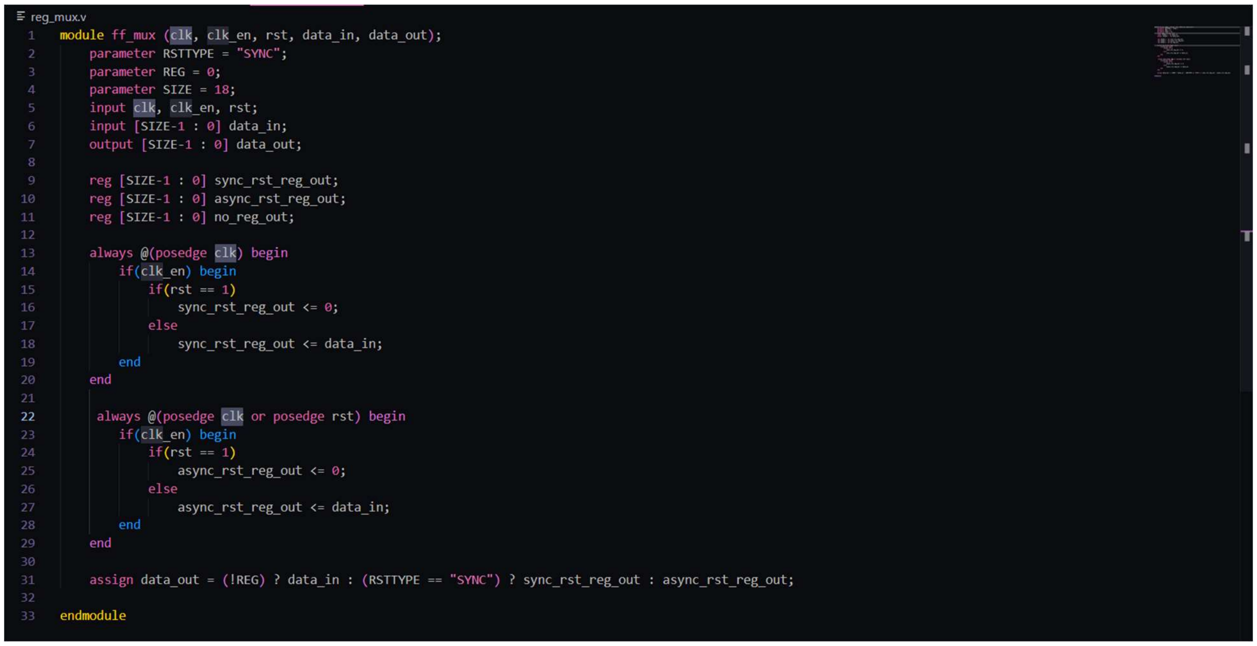
Project1: Spartan6 - DSP48A1

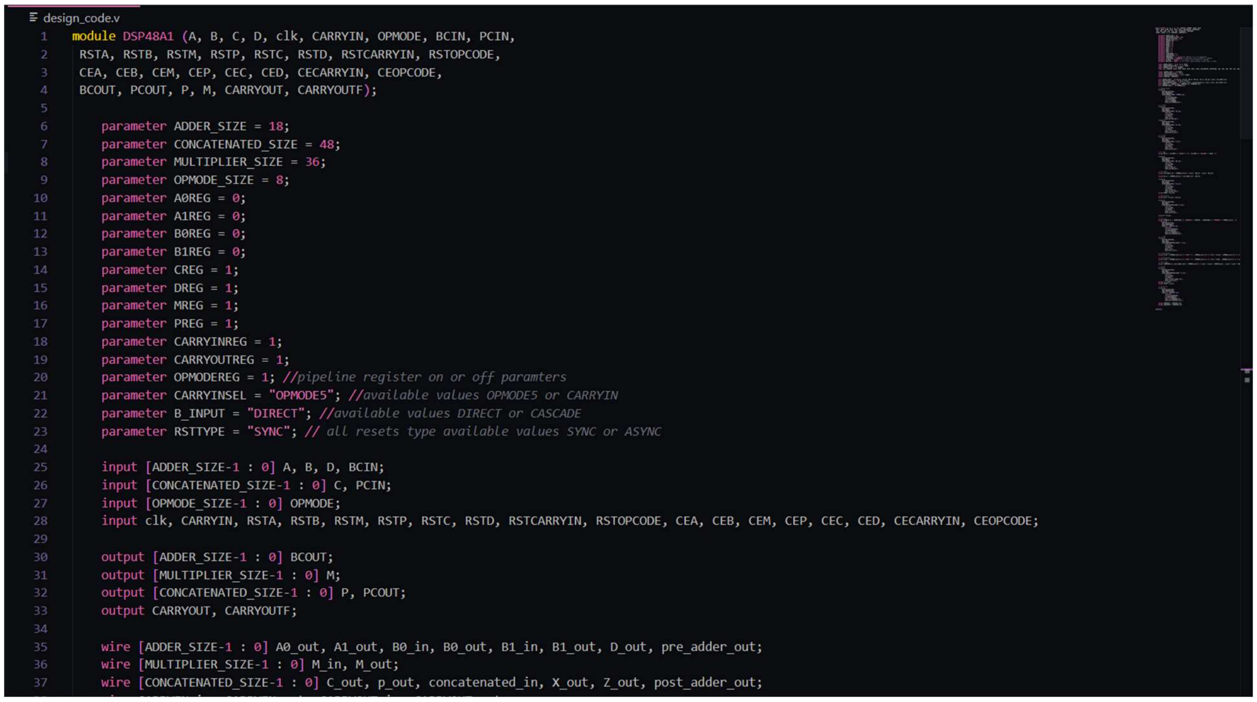
Malak Waleed Fouad

RTL code:



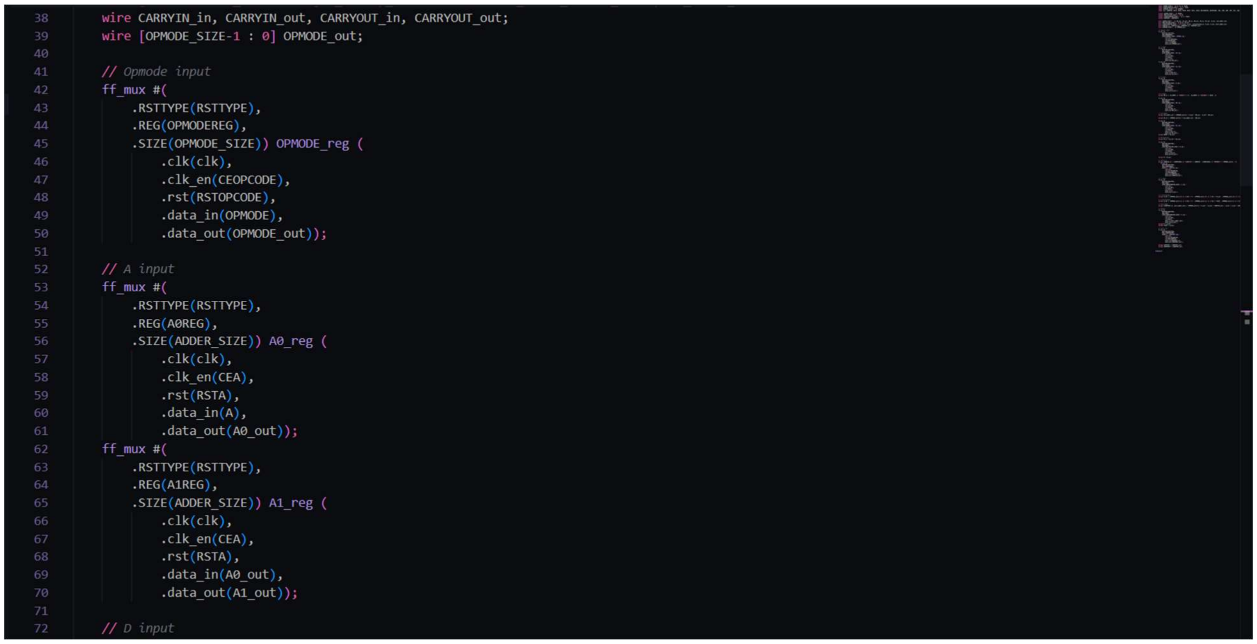
reg_mux.v

```verilog
module ff_mux (clk, clk_en, rst, data_in, data_out);
    parameter RSTTYPE = "SYNC";
    parameter REG = 0;
    parameter SIZE = 18;
    input clk, clk_en, rst;
    input [SIZE-1 : 0] data_in;
    output [SIZE-1 : 0] data_out;

    reg [SIZE-1 : 0] sync_rst_reg_out;
    reg [SIZE-1 : 0] async_rst_reg_out;
    reg [SIZE-1 : 0] no_reg_out;

    always @(posedge clk) begin
        if(clk_en) begin
            if(rst == 1)
                sync_rst_reg_out <= 0;
            else
                sync_rst_reg_out <= data_in;
        end
    end

     always @(posedge clk or posedge rst) begin
        if(clk_en) begin
            if(rst == 1)
                async_rst_reg_out <= 0;
            else
                async_rst_reg_out <= data_in;
        end
    end

    assign data_out = (!REG) ? data_in : (RSTTYPE == "SYNC") ? sync_rst_reg_out : async_rst_reg_out;

endmodule
```

