

VLSI Final Project Report

Testing & Fault Tolerance in Digital Systems

John Adams && Mary Mouro | VLSI Testing | April 30, 2019

## compile instructions

1. Download and unzip our project named “vlsi\_Final\_Project”from canvas
2. Run command $ cd vlsi\_Final\_Project
3. Run command $ pip3 install sympy
4. Run command $ sudo apt-get install libz-dev
5. Run command $ cd vender/minisat
6. Run command $ make config prefix=$PREFIX
7. Run command $ sudo make install
8. Run command $ mv build/dynamic/bin/minisat $PATH TO vlsi\_Final\_Project/bin
9. move benchmark (.ckt files) to the project vlsi\_Fault\_TG/benchmarks
10. cd to project directory vlsi\_Final\_Project/src
11. run command "python3 main.py"

### NOTES:

Tested on linux Ubuntu 18.04 LTS

Minisat set solver has some dependencies that need to be installed, other project dependencies are listed below

### DEPENDENCIES:

sympy

libz-dev

python3

g++

## code structure

Class circuit

array Pis,Pos, gates

makeckt() → readnetlist()

D-algo()→propagate()→ justify() → makeDfrontier() → makeJfrontier()→ solve()

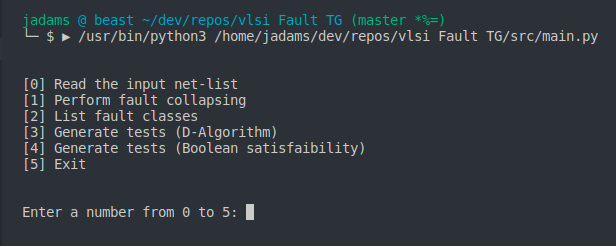
setsolver() → getExpr() → getclauses() → runMiniSAT() → getSAT()

getFaultlist()

collapseFaults() → func\_eq → dom()

## Results

The program starts by prompting the user with the required interactive menu



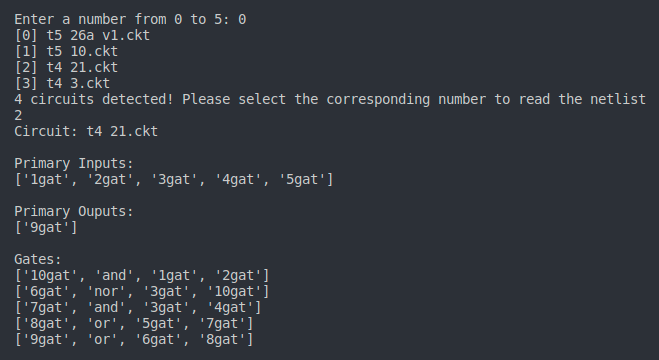
# Summary

To replace the placeholder text on this page, you can just select it all and then start typing. But don’t do that just yet!

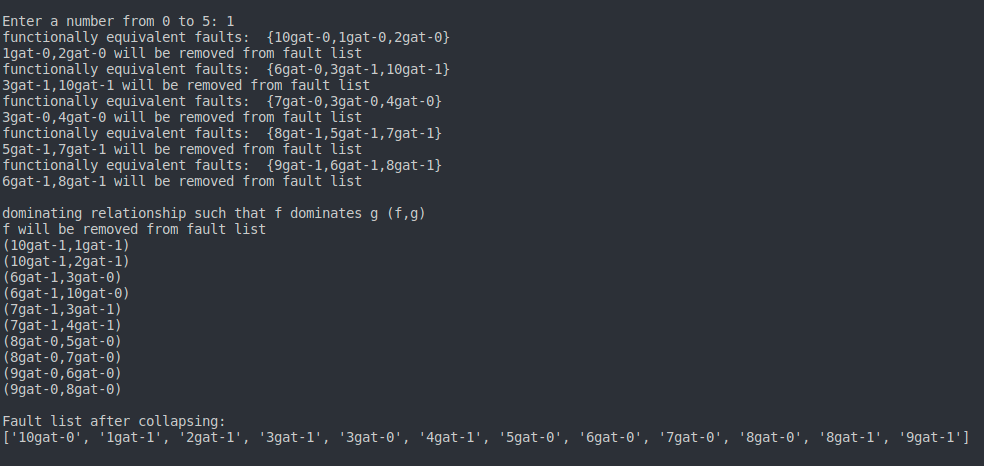
First check out a few tips to help you quickly format your report. You might be amazed at how easy it is.

**Option [0] Read Netlist**

we read in the netlist and show info about the circuit such as primary inputs and outputs and gate info.

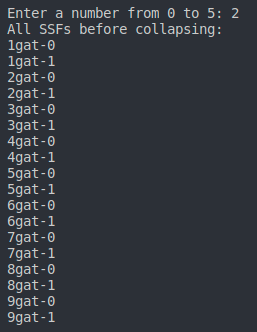
It prompts you again to select which circuit and after a selection is made it prints the circuit information

**Option[1] Fault Collapsing**

It will show the relationship for dominance and equivalences and show the collapsed fault list

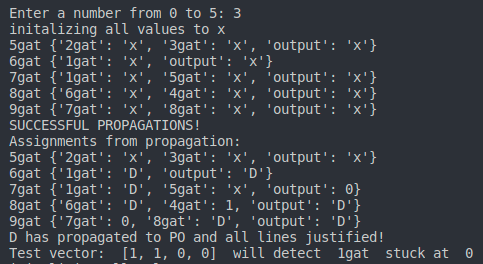
**Option[2] List fault classes**

The 2nd option will a fault list of all SSFs if option [1] is not selected first, otherwise it will show the fault list similar to above. Here’s an example of listing fault classes with fault collapsing.



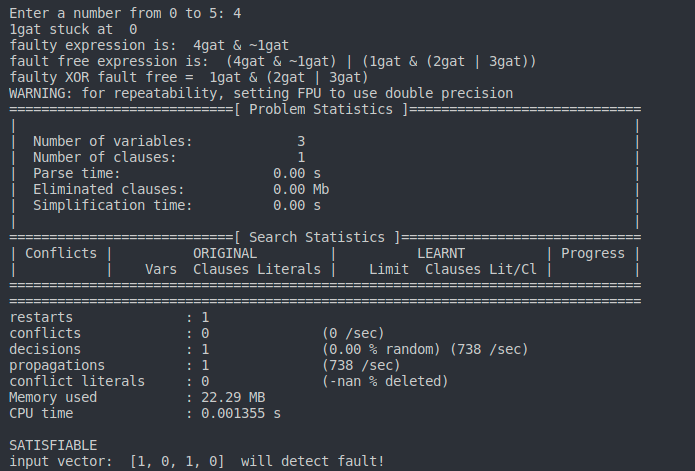
**Option[3] D-Algorithm**

this option will show the assignments made to propagate the error and the test vector that detects a fault if it is detectable.



**Option[4] Boolean Satisfiability**

This option uses a modern SAT solver to generate the test vectors to detect single stuck at faults.



## SUMMARY

The project uses the circuit class functions shown in the UML diagram in the code structure section and performs the reading of a netlist file, fault collapsing using fault equivalence and fault dominance relationships, and finally test generation using the D-Algorithm and the miniSAT solver. The D-Algorithm first attempts to propagate the error to the output and then goes back and justifies all the lines and returns the test vector that can detect the faults in the fault list. The setSolver function calculates the circuit’s Boolean expression for the fault free circuit and the expression for the faulty circuit, XORS them, converts the XOR expression to CNF and then calls the miniSAT set solver.