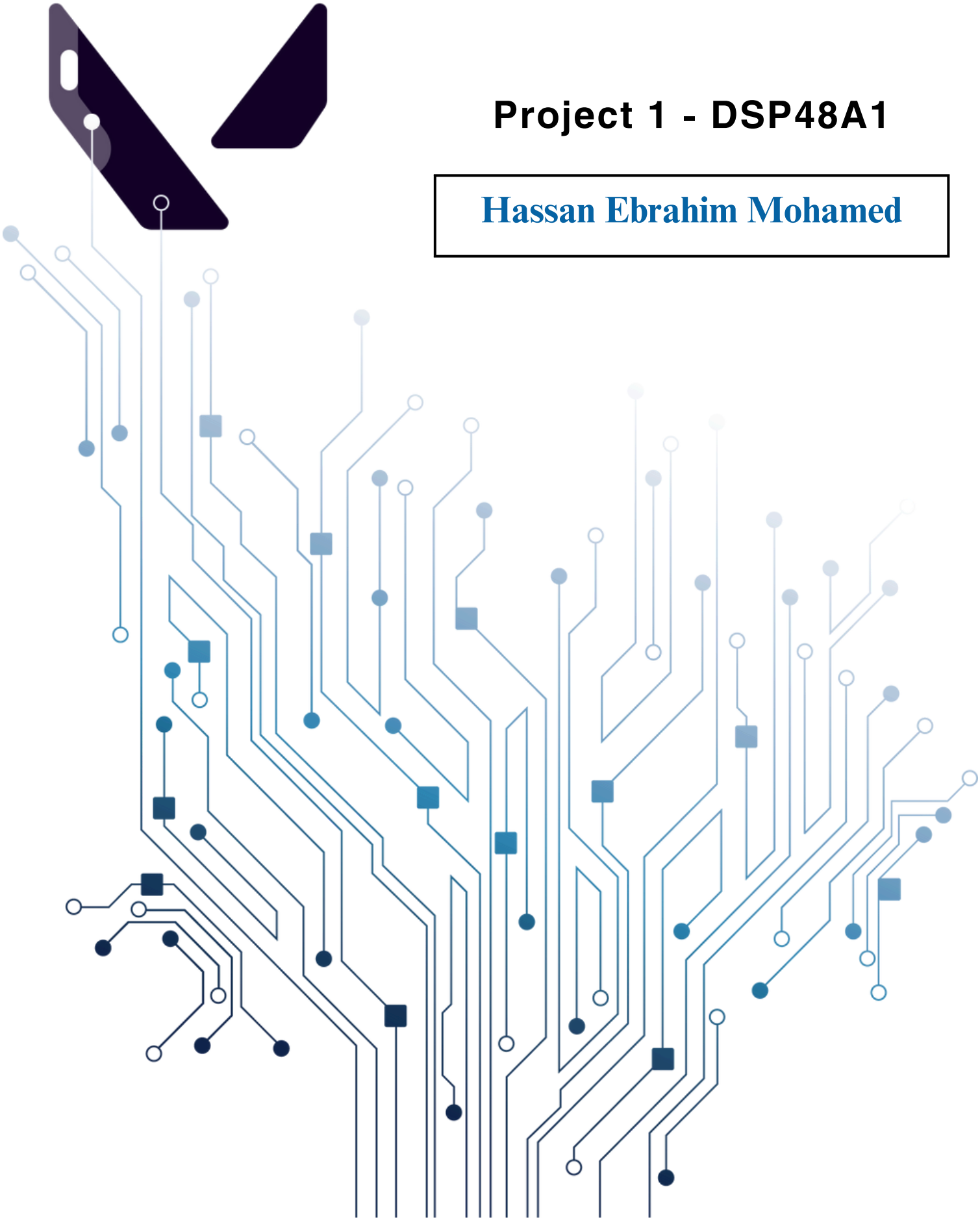


Project 1 - DSP48A1

Hassan Ebrahim Mohamed

Table of Contents

RTL Code

Instantiated Modules in DSP48A1

```
module mux (A , clk , rst , CEA , out , SEL );
parameter WIDTH = 18 ;
parameter RSTTYPE = "SYNC" ;
input SEL ;
input [WIDTH - 1 : 0] A ;
input clk , rst , CEA ;
output [WIDTH - 1 : 0] out ;

reg [WIDTH - 1 : 0] A_reg ;

generate
    if (RSTTYPE == "ASYNC") begin
        always @(posedge clk or posedge rst) begin
            if (rst) begin
                A_reg <= 0 ;
            end
            else if (CEA) begin
                A_reg <= A ;
            end
        end

        assign out = (SEL == 0 )? A : A_reg ;

    end
    else if (RSTTYPE == "SYNC") begin
        always @(posedge clk ) begin
            if (rst) begin
                A_reg <= 0 ;
            end
            else if (CEA) begin
                A_reg <= A ;
            end
        end

        assign out = (SEL == 0 )? A : A_reg ;

    end
endgenerate
endmodule
```

```
module multiply (A , B , out);
parameter WIDTH_1 = 18 ;
parameter WIDTH_2 = 18 ;
input [WIDTH_1 - 1 : 0] A ;
input [WIDTH_2 - 1 : 0] B ;
output [WIDTH_1 + WIDTH_2 - 1 : 0] out ;

assign out = A * B ;

endmodule
```

```
module ADD_SUBTRACT (D , B , operation , out , cout , cin);
parameter SEL = "Pre" ;
parameter WIDTH_1 = 18 ;
parameter WIDTH_2 = 18 ;
input [WIDTH_1 - 1 : 0] D ;
input [WIDTH_2 - 1 : 0] B ;
input cin , operation ;
output reg cout ;
output reg [WIDTH_1- 1 : 0] out ;

always @(*) begin
    if(SEL == "Pre") begin
        if(operation)begin
            out = D - B ;
        end
        else begin
            out = D + B ;
        end
    end
    else if (SEL == "Post") begin
        if(operation)begin
            out = D - (B + cin) ;
            cout = 0 ;
        end
        else begin
            {cout , out} = D + B + cin ;
        end
    end
end


endmodule
```

Design Code

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


parameter A0REG = 0 ; // If equal (0) --> No register, else --> Registered
parameter A1REG = 1 ; // If equal (0) --> No register, else --> Registered
parameter B0REG = 0 ; // If equal (0) --> No register, else --> Registered
parameter B1REG = 1 ; // If equal (0) --> No register, else --> Registered
parameter CREG = 1  ; // If equal (0) --> No register, else --> Registered
parameter DREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter MREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter PREG = 1  ; // If equal (0) --> No register, else --> Registered
parameter CARRYINREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter CARRYOUTREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter OPMODEREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter CARRYINSEL = "OPMODE5"; // Select between (CARRYIN or OPMODE5)
parameter B_INPUT = "DIRECT" ; // Select between (DIRECT or CASCADE)
parameter RSTTYPE = "SYNC" ; // Select between (SYNC or A SYNC)


input [17:0] A , B , D ;
input [17:0] BCIN ;
input [47:0] C ;
input CARRYIN , clk ;
input [47:0] PCIN ;  // Cascade input for Port P
input [7:0] OPMODE ;

input CEA ; // Clock enable for the A port registers: (A0REG & A1REG)
input CEB ; // Clock enable for the B port registers: (B0REG & B1REG)
input CEC ; // Clock enable for the C port registers (CREG)
input CECARRYIN ; // Clock enable for the carry-in register (CYI) and the carry-out register (CYO)
input CED ; // Clock enable for the D port register (DREG)
input CEM ; // Clock enable for the multiplier register (MREG)
input CEOPMODE ; // Clock enable for the opmode register (OPMODEREG)
input CEP ; // Clock enable for the P output port registers (PREG = 1)

