RTL Code

Code For RegMux

```
 1. module RegMux (clk, CE, rst, data, out);
 2.     parameter N = 18;
 3.     parameter RSTTYPE = "SYNC";
 4.     parameter DATAREG = 0;
 5.
 6.     input clk, CE, rst;
 7.     input [N - 1 : 0] data;
 8.
 9.     output [N - 1 : 0] out;
10.
11.     reg [N - 1 : 0] data_reg;
12.
13.     generate
14.         if (DATAREG) begin
15.             if (RSTTYPE == "SYNC") begin
16.                 always @(posedge clk) begin
17.                     if (rst) begin
18.                         data_reg <= 0;
19.                     end else if (CE) begin
20.                         data_reg <= data;
21.                     end
22.                 end
23.             end else begin
24.                 always @(posedge clk or posedge rst) begin
25.                     if (rst) begin
26.                         data_reg <= 0;
27.                     end else if (CE) begin
28.                         data_reg <= data;
29.                     end
30.                 end
31.             end
32.         end
33.     endgenerate
34.
35.     generate
36.         if (DATAREG) begin
37.             assign out = data_reg;
38.         end else begin
39.             assign out = data;
40.         end
41.     endgenerate
42. endmodule
43.
```

Code For Mul

```
1. module Mul (in0, in1, out);
2.     input [17:0] in0, in1;
3.
4.     output [35:0] out;
5.
6.     assign out = in0 * in1;
7. endmodule
8.
```

Code For PreAddSub

```
1. module PreAddSub (mode, in0, in1, out);
2.     input mode;
3.     input [17:0] in0, in1;
4.
5.     output [17:0] out;
6.
7.     assign out = (mode == 0)? in0 + in1 : in0 - in1;
8. endmodule
9.
```

Code For PostAddSub

```
 1. module PostAddSub (mode, in0, in1, cin, out, carryOut);
 2.     input mode, cin;
 3.     input [47:0] in0, in1;
 4.
 5.     output [47:0] out;
 6.     output carryOut;
 7.
 8.     assign {carryOut, out} = (mode == 0)? in0 + in1 + cin : in1 - in0 - cin;
 9. endmodule
10.
```

Code For DSP

```
 1. module DSP (clk, opMode, CEA, CEB, CEC, CECarryIn, CED, CEM, CEOpMode, CEP,
 2.                 rstA, rstB, rstC, rstCarryIn, rstD, rstM, rstOpMode, rstP,
 3.                 A, B, D, C, carryIn, BCIn, PCIn,
 4.                 BCOut, PCOut, P, M, carryOut, carryOutF);
 5.     parameter A0REG = 0;
 6.     parameter A1REG = 1;
 7.     parameter B0REG = 0;
 8.     parameter B1REG = 1;
 9.
10.     parameter CREG = 1;
11.     parameter DREG = 1;
12.     parameter MREG = 1;
13.     parameter PREG = 1;
14.     parameter CARRYINREG = 1;
15.     parameter CARRYOUTREG = 1;
16.     parameter OPMODEREG = 1;
17.
18.     parameter CARRYINSEL = "OPMODE5";
19.     parameter B_INPUT = "DIRECT";
20.     parameter RSTTYPE = "SYNC";
21.
22.     input clk, carryIn, rstA, rstB, rstC, rstCarryIn, rstD, rstM, rstOpMode, rstP,
23.             CEA, CEB, CEC, CECarryIn, CED, CEM, CEOpMode, CEP;
24.     input [17:0] A, B, D, BCIn;
25.     input [47:0] C, PCIn;
26.     input [7:0] opMode;
27.
28.     output carryOut, carryOutF;
29.     output [17:0] BCOut;
30.     output [47:0] PCOut, P;
31.     output [35:0] M;
32.
```

```
33.     wire [7:0] opModeRegOut;
34.     wire [17:0] dRegOut, bIn, b0RegOut, a0RegOut, PreASOut, b1RegIn,
35.                 b1RegOut, a1RegOut;
36.     wire [35:0] mulOut;
37.     wire carryInRegIn, carryInRegOut, carryOutRegIn;
38.     wire [47:0] cRegOut, Z, X, pRegIn;
39.
40.     RegMux #(8, RSTTYPE, OPMODEREG) opReg (clk, CEOpMode, rstOpMode, opMode, opModeRegOut);
41.     RegMux #(18, RSTTYPE, DREG) dReg (clk, CED, rstD, D, dRegOut);
42.     RegMux #(18, RSTTYPE, B0REG) b0Reg (clk, CEB, rstB, bIn, b0RegOut);
43.     RegMux #(18, RSTTYPE, A0REG) a0Reg (clk, CEA, rstA, A, a0RegOut);
44.     RegMux #(48, RSTTYPE, CREG) cReg (clk, CEC, rstC, C, cRegOut);
45.
46.     PreAddSub PreAS (opModeRegOut[6], dRegOut, b0RegOut, PreASOut);
47.
48.     RegMux #(18, RSTTYPE, B1REG) b1Reg (clk, CEB, rstB, b1RegIn, b1RegOut);
49.     RegMux #(18, RSTTYPE, A1REG) a1Reg (clk, CEA, rstA, a0RegOut, a1RegOut);
50.
51.     Mul mul (b1RegOut, a1RegOut, mulOut);
52.
53.     RegMux #(36, RSTTYPE, MREG) mulReg (clk, CEM, rstM, mulOut, M);
54.     RegMux #(1, RSTTYPE, CARRYINREG) carryInReg (clk, CECarryIn, rstCarryIn, carryInRegIn,
carryInRegOut);
55.
56.     PostAddSub PostAS (opModeRegOut[7], X, Z, carryInRegOut, pRegIn, carryOutRegIn);
57.
58.     RegMux #(1, RSTTYPE, CARRYOUTREG) carryOutReg (clk, CECarryIn, rstCarryIn, carryOutRegIn,
carryOut);
59.     RegMux #(48, RSTTYPE, PREG) pReg (clk, CEP, rstP, pRegIn, P);
60.
61.     generate
62.         if (B_INPUT == "DIRECT") begin
63.             assign bIn = B;
64.         end else if (B_INPUT == "CASCADE") begin
65.             assign bIn = BCIn;
66.         end else begin
67.             assign bIn = 0;
68.         end
69.     endgenerate
70.
71.     assign b1RegIn = (opModeRegOut[4] == 0)? b0RegOut : PreASOut;
72.
73.     generate
74.         if (CARRYINSEL == "OPMODE5") begin
75.             assign carryInRegIn = opModeRegOut[5];
76.         end else if (CARRYINSEL == "CARRYIN") begin
77.             assign carryInRegIn = carryIn;
78.         end else begin
79.             assign carryInRegIn = 0;
80.         end
81.     endgenerate
82.
83.     assign Z = (opModeRegOut[3:2] == 0)? 0 : (opModeRegOut[3:2] == 1)? PCIn :
84.                 (opModeRegOut[3:2] == 2)? P : cRegOut;
85.
86.     assign X = (opModeRegOut[1:0] == 0)? 0 : (opModeRegOut[1:0] == 1)? M :
87.                 (opModeRegOut[1:0] == 2)? P : {dRegOut, a1RegOut, b1RegOut};
88.
89.     assign BCOut = b1RegOut;
90.     assign carryOutF = carryOut;
91.     assign PCOut = P;
92. endmodule
93.
```

Testbench Code

```
 1. module DSP_tb ();
 2.     reg clk, carryIn_tb, rstA_tb, rstB_tb, rstC_tb, rstCarryIn_tb, rstD_tb, rstM_tb,
rstOpMode_tb, rstP_tb,
 3.             CEA_tb, CEB_tb, CEC_tb, CECarryIn_tb, CED_tb, CEM_tb, CEOpMode_tb, CEP_tb;
 4.     reg [17:0] A_tb, B_tb, D_tb, BCIn_tb;
 5.     reg [47:0] C_tb, PCIn_tb;
 6.     reg [7:0] opMode_tb;
 7.
 8.     wire carryOut_dut, carryOutF_dut;
 9.     wire [17:0] BCOut_dut;
10.     wire [47:0] PCOut_dut, P_dut;
11.     wire [35:0] M_dut;
12.
13.     reg carryOut_exp, carryOutF_exp;
14.     reg [17:0] BCOut_exp;
15.     reg [47:0] PCOut_exp, P_exp;
16.     reg [35:0] M_exp;
17.
18.     DSP dut (clk, opMode_tb, CEA_tb, CEB_tb, CEC_tb, CECarryIn_tb, CED_tb, CEM_tb, CEOpMode_tb,
CEP_tb,
19.                 rstA_tb, rstB_tb, rstC_tb, rstCarryIn_tb, rstD_tb, rstM_tb, rstOpMode_tb,
rstP_tb,
20.                 A_tb, B_tb, D_tb, C_tb, carryIn_tb, BCIn_tb, PCIn_tb,
21.                 BCOut_dut, PCOut_dut, P_dut, M_dut, carryOut_dut, carryOutF_dut);
22.
23.     initial begin
24.         clk = 0;
25.         forever begin
26.             #1 clk = ~clk;
27.         end
28.     end
29.
30.     initial begin
31.         rstA_tb = 1;
32.         rstB_tb = 1;
33.         rstC_tb = 1;
34.         rstCarryIn_tb = 1;
35.         rstD_tb = 1;
36.         rstM_tb = 1;
37.         rstOpMode_tb = 1;
38.         rstP_tb = 1;
39.
40.         CEA_tb = $random;
41.         CEB_tb = $random;
42.         CEC_tb = $random;
43.         CECarryIn_tb = $random;
44.         CED_tb = $random;
45.         CEM_tb = $random;
46.         CEOpMode_tb = $random;
47.         CEP_tb = $random;
48.
49.         carryIn_tb = $random;
50.         A_tb = $random;
51.         B_tb = $random;
52.         D_tb = $random;
53.         BCIn_tb = $random;
54.         C_tb = $random;
55.         PCIn_tb = $random;
56.         opMode_tb = $random;
57.
58.         carryOut_exp = 0;
```

```
59.         carryOutF_exp = 0;
60.         BCOut_exp = 0;
61.         PCOut_exp = 0;
62.         P_exp = 0;
63.         M_exp = 0;
64.
65.         @(negedge clk);
66.
