Ahmed Alaa Mohamed Ain shams

Mini Project

Verilog code:

```
module project (A,B,C,BCIN,D,CARRYIN,M,P,CARRYOUT,CARRYOUTF,CLK,OPMODE,CEA,CEB,CECARRYIN,CEC,CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RST
parameter A0REG = 0 ;
parameter A1REG = 1 ;
parameter B0REG = 0 ;
parameter B1REG = 1 ;
parameter CREG=0, DREG=0, MREG=0,PREG=0, CARRYINREG=0, CARRYOUTREG=0,OPMODEREG = 0 ;
parameter CARRYINSEL = "OPMODE5";
parameter B_INPUT = "DIRECT" ;
parameter RSTTYPE = "SYNC" ;
input [17:0] A,B,D,BCIN ;
input [47:0] C,PCIN;
input [7:0] OPMODE ;
input CARRYIN,CLK,CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP ;
input RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP ;
output [35:0] M ;
output [47:0] P ;
output CARRYOUTF,CARRYOUT ;
output [17:0] BCOUT ;
output [47:0] PCOUT ;
wire [17:0] W1,W22,W3,W4,W5,W6,W10,W2;
wire [47:0] W7,CON,MMUX,W15;
reg [47:0] W11,W14 ;
wire [35:0] W8,W9;
wire[1:0] OP,OP2 ;
wire W12,W13,W16,W17 ;
assign W17 = (RSTTYPE=="SYNC") ?  1 : 0 ; //we should sent W12 to any muxreg instantiation when we need to change from sync to asuncrous but know as
muxreg m1(D,CLK,DREG,RSTD,CED,W1) ;
muxreg m2(B,CLK,B0REG,RSTB,CEB,W22) ;
assign W2 = (B_INPUT == "DIRECT") ? W22 : (B_INPUT == "CASCADE") ? BCIN : 0 ;
ADDER A1(W1,W2,OPMODE[6],W3) ;
assign W10 = (OPMODE[4]==1) ?   W3:W2 ;
muxreg m3(W10,CLK,B1REG,RSTB,CEB,W4) ;
assign BCOUT = W4 ;
muxreg m4(A,CLK,A0REG,RSTA,CEA,W5) ;
muxreg m5(W5,CLK,A1REG,RSTA,CEA,W6) ;
muxreg #(.size(48))m6(C,CLK,CREG,RSTC,CEC,W7) ;
assign W8 = W4*W6 ;
muxreg #(.size(36))m7(W8,CLK,MREG,RSTM,CEM,W9) ;
assign M = W9 ;
assign CON = {D[11:0], A[17:0], B[17:0]};
assign MMUX = {12'b000000000000,M} ;
assign OP = OPMODE[1:0] ;
always @(*) begin
    case (OP)
       2'b00 : W11 = 0 ;
       2'b01 : W11 = MMUX ;
       2'b10 : W11 = P ;
       2'b11 : W11 = CON ;

    endcase
end
assign W12 = (CARRYINSEL=="OPMODE5") ? OPMODE[5] : (CARRYINSEL=="CARRYIN") ? CARRYIN : 0 ;
muxreg #(.size(1))m8(W12,CLK,CARRYINREG,RSTCARRYIN,CECARRYIN,W13) ;
assign OP2 = OPMODE[3:2] ;
always @(*) begin
    case (OP2)
    2'b00 : W14 = 0 ;
    2'b01 : W14 = PCIN ;
    2'b10 : W14 = P ;
    2'b11 : W14 = W7 ;
    endcase
end
post A2(W14,W11,W13,OPMODE[7],W15,W16) ;
muxreg #(.size(48))m9(W15,CLK,PREG,RSTP,CEP,P) ;
assign PCOUT = P ;
muxreg #(.size(1))m10(W16,CLK,CARRYOUTREG,RSTCARRYIN,CECARRYIN,CARRYOUT) ;
assign CARRYOUTF = CARRYOUT ;


endmodule
```

Pre Adder module:

```
module ADDER (in1,in2,op,out);
parameter siz = 18 ;
input [siz-1:0] in1,in2 ;
input op ;
output  reg [siz-1: 0] out ;
always @(*) begin
    if(op)
    out = in1-in2 ;
    else
    out = in1+in2 ;
end

endmodule
```

Post adder module:

```
module post (in1,in2,cin,opcode,out,cout);
input [47:0] in1,in2 ;
input cin,opcode ;
output reg [47:0] out ;
output reg cout ;
always @(*) begin
    if (opcode) begin
        out = in1 -(in2+cin) ;
    end
    else begin
        {cout,out} = in1 + in2 + cin ;
    end
end

endmodule
```

Mux register module:

```
module muxreg (in,clk,select,reset,clk_en,q);
    parameter size = 18 ;
    parameter RSTYPE = 1 ;
    input select,reset,clk_en ;
    input clk ;
    input [size-1:0] in ;
    output  [size-1:0] q ;
    reg [size-1:0] wire1 ;
   generate
  if(RSTYPE)begin

    always @(posedge clk) begin
       if (reset)
          wire1=0 ;

        else
         if(clk_en)
            wire1 = in ;

    end
    assign q = (select==1) ? wire1: in ;
   end
   else begin
      always @(posedge clk or posedge reset) begin
       if (reset)
          wire1=0 ;

        else
         if(clk_en)
            wire1 = in ;

    end
    assign q = (select==1) ? wire1: in ;
   end
  endgenerate
```

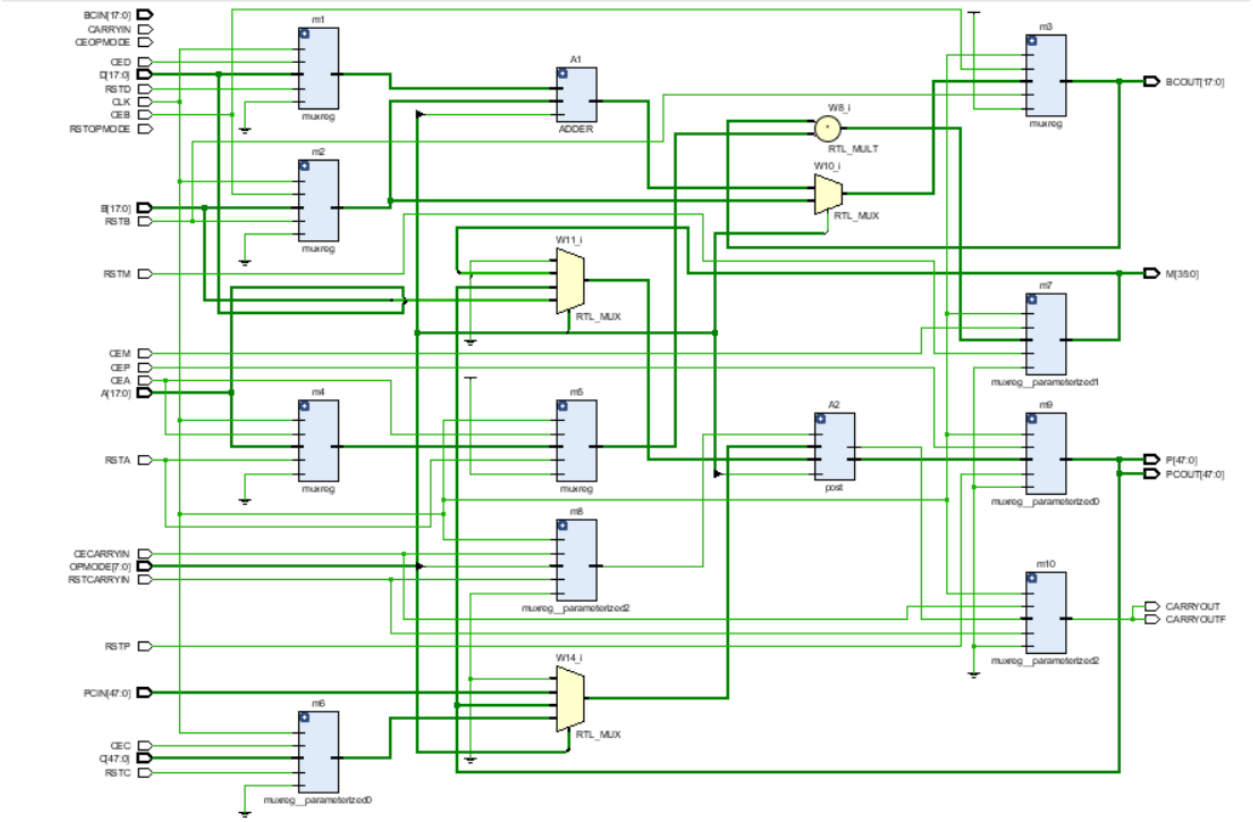
Testbench:

```
module project_tb ();
reg [17:0] A_tb,B_tb,D_tb,BCIN_tb ;
reg [47:0] C_tb,PCIN_tb ;
reg  [7:0] OPMODE_tb ;
reg CARRYIN_tb,CLK,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  [35:0] M_tb ;
wire [47:0] P_tb ;
wire CARRYOUTF_tb,CARRYOUT_tb ;
wire [17:0] BCOUT_tb ;
wire [47:0] PCOUT_tb ;
project DUT(A_tb,B_tb,C_tb,BCIN_tb,D_tb,CARRYIN_tb,M_tb,P_tb,CARRYOUT_tb,CARRYOUTF_tb,CLK,OPMODE_tb,CEA_tb,CEB_tb,CECARRYIN_tb,CEC_tb,CED_tb,CEM_tb,
initial begin
    CLK = 0 ;
    forever begin
       #10 CLK = ~CLK ;
    end
end
initial begin
    RSTA_tb = 1 ;
    RSTB_tb = 1 ;
    RSTC_tb = 1 ;
    RSTCARRYIN_tb = 1;
    RSTD_tb = 1 ;
    RSTM_tb = 1 ;
    RSTOPMODE_tb = 1 ;
    RSTP_tb = 1 ;
    CARRYIN_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 ;
    A_tb = 5 ;
    B_tb = 10 ;
    C_tb = 1 ;
    D_tb = 7 ;
    BCIN_tb = 0 ;
    PCIN_tb = 0 ;
    OPMODE_tb = 8'b00011110 ;
    repeat(10) @(negedge CLK)    ;
    RSTA_tb = 0 ;
    RSTB_tb = 0 ;
    RSTC_tb = 0 ;
    RSTCARRYIN_tb = 0;
    RSTD_tb = 0 ;
    RSTM_tb = 0 ;
    RSTOPMODE_tb = 0 ;
    RSTP_tb = 0 ;
    CARRYIN_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 ;
    A_tb = 5 ;
    B_tb = 10 ;
    C_tb = 1 ;
    D_tb = 7 ;
    BCIN_tb = 0 ;
    PCIN_tb = 0 ;
    OPMODE_tb = 8'b00011101 ;
    repeat(10) @(negedge CLK)    ;
    RSTA_tb = 0 ;
    RSTB_tb = 0 ;
```

```
71      RSTB_tb = 0 ;
72      RSTC_tb = 0 ;
73      RSTCARRYIN_tb = 0;
74      RSTD_tb = 0 ;
75      RSTM_tb = 0 ;
76      RSTOPMODE_tb = 0 ;
77      RSTP_tb = 0 ;
78      CARRYIN_tb = 0;
79      CEA_tb = 1 ;
80      CEB_tb = 1 ;
81      CEC_tb = 1 ;
82      CECARRYIN_tb = 1 ;
83      CED_tb = 1 ;
84      CEM_tb = 1 ;
85      CEOPMODE_tb = 1 ;
86      CEP_tb = 1 ;
87      A_tb = 11 ;
88      B_tb = 30 ;
89      C_tb = 87 ;
90      D_tb = 9 ;
91      BCIN_tb = 5 ;
92      PCIN_tb = 6 ;
93      OPMODE_tb = 8'b00011101 ;
94      repeat(10) @(negedge CLK)    ;
95        RSTA_tb = 0 ;
96      RSTB_tb = 0 ;
97      RSTC_tb = 0 ;
98      RSTCARRYIN_tb = 0;
99      RSTD_tb = 0 ;
100     RSTM_tb = 0 ;
101     RSTOPMODE_tb = 0 ;
102     RSTP_tb = 0 ;
103     CARRYIN_tb = 1;
104     CEA_tb = 1 ;
105     CEB_tb = 1 ;
106     CEC_tb = 1 ;
107     CECARRYIN_tb = 1 ;
108     CED_tb = 1 ;
109     CEM_tb = 1 ;
110     CEOPMODE_tb = 1 ;
111     CEP_tb = 1 ;
112     A_tb = 5 ;
113     B_tb = 10 ;
```

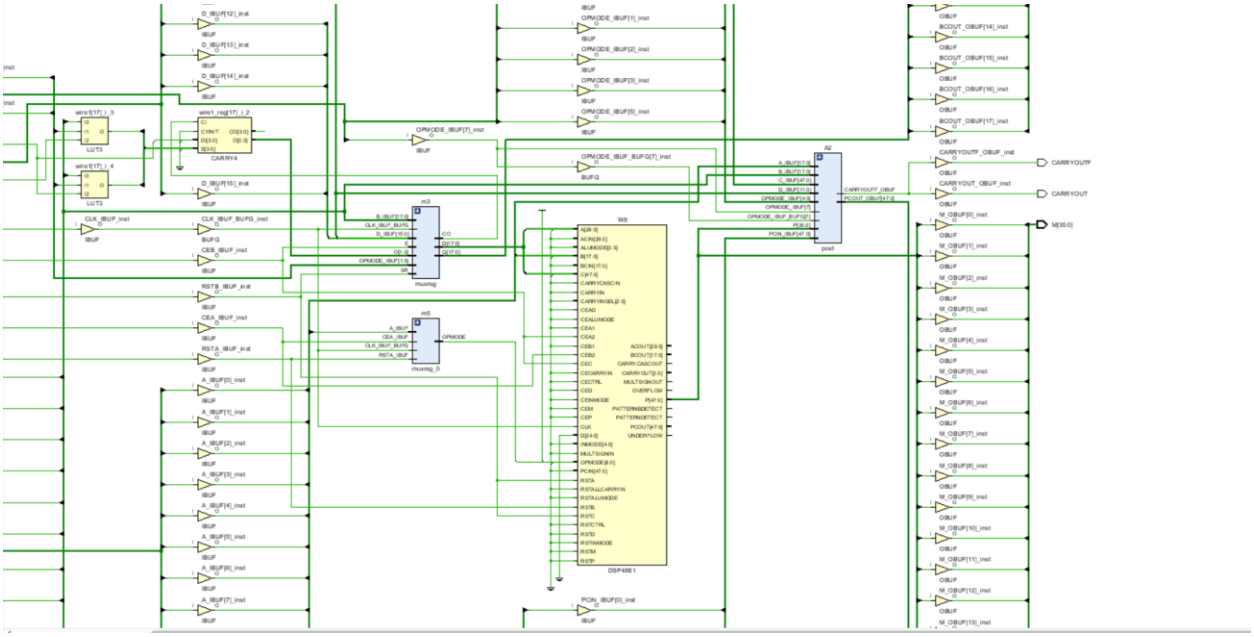
```
114     C_tb = 1 ;
115     D_tb = 7 ;
116     BCIN_tb = 0 ;
117     PCIN_tb = 0 ;
118     OPMODE_tb = 8'b00011111 ;
119     repeat(10) @(negedge CLK)    ;
120       RSTA_tb = 0 ;
121     RSTB_tb = 0 ;
122     RSTC_tb = 0 ;
123     RSTCARRYIN_tb = 0;
124     RSTD_tb = 0 ;
125     RSTM_tb = 0 ;
126     RSTOPMODE_tb = 0 ;
127     RSTP_tb = 0 ;
128     CARRYIN_tb = 1;
129     CEA_tb = 1 ;
130     CEB_tb = 1 ;
131     CEC_tb = 1 ;
132     CECARRYIN_tb = 1 ;
133     CED_tb = 1 ;
134     CEM_tb = 1 ;
135     CEOPMODE_tb = 1 ;
136     CEP_tb = 1 ;
137     A_tb = 5 ;
138     B_tb = 10 ;
139     C_tb = 1 ;
140     D_tb = 7 ;
141     BCIN_tb = 0 ;
142     PCIN_tb = 0 ;
143     OPMODE_tb = 8'b00010101 ;
144     repeat(10) @(negedge CLK)    ;
145     $stop ;
146 end
147 endmodule
148
```

RTL:

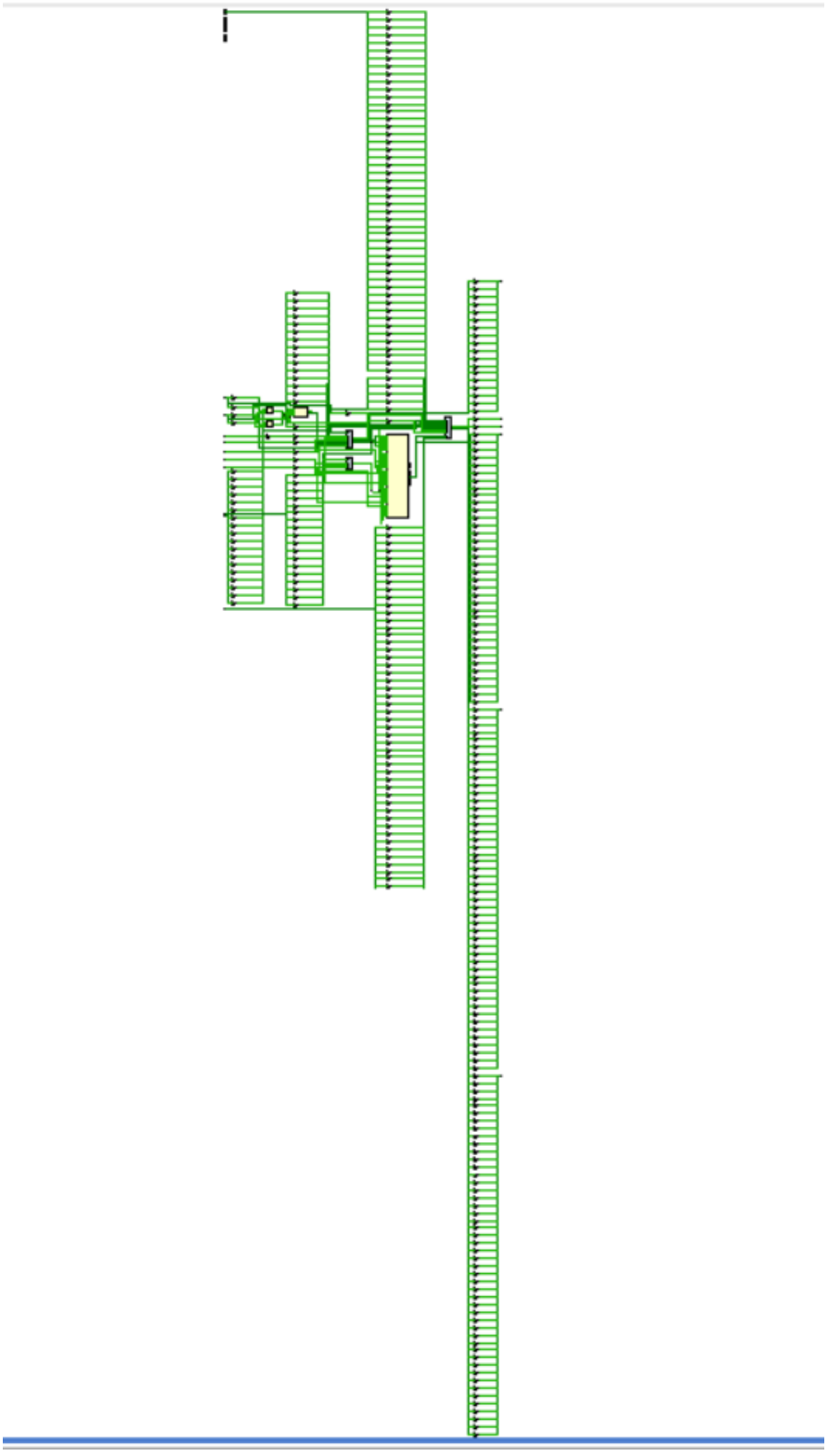

BCIN[17:0]
CARRYIN
CEOPMODE
CED
D[17:0]
RSTD
CLK
CEB
RSTOPMODE
B[17:0]
RSTB
RSTM
CEM
CEP
CEA
A[17:0]
RSTA
CECARRYIN
OPMODE[7:0]
RSTCARRYIN
RSTP
PCIN[47:0]
CEC
C[47:0]
RSTC
m1
muxreg
m2
muxreg
m3
muxreg
m4
muxreg
m5
muxreg
m6
muxreg__parameterized0
m7
muxreg__parameterized1
m8
muxreg__parameterized2
m9
muxreg__parameterized0
m10
muxreg__parameterized2
A1
ADDER
A2
post
W8_i
RTL_MULT
W10_i
RTL_MUX
W11_i
RTL_MUX
W14_i
RTL_MUX
BCOUT[17:0]
M[35:0]
P[47:0]
PCOUT[47:0]
CARRYOUT
CARRYOUTF

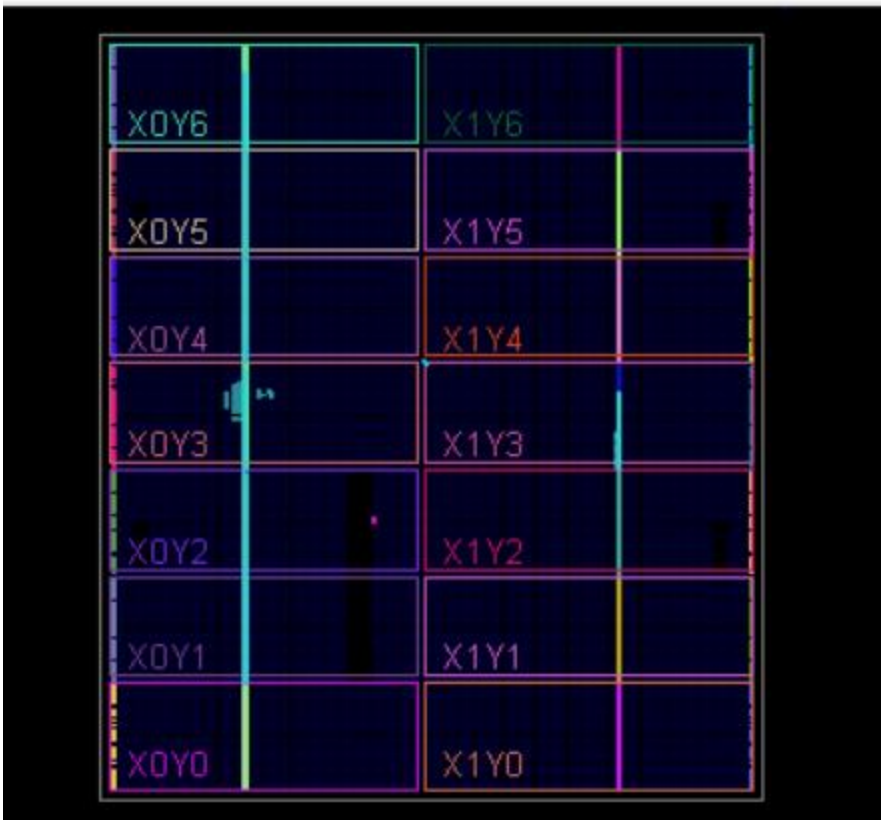
Ahmed Alaa Mohamed Ain shams

Schematic



Ahmed Alaa Mohamed Ain shams





Utilization report:

Reports | Design Runs | Power | DRC | Methodology | Timing | Utilization

Hierarchy

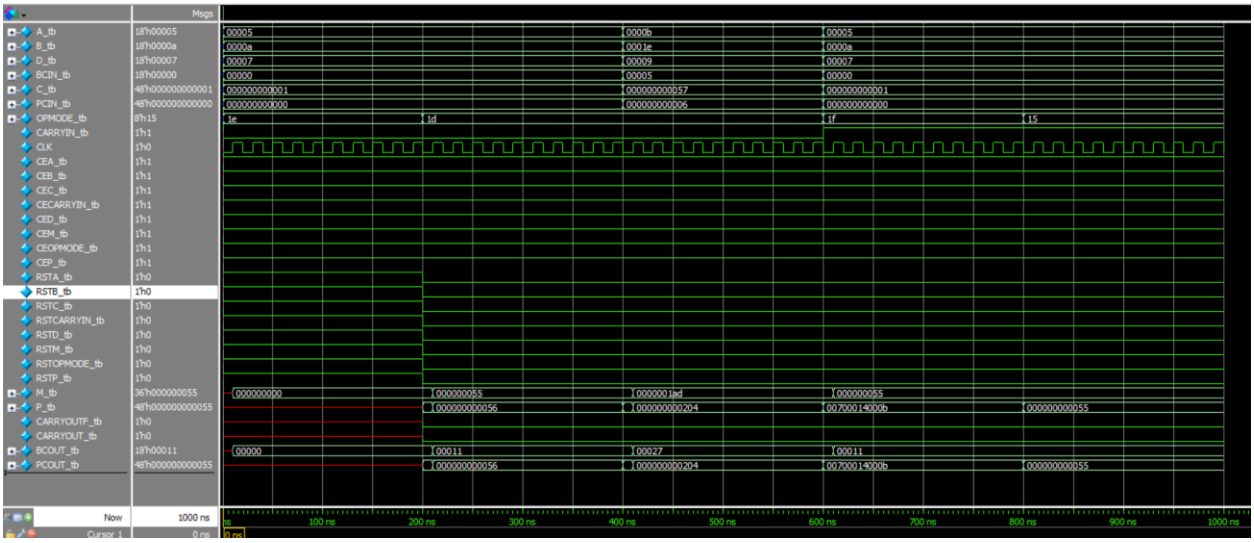
Name	Slice LUTs (303600)	Slice Registers (607200)	Slice (75900)	LUT as Logic (303600)	DSPs (2800)	Bonded IOB (600)	BUFGCTRL (32)
project	340	20	95	340	1	315	2
A2 (post)	305	1	79	305	0	0	0
m3 (muxreg)	33	18	15	33	0	0	0
m5 (muxreg_0)	0	1	1	0	0	0	0

Timing report:

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	inf	Worst Hold Slack (WHS):	inf	Worst Pulse Width Slack (WPWS):	4.600 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	269	Total Number of Endpoints:	269	Total Number of Endpoints:	20

All user specified timing constraints are met.

Waveform:



Clock Summary

Name	Waveform	Period (ns)	Frequency (MHz)
sys_clk_pin	{0.000 5.000}	10.000	100.000