CSE435 Introduction to EDA & Testing - Spring 2022
Homework Assignment #4
Shao-Hsuan Chu - B073040018

1. (25%)

   (a) (10%) For state transition fault model, explain why there are M(N-1) faults for a M-transition N-state machine. Similarly explain why there are N^M-1 multiple state transition faults.

   > **Solution:** Denote M and N as the number of transitions and states, respectively.
   >
   > **Single-state-transition (SST) fault** For each transition, the possible destination could be either one of the states, so there are (N-1) faulty cases and 1 fault-free case. The SST could occur at either one of the transitions, producing M(N-1) distinguishable faults.
   >
   > **Multiple-state-transition (MST) fault** For each transition, the possible destination could be either one of the states, so there are N cases. Since each transition is independent to other transitions, there are N^M distinguishable state-transition diagrams. Disregarding 1 fault-free diagram, the number of faults in MST is thus N^M-1.

   (b) (10%) For stuck-at fault model, explain why there are 2K single stuck-at faults. Similarly explain why there are 3^K-1 multiple stuck-at faults.

   > **Solution:** Denote K as the number of lines in the circuit.
   >
   > **Single-stuck-at (SSA) fault** The SSA fault would occur at either one of the lines, and in the SSA faults, there will be the case of stuck-at-0 and stuck-at-1. So the number of distinguishable SSA faults would be 2K.
   >
   > **Multiple-stuck-at (MSA) fault** For each lines, there are three possible cases: no-error, stuck-at-0 or stuck-at-1. With K lines, there would be 3^K possible circuits, including 3^K-1 faulty ones and 1 fault-free one.

   (c) (5%) Please show the similarity and differences of (single, multiple) fault numbers between the state transition fault model and the stuck-at fault model.

   > **Solution:**

> **Single** Both the numbers of faults of SST and SSA fault are multiples of the numbers of transitions and lines, respectively. However, the multiplier of the SST fault depends on the number of states, while the one of the SSA is fixed to 2 (sa0, sa1).
>
> **Multiple** In the power of both the numbers of faults of SST and SSA fault. The exponents are the numbers of transitions and lines, respectively. However, the base of the MST fault depends on the number of states, while the one of the MSA is fixed to 3 (no-error, sa0, sa1).

2. (20%) Prove that for combinational circuits **faults dominance is a transitive relation**, i.e. if f dominates g and g dominates h, then f dominates h.

> **Solution:** A fault α is said to dominate another fault β in an irredundant circuit, iff every test for β is also a test for α. If f dominates g, then every test for g is also a test for f. If g dominates h, then every test for h is also a test for g. Since every test for g is a test for f, every test for h is thus also a test for f.

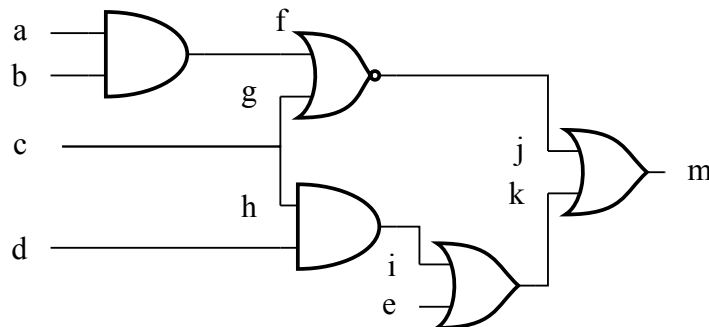3. (55%) In the circuit shown in Figure 1,

Figure 1

(a) (5%) How many single stuck-at faults needed to be considered initially?

> **Solution:** There are 12 lines in the circuit. Therefore, there are $2 \times 12 = 24$ SSA faults initially.

(b) (25%) Applying the **check point theorem (incl. fault dominance)**, how many check point faults needed to be considered?

