Date:18/5/2022

Fall Semester 2021-2022

ECE5014 – ASIC Design

M.Tech VLSI Design

School of Electronics Engineering

Vellore Institute of Technology

Name: Shreyas S Bagi

Register Number: 21MVD0086

Slot: L3+L4

Date:18/5/2022

Lab Task 04

Gate Level Simulation, Formal Verification and Pre-Layout STA

Section 1 Gate Level Simulation

Aim: The RTL code for Design of Simple Processor.

RTL Coding:

1. **Control path Verilog Code**

```
module controller_new (DIN, Run, Resetn, Clock, Done, R0out, R1out,
R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout, LdR0, LdR1, LdR2, LdR3, LdR4, LdR5, LdR6,
LdR7, LdA, LdG, Add_sub);
input Run, Resetn, Clock;
input [8:0] DIN;
output reg LdR0, LdR1, LdR2, LdR3, LdR4, LdR5, LdR6, LdR7, LdA, LdG,
Add_sub;
output reg R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
Done, DINout;
wire [9:0] Xreg, Yreg;
reg [9:0] Sel; //output DINout
reg [3:0] PS,NS;
reg [8:0] IR; // Instruction Register
// One - hot Encoding
localparam T0 = 4'b0001, T1 = 4'b0010, T2 = 4'b0100, T3 = 4'b1000; //One
hot Encoding
localparam mv = 3'b000, add = 3'b001, sub = 3'b010, mvi = 3'b011;
reg IRin;
// T0 is for loading state
// If mv instruction then go T1 and RYout and LdRx enabled Control signals
go high
// [8:0] IR
// [8,7,6] IR ------> Opcode
// [5,4,3] IR ------> RX
// [2,1,0] IR ------> RY

wire [2:0] opcode_decoded;
reg [2:0] opcode;


// add R1,R2
//001 001 010 ----> Din at Time t=0
// State is in T0
//opcode = Data_reg [8:6];
```

```
dec3to8 Z1 (IR[5:3],Xreg); // RX register -----> Xreg is Destination Register
dec3to8 Z2 (IR[2:0],Yreg); // RY register -----> Yreg is Source Register
dec3to8_3bit Z3 (IR[8:6], opcode_decoded); // Opcode decoding


// Current State Logic Generation
always @(posedge Clock)
begin
if(Resetn)
PS <= NS;
else
PS <= T0;
end


always @(posedge Clock)
begin
if(IRin)
opcode <= opcode_decoded;
else
opcode <= opcode;
end

// Next State Generation Logic
always @(PS,Run,Done) // removed Run or Done signal in senstivity list.
begin
case(PS) // Use only Blocking statements.
T0: NS = Run? T1:T0;

T1: begin   // Destination Register Load Signal.
        case(opcode)
                mv   : NS =T0; // Change go back to stage T0 not to be in T1.
                add  : NS =T2;
                sub  : NS =T2;
                mvi  : NS = T2;
                default :NS = T0;
         endcase
     end

T2:begin  // Source Register Load Signal.
        case(opcode)
                add  :  NS=T3;
                sub  :  NS=T3;
                mvi  :  NS = T0;
                default : NS = T0;
```

```
            endcase
        end

  T3:begin  // Source Register Load Signal.
            case(opcode)
                    add  : NS=T0;
                    sub  :  NS=T0;
                    default : NS = T0;
            endcase
        end
default : NS = T0;
endcase
end


// Output Generation always block that is Control Signal Generation Logic
always @(PS)
begin
case(PS) // USe only Blocking statements.
T0: begin
        // Initial Reset all the Load Signals
        LdR0 = 0; LdR1 = 0; LdR2 = 0; LdR3 = 0; LdR4 =0;  LdR5 = 0;  LdR6
= 0;  LdR7 = 0;LdA = 0; LdG =0;
        IRin = 1;
        if(!Run)
                begin
                  Done = 0;
               end
            else
                    begin
                    IR   = DIN;
                    Done = 0;
                    end
       end
T1: begin   // Destination Register Load Signal.
            IRin = 0;
       if (opcode == mv)
       begin
       // Ryout corresponding Reg should be high
       // Rx in corresponding Reg should be high
       // Done 1
       // mv R1,R2   R1<----- R2
       Sel = Yreg;
      {R0out,  R1out,  R2out,  R3out,  R4out,  R5out,  R6out,  R7out,  Gout,
DINout} = Sel;
             {LdR0,LdR1,LdR2,LdR3,LdR4,LdR5,LdR6,LdR7} = Xreg[9:2];
            // changed to reg from sel
        Done = 1;
       end

      else if(opcode == add)
       begin
       // Ryout corresponding Reg should be high
       // Rx in corresponding Reg should be high
       // Done 1
       // add R1,R2   R1<----R1+R2
       Sel = Xreg;
       {R0out,  R1out,  R2out,  R3out,  R4out,  R5out,  R6out,  R7out,  Gout,
DINout} = Sel;
       LdA = 1;
       LdG = 0;
      // A = BusWires;
       Add_sub = 1; // 1 for add
       end

      else if(opcode == sub)
      begin
       Sel = Xreg;
       {R0out,  R1out,  R2out,  R3out,  R4out,  R5out,  R6out,  R7out,  Gout,
DINout} = Sel;
       LdA = 1;
       LdG = 0;
       //A <= BusWires;
       Add_sub = 0; // 1 for sub
      end

      else if(opcode == mvi)
      begin
      Sel = 10'b0000000001; // Sel = Din out
     {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout, DINout}
= Sel;
            {LdR0,LdR1,LdR2,LdR3,LdR4,LdR5,LdR6,LdR7} = Xreg[9:2];

     end
     end

T2:begin  // Source Register Load Signal.
            //IRin = 0;
              if(opcode == add)
       begin
       Sel = Yreg;
       {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout} = Sel;
       LdA = 0;
       LdG = 1;
       //G = Z; // Z is result of addition from the adder block;
       end

       else if(opcode == mvi)
       begin
       Sel = 10'b0000000001; // Sel = Din out
       {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout} = Sel;
              {LdR0,LdR1,LdR2,LdR3,LdR4,LdR5,LdR6,LdR7} = Xreg[9:2];
       Done = 1;
       end

      else if(opcode == sub)
      begin
       Sel = Yreg;
       {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout} = Sel;
       LdA = 0;
       LdG = 1;
      end
              end

T3: begin
            //IRin = 0;
       if(opcode == add)
       begin
       Done = 1;
       Sel=10'b0000000010; // Data from G Register to Destination Register.
Gout=1
       {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout} = Sel;
              {LdR0,LdR1,LdR2,LdR3,LdR4,LdR5,LdR6,LdR7} = Xreg[9:2];

       end

      else if(opcode == sub)
      begin
      Done = 1;
      Sel=10'b0000000010; // Data from G Register to Destination Register.
      {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout} = Sel;
              {LdR0,LdR1,LdR2,LdR3,LdR4,LdR5,LdR6,LdR7} = Xreg[9:2];
```

```
    end
    end
default : begin //IRin = 0;
            end
endcase
end

endmodule
module dec3to8 (W, Y);
input [2:0] W;
output reg [9:0] Y;

always @(W)
begin
case (W)
3'b000: Y = 10'b1000000000;
3'b001: Y = 10'b0100000000;
3'b010: Y = 10'b0010000000;
3'b011: Y = 10'b0001000000;
3'b100: Y = 10'b0000100000;
3'b101: Y = 10'b0000010000;
3'b110: Y = 10'b0000001000;
3'b111: Y = 10'b0000000100;
//3'b1000: Y = 10'b0000000010;
endcase
end
endmodule


module dec3to8_3bit (W, Y);
input [2:0] W;
output reg [2:0] Y;

always @(W)
begin
case (W)
3'b000: Y = 3'b000;
3'b001: Y = 3'b001;
3'b010: Y = 3'b010;
3'b011: Y = 3'b011;
3'b100: Y = 3'b100;
3'b101: Y = 3'b101;
3'b110: Y = 3'b110;
3'b111: Y = 3'b111;
default: Y = 3'b000;
```

```
endcase
end
endmodule
```

2. **Datapath Verilog Code**

```
/*

File_Name : Data_path_work

revision: 2

Last_ Modified : 12/03/2021

*/

//`timescale 1ns/1ns

module datapath_register_array (R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout, DINout,

Clock,rst,R0in, R1in,R2in,R3in,R4in,R5in,R6in,R7in,Ain,Bus,DIN,AddSub,Gin);


input R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout, DINout, Clock,R0in, R1in,R2in,R3in,R4in,R5in,R6in,R7in,Ain,rst,AddSub,Gin;

input [8:0] DIN;

output [8:0] Bus;

wire [8:0] R0_data_out, R1_data_out, R2_data_out, R3_data_out, R4_data_out;

wire [8:0] R5_data_out, R6_data_out, R7_data_out, A_data_out, Sum,G;


mux_10to1 m1 (.R0out(R0out), .R1out(R1out), .R2out(R2out), .R3out(R3out), .R4out(R4out), .R5out(R5out), .R6out(R6out), .R7out(R7out), .Gout(Gout), .DINout(DINout),.DIN(DIN), .R0(R0_data_out),.R1(R1_data_out), .R2(R2_data_out), .R3(R3_data_out), .R4(R4_data_out), .R5(R5_data_out),.R6(R6_data_out), .R7(R7_data_out), .G(G), .Bus(Bus)); //Priority Multiplexer


Add_Sub add_top (.A(A_data_out), .Bus(Bus), .AddSub(AddSub),.ALU_out(Sum)); //Add_Sub (Aout, Bus, AddSub,ALU_out);

reg_G g2 (.Sum(Sum),.Gin(Gin), .Clock(Clock), .rst(rst), .Z(G));


Register Reg0 (Clock, R0_data_out, Bus, R0in,rst);

Register Reg1 (Clock, R1_data_out, Bus, R1in,rst);
Register Reg2 (Clock, R2_data_out, Bus, R2in,rst);
Register Reg3 (Clock, R3_data_out, Bus, R3in,rst);
Register Reg4 (Clock, R4_data_out, Bus, R4in,rst);
Register Reg5 (Clock, R5_data_out, Bus, R5in,rst);
Register Reg6 (Clock, R6_data_out, Bus, R6in,rst);
Register Reg7 (Clock, R7_data_out, Bus, R7in,rst);
Register A5   (Clock, A_data_out,   Bus, Ain,rst);
endmodule



module mux_10to1 (R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout,
DIN, R0,R1, R2, R3, R4, R5,R6, R7, G, Bus);

input R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout, DINout;
input [8:0] DIN, R0,R1, R2, R3, R4, R5,R6, R7, G;
output reg [8:0] Bus;
wire [9:0] sel; //selection line of mux_10to1

assign sel = {R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout};
always @ (*)
 begin
   if (sel == 10'b0000000001)
         Bus = DIN;
    else if (sel == 10'b0000000010)
         Bus = G;
    else if (sel == 10'b0000000100)
         Bus = R7;
            else if (sel == 10'b0000001000)
         Bus = R6;
```

```
            else if (sel == 10'b0000010000)
          Bus = R5;
            else if (sel == 10'b0000100000)
          Bus = R4;
            else if (sel == 10'b0001000000)
          Bus = R3;
            else if (sel == 10'b0010000000)
          Bus = R2;
            else if (sel == 10'b0100000000)
          Bus = R1;
            else if (sel == 10'b1000000000)
          Bus = R0;
   else
          Bus = Bus;
 end
endmodule

//For register
module Register (Clock,dout,din,EN,rst);
input Clock;
input EN,rst;
input [8:0] din;
output reg [8:0] dout;

always @(posedge Clock)
begin
if(EN)
            dout <= din;
else if(~rst)
            dout <= 9'b0;
```

```
end
endmodule

//ADD_SUB
module Add_Sub (A, Bus, AddSub,ALU_out);
input [8:0] A, Bus;
input AddSub;
output [8:0] ALU_out;
wire [8:0] xrout;
wire Cout;

xor       (xrout[0],Bus[0],AddSub),       (xrout[1],Bus[1],AddSub),
(xrout[2],Bus[2],AddSub), (xrout[3],Bus[3],AddSub),
    (xrout[4],Bus[4],AddSub),             (xrout[5],Bus[5],AddSub),
(xrout[6],Bus[6],AddSub), (xrout[7],Bus[7],AddSub),
    (xrout[8],Bus[8],AddSub);
csa_9bit add1 (.a(A), .b(xrout),.cin(AddSub) , .sum(ALU_out), .cout(Cout));

endmodule

module reg_G(Sum,Gin, Clock,rst, Z);
input Gin;
input [8:0] Sum;
input Clock, rst;
output reg [8:0] Z;
always @ (posedge Clock)
begin
if (~rst)
begin
Z <= 9'b0;
end
```

```
else
begin
if(Gin == 1)
Z <= Sum;
else
Z <= Z;
end
end
endmodule


module csa_9bit(a, b,cin, sum, cout);
//In main module a= A, b= Bus , c= AddSub
//module carry_select_adder_16bit(a, b, cin, sum, cout);
input [8:0] a,b;
input cin;
output [8:0] sum;
output cout;

wire [1:0] c;
// assign {cout,sum} = a+b+cin;
//full adder
full_adder fa0(.a(a[0]),.b(b[0]),.cin(cin),.sum(sum[0]),.cout(c[0]));
carry_select_adder_4bit_slice
csa_slice1(.a(a[4:1]),.b(b[4:1]),.cin(c[0]),.sum(sum[4:1]),.cout(c[1]));
carry_select_adder_4bit_slice
csa_slice2(.a(a[8:5]),.b(b[8:5]),.cin(c[1]),.sum(sum[8:5]), .cout(cout));

endmodule
```

```
////////////////////////////////////
//4-bit Carry Select Adder Slice
////////////////////////////////////

module carry_select_adder_4bit_slice(a, b, cin, sum, cout);
input [3:0] a,b;
input cin;
output [3:0] sum;
output cout;

wire [3:0] s0,s1;
wire c0,c1;

ripple_carry_4_bit rca1(.a(a[3:0]),.b(b[3:0]),.cin(1'b0),.sum(s0[3:0]),.cout(c0));
ripple_carry_4_bit rca2(.a(a[3:0]),.b(b[3:0]),.cin(1'b1),.sum(s1[3:0]),.cout(c1));

mux2X1 ms0(.in0(s0[3:0]),.in1(s1[3:0]),.sel(cin),.out(sum[3:0]));
mux2X1_1 mc0(.in0(c0),.in1(c1),.sel(cin),.out(cout));
endmodule

////////////////////
//2X1 Mux
////////////////////

module mux2X1( in0,in1,sel,out);
//parameter width=16;
input [3:0] in0,in1;
input sel;
output [3:0] out;
assign out=(sel)?in1:in0;
```

```
endmodule

 module mux2X1_1( in0,in1,sel,out);
//parameter width=16;
input  in0,in1;
input sel;
output  out;
assign out=(sel)?in1:in0;
endmodule
///////////////////////////////
//4-bit Ripple Carry Adder
///////////////////////////////
module ripple_carry_4_bit(a, b, cin, sum, cout);
input [3:0] a,b;
input cin;
output [3:0] sum;
output cout;
 wire c1,c2,c3;
 full_adder fa0(.a(a[0]),.b(b[0]),.cin(cin),.sum(sum[0]),.cout(c1));
full_adder fa1(.a(a[1]),.b(b[1]),.cin(c1),.sum(sum[1]),.cout(c2));
full_adder fa2(.a(a[2]),.b(b[2]),.cin(c2),.sum(sum[2]),.cout(c3));
full_adder fa3(.a(a[3]),.b(b[3]),.cin(c3),.sum(sum[3]),.cout(cout));
endmodule

////////////////////
//1bit Full Adder
////////////////////

module full_adder(a,b,cin,sum,cout);
input a,b,cin;
```

```
output sum, cout;
wire x,y,z;
half_adder h1(.a(a), .b(b), .sum(x), .cout(y));
half_adder h2(.a(x), .b(cin), .sum(sum), .cout(z));
or or_1(cout,z,y);
endmodule

/////////////////////
// 1 bit Half Adder
/////////////////////`timescale 1ns / 1ns
module half_adder( a,b, sum, cout );
input a,b;
output sum, cout;
xor xor_1 (sum,a,b);
and and_1 (cout,a,b);
endmodule
```

3. Top Module Code

```
module simple_processor_Top (Run, Resetn, Clock, DIN, Bus, Done);
input Run, Resetn, Clock;
input [8:0] DIN;
output [8:0] Bus;
output Done;

wire R0out, R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout,
DINout, LdR0, LdR1, LdR2, LdR3, LdR4, LdR5, LdR6, LdR7, LdA, LdG,
Add_sub;

controller_new G1_Controller (DIN, Run, Resetn, Clock, Done, R0out,
R1out, R2out, R3out, R4out, R5out, R6out, R7out, Gout, DINout, LdR0,
LdR1, LdR2, LdR3, LdR4, LdR5, LdR6, LdR7, LdA, LdG, Add_sub);

datapath_register_array G2_Datapath (R0out, R1out, R2out, R3out, R4out,
R5out, R6out, R7out, Gout, DINout, Clock, Resetn, LdR0, LdR1, LdR2,
LdR3, LdR4, LdR5, LdR6, LdR7, LdA, Bus, DIN, Add_sub, LdG);

endmodule
```

Script File for run

```
vcs -V -R -f Simple_Processor_filelist -full64 -debug_all -o gls -gui
```

File list

```
saed14nm_hvt.v
saed14nm_rvt.v
saed14nm_lvt.v
Simple_Processor_netlist.v
Simple_processor_testbench_22_april_mv_work.v
```

Date:18/5/2022

Testbench Codes based on Operations

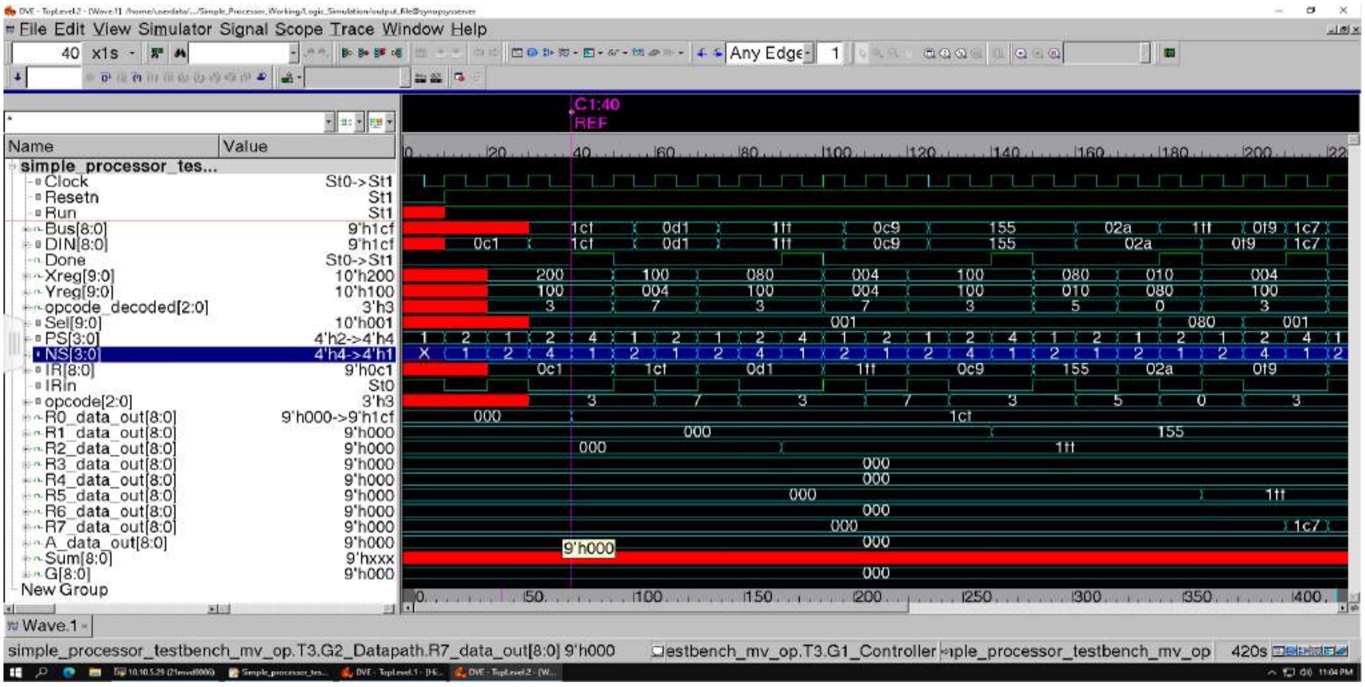
1. **Move Immediate Operation**

```
`timescale 1ns/1ns
module simple_processor_testbench ();
reg Run, Resetn, Clock;
reg [8:0] DIN;
wire [8:0] Bus;
wire Done;

controller_new G1 (DIN, Run, Resetn, Clock, Done, R0out, R1out,
R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout, LdR0, LdR1, LdR2, LdR3, LdR4, LdR5,
LdR6, LdR7, LdA, LdG, Add_sub);

datapath_register_array G2 (R0out, R1out, R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout, Clock, Resetn, LdR0,
LdR1, LdR2, LdR3, LdR4, LdR5, LdR6, LdR7, LdA, Bus, DIN,
Add_sub, LdG);

initial
Clock = 1'b1;

always #5 Clock = ~Clock;
initial
begin

$set_toggle_region(simple_processor_testbench.G1,
simple_processor_testbench.G2);
$toggle_start();
// ...

Resetn = 1'b0;
#10 Resetn = 1'b1;
Run = 1'b1;

DIN = 9'b011000001; // mvi operation
#20 DIN = 9'b111001111; // immediate data
//R0 is loaded with 111001111;
#5;
#20 DIN = 9'b011010001; // mvi operation
#20 DIN = 9'b111111111; // immediate data after 50ns from 1st instuction
@t=10
// R2 is loaded with 111111111;
#30 DIN = 9'b011001001; // mvi operation
#20 DIN = 9'b101010101; // immediate data
//R1 is loaded with 101010101;
```

```
#5;
#30 DIN = 9'b011011001; // mvi operation
#20 DIN = 9'b101010111; // immediate data
//R3 is loaded with 101010101;
#5;
#30 DIN = 9'b011100001; // mvi operation
#20 DIN = 9'b111101111; // immediate data
//R4 is loaded with 101010101;
#5;
#30 DIN = 9'b011101001; // mvi operation
#20 DIN = 9'b110110110; // immediate data
//R5 is loaded with 101010101;
#5;
#30 DIN = 9'b011110001; // mvi operation
#20 DIN = 9'b111011011; // immediate data
//R6 is loaded with 101010101;
#5;
#30 DIN = 9'b011111001; // mvi operation
#20 DIN = 9'b111111111; // immediate data
//R7 is loaded with 101010101;
// ...
$toggle_stop();
$toggle_report("Simple_Processor_SAIF_Move_Immediate_Operation.
saif", 1.0e-12, "simple_processor_testbench");
#5; $stop;
Run = 1'b0;
end
end
endmodule
```

