TEAM: VIPER

Project 1 - DSP48A1

Nouran Hamdy Mohamed
Abdelrahman Mostafa Shawky Mady
Nardeen Hishmat Ageeb

RTL code

```verilog
module buffer(clk, D, Q);
    parameter WIDTH = 1;
    input clk;
    input [WIDTH-1:0] D;
    output reg [WIDTH-1:0] Q;

    always @(posedge clk) begin
        Q <= D;
    end
endmodule
```

```
module reg_Sync(clk, rst, CE, D, Q);
    parameter WIDTH = 1;
    input clk, rst, CE;
    input [WIDTH-1:0] D;
    output reg [WIDTH-1:0] Q;

    always @(posedge clk) begin
        if(rst) Q <= 0;
        else if(CE) Q <= D;
    end
endmodule
```

```
module reg_Async(clk, rst, CE, D, Q);
    parameter WIDTH = 1;
    input clk, rst, CE;
    input [WIDTH-1:0] D;
    output reg [WIDTH-1:0] Q;

    always @(posedge clk or posedge rst) begin
        if(rst) Q <= 0;
        else if(CE) Q <= D;
    end
endmodule
```

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

    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";
    parameter B_INPUT = "DIRECT";
    parameter RSTTYPE = "SYNC";

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

    output [17:0] BCOUT;
    output [47:0] PCOUT;
    output [47:0] P;
    output [35:0] M;
    output CARRYOUT;
    output CARRYOUTF;

    reg [17:0] B0reg_in, pre_out, B1reg_in;
    wire [17:0] Dreg_out, B0reg_out, A0reg_out, B1reg_out, A1reg_out, Dmux_out, B0mux_out, A0mux_out, B1mux_out, A1mux_out;
    wire [35:0] mult_out, Mreg_out, Mmux_out;
    reg [47:0] X_out, Z_out, post_out;
    wire [47:0] Creg_out, Cmux_out, Preg_out;
    wire [7:0] OPMODEreg_out, OPMODE_out;
    reg carrycsc_out, post_carryout;
    wire CYI_out, CIN, CYO_out;
    wire [47:0] D_A_B;

    generate
        if(RSTTYPE == "SYNC") begin
            reg_Sync #(8) OPMODEreg_sync(CLK, RSTOPMODE, CEOPMODE, OPMODE, OPMODEreg_out);
            reg_Sync #(18) Dreg_sync(CLK, RSTD, CED, D, Dreg_out);
            reg_Sync #(18) B0reg_sync(CLK, RSTB, CEB, B0reg_in, B0reg_out);
            reg_Sync #(18) A0reg_sync(CLK, RSTA, CEA, A, A0reg_out);
            reg_Sync #(48) Creg_sync(CLK, RSTC, CEC, C, Creg_out);
            reg_Sync #(18) B1reg_sync(CLK, RSTB, CEB, B1reg_in, B1reg_out);
            reg_Sync #(18) A1reg_sync(CLK, RSTA, CEA, A0mux_out, A1reg_out);
            reg_Sync #(36) Mreg_sync(CLK, RSTM, CEM, mult_out, Mreg_out);
            reg_Sync #(1) CARRYINreg_sync(CLK, RSTCARRYIN, CECARRYIN, carrycsc_out, CYI_out);
            reg_Sync #(1) CARRYOUTreg_sync(CLK, RSTCARRYIN, CECARRYIN, post_carryout, CYO_out);
            reg_Sync #(48) Preg_sync(CLK, RSTP, CEP, post_out, Preg_out);
        end
        else if(RSTTYPE == "ASYNC") begin
            reg_Async #(8) OPMODEreg_async(CLK, RSTOPMODE, CEOPMODE, OPMODE, OPMODEreg_out);
            reg_Async #(18) Dreg_async(CLK, RSTD, CED, D, Dreg_out);
            reg_Async #(18) B0reg_async(CLK, RSTB, CEB, B0reg_in, B0reg_out);
            reg_Async #(18) A0reg_async(CLK, RSTA, CEA, A, A0reg_out);
            reg_Async #(48) Creg_async(CLK, RSTC, CEC, C, Creg_out);
            reg_Async #(18) B1reg_async(CLK, RSTB, CEB, B1reg_in, B1reg_out);
            reg_Async #(18) A1reg_async(CLK, RSTA, CEA, A0mux_out, A1reg_out);
            reg_Async #(36) Mreg_async(CLK, RSTM, CEM, mult_out, Mreg_out);
            reg_Async #(1) CARRYINreg_async(CLK, RSTCARRYIN, CECARRYIN, carrycsc_out, CYI_out);
            reg_Async #(1) CARRYOUTreg_async(CLK, RSTCARRYIN, CECARRYIN, post_carryout, CYO_out);
            reg_Async #(48) Preg_async(CLK, RSTP, CEP, post_out, Preg_out);
        end

    endgenerate

    //Internal Signals
    assign OPMODE_out = (OPMODEREG) ? OPMODEreg_out : OPMODE;
    assign Dmux_out = (DREG) ? Dreg_out : D;
    assign B0mux_out = (B0REG) ? B0reg_out : B0reg_in;
    assign A0mux_out = (A0REG) ? A0reg_out : A;
    assign Cmux_out = (CREG) ? Creg_out : C;

    assign B1mux_out = (B1REG) ? B1reg_out : B1reg_in;
    assign A1mux_out = (A1REG) ? A1reg_out : A0mux_out;

    assign Mmux_out = (MREG) ? Mreg_out : mult_out;

    assign CIN = (CARRYINREG) ? CYI_out : carrycsc_out;

    assign D_A_B = {Dmux_out[11:0], A1mux_out[17:0], B1mux_out[17:0]};
    assign mult_out = B1mux_out * A1mux_out;

    //Outputs
    assign BCOUT = B1mux_out;
    buffer #(36) M_out(CLK, Mmux_out, M);
    assign CARRYOUT = (CARRYINREG) ? CYO_out : post_carryout;
    assign CARRYOUTF = CARRYOUT;
    assign P = (PREG) ? Preg_out : post_out;
    assign PCOUT = P;


    always @(*) begin
        //B_input
        if(B_INPUT == "DIRECT") B0reg_in = B;
        else if(B_INPUT == "CASCADE") B0reg_in = BCIN;
        else B0reg_in = 0;

        //Pre-Adder/Subtracter
        if(OPMODE_out[6]) pre_out = Dmux_out - B0mux_out;
        else pre_out = Dmux_out + B0mux_out;

        //B1_REG_input
        if(OPMODE_out[4]) B1reg_in = pre_out;
        else B1reg_in = B0mux_out;

        //Carry_Cascade
        if(CARRYINSEL == "OPMODE5") carrycsc_out = OPMODE_out[5];
        else if(CARRYINSEL == "CARRYIN") carrycsc_out = CARRYIN;
        else carrycsc_out = 0;

        //X_Multiplexer
        case(OPMODE_out[1:0])
        2'b00: X_out = 0;
        2'b01: X_out = {{12{Mmux_out[35]}}, Mmux_out};
        2'b10: X_out = P;
        2'b11: X_out = D_A_B;
        endcase

        //Z_Multiplexer
        case(OPMODE_out[3:2])
        2'b00: Z_out = 0;
        2'b01: Z_out = PCIN;
        2'b10: Z_out = P;
        2'b11: Z_out = Cmux_out;
        endcase

        //Post-Adder/Subtracter
        if(OPMODE_out[7]) {post_carryout, post_out} = Z_out - (X_out + CIN);
        else {post_carryout, post_out} = Z_out + X_out + CIN;
    end

endmodule
```

TestBench

```
// dsp test bench ,
module dsp_tb (); // default parameters

    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";
  parameter B_INPUT = "DIRECT";
  parameter RSTTYPE = "SYNC";



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

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

  reg [17:0] ex_BCOUT;
  reg [47:0] ex_P;
  reg [35:0] ex_M;
  reg ex_CARRYOUT;


  // temporary wires
  reg [17:0] t_pre;
  reg [47:0] t_post;
  reg [35:0] t_multi;
  reg [47:0] t_multi_ext;



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

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

integer i;
```

```
initial begin
// reset inputs
  RSTA=1;
  RSTB=1;
  RSTM=1;
  RSTP=1;
  RSTC=1;
  RSTD=1;
  RSTCARRYIN=1;
  RSTOPMODE=1;
// control enable inputs
  CEA=1;
  CEB=1;
  CEM=1;
  CEP=1;
  CEC=1;
  CED=1;
  CECARRYIN=1;
  CEOPMODE=1;
// data inputs
  CARRYIN=1'b0;
  A=18'b0;
  B=18'b0;
  D=18'b0;
  C=48'b0;
  BCIN=18'b0;
  PCIN=48'b0;
// opmode
  OPMODE=8'b0000_00_00;
  #30;
  // unreset inputs
  RSTA=0;
  RSTB=0;
  RSTM=0;
  RSTP=0;
  RSTC=0;
  RSTD=0;
  RSTCARRYIN=0;
  RSTOPMODE=0;
  #10;
//8'b post-add_pre-add_carry-cascade_B1-REG-input_Z-Multiplexer_X-Multiplexer;