> **Solution:** Primary inputs and fanout branches form a sufficient set of checkpoints in an irredundant combinational circuit. Therefore, there are 7 checkpoints in this circuit, as shown in Figure 2.
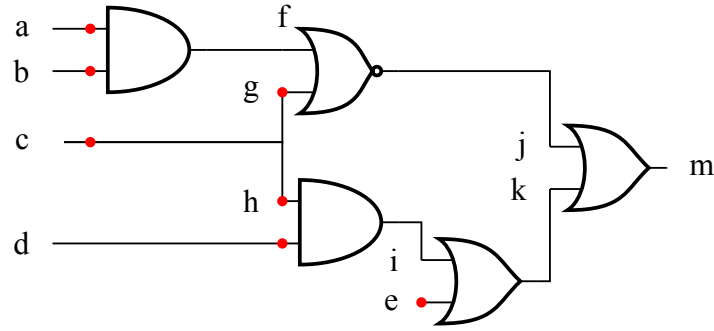
Figure 2

These 7 checkpoints can produce $2 \times 7 = 14$ SSA faults. To further reduce the number of faults, we can use fault collapsing method, including fault equivalence and fault dominance.
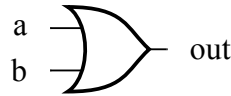

Figure 3

Fault equivalence   For OR gate shown in Figure 3 as example, to test the stuck-at-1 (sa1) fault at a, denoted as a-sa1, we must force logic-0 at a to activate the fault, and we also need the other input (b) as non-controlling value, which is 0 for OR gate. The test vector is thus (0, 0), which is also the test vector for sa1 at b, denoted as b-sa1. To test sa1 at output, we only need to make all the inputs of OR gate logic-0 to activate the fault. We can observe that all the test vectors for a-sa1, b-sa1 and out-sa1 are identical, so the OR gate has an Equivalence Set of a-sa1, b-sa1, out-sa1, indicating we can only keep one fault and collapse others.

Fault dominance   For OR gate shown in Figure 3 as example, to test the stuck-at-0 (sa0) fault at a, denoted as a-sa0, we must force logic-1 at a to activate the fault, and we also need the other input (b) as non-controlling value, which is 0 for OR gate. The test vector is thus (1, 0). Similarly, the test vector for sa0 at b, denoted as b-sa0, is (0, 1). To test sa0 at output, we need to make one of the inputs of OR gate logic-1 to activate the fault. The test vector is thus $(1, \times)$ or $(\times, 1)$. We can observe that the test vector for out-sa0 covers the ones for a-sa0 and b-sa0, i.e., out-sa0 dominates a-sa0 and b-sa0, so the OR gate has a Dominance Set of out-sa0: a-sa0, b-sa0, indicating we can collapse the dominator since the test inputs for the others are sufficient.

Table 1 shows the equivalence sets and dominance sets for common gates.

Gate	Equivalence Set(s)	Dominance Set(s)
OR	{a-sa1, b-sa1, out-sa1}	{out-sa0: a-sa0, b-sa0}
NOR	{a-sa1, b-sa1, out-sa0}	{out-sa1: a-sa0, b-sa0}
AND	{a-sa0, b-sa0, out-sa0}	{out-sa1: a-sa1, b-sa1}
NAND	{a-sa0, b-sa0, out-sa1}	{out-sa0: a-sa1, b-sa1}
NOT	{in-sa1, out-sa0} {in-sa0, out-sa1}	null
Buffer	{in-sa0, out-sa0} {in-sa1, out-sa1}	null

Table 1: The equivalence sets and dominance sets for common gates.

