Digital Design Diploma

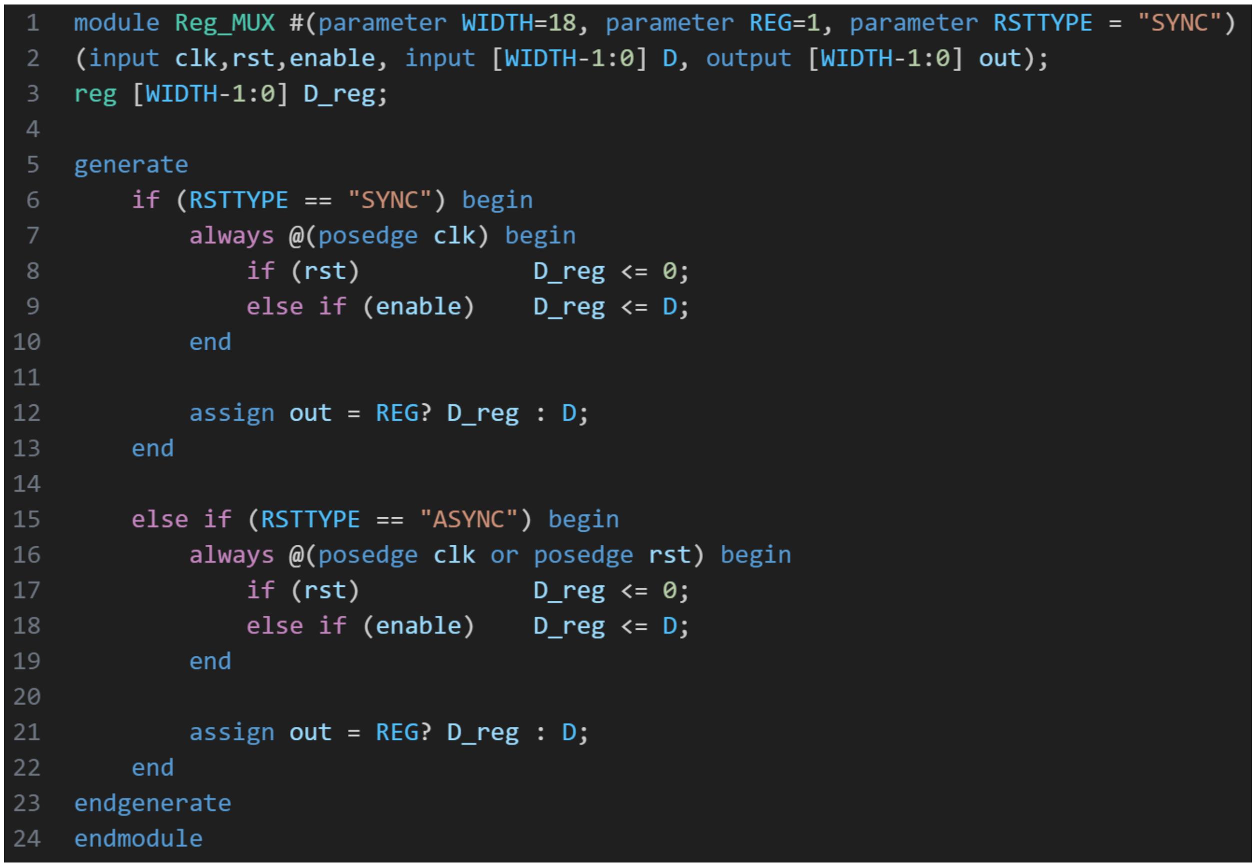
Project 1 : DSP48A1

Name	Kareem Hassan Atif
Group	G2

Submitted to: Eng. Kareem Waseem

Contents:

1) Register with MUX:



```verilog
module Reg_MUX #(parameter WIDTH=18, parameter REG=1, parameter RSTTYPE = "SYNC")
(input clk,rst,enable, input [WIDTH-1:0] D, output [WIDTH-1:0] out);
reg [WIDTH-1:0] D_reg;

generate
    if (RSTTYPE == "SYNC") begin
        always @(posedge clk) begin
            if (rst)            D_reg <= 0;
            else if (enable)    D_reg <= D;
        end

        assign out = REG? D_reg : D;
    end

    else if (RSTTYPE == "ASYNC") begin
        always @(posedge clk or posedge rst) begin
            if (rst)            D_reg <= 0;
            else if (enable)    D_reg <= D;
        end

        assign out = REG? D_reg : D;
    end
endgenerate
endmodule
```

Figure 1: Reg_MUX Code

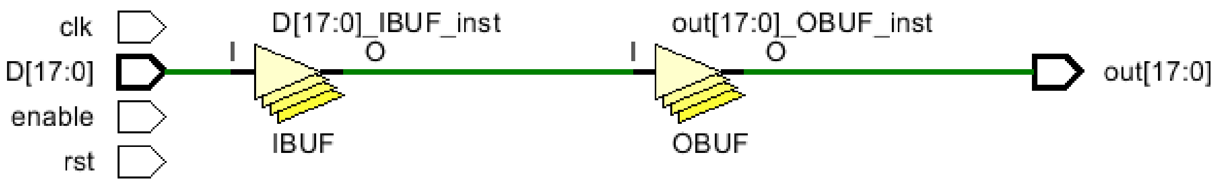


Figure 2: Reg_MUX RTL (Select=0)

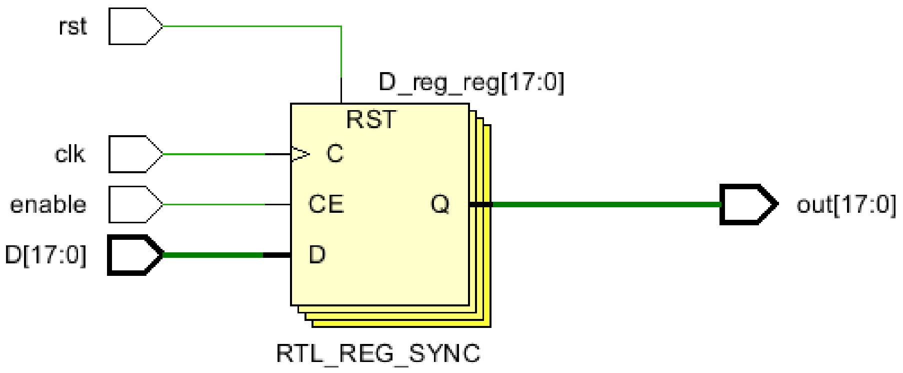
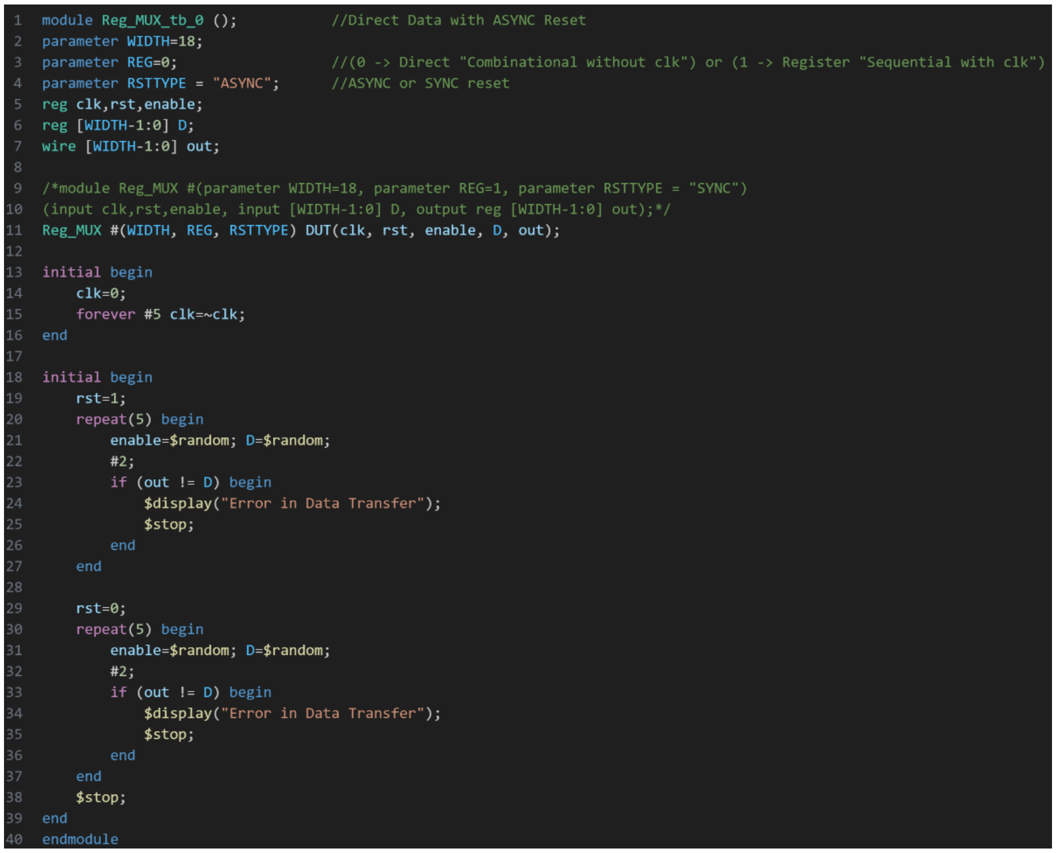
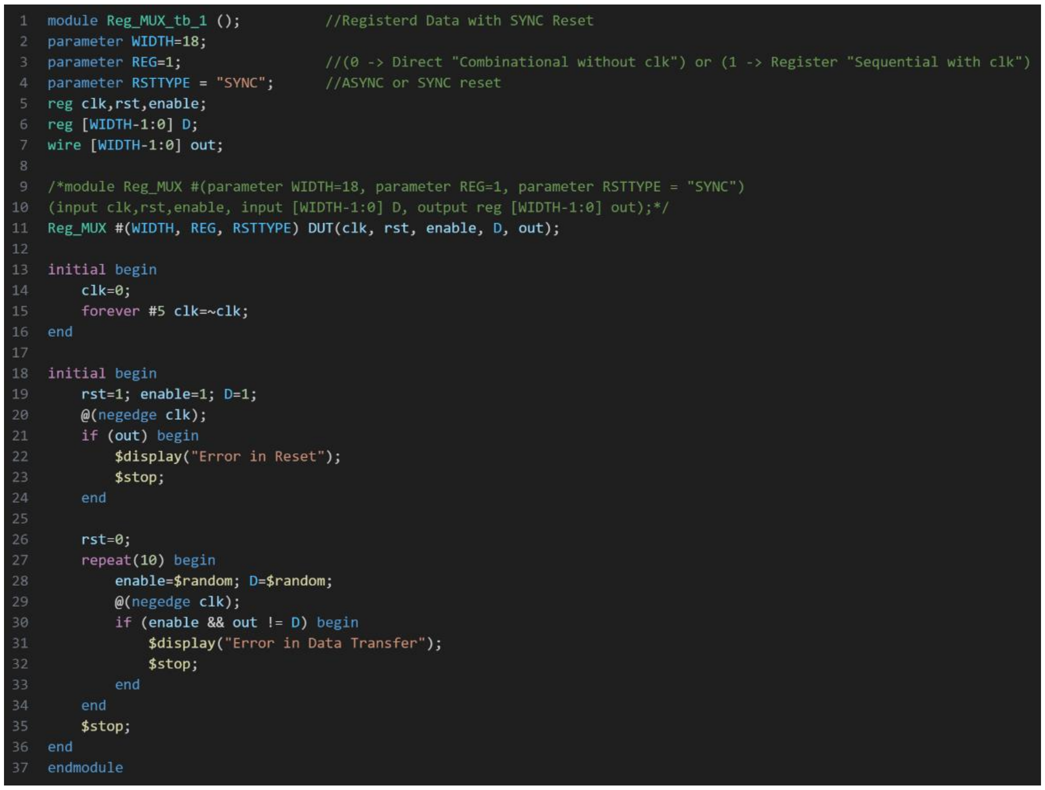


Figure 3: Reg_MUX RTL (Select=1)



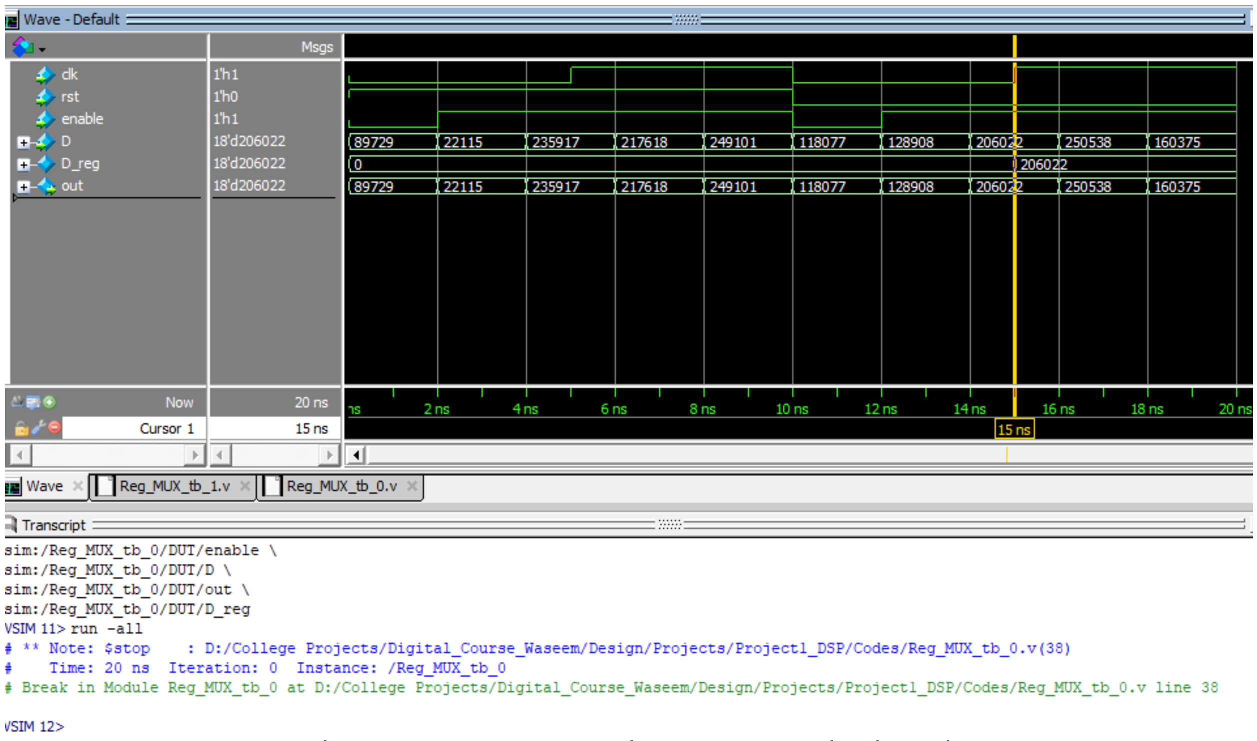
```verilog
module Reg_MUX_tb_0 ();           //Direct Data with ASYNC Reset
parameter WIDTH=18;
parameter REG=0;                  //(0 -> Direct "Combinational without clk") or (1 -> Register "Sequential with clk")
parameter RSTTYPE = "ASYNC";      //ASYNC or SYNC reset
reg clk,rst,enable;
reg [WIDTH-1:0] D;
wire [WIDTH-1:0] out;

/*module Reg_MUX #(parameter WIDTH=18, parameter REG=1, parameter RSTTYPE = "SYNC")
(input clk,rst,enable, input [WIDTH-1:0] D, output reg [WIDTH-1:0] out);*/
Reg_MUX #(WIDTH, REG, RSTTYPE) DUT(clk, rst, enable, D, out);

initial begin
    clk=0;
    forever #5 clk=~clk;
end

initial begin
    rst=1;
    repeat(5) begin
        enable=$random; D=$random;
        #2;
        if (out != D) begin
            $display("Error in Data Transfer");
            $stop;
        end
    end

    rst=0;
    repeat(5) begin
        enable=$random; D=$random;
        #2;
        if (out != D) begin
            $display("Error in Data Transfer");
            $stop;
        end
    end
    $stop;
end
endmodule
```

Figure 4: Reg_MUX Testbench (Async Reset and Select=0)



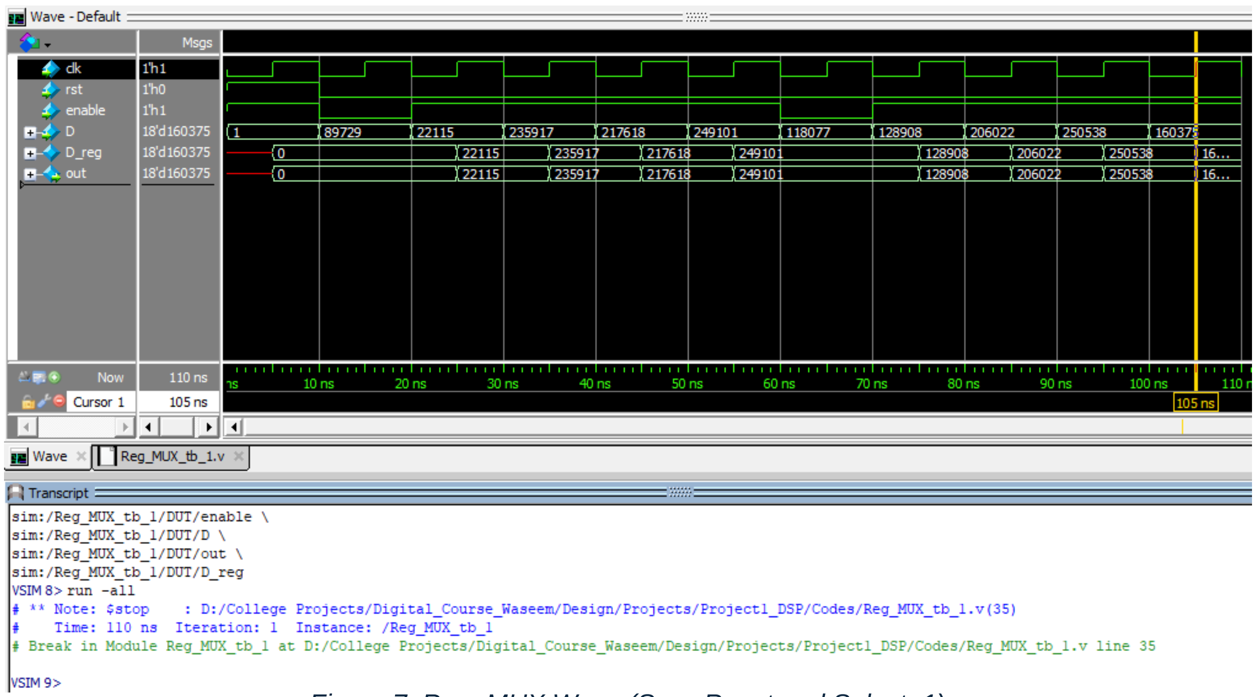
```verilog
module Reg_MUX_tb_1 ();           //Registerd Data with SYNC Reset
parameter WIDTH=18;
parameter REG=1;                  //(0 -> Direct "Combinational without clk") or (1 -> Register "Sequential with clk")
parameter RSTTYPE = "SYNC";       //ASYNC or SYNC reset
reg clk,rst,enable;
reg [WIDTH-1:0] D;
wire [WIDTH-1:0] out;

/*module Reg_MUX #(parameter WIDTH=18, parameter REG=1, parameter RSTTYPE = "SYNC")
(input clk,rst,enable, input [WIDTH-1:0] D, output reg [WIDTH-1:0] out);*/
Reg_MUX #(WIDTH, REG, RSTTYPE) DUT(clk, rst, enable, D, out);

initial begin
    clk=0;
    forever #5 clk=~clk;
end

initial begin
    rst=1; enable=1; D=1;
    @(negedge clk);
    if (out) begin
        $display("Error in Reset");
        $stop;
    end

    rst=0;
    repeat(10) begin
        enable=$random; D=$random;
        @(negedge clk);
        if (enable && out != D) begin
            $display("Error in Data Transfer");
            $stop;
        end
    end
    $stop;
end
endmodule
```

Figure 5: Reg_MUX Testbench (Sync Reset and Select=1)



```
sim:/Reg_MUX_tb_0/DUT/enable \
sim:/Reg_MUX_tb_0/DUT/D \
sim:/Reg_MUX_tb_0/DUT/out \
sim:/Reg_MUX_tb_0/DUT/D_reg
VSIM 11> run -all
# ** Note: $stop    : D:/College Projects/Digital_Course_Waseem/Design/Projects/Project1_DSP/Codes/Reg_MUX_tb_0.v(38)
#    Time: 20 ns  Iteration: 0  Instance: /Reg_MUX_tb_0
# Break in Module Reg_MUX_tb_0 at D:/College Projects/Digital_Course_Waseem/Design/Projects/Project1_DSP/Codes/Reg_MUX_tb_0.v line 38

VSIM 12>
```

