Name: Omar Ahmed Gamal
Email: omar.ahmed.ga.04@gmail.com

Project 1 (DSP48A1)

1-RTL code:

1) The design of the register followed by mux whose selector is a parameter:

reg_mux_input.v > reg_mux_input

```
1   module reg_mux_input(in,clk,rst,ce,out);
2   parameter SELECTOR=1;
3   parameter INPUT_WIDTH=18;
4   parameter RSTTYPE="SYNC";
5       input  clk, rst, ce;
6       input  [INPUT_WIDTH-1:0] in;
7       output [INPUT_WIDTH-1:0] out;
8
9       reg [INPUT_WIDTH-1:0] q;
10  generate
11      if(RSTTYPE == "SYNC") begin
12          always @(posedge clk) begin
13              if (rst) begin
14                  q <= 0;
15              end else if (ce) begin
16                  q <= in;
17              end
18          end
19      end else if (RSTTYPE == "ASYNC") begin
20          always @(posedge clk or posedge rst) begin
21              if (rst) begin
22                  q <= 0;
23              end else if (ce) begin
24                  q <= in;
25              end
26          end
27      end else begin
28          always @(posedge clk) begin
29              if (rst) begin
30                  q <= 0;
31              end else if (ce) begin
32                  q <= in;
33              end
34          end
35      end
36  endgenerate
37  assign out = (SELECTOR == 1) ? q : in;
38
39  endmodule
```

2) The DSP48A1 top module (which is the main design):

DSP48A1_TOP.V > DSP48A1

```
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,CEOPMODE,
PCIN,BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF);
parameter A0REG=0;
parameter A1REG=1;
parameter B0REG=0;
parameter B1REG=1;
parameter CREG=1;
parameter DREG=1;
parameter PREG=1;
parameter MREG=1;
parameter CARRYINREG=1;
parameter OPMODEREG=1;
parameter CARRYOUTREG=1;
parameter CARRYINSEL="OPMODE5";
parameter B_INPUT="DIRECT";
parameter RSTTYPE="SYNC";
input [17:0] A, B, D;
input [47:0] C;
input CLK, CARRYIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPMODE, CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPMODE;
input [7:0] OPMODE;
input [17:0]  BCIN;
input [47:0] PCIN;
output [47:0]  PCOUT;
output [17:0]  BCOUT;
output [47:0] P;
output [35:0] M;
output CARRYOUT, CARRYOUTF;
wire [17:0] a0reg_out, a1reg_out;
wire [17:0] b0reg_out, b1reg_out,mux_bcascade_out;
wire [47:0] creg_out;
wire [17:0] dreg_out;
wire [7:0] opmodereg_out;
wire [35:0] mreg_out;
wire [47:0] DAB_concatenated;
wire carrycascade_out;
wire carryinreg_out;
wire carryoutreg_out;
wire [47:0] preg_out;
wire [47:0] mreg_extended_out;
reg [17:0] preaddsub_out;
reg [17:0] mux_preaddsub_out;
reg [35:0] mul_out;
reg [47:0] mux_x_out;
reg [47:0] mux_z_out;
reg [47:0] postaddsub_out;
reg carryout_postaddsub;
assign mux_bcascade_out = (B_INPUT == "DIRECT") ? B : (B_INPUT == "CASCADE") ? BCIN : 18'b0; // MUX THAT INDICATES THE INPUT FOR B0REG
assign carrycascade_out = (CARRYINSEL == "OPMODE5") ? opmodereg_out[5] : (CARRYINSEL == "CARRYIN") ? CARRYIN : 1'b0;//MUX THAT INDICATES THE INPUT FOR CARRYINREG
assign M=mreg_out;
assign P=preg_out;
assign PCOUT = preg_out;
assign mreg_extended_out = {12'b0,mreg_out}; // Extend MREG to 48 bits to enter the x mux with 48bit output
assign BCOUT = b1reg_out;
assign CARRYOUT = carryoutreg_out;
assign CARRYOUTF = CARRYOUT; // CARRYOUTF is the same as CARRYOUT in DSP48A1
assign DAB_concatenated={dreg_out[11:0],a1reg_out[17:0],b1reg_out[17:0]}; // Concatenate 48 bits
```

```
// Instantiate the input registers based on the parameters
//A0REG
reg_mux_input #(.SELECTOR(A0REG), .INPUT_WIDTH(18), .RSTTYPE(RSTTYPE)) a0reg (
    .in(A),
    .clk(CLK),
    .rst(RSTA),
    .ce(CEA),
    .out(a0reg_out)
);
//A1REG
reg_mux_input #(.SELECTOR(A1REG), .INPUT_WIDTH(18), .RSTTYPE(RSTTYPE)) a1reg (
    .in(a0reg_out),
    .clk(CLK),
    .rst(RSTA),
    .ce(CEA),
    .out(a1reg_out)
);
//B0REG
reg_mux_input #(.SELECTOR(B0REG), .INPUT_WIDTH(18), .RSTTYPE(RSTTYPE)) b0reg (
    .in(mux_bcascade_out),
    .clk(CLK),
    .rst(RSTB),
    .ce(CEB),
    .out(b0reg_out)
);
//B1REG
reg_mux_input #(.SELECTOR(B1REG), .INPUT_WIDTH(18), .RSTTYPE(RSTTYPE)) b1reg (
    .in(mux_preaddsub_out),
    .clk(CLK),
    .rst(RSTB),
    .ce(CEB),
    .out(b1reg_out)
);
//CREG
reg_mux_input #(.SELECTOR(CREG), .INPUT_WIDTH(48), .RSTTYPE(RSTTYPE)) creg (
    .in(C),
    .clk(CLK),
    .rst(RSTC),
    .ce(CEC),
    .out(creg_out)
);
//DREG
reg_mux_input #(.SELECTOR(DREG), .INPUT_WIDTH(18), .RSTTYPE(RSTTYPE)) dreg (
    .in(D),
    .clk(CLK),
    .rst(RSTD),
    .ce(CED),
    .out(dreg_out)
);
//OPMODEREG
reg_mux_input #(.SELECTOR(OPMODEREG), .INPUT_WIDTH(8), .RSTTYPE(RSTTYPE)) opmodereg (
    .in(OPMODE),
    .clk(CLK),
    .rst(RSTOPMODE),
    .ce(CEOPMODE),
    .out(opmodereg_out)
);
//MREG
reg_mux_input #(.SELECTOR(MREG), .INPUT_WIDTH(36), .RSTTYPE(RSTTYPE)) mreg (
    .in(mul_out),
    .clk(CLK),
    .rst(RSTM),
    .ce(CEM),
    .out(mreg_out)
);
//CARRYINREG
reg_mux_input #(.SELECTOR(CARRYINREG), .INPUT_WIDTH(1), .RSTTYPE(RSTTYPE)) carryinreg (
    .in(carrycascade_out),
    .clk(CLK),
    .rst(RSTCARRYIN),
    .ce(CECARRYIN),
    .out(carryinreg_out)
);
//CARRYOUTREG
reg_mux_input #(.SELECTOR(CARRYOUTREG), .INPUT_WIDTH(1), .RSTTYPE(RSTTYPE)) carryoutreg (
    .in(carryout_postaddsub),
    .clk(CLK),
    .rst(RSTCARRYIN),
    .ce(CECARRYIN),
    .out(carryoutreg_out)
);
//PREG
reg_mux_input #(.SELECTOR(PREG), .INPUT_WIDTH(48), .RSTTYPE(RSTTYPE)) preg (
    .in(postaddsub_out),
    .clk(CLK),
    .rst(RSTP),
    .ce(CEP),
    .out(preg_out)
);
```

```
always @(dreg_out or b0reg_out or opmodereg_out[6] or opmodereg_out[4] or b1reg_out or a1reg_out) begin
   case (opmodereg_out[6])
       1'b0: preaddsub_out = dreg_out + b0reg_out; // ADD
       1'b1: preaddsub_out = dreg_out - b0reg_out; // SUB
   endcase
    if (opmodereg_out[4]) begin
        mux_preaddsub_out = preaddsub_out; // If OPMODE[4]=1, use the preaddsub_out
    end else begin
        mux_preaddsub_out = b0reg_out; // Otherwise, use b0reg_out
    end
    mul_out = a1reg_out * b1reg_out; // Perform multiplication
end
always @(preg_out or DAB_concatenated or mreg_extended_out or creg_out or PCIN or opmodereg_out [3:0] or carryinreg_out) begin
    //multiplexior X
    case (opmodereg_out[1:0])
       2'b00: mux_x_out = 48'h0; // 0
       2'b01: mux_x_out = mreg_extended_out; // multiplier output
       2'b10: mux_x_out = preg_out; // PREG output
       2'b11: mux_x_out = DAB_concatenated; // Concatenated DAB
    endcase
    //multiplexior Z
    case (opmodereg_out[3:2])
       2'b00: mux_z_out = 48'h0; // 0
       2'b01: mux_z_out = PCIN; // PCIN input
       2'b10: mux_z_out = preg_out; // PREG output
       2'b11: mux_z_out = creg_out; // CREG output
    endcase
    //postaddsub_out
    if (!opmodereg_out[7]) begin
      {carryout_postaddsub, postaddsub_out} = mux_x_out + mux_z_out + carryinreg_out; // If OPMODE[7]=0, perform addition
    end else begin
       {carryout_postaddsub, postaddsub_out} = mux_z_out - (mux_x_out + carryinreg_out); // Otherwise, perform subtraction

    end
end



endmodule
```

2-Testbench code:

DSP48A1_tb.v > DSP48A1_tb

```
module DSP48A1_tb ();
// parameters added if needed to change them
parameter A0REG=0;parameter A1REG=1;parameter B0REG=0;parameter B1REG=1;
parameter CREG=1;parameter DREG=1;parameter PREG=1;parameter MREG=1;
parameter CARRYINREG=1;parameter OPMODEREG=1;parameter CARRYOUTREG=1;
parameter CARRYINSEL="OPMODE5";parameter B_INPUT="DIRECT";parameter RSTTYPE="SYNC";
//input signals
reg [17:0] A, B, D;
reg [47:0] C;
reg CLK, CARRYIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPMODE;
reg CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPMODE;
reg [7:0] OPMODE;
reg [17:0] BCIN;
reg [47:0] PCIN;
//output signals
wire [17:0] BCOUT;
wire [47:0] PCOUT;
wire [47:0] P;
wire [35:0] M;
wire CARRYOUT, CARRYOUTF;

reg [47:0] past_p_out;
reg past_carryout_out;
// Instantiate the DSP48A1_TOP module
DSP48A1 #(
    .A0REG(A0REG), .A1REG(A1REG), .B0REG(B0REG), .B1REG(B1REG),
    .CREG(CREG), .DREG(DREG), .PREG(PREG), .MREG(MREG),
    .CARRYINREG(CARRYINREG), .OPMODEREG(OPMODEREG), .CARRYOUTREG(CARRYOUTREG),
    .CARRYINSEL(CARRYINSEL), .B_INPUT(B_INPUT), .RSTTYPE(RSTTYPE)
) DUT (
    .A(A), .B(B), .C(C), .D(D),
    .CLK(CLK), .CARRYIN(CARRYIN),
    .OPMODE(OPMODE), .BCIN(BCIN),
    .RSTA(RSTA), .RSTB(RSTB), .RSTM(RSTM),
    .RSTP(RSTP), .RSTC(RSTC), .RSTD(RSTD),
    .RSTCARRYIN(RSTCARRYIN), .RSTOPMODE(RSTOPMODE),
    .CEA(CEA), .CEB(CEB), .CEM(CEM),
    .CEP(CEP), .CEC(CEC), .CED(CED),
    .CECARRYIN(CECARRYIN), .CEOPMODE(CEOPMODE),
    .PCIN(PCIN),
    .BCOUT(BCOUT),
    .PCOUT(PCOUT),
    .P(P),
    .M(M),
    .CARRYOUT(CARRYOUT),
    .CARRYOUTF(CARRYOUTF)
);
// clock generation
initial begin
    CLK = 0;
    forever #1 CLK = ~CLK;
end
```

Verify Reset Operation:

```
//test module
initial begin
  //test 1: Verify Reset Operation
  RSTA = 1; RSTB = 1; RSTM = 1; RSTP = 1; RSTC = 1; RSTD = 1; RSTCARRYIN = 1; RSTOPMODE = 1; // Assert all resets by setting them 1
  A=$random; B=$random; C=$random; D=$random; OPMODE=$random; BCIN=$random; PCIN=$random;CARRYIN=$random;
  CEA=$random; CEB=$random; CEM=$random; CEP=$random; CEC=$random; CED=$random; CECARRYIN=$random; CEOPMODE=$random;//Drive remaining inputs with (random) values.
  @(negedge CLK);
  if (BCOUT!=0||PCOUT!=0||P!=0||M!=0||CARRYOUT!=0||CARRYOUTF!=0) begin
    $display("ERROR - (Verify Reset Operation test) Failed - Outputs not zero after reset");
    $stop;
  end else begin
    $display("Correct outputs - (Verify Reset Operation test) Passed - Outputs are zero after reset");
  end
  RSTA = 0; RSTB = 0; RSTM = 0; RSTP = 0; RSTC = 0; RSTD = 0; RSTCARRYIN = 0; RSTOPMODE = 0; // Deassert resets
  CEA=1; CEB=1; CEM=1; CEP=1; CEC=1; CED=1; CECARRYIN=1; CEOPMODE=1; //assert all clock enable signals
```

Verify DSP Path 1:

```
//test 2: Verify DSP Path 1
OPMODE = 8'b11011101;
  A=18'd20; B=18'd10; C=48'd350; D=18'd25;
  BCIN=$random; PCIN=$random; CARRYIN=$random;
  repeat(4) @(negedge CLK); // Wait for 4 clock cycles
  if (BCOUT!=18'hf) begin
      $display("ERROR - (verify DSP path 1 test) Failed - BCOUT output is wrong value BCOUT=%h  expected= 18'hf", BCOUT);
      $stop;
  end else begin
      $display("Correct BCOUT output  BCOUT=%h  expected= 18'hf", BCOUT);
  end
   if (M!=36'h12c) begin
      $display("ERROR - (verify DSP path 1 test) Failed - M output is wrong value M=%h  expected= 36'h12c", M);
      $stop;
  end else begin
      $display("Correct M output  M=%h  expected= 36'h12c", M);
  end
   if (P!=48'h32) begin
      $display("ERROR - (verify DSP path 1 test) Failed - P output is wrong value P=%h  expected= 48'h32", P);
      $stop;
   end else if (PCOUT!=P) begin
      $display("ERROR - (verify DSP path 1 test) Failed - PCOUT output is NOT equal P output value    PCOUT=%h  expected=%h", PCOUT, P);
      $stop;
  end else begin
      $display("Correct P and PCOUT output values  P=%h   PCOUT=%h  expected= 48'h32", P, PCOUT);
  end
  if (CARRYOUT!=1'b0) begin
          $display("ERROR - (verify DSP path 1 test) Failed - CARRYOUT output is wrong value CARRYOUT=%b  expected= 1'b0", CARRYOUT);
          $stop;
  end else if (CARRYOUTF!=CARRYOUT) begin
          $display("ERROR - (verify DSP path 1 test) Failed - CARRYOUTF output is NOT equal CARRYOUT output value    CARRYOUTF=%b  expected= %b", CARRYOUTF, CARRYOUT);
  end else begin
          $display("Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=%b  CARRYOUTF=%b  expected= 1'b0", CARRYOUT, CARRYOUTF);
  end
       $display("Correct outputs - (Verify DSP Path 1 test) Passed - Outputs are correct after DSP path 1 operation");
```

Verify DSP Path 2:

```
//test 3: Verify DSP Path 2
  OPMODE=8'b00010000;
   A=18'd20; B=18'd10; C=48'd350; D=18'd25;
   BCIN=$random; PCIN=$random; CARRYIN=$random;
   repeat(3) @(negedge CLK); // Wait for 3 clock cycles
   if (BCOUT!=18'h23) begin
       $display("ERROR - (verify DSP path 2 test) Failed - BCOUT output is wrong value BCOUT=%h  expected= 18'h23", BCOUT);
       $stop;
   end else begin
       $display("Correct BCOUT output  BCOUT=%h  expected= 18'h23", BCOUT);
   end
    if (M!=36'h2bc) begin
       $display("ERROR - (verify DSP path 2 test) Failed - M output is wrong value M=%h  expected= 36'h2bc", M);
       $stop;
   end else begin
       $display("Correct M output  M=%h  expected= 36'h2bc", M);
   end
    if (P!=48'h0) begin
       $display("ERROR - (verify DSP path 2 test) Failed - P output is wrong value P=%h  expected= 48'h0", P);
       $stop;
    end else if (PCOUT!=P) begin
       $display("ERROR - (verify DSP path 2 test) Failed - PCOUT output is NOT equal P output value    PCOUT=%h  expected=%h", PCOUT, P);
       $stop;
   end else begin
       $display("Correct P and PCOUT output values  P=%h   PCOUT=%h  expected= 48'h0", P, PCOUT);
   end
   if (CARRYOUT!=1'b0) begin
           $display("ERROR - (verify DSP path 2 test) Failed - CARRYOUT output is wrong value CARRYOUT=%b  expected= 1'b0", CARRYOUT);
           $stop;
   end else if (CARRYOUTF!=CARRYOUT) begin
           $display("ERROR - (verify DSP path 2 test) Failed - CARRYOUTF output is NOT equal CARRYOUT output value    CARRYOUTF=%b  expected= %b", CARRYOUTF, CARRYOUT);
   end else begin
           $display("Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=%b  CARRYOUTF=%b  expected= 1'b0", CARRYOUT, CARRYOUTF);
   end
           $display("Correct outputs - (Verify DSP Path 2 test) Passed - Outputs are correct after DSP path 2 operation");
```