input RSTA ; // Reset for the A registers: (A0REG & A1REG)
input RSTB ; // Reset for the B registers: (B0REG & B1REG)
input RSTC ; // Reset for the C registers (CREG)
input RSTCARRYIN ; // Reset for the carry-in register (CYI) and the carry-out register (CYO)
input RSTD ; // Reset for the D register (DREG)
input RSTM ; // Reset for the multiplier register (MREG)
input RSTOPMODE ; // Reset for the opmode register (OPMODEREG)
input RSTP ; // Reset for the P output registers (PREG = 1)

output [35:0] M ;
output [17:0] BCOUT ; //Cascade output for Port B
output [47:0] P ;
output [47:0] PCOUT ; // Cascade output for Port P
output CARRYOUT , CARRYOUTF ;

wire [7:0] OPMODE_reg ;
// module mux (A , clk , rst , CEA , out , SEL );
mux #(8 , RSTTYPE) OPMODE_part (OPMODE , clk , RSTOPMODE , CEOPMODE ,  OPMODE_reg , OPMODEREG );



wire [17:0] A0_reg ;
wire [17:0] D_reg ;
wire [47:0] C_reg ;
reg [17:0] B0_before_reg ;
wire [17:0] B0_reg ;

// module mux (A , clk , rst , CEA , out , SEL );
mux #(18 , RSTTYPE) A_part_1 (A , clk , RSTA , CEA ,  A0_reg , A0REG );
mux #(18 , RSTTYPE) D_part (D , clk , RSTD , CED ,  D_reg , DREG );
mux #(48 , RSTTYPE) C_part (C , clk , RSTC , CEC ,  C_reg , CREG );
mux #(18 , RSTTYPE) B_part_1 (B0_before_reg , clk , RSTB , CEB ,  B0_reg , B0REG );

always @(B , BCIN) begin
    if (B_INPUT == "DIRECT") begin
        B0_before_reg = B ;
    end
    else if (B_INPUT == "CASCADE") begin
        B0_before_reg = BCIN ;
    end
    else begin
        B0_before_reg = 0 ;
    end
end

wire [17:0] Pre_out ;
// module ADD_SUBTRACT (D , B , operation , out , cout , cin);
assign Pre_out = (OPMODE_reg[6] == 1)? (D_reg-B0_reg) : (D_reg+B0_reg) ;

wire [17:0] B1_before_reg ;
assign B1_before_reg = (OPMODE_reg[4] == 0 )? B : Pre_out ;
```

```
wire [17:0] A1_reg ;
wire [17:0] B1_reg ;
// module mux (A , clk , rst , CEA , out , SEL );
mux #(.WIDTH(18) , .RSTTYPE(RSTTYPE)) A_part_2 (.A(A0_reg) , .clk(clk) , .rst(RSTA) , .CEA(CEA) , .out(A1_reg) , .SEL(A1REG) );
mux #(18 , RSTTYPE) B_part_2 (B1_before_reg , clk , RSTB , CEB ,  B1_reg , B1REG );

assign BCOUT = B1_reg ;

wire [35:0] M_before ;
wire [35:0] M_reg ;

// module multiply (A , B , out);
multiply #(.WIDTH_1(18) , .WIDTH_2(18)) multiplier (B1_reg , A1_reg , M_before);
mux #(36 , RSTTYPE) M_part (M_before , clk , RSTM , CEM ,  M_reg , MREG );

assign M = M_reg ;
assign PCOUT = P ;

wire [47 : 0] in0_of_X = {D[11:0] , A[17:0] , B[17:0]};

wire [47 : 0] mux_X_out ;
wire [47 : 0] mux_Z_out ;

assign mux_X_out = (OPMODE_reg[1:0] == 1)? M_reg : (OPMODE_reg[1:0] == 2)? P : (OPMODE_reg[1:0] == 3)? in0_of_X : 0 ;
assign mux_Z_out = (OPMODE_reg[3:2] == 1)? PCIN : (OPMODE_reg[3:2] == 2)? P : (OPMODE_reg[3:2] == 3)? C_reg : 0 ;

wire CYI , CIN ;

assign CYI = (CARRYINSEL == "OPMODE5")? OPMODE_reg[5] :(CARRYINSEL == "CARRYIN")? CARRYIN : 0  ;

// module mux (A , clk , rst , CEA , out , SEL );
mux #(1 , RSTTYPE) CYI_part (CYI , clk , RSTCARRYIN , CECARRYIN ,  CIN , CARRYINREG);


wire [47:0] Post_out ;
wire Post_cout ;

// module ADD_SUBTRACT (D , B , operation , out , cout , cin);
ADD_SUBTRACT #(.SEL("Post") , .WIDTH_1(48) , .WIDTH_2(48)) Post_Adder_or_Sutracter (mux_X_out , mux_Z_out , OPMODE_reg[7] , Post_out , Post_cout , CIN);


// module mux (A , clk , rst , CEA , out , SEL );
mux #(1 , RSTTYPE) CYO_part (Post_cout , clk , RSTCARRYIN , CECARRYIN ,  CARRYOUT , CARRYOUTREG);
mux #(48 , RSTTYPE) P_part (Post_out , clk , RSTP , CEP ,  P , PREG);

assign CARRYOUTF = CARRYOUT ;
endmodule
```

Testbench Code

```
module DSP48A1_tb ( );
parameter A0REG = 0 ; // If equal (0) --> No register, else --> Registered
parameter A1REG = 1 ; // If equal (0) --> No register, else --> Registered
parameter B0REG = 0 ; // If equal (0) --> No register, else --> Registered
parameter B1REG = 1 ; // If equal (0) --> No register, else --> Registered
parameter CREG = 1  ; // If equal (0) --> No register, else --> Registered
parameter DREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter MREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter PREG = 1  ; // If equal (0) --> No register, else --> Registered
parameter CARRYINREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter CARRYOUTREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter OPMODEREG = 1 ; // If equal (0) --> No register, else --> Registered
parameter CARRYINSEL = "OPMODE5"; // Select between (CARRYIN or OPMODE5)
parameter B_INPUT = "DIRECT" ; // Select between (DIRECT or CASCADE)
parameter RSTTYPE = "SYNC" ; // Select between (SYNC or A SYNC)


reg [17:0] A , B , D ;
reg [17:0] BCIN ;
reg [47:0] C ;
reg CARRYIN , clk ;
reg [47:0] PCIN ;  // Cascade input for Port P
reg [7:0] OPMODE ;

reg CEA ; // Clock enable for the A port registers: (A0REG & A1REG)
reg CEB ; // Clock enable for the B port registers: (B0REG & B1REG)
reg CEC ; // Clock enable for the C port registers (CREG)
reg CECARRYIN ; // Clock enable for the carry-in register (CYI) and the carry-out register (CYO)
reg CED ; // Clock enable for the D port register (DREG)
reg CEM ; // Clock enable for the multiplier register (MREG)
reg CEOPMODE ; // Clock enable for the opmode register (OPMODEREG)
reg CEP ; // Clock enable for the P output port registers (PREG = 1)

reg RSTA ; // Reset for the A registers: (A0REG & A1REG)
reg RSTB ; // Reset for the B registers: (B0REG & B1REG)
reg RSTC ; // Reset for the C registers (CREG)
reg RSTCARRYIN ; // Reset for the carry-in register (CYI) and the carry-out register (CYO)
reg RSTD ; // Reset for the D register (DREG)
reg RSTM ; // Reset for the multiplier register (MREG)
reg RSTOPMODE ; // Reset for the opmode register (OPMODEREG)
reg RSTP ; // Reset for the P output registers (PREG = 1)

wire [35:0] M_tb ;
wire [17:0] BCOUT_tb ; //Cascade output for Port B
wire [47:0] P_tb ;
wire [47:0] PCOUT_tb ; // Cascade output for Port P
wire CARRYOUT_tb , CARRYOUTF_tb ;