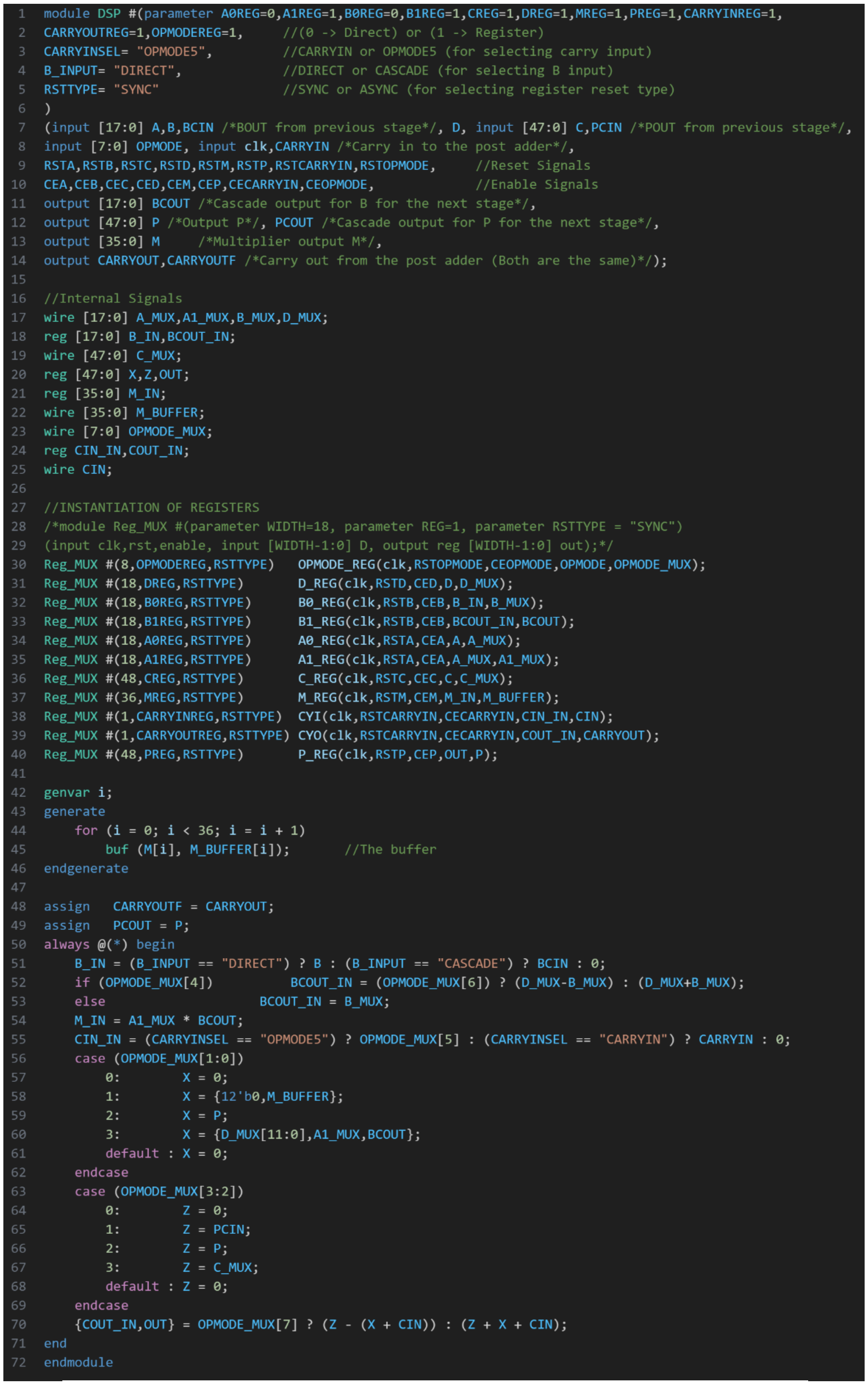
Figure 6: Reg_MUX Wave (Async Reset and Select=0)



```
sim:/Reg_MUX_tb_1/DUT/enable \
sim:/Reg_MUX_tb_1/DUT/D \
sim:/Reg_MUX_tb_1/DUT/out \
sim:/Reg_MUX_tb_1/DUT/D_reg
VSIM 8> run -all
# ** Note: $stop    : D:/College Projects/Digital_Course_Waseem/Design/Projects/Project1_DSP/Codes/Reg_MUX_tb_1.v(35)
#    Time: 110 ns  Iteration: 1  Instance: /Reg_MUX_tb_1
# Break in Module Reg_MUX_tb_1 at D:/College Projects/Digital_Course_Waseem/Design/Projects/Project1_DSP/Codes/Reg_MUX_tb_1.v line 35

VSIM 9>
```

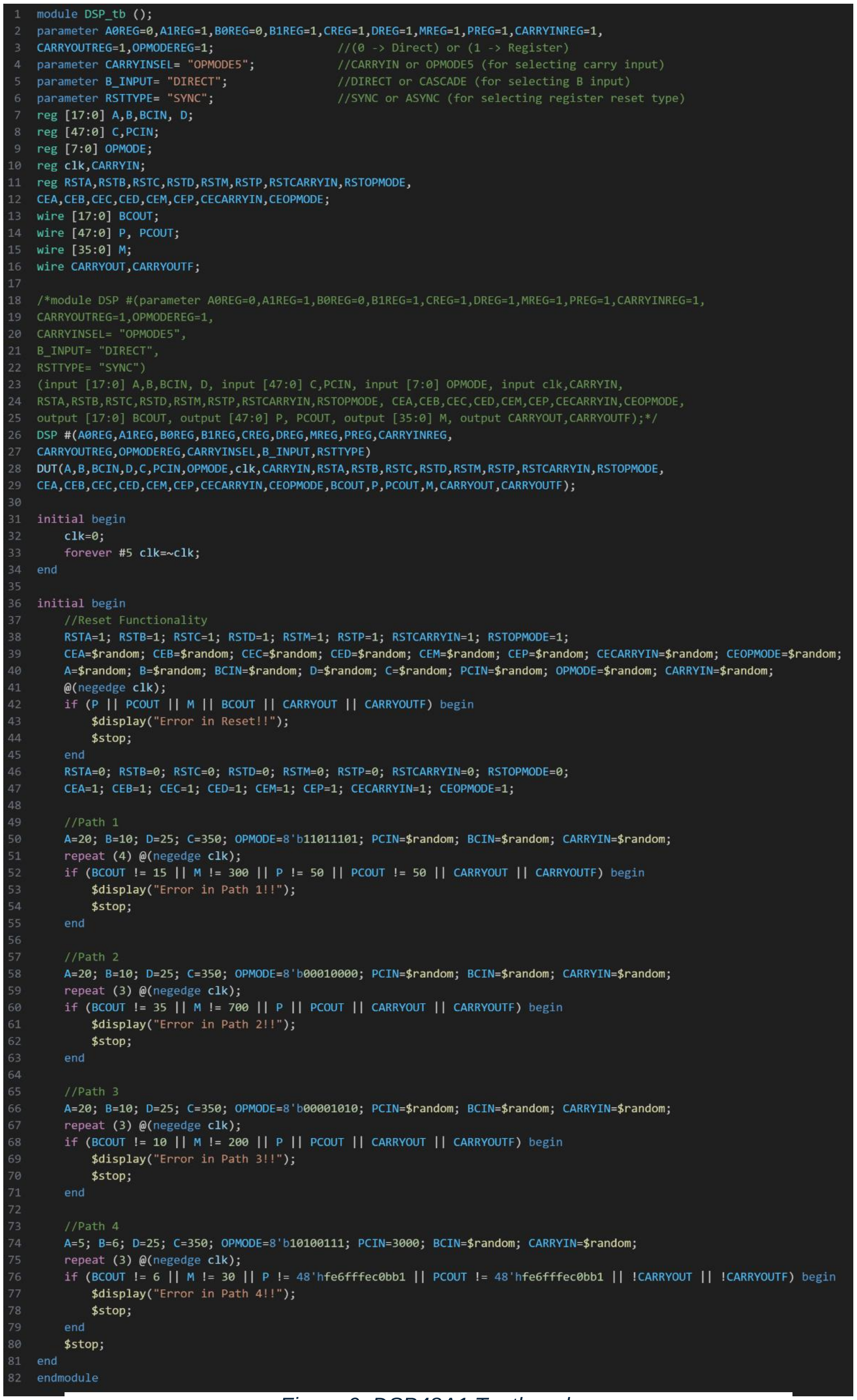
Figure 7: Reg_MUX Wave (Sync Reset and Select=1)

2) DSP48A1:



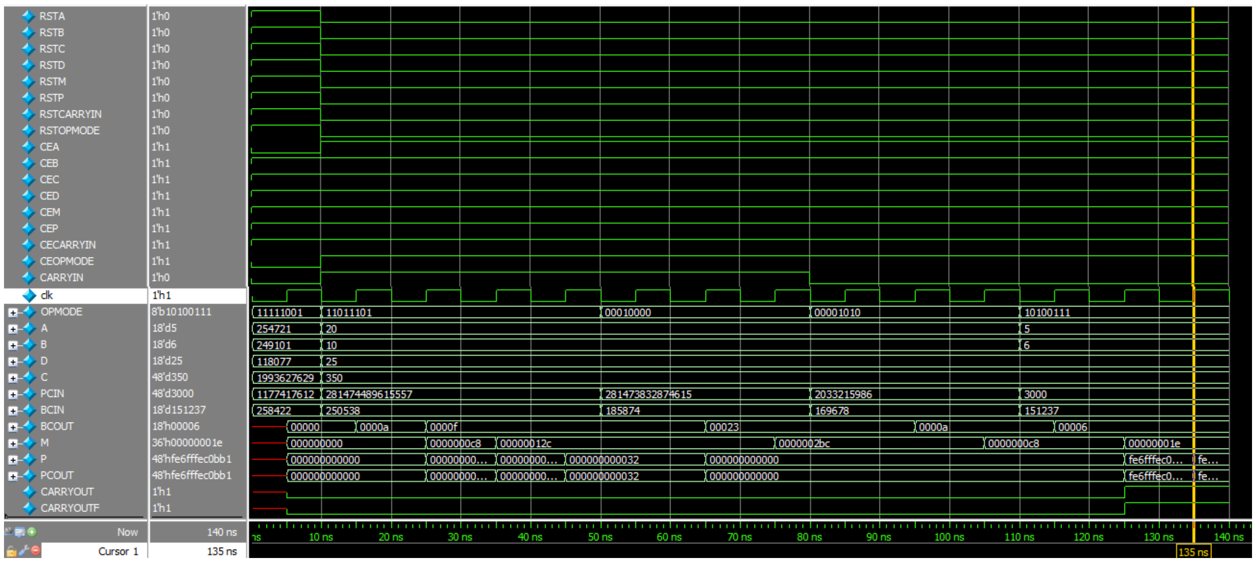
```verilog
module DSP #(parameter A0REG=0,A1REG=1,B0REG=0,B1REG=1,CREG=1,DREG=1,MREG=1,PREG=1,CARRYINREG=1,
CARRYOUTREG=1,OPMODEREG=1,     //(0 -> Direct) or (1 -> Register)
CARRYINSEL= "OPMODE5",         //CARRYIN or OPMODE5 (for selecting carry input)
B_INPUT= "DIRECT",             //DIRECT or CASCADE (for selecting B input)
RSTTYPE= "SYNC"                //SYNC or ASYNC (for selecting register reset type)
)
(input [17:0] A,B,BCIN /*BOUT from previous stage*/, D, input [47:0] C,PCIN /*POUT from previous stage*/,
input [7:0] OPMODE, input clk,CARRYIN /*Carry in to the post adder*/,
RSTA,RSTB,RSTC,RSTD,RSTM,RSTP,RSTCARRYIN,RSTOPMODE,     //Reset Signals
CEA,CEB,CEC,CED,CEM,CEP,CECARRYIN,CEOPMODE,             //Enable Signals
output [17:0] BCOUT /*Cascade output for B for the next stage*/,
output [47:0] P /*Output P*/, PCOUT /*Cascade output for P for the next stage*/,
output [35:0] M     /*Multiplier output M*/,
output CARRYOUT,CARRYOUTF /*Carry out from the post adder (Both are the same)*/);

//Internal Signals
wire [17:0] A_MUX,A1_MUX,B_MUX,D_MUX;
reg [17:0] B_IN,BCOUT_IN;
wire [47:0] C_MUX;
reg [47:0] X,Z,OUT;
reg [35:0] M_IN;
wire [35:0] M_BUFFER;
wire [7:0] OPMODE_MUX;
reg CIN_IN,COUT_IN;
wire CIN;

//INSTANTIATION OF REGISTERS
/*module Reg_MUX #(parameter WIDTH=18, parameter REG=1, parameter RSTTYPE = "SYNC")
(input clk,rst,enable, input [WIDTH-1:0] D, output reg [WIDTH-1:0] out);*/
Reg_MUX #(8,OPMODEREG,RSTTYPE)   OPMODE_REG(clk,RSTOPMODE,CEOPMODE,OPMODE,OPMODE_MUX);
Reg_MUX #(18,DREG,RSTTYPE)       D_REG(clk,RSTD,CED,D,D_MUX);
Reg_MUX #(18,B0REG,RSTTYPE)      B0_REG(clk,RSTB,CEB,B_IN,B_MUX);
Reg_MUX #(18,B1REG,RSTTYPE)      B1_REG(clk,RSTB,CEB,BCOUT_IN,BCOUT);
Reg_MUX #(18,A0REG,RSTTYPE)      A0_REG(clk,RSTA,CEA,A,A_MUX);
Reg_MUX #(18,A1REG,RSTTYPE)      A1_REG(clk,RSTA,CEA,A_MUX,A1_MUX);
Reg_MUX #(48,CREG,RSTTYPE)       C_REG(clk,RSTC,CEC,C,C_MUX);
Reg_MUX #(36,MREG,RSTTYPE)       M_REG(clk,RSTM,CEM,M_IN,M_BUFFER);
Reg_MUX #(1,CARRYINREG,RSTTYPE)  CYI(clk,RSTCARRYIN,CECARRYIN,CIN_IN,CIN);
Reg_MUX #(1,CARRYOUTREG,RSTTYPE) CYO(clk,RSTCARRYIN,CECARRYIN,COUT_IN,CARRYOUT);
Reg_MUX #(48,PREG,RSTTYPE)       P_REG(clk,RSTP,CEP,OUT,P);

genvar i;
generate
    for (i = 0; i < 36; i = i + 1)
        buf (M[i], M_BUFFER[i]);       //The buffer
endgenerate

assign   CARRYOUTF = CARRYOUT;
assign   PCOUT = P;
always @(*) begin
    B_IN = (B_INPUT == "DIRECT") ? B : (B_INPUT == "CASCADE") ? BCIN : 0;
    if (OPMODE_MUX[4])          BCOUT_IN = (OPMODE_MUX[6]) ? (D_MUX-B_MUX) : (D_MUX+B_MUX);
    else                    BCOUT_IN = B_MUX;
    M_IN = A1_MUX * BCOUT;
    CIN_IN = (CARRYINSEL == "OPMODE5") ? OPMODE_MUX[5] : (CARRYINSEL == "CARRYIN") ? CARRYIN : 0;
    case (OPMODE_MUX[1:0])
        0:         X = 0;
        1:         X = {12'b0,M_BUFFER};
        2:         X = P;
        3:         X = {D_MUX[11:0],A1_MUX,BCOUT};
        default : X = 0;
    endcase
    case (OPMODE_MUX[3:2])
        0:         Z = 0;
        1:         Z = PCIN;
        2:         Z = P;
        3:         Z = C_MUX;
        default : Z = 0;
    endcase
    {COUT_IN,OUT} = OPMODE_MUX[7] ? (Z - (X + CIN)) : (Z + X + CIN);
end
endmodule
```

