

SPARTAN6 – DSP48A1

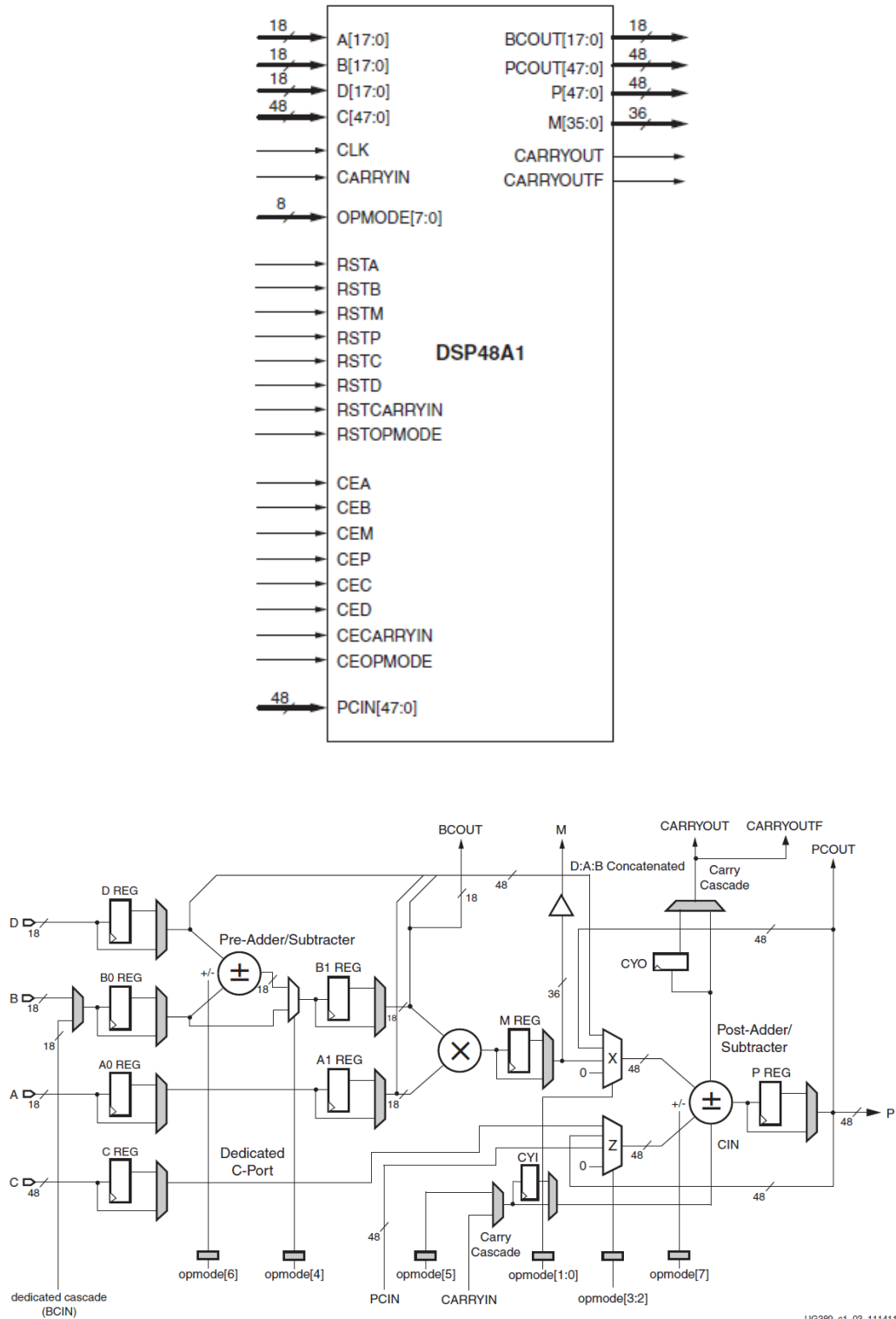
Abdelrahman Wael

Table of Contents

RTL code	4
Testbench code	8
The do file:	15
QuestaSim Snippets	15
2.1. Verify Reset Operation	15
• Waveform	16
• Transcript	16
2.2. Verify DSP Path 1	17
• Waveform	17
• Transcript	17
2.3. Verify DSP Path 2	18
• Waveform	18
• Transcript	18
2.4. Verify DSP Path 3	19
• Waveform	19
• Transcript	19
2.5. Verify DSP Path 4	20
• Waveform	20
• Transcript	20
Constraint File	21
Elaboration	21
• Messages	21
• Schematic snippets	22
Synthesis	22
• Messages	22
• Utilization report	23
• Timing report	23
• Schematic snippets	24

Implementation	25
• Messages	25
• Utilization report	25
• Timing report	25
• Device snippets	26
Linting	26

SPARTAN6 – DSP48A1



RTL code

```
module DSP48A1(A,B,C,D,M,P,CARRYIN,CLK,OPMODE,CEA,CEC,CED,CEM,
CEOPMODE,CEP,CEB,RSTOPMODE,RSTA,RSTM,RSTP,RSTB,RSTC,RSTD,RSTCARRYIN,
PCIN,PCOUT,BCOUT,CARRYOUT,CARRYOUTF,CECARRYIN,BCIN );

input [17:0] A,B,D,BCIN;
input [47:0] C;
input CARRYIN,CLK,CEA,CEC,CED,CEM,
CEOPMODE,CEP,CEB,RSTOPMODE,RSTA,RSTM,RSTP,RSTB,RSTC,RSTD,RSTCARRYIN,CECARRYIN;
input [47:0] PCIN;
input [7:0] OPMODE;
output CARRYOUT,CARRYOUTF;
output [47:0] PCOUT,P;
output [35:0] M;
output [17:0] BCOUT;

// 1=registered 0 not registered
parameter A0REG=0, A1REG=1,B0REG=0,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";
parameter RSTTYPE="SYNC";
parameter B_INPUT="DIRECT";

wire [17:0] A0_reg,B0_reg,D_reg;
wire [17:0] A1_reg,B1_reg;
wire [47:0] C_reg;
wire [35:0] M_reg;
wire [47:0] P_reg;
wire [7:0] OPMODE_reg;
wire CARRYOUT_reg,CARRYOUTF_reg;
wire [17 : 0] a1, b1;
wire [17 : 0] B_SELECT;
wire CYI;
wire CYO;
```

```

// Assign B_SELECT based on B_INPUT parameter
assign B_SELECT = (B_INPUT=="DIRECT")?B:(B_INPUT=="CASCADE")? BCIN:18'b0;
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(8)) OPMODE_REG (CLK, RSTOPMODE, CEOPMODE,
OPMODE_REG, OPMODE, OPMODE_reg);
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(18)) D0_STAGE (CLK, RSTD, CED, DREG, D,
D_reg);
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(18)) B0_STAGE (CLK, RSTB, CEB, B0REG,
B_SELECT, B0_reg);

//pre-adder/subtractor
wire [17:0] pre_out;
assign pre_out = (~OPMODE_reg[4] )? B0_reg :
                (OPMODE_reg[6] ? D_reg - B0_reg : D_reg + B0_reg);
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(18)) B1_STAGE (CLK, RSTB, CEB, B1REG,
pre_out, B1_reg);
//*****
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(18)) A0_STAGE (CLK, RSTA, CEA, A0REG,
A,A0_reg);
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(18)) A1_STAGE (CLK, RSTA, CEA, A1REG,
A0_reg, A1_reg);

//multiplier
wire [35:0] MUL_out;
assign MUL_out = (B1_reg*A1_reg);
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(36)) M_STAGE (CLK, RSTM, CED, MREG,
MUL_out, M_reg);
assign M=M_reg;
//*****

//the X_MUX
wire [47:0] X_OUT;
wire [47:0] D_A_B_CONC;
assign D_A_B_CONC = {D_reg[11:0], A1_reg, B1_reg};
assign X_OUT = (OPMODE_reg[1:0] == 2'b00) ? 48'b0 :
                (OPMODE_reg[1:0] == 2'b01) ? {12'b0,M_reg} :
                (OPMODE_reg[1:0] == 2'b10) ? PCOUT :
                (OPMODE_reg[1:0] == 2'b11) ? D_A_B_CONC :
                48'b0;
//*****

