

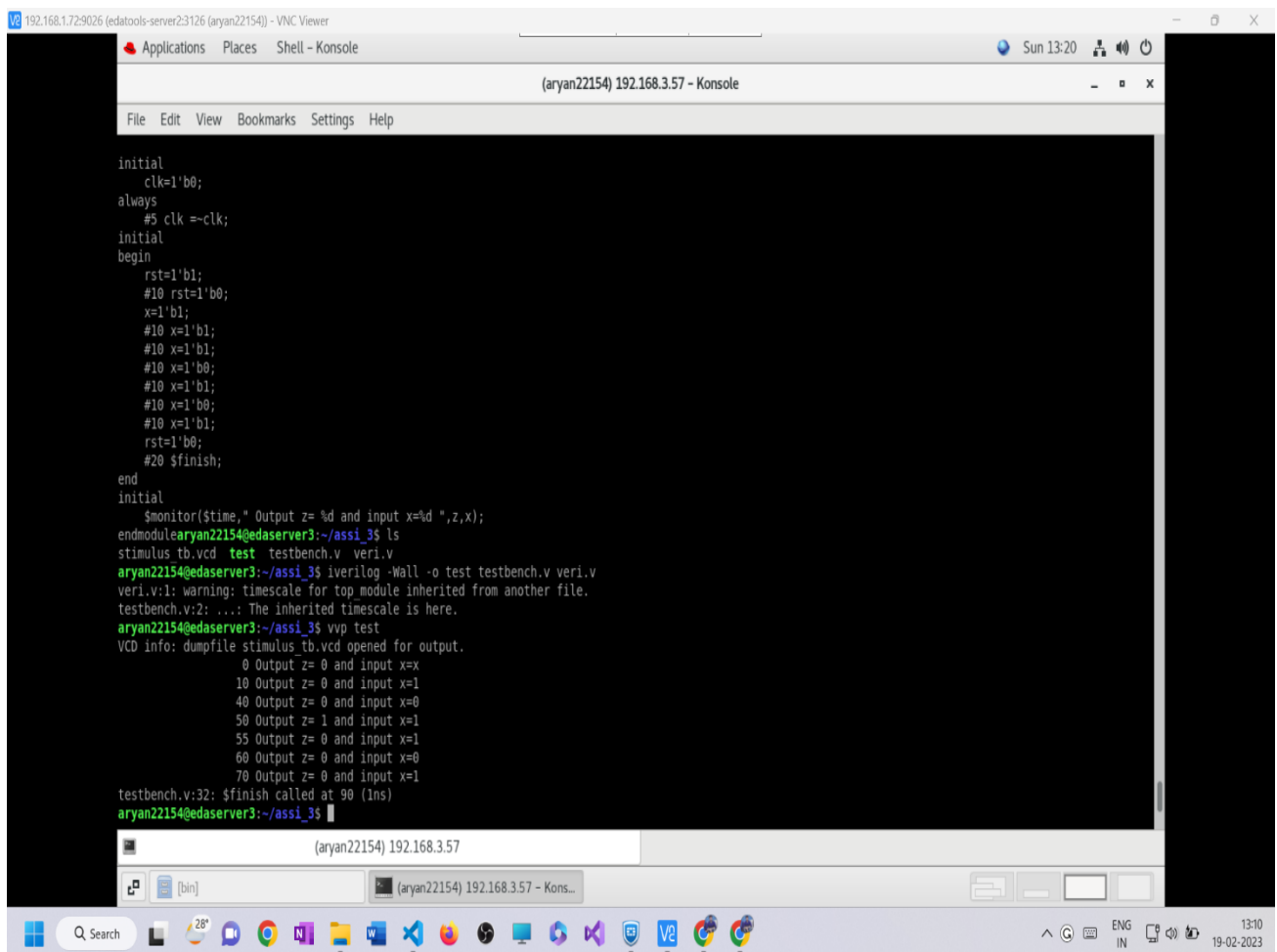
ASSIGNMENT -3

Name: - Aryan Gupta

Roll No- MT22154

VLSI Design Flow (Submitted to Prof. Sneha Saurabh)

1. Show a screenshot or snippet of the log file to prove that you have indeed run the tool
[If this is not given then your assignment will not be evaluated].



The screenshot shows a VNC Viewer window titled "192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer". The main window displays a terminal session with the following content:

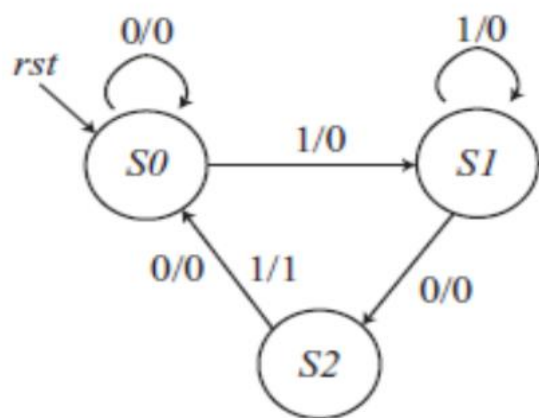
```
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help

initial
  clk='b0;
always
  #5 clk =~clk;
initial
begin
  rst='b1;
  #10 rst='b0;
  x='b1;
  #10 x='b1;
  #10 x='b1;
  #10 x='b0;
  #10 x='b1;
  #10 x='b0;
  #10 x='b1;
  rst='b0;
  #20 $finish;
end
initial
  $monitor($time," Output z= %d and input x=%d ",z,x);
endmodule
aryan22154@edaserver3:~/assi_3$ ls
stimulus tb.vcd test testbench.v veri.v
aryan22154@edaserver3:~/assi_3$ iverilog -Wall -o test testbench.v veri.v
veri.v:1: warning: timescale for top module inherited from another file.
testbench.v:2: ...: The inherited timescale is here.
aryan22154@edaserver3:~/assi_3$ vvp test
VCD info: dumpfile stimulus.tb.vcd opened for output.
      0 Output z= 0 and input x=x
     10 Output z= 0 and input x=1
     40 Output z= 0 and input x=0
     50 Output z= 1 and input x=1
     55 Output z= 0 and input x=1
     60 Output z= 0 and input x=0
     70 Output z= 0 and input x=1
testbench.v:32: $finish called at 90 (ins)
aryan22154@edaserver3:~/assi_3$
```

The terminal window is titled "(aryan22154) 192.168.3.57 - Kons...". The taskbar at the bottom shows various application icons and the system clock indicating 13:10 on 19-02-2023.

2. Consider the state machine shown below.

(a) Write a Verilog model for the following state machine.



The states are: S0, S1, and S2.

The input values are {0, 1}.

Output values are: {0,1}.

```
module top_module(input clk,input rst,input x,output reg z);
parameter s0 = 3'b000;
parameter s1 = 3'b001;
parameter s2 = 3'b010;
reg [2:0] cur_state,next_state;
always@(posedge clk or posedge rst)
begin
    if(rst)
        cur_state<=s0;
    else
        cur_state<=next_state;
end
always@(*)
begin
    case(cur_state)
```

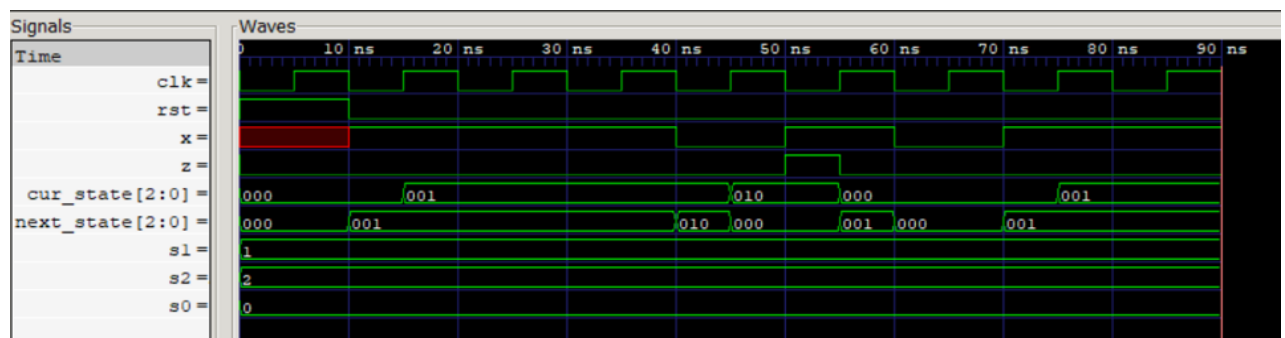
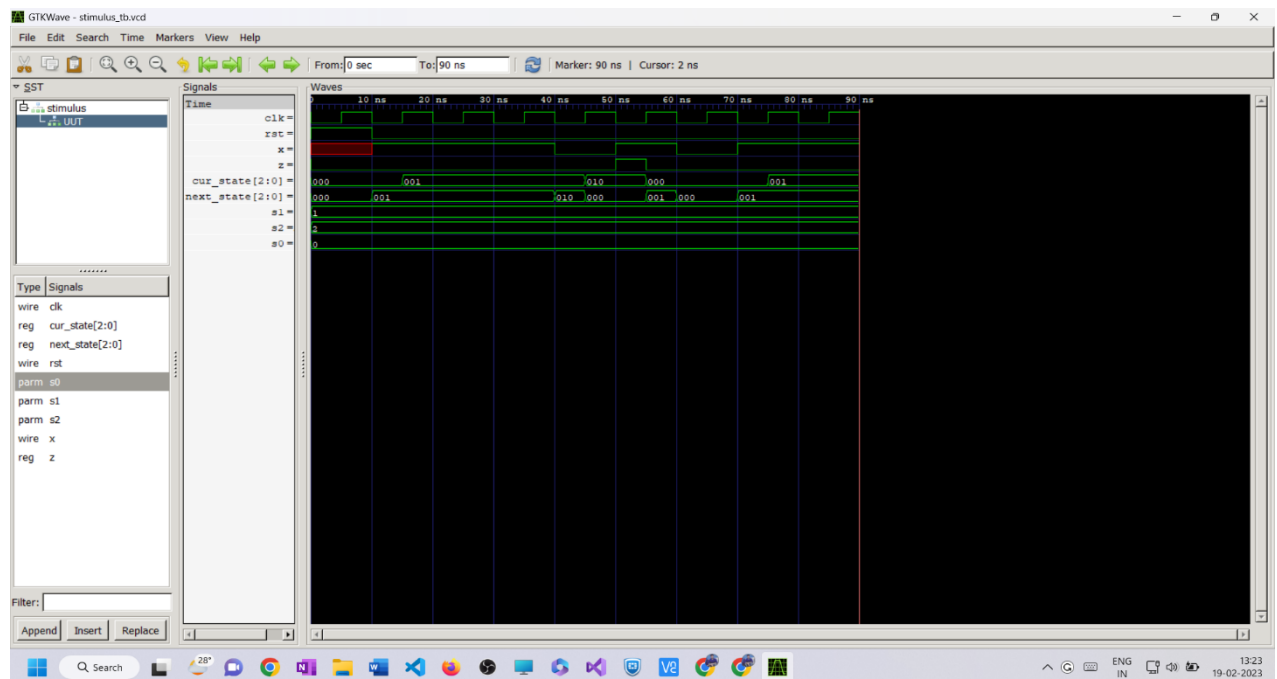
```
s0 : if(x)
    next_state<=s1;
    else
    next_state<=s0;
s1 : if(x)
    next_state<=s1;
    else
    next_state<=s2;
s2 : if(x)
    next_state<=s0;
    else
    next_state<=s0;
default : next_state<=s0;
endcase
end
always@(*)
begin
    case(cur_state)
    s0 : z=1'b0;
    s1 : z=1'b0;
    s2 : if(x)
        z=1'b1;
        else
        z=1'b0;
    default : z=1'b0;
    endcase
end
endmodule
```