Figure 8: DSP48A1 Code



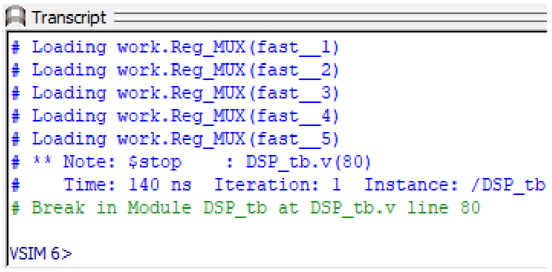
```verilog
module DSP_tb ();
parameter A0REG=0,A1REG=1,B0REG=0,B1REG=1,CREG=1,DREG=1,MREG=1,PREG=1,CARRYINREG=1,
CARRYOUTREG=1,OPMODEREG=1;                  //(0 -> Direct) or (1 -> Register)
parameter CARRYINSEL= "OPMODE5";            //CARRYIN or OPMODE5 (for selecting carry input)
parameter B_INPUT= "DIRECT";                //DIRECT or CASCADE (for selecting B input)
parameter RSTTYPE= "SYNC";                  //SYNC or ASYNC (for selecting register reset type)
reg [17:0] A,B,BCIN, D;
reg [47:0] C,PCIN;
reg [7:0] OPMODE;
reg clk,CARRYIN;
reg RSTA,RSTB,RSTC,RSTD,RSTM,RSTP,RSTCARRYIN,RSTOPMODE,
CEA,CEB,CEC,CED,CEM,CEP,CECARRYIN,CEOPMODE;
wire [17:0] BCOUT;
wire [47:0] P, PCOUT;
wire [35:0] M;
wire CARRYOUT,CARRYOUTF;

/*module DSP #(parameter A0REG=0,A1REG=1,B0REG=0,B1REG=1,CREG=1,DREG=1,MREG=1,PREG=1,CARRYINREG=1,
CARRYOUTREG=1,OPMODEREG=1,
CARRYINSEL= "OPMODE5",
B_INPUT= "DIRECT",
RSTTYPE= "SYNC")
(input [17:0] A,B,BCIN, D, input [47:0] C,PCIN, input [7:0] OPMODE, input clk,CARRYIN,
RSTA,RSTB,RSTC,RSTD,RSTM,RSTP,RSTCARRYIN,RSTOPMODE, CEA,CEB,CEC,CED,CEM,CEP,CECARRYIN,CEOPMODE,
output [17:0] BCOUT, output [47:0] P, PCOUT, output [35:0] M, output CARRYOUT,CARRYOUTF);*/
DSP #(A0REG,A1REG,B0REG,B1REG,CREG,DREG,MREG,PREG,CARRYINREG,
CARRYOUTREG,OPMODEREG,CARRYINSEL,B_INPUT,RSTTYPE)
DUT(A,B,BCIN,D,C,PCIN,OPMODE,clk,CARRYIN,RSTA,RSTB,RSTC,RSTD,RSTM,RSTP,RSTCARRYIN,RSTOPMODE,
CEA,CEB,CEC,CED,CEM,CEP,CECARRYIN,CEOPMODE,BCOUT,P,PCOUT,M,CARRYOUT,CARRYOUTF);

initial begin
    clk=0;
    forever #5 clk=~clk;
end

initial begin
    //Reset Functionality
    RSTA=1; RSTB=1; RSTC=1; RSTD=1; RSTM=1; RSTP=1; RSTCARRYIN=1; RSTOPMODE=1;
    CEA=$random; CEB=$random; CEC=$random; CED=$random; CEM=$random; CEP=$random; CECARRYIN=$random; CEOPMODE=$random;
    A=$random; B=$random; BCIN=$random; D=$random; C=$random; PCIN=$random; OPMODE=$random; CARRYIN=$random;
    @(negedge clk);
    if (P || PCOUT || M || BCOUT || CARRYOUT || CARRYOUTF) begin
        $display("Error in Reset!!");
        $stop;
    end
    RSTA=0; RSTB=0; RSTC=0; RSTD=0; RSTM=0; RSTP=0; RSTCARRYIN=0; RSTOPMODE=0;
    CEA=1; CEB=1; CEC=1; CED=1; CEM=1; CEP=1; CECARRYIN=1; CEOPMODE=1;

    //Path 1
    A=20; B=10; D=25; C=350; OPMODE=8'b11011101; PCIN=$random; BCIN=$random; CARRYIN=$random;
    repeat (4) @(negedge clk);
    if (BCOUT != 15 || M != 300 || P != 50 || PCOUT != 50 || CARRYOUT || CARRYOUTF) begin
        $display("Error in Path 1!!");
        $stop;
    end

    //Path 2
    A=20; B=10; D=25; C=350; OPMODE=8'b00010000; PCIN=$random; BCIN=$random; CARRYIN=$random;
    repeat (3) @(negedge clk);
    if (BCOUT != 35 || M != 700 || P || PCOUT || CARRYOUT || CARRYOUTF) begin
        $display("Error in Path 2!!");
        $stop;
    end

    //Path 3
    A=20; B=10; D=25; C=350; OPMODE=8'b00001010; PCIN=$random; BCIN=$random; CARRYIN=$random;
    repeat (3) @(negedge clk);
    if (BCOUT != 10 || M != 200 || P || PCOUT || CARRYOUT || CARRYOUTF) begin
        $display("Error in Path 3!!");
        $stop;
    end

    //Path 4
    A=5; B=6; D=25; C=350; OPMODE=8'b10100111; PCIN=3000; BCIN=$random; CARRYIN=$random;
    repeat (3) @(negedge clk);
    if (BCOUT != 6 || M != 30 || P != 48'hfe6fffec0bb1 || PCOUT != 48'hfe6fffec0bb1 || !CARRYOUT || !CARRYOUTF) begin
        $display("Error in Path 4!!");
        $stop;
    end
    $stop;
end
endmodule
```

Figure 9: DSP48A1 Testbench



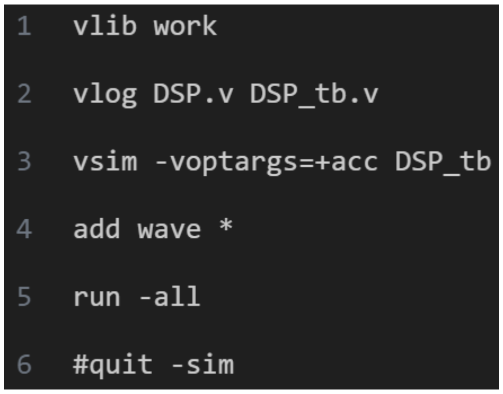
```
RSTA          1'h0
RSTB          1'h0
RSTC          1'h0
RSTD          1'h0
RSTM          1'h0
RSTP          1'h0
RSTCARRYIN    1'h0
RSTOPMODE     1'h0
CEA           1'h1
CEB           1'h1
CEC           1'h1
CED           1'h1
CEM           1'h1
CEP           1'h1
CECARRYIN     1'h1
CEOPMODE      1'h1
CARRYIN       1'h0
clk           1'h1
OPMODE        8'b10100111
A             18'd5
B             18'd6
D             18'd25
C             48'd350
PCIN          48'd3000
BCIN          18'd151237
BCOUT         18'h00006
M             36'h00000001e
P             48'hfe6fffec0bb1
PCOUT         48'hfe6fffec0bb1
CARRYOUT      1'h1
CARRYOUTF     1'h1
Now           140 ns
Cursor 1      135 ns
```

