led[5]}]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports
{led[6]}]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports
{led[7]}]
#set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports
{led[8]}]
#set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports
{led[9]}]
#set_property -dict { PACKAGE_PIN W3    IOSTANDARD LVCMOS33 } [get_ports
{led[10]}]
#set_property -dict { PACKAGE_PIN U3    IOSTANDARD LVCMOS33 } [get_ports
{led[11]}]
#set_property -dict { PACKAGE_PIN P3    IOSTANDARD LVCMOS33 } [get_ports
{led[12]}]
#set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports
{led[13]}]
```

```
#set_property -dict { PACKAGE_PIN P1    IOSTANDARD LVCMOS33 } [get_ports
{led[14]}]
#set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports
{led[15]}]


##7 Segment Display
#set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports
{seg[0]}]
#set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports
{seg[1]}]
#set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports
{seg[2]}]
#set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports
{seg[3]}]
#set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports
{seg[4]}]
#set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports
{seg[5]}]
#set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports
{seg[6]}]

#set_property -dict { PACKAGE_PIN V7   IOSTANDARD LVCMOS33 } [get_ports
dp]

#set_property -dict { PACKAGE_PIN U2   IOSTANDARD LVCMOS33 } [get_ports
{an[0]}]
#set_property -dict { PACKAGE_PIN U4   IOSTANDARD LVCMOS33 } [get_ports
{an[1]}]
#set_property -dict { PACKAGE_PIN V4   IOSTANDARD LVCMOS33 } [get_ports
{an[2]}]
#set_property -dict { PACKAGE_PIN W4   IOSTANDARD LVCMOS33 } [get_ports
{an[3]}]


##Buttons
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports
rst]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports
btnU]
```

```
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports
btnL]
#set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports
btnR]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports
btnD]


##Pmod Header JA
#set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports
{JA[0]}];#Sch name = JA1
#set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports
{JA[1]}];#Sch name = JA2
#set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports
{JA[2]}];#Sch name = JA3
#set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports
{JA[3]}];#Sch name = JA4
#set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports
{JA[4]}];#Sch name = JA7
#set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports
{JA[5]}];#Sch name = JA8
#set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports
{JA[6]}];#Sch name = JA9
#set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports
{JA[7]}];#Sch name = JA10

##Pmod Header JB
#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports
{JB[0]}];#Sch name = JB1
#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports
{JB[1]}];#Sch name = JB2
#set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVCMOS33 } [get_ports
{JB[2]}];#Sch name = JB3
#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports
{JB[3]}];#Sch name = JB4
#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports
{JB[4]}];#Sch name = JB7
#set_property -dict { PACKAGE_PIN A17   IOSTANDARD LVCMOS33 } [get_ports
{JB[5]}];#Sch name = JB8
```

```
#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports
{JB[6]}];#Sch name = JB9
#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports
{JB[7]}];#Sch name = JB10

##Pmod Header JC
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports
{JC[0]}];#Sch name = JC1
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports
{JC[1]}];#Sch name = JC2
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports
{JC[2]}];#Sch name = JC3
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports
{JC[3]}];#Sch name = JC4
#set_property -dict { PACKAGE_PIN L17   IOSTANDARD LVCMOS33 } [get_ports
{JC[4]}];#Sch name = JC7
#set_property -dict { PACKAGE_PIN M19   IOSTANDARD LVCMOS33 } [get_ports
{JC[5]}];#Sch name = JC8
#set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports
{JC[6]}];#Sch name = JC9
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports
{JC[7]}];#Sch name = JC10

##Pmod Header JXADC
#set_property -dict { PACKAGE_PIN J3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[0]}];#Sch name = XA1_P
#set_property -dict { PACKAGE_PIN L3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[1]}];#Sch name = XA2_P
#set_property -dict { PACKAGE_PIN M2   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[2]}];#Sch name = XA3_P
#set_property -dict { PACKAGE_PIN N2   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[3]}];#Sch name = XA4_P
#set_property -dict { PACKAGE_PIN K3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[4]}];#Sch name = XA1_N
#set_property -dict { PACKAGE_PIN M3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[5]}];#Sch name = XA2_N
#set_property -dict { PACKAGE_PIN M1   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[6]}];#Sch name = XA3_N
#set_property -dict { PACKAGE_PIN N1   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[7]}];#Sch name = XA4_N
```

