Design Techniques and Test Methodology for Low-Power TCAMs

Nitin Mohan, *Student Member, IEEE*, Wilson Fung, *Member, IEEE*, Derek Wright, *Student Member, IEEE*, and Manoj Sachdev, *Senior Member, IEEE*

***Abstract*—Ternary content addressable memories (TCAMs) are gaining importance in high-speed lookup-intensive applications. However, the high cost and power consumption are limiting their popularity and versatility. TCAM testing is also time consuming due to the complex integration of logic and memory. In this paper, we present a comprehensive review of the design techniques for low-power TCAMs. We also propose a novel test methodology for various TCAM components. The proposed test algorithms show significant improvement over the existing algorithms both in test complexity and fault coverage.**

***Index Terms*—Associative memories, content addressable memory (CAM), low power, priority encoder (PE), testing.**

I. Introduction

CONTENT addressable memory (CAM) is an outgrowth of random access memory (RAM) technology. Unlike RAMs which access a word based on its address, CAMs access a word based on its contents. A CAM compares an incoming key with all the words in parallel and returns the address of the best match. Historically, CAMs have been attractive for artificial-intelligence (AI) applications and translation look-aside buffers (TLBs) in microprocessors. CAMs are also used for tag comparison in cache memory, data compression, and radar signal tracking. Recent applications include real-time pattern matching in virus-detection and intrusion-detection systems, gene pattern searching in bioinformatics, and image processing.

CAMs can perform fast and deterministic pattern searches for large databases. A binary CAM stores and searches only "0"s and "1"s. Hence, its utility is limited to exact-match SEARCH operations. A ternary CAM (TCAM) can store and search an additional state, called "mask" or "don't care." Therefore, a TCAM can also perform partial matching. This partial-match feature makes TCAMs attractive for applications such as packet forwarding and classification in network routers. Increasing line rates, quality of service (QoS), and network security requirements demand routing tables with high-speed lookups. Moreover, an increasing number of Internet users and the introduction of IPv6 are further increasing the size of routing tables. Hence, current network routers require large-capacity TCAMs with high search speeds.

Despite the attractive features of TCAMs, high power consumption and manufacturing costs are the most critical challenges faced by TCAM designers. The parallel nature of TCAMs leads to high-power consumption. For example, an 18–Mb TCAM running at 250 million searches per second (MSPS) consumes 15 W [1]. The high-power consumption increases junction temperature, which increases leakage currents, reduces chip performance, and degrades reliability. The high cost of existing TCAM chips is mainly due to limited storage capacity per chip, which is caused by large cell area and high-power consumption. In addition, a complex integration of memory and logic makes TCAM testing very time consuming. These issues drive the need of innovative design techniques and efficient test algorithms for manufacturing large-capacity and cost-effective TCAMs.

Many low-power techniques have been proposed for TCAMs. However, the published literature is largely fragmented. Most of the existing publications address only some of the design issues. There is a growing need for a comprehensive study on TCAM design. Similarly, the existing TCAM test methods have limited fault coverage due to the lack of defect-oriented algorithms. In this paper, we present a comparative study of various design techniques for low-power TCAMs. In addition, we propose a defect-oriented test methodology for TCAMs and compare it with the existing TCAM test algorithms. The remainder of the paper is organized as follows. Section II presents an overview of TCAM organization and cell design techniques. Section III analyzes various low-power TCAM circuits. Section IV focuses on priority-encoder (PE) design techniques. Section V proposes a comprehensive test methodology and compares it with the existing TCAM test algorithms. Section VI reviews the methods to implement redundancy in TCAMs. Finally, Section VII concludes the paper with key observations and recommendations.

II. TCAM Organization and Cell Design

A typical TCAM chip consists of three major parts: 1) TCAM arrays for ternary data storage; 2) peripheral circuitry for READ, WRITE, and SEARCH operations; and 3) test and repair circuitry for functional verification and yield improvement. The peripheral circuits include decoders, bit line sense amplifiers (BLSAs), search line (SL) drivers, match line sense amplifiers (MLSAs), and PEs. The test and repair circuitry

Manuscript received December 21, 2004; revised January 22, 2006. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and Micronet R&D.

N. Mohan and M. Sachdev are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: nitinm@ieee.org; msachdev@ece.uwaterloo.ca).

W. Fung is with the ATI Technologies Inc., Markham, ON L3T 7X6, Canada (e-mail: wifung@ati.com).

D. Wright is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: derek.wright@utoronto.ca).

Digital Object Identifier 10.1109/TVLSI.2006.878206

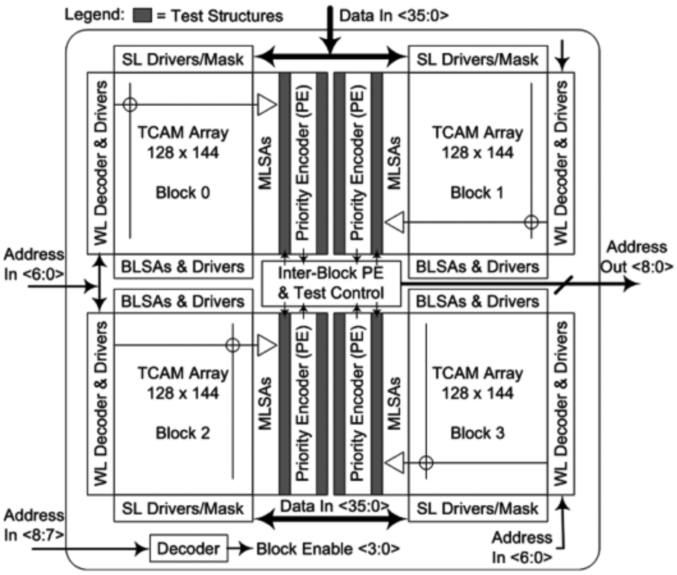


Fig. 1. Simplified block diagram of a 512 × 144 TCAM.

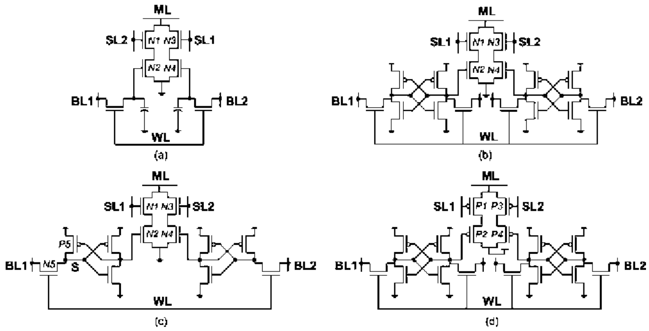


Fig. 2. (a) 6T dynamic TCAM cell. (b) Conventional 16T static TCAM cell. (c) 12T static TCAM cell. (d) Balanced 16T static TCAM cell.

TABLE I
AREAS OF DIFFERENT TCAM CELLS (SHOWN IN FIG. 2)

Fig. 2	TCAM Cell	Area (μm^2)	Process Technology
(a)	6T dynamic TCAM	3.59	0.13μm [2]
(b)	Conventional 16T static TCAM	6.73	0.13μm [5]
(c)	12T static TCAM	17.54	0.18μm[a] [3]
(d)	Balanced 16T static TCAM	7.08	0.13μm[a] [4]

[a] standard design rules

includes on-chip test structures and redundancy. Fig. 1 shows a simplified block diagram of a 512 × 144 TCAM. The TCAM is implemented as four smaller TCAM arrays. Each row in a TCAM array stores a word. Within a word, a bit is located by its column number. All the TCAM cells in a row share a word line (WL) and a match line (ML). Similarly, all the TCAM cells in a column share bit lines (BLs) and SLs. Partial matching in TCAMs may result in multiple matches. PEs are used to determine the highest priority match. Conventionally, a word with a lower address is given a higher priority. PEs also generate a signal which indicates the presence or absence of multiple matches. Typically, the highest priority match from a TCAM is encoded ("Address Out" in Fig. 1) to access the corresponding memory location in an off-chip RAM. A high-density TCAM chip also employs test and repair circuitry for identifying the faulty components and replacing them with their redundant counterparts.

As mentioned earlier, a large-capacity TCAM chip is expensive due to the large cell area. A smaller TCAM cell significantly improves the storage capacity and reduces the cost of a TCAM chip. Each TCAM cell consists of two RAM cells and a comparison logic circuit. Fig. 2 illustrates some dynamic and static TCAM cells. The 6T dynamic cell [Fig. 2(a)] is relatively smaller but it requires a specialized embedded DRAM process [2]. The static cells are more attractive due to their compatibility with the standard logic process. A 12T static TCAM cell [Fig. 2(c)] is advantageous in terms of smaller cell area [3]. It maintains a "0" state at node "S" by satisfying the following two conditions: 1) BLs are discharged to ground and 2) the N5 leakage is higher than the P5 leakage. The second condition is fulfilled under all the process variations by keeping WLs at a nonzero voltage $(V_{WL} \approx 200)$ mV [3]. This condition increases the BL leakages by 2–3 orders of magnitude. Therefore, this cell is not appropriate for low-power TCAMs. Moreover, this cell is not suitable for the READ operation, which is required for chip verification. Fig. 2(d) shows a balanced 16T static TCAM cell [4]. The layout of this cell is more compact than that of the conventional 16T cell [Fig. 2(b)] because it has an equal number of PMOS and NMOS transistors. Table I compares the areas of different TCAM cells shown in Fig. 2. It should be noted that the balanced 16T static TCAM cell has been laid out using standard-logic design rules. As a result, the reported area of this cell is slightly higher than that of the conventional cell.

In order to minimize the TCAM cell area, the transistors and interconnects must be laid out at a minimum distance defined by the design rules. Such a dense layout leads to high inter-wire capacitance. The parasitic capacitances of BLs and WLs are not critical because READ or WRITE operations are performed only during the table updates, maintenance, and testing. During the SEARCH operation, most of the power is consumed in switching SLs and MLs. Hence, their parasitic capacitances must be minimized. The inter-wire capacitances of SLs and MLs are reduced by placing them equally apart from the other parallel lines. Further reductions in the line capacitances are achieved by minimizing the wire widths of SLs and MLs. However, the lines must be wide enough to avoid problems such as electromigration and poor-signal integrity under the worst case operating conditions.

III. LOW-POWER TCAM CIRCUITS

In most applications, TCAM activity is dominated by the parallel SEARCH operation, which is expensive in terms of power consumption. The main peripheral circuits that perform the SEARCH operation are MLSAs and SL drivers. As a consequence, most TCAM design techniques focus on these circuits. Increasing static power consumption is also becoming a serious issue for large-capacity TCAMs. Circuit and architecture innovations are needed to limit the increasing static power in TCAMs.

