Project

RTL code:

-FF:

```
module dff_mux #(

    parameter size = 18 ,
    parameter pipeline = 0 ,
    parameter RSTTYPE = "SYNC"
    )(
    input[size - 1 : 0]  in,
    input clk ,
    input EN ,
    input rst ,
    output reg[size - 1 : 0] out
    );
    generate
    if(pipeline)begin
        if(RSTTYPE == "SYNC")  begin
            always @(posedge clk ) begin
                if (rst) begin
                    out <= 0;
                end
                else begin
                    if(EN) begin
                        out <= in;//pipelining enabled
                    end
                end
            end
        end
        else begin
            always @(posedge clk or posedge rst) begin
                if (rst) begin
                    out <= 0;
                end
                else begin
                    if(EN) begin
                        out <= in;//pipelining enabled
                    end
                end
            end
        end
    end
    else always@(in) out = in;
endgenerate




endmodule
```

PortA:

```
module PortA(input[17:0] A,
             input CLK,
             input CEA,
             input RSTA,
             output[17:0] outA1);

wire[17:0] outA0;

dff_mux #(.size(18),.pipeline(0)) A0REG(.in(A),.clk(CLK),.EN(CEA),.rst(RSTA),.out(outA0));
dff_mux #(.size(18),.pipeline(1)) A1REG(.in(outA0),.clk(CLK),.EN(CEA),.rst(RSTA),.out(outA1));

endmodule
```

PortB&D:

```
module PortB_D#(parameter B_INPUT = "DIRECT")
               (input[17:0] B,
               input[17:0] D,
               input[17:0] BCIN,
               input OPMODE4,
               input OPMODE6,
               input CLK,
               input CEB,
               input CED,
               input RSTB,
               input RSTD,
               output[17:0] outB1);


wire[17:0] outD;
wire[17:0] outB0;
wire[17:0] preadderout;
wire[17:0] B1REGin;
wire[17:0] out;

generate
    if(B_INPUT == "DIRECT") dff_mux #(.size(18),.pipeline(0)) B0REG(.in(B),.clk(CLK),.EN(CEB),.rst(RSTB),.out(outB0));
    else dff_mux #(.size(18),.pipeline(0)) B0REG(.in(BCIN),.clk(CLK),.EN(CEB),.rst(RSTB),.out(outB0));

endgenerate

dff_mux #(.size(18),.pipeline(1)) DREG(.in(D),.clk(CLK),.EN(CED),.rst(RSTD),.out(outD));

assign preadderout = OPMODE6?outD-outB0:outD+outB0;

assign B1REGin = OPMODE4?preadderout:outB0;

dff_mux #(.size(18),.pipeline(1)) B1REG(.in(B1REGin),.clk(CLK),.EN(CEB),.rst(RSTB),.out(out));

assign outB1 = out;


endmodule
```

PortC:

```
module PortC(input[47:0] C,
             input CLK,
             input CEC,
             input RSTC,
             output[47:0] outC);


dff_mux #(.size(48),.pipeline(1)) CREG(.in(C),.clk(CLK),.EN(CEC),.rst(RSTC),.out(outC));

endmodule
```

PortOPMODE:

```
module PortOPMODE(input[7:0] OPMODE,
                  input CLK,
                  input CEOPMODE,
                  input RSTOPMODE,
                  output[7:0] opmodeout);


dff_mux #(.size(8),.pipeline(1)) OPMODEREG(.in(OPMODE),.clk(CLK),.EN(CEOPMODE),.rst(RSTOPMPDE),.out(opmodeout));

endmodule
```

PortM:

```
module PortM(input[17:0] A,
             input[17:0] B,
             input[17:0] D,
             input[17:0] BCIN,
             input CLK,
             input CEM,
             input RSTM,
             input OPMODE4,
             input OPMODE6,
             input CEB,
             input CED,
             input RSTB,
             input RSTD,
             input CEA,
             input RSTA,
             output[17:0] BCOUT,
             output[35:0] xmux);


wire[17:0] outB1;


wire [17:0] outA1;
wire [35:0] MREGin;
wire [35:0] MREGout;

PortB_D Minput1(.B(B),.D(D),.BCIN(BCIN),.CLK(CLK),.CEB(CEB),.CED(CED),.RSTB(RSTB),.RSTD(RSTD),
                .OPMODE4(OPMODE4),.OPMODE6(OPMODE6),.outB1(outB1));

assign BCOUT = outB1;

PortA Minput2(.A(A),.CLK(CLK),.CEA(CEA),.RSTA(RSTA),.outA1(outA1));

assign MREGin = outA1*outB1;

dff_mux #(.size(36),.pipeline(1)) mREG(.in(MREGin),.clk(CLK),.EN(CEM),.rst(RSTM),.out(MREGout));

assign xmux = MREGout;

endmodule
```

PortP:

```
module PortP(input[47:0] Postaddsub,
             input CLK,
             input CEP,
             input RSTP,
             output[47:0] outP);


dff_mux #(.size(48),.pipeline(1)) PREG(.in(Postaddsub),.clk(CLK),.EN(CEP),.rst(RSTP),.out(outP));

endmodule
```

CYIN:

```
module Carryin #(parameter CARRYINSEL = "OPMODE5")
                (input CARRYIN,
                input OPMODE5,
                input CLK,
                input CECARRYIN,
                input RSTCARRYIN,
                output CIN);

generate
    if(CARRYINSEL == "OPMODE5") dff_mux #(.size(1),.pipeline(1)) CARRYINREG(.in(OPMODE5),.clk(CLK),.EN(CECARRYIN),.rst(RSTCARRYIN),.out(CIN));
    else dff_mux #(.size(1),.pipeline(1)) CARRYINREG(.in(CARRYIN),.clk(CLK),.EN(CECARRYIN),.rst(RSTCARRYIN),.out(CIN));
endgenerate

endmodule
```

DSPSlide:

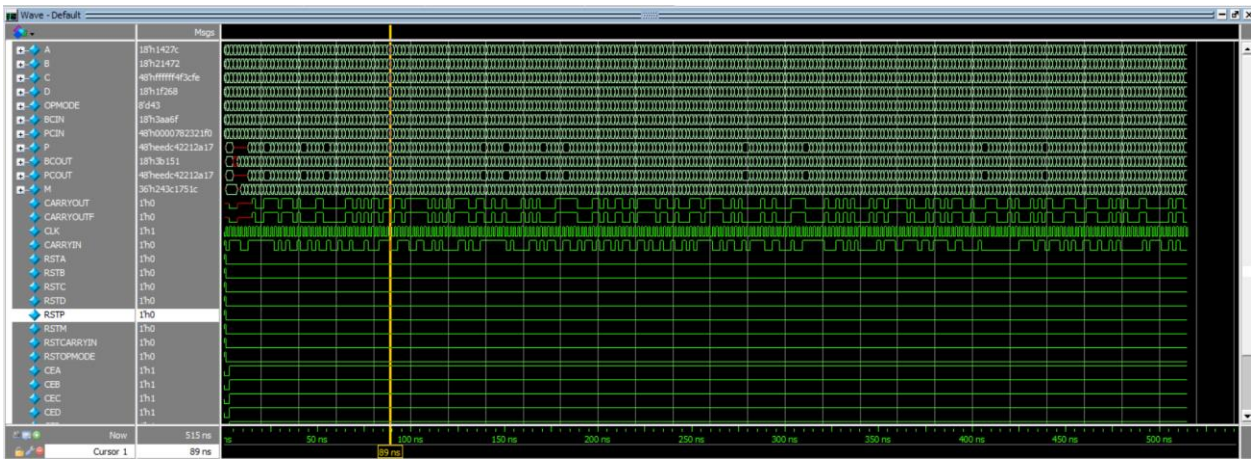
```
module DSP(input[17:0] A,input[17:0] B,input[47:0] C,input[17:0] D,
                input CLK,input CARRYIN,
                input[7:0] OPMODE,
                input[17:0] BCIN,
                input RSTA,input RSTB,input RSTM,input RSTP,input RSTC,input RSTD,input RSTCARRYIN,input RSTOPMODE,
  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

    wire[35:0] xmuxin;
    wire[17:0] BOUT;
    wire CIN;
    wire[7:0] opmodeout;
    wire[47:0] outC;
    wire[47:0] xmuxout;
    wire[47:0] zmuxout;
    wire[47:0] outP;
    wire[47:0] inP;
    wire CYOout;

    PortOPMODE OPMODEREG(.OPMODE(OPMODE),.CLK(CLK),.CEOPMODE(CEOPMODE),.RSTOPMODE(RSTOPMODE),.opmodeout(opmodeout));

    PortC CREG(.C(C),.CLK(CLK),.CEC(CEC),.RSTC(RSTC),.outC(outC));

    PortM MREG(.A(A),.B(B),.D(D),.BCIN(BCIN),.CLK(CLK),
               .OPMODE4(opmodeout[4]),.OPMODE6(opmodeout[6]),.RSTA(RSTA),
               .RSTB(RSTB),.RSTD(RSTD),.RSTM(RSTM),.CEA(CEA),.BCOUT(BOUT),
               .CEB(CEB),.CED(CED),.CEM(CEM),.xmux(xmuxin));

    assign M = xmuxin;
    assign BCOUT = BOUT;


    Carryin CYI(.CARRYIN(CARRYIN),.OPMODE5(opmodeout[5]),.CLK(CLK),.CECARRYIN(CECARRYIN),.RSTCARRYIN(RSTCARRYIN),.CIN(CIN));

    PortP PREG(.Postaddsub(inP),.CLK(CLK),.CEP(CEP),.RSTP(RSTP),.outP(outP));

    assign P = outP ;

    assign xmuxout = (opmodeout[1:0]==0)?0:(opmodeout[1:0]==1)?xmuxin:(opmodeout[1:0]==2)?outP:{D[11:0],A,B};

    assign zmuxout = (opmodeout[3:2]==0)?0:(opmodeout[3:2]==1)?PCIN:(opmodeout[3:2]==2)?outP:outC;

    assign inP = opmodeout[7]?zmuxout-(xmuxout+CIN):zmuxout+xmuxout+CIN;

    assign PCOUT = outP;

    dff_mux #(.size(1),.pipeline(1)) CYO(.in(outP),.clk(CLK),.EN(1'b1),.rst(1'b0),.out(CYOout));

    assign CARRYOUT = CYOout ;
    assign CARRYOUTF = CYOout ;

    endmodule
```

