Testing Ternary Content Addressable Memories With Comparison Faults Using March-Like Tests

Jin-Fu Li, *Member, IEEE*

***Abstract*—Ternary content addressable memory (TCAM) plays an important role in various applications for its fast lookup operation. This paper proposes several comparison fault models (i.e., the faults cause Compare operation fail) of TCAMs based on electrical defects, such as shorts between two circuit nodes and transistor stuck-open and stuck-on faults. Two March-like tests for detecting comparison faults are also proposed. The first March-like test requires $4N$ Write operations, $3N$ Erase operations, and $4N + 2B$ Compare operations to cover 100% of targeted comparison faults for an $N \times B$-bit TCAM with Hit output only. The second March-like test requires $2N$ Write operations, $2N$ Erase operations, and $4N + 2B$ Compare operations to cover 100% of targeted comparison faults for an $N \times B$-bit TCAM with Hit and Priority Address Encoder outputs. Compared with the previous work, the proposed tests have lower time complexity for typical TCAMs; they can be used to test TCAMs with different comparator structures; and their time complexities are independent of the number of stuck-on faults. Also, they can cover delay faults in comparison circuits.**

***Index Terms*—Comparison faults, content addressable memories (CAMs), delay faults, march tests, memory testing, ternary content addressable memories.**

I. Introduction

TERNARY content addressable memories (TCAMs) play an important role in networking applications such as packet forwarding or address classification [1]–[3]. High-performance network routers need large-capacity and fast TCAMs for achieving the fast search speed in large routing tables. Moreover, emerging applications require the longest match searches, such as flow analysis and classless interdomain routing. TCAMs can provide a high-quality solution for these applications. Therefore, high-density large-capacity TCAMs are more and more popular and important due to these applications. Thus, testing these large embedded or stand-alone TCAMs is becoming an important issue. For performing high-speed parallel comparison, however, a TCAM cell consists of a storage and a comparator. The special TCAM cell structure causes that the testing of TCAMs is very difficult.

Several test and diagnosis algorithms for binary CAMs (BCAMs) have been proposed in [4]–[13]. In [4], a functional fault model for BCAMs was derived by investigating the functional failures in the storage cell and comparison logic. In [5], an approach for modeling and testing memories and its application to BCAMs was introduced. March-like test algorithms reported in [6] are used to detect the comparison faults of BCAMs. Assume that the comparison result is observed only by the Hit output, the test algorithms can detect 100% comparison faults. In [7], test algorithms for BCAMs, which can perform Read and Compare operations concurrently, were proposed. The basic BCAM cells with individual bit lines and comparison lines are assumed. Also, the comparison result observed by the priority address encoder (PAE) is assumed. In [8] and [9], test algorithms for covering comparison faults and RAM faults were reported. Fault-location algorithms also were presented to identify faulty cells in a faulty word or a faulty column. In [10], a test methodology for detecting delay faults of BCAMs was proposed. In [11], a three-phase diagnosis procedure for BCAMs was presented. The user can distinguish different types of comparison faults and RAM faults and locate the faulty cells with the diagnosis procedure. Recently, testing the peripheral circuitry of the BCAMs is also concerned. A test algorithm for detecting the stuck-at faults in the PAE of BCAMs was presented in [13].

If dynamic CAMs are considered, testing neighborhood pattern-sensitive faults (NPSFs) also is an important issue. Several works introducing the test algorithms for NPSFs in BCAMs were reported in [14] and [15]. In [14], static pattern-sensitive fault and dynamic pattern-sensitive fault are used to develop the test algorithm. Due to the additional constraint on the fault model and design-for-testability (DFT) circuitry in the BCAM under test, the test complexity of the proposed test algorithm is low. In [15], a built-in self-test (BIST) scheme executing a parallel test algorithm for BCAMs was proposed. All cells in a BCAM can be separated to one based-cell group and eight deleted neighborhood groups for detecting NPSFs. By inserting DFT circuitry, the cells in the same group can be accessed simultaneously. Therefore, the test complexity for the NPSF is not related to the address space. Only 19 464 operations are required to cover the NPSFs regardless the size of the BCAM [15].

Although each TCAM cell can logically be regarded as a combination of two BCAM cells [16], the TCAM testing is more difficult than the BCAM testing since a TCAM cell is more complex than a BCAM cell. For example, the TCAM cell can store and compare an additional masking state. To support the masking state, a TCAM cell consists of a two-bit storage element and a comparing logic. On the other hand, since two-bit binary data can encode four states and only three states are used for a TCAM cell, one state is an illegal state for the

Manuscript received January 26, 2006; revised May 15, 2006 and July 22, 2006. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC 94-2215-E-008-021. This paper was recommended by Associate Editor K. Chakrabarty.

The author is with the Advanced Reliable Systems (ARES) Laboratory, Department of Electrical Engineering, National Central University, Jhongli 320, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCAD.2006.884415

TCAM cell. This issue also should be considered for TCAM testing. Recently, several research works on the testing of TCAMs were presented in [17]–[22]. In [17], a BIST scheme for TCAMs was presented. The BIST tests one row of the TCAM at a time. This can reduce the testing power. But this causes that the test complexity of the test algorithm is $O(NB)$ for an $N \times B$-bit TCAM. In [18], a search path test algorithm (SPTA) was developed to test the stuck-on and stuck-open faults in the search path of the dynamic TCAMs. However, the transistor-level faults are targeted, such that the algorithm only can be used to test the TCAM with a specific comparator structure. Also, time complexity of the test algorithm is related to the number of transistor stuck-on faults. Thus, the testing procedure is related to the number of stuck-on faults in the TCAM under test. This causes the BIST circuitry for performing the test algorithm to be more complicated. In [19], the test algorithm is developed based on a TCAM with parallel access capabilities. Therefore, the test algorithm requires $(30B + 3)$ Write operations and $36B$ Compare operations to test an $N \times B$-bit TCAM. However, some design issues must be resolved to allow the parallel access. In [20], the author designed a test algorithm for TCAMs based on the comparison faults of BCAMs. This can improve the resolution of fault location if fault diagnosis is considered. However, the test algorithm cannot cover the defects between adjacent bit lines of the two storages in a TCAM cell. Later, fault modeling and testing for TCAMs with Hit output only were introduced in [21]. In [22], a test methodology for active NPSFs in ternary CAMs was proposed. The methodology combines two-group and March-like tests to cover 100% active NPSFs.

This paper presents functional comparison fault models of TCAMs. These comparison faults are defined by injecting electrical defects, such as shorts between two circuit nodes, transistor stuck-on, and transistor stuck-open faults. Two March-like functional tests (T_{tcam1} and T_{tcam2}) are also proposed. T_{tcam1} can cover 100% of targeted comparison faults with $4N$ Write operations, $4N + 2B$ Compare operations, and $3N$ Erase operations for an $N \times B$-bit TCAM with Hit output only. T_{tcam2} needs $2N$ Write operations, $4N + 2B$ Compare operations, and $2N$ Erase operations to detect 100% of targeted comparison faults for an $N \times B$-bit TCAM with Hit and PAE outputs. The two tests also can detect the delay faults in comparison circuits. Compared with the previous works [17]–[19], the works [17], [19] need DFT insertion. Also, the test scheme reported in [17] is very time consuming, and many design issues must be resolved to support the test scheme reported in [19]. The test reported in [18] and ours do not need DFT insertion and time complexity of these two test schemes is only proportional to N and B. However, our tests can be used to test different TCAMs with various cell structures. Also, time complexity of the proposed tests is lower for typical TCAMs and is independent of the number of stuck-on faults. Moreover, the proposed tests can cover delay faults in comparison circuits.

The rest of this paper is organized as follows. The next section reviews the architecture and functions of TCAMs. Section III defines functional comparison faults of TCAMs. Section IV introduces two proposed March-like tests for covering the comparison faults. Section V discusses why the proposed tests also can cover the delay faults in comparison circuits. Section VI describes the time complexity of the proposed tests and comparison results. Finally, Section VII concludes this paper.

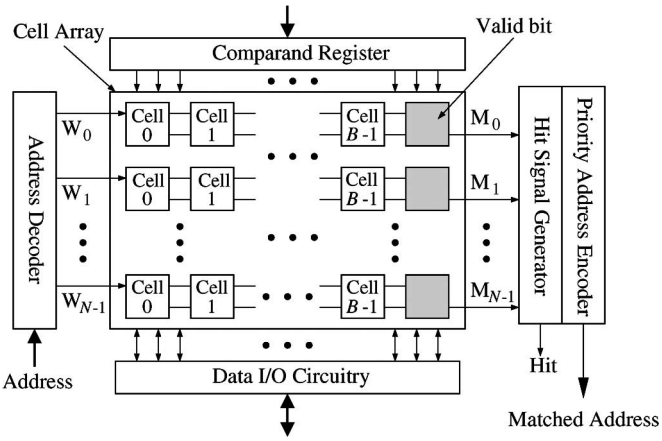

Fig. 1. Typical $N \times B$-bit TCAM organization.

II. Preliminary

Fig. 1 depicts a typical $N \times B$-bit TCAM organization. The Address Decoder and Data I/O circuitry are similar to those in a RAM. The cell array consists of N words. Each word has B cells and a Valid bit which indicates whether the corresponding match signal is valid or invalid. A TCAM usually has the following basic operations: Write, Compare (Search), Read, and Erase. The Read and Write operations are the same as those of a RAM. The Write operation also sets the Valid bit of the corresponding word (so that it is in the valid state). The Erase operation resets the corresponding Valid bit of a specified word. The Compare operation compares a comparand (compared data) with all words in the TCAM simultaneously. When the TCAM executes Compare operation, the input pattern (comparand) is prefetched to the Comparand Register and then is parallel compared with the symbols stored in all the words. The Hit Signal Generator evaluates the valid match signals, and generates a hit output $(\mathrm{Hit} = 1)$ if there is at least one valid match. The PAE exports the highest priority matched address (either the lowest matched address or the highest matched address). Note that the PAE and Hit Signal Generator may not coexist. For some specific applications, a TCAM does not be equipped with a PAE and only a Hit Signal Generator is realized to shorten critical-path delay for performing a Compare operation, e.g., a TCAM without a PAE is reported in [3].