Figure 10: DSP48A1 Wave



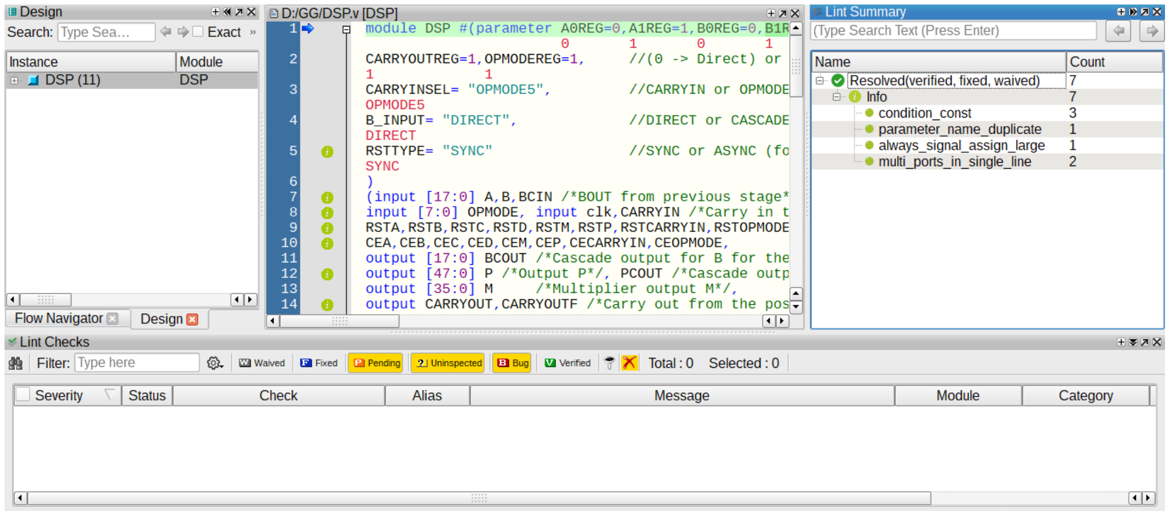
```
Transcript
# Loading work.Reg_MUX(fast__1)
# Loading work.Reg_MUX(fast__2)
# Loading work.Reg_MUX(fast__3)
# Loading work.Reg_MUX(fast__4)
# Loading work.Reg_MUX(fast__5)
# ** Note: $stop    : DSP_tb.v(80)
#    Time: 140 ns  Iteration: 1  Instance: /DSP_tb
# Break in Module DSP_tb at DSP_tb.v line 80

VSIM 6>
```

Figure 11: DSP48A1 Transcript



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

Figure 12: DSP48A1 DO File



Design — Search: Type Sea… Exact

Instance	Module
DSP (11)	DSP

D:/GG/DSP.v [DSP]

```
module DSP #(parameter A0REG=0,A1REG=1,B0REG=0,B1R
                                0      1      0      1
CARRYOUTREG=1,OPMODEREG=1,      //(0 -> Direct) or
1            1
CARRYINSEL= "OPMODE5",          //CARRYIN or OPMODE
OPMODE5
B_INPUT= "DIRECT",              //DIRECT or CASCADE
DIRECT
RSTTYPE= "SYNC"                 //SYNC or ASYNC (fo
SYNC
)
(input [17:0] A,B,BCIN /*BOUT from previous stage*
input [7:0] OPMODE, input clk,CARRYIN /*Carry in t
RSTA,RSTB,RSTC,RSTD,RSTM,RSTP,RSTCARRYIN,RSTOPMODE
CEA,CEB,CEC,CED,CEM,CEP,CECARRYIN,CEOPMODE,
output [17:0] BCOUT /*Cascade output for B for the
output [47:0] P /*Output P*/, PCOUT /*Cascade outp
output [35:0] M     /*Multiplier output M*/,
output CARRYOUT,CARRYOUTF /*Carry out from the pos
```

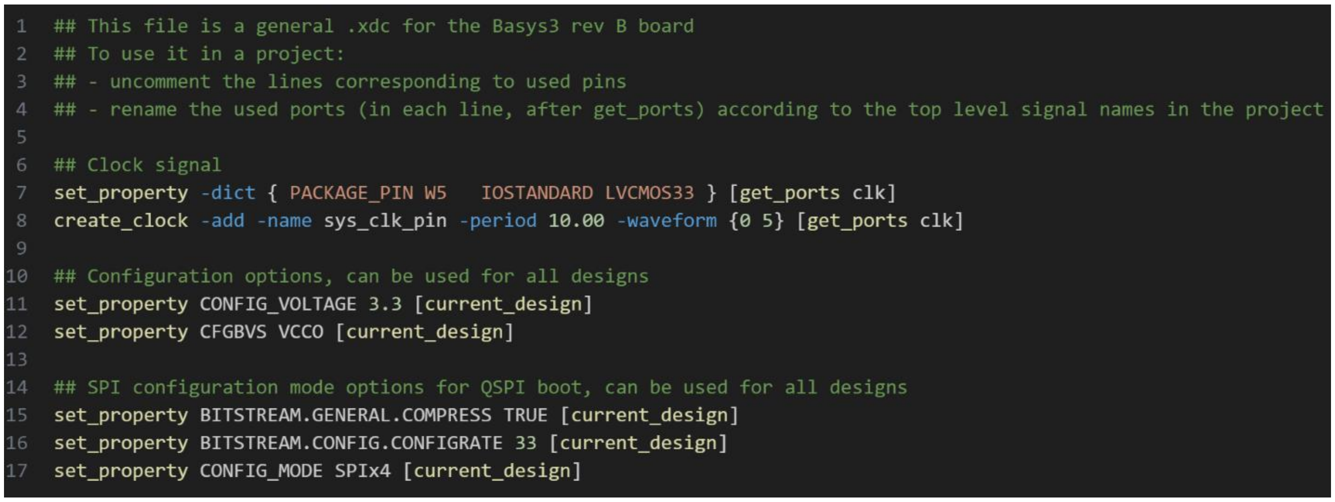
Lint Summary

Name	Count
Resolved(verified, fixed, waived)	7
Info	7
condition_const	3
parameter_name_duplicate	1
always_signal_assign_large	1
multi_ports_in_single_line	2

Lint Checks — Filter: Type here — Waived, Fixed, Pending, Uninspected, Bug, Verified — Total : 0 Selected : 0

Severity	Status	Check	Alias	Message	Module	Category

Figure 13: DSP48A1 Linting



```
## 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]

## 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]
```

