Low-Power High-Performance Ternary Content Addressable Memory Circuits

by

Nitin Mohan

A thesis
presented to the University of Waterloo
in the fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2006

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Nitin Mohan

Abstract

Ternary content addressable memories (TCAMs) are hardware-based parallel lookup tables with bit-level masking capability. They are attractive for applications such as packet forwarding and classification in network routers. Despite the attractive features of TCAMs, high power consumption is one of the most critical challenges faced by TCAM designers. This work proposes circuit techniques for reducing TCAM power consumption. The main contribution of this work is divided in two parts: (i) reduction in match line (ML) sensing energy, and (ii) static-power reduction techniques. The ML sensing energy is reduced by employing (i) positive-feedback ML sense amplifiers (MLSAs), (ii) low-capacitance comparison logic, and (iii) low-power ML-segmentation techniques. The positive-feedback MLSAs include both resistive and active feedback to reduce the ML sensing energy. A body-bias technique can further improve the feedback action at the expense of additional area and ML capacitance. The measurement results of the active-feedback MLSA show 50-56% reduction in ML sensing energy. The measurement results of the proposed low-capacitance comparison logic show 25% and 42% reductions in ML sensing energy and time, respectively, which can further be improved by careful layout. The low-power ML-segmentation techniques include dual ML TCAM and charge-shared ML. Simulation results of the dual ML TCAM that connects two sides of the comparison logic to two ML segments for sequential sensing show 43% power savings for a small (4%) trade-off in the search speed. The charge-shared ML scheme achieves power savings by partial recycling of the charge stored in the first ML segment. Chip measurement results show that the charge-shared ML scheme results in 11% and 9% reductions in ML sensing time and energy, respectively, which can be improved to 19-25% by using a digitally controlled charge sharing time-window and a slightly modified MLSA. The static power reduction is achieved by a dual-V_{DD} technique and low-leakage TCAM cells. The dual-V_{DD} technique trades-off the excess noise margin of MLSA for smaller cell leakage by applying a smaller V_{DD} to TCAM cells and a larger V_{DD} to the peripheral circuits. The low-leakage TCAM cells trade off the speed of READ and WRITE operations for smaller cell area and leakage. Finally, design and testing of a complete TCAM chip are presented, and compared with other published designs.

Acknowledgements

This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Postgraduate Graduate Scholarship (PGS B) and Micronet R&D. I would like to thank the Canadian Microelectronics Corporation (CMC) for fabricating the test chips, and providing design tools, test equipment and technical support for chip design.

I would like to express my deep gratitude to my supervisor, Prof. Manoj Sachdev, for providing moral and financial support, technical guidance and encouragement. Right from the beginning of my Ph.D., he had clear vision and expectations about my research. He gave me full freedom to develop my research. At the same time, he was easily accessible whenever I needed his guidance and feedback. During the tough times (when I was having difficulties in chip testing), he provided me moral support and useful tips. My association with him made my Ph.D. an enjoyable learning experience. I am honored to have Prof. B. Cockburn as the external examiner, and Prof. A. Sedra, Prof. M. Aagaard and Prof. J. Vanderkooy as the members of my Ph.D. examination committee.

I would like to acknowledge Wilson Fung and Derek Wright, who were my groupmates working on priority encoders and test algorithms for TCAMs, respectively. Our regular meetings included several stimulating discussions on design, testing and applications of TCAMs. I immensely benefited from their thorough feedback on my work. I would like to thank my colleagues and friends in the CMOS Design and Reliability (CDR) group: B. Chatterjee, M. Sharifkhani, S. M. Jahinuzzaman, H. Sarbishaei, S. Ardalan, O. Semenov, M. Maymandi-Nejad, A. Vassighi, M. El-Gebaly, A. Bagga, C. Kwong, D. Li, I. Lovich, S. Naraghi, and F. Asgari for their support during my study and research. I am especially thankful to David Rennie and Andrei Pavlov for helping me to design and test my chips. They were always helpful whenever I had trouble with my computer or test equipment. Special thanks to my best friends in Waterloo: Kapil Sakariya and Anil Kumar, who made my stay in Waterloo full of fun and learning. I'll treasure our late evening discussions on every topic under the sun. A very special thanks to Phil and Fernando for their kind support with chip design tools and computers in all urgent situations and deadlines. I would like to thank the ECE office staff, especially Wendy Boles and Janet Reimer.

Last, but definitely not the least, a very very special thanks to my loving wife Mohini, who always supported me with her unending love and care during all ups and downs of my Ph.D. Whenever I felt down, she was there to motivate me. I would also like to express my deep gratitude to my dearest parents. Their encouragement, moral support, faith and love were the source of my motivation, energy and enthusiasm. A special thanks to my brother Sachin, who always supported me with his words of encouragement. I would also like to acknowledge my sister-in-law, loving nephew, and my in-laws for their love and support.

Nitin Mohan
August 2006
Waterloo, Ontario, Canada

To my wife and parents

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Phenomenal growth in the number of Internet users and the increasing popularity of bandwidth-hungry real-time applications have resulted in a demand for very high-speed networks. The Internet is a mesh of routers and switches, which process data packets and forwards them toward their destinations. Each packet contains a header and a payload. The header contains information such as a source address, a destination address, the data length, a sequence number and the data type of the packet [1]. A network switch transfers an incoming data packet to an output port based on the information in the header of the packet. A router is a more sophisticated switch that forwards an incoming packet after routing its path from the source to the destination [1]. Each router maintains a routing table and forwards incoming packets based on the information stored in the routing table. Routers also communicate with one another to update their routing tables.

Typically, the physical medium that transports the data from one router to another is made of optical fiber. Advances in optical fiber technologies, such as wavelength division multiplexing, have drastically increased the data transfer rates over optical fibers. In order to

utilize the full potential of optical fiber technology, routers need to meet the growing data rate [2]. One of the most time consuming task in a network switch or a router is table lookup. The growing demand for high-speed networks is pushing the existing solutions to their limits in order to meet the increasing packet processing rates. For example, the packet processing rates for ATM at OC-48 (2.5Gb/s), OC-192 (10Gb/s) and OC-768 (40Gb/s) line rates are approximately 8, 33 and 134 million packets/s, respectively [3]. New features such as flow analysis, policy based routing, and Quality of Service (QoS) are increasing the quantity and variety of the table lookups. The high priority packets (such as voice and video) are processed before the low priority packets (such as data) to maintain the QoS. These features require multiple lookups for each incoming packet before it is forwarded to the next node. For OC-192 line rates, this translates to over 100 million searches per second (Msps).

The current version of Internet protocol (IP), commonly known as IPv4, supports only 32-bit IP addresses. Due to the rapid increase in the number of Internet users, there is a growing shortage of IPv4 addresses, which are needed by all new machines added to the Internet. Hence, a new version of IP (IPv6) has been introduced that supports 128-bit addresses. IPv6 is expected to gradually replace IPv4, with the two coexisting during the transition period. The "IPv4 to IPv6 migration" has different design implications on packet forwarding and policy lookups. The increasing number of network nodes supported by IPv6 significantly increases the capacity and word-size of the routing table used for packet forwarding [4]. For packet classification or policy lookups of the IPv6 packets, interface bandwidth and latency are more critical challenges than capacity due to two main reasons [4]. First, the packet classification requires five-tuple lookup (IP source and destination address, layer 4 sources and destination, and layer 4 protocol). Thus, the word-size increases from 104 bits (IPv4) to 296 bits (IPv6), which slows down the lookup speed [4]. Second, multiple policy lookups are performed on most packets [4]. These lookups are often recursive, where the result of one lookup affects the following lookup. Thus, a large latency can significantly slow down the policy lookups.

Software methods for table lookup such as radix trees are relatively slow, and they do not scale well with the table size [5]. Under good conditions, a hash function can perform the lookup in one memory access. However, its worst-case search time, which depends on the table size and the hashing function, can be considerably worse than the tree searches [5].

Therefore, many of the table lookup tasks at different network layers that were originally implemented in software are now being replaced by hardware solutions to meet the performance requirements. An efficient hardware solution to perform table lookup is the content addressable memory (CAM). A CAM can be used as a co-processor for the network processing unit (NPU) to offload the table lookup tasks. Besides the networking equipment, CAMs are also attractive for other key applications such as translation look-aside buffers (TLBs) in virtual memory systems [6][7], tag directories in associative cache memories [8][9], database accelerators [10], data compression [11], and image processing [12]. Recent applications of CAMs include real-time pattern matching in virus/intrusion-detection systems and gene pattern searching in bioinformatics [13][14]. Since the capacities and word-sizes of CAMs used in most of these applications are much smaller than the CAMs used in networking equipment, the current CAM research is primarily driven by the networking applications, which require high capacity CAMs with low-power and high-speed operation.

CAM is an outgrowth of random access memory (RAM). In addition to the conventional READ and WRITE operations, CAMs also support SEARCH operations. A CAM stores a number of data words and compares a search key with all the stored entries in parallel. If a match is found, the corresponding memory location is retrieved. In the presence of multiple matches, a priority encoder (PE) resolves the highest priority match [15]-[17]. CAM-based table lookup is very fast due to the parallel nature of the SEARCH operation.

1.1. Binary versus Ternary CAMs

CAMs can be divided into two categories: (i) binary CAMs and (ii) ternary CAMs (TCAMs). A binary CAM can store and search binary words (made of ‘0’s and ‘1’s). Thus, binary CAMs are suitable for applications that require only exact-match searches. A more powerful and feature-rich TCAM can store and search ternary states (‘1’, ‘0’, and ‘X’). The state ‘X’, also called ‘mask’ or ‘don’t care’, can be used as a wild card entry to perform partial matching. Masking can be done both globally (in the search key) and locally (in the table entries). Figure 1.1 shows examples of global and local masking in TCAMs. In Figure 1.1(a), the search key 10110XXX will match with all the entries that fall in the following range: 10110000 to 10110111 (words located at addresses 1 and 4 in this case). It is called global masking because the last three bits of all the table entries are ignored. In Figure 1.1(b), word

110-XX-010 (located at address 2) will match with any of the following search keys: 110-00-010, 110-01-010, 110-10-010 and 110-11-010. The mask feature is particularly suitable for longest-prefix match searches in classless inter-domain routing (CIDR) [18][19]. TCAMs are also becoming popular in solving other problems such as sorting and range searching [20].

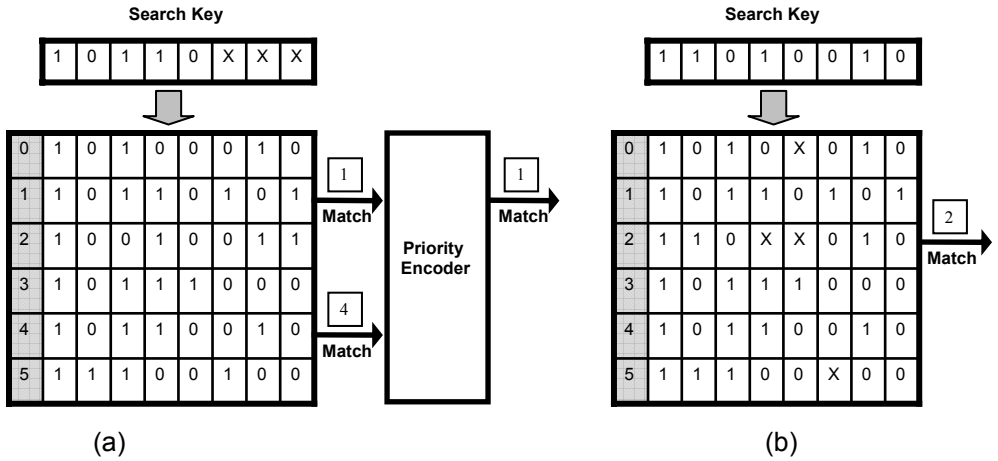


Figure 1.1: SEARCH operation in a TCAM with (a) global masking and (b) local masking

1.1.1. The Binary CAM Cell

A typical 10-transistor (10T) binary CAM cell is shown in Figure 1.2. The bit storage portion is a standard 6T static RAM (SRAM) cell. Hence, this cell performs READ and WRITE operations similar to an SRAM cell. Transistors N1-N4 implement the XNOR logic to compare the table entry with the search key. In order to avoid confusion, we will follow the active high convention consistently: '1' = V_{DD}, and '0' = 0V or ground (GND). Logic states of the cell are defined as shown in Table 1.1.

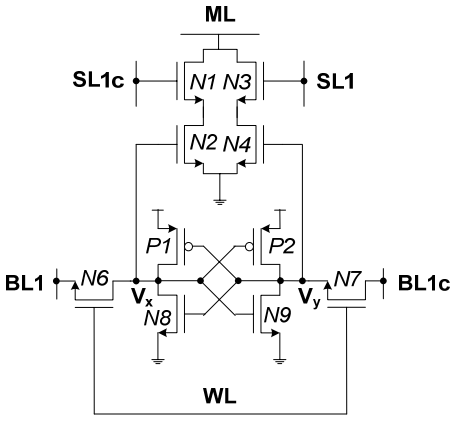


Figure 1.2: Circuit schematic of a conventional 10T binary CAM cell

Table 1.1: Definition of logic states in a binary CAM cell

Logic State	V_x *(V)*	V_y *(V)*
'0'	0	V_{DD}
'1'	V_{DD}	0

1.1.1.1. WRITE Operation

The WRITE operation is performed by placing the data on the bit lines (BLs) and enabling the word line (WL). This turns on the access transistors (N6-N7), and the internal nodes of the cross-coupled inverters are written by the BL data. Figure 1.3 shows the WRITE operation when '0' is being written to a cell which originally stored '1'. Originally, V_x = '1' and V_y = '0', P1 and N9 were 'ON', and P2 and N8 were 'OFF'. When WL is enabled (WL = '1'), access transistors (N6-N7) conduct resulting in BL currents I_0 and I_1 (shown by dashed arrows in Figure 1.3). These transient currents form voltage dividers (P1-N6 and N7-N9). If these transient currents can pull one of the nodes (V_x and V_y) to the inverter threshold voltage, the other node will flip due to the feedback action of the cross-coupled inverters. If the inverter threshold voltage is $\frac{V_{DD}}{2}$, N7 needs to be much larger (≥10x) than N9 to pull V_y above this value because it is difficult to pass logic '1' using an NMOS transistor [21]. On the other hand, V_x can be pulled below this value by choosing same size P1 and N6 (shown by encircled W in Figure 1.3). Thus, the latter sizing is adopted almost universally.

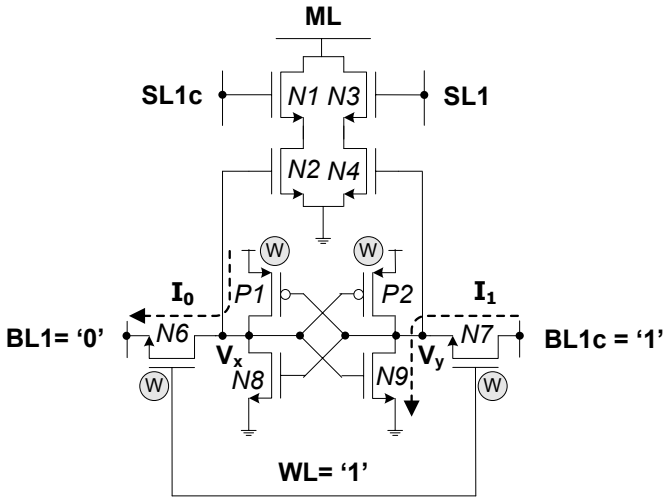


Figure 1.3: WRITE operation in a conventional 10T binary CAM cell

1.1.1.2. READ Operation

The READ operation is performed by pre-charging the BLs to V_{DD} and enabling the WL. Figure 1.4 shows the READ operation, when '0' is stored (i.e. V_x = '0', V_y = '1'). Since the BL drivers are turned off during the READ operation, current I_{READ} discharges BL1 (through N6 and N8). BL1c remains at V_{DD} because V_y = '1'. Therefore, a small differential voltage develops between BL1 and BL1c, which is amplified to a rail-to-rail voltage by a BL sense amplifier (BLSA). Since the BLs are shared among all the cells in a column, they are highly capacitive. The small voltage swing in the BLs reduces power consumption and the access time during the READ operation. As shown in Figure 1.4, the current I_{READ} raises the voltage V_x. Thus, the driver transistors (N8-N9) are sized such that V_x remains below the inverter threshold voltage, and hence the cell does not flip during the READ operation. Typically, the driver transistors (N8-N9) are sized 1.5 times wider than the access transistors (N6-N7).

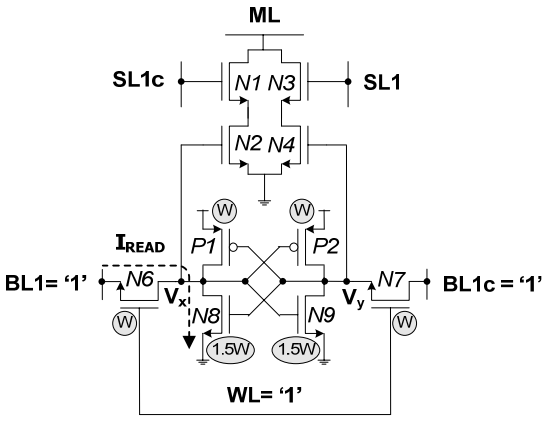


Figure 1.4: READ operation in a conventional 10T binary CAM cell

1.1.1.3. SEARCH Operation

The conventional SEARCH operation is performed in three steps. First, search lines (SLs) SL1 and SL1c are reset to GND. Second, ML is pre-charged to V_{DD}. Finally, the search key bit and its complementary value are placed on SL1 and SL1c, respectively. If the search key bit is identical to the stored value (SL1=BL1, SL1c=BL1c), both ML-to-GND pull-down paths remain 'OFF', and the ML remains at V_{DD} indicating a "match". Otherwise, if the search key bit is different from the stored value, one of the pull-down paths conducts and discharges the ML to GND indicating a "mismatch". Resetting SL1 and SL1c to GND during

the ML pre-charge phase ensures that both pull-down paths are 'OFF', and hence do not conflict with the ML pre-charging. Figure 1.5 shows the SEARCH operation when '0' is stored in the cell (V_x = '0' and V_y = '1'). For SL1 = '1' (SL1c = '0'), ML is discharged to '0' detecting "mismatch" as shown in Figure 1.5(a). Similarly for SL1 = '0', ML remains at '1' detecting "match" as shown in Figure 1.5(b).

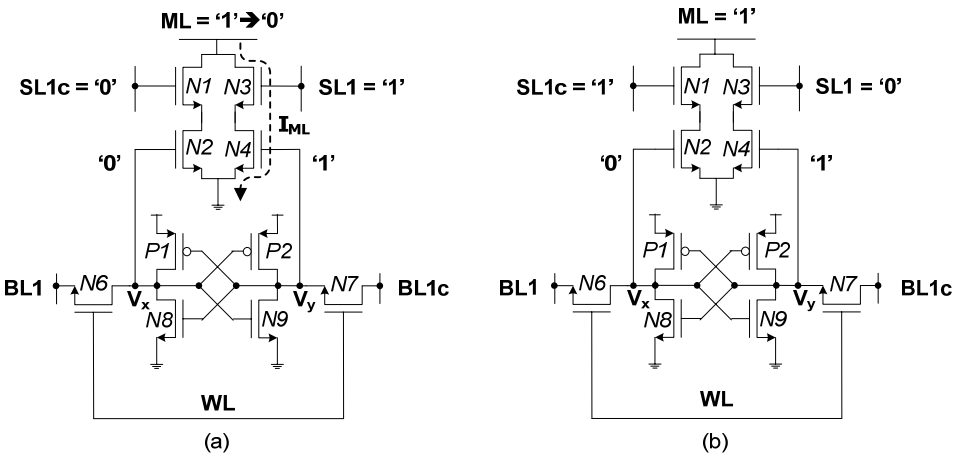


Figure 1.5: SEARCH operation in a 10T binary CAM cell for (a) "mismatch", and (b) "match"

1.1.2. TCAM Cell

A typical 16T static TCAM cell is shown in Figure 1.6. It is similar to the binary CAM cell except that it has two SRAM cells to store ternary data, as shown in Figure 1.6. READ, WRITE and SEARCH operations in this cell are performed in the same way as described in section 1.1.1. For the given circuit style, masking can be achieved by turning off both ML-to-GND pull-down paths. For example, global masking is performed by SL1 = SL2 = '0', and local masking is achieved by V_x = V_y = '0'.

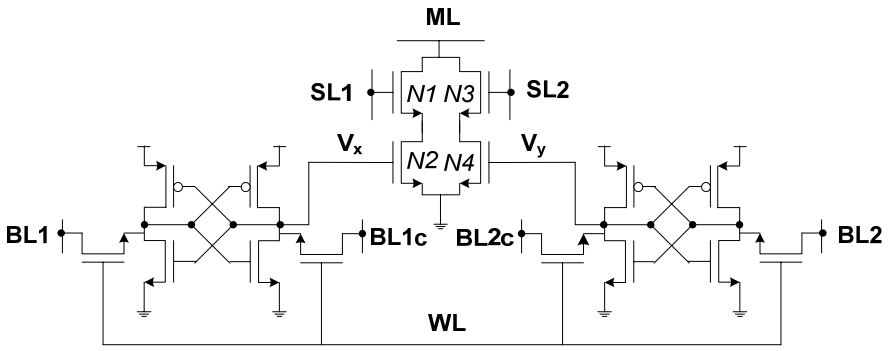


Figure 1.6: Circuit schematic of a conventional 16T static TCAM cell

1.1.3. CAM Array

A CAM word (*l*-bit) is implemented by connecting (*l*) CAM cells in parallel (each row in Figure 1.7). All the cells in a CAM word share an ML but they have separate SLs. The ML is connected to a ML sense amplifier (MLSA), which determines if the corresponding word matches with the search key. As described in Section 1.1.1, a conventional MLSA pre-charges the ML to V_{DD} and places the search key on the SLs. During this operation, the ML remains at V_{DD} only if the cell-level comparisons of all the bits result in "match". In other words, even a single-bit "mismatch" can create a discharge path for ML indicating a word (*l*-bit) "mismatch". A CAM array (*n* x *l*) is implemented by connecting (*n*) CAM words sharing the same set of SLs. When the search key (*l*-bit) is written on SLs, it is compared with all the (*n*) words in parallel. Figure 1.7 illustrates the main components of an (*n* x *l*) CAM array including the SL drivers and a search key register.

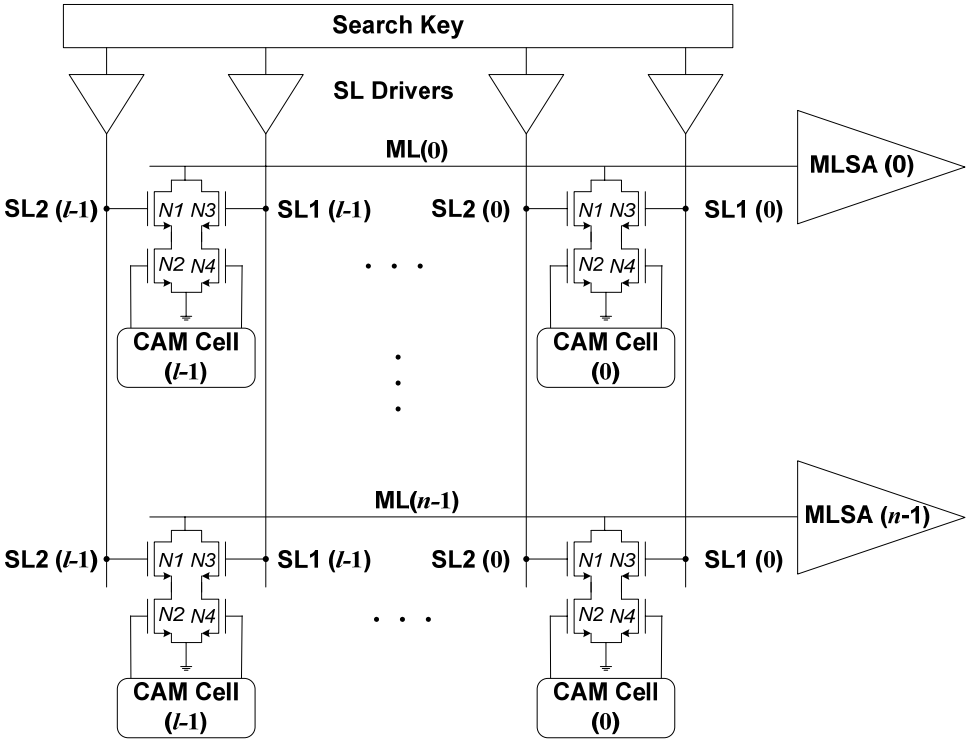


Figure 1.7: CAM array for storing *n* words (*l*-bit wide)

In networking applications, TCAMs perform READ and WRITE operations only for table-update, maintenance and testing. Hence, the total power consumption of a TCAM chip is dominated by the SEARCH operations. All MLs are precharged to V_{DD} prior to every

SEARCH operation. If all table entries match with the search key, MLs remain at V_{DD}, and the subsequent SEARCH operation does not consume energy to pre-charge them. Unfortunately in most applications, the majority of the table entries do not match with the search key. As a consequence, almost all MLs are discharged to GND in every SEARCH operation. The switching activity of the SLs depends on the statistics of the search key. Assuming random data, each bit in the search key has equal probability of being '0' or '1'. Thus, almost half of the SLs switch in every SEARCH operation. Each ML (SL) is highly capacitive because it is shared by all cells in a row (column) as shown in Figure 1.7. Therefore, owing to their large capacitances and high switching activities, the MLs and SLs consume a significant portion of the total TCAM power.

1.1.4. TCAM Chip Organization

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 address decoders for READ/WRITE operations, BLSAs for READ operations, and SL drivers, MLSAs and PE for SEARCH operations. The test and repair circuitry includes on-chip test structures (such as multiplexers and scan chains) and redundancy. Figure 1.8 shows a simplified block diagram of a 512 x 144 TCAM implemented as four smaller TCAM arrays. As mentioned in section 1.1.3, each row in a TCAM array stores a word. Within a word, a bit is located by its column number. Since partial matching in a TCAM may result in multiple matches, a PE is used to determine the highest priority match. Conventionally, a word with lower address is given a higher priority. In addition, the PE also generates a signal which indicates the presence or absence of multiple matches [17][22]. Typically, the highest priority match from a TCAM is encoded in binary format ("Address Out" in Figure 1.8) 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. Both row and column redundancy can be used to replace the faulty components depending on the number and locations of the faults. However, row redundancy requires special techniques to preserve the logical address-order for valid multiple match resolution and address encoding [15][23][24]. On the other hand, excessive

column redundancy increases the delay and energy consumption due to the additional ML-capacitance imposed by the redundant cells.

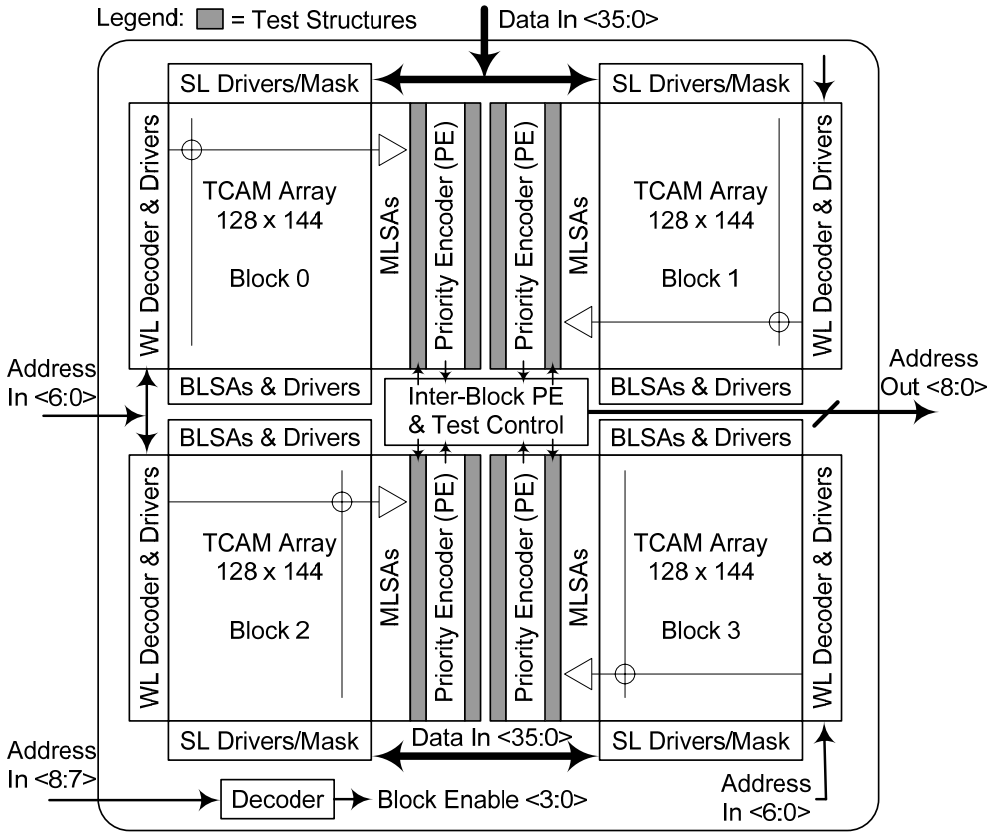


Figure 1.8: Simplified block diagram of a 512 x 144 TCAM

1.2. Low-Power and Low-Energy CAMs

Although a TCAM-based network search engine (NSE) significantly increases the speed of table lookups in network systems, it is power hungry due to the parallel SEARCH operation. For example, an 18Mb TCAM running at 250 Msps consumes 15W [4]. The power consumption is expected to grow further due to two main trends: (i) IPv4 to IPv6 migration, and (ii) the growing demand for higher line rates (such as OC-768). The first trend implies longer Internet address and hence larger CAMs, which are slower and consume more energy. The second trend requires techniques to increase the speed of the SEARCH operation. A side effect of this trend is higher power consumption. The maximum power dissipation capacities of integrated circuits (ICs) are constrained by packaging thermal impedance and require

expensive cooling mechanisms to increase the power handling capability. Thus, the growing power consumption of CAMs is increasing the packaging cost and the junction temperature, which also leads to other issues such as reduced performance, poor reliability, etc. The power consumption should be reduced in order to implement cost-effective and reliable CAMs. CAMs are also attractive for portable applications, which require low energy operation for longer battery life.

The power consumption consists of two parts: (i) static power, and (ii) dynamic power. Conventionally, static power is a small part of the total power consumption, and the dynamic power dominates. A general expression of the dynamic power consumption can be written as follows [21]:

$$P_{DYN} = \alpha\, C\, V_{SW}\, V_{DD}\, f \qquad (1.1)$$

where α = average switching activity

C = total node capacitance

V_{SW} = average voltage swing

V_{DD} = power supply voltage

f = frequency of operation.

The dynamic power can be reduced by decreasing the frequency of operation. However, this method is not acceptable due to the high-speed requirements. A better metric to show the superiority of a circuit is the average energy of operation, which can be given by (1.2).

$$E_{AVG} = \alpha\, C\, V_{SW}\, V_{DD} \qquad (1.2)$$

Equation (1.2) suggests that slowing down the circuit does not affect the average energy of operation. For the same speed, the power consumption directly scales with the energy of operation. Since a larger CAM has a larger node capacitance, it consumes more energy per SEARCH operation. Hence, a more realistic figure of merit to compare different CAM circuits is the average energy consumption per bit per search.

1.3. SRAM-based versus DRAM-based CAMs

So far, we have discussed only SRAM-based CAMs (Figures 1.2-1.6). Similar configurations of DRAM-based CAMs are also available. A DRAM-based TCAM cell, shown in Figure 1.9, is attractive due to its smaller cell area. It requires only 6 transistors and two capacitors (compared to 16 transistors in an SRAM-based TCAM cell).

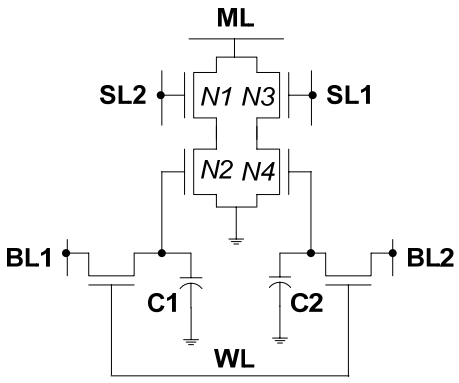


Figure 1.9: Circuit schematic of conventional DRAM-based TCAM cell

The high bit density of DRAM-based TCAMs comes at the expense of special processing, and extra refreshing circuitry to compensate for the charge leakage from the storage capacitors. The special processing makes the embedded DRAM technology more expensive than the pure logic technology with the same feature size [25]. In addition, further technology scaling is increasing the transistor leakages dramatically. Thus, in the more advanced technologies (90nm and below), DRAMs may require very high bit refreshing rates. This directly translates to additional power and performance penalty since the DRAM-based TCAM is not accessible during a refresh cycle. This work focuses on the SRAM-based TCAMs due to two main reasons:

(i) SRAM-based TCAMs are compatible with the regular CMOS logic process.

(ii) Most commercial TCAM vendors have moved away from DRAM-based TCAMs.

1.4. Motivation

As described earlier, very high-speed routers are needed to meet the requirements of the current and future generations of high-speed networks. Increasing security concerns require rigorous scrutiny of each packet before it is forwarded to the next node. This also increases the number of lookups needed for each packet. Therefore, the speed requirement of routing table lookups is increasing rapidly due to two major trends:

(i) Increasing numbers of packets need to be processed per second.

(ii) Increasing numbers of lookups need to be performed per packet.

Table 1.2 highlights the approximate data throughput and packet processing time for ATM over SONET at different line rates [3].

Table 1.2: Approximate packet processing time budgets for ATM over SONET [3]

Media	*Link rate*	*Packets/sec (Million)*	*Time per packet (ns)*
OC-3	150 Mbps	0.5	2034.5
OC-12	625 Mbps	2.0	488.2
OC-48	2.5 Gbps	8.4	119.2
OC-192	10 Gbps	33.5	29.8
OC-768	40 Gbps	134.2	7.4

Considering the small time available to process each packet, TCAM is undoubtedly the most time-efficient solution for packet classification and forwarding. However, it is still considered to be a costly solution by the networking industry because the commercially available TCAM chips are not only more expensive but also exhibit higher power consumption and lower bit density than the same capacity RAM chips. Although the density and power efficiency of TCAMs will improve with further technology scaling, circuit and architectural techniques are needed to further reduce the power budget and associated cost. Power reduction is also attractive for integrating the TCAM co-processor with the network processor and other accompanying components. Most of the system-on-chip (SoC) implementations also require low-power sub-systems.

Considering the high-speed table lookup requirements from Table 1.2, reduction in TCAM search time is also essential to meet the line rates of OC-768 and above. Table 1.3 highlights the speed of some of the commercially available TCAM-based NSEs. Depending on the number of lookups per packet, only some of the commercially available NSEs can support the line rates for OC-192 and OC-768. As the number of look-ups per packet increases due to the higher complexity of networking applications, the present generation of TCAMs will not be able to sustain the high network traffic for OC-768 and beyond. Innovative circuit techniques can make TCAMs more attractive for networking applications by reducing their power and search delay. As the cost and power consumption of TCAMs goes down, their appealing features can also be exploited by other applications, which are currently using software approaches. According to a *Semico* market research study published in 2003, CAM market was expected to grow at a compound annual growth rate of 20.4% in revenue and 28.9% in units (from $85.7 million in 2003 to $300 million in 2006) [26].

Table 1.3: Key features of the commercially available TCAM-based NSEs

TCAM Vendor	*Sustained search rate (Msps)*	*Memory size (Mb)*	*Supported data words (bits)*	*Search pipeline delay (ns)*	*Core Supply Voltage (V)*
Netlogic [27]	400	18	72/144	2.5	1.2
	-	18	36/288/576	-	1.2
Cypress [28]	266	18	72/144	3.8	1.2
	133	18	36/288	7.5	1.2
	66.5	18	576	15	1.2
IDT [29]	250	18	72/144	4	1.2
	-	18	36/288/576	-	1.2

This work proposes circuit techniques to reduce the energy and search delay of TCAMs. If the additional speed is not needed for a given application, it can be traded for lower power. Although this work focuses on SRAM-based TCAMs, most circuit techniques proposed by this work are equally applicable to the DRAM-based TCAMs and binary CAMs. The next chapter reviews some of the existing low-power techniques for binary and ternary CAMs. Traditionally, most CAM design techniques have been invented for binary CAMs. In recent years, more work has been reported on TCAM designs. In that chapter, we analyze the suitability of these techniques for modern TCAMs, and assess their speed, power, area and noise margin trade-offs.

Chapter 3 presents low-power TCAM cell design techniques. It highlights the growing importance of static power consumption in TCAMs, and presents two techniques to reduce the static power: dual-V_{DD} and low-leakage TCAM cells. The dual-V_{DD} technique reduces the static power by employing a smaller V_{DD} in the TCAM storage portion and a larger V_{DD} in the peripheral circuits. The low-leakage TCAM cells trade the speed of READ and WRITE operations for smaller leakage and cell area. Chapter 3 also proposes a cell-level comparison logic that makes the MLs less capacitive. A smaller ML capacitance reduces the energy and delay of ML sensing. We present circuit analyses of these schemes, and substantiate them by simulation and chip measurement results.

Chapter 4 proposes circuit techniques for low-power ML sensing. First, we present two word-level power reduction techniques: dual ML TCAM and charge-shared MLs. Both techniques are based on "ML segmentation", which divides a wide ML into smaller segments and then senses them sequentially. The dual ML TCAM reduces the power consumption by exploiting the fact that in most TCAM applications, most table entries exhibit multiple-bit "mismatch" with the search key. The charge-shared ML scheme reduces the delay and worst-case energy of ML sensing by partial recycling of the ML charge. This chapter also presents three novel MLSAs that employ positive feedback techniques to reduce the power consumption in TCAMs. First MLSA uses a transistor in the triode region to create a resistive feedback loop. Second MLSA uses three transistors to form an active feedback loop. Although the former is more area-efficient, the latter achieves higher power savings due to larger loop gain and faster response. Third MLSA uses body-bias to improve the feedback action of the active-feedback MLSA. After describing the detailed operation of these MLSAs, we present simulation and chip measurement results to substantiate the circuit analyses. We also compare the measurement results with those of the other published TCAM designs.

Chapter 5 is devoted to the design and testing of a complete TCAM chip which integrates the individual components such as TCAM array, priority encoder, address and column decoders, data multiplexer, data registers, design for testability (DFT) structures, line drivers, bias and control generation, etc. We also discuss the physical design issues of such a complex chip containing nearly 400,000 transistors. We present the analysis and implementation of a large on-chip decoupling capacitance (nearly 1nF) required by this chip. We discuss the chip-level simulation strategy, PCB development, power measurement, test planning and execution of this chip. We also present the measurement results of individual components and for a 144x64 TCAM block.

Finally, Chapter 6 concludes this work highlighting its main contributions and key observations. It also discusses the current trends and possible directions for the future research in this area.

Chapter 2

CAM Review

In Chapter 1, we discussed only NOR-type CAMs along with the conventional pre-charge MLSA. In recent years, several alternative circuits and architectures have been proposed for reducing the cell area and power consumption of CAMs. This chapter provides a brief review of various TCAM cells, MLSAs, SL drivers, and other low-power techniques developed for CAMs. Most of these techniques reduce the power consumption at the expense of at least one of the following: lower speed, reduced noise-margin, and larger area. Hence, these techniques provide trade-offs between speed, power, area and noise margin.

In order to understand various trade-offs in CAM design, it is useful to write a general expression for the SEARCH operation. When a search key is applied to a CAM-based table, it is compared bit-by-bit with all the table entries. If all the bits of a table entry match with the respective bits of the search key, it indicates a "match". Hence, the match signal can be represented by the following Boolean expressions,

$$M = \overline{[SL(l-1) \oplus BL(l-1)]} \cdot \overline{[SL(l-2) \oplus BL(l-2)]} \ldots \overline{[SL(0) \oplus BL(0)]} \qquad (2.1)$$

$$\overline{[SL(l-1) \oplus BL(l-1)]} = [SL(l-1) \cdot BL(l-1)] + [\overline{SL(l-1)} \cdot \overline{BL(l-1)}] \qquad (2.2)$$

Here $SL(l\text{-}1)$ and $BL(l\text{-}1)$ are 'l^{th}' bits (or in this case, the most significant bits) of the search key and the table entry, respectively. Notice that the logical function shown in equation (2.2) is essentially the XNOR logic i.e. $\overline{[SL(l-1) \oplus BL(l-1)]}$ is '1' if $SL(l-1) = BL(l-1) = '1'$ or $SL(l-1) = BL(l-1) = '0'$. Therefore, a "match" (M = '1') will occur when $\overline{[SL(l-1) \oplus BL(l-1)]} = '1'$ for all the bits: 0 to (l-1). Taking the complements of both sides, equation (2.1) can be rewritten as

$$MM = \overline{M} = [SL(l-1) \oplus BL(l-1)] + [SL(l-2) \oplus BL(l-2)] \ldots [SL(0) \oplus BL(0)] \quad (2.3)$$

$$[SL(l-1) \oplus BL(l-1)] = [SL(l-1) \cdot \overline{BL(l-1)}] + [\overline{SL(l-1)} \cdot BL(l-1)] \quad (2.4)$$

Hence, the "mismatch" (MM = '1') will occur when at least one bit of the search key fails to match the corresponding bit of a table entry. The logical function $[SL(l-1) \oplus BL(l-1)]$ in equation (2.4) is essentially the XOR logic.

The above analysis implies that the SEARCH operation can be performed either for "match" or for "mismatch" using equations (2.1) or (2.3), respectively. The bit level "match" and "mismatch" can be implemented using equations (2.2) and (2.4), respectively. The implementation of equations (2.1) and (2.3) requires l-input AND and OR logic, respectively. Consequently, CAMs can be implemented in two ways: NAND-type and NOR-type. Both methods have their benefits and shortcomings that will be discussed in the next section.

2.1. NAND versus NOR CAMs

Considering the large fan-in (typically l = 144) requirement, dynamic logic based designs are more suitable for lower transistor count and higher speed. The NAND and NOR designs depicting "match" (M) and "mismatch" (MM), respectively, are shown in Figure 2.1. It can be noticed that Figures 2.1(a) and 2.1(b) are consistent with equations (2.1) and (2.3), respectively. Also, the bit-level logic circuits in Figures 2.1(a) and 2.1(b) are equivalent to equations (2.2) and (2.4), respectively. Figure 2.1(a) is valid only for binary CAMs, where SL1c and BL1c represent the complementary values of SL1 and BL1, respectively. Figure 2.1(b) is valid for both binary and ternary CAMs. For binary CAMs, SL2 and BL2 can be replaced by SL1c and BL1c, respectively. The operation of the NOR-type CAM has already been described in Chapter 1. The NAND-type match line (ML_{NAND}) can also be sensed using the conventional MLSA (described in Chapter 1). Initially, ML_{NAND} is pre-charged to V_{DD}.

Now the ML_{NAND} discharges to GND only if every bit of the stored word [BL1(*l*-1) to BL1(0)] matches with the corresponding bit in the search key [SL1(*l*-1) to SL1(0)], i.e. all the series connected NMOS transistors are turned 'ON'. Typically, a NAND-type CAM consumes much less power than a NOR-type CAM because only a few table entries match with the search key, and most NAND-type MLs remain at the pre-charged value.

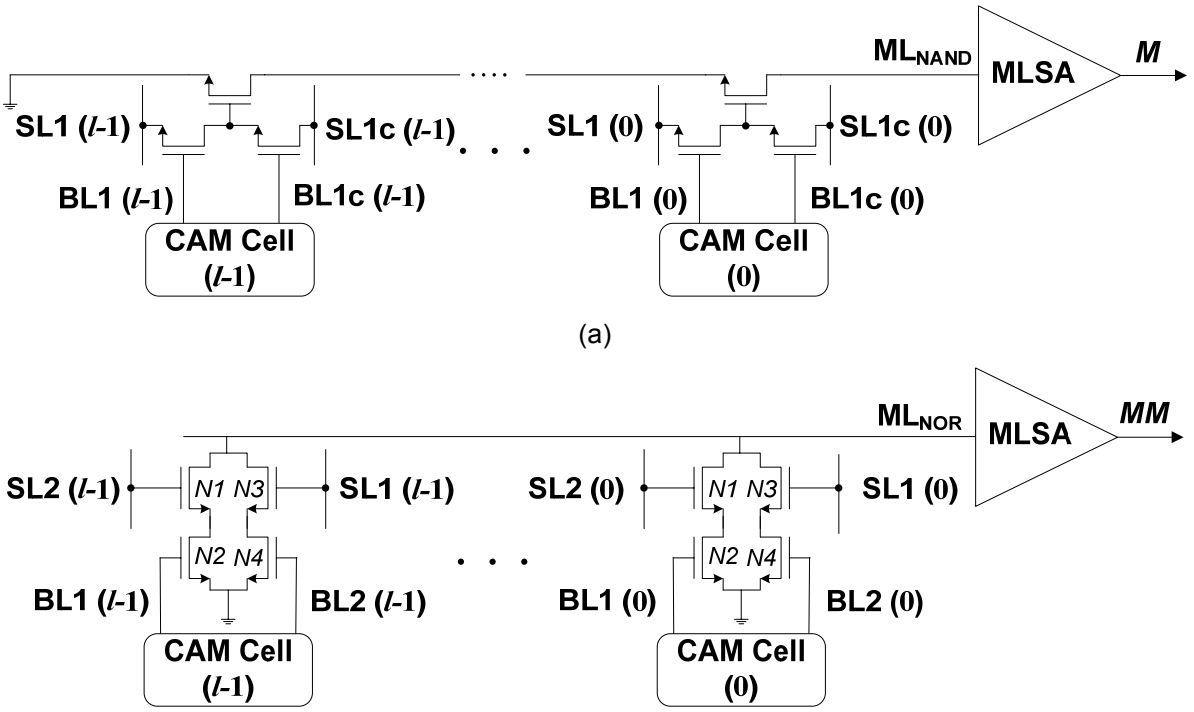


Figure 2.1: CAM "match" and "mismatch" logic implementation using (a) NAND- and (b) NOR-type logic, respectively

The low-power operation of a NAND-type CAM comes at the expense of slower evaluation due to the large number of series-connected transistors in the discharge path. In addition, the NAND-type CAM may cause a false "match" due to charge sharing among the internal nodes. For example, if there is a "mismatch" in CAM Cell (*l*-1) and a "match" in the remaining cells, all the series-connected transistors turn 'ON' except the one connected to the CAM Cell (*l*-1). As a consequence, the charge on the node ML_{NAND} is shared with all the internal nodes in the series-connected transistors. The charge sharing may decrease the ML_{NAND} voltage to a value less than the MLSA threshold causing a false "match". The

charge sharing problem can be alleviated by pre-charging the internal nodes to ($V_{DD} - V_{tn}$), where V_{tn} is the NMOS threshold voltage [30]. This can be done in the ML_{NAND} pre-charge phase by turning 'ON' all the series-connected transistors (SL1 = SL1c = '1' in Figure 2.1(a)). The pre-charging of the internal nodes increases the delay and energy consumption due to two main reasons. First of all, the charging and discharging of ML_{NAND} takes more time than usual because the internal nodes are also charged to ($V_{DD} - V_{tn}$). In addition, the partial discharge of the internal nodes (transistors close to GND i.e. towards the left side in Figure 2.1(a)) further increases the energy consumption. Second, the condition SL1 = SL1c = '1' makes the SEARCH cycle longer than usual due to the additional time needed to switch SLs. Moreover, the increased SL switching activity results in higher energy consumption. In order to achieve both low power and high performance, a mixed serial-parallel CAM has also been proposed [31]. However, its irregular structure in not suitable for high-density TCAMs due to area penalty and layout difficulties.

