Copyright 1999 IEEE. Published in the Records of MTDT'99, August 1999 San Jose, California. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

Modeling and Testing Transistor Faults in Content-Addressable Memories*

Piotr R. Sidorowicz
Department of Computer Science
University of Waterloo
Waterloo, ON, Canada, N2L 3G1
prsidoro@maveric.uwaterloo.ca

Abstract

A behavioral analysis of transistor faults and cell-stuck-at faults in a n-word by l-bit static CMOS CAM array is presented. First, a CAM cell is analyzed at the transistor-network, event-sequence and finite-state machine (FSM) level. Then, a transistor on/open and cell-stuck-at fault model for a CAM is defined. We show that two out of eighteen possible CAM cell's transistor faults cannot be tested reliably by functional tests; however, among faults that are testable, are all those which comprise data-retention faults. We also show that a test, originally designed to detect input stuck-at faults, also detects all reliably testable transistor faults and all cell-stuck-at faults in the model.

1 Introduction

CAMs are word-oriented storage devices that allow instantaneous searches of their content for a given key word. Their increasing utilization in mainstream integrated circuit designs mandates the generation of reliable and efficient tests for these memories.

Transistor faults constitute an important class of faults that occur in static memory cells. They are a result of defects that render transistors inoperable, in the sense that these transistors either always conduct or never conduct. These two types of transistor faults are known as *stuck-on* and *stuck-open* faults, respectively. Depending on the location of the faulty transistor, such faults may cause spontaneous changes of the cell's state that are slow enough not to be detected by typical testing algorithms and thus are classified as *data retention* faults. Transistor stuck-on faults are more readily detectable, as they result in an increase in the quiescent supply current I_{DDQ}; hence they are detectable by parametric tests. This is not the case with transistor stuck-open faults; several testing methods for transistor stuck-open faults in SRAMs have been proposed [4, 8].

In this paper we investigate the effect of transistor and cell-stuck-at faults on the behavior of a static CAM cell of Fig. 1 in order to establish the coverage of these faults by functional CAM tests.

A systematic approach to modeling and testing memories has been presented in [9]. It includes circuit analysis, asynchronous behavior analysis, and a finite state machine representation of the fault-free and faulty circuits. The basis for this approach to modeling and testing memories is the framework of [2]. This framework has been successfully utilized for modeling common faults in RAMs (See [3, 5, 6], for examples).

The remainder of this paper is organized as follows: The CAM cell and its model are presented in Section 2. Section 3 describes the event-sequence model of a cell's behavior. A higher level, finite-state machine (FSM) model of a cell's behavior is introduced in Section 4. A test by Sidorowicz & Brzozowski [9] is described and its coverage with respect to the new fault model is determined in Sections 5. Section 6 concludes the paper. A detailed analysis of faulty cell operations is presented in Appendix A.

2. CAM Circuit and Model

Figure 1 shows a cell of a common static CAM circuit [1, 7]. It consists of a *storage section* and a *comparison section*. The storage section has two cross-coupled CMOS inverters (T_1 – T_4), differential bit lines (*bit*/$\overline{bit}$) used for reading and writing data into a column of cells, and a word select line (*WL*) that enables these operations in a row of cells via transistors T_5 and T_6. In a quiescent state *WL* is driven low, the *bit*/$\overline{bit}$ lines are driven high and the cell stores either 0 or 1. During a 'write 1' or a 'write 0' operation, the *bit*/$\overline{bit}$ lines are driven to a true/complementary representation of the desired bit value. Raising and then lowering *WL* stores the bit in the cell. The 'write $\times$' operation preserves the state of the cell. Its implementation

*This research was supported by a grant from the Information Technology Research Centre.

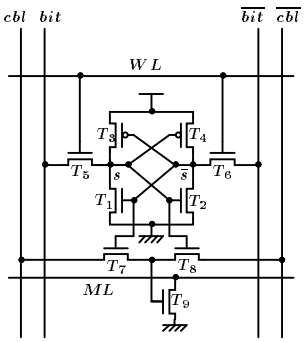


Figure 1. Static CAM cell with dedicated compare lines.

requires both of the *bit*/$\overline{bit}$ lines to be driven high; proper margining of transistors $T_1 - T_4$ assures the preservation of the cell's state. A 'read' operation is done by isolating both *bit*/$\overline{bit}$ lines, and then raising *WL*, thus causing one of the *bit*/$\overline{bit}$ lines to discharge. The voltage differential between the *bit*/$\overline{bit}$ lines is detected giving the cell's content.

The comparison section consists of the transistors T_7, T_8 and T_9, the differential compare bit lines (*cbl*/$\overline{cbl}$) for matching operations, and the match line (*ML*). In a quiescent state *cbl*/$\overline{cbl}$ are driven low, *ML* is driven high and either T_7 or T_8 conducts, depending on the state of the cell. During a 'compare' operation, *ML* is first isolated. Next, *cbl*/$\overline{cbl}$ are driven according to the desired search key. If a mismatch occurs, transistor T_9 will be forced to conduct, thus discharging *ML*. In the case of a 'compare $\times$' operation, both of *cbl*/$\overline{cbl}$ remain low, resulting in an unconditional match.

Altogether seven operations can be performed on this cell: 'read', 'write 0', 'write 1', 'write $\times$', 'compare 0', 'compare 1' and 'compare $\times$'. The *transistor and cell-stuck-at fault model* comprises all the transistor stuck-(on/open) faults of the CAM cell's as well as stuck-at faults of node s ($\bar{s}$), under the single fault assumption.