```
##VGA Connector
#set_property -dict { PACKAGE_PIN G19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[0]}]
#set_property -dict { PACKAGE_PIN H19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[1]}]
#set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[2]}]
#set_property -dict { PACKAGE_PIN N19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[3]}]
#set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[0]}]
#set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[1]}]
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[2]}]
#set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[3]}]
#set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[0]}]
#set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[1]}]
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[2]}]
#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[3]}]
#set_property -dict { PACKAGE_PIN P19   IOSTANDARD LVCMOS33 } [get_ports
Hsync]
#set_property -dict { PACKAGE_PIN R19   IOSTANDARD LVCMOS33 } [get_ports
Vsync]


##USB-RS232 Interface
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports
RsRx]
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports
RsTx]
```

```
##USB HID (PS/2)
#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33   PULLUP true
} [get_ports PS2Clk]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33   PULLUP true
} [get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it
using the
##STARTUPE2 primitive.
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[0]}]
#set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[1]}]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[2]}]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[3]}]
#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports
QspiCSn]


## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]

## SPI configuration mode options for QSPI boot, can be used for all
designs
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]

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 {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]}]]
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 48 [get_debug_ports u_ila_0/probe1]
connect_debug_port u_ila_0/probe1 [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]}]]
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 48 [get_debug_ports u_ila_0/probe2]
