California State University, Northridge

Department of Electrical & Computer Engineering

ECE 526L

Lab 8 Report

By

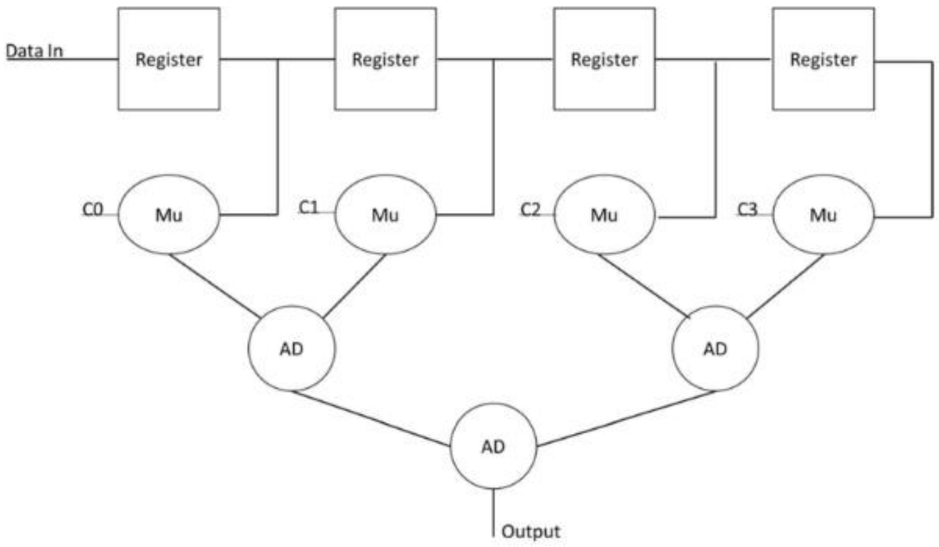
Avinash Damse

CSUN ID- 203131064

1: Introduction

The objective of this lab is to construct a Sum of Product circuit using three level hierarchy method (using scalable Register, Adder and Multiplier).

Here we have to use following diagram to build SOP.



2: Procedure

a. Part 1: Creating REG Module

In this lab I have created a REG module for scalable register. Inside the module I have assigned "data, clk and rst," as input variables and "q and qbar" as output variable. Then I wrote the scalable register logic inside module. After completing the code I ended the module using and saved the file with name "REG.v".

b. Part 2: Creating ADDR Module

In this lab I have created a scalable ADDR module. Inside the module I have assigned " A1 and A2," as input variables and "AD_out" as output variable. Then

I wrote the addition logic inside module. After completing the code I ended the module using and saved the file with name “ADDR.v”.

c. Part 3: Creating MULT Module

In this lab I have created a scalable MULT module. Inside the module I have assigned “ M and C,” as input variables and “out” as output variable. Then I wrote the multiplication logic inside module. After completing the code I ended the module using and saved the file with name “MULT.v”.

I have written the test bench for the REG module. We require test bench just to make sure that the module we have created is working properly. Here, in this testbench I have performed write to and read from operation and demonstrated an individual and block read.

d. Part 4: Creating non_hi_sop Module

In this lab I have created a non_hi_sop module. Inside the module I have assigned “ C0,C1,C2,C3, Data_in and clk,” as input variables and “Output” as output variable. Then I instantiated three module of each REG, ADDR and MULT to perform nonhierarchical sum of product . After completing the code I ended the module using and saved the file with name “non_hi_sop.v”.

e. Part 4: Creating non_exhstv Testbench

I have written the non-exhaustive test bench for the **non_hi_sop** module. We require test bench just to make sure that the module we have created is working properly. Here, in this testbench I have used small number of robust vectors to test strategy.

f. Part 4: Creating exhstv Testbench

I have written the exhaustive test bench for the **non_hi_sop** module. We require test bench just to make sure that the module we have created is working properly.

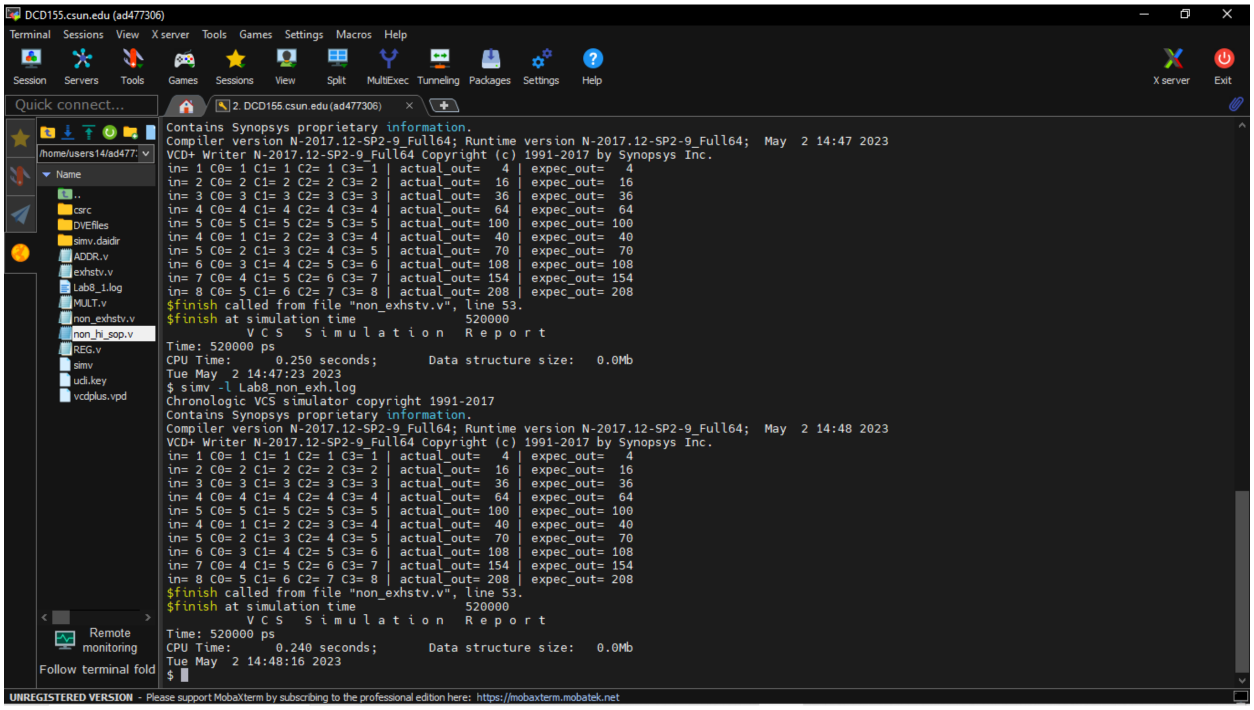
c. Part 3: execution.

Using “vcs -debug -full64 REG.v ADDR.v MULT.v non_hi_sop.v non_exhsv.b” command I executed first testbench file.
Again using “vcs -debug -full64 REG.v ADDR.v MULT.v non_hi_sop.v exhsv.b” command I executed second testbench file.

d. Part 4: Simulation

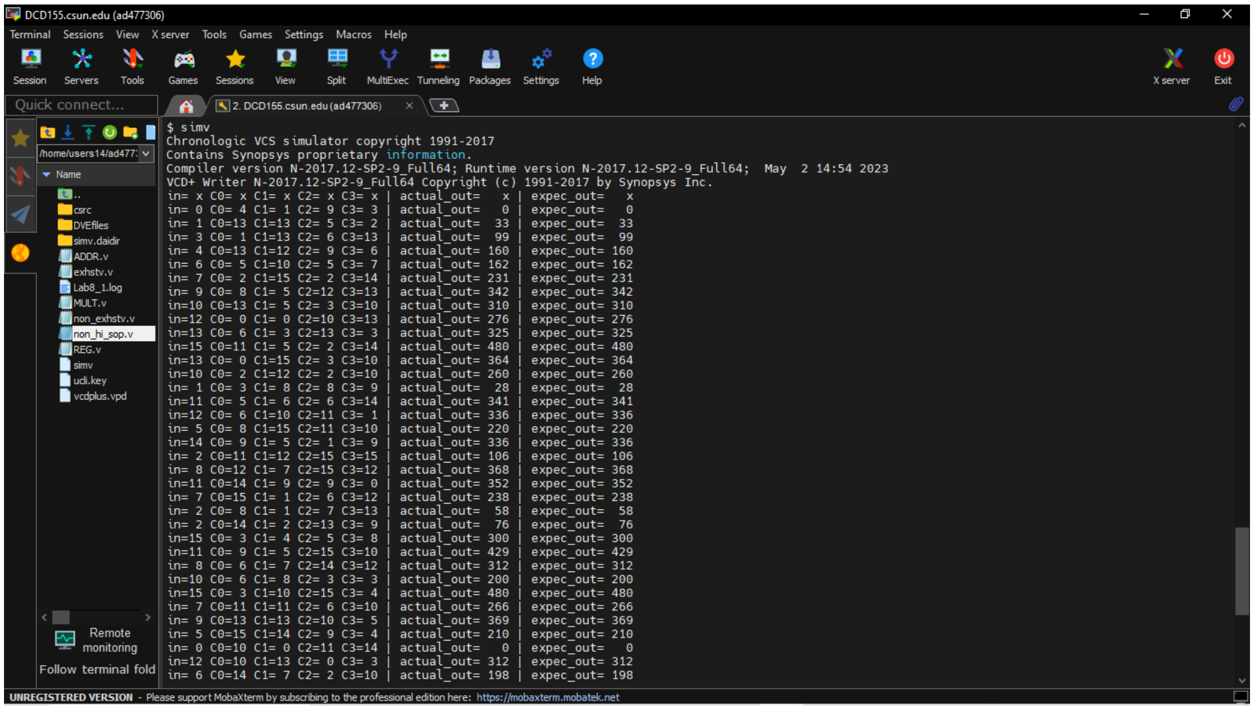
After an execution of all modules, I have run the command “simv” for simulation.

Simulation for Non-Exhaustive Test Strategy :

