EECS 478 Fall 2014 Project 3 Report

Name: Pengxiang Jiang Uniquename: jpxx

Task 1 – Topological Sorting

This function will generate a topological ordering of the gates and print them in topological order. I searched online and found that there were usually two algorithms for implementing topological sort. The first one is source removal algorithm. The second one is depth-first search (DFS) algorithm for topological sorting. I chose to use the second method. My final topological sort implementation was based on the pseudo code which was published in Wikipedia:

http://en.wikipedia.org/wiki/Topological_sorting

For coding this function, I added *vector<Node*> L* inside the Circuit class to store the results of topological sorting. And I created an integer variable *mark* inside the Node class to help to decide if a node has been marked.

The tricky part here is to understand the different meaning and functions of STL containers like vector, set, queue, and to use them correctly. I generated Node vector here because I think it is a good choice and it is convenient to store all of the sorted nodes.

Task 2 – Functional Simulation

This function will simulate and generate the output values of a circuit.

For implementing this function, I created a *truthType value* variable in the Node class to store the simulated value for each node. I generated a function inside *truthTable.cpp* called *assignment()* to assign the output value for internal nodes. Although in this project, we didn't require to handle with the "don't care" value, I considered and solved this case in the *assignment()* function by adding an if statement: *if (logic[i][j] == DC) continue.* The reason I tried to solve the DC case is that it will be convenient for me to verify the correctness of my simulation function. We could get many circuit modules with *blif* format by running Project 2. Then by generating corresponding input file, we could easily implement verification for the simulation function. But the problem is, most of the *blif* file generated by Project 2 contained "don't care" value. So I believe that it is necessary to solve this problem in this project. And it is not a complex task.

Code Verification

After finishing the preceding two tasks, I ran the whole code to verify its function. I used *adder16.blif* file given in Project 2. And I generated its input file to give all the input nodes and values. First, I verified the topological sorting function. I checked the order of each PI nodes and made sure that they were correct. Because I used the algorithm based on DFS which loops through nodes in arbitrary order, the final output order could cost more time to check its correctness. For functional simulation, I changed the input values many times to check the performance of the code. Finally, I made sure that all of the output results were correct.