Lecture 9: Boolean Testing Using Fault Models (D-Algorithm, PODEM)

Instructor: M. Tahoori
Example: A/0

Step 1
- D-Drive: Set $A = 1$

Step 2
- D-Drive: Set $f = 0$
Example: A/0

- Step 3
  - D-Drive: Set $k = 1$

Example: A/0

- Step 4
  - Consistency: Set $g = 1$
Example: A/0

Step 5
- Consistency: f = 0
  - Already set

Step 6
- Consistency: Set c = 0, Set e = 0
Example: A/0

- **Step 7**
  - Consistency: Set $B = 0$
  - Test found: $ABCD = 100X$

Example s/1

- Primitive D-cube of Failure
Example s/1

Propagation D-cube for v

Forward & Backward Implications
Example s/1

- Propagation D-cube for Z
- Test found!

Example: u/1

- Primitive D-cube of Failure
Example: \( u/1 \)

- Propagation D-cube for \( v \)

- Forward and backward implications
Example: u/1

- Inconsistency
  - $d = 0$ and $m = 1$
    - cannot justify $r = 1$ (equivalence)
- Backtrack
  - Remove $B = 0$ assignment
Example: $u/1$

- Propagation D-cube for $v$

Example: $u/1$

- Propagation D-cube for $Z$
Example: u/1

- Propagation D-cube for Z
- Implications
- Test Found!

PODEM
Motivation

- IBM introduced semiconductor DRAM memory into its mainframes – late 1970’s
- Memory had error correction and translation circuits
  - To improved reliability
- D-ALG failed to generate test for these circuits
  - Search too undirected
  - Large XOR-gate trees
  - Must set all external inputs to define output
- Needed a better ATPG tool

PODEM -- Goel IBM (1981)

- Path Oriented DEcision Making
- New concepts introduced:
  - Expand binary decision tree only around primary inputs
    - This reduced size of tree from $2^n$ to $2^{num_{PI}}$
  - Use X-PATH-CHECK
    - To test whether D-frontier still there
    - D-Algorithm tends to continue intersecting D-Cubes
      - Even when D-Frontier disappeared
  - Objectives
    - bring ATPG closer to propagating D (D) to PO
  - Backtracing
    - To obtain a PI assignment given an initial objective
Assigning Input Values (PODEM)

1. Assign value to an unassigned primary input
2. Determine all implications of assignment
3. If test is generated, exit; else
4. Is test is possible with additional input assignments?
   - fault site doesn’t have fault value assigned
   - Path of unassigned leads from D (D̅) to an output
   - If yes, go to 1, if no
5. Change input assignments to untried combination, go to 2
   - If no untried combination exists — untestable fault

Example: Test For k/1

- Put D’ on k
  - D-Alg: assigned a D’ to k and propagate it to output f
  - PODEM: try to justify 0 on k
Example: Test For k/1

- Justify 0 on d
- Implication

Example: Test For k/1

- K still hasn’t D’
- Justify 0 on c
- Implication: k=h=m=D’
Example: Test For k/1

- Propagate through w
- Set g = 1
- Implication

Example: Test For k/1

- Conflict
  - f is 1 so propagation is blocked
- Reverse the last assignment made to a PI
  - Set b = 0
- Implication
Example: Test For k/1

- There is a propagation path from m to f
  - Set p = 1
- Implication
- Test found
  - abcd = 0001

Another Example
Example: S/1

- Select path s - Y for fault propagation

Example: S/1

- Initial objective:
  - Set r to 1 to sensitize fault
Example: S/1

- Backtrace from r

Example: S/1

- Set A = 0 in implication stack
Example: S/1

- Forward implications: \( d = 0, X = 1 \)

Example: S/1

- Initial objective: set \( r \) to 1
Example: S/1

Backtrace from r again

Example: S/1

Set B to 1.

Implications in stack: \( A = 0, B = 1 \)
Example: S/1

- Forward implications:
  - $k = 1$, $m = 0$, $r = 1$, $q = 1$, $Y = 1$, $s = \overline{D}$, $u = \overline{D}$, $v = \overline{D}$, $Z = 1$

```
  0 0 1 sa1 1 D 1
  0 1 1 1 1 1
```

Example: S/1

- X-PATH-CHECK
  - paths $s - Y$ and $s - u - v - Z$ blocked

```
  0 0 1 sa1 1 D 1
  0 1 1 1 1 1
```

Copyright 2010, M. Tahoori
Example: S/1

- Set B = 0 (alternate assignment)

Forward implications:
- $d = 0$, $X = 1$, $m = 1$, $r = 0$, $s = 1$, $q = 0$, $Y = 1$, $v = 0$, $Z = 1$
- Fault not sensitized
Example: S/1

Set A = 1 (alternate assignment)

Example: S/1

Backtrace from r again
Example: S/1

- Set B = 0.
- Implications in stack: A = 1, B = 0

Example: S/1

- Forward implications:
  - d = 0, X = 1, m = 1, r = 0.
- Conflict: fault not sensitized. Backtrack
Example: S/1

- Set $B = 1$ (alternative assignment)

Forward implications:
- $d = 1$, $m = 1$, $r = 1$, $q = 0$, $s = \overline{D}$, $v = \overline{D}$, $X = 0$, $Y = \overline{D}$
- Test found
PODEM

Major aspects
- Which primary input should be assigned a logic value?
- What value to assign to the selected primary input?
- Determining inconsistencies in primary input assignments
- Handling inconsistencies

Which PI to Choose?

- Decision gate
  - Logic value at the output of a gate is such that only one input of the gate can control its output to the desired value
    - AND with output 0
- Imly gate
  - Logic value at the output of a gate is such that all inputs of the gate must be at a particular value in order to control its output to the desired value
    - AND with output 1
- To justify a logic value at the output of a decision gate, choose the “easiest” input.
  - The shortest logical path to primary inputs or has the best controllability
- To justify a logic value at the output of an imply gate, choose the “hardest” input
  - The longest logical path to primary inputs or has the worst controllability
What Value to Assign?

- Path from the objective site to the selected primary input has an **even** number of inversions
  - Assign the **same** value to PI as the objective
- Path from the objective site to the selected primary input has an **odd** number of inversions
  - Assign the **opposite** value of the objective to PI

Inconsistencies in PI Assignment

- After every primary input assignment, an implication step is performed.
- During implication, inconsistencies in primary input assignments are detected using the following rules:
  - If there are conflicting assignments at the same signal line of the network
  - If the logic value at the fault site doesn’t activate the fault
  - If there is no path from the fault site to a primary output such that all side inputs of that path are either X or set at non-controlling values
Handling Inconsistencies

- Backtracking
  - Flip the logic value at the primary input
    - Which was the last one to be assigned a value
  - Stack of primary inputs that have been assigned values
  - After flipping implication step is performed
    - No inconsistency detected
      - Continue
    - Otherwise
      - That primary input is removed from the stack and
        - X is assigned to that primary input
      - POP the next assigned PI from stack and repeat