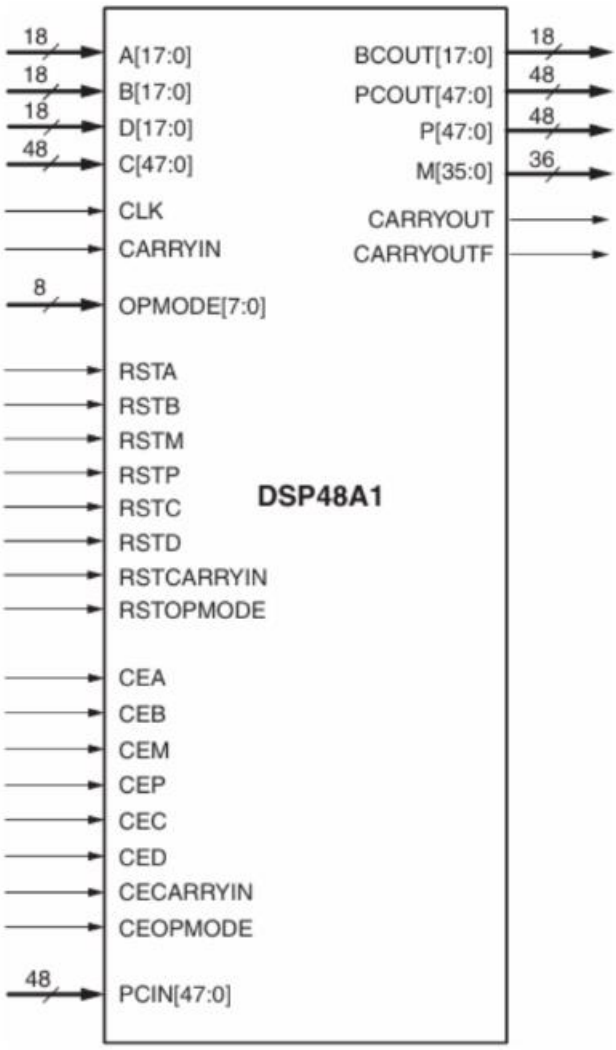
Spartan6 - DSP48A1



By: **Shahd Gamal Mahmoud**

Table of Contents

1. RTL Code

- BLOCK (REG + MUX)

```
1   module REG_MUX_pair (BLOCK_IN, SEL, CLK, RST, CE, BLOCK_OUT);
2   parameter RSTTYPE = "SYNC";
3   parameter WIDTH = 18;
4   input SEL,CLK,RST,CE;
5   input   [WIDTH-1:0] BLOCK_IN;
6   output  [WIDTH-1:0] BLOCK_OUT;
7   reg     [WIDTH-1:0] REG_OUT;
8
9   generate
10      if (RSTTYPE == "SYNC") begin
11          always @(posedge CLK) begin
12              if (RST) begin
13                  REG_OUT <= 0;
14              end
15              else if (CE) begin
16                  REG_OUT <= BLOCK_IN;
17              end
18          end
19      end
20      else if (RSTTYPE == "ASYNC") begin
21          always @(posedge CLK or posedge RST) begin
22              if (RST) begin
23                  REG_OUT <= 0;
24              end
25              else if (CE) begin
26                  REG_OUT<=BLOCK_IN;
27              end
28          end
29      end
30  endgenerate
31
32  assign BLOCK_OUT = (SEL)? REG_OUT : BLOCK_IN ;
33  endmodule
34
```

➢ Design DSP

```
`timescale 1ns/1ps

module DSP48A1 #(
    // Parameterized pipeline register enables (1 = register enabled, 0 = bypassed)
    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,

    // Functional parameters controlling carry selection and input mode
    parameter CARRYINSEL = "OPMODE5",  // Controls carry input selection
    parameter B_INPUT = "DIRECT",  // Selects direct B input or cascade from BCIN
    parameter RSTTYPE = "SYNC"  // Determines reset type (SYNC/ASYNC)
)(
    // Data inputs
    input  [17:0] A, B, D,
    input  [47:0] C,
    input  [47:0] PCIN,
    input  [17:0] BCIN,
    input  [7:0] OPMODE,

    // Data outputs
    output [35:0] M,
    output [47:0] P,
    output [17:0] BCOUT,
    output [47:0] PCOUT,

    // Clock, carry input, and control signals
    input  CLK,
    input  CARRYIN,
    input  CEA, CEB, CEC, CECARRYIN, CED, CEM, CEOPMODE, CEP,
    input  RSTA, RSTB, RSTC, RSTCARRYIN, RSTD, RSTM, RSTOPMODE, RSTP,

    // Carry outputs
    output CARRYOUT, CARRYOUTF
);

```

```
// ---------------------------------------
// Stage 1: Input Registers and Muxing
// ---------------------------------------
wire [17:0] D_BLOCK_OUT, B0_MUX_OUT, B0_BLOCK_OUT, A0_BLOCK_OUT;
wire [47:0] C_BLOCK_OUT;

// Select B input source (direct or cascade)
assign B0_MUX_OUT = (B_INPUT == "DIRECT") ? B : (B_INPUT == "CASCADE") ? BCIN : 18'b0;

// Register inputs if pipeline is enabled
REG_MUX_pair #(.WIDTH(18), .RSTTYPE(RSTTYPE)) D_BLOCK (.BLOCK_IN(D), .SEL(DREG), .CLK(CLK), .RST(RSTD), .CE(CED), .BLOCK_OUT(D_BLOCK_OUT));
REG_MUX_pair #(.WIDTH(18), .RSTTYPE(RSTTYPE)) B0_BLOCK (.BLOCK_IN(B0_MUX_OUT), .SEL(B0REG), .CLK(CLK), .RST(RSTB), .CE(CEB), .BLOCK_OUT(B0_BLOCK_OUT));
REG_MUX_pair #(.WIDTH(18), .RSTTYPE(RSTTYPE)) A0_BLOCK (.BLOCK_IN(A), .SEL(A0REG), .CLK(CLK), .RST(RSTA), .CE(CEA), .BLOCK_OUT(A0_BLOCK_OUT));
REG_MUX_pair #(.WIDTH(48), .RSTTYPE(RSTTYPE)) C_BLOCK (.BLOCK_IN(C), .SEL(CREG), .CLK(CLK), .RST(RSTC), .CE(CEC), .BLOCK_OUT(C_BLOCK_OUT));

```

```
60  // ---------------------------------------
61  // Stage 2: OPMODE Register and Pre-Adder
62  // ---------------------------------------