67.         if (carryOut_dut != carryOut_exp || carryOutF_dut != carryOutF_exp
68.             || BCOut_dut != BCOut_exp || PCOut_dut != PCOut_exp
69.             || P_dut != P_exp || M_dut != M_exp) begin
70.             $display("Error in rst, carryIn: %d, A: %d, B: %d, D: %d, BCIn: %d, C: %d, PCIn:
%d, opMode: %d,\n
71.                       carryOut_dut: %d, carryOutF_dut: %d, BCOut_dut: %d, PCOut_dut: %d,
P_dut: %d, M_dut: %d,\n
72.                       carryOut_exp: %d, carryOutF_exp: %d, BCOut_exp: %d, PCOut_exp: %d,
P_exp: %d, M_exp: %d\n",
73.                       carryIn_tb, A_tb, B_tb, D_tb, BCIn_tb, C_tb, PCIn_tb, opMode_tb,
74.                       carryOut_dut, carryOutF_dut, BCOut_dut, PCOut_dut, P_dut, M_dut,
75.                       carryOut_exp, carryOutF_exp, BCOut_exp, PCOut_exp, P_exp, M_exp);
76.             $stop;
77.         end
78.
79.         rstA_tb = 0;
80.         rstB_tb = 0;
81.         rstC_tb = 0;
82.         rstCarryIn_tb = 0;
83.         rstD_tb = 0;
84.         rstM_tb = 0;
85.         rstOpMode_tb = 0;
86.         rstP_tb = 0;
87.
88.         CEA_tb = 1;
89.         CEB_tb = 1;
90.         CEC_tb = 1;
91.         CECarryIn_tb = 1;
92.         CED_tb = 1;
93.         CEM_tb = 1;
94.         CEOpMode_tb = 1;
95.         CEP_tb = 1;
96.
97.         carryIn_tb = $random;
98.         A_tb = 20;
99.         B_tb = 10;
100.        D_tb = 25;
101.        BCIn_tb = $random;
102.        C_tb = 350;
103.        PCIn_tb = $random;
104.        opMode_tb = 8'b1101_1101;
105.
106.        carryOut_exp = 0;
107.        carryOutF_exp = 0;
108.        BCOut_exp = 'hf;
109.        PCOut_exp = 'h32;
110.        P_exp = 'h32;
111.        M_exp = 'h12c;
112.
113.        repeat (4) begin
114.            @(negedge clk);
115.        end
116.
117.        if (carryOut_dut != carryOut_exp || carryOutF_dut != carryOutF_exp
118.            || BCOut_dut != BCOut_exp || PCOut_dut != PCOut_exp
119.            || P_dut != P_exp || M_dut != M_exp) begin
```

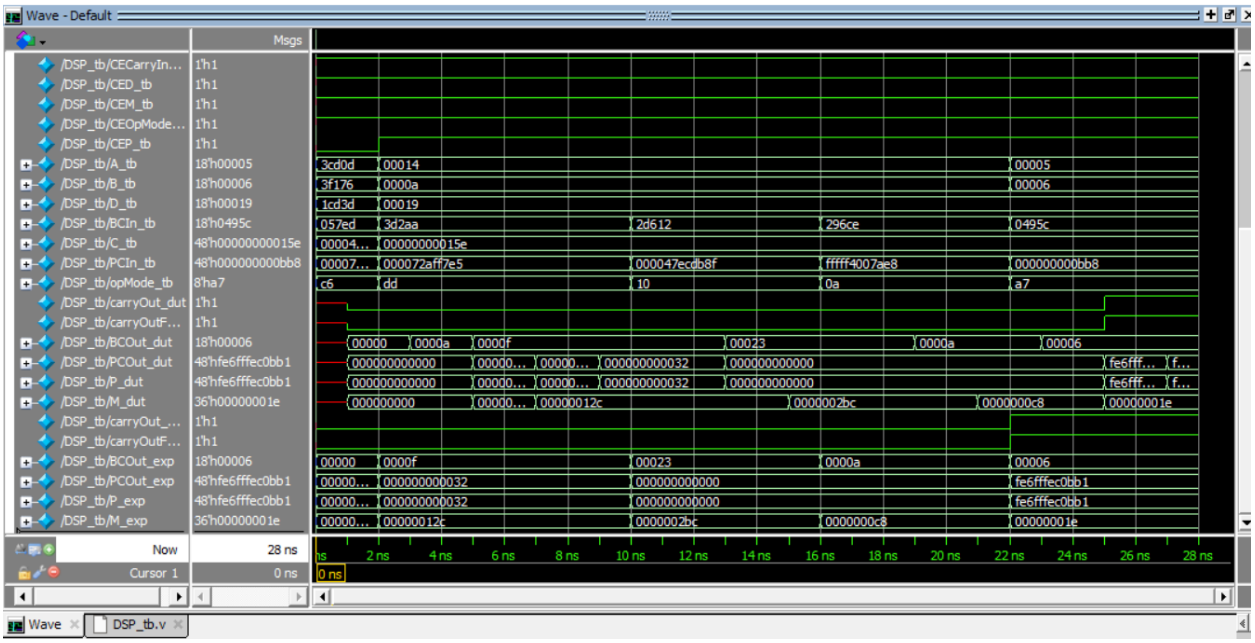
```
120.             $display("Error in path 1, carryIn: %d, A: %d, B: %d, D: %d, BCIn: %d, C: %d, PCIn:
%d, opMode: %d,\n
121.                          carryOut_dut: %d, carryOutF_dut: %d, BCOut_dut: %d, PCOut_dut: %d,
P_dut: %d, M_dut: %d,\n
122.                          carryOut_exp: %d, carryOutF_exp: %d, BCOut_exp: %d, PCOut_exp: %d,
P_exp: %d, M_exp: %d\n",
123.                          carryIn_tb, A_tb, B_tb, D_tb, BCIn_tb, C_tb, PCIn_tb, opMode_tb,
124.                          carryOut_dut, carryOutF_dut, BCOut_dut, PCOut_dut, P_dut, M_dut,
125.                          carryOut_exp, carryOutF_exp, BCOut_exp, PCOut_exp, P_exp, M_exp);
126.             $stop;
127.         end
128.