Figure 14: DSP48A1 Constraint File

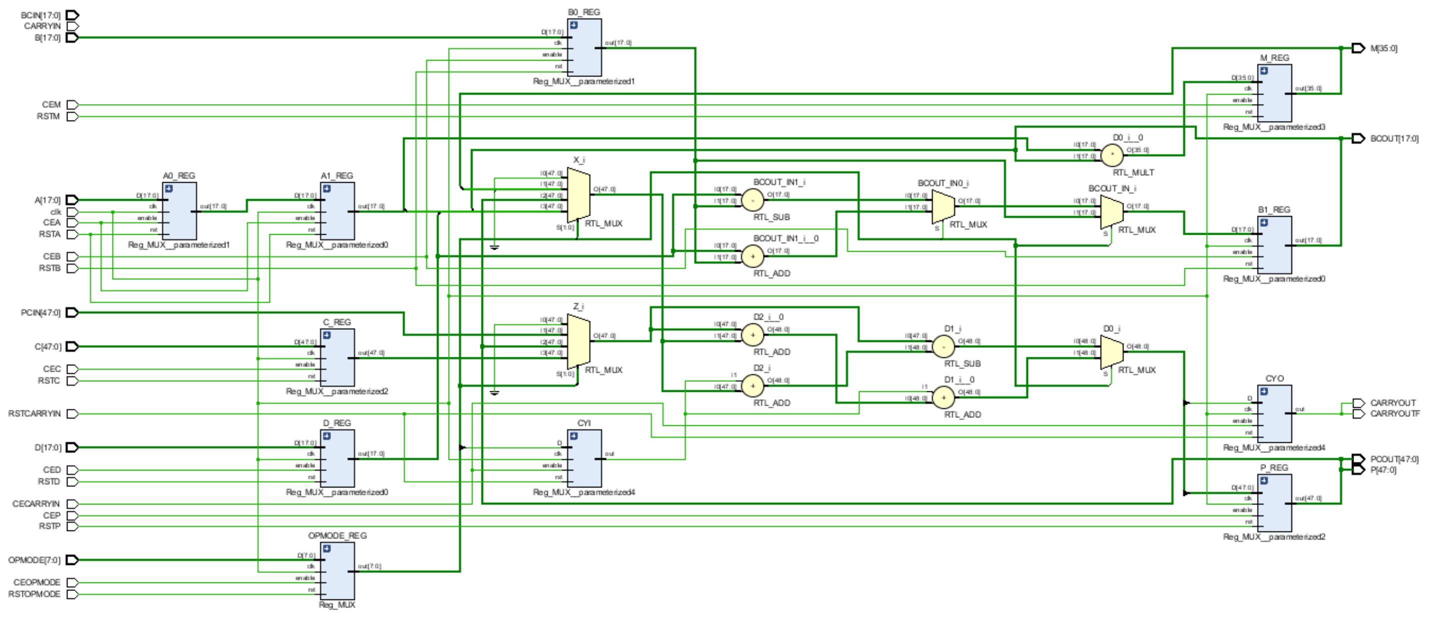


Figure 15: DSP48A1 RTL

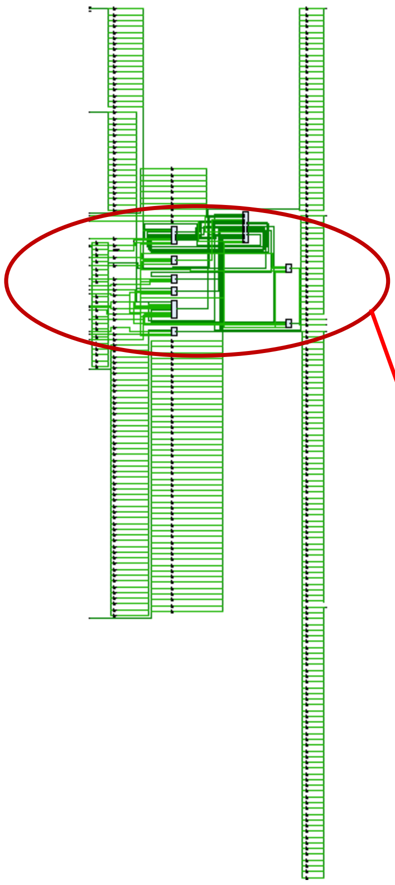


Figure 16: DSP48A1 Synthesis Schematic

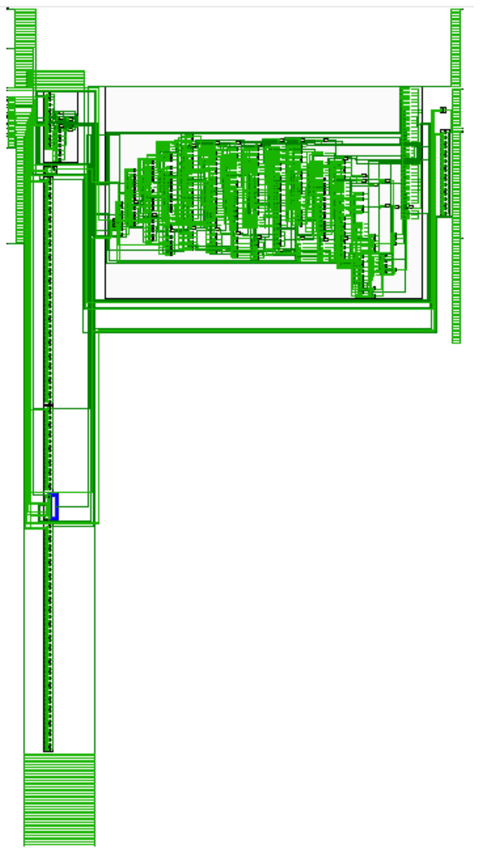


Figure 17: DSP48A1 Detailed Synthesis Schematic

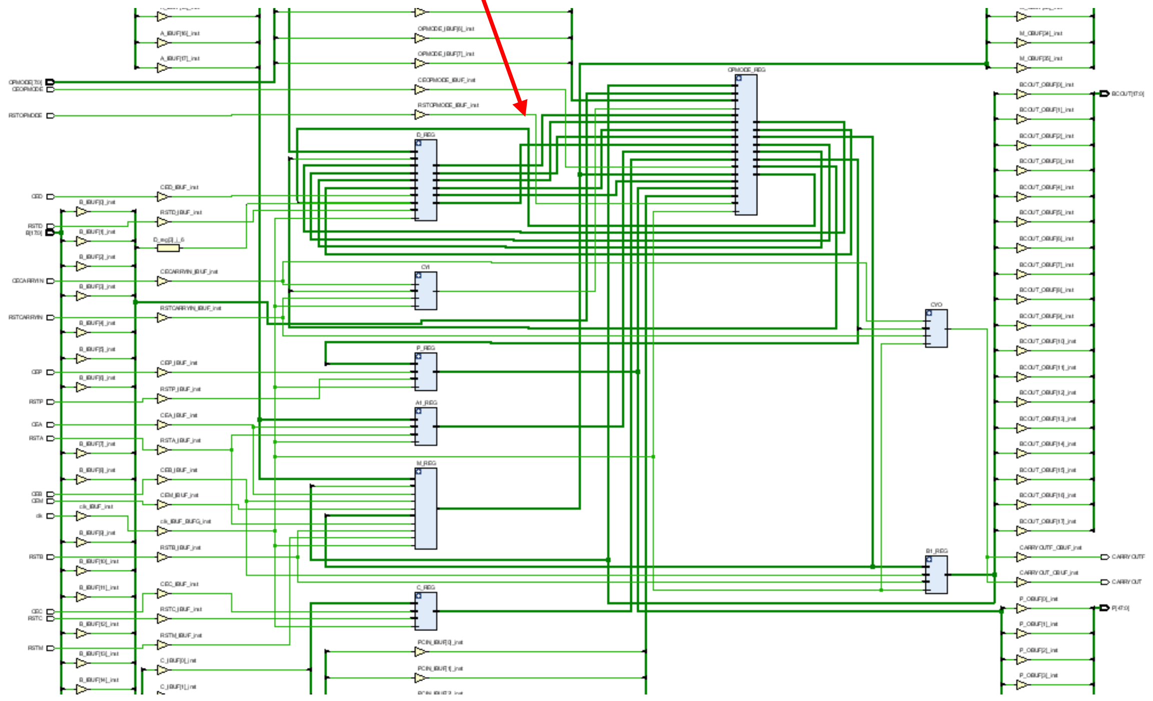


Figure 18: DSP48A1 Zoomed Synthesis Schematic (1)

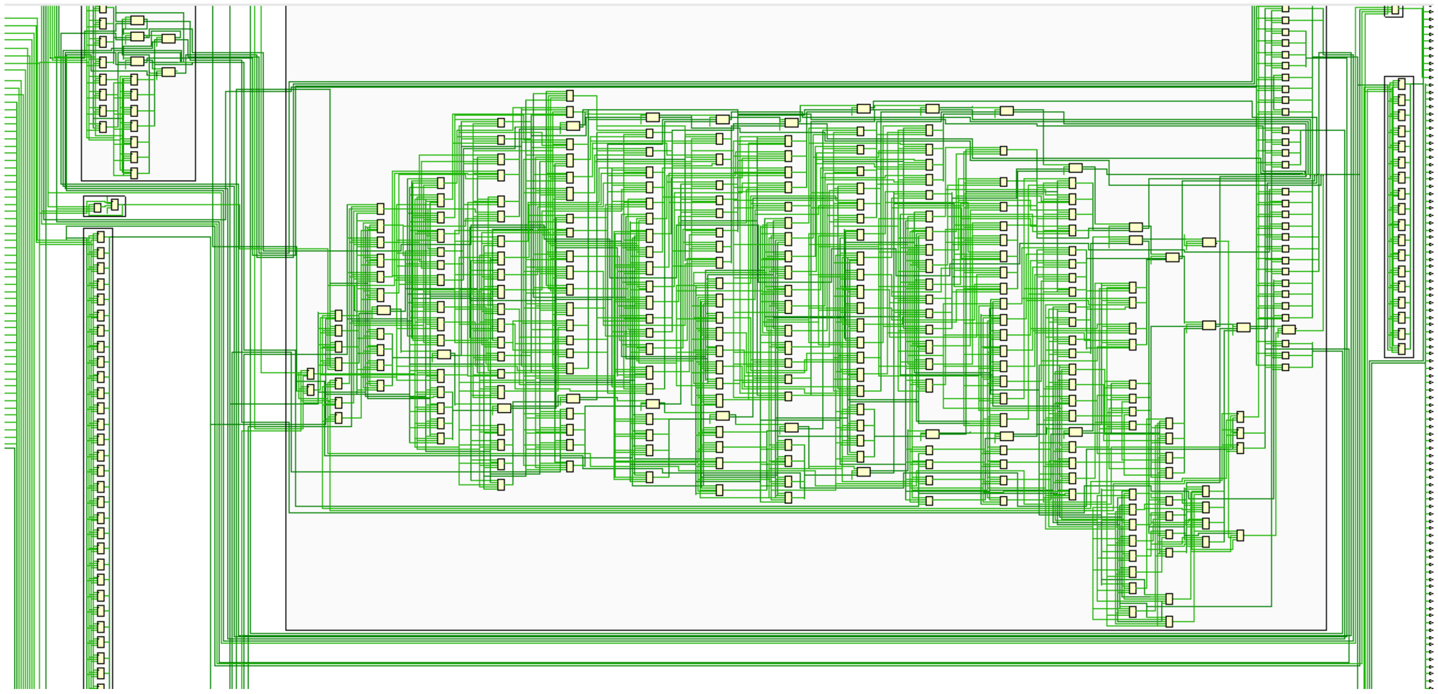


Figure 19: DSP48A1 Zoomed Detailed Synthesis Schematic

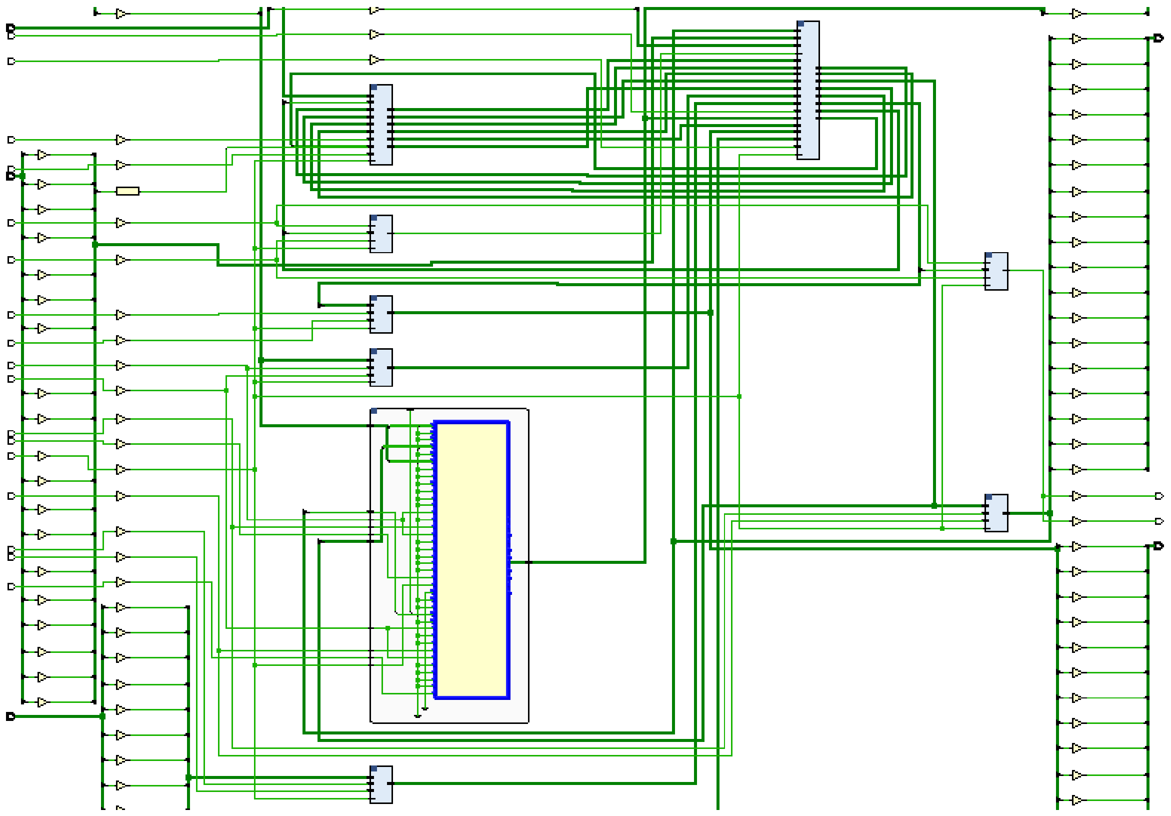


Figure 20: DSP48A1 Zoomed Synthesis Schematic (2)

