

Project 1

DSP

Maryam Ahmed Kamal
Maryam Hossam Abd El_Ghaffar
Yasmin Hany Momtaz Abo El-Magd

1. RTL DESIGN

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

input [47:0] C, PCIN;
input [17:0] A, B, D;
input [7:0] OPMODE;
input CARRYIN, CLK, CEA, CEB, CEC, CED, CECARRYIN, CEOPMODE, CEM, CEP;
input RSTA, RSTB, RSTC, RSTD, RSTCARRYIN, RSTOPMODE, RSTM, RSTP;
output [47:0] P, PCOUT;
output [35:0] M;
output [17:0] BCOUT;
output CARRYOUT, CARRYOUTF;
parameter A0REG = 0;      //no register, A0 is the first stage of pipeline
parameter A1REG = 1;      //register, A1 is the second stage of pipeline
parameter B0REG = 0;      //no register, B0 is the first stage of pipeline
parameter B1REG = 1;      //register, B1 is the first stage of pipeline
parameter CREG = 1,
           DREG = 1,
           MREG = 1,
           PREG = 1,
           CARRYINREG = 1,
           CARRYOUTREG = 1,
           OPMODEREG = 1;
parameter CARRYINSEL = "OPMODE5";
parameter B_INPUT = "DIRECT";    //input to port B is routed from the B input
parameter RSTTYPE = "SYNC";

wire [17:0] A0_out, A1_out, B0_out, B1_out, D_out;
wire [35:0] M_out;
wire [47:0] C_out, P_out;
wire [7:0] OPMODE_OUT;
wire CIN, COUT, CYO_in;
reg [17:0] B0_in, B1_in, Pre_out;
reg [47:0] Post_out, X_out, Z_out;
reg [35:0] Mul_out;
reg C_in;

//Modules Instantiation
reg_mux #(.RESET(RSTTYPE), .SEL(OPMODEREG), .W(8)) op (CLK, CEOPMODE, RSTOPMODE, OPMODE, OPMODE_OUT);

reg_mux #(.RESET(RSTTYPE), .SEL(DREG), .W(18)) D_reg (CLK, CED, RSTD, D, D_out);
reg_mux #(.RESET(RSTTYPE), .SEL(B0REG), .W(18)) B0_reg (CLK, CEB, RSTB, B0_in, B0_out);
```

```

reg_mux #(.RESET(RSTTYPE), .SEL(A0REG), .W(18)) A0_reg (CLK, CEA, RSTA, A, A0_out);
reg_mux #(.RESET(RSTTYPE), .SEL(CREG), .W(48)) C_reg (CLK, CEC, RSTC, C, C_out);
reg_mux #(.RESET(RSTTYPE), .SEL(A1REG), .W(18)) A1_reg (CLK, CEA, RSTA, A0_out, A1_out);
reg_mux #(.RESET(RSTTYPE), .SEL(B1REG), .W(18)) B1_reg (CLK, CEB, RSTB, B1_in, B1_out);

reg_mux #(.RESET(RSTTYPE), .SEL(MREG), .W(36)) M_reg (CLK, CEM, RSTM, Mul_out, M_out);

reg_mux #(.RESET(RSTTYPE), .SEL(CARRYINREG), .W(1)) CYI_reg (CLK, CECARRYIN, RSTCARRYIN, C_in, CIN);
reg_mux #(.RESET(RSTTYPE), .SEL(CARRYOUTREG), .W(1)) CYO_reg (CLK, CECARRYIN, RSTCARRYIN, CYO_in,
COUT);

reg_mux #(.RESET(RSTTYPE), .SEL(PREG), .W(48)) P_reg (CLK, CEP, RSTP, Post_out, P_out);

// B0 input selection
always @ (*) begin
    case (B_INPUT)
        "DIRECT" : B0_in = B;
        "CASCADE" : B0_in = B1_out;
        default : B0_in = 0;
    endcase
end

// pre adder
always @ (*) begin
    case (OPMODE_OUT[6])
        0 : Pre_out = D_out + B0_out;
        1 : Pre_out = D_out - B0_out;
    endcase
end

// B1 input selection
always @ (*) begin
    case (OPMODE_OUT[4])
        0 : B1_in = B0_out;
        1 : B1_in = Pre_out;
    endcase
end

//Multiplier
always @ (*) begin
    Mul_out = B1_out * A1_out;
end

// carryin input selection
always @ (*) begin
    case (CARRYINSEL)
        "OPMODE5" : C_in = OPMODE_OUT[5];
        "CARRYIN" : C_in = CARRYIN;
        default : C_in = 0;
    endcase
end

```

```

// X mux
always @ (*) begin
  case (OPMODE_OUT[1:0])
    0 : X_out = 0;
    1 : X_out = M_out;
    2 : X_out = P_out;
    3 : X_out = {D[11:0], A[17:0], B[17:0]};
  endcase
end

// Z mux
always @ (*) begin
  case (OPMODE_OUT[3:2])
    0 : Z_out = 0;
    1 : Z_out = PCIN;
    2 : Z_out = P_out;
    3 : Z_out = C_out;
  endcase
end

// post adder
always @ (*) begin
  case (OPMODE_OUT[7])
    0 : Post_out = Z_out + X_out;
    1 : Post_out = Z_out - (X_out + CIN);
  endcase
end

// carryout assignment
assign CYO_in = Post_out[47];

// output assignments
assign BCOUT = B1_out;
assign M = M_out;
assign P = P_out;
assign PCOUT = P_out;
assign CARRYOUT = COUT;
assign CARRYOUTF = COUT;

endmodule

```

```

module reg_mux (clk, cen, rst, data_in, out);

parameter SEL = 1;
parameter W = 18;
parameter RESET = "ASYNC";
input clk, cen, rst;
input [W-1:0] data_in;
output reg [W-1:0] out;

generate
    if(SEL && RESET == "SYNC") begin
        always @(posedge clk) begin
            if(rst) begin
                out <= 0;
            end
            else if(cen) begin
                out <= data_in;
            end
        end
    end
endgenerate

generate
    if(SEL && RESET == "ASYNC") begin
        always @(posedge clk or posedge rst) begin
            if (rst) begin
                out <=0 ;
            end
            else if (cen) begin
                out <= data_in;
            end
        end
    end
endgenerate

generate
    if(SEL == 0) begin
        always @(*) begin
            out = data_in;
        end
    end
endgenerate

endmodule

```

2. Testbench code

```
module DSP_tb1();

reg [17:0] a, b, d;
reg [47:0] c, pcin;
reg carryin, clk, rsta, rstb, rstm, rstp, rstc, rstd, rstcarryin, rstopmode;
reg cea, ceb, cem, cep, cec, ced, cecarryin, ceopmode;
reg [7:0] opmode;
wire [35:0] m;
wire [17:0] bcout;
wire [47:0] p, pcout;
wire carryout, carryoutf;

DSP dsp_design (a, b, c, d, carryin, m, p, carryout, carryoutf, clk, opmode, cea,
ceb, cec, ced, cecarryin, ceopmode, cem, cep, rsta,rstb, rstc, rstd, rstcarryin,
rstopmode, rstm, rstp, pcin, bcout, pcout);

integer i;
initial begin
  clk=0;
  forever #1 clk=~clk;
end

initial begin
rsta=1;
rstb=1;
rstm=1;
rstp=1;
rstc=1;
rstd=1;
rstcarryin=1;
rstopmode=1;
$display("cea=%b, ceb=%b, cec=%b, ced=%b, cecarryin=%b, ceopmode=%b, cem=%b, cep=%b,
rsta=%b,rstb=%b, rstc=%b, rstd=%b, rstcarryin=%b, rstopmode=%b, rstm=%b, rstp=%b
",cea, ceb, cec, ced, cecarryin, ceopmode, cem, cep, rsta,rstb, rstc, rstd,
rstcarryin, rstopmode,rstm, rstp);

for(i=0 ; i<30 ; i=i+1) begin
  a=$random;
  b=$random;
  d=$random;
  c=$random;
  opmode=$random;
  carryin=$random;
  pcin=$random;
  cea=$random;
```

