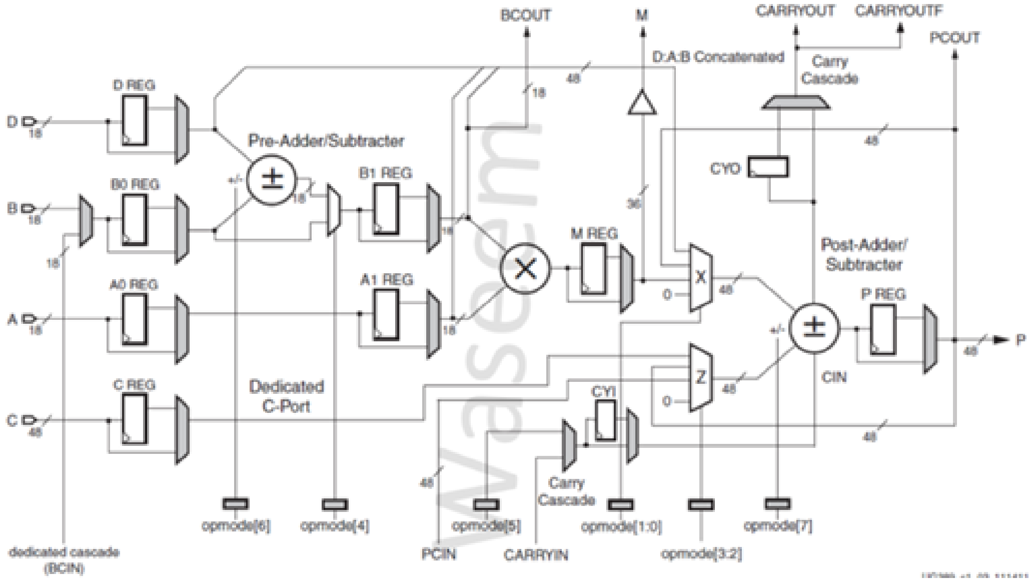
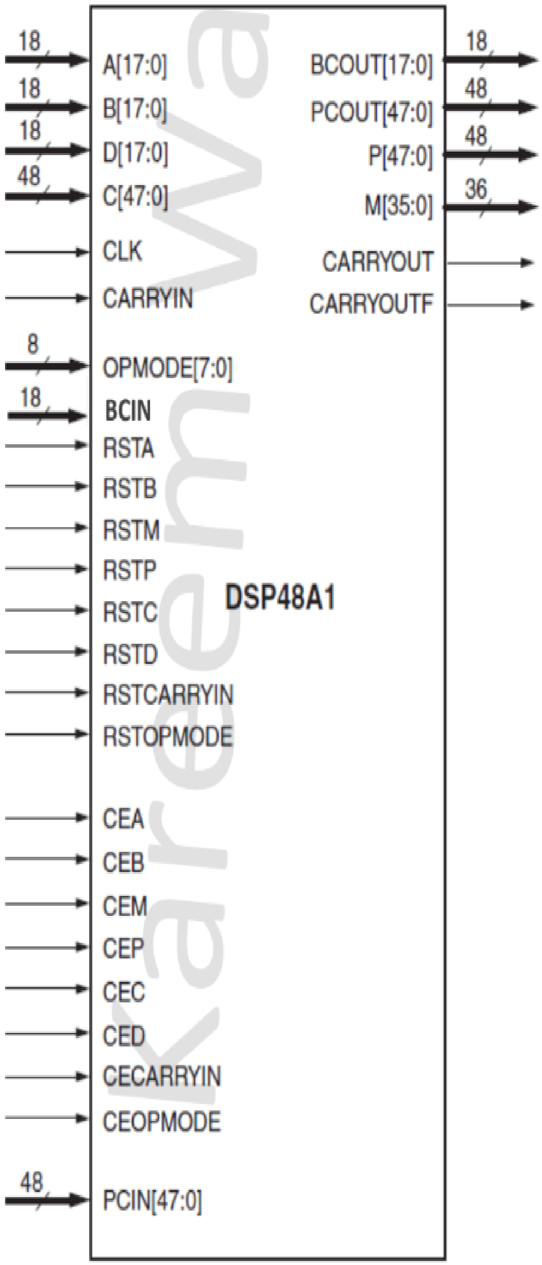
DSP Project

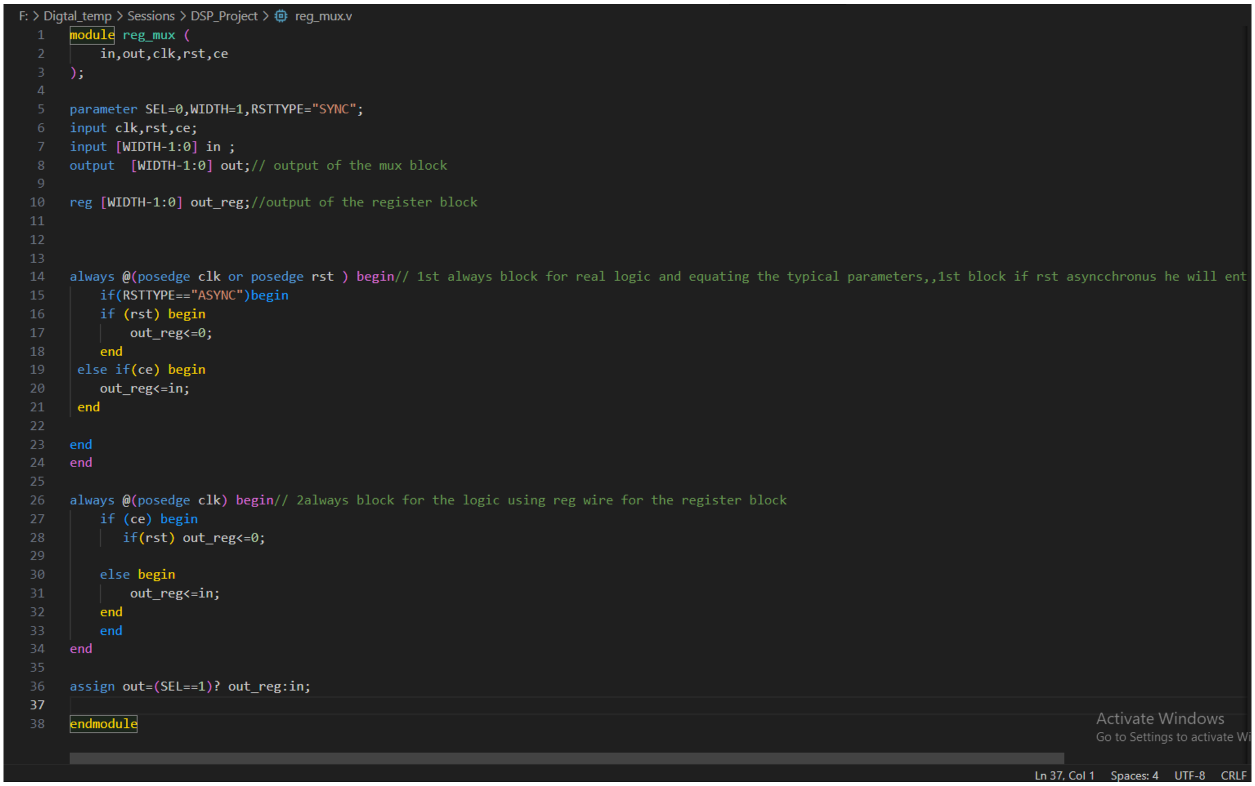
SPARTAN6-DSP48A1

DSP48A1 A Powerful Building Block for FPGA-based DSP
DSP48A1 is a dedicated digital signal processing (DSP) slice found in certain generations of Xilinx FPGAs, specifically the Spartan-6 family. It's designed to accelerate computationally intensive DSP algorithms by providing specialized hardware resources within the FPGA fabric.