Two other variations of CAM comparison logic circuits are shown in Figure 2.2. The first circuit (Figure 2.2(a)) is a variant of the NOR-type CAM. It requires one less transistor

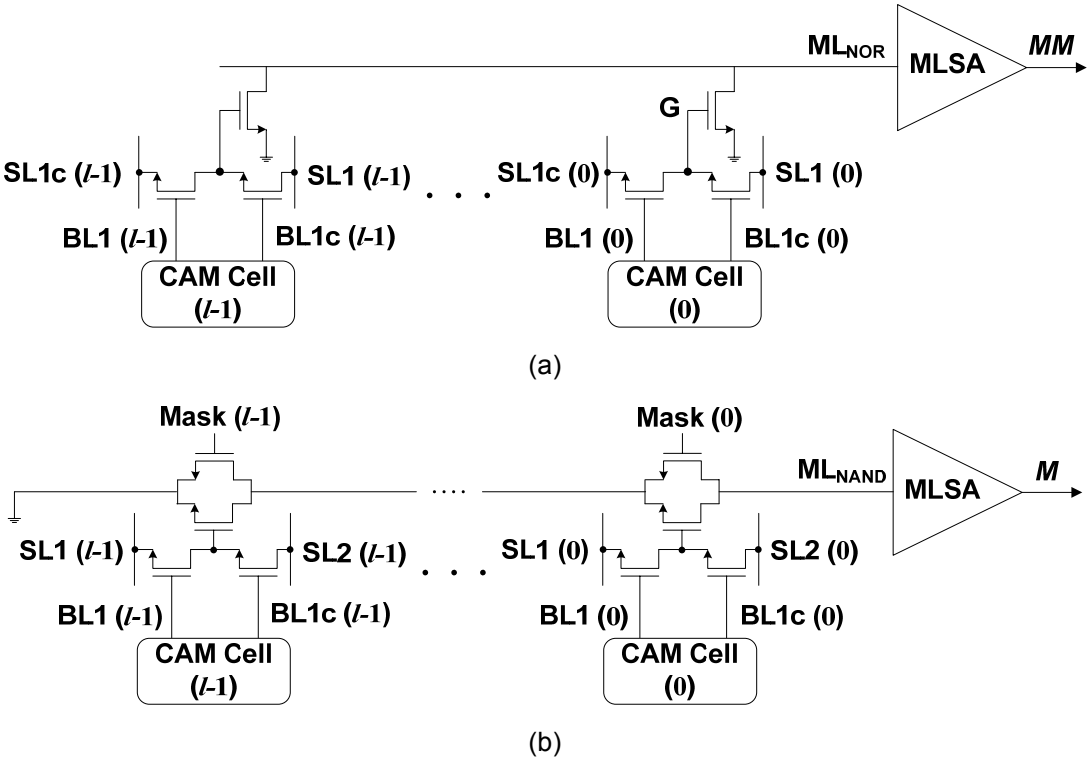


Figure 2.2: Cell variants of the (a) NOR- and (b) NAND-type CAM implementations

per cell than the NOR-type implementation shown in Figure 2.1(b). However, node 'G' (in Figure 2.2(a)) can rise to only ($V_{DD} - V_{tn}$), which increases the resistance of the pull-down path. Figure 2.2(a) is valid only for binary CAMs because it requires complementary BL1 and BL1c. Figure 2.2(b) shows a variant of the NAND-type implementation for TCAMs. The 'Mask' signal is supplied by the CAM cell to implement the local masking. The 'Mask' signal overrides the bit-level comparison result by shorting the series-connected transistor irrespective of the search key. The global masking can be achieved by ensuring SL1 = SL2 = '1'. Typically, the NOR-type circuit (shown in Figure 2.1(b)) is popular among the TCAMs designed for networking applications (typical word size l = 144).

2.2. Low-Area TCAM Cells

As mentioned in Chapter 1, a large-capacity TCAM chip is expensive partially due to the large cell area. A smaller TCAM cell can reduce the cost of a TCAM chip by improving the layout density. The 6T dynamic cell (described in Chapter 1) is relatively smaller but it requires a specialized embedded DRAM process. Hence, the static cells are more attractive due to their compatibility with the standard logic process.

Figure 2.3 illustrates two TCAM cells that are more area-efficient than the conventional 16T static TCAM cell. A 12T static TCAM cell (Figure 2.3(a)) reduces area by eliminating two access transistors and two driver transistors [32]. It maintains a '0' state at node 'S' by satisfying the following two conditions: (i) BLs are discharged to ground, and (ii) the N5 leakage is higher than the P5 leakage. The second condition is fulfilled under all process and temperature variations by keeping the WLs at a non-zero voltage ($V_{WL} \approx$ 200mV) [32]. 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. Figure 2.3(b) shows a balanced 16T cell [33]. The layout of this cell is more compact than that of the conventional 16T cell because it has an equal number of PMOS and NMOS transistors.

In order to minimize the TCAM cell area, the transistors and interconnects must be laid-out at the minimum distance specified by the design rules. Although such a dense layout is area-efficient, it 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, they should be routed such that their parasitic capacitances are minimized. The inter-wire capacitances of SLs and MLs can be reduced by placing them equally apart from the other parallel lines. Further reduction in the line capacitance can be achieved by minimizing the wire-widths of SLs and MLs. However, the lines should be wide enough to avoid problems such as electromigration and poor signal integrity under the worst-case operating conditions [21].

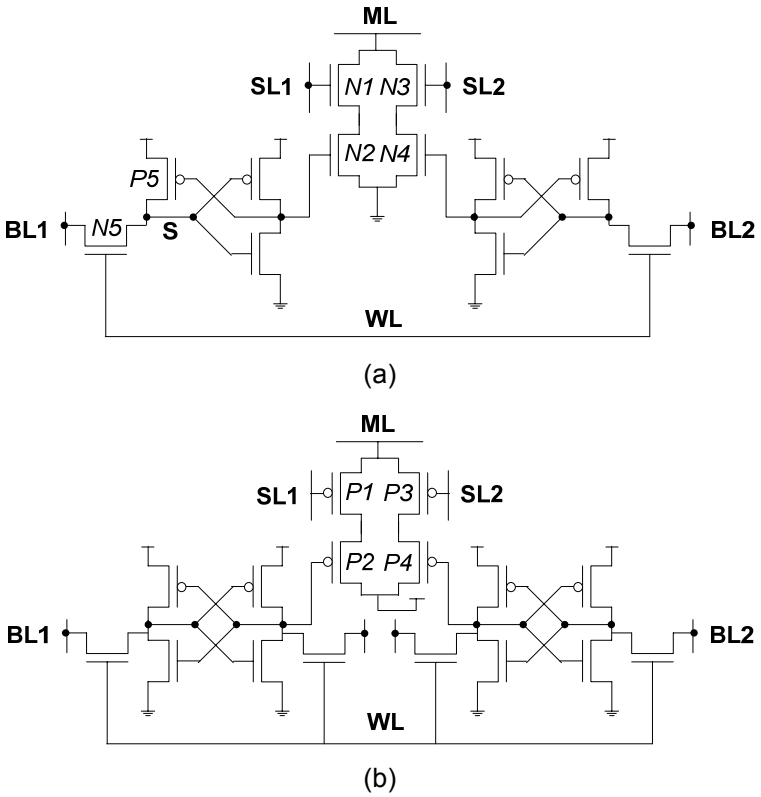


Figure 2.3: Low-area static TCAM cells: (a) 12T cell, and (b) balanced 16T cell

2.3. Power Reduction Techniques

As described earlier, TCAMs (particularly NOR-type) suffer from high power consumption. A number of techniques have been proposed in the past to reduce the power consumption in TCAMs. In most applications, TCAM activity is dominated by the parallel SEARCH operation. 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.

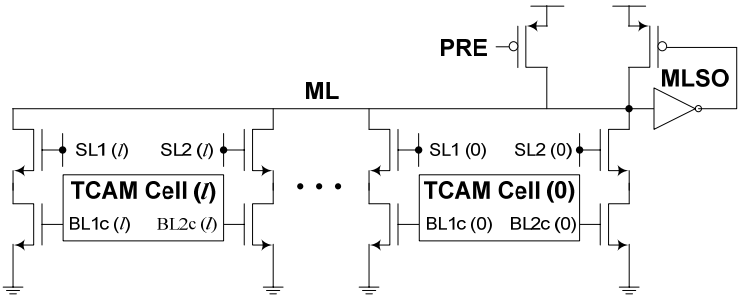
Increasing static power consumption is also becoming a serious issue for large-capacity TCAMs employing low-power architectures. Therefore, circuit and architecture innovations are needed to limit the increasing static power in TCAMs.

2.3.1. Match Line Sense Amplifiers

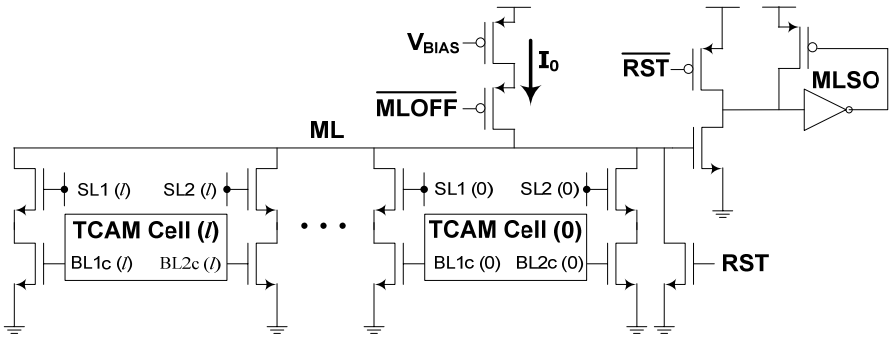
Most low-power MLSAs strive to minimize the ML voltage swing. Figure 2.4(a) illustrates the conventional MLSA described in Chapter 1. Initially, all the MLs are pre-charged to V_{DD}, and the search key is written on the SLs. If a TCAM word is identical to the search key, the ML remains at V_{DD}. Otherwise, it discharges to GND through mismatching cells. In order to avoid a short-circuit current, the SLs are switched to GND during the pre-charge phase. Hence, most of the SLs switch in every SEARCH operation, causing high power consumption.

Figure 2.4(b) shows a current-race sensing scheme [32]. This scheme has the ML connected to GND during the pre-charge phase, so the SLs can remain at their previous values. Thus, the average SL switching activity can be reduced approximately by half. This scheme achieves further power reduction by lowering the ML voltage swing. The ML sensing is initiated by charging up the ML using a constant current source. Since a matching ML does not have a current discharge path, it charges at a faster rate than a mismatching ML. When the matching ML charges to the NMOS threshold voltage (V_{tn}), its MLSO changes from '0' to '1' (Figure 2.4(b)). A dummy ML emulating the "match" condition generates an MLOFF signal to end the ML sensing.

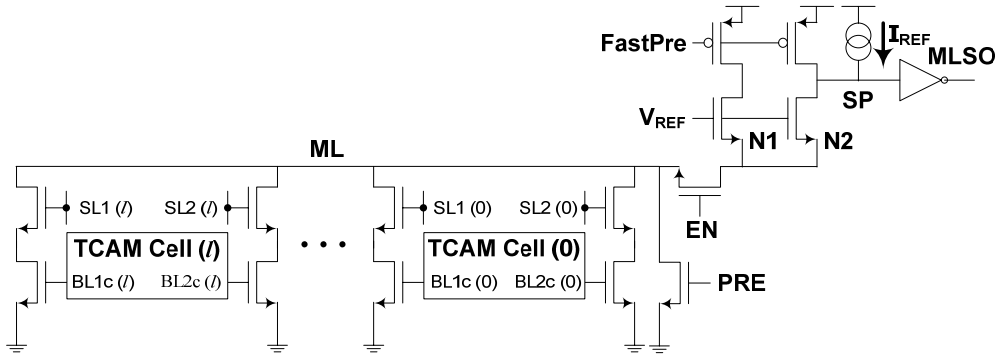
Figure 2.4(c) shows another MLSA that reduces the ML voltage swing using charge-redistribution [34]. This scheme also has the MLs connected to GND during the pre-charge phase. The ML sensing begins with fast pre-charging of the MLs using a FastPre signal. Transistors N1 and N2 restrict the ML voltage swing to ($V_{REF} - V_{tn}$). After the FastPre pulse, the MLs are left floating. Under the "mismatch" condition, the ML voltage drops below ($V_{REF} - V_{tn}$), and transistors N1 and N2 turn on. Transistor N2 equalizes the voltages of nodes ML and SP by redistributing charge at the two nodes (Figure 2.4(c)). A small current source (I_{REF}) feeds the node SP to compensate for ML leakages. The voltage V_{REF} can be varied to trade off power consumption with the speed of operation. This method can reduce



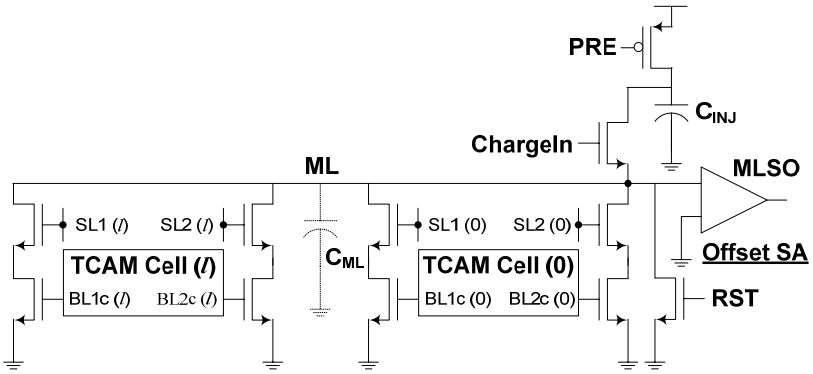
(a)



(b)



(c)



(d)

Figure 2.4: Match line sense amplifiers: (a) conventional precharge, (b) current-race, (c) charge-redistribution, (d) charge-injection

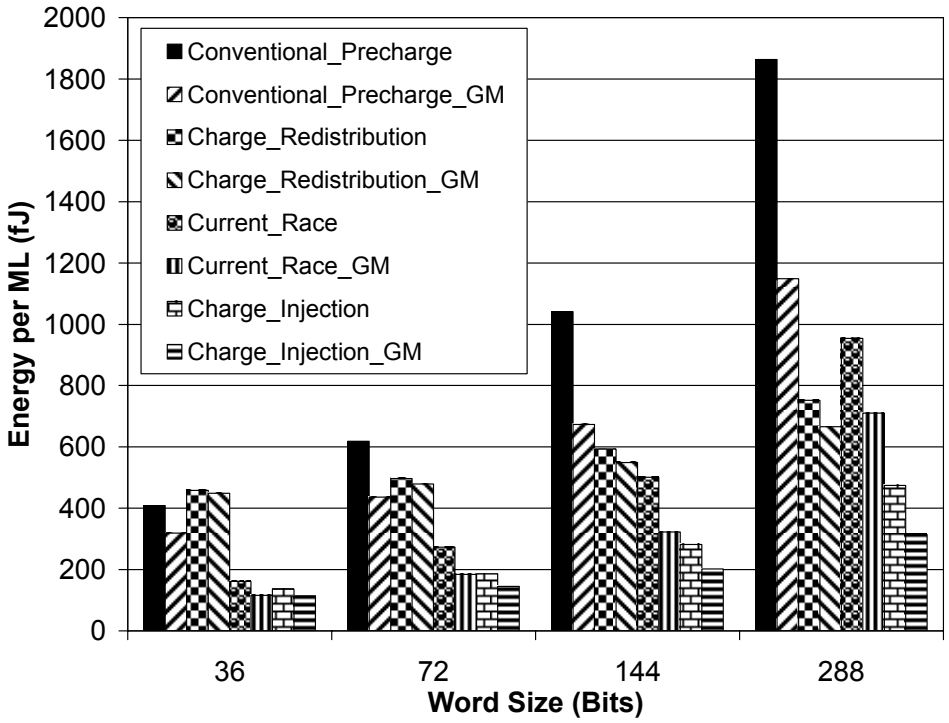
the ML voltage swing to even below V_t. However, the fast pre-charging of mismatching MLs causes short circuit power dissipation.

A charge-injection match detection circuit (CIMDC) eliminates this short circuit power (Figure 2.4(d)) [35]. CIMDC uses an injection capacitor (C_{INJ}) for each ML. Typically, C_{INJ} is sized 3-4 times smaller than C_{ML} [35]. 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 the ChargeIn signal (Figure 2.4(d)). Under the "match" condition, the voltage of C_{ML} rises to a voltage determined by the ratio of C_{INJ} and C_{ML}. Under the "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 lower noise margin and area penalty due to C_{INJ}.

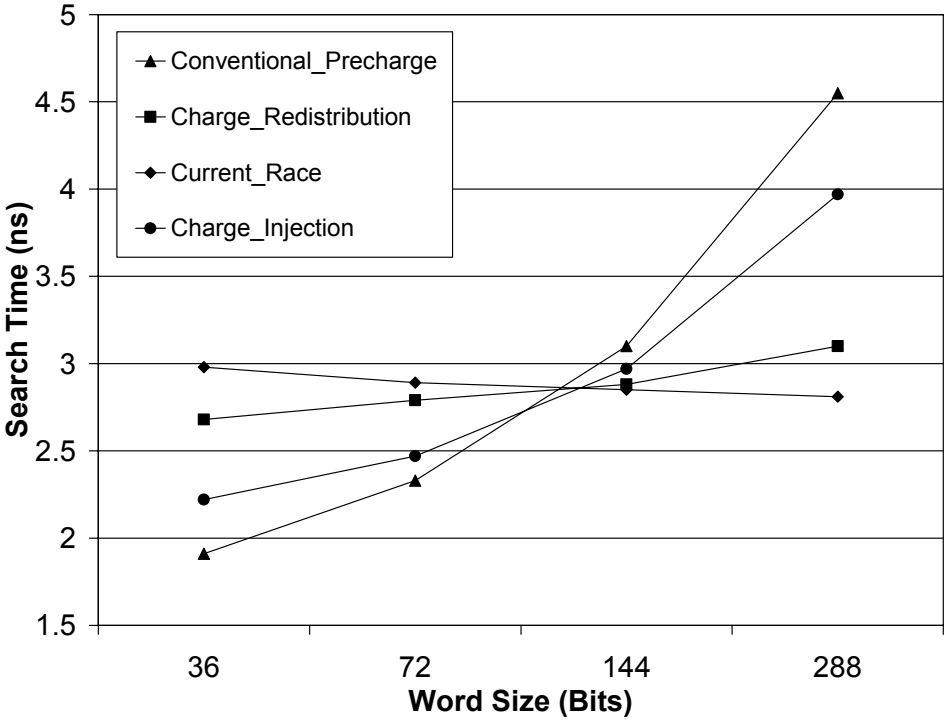
Figure 2.5 shows the delay and energy consumption of the above ML sensing schemes for different word sizes when they are simulated in 0.18µm CMOS technology. Global masking (GM) also alters the delay and energy by changing the ML capacitance. The ML capacitance can be given by equation (2.5):

$$C_{ML} = [2g + 4(l - g)]C_{DRAIN} + C_{INT} + C_{MLSA} \qquad (2.5)$$

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, C_{INT} is the interconnect capacitance of each ML, and C_{MLSA} is the MLSA input capacitance. Like the first term in equation (2.5), C_{INT} is also proportional to l. However for large values of l, C_{MLSA} is negligible as compared to the first two terms. When a bit is globally masked (SL1 = SL2 = '0'), only the drain capacitances of transistors N1 and N3 (shown in Figure 2.1(b)) 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 ($g = l$). Figure 2.5(a) shows the energies of operation for both extremes. The search speed in Figure 2.5(b) corresponds to the worst case. All MLSAs have the same (1ns) precharge (or reset) duration for fair comparison. We used C_{INT} = 0.18fF/cell from the post-layout extraction of a TCAM layout with MLs routed in metal 4 (0.18µm CMOS process). Also, C_{INJ} is sized to be one-third of C_{ML}.



(a)



(b)

Figure 2.5: (a) Energy per match line sensing and (b) search time for four alternative match line sense amplifiers

Figure 2.5 shows that ML sensing energy and search time increase with word size due to the increasing C_{ML}. The search speed remains almost constant for the current-race sensing scheme because the current sources are also scaled with the word size. Similarly, the search speed of the charge-redistribution scheme is also constant because the speed is governed by the capacitance of node SP, which does not change with word size (Figure 2.4(c)). Figure 2.5(a) affirms that the charge-injection scheme is the most energy efficient technique for the given range of word sizes. However, a smaller noise margin and a larger 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 NOR-type 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 the global masking.

It should be noted that equation (2.5) 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 Figure 2.1(b)). The capacitance of the internal nodes (N1-N2 and N3-N4 in Figure 2.1(b)) 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.

2.3.2. Match Line Segmentation

So far, it has been assumed that all the bits of a word share the same ML. The power consumption of ML sensing can be significantly reduced by segmenting MLs. One of the most popular ML-segmentation techniques is selective-precharge [36]. Several variations of the original scheme have been widely used in industry. A conventional TCAM performs a SEARCH operation in one step for all the bits as shown in Figure 2.6(a). The selective-precharge scheme divides the SEARCH operation into multiple stages. Figure 2.6(b) illustrates the most common implementation of this scheme using 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 the incoming data (search key). For example, a selective-precharge TCAM designed for networking applications with $l = 144$ and $k = 36$ can save up to 75% of the ML power, where *l* is the total number of bits per word. A recent design further extends the original selective-precharge scheme by dividing each ML into five segments, which also enables the use of hierarchical SLs as explained in the next subsection [37].

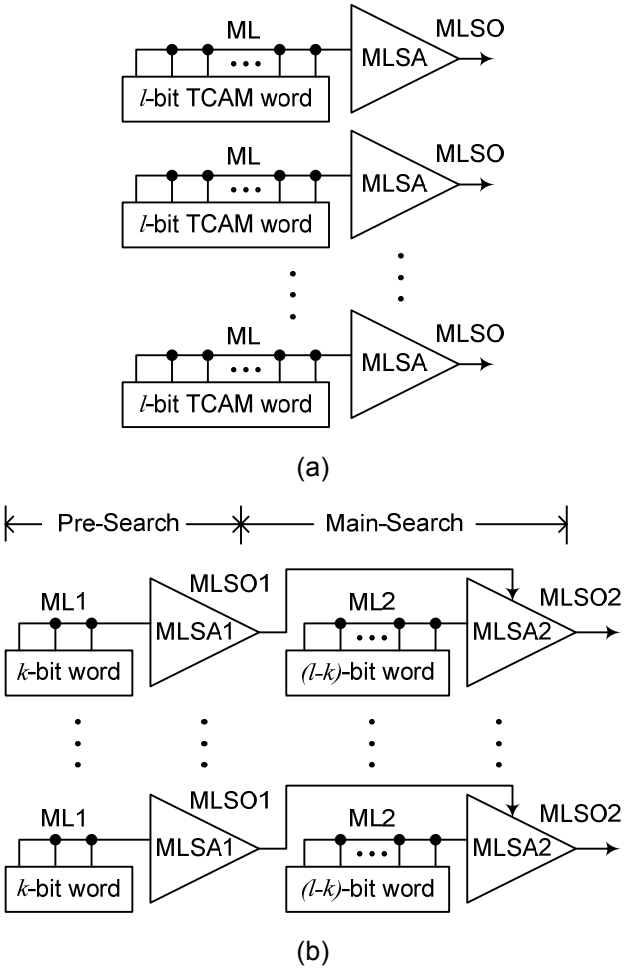


Figure 2.6: Match line segmentation: (a) conventional TCAM, (b) selective-precharge TCAM

2.3.3. Search Line Drivers

A significant portion of the TCAM power is also consumed by SL drivers in switching highly capacitive SLs. The SL switching activity depends on the incoming data statistics. For random data, almost half of the SLs are switched in every SEARCH operation. Thus, a significant amount of power can be saved by reducing the voltage swing of SLs. It can be shown from Figure 2.1(b) that a smaller SL voltage swing reduces the I_{ON}/I_{OFF} ratio of the ML pull-down paths. Therefore, most TCAM designs do not reduce the SL voltage swing. Some recently published designs extend the selective-precharge idea to SLs by dividing them into a two-level hierarchy of global SLs (GSLs) and local SLs (LSLs) as shown in Figure 2.7 [24][37]. These hierarchical SLs are implemented along with the ML segments described in section 2.3.2 [37]. For example, a group of 64 ML1-words can be defined as Block1, a corresponding group of ML2-words can be defined as Block2, and so on (shown by shaded boxes in Figure 2.7). During a SEARCH operation, GSLs broadcast the search key throughout the TCAM, but LSLs within a block are activated only when there is at least one "match" in the previous block. For example, the LSLs of Block2(*m*) will be activated only when at least one "match" is found in Block1(*m*).

In every SEARCH operation, only a few words match with the search key. Thus, most blocks will not contain even a single "match", and this scheme will save power by keeping the LSLs of these blocks inactive (Figure 2.7). The I_{ON}/I_{OFF} ratio of the ML pull-down paths is maintained by having a rail-to-rail voltage swing (1.8V) in LSLs. The power consumption is reduced by having a smaller voltage swing (0.45V) in GSLs [37]. The low-swing GSL signals are converted to the full-swing LSL signals using low-swing amplifiers. This scheme reduces the SL power consumption by 60% [37]. However, the power reduction comes at the expense of area overhead due to wide OR-gates (64-input), low-swing amplifiers, and other control circuits, which are embedded in the TCAM array. This scheme requires two separate power supply pins and an on-chip distribution network to support the low-swing GSL-drivers and the full-swing LSL-drivers. Since the area consumed by the power supply distribution network is not negligible, this scheme further reduces the effective on-chip area available for the core TCAM array. This scheme can be implemented only if the MLs are divided into multiple segments and the incoming data is searched sequentially. This constraint also degrades the search speed.

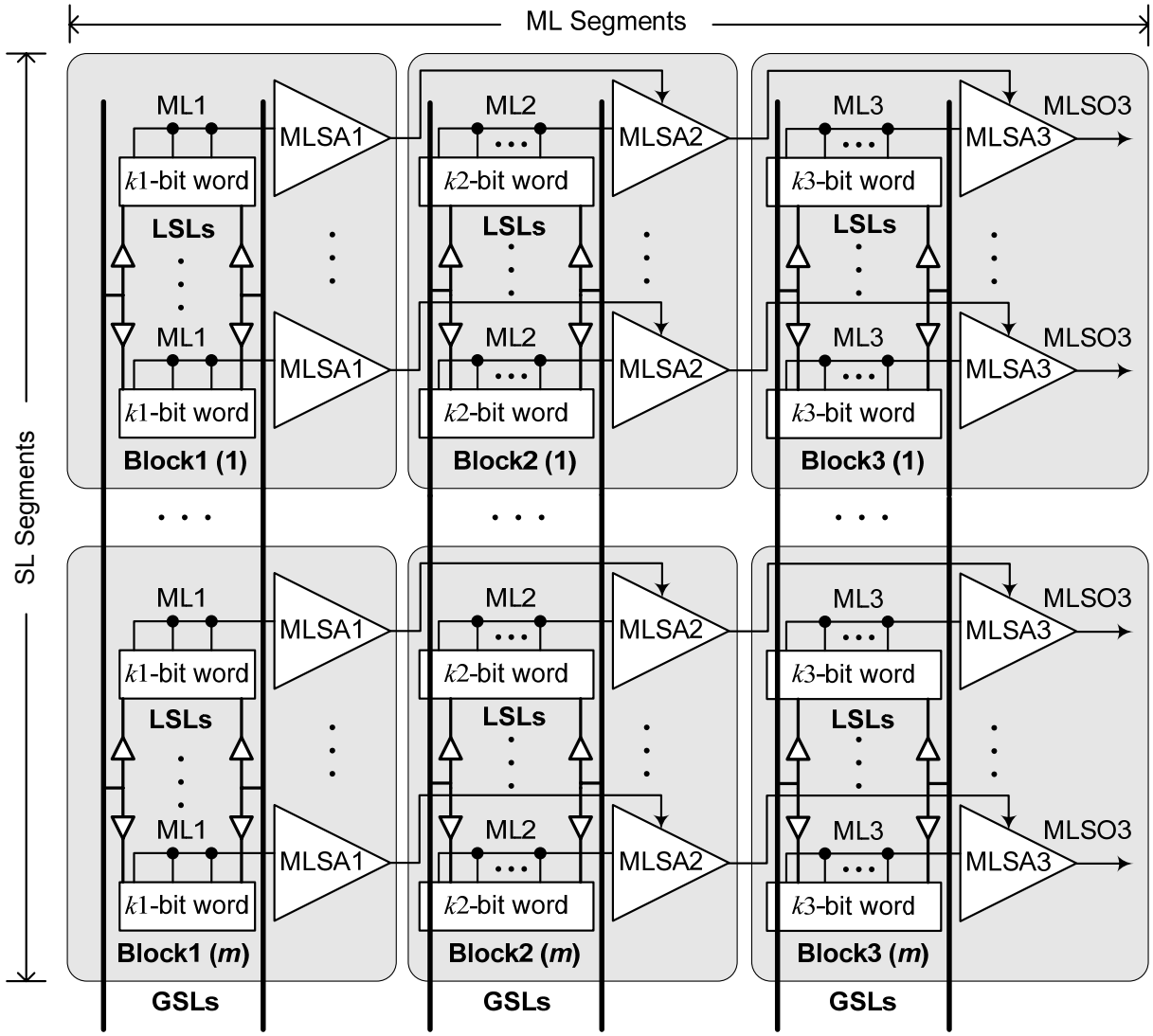


Figure 2.7: Hierarchical search line scheme

2.3.4. Low-Power CAM Architectures

So far, we discussed only circuit techniques for designing low-power CAMs. Several low-power CAM architectures have also been proposed over the years showing dramatic power reduction in specific applications. Although the focus of this work is to develop application-independent low-power CAM circuits, CAM architectures also deserve a mention because they can yield significant power savings in specific applications. In addition, some CAM architectures reduce the dynamic power so much that the static power becomes a considerable portion of the total power. Hence, the static power consumption, which has been largely ignored in CAMs, is also becoming important.

In this section, we will discuss only some recent low-power CAM architectures. Other architectures can be found in [38][39]. A popular CAM architecture is bank-selection, which divides the whole TCAM in multiple banks, and enables only one bank by decoding certain bits of the search-key [35]. For example, if a TCAM is divided in 8 banks and only one of them is enabled at a time, the power consumption is reduced to $1/8^{th}$ of the original value (i.e. without the bank-selection). However, depending on the data statistics, this scheme may cause one bank to overflow while the others have empty locations available. Hence, this scheme does not use the storage capacity efficiently. The overflow problem can be alleviated by re-partitioning the banks periodically. Some algorithms have been reported that effectively partition the input data space in packet-forwarding applications [40][41].

Another such scheme, suitable for binary CAMs, performs pre-computation on each table entry and stores this extra information along with the table entries [42]. When a search key is applied, the pre-computed information of the search key is first compared with the corresponding information of the table entries. Then only those entries are compared with the search key whose first comparison results in "match". For example, a pre-computation circuit can count the number of '1's in a word. Since there can be (l+1) '1's count in an l-bit word, each word requires only $\log_2(l+1)$ extra bits to store this pre-computed information. Here, one extra count denotes the possibility of a word with all '0's. Typically, there are only a few table entries with equal number of '1's as the number of '1's in the search key. Hence, this scheme reduces the power consumption at the expense of additional circuitry and time-budget required for the pre-computation.

An application-specific CAM has also been reported for LZ data compression [43]. This CAM architecture reduces the power consumption by disabling the unnecessary comparisons. The redundancy is directly derived from the data compression algorithm. For example, each table entry in the LZ-CAM contains a flag bit, which indicates whether the word and its predecessors are still candidates for a match string. The LZ data compression algorithm sets a word's flag to '0' if any of the following two conditions hold: (i) the word does not match the input symbol, (ii) the flag of the word's neighboring predecessor is '0' just before the current comparison. Hence, the CAM needs to compare only those words whose neighboring predecessors' flags are '1' after the previous cycle. Experimental results of this scheme show about 80% power reduction over the conventional CAM [43]. However,

this technique is suitable only for binary CAMs performing LZ data compression. Similar architectures can be developed for other applications that can reduce power consumption by eliminating the redundant comparisons.

2.4. Issues with Large-Capacity TCAMs

Modern applications require large-capacity TCAMs to store and search large databases. As described in Chapter 1, the table and word sizes are getting larger in networking applications due to increasing number of routes and "IPv4 to IPv6 migration". Large-capacity TCAM arrays can be implemented as multiple banks of smaller arrays on the same chip. If the banks are activated in parallel, the speed penalty is minimal. The energy per SEARCH operation increases linearly with the TCAM storage capacity. Implementing wide TCAM arrays (larger word-size) is more challenging because the lower noise margin between "match" and "mismatch" degrades the reliability of ML sensing. Figure 2.8 illustrates this issue by depicting the ML-to-GND pull down currents for "match" (ML_0) and 1-bit "mismatch" (ML_1) conditions, where I_{OFF} is the leakage current of a transistor when it is turned 'OFF'.

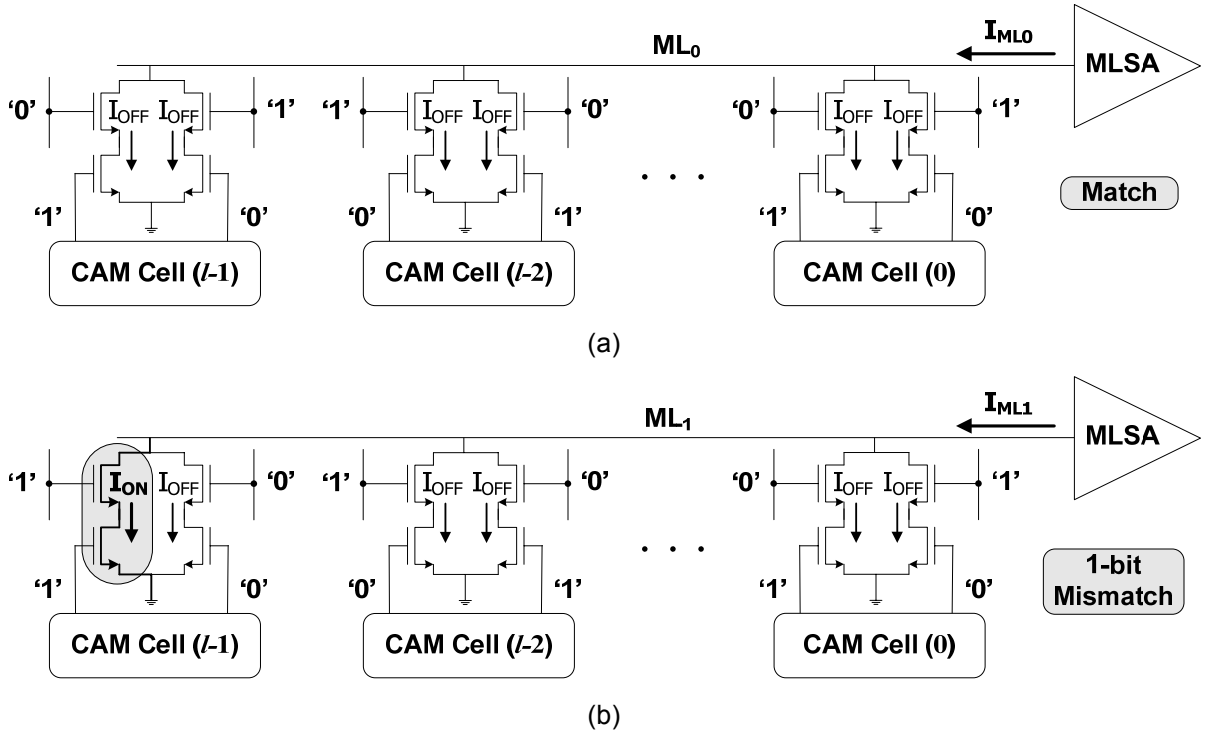


Figure 2.8: ML-to-GND pull-down currents for (a) "match" and (b) 1-bit "mismatch" conditions

From Figure 2.8, the total ML currents of an *l*-bit word for "match" and 1-bit "mismatch" (I_{ML0} and I_{ML1}, respectively) can be written as follows:

$$I_{ML0} = 2l \times I_{OFF} \tag{2.6}$$

$$I_{ML1} = I_{ON} + (2l - 1)I_{OFF} \tag{2.7}$$

Equations (2.6) and (2.7) highlight two important trends:

(i) As the word-size (*l*) increases, the leakage contribution in I_{ML0} and I_{ML1} also increases.

(ii) As the I_{ON}/I_{OFF} ratio decreases, the leakage contribution in I_{ML0} and I_{ML1} further increases.

Hence, it is becoming increasingly difficult to distinguish "match" from the 1-bit "mismatch" in TCAMs with large word-sizes. 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_{ON}/I_{OFF} ratio of the pull-down paths. For example, if a process technology offers multiple-V_t devices, the transistors with highest I_{ON}/I_{OFF} ratio should be used in the comparison logic. The I_{ON}/I_{OFF} ratio can be further improved using devices with non-minimum channel lengths [44]. There is also a growing need for innovative MLSAs to achieve reliable operation even for a small I_{ON}/I_{OFF} ratio of the ML pull-down paths.

2.5. Conclusions

In this chapter, we reviewed some of the existing CAM circuits and architectures. It can be observed that voltage swing reduction has been the most common circuit technique for TCAM power reduction. On the other hand, most low-power CAM architectures exploit the data statistics to reduce the chip activity. Most of the above mentioned techniques reduce the best-case or average power and ignore the worst-case or peak power. Peak power can be a serious issue in the board-level design because a board employing multiple TCAM chips must be able to support the total peak power consumption of all the components [39].

In the subsequent chapters, we will present novel circuit techniques that reduce TCAM energy and/or delay by reducing the line capacitance, V_{DD}, switching activity and voltage swing. One of our schemes also reduces the peak power, which has been largely ignored so far. We will discuss techniques to optimize the subthreshold and gate leakages in TCAMs. In addition, we will present two ternary storage cells for area and leakage reduction.

Chapter 3

Low-Power TCAM Cell Design

The core of a TCAM chip consists of TCAM cell arrays. A carefully designed TCAM cell can result in significant chip-level improvements such as smaller area, lower power and higher performance. A smaller TCAM cell can improve the storage capacity of TCAM chips by integrating more cells on the same die. Alternatively, it can reduce the TCAM manufacturing cost per unit storage capacity ($/Mb) by fitting the same number of cells in a smaller die. Since TCAM arrays occupy a major portion of a TCAM chip, they are the leading contributors of the static power. Conventionally, the static power has been ignored by TCAM designers because the parallel SEARCH operations activate the whole TCAM chip resulting in high dynamic power. However, the static power becomes important in TCAMs employing low-power architectures (described in Chapter 2) due to a significant reduction in signal switching activity. The significance of static power is expected to increase further in sub-100nm CMOS technologies due to increasing transistor leakages.

In this chapter, we present two static power reduction techniques and a cell-level comparison logic, which makes the ML less capacitive than the conventional ML. In order to

reduce the static power in TCAMs, we propose a dual-V_{DD} technique that trades some of the MLSA noise margin for a smaller cell leakage. In addition, we present two ternary storage cells that exhibit less leakage and a smaller cell area than the conventional TCAM cell. As described in Chapter 2, most of the existing TCAM circuits save power by reducing the ML voltage swing, and no scheme has been proposed to reduce the ML-capacitance. In this chapter, we analyze the different components of ML capacitance and present a comparison logic that offers less ML capacitance than the conventional comparison logic.

3.1. Static Power Reduction in TCAMs

This section explains two dominant leakage currents in scaled MOS transistors, and presents techniques to reduce the overall leakage in TCAMs. In order to understand various trade-offs and trends associated with the static power, it is helpful to write an equation relating the static power with design/process parameters and operating conditions. The static power of a CMOS circuit can be given by equation (3.1).

$$P_S = I_L \times V_{DD} \tag{3.1}$$

where I_L is the total leakage current of the circuit. In scaled MOS transistors, two dominant leakage currents are subthreshold leakage and gate leakage. The subthreshold leakage current of an NMOS transistor (between drain and source terminals) with zero gate-to-source voltage and full-swing drain-to-source voltage (V_{DS}=V_{DD}) can be given by equation (3.2) [45].

$$I_{SN} = I_0 \exp\left(-\frac{k_1\sqrt{\phi_S} - k_2\phi_S - \eta V_{DD}}{nV_T}\right) \tag{3.2}$$

where, $I_0 = \mu_0 C_{ox} (\frac{W_{eff}}{L_{eff}}) V_T^2 e^{1.8}$

ϕ_S = two times the Fermi potential

η = drain induced barrier lowering (DIBL) parameter

k_1, k_2 = non-uniform doping effect parameters

V_T = thermal voltage = kT/q = 26mV (at 300K)

μ_0 = mobility of electrons

L_{eff} = effective channel length of the transistor

W_{eff} = effective channel width of the transistor.

Equation (3.2) shows an exponential relationship between I_{SN} and V_{DD}. A reduction in V_{DD} decreases the subthreshold leakage current by reducing the drain induced barrier lowering (DIBL) [45].

As the CMOS technologies scale down to sub-100nm regime, gate oxide thickness is reduced to sub-20A° to achieve high drive currents and low DIBL effect. Such ultra-thin oxide results in a gate leakage current mainly due to two tunneling mechanisms: (i) Fowler-Nordheim (FN) tunneling, and (ii) direct tunneling. FN tunneling takes place when the voltage drop across the oxide (V_{ox}) is larger than the oxide potential barrier (Φ_{ox}) [46]. In FN tunneling, electrons tunnel through a triangular potential barrier. In the case of direct tunneling, V_{ox} is smaller than Φ_{ox}, and electrons tunnel through a trapezoidal potential barrier [46]. The current densities due to FN and direct tunneling are given by equations (3.3) and (3.4), respectively [47][48].

$$J_{FN} = AE_{ox}^{\ 2} e^{-\frac{B}{E_{ox}}} \tag{3.3}$$

$$J_{DT} = AE_{ox}^{\ 2}\left(\frac{\Phi_{ox}}{V_{ox}}\right)\left(\frac{2\Phi_{ox}}{V_{ox}} - 1\right) e^{-\frac{B\left[1-\left(1-\frac{V_{ox}}{\Phi_{ox}}\right)^{3/2}\right]}{E_{ox}}} \tag{3.4}$$

where, $A = \dfrac{q^3}{16\pi^2 \hbar \Phi_{ox}}$

$$B = \frac{4\sqrt{2m^*}\,\Phi_{ox}^{\ 3/2}}{3\hbar q}$$

V_{ox} = voltage drop across the oxide

E_{ox} = electric field in the oxide = V_{ox} / T_{ox}

T_{ox} = oxide thickness

Φ_{ox} = potential barrier height for electrons in the conduction band

m^* = effective mass of an electron in the conduction band.

ħ = reduced Planck's constant = 6.63×10^{-34} J.s

q = electron charge = 1.6×10^{-19} C

It can be observed from equations (3.3) and (3.4) that the gate leakage is a strong function of E_{ox}, which is dependent on V_{ox} and T_{ox}. Hence, a reduction in V_{DD} can reduce the gate leakage substantially.

3.1.1. Dual-V_{DD} Technique

In order to reduce dynamic power consumption, many high-speed digital systems adopt multiple-V_{DD} or multiple-V_t techniques. Most of these techniques carefully trade the excess speed for lower power consumption by reducing the V_{DD} of non-critical blocks. Since the static power is also strongly dependent on V_{DD}, a reduction in V_{DD} can also result in smaller static power. However, the V_{DD} reduction may also increase the propagation delay. Thus, the smaller V_{DD} should be applied only to components that are not speed critical, and the remaining components should use the normal V_{DD}.

As discussed in Chapter 1 (Figure 1.7), a TCAM has three main components: (i) TCAM cells, (ii) MLSAs, and (iii) SL drivers. Since majority of the static power is consumed by the TCAM cells, a reduction in the cell supply voltage (V_{DD_CELL}) can significantly reduce the static power. The SEARCH speed can be maintained by applying the normal V_{DD} to MLSAs and SL drivers. For example, if the current-race MLSA (described in section 2.3.1) is used, the timing signals are generated by a dummy word that emulates the match condition. Under the match condition, all ML-to-GND pull-down paths are turned 'OFF' (I_{ML0} in Figure 2.8), and they are not affected by the reduced V_{DD_CELL}. However, a smaller V_{DD_CELL} reduces the ML-to-GND pull-down current under the mismatch condition (I_{ML1} in Figure 2.8). A smaller I_{ML1}/I_{ML0} ratio reduces the noise margin of MLSA as described in section 2.4. Therefore, a dual-V_{DD} implementation (large V_{DD} and small V_{DD_CELL}) can achieve a smaller static power at the expense of a small reduction in the noise margin [49]. For the current-race MLSA, the noise margin is defined as the difference between the MLSA threshold voltage and the maximum voltage of ML_1 [32]. Higher noise margin can handle larger variations in process parameters and operating conditions. Figure 3.1 shows the noise margin variation with V_{DD_CELL} when the current-race MLSA is simulated with 144-bit TCAM word in 0.18µm CMOS technology. For $V_{DD\_CELL} \geq 1.3V$, the reduction in noise margin is less than 5%. Since the leakage decreases exponentially with V_{DD_CELL}, a 28% reduction in V_{DD_CELL} (1.8V → 1.3V) can result in a significant reduction in the cell leakage.

As described earlier, the static power in TCAMs has been a very small portion of the total power. For example, an 18Mb TCAM (fabricated in 0.13µm CMOS process) running at 250 Msps consumes nearly 15W at 1.2V [4]. We estimated the static power of this TCAM using 0.13µm CMOS predictive technology model (PTM) to 0.5W, which is only 3.3% of

the total power [50][51]. However, as highlighted in Chapter 2, many architectural innovations such as paged-TCAM and EaseCAM achieve significant power reduction by activating only a small portion of the TCAM [40][52]. For example, the paged-TCAM activates only $1/8^{th}$ or $1/64^{th}$ portion of a TCAM-based routing table [40]. TCAMs employing these techniques can have a significant leakage contribution in the total power because only the active portion is consuming the dynamic power while all cells are leaking. This contribution is expected to increase further with technology scaling due to increasing transistor leakages.