// module DSP48A1 (A , B , D , BCIN , C , CARRYIN , clk , PCIN , OPMODE , CEA , CEB , CEC , CECARRYIN , CED , CEM , CEOPMODE , CEP , RSTA , RSTB , RSTC , RST

DSP48A1 testbench (A , B , D , BCIN , C , CARRYIN , clk , PCIN , OPMODE , CEA , CEB , CEC , CECARRYIN , CED , CEM , CEOPMODE , CEP , RSTA , RSTB , RSTC , RST

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


initial begin
// 2.1. Verify Reset Operation
    RSTA = 1 ;
    RSTB = 1 ;
    RSTC = 1 ;
    RSTCARRYIN = 1 ;
    RSTD = 1 ;
    RSTM = 1 ;
    RSTOPMODE = 1 ;
    RSTP = 1 ;

    repeat (15)begin
        A = $random ;
        B = $random ;
        D = $random ;
        BCIN = $random ;
        C = $random ;
        CARRYIN = $random ;
        PCIN = $random ;
        OPMODE = $random ;

        @(negedge clk )
        if(M_tb !== 0 || BCOUT_tb !== 0 || P_tb !== 0 || PCOUT_tb !== 0 || CARRYOUT_tb !== 0 || CARRYOUTF_tb !== 0 ) begin
            $display ("Error - Output is incorrect");
            $stop;
        end
    end
```

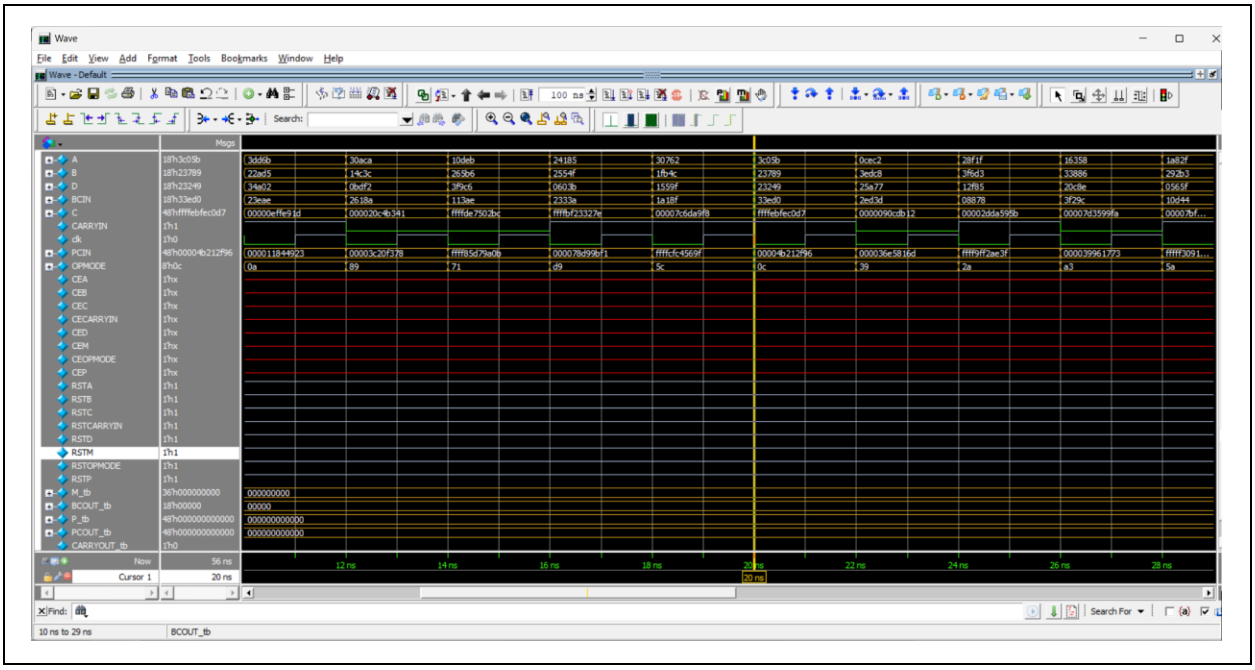
```
    RSTA = 0 ;
    RSTB = 0 ;
    RSTC = 0 ;
    RSTCARRYIN = 0 ;
    RSTD = 0 ;
    RSTM = 0 ;
    RSTOPMODE = 0 ;
    RSTP = 0 ;

    CEA = 1 ;
    CEB = 1 ;
    CEC = 1 ;
    CECARRYIN = 1 ;
    CED = 1 ;
    CEM = 1 ;
    CEOPMODE = 1 ;
    CEP = 1 ;

// 2.2. Verify DSP Path 1
    OPMODE = 8'b1101_1101 ;
    A = 20 ;
    B = 10 ;
    C = 350 ;
    D = 25 ;
    BCIN = $random ;
    CARRYIN = $random ;
    PCIN = $random ;

    @(negedge clk)
    @(negedge clk)
    @(negedge clk)
    @(negedge clk)
    if(BCOUT_tb !== 'hf || M_tb !== 'h12c || P_tb !== PCOUT_tb || PCOUT_tb !== 'h32 || CARRYOUT_tb !== 0  || CARRYOUTF_tb !== 0)begin
        $display ("Error - Output is incorrect");
        $stop;
    end

// 2.3. Verify DSP Path 2
    OPMODE = 8'b00010000 ;
    BCIN = $random ;
    CARRYIN = $random ;
    PCIN = $random ;

    @(negedge clk)
    @(negedge clk)
    @(negedge clk)
    if(BCOUT_tb !== 'h23 || M_tb !== 'h2bc || P_tb !== PCOUT_tb || PCOUT_tb !== 0 || CARRYOUT_tb !== 0  || CARRYOUTF_tb !== 0)begin
        $display ("Error - Output is incorrect");
        $stop;
    end

// 2.4. Verify DSP Path 3
    OPMODE = 8'b00001010 ;
    BCIN = $random ;
    CARRYIN = $random ;
    PCIN = $random ;

    @(negedge clk)
    @(negedge clk)
    @(negedge clk)
    if(BCOUT_tb !== 'ha || M_tb !== 'hc8 || P_tb !== PCOUT_tb  || CARRYOUT_tb !== CARRYOUTF_tb)begin
        $display ("Error - Output is incorrect");
        $stop;
    end

// 2.5. Verify DSP Path 4
    OPMODE = 8'b10100111 ;
    A = 5 ;
    B = 6 ;
    PCIN = 3000 ;
    BCIN = $random ;
    CARRYIN = $random ;

    @(negedge clk)
    @(negedge clk)
    @(negedge clk)
    if(BCOUT_tb !== 'h6 || M_tb !== 'h1e || P_tb !== PCOUT_tb || PCOUT_tb !== 'hfe6fffec0bb1  || CARRYOUT_tb !== CARRYOUTF_tb   || CARRYOUTF_tb !== 0)begin
        $display ("Error - Output is incorrect");
        $display ("BCOUT_tb = %0h , M_tb = %0h , P_tb = %0h , PCOUT_tb = %0h , CARRYOUT_tb = %0h , CARRYOUTF_tb = %0h" , BCOUT_tb , M_tb , P_tb , PCOUT_tb , CARRYOUT_tb , CARRYOUTF_tb );

