

Digital IC Design

DSP Project

Sparten6 – DSP48A1



Prepared by: *Omar Ahmed Abdelaty*

Submitted to: *Eng. Kareem Waseem*

omarahmedabdelaty231@gmail.com

Table of Contents

□ Sparten6 – DSP48A1	
• Overview	4
• Diagrams	5
□ RTL Code	
• Pipeline	6
• DSP	7
□ Testbench	
• TB Code	9
• DO File	11
□ Functional Verification	
• Reset Operation	12
• Simulation Flow (Path 1)	13
• Simulation Flow (Path 2)	14
• Simulation Flow (Path 3)	15
• Simulation Flow (Path 4)	16
• Transcript Output	17
□ Constraints File (.xdc)	18

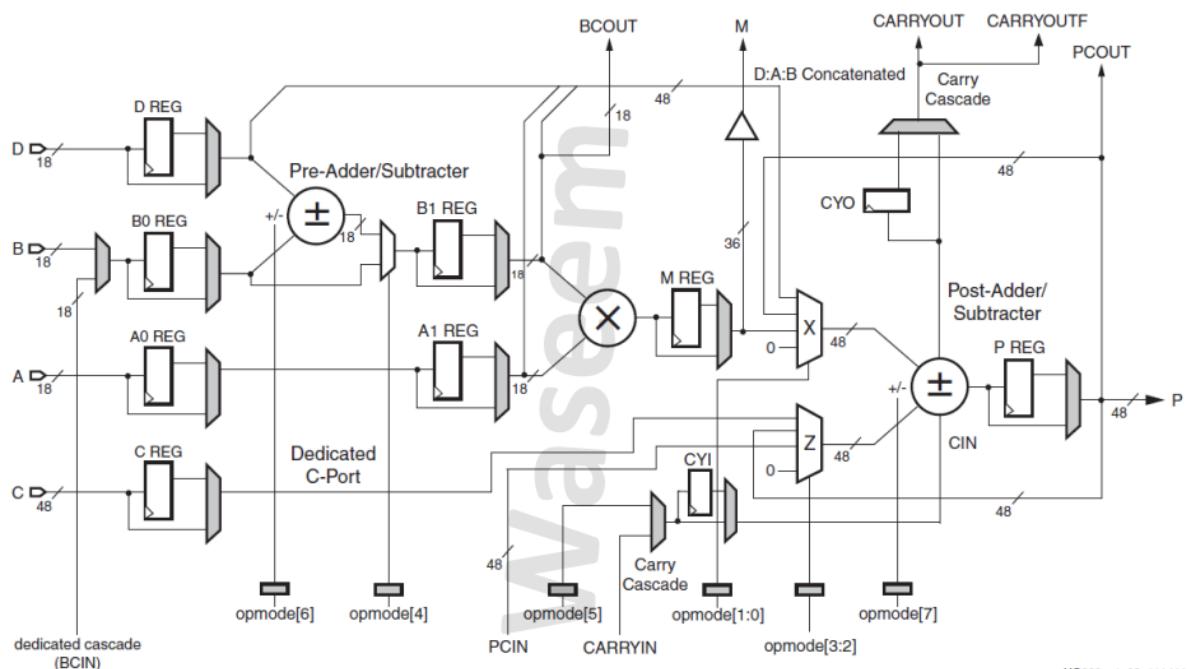
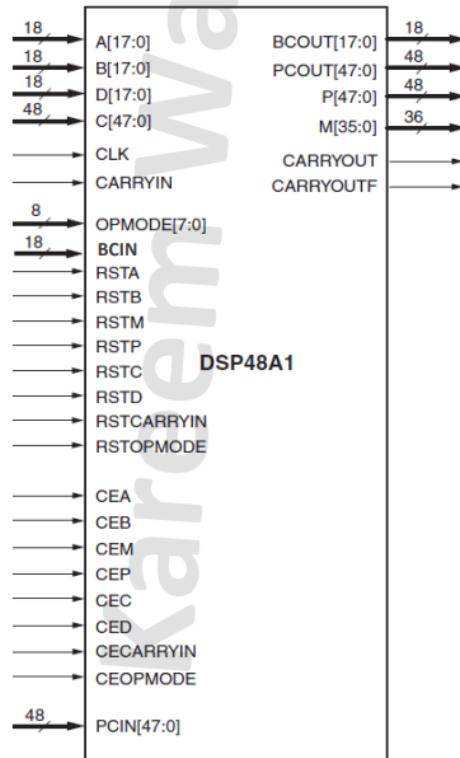
□ Elaboration	
• Schematic	19
• Messages	19
□ Synthesis	
• Schematic	20
• Messages	21
• Utilization Report	21
• Timing Report	21
• Power Report	22
• Worst Slack	23
□ Implementation	
• Device View	24
• Messages	25
• Utilization Report	25
• Timing Report	25
• Power Report	26
• Worst Slack	26
□ Linting	27

Sparten6 – DSP48A1

Overview

The Spartan-6 FPGA family from Xilinx is known for offering a cost-effective balance of logic resources, memory, and digital signal processing (DSP) capabilities. It features the DSP48A1 slice — a highly versatile and efficient arithmetic processing block that is well-suited for computationally intensive operations such as multiply-accumulate (MAC), filtering, and signal modulation. The DSP48A1 slice is especially optimized for applications in digital signal processing, control systems, wireless communication, and image processing, where speed and arithmetic throughput are critical.

Diagrams



UG389_c1_03_111411

RTL Code

Pipeline

```
1  module pipeline(D, clk, enable, reset, Q);
2    parameter F = 18;
3    input [F-1:0] D;
4    input clk, enable, reset;
5    output reg [F-1:0] Q;
6    parameter RSTTYPE = "SYNC";
7
8    generate
9      if (RSTTYPE == "SYNC") begin
10        always @(posedge clk) begin
11          if (reset) Q <= 0;
12          else if (enable) Q <= D;
13        end
14      end else begin
15        always @(posedge clk or posedge reset) begin
16          if (reset) Q <= 0;
17          else if (enable) Q <= D;
18        end
19      end
20    endgenerate
21  endmodule
```

DSP

```
 1 module DSP_TOP(
 2   A, B, BCIN, BCOUT, C, D, carryin, M, clk, opmode, P, PCIN, PCOUT,
 3   carryout, carryoutF, RSTA, RSTB, RSTC, RSTD, RSTM, RSTCARRYIN,
 4   RSTOPMODE, RSTP, CEA, CEB, CEC, CECARRYIN, CED, CEM, CEP, CEOPMODE
 5 );
 6
 7   input RSTA, RSTB, RSTC, RSTD, RSTM, RSTCARRYIN, RSTOPMODE, RSTP;
 8   input CEA, CEB, CEC, CECARRYIN, CED, CEM, CEP, CEOPMODE;
 9   input [17:0] A, B, D, BCIN;
10   input [47:0] C;
11   input clk, carryin;
12   input [7:0] opmode;
13   input [47:0] PCIN;
14
15   output reg [35:0] M;
16   output reg [47:0] P, PCOUT;
17   output reg carryout, carryoutF;
18   output reg [17:0] BCOUT;
19
20   parameter A0REG = 0, A1REG = 1, B0REG = 0, B1REG = 1;
21   parameter CREG = 1, DREG = 1, MREG = 1, PREG = 1;
22   parameter CARRYINREG = 1, CARRYOUTREG = 1, OPMODEREG = 1;
23   parameter CARRYINSEL = "OPMODE[5]";
24   parameter B_INPUT = "DIRECT";
25   parameter RSTTYPE = "SYNC";
26
27   reg [17:0] A_out, A_out2, D_out, B_out, B_out2, B_out3, B2;
28   reg [47:0] out_mux_x, out_mux_z;
29   reg CIN, COUT;
30   reg [47:0] post_adder_out;
31   reg [7:0] opmode_out;
32   reg Carry_Cascade_out;
33   reg [47:0] C_out;
34   reg [35:0] M_out, M_out_mult;
35
36   wire [17:0] A1_out, A1_out2, D1_out, B1_out, B1_out3;
37   wire CIN1, carryout1;
38   wire [35:0] M1_out;
39   wire [47:0] P1, C1_out;
40   wire [7:0] opmode1;
41
42   pipeline #(8) m0 (.reset(RSTOPMODE), .clk(clk), .enable(CEOPMODE), .Q(opmode1), .D(opmode));
43   pipeline m1 (.reset(RSTA), .clk(clk), .enable(CEA), .Q(A1_out), .D(A));
44   pipeline m2 (.reset(RSTA), .clk(clk), .enable(CEA), .Q(A1_out2), .D(A_out));
45   pipeline m3 (.reset(RSTD), .clk(clk), .enable(CED), .Q(D1_out), .D(D));
46   pipeline m4 (.reset(RSTB), .clk(clk), .enable(CEB), .Q(B1_out), .D(B2));
47   pipeline m5 (.reset(RSTB), .clk(clk), .enable(CEB), .Q(B1_out3), .D(B_out2));
48   pipeline #(36) m6 (.reset(RSTM), .clk(clk), .enable(CEM), .Q(M1_out), .D(M_out_mult));
49   pipeline #(48) m7 (.reset(RSTC), .clk(clk), .enable(CEC), .Q(C1_out), .D(C));
50   pipeline #(1) m8 (.reset(RSTCARRYIN), .clk(clk), .enable(CECARRYIN), .Q(CIN1), .D(Carry_Cascade_out));
51   pipeline #(1) m9 (.reset(RSTCARRYIN), .clk(clk), .enable(CECARRYIN), .Q(carryout1), .D(COUT));
52   pipeline #(48) m10 (.reset(RSTP), .clk(clk), .enable(CEP), .Q(P1), .D(post_adder_out));
53
```

```

1  always @(*) begin
2    opmode_out = OPMODEREG ? opmode1 : opmode;
3    A_out     = A0REG ? A1_out : A;
4    A_out2   = A1REG ? A1_out2 : A_out;
5    D_out     = DREG ? D1_out : D;
6
7    case (B_INPUT)
8      "DIRECT": B2 = B;
9      "CASCADE": B2 = BCIN;
10     default: B2 = 0;
11   endcase
12
13  B_out = B0REG ? B1_out : B2;
14
15  if (opmode_out[4] == 1) begin
16    case (opmode_out[6])
17      1'b0: B_out2 = D_out + B_out;
18      1'b1: B_out2 = D_out - B_out;
19    endcase
20  end else begin
21    B_out2 = B_out;
22  end
23
24  B_out3 = B1REG ? B1_out3 : B_out2;
25  BCOUT = B_out3;
26  M_out_mult = B_out3 * A_out2;
27  M_out = MREG ? M1_out : M_out_mult;
28  M = ~(~M_out);
29
30  case (opmode_out[1:0])
31    2'b00: out_mux_x = 0;
32    2'b01: out_mux_x = { {12{M_out[35]}}, M_out };
33    2'b10: out_mux_x = P1;
34    2'b11: out_mux_x = {D_out[11:0], A, B};
35  endcase
36
37  C_out = CREG ? C1_out : C;
38
39  case (opmode_out[3:2])
40    2'b00: out_mux_z = 0;
41    2'b01: out_mux_z = PCIN;
42    2'b10: out_mux_z = P1;
43    2'b11: out_mux_z = C_out;
44  endcase
45
46  Carry_Cascade_out = (CARRYINSEL == "OPMODE[5]") ? opmode_out[5] : (CARRYINSEL == "CARRYIN") ? carryin : 0;
47  CIN = CARRYINREG ? CIN1 : Carry_Cascade_out;
48
49  if (opmode_out[7])
50    {COUT, post_adder_out} = out_mux_z - (out_mux_x + CIN);
51  else
52    {COUT, post_adder_out} = out_mux_z + out_mux_x;
53
54  carryout = CARRYOUTREG ? carryout1 : COUT;
55  carryoutF = carryout;
56  P = PREG ? P1 : post_adder_out;
57  PCOUT = P;
58  end
59
60 endmodule
61

```

Testbench

TB Code

```
1  module DSP_TB();
2
3  // Reset signals
4  reg RSTA, RSTB, RSTC, RSTD, RSTM, RSTCARRYIN, RSTOPMODE, RSTP;
5  // Enable signals
6  reg CEA, CEB, CEC, CECARRYIN, CED, CEM, CEP, CEOPMODE;
7  // Inputs
8  reg [17:0] A, B, D, BCIN;
9  reg [47:0] C, PCIN;
10 reg clk, carryin;
11 reg [7:0] opmode;
12 // Outputs
13 wire [35:0] M;
14 wire [47:0] P, PCOUT;
15 wire carryout, carryoutF;
16 wire [17:0] BCOUT;
17
18 DSP_TOP dut (
19   A, B, BCIN, BCOUT, C, D, carryin, M, clk, opmode, P, PCIN, PCOUT,
20   carryout, carryoutF, RSTA, RSTB, RSTC, RSTD, RSTM, RSTCARRYIN,
21   RSTOPMODE, RSTP, CEA, CEB, CEC, CECARRYIN, CED, CEM, CEP, CEOPMODE
22 );
23
24 initial begin clk = 0; forever #1 clk = ~clk; end
25
26 task display_values;
27 begin
28   $display("Time = %0t", $time);
29   $display("Inputs: A=%0d, B=%0d, D=%0d, C=%0d, PCIN=%0d, carryin=%0b", A, B, D, C, PCIN, carryin);
30   $display("OPMODE = %b", opmode);
31   $display("Outputs: M=%0h, P=%0h, PCOUT=%0h, carryout=%0h, carryoutF=%0h, BCOUT=%0h", M, P, PCOUT, carryout, carryoutF, BCOUT);
32 end
33 endtask
34
35 initial begin
36   RSTA=1; RSTB=1; RSTC=1; RSTD=1; RSTM=1; RSTCARRYIN=1; RSTOPMODE=1; RSTP=1;
37   CEA=1; CEB=1; CEC=1; CECARRYIN=1; CED=1; CEM=1; CEP=1; CEOPMODE=1;
38   A=$random; B=$random; D=$random; C=$random; PCIN=$random; carryin=$random; opmode=8'b00000000;
39   #5 @(posedge clk);
40   display_values();
41   if (M !== 0 || P !== 0 || PCOUT !== 0 || BCOUT !== 0 || carryout !== 0 || carryoutF !== 0)
42   | $display("Reset FAIL");
43   else
44   | $display("Reset PASS
45   | -----");
```

```

49      RSTA=0; RSTB=0; RSTC=0; RSTD=0; RSTM=0; RSTCARRYIN=0; RSTOPMODE=0; RSTP=0;
50
51 // PATH 1: pre-sub, post-sub
52 A=20; B=10; D=25; C=350; PCIN=0; carryin=0; BCIN=0; opmode=8'b11011101;
53 #10 repeat(4) @(negedge clk);
54 display_values();
55 if (BCOUT === 18'hf && M === 36'h12c && P === 48'h32 && carryout == 0)
56   $display("Path 1 PASS
57   -----");
58 else
59   $display("Path 1 FAIL");
60
61 // PATH 2: pre-add, post-add
62 A=20; B=10; D=25; C=350; PCIN=0; carryin=0; BCIN=0; opmode=8'b00010000;
63 #10 repeat(4) @(negedge clk);
64 display_values();
65 if (BCOUT === 18'h23 && M === 36'h2bc && P === 0 && carryout == 0)
66   $display("Path 2 PASS
67   -----");
68 else
69   $display("Path 2 FAIL");
70
71 // PATH 3: P feedback
72 A=20; B=10; D=25; C=350; PCIN=0; carryin=0; BCIN=0; opmode=8'b00001010;
73 #10 repeat(4) @(negedge clk);
74 display_values();
75 if (BCOUT === 18'ha && M === 36'hc8)
76   $display("Path 3 PASS
77   -----");
78 else
79   $display("Path 3 FAIL");
80
81 // PATH 4: DAB concat + subtract
82 A=5; B=6; D=25; C=350; PCIN=3000; carryin=0; BCIN=0; opmode=8'b10100111;
83 #10 repeat(4) @(negedge clk);
84 display_values();
85 if (BCOUT === 18'd6 && M === 36'h1e && carryout == 1)
86   $display("Path 4 PASS
87   -----");
88 else
89   $display("Path 4 FAIL");
90   $stop;
91 end
92 endmodule

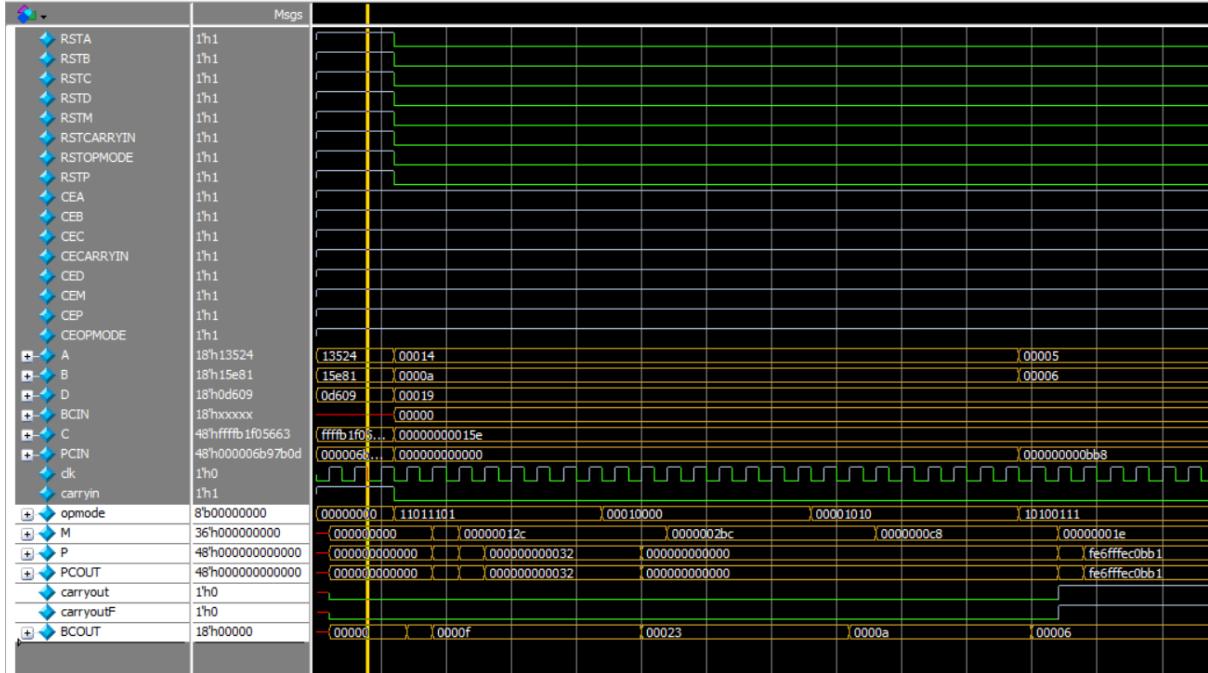
```

DO File

```
1 vlib work
2 vlog DSP_TOP.v DSP_TB.v
3 vsim -voptargs=+acc work.DSP_TB
4 add wave *
5 run -all
6 #quit -sim
7
```

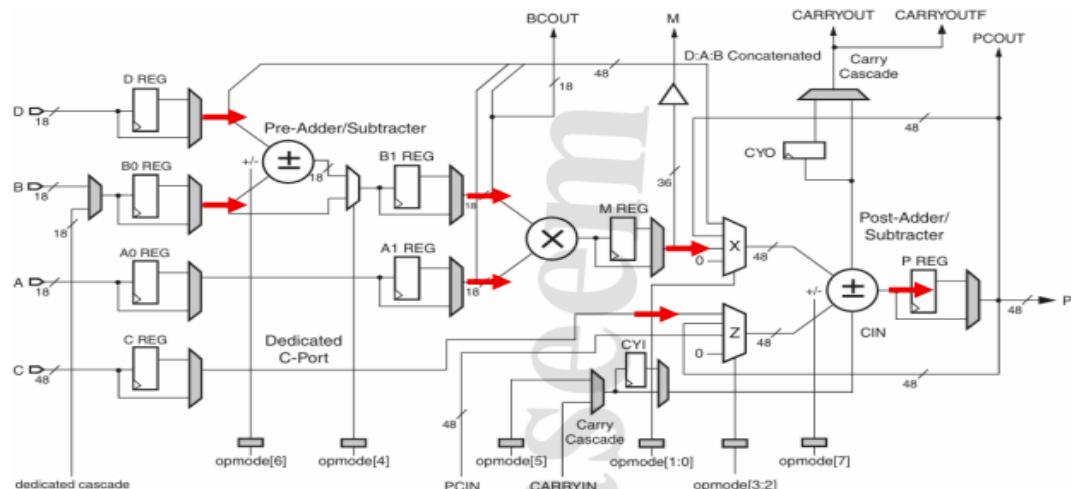
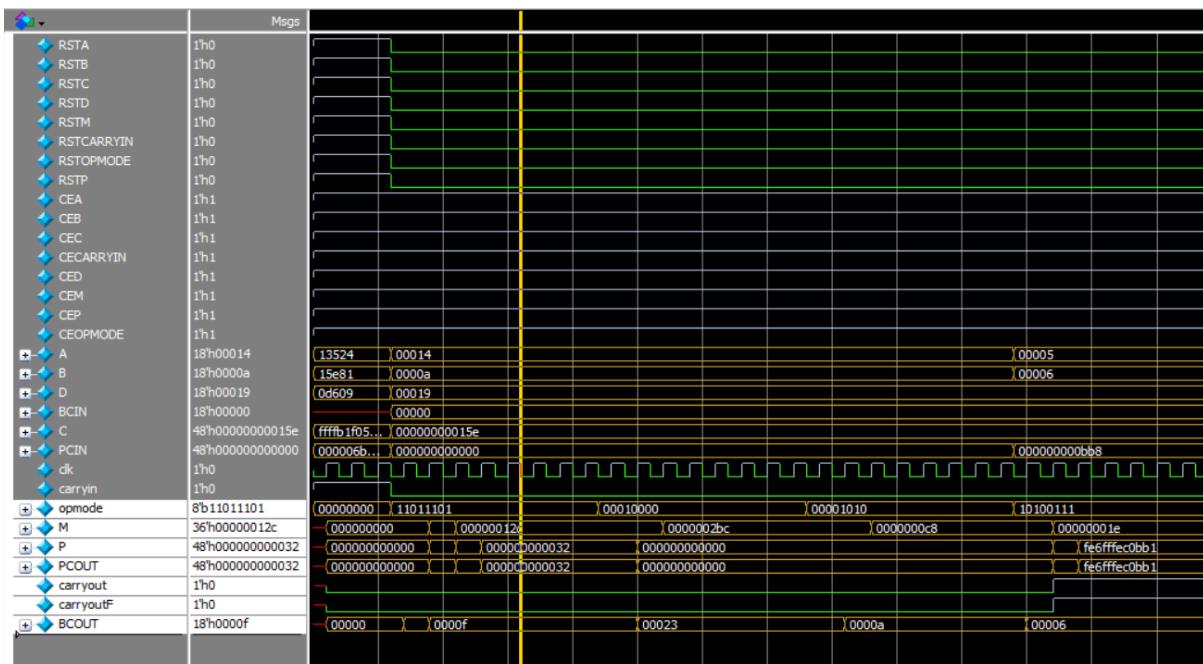
Functional Verification

