COE 571 Dr. El-Malek

HW#2 Solution

Q1. Based on Williams & Brown model:

$$DL = 1 - Y^{(1-Q)}$$

$$\Rightarrow 0.001 = 1 - (0.7)^{(1-Q)}$$

$$\Rightarrow 0.7^{(1-Q)} = 1 - 0.001 = 0.999$$

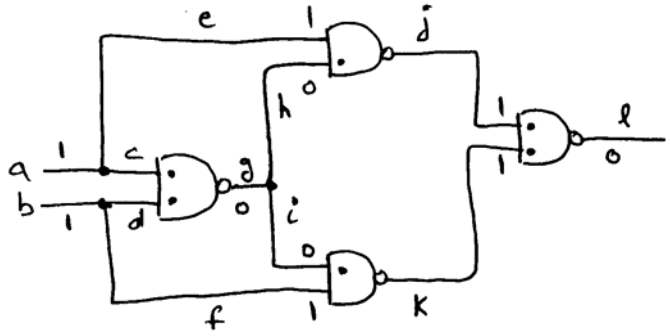
$$\Rightarrow 1-Q = \log 0.999 / \log 0.7$$

$$\Rightarrow 1-Q = 0.0028$$

$$\Rightarrow Q = 0.9972$$

Thus, the required fault coverage to achieve a defect level of 0.001 with yield = 70% is 99.72%

Q2.



Fault list = {$a_0, a_1, b_0, b_1, c_1, d_1, g_0, g_1, e_1, h_1, i_1, f_1, j_1, K_1, l_1, l_0$}

(i) Deductive fault simulation

$L_a = \{a_0\}$, $L_b = \{b_0\}$

$L_c = L_a = \{a_0\}$, $L_d = L_b = \{b_0\}$

$L_e = L_a = \{a_0\}$, $L_f = L_b = \{b_0\}$

$L_g = \{g_1\} \cup L_c \cup L_d = \{g_1, a_0, b_0\}$

$L_h = \{h_1\} \cup L_g = \{h_1, g_1, a_0, b_0\}$

$L_i = \{i_1\} \cup L_g = \{i_1, g_1, a_0, b_0\}$

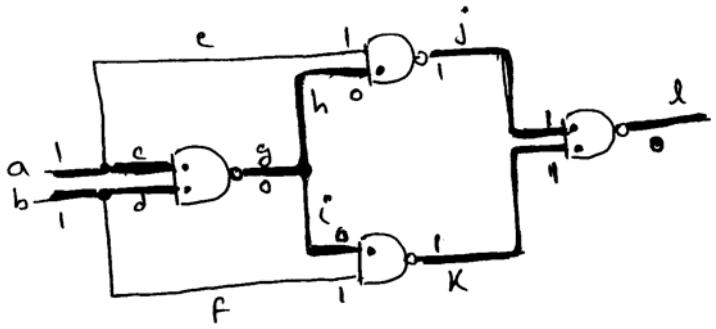
$L_j = L_h - L_e = \{h_1, g_1, b_0\}$

$L_K = L_i - L_f = \{i_1, g_1, a_0\}$

$L_l = \{l_1\} \cup L_j \cup L_K = \{l_1, h_1, g_1, b_0, i_1, a_0\}$

Thus, there are 6 detected faults by the test vector 11.

(ii) Critical Path Tracing



The critical lines are shown highlighted.

FFR	critical Lines	stems to check	stem checked	capture Lines
l	l, j, k, h, i	g	g	l
g	g, c, d	a, b	a	g or k or l
a	a	b	b	g or j or l
b	b	∅		

Details of checking criticality of stems:

1. Stem g:

Frontier = {j, k}

since both j & k are of the same level, any one of them can be picked.

If j is picked first, it is found that the fault propagates across j and Frontier = {k, l}. Next, k is picked since it is of a lower level. Since the fault propagates across k, Frontier = {l}. Since the fault propagates across l and l is critical, then g is marked as critical. Thus, l is a capture line for g.

2. Stem a:

Frontier = {g, j}

since both gates are of the same level, any one of them can be picked.

If j is picked, then it is found that j does not propagate the fault and Frontier = {g} since g propagates the fault and g is critical, a is marked as critical and g is a capture line.

However, if g is picked first, then Frontier = {j, k} since g propagates the fault. Next, either j or k could be picked since they are at the same level.

If j is picked, then Frontier = {k} since j does not propagate the fault. Since k propagates the fault, then a is marked as critical since k is critical. Thus, k is a capture line for a.

However, if k is picked first, then Frontier = {j, l}. Since j does not propagate the fault, then Frontier = {l}. Since l propagates the fault and is critical, then a is critical and l is a capture line.

3. Stem b:

The same analysis done for stem b will result in the conclusion that b is critical and that either g or j or l are capture lines.

(iii) Parallel Fault Simulation:

Fault Representation:

a0	a1	b0	b1	c1	d1	g0	g1	e1	h1	i1	f1	j1	k1	l0	l1	ff

For Line **a**:

Ia:

1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sa:

0	1															

Line **a** before fault injection:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Line **a** after fault injection:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

For Line **b**:

Ib:

0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Sb:

		0	1													

Line **b** before fault injection:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Line **b** after fault injection:

1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

For Line **c**:

Ic:

0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Sc:

				1												

Line **c** before fault injection:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Line **c** after fault injection:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

For Line **d**:

Id:

0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Sd:

					1											

Line **d** before fault injection:

1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Line **d** after fault injection:

1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

For Line **g**:

Ig:

0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Sg:

						0	1									

Line **g** before fault injection:

1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Line **g** after fault injection:

1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

For Line **e**:

Ie:

0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Se:

								1								

Line **e** before fault injection:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Line **e** after fault injection:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

For Line **h**:

Ih:

0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Sh:

									1							

Line **h** before fault injection:

1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Line **h** after fault injection:

1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0

For Line **i**:

Ii:

0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Si:

										1						

Line **i** before fault injection:

1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Line **i** after fault injection:

1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0

For Line **f**:

If:

0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Sf:

											1					

Line **f** before fault injection:

1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Line **f** after fault injection:

1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

For Line **j**:

Ij:

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Sj:

												1				

Line **j** before fault injection:

1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1

Line **j** after fault injection:

1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1

For Line **k**:

Ik:

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Sk:

													1			

Line **k** before fault injection:

0	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1

Line **k** after fault injection:

0	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1

For Line **l**:

Il:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Sl:

														0	1	

Line **l** before fault injection:

1	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0

Line **l** after fault injection:

1	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	0

a0	a1	b0	b1	c1	d1	g0	g1	e1	h1	i1	f1	j1	k1	l0	l1	ff

Since l is a primary outputs, the detected faults are those that produce different values than the fault free value which include: {a0. b0, g1, h1, i1, l1}. This is consistent with the results obtained by deductive fault simulation and critical path tracing.

(iv) Verification using CPT:

Netlist:

```
INPUT(a)
INPUT(b)
OUTPUT(l)

g   = NAND(a, b)
j   = NAND(a, g)
k   = NAND(g, b)
l   = NAND(j, k)
```

CPT Fault Simulation Results:

```
Test #   1:     6 faults detected (vs 6 from CPT)
HOPE: newly detected faults are:
        l /1
        g->k /1
        g->j /1
        a /0
        g /1
        b /0
CPT: newly detected faults are:
      l /1 through PO l_PO
      g->k /1 through PO l_PO
      g->j /1 through PO l_PO
      a /0 through PO l_PO
      g /1 through PO l_PO
      b /0 through PO l_PO
```

(v) Verification using PROOFS:

Circuit Netlist	**Signal Names**
INPUT(a) INPUT(b) OUTPUT(l) g = NAND(a, b) j = NAND(a, g) k = NAND(g, b) l = NAND(j, k)	1 a 2 b 3 g 4 j 5 k 6 l 7 l_$OUTPUT
Fault Simulation Statistics	Detected Faults
user time 0.000000 sec sys time 0.000000 sec det faults 6 tot faults 16 coverage 0.3750000	7 0 1; 5 1 1; 3 0 1; 2 0 0; 1 0 0; 4 1 1;