Middle-East Journal of Scientific Research 23 (8): 1921-1929, 2015
ISSN 1990-9233
© IDOSI Publications, 2015
DOI: 10.5829/idosi.mejsr.2015.23.08.22505

Design & Testing of Tcam Faults Using T_h Algorithm

[1]M. Anto Bennet, [1]G. Sankar Babu, [1]R. Suresh,
[1]S. Mohammed Sulaiman, 1M. Sheriff, [2]G. Janakiraman and [3]S. Natarajan

[1]Department of ECE, VELTECH, Chennai, India
[2]Department of Automobile Engg,
Dr. Mahalingam College of Engg & Tech, Pollachi, India
[3]Department of EEE, VELTECH, Chennai, India

Abstract: Ternary content addressable memory (TCAM) is one key component in high-performance networking applications. An asymmetric TCAM cell consists of a binary content addressable memory (BCAM) bit and a mask bit. In this paper, we analyze comparison faults of the asymmetric TCAM cell based on BCAM comparison faults. Then march-like test algorithms T_H are proposed to cover the comparison faults of the comparison circuits in TCAMs with asymmetric cells. The test algorithm T_H requires 7N Write operations and (3N+2B) Compare operations to cover the comparison faults of an N × B bit TCAM with Hit output only; and the T_{PAE} test algorithm requires 4N Write operations and (3N+2B) Compare operations to cover the comparison faults of an N × B bit TCAM with priority address encoder (PAE) output.

Key words: Ternary content addressable memory (TCAM) · Content-addressable memory (CAM) · T_H Algorithm · Comparand Register · BCAM cell

INTRODUCTION

Content-addressable memory (CAM) is a special type of computer memory used in certain very high speed searching applications. It is also known as associative memory, associative storage, or associative array, although the last term is more often used for a programming data structure. Binary CAM is the simplest type of CAM which uses data search words consisting entirely of 1s and 0s. Ternary CAM (TCAM) allows a third matching state of "X" or "Don't Care" for one or more bits in the stored data word, thus adding flexibility to the search. For example, a ternary CAM might have a stored word of "10XX0" which will match any of the four search words "10000", "10010", "10100", or "10110". The added search flexibility comes at an additional cost over binary CAM as the internal memory cell must now encode three possible states instead of the two of binary CAM. This additional state is typically implemented by adding a mask bit ("care" or "don't care" bit) to every memory cell. CAM is widely used to design high-performance search engines. CAMs can be classified into two categories, binary CAMs (BCAMs) and ternary CAMs (TCAMs), according to the number of states which a CAM cell can represent. Emerging applications require the longest match searches, such as flow analysis and classless inter domain routing. Testing these large embedded or stand-alone TCAMs is becoming an important issue. The special TCAM cell structure and complicated function cause that the TCAM testing is very difficult [1].

A TCAM cell consists of a storage (two SRAM cells) and a comparator for executing high speed parallel comparison. That is, in addition to the faults in comparator (comparison faults), the faults in the storage (RAM 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. The testing of RAMs has been considered a mature area of research and many existing algorithms can provide adequate coverage for the RAM faults [2]. Therefore, popular March tests can be used to test the RAM faults of TCAMs. However, the tests for comparison faults and the tests for RAM faults can be combined to reduce the test cost. A CAM basically implements the hush function in hardware; apart from the traditional functions of a memory (READ and WRITE), a

Corrresponding Author: M. Anto Bennet, Department of ECE, VELTECH, Chennai, India.