        $stop;
    end


$stop;
end
endmodule
```

Do file

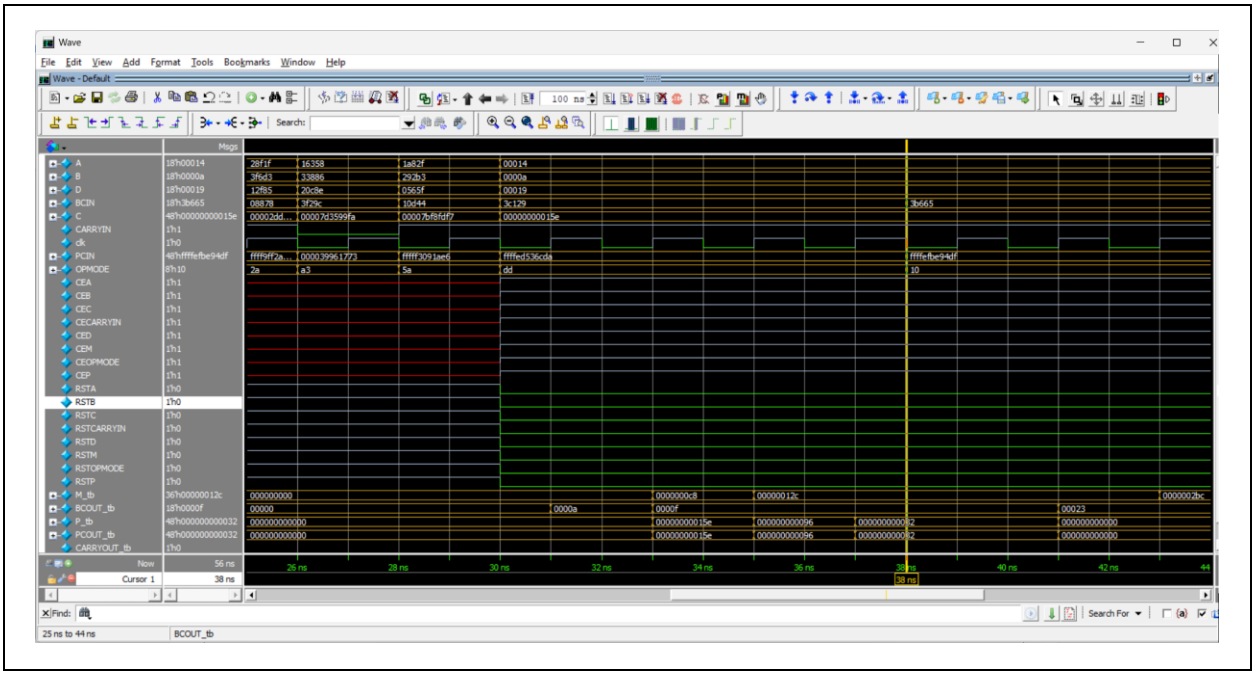
```
vlib work
vlog DSP48A1.v tb.v
vsim -voptargs=+acc work.DSP48A1_tb
add wave *
run -all
#quit -sim
```

QuestaSim Snippets

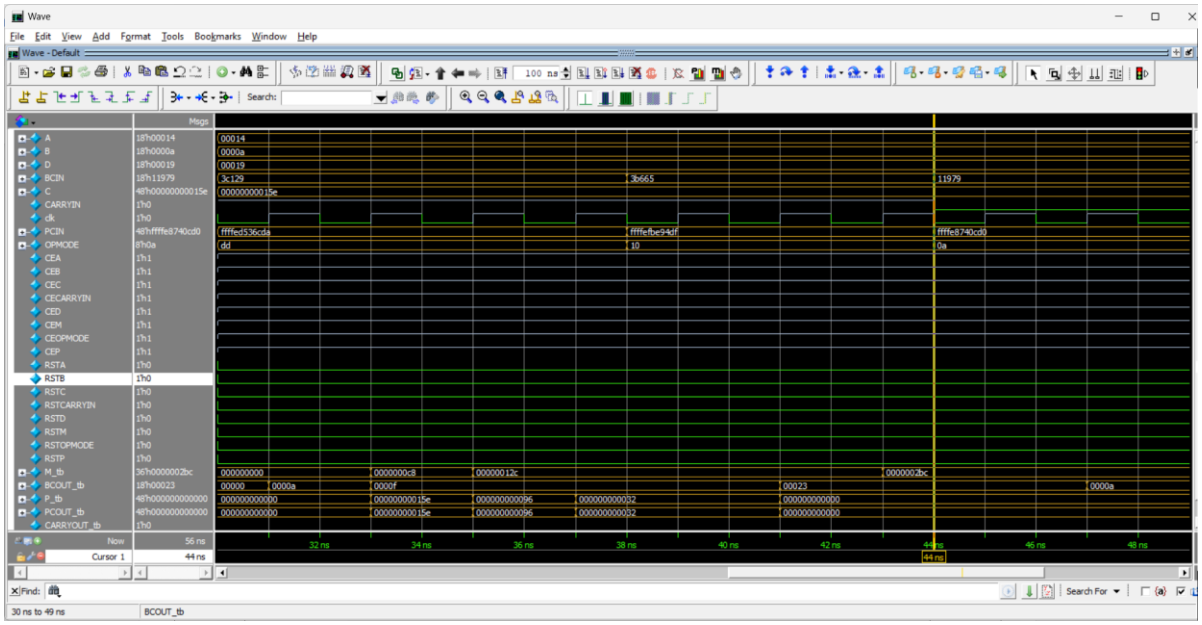
→ Verify Reset Operation



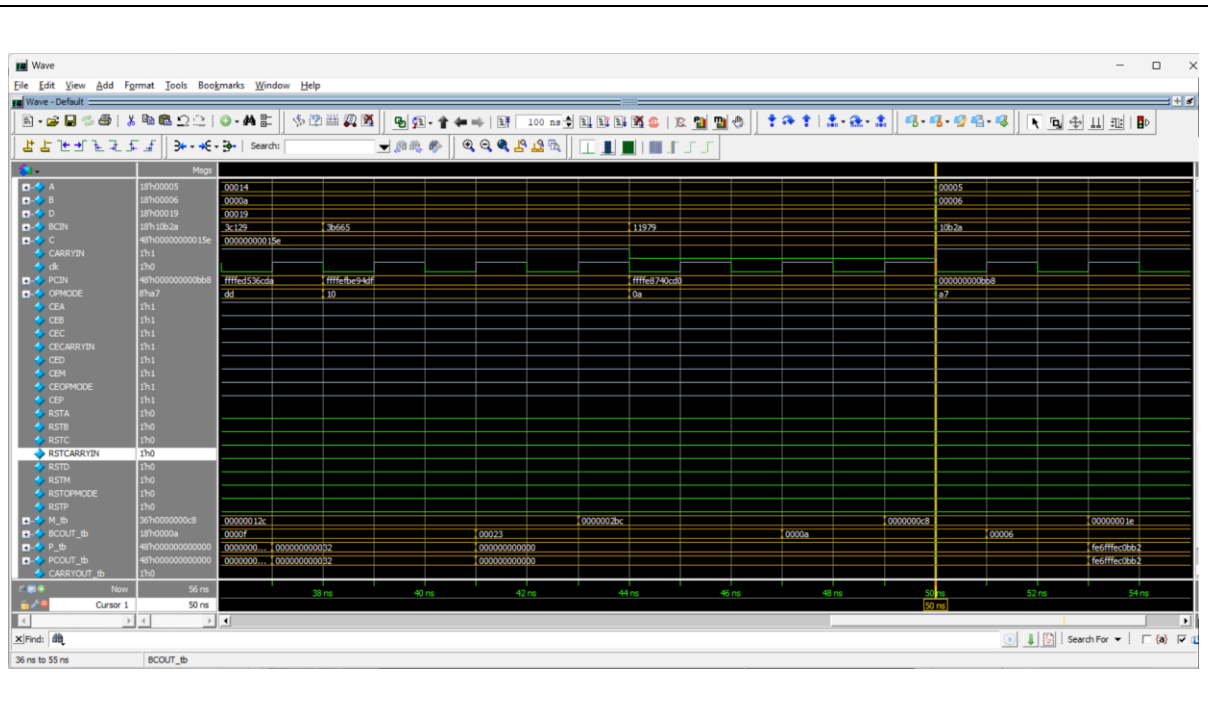
→ Verify DSP Path 1



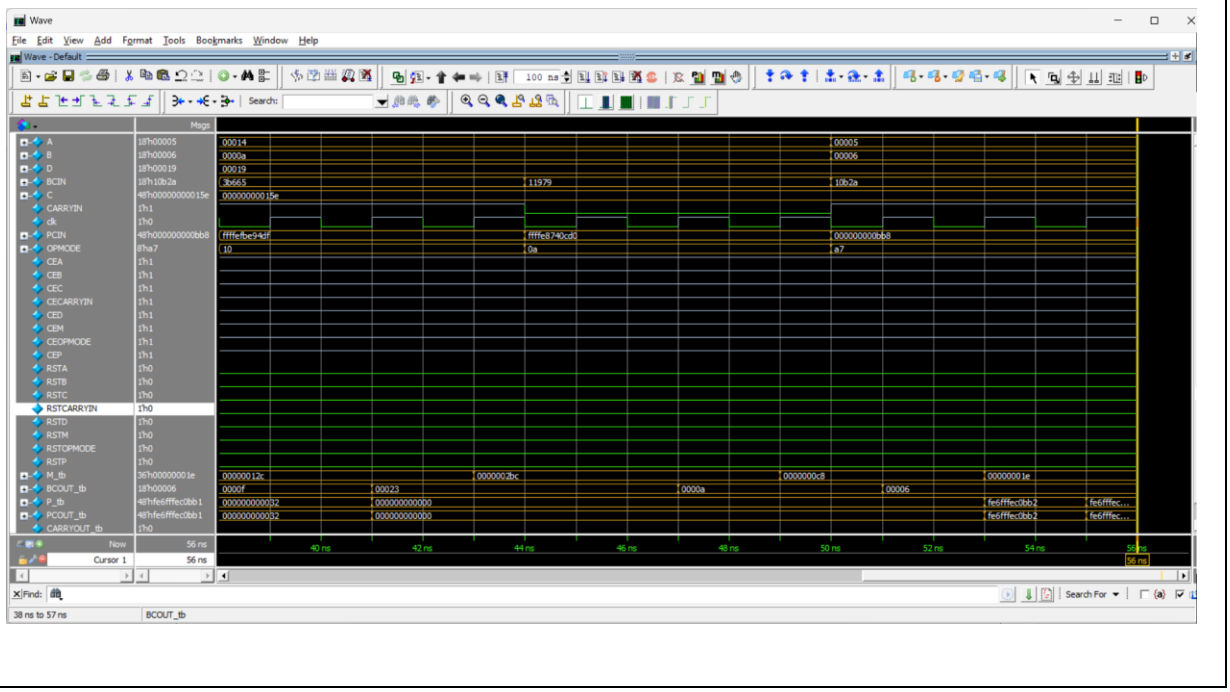
→ Verify DSP Path 2



→ Verify DSP Path 3



→ Verify DSP Path 4



Constraint File

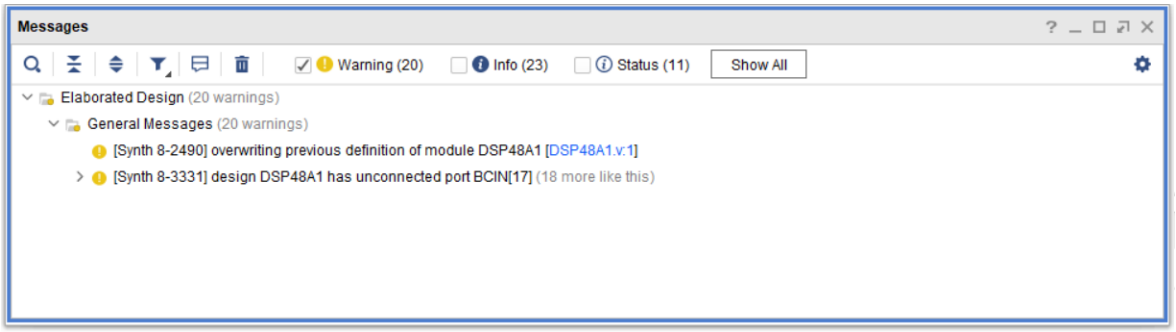
```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