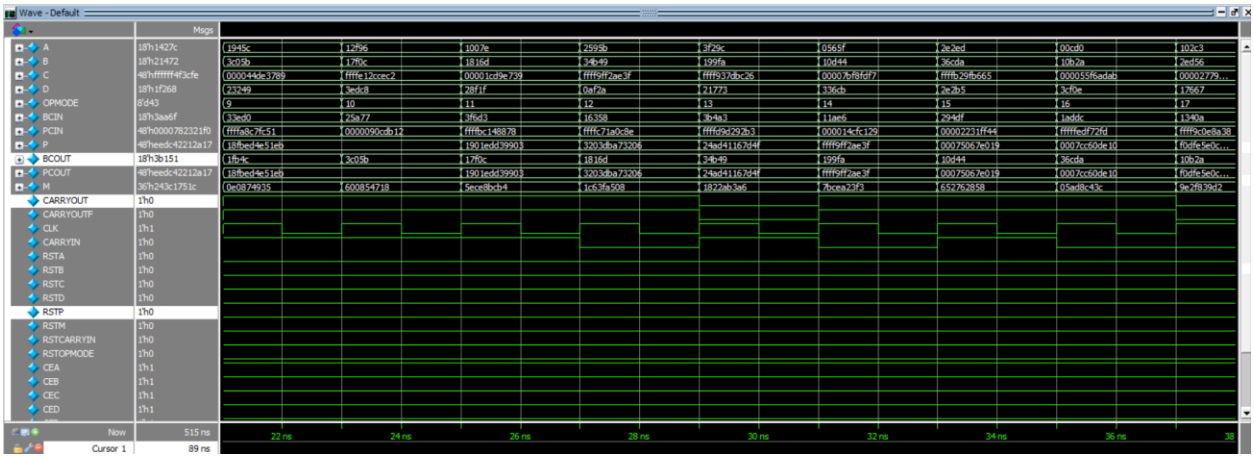
TB:

```
module DSP48A1_TB;

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

DSP DSP48A1_TB(.A(A),.B(B),.C(C),.D(D),.BCIN(BCIN),.CLK(CLK),
                .CARRYIN(CARRYIN),.OPMODE(OPMODE),.RSTA(RSTA),
                .RSTB(RSTB),.RSTC(RSTC),.RSTD(RSTD),
                .RSTP(RSTP),.RSTM(RSTM),.RSTCARRYIN(RSTCARRYIN),
                .RSTOPMODE(RSTOPMODE),.CEA(CEA),.CEB(CEB),
                .CEC(CEC),.CED(CED),.CEP(CEP),.CEM(CEM),
                .CECARRYIN(CECARRYIN),.CEOPMODE(CEOPMODE),
                .PCIN(PCIN),.BCOUT(BCOUT),.P(P),.PCOUT(PCOUT),.M(M),
                .CARRYOUT(CARRYOUT),.CARRYOUTF(CARRYOUTF));

initial
    begin
        CLK = 0 ;
        forever #1 CLK=~CLK;
    end
integer i ;
initial
    begin
        A = $random ;
        B = $random ;
        C = $random ;
        D = $random ;
        BCIN = $random ;
        CARRYIN = $random ;
        PCIN = $random ;
        OPMODE = $random ;
        RSTA = 1 ;
        RSTB = 1 ;
        RSTC = 1 ;
        RSTD = 1 ;
        RSTP = 1 ;
        RSTM = 1 ;
        RSTCARRYIN = 1;
        RSTOPMODE = 1;
        CEA = 0 ;
        CEB = 0 ;
        CEC = 0 ;
        CED = 0 ;
```

```
58          CECARRYIN = 0 ;
59          CEOPMODE = 0 ;
60          @(posedge CLK);
61          A = $random ;
62          B = $random ;
63          C = $random ;
64          D = $random ;
65          BCIN = $random ;
66          CARRYIN = $random ;
67          PCIN = $random ;
68          OPMODE = $random ;
69          RSTA = 0 ;
70          RSTB = 0 ;
71          RSTC = 0 ;
72          RSTD = 0 ;
73          RSTP = 0 ;
74          RSTM = 0 ;
75          RSTCARRYIN = 0;
76          RSTOPMODE = 0;
77          CEA = 0 ;
78          CEB = 0 ;
79          CEC = 0 ;
80          CED = 0 ;
81          CEP = 0 ;
82          CEM = 0 ;
83          CECARRYIN = 0 ;
84          CEOPMODE = 0 ;
85          @(posedge CLK);
86          for(i = 0 ; i<256 ; i=i+1)begin
87          A = $random ;
88          B = $random ;
89          C = $random ;
90          D = $random ;
91          BCIN = $random ;
92          CARRYIN = $random ;
93          PCIN = $random ;
94          OPMODE = i ;
95          RSTA = 0 ;
96          RSTB = 0 ;
97          RSTC = 0 ;
98          RSTD = 0 ;
99          RSTP = 0 ;
100         RSTM = 0 ;
101         RSTCARRYIN = 0;
102         RSTOPMODE = 0;
103         CEA = 1 ;
104         CEB = 1 ;
105         CEC = 1 ;
106         CED = 1 ;
107         CEP = 1 ;
108         CEM = 1 ;
109         CECARRYIN = 1 ;
110         CEOPMODE = 1 ;
111         @(posedge CLK);
112         end
113         $stop;
```

Waveform:

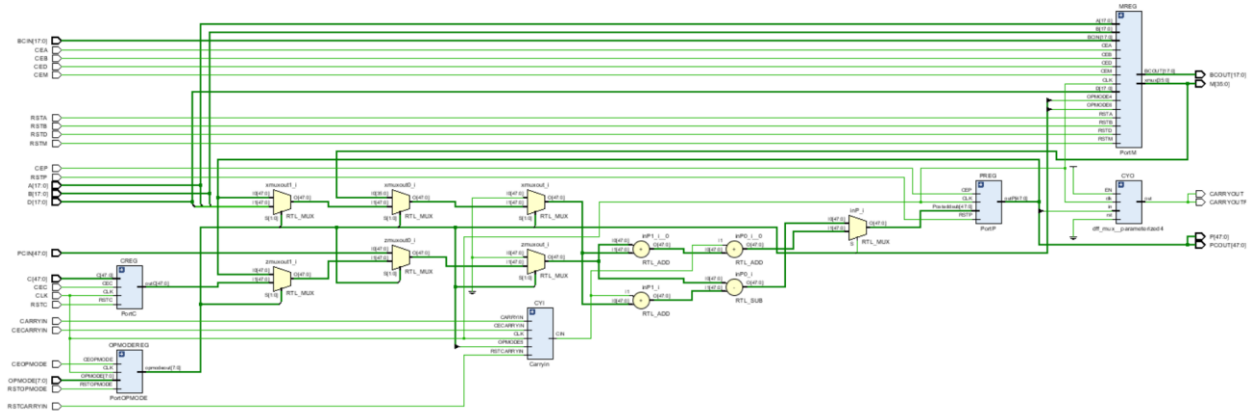


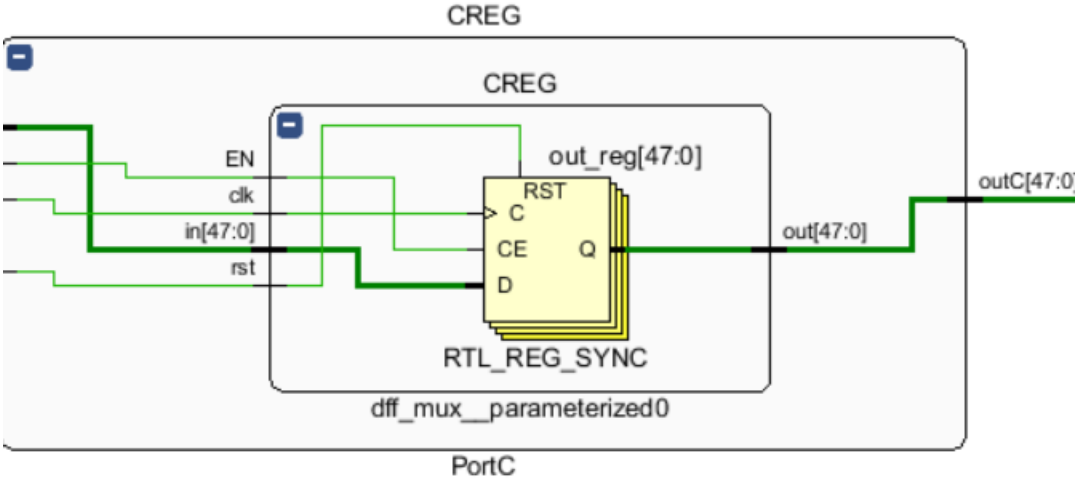


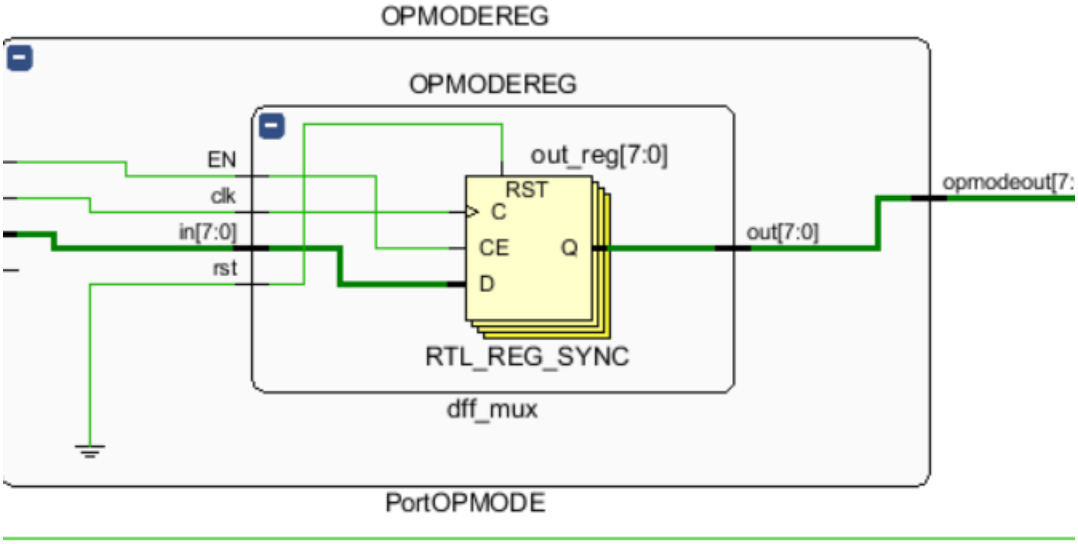
Constraints file:

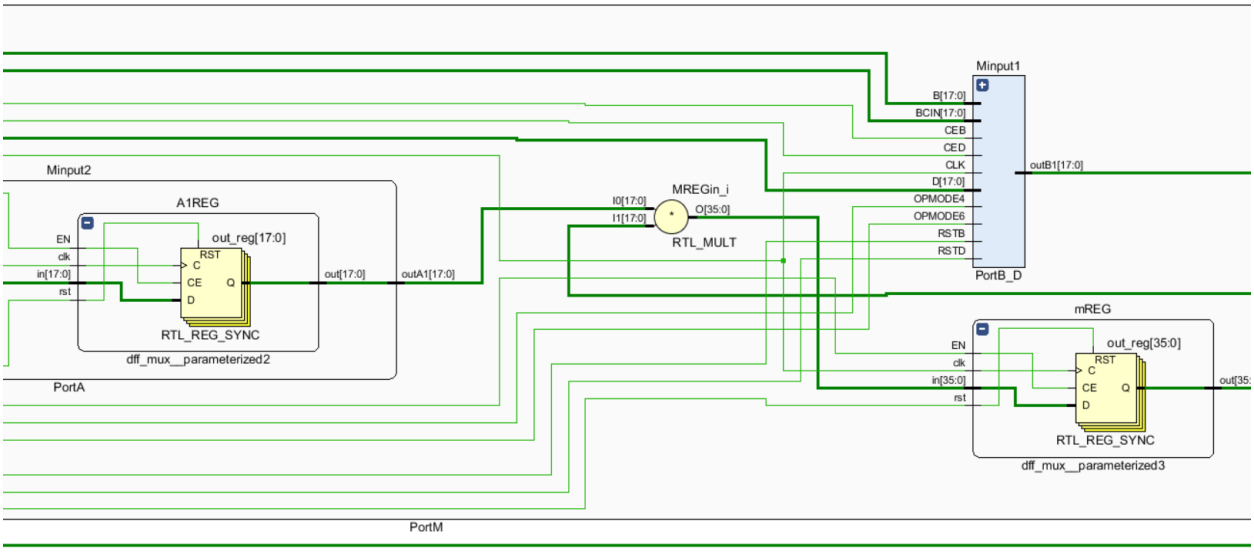
```
## Clock signal
set_property -dict {PACKAGE_PIN W5 IOSTANDARD LVCMOS33} [get_ports CLK]
create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [get_ports CLK]

## 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]
```

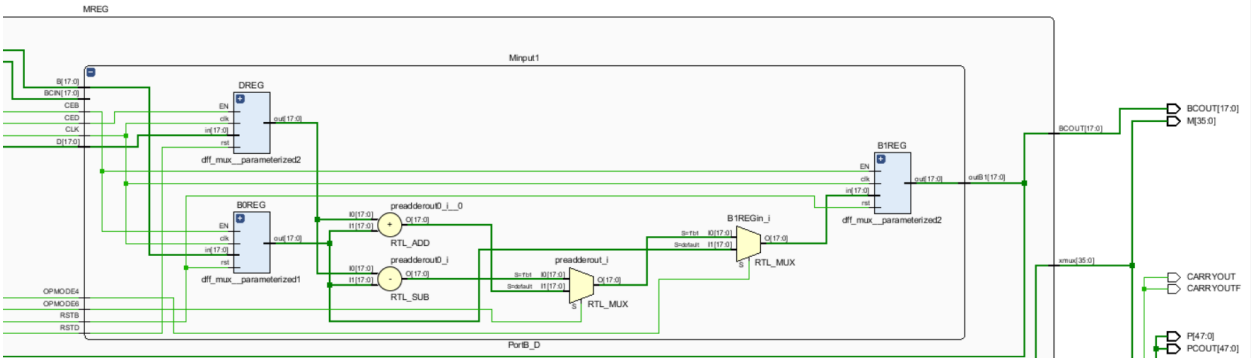
Elaboration:

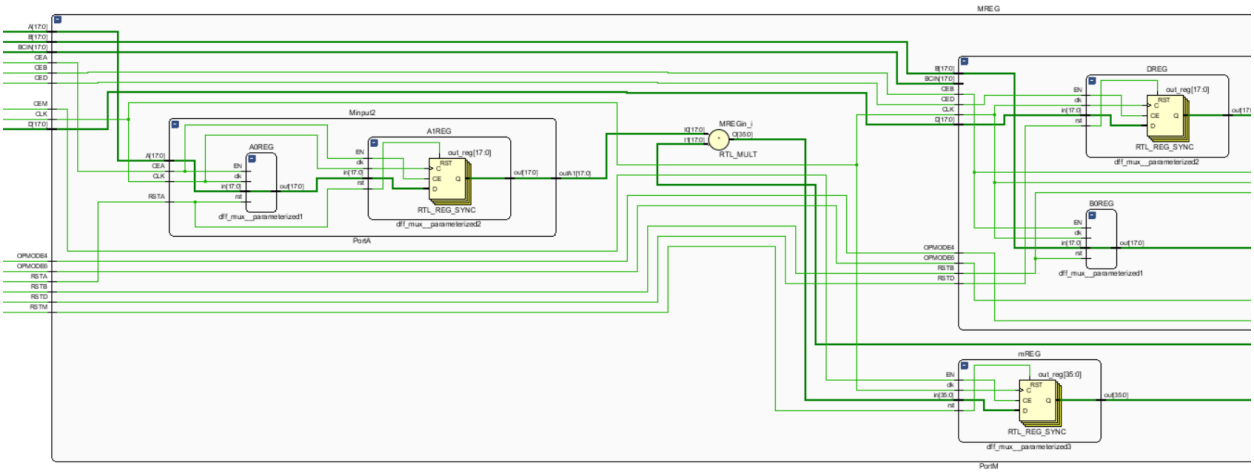




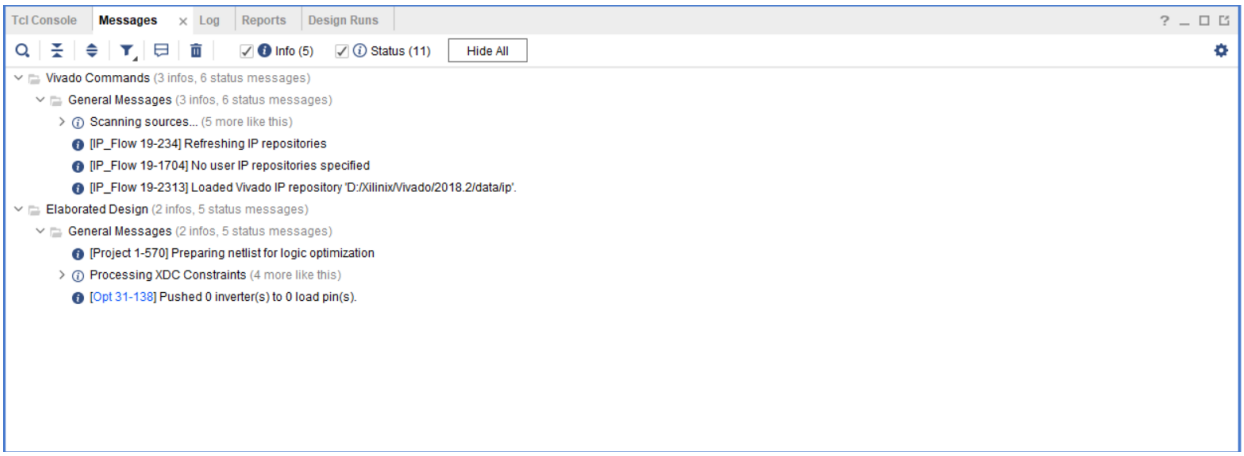



Minput1
B[17:0]
BCIN[17:0]
CEB
CED
CLK
D[17:0]
OPMODE4
OPMODE6
RSTB
RSTD
outB1[17:0]
PortB_D
Minput2
A1REG
EN
clk
in[17:0]
rst
out_reg[17:0]
RST
C
CE
D
Q
RTL_REG_SYNC
dff_mux__parameterized2
out[17:0]
outA1[17:0]
PortA
I0[17:0]
I1[17:0]
MREGin_i
O[35:0]
RTL_MULT
mREG
out_reg[35:0]
in[35:0]
dff_mux__parameterized3
PortM

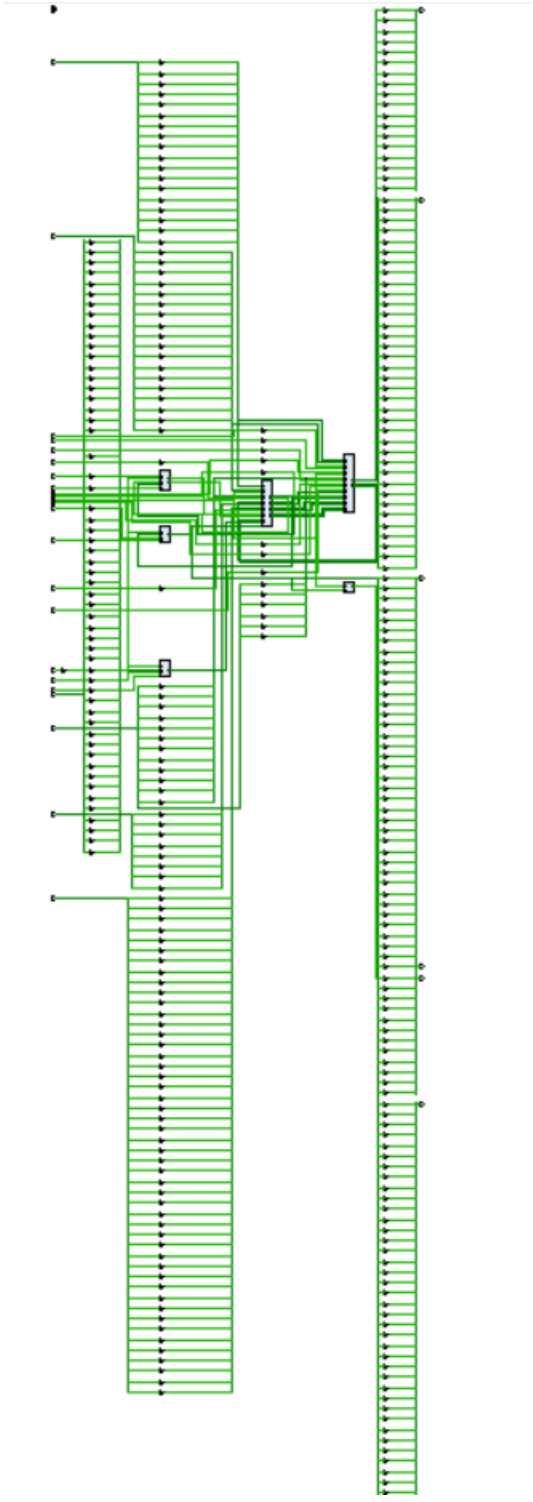

MREG
Minput1
DREG
B0REG
B1REG
preadderout0_i__0
preadderout0_i
preadderout_i
B1REGin_i
RTL_ADD
RTL_SUB
RTL_MUX
dff_mux__parameterized2
dff_mux__parameterized1
BCOUT[17:0]
M[35:0]
CARRYOUT
CARRYOUTF
P[47:0]
PCOUT[47:0]
PortB_D

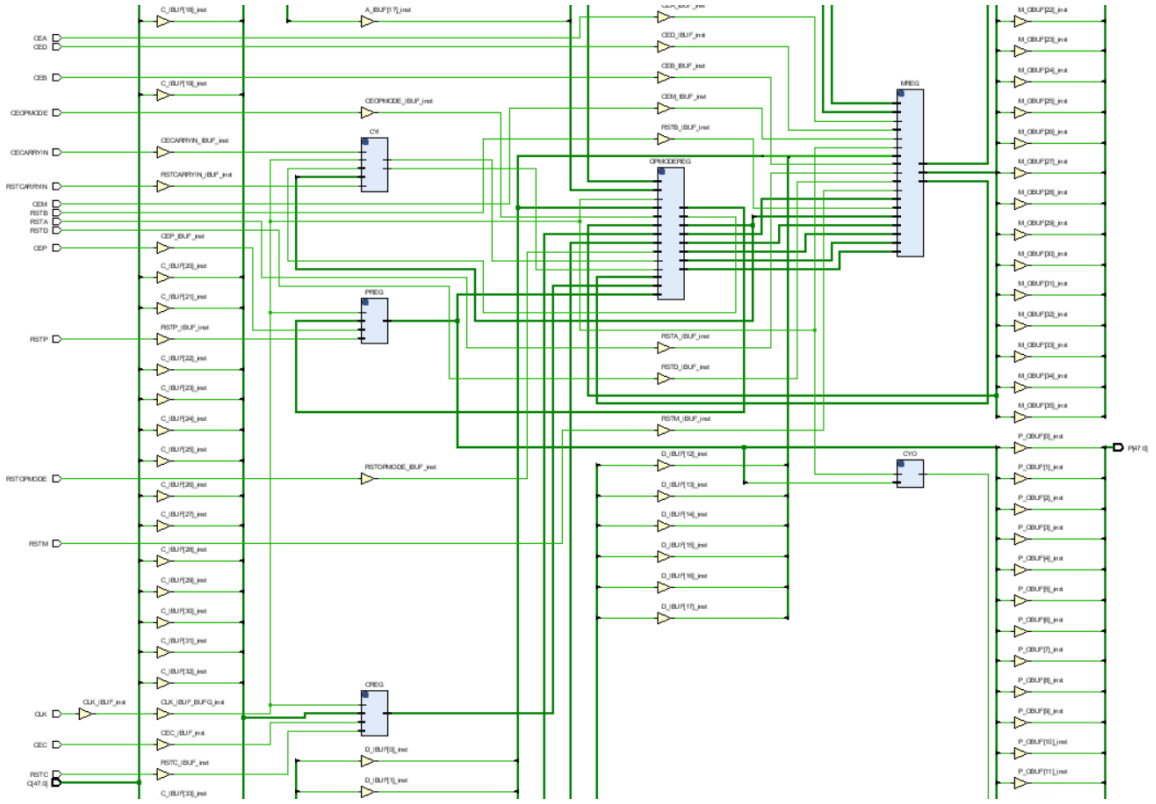

MREG
Minput2
A1REG
A0REG
MREGin_i
RTL_MULT
DREG
B0REG
mREG
RTL_REG_SYNC
PortA
PortM

Messages:



Synthesis:

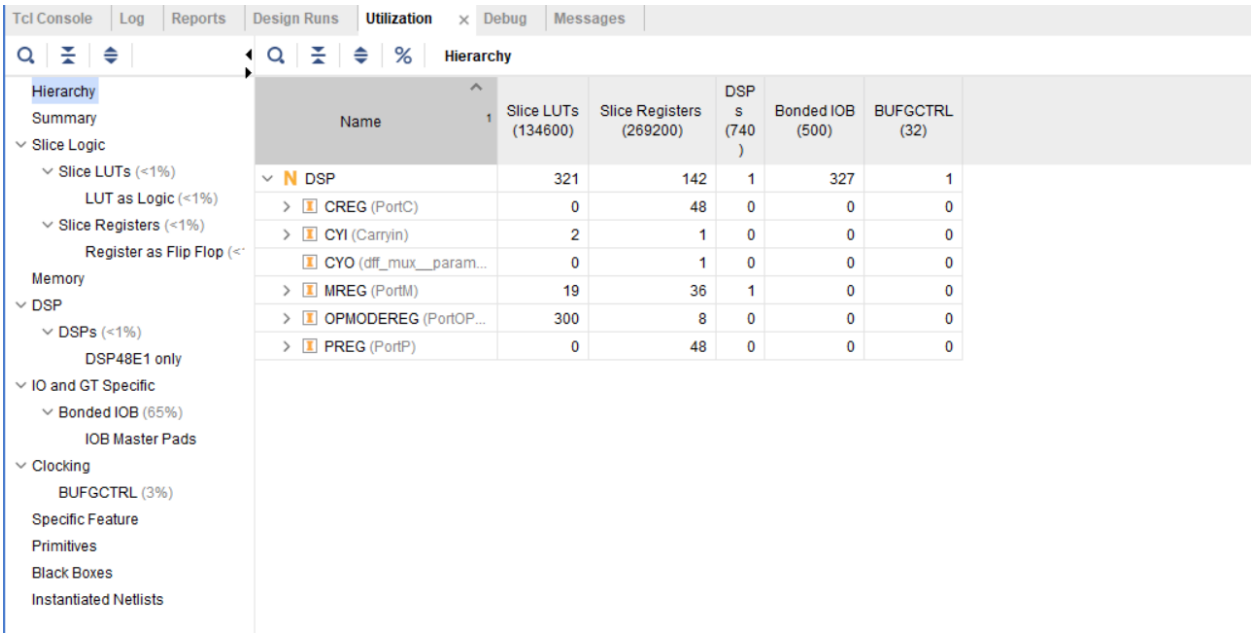




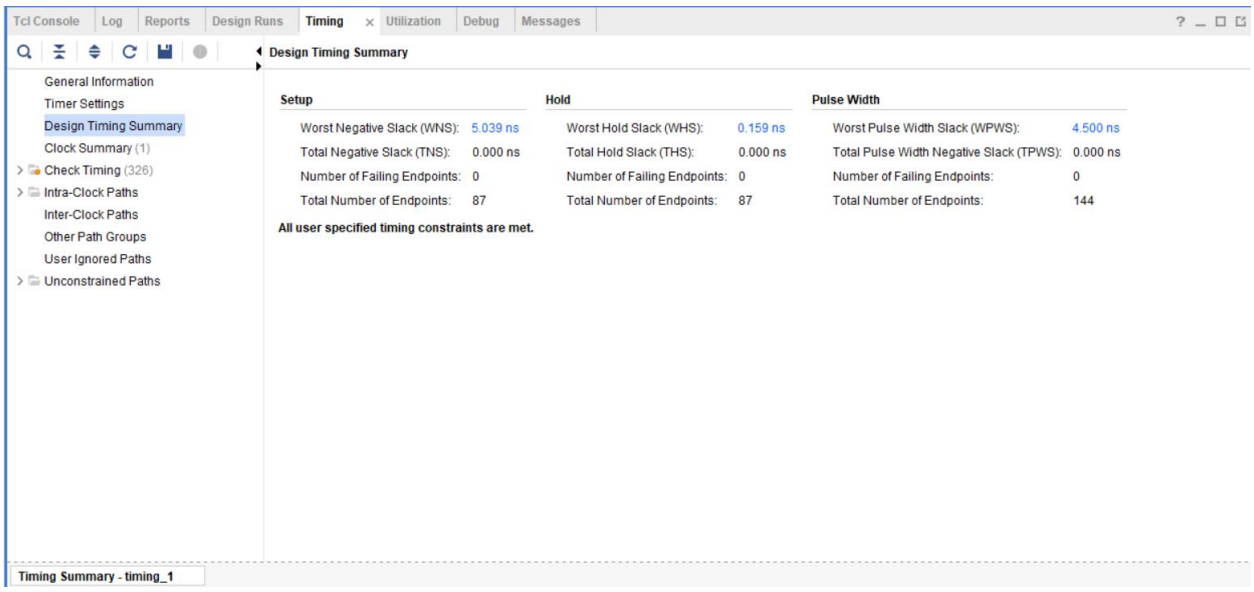
Messages:

- Synthesis (42 warnings, 46 infos, 11 status messages)
  - Command: synth_design -top DSP -part xc7a200tffg1156-3 (10 more like this)
  - [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a200t'
  - [Synth 8-6157] synthesizing module 'DSP' [DSP48A1.v:1] (13 more like this)
  - [Synth 8-6155] done synthesizing module 'dff_mux' (1#1) [dff_mux.v:1] (13 more like this)
  - [Synth 8-3331] design Carryin has unconnected port CARRYIN (40 more like this)
  - [Device 21-403] Loading part xc7a200tffg1156-3
  - [Project 1-236] Implementation specific constraints were found while reading constraint file [D:/Digital IC Design Diploma/Project/DSP/DSP.srcs/constrs_1/new/xc7a200tffg1156-3.xdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [.Xil/DSP_propImpl.xdc].
    Resolution: To avoid this warning, move constraints listed in ['Undefined'] to another XDC file and exclude this new file from synthesis with the used_in_synthesis property (File Properties dialog in GUI) and re-run elaboration/synthesis.
  - [Synth 8-5818] HDL ADVISOR - The operator resource <adder> is shared. To prevent sharing consider applying a KEEP on the output of the operator [PortB_D.v:29] (1 more like this)
  - [Synth 8-5842] Cannot pack DSP OPMODE registers because of constant '1' value. Packing the registers will cause simulation mismatch at initial cycle [PortM.v:34]
  - [Project 1-571] Translating synthesized netlist
  - [Netlist 29-17] Analyzing 205 Unisim elements for replacement
  - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
  - [Project 1-570] Preparing netlist for logic optimization (1 more like this)
  - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
  - [Project 1-111] Unisim Transformation Summary:
    No Unisim elements were transformed. (1 more like this)
  - [Common 17-83] Releasing license: Synthesis
  - [Constraints 18-5210] No constraint will be written out.
  - [Common 17-1381] The checkpoint 'D:/Digital IC Design Diploma/Project/DSP/DSP.runs/synth_1/DSP.dcp' has been generated.
  - [runtcl-4] Executing : report_utilization -file DSP_utilization_synth.rpt -pb DSP_utilization_synth.pb
  - [Common 17-206] Exiting Vivado at Sun Jul 28 14:47:40 2024...
- Synthesized Design (6 infos, 2 status messages)
  - General Messages (6 infos, 2 status messages)
    - [Netlist 29-17] Analyzing 205 Unisim elements for replacement

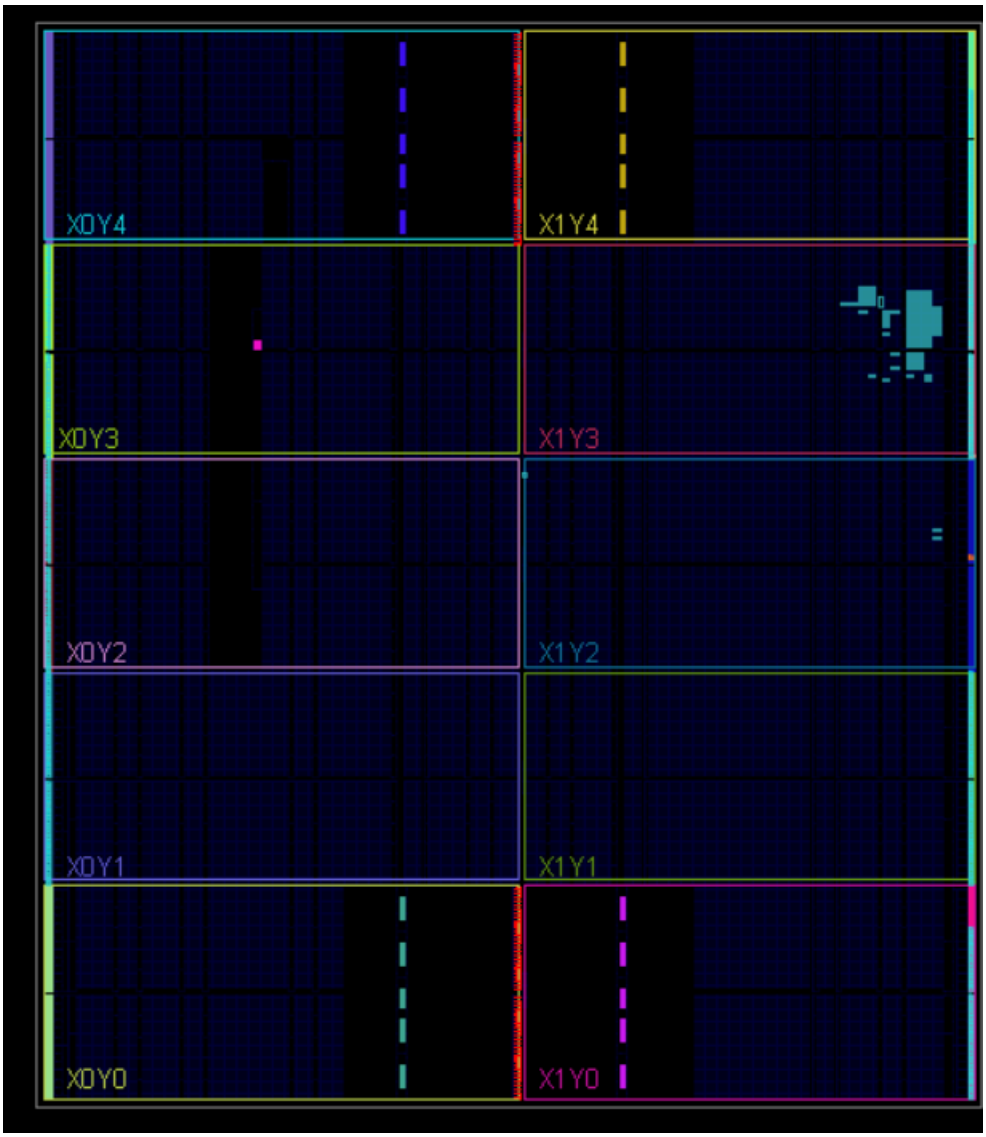
Utilization:



Timing:

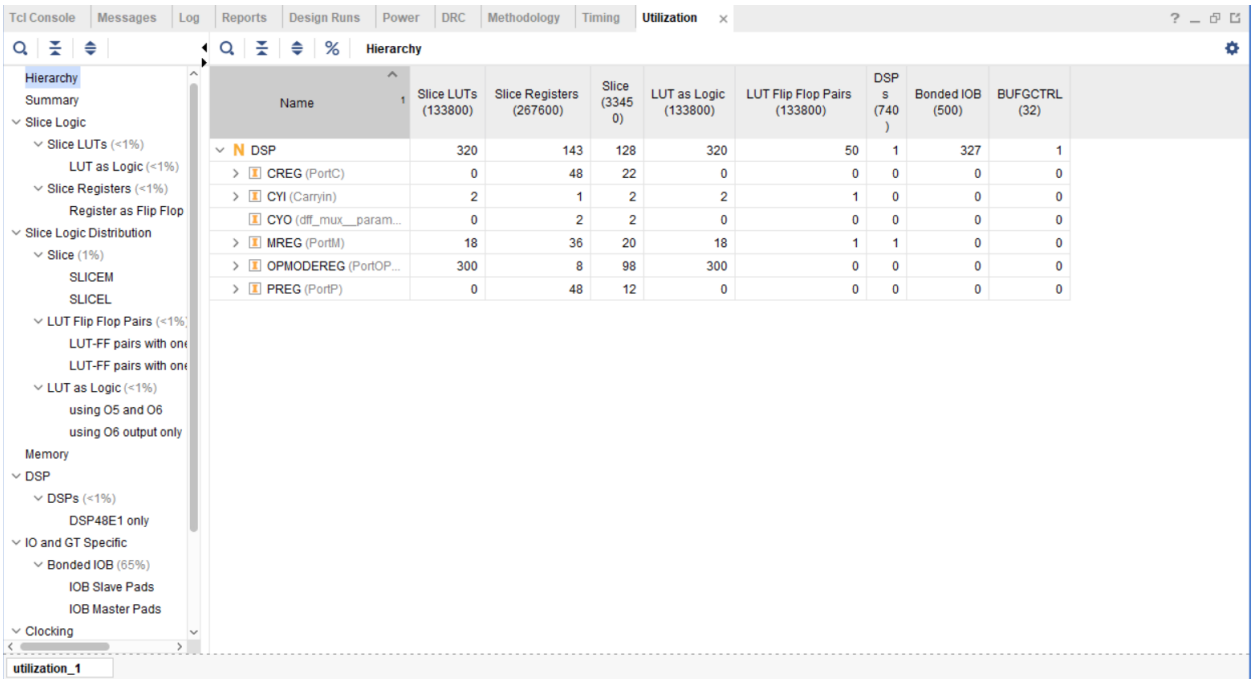


Implementation:



- Implementation (1 warning, 89 infos, 219 status messages)
- Implemented Design (9 infos, 4 status messages)
  - General Messages (9 infos, 4 status messages)

Utilization:



Timing: