Digital Design using Verilog

by Karim Wassem

Project (1) Spartan6 - DSP48A1

Name: Shahd Bassem Mohamed Helmy

Group: 1

- **(SEC.1)** Verilog Code for the design

Spartan6.v > Spartan6

```v
module Spartan6(A,B,D,C,clk,CARRYIN,opmode,BCIN,RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTopmode,RSTCARRYIN,
CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEopmode,PCIN,BCOUT,P,PCOUT,M,CARRYOUT,CARRYOUTF);

parameter A0REG=0;          // For piplining
parameter A1REG=1;          // For piplining
parameter B0REG=0;          // For piplining
parameter B1REG=1;          // For piplining
parameter CREG=1;           // For piplining
parameter DREG=1;           // For piplining
parameter MREG=1;           // For piplining
parameter PREG=1;           // For piplining
parameter CARRYINREG=1;     // For piplining
parameter CARRYOUTREG=1;    // For piplining
parameter OPMODEREG=1;      // For piplining
parameter CARRYINSEL="OPMODE5";     // For Carryin selection
parameter B_INPUT="DIRECT";         // Direct for B and cascade for BCIN
parameter RSTTYPE="SYNC";           // If resets could be sync or async

input  [17:0] A,B,D;     // Inputs
input  [47:0] C;         // Inputs
input  [17:0] BCIN;      // Dedicated BCIN
input   clk;
input   CARRYIN;
input  [7:0] opmode;
input   RSTA,RSTB,RSTM,RSTP,RSTC,RSTD,RSTopmode,RSTCARRYIN;   // Resets
input   CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEopmode;          // Clock enables
input  [47:0] PCIN;
output [17:0] BCOUT;
output [47:0] P;
output [47:0] PCOUT;
output [35:0] M;        // Multiplier data output
output  CARRYOUT;
output  CARRYOUTF;

//OUTPUTS OF MUXS
wire [17:0] D_pipe, B0_pipe, B1_pipe, A0_pipe, A1_pipe, opmode_pipe;
wire [47:0] C_pipe;
wire [17:0] B1_INPUT;           ///output of MUX after the pre adder
reg  [47:0] X_out, Z_out;        ///outputs of MUXs X and Z
//OUTPUTS OF FFs
reg  [47:0] C_reg, P_reg ;
reg  [17:0] D_reg, B0_reg, B1_reg, A0_reg, A1_reg;
reg  [7:0]  opmode_reg;
reg  [35:0] M_reg;
reg   CYI_reg,CYO_reg;          ///OUTPUT OF CYI FF AND CYO FF

reg  [17:0] preAdder_result;     ///OUTPUT of PRE-ADDER/SUBTRACTOR
reg  [35:0] mult_out;            ///OUTPUT of MULTIPLIER
reg  [47:0] concatenated;        ///D:A:B CONCATENATED
wire [47:0] postAdder_result;    ///OUTPUT of POST-ADDER/SUBTRACTOR
wire  carryout;                  ///CARRYOUT of POST-ADDER/SUBTRACTOR
wire  CIN;                       ///CARRYIN for POST-ADDER/SUBTRACTOR

wire [17:0] B_in;
assign B_in = (B_INPUT=="CASCADE") ? BCIN : B;

wire carryin_input;
assign  carryin_input = (CARRYINSEL=="CARRYIN") ? CARRYIN : opmode_pipe[5];
```

```
60  // Generate block for reset logic
61  generate
62      if(RSTTYPE=="ASYNC")begin
63              always @(posedge clk or posedge RSTD) begin
64                  if(RSTD)
65                      D_reg <= 18'b0;
66                  else if(CED)
67                      D_reg <= D;
68              end
69
70              always @(posedge clk or posedge RSTB) begin
71                  if(RSTB)
72                  B0_reg <= 18'b0;
73                  else if (CEB)
74                  B0_reg <= B_in;
75              end
76
77              always @(posedge clk or posedge RSTB) begin
78                  if(RSTB)
79                  B1_reg <= 18'b0;
80                  else if (CEB)
81                  B1_reg <= B1_INPUT;
82              end
83
84              always @(posedge clk or posedge RSTC) begin
85                  if(RSTC)
86                      C_reg <= 48'b0;
87                  else if (CEC)
88                      C_reg <= C;
89              end
90
91              always @(posedge clk or posedge RSTA) begin
92                  if(RSTA)
93                      A0_reg <= 18'b0;
94                  else if(CEA)
95                      A0_reg<= A;
```

```
96          end
97
98          always @(posedge clk or posedge RSTA) begin
99              if(RSTA)
100                 A1_reg <= 18'b0;
101             else if(CEA)
102                 A1_reg <= A0_pipe;
103         end
104
105         always @(posedge clk or posedge RSTopmode) begin
106             if(RSTopmode)
107                 opmode_reg <= 8'b0;
108             else if (CEopmode)
109                 opmode_reg <= opmode;
110         end
111
112         always @(posedge clk or posedge RSTM) begin
113         if(RSTM)
114             M_reg <= 36'b0;
115         else if (CEM)
116             M_reg <= mult_out;
117         end
118
119         always @(posedge clk or posedge RSTCARRYIN) begin
120             if(RSTCARRYIN)
121             CYI_reg <= 1'b0;
122             else if (CECARRYIN)
123             CYI_reg <= carryin_input;
124         end
125
126         always @(posedge clk or posedge RSTP) begin
127             if(RSTP)
128                 P_reg <= 48'b0;
129             else if(CEP)
130                 P_reg <= postAdder_result;
131         end
```

```
132
133             always @(posedge clk or posedge RSTCARRYIN) begin
134                 if(RSTCARRYIN)
135                 CYO_reg <= 1'b0;
136                 else if (CECARRYIN)
137                 CYO_reg <= carryout;
138             end
139         end
140
141         else begin  ///synchronous rst
142          always @(posedge clk) begin
143                 //D
144                 if(RSTD)
145                     D_reg <= 18'b0;
146                 else if(CED)
147                     D_reg <= D;
148                 //B0
149                 if(RSTB)
150                     B0_reg<=18'b0;
151                 else if (CEB)
152                     B0_reg<=B_in;
153                 //B1
154                 if(RSTB)
155                     B1_reg<=18'b0;
156                 else if (CEB)
157                     B1_reg<=B1_INPUT;
158                 //C
159                 if(RSTC)
160                     C_reg <= 48'b0;
161                 else if (CEC)
162                     C_reg <= C;
163                 //A0
164                 if(RSTA)
165                     A0_reg <= 18'b0;
166                 else if(CEA)
167                     A0_reg<= A;
```

```
168             //A1
169             if(RSTA)
170                 A1_reg <= 18'b0;
171             else if(CEA)
172                 A1_reg <= A0_pipe;
173             //OPMODE
174             if(RSTopmode)
175                 opmode_reg <= 8'b0;
176             else if (CEopmode)
177                 opmode_reg <= opmode;
178             ///M    (multiplier output)
179             if(RSTM)
180                 M_reg <= 36'b0;
181             else if (CEM)
182                 M_reg <= mult_out;
183             ///CARRY INPUT CASCADE
184             if(RSTCARRYIN)
185                 CYI_reg <= 1'b0;
186             else if (CECARRYIN)
187                 CYI_reg <= carryin_input;
188             ///P OUTPUT
189             if(RSTP)
190                 P_reg <= 48'b0;
191             else if(CEP)
192                 P_reg <= postAdder_result;
193             ///CARRY OUTPUT CASCADE
194             if(RSTCARRYIN)
195                 CYO_reg <= 1'b0;
196             else if (CECARRYIN)
197                 CYO_reg <= carryout;
198         end
199     end
200     endgenerate
```

```
201
202     always @(*) begin
203         preAdder_result = (opmode_pipe[6])? D_pipe - B0_pipe : D_pipe + B0_pipe;
204         mult_out        = B1_pipe * A1_pipe;
205         concatenated    = {D_pipe[11:0],A1_pipe,B1_pipe};
206
207         case (opmode_pipe[1:0])
208         2'b00: X_out = 48'b0;
209         2'b01: X_out = {12'b0,M};
210         2'b10: X_out = PCOUT;
211         2'b11: X_out = concatenated;
212             default: X_out = 48'b0;
213         endcase
214
215         case (opmode_pipe[3:2])
216         2'b00: Z_out = 48'b0;
217         2'b01: Z_out = PCIN;
218         2'b10: Z_out = PCOUT;
219         2'b11: Z_out = C_pipe;
220             default: Z_out = 48'b0;
221         endcase
222     end
223
224             assign B1_INPUT    = (opmode_pipe[4]) ? preAdder_result : B0_pipe;
225
226             assign D_pipe      = (DREG)  ? D_reg  : D;
227             assign B0_pipe     = (B0REG) ? B0_reg : B_in;
228             assign B1_pipe     = (B1REG) ? B1_reg : B1_INPUT;
229             assign BCOUT       = B1_pipe;
230             assign C_pipe      = (CREG)  ? C_reg  : C;
231             assign A0_pipe     = (A0REG) ? A0_reg : A;
232             assign A1_pipe     = (A1REG) ? A1_reg : A0_pipe;
233             assign opmode_pipe = (OPMODEREG) ? opmode_reg : opmode;
234             assign M           = (MREG)  ? M_reg  : mult_out;
235             assign CIN         = (CARRYINREG) ? CYI_reg : carryin_input;
```

```
236             assign {carryout,postAdder_result} = (opmode_pipe[7])  ?
237                                                  Z_out-(X_out+CIN) :
238                                                  Z_out+X_out+CIN   ;
239
240             assign P           = (PREG)  ? P_reg  : postAdder_result;
241             assign CARRYOUT    = (CARRYINREG) ? CYO_reg : carryout;
242             assign CARRYOUTF   = CARRYOUT;
243             assign PCOUT       = P;
244
245 endmodule
```