## 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
```

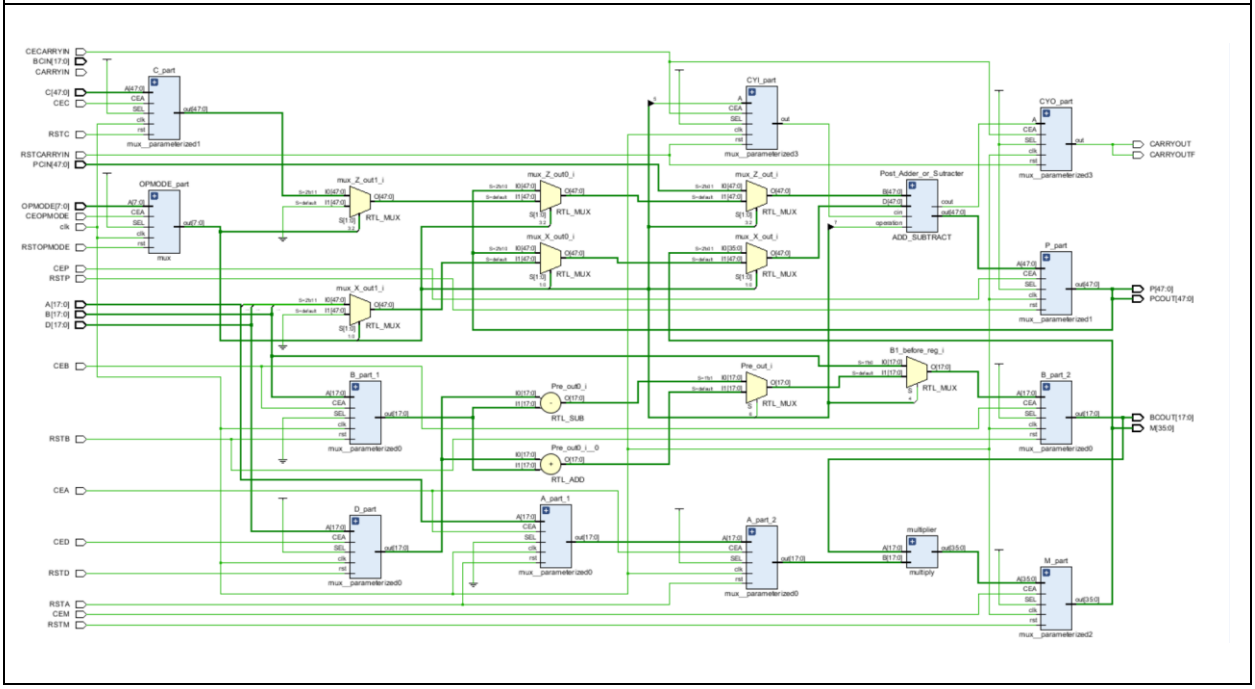
```
##Pmod Header JC
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports {JC[0]}];#Sch name = JC1
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports {JC[1]}];#Sch name = JC2
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports {JC[2]}];#Sch 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 {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_IBU
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]} {PCI
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 36 [get_debug_ports u_ila_0/probe2]
connect_debug_port u_ila_0/probe2 [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_OBU
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe3]
set_property port_width 48 [get_debug_ports u_ila_0/probe3]
connect_debug_port u_ila_0/probe3 [get_nets [list {PCOUT_OBUF[0]} {PCOUT_OBUF[1]} {PCOUT_OBUF[2]} {PCOUT_OBUF[3]} {PCOUT_OBUF[4]} {PCOUT_OBUF[5]} {PCOUT_OBUF[6]} {PCOUT_OBUF[7]} {PCOUT_OBUF
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe4]
set_property port_width 18 [get_debug_ports u_ila_0/probe4]
connect_debug_port u_ila_0/probe4 [get_nets [list {A_IBUF[0]} {A_IBUF[1]} {A_IBUF[2]} {A_IBUF[3]} {A_IBUF[4]} {A_IBUF[5]} {A_IBUF[6]} {A_IBUF[7]} {A_IBUF[8]} {A_IBUF[9]} {A_IBUF[10]} {A_IBU
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 8 [get_debug_ports u_ila_0/probe5]
connect_debug_port u_ila_0/probe5 [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/probe6]
set_property port_width 18 [get_debug_ports u_ila_0/probe6]
connect_debug_port u_ila_0/probe6 [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_IBU
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 48 [get_debug_ports u_ila_0/probe7]
```