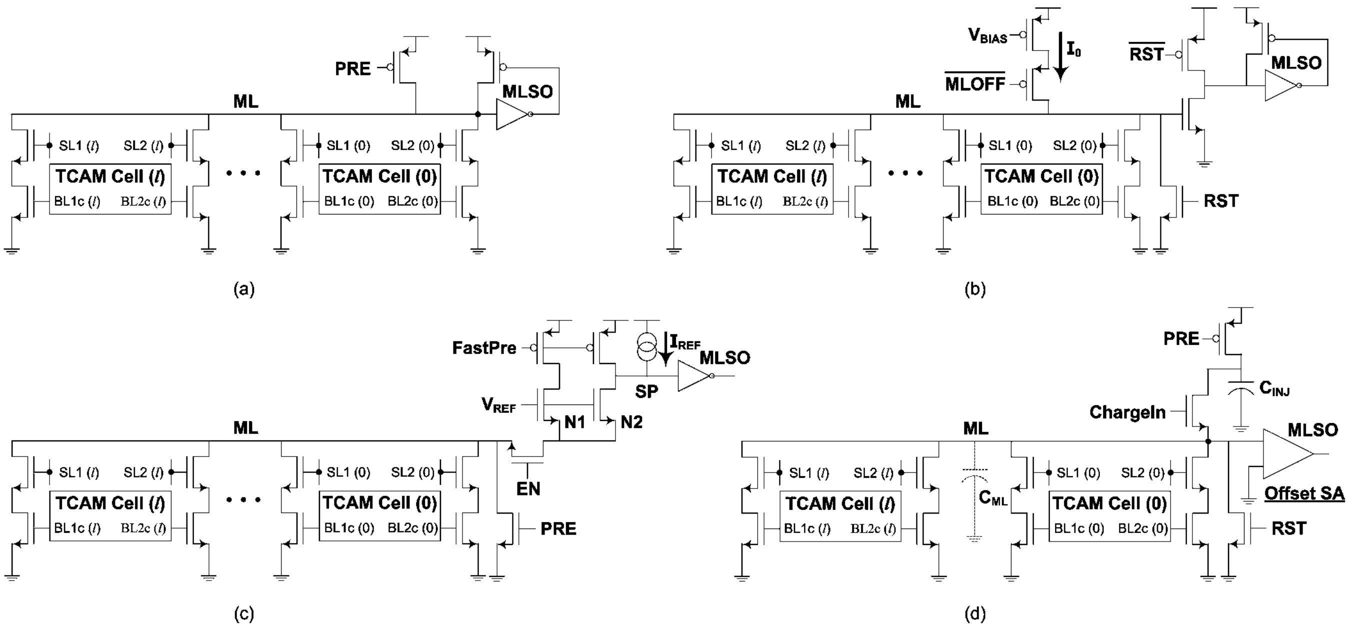


Fig. 3. ML sense amplifiers: (a) conventional precharge; (b) current-race; (c) charge-redistribution; (d) charge-injection.

A. MLSA

Most low-power MLSAs strive to minimize the ML voltage swing. Fig. 3(a) illustrates the conventional MLSA. Initially, all the MLs are precharged to $V_{\rm DD}$, and the search key is written on the SLs. If a TCAM word is identical to the search key, the ML remains at $V_{\rm DD}$. Otherwise, it discharges to ground through mismatching cells. In order to avoid short-circuit current, the SLs are switched to ground during the precharge phase. Hence, most of the SLs switch in every SEARCH operation, causing high-power consumption. Fig. 3(b) shows a current-race sensing scheme [3]. This scheme has the MLs at the ground voltage during the precharge phase, so the SLs can remain at their previous values. It reduces the average SL switching activity by half. This scheme achieves further power reduction by lowering the ML voltage swing. The ML sensing is initiated by charging up the MLs using constant current sources. The matching MLs charge at a faster rate than the mismatching MLs. When a matching ML charges to the NMOS threshold voltage (V_t), its MLSO changes from “0” to “1” [Fig. 3(b)]. A dummy ML emulating the “match” condition generates an MLOFF signal to end the ML sensing. Fig. 3(c) shows another MLSA that reduces ML voltage swing using charge redistribution [6]. This scheme also has MLs at the ground voltage during the precharge phase. The ML sensing begins with fast precharging of MLs using a FastPre signal. Transistors N1 and N2 restrict the ML voltage swing to $(V_{\rm REF} - V_t)$. After the FastPre pulse, the MLs are left floating. For the “mismatch” condition, the ML voltage drops below $(V_{\rm REF} - V_t)$ and the transistors N1 and N2 turn on. The transistor N2 equalizes the voltages of nodes ML and SP by redistributing charge at the two nodes [Fig. 3(c)]. A small current source $(I_{\rm REF})$ feeds the SP node to compensate for ML leakages. The voltage $V_{\rm REF}$ can be varied to tradeoff power consumption with speed of operation. This method can reduce the ML voltage swing even below V_t. However, the fast precharging of mismatching MLs causes short-circuit power dissipation. A charge-injection match detection circuit (CIMDC) eliminates this short-circuit power [Fig. 3(d)] [5]. CIMDC uses an injection capacitor $(C_{\rm INJ})$ for each ML. Typically, $C_{\rm INJ}$ is sized 3–4 times smaller than $C_{\rm ML}$[5]. Initially, all the injection capacitors are precharged to $V_{\rm DD}$ and all the MLs are discharged to ground. At evaluation, charge is injected from $C_{\rm INJ}$ to $C_{\rm ML}$ using ChargeIn signal [Fig. 3(d)]. For “match” condition, the voltage of $C_{\rm ML}$ rises to a voltage determined by the ratio of $C_{\rm INJ}$ and $C_{\rm ML}$. For “mismatch” condition, ML is discharged to ground. An offset sense amplifier differentiates between the “match” and “mismatch” conditions. Although the charge-injection scheme reduces the ML swing to very small voltages, it suffers from a lower noise margin and an area penalty due to $C_{\rm INJ}$.

Fig. 4 shows the delay and energy of the above ML sensing schemes for different word sizes when they are simulated in 0.18–μm CMOS technology. Global masking (GM) also alters the delay and energy by changing the ML capacitance. The ML capacitance can be given by (1)

$$C_{\rm ML} = [2g + 4(l - g)]C_{\rm DRAIN} + C_{\rm INT} \qquad (1)$$

where g is the number of globally masked bits, l is the total number of bits per word, $C_{\rm DRAIN}$ is the drain capacitance of each transistor in the comparison logic, and $C_{\rm INT}$ is the interconnect capacitance of each ML. When a bit is globally masked $(SL1 = SL2 = \text{“0”})$, only the drain capacitances of transistors N1 and N3 (shown in Fig. 2) contribute to $C_{\rm ML}$. Otherwise, $C_{\rm ML}$ also includes the capacitance of the internal nodes. Therefore, the worst case $C_{\rm ML}$ corresponds to no global masking $(g = 0)$, and the best case $C_{\rm ML}$ relates to full global masking $(g = l)$. Fig. 4(a) shows the energies of operation for both extremes. The search speed in Fig. 4(b) corresponds to the worst case. The precharge (or reset) duration is the same (1 ns) for fair comparison. We used $C_{\rm INT} = 0.18\ {\rm fF/cell}$ from the post-layout extraction of TCAM layout with MLs routed in metal 4 (0.18–μm CMOS process). Also $C_{\rm INJ}$ is sized to one-third of $C_{\rm ML}$. Fig. 4

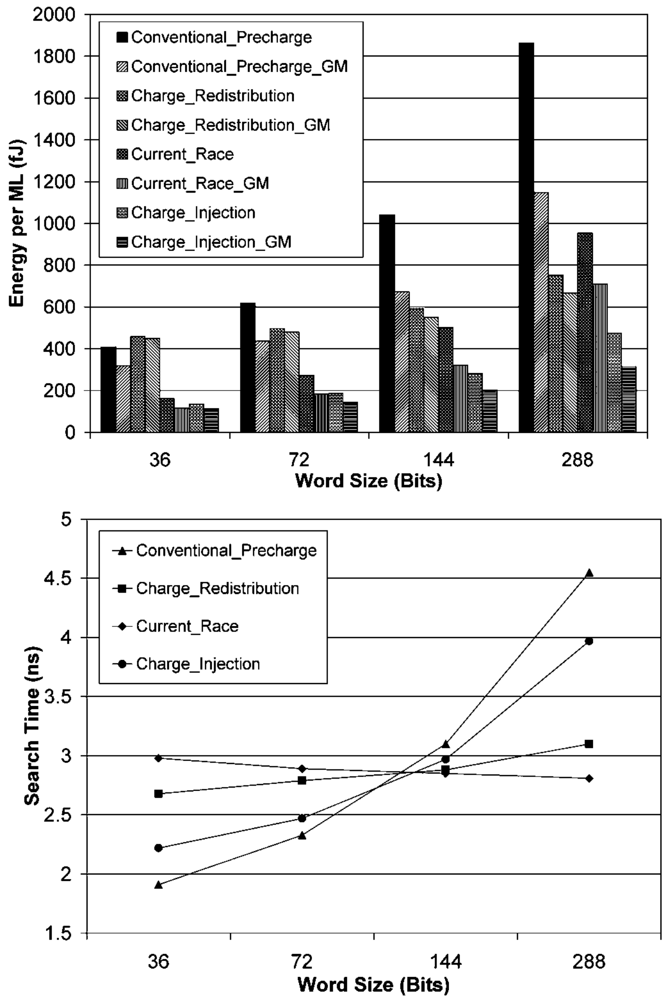


Fig. 4. Energy of operation per ML and search time for various MLSAs.