```

ceb=$random;
cem=$random;
cep=$random;
cec=$random;
ced=$random;
cecarryin=$random;
ceopmode=$random;
@(negedge clk);
end
rsta=0;
rstb=0;
rstm=0;
rstp=0;
rstc=0;
rstd=0;
rstcarryin=0;
rstopmode=0;
cea=1;
ceb=1;
cem=1;
cep=1;
cec=1;
ced=1;
cecarryin=1;
ceopmode=1;
$display("cea=%b, ceb=%b, cec=%b, ced=%b, cecarryin=%b, ceopmode=%b, cem=%b, cep=%b,
rsta=%b,rstb=%b, rstc=%b, rstd=%b, rstcarryin=%b, rstopmode=%b, rstm=%b, rstp=%b
",cea, ceb, cec, ced, cecarryin, ceopmode, cem, cep, rsta,rstb, rstc, rstd,
rstcarryin, rstopmode,rstm, rstp);
opmode[1:0]=2'b11;

for(i=0;i<50;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[3:2]=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

```

```
opmode[1:0]=2'b10;

for(i=0;i<50;i=i+1) begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[3:2]=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[1:0]=2'b01;
opmode[4]=0;

for(i=0;i<50;i=i+1) begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[3:2]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[1:0]=2'b01;
opmode[4]=1'b1;

for(i=0;i<50;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[3:2]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
```

```
    @(negedge clk);
end

opmode[1:0]=2'b00;

for(i=0;i<50;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[3:2]=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[3:2]=2'b11;

for(i=0;i<20;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[3:2]=2'b10;

for(i=0;i<20;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
```

```

pcin=$random;
@(negedge clk);
end

opmode[3:2]=2'b01;

for(i=0;i<40;i=i+1) begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[3:2]=2'b00;

for(i=0;i<10;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[3:2]=2'b11;

for(i=0;i<20;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[1:0]=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;

```

```
carryin=$random;
pcin=$random;
@(negedge clk);
end

opmode[3:2]=2'b10;

for(i=0;i<20;i=i+1) begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[1:0]=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[3:2]=2'b01;

for(i=0;i<40;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[1:0]=$random;
    opmode[4]=$random;
    opmode[5]=$random;
    opmode[6]=$random;
    opmode[7]=$random;
    carryin=$random;
    pcin=$random;
    @(negedge clk);
end

opmode[3:2]=2'b00;

for(i=0;i<10;i=i+1)begin
    a=$random;
    b=$random;
    d=$random;
    c=$random;
    opmode[1:0]=$random;
    opmode[4]=$random;
```

```

opmode[5]=$random;
opmode[6]=$random;
opmode[7]=$random;
carryin=$random;
pcin=$random;
@(negedge clk);
end
$stop;
end

initial begin
$monitor("a=%b, b=%b, c=%b, d=%b, carryin=%b, m=%b, p=%b, carryout=%b,
carryoutf=%b, clk=%b, opmode=%b, pcin=%b, bcout=%b, pcout=%b",a, b, c, d, carryin,
m, p, carryout, carryoutf,clk, opmode, pcin,bcout,pcout);
end