```
connect_debug_port u_ila_0/probe7 [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_
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 18 [get_debug_ports u_ila_0/probe8]
connect_debug_port u_ila_0/probe8 [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_C
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe9]
set_property port_width 1 [get_debug_ports u_ila_0/probe9]
connect_debug_port u_ila_0/probe9 [get_nets [list CARRYOUTF_OBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe10]
set_property port_width 1 [get_debug_ports u_ila_0/probe10]
connect_debug_port u_ila_0/probe10 [get_nets [list CEA_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe11]
set_property port_width 1 [get_debug_ports u_ila_0/probe11]
connect_debug_port u_ila_0/probe11 [get_nets [list CEB_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe12]
set_property port_width 1 [get_debug_ports u_ila_0/probe12]
connect_debug_port u_ila_0/probe12 [get_nets [list CEC_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe13]
set_property port_width 1 [get_debug_ports u_ila_0/probe13]
connect_debug_port u_ila_0/probe13 [get_nets [list CECARRYIN_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe14]
set_property port_width 1 [get_debug_ports u_ila_0/probe14]
connect_debug_port u_ila_0/probe14 [get_nets [list CED_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe15]
set_property port_width 1 [get_debug_ports u_ila_0/probe15]
connect_debug_port u_ila_0/probe15 [get_nets [list CEM_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe16]
set_property port_width 1 [get_debug_ports u_ila_0/probe16]