```

```

//the Z_MUX
wire [47:0] Z_OUT;
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(48)) C0_STAGE (CLK, RSTC, CEC, CREG, C,
C_reg);
assign Z_OUT = (OPMODE_reg[3:2] == 2'b00) ? 48'b0 :
                (OPMODE_reg[3:2] == 2'b01) ? PCIN :
                (OPMODE_reg[3:2] == 2'b10) ? PCOUT :
                (OPMODE_reg[3:2] == 2'b11) ? C_reg :
                48'b0;
//*****

//THE CARRYIN
wire carry_select;
assign carry_select =
(CARRYINSEL=="OPMODE5")?OPMODE_reg[5]:(CARRYINSEL=="CARRYIN")?CARRYIN:
1'b0;
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(1)) CARRYIN_STAGE (CLK, RSTCARRYIN,
CECARRYIN, CARRYINREG,
carry_select
,CYI);
//*****

//post-adder/subtracter
wire carry_out;
wire [47:0] out_post;
assign {carry_out,out_post} = (OPMODE_reg[7]) ?
                (Z_OUT - (X_OUT + {{47{1'b0}}}, CYI)) :
                (Z_OUT + X_OUT + {{47{1'b0}}}, CYI));
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(1)) CARRYOUT_STAGE (CLK, RSTCARRYIN,
CECARRYIN, CARRYOUTREG, carry_out
,CYO);
assign CARRYOUT=CYO;
assign CARRYOUTF=CARRYOUT;

//PCOUT BCOUT OUTPUTS
REG_MUX #(.sync_type(RSTTYPE), .WIDTH(48)) P_STAGE (CLK, RSTP, CEP, PREG,
out_post, P_reg);
assign P=P_reg;
assign PCOUT=P_reg;
assign BCOUT = B1_reg;

```

```

endmodule
//The reg mux module
module REG_MUX #(parameter [5:0] WIDTH = 8, parameter sync_type = "SYNC") (
    input clk, rst, clk_enable, select,
    input [WIDTH-1:0] in,
    output reg [WIDTH-1:0] out
);

reg [WIDTH-1:0] d_ff;

localparam synchronous = (sync_type == "SYNC");
localparam asynchronous = (sync_type == "ASYNC");

generate
    if (asynchronous) begin
        always @(posedge clk or posedge rst) begin
            if (rst)
                d_ff <= 0;
            else if (clk_enable)
                d_ff <= in;
        end
    end else if (synchronous) begin
        always @(posedge clk) begin
            if (rst)
                d_ff <= 0;
            else if (clk_enable)
                d_ff <= in;
        end
    end
endgenerate

always@(*) begin
    if(select == 1) begin
        out = d_ff;
    end
    else begin
        out = in;
    end
end

endmodule

```


Testbench code

```
module DSP48A1_TB();
parameter A0REG=0, A1REG=1,B0REG=0,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";
parameter RSTTYPE="SYNC";
parameter B_INPUT="DIRECT";

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

//OUTPUT STIMLUS
wire CARRYOUT_dut,CARRYOUTF_dut;
wire [47:0] PCOUT_dut,P_dut;
wire [35:0] M_dut;
wire [17:0] BCOUT_dut;
//*****

reg CARRYOUT_expected,CARRYOUTF_expected;
reg [47:0] PCOUT_expected,P_expected;
reg [35:0] M_expected;
reg [17:0] BCOUT_expected;
reg [47:0] P_old;
reg CARROUT_old;
```

```

//int
DSP48A1 m1(A,B,C,D,M_dut,P_dut,CARRYIN,CLK,OPMODE,CEA,CEC,CED,CEM,
CEOPMODE,CEP,CEB,RSTOPMODE,RSTA,RSTM,RSTP,RSTB,RSTC,RSTD,RSTCARRYIN,
PCIN,PCOUT_dut,BCOUT_dut,CARRYOUT_dut,CARRYOUTF_dut,CECARRYIN,BCIN );

initial begin
    CLK = 0;
    forever begin
        #1 CLK = ~CLK;
    end
end

initial begin
    //Just intialization to avoid being unknown on the wave forms
    A = 0; B = 0; C = 0; D = 0; BCIN = 0; CARRYIN = 0;
    OPMODE = 0; CEA = 0; CEB = 0; CEC = 0; CECARRYIN = 0;
    CED = 0; CEM = 0; CEOPMODE = 0; CEP = 0; PCIN = 0;
    BCOUT_expected = 0; PCOUT_expected = 0; M_expected = 0; P_expected = 0;
    CARRYOUT_expected = 0; CARRYOUTF_expected = 0;
    //*****
    RSTOPMODE=1;
    RSTA=1;
    RSTM=1;
    RSTP=1;
    RSTB=1;
    RSTC=1;
    RSTD=1;
    RSTCARRYIN=1;

    A = $random;
    B = $random;
    D = $random;
    BCIN = $random;
    C = $random;
    CARRYIN = $random;
    CEA = $random;
    CEC = $random;
    CED = $random;
    CEM = $random;
    CEOPMODE = $random;
    CEP = $random;
    CEB = $random;
    RSTOPMODE = 1;

