

VLSI Testing Final Project

Fault Diagnosis

Introduction:

As the best test expert in our *NTU-ATPG* company, your team are asked to implement a diagnosis tool for **single stuck-at faults** and **multiple stuck-at faults**. To test our tool, we will have to first generate *fail log* files. Then you will need to **diagnose these fail log** files and **give a ranked list of suspects**. The failing output will then be diagnosed by your tool. This project will be graded based on **diagnosis accuracy**, and **diagnosis resolution**, and **run time**.

Example:

For the C17 circuit and test pattern applied. Suppose we test the C17 circuit and we get a Fail Log file for a certain CUD (circuit under diagnosis).

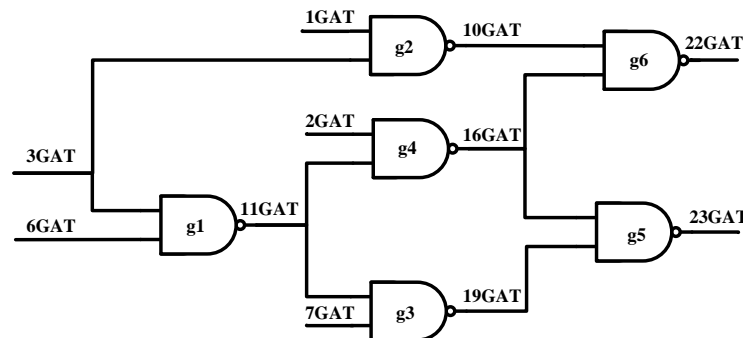


Figure 1. *c17* circuit

```
#Circuit Summary:
```

```
#-----
```

```
#number of inputs = 5
```

```
#number of outputs = 2
```

```
#number of gates = 6
```

```
#number of wires = 11
```

```
T'00110'
```

```
T'01000'
```

```
T'00001'
```

Figure 2. *c17.pat*

To emulate the faulty CUT output, you should generate a Fail Log file,

```
./atpg -genFailLog <pattern file> <circuit file> -fault <wire> <gate> <io> <fault type>
```

The generated Fail Log is a text file in the following format:

```
<pattern index> <faulty output wire> <expect value> <faulty value> # <pattern>
```

...

For example, if we run the following command on *c17*,

```
./atpg -genFailLog ../patterns/c17.pat ../sample_circuits/c17.ckt -fault 16GAT g4 GO SA0
>c17-001.failLog
```

After execution, your fail log should be like this:

vector [0]	22GAT	expect L, observe H	#	T'00110'
vector [0]	23GAT	expect L, observe H	#	T'00110'
vector [2]	22GAT	expect L, observe H	#	T'00001'

Figure 3. *Example Fail Log File*

To generate fail logs, please add double quotes before and after brackets. Or add backslash before brackets. For example, please type:

```
$ ./atpg -genFailLog ../patterns/golden_c499.ptn ../sample_circuits/c499.ckt -fault ID7("("7")" g389 GI SA1
```

If you want to generate fail logs of **multiple** stuck at faults, please type -fault before each fault. For example, please type:

```
$ ./atpg -genFailLog ../patterns/golden_c499.ptn ../sample_circuits/c499.ckt -fault ID7("("7")" g389 GI SA1 -fault
```

```
ID16("("16")"* g52 GI SA1
```

(Multiple stuck-at faults means that more than one fault exists in the circuit **at the same time.**)

Next, you can run the diagnosis tool by the following command,

```
./atpg -diag ../patterns/c17.pat ../sample_circuits/c17.ckt ../FailLog/c17-001.failLog
```

The following is an example diagnosis report. The suspect faults are ranked by their score. (you can define your own score.) Please perform equivalent fault collapsing. Please put all equivalent faults at the end of each line. You can find a sample solution `info_failLog` excel file in the `failLog` directory.

```
#Circuit Summary:
#-----
#number of inputs = 5
#number of outputs = 2
#number of gates = 6
#number of wires = 11
#number of vectors = 3
#number of failing outputs = 3

Ranked suspect faults
No.1  16GAT g4 GO SA0,  TFSF=3, TPSF=0, TFSP=0, score=100.0[equivalent faults: x SA0, y SA1 ...]
No.2  22GAT g6 GO SA0,  TFSF=2, TPSF=0, TFSP=1, score=66.7 [equivalent faults:  ...]
...

# run time = 0.20 s
```

Figure 4. *Example Diagnosis Report*

We have a reference binary executable in the `bin_reference` directory. You can try it and see if you have any question.

Assignments:

1. You are required to implement a **-genFailLog** function.
2. You are required to implement a **stuck-at fault diagnosis tool**. You will be given fail log files: some of them are **single** stuck-at faults; the other of them are **multiple stuck-at** faults. Please note that **smaller number** of diagnosed suspect faults is better than larger number of diagnosed suspect faults. For example, if you can find one correctly diagnosed

suspect fault is better than two suspect faults, even when their failLogs are the same.

Please note that you may not find a perfect suspect fault for some fail logs. That means, the diagnosis score of the top ranked suspect fault may be lower than 100.

Diagnosis accuracy is defined as :
$$\frac{\text{number of correctly diagnosed faults in your report}}{\text{number of total faults injected by TA}} \quad (1)$$

Correctly diagnosed faults mean those faults ranked within **top 5** in your diagnosis report.

Diagnosis resolution is defined as :
$$\begin{cases} 30, & \text{if number of diagnosed faults is zero} \\ \frac{\text{number of diagnosed faults in your report}}{\text{number of correctly diagnosed faults}}, & \text{otherwise} \end{cases} \quad (2)$$

Please note that, **all equivalent faults are counted as one fault** for both equations.

If some of your diagnosed faults, which are different from that injected by TA, can result in the same fail log, you can use those diagnosed faults as **your injected faults**.

In this case, your diagnosis accuracy is 1 if the number of your injected faults is less than the number of TA's injected faults, and is 0 otherwise. As for the diagnosis resolution, the denominator of the lower part of (2) would become the number of your injected faults ranked within **top 5** in your diagnosis report.

Please note that you should mark your injected faults in this case so that TA can validate your results.

3. In your report, please fill in the following table.

Circuit number	Average diagnosis accuracy*	Average diagnosis resolution	Average Run time
C432			
C499			
C880			
C1355			
C2670			
C3540			
C6288			
C7552			
Average			

* the average results of all faillogs for each circuit.

Grading:

85% Diagnosis Results (see the following)

15% Presentation and report (see the other file)

Scores of diagnosis results:

Average diagnosis results are ranked by three factors: **diagnosis accuracy, diagnosis resolution, and run time**. The team(s) with the best diagnosis accuracy will get the highest points (say, 100 points). The second rank will be deducted by four points and

etc. Teams with very close accuracy will be ranked the same. Diagnosis resolution is ranked in a similar way but the full score is half of diagnosis accuracy (e.g., 50 points) with 2 points deducted for each rank below. Runtime is also ranked in a similar way but the full score is half of diagnosis accuracy (e.g., 50 points) with 1 point deducted for each rank below. **The runtime limit for each fail log is 3,600 seconds.** The total score is the summation of three scores.

Deadlines: See the other file

References

- [Ye 2010]Jing Ye, Yu Hu, Xiaowei Li “Diagnosis of multiple arbitrary faults with mask and reinforcement effect,” DATE 2010.
- [Desineni 2006] Desineni, R., Poku, O. ; Blanton, R.D. “A Logic Diagnosis Methodology for Improved Localization and Extraction of Accurate Defect Behavior,” ITC 2006.
- [Yu ITC’08] X. Yu, R.D. Blanton “An Effective and Flexible Multiple Defect Diagnosis Methodology Using Error Propagation Analysis,” ITC 2008.
- [Yu DAC’08] X. Yu, R.D. Blanton “Multiple defect diagnosis using no assumptions on failing pattern characteristics, “ DAC 2008.