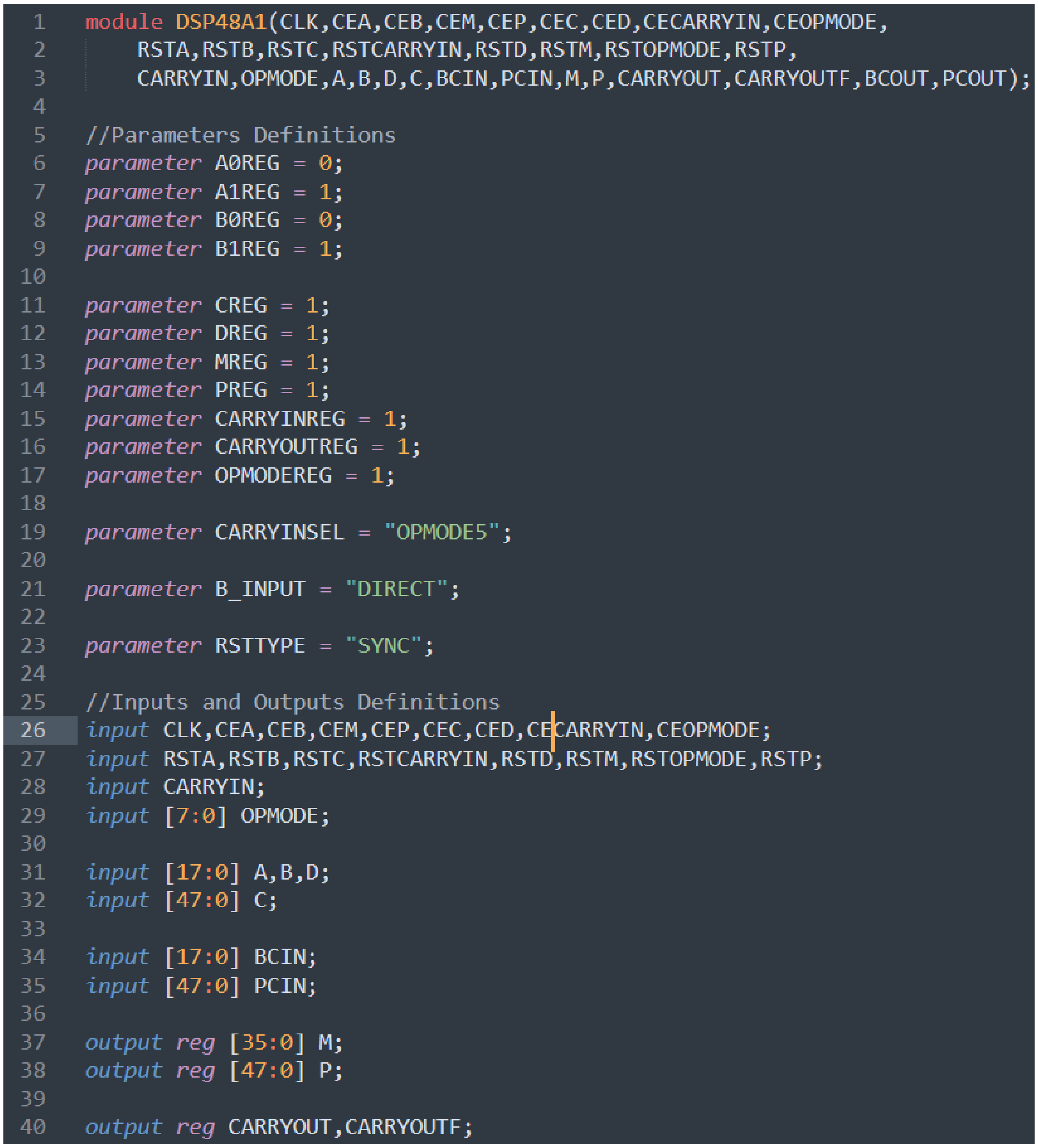
Digital Design using Verilog

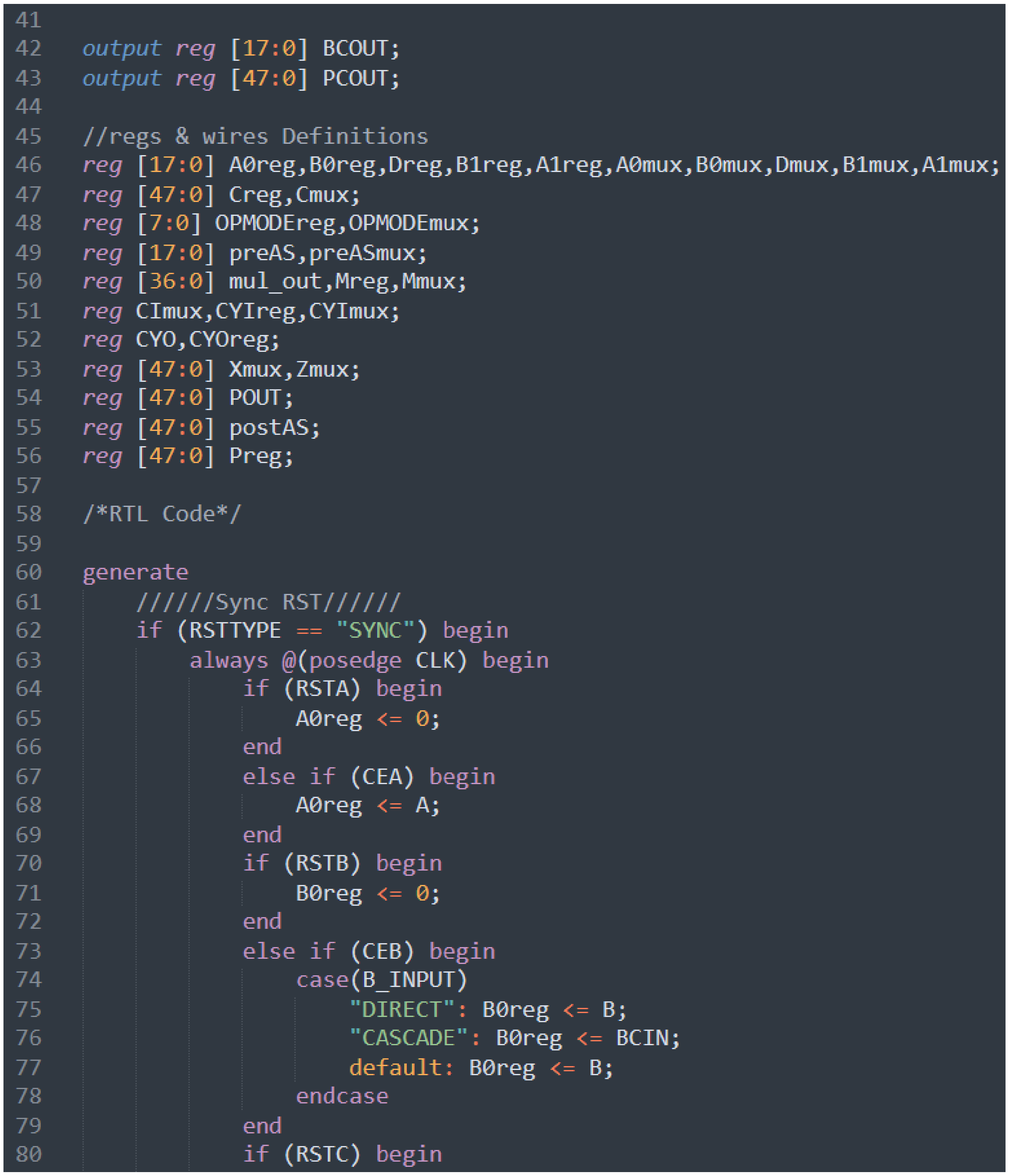
Project 1: DSP48A1

Name	Sohaila Ahmed Mohamed

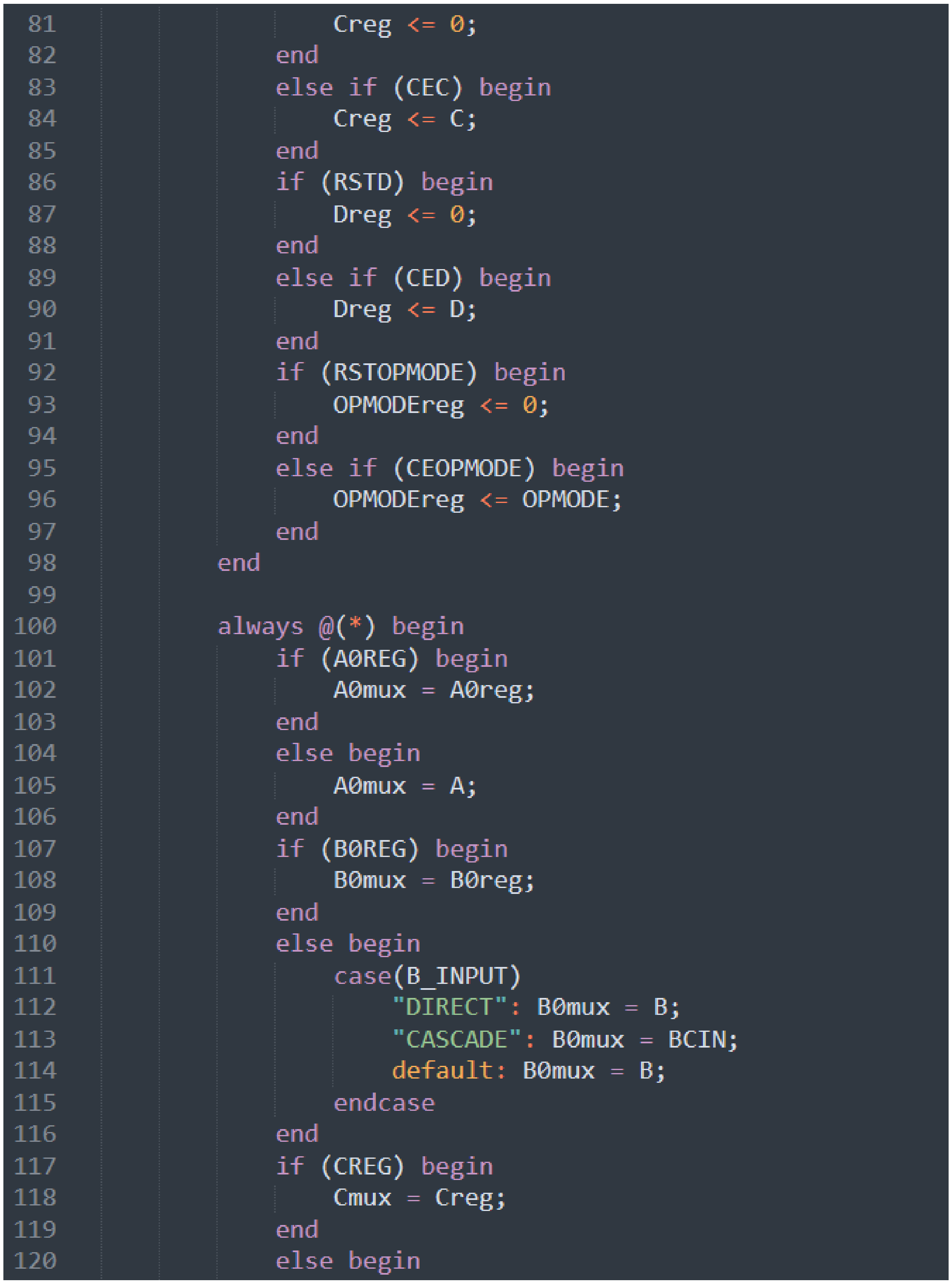
1. RTL code



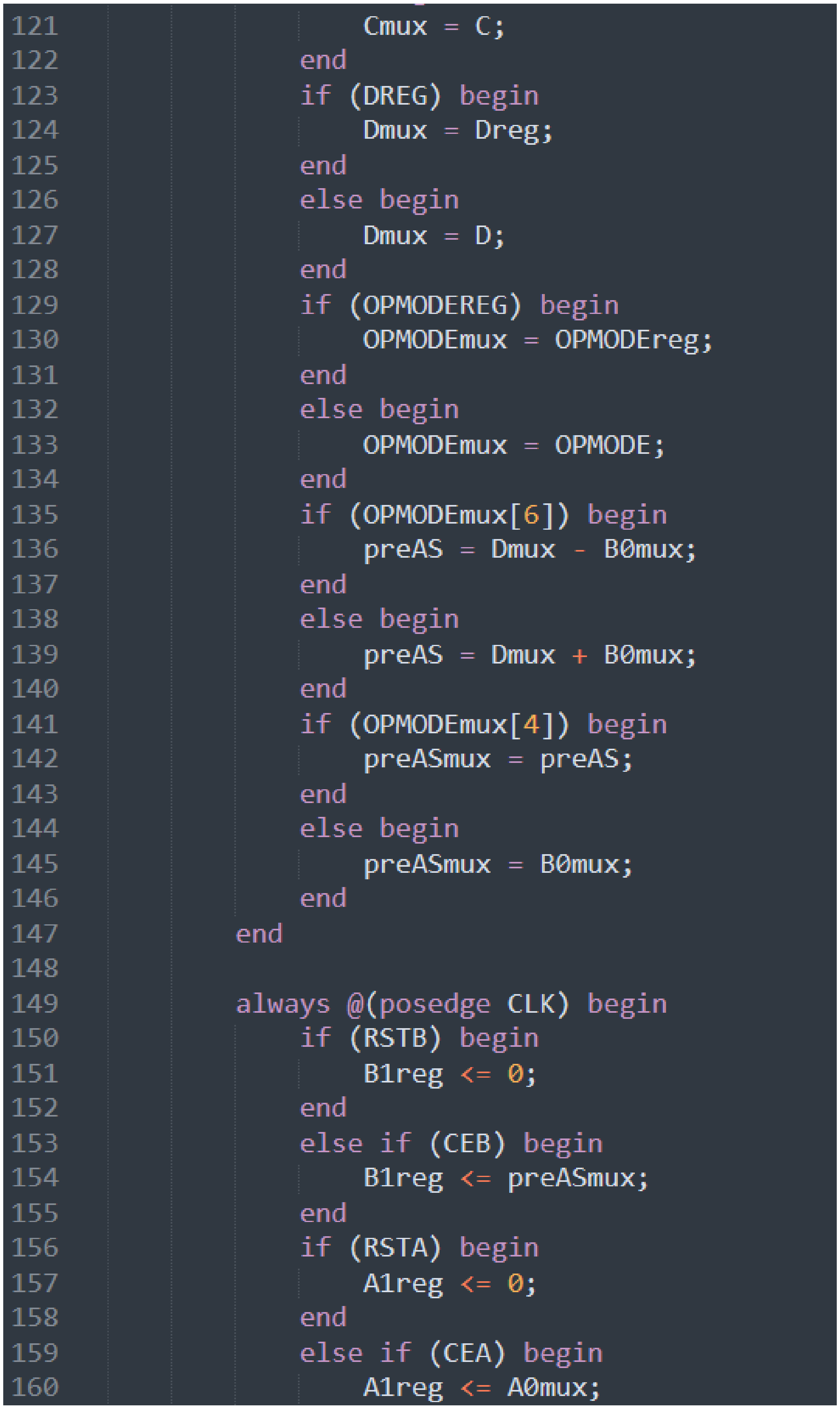
```verilog
module DSP48A1(CLK,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE,
    RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP,
    CARRYIN,OPMODE,A,B,D,C,BCIN,PCIN,M,P,CARRYOUT,CARRYOUTF,BCOUT,PCOUT);

//Parameters Definitions
parameter A0REG = 0;
parameter A1REG = 1;
parameter B0REG = 0;
parameter B1REG = 1;

parameter CREG = 1;
parameter DREG = 1;
parameter MREG = 1;
parameter PREG = 1;
parameter CARRYINREG = 1;
parameter CARRYOUTREG = 1;
parameter OPMODEREG = 1;

parameter CARRYINSEL = "OPMODE5";

parameter B_INPUT = "DIRECT";

parameter RSTTYPE = "SYNC";

//Inputs and Outputs Definitions
input CLK,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE;
input RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP;
input CARRYIN;
input [7:0] OPMODE;

input [17:0] A,B,D;
input [47:0] C;

input [17:0] BCIN;
input [47:0] PCIN;

output reg [35:0] M;
output reg [47:0] P;

output reg CARRYOUT,CARRYOUTF;
```