For Figure 4-8 below, the red arrows indicate the necessary stuck-at faults, while the black ones indicates the collapsed faults after each steps.
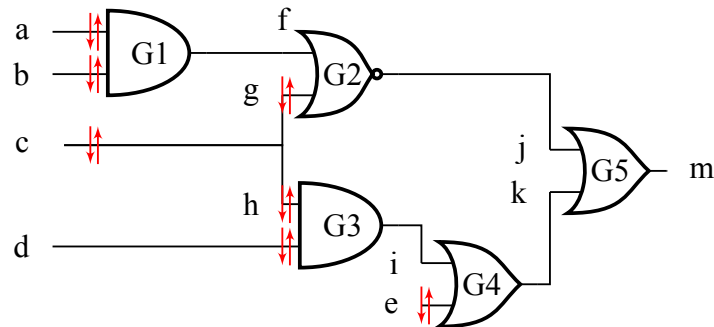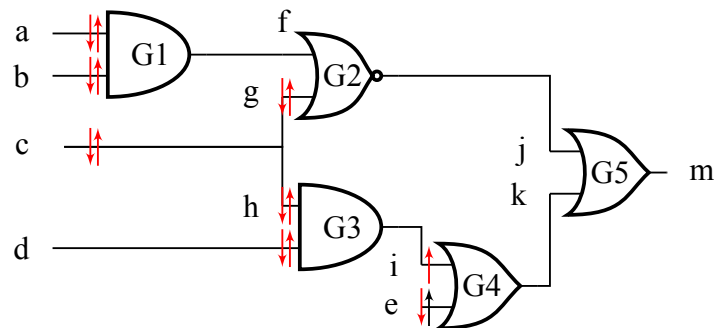

Figure 4: Initial 14 stuck-at faults.

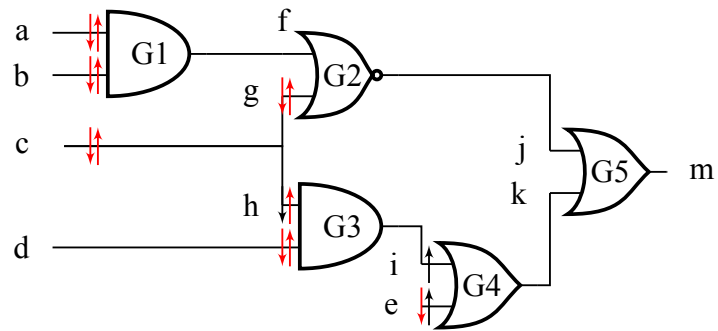Figure 5: At G4, e-sa1 is equivalent to i-sa1.

Figure 6: At G3, i-sa1 dominates h-sa1 and d-sa1. h-sa0 is equivalent to d-sa0.
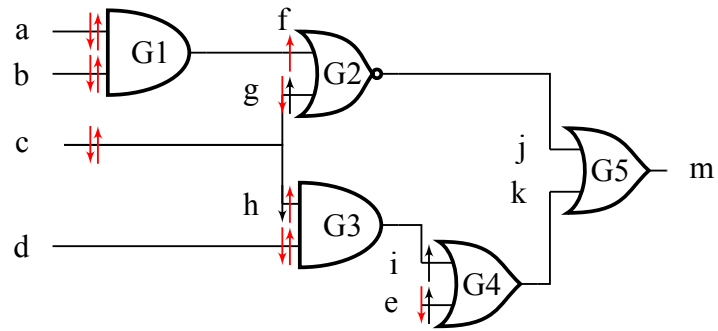
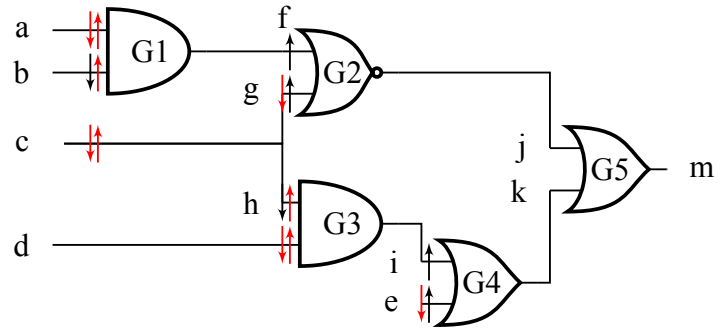
Figure 7: At G2, g-sa1 is equivalent to f-sa1.

Figure 8: At G1, f-sa1 dominates a-sa1 and b-sa1. b-sa0 is equivalent to a-sa0.

(c) (25%) Using **fault dominance** and **fault equivalence** relations to further reduce the number of stuck-at faults? How many remaining faults needed to be considered?

Solution: