Fault Equivalence

# Pseudo code

* Make a list of Primary inputs
  + (PIs are the ones whose ‘type’ is ‘input’)
* Make a list of Primary Outputs
  + (POs are the ones whose ‘fanout’ is zero)
* Make a dictionary of faults at each node and assign s\_a\_0, s\_a\_1 to all nodes
* Call the fault\_equi(node) function for each output node in a for loop

# Fault\_equi(node)

* Return if the node is a PI.
* Check the available faults at the node. If both faults occur, then one of the faults of fanins of that node will surely cancel
* If the node is a branch, then assign the same fault as that of node to the branch and call the fault\_equi() function for the branch.
* Else Check the gate of that node
* Based on the gate and the fault available at the node, set the s\_a\_0/s\_a\_1 fault of both the fanins to False.
* Call the function for both the fanins for all gates except not.
* Call the function for fin1 for not gate.
* Call the function for branch, if the node is a stem.