Testing Comparison Faults of Ternary Content Addressable Memories with Asymmetric Cells

Jin-Fu Li
Advanced Reliable Systems (ARES) Laboratory
Department of Electrical Engineering
National Central University
Jhongli, Taiwan, 320

***Abstract*— Ternary content addressable memory (TCAM) is one key component in the dedicated hardware modulars for high-performance networking applications. Symmetric and asymmetric cells are two widely used cell structures in TCAMs. An asymmetric cell consists of a binary content addressable memory (BCAM) bit and a mask bit. This paper proposes two march-like test algorithms, T_{Hit} and T_{PAE}, to cover the comparison faults of the BCAM cell and the comparison logic faults of the masking cell. T_{Hit} requires $7N$ Write operations and ($3N$+$2B$) Compare operations to cover the comparison faults of an $N \times B$-bit TCAM with Hit output only. T_{PAE} requires $4N$ Write operations and ($3N$+$2B$) Compare operations to cover the comparison faults of an $N \times B$-bit TCAM with priority address encoder (PAE) output.**

I. Introduction

Ternary content addressable memories (TCAMs) are widely used in the network applications. Moreover, emerging applications require the longest match searches, such as flow analysis and classless inter domain routing. TCAMs can provide a high-quality solution for these applications. But, their special and complicated functions cause that the TCAM testing is very difficult.

Most of the previous CAM testing schemes targeted the testing of binary CAMs (BCAMs), e.g., [1]–[8]. In [1], the authors proposed test algorithms for detecting stuck-at faults, static pattern-sensitive faults (PSFs), and dynamic PSFs. In [2], a functional fault model for BCAMs was derived by investigating the functional failures in the storage cell and comparison logic. In [4], an approach for modeling and testing memories and its application to BCAMs was introduced. In [5], comparison faults were defined and March-like tests were proposed to detect the comparison faults of BCAMs. The March-like tests were developed on a BCAM with the Hit output only. In [6], test algorithms for BCAMs which can perform Read and Compare operations concurrently were proposed. Also, the comparison result observed by the priority address encoder is assumed. In [7], tests for covering comparison faults and RAM faults were reported. In [8], a test methodology for detecting delay faults of BCAMs was proposed.

Recently, several research works on the testing of TCAMs were presented in [9]–[15]. In [9], a built-in self-testing scheme for TCAMs was presented. The BIST tests one row of the TCAM at a time to 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 [10], [11], a search path test algorithm 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. In [12], the test algorithm is developed based on a TCAM with parallel access capabilities. So 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 [13], 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 and with Hit/PAE outputs were introduced in [14]. In [15], test algorithms for detecting active NPSFs of TCAMs were proposed.

Previous TCAM test algorithms were developed on the TCAM with symmetric cells. This paper proposes two test algorithms to test comparison faults of TCAMs with asymmetric cells. The test algorithms are developed by decomposing an asymmetric TCAM cell into a BCAM bit and a mask bit. The first test algorithm T_{Hit} uses $7N$ Write operations and ($3N$+$2B$) Compare operations to detect 100% targeted comparsion faults of an $N \times B$-bit TCAM with Hit output only. The second test algorithm T_{PAE} uses $4N$ Write operations and ($3N$+$2B$) Compare operations to detect 100% targeted comparsion faults of an $N \times B$-bit TCAM with the PAE output.

II. Preliminary

A. TCAM Architecture and Cell Structure

Figure 1 depicts a typical $N \times B$-bit CAM organization. The Address Decoder and Data I/O 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 match signal of the corresponding word is valid or invalid. If a TCAM (BCAM) is considered, the cells in Cell Array are replaced by TCAM (BCAM) cells. When the CAM executes Compare operation, the data 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,

1081-7735/07 $25.00 © 2007 IEEE
DOI 10.1109/ATS.2007.55

and generates a hit output (Hit=1) if there is at least one valid match. The priority address encoder exports the highest priority matched address (either the lowest matched address or the highest matched address).