CAM has the additional function of MATCH CAMs which can be implemented using SRAMs (Static Random Access Memories); a cell of an SRAM-based CAM (dual port) uses nine transistors. An SRAM based CAM is substantially different from a traditional SRAM because it requires a larger number of transistors to perform the additional function of MATCH (as well as WRITE and READ). The difference in functionality and the internal structure of the cell also affect the testing process of these devices. CAMs are word-oriented storage devices that allow instantaneous searches of the memory content for a given key word. This type of memory plays a vital role in caches, table-look aside, ATM switches and other ASICs where searching is a critical operation. CAMs have been used in fields which need to have fast data search such as computer graphics, image processing, artificial intelligence, logic comparison and database machines, etc. In addition, the possibility of the general purpose usage of CAMs has increased greatly, since the bit density and chip size have been recently enhanced. A new test methodology is devised which can test CAMs in a far more efficient way. The read operation in CAMs can be replaced by one parallel content addressable search operation and peripheral circuit structures are modified to achieve parallel writing. For the functional testing of RAMS, the various physical defects in the address decoder, the memory cell array and the read/write logic are mapped onto the fault models. Internet protocol lookup forms a bottleneck in packet forwarding in modern IP routers because the lookup speed is unable to catch up with the increase in link bandwidth. Ternary Content Addressable Memories have been emerging as viable devices for designing high throughput forwarding engines on routers. They store don't care states in addition to 0s and 1s and search the data in parallel by providing a comparator in each cell. This property makes TCAMs particularly attractive for packet forwarding and classifications, where variable length prefix matching is necessary in every cycle. Content-addressable memories are a class of memories in which data is accessed on the basis of content instead of data-location address. Such memories are distributed-logic fine-grain parallel processing computational structures capable of supporting a wide range of numeric and non-numeric computational tasks. Their computing power lies in their ability of accessing data based on content independent of the number of words that reside in the memory. Furthermore, their regular and iterable structure makes them highly suitable for VLSI implementations that require intensive data management. CAMs have been used in applications such as memory management, database machines, image processing and artificial intelligence.In a CAM device implementing the exact-match operation, the memory is searched in parallel for words that exactly match an input word. In this type of CAM architecture, stored words are compared independently and simultaneously with the input word CAMs support a wide range of computing and communications applications, such as artificial intelligence, data compression, signal processing, large-scale database management and address filtering or translation in computer and communication networks. Although traditionally CAMs have much lower capacity than SRAMs due to the overhead from the extra comparison logic, high-density large-capacity CAMs are more and more popular due to the emerging applications. Testing these large CAMs (embedded or stand-alone), thus, is becoming an important issue [3-6].

MATERIALS AND METHODS

The search path consists of the match line and the four discharge transistors on the cell. A successful test algorithm will detect if any one of these transistors is SON or SOP, but an efficient test algorithm will do it quickly [7, 8]. The algorithm presented in this paper exploits the fact that the search function returns address information. It detects discrepancies between the expected returned addresses and the actual returned addresses. This algorithm will detect all SL and BL transistor SON and SOP faults. It was assumed during the development of this algorithm that the CAM is able to return all matched addresses sequentially, as well as the CAM being modeled [9-11]. In the case of a CAM that only returns the highest priority match, all matching addresses can be returned in sequence by writing mismatching values to the returned address. When a matching address is returned, the complement of the search value is written to the address, thus ensuring a mismatch. The next highest priority address will then be returned. Once the searching is complete the original values can be re-written to the affected addresses [12-15].

Proposed Method: 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 network applications. Testing these large embedded or stand-alone TCAMs is becoming an important issue. The comparison faults of the asymmetric TCAM cell based on BCAM comparison faults are proposed. Then march-like test algorithms T_H are proposed to cover the comparison faults