129.         carryIn_tb = $random;
130.         A_tb = 20;
131.         B_tb = 10;
132.         D_tb = 25;
133.         BCIn_tb = $random;
134.         C_tb = 350;
135.         PCIn_tb = $random;
136.         opMode_tb = 8'b0001_0000;
137.
138.         carryOut_exp = 0;
139.         carryOutF_exp = 0;
140.         BCOut_exp = 'h23;
141.         PCOut_exp = 0;
142.         P_exp = 0;
143.         M_exp = 'h2bc;
144.
145.         repeat (3) begin
146.             @(negedge clk);
147.         end
148.
149.         if (carryOut_dut != carryOut_exp || carryOutF_dut != carryOutF_exp
150.             || BCOut_dut != BCOut_exp || PCOut_dut != PCOut_exp
151.             || P_dut != P_exp || M_dut != M_exp) begin
152.             $display("Error in path 2, carryIn: %d, A: %d, B: %d, D: %d, BCIn: %d, C: %d, PCIn:
%d, opMode: %d,\n
153.                          carryOut_dut: %d, carryOutF_dut: %d, BCOut_dut: %d, PCOut_dut: %d,
P_dut: %d, M_dut: %d,\n
154.                          carryOut_exp: %d, carryOutF_exp: %d, BCOut_exp: %d, PCOut_exp: %d,
P_exp: %d, M_exp: %d\n",
155.                          carryIn_tb, A_tb, B_tb, D_tb, BCIn_tb, C_tb, PCIn_tb, opMode_tb,
156.                          carryOut_dut, carryOutF_dut, BCOut_dut, PCOut_dut, P_dut, M_dut,
157.                          carryOut_exp, carryOutF_exp, BCOut_exp, PCOut_exp, P_exp, M_exp);
158.             $stop;
159.         end
160.
161.         carryIn_tb = $random;
162.         A_tb = 20;
163.         B_tb = 10;
164.         D_tb = 25;
165.         BCIn_tb = $random;
166.         C_tb = 350;
167.         PCIn_tb = $random;
168.         opMode_tb = 8'b0000_1010;
169.
170.         BCOut_exp = 'ha;
171.         M_exp = 'hc8;
172.
173.         repeat (3) begin
174.             @(negedge clk);
175.         end
176.
177.         if (carryOut_dut != carryOut_exp || carryOutF_dut != carryOutF_exp
178.             || BCOut_dut != BCOut_exp || PCOut_dut != PCOut_exp
```

```
            || P_dut != P_exp || M_dut != M_exp) begin
            $display("Error in path 3, carryIn: %d, A: %d, B: %d, D: %d, BCIn: %d, C: %d, PCIn:
%d, opMode: %d,\n
                        carryOut_dut: %d, carryOutF_dut: %d, BCOut_dut: %d, PCOut_dut: %d,
P_dut: %d, M_dut: %d,\n
                        carryOut_exp: %d, carryOutF_exp: %d, BCOut_exp: %d, PCOut_exp: %d,
P_exp: %d, M_exp: %d\n",
                        carryIn_tb, A_tb, B_tb, D_tb, BCIn_tb, C_tb, PCIn_tb, opMode_tb,
                        carryOut_dut, carryOutF_dut, BCOut_dut, PCOut_dut, P_dut, M_dut,
                        carryOut_exp, carryOutF_exp, BCOut_exp, PCOut_exp, P_exp, M_exp);
            $stop;
        end

        carryIn_tb = $random;
        A_tb = 5;
        B_tb = 6;
        D_tb = 25;
        BCIn_tb = $random;
        C_tb = 350;
        PCIn_tb = 3000;
        opMode_tb = 8'b1010_0111;