endmodule

```

3. Do File

```

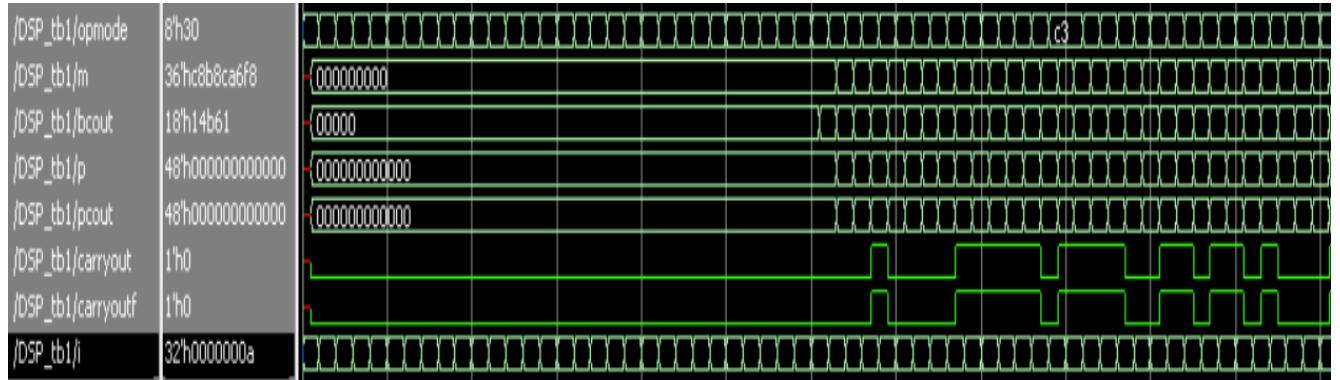
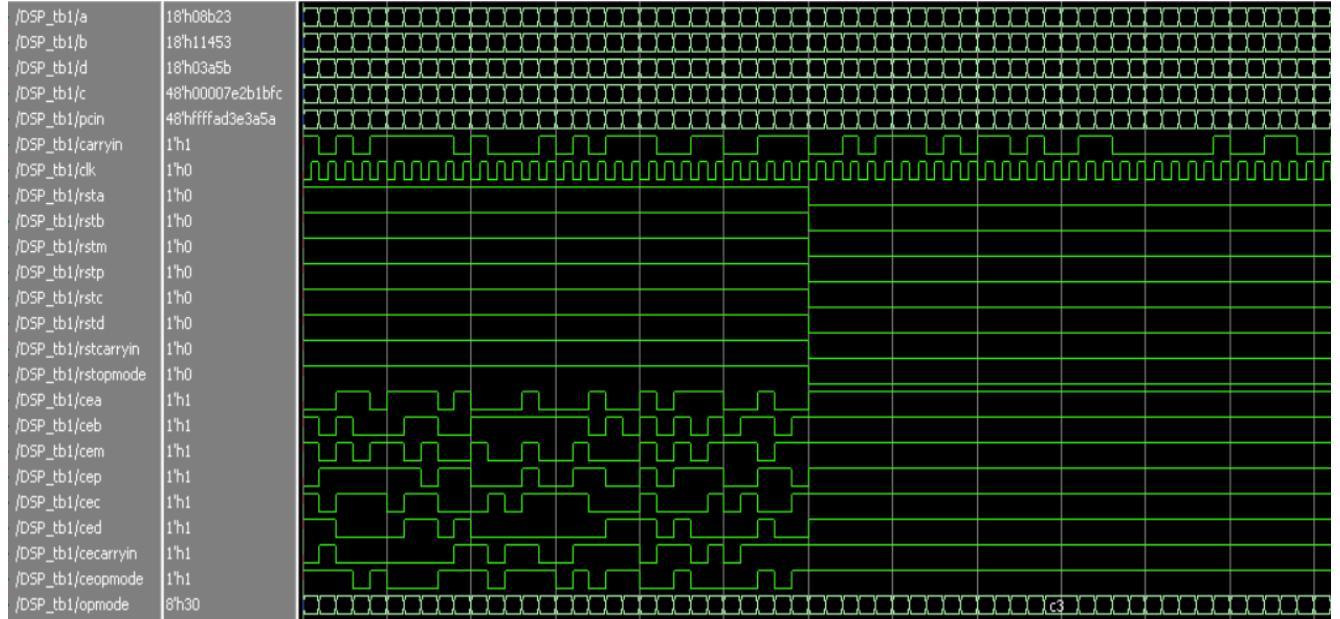
vlib work
vlog DSP.v DSP_tb.v reg_mux.v
vsim -voptargs=+acc work.DSP_tb1
add wave *
run -all
#quit -sim

```

4. QuestaSim Snippets

→ First 30 cycles: reset is activated and clock enable is randomized, so all outputs are zero.

→ Then, reset is deactivated and clock enable is forced to be enabled.

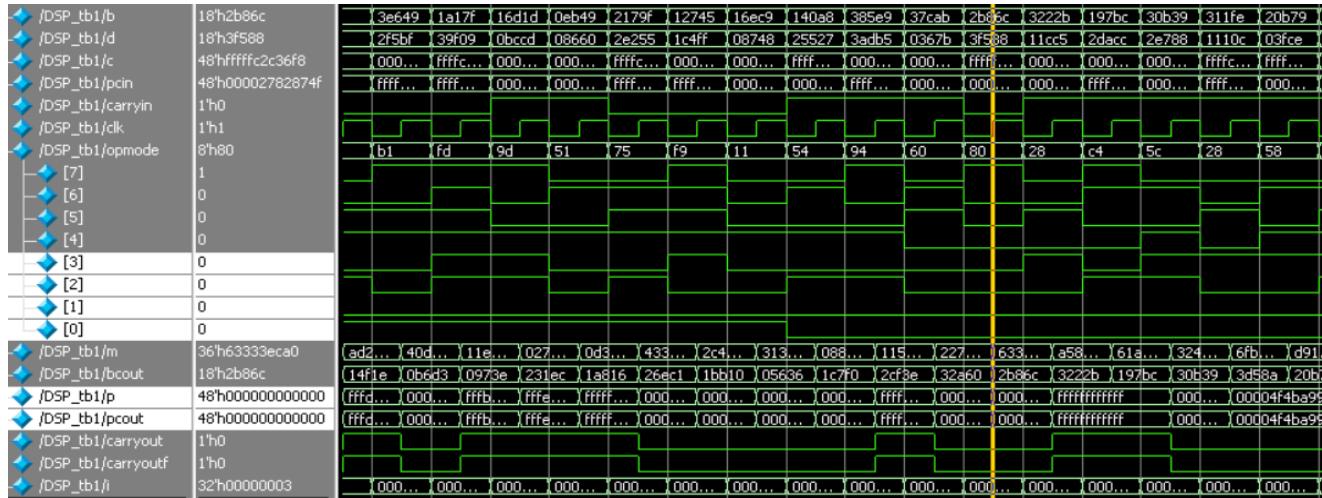


- Some Random checks

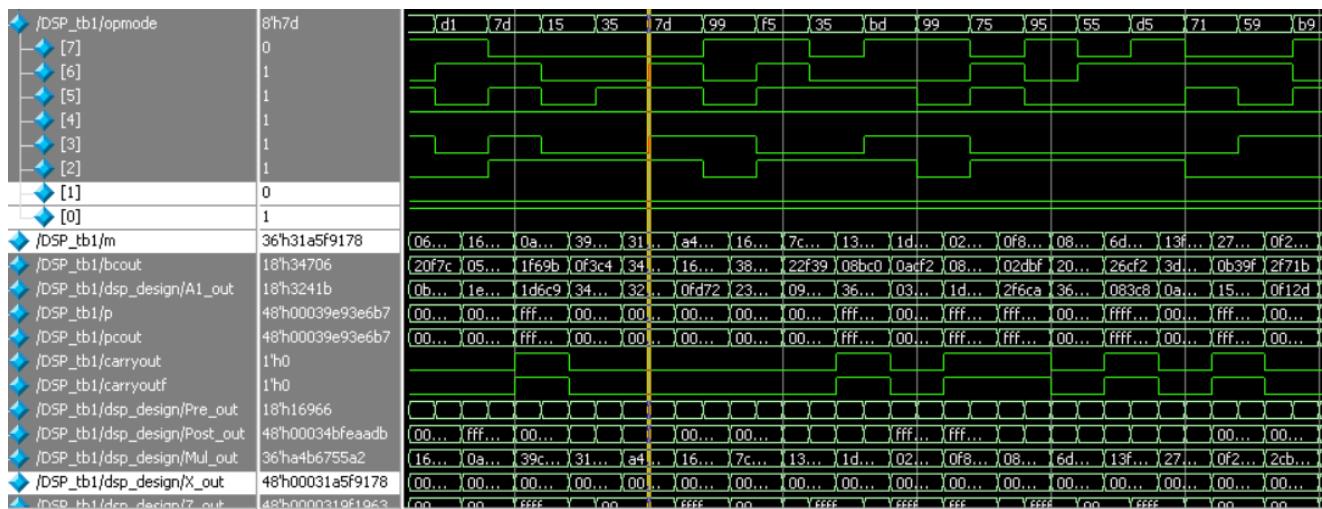
→ Clock is enabled and reset is deactivated

1. Test X and Z multiplexers behavior