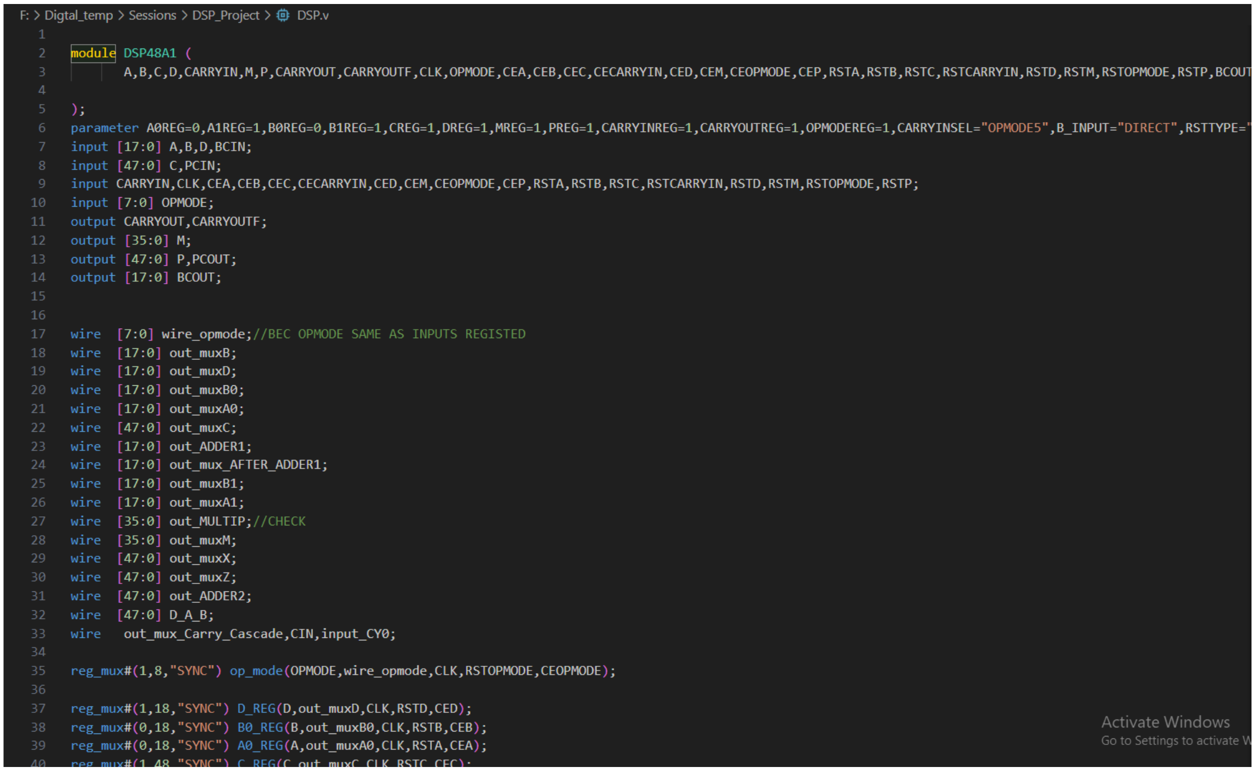
Benefits of using DSP48A1
Improved performance: Dedicated hardware accelerates DSP operations compared to using general-purpose FPGA logic.
Lower power consumption: Optimized hardware design for DSP functions reduces power consumption.
Reduced FPGA resource utilization: Effectively utilizes FPGA resources for DSP-intensive applications.



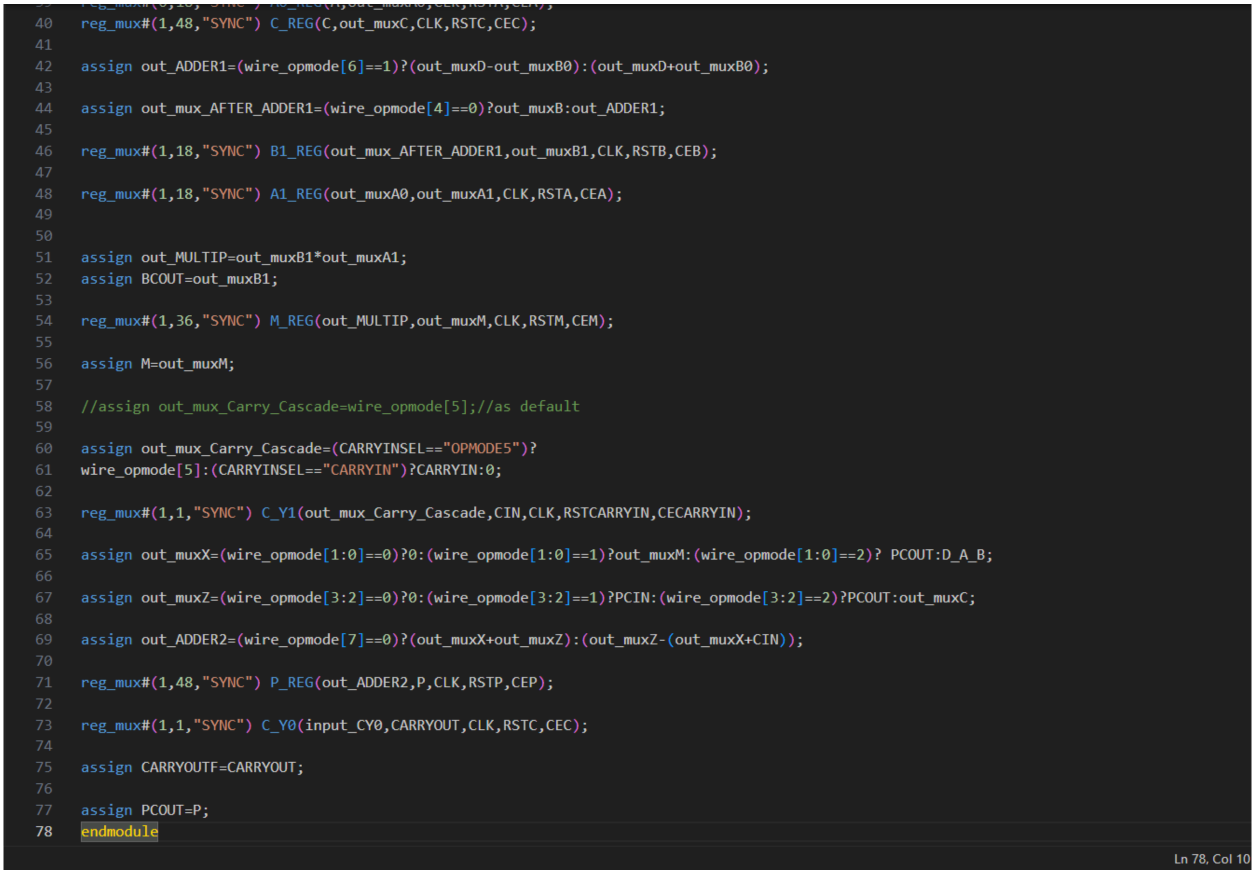

18
A[17:0]
18
B[17:0]
18
D[17:0]
48
C[47:0]
CLK
CARRYIN
8
OPMODE[7:0]
18
BCIN
RSTA
RSTB
RSTM
RSTP
RSTC
RSTD
RSTCARRYIN
RSTOPMODE
CEA
CEB
CEM
CEP
CEC
CED
CECARRYIN
CEOPMODE
48
PCIN[47:0]
DSP48A1
18
BCOUT[17:0]
48
PCOUT[47:0]
48
P[47:0]
36
M[35:0]
CARRYOUT
CARRYOUTF