        carryOut_exp = 1;
        carryOutF_exp = 1;
        BCOut_exp = 'h6;
        PCOut_exp = 'hfe6fffec0bb1;
        P_exp = 'hfe6fffec0bb1;
        M_exp = 'h1e;

        repeat (3) begin
            @(negedge clk);
        end

        if (carryOut_dut != carryOut_exp || carryOutF_dut != carryOutF_exp
            || BCOut_dut != BCOut_exp || PCOut_dut != PCOut_exp
            || P_dut != P_exp || M_dut != M_exp) begin
            $display("Error in path 4, carryIn: %d, A: %d, B: %d, D: %d, BCIn: %d, C: %d, PCIn:
%d, opMode: %d,\n
                        carryOut_dut: %d, carryOutF_dut: %d, BCOut_dut: %d, PCOut_dut: %d,
P_dut: %d, M_dut: %d,\n
                        carryOut_exp: %d, carryOutF_exp: %d, BCOut_exp: %d, PCOut_exp: %d,
P_exp: %d, M_exp: %d\n",
                        carryIn_tb, A_tb, B_tb, D_tb, BCIn_tb, C_tb, PCIn_tb, opMode_tb,
                        carryOut_dut, carryOutF_dut, BCOut_dut, PCOut_dut, P_dut, M_dut,
                        carryOut_exp, carryOutF_exp, BCOut_exp, PCOut_exp, P_exp, M_exp);
            $stop;
        end

        $stop;
    end
endmodule

```

Do File

```
vlib work
vlog DSP.v DSP_tb.v RegMux.v PreAddSub.v PostAddSub.v Mul.v
vsim -voptargs=+acc work.DSP_tb
add wave *
run -all
#quit -sim

```

QuestaSim Snippets



Constraints 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 {sw[0]}]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {sw[1]}]