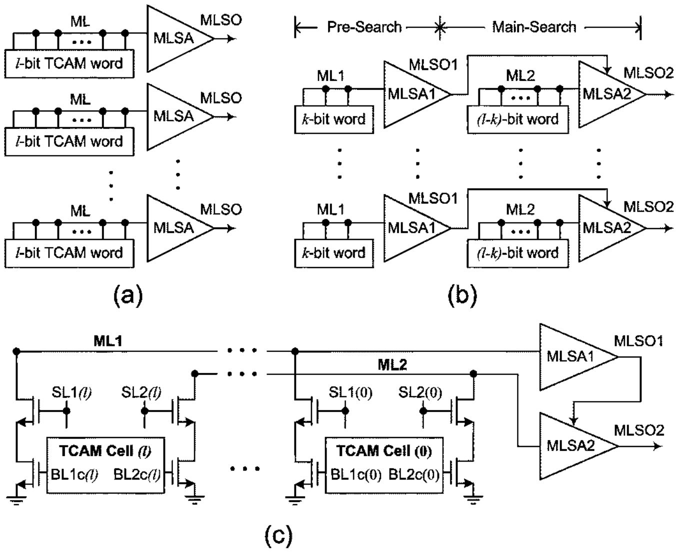


Fig. 5. (a) Conventional TCAM. (b) Selective-precharge TCAM. (c) Dual-ML TCAM.

shows that ML sensing energy and search time increase with word size due to increasing C_{ML}. The search speed remains almost constant for the current-race sensing scheme because current sources are also scaled with word size. Similarly, the search speed of the charge-redistribution scheme is also constant because speed is governed by the capacitance of node SP, which does not change with word size [Fig. 3(c)]. Fig. 4(a) affirms that the charge-injection scheme is the most energy-efficient technique for the given range of word sizes. However, a low noise margin and a large area penalty (due to C_{INJ}) make this scheme less attractive for high-density TCAMs. C_{INJ} can be implemented using a smaller size dummy ML to track process and temperature variations in regular MLs. The area penalty of C_{INJ} can be reduced by implementing it using a small array of comparison logic circuits.

The energies of operation of the remaining schemes increase with word size almost linearly but with different slopes. Therefore, the selection of optimal scheme depends on the word size. For example, the current-race scheme is more energy efficient for small word sizes, while the charge-redistribution scheme is better for large word sizes. In addition, the energy of operation for the charge-redistribution scheme is more predictable because it is less sensitive to global masking.

It should be noted that (1) overemphasizes the impact of the drain capacitance on C_{ML}. In reality, C_{ML} also depends on the layout of the comparison logic. For example, C_{ML} can be reduced by merging the drains of transistors N1 and N3 (shown in Fig. 2). The capacitance of the internal nodes (N1–N2 and N3–N4 in Fig. 2) can be reduced by removing their drain contacts since these nodes are not connected to any wire. Therefore, efficient layout can make the C_{ML} less sensitive to the global masking.

B. ML-Segmentation Techniques

In the previous section, it was assumed that all the bits of a word share the same ML. The power consumption of ML sensing can be reduced by segmenting MLs. One of the most popular ML-segmentation techniques is selective precharge [7]. Several variations of this scheme have been widely used in industry. A conventional TCAM performs a SEARCH operation in one step for all the bits [Fig. 5(a)]. The selective-precharge scheme divides the SEARCH operation into multiple stages. Fig. 5(b) illustrates this scheme for two stages: Pre-Search and Main-Search. The Pre-Search stage performs the SEARCH operation on the first segment (k-bit wide). If this results in "match," the Main-Search stage also performs the SEARCH operation on the second segment. This scheme can achieve significant power savings if the Pre-Search stage causes "mismatch" in most of the words. For small values of k, the energy consumed by Pre-Search stage is small. However, k should be large enough to cause "mismatch" in most of the words. The optimal value of k for minimum average energy depends on the statistics of incoming data. For example, a selective-precharge TCAM designed for networking applications with $l = 144$ and $k = 36$ can save up to 75% of the ML dynamic power, where l is the total number of bits per word.

We recently proposed a dual-ML TCAM that eliminates such dependency and achieves power savings irrespective of the incoming data statistics [8]. The dual-ML TCAM employs two wires (ML1 and ML2) connecting to the left and right sides of

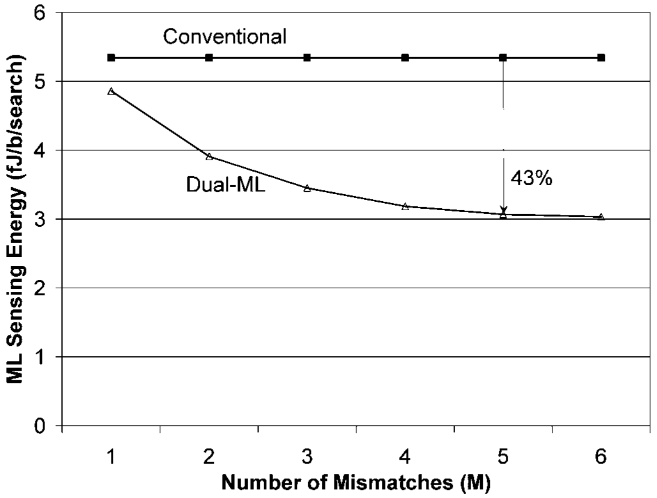


Fig. 6. Average ML sensing energy of conventional- and dual-ML TCAMs.

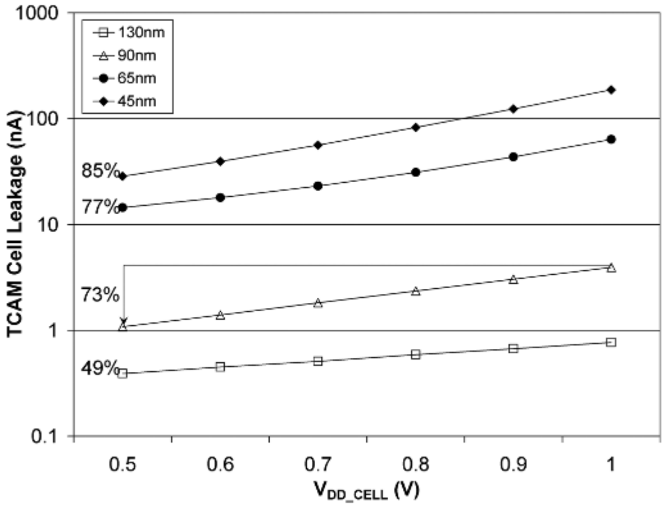


Fig. 7. TCAM cell leakage for different technology nodes at different values of $V_{\mathrm{DD\_CELL}}$.

the comparison logic, respectively [Fig. 5(c)]. Both ML1 and ML2 have separate sense amplifiers (MLSA1 and MLSA2). First, MLSA1 is enabled. If MLSA1 detects "mismatch," it does not enable MLSA2, and saves power. This scheme assumes: 1) most of the words in a TCAM array have multiple mismatches and 2) the probability of MLSA1 detecting "mismatch" increases with the number of mismatches.

We simulated the conventional and dual-ML TCAMs for 144-bit words in 0.18-μm CMOS technology using current-race MLSA. Fig. 6 shows the average ML sensing energy of conventional and dual-ML TCAMs for different number of mismatches. For five or more mismatches, this scheme results in a 43% energy reduction at the expense of a small tradeoff in speed (4%) [8]. In the dual-ML TCAM, both ML1 and ML2 are connected to every bit of a word. Thus, it is not as data dependent as the selective-precharge TCAM. In the selective-precharge TCAM, MLSO1 lines run over the Main-Search TCAM array to enable MLSA2 circuits [Fig. 5(b)]. The parasitic capacitance, due to these lines, increases the search delay and power. The dual-ML TCAM eliminates this additional parasitic capacitance by placing both MLSA1 and MLSA2 on the same side of the TCAM array [Fig. 5(c)]. Therefore, if the incoming data statistics are unpredictable, the dual-ML TCAM can achieve better power savings than the selective-precharge scheme. The sequential SEARCH operation of ML-segmentation results in larger search time. However, the speed penalty is not significant for large-size segments since the charging (or discharging) time of a highly capacitive ML is much larger than the propagation delay of the MLSAs.

C. SL Drivers

The SL switching activity depends on the incoming data statistics. For random data, half of the SLs switch in every SEARCH operation. Significant power savings can be achieved by reducing the voltage swing of SLs. It can be shown from Fig. 2 that the SL voltage swing strongly affects the $I_{\mathrm{ON}}/I_{\mathrm{OFF}}$ ratio of the ML pull-down paths. Therefore, most TCAM designs do not reduce the SL voltage swing. A recently published scheme breaks the SLs into global and local SLs (GSLs and LSLs) [9]. The $I_{\mathrm{ON}}/I_{\mathrm{OFF}}$ ratio of the ML pull-down paths is maintained by having rail-to-rail voltage swing (1.8 V) at LSLs. The power consumption is reduced by having a smaller voltage swing (0.45 V) at GSLs [9]. This scheme reduces the SL power consumption by 60%. However, the power reduction comes at the expense of area overhead due to wide OR-gates (64-input), low-swing receivers, and other control circuitry. This scheme can be implemented only if the MLs are divided into multiple segments and the incoming data is searched sequentially. This constraint degrades the search speed. In addition, this scheme requires the control circuitry to be embedded in the TCAM array, which deteriorates the density of the TCAM array.

D. Static Power Reduction

Conventionally, TCAM power has been dominated by the dynamic power due to the parallel SEARCH operation. As a result, most low-power TCAMs focus on dynamic power reduction. However, technology scaling is reducing the dynamic power and increasing the transistor leakage. The dynamic power is further reduced by architectural-level innovations [5], [9]. Thus, static-power reduction is becoming increasingly important in TCAMs. We simulated the leakage current (subthreshold and gate) of a TCAM cell (including BL leakage) using predictive technology models [10]. Fig. 7 shows the TCAM cell leakage variation with cell supply voltage ($V_{\mathrm{DD\_CELL}}$) in various technologies. For 45-nm technology, TCAM cell leakage is 187 nA at 1 V. Thus, an 18-Mb TCAM in this technology will consume 3.4 W of static power at 1 V. We recently proposed a dual-V_{DD} technique that reduces TCAM static power without compromising the search speed [11]. In this scheme, SL drivers and MLSAs use a higher V_{DD}, and TCAM storage employs a lower $V_{\mathrm{DD\_CELL}}$. Simulation results show 85% reduction in TCAM leakage when $V_{\mathrm{DD\_CELL}}$ is reduced from 1 to 0.5 V (45-nm technology in Fig. 7). A lower $V_{\mathrm{DD\_CELL}}$ also reduces the ML pull-down current for the "mismatch" case. However, the sensing speed of the current-race scheme only depends on the "match" case. Hence, $V_{\mathrm{DD\_CELL}}$ reduction does not affect the ML sensing speed. Lower $V_{\mathrm{DD\_CELL}}$ may raise other issues such as reduced noise margin and soft-error immunity. Fortunately, these issues are less severe in TCAMs because the comparison logic increases their storage node capacitance.

