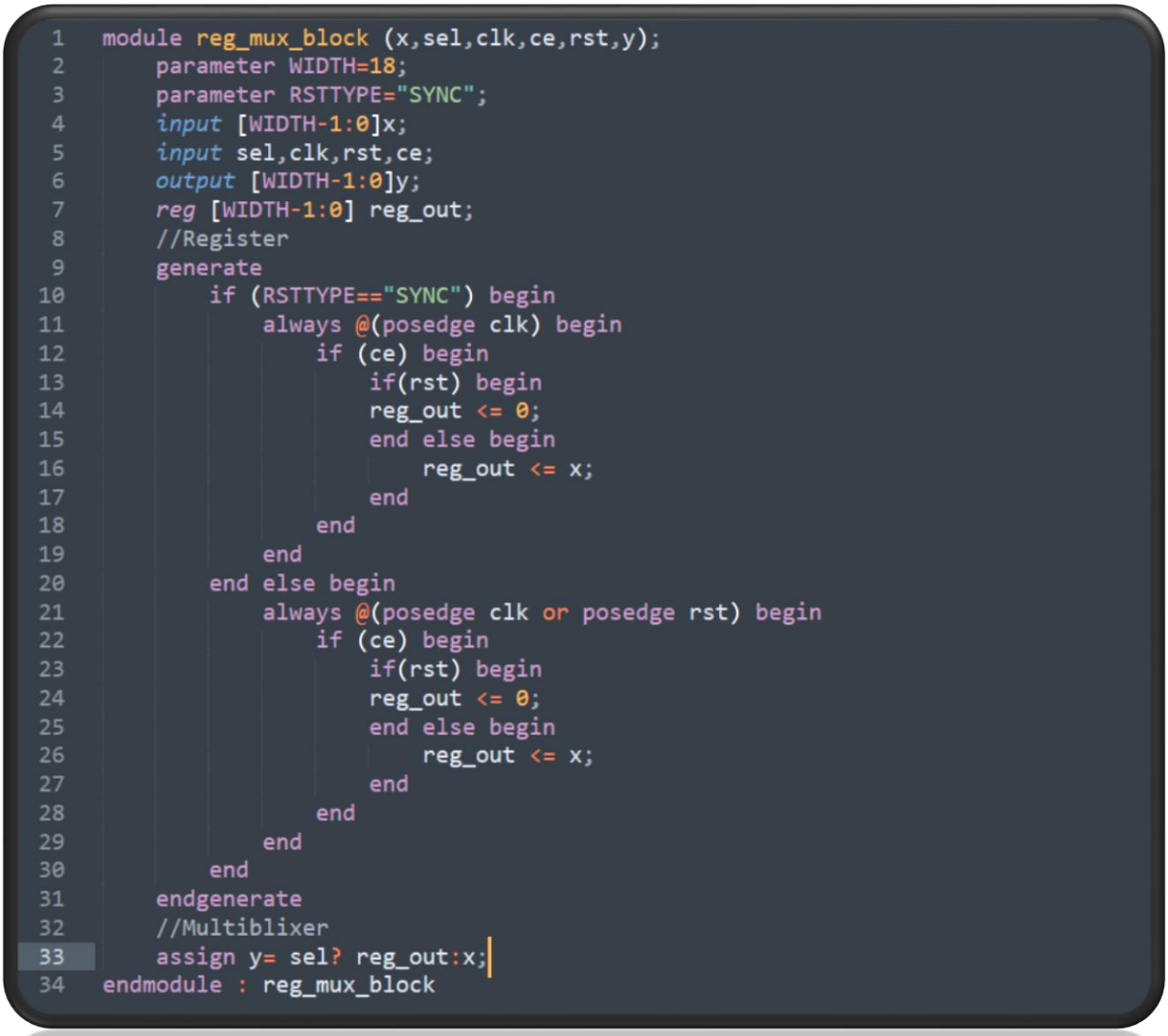
Spartan6_DSP48A1

RTL and FPGA design

RTL code

The internal register_mux block



```
module reg_mux_block (x,sel,clk,ce,rst,y);
    parameter WIDTH=18;
    parameter RSTTYPE="SYNC";
    input [WIDTH-1:0]x;
    input sel,clk,rst,ce;
    output [WIDTH-1:0]y;
    reg [WIDTH-1:0] reg_out;
    //Register
    generate
        if (RSTTYPE=="SYNC") begin
            always @(posedge clk) begin
                if (ce) begin
                    if(rst) begin
                    reg_out <= 0;
                    end else begin
                        reg_out <= x;
                    end
                end
            end
        end else begin
            always @(posedge clk or posedge rst) begin
                if (ce) begin
                    if(rst) begin
                    reg_out <= 0;
                    end else begin
                        reg_out <= x;
                    end
                end
            end
        end
    endgenerate
    //Multiblixer
    assign y= sel? reg_out:x;
endmodule : reg_mux_block
```