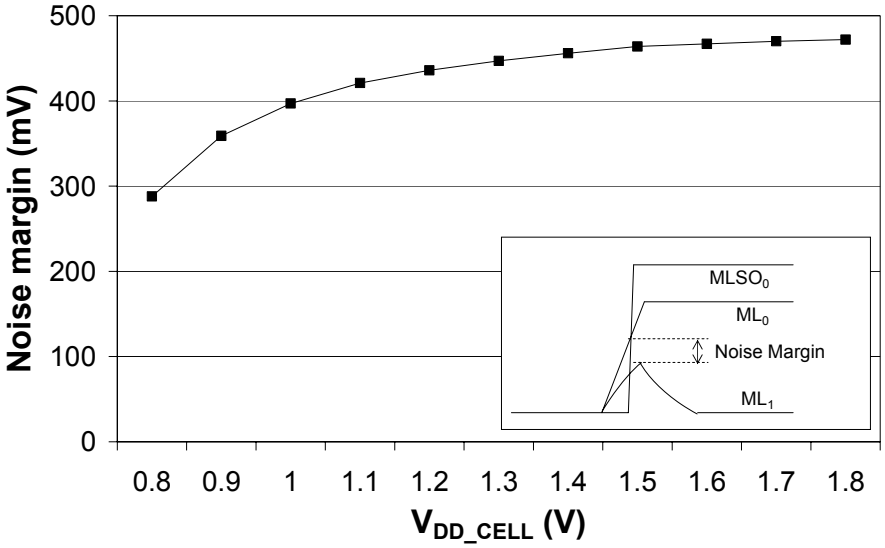


Figure 3.1: Noise margin variations with V_{DD_CELL}

Figure 3.2 shows total (both subthreshold and gate) leakage of the conventional TCAM cell simulated for various V_{DD_CELL} and technology nodes using PTM [53]. It can be observed that the leakage versus V_{DD_CELL} graph is a straight line on a log-scale. Thus, the leakage increases exponentially with V_{DD_CELL}. Moreover, the slope of leakage-V_{DD_CELL} graph increases as the technology feature size decreases. Hence, the dual-V_{DD} technique becomes more effective for smaller feature sizes. For example, a TCAM cell designed in 130nm technology exhibits 49% less leakage when V_{DD_CELL} is reduced from 1V to 0.5V. On the other hand, a cell designed in 90nm technology shows 73% less leakage when V_{DD_CELL} is reduced from 1V to 0.5V. Further reduction in the TCAM static power can be achieved by careful cell design as described in the next subsection.

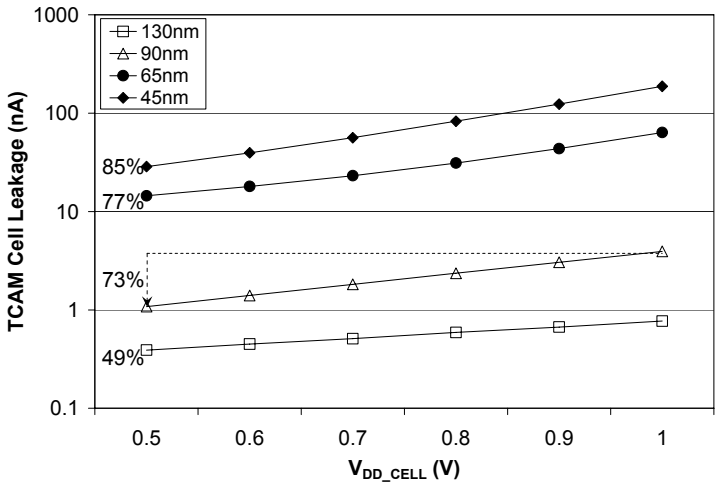


Figure 3.2: TCAM cell leakage for different technology nodes at different values of V_{DD_CELL}

3.1.2. Low-Leakage TCAM Cells

Each commercially available TCAM chip can have several million TCAM cells. Hence, cell-level leakage and area optimizations can yield significant reduction in overall chip leakage and die area. As described in Chapter 1, the storage portion of the conventional static TCAM cell is made of two 6T-SRAM cells, which are attractive for fast READ and WRITE operations due to the availability of complementary BLs. However, TCAM applications do not require very high-speed READ or WRITE, and the conventional 6T-SRAM is an over-design for TCAMs. For example, a TCAM performs WRITE operations only when the table is updated. In networking applications, the table update rate is less than 2000 updates per second [52]. Although the update rate is expected to increase in future networks, it will still be significantly less than the table lookup or SEARCH speeds [52]. Similarly, a TCAM performs READ operations only during the test phase. Hence, the performance of a TCAM is mainly determined by its SEARCH speed, and the speed of other operations (READ or WRITE) can be traded for reduced leakage and cell area.

3.1.2.1. Leakage in the 6T-SRAM-based TCAM Cell

Figure 3.3 shows the leakage paths in a 6T-SRAM-based TCAM cell when the BLs are precharged to the 'mask' state (BL1 = BL2 = '0'), and minimum-size transistors are used. The NMOS and PMOS subthreshold leakages are denoted by I_{SN} and I_{SP}, respectively. NMOS gate leakages are specified by I_{GON} and I_{GOFF} for 'ON' and 'OFF' transistors,

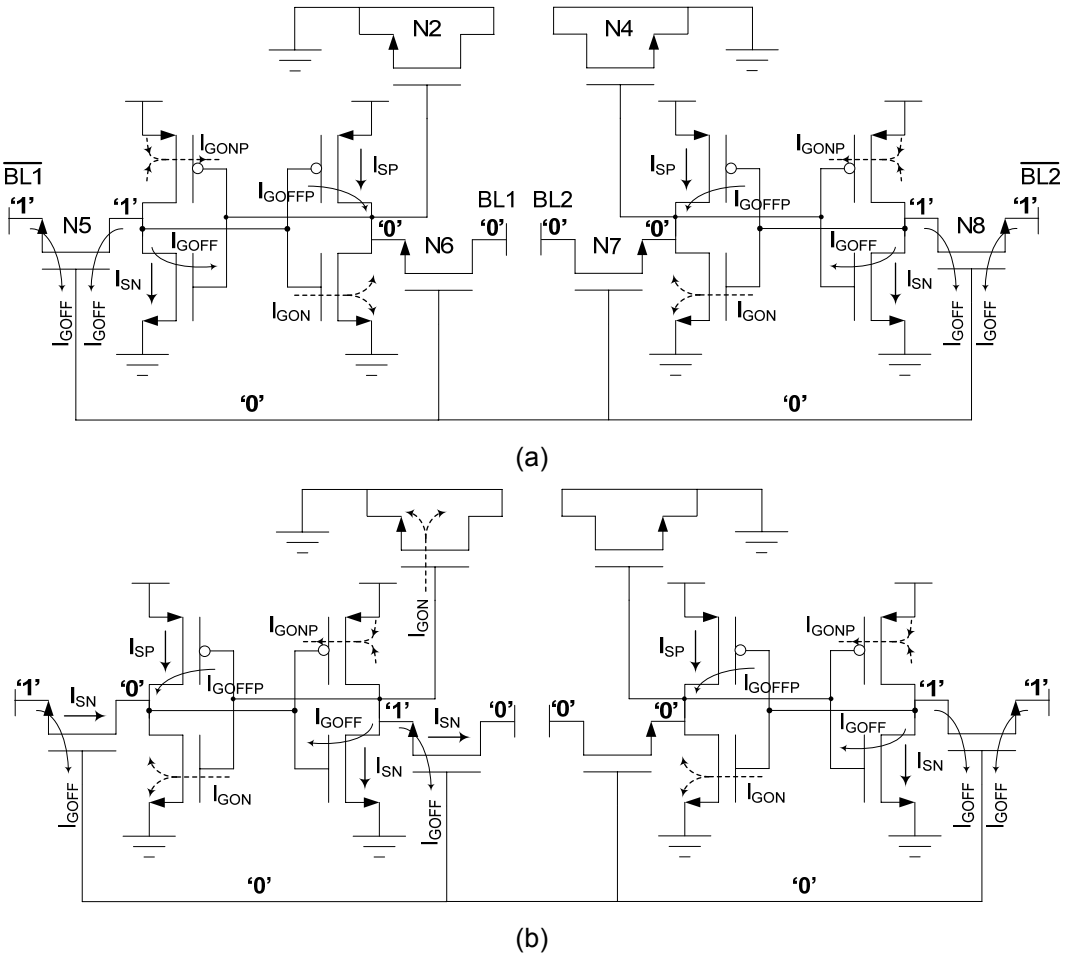


Figure 3.3: Leakage paths in the conventional TCAM cell when the stored value is (a) 'mask', and (b) '0' or '1'

respectively. Similarly, PMOS gate leakages are expressed by I_{GONP} and I_{GOFFP}. Assuming random data, a large TCAM column with shared BLs has the same probability of storing '0', '1' and 'mask' states. Hence, one-third of the bits will be masked and setting the BLs to the 'mask' state minimizes the average subthreshold leakage through the access transistors (N5 to N8). For example, if the BLs are set to '0' (BL1 = '0', BL2 = '1'), the subthreshold leakage through the access transistors will be $2I_{SN}$ when the stored value is 'mask' and $4I_{SN}$ when the stored value is '1'. In order to further substantiate the above assumption, we downloaded a recent routing table from PAIX router (Palo Alto, CA) and analyzed it [54]. We observed that the percentage of masked bits in the routing table was around 30%, which is reasonably close to the above assumption (one-third). Note in Figure 3.3 that the comparison logic transistors (N2 and N4) consume gate leakage only when a '1' or '0' is

stored. The subthreshold leakages through the comparison logic transistors are ignored because most modern match line sense amplifiers (MLSAs) reset both sides of the comparison logic to GND when they are idle (as described in Chapter 2). In order to simplify the analysis, the gate leakages of the two comparison-logic transistors connected to SLs (N1 and N3 in Figure 1.6) are not included because they are solely dependent on the SL data.

Typically, the driver transistors (NMOS in the cross-coupled inverters) are sized nearly 1.5 to 2 times larger than the access transistors to perform fast READ operation without disturbing the stored data. Larger transistors result in greater leakages. Since the READ speed is not critical in a TCAM, minimum size transistors can be employed. This choice also reduces the cell area. Conventional SRAMs also precharge BLs to V_{DD} in order to perform fast READ operation. In TCAMs, BLs can be precharged to the state, which results in the minimum leakage. As explained earlier, precharging the BLs to the 'mask' state minimizes the subthreshold leakages through the access transistors. Figure 3.3 can be used to calculate the total leakage current for a 6T-SRAM-based TCAM cell as given by equation (3.5) for different storage conditions.

$$I_{6T\_Mask}\Big|_{BLs=Mask} = 2I_{SN} + 2I_{SP} + 2I_{GON} + 6I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.5a)$$

$$I_{6T\_0/1}\Big|_{BLs=Mask} = 4I_{SN} + 2I_{SP} + 3I_{GON} + 6I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.5b)$$

$$I_{6T\_AVG}\Big|_{BLs=Mask} = 3.33I_{SN} + 2I_{SP} + 2.67I_{GON} + 6I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.5c)$$

where I_{6T_AVG} is the average leakage of this TCAM cell assuming equal probabilities of storing '0', '1' and 'mask'. Typically, the PMOS gate leakages are much smaller than subthreshold and NMOS gate leakages. It can be observed from equation (3.2) that subthreshold leakages increase rapidly with temperature. On the other hand, the gate leakages in equations (3.3) and (3.4) do not show any primary temperature dependence. Thus, if the junction temperature of a TCAM chip increases due to the high-power consumption, subthreshold leakage becomes the dominant leakage mechanism.

If the gate leakages are dominant, precharging the BLs to GND minimizes the overall leakage because the gate terminals of the access transistors (word lines or WLs) are also connected to GND. The total leakage in this case can be given by equation (3.6).

$$I_{6T\_Mask}\Big|_{BLs=GND} = 4I_{SN} + 2I_{SP} + 2I_{GON} + 4I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.6a)$$

$$I_{6T\_0/1}\Big|_{BLs=GND} = 4I_{SN} + 2I_{SP} + 3I_{GON} + 4I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.6b)$$

$$I_{6T\_AVG}\Big|_{BLs=GND} = 4I_{SN} + 2I_{SP} + 2.67I_{GON} + 4I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.6c)$$

Here, the gate leakage is reduced to $4I_{GOFF}$ because two gate leakage paths through the access transistors are eliminated. Comparing equations (3.5c) and (3.6c), we can write a relation between leakage components that determines the BL precharge condition for minimum leakage. Therefore, precharging the BLs to 'mask' value gives the minimum leakage only if condition (3.7) is satisfied. Otherwise, BLs should be precharged to GND for minimum leakage.

$$0.67I_{SN} > 2I_{GOFF} \Rightarrow I_{SN} > 3I_{GOFF} \quad (3.7)$$

READ Operation: If the 6T-SRAM-based cell employs minimum size driver transistors for area and leakage reduction, the READ operation requires slight modification so that it does not disturb the stored value. During the READ operation, instead of enabling WL with full voltage swing (V_{DD}), it should be enabled with a voltage small enough to keep the voltage V_x less than the inverter threshold in Figure 1.4. This condition can be fulfilled by applying a voltage ($V_{DD} - V_{tn}$) at WL during the READ operation. Although this modified WL voltage slows down the READ operation by reducing the BL pull-down current (I_{READ} in Figure 1.4), it is not an issue for TCAM applications as described earlier. A simple circuit to generate the modified WL voltage ($V_{DD} - V_{tn}$) during the READ operation (RE = '1') is shown in Figure 3.4. Since the V_{DD} is passed through an NMOS transistor whose gate terminal is also connected to V_{DD}, a voltage drop of V_{tn} is introduced at the output (WL_{NEW}). The relation between WL and WL_{NEW} can be expressed by equation (3.8).

If RE = '1' and WL= '1': $WL_{NEW} \approx (V_{DD} - V_{tn})$ (3.8a)

Otherwise: $WL_{NEW} = WL$ (3.8b)

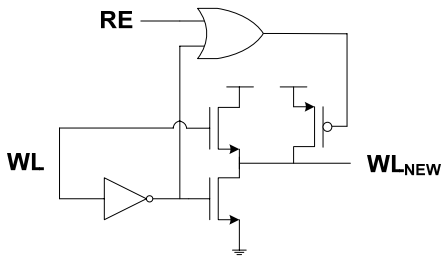


Figure 3.4: A simple circuit to reduce the WL voltage swing during the READ operation (RE = '1')

3.1.2.2. Leakage in the 5T-SRAM-based TCAM Cell

Typically, each static TCAM cell contains two 6T-SRAM cells. The conventional 6T-SRAM makes the TCAM cell relatively large, which is partially responsible for high manufacturing costs of the commercial TCAM chips. In order to reduce the TCAM cell area, alternative cells have been explored. A 4T-SRAM based TCAM cell (described in Chapter 2), achieves area reduction at the expense of a significant increase in leakage and unavailability of the READ operation [32]. Hence, the 4T-SRAM based cell is not suitable for low-power TCAMs, which may require the READ operation for chip testing. 5T cells have been reported for specialized SRAMs but they have not been adopted in TCAMs [55][56]. A 5T-SRAM cell consumes less leakage and smaller area than a 6T-SRAM cell. The slower READ and WRITE of 5T-SRAM due to the unavailability of the complementary BLs is not an issue for TCAM applications as explained earlier. Thus, 5T-SRAM is an attractive option for leakage and area reduction in TCAMs.

Figure 3.5 shows leakage paths in a 5T-SRAM-based TCAM cell when the BLs are precharged to the 'mask' state. As expected, it has fewer leakage paths than the conventional TCAM cell due to the removal of two access transistors. This choice also results in a smaller cell area. The total leakage current for a 5T-SRAM-based TCAM cell can be given by equation (3.9) for different storage conditions.

$$I_{5T\_Mask}\Big|_{BLs=Mask} = 2I_{SN} + 2I_{SP} + 2I_{GON} + 2I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.9a)$$

$$I_{5T\_0/1}\Big|_{BLs=Mask} = 3I_{SN} + 2I_{SP} + 3I_{GON} + 3I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.9b)$$

$$I_{5T\_AVG}\Big|_{BLs=Mask} = 2.67I_{SN} + 2I_{SP} + 2.67I_{GON} + 2.67I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.9c)$$

where I_{5T_AVG} is the average leakage of this TCAM cell. A comparison between equations (3.5c) and (3.9c) shows that the 5T-SRAM-based cell has less I_{SN} and I_{GOFF} leakage components than the 6T-SRAM-based cell due to the removal of two access transistors. It can be noticed in Figure 3.5 that each 5T-SRAM cell has the comparison logic transistor and the access transistor connected to the same node. This choice merges the two BL precharge options described in subsection 3.1.2.1. For example, if the BLs are precharged to the 'mask' value for minimum subthreshold leakage, the BLs are actually precharged to GND, which also minimizes the gate leakage. Hence, the same precharge value minimizes both subthreshold and gate leakages through the access transistors.

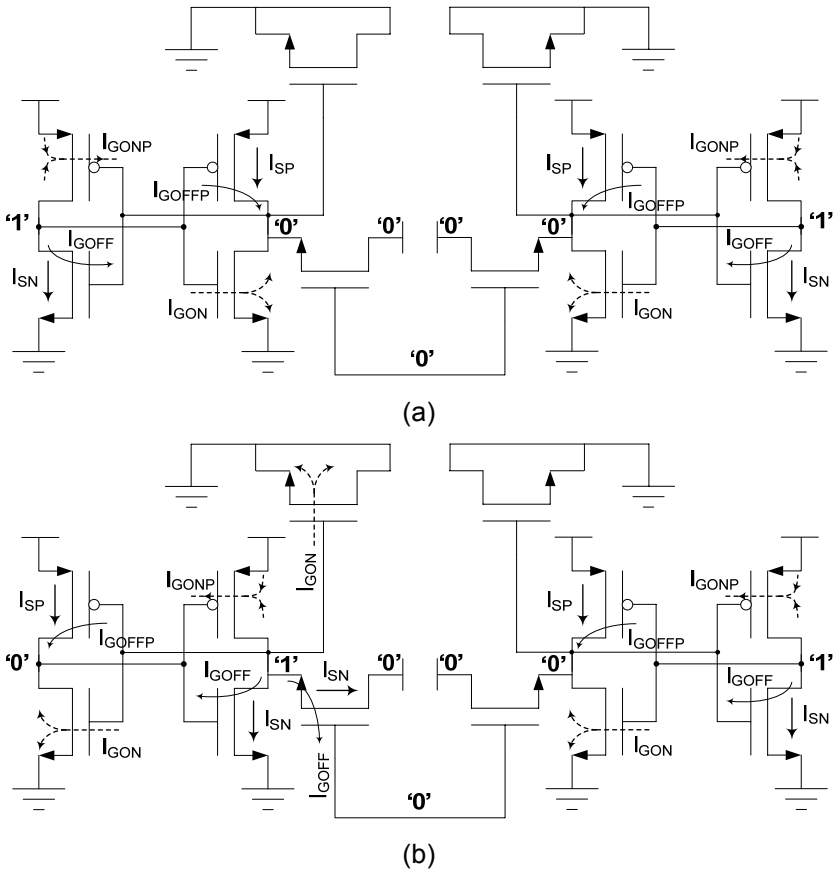


Figure 3.5: Leakage paths in the 5T-SRAM-based TCAM cell when the stored value is (a) 'mask', and (b) '0' or '1'

READ Operation: A 5T-SRAM cell requires a single-ended READ operation due to the unavailability of complementary BLs. The single-ended READ operation can be performed by a differential BLSA using a reference or dummy BL (DBL). If DBL is sized such that its capacitance is nearly two times the capacitance of a regular BL, the single ended BL sensing can be performed by using the BLSA shown in Figure 3.6 and connecting a cell with logic '0' to DBL. Initially, the BLSA is disabled (SAEN = '0'), and BLs are precharged to V_{DD} ($\overline{PRE}$ = '0'). Since transistors P1 and P2 are 'ON', nodes V_A and V_B are also precharged to V_{DD}. The feedback action of the cross-coupled inverters does not start because transistor N1 is still 'OFF'. The READ operation is initiated by disabling the precharge transistors ($\overline{PRE}$ = '1') and enabling WL of the selected cell with a reduced voltage swing ($V_{DD} - V_{tn}$) as described in section 3.1.2.1. Now if the selected cell has logic '1', BL remains at V_{DD}. However, DBL discharges due to the cell (with logic '0') connected to it, and a small

differential voltage develops between V_A and V_B ($V_A > V_B$). This differential voltage is amplified to rail-to-rail voltage ($V_A = V_{DD}$ and V_B = GND) by enabling the BLSA (SAEN = '1'). On the other hand, if the selected cell has logic '0', the cell current discharges the BL similar to the conventional READ operation (in Figure 1.4). Since the cell connected to DBL also has logic '0', DBL also discharges but at half the rate of BL discharging because DBL is twice more capacitive than BL. Hence, a small differential voltage develops between V_A and V_B ($V_A < V_B$) that is also amplified to rail-to-rail voltage (V_A = GND and $V_B = V_{DD}$) by enabling the BLSA. SAEN signal also isolates BL and DBL from the internal nodes by disabling transistors P1 and P2. Hence, the rail-to-rail voltage is avoided in the highly capacitive BL and DBL reducing the energy and delay of the READ operation.

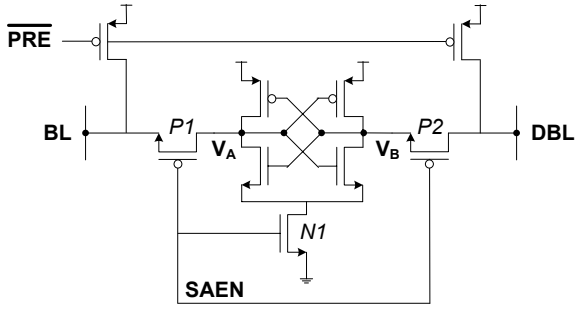


Figure 3.6: Bit line sense amplifier for single-ended READ operation using a dummy BL

WRITE Operation: The 5T-SRAM-based cell can be written with logic '0' (WRITE0) simply by enabling the WL (WL=V_{DD}) because an (NMOS) access transistor can easily override a load (PMOS) transistor even if they are both minimum size (Figure 1.3). However, writing logic '1' (WRITE1) is non-trivial in a 5T-SRAM-based cell. As described in Chapter 1 (Figure 1.3), the access transistor needs to be much larger (>10x) than the driver transistor for successful single-ended WRITE1 operation. This sizing is not only impractical in terms of area but it also makes the READ operation extremely difficult. Hence, alternative methods have been proposed [55][56].

The first method, illustrated in Figure 3.7, is adapted from [55]. The original scheme was designed for SRAMs and thus it used only a single SRC line for each column. In TCAM design, each column requires two lines (SRC1 and SRC2) to perform WRITE operation because each TCAM cell is made of two SRAM cells (see Figure 3.7). The comparison logic

transistors are not shown for clarity. This method disconnects the GND connection of the driver transistors (N1, N4, etc.) during the WRITE operation ($\overline{WE}$ = '0'). The GND connections of all the driver transistors in one column are tied to a single node (SRC1 for left-side cells and SRC2 for the right-side cells). During the WRITE operation, these nodes are left floating by turning 'OFF' N6 and N7. Otherwise, these transistors are 'ON' connecting SRC1 and SRC2 to GND. Figure 3.7 illustrates the WRITE1 operation in the first word (WL1 = '1'). Since the source terminal of N1 is not connected to GND, node SRC1 is charged through N1 and N2 until transistor N3 turns 'ON' and switches 'OFF' N1. Since there is always some voltage drop across N1 and N4, the unselected cell (WLn = '0') does not get disturbed (N5 remains 'OFF'). During READ and SEARCH operations ($\overline{WE}$ = '1'), SRC1 and SRC2 are connected to GND by turning 'ON' transistors N6 and N7.

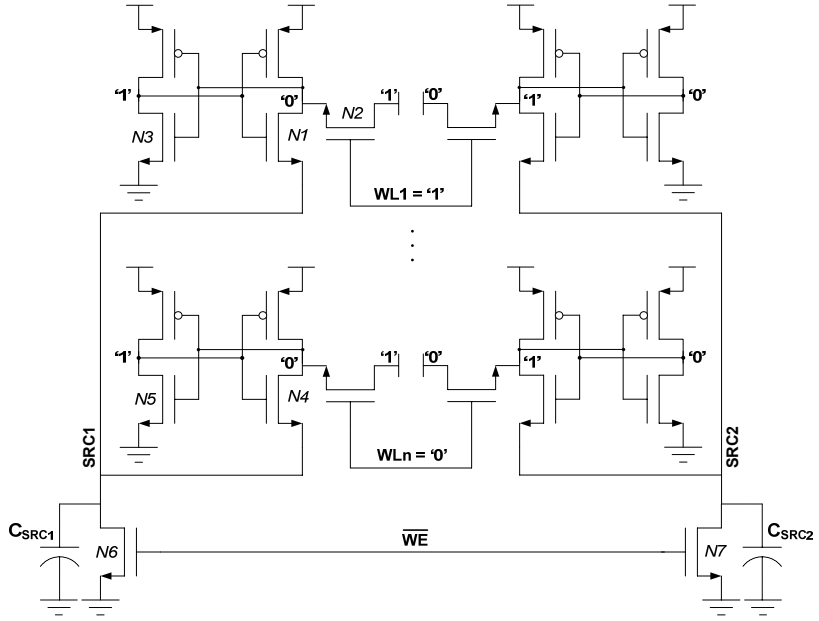


Figure 3.7: Method for writing '1' in a 5T-SRAM-based TCAM cell (adapted from [55])

The capacitance of the common source node (C_{SRC1} or C_{SRC2}) affects the reliability and delay of WRITE1 operation. Figure 3.8 shows the delay and voltage margin of the WRITE1 operation for different values of C_{SRC1} when simulated in CMOS 0.18µm technology. The voltage margin is defined as the difference between the threshold voltage of transistor N5 (V_{tn}) and the maximum SRC1 voltage. A large positive value of the voltage margin indicates a reliable operation. For large value of C_{SRC1}, it takes more time to charge

SRC1, and the delay of WRITE1 operation increases linearly. Since SRC1 charges at a slower rate for larger C_{SRC1}, the slower transient response improves the voltage margin to a certain extent, and further increasing C_{SRC1} saturates the voltage margin. Figure 3.8 also shows the effects of process variations. The WRITE1 operation performance is improved in fast N and slow P (FS) process corner.

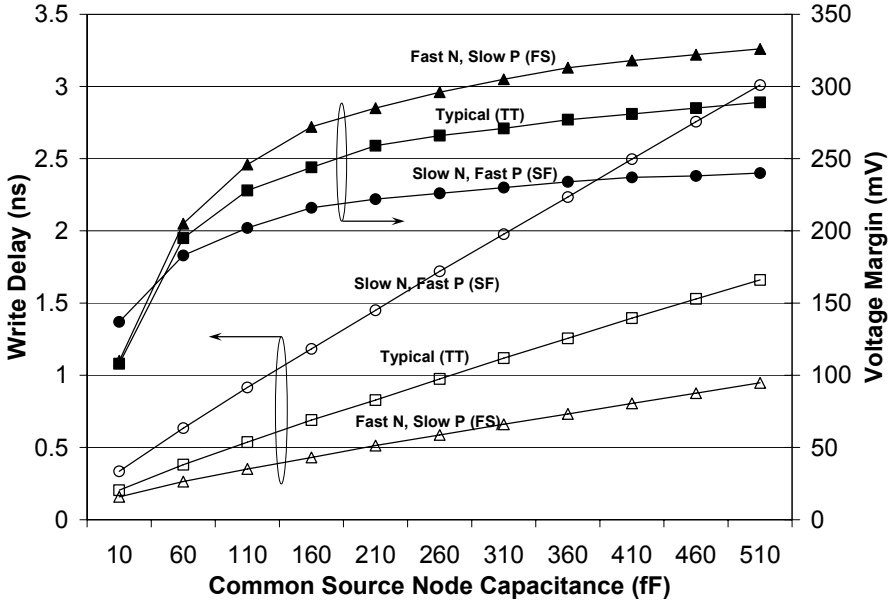


Figure 3.8: Effect of the common source node capacitance on delay and voltage margin

The second method for WRITE1 operation in 5T-SRAM-based cells is adapted from [56]. We extended the original idea of precharging SRAM cells (to 111…111 before they are written) to the TCAM cells. This method is illustrated in Figure 3.9. It connects half of driver transistors (N1, N4, etc.) in the same row to a common precharge line (PL1). A precharge cycle precedes the WRITE operation and sets the whole row to 111...111 by enabling its precharge line. For example, if row #1 needs to be updated, PL1 is enabled (PL1 = '1'). Since both sides of inverters P1-N1 and P2-N2 are now at V_{DD}, their internal nodes *A* and *C* are also pulled to V_{DD}. This turns 'ON' transistors N3 and N5 pulling nodes *B* and *D* to GND, which switches 'ON' transistors P1 and P2. As a consequence, nodes *A* and *C* remain at V_{DD} even after disabling PL1 (PL1 = '0'). Now the WRITE operation is performed by placing the data on BLs and enabling WL1 (WL1 = '1'). Since the access transistors (NMOS) can easily override the load transistors (PMOS), '0's are written in the desired cells,

and the remaining cells retain the precharge value (‘1’). This method provides a robust operation at the expense of additional delay and energy due to the precharge cycle. However, it is still attractive for TCAM applications, where the excess delay and energy of the WRITE operation are not critical. Therefore, two 5T-SRAM cells can be used to implement a 14T-TCAM cell that consumes less leakage and area than the conventional 16T-TCAM cell. In following subsections, we propose two 14T-TCAM cells that exhibit less subthreshold leakage than the 5T-SRAM-based TCAM cell.

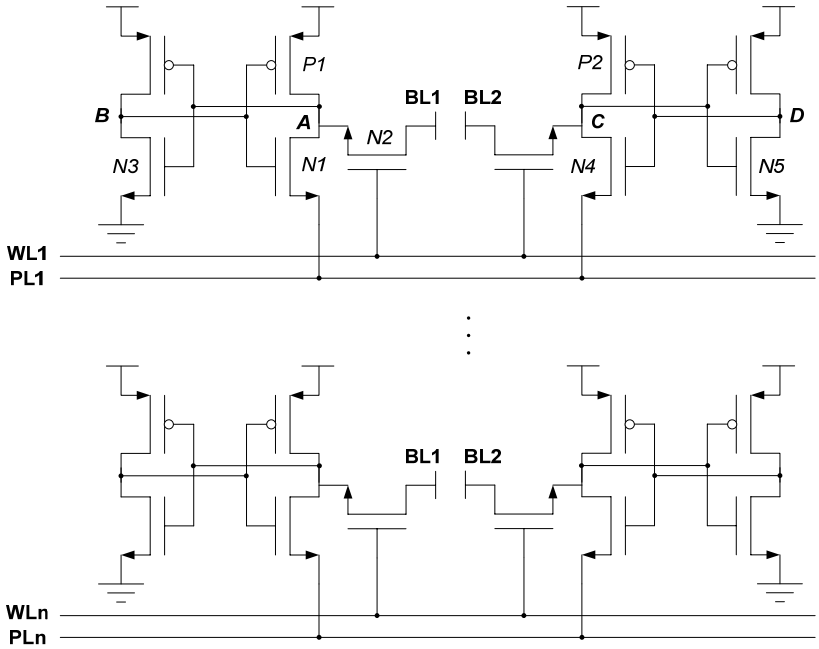


Figure 3.9: Alternative method for writing ‘1’ in a 5T-SRAM-based TCAM cell (adapted from [56])

3.1.2.3. NMOS-Coupled TCAM Cell

As explained earlier, each TCAM cell contains two SRAM cells to store the ternary value. These SRAM cells can have four combinations: ‘00’, ‘01’, ‘10’, and ‘11’. However, only three of them are used for the ternary value, and the “unused” state (typically ‘11’) is forbidden. We proposed a novel ternary storage cell that trades this “unused” state for a smaller leakage by coupling two 5T-SRAM cells and eliminating a subthreshold leakage path [57]. Figure 3.10 shows the leakage paths of the proposed NMOS-coupled (NC) ternary storage cell that connects two 5T-SRAM cells using NMOS transistors. For storing a ternary ‘0’ or ‘1’, one of the storage nodes (connected to the access transistors) is held at logic ‘0’

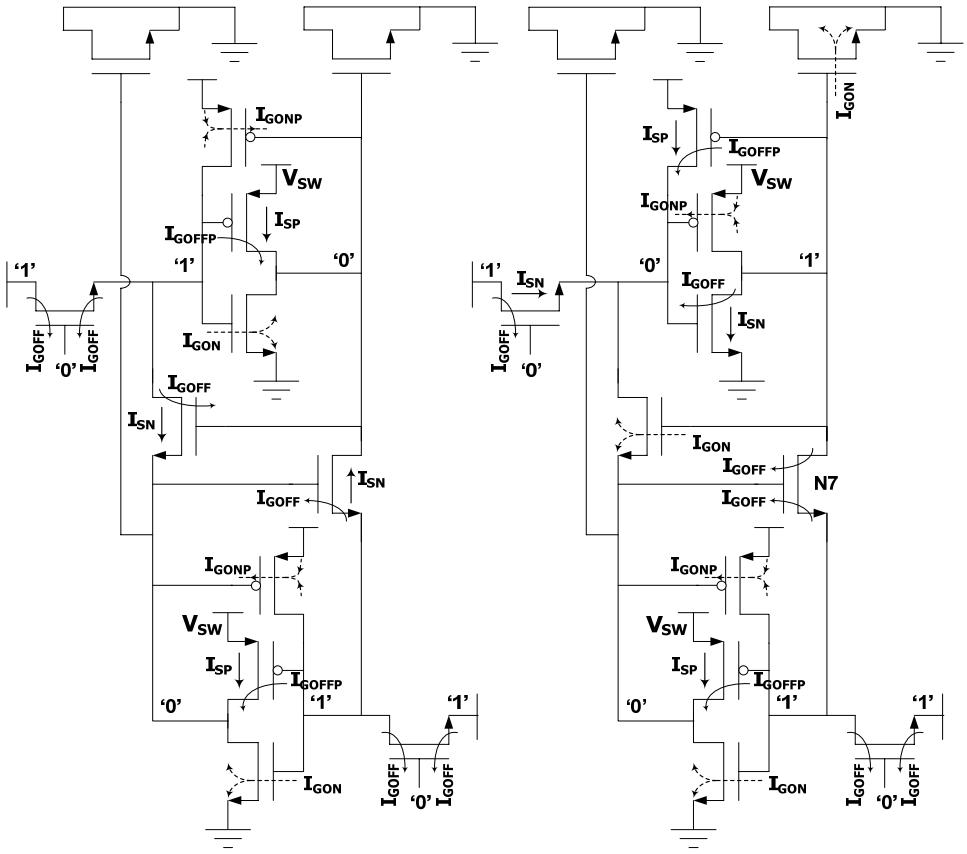


Figure 3.10: Leakage paths in the proposed NC-TCAM cell when the stored value is (a) ‘mask’, and (b) ‘0’ or ‘1’

through the coupling NMOS transistor. Similarly, when the ‘mask’ state is stored, both coupling NMOS transistors are ‘OFF’. Note that when both coupling NMOS transistors are ‘ON’, the cell is not stable. Hence, this cell can store only three states. In Figure 3.10, the BLs are precharged to the ‘mask’ condition to minimize the subthreshold leakage through the access transistors as explained in section 3.1.2.1. Under ‘0’ and ‘1’ conditions, one of the coupling NMOS transistors (N7) is ‘OFF’ but it does not contribute subthreshold leakage because its source and drain both are at logic ‘1’ (Figure 3.10(b)). However, the reduction in subthreshold leakage comes at the expense of additional gate leakage (I_{GOFF}) through transistor N7. Hence, this cell will exhibit smaller leakage than the 5T-SRAM-based cell if the subthreshold leakage is much larger than the gate leakage. This condition is easily satisfied at elevated temperatures, which is commonly the case for high-power TCAM chips. It can be noticed in Figure 3.10 that the proposed cell can store only three states because the

coupling does not allow the "unused" state. The total leakage current for an NC-TCAM cell (with BLs = 'mask') can be given by equation (3.10).

$$I_{NC\_Mask}\Big|_{BLs=Mask} = 2I_{SN} + 2I_{SP} + 2I_{GON} + 6I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.10a)$$

$$I_{NC\_0/1}\Big|_{BLs=Mask} = 2I_{SN} + 2I_{SP} + 3I_{GON} + 6I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.10b)$$

$$I_{NC\_AVG}\Big|_{BLs=Mask} = 2I_{SN} + 2I_{SP} + 2.67I_{GON} + 6I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.10c)$$

where I_{NC_AVG} is the average leakage of this TCAM cell. A comparison between equations (3.9c) and (3.10c) shows that the NC-TCAM cell has less I_{SN} and more I_{GOFF} leakage components than the 5T-SRAM-based cell. Thus, it will exhibit less leakage than the 5T-SRAM-based cell only if condition (3.11) is satisfied:

$$0.67I_{SN} > 3.33I_{GOFF} \Rightarrow I_{SN} > 5I_{GOFF} \quad (3.11)$$

If the gate leakage is comparable to the subthreshold leakage, pre-charging both BLs to GND minimizes the total leakage similar to the 6T-SRAM-based cell. For this condition, the total leakage current for the NC-TCAM cell can be given by equation (3.12).

$$I_{NC\_Mask}\Big|_{BLs=GND} = 4I_{SN} + 2I_{SP} + 2I_{GON} + 4I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.12a)$$

$$I_{NC\_0/1}\Big|_{BLs=GND} = 2I_{SN} + 2I_{SP} + 3I_{GON} + 4I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.12b)$$

$$I_{NC\_AVG}\Big|_{BLs=GND} = 2.67I_{SN} + 2I_{SP} + 2.67I_{GON} + 4I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.12c)$$

WRITE Operation: The WRITE operation can be performed on the NC-TCAM cell in two steps using the method shown in Figure 3.9. First, each bit of the row that needs to be updated is precharged to the 'mask' value. This is done by temporarily pulling node V_{SW} to GND. The node V_{SW} (shown in Figure 3.10) is shared by all the bits in one row, and it is normally connected to V_{DD}. Second, in order to write '1' or '0', the appropriate side of the NC-TCAM cell is pulled to GND by enabling the access transistor (NMOS) and overriding the load transistor (PMOS).

READ Operation: The READ operation can be performed on the NC-TCAM cell using a reduced swing WL voltage ($V_{DD} - V_{tn}$) as described in section 3.1.2.2. This method ensures that the stored data is not disturbed during the READ operation.

3.1.2.4. PMOS-Coupled TCAM Cell

The coupling between the two 5T-SRAM cells can also be achieved by PMOS transistors. Figure 3.11 shows the leakage paths of the proposed PMOS-coupled (PC) TCAM cell. Similar to the NC-TCAM cell, one of the coupling PMOS transistors does not consume subthreshold leakage under '0' and '1' conditions. Thus, it will also exhibit smaller leakage than the 5T-SRAM-based cell if the subthreshold leakage is much larger than the gate leakage. The total leakage current for a PC-TCAM cell (with BLs set to the 'mask' condition) can be given by equation (3.13).

$$I_{PC\_Mask}\Big|_{BLs=Mask} = 2I_{SN} + 2I_{SP} + 2I_{GON} + 2I_{GOFF} + 2I_{GONP} + 2I_{GOFFP} \quad (3.13a)$$

$$I_{PC\_0/1}\Big|_{BLs=Mask} = 3I_{SN} + I_{SP} + 3I_{GON} + 3I_{GOFF} + 2I_{GONP} + 3I_{GOFFP} \quad (3.13b)$$

$$I_{PC\_AVG}\Big|_{BLs=Mask} = 2.67I_{SN} + 1.33I_{SP} + 2.67I_{GON} + 2.67I_{GOFF} + 2I_{GONP} + 2.67I_{GOFFP} \quad (3.13c)$$

Similar to the 5T-SRAM-based cell, this BL precharge condition minimizes both subthreshold and gate leakages. A comparison between equations (3.9c) and (3.13c) shows that the PC-TCAM cell will consume less leakage than the 5T-SRAM-based cell only if condition (3.14) is satisfied.

$$0.67I_{SP} > 0.67I_{GOFFP} \Rightarrow I_{SP} > I_{GOFFP} \quad (3.14)$$

Similarly, a comparison between equations (3.10c) and (3.13c) shows that the PC-TCAM cell will consume less leakage than the NC-TCAM cell (when BLs = 'mask') only if condition (3.15) is satisfied.

$$0.67I_{SP} + 3.33I_{GOFF} > 0.67I_{SN} + 0.67I_{GOFFP} \Rightarrow I_{SP} + 5I_{GOFF} > I_{SN} + I_{GOFFP} \quad (3.15)$$

If the gate leakage is comparable to the subthreshold leakage ($I_{SN} < 3I_{GOFF}$) and BLs of NC-TCAM are at GND, the PC-TCAM cell will consume less leakage than the NC-TCAM cell if condition (3.16) is satisfied.

$$0.67I_{SP} + 1.33I_{GOFF} > 0.67I_{GOFFP} \Rightarrow I_{SP} + 2I_{GOFF} > I_{GOFFP} \quad (3.16)$$

Most CMOS processes will satisfy condition (3.16) because the PMOS subthreshold leakage and the NMOS gate leakage both are typically larger than the PMOS gate leakage.

WRITE Operation: The WRITE operation can be performed on the PC-TCAM cell using the method shown in Figure 3.7. The left and right sides of all cells in the same column are connected to nodes SRC1 and SRC2, respectively (Figure 3.11). SRC1 and SRC2 are

normally connected to GND. However during the WRITE operation, they are disconnected from GND, and the WRITE operation proceeds as illustrated in Figure 3.7.

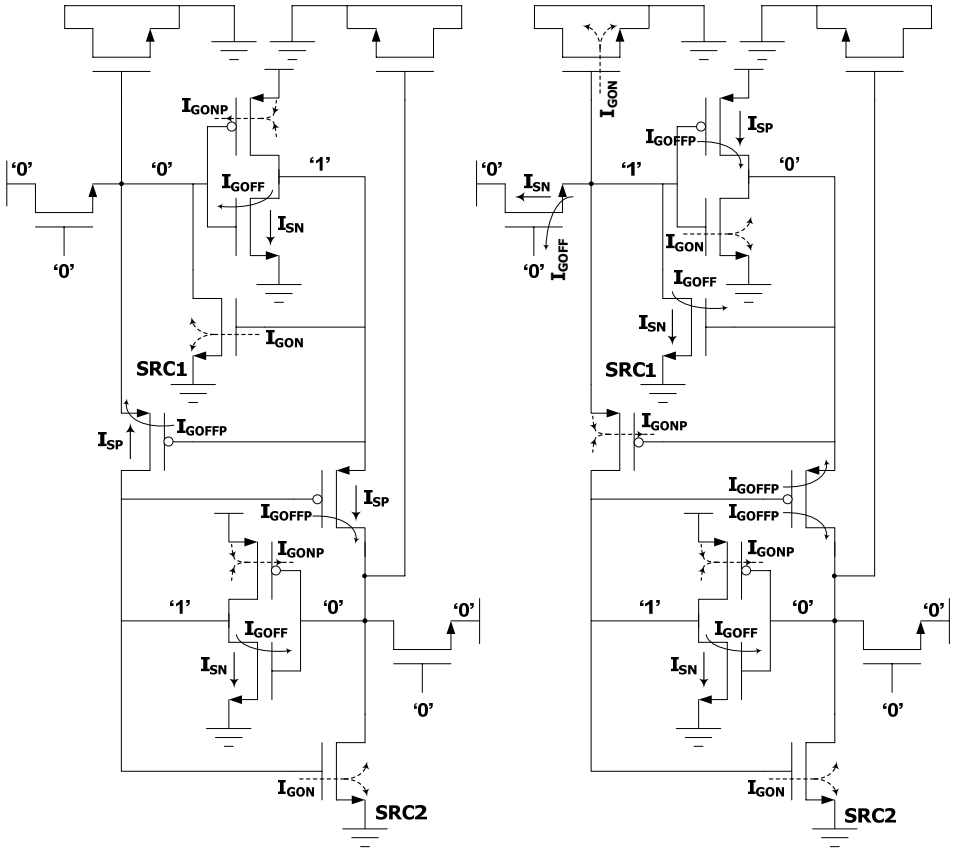


Figure 3.11: Leakage paths in the proposed PC-TCAM cell when the stored value is (a) 'mask', and (b) '0' or '1'

READ Operation: The READ operation can be performed on the PC-TCAM cell using a reduced swing WL voltage ($V_{DD} - V_{tn}$) as described in section 3.1.2.2.

The leakages of the above cells (when BLs = 'mask') are summarized in Table 3.1. The average currents of different cells are shown by I_{6T_AVG}, I_{5T_AVG}, I_{PC_AVG} and I_{NC_AVG}. Table 3.2 summarizes the leakage currents of NC-TCAM and 6T-SRAM-based cells when BLs = GND. The leakages of PC-TCAM and 5T-SRAM-based cells are not shown in Table 3.2 because they remain the same as Table 3.1. It can be shown from Table 3.1 and Table 3.2 that the 6T-SRAM-based cell always consumes more leakage than the other three cells under all conditions. Table 3.3 summarizes the conditions that can determine the minimum-leakage

TCAM cell. Condition #1 determines the BL precharge value for minimum leakage in 6T-SRAM-based cell and NC-TCAM cell. Table 3.3 can be used to determine the minimum-leakage TCAM cell in a given process technology from the relative magnitudes of the different leakage components.