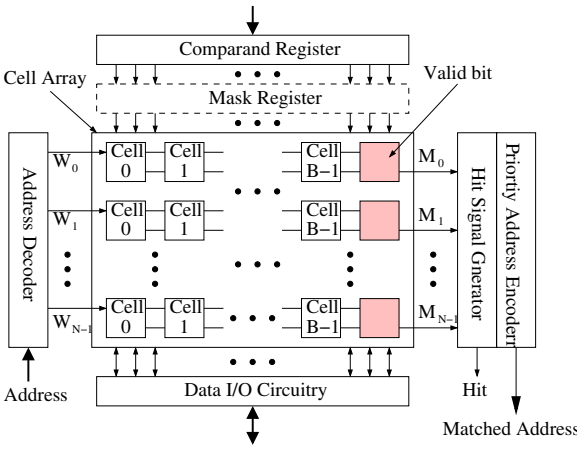

Fig. 1. A typical $N \times B$-bit CAM organization.

The major difference between a BCAM and a TCAM is that the TCAM cell can store three states "logic 0", "logic 1", and "don't care (X)" states. The X state enables TCAM to execute partial match between the comparand and data bits. However, the BCAM executes only the exact match between the comparand and data bits. To store three states, a TCAM cell consists of two storage elements. Two widely used TCAM cell structures are symmetric cell and asymmetric cell. A symmetric TCAM cell is composed of two identical bits as shown in Fig. 2(a) [16], where each bit consists of a storage bit and a comparison logic. An asymmetric cell consists of a BCAM bit and a mask bit as shown in Fig. 2(b) [17]. In this paper, we focus on the testing of the TCAM with asymmetric cells. Therefore, we introduce the operations of an asymmetric cell in more detail. When the TCAM executes a Compare operation, the ML is pre-charged to V_{DD} in the precharge phase. In the evaluation phase, the value of ML is determined by the status of the four NMOS transistors: m1, m2, m3, and m4. As Fig. 2(b) shows, if the lower SRAM cell stores the data 0, i.e., Q_L=0, then the TCAM cell is masked. Thus, the ML will remain at V_{DD} since the m4 is turned off. On the contrary, if Q_L=1, then the ML value is determined by the comparison result of the upper BCAM bit. In this paper, we assume that (Q_U=0,Q_L=1)=**0**, (Q_U=1,Q_L=1)=**1**, and (Q_U=0,Q_L=0)=X, where **0**, **1**, and X denote the ternary data. Also, the representation of comparand is as follows: (C_U=0,$\overline{C}_U$=1)=**0**, (C_U=1,$\overline{C}_U$=0)=**1**, and (C_U=0,$\overline{C}_U$=0)=X.

B. Comparison Faults of BCAMs

In this subsection, the comparison faults of BCAMs defined in [5] are described. Definitions of the comparison faults are as follows. (1) stuck-matched fault (SMF)—a cell always matches the corresponding input bit regardless of the BCAM cell state and input pattern; (2) stuck-mismatched fault

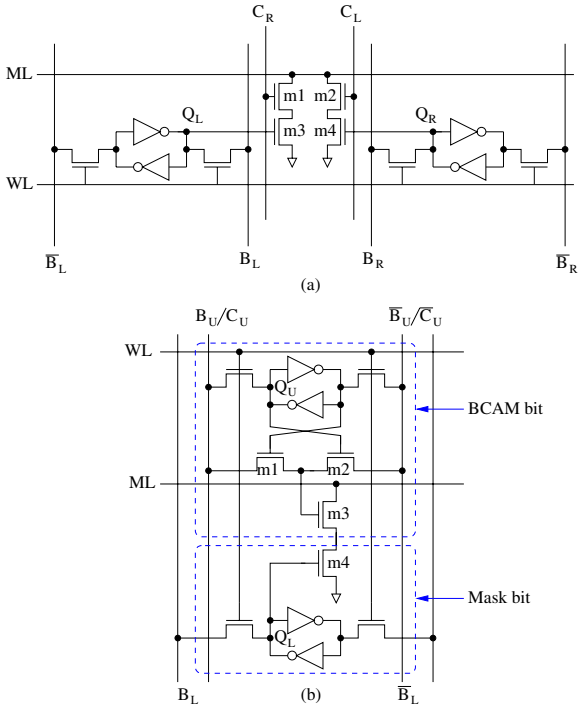

Fig. 2. (a) Symmetric TCAM cell. (b) Asymmetric TCAM cell.