E. Issues With Large-Capacity TCAMs

Modern applications require large-capacity TCAMs to store and search large databases. For example, the new version of Internet Protocol (IPv6) needs deep routing tables with a wide word size. Deep TCAM arrays can be implemented as multiple banks of smaller arrays on the same chip. Since the banks are activated in parallel, the speed penalty is minimal. The energy per SEARCH operation increases linearly with the TCAM depth. Implementing wide TCAM arrays is more challenging because the lower noise margin between “match” and “mismatch” degrades the reliability of ML sensing. This problem is getting worse with technology scaling due to increasing transistor leakages. The robustness of ML sensing can be improved by maximizing the $I_{\mathrm{ON}}/I_{\mathrm{OFF}}$ ratio of the pull-down paths. For example, if a technology offers multiple-V_t devices, the transistors with the highest $I_{\mathrm{ON}}/I_{\mathrm{OFF}}$ ratio should be used in the comparison logic. This ratio can be further improved using devices with non-minimum channel length. There is also a growing need for innovative MLSAs to achieve reliable operation even for a small $I_{\mathrm{ON}}/I_{\mathrm{OFF}}$ ratio of the ML pull-down paths.

IV. PE

TCAMs require wide-input PEs to resolve multiple matches at the MLSA outputs. Generally, PEs consist of two stages: 1) multiple match resolver (MMR) and 2) match address encoder (MAE).

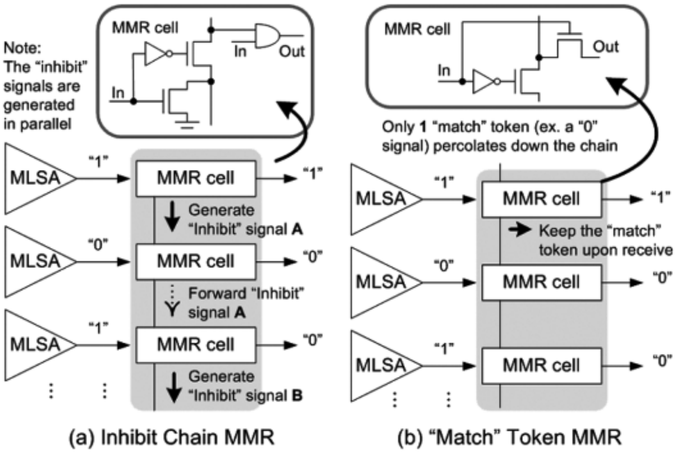
A. MMR

Similar to any other parallel operation, a TCAM lookup can lead to resource conflicts due to the possibility of multiple matches. Hence, the MLSA outputs must be post-processed to determine the best match in a search. For multiple match resolution, the most widely used approach is based on priority encoding. Each TCAM word is prioritized, and the priority is determined by its physical address. Conventionally, the lowest-address word has the highest priority. The application software stores data into the appropriate memory address so that the PE can accurately determine the best match in a TCAM lookup.

An MMR is an n-bit input, n-bit output datapath circuit. Following the active high convention (“match” = “1” and “mismatch” = “0”), an output bit is a “1” if: 1) the corresponding input bit is a “1” and 2) all the higher priority input bits are “0”s. The function of a PE can be described by the Boolean expressions in (2)

$$\begin{aligned}\mathrm{Out}_0 &= ML_0\\ \mathrm{Out}_1 &= ML_1 \cdot \overline{ML_0}\\ &\vdots\\ \mathrm{Out}_n &= ML_n \cdot \overline{ML_{n-1}} \cdots \overline{ML_1} \cdot \overline{ML_0}. \end{aligned} \tag{2}$$

Early works on MMRs are direct translations of the above expressions into CMOS circuits. These circuits are simple, but their layouts are highly irregular. It is also challenging to pitch match these large fan-in logic gates to TCAM array. Some recent studies have proposed domino logic based implementations of MMRs [12]. However, a domino-based MMR is not suitable for low-power TCAMs because of the power-hungry clock drivers. Block level power-reduction techniques such as clock gating, are not applicable here due to the parallel operation of the MMR.


MMR cell
Note:
The “inhibit” signals are generated in parallel
In
In
Out
MLSA
“1”
MMR cell
“1”
Generate “Inhibit” signal A
MLSA
“0”
MMR cell
“0”
Forward “Inhibit” signal A
MLSA
“1”
MMR cell
“0”
Generate “Inhibit” signal B
MMR cell
In
Out
Only 1 “match” token (ex. a “0” signal) percolates down the chain
MLSA
“1”
MMR cell
“1”
Keep the “match” token upon receive
MLSA
“0”
MMR cell
“0”
MLSA
“1”
MMR cell
“0”

(a) Inhibit Chain MMR (b) “Match” Token MMR

Fig. 8. MMR design using a common pass-transistor chain.

An alternative approach is to design the MMR using a common pass-transistor chain. Fig. 8 illustrates this approach using two MMR implementations. Fig. 8(a) shows an “inhibit chain” MMR. If an input bit is signaling a “match,” the MMR cell sets the corresponding output bit to “1” and generates an “inhibit” signal. This “inhibit” signal percolates down the pass-transistor chain to reset all the lower priority output bits to “0.” The output bit that survives until the end of the evaluation process represents the highest priority “match.” The worst case delay is the time to pass the inhibit signal from the highest priority word to the lowest priority word. This scheme is fast but also power hungry due to the high-switching activities at the internal and output nodes. Fig. 8(b) shows a “match token” MMR. Instead of broadcasting an “inhibit” signal for every “match” at the MMR inputs, this scheme has only one “match token” percolating down the chain. If a matched MMR cell (input bit = “1”) receives the “match token,” the cell sets its output to “1.” It also retains the outputs of lower priority bits at “0” by not passing the “match token.” This scheme is energy efficient and offers the same worst case delay. One drawback of this scheme is that all pass transistor switches must be settled before initiating the “match token.”

Some examples of “inhibit-chain” MMR cells are depicted in Fig. 9(a)–(c). The inhibit signal, either a “V_{DD}” or a “V_{SS}” signal, is generated by transistor H as shown in each diagram. Fig. 9(a) shows an 11-T cell which has active-low input and active-high output [13]. The MMR cells form a transmission-gate (TG) chain with one end tied to ground. During precharge, all MMR inputs are inactive (logic “1”). Hence, all the TGs along the chain are ON, and the intermediate nodes are discharged to “0.” At evaluation, if the input is signaling a “match” (input = “0”), the corresponding TG is turned OFF and the intermediate node is pulled to “1” through transistor “H” [Fig. 9(a)]. This

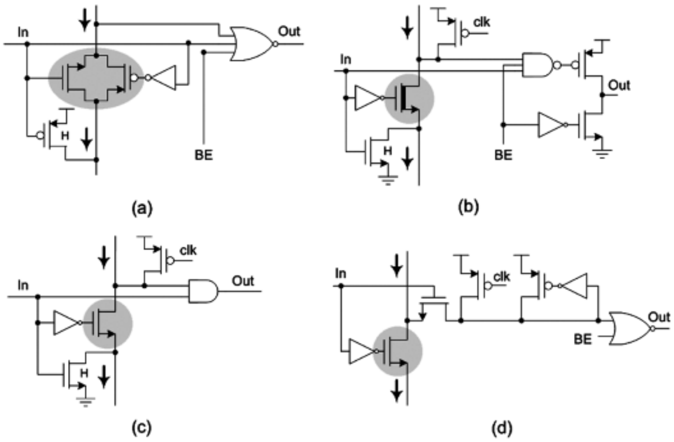


Fig. 9. MMR cells: (a) 11T cell; (b) 14T cell with low-V_t pass-transistors; (c) 9T cell; and (d) 12T cell.

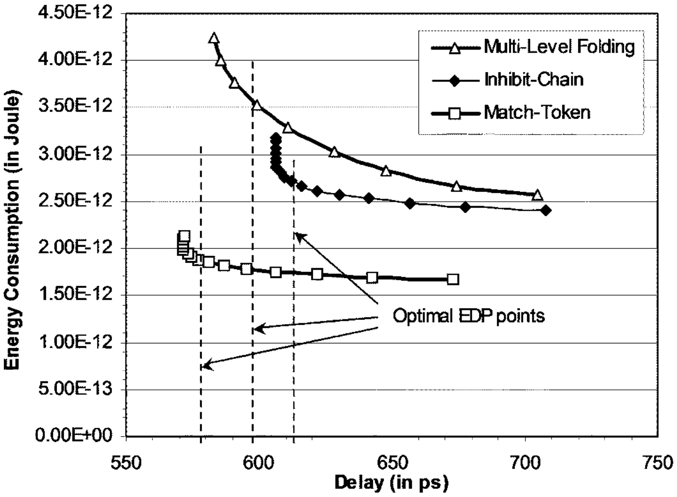


Fig. 10. Energy versus delay curve for various MMR schemes.

signal percolates down the TG chain and resets the outputs of the lower priority cells to "0." The block-enable (BE) signal is also active low. It is used to facilitate a multilevel MMR. For example, if there is a "match" in a higher priority block, the BE signal of the current block is held at an inactive state (logic "1"). The TG chain in this scheme offers good noise margin but it requires complementary enable signals. Fig. 9(b) shows a 14T-MMR cell with low-V_t pass-transistors [14]. This cell has active-high input and active-high output. Low-V_t transistors reduce the worst case delay, but their high leakage may cause a false discharge on the highest-priority bit. A large PMOS keeper can compensate for the pass-transistor leakage. However, this method is not reliable because the leakage of a low-V_t transistor is more sensitive to process variations. In addition, large keeper transistors also degrade the performance of the pass-transistor chain. Fig. 9(c) shows a 9T-MMR cell with an NMOS pass-transistor chain [15]. It also has active-high input and active-high output. During precharge, all the intermediate nodes are precharged to $V_{\rm DD}$ using a "clk" signal. For a "match" at the MMR inputs (input = "1"), transistor "H" turns on and resets the lower priority bits to "0." All three cells described above are based on the "inhibit-chain" concept. An example of a "match-token" based 12T-MMR cell is illustrated in Fig. 9(d) [16]. This cell is only a passive element and does not generate an inhibit signal. A "match-token" is percolated from the highest-priority bit to the lowest-priority bit. When an MMR cell with "match" (input = "1") receives the "token" (logic "0"), the output bit is switched to "1." Otherwise, the "token" is forwarded to the lower-priority bit. Hence, the highest-priority "match" keeps the "token," and the outputs of the lower-priority bits remain at logic "0."

Fig. 10 shows the energy versus delay curve for a 64-bit MMR implemented using various schemes in a CMOS 0.18-μm technology. Here, we are comparing the multi-level folding method described by Huang in [12], against the inhibit-chain method in Fig. 9(c) and the match-token method in Fig. 9(d). The circuit for each method is sized accordingly to achieve the optimal energy-delay product (EDP). As mentioned earlier in this section, the NP-domino method in [12] is power efficient only when completely ignoring the energy consumed by the clock buffers that drive the NP evaluation transistors. When taking the total energy into consideration, the method in [12] is nearly twice as power hungry when compared to the match-token method for the same worst case delay. Likewise, the inhibit-chain method is more energy-consuming than the match-token method. This is intuitive because the inhibit-chain, for passing the inhibit signal, is more likely to switch during evaluation, and a higher switching activity results in a higher EDP.

