PES UNIVERSITY

UE21CS251A
DIGITAL DESIGN AND COMPUTER ORGANISATION
ASSIGNMENTS

Name: SAAD BIN KHALID

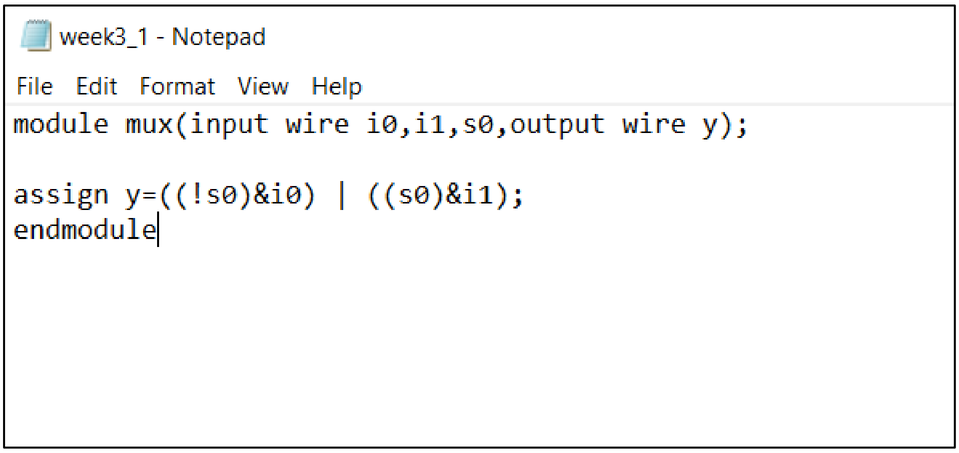
SRN: PES1UG21CS508

ROLL NO: 16

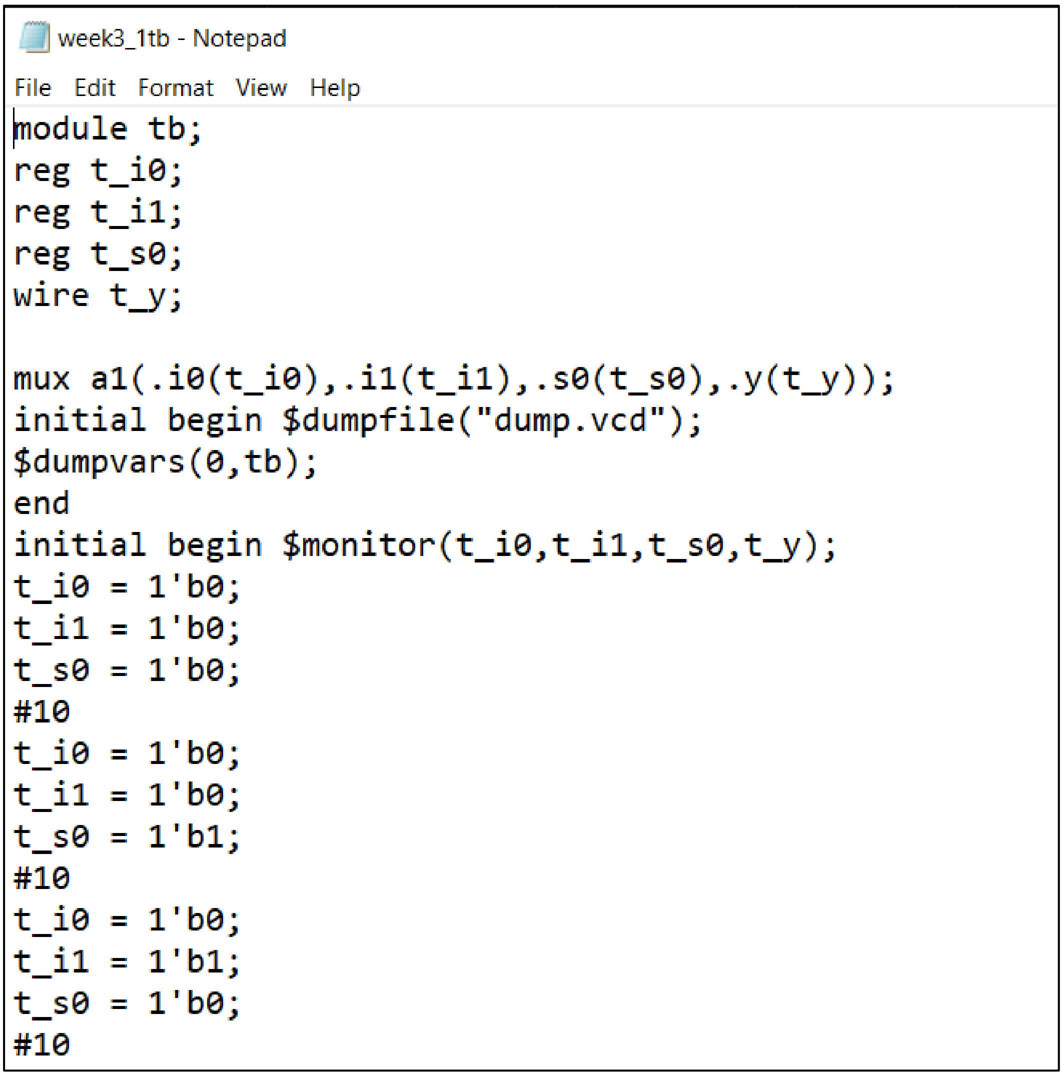
SEMESTER: III

SECTION: I

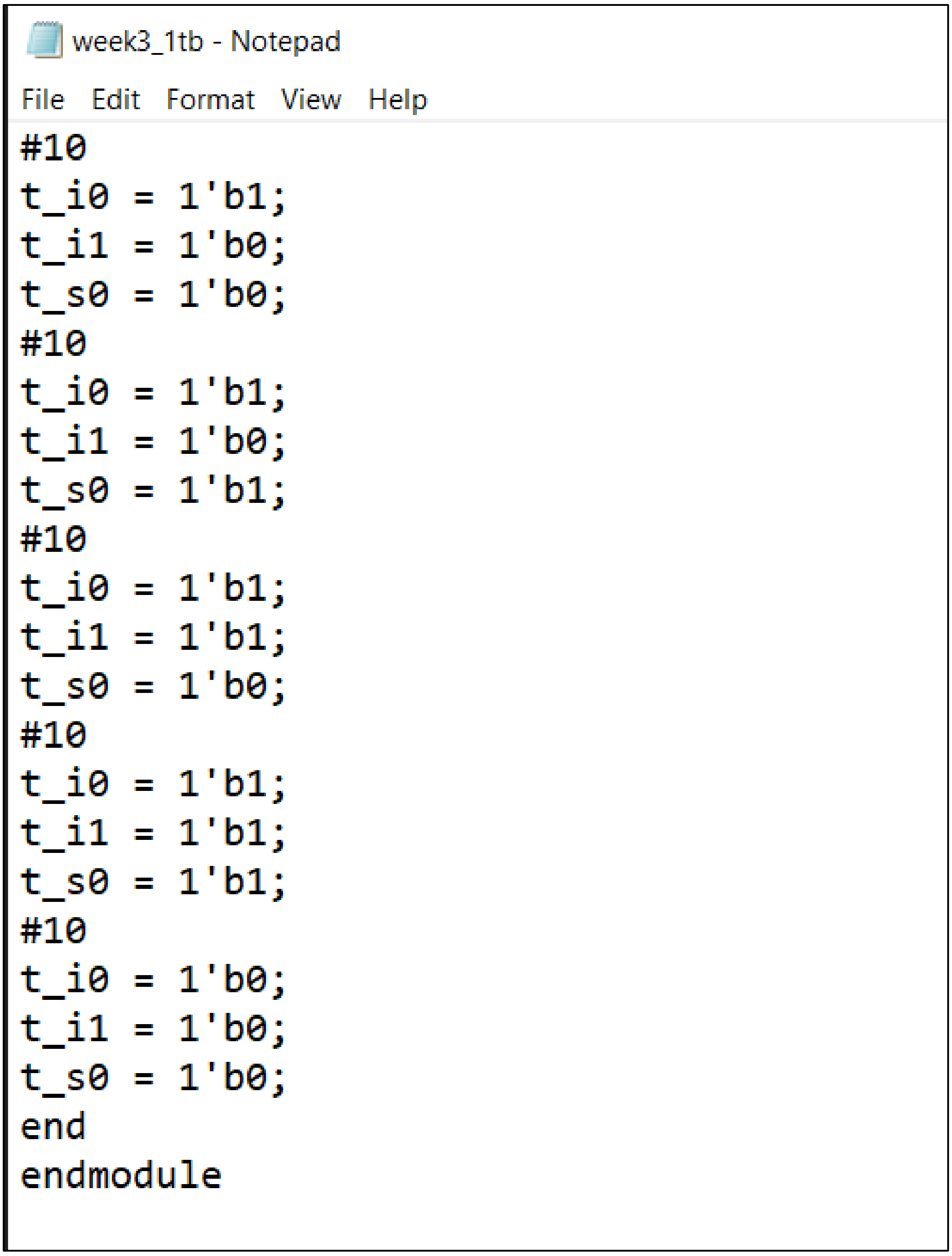
PROGRAM 1: 2:1 MUX



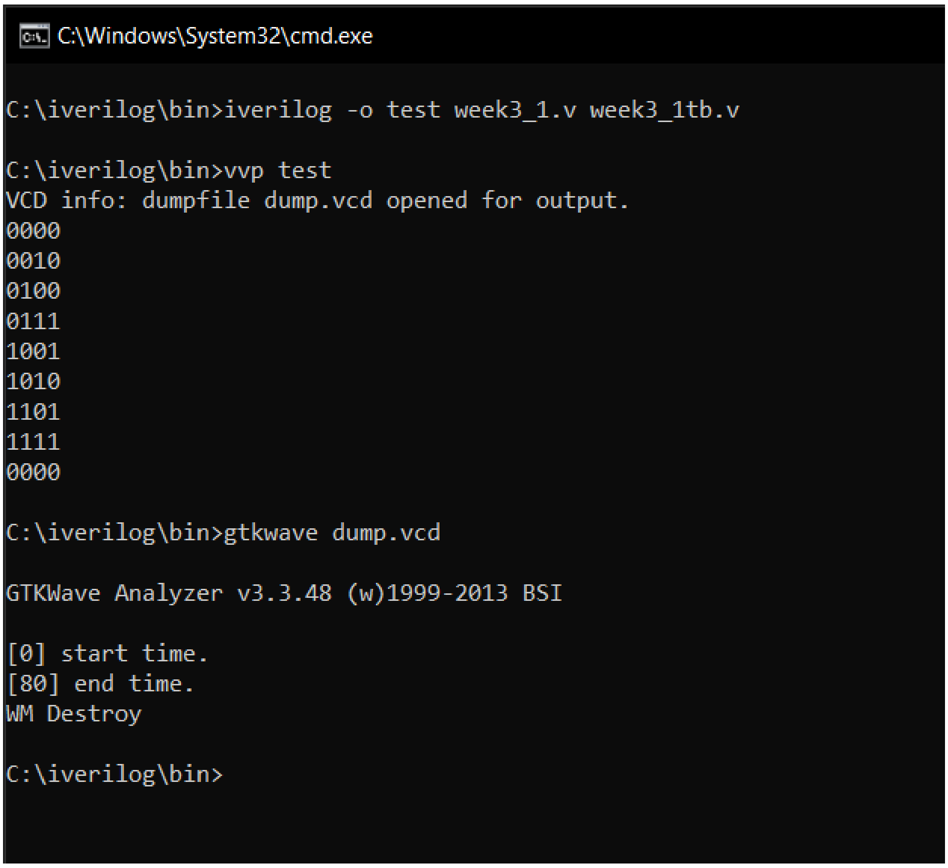
```
module mux(input wire i0,i1,s0,output wire y);

assign y=((!s0)&i0) | ((s0)&i1);
endmodule
```



```
module tb;
reg t_i0;
reg t_i1;
reg t_s0;
wire t_y;

mux a1(.i0(t_i0),.i1(t_i1),.s0(t_s0),.y(t_y));
initial begin $dumpfile("dump.vcd");
$dumpvars(0,tb);
end
initial begin $monitor(t_i0,t_i1,t_s0,t_y);
t_i0 = 1'b0;
t_i1 = 1'b0;
t_s0 = 1'b0;
#10
t_i0 = 1'b0;
t_i1 = 1'b0;
t_s0 = 1'b1;
#10
t_i0 = 1'b0;
t_i1 = 1'b1;
t_s0 = 1'b0;
#10
```

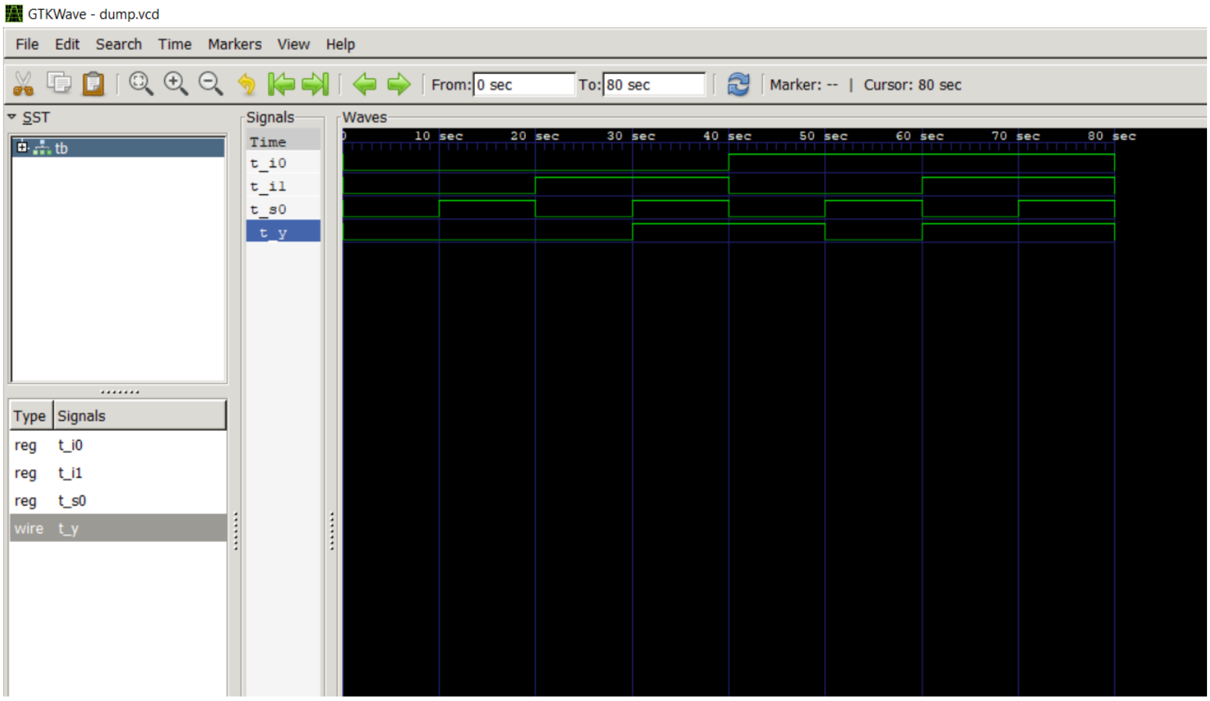


```
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_s0 = 1'b0;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_s0 = 1'b1;
#10
t_i0 = 1'b1;
t_i1 = 1'b1;
t_s0 = 1'b0;
#10
t_i0 = 1'b1;
t_i1 = 1'b1;
t_s0 = 1'b1;
#10
t_i0 = 1'b0;
t_i1 = 1'b0;
t_s0 = 1'b0;
end
endmodule
```