```

```

RSTA = 1;
RSTM = 1;
RSTP = 1;
RSTB = 1;
RSTC = 1;
RSTD = 1;
RSTCARRYIN = 1;
CECARRYIN = $random;
PCIN = $random;
OPMODE = $random;
@(negedge CLK);
BCOUT_expected = 0; PCOUT_expected = 0; M_expected = 0; P_expected = 0;
CARRYOUT_expected = 0; CARRYOUTF_expected = 0;
if (P_dut == P_expected )
    $display("Reset Test Passed for P");
else begin
    $display("Rest test failed for P");
end
if (M_dut == M_expected )
    $display("Reset Test Passed for M");
else begin
    $display("Rest test failed for M");
end
if (BCOUT_dut == BCOUT_expected )
    $display("Reset Test Passed for BCOUT");
else begin
    $display("Rest test failed for BCOUT");
end
if (PCOUT_dut == PCOUT_expected )
    $display("Reset Test Passed for PCOUT");
else begin
    $display("Rest test failed for PCOUT");
end
if (CARRYOUT_dut == CARRYOUT_expected )
    $display("Reset Test Passed for CARRYOUT");
else
    $display("Rest test failed for CARRYOUT");
if (CARRYOUTF_dut == CARRYOUTF_expected)
    $display("Reset Test Passed for CARRYOUTF");
else
    $display("Rest test failed for CARRYOUTF");

//*****

```

```

//Deassert all reset signals and assert all clock enable signals to
//validate the functionality of the subsequent DSP paths.
    RSTA = 0; RSTB = 0; RSTC = 0; RSTCARRYIN = 0; RSTD = 0; RSTM = 0;
    RSTOPMODE = 0; RSTP = 0; CEA = 1; CEB = 1; CEC = 1; CECARRYIN = 1;
    CED = 1; CEM = 1; CEOPMODE = 1; CEP = 1;
    //TEST 2.2

    OPMODE = 8'b11011101;
    A = 18'd20;
    B = 18'd10;
    C = 48'd350;
    D = 18'd25;
    BCIN=$random;
    PCIN=$random;
    CARRYIN=$random;
    BCOUT_expected = 18'hf;
    M_expected = 36'h12c;
    P_expected = 48'h32;
    PCOUT_expected = 48'h32 ;
    CARRYOUT_expected = 0;
    CARRYOUTF_expected = 0;
    repeat(4) @(negedge CLK);
    if (P_dut === P_expected )
        $display("Path 1 test Passed for P");
    else begin
        $display("Path 1 test failed for P");
    end
    if (M_dut === M_expected )
        $display("Path 1 test Passed for M");
    else begin
        $display("Path 1 test failed for M");
    end
    if (BCOUT_dut === BCOUT_expected )
        $display("Path 1 test Passed for BCOUT");
    else begin
        $display("Path 1 test failed for BCOUT");
    end
    if (PCOUT_dut === PCOUT_expected )
        $display("Path 1 test Passed for PCOUT");
    else begin
        $display("Path 1 test failed for PCOUT");
    end
end

```

```

if (CARRYOUT_dut === CARRYOUT_expected )
    $display("Path 1 test Passed for CARRYOUT");
else
    $display("Path 1 test failed for CARRYOUT");
if (CARRYOUTF_dut === CARRYOUTF_expected)
    $display("Path 1 test Passed for CARRYOUTF");
else
    $display("Path 1 test failed for CARRYOUTF");
//Test 2.3
OPMODE = 8'b00010000 ;
A = 18'd20;
B = 18'd10;
C = 48'd350;
D = 18'd25;
BCIN=$random;
PCIN=$random;
CARRYIN=$random;
BCOUT_expected = 18'h23;
M_expected = 36'h2bc;
P_expected = 48'h0;
PCOUT_expected = 48'h0 ;
CARRYOUT_expected =0;
CARRYOUTF_expected = 0;
repeat(3) @(negedge CLK);
if (P_dut === P_expected )
    $display("Path 2 test Passed for P");
else begin
    $display("Path 2 test failed for P");
end
if (M_dut === M_expected )
    $display("Path 2 test Passed for M");
else begin
    $display("Path 2 test failed for M");
end
if (BCOUT_dut === BCOUT_expected )
    $display("Path 2 test Passed for BCOUT");
else begin
    $display("Path 12test failed for BCOUT");
end
if (PCOUT_dut === PCOUT_expected )
    $display("Path 2 test Passed for PCOUT");
else begin
    $display("Path 2 test failed for PCOUT");
end

```

```

if (CARRYOUT_dut === CARRYOUT_expected )
    $display("Path 2 test Passed for CARRYOUT");
else
    $display("Path 2 test failed for CARRYOUT");
if (CARRYOUTF_dut === CARRYOUTF_expected)
    $display("Path 2 test Passed for CARRYOUTF");
else
    $display("Path 2 test failed for CARRYOUTF");

//Test 2.4
OPMODE = 8'b00001010;
P_old=P_expected;
CARROUT_old=CARRYOUT_expected;
A = 18'd20;
B = 18'd10;
C = 48'd350;
D = 18'd25;
BCIN=$random;
PCIN=$random;
CARRYIN=$random;
BCOUT_expected = 18'ha;
M_expected = 36'hc8;
P_expected = P_old;
PCOUT_expected = P_old ;
CARRYOUT_expected =CARROUT_old;
CARRYOUTF_expected = CARROUT_old;
repeat(3) @(negedge CLK);
if (P_dut === P_expected )
    $display("Path 3 test Passed for P");
else begin
    $display("Path 3 test failed for P");
end
if (M_dut === M_expected )
    $display("Path 3 test Passed for M");
else begin
    $display("Path 3 test failed for M");
end
if (BCOUT_dut === BCOUT_expected )
    $display("Path 3 test Passed for BCOUT");
else begin
    $display("Path 2test failed for BCOUT");
end

```

```

if (PCOUT_dut === PCOUT_expected )
    $display("Path 3 test Passed for PCOUT");
else begin
    $display("Path 3 test failed for PCOUT");
end
if (CARRYOUT_dut === CARRYOUT_expected )
    $display("Path 3 test Passed for CARRYOUT");
else
    $display("Path 3 test failed for CARRYOUT");
if (CARRYOUTF_dut === CARRYOUTF_expected)
    $display("Path 3 test Passed for CARRYOUTF");
else
    $display("Path 3 test failed for CARRYOUTF");
// Test 2.5
OPMODE = 8'b10100111;
A = 18'd5;
B = 18'd6;
C = 48'd350;
D = 18'd25;
PCIN = 3000 ;
BCIN=$random;
CARRYIN=$random;
BCOUT_expected = 18'd6;
M_expected = 36'h1e;
P_expected = 48'hfe6fffec0bb1;
PCOUT_expected = 48'hfe6fffec0bb1 ;
CARRYOUT_expected =1;
CARRYOUTF_expected = 1;
repeat(3) @(negedge CLK);
if (P_dut === P_expected )
    $display("Path 4 test Passed for P");
else begin
    $display("Path 4 test failed for P");
end
if (M_dut === M_expected )
    $display("Path 4 test Passed for M");
else begin
    $display("Path 4 test failed for M");
end
if (BCOUT_dut === BCOUT_expected )
    $display("Path 4 test Passed for BCOUT");
else begin
    $display("Path 4 test failed for BCOUT");
end

