CSE435 Introduction to EDA & Testing - Spring 2022

Homework Assignment #6

Shao-Hsuan Chu - B073040018

About Path-Oriented Decision Making (PODEM), please answer the following questions according to Figure 1.
1-4 PODEM (See Figure 1)

1. (15%) Derive the effective test set for G5 SA0 by PODEM

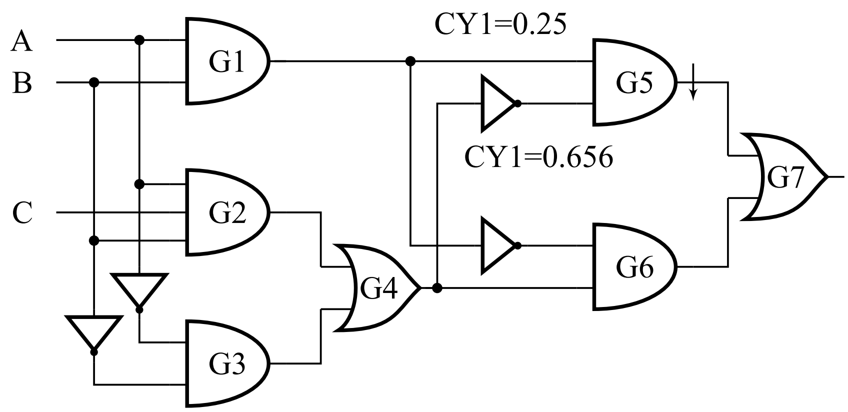

Figure 1

Solution:

1. Activate G5 SA0. Initial objective: (G5, D).
2. Backtrace to A. With an even inversion parity, A = 1. Simulation, objective not achieved.
3. Backtrace to B. With an even inversion parity, B = 1. Simulation, objective not achieved.
4. Backtrace to C. With an odd inversion parity, C = 0. Simulation, objective achieved. The fault has reached the primary output.
5. Test generated, (A, B, C) = (1, 1, 0).

Answer: (A, B, C) = (1, 1, 0).

2. (15%) Derive the effective test set for G5 SA0 by PODEM with More Intelligent Backtracing

Solution:

1. Activate G5 SA0. Initial objective: (G5, D).
2. (a) Since G5 is an imply gate, backtrace to the hardest unknown input, G1, with CY1(G1) = 0.25. With an even inversion parity, intermediate objective: (G1, 1).
   (b) Since G1 is an imply gate, keep backtrace to the hardest unknown input, A, (CY1(A) = CY1(B) = 0.5, so we may choose either one). With an even inversion parity, intermediate objective: (A, 1).
   (c) Since A is not a gate, stop backtracing. Assign 1 to A. Simulation, objective not achieved. Implication: G3 = 0.
3. (a) Since G5 is an imply gate, backtrace to the hardest unknown input, G1, with CY1(G1) = 0.25. With an even inversion parity, intermediate objective: (G1, 1).
   (b) Since G1 is an imply gate, keep backtrace to the hardest unknown input, B. With an even inversion parity, intermediate objective: (B, 1).
   (c) Since B is not a gate, stop backtracing. Assign 1 to B. Simulation, objective not achieved. Implication: G1 = 1, G6 = 0.
4. (a) Since G5 is an imply gate, backtrace to the hardest unknown input, G4. With an odd inversion parity, intermediate objective: (G4, 0).
   (b) Since G4 is an imply gate, keep backtrace to the hardest unknown input, G2. With an even inversion parity, intermediate objective: (G2, 0).
   (c) Since G2 is an imply gate, keep backtrace to the hardest unknown input, C. With an even inversion parity, intermediate objective: (C, 0).
   (d) Since C is not a gate, stop backtracing. Assign 0 to C. Simulation, objective achieved. Implication: G2 = 0, G4 = 0, G5 = D, G7 = D. The fault has reached the primary output.
5. Test generated, (A, B, C) = (1, 1, 0).

Answer: (A, B, C) = (1, 1, 0).

3. (15%) Derive the effective test set for G5 SA0 by PODEM with Unguided Backtracing

Solution: We intentionally don't follow the controllability guidance in this example.