C:\Windows\System32\cmd.exe

```
C:\iverilog\bin>iverilog -o test week3_1.v week3_1tb.v

C:\iverilog\bin>vvp test
VCD info: dumpfile dump.vcd opened for output.
0000
0010
0100
0111
1001
1010
1101
1111
0000

C:\iverilog\bin>gtkwave dump.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[80] end time.
WM Destroy

C:\iverilog\bin>
```

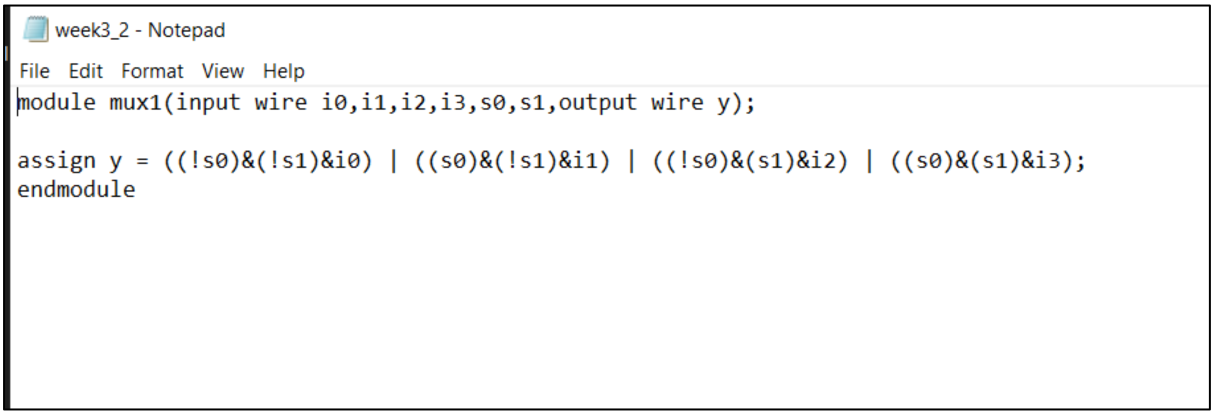


GTKWave - dump.vcd

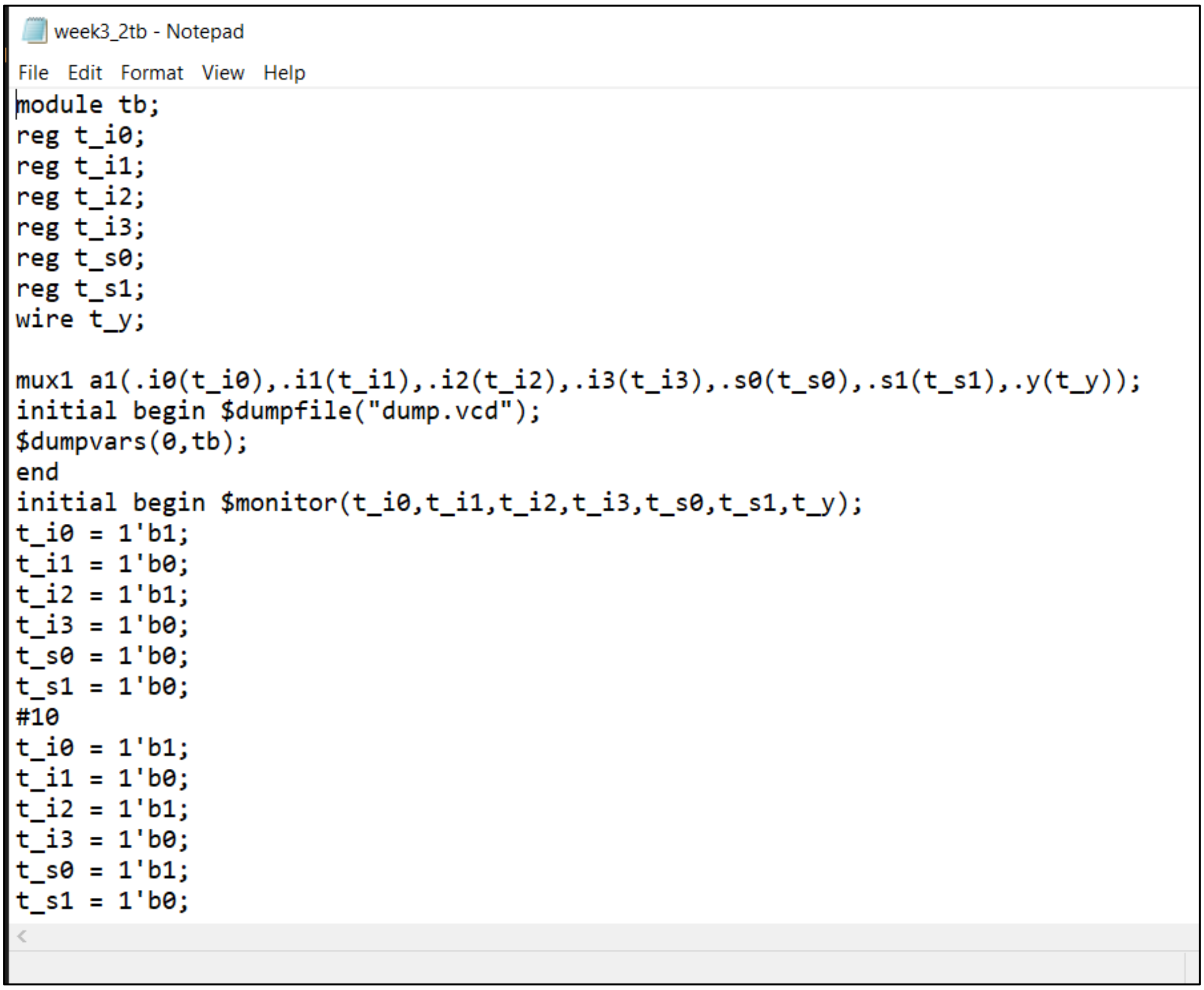
From: 0 sec To: 80 sec Marker: -- | Cursor: 80 sec

Type	Signals
reg	t_i0
reg	t_i1
reg	t_s0
wire	t_y

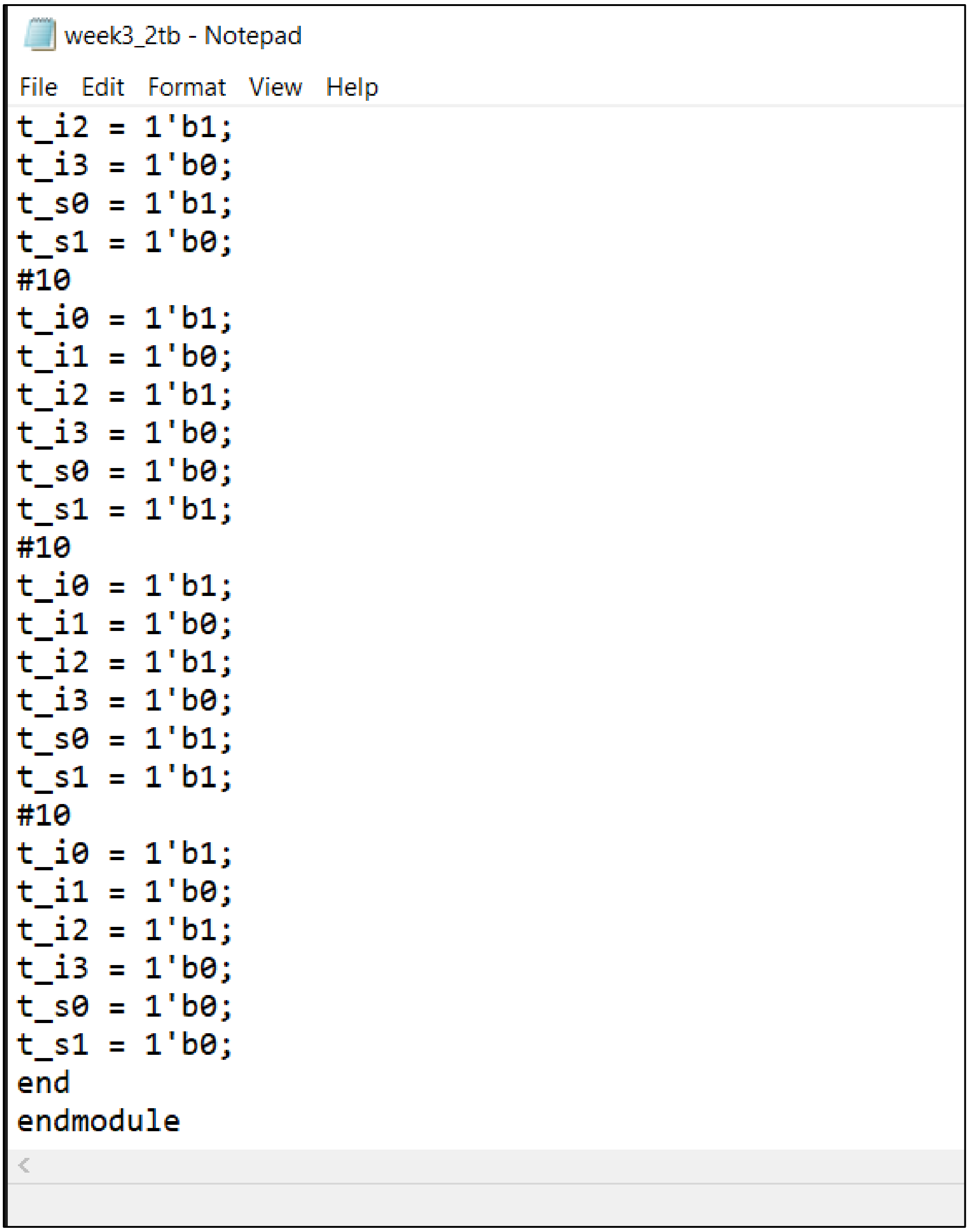
PROGRAM 2: 4:1 MUX



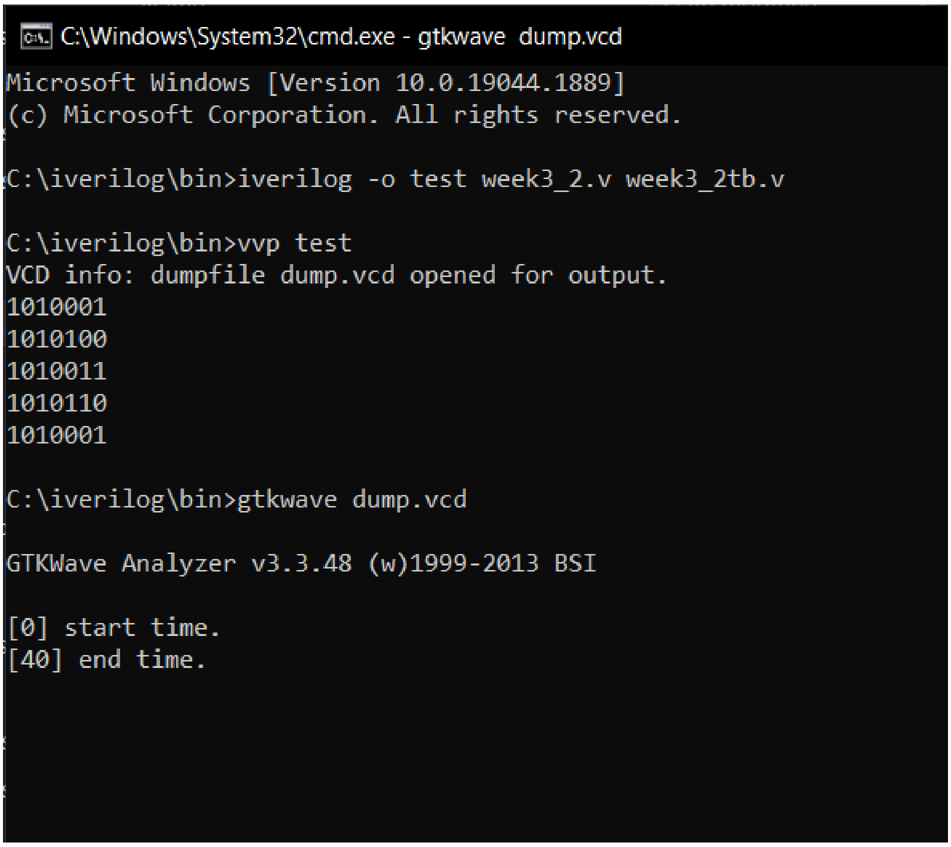
```
module mux1(input wire i0,i1,i2,i3,s0,s1,output wire y);

assign y = ((!s0)&(!s1)&i0) | ((s0)&(!s1)&i1) | ((!s0)&(s1)&i2) | ((s0)&(s1)&i3);
endmodule
```



```
module tb;
reg t_i0;
reg t_i1;
reg t_i2;
reg t_i3;
reg t_s0;
reg t_s1;
wire t_y;

mux1 a1(.i0(t_i0),.i1(t_i1),.i2(t_i2),.i3(t_i3),.s0(t_s0),.s1(t_s1),.y(t_y));
initial begin $dumpfile("dump.vcd");
$dumpvars(0,tb);
end
initial begin $monitor(t_i0,t_i1,t_i2,t_i3,t_s0,t_s1,t_y);
t_i0 = 1'b1;
t_i1 = 1'b0;
t_i2 = 1'b1;
t_i3 = 1'b0;
t_s0 = 1'b0;
t_s1 = 1'b0;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_i2 = 1'b1;
t_i3 = 1'b0;
t_s0 = 1'b1;
t_s1 = 1'b0;
```

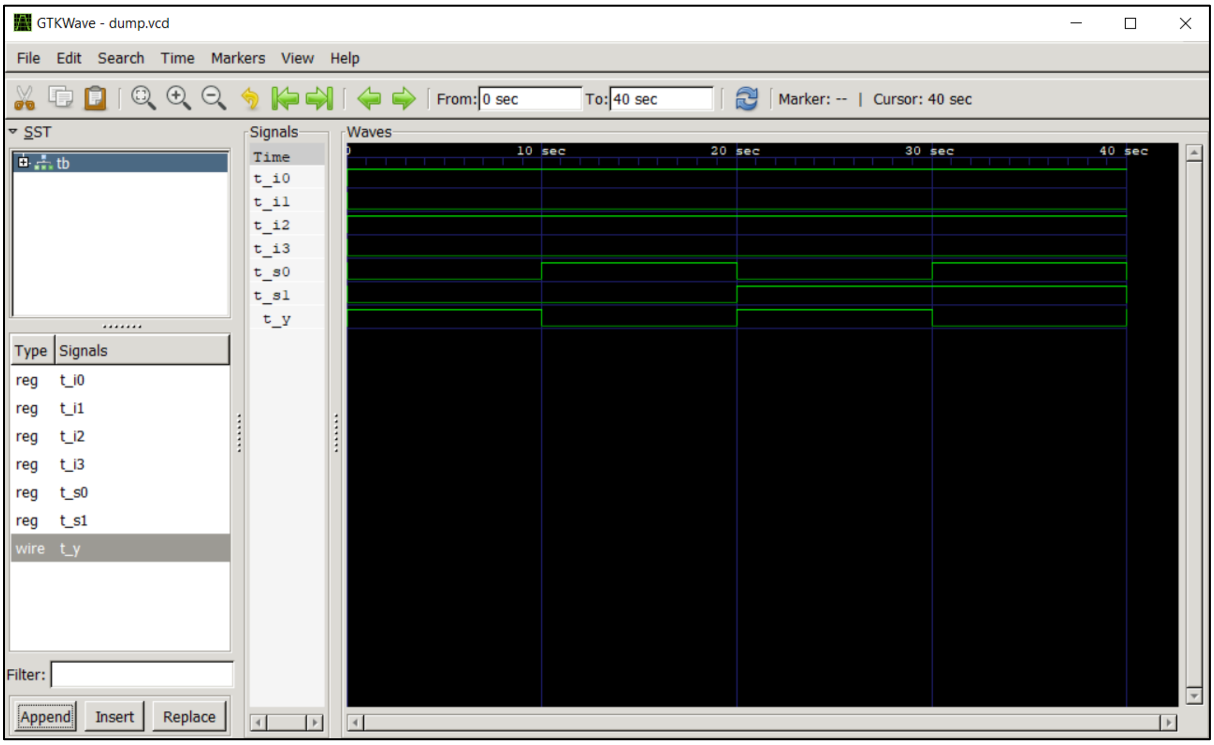


```
t_i2 = 1'b1;
t_i3 = 1'b0;
t_s0 = 1'b1;
t_s1 = 1'b0;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_i2 = 1'b1;
t_i3 = 1'b0;
t_s0 = 1'b0;
t_s1 = 1'b1;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_i2 = 1'b1;
t_i3 = 1'b0;
t_s0 = 1'b1;
t_s1 = 1'b1;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_i2 = 1'b1;
t_i3 = 1'b0;
t_s0 = 1'b0;
t_s1 = 1'b0;
end
endmodule
```



C:\Windows\System32\cmd.exe - gtkwave dump.vcd

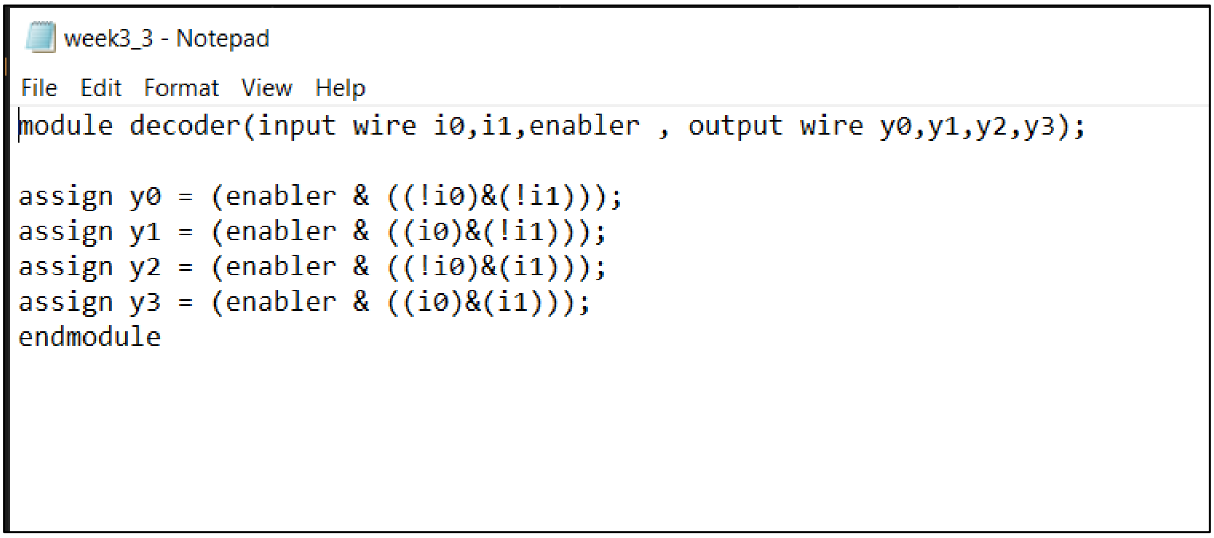
```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\iverilog\bin>iverilog -o test week3_2.v week3_2tb.v

C:\iverilog\bin>vvp test
VCD info: dumpfile dump.vcd opened for output.
1010001
1010100
1010011
1010110
1010001

C:\iverilog\bin>gtkwave dump.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[40] end time.
```



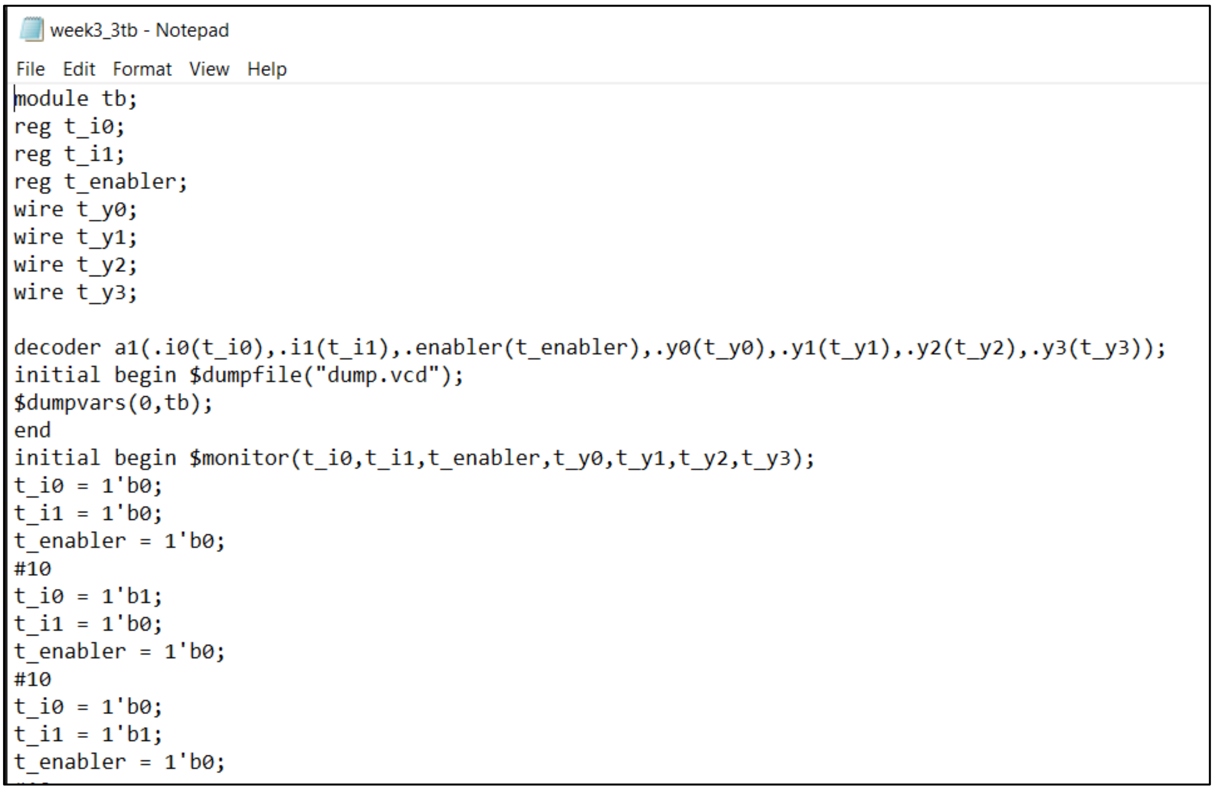
From: 0 sec To: 40 sec Marker: -- | Cursor: 40 sec

Type	Signals
reg	t_i0
reg	t_i1
reg	t_i2
reg	t_i3
reg	t_s0
reg	t_s1
wire	t_y

PROGRAM 3: 2:4 DECODER

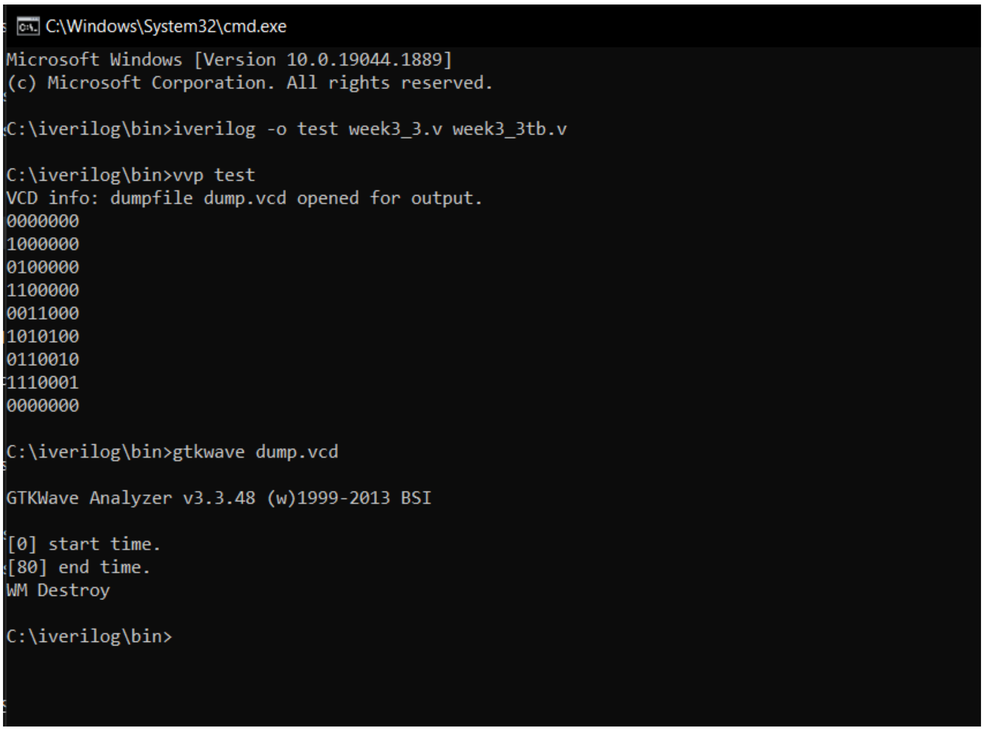


```
module decoder(input wire i0,i1,enabler , output wire y0,y1,y2,y3);

assign y0 = (enabler & ((!i0)&(!i1)));
assign y1 = (enabler & ((i0)&(!i1)));
assign y2 = (enabler & ((!i0)&(i1)));
assign y3 = (enabler & ((i0)&(i1)));
endmodule
```

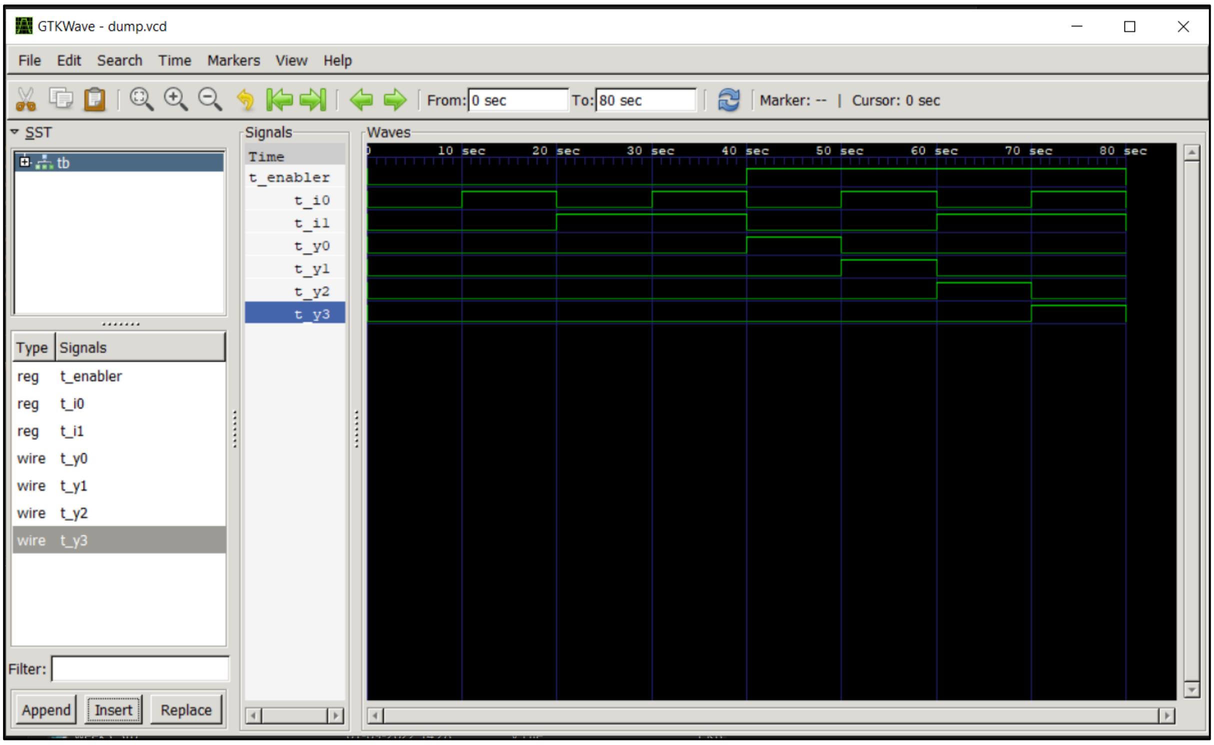


```
module tb;
reg t_i0;
reg t_i1;
reg t_enabler;
wire t_y0;
wire t_y1;
wire t_y2;
wire t_y3;

decoder a1(.i0(t_i0),.i1(t_i1),.enabler(t_enabler),.y0(t_y0),.y1(t_y1),.y2(t_y2),.y3(t_y3));
initial begin $dumpfile("dump.vcd");
$dumpvars(0,tb);
end
initial begin $monitor(t_i0,t_i1,t_enabler,t_y0,t_y1,t_y2,t_y3);
t_i0 = 1'b0;
t_i1 = 1'b0;
t_enabler = 1'b0;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_enabler = 1'b0;
#10
t_i0 = 1'b0;
t_i1 = 1'b1;
t_enabler = 1'b0;
```

```
#10
t_i0 = 1'b1;
t_i1 = 1'b1;
t_enabler = 1'b0;
#10
t_i0 = 1'b0;
t_i1 = 1'b0;
t_enabler = 1'b1;
#10
t_i0 = 1'b1;
t_i1 = 1'b0;
t_enabler = 1'b1;
#10
t_i0 = 1'b0;
t_i1 = 1'b1;
t_enabler = 1'b1;
#10
t_i0 = 1'b1;
t_i1 = 1'b1;
t_enabler = 1'b1;
#10
t_i0 = 1'b0;
t_i1 = 1'b0;
t_enabler = 1'b0;
end
endmodule
```



```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\iverilog\bin>iverilog -o test week3_3.v week3_3tb.v

C:\iverilog\bin>vvp test
VCD info: dumpfile dump.vcd opened for output.
0000000
1000000
0100000
1100000
0011000
1010100
0110010
1110001
0000000

C:\iverilog\bin>gtkwave dump.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[80] end time.
WM Destroy

C:\iverilog\bin>
```



