EE 658

ASSIGNMENT 6

JAIN, AKASH*: 3349 2057 74*

ISMAIL, SHEHZAD: 9022 7106 48

A –

Here, since we need to consider the faulty parts of the circuit, we take 6 valued cover as our reference for the values at each gate input and output.

Initially, all the gates have been implied to assume all values. i.e. the D-Frontier is considered to have X as possible values. As we start getting the values, either at the input or the output, we start modifying the D-frontier accordingly. All the values which are not possible are removed from the D-Frontier of that line (either input or output point). We chose a random value and continue forward until we get to the primary output. To cover all the values at other nodes, we do backward implication to find the complete result.

B –

Here our task is to insert a fault at a particular site. When we assign the value, we make sure that it cannot be changed under any circumstance. We then check the location of the fault and update the to-do list. We then perform backward and forward implication and complete the value assignment in the circuit.

C –

Here we have a fault at a particular point. We insert the fault at the appropriate location. We initialize all the Primary inputs with K(kai-6). Then we initialize all the transitive fanouts of the fault site with X6. After this has been done, we check if any node has not been initialized yet. If there is any remaining, they will be initialized by K (kai-6). This last step is done because these nodes will never face the effect of the fault. So we can never assume the value of D or DBar.

D –

This module can be divided into some steps. Initially, we need to find the output at the faulty site gate. We need to make sure that initialization has been done. This includes Primary inputs with K6 and transitive fanouts having D-Frontier values as X6. We then choose an appropriate input vector and check if it can be used to find the fault. If it is not, we need to employ backtracking.

E –

Did not generate any algorithm for this part

F –

This algorithm provides us flexibility of going back to a state if we reach a conflict. This is simply done by saving the state at only those places where we have more than one input combination to choose from. We do not need to save state if we do not get any point where we have only one implied values to take. This prevents us from having many stages. Also it will be an inconvenience when we backtrack as we will have no other option to choose from.