1. Activate G5 SA0. Initial objective: (G5, D).

2. (a) Since G5 is an imply gate, instead of backtrace to the hardest unknown input, G1, with CY1(G1) = 0.25, we backtrace to G4, with CY1(G4) = 0.656. With an odd inversion parity, intermediate objective: (G4, 0).
   (b) Since G4 is an imply gate, instead of keep backtrace to the hardest unknown input, G3, with CY0(G3) = 0.75, we backtrace to G2, with CY0(G2) = 0.875. With an even inversion parity, intermediate objective: (G2, 0).
   (c) Since G2 is a decision gate, we back trace to A, (CY0(A) = CY0(B) = CY0(C) = 0.5, so we may choose either one). With an even inversion parity, intermediate objective: (A, 0).
   (d) Since A is not a gate, stop backtracing. Assign 0 to A. Simulation, objective not achieved. Implication: G1 = 0, ... , G5 = 0, conflict with the objective.
   (e) Backtrack and flip A to 1. Simulation, objective not achieved. Implication: G3 = 0;

3. (a) Since G5 is an imply gate, instead of backtrace to the hardest unknown input, G1, with CY1(G1) = 0.25, we backtrace to G4, with CY1(G4) = 0.656. With an odd inversion parity, intermediate objective: (G4, 0).
   (b) Since G4 is an imply gate, keep backtrace to the hardest unknown input, G2. With an even inversion parity, intermediate objective: (G2, 0).
   (c) Since G2 is a decision gate, we back trace to B. With an even inversion parity, intermediate objective: (B, 0).
   (d) Since B is not a gate, stop backtracing. Assign 0 to B. Simulation, objective not achieved. Implication: G1 = 1, ... , G5 = 0, conflict with the objective.
   (e) Backtrack and flip B to 1. Simulation, objective not achieved. Implication: G1 = 1, G6 = 0.

4. (a) Since G5 is an imply gate, backtrace to the hardest unknown input, G4. With an odd inversion parity, intermediate objective: (G4, 0).
   (b) Since G4 is an imply gate, keep backtrace to the hardest unknown input, G2. With an even inversion parity, intermediate objective: (G2, 0).
   (c) Since G2 is a decision gate, we back trace to C. With an even inversion parity, intermediate objective: (C, 0).
   (d) Since C is not a gate, stop backtracing. Assign 0 to C. Simulation, objective achieved. Implication: G2 = 0, G4 = 0, G5 = D, G7 = D. The fault has reached the primary output.

5. Test generated, (A, B, C) = (1, 1, 0).

Answer: (A, B, C) = (1, 1, 0).

4. (15%) Please compare the above (1), (2), (3). [Chapter 10, slides 7-23]

   (a) (7%) What are the differences between (1) and (2)/(3)?

Solution:

(1) do not check the controllability of each line, so it don't make intermediate objectives. It just sets the primary input based on the inversion parity of the entire backtracing path.

(2)/(3) utilize the controllability of each line to choose the backtracing path. They thus set the intermediate objectives in order to apply the heuristics. These heuristics reduce the possibility of encountering a conflict in comparison to (1).

   (b) (8%) What are the differences between (2) and (3)?

Solution:

(2) follows the guidance of the controllability measures. It encounters zero conflict and thus performs no backtracking.

(3) intentionally disobeys the guidance of the controllability measures. It encounters two conflict and thus performs two backtracking. The empirical result shows the heuristic does reduce the possibility of encountering a conflict.

5. (15%) What are the difference between Backtracking and Backtracing? Please show by example.

Solution:

A backtracing procedure maps the objective into a PI assignment that is likely to contribute to achieve the objective, while a backtracking procedure goes back to the last decision when encountering a conflict.

Take the circuit above as an example, we backtrace to the primary inputs until they satisfy the given objective in the simulation. After each backtracing, if the assignment to the primary input conflicts with the objective, we undo the assignment and try another test pattern.

6. (25%) FAN (See Figure 2)

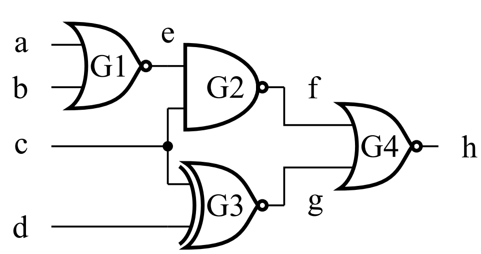

Figure 2

(a) (15%) Generate a test for the fault f-sa1 by using FAN algorithm.

Solution:

1. Identify headlines: c, d, e.
2. Activate f-sa1. Assign D' to f.
3. Backward implication: c = e = 1, a = b = 0.
4. Since G4 is an unique D-frontier, perform unique sensitization. Assign 0 to g.
5. Backward implication: d = 0.
6. Forward implication: h = D. The fault has reached the primary output.
7. No justification needed. Test generated, (a, b, c, d) = (0, 0, 1, 0).

Answer: (a, b, c, d) = (0, 0, 1, 0).

(b) (10%) What are the differences between PODEM and FAN?

Solution:

	PODEM	**FAN**
Assignment location	primary inputs	headlines (internal lines)
Conflict location	primary inputs	headlines (internal lines)
Justification	no	headlines to the primary inputs
Implication	forward only	forward & backward
Backtracing	single backtracing to the primary input (DFS)	multiple parallel backtracing to the headlines (BFS)
Unique sensitization	no	yes

Table 1: The differences between PODEM and FAN.