Date:18/5/2022

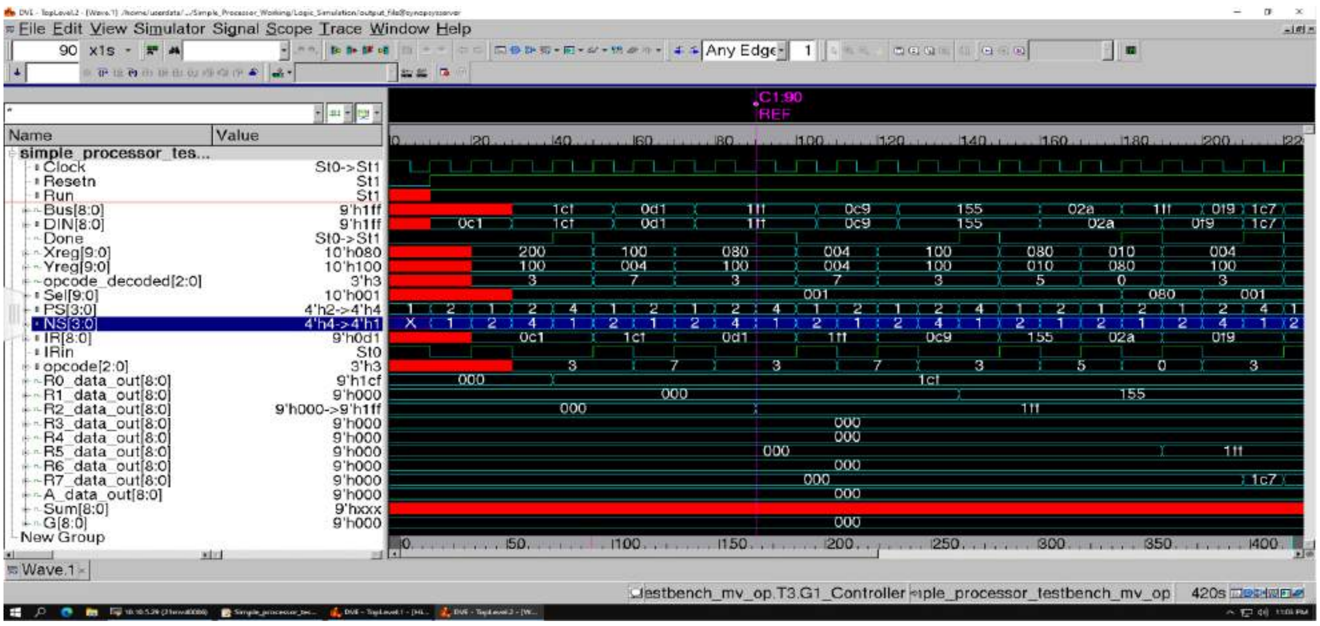
Waveforms:



```
Name                      Value
simple_processor_tes...
  Clock                   St0->St1
  Resetn                  St1
  Run                     St1
  Bus[8:0]                9'h1cf
  DIN[8:0]                9'h1cf
  Done                    St0->St1
  Xreg[9:0]               10'h200
  Yreg[9:0]               10'h100
  opcode_decoded[2:0]     3'h3
  Sel[9:0]                10'h001
  PS[3:0]                 4'h2->4'h4
  NS[3:0]                 4'h4->4'h1
  IR[8:0]                 9'h0c1
  IRin                    St0
  opcode[2:0]             3'h3
  R0_data_out[8:0]        9'h000->9'h1cf
  R1_data_out[8:0]        9'h000
  R2_data_out[8:0]        9'h000
  R3_data_out[8:0]        9'h000
  R4_data_out[8:0]        9'h000
  R5_data_out[8:0]        9'h000
  R6_data_out[8:0]        9'h000
  R7_data_out[8:0]        9'h000
  A_data_out[8:0]         9'h000
  Sum[8:0]                9'hxxx
  G[8:0]                  9'h000
New Group

simple_processor_testbench_mv_op.T3.G2_Datapath.R7_data_out[8:0] 9'h000
```

Figure 1.1: The data stored in Register and Move Immediate Operation.

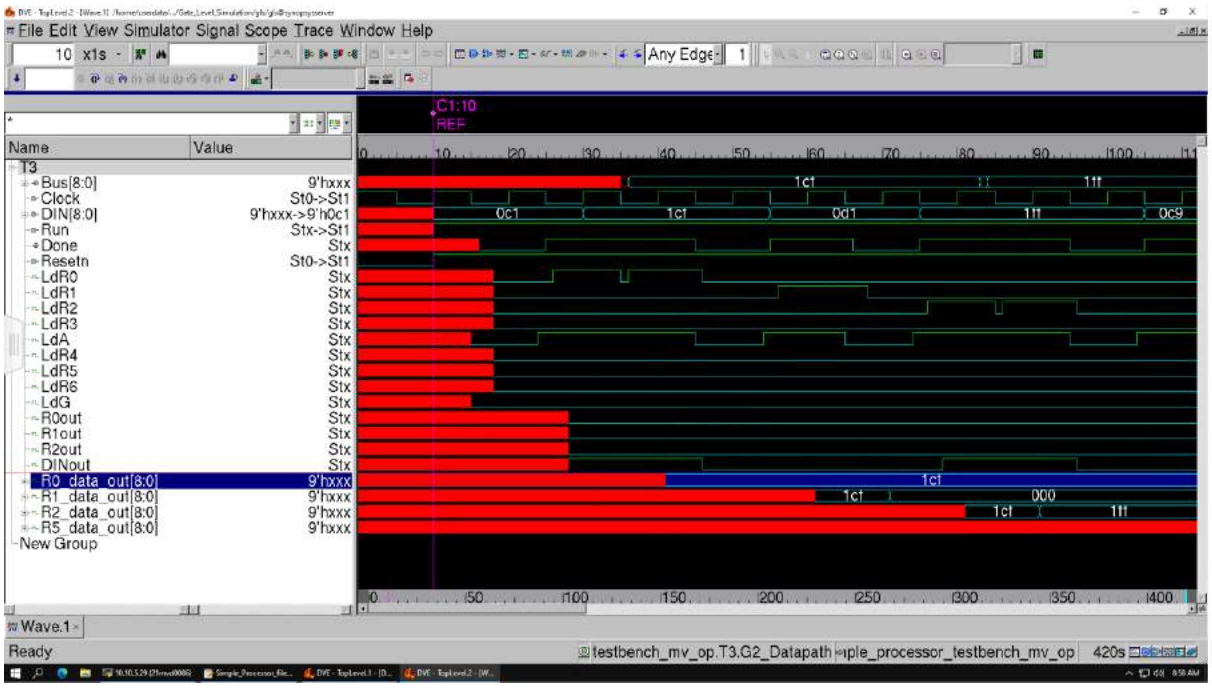
In the Figure 1.1 we are doing Move immediate operation. At time instant from t=0ns to 40ns we observe that Register R0 is stored with 0 value and after 40ns we store the data 9'h1CF data. The flow of data is like this at 10ns the (2nd Clock pulse), Resetn = 1 and Run = 1. These conditions are met then only start the operation of processor. The Move immediate operation has three states T0 to T1 to T2. In these states the operation occurs. At time t=40ns we observe R0 changes data. Similarly In Figure 1.2 we observe the move immediate operation taking place for R2 register and R1 register.



```
Name                      Value
simple_processor_tes...
  Clock                   St0->St1
  Resetn                  St1
  Run                     St1
  Bus[8:0]                9'h1ff
  DIN[8:0]                9'h1ff
  Done                    St0->St1
  Xreg[9:0]               10'h080
  Yreg[9:0]               10'h100
  opcode_decoded[2:0]     3'h3
  Sel[9:0]                10'h001
  PS[3:0]                 4'h2->4'h4
  NS[3:0]                 4'h4->4'h1
  IR[8:0]                 9'h0d1
  IRin                    St0
  opcode[2:0]             3'h3
  R0_data_out[8:0]        9'h1cf
  R1_data_out[8:0]        9'h000
  R2_data_out[8:0]        9'h000->9'h1ff
  R3_data_out[8:0]        9'h000
  R4_data_out[8:0]        9'h000
  R5_data_out[8:0]        9'h000
  R6_data_out[8:0]        9'h000
  R7_data_out[8:0]        9'h000
  A_data_out[8:0]         9'h000
  Sum[8:0]                9'hxxx
  G[8:0]                  9'h000
New Group
```

Figure 1.2: The data stored in Register and Move Immediate Operation.

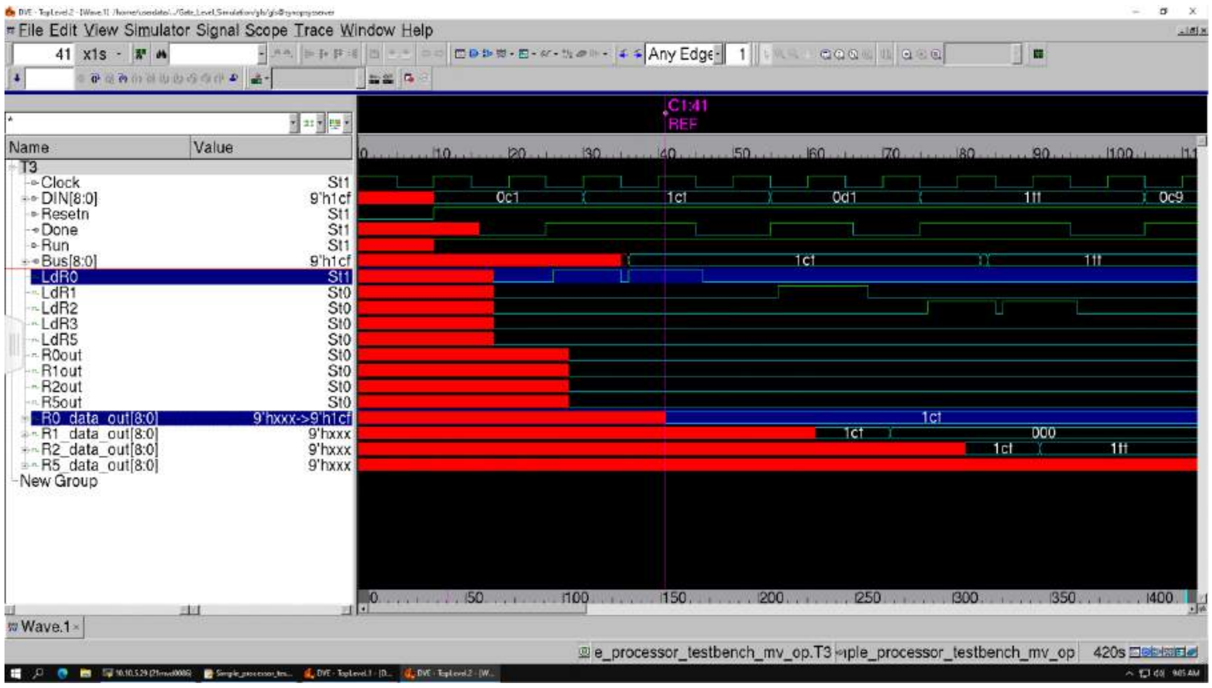
Gate Level Simulation Waveforms:



Name	Value
T3	
Bus[8:0]	9'hxxx
Clock	St0->St1
DIN[8:0]	9'hxxx->9'h0c1
Run	Stx->St1
Done	Stx
Resetn	St0->St1
LdR0	Stx
LdR1	Stx
LdR2	Stx
LdR3	Stx
LdA	Stx
LdR4	Stx
LdR5	Stx
LdR6	Stx
LdG	Stx
R0out	Stx
R1out	Stx
R2out	Stx
DINout	Stx
R0_data_out[8:0]	9'hxxx
R1_data_out[8:0]	9'hxxx
R2_data_out[8:0]	9'hxxx
R5_data_out[8:0]	9'hxxx
New Group	

Ready testbench_mv_op.T3.G2_Datapath ple_processor_testbench_mv_op 420s

Figure 1.3: The data stored in Register and Move Immediate Operation.



Name	Value
T3	
Clock	St1
DIN[8:0]	9'h1cf
Resetn	St1
Done	St1
Run	St1
Bus[8:0]	9'h1cf
LdR0	St1
LdR1	St0
LdR2	St0
LdR3	St0
LdR5	St0
R0out	St0
R1out	St0
R2out	St0
R5out	St0
R0_data_out[8:0]	9'hxxx->9'h1cf
R1_data_out[8:0]	9'hxxx
R2_data_out[8:0]	9'hxxx
R5_data_out[8:0]	9'hxxx
New Group	

e_processor_testbench_mv_op.T3 ple_processor_testbench_mv_op 420s

Figure 1.4: The data stored using Move Immediate Operation for Register R0.

In the Figure 1.3 we are doing Move immediate operation. At time instant from t=0ns to 40ns we observe that Register R0 is stored with 0 value and after 40ns we store the data 9’h1CF data observe in Figure 1.4 with 1ns delay. Similarly In Figure 1.5 and Figure 1.6 we observe the move immediate operation taking place for R2 register and R1 register.

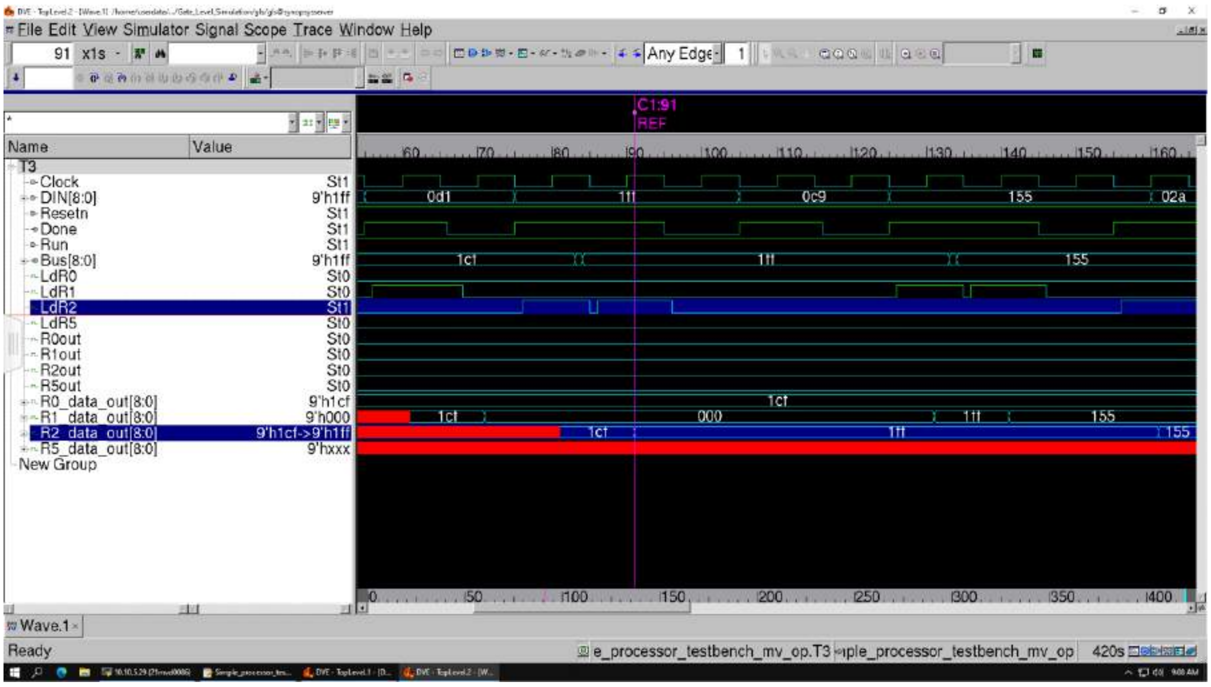


Figure 1.5: The data stored using Move Immediate Operation for Register R2.

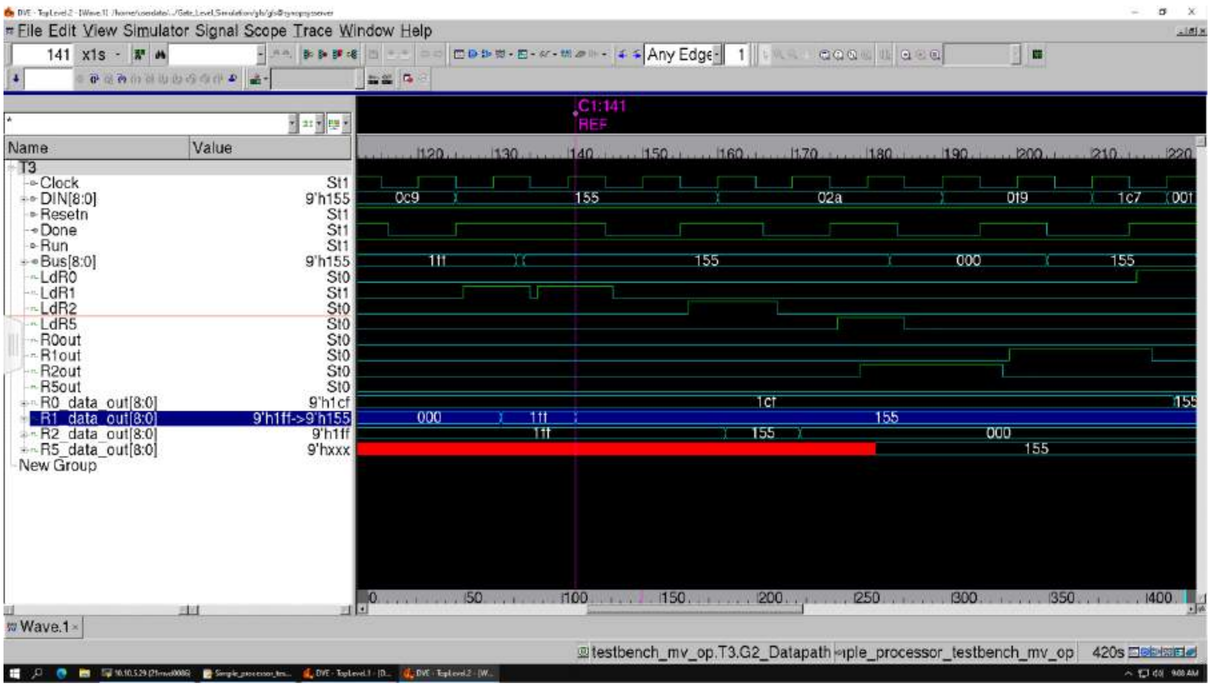


Figure 1.6: The data stored using Move Immediate Operation for Register R1.

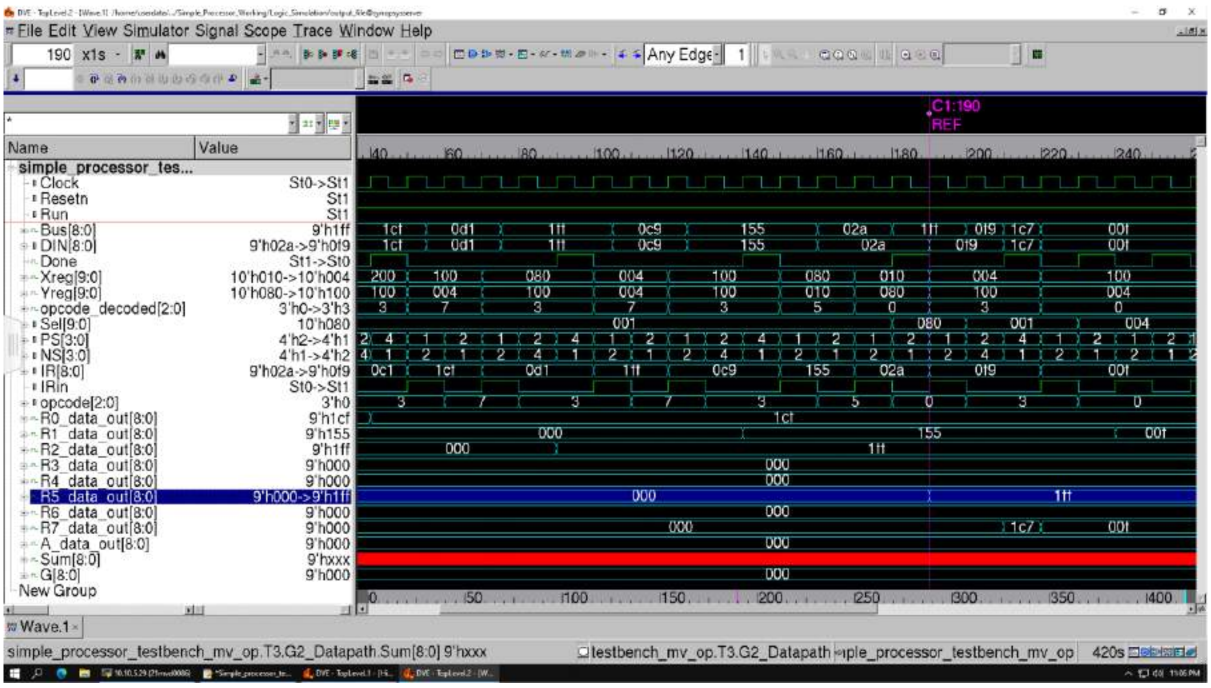
2. Move Operation

```
`timescale 1ns/1ns
module simple_processor_testbench_mv_op ();
reg Run, Resetn, Clock;
reg [8:0] DIN;
wire [8:0] Bus;
wire Done;

controller_new U1 (DIN, Run, Resetn, Clock, Done, R0out, R1out,
R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout, LdR0, LdR1, LdR2, LdR3, LdR4, LdR5,
LdR6, LdR7, LdA, LdG, Add_sub);

datapath_register_array U2 (R0out, R1out, R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout,
Clock,rst,R0in,
R1in,R2in,R3in,R4in,R5in,R6in,R7in,Ain,Bus,DIN,AddSub,Gin);


initial
Clock = 1'b1;

always #5 Clock = ~Clock;

initial
begin
$set_toggle_region(simple_processor_testbench_mv_op.U1,
simple_processor_testbench_mv_op.U2);
$toggle_start();
// ...

Resetn = 1'b0;
#10 Resetn = 1'b1;
Run = 1'b1;

DIN = 9'b011_000_001; // mvi operation
#20 DIN = 9'b111_001_111; // immediate data
//R0 is loaded with 111001111;
#5;

#20 DIN = 9'b011_010_001; // mvi operation
#20 DIN = 9'b111_111_111; // immediate data after 50ns from 1st
instuction @t=10
// R2 is loaded with 111111111;
```

```
#30 DIN = 9'b011_001_001; // mvi operation
#20 DIN = 9'b101_010_101; // immediate data
//R1 is loaded with 101010101;
#5;

#30 DIN = 9'b000_101_010;

#30 DIN = 9'b011_111_001; // mvi operation
#20 DIN = 9'b111_000_111; // immediate data
//R1 is loaded with 101010101;
#10 DIN = 9'b000_001_111;

// ...
$toggle_stop();
$toggle_report("Simple_Processor_SAIF_Move_Operation.saif",    1.0e-
12, "simple_processor_testbench_mv_op");

#200 $stop;
end

endmodule
```

Date:18/5/2022

Waveforms:

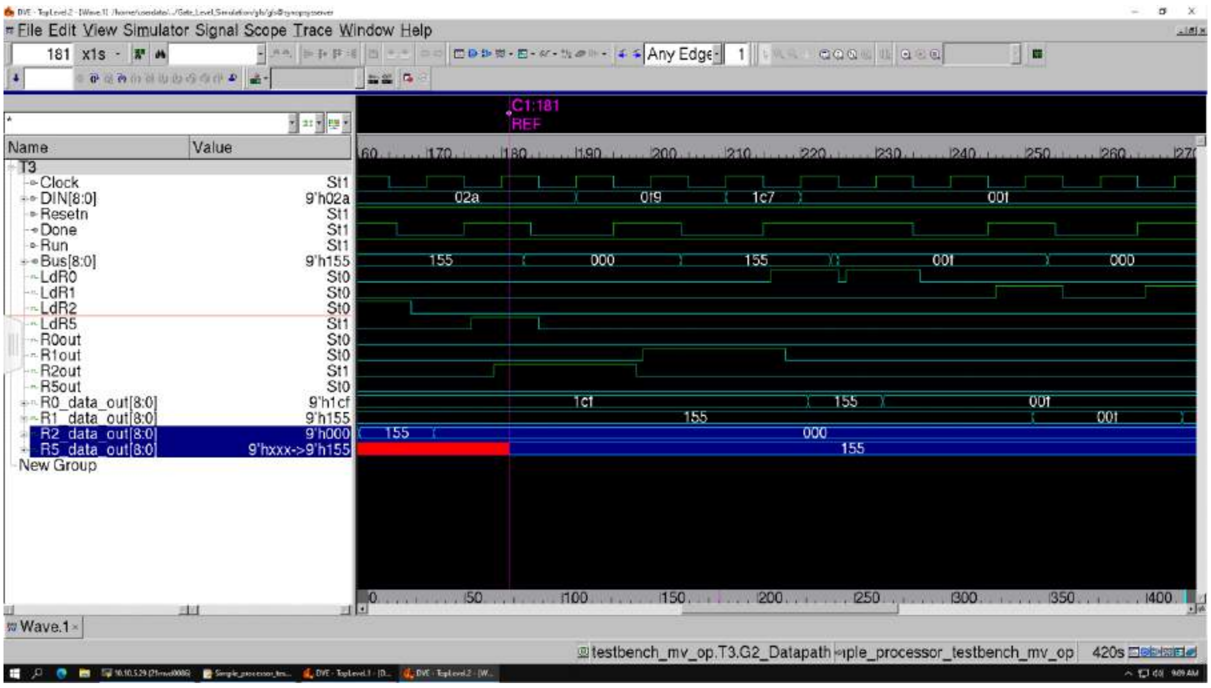


```
Name                    Value
simple_processor_tes...
  Clock                 St0->St1
  Resetn                St1
  Run                   St1
  Bus[8:0]              9'h1ff
  DIN[8:0]              9'h02a->9'h019
  Done                  St1->St0
  Xreg[9:0]             10'h010->10'h004
  Yreg[9:0]             10'h080->10'h100
  opcode_decoded[2:0]   3'h0->3'h3
  Sel[9:0]              10'h080
  PS[3:0]               4'h2->4'h1
  NS[3:0]               4'h1->4'h2
  IR[8:0]               9'h02a->9'h019
  IRin                  St0->St1
  opcode[2:0]           3'h0
  R0_data_out[8:0]      9'h1cf
  R1_data_out[8:0]      9'h155
  R2_data_out[8:0]      9'h1ff
  R3_data_out[8:0]      9'h000
  R4_data_out[8:0]      9'h000
  R5_data_out[8:0]      9'h000->9'h1ff
  R6_data_out[8:0]      9'h000
  R7_data_out[8:0]      9'h000
  A_data_out[8:0]       9'h000
  Sum[8:0]              9'hxxx
  G[8:0]                9'h000
New Group

simple_processor_testbench_mv_op.T3.G2_Datapath.Sum[8:0] 9'hxxx    testbench_mv_op.T3.G2_Datapath  ple_processor_testbench_mv_op  420s
```

Figure 1.7: The Move operation taking place from R2 to R5.

In the Figure 1.7 we are doing Move operation. We have done the Move immediate operation for Register R0, R1 and R2 already. At time instant from t=170ns to 190ns we observe that. The Move operation takes two states (two clock pulses) to complete the operation. We observe the data stored in Register R2 is copied to Register R5

Gate Level Simulation Waveforms:



```
Name                    Value
T3
  Clock                 St1
  DIN[8:0]              9'h02a
  Resetn                St1
  Done                  St1
  Run                   St1
  Bus[8:0]              9'h155
  LdR0                  St0
  LdR1                  St0
  LdR2                  St0
  LdR5                  St1
  R0out                 St0
  R1out                 St0
  R2out                 St1
  R5out                 St0
  R0_data_out[8:0]      9'h1cf
  R1_data_out[8:0]      9'h155
  R2_data_out[8:0]      9'h000
  R5_data_out[8:0]      9'hxxx->9'h155
New Group

testbench_mv_op.T3.G2_Datapath  ple_processor_testbench_mv_op  420s
```

Figure 1.8: The Move operation taking place from R2 to R5.

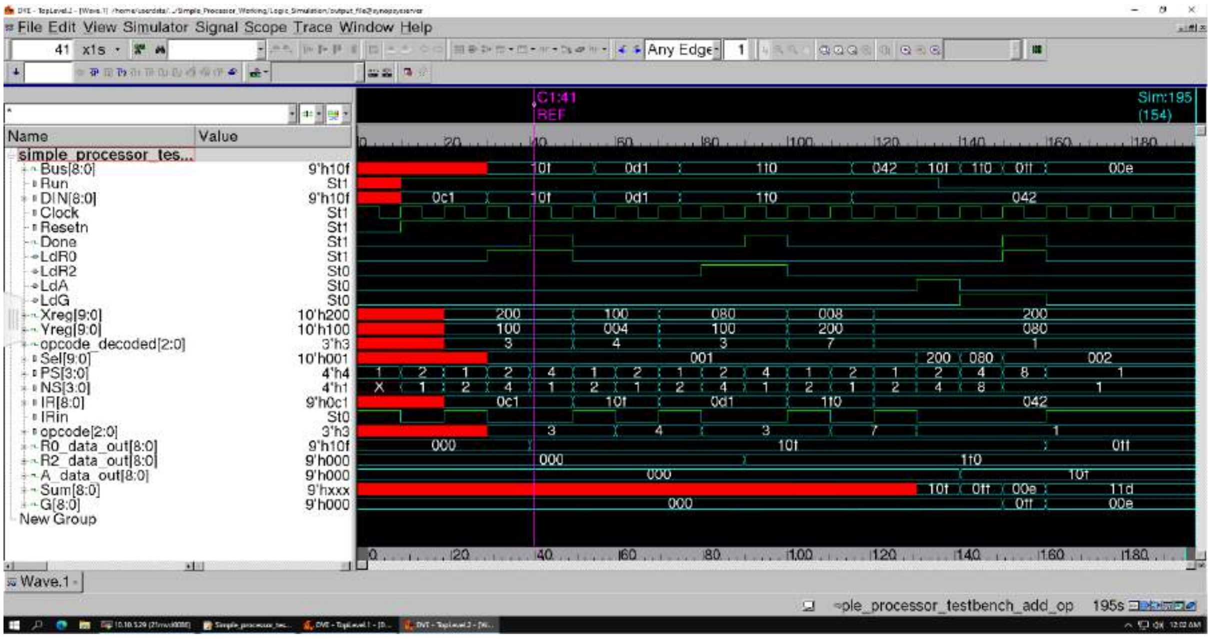
Date:18/5/2022

3. Addition Operation

```
`timescale 1ns/1ns
module simple_processor_testbench_add_op();
reg Run, Resetn, Clock;
reg [8:0] DIN;
wire [8:0] Bus;
wire Done;

controller_new G1 (DIN, Run, Resetn, Clock, Done, R0out, R1out,
R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout, LdR0, LdR1, LdR2, LdR3, LdR4, LdR5,
LdR6, LdR7, LdA, LdG, Add_sub);

datapath_register_array G2 (R0out, R1out, R2out, R3out, R4out, R5out,
R6out, R7out, Gout, DINout, Clock, Resetn, LdR0,
LdR1, LdR2, LdR3, LdR4, LdR5, LdR6, LdR7, LdA, Bus, DIN,
Add_sub, LdG);

initial
Clock = 1'b1;

always #5 Clock = ~Clock;

initial
begin
$set_toggle_region(simple_processor_testbench_add_op.G1,
simple_processor_testbench_add_op.G2);
$toggle_start();
// ...

Resetn = 1'b0;
#10 Resetn = 1'b1;
Run = 1'b1;

DIN = 9'b011_000_001; // mvi operation
#20 DIN = 9'b111_001_111; // immediate data
//R0 is loaded with 111001111;
#5;

/*
#20 DIN = 9'b011_010_001; // mvi operation
#20 DIN = 9'b111_111_111; // immediate data after 50ns from 1st
instuction @t=10
// R2 is loaded with 111111111;
*/
```

```
/*
#30 DIN = 9'b011_001_001; // mvi operation
#20 DIN = 9'b101_010_101; // immediate data
//R1 is loaded with 101010101;
*/
#40 DIN = 9'b001_000_010;
#20 Run =0;

// ...
$toggle_stop();
$toggle_report("Simple_Processor_SAIF_ADD_Operation.saif",1.0e-
12, "simple_processor_testbench_add_op");

#10 $stop;
end
endmodule
```

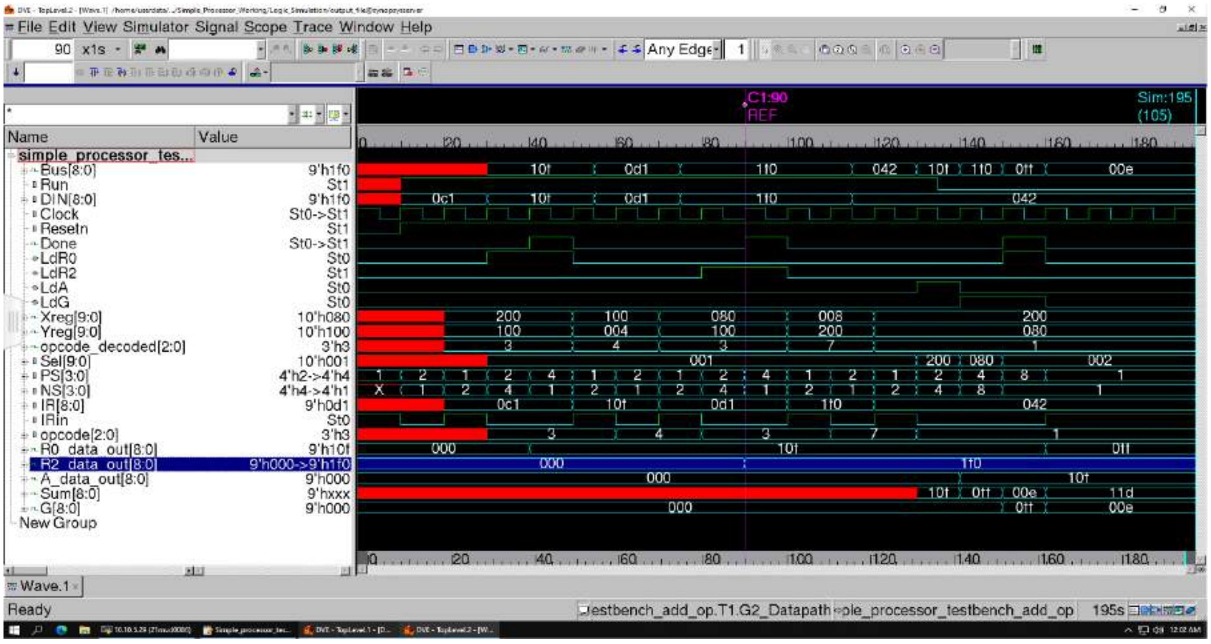
Waveforms:



Name	Value
simple_processor_tes...	
Bus[8:0]	9'h10f
Run	St1
DIN[8:0]	9'h10f
Clock	St1
Resetn	St1
Done	St1
LdR0	St1
LdR2	St0
LdA	St0
LdG	St0
Xreg[9:0]	10'h200
Yreg[9:0]	10'h100
opcode_decoded[2:0]	3'h3
Sel[9:0]	10'h001
PS[3:0]	4'h4
NS[3:0]	4'h1
IR[8:0]	9'h0c1
IRin	St0
opcode[2:0]	3'h3
R0_data_out[8:0]	9'h10f
R2_data_out[8:0]	9'h000
A_data_out[8:0]	9'h000
Sum[8:0]	9'hxxx
G[8:0]	9'h000
New Group	

Figure 1.9: The Load operation for R0.

In the figure we observe that the data 9’h10F loaded to register R0. Observe from time instant from 0 to 40ns no data loaded. At t=40ns we observe that R0 is loaded with the data 9’h10F. We are doing Move immediate operation for Register R0.

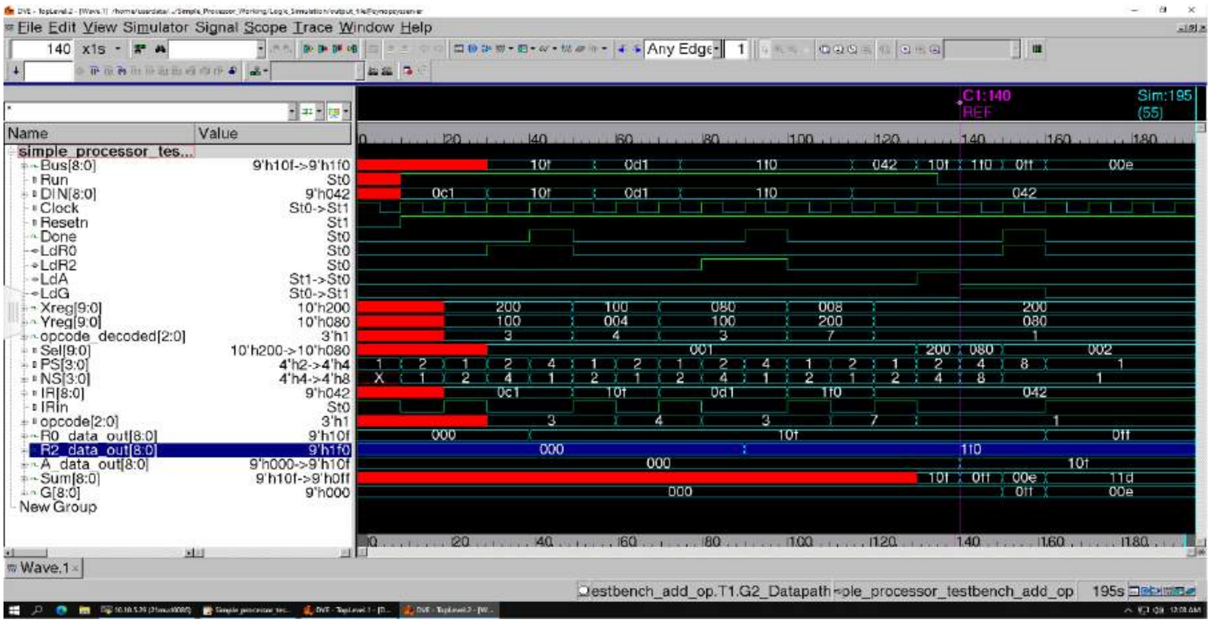


Name	Value
simple_processor_tes...	
Bus[8:0]	9'h1f0
Run	St1
DIN[8:0]	9'h1f0
Clock	St0->St1
Resetn	St1
Done	St0->St1
LdR0	St0
LdR2	St1
LdA	St0
LdG	St0
Xreg[9:0]	10'h080
Yreg[9:0]	10'h100
opcode_decoded[2:0]	3'h3
Sel[9:0]	10'h001
PS[3:0]	4'h2->4'h4
NS[3:0]	4'h4->4'h1
IR[8:0]	9'h0d1
IRin	St0
opcode[2:0]	3'h3
R0_data_out[8:0]	9'h10f
R2_data_out[8:0]	9'h000->9'h1f0
A_data_out[8:0]	9'h000
Sum[8:0]	9'hxxx
G[8:0]	9'h000
New Group	

Figure 1.10: The Load operation for R2.

In the figure we observe that the data 9’h1F0 loaded to register R2. Observe from time instant from 0 to 90ns no data loaded. At t=90ns we observe that R2 is loaded with the data 9’h1F0. We are doing Move immediate operation for Register R2.

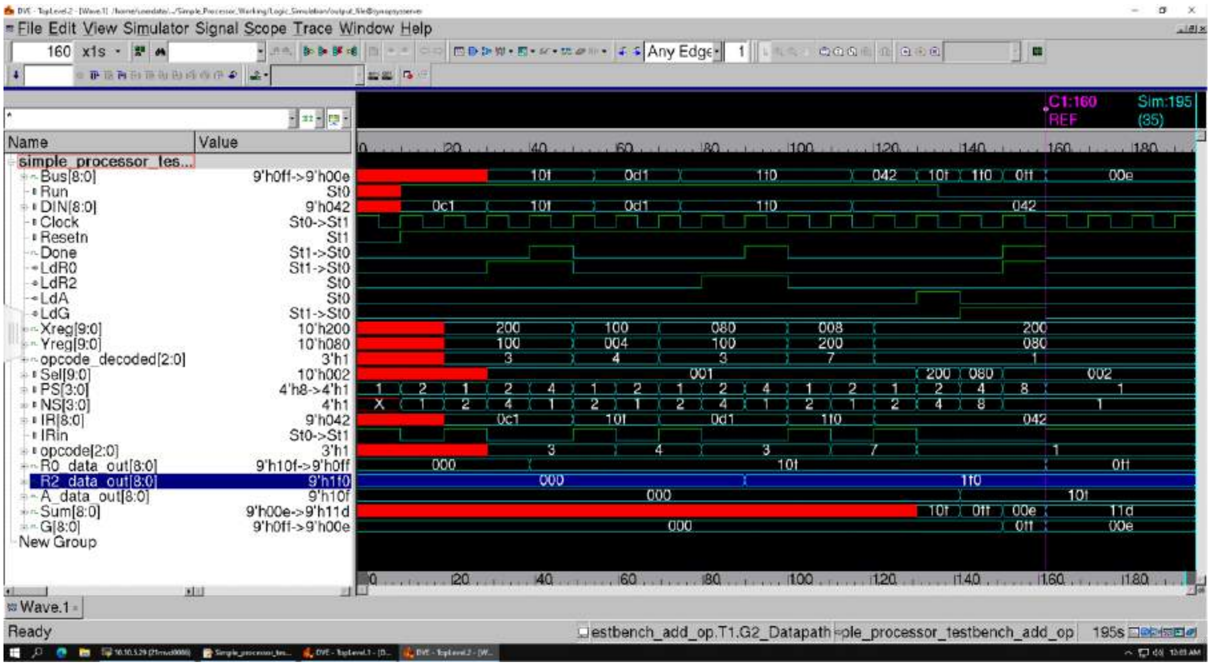
Date:18/5/2022



Name	Value
simple_processor_tes...	
Bus[8:0]	9'h10f->9'h1f0
Run	St0
DIN[8:0]	9'h042
Clock	St0->St1
Resetn	St1
Done	St0
LdR0	St0
LdR2	St0
LdA	St1->St0
LdG	St0->St1
Xreg[9:0]	10'h200
Yreg[9:0]	10'h080
opcode_decoded[2:0]	3'h1
Sel[9:0]	10'h200->10'h080
PS[3:0]	4'h2->4'h4
NS[3:0]	4'h4->4'h8
IR[8:0]	9'h042
IRin	St0
opcode[2:0]	3'h1
R0_data_out[8:0]	9'h10f
R2_data_out[8:0]	9'h1f0
A_data_out[8:0]	9'h000->9'h10f
Sum[8:0]	9'h10f->9'h0ff
G[8:0]	9'h000
New Group	

Figure 1.11: The Addition operation opcode decoding.

In the Figure 1.6 we observe that the R0 data value is copied to Register A. The control signal LdA is generated high and goes low after that, the addition takes place with bus loaded with R2 data. In next clock pulse we observe that the LdG signal became high and addition result is stored to Register G.

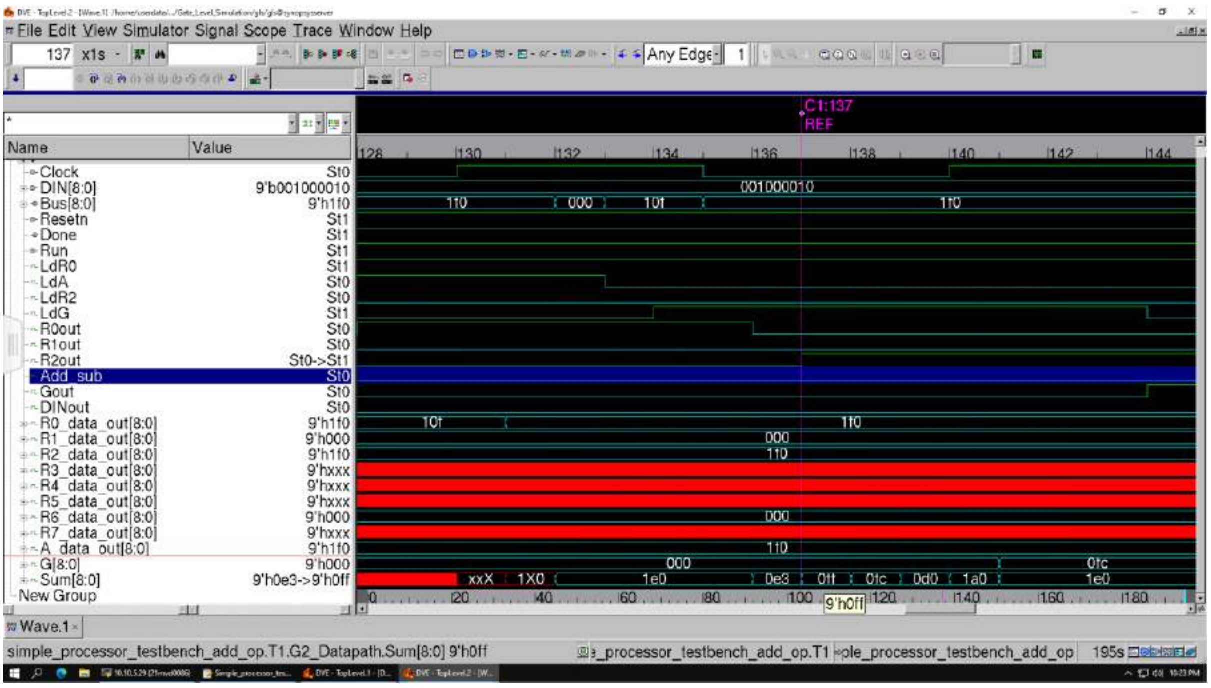


Name	Value
simple_processor_tes...	
Bus[8:0]	9'h0ff->9'h00e
Run	St0
DIN[8:0]	9'h042
Clock	St0->St1
Resetn	St1
Done	St1->St0
LdR0	St1->St0
LdR2	St0
LdA	St0
LdG	St1->St0
Xreg[9:0]	10'h200
Yreg[9:0]	10'h080
opcode_decoded[2:0]	3'h1
Sel[9:0]	10'h002
PS[3:0]	4'h8->4'h1
NS[3:0]	4'h1
IR[8:0]	9'h042
IRin	St0->St1
opcode[2:0]	3'h1
R0_data_out[8:0]	9'h10f->9'h0ff
R2_data_out[8:0]	9'h1f0
A_data_out[8:0]	9'h10f
Sum[8:0]	9'h00e->9'h11d
G[8:0]	9'h0ff->9'h00e
New Group	

Figure 1.12: The Addition operation opcode decoding.