DSP RTL design

```
module DSP (clk,rsta,rstb,rstm,rstp,rstc,rstd,rstcarryin,rstopmode,cea,ceb,cem,cep,cec,ced,ceopmode,cecarryin,
carryin,opmode,a,b,d,c,bcin,pcin,bcout,pcout,p,m,carryout,carryoutf);
parameter A0REG=0;
parameter A1REG=1;
parameter B0REG=0;
parameter B1REG=1;
parameter CREG=1;
parameter DREG=1;
parameter MREG=1;
parameter PREG=1;
parameter CARRYINREG=1;
parameter CARRYOUTREG=1;
parameter OPMODEREG=1;
parameter CARRYINSEL="opmode5";
parameter B_INPUT="DIRECT";
input  [7:0]opmode;
input  [17:0]a,b,d,bcin;
input  [47:0]c,pcin;
input clk,rsta,rstb,rstm,rstp,rstc,rstd,rstcarryin,rstopmode,cea,ceb,cem,cep,cec,ced,cecarryin,ceopmode,carryin;
output  [17:0]bcout;
output  [47:0]p,pcout;
output  [35:0]m;
output  carryout,carryoutf;
// wires for internal signals
wire [7:0]opmodew;
wire [17:0]a0w,a1w,dw,b0w,b1w,pre_adder,b1_inw,b0_inw;
wire [35:0]mw,mult_outw;
wire [47:0]post_adder,cw;
wire cinw,cout,carry_cascade;
reg [47:0]x_out,z_out;
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//selections of internal multiplixers
always @(*) begin
    //Multiplixer X
    case (opmodew[1:0])
        0: x_out=0;
        1: x_out={12'h000,mw};
        2: x_out=pcout;
        3: x_out={dw[11:0], a1w[17:0],b1w[17:0]};
        default : x_out=0;
    endcase
    //Multiplixer Z
    case (opmodew[3:2])
        0: z_out=0;
        1: z_out=pcin;
        2: z_out=pcout;
        3: z_out=cw;
        default : z_out=0;
    endcase
end
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//inistantiation of the reg_mux blocks for inputs and outputs and internal signals
reg_mux_block #(.WIDTH(18),.RSTTYPE("SYNC")) D (.x(d),.sel(DREG),.clk(clk),.ce(ced),.rst(rstd),.y(dw));
reg_mux_block #(.WIDTH(18),.RSTTYPE("SYNC")) B0 (.x(b0_inw),.sel(B0REG),.clk(clk),.ce(ceb),.rst(rstb),.y(b0w));
reg_mux_block #(.WIDTH(18),.RSTTYPE("SYNC")) A0 (.x(a),.sel(A0REG),.clk(clk),.ce(cea),.rst(rsta),.y(a0w));
reg_mux_block #(.WIDTH(48),.RSTTYPE("SYNC")) C (.x(c),.sel(CREG),.clk(clk),.ce(cec),.rst(rstc),.y(cw));
reg_mux_block #(.WIDTH(8),.RSTTYPE("SYNC")) OPMODE (.x(opmode),.sel(OPMODEREG),.clk(clk),.ce(ceopmode),.rst(rstopmode),.y(opmodew));
reg_mux_block #(.WIDTH(18),.RSTTYPE("SYNC")) B1 (.x(b1_inw),.sel(B1REG),.clk(clk),.ce(ceb),.rst(rstb),.y(b1w));
reg_mux_block #(.WIDTH(18),.RSTTYPE("SYNC")) A1 (.x(a0w),.sel(A1REG),.clk(clk),.ce(cea),.rst(rsta),.y(a1w));
reg_mux_block #(.WIDTH(1),.RSTTYPE("SYNC")) CYI (.x(carry_cascade),.sel(CARRYINREG),.clk(clk),.ce(cecarryin),.rst(rstcarryin),.y(cinw));
reg_mux_block #(.WIDTH(36),.RSTTYPE("SYNC")) M (.x(mult_outw),.sel(MREG),.clk(clk),.ce(cem),.rst(rstm),.y(mw));
reg_mux_block #(.WIDTH(48),.RSTTYPE("SYNC")) P (.x(post_adder),.sel(PREG),.clk(clk),.ce(cep),.rst(rstp),.y(p));
reg_mux_block #(.WIDTH(1),.RSTTYPE("SYNC")) CYO (.x(cout),.sel(CARRYOUTREG),.clk(clk),.ce(cecarryin),.rst(rstcarryin),.y(carryout));
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
64  ////////////////////////////////////////////////////////////////////////////////////////////////////////
65  //B0 input selection
66  assign b0_inw=(B_INPUT=="DIRECT")? b:(B_INPUT=="CASCADE")?bcin:0;
67  //B1 input selection
68  assign b1_inw=(opmodew[4])? pre_adder:b0w;
69  //Carry in selection
70  assign carry_cascade=(CARRYINSEL=="opmode5")? opmodew[5]:(B_INPUT=="CARRYIN")?carryin:0;
71  //pre Adder_Subtractor
72  assign pre_adder=(opmodew[6])? dw-b0w:dw+b0w;
73  //post Adder_Subtractor
74  assign {carryout,post_adder}=(opmodew[7])? z_out-(x_out+cinw):z_out+x_out+cinw;
75  // Internal Multipier
76  assign mult_outw=b1w*a1w;
77  // continoues assignments for outputs
78  assign m=mw;
79  assign pcout=p;
80  assign bcout=b1w;
81  assign carryoutf=carryout;
82  endmodule : DSP
```