(SMMF)—there is no match for the cell irrespective of the BCAM cell state and input pattern; (3) partial-match fault (PMF)—a BCAM cell is stuck-matched for all subsequent Compare operations when a logic value x (either 0, i.e., PM0F, or 1, i.e., PM1F) is written into the cell, and stuck-mismatched when $\overline{x}$ is written into it; (4) conditional-match fault (CMF)—a cell function is correct if it stores a logic value x (either 0, i.e., CM0F, or 1, i.e.,CM1F), but it always provides an incorrect result for the subsequent Compare operations if it stores $\overline{x}$; (5) equivalence-mismatch fault (EMMF)—the Compare operation fails if the BCAM cell stores a value x (either 0, i.e., EMM0F, or 1, i.e., EMM1F) and is compared with the same input value x; (6) inequivalence-match fault (IMF)–the Compare operation fails if the BCAM cell stores a value x (either 0, i.e., IM0F, or 1, i.e., IM1F) and is compared with the input value $\overline{x}$.

Table I summarizes the cell responses to all Compare-after-Write operations under various single cell comparison faults [5]. In the table, "M" and "MM" denote the "match" and "mismatch" results of the corresponding (wx, cy) operations, respectively, where $x, y \in \{0,1\}$ and wx and cy denote a Write-x operation and a Compare-y operation, respectively. For example, if a cell has a conditional-match-0 fault (CM0F), then the behavior of the faulty cell is as follows. If a 0 is stored in the cell, then the cell has a correct Compare operation, but its Compare result is incorrect if the cell stores a 1. Possible defects causing these comparison faults are reported in [5].

III. Fault Analysis of Asymmetric TCAM Cells

In this paper, we focus on the testing of comparison faults. Consider the asymmetric TCAM cell shown in Fig. 2(b). If

TABLE I

BCAM CELL RESPONSE TO COMPARE-AFTER-WRITE OPERATION [5].

	$w0, c0$	$w0, c1$	$w1, c0$	$w1, c1$
SMF	M	M	M	M
SMMF	MM	MM	MM	MM
CM1F	MM	M	MM	M
CM0F	M	MM	M	MM
PM1F	MM	MM	M	M
PM0F	M	M	MM	MM
EMM1F	M	MM	MM	MM
EMM0F	MM	MM	MM	M
IM1F	M	MM	M	M
IM0F	M	M	MM	M

logic 1 is stored in the mask bit, i.e., Q_L=1, then the equivalent circuit of the TCAM cell is a BCAM cell as shown in Fig. 3(a). Therefore, we can test the TCAM cell as the BCAM cell. Thus, the BCAM comparison faults can be used to cover the defects which cause the comparison function of the TCAM cell to fail. Any defects which cause the transistor m4 to be stuck-open also can be covered. Because this defect causes the TCAM cell always match the corresponding comparand regardless of the data stored in the TCAM cell, the defect can be covered by the SMF.

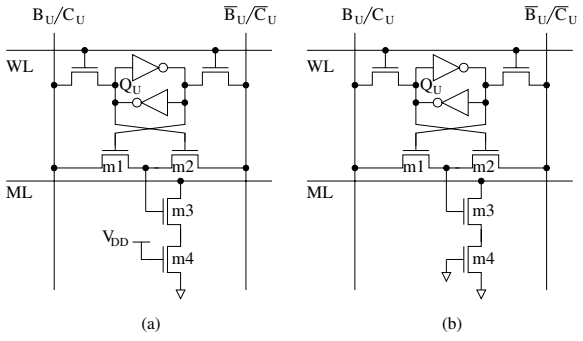

Fig. 3. Equivalent circuit of the asymmetric TCAM cell when (a) Q_L=1 and (b) Q_L=0.

However, defects cause that the transistor m4 to be stuck-on can not be covered. To sensitize this type of defects, the mask bit of the TCAM cell must store logic 0 value, i.e., Q_L=0. Figure 3(b) shows the equivalent circuit of the TCAM cell when Q_L=0. Also, the gate of m3 must be set to logic 1 to propagate the fault effect. To store logic 0 in the mask bit, a wX operation should be executed on the TCAM cell. The wX operation also forces the BCAM bit to store logic 0, i.e., Q_U=0. Subsequently, a $c\mathbf{0}$ should be executed to set the gate of m3 to logic 1, since the Compare operation sets (C_U, $\overline{C}_U$) to (0, 1). Therefore, a $c\mathbf{0}$ after the wX operation must be executed on the TCAM cell to cover the defects which cause the m4 to be stuck-on. According to the discussion above, we can regard the comparison fault testing of the asymmetric TCAM cell as follows: the testing of the comparison faults of the BCAM bit when the mask bit stores logic 1; and the testing of the stuck-on faults of the transistor m4 when the mask bit stores logic 0.

