

EEIE30069: VLSI Testing

Assignment #2 (Due: Sep. 30, 2024 23:59:99)

Last update: Sep. 15, 2024

Reading:

- Source code provided in assignment #0
- Please put all the results into a brief report and try to discuss them
- Please compress the folder containing all the source code and the short report into a file named by ASS2_<your-student-ID> and upload it to new E3
- The report should include:
 - The algorithm or idea of your code
 - Several case results
 - Discussion about your results
 - How to compile your code
 - Other information (optional)

Homework Description:

1. (10 pts) Run the program as a logic simulator on circuits c17.bench and c7552.bench.
 - To verify the correctness, please use [c17.input](#) as a set of input vectors for c17.bench and their corresponding correct output responses from [c17.output](#). Also the [c7552.input](#) (inputs) and [c7552.output](#) (outputs) are available for c7552.bench.
2. (90 pts) **Single Pattern Logic Simulator**
 - a. (40 pts) Generate 100, 1000, 10000 or more random patterns, and apply the generated patterns to logic simulation of combinational benchmarks. Please compare CPU times and memory usage of different circuits in your report.
 - To measure memory usage, please check [the suggestions](#).
 - You will need to write a small routine for the random number generator. Please be careful that you can't simply take the last bit out of any random functions such as rand(), because it's not a random bit at all. It is recommended (but not required) using [GSL](#). Please read [the explanation page for generating pseudo-random numbers](#).
 - b. (50 pts) In our current implementation of the logic simulator, we use different tables for AND, OR, and NOT operations. Please revise the program such that we use CPU instructions to compute the results directly. Note that we have to consider unknown values (X). To handle unknowns, we would need at least two bits to represent a 3-

value variable (for 0, 1, and X). Also, please generate random patterns with Xs, so that you can test your new program.

Grading:

- **Correctness 90%**
- **Report 10%**

Attention:

1. The commands for 2.-a. are as follows:

- `./atpg -pattern -num {integer} -output [output_name] [circuit_name]` (To generate patterns w/o unknown)
- `./atpg -pattern -num {integer} -unknown -output [output_name] [circuit_name]` (To generate patterns w/ unknown)

Example:

```
./atpg -pattern -num 100 -output c17.input c17.bench
```

2. The command for 2.-b. is as follows:

- `./atpg -mod_logicsim -input [input_name] [circuit_name]`

Example:

```
./atpg -mod_logicsim -input c17.input c17.bench
```

3. Memory usage example:

- In command line:
`/usr/bin/time -f "Average memory usage: %K \nMaximum memory usage: %M\n" ./atpg c17.bench`
- In main.cc, add:
`#include<sys/types.h>`
`#include<unistd.h>`
`int pid=(int) getpid();`
`char buf[1024];`
`sprintf(buf, "cat /proc/%d/statm",pid);`
`system(buf);`
- In main.cc, add:
`system("ps aux | grep atpg");`