Several techniques have been studied for reducing the delay of an MMR. For instance, a wide-input MMR is usually segmented into smaller MMRs and organized in multiple hierarchical levels. This architecture facilitates multilevel "Priority Lookahead" (PLA), which is analogous to the "Carry Lookahead" concept in ALU design. The PLAs can greatly simplify the logic equations and reduce the delay. Huang *et al.* proposed a PLA scheme based on a paper-folding concept [12]. However, the design is impractical due to too many interconnect routings running in the vertical dimension if the circuit is laid out in a single column for interfacing with the MLSAs (Fig. 1).

Conventionally each TCAM block contains a local MMR. Alternatively, an MMR can be shared among several blocks, and switching circuitry can be built into each block for MMR arbitration [2]. While this offers area reduction and increases the effective TCAM density, it also introduces two major issues: 1) additional inter-wire capacitance on MLs and 2) injection of capacitive and inductive crosstalk from the MMR to MLs. The shared MMR scheme also demands an entire metal layer for routings over the TCAM cells. It is difficult to satisfy such constraint because a typical TCAM cell already requires routings over four or five metal layers (WL, 4 BLs, ML, 2 SLs, $V_{\rm SS}$, $V_{\rm DD}$).

B. MAEs

The highest-priority "match" signal can serve as an index to retrieve search results if there is an on-chip SRAM coupled to the TCAM. However, modern TCAMs usually omit such on-chip SRAM for two main reasons: 1) its absence offers a

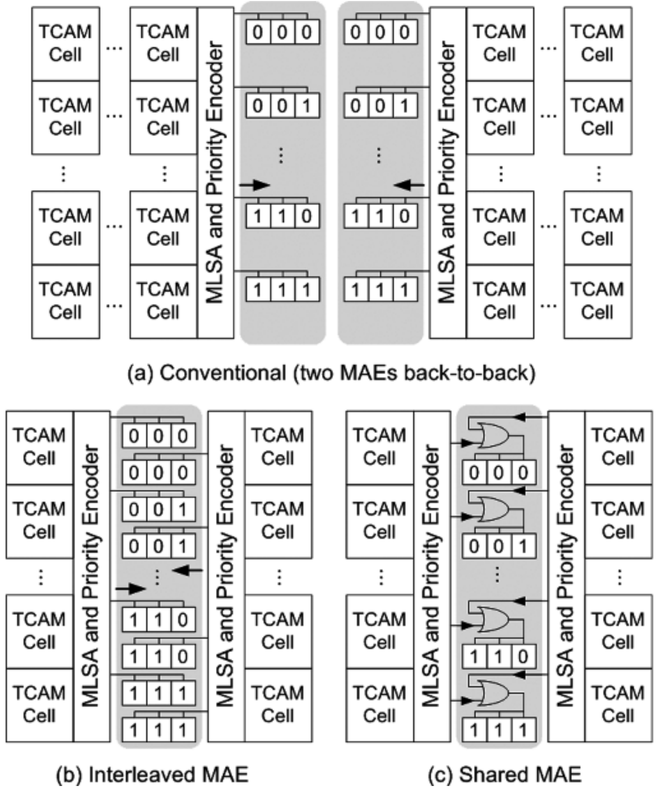

TCAM Cell
MLSA and Priority Encoder
0 0 0
0 0 1
1 1 0
1 1 1
(a) Conventional (two MAEs back-to-back)
(b) Interleaved MAE
(c) Shared MAE

Fig. 11. MAE.

higher effective TCAM capacity and 2) many lookup applications require a non-1-to-1 correspondence between TCAM and RAM. The associated data is typically stored in off-chip SRAMs, at a location specified by the TCAM "match" address encoded in binary form. This justifies the need of having MAEs in TCAMs.

Typically, the local address encoders are ROM-based structures. They are positioned back-to-back between two blocks of TCAM arrays as shown in Fig. 11(a). A ROM cell is composed of one or two transistors, and it is much smaller than a TCAM cell. Hence, pitch matching the ROM cells to the TCAM array may waste a lot of chip area. The area efficiency can be improved by merging two local MAEs as depicted in Fig. 11(b) [16]. However, this scheme increases the capacitance of BLs because they are shared by the ROM cells of both MAEs. Consequently, the BL sensing speed of this scheme is worse than that of the conventional scheme. In addition, this interleaved method can raise a conflict in the address encoder if there is a match in both arrays, but their local match addresses are different. To resolve such conflict, the MMR outputs can be registered with a clock gated by a block-level PE signal (Inter-Block PE in Fig. 1). Fig. 11(c) proposes a new scheme that allows both the TCAM blocks to share WLs using wired-OR gates. The MMR operation avoids the access conflict by ensuring that, at most, one bit of the MMR output can be at logic "1." The wired-OR gates can be placed into the unused spaces without any area penalty. This scheme can achieve a 40% reduction in MAE bit-line capacitance as compared to the interleaved WL approach.

The SEARCH operation typically generates only a few matched words over the entire TCAM chip. Therefore, the enabling clock can be gated by a "Block Hit" (BH) signal to avoid unnecessary turn-on of BLSAs in the MAE. For applications that generate multiple matches within a block, the MAE can be designed such that the WLs with lower physical addresses cause less switching on BLs (Fig. 11). This approach saves power since the MMR always favors the lower physical addresses.

V. TCAM Testing

In spite of the growing popularity of TCAMs, their test issues have not been addressed adequately. Most of the previous work on CAM testing is focused on binary CAMs [17], [18]. Algorithms developed for binary CAMs cannot be directly applied to TCAMs due to unique masking capabilities of TCAMs and the difference in their comparison logic circuits [17].

A. *TCAM-Cell Fault Analysis*

In order to develop a meaningful test algorithm, we performed a transistor-level fault analysis on a TCAM cell. Since RAM testing is a mature area of research, existing algorithms can provide adequate fault coverage for the RAM cells [19]. It has been observed that most defects in RAM cells result in stuck-at faults [20]. Therefore, we performed a fault analysis on the search-path transistors (N1–N4 in Fig. 2) assuming that the defects in RAM cells cause stuck-at faults (SA1 and SA0) in the storage nodes.

Due to symmetry of the cell, we performed the fault analysis on one half of the TCAM cell with results being equally valid for the other half [21]. Fault analysis results in five possible transistor-level faults: 1) source/drain contact defect; 2) gate contact defect; 3) gate to drain oxide failure; 4) gate to source oxide failure; and 5) subthreshold conduction. Table II describes these faults for one half of a TCAM cell (defects 1–12) and their detection methods. It also describes other possible inter-transistor faults (defects 13–19) and their detection methods. Table II assumes that the 6T-dynamic TCAM cell [Fig. 2(a)] is used. When a different cell is used, the detection methods can be altered accordingly. For example, an equivalent Table II for Fig. 2(b) will replace transistors N1 and N2 by N3 and N4, respectively, and *vice versa*. The last operation in each method in the column "Detection Method" refers to the result under correct operating conditions. The column labeled "Induced Fault" refers to the type of functional fault that a test algorithm would detect as a result of the specific defect and detection method. For example, defect 3 makes N2 appear to be stuck-open (SOP) since the source or drain contact has a defect. Similarly, defect 16 allows conduction through N3 and N2, making N4 appear to be stuck-on (SON) from a functional perspective.

Detection methods of defects 4 and 9, require a "wait" operation whose duration determines the resistance range of defects covered by these detection methods. For example, a longer "wait" can detect a larger resistance range of defects. Such a precisely controlled "wait" operation is not always feasible. Therefore, we developed high-level algorithms assuming that weak defects ultimately result in SON or SOP faults as shown in the last column of Table II.

TABLE II
POSSIBLE TCAM CELL FAULTS

#	Description	Detection Method	Induced Fault
1	Storage node SA0	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) search for mismatch	N2 SOP
2	Storage node SA1	(a) write “0”; (b) SL2 = “1”; SL1 = “0”; (c) search for match	N2 SON
3	N2 source/drain contact defect	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) search for mismatch	N2 SOP
4	N2 gate contact defect	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) wait; (d) write “0”; (e) search for match	N2 SON[a]
5	N2 gate-drain oxide failure	(a) write “0”; (b) SL2 = “1”; SL1 = “0”; (c) search for match	N2 SON
6	N2 gate-source oxide failure	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) search for mismatch	N2 SOP
7	N2 subthreshold conduction	(a) write “0”; (b) SL2 = “1”; SL1 = “0”; (c) search for match	N2 SON
8	N1 source/drain contact defect	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) search for mismatch	N1 SOP
9	N1 gate contact defect	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) wait; (d) SL2 = “0”; (e) search for match	N1 SON[a]
10	N1 gate to drain oxide failure	(a) SL2 = SL1 = “0”; (b) search for match	N1 SON
11	N1 gate-source oxide failure	(a) write “1”; (b) SL2 = “1”; SL1 = “0”; (c) search for mismatch	N1 SOP
12	N1 subthreshold conduction	(a) write “1”; (b) SL2 = “0”; SL1 = “1”; (c) search for match	N1 SON
13	N1 gate to N2 gate short	(a) write “0”; (b) SL2 = “1”; SL1 = “0”; (c) search for match	N1 SON
14	N1 source to N4 drain short	(a) write “1”; (b) SL2 = “0”; SL1 = “1”; (c) search for match	N1/N4 SON
15	N1 gate to N4 drain short	(a) write “1”; (b) SL2 = “0”, SL1 = “1”; (c) search for match	N4 SON
16	N2 gate to N4 drain short	(a) write “mask” (‘0 0’); (b) SL2 = “0”, SL1 = “1”; (c) search for match	N4 SON
17	N1 gate to N3 gate short	(a) write “1”; (b) SL2 = “0”, SL1 = “1”; (c) search for match	N1 SON
18	N1 gate to N4 gate short	(a) write “0”; (b) SL2 = “0”, SL1 = “1”; (c) search for mismatch	N4 SOP
19	N2 gate to N4 gate short	(a) write “1”; (b) SL2 = “0”, SL1 = “1”; (c) search for match	N4 SON

[a] requires a “wait” operation

B. DFT

The TCAM test complexity may be reduced significantly using design for testability (DFT). One may employ the “divide-and-conquer” approach to reduce the test complexity. First, the PE is tested using scan chains. Subsequently, the TCAM array is tested using the fault-free PE. Fig. 12 illustrates DFT structures to access and control different TCAM components individually and as a group. Multiplexers (A, B, C, and D) allow the inputs of MMR and MAE to switch between test vectors and the outputs of previous stages. Scan chains (SC1 and SC2) are used to feed-in the test vectors serially, and can also be used to scan-out the outputs of MLSAs and the MMR, respectively. Multiplexers B and D are used to bypass the scan-chains during normal TCAM operation.