connect_debug_port u_ila_0/probe2 [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]}]]
create_debug_port u_ila_0 probe
```

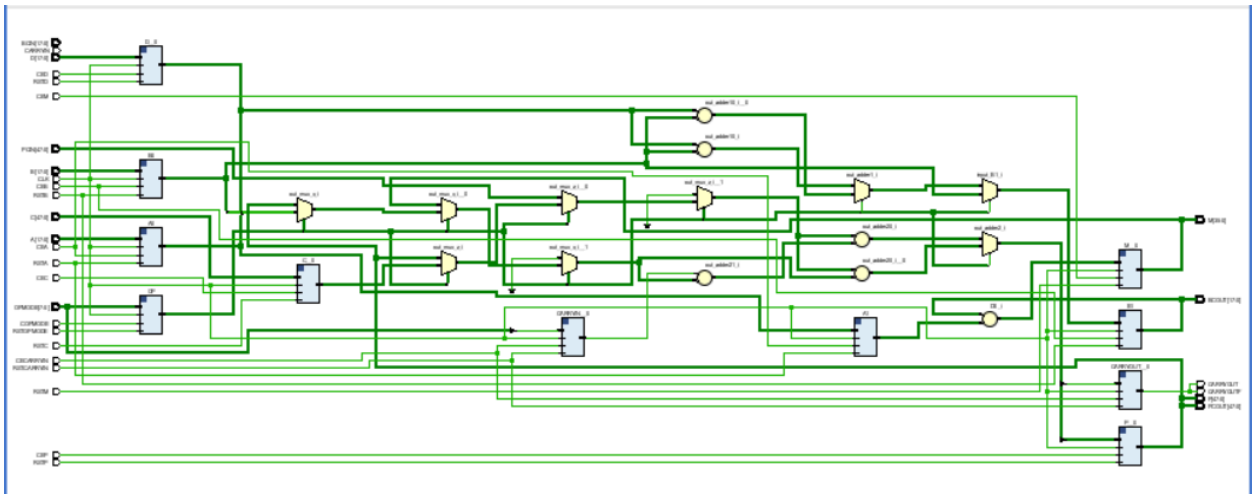
```
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe3]
set_property port_width 18 [get_debug_ports u_ila_0/probe3]
connect_debug_port u_ila_0/probe3 [get_nets [list {D_IBUF[0]} {D_IBUF[1]}
{D_IBUF[2]} {D_IBUF[3]} {D_IBUF[4]} {D_IBUF[5]} {D_IBUF[6]} {D_IBUF[7]}
{D_IBUF[8]} {D_IBUF[9]} {D_IBUF[10]} {D_IBUF[11]} {D_IBUF[12]}
{D_IBUF[13]} {D_IBUF[14]} {D_IBUF[15]} {D_IBUF[16]} {D_IBUF[17]}]]
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 {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]}]]
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 18 [get_debug_ports u_ila_0/probe5]
connect_debug_port u_ila_0/probe5 [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]}]]
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 36 [get_debug_ports u_ila_0/probe6]
connect_debug_port u_ila_0/probe6 [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]}]]
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]} {OPMODE_IBUF[6]} {OPMODE_IBUF[7]}]]
create_debug_port u_ila_0 probe
```

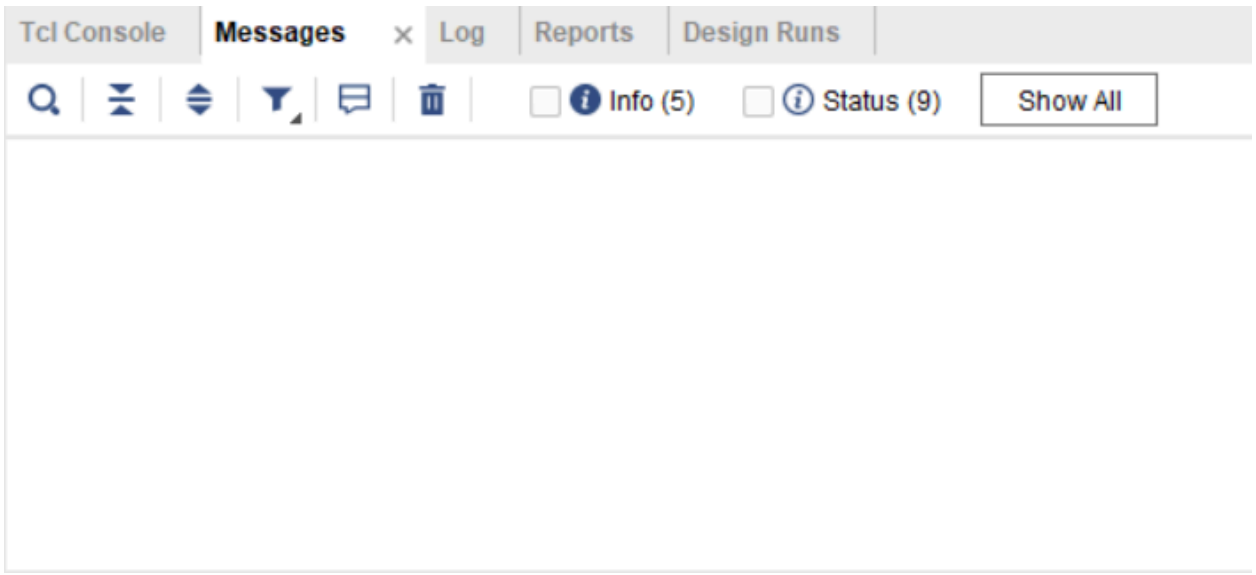