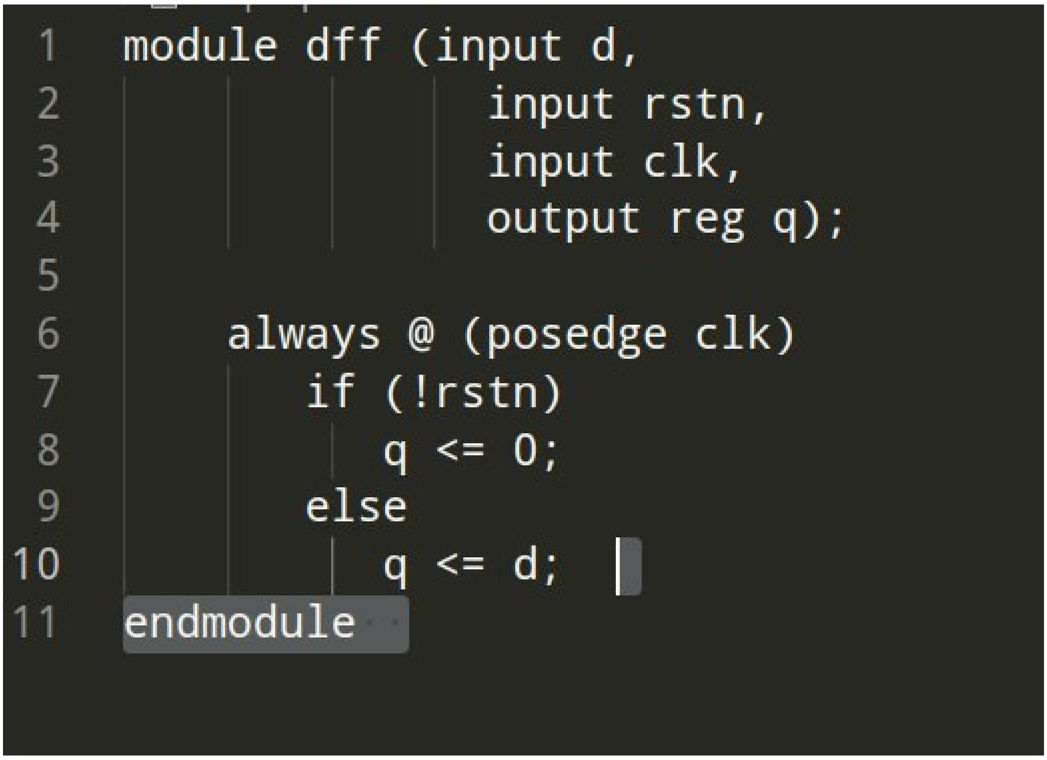
Type	Signals
reg	t_enabler
reg	t_i0
reg	t_i1
wire	t_y0
wire	t_y1
wire	t_y2
wire	t_y3

NAME: SAAD BIN KHALID

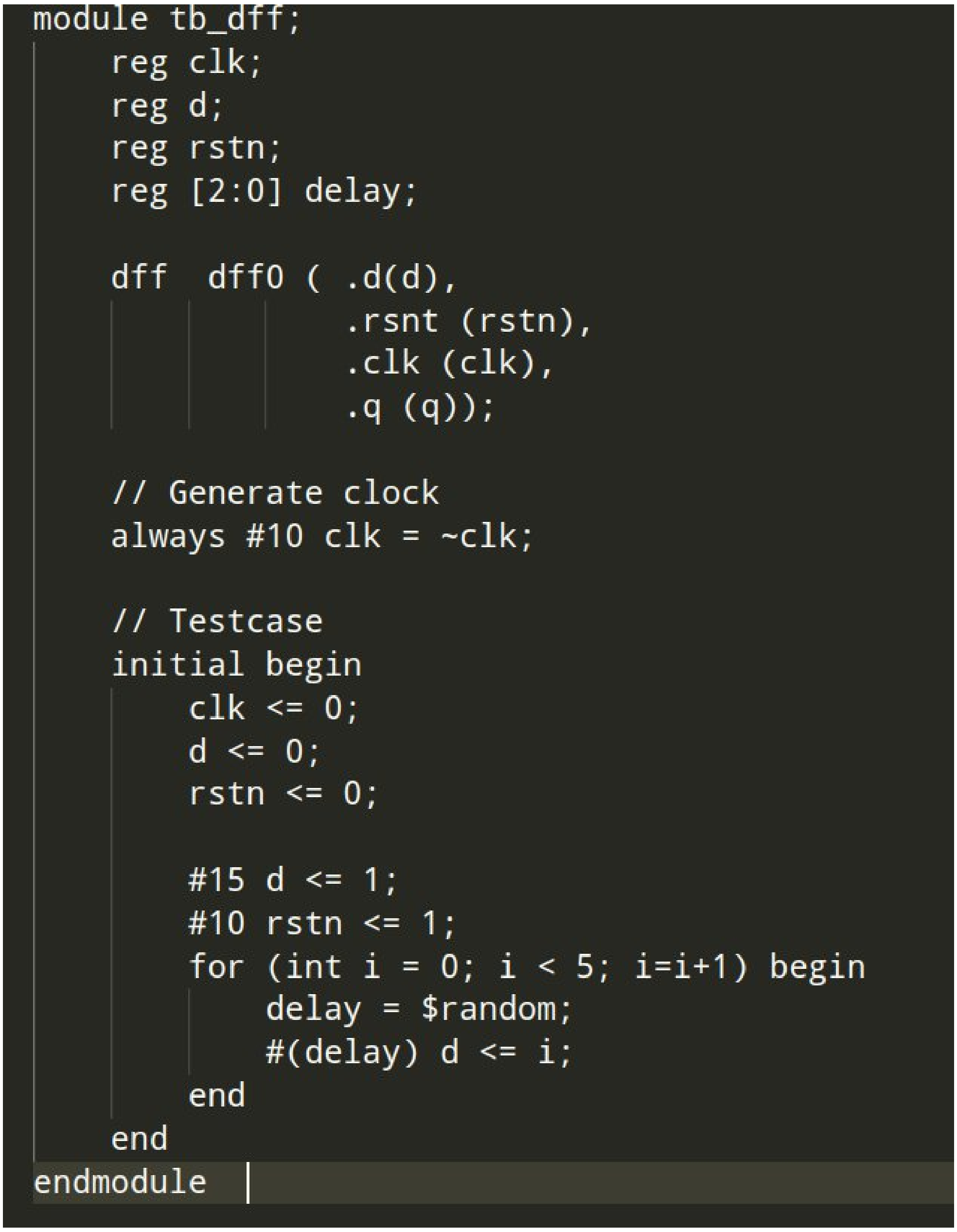
SRN: PES1UG21CS508

ROLL NO: 16

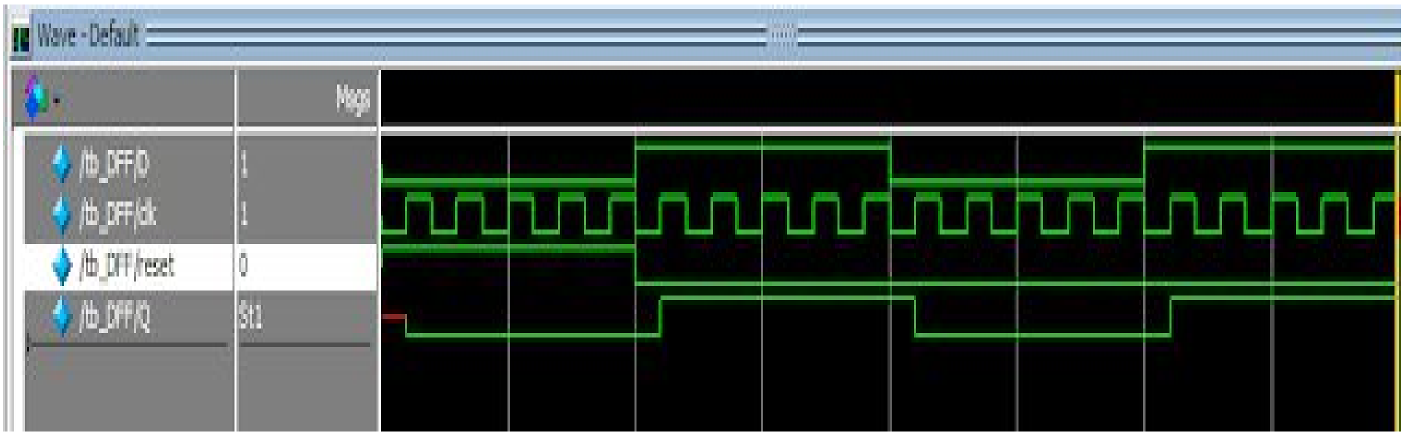
D-FLIP FLOP



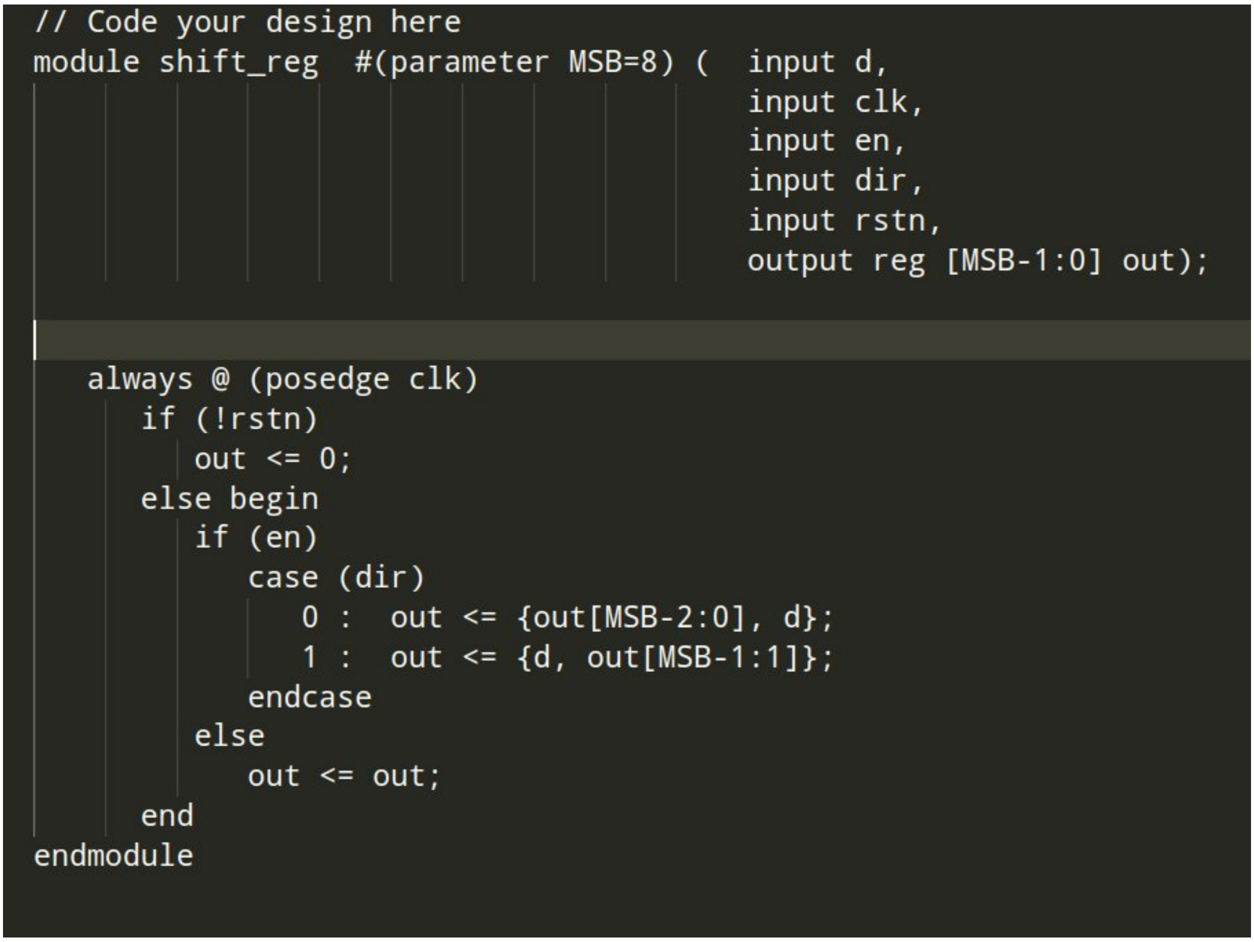
```
module dff (input d,
                    input rstn,
                    input clk,
                    output reg q);

    always @ (posedge clk)
       if (!rstn)
          q <= 0;
       else
          q <= d;
endmodule
```



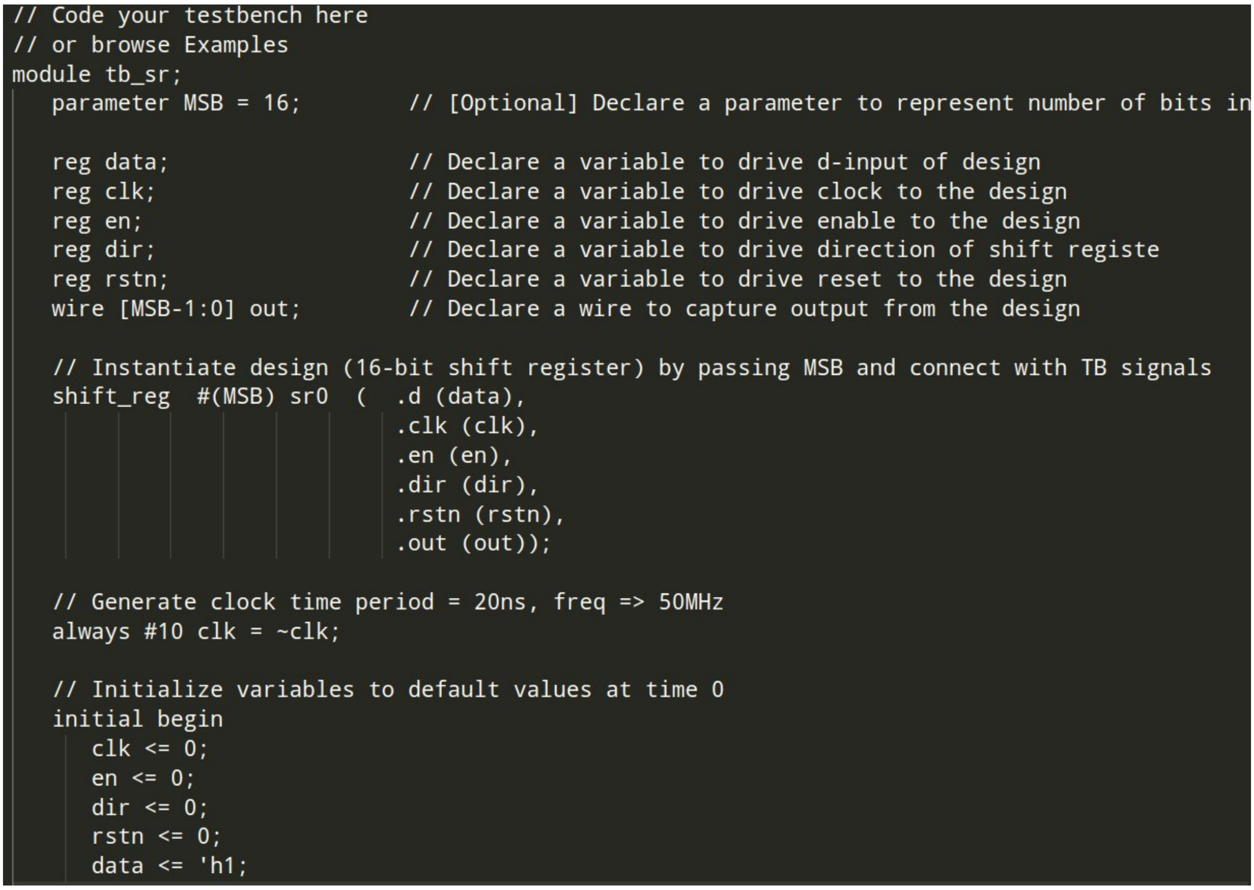
```
module tb_dff;
    reg clk;
    reg d;
    reg rstn;
    reg [2:0] delay;

    dff  dff0 ( .d(d),
                .rsnt (rstn),
                .clk (clk),
                .q (q));

    // Generate clock
    always #10 clk = ~clk;

    // Testcase
    initial begin
        clk <= 0;
        d <= 0;
        rstn <= 0;

        #15 d <= 1;
        #10 rstn <= 1;
        for (int i = 0; i < 5; i=i+1) begin
            delay = $random;
            #(delay) d <= i;
        end
    end
endmodule
```



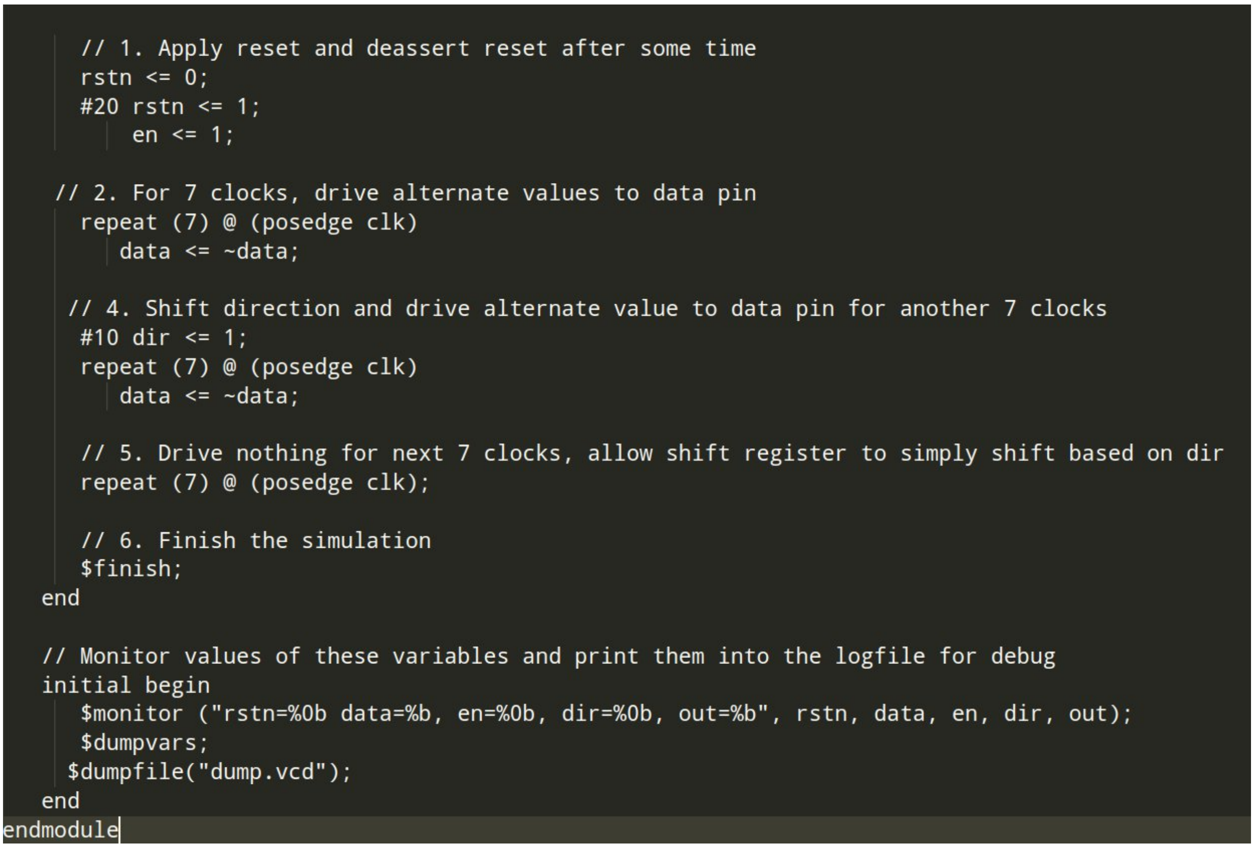
REGISTER:



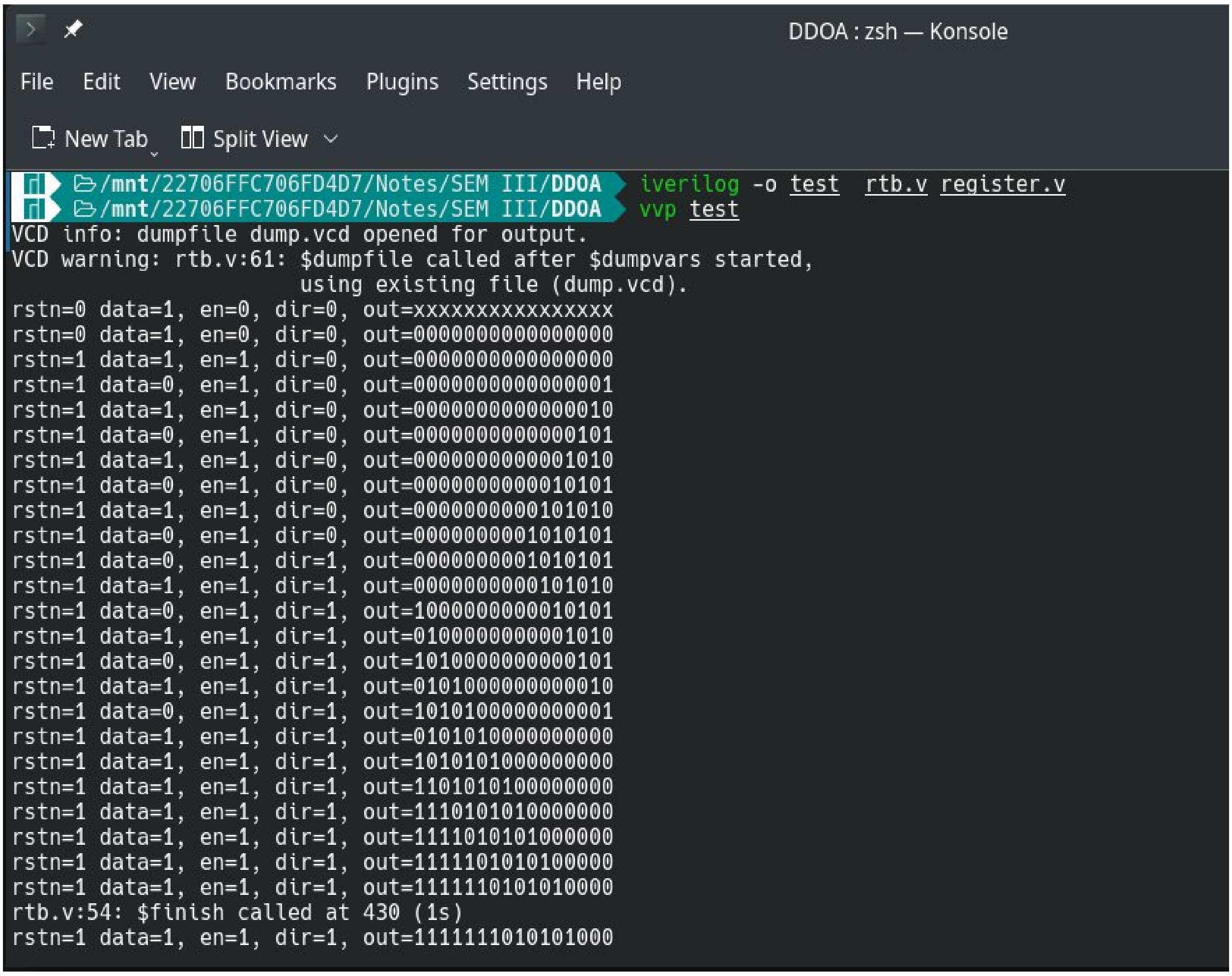
```
// Code your design here
module shift_reg  #(parameter MSB=8) (  input d,
                                        input clk,
                                        input en,
                                        input dir,
                                        input rstn,
                                        output reg [MSB-1:0] out);


   always @ (posedge clk)
      if (!rstn)
         out <= 0;
      else begin
         if (en)
            case (dir)
               0 :  out <= {out[MSB-2:0], d};
               1 :  out <= {d, out[MSB-1:1]};
            endcase
         else
            out <= out;
      end
endmodule
```



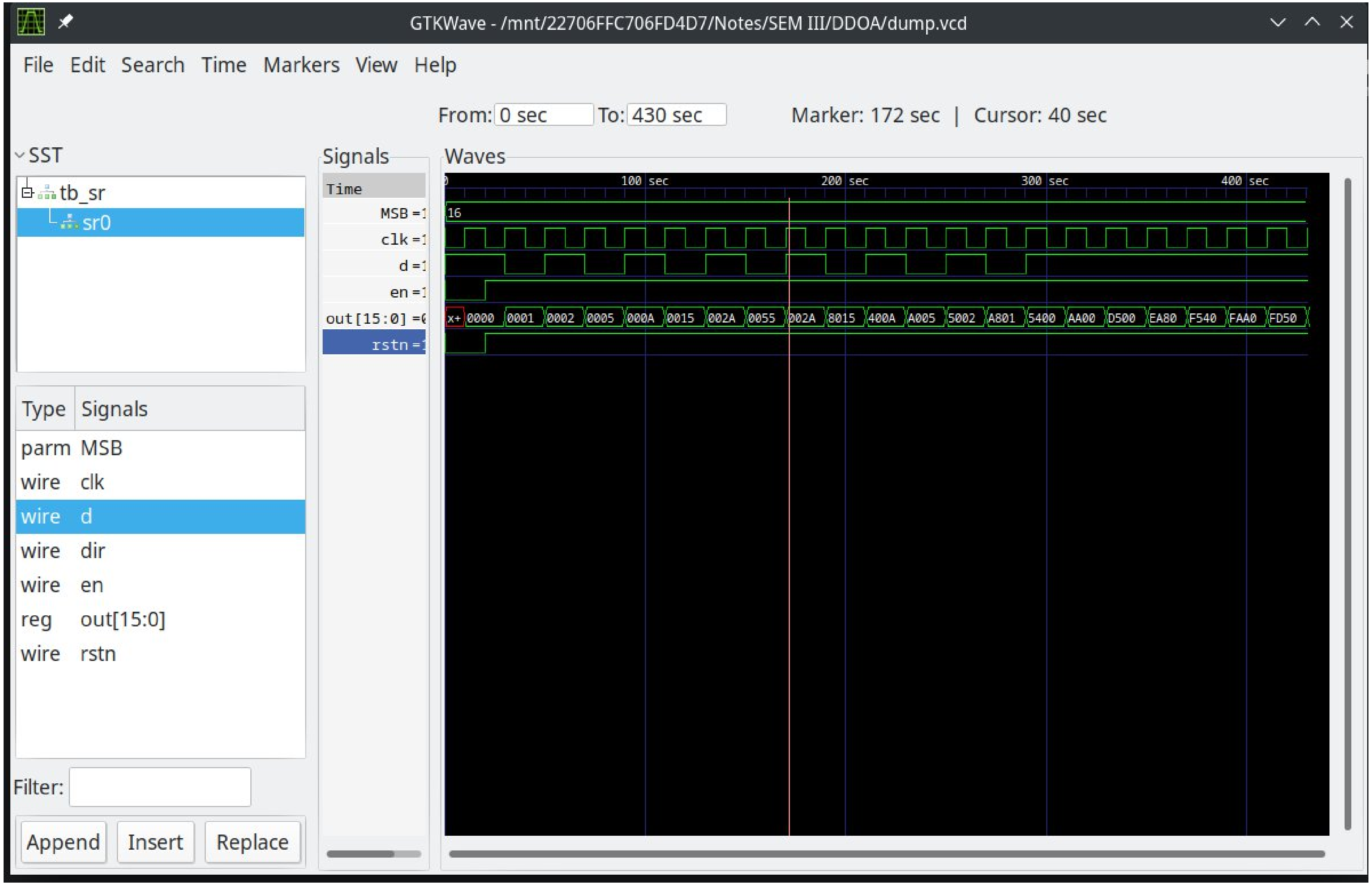
```
// Code your testbench here
// or browse Examples
module tb_sr;
   parameter MSB = 16;        // [Optional] Declare a parameter to represent number of bits in

   reg data;                  // Declare a variable to drive d-input of design
   reg clk;                   // Declare a variable to drive clock to the design
   reg en;                    // Declare a variable to drive enable to the design
   reg dir;                   // Declare a variable to drive direction of shift registe
   reg rstn;                  // Declare a variable to drive reset to the design
   wire [MSB-1:0] out;        // Declare a wire to capture output from the design

   // Instantiate design (16-bit shift register) by passing MSB and connect with TB signals
   shift_reg  #(MSB) sr0  (  .d (data),
                             .clk (clk),
                             .en (en),
                             .dir (dir),
                             .rstn (rstn),
                             .out (out));

   // Generate clock time period = 20ns, freq => 50MHz
   always #10 clk = ~clk;

   // Initialize variables to default values at time 0
   initial begin
      clk <= 0;
      en <= 0;
      dir <= 0;
      rstn <= 0;
      data <= 'h1;
```



```
    // 1. Apply reset and deassert reset after some time
    rstn <= 0;
    #20 rstn <= 1;
        en <= 1;

  // 2. For 7 clocks, drive alternate values to data pin
    repeat (7) @ (posedge clk)
       data <= ~data;

   // 4. Shift direction and drive alternate value to data pin for another 7 clocks
    #10 dir <= 1;
    repeat (7) @ (posedge clk)
       data <= ~data;

    // 5. Drive nothing for next 7 clocks, allow shift register to simply shift based on dir
    repeat (7) @ (posedge clk);

    // 6. Finish the simulation
    $finish;
  end

  // Monitor values of these variables and print them into the logfile for debug
  initial begin
    $monitor ("rstn=%0b data=%b, en=%0b, dir=%0b, out=%b", rstn, data, en, dir, out);
    $dumpvars;
   $dumpfile("dump.vcd");
  end
endmodule
```



```
/mnt/22706FFC706FD4D7/Notes/SEM III/DDOA  iverilog -o test  rtb.v register.v
/mnt/22706FFC706FD4D7/Notes/SEM III/DDOA  vvp test
VCD info: dumpfile dump.vcd opened for output.
VCD warning: rtb.v:61: $dumpfile called after $dumpvars started,
                       using existing file (dump.vcd).
rstn=0 data=1, en=0, dir=0, out=xxxxxxxxxxxxxxxx
rstn=0 data=1, en=0, dir=0, out=0000000000000000
rstn=1 data=1, en=1, dir=0, out=0000000000000000
rstn=1 data=0, en=1, dir=0, out=0000000000000001
rstn=1 data=1, en=1, dir=0, out=0000000000000010
rstn=1 data=0, en=1, dir=0, out=0000000000000101
rstn=1 data=1, en=1, dir=0, out=0000000000001010
rstn=1 data=0, en=1, dir=0, out=0000000000010101
rstn=1 data=1, en=1, dir=0, out=0000000000101010
rstn=1 data=0, en=1, dir=0, out=0000000001010101
rstn=1 data=0, en=1, dir=1, out=0000000001010101
rstn=1 data=1, en=1, dir=1, out=0000000000101010
rstn=1 data=0, en=1, dir=1, out=1000000000010101
rstn=1 data=1, en=1, dir=1, out=0100000000001010
rstn=1 data=0, en=1, dir=1, out=1010000000000101
rstn=1 data=1, en=1, dir=1, out=0101000000000010
rstn=1 data=0, en=1, dir=1, out=1010100000000001
rstn=1 data=1, en=1, dir=1, out=0101010000000000
rstn=1 data=1, en=1, dir=1, out=1010101000000000
rstn=1 data=1, en=1, dir=1, out=1101010100000000
rstn=1 data=1, en=1, dir=1, out=1110101010000000
rstn=1 data=1, en=1, dir=1, out=1111010101000000
rstn=1 data=1, en=1, dir=1, out=1111101010100000
rstn=1 data=1, en=1, dir=1, out=1111110101010000
rtb.v:54: $finish called at 430 (1s)
rstn=1 data=1, en=1, dir=1, out=1111111010101000
```


GTKWave - /mnt/22706FFC706FD4D7/Notes/SEM III/DDOA/dump.vcd
File Edit Search Time Markers View Help
SST
tb_sr
sr0
Signals
Waves
Time
MSB =1
clk =1
d =1
en =1
out[15:0] =0
rstn =1
0
100 sec
200 sec
300 sec
400 sec
16
x+ 0000 0001 0002 0005 000A 0015 002A 0055 002A 8015 400A A005 5002 A801 5400 AA00 D500 EA80 F540 FAA0 FD50
Filter:
Append
Insert
Replace

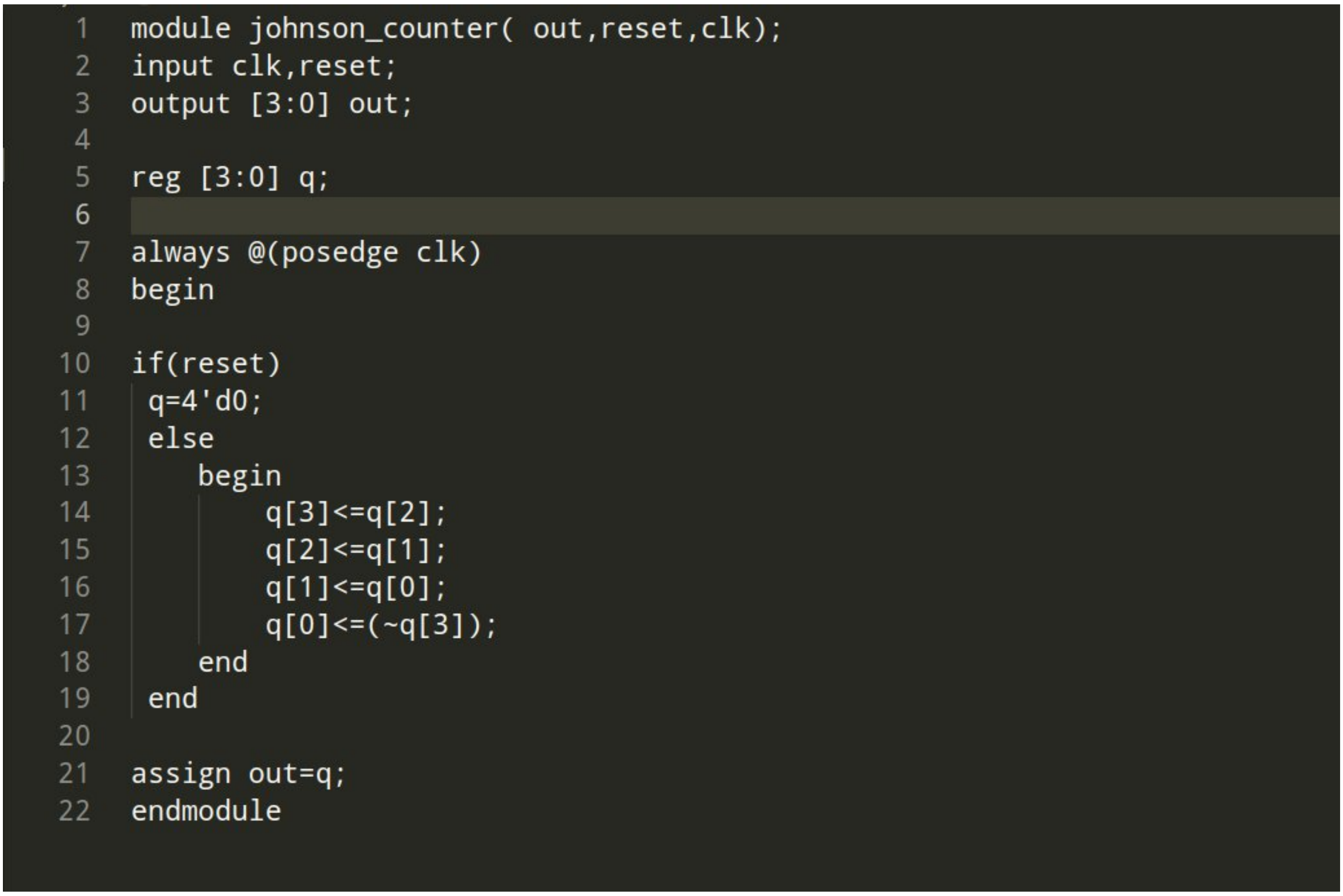
From: 0 sec To: 430 sec Marker: 172 sec | Cursor: 40 sec