3. The Event-Sequence Model

Since no clock signal is supplied to individual cells, a single CAM cell can be viewed as an asynchronous sequential circuit, whose behavior can be modeled by the finite state machine of Fig. 2. Since *bit*/$\overline{bit}$ lines and *ML* are used for both input and output in the circuit, they are represented by separate variables in the model. For brevity we use b, $\bar{b}$, w, etc., for *bit*, $\overline{bit}$, *WL*, etc. Now, b, $\bar{b}$ w, c, $\bar{c}$ and m are inputs, and $\widehat{b}$, $\widehat{\bar{b}}$, and $\widehat{m}$ are outputs. The total state of the cell is defined by the values on the input and output lines of the cell, and by its internal state s. However, it turns out that in the correct cell, as well as in the presence of transistor and cell-stuck-at faults, *the output values are identical to those of the inputs*; hence, we omit them. Thus the total state of the cell is formally represented by the 7-tuple: $C = (w, b, \bar{b}, c, \bar{c}, m, s)$, but for notational simplicity, we represent it as $C = w\ b\bar{b}\ c\bar{c}\ m \cdot s$, where the input variables have been separated by spaces for readability, and the $\cdot$ has been inserted to separate the input variables from the internal state.

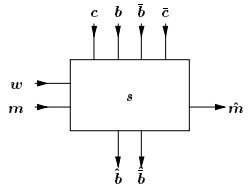


Figure 2. Static CAM cell model.

The domain of each variable in C is the set $Y = \{0, 1, \tilde{0}, \tilde{1}, I\}$. The values 0 and 1 represent lines driven to the logic values 0 and 1 respectively, while $\tilde{0}$ and $\tilde{1}$ denote lines that were first discharged and then isolated (*floated low*) or precharged and then isolated (*floated high*). Symbol I stands for an intermediate logic value, which is caused in a CMOS circuit when both pull-up and pull-down transistors conduct.

A correct CAM cell has two possible initial states: $0\ 11\ 00\ 1 \cdot 0$ and $0\ 11\ 00\ 1 \cdot 1$.

Table 1. A 'compare 1' in a correct and faulty CAM cell.

Compare 1 ($s = 1$)		
	Correct	T_7-*on*
Description	$w\ b\bar{b}\ c\bar{c}\ m \cdot s$	$w\ b\bar{b}\ c\bar{c}\ m \cdot s$
initial state	$0\ 11\ 00\ 1 \cdot 1$	$0\ 11\ 00\ 1 \cdot 1$
float m	$0\ 11\ 00\ \tilde{1} \cdot 1$	$0\ 11\ 00\ \tilde{1} \cdot 1$
set c/$\overline{c}$	$0\ 01\ 00\ \tilde{1} \cdot 1$	$0\ 11\ 00\ \tilde{1} \cdot 1$
m discharges	$0\ 11\ 00\ \tilde{1} \cdot 1$	$0\ 11\ 00\ \mathbf{0} \cdot 1$
read m & ground c/$\overline{c}$	$0\ 11\ 00\ \tilde{1} \cdot 1$	$0\ 11\ 00\ \mathbf{0} \cdot 1$
raise m	$0\ 11\ 00\ 1 \cdot 1$	$0\ 11\ 00\ 1 \cdot 1$

The behavior of a cell is represented by sequences of events that take place during the seven operations. Table 1 lists events that occur during a 'compare 1' operation in a correct and a faulty cell when $s = 1$. By events we mean changes in the value of the total state C. The occurrences of these events are ordered from top to bottom. Note that each operation begins and ends in an initial state. Faulty entries are shown in boldface type. An exhaustive analysis of all operations is given in Appendix A.

Example: 'Compare 1' from state 1 in a correct cell and in the presence of the T_7-*on* fault. Initial state of the cell: $0\ 11\ 00\ 1 \cdot 1$. First, m is floated: $0\ 11\ 00\ \tilde{1} \cdot 1$. Note that both transistors T_7 and T_8 in the faulty cell are on. Then c is raised: $0\ 11\ 10\ \tilde{1} \cdot 1$. Since in the correct cell T_7 is off, T_9 does not conduct and m remains floating : $0\ 11\ 10\ \tilde{1} \cdot 1$. In the faulty cell, the conducting transistors T_7 and T_8 act like

a voltage divider, and hence, the voltage on the gate of T_9 will rise high enough to discharge m : 1 11 10 $\mathbf{0} \cdot 1$. Next, c is lowered: 0 11 00 $\tilde{1} \cdot 1$ (correct), 0 11 00 $\mathbf{0} \cdot 1$ (faulty). Finally, m is raised: 0 11 00 $1 \cdot 1$.

If it were possible to control any input and observe any output, testing would be a matter of comparing the state of each line with its expected value; any disagreement would indicate a fault. Unfortunately, monitoring in such detail is not feasible; thus, a more abstract model is necessary.

4. The FSM Model

Most testing algorithms utilize sequences of 'read', 'write' and 'compare' operations as input, and observe the resulting output. Accordingly, we introduce an high-level model of a CAM cell, derived from the event-sequence model. We represent the behavior of a CAM cell as an FSM of Fig. 3. The input x of this FSM comprises all seven operations of a CAM cell, and the output y comprises the responses to these operations without regard to the output's origin, i.e., the $b/\bar{b}$ lines or m. State s represents the value stored by the cell. An FSM model for all transistor and cell-stuck-at faults has been constructed. An example is shown in Fig. 4.