→ When opmode [1:0] = 'b00 & opmode [3:2] = 'b00, post adder out = 0 & hence P = 0.

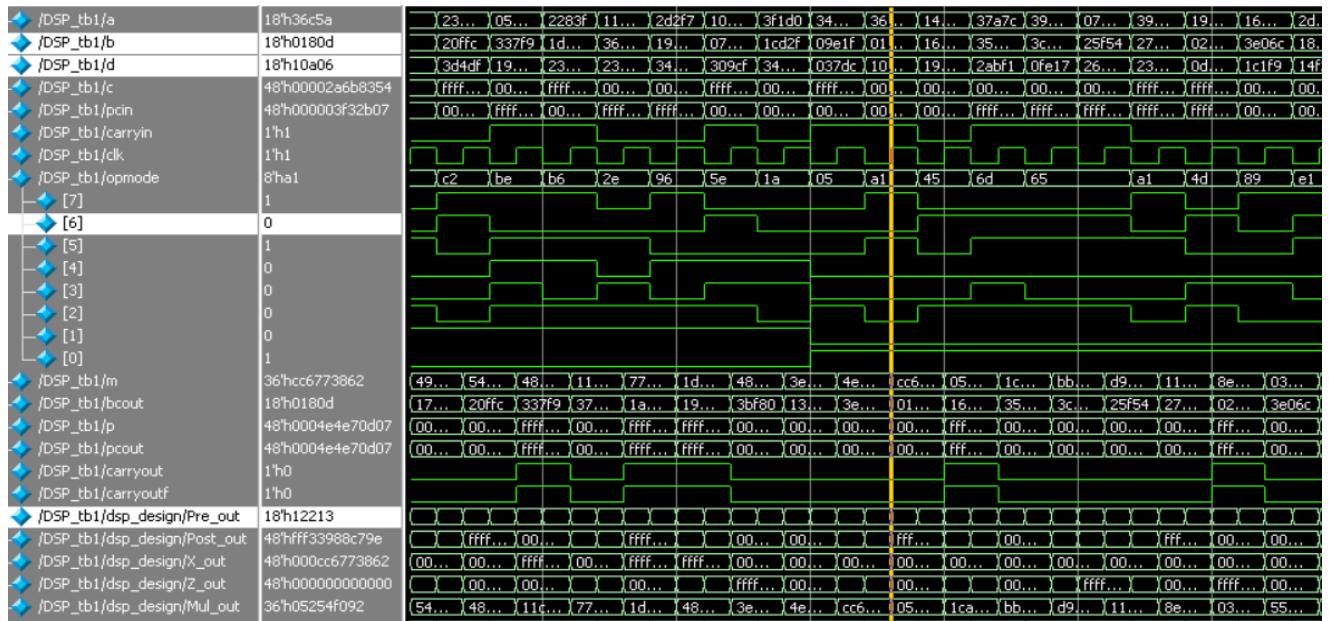


→ When opmode [1:0] = 'b01, X_out = M.



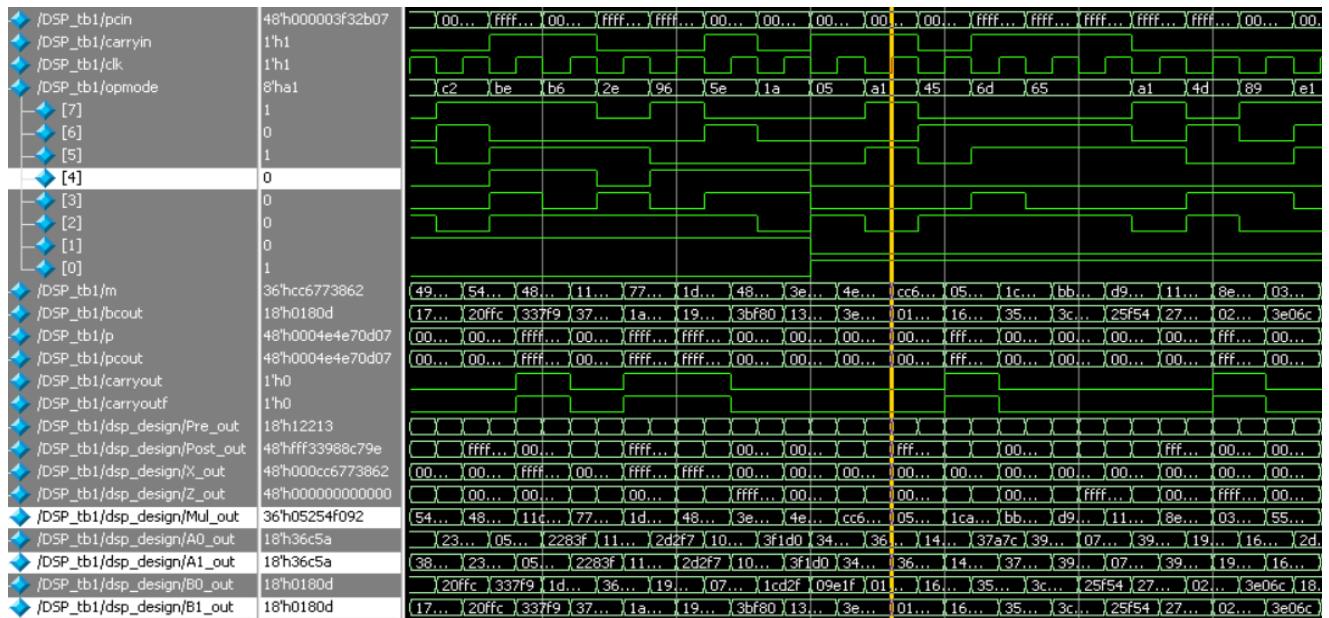
2. Test pre adder behavior

→ When opmode [6] = 0, pre adder adds B & D then gives the result to pre_out.



3. Test multiplier behavior

→ When opmode [4] = 0, multiplier multiplies A1_out times B1_out and gives the result to Mul_out.

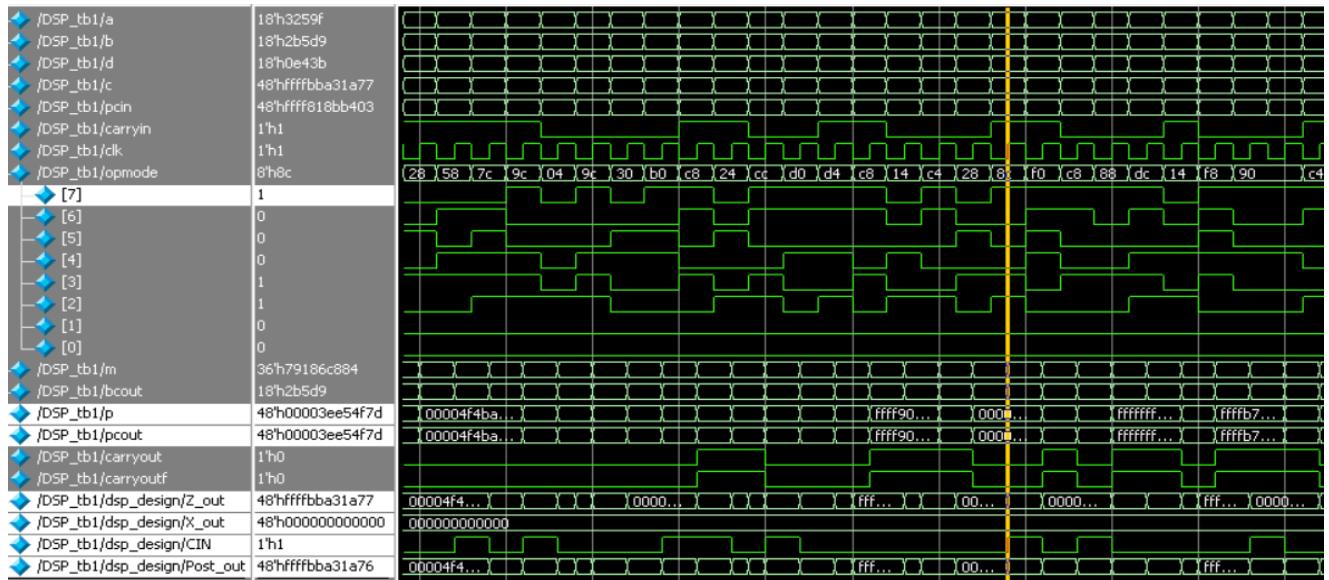


4. Test post adder behavior

→ When opmode [7] = 1, operation of post adder is:

`Post_out = Z_out - (X_out + CIN);`

Which is achieved here, and also P = post_out.



5. Constraint File