C. Test Algorithms

Each TCAM cell contains a comparison logic circuit that discharges the ML under “mismatch.” Hence, conventional-TCAM test schemes have the complexity of $O(nl)$, where n is the number of words and l is the number of bits per word [22]. For an 18-Mb TCAM, this test complexity corresponds to O(18 M), which makes TCAM-testing a time consuming process. In addition, other TCAM components, such as the wide-input PEs, require extensive test algorithm development. Since the SEARCH operation proceeds from TCAM array to MMR to MAE, these components should be tested in reverse order.

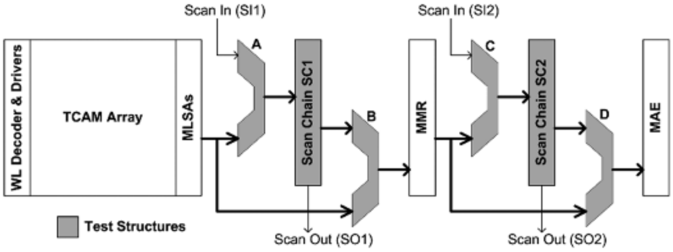


Fig. 12. TCAM DFT structures.

1) MAE Testing: The MAE is tested by encoding every possible address and examining validity of the output. The test is initiated by resetting the scan chain SC2 to all “0”s (Fig. 12). A “1” is shifted into SC2 followed by “0”s, and the MAE’s outputs are checked for the correct address. Thus, MAE testing requires n shifts.

2) MMR Testing: The block-level MMR (128-bit input) is implemented as a hierarchical tree structure of smaller MMRs. For example, each 128-bit MMR is implemented in two levels. The first level (L1), is made of sixteen 8-bit MMRs. The second level (L2), consists of a 16-bit MMR which resolves the inter-MMR priority conflicts of L1. A linear feedback shift register (LFSR) can be used to generate a pseudorandom binary sequence (PRBS) that includes all possible patterns of p bits (excluding the all-zeros pattern) in 2^p clock cycles [23]. Thus, the serial shifting of all p-bit patterns will require $(2^p + p)$ clock cycles. If the full block is tested exhaustively, $(2^{128} + 128)$ clock cycles are needed to test all the possible combinations. The test complexity is significantly reduced by testing the L1-MMRs (8 bit) in parallel (switch position $\mathbf{A}$ in Fig. 13). Since the L2-MMR is isolated during the L1 testing, it can be tested in parallel with L1-MMRs (switch position $\mathbf{E}$ in Fig. 13). L2 scan chains do not interface with MLSAs and MAE because they are internal to the MMR. Exhaustive testing of a 16-bit L2-MMR requires a large number $(\sim 64\text{ k})$ of clock cycles. Hence, the L2-MMR can be tested by resetting all its inputs to “0”s, and then shifting “1”s from its lowest-priority pin to its highest-priority pin. This method eliminates the time penalty by trading off test coverage. However, the test coverage is not sacrificed significantly because the L2-MMR is much smaller in total area than the L1-MMRs (almost one-eigth). Thus, the L2-MMR is less likely to have a defect. In addition, the inputs of the L2-MMR are physically further apart from one another (Fig. 13), and most commonly occurring defects will not be able to connect two inputs of the L2-MMR which are far apart. Thus, complex test patterns (with nonconsecutive active inputs) are not needed to test the L2-MMR, and a simple functional test is sufficient. If all the 8-bit MMRs are fault-free, they are reconnected in the tree structure (switch position $\mathbf{A'}$ in Fig. 13) for block-level testing. Initially, the scan chain is reset to “0,” and a string of “1”s are shifted.

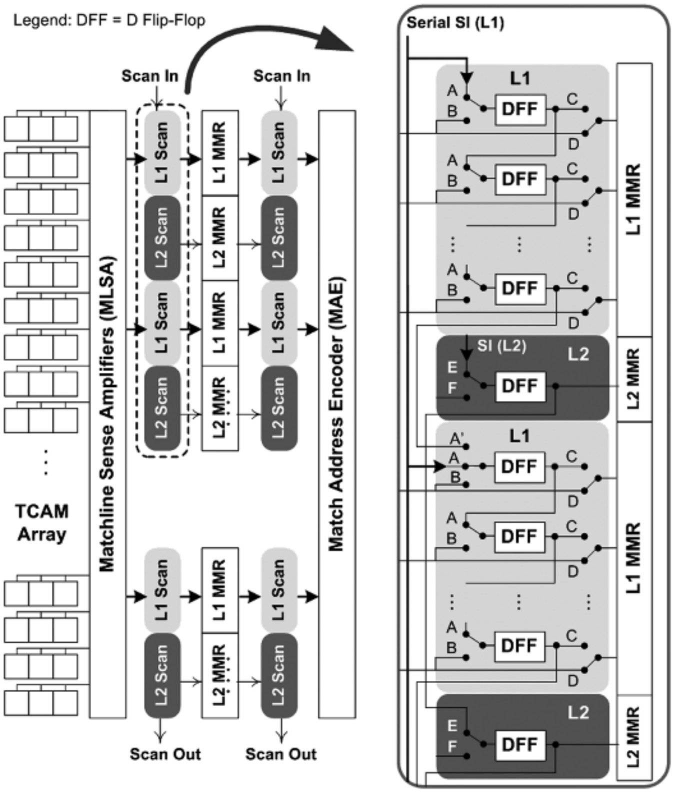


Fig. 13. Scan chains for MMR testing.

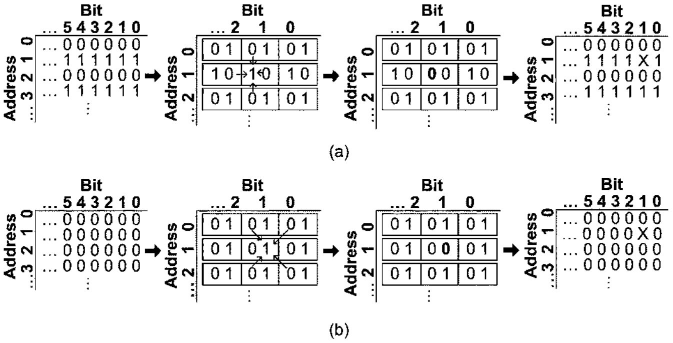


Fig. 14. Data patterns to stimulate: (a) horizontal/vertical and (b) diagonal inter-cell faults.

Recently, a PE test algorithm has been reported that uses the CAM array to test stuck-at faults in the PE [24]. Since it assumes a fault-free CAM array, it cannot be used with CAM test algorithms that require a PE. It also assumes that the n-bit PE is designed in one level. Thus, it does not exploit the multilevel hierarchical structure of the n-bit PE. It can be used in conjunction with our scheme (e.g., in L1-MMR testing) by inserting DFT structures, as shown in Figs. 12 and 13. For 8-bit L1-MMRs, it does not make much difference in the total test complexity. It can benefit designs with 16-bit or larger L1-MMRs but they are difficult to implement in pass transistor logic (as shown in Fig. 9).

3) TCAM Array Testing: As discussed in Section V-A, weak intra-cell defects ultimately result in SON or SOP faults. We developed a high-level algorithm to detect these faults with column level diagnostics. The proposed algorithm also detects horizontal, vertical and diagonal inter-cell coupling faults. Fig. 14(a) and (b) show data patterns to stimulate horizontal/vertical and diagonal faults, respectively. Fig. 14 also shows the bits in ternary format (“0” ≡ “0 1” and “1” ≡ “1 0”). As shown in Fig. 14, an inter-cell fault can change a TCAM cell’s value to “mask (“0 0”)” state. The remaining inter-cell faults can be stimulated by inverting these patterns. A coupling fault can also change a TCAM cell’s value to an invalid “1 1” state. It can be shown from Fig. 2 that under this condition, transistors N2 and N4 will conduct, and the affected word will always mismatch. However, this becomes a “0 0” fault under the inverse data conditions.

Table III illustrates different steps of the proposed test algorithm. In some steps, multiple words match with the search key. Thus, matching addresses are readout sequentially. Such address readouts are shown in the last column of Table III. SOP faults can unintentionally mask-out some of the ML pull-down paths, which causes erroneous “match” (steps 6 and 10). Similarly, SON faults can cause unintentional “mismatch” (steps 2 and 8). The words with SON faults in BL transistors (N2 and N4 in Fig. 2), are identified by searching “$XXX, \ldots, XXX$.” The column location of a BL-transistor SON fault is determined by masking the search key one bit at a time to avoid multiple “match” (step 4). Thus, step 4 requires l SEARCH operations in the worst case and an average of half SEARCH operations. Similarly, the column location of an SL-transistor SON fault is determined by masking the faulty word (step 5). Steps 3 through 5 are designed to avoid multiple “match” conditions, which lead to address readout operations. For example, the multiple “match” condition of step 2 is eliminated in step 3 by inverting the MSBs of the faulty word and the search key. The proposed scheme also schedule different steps so that the WRITE operations are minimized. For example, step 6 requires only $0.5n$ WRITE operations to fill “1”s in alternate rows. The remaining rows are already filled by “0”s in step 1. Although it is possible to write the same value in all the rows by enabling all the WLs simultaneously, we avoided this assumption due to the following two reasons. 1) If BL drivers remain the same, the parallel WRITE operation becomes extremely slow. Otherwise, BL drivers must be sized-up drastically to be able to write all the TCAM cells in a regular WRITE cycle. Such large BL drivers will be very power hungry and area consuming. 2) If all the WL drivers are turned on simultaneously, the transient *IR* and $L(di/dt)$ voltage drop in the $V_{\rm DD}$ and ground buses will slow down the WRITE operation. Otherwise, the widths of $V_{\rm DD}$ and ground buses of WL drivers must be increased by n, which will be very area consuming.

