Q1. *(15 points)* Solve Problem 4.6 (page 127). Hint: make use of truth tables.

**Answer:**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>C</th>
<th>fault-free</th>
<th>c s-a-1</th>
<th>a s-a-0</th>
<th>{c s-a-1,a s-a-0}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. No test can detect the fault.
b. The test is 110.
c. No test can detect the multiple fault.
Q2. *(15 points)* Solve Problem 5.3 (page 177 and 140).

**Answer:**

For the test 11010, the fault-free signal values are shown below.

The fault lists are computed as follows:

$L_a = \{a_0\}, L_b = \emptyset, L_c = \{c_1\}, L_d = \emptyset, L_e = \emptyset; \ (b_0, d_0, e_1$ are not simulated.)

$L_f = L_a \cup L_b = \{a_0\}, L_g = L_c = \{c_1\}, L_h = L_c \cup \{h_1\} = \{c_1, h_1\}; \ (g_1$ is not simulated.)

$L_j = L_f - L_g = \{a_0\}, L_i = L_h - L_d = \{c_1, h_1\};

L_k = L_i \cup L_e = \{c_1, h_1\};$

$L_m = L_j \cup L_k = \{a_0, c_1, h_1\}.

So 11010 will detect $a_0, c_1, \text{and } h_1.$
Q3. (20 points) Solve Problem 5.9 (page 179).

Answer:

a. Checkpoint faults: 2 each for \{a, b, c, b_1, b_2, d_1, d_2, e_1, e_2\}. Let’s use a superscript \( x \) to refer to the s-a-x fault. After fault collapsing, we have 14 faults:

\[ \{a^0, a^1, b^0, b^1, c^0, c^1, b_1^1, d_1^1, e_1^1, b_2^0, d_2^1, e_2^0\} \]

We name the gates as follows:

Here are the progress of concurrent fault simulation.

For gate D,

* Fault-free: \( AND(a, b) = AND(1, 1) \Rightarrow 1 \).
* Local faults: \( a^0 AND(0, 1) \Rightarrow 0, a^1 AND(1, 1) \Rightarrow 1, b^0 AND(1, 0) \Rightarrow 0, b^1 AND(1, 1) \Rightarrow 1, b_1^1 AND(1, 1) \Rightarrow 1 \).
* So \( a^0 \) and \( b^0 \) will be propagated to \( d \).

For gate E,

* Fault-free: \( NAND(b, c) = NAND(1, 1) \Rightarrow 0 \).
* Local faults: \( b^0 NAND(0, 1) \Rightarrow 1, b^1 NAND(1, 1) \Rightarrow 0, c^0 NAND(1, 0) \Rightarrow 1, c^1 NAND(1, 1) \Rightarrow 0, b_2^1 NAND(1, 1) \Rightarrow 0 \).
* So \( b^0 \) and \( c^0 \) will be propagated to \( e \).

For gate F,

* Fault-free: \( NAND(d, e) = NAND(1, 0) \Rightarrow 1 \).
* Local faults: \( d_1^0 NAND(0, 0) \Rightarrow 1, d_1^1 NAND(1, 0) \Rightarrow 1, e_1^1 NAND(1, 1) = 0 \).
* Propagated faults: \( a^0 NAND(0, 0) \Rightarrow 1, b^0 NAND(0, 1) \Rightarrow 1 \) (both \( d \) and \( e \) are faulty), \( c^0 NAND(1, 1) \Rightarrow 0 \).
* So \( e_1^1 \) and \( c^0 \) will be propagated to \( f \).

For gate G,

* Fault-free: \( OR(d, e) = OR(1, 0) \Rightarrow 1 \).
* Local faults: $d_2^0 \text{OR}(0,0) = \rightarrow 0, d_2^0 \text{OR}(1,0) = \rightarrow 1, e_2^0 \text{OR}(1,0) = 1$.
* Propagated faults: $a^0 \text{OR}(0,0) = \rightarrow 0, b^0 \text{OR}(0,1) = \rightarrow 1$ (both $d$ and $e$ are faulty), $c^0 \text{OR}(1,1) = \rightarrow 1$.
* So $d_2^0$ and $a^0$ will be propagated to $g$.

For gate $H$,
* Fault-free: $\text{AND}(f, g) = \text{AND}(1,1) = \rightarrow 1$.
* No local faults.
* Propagated faults: $e_1^1 \text{AND}(0,1) = \rightarrow 0, c^0 \text{AND}(0,1) = \rightarrow 0, d_2^0 \text{AND}(1,0) = \rightarrow 0, a^0 \text{AND}(1,0) = \rightarrow 0$.
* So $e_1^1$, $c^0$, $d_2^0$ and $a^0$ will be propagated to $h$ and are detected by the test 111.

b. First of all, we need to identify FFRs as follows.
1. Inputs $d_1, e_1, e_2, d_2$, output $h$.
2. Inputs $a, b_1$, output $d$.
3. Inputs $b_2, c$, output $e$.
4. Input/output $b$.

The execution trace (similar to Fig. 5.31, p164):

<table>
<thead>
<tr>
<th>FFR traced</th>
<th>Critical lines</th>
<th>Stems to check</th>
<th>Stem checked</th>
<th>Capture line</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>$h, f, e_1, g, d_2$</td>
<td>$d, e$</td>
<td>$e$</td>
<td>$d$</td>
</tr>
<tr>
<td>$d$</td>
<td>$d, a, b_1$</td>
<td>$e, b$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e$</td>
<td>$e, c, b_2$</td>
<td>$b$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\emptyset$</td>
<td>$b$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

So the test 111 can detect $h^0, f^0, g^0, e_1^1, d_2^0, d^0, a^0, b_1^0, e^1, c^0, b_2^0$. 