Testbench

```
module DSP_tb ();
    reg [7:0]opmode;
    reg [17:0]a,b,d,bcin;
    reg [47:0]c,pcin;
    reg clk,rsta,rstb,rstm,rstp,rstc,rstd,rstcarryin,rstopmode,cea,ceb,cem,cep,cec,ced,cecarryin,ceopmode,carryin;
    wire  [17:0]bcout;
    wire  [47:0]p,pcout;
    wire  [35:0]m;
    wire  carryout,carryoutf;
    // expected values to compare with
    reg  [17:0]bcout_expected;
    reg  [47:0]p_expected,pcout_expected;
    reg  [35:0]m_expected;
    reg  carryout_expected,carryoutf_expected;
    // inistantiation of our RTL design
    DSP dut (clk,rsta,rstb,rstm,rstp,rstc,rstd,rstcarryin,rstopmode,cea,ceb,cem,cep,cec,ced,ceopmode,cecarryin,
    carryin,opmode,a,b,d,c,bcin,pcin,bcout,pcout,p,m,carryout,carryoutf);
    // clock
    initial begin
        clk=0;
        forever begin
            #5; clk=~clk;
        end
    end
    initial begin
```

```
    initial begin
        // test the rst of the input registers
        rsta=1; rstb=1; rstc=1; rstd=1; rstcarryin=1; rstopmode=0; rstp=0; rstm=0; // resets
        a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b00111111; // inputs
        cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables

        bcout_expected=0; p_expected=0; pcout_expected=0; // expected values
        m_expected=0; carryout_expected=0; carryoutf_expected=0;
        @(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
        if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
        ||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
            $display("There is an error");
            $stop;
        end
```

```
    ////////////////////////////////////////////////////////////////////////////////////////////////
    // test the rst of the output registers
    rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=1; rstopmode=0; rstp=1; rstm=1; // resets
    a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b00111111; // inputs
    cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables
    bcout_expected=60; p_expected=0; pcout_expected=0; // expected values
    m_expected=0; carryout_expected=0; carryoutf_expected=0;
    @(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
    if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
    ||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
        $display("There is an error");
        $stop;
    end
```

```
    ////////////////////////////////////////////////////////////////////////////////////////////////
    // test of the clk enables
    rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=0; rstopmode=0; rstp=0; rstm=0; // resets
    a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b00111111; // inputs
    cea=0; ceb=0; cem=0; cep=0; cec=0; ced=0; cecarryin=0; ceopmode=0; // clk enables
    bcout_expected=60; p_expected=0; pcout_expected=0; // expected values
    m_expected=0; carryout_expected=0; carryoutf_expected=0;
    @(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
    if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
    ||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
        $display("There is an error");
        $stop;
    end
```

```
////////////////////////////////////////////////////////////////////////////////////////////////
// test of the outputs with diffrent operation modes
rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=0; rstopmode=0; rstp=0; rstm=0; // resets
a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b00000000; // inputs
cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables
bcout_expected=20; p_expected=0; pcout_expected=0; // expected values
m_expected=200; carryout_expected=0; carryoutf_expected=0;
@(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
    $display("There is an error");
    $stop;
end
```

```
    ////////////////////////////////////////////////////////////////////////////////////////////////
    // test the rst of the output registers
    rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=1; rstopmode=0; rstp=1; rstm=1; // resets
    a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b00111111; // inputs
    cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables
    bcout_expected=60; p_expected=0; pcout_expected=0; // expected values
    m_expected=0; carryout_expected=0; carryoutf_expected=0;
    @(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
    if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
    ||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
        $display("There is an error");
        $stop;
    end
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////
// test the carryin (OPMODE5)
rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=0; rstopmode=0; rstp=0; rstm=0; // resets
a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b00100101; // inputs
cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables
bcout_expected=20; p_expected=251; pcout_expected=251; // expected values
m_expected=200; carryout_expected=0; carryoutf_expected=0;
@(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
    $display("There is an error");
    $stop;
end
////////////////////////////////////////////////////////////////////////////////////////////////////
```