#set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {sw[2]}]
#set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {sw[3]}]
#set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {sw[4]}]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {sw[5]}]
#set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports {sw[6]}]
#set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {sw[7]}]
```

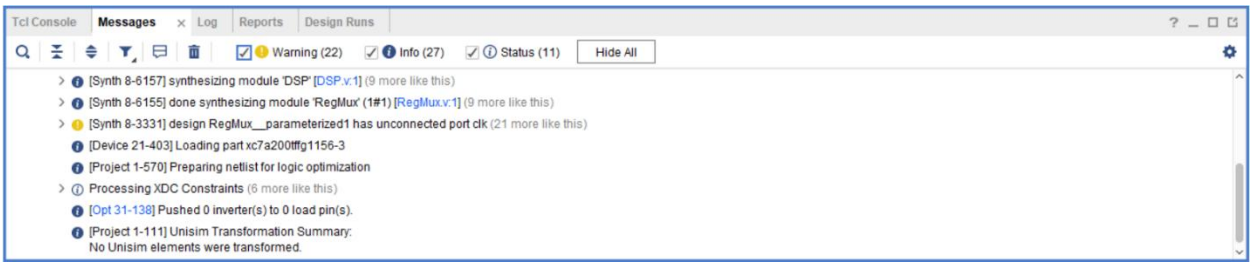
```
#set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports {sw[8]}]
#set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports {sw[9]}]
#set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports {sw[10]}]
#set_property -dict { PACKAGE_PIN R3    IOSTANDARD 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 {led[0]}]
#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]}]
```

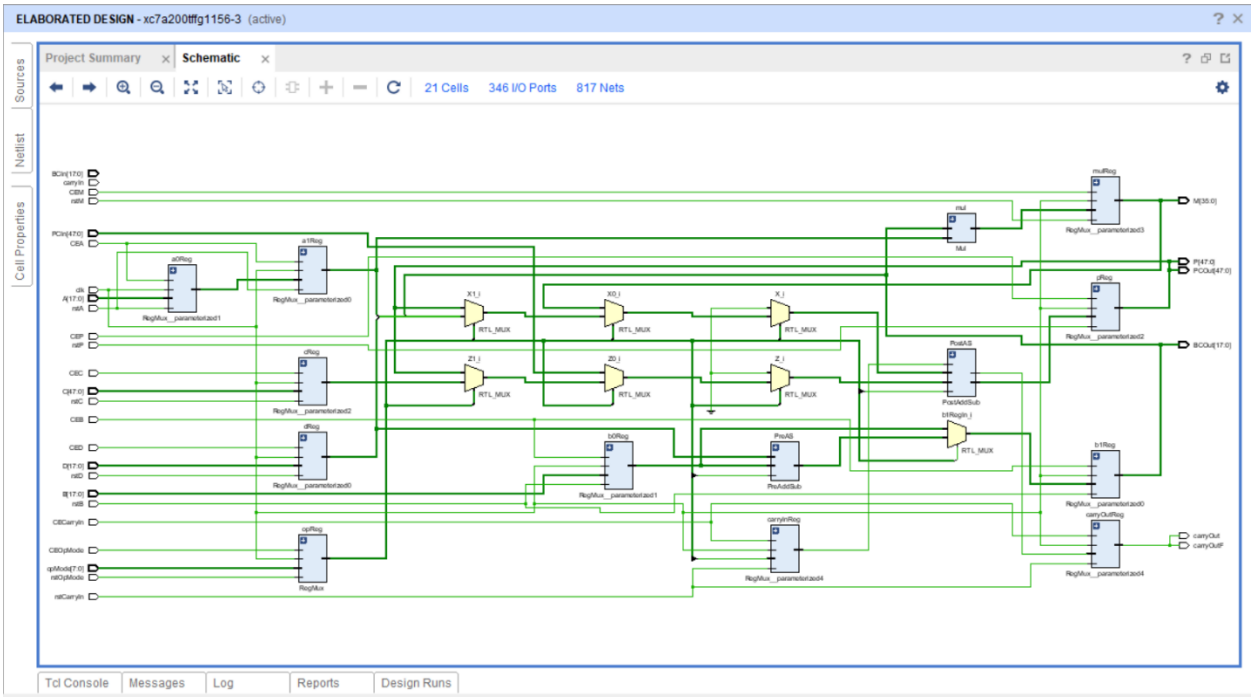
```
#set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[2]}]
#set_property -dict { PACKAGE_PIN N19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[3]}]
#set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[0]}]
#set_property -dict { PACKAGE_PIN L18   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]

```

Elaboration

Messages



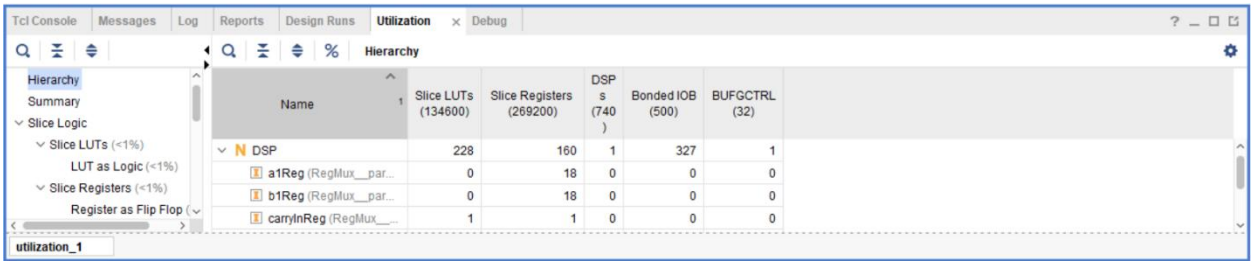
Schematic



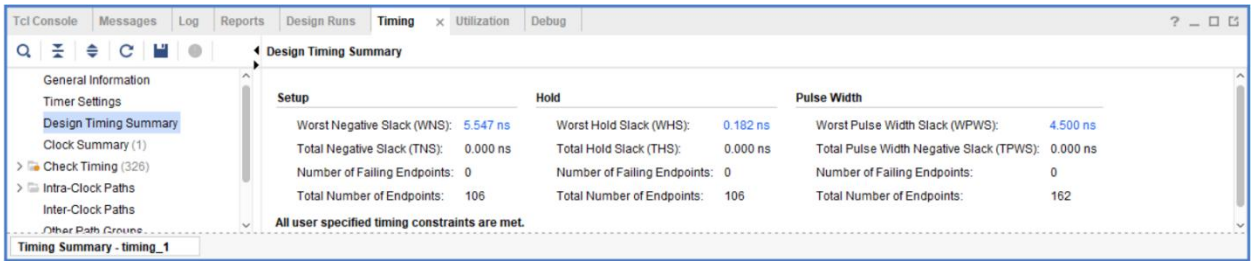
Synthesis

Messages

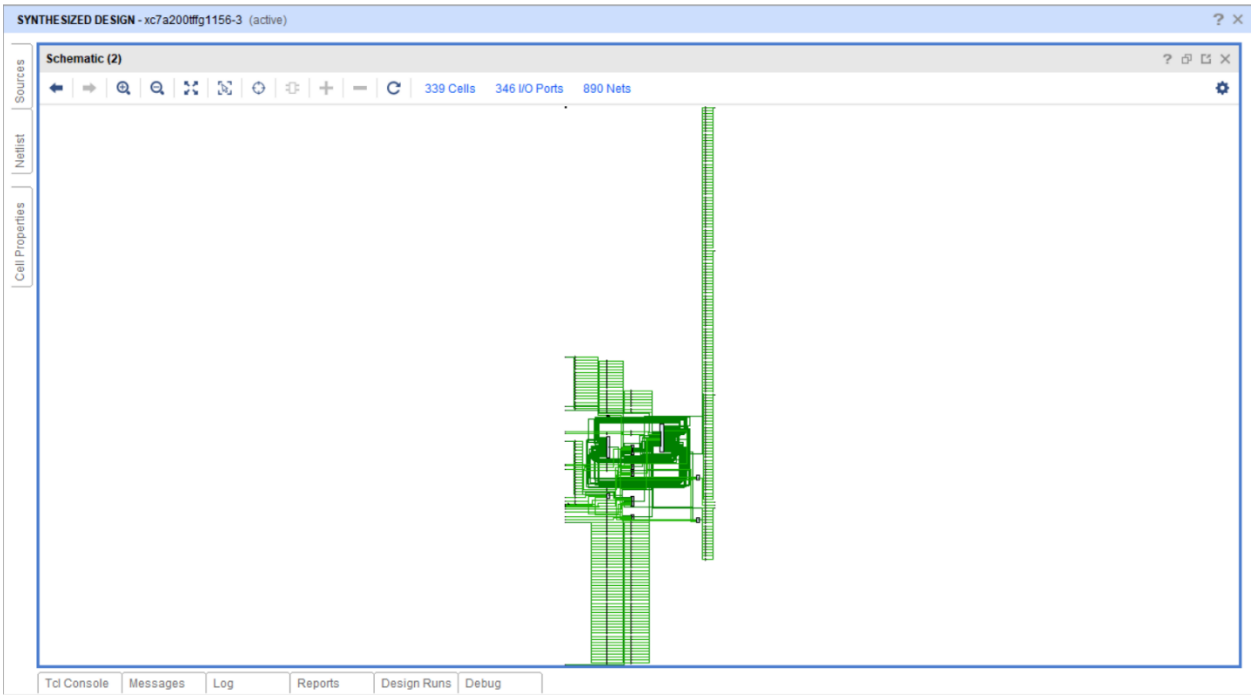
Utilization Report



Timing Report

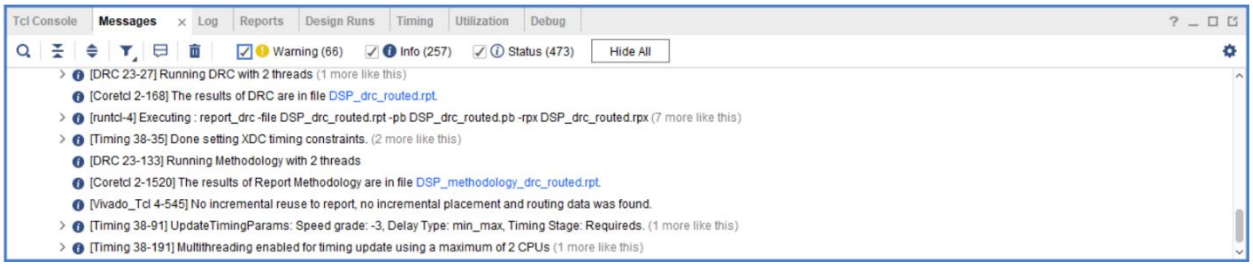


Schematic

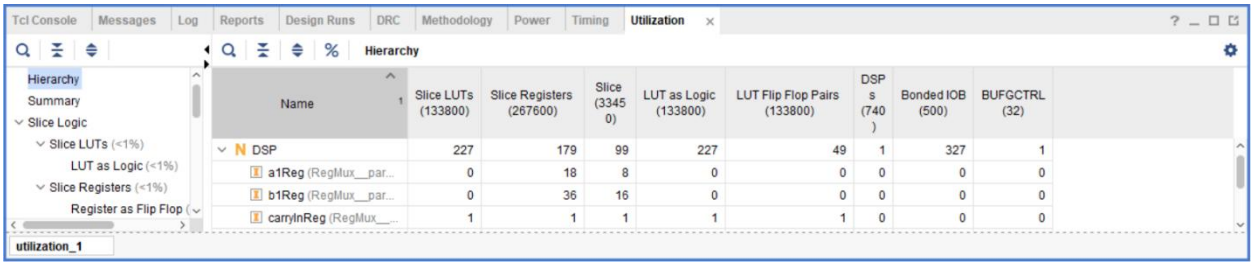


Implementation

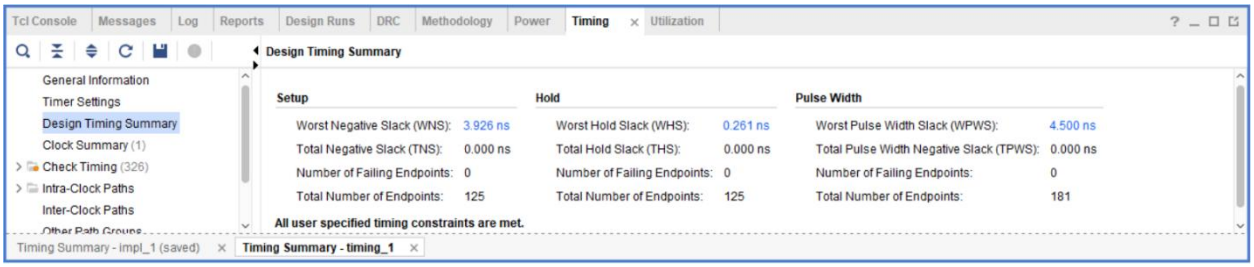
Messages



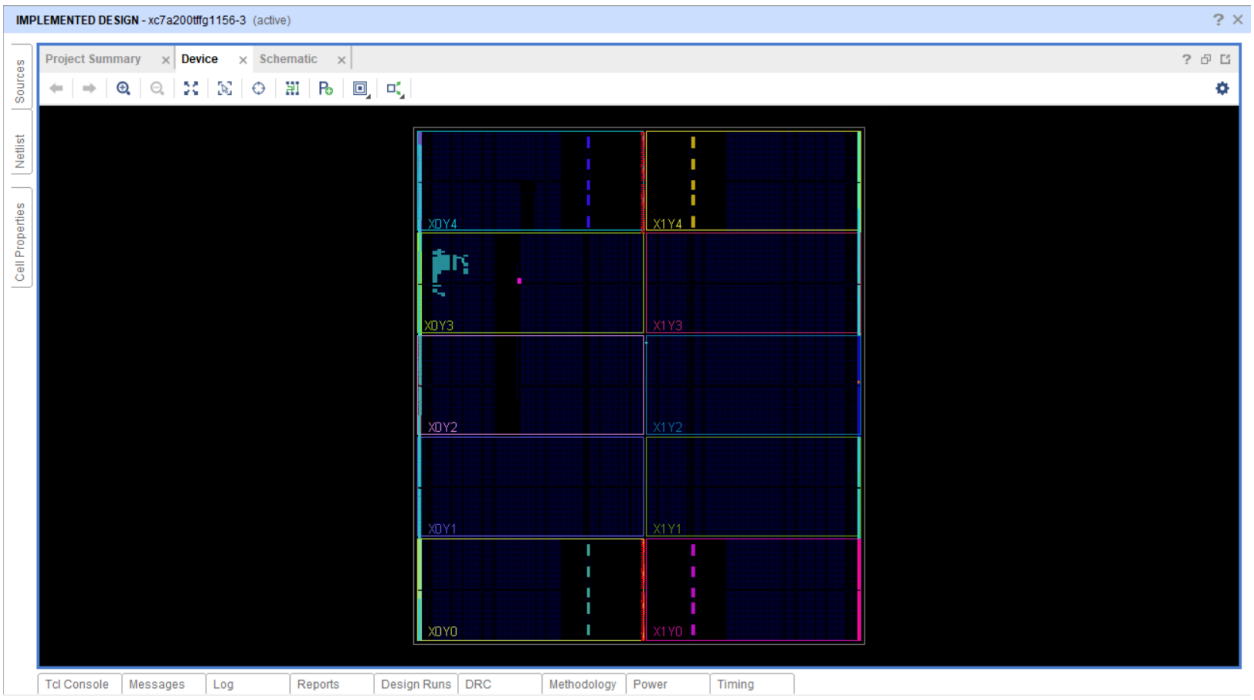
Utilization Report



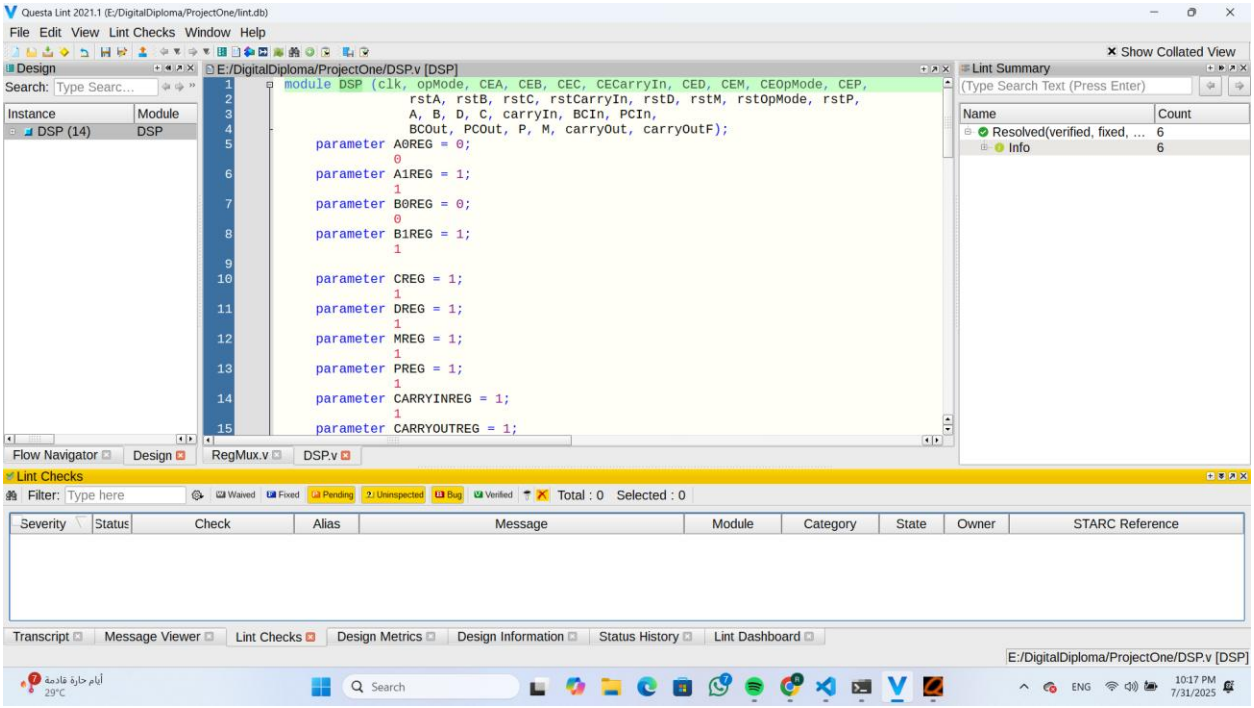
Timing Report



Device



Linting


Questa Lint 2021.1 (E:/DigitalDiploma/ProjectOne/lint.db)
File Edit View Lint Checks Window Help
Show Collated View
Design
Search: Type Searc...
Instance Module
DSP (14) DSP
E:/DigitalDiploma/ProjectOne/DSP.v [DSP]
1 module DSP (clk, opMode, CEA, CEB, CEC, CECarryIn, CED, CEM, CEOpMode, CEP,
2 rstA, rstB, rstC, rstCarryIn, rstD, rstM, rstOpMode, rstP,
3 A, B, D, C, carryIn, BCIn, PCIn,
4 BCOut, PCOut, P, M, carryOut, carryOutF);
5 parameter A0REG = 0;
0
6 parameter A1REG = 1;
1
7 parameter B0REG = 0;
0
8 parameter B1REG = 1;
1
9
10 parameter CREG = 1;
1
11 parameter DREG = 1;
1
12 parameter MREG = 1;
1
13 parameter PREG = 1;
1
14 parameter CARRYINREG = 1;
1
15 parameter CARRYOUTREG = 1;
Lint Summary
(Type Search Text (Press Enter)
Name Count
Resolved(verified, fixed, ... 6
Info 6
Flow Navigator Design RegMux.v DSP.v
Lint Checks
Filter: Type here
Waived Fixed Pending Uninspected Bug Verified
Total : 0 Selected : 0
Severity Status Check Alias Message Module Category State Owner STARC Reference
Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard
E:/DigitalDiploma/ProjectOne/DSP.v [DSP]
Search
ENG
10:17 PM
7/31/2025