A Comparative Study of Low-Power Techniques for Ternary CAMs

Nitin Mohan and Manoj Sachdev
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada – N2L 3G1
nitinm@ieee.org

***Abstract*—Ternary content addressable memories (TCAMs) are attractive for applications such as packet forwarding and classification in network routers. However, the high cost and power consumption are limiting their popularity and versatility. In this paper, we present a comparative study of the design techniques for low-power TCAMs.**

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. 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 migration of the Internet Protocol (IP) from IPv4 to IPv6 are further increasing the word-size and storage capacity of routing tables. Hence, current network routers require large-capacity TCAMs with high search speeds.

Fig. 1 illustrates the conventional 16T static TCAM cell. It consists of two SRAM cells to store ternary states ('0', '1', and "don't care"). Transistors N1 through N4 form a bit-level comparison logic to compare the stored value with the corresponding bit of the search key. A TCAM word is implemented by connecting several TCAM cells in parallel (in a row). Similarly, a TCAM array is formed by connecting several TCAM words in parallel (in a column).

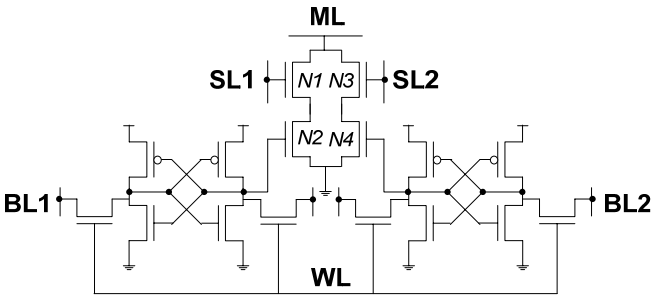

Fig. 1: Conventional 16T static TCAM cell

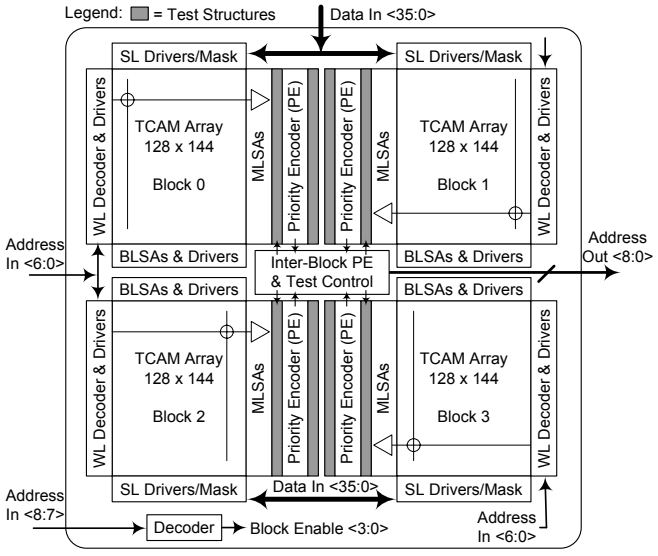

Fig. 2: Conventional 16T static TCAM cell

A typical TCAM chip consists of three major parts: (i) TCAM arrays for ternary data storage, (ii) peripheral circuitry for READ, WRITE, and SEARCH operations, and (iii) 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 priority encoders (PEs). The test and repair circuitry includes on-chip test structures and redundancy. Fig. 2 shows a simplified block diagram of a 512 x 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 lower address is given a higher priority. In addition, 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. 2) 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.

Despite the attractive features of TCAMs, high power consumption is the most critical challenges faced by TCAM designers. The parallel nature of TCAMs leads to high power consumption. For example, an 18Mb TCAM running at 250 million searches per second (MSPS) consumes 15W [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. These issues drive the need of innovative design techniques 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. In this paper, we present a comparative study of various design techniques for low-power TCAMs. The remaining paper is organized as follows: Section II presents various low-power match line sense amplifiers (MLSAs) and compares their trade-offs. Section III focuses on ML-segmentation techniques to reduce power. Finally, section IV concludes the paper with the key observations and recommendations.