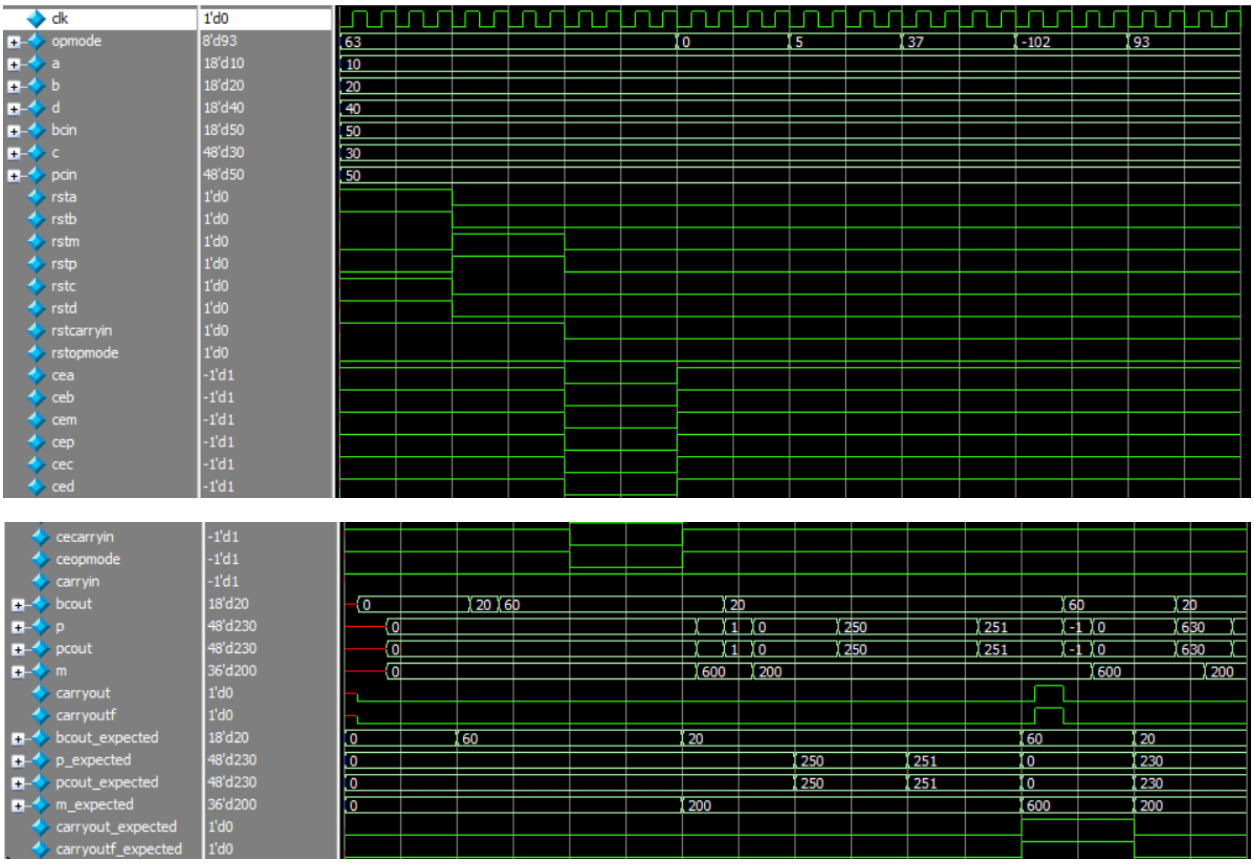
```
////////////////////////////////////////////////////////////////////////////////////////////////////
// test the post adder subtractor and the carry out
rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=0; rstopmode=0; rstp=0; rstm=0; // resets
a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b10011010; // inputs
cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables
bcout_expected=60; p_expected=0; pcout_expected=0; // expected values
m_expected=600; carryout_expected=1; carryoutf_expected=1;
@(negedge clk);
if (carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
    $display("There is an error");
    $stop;
end
@(negedge clk); @(negedge clk) @(negedge clk);
if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m) begin
    $display("There is an error");
    $stop;
end
////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
20          ////////////////////////////////////////////////////////////////////////////////////////////////////
21          // test the pre adder subtractor (subtract) and the c port
22          rsta=0; rstb=0; rstc=0; rstd=0; rstcarryin=0; rstopmode=0; rstp=0; rstm=0; // resets
23          a=10; b=20; c=30; d=40; bcin=50; pcin=50; carryin=1; opmode=8'b01011101; // inputs
24          cea=1; ceb=1; cem=1; cep=1; cec=1; ced=1; cecarryin=1; ceopmode=1; // clk enables
25          bcout_expected=20; p_expected=230; pcout_expected=230; // expected values
26          m_expected=200; carryout_expected=0; carryoutf_expected=0;
27          @(negedge clk); @(negedge clk); @(negedge clk); @(negedge clk);// 4 clk cycle
28          if (bcout_expected!=bcout||p_expected!=p||pcout_expected!=pcout||m_expected!=m
29          ||carryout_expected!=carryout||carryoutf_expected!=carryoutf) begin
30              $display("There is an error");
31              $stop;
32          end
33      $stop;
34      end
35  endmodule : DSP_tb
```

Do file

```
1  vlib work
2  vlog DSP.v DSP_tb.v
3  vsim -voptargs=+acc work.DSP_tb
4  add wave *
5  run -all
6  #quit -sim
```

Simulation


clk 1'd0
opmode 8'd93 63 0 5 37 -102 93
a 18'd10 10
b 18'd20 20
d 18'd40 40
bcin 18'd50 50
c 48'd30 30
pcin 48'd50 50
rsta 1'd0
rstb 1'd0
rstm 1'd0
rstp 1'd0
rstc 1'd0
rstd 1'd0
rstcarryin 1'd0
rstopmode 1'd0
cea -1'd1
ceb -1'd1
cem -1'd1
cep -1'd1
cec -1'd1
ced -1'd1
cecarryin -1'd1
ceopmode -1'd1
carryin -1'd1
bcout 18'd20 0 20 60 20 60 20
p 48'd230 0 1 0 250 251 -1 0 630
pcout 48'd230 0 1 0 250 251 -1 0 630
m 36'd200 0 600 200 600 200
carryout 1'd0
carryoutf 1'd0
bcout_expected 18'd20 0 60 20 60 20
p_expected 48'd230 0 250 251 0 230
pcout_expected 48'd230 0 250 251 0 230
m_expected 36'd200 0 200 600 200
carryout_expected 1'd0
carryoutf_expected 1'd0

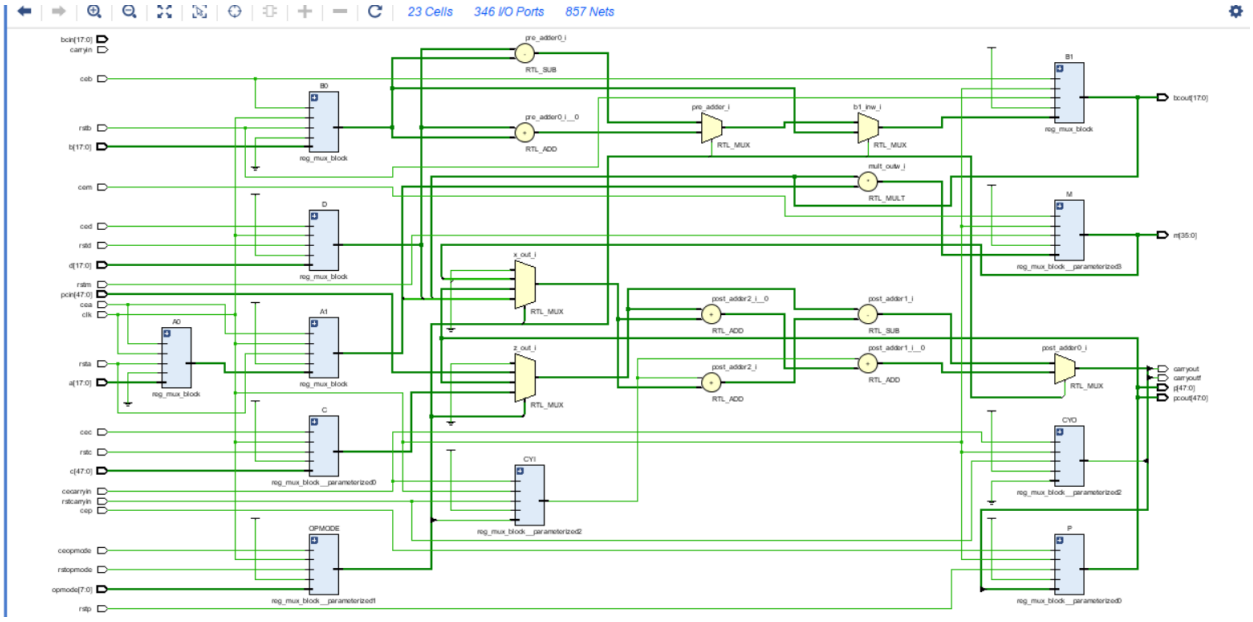
Constrain file and debug core

```
## 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 proje

## Clock signal
set_property -dict {PACKAGE_PIN W5 IOSTANDARD LVCMOS33} [get_ports clk]
create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [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]
```