Table 3.1: Leakage currents of TCAM cells when BLs = 'mask'

Cell	***Stored Value***	***Subthreshold Leakage***	***NMOS Gate Leakage***	***PMOS Gate Leakage***
6T	X	$2I_{SN} + 2I_{SP}$	$2I_{GON} + 6I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	0,1	$4I_{SN} + 2I_{SP}$	$3I_{GON} + 6I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	I_{6T_AVG}	**$3.33I_{SN} + 2I_{SP}$**	**$2.67I_{GON} + 6I_{GOFF}$**	**$2I_{GONP} + 2I_{GOFFP}$**
5T	X	$2I_{SN} + 2I_{SP}$	$2I_{GON} + 2I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	0,1	$3I_{SN} + 2I_{SP}$	$3I_{GON} + 3I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	I_{5T_AVG}	**$2.67I_{SN} + 2I_{SP}$**	**$2.67I_{GON} + 2.67I_{GOFF}$**	**$2I_{GONP} + 2I_{GOFFP}$**
PC	X	$2I_{SN} + 2I_{SP}$	$2I_{GON} + 2I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	0,1	$3I_{SN} + I_{SP}$	$3I_{GON} + 3I_{GOFF}$	$2I_{GONP} + 3I_{GOFFP}$
	I_{PC_AVG}	**$2.67I_{SN} + 1.33I_{SP}$**	**$2.67I_{GON} + 2.67I_{GOFF}$**	**$2I_{GONP} + 2.67I_{GOFFP}$**
NC	X	$2I_{SN} + 2I_{SP}$	$2I_{GON} + 6I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	0,1	$2I_{SN} + 2I_{SP}$	$3I_{GON} + 6I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	I_{NC_AVG}	**$2I_{SN} + 2I_{SP}$**	**$2.67I_{GON} + 6I_{GOFF}$**	**$2I_{GONP} + 2I_{GOFFP}$**

Table 3.2: Leakage currents of TCAM cells when BLs = GND

Cell	***Stored Value***	***Subthreshold Leakage***	***NMOS Gate Leakage***	***PMOS Gate Leakage***
6T	X	$4I_{SN} + 2I_{SP}$	$2I_{GON} + 4I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	0,1	$4I_{SN} + 2I_{SP}$	$3I_{GON} + 4I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	I_{6T_AVG}	**$4I_{SN} + 2I_{SP}$**	**$2.67I_{GON} + 4I_{GOFF}$**	**$2I_{GONP} + 2I_{GOFFP}$**
NC	X	$4I_{SN} + 2I_{SP}$	$2I_{GON} + 4I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	0,1	$2I_{SN} + 2I_{SP}$	$3I_{GON} + 4I_{GOFF}$	$2I_{GONP} + 2I_{GOFFP}$
	I_{NC_AVG}	**$2.67I_{SN} + 2I_{SP}$**	**$2.67I_{GON} + 4I_{GOFF}$**	**$2I_{GONP} + 2I_{GOFFP}$**

Table 3.3: Conditions to determine the minimum leakage cell

Condition #1	*BLs*	*Condition #2*	*Condition #3*	*Cell with I_{MIN}*
$I_{SN} > 3I_{GOFF}$	'X'	$I_{SN} > 5I_{GOFF}$	$I_{SN} + I_{GOFFP} > I_{SP} + 5I_{GOFF}$	NC
		$I_{SP} > I_{GOFFP}$	$I_{SN} + I_{GOFFP} < I_{SP} + 5I_{GOFF}$	PC
		$I_{SP} < I_{GOFFP}$	$I_{SN} < 5I_{GOFF}$	5T
$I_{SN} < 3I_{GOFF}$	GND	$I_{SP} > I_{GOFFP}$	-	PC
		$I_{SP} < I_{GOFFP}$	-	5T

3.1.2.5. Simulation Results

We simulated the above TCAM cells using PTM [50][51]. Figure 3.12 shows the average leakages of different TCAM cells in 90nm, 65nm, 45nm and 32nm CMOS technologies at 300°K. The leakage components are shown for 32nm technology in Figure 3.13. It can be noticed that the magnitudes of different leakage components are related as follows:

$$I_{SN} > I_{SP} >> I_{GON} > I_{GOFF} >> I_{GONP} > I_{GOFFP} \quad (3.17)$$

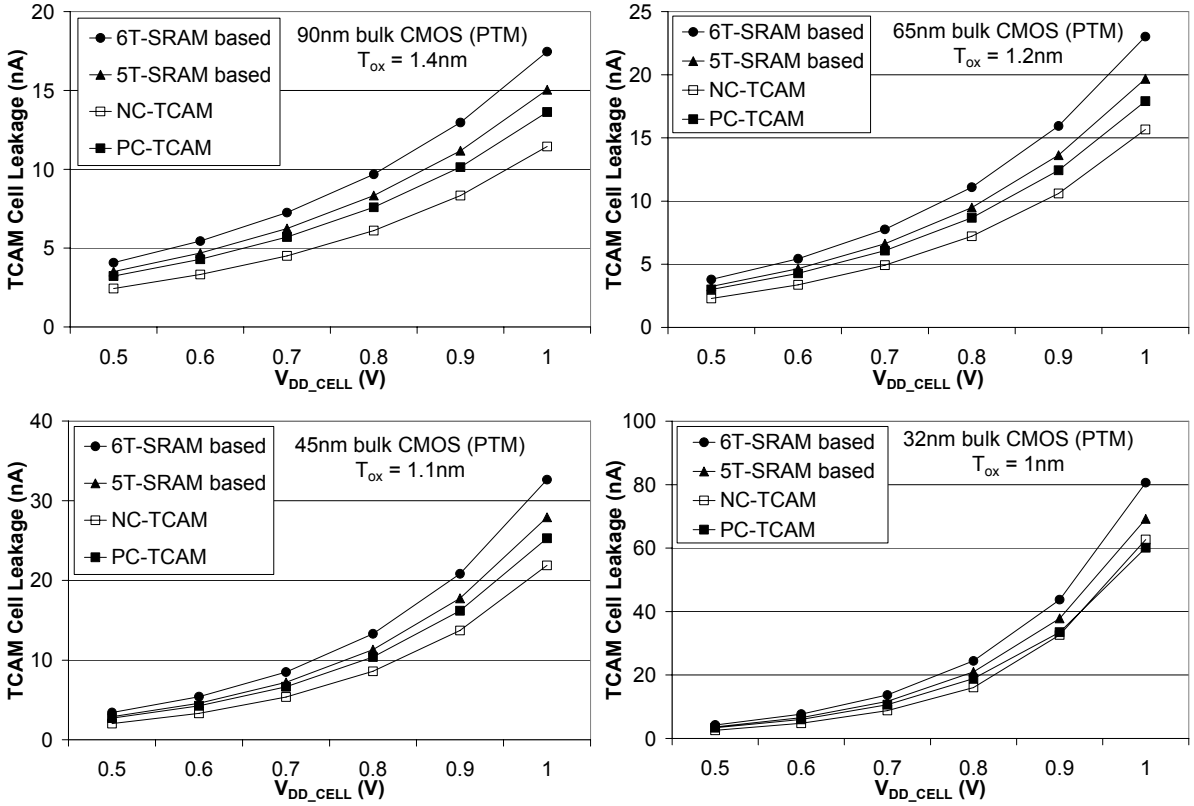


Figure 3.12: TCAM cell leakages for different technology nodes (PTM) at 300°K

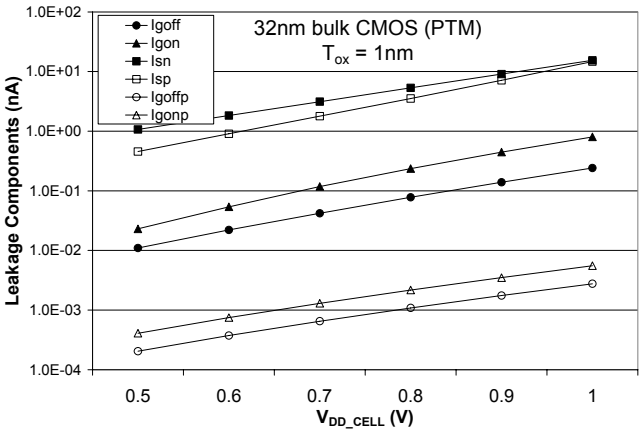


Figure 3.13: Leakage components of 32nm bulk CMOS transistors (PTM) at 300°K

Thus, an NMOS leakage component is relatively larger than the corresponding PMOS leakage component. Also, subthreshold leakages are much larger than the gate leakages. Finally, the PMOS gate leakage is much smaller than the NMOS gate leakage. Since gate leakages are relatively smaller and I_{SN} is larger than I_{SP}, NC-TCAM cell exhibits up to 40% less leakage than the 6T-SRAM-based cell as estimated by Table 3.3. In 32nm, the PC-TCAM cell exhibits the minimum leakage at V_{DD_CELL}=1V because I_{SP} increases to a value close to I_{SN}, and the effect of gate leakage becomes noticeable.

3.1.2.6. Chip Design and Measurement Results

We designed a column of 256 PC-TCAM cells on a test chip (Figure 3.14) in 0.18µm CMOS technology to demonstrate the READ/WRITE functionality of the proposed cell. All 256 cells share nodes SRC1 and SRC2 illustrated in Figure 3.7 and Figure 3.11. The measurement results in Figure 3.15 show that the proposed cell successfully writes '0' and '1' in 0.48ns and 3.6ns, respectively. We also added a 256-bit scan chain to demonstrate that the WRITE1 operation in the selected cell does not disturb the remaining cells sharing the same SRC1 and SRC2. Initially, logic '0' was written to the whole column. Then, logic '1' was written to the selected cell. Finally, the whole column was latched and scanned-out. By observing the scan chain output, it was verified that no other cell has been flipped from '0' to '1' during the WRITE1 operation. The READ operation was performed in 1.1ns using reference BLs and the single ended BLSA illustrated in Figure 3.6. Therefore, the READ and WRITE operations of the proposed cell are fast enough to support present and future TCAM applications. The chip measurement results are summarized in Table 3.4.

Table 3.4: Chip measurement result summary for a column of 256 PC-TCAM cells

S. No.	*Feature*	*Value/Result*
1.	Process Technology	0.18µm 1.8V bulk CMOS
3.	T_{WRITE0} (ns)	0.48
4.	T_{WRITE1} (ns)	3.6
5.	T_{READ} (ns)	1.1

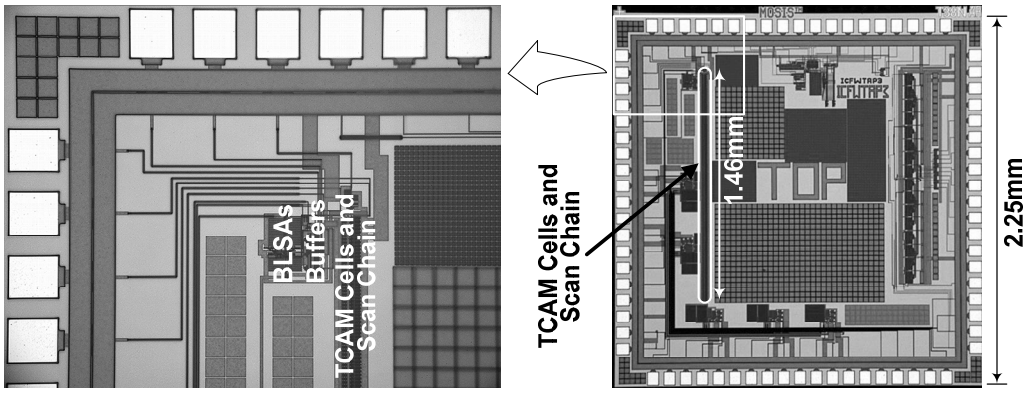


Figure 3.14: Chip micrograph of a column of 256 PC-TCAM cells

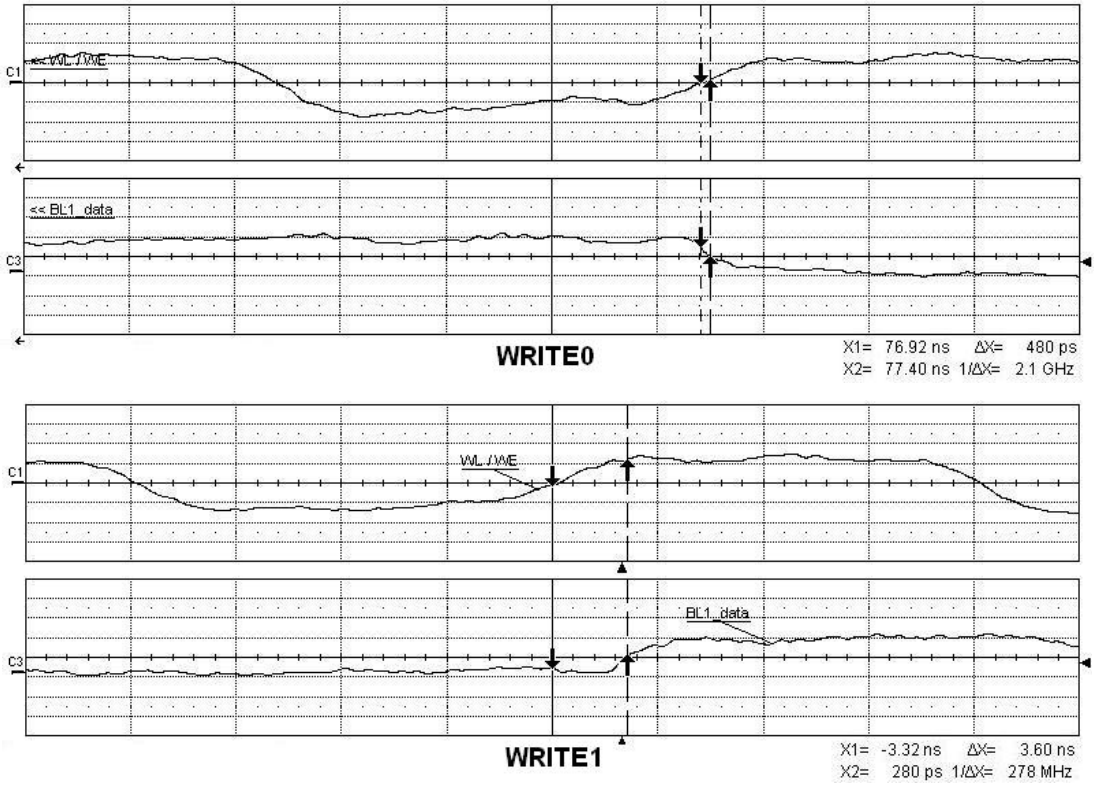


Figure 3.15: Chip measurement results of WRITE0 and WRITE1

3.2. Low-Capacitance Comparison Logic

A significant portion of the TCAM power is consumed in switching highly capacitive MLs. It can be observed from Figure 1.7 that there are two main sources of the ML capacitance: (i) interconnect capacitance of the metal used for ML routing, and (ii) drain capacitances of the comparison logic transistors. The ML interconnect capacitance mainly consists of (i) ML-to-substrate capacitance, and (ii) the coupling capacitance between MLs and other parallel lines such as WLs, GND and V_{DD} buses (running horizontally in Figure 1.7). The ML-to-substrate capacitance can be reduced by choosing a high-level metal (such as M4) with minimum width (as specified by the design rules) for routing MLs. Similarly, the ML coupling capacitance can be minimized by (i) routing MLs and other parallel lines in different metals, and (ii) placing MLs equally apart from the other parallel lines.

Drain capacitances of the comparison logic transistors also contribute to the ML capacitance. A significant reduction in the ML capacitance can be achieved by employing minimum size transistors in the comparison logic circuits. If secondary effects are ignored, the drain capacitance, I_{ON} and I_{OFF} are directly proportional to the channel width. As a consequence, the speed and robustness of the ML sensing is not affected by the channel width. Therefore, the minimum size transistors reduce the search energy without degrading the I_{ON}/I_{OFF} ratio. In sub-100nm CMOS technologies, the channel width can be slightly larger than the minimum size specified by the design rules to avoid excessive process variations and secondary effects (such as normal and reverse narrow channel effects).

3.2.1. Conventional Comparison Logic

Figure 3.16 shows a conventional 16T TCAM cell and its contribution to the ML capacitance under different masking conditions. If a cell is not masked, it adds a capacitance of $4C_D$ to the corresponding ML, where C_D is the drain capacitance of a comparison logic transistor that includes the bottom-plate and side-wall junction capacitances (Figure 3.16(b)). Similarly, globally and locally masked cells add capacitances of $2C_D$ and $4C_D$, respectively (Figure 3.16(c) and Figure 3.16(d)). Therefore, each conventional TCAM cell contributes a capacitance of $2C_D$ or $4C_D$ depending on the masking conditions.

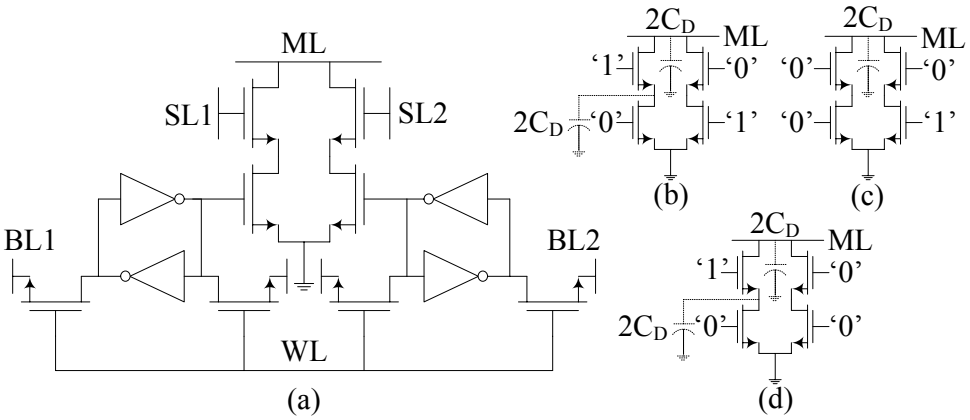


Figure 3.16: (a) A 16T TCAM cell and its contribution to the ML capacitance under the following conditions: (b) no masking, (c) global masking, (d) local masking

3.2.2. Proposed Comparison Logic

We proposed a cell-level comparison logic (shown in Figure 3.17) that offers a smaller ML capacitance [58]. The proposed comparison logic requires an additional line (SelGbl) to keep node 'G' at ground under the global masking condition (SL1=SL2='0'). SelGbl is generated by NORing SL1 and SL2, and it is shared by all the cells in the same column. Since SL1=SL2='1' is an invalid state, the possibility of shorting the inverter outputs is eliminated. Similar comparison logic (without transistor M2) has been used in binary CAMs [59]. However, it has not been reported in TCAMs possibly due to floating node 'G' in a globally masked cell. The proposed comparison logic employs transistor M2 for driving node 'G' to ground in a globally masked cell.

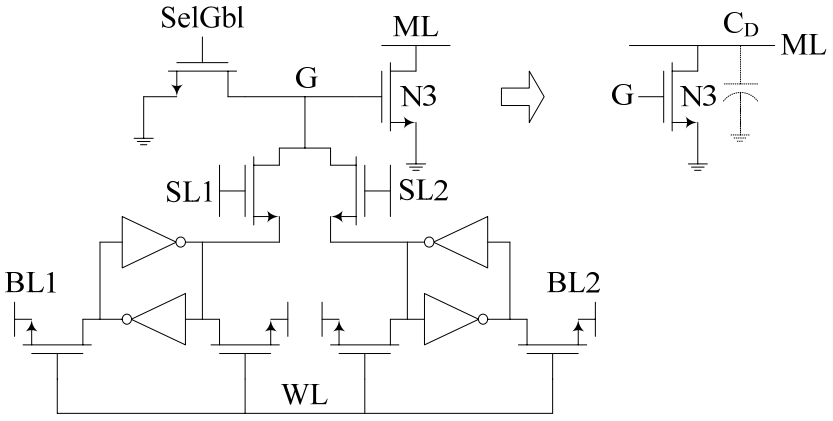


Figure 3.17: A TCAM cell based on the proposed comparison logic, and its contribution to the ML capacitance under all the masking conditions

If none of the bits are globally masked and the interconnect capacitance is ignored, the proposed comparison logic reduces the ML capacitance by 75% ($4C_D$ to C_D). Similarly, if all the bits are globally masked, the ML capacitance is reduced by 50% ($2C_D$ to C_D). Therefore, the capacitance reduction varies between 75% and 50% depending on the number of globally masked bits. However, this reduction in ML capacitance comes at the expense of additional lines (SelGbl) and associated energy consumption. Fortunately, the rate of updating the global mask registers is negligibly less than the table lookup frequency in most TCAM applications [60]-[63]. Thus, the power consumed in switching SelGbls is negligibly less than the power consumed in switching MLs.

As described in Chapter 2 (section 2.4), the robustness of ML sensing can be improved by increasing the I_{ON}/I_{OFF} ratio of the ML pull-down path. The proposed comparison logic has only one NMOS transistor in the ML pull-down path (Figure 3.17). Hence, its worst-case ML 'ON' current (I_{ML1}) will be greater than that of the conventional comparison logic, which has two series-connected transistors instead (Figure 3.16). Measurement results (shown in the following subsections) also support the above deduction even though the voltage swing of node 'G' is limited to ($V_{DD} - V_{tn}$). Furthermore, the proposed comparison logic has only one ML leakage path per cell. Thus, its ML 'OFF' current (I_{ML0}) will be less than that of the conventional comparison logic, which has two ML leakage paths per cell. Although one of the ML leakage paths of a masked cell can have a lower leakage due to the body-effect, the I_{ML0} of the conventional scheme is still larger than that of the proposed scheme due to absence of the body-effect in the remaining leakage paths (Figures 3.16(b), (c), (d)).

A larger I_{ML1}/I_{ML0} ratio has two main advantages. First, it makes the ML sensing less sensitive to process variations and operating conditions. For example, if an ML with one-bit mismatch (ML_1) is receiving a larger current than the dummy ML due to the mismatch in their current sources, ML_1 may be detected as a "match" before the output of the dummy MLSA turns 'OFF' the current sources. This problem may also be caused by a threshold voltage mismatch between the dummy MLSA and the MLSA connected to an ML_1. A larger margin between I_{ML1} and I_{ML0} can cope with larger process variations. Thus, the same dummy ML can be used for a larger block of TCAM words increasing the layout density of the

TCAM chip. Secondly, a larger I_{ML1}/I_{ML0} ratio allows the implementation of wide TCAMs because I_{ML0} is proportional to the word size (l), and a larger value of l diminishes the difference between I_{ML1} and I_{ML0}.

We analyzed the proposed and conventional comparison logic circuits by implementing them in two 145-bit wide TCAM words as shown in Figure 3.18(a). A charge-redistribution MLSA (also described in Chapter 2) is used for ML sensing whose timing diagram is shown in Figure 3.18(b) [34]. All the control signals are common to both MLSAs. Initially, the MLs are discharged to ground using PRE. The search operation is initiated by the rising edge of EN, and the falling edges of FastPre and PRE. The ML voltage swing is restricted by the NMOS transistors (N1 and N2) whose gates are connected to a reference voltage (V_{REF}). The FastPre pulse precharges the MLs to a voltage near ($V_{REF} - V_{tn}$). The evaluation begins with the rising edge of the FastPre signal. Under the match condition, the ML does not have a pull down path, and its node SP remains at V_{DD}. Under the mismatch condition, the node SP is pulled down to GND through N2 and ML discharge path. A small current source (I_{REF}) at the node SP compensates for ML leakages. In our design, I_{REF} has been set to one-fifth of I_{ON}.

The width of the FastPre pulse (T_{FP}) is the most critical parameter in the charge-redistribution MLSA [34]. If T_{FP} is too small, MLs will be precharged to a voltage much lower than ($V_{REF} - V_{tn}$). Under the match condition, this incomplete precharge can cause a false glitch at the MLSA output by charge sharing ML_New and SP (Figure 3.18). This false glitch increases energy consumption and affects the operation of the next stage. On the other hand, a wider T_{FP} pulse increases the energy consumption due to the direct current paths (from V_{DD} to GND) in the words that fail to match the search key. Larger values of T_{FP} also increase the search time. Therefore, the T_{FP} window is chosen just wide enough to avoid a false glitch under the match condition. The false glitch problem is less severe for the proposed comparison logic due to two main reasons. First, a lower capacitance implies a faster precharging of MLs to ($V_{REF} - V_{tn}$). When MLs are charged closer to ($V_{REF} - V_{tn}$), the duration of the false glitch is reduced. Faster precharging of MLs also reduces search time and energy. Secondly, the voltage drop at node SP due to charge sharing between nodes SP and ML is less severe if ML capacitance is smaller.

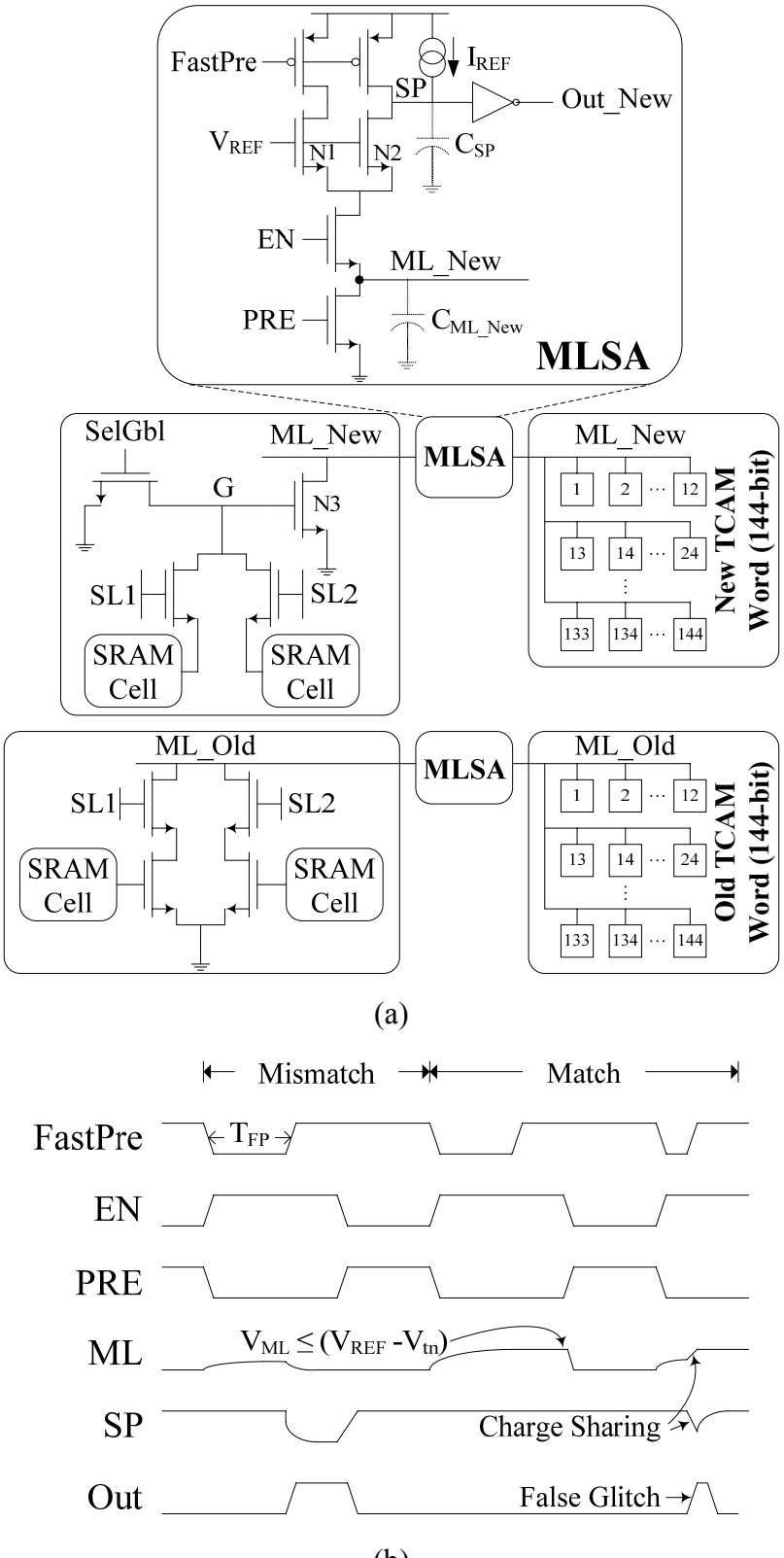


Figure 3.18. (a) TCAM words employing the proposed and conventional comparison logic circuits with the charge-redistribution MLSA, and (b) its timing diagram

3.2.3. Chip Design and Measurement Results

We implemented two 145-bit wide TCAM words (shown in Figure 3.18(a)) on a test chip fabricated in CMOS 0.18µm technology to demonstrate the benefits of the proposed comparison logic over its conventional counterpart. A micrograph of the test chip is shown in Figure 3.19. The 144-bit portion of each word is arranged in an array of 12x12 cells, and all the cells are hard-wired for the match condition due to area constraints. One conventional cell and one proposed cell are connected to the respective words in parallel. The match and mismatch conditions for the two words are obtained by changing the status of these two cells. Both words have separate MLSAs and power-supply pins to measure the energy consumption. A reference circuit is also included to generate bias voltages for the MLSAs. All the control signals are common to both MLSAs.

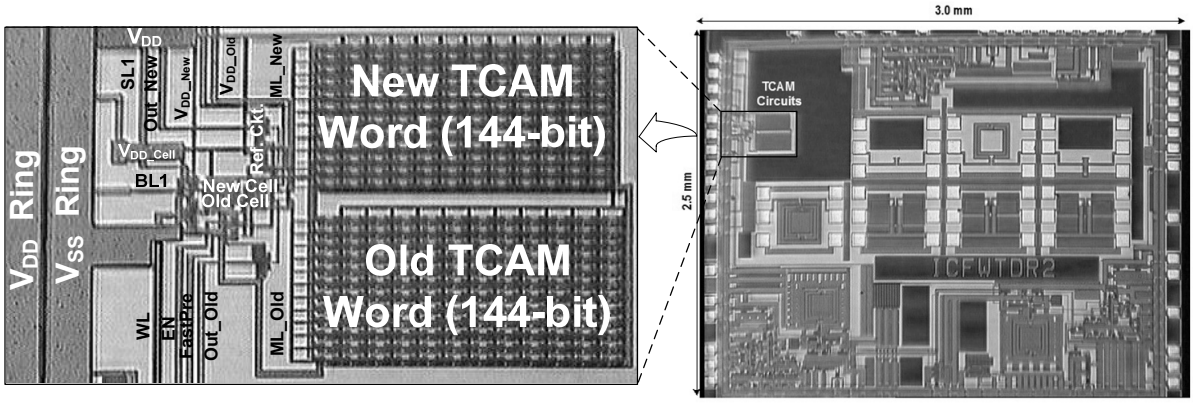


Figure 3.19. Micrograph of the test chip with the proposed and conventional comparison logic circuits

Figure 3.20 shows measurement results of the test chip. Here, T_{FP} and energy of the conventional and proposed schemes are shown for different values of TCAM cell supply voltage (V_{DD_CELL}) while the supply voltage of SL drivers and MLSAs remains at 1.8V. Energy is measured for the mismatch condition since most words fail to match the search key in typical TCAM applications. A reduction in V_{DD_CELL} reduces both I_{ON} and I_{REF}. However, the SEARCH operation is performed successfully as long as I_{REF} is large enough to compensate for ML leakages. A small V_{DD_CELL} also reduces the static power which is becoming a serious issue in sub-100nm technologies. Measurement results confirm that the ML with the proposed comparison logic gives consistent energy (25%) and time (42%)

savings for a large range of V_{DD_CELL}. Since C_{SP} is much smaller than C_{ML_New}, the SP voltage drops almost immediately after the rising edge of the FastPre pulse (Figure 3.18). Hence, the reduction in ML sensing time is almost equal to the reduction in T_{FP}. For smaller values of V_{DD_CELL}, T_{FP} increases due to a reduction in I_{REF}. Energy is less affected by the variations in V_{DD_CELL} because an increase in T_{FP} is compensated by a reduction in I_{ON}. For very low values of V_{DD_CELL}, the reduction in I_{ON} becomes more prominent and the energy consumption decreases.

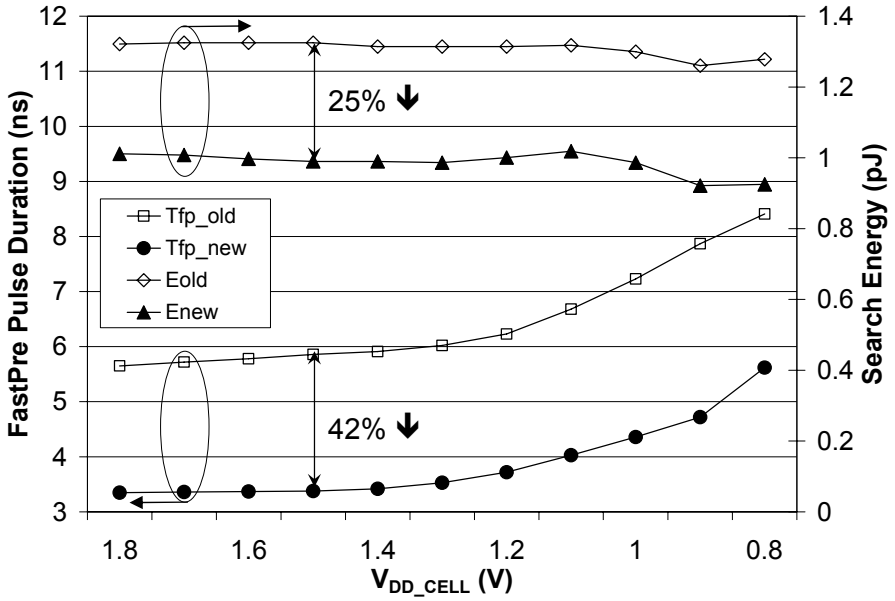


Figure 3.20. FastPre pulse duration (T_{FP}) and energy measurement results of conventional and low-capacitance ML schemes for different values of V_{DD_CELL}

3.2.4. Analysis and Discussion

In the chip implementation of the conventional comparison logic (Figure 3.19), the drains of upper two transistors were merged (Figure 3.18(a)). Thus, each unmasked cell added a capacitance of $3C_D$ to the corresponding ML (instead of $4C_D$ as estimated in subsection 3.2.1). Using the design parameters from the TSMC technology documents, we calculated The ML capacitances of the two 145-bit words, which are comprised of NMOS transistors with W/L = 0.6μm/0.18μm and C_D = 0.606fF:

$C_{ML\_Old} = 145 \times 3C_D + 141fF = 404.61fF$

$C_{ML\_New} = 145 \times C_D + 148fF = 235.87fF$

where second terms are the extracted interconnect capacitance (including the bottom-plate, fringe, and coupling capacitance) for the two MLs. Thus, the new comparison logic reduces the ML capacitance by 42%.

In order to verify the above calculations, we performed an indirect measurement of the ML capacitance. As explained in subsection 3.2.2, T_{FP} is typically chosen (3-6ns) just large enough to avoid the false glitch under the match condition. In order to charge both C_{ML_Old} and C_{ML_New} approximately to the same voltage ($V_{REF} - V_{th}$), we set both MLs in match condition and chose a much larger value of T_{FP} (T_{FP} = 15ns, period = 20ns). Since a matching ML has no conducting path to GND, the whole current drawn from the power-supply is consumed in charging the ML capacitance. The average currents drawn by C_{ML_Old} and C_{ML_New} from the power-supply have been measured to I_{Old} = 21.7μA and I_{New} = 13.9μA. Therefore, the ML capacitance is reduced by 36%, which is less than the expected value (42%). This implies that the value of T_{FP} (15ns) is not large enough, and C_{ML_New} is charged to a slightly higher voltage than C_{ML_Old}. Both ML capacitances can be charged to a voltage much closer to ($V_{REF} - V_{th}$) by further increasing T_{FP}. However, large values of T_{FP} reduce the average power-supply currents, and this method loses its accuracy due to two main reasons. First, the average power-supply currents may become comparable to the ML leakages. Second, it becomes difficult to measure a small current accurately. We also observed that a variation in V_{DD_CELL} does not change the measured power-supply currents, which reinforces the fact that there is no V_{DD_CELL}-dependent conducting path from ML to GND under the match condition.

The proposed comparison logic has a greater I_{ML1} than that of the conventional comparison logic as explained in subsection 3.2.2. We indirectly measured the approximate I_{ML1} of the proposed and conventional comparison logic circuits. We set both MLs in mismatch condition, and chose T_{FP} = 15ns and period = 20ns. In this case, the average current drawn from the power-supply is proportional to I_{ML1} once the ML voltage reaches steady state. Figure 3.21 shows the measured I_{ML1} of the two comparison logic circuits. For V_{DD_CELL} = 1.8V, I_{ML1} of the proposed comparison logic is 14% higher than that of the conventional comparison logic. For smaller values of V_{DD_CELL}, the difference between the two 'ON' currents reduces because the time taken in reaching steady state becomes comparable to T_{FP}, and the measurements become less accurate.

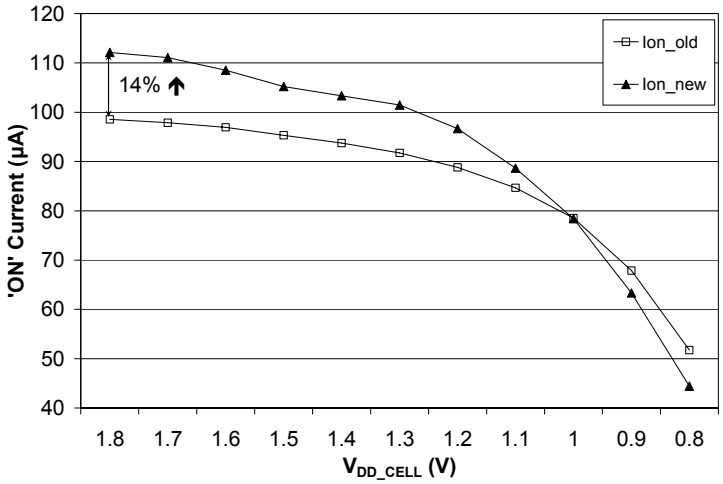


Figure 3.21. Measurement results for I_{ON} of the proposed and conventional comparison logic circuits

The proposed comparison logic can be further optimized using efficient layout techniques. For example, ML transistors (N3 in Figures 3.17 and 3.18) of two adjacent TCAM cells can share the same drain contact. Such a layout results in a smaller ML capacitance. In the present chip, we laid-out the 144-bit portion of each ML in an array of 12x12 cells (Figure 3.18). The extracted interconnect capacitance for ML is found to be 145fF. When we laid out each ML as one row, the interconnect capacitance is reduced to 52fF. Figure 3.22 also shows an improved layout of the conventional comparison logic where the contacts are removed from the internal nodes in order to reduce their capacitance. Using the above layout techniques and minimum size transistors $\left(\frac{W}{L}=\frac{0.42\mu m}{0.18\mu m}\right)$ in 0.18µm CMOS technology, capacitances of 144-bit ML_Old and ML_New can be calculated as,

C_{ML_Old} = 144 x (0.825fF) + 52fF = 170.8fF

C_{ML_New} = 144 x (0.309fF) + 52fF = 96.5fF

Thus, the proposed comparison logic with the modified layout can achieve a 44% reduction in ML capacitance. Hence, the actual capacitance reduction (44%) is smaller than the theoretical value (75% as predicted in subsection 3.2.2) due to layout techniques and interconnect capacitance. It should be noted here that the minimum size transistor has a width of 0.42µm instead of the minimum poly width specified by the design rules (W_{Poly} = 0.22µm). This transistor width ensures that the source and drain contacts (whose size is also specified by the design rules) fit in this width without resorting to unusual (bone-shaped)

transistor layouts. Such unusual layouts are not attractive for TCAM cells because they consume more area than a regular shaped transistor with a width of 0.42μm.

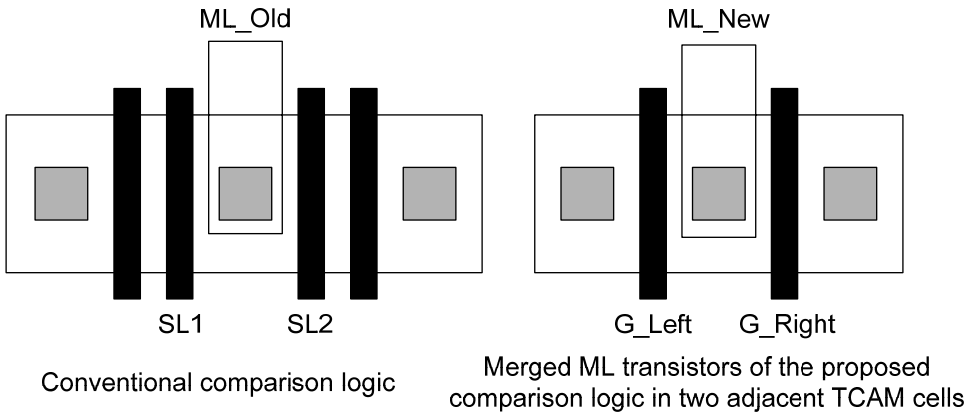


Figure 3.22. Suggested layouts of conventional and proposed comparison logic circuits

3.3. Conclusions

This chapter proposed three cell-level design techniques for TCAMs. The first technique reduces the cell-leakage by using a smaller V_{DD_CELL} in the storage portion and a higher V_{DD} in MLSAs and SL drivers. It provides a trade-off between the leakage and robustness of a TCAM chip. The second technique reduces the area and leakage of the conventional TCAM cell by removing two access transistors and eliminating a subthreshold leakage path. Simulation results of the proposed cells show up to 40% leakage reduction over the conventional TCAM cell. Chip measurement results show that the proposed cell performs READ/WRITE fast enough to support present and future TCAM applications. The third technique reduces the ML-capacitance by modifying the comparison logic of the conventional TCAM cell. The chip measurement results show 42% and 25% reduction in search time and energy, respectively. We analyzed the measurement results and proposed possible improvements. The ML capacitance can be further optimized by efficient layout techniques described in subsection 3.2.4.

Some or all of the above techniques can be adopted by TCAM designers depending on the word-size, fabrication technology, storage capacity, and target applications of the chip under consideration. They can also be combined with higher-level power reduction techniques described in the following chapters.

Chapter 4

Low-Power Match Line Sensing

In previous chapters, we highlighted the importance of power reduction in ML sensing. Since all MLs are initially precharged to V_{DD} and then discharged to GND in every SEARCH operation, their power consumption has been a serious concern. Thus, most low-power TCAM techniques have been developed for reducing power consumption in ML sensing. Typically, two low-power approaches have been explored for ML sensing. The first approach attempts to reduce the switching activity using ML segmentation. The second approach redesigns the MLSA such that the ML voltage swing is reduced. Since both approaches can be applied independently, their combination usually maximizes the power savings. In this chapter, we propose low-power techniques that follow both approaches.

4.1. Low-Power Match Line Segmentation

In Chapter 2, we explained the concept of ML segmentation, which can achieve power reduction by dividing each ML into two or more segments and sensing them sequentially. For example, the selective-precharge scheme divides each ML into two segments (ML1 and

ML2) each connected to a separate MLSA (MLSA1 and MLSA2). If ML1 does not match with the corresponding portion of the search-key, MLSA2 is not enabled, and ML2 is not sensed. For a given word-size (l), ML1 and ML2 are chosen such that the total power is minimized. If ML1 and ML2 contain k and (l-k) cells, respectively, the ML sensing energy can be given by equation (4.1).

$$E_{ML} = \left[P_{ML1} k + (1 - P_{ML1}) l \right] E_{ML\_CELL} \tag{4.1}$$

where P_{ML1} is the probability of finding a mismatch in ML1 and E_{ML_CELL} is the ML energy consumption per cell. Similarly, the conventional ML sensing (without segmentation) will consume an energy of (l x E_{ML_CELL}). In order to minimize E_{ML} in equation (4.1), it is usually desirable to maximize P_{ML1} (close to unity) and minimize k. This combination reduces the first term (P_{ML1} k) while making the second term $(1 - P_{ML1})(l - k)$ negligible. However, these two constraints (large P_{ML1} and small k) are difficult to achieve simultaneously. For example, a small value of k makes it less likely to find a mismatch in ML1 segments and reduces P_{ML1}. Typically, k is chosen to a value that minimizes E_{ML} for a given data statistics but the same value of k may not minimize power in a different application. Therefore, an alternative ML segmentation scheme has been explored that can be used in broad range of applications.

4.1.1. Dual ML TCAM

We proposed a Dual ML TCAM that eliminates the above data-dependency and achieves power savings irrespective of the incoming data statistics [64]. The dual-ML TCAM employs two wires (ML1 and ML2) connecting to the left and right sides of the comparison logic, respectively (Figure 4.1). Both ML1 and ML2 have separate sense amplifiers (MLSA1 and MLSA2). First MLSA1 is enabled. If MLSA1 detects a mismatch, it does not enable MLSA2 and saves power. Hence theoretically, the power consumption is reduced by half if mismatch is found in most ML1 segments.

In the above discussion, we neglected the effect of interconnect capacitance. Since ML1 and ML2 both run horizontally across the whole array, the interconnect capacitance of the dual ML TCAM is approximately two times larger than that of the conventional TCAM. This additional interconnect capacitance makes the dual ML TCAM slightly slower than the conventional TCAM, and it also limits the power reduction to a value less than 50%.

Assuming metal 4 routing of ML in CMOS 0.18µm technology, the interconnect capacitance was extracted to 0.19fF/cell. For l = 144, the total interconnect capacitance of each ML was estimated to be C_{INT} = 144 x 0.19fF = 27.36fF. In order to perform more realistic simulations, we connected this capacitance (C_{INT}) to every ML segment.

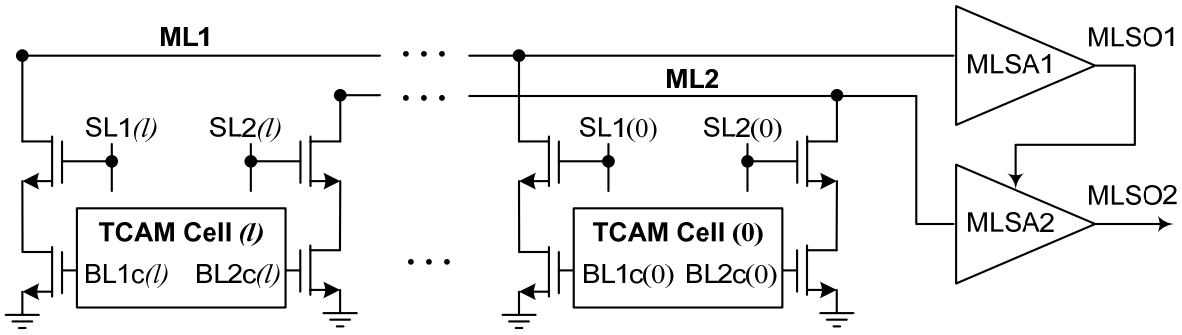


Figure 4.1: Proposed dual ML TCAM

4.1.1.1. Simulation Results

We simulated the conventional and dual-ML TCAMs for 144-bit words in 0.18µm CMOS technology using the current-race MLSA. Tables 4.1 and 4.2 show the simulation results with and without the interconnect capacitance (C_{INT} = 27.36fF). When C_{INT} is not included in the simulations, the search time of the dual ML TCAM remains the same as that of the conventional TCAM because both ML1 and ML2 are charged two times faster than the conventional ML. Energy is specified only for the mismatch condition because every SEARCH operation results in a mismatch for most TCAM words. Thus, the average energy is dominated by the mismatch condition. E_{ML1} is the energy consumption when the mismatch is found in the ML1 segment. Since ML2 is not enabled, the energy is reduced by 47%. However, if the mismatch is not detected in ML1, the energy consumption (E_{ML2}) increases by 23%. This increase can be explained using the circuit schematic of the current-race MLSA (Figure 2.4(b)). For the match condition, the MLSA output flips from '0' to '1' because the NMOS transistor connected to the ML overrides the PMOS keeper. Thus, for a small duration, both the NMOS transistor and the keeper are 'ON' resulting in a large transient current. The dual ML TCAM has a larger E_{ML2} because MLSA1 does not detect a mismatch, and flips MLSO1 from '0' to '1' (Figure 4.1). As a consequence, the total energy increases due to the transient current drawn by MLSA1.