Type	Signals
parm	MSB
wire	clk
wire	d
wire	dir
wire	en
reg	out[15:0]
wire	rstn

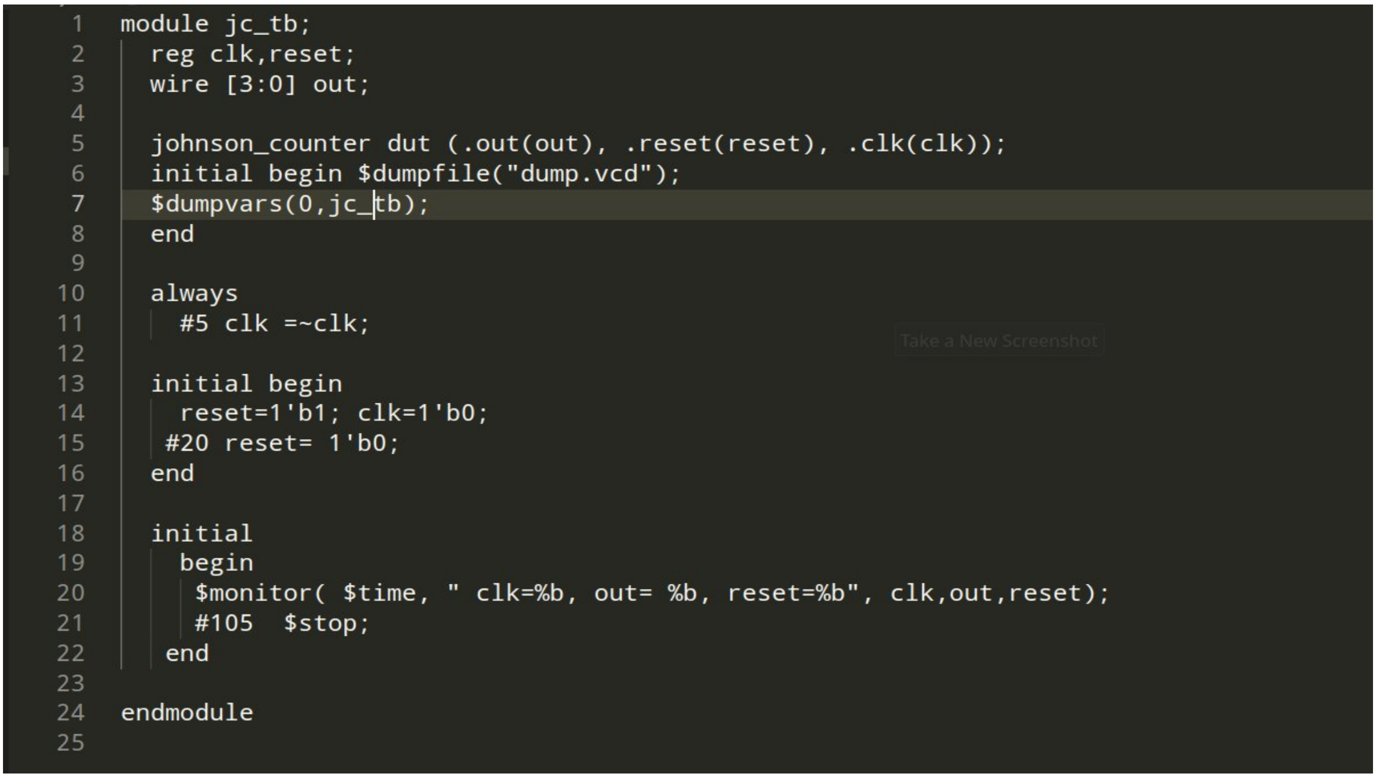
NAME: SAAD BIN KHALID

SRN: PES1UG21CS508

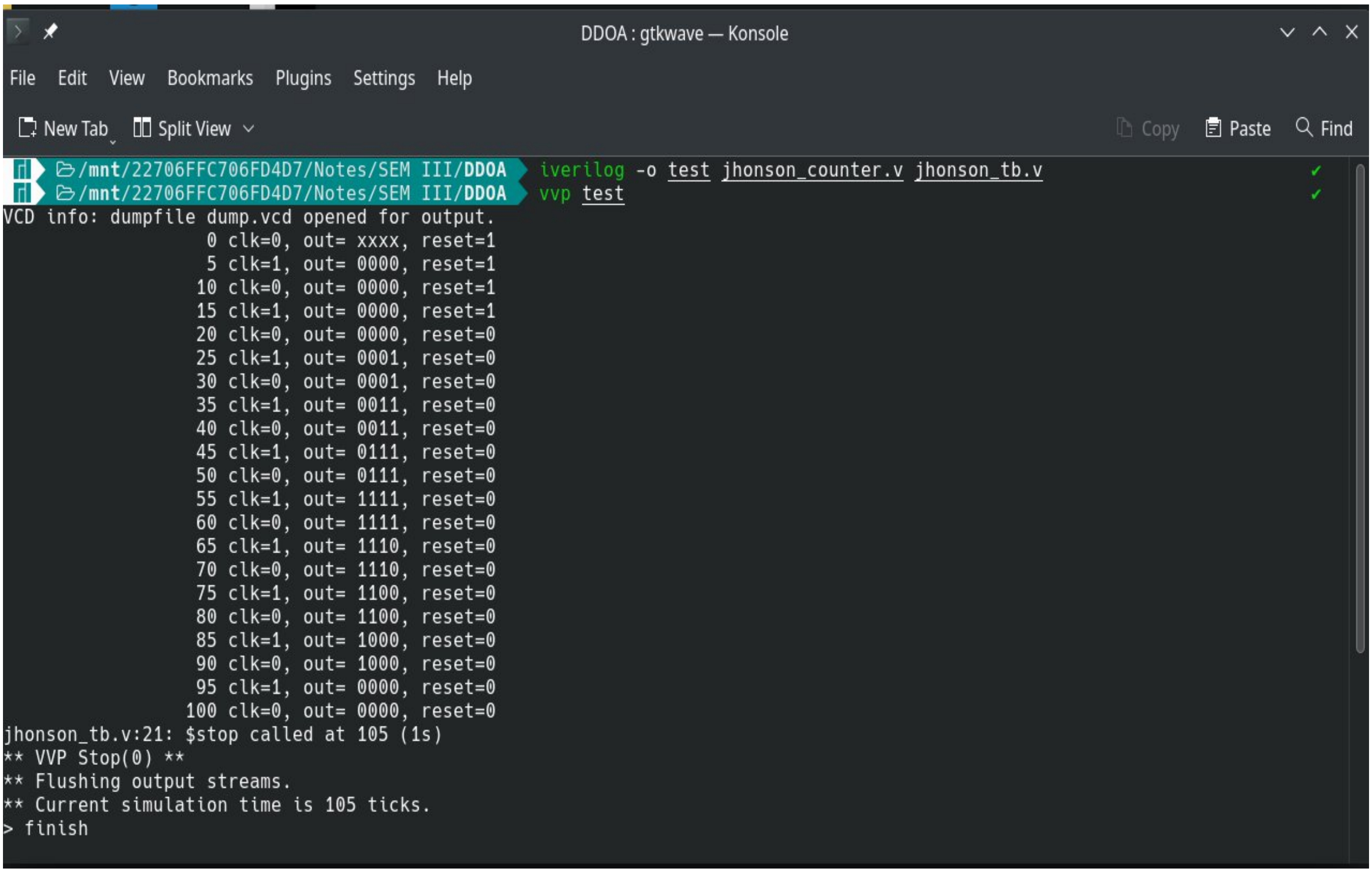
ROLL NO: 16



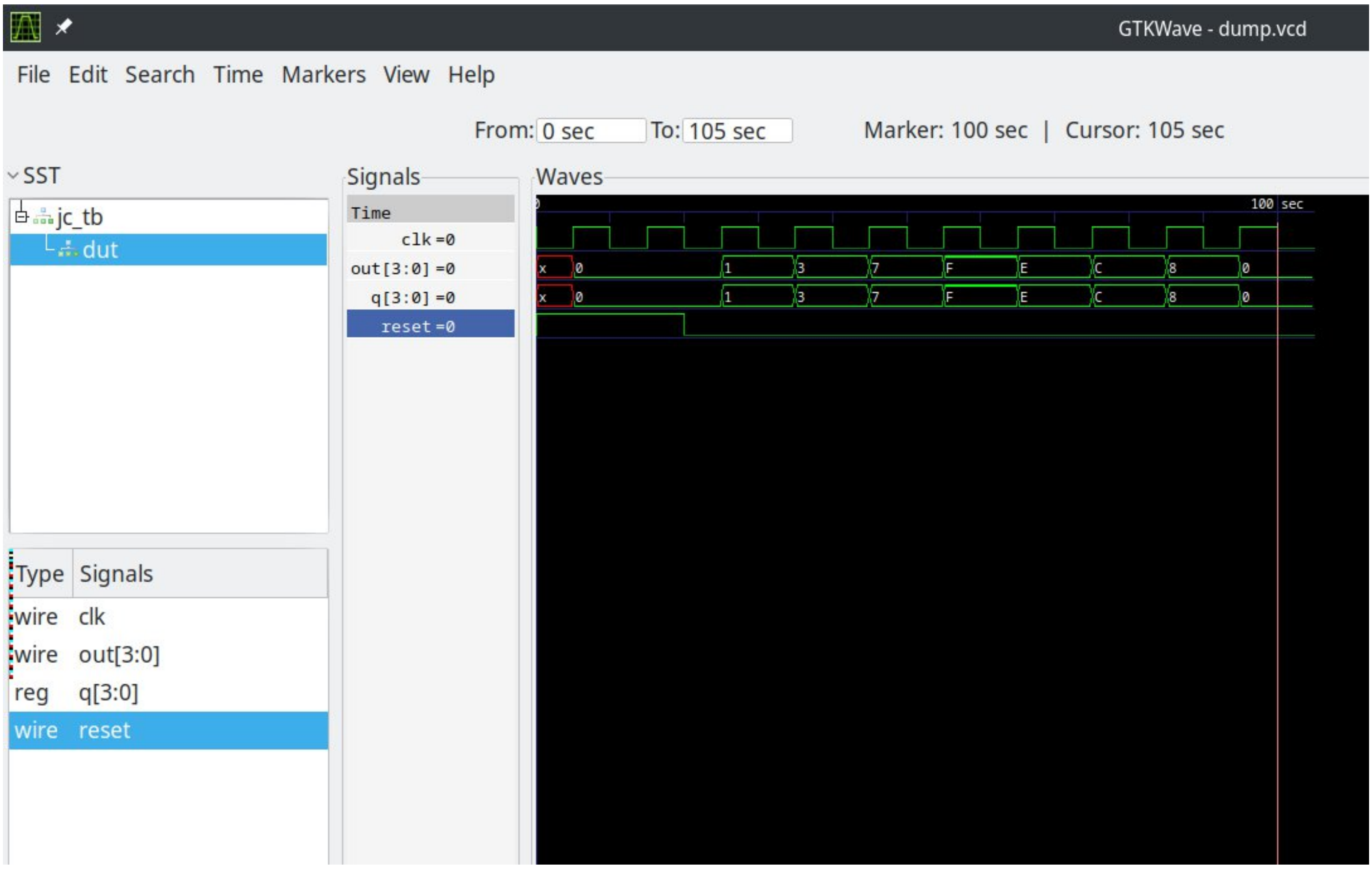
```
module johnson_counter( out,reset,clk);
input clk,reset;
output [3:0] out;

reg [3:0] q;

always @(posedge clk)
begin

if(reset)
 q=4'd0;
 else
    begin
        q[3]<=q[2];
        q[2]<=q[1];
        q[1]<=q[0];
        q[0]<=(~q[3]);
    end
 end

assign out=q;
endmodule
```



```
module jc_tb;
  reg clk,reset;
  wire [3:0] out;

  johnson_counter dut (.out(out), .reset(reset), .clk(clk));
  initial begin $dumpfile("dump.vcd");
  $dumpvars(0,jc_tb);
  end

  always
    #5 clk =~clk;

  initial begin
    reset=1'b1; clk=1'b0;
   #20 reset= 1'b0;
  end

  initial
    begin
     $monitor( $time, " clk=%b, out= %b, reset=%b", clk,out,reset);
     #105  $stop;
   end

endmodule
```



```
/mnt/22706FFC706FD4D7/Notes/SEM III/DDOA  iverilog -o test jhonson_counter.v jhonson_tb.v
/mnt/22706FFC706FD4D7/Notes/SEM III/DDOA  vvp test
VCD info: dumpfile dump.vcd opened for output.
                   0 clk=0, out= xxxx, reset=1
                   5 clk=1, out= 0000, reset=1
                  10 clk=0, out= 0000, reset=1
                  15 clk=1, out= 0000, reset=1
                  20 clk=0, out= 0000, reset=0
                  25 clk=1, out= 0001, reset=0
                  30 clk=0, out= 0001, reset=0
                  35 clk=1, out= 0011, reset=0
                  40 clk=0, out= 0011, reset=0
                  45 clk=1, out= 0111, reset=0
                  50 clk=0, out= 0111, reset=0
                  55 clk=1, out= 1111, reset=0
                  60 clk=0, out= 1111, reset=0
                  65 clk=1, out= 1110, reset=0
                  70 clk=0, out= 1110, reset=0
                  75 clk=1, out= 1100, reset=0
                  80 clk=0, out= 1100, reset=0
                  85 clk=1, out= 1000, reset=0
                  90 clk=0, out= 1000, reset=0
                  95 clk=1, out= 0000, reset=0
                 100 clk=0, out= 0000, reset=0
jhonson_tb.v:21: $stop called at 105 (1s)
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 105 ticks.
> finish
```