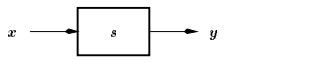
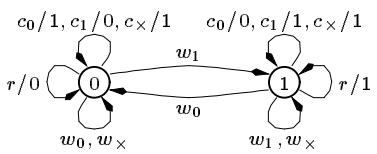
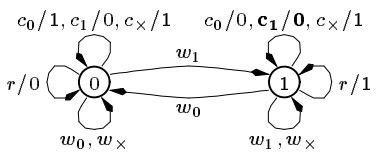


Figure 3. Simplified behavioral model.



(a)



(b)

Figure 4. Cell behavior: (a) correct (b) T_7-*on*.

In a single cell, every fault, except T_3-*on* and T_4-*on*, can be detected by tests of length at most 3. We refer to the shortest test that detects a particular fault in a single cell as an *elementary test*; these tests are essential to the detection of the associated faults. Examples of elementary tests for the reliably testable faults in our fault model are shown in Table 2. The results persented in this table indicate that transistor-open faults that affect a cell's data retention capabilities are easily detectable by functional tests, due to the availability of the 'write $\times$' operation.

Table 2. Summary of transistor and cell-stuck-at faults

Fault type	Test Sequence	Faulty Response	Fault-Free Response
T_1-*on*	$w_1 w_\times c_1$	0	1
T_1-*open*	$w_0 w_\times c_0$	0	1
T_2-*on*	$w_0 w_\times c_0$	0	1
T_2-*open*	$w_1 w_\times c_1$	0	1
T_3-*on*	Not testable reliably		
T_3-*open*	$w_1 w_\times c_1$	0	1
T_4-*on*	Not testable reliably		
T_4-*open*	$w_0 w_\times c_0$	0	1
T_5-*on*	$w_0 c_0$	0	1
T_5-*open*	$w_0 w_1 c_1$	0	1
T_6-*on*	$w_1 c_1$	0	1
T_6-*open*	$w_1 w_0 c_0$	0	1
T_7-*on*	$w_1 c_1$	0	1
T_7-*open*	$w_0 c_1$	1	0
T_8-*on*	$w_0 c_0$	0	1
T_8-*open*	$w_1 c_0$	1	0
T_9-*on*	$c_\times$	0	1
T_9-*open*	$c_1 c_0 / c_0 c_1$	11	10 or 01
s-sa-0	$w_1 c_1$	0	1
s-sa-1	$w_0 c_0$	0	1
$\bar{s}$-*sa-0*	$w_0 c_0$	0	1
$\bar{s}$-*sa-1*	$w_1 c_1$	0	1

5 Determining the Fault Coverage of a Test

A march test $T_{S\&B}$, of length $7n + 2l + 5$, for an $(n \times l)$ CAM was designed to detect stuck-at-0 and stuck-at-1 faults that occur on a cell's input lines [9]. We wish to determine the fault coverage of this test with respect to the transistor and cell-stuck-at fault model.

First, a few words about the test itself. Two assumptions were made about the peripheral circuitry of the fault-free CAM: 1) The address decoder can raise at most a single write line, 2) Match detection is done through a *hit* line, a *multi-hit* line, and an encoder that returns the addr ess k of the highest priority match line[1].

$$
\begin{aligned}
T_{S\&B} = \\
&(w^i_{0\ldots0})^{\updownarrow}(w^i_{1\ldots1} w^i_{\times\ldots\times} c_{1\ldots1})^{n\downarrow 1}[c_{\times^{j-1}0\times^{l-j}}]^{\leftrightarrow} \\
&(w^i_{0\ldots0} w^i_{\times\ldots\times} c_{0\ldots0})^{n\downarrow 1}[c_{\times^{j-1}1\times^{l-j}}]^{\leftrightarrow} \\
&(w^n_{1\ldots1} c_{1\ldots1} w^n_{0\ldots0} w^u_{1\ldots1} c_{1\ldots1}),
\end{aligned}
$$

where $0\ldots0$ denotes the all-0 word, etc., $\times^{j-1}0\times^{l-j}$ denotes the l-bit word with 0 in position j and $\times$ in every other position (the word $\times^{j-1}1\times^{l-j}$ is similarly defined), and $1 \leq u \leq n-1$. The symbol $[c_{\times^{j-1}0\times^{l-j}}]^{\leftrightarrow}$ represents a sequence of l 'compare' operations to words of the

[1] By convention, address 1 has the highest, and address n the lowest priority.

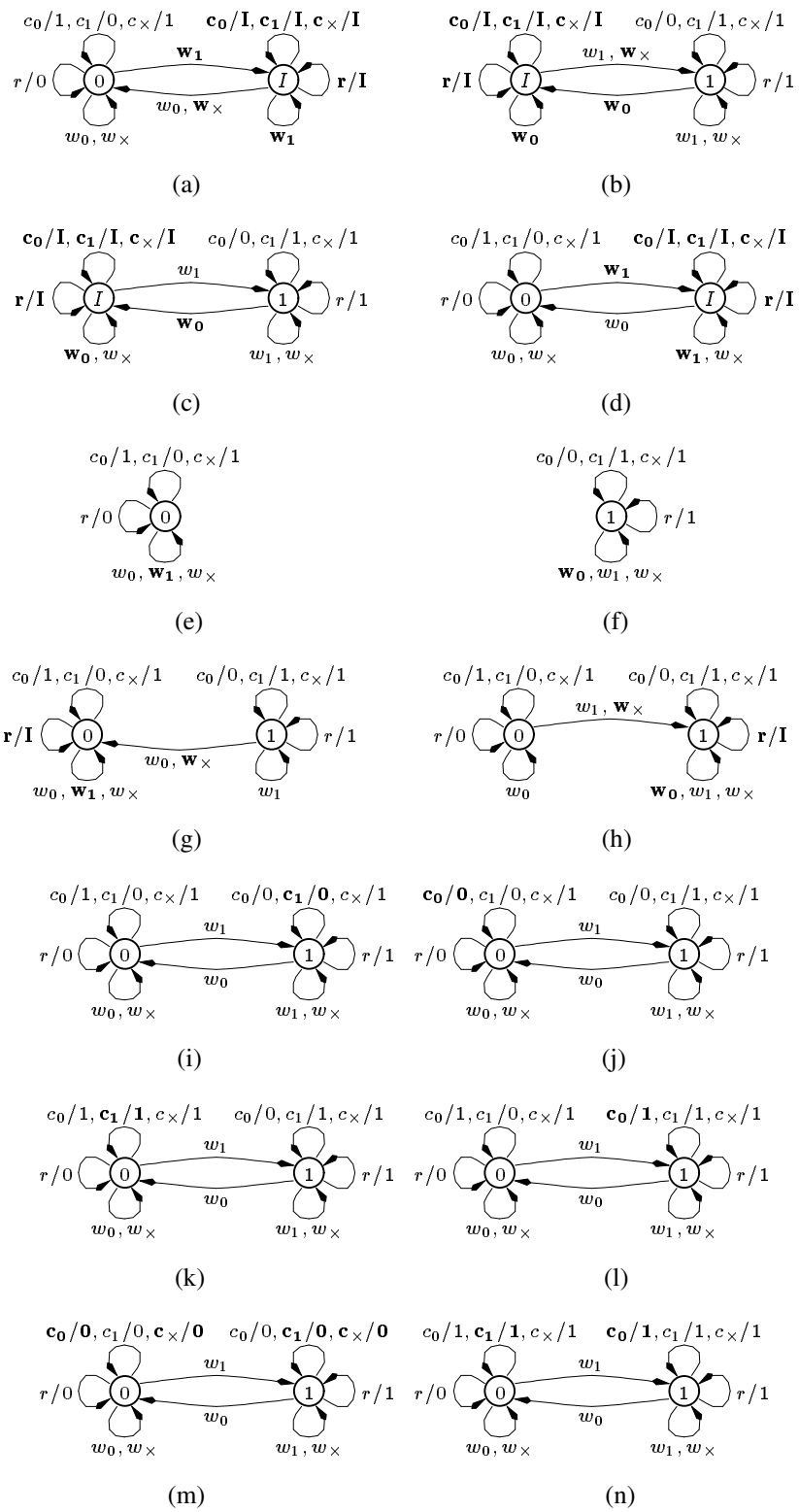


Figure 5. FSM models for faults: (a) T_1*-on*, T_2*-open* **and** T_3*-open*, **(b)** T_1*-open*, T_2*-on* **and** T_4*-open*, **(c)** T_3*-on*, **(d)** T_4*-on*, **(e)** T_6*-on*, *s-sa-0* **and** $\bar{s}$*-sa-1*, **(f)** T_5*-on*, *s-sa-1* **and** $\bar{s}$*-sa-0*, **(g)** T_5*-open*, **(h)** T_6*-open*, **(i)** T_7*-on*, **(j)** T_8*-on*, **(k)** T_7*-open*, **(l)** T_8*-open*, **(m)** T_9*-on*, **(n)** T_9*-open*.

form $\times^{j-1}0\times^{l-j}$, where j varies from 1 to l or from l to 1. The symbol $(\,)^{\updownarrow}$ denotes n operations done in order either from n to 1 or from 1 to n; hence the $\updownarrow$. The symbol $(\,)^{n\downarrow 1}$ denotes the direction of the march elements: from n to 1. This direction is dictated by the priority scheme used in the match line encoder and always follows from the lowest to the highest priority.

The $T_{S\&B}$ test has two parts. Part one consists of an initialization which sets all words to $0\ldots0$, followed by two march elements. For a fault-free CAM, each of the march elements should result in n hits with $k = i$, where k is the output of the match line priority encoder. Each march element should be followed by l mismatches at $k = 0$. Any other response indicates a fault. In part two (last line), multiple hits are monitored, since they indicate faults.

In order to determine the fault coverage of $T_{S\&B}$ test with respect to transistor and cell-stuck-at fault model, we first represent part one of this test from the perspective of an arbitrary cell (i, j) where $1 \le i \le n$ and $0 \le l-1$.

$$T'_{S\&B} = w_0 w_1 w_\times c_1[c_\times]^{l-1-j} c_0[c_\times]^j w_0 w_\times c_0[c_\times]^{l-1-j} c_1[c_\times]^j,$$

where the symbol $[\,]^p$ denotes p repetitions of a given operation. Indices i, j to bit-wise operations are implied, and have been omitted for clarity. Next, we check whether the tests of Table 2 are included in $T'_{S\&B}$. Table 3 indicates their location within in $T'_{S\&B}$ in boldface.

Table 3. Elementary tests within $T'_{S\&B}$.

Fault	Test Sequence for Faults within $T'_{S\&B}$
T_1-*on*	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_1-*open*	$\mathbf{w_0}\mathbf{w_1}\,w_\times c_1[c_\times]\mathbf{c_0}[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
T_2-*on*	$\mathbf{w_0}\mathbf{w_1}\,w_\times c_1[c_\times]\mathbf{c_0}[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
T_2-*open*	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_3-*open*	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_4-*open*	$\mathbf{w_0}\mathbf{w_1}\,w_\times c_1[c_\times]\mathbf{c_0}[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
T_5-*on*	$w_0w_1\,w_\times c_1[c_\times]c_0[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
T_5-*open*	$\mathbf{w_0}\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_6-*on*	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_6-*open*	$w_0\mathbf{w_1}\,w_\times c_1[c_\times]c_0[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
T_7-*on*	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_7-*open*	$w_0w_1\,w_\times c_1[c_\times]c_0[c_\times]\mathbf{w_0}\,w_\times c_0[c_\times]\mathbf{c_1}[c_\times]$
T_8-*on*	$w_0w_1\,w_\times c_1[c_\times]c_0[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
T_8-*open*	$w_0\mathbf{w_1}\,w_\times c_1[c_\times]\mathbf{c_0}[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
T_9-*on*	$w_0w_1\,w_\times c_1[\mathbf{c_\times}]c_0[\mathbf{c_\times}]w_0\,w_\times c_0[\mathbf{c_\times}]c_1[\mathbf{c_\times}]$
T_9-*open*	$w_0w_1\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
s-sa-0	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
s-sa-1	$w_0w_1\,w_\times c_1[c_\times]c_0[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
$\bar{s}$-*sa-0*	$w_0w_1\,w_\times c_1[c_\times]c_0[c_\times]\mathbf{w_0}\,w_\times\mathbf{c_0}[c_\times]c_1[c_\times]$
$\bar{s}$-*sa-1*	$w_0\mathbf{w_1}\,w_\times\mathbf{c_1}[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$

Verification of the fault coverage is given in Table 4, where the faulty responses are listed. The deviations from the fault-free response are indicated in boldface and the fault-free response is provided for comparison. Note that in both tables the index p is implied and was omitted for clarity.

Table 4. Fault coverage verification in $T'_{S\&B}$ test.

Fault	$T'_{\text{S\&B}}$ $w_0\,w_1\,w_\times c_1[c_\times]c_0[c_\times]w_0\,w_\times c_0[c_\times]c_1[c_\times]$
fault-free cell	– – – **1** [**1**] 0 [1] – – **1** [**1**] 0 [1]
T_1-*on*	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]
T_1-*open*	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
T_2-*on*	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
T_2-*open*	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]
T_3-*open*	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]
T_4-*open*	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
T_5-*on*	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
T_5-*open*	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]
T_6-*on*	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]
T_6-*open*	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
T_7-*on*	– – – **0** [**1**] 0 [1] – – **1** [**1**] 0 [1]
T_7-*open*	– – – **1** [**1**] 0 [1] – – **1** [**1**] **1** [1]
T_8-*on*	– – – **1** [**1**] 0 [1] – – **0** [**1**] 0 [1]
T_8-*open*	– – – **1** [**1**] **1** [1] – – **1** [**1**] 0 [1]
T_9-*on*	– – – **0** [**0**] 0 [**0**] – – **0** [**0**] 0 [**0**]
T_9-*open*	– – – **1** [**1**] **1** [1] – – **1** [**1**] **1** [1]
s-sa-0	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]
s-sa-1	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
$\bar{s}$-*sa-0*	– – – **1** [**1**] 0 [1] – – **0** [**1**] **1** [1]
$\bar{s}$-*sa-1*	– – – **0** [**1**] **1** [1] – – **1** [**1**] 0 [1]

From these two tables it is clear that every reliably testable transistor fault of an arbitrary cell in a $(n \times l)$ CAM array is detectable by $T_{S\&B}$.

6 Conclusions

A systematic behavioral analysis of transistor stuck-(on/open) faults and cell-stuck-at faults in a static $(n \times l)$ CMOS CAM was performed. Elementary tests, necessary to detect these faults in an arbitrary CAM cell were given. A test, originally designed for a different fault model, was determined to detect all the reliably testable faults in the above fault model. It was shown that transistor-open faults that compromise data retention in static memory cells can be easily detected by functional CAM tests, due to the enhanced functionality of this type of memory.

Acknowledgments

I thank Manoj Sachdev of the Department of Electrical and Computer Engineering at University of Waterloo for providing extensive information regarding various defects and their effects on the behavior of the CAM. I also thank my supervisor Janusz Brzozowski for his insightful comments and suggestions regarding this paper.

References

[1] H. Bergh, J. Eneland, and L.-E. Lundström. A fault-tolerant associative memory with high-speed operation. *IEEE Journal of Solid-State Circuits*, 25(4):912–919, Aug. 1990.

[2] J. A. Brzozowski and H. Jürgensen. A model for sequential machine testing and diagnosis. *Journal of Electronic Testing: Theory and Applications*, 3:219–234, 1992.

[3] B. F. Cockburn and J. A. Brzozowski. Near-optimal tests for classes of write-triggered coupling faults in RAMs. *Journal of Electronic Testing: Theory and Applications*, 3:251–264, 1992.

[4] A. J. v. d. Goor. *Testing Semiconductor Memories*. John Wiley & Sons, 1991.

[5] A. J. v. d. Goor and B. Smit. The automatic generation of march tests. In *Records of the IEEE Workshop on Memory Technology, Design and Testing*, pages 86–91, San Jose, CA, Aug. 1994. IEEE Computer Society Press.

[6] A. J. v. d. Goor and B. Smit. Automating the verification of memory tests. In *IEEE VLSI Test Symposium*, pages 312–318, Cherry Hill, NJ, Apr. 1994. IEEE Computer Society Press.

[7] S. Kornachuk, L. McNaughton, R. Gibbins, and B. Nadeau-Dostie. A high speed embedded cache design with non-intrusive BIST. In *Records of the IEEE Workshop on Memory Technology, Design and Testing*, pages 40–45. IEEE, Aug. 1994.

[8] A. Meixner and J. Banik. Weak write test mode: An SRAM cell stability design for test technique. In *Proc. International Test Conference*, pages 309–318. IEEE Computer Society Press, 1996.

[9] P. R. Sidorowicz and J. A. Brzozowski. An approach to modeling and testing memories and its application to CAMs. In *IEEE VLSI Test Symposium*, pages 411–416. IEEE Computer Society Press, Apr. 1998.

A Analysis of CAM cell Faults

Here, we discuss operations observably affected by the cell stuck-at faults and transistor stuck-(on/open) faults. The behaviors of operations not listed here may differ slightly from their fault-free counterparts in the event-sequence model; however in the FSM model these differences cannot be observed. We also assume that any output from state I is unreliable. Such operations are also omitted for clarity, unless they affect the state of the cell.

***s-sa-0* fault.** Table 5 describes the faulty behavior of a CAM cell in the presence of an *s-sa-0* fault. Deviations from the fault-free behavior are indicated by bold type. Since node s is permanently grounded due to some defect, the cell can only only be in state, 0. As the voltages on $s/\bar{s}$ are fixed, the 'read' and 'compare' operations provide determinate and correct output. The only operation that fails is w_1. Identical behavior is exhibited by the $\bar{s}$*-sa-1* fault.

Table 5. CAM cell with the *s-sa-0* fault.

Write operations ($s = 0$)	
Description	$\overset{w_1}{w\ b\bar{b}\ c\bar{c}\ m{\cdot}s}$
initial state	0 11 00 1·0
set $b/\bar{b}$	0 10 00 1·0
raise w	1 10 00 1·0
new state	1 10 00 1·**0**
lower w	0 10 00 1·**0**
raise $b/\bar{b}$	0 11 00 1·**0**

***s-sa-1* fault.** Analogously to the previous fault, the cell has only one state, 1, and the only operation that fails is w_0. Since the voltages on $s/\bar{s}$ are also fixed, the 'read' and 'compare' operations provide determinate and correct output. Table 6 describes the faulty behavior. Identical behavior is exhibited by the $\bar{s}$*-sa-0* fault.

Table 6. CAM cell with the *s-sa-1* fault.

Write operations ($s = 1$)	
Description	$\overset{w_0}{w\ b\bar{b}\ c\bar{c}\ m{\cdot}s}$
initial state	0 11 00 1·1
set $b/\bar{b}$	0 01 00 1·1
raise w	1 01 00 1·1
new state	1 01 00 1·1
lower w	0 01 00 1·1
raise $b/\bar{b}$	0 11 00 1·1

***T_1-on* fault.** Although similar to the *s-sa-0* fault, the cell may temporarily remain in state I, and switch to state 0 after some time. In state I both T_1 and T_3 conduct; hence this fault is also detectable by I_{DD} tests. Table 7 describes the faulty behaviors. During w_1, when $s = 0$, T_1 stays on after w has been asserted and $\bar{s}$ driven to 0. Since T_3 is on and b is high at the same time,

Table 7. CAM cell with the T_1-*on* fault.

Write operations			
Description	$\overset{w_1\ (s=0)}{w\ b\bar{b}\ c\bar{c}\ m{\cdot}s}$	$\overset{w_1\ (s=1)}{w\ b\bar{b}\ c\bar{c}\ m{\cdot}s}$	$\overset{w_\times\ (s=1)}{w\ b\bar{b}\ c\bar{c}\ m{\cdot}s}$
initial state	0 11 00 1·0	0 11 00 1·**I**	0 11 00 1·**I**
set $b/\bar{b}$	0 10 00 1·0	0 10 00 1·**I**	0 11 00 1·**I**
raise w	1 10 00 1·0	1 10 00 1·**I**	1 11 00 1·**I**
new state	1 10 00 1·**I**	1 10 00 1·**I**	1 11 00 1·**0**
lower w	0 10 00 1·**I**	0 10 00 1·**I**	0 11 00 1·**0**
raise $b/\bar{b}$	0 11 00 1·**I**|**0**	0 11 00 1·**I**|**0**	0 11 00 1·**0**

the voltage at s is intermediate. Once w is de-asserted the voltage at node s eventually becomes 0 again, as T_1 dominates T_3. The $w_\times$ accelerates this transition, by temporarily disabling T_3 and T_4, therefore allowing the faulty T_1 to discharge s. Analogous behavior is exhibited by the T_2-*on* fault.

T_1-*open* fault. In the presence of this fault, the cell cannot maintain state 0 and will flip to state 1. It may, however, remain in state I for a long time; hence this fault is a *data retention* fault. During w_0, s is pulled down solely by the bit-line driver. After w has been de-asserted, s floats; thus the T_2-T_4 inverter acts like a voltage divider. In effect, both $s/\bar{s}$ are in state I. A $w_\times$ forces both nodes to 1, which activates only T_2 (T_1 is open); hence $\bar{s}$ becomes 0 as soon as w is de-asserted. Table 8 describes the faulty behaviors. Analogous behavior is exhibited by the T_2-*open* fault.

Table 8. CAM cell with the T_1-*open* fault.

	Write operations		
Description	w_0 $(s=0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	$w_\times$ $(s=0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	w_0 $(s=1)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·**I**	0 11 00 1·**I**	0 11 00 1·1
set $b/\bar{b}$	0 01 00 1·**I**	0 11 00 1·**I**	0 01 00 1·1
raise w	1 01 00 1·**I**	1 11 00 1·**I**	1 01 00 1·1
new state	1 01 00 1·**I**	1 11 00 1·1	1 01 00 1·**I**
lower w	0 01 00 1·**I**	0 11 00 1·1	0 01 00 1·**I**
raise $b/\bar{b}$	0 11 00 1·**I**|1	0 11 00 1·1	0 11 00 1·**I**|1

T_3-*on* fault. In this case the cell exhibits anomalous behavior in what should be state 0, i.e., when T_3 does not normally conduct. Since T_3 is always on, the resulting state is I, which is detectable by I_{DDQ} tests. Because T_3 is usually weaker than T_1, the voltage on node s may be close to 0 immediately following a w_0. We assume, that the resistance of the faulty transistor is too large for this fault to be modeled by the *s-sa-1* fault, and the voltage on s will rise and remain at I. Since r in state I is unreliable, this fault is also a data retention fault. Table 9 describes the faulty behaviors of a CAM cell in the presence of a T_3-*on* fault. During r, when $s = I$, the differential voltages on the $b/\bar{b}$ lines will be smaller in magnitude; hence, depending on the sensitivity of the sense circuitry, r may yield an unpredictable response. Analogous behavior is exhibited by the T_4-*on* fault.

Table 9. CAM cell with the T_3-*on* fault.

	Write operations	
Description	w_0 $(s=0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	w_0 $(s=1)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·**I**	0 11 00 1·1
set $b/\bar{b}$	0 01 00 1·**I**	0 01 00 1·1
raise w	1 01 00 1·**I**	1 01 00 1·1
new state	1 01 00 1·0|**I**	1 01 00 1·0|**I**
lower w	0 01 00 1·0|**I**	0 01 00 1·0|**I**
raise $b/\bar{b}$	0 11 00 1·**I**	0 11 00 1·**I**

T_3-*open* fault. In the presence of this fault, the cell cannot maintain state 1 and will change to state 0. Since it may remain in state I for a long time, this is also a data retention fault. During w_1, s is pulled up by the bit-line driver. Since T_3 is open, T_1 acts like a DRAM cell; the charge stored at node s is dissipated due to leakage currents. Once the voltage on s drops below a certain threshold T_4 begins to conduct, raising the potential on $\bar{s}$. In turn, T_1 begins to conduct, and grounds s. A $w_\times$ forces both nodes to 1. After w is de-asserted, s will discharge more readily than $\bar{s}$ due to the fault; hence $s = 0$. Table 10 describes the faulty behaviors. Analogous behavior is exhibited by the T_4-*open* fault.

Table 10. CAM cell with the T_3-*open* fault.

	Write operations		
Description	w_1 $(s=0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	w_1 $(s=1)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	$w_\times$ $(s=1)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·0	0 11 00 1·1	0 11 00 1·**I**
set $b/\bar{b}$	0 10 00 1·0	0 10 00 1·1	0 11 00 1·**I**
raise w	1 10 00 1·0	1 10 00 1·1	1 11 00 1·**I**
new state	1 10 00 1·1	1 10 00 1·1	1 11 00 1·**I**
lower w	0 10 00 1·1	0 10 00 1·1	0 11 00 1·**0**
raise $b/\bar{b}$	0 11 00 1·**I**|**0**	0 11 00 1·**I**|**0**	0 11 00 1·**0**

T_5-*on* fault. This fault is somewhat similar to a cell stuck-at-0 fault. In a quiescent state b is driven high. Since T_5 is on, b is always connected to s; hence the cell always reverts to state 1, as indicated in Table 11. However, the cell can temporarily find itself in state 0, which may affect any concurrent 'compare' operations. This fault may also affect other cells connected to the same $b/\bar{b}$ lines but different w lines. For example, if the faulty cell is in state 1 and in the good cell is in state 0, an r of the good cell may be compromised. As T_5 is on, b will be driven by T_3 of the faulty cell, instead of just being floated-high; hence impeding the discharge of b. However, since T_3 of the faulty cell is weaker than T_1 of the good cell, this interference should be negligible. Analogous behavior is exhibited by the T_6-*on* fault.

Table 11. CAM cell with the T_5-*on* fault.

	Write operations $(s=1)$
Description	w_0 $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·1
set $b/\bar{b}$	0 01 00 1·1
raise w	1 01 00 1·1
new state	1 01 00 1·0
lower w	0 01 00 1·0
raise $b/\bar{b}$	0 11 00 1·1

T_5-***open*** **fault.** As a result of this fault, s is permanently isolated from b. An r, when $s = 0$, will yield unreliable results, as b cannot discharge through s. When

Table 12. CAM cell with the T_5-*open* fault.

Read operations		
Description	$(s = 0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	
initial state	0 11 00 1·0	
float $b/\bar{b}$	0 $\tilde{1}\tilde{1}$ 00 1·0	
raise w	1 $\tilde{1}\tilde{1}$ 00 1·0	
b or $\bar{b}$ discharges	1 $\tilde{\mathbf{1}}\tilde{1}$ 00 1·0	
read $b/\bar{b}$ & lower w	0 $\tilde{\mathbf{1}}\tilde{1}$ 00 1·0	
raise $b/\bar{b}$	0 11 00 1·0	

Write operations		
Description	$w_1\ (s = 0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	$w_\times\ (s = 1)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·0	0 11 00 1·1
set $b/\bar{b}$	0 10 00 1·0	0 11 00 1·1
raise w	1 10 00 1·0	1 11 00 1·1
new state	1 10 00 1·**1**	1 11 00 1·**0**
lower w	0 10 00 1·**1**	0 11 00 1·**0**
raise $b/\bar{b}$	0 11 00 1·**0**	0 11 00 1·**0**

$s = 0$, w_1 also fails since s is not driven high by b. Despite the conducting T_3, the change of state does not occur. However, $w_\times$, when $s = 1$ will cause the state to change, since a conducting T_1 will drive s to 0. These events are described in Table 12.

Analogous behavior is exhibited by the T_6-*open* fault.

T_7-***on*** **fault.** This fault affects c_1 when $s = 1$, i.e., when T_7 would normally be open. Table 13 describes this behavior.

As both T_7 and T_8 are on, they act as a voltage divider between $c/\bar{c}$. When the $c/\bar{c}$ are driven to $1/0$, the gate of T_9 is at potential I, roughly equal to $V_{dd}/2$. Since the threshold voltage V_T of T_9 is about $V_{dd}/6$, we assume that T_9 is on; hence, m will discharge.

Table 13. CAM cell with the T_7-*on* fault.

Compare operations $(s = 1)$	
Description	c_1 $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·1
float m	0 11 00 $\tilde{1}$·1
set $c/\bar{c}$	0 11 10 $\tilde{1}$·1
m discharges	0 11 10 **0**·1
read m & ground $c/\bar{c}$	0 11 00 **0**·1
raise m	0 11 00 1·1

Analogous behavior is exhibited by the T_8-*on* fault.

T_7-***open*** **fault.** This fault affects c_1 in state 0, when T_7 would normally be on. Table 14 describes the faulty behavior. Since T_7 is open, T_9 cannot be switched on when $c = 1$, hence m is not discharged.

Analogous behavior is exhibited by the T_8-*open* fault.

Table 14. CAM cell with the T_7-*open* fault.

Compare operations $(s = 0)$	
Description	c_1 $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·0
float m	0 11 00 $\tilde{1}$·0
set $c/\bar{c}$	0 11 10 $\tilde{1}$·0
m discharges	0 11 10 $\tilde{\mathbf{1}}$·0
read m & ground $c/\bar{c}$	0 11 00 $\tilde{\mathbf{1}}$·0
raise m	0 11 00 1·0

T_9-***on*** **fault.** This fault affects 'compare' operations as well. Since T_9 is always on, m is always connected

Table 15. CAM cell with the T_9-*on* fault.

Compare operations $(s = 0)$		
Description	c_0 $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	$c_\times$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 **0**·0	0 11 00 **0**·0
float m	0 11 00 **0**·0	0 11 00 **0**·0
set $c/\bar{c}$	0 11 01 **0**·0	0 11 00 **0**·0
m discharges	0 11 01 **0**·0	0 11 00 **0**·0
read m & ground $c/\bar{c}$	0 11 00 **0**·0	0 11 00 **0**·0
raise m	0 11 00 **0**·0	0 11 00 **0**·0
Compare operations $(s = 1)$		
Description	c_1 $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	$c_\times$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 **0**·1	0 11 00 **0**·1
float m	0 11 00 **0**·1	0 11 00 **0**·1
set $c/\bar{c}$	0 11 10 **0**·1	0 11 00 **0**·1
m discharges	0 11 10 **0**·1	0 11 00 **0**·1
read m & ground $c/\bar{c}$	0 11 00 **0**·1	0 11 00 **0**·1
raise m	0 11 00 **0**·1	0 11 00 **0**·1

to ground which implies an unconditional mismatch as shown in Table 15. It should be noted that since m is driven high when the cell is in a quiescent state, this fault would cause an increased I_{DDQ}.

T_9-***open*** **fault.** Analogously, this fault affects 'compare' operations, as shown in Table 16.

Table 16. CAM cell with the T_9-*open* fault.

Compare operations		
Description	$c_1\ (s = 0)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$	$c_0\ (s = 1)$ $w\ b\bar{b}\ c\bar{c}\ m{\cdot}s$
initial state	0 11 00 1·0	0 11 00 1·1
float m	0 11 00 1·0	0 11 00 1·1
set $c/\bar{c}$	0 11 10 1·0	0 11 01 1·1
m discharges	0 11 10 1·0	0 11 01 1·1
read m & ground $c/\bar{c}$	0 11 00 1·0	0 11 00 1·1
raise m	0 11 00 1·0	0 11 00 1·1

Since T_9 is open, the match line m is never discharged, producing an unconditional match.