Table 4.1: Simulation results without the interconnect capacitance

	Conventional	***Dual ML***	***Difference***
Search time (T_S)	7.88ns	7.88ns	-
Mismatch Energy: ML1 (E_{ML1})	747fJ	394fJ	47% ↓
Mismatch Energy: ML2 (E_{ML2})	747fJ	918fJ	23% ↑

Table 4.2: Simulation results with the interconnect capacitance (C_{INT} = 27.36fF)

	Conventional	***Dual ML***	***Difference***
Search time (T_S)	8.14ns	8.46ns	3.9% ↑
Mismatch Energy: ML1 (E_{ML1})	769fJ	426fJ	45% ↓
Mismatch Energy: ML2 (E_{ML2})	769fJ	973fJ	26% ↑

When C_{INT} is included in the simulations (Table 4.2), the dual ML TCAM shows a slightly larger search time (3.9%) due to the increased ML capacitance, which also increases E_{ML1} and E_{ML2}. It can be noticed from Tables 4.1 and 4.2 that the dual ML scheme is power efficient only if the mismatch is detected by MLSA1 itself. If the probability of this event (mismatch detected by MLSA1) is denoted by P_{ML1}, the average ML sensing energy can be given by equation (4.2).

$$E_{Dual\_ML} = P_{ML1}E_{ML1} + P_{ML2}E_{ML2} \tag{4.2}$$

where P_{ML2} is the probability of the event that a mismatch is detected by MLSA2. In typical TCAM applications, most MLs fail to match the search key. Hence, the probability of finding a mismatch (either by MLSA1 or MLSA2) is close to unity as shown by equation (4.3).

$$P_{ML1} + P_{ML2} = 1 \tag{4.3}$$

Substituting P_{ML2} from equation (4.3) to equation (4.2):

$$E_{Dual\_ML} = P_{ML1}E_{ML1} + (1 - P_{ML1})E_{ML2} \tag{4.4}$$

If P_{ML1} is also close to unity, it can be shown from equation (4.4) that the dual ML scheme will result in significant power savings. In the next subsection, we will derive an expression for P_{ML1} and demonstrate that P_{ML1} is close to unity in most TCAM applications.

4.1.1.2. Analysis and Discussion

In order to determine the average energy of the dual ML TCAM, an expression for P_{ML1} is required. By definition, the probability of an event is equal to the number of favourable cases divided by the total number of cases. Thus, all possible types of mismatches should be determined before P_{ML1} is calculated. Normally, each TCAM cell can have one of the two types of mismatches shown in Table 4.3.

Table 4.3: Type of mismatches in a TCAM cell

Mismatch Type	***SL1***	***SL2***	***BL1***	***BL2***
Type I	0	1	1	0
Type II	1	0	0	1

Since both types of mismatches are equally probable, their probability of occurrence is 0.5. It can be shown from Figure 4.1 that the Type I mismatch does not create an ML1-to-GND discharge path. Hence, it cannot be detected by MLSA1. On the other hand, the Type II mismatch can be detected by MLSA1 (Figure 4.1). Typically, most TCAM words have multiple-bit mismatch, and all the cells in a word share the same ML1. If a word has multiple-bit mismatch, only one Type II mismatch is sufficient for the MLSA1 to detect the word-level mismatch. Assuming the number of bit-level mismatches in a word is *m*, the probability of the event that all the mismatches are of Type I is $(0.5)^m$. Hence, the probability that at least one of the mismatches belongs to Type II can be given by equation (4.5):

$$P_{ML1} = 1 - (0.5)^m \tag{4.5}$$

The above probability has been equated to P_{ML1} because MLSA1 will detect the word-level mismatch if at least one bit has a Type II mismatch. Substituting P_{ML1} from (4.5) into (4.4):

$$E_{Dual\_ML} = \left[1 - (0.5)^m\right] E_{ML1} + (0.5)^m E_{ML2} \tag{4.6}$$

The average energy of the dual ML TCAM can be estimated by equation (4.6). The average number of mismatches (*m*) can be determined from the data statistics, and E_{ML1} and E_{ML2} can be determined from simulations as shown in Tables 4.1 and 4.2.

Table 4.4 shows P_{ML1} and E_{Dual_ML} for increasing values of *m*. Here, E_{ML1} and E_{ML2} are substituted from the simulation results of Table 4.2. The variation of P_{ML1} with *m* is

shown in Figure 4.2. As expected, P_{ML1} dramatically increases with m and reaches a value close to unity. Figure 4.3 shows the average ML sensing energy of the conventional and dual-ML TCAMs for different values of m. For $m \geq 5$, the dual-ML TCAM results in a 43% reduction in ML sensing energy as shown in Figure 4.3. However, this energy reduction

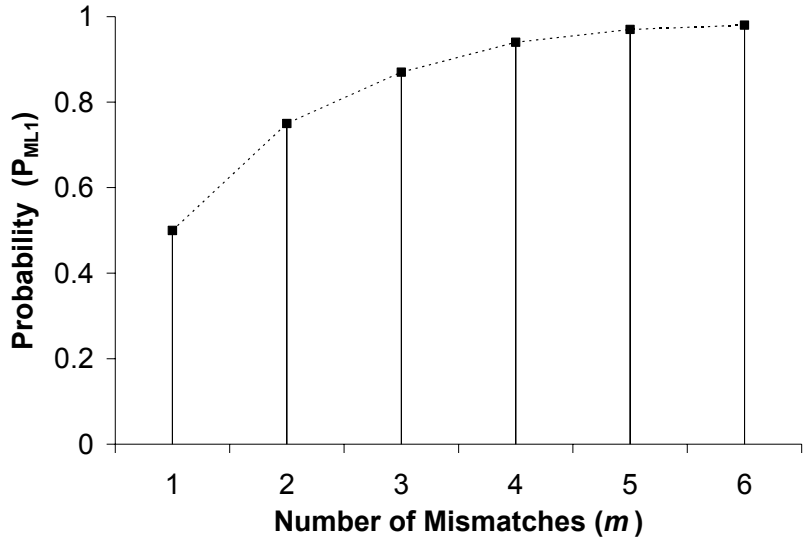


Figure 4.2: Variation of P_{ML1} with the number of mismatches

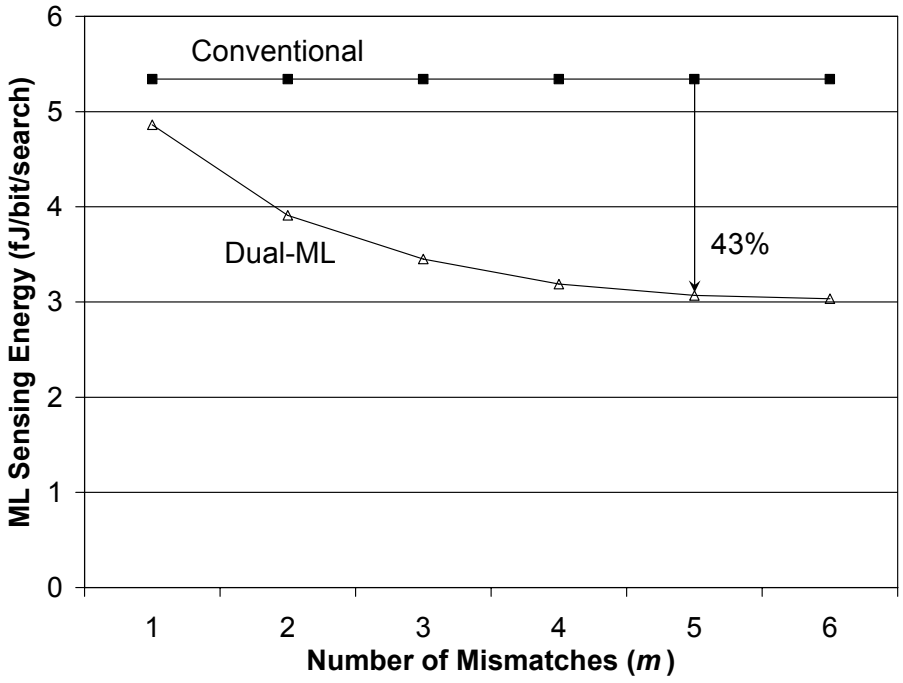


Figure 4.3: Average ML sensing energy of conventional and dual-ML TCAMs

comes at the expense of a small increase in the search time (4%). 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 described in Chapter 2. In the selective-precharge TCAM, MLSO1 lines run over the Main-Search TCAM array to enable MLSA2 circuits (Figure 2.6(b)). The parasitic capacitance due to these lines increases the search delay and power consumption. The dual-ML TCAM eliminates this additional parasitic capacitance by placing both MLSA1 and MLSA2 on the same side of TCAM array (Figure 4.1). Therefore, if the incoming data statistics is unpredictable, the dual-ML TCAM can achieve better power savings than the selective-precharge scheme.

Table 4.4: P_{ML1} and E_{Dual_ML} for increasing values of the number of mismatches (*m*)

No. of Mismatches (m)	P_{ML1}	E_{Dual_ML} ***(fJ/word/search)***	***Reduction in energy (%)***
1	0.50	700	9
2	0.75	563	27
3	0.87	497	35
4	0.94	459	40
5	0.97	442	43
6	0.98	437	43

The dual ML scheme is particularly attractive for TCAMs with large word sizes, which increases the probability of having five or more mismatches in each word. It can also be combined with the selective-precharge scheme by further dividing the Main-Search segment (shown in Figure 2.6(b)) into two sub-segments as shown in Figure 4.1. The dual ML scheme may not be attractive for the Pre-Search segment (Figure 2.6(b)) because it is usually much smaller than the Main-Search segment, and a smaller segment normally has a lower probability of having five or more mismatches.

4.1.1.3. Layout Design Considerations

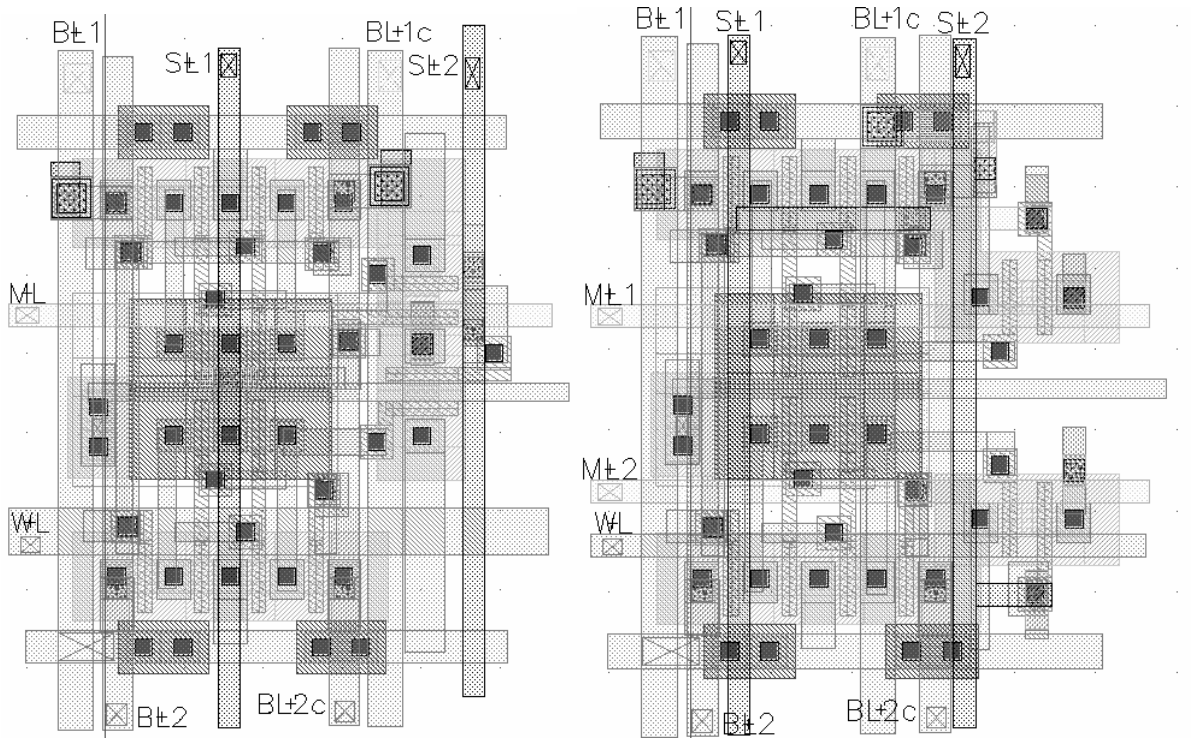
A reduction in ML1 capacitance through careful layout can reduce the average energy consumption of the dual ML scheme (E_{Dual_ML}). It can be shown from equation (4.4) that E_{ML1} and E_{ML2} affect the average energy very differently. For example, E_{ML1} is multiplied by

P_{ML1}, which is close to unity as shown in Figure 4.2. On the other hand, E_{ML2} is multiplied by $(1 - P_{ML1})$, which is close to zero. Therefore, ML1 and ML2 must be routed such that the ML1 capacitance is minimized even at the expense of a larger ML2 capacitance. A smaller ML1 capacitance results in a smaller E_{ML1}, which also reduces the average energy consumption. On the other hand, a larger ML2 capacitance remains disabled in most words, and its contribution to the average energy remains insignificant.

Sample layouts of conventional and dual ML TCAM cells in 0.18µm CMOS technology are shown in Figure 4.4. Note in Figure 4.4(b) that ML2 has been placed close to WL, while ML1 is placed at a larger distance from both ML2 and WL (lines parallel to ML1). Both cells have the same dimensions: 6.6µm x 5.7µm. The extracted values of the interconnect capacitance for 144-bit wide words are given below:

C_{ML} = 25fF, C_{ML1} = 31fF, C_{ML2} = 42fF

The dual ML segments have larger interconnect capacitance due to the closer placement of ML1 and ML2. However, a smaller value of C_{ML1} helps to reduce the average energy consumption.



(a) (b)

Figure 4.4: Layouts of (a) conventional and (b) dual-ML TCAM cells

4.1.1.4. Limitations and Trade-Offs

As mentioned earlier, the dual ML scheme can achieve significant power savings if the incoming data pattern (search-key) is unpredictable. In such applications, the selective-precharge scheme does not give the desired results because it is difficult to predict which bits (least significant bits or most significant bits: LSBs or MSBs) should be connected to the Pre-Search segment (Figure 2.6(b)). In addition, it is difficult to determine the minimum size of the Pre-Search segment that can detect mismatch for a variety of data patterns. However, the theoretical limit of power-reduction in the dual ML scheme is 50%. In realistic designs (with interconnect capacitances), the power reduction is less than 50% as shown by the simulation results in Figure 4.3. The layouts shown in Figure 4.4 are not optimized for minimum area. In denser layouts, routing an additional line (ML2) also increases the effective cell area and the interconnect capacitance. In that situation, the energy savings of the dual ML scheme may be even smaller than 43% (Figure 4.3). Therefore, the selective-precharge scheme is still preferable in power-critical applications where the data statistics are somewhat predictable. The dual ML scheme gives an alternative to perform ML segmentation in another design dimension. The selection of one of these techniques or a combination of these techniques depends on the target application(s), TCAM size and data statistics. The worst-case power of both schemes is essentially the same as that of the conventional TCAM. In order to make the ML-segmentation more attractive, we explored another word-level technique to reduce the search time and the worst-case power consumption as explained in the next subsection.

4.1.2. Charge-Shared MLs

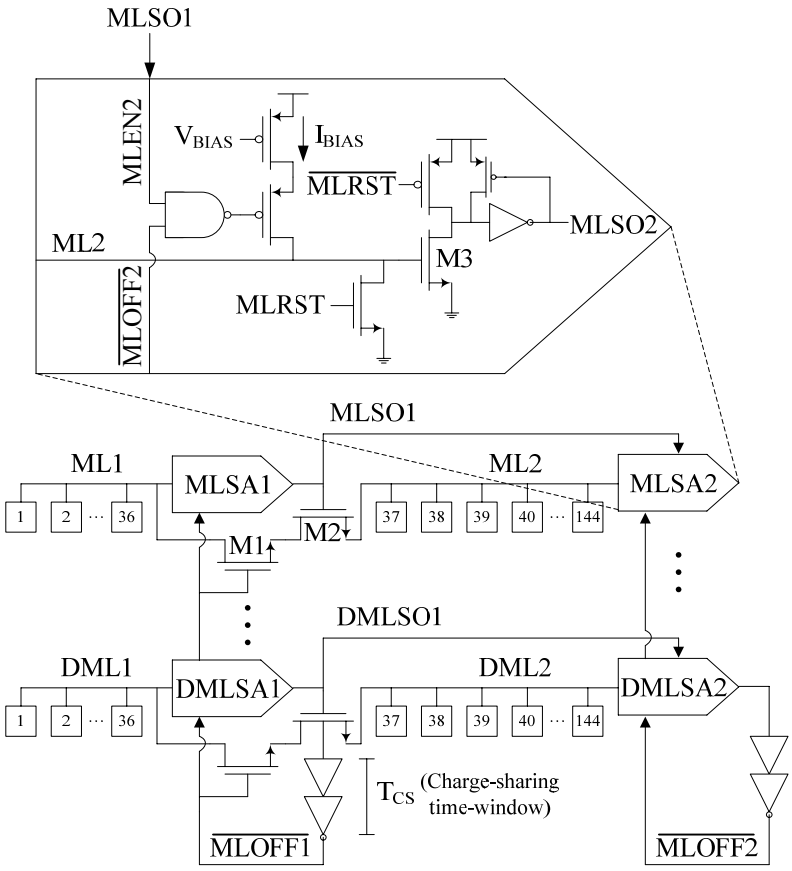
Most low-power TCAMs employ the selective-precharge scheme (or its modified version) for power reduction. As described earlier, the selective-precharge scheme divides large MLs into smaller segments and sense them sequentially [36][37][65][66]. For example, a 144-bit wide ML can be divided into two segments of 36 and 108 bits [65]. Each segment has a separate MLSA. First the smaller segment (ML1) is sensed. The larger segment (ML2) is sensed only if ML1 matches the corresponding portion of the search key. Therefore, this scheme saves power only in the best-case, which occurs when the first segments of most words do not match the same portion of the search key. The optimum size of ML1 is

determined from the data statistics. If a TCAM has segments optimized for one application, it will not give the best-case power in other applications. Thus, the actual power consumption varies between the best-case and the worst-case depending on the application.

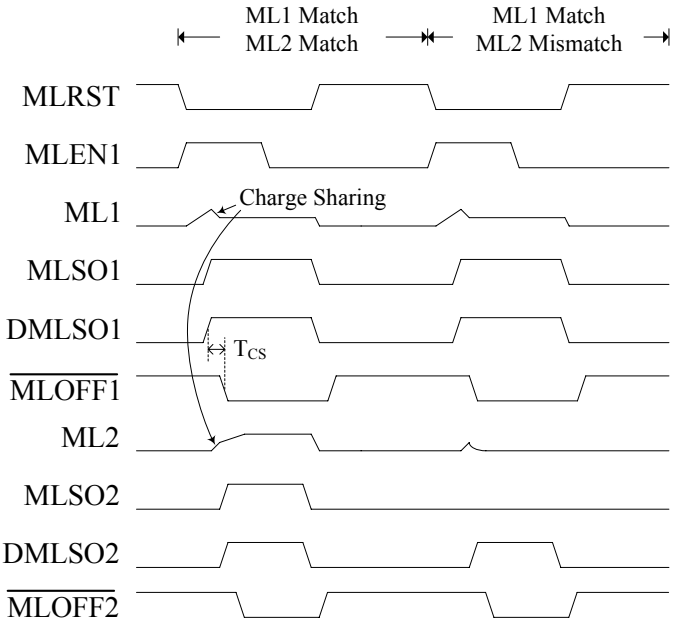
4.1.2.1. Charge sharing between ML Segments

We propose a charge-shared ML scheme that reduces the search time and the worst-case energy consumption [58]. Figure 4.5 illustrates the scheme and its timing diagram using a current-race MLSA [32]. The current-race MLSA requires a dummy word to generate the control signals. The dummy ML is also divided into two segments (DML1 and DML2). All the cells of the dummy segments are locally masked, so both dummy MLSAs generate a match in every SEARCH cycle. Since the ML capacitance varies with global masking, DML1 and DML2 should also track these variations. This is ensured by sharing the common SLs with the dummy word. A rising edge of MLEN1 begins the SEARCH operation by enabling all the MLSAs in the first segment (MLSA1, DMLSA1, etc.). A rising edge of DMLSO1 indicates the completion of the SEARCH operation in the first segment. A delayed version of DMLSO1 ($\overline{MLOFF1}$) is used to turn off the MLSAs in the first segment. The delay (T_{CS}) ensures that all the matched words are detected before the MLSAs are turned off. If the first segment of an ML matches with the corresponding portion of the search key, its MLSO1 turns on the corresponding MLSA2.

At the end of every SEARCH cycle, the conventional schemes discharge the residual ML1 charge to GND. The proposed scheme recycles the ML1 charge to reduce the search time and the worst-case energy consumption. If the first segment of a word results in a "match", its ML1 is charge-shared with its ML2 using transistors M1 and M2 (Figure 4.5). The charge sharing between DML1 and DML2 expedites the arrival of DMLSO2, which turns off MLSA2s. Since the MLSA2s are enabled for a smaller duration, this scheme reduces the search time and the worst-case energy. The charge sharing between ML1 and ML2 begins at the rising edge of MLSO1 and ends at the falling edge of $\overline{MLOFF1}$ (T_{CS} in Figure 4.5(b)). The time needed to charge-share ML1 and ML2 depends on the size of transistors M1 and M2. Larger transistors equalize ML1 and ML2 faster. However, oversized transistors also increase the ML capacitance. Therefore, their sizes should be optimized by simulation.



(a)



(b)

Figure 4.5: Circuit schematic of the proposed charge-shared ML scheme using a current-race MLSA, and (b) its timing diagram

4.1.2.2. Chip Design and Measurement Results

We implemented the charge-shared ML scheme (illustrated in Figure 4.5) on a test chip in 0.18µm CMOS technology. A micrograph of the test chip is shown in Figure 4.6. It contains two 144-bit TCAM words and their dummies. One word and its dummy employ the standard current-race MLSA [32]. The other word and its dummy employ the current-race MLSA with the proposed charge-shared MLs (Figure 4.5). Typically, a full-size TCAM block (256x144) would only contain one dummy word. Thus, the signals $\overline{MLOFF1}$ and $\overline{MLOFF2}$ are shared by 256 words (Figure 4.5). In order to imitate this capacitive loading, we included dummy loads on the test chip. Figure 4.7 shows the search time and search energy of the conventional and the charge-shared ML schemes measured for a range of I_{BIAS} (Figure 4.5). Increasing I_{BIAS} reduces the search time but also increases the search energy. The charge-shared ML scheme gives a consistent improvement over the conventional scheme for the given range of I_{BIAS}.

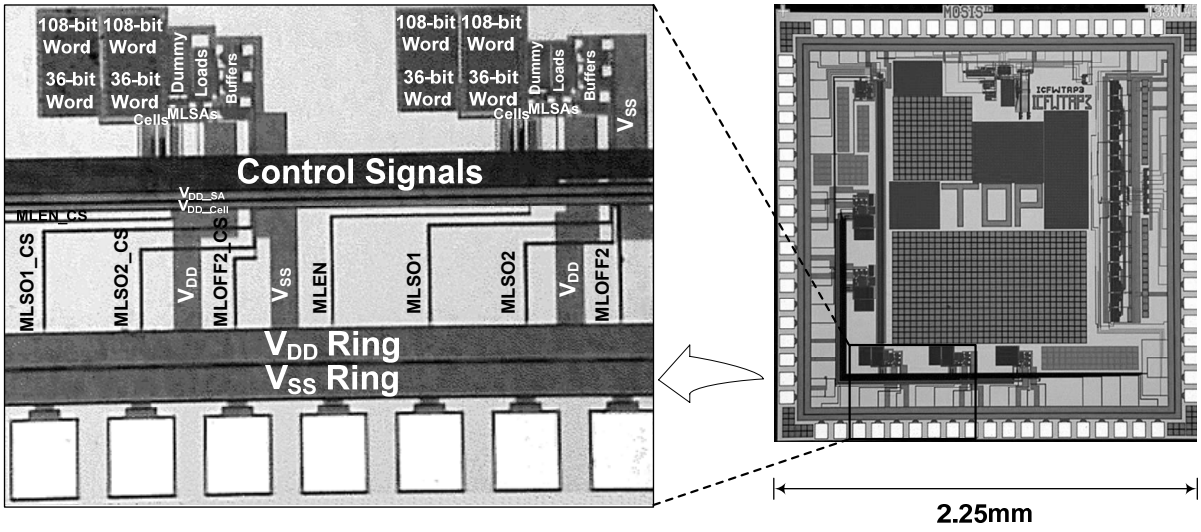


Figure 4.6: Micrograph of the test chip with conventional and charge-shared ML schemes

4.1.2.3. Analysis and Discussion

The chip measurement results in Figure 4.7 confirm the effectiveness of the charge-shared ML scheme in reducing the search time and search energy. For further improvements, a theoretical analysis can be performed to examine the above results and achieve optimum charge sharing between the ML segments. This analysis can also help to determine the optimum ratio of the two segments to minimize the search time and search energy.

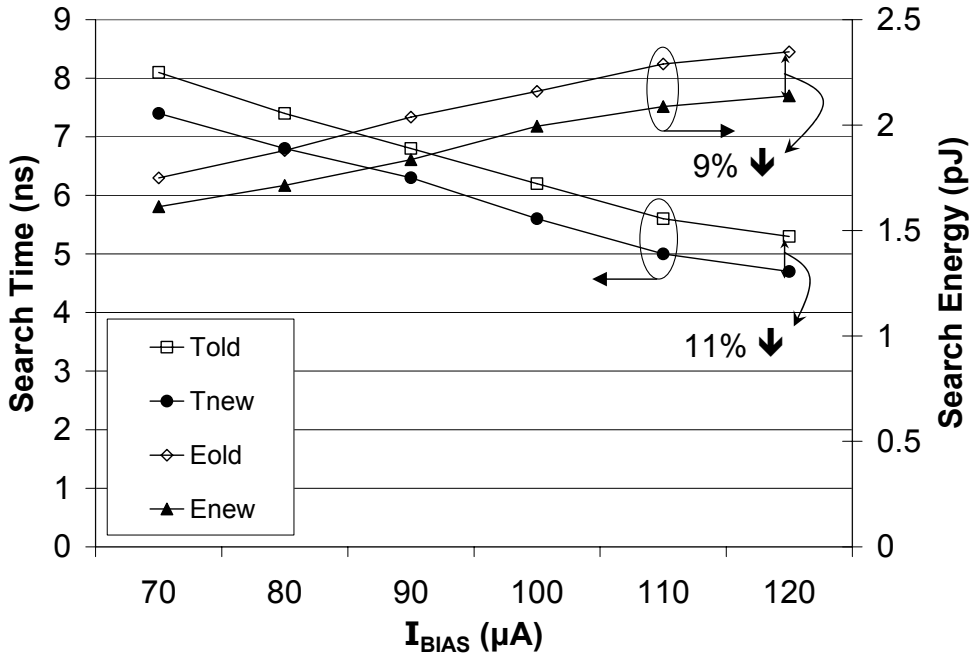


Figure 4.7: Search time and search energy measurement results of conventional and charge-shared ML schemes for different values of the bias current (I_{BIAS})

The energy reduction in the charge-shared ML scheme varies with the ratio of the ML1 to ML2 capacitances. If the ML1 and ML2 capacitances are C_{ML1} and C_{ML2}, and the ML voltage swing is V_{ML}, the total charge consumed in the worst-case ML sensing for a conventional ML can be expressed by equation (4.7):

$$Q_{OLD} = (C_{ML1} + C_{ML2})V_{ML} \tag{4.7}$$

Similarly, the total charge consumed in the worst-case ML sensing for a charge-shared ML can be expressed by equation (4.8):

$$Q_{NEW} = C_{ML1}V_{ML} + C_{ML2}(V_{ML} - V_{CS}) \tag{4.8}$$

where V_{CS} is the common-voltage of ML1 and ML2 after the charge sharing. V_{CS} can be calculated by applying charge conservation before and after the charge sharing as shown in equation (4.9):

$$V_{CS} = \frac{C_{ML1}V_{ML}}{(C_{ML1} + C_{ML2})} \tag{4.9}$$

Substituting V_{CS} from equation (4.9) and Q_{OLD} from equation (4.7), equation (4.8) can be re-written as equation (4.10):

$$Q_{NEW} = Q_{OLD} - \frac{C_{ML1}C_{ML2}V_{ML}}{(C_{ML1} + C_{ML2})} \tag{4.10}$$

Since energy is proportional to the charge drawn from the power-supply, the relative energy reduction of the charged-shared ML scheme (with respect to the conventional scheme) can be given by equation (4.11):

$$E_{Red} = \frac{(Q_{OLD} - Q_{NEW})}{Q_{OLD}} = \frac{C_{ML1}C_{ML2}}{(C_{ML1} + C_{ML2})^2} = \frac{\left(\frac{C_{ML1}}{C_{ML2}}\right)}{\left(1 + \frac{C_{ML1}}{C_{ML2}}\right)^2} \tag{4.11}$$

Figure 4.8 shows a plot of E_{Red} for different values of $\frac{C_{ML1}}{C_{ML2}}$. It reaches a maximum of 25% for $C_{ML1} = C_{ML2}$. Therefore, the charge-shared ML scheme is most suitable for TCAMs that have ML1 comparable to ML2. Substituting $\frac{C_1}{C_2} = \frac{36}{108} = \frac{1}{3}$ in equation (4.11), E_{Red} = 19%, which implies that the measured energy reduction (9%) is less than the theoretical value (19%). There are two possible reasons for this difference. First, the charge sharing time-window (T_{CS}), which is fixed and equal to an inverter-chain delay, might not be wide enough to fully equalize ML1 and ML2 (Figure 4.5). Second, in the present implementation, a current source (I_{BIAS}) charges ML1 during the charge sharing (Figure 4.5). Thus, the ML1 voltage is slightly higher than the ML2 voltage during the charge sharing. The charge sharing time-window could be optimized by using a digitally-controlled delay between DMLSO1 and $\overline{MLOFF1}$ [67]. The second issue can be eliminated by using the rising edge of MLSO1 to turn-off the corresponding I_{BIAS} during the charge sharing time-window (Figure 4.5).

In order to compare our results with the existing designs, we surveyed the published literature. The only published TCAM design with the current-race MLSA and chip measurement results is found in [32]. This 144-bit TCAM, also implemented in 0.18μm CMOS technology, achieves a search time of 3ns for I_{BIAS} = 260μA [32]. Our charge-shared ML scheme achieves a search time of 4.7ns for I_{BIAS} = 120μA (Figure 4.7). Extrapolating the above results, our scheme shows 27% improvement in speed for the same I_{BIAS}.

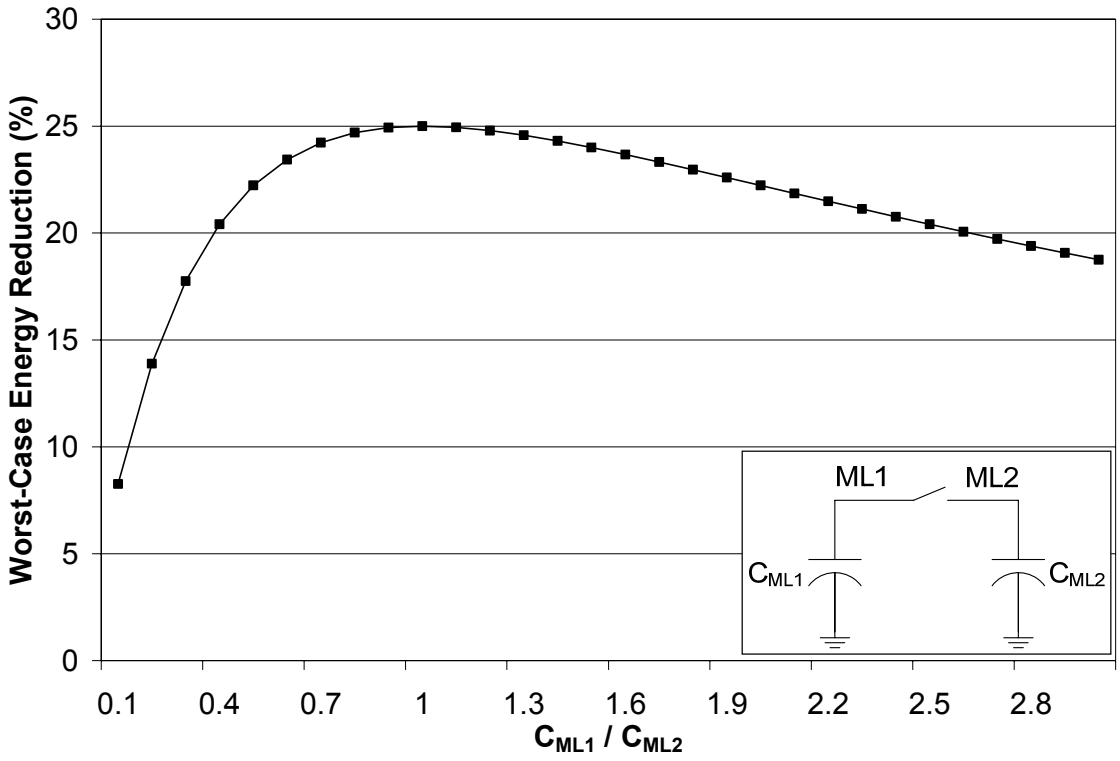


Figure 4.8: Worst-case energy reduction in charge-shared MLs for different values of the ML1 to ML2 capacitance ratio

4.2. Positive-Feedback Match Line Sense Amplifiers

In Chapter 2, we described some MLSAs that achieve power-savings by reducing the ML voltage swing. The current-race MLSA (CR-MLSA) is particularly attractive for high-speed and low-power operation. We analyzed the operation of the conventional CR-MLSA using the array of 144-bit TCAM words shown in Figure 4.9. In this implementation, instead of using the conventional comparison logic, we used the low-capacitance comparison logic (proposed in Chapter 3) to achieve high-speed operation at a lower energy. Before starting the SEARCH operation, the MLs are discharged to GND and the MLSA outputs (MLSOs) are reset to '0'. The SEARCH operation is initiated by enabling the ML current sources (I_{BIAS}) at the positive edge of MLEN signal. If a TCAM word matches with the search key, its ML does not have a current discharge path. Thus, it charges faster than the MLs with 1-bit mismatch or multiple-bit mismatch conditions (Figure 4.10). In rest of this chapter, we will denote matching MLs by ML_0 and MLs with a k-bit mismatch by ML_k (as shown in Figure 4.10), where $k \geq 1$. A dummy word, which matches in every search operation regardless of the

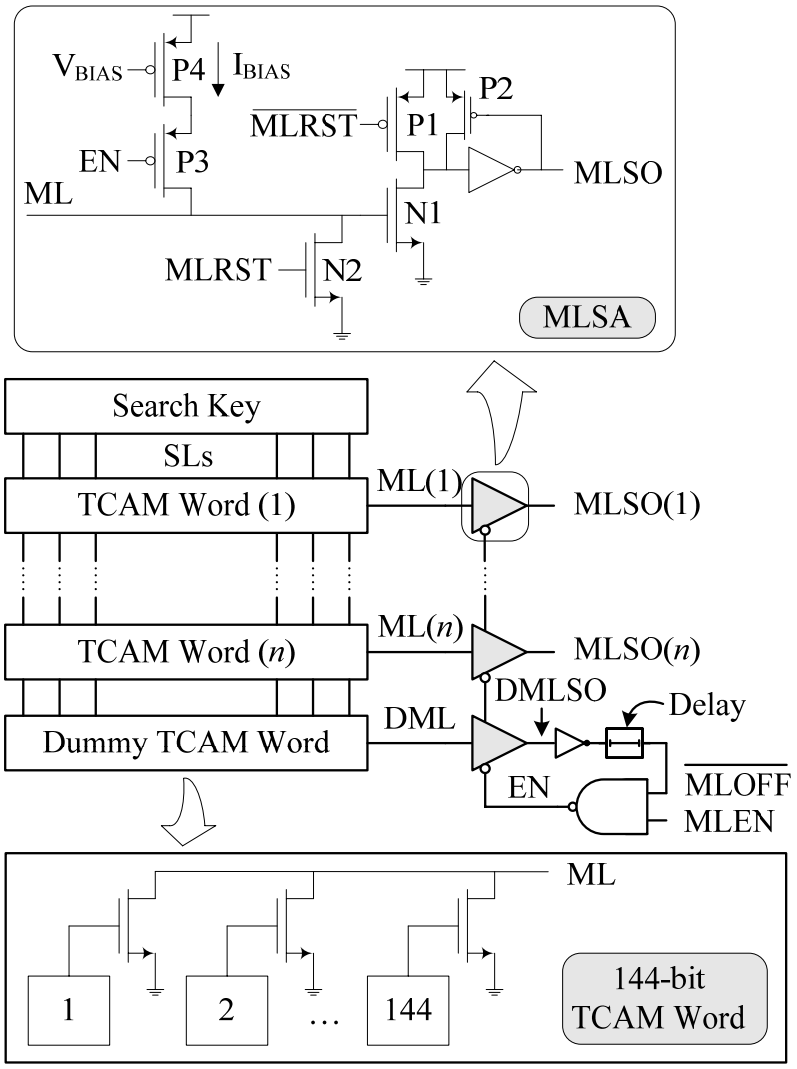


Figure 4.9: A TCAM array with the conventional current-race MLSA

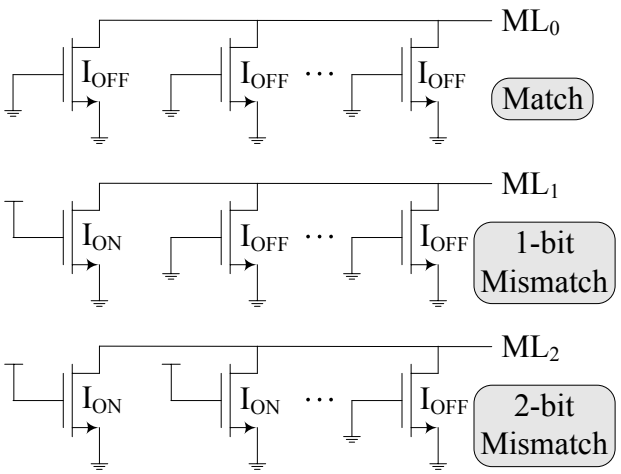


Figure 4.10: ML current discharge paths for different match/mismatch conditions

search key, generates a signal ($\overline{MLOFF}$) that turns off the current sources indicating the completion of the SEARCH operation.

The conventional CR-MLSA charges all MLs (ML_0 and ML_k) with the same magnitude of current (I_{BIAS}). The MLSO of ML_0 flips from '0' to '1' when the ML_0 voltage exceeds the NMOS threshold voltage (V_{tn}). Since the voltage of any ML_k does not exceed V_{tn}, its MLSO remains at '0' even after enabling the current-sources. Hence, the energy spent in charging ML_k's is wasted. This unnecessary power consumption forms a significant portion of the total power because most TCAM words do not match with the search key. In the conventional CR-MLSA, the speed is determined by the charging current of ML_0 (I_{ML0}) and the power is governed by the charging current of ML_k (I_{MLk}). Therefore, an ideal CR-MLSA should provide the maximum current to ML_0 (maximizing the speed) and the minimum current to ML_k (minimizing the power). A relatively smaller value of I_{ML1} (with respect to I_{ML0}) also improves the robustness of the MLSA by making it easier to detect the difference between ML_0 and ML_1. Thus, a small value of the $\frac{I_{ML1}}{I_{ML0}}$ ratio is desired for better robustness of ML sensing.

A mismatch-dependent MLSA (MD-MLSA) has been published by Arsovski et al. with simulation results showing 40% energy reduction over the conventional CR-MLSA [68]. However, there are several deficiencies in their implementation. First of all, the MD-MLSA consumes static power, which becomes significant in TCAMs employing low-power architectures for reduced chip activity (as described in Chapter 3). Second, it requires a level-shifter with skewed (large W/L and L/W ratios) PMOS transistors, which reduce the gain and bandwidth of the feedback-loop resulting in a slower transient response [69]. The slower operation decreases the non-uniformity between the I_{ML0} and I_{MLk} transients, which negatively affects the energy savings. Finally, the MD-MLSA circuit is difficult to reproduce in other technologies due to the circuit complexity. The complicated circuit of the MD-MLSA is prone to mismatches in repetitive structures such as TCAMs. The 40% energy reduction claims are also questionable because they have not been substantiated by the chip measurement results.

In this section, we present three novel CR-MLSAs that achieve power reduction by applying a positive feedback technique in ML sensing. If an ML is rising at a faster rate, the

positive feedback action ensures that it also receives a higher current. As a consequence, an ML_0 receives larger current than the current received by an ML_k ($k \geq 1$). This combination maximizes the speed and minimizes the power consumption. Unlike the MD-MLSA, the proposed MLSAs do not consume any static power. They also outperform the MD-MLSA in speed, energy, area and robustness as described in the following subsections.

4.2.1. Resistive-Feedback MLSA

Figure 4.11 shows the proposed MLSA with resistive feedback [70]. It uses an NMOS transistor (N3) in the triode region to decouple the ML and its MLSA. The N3 channel resistance shields the sensing point (SP in Figure 4.11) from the highly capacitive ML. This way the current source (I_{BIAS}) can be sized down to save power without sacrificing the sensing speed. It can be noticed that due to the body effect and the decreasing gate-to-source voltage (V_{GS_N3}), the N3 channel resistance increases when the ML voltage is rising up. Since the ML voltage rises faster as the value of k decreases, the increase in the N3 channel resistance is strongly affected by the number of mismatch bits (k). For instance, ML_0 would be rising faster than ML_1, which implies that the N3 of ML_0 has a higher resistance to shield the node SP. Since less current is now being diverted to the ML, the node SP charges much faster to reach the threshold voltage. Thus the increasing N3 resistance expedites the arrival of the corresponding MLSO ('0' → '1'). Faster sensing of the dummy word (emulating ML_0) also reduces energy consumption because the faster arrival of DMLSO (and hence $\overline{MLOFF}$) shuts-down the ML current sources sooner. The charging current of an ML_k is less affected by the N3 resistance because it has a larger V_{GS_N3} and a weaker body effect than ML_0. In other words, the N3 channel resistance creates a level-shift between ML and SP. As the ML voltage increases, the amount of level-shift also increases rapidly, and SP rises to the MLSA threshold voltage more quickly. Therefore, the overall effect is similar to a positive feedback between ML and SP.

The energy and delay of the resistive-feedback MLSA can be further reduced by decreasing V_{RES}. Although the positive feedback action results in a large voltage margin between ML_0 and ML_1, a combination of small V_{RES} and large I_{BIAS} may reduce the voltage margin causing a false match for ML_1. For example, a reduction in V_{RES} increases the N3 channel resistance, which may not be able to divert enough current to ML_1 (and subsequently

to GND) particularly if I_{BIAS} is large. As a consequence, the node SP of ML_1 may exceed the MLSA threshold voltage indicating a false match (MLSO: '0' → '1'). Figure 4.12 shows the effect of V_{RES} on the energy delay product (EDP) and the voltage margin for different I_{BIAS}. Similar to the "noise margin" in Chapter 3, the voltage margin is defined as the difference between the MLSA threshold voltage and the maximum voltage of the node SP associated with ML_1. Note in Figure 4.12 that a reduction in V_{RES} decreases the EDP more rapidly than the voltage margin. Hence, a moderate reduction in V_{RES} can improve the EDP without much effect on the voltage margin. Alternatively, for a small V_{RES}, a reduction in I_{BIAS} improves the voltage margin significantly without making much difference in the EDP.

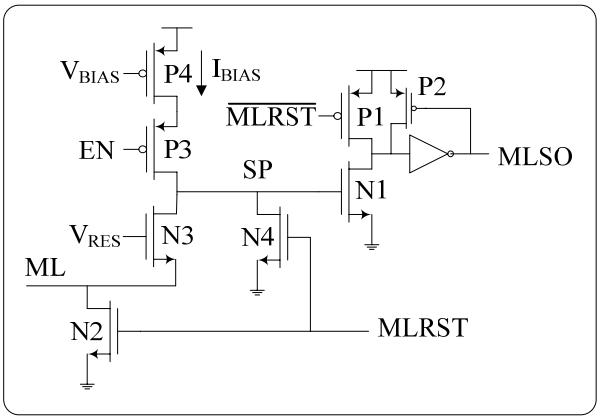


Figure 4.11: Proposed MLSA with resistive feedback

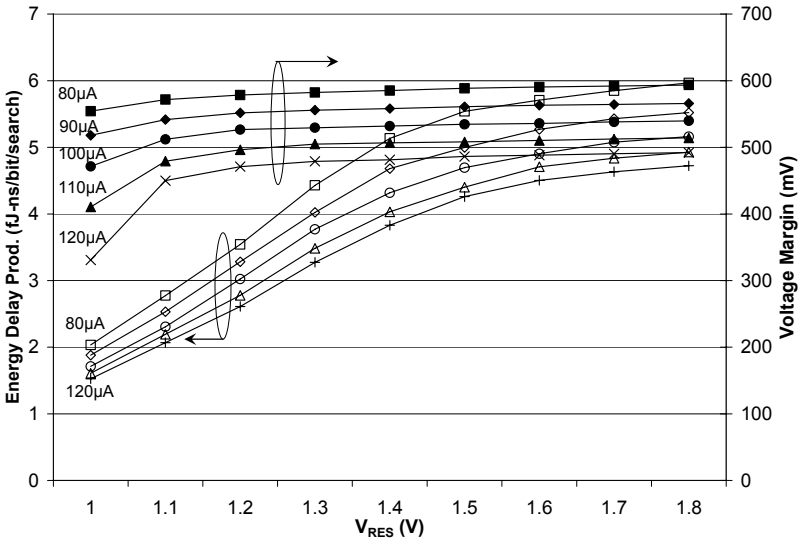


Figure 4.12: Effect of V_{RES} on energy delay product and voltage margin between ML_0 and ML_1

4.2.2. Active-Feedback MLSA

In order to reduce the EDP without sacrificing the voltage margin, we developed a CR-MLSA with active feedback. The proposed MLSA is shown in Figure 4.13. Transistor N3 operates as a constant current source (I_{FB}) to bias the feedback circuit. The MLEN signal (shown in Figure 4.9) enables the MLSA by activating EN, I_{BIAS} and I_{FB} (Figure 4.13). Initially, all MLs receive the same current from the current sources (I_{BIAS}). As ML_0 charges at a faster rate than ML_k, its P6 source-to-gate voltage (V_{SG_P6}) becomes smaller than that of ML_k. In order to keep the current through P6 constant (I_{FB}), a reduction in V_{SG_P6} is compensated by an increase in the P6 source-to-drain voltage (V_{SD_P6}). Since the source terminal of P6 is at V_{DD} (P7 is acting as a switch), a larger V_{SD_P6} results in a smaller V_{CS}. Thus, the faster charging of ML_0 makes its V_{CS} (V_{CS0}) smaller than that of ML_k (V_{CSk}). As a consequence, ML_0 receives more current and charges more rapidly than ML_k. This positive feedback action continues until DML (emulating ML_0) reaches the MLSA threshold voltage and switches DMLSO ('0' → '1'). This transition flips $\overline{MLOFF}$ ('1' → '0'), which turns off all of the current sources (as shown in Figure 4.9) by switching EN ('0' → '1').

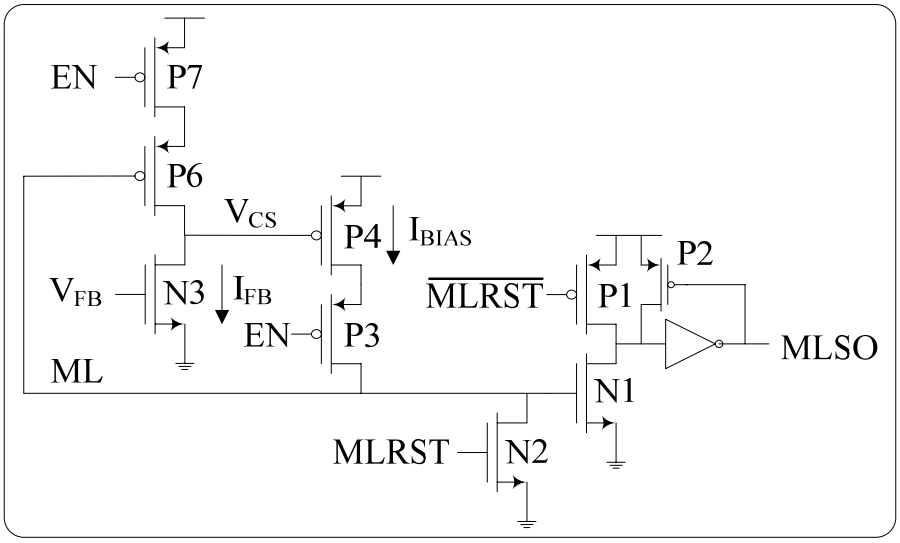


Figure 4.13: Proposed MLSA with active feedback

Figure 4.14 shows the effect of I_{FB} on the EDP and the ratio of average currents flowing into ML_1 and ML_0 (I_{ML1}/I_{ML0}). As mentioned earlier, the ratio I_{ML1}/I_{ML0} should be minimized to improve the robustness of the ML sensing. For $I_{FB} \geq 16\mu A$, the EDP does not

change significantly. However, I_{ML1}/I_{ML0} increases rapidly for I_{FB}>19μA. Thus, choosing I_{FB}=16μA or 17μA provides a good trade-off between the MLSA's performance and its robustness. Figure 4.15(a) shows the ML_0 and ML_1 voltage waveforms of a 144-bit TCAM word when it is simulated with the proposed active-feedback MLSA in 0.18μm CMOS technology. The positive feedback action starts around 250ps after enabling the current sources. Subsequently, ML_0 and ML_1 diverge significantly from each other. Figure 4.15(b) shows the current waveforms for ML_0 and ML_k (for k=1 through 6) highlighting two main features of the proposed scheme: (i) I_{ML0} is significantly larger than I_{ML1}, and (ii) I_{MLk} is a weak function of 'k'. The former confirms that the scheme has a large sense margin between ML_0 and ML_1, and the latter suggests that this scheme is equally applicable to TCAM applications where the average value of 'k' is small.

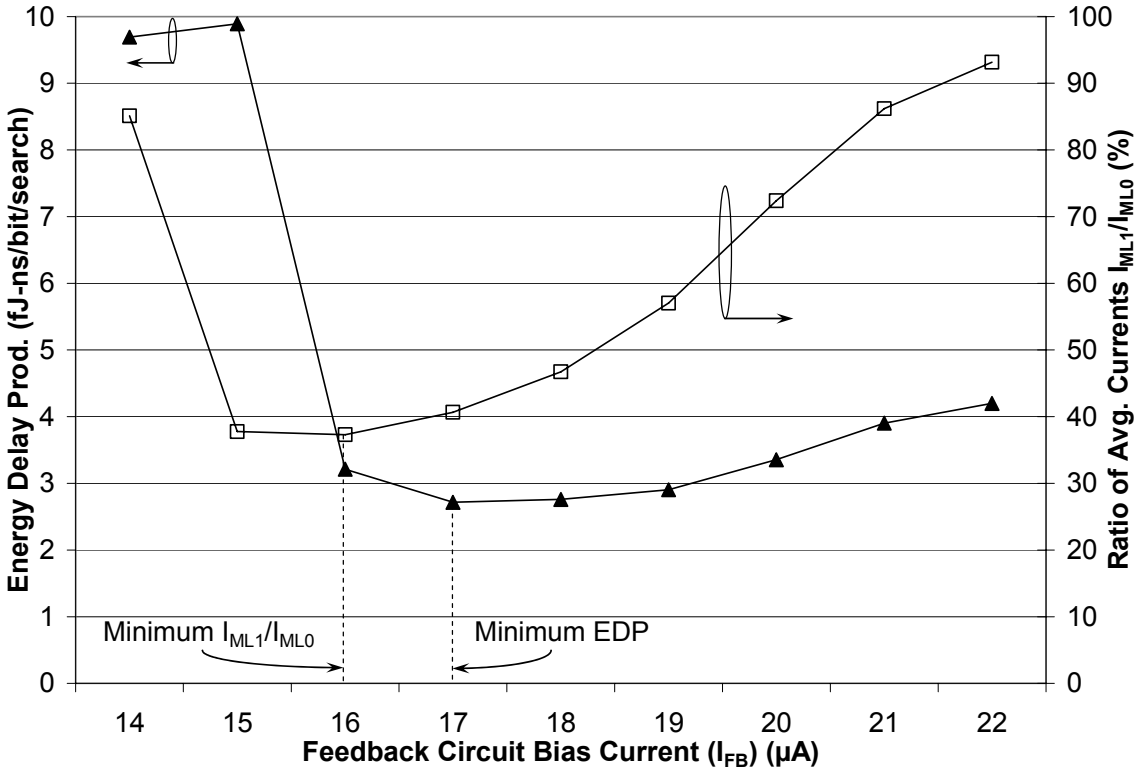
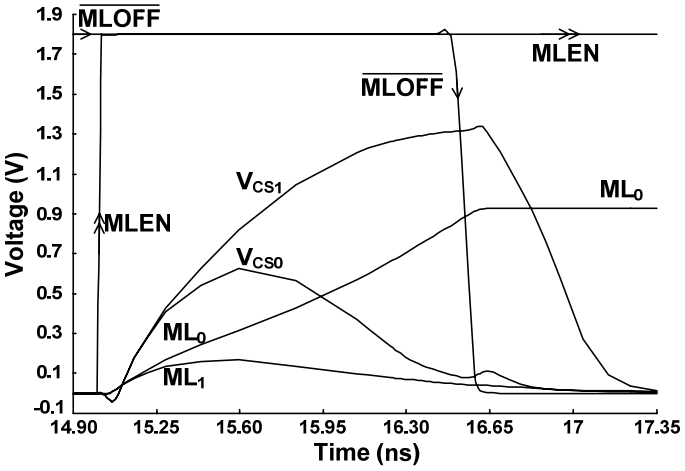
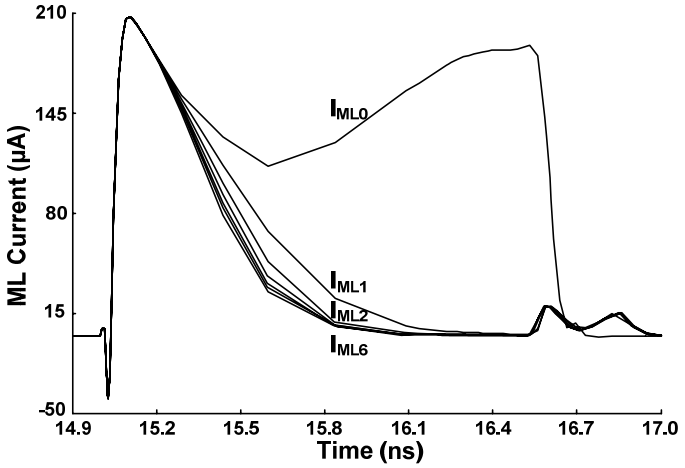


Figure 4.14: Effect of I_{FB} on energy-delay product and the ratio of average currents flowing into ML_1 and ML_2 (I_{ML1}/I_{ML0})



(a)



(b)

Figure 4.15: (a) Voltage, and (b) current waveforms of the proposed active-feedback MLSA

4.2.3. Active-Feedback MLSA with Body-Bias

The positive feedback action of the above mentioned MLSA can be further improved by modulating the MLSA threshold voltage with the ML voltage. It can be noticed in Figure 4.15 that the ML voltage always remains below 0.9V. Thus, the body effect can be exploited in favour of the positive feedback technique by connecting the ML to the body (substrate) of N1 and N3. For example, the higher voltage of ML_0, if connected to the body of N1, reduces the threshold voltage (V_{tn}) of N1, which results in a lower ML voltage swing and faster switching ('0'→'1') of the corresponding MLSO. It also expedites the arrival of $\overline{MLOFF}$ that turns off the current sources quickly and saves energy (Figure 4.9). Similarly, the higher voltage of ML_0 also reduces the threshold voltage (V_{tn}) of N3, which increases I_{FB} (and thus

decreases V_{CS}) resulting in a larger I_{BIAS} feeding ML_0. As a consequence, the body-bias further enhances the positive feedback action.

The circuit schematic of the proposed active-feedback MLSA is shown in Figure 4.16. Note that some current may be consumed by the body-source diodes, which become forward-biased if the ML voltage exceeds 0.7V. However, the ML voltage, already close to 0.7V, is expected to decrease further in modern technologies along with reductions in V_{DD}. Thus, the body-source diode current can be neglected. Figure 4.17 shows the voltage and current waveforms of 144-bit wide TCAM words employing the MLSAs with body-bias. As expected, the ML_0 voltage is limited to 0.7V. It should be noted that the waveforms shown in Figures 4.15 and 4.17 are drawn for the same speed of ML sensing (T_{MLSA}=1.5ns). Since the body-bias scheme has a stronger feedback, it can achieve the same T_{MLSA} for a smaller I_{FB}. As a consequence, after the initial transient currents, I_{ML0} of the body-bias scheme (Figure 4.17) settles at a smaller value than the I_{ML0} of the regular active-feedback scheme (Figure 4.15). However in the body-bias scheme, as the ML_0 voltage rises, it reduces the V_{tn} of N3, which increases I_{ML0} as described above.

In a typical n-well fabrication process, the body terminals of all NMOS transistors are connected to a common substrate. Thus, the body-bias technique is applicable only to those technologies that allow a transistor to have an independent body connection (not shared with other transistors). Some n-well fabrication processes have a deep n-well (DNW) layer that can be used to isolate the N1-body from rest of the substrate. The cross-sectional view of one such transistor is shown in Figure 4.18. Here, the combination of n-wells and DNW isolate the p-well (body of the NMOS transistor) from the p-substrate. In order to implement the proposed body-bias technique (shown in Figure 4.16), transistors N1 and N3 can be laid-out with DNW (as shown in Figure 4.18) with the gate terminal of N1 connected to the body of N1 and N3. However, the body of N1 and N3 has some capacitance with the surrounding n-wells, which are connected to V_{DD}. Therefore, this technique increases the ML capacitance while decreasing the ML voltage swing. It can achieve energy reduction only if the reduction in ML voltage swing is more prominent than the increase in ML capacitance. The resulting energy reduction is also highly dependent on the process technology. For example, the body-bias active-feedback MLSA implemented in a process technology with a larger DNW design rules and/or a higher capacitance per unit n-well area will consume more energy.

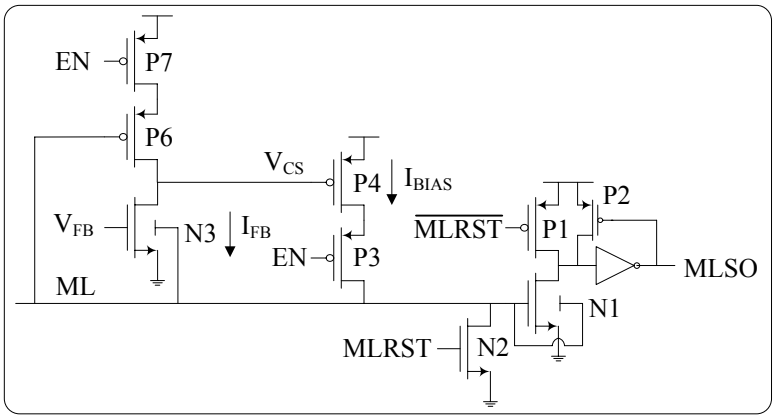
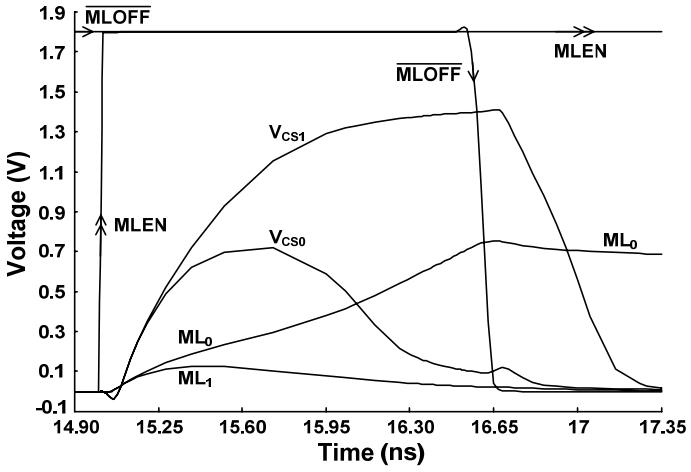
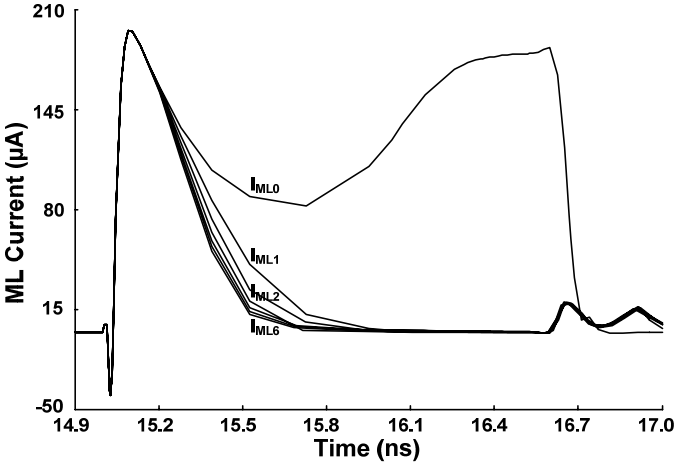


Figure 4.16: Proposed active-feedback MLSA with body-bias



(a)



(b)

Figure 4.17: (a) Voltage, and (b) current waveforms of the active-feedback MLSA with body-bias

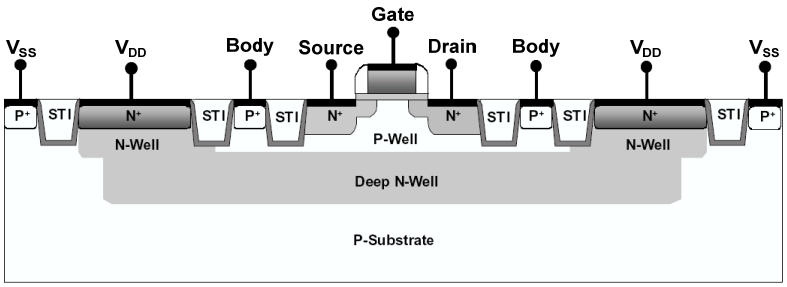


Figure 4.18: Cross-sectional view of an NMOS transistor with DNW (adapted from [71])

4.2.4. Simulation Results

We simulated 144-bit wide TCAM words employing the above mentioned MLSAs in 0.18µm CMOS technology. Figure 4.19 shows the energy simulation results comparing the proposed MLSAs with the conventional CR-MLSA for a T_{MLSA} of 1.5ns. As expected, the resistive-feedback MLSA consumes less energy for a smaller V_{RES} and remains functional for $V_{RES} \geq 1V$. The active-feedback MLSA shows a 43% reduction in ML sensing energy even for 1-bit mismatch (k=1). The energy reduction reaches 51% for higher numbers of mismatches. The body-bias technique further improves the energy efficiency of the active-feedback MLSA. For k=1, the body-bias technique shows 53% reduction in energy, which becomes 59% for higher number of mismatches. The simulation results reinforce the effectiveness of the body-bias technique.

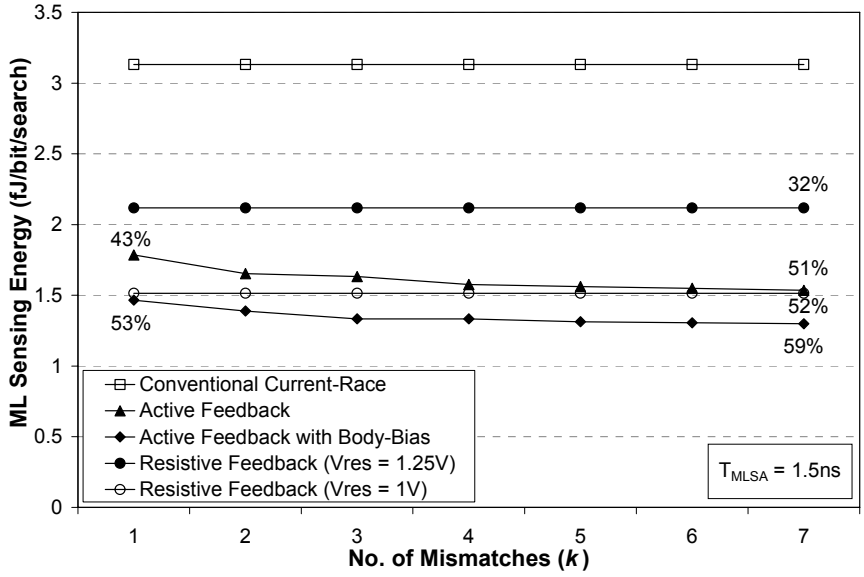


Figure 4.19: Simulation results of the conventional and proposed CR-MLSAs

4.2.5. Chip Design and Measurement Results

In order to substantiate the above mentioned theoretical analysis and simulation results, we implemented the conventional and proposed CR-MLSAs on a test chip (1mm x 2mm) in 0.18µm CMOS technology. A micrograph of the test chip is shown in Figure 4.20. It contains a 20Kb (144x144) TCAM array divided into 4 blocks. Blocks 1 and 2 contain 64 words each with the conventional and the proposed active-feedback CR-MLSAs, respectively. Blocks 3 and 4 contain 8 words each with the proposed resistive-feedback and body-bias active-feedback CR-MLSAs. In order to perform exhaustive testing of the test chip, we also included other peripheral components such as priority encoders, address and column decoders, registers, data multiplexers, scan chains, etc. The test chip was designed to perform only WRITE and SEARCH operations, and bit line sense amplifiers (for READ operations) were not included due to die-area constraints. The test chip was fabricated in TSMC CMOS 0.18µm technology with the DNW option. In order to implement the body-bias technique, we used the DNW layer to isolate the body of N1 and N3 (shown in Figure 4.16) from the rest of the p-substrate. However, this choice results in a fourfold increase in the MLSA area in TSMC 0.18µm CMOS technology due to large DNW design rules.

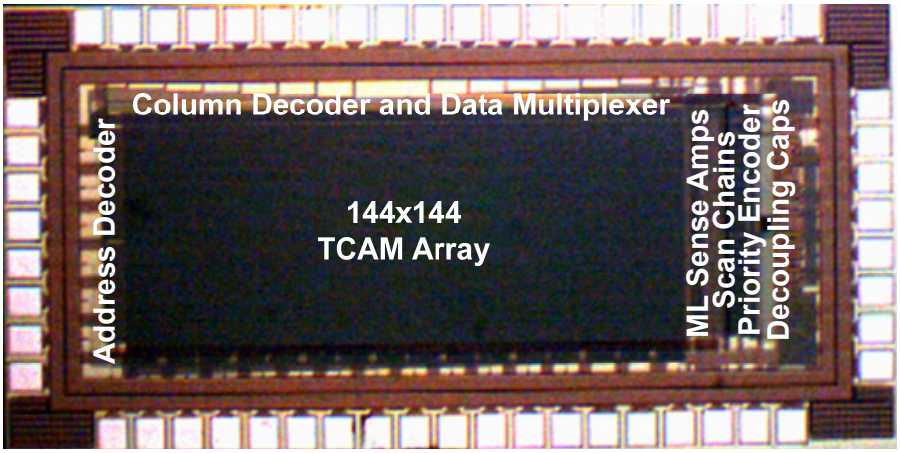


Figure 4.20: Micrograph of the test chip including a 144x144 TCAM with the conventional and proposed CR-MLSAs

Figure 4.21 shows the test chip measurement results comparing the energy consumption of the proposed MLSAs with that of the conventional CR-MLSA for T_{MLSA}=1.7ns at V_{DD}=1.8V. Note that the measured results are reasonably close to the

simulation results (Figure 4.19) except for the body-bias active-feedback MLSA. This MLSA shows degradation in energy savings possibly due to a large body capacitance (connected to MLs), which has not been modeled accurately in the simulations. Therefore, the body-bias technique is not suitable for the given process technology. However, this technique can be attractive in other process technologies such as silicon-on-insulator (SOI), where the body capacitance and area penalty are much smaller.

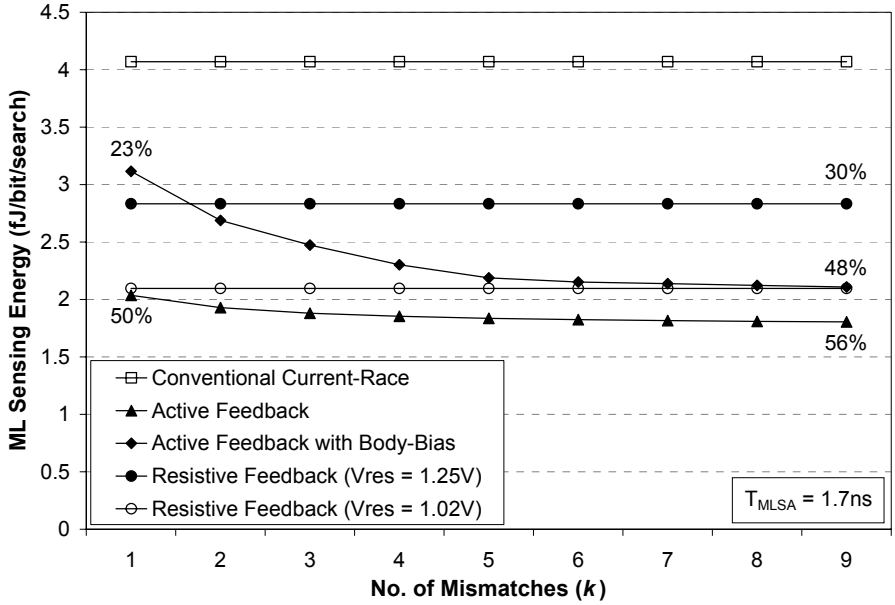


Figure 4.21: Chip measurement results of the conventional and proposed CR-MLSAs

During the chip measurements, it was found that the resistive-feedback MLSA remains functional for $V_{RES} \geq 1.02V$, and it achieves an energy reduction of 48% over the conventional CR-MLSA at $V_{RES}=1.02V$ ($I_{BIAS}=79.69\mu A$ and $T_{MLSA}=1.7ns$). If an additional design margin of 0.25V is provided ($V_{RES}=1.25V$, $I_{BIAS}=114.39\mu A$ and $T_{MLSA}=1.7ns$), the resistive-feedback MLSA still achieves an energy reduction of 30%. The active-feedback MLSA shows an energy reduction of 50% even for a 1-bit mismatch. The energy consumption further reduces to 56% for a higher number of mismatches. It should be noted in Figure 4.21 that the energy consumption is a weak function of the number of mismatches. Hence, the active-feedback MLSA is equally attractive for TCAM applications where the average number of mismatches is small. The MLSA sensing time $T_{MLSA}=1.7ns$ was obtained at $I_{FB}=16.58\mu A$ (shown in Figure 4.13) by EDP minimization. The body-bias active-feedback

MLSA and the conventional CR-MLSA achieve the same speed (T_{MLSA}=1.7ns) at I_{FB}=14.76μA and I_{BIAS}=141.37μA, respectively. The details of the chip design and test results will be covered in the next chapter.

4.2.6. Discussion and Comparison

In order to compare the proposed active-feedback MLSA with the existing MD-MLSA, we used the conventional CR-MLSA as the reference design for two main reasons: (i) the complicated circuit of MD-MLSA was difficult to reproduce in the current technology (0.18μm CMOS), and (ii) it was difficult to control the energy and delay of MD-MLSA for comparison purposes. Table 4.5 compares the active-feedback MLSA and MD-MLSA with reference to the conventional CR-MLSA. Unlike the active-feedback MLSA results, the MD-MLSA results are based on circuit simulations rather than chip measurements. It can also be noticed that the MD-MLSA results have been obtained in a smaller feature size technology.

Table 4.5: Comparison of MD-MLSA and the proposed active-feedback MLSA

S. No.	***Feature***	***MD-MLSA [68][69]***	***Active-Feedback MLSA***
1.	Process technology (μm CMOS)	0.13	0.18
2.	Type of results	Simulations	Measurements
3.	ML sensing time: T_{MLSA} (ns)	2.0	1.7
4.	Energy reduction with reference to the conventional CR-MLSA (%)	40	56
5.	$\frac{I_{ML1}}{I_{ML0}}$ ratio (%)	75[a]	50
6.	Static power consumption	Yes	No
7.	Number of additional transistors	5	3
8.	Feedback circuit bias current: I_{FB} (μA)	~1.6[a]	16.58
9.	Level shifter	Yes	No

[a] Deduced from [68]

The MD-MLSA consumes static power which is around 2% of its dynamic power [68]. Since the average ML current in the MD-MLSA is around 80μA, the feedback circuit consumes a static current of 1.6μA [68]. Many architecture-level techniques reduce the chip activity in TCAMs [35]-[37][40][52]. For example, selective-precharge and pipelined ML schemes divide each ML into two or more segments [36][37]. Each segment has a separate MLSA, and only one MLSA is activated in most of the words. Considering the large number of inactive segments, the static power in MLSAs can become a significant portion of the total TCAM power. Similarly, other techniques such as bank selection, paged-TCAM and EaseCAM activate only a small portion ($1/8^{th}$ or $1/64^{th}$) of a TCAM-based lookup table [35][40][52].

The static power in MD-MLSA can be eliminated by adding a switch between its feedback bias current source and the remaining circuit. However, the turn-on time of the feedback circuit will be reasonably large due to a small feedback bias current (~1.6μA). The proposed active-feedback MLSA turns-on faster due to relatively larger feedback bias current (I_{FB}=16.58μA). In addition, MD-MLSA uses a level-shifter (PMOS source follower) in the feedback loop, which reduces the feedback loop gain and bandwidth. The skewed sizing of two series-connected PMOS transistors $\left(\left.\frac{W}{L}\right|_{P1} = \frac{0.2\mu m}{2.0\mu m} \text{ and } \left.\frac{W}{L}\right|_{P2} = \frac{2.4\mu m}{0.13\mu m} \right)$ also increases the settling time of the level-shifter [69]. A smaller loop-gain reduces the sense margin between I_{ML0} and I_{ML1}, and also makes the circuit relatively slower. Table 4.5 confirms this deduction as the higher loop gain makes the proposed active-feedback MLSA outperform the MD-MLSA both in speed and sense margin even though it is implemented in a larger feature size technology.

The robustness of an MLSA is determined by its ability to detect the difference between discharge currents of ML_0 and ML_1. As illustrated in Figure 4.10, the ML discharge current increases with the number of mismatches. Since ML_0 has no mismatch, it has no discharge path to GND except the leakage currents (I_{OFF}). Since the transistor leakage and TCAM word-size are increasing due to technology scaling and new applications (such as IPv6 described in Chapter 1), respectively, the difference between the discharge currents of ML_0 and ML_1 is decreasing. Thus, detecting the difference between the ML_0 and ML_1 is becoming increasingly difficult. The proposed active-feedback MLSA is more robust than

MD-MLSA because it has a larger sense margin (I_{ML1} is 50% of I_{ML0}). The larger sense margin also helps in coping with process variations, which are increasing with technology scaling.

4.2.7. Effects of Process Variations on Active-Feedback MLSA

The active-feedback MLSA achieves fast and energy-efficient ML sensing due to the positive feedback action. However, positive-feedback systems are usually sensitive to process variations. In this sub-section, we present process variation analysis and worst-case simulations of the active-feedback MLSA. Our discussion is limited to the active-feedback MLSA because a stronger feedback makes it more susceptible to process variations than the resistive-feedback MLSA.

Similar to the conventional current-race MLSA, the noise margin of the active-feedback MLSA is defined by the difference between the MLSA threshold voltage (V_{th}) and the maximum voltage of ML_1. If ML_1 reaches V_{th} before the current sources are turned off (i.e. zero or negative noise margin), it will be detected as ML_0. Since the value of V_{th} also has some uncertainty due to the process variations, correct ML sensing requires that ML_1 remains below the minimum possible value of V_{th} (V_{th_MIN}). For example, if a process technology exhibits the maximum V_{th} variation of ΔV_{th}, the MLSA should have a positive noise margin greater than ΔV_{th}. It can be observed from Figure 4.10 that the ML_1 pull-down path is an NMOS transistor in the triode region ($V_{GS} = V_{DD}$, $V_{DS} < V_{DD}/2$). Therefore, the current I_{ON} increases with ML_1 voltage, and the NMOS transistor can be modeled as a resistor (R_{ON}) as shown in Figure 4.22.

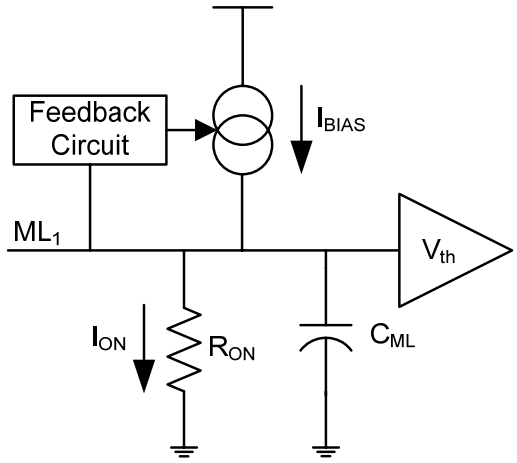


Figure 4.22: Simplified model of an ML_1 (1-bit mismatch) with the active-feedback MLSA

In the beginning of ML sensing, I_{ON} is zero (because $V_{ML1} = 0$), and the entire current I_{BIAS} is used for charging ML_1 capacitance (C_{ML}). Rising ML_1 voltage also increases I_{ON} that diverts some portion of I_{BIAS} leaving a smaller charging current ($I_{BIAS} - I_{ON}$) for ML_1. In the steady-state condition, I_{ON} becomes equal to I_{BIAS}, and ML_1 voltage saturates to a value $V_{ML1} = I_{BIAS} \times R_{ON}$. If this steady-state value of V_{ML1} is less than V_{th_MIN}, the correct MLSA functionality can be guaranteed. Hence, for a given value of R_{ON}, I_{BIAS} can be reduced such that V_{ML1} remains below V_{th_MIN}. However, the trade-off in this case is slower ML sensing due to a smaller I_{BIAS}. It can be noticed that the above analysis is overly pessimistic because it does not consider the transient response of the MLSA. In reality, the current source (I_{BIAS}) is turned off as soon as V_{ML0} reaches V_{th}. As a consequence, V_{ML1} may not reach the steady state, and the maximum value of V_{ML1} may remain below the steady-state value ($I_{BIAS} \times R_{ON}$). Therefore, the noise margin should be determined using the transient response simulation of the MLSA under the process variations that result in the fastest charging of ML_1. It can be noticed in Figure 4.22 that following process variations will increase the rate of charging ML_1:

- An increase in I_{BIAS}
- A reduction in C_{ML}
- An increase in R_{ON}

We performed simulations with the above non-idealities in ML_1 while keeping ML_0 under the typical conditions as shown in Figure 4.23. The simulation results under different process variations along with their simplified models are summarized in Table 4.6.

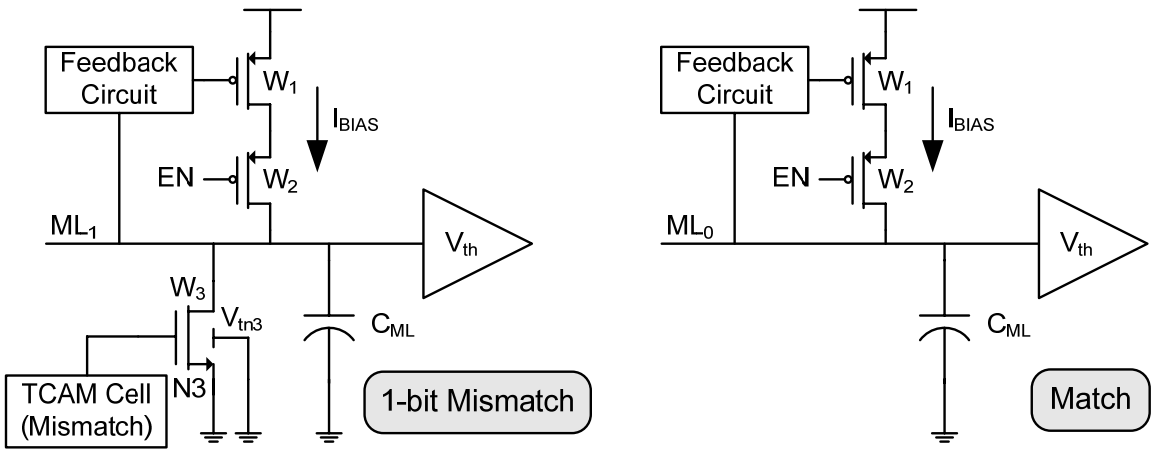


Figure 4.23: Simplified models of ML_1 and ML_0 with typical values of circuit parameters

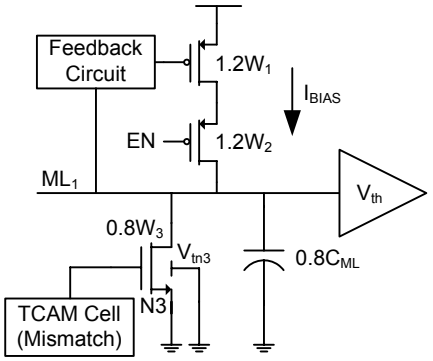
Table 4.6: Worst-case simulation results under different process variations

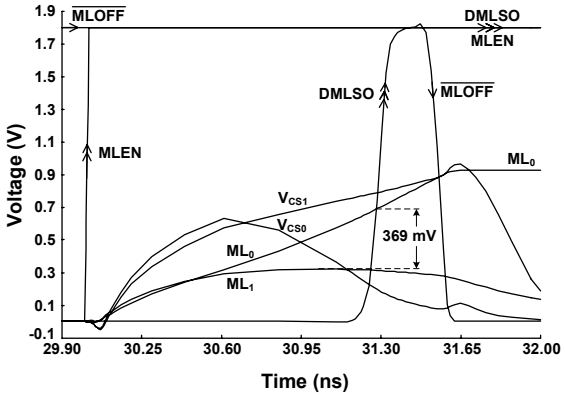
<table>
<tr>
<td><u>Case I</u>

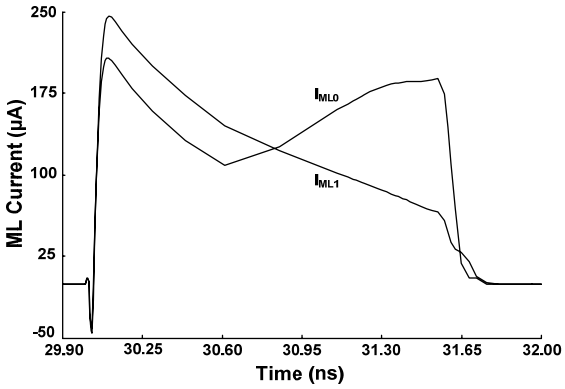
W_1:
20 % (↑)

W_2:
20 % (↑)

W_3:
20 % (↓)

C_{ML}:
20 % (↓)</td>
<td> 

ML1 with process variations in transistor sizing and CML</td>
<td> 

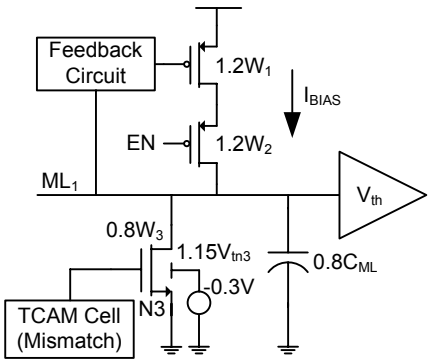
 
</td>
</tr>
<tr>
<td><u>Case II</u>

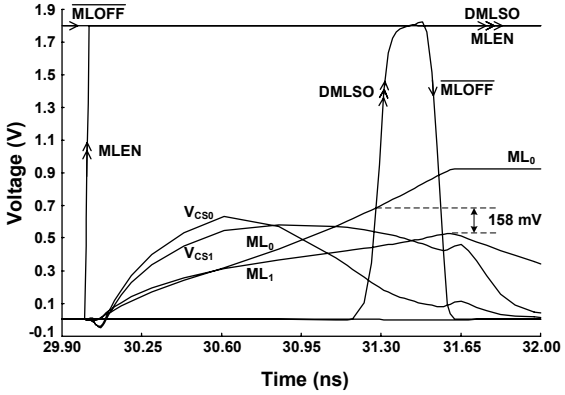
W_1:
20 % (↑)

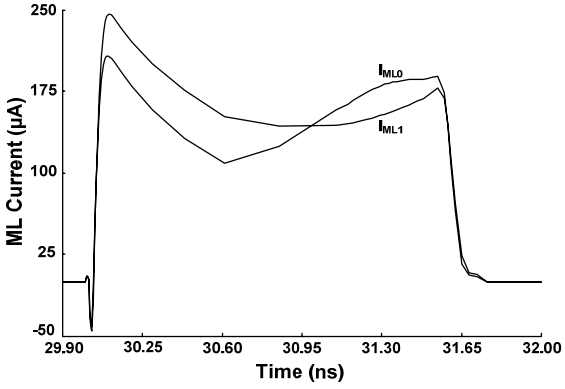
W_2:
20 % (↑)

W_3:
20 % (↓)

C_{ML}:
20 % (↓)

V_{tn3}:
15 % (↑)</td>
<td> 

ML1 with process variations in transistor sizing, CML and Vtn (simulated using body-bias)</td>
<td> 

 
</td>
</tr>
</table>

Case I in Table 4.6 incorporates the process variations in the device geometries and ML capacitance to favor the fast charging of ML_1. In these simulations, we increased the sizes of current source transistors (W_1 and W_2) by 20%, and reduced the size of pull-down transistor N3 (W_3) and ML capacitance (C_{ML}) by 20%. It should be noticed from the voltage and current waveforms that ML_1 initially receives a higher current (I_{ML1} is larger than I_{ML0}), and ML_1 rises faster than ML_0. However, the pull-down current of ML_1 also increases rapidly, and ML_1 is pulled below ML_0 within 0.6ns. Beyond this point, the gap between ML_1 and ML_0 increases due to the positive feedback action that finally results in a noise margin of 369mV. This voltage margin is large enough to handle significant variations in the MLSA threshold voltage (V_{th}).

Case II in Table 4.6 further speeds up the ML_1 charging by incorporating threshold voltage variation in the ML_1 pull-down path (transistor N3). Besides considering all the non-idealities included in Case I, Case II also increases the threshold voltage of transistor N3 (V_{tn3}) by 15%. In SPICE simulations, this increase in V_{tn3} can be achieved by applying a negative body-bias. For example, the application of -0.3V to the body of the NMOS transistor N3 increases V_{tn3} from 521mV to 597mV (76mV or 15%) as illustrated in Table 4.6. As expected, ML_1 rises faster in this case due to further reduction in the pull-down current. However, the pull-down current is still large enough to bring ML_1 below ML_0 within 0.8ns. Beyond this point, the gap between ML_0 and ML_1 widens due to the positive feedback action that finally results in a noise margin of 158mV. Although noise margin is smaller in this case (as expected), it is still large enough to handle significant V_{th} variations. These worst-case simulations confirm that the proposed active-feedback MLSA is immune to process variations mainly due to the ML_1 pull-down path, which creates a natural gap between ML_0 and ML_1. The positive feedback action further widens this gap making it more robust to V_{th} variations in MLSAs.

4.3. Conclusions

In this chapter, we proposed two low-power ML-segmentation techniques and three low-power MLSAs. Both ML-segmentation (dual ML and charge-shared ML) schemes divide a large ML into two smaller segments and sense them sequentially. The sequential SEARCH operation may increase the search time. However, the speed penalty is not significant for

large-size segments since the charging (or discharging) time of the highly capacitive ML is much larger than the MLSA propagation delay.

The dual ML scheme shows a 43% reduction in ML energy for a small (4%) trade-off in the search speed if it contains five or more mismatches in most words. However, this scheme has a theoretical limit of 50% reduction in ML energy. Actual energy savings depend on the layout and number of mismatches per word. In densely laid-out TCAMs, if the ML1 segments are routed very close to the other parallel lines, the coupling capacitance increases significantly, and this scheme loses its effectiveness. The ML1 capacitance can be reduced by routing the ML1s at a larger distance from the other parallel lines. Such a routing can increase the cell area, which may not be suitable for large-capacity TCAMs. Therefore, the dual ML is more attractive than the selective-precharge only when the data statistics is unpredictable.

The charge-shared ML scheme reduces the search time and the worst-case energy over the conventional ML by 11% and 9%, respectively. We analyzed the measurement results and proposed possible improvements. For example, the charge-shared ML scheme can achieve better results for MLs with a larger first segment, a digitally-controlled charge sharing time window, and a slightly modified MLSA as discussed in subsection 4.1.2.3. Although the charge-shared ML scheme has been demonstrated for the selective-precharge type of ML-segmentation, it can be easily extended to the dual ML scheme or even to multiple segments. Therefore, a segment should be sensed after charge sharing it with the previous segment so that the charge in the previous segment is partially recycled before it is discarded to GND.

The three MLSAs presented in this chapter use positive feedback to reduce the power consumption in ML sensing. The resistive-feedback MLSA inserts one additional transistor in the conventional CR-MLSA and exploits the body effect to reduce the EDP without affecting the voltage margin. The active-feedback MLSA employs a separate feedback circuit (containing three transistors) whose gain and bandwidth are controlled by its bias current (I_{FB}). A modified version of the active-feedback MLSA uses body-bias to reduce the MLSA threshold voltage and hence the ML voltage swing, which decreases the delay and energy of ML sensing. Although the body-bias scheme shows promising results in circuit simulations, test chip measurement results are not attractive. It was observed that the large DNW design

rules of CMOS 0.18µm technology increase the ML capacitance, which offsets the energy savings obtained by the reduced voltage swing. The body-bias technique also results in a fourfold increase in the MLSA area in CMOS 0.18µm technology. Therefore, it is attractive in technologies (such as SOI), where the body capacitance and area penalty are much smaller.

We compared our test results with the existing MD-MLSA that also achieves power reduction by exploiting the positive-feedback technique. The proposed active-feedback MLSA requires fewer transistors than the MD-MLSA. Furthermore, the active-feedback MLSA improves energy savings and voltage margin without consuming any static power. Energy measurement results of the active-feedback and resistive-feedback MLSAs show a reduction of 56% and 48%, respectively over the conventional CR-MLSA. We also analyzed the effects of process variations on the active-feedback MLSA. We found that even in the presence of 20% variations in the device geometries and 15% variation in the NMOS threshold voltage, the active-feedback MLSA has a noise margin of 158mV, which is large enough to handle significant variations in the MLSA threshold voltage. In this chapter, the discussion was limited to ML sensing only. The following chapter provides complete design and test details of a TCAM chip including the TCAM array, MLSAs, priority encoders, address and column decoders, registers, data multiplexers, scan chains, etc.

Chapter 5

TCAM Chip Integration

In previous chapters, we discussed techniques for designing low-power high-performance components for a TCAM chip. This chapter describes design and test considerations when these individual components are integrated on a TCAM chip. Although well-designed components help in implementing a low-power high-performance chip, several chip-level design tradeoffs also affect the overall speed and power consumption of a TCAM. These trade-offs include area-efficient layout, component placement, interconnect widths, and the selection of metal layers for different lines. The components should be placed to minimize the chip-level routing. The width of high-current interconnects should be determined from simulations since excessive current through a narrow wire may cause permanent interconnect failure due to electromigration [21]. The metal selection and placement of high-activity lines (such as SLs and MLs) should be done such that their capacitances are minimized. In order to improve the matching among similar components, analog layout techniques may be adopted [72]. A complex integration of memory and logic makes TCAM testing a complicated process. Hence, an extensive test plan should be developed during the design phase itself. For

better observability and controllability of the TCAM chip, design for testability (DFT) components should also be included on the chip. DFT structures may also be used to execute various TCAM test algorithms [73].

5.1. Chip Design

A TCAM array is a repetitive structure. Thus, the TCAM cell layout should be optimized for minimum area. In addition, the layouts of the peripheral components should be pitch-matched with the TCAM array. Since the TCAM array consumes the biggest portion of the chip, its layout helps in estimating the final chip area. As explained in Chapter 3, the TCAM is usually employed as a lookup table with the lookup rate much higher than the update rate. Hence, the TCAM design is mainly focused on the components and interconnects in the search path. The BLSAs are used to perform the READ operation, which is done only when a tester runs test algorithms on the TCAM chip to make the pass/fail decision. Therefore, BLSAs may be omitted in a prototype or test chip.

In our design, we did not include BLSAs on the test chip due to die-area constraints, and the chip was designed only to perform SEARCH and WRITE operations. The top-level block diagram of the test chip is shown in Figure 5.1. Besides demonstrating the integration of different components, this chip was also designed to examine the energy savings of the three positive-feedback MLSAs over the conventional CR-MLSA (described in Chapter 4). Hence, the 20Kb (144x144) TCAM array was divided into four blocks: (i) Block 1 contained 64 words employing the conventional CR-MLSA, (ii) Block 2 contained 64 words employing the active-feedback MLSA, (iii) Block 3 contained eight words employing the resistive-feedback MLSA, and (iv) Block 4 contained eight words employing the body-bias active-feedback MLSA. We employed the active-feedback MLSA in the larger block for better measurement accuracy because it was the most promising MLSA among the three novel designs (demonstrated in Chapter 4). Blocks 3 and 4 were sized smaller due to die area constraints. Blocks 1 and 2 were connected to 64-bit PEs, and blocks 3 and 4 were connected to 8-bit PEs.

In order to perform exhaustive testing of the TCAM chip, we also included other peripheral components such as address and column decoders, registers, data multiplexers, scan chains, etc. The large on-chip output buffers may cause V_{DD} or GND bouncing when

they draw large transient currents from the power supplies [21]. In order to alleviate this issue, we reduced the number of output pins and noisy output buffers using scan chains and multiplexers. Several design for testability features were added to the chip in order to facilitate the testing of individual components even when one or more components are not functional. Similarly, extra logic was added on the chip for multiple ways of testing individual components and for easy insertion of the test vectors. The individual components of the test chip will be described in the following subsections.

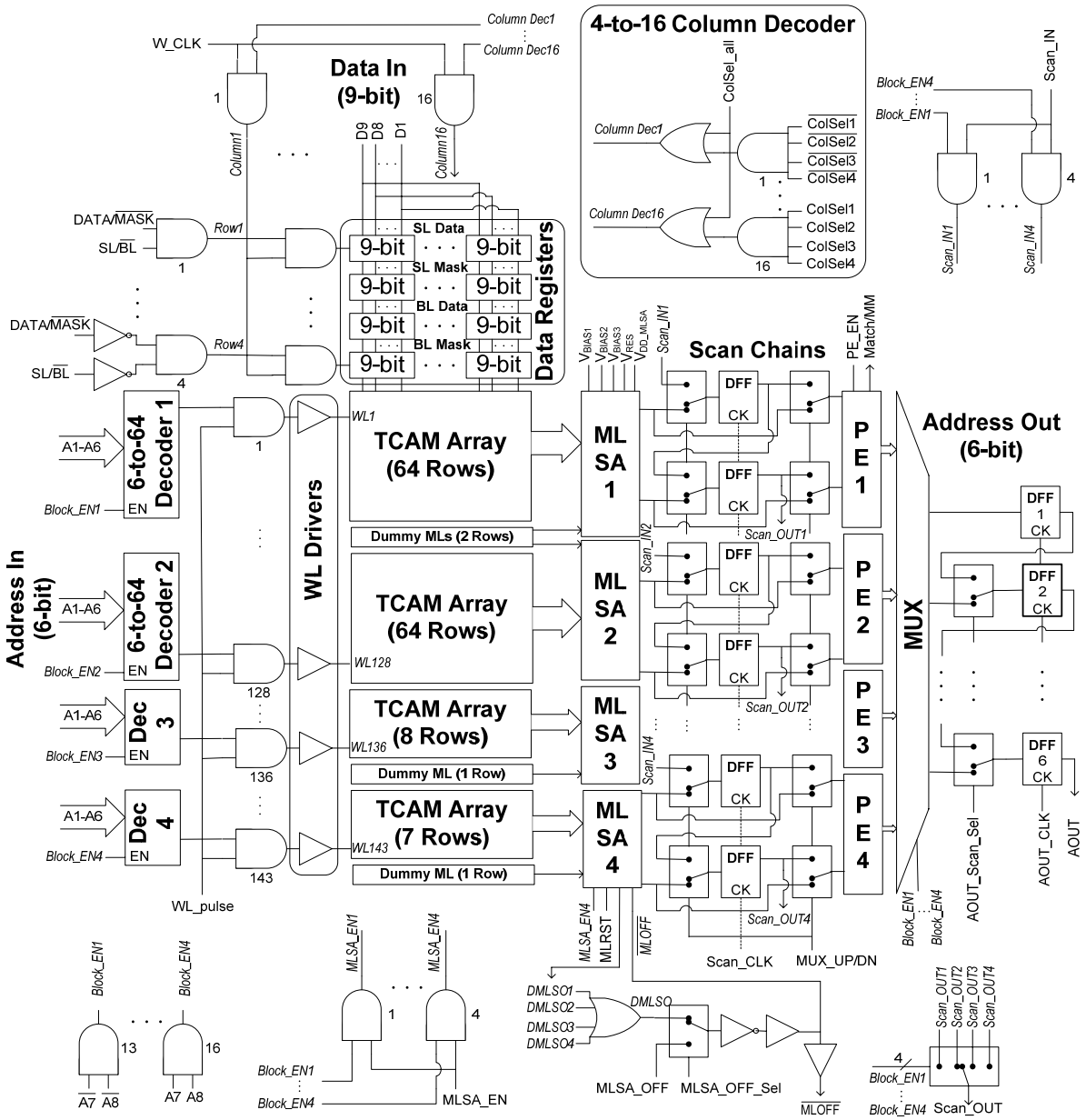


Figure 5.1: Top-level block diagram of the test chip

5.1.1. TCAM Array

As described earlier, the 144x144 TCAM array was divided into four blocks (see Figure 5.1). Each row in the array contains 144 low-capacitance TCAM cells described in Chapter 3 (Figure 3.17). In order to minimize the ML capacitance, the N3 transistors of two adjacent cells shared the same drain contact (Figure 3.22). The schematic and layout of two adjacent TCAM cells are shown in Figure 5.2. Each TCAM cell contains two SRAM cells, and the comparison logic circuits are placed in the area between the two TCAM cells.

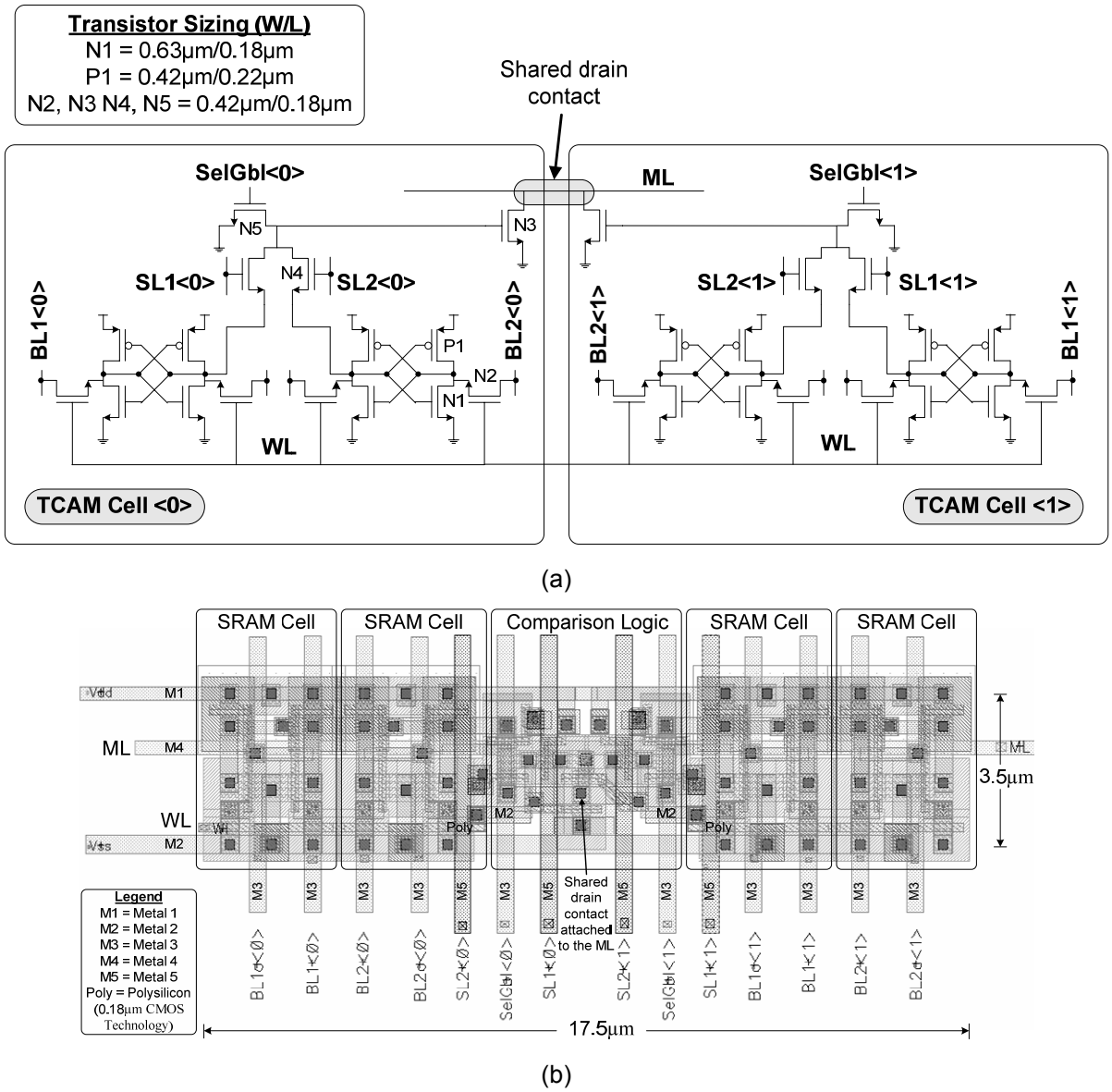


Figure 5.2: (a) Schematic and (b) layout of two adjacent TCAM cells with a shared drain contact for the comparison logic transistors connected to the ML

The SRAM area was minimized by choosing minimum size transistors $\left(\frac{W}{L}=\frac{0.42\mu m}{0.18\mu m}\right)$ wherever possible. Originally, the cells were designed to perform the READ operation as well. Thus, the driver transistors (N1 in Figure 5.2(a)) were sized 1.5 times larger than the access transistors (N2). Note in Figure 5.2(b) that the ML is routed in a higher-level metal (M4). This choice results in a smaller ML-to-bulk capacitance. ML coupling capacitance was minimized by routing other parallel lines in layers other than M4. For example, V_{DD}, V_{SS} and WL were routed in M1, M2 and Poly, respectively. Since the speed of the WRITE operation is not important in TCAM applications, the WLs were routed mainly in Poly with small segments of M2. Similarly, the SLs were routed in M5 to minimize the SL-to-bulk capacitance. Since other parallel lines (BLs and SelGbls) have much smaller switching activities, they were routed in M3. Vertically adjacent cells were joined after flipping them vertically. This arrangement minimizes the area by ensuring that their V_{DD} or V_{SS} metals are overlapped. The TCAM array was generated using an array of instances in *Cadence Virtuoso Layout Editor* [74].

5.1.2. Match Line Sense Amplifiers

As mentioned earlier, the four blocks in the test chip employ four different types of MLSAs: conventional CR-MLSA, active-feedback MLSA, resistive-feedback MLSA and body-bias active-feedback MLSA. In the conventional CR-MLSA, shown in Figure 5.3(a), the ML current source (I_{BIAS}) was implemented using large-size PMOS transistors to support a current that is high enough to match the speed of the positive-feedback MLSAs. In order to achieve pitch-matching with the TCAM array (3.5µm), such large transistors were laid-out in multi-finger fashion (Figure 5.3(b)). A weak transistor P2 was included to compensate for N1-leakage while holding the node MLSO at '0'. Transistor N1 was sized relatively large to override P2 at a smaller ML voltage, which results in a lower energy consumption and a smaller propagation delay. The common control signals (shared by all MLSAs) were routed vertically. Similar to the TCAM array, two vertically adjacent MLSAs were joined by overlapping their V_{DD} or V_{SS} metals.

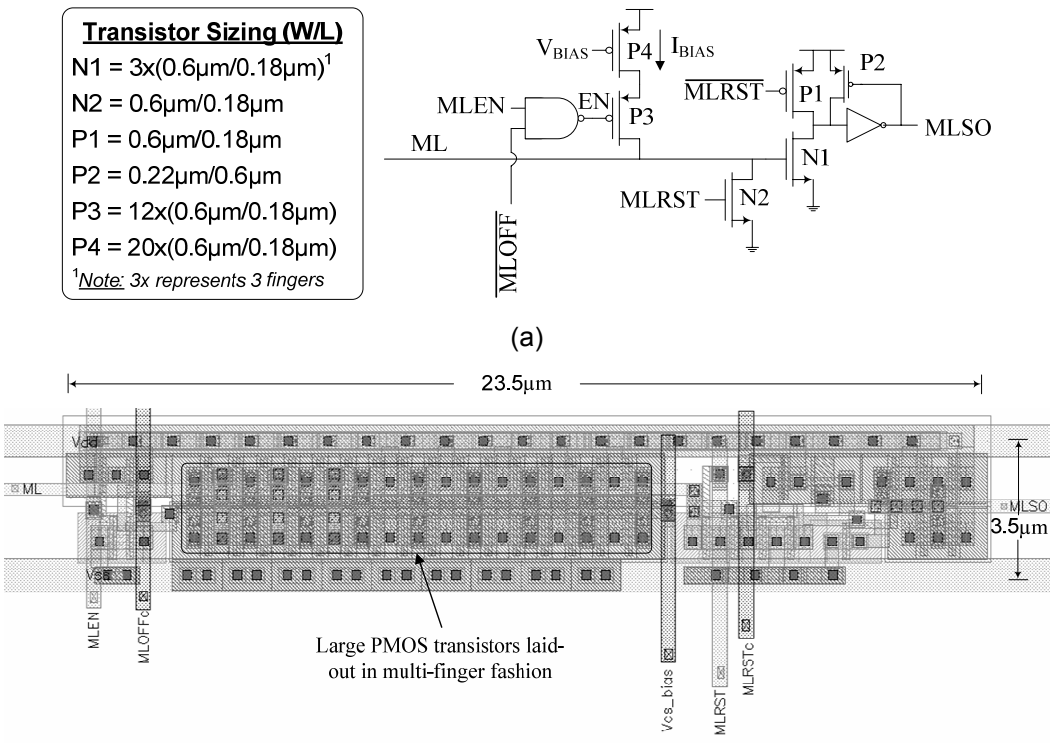
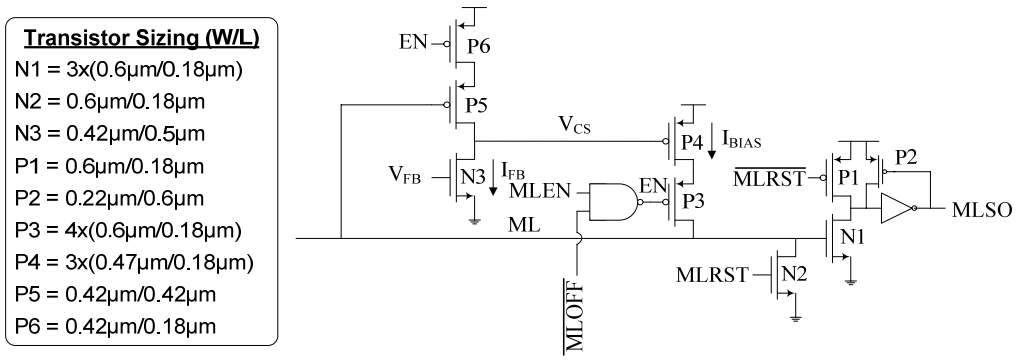


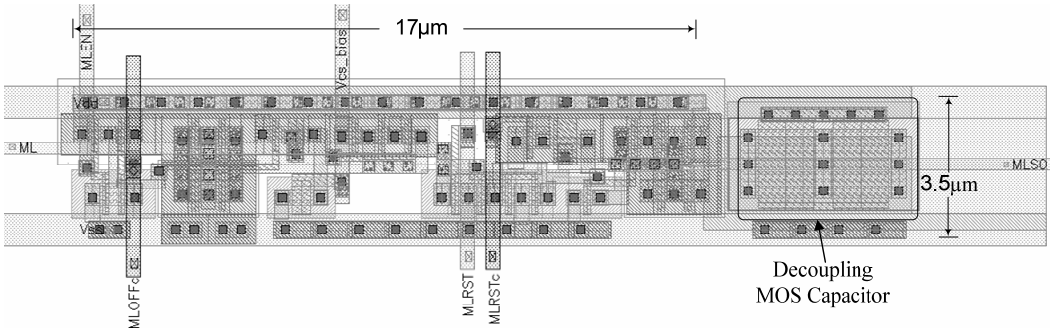
Figure 5.3: (a) Schematic and (b) layout of the conventional CR-MLSA (employed in Block 1)

Figure 5.4 shows the circuit schematic and layout of the active-feedback MLSA. Since this MLSA uses a non-linear current source, it does not require large-size PMOS transistors and consumes less area than the conventional CR-MLSA. When Block 1 and Block 2 were implemented in a column, the smaller MLSAs in Block 2 resulted in some empty area. We inserted decoupling capacitors in this empty area. The decoupling capacitors help in stabilizing the power-supplies when large transient currents are drawn by the MLSAs.

Figure 5.5 shows the circuit schematic and layout of the resistive-feedback MLSA. This MLSA consumes almost the same area as the active-feedback MLSA. Thus, decoupling capacitors were added in the empty area. Figure 5.6 illustrates the circuit schematic and layout of the active-feedback MLSA with body-bias. As explained earlier, DNW along with N-Wells were used to isolate the body of transistors N1 and N3 (Figure 5.6(b)). Owing to the large design rules of DNW layer in TSMC 0.18µm CMOS technology, the body-bias technique resulted in a fourfold increase in the MLSA area. In order to pitch-match this MLSA with the TCAM array, we placed four MLSA in the horizontal direction. The layout of the eight MLSAs of Block 4 is shown in Figure 5.7. These MLSAs were laid-out as two rows with each row containing four MLSAs as shown in Figure 5.7.

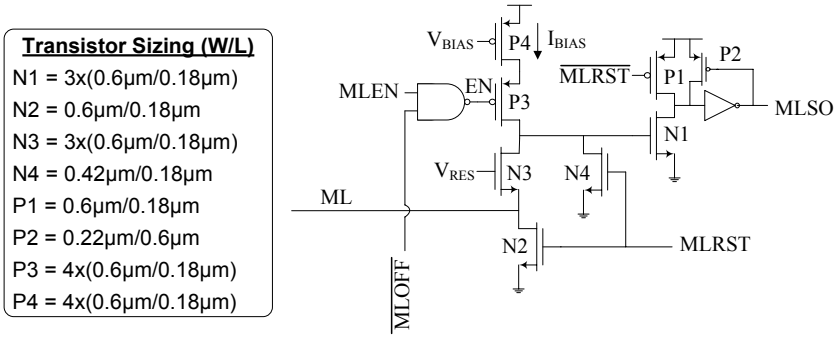


(a)

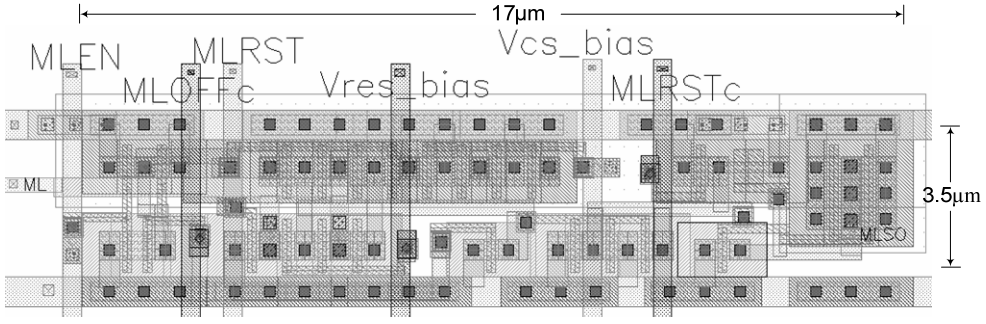


(b)

Figure 5.4: (a) Schematic and (b) layout of the active-feedback MLSA (employed in Block 2)

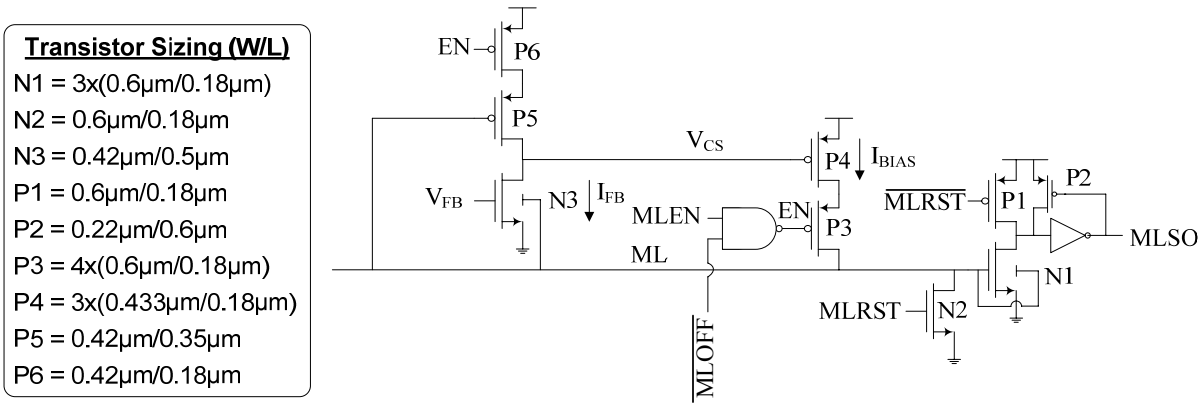


(a)

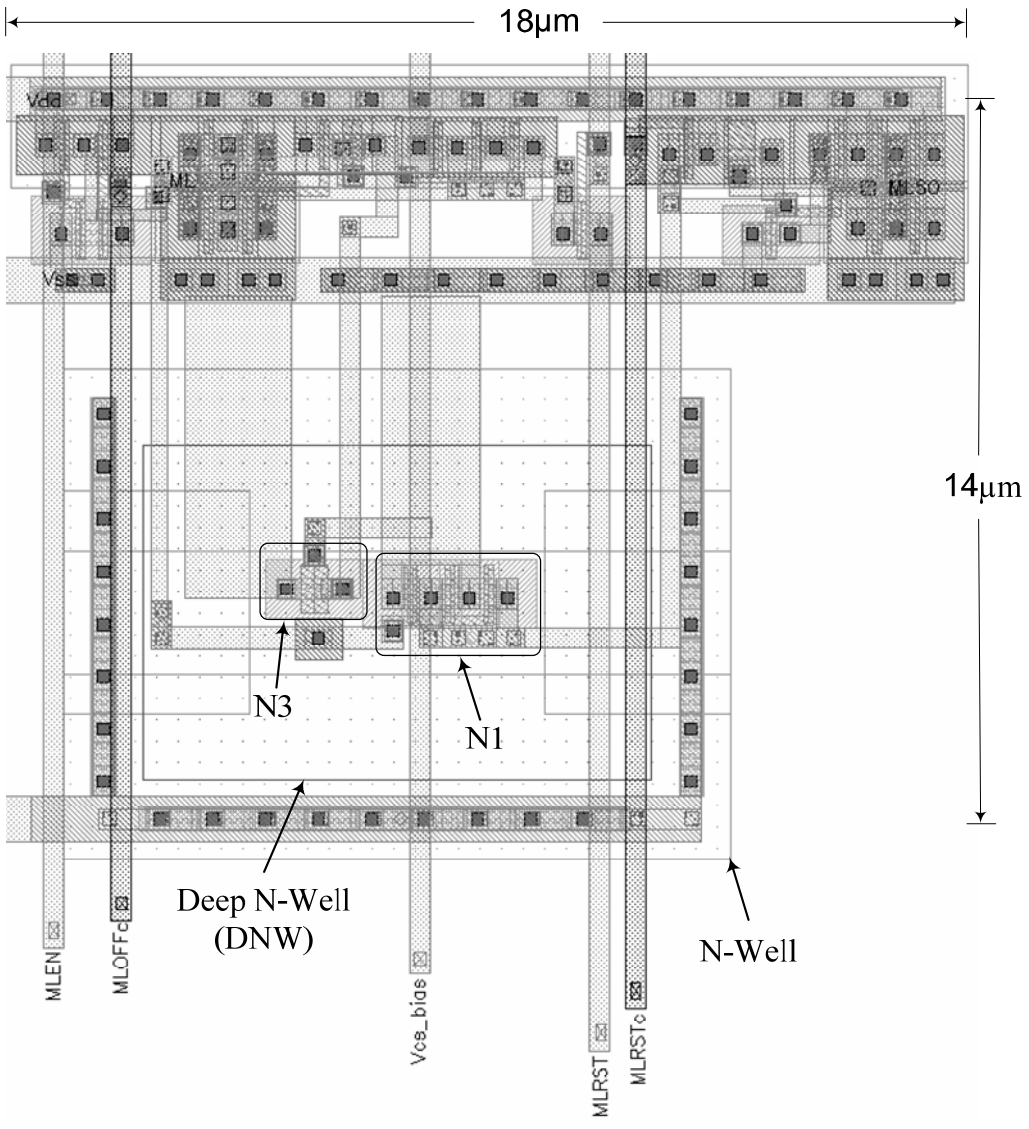


(b)

Figure 5.5: (a) Schematic and (b) layout of the resistive-feedback MLSA (employed in Block 3)



(a)



(b)

Figure 5.6: (a) Schematic and (b) layout of the active-feedback MLSA with body-bias (employed in Block 4)

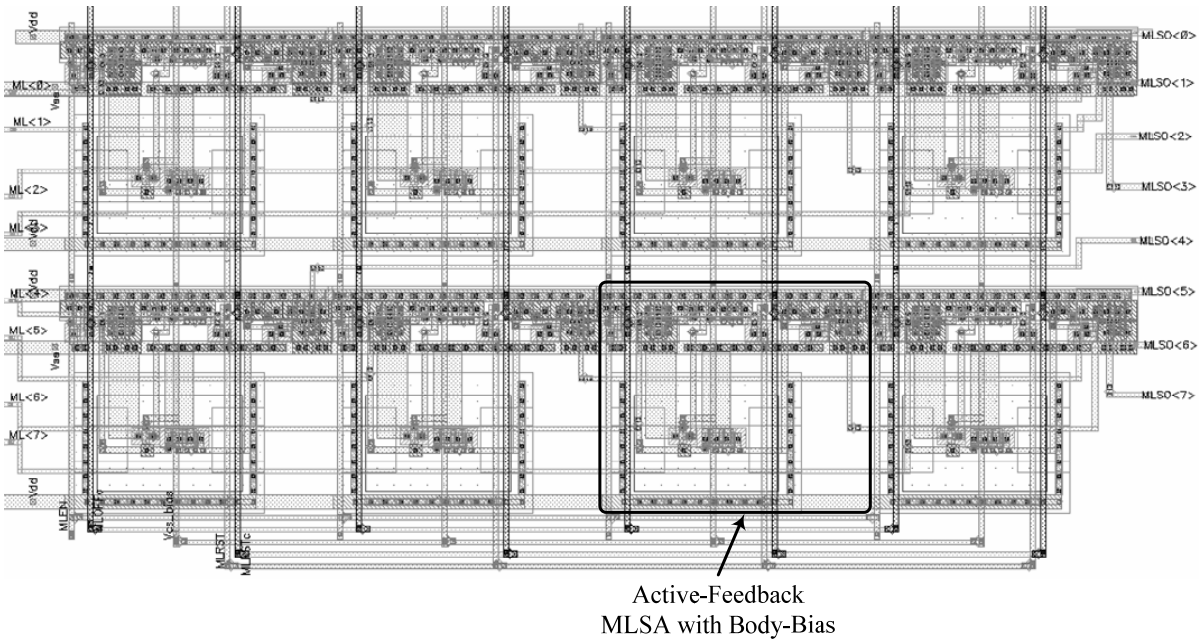


Figure 5.7: Layout of eight body-bias active-feedback MLSAs pitch-matched with the TCAM array (employed in Block 4)

5.1.3. Data Demultiplexers, Registers and Drivers

In order to store and search 144-bit words, the TCAM chip requires a 144-bit wide data bus. Considering the package and die area constraints, we dedicated nine pins for the off-chip data (D1-D9). Internally, the 144-bit data bus was constructed using a demultiplexer as shown in Figure 5.1. The 144-bit data bus was divided into sixteen 9-bit columns. The 9-bit off-chip data bus was connected to a column depending on the control signals (4-bit ColSel signals). An optional ColSel_all signal was provided to write the same 9-bit data to all the columns. Since there are four types of data (BL data, ML mask, SL data, SL mask), we included four sets of sixteen 9-bit registers (four rows of sixteen 9-bit columns in Figure 5.1). All 9-bit registers (4x16) share the same 9-bit off-chip data bus. However, their clock signals are gated with the row and column decoder outputs (*Row1* - *Row4* and *Column Dec1* - *Column Dec16*). Once a 9-bit register is selected by the row and column decoders, the 9-bit data is stored by the register at the positive edge of the W_CLK signal. SelGbl signals were generated by applying the OR-logic on SLs (as described in Chapter 3). Finally, buffers (line drivers) were inserted to drive the highly-capacitive data-lines (BLs, SLs, and SelGbls). The buffer layout was pitch-matched with the TCAM array to simplify the top-level integration.

5.1.4. Address Decoders and WL Drivers

The 144 words in the TCAM chip can be uniquely located using an 8-bit address bus (A8-A1). We divided the address-space into 4 blocks. Each block can be selected using one of the *Block_EN* signals as shown in Figure 5.1. These Block_EN signals were generated using the two MSBs (A8-A7) of the address bits. The remaining six bits (A6-A1) were used to select a word within a block. Decoders 1 and 2 use all six bits (A6-A1) to generate 64 unique addresses. Since decoders 3 and 4 require only eight addresses, they use only the three LSBs (A3-A1). The outputs of the address decoders were logically ANDed with an external WL_pulse signal to select a word for the WRITE operation. Alternatively, an on-chip pulse can also be generated. We used the external pulse for easier control of the pulse-width. The external pulse is acceptable for the TCAM test chip because speed of the WRITE operation is not critical in TCAM applications. The layout of a WL-driver was also pitch-matched with the TCAM word for easier integration.

5.1.5. Priority Encoders

In typical TCAM applications, the TCAM array is coupled with an off-chip SRAM, and each TCAM word is mapped into a corresponding SRAM word. Once the TCAM array is searched, a PE determines the highest priority match and encodes its location into binary format. The encoded address is used by the SRAM to retrieve the corresponding data. If there are on-chip SRAM blocks coupled to the TCAM arrays, the address encoding stage can be eliminated, and the address with the highest priority match can serve as an index to retrieve the search results [75]. However, modern TCAMs usually omit the on-chip SRAM because its absence offers a higher effective TCAM capacity, and many lookup applications may require a non 1-to-1 correspondence between TCAM and RAM [76]. Therefore, modern TCAMs employ PEs for resolving multiple matches and encoding the best match. As a result, a PE is usually designed in two stages: (i) multiple match resolver (MMR), and (ii) match address encoder (MAE). For the test chip, we employed individual PEs for each blocks and multiplexed their outputs using the *Block_EN* signals. Therefore, 64-bit PEs were used in Blocks 1 and 2, and 8-bit PEs were used in Blocks 3 and 4. The layout of the PE cell was also pitch-matched with the TCAM cell.

5.1.5.1. Multiple Match Resolver

An MMR is an *n*-bit input, *n*-bit output datapath circuit, where *n* is the number of words in the TCAM array. Assuming the active-high logic convention and the highest priority for the lowest physical address, an MMR can be described by the Boolean expressions in (5.1).

$$\begin{aligned} Out_0 &= In_0 \\ Out_1 &= In_1 \cdot \overline{In_0} \\ &\vdots \\ Out_n &= In_n \cdot \overline{In_{n-1}} \cdots \overline{In_1} \cdot \overline{In_0} \end{aligned} \tag{5.1}$$

As proposed in [77], a single-level folding scheme can reduce the worst-case delay (Out_n in equation (5.1)) by providing lookahead signals from the highest priority cell to the lowest priority cell, and the second highest to the second lowest, and so on [77]. This folding technique is further extended to multiple levels [77]. However, this scheme is not suitable for integration with a TCAM array due to too many interconnect routings and excessive clock loading (and resulting clock skew) if the circuit is laid out in a single column. Thus, a regular cell-based design using a pass-transistor chain is more suitable for a TCAM.

Figure 5.8 shows the circuit schematic of the 64-bit MMR used in our TCAM chip [70]. This circuit was proposed by another graduate student in our group working on "multiple match resolution and detection in TCAMs" [17]. In order to achieve fast and reliable operation, the MMR was designed in two levels. The first-level was divided into eight macro-blocks (Figure 5.8(a)). A wired-OR circuit was built into each macro-block to detect the presence of one or more matches at its inputs. The output of this wired-OR gate is a lookahead (LA) signal for interfacing with the second-level MMR, which identifies the block that contains the highest priority match. The resolved second-level signals act as block enable (BE) signals for the first-level MMRs. In order to layout both levels of MMRs into one column, the MMR cells in the second-level were distributed between the first-level blocks as shown in Figure 5.8(b).

The MMR circuit is based on the "match token" concept [16][17]. A brief description of this circuit is included here. A more detailed description can be found in [17]. As shown in Figure 5.8(c), the clock signal initially precharges the internal nodes of the pass transistor chain to a voltage close to ($V_{DD} - V_t$). Since the MMR inputs (In_0-In_N) are connected to MLSA outputs (MLSO in Figure 5.3(a)), all MMR inputs are reset to GND during the reset

phase of MLSAs. Hence, node C is charged to V_{DD} and node D to '0', which in turn switches off T5 and precharge node E to V_{DD}. If there are two matches in the TCAM array at words 1 and N, In_1 and In_N are raised to V_{DD}. The rest of the input bits remain at '0'. The LA signal is applied to the input of a delay element for generating the strobe (SS) signal. Since the first-level pass-transistor chain is not the critical path of a multi-level MMR, this delay reduces the LA node capacitance without slowing down the critical path. Switching SS from '1 → 0' allows the discharging of the pass-transistor chain down to the highest priority match in the local macro-block, which is analogous to a '0' ("match token") percolating down the pass-transistor chain. The internal nodes C and D of the highest priority cell are inverted, and T5 is turned 'ON'. Upon the arrival of the BE signal, node E is discharged to '0', which in turn switches the output R_1 to '1' to indicate that word 1 is the highest priority match.

This MMR achieves high-speed and low-power operation due to several circuit improvements over previous designs [16]. First of all, the BE is shielded from the output inverter capacitance by transistor T5 (except the local highest priority cell). Thus, the size of a macro-block is not limited by the BE capacitance. Second, node C is precharged using the corresponding MMR input. This eliminates the unnecessary clock power consumed in precharging these nodes. Third, the "match token" is generated only if there is at least one match in a macro-block. Finally, if the lowest priority cell receives the "match token", this cell must be the only match in the macro-block. Thus, the transistors T8 and T9 can be removed to reduce the worst-case RC delay.

5.1.5.2. Match Address Encoder

In the presence of multiple matches, the MMR always favours the highest priority match (lowest physical address). We designed the MAE to take advantage of this property. Figure 5.9 illustrates this idea using a dynamic CMOS encoder. In this circuit, the worst-case power consumption corresponds to the least likely condition (R_N = '1'). In our design, we chose a pseudo-NMOS based MAE (clock signal in Figure 5.9 is connected to GND) because it consumes less power for high-speed TCAM operations due to the absence of the clock power. Since the MAE is designed to consume less power for lower physical (higher priority) addresses, the impact of static power (due to the pseudo-NMOS design) is also minimized by the MMR action.

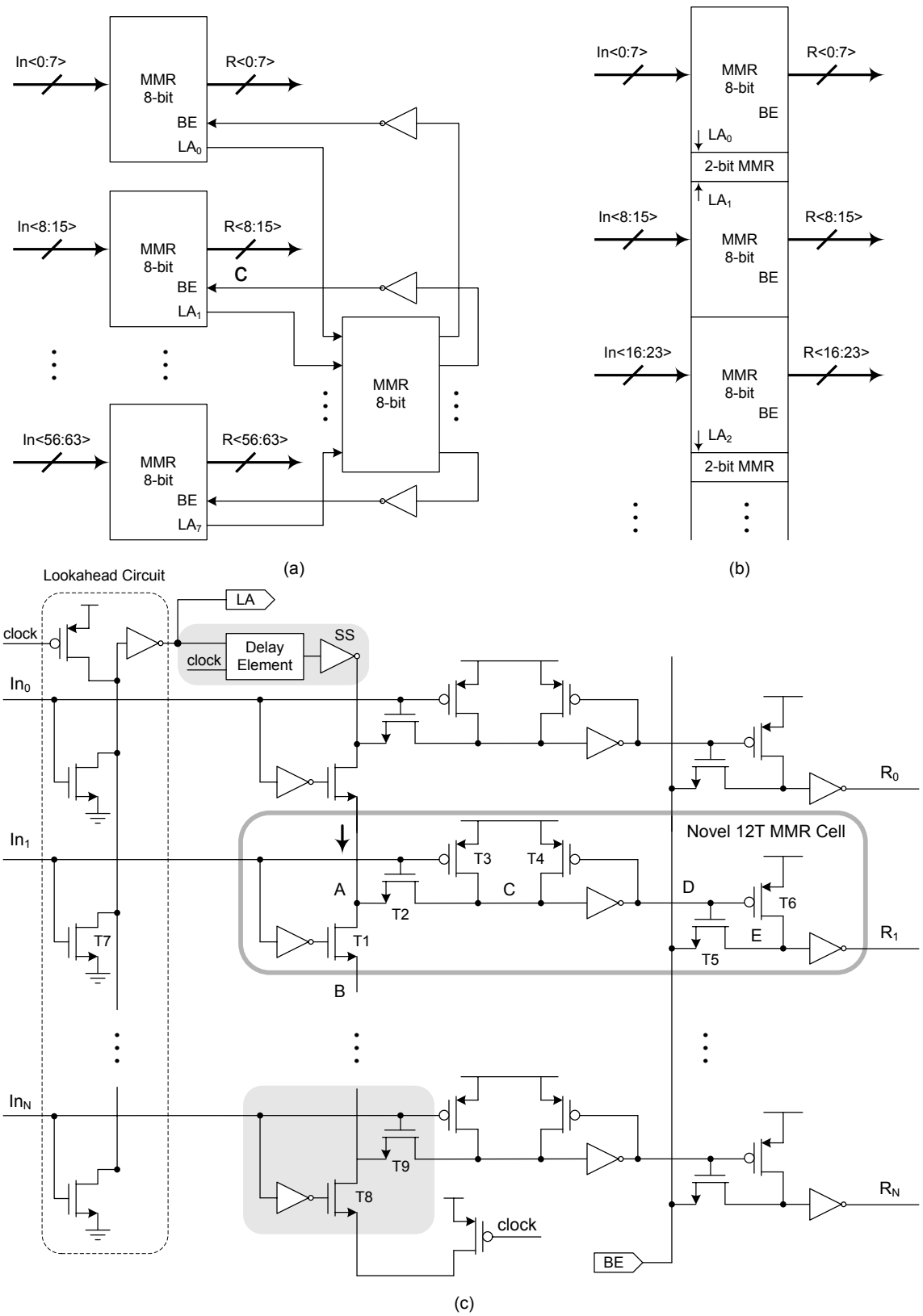


Figure 5.8: (a) Block diagram, (b) floorplan, and (c) circuit schematic of the 64-bit MMR [17]

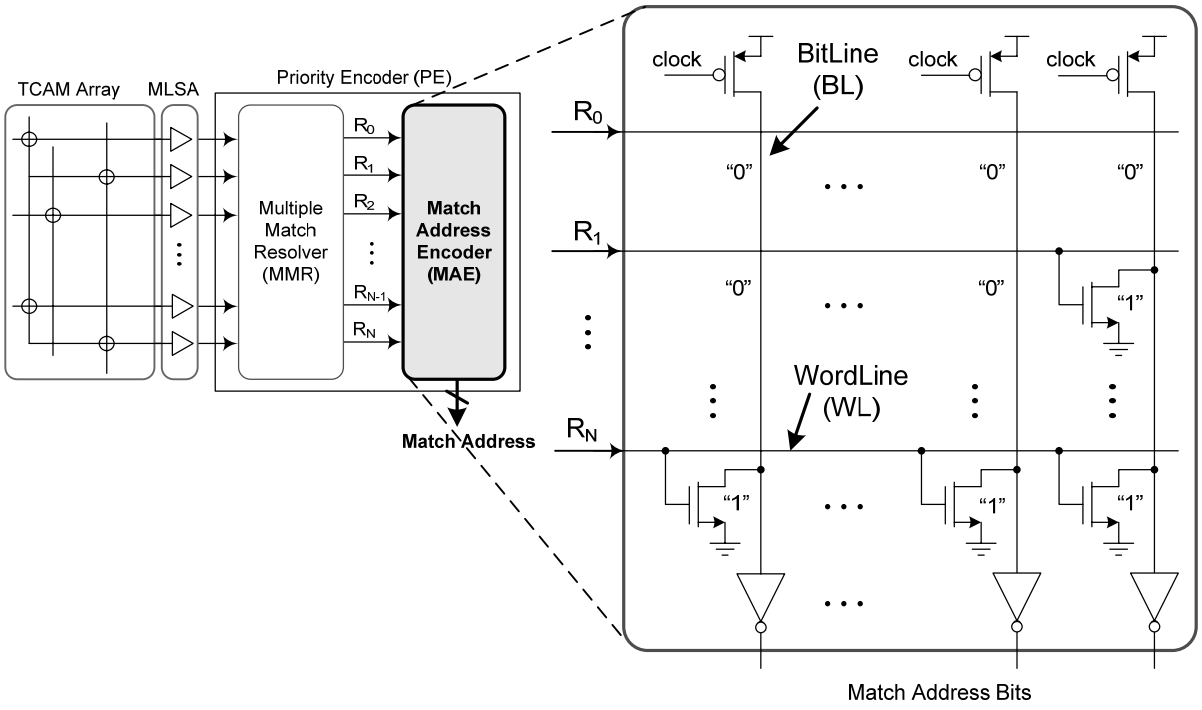


Figure 5.9: CMOS dynamic MAE favouring the highest priority address [17]

5.1.6. Bias Circuitry and I/O Design

As described in subsection 5.1.2, each MLSA has a current source that requires a bias voltage. These bias voltages can be either supplied off-chip or generated on-chip by mirroring the external currents. Since the ML sensing speed varies almost linearly with the bias current, it is more convenient to control the bias current (rather than the bias voltage) for the chip measurements. We generated the bias currents using on-chip current mirrors and bias circuitry as shown in Figure 5.10. Blocks 1 and 3 use PMOS-based current sources for charging MLs (I_{BIAS} in Figures 5.3 and 5.5). Simulation results show that the minimum bias current for the 144-bit TCAM word (0.18µm 1.8V CMOS technology) within the region of interest always remains greater than 20µA. Thus, a variable resistor of 100KΩ can be connected between the bias pin (bonding pad in Figure 5.10(a)) and external GND. This setup can generate bias currents above 20µA by varying R1. A similar setup can also be used for the bias circuitry of Block 3 as shown in Figure 5.10(c). On the other hand, Blocks 2 and 4 require NMOS-based current sinks, and a suitable bias-circuitry is shown in Figure 5.10(b). Since much smaller currents (almost one-tenths of I_{BIAS}) are needed for the feedback circuit (I_{FB}), a relatively larger resistor (R2 = 1MΩ) can be used in this case.

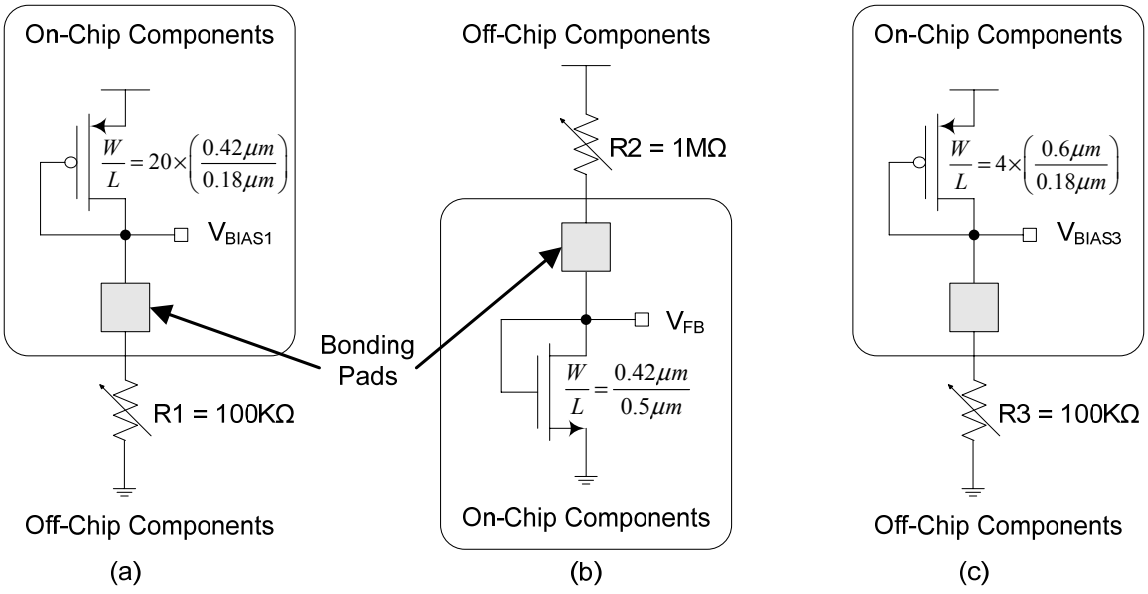


Figure 5.10: Bias circuitry for (a) Block 1, (b) Blocks 2 and 4, and (c) Block 3

In order to reduce *L(di/dt)* voltage drop (due to bond-wire inductance) when the test chip is drawing a large current from the power supplies, we dedicated 11 pins for power supply connections, and also included a large on-chip decoupling capacitance (approximately 1nF). The power supply pins were placed uniformly for even current-distribution across the chip. The decoupling capacitance was designed using thick-oxide (3.3V) native NMOS transistors because they exhibit less leakage, higher reliability and smaller series resistance (faster transient response) than the standard NMOS transistors. The MOS capacitance is a popular choice for on-chip decoupling because the thin oxide layer in the MOS structure results in much higher capacitance per unit area than that of the metal-insulator-metal (MiM) capacitance. Still, the MiM capacitance can complement the MOS capacitance wherever possible because a MiM capacitor can handle much faster but smaller current transients (smaller capacitance per unit area). On the other hand, a MOS capacitor can work well for relatively slower but larger current transients. In CMOS 0.18µm technology, a MiM capacitor requires metals M5 and M6, and a MOS capacitor can be laid-out using M1 and M2. Thus, MiM capacitors can be laid-out on top of the MOS capacitors. In our test chip, we placed the MOS decoupling capacitors on both sides of the MLSAs, under the power supply rings, chip corners, and all other empty spaces. MiM capacitors were placed in chip corners over the MOS capacitors. The power supply rings (V_{DD} and V_{SS}) rings included all the metals

except M1 and M2, which were utilized by the MOS decoupling capacitors. The placement of MOS and MiM decoupling capacitors in the test chip is shown in Figure 5.11.