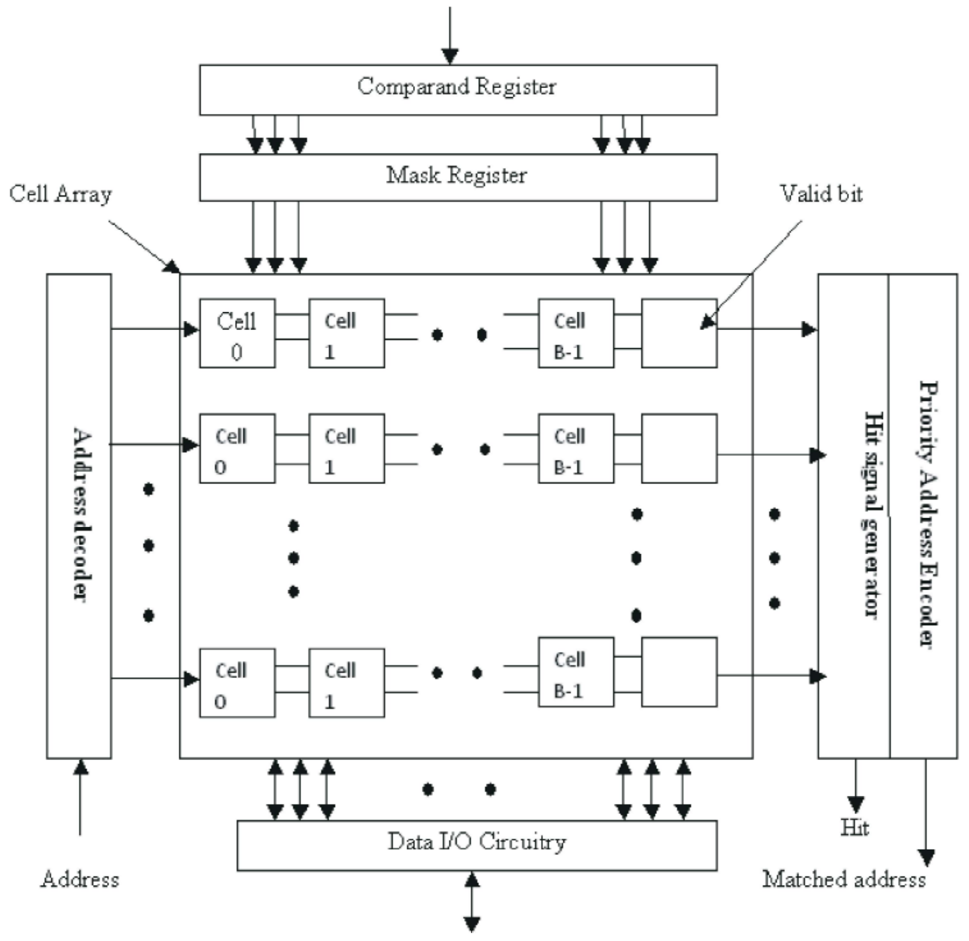

Fig. 1: Typical N×B bit CAM organization

of the comparison circuits in TCAMs with asymmetric cells. The test algorithm T_H requires 7N Write operations and (3N+2B) Compare operations to cover the comparison faults of an N*B-bit TCAM with Hit output only; and the test algorithm T_{PAE} requires 4N Write operations and (3N+2B) Compare operations to cover the comparison faults of an N*B -bit TCAM with priority address encoder (PAE) output. The Address Decoder and Data I/O are similar to those in a RAM. The cell array consists of 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 a Compare operation, the compared data (comparand) is prefetched to the Comparand Register and then the data in the Comparand Register is parallel compared with the data stored in all the words. The Hit Signal Generator (HSG) evaluates the valid match signals 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). The PAE and HSG may not coexist. For some applications, a TCAM only has a HSG to shorten critical-path delay for performing a Compare operation. The major difference between a BCAM and a TCAM is that the TCAM cell can represent three possible states “logic 0”, “logic 1”, or “don’t care (X)” states, but the BCAM cell only can represent two possible states “logic 0” or “logic 1”. The X state enables the 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. An asymmetric cell consists of a BCAM bit and a mask bit as shown in Fig 2. We focus on the testing of the TCAM with asymmetric cells. Therefore, we introduce the operations of an asymmetric cell in more details. When the TCAM executes a Compare operation, the match line (ML)

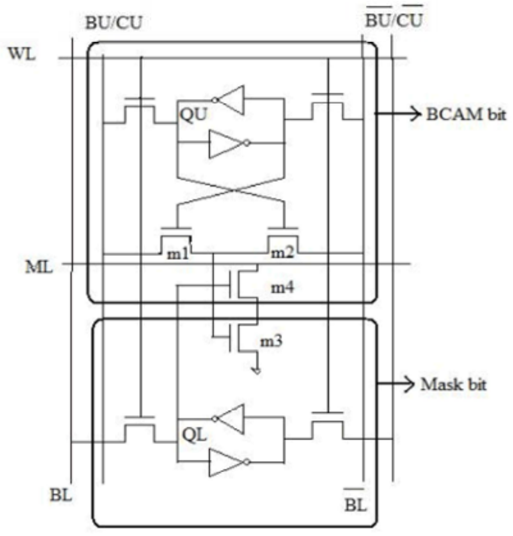

Fig. 2: Asymmetric TCAM cell

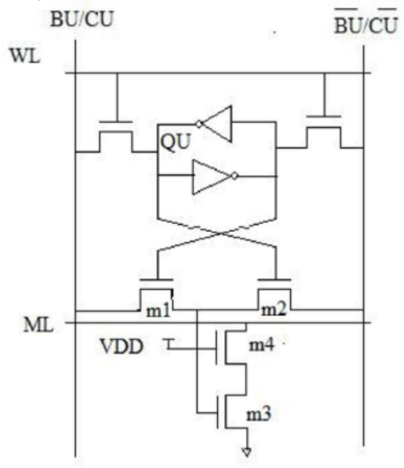

Fig. 3(a): Equivalent circuit of asymmetric TCAM cell when Q_L=1

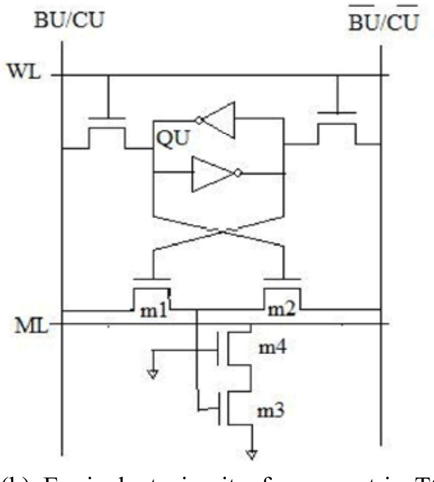

Fig. 3(b): Equivalent circuit of asymmetric TCAM cell when Q_L=0.