63  wire [7:0] OPMODE_BLOCK_OUT;
64  wire [17:0] PREADDER_OUT, PREADDER_MUX_OUT;
65
66  // Register OPMODE control signals
67  REG_MUX_pair #(.WIDTH(8), .RSTTYPE(RSTTYPE)) OPMODE_BLOCK (.BLOCK_IN(OPMODE), .SEL(OPMODEREG), .CLK(CLK), .RST(RSTOPMODE), .CE(CEOPMODE), .BLOCK_OUT(OPMODE_BLOCK_OUT));
68
69  // Pre-Adder operation based on OPMODE[6]
70  assign PREADDER_OUT = (OPMODE_BLOCK_OUT[6]) ? (D_BLOCK_OUT - B0_BLOCK_OUT) : (D_BLOCK_OUT + B0_BLOCK_OUT);
71
72  // Select whether to use the pre-adder output or bypass it
73  assign PREADDER_MUX_OUT = (OPMODE_BLOCK_OUT[4]) ? PREADDER_OUT : B0_BLOCK_OUT;
74
75  // ---------------------------------------
76  // Stage 3: Registering Pre-Adder Outputs
77  // ---------------------------------------
78  wire [17:0] B1_BLOCK_OUT, A1_BLOCK_OUT;
79
80  REG_MUX_pair #(.WIDTH(18), .RSTTYPE(RSTTYPE)) B1_BLOCK (.BLOCK_IN(PREADDER_MUX_OUT), .SEL(B1REG), .CLK(CLK), .RST(RSTB), .CE(CEB), .BLOCK_OUT(B1_BLOCK_OUT));
81  REG_MUX_pair #(.WIDTH(18), .RSTTYPE(RSTTYPE)) A1_BLOCK (.BLOCK_IN(A0_BLOCK_OUT), .SEL(A1REG), .CLK(CLK), .RST(RSTA), .CE(CEA), .BLOCK_OUT(A1_BLOCK_OUT));
82
```

```
83  // ---------------------------------------
84  // Stage 4: Multiplication and Carry Handling
85  // ---------------------------------------
86  wire [35:0] MUL_OUT, MUL_BLOCK_OUT;
87  wire CARRY_MUX_OUT, CIN;
88
89  // Cascade output assignment
90  assign BCOUT = B1_BLOCK_OUT;
91
92  // Perform 18-bit x 18-bit multiplication
93  assign MUL_OUT = B1_BLOCK_OUT * A1_BLOCK_OUT;
94
95  // Select carry input source
96  assign CARRY_MUX_OUT = (CARRYINSEL == "OPMODE5") ? OPMODE_BLOCK_OUT[5] : (CARRYINSEL == "CARRYIN") ? CARRYIN : 1'b0;
97
98  // Generate Block: Buffering Multiplication Output
99  generate
100     genvar i;
101     for(i = 0;i<36;i=i+1)
102         buf(M[i],MUL_BLOCK_OUT[i]);
103 endgenerate
104 //assign M = MUL_BLOCK_OUT;
105
106 // Register multiplication output and carry input
107 REG_MUX_pair #(.WIDTH(36), .RSTTYPE(RSTTYPE)) MUL_BLOCK (.BLOCK_IN(MUL_OUT), .SEL(MREG), .CLK(CLK), .RST(RSTM), .CE(CEM), .BLOCK_OUT(MUL_BLOCK_OUT));
108 REG_MUX_pair #(.WIDTH(1), .RSTTYPE(RSTTYPE)) CYI_BLOCK (.BLOCK_IN(CARRY_MUX_OUT), .SEL(CARRYINREG), .CLK(CLK), .RST(RSTCARRYIN), .CE(CECARRYIN), .BLOCK_OUT(CIN));
109
```

```
110 // ---------------------------------------
111 // Stage 5: Post-Adder and Final Output
112 // ---------------------------------------
113 wire [47:0] MUXX_OUT, D_A_B_CONCAT, MUXZ_OUT, POSTADDER_SUM;
114 wire POSTADDER_COUT;
115
116 // Assign output control signals
117 assign CARRYOUTF = CARRYOUT;
118 assign PCOUT = P;
119
120 // Concatenation of inputs for post-adder
121 assign D_A_B_CONCAT = {D[11:0], A[17:0], B[17:0]};
122
123 // Select input to post-adder
124 assign MUXX_OUT = (OPMODE_BLOCK_OUT[1:0] == 2'b00) ? 48'b0 :
125                   (OPMODE_BLOCK_OUT[1:0] == 2'b01) ? MUL_BLOCK_OUT :
126                   (OPMODE_BLOCK_OUT[1:0] == 2'b10) ? P : D_A_B_CONCAT;
127
128 assign MUXZ_OUT = (OPMODE_BLOCK_OUT[3:2] == 2'b00) ? 48'b0 :
129                   (OPMODE_BLOCK_OUT[3:2] == 2'b01) ? PCIN :
130                   (OPMODE_BLOCK_OUT[3:2] == 2'b10) ? P : C_BLOCK_OUT;
131
132 // Final addition/subtraction with carry-in
133 assign {POSTADDERCOUT, POSTADDER_SUM} = (OPMODE_BLOCK_OUT[7]) ?
134        (MUXZ_OUT - (MUXX_OUT + CIN)) : (MUXZ_OUT + MUXX_OUT + CIN);
135
136 // Registering final outputs
137 REG_MUX_pair #(.WIDTH(1), .RSTTYPE(RSTTYPE)) CYO_BLOCK (.BLOCK_IN(POSTADDERCOUT), .SEL(CARRYOUTREG), .CLK(CLK), .RST(RSTCARRYIN), .CE(CECARRYIN), .BLOCK_OUT(CARRYOUT));
138 REG_MUX_pair #(.WIDTH(48), .RSTTYPE(RSTTYPE)) P_BLOCK (.BLOCK_IN(POSTADDER_SUM), .SEL(PREG), .CLK(CLK), .RST(RSTP), .CE(CEP), .BLOCK_OUT(P));
139
140 endmodule
```

2. Testbench Code

```
`timescale 1ns/1ps

module DSP48A1_tb();

// ------------------------------
// Parameter Definitions
// ------------------------------
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";  // Determines carry input selection
parameter B_INPUT = "DIRECT";  // Specifies whether B input is direct or cascaded
parameter RSTTYPE = "SYNC";  // Specifies whether resets are synchronous or asynchronous

// ------------------------------
// Signal Declarations
// ------------------------------
reg [17:0] A, B, D;
reg [47:0] C, PCIN;
reg [17:0] BCIN;
reg CLK, CARRYIN;
reg [7:0] OPMODE;
reg RSTA, RSTB, RSTM, RSTP, RSTC, RSTD, RSTCARRYIN, RSTOPMODE;
reg CEA, CEB, CEM, CEP, CEC, CED, CECARRYIN, CEOPMODE;

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

```
// ------------------------------
// Device Under Test (DUT) Instantiation
// ------------------------------
DSP48A1 #(
    .A0REG(A0REG), .A1REG(A1REG), .B0REG(B0REG), .B1REG(B1REG),
    .CREG(CREG), .DREG(DREG), .MREG(MREG), .PREG(PREG),
    .CARRYINREG(CARRYINREG), .CARRYOUTREG(CARRYOUTREG),
    .OPMODEREG(OPMODEREG), .CARRYINSEL(CARRYINSEL), .B_INPUT(B_INPUT), .RSTTYPE(RSTTYPE)
) DUT (
    .A(A), .B(B), .D(D), .C(C), .CLK(CLK), .CARRYIN(CARRYIN), .OPMODE(OPMODE),
    .BCIN(BCIN), .RSTA(RSTA), .RSTB(RSTB), .RSTM(RSTM), .RSTP(RSTP),
    .RSTC(RSTC), .RSTD(RSTD), .RSTCARRYIN(RSTCARRYIN), .RSTOPMODE(RSTOPMODE),
    .CEA(CEA), .CEB(CEB), .CEM(CEM), .CEP(CEP), .CEC(CEC), .CED(CED),
    .CECARRYIN(CECARRYIN), .CEOPMODE(CEOPMODE), .PCIN(PCIN),
    .BCOUT(BCOUT), .PCOUT(PCOUT), .P(P), .M(M), .CARRYOUT(CARRYOUT), .CARRYOUTF(CARRYOUTF)
);

// ------------------------------
// Clock Generation
// ------------------------------
initial begin
    CLK = 0;
    forever #1 CLK = ~CLK;  // Generates a clock signal with a period of 2 ns
end
```

```
64  // ------------------------------
65  // Test Stimulus Generator
66  // ------------------------------
67  initial begin
68      // Initialize and assert reset signals
69      RSTA = 1; RSTB = 1; RSTM = 1;       RSTP = 1;
70      RSTC = 1; RSTD = 1; RSTCARRYIN = 1; RSTOPMODE = 1;
71      CEA = 1;  CEB = 1;  CEM = 1;        CEP = 1;
72      CEC = 1;  CED = 1;  CECARRYIN = 1;  CEOPMODE = 1;
73      A = 0;    B = 0;    C = 0;          D = 0;
74      CARRYIN = 0;        BCIN = 0;       PCIN = 0;
75      OPMODE = 8'b00000000;
76
77      repeat(5) @(negedge CLK);
78
79      // Release resets
80      RSTA = 0; RSTB = 0; RSTM = 0;       RSTP = 0;
81      RSTC = 0; RSTD = 0; RSTCARRYIN = 0; RSTOPMODE = 0;
82
83      // ------------------------------
84      // Test Cases
85      // ------------------------------
86
87      // Test Case 1: Basic addition
88      A = 20; B = 50; C = 10; D = 100; CARRYIN = 0; BCIN = 5; PCIN = 40;
89      OPMODE = 8'b01101111; // A + B + D + C
90      repeat(5) @(negedge CLK);
91
92      // Test Case 2: Basic subtraction
93      A = 80; B = 10; C = 10; D = 10; CARRYIN = 0;
94      OPMODE = 8'b01010100; // A - B - D - C
95      repeat(5) @(negedge CLK);
96
97      // Test Case 3: Multiplication with addition
98      A = 15; B = 10; C = 100; D = 5; CARRYIN = 0;
99      OPMODE = 8'b00101010; // A * B + D + C
100     repeat(5) @(negedge CLK);
101
102     // Test Case 4: Accumulation with CARRYIN
103     A = 10; B = 20; C = 50; D = 10; CARRYIN = 1;
104     OPMODE = 8'b10001101; // A + B + D + CARRYIN
105     repeat(5) @(negedge CLK);
```

```
107     // Test Case 5: Chained operations with different OPMODE values
108     A = 50; B = 25; C = 200; D = 10; CARRYIN = 0;
109     OPMODE = 8'b01111111; // Complex operation
110     repeat(5) @(negedge CLK);
111
112     // Test Case 6: Cascade input (BCIN) handling
113     A = 60; B = 30; C = 100; D = 10; CARRYIN = 0; BCIN = 10;
114     OPMODE = 8'b11000010; // Perform operation using BCIN
115     repeat(5) @(negedge CLK);
116
117     // Test Case 7: Complex operation using all inputs
118     A = 70; B = 40; C = 150; D = 20; CARRYIN = 1;
119     OPMODE = 8'b10101010; // Another complex operation
120     repeat(5) @(negedge CLK);
121
122     // Test Case 8: Another OPMODE setting with all inputs
123     A = 80; B = 50; C = 175; D = 25; CARRYIN = 0;
124     OPMODE = 8'b00011000; // Another operation mode
125     repeat(5) @(negedge CLK);
126
127     // Test Case 9: Overflow scenario (max values)
128     A = 18'h3FFFF; B = 18'h3FFFF; C = 48'hFFFFFFFFFFFF; D = 18'h3FFFF;
129     CARRYIN = 1;
130     OPMODE = 8'b10000000; // Checking max value addition
131     repeat(5) @(negedge CLK);
```

```
132
133         // Test Case 10: Zero Inputs (A, B, C, D all zero)
134         A = 0; B = 0; C = 0; D = 0; CARRYIN = 0;
135         OPMODE = 8'b00001111; // Testing zero addition