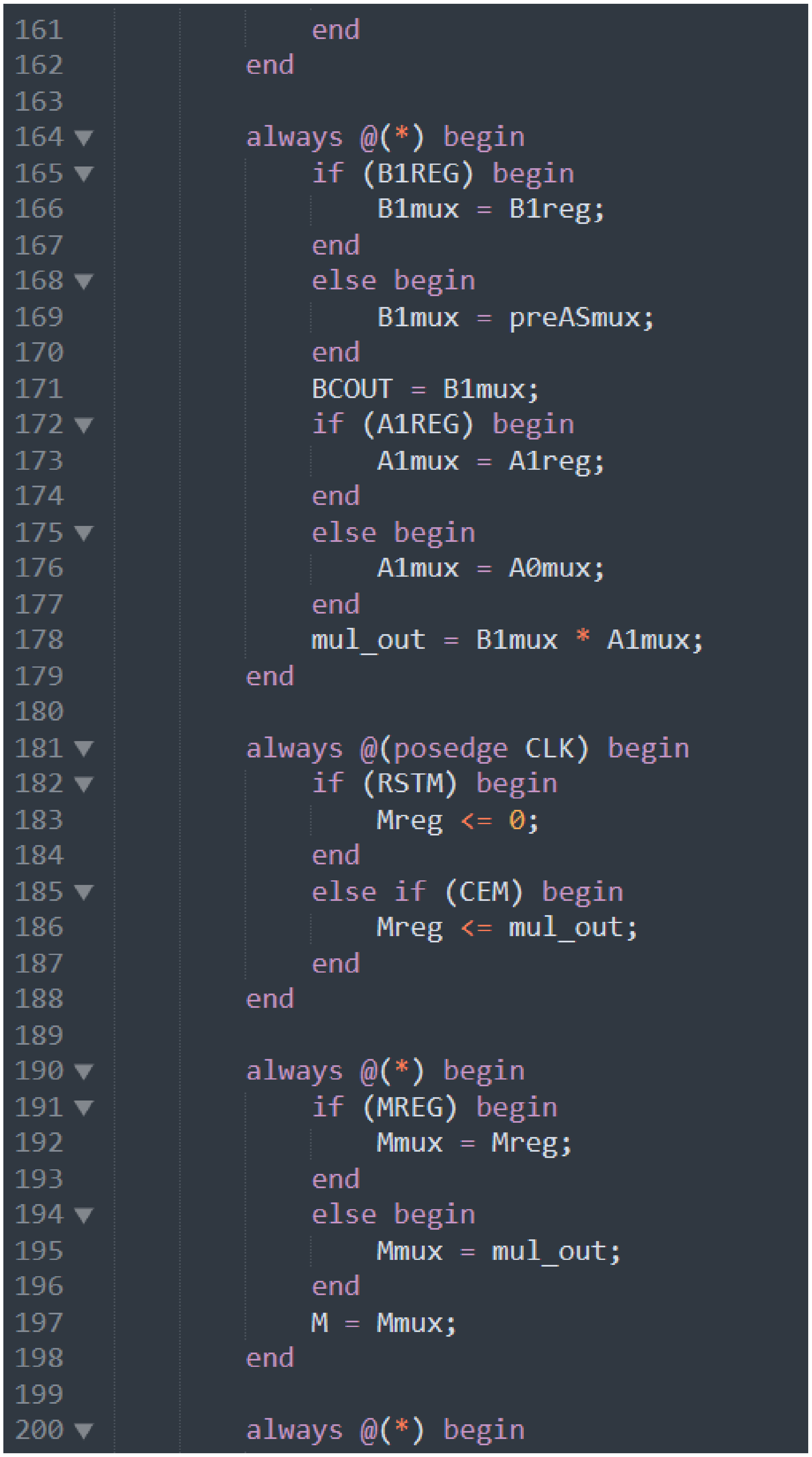
```verilog

output reg [17:0] BCOUT;
output reg [47:0] PCOUT;

//regs & wires Definitions
reg [17:0] A0reg,B0reg,Dreg,B1reg,A1reg,A0mux,B0mux,Dmux,B1mux,A1mux;
reg [47:0] Creg,Cmux;
reg [7:0] OPMODEreg,OPMODEmux;
reg [17:0] preAS,preASmux;
reg [36:0] mul_out,Mreg,Mmux;
reg CImux,CYIreg,CYImux;
reg CYO,CYOreg;
reg [47:0] Xmux,Zmux;
reg [47:0] POUT;
reg [47:0] postAS;
reg [47:0] Preg;

/*RTL Code*/

generate
    //////Sync RST//////
    if (RSTTYPE == "SYNC") begin
        always @(posedge CLK) begin
            if (RSTA) begin
                A0reg <= 0;
            end
            else if (CEA) begin
                A0reg <= A;
            end
            if (RSTB) begin
                B0reg <= 0;
            end
            else if (CEB) begin
                case(B_INPUT)
                    "DIRECT": B0reg <= B;
                    "CASCADE": B0reg <= BCIN;
                    default: B0reg <= B;
                endcase
            end
            if (RSTC) begin
```



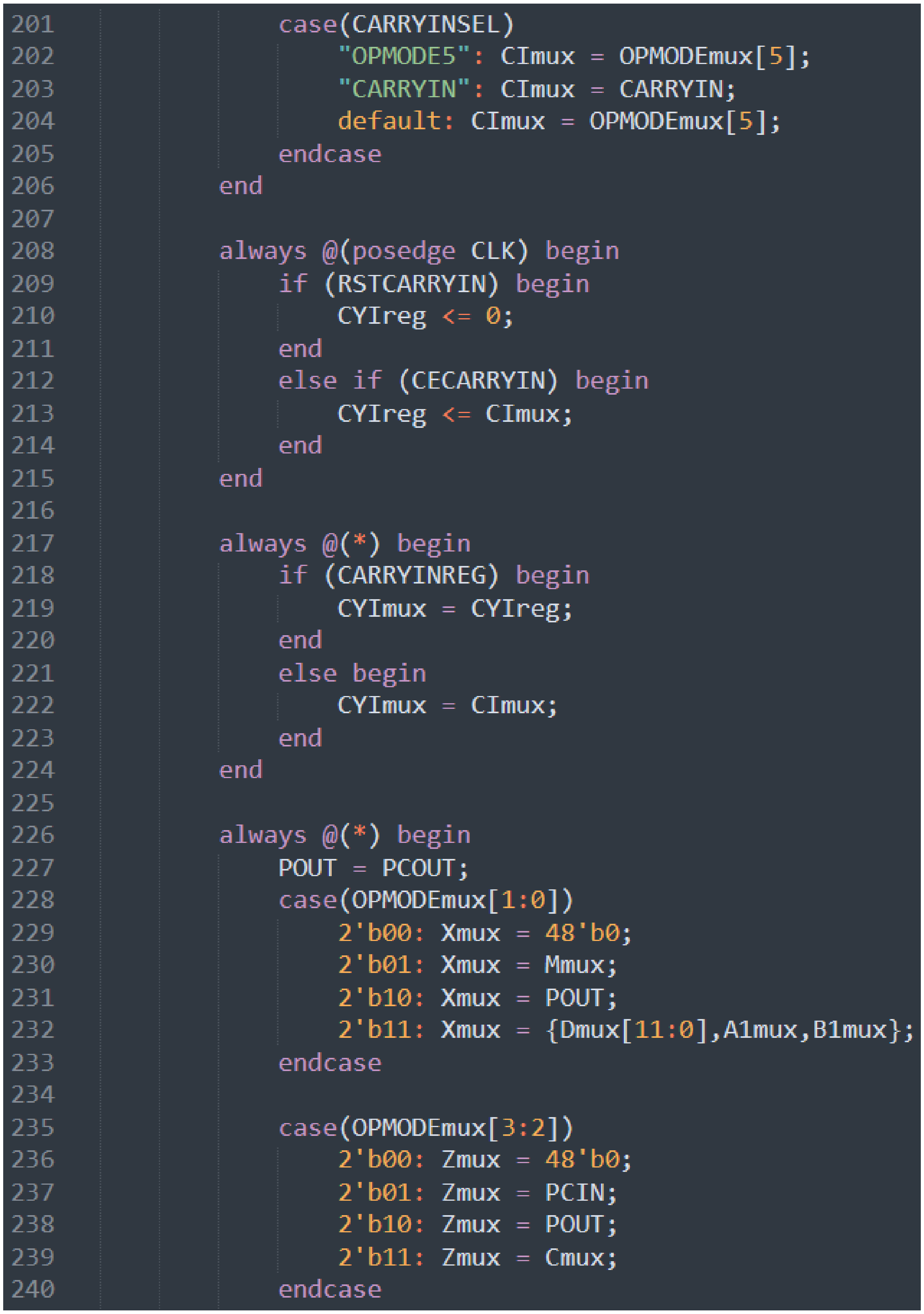
```
                Creg <= 0;
            end
            else if (CEC) begin
                Creg <= C;
            end
            if (RSTD) begin
                Dreg <= 0;
            end
            else if (CED) begin
                Dreg <= D;
            end
            if (RSTOPMODE) begin
                OPMODEreg <= 0;
            end
            else if (CEOPMODE) begin
                OPMODEreg <= OPMODE;
            end
        end

        always @(*) begin
            if (A0REG) begin
                A0mux = A0reg;
            end
            else begin
                A0mux = A;
            end
            if (B0REG) begin
                B0mux = B0reg;
            end
            else begin
                case(B_INPUT)
                    "DIRECT": B0mux = B;
                    "CASCADE": B0mux = BCIN;
                    default: B0mux = B;
                endcase
            end
            if (CREG) begin
                Cmux = Creg;
            end
            else begin
```