connect_debug_port u_ila_0/probe16 [get_nets [list CEOPMODE_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe17]
set_property port_width 1 [get_debug_ports u_ila_0/probe17]
connect_debug_port u_ila_0/probe17 [get_nets [list CEP_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe18]
set_property port_width 1 [get_debug_ports u_ila_0/probe18]
connect_debug_port u_ila_0/probe18 [get_nets [list clk_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe19]
set_property port_width 1 [get_debug_ports u_ila_0/probe19]
connect_debug_port u_ila_0/probe19 [get_nets [list RSTA_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe20]
set_property port_width 1 [get_debug_ports u_ila_0/probe20]
connect_debug_port u_ila_0/probe20 [get_nets [list RSTB_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe21]
set_property port_width 1 [get_debug_ports u_ila_0/probe21]
connect_debug_port u_ila_0/probe21 [get_nets [list RSTC_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe22]
set_property port_width 1 [get_debug_ports u_ila_0/probe22]
connect_debug_port u_ila_0/probe22 [get_nets [list RSTCARRYIN_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe23]
set_property port_width 1 [get_debug_ports u_ila_0/probe23]
connect_debug_port u_ila_0/probe23 [get_nets [list RSTD_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe24]
set_property port_width 1 [get_debug_ports u_ila_0/probe24]
connect_debug_port u_ila_0/probe24 [get_nets [list RSTM_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe25]
set_property port_width 1 [get_debug_ports u_ila_0/probe25]