Fig. 2 shows the structure of a typical static TCAM cell [16], [23]. A static TCAM cell consists of two static RAM (SRAM) cells and one comparator. Fig. 2(a) is a two-port cell structure and Fig. 2(b) is a single-port cell structure. The difference between the two-port cell and single-port cell is that the single-port cell shares the comparison and bit lines. Therefore, the two-port TCAM cell can execute the Read/Write and Compare operation concurrently, but the single-port cell cannot. We describe the TCAM cell function based on the two-port cell structure without loss of generality. The SRAM cells of a TCAM cell are used to store two binary values for encoding

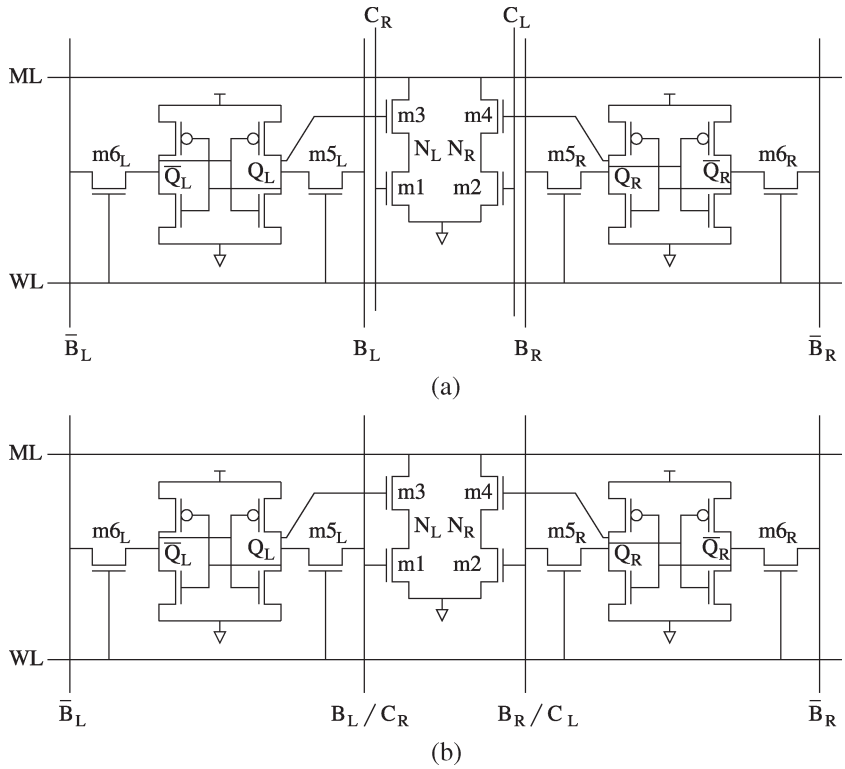

Fig. 2. Typical TCAM cell structures: (a) a two-port cell structure and (b) a single-port cell structure.

ternary data. As the figure shows, the data of the left and right SRAM cells are $Q_{\rm L}$ and $Q_{\rm R}$, respectively. Assume that the encoding scheme is as follows: **0** $(Q_{\rm L}=0,\ Q_{\rm R}=1)$, **1** $(Q_{\rm L}=1,\ Q_{\rm R}=0)$, and X $(Q_{\rm L}=0,\ Q_{\rm R}=0)$, where **0**, **1**, and X (don't care) denote the ternary data. The comparator is usually composed of four NMOS transistors, i.e., m1, m2, m3, and m4 [16], [23], [24]. Note that the encoding scheme can be changed with the cell structure. When the cell executes a Compare operation, the match line (ML) is first precharged to $V_{\rm dd}$. Subsequently, the data $(Q_{\rm L}, Q_{\rm R})$ is compared with the comparand $(C_{\rm L},\ C_{\rm R})$. If the data is equivalent to the comparand, the two pull-down paths are turned-off and the logic value of the ML is remained at logic 1 (Match). On the contrary, one of the two pull-down paths is turned-on and the logic value of the ML is discharged to logic 0 (Mismatch) when the data is different from the comparand.

If the cell stores X data, the m3 and m4 are turned-off, i.e., the cell is masked and not anticipated the Compare operation, which is called as local masking function. On the other hand, if one bit of the comparand is X, the corresponding bits of all words in the TCAM are masked since the m1 and m2 of these bits are turned-off, which is called as global masking function. Note that $(Q_{\rm L},\ Q_{\rm R})=(1,\ 1)$ or $(C_{\rm L},\ C_{\rm R})=(1,\ 1)$ is illegal data for this cell structure. We use the notation U to denote the undefined (illegal) data (1, 1). Table I summarizes the truth table of a TCAM cell when it performs a Compare operation with $\mathbf{0}/\mathbf{1}$ data and comparand.

As described above, a TCAM cell consists of a storage (two SRAM cells) and a comparator for executing high-speed parallel comparison. However, their special and complicated functions cause that the TCAM testing is more difficult than the RAM testing, since the defects in the comparator also must be detected. That is, in addition to the faults in storage (RAM faults), the faults in the comparator (comparison faults) must be considered. Furthermore, the fault effect of RAM faults can be observed by the results of Read or Compare operations, but the fault effect of comparison faults only can be observed by the results of Compare operations.

TABLE I
TRUTH TABLE OF A TCAM CELL EXECUTING A COMPARE OPERATION

$Q_{\rm L}$ $Q_{\rm R}$	$C_{\rm R}$ $C_{\rm L}$	ML
0 1 (**0**)	1 0 (**0**)	Match
0 1 (**0**)	0 1 (**1**)	Mismatch
1 0 (**1**)	1 0 (**0**)	Mismatch
1 0 (**1**)	0 1 (**1**)	Match

Since a TCAM cell has two SRAM cells, the testing of RAM faults is required. The testing of RAMs has been considered a mature area of research and many existing algorithms can provide adequate coverage for the RAM faults, e.g., [25], [26]. Therefore, popular March tests can be used to test the RAM faults of TCAM. However, the tests for comparison faults and the tests for RAM faults can be combined to reduce the test cost. For example, several tests for comparison faults and RAM faults in BCAMs have been reported in [6] and [7]. The sequel of this paper only focuses on the testing of comparison faults of TCAMs.

III. CELL-LEVEL FAULT MODELING

A. Definition of Electrical Defects

The behaviors of the defined comparison faults are verified with a circuit-level simulator (Hspice) by injecting the following electrical defects. 1) Shorts—represent unwanted

TABLE II
POSSIBLE DEFECTS LEAD TO AN SMF OR SMMF

Defect	Nodes/Transistor	Fault Type
Short	ML and V_{dd}	SMF
Bridge	C_L and C_R	SMF
Bridge	Q_L and Q_R	SMF
Bridge	$\bar{Q}_L$ and $\bar{Q}_R$	SMF
Short	ML and Gnd	SMMF
Short	B_L and V_{dd}	SMMF*
Short	B_R and V_{dd}	SMMF*

impedance's between a signal line and $V_{\rm dd}$ or Gnd [27]. Here, 1-Ω short resistance is used for simulating the shorts. 2) Bridges—represent unwanted impedance's between two signal lines [27]. Similarly, 1-Ω bridge resistance is added for simulating bridges. A bridge between two lines with opposite voltage levels may bring about an intermediate voltage level, a high-voltage level, or a low-voltage level on both nodes. The actual behavior of a bridge depends on many factors, including operating voltage, transistor sizing, fabrication process, resistance of a bridge caused by a particle, etc. [6]. We will consider as many faulty situations under a bridge as possible. In the following discussion, we consider the voltage level on both lines (nodes) with opposite voltage levels as a voltage level that will result in a fault. 3) Transistor stuck-on—the transistor is always turned-on regardless of the voltage level of its controlling gate. Here, the transistor stuck-on is simulated by connecting its controlling gate of NMOS (PMOS) to $V_{\rm dd}$ (Gnd). 4) Transistor stuck-open—the transistor is always turned-off regardless of the voltage level of its controlling gate. Similarly, the transistor stuck-open is simulated by connecting its controlling gate of NMOS (PMOS) to Gnd $(V_{\rm dd})$.

B. Electrical Fault Models

In this section, we define comparison fault models for TCAMs based on the typical TCAM cell structures shown in Fig. 2. We first define the notation used in the sequel of this paper. Let x (y) be a ternary data (comparand) written into a TCAM cell and $\bar{x}(\bar{y})$ be any one of two ternary data (comparand) which excludes the ternary data (comparand) x (y). For example, if $x = \mathbf{0}$, then $\bar{x} = \mathbf{1}$ or X. Also, if a fault is manifested by a defect in a single-port TCAM cell only, a star symbol $(^*)$ is added at the superscript of the fault type.

A stuck-match fault [6] (SMF) causes a TCAM cell to always match its corresponding comparand regardless of the state of the TCAM cell. On the contrary, if a TCAM cell has a stuck-mismatch fault [6] (SMMF), the TCAM cell always mismatches the comparand regardless of the state of the TCAM cell. Table II lists several possible defects that lead to SMFs and SMMFs. For example, if a short defect exists between the ML and $V_{\rm dd}$ of a TCAM cell, the ML always signals a match when a Compare operation is executed. Thus, the TCAM cell has an SMF. If a bridge defect exists between two lines, $C_{\rm L}$ and $C_{\rm R}$, and the bridge causes low-voltage level on both lines, then the TCAM cell has an SMF since the transistors m1 and m2 are always turned-off. If a short defect occurs at the $B_{\rm L}$ and $V_{\rm dd}$ of a single-port TCAM cell, then only the data **1** or U (11) can be written into the TCAM cell regardless of the written data. However, the comparand received by the TCAM cell only is **0** or U. Therefore, the TCAM cell behaves as an SMMF, since at least one pull-down path is turned-on regardless of the data and comparand.

TABLE III
POSSIBLE DEFECTS LEAD TO A PMCF

Defect	Nodes/Transistor	Fault Type
Bridge	C_L and ML	PMC**1**F
Bridge	B_R and ML	PMC**1**F*
Bridge	C_R and ML	PMC**0**F
Bridge	B_L and ML	PMC**0**F*

TABLE IV
POSSIBLE DEFECTS LEAD TO A CMMF

Defect	Nodes/Transistor	Fault Type
Bridge	C_R and N_L	CMM**1**F
Bridge	B_L and N_L	CMM**1**F*
Bridge	C_L and N_R	CMM**0**F
Bridge	B_R and N_R	CMM**0**F*

A TCAM cell with a partial-match comparand fault (PMCF) will be always match when the comparand is y. But, the TCAM cell will be always mismatch if the comparand is $\bar{y}$. The PMCF can be further divided into the partial-match comparand-**0** fault (PMC**0**F) and partial-match comparand-**1** fault (PMC**1**F) for $y = \mathbf{0}$ and $y = \mathbf{1}$, respectively. Table III lists several possible defects leading to a PMCF. For example, if a bridge exists between $C_{\rm L}$ and ML, the result of the ML is affected by the $C_{\rm L}$. Therefore, if the comparand is **1**, the comparison result is match. If the comparand is **0** or X, the comparison result is mismatch. Thus, the TCAM cell has a PMC**1**F.

A TCAM cell with a conditional-mismatch fault (CMMF) will function incorrectly if the TCAM cell stores X. However, the TCAM cell functions correctly if the state of the TCAM cell is $\bar{x}$. The CMMF can be divided into CMM**0**F and CMM**1**F for $x = \mathbf{0}$ and $x = \mathbf{1}$, respectively. Several electrical defects leading to a CMMF are listed in Table IV. For example, if a $C_{\rm R} - N_{\rm L}$ bridge exists in a TCAM cell and the TCAM cell stores data **1**, the logic value of the ML is determined by the $C_{\rm R}$, since the transistor m3 is turned-on and $C_{\rm R}$ is bridged to the node $N_{\rm L}$. Then, the TCAM cell signals incorrect results for all subsequently Compare operations. On the contrary, if the data **0** or X is in the TCAM cell, the TCAM cell can correctly execute all subsequently Compare operations. The reason is that the m3 is turned-off such that the defect cannot affect the ML.

A TCAM cell with a conditional-SMF (CSMF) will always match the comparand for all subsequent Compare operations when the TCAM cell stores data X. But the TCAM cell will function correctly if it stores a logic value $\bar{x}$. The CSMF can further be divided into CSM**1**F and CSM**0**F for $x = \mathbf{1}$ or $x = \mathbf{0}$, respectively. Table V lists several electrical defects leading to a CSMF. For example, if a $Q_{\rm L}$–Gnd short exists in a TCAM cell, the gate of the transistor m3 is always connected to logic 0. Then, the TCAM cell will always match for all subsequently Compare operations when it stores **1**. But, the TCAM cell can correctly execute all subsequently Compare operations

TABLE V
POSSIBLE DEFECTS LEAD TO A CSMF

Defect	Nodes/Transistor	Fault Type
Short	Q_L and Gnd	CSM**1**F
Short	B_L and Gnd	CSM**1**F
Short	C_R and Gnd	CSM**1**F
Short	N_L and V_{dd}	CSM**1**F
Bridge	Q_L and $\bar{Q}_L$	CSM**1**F
Bridge	C_R and B_L	CSM**1**F
Bridge	C_R and B_R	CSM**1**F
Stuck-open	m1 or m3	CSM**1**F
Short	Q_R and Gnd	CSM**0**F
Short	B_R and Gnd	CSM**0**F
Short	C_L and Gnd	CSM**0**F
Short	N_R and V_{dd}	CSM**0**F
Bridge	Q_R and $\bar{Q}_R$	CSM**0**F
Bridge	C_L and B_L	CSM**0**F
Bridge	C_L and B_R	CSM**0**F
Stuck-open	m2 or m4	CSM**0**F

TABLE VI
POSSIBLE DEFECTS LEAD TO A CSMMF

Defect	Nodes/Transistor	Fault Type
Short	C_R and V_{dd}	CSMM**1**F
Bridge	C_R and B_L	CSMM**1**F
Bridge	C_R and B_R	CSMM**1**F
Short	N_L and Gnd	CSMM**1**F
Stuck-on	m1	CSMM**1**F
Short	C_L and V_{dd}	CSMM**0**F
Bridge	C_L and B_L	CSMM**0**F
Bridge	C_L and B_R	CSMM**0**F
Short	N_R and Gnd	CSMM**0**F
Stuck-on	m2	CSMM**0**F

TABLE VII
POSSIBLE DEFECTS LEAD TO A SCSMMF

Defect	Nodes/Transistor	Fault Type
Short	Q_R and V_{dd}	SC**1**SMMF
Short	B_R and V_{dd}	SC**1**SMMF
Bridge	Q_R and C_L	SC**1**SMMF
Bridge	$\bar{Q}_R$ and C_R	SC**1**SMMF
Bridge	ML and N_R	SC**1**SMMF
Bridge	$\bar{Q}_R$ and Q_R	SC**1**SMMF
Stuck-on	m4	SC**1**SMMF
Bridge	Q_R and B_R	SC**1**SMMF*
Bridge	$\bar{Q}_R$ and B_L	SC**1**SMMF*
Stuck-on	m2	SC**1**SMMF*
Short	Q_L and V_{dd}	SC**0**SMMF
Short	B_L and V_{dd}	SC**0**SMMF
Bridge	Q_L and C_R	SC**0**SMMF
Bridge	$\bar{Q}_L$ and C_L	SC**0**SMMF
Bridge	ML and N_L	SC**0**SMMF
Bridge	$\bar{Q}_L$ and Q_L	SC**0**SMMF
Stuck-on	m3	SC**0**SMMF
Bridge	Q_L and B_L	SC**0**SMMF*
Bridge	$\bar{Q}_L$ and B_R	SC**0**SMMF*
Stuck-on	m1	SC**0**SMMF*

TABLE VIII
POSSIBLE DEFECTS LEAD TO A SDCMMF

Defect	Nodes/Transistor	Fault Type
Bridge	N_L and N_R	SDCMMF
Bridge	Q_L and Q_R	SDCMMF
Bridge	C_R and C_L	SDCMMF
Bridge	B_L and B_R	SDCMMF

when it stores **0** or X. Thus, the short causes the TCAM cell with a CSM**1**F.

A TCAM cell with a conditional-SMMF (CSMMF) will be always mismatch all subsequent Compare operations when the state of the TCAM cell is X. However, the TCAM cell will function correctly if it stores $\bar{x}$. The CSMMF has two subtypes: CSMM**0**F and CSMM**1**F for $x = \mathbf{0}$ and $x = \mathbf{1}$, respectively. Table VI lists some electrical defects that cause a TCAM cell with a CSMMF. For example, if a TCAM cell has a $C_{\rm R} - V_{\rm dd}$ short, the gate of transistor m1 is connected to $V_{\rm dd}$. Therefore, if data **1** is written into the TCAM cell, the m3 is also turned-on. Then, the pull-down path (m1 and m3) is always turned-on, which results in mismatch for all subsequently Compare operations. But, if the TCAM cell stores **0** or X, it will function correctly. The reason is that the m3 is turned-off such that the short defect cannot affect all subsequent Compare operations. Thus, this defect results in a CSMM**1**F.

A TCAM cell with specific compared SMMF (SCSMMF) will be always be a mismatch regardless of the TCAM cell state. The SCSMMF can further be divided into SC**0**SMMF and SC**1**SMMF for the comparand **0** and **1**, respectively. Table VII lists several defects that lead to an SCSMMF. Consider a TCAM cell with a $Q_{\rm R} - V_{\rm dd}$ short. If a Compare-**1** operation is executed, the m2 is also turned-on. Therefore, the comparison result is always mismatch regardless of the TCAM cell state. Thus, the TCAM cell has an SC**1**SMMF.

A TCAM cell with a specific data-comparand mismatch fault (SDCMMF) will always be mismatch if the TCAM cell stores X and the comparand is y for $x = y = \mathbf{0}$ or **1**. Table VIII lists several electrical defects that lead to an SDCMMF. Consider an $N_{\rm L} - N_{\rm R}$ bridge in a TCAM cell. If a **0** or **1** is written into the TCAM cell, one of the two pull-down paths will be turned-on when the comparand **0** or **1** is applied, respectively. Then, the comparison result is always mismatch. Thus, the cell has an SDCMMF.

A TCAM cell with a data-unmatchable fault (DUMF) will be unmatchable if the TCAM cell stores X. However, the TCAM cell will function correctly if the state of the TCAM cell is $\bar{x}$. The DUMF can further be divided into D**1**UMF and D**0**UMF for $x = \mathbf{1}$ and $x = \mathbf{0}$, respectively. Table IX lists several possible defects that lead to a DUMF. For example, if a TCAM cell has a $C_{\rm L} - N_{\rm L}$ bridge, the logic value of $N_{\rm L}$ is equal to $C_{\rm L}$. That is, the exact comparand received by the cell is $(C_{\rm L}, C_{\rm L})$. Therefore, if data **1** is written into the cell, the transistor m3 is turned-on. Subsequently, if the comparand X is applied, the ML will be discharged by the $C_{\rm L}$ through the m3. Thus, the Compare-X operation cannot work correctly. On the contrary, if the data **0** or X is written into the TCAM cell, the transistor m3 is turned-off. This means that the short cannot

TABLE IX
POSSIBLE DEFECTS LEAD TO A DUMF

Defect	Nodes/Transistor	Fault Type
Bridge	C_L and N_L	D**1**UMF
Bridge	B_R and N_L	D**1**UMF*
Bridge	C_R and N_R	D**0**UMF
Bridge	B_L and N_R	D**0**UMF*

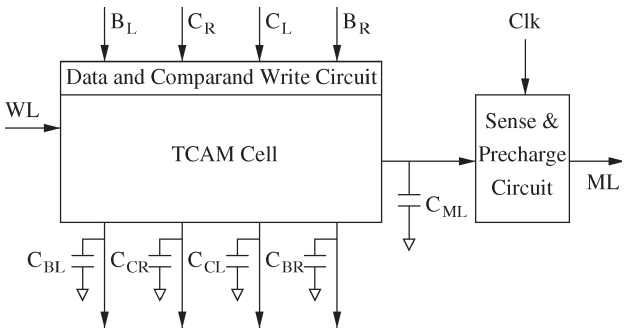

Fig. 3. Hspice simulation model.

affect the Compare-X operation. Therefore, the defect causes a D**1**UMF.

To make the definitions of these comparison faults described above more clearer, the responses of a TCAM cell to all possible Compare-after-Write operations under different comparison faults are summarized in Table X, where $(w\mathrm{X}, cy)$ denotes that a TCAM cell undergoes a Write-X operation and then a Compare-y operation. In the table, "M" or "MM" represents that the comparison result is match or mismatch, respectively.

C. Simulation Results

All the defined comparison faults described above are verified by Hspice simulations. Fig. 3 shows the Hspice model for verifying the defined comparison faults. Only the peripheral circuits for Write and Compare operations are implemented since the defined comparison faults are only related to these two operations. The ML, comparison lines $(C_\mathrm{L}, C_\mathrm{R})$, and bit lines $(B_\mathrm{L}, B_\mathrm{R})$ include the capacitive loads for simulating the other TCAM cells in the same bit and the same word. The precharge and sense circuits are synchronizing by the clock signal (Clk).

Subsequently, Hspice simulation waveforms for several defined fault models are shown. Fig. 4 shows the simulation waveform for the CSM**0**F while a short defect exists between Q_R and Gnd. Note that the word line of the TCAM cell is set to 1 or 0 for executing a Write or a Compare operation, respectively. Also, the output of ML is synchronized by the Clk signals. If the logic value of ML is high (low), the corresponding comparison result is match (mismatch). Q_L and Q_R are the data of the SRAM cells in the TCAM cell. As the waveform shows, a Write-**0** $(w\mathbf{0})$ operation first writes data **0** into the cell and the value of $(Q_\mathrm{L}, Q_\mathrm{R}) = (0, 1)$. Therefore, the data **0** is written correctly. Then, three Compare operations, Compare-**0** $(c\mathbf{0})$, Compare-**1** $(c\mathbf{1})$, and Compare-X $(c\mathrm{X})$, are performed in sequence. The comparison results for these three Compare operations all are match. In a similar way, a $w\mathbf{1}$ is performed and then three Compare operations $(c\mathbf{0}, c\mathbf{1}, c\mathrm{X})$ are executed in sequence. The comparison results are mismatch, match, and match. Finally, a $w\mathrm{X}$ is done and then three Compare operations $(c\mathbf{0}, c\mathbf{1}, c\mathrm{X})$ are performed in sequence. The comparison results all are match. Therefore, we can see that all the possible Compare-after-Write operations are performed and the corresponding comparison results are the same as those of CSM**0**F shown in Table X.

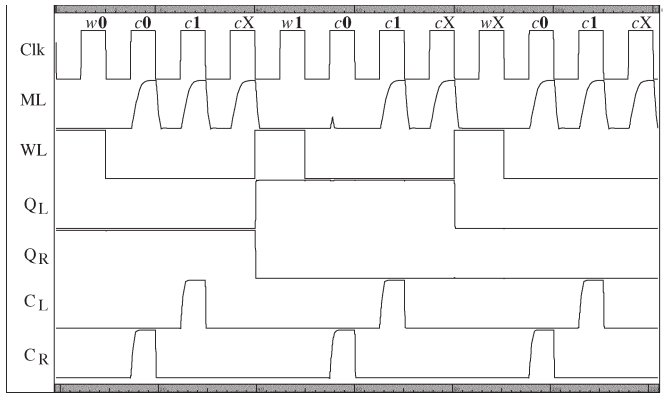

Fig. 4. Hspice simulation waveforms for CSM**0**F.

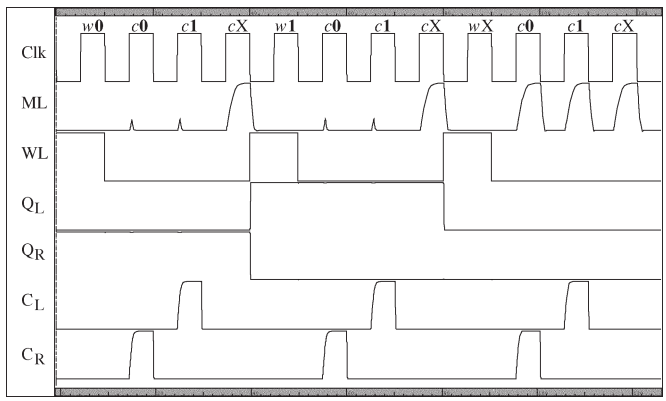

Fig. 5. Hspice simulation waveforms for SDCMMF.

Fig. 5 depicts the simulation waveform for the SDCMMF while a bridge defect exists between N_L and N_R. As the waveform shows, a $w\mathbf{0}$ operation first writes data **0** into the cell. Then, three Compare operations, $c\mathbf{0}$, $c\mathbf{1}$, and $c\mathrm{X}$, are performed in sequence. The comparison results for these three Compare operations are mismatch, mismatch, and match. Subsequently, a $w\mathbf{1}$ is performed and then three Compare operations $(c\mathbf{0}, c\mathbf{1}, c\mathrm{X})$ are executed in sequence. The comparison results also are mismatch, mismatch, and match. Finally, a $w\mathrm{X}$ is executed and then three Compare operations $(c\mathbf{0}, c\mathbf{1}, c\mathrm{X})$ are performed in sequence. The comparison results all are match. As the waveform shows, therefore, all the possible Compare-after-Write operations are simulated and the corresponding comparison results are the same as those of SDCMMF listed in Table X.

Fig. 6 shows the simulation waveform for the SC**0**SMMF caused by a bridge defect which exists between Q_L and C_R. The bridge defect causes that the value of Q_L will be equal to the C_R while the Compare operation is executed. As Fig. 6 depicts, Q_L is changed with the C_R. The $w\mathbf{0}$ operation first writes data **0** into the cell. Then, three Compare operations, $c\mathbf{0}$, $c\mathbf{1}$, and $c\mathrm{X}$, are performed in sequence. While the $c\mathbf{0}$ is executed, the Q_L becomes 1. This causes the comparand to

TABLE X
TCAM CELL RESPONSES TO THE COMPARE-AFTER-WRITE OPERATION FOR DIFFERENT COMPARISON FAULTS

	$w\mathbf{0}, c\mathbf{0}$	$w\mathbf{0}, c\mathbf{1}$	$w\mathbf{0}, c\mathrm{X}$	$w\mathbf{1}, c\mathbf{0}$	$w\mathbf{1}, c\mathbf{1}$	$w\mathbf{1}, c\mathrm{X}$	$w\mathrm{X}, c\mathbf{0}$	$w\mathrm{X}, c\mathbf{1}$	$w\mathrm{X}, c\mathrm{X}$
Fault-free	M	MM	M	MM	M	M	M	M	M
SMF	M	M	M	M	M	M	M	M	M
SMMF	MM	MM	MM	MM	MM	MM	MM	MM	MM
PMC**1**F	MM	M	MM	MM	M	MM	MM	M	MM
PMC**0**F	M	MM	MM	M	MM	MM	M	MM	MM
CMM**1**F	M	MM	M	M	MM	MM	M	M	M
CMM**0**F	MM	M	MM	MM	M	M	M	M	M
CSM**1**F	M	MM	M	M	M	M	M	M	M
CSM**0**F	M	M	M	MM	M	M	M	M	M
CSMM**1**F	M	MM	M	MM	MM	MM	M	M	M
CSMM**0**F	MM	MM	MM	MM	M	M	M	M	M
SC**1**SMMF	M	MM	M	MM	MM	M	M	MM	M
SC**0**SMMF	MM	MM	M	MM	M	M	MM	M	M
SDCMMF	MM	MM	M	MM	MM	M	M	M	M
D**1**UMF	M	MM	M	MM	M	MM	M	M	M
D**0**UMF	M	MM	MM	MM	M	M	M	M	M

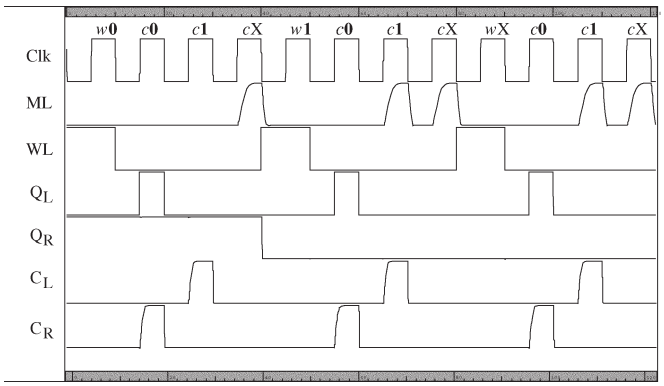

Fig. 6. Hspice simulation waveforms for SC**0**SMMF.

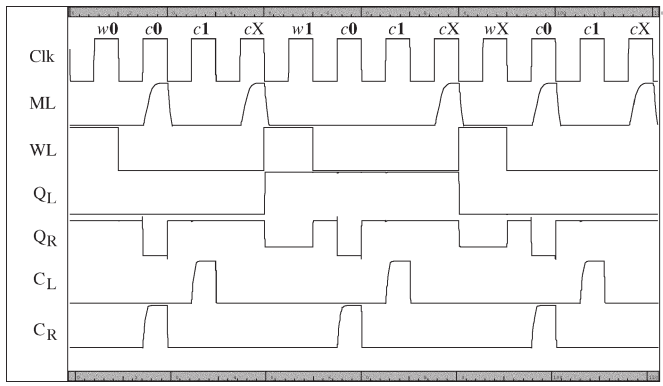

Fig. 7. Hspice simulation waveforms for SC**1**SMMF.

be different from the stored data and results in a mismatch. Subsequently, the $c\mathbf{1}$ and $c\mathrm{X}$ are performed and the C_R is 0 for these two comparands, and the data of Q_R is the same as fault-free data. Therefore, the comparison results for these two Compare operations are correct. In a similar way, the other cases can be analyzed. As the waveform shows, therefore, all the possible Compare-after-Write operations are simulated and the corresponding comparison results are the same as those of SS**0**SMMF listed in Table X.

Fig. 7 depicts the simulation waveform for the SC**1**SMMF caused by a $\overline{Q}_R - C_\mathrm{R}$ bridge defect. The bridge defect causes that the value of $\overline{Q}_R$ is changed with the C_R while the Compare operation is executed. As the figure shows, the value of Q_R is changed with the $\overline{C}_R$ while Compare operations are performed. On the other hand, the value of Q_R is changed with the written data while Write operations are executed. The $w\mathbf{0}$ first writes data **0** into the cell and $(Q_\mathrm{L},\ Q_\mathrm{R}) = (0, 1)$. Then, $c\mathbf{0}$, $c\mathbf{1}$, and $c\mathrm{X}$ operations are applied in sequence. When $c\mathbf{0}$ is performed, the data Q_R is pulled down since C_R is forced to 1. Since $(Q_\mathrm{L},\ Q_\mathrm{R}) = (0,\ 0)$, the comparison result is match regardless of the comparand. While $c\mathbf{1}$ is applied, the Q_R is 1 since the C_R is 0. Therefore, the comparison result is mismatch since the data is the complement of the comparand. While $c\mathrm{X}$ is executed, the comparison result is match since $(C_\mathrm{L},\ C_\mathrm{R}) = (0,\ 0)$. In a similar way, the other cases can be verified. As Fig. 7 shows, the comparison results shown in ML for Compare operations are the same as those of SC**1**SMMF listed in Table X.

IV. TESTS FOR COMPARISON FAULTS

A. Test for TCAMs With Hit Output Only

This section describes a March-like test (T_tcam1) for detecting the defined comparison faults in TCAMs with Hit output only. T_tcam1 consists of the following test elements (TEs):

TE1: $\Updownarrow$ (ERS);
TE2: $\Updownarrow (w\mathbf{0}, cP_\mathbf{0}, cP_\mathrm{X}, \mathrm{ERS})$;
TE3: $\Updownarrow (w\mathbf{1}, cP_\mathbf{1}, cP_\mathrm{X}, \mathrm{ERS})$;
TE4: $\Updownarrow (w\mathbf{1})$;
TE5: $(cP_{\mathrm{X}\ldots\mathrm{X}\mathbf{0}}, cP_{\mathrm{X}\ldots\mathbf{0}\mathrm{X}}, \ldots, cP_{\mathbf{0}\mathrm{X}\ldots\mathrm{X}})$;
TE6: $\Updownarrow (w\mathbf{0})$; and
TE7: $(cP_{\mathrm{X}\ldots\mathrm{X}\mathbf{1}}, cP_{\mathrm{X}\ldots\mathbf{1}\mathrm{X}}, \ldots, cP_{\mathbf{1}\mathrm{X}\ldots\mathrm{X}})$.

Following the notations and definitions given in [6] and [9], a test algorithm consists of a sequence of TEs. Each TE has a number of TCAM operations (test operations), possible with

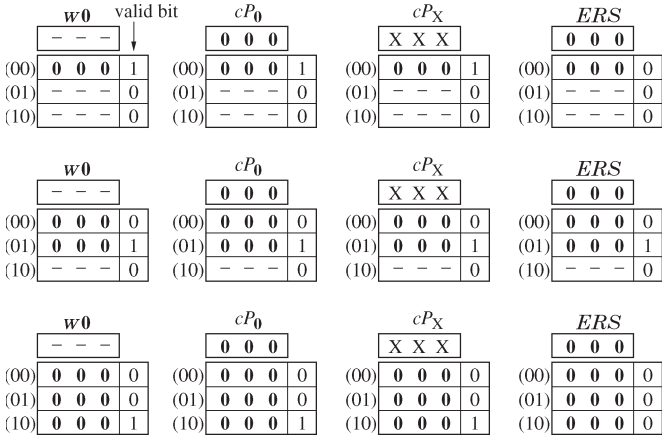

Fig. 8. Fault-free status of a 3 × 3-bit TCAM when TE2 of T_{tcam1} is executed.

a prespecified address sequence, which can be ascending ($\Uparrow$), descending ($\Downarrow$), or either way ($\Updownarrow$). Each Compare operation within the TEs TE5 and TE7 is executed on all words of a TCAM, so no prespecified address sequence is needed and the number of test operations is equal to the number of bits in a word. Thus, if a TCAM with B-bit words is considered, each one of the two TEs individually has B test operations.

The notations for representing the TCAM operations include: 1) wD—write an input pattern to the addressed word and set the corresponding Valid bit to valid; 2) cP_D—compare an input pattern D with all words in the TCAM; 3) Erase Operation (ERS)—set the Valid bit of the addressed word to invalid. For brevity, the D may only consist of one-bit data for denoting B-bit homogeneous data or multiple-bit data for expressing B-bit heterogeneous data. For example, if a TCAM with 4-bit words is tested, then $w\mathbf{1}$ represents a Write operation with data **1111**, and $cP_{\mathrm{X...X0}}$ denotes the Compare operation with the comparand XXX**0**.

Subsequently, we describe the T_{tcam1} in detail. TE1 first resets all the Valid bits to invalid. Then, TE2 performs a Write-**0**, a Compare-**0**, a Compare-X, and an Erase operation for each word in either ascending or descending address sequence. An example is illustrated to explain the TE2 in detail. Fig. 8 depicts the fault-free status of a 3 × 3-bit TCAM when the TE2 is executed, where the data in the parenthesis denotes the address and—denotes the data is unknown. Also, the figures in the first row, the second row, and the third row denote the fault-free status of the TCAM when the TE2 applies test operations on Word_{00}, Word_{01}, and Word_{10}, respectively. Once the TE1 is completed, all the valid bits of the TCAM are set to 0. Then, when the TE2 applies the test operations on the first word, the fault-free status of the TCAM is shown in the first row of Fig. 8. The Write-**0** operation writes all-**0** data into Word_{00} and sets the corresponding Valid bit to 1. Subsequently, the Compare-**0** operation is performed and the comparison result is M if Word_{00} is fault-free. On the contrary, if the comparison result is MM, the Word_{00} has comparison faults. Then, the TCAM executes Compare-X operation and the comparison result is M if Word_{00} is fault-free. Again, if Word_{00} has comparison faults, the Hit output will be MM. Finally, the Erase operation resets the valid of Word_{00} to 0. In a similar way, the four operations of

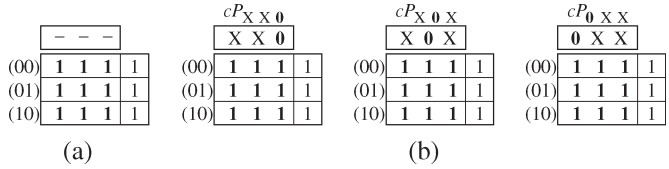

Fig. 9. (a) Fault-free status when TE4 of T_{tcam1} is executed. (b) Fault-free status when TE5 of T_{tcam1} is executed.

TE2 will be performed on the other words in sequence. Therefore, each bit of the TCAM undergoes $(w\mathbf{0}, c\mathbf{0})$ and $(w\mathbf{0}, c\mathrm{X})$ operations and the corresponding comparison results can be observed by the Hit output when TE2 is completed. In the same way, we can see that all bits of the TCAM undergoes $(w\mathbf{1}, c\mathbf{1})$ and $(w\mathbf{1}, c\mathrm{X})$ operations and the corresponding comparison results can be exported to Hit output when the test operations of TE3 are finished.

When the execution of TE4 is completed, the status of the TCAM under test is all-**1** and the Valid bits of all words are set to valid as shown in Fig. 9(a), where a 3 × 3-bit TCAM is considered as an example. Then, the TE5 executes Compare-**0** operation for each bit of all words. As Fig. 9(b) shows, the TE5 executes three Compare operations. The first Compare operation compares the comparand XX**0** with all the words. Because the first two bits of the comparand are Xs, only the **0** is compared with all the last bits of the words. If the bits of the last column are fault free, then the Hit $= 0$. However, if any one of bits is faulty and its faulty response is M, then the Hit $= 1$. When all the Compare operations of the TE5 are performed, every cell of the TCAM has undergone the $(w\mathbf{1}, c\mathbf{0})$ operations, and the corresponding fault effect, M, can be observed at Hit output. In the same way, we see that the TEs TE6 and TE7 verify whether every TCAM cell can correctly execute the $(w\mathbf{0}, c\mathbf{1})$ operations.

According to the discussion above, we see that the proposed test algorithm T_{tcam1} verifies the following six Compare-after-Write operations $(w\mathbf{0}, c\mathbf{0})$, $(w\mathbf{0}, c\mathrm{X})$, $(w\mathbf{1}, c\mathbf{1})$, $(w\mathbf{1}, c\mathrm{X})$, $(w\mathbf{1}, c\mathbf{0})$, and $(w\mathbf{0}, c\mathbf{1})$ on every cell in the TCAM. Table XI summarizes the failed response of a TCAM cell which undergoes the six different Compare-after-Write operations and the corresponding TE which detects the fail.

Subsequently, we show that the T_{tcam1} can detect 100% comparison faults. As Table X shows, we see that if every TCAM cell of a TCAM under test undergoes a Compare-after-Write operation-$(w\mathbf{0}, c\mathbf{0})$, and the corresponding TCAM cell response can be observed by Hit output after the operation, then the comparison faults, SMMF, PMC**1**F, CMM**0**F, CSMM**0**F, SC**0**SMMF, and SDCMMF, can be detected since their faulty response is MM and it is different from the fault-free response. Especially, the comparison fault D**0**UMF and D**1**UMF only can be detected by executing the $(w\mathbf{0}, c\mathrm{X})$ and $(w\mathbf{1}, c\mathrm{X})$ operations, respectively. By the same way, we can see that if every TCAM cell also can execute the following operations: $(w\mathbf{0}, c\mathbf{1})$, $(w\mathbf{1}, c\mathbf{1})$, and $(w\mathbf{1}, c\mathbf{0})$ and the corresponding responses can be observed, then all the comparison faults listed in the Table X can be detected. According to Table XI, therefore, we conclude that the T_{tcam1} can detect 100% comparison faults.

TABLE XI
FAILED RESPONSES OF COMPARE-AFTER-WRITE OPERATIONS WITH RESPECT TO THE DETECTION TEST ELEMENTS OF $T_{\rm tcam1}$

Operation	Failed Response	Detection Test Element
$w\mathbf{0}$,$c\mathbf{0}$	MM	TE2
$w\mathbf{0}$,cX	MM	TE2
$w\mathbf{1}$,$c\mathbf{1}$	MM	TE3
$w\mathbf{1}$,cX	MM	TE3
$w\mathbf{1}$,$c\mathbf{0}$	M	TE5
$w\mathbf{0}$,$c\mathbf{1}$	M	TE7

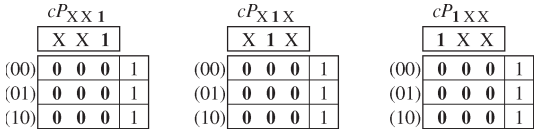

Fig. 10. Fault-free status of a 3 × 3-bit TCAM when TE2 of $T_{\rm tcam2}$ is executed.

B. Test for TCAMs With Hit and PAE Outputs

This section describes a test for detecting comparison faults of TCAMs with Hit and PAE outputs. That is, the comparison results can be observed by the Hit and PAE outputs. Without loss the generality, here the lowest matched address with the highest priority is assumed. The proposed test ($T_{\rm tcam2}$) for TCAMs with Hit and PAE outputs is summarized as follows:

TE1: $\Updownarrow (w\mathbf{0})$;
TE2: $(cP_{\rm X...X\mathbf{1}}, cP_{\rm X...\mathbf{1}X}, \ldots, cP_{\mathbf{1}\rm X...X})$.
TE3: $\Uparrow (cP_{\mathbf{0}}, cP_{\rm X}, {\rm ERS})$;
TE4: $\Updownarrow (w\mathbf{1})$;
TE5: $(cP_{\rm X...X\mathbf{0}}, cP_{\rm X...\mathbf{0}X}, \ldots, cP_{\mathbf{0}\rm X...X})$; and
TE6: $\Uparrow (cP_{\mathbf{1}}, cP_{\rm X}, {\rm ERS})$.

TE1 initializes the TCAM to all-$\mathbf{0}$ state. Then, TE2 performs B Compare operations for an $N \times B$-bit TCAM. Also, each Compare operation only compares one-bit data with the corresponding bit of the TCAM. Fig. 10 depicts the fault-free status of a 3 × 3-bit TCAM when the TE2 is executed. The left figure shows the fault-free status when the TCAM executes $cP_{\rm XX1}$. If the corresponding bits in the TCAM do not have comparison faults, the output of PAE is invalid or the output of Hit is MM. However, if the PAE exports any one of valid addresses (i.e., 00, 01, or 10) or the Hit outputs M, the corresponding bits with comparison faults in the first column are detected. Therefore, each bit of the TCAM undergoes the $(w\mathbf{0}, c\mathbf{1})$ operations and the corresponding comparison results can be observed by the Hit or PAE outputs when TE2 are completed.

Subsequently, TE3 performs the test operations ($cP_{\mathbf{0}}$, $cP_{\rm X}$, ERS) in ascending address sequence. Fig. 11 shows the fault-free status of a 3 × 3-bit TCAM when TE3 is performed. As the first row of Fig. 11 shows, when $\rm Word_{00}$ is addressed and $cP_{\mathbf{0}}$ is executed, the matched address is (00) if $\rm Word_{00}$ is fault-free. Otherwise, the matched address is not (00) if $\rm Word_{00}$ is faulty. Similarly, when the $cP_{\rm X}$ is performed, we can analyze the responses analogously. The last operation of TE3 resets the Valid bit of the first word to invalid. Therefore, all bits in the first word undergoes $(w\mathbf{0}, c\mathbf{0})$ and $(w\mathbf{0}, c$X$)$ operations and the corresponding comparison results can be observed by the PAE output when TE3 completes the test operations on the

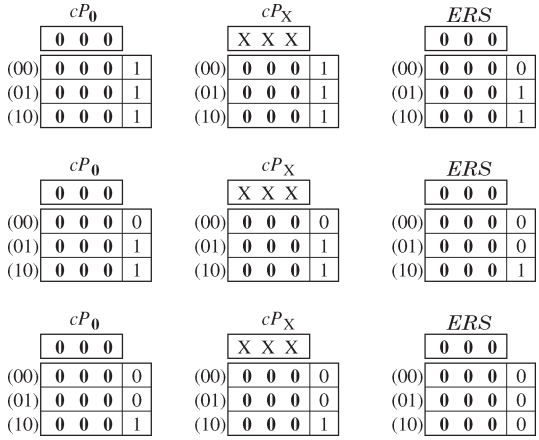

Fig. 11. Fault-free status of a 3 × 3-bit TCAM when element ⇑ $(cP_{\mathbf{0}}, cP_{\rm X}, {\rm ERS})$ is executed.

TABLE XII
FAILED RESPONSES OF COMPARE-AFTER-WRITE OPERATIONS WITH RESPECT TO THE DETECTION TEST ELEMENTS OF $T_{\rm tcam2}$

Operation	Failed Response	Detection Test Element
$w\mathbf{0}$,$c\mathbf{0}$	Unexpected address	TE3
$w\mathbf{0}$,cX	Unexpected address	TE3
$w\mathbf{1}$,$c\mathbf{1}$	Unexpected address	TE6
$w\mathbf{1}$,cX	Unexpected address	TE6
$w\mathbf{1}$,$c\mathbf{0}$	M or Unexpected address	TE5
$w\mathbf{0}$,$c\mathbf{1}$	M or Unexpected address	TE2

first word. In the same way, we see that all bits of the second and the third words can undergo the same operations when the TE3 performs test operations on the second word and the third word. Therefore, all bits of the TCAM undergo $(w\mathbf{0}, c\mathbf{1})$, $(w\mathbf{0}, c\mathbf{0})$, and $(w\mathbf{0}, c$X$)$ and the corresponding responses can be checked when TE1, TE2, and TE3 are completed. Apparently, when TE4, TE5, and TE6 also are completed, all bits of the TCAM undergo $(w\mathbf{1}, c\mathbf{0})$, $(w\mathbf{1}, c\mathbf{1})$, and $(w\mathbf{1}, c$X$)$ operations and whether the TCAM has the comparison faults can be checked by the PAE output.

Consequently, the proposed test algorithm $T_{\rm tcam2}$ verifies the following six Compare-after-Write operations $(w\mathbf{0}, c\mathbf{0})$, $(w\mathbf{0}, c$X$)$, $(w\mathbf{1}, c\mathbf{1})$, $(w\mathbf{1}, c$X$)$, $(w\mathbf{1}, c\mathbf{0})$, and $(w\mathbf{0}, c\mathbf{1})$ on every cell in the TCAM. Table XII summarizes the failed response of a TCAM cell, which undergoes these Compare-after-Write operations, and the corresponding TE, which detects the fail. Apparently, we can conclude that the $T_{\rm tcam2}$ can cover 100% comparison faults.

V. TESTING DELAY FAULTS IN COMPARISON CIRCUIT

The proposed test algorithms $T_{\rm tcam1}$ and $T_{\rm tcam2}$ also can cover the delay faults in comparison circuits of a TCAM. Before showing the detection ability of delay faults of the two test algorithms, we first describe the critical paths in the comparison circuits of a TCAM. Fig. 12 depicts a simplified TCAM schematic diagram. Before a TCAM executes a Compare operation, all comparison lines first are reset to low, such that all pull-down paths in all MLs are turned-off regardless of the

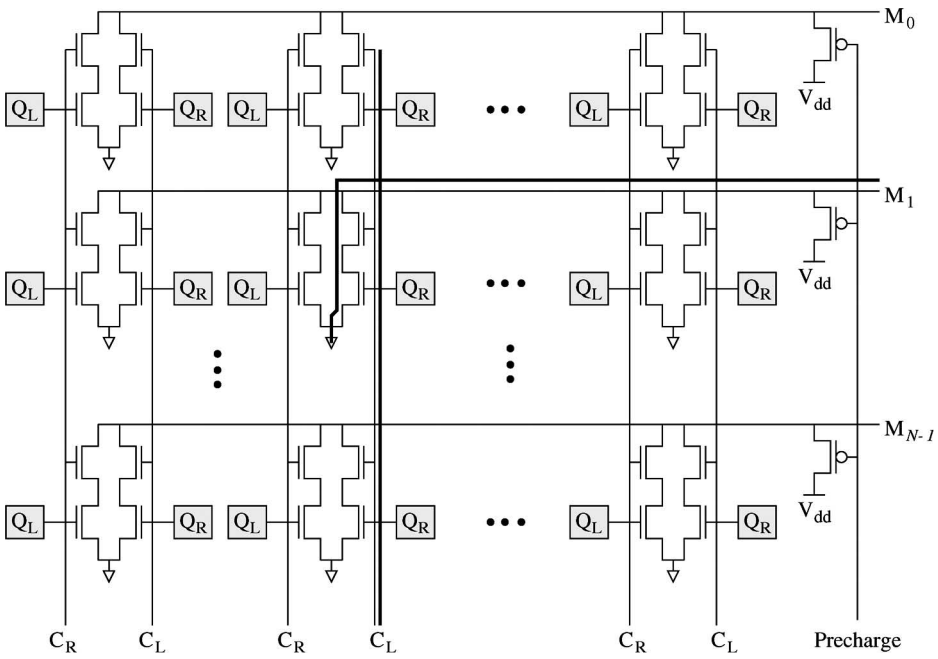

Fig. 12. Simplified TCAM schematic diagram.

stored data of the TCAM. Then, all MLs $(\mathrm{M}_0, \mathrm{M}_1, \ldots, \mathrm{M}_{N-1})$ are precharged to V_{dd}. Subsequently, the comparand is applied through comparison lines. If at least one bit stored data of a word is different from the comparand, the corresponding ML is discharged to low through one of the pull-down paths of the corresponding bit. The ML of a word is slowest to fall when only one pull-down path is turned-on and the NMOS transistors controlled by the comparison lines of the other pull-down paths in the word must be turned-off. If the NMOS transistors are not turned-off, the drain-diffusion capacitances existing between the two NMOS transistors of the pull-down paths will share a portion of the charge of the MLs. This shortens the discharging time of the ML. Thus, the ML of a word is slowest to fall when only one bit in the word mismatches the corresponding bit of the comparand and the other bits of the comparand are Xs. Also, if a test will cover 100% delay faults in comparison circuits, the test must check whether each pull-down path of all MLs in the TCAM, i.e., the right and left pull-down paths of each TCAM cell must be checked.

Subsequently, we show that the proposed tests T_{tcam1} and T_{tcam2} can cover all the delay faults of comparison circuits. Consider the TEs TE4 and TE5 of T_{tcam1}. The TE4 writes all-**1** data into all the words of the TCAM. In each TCAM cell, the state of $(Q_{\mathrm{L}}, Q_{\mathrm{R}})$ is (1, 0). Then, TE5 performs a sequence of Compare operations with walking-**0** in the all Xs comparands from right to left, i.e., comparands are $(\mathrm{X}\ldots\mathrm{X}\mathbf{0}), (\mathrm{X}\ldots\mathrm{X}\mathbf{0}\mathrm{X}), \ldots, (\mathbf{0}\mathrm{X}\ldots\mathrm{X})$. When the Compare operation with the comparand $(\mathrm{X}\ldots\mathrm{X}\mathbf{0})$ is performed, only the left pull-down path of the comparison logic of each TCAM cell in the last column is turned-on. Then, delay faults caused by any one of the left pull-down paths of the TCAM cells in the last column can be detected. Similarly, if a Compare operation with the comparand $(\mathrm{X}\ldots\mathrm{X}\mathbf{0}\mathrm{X})$ is executed, delay fault caused by any one of the left pull-down paths of the TCAM cells in the last second column can be detected. Therefore, delay fault caused by any one of the left pull-down paths of the TCAM cells in the TCAM can be detected while TE4 and TE5 are performed. In a similar way, delay fault caused by any one of the right pull-down paths of the TCAM cells in the TCAM can be detected while TE6 and TE7 of T_{tcam1} are performed. Since T_{tcam2} also has the four TEs, T_{tcam2} can cover the delay faults in the comparison circuits of TCAMs as well.

In addition to the defects causing the critical path delay can be tested, resistive defects causing the delay in the pull-down and pull-up of C_{R} and C_{L} also can be detected by the T_{tcam1} and T_{tcam2}. The reason is explained as follows. If a resistive defect exists in the comparison line and causes the comparison line to have a slow-to-rise transition or slow-to-fall transition, it will cause an incorrect comparison result. For more detail, if comparison lines have slow-to-fall transition faults, they can easily be detected since the comparison lines must be reset to 0 before precharing the MLs. Therefore, if comparison lines have slow-to-fall transition faults, the corresponding MLs cannot be precharged to V_{dd}. This causes an incorrect comparison result. On the other hand, if comparison lines have slow-to-rise transition faults, they can be detected by the TE5 and TE7. Consider the TEs TE4 and TE5 of T_{tcam1}. The TE4 writes all-**1** data into all the words of the TCAM. In each TCAM cell, the state of $(Q_{\mathrm{L}}, Q_{\mathrm{R}})$ is (1, 0). Then, TE5 performs a sequence of Compare operations with comparands $(\mathrm{X}\ldots\mathrm{X}\mathbf{0}), (\mathrm{X}\ldots\mathrm{X}\mathbf{0}\mathrm{X}), \ldots, (\mathbf{0}\mathrm{X}\ldots\mathrm{X})$. While the first comparand is applied, the rightest comparand is from X to **0**, since all comparsion lines must be reset to X while the MLs are p-recharged. Therefore, the C_{R} of the rightest bit will undergo a 0-to-1 transition. If the C_{R} has a resistive defect and causes a slow-to-rise transition, the received comparand by the TCAM cells in the bit line will still be X while the Compare operation is executed. This results in the comparsion result will be match which is different from the expected comparison result mismatch. Therefore, whether all the C_{R} lines have a

TABLE XIII
COMPARISON RESULTS OF THE $T_{\rm tcam1}$, $T_{\rm tcam2}$, AND SPTA FOR AN $N \times B$-bit TCAM

	SPTA [18]	$T_{\rm tcam1}$	$T_{\rm tcam2}$
Complexity	$2N$+$\log_2 B\times$SON Writes $\frac{2NB}{\log_2 N}$+$\log_2 B\times$SON Compares	$4N$ Writes $(4N{+}2B)$ Compares $3N$ Erases	$2N$ Writes $(4N{+}2B)$ Compares $2N$ Erases
Fault Model	Transistor-level faults	Cell-level faults	Cell-level faults
Fault Observation	PAE	Hit	PAE
Delay Fault Detection	No	Yes	Yes

TABLE XIV
COMPARISON OF THE TEST COMPLEXITIES OF SPTA, $T_{\rm tcam1}$, AND $T_{\rm tcam2}$ FOR TCAMS WITH VARIOUS CONFIGURATIONS

Product	Configurations	(SPTA)	($T_{\rm tcam1}$)	($T_{\rm tcam2}$)	$\frac{(T_{\rm tcam1})}{(\rm SPTA)}$%	$\frac{(T_{\rm tcam2})}{(\rm SPTA)}$%
TCAM1	64K×36	416K	704K+72	512K+72	48.8%	35.5%
	32K×72	371.2K	352K+144	256K+144		
	16K×144	361K	176K+288	128K+288		
TCAM2	256K×36	1536K	2816K+72	2048K+72	27.5%	20.0%
	128K×72	1340K	1408K+144	1024K+144		
	64K×144	1280K	704K+288	512K+288		
	32K×288	1292.8K	352K+576	256K+576		

slow-to-rise transition can be checked while all the Compare operations of TE5 are completed. In a similar way, whether all the $C_{\rm L}$ lines have a slow-to-rise transition can be checked while all Compare operations of TE7 are completed.

VI. ANALYSIS AND COMPARISON

Several test schemes for TCAMs have been proposed in [17]–[19]. In [17], a BIST scheme for performing a proposed test algorithm for TCAMs was presented. Each row of the TCAM is composed of three fields: valid, tag, and data. The test algorithm consists of three phases: valid testing, tag testing, and data testing. Each row is divided into two subrows and a Write operation only can write data into one subrow at a time. The BIST testing of the TCAM is configured to test the TCAM cores separate from PAE. This enables BIST testing of one row of the TCAM core at a time. Then, the testing power can be reduced. However, this causes that the test complexity of the test algorithm is $O(NB)$ for an $N \times B$-bit TCAM. Therefore, the testing time is very long. In [19], the test algorithm is developed based on a TCAM with word parallel access and bit parallel access capabilities. Therefore, the test algorithm only requires $(30B + 3)$ Write operations and $36B$ Compare operations to test an $N \times B$-bit TCAM. However, some design issues must be considered if the parallel access is allowed. For example, the bitline drivers must be size-up drastically such that the parallel write can be done in a regular Write cycle. On the other hand, the issues of transient IR and $L(di/dt)$ voltage drops in the power and ground lines will arise while parallel word access is executed. Therefore, many design issues must be resolved if the user want to use this test scheme. According to the discussion above, we see that both the two previous works [17], [19] need DFT insertions. Also, the test algorithm reported in [17] is very time consuming, and many design issues must be resolved to support the test scheme reported in [19]. Therefore, only the test scheme reported in [18] is compared with the proposed test scheme in detail, since these two test schemes do not need DFT insertion and time complexity of the two test schemes is only proportional to N and B.

In [18], a SPTA was proposed to detect the transistor stuck-on and stuck-open faults in the search paths of a TCAM. That is, only transistor stuck-on and stuck-open defects in the pull-down paths of a TCAM are assumed. Table XIII summarizes the comparison results of the SPTA, $T_{\rm tcam1}$, and $T_{\rm tcam2}$ for $N \times B$-bit TCAMs. The second row reports the time complexity of the three test algorithms. The SPTA requires $2N + \log_2 B \times$ SON Write operations and $(2NB/\log_2 N) + \log_2 B \times$ SON Compare (Search) operations to cover 100% stuck-on and stuck-open faults in the search paths, where SON represents the number of stuck-on faults in the search-path transistors. The $T_{\rm tcam1}$ requires $4N$ Write operations, $4N + 2B$ Compare operations, and $3N$ Erase operations. The $T_{\rm tcam2}$ needs $2N$ Write operations, $4N + 2B$ Compare operations, and $2N$ Erase operations. These two tests can cover 100% defined comparison faults.

If consider the best case for the SPTA, i.e., SON $= 0$ and the Erase, Compare, and Write operations have the same cycle time, then the SPTA requires $2N + (2NB/\log_2 N)$ operations. The $T_{\rm tcam1}$ and $T_{\rm tcam2}$ need $11N + 2B$ and $8N + 2B$ operations, respectively. Assume that $N \gg 2B$, this is true for most of TCAM products. Thus, we can approximately estimate the time complexities of $T_{\rm tcam1}$ and $T_{\rm tcam2}$ as $11N$ and $8N$ operations. Therefore, if $(B/\log_2 N) > 4.5$, then the complexity of $T_{\rm tcam1}$ is lower than the complexity of SPTA. If $(B/\log_2 N) > 3$, then the needed test operations of $T_{\rm tcam2}$ are lower than those of SPTA. These are true for most used TCAMs. For example, we compare the test complexity of the three test algorithms based on commercial TCAM products. A commercial TCAM usually has various configurations on the number of words and the width of a word. Table XIV summarizes the comparison results of the SPTA, $T_{\rm tcam1}$, and $T_{\rm tcam2}$, where (SPTA), $(T_{\rm tcam1})$, and $(T_{\rm tcam2})$ denote the number of test operations of the tests SPTA, $T_{\rm tcam1}$, and $T_{\rm tcam2}$, respectively. The second column of Table XIV lists

TABLE XV
COMPARISON OF THE NUMBER OF COMPARE OPERATIONS OF SPTA, $T_{\rm tcam1}$, AND $T_{\rm tcam2}$ FOR TCAMS WITH VARIOUS CONFIGURATIONS

Product	Configurations	$(SPTA)_C$	$(T_{tcam1})_C$=$(T_{tcam2})_C$	$\frac{(T_{tcam1})_C}{(SPTA)_C}$ %
TCAM1	64K×36	288K	256K+72	19.5%
	32K×72	307.2K	128K+144	
	16K×144	329K	64K+288	
TCAM2	256K×36	1024K	1024K+72	11.2%
	128K×72	1084K	512K+144	
	64K×144	1152K	256K+288	
	32K×288	1228.8K	128K+576	

the possible configurations of the TCAM1 and TCAM2. Note that the TCAMs under test without transistor stuck-on faults is assumed. The third, fourth, and fifth columns show the number of test operations with respect to the TCAM configurations for the tests SPTA, $T_{\rm tcam1}$, and $T_{\rm tcam2}$, respectively. The sixth column shows the ratio of $(T_{\rm tcam1})$ to (SPTA), and the last column shows the ratio of $(T_{\rm tcam2})$ to (SPTA). The ratios are calculated by selecting the minimum number of test operations from each test. That is, we can configure the TCAM into a best configuration for testing such that the test time is minimized. For example, we can configure TCAM1 as 16K $\times$ 144-bit for the testing purpose, since this configuration results in the lowest time complexity for these three tests. As Table XIV shows, we see that the time complexity of $T_{\rm tcam1}$ or $T_{\rm tcam2}$ is much lower than that of SPTA.

In TCAMs, a Compare operation consumes much more time than a Write or an Erase operation. The reason is that subblock access method can be used for Read/Write/Erase operations in the TCAM, but it cannot be applied for the Compare operation. Moreover, the comparison result must be evaluated by the Hit Signal Generator and PAE. This causes that the Compare operation is more time consuming. For example, working frequencies of the Write and Compare operations of the TCAM reported in [28] are 175 and 155 MHz, respectively. Therefore, we further compare the number of Compare operations of the three tests. Table XV shows the comparison results of the number of Compare operations of the tests SPTA, $T_{\rm tcam1}$, and $T_{\rm tcam2}$. In the table, $(SPTA)_C$, $(T_{\rm tcam1})_C$, and $(T_{\rm tcam2})_C$ denote the number of Compare operations of SPTA, $T_{\rm tcam1}$, and $T_{\rm tcam2}$, respectively. As Table XV shows, the ratio of the number of Compare operations of $T_{\rm tcam1}$ or $T_{\rm tcam2}$ to the number of Compare operations of SPTA is only about 19.5% and 11.2% for TCAM1 and TCAM2. Here, the number of Compare operations is calculated by selecting the TCAM configurations with the minimum number of total test operations. For example, 16K $\times$ 144 configuration for TCAM1 is selected for calculating the Compare operations for the tests, since this configuration results in the minimum number of total test operations (see Table XIV).

Subsequently, we describe the difference of fault models for the three tests. The transistor-level faults, i.e., transistor stuck-on and stuck-open, are used to develop the SPTA. Only the defects that cause the transistors of search paths to become stuck-on or stuck-open are detected. However, many defects may not cause the transistors in the search path to become stuck-on or stuck-open, but they also cause the search paths

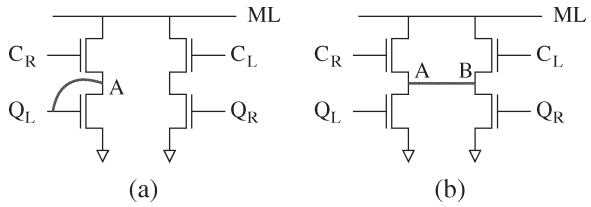

Fig. 13. Examples of short defects.

fail. As Fig. 13(a) shows, for example, if a cell exists a short between the storage node and the node A, then the short cannot be modeled as stuck-on or stuck-open faults since the transistor can correctly be operated but the search path cannot correctly perform Compare operation. Fig. 13(b) shows another example, if a short defect exists between the nodes A and B, the cell also cannot correctly execute the Compare operation. Again, this defect cannot cause a transistor stuck-on fault or stuck-open fault. Thus, the SPTA may not detect the defect. According to the description in Section III, however, we see that many short defects cause a failed Compare operation. On the contrary, the cell-level faults are used to develop the $T_{\rm tcam1}$ and $T_{\rm tcam2}$. Therefore, defects that causes a failed Compare operation can be detected by these two tests. Also, the two tests can be used to test TCAMs with different comparator structures.

As the last row of Table XIII, moreover, the $T_{\rm tcam1}$ and $T_{\rm tcam2}$ can cover the delay faults of comparison circuits in TCAMs (as Section V describes). However, the SPTA cannot cover the delay faults in the comparison circuits, since the test operations of SPTA cannot activate the slowest delay path. The reason is that at least two pull-down paths are turned ON when the SPTA performs a Compare operation and the comparand and data are different. Finally, the $T_{\rm tcam1}$ and $T_{\rm tcam2}$ have the following limitations. They lacks the test procedure for intercell fault detection and column level diagnostics for stuck-on faults. Moreover, the test procedure does not verify if the X value can be properly stored and compared.

VII. CONCLUSION

In this paper, we have proposed two functional tests for the TCAMs. Major comparison fault models for the TCAMs have been discussed. The comparison faults are defined based on electrical defects such as shorts between two nodes, transistor stuck-on, and transistor stuck-open. The two proposed functional tests, $T_{\rm tcam1}$ and $T_{\rm tcam2}$, can cover the comparison faults and delay faults in comparison circuits. The $T_{\rm tcam1}$ requires $4N$ Write operations, $3N$ Erase operations, and $4N + 2B$ Compare operations to cover 100% comparison faults for an $N \times B$-bit TCAM with Hit output only. The $T_{\rm tcam1}$ requires $2N$ Write operations, $2N$ Erase operations, and $4N + 2B$ Compare operations to cover 100% comparison faults for an $N \times B$-bit TCAM with Hit and PAE outputs. Compared with the previous work, the $T_{\rm tcam1}$ and $T_{\rm tcam2}$ have the following major advantages. First, they can be used to test different TCAMs with various cell structures. Second, they require fewer test operations for testing typical TCAMs. Especially, they use much fewer Compare operations. Third, they can cover the delay faults in comparison circuits. Fourth, their time complexities are not related to the number of stuck-on faults. But the proposed tests have the

following limitations. They lacks the test procedure for intercell fault detection and column level diagnostics for stuck-on faults, and the test procedure does not verify if the X value can be properly stored and compared.

ACKNOWLEDGMENT

The author would like to thank Associate Editor K. Chakrabarty and the anonymous reviewers for their helpful comments to improve this paper. The author would also like to thank C.-K. Lin and S.-H. Yang for their help on some of the simulation results.

REFERENCES

[1] V. Ravikumar, R. N. Mahapatra, and L. N. Bhuyan, "EaseCAM: An energy and storage efficient TCAM-based router architecture for IP lookup," *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 521–533, May 2005.

[2] W. Wu, J. Shi, L. Zuo, and B. Shi, "Power-efficient TCAMs for bursty access patterns," *IEEE Micro*, vol. 25, no. 4, pp. 64–72, Aug. 2005.

[3] M.-J. Akhbarizadeh, M. Nourani, and C. D. Cantrell, "Prefix segregation scheme for a TCAM-based IP forwarding engine," *IEEE Micro*, vol. 25, no. 4, pp. 48–63, Aug. 2005.

[4] W. K. Al-Assadi, A. P. Jayasumana, and Y. K. Malaiya, "On fault modeling and testing of content-addressable memories," in *Proc. IEEE Int. Workshop MTDT*, 1994, pp. 78–81.

[5] P. R. Sidorowicz and J. A. Brzozowski, "An approach to modeling and testing memories and its application to CAMs," in *Proc. IEEE VTS*, Apr. 1998, pp. 411–416.

[6] K.-J. Lin and C.-W. Wu, "Testing content-addressable memories using functional fault models and March-like algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 5, pp. 577–588, May 2000.

[7] J. Zhao, S. Irrinki, M. Puri, and F. Lombardi, "Testing SRAM-based content addressable memories," *IEEE Trans. Comput.*, vol. 49, no. 10, pp. 1054–1063, Oct. 2000.

[8] J.-F. Li, R.-S. Tzeng, and C.-W. Wu, "Testing and diagnosing embedded content addressable memories," in *Proc. IEEE VTS*, Monterey, CA, Apr. 2002, pp. 389–394.

[9] J.-F. Li, R.-S. Tzeng, and C.-W. Wu, "Testing and diagnosis methodologies for embedded content addressable memories," *J. Electron. Test.: Theory Appl.*, vol. 19, no. 2, pp. 207–215, Apr. 2003.

[10] X. Du, S. M. Reddy, J. Rayhawk, and W.-T. Cheng, "Testing delay faults in embedded CAMs," in *Proc. IEEE ATS*, 2003, pp. 378–383.

[11] J.-F. Li, "Diagnosing binary content addressable memories with comparison and RAM faults," *IEICE Trans. Inf. Syst.*, vol. E87-D, no. 3, pp. 601–608, Mar. 2004.

[12] D. K. Bhavsar, "A built-in self-test method for write-only content addressable memories," in *Proc. IEEE VTS*, May 2005, pp. 9–14.

[13] J.-F. Li, "Testing priority address encoder faults in content addressable memories," presented at the Proc. ITC, pp. 1–8, Austin, TX, Nov. 2005, Paper 33.2.

[14] P. Mazumder, J. H. Patel, and W. K. Fuchs, "Methodologies for testing embedded content addressable memories," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 7, no. 1, pp. 11–20, Jan. 1988.

[15] Y. S. Kang, J. C. Lee, and S. Kang, "Parallel BIST architecture for CAMs," *Electron. Lett.*, vol. 33, no. 1, pp. 30–31, Jan. 1997.

[16] S. R. Ramirez-Chavez, "Encoding don't cares in static and dynamic content addressable memories," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 8, pp. 575–578, Aug. 1992.

[17] S. Gupta and R. Gibson, "Methods and circuitry for built-in self-testing of content addressable memories," R.O.C. Patent 6 609 222 B1, Aug. 19, 2003.

[18] D. Wright and M. Sachdev, "Transistor-level fault analysis and test algorithm development for ternary dynamic content addressable memories," in *Proc. ITC*, Sep. 2003, pp. 39–47.

[19] K.-J. Lee, C. Kim, S. Kim, U.-R. Cho, and H.-G. Byun, "Modeling and testing of faults in TCAMs," in *Proc. Asian Simul. Conf.*, Jeju Island, Korea, Oct. 2004, pp. 521–528.

[20] J.-F. Li, "Testing comparison faults of ternary CAMs based on comparison faults of binary CAMs," in *Proc. ASP-DAC*, Shanghai, China, Jan. 2005, pp. 65–70.

[21] J.-F. Li and C.-K. Lin, "Modeling and testing comparison faults for ternary content addressable memories," in *Proc. IEEE VTS*, Palm Springs, CA, May 2005, pp. 60–65.

[22] Y.-J. Huang and J.-F. Li, "Testing active neighborhood pattern-sensitive faults of ternary content addressable memories," in *Proc. IEEE ETS*, Southampton, U.K., May 2006, pp. 55–60.

[23] I. MOSAID Technologies. (1999). *The Next Generation of Content Addressable Memories*. [Online]. Available: http://www.mosaid.com/

[24] V. Lines, A. Ahmed, P. Ma, S. Ma, R. McKenzie, H.-S. Kim, and C. Mar, "66 MHz 2.3 M ternary dynamic content addressable memory," in *Proc. IEEE Int. Workshop MTDT*, San Jose, CA, Aug. 2000, pp. 101–105.

[25] J.-F. Li, K.-L. Cheng, C.-T. Huang, and C.-W. Wu, "March-based RAM diagnosis algorithms for stuck-at and coupling faults," in *Proc. ITC*, Baltimore, MD, Oct. 2001, pp. 758–767.

[26] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*. Chichester, U.K.: Wiley, 1991.

[27] Z. Al-Ars and A. J. V. de Goor, "Static and dynamic behavior of memory cell array spot defects in embedded DRAMs," *IEEE Trans. Comput.*, vol. 52, no. 3, pp. 293–309, Mar. 2003.

[28] G. Bollano, S. Claretto, E. Filippi, A. Torielli, and M. Turolla, "Merging hardware and software: Intellectual property cores for internet applications," in *Proc. IEEE CICC*, Orlando, FL, May 2000, pp. 537–540.

Jin-Fu Li (S'01–M'03) received the B.S. degree from the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C., in 1995, and the M.S. and Ph.D. degrees, both from the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1999 and 2002, respectively.

Since 2002, he has been with the Department of Electrical Engineering, National Central University, Jhongli, Taiwan, R.O.C., where he is currently an Associate Professor. From July 2006 to June 2007, he also is a Visiting Scholar with the ECE Department, University of California, Santa Barbara, where he joins the system-on-a-chip (SOC) Design and Test Laboratory supervised by Prof. K.-T. (Tim) Cheng. His research interests include advanced very large scale integration (VLSI)/SOC design and test, memory testing and repair, and reliable VLSI circuits and systems.

Dr. Li received the Industry Academy Collaboration Award from the Taiwan Ministry of Education in 2003 and the Ph.D. Thesis Award from the Taiwan IC Design (TICD) Society in 2003. He is a Life Member of Chinese Institute of Electrical Engineering and TICD Society.