//////////////////////////////// first mode
for(i=0;i<10;i=i+1)begin
OPMODE = 8'b0001_11_01; // post-addition,per-addition,zero-carry,use-pre-add,use-c,use-multi
// data inputs
CARRYIN=1'b1;
  A=$random;
  B=$random;
  D=$random;
  C=$random;
  BCIN=$random;
  PCIN=$random;
// getting expected output
t_pre = D+B;
t_multi = t_pre*A;
t_multi_ext={{12{t_multi[35]}},t_multi};
{ex_CARRYOUT,t_post} = C+(OPMODE[5]+t_multi_ext);
ex_P=t_post;
ex_M=t_multi_ext;
```

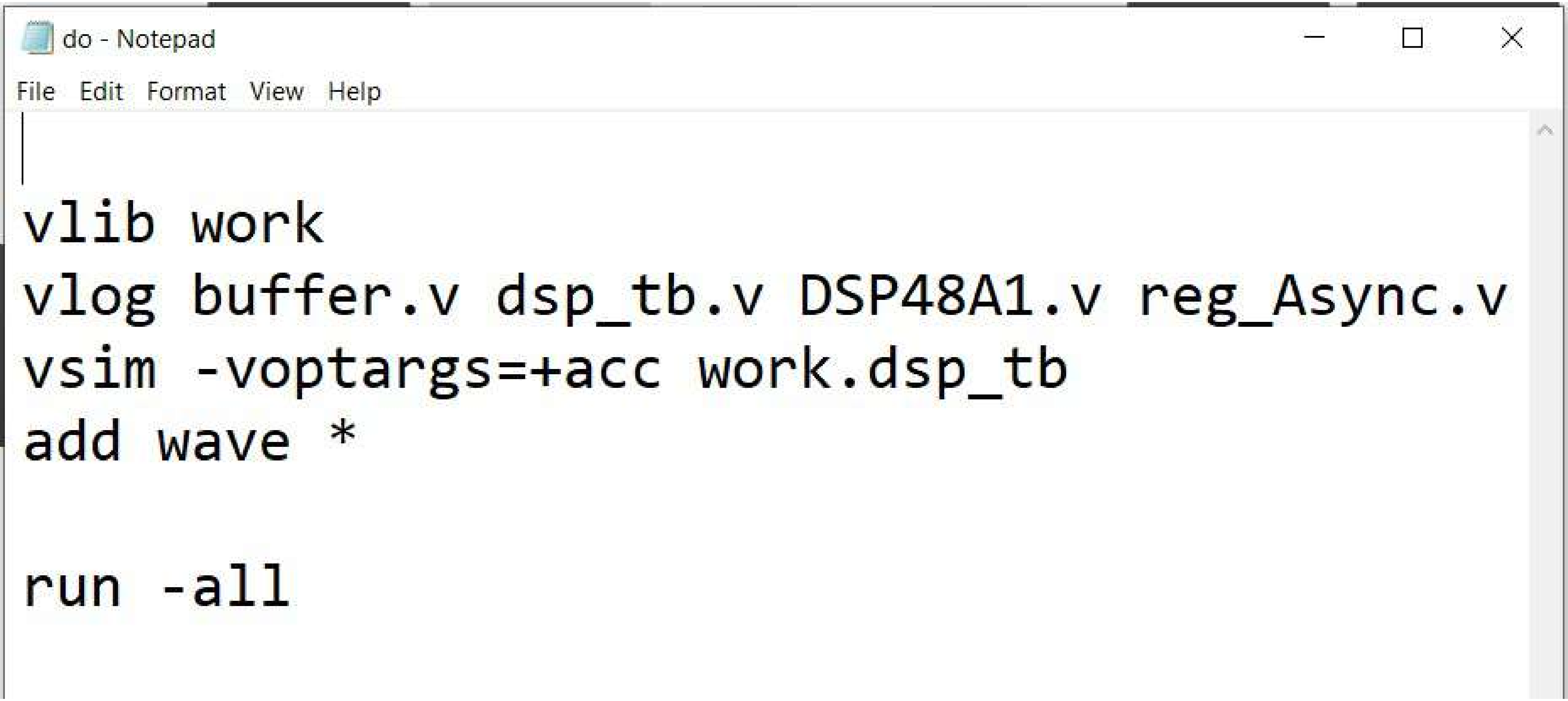
```
ex_BCOUT=t_pre;
#20;
// Comparison
if(BCOUT!=ex_BCOUT)begin
$display("error BCOUT");
$stop;
end
if(P!=ex_P)begin
$display("error P");
$stop;
end
if(M!=ex_M)begin
$display("error M");
$stop;
end
if(CARRYOUT!=ex_CARRYOUT)begin
$display("error CARRYOUT");
$stop;
end
end


//////////////////////////////// second mode
for(i=0;i<10;i=i+1)begin

OPMODE = 8'b1111_11_01; // post-subtraction,per-subtraction,zero-carry,use-pre-add,use-c,use-multi
// data inputs
CARRYIN=1'b1;
  A=$random;
  B=$random;
  D=$random;
  C=$random;
  BCIN=$random;
  PCIN=$random;
// getting expected output
t_pre = D-B;
t_multi = t_pre*A;
t_multi_ext={{12{t_multi[35]}},t_multi};
{ex_CARRYOUT,t_post} = C-(OPMODE[5]+t_multi_ext);
ex_P=t_post;
ex_M=t_multi_ext;
ex_BCOUT=t_pre;
#20;
// Comparison
if(BCOUT!=ex_BCOUT)begin
$display("error BCOUT");
$stop;
end
if(P!=ex_P)begin
$display("error P");
$stop;
end
if(M!=ex_M)begin
$display("error M");
$stop;
end
if(CARRYOUT!=ex_CARRYOUT)begin
$display("error CARRYOUT");
$stop;
end
```

```
end


//////////////////////////////// third mode
for(i=0;i<10;i=i+1)begin

OPMODE = 8'b0001_01_00;
// data inputs
CARRYIN=1'b1;
  A=$random;
  B=$random;
  D=$random;
  C=$random;
  BCIN=$random;
  PCIN=$random;
// getting expected output
t_pre = D+B;
t_multi = t_pre*A;
t_multi_ext={{12{t_multi[35]}},t_multi};
{ex_CARRYOUT,t_post} = PCIN+0;
ex_P=PCIN;
ex_M=t_multi_ext;
ex_BCOUT=t_pre;
#20;
// Comparison
if(BCOUT!=ex_BCOUT)begin
$display("error BCOUT");
$stop;
end
if(P!=ex_P)begin
$display("error P");
$stop;
end
if(M!=ex_M)begin
$display("error M");
$stop;
end
if(CARRYOUT!=ex_CARRYOUT)begin
$display("error CARRYOUT");
$stop;
end
end


//////////////////////////////// fourth mode
for(i=0;i<10;i=i+1)begin

OPMODE = 8'b0000_00_01;
// data inputs
CARRYIN=1'b1;
  A=$random;
  B=$random;
  D=$random;
  C=$random;
  BCIN=$random;
  PCIN=$random;
// getting expected output
t_multi = B*A;
t_multi_ext={{12{t_multi[35]}},t_multi};
{ex_CARRYOUT,t_post} = t_multi_ext+0;
```

```
ex_P=t_post;
ex_M=t_multi_ext;
ex_BCOUT=B;
#20;
// Comparison
if(BCOUT!=ex_BCOUT)begin
$display("error BCOUT");
$stop;
end
if(P!=ex_P)begin
$display("error P");
$stop;
end
if(M!=ex_M)begin
$display("error M");
$stop;
end
if(CARRYOUT!=ex_CARRYOUT)begin
$display("error CARRYOUT");
$stop;
end
end


//////////////////////////////// fifth mode

for(i=0;i<10;i=i+1)begin

OPMODE = 8'b0000_10_11;


// data inputs
CARRYIN=1'b1;
  A=$random;
  B=$random;
  D=$random;
  C=$random;
  BCIN=$random;
  PCIN=$random;
#20;
end


//////////////////////////////// sixth mode

for(i=0;i<10;i=i+1)begin

OPMODE = 8'b0001_11_10;
CARRYIN=1'b1;


  A=$random;
  B=$random;
  D=$random;
  C=$random;
  BCIN=$random;
  PCIN=$random;
#20;
end
```

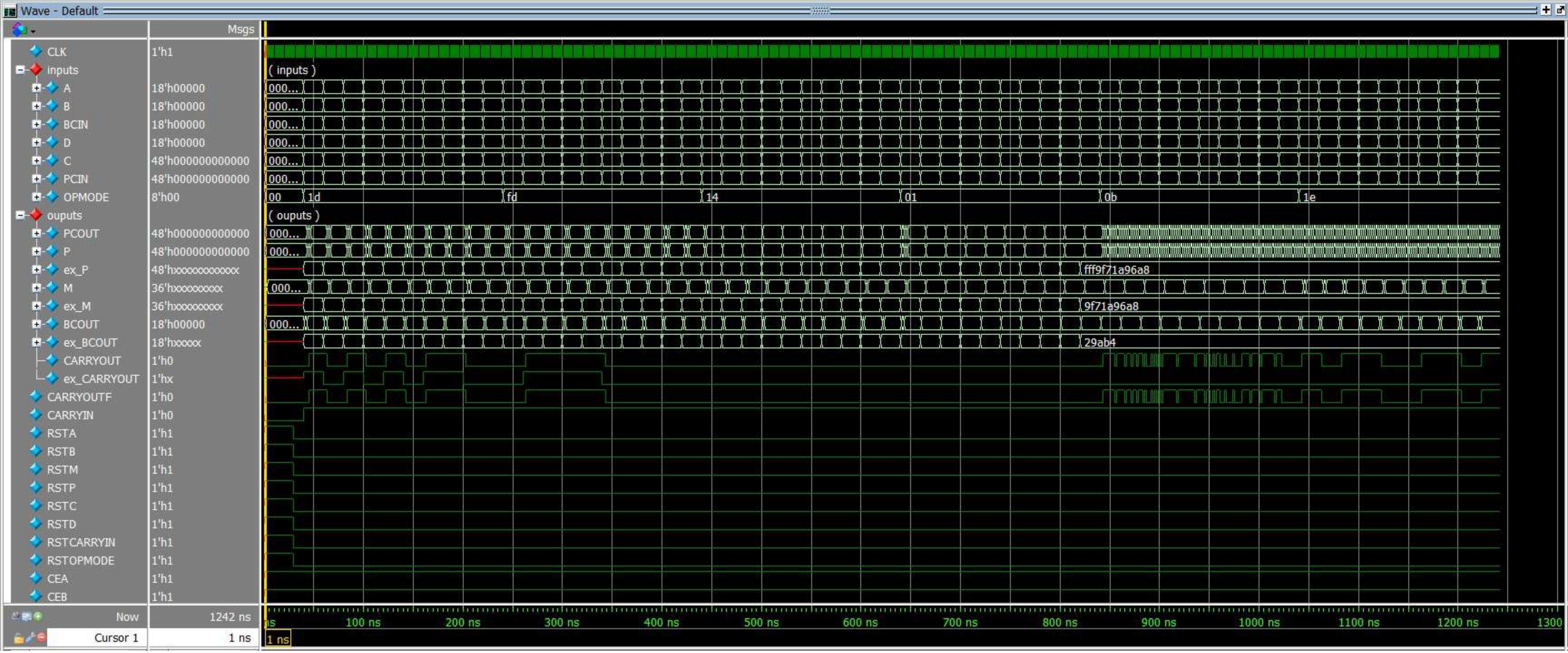
```
#2 $stop;
end
endmodule
```

Do file



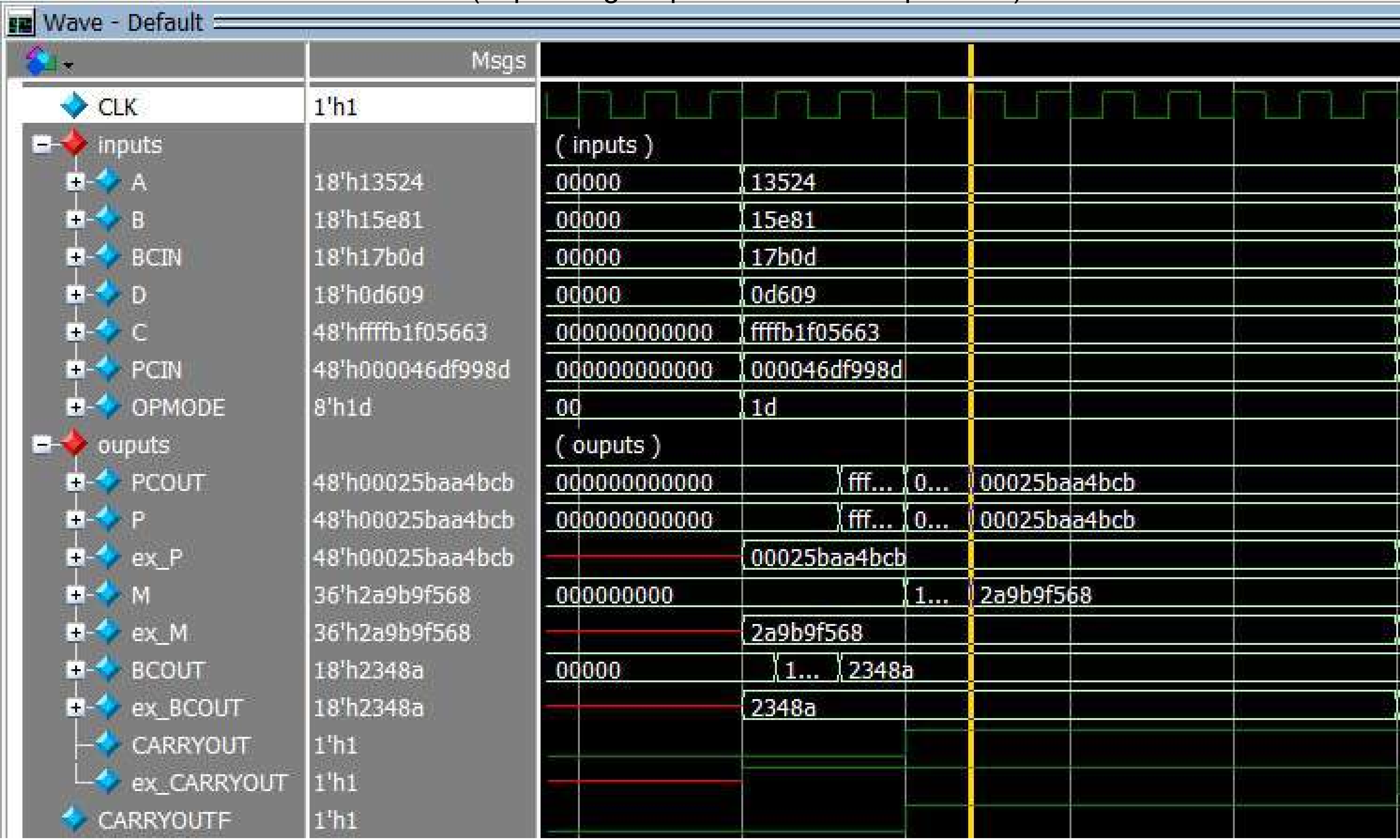
```
vlib work
vlog buffer.v dsp_tb.v DSP48A1.v reg_Async.v
vsim -voptargs=+acc work.dsp_tb
add wave *

