Project 1

Eng/ Kareem Waseem

By: Omar Ahmed Fouad

Contents

1-RTL code

```
module Reg_Mux(In,Sel,Out,Clk,Rst,Enable);

parameter WIDTH_REG_MUX_IN = 18;
parameter WIDTH_REG_MUX_OUT=18;
parameter TYPE = "SYNC";

input [WIDTH_REG_MUX_IN-1:0] In;
input Sel,Rst,Clk,Enable;

output [WIDTH_REG_MUX_OUT-1:0] Out;

reg [WIDTH_REG_MUX_IN-1:0] In_reg;
//Async or Sync Rst
generate
    if (TYPE == "SYNC") begin
        always @(posedge Clk) begin
            if(Rst)
                In_reg <= 0;
            else begin
                if(Enable)
                    In_reg <= In;
            end
        end
    end
    else begin
        always @(posedge Clk or posedge Rst) begin
            if(Rst)
                In_reg <= 0;
            else begin
                if(Enable)
                    In_reg <= In;
            end
        end
    end
endgenerate

assign Out = (Sel)? In_reg : In;

endmodule
```

```
module Mux(In0,In1,Sel,Out);

parameter WIDTH_MUX_IN0 = 18;
parameter WIDTH_MUX_IN1 = 18;
parameter WIDTH_MUX_OUT = 18;

input [WIDTH_MUX_IN0-1:0] In0;
input [WIDTH_MUX_IN1-1:0] In1;
input Sel;

output [WIDTH_MUX_OUT-1:0] Out;

assign Out = (Sel)? In1 : In0;

endmodule
```

```
module Mux_4x1(In0,In1,In2,In3,Sel,Out);

input [47:0] In0,In1,In2,In3;
input [1:0] Sel;

output wire [47:0] Out;

assign Out = (Sel == 2'b00) ? In0 :
             (Sel == 2'b01) ? In1 :
             (Sel == 2'b10) ? In2 :
             (Sel == 2'b11) ? In3 :
             In0; // Default case

endmodule
```

```
module Spartan_6(
    A,B,D,C
    ,CLK,CARRYIN
    ,OPMODE,BCIN,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE
    ,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE
    ,PCIN
    ,BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF
    );
//parameters for A and B inputs that define whether the mux out is reg or not (SELECT LINE OF REG_MUX)
parameter A0REG = 0;
parameter B0REG = 0; //0 refers to no registers
parameter A1REG = 1; //1 refers to registers
parameter B1REG = 1;
//parameters for C,D,M,P,CARRYIN,CARRYOUT,OPMODE that define whether the mux out is reg or not (SELECT LINE OF REG_MUX)
parameter CREG = 1;
parameter DREG = 1;
parameter MREG = 1;
parameter PREG = 1;
parameter CARRYINREG = 1;
parameter CARRYOUTREG = 1;
parameter OPMODEREG = 1;
//CARRYINSEL either the CARRYIN input will be considered or the value of opcode[5] , Values = "CARRYIN" , "OPMODE5"
wire CARRYIN_SEL;
parameter CARRYINSEL = "OPMODE5";

assign CARRYIN_SEL = (CARRYINSEL == "OPMODE5")? 1 : 0;

//B_INPUT defines whether the input to the B port is <B input> (DIRECT) or <cascaded input BCIN> from the previous DSP48A1 (CASCADE)
wire B_SEL;
parameter B_INPUT = "DIRECT";

assign B_SEL = (B_INPUT == "DIRECT")? 1 : 0;

//RSTTYPE selects whether all resets for the DSP48A1 should have a synchronous or asynchronous reset , Values = "ASYNC" , "SYNC"
parameter RSTTYPE = "SYNC";
//inputs
input [17:0] A,B,D,BCIN;
input [7:0] OPMODE;
input [47:0] C,PCIN;

input CARRYIN,CLK;
input CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP; //clk enable for the Reg_Mux for each input
input RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP; //rst active high for the Reg_Mux for each input, Sync or Async depending on RSTTYPE
```

```
//output
output CARRYOUT,CARRYOUTF;
output [17:0] BCOUT;
output [35:0] M;
output [47:0] PCOUT,P;
//Needed Internal Signals
wire CIN,COUT,Carry_Cascade;
wire [7:0] OPMODE_Mux;
wire [17:0] D_Mux,B_Mux,B0_Mux,A0_Mux,Pre_Add_Mux_Out,B1_Mux,A1_Mux;
reg [17:0] Pre_Add;
wire [35:0] Multi_Value,M_Reg_Mux_Out;
wire [47:0] C_Mux,Concatenation,X_Out,Z_Out,Post_Add;
wire [48:0] Full_Add;
//inputs into Reg_Mux
Mux B_Input (.In0(BCIN),.In1(B),.Sel(B_SEL),.Out(B_Mux));//which will be B_port

Reg_Mux #(.TYPE(RSTTYPE)) D_Entry (.In(D),.Sel(DREG),.Out(D_Mux),.Clk(CLK),.Rst(RSTD),.Enable(CED));//D input entry
Reg_Mux #(.TYPE(RSTTYPE)) B_Entry (.In(B_Mux),.Sel(B0REG),.Out(B0_Mux),.Clk(CLK),.Rst(RSTB),.Enable(CEB));//B input entry
Reg_Mux #(.TYPE(RSTTYPE)) A_Entry (.In(A),.Sel(A0REG),.Out(A0_Mux),.Clk(CLK),.Rst(RSTA),.Enable(CEA));//A0 input entry
Reg_Mux #(.WIDTH_REG_MUX_IN(48),.WIDTH_REG_MUX_OUT(48),.TYPE(RSTTYPE)) C_Entry (.In(C),.Sel(CREG),.Out(C_Mux),.Clk(CLK),.Rst(RSTC),.Enable(CEC));//C input entry
Reg_Mux #(.WIDTH_REG_MUX_IN(8),.WIDTH_REG_MUX_OUT(8),.TYPE(RSTTYPE)) OPMODE_Entry (.In(OPMODE),.Sel(OPMODEREG),.Out(OPMODE_Mux),.Clk(CLK),.Rst(RSTOPMODE),.Enable(CEOPMODE));
//operations till multiplication
always@(*) begin
    if(OPMODE_Mux[6])
        Pre_Add = D_Mux - B0_Mux;
    else
        Pre_Add = D_Mux + B0_Mux;
end
Mux Pre_Add_Mux (.In0(B0_Mux),.In1(Pre_Add),.Sel(OPMODE_Mux[4]),.Out(Pre_Add_Mux_Out));
//B1,A1 reg
Reg_Mux #(.TYPE(RSTTYPE)) B1_Reg_Mux (.In(Pre_Add_Mux_Out),.Sel(B1REG),.Out(B1_Mux),.Clk(CLK),.Rst(RSTB),.Enable(CEB));
Reg_Mux #(.TYPE(RSTTYPE)) A1_Reg_Mux (.In(A0_Mux),.Sel(A1REG),.Out(A1_Mux),.Clk(CLK),.Rst(RSTA),.Enable(CEA));
assign BCOUT = B1_Mux;
//Multiplication
assign Multi_Value = (B1_Mux * A1_Mux);
Reg_Mux #(.WIDTH_REG_MUX_IN(36), .WIDTH_REG_MUX_OUT(36),.TYPE(RSTTYPE)) M_Reg_Mux (.In(Multi_Value),.Sel(MREG),.Out(M_Reg_Mux_Out),.Clk(CLK),.Rst(RSTM),.Enable(CEM));
assign M = M_Reg_Mux_Out;
//X_MUX , Z_MUX
assign Concatenation = {D_Mux[11:0],A1_Mux[17:0],B1_Mux[17:0]};

Mux_4x1 X (.In0(48'b0),.In1({12'b0,M_Reg_Mux_Out}),.In2(P),.In3(Concatenation),.Sel(OPMODE_Mux[1:0]),.Out(X_Out));

Mux_4x1 Z (.In0(48'b0),.In1(PCIN),.In2(P),.In3(C_Mux),.Sel(OPMODE_Mux[3:2]),.Out(Z_Out));
```

```
//pereparing CIN for the 2nd ADD
assign Carry_Cascade = (CARRYIN_SEL)? OPMODE_Mux[5] : CARRYIN;
Reg_Mux #(.TYPE(RSTTYPE),.WIDTH_REG_MUX_IN(1),.WIDTH_REG_MUX_OUT(1)) CYI (.In(Carry_Cascade),.Sel(CARRYINREG),.Out(CIN),.Clk(CLK),.Rst(RSTCARRYIN),.Enable(CECARRYIN));
//second Add
assign Full_Add = (OPMODE_Mux[7])? ({1'b0,Z_Out}-({1'b0,X_Out}+CIN)) : ({1'b0,Z_Out}+{1'b0,X_Out}+CIN);
assign Post_Add = Full_Add [47:0];
assign COUT = Full_Add [48];

Reg_Mux #(.WIDTH_REG_MUX_IN(48),.WIDTH_REG_MUX_OUT(48),.TYPE(RSTTYPE)) P_Reg_Mux (.In(Post_Add),.Sel(PREG),.Out(P),.Clk(CLK),.Rst(RSTP),.Enable(CEP));
assign PCOUT = P;
Reg_Mux #(.TYPE(RSTTYPE),.WIDTH_REG_MUX_IN(1),.WIDTH_REG_MUX_OUT(1)) COUT_Reg_Mux (.In(COUT),.Sel(CARRYOUTREG),.Out(CARRYOUT),.Clk(CLK),.Rst(RSTCARRYIN),.Enable(CECARRYIN));
assign CARRYOUTF = CARRYOUT;

endmodule
```