136         repeat(5) @(negedge CLK);
137
138         // Test Case 11: Negative values (2's complement simulation)
139         A = -10; B = -5; C = -50; D = -20; CARRYIN = 1;
140         OPMODE = 8'b11100001; // Negative accumulation
141         repeat(5) @(negedge CLK);
142
143         // Test Case 12: Carry propagation
144         A = 50; B = 25; C = 200; D = 10; CARRYIN = 1;
145         OPMODE = 8'b10001101; // Carry should propagate
146         repeat(5) @(negedge CLK);
147
148         // Test Case 13: Cascade input (BCIN) handling
149         A = 60; B = 30; C = 100; D = 10; CARRYIN = 0; BCIN = 10;
150         OPMODE = 8'b11000010; // Perform operation using BCIN
151         repeat(5) @(negedge CLK);
152
153         // Test Case 14: Different shift and accumulation
154         A = 12; B = 5; C = 60; D = 4; CARRYIN = 0;
155         OPMODE = 8'b10101010; // Shift-accumulate
156         repeat(5) @(negedge CLK);
```

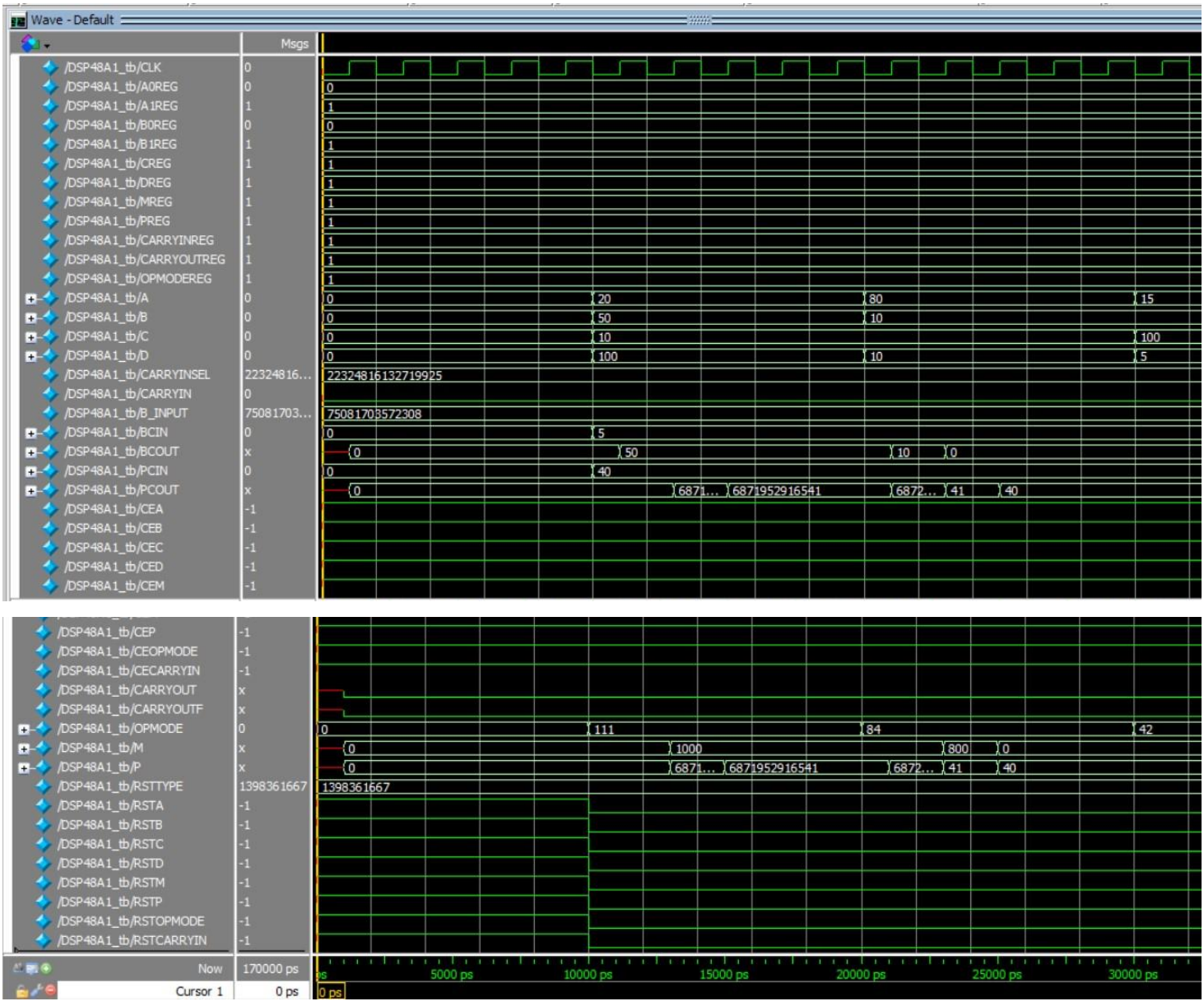
```
158     // ------------------------------
159     // Reset and Finish Simulation
160     // ------------------------------
161     RSTA = 1; RSTB = 1; RSTM = 1;       RSTP = 1;
162     RSTC = 1; RSTD = 1; RSTCARRYIN = 1; RSTOPMODE = 1;
163     CEA = 0;  CEB = 0;  CEM = 0;        CEP = 0;
164     CEC = 0;  CED = 0;  CECARRYIN = 0;  CEOPMODE = 0;
165     A = 0;    B = 0;    C = 0;          D = 0;
166     CARRYIN = 0;        BCIN = 0;       PCIN = 0;
167     OPMODE = 8'b00000000;
168
169     repeat(10) @(negedge CLK);
170     $stop;  // Stop simulation
171 end
172
173 // ------------------------------
174 // Test Monitor and Debugging
175 // ------------------------------
176 initial begin