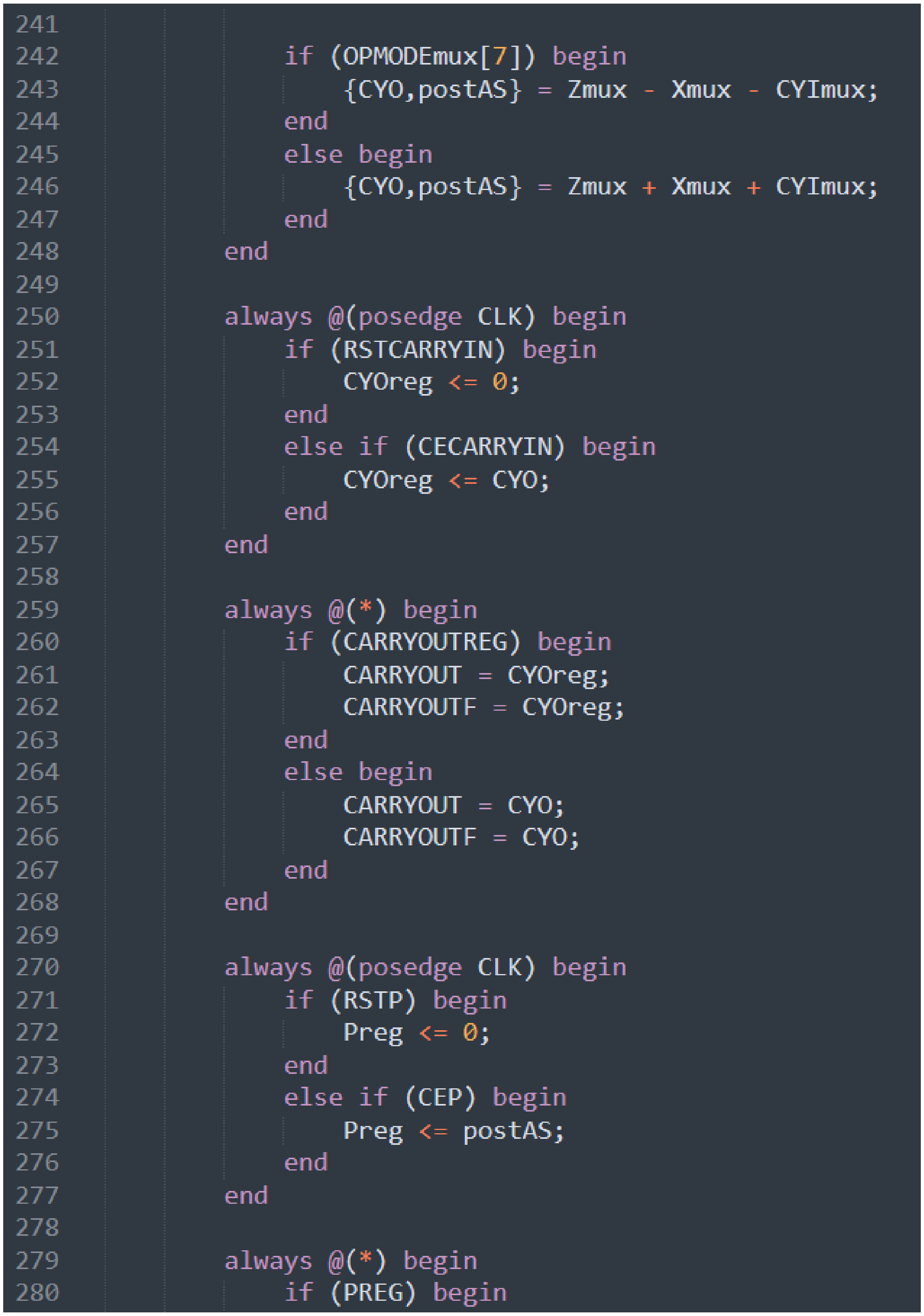
```
                    Cmux = C;
                end
                if (DREG) begin
                    Dmux = Dreg;
                end
                else begin
                    Dmux = D;
                end
                if (OPMODEREG) begin
                    OPMODEmux = OPMODEreg;
                end
                else begin
                    OPMODEmux = OPMODE;
                end
                if (OPMODEmux[6]) begin
                    preAS = Dmux - B0mux;
                end
                else begin
                    preAS = Dmux + B0mux;
                end
                if (OPMODEmux[4]) begin
                    preASmux = preAS;
                end
                else begin
                    preASmux = B0mux;
                end
            end

            always @(posedge CLK) begin
                if (RSTB) begin
                    B1reg <= 0;
                end
                else if (CEB) begin
                    B1reg <= preASmux;
                end
                if (RSTA) begin
                    A1reg <= 0;
                end
                else if (CEA) begin
                    A1reg <= A0mux;
```



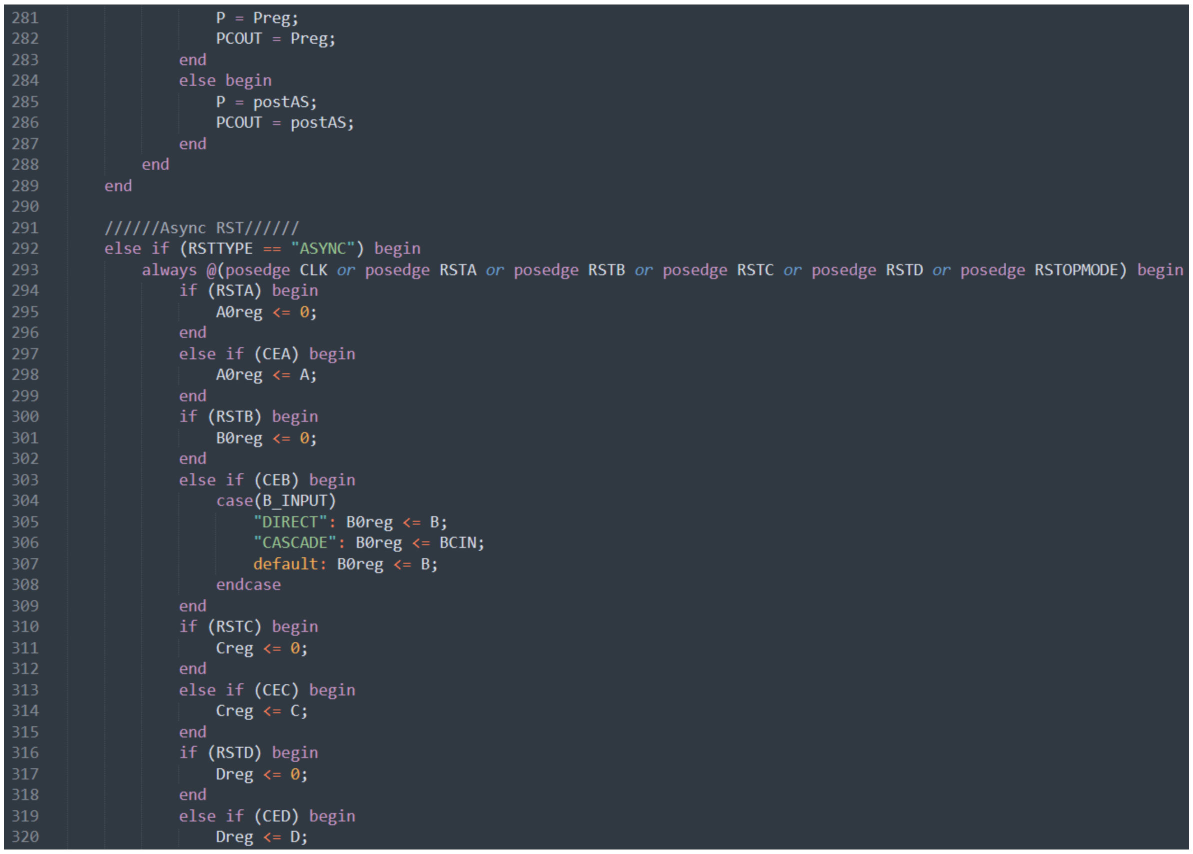
```
            end
        end

        always @(*) begin
            if (B1REG) begin
                B1mux = B1reg;
            end
            else begin
                B1mux = preASmux;
            end
            BCOUT = B1mux;
            if (A1REG) begin
                A1mux = A1reg;
            end
            else begin
                A1mux = A0mux;
            end
            mul_out = B1mux * A1mux;
        end

        always @(posedge CLK) begin
            if (RSTM) begin
                Mreg <= 0;
            end
            else if (CEM) begin
                Mreg <= mul_out;
            end
        end

        always @(*) begin
            if (MREG) begin
                Mmux = Mreg;
            end
            else begin
                Mmux = mul_out;
            end
            M = Mmux;
        end

        always @(*) begin
```



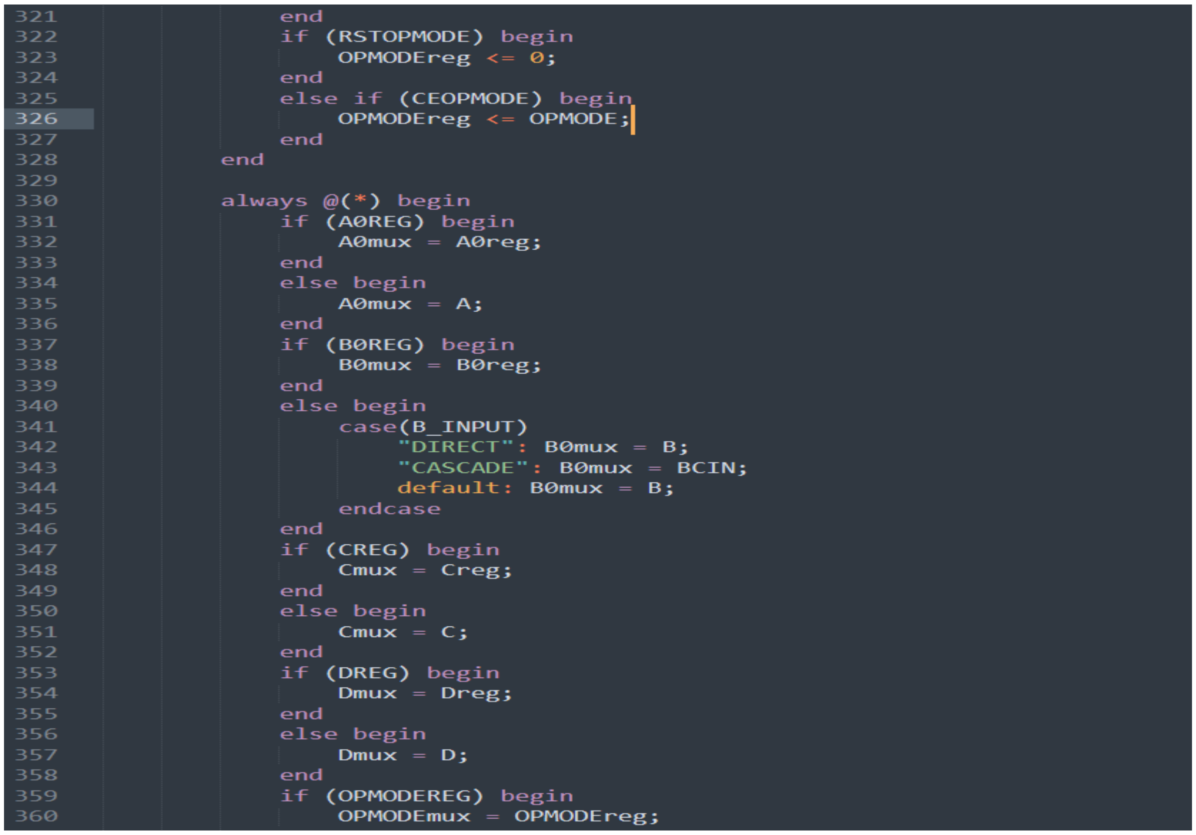
```
                case(CARRYINSEL)
                    "OPMODE5": CImux = OPMODEmux[5];
                    "CARRYIN": CImux = CARRYIN;
                    default: CImux = OPMODEmux[5];
                endcase
            end

            always @(posedge CLK) begin
                if (RSTCARRYIN) begin
                    CYIreg <= 0;
                end
                else if (CECARRYIN) begin
                    CYIreg <= CImux;
                end
            end

            always @(*) begin
                if (CARRYINREG) begin
                    CYImux = CYIreg;
                end
                else begin
                    CYImux = CImux;
                end
            end

            always @(*) begin
                POUT = PCOUT;
                case(OPMODEmux[1:0])
                    2'b00: Xmux = 48'b0;
                    2'b01: Xmux = Mmux;
                    2'b10: Xmux = POUT;
                    2'b11: Xmux = {Dmux[11:0],A1mux,B1mux};
                endcase

                case(OPMODEmux[3:2])
                    2'b00: Zmux = 48'b0;
                    2'b01: Zmux = PCIN;
                    2'b10: Zmux = POUT;
                    2'b11: Zmux = Cmux;
                endcase
```



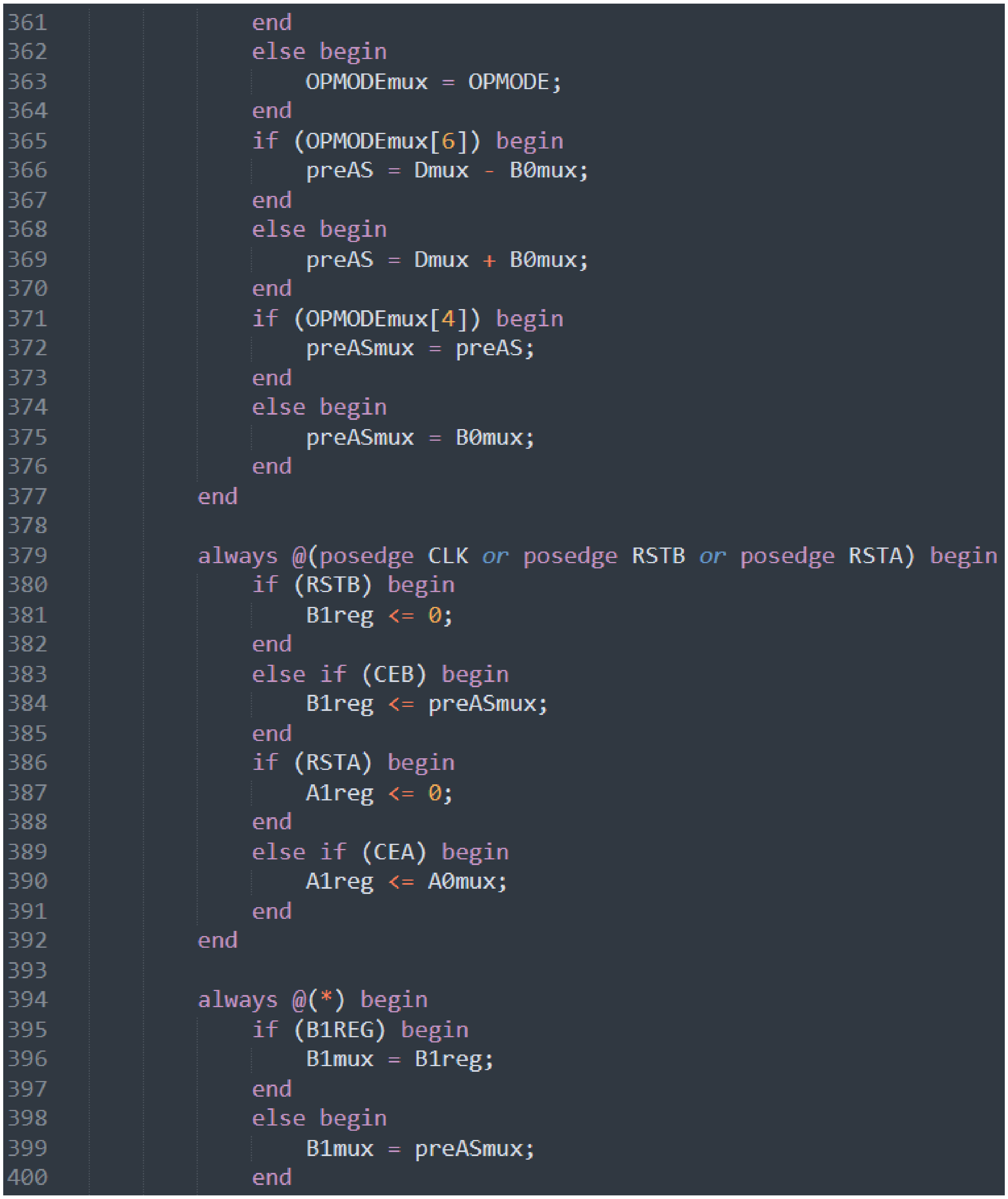
```verilog

            if (OPMODEmux[7]) begin
                {CYO,postAS} = Zmux - Xmux - CYImux;
            end
            else begin
                {CYO,postAS} = Zmux + Xmux + CYImux;
            end
        end

        always @(posedge CLK) begin
            if (RSTCARRYIN) begin
                CYOreg <= 0;
            end
            else if (CECARRYIN) begin
                CYOreg <= CYO;
            end
        end

        always @(*) begin
            if (CARRYOUTREG) begin
                CARRYOUT  = CYOreg;
                CARRYOUTF = CYOreg;
            end
            else begin
                CARRYOUT  = CYO;
                CARRYOUTF = CYO;
            end
        end

        always @(posedge CLK) begin
            if (RSTP) begin
                Preg <= 0;
            end
            else if (CEP) begin
                Preg <= postAS;
            end
        end

        always @(*) begin
            if (PREG) begin
```