In the Figure 1.7 we observe that sum result stored Register G is stored back to register R0.

Date:18/5/2022



Name	Value
Clock	St0
DIN[8:0]	9'b001000010
Bus[8:0]	9'h1f0
Resetn	St1
Done	St1
Run	St1
LdR0	St1
LdA	St0
LdR2	St0
LdG	St1
R0out	St0
R1out	St0
R2out	St0->St1
Add_sub	St0
Gout	St0
DINout	St0
R0_data_out[8:0]	9'h1f0
R1_data_out[8:0]	9'h000
R2_data_out[8:0]	9'h1f0
R3_data_out[8:0]	9'hxxx
R4_data_out[8:0]	9'hxxx
R5_data_out[8:0]	9'hxxx
R6_data_out[8:0]	9'h000
R7_data_out[8:0]	9'hxxx
A_data_out[8:0]	9'h1f0
G[8:0]	9'h000
Sum[8:0]	9'h0e3->9'h0ff
New Group	

simple_processor_testbench_add_op.T1.G2_Datapath.Sum[8:0] 9'h0ff

Figure 1.13: The Addition operation not working properly due to gate delays the Multiple Load signals are generating at the same time. This problem is causing the result not to be attached to the registers. The Addition result is 0x0ff available at 137ns the data is not latched to the register R0.

Date:18/5/2022

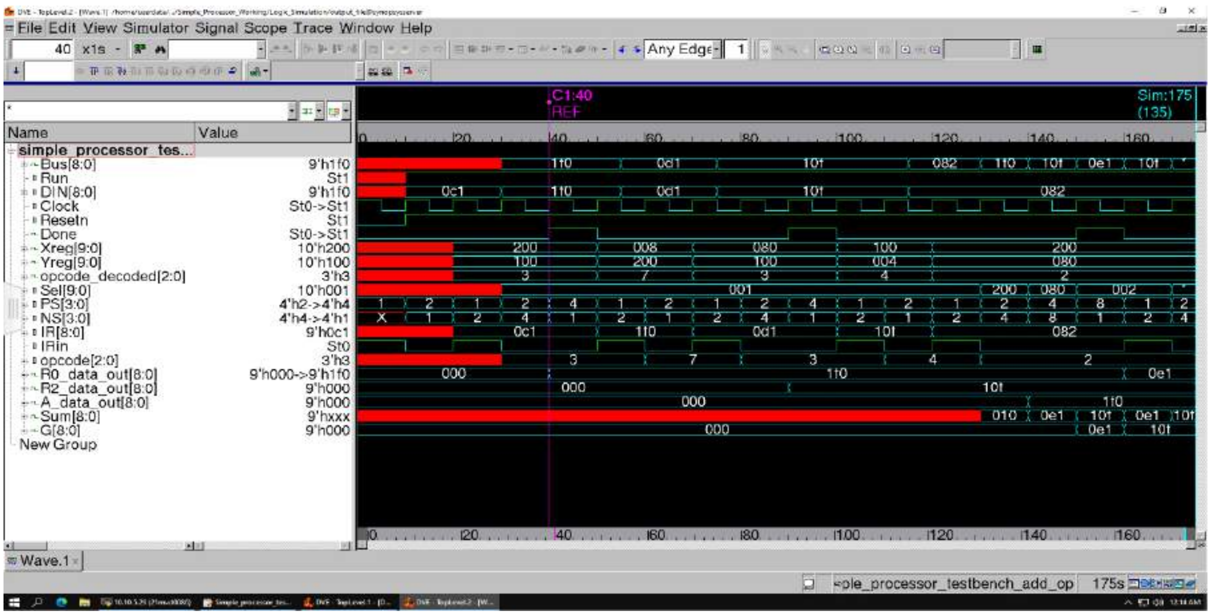
4. **Subtraction Operation**

```
`timescale 1ns/1ns
module simple_processor_testbench_add_op();
reg Run, Resetn, Clock;
reg [8:0] DIN;
wire [8:0] Bus;
wire Done;

simple_processor_Top T1 (Run,Resetn,Clock,DIN,Bus,Done);

initial
Clock = 1'b1;

always #5 Clock = ~Clock;

initial
begin
$set_toggle_region(simple_processor_testbench_add_op.T1);
$toggle_start();
// ...

Resetn = 1'b0;
#10 Resetn = 1'b1;
Run = 1'b1;

DIN = 9'b011_000_001; // mvi operation
#20 DIN = 9'b111_110_000; // immediate data
//R0 is loaded with 111001111;
#5;


#20 DIN = 9'b011_010_001; // mvi operation
#20 DIN = 9'b100_001_111; // immediate data after 50ns from 1st instuction @t=10
// R2 is loaded with 111111111;


/*
#30 DIN = 9'b011_001_001; // mvi operation
#20 DIN = 9'b101_010_101; // immediate data
//R1 is loaded with 101010101;
*/
```

```
#40 DIN = 9'b010_000_010; // sub R0=R0-R2
#60 Run = 0;

// ...
$toggle_stop();
$toggle_report("Simple_Processor_SAIF_ADD_Operation.saif",    1.0e-
12, "simple_processor_testbench_add_op");

#0 $stop;
end
endmodule
```

Date:18/5/2022

Waveforms:

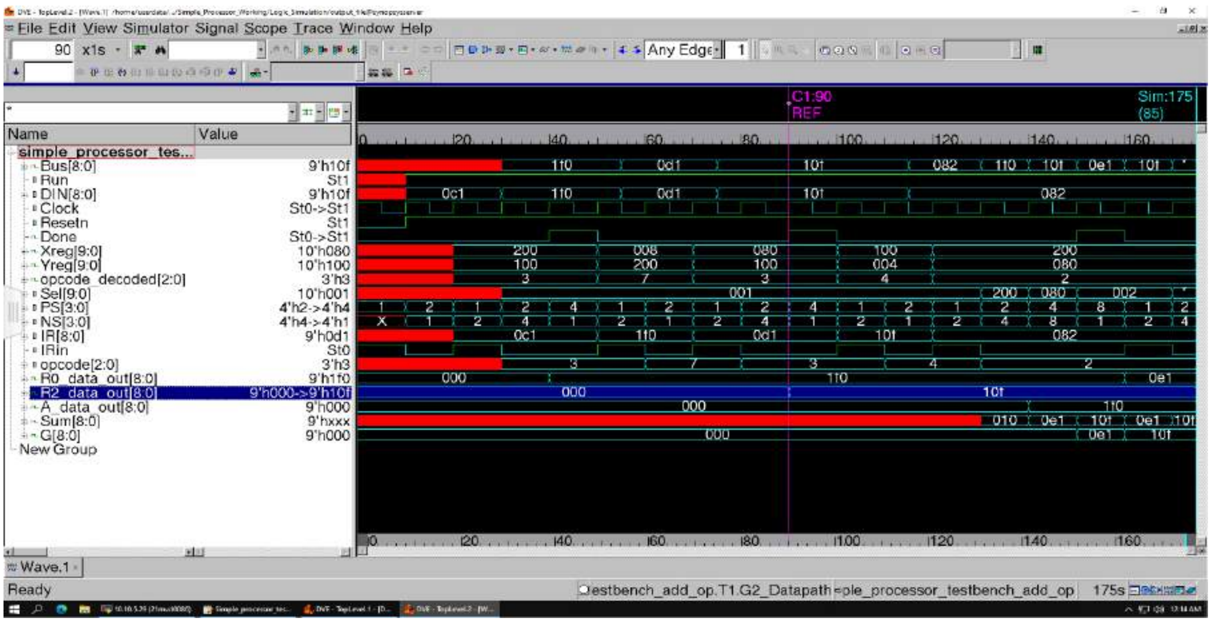


Name	Value
simple_processor_tes...	
Bus[8:0]	9'h1f0
Run	St1
DIN[8:0]	9'h1f0
Clock	St0->St1
Resetn	St1
Done	St0->St1
Xreg[9:0]	10'h200
Yreg[9:0]	10'h100
opcode_decoded[2:0]	3'h3
Sel[9:0]	10'h001
PS[3:0]	4'h2->4'h4
NS[3:0]	4'h4->4'h1
IR[8:0]	9'h0c1
IRin	St0
opcode[2:0]	3'h3
R0_data_out[8:0]	9'h000->9'h1f0
R2_data_out[8:0]	9'h000
A_data_out[8:0]	9'h000
Sum[8:0]	9'hxxx
G[8:0]	9'h000
New Group	

File Edit View Simulator Signal Scope Trace Window Help
Any Edge
C1:40
REF
Sim:175
(135)
Wave.1
ple_processor_testbench_add_op
175s

Figure 1.14: The Load operation for R0.

In the Figure 1.14 we observe that the data 9’h1F0 loaded to register R0. Observe from time instant from 0 to 40ns no data loaded. At t=40ns we observe that R0 is loaded with the data 9’h1F0. We are doing Move immediate operation for Register R0.

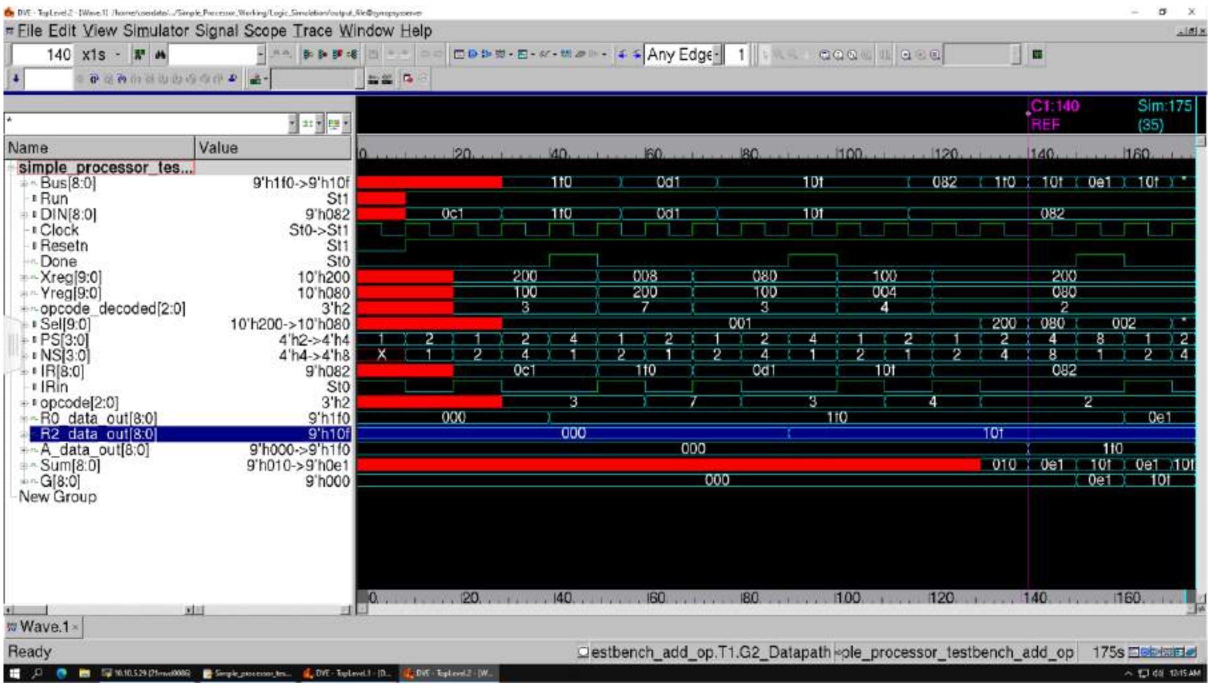


Name	Value
simple_processor_tes...	
Bus[8:0]	9'h10f
Run	St1
DIN[8:0]	9'h10f
Clock	St0->St1
Resetn	St1
Done	St0->St1
Xreg[9:0]	10'h080
Yreg[9:0]	10'h100
opcode_decoded[2:0]	3'h3
Sel[9:0]	10'h001
PS[3:0]	4'h2->4'h4
NS[3:0]	4'h4->4'h1
IR[8:0]	9'h0d1
IRin	St0
opcode[2:0]	3'h3
R0_data_out[8:0]	9'h1f0
R2_data_out[8:0]	9'h000->9'h10f
A_data_out[8:0]	9'h000
Sum[8:0]	9'hxxx
G[8:0]	9'h000
New Group	

File Edit View Simulator Signal Scope Trace Window Help
Any Edge
C1:90
REF
Sim:175
(85)
Wave.1
Ready
estbench_add_op.T1.G2_Datapath
ple_processor_testbench_add_op
175s

Figure 1.15: The Load operation for R2.

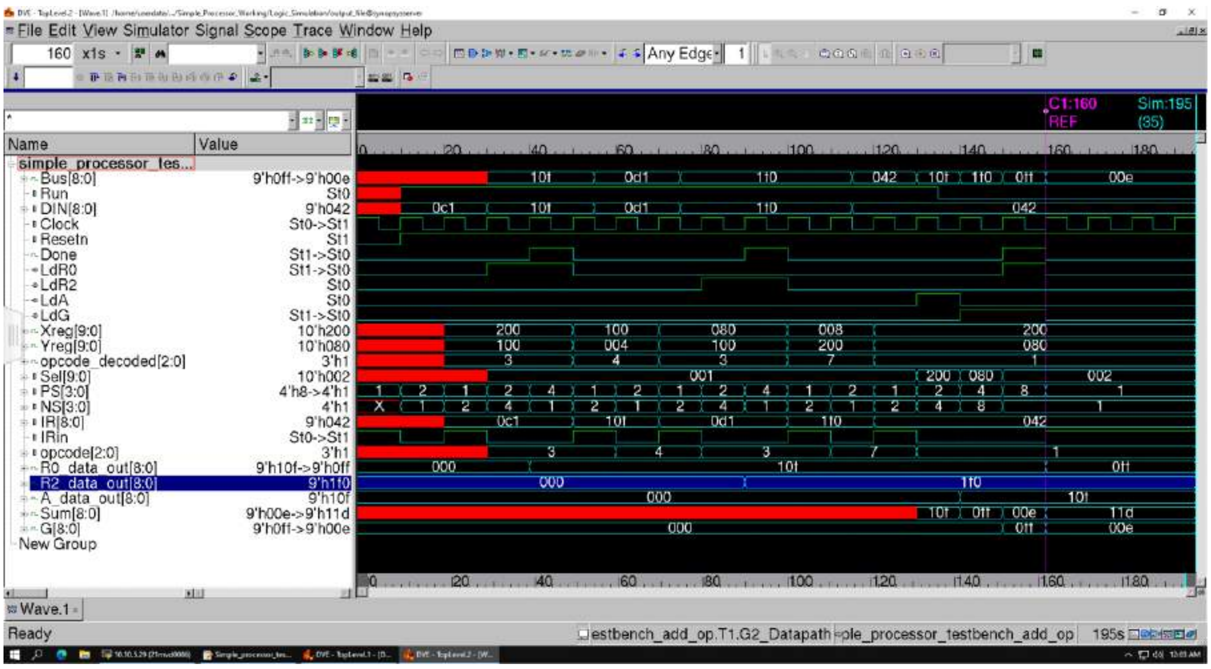
In the Figure 1.15 we observe that the data 9’h10F loaded to register R2. Observe from time instant from 0 to 90ns no data loaded. At t=90ns we observe that R2 is loaded with the data 9’h10F. We are doing Move immediate operation for Register R2.



Name	Value
simple_processor_tes...	
Bus[8:0]	9'h110->9'h10f
Run	St1
DIN[8:0]	9'h082
Clock	St0->St1
Resetn	St1
Done	St0
Xreg[9:0]	10'h200
Yreg[9:0]	10'h080
opcode_decoded[2:0]	3'h2
Sel[9:0]	10'h200->10'h080
PS[3:0]	4'h2->4'h4
NS[3:0]	4'h4->4'h8
IR[8:0]	9'h082
IRin	St0
opcode[2:0]	3'h2
R0_data_out[8:0]	9'h110
R2_data_out[8:0]	9'h10f
A_data_out[8:0]	9'h000->9'h110
Sum[8:0]	9'h010->9'h0e1
G[8:0]	9'h000
New Group	

Figure 1.16: The Subtraction operation opcode decoding.

In the Figure 1.16 we observe that the R0 data value is copied to Register A. The control signal LdA is generated high and goes low after that, the subtraction takes place with bus loaded with R2 data. In next clock pulse we observe that the LdG signal became high and subtraction result is stored to Register G.



Name	Value
simple_processor_tes...	
Bus[8:0]	9'h0ff->9'h00e
Run	St0
DIN[8:0]	9'h042
Clock	St0->St1
Resetn	St1
Done	St1->St0
LdR0	St1->St0
LdR2	St0
LdA	St0
LdG	St1->St0
Xreg[9:0]	10'h200
Yreg[9:0]	10'h080
opcode_decoded[2:0]	3'h1
Sel[9:0]	10'h002
PS[3:0]	4'h8->4'h1
NS[3:0]	4'h1
IR[8:0]	9'h042
IRin	St0->St1
opcode[2:0]	3'h1
R0_data_out[8:0]	9'h10f->9'h0ff
R2_data_out[8:0]	9'h110
A_data_out[8:0]	9'h10f
Sum[8:0]	9'h00e->9'h11d
G[8:0]	9'h0ff->9'h00e
New Group	

Figure 1.17: The Subtraction operation opcode decoding.

In the Figure 1.17 we observe that difference result stored Register G is stored back to register R0.

Date:18/5/2022

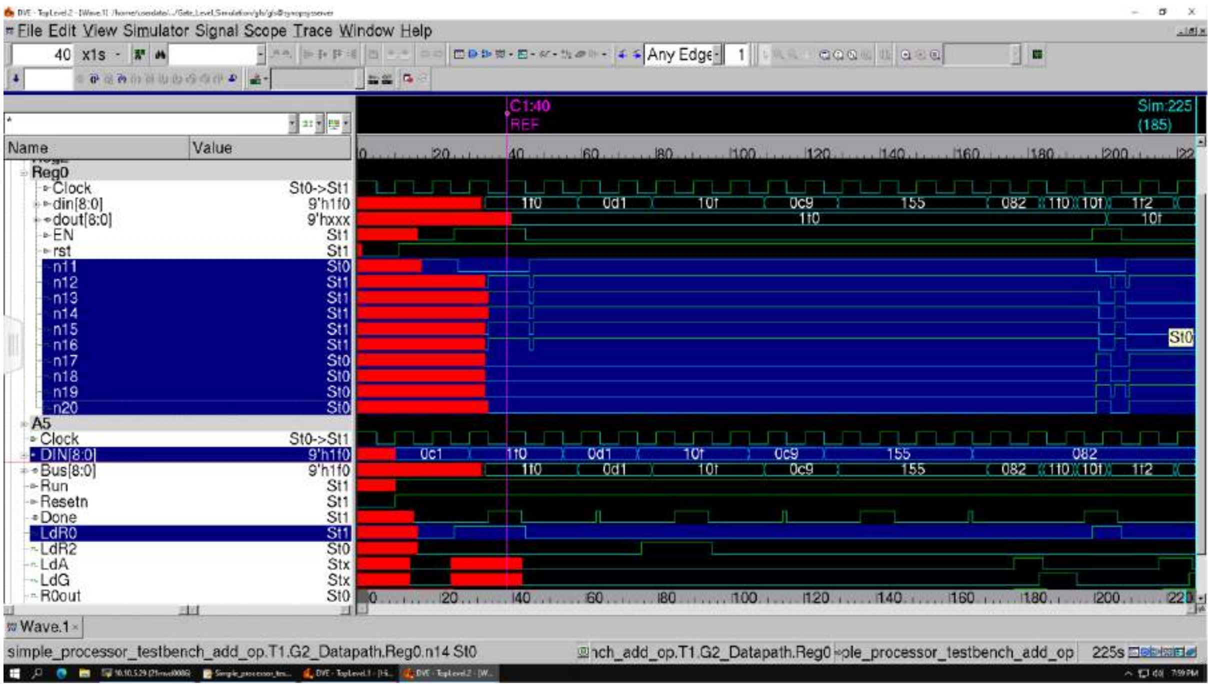


Figure 1.18: The Load operation for R0.