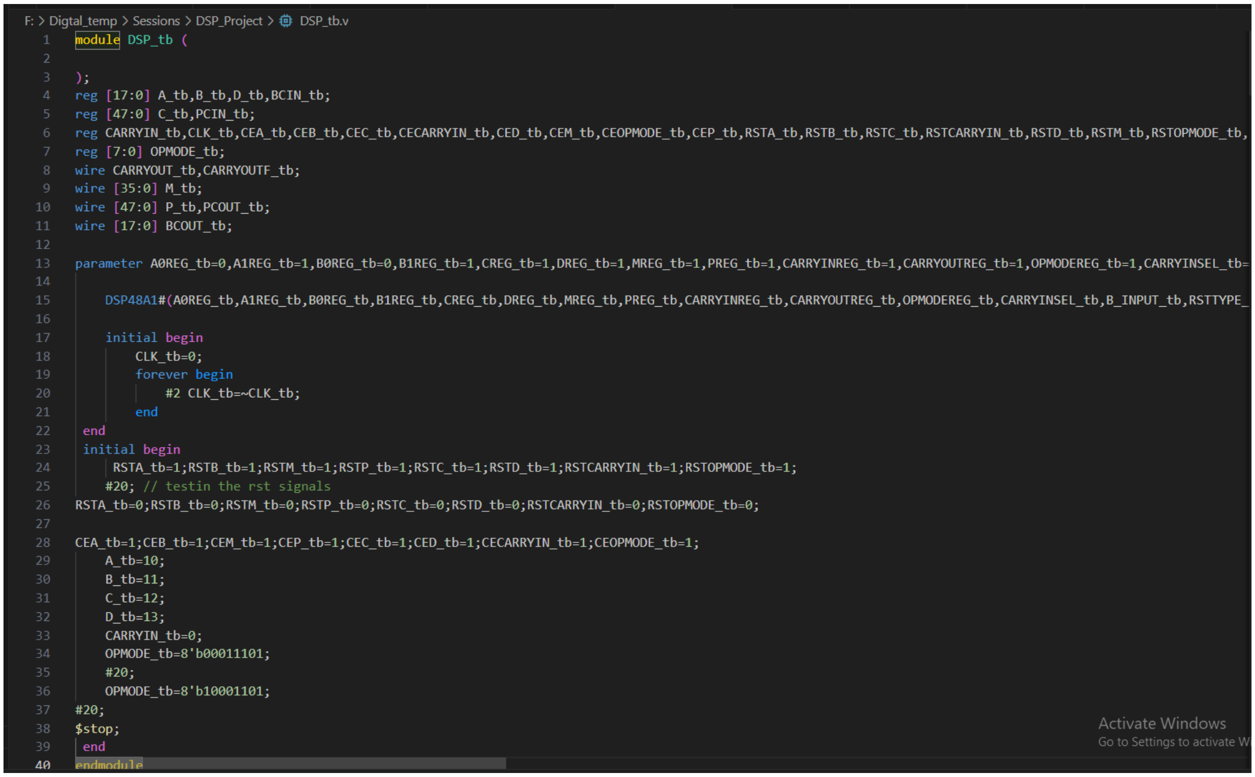
```verilog
module reg_mux (
    in,out,clk,rst,ce
);

parameter SEL=0,WIDTH=1,RSTTYPE="SYNC";
input clk,rst,ce;
input [WIDTH-1:0] in ;
output  [WIDTH-1:0] out;// output of the mux block

reg [WIDTH-1:0] out_reg;//output of the register block



always @(posedge clk or posedge rst ) begin// 1st always block for real logic and equating the typical parameters,,1st block if rst asyncchronus he will ent
    if(RSTTYPE=="ASYNC")begin
    if (rst) begin
        out_reg<=0;
    end
 else if(ce) begin
    out_reg<=in;
 end

end
end

always @(posedge clk) begin// 2always block for the logic using reg wire for the register block
    if (ce) begin
       if(rst) out_reg<=0;

    else begin
        out_reg<=in;
    end
    end
end

assign out=(SEL==1)? out_reg:in;

endmodule
```



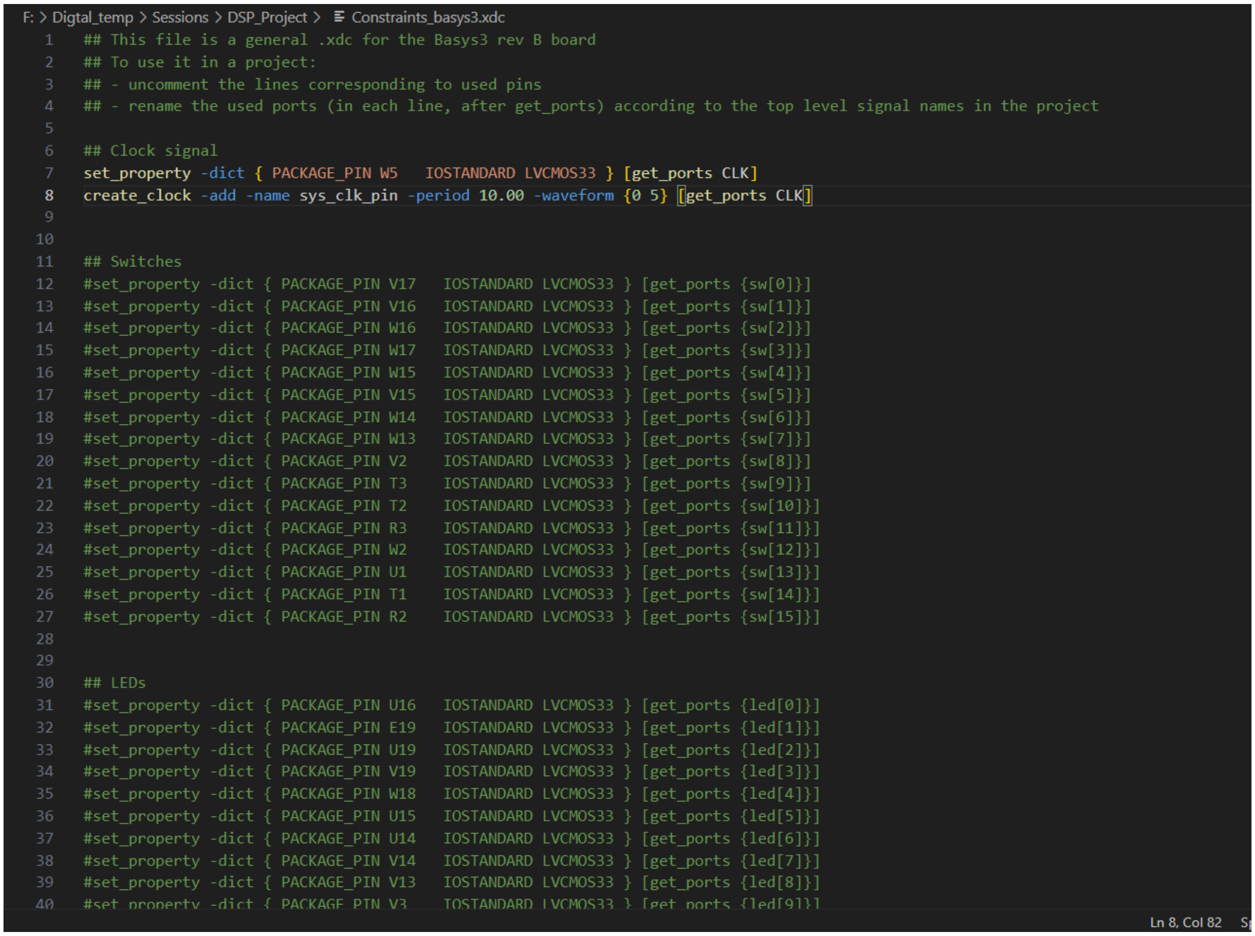
```verilog

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

);
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="
input [17:0] A,B,D,BCIN;
input [47:0] C,PCIN;
input CARRYIN,CLK,CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP;
input [7:0] OPMODE;
output CARRYOUT,CARRYOUTF;
output [35:0] M;
output [47:0] P,PCOUT;
output [17:0] BCOUT;


wire  [7:0] wire_opmode;//BEC OPMODE SAME AS INPUTS REGISTED
wire  [17:0] out_muxB;
wire  [17:0] out_muxD;
wire  [17:0] out_muxB0;
wire  [17:0] out_muxA0;
wire  [47:0] out_muxC;
wire  [17:0] out_ADDER1;
wire  [17:0] out_mux_AFTER_ADDER1;
wire  [17:0] out_muxB1;
wire  [17:0] out_muxA1;
wire  [35:0] out_MULTIP;//CHECK
wire  [35:0] out_muxM;
wire  [47:0] out_muxX;
wire  [47:0] out_muxZ;
wire  [47:0] out_ADDER2;
wire  [47:0] D_A_B;
wire   out_mux_Carry_Cascade,CIN,input_CY0;

reg_mux#(1,8,"SYNC") op_mode(OPMODE,wire_opmode,CLK,RSTOPMODE,CEOPMODE);

reg_mux#(1,18,"SYNC") D_REG(D,out_muxD,CLK,RSTD,CED);
reg_mux#(0,18,"SYNC") B0_REG(B,out_muxB0,CLK,RSTB,CEB);
reg_mux#(0,18,"SYNC") A0_REG(A,out_muxA0,CLK,RSTA,CEA);
reg_mux#(1,48,"SYNC") C_REG(C,out_muxC,CLK,RSTC,CEC);
```



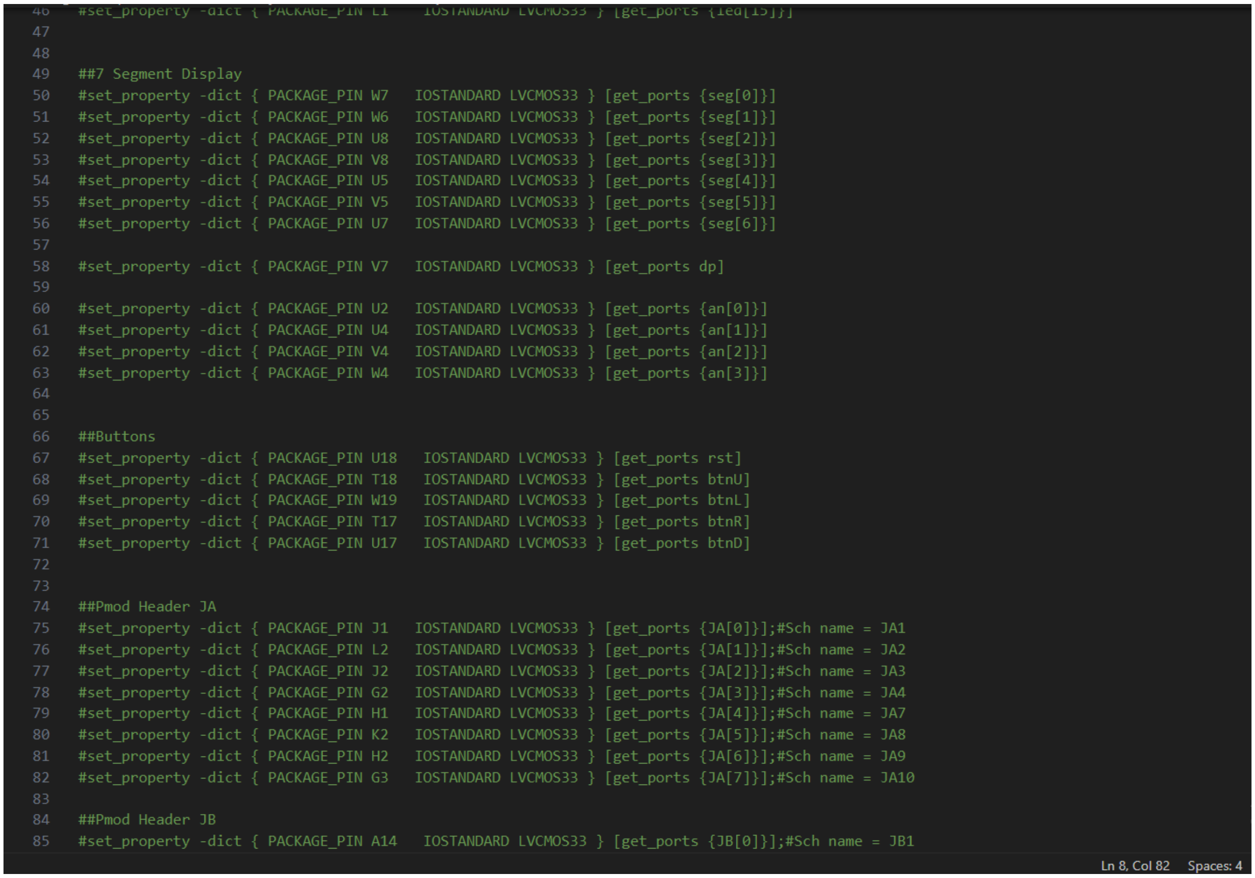
```verilog
reg_mux#(0,18,"SYNC") A0_REG(A,out_muxA0,CLK,RSTA,CEA);
reg_mux#(1,48,"SYNC") C_REG(C,out_muxC,CLK,RSTC,CEC);

assign out_ADDER1=(wire_opmode[6]==1)?(out_muxD-out_muxB0):(out_muxD+out_muxB0);

assign out_mux_AFTER_ADDER1=(wire_opmode[4]==0)?out_muxB:out_ADDER1;

reg_mux#(1,18,"SYNC") B1_REG(out_mux_AFTER_ADDER1,out_muxB1,CLK,RSTB,CEB);

reg_mux#(1,18,"SYNC") A1_REG(out_muxA0,out_muxA1,CLK,RSTA,CEA);


assign out_MULTIP=out_muxB1*out_muxA1;
assign BCOUT=out_muxB1;

reg_mux#(1,36,"SYNC") M_REG(out_MULTIP,out_muxM,CLK,RSTM,CEM);

assign M=out_muxM;

//assign out_mux_Carry_Cascade=wire_opmode[5];//as default

assign out_mux_Carry_Cascade=(CARRYINSEL=="OPMODE5")?
wire_opmode[5]:(CARRYINSEL=="CARRYIN")?CARRYIN:0;

reg_mux#(1,1,"SYNC") C_Y1(out_mux_Carry_Cascade,CIN,CLK,RSTCARRYIN,CECARRYIN);

assign out_muxX=(wire_opmode[1:0]==0)?0:(wire_opmode[1:0]==1)?out_muxM:(wire_opmode[1:0]==2)? PCOUT:D_A_B;

assign out_muxZ=(wire_opmode[3:2]==0)?0:(wire_opmode[3:2]==1)?PCIN:(wire_opmode[3:2]==2)?PCOUT:out_muxC;

assign out_ADDER2=(wire_opmode[7]==0)?(out_muxX+out_muxZ):(out_muxZ-(out_muxX+CIN));

reg_mux#(1,48,"SYNC") P_REG(out_ADDER2,P,CLK,RSTP,CEP);

reg_mux#(1,1,"SYNC") C_Y0(input_CY0,CARRYOUT,CLK,RSTC,CEC);

assign CARRYOUTF=CARRYOUT;

assign PCOUT=P;
endmodule
```