connect_debug_port u_ila_0/probe25 [get_nets [list RSTOPMODE_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe26]
set_property port_width 1 [get_debug_ports u_ila_0/probe26]
connect_debug_port u_ila_0/probe26 [get_nets [list RSTP_IBUF]]
set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
connect_debug_port dbg_hub/clk [get_nets clk_IBUF_BUFG]
```

Elaboration

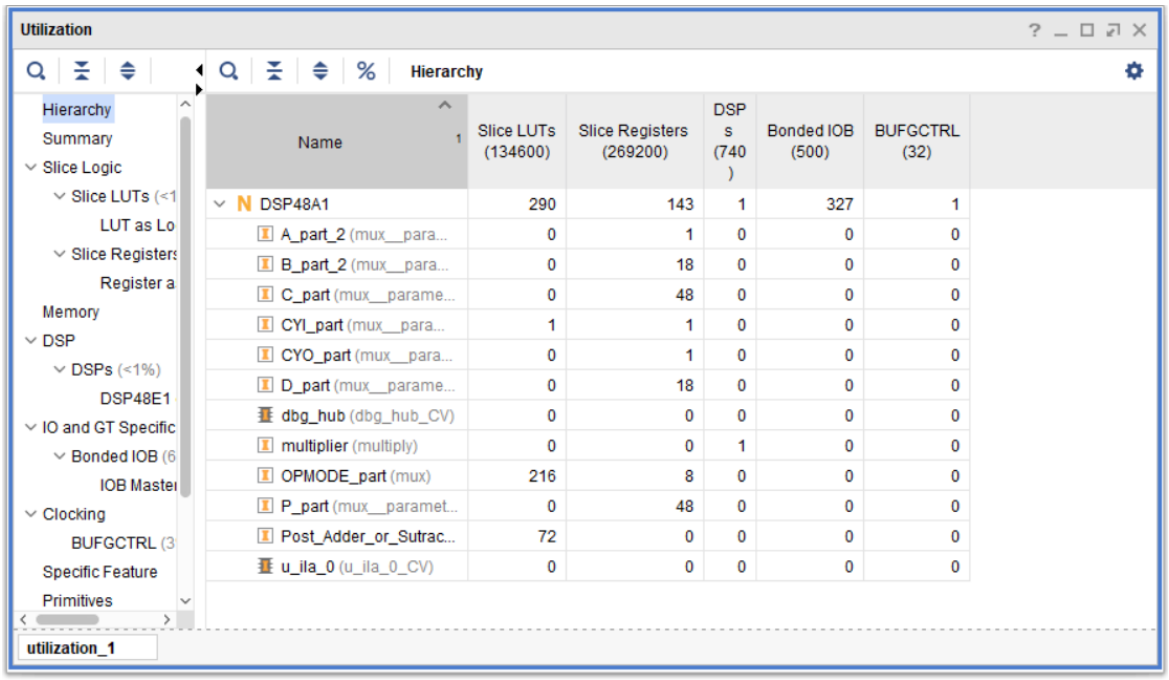
(“Messages” tab & Schematic snippets)

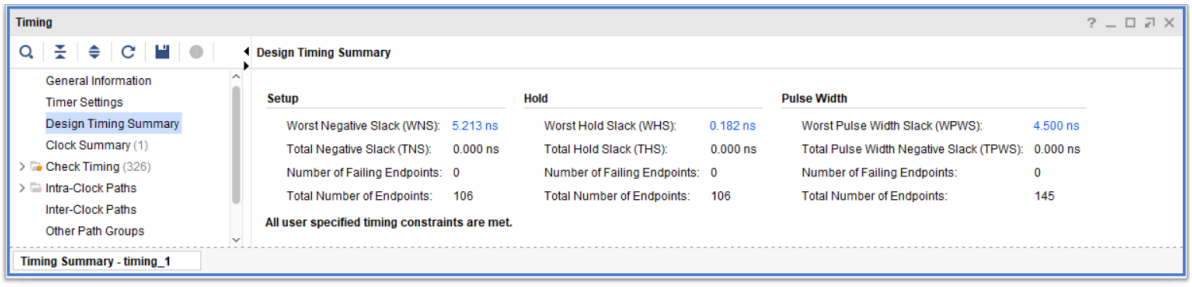


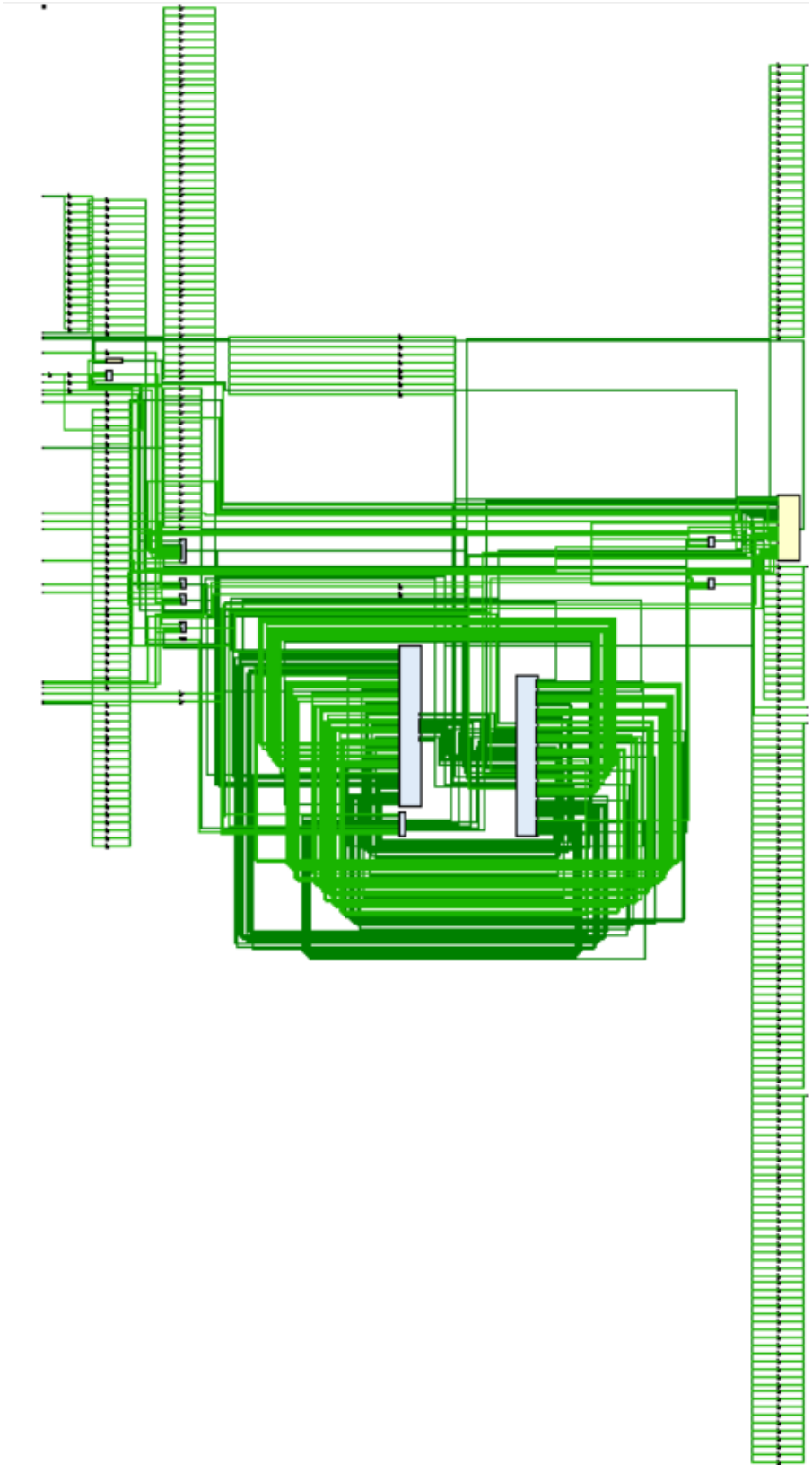


Synthesis

(“Messages” tab, Utilization report, timing report & Schematic snippets)

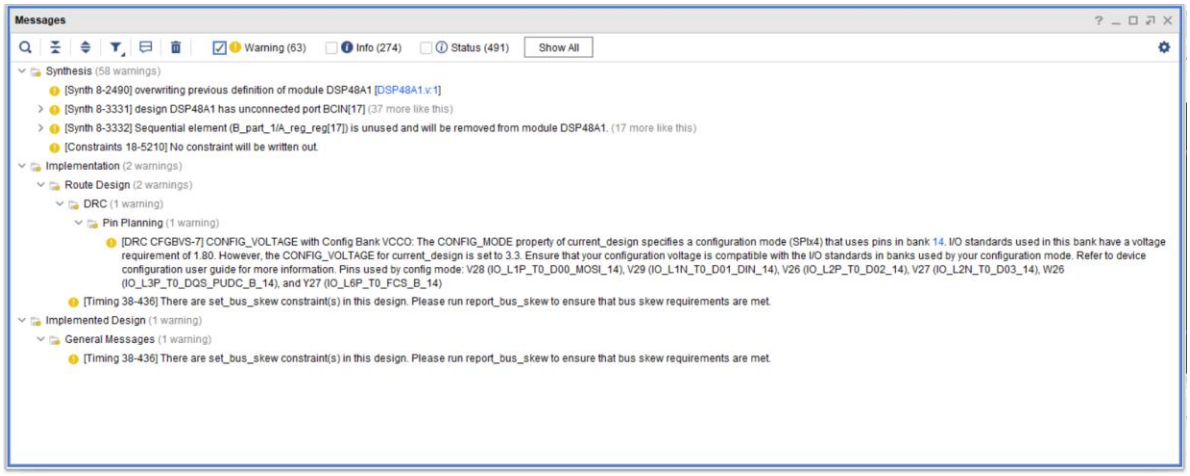


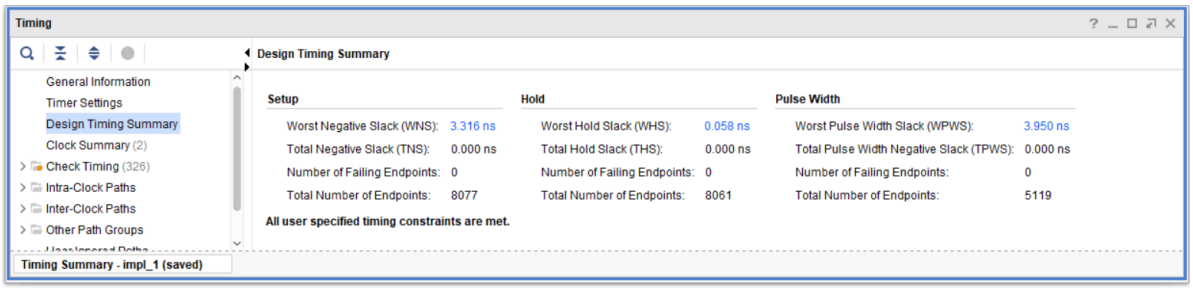


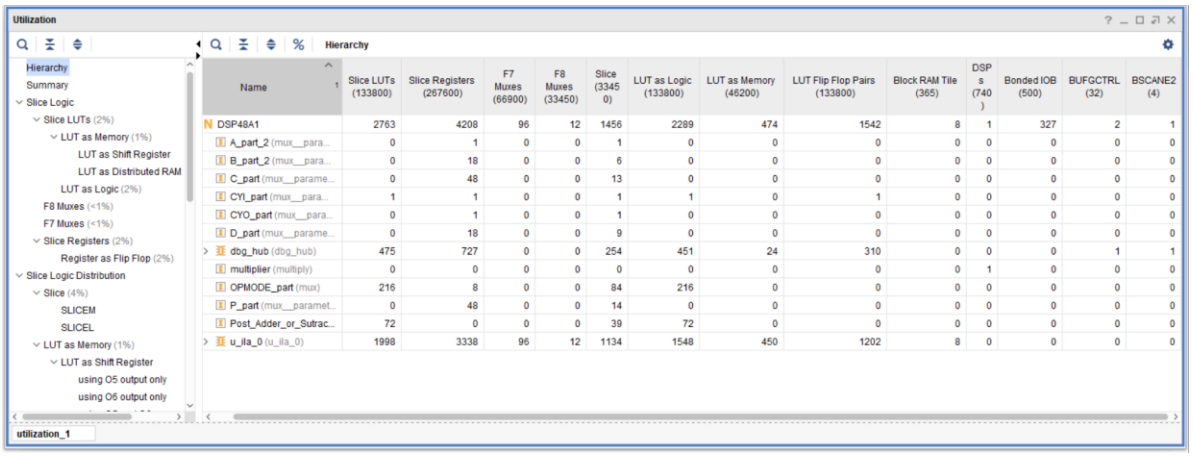


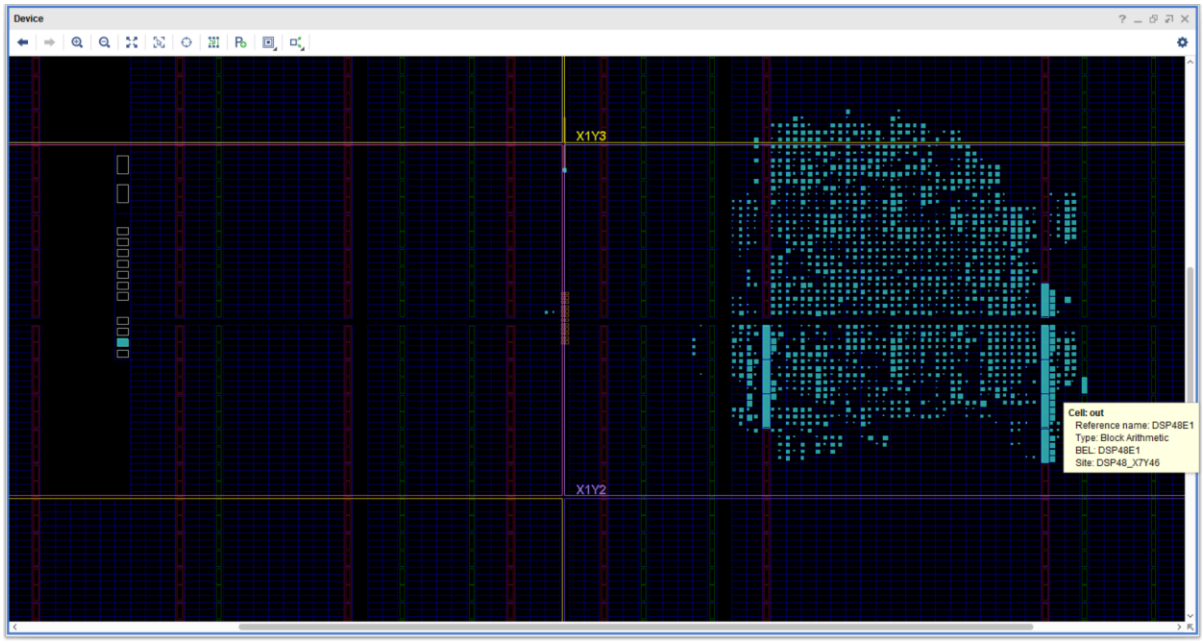
Implementation

(“Messages” tab, Utilization report, timing report & device snippets)

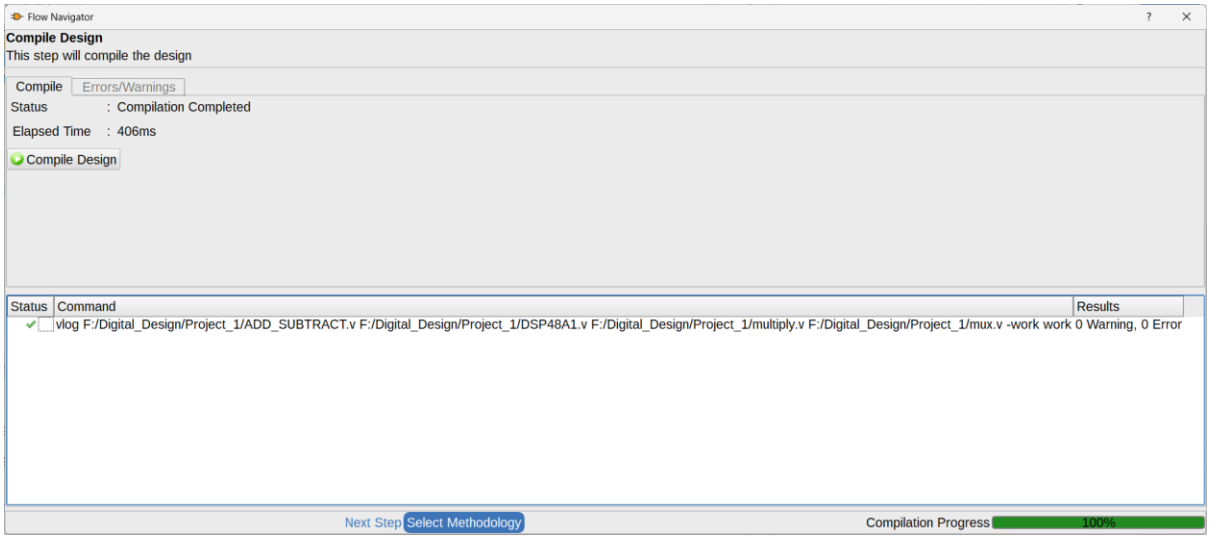


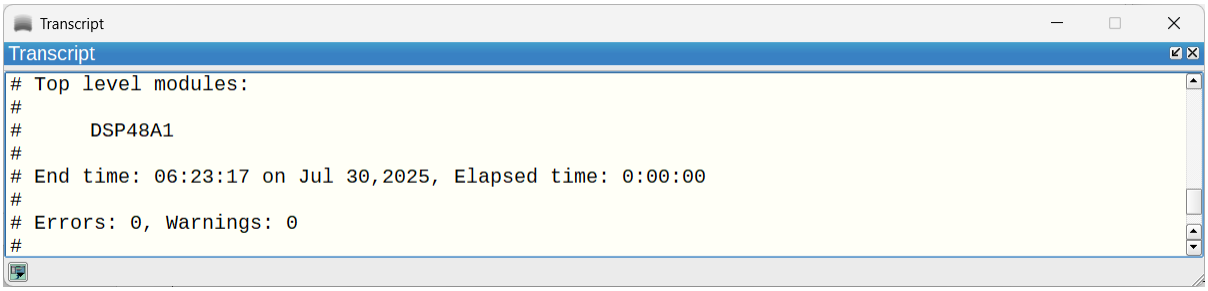





Device
X1Y3
X1Y2
Cell: out
Reference name: DSP48E1
Type: Block Arithmetic
BEL: DSP48E1
Site: DSP48_X7Y46

Questa Lint


Flow Navigator
Compile Design
This step will compile the design
Compile
Errors/Warnings
Status : Compilation Completed
Elapsed Time : 406ms
Compile Design
Status
Command
Results
vlog F:/Digital_Design/Project_1/ADD_SUBTRACT.v F:/Digital_Design/Project_1/DSP48A1.v F:/Digital_Design/Project_1/multiply.v F:/Digital_Design/Project_1/mux.v -work work 0 Warning, 0 Error
Next Step
Select Methodology
Compilation Progress
100%


Transcript
Transcript
Top level modules:
#
DSP48A1
#
End time: 06:23:17 on Jul 30,2025, Elapsed time: 0:00:00
#
Errors: 0, Warnings: 0
#