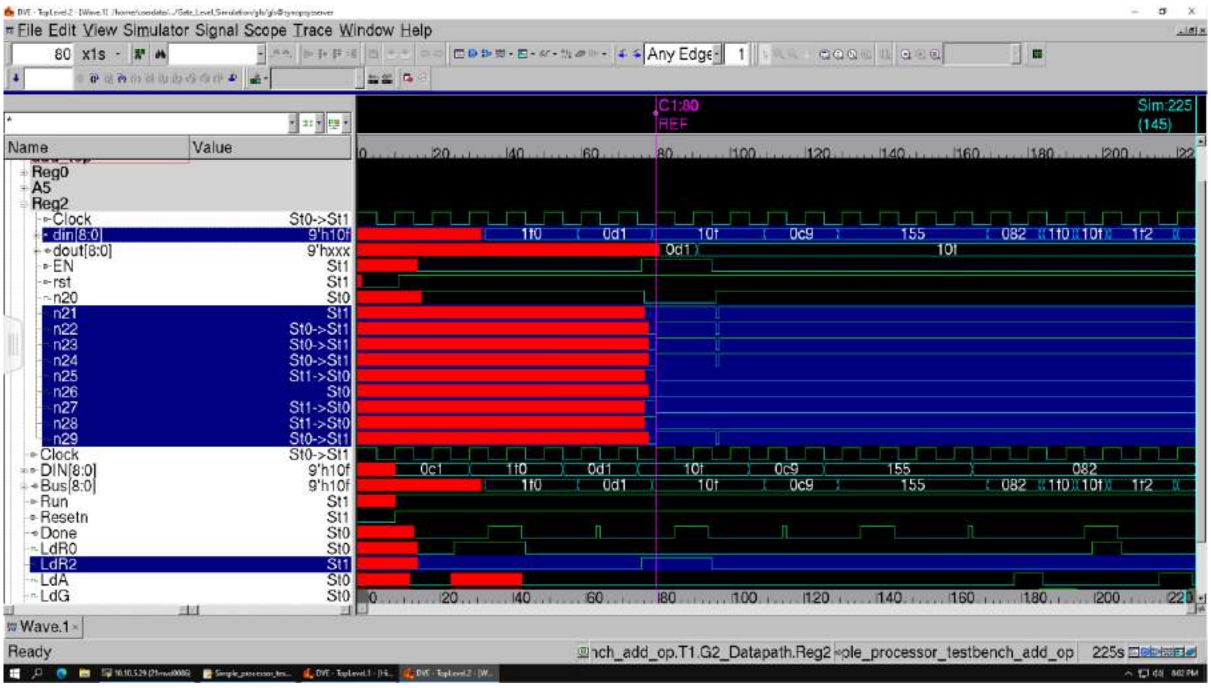
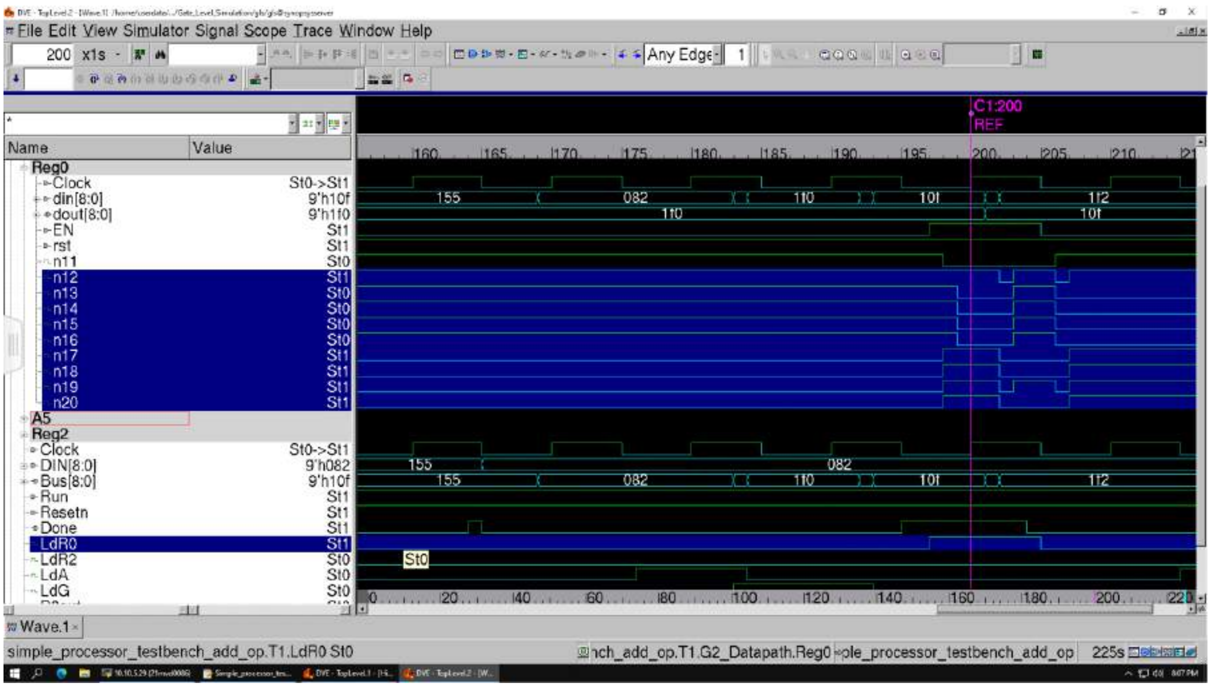


Figure 1.19: The Load operation for R2.



Name	Value
Reg0	
Clock	St0->St1
din[8:0]	9'h10f
dout[8:0]	9'h110
EN	St1
rst	St1
n11	St0
n12	St1
n13	St0
n14	St0
n15	St0
n16	St0
n17	St1
n18	St1
n19	St1
n20	St1
A5	
Reg2	
Clock	St0->St1
DIN[8:0]	9'h082
Bus[8:0]	9'h10f
Run	St1
Resetn	St1
Done	St1
LdR0	St1
LdR2	St0
LdA	St0
LdG	St0

simple_processor_testbench_add_op.T1.LdR0 St0

Figure 1.20 The Subtraction operation of R0 and R2 performed due to gate delays the Multiple Load signals are generating at the same time. This problem is causing the result not to be attached to the registers.

Inference:

1. The Simple Processor 9 bit was designed using Verilog code. The simulation was done using Synopsys VCS tool.
2. The opcodes used were Move, move immediate, Addition and Subtraction operation was done and verified.
3. The Gate Level Simulation was done using VCS tool by running the Testbench file and Netlist file generated from it.

Date:18/5/2022

Section 2 Formal Verification

Aim: The Simple Processor Formal Verification

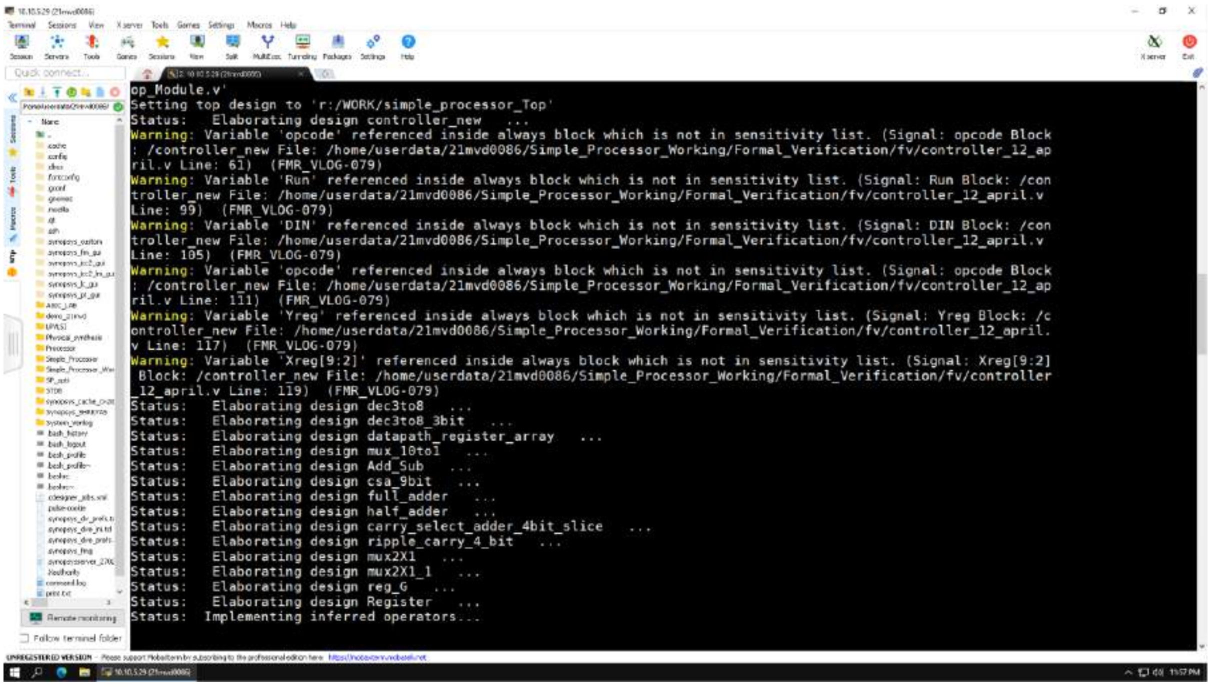
The Formal verification is act of proving or disproving the correctness of intended algorithms underlying a system with respect to a certain formal specification or property. In this project we are using fm shell of Synopsys tool for verification.

The FM Shell Tickle Script:

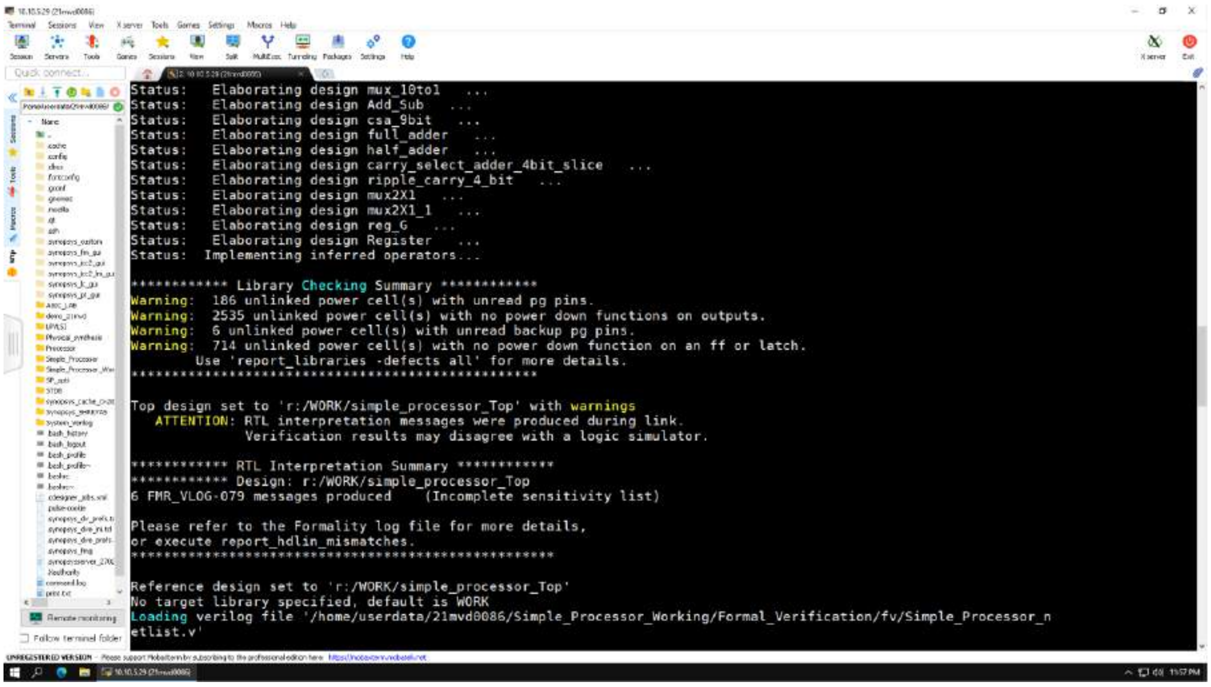
```
#set Synopsys Auto Setup flow from minimum setup
set synopsys_auto_setup true
set DESIGN_REF_PATH "/home/synopsys/SAED14nm_EDK"
set_svf Simple_Processor.svf;
# SVF file
read_db -tech " \
${DESIGN_REF_PATH}/stdcell_hvt/db_nldm/saed14hvt_tt0p8v125c.db \
${DESIGN_REF_PATH}/stdcell_lvt/db_nldm/saed14lvt_tt0p8v125c.db \
${DESIGN_REF_PATH}/stdcell_rvt/db_nldm/saed14rvt_tt0p8v125c.db"
read_verilog -r ./controller_12_april.v
read_verilog -r ./Datapath_path_work_version_2.v
read_verilog -r ./Decoders.v
read_verilog -r ./Simple_Processor_Top_Module.v
set_top simple_processor_Top
read_verilog -i ./Simple_Processor_netlist.v
set_top simple_processor_Top
match
verify
report_guidance -summary
```

Date:18/5/2022

The Formal Verification Screenshots:



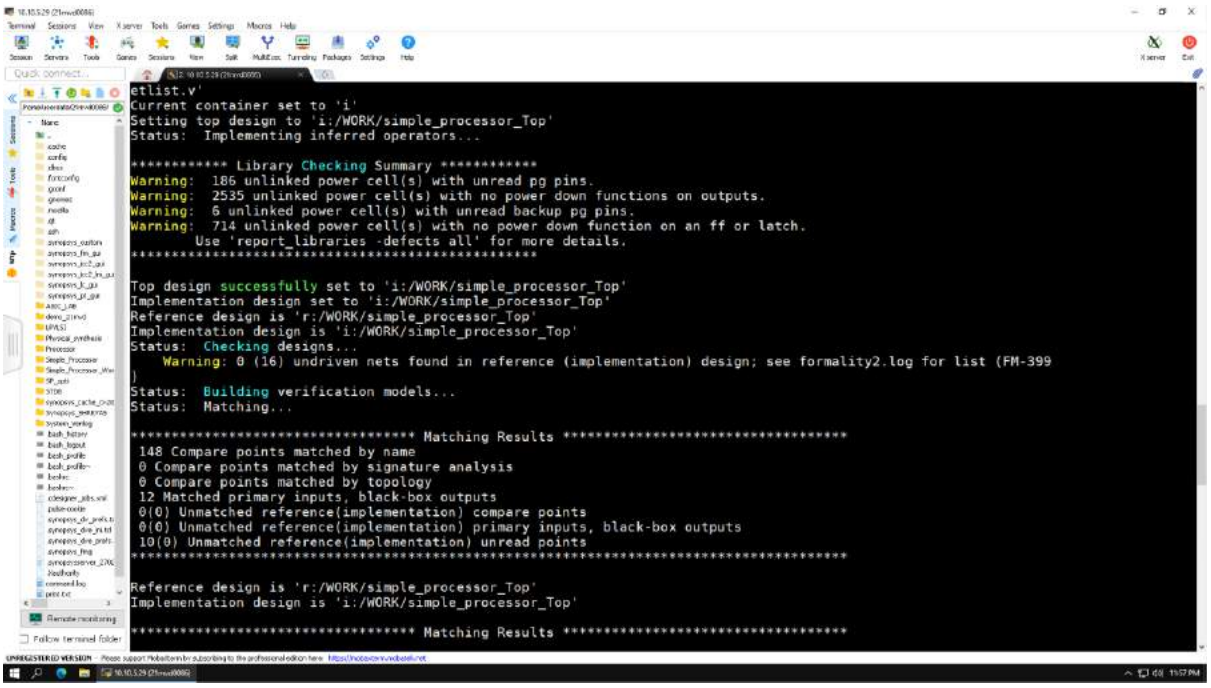
```
op_Module.v'
Setting top design to 'r:/WORK/simple_processor_Top'
Status:   Elaborating design controller_new   ...
Warning: Variable 'opcode' referenced inside always block which is not in sensitivity list. (Signal: opcode Block
: /controller_new File: /home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/controller_12_ap
ril.v Line: 61)  (FMR_VLOG-079)
Warning: Variable 'Run' referenced inside always block which is not in sensitivity list. (Signal: Run Block: /con
troller_new File: /home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/controller_12_april.v
Line: 99)  (FMR_VLOG-079)
Warning: Variable 'DIN' referenced inside always block which is not in sensitivity list. (Signal: DIN Block: /con
troller_new File: /home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/controller_12_april.v
Line: 105)  (FMR_VLOG-079)
Warning: Variable 'opcode' referenced inside always block which is not in sensitivity list. (Signal: opcode Block
: /controller_new File: /home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/controller_12_ap
ril.v Line: 111)  (FMR_VLOG-079)
Warning: Variable 'Yreg' referenced inside always block which is not in sensitivity list. (Signal: Yreg Block: /c
ontroller_new File: /home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/controller_12_april.
v Line: 117)  (FMR_VLOG-079)
Warning: Variable 'Xreg[9:2]' referenced inside always block which is not in sensitivity list. (Signal: Xreg[9:2]
 Block: /controller_new File: /home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/controller
_12_april.v Line: 119)  (FMR_VLOG-079)
Status:   Elaborating design dec3to8   ...
Status:   Elaborating design dec3to8_3bit   ...
Status:   Elaborating design datapath_register_array   ...
Status:   Elaborating design mux_10to1   ...
Status:   Elaborating design Add_Sub   ...
Status:   Elaborating design csa_9bit   ...
Status:   Elaborating design full_adder   ...
Status:   Elaborating design half_adder   ...
Status:   Elaborating design carry_select_adder_4bit_slice   ...
Status:   Elaborating design ripple_carry_4_bit   ...
Status:   Elaborating design mux2X1   ...
Status:   Elaborating design mux2X1_1   ...
Status:   Elaborating design reg_G   ...
Status:   Elaborating design Register   ...
Status:  Implementing inferred operators...
```



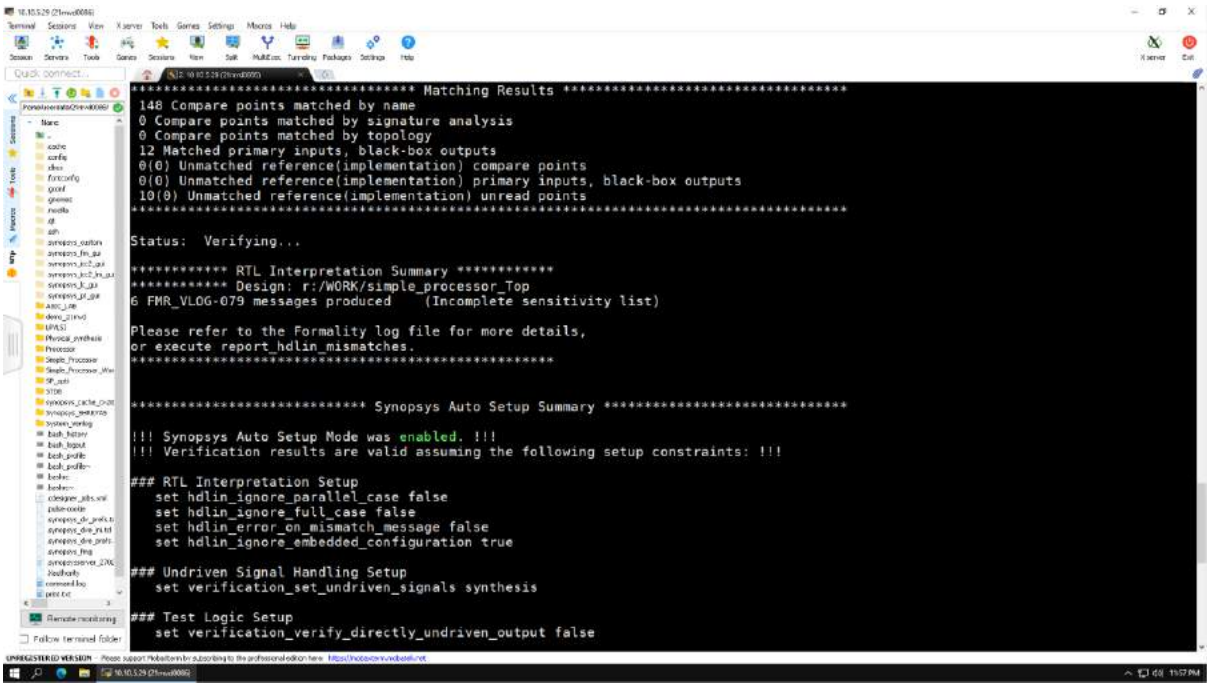
```
Status:   Elaborating design mux_10to1   ...
Status:   Elaborating design Add_Sub   ...
Status:   Elaborating design csa_9bit   ...
Status:   Elaborating design full_adder   ...
Status:   Elaborating design half_adder   ...
Status:   Elaborating design carry_select_adder_4bit_slice   ...
Status:   Elaborating design ripple_carry_4_bit   ...
Status:   Elaborating design mux2X1   ...
Status:   Elaborating design mux2X1_1   ...
Status:   Elaborating design reg_G   ...
Status:   Elaborating design Register   ...
Status:  Implementing inferred operators...

************ Library Checking Summary ************
Warning:  186 unlinked power cell(s) with unread pg pins.
Warning:  2535 unlinked power cell(s) with no power down functions on outputs.
Warning:  6 unlinked power cell(s) with unread backup pg pins.
Warning:  714 unlinked power cell(s) with no power down function on an ff or latch.
        Use 'report_libraries -defects all' for more details.
**************************************************

Top design set to 'r:/WORK/simple_processor_Top' with warnings
   ATTENTION: RTL interpretation messages were produced during link.
              Verification results may disagree with a logic simulator.

************ RTL Interpretation Summary ************
************ Design: r:/WORK/simple_processor_Top
6 FMR_VLOG-079 messages produced    (Incomplete sensitivity list)

Please refer to the Formality log file for more details,
or execute report_hdlin_mismatches.
****************************************************

Reference design set to 'r:/WORK/simple_processor_Top'
No target library specified, default is WORK
Loading verilog file '/home/userdata/21mvd0086/Simple_Processor_Working/Formal_Verification/fv/Simple_Processor_n
etlist.v'
```

Figure 2.1 The Formal Verification Screenshots part 1.

Date:18/5/2022

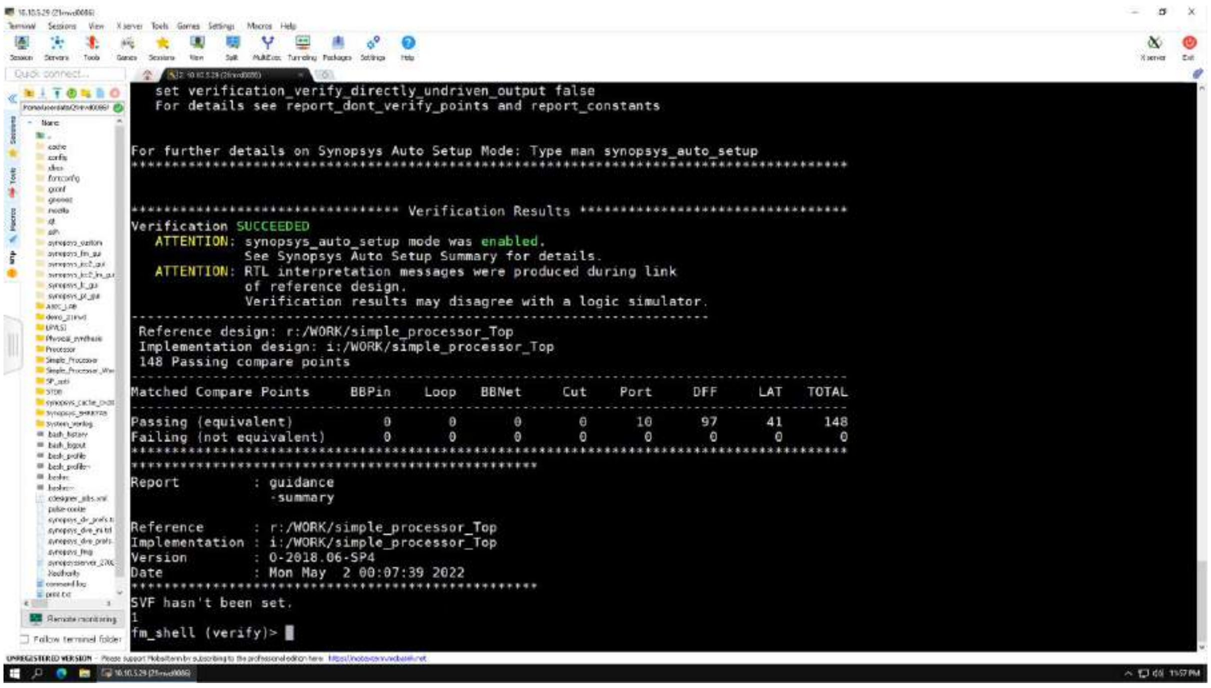


```
etlist.v'
Current container set to 'i'
Setting top design to 'i:/WORK/simple_processor_Top'
Status:  Implementing inferred operators...

************ Library Checking Summary ************
Warning:  186 unlinked power cell(s) with unread pg pins.
Warning:  2535 unlinked power cell(s) with no power down functions on outputs.
Warning:  6 unlinked power cell(s) with unread backup pg pins.
Warning:  714 unlinked power cell(s) with no power down function on an ff or latch.
        Use 'report_libraries -defects all' for more details.
**************************************************

Top design successfully set to 'i:/WORK/simple_processor_Top'
Implementation design set to 'i:/WORK/simple_processor_Top'
Reference design is 'r:/WORK/simple_processor_Top'
Implementation design is 'i:/WORK/simple_processor_Top'
Status:  Checking designs...
    Warning: 0 (16) undriven nets found in reference (implementation) design; see formality2.log for list (FM-399
)
Status:  Building verification models...
Status:  Matching...

********************************** Matching Results **********************************
 148 Compare points matched by name
 0 Compare points matched by signature analysis
 0 Compare points matched by topology
 12 Matched primary inputs, black-box outputs
 0(0) Unmatched reference(implementation) compare points
 0(0) Unmatched reference(implementation) primary inputs, black-box outputs
 10(0) Unmatched reference(implementation) unread points
**************************************************************************************

Reference design is 'r:/WORK/simple_processor_Top'
Implementation design is 'i:/WORK/simple_processor_Top'

********************************** Matching Results **********************************
```



```
********************************** Matching Results **********************************
 148 Compare points matched by name
 0 Compare points matched by signature analysis
 0 Compare points matched by topology
 12 Matched primary inputs, black-box outputs
 0(0) Unmatched reference(implementation) compare points
 0(0) Unmatched reference(implementation) primary inputs, black-box outputs
 10(0) Unmatched reference(implementation) unread points
**************************************************************************************

Status:  Verifying...

************ RTL Interpretation Summary ************
************ Design: r:/WORK/simple_processor_Top
6 FMR_VLOG-079 messages produced    (Incomplete sensitivity list)

Please refer to the Formality log file for more details,
or execute report_hdlin_mismatches.
****************************************************


**************************** Synopsys Auto Setup Summary ******************************

!!! Synopsys Auto Setup Mode was enabled. !!!
!!! Verification results are valid assuming the following setup constraints: !!!

### RTL Interpretation Setup
   set hdlin_ignore_parallel_case false
   set hdlin_ignore_full_case false
   set hdlin_error_on_mismatch_message false
   set hdlin_ignore_embedded_configuration true

### Undriven Signal Handling Setup
   set verification_set_undriven_signals synthesis

### Test Logic Setup
   set verification_verify_directly_undriven_output false
```

Figure 2.2 The Formal Verification Screenshots part 2.



```
   set verification_verify_directly_undriven_output false
   For details see report_dont_verify_points and report_constants


For further details on Synopsys Auto Setup Mode: Type man synopsys_auto_setup
****************************************************************************************


********************************* Verification Results *********************************
Verification SUCCEEDED
   ATTENTION: synopsys_auto_setup mode was enabled.
              See Synopsys Auto Setup Summary for details.
   ATTENTION: RTL interpretation messages were produced during link
              of reference design.
              Verification results may disagree with a logic simulator.
-----------------------------------------------------------------------
 Reference design: r:/WORK/simple_processor_Top
 Implementation design: i:/WORK/simple_processor_Top
 148 Passing compare points
----------------------------------------------------------------------------------------
Matched Compare Points     BBPin    Loop   BBNet     Cut    Port     DFF     LAT   TOTAL
----------------------------------------------------------------------------------------
Passing (equivalent)           0       0       0       0      10      97      41     148
Failing (not equivalent)       0       0       0       0       0       0       0       0
****************************************************************************************
**************************************************
Report         : guidance
                 -summary

Reference      : r:/WORK/simple_processor_Top
Implementation : i:/WORK/simple_processor_Top
Version        : O-2018.06-SP4
Date           : Mon May  2 00:07:39 2022
**************************************************
SVF hasn't been set.
1
fm_shell (verify)> 
```

Figure 2.3 The Formal Verification Screenshot showing SUCCEEDED.

Inference:

1. The Simple Processor 9 bit was designed using Verilog code. The simulation was done using Synopsys VCS tool.
2. The Formal Verification was done using fm shell by running the fm script tickle script. This file includes the netlist generated from Logic synthesis and matches with Simple Processor Top Module.

Date:18/5/2022

Section 3 Pre-Layout Simulation

Aim: The Simple Processor Pre-Layout Simulation

Script:

```
set DESIGN_REF_PATH "/home/synopsys/SAED14nm_EDK"

set SEARCH_PATH      " ./ \
        ${DESIGN_REF_PATH}/stdcell_rvt/db_nldm \
        ${DESIGN_REF_PATH}/stdcell_hvt/db_nldm \
        ${DESIGN_REF_PATH}/stdcell_lvt/db_nldm "

set TARGET_LIBRARY_FILES    " \
${DESIGN_REF_PATH}/stdcell_hvt/db_nldm/saed14hvt_tt0p8v125c.db \
${DESIGN_REF_PATH}/stdcell_lvt/db_nldm/saed14lvt_tt0p8v125c.db \
${DESIGN_REF_PATH}/stdcell_rvt/db_nldm/saed14rvt_tt0p8v125c.db"

set LINK_LIBRARY_FILES   "* \
${DESIGN_REF_PATH}/stdcell_hvt/db_nldm/saed14hvt_tt0p8v125c.db \
${DESIGN_REF_PATH}/stdcell_lvt/db_nldm/saed14lvt_tt0p8v125c.db \
${DESIGN_REF_PATH}/stdcell_rvt/db_nldm/saed14rvt_tt0p8v125c.db"

set_app_var search_path "$SEARCH_PATH"
set_app_var target_library "$TARGET_LIBRARY_FILES"
set_app_var link_library " $LINK_LIBRARY_FILES "

read_verilog./Simple_Processor_mvt_netlist.v
current_design simple_processor_Top
link

source ./constraints_processor.sdc
```

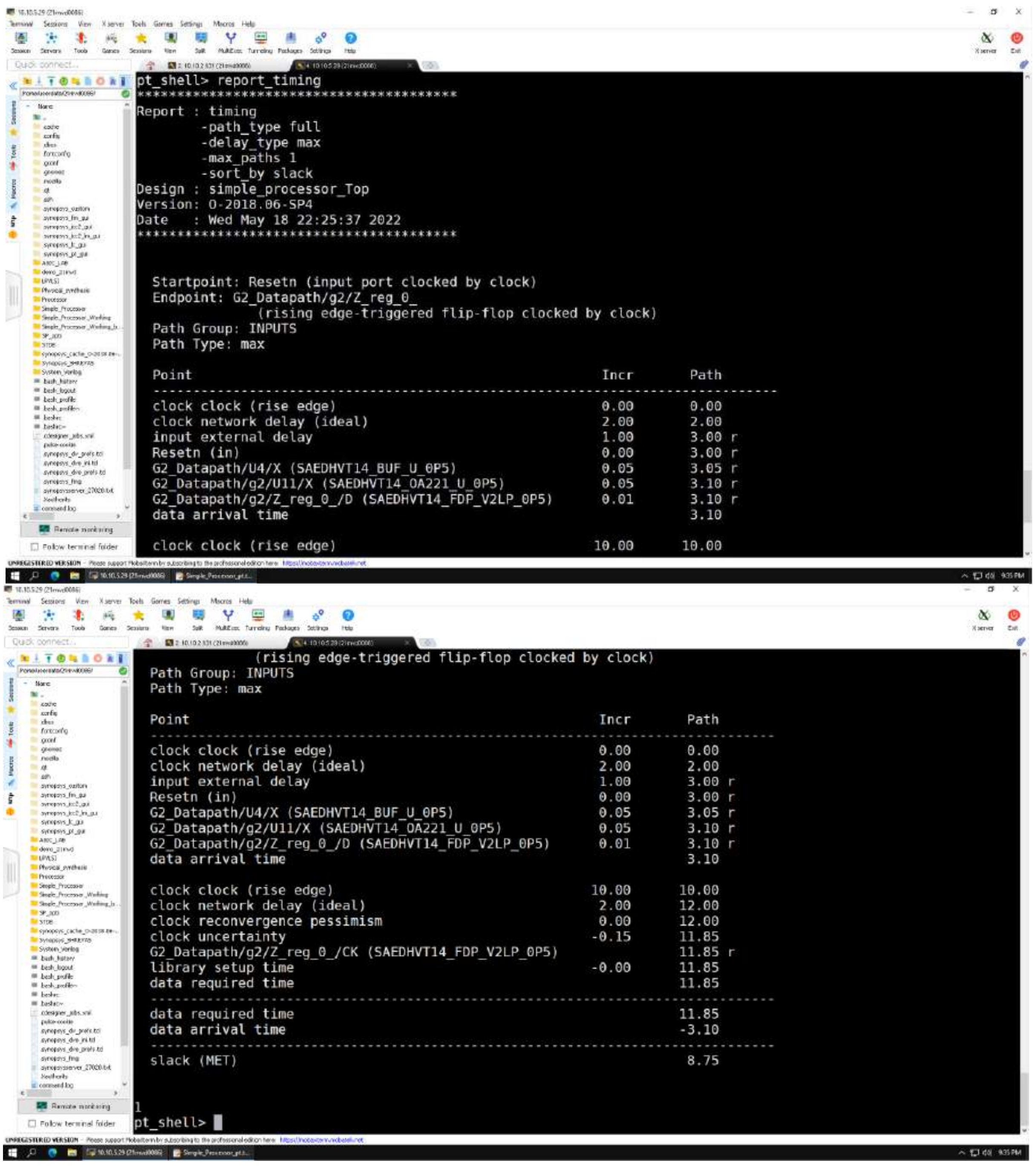
Date:18/5/2022

```
check_timing

report_timing

report_timing -delay_type min

report_analysis_coverage


report_timing -from [all_inputs] -max_paths 20 -to [all_registers -data_pins]  > reports/timing.rpt

report_timing -from [all_register -clock_pins] -max_paths 20 -to [all_registers -data_pins] > reports/su.timing.rpt

report_timing -from [all_register -clock_pins] -max_paths 20 -to [all_registers -data_pins] -delay_type min > reports/hold.timing.rpt

report_timing -from [all_registers -clock_pins] -max_paths 20 -to [all_outputs] >> reports/timing.rpt

report_timing -from [all_inputs] -to [all_outputs] -max_paths 20 >> reports/timing.rpt

report_timing -from [all_registers -clock_pins] -to [all_registers -data_pins] -delay_type max  >> reports/timing.rpt

report_timing -from [all_registers -clock_pins] -to [all_registers -data_pins] -delay_type min  >> reports/timing.rpt

report_timing -transition_time -capacitance -nets -input_pins -from [all_registers -clock_pins] -to [all_registers -data_pins]  > reports/timing.tran.cap.rpt
```

Date:18/5/2022

Screenshots:

1. Report Timing

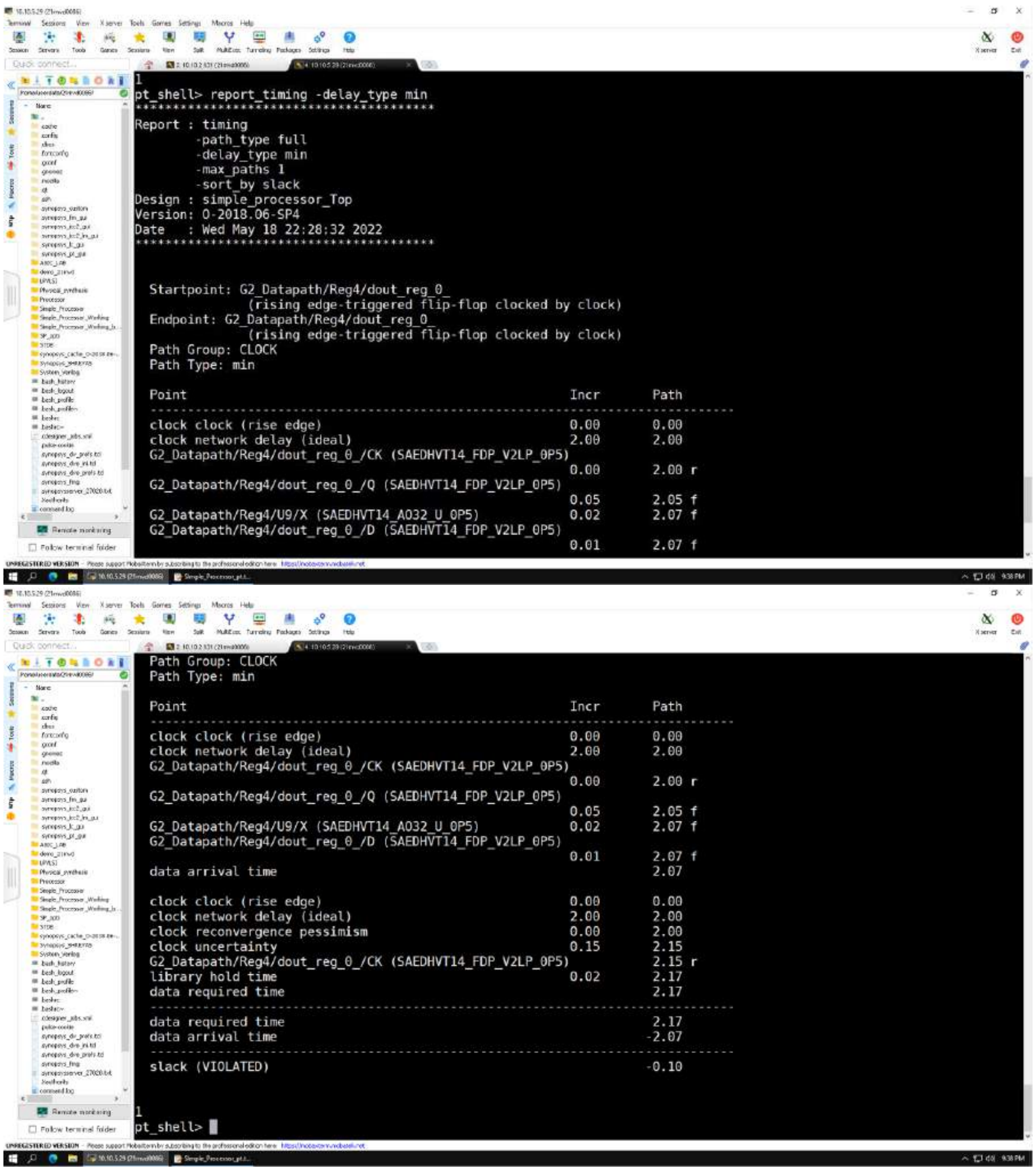


```
pt_shell> report_timing
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -sort_by slack
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Wed May 18 22:25:37 2022
****************************************


  Startpoint: Resetn (input port clocked by clock)
  Endpoint: G2_Datapath/g2/Z_reg_0_
               (rising edge-triggered flip-flop clocked by clock)
  Path Group: INPUTS
  Path Type: max

  Point                                                   Incr       Path
  ---------------------------------------------------------------------------
  clock clock (rise edge)                                 0.00       0.00
  clock network delay (ideal)                             2.00       2.00
  input external delay                                    1.00       3.00 r
  Resetn (in)                                             0.00       3.00 r
  G2_Datapath/U4/X (SAEDHVT14_BUF_U_0P5)                  0.05       3.05 r
  G2_Datapath/g2/U11/X (SAEDHVT14_OA221_U_0P5)            0.05       3.10 r
  G2_Datapath/g2/Z_reg_0_/D (SAEDHVT14_FDP_V2LP_0P5)      0.01       3.10 r
  data arrival time                                                  3.10

  clock clock (rise edge)                                10.00      10.00
```

```
               (rising edge-triggered flip-flop clocked by clock)
  Path Group: INPUTS
  Path Type: max

  Point                                                   Incr       Path
  ---------------------------------------------------------------------------
  clock clock (rise edge)                                 0.00       0.00
  clock network delay (ideal)                             2.00       2.00
  input external delay                                    1.00       3.00 r
  Resetn (in)                                             0.00       3.00 r
  G2_Datapath/U4/X (SAEDHVT14_BUF_U_0P5)                  0.05       3.05 r
  G2_Datapath/g2/U11/X (SAEDHVT14_OA221_U_0P5)            0.05       3.10 r
  G2_Datapath/g2/Z_reg_0_/D (SAEDHVT14_FDP_V2LP_0P5)      0.01       3.10 r
  data arrival time                                                  3.10

  clock clock (rise edge)                                10.00      10.00
  clock network delay (ideal)                             2.00      12.00
  clock reconvergence pessimism                           0.00      12.00
  clock uncertainty                                      -0.15      11.85
  G2_Datapath/g2/Z_reg_0_/CK (SAEDHVT14_FDP_V2LP_0P5)               11.85 r
  library setup time                                     -0.00      11.85
  data required time                                                11.85
  ---------------------------------------------------------------------------
  data required time                                                11.85
  data arrival time                                                 -3.10
  ---------------------------------------------------------------------------
  slack (MET)                                                        8.75


1
pt_shell> 
```

Figure 3.1: Report timing log showing output.

2. report timing -delay type min

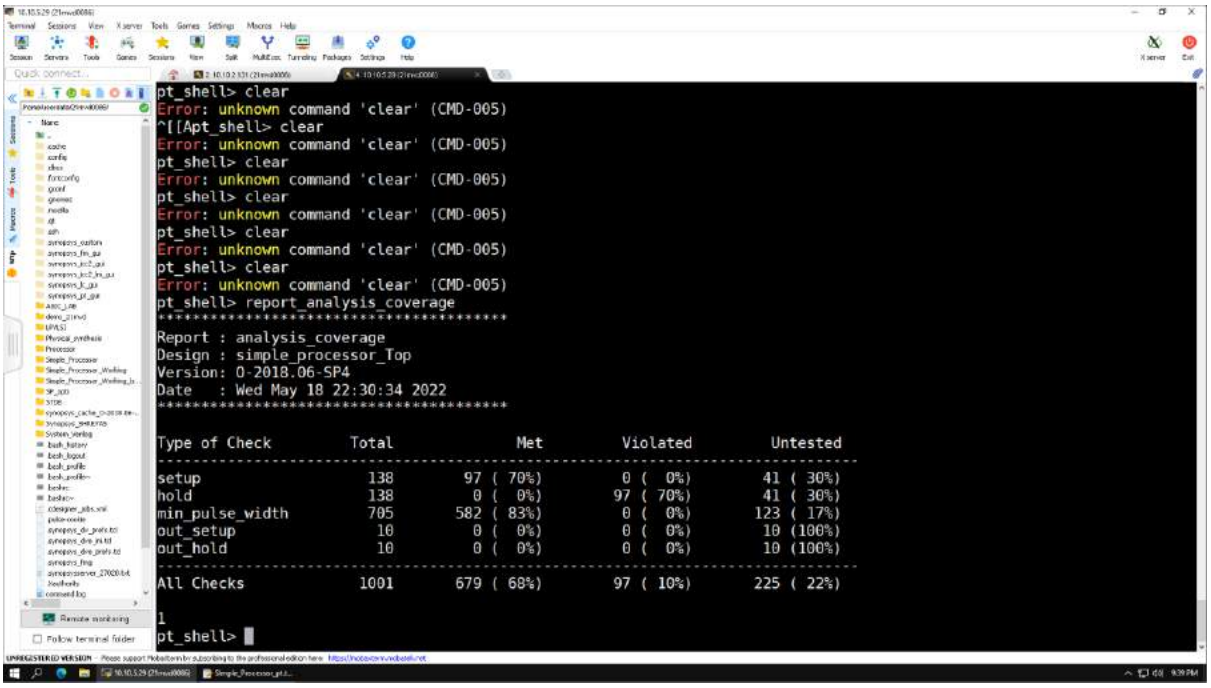


```
1
pt_shell> report_timing -delay_type min
****************************************
Report : timing
        -path_type full
        -delay_type min
        -max_paths 1
        -sort_by slack
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Wed May 18 22:28:32 2022
****************************************


  Startpoint: G2_Datapath/Reg4/dout_reg_0_
              (rising edge-triggered flip-flop clocked by clock)
  Endpoint: G2_Datapath/Reg4/dout_reg_0_
              (rising edge-triggered flip-flop clocked by clock)
  Path Group: CLOCK
  Path Type: min

  Point                                                        Incr       Path
  ------------------------------------------------------------------------------
  clock clock (rise edge)                                      0.00       0.00
  clock network delay (ideal)                                  2.00       2.00
  G2_Datapath/Reg4/dout_reg_0_/CK (SAEDHVT14_FDP_V2LP_0P5)
                                                               0.00       2.00 r
  G2_Datapath/Reg4/dout_reg_0_/Q (SAEDHVT14_FDP_V2LP_0P5)
                                                               0.05       2.05 f
  G2_Datapath/Reg4/U9/X (SAEDHVT14_AO32_U_0P5)                 0.02       2.07 f
  G2_Datapath/Reg4/dout_reg_0_/D (SAEDHVT14_FDP_V2LP_0P5)
                                                               0.01       2.07 f
```

```
  Path Group: CLOCK
  Path Type: min

  Point                                                        Incr       Path
  ------------------------------------------------------------------------------
  clock clock (rise edge)                                      0.00       0.00
  clock network delay (ideal)                                  2.00       2.00
  G2_Datapath/Reg4/dout_reg_0_/CK (SAEDHVT14_FDP_V2LP_0P5)
                                                               0.00       2.00 r
  G2_Datapath/Reg4/dout_reg_0_/Q (SAEDHVT14_FDP_V2LP_0P5)
                                                               0.05       2.05 f
  G2_Datapath/Reg4/U9/X (SAEDHVT14_AO32_U_0P5)                 0.02       2.07 f
  G2_Datapath/Reg4/dout_reg_0_/D (SAEDHVT14_FDP_V2LP_0P5)
                                                               0.01       2.07 f
  data arrival time                                                       2.07

  clock clock (rise edge)                                      0.00       0.00
  clock network delay (ideal)                                  2.00       2.00
  clock reconvergence pessimism                                0.00       2.00
  clock uncertainty                                            0.15       2.15
  G2_Datapath/Reg4/dout_reg_0_/CK (SAEDHVT14_FDP_V2LP_0P5)                2.15 r
  library hold time                                            0.02       2.17
  data required time                                                      2.17
  ------------------------------------------------------------------------------
  data required time                                                      2.17
  data arrival time                                                      -2.07
  ------------------------------------------------------------------------------
  slack (VIOLATED)                                                       -0.10


1
pt_shell>
```

Figure 3.2: Report timing -delay type log showing output.

