ESE549 Advanced VLSI System Testing Feb 13th, 2018

Report of Project1

Instructor: Peter Milder *Xuechun Xie 111684717*

Part 1. Answers for questions are as below:

1. Circuit* mycircuit;
2. Gate* mygate;
3. mygate->get_gateType();
4. mygate->get_gateOutputs();
5. LOGIC_UNSET
6. vector<Gate *>;
7. vector<Gate *>myPIGates;
   myPIGates=myCircuit->getPIGates();
8. vector<Gate *>myPIGates;
   myPIGates=myCircuit->getPOGates();
9. a controlling value is no matter what other input is, if there is a controlling value then the PO has the same value with controlling value. OR gate's controlling value is 1, XOR doesn't have controlling value

Part 2&3 .

1. The solutions for part2 and part3 are as the algorithms below

Algorithm 1 Solution for Part2

```
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  this gate's depth=0
  else
    Compare the depth between predecessors, and return the max one
    Recursively run the above step
  end if
end while
Return the depth of PO
```

Algorithm 2 Solution for Part3

```
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
    go through all the predecessors of the gate
    check if the input contains 0,1,x
    according to the input and gate type return the output of this gate
    Recursively run the above 3 steps
  end if
end while
Return the output of PO, and set the value
```

2. Below are the steps that I verify my solution
   - **a.** Draw out the layout of the circuit of C17 according to the description in bench file
   - **b.** Print every gate's depth in console and compare to the one which I calculated from the layout
   - **c.** Run the script and check myc17.out and compare to c17.refout in person
   - **d.** Run all the tests provide in the .\tests folder
3. The major problem that I had were
   - **a.** Don't know how to iterate or recursive a multi-way tree
   - **b.** Had difficulty in figuring out how to simulate the output with input value 0,1,X