Reset Operation



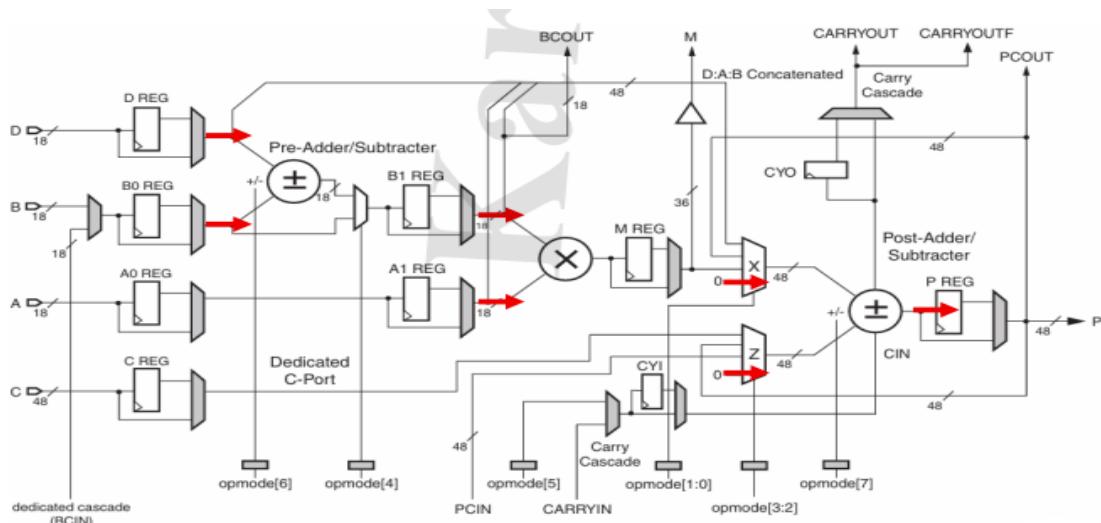
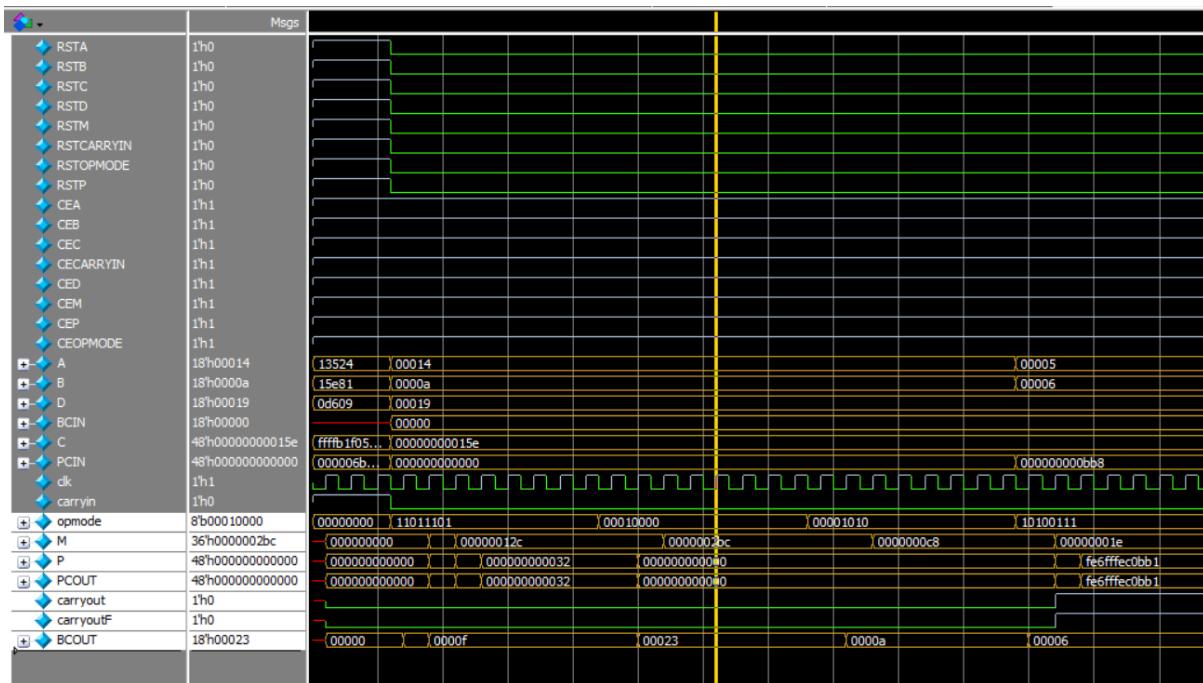
```
RSTA=1; RSTB=1; RSTC=1; RSTD=1; RSTM=1; RSTCARRYIN=1; RSTOPMODE=1; RSTP=1;
CEA=1; CEB=1; CEC=1; CECARRYIN=1; CED=1; CEM=1; CEP=1; CEOPMODE=1;
A=$random; B=$random; D=$random; C=$random; PCIN=$random; carryin=$random; opmode=8'b00000000;
#5 @(posedge clk);
display_values();
if (M !== 0 || P !== 0 || PCOUT !== 0 || BCOUT !== 0 || carryout !== 0 || carryoutF !== 0)
    $display("Reset FAIL");
else
    $display("Reset PASS
");
```

Simulation Flow (Path 1)



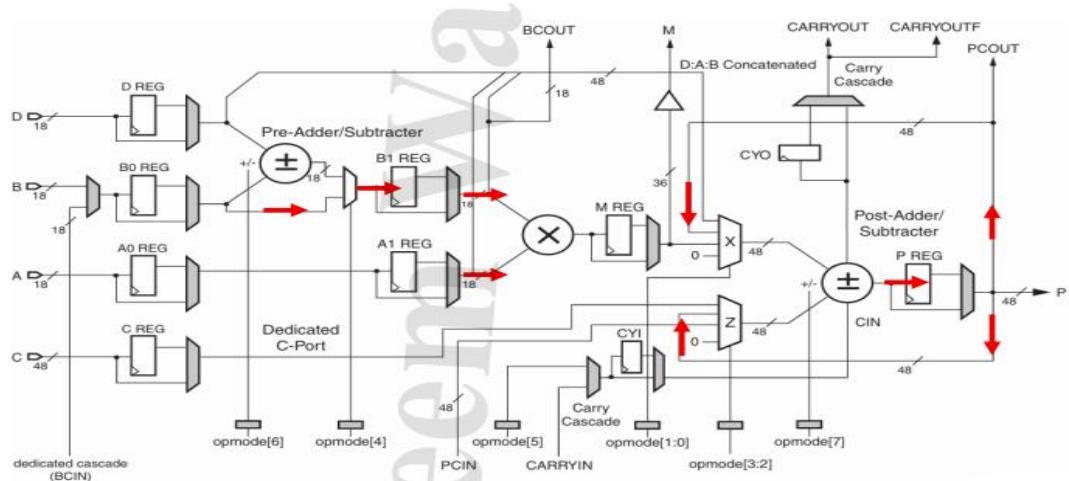
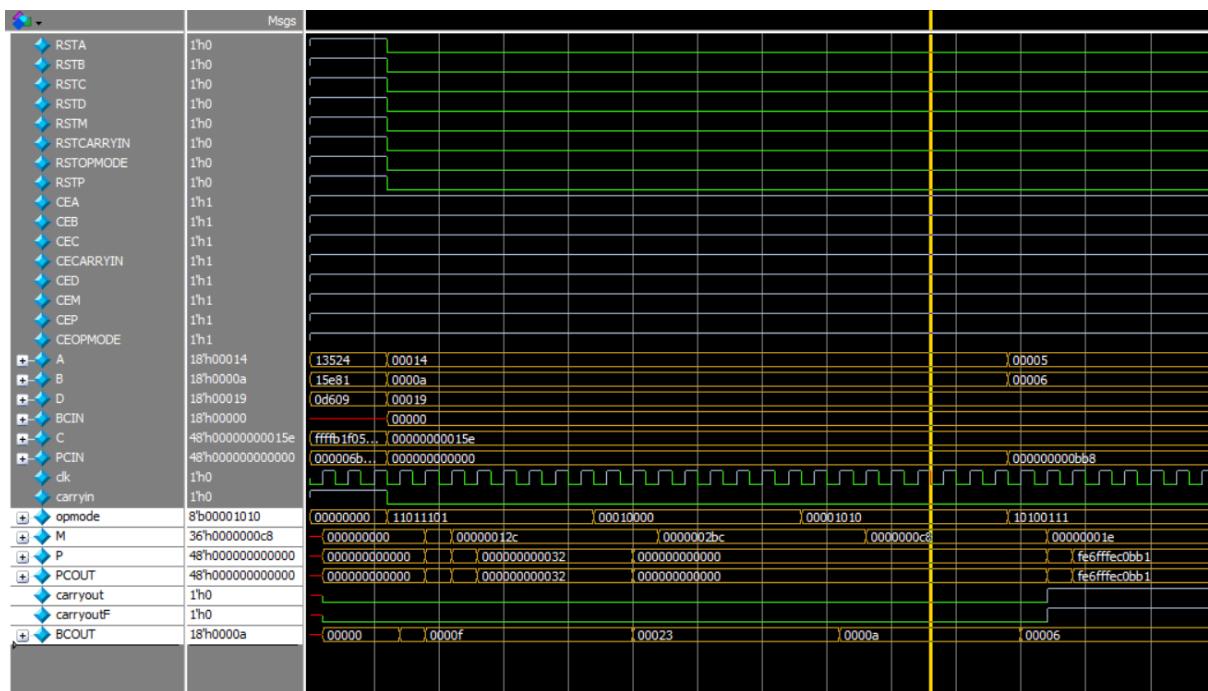
```
// PATH 1: pre-sub, post-sub
A=20; B=10; D=25; C=350; PCIN=0; carryin=0; BCIN=0; opmode=8'b11011101;
#10 repeat(4) @ (negedge clk);
display_values();
if (BCOUT == 18'hf && M == 36'h12c && P == 48'h32 && carryout == 0)
    $display("Path 1 PASS
-----");
else
    $display("Path 1 FAIL");
```