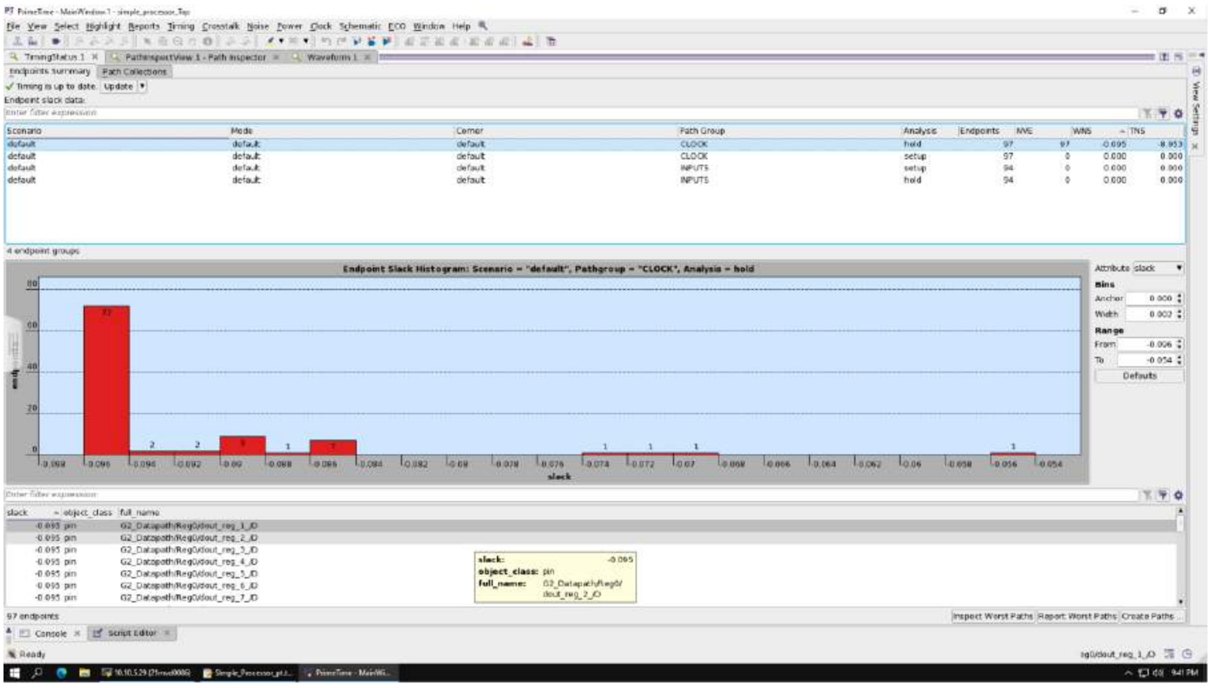
3. report_analysis_coverage



```
pt_shell> clear
Error: unknown command 'clear' (CMD-005)
^[[Apt_shell> clear
Error: unknown command 'clear' (CMD-005)
pt_shell> clear
Error: unknown command 'clear' (CMD-005)
pt_shell> clear
Error: unknown command 'clear' (CMD-005)
pt_shell> clear
Error: unknown command 'clear' (CMD-005)
pt_shell> clear
Error: unknown command 'clear' (CMD-005)
pt_shell> report_analysis_coverage
****************************************
Report : analysis_coverage
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Wed May 18 22:30:34 2022
****************************************

Type of Check         Total              Met         Violated         Untested
--------------------------------------------------------------------------------
setup                   138         97 ( 70%)        0 (  0%)        41 ( 30%)
hold                    138          0 (  0%)       97 ( 70%)        41 ( 30%)
min_pulse_width         705        582 ( 83%)        0 (  0%)       123 ( 17%)
out_setup                10          0 (  0%)        0 (  0%)        10 (100%)
out_hold                 10          0 (  0%)        0 (  0%)        10 (100%)
--------------------------------------------------------------------------------
All Checks             1001        679 ( 68%)       97 ( 10%)       225 ( 22%)

1
pt_shell> 
```

Figure 3.3: report_analysis_coverage type log showing output.

Prime Time GUI Screenshots



Endpoints summary

Timing is up to date. Update

Endpoint slack data:

Scenario	Mode	Corner	Path Group	Analysis	Endpoints	NVE	WNS	TNS
default	default	default	CLOCK	hold	97	97	-0.095	-8.953
default	default	default	CLOCK	setup	97	0	0.000	0.000
default	default	default	INPUTS	setup	94	0	0.000	0.000
default	default	default	INPUTS	hold	94	0	0.000	0.000

4 endpoint groups

Endpoint Slack Histogram: Scenario = "default", Pathgroup = "CLOCK", Analysis = hold

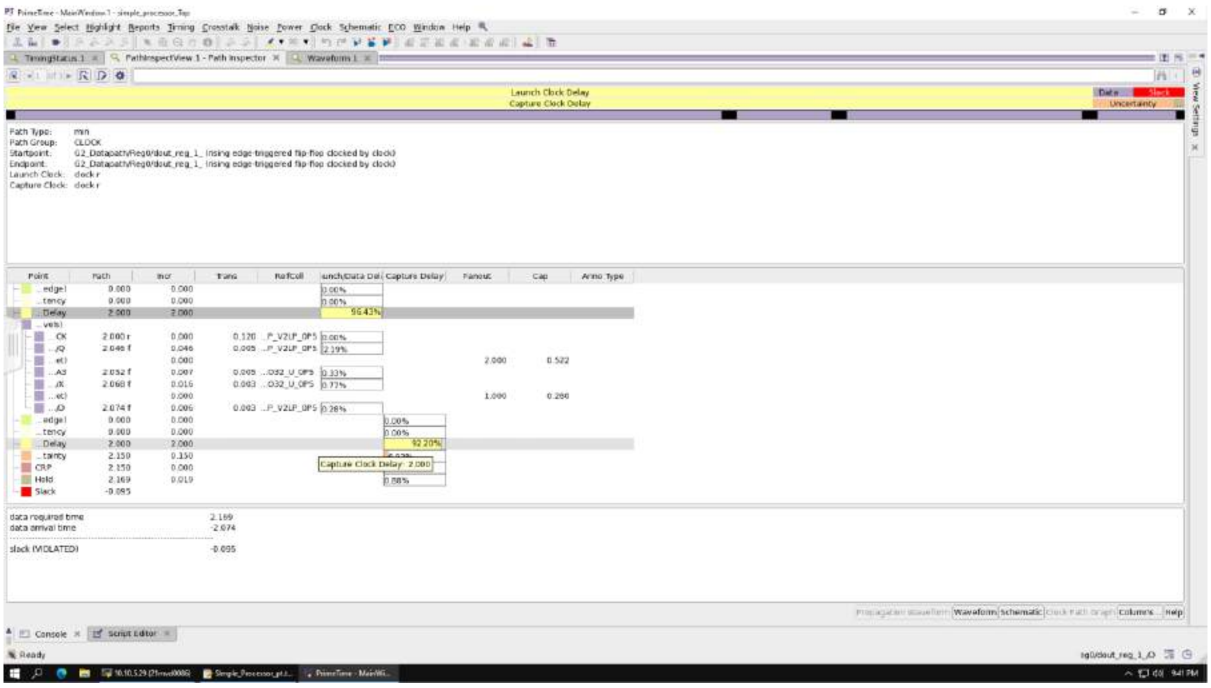
slack	object_class	full_name
-0.095	pin	G2_Datapath/Reg0/dout_reg_1_/D
-0.095	pin	G2_Datapath/Reg0/dout_reg_2_/D
-0.095	pin	G2_Datapath/Reg0/dout_reg_3_/D
-0.095	pin	G2_Datapath/Reg0/dout_reg_4_/D
-0.095	pin	G2_Datapath/Reg0/dout_reg_5_/D
-0.095	pin	G2_Datapath/Reg0/dout_reg_6_/D
-0.095	pin	G2_Datapath/Reg0/dout_reg_7_/D

slack: -0.095
object_class: pin
full_name: G2_Datapath/Reg0/dout_reg_2_/D

97 endpoints

Figure 3.4: The number of paths violating Hold.

Date:18/5/2022



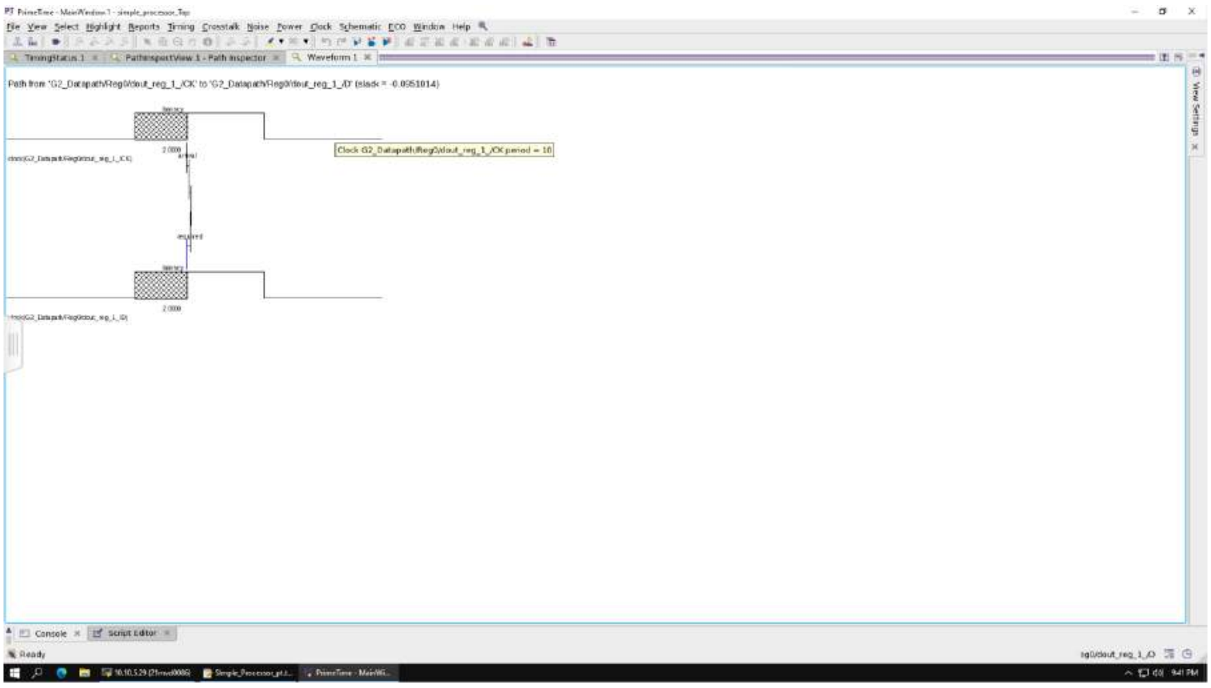
```
Path Type:      min
Path Group:     CLOCK
Startpoint:     G2_Datapath/Reg0/dout_reg_1_ (rising edge-triggered flip-flop clocked by clock)
Endpoint:       G2_Datapath/Reg0/dout_reg_1_ (rising edge-triggered flip-flop clocked by clock)
Launch Clock:   clock r
Capture Clock:  clock r
```

Point	Path	Incr	Trans	RefCell	unch/Data Del	Capture Delay	Fanout	Cap	Anno Type
...edge)	0.000	0.000			0.00%				
...tency	0.000	0.000			0.00%				
...Delay	2.000	2.000			96.43%				
...vels)									
...CK	2.000 r	0.000	0.120	...P_V2LP_0P5	0.00%				
.../Q	2.046 f	0.046	0.005	...P_V2LP_0P5	2.19%				
...et)		0.000					2.000	0.522	
...A3	2.052 f	0.007	0.005	...O32_U_0P5	0.33%				
.../X	2.068 f	0.016	0.003	...O32_U_0P5	0.77%				
...et)		0.000					1.000	0.280	
.../D	2.074 f	0.006	0.003	...P_V2LP_0P5	0.28%				
...edge)	0.000	0.000				0.00%			
...tency	0.000	0.000				0.00%			
...Delay	2.000	2.000				92.20%			
...tainty	2.150	0.150				[illegible]			
CRP	2.150	0.000							
Hold	2.169	0.019				0.88%			
Slack	-0.095								

Capture Clock Delay: 2.000

```
data required time          2.169
data arrival time          -2.074
---------------------------------
slack (VIOLATED)           -0.095
```

Figure 3.5: The detailed analysis of worst path showing the timing values.



Path from 'G2_Datapath/Reg0/dout_reg_1_/CK' to 'G2_Datapath/Reg0/dout_reg_1_/D' (slack = -0.0951014)

Clock G2_Datapath/Reg0/dout_reg_1_/CK period = 10

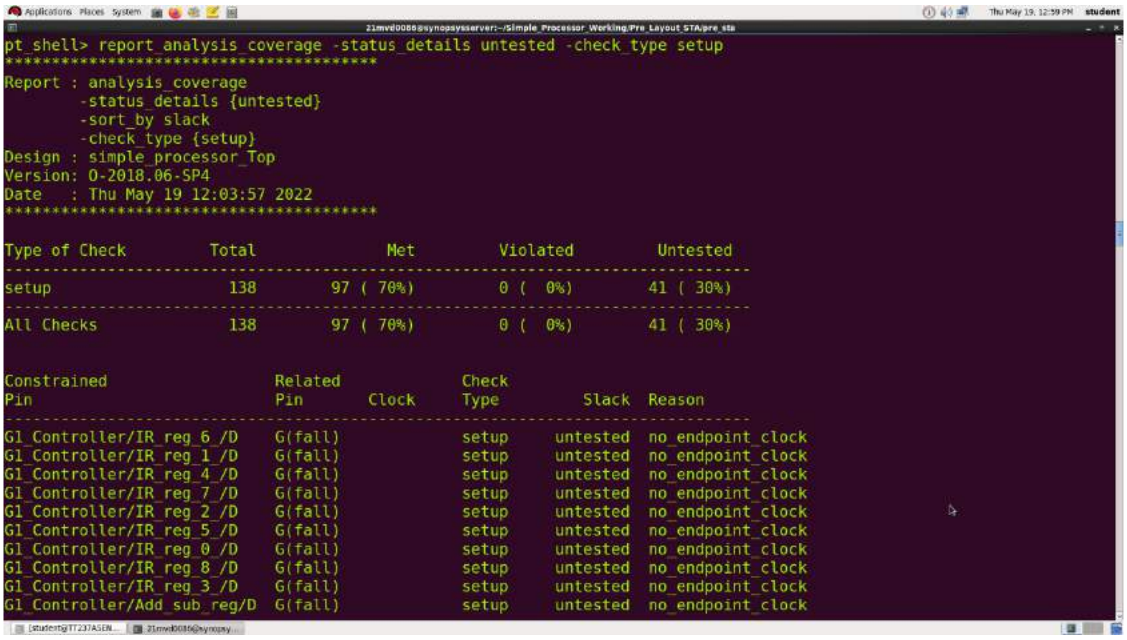
Figure 3.6: The hold violation in waveform window

Date:18/5/2022

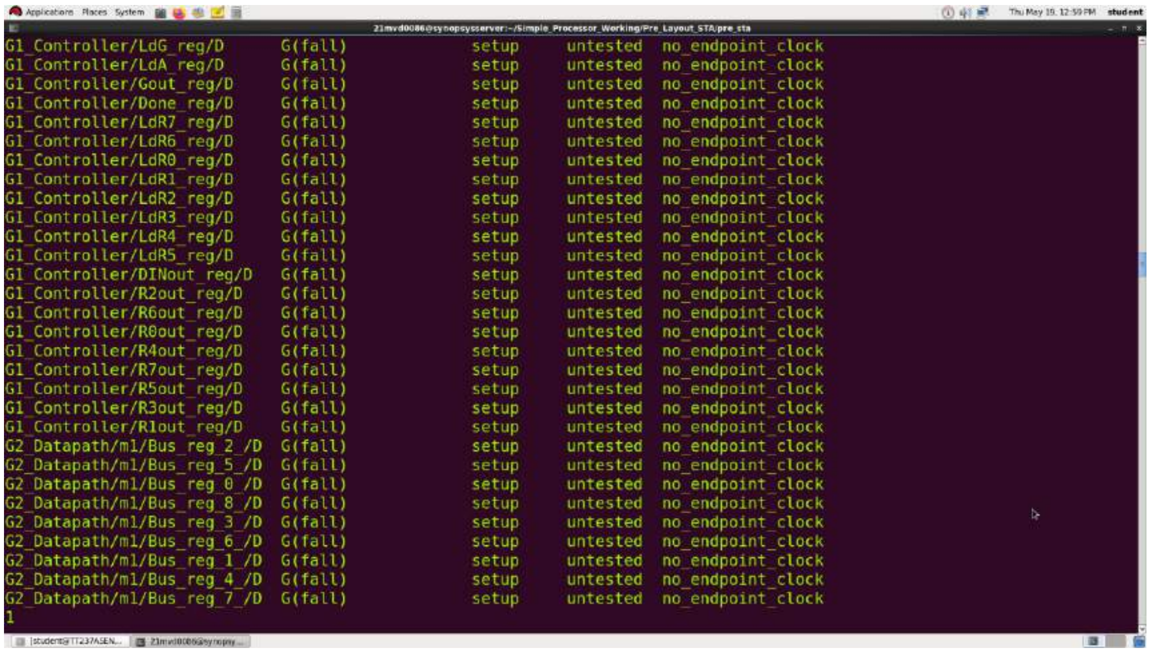
The Commands used for Untested path and reasons why it is not untested?

Commands:

1. report_analysis_coverage -status_details untested -check_type setup
2. report_analysis_coverage -status_details untested -check_type hold
3. report_analysis_coverage -status_details untested -check_type out_setup
4. report_analysis_coverage -status_details untested -check_type out_hold
5. report_analysis_coverage -status_details untested -check_type min_pulse_width

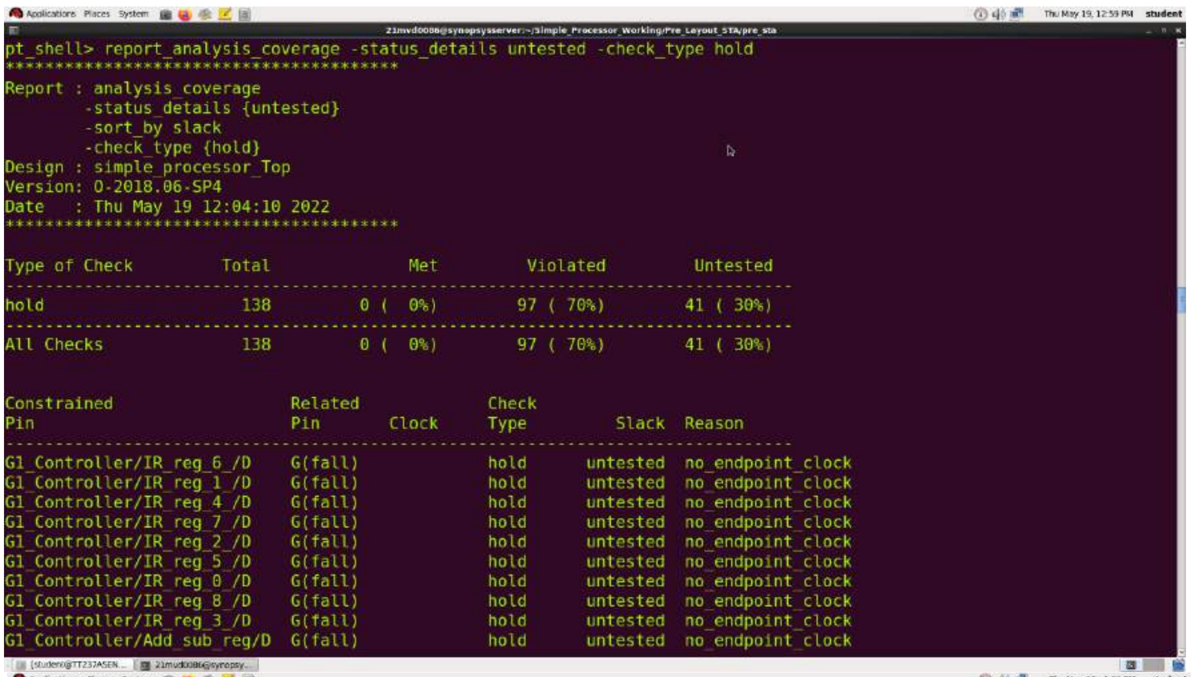


```
pt_shell> report_analysis_coverage -status_details untested -check_type setup
****************************************
Report : analysis_coverage
        -status_details {untested}
        -sort_by slack
        -check_type {setup}
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Thu May 19 12:03:57 2022
****************************************

Type of Check         Total              Met         Violated         Untested
------------------------------------------------------------------------------
setup                   138        97 ( 70%)          0 (  0%)        41 ( 30%)
------------------------------------------------------------------------------
All Checks              138        97 ( 70%)          0 (  0%)        41 ( 30%)


Constrained                Related             Check
Pin                        Pin       Clock     Type          Slack  Reason
------------------------------------------------------------------------------
G1_Controller/IR_reg_6_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_1_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_4_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_7_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_2_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_5_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_0_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_8_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/IR_reg_3_/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/Add_sub_reg/D G(fall)            setup     untested  no_endpoint_clock
```

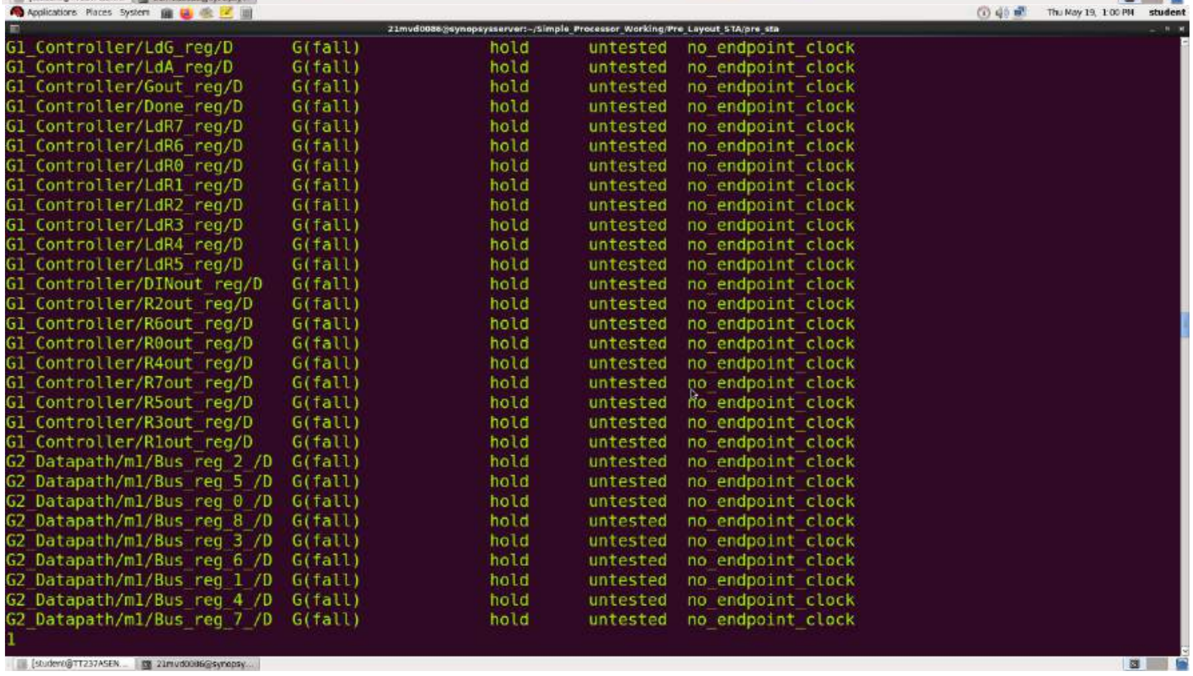


```
G1_Controller/LdG_reg/D    G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdA_reg/D    G(fall)             setup     untested  no_endpoint_clock
G1_Controller/Gout_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/Done_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR7_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR6_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR0_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR1_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR2_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR3_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR4_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/LdR5_reg/D   G(fall)             setup     untested  no_endpoint_clock
G1_Controller/DINout_reg/D G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R2out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R6out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R0out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R4out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R7out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R5out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R3out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G1_Controller/R1out_reg/D  G(fall)             setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_2_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_5_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_0_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_8_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_3_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_6_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_1_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_4_/D G(fall)            setup     untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_7_/D G(fall)            setup     untested  no_endpoint_clock
1
```

Figure 3.7: The report analysis coverage details for untested path for setup.



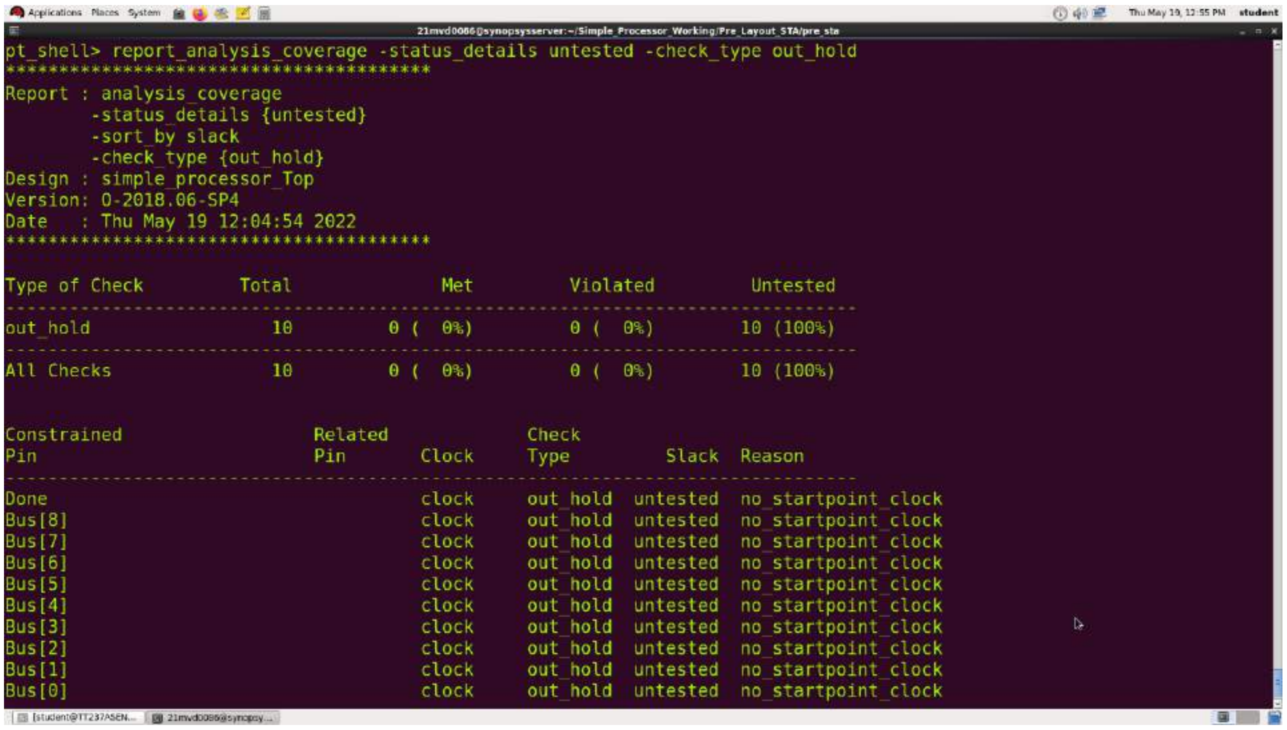
```
pt_shell> report_analysis_coverage -status_details untested -check_type hold
****************************************
Report : analysis_coverage
        -status_details {untested}
        -sort_by slack
        -check_type {hold}
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Thu May 19 12:04:10 2022
****************************************

Type of Check         Total              Met         Violated         Untested
--------------------------------------------------------------------------------
hold                    138         0 (  0%)        97 ( 70%)        41 ( 30%)
--------------------------------------------------------------------------------
All Checks              138         0 (  0%)        97 ( 70%)        41 ( 30%)


Constrained                   Related             Check
Pin                           Pin       Clock     Type          Slack  Reason
--------------------------------------------------------------------------------
G1_Controller/IR_reg_6_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_1_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_4_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_7_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_2_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_5_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_0_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_8_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/IR_reg_3_/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/Add_sub_reg/D   G(fall)             hold      untested  no_endpoint_clock
```



```
G1_Controller/LdG_reg/D       G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdA_reg/D       G(fall)             hold      untested  no_endpoint_clock
G1_Controller/Gout_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/Done_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR7_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR6_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR0_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR1_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR2_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR3_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR4_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/LdR5_reg/D      G(fall)             hold      untested  no_endpoint_clock
G1_Controller/DINout_reg/D    G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R2out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R6out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R0out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R4out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R7out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R5out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R3out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G1_Controller/R1out_reg/D     G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_2_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_5_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_0_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_8_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_3_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_6_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_1_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_4_/D   G(fall)             hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_7_/D   G(fall)             hold      untested  no_endpoint_clock
1
```

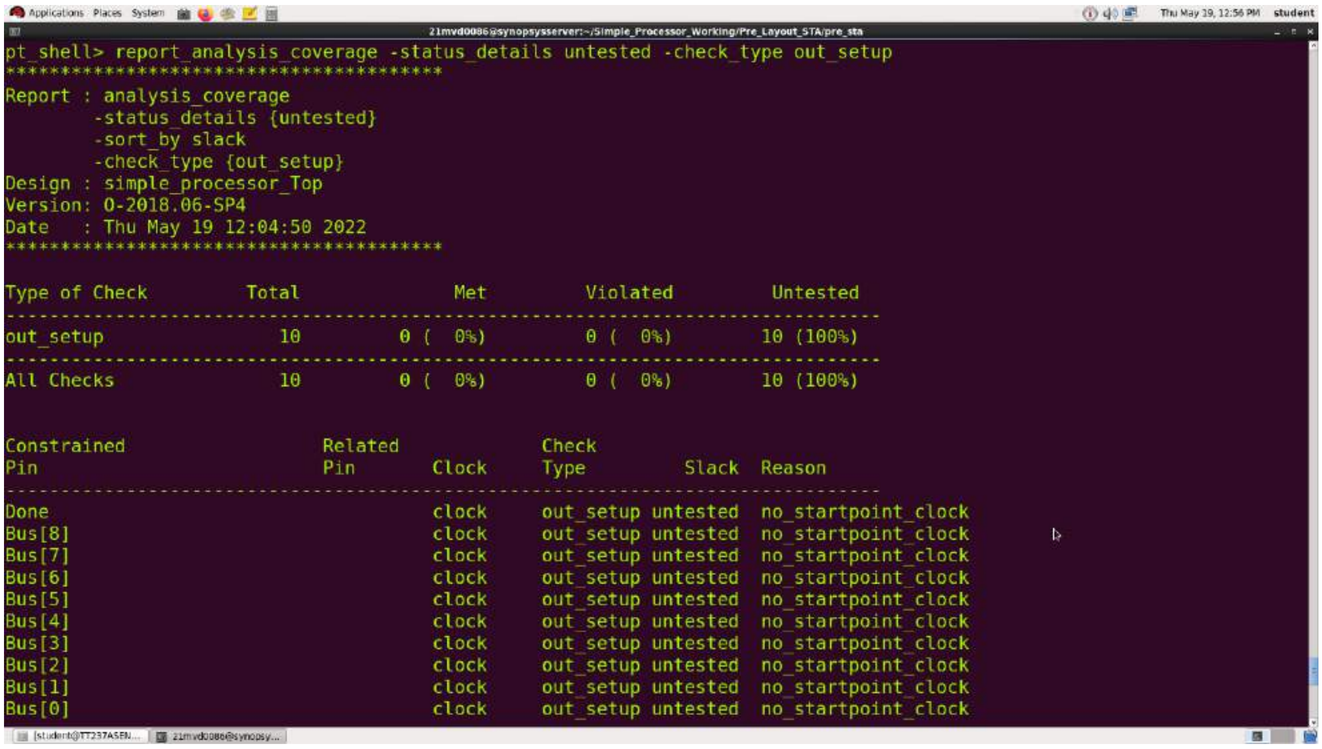
Figure 3.8: The report analysis coverage details for untested path for hold.

Date:18/5/2022



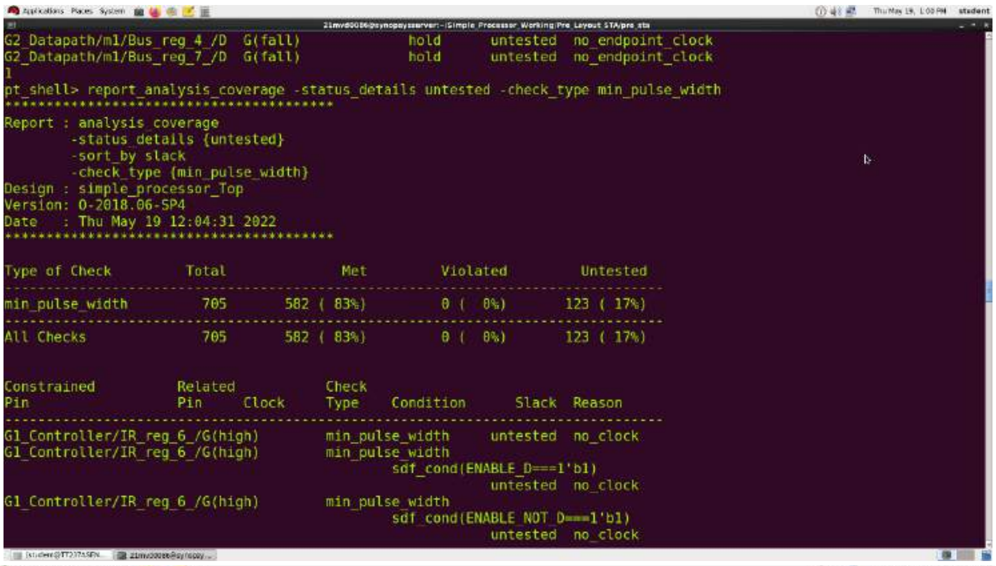
```
pt_shell> report_analysis_coverage -status_details untested -check_type out_hold
****************************************
Report : analysis_coverage
        -status_details {untested}
        -sort_by slack
        -check_type {out_hold}
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Thu May 19 12:04:54 2022
****************************************

Type of Check         Total              Met         Violated         Untested
--------------------------------------------------------------------------------
out_hold                 10          0 (  0%)          0 (  0%)        10 (100%)
--------------------------------------------------------------------------------
All Checks               10          0 (  0%)          0 (  0%)        10 (100%)


Constrained                    Related             Check
Pin                            Pin       Clock     Type         Slack  Reason
--------------------------------------------------------------------------------
Done                                     clock     out_hold  untested  no_startpoint_clock
Bus[8]                                   clock     out_hold  untested  no_startpoint_clock
Bus[7]                                   clock     out_hold  untested  no_startpoint_clock
Bus[6]                                   clock     out_hold  untested  no_startpoint_clock
Bus[5]                                   clock     out_hold  untested  no_startpoint_clock
Bus[4]                                   clock     out_hold  untested  no_startpoint_clock
Bus[3]                                   clock     out_hold  untested  no_startpoint_clock
Bus[2]                                   clock     out_hold  untested  no_startpoint_clock
Bus[1]                                   clock     out_hold  untested  no_startpoint_clock
Bus[0]                                   clock     out_hold  untested  no_startpoint_clock
```

Figure 3.9: The report analysis coverage details for untested path for out_hold.

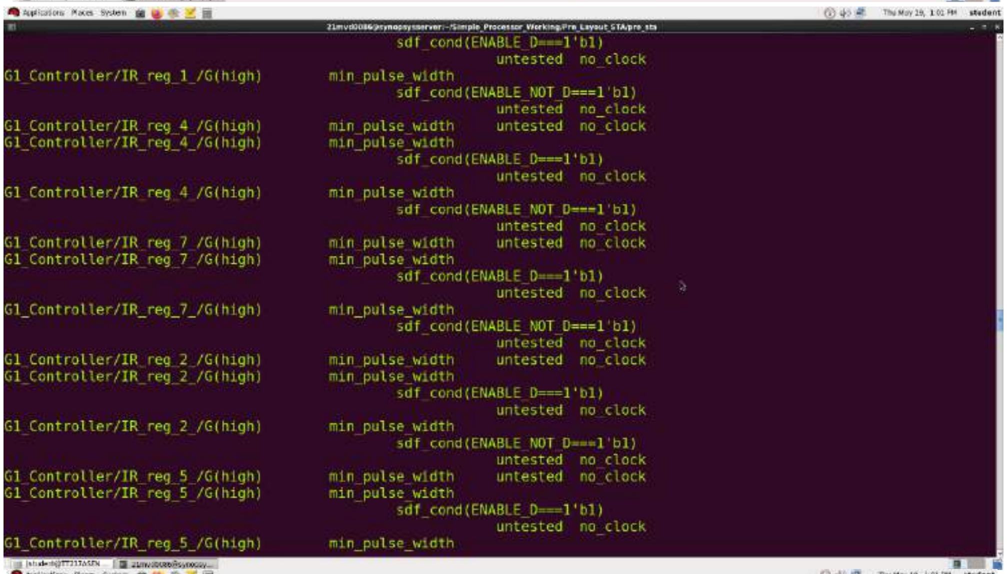


```
pt_shell> report_analysis_coverage -status_details untested -check_type out_setup
****************************************
Report : analysis_coverage
        -status_details {untested}
        -sort_by slack
        -check_type {out_setup}
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Thu May 19 12:04:50 2022
****************************************

Type of Check         Total              Met         Violated         Untested
--------------------------------------------------------------------------------
out_setup                10          0 (  0%)          0 (  0%)        10 (100%)
--------------------------------------------------------------------------------
All Checks               10          0 (  0%)          0 (  0%)        10 (100%)


Constrained                    Related             Check
Pin                            Pin       Clock     Type         Slack  Reason
--------------------------------------------------------------------------------
Done                                     clock     out_setup untested  no_startpoint_clock
Bus[8]                                   clock     out_setup untested  no_startpoint_clock
Bus[7]                                   clock     out_setup untested  no_startpoint_clock
Bus[6]                                   clock     out_setup untested  no_startpoint_clock
Bus[5]                                   clock     out_setup untested  no_startpoint_clock
Bus[4]                                   clock     out_setup untested  no_startpoint_clock
Bus[3]                                   clock     out_setup untested  no_startpoint_clock
Bus[2]                                   clock     out_setup untested  no_startpoint_clock
Bus[1]                                   clock     out_setup untested  no_startpoint_clock
Bus[0]                                   clock     out_setup untested  no_startpoint_clock
```

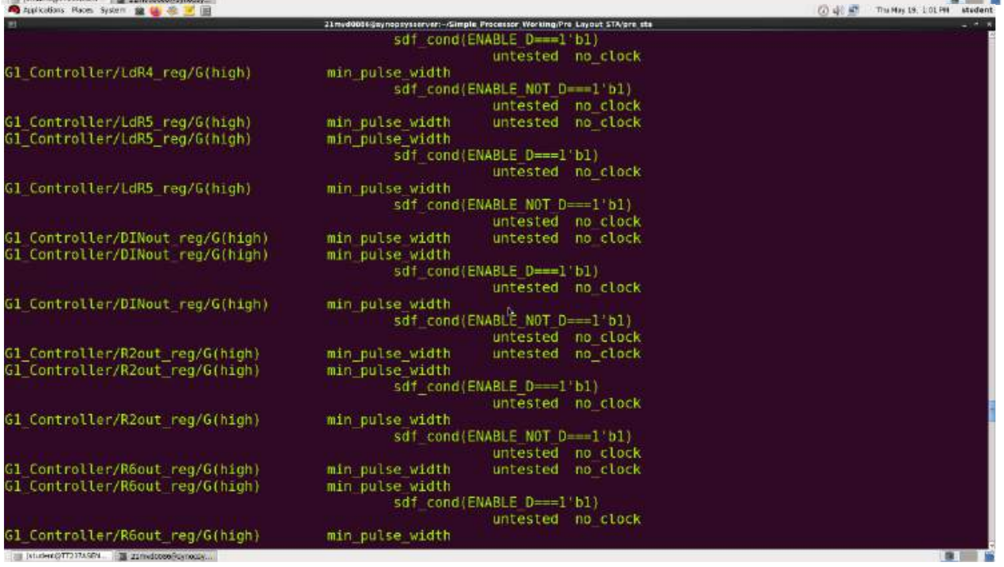
Figure 3.10: The report analysis coverage details for untested path for out_setup.



```
G2_Datapath/m1/Bus_reg_4_/D  G(fall)              hold      untested  no_endpoint_clock
G2_Datapath/m1/Bus_reg_7_/D  G(fall)              hold      untested  no_endpoint_clock
1
pt_shell> report_analysis_coverage -status_details untested -check_type min_pulse_width
****************************************
Report : analysis_coverage
        -status_details {untested}
        -sort_by slack
        -check_type {min_pulse_width}
Design : simple_processor_Top
Version: O-2018.06-SP4
Date   : Thu May 19 12:04:31 2022
****************************************

Type of Check         Total              Met         Violated         Untested
--------------------------------------------------------------------------------
min_pulse_width         705       582 ( 83%)         0 (  0%)       123 ( 17%)
--------------------------------------------------------------------------------
All Checks              705       582 ( 83%)         0 (  0%)       123 ( 17%)


Constrained          Related              Check
Pin                  Pin      Clock       Type     Condition      Slack  Reason
--------------------------------------------------------------------------------
G1_Controller/IR_reg_6_/G(high)           min_pulse_width     untested  no_clock
G1_Controller/IR_reg_6_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_6_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock
```



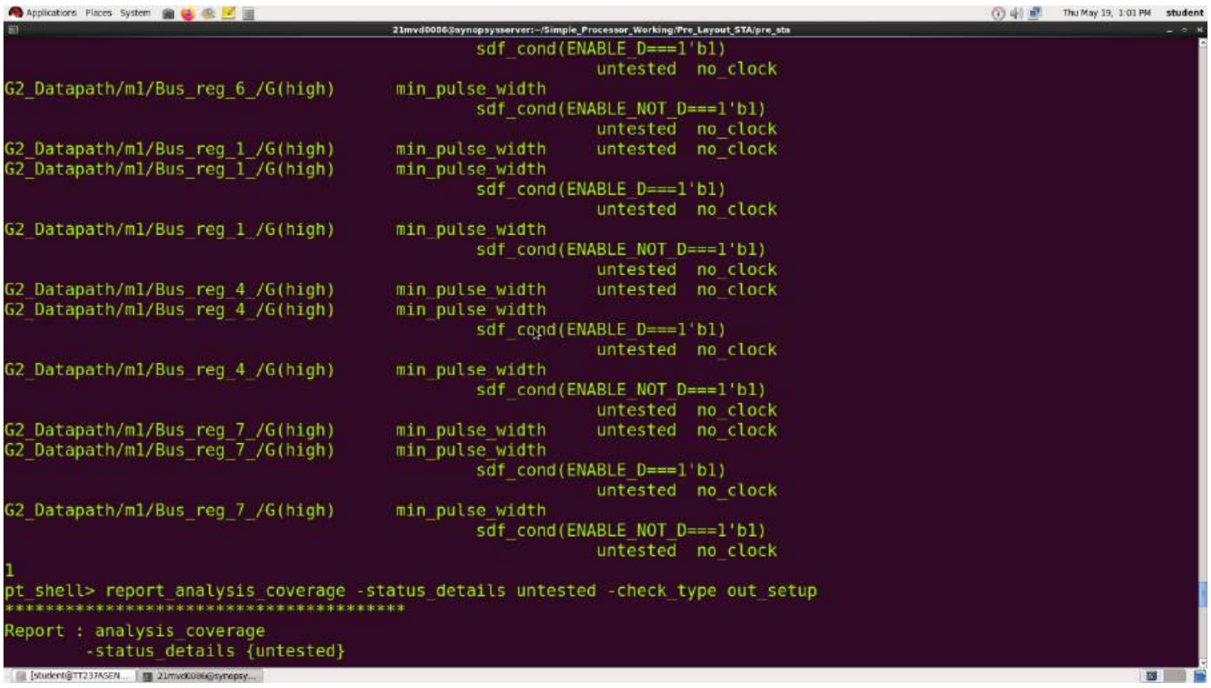
```
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_1_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_4_/G(high)           min_pulse_width     untested  no_clock
G1_Controller/IR_reg_4_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_4_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_7_/G(high)           min_pulse_width     untested  no_clock
G1_Controller/IR_reg_7_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_7_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_2_/G(high)           min_pulse_width     untested  no_clock
G1_Controller/IR_reg_2_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_2_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_5_/G(high)           min_pulse_width     untested  no_clock
G1_Controller/IR_reg_5_/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/IR_reg_5_/G(high)           min_pulse_width
```



```
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/LdR4_reg/G(high)            min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/LdR5_reg/G(high)            min_pulse_width     untested  no_clock
G1_Controller/LdR5_reg/G(high)            min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/LdR5_reg/G(high)            min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/DINout_reg/G(high)          min_pulse_width     untested  no_clock
G1_Controller/DINout_reg/G(high)          min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/DINout_reg/G(high)          min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/R2out_reg/G(high)           min_pulse_width     untested  no_clock
G1_Controller/R2out_reg/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/R2out_reg/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_NOT_D===1'b1)
                                                              untested  no_clock

G1_Controller/R6out_reg/G(high)           min_pulse_width     untested  no_clock
G1_Controller/R6out_reg/G(high)           min_pulse_width
                                                   sdf_cond(ENABLE_D===1'b1)
                                                              untested  no_clock

G1_Controller/R6out_reg/G(high)           min_pulse_width
```

Figure 3.10: The report analysis coverage details for untested path for min_pulse_width.

Date:18/5/2022



```
                                               sdf_cond(ENABLE_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_6_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_NOT_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_1_/G(high)      min_pulse_width     untested  no_clock
G2_Datapath/m1/Bus_reg_1_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_1_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_NOT_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_4_/G(high)      min_pulse_width     untested  no_clock
G2_Datapath/m1/Bus_reg_4_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_4_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_NOT_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_7_/G(high)      min_pulse_width     untested  no_clock
G2_Datapath/m1/Bus_reg_7_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_D===1'b1)
                                                           untested  no_clock
G2_Datapath/m1/Bus_reg_7_/G(high)      min_pulse_width
                                               sdf_cond(ENABLE_NOT_D===1'b1)
                                                           untested  no_clock
1
pt_shell> report_analysis_coverage -status_details untested -check_type out_setup
****************************************
Report : analysis_coverage
        -status_details {untested}
```

Figure 3.11: The report analysis coverage details for untested path for min_pulse_width.

Inference:

1. The Simple Processor 9 bit was designed using Verilog code. The simulation was done using Synopsys VCS tool.
2. The Pre-Layout STA was done using pt shell by running the pt tickle script. We do report analysis and find how many paths are violating the time. Mainly, the hold time is violating in 97 paths in our design. We get histogram plot for number of path violating, waveform display for violating the paths.