Verify DSP Path 3:

```
//test 4: Verify DSP Path 3
  OPMODE=8'b00001010;
   A=18'd20; B=18'd10; C=48'd350; D=18'd25;
   BCIN=$random; PCIN=$random; CARRYIN=$random;
  past_p_out = P; // Store the previous P output value
  past_carryout_out = CARRYOUT; // Store the previous CARRYOUT output value
  repeat(3) @(negedge CLK); // Wait for 3 clock cycles
   if (BCOUT!=18'ha) begin
       $display("ERROR - (verify DSP path 3 test) Failed - BCOUT output is wrong value BCOUT=%h  expected= 18'ha", BCOUT);
       $stop;
   end else begin
       $display("Correct BCOUT output  BCOUT=%h  expected= 18'ha", BCOUT);
   end
    if (M!=36'hc8) begin
       $display("ERROR - (verify DSP path 3 test) Failed - M output is wrong value M=%h  expected= 36'hc8", M);
       $stop;
   end else begin
       $display("Correct M output  M=%h  expected= 36'hc8", M);
   end
    if (P!=past_p_out) begin
       $display("ERROR - (verify DSP path 3 test) Failed - P output is wrong value P=%h  expected= %h", P, past_p_out);
       $stop;
    end else if (PCOUT!=P) begin
       $display("ERROR - (verify DSP path 3 test) Failed - PCOUT output is NOT equal P output value    PCOUT=%h  expected=%h", PCOUT, P);
       $stop;
   end else begin
       $display("Correct P and PCOUT output values  P=%h   PCOUT=%h  expected=%h", P, PCOUT, past_p_out);
   end
   if (CARRYOUT!=past_carryout_out) begin
           $display("ERROR - (verify DSP path 3 test) Failed - CARRYOUT output is wrong value CARRYOUT=%b  expected= %b", CARRYOUT, past_carryout_out);
           $stop;
   end else if (CARRYOUTF!=CARRYOUT) begin
           $display("ERROR - (verify DSP path 3 test) Failed - CARRYOUTF output is NOT equal CARRYOUT output value    CARRYOUTF=%b  expected= %b", CARRYOUTF, CARRYOUT);
   end else begin
           $display("Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=%b  CARRYOUTF=%b  expected= %b", CARRYOUT, CARRYOUTF, past_carryout_out);
   end
           $display("Correct outputs - (Verify DSP Path 3 test) Passed - Outputs are correct after DSP path 3 operation");
```

Verify DSP Path 4:

```
//test 5: Verify DSP Path 4
   OPMODE=8'b10100111;
    A=18'd5; B=18'd6; C=48'd350; D=18'd25;
    BCIN=$random; PCIN=48'd3000; CARRYIN=$random;
    repeat(3) @(negedge CLK); // Wait for 3 clock cycles
    if (BCOUT!=18'h6) begin
        $display("ERROR - (verify DSP path 4 test) Failed - BCOUT output is wrong value BCOUT=%h  expected= 18'h6", BCOUT);
        $stop;
    end else begin
        $display("Correct BCOUT output  BCOUT=%h  expected= 18'h6", BCOUT);
    end
     if (M!=36'h1e) begin
        $display("ERROR - (verify DSP path 4 test) Failed - M output is wrong value M=%h  expected= 36'h1e", M);
        $stop;
    end else begin
        $display("Correct M output  M=%h  expected= 36'h1e", M);
    end
     if (P!=48'hfe6fffec0bb1) begin
        $display("ERROR - (verify DSP path 4 test) Failed - P output is wrong value P=%h  expected= %h", P, 48'hfe6fffec0bb1);
        $stop;
     end else if (PCOUT!=P) begin
        $display("ERROR - (verify DSP path 4 test) Failed - PCOUT output is NOT equal P output value    PCOUT=%h  expected=%h", PCOUT, P);
        $stop;
    end else begin
        $display("Correct P and PCOUT output values  P=%h   PCOUT=%h  expected=%h", P, PCOUT, 48'hfe6fffec0bb1);
    end
    if (CARRYOUT!=1'b1) begin
            $display("ERROR - (verify DSP path 4 test) Failed - CARRYOUT output is wrong value CARRYOUT=%b  expected= %b", CARRYOUT, 1'b1);
            $stop;
    end else if (CARRYOUTF!=CARRYOUT) begin
            $display("ERROR - (verify DSP path 4 test) Failed - CARRYOUTF output is NOT equal CARRYOUT output value    CARRYOUTF=%b  expected= %b", CARRYOUTF, CARRYOUT);
    end else begin
            $display("Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=%b  CARRYOUTF=%b  expected= %b", CARRYOUT, CARRYOUTF, 1'b1);
    end
            $display("Correct outputs - (Verify DSP Path 4 test) Passed - Outputs are correct after DSP path 4 operation");
            $display("ALL tests are passed");
            $stop;
end

endmodule
```

3-DO file:

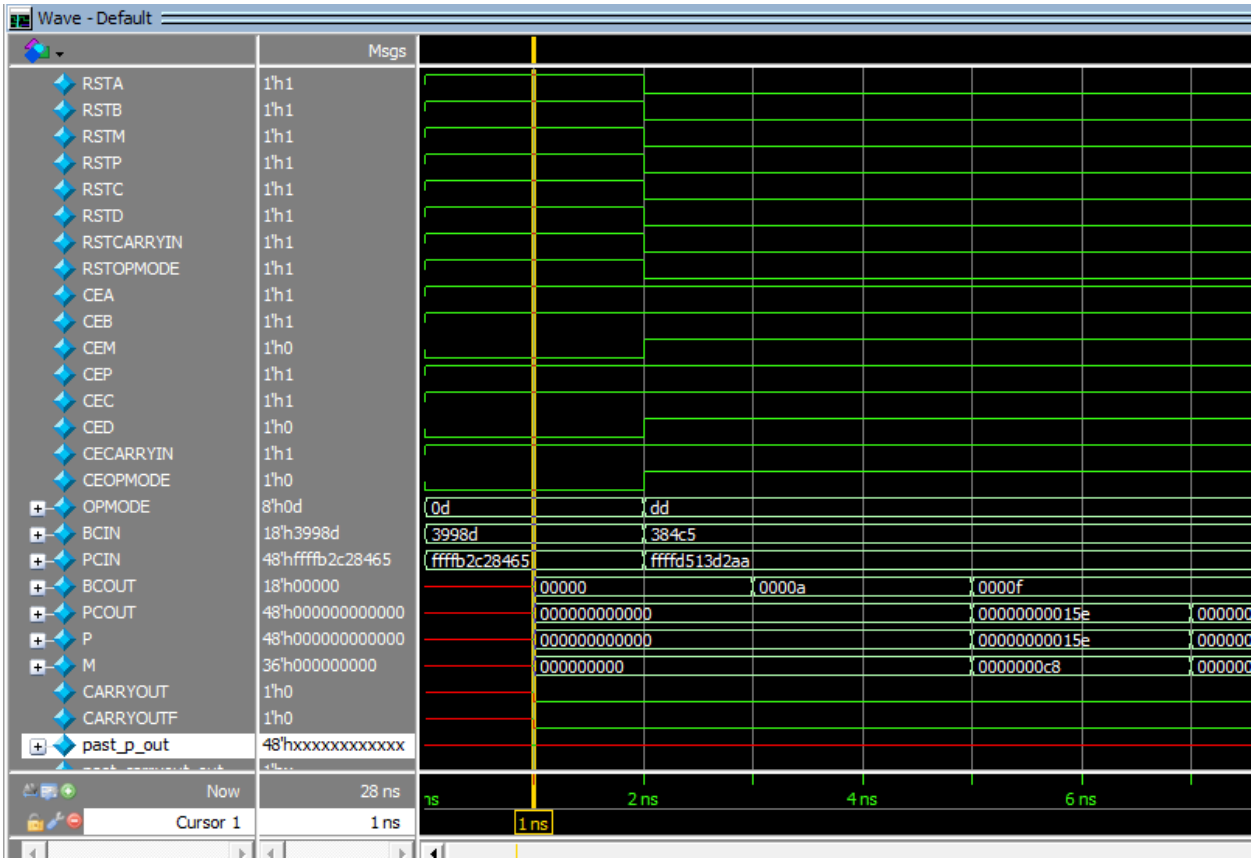
run.do

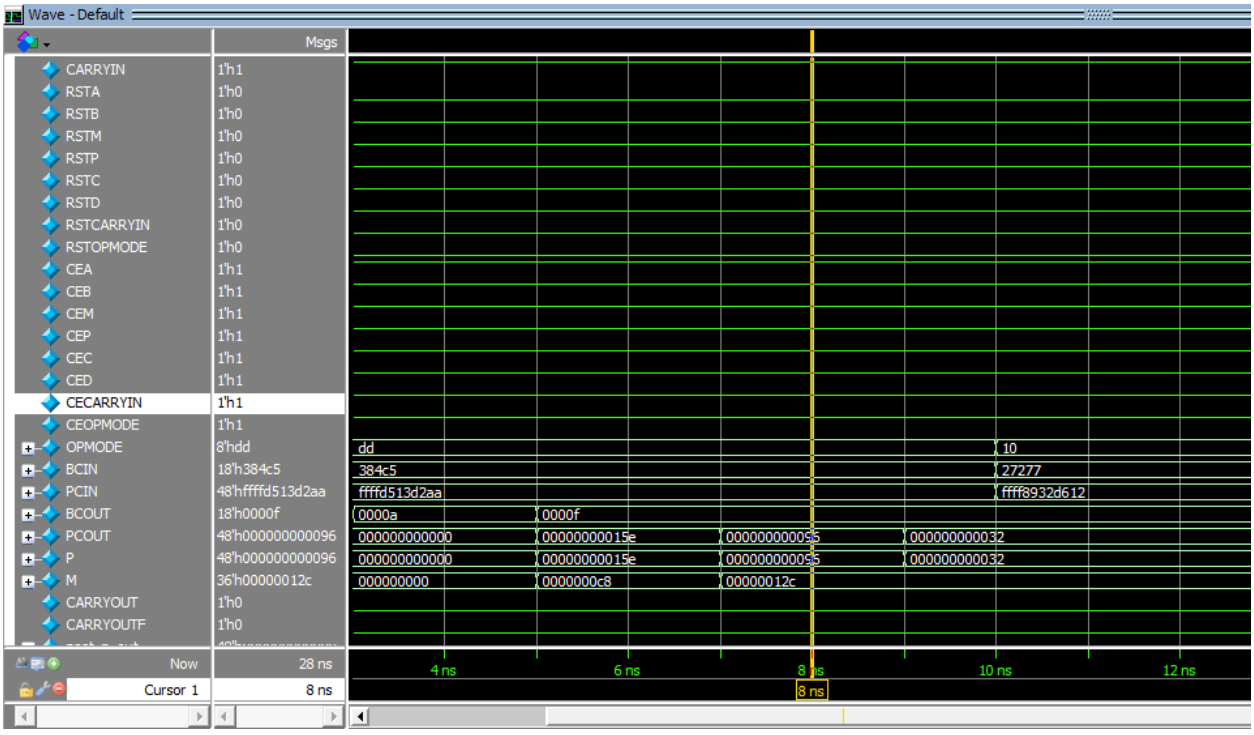
```
1   vlib work
2   vlog  reg_mux_input.v DSP48A1_TOP.V DSP48A1_tb.v
3   vsim -voptargs=+acc work.DSP48A1_tb
4   add wave *
5   run -all
```