```

```

if (PCOUT_dut === PCOUT_expected )
    $display("Path 4 test Passed for PCOUT");
else begin
    $display("Path 4 test failed for PCOUT");
end
if (CARRYOUT_dut === CARRYOUT_expected )
    $display("Path 4 test Passed for CARRYOUT");
else
    $display("Path 4 test failed for CARRYOUT");
if (CARRYOUTF_dut === CARRYOUTF_expected)
    $display("Path 4 test Passed for CARRYOUTF");
else
    $display("Path 4 test failed for CARRYOUTF");
$stop;

end
endmodule

```

The do file:

```

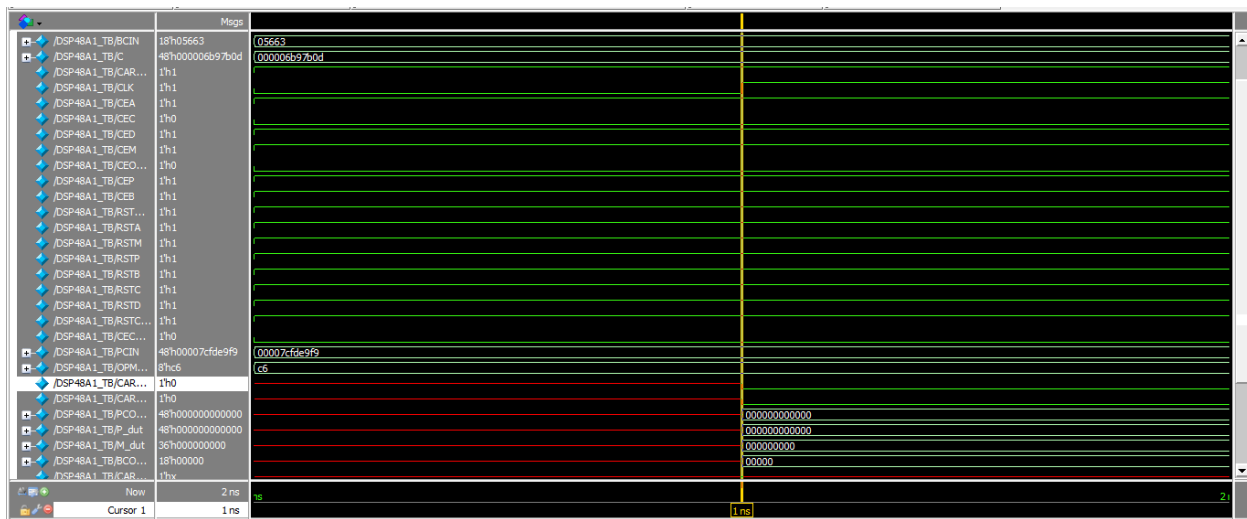
1  vlib work
2  vlog DSP48A1.v DSP48A1_TB.v
3  vsim -voptargs=+acc work.DSP48A1_TB
4  add wave *
5  run -all
6  #quit -sim

```

QuestaSim Snippets

2.1. Verify Reset Operation

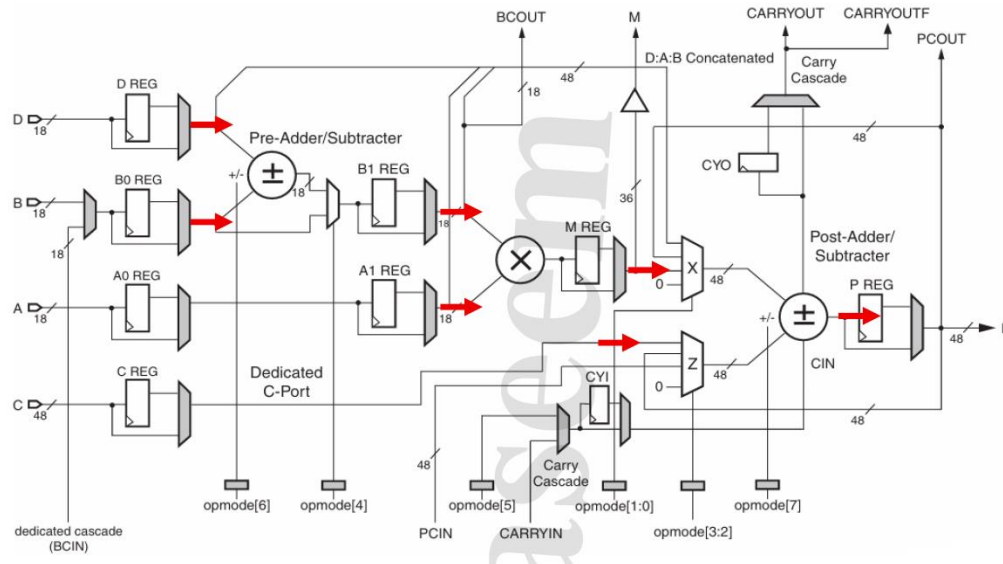
• Waveform



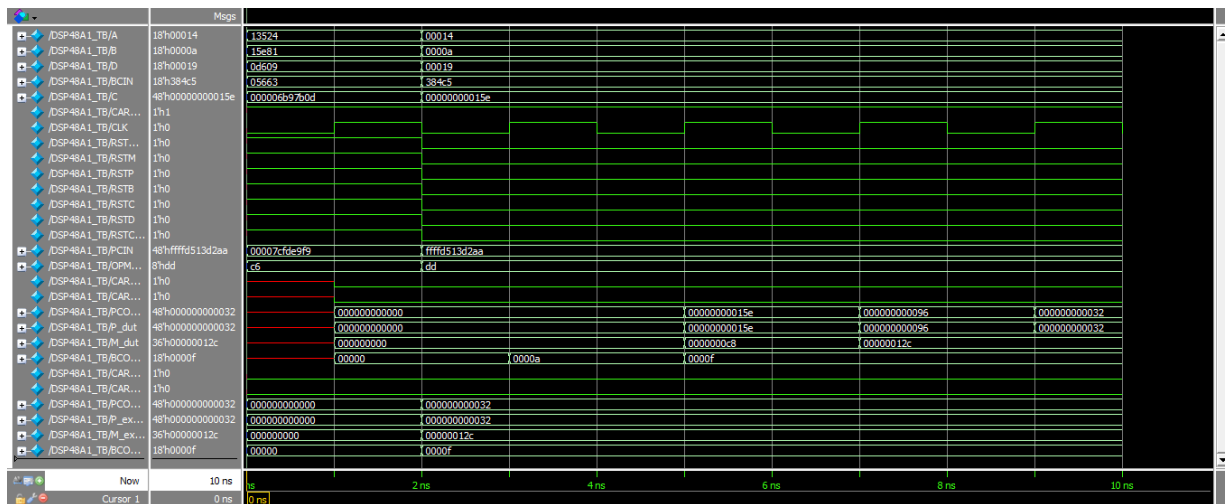
• Transcript

```
# Reset Test Passed for P
# Reset Test Passed for M
# Reset Test Passed for BCOUT
# Reset Test Passed for PCOUT
# Reset Test Passed for CARRYOUT
# Reset Test Passed for CARRYOUTF
# ** Note: $stop      : DSP48A1_TB.v(110)
#   Time: 2 ns  Iteration: 1  Instance: /DSP48A1_TB
# Break in Module DSP48A1_TB at DSP48A1_TB.v line 110
```

2.2. Verify DSP Path 1



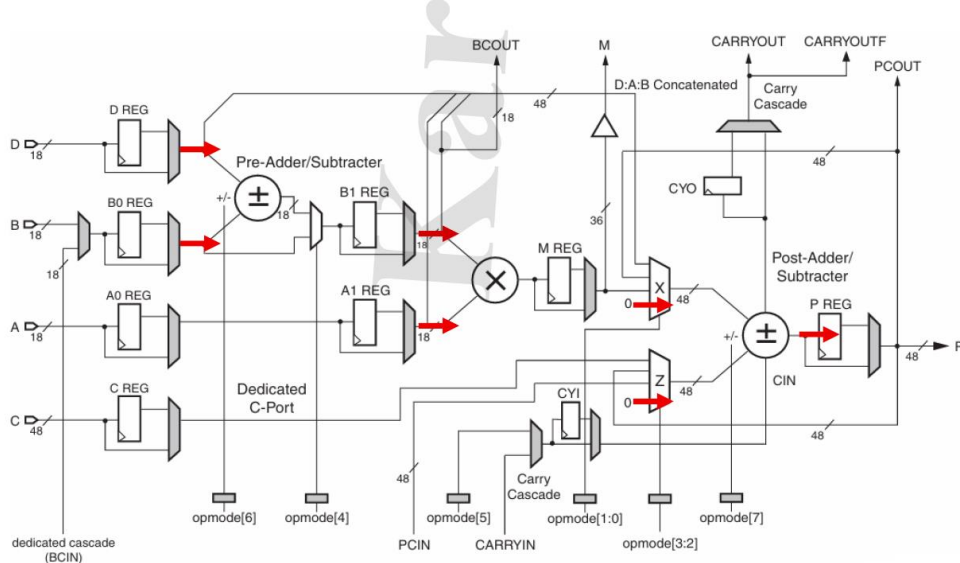
- **Waveform**



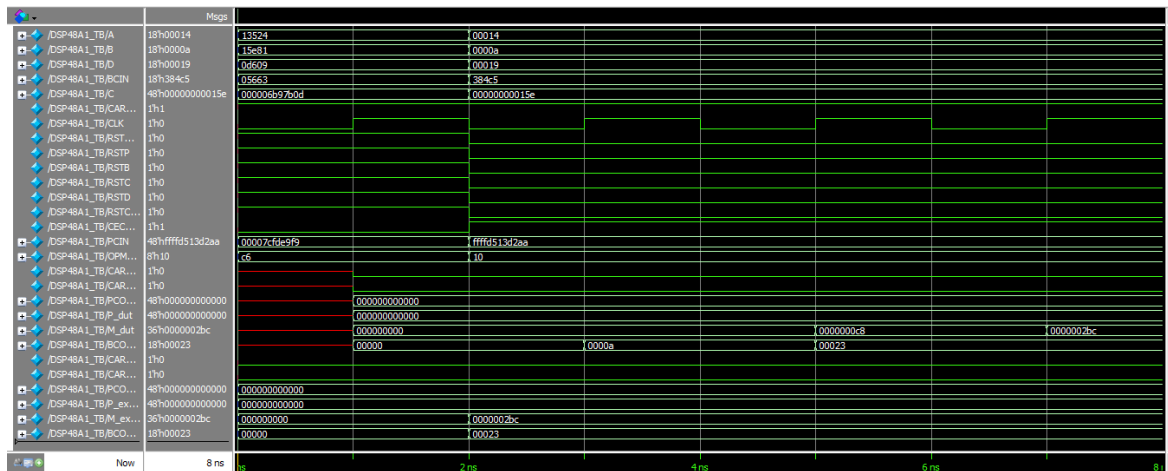
- Transcript

```
# Path 1 test Passed for P
# Path 1 test Passed for M
# Path 1 test Passed for BCOUT
# Path 1 test Passed for PCOUT
# Path 1 test Passed for CARRYOUT
# Path 1 test Passed for CARRYOUTF
# ** Note: $stop      : DSP48A1_TB.v(167)
#      Time: 10 ns   Iteration: 1   Instance: /DSP48A1_TB
# Break in Module DSP48A1_TB at DSP48A1_TB.v line 167
```