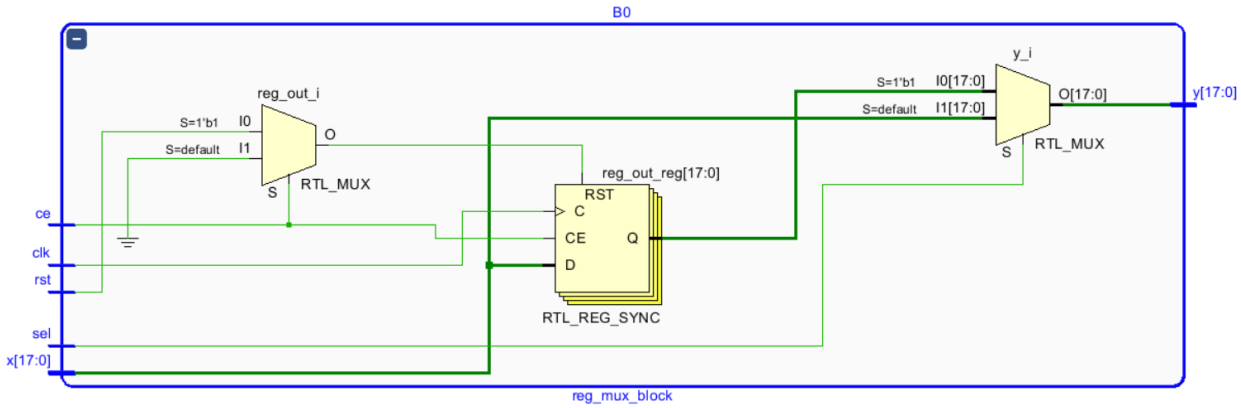
```

create_debug_core u_ila_0 ila
set_property ALL_PROBE_SAME_MU true [get_debug_cores u_ila_0]
set_property ALL_PROBE_SAME_MU_CNT 1 [get_debug_cores u_ila_0]
set_property C_ADV_TRIGGER false [get_debug_cores u_ila_0]
set_property C_DATA_DEPTH 1024 [get_debug_cores u_ila_0]
set_property C_EN_STRG_QUAL false [get_debug_cores u_ila_0]
set_property C_INPUT_PIPE_STAGES 0 [get_debug_cores u_ila_0]
set_property C_TRIGIN_EN false [get_debug_cores u_ila_0]
set_property C_TRIGOUT_EN false [get_debug_cores u_ila_0]
set_property port_width 1 [get_debug_ports u_ila_0/clk]
connect_debug_port u_ila_0/clk [get_nets [list clk_IBUF_BUFG]]
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe0]
set_property port_width 18 [get_debug_ports u_ila_0/probe0]
connect_debug_port u_ila_0/probe0 [get_nets [list {bcout_OBUF[0]} {bcout_OBUF[1]} {bcout_OBUF[2]} {bcout_OBUF[3]} {bcout_OBUF[4]} {bcout_OBUF
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe1]
set_property port_width 48 [get_debug_ports u_ila_0/probe1]
connect_debug_port u_ila_0/probe1 [get_nets [list {c_IBUF[0]} {c_IBUF[1]} {c_IBUF[2]} {c_IBUF[3]} {c_IBUF[4]} {c_IBUF[5]} {c_IBUF[6]} {c_IBUF
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe2]
set_property port_width 36 [get_debug_ports u_ila_0/probe2]
connect_debug_port u_ila_0/probe2 [get_nets [list {m_OBUF[0]} {m_OBUF[1]} {m_OBUF[2]} {m_OBUF[3]} {m_OBUF[4]} {m_OBUF[5]} {m_OBUF[6]} {m_OBUF
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe3]
set_property port_width 48 [get_debug_ports u_ila_0/probe3]
connect_debug_port u_ila_0/probe3 [get_nets [list {pcin_IBUF[0]} {pcin_IBUF[1]} {pcin_IBUF[2]} {pcin_IBUF[3]} {pcin_IBUF[4]} {pcin_IBUF[5]} {
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe4]
set_property port_width 48 [get_debug_ports u_ila_0/probe4]
connect_debug_port u_ila_0/probe4 [get_nets [list {p_OBUF[0]} {p_OBUF[1]} {p_OBUF[2]} {p_OBUF[3]} {p_OBUF[4]} {p_OBUF[5]} {p_OBUF[6]} {p_OBUF
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe5]
set_property port_width 18 [get_debug_ports u_ila_0/probe5]
```

```
set_property port_width 1 [get_debug_ports u_ila_0/probe10]
connect_debug_port u_ila_0/probe10 [get_nets [list ceb_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe11]
set_property port_width 1 [get_debug_ports u_ila_0/probe11]
connect_debug_port u_ila_0/probe11 [get_nets [list cec_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe12]
set_property port_width 1 [get_debug_ports u_ila_0/probe12]
connect_debug_port u_ila_0/probe12 [get_nets [list cecarryin_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe13]
set_property port_width 1 [get_debug_ports u_ila_0/probe13]
connect_debug_port u_ila_0/probe13 [get_nets [list ced_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe14]
set_property port_width 1 [get_debug_ports u_ila_0/probe14]
connect_debug_port u_ila_0/probe14 [get_nets [list cem_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe15]
set_property port_width 1 [get_debug_ports u_ila_0/probe15]
connect_debug_port u_ila_0/probe15 [get_nets [list ceopmode_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe16]
set_property port_width 1 [get_debug_ports u_ila_0/probe16]
connect_debug_port u_ila_0/probe16 [get_nets [list cep_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe17]
set_property port_width 1 [get_debug_ports u_ila_0/probe17]
connect_debug_port u_ila_0/probe17 [get_nets [list clk_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe18]
set_property port_width 1 [get_debug_ports u_ila_0/probe18]
connect_debug_port u_ila_0/probe18 [get_nets [list rsta_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe19]
set_property port_width 1 [get_debug_ports u_ila_0/probe19]
connect_debug_port u_ila_0/probe19 [get_nets [list rstb_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe20]
set_property port_width 1 [get_debug_ports u_ila_0/probe20]
connect_debug_port u_ila_0/probe20 [get_nets [list rstc_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe21]
set_property port_width 1 [get_debug_ports u_ila_0/probe21]
connect_debug_port u_ila_0/probe21 [get_nets [list rstcarryin_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe22]
set_property port_width 1 [get_debug_ports u_ila_0/probe22]
connect_debug_port u_ila_0/probe22 [get_nets [list rstd_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe23]
set_property port_width 1 [get_debug_ports u_ila_0/probe23]
connect_debug_port u_ila_0/probe23 [get_nets [list rstm_IBUF]]
```

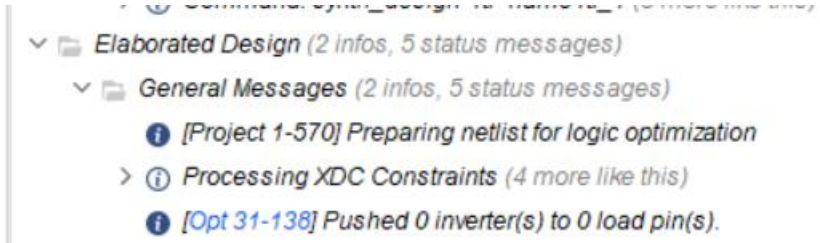
Elaboration



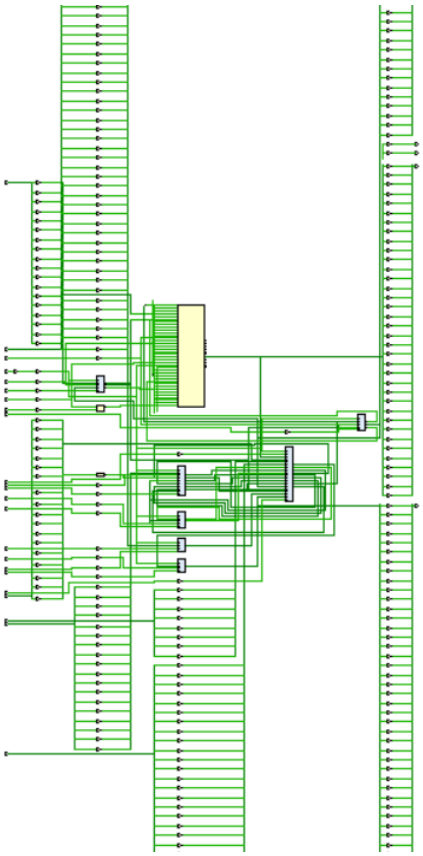
Internal block



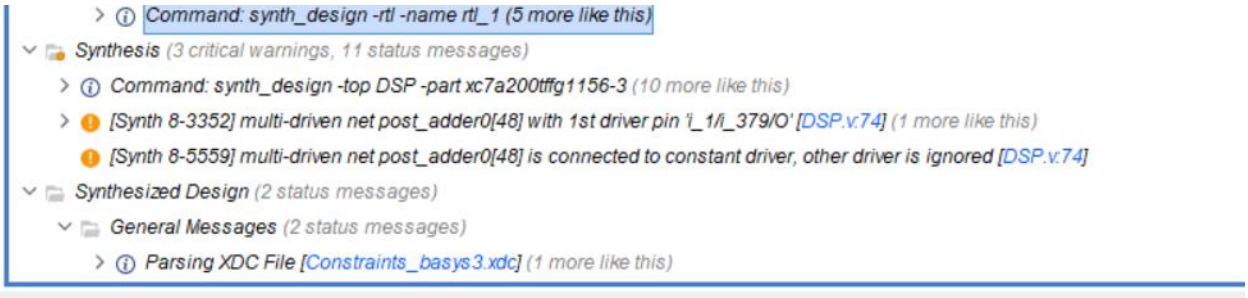
Messages



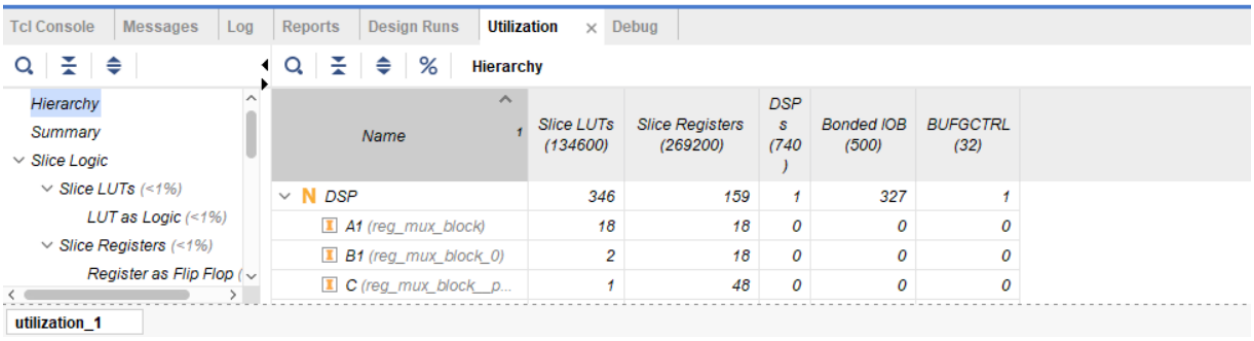
Synthesis



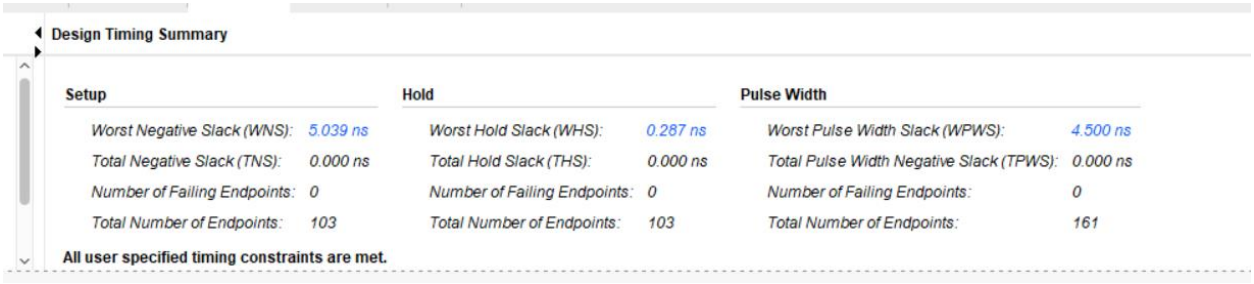
Messages



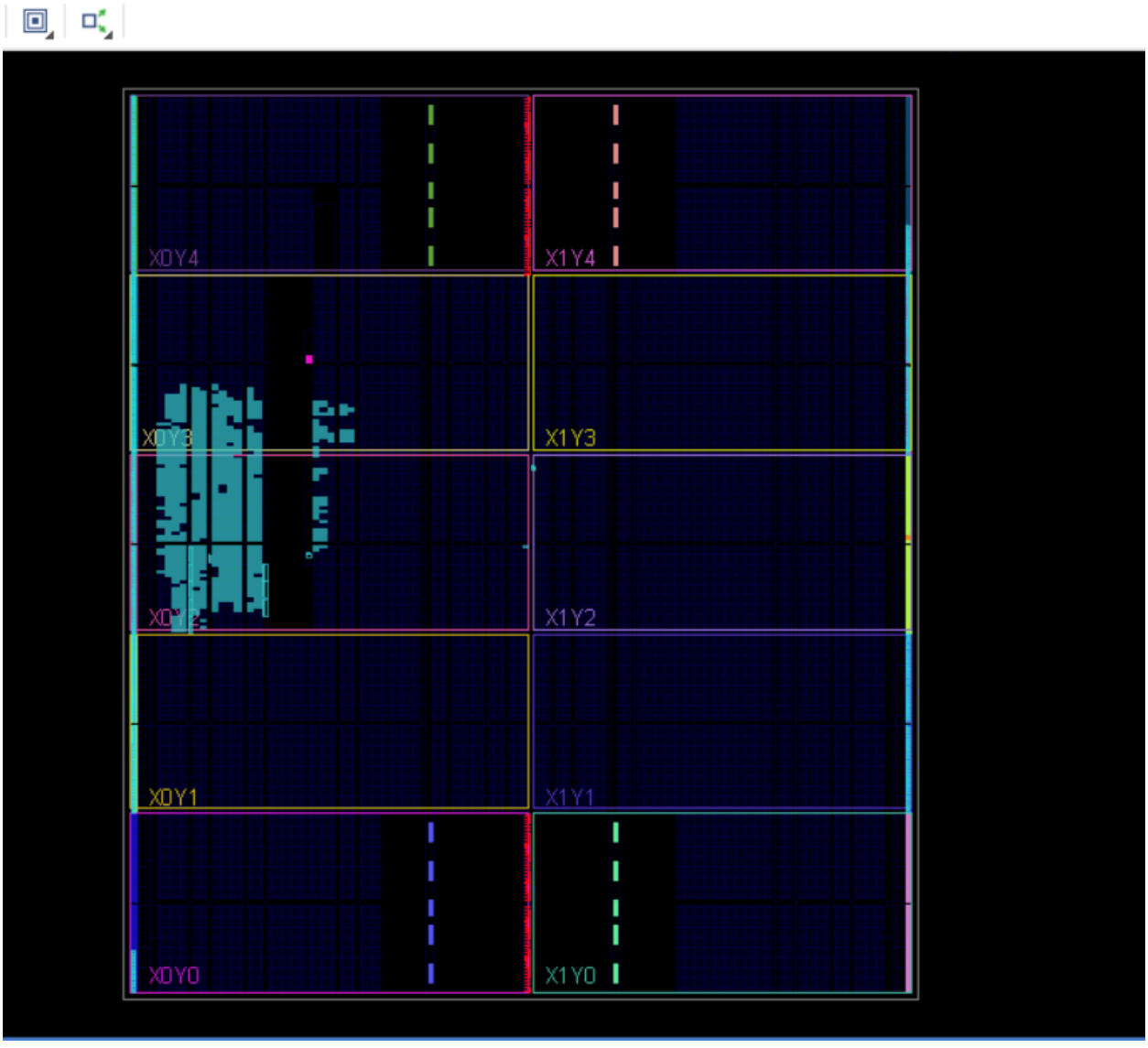
Utilization report



Timing



Implementation



Messages

- Implementation (234 status messages)
  - Design Initialization (7 status messages)
    - Command: link_design -top DSP -part xc7a200tffg1156-3 (6 more like this)
  - Opt Design (54 status messages)
    - Command: opt_design (53 more like this)
  - Place Design (90 status messages)
    - Command: place_design (89 more like this)
  - Route Design (83 status messages)

Utilization report

Hierarchy

1	Slice LUTs (133800)	Slice Registers (267600)	F7 Muxes (66900)	F8 Muxes (33450)	Slice (33450)	LUT as Logic (133800)	LUT as Memory (46200)	LUT Flip Flop Pairs (133800)	Block RAM Tile (365)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)	E
	2767	4174	97	11	1301	2298	469	1538	8	1	327	2	
	14	18	0	0	13	14	0	0	0	0	0	0	
0)	2	18	0	0	9	2	0	0	0	0	0	0	
	1	48	0	0	21	1	0	0	0	0	0	0	

Timing

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.099 ns	Worst Hold Slack (WHS):	0.058 ns	Worst Pulse Width Slack (WPWS):	3.950 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:	7964	Total Number of Endpoints:	7948	Total Number of Endpoints:	5075

All user specified timing constraints are met.