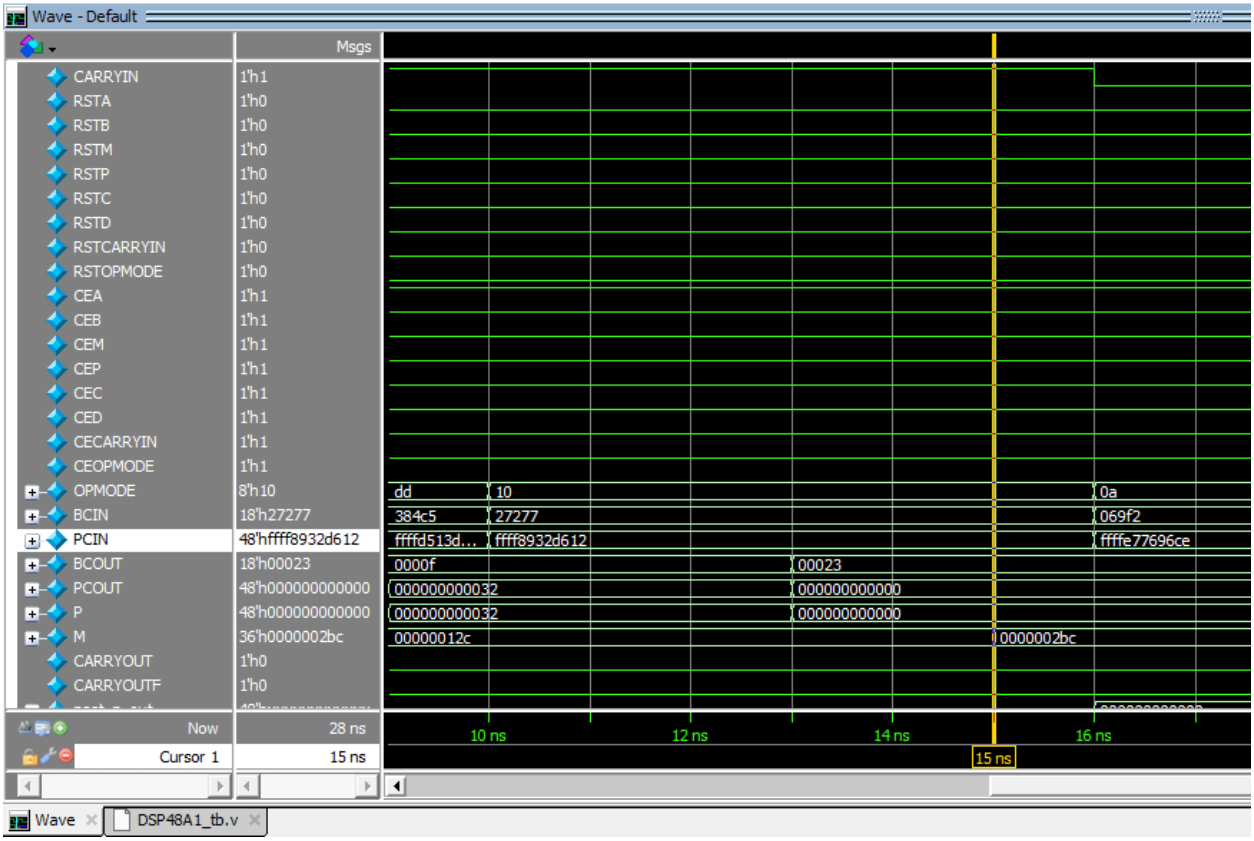
```
do run.do
```

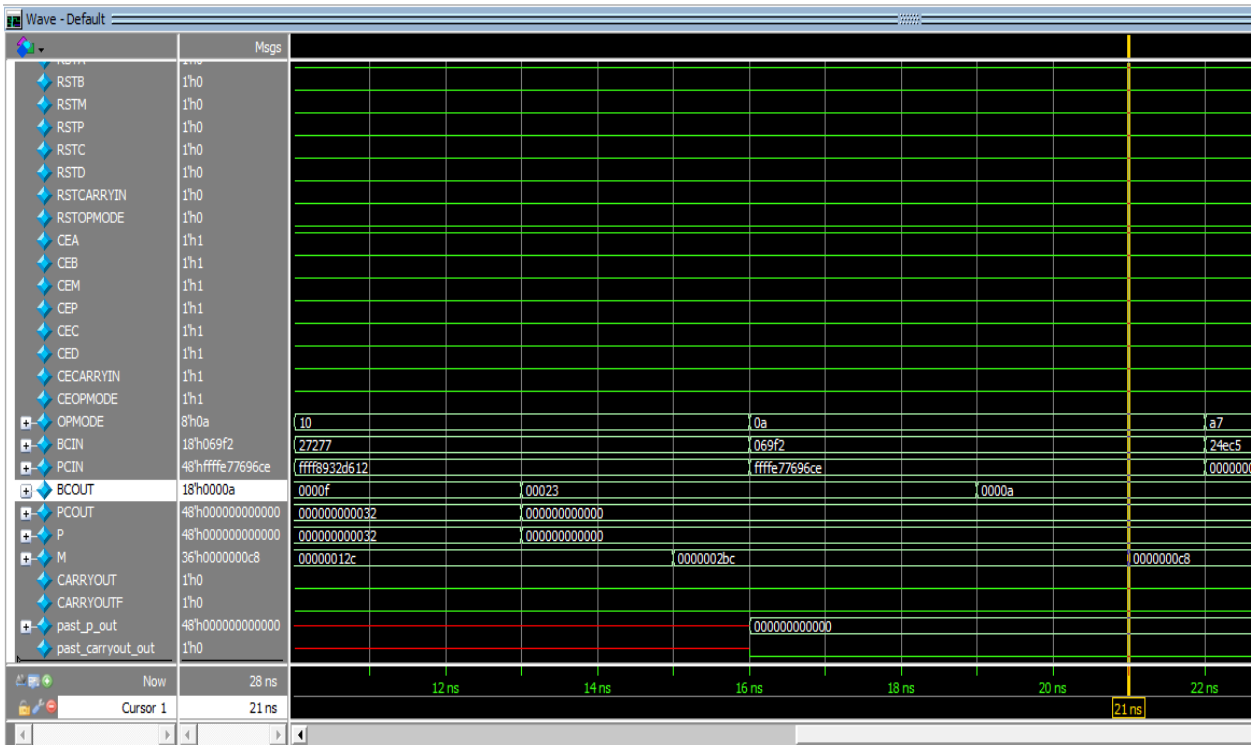
```
VSIM 8> do run.do
```

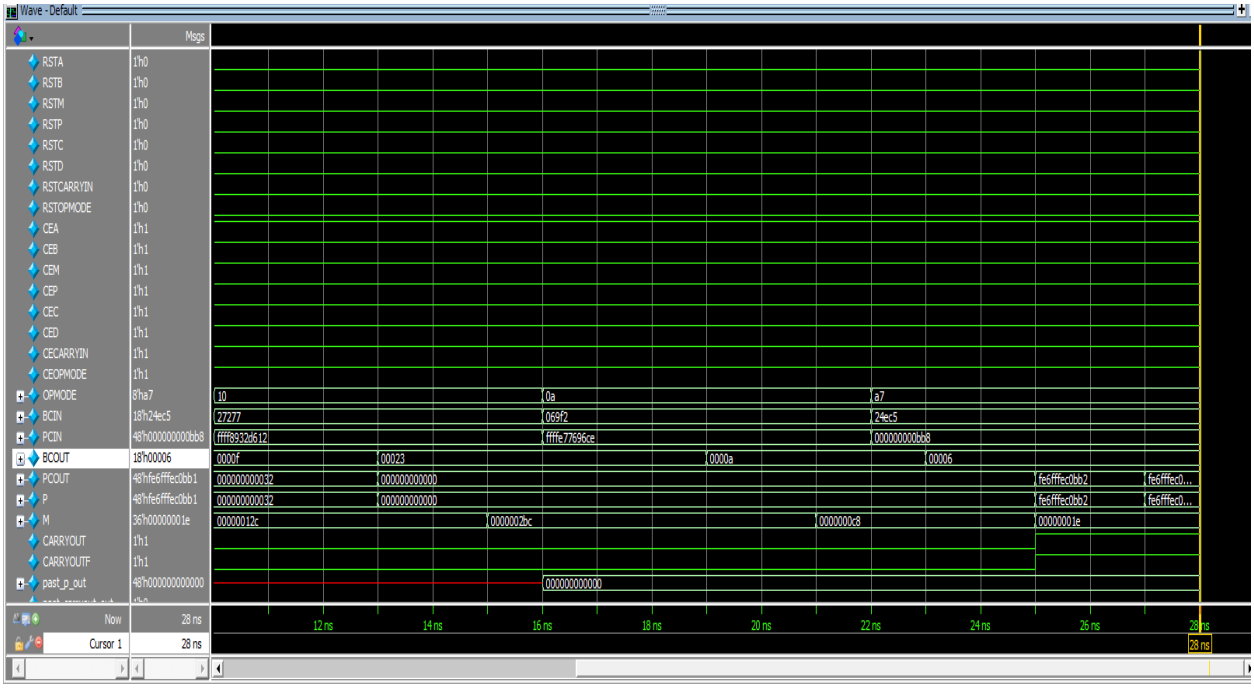
4- QuestaSim Snippets:






Wave - Default
Msgs
CARRYIN 1'h1
RSTA 1'h0
RSTB 1'h0
RSTM 1'h0
RSTP 1'h0
RSTC 1'h0
RSTD 1'h0
RSTCARRYIN 1'h0
RSTOPMODE 1'h0
CEA 1'h1
CEB 1'h1
CEM 1'h1
CEP 1'h1
CEC 1'h1
CED 1'h1
CECARRYIN 1'h1
CEOPMODE 1'h1
OPMODE 8'h10 dd 10 0a
BCIN 18'h27277 384c5 27277 069f2
PCIN 48'hffff8932d612 ffffd513d... ffff8932d612 ffffe77696ce
BCOUT 18'h00023 0000f 00023
PCOUT 48'h000000000000 000000000032 000000000000
P 48'h000000000000 000000000032 000000000000
M 36'h0000002bc 00000012c 0000002bc
CARRYOUT 1'h0
CARRYOUTF 1'h0
Now 28 ns
Cursor 1 15 ns
10 ns 12 ns 14 ns 16 ns
15 ns
Wave
DSP48A1_tb.v


Wave - Default
Msgs
RSTB 1'h0
RSTM 1'h0
RSTP 1'h0
RSTC 1'h0
RSTD 1'h0
RSTCARRYIN 1'h0
RSTOPMODE 1'h0
CEA 1'h1
CEB 1'h1
CEM 1'h1
CEP 1'h1
CEC 1'h1
CED 1'h1
CECARRYIN 1'h1
CEOPMODE 1'h1
OPMODE 8'h0a 10 0a a7
BCIN 18'h069f2 27277 069f2 24ec5
PCIN 48'hffffe77696ce ffff8932d612 ffffe77696ce 0000000
BCOUT 18'h0000a 0000f 00023 0000a
PCOUT 48'h000000000000 000000000032 000000000000
P 48'h000000000000 000000000032 000000000000
M 36'h0000000c8 00000012c 0000002bc 0000000c8
CARRYOUT 1'h0
CARRYOUTF 1'h0
past_p_out 48'h000000000000 000000000000
past_carryout_out 1'h0
Now 28 ns
Cursor 1 21 ns
12 ns 14 ns 16 ns 18 ns 20 ns 22 ns
21 ns

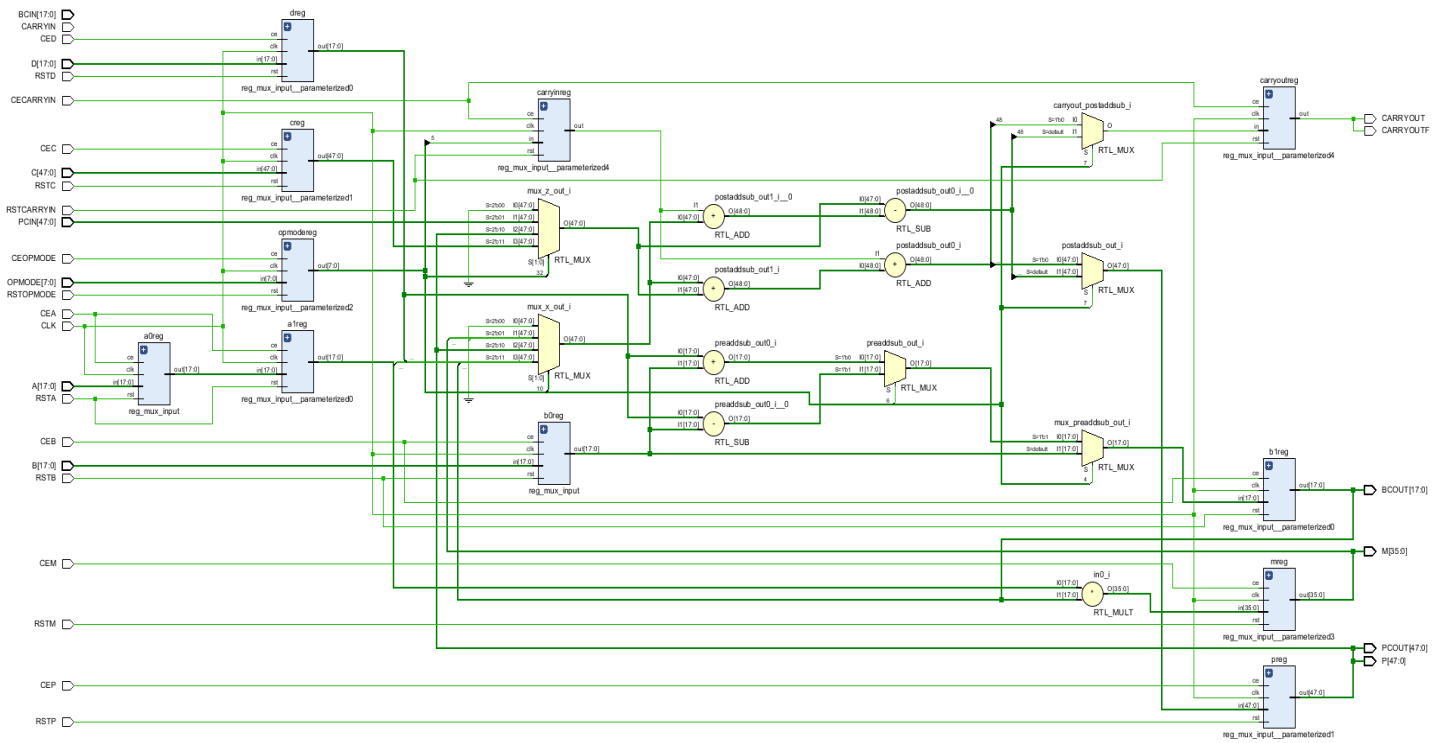


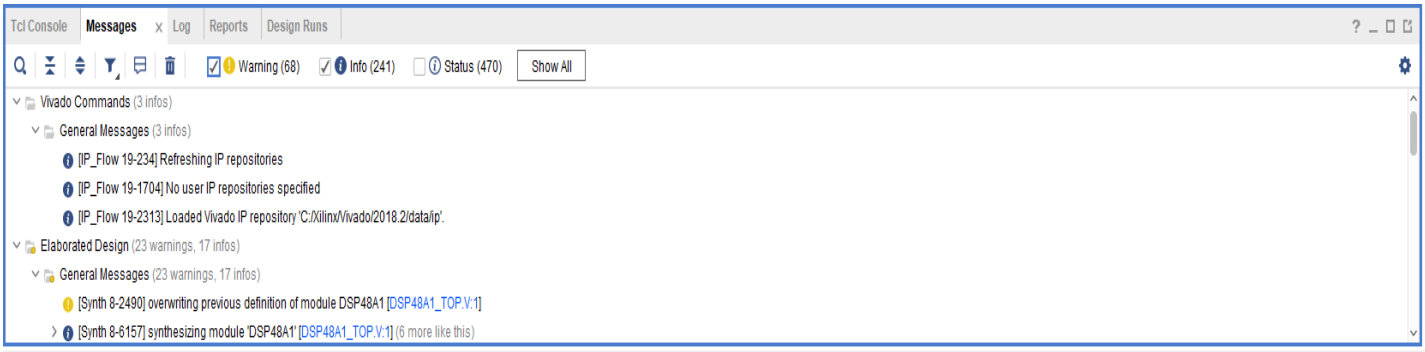

```
QuestaSim> do run.do
# ** Warning: (vlib-34) Library already exists at "work".
# Errors: 0, Warnings: 1
# QuestaSim-64 vlog 2021.1 Compiler 2021.01 Jan 19 2021
# Start time: 22:31:33 on Aug 03,2025
# vlog -reportprogress 300 reg_mux_input.v DSP48A1_TOP.V DSP48A1_tb.v
# -- Compiling module reg_mux_input
# -- Compiling module DSP48A1
# -- Compiling module DSP48A1_tb
#
# Top level modules:
#       DSP48A1_tb
# End time: 22:31:33 on Aug 03,2025, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
# vsim -voptargs="+acc" work.DSP48A1_tb
# Start time: 22:31:33 on Aug 03,2025
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# Loading work.DSP48A1_tb(fast)
# Loading work.DSP48A1(fast)
# Loading work.reg_mux_input(fast)
# Loading work.reg_mux_input(fast__1)
# Loading work.reg_mux_input(fast__2)
# Loading work.reg_mux_input(fast__3)
# Loading work.reg_mux_input(fast__4)
# Loading work.reg_mux_input(fast__5)
# Correct outputs - (Verify Reset Operation test) Passed - Outputs are zero after reset
# Correct BCOUT output  BCOUT=0000f  expected= 18'hf
# Correct M output  M=00000012c  expected= 36'h12c
# Correct P and PCOUT output values  P=000000000032   PCOUT=000000000032  expected= 48'h32
# Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=0  CARRYOUTF=0  expected= 1'b0
# Correct outputs - (Verify DSP Path 1 test) Passed - Outputs are correct after DSP path 1 operation
# Correct BCOUT output  BCOUT=00023  expected= 18'h23
# Correct M output  M=0000002bc  expected= 36'h2bc
# Correct P and PCOUT output values  P=000000000000   PCOUT=000000000000  expected= 48'h0
# Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=0  CARRYOUTF=0  expected= 1'b0
# Correct outputs - (Verify DSP Path 2 test) Passed - Outputs are correct after DSP path 2 operation
# Correct BCOUT output  BCOUT=0000a  expected= 18'ha
# Correct M output  M=0000000c8  expected= 36'hc8
# Correct P and PCOUT output values  P=000000000000   PCOUT=000000000000  expected=000000000000
# Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=0  CARRYOUTF=0  expected= 0
# Correct outputs - (Verify DSP Path 3 test) Passed - Outputs are correct after DSP path 3 operation
# Correct BCOUT output  BCOUT=00006  expected= 18'h6
# Correct M output  M=00000001e  expected= 36'hle
# Correct P and PCOUT output values  P=fe6fffec0bb1   PCOUT=fe6fffec0bb1  expected=fe6fffec0bb1
# Correct CARRYOUT and CARRYOUTF output values  CARRYOUT=1  CARRYOUTF=1  expected= 1
# Correct outputs - (Verify DSP Path 4 test) Passed - Outputs are correct after DSP path 4 operation
# ALL tests are passed
# ** Note: $stop    : DSP48A1_tb.v(220)
#    Time: 28 ns  Iteration: 1  Instance: /DSP48A1_tb
# Break in Module DSP48A1_tb at DSP48A1_tb.v line 220
```