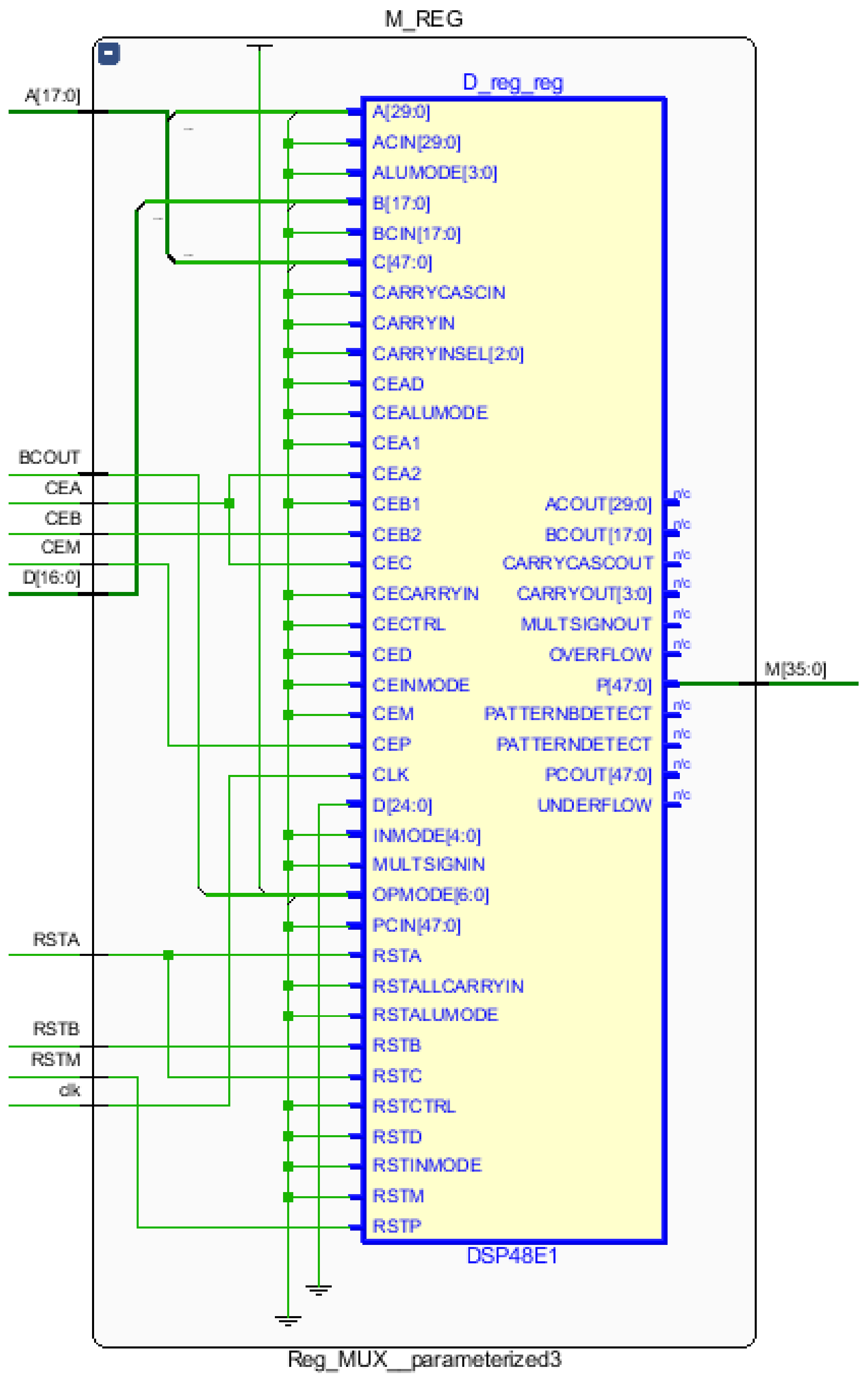


Figure 21: M_REG with DSP48E1

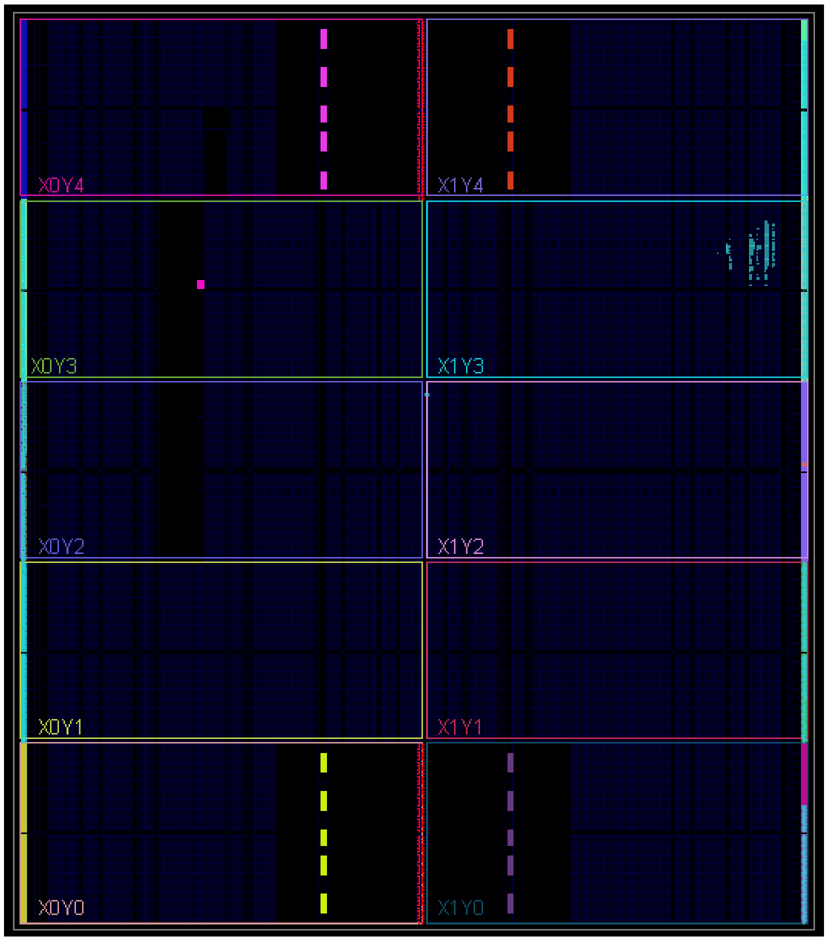


Figure 22: Device

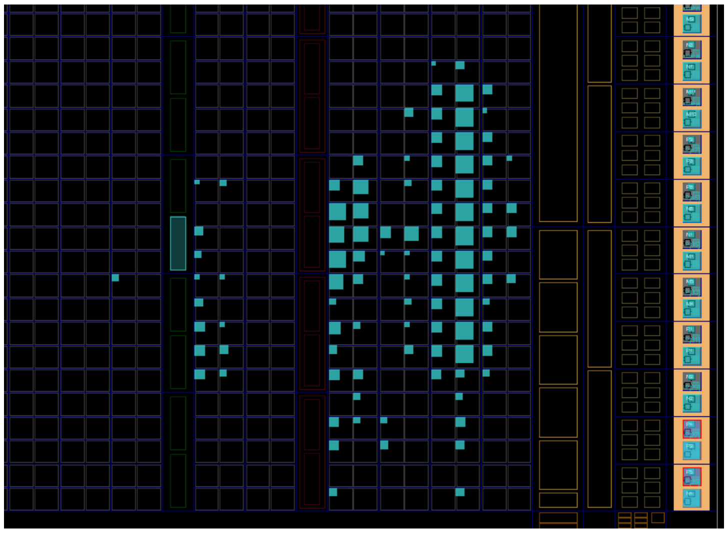
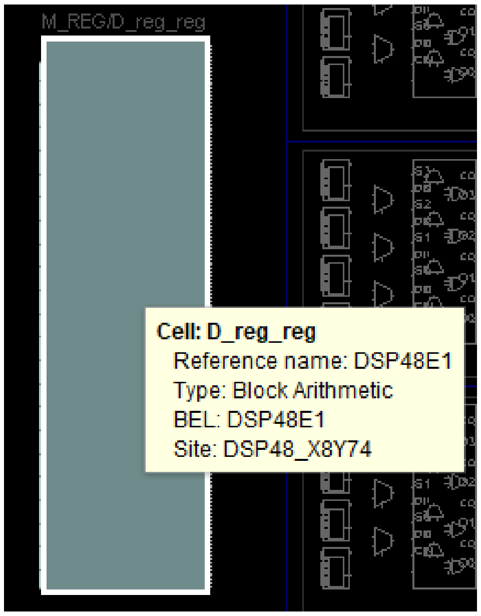
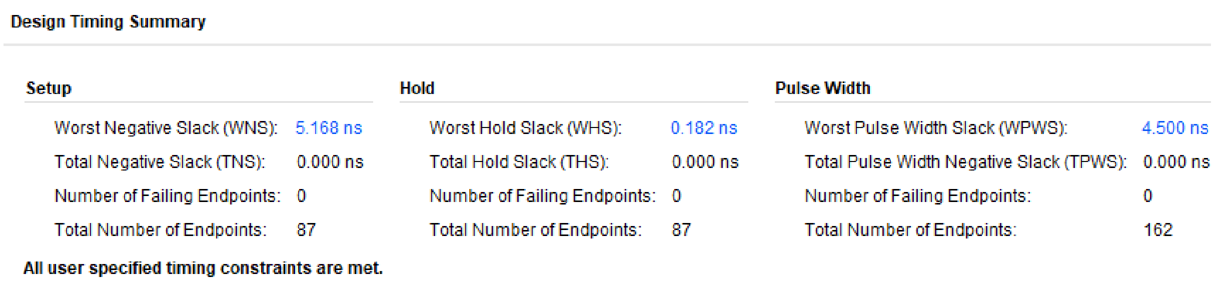


Figure 23: Zoomed Device



Cell: D_reg_reg
Reference name: DSP48E1
Type: Block Arithmetic
BEL: DSP48E1
Site: DSP48_X8Y74

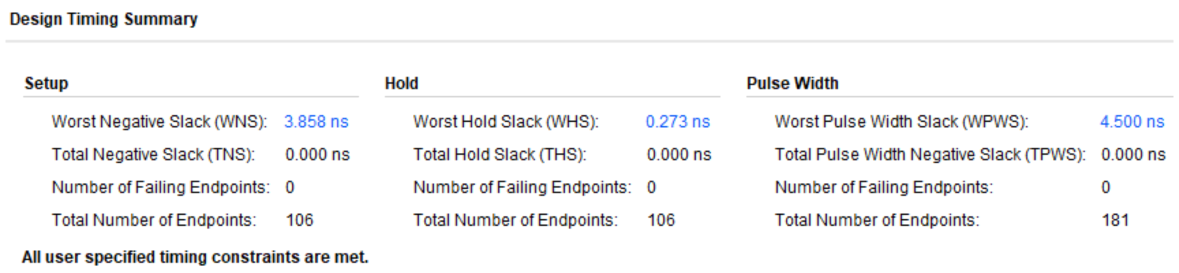
Figure 24: DSP48E1


Design Timing Summary

Setup		**Hold**		**Pulse Width**	
Worst Negative Slack (WNS):	5.168 ns	Worst Hold Slack (WHS):	0.182 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:	87	Total Number of Endpoints:	87	Total Number of Endpoints:	162

All user specified timing constraints are met.

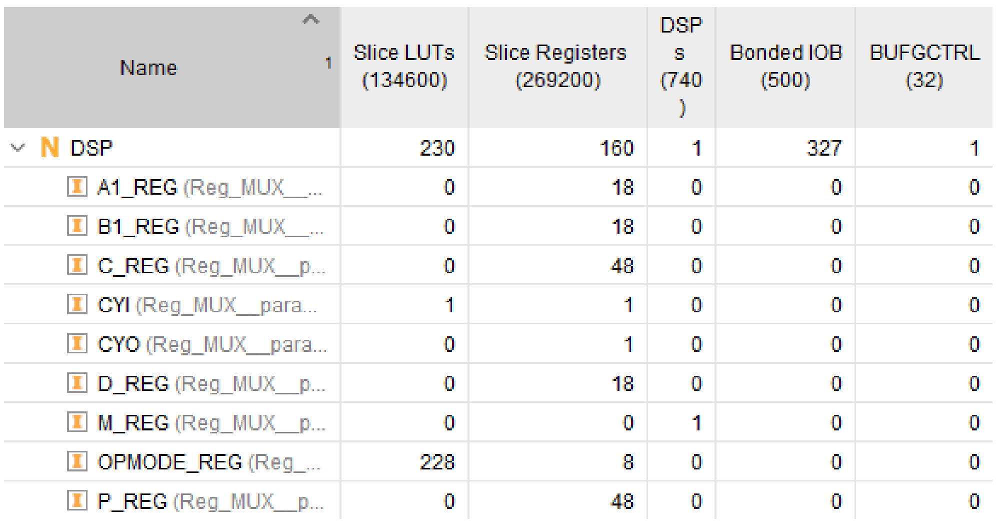
Figure 25: DSP48A1 Synthesis Timing Report


Design Timing Summary

Setup		**Hold**		**Pulse Width**	
Worst Negative Slack (WNS):	3.858 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:	106	Total Number of Endpoints:	106	Total Number of Endpoints:	181

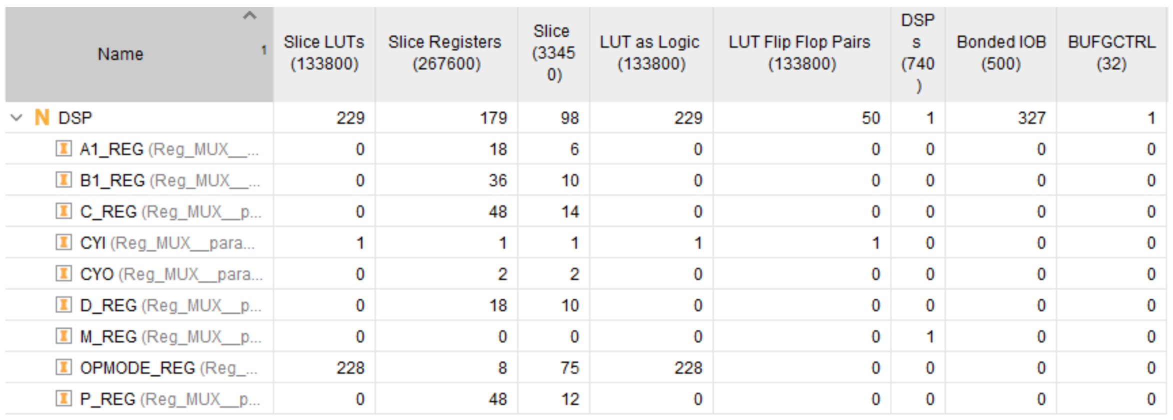
All user specified timing constraints are met.