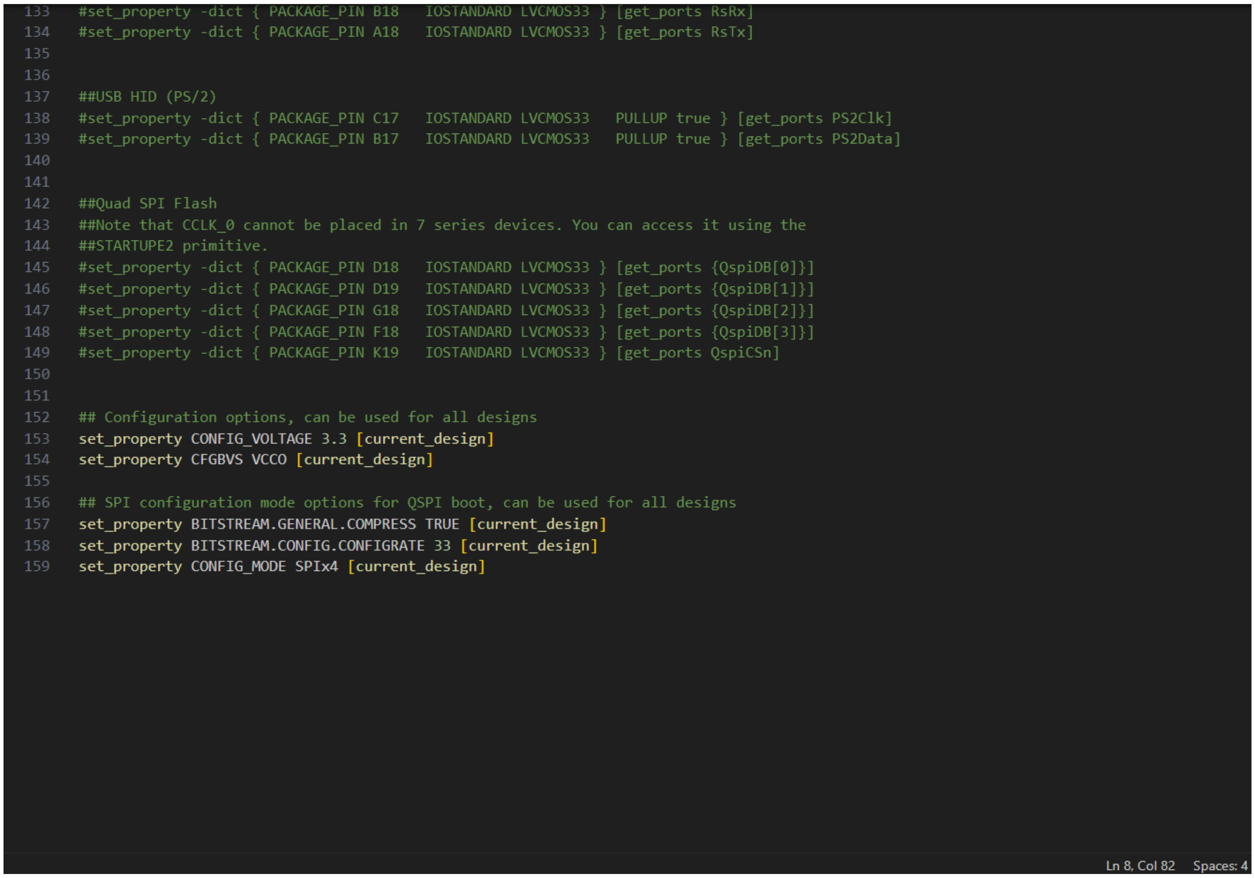
```verilog
module DSP_tb (

);
reg [17:0] A_tb,B_tb,D_tb,BCIN_tb;
reg [47:0] C_tb,PCIN_tb;
reg CARRYIN_tb,CLK_tb,CEA_tb,CEB_tb,CEC_tb,CECARRYIN_tb,CED_tb,CEM_tb,CEOPMODE_tb,CEP_tb,RSTA_tb,RSTB_tb,RSTC_tb,RSTCARRYIN_tb,RSTD_tb,RSTM_tb,RSTOPMODE_tb,
reg [7:0] OPMODE_tb;
wire CARRYOUT_tb,CARRYOUTF_tb;
wire [35:0] M_tb;
wire [47:0] P_tb,PCOUT_tb;
wire [17:0] BCOUT_tb;

parameter A0REG_tb=0,A1REG_tb=1,B0REG_tb=0,B1REG_tb=1,CREG_tb=1,DREG_tb=1,MREG_tb=1,PREG_tb=1,CARRYINREG_tb=1,CARRYOUTREG_tb=1,OPMODEREG_tb=1,CARRYINSEL_tb=

    DSP48A1#(A0REG_tb,A1REG_tb,B0REG_tb,B1REG_tb,CREG_tb,DREG_tb,MREG_tb,PREG_tb,CARRYINREG_tb,CARRYOUTREG_tb,OPMODEREG_tb,CARRYINSEL_tb,B_INPUT_tb,RSTTYPE_

    initial begin
        CLK_tb=0;
        forever begin
            #2 CLK_tb=~CLK_tb;
        end
 end
 initial begin
     RSTA_tb=1;RSTB_tb=1;RSTM_tb=1;RSTP_tb=1;RSTC_tb=1;RSTD_tb=1;RSTCARRYIN_tb=1;RSTOPMODE_tb=1;
    #20; // testin the rst signals
RSTA_tb=0;RSTB_tb=0;RSTM_tb=0;RSTP_tb=0;RSTC_tb=0;RSTD_tb=0;RSTCARRYIN_tb=0;RSTOPMODE_tb=0;

CEA_tb=1;CEB_tb=1;CEM_tb=1;CEP_tb=1;CEC_tb=1;CED_tb=1;CECARRYIN_tb=1;CEOPMODE_tb=1;
    A_tb=10;
    B_tb=11;
    C_tb=12;
    D_tb=13;
    CARRYIN_tb=0;
    OPMODE_tb=8'b00011101;
    #20;
    OPMODE_tb=8'b10001101;
#20;
$stop;
 end
endmodule
```