5-Constraint File :

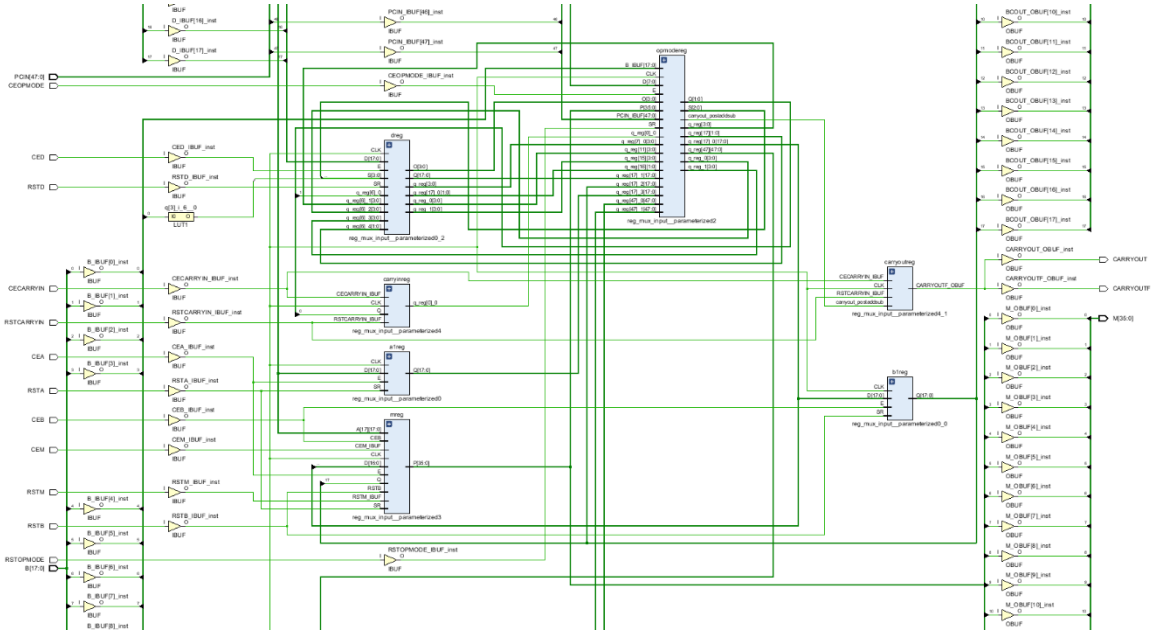
```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports CLK]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [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]}]
```

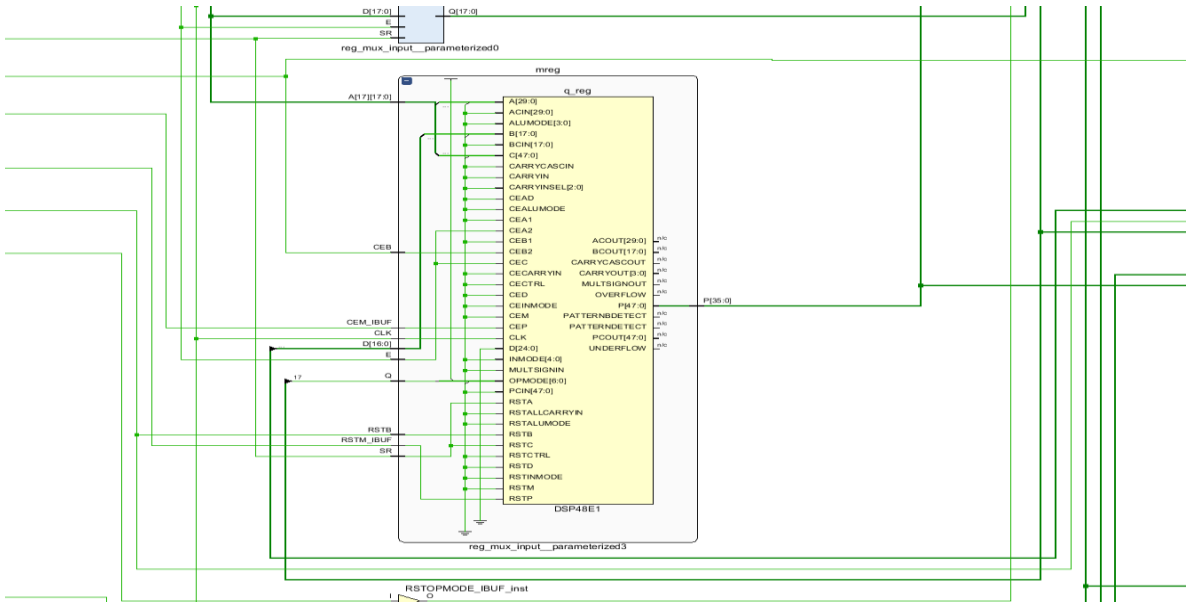
6-Elaboration (“Messages” tab & Schematic snippets) :

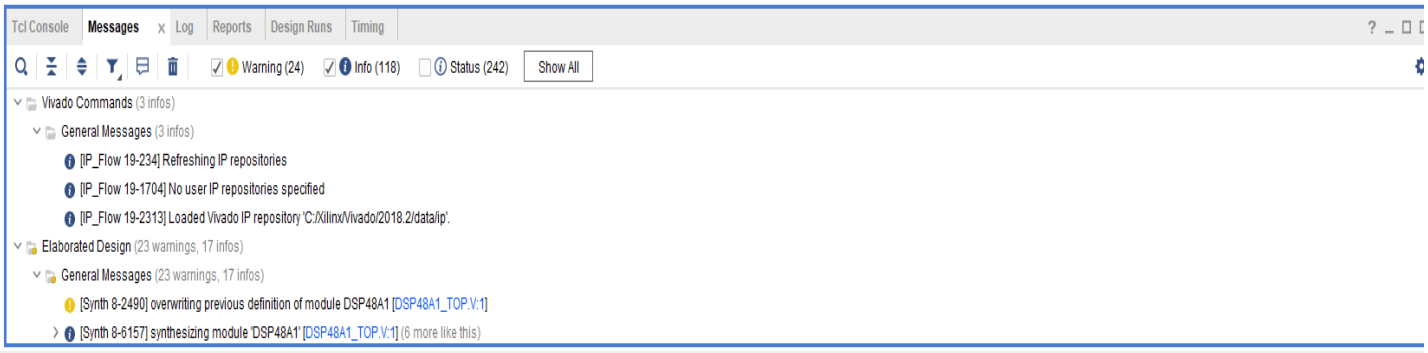


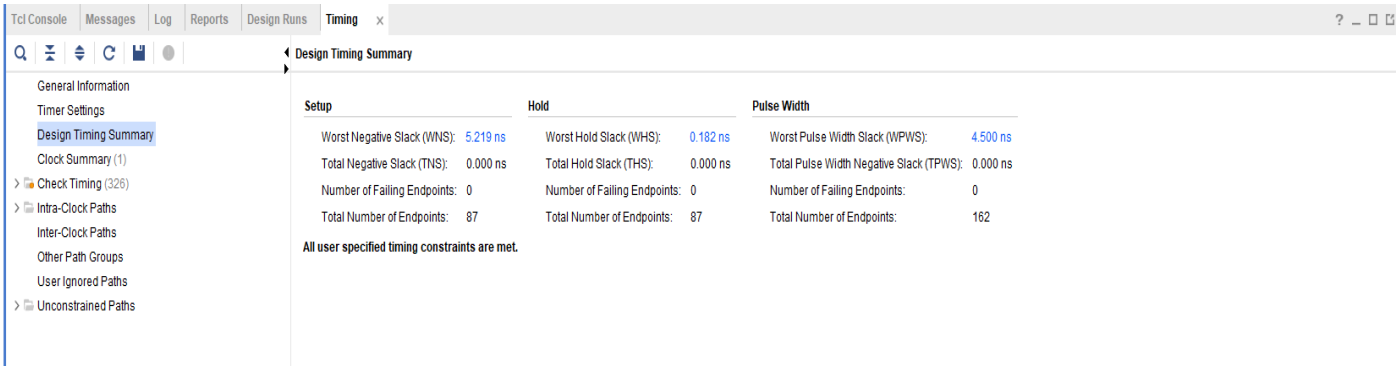


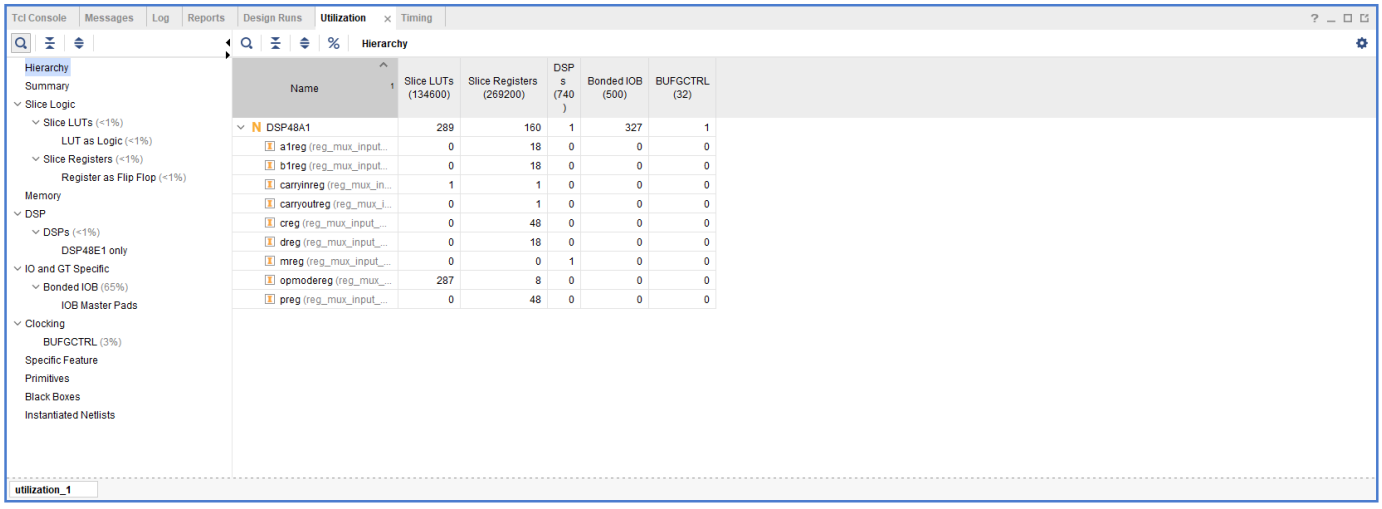
7-Synthesis (“Messages” tab, Utilization report, timing report & Schematic snippets) :



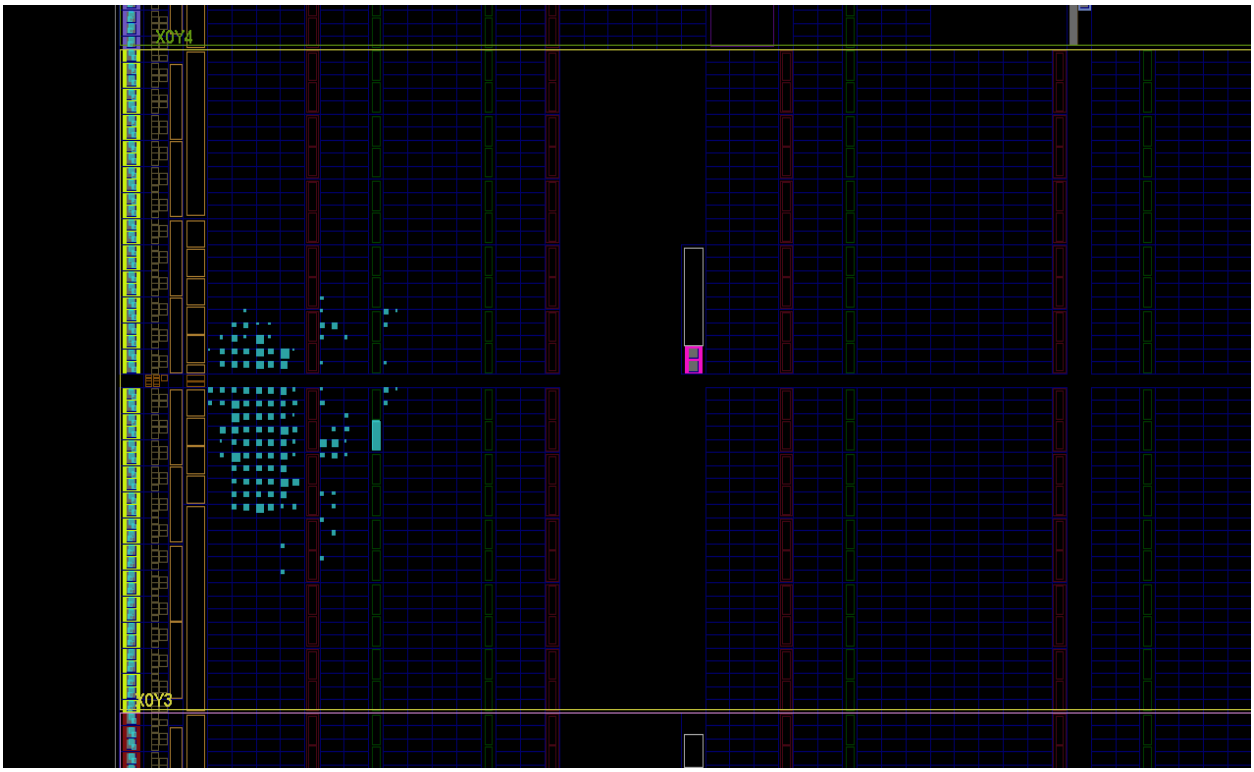


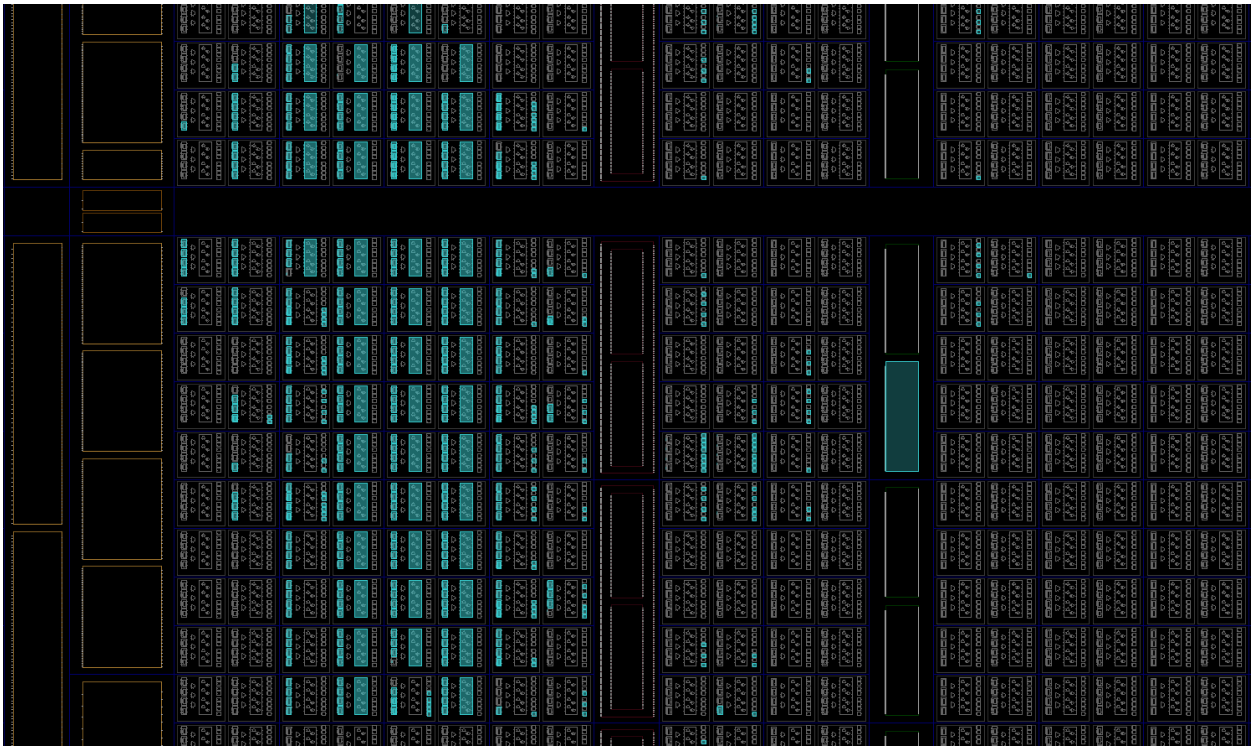

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Warning (24)
Info (118)
Status (242)
Show All
Vivado Commands (3 infos)
General Messages (3 infos)
[IP_Flow 19-234] Refreshing IP repositories
[IP_Flow 19-1704] No user IP repositories specified
[IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2018.2/data/ip'.
Elaborated Design (23 warnings, 17 infos)
General Messages (23 warnings, 17 infos)
[Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1_TOP.V:1]
[Synth 8-6157] synthesizing module 'DSP48A1' [DSP48A1_TOP.V:1] (6 more like this)