run -all
```

Questa Sim snippets



```
Wave - Default
                     Msgs
CLK                  1'h1
inputs
  A                  18'h00000
  B                  18'h00000
  BCIN               18'h00000
  D                  18'h00000
  C                  48'h000000000000
  PCIN               48'h000000000000
  OPMODE             8'h00
ouputs
  PCOUT              48'h000000000000
  P                  48'h000000000000
  ex_P               48'hxxxxxxxxxxxx
  M                  36'hxxxxxxxxx
  ex_M               36'hxxxxxxxxx
  BCOUT              18'h00000
  ex_BCOUT           18'hxxxxx
  CARRYOUT           1'h0
  ex_CARRYOUT        1'hx
CARRYOUTF            1'h0
CARRYIN              1'h0
RSTA                 1'h1
RSTB                 1'h1
RSTM                 1'h1
RSTP                 1'h1
RSTC                 1'h1
RSTD                 1'h1
RSTCARRYIN           1'h1
RSTOPMODE            1'h1
CEA                  1'h1
CEB                  1'h1
Now                  1242 ns
Cursor 1             1 ns
```

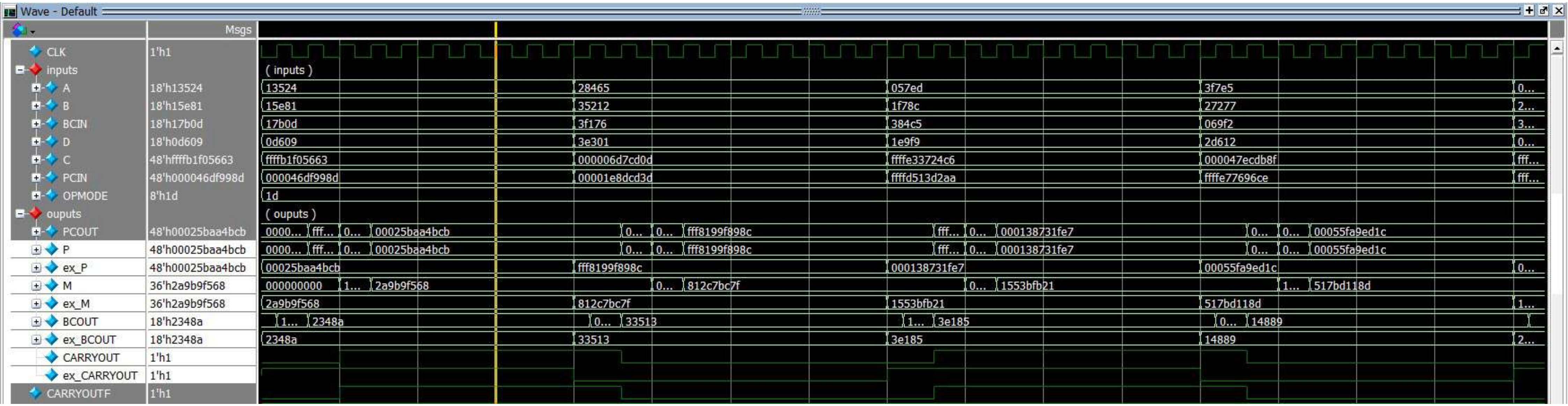
The correct output is after 4 clk cycles.
(depending on parameters and op mode)



Wave - Default

	Msgs
CLK	1'h1
inputs	
A	18'h13524
B	18'h15e81
BCIN	18'h17b0d
D	18'h0d609
C	48'hffffb1f05663
PCIN	48'h000046df998d
OPMODE	8'h1d
ouputs	
PCOUT	48'h00025baa4bcb
P	48'h00025baa4bcb
ex_P	48'h00025baa4bcb
M	36'h2a9b9f568
ex_M	36'h2a9b9f568
BCOUT	18'h2348a
ex_BCOUT	18'h2348a
CARRYOUT	1'h1
ex_CARRYOUT	1'h1
CARRYOUTF	1'h1

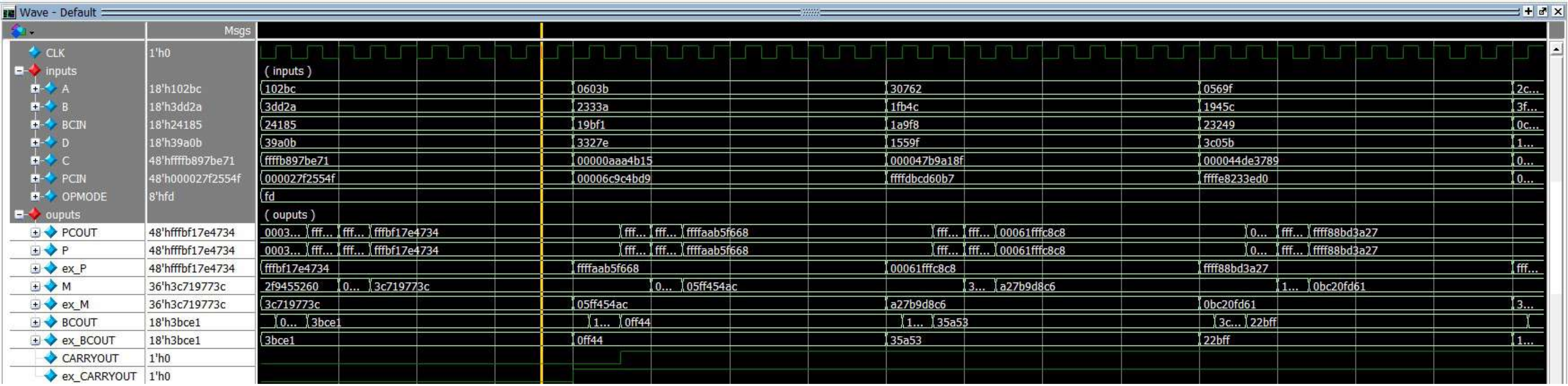
testbench mode 1


Wave - Default
Msgs
(inputs)
13524 | 28465 | 057ed | 3f7e5 | 0...
15e81 | 35212 | 1f78c | 27277 | 2...
17b0d | 3f176 | 384c5 | 069f2 | 3...
0d609 | 3e301 | 1e9f9 | 2d612 | 0...
ffffb1f05663 | 000006d7cd0d | fffe33724c6 | 000047ecdb8f | fff...
000046df998d | 00001e8dcd3d | ffffd513d2aa | ffffe77696ce | fff...
1d
(ouputs)
0000... | fff... | 0... | 00025baa4bcb | 0... | 0... | fff8199f898c | fff... | 0... | 000138731fe7 | 0... | 0... | 00055fa9ed1c
0000... | fff... | 0... | 00025baa4bcb | 0... | 0... | fff8199f898c | fff... | 0... | 000138731fe7 | 0... | 0... | 00055fa9ed1c
00025baa4bcb | fff8199f898c | 000138731fe7 | 00055fa9ed1c | 0...
000000000 | 1... | 2a9b9f568 | 0... | 812c7bc7f | 0... | 1553bfb21 | 1... | 517bd118d
2a9b9f568 | 812c7bc7f | 1553bfb21 | 517bd118d | 1...
1... | 2348a | 0... | 33513 | 1... | 3e185 | 0... | 14889
2348a | 33513 | 3e185 | 14889 | 2...

CLK	1'h1
inputs	
A	18'h13524
B	18'h15e81
BCIN	18'h17b0d
D	18'h0d609
C	48'hffffb1f05663
PCIN	48'h000046df998d
OPMODE	8'h1d
ouputs	
PCOUT	48'h00025baa4bcb
P	48'h00025baa4bcb
ex_P	48'h00025baa4bcb
M	36'h2a9b9f568
ex_M	36'h2a9b9f568
BCOUT	18'h2348a
ex_BCOUT	18'h2348a
CARRYOUT	1'h1
ex_CARRYOUT	1'h1
CARRYOUTF	1'h1

OPMODE 1 = 8'b0001_11_01;
OPMODE[1:0] >> X input: multiplier
OPMODE[3:2] >> Z input: C port
OPMODE[4] >> use pre-adder or pre-subtracter
OPMODE[5] >> force carry = 0
OPMODE[6] >> specifies pre-adder
OPMODE[7] >> specifies post-adder

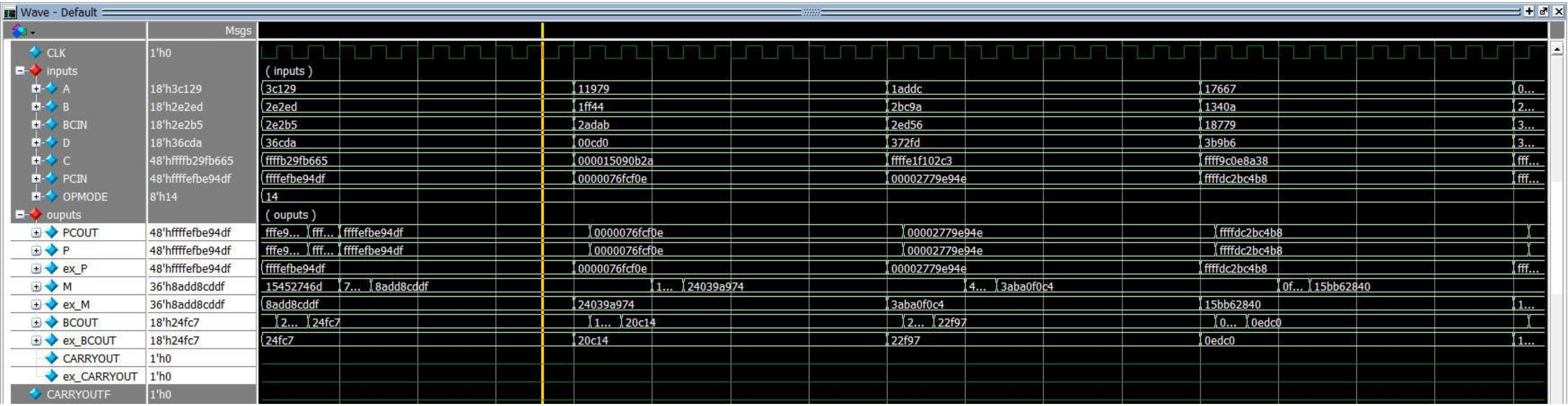
testbench mode 2



Signal	Msgs
CLK	1'h0
inputs	
A	18'h102bc
B	18'h3dd2a
BCIN	18'h24185
D	18'h39a0b
C	48'hffffb897be71
PCIN	48'h000027f2554f
OPMODE	8'hfd
ouputs	
PCOUT	48'hfffbf17e4734
P	48'hfffbf17e4734
ex_P	48'hfffbf17e4734
M	36'h3c719773c
ex_M	36'h3c719773c
BCOUT	18'h3bce1
ex_BCOUT	18'h3bce1
CARRYOUT	1'h0
ex_CARRYOUT	1'h0

OPMODE 2 = 8'b1111_11_01;
OPMODE[1:0] >> X input: multiplier
OPMODE[3:2] >> Z input: C port
OPMODE[4] >> use pre-adder or pre-subtracter
OPMODE[5] >> force carry = 1
OPMODE[6] >> specifies pre-subtractor
OPMODE[7] >> specifies post-subtractor

testbench mode 3