Simulation Flow (Path 2)



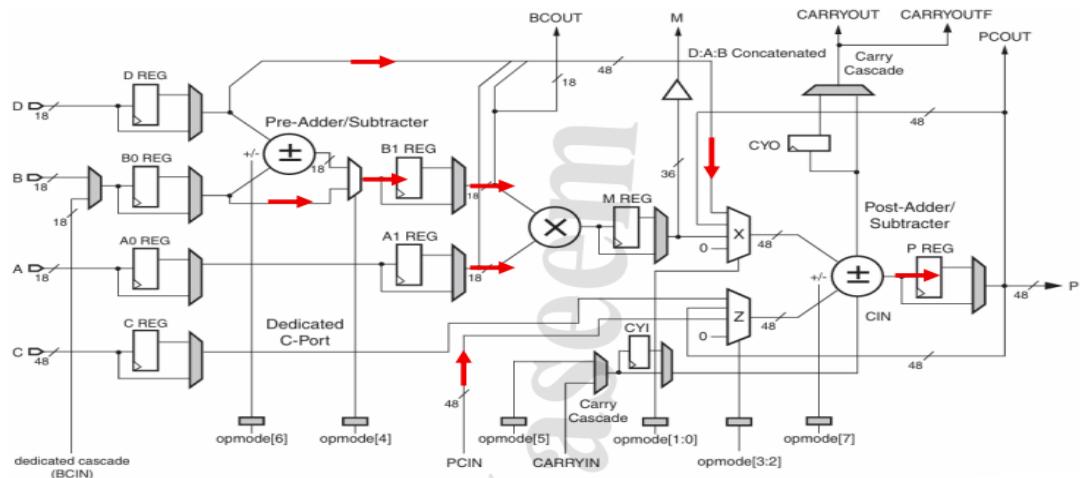
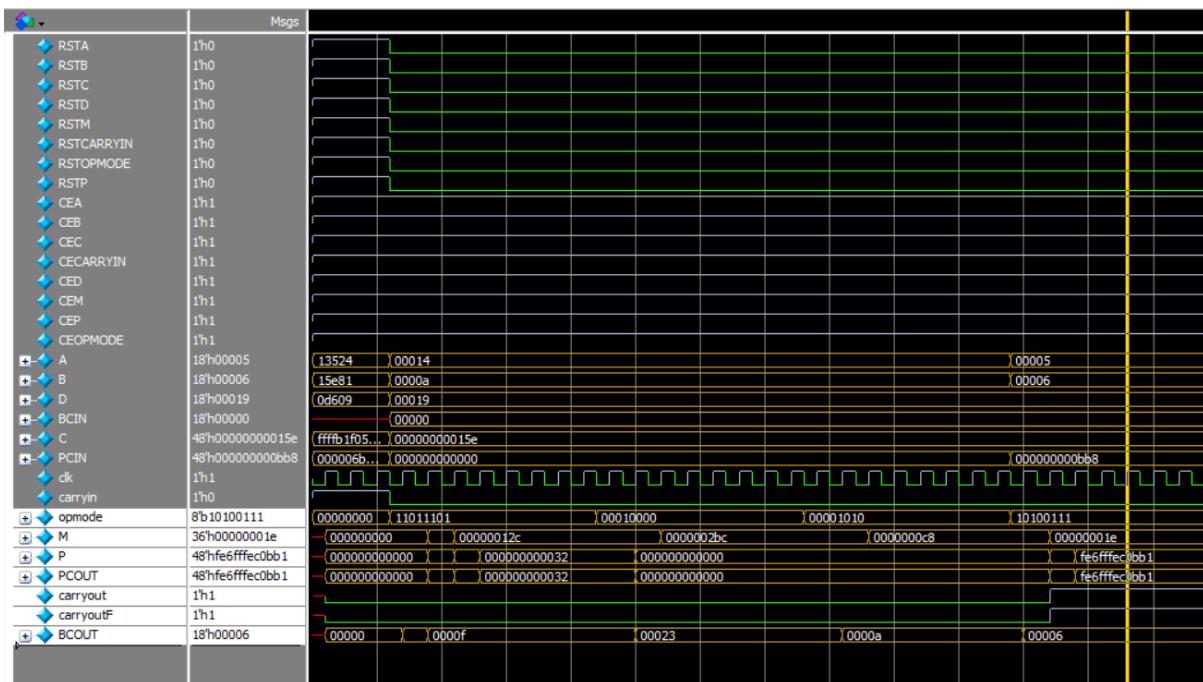
```
// PATH 2: pre-add, post-add
A=20; B=10; D=25; C=350; PCIN=0; carryin=0; BCIN=0; opmode=8'b00010000;
#10 repeat(4) @ (negedge clk);
display_values();
if [BCOUT == 18'h23 && M == 36'h2bc && P == 0 && carryout == 0]
    $display("Path 2 PASS
");
else
    $display("Path 2 FAIL");
```

Simulation Flow (Path 3)



```
// PATH 3: P feedback
A=20; B=10; D=25; C=350; PCIN=0; carryin=0; BCIN=0; opmode=8'b00001010;
#10 repeat(4) @(negedge clk);
display_values();
if (BCOUT === 18'ha && M === 36'hc8)
    $display("Path 3 PASS
-----");
else
    $display("Path 3 FAIL");
```

Simulation Flow (Path 4)



```
// PATH 4: DAB concat + subtract
A=5; B=6; D=25; C=350; PCIN=3000; carryin=0; BCIN=0; opmode=8'b10100111;
#10 repeat(4) @ (negedge clk);
display_values();
if (BCOUT == 18'd6 && M == 36'h1e && carryout == 1)
    $display("Path 4 PASS
    ");
else
    $display("Path 4 FAIL");