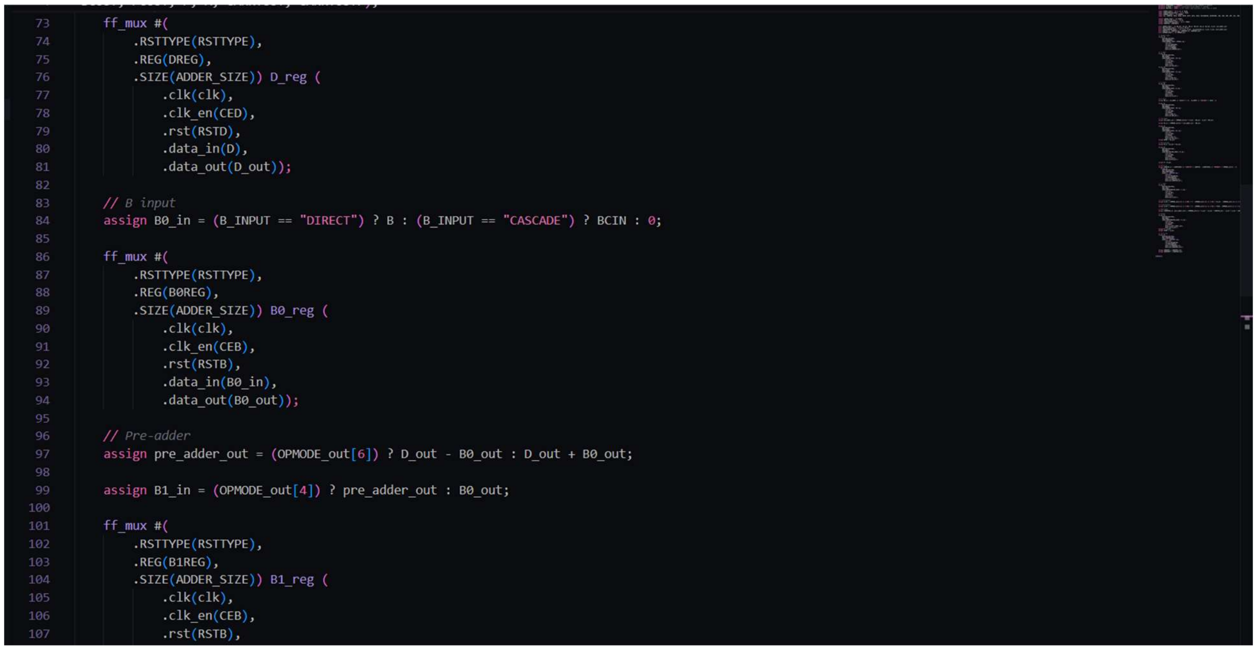


design_code.v

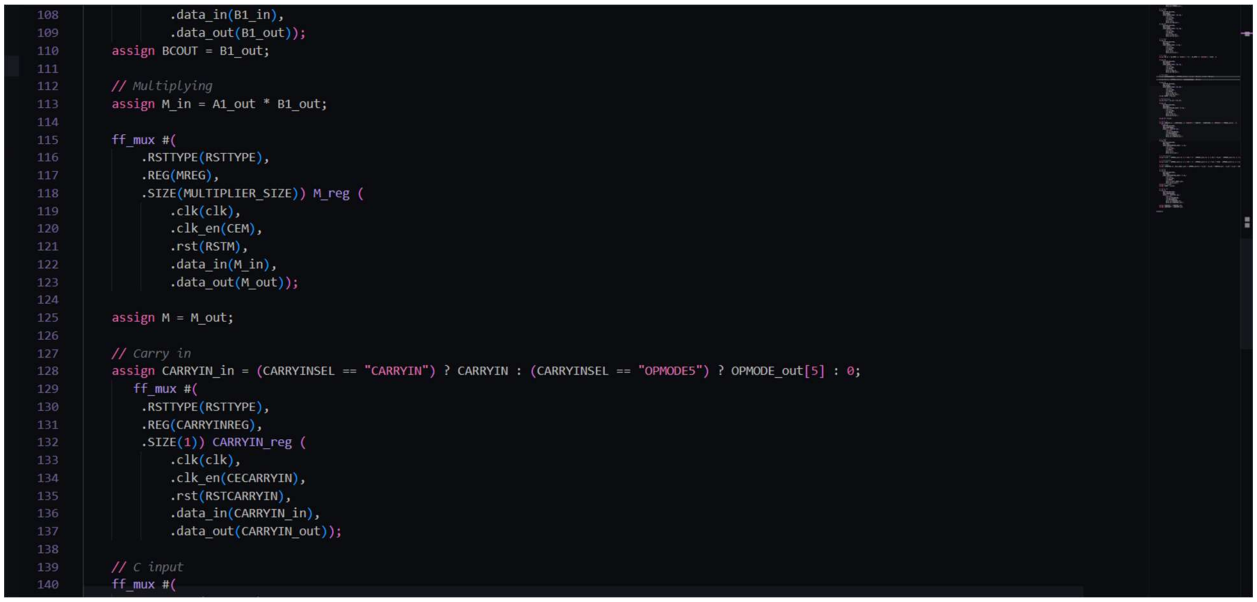
```verilog
module DSP48A1 (A, B, C, D, clk, CARRYIN, OPMODE, BCIN, PCIN,
 RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPCODE,
 CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPCODE,
 BCOUT, PCOUT, P, M, CARRYOUT, CARRYOUTF);

    parameter ADDER_SIZE = 18;
    parameter CONCATENATED_SIZE = 48;
    parameter MULTIPLIER_SIZE = 36;
    parameter OPMODE_SIZE = 8;
    parameter A0REG = 0;
    parameter A1REG = 0;
    parameter B0REG = 0;
    parameter B1REG = 0;
    parameter CREG = 1;
    parameter DREG = 1;
    parameter MREG = 1;
    parameter PREG = 1;
    parameter CARRYINREG = 1;
    parameter CARRYOUTREG = 1;
    parameter OPMODEREG = 1; //pipeline register on or off paramters
    parameter CARRYINSEL = "OPMODE5"; //available values OPMODE5 or CARRYIN
    parameter B_INPUT = "DIRECT"; //available values DIRECT or CASCADE
    parameter RSTTYPE = "SYNC"; // all resets type available values SYNC or ASYNC

    input [ADDER_SIZE-1 : 0] A, B, D, BCIN;
    input [CONCATENATED_SIZE-1 : 0] C, PCIN;
    input [OPMODE_SIZE-1 : 0] OPMODE;
    input clk, CARRYIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPCODE, CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPCODE;

    output [ADDER_SIZE-1 : 0] BCOUT;
    output [MULTIPLIER_SIZE-1 : 0] M;
    output [CONCATENATED_SIZE-1 : 0] P, PCOUT;
    output CARRYOUT, CARRYOUTF;

    wire [ADDER_SIZE-1 : 0] A0_out, A1_out, B0_in, B0_out, B1_in, B1_out, D_out, pre_adder_out;
    wire [MULTIPLIER_SIZE-1 : 0] M_in, M_out;
    wire [CONCATENATED_SIZE-1 : 0] C_out, p_out, concatenated_in, X_out, Z_out, post_adder_out;
```



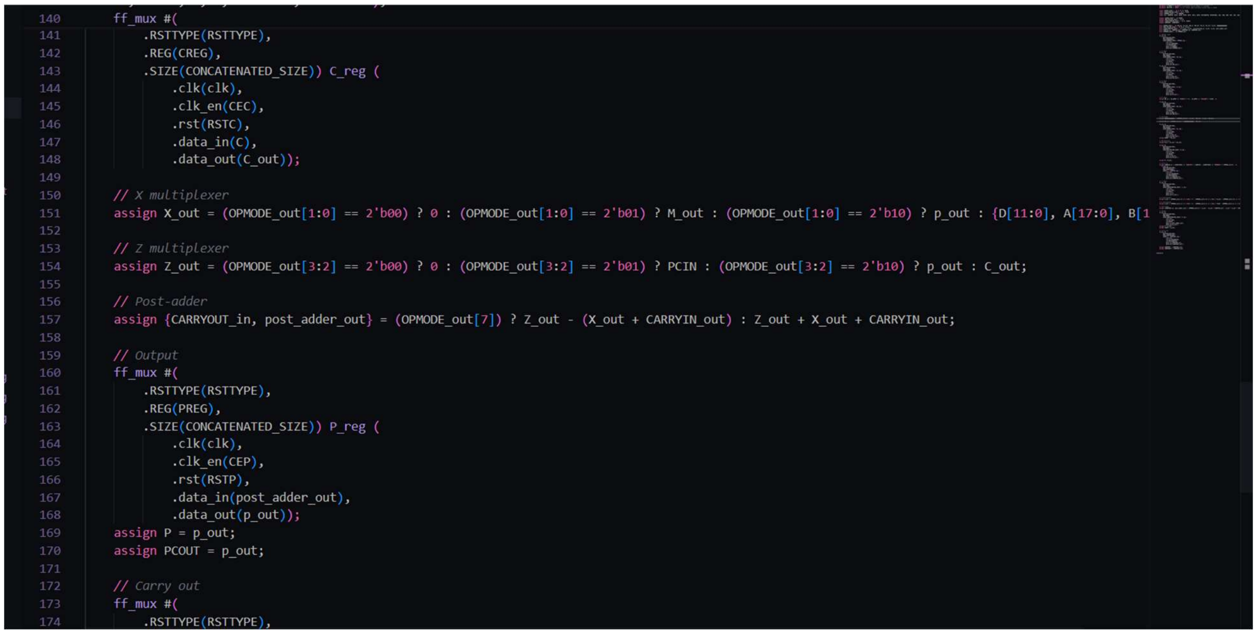
```verilog
    wire CARRYIN_in, CARRYIN_out, CARRYOUT_in, CARRYOUT_out;
    wire [OPMODE_SIZE-1 : 0] OPMODE_out;

    // Opmode input
    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(OPMODEREG),
        .SIZE(OPMODE_SIZE)) OPMODE_reg (
            .clk(clk),
            .clk_en(CEOPCODE),
            .rst(RSTOPCODE),
            .data_in(OPMODE),
            .data_out(OPMODE_out));

    // A input
    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(A0REG),
        .SIZE(ADDER_SIZE)) A0_reg (
            .clk(clk),
            .clk_en(CEA),
            .rst(RSTA),
            .data_in(A),
            .data_out(A0_out));
    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(A1REG),
        .SIZE(ADDER_SIZE)) A1_reg (
            .clk(clk),
            .clk_en(CEA),
            .rst(RSTA),
            .data_in(A0_out),
            .data_out(A1_out));

    // D input
```