is precharged to V_{DD} in the pre charge 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 3(a) 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. As Fig 3(a) On the contrary, if $Q_L = 1$, then the ML value is determined by the comparison result of the upper BCAM bit. We assume that when ($Q_U = 0$, $Q_L = 1$) then the value is 0, ($Q_U = 1$, $Q_L = 1$) then the value is 1, ($Q_U = 1$, $Q_L = 0$) then the value is X and ($Q_U = 0$, $Q_L = 0$) then the value is X, where 0,1 and X denote the ternary data.

The Compare operation fails if the CAM cell stores a value x and is compared with the complementary input value $\bar{x}$.

Functional Comparison Faults: If 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 a 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. However, defects that cause the transistor m4 to be stuck-on cannot 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. Fig. 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. The wX operation also forces the BCAM bit to store logic 0, i.e., Q_U=0. Subsequently, a c0 should be executed to set the gate of m3 to logic 1, since the Compare operation sets (C_U,$\bar{C}_U$) to (0, 1). Therefore, a c0 after 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 the testing of the comparison faults of the BCAM bit when the mask bit stores logic 1 and the testing of the transistor m4 stuck-on fault when the mask bit stores logic 0.

Experimental Results

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

TE 1: ↕ (w1);

TE 2: ↕(wX,cP_0,w0,cP_0,w1);

TE 3:($cP_{X...X0}$,$cP_{X...0X,...,}$$cP_{0X...X}$) ;

TE 4: ↕(w0);

TE 5: ↕(w1,cP_1,w0);

TE 6: ($cP_{X...X1}$,$cP_{X...1X,...,}$$cP_{1X...X}$) ;

T_H 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(↑), descending (↓), or either way (↕). Each Compare operation within the third 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 third or sixth test element individually has 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 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 w1 represents a Write operation with data 1111 and $cP_{X...X0}$ denotes the Compare operation with the comparand XXX0.

According to Table 1, 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 undergo (w0,c0),(w0,c1),(w1,c0) and (w1,c1) and the corresponding TCAM cell response should be observed by the TCAM output. Also, if a TCAM cell undergoes (wX,c0) operations and the corresponding response can be observed by TCAM output, then the faults cause the m4 to be stuck-on can be detected. Therefore, we should show that T_H can guarantee that every TCAM cell can undergo the five Compare-after-Write operations, (w0,c0), (w0,c1), (w1,c0) and (w1,c1) and (wX,c0) the corresponding cell response can be observed by the TCAM output. We use a 3×3-bit TCAM as an example to explain the T_H. Also, the test elements executed in the ascending address sequence are assumed. In T_H, the TE1 initializes the TCAM array to the all-1 state as shown in Fig. 4.

Table 1: BCAM cell response to compare-after-write operation

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

Fig. 5 shows the fault-free status of the TCAM when the test element TE2 is executed. The first column of Fig.5 shows the status of the TCAM when the W0 is addressed. When the cP_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 output of Hit becomes 0.Therefore, every bit of W_0 undergoes a (wX,c0) operation and its faulty response can be observed by Hit output. Subsequently, the w0 operation writes all-0 data into W_0. When the second cP_0 of TE2 is executed, the fault-free Hit output should be 1, since the state of W_0 is the same as the comparand of cP_0. But, if there is any faulty bit in and its faulty response is mismatch, then the output of Hit becomes 0. Thus, every bit of W_0 undergoes a (w0,c0) 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 (wX,c0) and (w0,c0) operations and the corresponding fault effect, Mismatch, can be observed at Hit output when the TE2 is completed. After the execution of TE2, the fault-free status of the TCAM array will be all-1 state.

Subsequently, the TE3 is executed. Fig. 6 shows the fault-free status of the TCAM when TE3 is executed. As Fig.6 shows, the TE3 executes three Compare operations. The first Compare operation compares the comparand XX0 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

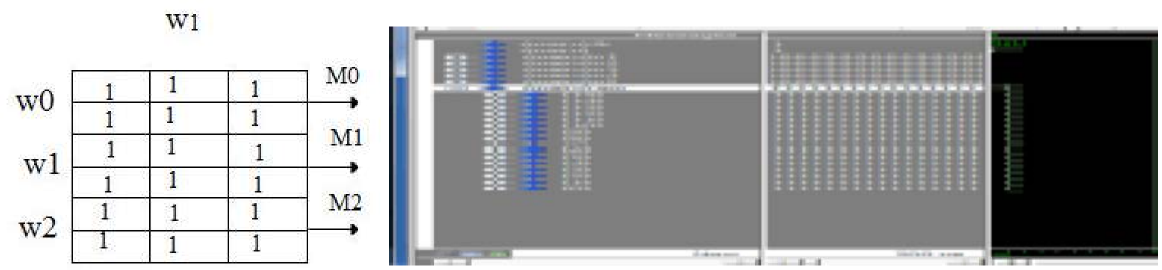

Fig. 4: Fault-free status of a 3×3-bit TCAM when TE1 is executed.

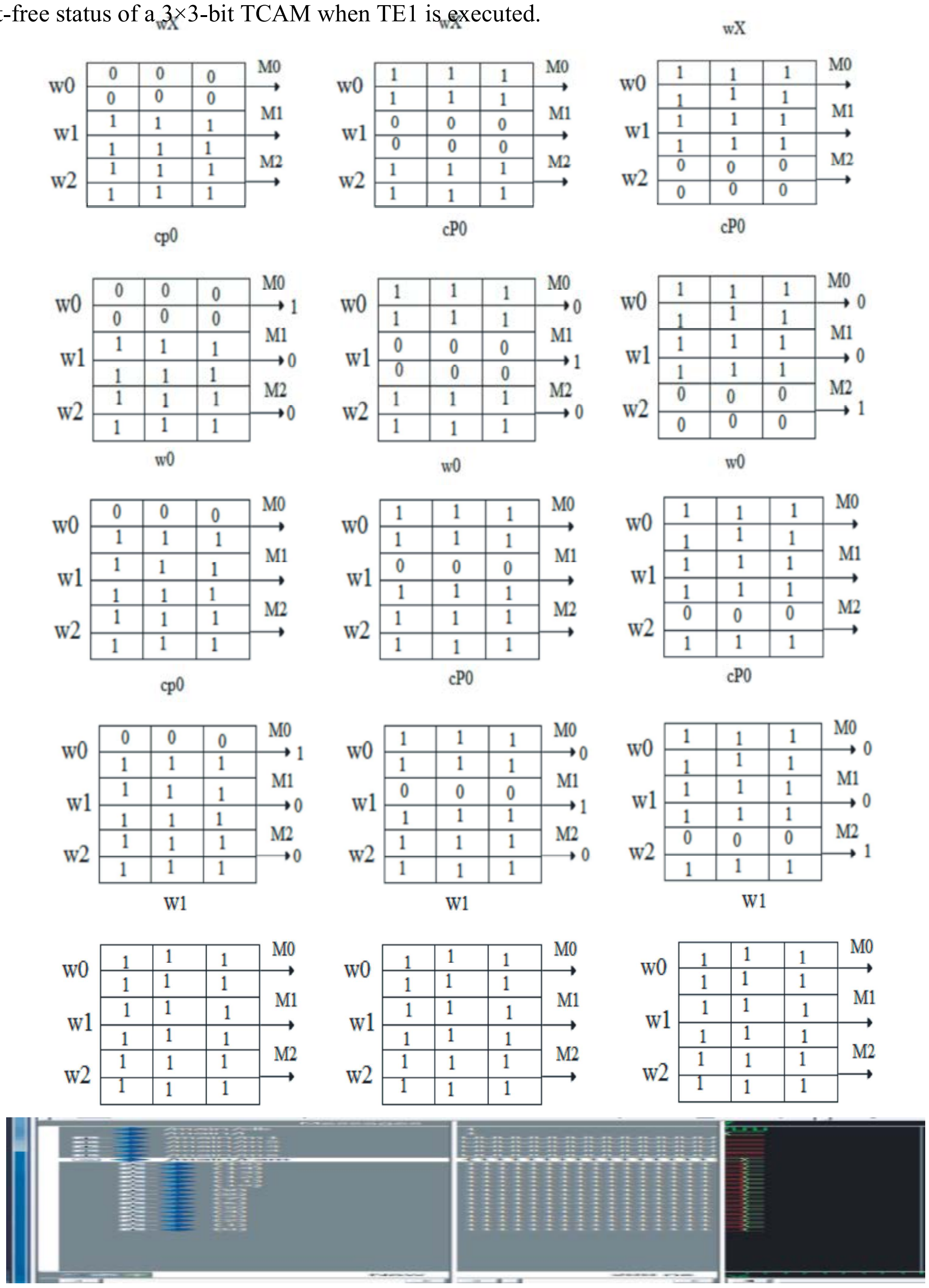

Fig. 5: Fault-free status of a 3×3-bit TCAM when TE2 is executed.

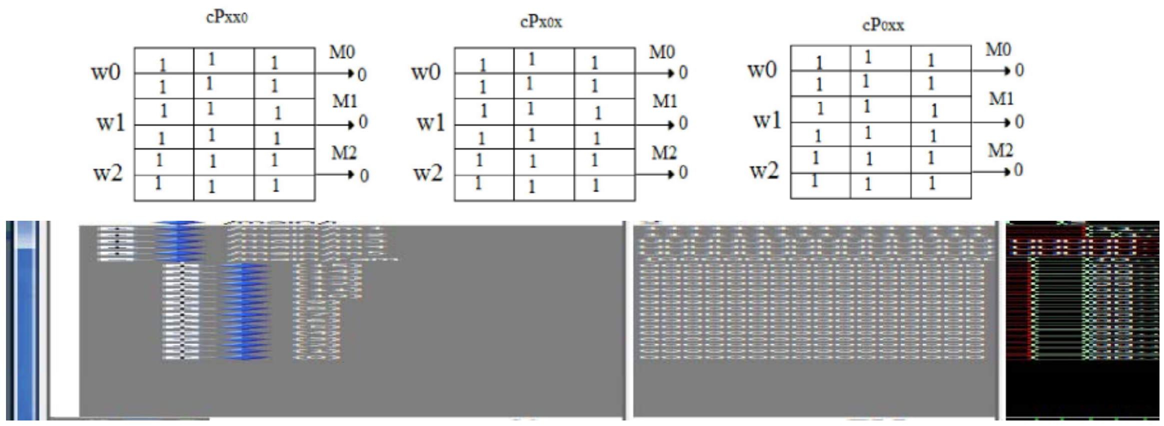

Fig. 6: Fault-free status of a 3×3-bit TCAM when TE3 is executed.

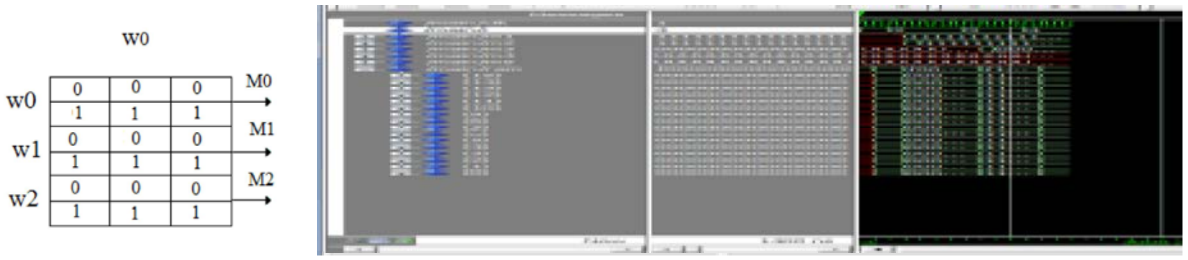

Fig. 7: Fault-free status of a 3×3-bit TCAM when TE4 is executed.

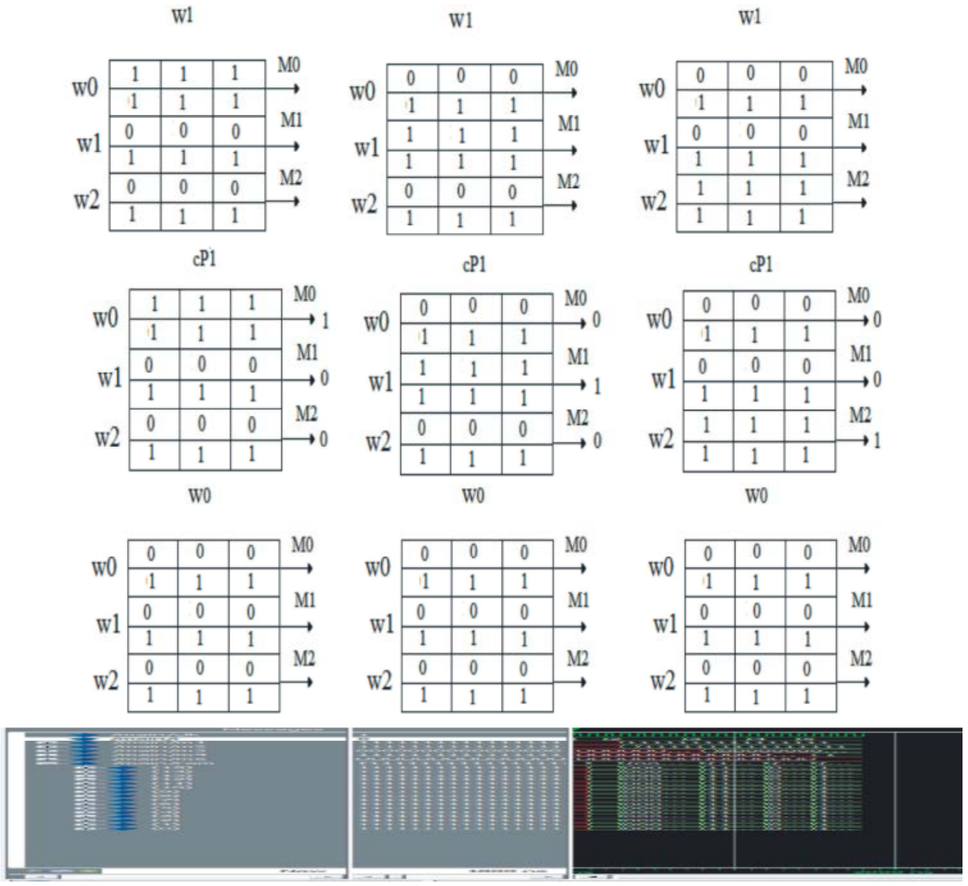

Fig. 8: Fault-free status of a 3×3-bit TCAM when TE5 is executed.

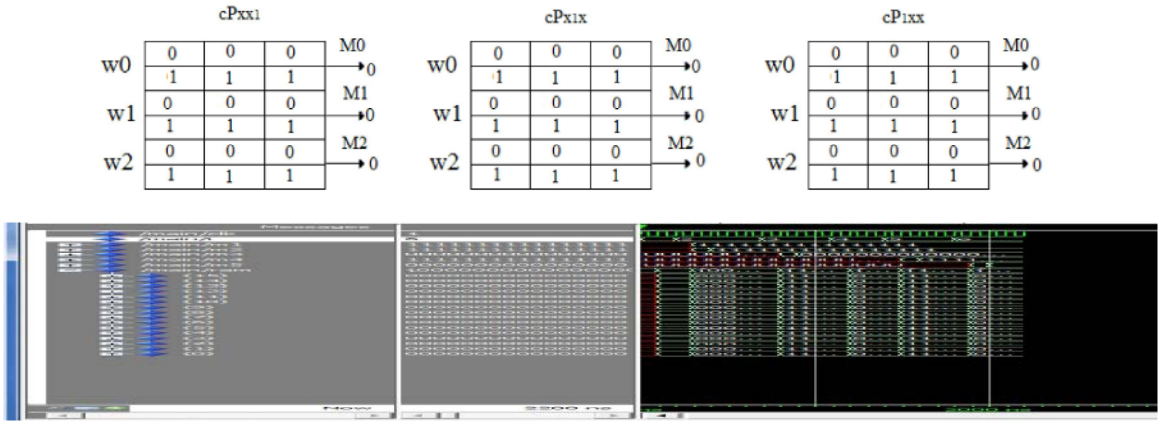

Fig. 9: Fault-free status of a 3×3-bit TCAM when TE6 is executed.

Table 2: Compare-after-write operations corresponding to the detection test elements of T_H

Operation	Fail response	Detection test element
wX/c0	MM	TE2
w0/c0	MM	TE2
w1/c0	M	TE3
w1/c1	MM	TE5
w0/c1	M	TE6

Table 3: Fault dictionary of the test algorithm T_H

	E_0	E_1	E_2	E_3	E_4
SMF	0	0	1	0	1
SMMF	0	1	0	1	0
CM1F	0	1	0	0	1
CM0F	0	0	1	1	0
PM1F	1	1	1	0	0
PM0F	0	0	0	1	1
EMM1F	0	0	0	1	0
EMM0F	0	1	0	0	0
IM1F	0	0	1	0	0
IM0F	0	0	0	0	1
Stuck on fault	1	0	0	0	0

is faulty and its faulty response is Match, then the Hit=1. The second and third compare operations compare X0X, 0XX with all the words and produces the Hit=0 when it is fault-free.