```

Transcript Output

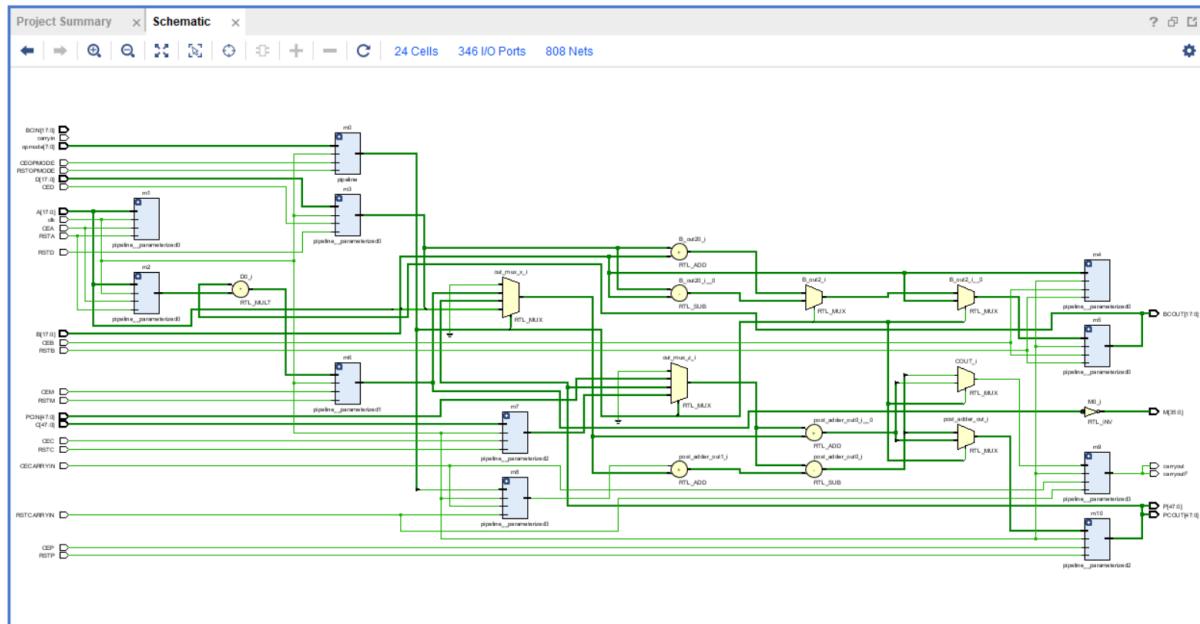
```
# Time = 6
# Inputs: A=79140, B=89729, D=54793, C=281473667061347, PCIN=112818957, carryin=1
# OPMODE = 00000000
# Outputs: M=0, P=0, PCOUT=0, carryout=0, carryoutF=0, BCOUT=0
# Reset PASS
#
# -----
# Time = 22
# Inputs: A=20, B=10, D=25, C=350, PCIN=0, carryin=0
# OPMODE = 11011101
# Outputs: M=12c, P=32, PCOUT=32, carryout=0, carryoutF=0, BCOUT=f
# Path 1 PASS
#
# -----
# Time = 38
# Inputs: A=20, B=10, D=25, C=350, PCIN=0, carryin=0
# OPMODE = 00010000
# Outputs: M=2bc, P=0, PCOUT=0, carryout=0, carryoutF=0, BCOUT=23
# Path 2 PASS
#
# -----
# Time = 54
# Inputs: A=20, B=10, D=25, C=350, PCIN=0, carryin=0
# OPMODE = 00001010
# Outputs: M=c8, P=0, PCOUT=0, carryout=0, carryoutF=0, BCOUT=a
# Path 3 PASS
#
# -----
# Time = 70
# Inputs: A=5, B=6, D=25, C=350, PCIN=3000, carryin=0
# OPMODE = 10100111
# Outputs: M=1e, P=fe6ffffec0bb1, PCOUT=fe6ffffec0bb1, carryout=1, carryoutF=1, BCOUT=6
# Path 4 PASS
# -----
```

Constraints File (.xdc)

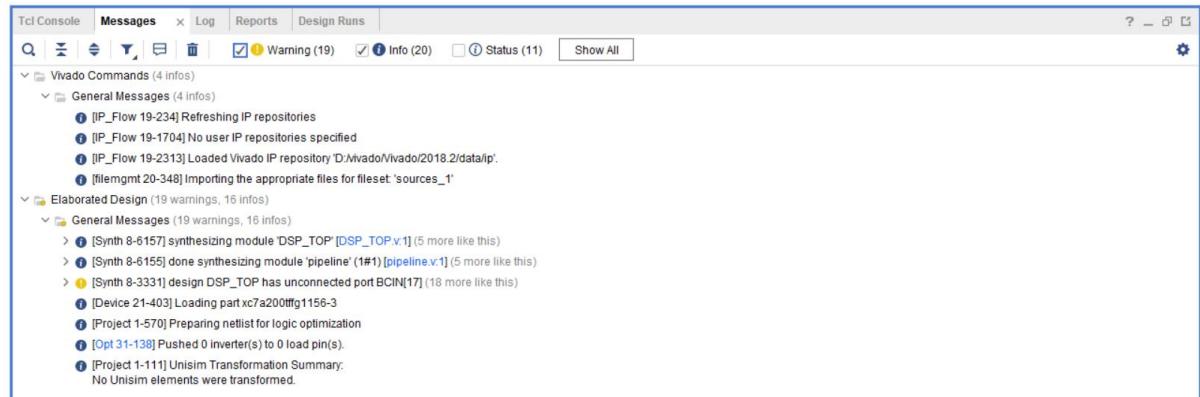
```
1 ## Clock signal
2 set_property -dict { PACKAGE_PIN W5    IOSTANDARD LVC MOS33 } [get_ports clk]
3 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
4
5 ## Configuration options, can be used for all designs
6 set_property CONFIG_VOLTAGE 3.3 [current_design]
7 set_property CFG_BVS VCCO [current_design]
8
9 ## SPI configuration mode options for QSPI boot, can be used for all designs
10 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
11 set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
12 set_property CONFIG_MODE SPIx4 [current_design]
```

Elaboration

Schematic



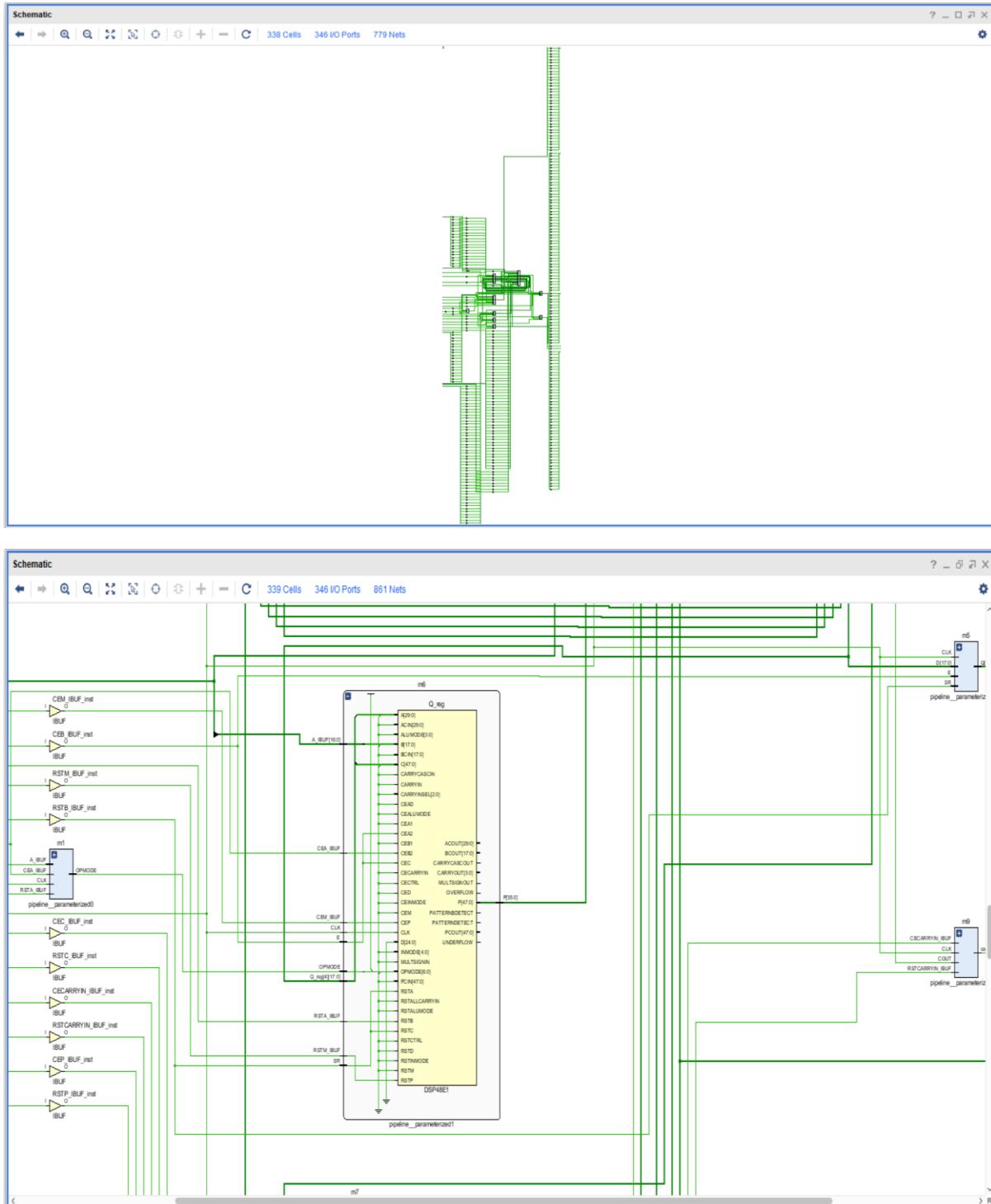
Messages



No errors

Synthesis

Schematic



Messages

SYNTHESIZED DESIGN - xc7a200tffg1156-3 (active)

Tcl Console Messages Log Reports Design Runs Utilization Timing Debug

Hide All

Vivado Commands (4 infos, 6 status messages)