F: > Digtal_temp > Sessions > DSP_Project > Constraints_basys3.xdc
Ln 8, Col 82

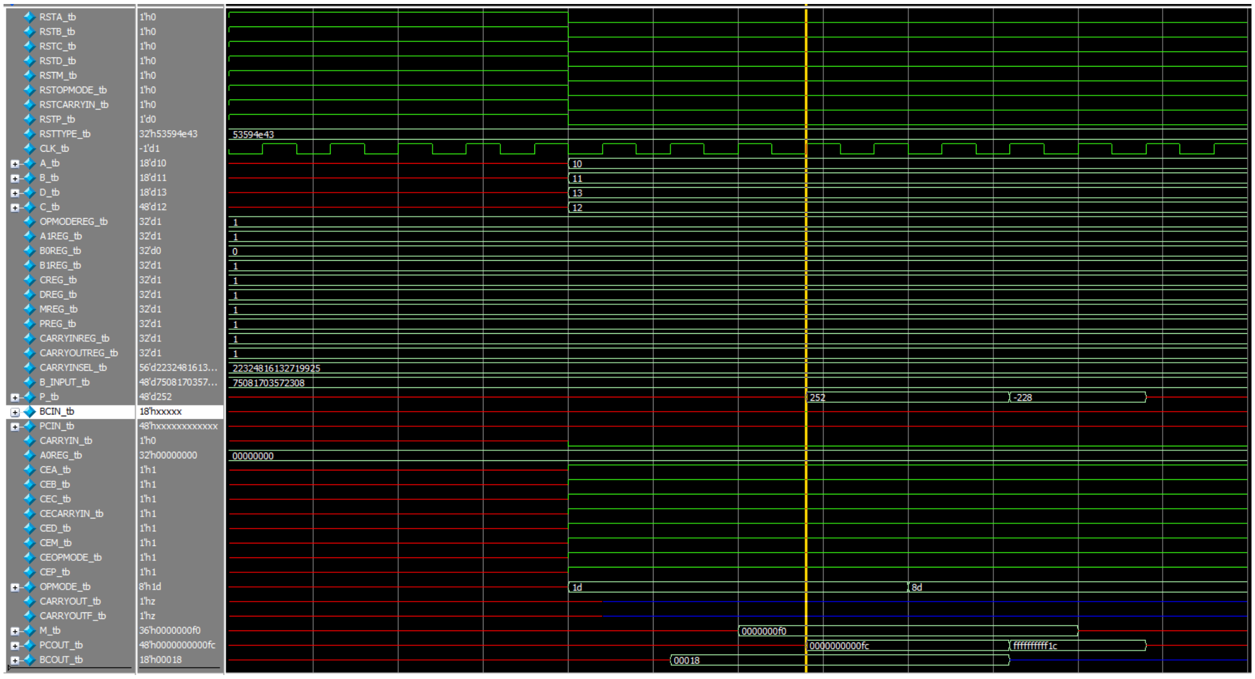
```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

## 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]}]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
#set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
#set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
```



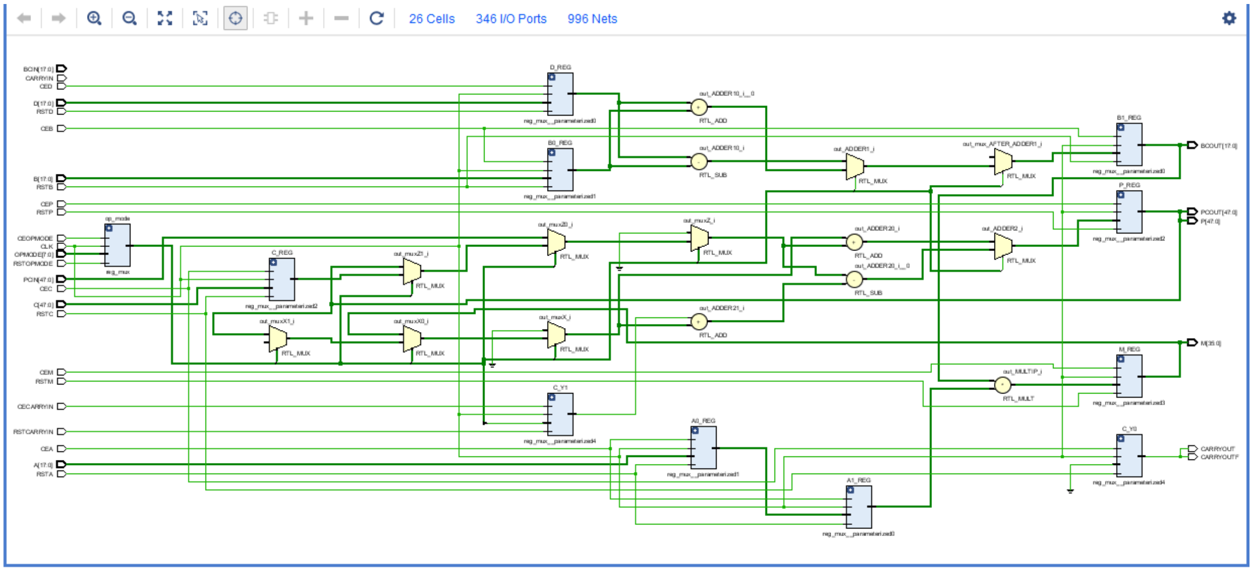
```
#set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports {led[15]}]


##7 Segment Display
#set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
#set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
#set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
#set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
#set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
#set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
#set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]

#set_property -dict { PACKAGE_PIN V7   IOSTANDARD LVCMOS33 } [get_ports dp]

#set_property -dict { PACKAGE_PIN U2   IOSTANDARD LVCMOS33 } [get_ports {an[0]}]
#set_property -dict { PACKAGE_PIN U4   IOSTANDARD LVCMOS33 } [get_ports {an[1]}]
#set_property -dict { PACKAGE_PIN V4   IOSTANDARD LVCMOS33 } [get_ports {an[2]}]
#set_property -dict { PACKAGE_PIN W4   IOSTANDARD LVCMOS33 } [get_ports {an[3]}]


##Buttons
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports rst]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports btnU]
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports btnL]
#set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports btnR]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports btnD]


##Pmod Header JA
#set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports {JA[0]}];#Sch name = JA1
#set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports {JA[1]}];#Sch name = JA2
#set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports {JA[2]}];#Sch name = JA3
#set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports {JA[3]}];#Sch name = JA4
#set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports {JA[4]}];#Sch name = JA7
#set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports {JA[5]}];#Sch name = JA8
#set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports {JA[6]}];#Sch name = JA9
#set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports {JA[7]}];#Sch name = JA10

##Pmod Header JB
#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports {JB[0]}];#Sch name = JB1
```