II. Match Line Sense Amplifiers

In most TCAM applications, READ/WRITE operations are performed only when the chip is tested or the table is updated. Thus, 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.

Most low-power MLSAs strive to minimize the ML voltage swing. Fig. 2 illustrates the conventional MLSA. Initially, all the MLs are pre-charged to V_{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_{DD}. Otherwise, it discharges to ground through mismatching cells. In order to avoid short-circuit current, the SLs are switched to ground during the pre-charge phase. Hence, most of the SLs switch in every SEARCH operation, causing high power consumption. Fig. 3 shows a current-race sensing scheme [3]. This scheme has the MLs at the ground voltage during the pre-charge 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 (Vt), its MLSO changes from '0' to '1' (Fig. 3). A dummy ML emulating the "match" condition generates an MLOFF signal to end the ML sensing. Fig. 4 shows another MLSA that reduces ML voltage swing using charge-redistribution [6]. This scheme also has MLs at the ground voltage during the pre-charge phase. The ML sensing begins with fast pre-charging of MLs using a FastPre signal. Transistors N1 and N2 restrict the ML voltage swing to ($V_{REF} - V_t$). After the FastPre pulse, the MLs are left floating. For the "mismatch" condition, the ML voltage drops below ($V_{REF} - V_t$), 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. 4). A small current source (I_{REF}) feeds the SP node to compensate for ML leakages. The voltage V_{REF} can be varied to trade off power consumption with speed of operation. This method can reduce the ML voltage swing even below V_t. However, the fast pre-charging of mismatching MLs causes short circuit power dissipation. A charge-injection match detection circuit (CIMDC) eliminates this short circuit power (Fig. 5) [5]. CIMDC uses an injection capacitor (C_{INJ}) for each ML. Typically, C_{INJ} is sized 3-4 times smaller than C_{ML} [5]. Initially, all the injection capacitors are pre-charged to V_{DD} and all the MLs are discharged to ground. At evaluation, charge is injected from C_{INJ} to C_{ML} using ChargeIn signal (Fig. 5). For "match" condition, the voltage of C_{ML} rises to a voltage determined by the ratio of C_{INJ} and C_{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 (~ 300mV), it suffers from a lower noise margin and an area penalty due to C_{INJ}.