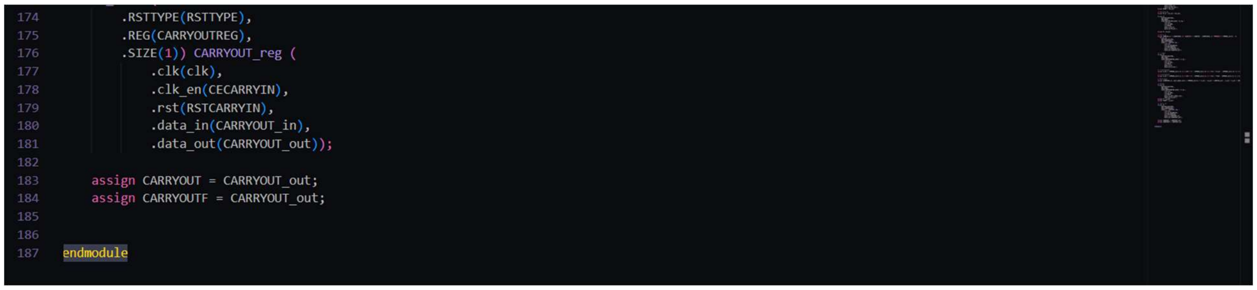
```verilog
    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(DREG),
        .SIZE(ADDER_SIZE)) D_reg (
            .clk(clk),
            .clk_en(CED),
            .rst(RSTD),
            .data_in(D),
            .data_out(D_out));

    // B input
    assign B0_in = (B_INPUT == "DIRECT") ? B : (B_INPUT == "CASCADE") ? BCIN : 0;

    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(B0REG),
        .SIZE(ADDER_SIZE)) B0_reg (
            .clk(clk),
            .clk_en(CEB),
            .rst(RSTB),
            .data_in(B0_in),
            .data_out(B0_out));

    // Pre-adder
    assign pre_adder_out = (OPMODE_out[6]) ? D_out - B0_out : D_out + B0_out;

    assign B1_in = (OPMODE_out[4]) ? pre_adder_out : B0_out;

    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(B1REG),
        .SIZE(ADDER_SIZE)) B1_reg (
            .clk(clk),
            .clk_en(CEB),
            .rst(RSTB),
```



```
            .data_in(B1_in),
            .data_out(B1_out));
    assign BCOUT = B1_out;

    // Multiplying
    assign M_in = A1_out * B1_out;

    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(MREG),
        .SIZE(MULTIPLIER_SIZE)) M_reg (
            .clk(clk),
            .clk_en(CEM),
            .rst(RSTM),
            .data_in(M_in),
            .data_out(M_out));

    assign M = M_out;

    // Carry in
    assign CARRYIN_in = (CARRYINSEL == "CARRYIN") ? CARRYIN : (CARRYINSEL == "OPMODE5") ? OPMODE_out[5] : 0;
       ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(CARRYINREG),
        .SIZE(1)) CARRYIN_reg (
            .clk(clk),
            .clk_en(CECARRYIN),
            .rst(RSTCARRYIN),
            .data_in(CARRYIN_in),
            .data_out(CARRYIN_out));

    // C input
    ff_mux #(
```

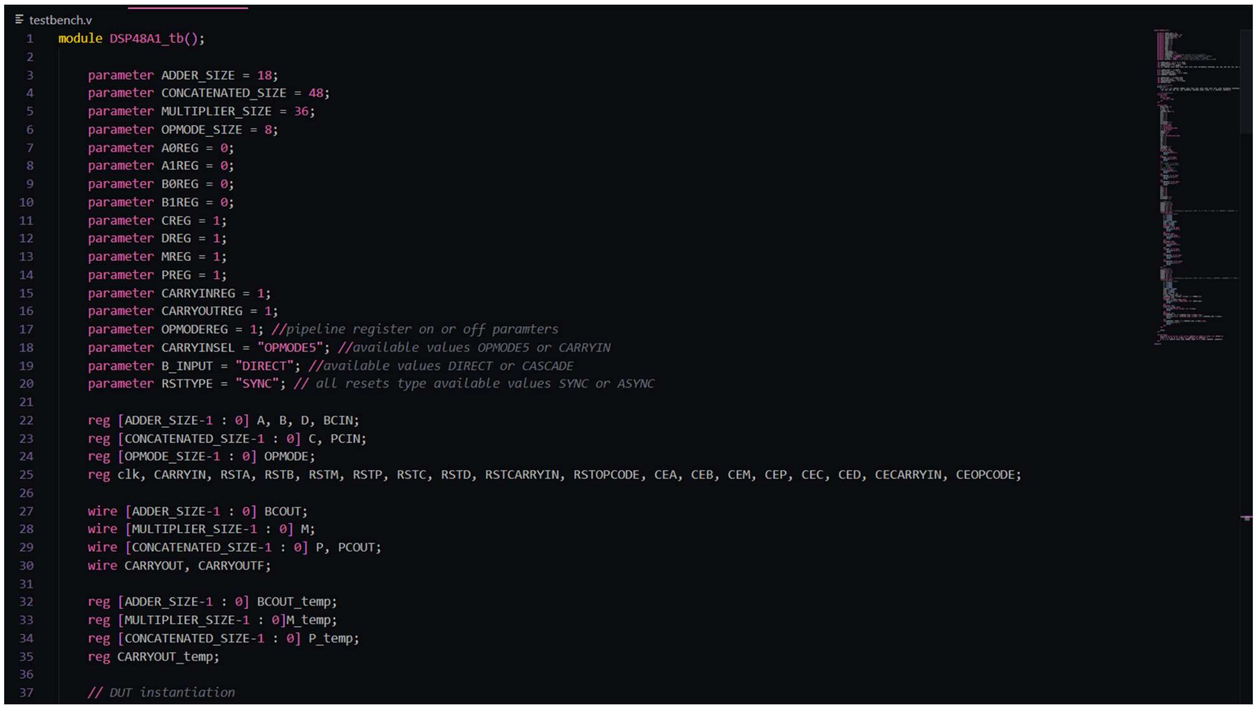


```
    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(CREG),
        .SIZE(CONCATENATED_SIZE)) C_reg (
            .clk(clk),
            .clk_en(CEC),
            .rst(RSTC),
            .data_in(C),
            .data_out(C_out));

    // X multiplexer
    assign X_out = (OPMODE_out[1:0] == 2'b00) ? 0 : (OPMODE_out[1:0] == 2'b01) ? M_out : (OPMODE_out[1:0] == 2'b10) ? p_out : {D[11:0], A[17:0], B[1

    // Z multiplexer
    assign Z_out = (OPMODE_out[3:2] == 2'b00) ? 0 : (OPMODE_out[3:2] == 2'b01) ? PCIN : (OPMODE_out[3:2] == 2'b10) ? p_out : C_out;

    // Post-adder
    assign {CARRYOUT_in, post_adder_out} = (OPMODE_out[7]) ? Z_out - (X_out + CARRYIN_out) : Z_out + X_out + CARRYIN_out;

    // Output
    ff_mux #(
        .RSTTYPE(RSTTYPE),
        .REG(PREG),
        .SIZE(CONCATENATED_SIZE)) P_reg (
            .clk(clk),
            .clk_en(CEP),
            .rst(RSTP),
            .data_in(post_adder_out),
            .data_out(p_out));
    assign P = p_out;
    assign PCOUT = p_out;

    // Carry out
    ff_mux #(
        .RSTTYPE(RSTTYPE),
```



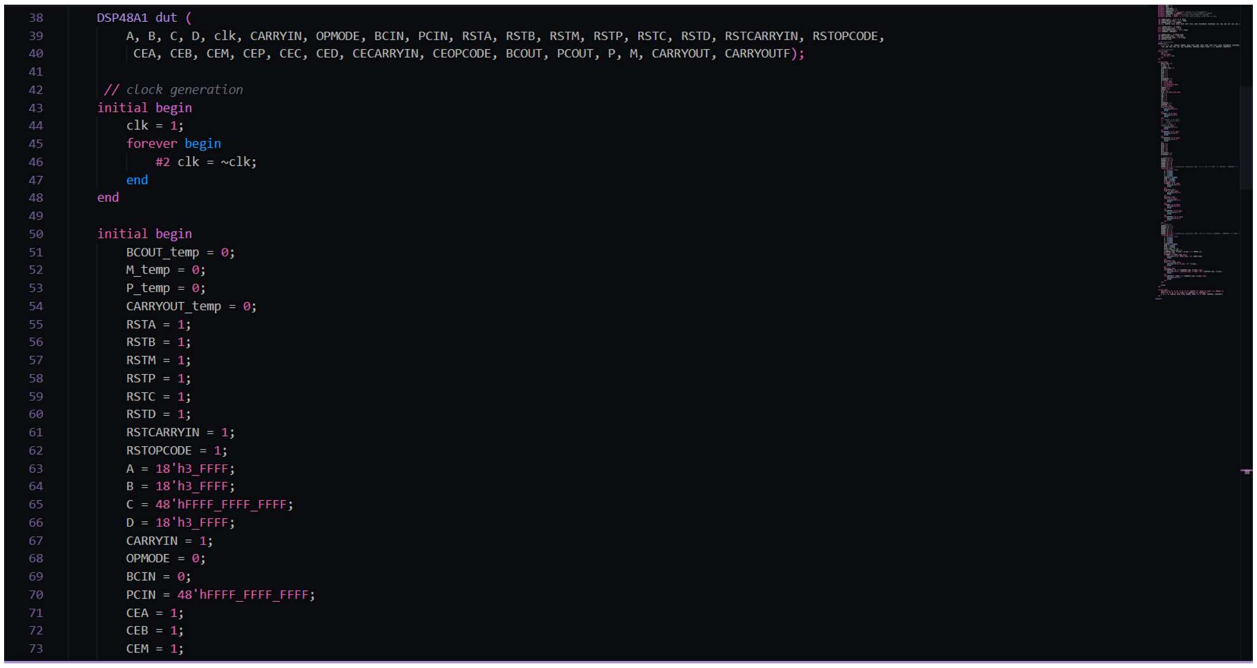
```
        .RSTTYPE(RSTTYPE),
        .REG(CARRYOUTREG),
        .SIZE(1)) CARRYOUT_reg (
            .clk(clk),
            .clk_en(CECARRYIN),
            .rst(RSTCARRYIN),
            .data_in(CARRYOUT_in),
            .data_out(CARRYOUT_out));

    assign CARRYOUT = CARRYOUT_out;
    assign CARRYOUTF = CARRYOUT_out;


endmodule
```