```
Wave - Default
                 Msgs
CLK              1'h0
inputs                            ( inputs )
  A              18'h3c129        3c129           11979           1addc           17667           0...
  B              18'h2e2ed        2e2ed           1ff44           2bc9a           1340a           2...
  BCIN           18'h2e2b5        2e2b5           2adab           2ed56           18779           3...
  D              18'h36cda        36cda           00cd0           372fd           3b9b6           3...
  C              48'hffffb29fb665 ffffb29fb665    000015090b2a    ffffe1f102c3    ffff9c0e8a38    fff...
  PCIN           48'hffffefbe94df ffffefbe94df    0000076fcf0e    00002779e94e    ffffdc2bc4b8    fff...
  OPMODE         8'h14            14
ouputs                            ( ouputs )
  PCOUT          48'hffffefbe94df fffe9... fff... ffffefbe94df  0000076fcf0e  00002779e94e  ffffdc2bc4b8
  P              48'hffffefbe94df fffe9... fff... ffffefbe94df  0000076fcf0e  00002779e94e  ffffdc2bc4b8
  ex_P           48'hffffefbe94df ffffefbe94df    0000076fcf0e    00002779e94e    ffffdc2bc4b8    fff...
  M              36'h8add8cddf    15452746d  7...  8add8cddf  1...  24039a974  4...  3aba0f0c4  0f...  15bb62840
  ex_M           36'h8add8cddf    8add8cddf       24039a974       3aba0f0c4       15bb62840       1...
  BCOUT          18'h24fc7        2...  24fc7     1...  20c14     2...  22f97     0...  0edc0
  ex_BCOUT       18'h24fc7        24fc7           20c14           22f97           0edc0           1...
  CARRYOUT       1'h0
  ex_CARRYOUT    1'h0
CARRYOUTF        1'h0
```

OPMODE 3 = 8'b0001_01_00;

OPMODE[1:0] >> X input: Zeros

OPMODE[3:2] >> Z input: use PCIN

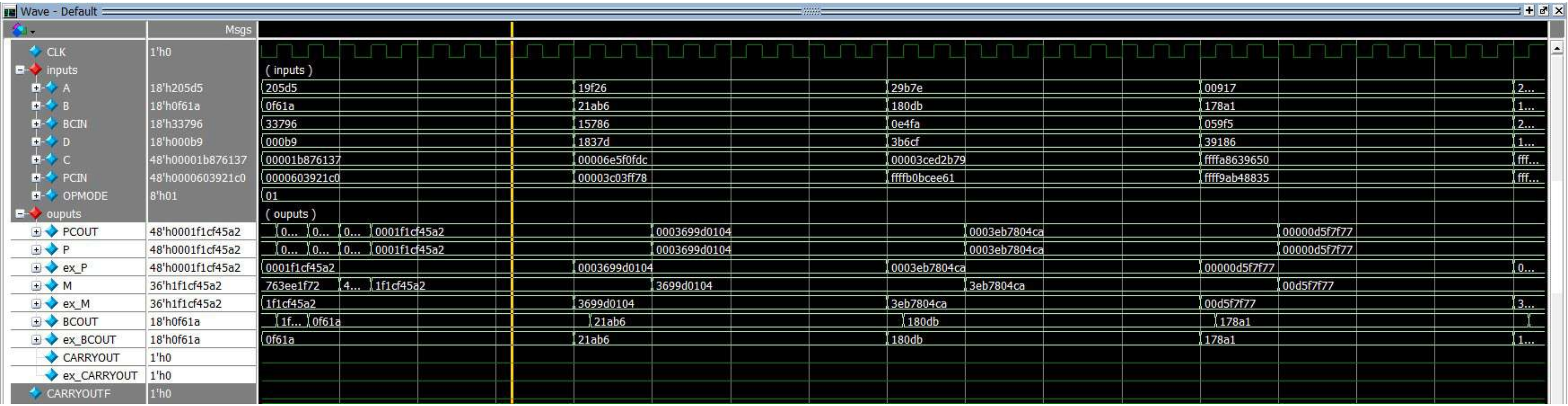
OPMODE[4] >> use pre-adder or pre-subtracter

OPMODE[5] >> force carry = 0

OPMODE[6] >> specifies pre-adder

OPMODE[7] >> specifies post-adder

testbench mode 4



Wave - Default

	Msgs
CLK	1'h0
inputs	
A	18'h205d5
B	18'h0f61a
BCIN	18'h33796
D	18'h000b9
C	48'h00001b876137
PCIN	48'h0000603921c0
OPMODE	8'h01
ouputs	
PCOUT	48'h0001f1cf45a2
P	48'h0001f1cf45a2
ex_P	48'h0001f1cf45a2
M	36'h1f1cf45a2
ex_M	36'h1f1cf45a2
BCOUT	18'h0f61a
ex_BCOUT	18'h0f61a
CARRYOUT	1'h0
ex_CARRYOUT	1'h0
CARRYOUTF	1'h0

(inputs)
205d5 19f26 29b7e 00917 2...
0f61a 21ab6 180db 178a1 1...
33796 15786 0e4fa 059f5 2...
000b9 1837d 3b6cf 39186 1...
00001b876137 00006e5f0fdc 00003ced2b79 ffffa8639650 fff...
0000603921c0 00003c03ff78 ffffb0bcee61 ffff9ab48835 fff...
01
(ouputs)
0... 0... 0... 0001f1cf45a2 0003699d0104 0003eb7804ca 00000d5f7f77
0... 0... 0... 0001f1cf45a2 0003699d0104 0003eb7804ca 00000d5f7f77
0001f1cf45a2 0003699d0104 0003eb7804ca 00000d5f7f77 0...
763ee1f72 4... 1f1cf45a2 3699d0104 3eb7804ca 00d5f7f77
1f1cf45a2 3699d0104 3eb7804ca 00d5f7f77 3...
1f... 0f61a 21ab6 180db 178a1
0f61a 21ab6 180db 178a1 1...

OPMODE 4 = 8'b0000_00_01;

OPMODE[1:0] >> X input: multiplier

OPMODE[3:2] >> Z input: Zeros

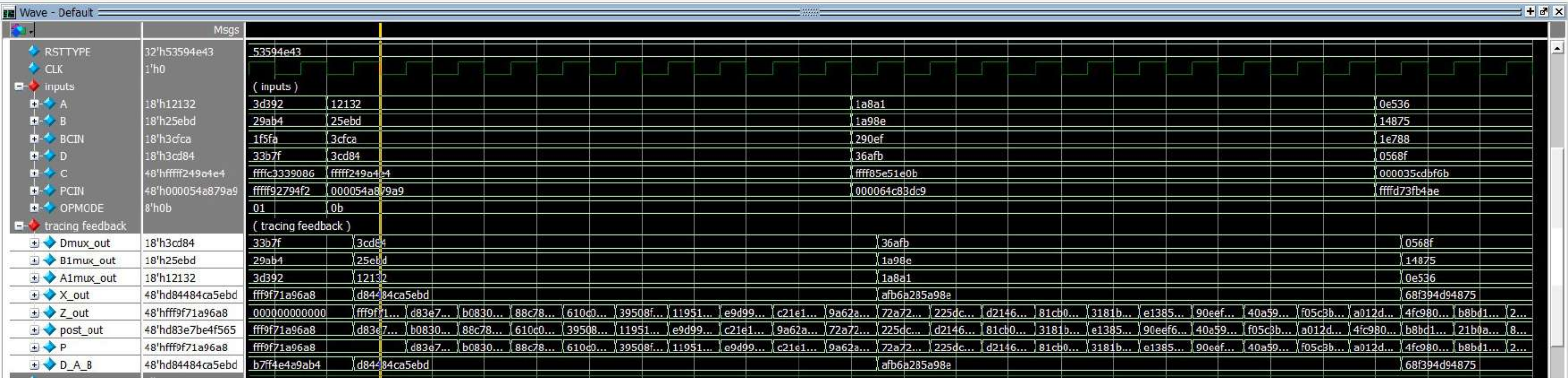
OPMODE[4] >> Bypass pre-adder or pre-subtracter

OPMODE[5] >> force carry = 0

OPMODE[6] >> specifies pre-adder

OPMODE[7] >> specifies post-adder

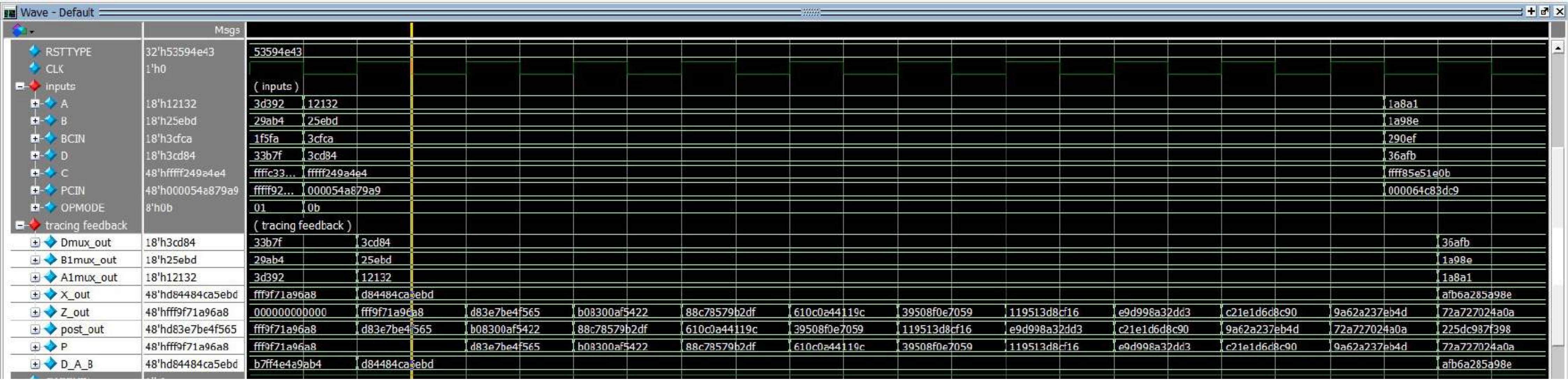
testbench mode 5


Wave - Default
Msgs
53594e43
(inputs)
3d392 12132 1a8a1 0e536
29ab4 25ebd 1a98e 14875
1f5fa 3cfca 290ef 1e788
33b7f 3cd84 36afb 0568f
ffffc3339086 fffff249a4e4 ffff85e51e0b 000035cdbf6b
fffff92794f2 000054a879a9 000064c83dc9 ffffd73fb4ae
01 0b
(tracing feedback)
33b7f 3cd84 36afb 0568f
29ab4 25ebd 1a98e 14875
3d392 12132 1a8a1 0e536
fff9f71a96a8 d84484ca5ebd afb6a285a98e 68f394d94875
000000000000 fff9f71... d83e7... b0830... 88c78... 610c0... 39508f... 11951... e9d99... c21e1... 9a62a... 72a72... 225dc... d2146... 81cb0... 3181b... e1385... 90eef... 40a59... f05c3b... a012d... 4fc980... b8bd1... 2...
fff9f71a96a8 d83e7... b0830... 88c78... 610c0... 39508... 11951... e9d99... c21e1... 9a62a... 72a72... 225dc... d2146... 81cb0... 3181b... e1385... 90eef6... 40a59... f05c3b... a012d... 4fc980... b8bd1... 21b0a... 8...
fff9f71a96a8 d83e7... b0830... 88c78... 610c0... 39508f... 11951... e9d99... c21e1... 9a62a... 72a72... 225dc... d2146... 81cb0... 3181b... e1385... 90eef... 40a59... f05c3b... a012d... 4fc980... b8bd1... 2...
b7ff4e4a9ab4 d84484ca5ebd afb6a285a98e 68f394d94875

	Msgs
RSTTYPE	32'h53594e43
CLK	1'h0
inputs	
A	18'h12132
B	18'h25ebd
BCIN	18'h3cfca
D	18'h3cd84
C	48'hfffff249a4e4
PCIN	48'h000054a879a9
OPMODE	8'h0b
tracing feedback	
Dmux_out	18'h3cd84
B1mux_out	18'h25ebd
A1mux_out	18'h12132
X_out	48'hd84484ca5ebd
Z_out	48'hfff9f71a96a8
post_out	48'hd83e7be4f565
P	48'hfff9f71a96a8
D_A_B	48'hd84484ca5ebd

OPMODE 5 = 8'b0000_10_11;
OPMODE[1:0] >> X input: use D_A_B
OPMODE[3:2] >> Z input: accumulator P
OPMODE[4] >> Bypass pre-adder or pre-subtracter
OPMODE[5] >> force carry = 0
OPMODE[6] >> specifies pre-adder
OPMODE[7] >> specifies post-adder

testbench mode 5



Wave - Default

	Msgs
RSTTYPE	32'h53594e43
CLK	1'h0
inputs	
A	18'h12132
B	18'h25ebd
BCIN	18'h3cfca
D	18'h3cd84
C	48'hffff249a4e4
PCIN	48'h000054a879a9
OPMODE	8'h0b
tracing feedback	
Dmux_out	18'h3cd84
B1mux_out	18'h25ebd
A1mux_out	18'h12132
X_out	48'hd84484ca5ebd
Z_out	48'hfff9f71a96a8
post_out	48'hd83e7be4f565
P	48'hfff9f71a96a8
D_A_B	48'hd84484ca5ebd

53594e43
(inputs)
3d392 12132 1a8a1
29ab4 25ebd 1a98e
1f5fa 3cfca 290ef
33b7f 3cd84 36afb
ffffc33... fffff249a4e4 ffff85e51e0b
ffff92... 000054a879a9 000064c83dc9
01 0b
(tracing feedback)
33b7f 3cd84 36afb
29ab4 25ebd 1a98e
3d392 12132 1a8a1
fff9f71a96a8 d84484ca5ebd afb6a285a98e
000000000000 fff9f71a96a8 d83e7be4f565 b08300af5422 88c78579b2df 610c0a44119c 39508f0e7059 119513d8cf16 e9d998a32dd3 c21e1d6d8c90 9a62a237eb4d 72a727024a0a
fff9f71a96a8 d83e7be4f565 b08300af5422 88c78579b2df 610c0a44119c 39508f0e7059 119513d8cf16 e9d998a32dd3 c21e1d6d8c90 9a62a237eb4d 72a727024a0a 225dc987f398
fff9f71a96a8 d83e7be4f565 b08300af5422 88c78579b2df 610c0a44119c 39508f0e7059 119513d8cf16 e9d998a32dd3 c21e1d6d8c90 9a62a237eb4d 72a727024a0a
b7ff4e4a9ab4 d84484ca5ebd afb6a285a98e

OPMODE 5 = 8'b0000_10_11;

OPMODE[1:0] >> X input: use D_A_B

OPMODE[3:2] >> Z input: accumulator P

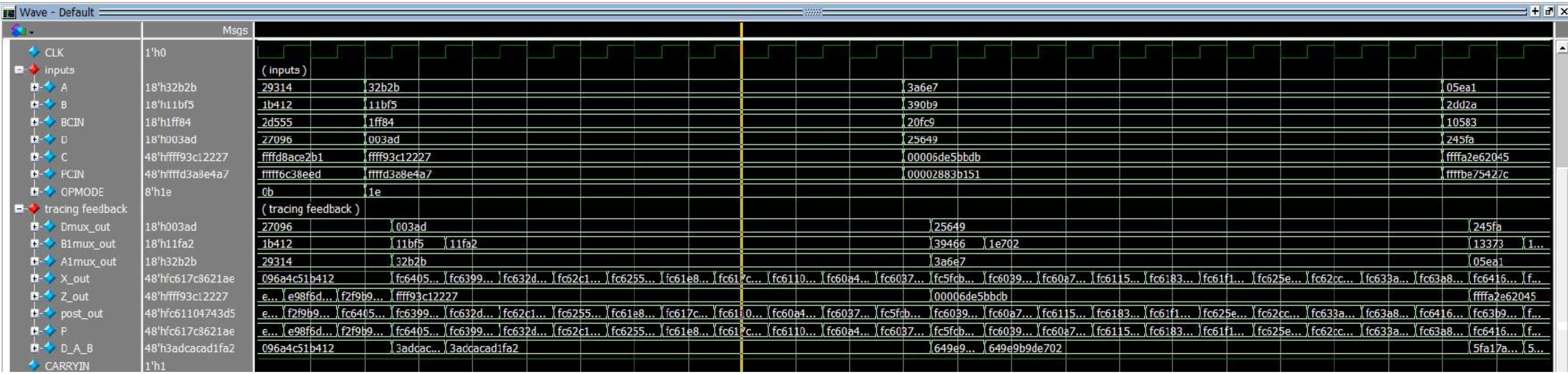
OPMODE[4] >> Bypass pre-adder or pre-subtracter

OPMODE[5] >> force carry = 0

OPMODE[6] >> specifies pre-adder

OPMODE[7] >> specifies post-adder

testbench mode 6



OPMODE 6 = 8'b0001_11_10;

OPMODE[1:0] >> X input: accumulator P

OPMODE[3:2] >> Z input: C port

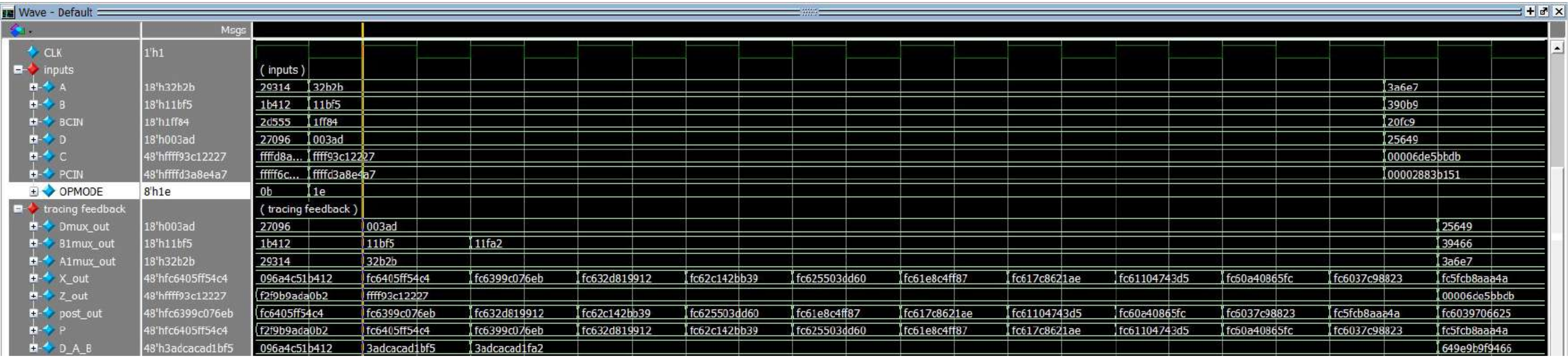
OPMODE[4] >> use pre-adder or pre-subtracter

OPMODE[5] >> force carry = 0

OPMODE[6] >> specifies pre-adder

OPMODE[7] >> specifies post-adder

testbench mode 6



Wave - Default