(b) Write the testbench to test the Verilog model.

```
`timescale 1ns/1ns
`include "veri.v"
module stimulus;
reg clk;
reg rst;
reg x;
wire z;
top_module UUT(clk,rst,x,z);
initial
begin
    $dumpfile("stimulus_tb.vcd");
    $dumpvars;
end

initial
    clk=1'b0;
always
    #5 clk =~clk;
initial
begin
    rst=1'b1;
    #10 rst=1'b0;
    x=1'b1;
    #10 x=1'b1;
    #10 x=1'b1;
    #10 x=1'b0;
    #10 x=1'b1;
    #10 x=1'b0;
    #10 x=1'b1;
    rst=1'b0;
    #20 $finish;
end
initial
    $monitor($time," Output z= %d and input x=%d ",z,x);
endmodule
```

(c) Show the simulation output waveform and explain the correctness of the model.



So, at the time, $t=0$ clk starts with the time period of 10ns.

Firstly, the rst is 1 till $t=10$, then the rst is 0. So, the state can be changed from S0 to other states.

Now from the $X=1$ till 40ns. With the state diagram, the state change from 000(S0) to 001(S1) in the first posedge of clk, the next_state change to 001, and then curr_state change to 001.

Because for S1(001) to S2(010), the X needs to be 0. So, no state change from 20ns to 40ns.

From $t=40$ ns to $t=50$ ns, $X=0$, so the state change their according to the state diagram.

001(S1) to 010(S2) in 40ns to 50ns time frame, first next_state change to 010 then curr_state change to 010.

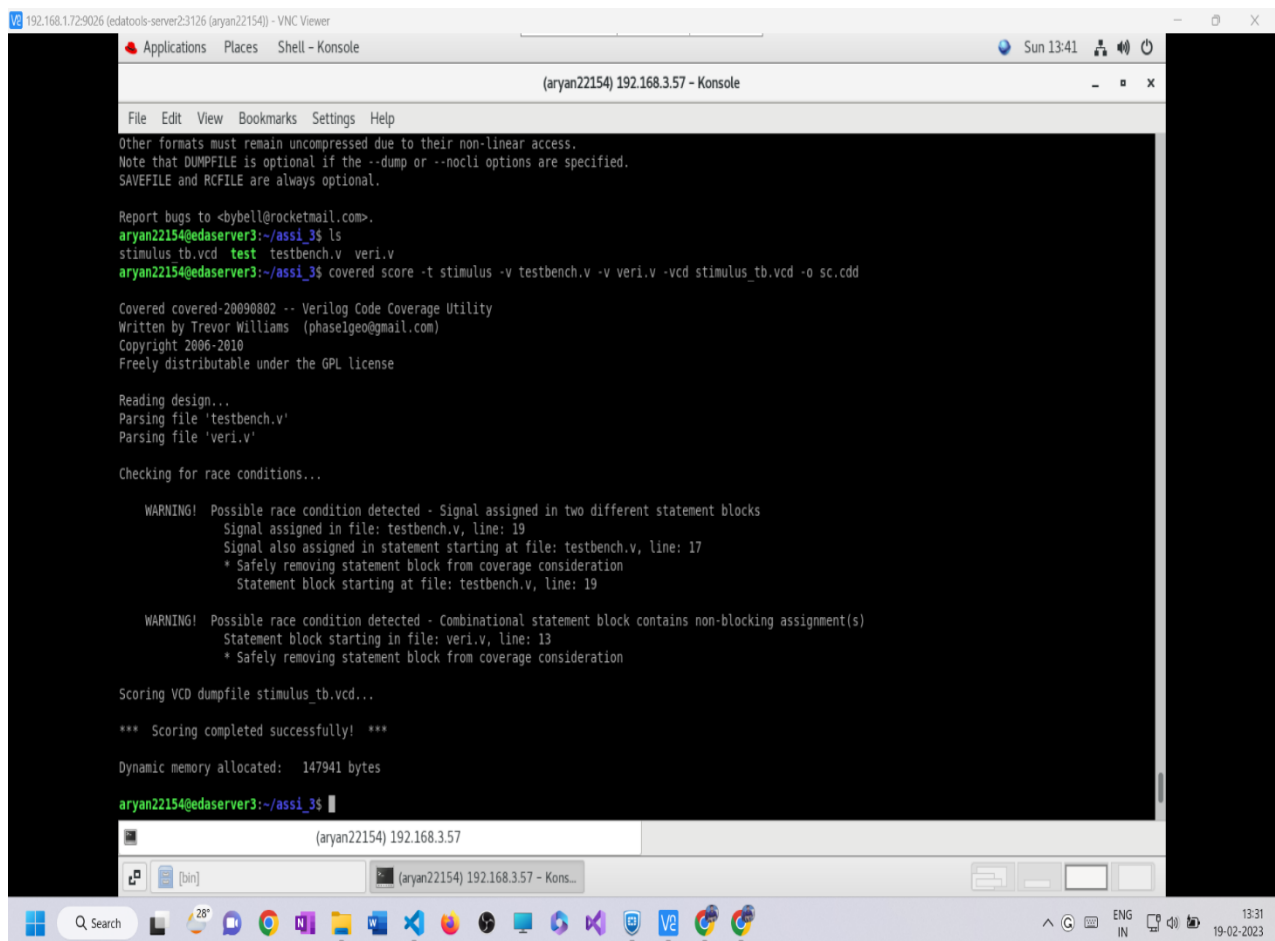
And after that whether $X=0$ or $X=1$, the state need to be changed(as per the state machine) so At $t=55$ ns the state change from 010(S2) to 000(S0).

But also, in that time frame the z is also define to change so the Z change from the 0 to 1 in this state.

So, Z=1 from 50ns to 55ns.

Like that it going to happen. The model is perfect, and it is also shown in the GTKWAVE.

(d) Show line coverage report for the above testbench. Explain the report.



```
192.168.172.9026 (edatools-server:23126 (aryan22154)) - VNC Viewer
(aryan22154) 192.168.3.57 - Konsole

File Edit View Bookmarks Settings Help

Other formats must remain uncompressed due to their non-linear access.
Note that DUMPFIL is optional if the --dump or --nocli options are specified.
SAVEFILE and RCFIL are always optional.

Report bugs to <bybell@rocketmail.com>.
aryan22154@edaserver3:~/assi_3$ ls
stimulus_tb.vcd test testbench.v veri.v
aryan22154@edaserver3:~/assi_3$ covered score -t stimulus -v testbench.v -v veri.v -vcd stimulus_tb.vcd -o sc.cdd

Covered covered-20090802 -- Verilog Code Coverage Utility
Written by Trevor Williams (phasegeo@gmail.com)
Copyright 2006-2010
Freely distributable under the GPL license

Reading design...
Parsing file 'testbench.v'
Parsing file 'veri.v'

Checking for race conditions...

WARNING! Possible race condition detected - Signal assigned in two different statement blocks
Signal assigned in file: testbench.v, line: 19
Signal also assigned in statement starting at file: testbench.v, line: 17
* Safely removing statement block from coverage consideration
Statement block starting at file: testbench.v, line: 19

WARNING! Possible race condition detected - Combinational statement block contains non-blocking assignment(s)
Statement block starting in file: veri.v, line: 13
* Safely removing statement block from coverage consideration

Scoring VCD dumpfile stimulus_tb.vcd...

*** Scoring completed successfully! ***

Dynamic memory allocated: 147941 bytes

aryan22154@edaserver3:~/assi_3$
```

192.168.172.9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer

Applications Places Shell - Konsole

Sun 13:52

(aryan22154) 192.168.3.57 - Konsole

File Edit View Bookmarks Settings Help

```
sc.cdd stimulus tb.vcd test testbench.v veri.v
aryan22154@edaserver3:~/assi_3$ covered report -d v sc.cdd
```

Covered covered-20090802 -- Verilog Code Coverage Utility
Written by Trevor Williams (phaseigeo@gmail.com)
Copyright 2006-2010
Freely distributable under the GPL license

```

:::
::
:: Covered -- Verilog Coverage Verbose Report ::
::
:::

```

----- GENERAL INFORMATION -----

* Report generated from CDD file : sc.cdd

* Reported by : Module

----- LINE COVERAGE RESULTS -----

Module/Task/Function	Filename	Hit/ Miss/Total	Percent hit
\$root	NA	0/ 0/ 0	100%
stimulus	testbench.v	10/ 0/ 10	100%
top_module	veri.v	10/ 1/ 11	91%
Accumulated		20/ 1/ 21	95%

Module: top_module, File: veri.v

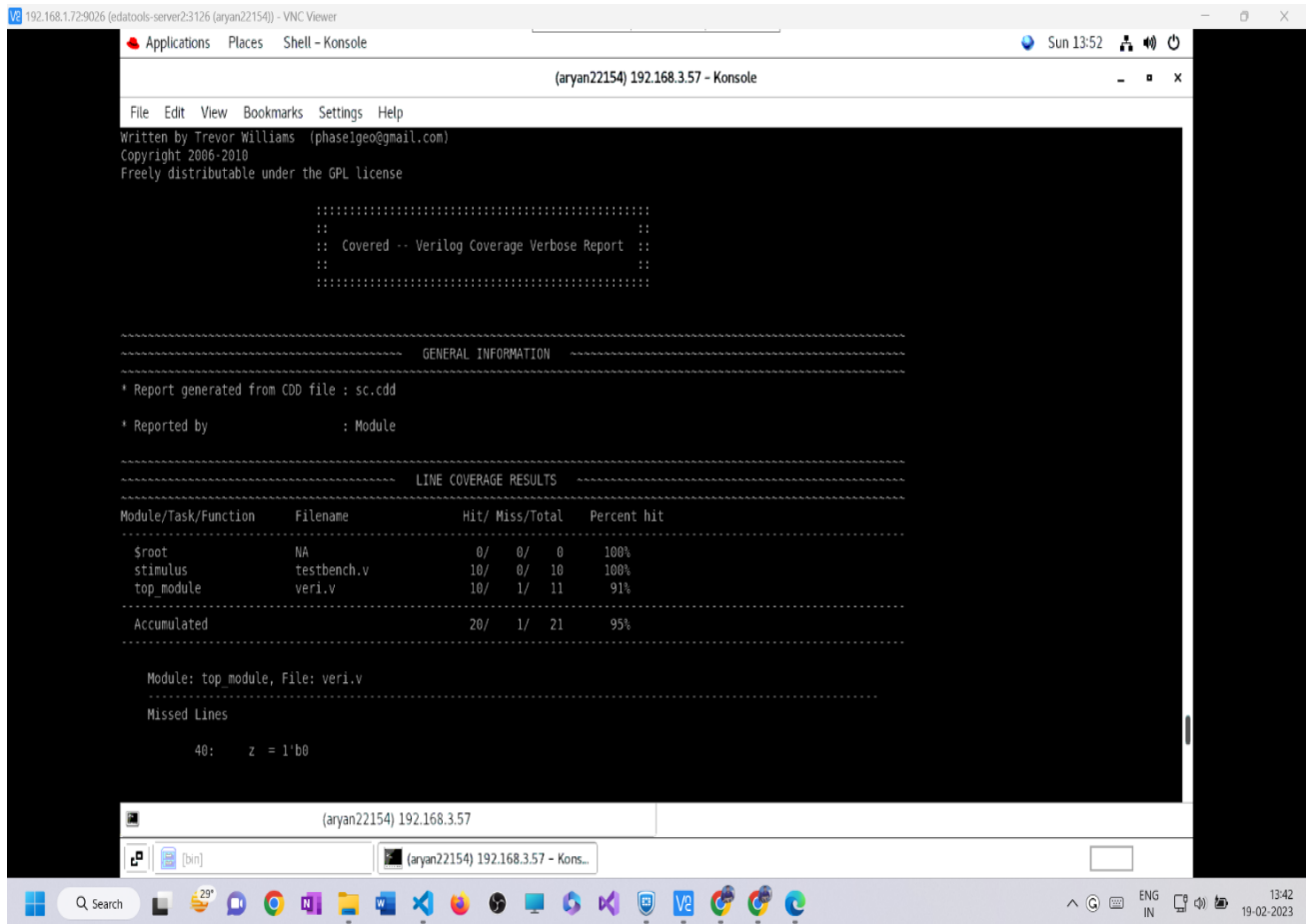
Missed Lines

(aryan22154) 192.168.3.57

[bin] (aryan22154) 192.168.3.57 - Kons...