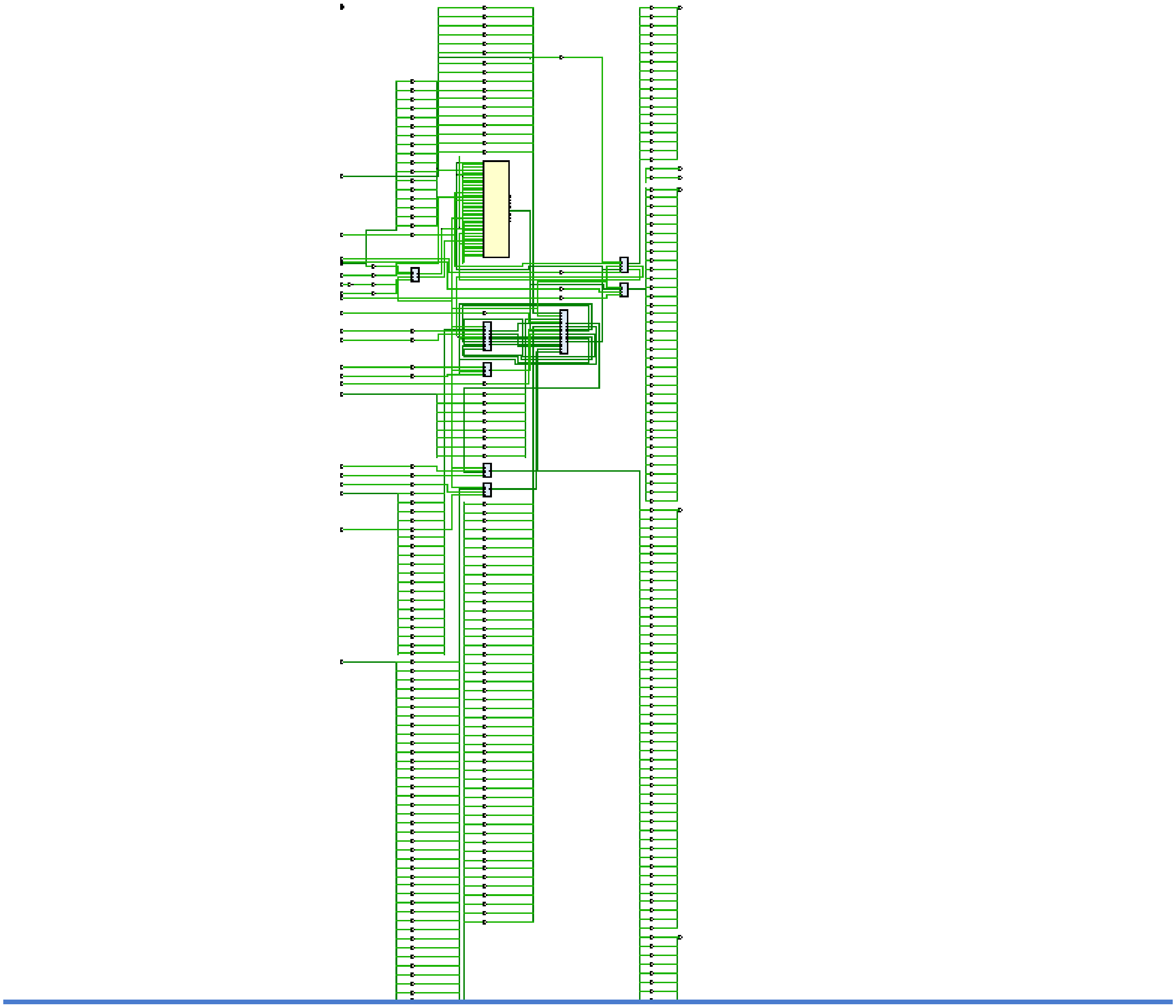


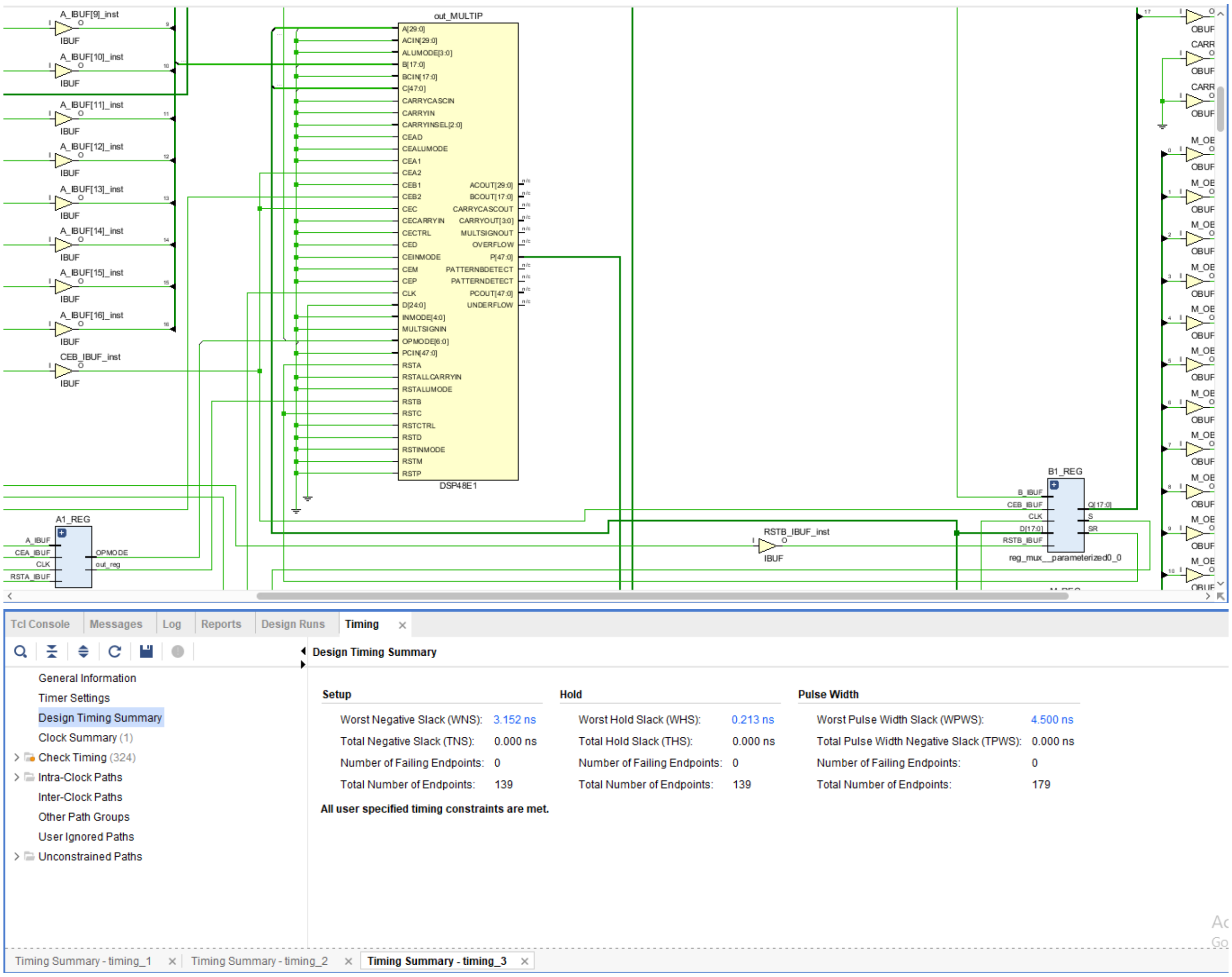
```
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports RsRx]
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports RsTx]


##USB HID (PS/2)
#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports PS2Clk]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the
##STARTUPE2 primitive.
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[0]}]
#set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[1]}]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[2]}]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[3]}]
#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports QspiCSn]


## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]

## SPI configuration mode options for QSPI boot, can be used for all designs
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
```



```
RSTA_tb           1'h0
RSTB_tb           1'h0
RSTC_tb           1'h0
RSTD_tb           1'h0
RSTM_tb           1'h0
RSTOPMODE_tb      1'h0
RSTCARRYIN_tb     1'h0
RSTP_tb           1'd0
RSTTYPE_tb        32'h53594e43
CLK_tb            -1'd1
A_tb              18'd10
B_tb              18'd11
D_tb              18'd13
C_tb              48'd12
OPMODEREG_tb      32'd1
A1REG_tb          32'd1
B0REG_tb          32'd0
B1REG_tb          32'd1
CREG_tb           32'd1
DREG_tb           32'd1
MREG_tb           32'd1
PREG_tb           32'd1
CARRYINREG_tb     32'd1
CARRYOUTREG_tb    32'd1
CARRYINSEL_tb     56'd2232481613...
B_INPUT_tb        48'd7508170357...
P_tb              48'd252
BCIN_tb           18'hxxxxx
PCIN_tb           48'hxxxxxxxxxxxx
CARRYIN_tb        1'h0
A0REG_tb          32'h00000000
CEA_tb            1'h1
CEB_tb            1'h1
CEC_tb            1'h1
CECARRYIN_tb      1'h1
CED_tb            1'h1
CEM_tb            1'h1
CEOPMODE_tb       1'h1
CEP_tb            1'h1
OPMODE_tb         8'h1d
CARRYOUT_tb       1'hz
CARRYOUTF_tb      1'hz
M_tb              36'h0000000f0
PCOUT_tb          48'h0000000000fc
BCOUT_tb          18'h00018
```


26 Cells
346 I/O Ports
996 Nets
BCIN[17:0]
CARRYIN
CED
D[17:0]
RSTD
CEB
B[17:0]
RSTB
CEP
RSTP
CEOPMODE
CLK
OPMODE[7:0]
RSTOPMODE
PCIN[47:0]
CEC
C[47:0]
RSTC
CEM
RSTM
CECARRYIN
RSTCARRYIN
CEA
A[17:0]
RSTA
op_mode
reg_mux
D_REG
reg_mux__parameterized0
B0_REG
reg_mux__parameterized1
C_REG
reg_mux__parameterized2
out_muxZ1_i
out_muxZ0_i
out_muxX1_i
out_muxX0_i
out_muxX_i
out_muxZ_i
RTL_MUX
C_Y1
reg_mux__parameterized4
A0_REG
reg_mux__parameterized1
A1_REG
reg_mux__parameterized0
out_ADDER10_i__0
RTL_ADD
out_ADDER10_i
RTL_SUB
out_ADDER1_i
out_mux_AFTER_ADDER1_i
out_ADDER20_i
out_ADDER20_i__0
out_ADDER2_i
out_ADDER21_i
out_MULTIP_i
RTL_MULT
B1_REG
reg_mux__parameterized0
P_REG
reg_mux__parameterized2
M_REG
reg_mux__parameterized3
C_Y0
reg_mux__parameterized4
BCOUT[17:0]
PCOUT[47:0]
P[47:0]
M[35:0]
CARRYOUT
CARRYOUTF





Tcl Console | Messages | Log | Reports | Design Runs | Timing ×

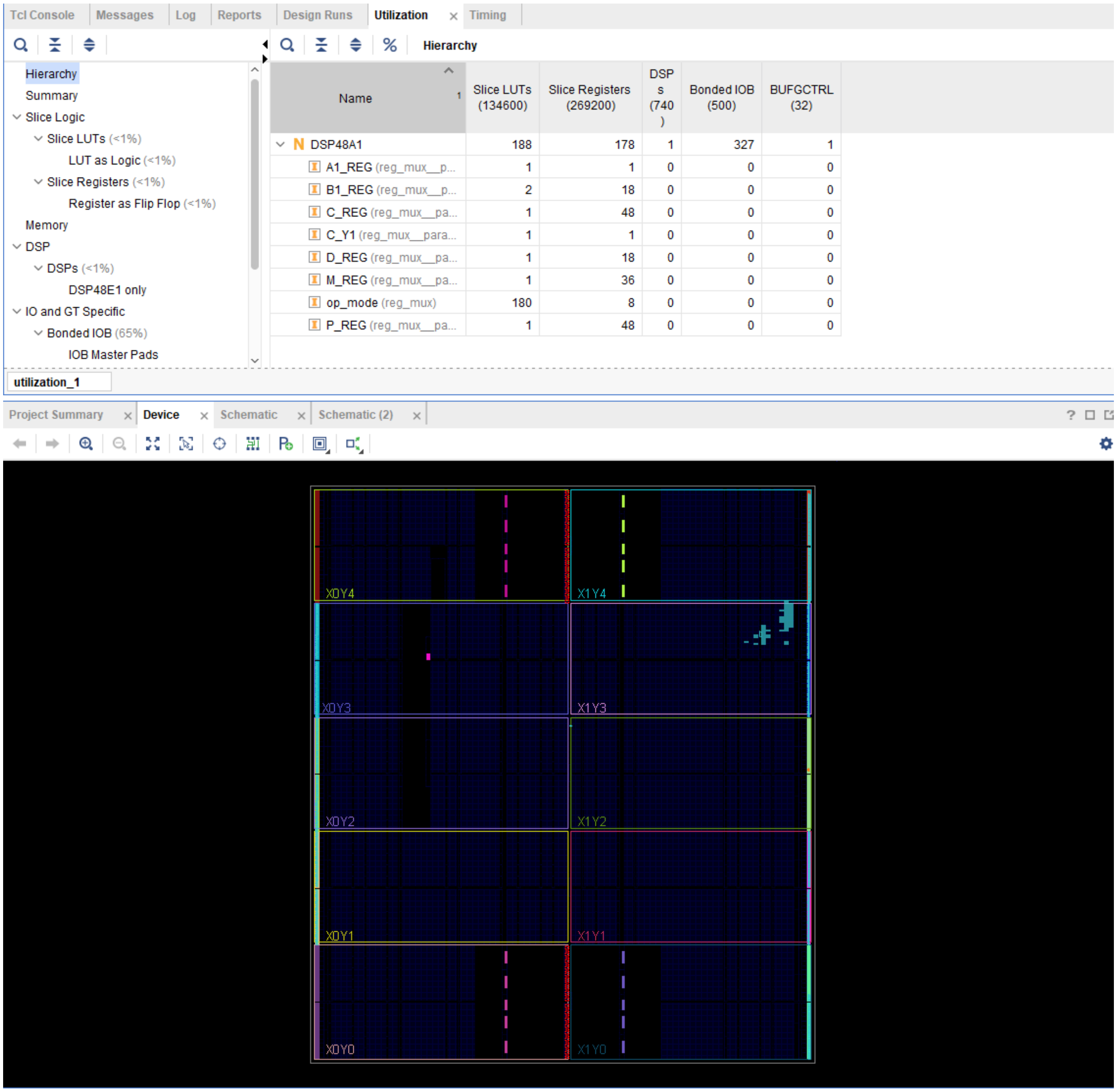
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (324)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.152 ns	Worst Hold Slack (WHS):	0.213 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:	139	Total Number of Endpoints:	139	Total Number of Endpoints:	179

All user specified timing constraints are met.

Timing Summary - timing_1 × | Timing Summary - timing_2 × | Timing Summary - timing_3 ×



- 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

Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
N DSP48A1	188	178	1	327	1
A1_REG (reg_mux__p...	1	1	0	0	0
B1_REG (reg_mux__p...	2	18	0	0	0
C_REG (reg_mux__pa...	1	48	0	0	0
C_Y1 (reg_mux__para...	1	1	0	0	0
D_REG (reg_mux__pa...	1	18	0	0	0
M_REG (reg_mux__pa...	1	36	0	0	0
op_mode (reg_mux)	180	8	0	0	0
P_REG (reg_mux__pa...	1	48	0	0	0

utilization_1