**ANOSH DARUWALLA FINAL PROJECT REPORT**

**ECE 6140 12/10/13**

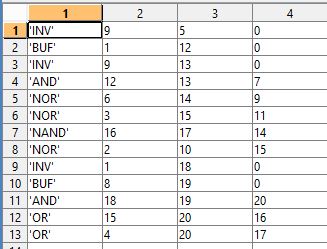
**GTid: 902765842**

**Final Project Report:** Digital Systems Testing

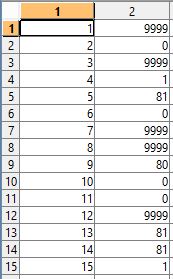
**PODEM:**

**Structures used:**

The cell arrays were used to store the circuit nodes. These arrays divide the nodes and the gate names into individual cells of the array, so that be accessed easily. The array is as shown in the figure below:

Cell for s27.txt

Cells are also used to keep track of the D-frontier generation. Whenever a gate is detected to be in the D-frontier, it gets added to that cell and the cell is checked for propagation of D-frontier and whenever it is empty, we detect a failure if the current node is NOT one of the output nodes. A cell is also used to keep track of every node and its value so that it can be accessed just like a matrix.



This is the node list for all the nodes of the circuit s27.txt. After the code is run, the nodes show their corresponding values. (don’t care – 9999, D – 80, D’ – 81).

**Process flow:**

The process flow or pseudo algorithm is shown below. The same algorithm given the text book was used. podem() is a recursive function, which returns a value of ‘true’ when it succeeds and ‘false’ when it fails. Three other functions used are for the objective(), backtrace() and imply().

enter input file and fault

define 5 value truth tables

initialize all cells and variables

**podem()**

begin

if the node is at a primary output, return SUCCESS

if (test is not possible), return FAILURE

Q1{}=Q{}; *%store older values of nodes in case of reversal*

[k,v]=**objective()** *%sets the objective of the fault*

[j,v]=**backtrace(k,v)** *%finds an x-path and backtraces to a primary input*

**imply(j,v)** *%implies the values at nodes to see fault activation/Dfrontier propagation*

if podem()==SUCCESS, return SUCCESS

Q{}=Q1{}; *%restore previous values*

**imply(j,v’)** *% decision reversal*

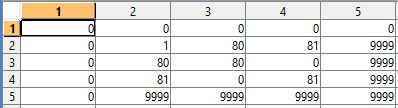
if podem()==SUCCESS, return SUCCESS

**imply(j,x)**

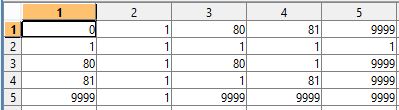
return FAILURE

end

5-value truth tables for AND and OR are shown below:



AND

OR

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Circuit | Fault | Result | Corresponding test pattern |
| s27 | 16-a-0 | SUCCESS | 0 0 X X 0 1 X |
|  | 10-a-1 | SUCCESS | X 0 0 X X X 0 (fault at input) |
|  | 12-a-0 | SUCCESS | 1 X X X 1 X X |
|  | 18-a-1 | SUCCESS | 1 X X 1 0 1 1 |
|  |  |  |  |
| s298 | 70-a-1 | SUCCESS | 1 0 1 1 X X X X X X X X X X 0 X X |
|  | 73-a-0 | SUCCESS | 1 1 1 X X X X X X X X X X X 0 X X |
|  | 26-a-1 | SUCCESS | X X X X X X X X X X X X X X X X X (fault at output) |
|  | 92-a-0 | SUCCESS | X 1 0 1 0 1 X X X X X X 0 X 0 X X |
|  |  |  |  |
| s344 | 166-a-0 | SUCCESS | 0 1 X 0 0 X X X X X 0 1 1 X X 0 X X X X X X X X |
|  | 71-a-1 | SUCCESS | 1 0 X X X X X X X X X X X X X X X X X X X X X X |
|  | 16-a-0 | SUCCESS | 1 X 0 X X X X X X X X X X X X 1 X X X X X X X X (f.a input) |
|  | 91-a-1 | SUCCESS | 1 1 1 X X X X X X X X X X X X X X X X X X X X X |
|  |  |  |  |
| s349 | 25-a-1 | SUCCESS | X X X X X X X X X X X X X X X X X X X X X X X X (f.a.o) |
|  | 51-a-0 | SUCCESS | 1 1 1 X X X X X X X X X X X X 0 X X X X X X X X |
|  | 105-a-1 | SUCCESS | 1 X 0 1 1 X X X X X 0 1 0 0 X 0 X X X X X X X X |
|  | 7-a-0 | SUCCESS | X X X X X X 1 X X X X X X X X X X X X X X X X X (f.a.i) |