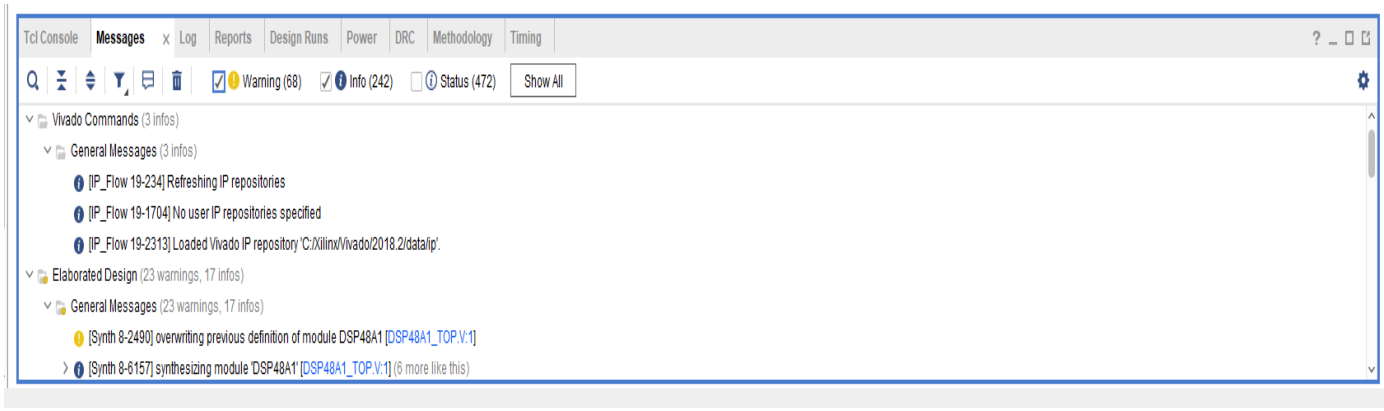

Tcl Console
Messages
Log
Reports
Design Runs
Timing
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (326)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths
Setup
Worst Negative Slack (WNS): 5.219 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.

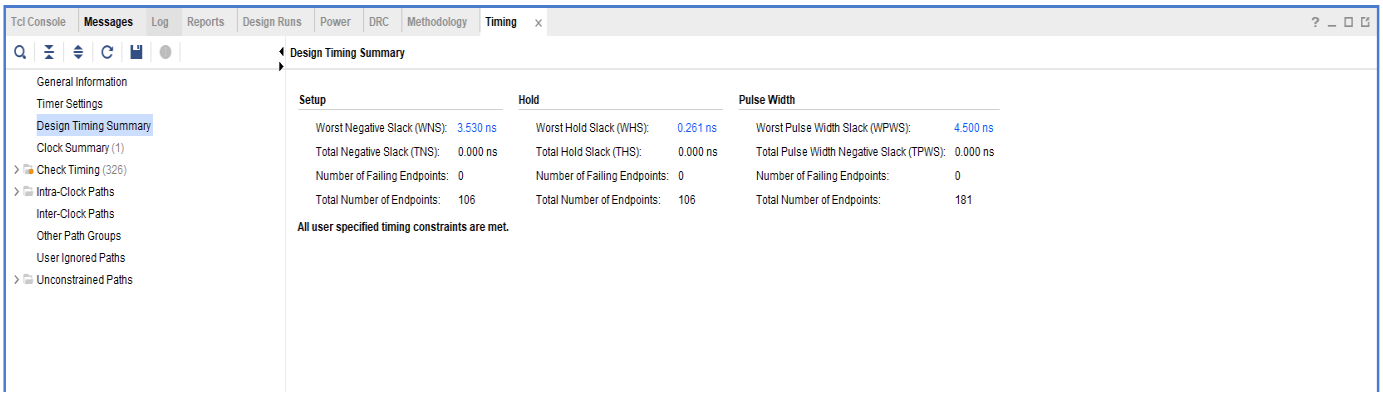

Tcl Console
Messages
Log
Reports
Design Runs
Utilization
Timing
Hierarchy
Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Flip Flop (<1%)
Memory
DSP
DSPs (<1%)
DSP48E1 only
IO and GT Specific
Bonded IOB (65%)
IOB Master Pads
Clocking
BUFGCTRL (3%)
Specific Feature
Primitives
Black Boxes
Instantiated Netlists
Name
Slice LUTs (134600)
Slice Registers (269200)
DSPs (740)
Bonded IOB (500)
BUFGCTRL (32)
DSP48A1 289 160 1 327 1
a1reg (reg_mux_input... 0 18 0 0 0
b1reg (reg_mux_input... 0 18 0 0 0
carryinreg (reg_mux_in... 1 1 0 0 0
carryoutreg (reg_mux_i... 0 1 0 0 0
creg (reg_mux_input_... 0 48 0 0 0
dreg (reg_mux_input_... 0 18 0 0 0
mreg (reg_mux_input_... 0 0 1 0 0
opmodereg (reg_mux_... 287 8 0 0 0
preg (reg_mux_input_... 0 48 0 0 0
utilization_1

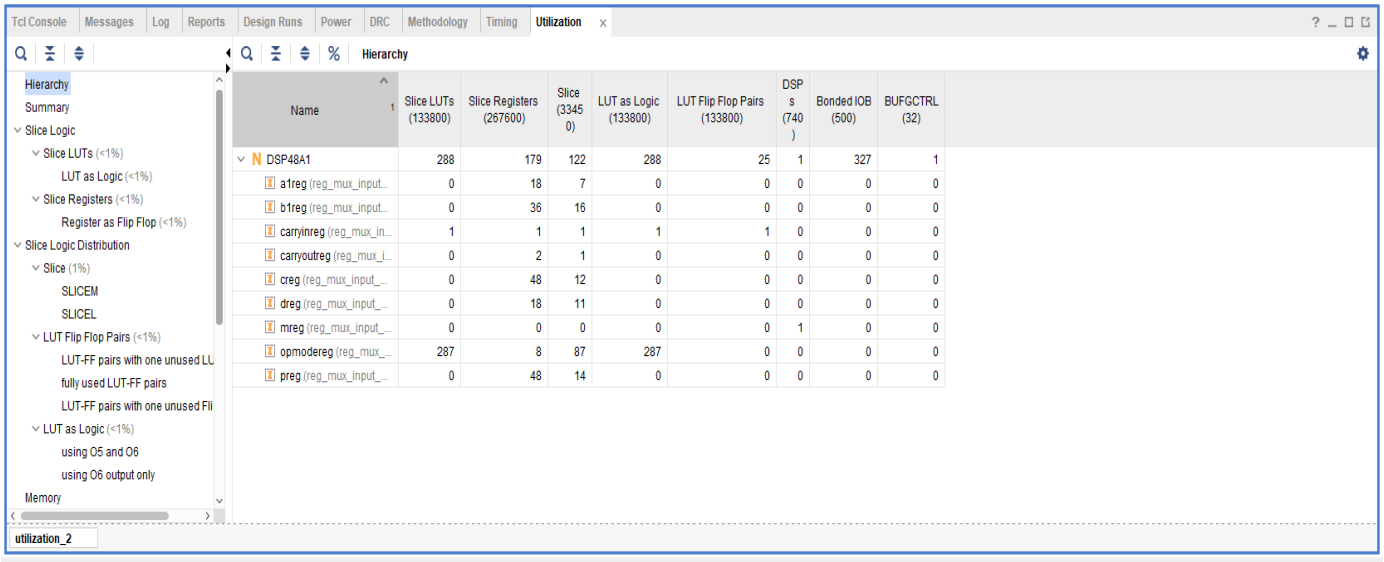
8-Implementation (“Messages” tab, Utilization report, timing report & device snippets) :






Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Warning (68)
Info (242)
Status (472)
Show All
Vivado Commands (3 infos)
General Messages (3 infos)
[IP_Flow 19-234] Refreshing IP repositories
[IP_Flow 19-1704] No user IP repositories specified
[IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2018.2/data/ip'.
Elaborated Design (23 warnings, 17 infos)
General Messages (23 warnings, 17 infos)
[Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1_TOP.V:1]
[Synth 8-6157] synthesizing module 'DSP48A1' [DSP48A1_TOP.V:1] (6 more like this)


Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (326)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths
Design Timing Summary
Setup
Worst Negative Slack (WNS): 3.530 ns
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 106
Hold
Worst Hold Slack (WHS): 0.261 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.


Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Utilization
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 unused LU
fully used LUT-FF pairs
LUT-FF pairs with one unused Fli
LUT as Logic (<1%)
using O5 and O6
using O6 output only
Memory
Hierarchy
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 288 179 122 288 25 1 327 1
a1reg (reg_mux_input... 0 18 7 0 0 0 0 0
b1reg (reg_mux_input... 0 36 16 0 0 0 0 0
carryinreg (reg_mux_in... 1 1 1 1 1 0 0 0
carryoutreg (reg_mux_i... 0 2 1 0 0 0 0 0
creg (reg_mux_input_... 0 48 12 0 0 0 0 0
dreg (reg_mux_input_... 0 18 11 0 0 0 0 0
mreg (reg_mux_input_... 0 0 0 0 0 1 0 0
opmodereg (reg_mux_... 287 8 87 287 0 0 0 0
preg (reg_mux_input_... 0 48 14 0 0 0 0 0
utilization_2