- General Messages (4 infos, 6 status messages)
 - [IP_Flow 19-234] Refreshing IP repositories
 - [IP_Flow 19-1704] No user IP repositories specified
 - [IP_Flow 19-2313] Loaded Vivado IP repository 'D:/Vivado/Vivado/2018.2/data/ip'.
 - [filemgmt 20-348] Importing the appropriate files for filerset 'sources_1'
 - Command: synth_design -rtl -name rtl_1 (5 more like this)
- Synthesis (57 warnings, 30 infos, 11 status messages)
 - Command: synth_design -top DSP_TOP -part xc7a200tffg1156-3 (10 more like this)
 - [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a200t'
 - [Synth 8-6157] synthesizing module 'DSP_TOP' [DSP_TOP.v1] (5 more like this)
 - [Synth 8-6155] done synthesizing module 'pipeline' (#1)[pipeline.v1] (5 more like this)
 - [Synth 8-3331] design DSP_TOP has unconnected port BCIN[17] (37 more like this)
 - [Device 21-403] Loading part xc7a200tffg1156-3
 - [Project 1-26] Implementation specific constraints were found while reading constraint file [D:\Digital_Design\Kareem_Waseem_Diploma\Projects\DSP_Project_Final\project_DSP_Final\project_DSP_Final.srscs\constrs_1\imports\DSP_Project_Final\Constraints.xdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [xil\DSP_TOP_proplmp.xdc]. Resolution: To avoid this warning, move constraints listed in [Undefined] to another XDC file and exclude this new file from synthesis with the used_in_synthesis property (File Properties dialog in GUI) and re-run elaboration/synthesis.

Timing Report

SYNTHESIZED DESIGN - xc7a200tffg1156-3 (active)

Tcl Console Messages Log Reports Design Runs Timing Debug

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.219 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.

Utilization Report

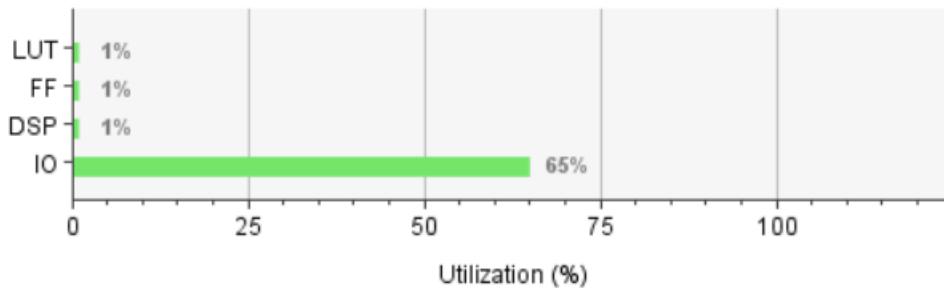
Tcl Console Messages Log Reports Design Runs Utilization Timing Debug

Hierarchy

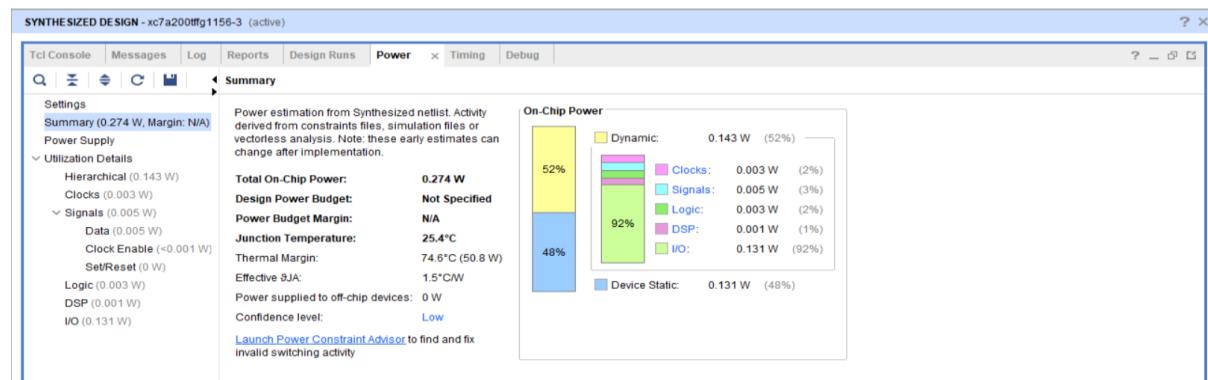
Name	1	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
N_DSP_TOP	254	143	1	327	1	
m0 (pipeline)	252	8	0	0	0	
m1 (pipeline_param...	0	1	0	0	0	
m3 (pipeline_param...	0	18	0	0	0	
m5 (pipeline_param...	0	18	0	0	0	
m6 (pipeline_param...	0	0	1	0	0	
m7 (pipeline_param...	0	48	0	0	0	
m8 (pipeline_param...	1	1	0	0	0	
m9 (pipeline_param...	0	1	0	0	0	
m10 (pipeline_param...	0	48	0	0	0	

Summary

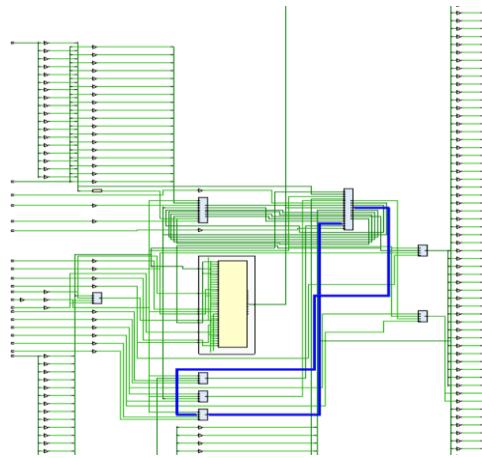
Resource	Utilization	Available	Utilization %
LUT	254	134600	0.19
FF	143	269200	0.05
DSP	1	740	0.14
IO	327	500	65.40