2.3. Verify DSP Path 2



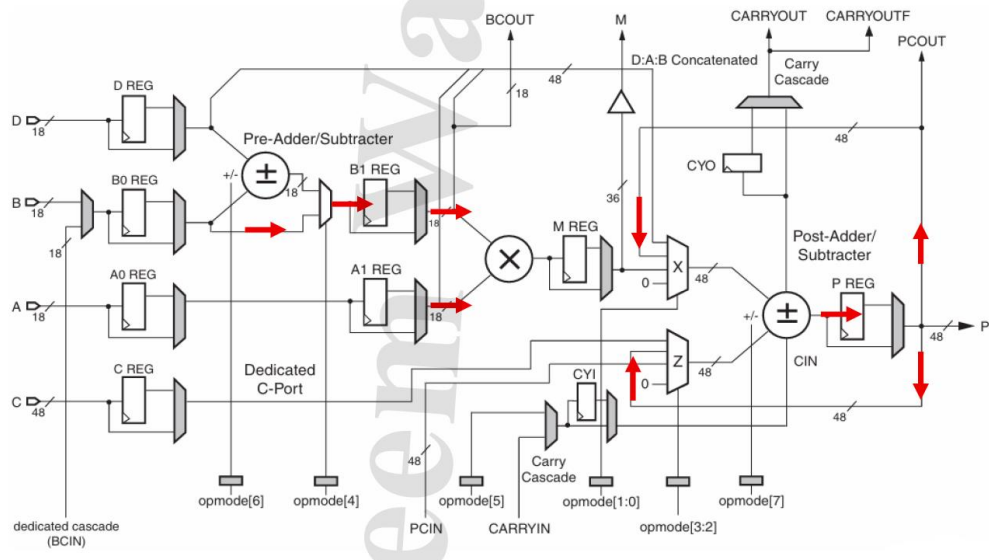
- **Waveform**



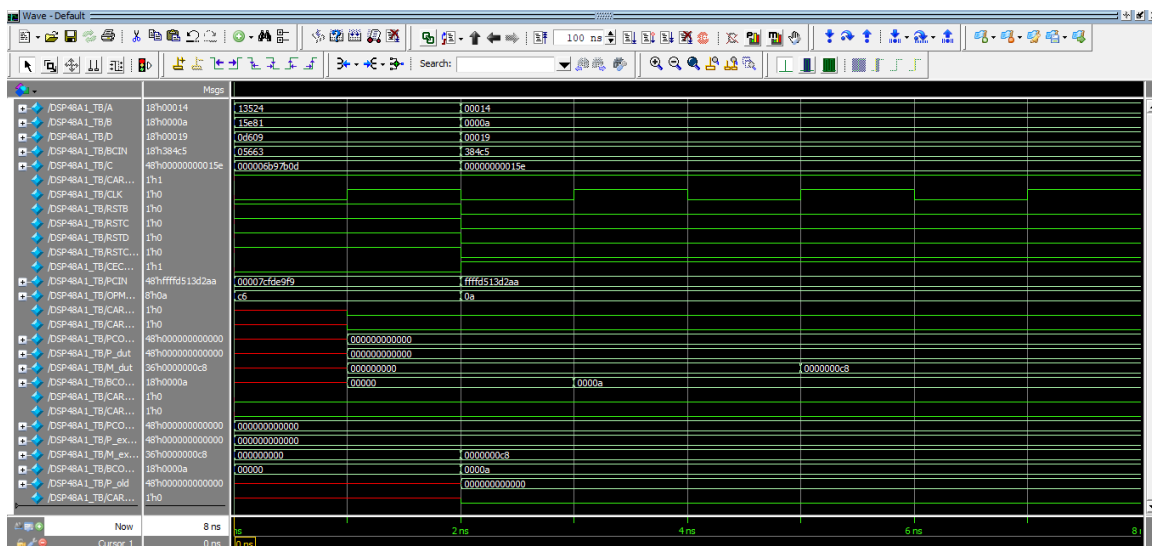
- Transcript

```
# Path 2 test Passed for P
# Path 2 test Passed for M
# Path 2 test Passed for BCOUT
# Path 2 test Passed for PCOUT
# Path 2 test Passed for CARRYOUT
# Path 2 test Passed for CARRYOUTF
# ** Note: $stop      : DSP48A1_TB.v(215)
#   Time: 8 ns   Iteration: 1   Instance: /DSP48A1_TB
# Break in Module DSP48A1_TB at DSP48A1_TB.v line 215
```

2.4. Verify DSP Path 3



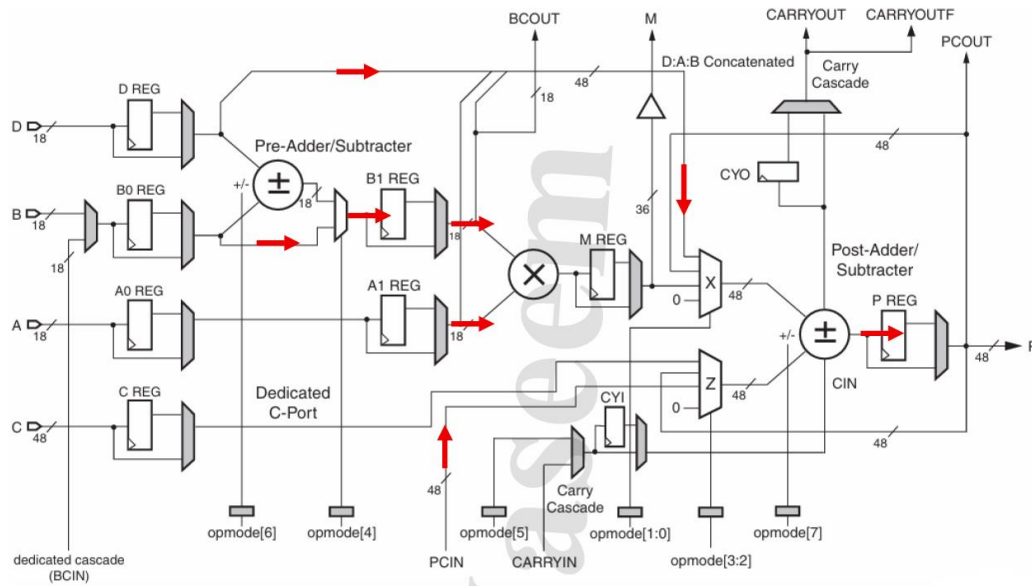
• Waveform



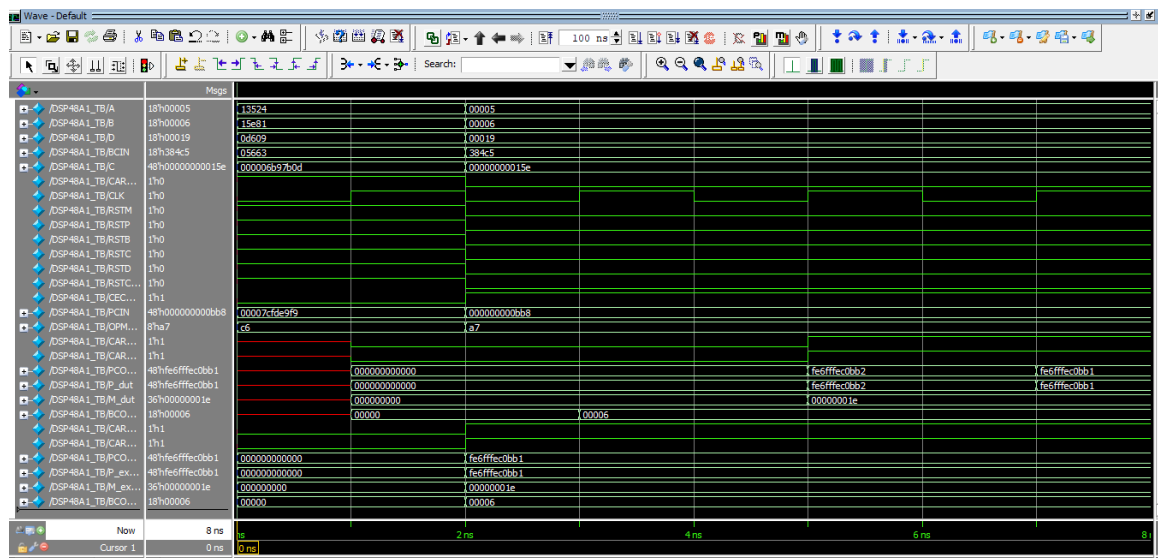
• Transcript

```
# Path 3 test Passed for P
# Path 3 test Passed for M
# Path 3 test Passed for BCOUT
# Path 3 test Passed for PCOUT
# Path 3 test Passed for CARRYOUT
# Path 3 test Passed for CARRYOUTF
# ** Note: $stop      : DSP48A1_TB.v(264)
#   Time: 8 ns   Iteration: 1   Instance: /DSP48A1_TB
# Break in Module DSP48A1_TB at DSP48A1_TB.v line 264
```