```

1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6 ## Clock signal
7 set_property -dict { PACKAGE_PIN W5 IOSTANDARD LVCMS33 } [get_ports CLK]
8 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports CLK]
9
10
11 ## Switches
12 #set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMS33 } [get_ports {sw[0]}]
13 #set_property -dict { PACKAGE_PIN V16 IOSTANDARD LVCMS33 } [get_ports {sw[1]}]
14 #set_property -dict { PACKAGE_PIN W16 IOSTANDARD LVCMS33 } [get_ports {sw[2]}]
15 #set_property -dict { PACKAGE_PIN W17 IOSTANDARD LVCMS33 } [get_ports {sw[3]}]
16 #set_property -dict { PACKAGE_PIN W15 IOSTANDARD LVCMS33 } [get_ports {sw[4]}]
17 #set_property -dict { PACKAGE_PIN V15 IOSTANDARD LVCMS33 } [get_ports {sw[5]}]
18 #set_property -dict { PACKAGE_PIN W14 IOSTANDARD LVCMS33 } [get_ports {sw[6]}]
19 #set_property -dict { PACKAGE_PIN W13 IOSTANDARD LVCMS33 } [get_ports {sw[7]}]
20 #set_property -dict { PACKAGE_PIN V2 IOSTANDARD LVCMS33 } [get_ports {sw[8]}]
21 #set_property -dict { PACKAGE_PIN T3 IOSTANDARD LVCMS33 } [get_ports {sw[9]}]
22 #set_property -dict { PACKAGE_PIN T2 IOSTANDARD LVCMS33 } [get_ports {sw[10]}]
23 #set_property -dict { PACKAGE_PIN R3 IOSTANDARD LVCMS33 } [get_ports {sw[11]}]
24 #set_property -dict { PACKAGE_PIN W2 IOSTANDARD LVCMS33 } [get_ports {sw[12]}]
25 #set_property -dict { PACKAGE_PIN U1 IOSTANDARD LVCMS33 } [get_ports {sw[13]}]
26 #set_property -dict { PACKAGE_PIN T1 IOSTANDARD LVCMS33 } [get_ports {sw[14]}]
27 #set_property -dict { PACKAGE_PIN R2 IOSTANDARD LVCMS33 } [get_ports {sw[15]}]
28
29

```

```

30 ## LEDs
31 #set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMOS33 } [get_ports {led[0]}]
32 #set_property -dict { PACKAGE_PIN E19 IOSTANDARD LVCMOS33 } [get_ports {led[1]}]
33 #set_property -dict { PACKAGE_PIN U19 IOSTANDARD LVCMOS33 } [get_ports {led[2]}]
34 #set_property -dict { PACKAGE_PIN V19 IOSTANDARD LVCMOS33 } [get_ports {led[3]}]
35 #set_property -dict { PACKAGE_PIN W18 IOSTANDARD LVCMOS33 } [get_ports {led[4]}]
36 #set_property -dict { PACKAGE_PIN U15 IOSTANDARD LVCMOS33 } [get_ports {led[5]}]
37 #set_property -dict { PACKAGE_PIN U14 IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
38 #set_property -dict { PACKAGE_PIN V14 IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
39 #set_property -dict { PACKAGE_PIN V13 IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
40 #set_property -dict { PACKAGE_PIN V3 IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
41 #set_property -dict { PACKAGE_PIN W3 IOSTANDARD LVCMOS33 } [get_ports {led[10]}]
42 #set_property -dict { PACKAGE_PIN U3 IOSTANDARD LVCMOS33 } [get_ports {led[11]}]
43 #set_property -dict { PACKAGE_PIN P3 IOSTANDARD LVCMOS33 } [get_ports {led[12]}]
44 #set_property -dict { PACKAGE_PIN N3 IOSTANDARD LVCMOS33 } [get_ports {led[13]}]
45 #set_property -dict { PACKAGE_PIN P1 IOSTANDARD LVCMOS33 } [get_ports {led[14]}]
46 #set_property -dict { PACKAGE_PIN L1 IOSTANDARD LVCMOS33 } [get_ports {led[15]}]
47
48
49 ## Segment Display
50 #set_property -dict { PACKAGE_PIN W7 IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
51 #set_property -dict { PACKAGE_PIN W6 IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
52 #set_property -dict { PACKAGE_PIN U8 IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
53 #set_property -dict { PACKAGE_PIN V8 IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
54 #set_property -dict { PACKAGE_PIN U5 IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
55 #set_property -dict { PACKAGE_PIN V5 IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
56 #set_property -dict { PACKAGE_PIN U7 IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]
57
58 #set_property -dict { PACKAGE_PIN V7 IOSTANDARD LVCMOS33 } [get_ports dp]
59
60 #set_property -dict { PACKAGE_PIN U2 IOSTANDARD LVCMOS33 } [get_ports {an[0]}]
61 #set_property -dict { PACKAGE_PIN U4 IOSTANDARD LVCMOS33 } [get_ports {an[1]}]
62 #set_property -dict { PACKAGE_PIN V4 IOSTANDARD LVCMOS33 } [get_ports {an[2]}]
63 #set_property -dict { PACKAGE_PIN W4 IOSTANDARD LVCMOS33 } [get_ports {an[3]}]
64
65
66 ##Buttons
67 #set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMOS33 } [get_ports btnC]
68 #set_property -dict { PACKAGE_PIN T18 IOSTANDARD LVCMOS33 } [get_ports btnU]
69 #set_property -dict { PACKAGE_PIN W19 IOSTANDARD LVCMOS33 } [get_ports btnL]
70 #set_property -dict { PACKAGE_PIN T17 IOSTANDARD LVCMOS33 } [get_ports btnR]
71 #set_property -dict { PACKAGE_PIN U17 IOSTANDARD LVCMOS33 } [get_ports btnD]
72
73
74 ##Pmod Header JA
75 #set_property -dict { PACKAGE_PIN J1 IOSTANDARD LVCMOS33 } [get_ports {JA[0]}];#Sch name = JA1
76 #set_property -dict { PACKAGE_PIN L2 IOSTANDARD LVCMOS33 } [get_ports {JA[1]}];#Sch name = JA2
77 #set_property -dict { PACKAGE_PIN J2 IOSTANDARD LVCMOS33 } [get_ports {JA[2]}];#Sch name = JA3
78 #set_property -dict { PACKAGE_PIN G2 IOSTANDARD LVCMOS33 } [get_ports {JA[3]}];#Sch name = JA4
79 #set_property -dict { PACKAGE_PIN H1 IOSTANDARD LVCMOS33 } [get_ports {JA[4]}];#Sch name = JA7
80 #set_property -dict { PACKAGE_PIN K2 IOSTANDARD LVCMOS33 } [get_ports {JA[5]}];#Sch name = JA8
81 #set_property -dict { PACKAGE_PIN H2 IOSTANDARD LVCMOS33 } [get_ports {JA[6]}];#Sch name = JA9
82 #set_property -dict { PACKAGE_PIN G3 IOSTANDARD LVCMOS33 } [get_ports {JA[7]}];#Sch name = JA10
83
84 ##Pmod Header JB
85 #set_property -dict { PACKAGE_PIN A14 IOSTANDARD LVCMOS33 } [get_ports {JB[0]}];#Sch name = JB1
86 #set_property -dict { PACKAGE_PIN A16 IOSTANDARD LVCMOS33 } [get_ports {JB[1]}];#Sch name = JB2
87 #set_property -dict { PACKAGE_PIN B15 IOSTANDARD LVCMOS33 } [get_ports {JB[2]}];#Sch name = JB3
88 #set_property -dict { PACKAGE_PIN B16 IOSTANDARD LVCMOS33 } [get_ports {JB[3]}];#Sch name = JB4
89 #set_property -dict { PACKAGE_PIN A15 IOSTANDARD LVCMOS33 } [get_ports {JB[4]}];#Sch name = JB7
90 #set_property -dict { PACKAGE_PIN A17 IOSTANDARD LVCMOS33 } [get_ports {JB[5]}];#Sch name = JB8
91 #set_property -dict { PACKAGE_PIN C15 IOSTANDARD LVCMOS33 } [get_ports {JB[6]}];#Sch name = JB9
92 #set_property -dict { PACKAGE_PIN C16 IOSTANDARD LVCMOS33 } [get_ports {JB[7]}];#Sch name = JB10
93
94 ##Pmod Header JC
95 #set_property -dict { PACKAGE_PIN K17 IOSTANDARD LVCMOS33 } [get_ports {JC[0]}];#Sch name = JC1
96 #set_property -dict { PACKAGE_PIN M18 IOSTANDARD LVCMOS33 } [get_ports {JC[1]}];#Sch name = JC2
97 #set_property -dict { PACKAGE_PIN N17 IOSTANDARD LVCMOS33 } [get_ports {JC[2]}];#Sch name = JC3
98 #set_property -dict { PACKAGE_PIN P18 IOSTANDARD LVCMOS33 } [get_ports {JC[3]}];#Sch name = JC4
99 #set_property -dict { PACKAGE_PIN L17 IOSTANDARD LVCMOS33 } [get_ports {JC[4]}];#Sch name = JC7
100 #set_property -dict { PACKAGE_PIN M19 IOSTANDARD LVCMOS33 } [get_ports {JC[5]}];#Sch name = JC8
101 #set_property -dict { PACKAGE_PIN P17 IOSTANDARD LVCMOS33 } [get_ports {JC[6]}];#Sch name = JC9
102 #set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports {JC[7]}];#Sch name = JC10
103
104 ##Pmod Header JXADC
105 #set_property -dict { PACKAGE_PIN J3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[0]}];#Sch name = XA1_P
106 #set_property -dict { PACKAGE_PIN L3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[1]}];#Sch name = XA2_P
107 #set_property -dict { PACKAGE_PIN M2 IOSTANDARD LVCMOS33 } [get_ports {JXADC[2]}];#Sch name = XA3_P
108 #set_property -dict { PACKAGE_PIN N2 IOSTANDARD LVCMOS33 } [get_ports {JXADC[3]}];#Sch name = XA4_P
109 #set_property -dict { PACKAGE_PIN K3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[4]}];#Sch name = XA1_N
110 #set_property -dict { PACKAGE_PIN M3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[5]}];#Sch name = XA2_N
111 #set_property -dict { PACKAGE_PIN M1 IOSTANDARD LVCMOS33 } [get_ports {JXADC[6]}];#Sch name = XA3_N
112 #set_property -dict { PACKAGE_PIN N1 IOSTANDARD LVCMOS33 } [get_ports {JXADC[7]}];#Sch name = XA4_N
113
114
115 ##VGA Connector
116 #set_property -dict { PACKAGE_PIN G19 IOSTANDARD LVCMOS33 } [get_ports {vgaRed[0]}]
117 #set_property -dict { PACKAGE_PIN H19 IOSTANDARD LVCMOS33 } [get_ports {vgaRed[1]}]
118 #set_property -dict { PACKAGE_PIN J19 IOSTANDARD LVCMOS33 } [get_ports {vgaRed[2]}]
119 #set_property -dict { PACKAGE_PIN N19 IOSTANDARD LVCMOS33 } [get_ports {vgaRed[3]}]
120 #set_property -dict { PACKAGE_PIN N18 IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[0]}]
121 #set_property -dict { PACKAGE_PIN L18 IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[1]}]
122 #set_property -dict { PACKAGE_PIN K18 IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[2]}]
123 #set_property -dict { PACKAGE_PIN J18 IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[3]}]
124 #set_property -dict { PACKAGE_PIN J17 IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[0]}]
125 #set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[1]}]
126 #set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[2]}]
127 #set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[3]}]
128 #set_property -dict { PACKAGE_PIN P19 IOSTANDARD LVCMOS33 } [get_ports Hsync]
129 #set_property -dict { PACKAGE_PIN R19 IOSTANDARD LVCMOS33 } [get_ports Vsync]
130
131
132 ##USB-RS232 Interface
133 #set_property -dict { PACKAGE_PIN B18 IOSTANDARD LVCMOS33 } [get_ports RsRx]
134 #set_property -dict { PACKAGE_PIN A18 IOSTANDARD LVCMOS33 } [get_ports RsTx]
135
136
137 ##USB HID (PS/2)
138 #set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 PULLUP true } [get_ports PS2Clk]
139 #set_property -dict { PACKAGE_PIN B17 IOSTANDARD LVCMOS33 PULLUP true } [get_ports PS2Data]
140
141

```

```

142 ##Quad SPI Flash
143 ##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the
144 ##STARTUP_E2 primitive.
145 #set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMS33 } [get_ports {QspiDB[0]}]
146 #set_property -dict { PACKAGE_PIN D19 IOSTANDARD LVCMS33 } [get_ports {QspiDB[1]}]
147 #set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMS33 } [get_ports {QspiDB[2]}]
148 #set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMS33 } [get_ports {QspiDB[3]}]
149 #set_property -dict { PACKAGE_PIN K19 IOSTANDARD LVCMS33 } [get_ports QspiCSn]
150
151
152 ## Configuration options, can be used for all designs
153 set_property CONFIG_VOLTAGE 3.3 [current_design]
154 set_property CFGVBUS VCCO [current_design]
155
156 ## SPI configuration mode options for QSPI boot, can be used for all designs
157 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
158 set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
159 set_property CONFIG_MODE SPIx4 [current_design]

```

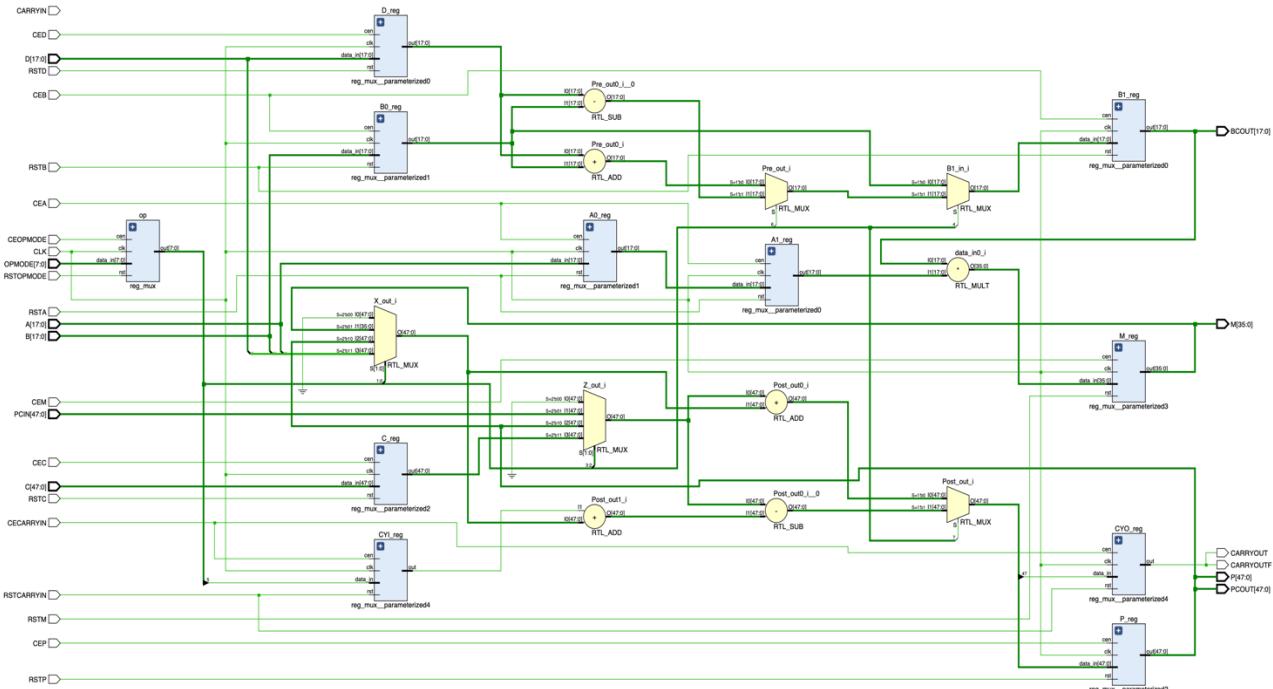
- Using Vivado

6. Elaboration

1. “Messages” Tab

- ✓ Elaborated Design (4 warnings, 18 infos)
 - ✓ General Messages (4 warnings, 18 infos)
 - > ⓘ [Synth 8-6157] synthesizing module 'DSP' [DSP.v:1] (6 more like this)
 - > ⓘ [Synth 8-6155] done synthesizing module 'reg_mux' (1#1) [reg_mux.v:1] (6 more like this)
 - > ⓘ [Synth 8-3331] design reg_mux__parameterized1 has unconnected port clk (3 more like this)
 - ⓘ [Device 21-403] Loading part xc7a200tffg1156-3
 - ⓘ [Project 1-570] Preparing netlist for logic optimization
 - ⓘ [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
 - ⓘ [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

2. Schematic Snippets



7. Synthesis

1. “Messages” Tab

The screenshot shows the Vivado IDE's 'Messages' tab for a project named 'xc7a200tfg1156-3'. The log is filled with synthesis messages, primarily warnings and infos. Key entries include:

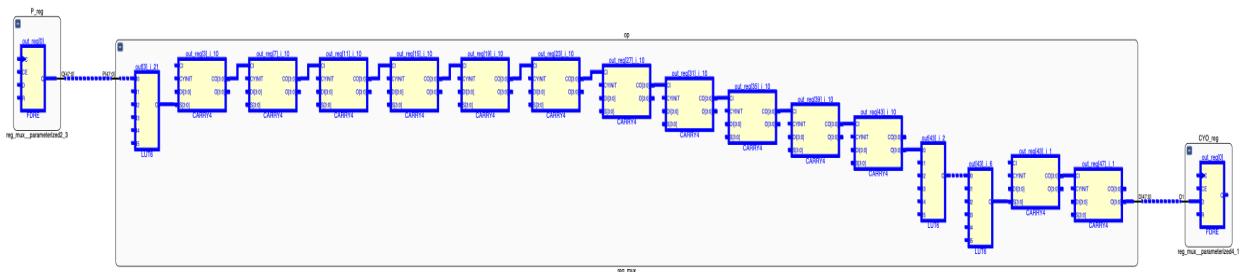
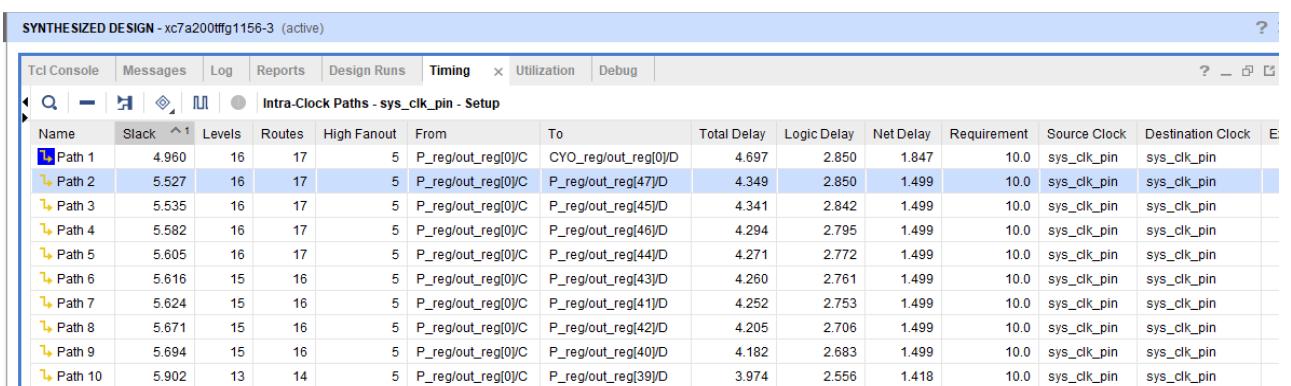
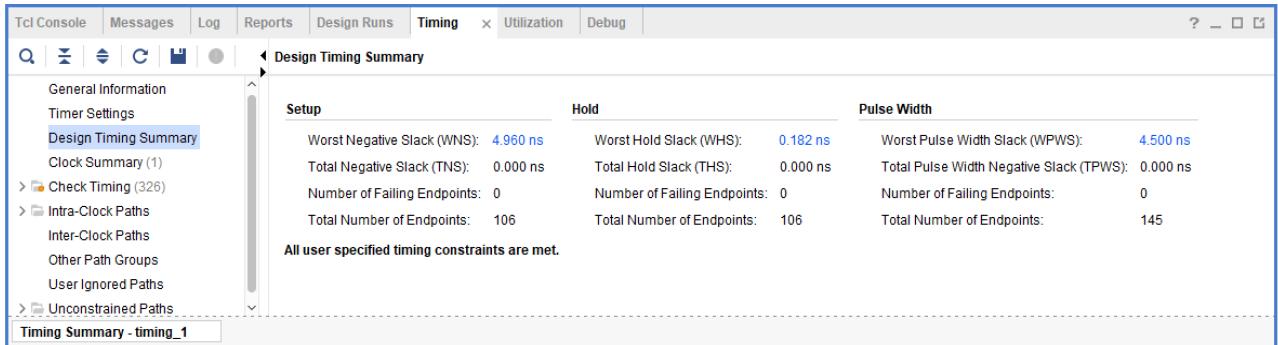
- Vivado Commands: IP_Flow 19-234] Refreshing IP repositories, IP_Flow 19-1704] No user IP repositories specified, IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2018.2/data/ip'.
- Elaborated Design: General Messages, Synthesis, and Synthesized Design sections containing numerous warnings and infos related to module synthesis, logic optimization, and resource usage.
- Synthesis: General Messages, Synthesis, and Synthesized Design sections containing warnings and infos related to license acquisition, resource sharing, and synthesis completion.

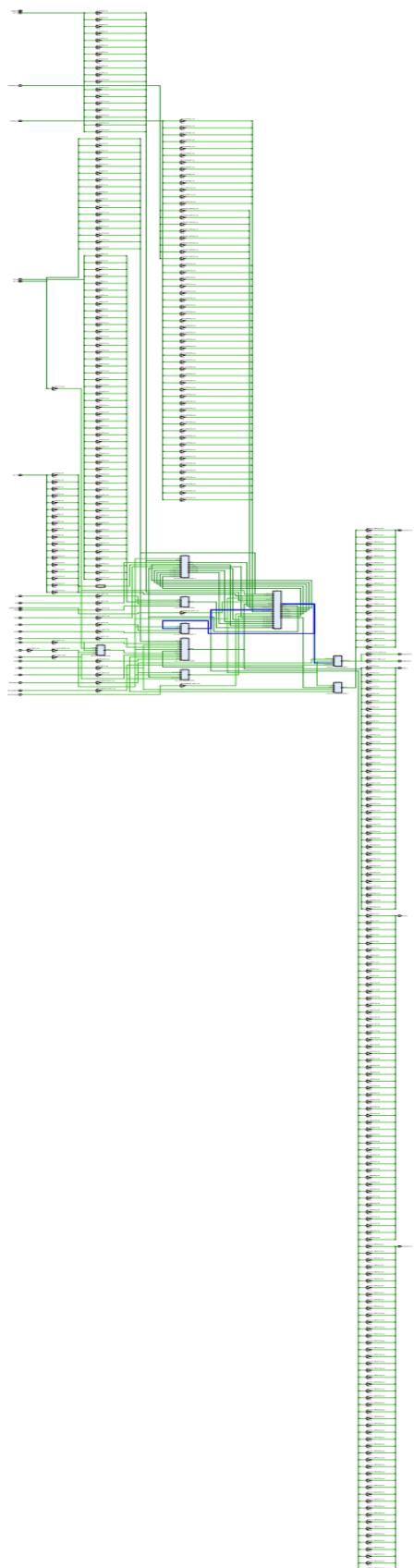
2. Utilization Report

The screenshot shows the Vivado IDE's 'Utilization' tab for the same project. The utilization report table provides a detailed breakdown of resource usage:

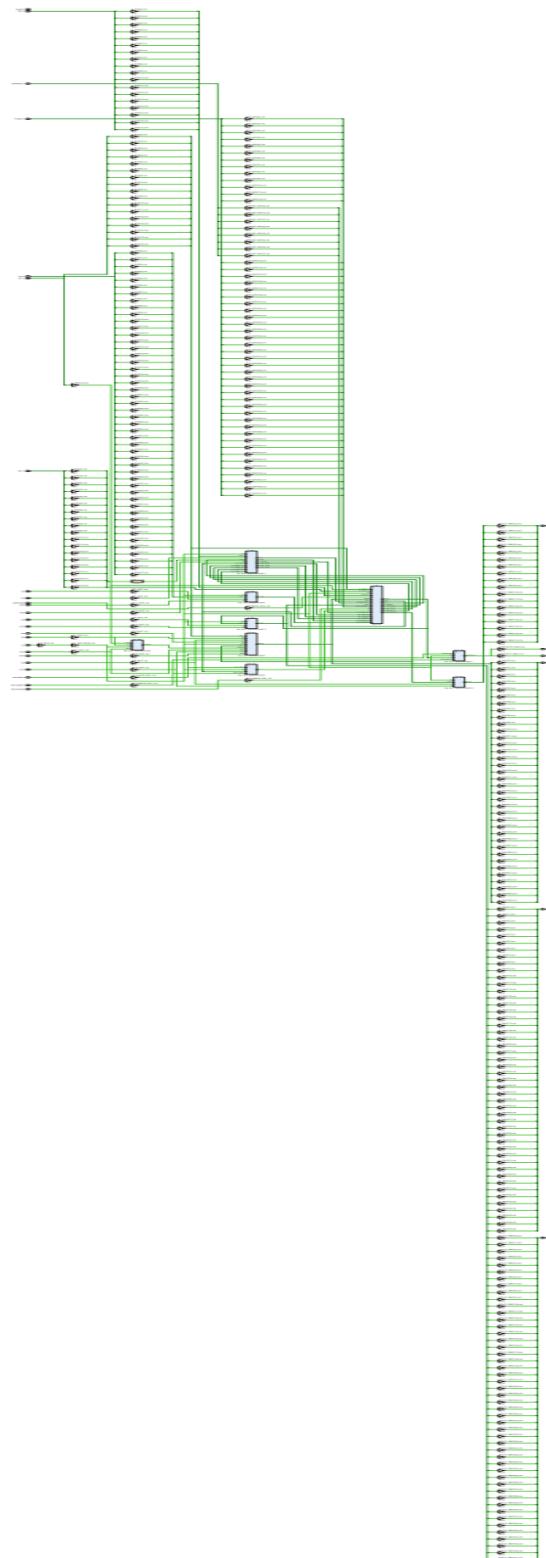
Hierarchy	Name	1 (134600)	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
Slice Logic	DSP	217	143	1	327	1	
	A1_reg (reg_mux_par...	0	1	0	0	0	
	B1_reg (reg_mux_pa...	0	18	0	0	0	
	C_reg (reg_mux_pa...	0	48	0	0	0	
	CYL_reg (reg_mux_p...	1	1	0	0	0	
Memory	CYO_reg (reg_mux_p...	0	1	0	0	0	
	D_reg (reg_mux_pa...	0	18	0	0	0	
	M_reg (reg_mux_pa...	0	0	1	0	0	
	op (reg_mux)	215	8	0	0	0	
	P_reg (reg_mux_pa...	0	48	0	0	0	

3. Timing Report





4. Schematic Snippets



8. Implementation

1. “Messages” Tab



2. Utilization Report

IMPLEMENTED DESIGN - xc7a200tffg1156-3 (active)

Utilization										
	Name	1	Slice LUTs (133800)	Slice Registers (267600)	Slice (33450)	LUT as Logic (133800)	LUT Flip Flop Pairs (133800)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
Hierarchy	N DSP	216		144	82	216		49	1	327
Summary	A1_reg (reg_mux_par...	0		1	1	0		0	0	0
Slice Logic	B1_reg (reg_mux_pa...	0		18	5	0		0	0	0
Slice LUTs (<1%)	C_reg (reg_mux_par...	0		48	14	0		0	0	0
Slice Registers (<1%)	CYL_reg (reg_mux_p...	1		1	1	1		1	0	0
Slice Logic Distribution	CYO_reg (reg_mux_p...	0		2	1	0		0	0	0
Slice (<1%)	D_reg (reg_mux_par...	0		18	10	0		0	0	0
SLICEM	M_reg (reg_mux_par...	0		0	0	0		0	1	0
SLICEL	op (reg_mux)	215		8	62	215		0	0	0
LUT Flip Flop Pairs (<1%)	P_reg (reg_mux_par...	0		48	12	0		0	0	0
LUT-FF pairs with o										
LUT-FF pairs with o										

3. Timing Report

Tcl Console Messages Log Reports Design Runs Power DRC Methodology Timing Utilization

Design Timing Summary

General Information	Setup	Hold	Pulse Width
Timer Settings	Worst Negative Slack (WNS): 3.820 ns	Worst Hold Slack (WHS): 0.266 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Design Timing Summary	Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Clock Summary (1)	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
> Check Timing (326)	Total Number of Endpoints: 107	Total Number of Endpoints: 107	Total Number of Endpoints: 146
Intra-Clock Paths	All user specified timing constraints are met.		
Inter-Clock Paths			
Other Path Groups			
User Ignored Paths			

Timing Summary - impl_1 (saved) Timing Summary - timing_1

Path Properties

Path 1

Summary

Name	Path 1
Slack	3.820ns

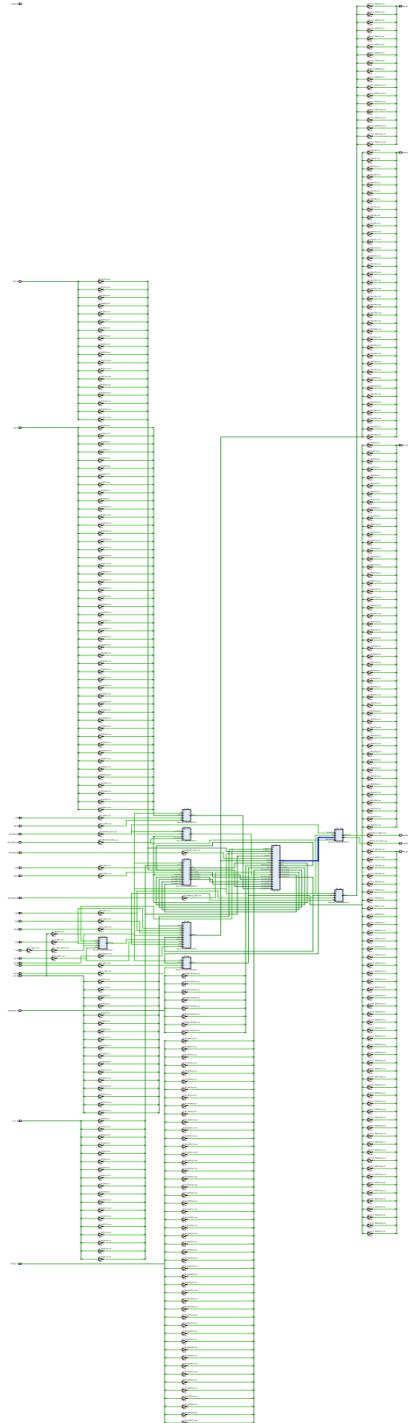
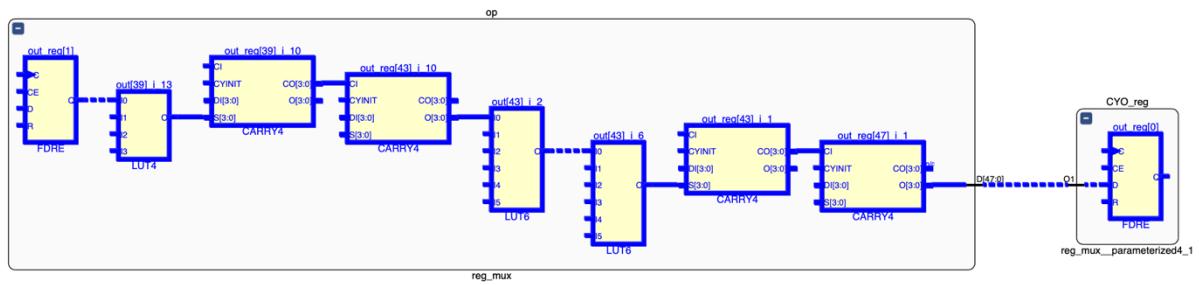
General Properties Report Cells Nets

IMPLEMENTED DESIGN - xc7a200tffg1156-3 (active)

Tcl Console Messages Log Reports Design Runs Power DRC Methodology Timing Utilization

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock
Path 1	3.820	7	6	96	op/out_reg[1]/C	CYO_reg/out_reg[0]/D	5.938	1.962	3.976	10.0	sys_clk_pin	sys_clk_pin
Path 2	3.930	7	6	96	op/out_reg[1]/C	CYO_reg/out_reg...lptp_replica/D	5.818	1.962	3.856	10.0	sys_clk_pin	sys_clk_pin
Path 3	4.839	7	6	96	op/out_reg[1]/C	P_reg/out_reg[47]/D	5.129	1.962	3.167	10.0	sys_clk_pin	sys_clk_pin
Path 4	4.864	7	5	96	op/out_reg[1]/C	P_reg/out_reg[45]/D	5.111	1.944	3.167	10.0	sys_clk_pin	sys_clk_pin
Path 5	4.913	7	5	96	op/out_reg[1]/C	P_reg/out_reg[46]/D	5.062	1.895	3.167	10.0	sys_clk_pin	sys_clk_pin
Path 6	4.935	7	5	96	op/out_reg[1]/C	P_reg/out_reg[44]/D	5.040	1.873	3.167	10.0	sys_clk_pin	sys_clk_pin
Path 7	5.179	6	4	96	op/out_reg[1]/C	P_reg/out_reg[43]/D	4.796	1.782	3.014	10.0	sys_clk_pin	sys_clk_pin
Path 8	5.196	6	4	96	op/out_reg[1]/C	P_reg/out_reg[41]/D	4.779	1.775	3.004	10.0	sys_clk_pin	sys_clk_pin
Path 9	5.224	6	4	96	op/out_reg[1]/C	P_reg/out_reg[42]/D	4.751	1.737	3.014	10.0	sys_clk_pin	sys_clk_pin
Path 10	5.267	6	4	96	op/out_reg[1]/C	P_reg/out_reg[40]/D	4.708	1.704	3.004	10.0	sys_clk_pin	sys_clk_pin



4. Schematic Snippets