IV. TESTING COMPARISON FAULTS

A. Test for TCAMs with Hit Output Only

In this subsection, we propose a test algorithm, called T_{Hit}, for detecting the comparison faults of TCAMs with Hit output only. The test algorithm is as follows:

TE 1: $\Updownarrow (w\mathbf{1})$;
TE 2: $\Updownarrow (w\mathrm{X}, cP_{\mathbf{0}}, w\mathbf{1})$;
TE 3: $\Updownarrow (w\mathbf{0}, cP_{\mathbf{0}}, w\mathbf{1})$;
TE 4: $(cP_{\mathrm{X...X0}}, cP_{\mathrm{X...0X}}, \ldots, cP_{\mathbf{0}\mathrm{X...X}})$;
TE 5: $\Updownarrow (w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0})$;
TE 6: $(cP_{\mathrm{X...X1}}, cP_{\mathrm{X...1X}}, \ldots, cP_{\mathbf{1}\mathrm{X...X}})$

T_{Hit} consists of six test elements (TEs). Each test element (TE) has a number of TCAM operations (test operations), possible with a prespecified address sequence, which can be ascending ($\Uparrow$), descending ($\Downarrow$), or either way ($\Updownarrow$). Each Compare operation within the fourth and sixth test elements is executed on all words of a TCAM, so no prespecified address sequence is needed and the number of test operations depends on the width of a word. Thus if a TCAM with B-bit words is considered, the fourth or sixth test element 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. For brevity, the D may only consist of one-bit data for denoting a B-bit homogeneous data for B-bit words or multiple-bit data for expressing 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**.

According to Table I, we have the following observation. If every BCAM bit of the TCAM under test undergoes the following five Compare-after-Write operations, $(w0, c0)$, $(w0, c1)$, $(w1, c0)$, and $(w1, c1)$ while the mask bit stores logic 1, and the corresponding TCAM cell response can be observed by TCAM output after each operation, then the possible faults listed in the table can be detected. Thus, a TCAM cell should undergoes ($w\mathbf{0}$, $c\mathbf{0}$), ($w\mathbf{0}$, $c\mathbf{1}$), ($w\mathbf{1}$, $c\mathbf{0}$), and ($w\mathbf{1}$, $c\mathbf{1}$) and the corresponding TCAM cell response should be observed by the TCAM output. Also, as Section III describes, if a TCAM cell undergoes (wX, $c\mathbf{0}$) operations and the corresponding response can be observed by TCAM output, then the faults causes the m4 to be stuck-on can be detected. Therefore, we should show that T_{Hit} can guarantee that every TCAM cell can undergo the five Compare-after-Write operations, ($w\mathbf{0}$, $c\mathbf{0}$), ($w\mathbf{0}$, $c\mathbf{1}$), ($w\mathbf{1}$, $c\mathbf{0}$), ($w\mathbf{1}$, $c\mathbf{1}$), and (wX, $c\mathbf{0}$) , and the corresponding cell response can be observed by the TCAM output.

Subsequently, we use a 3×3-bit TCAM as an example to explain the T_{Hit}. Also, the test elements executed in the ascending address sequence are assumed. In T_{Hit}, the TE1 initializes the TCAM array to the all-**1** state. Figure 4 shows the fault-free status of the TCAM when the test element TE2 is executed. The first row of Fig. 4 shows the status of the TCAM when the W_0 is addressed. When the $cP_{\mathbf{0}}$ is executed, the fault-free Hit output should be 1 (Hit=$M_0|M_1|M_2$, where

| denotes a bit-wise OR operation), since the state of W_0 is all-X and this causes M_0 to be 1. However, if there is any faulty bit in W_0 and its faulty response is mismatch, then the Hit becomes 0. Therefore, every bit of W_0 undergoes a (wX, $c\mathbf{0}$) operation and its faulty response can be observed by Hit output. In a similar way, we can see that every cell of the TCAM can undergo the (wX, $c\mathbf{0}$) operations and the corresponding fault effect, Mismatch, can be observed at Hit output when the TE2 is completed.

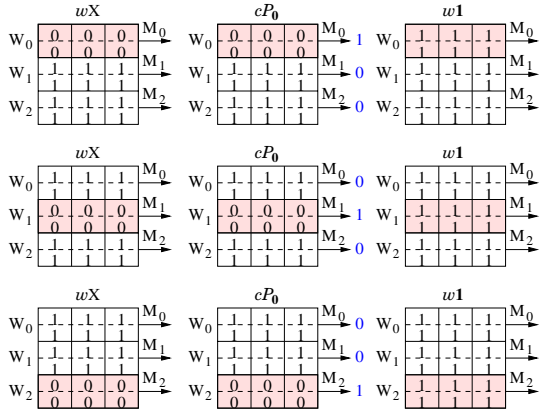

Fig. 4. Fault-free status of a 3×3-bit TCAM when TE2 is executed.

Figure 5 shows the fault-free status of the 3×3-bit TCAM when the TE3 is executed. The first row of Fig. 5 shows the status of the TCAM when the W_0 is addressed. When the $cP_\mathbf{0}$ is executed, the fault-free Hit output should be 1. However, if there is any bit is faulty in W_0 and its faulty response is mismatch, then the Hit becomes 0. Thus, every bit of W_0 can undergo a ($w\mathbf{0}$, $c\mathbf{0}$) operation and its faulty response can be observed by Hit output. In a similar way, as Fig. 5 shows, every cell of the TCAM has undergone the ($w\mathbf{0}, c\mathbf{0}$) operations and the corresponding fault effect, Mismatch, can be observed at Hit output when the TE3 is completed.

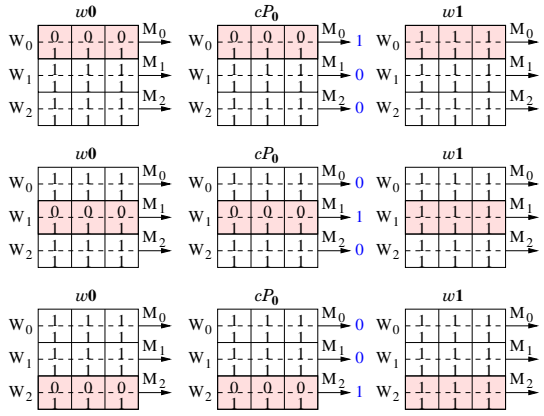

Fig. 5. Fault-free status of a 3×3-bit TCAM when TE3 is executed.

After the execution of TE3, the fault-free status of the TCAM array will be all-**1** state. Subsequently, the TE4 is executed. Figure 6 shows the fault-free status of the TCAM when TE4 is executed. As Fig. 6 shows, the TE4 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 Match, then the Hit=1. When all the Compare operations of the TE4 are performed, every cell of the TCAM has undergone the ($w\mathbf{1}, c\mathbf{0}$) operations, and the corresponding fault effect, Match, can be observed at Hit output.

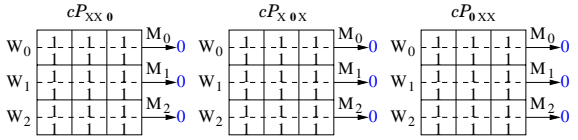

Fig. 6. Fault-free status of a 3×3-bit TCAM when TE4 is executed.

In a similar way, we can analyze the fault detection capability of the TE5 and TE6, since these two test elements are similar to the test elements TE3 and TE4. When the TE5 and TE6 are completed, every TCAM cell can undergo the ($w\mathbf{1}$, $c\mathbf{1}$) and ($w\mathbf{0}$, $c\mathbf{1}$) operations and the corresponding cell responses can be observed by the Hit output.

According to the discussion above, we see that the proposed test algorithm T_{Hit} can verify the five Compare-after-Write operations, ($w\mathbf{0}$,$c\mathbf{0}$), ($w\mathbf{0}$,$c\mathbf{1}$), ($w\mathbf{1}$,$c\mathbf{1}$), ($w\mathbf{1}$,$c\mathbf{0}$), and (wX,$c\mathbf{0}$), on every cell in the TCAM. Table II summarizes the fail response of a TCAM cell undergoing Compare-after-Write operations and the corresponding test element detecting the fail. Therefore, the T_{Hit} can detect 100% comparison faults of the TCAM with asymmetric TCAM cells.