2.5. Verify DSP Path 4



• Waveform



• Transcript

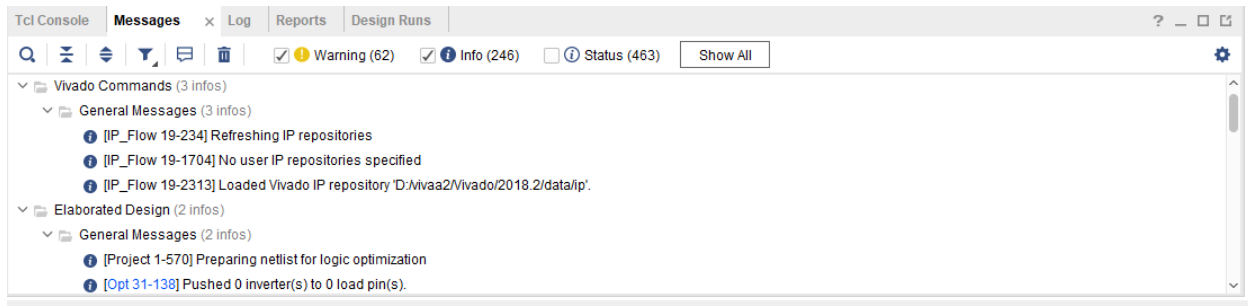
```
# Path 4 test Passed for P
# Path 4 test Passed for M
# Path 4 test Passed for BCOUT
# Path 4 test Passed for PCOUT
# Path 4 test Passed for CARRYOUT
# Path 4 test Passed for CARRYOUTF
# ** Note: $stop      : DSP48A1_TB.v(310)
#   Time: 8 ns   Iteration: 1   Instance: /DSP48A1_TB
# Break in Module DSP48A1_TB at DSP48A1_TB.v line 310
```