2-Testbench code

```
module Spartan_6_tb();

//stimuls and response
reg [17:0] A_tb,B_tb,D_tb,BCIN_tb;
reg [7:0] OPMODE_tb;
reg [47:0] C_tb,PCIN_tb;

reg CARRYIN_tb,CLK_tb;
reg CEA_tb,CEB_tb,CEC_tb,CECARRYIN_tb,CED_tb,CEM_tb,CEOPMODE_tb,CEP_tb;
reg RSTA_tb,RSTB_tb,RSTC_tb,RSTCARRYIN_tb,RSTD_tb,RSTM_tb,RSTOPMODE_tb,RSTP_tb;

wire CARRYOUT_DUT,CARRYOUTF_DUT;
wire [17:0] BCOUT_DUT;
wire [35:0] M_DUT;
wire [47:0] PCOUT_DUT,P_DUT;

reg CARRYOUT_Expected;
reg [17:0] BCOUT_Expected;
reg [35:0] M_Expected;
reg [47:0] P_Expected;

//DUT
Spartan_6 DUT (
    // -- Inputs --
    .A(A_tb),
    .B(B_tb),
    .D(D_tb),
    .C(C_tb),
    .CLK(CLK_tb),
    .CARRYIN(CARRYIN_tb),
    .OPMODE(OPMODE_tb),
    .BCIN(BCIN_tb),
    .PCIN(PCIN_tb),

    // -- Reset Inputs --
    .RSTA(RSTA_tb),
    .RSTB(RSTB_tb),
    .RSTC(RSTC_tb),
    .RSTD(RSTD_tb),
    .RSTM(RSTM_tb),
    .RSTP(RSTP_tb),
    .RSTCARRYIN(RSTCARRYIN_tb),
    .RSTOPMODE(RSTOPMODE_tb),
```

```

    // -- Clock Enable Inputs --
    .CEA(CEA_tb),
    .CEB(CEB_tb),
    .CEC(CEC_tb),
    .CED(CED_tb),
    .CEM(CEM_tb),
    .CEP(CEP_tb),
    .CEOPMODE(CEOPMODE_tb),
    .CECARRYIN(CECARRYIN_tb),

    // -- Outputs --
    .BCOUT(BCOUT_DUT),
    .PCOUT(PCOUT_DUT),
    .P(P_DUT),
    .M(M_DUT),
    .CARRYOUT(CARRYOUT_DUT),
    .CARRYOUTF(CARRYOUTF_DUT)
);
//clk generation
initial begin
    CLK_tb = 0;
    forever
        #1 CLK_tb = ~CLK_tb;
end
//start from known states <wait> >> (Randomize - wait - check)for loop
initial begin
//rst check
RSTA_tb = 1;
RSTB_tb = 1;
RSTC_tb = 1;
RSTCARRYIN_tb = 1;
RSTD_tb = 1;
RSTM_tb = 1;
RSTOPMODE_tb = 1;
RSTP_tb = 1;

A_tb = $random;
B_tb = $random;
D_tb = $random;
BCIN_tb = $random;
OPMODE_tb = $random;
C_tb = $random;
```

```
PCIN_tb = $random;
CARRYIN_tb = $random;

CEA_tb = $random;
CEB_tb = $random;
CEC_tb = $random;
CECARRYIN_tb = $random;
CED_tb = $random;
CEM_tb = $random;
CEOPMODE_tb = $random;
CEP_tb = $random;

@(negedge CLK_tb);
if(CARRYOUT_DUT != 0 || CARRYOUTF_DUT != 0 || BCOUT_DUT != 0 || M_DUT != 0 || PCOUT_DUT != 0 || P_DUT != 0) begin
    $display ("ERROR In Reset");
    $stop;
end
//Path 1 check
RSTA_tb = 0;
RSTB_tb = 0;
RSTC_tb = 0;
RSTCARRYIN_tb = 0;
RSTD_tb = 0;
RSTM_tb = 0;
RSTOPMODE_tb = 0;
RSTP_tb = 0;

CEA_tb = 1;
CEB_tb = 1;
CEC_tb = 1;
CECARRYIN_tb = 1;
CED_tb = 1;
CEM_tb = 1;
CEOPMODE_tb = 1;
CEP_tb = 1;

OPMODE_tb = 8'b11011101;
A_tb = 20;
B_tb = 10;
C_tb = 350;
D_tb = 25;

BCOUT_Expected = 'hf;
M_Expected = 'h12c;
```

```
P_Expected = 'h32;
CARRYOUT_Expected = 0;

repeat(4) @(negedge CLK_tb);
if(BCOUT_DUT != BCOUT_Expected || M_DUT != M_Expected || P_DUT != P_Expected || PCOUT_DUT != P_Expected || CARRYOUT_DUT != CARRYOUT_Expected || CARRYOUTF_DUT != CARRYOUT_Expected) begin
    $display("ERROR In Path 1");
    $stop;
end
//path 2 check
OPMODE_tb = 8'b00010000;
A_tb = 20;
B_tb = 10;
C_tb = 350;
D_tb = 25;

BCOUT_Expected = 'h23;
M_Expected = 'h2bc;
P_Expected = 'h0;
CARRYOUT_Expected = 0;

repeat(3) @(negedge CLK_tb);
if(BCOUT_DUT != BCOUT_Expected || M_DUT != M_Expected || P_DUT != P_Expected || PCOUT_DUT != P_Expected || CARRYOUT_DUT != CARRYOUT_Expected || CARRYOUTF_DUT != CARRYOUT_Expected) begin
    $display("ERROR In Path 2");
    $stop;
end
//path 3 check
OPMODE_tb = 8'b00001010;
A_tb = 20;
B_tb = 10;
C_tb = 350;
D_tb = 25;

BCOUT_Expected = 'ha;
M_Expected = 'hc8;
P_Expected = 'h0;
CARRYOUT_Expected = 0;

repeat(3) @(negedge CLK_tb);
if(BCOUT_DUT != BCOUT_Expected || M_DUT != M_Expected || P_DUT != P_Expected || PCOUT_DUT != P_Expected || CARRYOUT_DUT != CARRYOUT_Expected || CARRYOUTF_DUT != CARRYOUT_Expected) begin
    $display("ERROR In Path 3");
    $stop;
end
//path 4 check
OPMODE_tb = 8'b10100111;
```

```
A_tb = 5;
B_tb = 6;
C_tb = 350;
D_tb = 25;
PCIN_tb = 3000;

BCOUT_Expected = 'h6;
M_Expected = 'h1e;
P_Expected = 'hfe6fffec0bb1;
CARRYOUT_Expected = 1;

repeat(3) @(negedge CLK_tb);
if(BCOUT_DUT != BCOUT_Expected || M_DUT != M_Expected || P_DUT != P_Expected || PCOUT_DUT != P_Expected || CARRYOUT_DUT != CARRYOUT_Expected || CARRYOUTF_DUT != CARRYOUT_Expected) begin
    $display("ERROR In Path 4");
    $stop;
end
else
    $display("DSP Is Working");


$stop;
end

endmodule
```

3-Do file

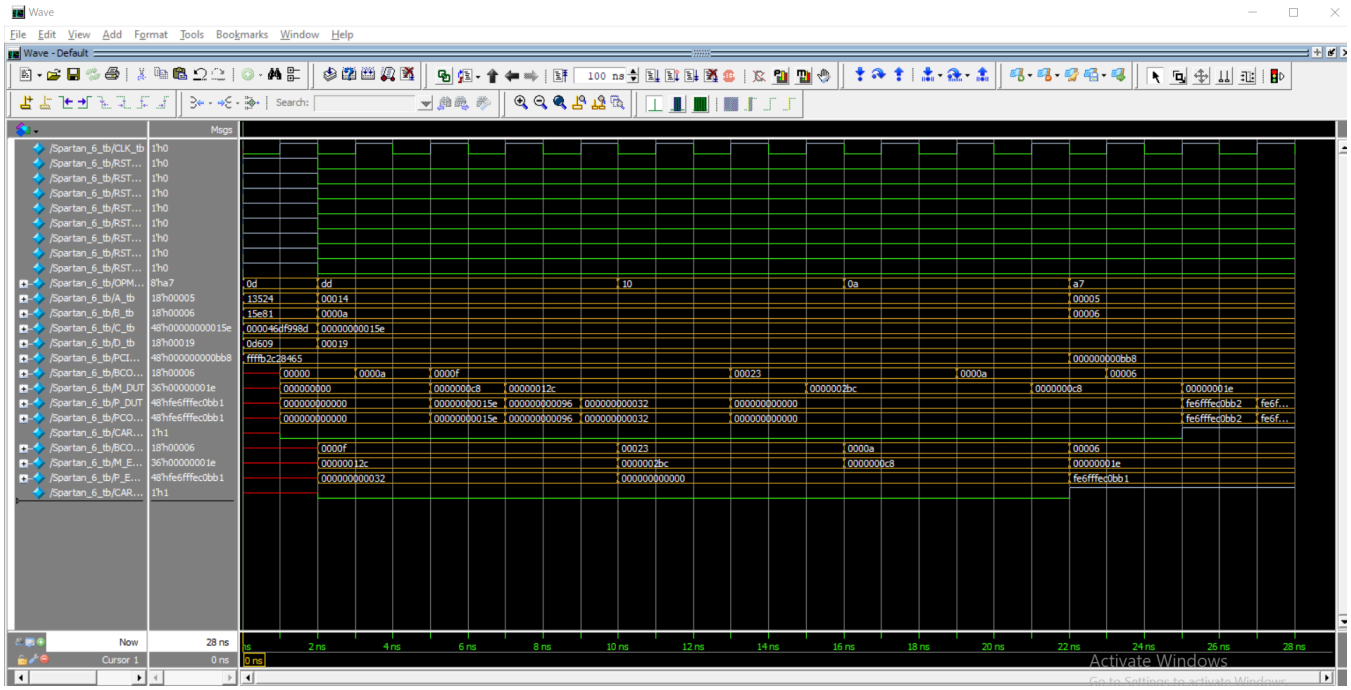
run_DSP_tb.do - Notepad

File Edit Format View Help

```
# Create library
vlib work

# Compile the Verilog file
vlog Spartan_6_tb.v

# Start the simulation with full visibility
vsim -voptargs=+acc Spartan_6_tb

# Add specific waves for debugging

add wave CLK_tb
add wave *RST*_tb

add wave OPMODE_tb
add wave A_tb
add wave B_tb
add wave C_tb
add wave D_tb
add wave PCIN_tb

add wave BCOUT_DUT
add wave M_DUT
add wave P_DUT
add wave PCOUT_DUT
add wave CARRYOUT_DUT

add wave BCOUT_Expected