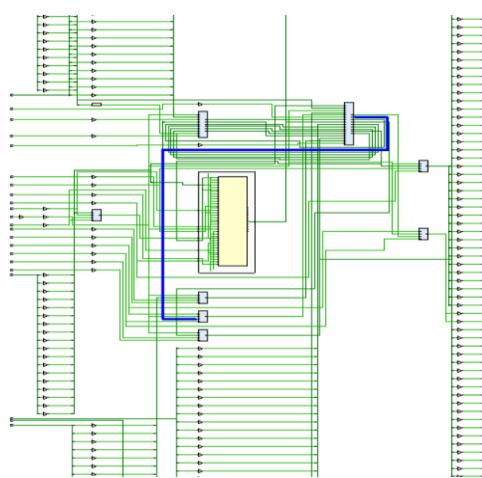
Power Report



Worst Negative Slack

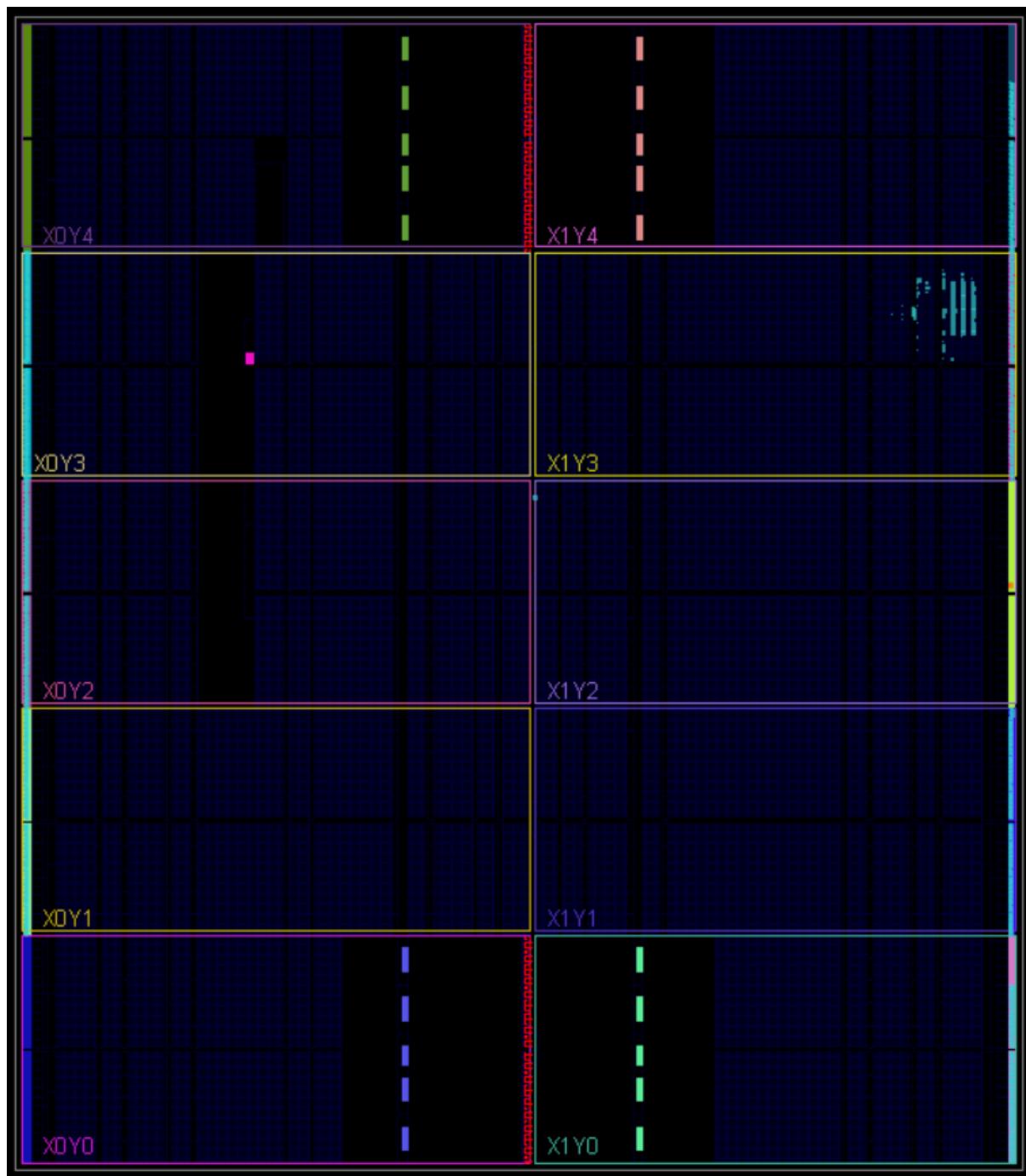


Worst Hold Slack



Implementation

Device View



Messages

The screenshot shows the 'IMPLEMENTED DESIGN - xc7a200tfg1156-3 (active)' window with the 'Messages' tab selected. The window displays various messages from the Vivado tool, categorized into 'Vivado Commands' and 'Synthesis'. Under 'Synthesis', there are 57 warnings, 30 infos, and 11 status messages. One warning is highlighted in yellow: '[IP_Flow 19-234] Refreshing IP repositories'. Other messages include '[IP_Flow 19-1704] No user IP repositories found' and '[Synth 8-6157] Got license for feature 'Synthesis' and/or device 'xc7a200t'.

Timing Report

The screenshot shows the 'IMPLEMENTED DESIGN - xc7a200tfg1156-3 (active)' window with the 'Timing' tab selected. The 'Design Timing Summary' section is displayed, showing setup, hold, and pulse width constraints. Key values include Worst Negative Slack (WNS) at 3.706 ns, Worst Hold Slack (WHS) at 0.264 ns, and Worst Pulse Width Slack (WPWS) at 4.500 ns. A note indicates that all user-specified timing constraints are met.

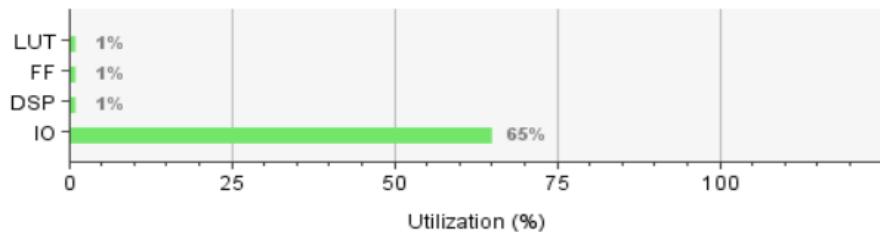
Utilization Report

The screenshot shows the 'IMPLEMENTED DESIGN - xc7a200tfg1156-3 (active)' window with the 'Utilization' tab selected. The 'Hierarchy' section is expanded to show the utilization of various logic components. The main table provides detailed statistics for each component, such as DSP usage, bonded IOBs, and BUFCTRLs. For example, the 'DSP_TOP' component uses 253 Slice LUTs, 144 Slice Registers, and 253 LUTs as Logic.

Name	Slice LUTs (133800)	Slice Registers (267600)	Slice (33450)	LUT as Logic (133800)	LUT Flip Flop Pairs (133800)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)
DSP_TOP	253	144	103	253	29	1	327	1
m0 (pipeline)	252	8	81	252	0	0	0	0
m1 (pipeline__param...	0	1	1	0	0	0	0	0
m3 (pipeline__param...	0	18	8	0	0	0	0	0
m5 (pipeline__param...	0	18	8	0	0	0	0	0
m6 (pipeline__param...	0	0	0	0	0	1	0	0
m7 (pipeline__param...	0	48	12	0	0	0	0	0
m8 (pipeline__param...	1	1	1	1	1	0	0	0
m9 (pipeline__param...	0	2	1	0	0	0	0	0
m10 (pipeline__param...	0	48	18	0	0	0	0	0

Summary

Resource	Utilization	Available	Utilization %
LUT	253	133800	0.19
FF	144	267600	0.05
DSP	1	740	0.14
IO	327	500	65.40



Power Report

IMPLEMENTED DESIGN - xc7a200tffg1156-3 (active)

Tcl Console Messages Log Reports Design Runs Power Methodology DRC Timing Utilization

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 0.279 W Design Power Budget: Not Specified Power Budget Margin: N/A

Junction Temperature: 25.4°C Thermal Margin: 74.6°C (50.8 W) Effective JJA: 1.5°C/W

Clocks (0.002 W) Signals (0.013 W) Data (0.013 W) Clock Enable (<0.001 W) Set/Reset (0 W) Logic (0.002 W) DSP (0.001 W) I/O (0.13 W)

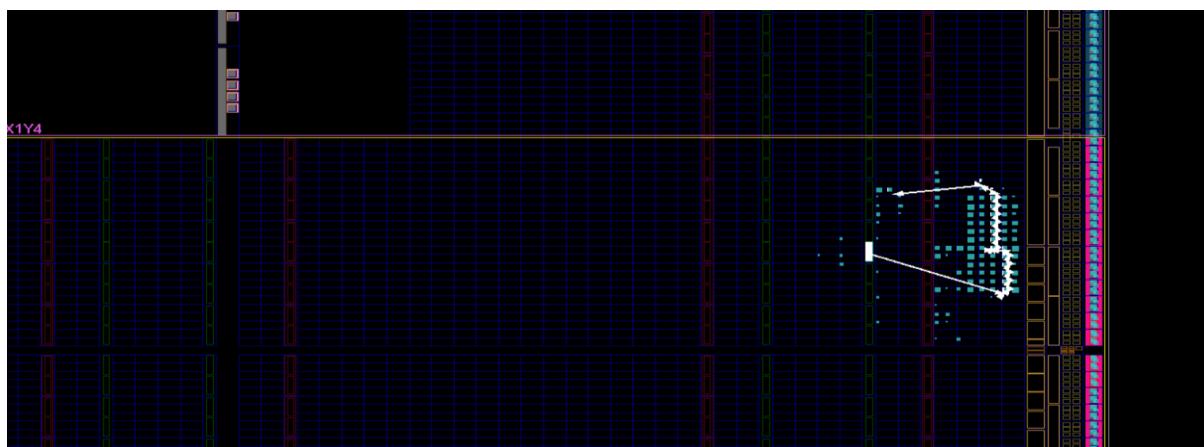
On-Chip Power

Dynamic: 0.148 W (53%)
9% Clocks: 0.002 W (1%)
Signals: 0.013 W (9%)
87% Logic: 0.002 W (2%)
DSP: 0.001 W (1%)
I/O: 0.130 W (87%)

Device Static: 0.131 W (47%)

Launch Power Constraint Advisor to find and fix invalid switching activity

Worst Negative Slack



Worst Hold Slack



Linting