Our algorithm assumes at most one SON fault per word. If some words mismatch for both local and global masking, there are SON faults in both SL and BL transistors of these words. It is difficult to determine exact bit locations of such multiple faults. They can be replaced with redundant rows

4) Proposed Versus Existing TCAM Test Algorithms: A simple TCAM test algorithm individually tests each bit’s ability to match and mismatch for both “1”s and “0”s [22]. It proceeds in two steps: 1) test the ability for an address to match and 2) test each bit’s ability to mismatch. Match ability is tested by writing “000...000” to every address, and searching for “000...000” to verify that every address matches. This

TABLE III
TEST PROCEDURE AND COMPLEXITIES OF DIFFERENT STEPS OF THE PROPOSED ALGORITHM

#	Stored Data Pattern	Search Data Pattern	Result	WRITE	SEARCH	Address Readout
1	TCAM Array 000...000 000...000 000...000 000...000	Search Words 100...000 010...000 ... 000...010 000...001 Time	"Match" detects diagonal inter-cell faults.	n	l	-
2	No change	Search Words XXX...XXX 000...000 Time	"Mismatch" detects words with SON faults, and their type (in BL- or SL-transistors).	-	2	$2n$
3	Invert MSB of the faulty word (0 →1)	Invert MSB of the search word (0 →1)	Multiple "match" is eliminated.	1	-	-
4	No change	Search Words 100...00X 100...0X0 ... 1X0...000 X00...000 Time	"Match" detects the location of BL-transistor SON fault in the faulty word.	-	$0.5l$	-
5	Binary tree search at the faulty word. Mask (locally) half of the faulty bits. Repeat this operation within the segment that has SON fault.	Search 100 ... 000 repeatedly after every change in the masked bits of the faulty word.	(i) "Match" ⇨ SON fault is within the masked bits. (ii) "Mismatch" ⇨ SON fault is within the unmasked bits.	$\log_2(l)$	$\log_2(l)$	-
6	TCAM Array 000...000 111...111 000...000 111...111	Search Words 100...000 010...000 ... 000...010 000...001 011...111 101...111 ... 111...101 111...110 Time	(i) "Match" detects horizontal and vertical inter-cell faults. (ii) "Match" detects SOP faults in comparison logic of each TCAM cell with column resolution.	$0.5n$	$2l$	-
7	TCAM Array 111...111 111...111 111...111 111...111	Search Words 011...111 101...111 ... 111...101 111...110 Time	"Match" detects remaining diagonal inter-cell faults.	$0.5n$	l	-
8	No change	Search Words XXX...XXX 111...111 Time	"Mismatch" detects remaining words with SON faults, and their type (BL or SL).	-	2	$2n$
9	Repeat # 3 to 5 for inverted values	Repeat # 3 to 5 for inverted values	Detects the location of BL/SL-transistor SON fault within the faulty word.	$1 + \log_2(l)$	$0.5l + \log_2(l)$	
10	TCAM Array 111...111 000...000 111...111 000...000	Same as step 6	"Match" detects remaining horizontal and vertical inter-cell faults and SOP faults with column level resolution.	$0.5n$	$2l$	-
11	TCAM Array XXX...XXX XXX...XXX XXX...XXX XXX...XXX	Search Words 000...000 111...111 Time	"Mismatch" detects defect # 16 in Table 2.	n	2	$2n$
Total Complexity				$3.5n + N_{SON} \times [\log_2(l) + 1]$	$6l + N_{SON} \times [\log_2(l) + 0.5l] + 6$	$6n$

N_{SON} = Total number of SON faults

process is repeated using inverted values. Mismatch ability is tested by writing "000 . . . 001" to all the addresses, and then searching for "000 . . . 000" to ensure that no address matches. The SEARCH operation is repeated after shifting the pattern by one bit ("000 . . . 010") and writing it to all the addresses. The above process is also repeated using inverted values. Fig. 15(a) shows the simple test procedure and its complexity. The total complexity assumes equal time penalties for WRITE, SEARCH, shift, and address-readout operations.

Some new TCAM array test algorithms have recently been proposed by Lee [25] and Li/Lin [26]. Lee's algorithm is developed for a single cell and is subsequently expanded to a TCAM array [25]. This bottom-up approach is not optimized to exploit the parallel search capabilities of the TCAM. In addition, it does not provide column-level resolution for SON faults and does not verify global masking in SLs [25]. It assumes word-parallel write access, which may not be realistic in a large TCAM as explained in Section V-C-3. It proceeds in three identical steps as shown in Fig. 15(b). Assuming a word-parallel write access, each step requires $(10l+1)$ WRITE $+12l$ SEARCH operations $+10nl$ address readouts. The huge number of address readouts is caused by multiple "match" conditions in most SEARCH operations.

Li/Lin's algorithm detects a subset of faults covered by our algorithm. Their algorithm lacks test procedures for inter-cell

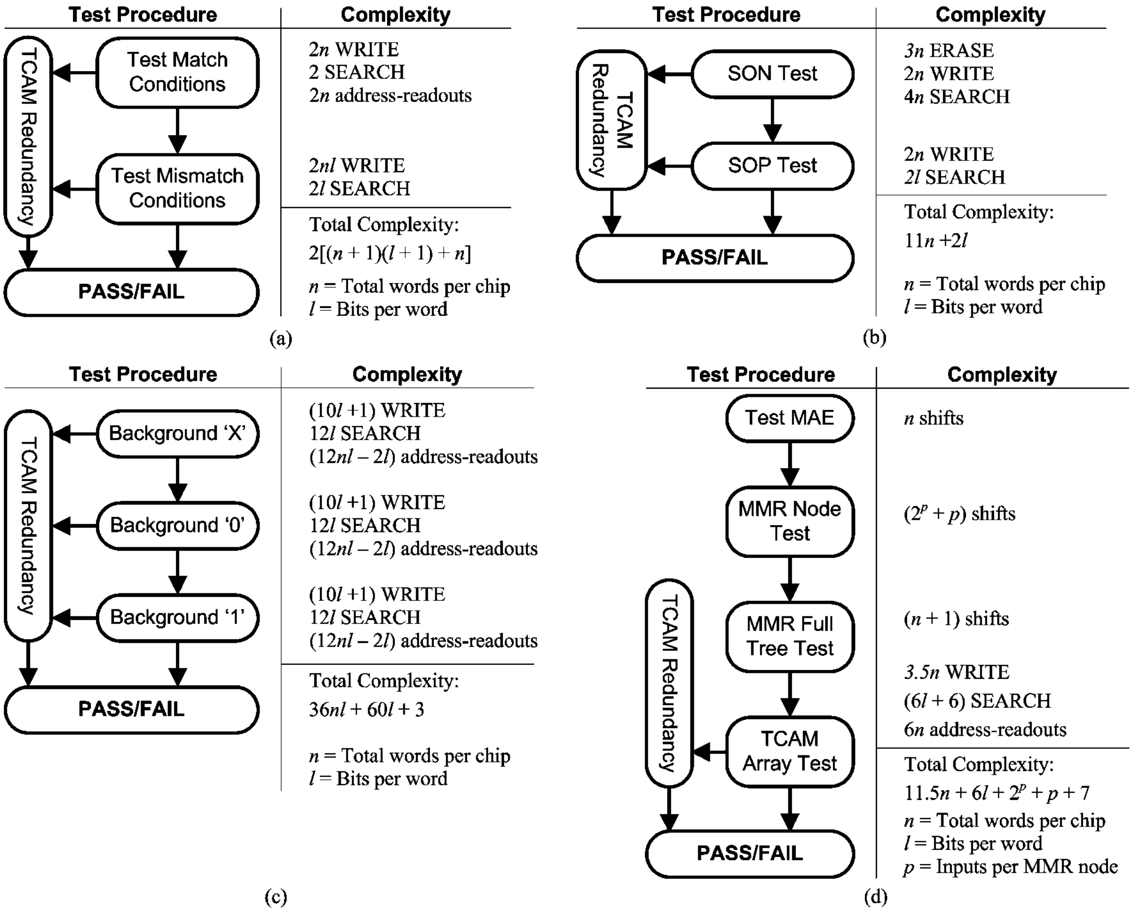


Fig. 15. Test procedures and complexities of: (a) simple [27]; (b) Li/Lin [31]; (c) Lee [30]; and (d) proposed algorithm.

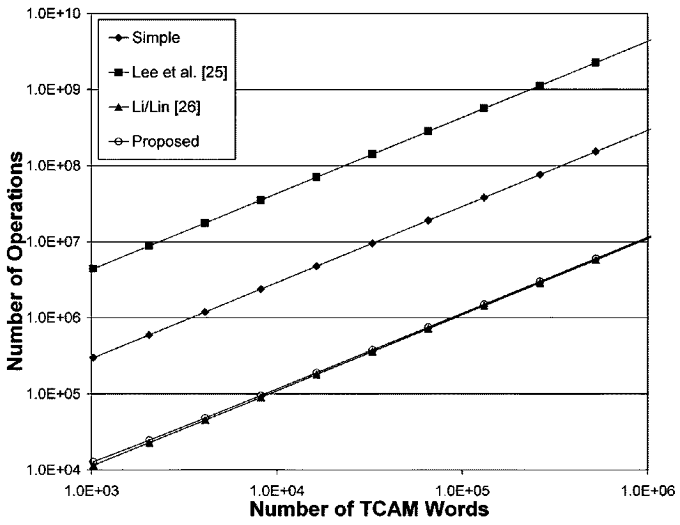


Fig. 16. Comparison of proposed and existing test algorithms.

fault detection and column-level diagnostics for SON faults [26]. Moreover, it does not verify if the "X" value can be properly stored and searched. Fig. 15(c) illustrates this test procedure along with its complexity. The ERASE operation requires an additional feature called valid bit, which determines if a word will participate in SEARCH operations.

Fig. 15(d) shows the complete test flow and complexity of the proposed algorithm. It assumes the availability of scan chains with reset. As shown in Table III, each SON fault increases the test complexity by $[2\log_2(l) + 0.5l + 1]$. For a typical TCAM $(l = 144)$, this results in 89 operations per SON fault, which is negligible as compared to the total test complexity. Thus, it is not included in complexity calculations in Fig. 15(d). Our algorithm achieves column-level diagnostics of SON faults, which is particularly useful if both row and column redundancy are employed.

Fig. 16 compares the complexities of the proposed TCAM test algorithm with the other three algorithms for different values of n. It assumes typical values of L1-MMR inputs and word size ($p = 8$, and $l = 144$). Although the proposed algorithm has almost the same complexity as Li/Lin's algorithm, it includes test procedures for MAE, MMR, inter-cell faults, and SON faults with column level diagnostics. It also outperforms other algorithms both in test complexity and fault coverage.

VI. TCAM Redundancy

Redundancy can be employed to improve the yield of TCAMs. If the test circuit detects faulty rows or columns, the defective elements are replaced by redundant elements, and the associated signals are rerouted. In RAMs, this can be easily accomplished by placing fuse links or multiplexers into the decoders. However, TCAMs require additional repair circuitry to preserve the logical address order for valid multiple-match resolution and address encoding. A good redundancy scheme must offer flexible repair at any location, and exhibit small-area overhead with little performance penalty.