Testbench code:

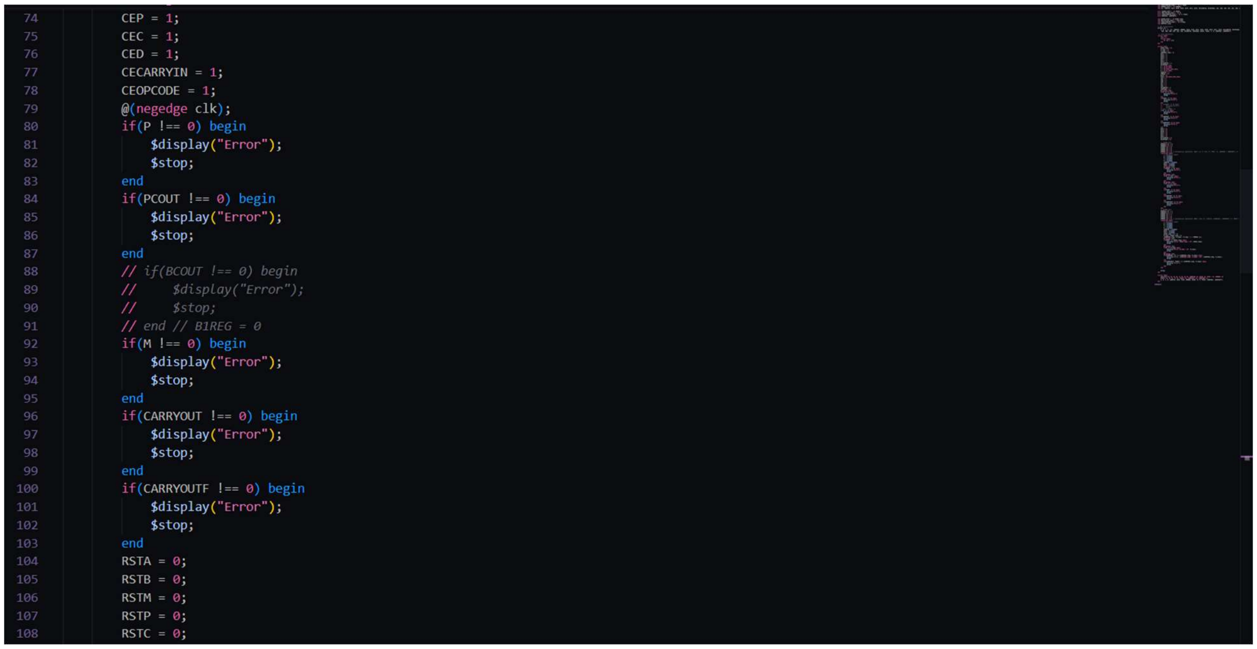


testbench.v

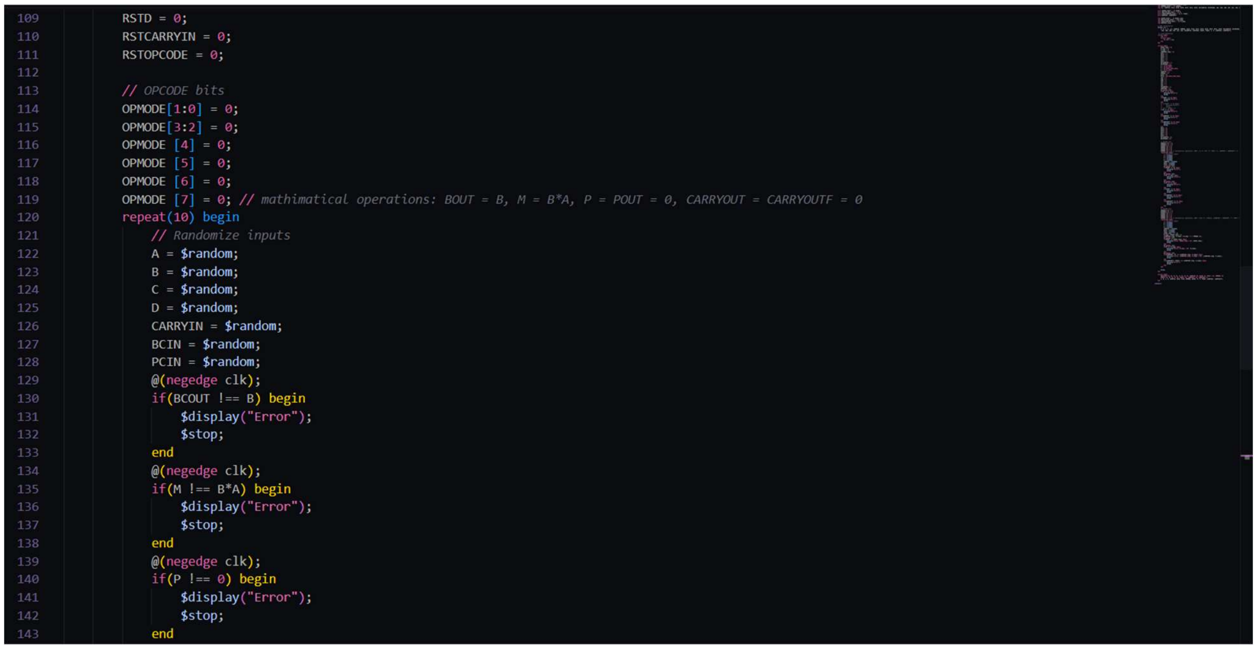
```verilog
module DSP48A1_tb();

    parameter ADDER_SIZE = 18;
    parameter CONCATENATED_SIZE = 48;
    parameter MULTIPLIER_SIZE = 36;
    parameter OPMODE_SIZE = 8;
    parameter A0REG = 0;
    parameter A1REG = 0;
    parameter B0REG = 0;
    parameter B1REG = 0;
    parameter CREG = 1;
    parameter DREG = 1;
    parameter MREG = 1;
    parameter PREG = 1;
    parameter CARRYINREG = 1;
    parameter CARRYOUTREG = 1;
    parameter OPMODEREG = 1; //pipeline register on or off paramters
    parameter CARRYINSEL = "OPMODE5"; //available values OPMODE5 or CARRYIN
    parameter B_INPUT = "DIRECT"; //available values DIRECT or CASCADE
    parameter RSTTYPE = "SYNC"; // all resets type available values SYNC or ASYNC

    reg [ADDER_SIZE-1 : 0] A, B, D, BCIN;
    reg [CONCATENATED_SIZE-1 : 0] C, PCIN;
    reg [OPMODE_SIZE-1 : 0] OPMODE;
    reg clk, CARRYIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPCODE, CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPCODE;

    wire [ADDER_SIZE-1 : 0] BCOUT;
    wire [MULTIPLIER_SIZE-1 : 0] M;
    wire [CONCATENATED_SIZE-1 : 0] P, PCOUT;
    wire CARRYOUT, CARRYOUTF;

    reg [ADDER_SIZE-1 : 0] BCOUT_temp;
    reg [MULTIPLIER_SIZE-1 : 0]M_temp;
    reg [CONCATENATED_SIZE-1 : 0] P_temp;
    reg CARRYOUT_temp;

    // DUT instantiation
```



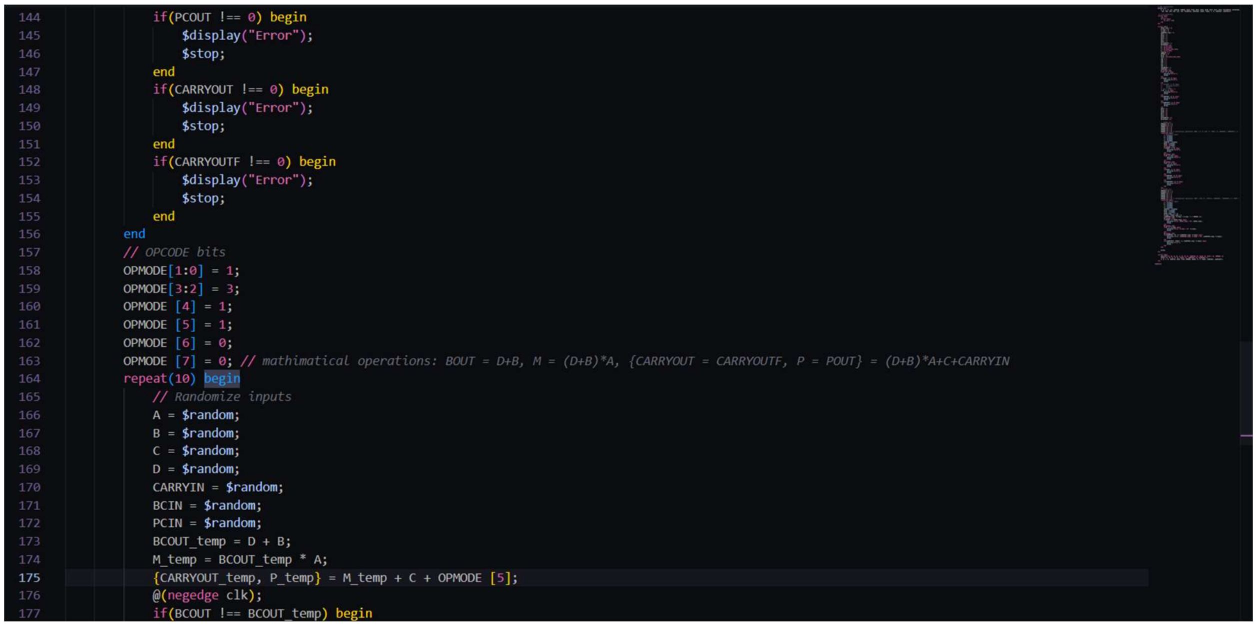
```verilog
    DSP48A1 dut (
        A, B, C, D, clk, CARRYIN, OPMODE, BCIN, PCIN, RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPCODE,
         CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPCODE, BCOUT, PCOUT, P, M, CARRYOUT, CARRYOUTF);

     // clock generation
    initial begin
        clk = 1;
        forever begin
            #2 clk = ~clk;
        end
    end

    initial begin
        BCOUT_temp = 0;
        M_temp = 0;
        P_temp = 0;
        CARRYOUT_temp = 0;
        RSTA = 1;
        RSTB = 1;
        RSTM = 1;
        RSTP = 1;
        RSTC = 1;
        RSTD = 1;
        RSTCARRYIN = 1;
        RSTOPCODE = 1;
        A = 18'h3_FFFF;
        B = 18'h3_FFFF;
        C = 48'hFFFF_FFFF_FFFF;
        D = 18'h3_FFFF;
        CARRYIN = 1;
        OPMODE = 0;
        BCIN = 0;
        PCIN = 48'hFFFF_FFFF_FFFF;
        CEA = 1;
        CEB = 1;
        CEM = 1;
```