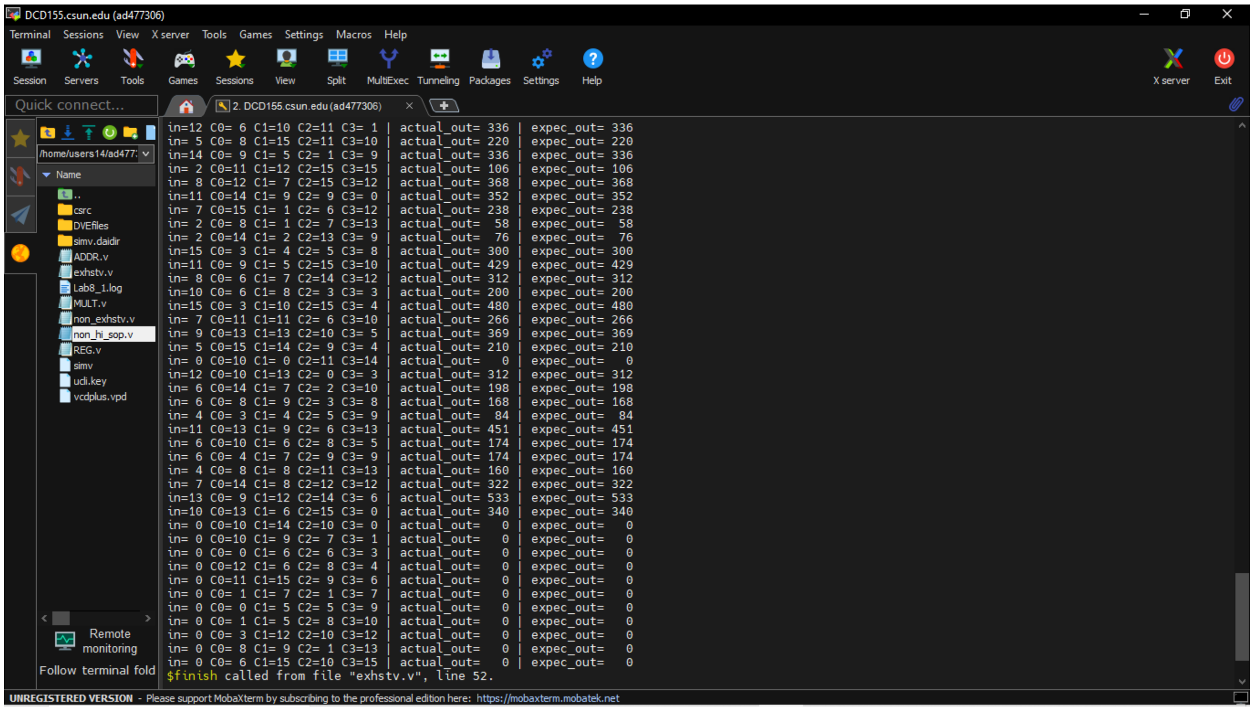

DCD155.csun.edu (ad477306)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
2. DCD155.csun.edu (ad477306)
/home/users14/ad477
Name
..
csrc
DVEfiles
simv.daidir
ADDR.v
exhstv.v
Lab8_1.log
MULT.v
non_exhstv.v
non_hi_sop.v
REG.v
simv
ucli.key
vcdplus.vpd
Remote monitoring
Follow terminal fold
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-9_Full64; Runtime version N-2017.12-SP2-9_Full64;  May  2 14:47 2023
VCD+ Writer N-2017.12-SP2-9_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
in= 1 C0= 1 C1= 1 C2= 1 C3= 1 | actual_out=   4 | expec_out=   4
in= 2 C0= 2 C1= 2 C2= 2 C3= 2 | actual_out=  16 | expec_out=  16
in= 3 C0= 3 C1= 3 C2= 3 C3= 3 | actual_out=  36 | expec_out=  36
in= 4 C0= 4 C1= 4 C2= 4 C3= 4 | actual_out=  64 | expec_out=  64
in= 5 C0= 5 C1= 5 C2= 5 C3= 5 | actual_out= 100 | expec_out= 100
in= 4 C0= 1 C1= 2 C2= 3 C3= 4 | actual_out=  40 | expec_out=  40
in= 5 C0= 2 C1= 3 C2= 4 C3= 5 | actual_out=  70 | expec_out=  70
in= 6 C0= 3 C1= 4 C2= 5 C3= 6 | actual_out= 108 | expec_out= 108
in= 7 C0= 4 C1= 5 C2= 6 C3= 7 | actual_out= 154 | expec_out= 154
in= 8 C0= 5 C1= 6 C2= 7 C3= 8 | actual_out= 208 | expec_out= 208
$finish called from file "non_exhstv.v", line 53.
$finish at simulation time                520000
           V C S   S i m u l a t i o n   R e p o r t
Time: 520000 ps
CPU Time:      0.250 seconds;       Data structure size:   0.0Mb
Tue May  2 14:47:23 2023
$ simv -l Lab8_non_exh.log
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-9_Full64; Runtime version N-2017.12-SP2-9_Full64;  May  2 14:48 2023
VCD+ Writer N-2017.12-SP2-9_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
in= 1 C0= 1 C1= 1 C2= 1 C3= 1 | actual_out=   4 | expec_out=   4
in= 2 C0= 2 C1= 2 C2= 2 C3= 2 | actual_out=  16 | expec_out=  16
in= 3 C0= 3 C1= 3 C2= 3 C3= 3 | actual_out=  36 | expec_out=  36
in= 4 C0= 4 C1= 4 C2= 4 C3= 4 | actual_out=  64 | expec_out=  64
in= 5 C0= 5 C1= 5 C2= 5 C3= 5 | actual_out= 100 | expec_out= 100
in= 4 C0= 1 C1= 2 C2= 3 C3= 4 | actual_out=  40 | expec_out=  40
in= 5 C0= 2 C1= 3 C2= 4 C3= 5 | actual_out=  70 | expec_out=  70
in= 6 C0= 3 C1= 4 C2= 5 C3= 6 | actual_out= 108 | expec_out= 108
in= 7 C0= 4 C1= 5 C2= 6 C3= 7 | actual_out= 154 | expec_out= 154
in= 8 C0= 5 C1= 6 C2= 7 C3= 8 | actual_out= 208 | expec_out= 208
$finish called from file "non_exhstv.v", line 53.
$finish at simulation time                520000
           V C S   S i m u l a t i o n   R e p o r t
Time: 520000 ps
CPU Time:      0.240 seconds;       Data structure size:   0.0Mb
Tue May  2 14:48:16 2023
$
```

Simulation for Exhaustive Test Strategy :



```
$ simv
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-9_Full64; Runtime version N-2017.12-SP2-9_Full64;  May  2 14:54 2023
VCD+ Writer N-2017.12-SP2-9_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
in= x C0= x C1= x C2= x C3= x | actual_out=   x | expec_out=   x
in= 0 C0= 4 C1= 1 C2= 9 C3= 3 | actual_out=   0 | expec_out=   0
in= 1 C0=13 C1=13 C2= 5 C3= 2 | actual_out=  33 | expec_out=  33
in= 3 C0= 1 C1=13 C2= 6 C3=13 | actual_out=  99 | expec_out=  99
in= 4 C0=13 C1=12 C2= 9 C3= 6 | actual_out= 160 | expec_out= 160
in= 6 C0= 5 C1=10 C2= 5 C3= 7 | actual_out= 162 | expec_out= 162
in= 7 C0= 2 C1=15 C2= 2 C3=14 | actual_out= 231 | expec_out= 231
in= 9 C0= 8 C1= 5 C2=12 C3=13 | actual_out= 342 | expec_out= 342
in=10 C0=13 C1= 5 C2= 3 C3=10 | actual_out= 310 | expec_out= 310
in=12 C0= 0 C1= 0 C2=10 C3=13 | actual_out= 276 | expec_out= 276
in=13 C0= 6 C1= 3 C2=13 C3= 3 | actual_out= 325 | expec_out= 325
in=15 C0=11 C1= 5 C2= 2 C3=14 | actual_out= 480 | expec_out= 480
in=13 C0= 0 C1=15 C2= 3 C3=10 | actual_out= 364 | expec_out= 364
in=10 C0= 2 C1=12 C2= 2 C3=10 | actual_out= 260 | expec_out= 260
in= 1 C0= 3 C1= 8 C2= 8 C3= 9 | actual_out=  28 | expec_out=  28
in=11 C0= 5 C1= 6 C2= 6 C3=14 | actual_out= 341 | expec_out= 341
in=12 C0= 6 C1=10 C2=11 C3= 1 | actual_out= 336 | expec_out= 336
in= 5 C0= 8 C1=15 C2=11 C3=10 | actual_out= 220 | expec_out= 220
in=14 C0= 9 C1= 5 C2= 1 C3= 9 | actual_out= 336 | expec_out= 336
in= 2 C0=11 C1=12 C2=15 C3=15 | actual_out= 106 | expec_out= 106
in= 8 C0=12 C1= 7 C2=15 C3=12 | actual_out= 368 | expec_out= 368
in=11 C0=14 C1= 9 C2= 9 C3= 0 | actual_out= 352 | expec_out= 352
in= 7 C0=15 C1= 1 C2= 6 C3=12 | actual_out= 238 | expec_out= 238
in= 2 C0= 8 C1= 1 C2= 7 C3=13 | actual_out=  58 | expec_out=  58
in= 2 C0=14 C1= 2 C2=13 C3= 9 | actual_out=  76 | expec_out=  76
in=15 C0= 3 C1= 4 C2= 5 C3= 8 | actual_out= 300 | expec_out= 300
in=11 C0= 9 C1= 5 C2=15 C3=10 | actual_out= 429 | expec_out= 429
in= 8 C0= 6 C1= 7 C2=14 C3=12 | actual_out= 312 | expec_out= 312
in=10 C0= 6 C1= 8 C2= 3 C3= 3 | actual_out= 200 | expec_out= 200
in=15 C0= 3 C1=10 C2=15 C3= 4 | actual_out= 480 | expec_out= 480
in= 7 C0=11 C1=11 C2= 6 C3=10 | actual_out= 266 | expec_out= 266
in= 9 C0=13 C1=13 C2=10 C3= 5 | actual_out= 369 | expec_out= 369
in= 5 C0=15 C1=14 C2= 9 C3= 4 | actual_out= 210 | expec_out= 210
in= 0 C0=10 C1= 0 C2=11 C3=14 | actual_out=   0 | expec_out=   0
in=12 C0=10 C1=13 C2= 0 C3= 3 | actual_out= 312 | expec_out= 312
in= 6 C0=14 C1= 7 C2= 2 C3=10 | actual_out= 198 | expec_out= 198
```



```
in=12 C0= 6 C1=10 C2=11 C3= 1 | actual_out= 336 | expec_out= 336
in= 5 C0= 8 C1=15 C2=11 C3=10 | actual_out= 220 | expec_out= 220
in=14 C0= 9 C1= 5 C2= 1 C3= 9 | actual_out= 336 | expec_out= 336
in= 2 C0=11 C1=12 C2=15 C3=15 | actual_out= 106 | expec_out= 106
in= 8 C0=12 C1= 7 C2=15 C3=12 | actual_out= 368 | expec_out= 368
in=11 C0=14 C1= 9 C2= 9 C3= 0 | actual_out= 352 | expec_out= 352
in= 7 C0=15 C1= 1 C2= 6 C3=12 | actual_out= 238 | expec_out= 238
in= 2 C0= 8 C1= 1 C2= 7 C3=13 | actual_out=  58 | expec_out=  58
in= 2 C0=14 C1= 2 C2=13 C3= 9 | actual_out=  76 | expec_out=  76
in=15 C0= 3 C1= 4 C2= 5 C3= 8 | actual_out= 300 | expec_out= 300
in=11 C0= 9 C1= 5 C2=15 C3=10 | actual_out= 429 | expec_out= 429
in= 8 C0= 6 C1= 7 C2=14 C3=12 | actual_out= 312 | expec_out= 312
in=10 C0= 6 C1= 8 C2= 3 C3= 3 | actual_out= 200 | expec_out= 200
in=15 C0= 3 C1=10 C2=15 C3= 4 | actual_out= 480 | expec_out= 480
in= 7 C0=11 C1=11 C2= 6 C3=10 | actual_out= 266 | expec_out= 266
in= 9 C0=13 C1=13 C2=10 C3= 5 | actual_out= 369 | expec_out= 369
in= 5 C0=15 C1=14 C2= 9 C3= 4 | actual_out= 210 | expec_out= 210
in= 0 C0=10 C1= 0 C2=11 C3=14 | actual_out=   0 | expec_out=   0
in=12 C0=10 C1=13 C2= 0 C3= 3 | actual_out= 312 | expec_out= 312
in= 6 C0=14 C1= 7 C2= 2 C3=10 | actual_out= 198 | expec_out= 198
in= 6 C0= 8 C1= 9 C2= 3 C3= 8 | actual_out= 168 | expec_out= 168
in= 4 C0= 3 C1= 4 C2= 5 C3= 9 | actual_out=  84 | expec_out=  84
in=11 C0=13 C1= 9 C2= 6 C3=13 | actual_out= 451 | expec_out= 451
in= 6 C0=10 C1= 6 C2= 8 C3= 5 | actual_out= 174 | expec_out= 174
in= 6 C0= 4 C1= 7 C2= 9 C3= 9 | actual_out= 174 | expec_out= 174
in= 4 C0= 8 C1= 8 C2=11 C3=13 | actual_out= 160 | expec_out= 160
in= 7 C0=14 C1= 8 C2=12 C3=12 | actual_out= 322 | expec_out= 322
in=13 C0= 9 C1=12 C2=14 C3= 6 | actual_out= 533 | expec_out= 533
in=10 C0=13 C1= 6 C2=15 C3= 0 | actual_out= 340 | expec_out= 340
in= 0 C0=10 C1=14 C2=10 C3= 0 | actual_out=   0 | expec_out=   0
in= 0 C0=10 C1= 9 C2= 7 C3= 1 | actual_out=   0 | expec_out=   0
in= 0 C0= 0 C1= 6 C2= 6 C3= 3 | actual_out=   0 | expec_out=   0
in= 0 C0=12 C1= 6 C2= 8 C3= 4 | actual_out=   0 | expec_out=   0
in= 0 C0=11 C1=15 C2= 9 C3= 6 | actual_out=   0 | expec_out=   0
in= 0 C0= 1 C1= 7 C2= 1 C3= 7 | actual_out=   0 | expec_out=   0
in= 0 C0= 0 C1= 5 C2= 5 C3= 9 | actual_out=   0 | expec_out=   0
in= 0 C0= 1 C1= 5 C2= 8 C3=10 | actual_out=   0 | expec_out=   0
in= 0 C0= 3 C1=12 C2=10 C3=12 | actual_out=   0 | expec_out=   0
in= 0 C0= 8 C1= 9 C2= 1 C3=13 | actual_out=   0 | expec_out=   0
in= 0 C0= 6 C1=15 C2=10 C3=15 | actual_out=   0 | expec_out=   0
$finish called from file "exhstv.v", line 52.
```

e. Part 5: Creating Log File

After running the simulation I created the log file for first stategy using the “simv -l Lab8_non_exh.log” command

```
Command: /home/users14/ad477306/Verilog/Lab8/./simv -l Lab8_non_exh.log
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-9_Full64; Runtime version N-2017.12-SP2-9_Full64;  May  2
14:48 2023
VCD+ Writer N-2017.12-SP2-9_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
in= 1 C0= 1 C1= 1 C2= 1 C3= 1 | actual_out=   4 | expec_out=   4
in= 2 C0= 2 C1= 2 C2= 2 C3= 2 | actual_out=  16 | expec_out=  16
in= 3 C0= 3 C1= 3 C2= 3 C3= 3 | actual_out=  36 | expec_out=  36
in= 4 C0= 4 C1= 4 C2= 4 C3= 4 | actual_out=  64 | expec_out=  64
in= 5 C0= 5 C1= 5 C2= 5 C3= 5 | actual_out= 100 | expec_out= 100
in= 4 C0= 1 C1= 2 C2= 3 C3= 4 | actual_out=  40 | expec_out=  40
in= 5 C0= 2 C1= 3 C2= 4 C3= 5 | actual_out=  70 | expec_out=  70
in= 6 C0= 3 C1= 4 C2= 5 C3= 6 | actual_out= 108 | expec_out= 108
in= 7 C0= 4 C1= 5 C2= 6 C3= 7 | actual_out= 154 | expec_out= 154
in= 8 C0= 5 C1= 6 C2= 7 C3= 8 | actual_out= 208 | expec_out= 208
$finish called from file "non_exhstv.v", line 53.
$finish at simulation time             520000
           V C S   S i m u l a t i o n   R e p o r t
Time: 520000 ps
CPU Time:      0.240 seconds;       Data structure size:   0.0Mb
Tue May  2 14:48:16 2023
```

Log for exhaustive strategy:

```
Command: /home/users14/ad477306/Verilog/Lab8/./simv -l Lab8_exh.log
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-9_Full64; Runtime version N-2017.12-SP2-9_Full64;  May  2
15:04 2023
VCD+ Writer N-2017.12-SP2-9_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
in= x C0= x C1= x C2= x C3= x | actual_out=   x | expec_out=   x
in= 0 C0= 4 C1= 1 C2= 9 C3= 3 | actual_out=   0 | expec_out=   0
in= 1 C0=13 C1=13 C2= 5 C3= 2 | actual_out=  33 | expec_out=  33
```

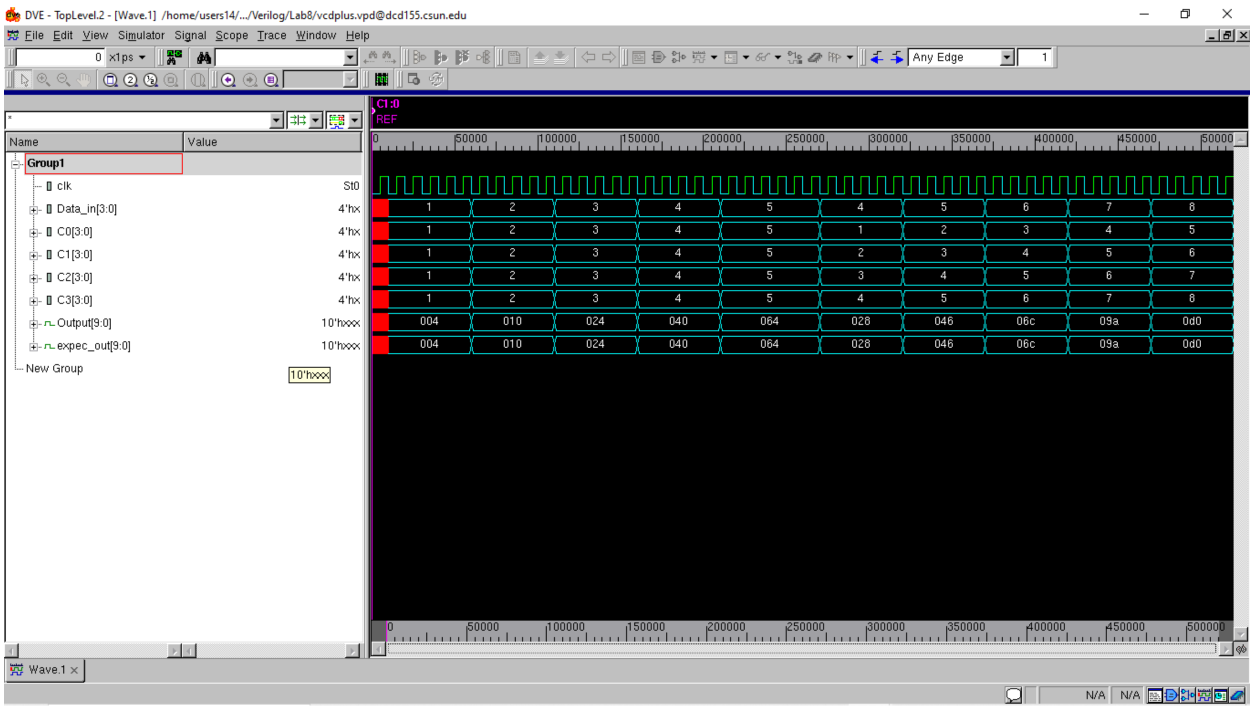
```
in= 3 C0= 1 C1=13 C2= 6 C3=13 | actual_out=  99 | expec_out=  99
in= 4 C0=13 C1=12 C2= 9 C3= 6 | actual_out= 160 | expec_out= 160
in= 6 C0= 5 C1=10 C2= 5 C3= 7 | actual_out= 162 | expec_out= 162
in= 7 C0= 2 C1=15 C2= 2 C3=14 | actual_out= 231 | expec_out= 231
in= 9 C0= 8 C1= 5 C2=12 C3=13 | actual_out= 342 | expec_out= 342
in=10 C0=13 C1= 5 C2= 3 C3=10 | actual_out= 310 | expec_out= 310
in=12 C0= 0 C1= 0 C2=10 C3=13 | actual_out= 276 | expec_out= 276
in=13 C0= 6 C1= 3 C2=13 C3= 3 | actual_out= 325 | expec_out= 325
in=15 C0=11 C1= 5 C2= 2 C3=14 | actual_out= 480 | expec_out= 480
in=13 C0= 0 C1=15 C2= 3 C3=10 | actual_out= 364 | expec_out= 364
in=10 C0= 2 C1=12 C2= 2 C3=10 | actual_out= 260 | expec_out= 260
in= 1 C0= 3 C1= 8 C2= 8 C3= 9 | actual_out=  28 | expec_out=  28
in=11 C0= 5 C1= 6 C2= 6 C3=14 | actual_out= 341 | expec_out= 341
in=12 C0= 6 C1=10 C2=11 C3= 1 | actual_out= 336 | expec_out= 336
in= 5 C0= 8 C1=15 C2=11 C3=10 | actual_out= 220 | expec_out= 220
in=14 C0= 9 C1= 5 C2= 1 C3= 9 | actual_out= 336 | expec_out= 336
in= 2 C0=11 C1=12 C2=15 C3=15 | actual_out= 106 | expec_out= 106
in= 8 C0=12 C1= 7 C2=15 C3=12 | actual_out= 368 | expec_out= 368
in=11 C0=14 C1= 9 C2= 9 C3= 0 | actual_out= 352 | expec_out= 352
in= 7 C0=15 C1= 1 C2= 6 C3=12 | actual_out= 238 | expec_out= 238
in= 2 C0= 8 C1= 1 C2= 7 C3=13 | actual_out=  58 | expec_out=  58
in= 2 C0=14 C1= 2 C2=13 C3= 9 | actual_out=  76 | expec_out=  76
in=15 C0= 3 C1= 4 C2= 5 C3= 8 | actual_out= 300 | expec_out= 300
in=11 C0= 9 C1= 5 C2=15 C3=10 | actual_out= 429 | expec_out= 429
in= 8 C0= 6 C1= 7 C2=14 C3=12 | actual_out= 312 | expec_out= 312
in=10 C0= 6 C1= 8 C2= 3 C3= 3 | actual_out= 200 | expec_out= 200
in=15 C0= 3 C1=10 C2=15 C3= 4 | actual_out= 480 | expec_out= 480
in= 7 C0=11 C1=11 C2= 6 C3=10 | actual_out= 266 | expec_out= 266
in= 9 C0=13 C1=13 C2=10 C3= 5 | actual_out= 369 | expec_out= 369
in= 5 C0=15 C1=14 C2= 9 C3= 4 | actual_out= 210 | expec_out= 210
in= 0 C0=10 C1= 0 C2=11 C3=14 | actual_out=   0 | expec_out=   0
in=12 C0=10 C1=13 C2= 0 C3= 3 | actual_out= 312 | expec_out= 312
in= 6 C0=14 C1= 7 C2= 2 C3=10 | actual_out= 198 | expec_out= 198
in= 6 C0= 8 C1= 9 C2= 3 C3= 8 | actual_out= 168 | expec_out= 168
in= 4 C0= 3 C1= 4 C2= 5 C3= 9 | actual_out=  84 | expec_out=  84
in=11 C0=13 C1= 9 C2= 6 C3=13 | actual_out= 451 | expec_out= 451
in= 6 C0=10 C1= 6 C2= 8 C3= 5 | actual_out= 174 | expec_out= 174
in= 6 C0= 4 C1= 7 C2= 9 C3= 9 | actual_out= 174 | expec_out= 174
in= 4 C0= 8 C1= 8 C2=11 C3=13 | actual_out= 160 | expec_out= 160
in= 7 C0=14 C1= 8 C2=12 C3=12 | actual_out= 322 | expec_out= 322
in=13 C0= 9 C1=12 C2=14 C3= 6 | actual_out= 533 | expec_out= 533
in=10 C0=13 C1= 6 C2=15 C3= 0 | actual_out= 340 | expec_out= 340
in= 0 C0=10 C1=14 C2=10 C3= 0 | actual_out=   0 | expec_out=   0
in= 0 C0=10 C1= 9 C2= 7 C3= 1 | actual_out=   0 | expec_out=   0
```

```
in= 0 C0= 0 C1= 6 C2= 6 C3= 3 | actual_out=   0 | expec_out=   0
in= 0 C0=12 C1= 6 C2= 8 C3= 4 | actual_out=   0 | expec_out=   0
in= 0 C0=11 C1=15 C2= 9 C3= 6 | actual_out=   0 | expec_out=   0
in= 0 C0= 1 C1= 7 C2= 1 C3= 7 | actual_out=   0 | expec_out=   0
in= 0 C0= 0 C1= 5 C2= 5 C3= 9 | actual_out=   0 | expec_out=   0
in= 0 C0= 1 C1= 5 C2= 8 C3=10 | actual_out=   0 | expec_out=   0
in= 0 C0= 3 C1=12 C2=10 C3=12 | actual_out=   0 | expec_out=   0
in= 0 C0= 8 C1= 9 C2= 1 C3=13 | actual_out=   0 | expec_out=   0
in= 0 C0= 6 C1=15 C2=10 C3=15 | actual_out=   0 | expec_out=   0
$finish called from file "exhstv.v", line 52.
$finish at simulation time            2770000
         V C S   S i m u l a t i o n   R e p o r t
Time: 2770000 ps
CPU Time:      0.240 seconds;       Data structure size:   0.0Mb
Tue May  2 15:04:09 2023
```

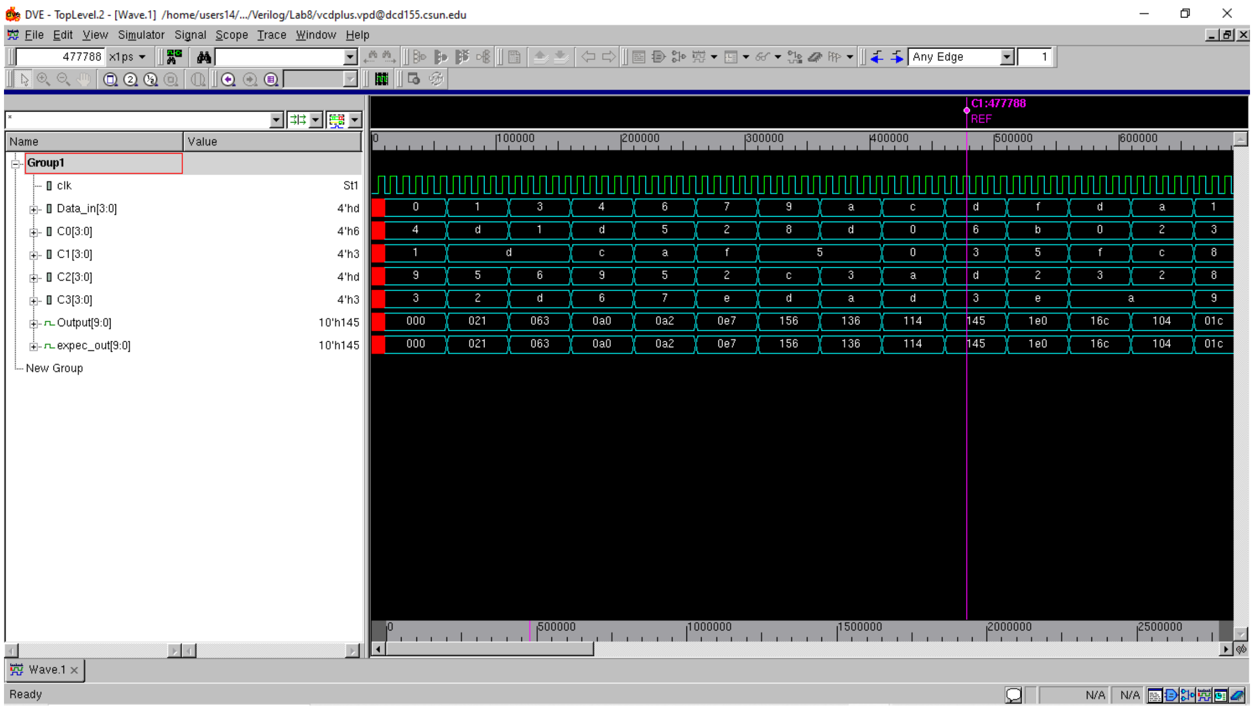
f. Part 6: Seeing the waveform.

After creating the log file I opened the DVE using “dve -full64 &” command to see the waveforms.

Non_Exhaustive:



Exhaustive:



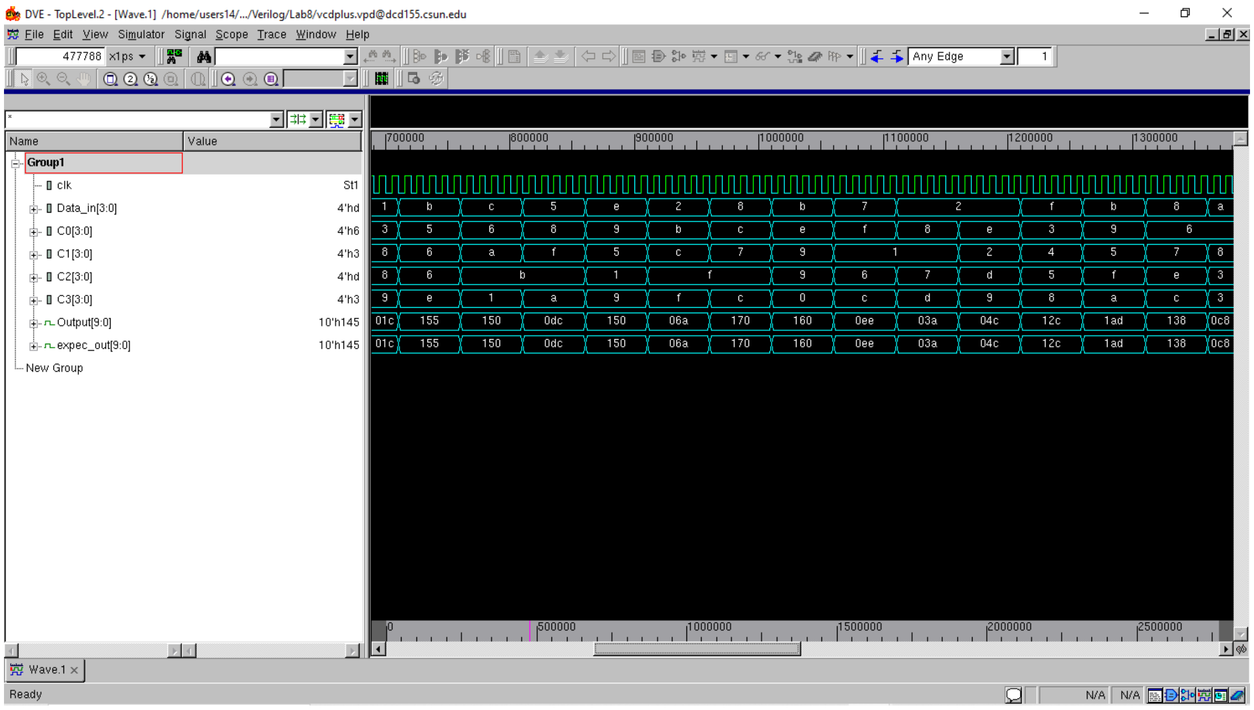
DVE - TopLevel.2 - [Wave.1] /home/users14/.../Verilog/Lab8/vcdplus.vpd@dcd155.csun.edu

```
Name              Value
Group1
  clk             St1
  Data_in[3:0]    4'hd
  C0[3:0]         4'h6
  C1[3:0]         4'h3
  C2[3:0]         4'hd
  C3[3:0]         4'h3
  Output[9:0]     10'h145
  expec_out[9:0]  10'h145
New Group
```

C1:477788 REF

```
Data_in:   0    1    3    4    6    7    9    a    c    d    f    d    a    1
C0:        4    d    1    d    5    2    8    d    0    6    b    0    2    3
C1:        1    d         c    a    f    5         0    3    5    f    c    8
C2:        9    5    6    9    5    2    c    3    a    d    2    3    2    8
C3:        3    2    d    6    7    e    d    a    d    3    e    a         9
Output:    000  021  063  0a0  0a2  0e7  156  136  114  145  1e0  16c  104  01c
expec_out: 000  021  063  0a0  0a2  0e7  156  136  114  145  1e0  16c  104  01c
```

Ready

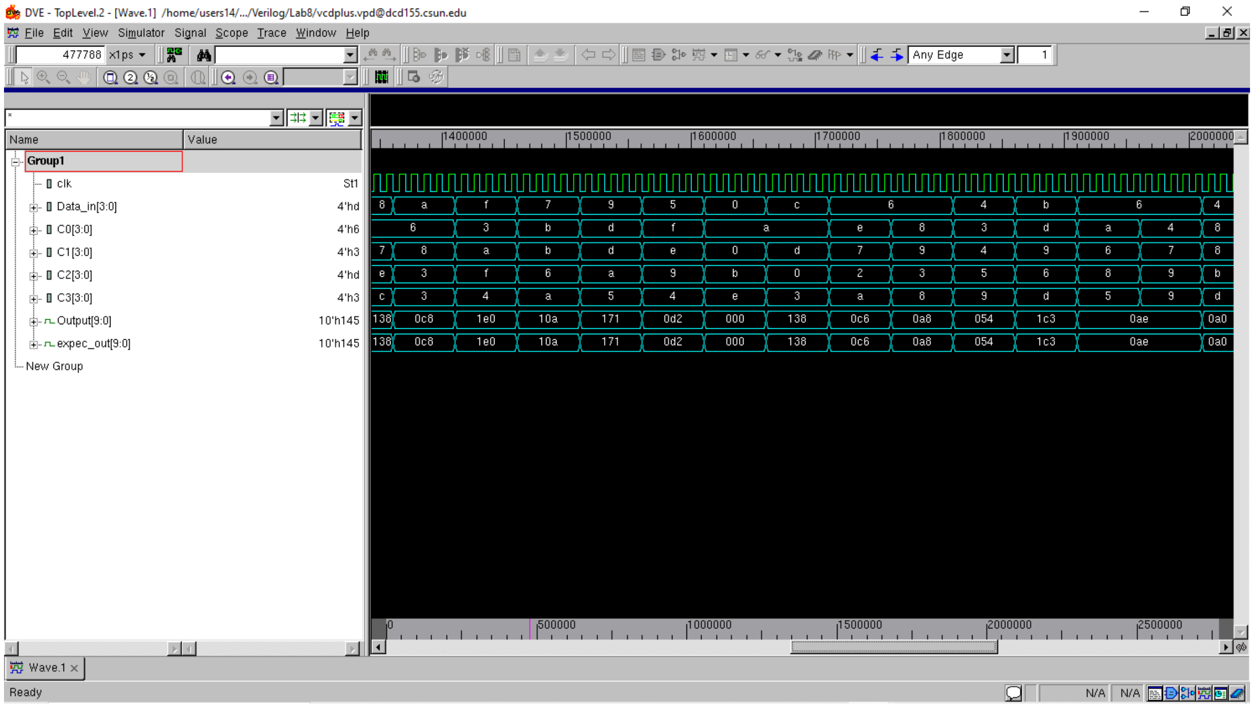


DVE - TopLevel.2 - [Wave.1] /home/users14/.../Verilog/Lab8/vcdplus.vpd@dcd155.csun.edu

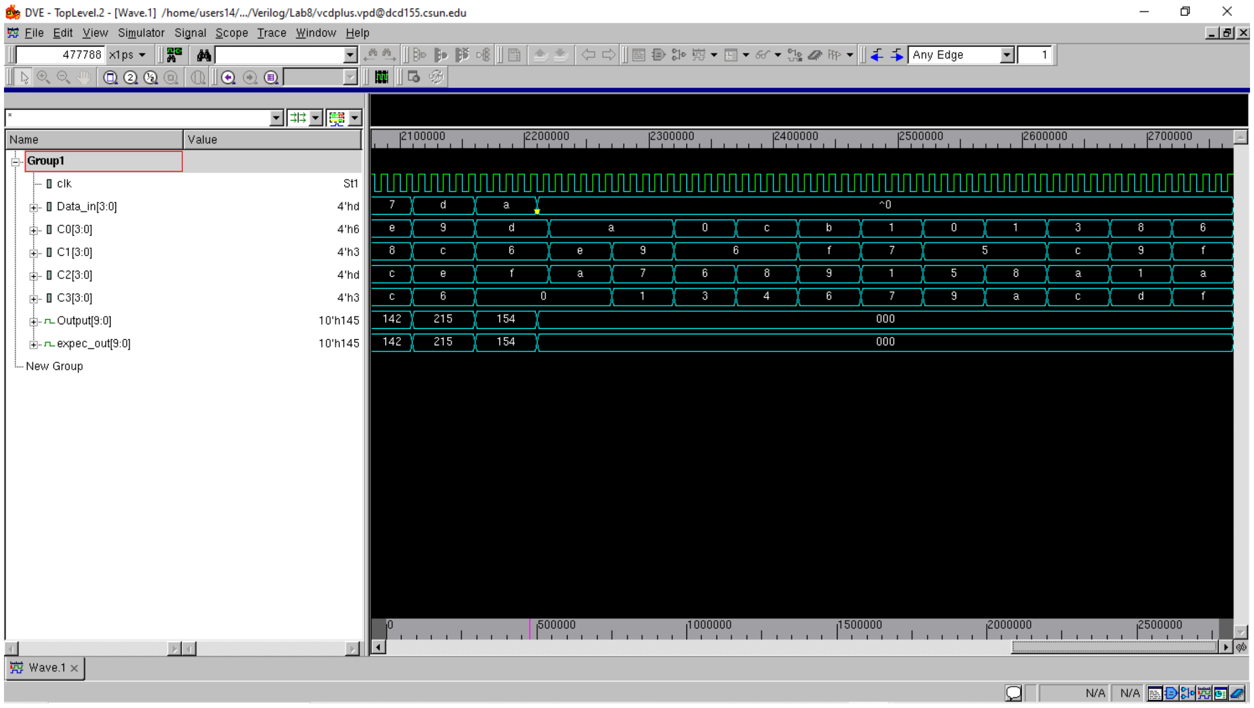
```
Name              Value
Group1
  clk             St1
  Data_in[3:0]    4'hd
  C0[3:0]         4'h6
  C1[3:0]         4'h3
  C2[3:0]         4'hd
  C3[3:0]         4'h3
  Output[9:0]     10'h145
  expec_out[9:0]  10'h145
New Group
```

```
Data_in:   1    b    c    5    e    2    8    b    7    2    f    b    8    a
C0:        3    5    6    8    9    b    c    e    f    8    e    3    9    6
C1:        8    6    a    f    5    c    7    9    1    2    4    5    7    8
C2:        8    6    b    1    f    9    6    7    d    5    f    e    3
C3:        9    e    1    a    9    f    c    0    c    d    9    8    a    c    3
Output:    01c  155  150  0dc  150  06a  170  160  0ee  03a  04c  12c  1ad  138  0c8
expec_out: 01c  155  150  0dc  150  06a  170  160  0ee  03a  04c  12c  1ad  138  0c8
```

Ready


DVE - TopLevel.2 - [Wave.1] /home/users14/.../Verilog/Lab8/vcdplus.vpd@dcd155.csun.edu
File Edit View Simulator Signal Scope Trace Window Help
477788 x1ps
Any Edge
1
Name
Value
Wave.1
Ready
N/A N/A

```
Name              Value     1400000   1500000   1600000   1700000   1800000   1900000   2000000
Group1
  clk             St1
  Data_in[3:0]    4'hd      8  a  f  7  9  5  0  c  6  4  b  6  4
  C0[3:0]         4'h6      6  3  b  d  f  a  e  8  3  d  a  4  8
  C1[3:0]         4'h3      7  8  a  b  d  e  0  d  7  9  4  9  6  7  8
  C2[3:0]         4'hd      e  3  f  6  a  9  b  0  2  3  5  6  8  9  b
  C3[3:0]         4'h3      c  3  4  a  5  4  e  3  a  8  9  d  5  9  d
  Output[9:0]     10'h145   138  0c8  1e0  10a  171  0d2  000  138  0c6  0a8  054  1c3  0ae  0a0
  expec_out[9:0]  10'h145   138  0c8  1e0  10a  171  0d2  000  138  0c6  0a8  054  1c3  0ae  0a0
New Group
                            0  500000  1000000  1500000  2000000  2500000
```


DVE - TopLevel.2 - [Wave.1] /home/users14/.../Verilog/Lab8/vcdplus.vpd@dcd155.csun.edu
File Edit View Simulator Signal Scope Trace Window Help
477788 x1ps
Any Edge
1
Name
Value
Wave.1
N/A N/A

```
Name              Value     2100000   2200000   2300000   2400000   2500000   2600000   2700000
Group1
  clk             St1
  Data_in[3:0]    4'hd      7  d  a  ^0
  C0[3:0]         4'h6      e  9  d  a  0  c  b  1  0  1  3  8  6
  C1[3:0]         4'h3      8  c  6  e  9  6  f  7  5  c  9  f
  C2[3:0]         4'hd      c  e  f  a  7  6  8  9  1  5  8  a  1  a
  C3[3:0]         4'h3      c  6  0  1  3  4  6  7  9  a  c  d  f
  Output[9:0]     10'h145   142  215  154  000
  expec_out[9:0]  10'h145   142  215  154  000
New Group
                            0  500000  1000000  1500000  2000000  2500000
```

Conclusion:

The Sum of Product model was successfully implemented using three level hierarchical scalable Register, Adder and Multiplies. And successfully tested using Non-Exhaustive and Exhaustive test strategies.

I hereby attest that this lab report is entirely my own work. I have not copied either code or text from anyone, nor have I allowed or will I allow anyone to copy my work.

Name (printed) Avinash Damse

Name(signed) [illegible] Date : 2-May-2023