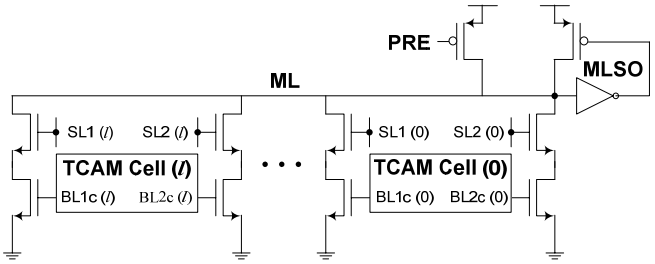

Fig. 2: Conventional precharge match line sense amplifier

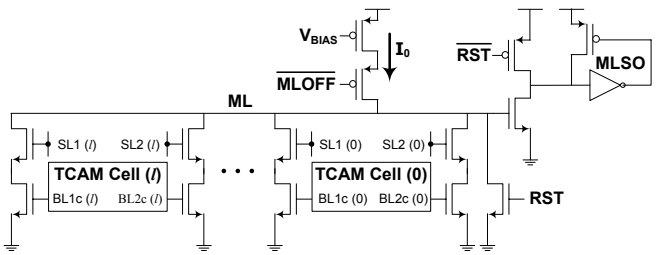

Fig. 3: Current-race match line sense amplifier

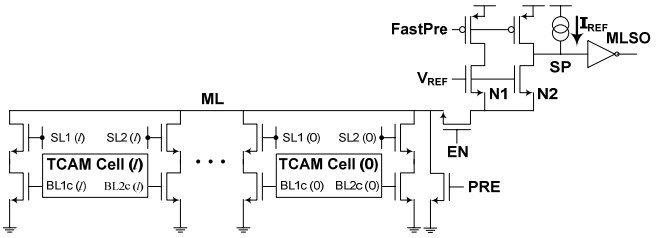

Fig. 4: Charge-redistribution match line sense amplifier

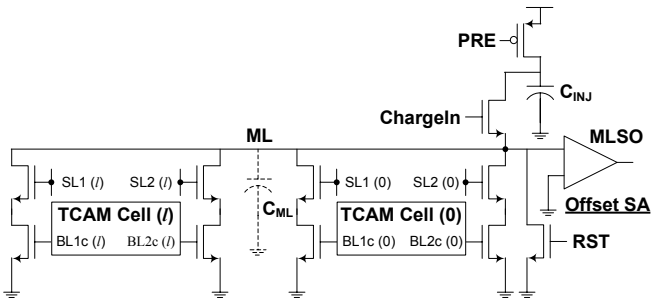

Fig. 5: Charge-injection match line sense amplifier

Fig. 6 shows the delay and energy consumption of the above ML sensing schemes for different word sizes when they are simulated in 0.18µm CMOS technology. Global masking (GM) also alters the delay and energy by changing the ML capacitance. The ML capacitance can be given by (1):

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

where 'g' is the number of globally masked bits, 'l' is the total number of bits per word, C_{DRAIN} is the drain capacitance of each transistor in the comparison logic, and C_{INT} is the interconnect capacitance of each ML. When a bit is globally masked (SL1 = SL2 = '0'), only the drain capacitances of transistors N1 and N3 (shown in Fig. 1) contribute to C_{ML}. Otherwise, C_{ML} also includes the capacitance of the internal nodes. Therefore, the worst case C_{ML} corresponds to no global masking (g = 0) and the best case C_{ML} relates to full global masking (g = *l*). Fig. 6(a) shows the energies of operation for both extremes. The search speed in Fig. 6(b) corresponds to the worst case. The precharge (or reset) duration is the same (1 ns) for fair comparison. We used C_{INT} = 0.18fF/cell from the post-layout extraction of TCAM layout with MLs routed in metal 4 (0.18µm CMOS process). Also C_{INJ} is sized to one-third of C_{ML}. Fig. 6 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. 4). Fig. 6(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.

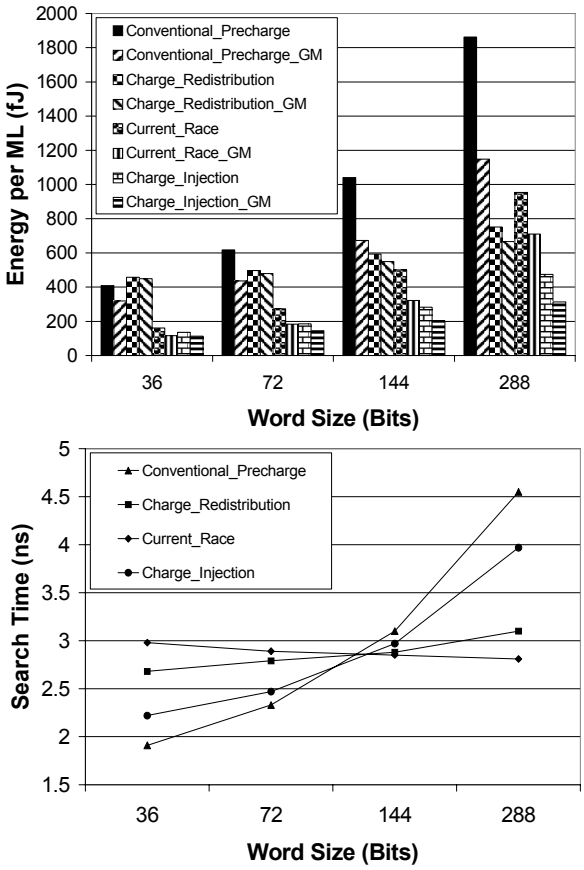

Fig. 6: Energy of operation per ML and search time for various ML sensing schemes

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 equation (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. 1). The capacitance of the internal nodes (N1-N2 and N3-N4 in Fig. 1) 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.