TABLE II
COMPARE-AFTER-WRITE OPERATIONS CORRESPONDING TO THE DETECTION TEST ELEMENTS OF T_{Hit}.

Operation	Fail Response	Detection Test Element
wX/$c\mathbf{0}$	MM	TE2
$w\mathbf{0}$/$c\mathbf{0}$	MM	TE3
$w\mathbf{1}$/$c\mathbf{0}$	M	TE4
$w\mathbf{1}$/$c\mathbf{1}$	MM	TE5
$w\mathbf{0}$/$c\mathbf{1}$	M	TE6

B. Test for TCAMs with PAE Output

In this subsection, we propose a test algorithm, called T_{PAE}, for detecting the comparison faults of TCAMs with the priority address encoder (PAE) output. Without loss the generality, here the lowest matched address with the highest priority is assumed. The test algorithm T_{PAE} is as follows:

TE 1: $\Updownarrow (w\mathbf{1})$;
TE 2: $\Downarrow (w\mathbf{0}, cP_\mathbf{0})$;
TE 3: $(cP_{\text{X...X}\mathbf{1}}, cP_{\text{X...}\mathbf{1}\text{X}}, \ldots, cP_{\mathbf{1}\text{X...X}})$;

TE 4: $\Downarrow (w\mathbf{1}, cP_{\mathbf{1}})$;
TE 5: $(cP_{\mathrm{X}\ldots\mathrm{X}\mathbf{0}}, cP_{\mathrm{X}\ldots\mathrm{0X}}, \ldots, cP_{\mathbf{0}\mathrm{X}\ldots\mathrm{X}})$;
TE 6: $\Downarrow (wX, cP_{\mathbf{0}})$

TE1 initializes the TCAM to all-**1** state. Then, TE2 executes ($w\mathbf{0}$, $cP_{\mathbf{0}}$) operation on the TCAM in descending address sequence. Figure 7 shows the fault-free status of a 3×3-bit TCAM when TE2 is executed. As the first row of Fig. 7 depicts, when W_2 is addressed and $cP_{\mathbf{0}}$ is executed, the match address of the PAE output is W_2. If the PAE outputs an invalid address, then every TCAM cell in W_2 fails to undergo ($w\mathbf{0}$, $c\mathbf{0}$) and the cell response is observed. Subsequently, when W_1 is addressed and the fault-free status of the TCAM is shown in the second row of Fig. 7. When the $cP_{\mathbf{0}}$ is executed, the comparand matches the data of W_2 and W_1, i.e., M_2=1 and M_1=1. However, the match address of the PAE output is W_1 since the lowest matched address with the highest priority is assumed. But, if the PAE outputs W_2 instead of W_1, then every TCAM cell in W_1 fails to undergo ($w\mathbf{0}$, $c\mathbf{0}$) and the cell response is observed. Therefore, every TCAM cell can undergo ($w\mathbf{0}$, $c\mathbf{0}$) operation and the cell response can be observed on the PAE output when the TE2 is completed.

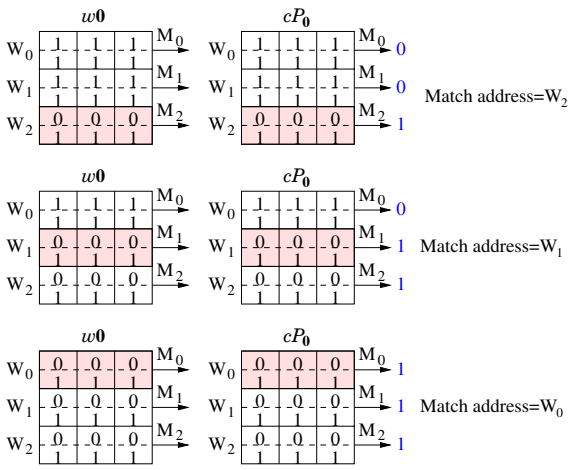

Fig. 7. Fault-free status of a 3×3-bit TCAM when TE2 is executed.

