

# **Project 1**

## **DSP**

### **ANUBIS TEAM**

- |    |                                 |
|----|---------------------------------|
| 1) | Mustafa Ibrahim Mohamed Ibrahim |
| 2) | Youssef Ehab Mamdouh Gorgy      |
| 3) | Nada Mamdouh Ismail             |

## 1) RTL CODE

### a) Design

```
module Spartan6_DSP48A1_DUT (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);

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";
parameter RSTTYPE = "SYNC";

input [17:0] A, B, D;
input [47:0] C;

input CLK, CARRYIN;

input [7:0] OPMODE;
input [17:0] BCIN;

input RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPMODE;

input CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPMODE;

input [47:0] PCIN;

output [17:0] BCOUT;
output [47:0] PCOUT, P;
output [35:0] M;
output CARRYOUT, CARRYOUTF;

reg [17:0] A0_REG, A1_REG, B0_REG, B1_REG, D_REG;
wire [17:0] mux_A0_out, mux_A1_out, mux_B0_out, mux_B1_out, mux_D_out,
Pre_Adder_out, mux_Pre_Adder_out;
reg [17:0] B_input_out;
reg [47:0] C_REG;
wire [47:0] mux_C_out, D_A_B_Concatenation;
reg [7:0] OPMODE_REG;
wire [7:0] mux_OPMODE_out;
reg [35:0] M_REG;
wire [35:0] multiplier_out, mux_multiplier_out;
reg [47:0] mux_X_out, mux_Z_out;
wire Post_Adder_CIN;
reg mux_CARRYINSEL_out, CYI_REG, CYO_REG;
wire [47:0] Post_Adder_out;
```

```

reg [47:0] P_REG;
wire Post_Adder_CO;

//////////////////////////////Sequential_Logic/////////////////
/////////////////////////////Sequential_Logic/////////////////
/////////////////////////////Sequential_Logic/////////////////


generate
    if (RSTTYPE == "SYNC") begin

        always @ (posedge CLK) begin
            if (RSTA) begin
                A0_REG <= 0;
                A1_REG <= 0;
            end
            else if (CEA) begin
                A0_REG <= A;
                A1_REG <= mux_A0_out;
            end
        end

        always @ (posedge CLK) begin
            if (RSTB) begin
                B0_REG <= 0;
                B1_REG <= 0;
            end
            else if (CEB) begin
                B0_REG <= B_input_out;
                B1_REG <= mux_Pre_Adder_out;
            end
        end

        always @ (posedge CLK) begin
            if (RSTC) begin
                C_REG <= 0;
            end
            else if (CEC) begin
                C_REG <= C;
            end
        end

        always @ (posedge CLK) begin
            if (RSTD) begin
                D_REG <= 0;
            end
            else if (CEC) begin
                D_REG <= D;
            end
        end

        always @ (posedge CLK) begin
            if (RSTOPMODE) begin
                OPMODE_REG <= 0;
            end
            else if (CEOPMODE) begin
                OPMODE_REG <= OPMODE;
            end
        end
    end
end

```

```

        end

    always @ (posedge CLK) begin
        if (RSTC) begin
            C_REG <= 0;
        end
        else if (CEC) begin
            C_REG <= C;
        end
    end

always @ (posedge CLK) begin
    if (RSTM) begin
        M_REG <= 0;
    end
    else if (CEM) begin
        M_REG <= multiplier_out;
    end
end

always @ (posedge CLK) begin
    if (RSTCARRYIN) begin
        CYI_REG <= 0;
        CYO_REG <= 0;
    end
    else if (CECARRYIN) begin
        CYI_REG <= mux_CARRYINSEL_out;
        CYO_REG <= Post_Adder_CO;
    end
end

always @ (posedge CLK) begin
    if (RSTP) begin
        P_REG <= 0;
    end
    else if (CEP) begin
        P_REG <= Post_Adder_out;
    end
end

end
else if (RSTTYPE == "ASYNC") begin

    always @ (posedge CLK or posedge RSTA) begin
        if (RSTA) begin
            A0_REG <= 0;
            A1_REG <= 0;
        end
        else if (CEA) begin
            A0_REG <= A;
            A1_REG <= mux_A0_out;
        end
    end

    always @ (posedge CLK or posedge RSTB) begin
        if (RSTB) begin
            B0_REG <= 0;
            B1_REG <= 0;
        end
        else if (CEB) begin

```

```

        B0_REG <= B_input_out;
        B1_REG <= mux_Pre_Adder_out;
    end
end

always @ (posedge CLK or posedge RSTC) begin
    if (RSTC) begin
        C_REG <= 0;
    end
    else if (CEC) begin
        C_REG <= C;
    end
end

always @ (posedge CLK or posedge RSTD) begin
    if (RSTD) begin
        D_REG <= 0;
    end
    else if (CEC) begin
        D_REG <= D;
    end
end

always @ (posedge CLK or posedge RSTOPMODE) begin
    if (RSTOPMODE) begin
        OPMODE_REG <= 0;
    end
    else if (CEOPMODE) begin
        OPMODE_REG <= OPMODE;
    end
end

always @ (posedge CLK or posedge RSTC) begin
    if (RSTC) begin
        C_REG <= 0;
    end
    else if (CEC) begin
        C_REG <= C;
    end
end

always @ (posedge CLK or posedge RSTM) begin
    if (RSTM) begin
        M_REG <= 0;
    end
    else if (CEM) begin
        M_REG <= multiplier_out;
    end
end

always @ (posedge CLK or posedge RSTCARRYIN) begin
    if (RSTCARRYIN) begin
        CYI_REG <= 0;
        CYO_REG <= 0;
    end
    else if (CECARRYIN) begin
        CYI_REG <= mux_CARRYINSEL_out;
        CYO_REG <= Post Adder CO;
    end
end

```

```

        end
    end
    always @ (posedge CLK or posedge RSTP) begin
        if (RSTP) begin
            P_REG <= 0;
        end
        else if (CEP) begin
            P_REG <= Post_Adder_out;
        end
    end
end
endgenerate

//////////////////////////////Combinational_Logic/////////////////
//////////////////////////////Combinational_Logic/////////////////
//////////////////////////////Combinational_Logic/////////////////


//Gray_colored_multiplexers of A input
assign mux_A0_out = (A0REG) ? A0_REG : A;
assign mux_A1_out = (A1REG) ? A1_REG : mux_A0_out;

//Gray_colored_multiplexer of C input
assign mux_C_out = (CREG) ? C_REG : C;

//Gray_colored_multiplexer of D input
assign mux_D_out = (DREG) ? D_REG : D;

//Gray_colored_multiplexers of B input
always @(*) begin
    if (B_INPUT == "DIRECT") begin
        B_input_out = B;
    end
    else if (B_INPUT == "CASCADE") begin
        B_input_out = BCIN;
    end
    else begin
        B_input_out = 0;
    end
end
assign mux_B0_out = (B0REG) ? B0_REG : B_input_out;
assign mux_B1_out = (B1REG) ? B1_REG : mux_Pre_Adder_out;

//Pre-Adder/Subtracter
assign Pre_Adder_out = (mux_OPMODE_out[6]) ? (mux_D_out - mux_B0_out) :
(mux_D_out + mux_B0_out);

//mux of Pre-Adder/Subtracter and B
assign mux_Pre_Adder_out = (mux_OPMODE_out[4]) ? Pre_Adder_out :
mux_B0_out;

//Gray_colored_multiplexer of OPMODE input
assign mux_OPMODE_out = (OPMODEREG) ? OPMODE_REG : OPMODE;

//output BCOUT
assign BCOUT = mux_B1_out;

//Concatenated D:A:B input signals

```

```

assign D_A_B_Concatenation = {mux_D_out,mux_A1_out,mux_B1_out};

//Multiplier
assign multiplier_out = mux_B1_out * mux_A1_out;
assign mux_multiplier_out = (MREG) ? M_REG : multiplier_out;

//output M (buffered)
assign M = ~(~mux_multiplier_out);

//mux_X
always @(*) begin
    case (mux_OPMODE_out[1:0])
        0: mux_X_out = 0;
        1: mux_X_out = mux_multiplier_out;
        2: mux_X_out = P; //accumulator
        3: mux_X_out = D_A_B_Concatenation;
    endcase
end

//mux_Z
always @(*) begin
    case (mux_OPMODE_out[3:2])
        0: mux_Z_out = 0;
        1: mux_Z_out = PCIN;
        2: mux_Z_out = P; //accumulator
        3: mux_Z_out = mux_C_out;
    endcase
end

//Gray_colored_multiplexer of CYI Cascade
always @(*) begin
    if (CARRYINSEL == "OPMODE5") begin
        mux_CARRYINSEL_out = mux_OPMODE_out[5];
    end
    else if (CARRYINSEL == "CARRYIN") begin
        mux_CARRYINSEL_out = CARRYIN;
    end
    else begin
        mux_CARRYINSEL_out = 0;
    end
end

//Gray_colored_multiplexer of CYI
assign Post_Adder_CIN = (CARRYINREG) ? CYI_REG : mux_CARRYINSEL_out;

//Post-Adder/Subtracter
assign {Post_Adder_CO , Post_Adder_out} = (mux_OPMODE_out[7]) ?
(mux_Z_out - (mux_X_out + Post_Adder_CIN)) : (mux_Z_out + mux_X_out +
Post_Adder_CIN);

//Gray_colored_multiplexer of P
//output P
assign P = (PREG) ? P_REG : Post_Adder_out;

//output PCOUT
assign PCOUT = P;

//Gray colored multiplexer of CYO Cascade

```

```

//output CARRYOUT
assign CARRYOUT = (CARRYOUTREG) ? CYO_REG : Post_Adder_CO;

//output CARRYOUTF
assign CARRYOUTF = CARRYOUT;

endmodule

```

b) Ref

```

module Spartan6_DSP48A1_REF
(
    input [17:0] A,B,D,BCIN,
    input [47:0] C,PCIN,
    input clk,
    CARRYIN,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,CEA,CEB,CEM,
    CEP,CEC,CED,CECARRYIN,CEOPMODE,
    input [7:0] OPMODE,
    output[17:0] BCOUT,
    output[47:0] PCOUT,P,
    output[35:0] M,
    output 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";
parameter RSTTYPE = "SYNC" ;

reg [17:0] A0_reg,B0_reg,A1_reg,B1_reg,D_reg;
wire[17:0]
pre_addORsub_out,mux_A0,mux_A1,mux_B0,mux_B1,mux_D,B0_input,B1_input;
reg [47:0] C_reg,P_reg;
wire[47:0]post_addORsub_out ,mux_C,output_MUX_Z , output_MUX_X;
wire[48:0] mux_P;
wire[35:0] mux_M,M_buff,mux_carryin,mux_carryout;
reg [7:0] OPMODE_reg;
wire[7:0] mux_OPMODE;
reg CARRYIN_reg,CYO_reg;
wire ADDER_CYO,CIN;
reg [35:0] M_reg;
wire[35:0] MULTP_out ;
generate
    if (RSTTYPE=="SYNC")
        always @ (posedge clk)begin
            // for input A
            if (RSTA) {A0_reg,A1_reg} <=0;

```

```

    else if (CEA) {A0_reg,A1_reg} <= {A,mux_A0};
    // for input B
    if (RSTB) {B0_reg,B1_reg} <=0;
    else if (CEB) {B0_reg,B1_reg} <= {B0_input,B1_input};
    // for input C
    if (RSTC) C_reg <=0;
    else if (CEC) C_reg <= C;
    // for input D
    if (RSTD) D_reg <=0;
    else if (CED) D_reg <= D;
    // for input opmode
    if (RSTOPMODE) OPMODE_reg <=0;
    else if (CEOPMODE) OPMODE_reg <= OPMODE;
    // for input carryin and carry out
    if (RSTCARRYIN) {CARRYIN_reg,CYO_reg} <=0;
    else if (CECARRYIN) {CARRYIN_reg,CYO_reg} <= {mux_carryin,
ADDER_CYO};

    // for M REG
    if (RSTM) M_reg <=0;
    else if (CEM) M_reg <= MULTP_out;
    // for P REG
    if (RSTP) P_reg <=0;
    else if (CEP) P_reg <= post_addORsub_out;
    end

else if (RSTTYPE=="ASYNC") begin
always @(posedge clk or posedge RSTA)
    // for input A
    if (RSTA) {A0_reg,A1_reg} <=0;
    else if (CEA) {A0_reg,A1_reg} <= {A,mux_A0};
always @(posedge clk or posedge RSTB)
    // for input B
    if (RSTB) {B0_reg,B1_reg} <=0;
    else if (CEB) {B0_reg,B1_reg} <= {B0_input,B1_input};
always @(posedge clk or posedge RSTC)
    // for input C
    if (RSTC) C_reg <=0;
    else if (CEC) C_reg <= C;
always @(posedge clk or posedge RSTD)
    // for input D
    if (RSTD) D_reg <=0;
    else if (CED) D_reg <= D;
always @(posedge clk or posedge RSTOPMODE)
    // for input opmode
    if (RSTOPMODE) OPMODE_reg <=0;
    else if (CEOPMODE) OPMODE_reg <= OPMODE;
always @(posedge clk or posedge RSTCARRYIN)
    // for input carryin and carry out
    if (RSTCARRYIN) {CARRYIN_reg,CYO_reg} <=0;
    else if (CECARRYIN) {CARRYIN_reg,CYO_reg} <= {mux_carryin,
ADDER_CYO};

always @(posedge clk or posedge RSTM)
    // for M REG
    if (RSTM) M_reg <=0;
    else if (CEM) M_reg <= MULTP_out;
always @(posedge clk or posedge RSTP)
    if (RSTP) P_reg <=0;
    else if (CEP) P_reg <= post_addORsub_out;
    end

```

```

endgenerate
//pre-adder / subtractor
assign pre_addORsub_out = (mux_OPMODE[6])? (mux_D - mux_B0) : (mux_D + mux_B0);
//post-adder/subtractor
assign {ADDER_CYO,post_addORsub_out} = (mux_OPMODE[7])? (output_MUX_Z - (output_MUX_X + CIN)) : (output_MUX_X + output_MUX_Z + CIN);
//multipler
assign MULTP_out = mux_A1 * mux_B1;
//MUX
assign mux_D = (DREG)? D_reg : D;
assign B0_input = (B_INPUT=="DIRECT")? B : BCIN;
assign mux_B0= (B0REG)? B0_reg : B0_input;
assign mux_B1= (B1REG)? B1_reg:B1_input;
assign mux_A0= (A0REG)? A0_reg : A;
assign mux_A1= (A1REG)? A1_reg:mux_A0;
assign mux_C = (CREG)? C_reg : C;
assign mux_OPMODE = (OPMODEREG)? OPMODE_reg : OPMODE;
assign B1_input = (mux_OPMODE[4])? pre_addORsub_out : mux_B0 ;
assign mux_D = (DREG)? D_reg : D;
assign mux_carryin =(CARRYINSEL=="OPMODE5")? mux_OPMODE[5]:CARRYIN;
assign CIN =(CARRYINREG)? CARRYIN_reg : mux_carryin;
assign mux_carryout =(CARRYOUTREG)? CYO_reg:ADDER_CYO;
assign mux_M = (MREG)? M_reg : MULTP_out;
assign output_MUX_X = (mux_OPMODE[1:0]==2'b00)? 0 : (mux_OPMODE[1:0]==2'b01)? mux_M : (mux_OPMODE[1:0]==2'b10)? P : {mux_D,mux_A1,mux_B1};
assign output_MUX_Z = (mux_OPMODE[3:2]==2'b00)? 0 : (mux_OPMODE[3:2]==2'b01)? PCIN : (mux_OPMODE[3:2]==2'b10)? P : mux_C;
assign mux_P = (PREG)? P_reg:post_addORsub_out;
assign M_buff = mux_M;
//outputs
assign P = mux_P;
assign PCOUT =P ;
assign BCOUT = mux_B1 ;
assign M = M_buff;
assign CARRYOUT = mux_carryout;
assign CARRYOUTF =CARRYOUT;
endmodule

```

## 2) TB CODE

```
module tb_Spartan6_DSP48A1();
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";
parameter RSTTYPE = "SYNC" ;
integer i=0;
reg [17:0] A,B,D,BCIN;
reg [47:0] C,PCIN;
reg clk,
CARRYIN,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE,CEA,CEB,CEM,
CEP,CEC,CED,CECARRYIN,CEOPMODE;
reg [7:0] OPMODE;
wire[17:0] BCOUT_1,BCOUT_2;
wire [47:0] PCOUT_1,PCOUT_2,P_1,P_2;
wire [35:0] M_1,M_2;
wire CARRYOUT_1 , CARRYOUTF_1,CARRYOUT_2 , CARRYOUTF_2;
Spartan6_DSP48A1_REF#(A0REG,A1REG,B0REG,B1REG,CREG,DREG,MREG,PREG,CARRY
INREG,CARRYOUTREG,OPMODEREG,CARRYINSEL,B_INPUT,RSTTYPE)
DUT(A,B,D,BCIN,C,PCIN,clk,CARRYIN,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARR
YIN,RSTOPMODE,CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEOPMODE, OPMODE,
BCOUT_1,PCOUT_1,P_1,M_1,CARRYOUT_1 , CARRYOUTF_1);
Spartan6_DSP48A1_DUT#(A0REG,A1REG,B0REG,B1REG,CREG,DREG,MREG,PREG,CARRY
INREG,CARRYOUTREG,OPMODEREG,CARRYINSEL,B_INPUT,RSTTYPE)
DUT2 (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_2, PCOUT_2, P_2, M_2, CARRYOUT_2,
CARRYOUTF_2);
initial begin
//initial values
clk=1;CARRYIN=0;CEA=1;CEB=1;CEM=1;CEP=1;CEC=1;CED=1;CECARRYIN=1;CEOPMOD
E=1;
A=1;B=2;D=3;C=4;PCIN=5;BCIN=6;OPMODE=8'b0000_0000;{RSTA,RSTB,RSTM,RSTP,
RSTC,RSTD,RSTCARRYIN,RSTOPMODE}=8'b1111_1111;
forever
#5 clk=~clk;
end
initial begin
#20;
for (OPMODE=0;OPMODE<=8'b1111_1111;OPMODE=OPMODE+1) begin
repeat(5)
@(negedge clk)
{RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE}=8'b0000_0000;
A=$urandom_range(5,15); B=$urandom_range(5,15);
C=$urandom_range(5,15); D=$urandom_range(5,15);
BCIN=$urandom_range(0,5);

```

```

if (BCOUT_2==BCOUT_1 && P_2==P_1 && PCOUT_2==PCOUT_1 && M_2==M_1
&&CARRYOUT_2==CARRYOUT_1 && CARRYOUTF_2==CARRYOUTF_1)
$display ("iteraion %d is correct",OPMODE);
else begin
$display ("Error in iteration %d",OPMODE);
$display ("BCOUT_1=%d, P_1=%d,
PCOUT_1=%d,M_1=%d,CARRYOUT_1=%d,CARRYOUTF_1=%d",BCOUT_1,P_1,PCOUT_1,M_1
,CARRYOUT_1,CARRYOUTF_1);
$display ("BCOUT_2=%d, P_2=%d,
PCOUT_2=%d,M_2=%d,CARRYOUT_2=%d,CARRYOUTF_2=%d",BCOUT_2,P_2,PCOUT_2,M_2
,CARRYOUT_2,CARRYOUTF_2);
$stop;
end
if(OPMODE==8'b1101_1111)
{RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTCARRYIN,RSTOPMODE}=8'b1111_1111;
if(OPMODE==8'b1111_1110)
{CEA,CEB,CEC,CECARRYIN,CECARRYIN,CED,CEM,CEOPMODE}=8'b0000_0000;
if (OPMODE==8'b1111_1111)
$stop;
end
$stop;
end
endmodule

```

### 3) DO FILE

```

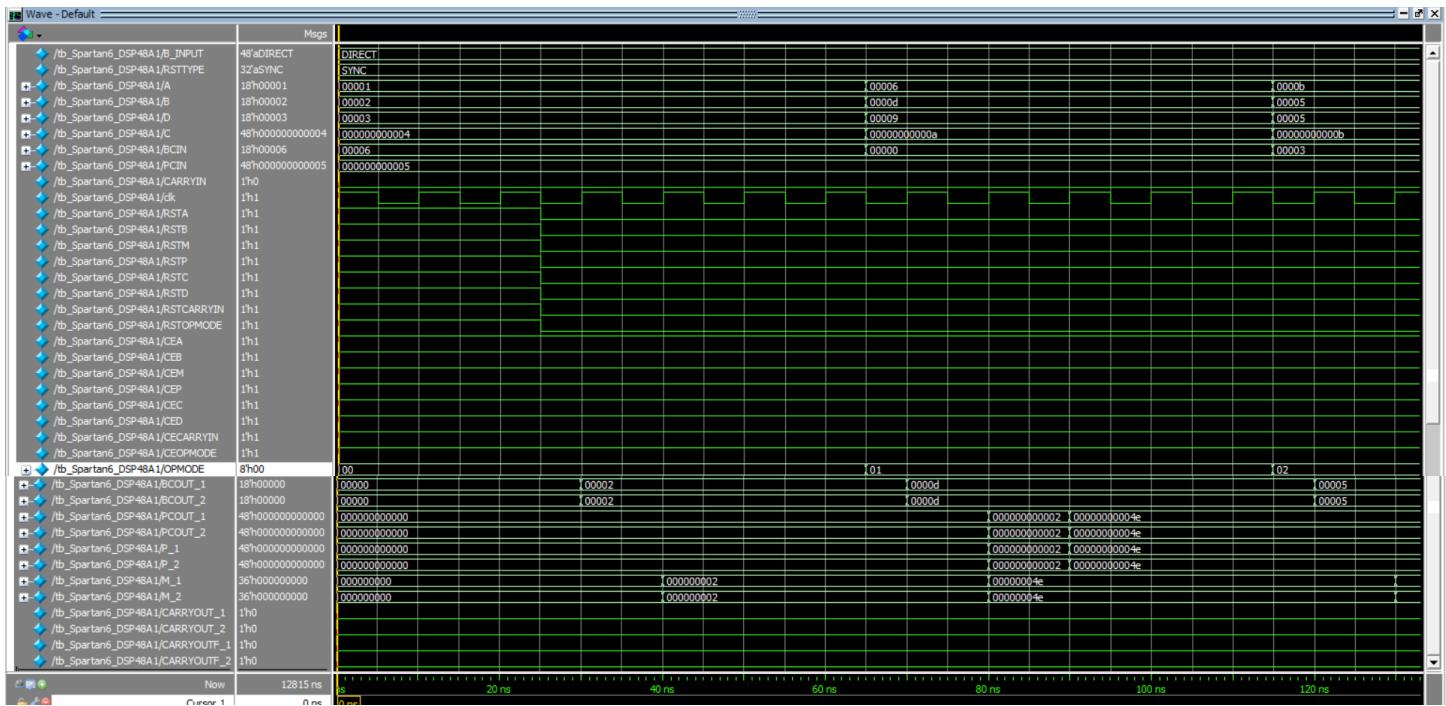
vlib work
vlog Spartan6_DSP48A1_REF.v Spartan6_DSP48A1_DUT.v testbench.v
vsim -voptargs=+acc work.tb_Spartan6_DSP48A1
add wave *
run -all

```

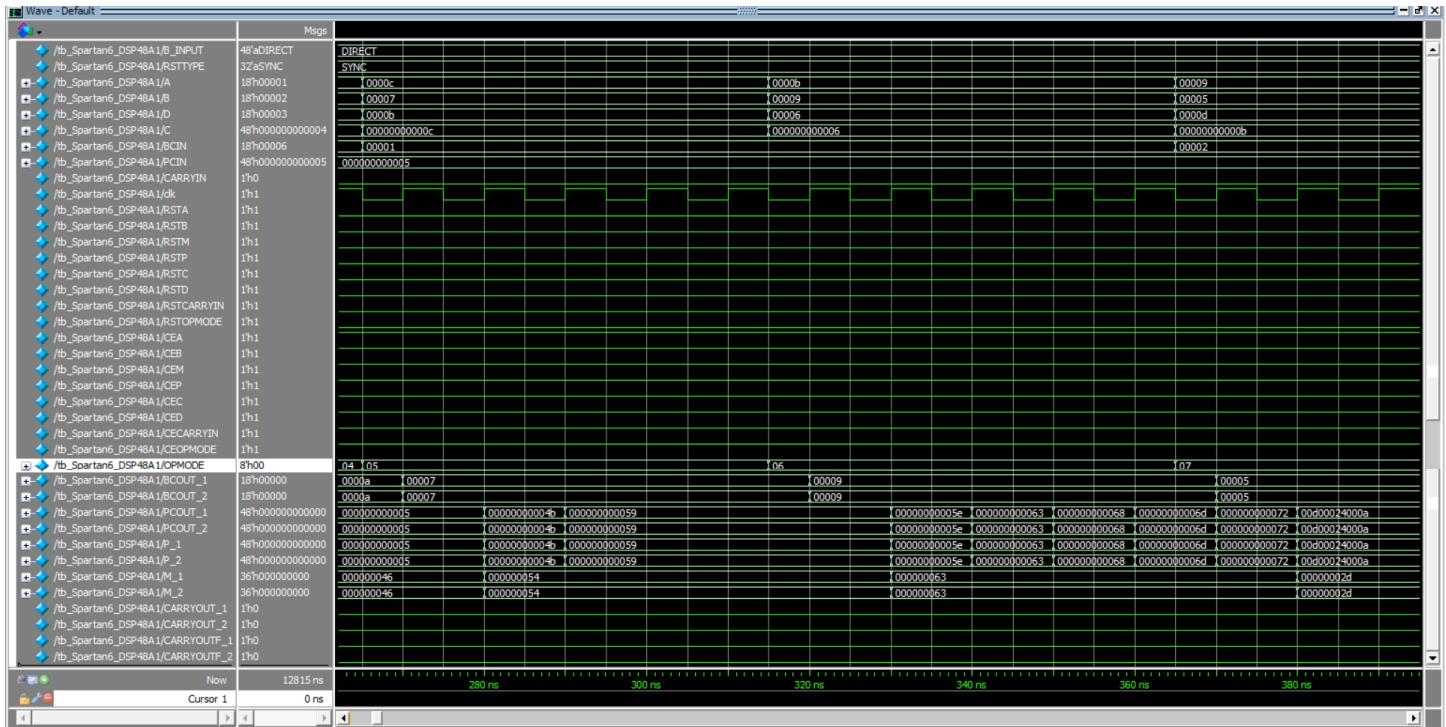
## 4) WAVE FORM

⇒ CARRYINSEL="OPMODE5", BINPUT="DIRECT", RSTTYPE="SYNC"

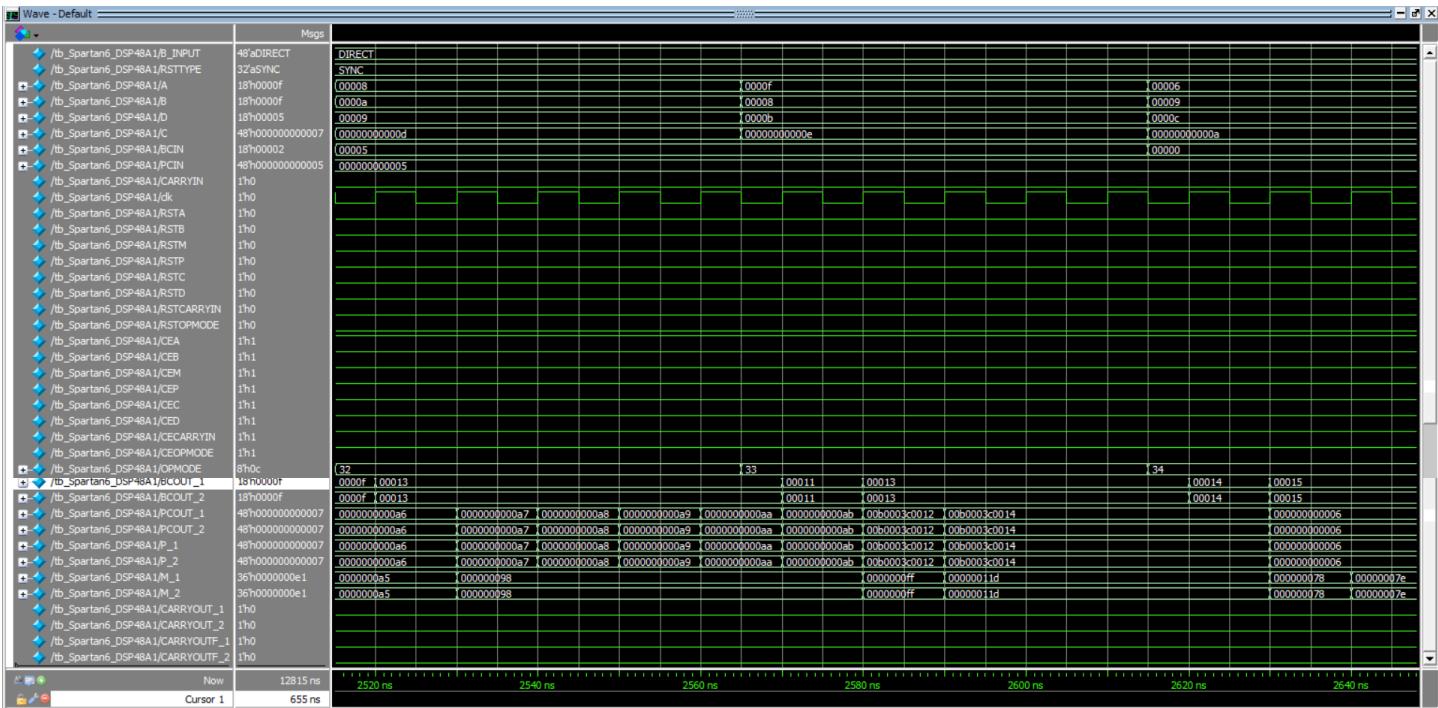
1-



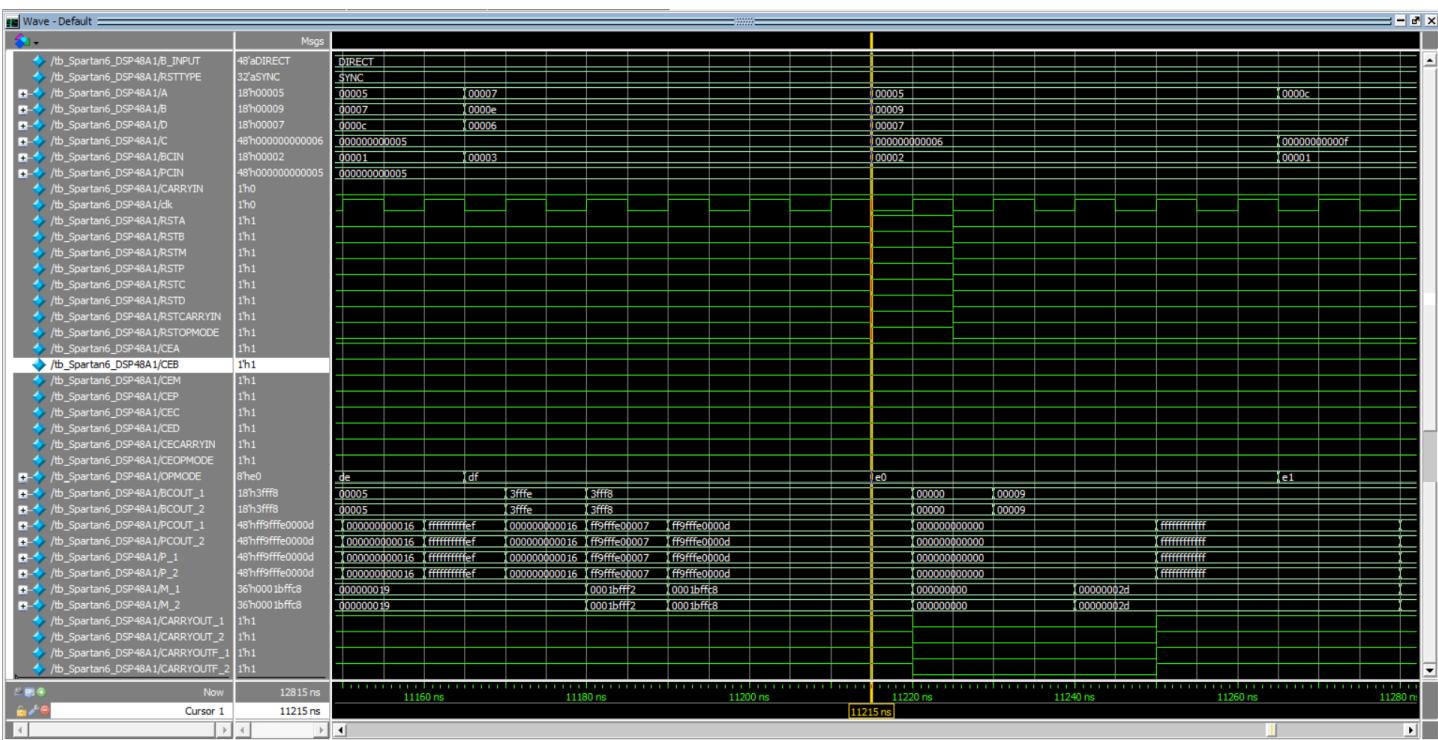
2-



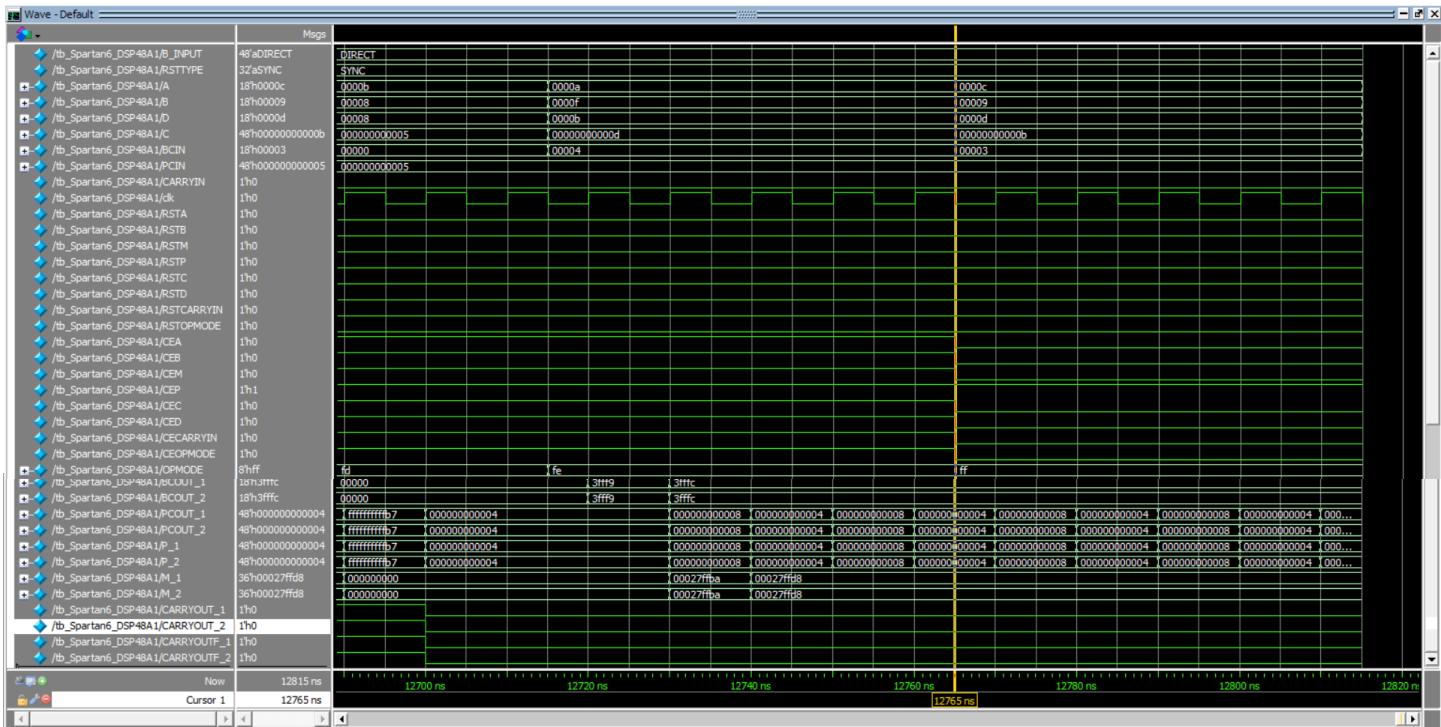
3-



4-

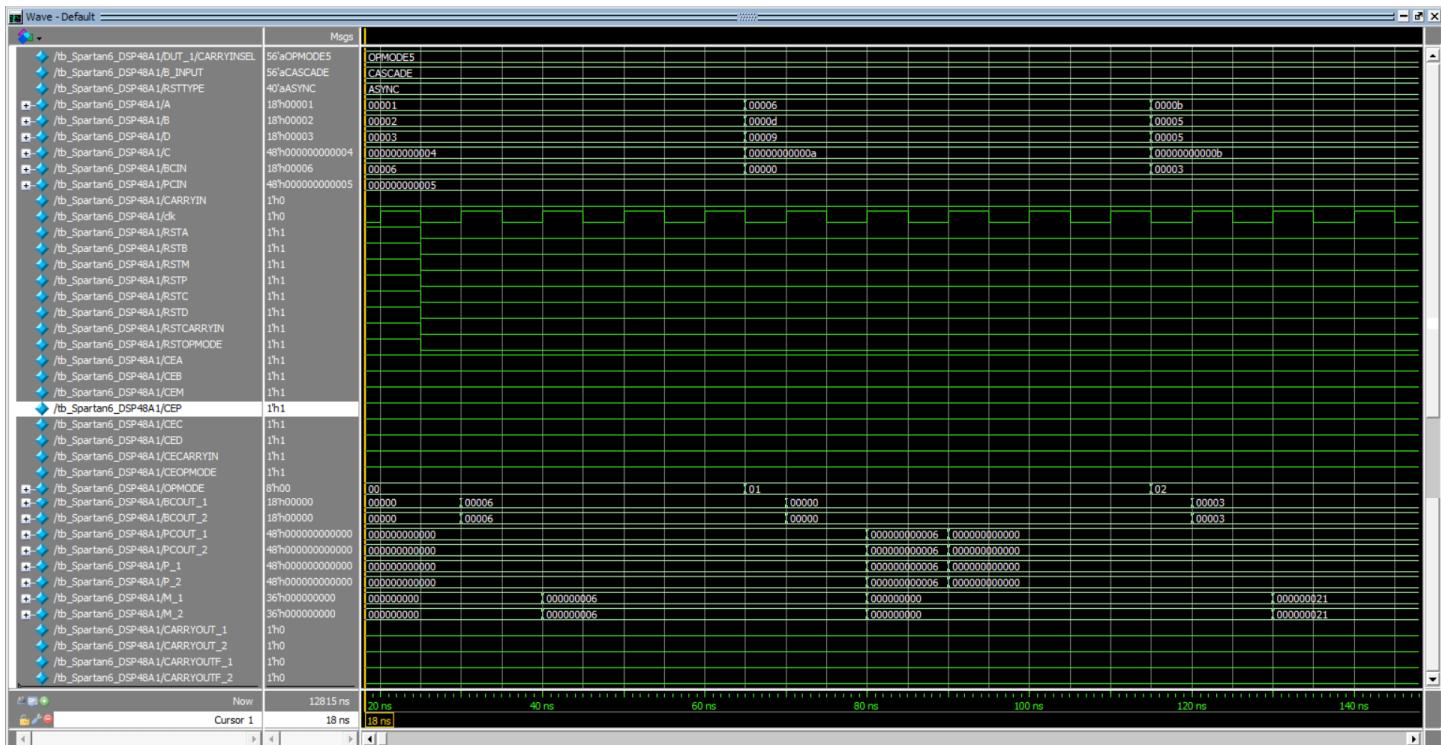


5-

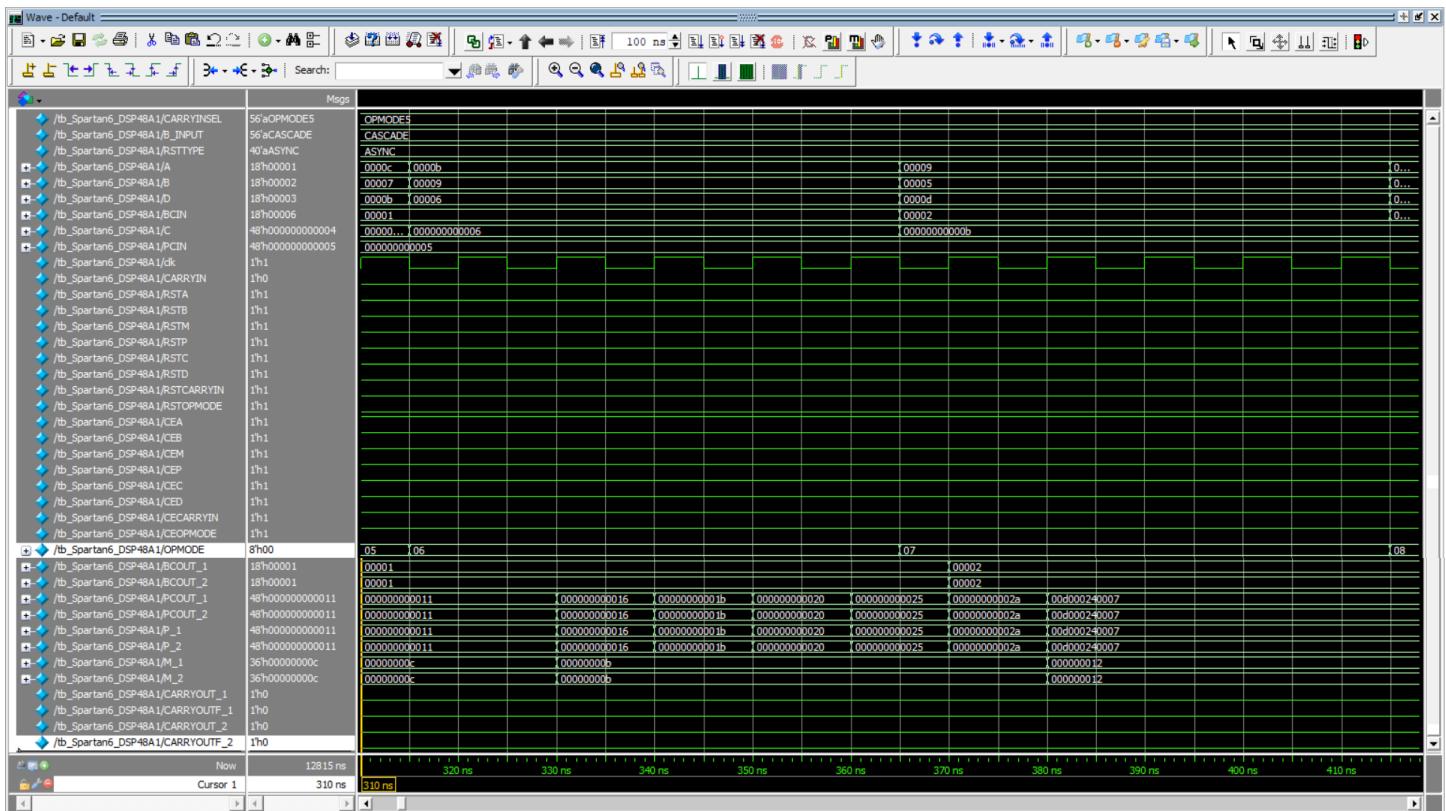


⇒ CARRYINSEL="OPMODE5", BINPUT="CASCADE", RSTTYPE="ASYNC"

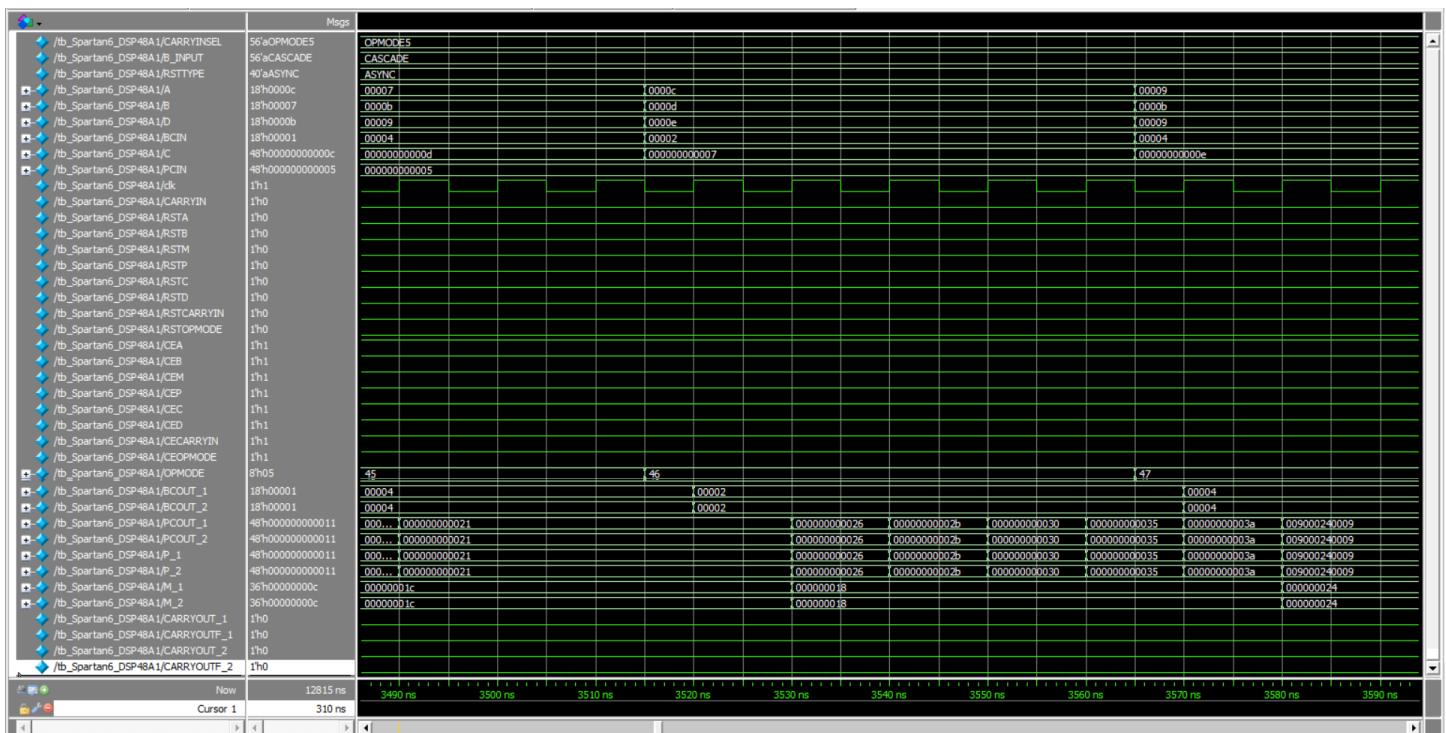
1-

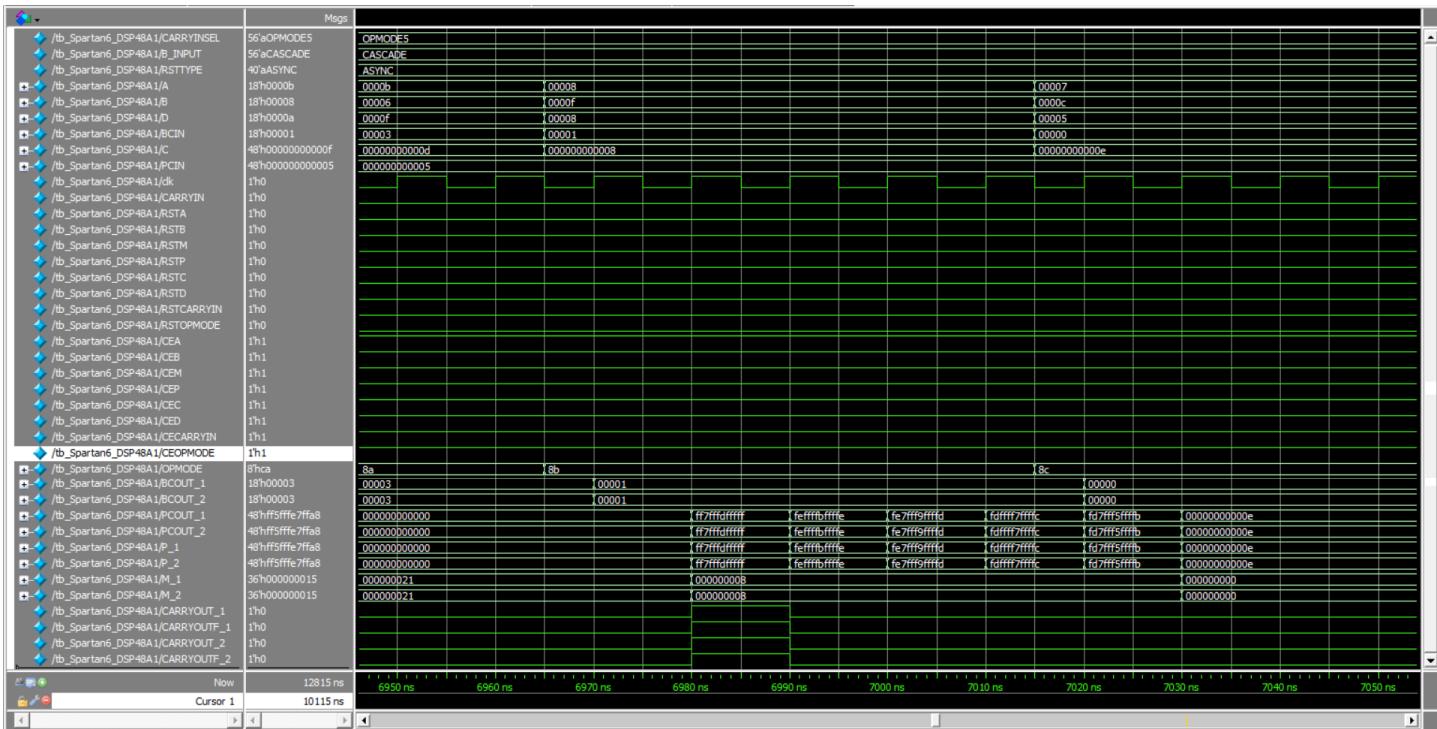


2-

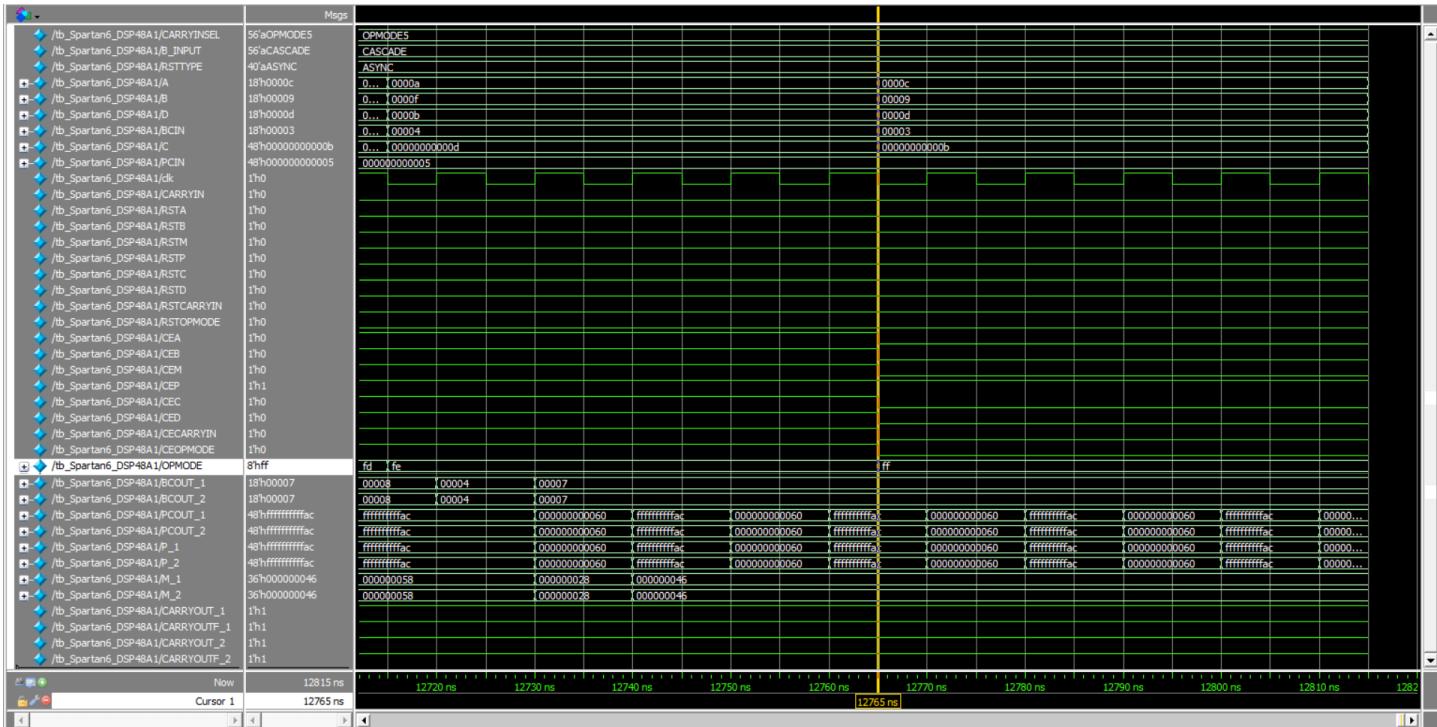


3-



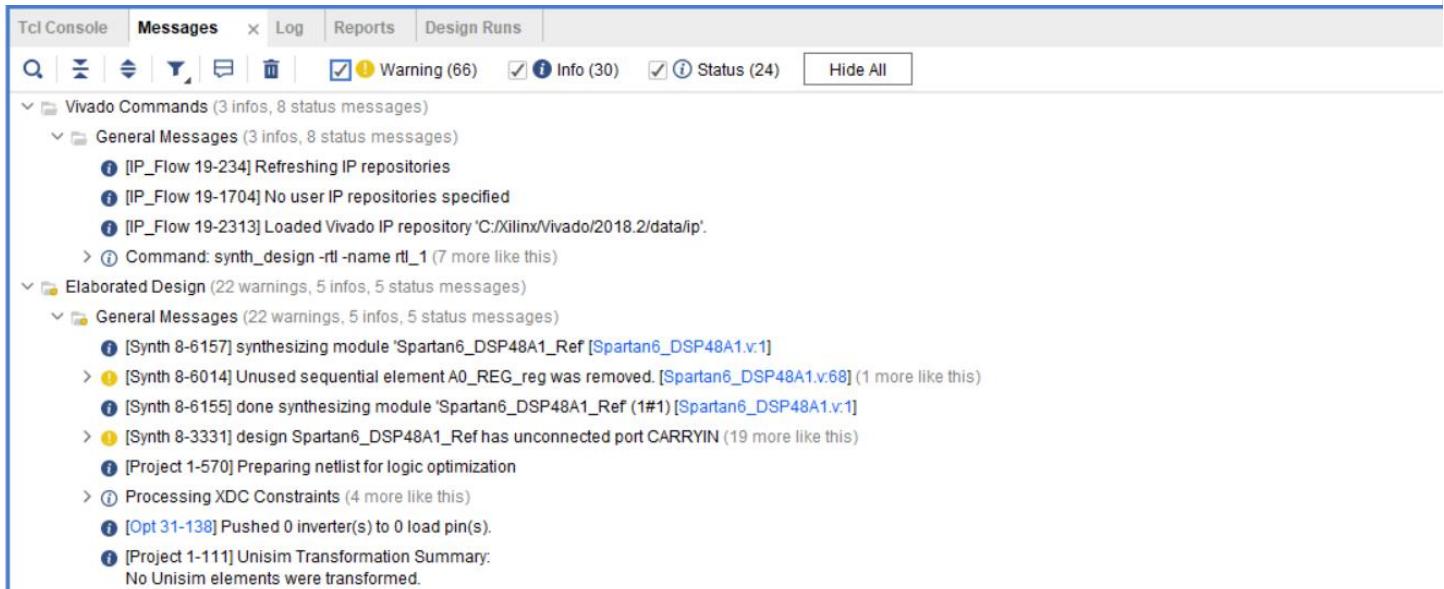
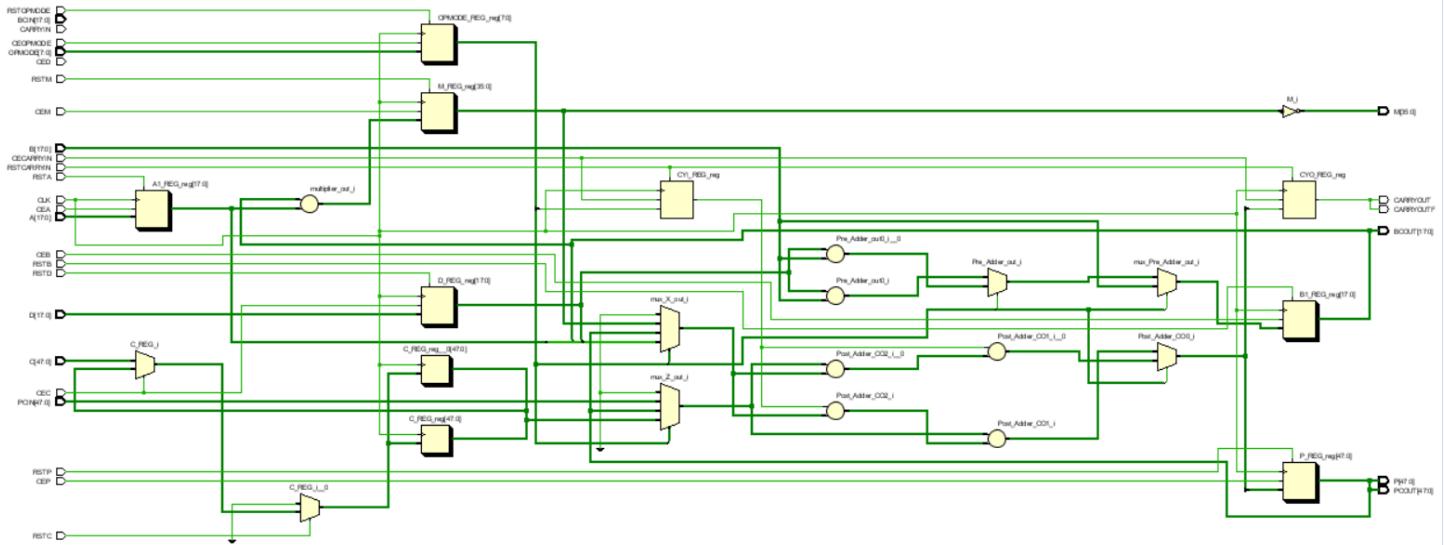


5-



## 5) ELABORATION

→ Design



## 6) SYNTHESIS

### → Design

↳ Synthesis (45 warnings, 67 infos, 6 status messages)

- > ⚠ Command: synth\_design -top Spartan6\_DSP48A1\_Ref -part xc7a200tffg1156-1 (5 more like this)
- ⠄ [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a200t'
- ⠄ [Synth 8-6157] synthesizing module 'Spartan6\_DSP48A1\_Ref [Spartan6\_DSP48A1.v:1]
- > ⚠ [Synth 8-6014] Unused sequential element A0\_REG\_reg was removed. [Spartan6\_DSP48A1.v:68] (3 more like this)
- ⠄ [Synth 8-6155] done synthesizing module 'Spartan6\_DSP48A1\_Ref (1#1) [Spartan6\_DSP48A1.v:1]
- > ⚠ [Synth 8-3331] design Spartan6\_DSP48A1\_Ref has unconnected port CARRYIN (39 more like this)
- ⠄ [Device 21-403] Loading part xc7a200tffg1156-1
- > ⚠ [Synth 8-5818] HDL ADVISOR - The operator resource <adder> is shared. To prevent sharing consider applying a KEEP on the output of the operator [Spartan6\_DSP48A1.v:273] (1 more like this)
- > ⚠ [Synth 8-4471] merging register 'A1\_REG\_reg[17:0]' into 'A1\_REG\_reg[17:0]' [Spartan6\_DSP48A1.v:69] (1 more like this)
- > ⚠ [Synth 8-4765] Removing register instance '(C\_REG\_reg[47]\_0)' from module (Spartan6\_DSP48A1\_Ref) as it is equivalent to (C\_REG\_reg[47]) and driving same net [Spartan6\_DSP48A1.v:90] (47 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 [Spartan6\_DSP48A1.v:69]
- ⠄ [Project 1-571] Translating synthesized netlist
- ⠄ [Netlist 29-17] Analyzing 206 Unisim elements for replacement
- ⠄ [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
- ⠄ [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.
- ⠄ [Common 17-83] Releasing license: Synthesis
- ⠄ [Constraints 18-5210] No constraint will be written out.
- ⠄ [Common 17-1381] The checkpoint 'D:/materials/diploma/project 1/project\_1/runs/synth\_1/Spartan6\_DSP48A1\_Ref.dcp' has been generated.
- ⠄ [runrtl-4] Executing : report\_utilization -file Spartan6\_DSP48A1\_Ref\_utilization\_synth.rpt -pb Spartan6\_DSP48A1\_Ref\_utilization\_synth.pb
- ⠄ [Common 17-206] Exiting Vivado at Fri Aug 18 23:53:32 2023...

↳ Synthesized Design (6 infos)

↳ General Messages (6 infos)

- ⠄ [Netlist 29-17] Analyzing 206 Unisim elements for replacement
- ⠄ [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
- ⠄ [Project 1-479] Netlist was created with Vivado 2018.2
- ⠄ [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.

Tcl Console	Messages	Log	Reports	Design Runs	Utilization	x	Debug
					Hierarchy		

↳ Hierarchy

↳ Summary

↳ Slice Logic

↳ Slice LUTs (<1%)

↳ LUT as Logic (<1%)

Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
Spartan6_DSP48A1_Ref	230	160	1	326	1

Tcl Console	Messages	Log	Reports	Design Runs	Utilization	x	Debug
					Summary		

↳ 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

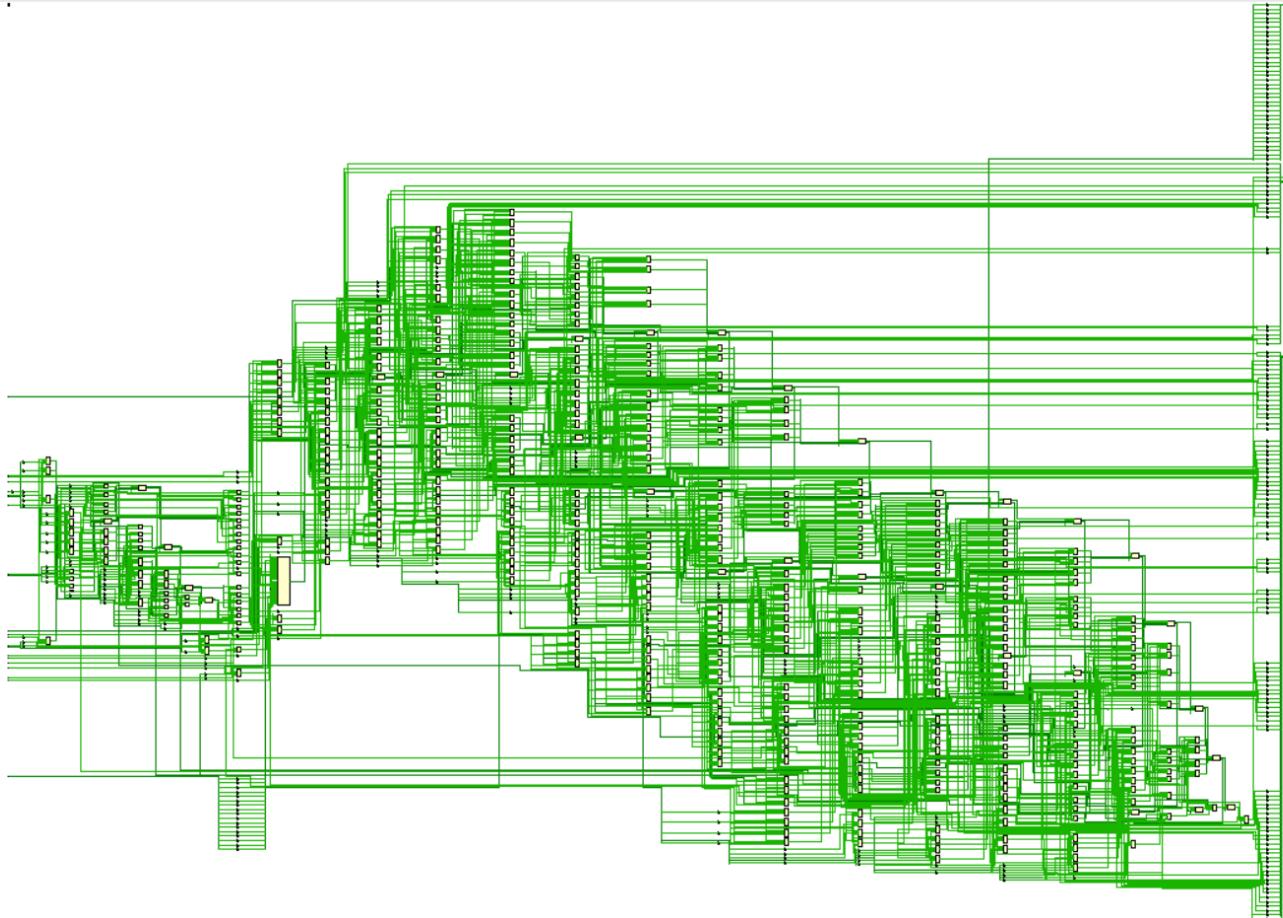
    Bonded IOB (65%)

↳ Clocking

Resource	Utilization	Available	Utilization %
LUT	230	134600	0.17
FF	160	269200	0.06
DSP	1	740	0.14
IO	326	500	65.20

↳ Utilization (%)

Utilization (%)



## 7) IMPLEMENTATION

### → Design

- ✓ Implementation (6 warnings, 73 infos, 178 status messages)
- ✓ Design Initialization (7 infos, 3 status messages)
  - > ⓘ Command: open\_checkpoint {D:/materials/diploma/project\_1/project\_1/runs/impl\_1/Spartan6\_DSP48A1\_Ref.dcp} (2 more like this)
    - ⓘ [Netlist 29-17] Analyzing 206 Unisim elements for replacement
    - ⓘ [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - ⓘ [Project 1-479] Netlist was created with Vivado 2018.2
    - ⓘ [Device 21-403] Loading part xc7a200tffg1156-1
    - ⓘ [Project 1-570] Preparing netlist for logic optimization
    - ⓘ [Project 1-111] Unisim Transformation Summary:  
No Unisim elements were transformed.
    - ⓘ [Project 1-604] Checkpoint was created with Vivado v2018.2 (64-bit) build 2258646
- ✓ Opt Design (1 warning, 23 infos, 41 status messages)
  - > ⓘ Command: opt\_design (40 more like this)
    - ⓘ [Common 17-349] Got license for feature 'Implementation' and/or device 'xc7a200t'
    - ⓘ [Project 1-461] DRC finished with 0 Errors
    - ⓘ [Project 1-462] Please refer to the DRC report (report\_drc) for more information.
    - ⓘ [Timing 38-35] Done setting XDC timing constraints.
    - ⓘ [Opt 31-49] Retargeted 0 cell(s).
    - > ⓘ [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s). (1 more like this)
    - > ⓘ [Opt 31-389] Phase Retarget created 0 cells and removed 0 cells (4 more like this)
      - ⓘ [Opt 31-662] Phase BUFG optimization created 0 cells of which 0 are BUFGs and removed 0 cells.
      - ⓘ [Pwropt 34-132] Skipping clock gating for clocks with a period < 2.00 ns.
    - ⓘ [Common 17-83] Releasing license: Implementation
    - ⓘ [Constraints 18-5210] No constraint will be written out.
    - ⓘ [Common 17-1381] The checkpoint 'D:/materials/diploma/project\_1/project\_1/runs/impl\_1/Spartan6\_DSP48A1\_Ref\_opt.dcp' has been generated.
    - ⓘ [runrtl-4] Executing : report\_drc-file Spartan6\_DSP48A1\_Ref\_drc\_opted.rpt -pb Spartan6\_DSP48A1\_Ref\_drc\_opted.pb -rpx Spartan6\_DSP48A1\_Ref\_drc\_opted.rpx
    - ⓘ [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'.
  - > ⓘ [DRC 23-27] Running DRC with 2 threads (1 more like this)
    - ⓘ [Corertl 2-168] The results of DRC are in file [Spartan6\\_DSP48A1\\_Ref\\_drc\\_opted.rpt](#).
- ✓ Place Design (2 warnings, 14 infos, 75 status messages)
  - > ⓘ Command: place\_design (74 more like this)
    - ⓘ [Common 17-349] Got license for feature 'Implementation' and/or device 'xc7a200t'
- ✓ Route Design (3 warnings, 29 infos, 59 status messages)
  - > ⓘ Command: route\_design (58 more like this)
    - ⓘ [Common 17-349] Got license for feature 'Implementation' and/or device 'xc7a200t'
    - ⓘ [Vivado\_Td 4-198] DRC finished with 0 Errors
    - ⓘ [Vivado\_Td 4-199] Please refer to the DRC report (report\_drc) for more information.
    - ⓘ [Route 35-254] Multithreading enabled for route\_design using a maximum of 2 CPUs
    - ⓘ [Route 35-64] No timing constraints were detected. The router will operate in resource-optimization mode.
    - ⓘ [Route 35-16] Router Completed Successfully
    - ⓘ [Common 17-83] Releasing license: Implementation
    - ⓘ [Constraints 18-5210] No constraint will be written out.
    - ⓘ [Common 17-1381] The checkpoint 'D:/materials/diploma/project\_1/project\_1/runs/impl\_1/Spartan6\_DSP48A1\_Ref\_routed.dcp' has been generated.
  - > ⓘ [DRC 23-27] Running DRC with 2 threads (1 more like this)
    - ⓘ [Corertl 2-168] The results of DRC are in file [Spartan6\\_DSP48A1\\_Ref\\_drc\\_routed.rpt](#).
  - > ⓘ [runrtl-4] Executing : report\_drc-file Spartan6\_DSP48A1\_Ref\_drc\_routed.rpt -pb Spartan6\_DSP48A1\_Ref\_drc\_routed.pb -rpx Spartan6\_DSP48A1\_Ref\_drc\_routed.rpx (7 more like this)
    - ⓘ [Timing 38-35] Done setting XDC timing constraints. (2 more like this)
    - ⓘ [DRC 23-133] Running Methodology with 2 threads
    - ⓘ [Corertl 2-1520] The results of Report Methodology are in file [Spartan6\\_DSP48A1\\_Ref\\_methodology\\_drc\\_routed.rpt](#).
    - ⓘ [Power 33-232] No user defined clocks were found in the design!  
Resolution: Please specify clocks using create\_clock/create\_generated\_clock for sequential elements. For pure combinatorial circuits, please specify a virtual clock, otherwise the vectorless estimation might be inaccurate
    - ⓘ [Timing 38-313] There are no user specified timing constraints. Timing constraints are needed for proper timing analysis.
    - ⓘ [Vivado\_Td 4-545] No incremental reuse to report, no incremental placement and routing data was found.
  - > ⓘ [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min\_max, Timing Stage: Requireds. (1 more like this)
  - > ⓘ [Timing 38-191] Multithreading enabled for timing update using a maximum of 2 CPUs (1 more like this)
- ✓ Implemented Design (1 warning, 5 infos, 4 status messages)
  - ✓ General Messages (1 warning, 5 infos, 4 status messages)
    - ⓘ [Netlist 29-17] Analyzing 206 Unisim elements for replacement
    - ⓘ [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - ⓘ [Project 1-479] Netlist was created with Vivado 2018.2
    - ⓘ [Project 1-570] Preparing netlist for logic optimization
  - > ⓘ Reading XDEF placement. (3 more like this)
    - ⓘ [Project 1-111] Unisim Transformation Summary:  
No Unisim elements were transformed.
    - ⓘ [Timing 38-313] There are no user specified timing constraints. Timing constraints are needed for proper timing analysis.

- Device



- 
- schematic