III. 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. 7(a)). The selective-precharge scheme divides the SEARCH operation into multiple stages. Fig. 7(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 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 the comparison logic respectively (Fig. 7(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: (i) most of the words in a TCAM array have multiple mismatches, and (ii) 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 0.18µm CMOS technology using current-race MLSA. Fig. 8 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 trade-off 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. 7(b)). The parasitic capacitance due to these lines increases the

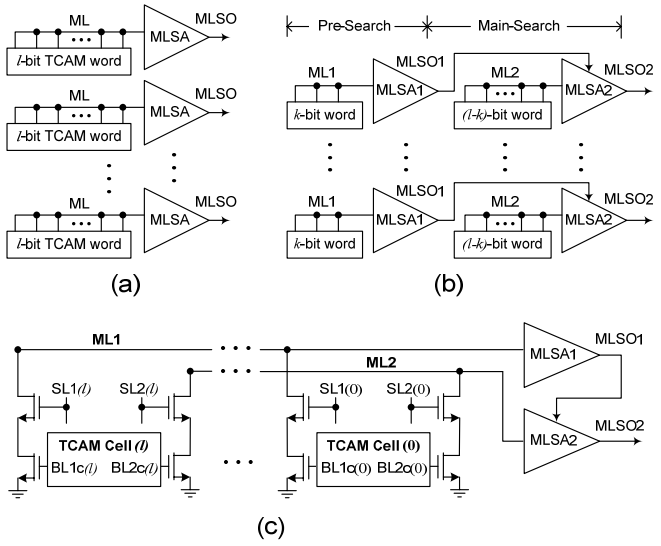

Fig. 7: (a) Conventional TCAM, (b) selective-precharge TCAM, (c) dual-ML TCAM

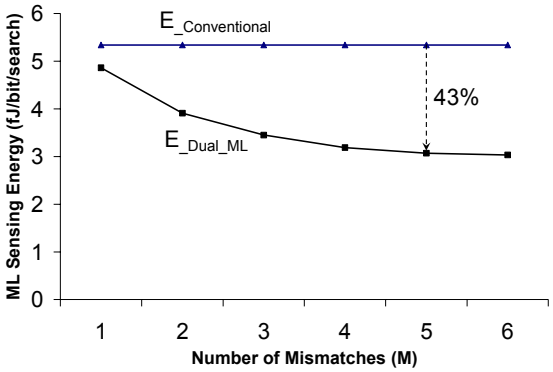

Fig. 8: Average ML sensing energy of conventional and dual-ML TCAMs

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. 7(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 time of a highly capacitive ML is much larger than the propagation delay of the MLSAs.

IV. Conclusions

We have presented a comparative study of several design techniques for low-power TCAMs. The existing TCAM design techniques accomplish power reduction by lowering the voltage swing of MLs. However, these schemes normally trade off 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 size and application of the TCAM. ML-segmentation techniques reduce power consumption independent of the MLSAs. They can be combined with low-power MLSAs for further power savings.

References

[1] K. Etzel, "Answering IPv6 Lookup Challenges," Technical Article, Cypress Semiconductor Corporation, Oct. 27, 2004, [Online], Available: http://www.cypress.com.

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

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

[4] P. Vlasenko, D. Perry, "Matchline sensing for content addressable memories," US Patent 6717876, Apr. 6, 2004.

[5] C. Zukowski, S. Wang, "Use of selective precharge for low-power CAMs," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 745-770, Jun. 9-12, 1997.

[6] N. Mohan, M. Sachdev, "Low-power dual matchline ternary content addressable memory," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 633-636, May 23-26, 2004.