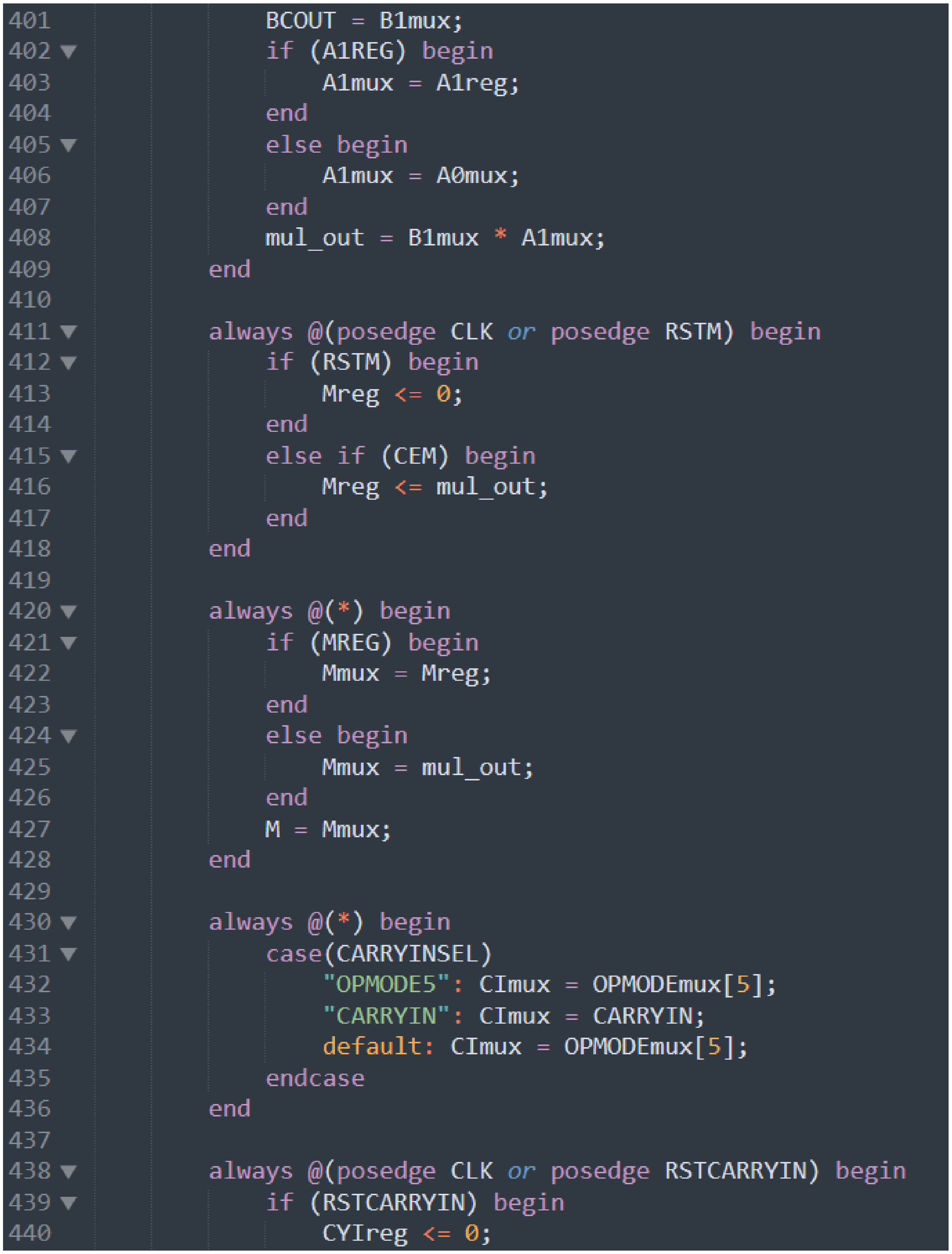
```
                P = Preg;
                PCOUT = Preg;
            end
            else begin
                P = postAS;
                PCOUT = postAS;
            end
        end
    end

    //////Async RST//////
    else if (RSTTYPE == "ASYNC") begin
        always @(posedge CLK or posedge RSTA or posedge RSTB or posedge RSTC or posedge RSTD or posedge RSTOPMODE) begin
            if (RSTA) begin
                A0reg <= 0;
            end
            else if (CEA) begin
                A0reg <= A;
            end
            if (RSTB) begin
                B0reg <= 0;
            end
            else if (CEB) begin
                case(B_INPUT)
                    "DIRECT": B0reg <= B;
                    "CASCADE": B0reg <= BCIN;
                    default: B0reg <= B;
                endcase
            end
            if (RSTC) begin
                Creg <= 0;
            end
            else if (CEC) begin
                Creg <= C;
            end
            if (RSTD) begin
                Dreg <= 0;
            end
            else if (CED) begin
                Dreg <= D;
```



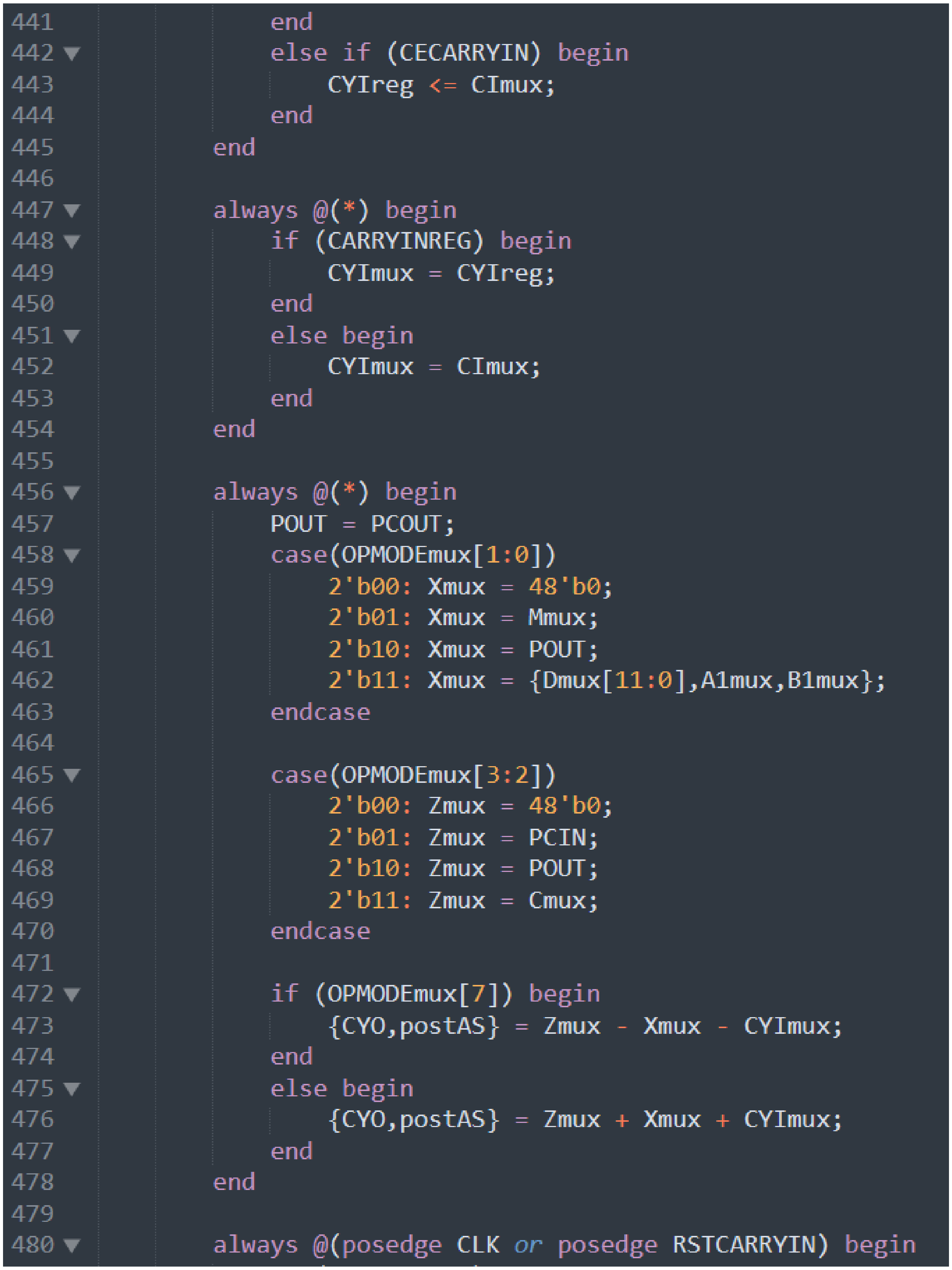
```
            end
            if (RSTOPMODE) begin
                OPMODEreg <= 0;
            end
            else if (CEOPMODE) begin
                OPMODEreg <= OPMODE;
            end
        end

        always @(*) begin
            if (A0REG) begin
                A0mux = A0reg;
            end
            else begin
                A0mux = A;
            end
            if (B0REG) begin
                B0mux = B0reg;
            end
            else begin
                case(B_INPUT)
                    "DIRECT": B0mux = B;
                    "CASCADE": B0mux = BCIN;
                    default: B0mux = B;
                endcase
            end
            if (CREG) begin
                Cmux = Creg;
            end
            else begin
                Cmux = C;
            end
            if (DREG) begin
                Dmux = Dreg;
            end
            else begin
                Dmux = D;
            end
            if (OPMODEREG) begin
                OPMODEmux = OPMODEreg;
```



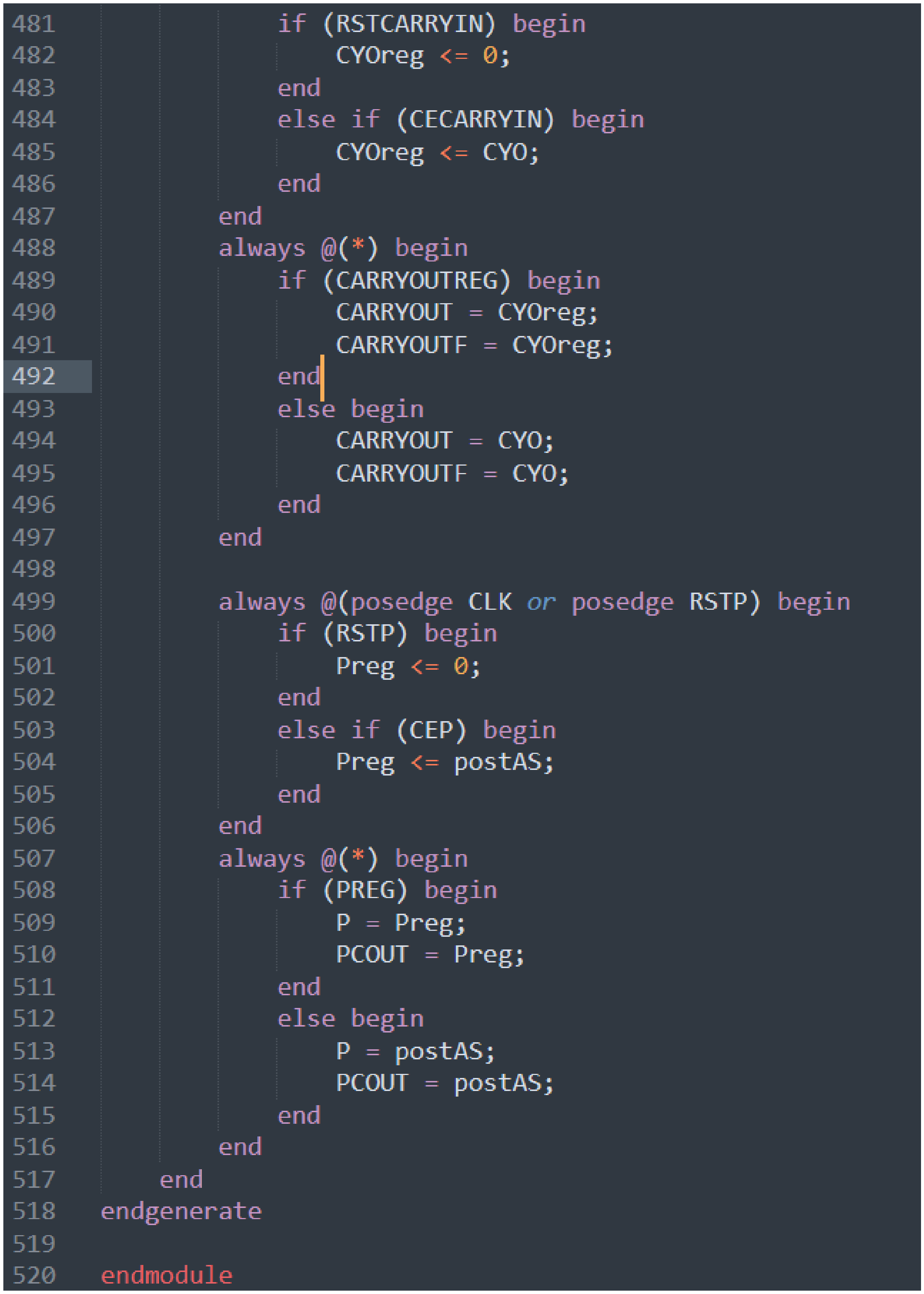
```
            end
            else begin
                OPMODEmux = OPMODE;
            end
            if (OPMODEmux[6]) begin
                preAS = Dmux - B0mux;
            end
            else begin
                preAS = Dmux + B0mux;
            end
            if (OPMODEmux[4]) begin
                preASmux = preAS;
            end
            else begin
                preASmux = B0mux;
            end
        end

        always @(posedge CLK or posedge RSTB or posedge RSTA) begin
            if (RSTB) begin
                B1reg <= 0;
            end
            else if (CEB) begin
                B1reg <= preASmux;
            end
            if (RSTA) begin
                A1reg <= 0;
            end
            else if (CEA) begin
                A1reg <= A0mux;
            end
        end

        always @(*) begin
            if (B1REG) begin
                B1mux = B1reg;
            end
            else begin
                B1mux = preASmux;
            end
```



```
            BCOUT = B1mux;
            if (A1REG) begin
                A1mux = A1reg;
            end
            else begin
                A1mux = A0mux;
            end
            mul_out = B1mux * A1mux;
        end

        always @(posedge CLK or posedge RSTM) begin
            if (RSTM) begin
                Mreg <= 0;
            end
            else if (CEM) begin
                Mreg <= mul_out;
            end
        end

        always @(*) begin
            if (MREG) begin
                Mmux = Mreg;
            end
            else begin
                Mmux = mul_out;
            end
            M = Mmux;
        end

        always @(*) begin
            case(CARRYINSEL)
                "OPMODE5": CImux = OPMODEmux[5];
                "CARRYIN": CImux = CARRYIN;
                default: CImux = OPMODEmux[5];
            endcase
        end

        always @(posedge CLK or posedge RSTCARRYIN) begin
            if (RSTCARRYIN) begin
                CYIreg <= 0;
```

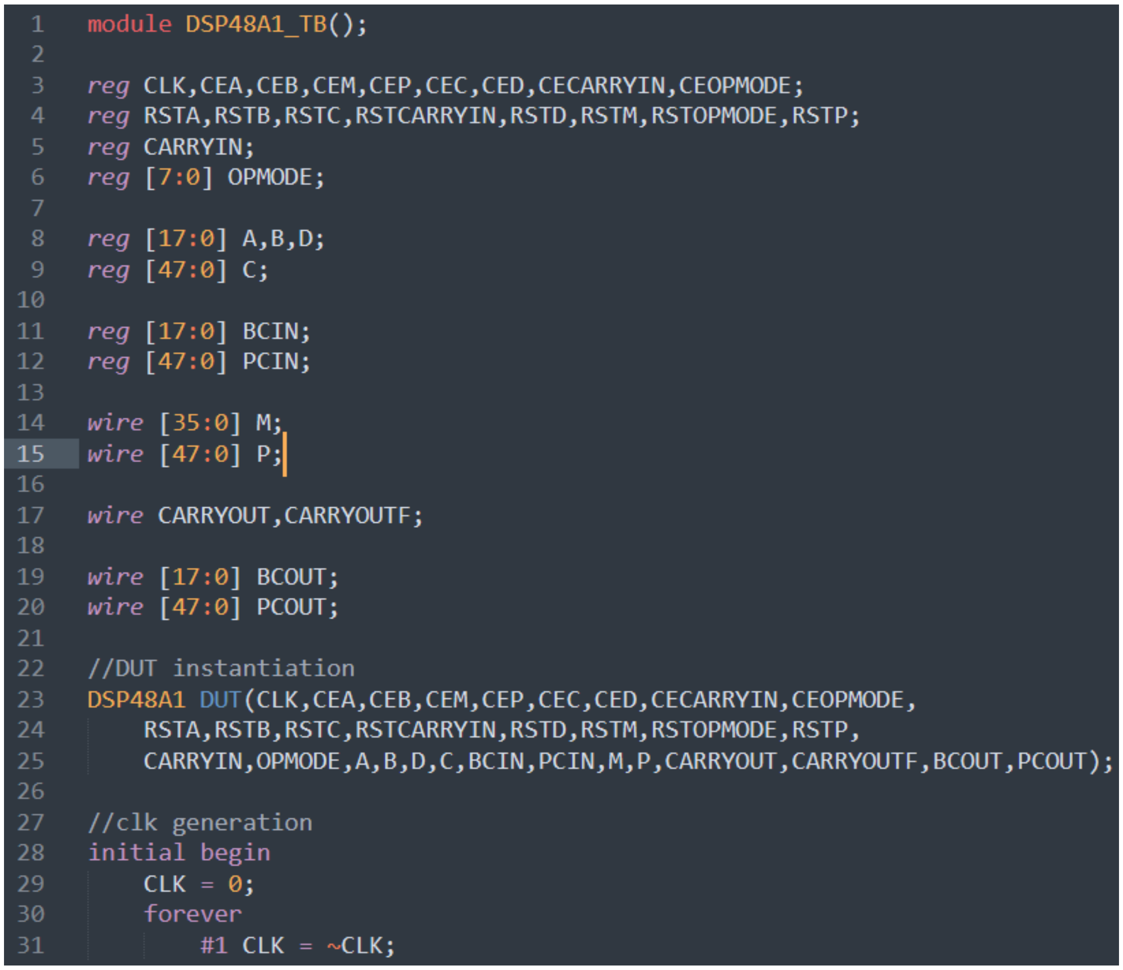


```verilog
            end
            else if (CECARRYIN) begin
                CYIreg <= CImux;
            end
        end

        always @(*) begin
            if (CARRYINREG) begin
                CYImux = CYIreg;
            end
            else begin
                CYImux = CImux;
            end
        end

        always @(*) begin
            POUT = PCOUT;
            case(OPMODEmux[1:0])
                2'b00: Xmux = 48'b0;
                2'b01: Xmux = Mmux;
                2'b10: Xmux = POUT;
                2'b11: Xmux = {Dmux[11:0],A1mux,B1mux};
            endcase

            case(OPMODEmux[3:2])
                2'b00: Zmux = 48'b0;
                2'b01: Zmux = PCIN;
                2'b10: Zmux = POUT;
                2'b11: Zmux = Cmux;
            endcase

            if (OPMODEmux[7]) begin
                {CYO,postAS} = Zmux - Xmux - CYImux;
            end
            else begin
                {CYO,postAS} = Zmux + Xmux + CYImux;
            end
        end

        always @(posedge CLK or posedge RSTCARRYIN) begin
```

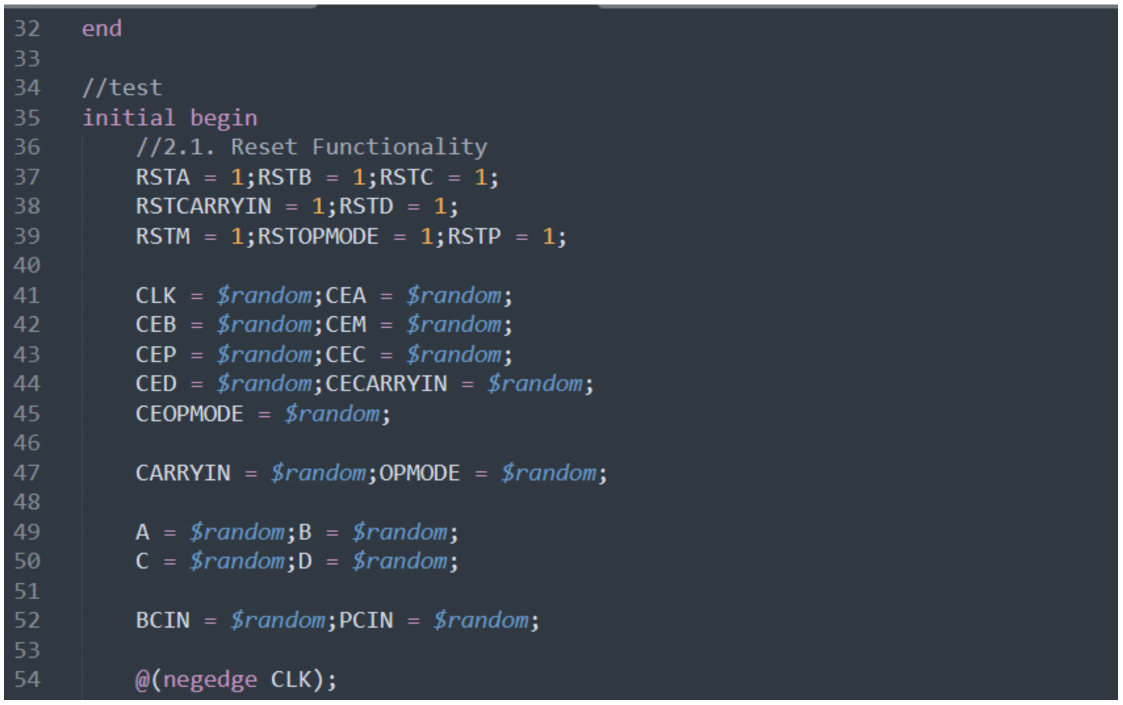


```verilog
                if (RSTCARRYIN) begin
                    CYOreg <= 0;
                end
                else if (CECARRYIN) begin
                    CYOreg <= CYO;
                end
            end
            always @(*) begin
                if (CARRYOUTREG) begin
                    CARRYOUT  = CYOreg;
                    CARRYOUTF = CYOreg;
                end
                else begin
                    CARRYOUT  = CYO;
                    CARRYOUTF = CYO;
                end
            end

            always @(posedge CLK or posedge RSTP) begin
                if (RSTP) begin
                    Preg <= 0;
                end
                else if (CEP) begin
                    Preg <= postAS;
                end
            end
            always @(*) begin
                if (PREG) begin
                    P  = Preg;
                    PCOUT  = Preg;
                end
                else begin
                    P  = postAS;
                    PCOUT = postAS;
                end
            end
        end
    endgenerate

endmodule
```

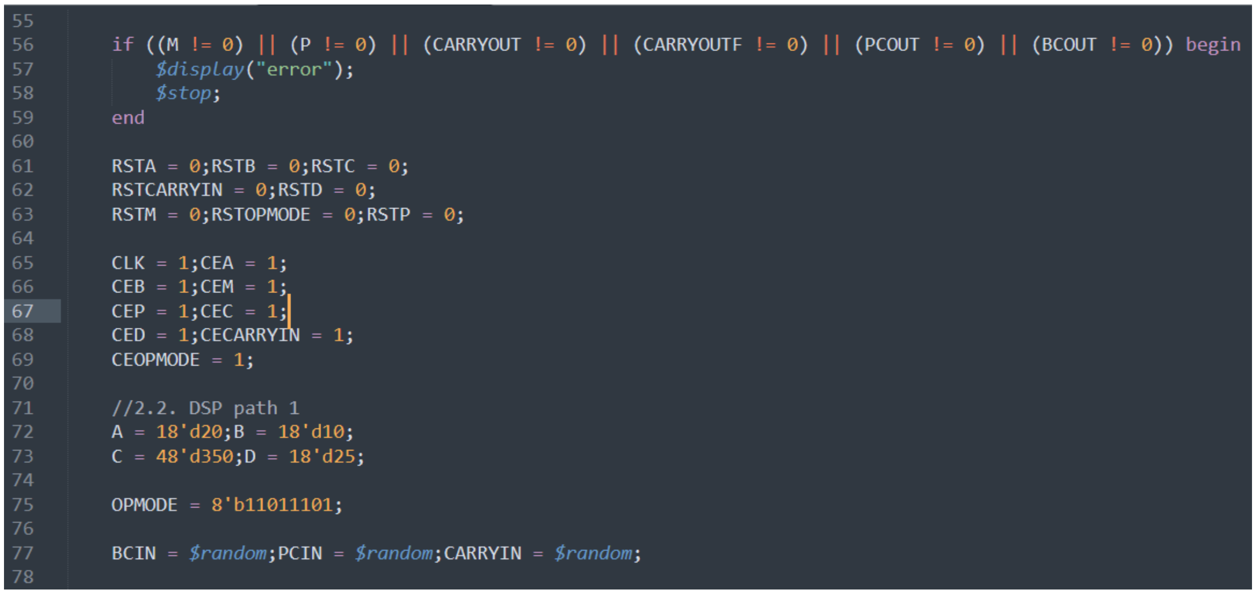
2. Testbench code



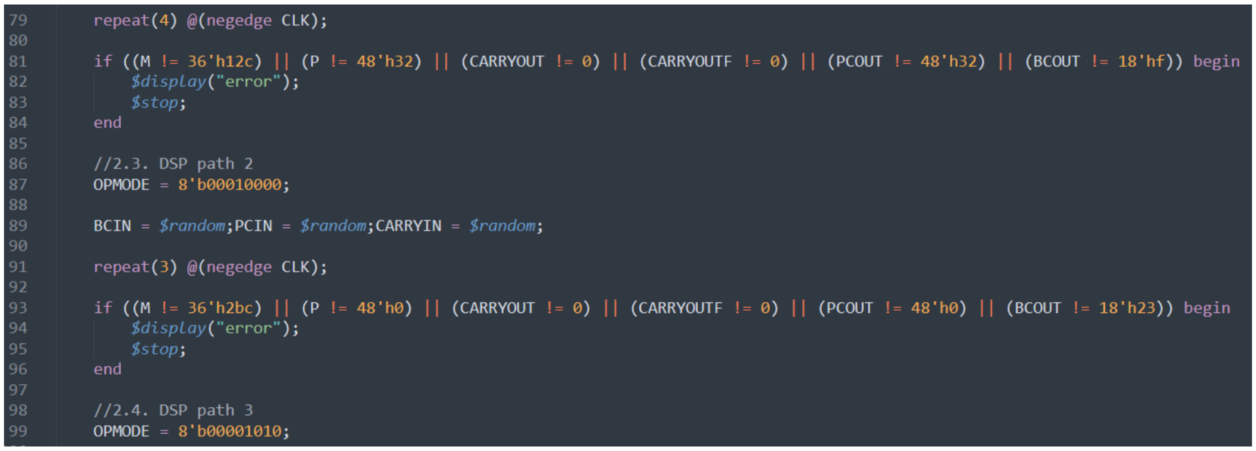
```verilog
module DSP48A1_TB();

reg CLK,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE;
reg RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP;
reg CARRYIN;
reg [7:0] OPMODE;

reg [17:0] A,B,D;
reg [47:0] C;

reg [17:0] BCIN;
reg [47:0] PCIN;

wire [35:0] M;
wire [47:0] P;

wire CARRYOUT,CARRYOUTF;

wire [17:0] BCOUT;
wire [47:0] PCOUT;

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

//clk generation
initial begin
    CLK = 0;
    forever
        #1 CLK = ~CLK;
```



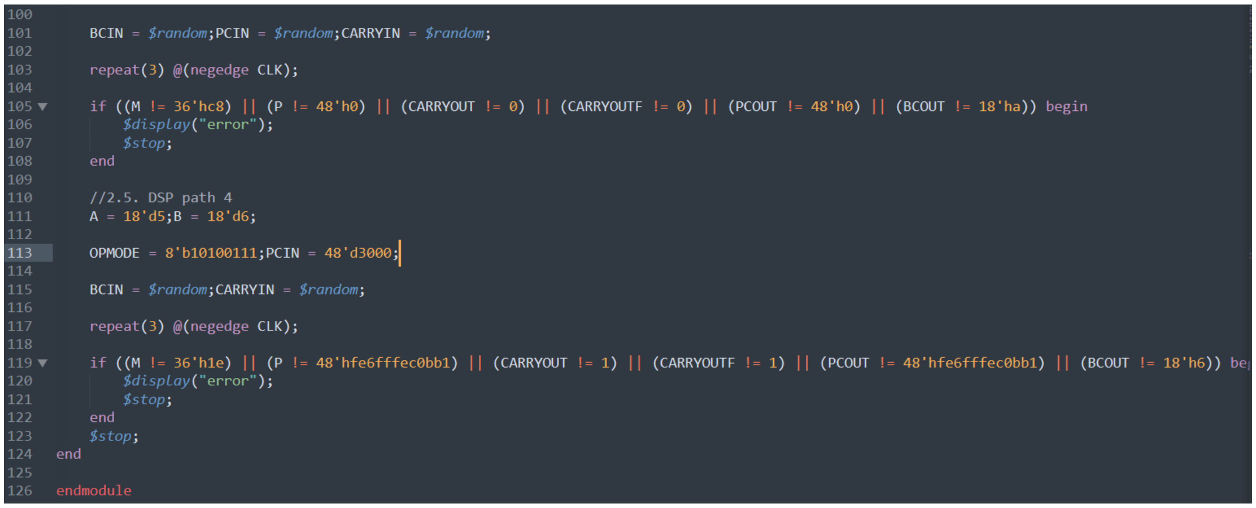
```verilog
end

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

    CLK = $random;CEA = $random;
    CEB = $random;CEM = $random;
    CEP = $random;CEC = $random;
    CED = $random;CECARRYIN = $random;
    CEOPMODE = $random;

    CARRYIN = $random;OPMODE = $random;

    A = $random;B = $random;
    C = $random;D = $random;

    BCIN = $random;PCIN = $random;

    @(negedge CLK);
```



```
    if ((M != 0) || (P != 0) || (CARRYOUT != 0) || (CARRYOUTF != 0) || (PCOUT != 0) || (BCOUT != 0)) begin
        $display("error");
        $stop;
    end

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

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

    //2.2. DSP path 1
    A = 18'd20;B = 18'd10;
    C = 48'd350;D = 18'd25;

    OPMODE = 8'b11011101;

    BCIN = $random;PCIN = $random;CARRYIN = $random;

```

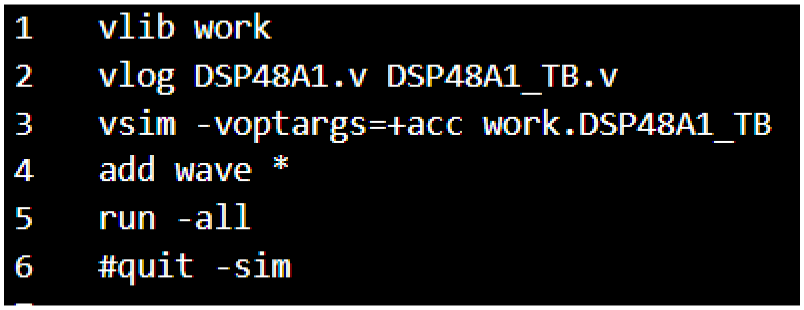


```
    repeat(4) @(negedge CLK);

    if ((M != 36'h12c) || (P != 48'h32) || (CARRYOUT != 0) || (CARRYOUTF != 0) || (PCOUT != 48'h32) || (BCOUT != 18'hf)) begin
        $display("error");
        $stop;
    end

    //2.3. DSP path 2
    OPMODE = 8'b00010000;

    BCIN = $random;PCIN = $random;CARRYIN = $random;

    repeat(3) @(negedge CLK);

    if ((M != 36'h2bc) || (P != 48'h0) || (CARRYOUT != 0) || (CARRYOUTF != 0) || (PCOUT != 48'h0) || (BCOUT != 18'h23)) begin
        $display("error");
        $stop;
    end

    //2.4. DSP path 3
    OPMODE = 8'b00001010;
```



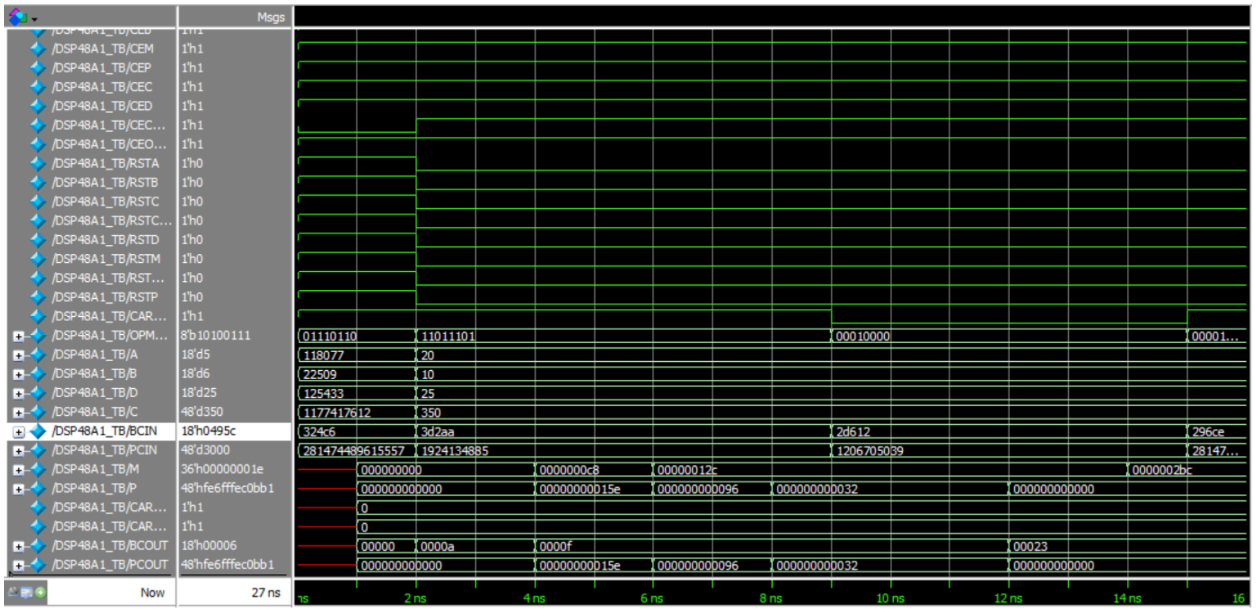
```

    BCIN = $random;PCIN = $random;CARRYIN = $random;

    repeat(3) @(negedge CLK);

    if ((M != 36'hc8) || (P != 48'h0) || (CARRYOUT != 0) || (CARRYOUTF != 0) || (PCOUT != 48'h0) || (BCOUT != 18'ha)) begin
        $display("error");
        $stop;
    end

    //2.5. DSP path 4
    A = 18'd5;B = 18'd6;

    OPMODE = 8'b10100111;PCIN = 48'd3000;

    BCIN = $random;CARRYIN = $random;

    repeat(3) @(negedge CLK);

    if ((M != 36'h1e) || (P != 48'hfe6fffec0bb1) || (CARRYOUT != 1) || (CARRYOUTF != 1) || (PCOUT != 48'hfe6fffec0bb1) || (BCOUT != 18'h6)) be
        $display("error");
        $stop;
    end
    $stop;
end

endmodule
```

3. Do file

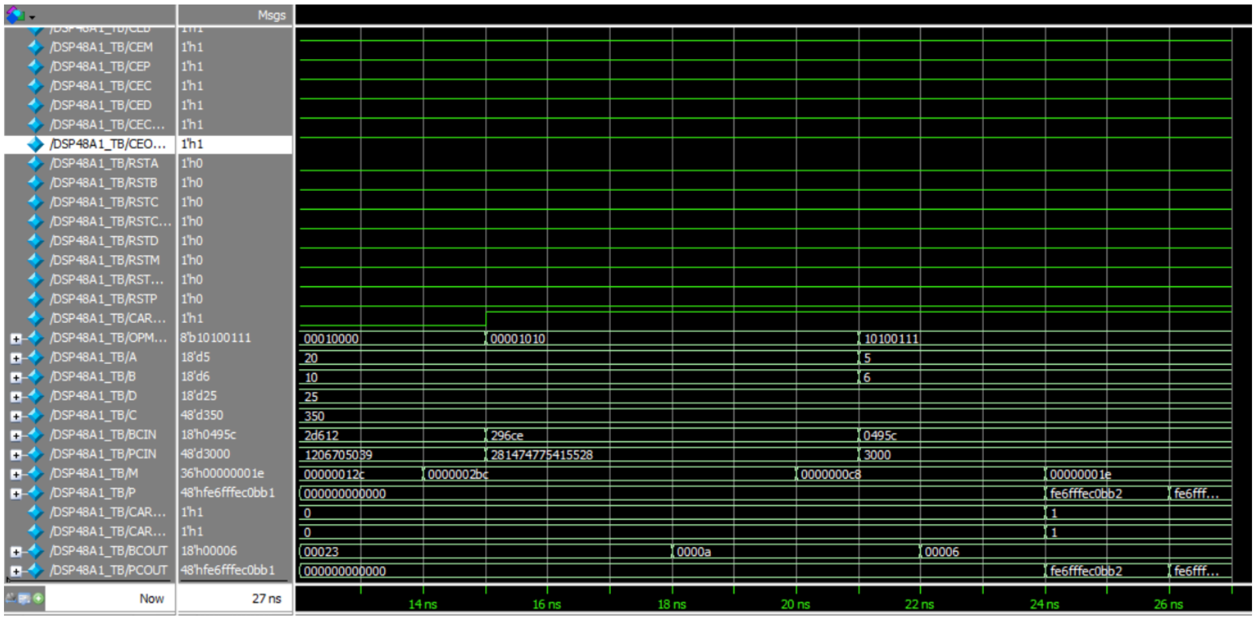


```
vlib work
vlog DSP48A1.v DSP48A1_TB.v
vsim -voptargs=+acc work.DSP48A1_TB
add wave *
run -all
#quit -sim
```

4. QuestaSim Snippets

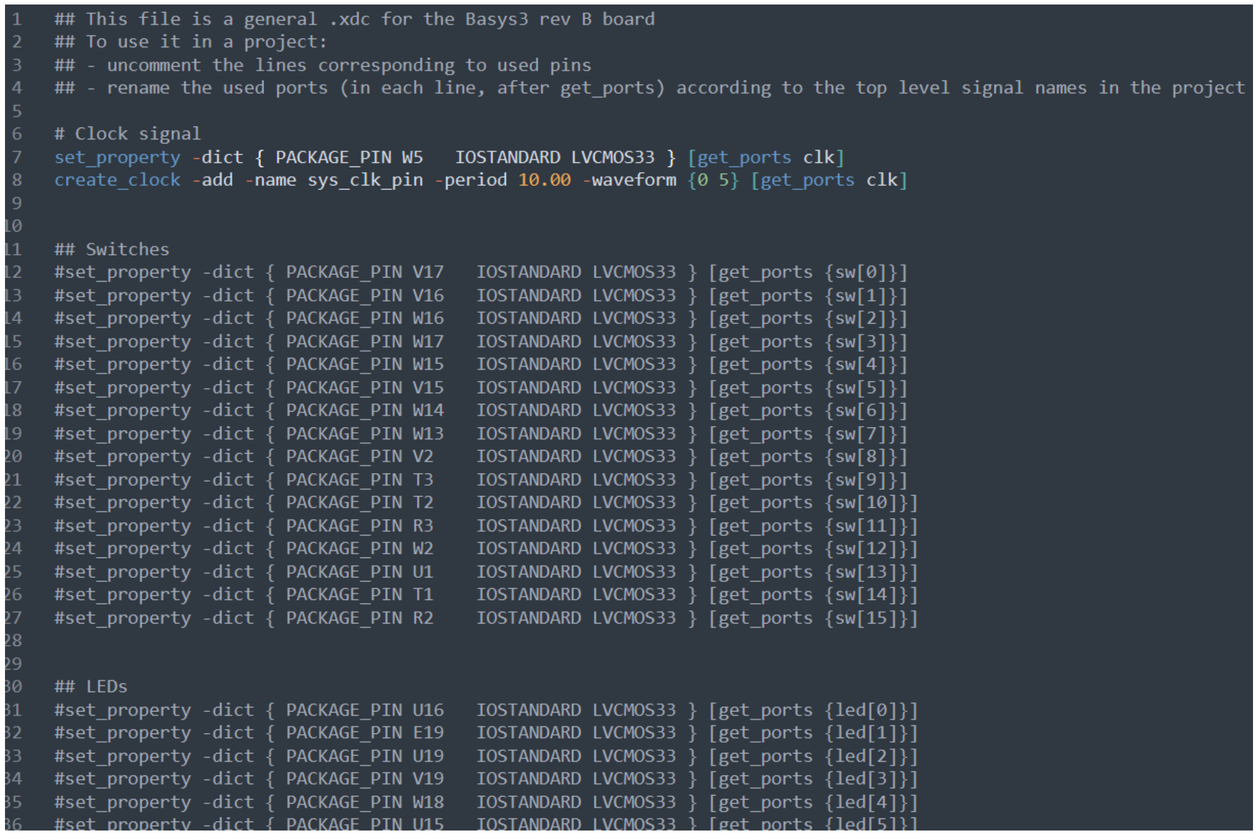


```
                          Msgs
/DSP48A1_TB/CEM           1'h1
/DSP48A1_TB/CEP           1'h1
/DSP48A1_TB/CEC           1'h1
/DSP48A1_TB/CED           1'h1
/DSP48A1_TB/CEC...        1'h1
/DSP48A1_TB/CEO...        1'h1
/DSP48A1_TB/RSTA          1'h0
/DSP48A1_TB/RSTB          1'h0
/DSP48A1_TB/RSTC          1'h0
/DSP48A1_TB/RSTC...       1'h0
/DSP48A1_TB/RSTD          1'h0
/DSP48A1_TB/RSTM          1'h0
/DSP48A1_TB/RST...        1'h0
/DSP48A1_TB/RSTP          1'h0
/DSP48A1_TB/CAR...        1'h1
/DSP48A1_TB/OPM...        8'b10100111
/DSP48A1_TB/A             18'd5
/DSP48A1_TB/B             18'd6
/DSP48A1_TB/D             18'd25
/DSP48A1_TB/C             48'd350
/DSP48A1_TB/BCIN          18'h0495c
/DSP48A1_TB/PCIN          48'd3000
/DSP48A1_TB/M             36'h00000001e
/DSP48A1_TB/P             48'hfe6fffec0bb1
/DSP48A1_TB/CAR...        1'h1
/DSP48A1_TB/CAR...        1'h1
/DSP48A1_TB/BCOUT         18'h00006
/DSP48A1_TB/PCOUT         48'hfe6fffec0bb1
Now                       27 ns
```



```
                          Msgs
/DSP48A1_TB/CEM           1'h1
/DSP48A1_TB/CEP           1'h1
/DSP48A1_TB/CEC           1'h1
/DSP48A1_TB/CED           1'h1
/DSP48A1_TB/CEC...        1'h1
/DSP48A1_TB/CEO...        1'h1
/DSP48A1_TB/RSTA          1'h0
/DSP48A1_TB/RSTB          1'h0
/DSP48A1_TB/RSTC          1'h0
/DSP48A1_TB/RSTC...       1'h0
/DSP48A1_TB/RSTD          1'h0
/DSP48A1_TB/RSTM          1'h0
/DSP48A1_TB/RST...        1'h0
/DSP48A1_TB/RSTP          1'h0
/DSP48A1_TB/CAR...        1'h1
/DSP48A1_TB/OPM...        8'b10100111
/DSP48A1_TB/A             18'd5
/DSP48A1_TB/B             18'd6
/DSP48A1_TB/D             18'd25
/DSP48A1_TB/C             48'd350
/DSP48A1_TB/BCIN          18'h0495c
/DSP48A1_TB/PCIN          48'd3000
/DSP48A1_TB/M             36'h00000001e
/DSP48A1_TB/P             48'hfe6fffec0bb1
/DSP48A1_TB/CAR...        1'h1
/DSP48A1_TB/CAR...        1'h1
/DSP48A1_TB/BCOUT         18'h00006
/DSP48A1_TB/PCOUT         48'hfe6fffec0bb1
Now                       27 ns
```

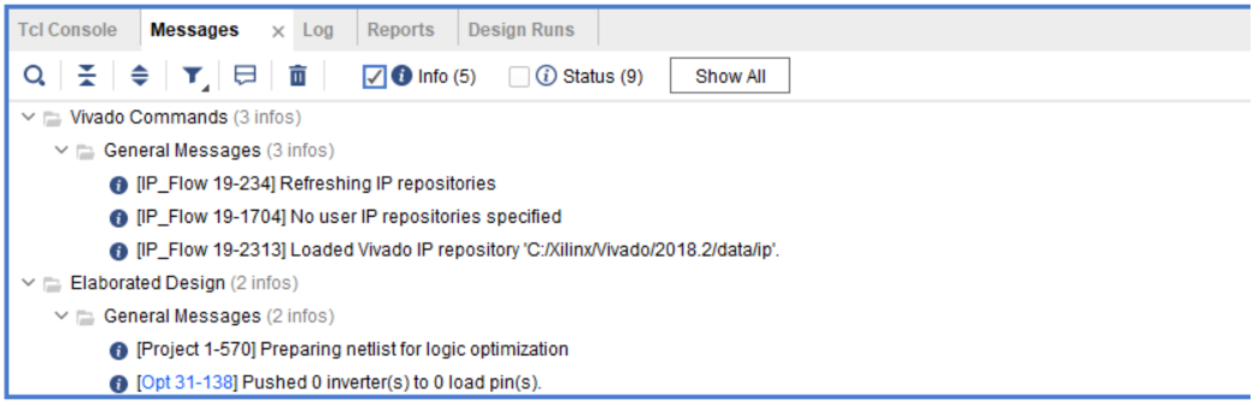
5. Constraint File



```
## 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]}]
#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]}]
```

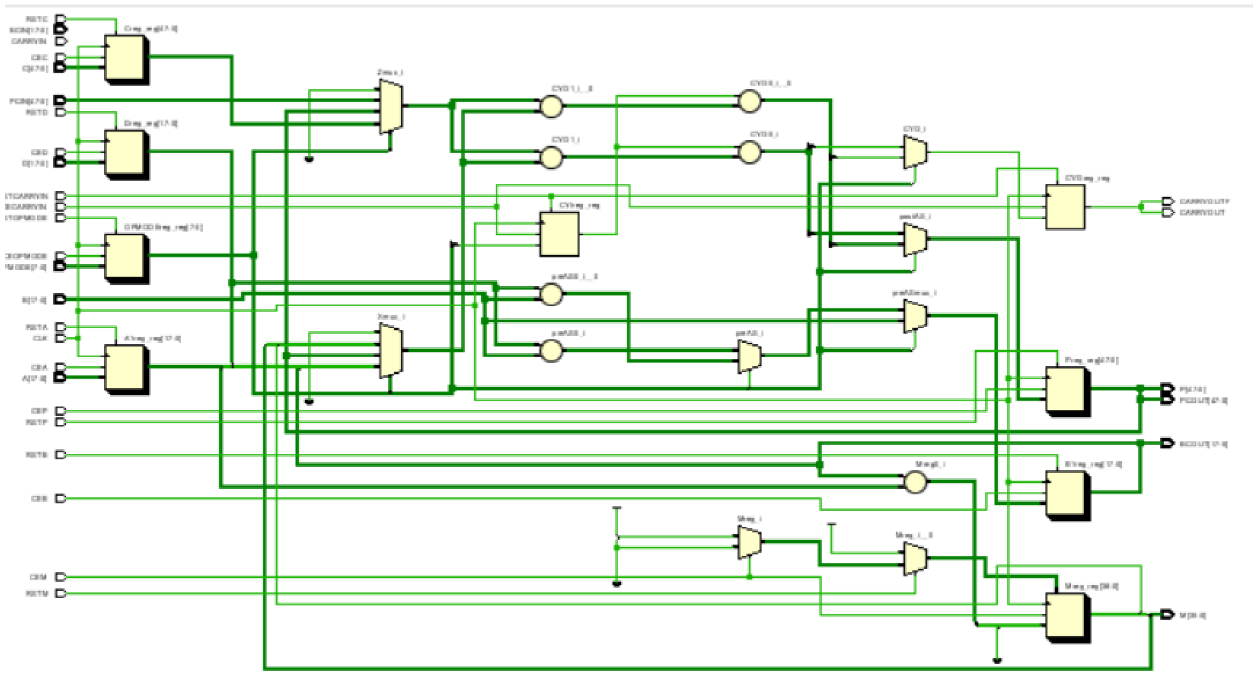
6. Elaboration ("Messages" tab & Schematic snippets)

- **Messages**



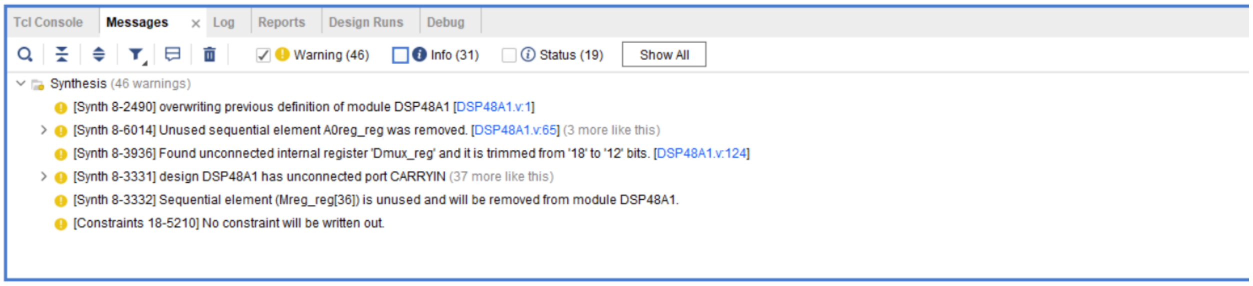
```
Vivado Commands (3 infos)
  General Messages (3 infos)
    [IP_Flow 19-234] Refreshing IP repositories
    [IP_Flow 19-1704] No user IP repositories specified
    [IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2018.2/data/ip'.
Elaborated Design (2 infos)
  General Messages (2 infos)
    [Project 1-570] Preparing netlist for logic optimization
    [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
```

- **Schematic**



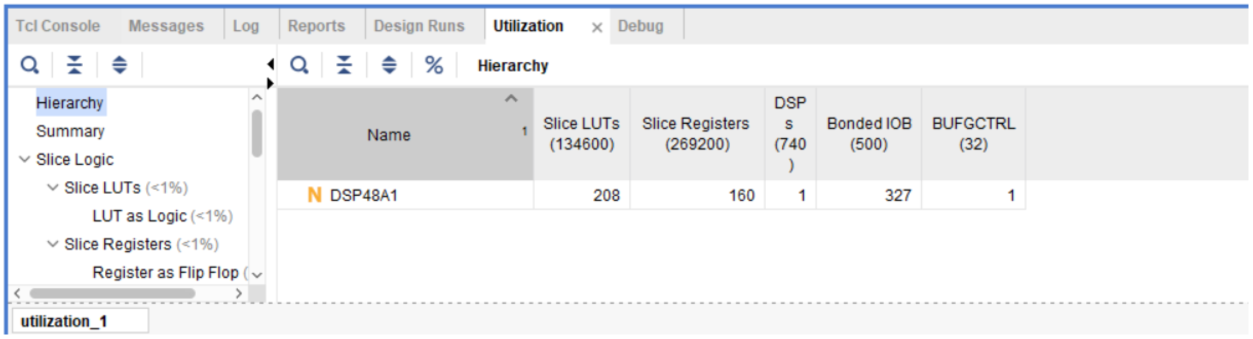
7. Synthesis ("Messages" tab, Utilization report, timing report & Schematic snippets)

- **Messages**



Synthesis (46 warnings)
[Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
[Synth 8-6014] Unused sequential element A0reg_reg was removed. [DSP48A1.v:65] (3 more like this)
[Synth 8-3936] Found unconnected internal register 'Dmux_reg' and it is trimmed from '18' to '12' bits. [DSP48A1.v:124]
[Synth 8-3331] design DSP48A1 has unconnected port CARRYIN (37 more like this)
[Synth 8-3332] Sequential element (Mreg_reg[36]) is unused and will be removed from module DSP48A1.
[Constraints 18-5210] No constraint will be written out.

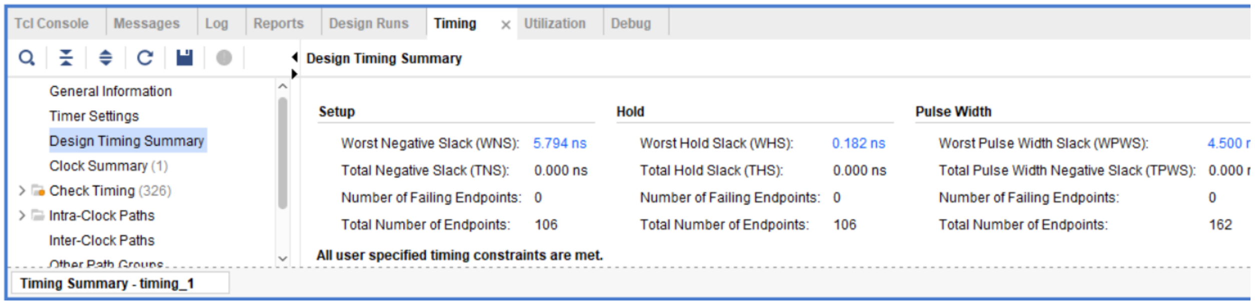
- **Utilization report**



Hierarchy

Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP48A1	208	160	1	327	1

- **Timing report**

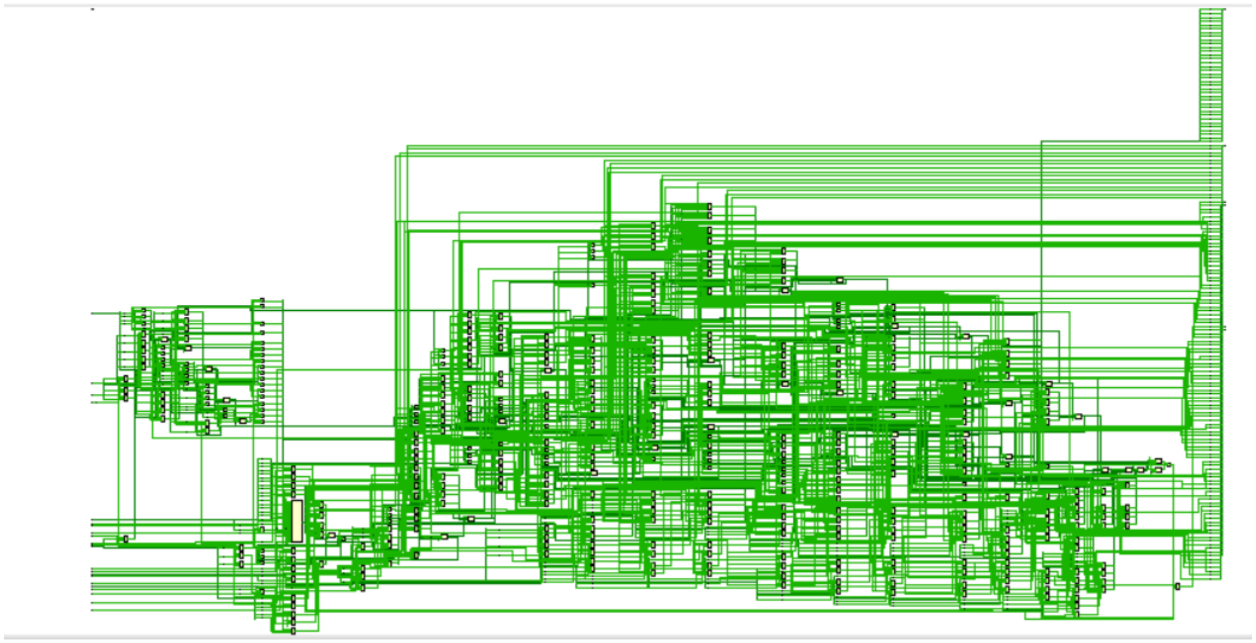


Design Timing Summary

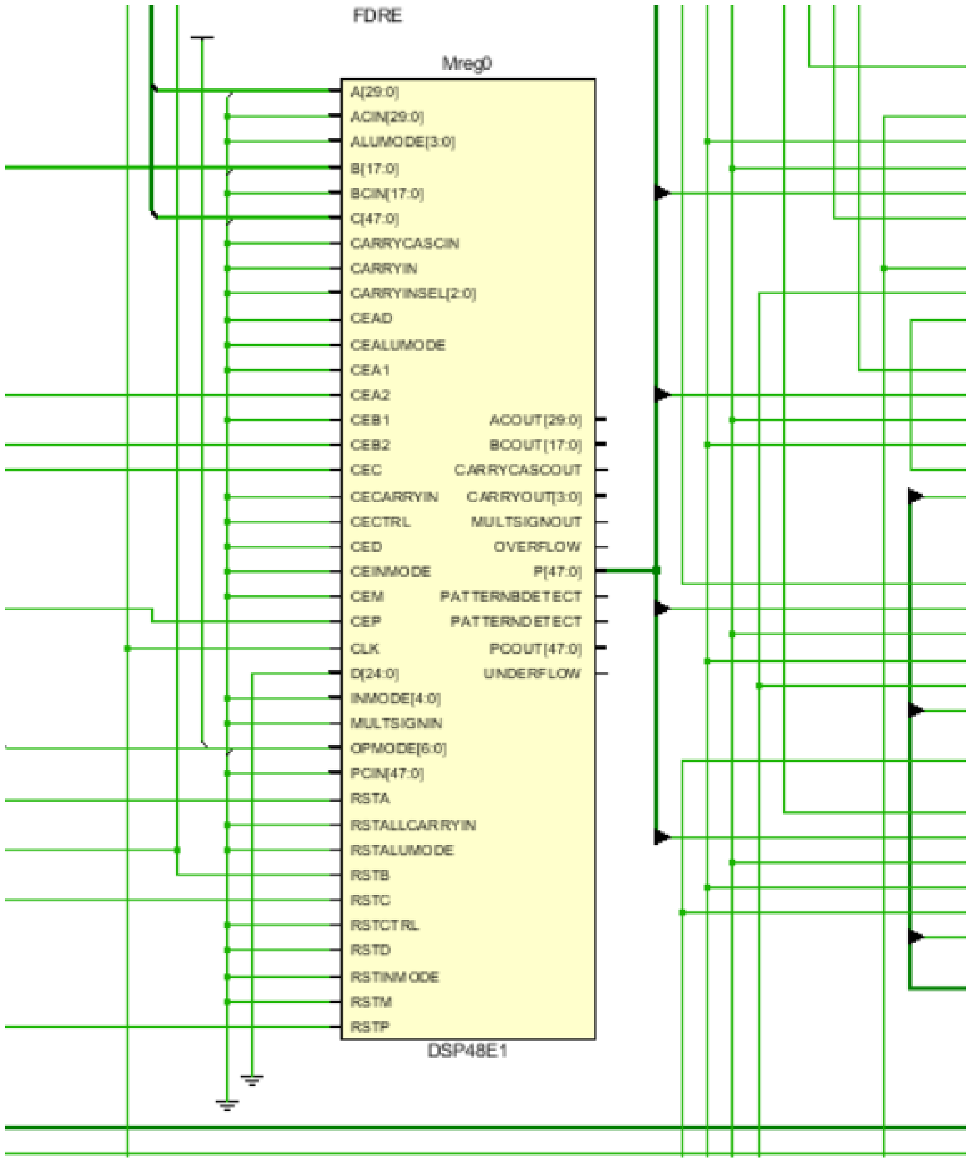
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.794 ns	Worst Hold Slack (WHS):	0.182 ns	Worst Pulse Width Slack (WPWS):	4.500 r
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 r
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	106	Total Number of Endpoints:	106	Total Number of Endpoints:	162

All user specified timing constraints are met.

- **Schematic**

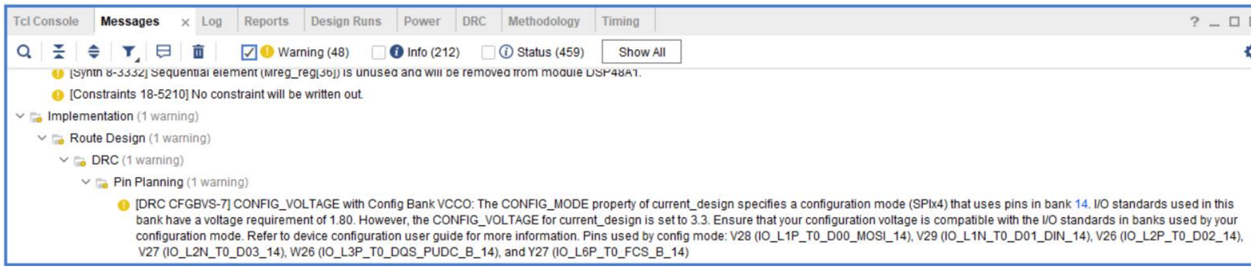


- **DSP block**


FDRE
Mreg0
A[29:0]
ACIN[29:0]
ALUMODE[3:0]
B[17:0]
BCIN[17:0]
C[47:0]
CARRYCASCIN
CARRYIN
CARRYINSEL[2:0]
CEAD
CEALUMODE
CEA1
CEA2
CEB1
CEB2
CEC
CECARRYIN
CECTRL
CED
CEINMODE
CEM
CEP
CLK
D[24:0]
INMODE[4:0]
MULTSIGNIN
OPMODE[6:0]
PCIN[47:0]
RSTA
RSTALLCARRYIN
RSTALUMODE
RSTB
RSTC
RSTCTRL
RSTD
RSTINMODE
RSTM
RSTP
ACOUT[29:0]
BCOUT[17:0]
CARRYCASCOUT
CARRYOUT[3:0]
MULTSIGNOUT
OVERFLOW
P[47:0]
PATTERNBDETECT
PATTERNDETECT
PCOUT[47:0]
UNDERFLOW
DSP48E1

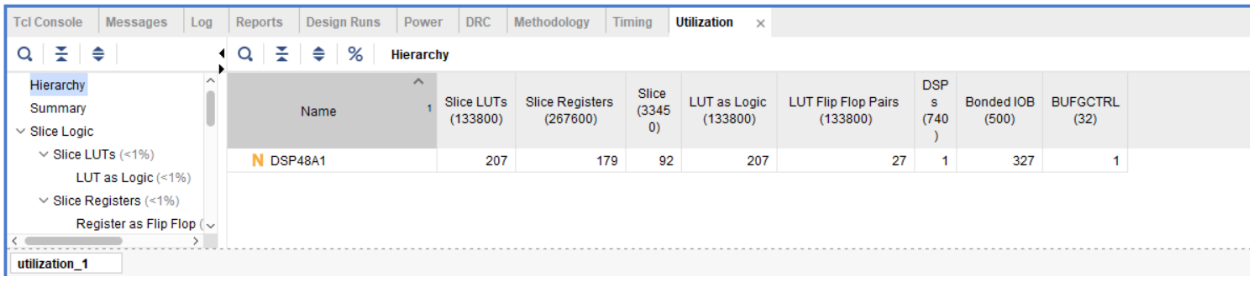
8. Implementation ("Messages" tab, Utilization report, timing report & device snippets)

- **Messages**



```
[Synth 8-3332] Sequential element (Mreg_reg[36]) is unused and will be removed from module DSP48A1.
[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)
```

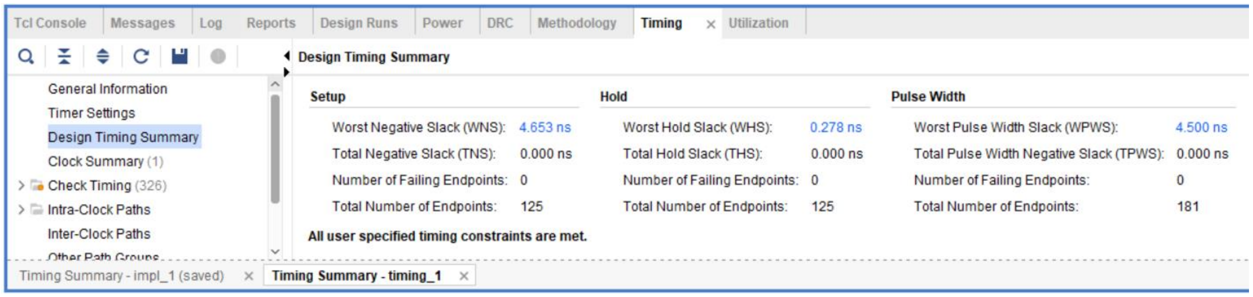
- **Utilization report**



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)
N DSP48A1	207	179	92	207	27	1	327	1

- **Timing report**

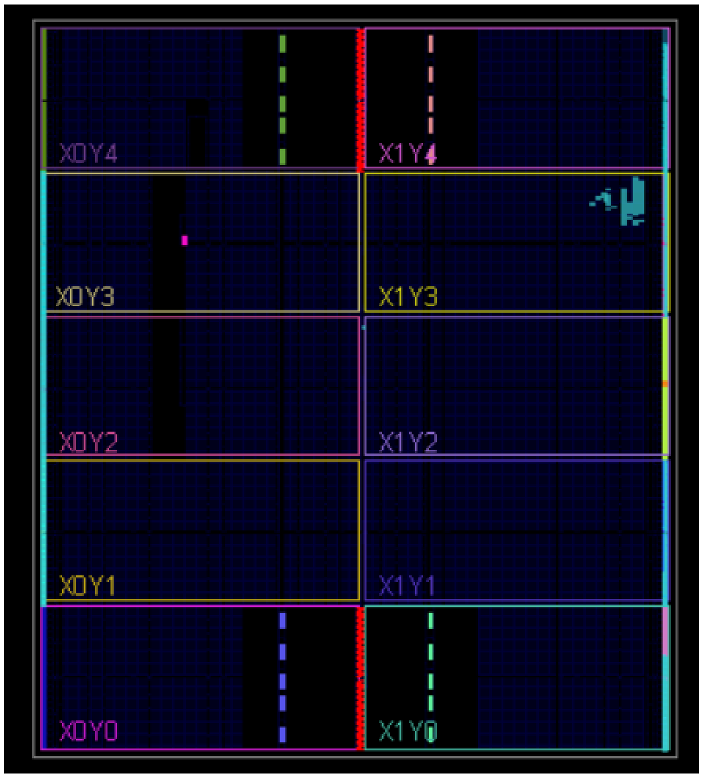


Design Timing Summary

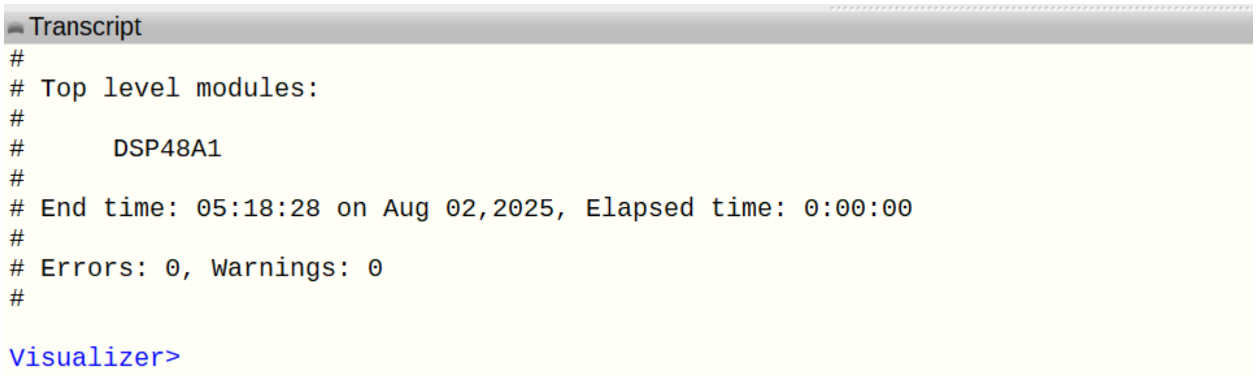
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.653 ns	Worst Hold Slack (WHS):	0.278 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	125	Total Number of Endpoints:	125	Total Number of Endpoints:	181

All user specified timing constraints are met.

- **Device**



9. **Linting (snippets showing no errors)**



```
#
# Top level modules:
#
#      DSP48A1
#
# End time: 05:18:28 on Aug 02,2025, Elapsed time: 0:00:00
#
# Errors: 0, Warnings: 0
#

Visualizer>
```