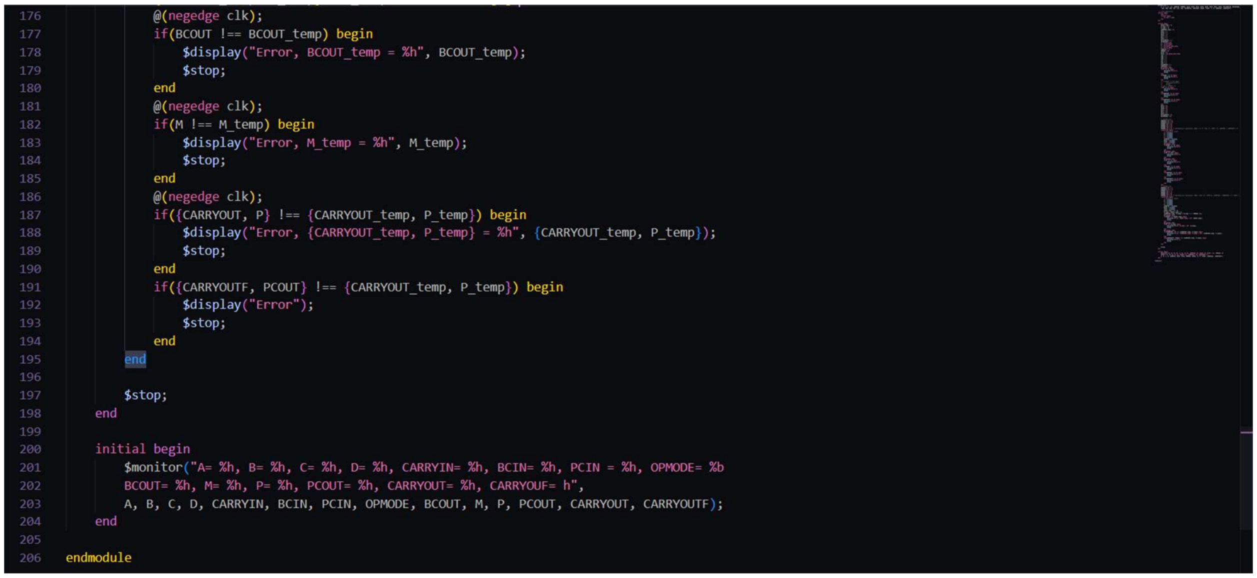
```
        CEP = 1;
        CEC = 1;
        CED = 1;
        CECARRYIN = 1;
        CEOPCODE = 1;
        @(negedge clk);
        if(P !== 0) begin
            $display("Error");
            $stop;
        end
        if(PCOUT !== 0) begin
            $display("Error");
            $stop;
        end
        // if(BCOUT !== 0) begin
        //     $display("Error");
        //     $stop;
        // end // B1REG = 0
        if(M !== 0) begin
            $display("Error");
            $stop;
        end
        if(CARRYOUT !== 0) begin
            $display("Error");
            $stop;
        end
        if(CARRYOUTF !== 0) begin
            $display("Error");
            $stop;
        end
        RSTA = 0;
        RSTB = 0;
        RSTM = 0;
        RSTP = 0;
        RSTC = 0;
```



```
        RSTD = 0;
        RSTCARRYIN = 0;
        RSTOPCODE = 0;

        // OPCODE bits
        OPMODE[1:0] = 0;
        OPMODE[3:2] = 0;
        OPMODE [4] = 0;
        OPMODE [5] = 0;
        OPMODE [6] = 0;
        OPMODE [7] = 0; // mathimatical operations: BOUT = B, M = B*A, P = POUT = 0, CARRYOUT = CARRYOUTF = 0
        repeat(10) begin
            // Randomize inputs
            A = $random;
            B = $random;
            C = $random;
            D = $random;
            CARRYIN = $random;
            BCIN = $random;
            PCIN = $random;
            @(negedge clk);
            if(BCOUT !== B) begin
                $display("Error");
                $stop;
            end
            @(negedge clk);
            if(M !== B*A) begin
                $display("Error");
                $stop;
            end
            @(negedge clk);
            if(P !== 0) begin
                $display("Error");
                $stop;
            end
```

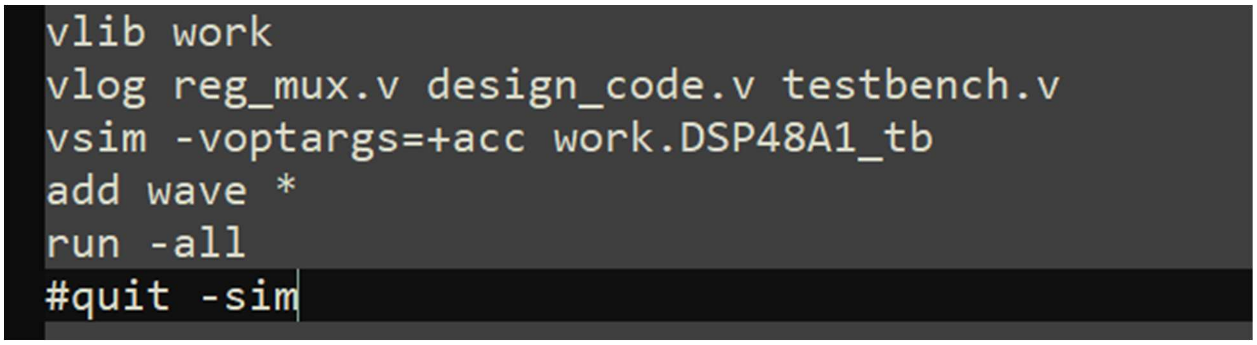


```
            if(PCOUT !== 0) begin
                $display("Error");
                $stop;
            end
            if(CARRYOUT !== 0) begin
                $display("Error");
                $stop;
            end
            if(CARRYOUTF !== 0) begin
                $display("Error");
                $stop;
            end
        end
        // OPCODE bits
        OPMODE[1:0] = 1;
        OPMODE[3:2] = 3;
        OPMODE [4] = 1;
        OPMODE [5] = 1;
        OPMODE [6] = 0;
        OPMODE [7] = 0; // mathimatical operations: BOUT = D+B, M = (D+B)*A, {CARRYOUT = CARRYOUTF, P = POUT} = (D+B)*A+C+CARRYIN
        repeat(10) begin
            // Randomize inputs
            A = $random;
            B = $random;
            C = $random;
            D = $random;
            CARRYIN = $random;
            BCIN = $random;
            PCIN = $random;
            BCOUT_temp = D + B;
            M_temp = BCOUT_temp * A;
            {CARRYOUT_temp, P_temp} = M_temp + C + OPMODE [5];
            @(negedge clk);
            if(BCOUT !== BCOUT_temp) begin
```



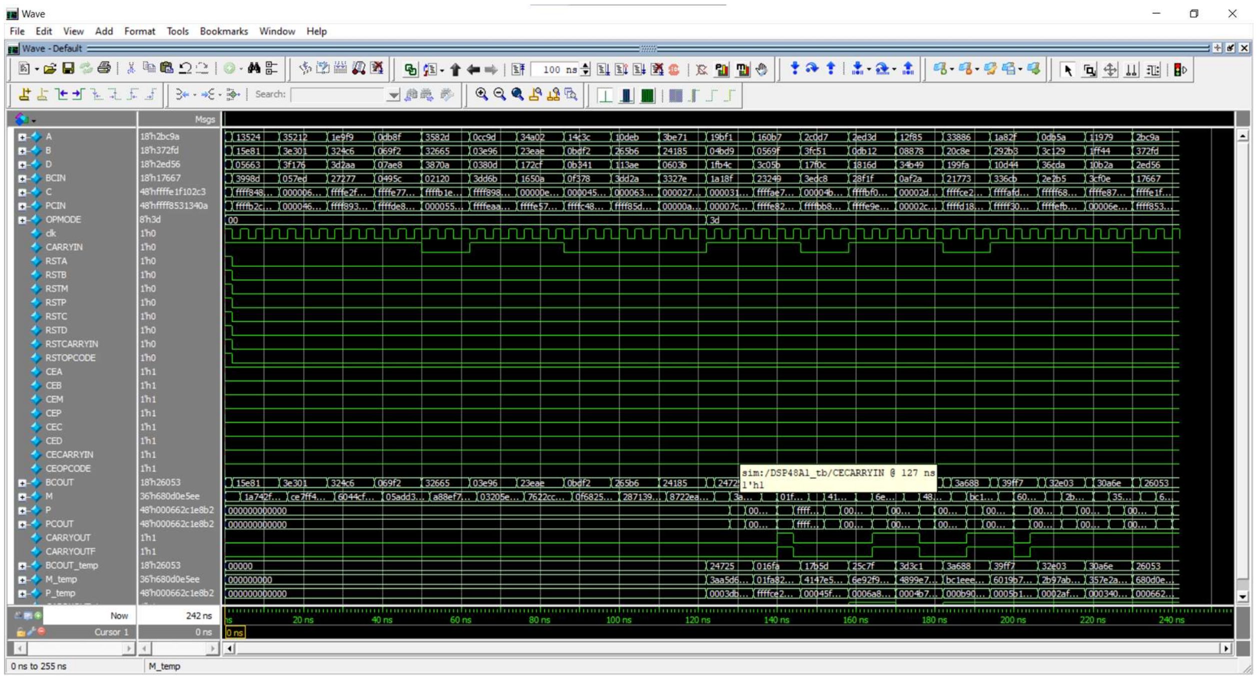
```
            @(negedge clk);
            if(BCOUT !== BCOUT_temp) begin
                $display("Error, BCOUT_temp = %h", BCOUT_temp);
                $stop;
            end
            @(negedge clk);
            if(M !== M_temp) begin
                $display("Error, M_temp = %h", M_temp);
                $stop;
            end
            @(negedge clk);
            if({CARRYOUT, P} !== {CARRYOUT_temp, P_temp}) begin
                $display("Error, {CARRYOUT_temp, P_temp} = %h", {CARRYOUT_temp, P_temp});
                $stop;
            end
            if({CARRYOUTF, PCOUT} !== {CARRYOUT_temp, P_temp}) begin
                $display("Error");
                $stop;
            end
        end

        $stop;
    end

    initial begin
        $monitor("A= %h, B= %h, C= %h, D= %h, CARRYIN= %h, BCIN= %h, PCIN = %h, OPMODE= %b
        BCOUT= %h, M= %h, P= %h, PCOUT= %h, CARRYOUT= %h, CARRYOUF= h",
        A, B, C, D, CARRYIN, BCIN, PCIN, OPMODE, BCOUT, M, P, PCOUT, CARRYOUT, CARRYOUTF);
    end

endmodule
```