- **(SEC.2)** Testbench Code for the design

Spartan6_tb.v > Spartan6_tb

```
1   module Spartan6_tb();
2
3   reg [17:0] A, B, D;
4   reg [47:0] C;
5   reg [17:0] BCIN;
6   reg clk;
7   reg CARRYIN;
8   reg [7:0] opmode;
9   reg RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTopmode, RSTCARRYIN;
10  reg CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEopmode;
11  reg [47:0] PCIN;
12
13  wire [17:0] BCOUT;
14  wire [47:0] P, PCOUT;
15  wire [35:0] M;
16  wire CARRYOUT, CARRYOUTF;
17
18  reg [47:0] prev_P;      //For PATH3
19
20      // Instantiate DUT
21      Spartan6  dut (A, B, D,C,clk,CARRYIN,opmode,BCIN,
22      RSTA, RSTB, RSTM,RSTP,RSTC,RSTD,RSTopmode,RSTCARRYIN,
23      CEA,CEB,CEM,CEP,CEC,CED,CECARRYIN,CEopmode,
24      PCIN,BCOUT,P,PCOUT,M,CARRYOUT,CARRYOUTF);
25
26      initial begin
27          clk = 0;
28          forever
29          #1 clk = ~clk;
30      end
31
32      initial begin
33          //Reset inputs
34          A = 0; B = 0; C = 0; D = 0;
35          BCIN = 0; CARRYIN = 0; PCIN = 0;
36          opmode = 0;
```

```
        // Activate Resets
        RSTA = 1; RSTB = 1; RSTM = 1; RSTP = 1;
        RSTC = 1; RSTD = 1; RSTopmode = 1; RSTCARRYIN = 1;

        CEA = 1; CEB = 1; CEM = 1; CEP = 1;
        CEC = 1; CED = 1; CECARRYIN = 1; CEopmode = 1;
        @(negedge clk);
        //inputs randomization
        A = $random;    B = $random;    C = $random;    D   = $random;
        opmode    = $random;  BCIN     = $random;
        CARRYIN   = $random;  PCIN     = $random;
        @(negedge clk);

        if(BCOUT!=0 || P!=0 || PCOUT != 0 || M!=0 || CARRYOUT!=0 || CARRYOUTF!=0) begin
            $display("Error");
            $stop;
        end
        else
            $display("Reset is done");
        //Deactivate Resets
        RSTA = 0; RSTB = 0; RSTM = 0; RSTP = 0;
        RSTC = 0; RSTD = 0; RSTopmode = 0; RSTCARRYIN = 0;
        repeat(4) @(negedge clk);

        //PATH1
       $display("Verify DSP Path 1");
       opmode = 8'b11011101;
       A = 20; B = 10; C = 350; D = 25;
       BCIN = $random; CARRYIN = $random; PCIN = $random;
        repeat(4) @(negedge clk);

         if (BCOUT !='hf ||M !='h12c ||P !='h32 ||PCOUT !='h32 ||CARRYOUT !=0 ||CARRYOUTF !=0) begin
                $display("Error .... Expected values: BCOUT=%h, M=%h, P=%h,PCOUT=%h, CO=%b, COF=%b",
                         'hf, 'h12c, 'h32,'h32, 0, 0);
                $stop;
                end
           else
                $display("Passed");
```

```
75          //PATH2
76          $display("Verify DSP Path 2");
77          opmode = 8'b00010000;
78          BCIN = $random; CARRYIN = $random; PCIN = $random;
79          repeat(3) @(negedge clk);
80          if (BCOUT !='h23 ||M !='h2bc ||P !=0 ||PCOUT !=0 ||CARRYOUT !=0 ||CARRYOUTF !=0) begin
81              $display("Error .... Expected values: BCOUT=%h, M=%h, P=%h,PCOUT=%h, CO=%b, COF=%b",
82                      'h23, 'h2bc, 0, 0, 0, 0);
83              $stop;
84          end
85          else
86              $display("Passed");
87
88          //PATH3
89          @(negedge clk) prev_P = P;
90
91          $display("Verify DSP Path 3");
92          opmode = 8'b00001010;
93          BCIN = $random; CARRYIN = $random; PCIN = $random;
94          repeat(3) @(negedge clk);
95          if (BCOUT !='ha ||M !='hc8 ||P !=prev_P ||PCOUT !=prev_P ||CARRYOUT !=0 || CARRYOUTF !=0) begin
96              $display("Error .... Expected values: BCOUT=%h, M=%h, P=%h,PCOUT=%h, CO=%b, COF=%b",
97                      'ha, 'hc8, prev_P, prev_P, 0, 0);
98              $stop;
99          end
100         else
101             $display("Passed");
102
103         //PATH4
104         $display("Verify DSP Path 4");
105         opmode = 8'b10100111;
106         A = 5; B = 6; C = 350; D = 25; PCIN = 3000;
107         BCIN = $random; CARRYIN = $random;
108         repeat(3) @(negedge clk);
109         if (BCOUT !='h6 ||M !='h1e || P !='hfe6fffec0bb1 ||PCOUT !='hfe6fffec0bb1 ||CARRYOUT !=1 ||CARRYOUTF !=1) begin
110             $display("Error .... Expected values: BCOUT=%h, M=%h, P=%h,PCOUT=%h, CO=%b, COF=%b",
111                     'h6, 'h1e, 'hfe6fffec0bb1, 'hfe6fffec0bb1, 1, 1);
112             $stop;
113         end
114         else
115             $display("Passed");
116
117     $stop;
118     end
119 endmodule
```

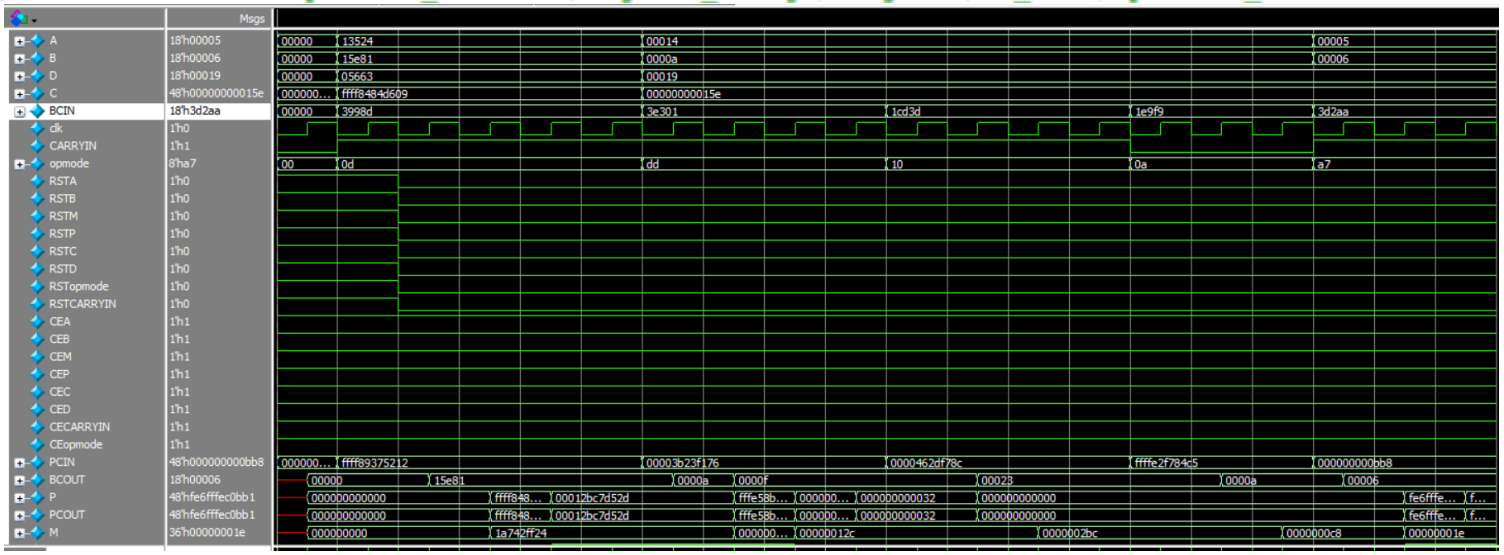
- **(SEC.3)** Do File

run_Spartan6.do

```
1 vlib work
2 vlog Spartan6.v Spartan6_tb.v
3 vsim -voptargs=+acc work.Spartan6_tb
4 add wave *
5 run -all
6 #quit -sim
```