177     $monitor("Time=%t | A=%d, B=%d, C=%d, D=%d, CARRYIN=%d, PCIN=%d, OPMODE=%b, P=%d, BCOUT=%d, M=%d, CARRYOUT=%d",
178              $time, A, B, C, D, CARRYIN, PCIN, OPMODE, P, BCOUT, M, CARRYOUT);
179 end
180
181 endmodule
```

3. Do File

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

4. QuestaSim Snippets

- Waves

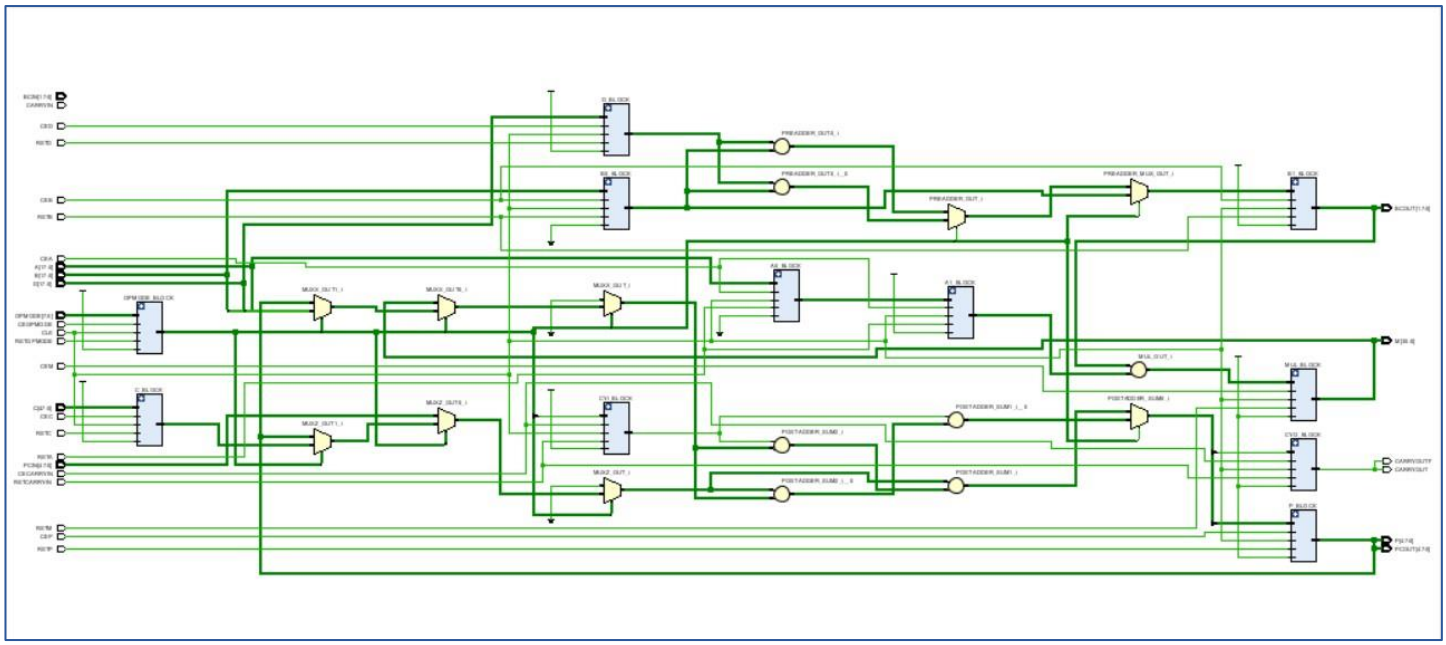


5. Constraint File

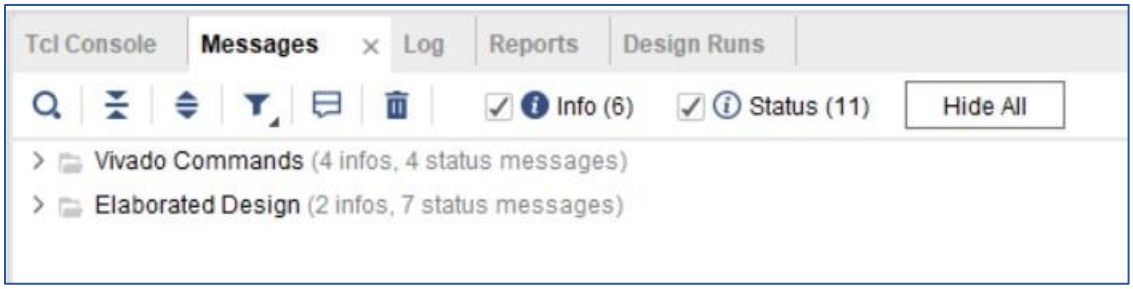
```
6   ## Clock signal
7   set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
8   create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9
```

6. Elaboration

- Schematic Snippets

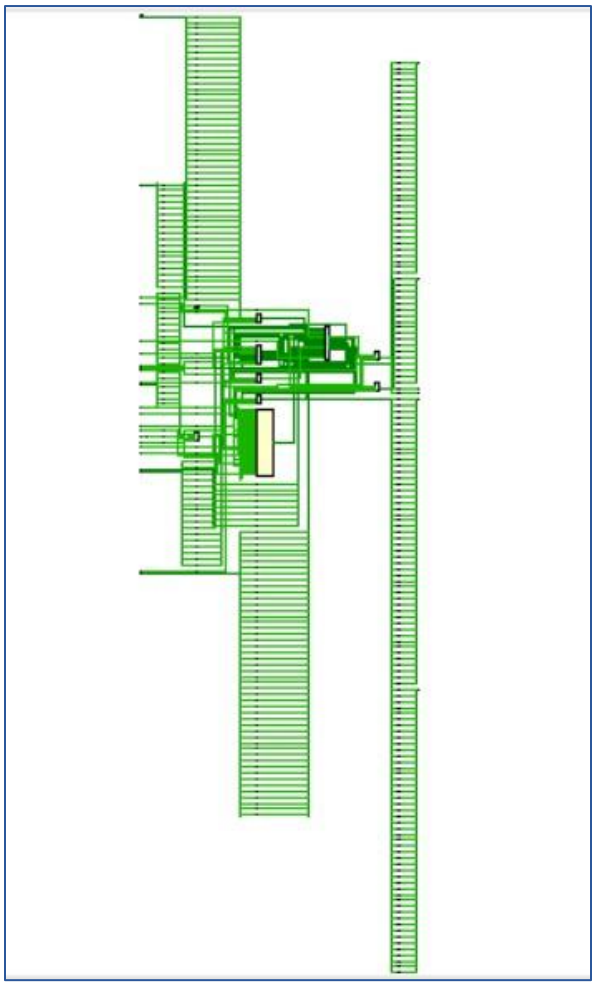


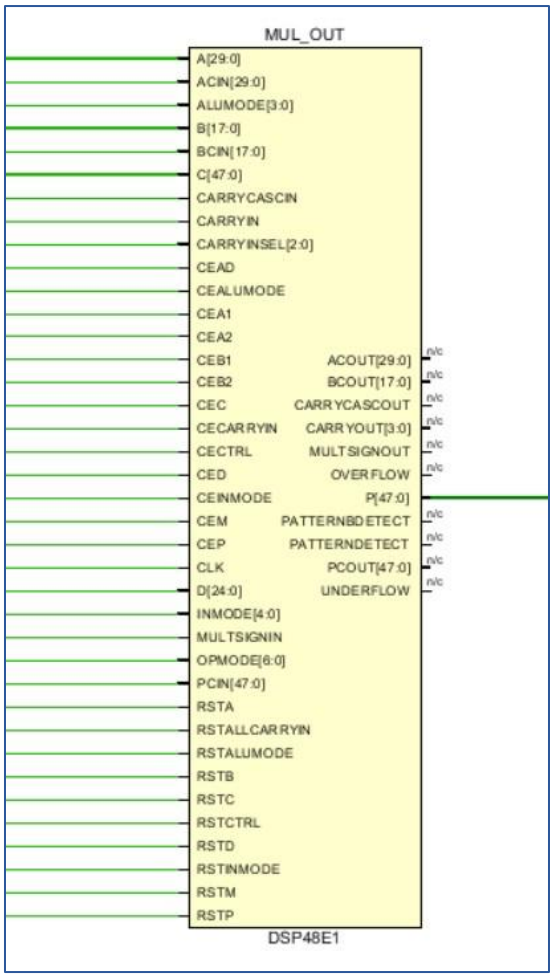
- Messages (No Errors or Warnings)



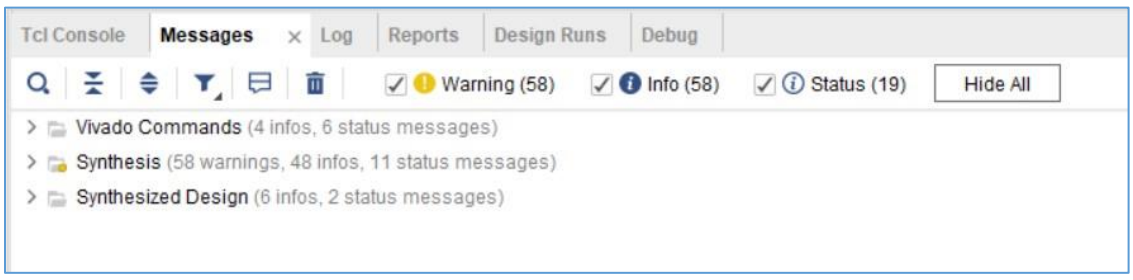
7. Synthesis

- Schematic Snippets





- Messages (No Errors)



➢ Utilization report

Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP48A1	230	143	1	327	1
P_BLOCK (REG_MUX_pair__parameterized0_3)	0	48	0	0	0
OPMODE_BLOCK (REG_MUX_pair__parameterized1)	228	8	0	0	0
D_BLOCK (REG_MUX_pair_2)	0	18	0	0	0
C_BLOCK (REG_MUX_pair__parameterized0)	0	48	0	0	0
CYO_BLOCK (REG_MUX_pair__parameterized3_1)	0	1	0	0	0
CYI_BLOCK (REG_MUX_pair__parameterized3)	1	1	0	0	0
B1_BLOCK (REG_MUX_pair_0)	0	18	0	0	0
A1_BLOCK (REG_MUX_pair)	0	1	0	0	0

➢ Time report

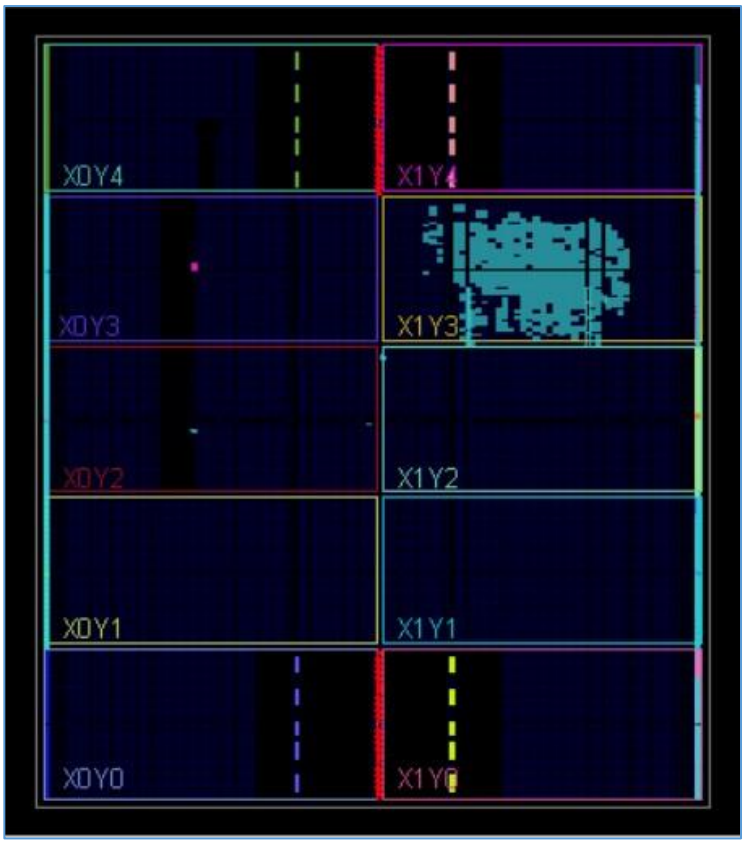
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.168 ns	Worst Hold Slack (WHS):	0.182 ns	Worst Pulse Width Slack (WPWS):	4.500 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:	106	Total Number of Endpoints:	106	Total Number of Endpoints:	145

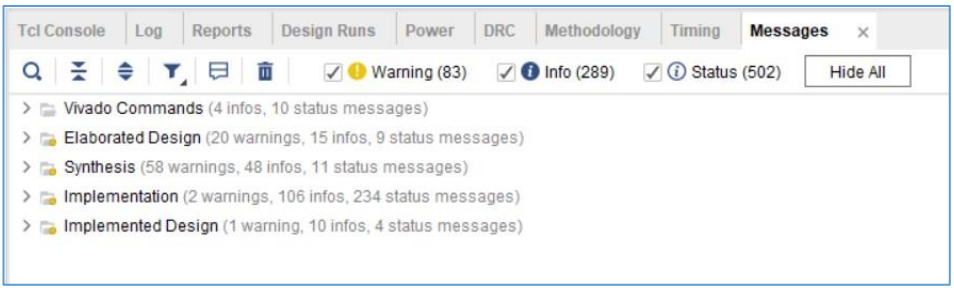
All user specified timing constraints are met.