Constraint File

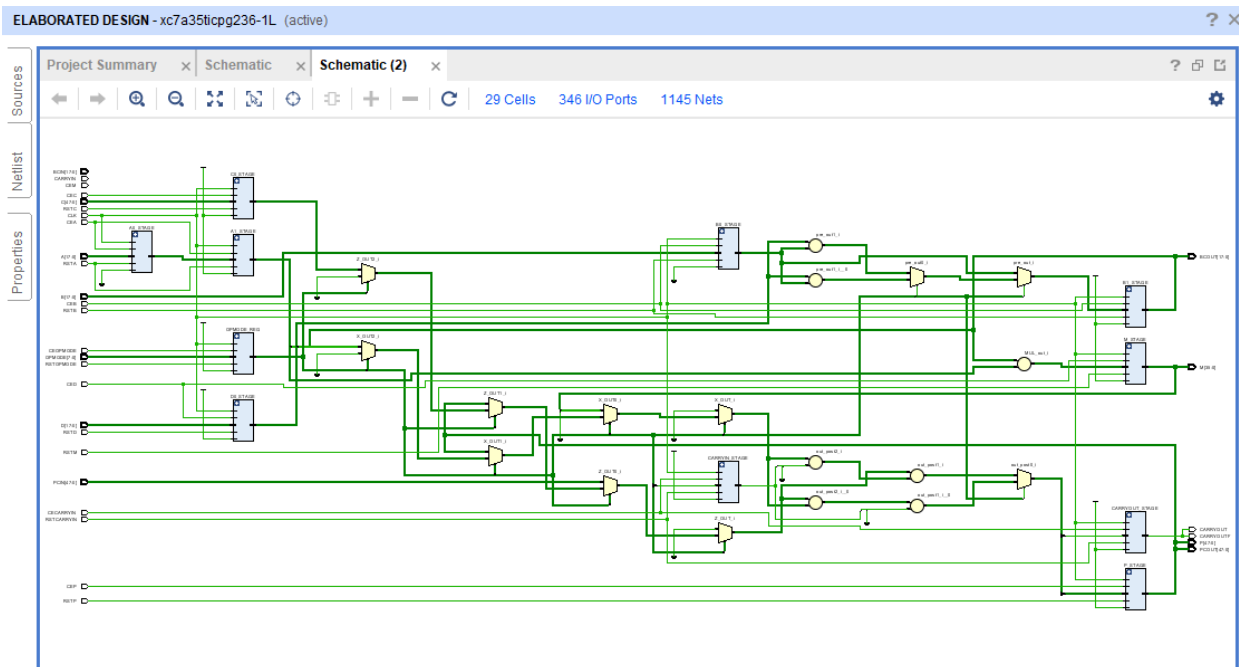
```
## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports CLK]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports CLK]
## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCC0 [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 snippets



Synthesis

• Messages

Tcl Console | Messages | Log | Reports | Design Runs | Timing | Utilization | Debug

Warning (81) Info (77) Status (35) Show All

- Vivado Commands (3 Infos)
 - General Messages (3 Infos)
 - [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:/viva2/Vivado/2018.2/data/ip'.
- Elaborated Design (21 warnings, 16 infos)
 - General Messages (21 warnings, 16 infos)
 - [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
 - [Synth 8-6157] synthesizing module 'DSP48A1' [DSP48A1.v:1] (5 more like this)

• Utilization report

Tcl Console Messages Log Reports Design Runs Utilization x Debug						
Hierarchy						
Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)	
▼ DSP48A1	230	160	1	326	1	
A1_STAGE (REG_MUX...	0	18	0	0	0	
B1_STAGE (REG_MUX...	0	18	0	0	0	
C0_STAGE (REG_MU...	0	48	0	0	0	

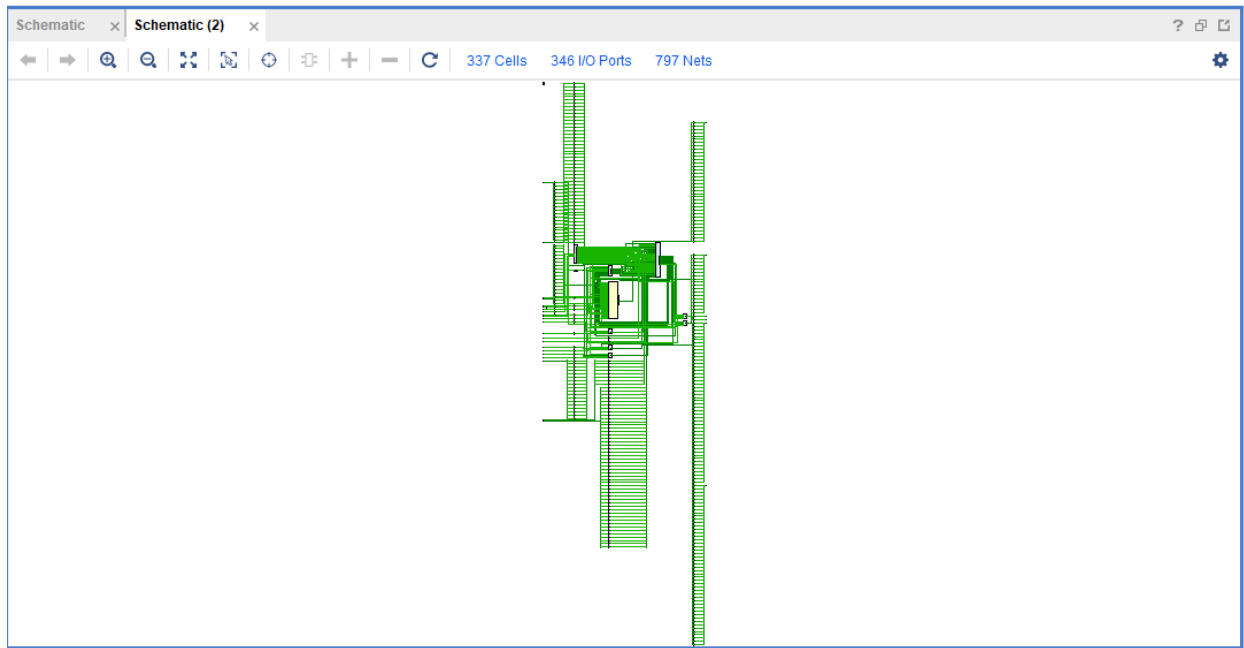
utilization_1

• Timing report

Tcl Console Messages Log Reports Design Runs Timing x Utilization Debug						
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: 162				

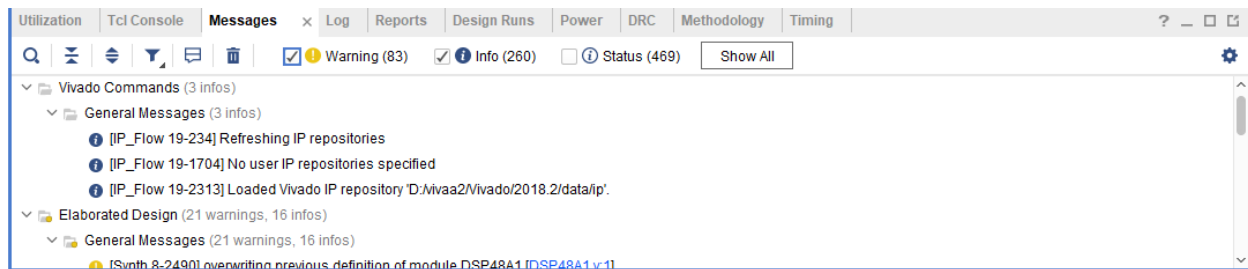
Timing Summary - timing_1

- Schematic snippets



Implementation

- Messages



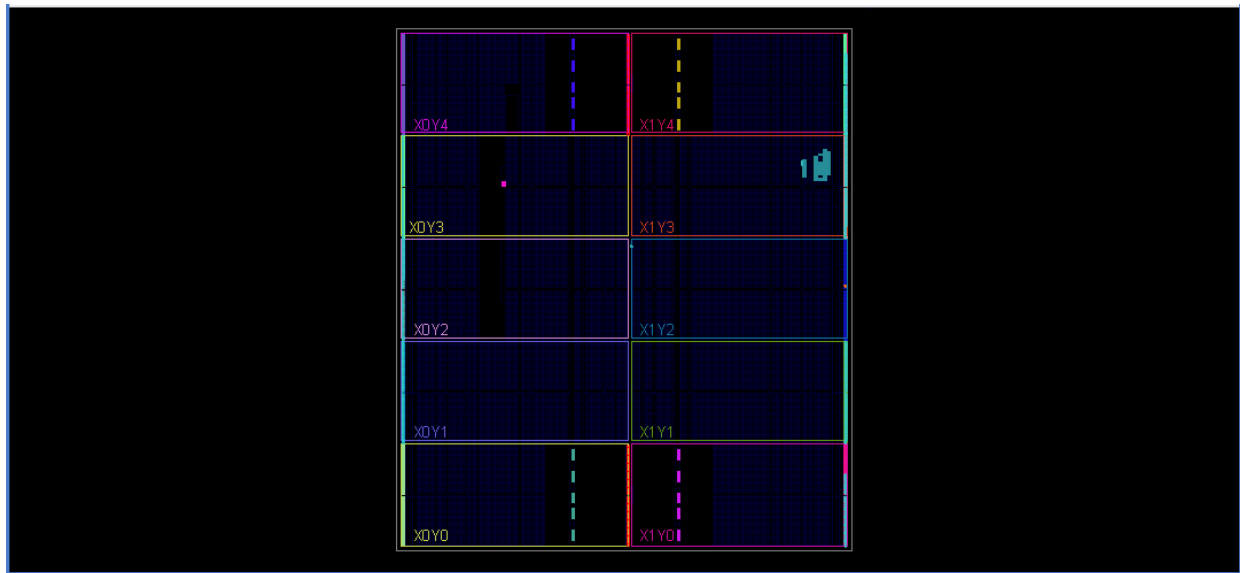
- Utilization report

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)
DSP48A1	229	179	98	229	50	1	326	1
A1_STAGE (REG_MUX...	0	18	7	0	0	0	0	0
B1_STAGE (REG_MUX...	0	36	11	0	0	0	0	0

- Timing report

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 3.643 ns	Worst Hold Slack (WHS): 0.274 ns	Worst Pulse Width Slack (WPWS): 4.500
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0

- Device snippets



Linting

Details (Wrap)

```

Check : mux_select_const
Module : REG_MUX
Signal(s) : C0_STAGE.out
Reference : D:/DSP48A1/DSP48A1.v, Line 137
Message : Constant value drives mux select pin. Signal C0_STAGE.out, Module REG_MUX, File D:/
          DSP48A1/DSP48A1.v, Line 137.
Status : uninspected
Owner : unassigned
  
```

Lint Summary

(Type Search Text (Press Enter))

Name	Count
Resolved(verified, fixed, waived)	10
Warning	6
Info	4

Lint Checks

Filter: Type here

Waived Fixed Pending Uninspected Bug Verified Total : 0 Selected : 0

Severity	Status	Check	Alias	Message	Module	Category
----------	--------	-------	-------	---------	--------	----------

Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard

D:/DSP48A1/DSP48A1.v [DSP48A1.M_STAGE]