- **(SEC.4)** Snippets of waveform ‘QuestaSim’

```
# Reset is done
# Verify DSP Path 1
# Passed
# Verify DSP Path 2
# Passed
# Verify DSP Path 3
# Passed
# Verify DSP Path 4
# Passed
# ** Note: $stop    : E:/Digital_Design/Project1/VS_code/Spartan6_tb.v(117)
#    Time: 40 ns  Iteration: 1  Instance: /Spartan6_tb
# Break in Module Spartan6_tb at E:/Digital_Design/Project1/VS_code/Spartan6_tb.v line 117
```



- **(SEC.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]

# ## Switches
# set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {in0[0]}]
# set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {in0[1]}]
# set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {in1[0]}]
# set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {in1[1]}]
# set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {in2[0]}]
# set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {in2[1]}]
# set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports {in3[0]}]
# set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {in3[7]}]
# #set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports {sw[8]}]
# #set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports {sw[9]}]
# #set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports {sw[10]}]
# #set_property -dict { PACKAGE_PIN R3    IOSTANDARD LVCMOS33 } [get_ports {sw[11]}]
```

```
# #set_property -dict { PACKAGE_PIN W2    IOSTANDARD LVCMOS33 } [get_ports {sw[12]}]
# #set_property -dict { PACKAGE_PIN U1    IOSTANDARD LVCMOS33 } [get_ports {sw[13]}]
# #set_property -dict { PACKAGE_PIN T1    IOSTANDARD LVCMOS33 } [get_ports {sw[14]}]
# #set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports {sw[15]}]

# ## LEDs
# set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {out}]
# #set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports {led[1]}]
# #set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports {led[2]}]
# #set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports {led[3]}]
# #set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports {led[4]}]
# #set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 } [get_ports {led[5]}]
# #set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
# #set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
# #set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
# #set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
# #set_property -dict { PACKAGE_PIN W3    IOSTANDARD LVCMOS33 } [get_ports {led[10]}]
# #set_property -dict { PACKAGE_PIN U3    IOSTANDARD LVCMOS33 } [get_ports {led[11]}]
# #set_property -dict { PACKAGE_PIN P3    IOSTANDARD LVCMOS33 } [get_ports {led[12]}]
# #set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports {led[13]}]
# #set_property -dict { PACKAGE_PIN P1    IOSTANDARD LVCMOS33 } [get_ports {led[14]}]
```

```
# #set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports {led[15]}]

# ##7 Segment Display
# #set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
# #set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
# #set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
# #set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
# #set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
# #set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
# #set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]

# #set_property -dict { PACKAGE_PIN V7   IOSTANDARD LVCMOS33 } [get_ports dp]

# #set_property -dict { PACKAGE_PIN U2   IOSTANDARD LVCMOS33 } [get_ports {an[0]}]
# #set_property -dict { PACKAGE_PIN U4   IOSTANDARD LVCMOS33 } [get_ports {an[1]}]
# #set_property -dict { PACKAGE_PIN V4   IOSTANDARD LVCMOS33 } [get_ports {an[2]}]
# #set_property -dict { PACKAGE_PIN W4   IOSTANDARD LVCMOS33 } [get_ports {an[3]}]

# ##Buttons
# set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports rst]
# #set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports btnU]
# #set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports btnL]
# #set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports btnR]
```

```
# #set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports btnD]

# ##Pmod Header JA
# #set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports {JA[0]}];#Sch name = JA1
# #set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports {JA[1]}];#Sch name = JA2
# #set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports {JA[2]}];#Sch name = JA3
# #set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports {JA[3]}];#Sch name = JA4
# #set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports {JA[4]}];#Sch name = JA7
# #set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports {JA[5]}];#Sch name = JA8
# #set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports {JA[6]}];#Sch name = JA9
# #set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports {JA[7]}];#Sch name = JA10

# ##Pmod Header JB
# #set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports {JB[0]}];#Sch name = JB1
# #set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports {JB[1]}];#Sch name = JB2
# #set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVCMOS33 } [get_ports {JB[2]}];#Sch name = JB3
# #set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports {JB[3]}];#Sch name = JB4
# #set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports {JB[4]}];#Sch name = JB7
# #set_property -dict { PACKAGE_PIN A17   IOSTANDARD LVCMOS33 } [get_ports {JB[5]}];#Sch name = JB8
# #set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports {JB[6]}];#Sch name = JB9
# #set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports {JB[7]}];#Sch name = JB10

# ##Pmod Header JC
```

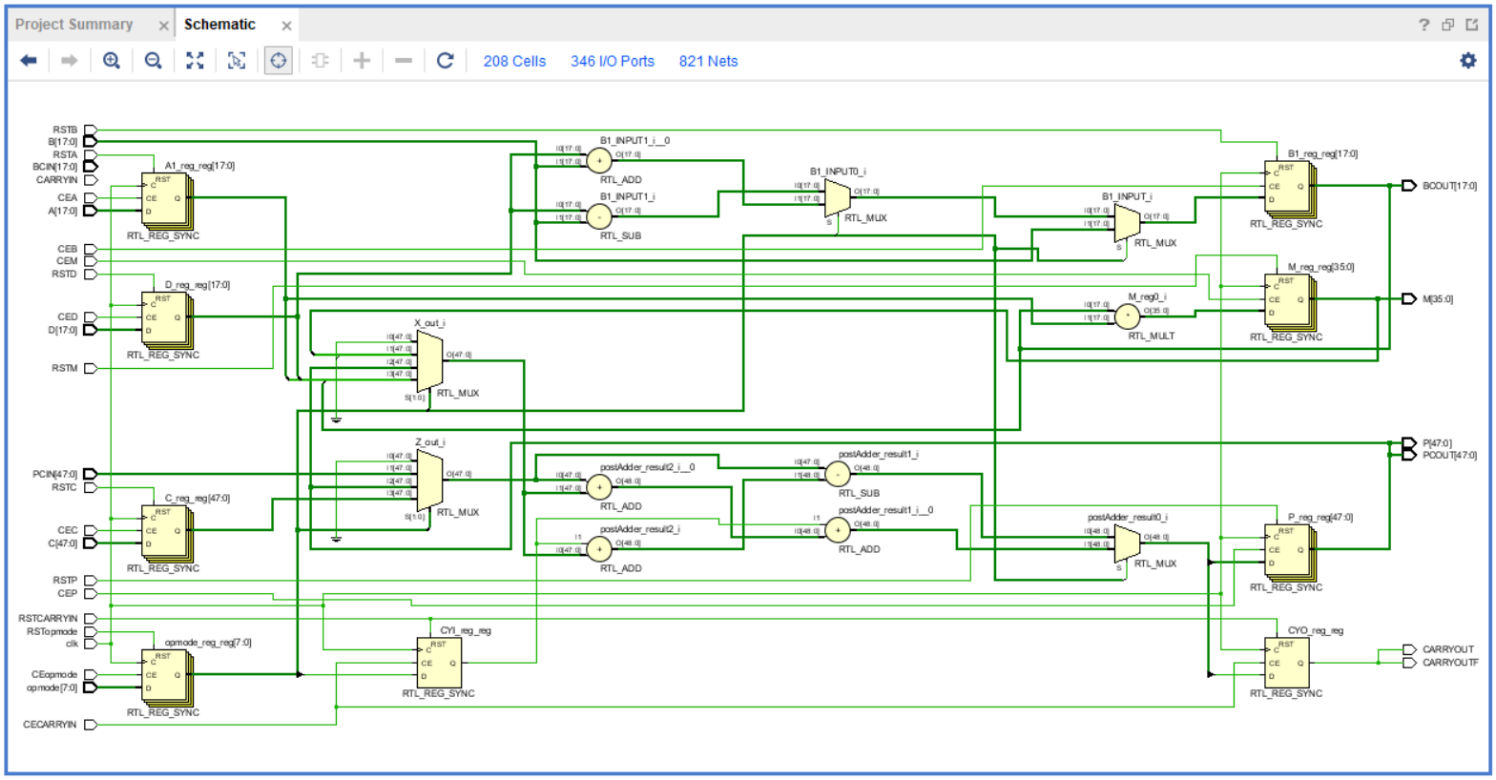
```
# #set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports {JC[0]}];#Sch name = JC1
# #set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports {JC[1]}];#Sch name = JC2
# #set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports {JC[2]}];#Sch name = JC3
# #set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports {JC[3]}];#Sch name = JC4
# #set_property -dict { PACKAGE_PIN L17   IOSTANDARD LVCMOS33 } [get_ports {JC[4]}];#Sch name = JC7
# #set_property -dict { PACKAGE_PIN M19   IOSTANDARD LVCMOS33 } [get_ports {JC[5]}];#Sch name = JC8
# #set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports {JC[6]}];#Sch name = JC9
# #set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports {JC[7]}];#Sch name = JC10

# ##Pmod Header JXADC
# #set_property -dict { PACKAGE_PIN J3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[0]}];#Sch name = XA1_P
# #set_property -dict { PACKAGE_PIN L3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[1]}];#Sch name = XA2_P
# #set_property -dict { PACKAGE_PIN M2   IOSTANDARD LVCMOS33 } [get_ports {JXADC[2]}];#Sch name = XA3_P
# #set_property -dict { PACKAGE_PIN N2   IOSTANDARD LVCMOS33 } [get_ports {JXADC[3]}];#Sch name = XA4_P
# #set_property -dict { PACKAGE_PIN K3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[4]}];#Sch name = XA1_N
# #set_property -dict { PACKAGE_PIN M3   IOSTANDARD LVCMOS33 } [get_ports {JXADC[5]}];#Sch name = XA2_N
# #set_property -dict { PACKAGE_PIN M1   IOSTANDARD LVCMOS33 } [get_ports {JXADC[6]}];#Sch name = XA3_N
# #set_property -dict { PACKAGE_PIN N1   IOSTANDARD LVCMOS33 } [get_ports {JXADC[7]}];#Sch name = XA4_N

# ##VGA Connector
# #set_property -dict { PACKAGE_PIN G19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[0]}]
# #set_property -dict { PACKAGE_PIN H19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[1]}]
```

```
# #set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[2]}]
# #set_property -dict { PACKAGE_PIN N19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[3]}]
# #set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[0]}]
# #set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[1]}]
# #set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[2]}]
# #set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[3]}]
# #set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[0]}]
# #set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[1]}]
# #set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[2]}]
# #set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[3]}]
# #set_property -dict { PACKAGE_PIN P19   IOSTANDARD LVCMOS33 } [get_ports Hsync]
# #set_property -dict { PACKAGE_PIN R19   IOSTANDARD LVCMOS33 } [get_ports Vsync]
#
# ##USB-RS232 Interface
# #set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports RsRx]
# #set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports RsTx]
#
# ##USB HID (PS/2)
# #set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports PS2Clk]
# #set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports PS2Data]
#
# ##Quad SPI Flash
# ##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the
```

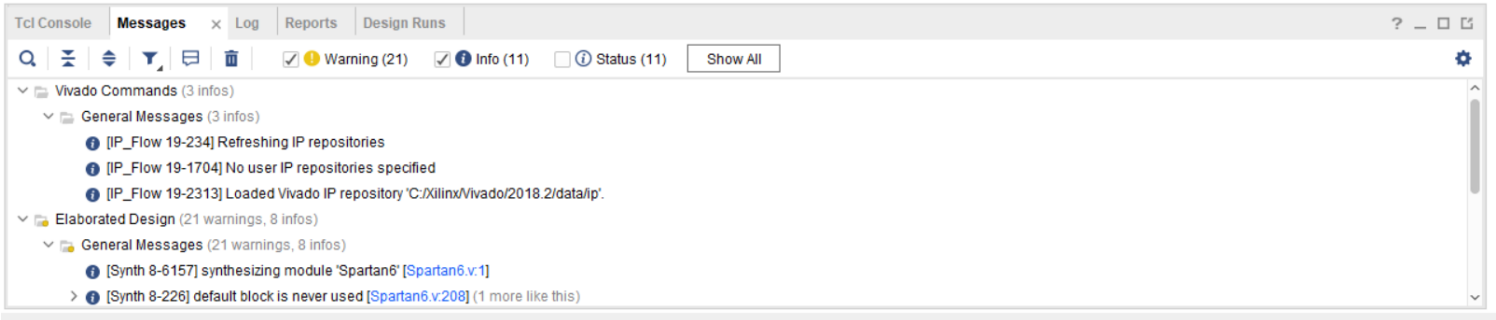
```
# ##STARTUPE2 primitive.
# #set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[0]}]
# #set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[1]}]
# #set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[2]}]
# #set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[3]}]
# #set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports QspiCSn]

# ## 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]
```

- **(SEC.6)** Vivado ‘Elaboration’

Schematic

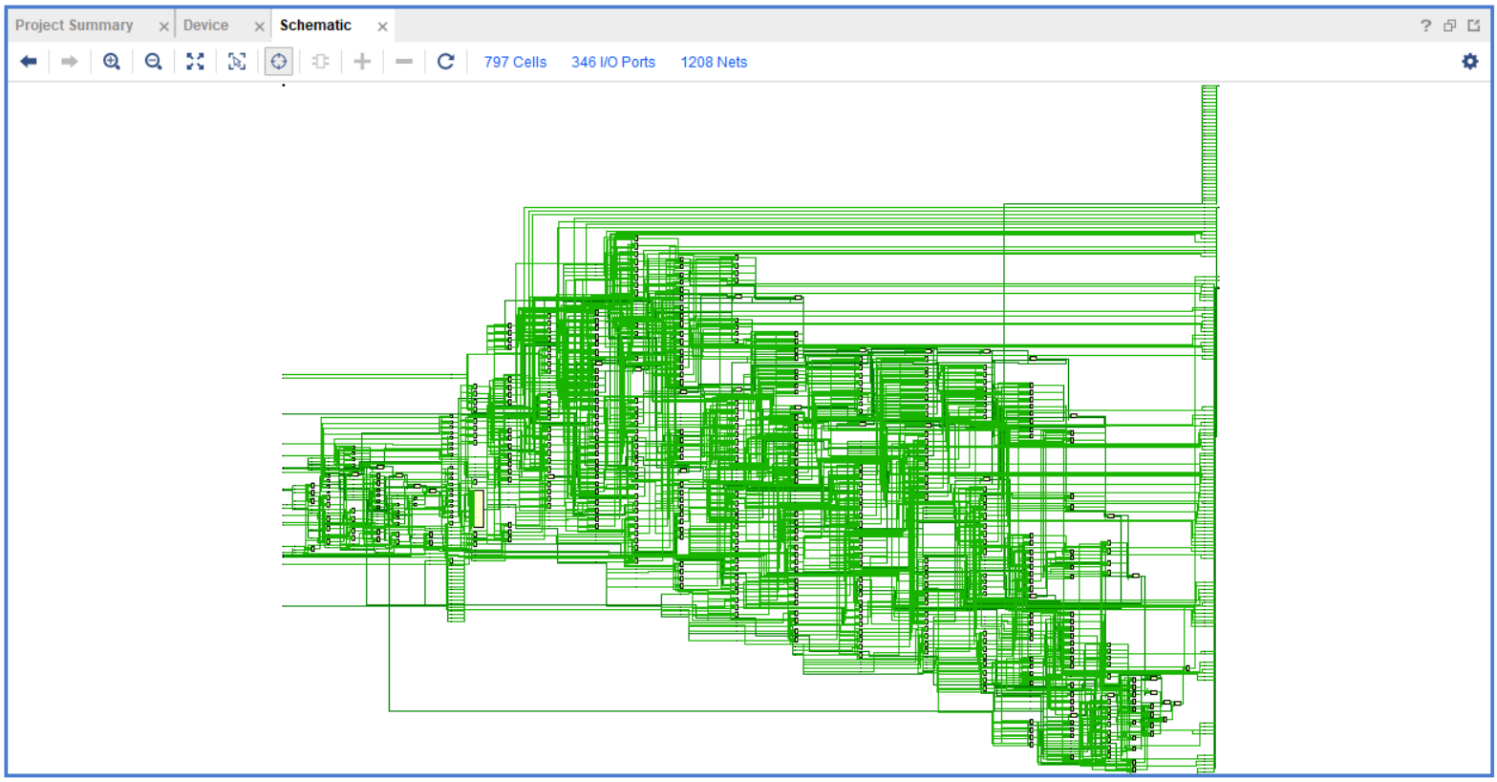


Message Tab



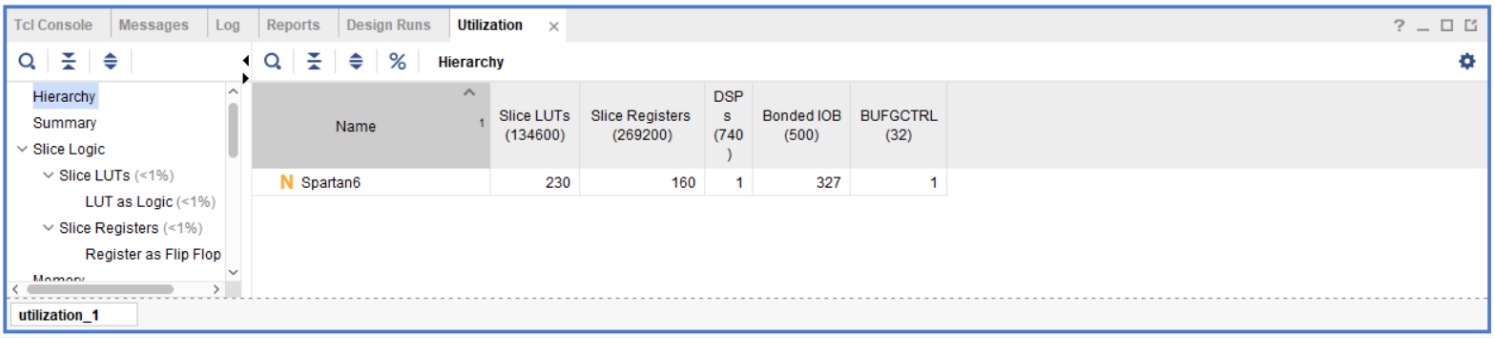
- **(SEC.7)** Vivado ‘Synthesis’

Schematic

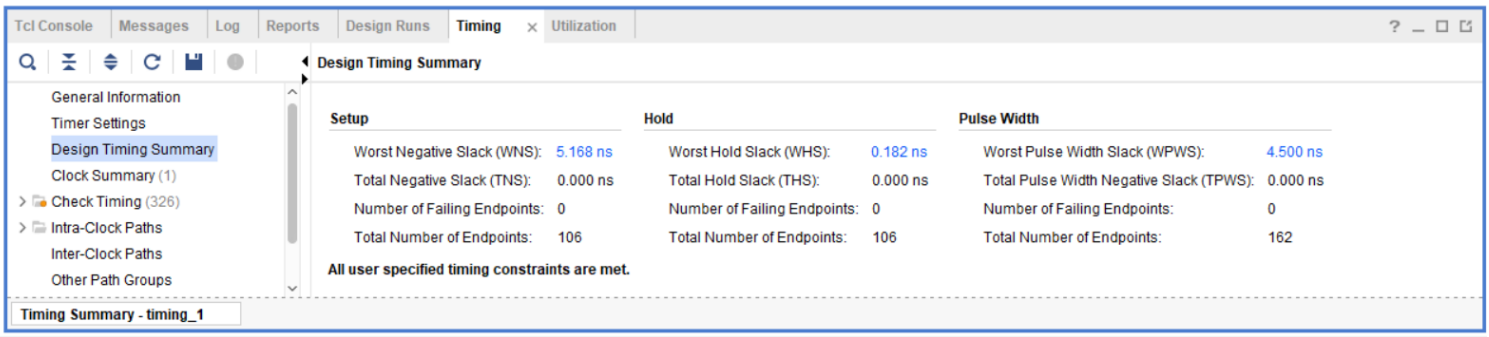


Message Tab

Utilization Report

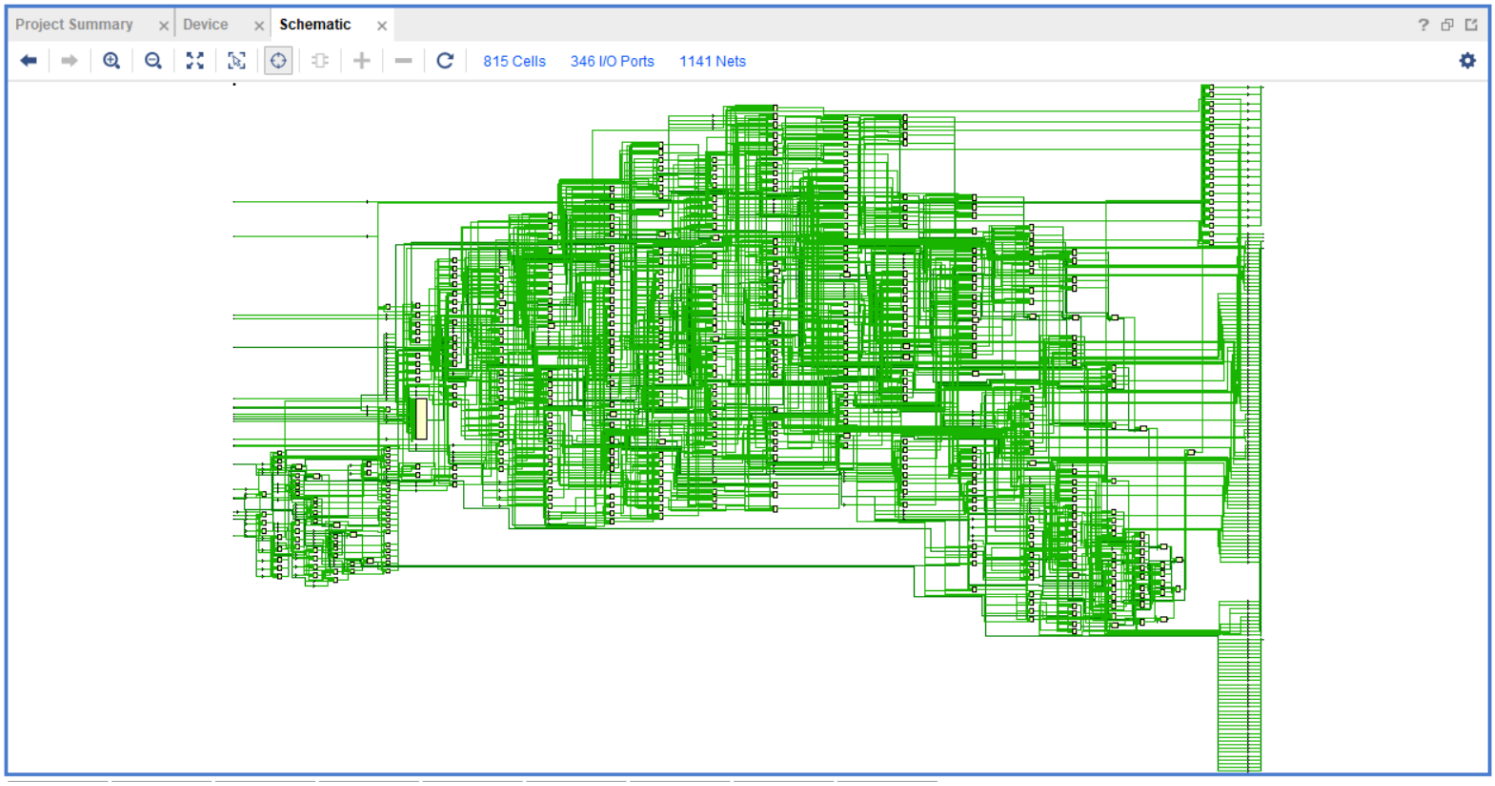


Timing Report

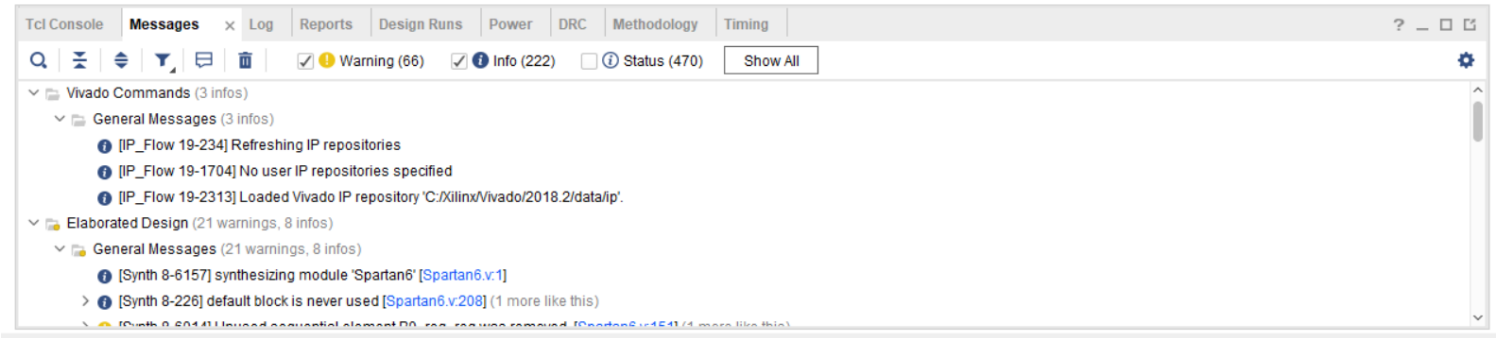


- **(SEC.8)** Vivado ‘Implementation’

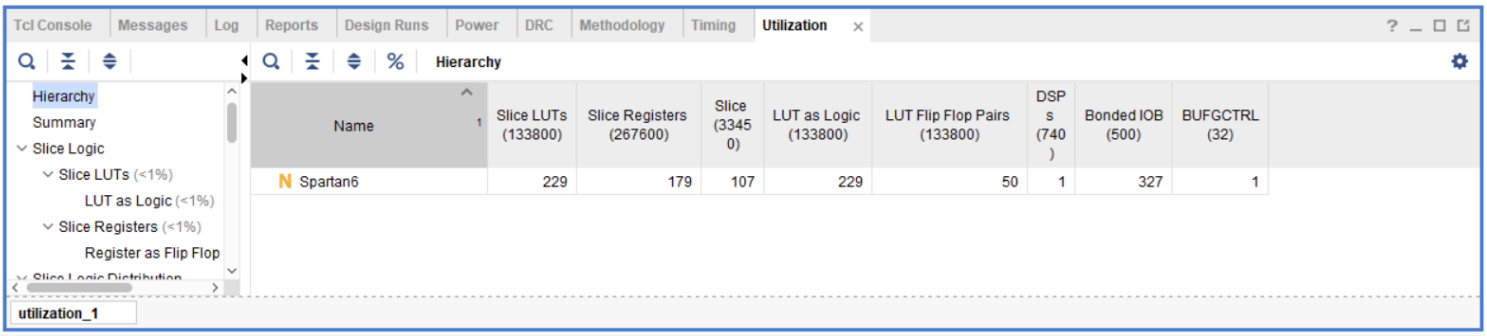
Schematic



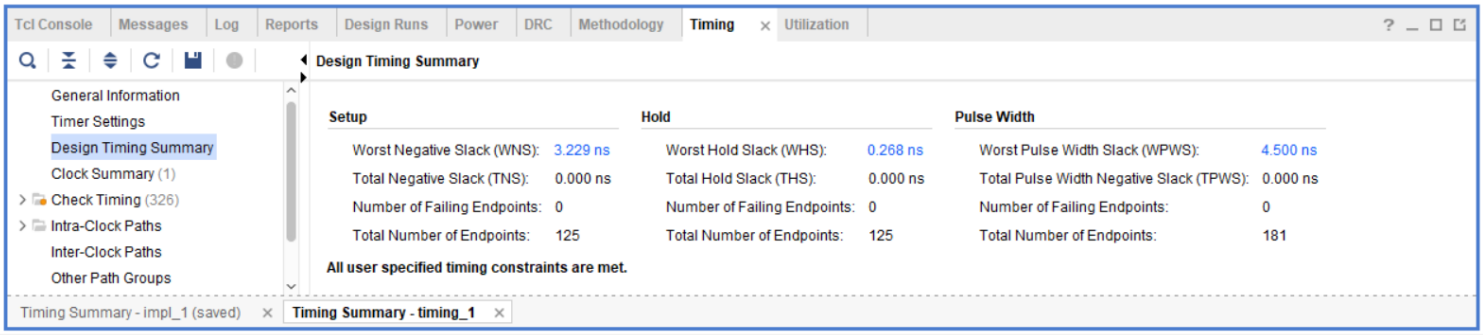
Message Tab



Utilization Report



Timing Report



- **(SEC.8)** QuestaLint ‘Linting’