Figure 26: DSP48A1 Implementation Timing Report



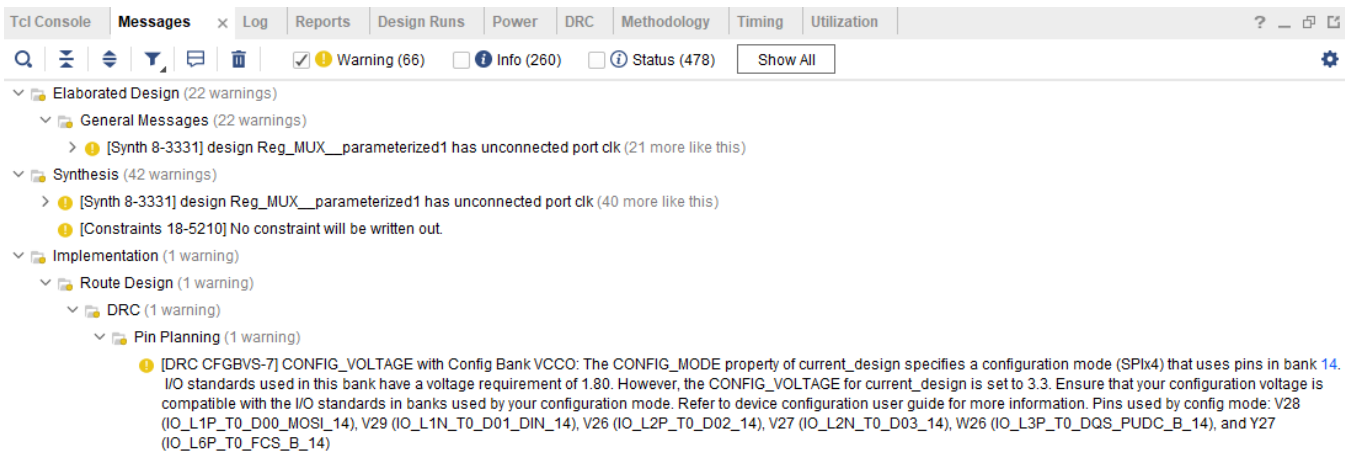
Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP	230	160	1	327	1
A1_REG (Reg_MUX__...	0	18	0	0	0
B1_REG (Reg_MUX__...	0	18	0	0	0
C_REG (Reg_MUX__p...	0	48	0	0	0
CYI (Reg_MUX__para...	1	1	0	0	0
CYO (Reg_MUX__para...	0	1	0	0	0
D_REG (Reg_MUX__p...	0	18	0	0	0
M_REG (Reg_MUX__p...	0	0	1	0	0
OPMODE_REG (Reg_...	228	8	0	0	0
P_REG (Reg_MUX__p...	0	48	0	0	0

Figure 27: DSP48A1 Synthesis Utilization Report



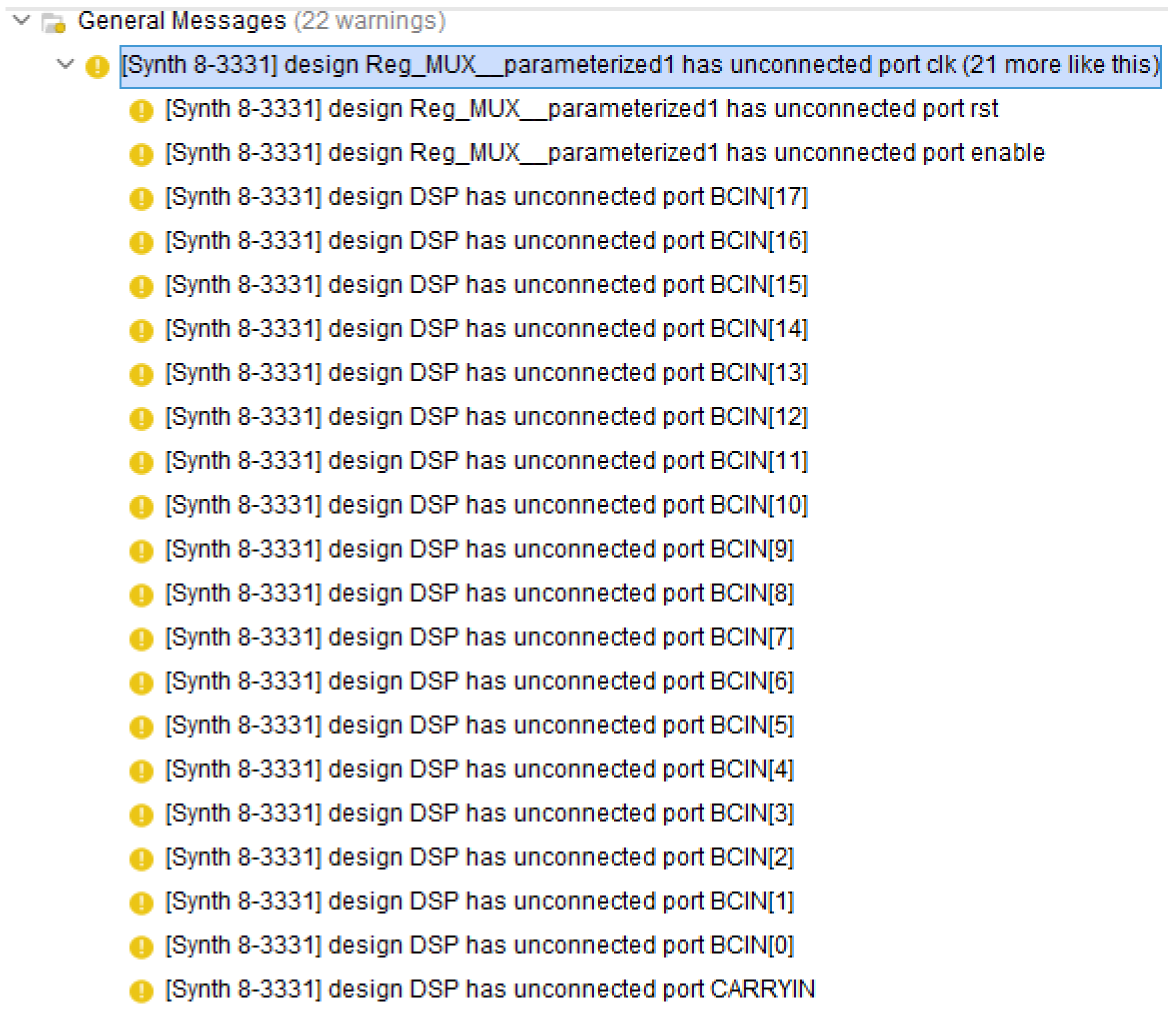
Name	Slice LUTs (133800)	Slice Registers (267600)	Slice (33450)	LUT as Logic (133800)	LUT Flip Flop Pairs (133800)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP	229	179	98	229	50	1	327	1
A1_REG (Reg_MUX__...	0	18	6	0	0	0	0	0
B1_REG (Reg_MUX__...	0	36	10	0	0	0	0	0
C_REG (Reg_MUX__p...	0	48	14	0	0	0	0	0
CYI (Reg_MUX__para...	1	1	1	1	1	0	0	0
CYO (Reg_MUX__para...	0	2	2	0	0	0	0	0
D_REG (Reg_MUX__p...	0	18	10	0	0	0	0	0
M_REG (Reg_MUX__p...	0	0	0	0	0	1	0	0
OPMODE_REG (Reg_...	228	8	75	228	0	0	0	0
P_REG (Reg_MUX__p...	0	48	12	0	0	0	0	0

Figure 28: DSP48A1 Implementation Utilization Report



```
Elaborated Design (22 warnings)
  General Messages (22 warnings)
    [Synth 8-3331] design Reg_MUX__parameterized1 has unconnected port clk (21 more like this)
Synthesis (42 warnings)
  [Synth 8-3331] design Reg_MUX__parameterized1 has unconnected port clk (40 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)
```

Figure 29: Messages



```
General Messages (22 warnings)
  [Synth 8-3331] design Reg_MUX__parameterized1 has unconnected port clk (21 more like this)
    [Synth 8-3331] design Reg_MUX__parameterized1 has unconnected port rst
    [Synth 8-3331] design Reg_MUX__parameterized1 has unconnected port enable
    [Synth 8-3331] design DSP has unconnected port BCIN[17]
    [Synth 8-3331] design DSP has unconnected port BCIN[16]
    [Synth 8-3331] design DSP has unconnected port BCIN[15]
    [Synth 8-3331] design DSP has unconnected port BCIN[14]
    [Synth 8-3331] design DSP has unconnected port BCIN[13]
    [Synth 8-3331] design DSP has unconnected port BCIN[12]
    [Synth 8-3331] design DSP has unconnected port BCIN[11]
    [Synth 8-3331] design DSP has unconnected port BCIN[10]
    [Synth 8-3331] design DSP has unconnected port BCIN[9]
    [Synth 8-3331] design DSP has unconnected port BCIN[8]
    [Synth 8-3331] design DSP has unconnected port BCIN[7]
    [Synth 8-3331] design DSP has unconnected port BCIN[6]
    [Synth 8-3331] design DSP has unconnected port BCIN[5]
    [Synth 8-3331] design DSP has unconnected port BCIN[4]
    [Synth 8-3331] design DSP has unconnected port BCIN[3]
    [Synth 8-3331] design DSP has unconnected port BCIN[2]
    [Synth 8-3331] design DSP has unconnected port BCIN[1]
    [Synth 8-3331] design DSP has unconnected port BCIN[0]
    [Synth 8-3331] design DSP has unconnected port CARRYIN
```

Figure 30: Warnings

- There are no errors. There are "unconnected port" warnings because of the unused inputs (BCIN – CARRYIN).