Do file:



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

QuestaSim snippets:



Signal	Msgs
A	18'h2bc9a
B	18'h372fd
D	18'h2ed56
BCIN	18'h17667
C	48'hffffe1f102c3
PCIN	48'hffff8531340a
OPMODE	8'h3d
clk	1'h0
CARRYIN	1'h0
RSTA	1'h0
RSTB	1'h0
RSTM	1'h0
RSTP	1'h0
RSTC	1'h0
RSTD	1'h0
RSTCARRYIN	1'h0
RSTOPCODE	1'h0
CEA	1'h1
CEB	1'h1
CEM	1'h1
CEP	1'h1
CEC	1'h1
CED	1'h1
CECARRYIN	1'h1
CEOPCODE	1'h1
BCOUT	18'h26053
M	36'h680d0e5ee
P	48'h000662c1e8b2
PCOUT	48'h000662c1e8b2
CARRYOUT	1'h1
CARRYOUTF	1'h1
BCOUT_temp	18'h26053
M_temp	36'h680d0e5ee
P_temp	48'h000662c1e8b2
Now	242 ns
Cursor 1	0 ns

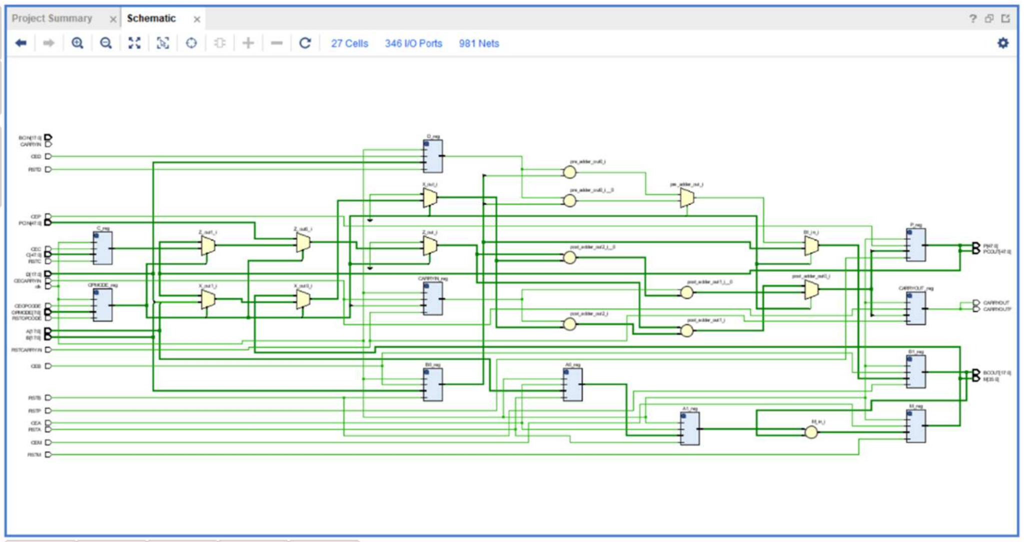
```
sim:/DSP48A1_tb/CECARRYIN @ 127 ns
1'h1
```

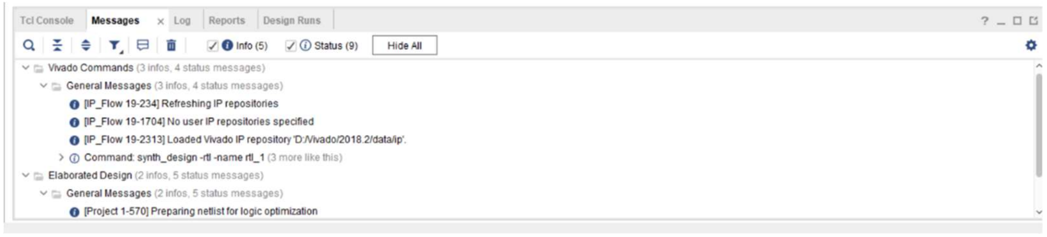
```
0 ns to 255 ns    M_temp
```

Constraints file:

```
## 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]
```

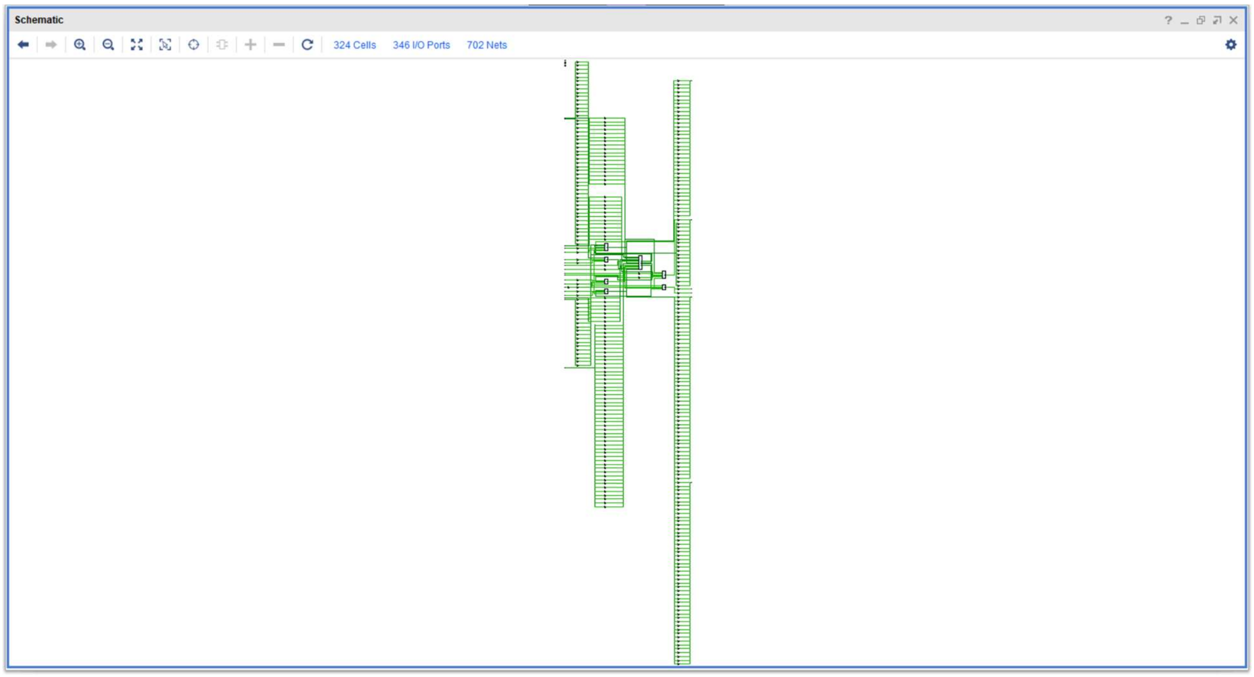
Elaboration:

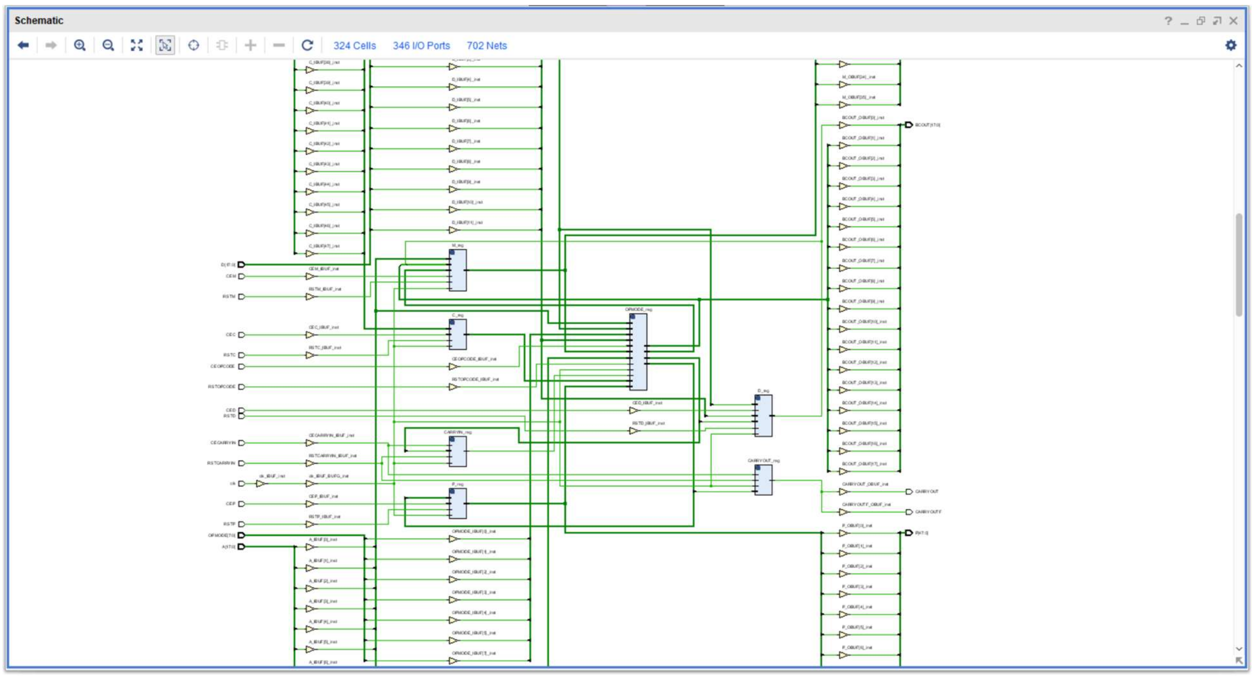


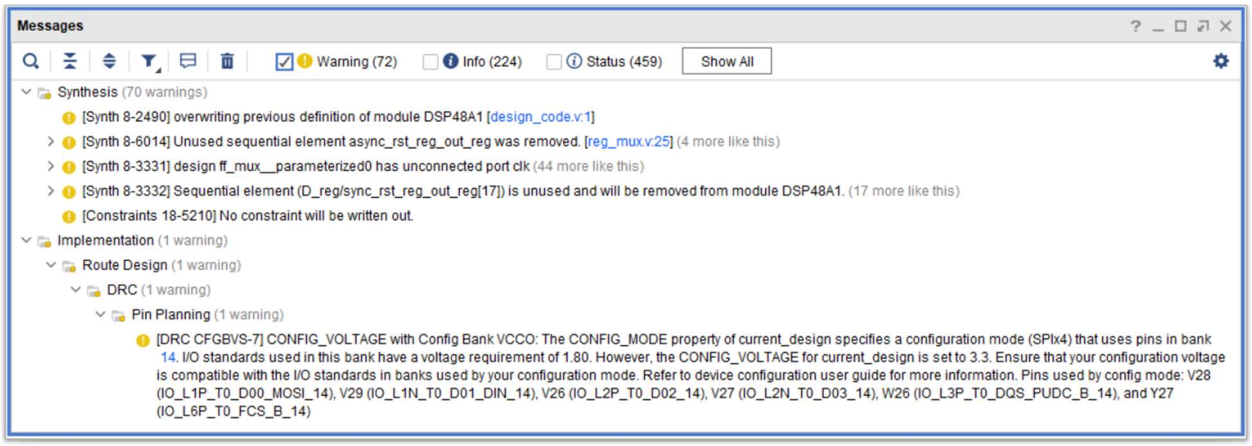


```
Vivado Commands (3 infos, 4 status messages)
  General Messages (3 infos, 4 status messages)
    [IP_Flow 19-234] Refreshing IP repositories
    [IP_Flow 19-1704] No user IP repositories specified
    [IP_Flow 19-2313] Loaded Vivado IP repository 'D:/Vivado/2018.2/data/ip'.
  > Command: synth_design -rtl -name rtl_1 (3 more like this)
Elaborated Design (2 infos, 5 status messages)
  General Messages (2 infos, 5 status messages)
    [Project 1-570] Preparing netlist for logic optimization
```

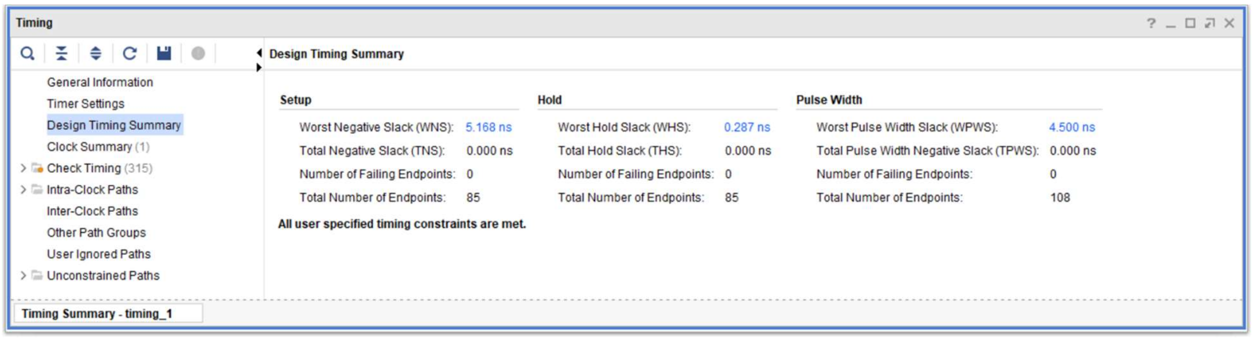
Synthesis:







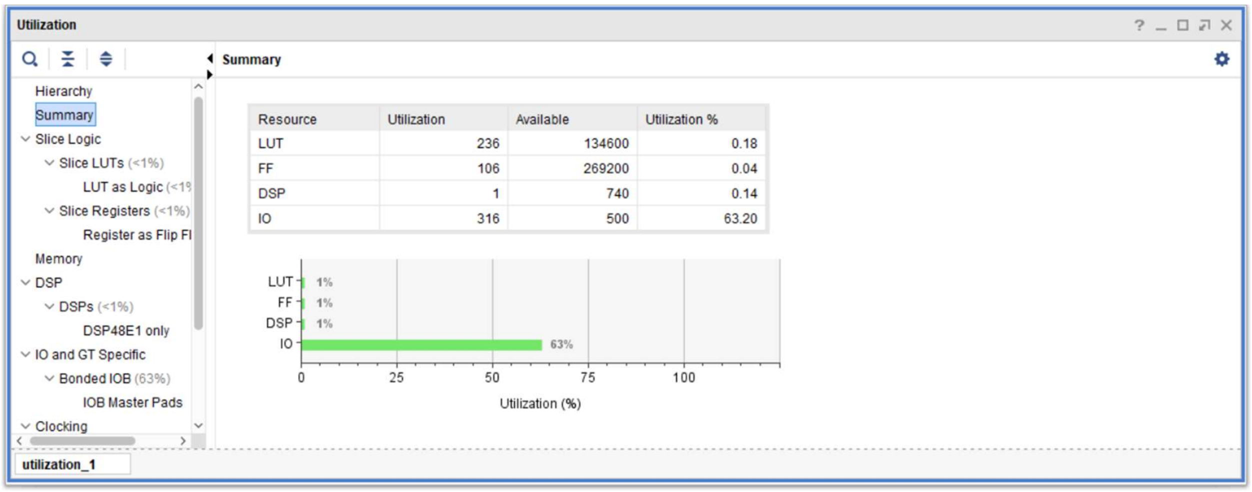
```
Synthesis (70 warnings)
    [Synth 8-2490] overwriting previous definition of module DSP48A1 [design_code.v:1]
  > [Synth 8-6014] Unused sequential element async_rst_reg_out_reg was removed. [reg_mux.v:25] (4 more like this)
  > [Synth 8-3331] design ff_mux__parameterized0 has unconnected port clk (44 more like this)
  > [Synth 8-3332] Sequential element (D_reg/sync_rst_reg_out_reg[17]) 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)
```



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.168 ns	Worst Hold Slack (WHS):	0.287 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	85	Total Number of Endpoints:	85	Total Number of Endpoints:	108

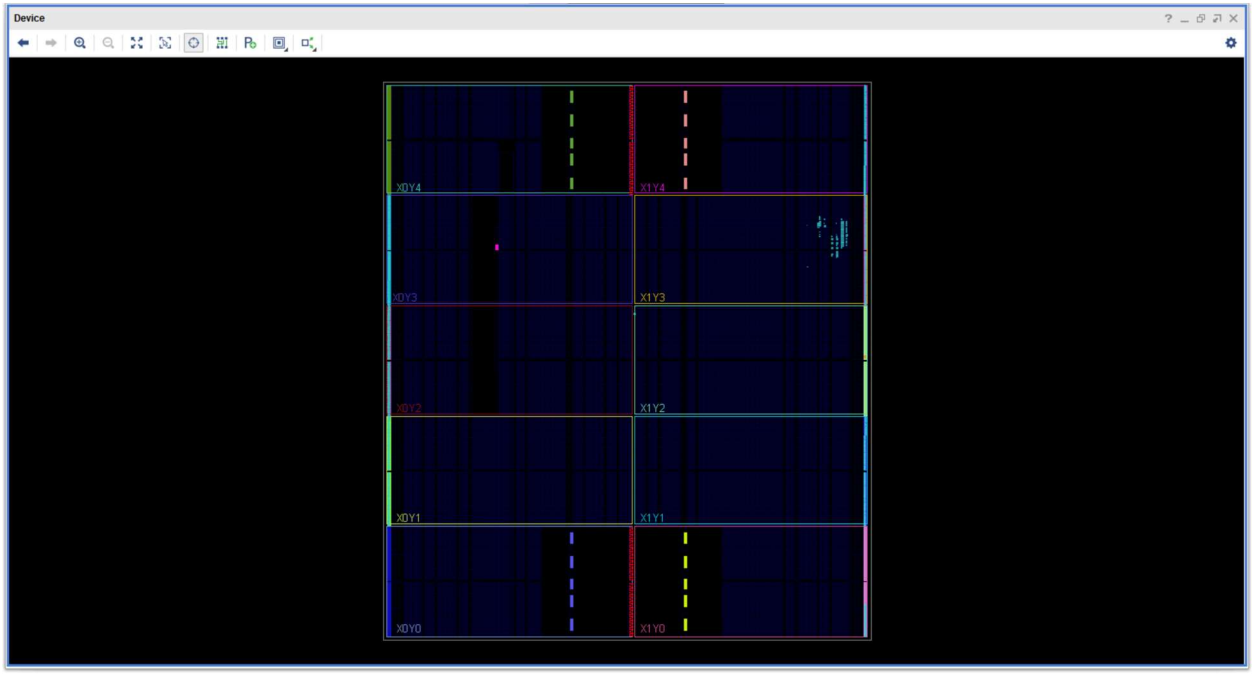
All user specified timing constraints are met.

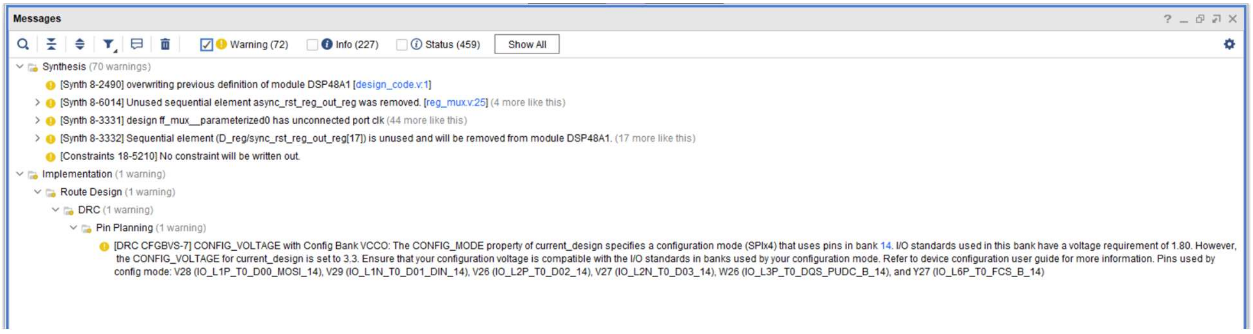


Summary

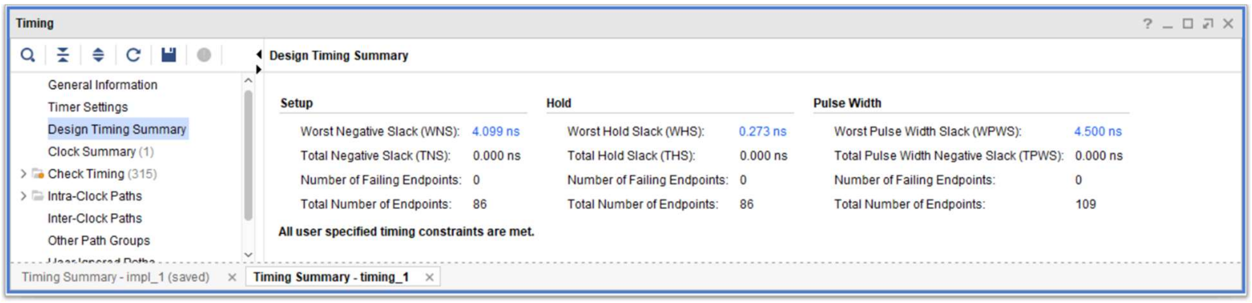
Resource	Utilization	Available	Utilization %
LUT	236	134600	0.18
FF	106	269200	0.04
DSP	1	740	0.14
IO	316	500	63.20

Implementation:





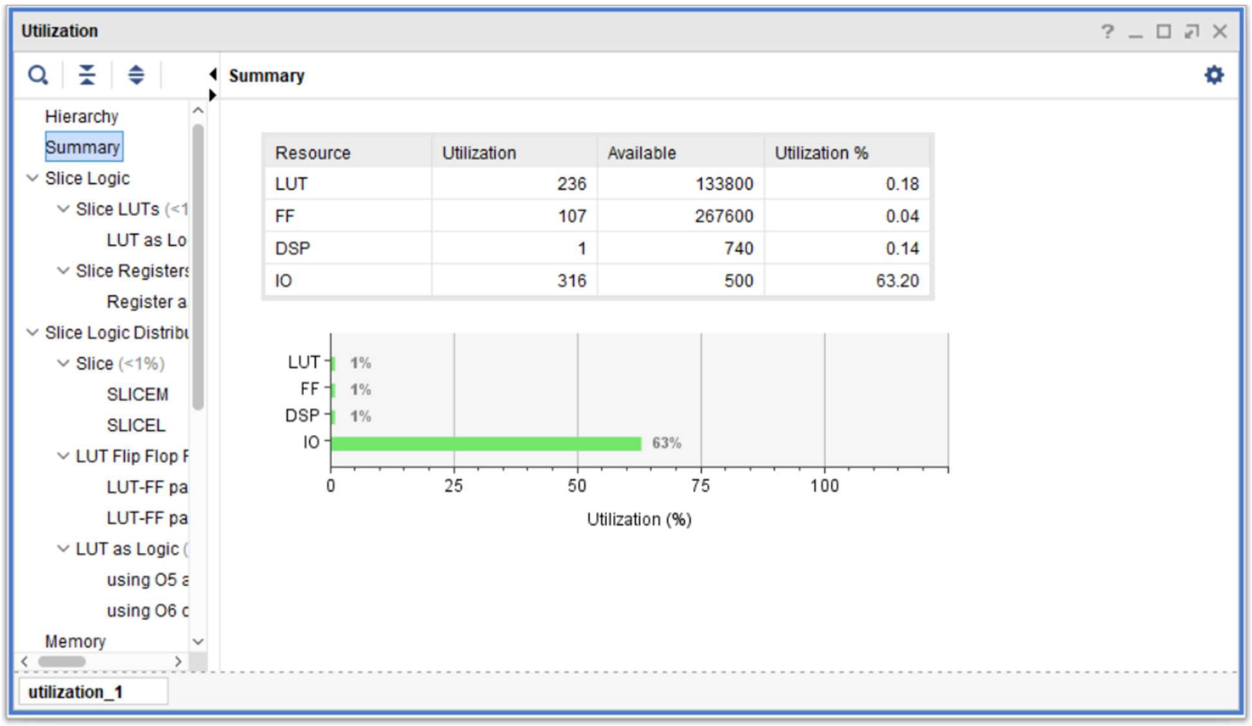
```
Synthesis (70 warnings)
    [Synth 8-2490] overwriting previous definition of module DSP48A1 [design_code.v:1]
  > [Synth 8-6014] Unused sequential element async_rst_reg_out_reg was removed. [reg_mux.v:25] (4 more like this)
  > [Synth 8-3331] design ff_mux__parameterized0 has unconnected port clk (44 more like this)
  > [Synth 8-3332] Sequential element (D_reg/sync_rst_reg_out_reg[17]) 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)
```



Design Timing Summary

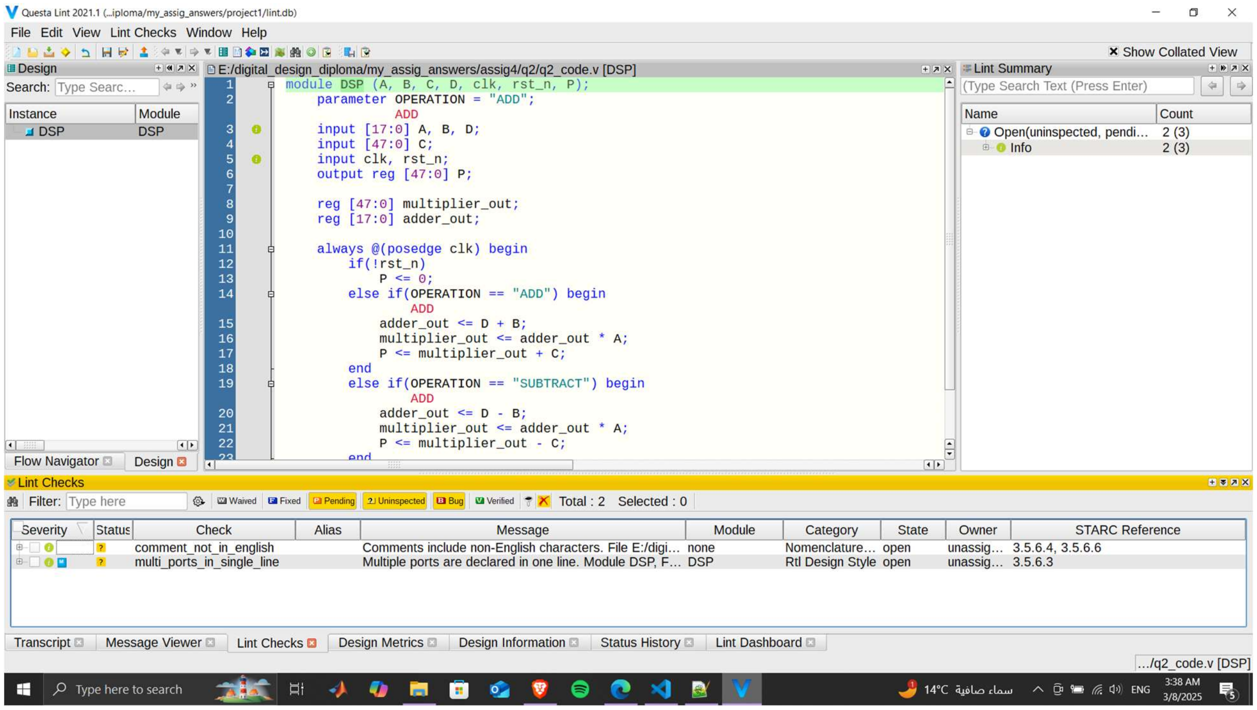
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.099 ns	Worst Hold Slack (WHS):	0.273 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	86	Total Number of Endpoints:	86	Total Number of Endpoints:	109

All user specified timing constraints are met.



Resource	Utilization	Available	Utilization %
LUT	236	133800	0.18
FF	107	267600	0.04
DSP	1	740	0.14
IO	316	500	63.20

Linting:



E:/digital_design_diploma/my_assig_answers/assig4/q2/q2_code.v [DSP]

```
module DSP (A, B, C, D, clk, rst_n, P);
    parameter OPERATION = "ADD";
              ADD
    input [17:0] A, B, D;
    input [47:0] C;
    input clk, rst_n;
    output reg [47:0] P;

    reg [47:0] multiplier_out;
    reg [17:0] adder_out;

    always @(posedge clk) begin
        if(!rst_n)
            P <= 0;
        else if(OPERATION == "ADD") begin
                ADD
            adder_out <= D + B;
            multiplier_out <= adder_out * A;
            P <= multiplier_out + C;
        end
        else if(OPERATION == "SUBTRACT") begin
                ADD
            adder_out <= D - B;
            multiplier_out <= adder_out * A;
            P <= multiplier_out - C;
        end
```

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference
	?	comment_not_in_english		Comments include non-English characters. File E:/digi...	none	Nomenclature...	open	unassig...	3.5.6.4, 3.5.6.6
	?	multi_ports_in_single_line		Multiple ports are declared in one line. Module DSP, F...	DSP	Rtl Design Style	open	unassig...	3.5.6.3