In T_H, the TE4 initializes the TCAM array to the all-0 state.

The TE5 performs the two write operations and one compare operation. The first write operation writes 1 in the addressed word. Then the words are compared with the comparand 1. If it is fault-free then the value of the Hit signal is 1. If there is any defect the faulty output 0 is produced. Again the 0 is written into the word.

As Fig. 9 shows, the TE6 executes three Compare operations. The first Compare operation compares the comparand XX1 with all the words. Because the first two bits of the comparand are Xs, only the 1 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. Similarly the second and third compare operations compare X1X, 1XX with all the words in TCAM array.

The table 2 shows the faulty response of the compare-after-write operations and it is detected by the corresponding detection test elements. The table 3 shows the fault dictionary of T_H algorithm where E_i represents the ith compare operation. The value 1 in the table shows that there is a faulty response in the corresponding compare operation.

CONCLUSION

In this paper, we have presented 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 test algorithm T_H uses 7N Write operations and (3N+2B) Compare operations to detect 100% targeted comparison faults of an -bit TCAM with Hit output only.

REFERENCES

1. Akhbarizadeh, M.J., M. Nourani and C.D. Cantrell, 2005. Prefix segregation scheme for a TCAM-based IP forwarding engine, IEEE Micro., 25(4): 48-63.
2. Al-Assadi, W.K., A.P. Jayasumana and Y.K. Malaiya, 1994. On fault modelling and testing of content-addressable memories, in Proc. IEEE Int. Workshop Memory Technol., Des., Test. (MTDT), pp: 78-81.

3. Al-Assadi, W.K., A.P. Jayasumana and Y.K. Malaiya, 1994. On fault modelling and testing of content-addressable memories, in Proc. IEEE Int. Workshop Memory Technol., Des., Test, pp: 78-81.
4. Kang, Y.S., J.C. Lee and S. Kang, 1997. Parallel BIST architecture for CAMs', Electron. Lett., 33(1) :30-31.
5. Lee, K.J., C. Kim, S. Kim, U.R. Cho and H.G. Byun, 2004. Modeling and testing of faults in TCAMs, in Proc. Asian Simulation Conf., Jeju Island, pp: 521-528.
6. Li, J.F., 2005. Testing comparison faults of ternary CAMs based on comparison faults of binary CAMs, in Proc. Asia South Pac. Des. Autom. Conf. (ASP-DAC), Shanghai, pp: 65-70.
7. Li, J.F., 2005. Testing priority address encoder faults in content addressable Memories',in Proc. Int. Test Conf. (ITC), Austin, TX, pp: 1-8.
8. Li, J.F., 2007. Testing ternary content addressable memories using march-like tests, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 26(5): 919-931.
9. Li, J.F., K.L. Cheng, C.T. Huang and C.W. Wu, 2001. March-based RAM diagnosis algorithms for stuck-at and coupling faults, in Proc.Int. Test Conf. (ITC), Baltimore, MD, pp: 758-767.
10. Lin, K.J. and C.W. Wu, 2000. Testing content-addressable memories using functional fault models and March-like algorithms, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 19(5): 577-588.
11. Mazumder, P., J.H. Patel and W.K. Fuchs, 1988. Methodologies for testing embedded content addressable memories, IEEE Trans Comput.-Aided Des. Integr. Circuits Syst., 7(1): 11-20.
12. Ravikumar, V., R.N. Mahapatra and L.N. Bhuyan, 2005. EaseCAM: an energy and storage efficient TCAM-based router architecture for IP lookup, IEEE Trans. Computers, 54(5): 521-533.
13. Wright, D. and M. Sachdev, 2003. Transistor-level fault analysis and test algorithm development for ternary dynamic content addressable memories, in Proc. Int. Test Conf. (ITC), pp: 39-47.
14. Wright, D. and M. Sachdev, 2003. Transistor-level fault analysis and test algorithm development for ternary dynamic content addressable memories, in Proc. Int. Test Conf. (ITC), pp: 39-47.
15. Zhao, J., S. Irrinki, M. Puri and F. Lombardi, 2000. Testing SRAM-based content addressable memories, IEEE Trans. Computers, 49(10): 1054-1063.