From: 0 sec To: 105 sec Marker: 100 sec | Cursor: 105 sec

Type	Signals
wire	clk
wire	out[3:0]
reg	q[3:0]
wire	reset

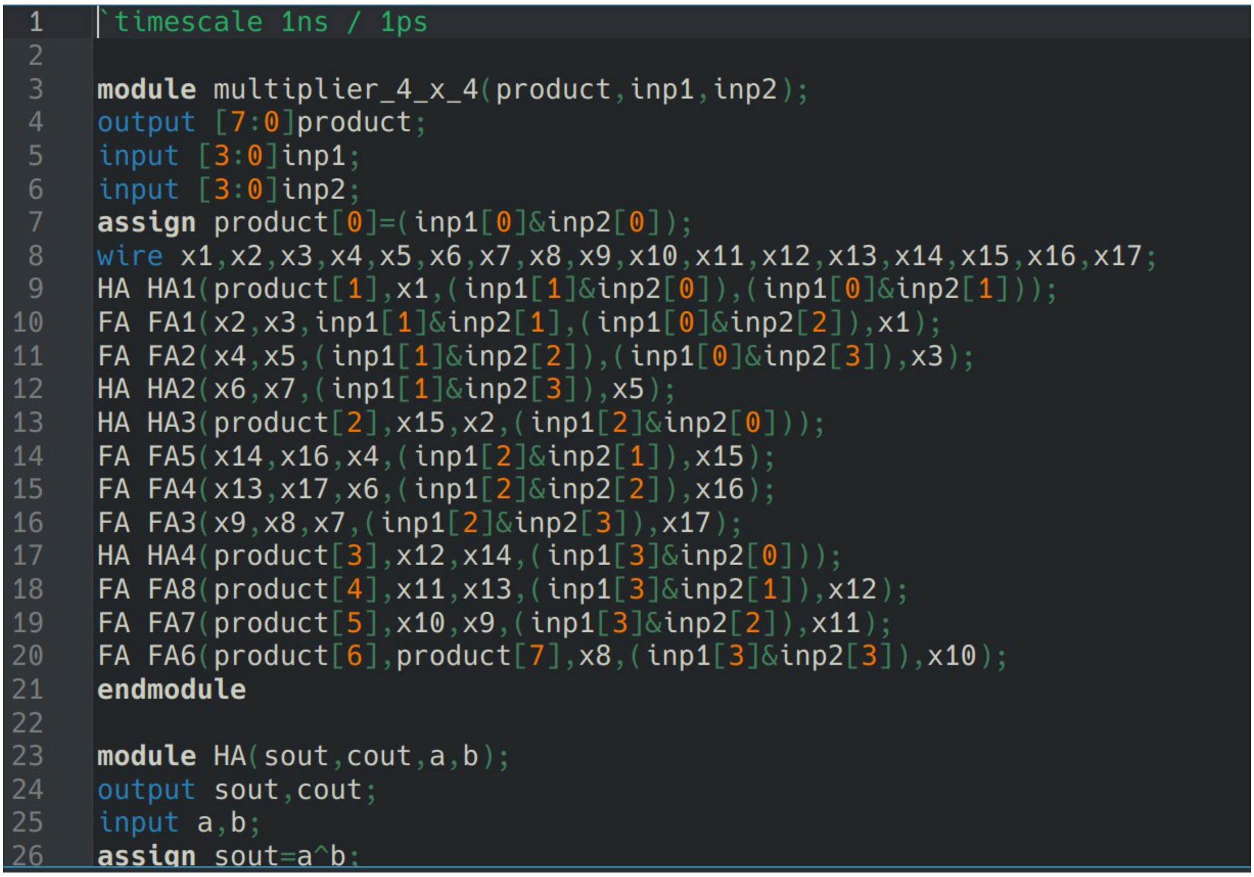
NAME: SAAD BIN KHALID

ROLL NO:16

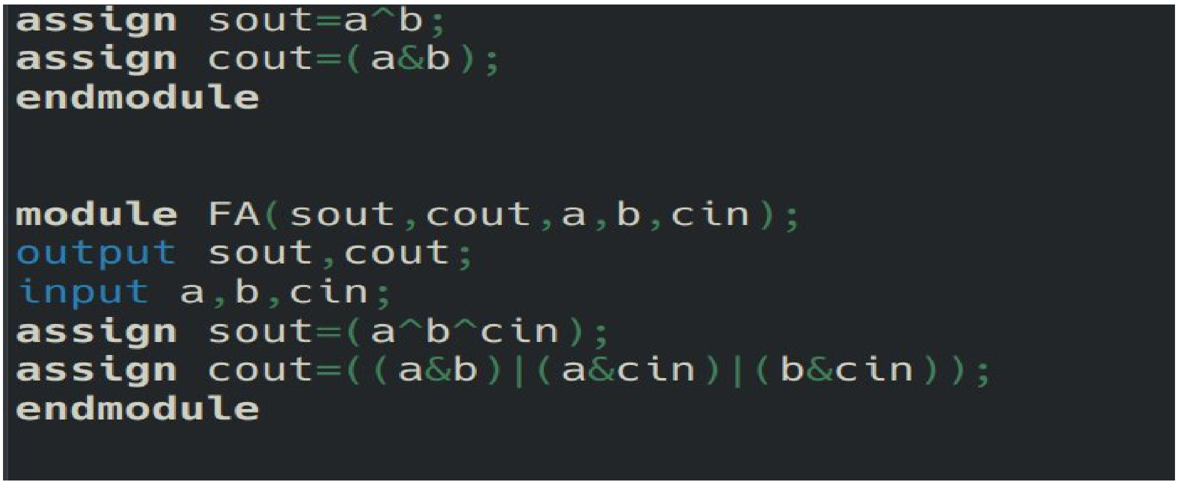
SRN:PES1UG21CS508

TOPIC: 4 BIT ARRAY MULTIPLIER

ArrayMultiplier.v

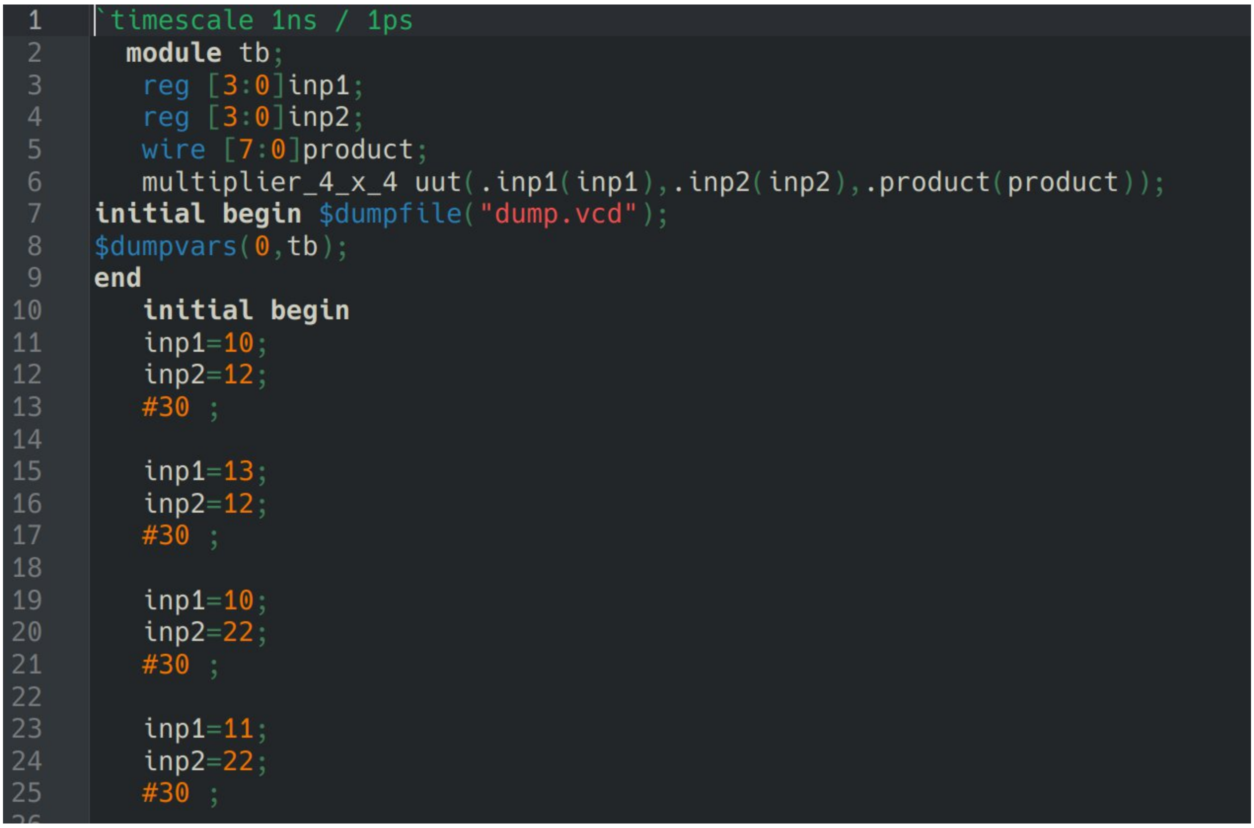


```
`timescale 1ns / 1ps

module multiplier_4_x_4(product,inp1,inp2);
output [7:0]product;
input [3:0]inp1;
input [3:0]inp2;
assign product[0]=(inp1[0]&inp2[0]);
wire x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17;
HA HA1(product[1],x1,(inp1[1]&inp2[0]),(inp1[0]&inp2[1]));
FA FA1(x2,x3,inp1[1]&inp2[1],(inp1[0]&inp2[2]),x1);
FA FA2(x4,x5,(inp1[1]&inp2[2]),(inp1[0]&inp2[3]),x3);
HA HA2(x6,x7,(inp1[1]&inp2[3]),x5);
HA HA3(product[2],x15,x2,(inp1[2]&inp2[0]));
FA FA5(x14,x16,x4,(inp1[2]&inp2[1]),x15);
FA FA4(x13,x17,x6,(inp1[2]&inp2[2]),x16);
FA FA3(x9,x8,x7,(inp1[2]&inp2[3]),x17);
HA HA4(product[3],x12,x14,(inp1[3]&inp2[0]));
FA FA8(product[4],x11,x13,(inp1[3]&inp2[1]),x12);
FA FA7(product[5],x10,x9,(inp1[3]&inp2[2]),x11);
FA FA6(product[6],product[7],x8,(inp1[3]&inp2[3]),x10);
endmodule

module HA(sout,cout,a,b);
output sout,cout;
input a,b;
assign sout=a^b;
```

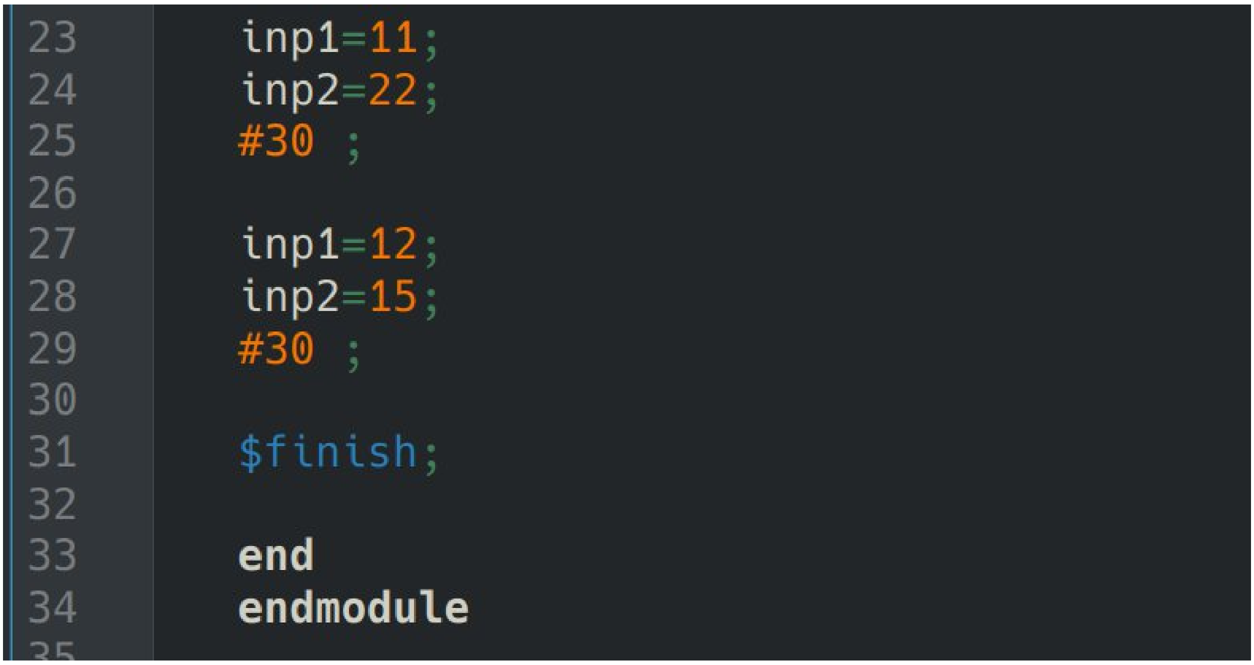


```
assign sout=a^b;
assign cout=(a&b);
endmodule


module FA(sout,cout,a,b,cin);
output sout,cout;
input a,b,cin;
assign sout=(a^b^cin);
assign cout=((a&b)|(a&cin)|(b&cin));
endmodule
```

Test.v

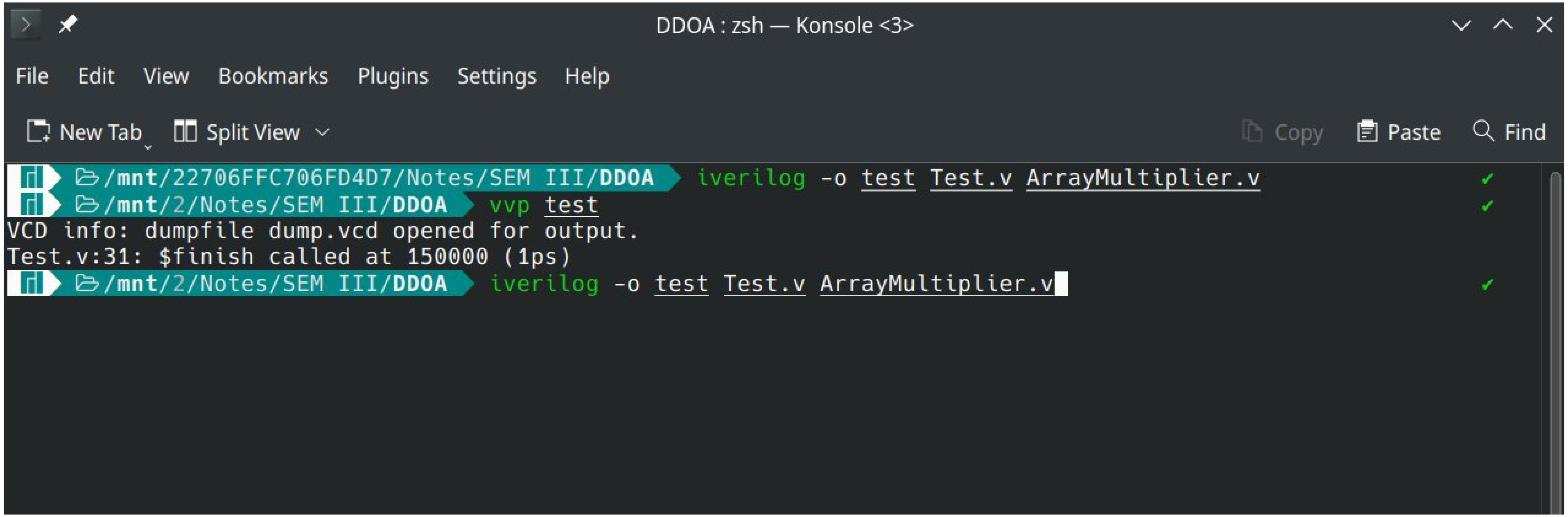


```verilog
`timescale 1ns / 1ps
  module tb;
   reg [3:0]inp1;
   reg [3:0]inp2;
   wire [7:0]product;
   multiplier_4_x_4 uut(.inp1(inp1),.inp2(inp2),.product(product));
initial begin $dumpfile("dump.vcd");
$dumpvars(0,tb);
end
   initial begin
   inp1=10;
   inp2=12;
   #30 ;

   inp1=13;
   inp2=12;
   #30 ;

   inp1=10;
   inp2=22;
   #30 ;

   inp1=11;
   inp2=22;
   #30 ;
```



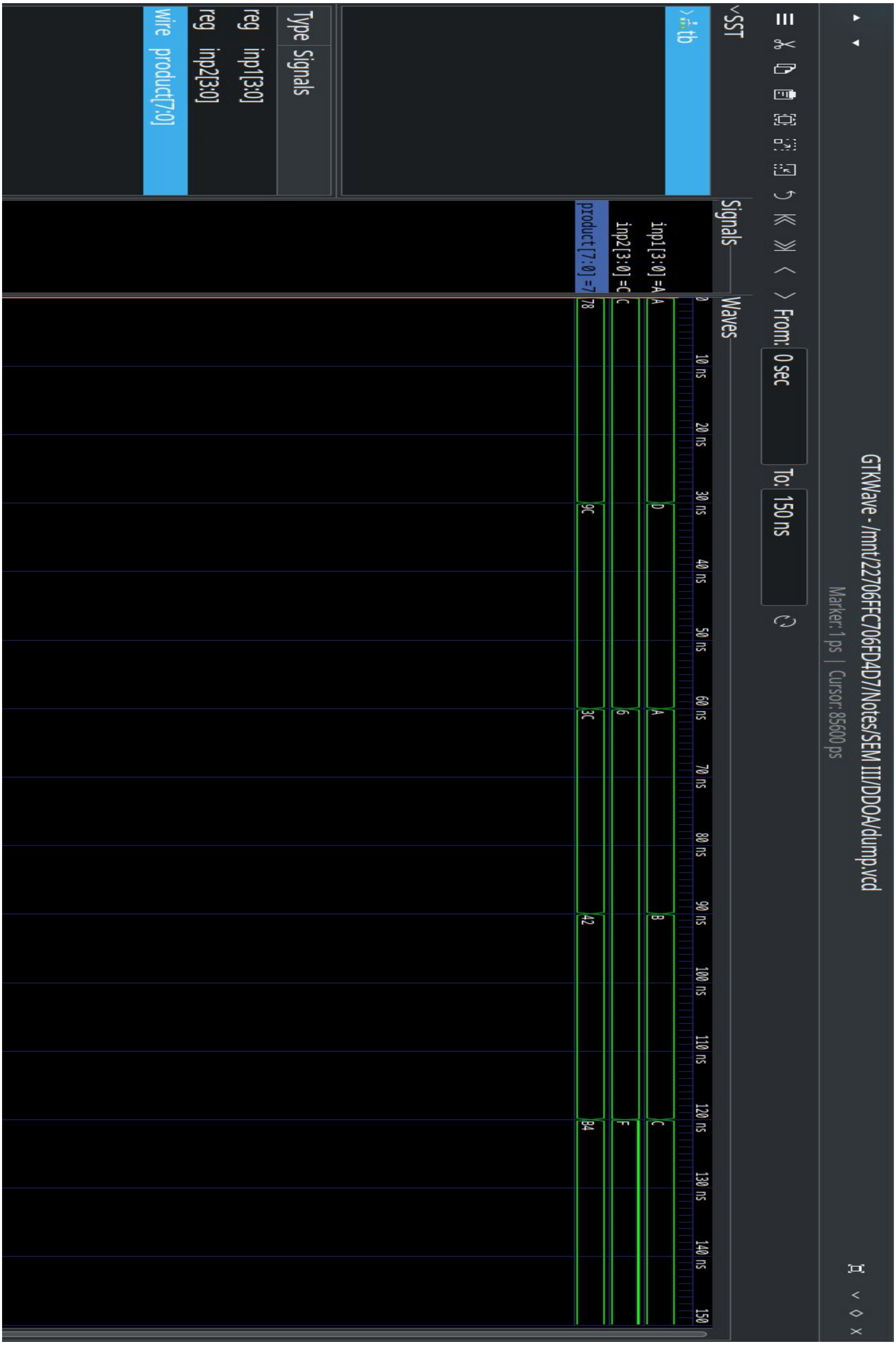
```verilog
   inp1=11;
   inp2=22;
   #30 ;

   inp1=12;
   inp2=15;
   #30 ;

   $finish;

   end
   endmodule
```

Console:



```
/mnt/22706FFC706FD4D7/Notes/SEM III/DDOA  iverilog -o test Test.v ArrayMultiplier.v   ✓
/mnt/2/Notes/SEM III/DDOA  vvp test                                                   ✓
VCD info: dumpfile dump.vcd opened for output.
Test.v:31: $finish called at 150000 (1ps)
/mnt/2/Notes/SEM III/DDOA  iverilog -o test Test.v ArrayMultiplier.v                  ✓
```

GtkWave:



GTKWave - /mnt/22706FFC706FD4D7/Notes/SEM III/DDOA/dump.vcd

Marker: 1 ps | Cursor: 85600 ps

From: 0 sec To: 150 ns

SST

tb

Type	Signals
reg	inp1[3:0]
reg	inp2[3:0]
wire	product[7:0]

Signals

inp1[3:0] =A

inp2[3:0] =C

product[7:0] =78