Figure 8 shows the fault-free status of a 3×3-bit TCAM when TE3 is performed. The analysis of TE3 of T_{PAE} is similar to that of TE4 of T_{Hit}. The difference is that here the comparison result is observed by the PAE output. As Fig. 8 shows, if the TCAM is fault-free, then the PAE outputs an invalid matched address when a Compare operation is executed. On the other hand, if the PAE outputs a matched address, then the TCAM has faulty cells. Therefore, every TCAM cell can undergo ($w\mathbf{0}$, $c\mathbf{1}$) operation and the cell response can be checked when TE3 is completed.

In a similar way, we can analyze the fault detection capability of the TE4 and TE5, since these two test elements are similar to the test elements TE2 and TE3. When the TE4 and TE5 are completed, every TCAM cell can undergo the ($w\mathbf{1}$, $c\mathbf{1}$) and ($w\mathbf{1}$, $c\mathbf{0}$) operations and the corresponding cell responses can be observed by the PAE output. Figure 9

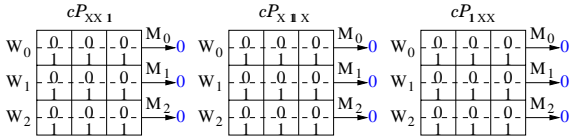

Fig. 8. Fault-free status of a 3×3-bit TCAM when TE3 is executed.

depicts the fault-free status when the TE6 is performed. As Fig. 9 shows, the analysis of fault detection capability of the TE6 is similar to that of TE2. Therefore, every TCAM cell can undergo (wX, $c\mathbf{0}$) operation and the corresponding cell response can be verified when TE6 is completed.

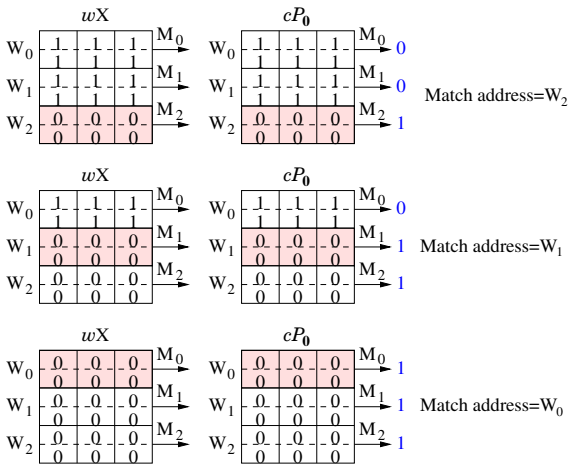

Fig. 9. Fault-free status of a 3×3-bit TCAM when TE6 is executed.

According to the discussion above, we see that the proposed test algorithm T_{PAE} can verify the following five Compare-after-Write operations, ($w\mathbf{0}$,$c\mathbf{0}$), ($w\mathbf{0}$,$c\mathbf{1}$), ($w\mathbf{1}$,$c\mathbf{1}$), ($w\mathbf{1}$,$c\mathbf{0}$), and (wX,$c\mathbf{0}$), on every cell in the TCAM. Table III summarizes the fail response of a TCAM cell undergoing Compare-after-Write operations and the corresponding test element detecting the fail. Thus, we conclude that the T_{PAE} can detect 100% comparison faults of the TCAM with asymmetric TCAM cells.

TABLE III
COMPARE-AFTER-WRITE OPERATIONS CORRESPONDING TO THE DETECTION TEST ELEMENTS OF T_{PAE}.

Operation	Fail Response	Detection Test Element
$w\mathbf{0}/c\mathbf{0}$	MM	TE2
$w\mathbf{0}/c\mathbf{1}$	M	TE3
$w\mathbf{1}/c\mathbf{1}$	MM	TE4
$w\mathbf{1}/c\mathbf{0}$	M	TE5
wX/$c\mathbf{0}$	MM	TE6

V. ANALYSIS AND COMPARISON

According to the descriptions of Section IV, we see that the T_{Hit} requires $7N$ Write operations and ($3N$+$2B$) Compare operations to cover comparison faults of an $N\times B$-bit TCAM with Hit output only. If an $N\times B$-bit TCAM with PAE output

TABLE IV

COMPARISON RESULTS OF DIFFERENT TESTS FOR AN $N \times B$-BIT TCAM.