```
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe8]
set_property port_width 48 [get_debug_ports u_ila_0/probe8]
connect_debug_port u_ila_0/probe8 [get_nets [list {P_OBUF[0]} {P_OBUF[1]}
{P_OBUF[2]} {P_OBUF[3]} {P_OBUF[4]} {P_OBUF[5]} {P_OBUF[6]} {P_OBUF[7]}
{P_OBUF[8]} {P_OBUF[9]} {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]}]]
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 CEP_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 CLK_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 COPMODE_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]]
```

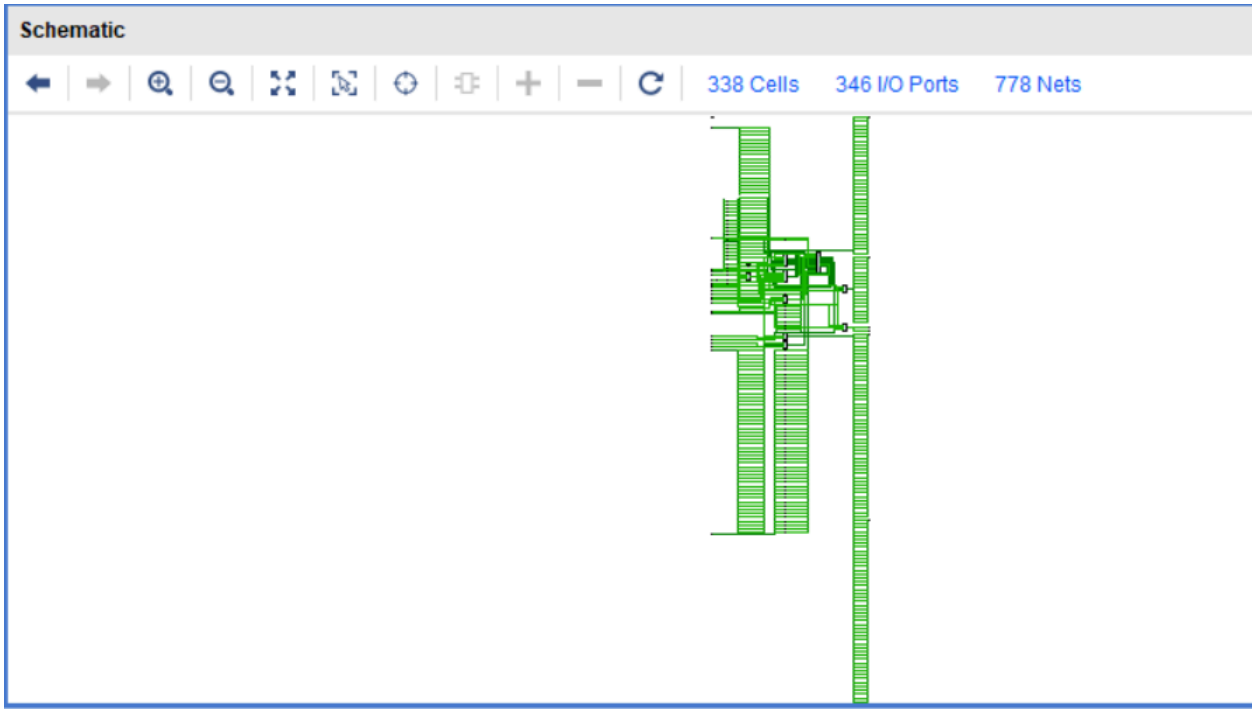
```
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe26]
set_property port_width 1 [get_debug_ports u_ila_0/probe26]
connect_debug_port u_ila_0/probe26 [get_nets [list RSTP_IBUF]]
set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
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]
```

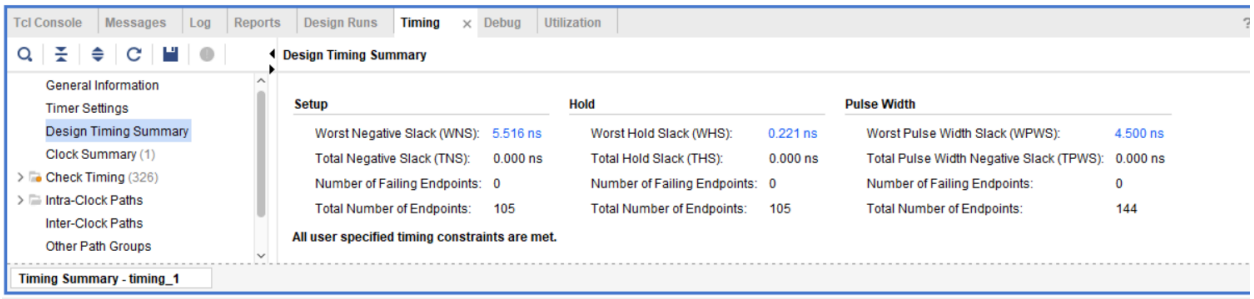
Elaboration:

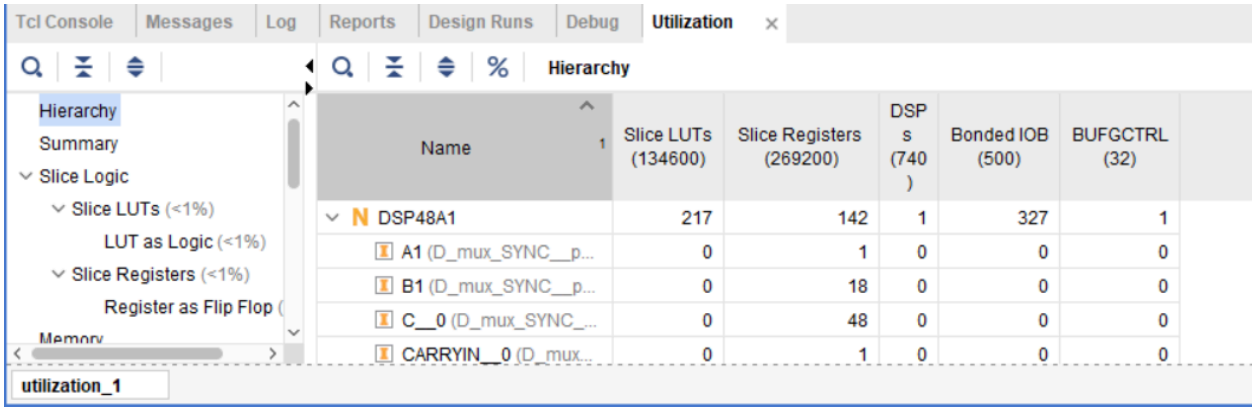


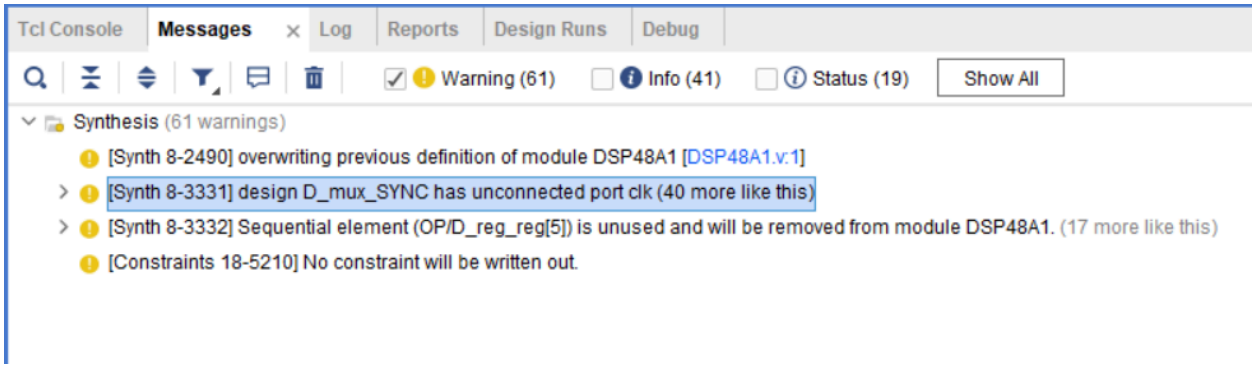


Synthesis:

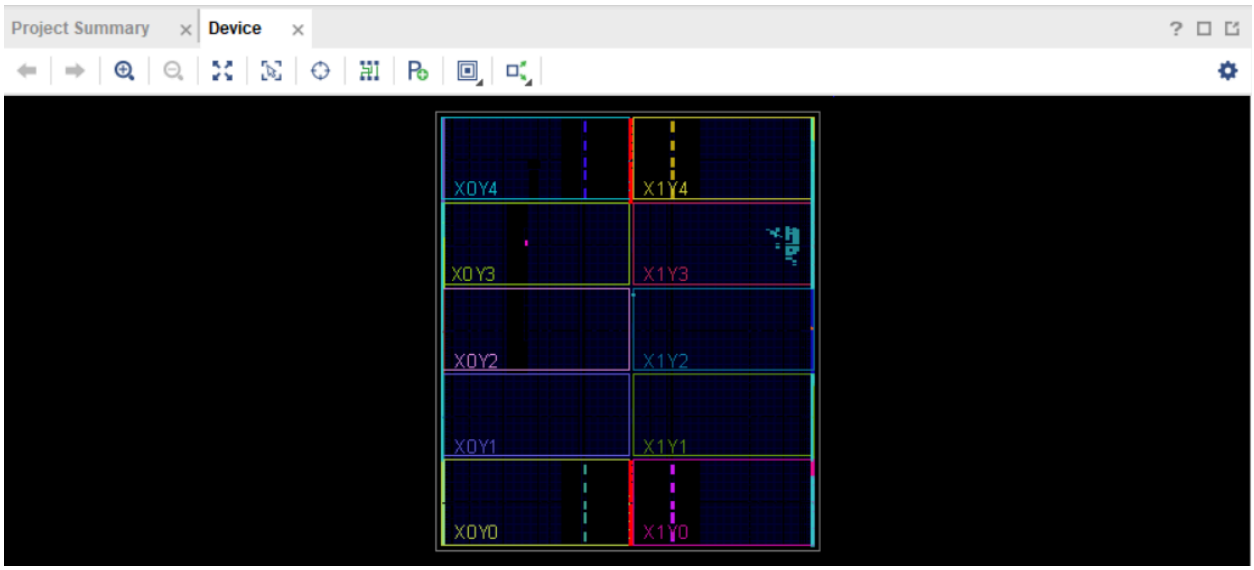

Schematic
338 Cells
346 I/O Ports
778 Nets

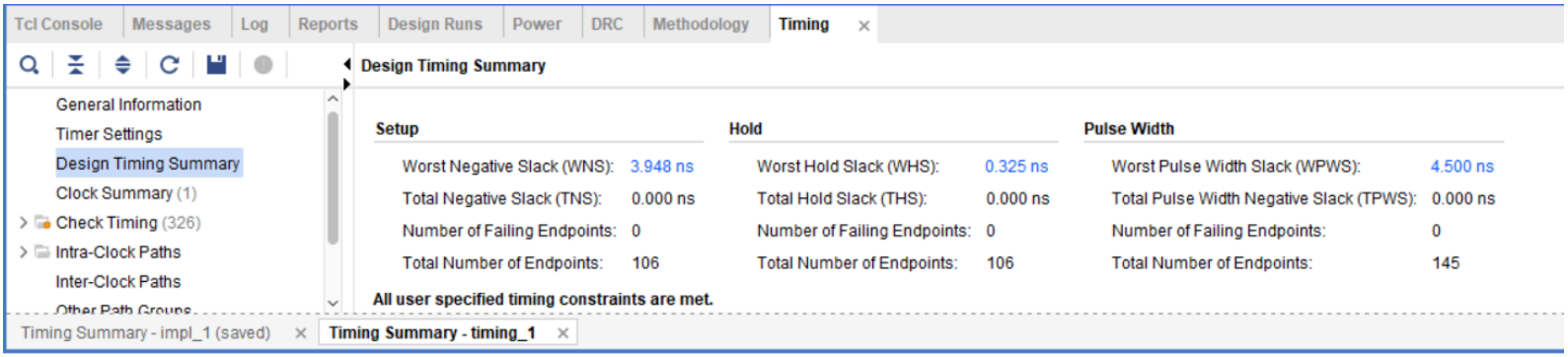

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Debug
Utilization
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (326)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
Setup
Worst Negative Slack (WNS): 5.516 ns
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 105
Hold
Worst Hold Slack (WHS): 0.221 ns
Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 105
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: 144
All user specified timing constraints are met.
Timing Summary - timing_1

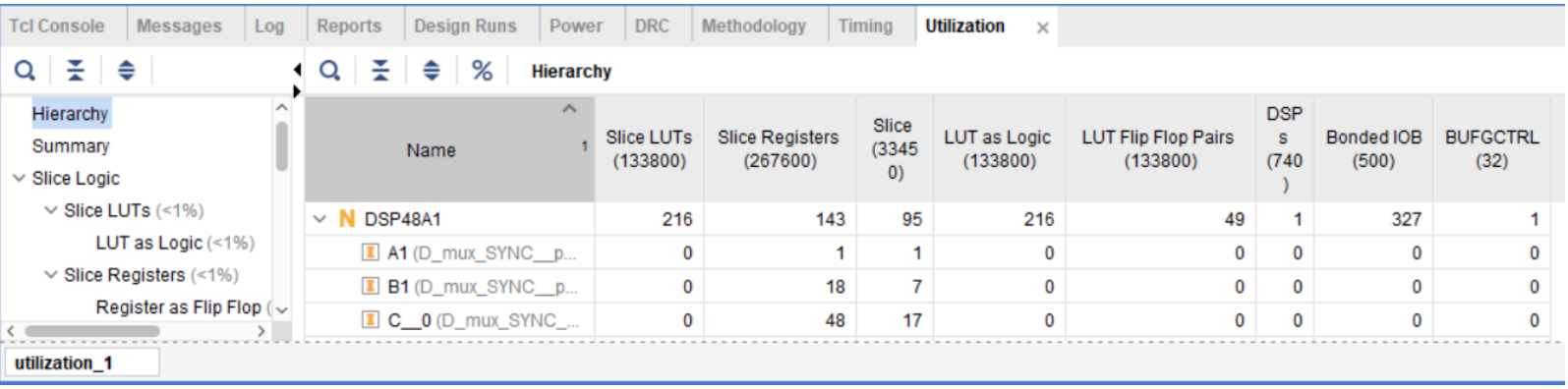

Tcl Console
Messages
Log
Reports
Design Runs
Debug
Utilization
Hierarchy
Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Flip Flop
Memory
Name
Slice LUTs (134600)
Slice Registers (269200)
DSPs (740)
Bonded IOB (500)
BUFGCTRL (32)
DSP48A1 217 142 1 327 1
A1 (D_mux_SYNC__p... 0 1 0 0 0
B1 (D_mux_SYNC__p... 0 18 0 0 0
C__0 (D_mux_SYNC_... 0 48 0 0 0
CARRYIN__0 (D_mux... 0 1 0 0 0
utilization_1

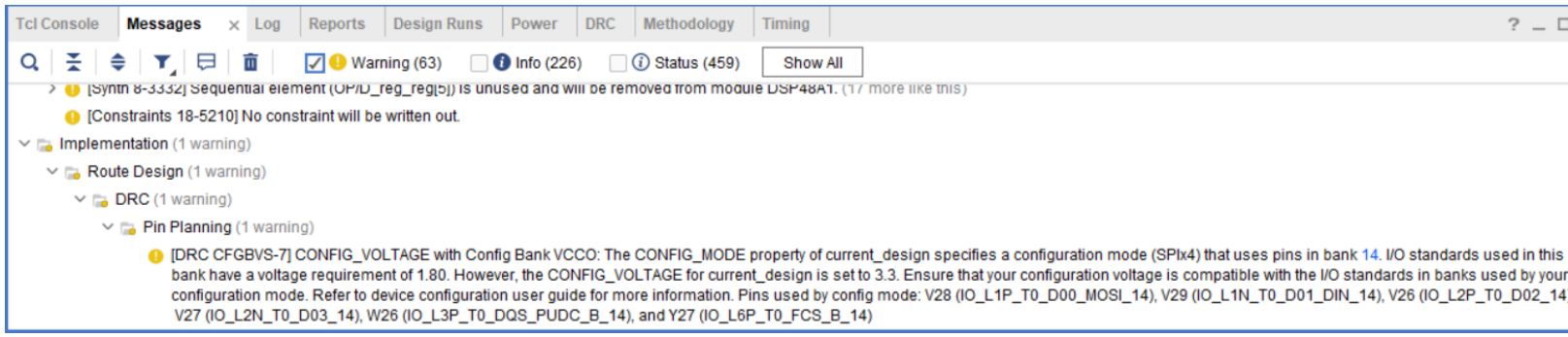


Implementation:






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
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 216 143 95 216 49 1 327 1
A1 (D_mux_SYNC__p... 0 1 1 0 0 0 0 0
B1 (D_mux_SYNC__p... 0 18 7 0 0 0 0 0
C__0 (D_mux_SYNC_... 0 48 17 0 0 0 0 0
utilization_1


Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Warning (63)
Info (226)
Status (459)
Show All
[Synth 8-3332] Sequential element (OP/D_reg_reg[5]) is unused and will be removed from module DSP48A1. (17 more like this)
[Constraints 18-5210] No constraint will be written out.
Implementation (1 warning)
Route Design (1 warning)
DRC (1 warning)
Pin Planning (1 warning)
[DRC CFGBVS-7] CONFIG_VOLTAGE with Config Bank VCCO: The CONFIG_MODE property of current_design specifies a configuration mode (SPIx4) that uses pins in bank 14. I/O standards used in this bank have a voltage requirement of 1.80. However, the CONFIG_VOLTAGE for current_design is set to 3.3. Ensure that your configuration voltage is compatible with the I/O standards in banks used by your configuration mode. Refer to device configuration user guide for more information. Pins used by config mode: V28 (IO_L1P_T0_D00_MOSI_14), V29 (IO_L1N_T0_D01_DIN_14), V26 (IO_L2P_T0_D02_14), V27 (IO_L2N_T0_D03_14), W26 (IO_L3P_T0_DQS_PUDC_B_14), and Y27 (IO_L6P_T0_FCS_B_14)