

Solutions to Homework 9 Problems

ELEC 7250 VLSI Testing (Spring 2006)

April 15, 2006

Problem 9.13 Dynamic coupling faults

We rigorously prove that the MARCH C- test detects all dynamic coupling faults. MARCH C- test is,

$$\{ M0 : \updownarrow (w0); M1 : \uparrow (r0, w1); M2 : \uparrow (r1, w0); \\ M3 : \downarrow (r0, w1); M4 : \downarrow (r1, w0); M5 : \updownarrow (r0) \}$$

and dynamic coupling faults are $\langle r0|w0;0 \rangle$, $\langle r0|w0;1 \rangle$, $\langle r1|w1;0 \rangle$ and $\langle r1|w1;1 \rangle$.

Necessary condition: After initializing the coupled cell, a read (write) of the coupling cell must be followed by a read of the coupled cell, without any intervening operations on the coupled cell.

Fault $\langle r0|w0;0 \rangle$: Address of coupled cell $i >$ Address of coupling cell j .

For a write,

i initialized by M1, j written by M2,

i checked by M2 (fault detected).

For a read,

i initialized by M3, j read by M3,

i checked by M4 (fault detected).

Address of coupled cell $i <$ Address of coupling cell j .

For a write,

i initialized by M3, j written by M4,

i checked by M4 (fault detected).

For a read,

i initialized by M3, j read by M4,

i checked by M4 (fault detected).

Fault $\langle r0|w0;1 \rangle$: Address of coupled cell $i >$ Address of coupling cell j .

For a write,

i initialized by M4, j written by M4,

i checked by M5 (fault detected).
 For a read,
 i initialized by M0, j read by M1,
 i checked by M1 (fault detected).
Address of coupled cell $i < \text{Address of coupling cell } j$.
 For a write,
 i initialized by M2, j written by M2
 i checked by M3 (fault detected).
 For a read,
 i initialized by M2, j read by M3,
 i checked by M3 (fault detected).

Fault $\langle r1|w1;0 \rangle$: *Address of coupled cell $i > \text{Address of coupling cell } j$.*

For a write,
 i initialized by M1, j written by M1,
 i checked by M2 (fault detected).
 For a read,
 i initialized by M1, j read by M2,
 i checked by M2 (fault detected).
Address of coupled cell $i < \text{Address of coupling cell } j$.
 For a write,
 i initialized by M3, j written by M3
 i checked by M4 (fault detected).
 For a read,
 i initialized by M4, j read by M4,
 i checked by M4 (fault detected).

Fault $\langle r1|w1;1 \rangle$: *Address of coupled cell $i > \text{Address of coupling cell } j$.*

For a write,
 i initialized by M0, j written by M1,
 i checked by M1 (fault detected).
 For a read,
 i initialized by M0, j read by M2,
 i checked by M2 (fault detected).
Address of coupled cell $i < \text{Address of coupling cell } j$.
 For a write,
 i initialized by M2, j written by M3
 i checked by M3 (fault detected).
 For a read,
 i initialized by M2, j read by M2,
 i checked by M3 (fault detected).

That completes the proof. ■

Problem 11.1 Parametric faults

For the circuit of Figure 11.10,

$$\begin{aligned}V_{out} &= V_{in} + \frac{1}{C + \frac{1}{R_f}} \int_0^t \frac{V_{in}}{R_1} dt \\ &= \frac{V_{in}t}{R_1C + \frac{R_1}{R_f}}\end{aligned}$$

This is an integrator or low-pass filter, with parameters

f_c - filter cutoff frequency: C , R_f

A - AC voltage gain: R_1 , C , R_f

A_4 - DC voltage gain: R_1 , R_f

There are no useful single parametric faults, two useful double parametric faults, and one useful triple parametric fault.

Problem 12.9 Path counting

We consider the combinational circuit as a directed graph with PIs, POs and gates as vertices, and the arcs drawn according to connectivity. We add two vertices, a vertex named *source* from which arcs are directed to all PI vertices, and a *sink* vertex to which arcs are directed from all PO vertices. Each vertex v is given a label, $N(v)$, whose value denotes the number of paths from *source* to v . The path counting algorithm is as follows:

1. *Initialization*: Set all labels to 0. Update $N(\text{source}) = 1$.
2. *Count*: Update each vertex only after all of its fanin vertices have been updated. Update of vertex v is done as follows:

$$N(v) = \sum_{i=1}^k N(v_i)$$

where v_1, v_2, \dots, v_k are the fanin vertices of v .

3. *Result*: $N(\text{sink}) =$ number of paths in the circuit.

Complexity: Since each vertex is processed once, there are n updates, where n is the number of vertices in the graph. Each update requires adding the labels of the fanin vertices. An upper bound on fanin is n . Thus, the complexity of the path counting algorithm is $O(n^2)$, where $n = PI + PO + gates + 2$. In general, however, the fanin of a gate does not grow with the number of gates, and the complexity remains closer to $O(n)$.