```
                      Msgs
CLK                   1'h1
inputs                                 ( inputs )
  A                   18'h32b2b        29314 | 32b2b | 3a6e7
  B                   18'h11bf5        1b412 | 11bf5 | 390b9
  BCIN                18'h1ff84        2d555 | 1ff84 | 20fc9
  D                   18'h003ad        27096 | 003ad | 25649
  C                   48'hffff93c12227 ffffd8a... | ffff93c12227 | 00006de5bbdb
  PCIN                48'hffffd3a8e4a7 fffff6c... | ffffd3a8e4a7 | 00002883b151
  OPMODE              8'h1e            0b | 1e
tracing feedback                       ( tracing feedback )
  Dmux_out            18'h003ad        27096 | 003ad | 25649
  B1mux_out           18'h11bf5        1b412 | 11bf5 | 11fa2 | 39466
  A1mux_out           18'h32b2b        29314 | 32b2b | 3a6e7
  X_out               48'hfc6405ff54c4 096a4c51b412 | fc6405ff54c4 | fc6399c076eb | fc632d819912 | fc62c142bb39 | fc625503dd60 | fc61e8c4ff87 | fc617c8621ae | fc61104743d5 | fc60a40865fc | fc6037c98823 | fc5fcb8aaa4a
  Z_out               48'hffff93c12227 f2f9b9ada0b2 | ffff93c12227 | 00006de5bbdb
  post_out            48'hfc6399c076eb fc6405ff54c4 | fc6399c076eb | fc632d819912 | fc62c142bb39 | fc625503dd60 | fc61e8c4ff87 | fc617c8621ae | fc61104743d5 | fc60a40865fc | fc6037c98823 | fc5fcb8aaa4a | fc6039706625
  P                   48'hfc6405ff54c4 f2f9b9ada0b2 | fc6405ff54c4 | fc6399c076eb | fc632d819912 | fc62c142bb39 | fc625503dd60 | fc61e8c4ff87 | fc617c8621ae | fc61104743d5 | fc60a40865fc | fc6037c98823 | fc5fcb8aaa4a
  D_A_B               48'h3adcacad1bf5 096a4c51b412 | 3adcacad1bf5 | 3adcacad1fa2 | 649e9b9f9466
```

OPMODE 6 = 8'b0001_11_10;

OPMODE[1:0] >> X input: accumulator P

OPMODE[3:2] >> Z input: C port

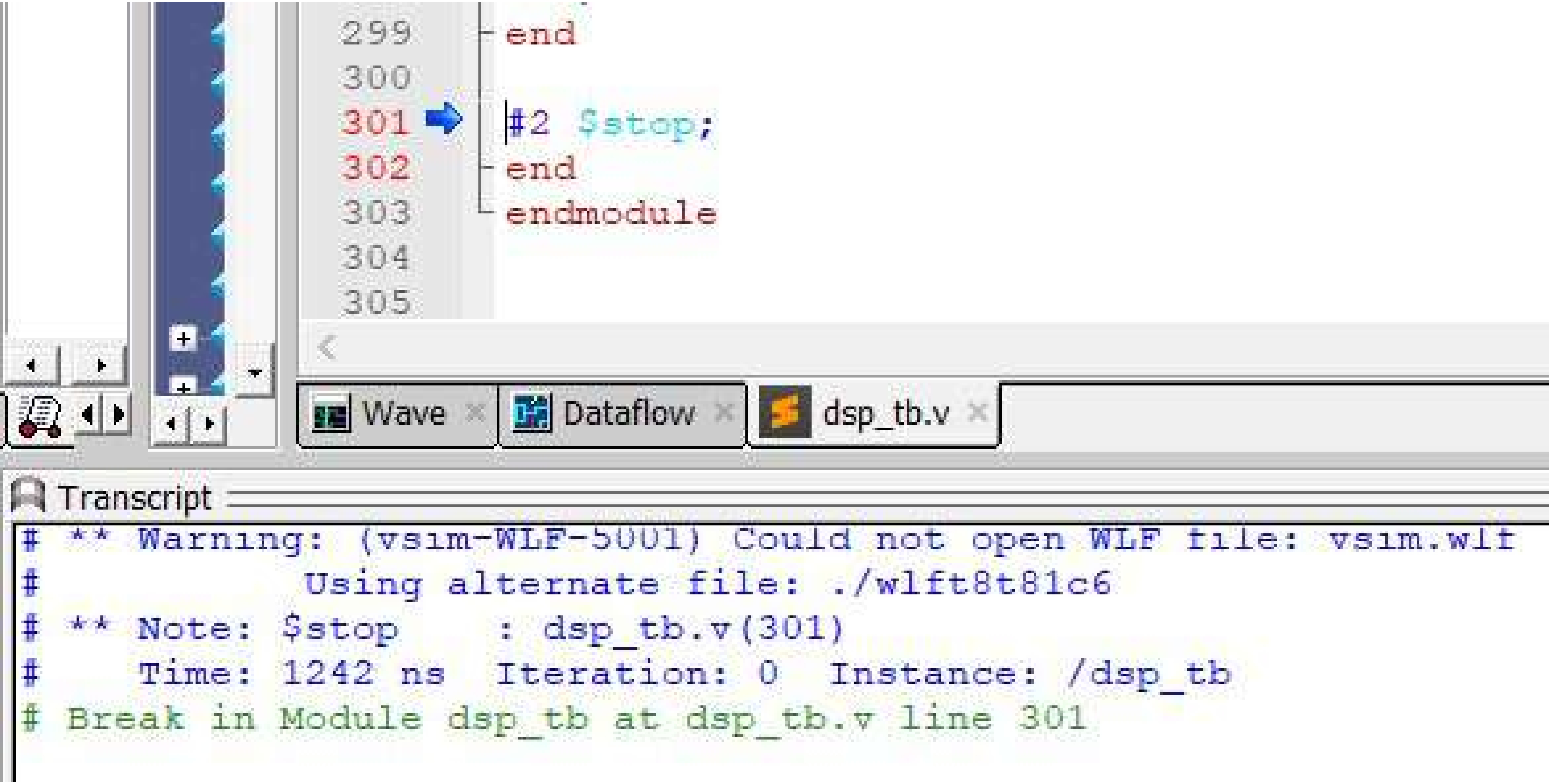
OPMODE[4]   >> use pre-adder or pre-subtracter

OPMODE[5]   >> force carry = 0

OPMODE[6]   >> specifies pre-adder

OPMODE[7]   >> specifies post-adder

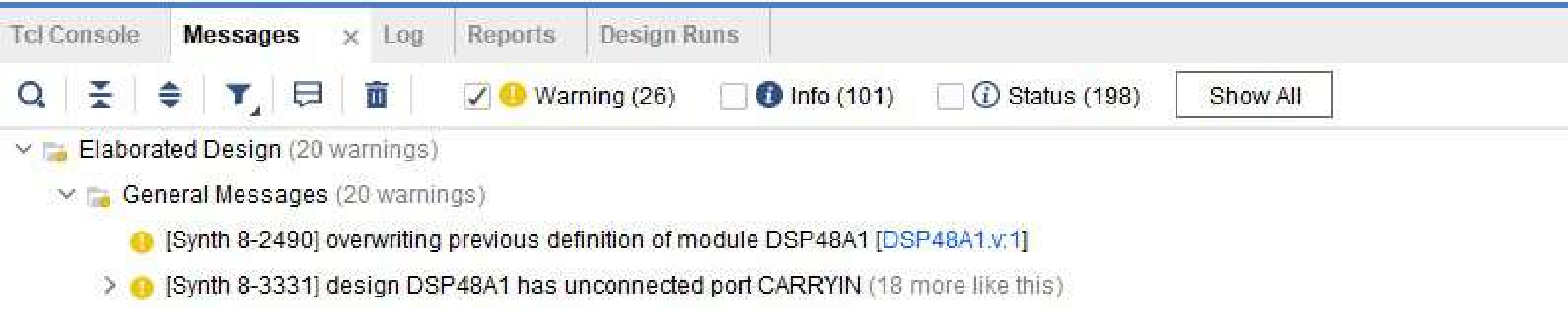
no error messages
(expected == actual)



```
299    end
300
301    #2 $stop;
302    end
303    endmodule
304
305
```

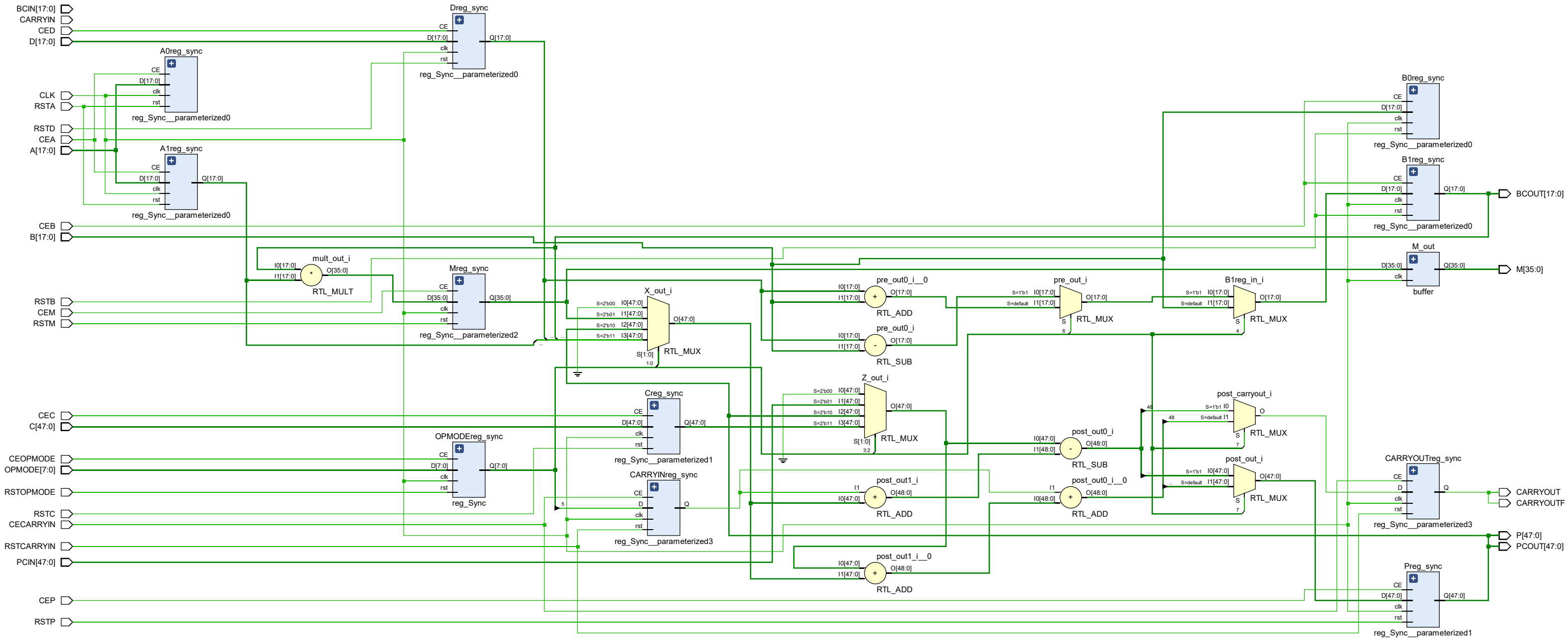
```
# ** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
#           Using alternate file: ./wlft8t81c6
# ** Note: $stop    : dsp_tb.v(301)
#    Time: 1242 ns  Iteration: 0  Instance: /dsp_tb
# Break in Module dsp_tb at dsp_tb.v line 301
```

Elaboration snippets

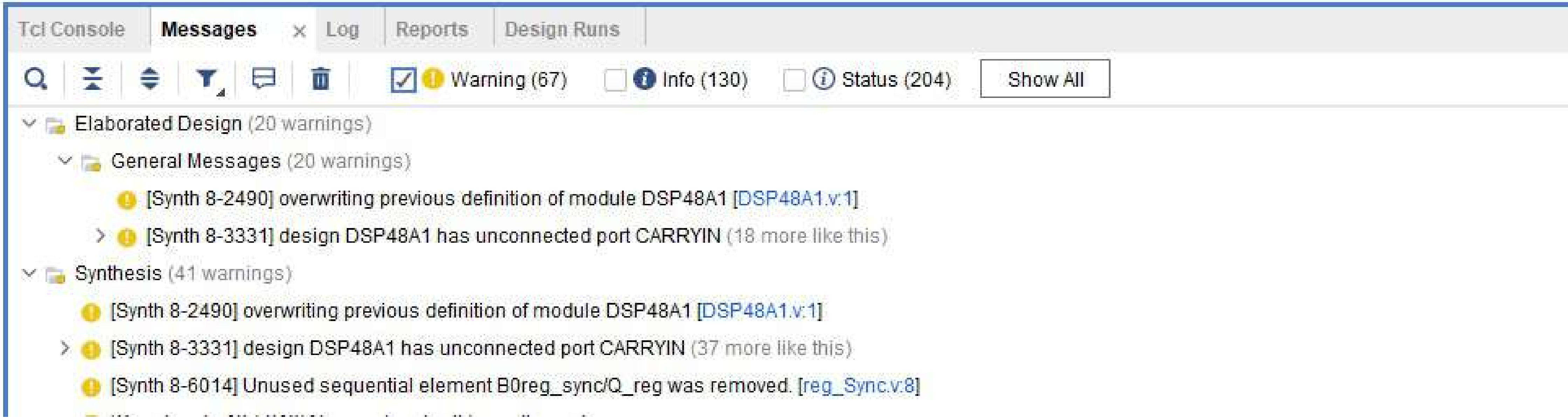


```
Elaborated Design (20 warnings)
  General Messages (20 warnings)
    [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
    [Synth 8-3331] design DSP48A1 has unconnected port CARRYIN (18 more like this)
```

Elaboration snippets

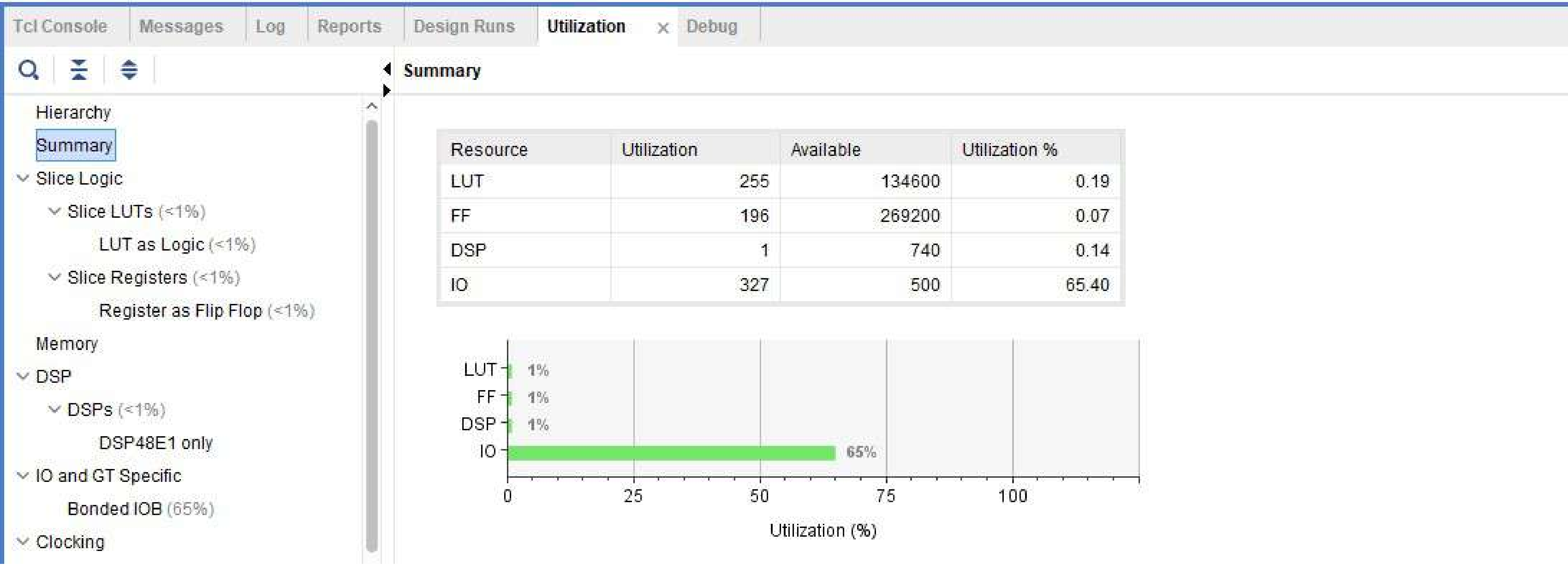


Synthesis snippets



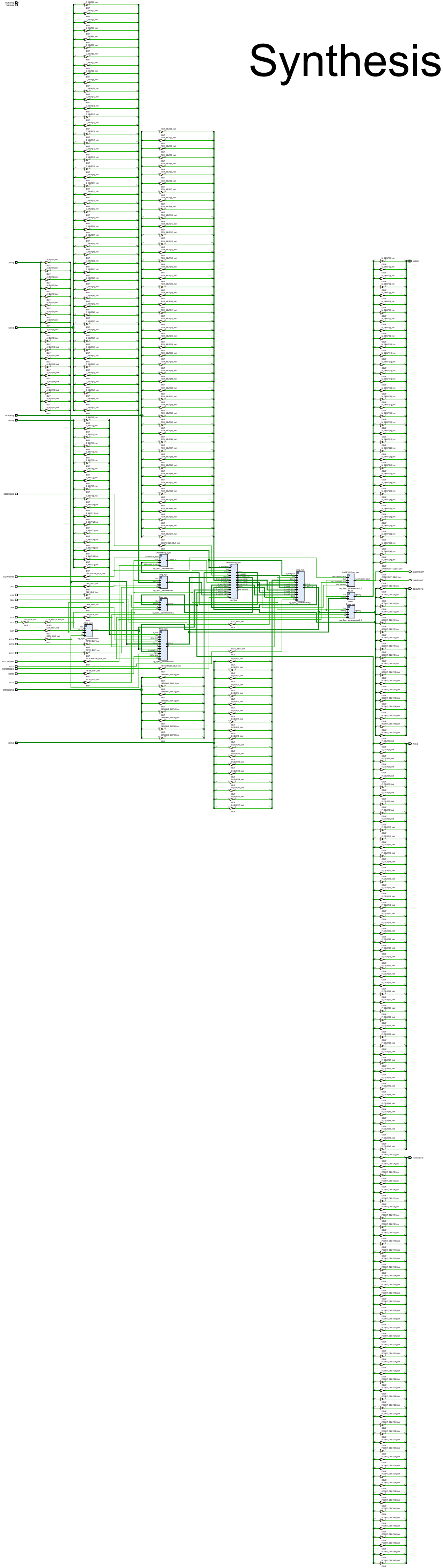
```
Elaborated Design (20 warnings)
    General Messages (20 warnings)
        [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
        [Synth 8-3331] design DSP48A1 has unconnected port CARRYIN (18 more like this)
Synthesis (41 warnings)
    [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
    [Synth 8-3331] design DSP48A1 has unconnected port CARRYIN (37 more like this)
    [Synth 8-6014] Unused sequential element B0reg_sync/Q_reg was removed. [reg_Sync.v:8]
```

Synthesis snippets

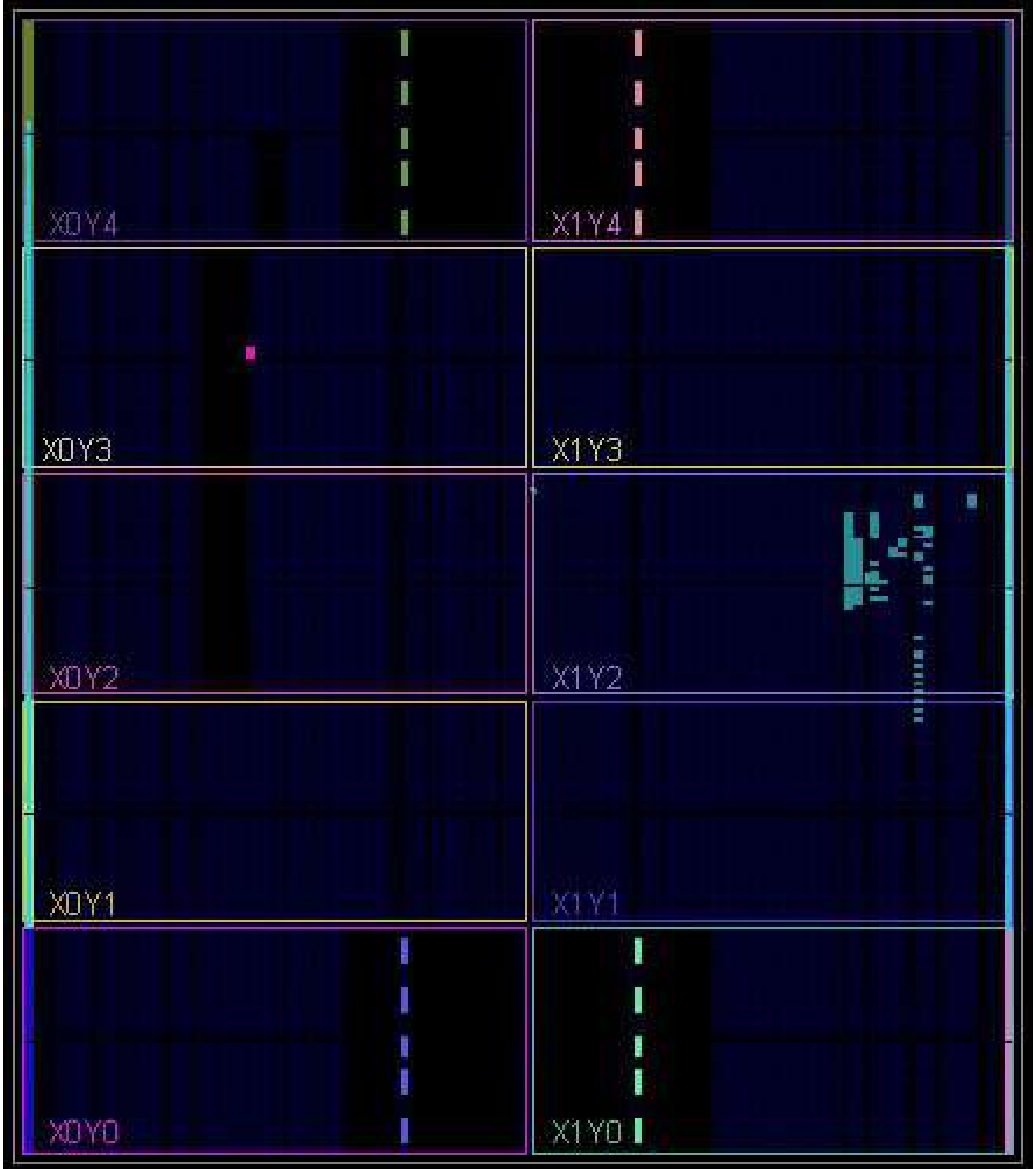


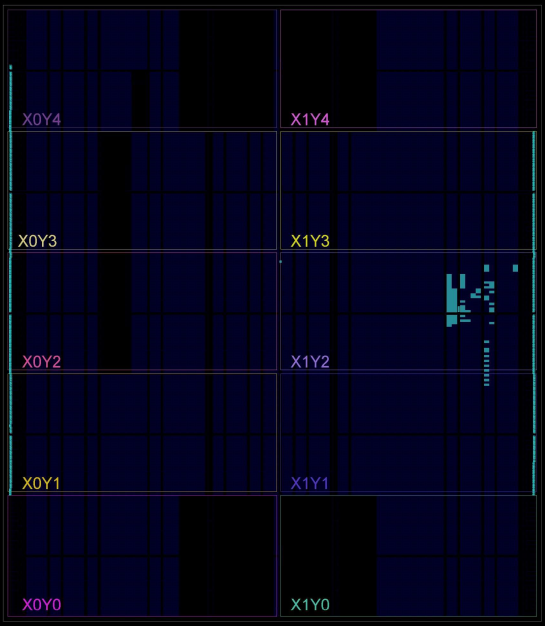
Resource	Utilization	Available	Utilization %
LUT	255	134600	0.19
FF	196	269200	0.07
DSP	1	740	0.14
IO	327	500	65.40

Synthesis snippets



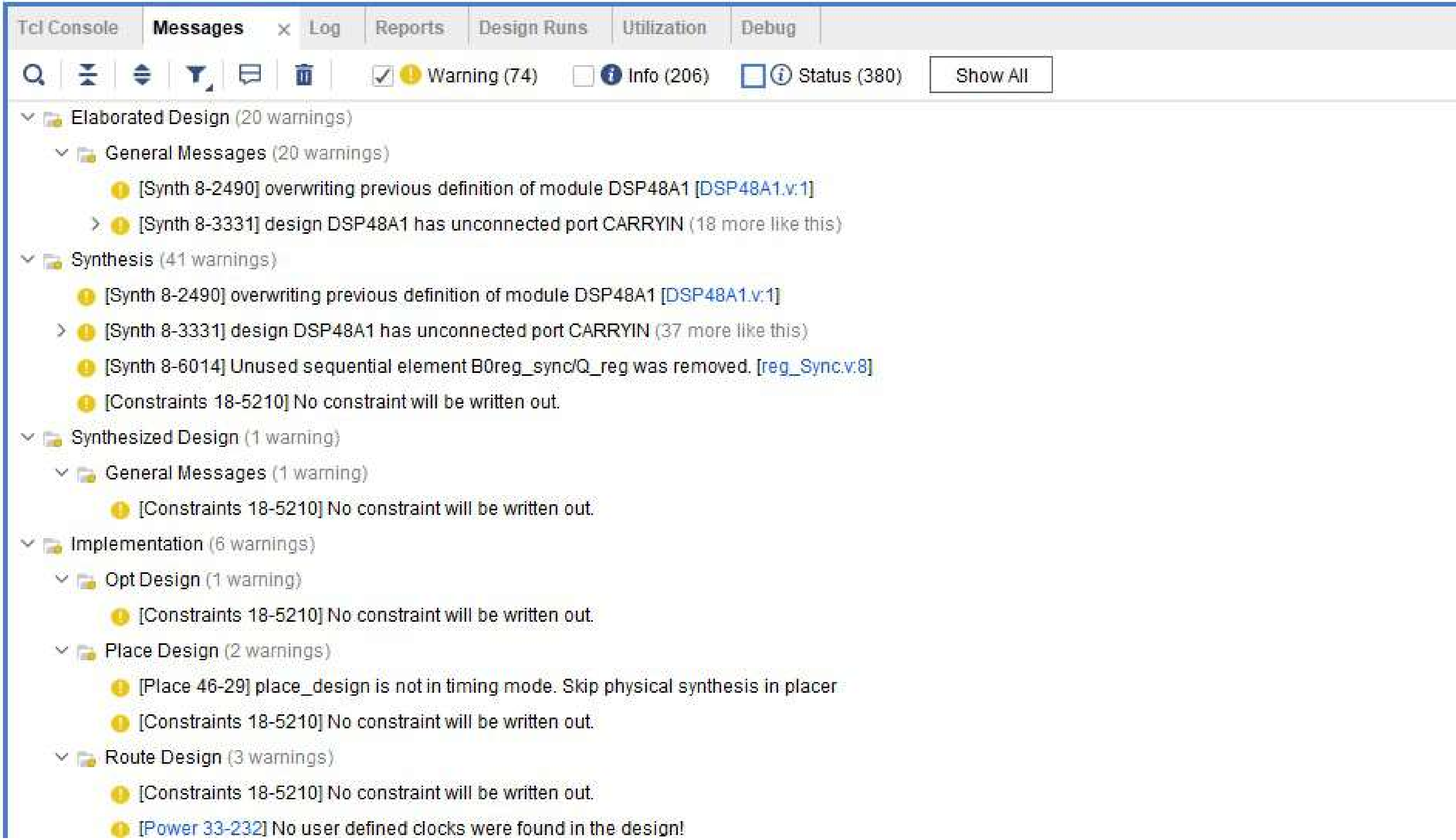
Implementation snippets





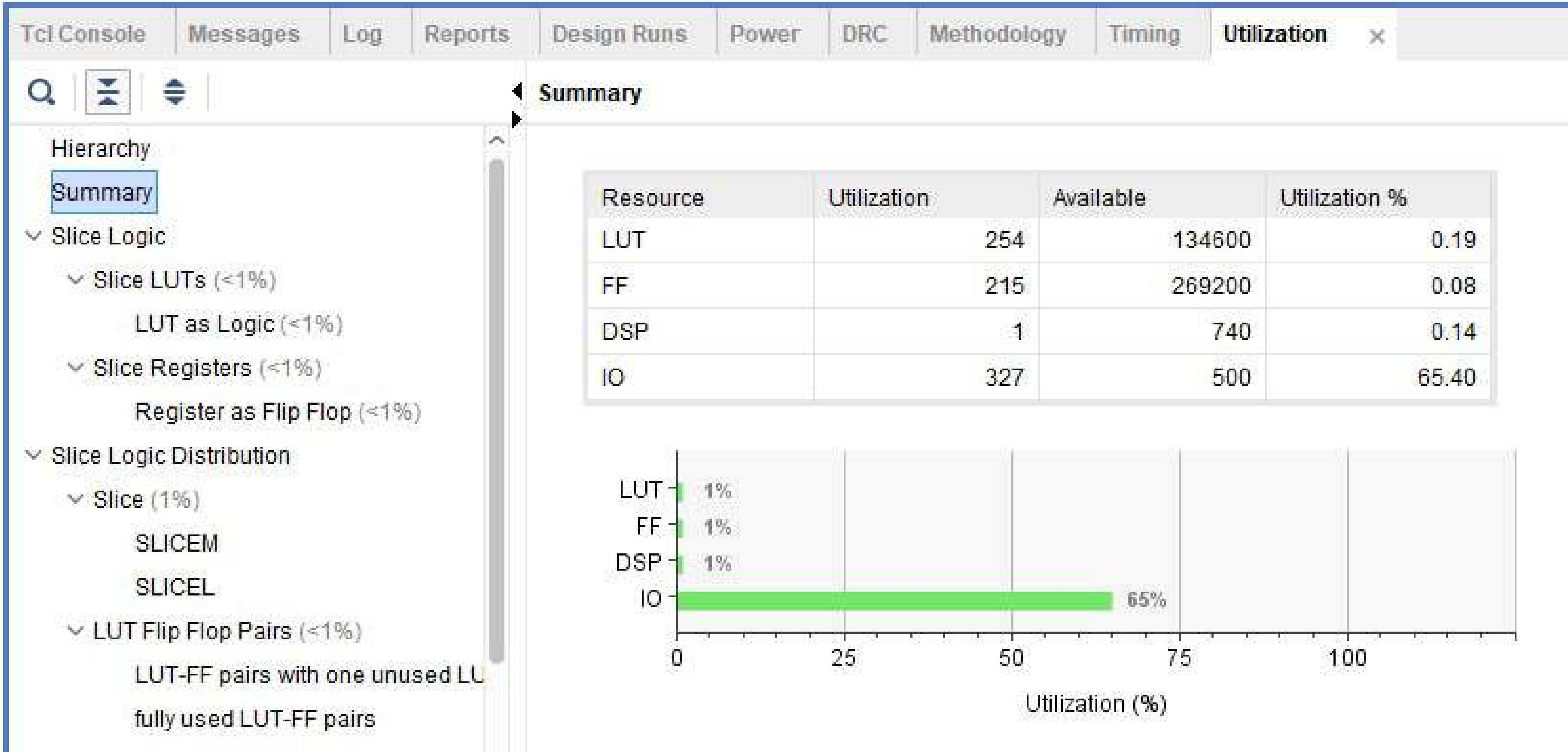
Implementation snippets

Implementation snippets



```
Elaborated Design (20 warnings)
  General Messages (20 warnings)
    [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
  > [Synth 8-3331] design DSP48A1 has unconnected port CARRYIN (18 more like this)
Synthesis (41 warnings)
  [Synth 8-2490] overwriting previous definition of module DSP48A1 [DSP48A1.v:1]
> [Synth 8-3331] design DSP48A1 has unconnected port CARRYIN (37 more like this)
  [Synth 8-6014] Unused sequential element B0reg_sync/Q_reg was removed. [reg_Sync.v:8]
  [Constraints 18-5210] No constraint will be written out.
Synthesized Design (1 warning)
  General Messages (1 warning)
    [Constraints 18-5210] No constraint will be written out.
Implementation (6 warnings)
  Opt Design (1 warning)
    [Constraints 18-5210] No constraint will be written out.
  Place Design (2 warnings)
    [Place 46-29] place_design is not in timing mode. Skip physical synthesis in placer
    [Constraints 18-5210] No constraint will be written out.
  Route Design (3 warnings)
    [Constraints 18-5210] No constraint will be written out.
    [Power 33-232] No user defined clocks were found in the design!
```

Implementation snippets



Resource	Utilization	Available	Utilization %
LUT	254	134600	0.19
FF	215	269200	0.08
DSP	1	740	0.14
IO	327	500	65.40