	SPTA [10]	T_{tcam1} [14]	T_{tcam2} [14]	T_{Hit}	T_{PAE}
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	$7N$ Writes $(3N+2B)$ Compares	$4N$ Writes $(3N+2B)$ Compares
Fault Model	Transistor-level faults	Cell-level faults	Cell-level faults	Cell-level faults	Cell-level faults
Fault Observation	PAE	Hit	PAE	Hit	PAE
Cell Types	Symmetric Cell	Symmetric Cell	Symmetric Cell	Asymmetric Cell	Asymmetric Cell

is considered, then the T_{PAE} can be used to detect the comparison faults using $4N$ Write operations and $(3N+2B)$ Compare operations. That is, the TCAM with PAE output can be tested easily. The reason is that the PAE can provide higher observability for the comparison results.

Subsequently, we compare the proposed tests with the previous works reported in [10], [14]. In [10], a search path test algorithm (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. In [14], the fault modeling and testing of TCAMs with symmetric cells was investigated. Although the tests in [14] are developed based on the symmetric cell structure, the tests also can be used for TCAMs with asymmetric cells since the cell-level faults are targeted. Table IV summarizes the comparison results of the SPTA, T_{tcam1}, T_{tcam2}, T_{Hit}, and T_{PAE} for $N \times B$-bit TCAMs. The second row reports the time complexity of the three test algorithms. The test time complexity of SPTA is related to the number of transistor stuck-on and stuck-open faults in the m1, m2, m3, and m4 of Fig. 2(a). The time complexities of T_{tcam1} and T_{tcam2} are larger than those of T_{Hit} and T_{PAE}, respectively. Although the T_{tcam1} and T_{tcam2} can also detect the comparison faults of TCAMs with asymmetric cells, the tests T_{Hit} and T_{PAE} developed using the decomposition concept have lower time complexity. That is, if the user develops a test algorithm based on the targeted TCAM cell structure, then the test complexity of the test algorithm can be optimized based on the cell structure.

VI. CONCLUSIONS

In this paper we have presented two March-like test algorithms for TCAMs with asymmetric cells. The test algorithms are developed by decomposing an asymmetric TCAM cell into a BCAM bit and a mask bit. The first test algorithm T_{Hit} uses $7N$ Write operations and $(3N+2B)$ Compare operations to detect 100% targeted comparsion faults of an $N \times B$-bit TCAM with Hit output only. The second test algorithm T_{PAE} uses $4N$ Write operations and $(3N+2B)$ Compare operations to detect 100% targeted comparsion faults of an $N \times B$-bit TCAM with PAE output.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, under Contract NSC 95-2221-E-008-152 and MOEA, Taiwan, under Contract 96-EC-17-A-01-S1-002.

REFERENCES

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

[2] W. K. Al-Assadi, A. P. Jayasumana, and Y. K. Malaiya, "On fault modeling and testing of content-addressable memories," in *Proc. IEEE Int'l Workshop on Memory Technology, Design and Testing (MTDT)*, 1994, pp. 78–81.

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

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

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

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

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

[8] X. Du, S. M. Reddy, J. Rayhawk, and W.-T. Cheng, "Testing delay faults in embedded CAMs," in *IEEE Asian Test Symp. (ATS)*, 2003, pp. 378–383.

[9] S. Gupta and R. Gibson, "Methods and circuitry for built-in self-testing of content addressable memories," R.O.C. Patent No. 6609222 B1, Aug. 2003.

[10] D. Wright and M. Sachdev, "Transistor-level fault analysis and test algorithm development for ternary dynamic content addressable memories," in *Proc. Int'l Test Conf. (ITC)*, Sep. 2003, pp. 39–47.

[11] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "Design techniques and test methodology for low-power TCAMs," *IEEE Trans. on VLSI Systems*, vol. 14, no. 6, pp. 573–586, June 2006.

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

[13] J.-F. Li, "Testing comparison faults of ternary CAMs based on comparison faults of binary CAMs," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Shanghai, Jan. 2005, pp. 65–70.

[14] ——, "Testing ternary content addressable memories using march-like tests," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 5, pp. 919–931, May 2007.

[15] Y.-X. Yang, J.-F. Li, H.-N. Liu, and C.-L. Wey, "Design of cost-efficient memory-based FFT processors using single-port memories," in *IEEE International SOC Conference*, Hsinchu, Sept. 2007 (submitted).

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

[17] 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.