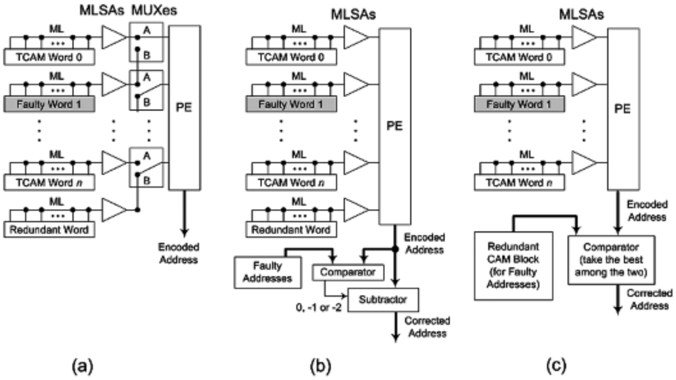


Fig. 17. TCAM row redundancy. (a) Shift redundancy. (b) Compare-then-subtract. (c) CAM-in-CAM.

Fig. 17 illustrates several row-redundancy schemes for TCAM. The simplest approach is called "shift redundancy" as shown in Fig. 17(a) [2]. This scheme skips the faulty row and shifts all lower-priority rows up the chain to remove the bubble. However, it can repair only one faulty row within an array unless multiple-input multiplexers are employed. Fig. 17(b) shows a "compare-then-subtract" scheme, which performs priority encoding by assuming no faulty word in the address space [14]. All faulty addresses are stored in off-array registers for comparisons against the encoded match address. If all faulty addresses are "larger" (having lower priority) than the match address, no subtraction is required. Otherwise, the match address is subtracted based upon the number of higher-priority faulty addresses. This scheme can repair more than one faulty row, but it also introduces additional circuit complexity which, in turn, reduces the effective TCAM density. Another alternative is to lump all the redundant rows into a redundant array [Fig. 17(c)]. This scheme introduces a small lookup table for faulty address translation inside the TCAM [27]. Unlike the regular TCAM array, a word in the redundant array is accessed by its "logical address," which is stored in the associated registers. A SEARCH operation is performed on both the regular array and the redundant array in parallel. If there is a "match" in both arrays, the higher priority "match" between the two is considered. This scheme saves area and improves repair flexibility at the expense of additional power.

The column-redundancy methods for TCAM are simpler than the row redundancy methods. Each cell in a redundant column is connected to the associated ML. If there is a defect in a regular column, this column is masked out by the global search lines, and the data is stored in a redundant column. The presence of this redundant column imposes additional capacitance on the ML, which in turn, reduces the TCAM search speed. Thus, only a small number of redundant columns can be added into each TCAM block.

VII. Conclusion

We have presented a comprehensive design and test methodology for low-power TCAMs. The existing TCAM design techniques accomplish power reduction by lowering the voltage swing of MLs. However, these schemes normally trade robustness and noise margin for reduced power consumption. In addition, some of the schemes are suitable for smaller word sizes and others are appropriate for larger word sizes. Therefore, a design technique should be carefully chosen based on the TCAM word size. Since the PE is in the critical path of the search operation, its design requires low-power techniques that do not degrade the performance. Moreover, PE-design techniques can exploit the facts that most of the words do not match in a SEARCH operation, and that the "match" in a higher-priority word is more likely to survive the MMR operation. We also presented a new TCAM test strategy which outperforms the existing algorithms on both fault coverage and test time. Testing of multilevel hierarchical MMRs poses tradeoffs between fault coverage and test time. Thus, the test strategy for the MMR should be chosen depending on the block size of the TCAM.

References

[1] K. Etzel, "Answering IPv6 Lookup Challenges" Cypress Semiconductor Corporation, San Jose, CA, 2004 [Online]. Available: http://www.cypress.com

[2] H. Noda, "A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 245–253, Jan. 2005.

[3] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.

[4] A. Roth, D. Foss, R. McKenzie, and D. Perry, "Advanced ternary CAM circuits on 0.13-μm logic process technology," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, 2004, pp. 465–468.

[5] G. Kasai, Y. Takarabe, K. Furumi, and M. Yoneda, "200 MHz/200 MSPS 3.2 W at 1.5 V VDD, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, 2003, pp. 387–390.

[6] P. Vlasenko and D. Perry, "Matchline sensing for content addressable memories," U.S. Patent 6 717 876, Apr. 6, 2004.

[7] C. Zukowski and S. Wang, "Use of selective precharge for low-power CAMs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 1997, pp. 745–770.

[8] N. Mohan and M. Sachdev, "Low-power dual matchline ternary content addressable memory," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2004, pp. 633–636.

[9] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, 2003, pp. 383–386.

[10] "Predictive Technology Model (PTM)" Arizona State Univ., Tempe, AZ [Online]. Available: http://www.eas.asu.edu/~ptm

[11] N. Mohan and M. Sachdev, "A static power reduction technique for ternary content addressable memories," in *Proc. IEEE Canadian Conf. Elect. Comput. Eng. (CCECE)*, 2004, pp. 711–714.

[12] C. H. Huang, J. S. Wang, and Y. C. Huang, "Design of high-performance cmos priority encoders and incrementer/decrementers using multiple lookahead and multilevel folding techniques," *IEEE J. Solid-State Circuits*, vol. 37, no. 1, pp. 63–76, Jan. 2002.

[13] H. Bergh, J. Eneland, and L.-E. Lundstrom, "A fault-tolerant associative memory with high-speed operation," *IEEE J. Solid-State Circuits*, vol. 25, no. 4, pp. 912–919, Aug. 1990.

[14] T. Miwa, H. Yamada, Y. Hirota, T. Satoh, and H. Hara, "A 1-Mb 2-Tr/b nonvolatile CAM based on flash memory technologies," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1601–1609, Nov. 1996.

[15] J. G. Delgado-Frias and J. Nyathi, "A high-performance encoder with priority lookahead," *IEEE Trans. Circuits Syst. I, Fundam. Theory and Appl.*, vol. 47, no. 9, pp. 1390–1393, Sep. 2000.

[16] R. Foss and A. Roth, "Priority encoder circuit and method for content addressable memory," Canadian Patent 2 365 891, Apr. 30, 2003.

[17] P. Sidorowicz, "Modeling and testing transistor faults in content-addressable memories," in *Int. Workshop Memory Technol. Des. Testing*, 1999, pp. 83–90.

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

[19] 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. Int. Test Conf. (ITC)*, 2001, pp. 758–767.

[20] R. Dekker, F. Beenker, and L. Thijssen, "Fault modeling and test algorithm development for static random access memories," in *Proc. Int. Test Conf. (ITC)*, 1988, pp. 343–352.

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

[22] S. Gupta and G. Gibson, "Methods and circuitry for built-in self-testing of content addressable memories," U.S. Patent 6 609 222, Aug. 19, 2003.

[23] M. J. S. Smith, *Application-Specific Integrated Circuits*. Reading, MA: Addison-Wesley, 1997.

[24] J.-F. Li, "Testing priority address encoder faults of content addressable memories," in *Proc. Int. Test Conf. (ITC)*, 2005, pp. 1–10.

[25] 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 Conf. (AsianSim)*, 2004, pp. 521–528.

[26] J.-F. Li and C.-K. Lin, "Modeling and testing comparison faults for ternary content addressable memories," in *Proc. IEEE VLSI Test Symp. (VTS)*, 2005, pp. 60–65.

[27] K. Batson, R. Busch, G. Koch, F. Towler, and R. Wistort, "Redundant array architecture for word replacement in CAM," U.S. Patent 6 791 855, Sep. 14, 2004.

Nitin Mohan (S'01) received the B.Tech. degree (with honors) in electronics engineering from the Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi, India, in 1999, and the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2001, where he is currently working toward the Ph.D. degree in the department of electrical and computer engineering.

From 1999 to 2000, he was with Wipro Technologies, Bangalore, India, working on the design and verification of field-programmable gate arrays (FPGAs). During 2001–2002, he was with Sirific Wireless Corporation, Waterloo, ON, Canada, designing CMOS integrated circuits. He spent the summer of 2005 at DALSA Corporation, Waterloo, ON, Canada, designing analog/mixed-signal circuits. His research interests include memory and analog circuit design.

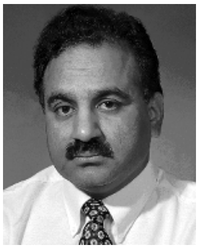
Mr. Mohan is a recipient of the Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship, the Ontario Graduate Scholarship, the President's Graduate Scholarship, and the Doctoral Thesis Completion Award.

Wilson Fung (S'00–M'05) received the B.A.Sc. degree in computer engineering and the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2004, respectively.

He was a member of the CMOS Design and Reliability Group at the University of Waterloo from 2003 to 2004. His research interests included the design of semiconductor memories such as DRAM and TCAM, and analog circuits for high-speed digital systems. In January 2005, he joined the I/O group at ATI Technologies Inc., Toronto, ON, Canada. He is currently working on high-speed I/O and on-chip ESD protection for the next-generation graphics processors.

Derek Wright (S'06) received the B.A.Sc. degree in electrical engineering and the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003 and 2005, respectively. He is currently working toward the Ph.D. degree in the collaborative electrical and biomedical engineering program at the University of Toronto, Toronto, ON, Canada.

His current research interests are in ultrasonic materials and systems.



Manoj Sachdev (SM'97) received the B.E. degree (with honors) in electronics and communication engineering from the University of Roorkee, Roorkee, India, and the Ph.D. degree from Brunel University, U.K., in 1984 and 1996, respectively.

He was with Semiconductor Complex Ltd., Chandigarh, India, from 1984 to 1989, where he designed CMOS integrated circuits. From 1989 to 1992, he worked in the ASIC division of SGS-Thomson, Agrate, Milan, Italy. In 1992, he joined Philips Research Laboratories, Eindhoven, The Netherlands, where he researched various aspects of VLSI testing and manufacturing. He is currently a Professor in the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. His research interests include low-power and high-performance digital circuit design, mixed-signal circuit design, and test and manufacturing issues of integrated circuits. He has written two books, two book chapters, and has contributed to over 125 technical articles in conferences and journals. He holds more than 15 granted and several pending U.S. patents in VLSI circuit design and test.

Dr. Sachdev is the recipient of several awards including the 1997 European Design and Test Conference Best Paper Award, the 1998 International Test Conference Honorable Mention Award, and the 2004 VLSI Test Symposium Best Panel Award.