add wave M_Expected
add wave P_Expected
add wave CARRYOUT_Expected

# Run the simulation until it finishes
run -all

# uncomment to automatically close the simulation
# quit -sim
```

4-QuestaSim Snippets

```
# DSP Is Working
# ** Note: $stop    : Spartan_6_tb.v(195)
#    Time: 28 ns  Iteration: 1  Instance: /Spartan_6_tb
# Break in Module Spartan_6_tb at Spartan_6_tb.v line 195
```

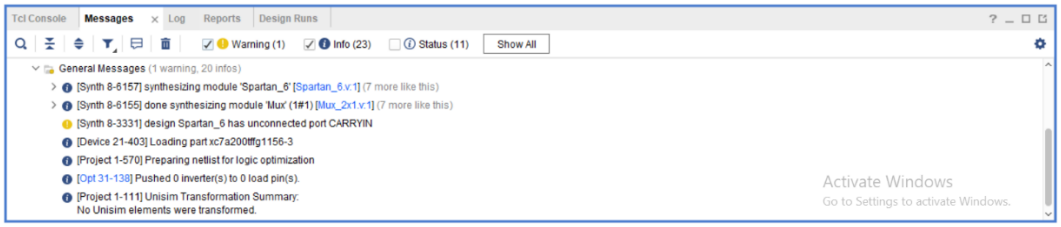


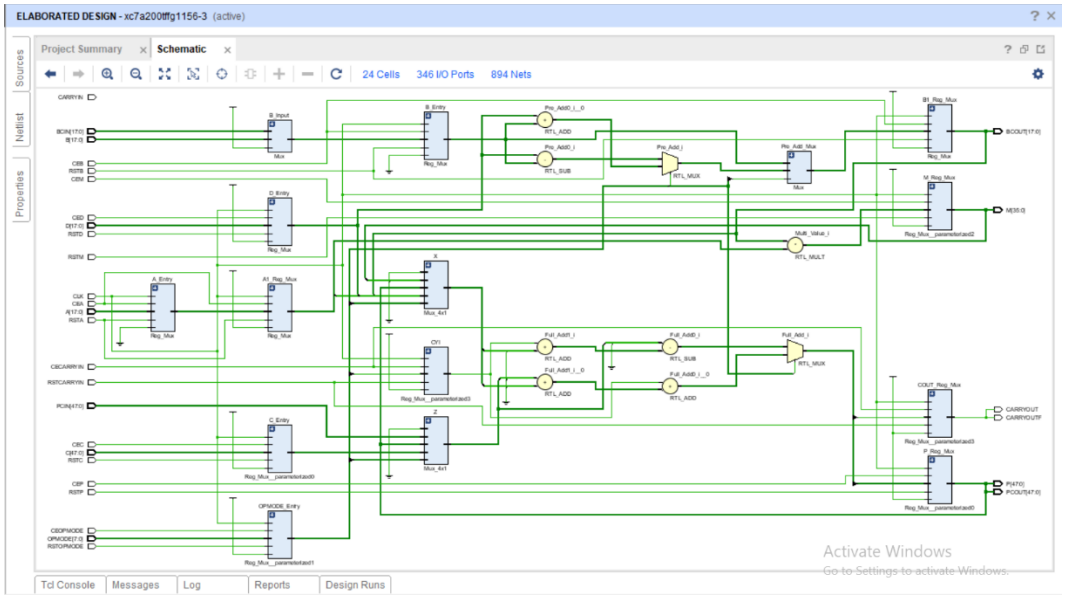
5-Constraint File

```
## 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]
```

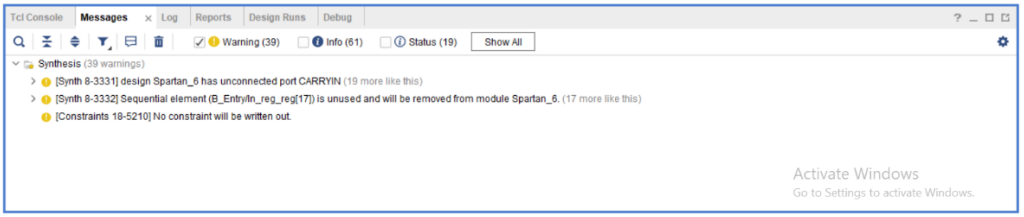
6-Vivado Snippets

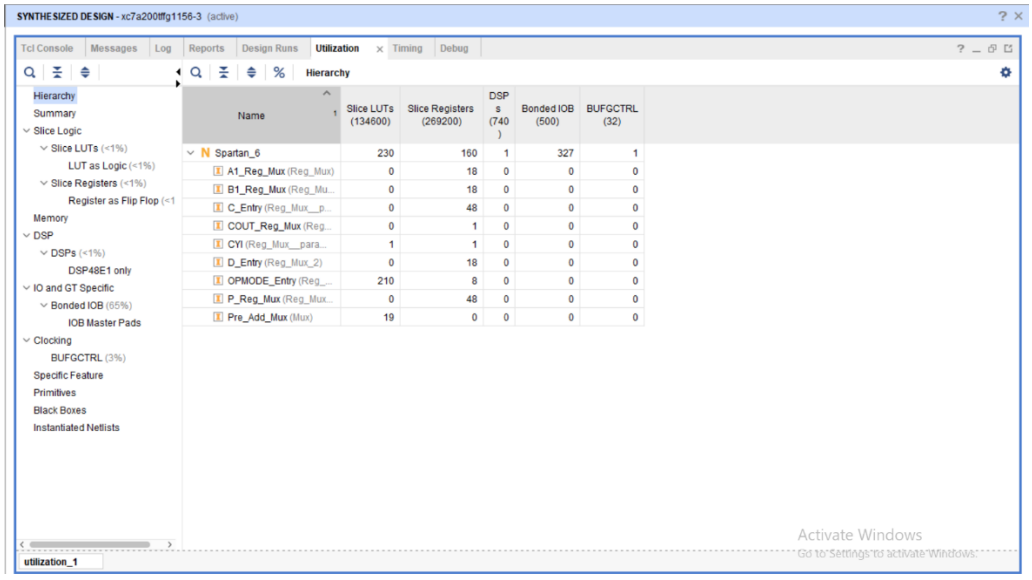
6.1 - Elaboration ("Messages" tab & Schematic snippets)

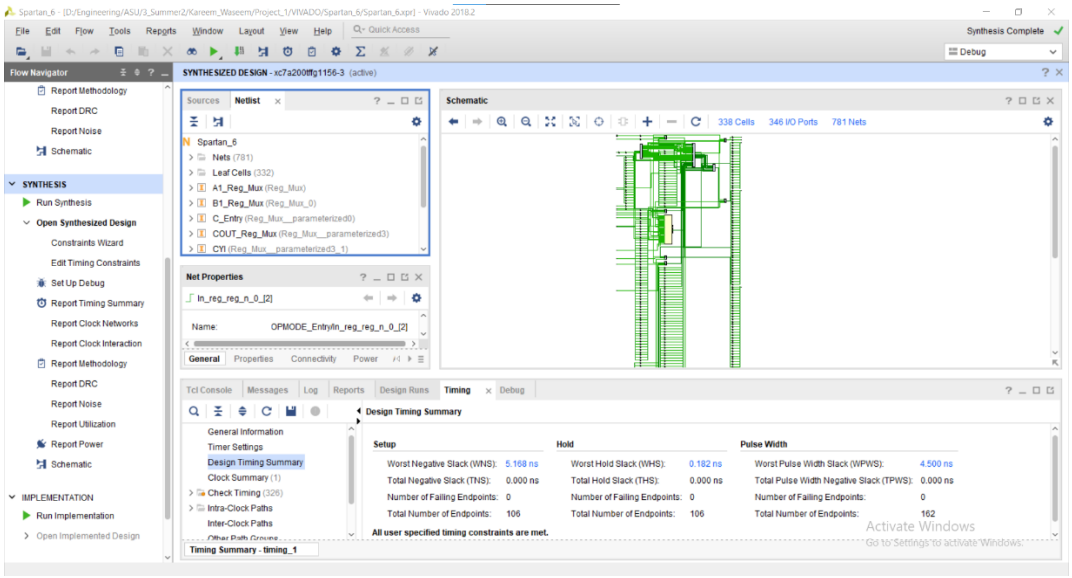


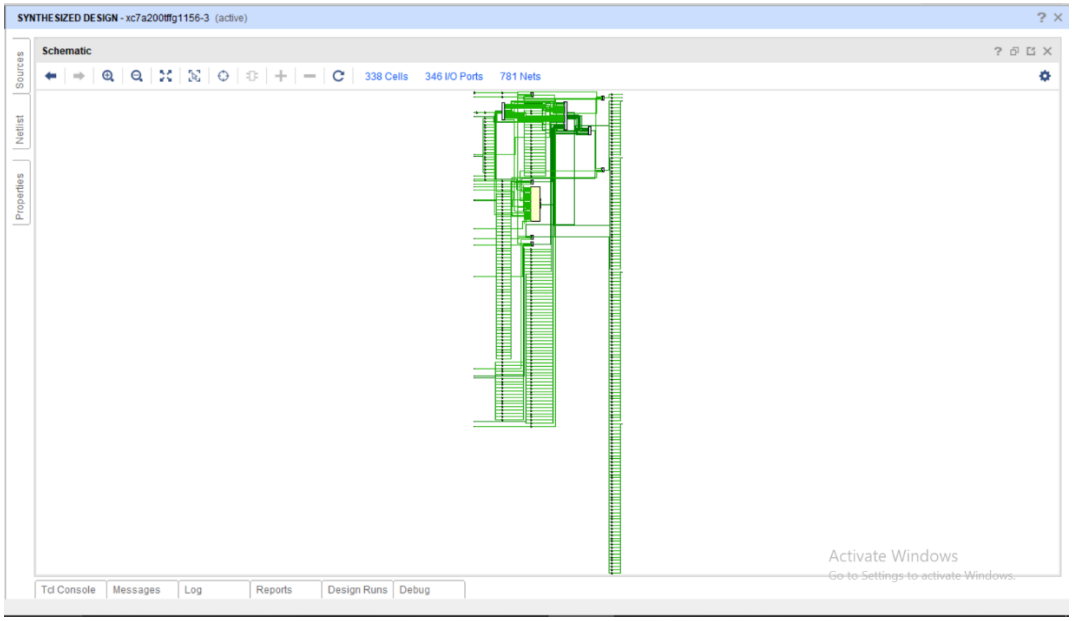


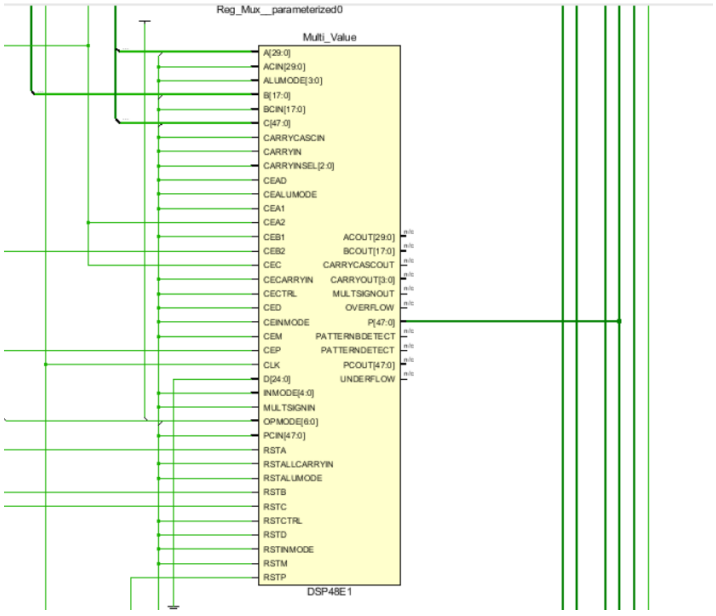
6.2 - Synthesis ("Messages" tab, Utilization report, timing report & Schematic snippets)



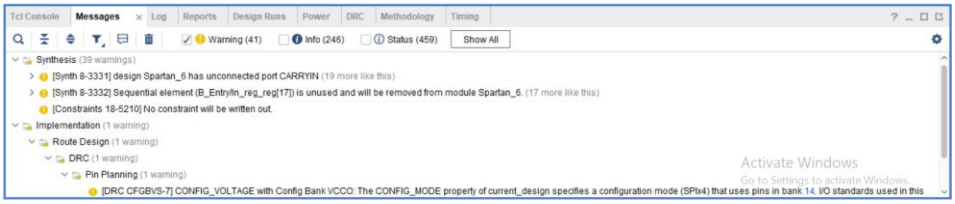


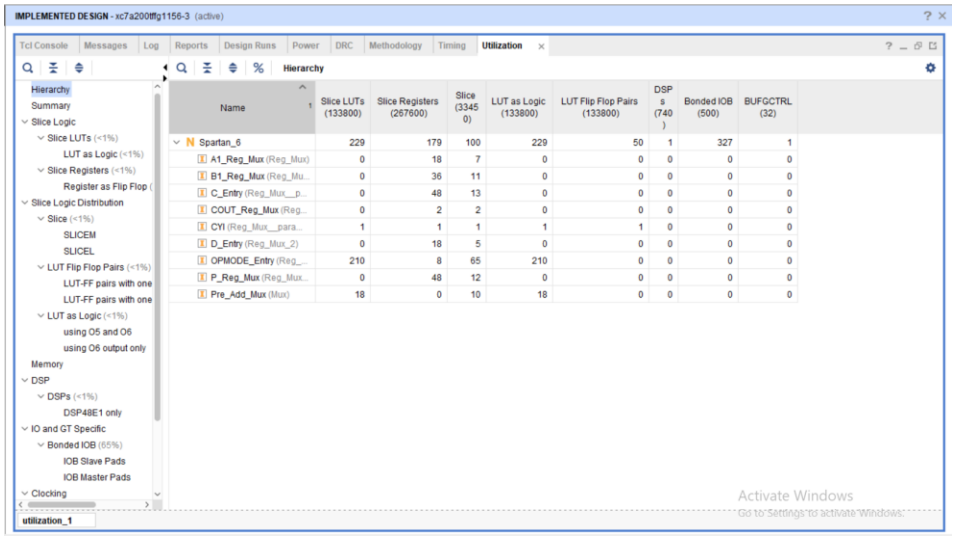

Synthesis Complete
SYNTHESIZED DESIGN - xc7a200tffg1156-3 (active)
Schematic
338 Cells
346 I/O Ports
781 Nets
Design Timing Summary
All user specified timing constraints are met.

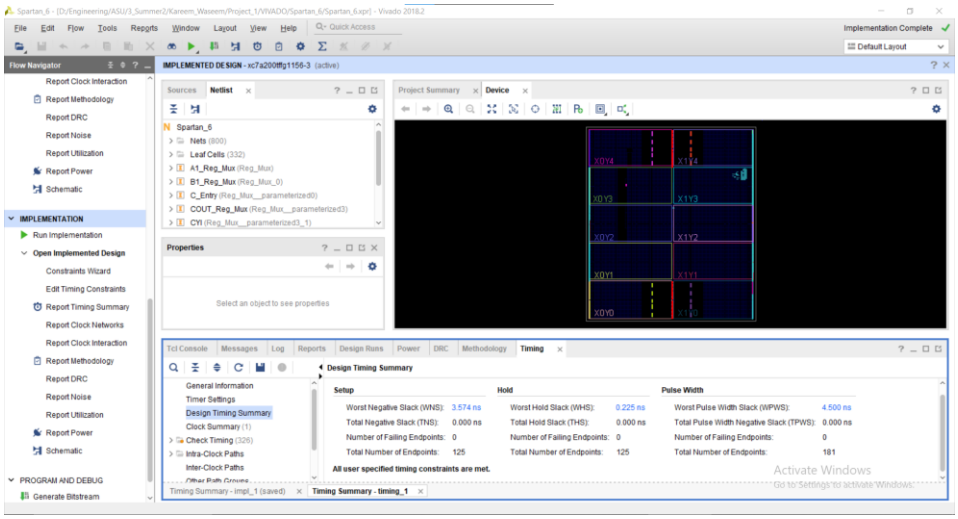

SYNTHESIZED DESIGN - xc7a200tffg1156-3 (active)
Schematic
338 Cells
346 I/O Ports
781 Nets

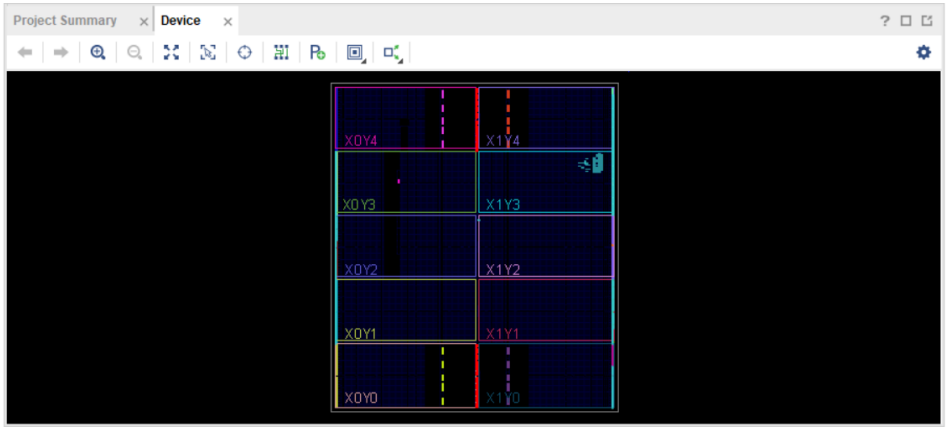

Reg_Mux__parameterized0
Multi_Value
DSP48E1

6.3 - Implementation ("Messages" tab, Utilization report, timing report & device snippets)









7-QuestaLint Snippets