8. Implementation

- Device Snippets



- Messages (No Errors)



➢ Utilization report

Name	Slice LUTs (133800)	Slice Registers (267600)	F7 Muxes (66900)	F8 Muxes (33450)	Slice (33450)	LUT as Logic (133800)	LUT as Memory (46200)	LUT Flip Flop Pairs (133800)	Block RAM Tile (365)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)	BSCANE2 (4)
DSP48A1	2701	4208	97	11	1362	2227	474	1583	8	1	327	2	1
A1_BLOCK (REG_MU...	0	1	0	0	1	0	0	0	0	0	0	0	0
B1_BLOCK (REG_MU...	0	18	0	0	6	0	0	0	0	0	0	0	0
C_BLOCK (REG_MUX...	0	48	0	0	18	0	0	0	0	0	0	0	0
CYI_BLOCK (REG_MU...	1	1	0	0	1	1	0	1	0	0	0	0	0
CYO_BLOCK (REG_M...	0	1	0	0	1	0	0	0	0	0	0	0	0
D_BLOCK (REG_MUX...	0	18	0	0	10	0	0	0	0	0	0	0	0
dbg_hub (dbg_hub)	475	727	0	0	236	451	24	306	0	0	0	1	1
OPMODE_BLOCK (RE...	228	8	0	0	74	228	0	0	0	0	0	0	0
P_BLOCK (REG_MUX...	0	48	0	0	12	0	0	0	0	0	0	0	0
u_ila_0 (u_ila_0)	1996	3338	97	11	1054	1546	450	1220	8	0	0	0	0

➢ Time report

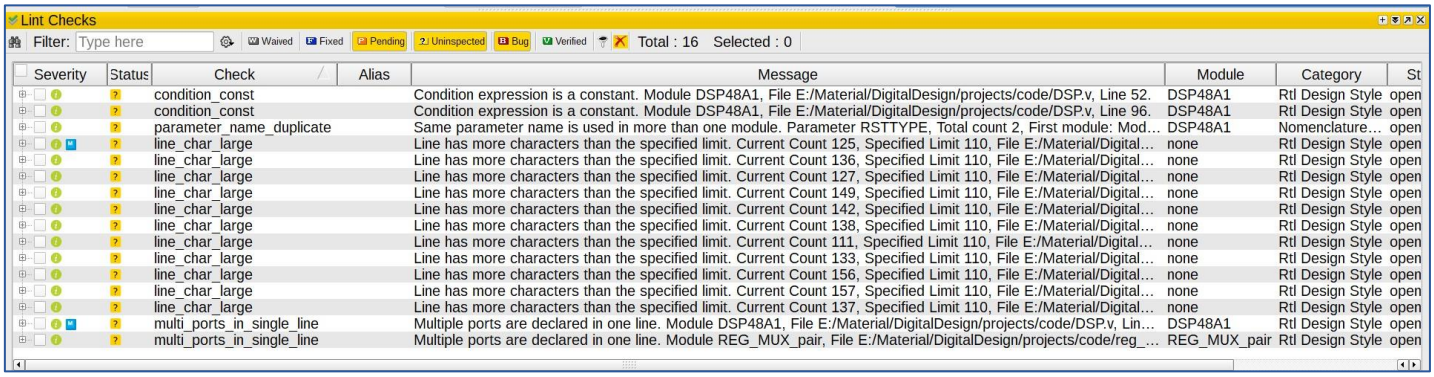
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.623 ns	Worst Hold Slack (WHS):	0.060 ns	Worst Pulse Width Slack (WPWS):	3.950 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:	8069	Total Number of Endpoints:	8053	Total Number of Endpoints:	5119

All user specified timing constraints are met.

9. Linting

- Lint Checks (No Errors or Warnings)



Severity	Status	Check	Alias	Message	Module	Category	St
		condition_const		Condition expression is a constant. Module DSP48A1, File E:/Material/DigitalDesign/projects/code/DSP.v, Line 52.	DSP48A1	Rtl Design Style	open
		condition_const		Condition expression is a constant. Module DSP48A1, File E:/Material/DigitalDesign/projects/code/DSP.v, Line 96.	DSP48A1	Rtl Design Style	open
		parameter_name_duplicate		Same parameter name is used in more than one module. Parameter RSTTYPE, Total count 2, First module: Mod…	DSP48A1	Nomenclature…	open
		line_char_large		Line has more characters than the specified limit. Current Count 125, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 136, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 127, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 149, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 142, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 138, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 111, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 133, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 156, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 157, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		line_char_large		Line has more characters than the specified limit. Current Count 137, Specified Limit 110, File E:/Material/Digital…	none	Rtl Design Style	open
		multi_ports_in_single_line		Multiple ports are declared in one line. Module DSP48A1, File E:/Material/DigitalDesign/projects/code/DSP.v, Lin…	DSP48A1	Rtl Design Style	open
		multi_ports_in_single_line		Multiple ports are declared in one line. Module REG_MUX_pair, File E:/Material/DigitalDesign/projects/code/reg_…	REG_MUX_pair	Rtl Design Style	open