13:42 19-02-2023

Line Coverage Result



The screenshot shows a VNC viewer window titled "192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer". Inside the viewer is a terminal window titled "(aryan22154) 192.168.3.57 - Konsole". The terminal displays the output of a Verilog coverage report. At the top, it says "File Edit View Bookmarks Settings Help" and "Written by Trevor Williams (phaselgeo@gmail.com) Copyright 2006-2010 Freely distributable under the GPL license". The report title is "Covered -- Verilog Coverage Verbose Report". It then shows "GENERAL INFORMATION" with details like "Report generated from CDD file : sc.cdd" and "Reported by : Module". The main section is "LINE COVERAGE RESULTS", which contains a table with columns: "Module/Task/Function", "Filename", "Hit/ Miss/Total", and "Percent hit". The table lists data for \$root, stimulus, top_module, and an accumulated total. Below the table, it shows "Module: top_module, File: veri.v" and "Missed Lines" with a specific line of code at 40: z = 1'b0.

```
File Edit View Bookmarks Settings Help
Written by Trevor Williams (phaselgeo@gmail.com)
Copyright 2006-2010
Freely distributable under the GPL license

:::
::
:: Covered -- Verilog Coverage Verbose Report ::
::
:::

-----
GENERAL INFORMATION
-----
* Report generated from CDD file : sc.cdd
* Reported by           : Module

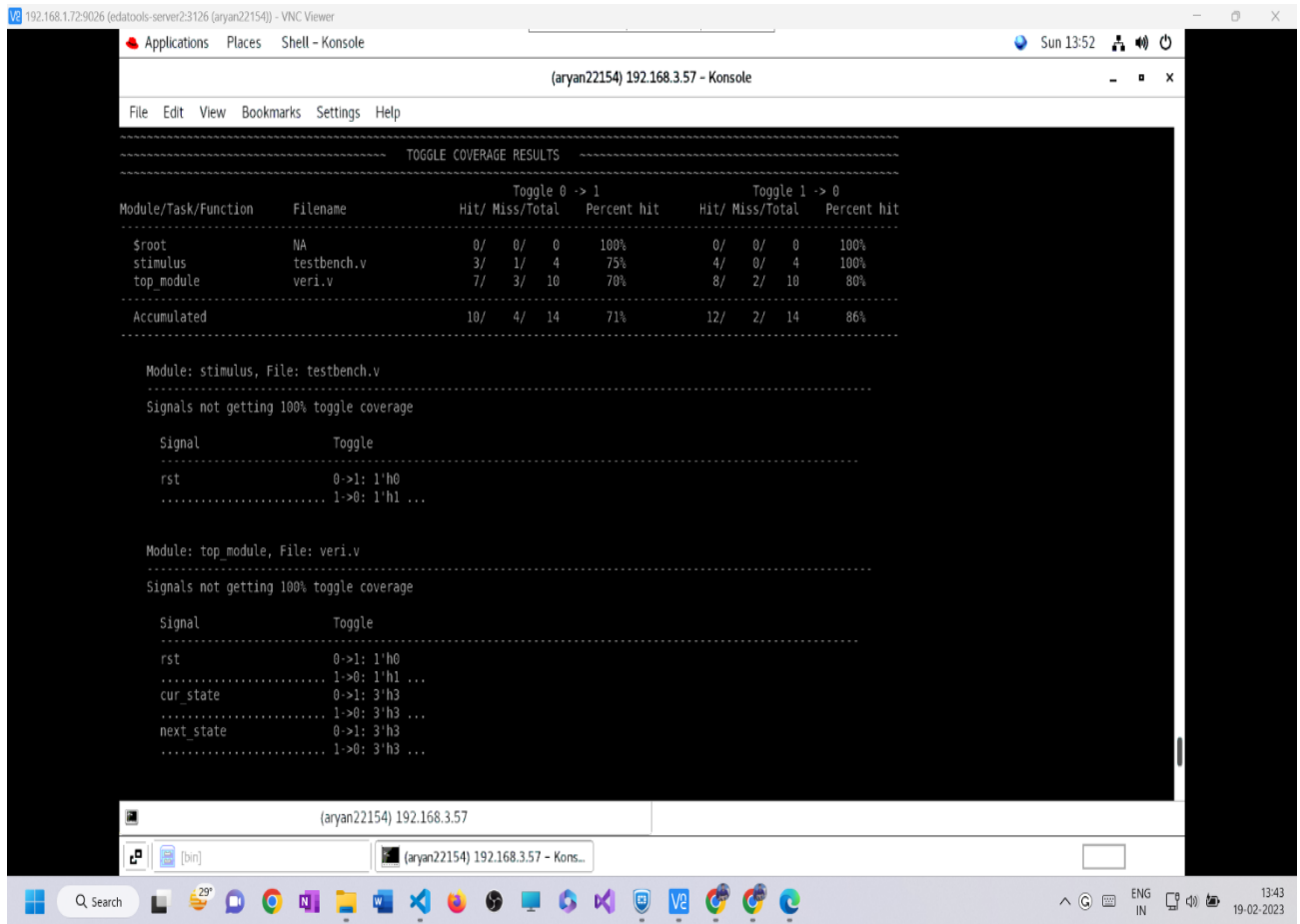
-----
LINE COVERAGE RESULTS
-----
Module/Task/Function  Filename      Hit/ Miss/Total  Percent hit
-----
$root                 NA            0/   0/   0      100%
stimulus              testbench.v   10/   0/  10      100%
top_module            veri.v        10/   1/  11       91%
-----
Accumulated           20/   1/  21       95%
-----

Module: top_module, File: veri.v
Missed Lines
40:      z = 1'b0
```

Line coverage refers to how many lines in my code were executed. The lines that are not executed are the else and the default line, so, in my case, there is one line at 40: `z=1'b0;`(which is the default case)

Overall, the Line Coverage result is 95%.

Toggle Coverage Result



```
192.168.1.72:9026 (edatools-server:2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help

=====
TOGGLE COVERAGE RESULTS
=====
Module/Task/Function  Filename      Toggle 0 -> 1  Percent hit  Toggle 1 -> 0  Percent hit
-----
$root                 NA            0/ 0/ 0      100%         0/ 0/ 0      100%
stimulus              testbench.v   3/ 1/ 4       75%          4/ 0/ 4      100%
top_module            veri.v        7/ 3/ 10      70%          8/ 2/ 10      80%
-----
Accumulated           10/ 4/ 14     71%          12/ 2/ 14     86%
=====

Module: stimulus, File: testbench.v
-----
Signals not getting 100% toggle coverage

Signal      Toggle
-----
rst          0->1: 1'h0
             1->0: 1'h1 ...

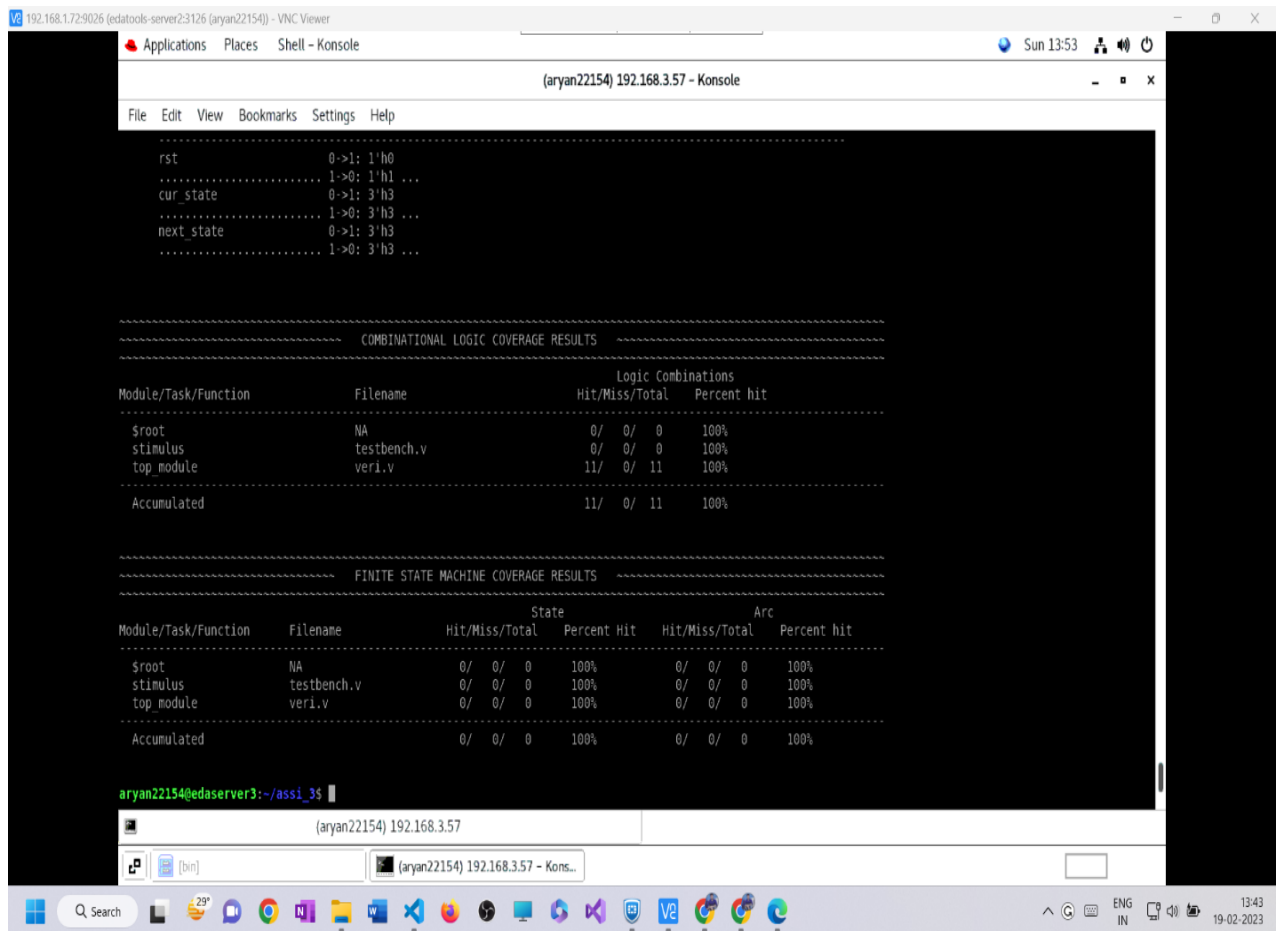
Module: top_module, File: veri.v
-----
Signals not getting 100% toggle coverage

Signal      Toggle
-----
rst          0->1: 1'h0
             1->0: 1'h1 ...
cur_state    0->1: 3'h3
             1->0: 3'h3 ...
next_state   0->1: 3'h3
             1->0: 3'h3 ...

(aryan22154) 192.168.3.57
[bin] (aryan22154) 192.168.3.57 - Kons...
```

Toggle coverage measures the unique values our expressions evaluated. So, basically, we don't need the toggle the rst. Due to this, my toggle coverage gets decreased but is highly good, which, is Overall around 85-90 %.

Combinational Logic Coverage Result



```
192.168.172.9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help

rst          0->1: 1'h0
.....    1->0: 1'h1 ...
cur_state    0->1: 3'h3
.....    1->0: 3'h3 ...
next_state   0->1: 3'h3
.....    1->0: 3'h3 ...

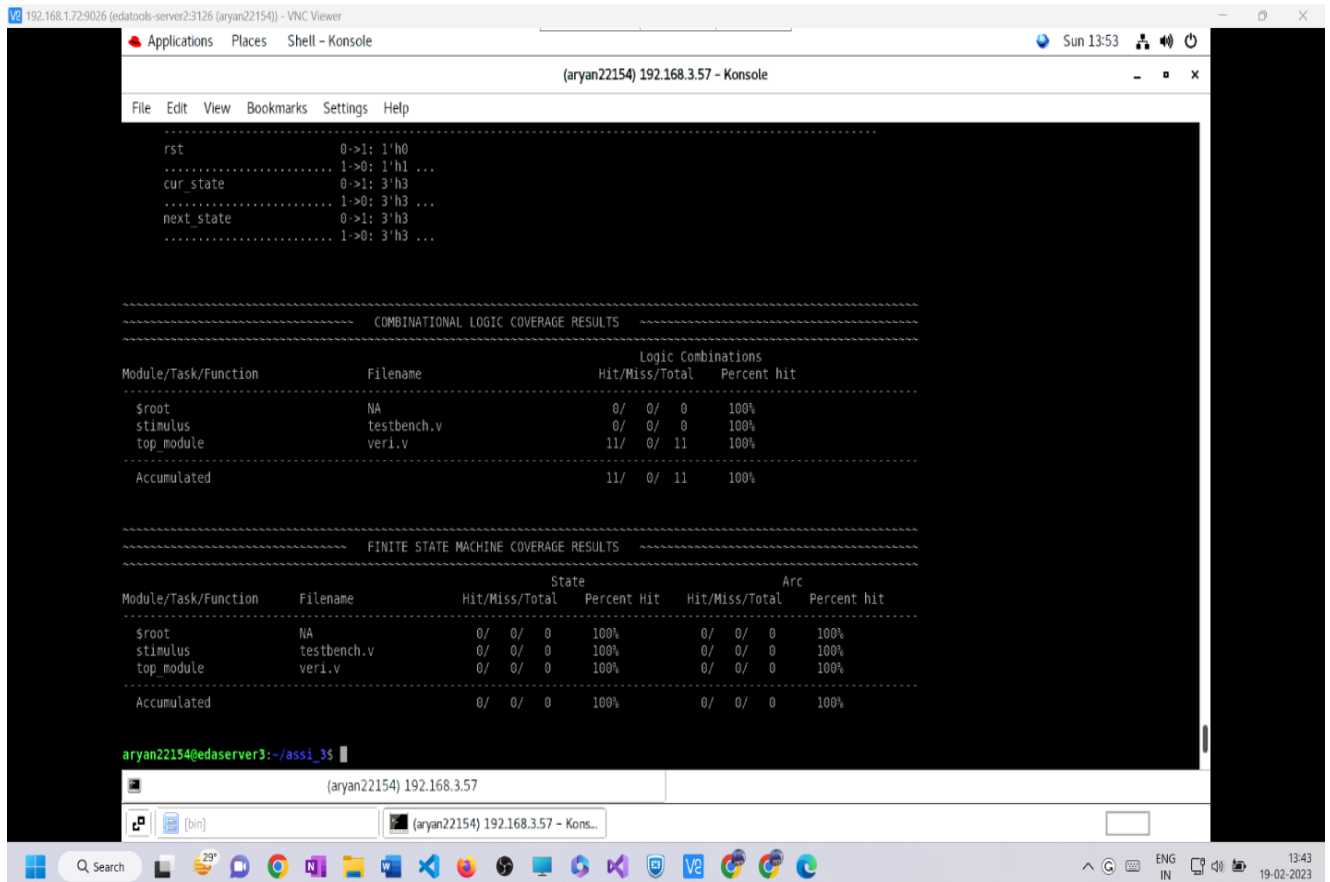
=====
COMBINATIONAL LOGIC COVERAGE RESULTS
=====
Module/Task/Function  Filename      Logic Combinations
                        Hit/Miss/Total  Percent hit
-----
sroot                 NA            0/ 0/ 0        100%
stimulus              testbench.v   0/ 0/ 0        100%
top_module            veri.v        11/ 0/ 11       100%
-----
Accumulated           11/ 0/ 11     100%

=====
FINITE STATE MACHINE COVERAGE RESULTS
=====
Module/Task/Function  Filename      State      Arc
                        Hit/Miss/Total  Percent Hit  Hit/Miss/Total  Percent hit
-----
sroot                 NA            0/ 0/ 0    100%      0/ 0/ 0        100%
stimulus              testbench.v   0/ 0/ 0    100%      0/ 0/ 0        100%
top_module            veri.v        0/ 0/ 0    100%      0/ 0/ 0        100%
-----
Accumulated           0/ 0/ 0      100%      0/ 0/ 0        100%

aryan22154@edaserver3:~/assi_3$
```

It measures the unique values our expressions evaluated. So in the test bench, all the unique values are evaluated whether x=1,0 or state 001,000,010, etc, or output z=0,1. So, I got 100% Combinational Logic Coverage.

Finite State Machine Coverage Result



```
192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help

rst ..... 0->1: 1'h0
1->0: 1'h1 ...
cur_state ..... 0->1: 3'h3
1->0: 3'h3 ...
next_state ..... 0->1: 3'h3
1->0: 3'h3 ...

=====
COMBINATIONAL LOGIC COVERAGE RESULTS
=====
Module/Task/Function  Filename      Logic Combinations
Hit/Miss/Total  Percent hit
-----
sroot              NA              0/ 0/ 0      100%
stimulus           testbench.v    0/ 0/ 0      100%
top_module         veri.v         11/ 0/ 11     100%
Accumulated                11/ 0/ 11     100%

=====
FINITE STATE MACHINE COVERAGE RESULTS
=====
Module/Task/Function  Filename      State      Arc
Hit/Miss/Total  Percent Hit  Hit/Miss/Total  Percent hit
-----
sroot              NA              0/ 0/ 0      100%      0/ 0/ 0      100%
stimulus           testbench.v    0/ 0/ 0      100%      0/ 0/ 0      100%
top_module         veri.v         0/ 0/ 0      100%      0/ 0/ 0      100%
Accumulated                0/ 0/ 0      100%      0/ 0/ 0      100%

aryan22154@edaserver3:~/assi_3$
```

This describes the robustness of my test bench. It answers if I have reached all possible states and taken all possible paths.

Since I my code all the states are reached so my Finite state machine coverage result is 100%.