ESE549 Advanced VLSI System Testing Mar 3rd, 2018

Report of Project2

Instructor: Peter Milder *Xuechun Xie 111684717,Chaoming Tseng 111446676*

1. The solutions is based on project 1,the algorithm is as below:

Algorithm 1 Solution for project 2

```
Initially get all the POs
while there is a PO that never been visited do
  choose that PO,and get its predecessors' size
  if gate's predecessor is 0 then
    return the input of this gate
  else
    if this predecessor is not set then
      Recursively simulation(Gate* gate) to go through gate's input value
    else
      return gate's input value
    end if
    check if the input contains 0,1,X,D,B and return the expected output of the gate
    if this gate has no fault then
      Set the gate's value=expected output
    else if this gate has SA0 fault then
      if Gate's expected value=0 or X then
        set gate's Value=expected output
      else
        set gate's value=D
      end if
    else if this gate has SA1 fault then
      if Gate's expected value=1 or X then
        set gate's Value=expected output
      else
        set gate's value=B
      end if
    end if
  end if
end while
```

2. Below are the steps that we verified our solution:

a. Compared my expected value in the source code with truth table

b. Run all the tests provided in tests folder,and C5315≈ 15.1s,C6288≈8.99s,c7552≈31.47s

3. The major problems that we met and the corresponding solutions are as below:

a. Problem: The PO's depth was always 1 greater than the expected value when I was trying to improve project1 by returning the gate's depth if it was set before

Solution: change the code in the iteration from gate->getDepth() to gatePredessors[i]->getDepth(), because if it is gate->getValue(), then in the last iteration the depth would be the PO's depth instead of the predecessors'.

b. Problem: when there is a SA0 and the expected output for the good circuit is 0,the actual output was always D. Similar situation for SA1.

Solution: Fistly we checked the faults and found the rule that all Expected value=0 (1 for SA1) has been set to D, then we noticed we should set the actual output to 0 (1 for SA1) instead of simply D (B for SA1).

c. Problem: The Actual output was set to one of 0/1/D/B while it the expected value is X.

Solution:

- Firstly, we ran all the benchmarks, and from c432 to c7552, some of the inputs cause the difference between mycxx.out with cxx.refout
- Secondly,we checked different gate types logic one by one, and from c17,tiny,fulladder,xor,we asserted that the gates' logic are correct
- In the 3rd step, we tried the way in item b to see if there was any rules, and found that the Actual output was set to one of 0/1/D/B while it the expected value is X
- We thought the defect may be caused by multi-input AND/NAND/OR/NOR gates, so we tried to mimic the output by drawing out the layout of the circuit, because there is no benchmark for us to test muti-input gates.
- We tried from the shortest path, and found it in a inverter with SA0 that when the input is X, my primary output was D while expected output is X.
- We started to think that maybe my understanding about D and DBAR was wrong,so after we discussed we found that when a inverter with SA0 and input=X,then the output should be X instead of DBAR.Problem solved.

d. One tiny question, is there any possible that we can have some benchmarks to test multi-input easily?