The test chip has four output pins ($\overline{MLOFF}$, Match/MM, Scan_OUT, and AOUT) shown by arrow signs in Figure 5.1. We included large on-chip buffers for an estimated load of 4pF at these output pins. The output load was estimated as follows:

Load capacitance due to the bonding pad and package pin ≈ 1pF

Load capacitance due to the PCB traces ≈ 1pF

Load capacitance due to the active probe ≈ 1pF

Miscellaneous ≈ 1pF

Owing to the large transistors and load capacitance of these buffers, they are very power hungry. They may inject large current transients in the power supplies causing a chip to malfunction. Thus, the number of output pins was minimized by adding a scan chain at the MAE output (AOUT) and by multiplexing the scan chain outputs (Scan_OUT). Large on-chip decoupling capacitance also helps in stabilizing the power supplies.

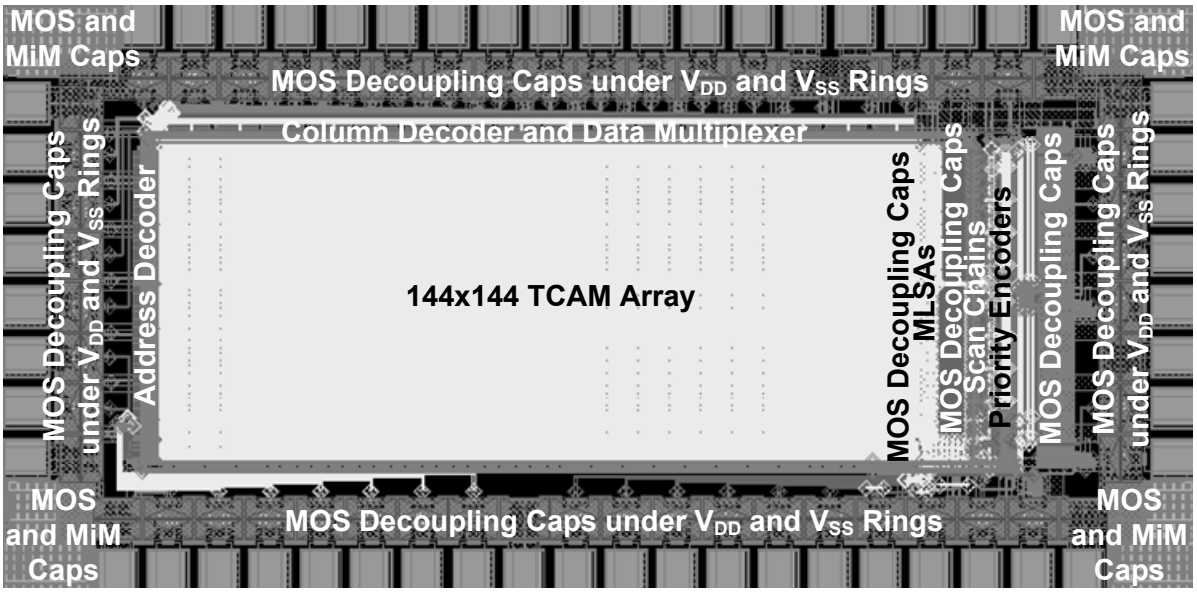


Figure 5.11: Placement of the MOS and MiM decoupling capacitors on the test chip

5.1.7. Chip-Level Simulation

The test chip contained 400,000 transistors, and simulating such a large chip in SPICE is a very time consuming process. We performed SPICE simulations only on a single TCAM

word and its dummy. For complete chip simulation, we used the *Synopsys NanoSim* circuit simulator that can simulate large chips at the expense of accuracy [78]. *Nanosim* provides an option of different accuracy settings for different hierarchical blocks. For example, the address decoders, data multiplexers, TCAM array and scan chains can be set to low-accuracy simulation, and MLSAs and PE can be set to high-accuracy simulation. This combination simultaneously achieves both better accuracy and small simulation time. We extracted the capacitances of different lines (BLs, SLs, MLs, etc) from the chip layout and performed circuit simulations after including them in the schematics. Table 5.1 shows the extracted capacitances of each line. The ML capacitance includes only 72 NMOS drains (instead of 144) because the ML transistors of two adjacent cells share the same drain contact, as shown in Figure 5.2.

Table 5.1: Extracted capacitances of different lines in the test chip

Lines	***Load Transistors (W (µm)/L(µm))***	***Line Capacitance (fF)***
SL1	144 NMOS Gate (0.42/0.18)	123
SL2	144 NMOS Gate (0.42/0.18)	144
ML	72 NMOS Drain (0.42/0.18)	64
BL1	144 NMOS Drain (0.42/0.18)	120
BL1c	144 NMOS Drain (0.42/0.18)	120
BL2	144 NMOS Drain (0.42/0.18)	126
BL2c	144 NMOS Drain (0.42/0.18)	132
SelGbl	144 NMOS Gate (0.42/0.18)	198
MLEN	144 NMOS Gate (1.02/0.18)	111
MLOFFc	144 NMOS Gate (1.02/0.18)	108
MLRST	144 NMOS Gate (0.6/0.18)	120
MLRSTc	144 PMOS Gate (0.6/0.18)	132

5.2. Chip Testing

In order to successfully test the TCAM chip, we included several on-chip structures to simplify the test process. Special attention was paid to the PCB design including the placement of components, V_{DD} and GND planes, off-chip decoupling capacitances, etc. Test

equipment was identified that can generate a large number of complex input patterns. We also devised a simple method to measure the power consumption of individual components and the whole chip. These details are described in the following subsections.

5.2.1. Design for Testability

The chip included several design for testability structures to increase the observability and controllability of individual components. These structures ensure the successful testing of a component even if other components are not functional. For example, the scan chains between the MLSAs and PEs can be used either to insert input vectors for PEs or to latch and scan-out the MLSA outputs. In normal operation, MLSA outputs are directly connected to the PE inputs through multiplexers. Typically, the ML sensing time is defined as the time-difference between MLSA_EN and $\overline{MLOFF}$. However, measuring the output pin $\overline{MLOFF}$ using an oscilloscope does not give accurate results because the propagation delay of the output buffer is also added to the actual value. Thus, we used other indirect methods to measure the ML sensing time more accurately as described in the following paragraph.

The first method uses the Scan_CLK to determine the time when valid data is available on MLSA outputs. Since ML sensing starts at the positive edge of MLSA_EN and the scan chain latches MLSA output at the positive edge of Scan_CLK, the delay between MLSA_EN and Scan_CLK can be used to determine the ML sensing time. For example, if MLSA_EN is delayed until the time-difference between MLSA_EN and Scan_CLK is just enough to latch the correct data in the scan chain, the difference between the two signals is close to the ML sensing time. This method introduces only a small inaccuracy due to the flip-flop setup time in the scan chain. The second method uses a multiplexer to replicate DMLSO with an external signal MLSA_OFF. Initially, the multiplexer connects DMLSO and $\overline{MLOFF}$ through the inverter-chain. The output pin $\overline{MLOFF}$ is observed and its negative edge is marked on the oscilloscope. Then, the multiplexer connects MLSA_OFF to $\overline{MLOFF}$ through the inverter-chain. If the input signal MLSA_OFF is varied such that the negative edge of $\overline{MLOFF}$ matches the previously marked location, the difference between MLSA_EN and MLSA_OFF will be equal to the ML sensing time. These design for testability features simplify the test process and improve the measurement accuracy.

5.2.2. PCB Design

We designed a two-layer PCB to perform various measurements on the test chip. The top-layer was used for the GND plane and the bottom-layer was used for the V_{DD} plane. The signals were shielded by V_{DD} or GND wherever possible as shown in Figure 5.12. Two voltage regulators were used to generate V_{DD} and V_{RES} (for resistive-feedback MLSAs). The test chip contains 55 bonding pads including 36 voltage inputs, 4 current inputs, 4 outputs and 11 power supplies. The chip was packaged in 80-pin surface-mount ceramic quad flat package (CQFP) with the pin-out shown in Figure 5.13. The voltage inputs and outputs of the test chip were accessed using 2-pin connector headers. One pin of each header was connected to GND for reference (Figure 5.12). Each current input was connected to V_{DD} or GND through a variable resistor, and the current was controlled by varying the resistance, as shown in Figure 5.10. The output pins were monitored using active-probes and oscilloscope. The inductance of PCB traces may cause $L\frac{di}{dt}$ voltage drop at the power-supply pins. Thus, decoupling capacitors were connected between V_{DD} and GND to stabilize the power-supply pins. The effect of PCB inductance was reduced by placing decoupling capacitors close to the chip.

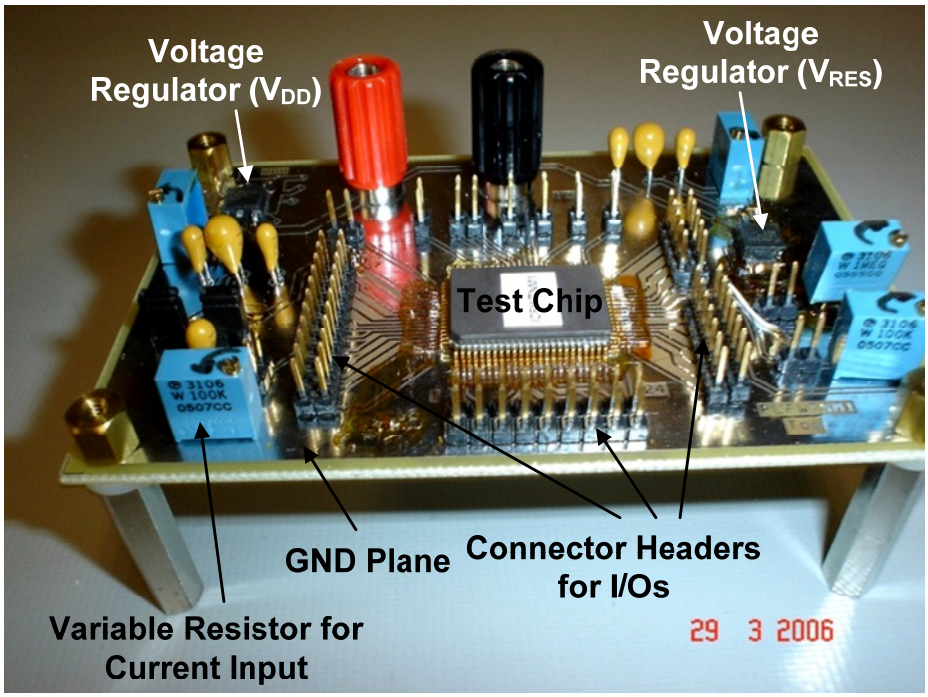


Figure 5.12: Two-layer PCB for the test chip measurements

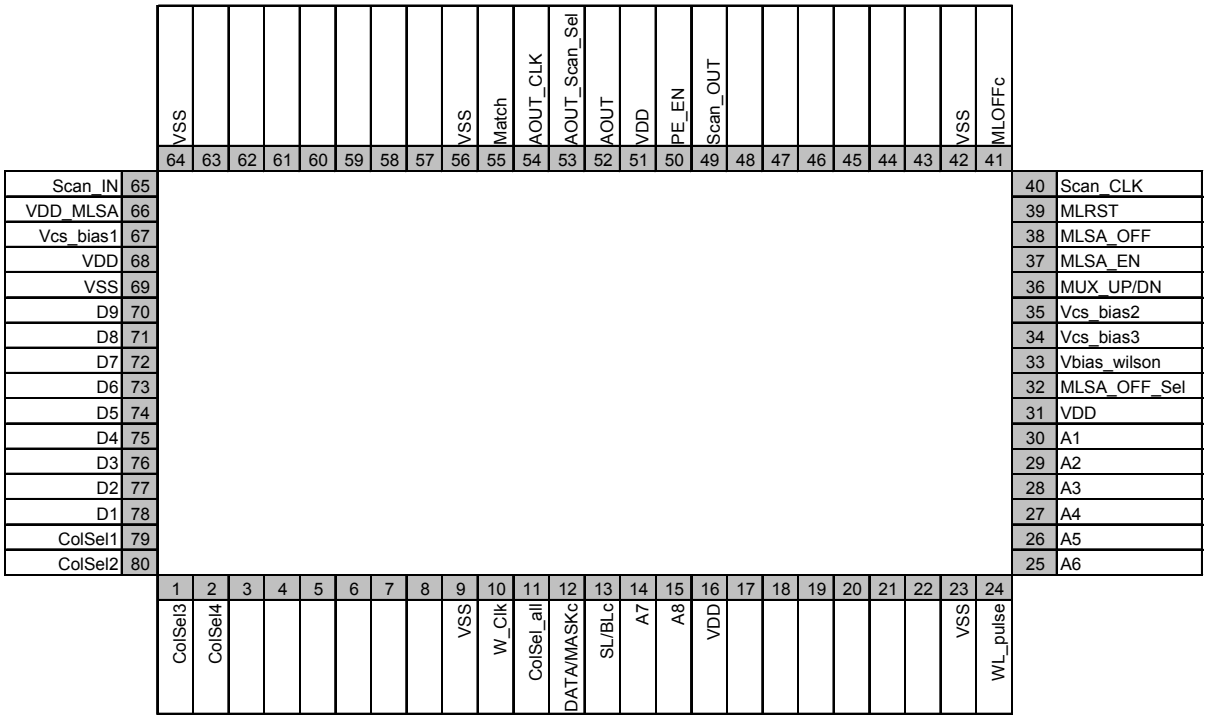


Figure 5.13: Pin-out of the test chip packaged in a 80-pin CQFP

5.2.3. Test Pattern Generation

The input test pattern was generated using Tek DG2020A data generator with 36-channels. The measurements were done in four steps. First, the same data pattern was written to all memory locations and the search key register. Second, the SEARCH operation was performed in all blocks, MLSA outputs were latched using Scan_CLK, and Scan_OUT was observed to ensure match in all words. Third, one bit of the search key was inverted, and the above operations were repeated to ensure that all words fail to match the search key. In this step, power consumption was also measured using the method described in the next subsection. Then, the third step was repeated for increasing number of mismatches. Finally, the PE was tested by creating match condition at different addresses, latching PE outputs with AOUT_CLK, and monitoring the Match/MM and AOUT pins. The MLSA delay was measured by replicating DMLSO with MLSA_OFF as explained in subsection 5.2.1. Similarly, the total delay of SL drivers and MLSAs was calculated by measuring the difference between W_CLK and DMLSO. The PE delay was calculated by measuring the difference between DMLSO and AOUT_CLK.

5.2.4. Power Measurements

The power consumption of a chip can be calculated by measuring the average current through the power supply pin. The commercially available voltage generators also display the current through the power supply but this current usually has accuracy only in the milliampere (mA) range. In addition, we used voltage regulators in our test setup to stabilize the power supply voltage of the chip. Thus, we performed the current measurements using an ammeter with accuracy in one-tenth of a microampere (μA) range. We applied periodically varying inputs such that the power supply draws the same current in each cycle. However, it was difficult for the ammeter to respond to high-frequency transient currents drawn by the power supply. In order to solve this problem, we inserted low-pass filters at V_{DD} and V_{DD_MLSA} to remove those transients and convert them to low-frequency average currents as shown in Figure 5.14. The series resistance of 10Ω was added to create an RC low-pass filter. Larger values of R may result in performance degradation due to IR voltage drop. In normal operation, switches S1 and S2 remain closed (short-circuited).

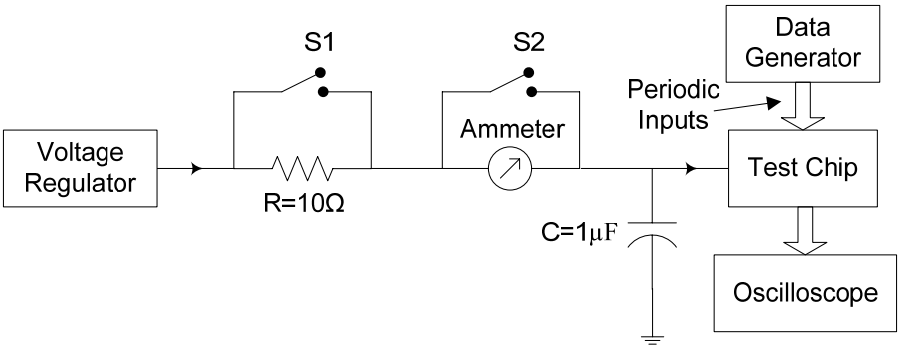


Figure 5.14: Test setup for power measurements

As long as the average current remains less than 500μA, the ammeter series resistance remains negligible, and no degradation in the chip performance was observed. We also observed that ammeter series resistance is usually sufficient to result in stable measurements, and the resistance R can be eliminated by closing the switch S1. After measuring the current, the energy consumption was calculated by multiplying this average current by the time-period of the input patterns and V_{DD}. The energy consumption of MLSAs was measured using the current through the V_{DD_MLSA} pin. The energy consumption of the remaining blocks was calculated by subtracting the current through V_{DD_MLSA} from the current through V_{DD}.

The energy consumption of an individual block was measured by enabling only one block at a time and repeating the above procedure. For example, the energy consumed by the SL drivers was measured by disabling MLSAs and PEs. Similarly, the PE energy consumption was measured by subtracting the MLSA energy from the total energy when the same search key is used repeatedly. As described earlier, the MAE was designed using the pseudo-NMOS logic. Thus, its power consumption is dominated by the static current from V_{DD}. It is difficult to measure this current when the MMR dynamic power is comparable. However, the static current becomes dominant at lower frequencies (such as 1KHz) and the measured current approaches the static current consumed by the MAE. It was observed that further reductions in frequency did not reduce the current through V_{DD}. The MMR energy consumption was calculated by subtracting the MAE energy from the total PE energy. In order to calculate the power consumption at a given SEARCH rate, the measured energy was divided by the time-period of the SEARCH operations.

5.3. Measurement Results

The measurement results of the MLSA section were already discussed in Chapter 4. It has been shown that the proposed active-feedback MLSA consumes 50-56% less energy than the conventional CR-MLSA. A micrograph of the 1mm x 2mm TCAM chip in 0.18µm CMOS technology has also been shown in Chapter 4. This section highlights the energy and performance of the other key components on the test chip (such as SL drivers and PEs). In addition, it shows the energy and performance of the complete chip when these individual components are integrated together. These results are highlighted in Table 5.2. ML sensing time and energy are depicted for the active-feedback MLSA. The total search path latency is 4.3ns. Thus, the SEARCH operation can be performed at 200MHz without pipelining. If SL-drivers, MLSAs, and PEs are pipelined, the SEARCH operation can be performed at 500MHz. Since the MAE consumes static current, its energy consumption per cycle can be minimized by increasing the SEARCH rate. Commercially available TCAMs are already running at 250 Msps. It is expected that the next generation of TCAMs will run at 500Msps or higher. Hence, we estimated the MAE energy assuming that it is enabled only for 2ns per cycle. Also MAE energy is negligible if there is a match at address 1. On the other hand, the MAE energy is significantly higher when the match is located at address 64. Since the MMR

action favours the lower physical addresses, the MAE is likely to have the best case or near best case energy almost always as described in subsection 5.1.5.2. MMR energy also varies with the match locations as shown in Table 5.2.

Table 5.2: Measurement result of the TCAM chip (active-feedback MLSA and 64-bit PE)

S. No.	***Feature***	***Result***
1.	Process technology	0.18μm CMOS
2.	Power supply voltage	1.8V
3.	ML sensing time (T_{MLSA})	1.7ns
4.	PE propagation delay (T_{PE})	1.5ns
5.	SL driver propagation delay (T_{SL})	1.1ns
6.	Total search path latency (T_{Total})	4.3ns
7.	ML sensing energy (E_{ML})	1.8fJ/bit/search
8(a).	MMR energy per cycle (Match at address 1)	1.15pJ
8(b).	MMR energy per cycle (Match at address 64)	1.18pJ
8(c).	MMR energy per cycle (Match at addresses 1 and 64)	1.61pJ
9(a).	MAE energy per cycle @500MHz (Match at address 1)	8fJ
9(b).	MAE energy per cycle @500MHz (Match at address 64)	2.23pJ
10(a).	PE energy per cycle (Match at address 1)	1.16pJ
10(b).	PE energy per cycle (Match at address 64)	3.41pJ
10(c).	PE energy per cycle (Matches at addresses 1 and 64)	3.84pJ
11.	Average energy consumption of SL drivers	3.22fJ/bit/search
12(a).	Total energy of a SEARCH operation in a 64x144 TCAM block (Best case)	47.42pJ
12(b).	Total energy of a SEARCH operation in a 64x144 TCAM block (Worst case)	50.1pJ

Table 5.3 compares the proposed PE with other recently published designs. Although the other two schemes were fabricated in a different process technology, the proposed PE outperforms them quite significantly. In addition, the cell-based regular structure of the proposed PE is more suitable for integration with the TCAM array. The other two designs

were fabricated in isolation (without TCAM arrays), and their large number of complicated interconnections make them impractical for integration with the TCAM array [77][79].

Table 5.3: Proposed 64-bit PE along with the other recent designs

Feature	***Proposed PE***	***Wang [79]***	***Huang [77]***
Size	64-bit	32-bit	256-bit
Process technology	0.18µm	0.6µm	0.6µm
V_{DD} (V)	1.8	3.0	3.0
Maximum speed	500MHz	100MHz	116MHz
Energy per cycle	1.16pJ-3.41pJ	90pJ	274pJ
TCAM compatibility	Yes	No	No

The average energy consumption of the SL drivers was measured by switching all the SLs in every SEARCH cycle and then dividing the measured value by two. It can be noticed that the SL energy consumption is higher than the ML sensing energy. Thus, more techniques (such as hierarchical SLs) are needed to reduce this energy. The total energy consumption of a 64x144 TCAM block can be estimated by adding the energies of different components. The worst case search energy of this block (9Kb) was estimated to be 50pJ. In order to compare these results with those of the existing designs, we surveyed the published literature for comparable size TCAMs in 0.18µm CMOS technology. Table 5.4 compares the overall measurement results of our TCAM with two other recent designs.

Table 5.4: Our design along with the other recent designs

Feature	***Our Design***	***Akhbarizadeh [80]***	***Pagiamtzis [37]***
Memory Size	9Kb (64x144)	2Kb (64x32)	36Kb (256x144)
Process technology (CMOS)	0.18µm	0.18µm	0.18µm
V_{DD}	1.8V	1.8V	1.8V
Speed	200MHz	229MHz	142MHz
Power	10mW	6.2mW	94mW
Norm. Power (1Kb@200MHz)	1.11mW	2.71mW	3.66mW
Power Performance	5.42fJ/b/search	13.20fJ/b/search	17.95fJ/b/search

Table 5.4 confirms that the circuit and chip-level design techniques proposed in this work show noticeable improvement in terms of a smaller power performance than the existing designs. Note that the results of the 36Kb TCAM implemented by Pagiamtzis et al. do not include the power and delay contribution of the priority encoders, which are included in our results.

In order to compare our design with the commercially available TCAM-based NSEs, we surveyed various NSEs from Cypress, IDT, NetLogic, Sibercore, and MOSAID. We found only two NSEs with publicly available datasheets: CYNSE70064A from Cypress and SCT2000C from Sibercore [81][82]. Table 5.5 compares these two NSEs with the extrapolated results of our design. Typically, TCAM energy consumption linearly increases with the memory size due to the corresponding increase in the ML/SL capacitances. Similarly, TCAM power consumption also increases linearly with the frequency of operation. Therefore, we can compare the power consumption of the three designs by normalizing their memory size and frequency of operation. For example, CYNSE70064A running at 83MHz consumes 5.4W. If we increase the operating frequency to 100MHz, the power consumption will also increase correspondingly (6.51W at 100MHz). Similarly, our design consumes 10mW at 200MHz. Hence, it will consume 5mW at 100MHz. If the memory size is increased by a factor of 222 (9 Kb → 2 Mb), the power consumption will also increase by the same factor (1.11W). It should be noticed here that our design is significantly smaller than the two

Table 5.5: Our design along with commercially available TCAM-based network search engines

Features	***Our Design***	***Cypress CYNSE70064A [81]***	***Sibercore SCT2000C [82]***
Memory Size	9 Kb	2 Mb	2 Mb
Speed	200 MHz	83 MHz	100 MHz
Process Technology	0.18 µm	0.18 µm	0.18 µm
Core Supply Voltage (V_{DD})	1.8 V	1.8 V	1.8 V
Supply Current during Search (I_{VDD})	-	3 A	2.7 A
Power Consumption	10 mW	5.4 W	4.86 W
Normalized Power Consumption (Watts/2Mb/100MHz)	1.11 W	6.51 W	4.86 W

commercial NSEs. Thus, a direct comparison with the linearly extrapolated values may result in a large error due to the following reasons:

- Commercial designs have several other logic blocks to support features beyond table lookup and priority encoding. These features include flexible word size, glueless interface to industry-standard SRAMs, IEEE 1149.1 test access port, option for cascading several NSEs, automatic learning, etc. The power consumption of these blocks is not included in the extrapolated results.
- Commercial designs include several pipelined stages, built in self test (BIST) circuitry, and redundant rows and columns for yield improvement. These features result in area and power overhead, which is not included in the extrapolated results.
- Large commercial TCAM chips with the above sophisticated features have a complex routing of power lines and interconnects, which results in a significant increase in the parasitic capacitance and associated power consumption.
- High power consumption in large chips also increases the junction temperature, which results in higher leakage and slower performance. Slow rise and fall times may also cause increased short-circuit current in CMOS circuits.

Therefore, we should account for the above sources of power consumption before comparing our design with the commercial TCAM chips. Assuming that the extra features supported by the commercial TCAM chips increase the power consumption by 100%, our design (2.22W) still consumes 54% less power than the commercial designs.

5.4. Conclusions

In this chapter, we presented the integration aspects of the TCAM chip. Several layout-level techniques were discussed that can help to reduce the capacitance of the performance-critical lines such as the SLs and MLs. In order to achieve a compact layout, peripheral components (such as MLSAs, PEs, address decoders and WL drivers, data multiplexer and line drivers, etc.) were pitch-matched with the TCAM array. The PE was implemented as a combination of two blocks: MMR and MAE. The 64-bit MMR was designed in two 8-bit stages for faster and more reliable operation. A cell-based MMR was designed for easy integration with the TCAM array. The MAE was designed such that the MMR action results in the MAE input settings that correspond to lower energy.

We also presented methods and techniques to simplify the test process. Several design for testability features were included on the chip including scan chains, multiplexers with external signals replicating the internal signals, etc. We described techniques used in PCB design for stable and accurate measurements. A simple technique for power measurement was also described. Finally, we presented the measurement results of individual components and estimated the chip level energy and performance. The PE measurement results show significant reductions in energy and delay over the recently published designs. We also calculated the power consumption of the 9Kb TCAM block, and compared it with the existing comparable designs in the same process technology. In order to compare our design with the publicly available data from two commercial TCAM chips (2Mb), we extrapolated our results appropriately (9 Kb → 2 Mb). After taking other sources of power consumption for the commercial TCAMs into account, we found that our design consumes 54% less power than the commercial designs. The overall results reinforce the power reduction capability of the chip design techniques proposed by this work.

Chapter 6

Conclusions

TCAMs are gaining importance in high-speed lookup-intensive applications. However, the high power consumption of TCAMs is limiting their popularity and versatility. This work proposed several circuit techniques to reduce TCAM power consumption, which is typically dominated by the frequent switching of MLs and SLs. The main contribution of this work has been in reducing the ML energy using: (i) positive-feedback MLSAs, (ii) low-capacitance comparison logic, and (iii) low-power ML segmentation techniques (such as dual ML and charge-shared ML). In addition, this work proposed techniques to reduce the static power consumption, which is becoming a concern in TCAMs employing low-power architectures.

A significant portion of the TCAM power is consumed by MLSAs for match detection. Thus, we developed three MLSAs that apply positive feedback for power reduction in ML sensing. Instead of providing the same current to all MLs, these MLSAs modulate the ML current source such that a larger current flows into the ML_0 (match) and a smaller current flows into the ML_k (mismatch). This combination maximizes the speed and minimizes the energy of ML sensing. The above statement implies that the proposed MLSAs

do not necessarily reduce the power consumption (Power = Energy x Frequency) because the increase in frequency offsets the reduction in energy. However, power reduction can be achieved if the frequency is not increased. Measurement results of the active-feedback MLSA (in 0.18μm CMOS technology) show 50-56% reduction in energy over the conventional CR-MLSA. In addition, the positive feedback action improves the robustness of ML sensing by feeding less current to ML_1 and more current to ML_0. The active-feedback action can be further improved by applying the body-bias technique described in Chapter 4. The body-bias technique shows impressive results in simulations (53-59% reduction in energy) but the measurement results are not so attractive (23-48% reduction in energy). This discrepancy can be explained by the additional capacitive load on ML due to the body-Nwell capacitance, which was not accurately modeled in simulations. This observation also reinforces the importance of the chip measurement results. However, the improvement in simulations indicate that manipulating the body-bias can be effective in other process technologies (such as SOI), where the body-capacitance and area penalty are much smaller.

In order to further reduce the ML sensing energy, we proposed comparison logic that makes the MLs less capacitive by reducing the number of transistors connected to an ML. The chip measurement results of the proposed comparison logic show 25% and 42% reductions in ML sensing energy and time, respectively, which can further be improved by careful layout. We also proposed two low-power ML segmentation techniques: dual ML and charge-shared ML. The dual ML technique divides a ML at the bit level, and two sides of the comparison logic are connected to the two ML segments. Thus, the power savings become independent of the data statistics and segment ordering. Simulation results show that the dual ML TCAM can achieve 43% power savings for a small (4%) trade-off in the search speed. The charge-shared ML scheme achieves power savings by partial recycling of the charge stored in the first ML segment. Chip measurement results show that the charge-shared scheme results in 11% and 9% reductions in ML sensing time and energy, respectively, which can be improved to 19-25% by using a digitally controlled charge sharing time-window and a slightly modified MLSA. It was observed that the low-power architectures are indeed reducing the TCAM switching activity significantly, and that now the static power is becoming a serious concern. We proposed two techniques to reduce the static power in TCAMs: dual-V_{DD} and low-leakage TCAM cells. A dual-V_{DD} technique trades-off excess

noise margin in the MLSA for smaller leakage by applying a smaller V_{DD} to TCAM cells and a larger V_{DD} to the peripheral circuits. The low-leakage TCAM cells trade off the speed of READ and WRITE operations for smaller cell area and leakage.

We also integrated the individual TCAM components on a TCAM prototype chip. We described chip-level design and test considerations that affect the overall speed, power consumption and testability of the TCAM chip. During the course of this work, we designed and tested three chips. The key observations and recommendations for successful implementation and testing of a TCAM chip were also described. Finally, the measurement results of different components were presented. We also calculated the speed and power consumption of a 9Kb (64x144) TCAM block, and compared it with the existing comparable designs in the same process technology. Our design show noticeable improvement in terms of a smaller normalized power than the existing designs. The comparison also demonstrates the effectiveness of the techniques proposed in this work.

Current trends are showing more research focus towards architecture-level power-reduction techniques for application-specific TCAMs. Some search engines are also using algorithmic techniques to emulate TCAM-like operation using SRAMs. However, their applications are limited to very specific cases, and more circuit techniques to reduce TCAM power consumption are still needed. An emerging issue for commercial TCAMs is the peak power consumption. Since the package and PCB must be designed to support the worst-case power consumption, they require significant over-design to be able to support the peak power. This over-design increases the manufacturing costs of both the chip and the PCB. Another significant source of power consumption in TCAMs is the frequent switching of SLs, which has not been addressed in this work. Hierarchical SLs reduce power consumption by increasing the circuit complexity and sacrificing layout density. Future research can be carried out in understanding the search algorithms and applying that information to reduce the switching activity in SLs. In addition, innovative circuit techniques can be developed for the comparison logic to reduce the voltage swing and capacitance of SLs. Since large cell area is also a serious concern for large-capacity TCAMs, future research can also include the design of low-area TCAM cells that are compatible with the standard CMOS process. Non-volatile TCAMs can also be explored if the process technology supports the integration of high-speed logic and non-volatile memory.

Publications from this Work

[1] N. Mohan, and M. Sachdev, "A Static Power Reduction Technique for Ternary Content Addressable Memories," *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Niagara Falls, pp. 711-714, May 2-5, 2004.

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

[3] N. Mohan, and M. Sachdev, "A comparative study of low-power techniques for ternary CAMs," *Proceedings of the IEEE International Conference for Upcoming Engineers (ICUE)*, Waterloo, Ontario, Canada, May 13-14, 2006.

[4] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "Design techniques and test methodology for low-power TCAMs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 6, pp. 573-586, Jun. 2006.

[5] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "Match line sense amplifiers with positive feedback for low-power content addressable memories," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, Sep. 11-13, 2006.

[6] N. Mohan, and M. Sachdev, "Novel ternary storage cells and techniques for leakage reduction in ternary CAM," *Proceedings of the IEEE International SOC Conference (SOCC)*, Austin, Texas, Sep. 24-27, 2006.

[7] N. Mohan, W. Fung, and M. Sachdev, "Low-power priority encoder and multiple match detection circuit for ternary content addressable memory," *Proceedings of the IEEE International SOC Conference (SOCC)*, Austin, Texas, Sep. 24-27, 2006.

[8] N. Mohan, and M. Sachdev, "Low-capacitance and charge-shared match lines for low-energy high-performance TCAMs," submitted to *IEEE Journal of Solid-State Circuits*.

[9] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "A ternary CAM with positive-feedback match line sense amplifiers and a match-token priority encoding scheme," submitted to *IEEE Journal of Solid-State Circuits*.

[10] D. Wright, N. Mohan, and M. Sachdev, "A comprehensive test method for TCAMs," submitted to *IEEE Design and Test of Computers*.

Bibliography

[1] A. S. Tanenbaum, "*Computer Networks*," Prentice Hall, Upper Saddle River, NJ, 2003.

[2] P. Gupta, "*Algorithms for routing lookups and packet classification*," Ph.D. Thesis, Department of Computer Science, Stanford University, CA, 2000.

[3] R. Sangireddy, and A. K. Somani, "High-speed IP routing with binary decision diagrams based hardware address lookup engine," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 513-521, May 2003.

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

[5] A. J. McAuley, and P. Francis, "Fast routing table lookup using CAMs," *Proceedings INFOCOM'93*, San Francisco, CA, vol. 3, pp. 1382-1391, 1993.

[6] H. Higuchi, S. Tachibana, M. Minami, and T. Nagano, "A 5-mW, 10-ns cycle TLB using a high-performance CAM with low-power match detection circuits," *IEICE Transactions on Electronics*, vol. E79-C, no. 6, Jun. 1996.

[7] M. Sumita, "A 800 MHz single cycle access 32 entry fully associative TLB with a 240ps access match circuit," *Digest of Technical Papers of the Symposium on VLSI Circuits*, pp. 231-232, Jun. 2001.

[8] P.-F. Lin, and J. B. Kuo, "A 1-V 128-kb four-way set-associative CMOS cache memory using wordline-oriented tag-compare WLOTC structure with the content-addressable memory (CAM) 10-transistor tag cell," *IEEE Journal of Solid-state Circuits*, vol. 36, no. 4, pp. 666-675, Apr. 2001.

[9] P.-F. Lin, and J. B. Kuo, "A 0.8-V 128-kb four-way set-associative two-level CMOS cache memory using two-stage wordline/bitline-oriented tag-compare (WLOTC/BLOTC) scheme," *IEEE Journal of Solid-state Circuits*, vol. 37, no. 10, pp. 1307-1317, Oct. 2002.

[10] J. P. Wade, and C. G. Sodini, "A ternary content-addressable search engine," *IEEE Journal of Solid-state Circuits*, vol. 24, no. 4, Aug. 1989.

[11] K.-J. Lin, and C.-W. Wu, "A low-power CAM design for LZ data compression," *IEEE Transactions on Computers*, vol. 49, no. 10, Oct. 2000.

[12] T. Ogura, M. Nakanishi, T. Baba, Y. Nakabayshi, and R. Kasai, "A 336-kb content addressable memory for highly parallel image processing," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC 1996)*, pp. 273-276, May 1996.

[13] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern-matching using TCAM," *Proceedings of the IEEE International Conference on Network Protocols (ICNP'04)*, Berlin, Germany, pp. 5.1.1-5.1.10, Oct. 5-8, 2004.

[14] F. Yu, and R H. Katz, "Efficient multi-match packet classification with TCAM," *Proceedings of the IEEE Symposium on High Performance Interconnects (HOTI'04)*, Stanford, CA, pp. 2.1.1-2.1.7, Aug. 25-27, 2004.

[15] 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 Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1601 – 1609, Nov. 1996.

[16] R. Foss, and A. Roth, "Priority encoder circuit and method for content addressable memory," Canadian Patent 2365891, Apr. 30, 2003.

[17] W. Fung, "Low power circuits for multiple match resolution and detection in ternary CAM," *MASc thesis*, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, 2004.

[18] M. Kobayashi, T. Murase, and A. Kuriyama, "A longest prefix match search engine for multi-gigabit IP processing," *Proceedings of the IEEE International Conference on Communications*, vol. 3, pp. 1360-1364, June 2000.

[19] NetLogic Microsystems Inc., "CIDR longest prefix matching in network search engines," *Application Note NCS05*, Mar. 2001, http://www.netlogicmicro.com/pdf/ncs05_r1_5.pdf.

[20] S. Sharma, and R. Panigrahy, "Sorting and searching using ternary CAMs," *Proceedings of the Symposium on High Performance Interconnects (HOTI'02)*, Aug. 2002.

[21] J. M. Rabaey, "*Digital Integrated Circuits: A Design Perspective*," First Edition, Prentice-Hall Inc., Upper Saddle River, New Jersey, 1996.

[22] S. Ma, and P. Ma, "Multiple match detection circuit and method," Canadian Patent 2310295, Nov. 30, 2001.

[23] K. Batson, R. Busch, G. Koch, F. Towler, and R. Wistort, "Redundant array architecture for word replacement in CAM," U.S. Patent 6791855, Sep. 14, 2004.

[24] H. Noda, K. Inoue, M. Kuroiwa, F. Igaue, K. Yamamoto, H. J. Mattausch, T. Koide, A. Amo, A. Hachisuka, S. Soeda, I. Hayashi, F. Morishita, K. Dosaka, K. Arimoto,

K. Fujishima, K. Anami, and T. Yoshihara, "A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 245-253, Jan. 2005.

[25] S. S. Iyer, and H. L. Kalter, "Embedded DRAM technology: opportunities and challenges," *IEEE Spectrum*, vol. 36, no. 4, pp. 56-64, Apr. 1999.

[26] Semico Research Corporation, "Reassessing the content addressable memory market," Jun. 2003, [Online], Available: http://www.semico.com.

[27] Netlogic Microsystems, "NSE5000 network search engine product brief," Oct. 2003, http://www.netlogicmicro.com/datasheets/nse5000.html.

[28] Cypress Semiconductor, "Ayama(TM) 20000 network search engine family data sheet," [Online], Available: http://www.cypress.com.

[29] Integrated Device Technology (IDT), "IDT 75K72100 network search engine datasheet brief," [Online], Available: http://www.idt.com.

[30] F. Shafai, K. J. Schultz, G. F. R. Gibson, A. G. Bluschke, and D. E. Somppi, "Fully parallel 30-MHz, 2.5-Mb CAM," *IEEE Journal of Solid-state Circuits*, pp. 1690-1696, vol. 33, no. 11, Nov. 1998.

[31] A. Efthymiou, and J. D. Garside, "A CAM with mixed serial-parallel comparison for use in low energy caches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 3, pp. 325-329, Mar. 2004.

[32] 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 Journal of Solid-state Circuits*, vol. 38, no. 1, pp. 155-158, Jan. 2003.

[33] A. Roth, D. Foss, R. McKenzie, and D. Perry, "Advanced ternary CAM circuits on 0.13µm logic process technology," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pp. 465-468, Oct. 2004.

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

[35] G. Kasai, Y. Takarabe, K. Furumi, and 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," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pp. 387-390, Sep. 2003.

[36] C. A. Zukowski, and S.-Y. Wang, "Use of selective precharge for low-power content-addressable memories," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1788-1791, Jun. 1997.

[37] K. Pagiamtzis, and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pp. 383-386, Sep. 2003.

[38] K. J. Schultz, and P. G. Gulak, "Architectures for large-capacity CAMs," *Integration, The VLSI Journal*, vol. 18, no. 2-3, pp. 151-171, Jun. 1995.

[39] K. Pagiamtzis, and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: a tutorial and survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 712-727, Mar. 2006.

[40] R. Panigrahy, and S. Sharma, "Reducing TCAM power consumption and increasing throughput," *Proceedings of the Symposium on High Performance Interconnects*, pp. 107-112, 2002.

[41] F. Zane, G. Narlikar, and A. Basu, "CoolCAMs: power-efficient TCAMs for forwarding engines," *Proceedings of the IEEE INFOCOM*, vol. 1, pp. 42-52, 2003.

[42] C.-S. Lin, J.-C. Chang, and B.-D. Liu, "A low-power precomputation-based content-addressable memory," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 654-662, Apr. 2003.

[43] K.-J. Lin, and C.-W. Wu, "A low-power CAM design for LZ data compression," *IEEE Transactions on Computers*, vol. 49, no. 10, Oct. 2000.

[44] B. Chatterjee, M. Sachdev, S. Hsu, R. Krishnamurthy, and S. Borkar, "Effectiveness and scaling trends of leakage control techniques for sub-130nm CMOS technologies," *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, Seoul, pp. 122-127, Aug. 25-27, 2003.

[45] R. X. Gu, and M. I. Elmasry, "Power dissipation analysis and optimization of deep submicron CMOS digital circuits," *IEEE Journal of Solid-state Circuits*, vol. 31, no. 5, pp. 707-713, May 1996.

[46] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305-327, Feb. 2003.

[47] Y. Taur, and T. H. Ning, "*Fundamentals of Modern VLSI Devices*," New York: Cambridge University Press, 1998.

[48] K. F. Schuegraf, and C. Hu, "Hole injection SiO_2 breakdown model for very low voltage lifetime extrapolation," *IEEE Transactions on Electron Devices*, vol. 41, no. 5, May 1994.

[49] N. Mohan, and M. Sachdev, "A Static Power Reduction Technique for Ternary Content Addressable Memories," *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Niagara Falls, pp. 711-714, May 2-5, 2004.

[50] Predictive Technology Model (PTM), [Online], Available: http://www.eas.asu.edu/~ptm

[51] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pp. 201-204, 2000.

[52] V.C. Ravikumar, R.N. Mahapatra, and L.N. Bhuyan, "EaseCAM: an energy and storage efficient TCAM-based router architecture for IP lookup," *IEEE Transactions on Computers*, vol. 54, no. 5, pp. 521- 533, May 2005.

[53] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "Design techniques and test methodology for low-power TCAMs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 6, pp. 573-586, Jun. 2006.

[54] PAIX Routing Information Service (RIS) Raw Data (http://www.ripe.net/), Feb. 10, 2006, [Online], Available: http://data.ris.ripe.net/rrc14/2006.02/

[55] H. Tran, "Demonstration of 5T SRAM and 6T dual-port RAM cell arrays," *Proceedings of the IEEE Symposium on VLSI Circuits*, Honolulu, pp. 13-15, 13-15 Jun. 1996.

[56] D. Rimondi, "Low power SRAM memory cell having a single bit line," US Patent 6459611, Oct. 1, 2002.

[57] N. Mohan, and M. Sachdev, "Novel ternary storage cells and techniques for leakage reduction in ternary CAM," submitted to the *IEEE International SOC Conference (SOCC)*, Austin, Texas, Sep. 24-27, 2006.

[58] N. Mohan, and M. Sachdev, "Low-capacitance and charge-shared match lines for low-energy high-performance TCAMs," submitted to *IEEE Journal of Solid-State Circuits*.

[59] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE Journal of Solid-state Circuits*, pp. 956-968, vol. 36, no. 6, Jun. 2001.

[60] D. Shah, and P. Gupta, "Fast updating algorithms for TCAMs," *IEEE Micro*, pp. 36-47, 2001.

[61] Z. Wang, H. Che, M. Kumar, and S. K. Das, "CoPTUA: consistent policy table update algorithm for TCAM without locking," *IEEE Transactions on Computers*, vol. 53, no. 12, Dec. 2004.

[62] M. Miller, and B. Tezcan, "Investigating design criteria for searching databases," White Paper, Integrated Device Technology, [Online], Feb. 2005, Available: http://www.idt.com/content/solutions_InvestigatingDesignCriteriaForSearchingDatabases_wp.pdf

[63] K. Lakshminarayanan, A. Rangarajan, and S. Venkatachary, "Algorithms for advanced packet classification with ternary CAMs," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 193 - 204, Oct. 2005.

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

[65] J.-K. Kim, P. Vlasenko, D. Perry, and P. B. Gillingham, "Low power content addressable memory architecture," US Patent 6584003, Jun. 24, 2003.

[66] K. Pagiamtzis, and A. Sheikholeslami, "A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1512–1519, Sep. 2004.

[67] M. Maymandi-Nejad, and M. Sachdev, "A digitally programmable delay element: design and analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 5, pp. 871-878, Oct. 2003.

[68] I. Arsovski, and A. Sheikholeslami, "A mismatch-dependent power allocation technique for match-line sensing in content addressable memories," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1958-1966, Nov. 2003.

[69] I. Arsovski, "Circuits for high-density low-power content addressable memories," *M.A.Sc. thesis*, Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, p. 34, 2003.

[70] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "A ternary CAM with positive-feedback match line sense amplifiers and a match-token priority encoding scheme," submitted to *IEEE Journal of Solid-State Circuits*.

[71] K. W. Chew, J. Zhang, K. Shao, W. B. Loh, and S-. F. Chu, "Impact of Deep N-well Implantation on Substrate Noise Coupling and RF Transistor Performance for Systems-on-a-Chip Integration," *Proceedings of the 32nd European Solid-State Device Research Conference (ESSDERC)*, pp. 251-254, Sep. 24-26, 2002.

[72] D. Johns, and K. Martin, "*Analog Integrated Circuit Design*," First Edition, John Wiley & Sons Inc., New York, 1997.

[73] D. Wright, "A Comprehensive Test and Diagnostic Strategy for TCAMs," *MASc thesis*, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, 2004.

[74] Cadence Design Systems, "Virtuoso® Layout Editor User Guide," Sep. 2003.

[75] K. Schultz, and P. G. Gulak, "Fully-parallel integrated CAM/RAM using pre-classification to enable large capacities", *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 689-699, May 1996.

[76] M. Miller, M. Baumann, "Content addressable memory array having flexible priority support," U.S. Patent 6996662, Feb. 7, 2006.

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

[78] Synopsys, Inc., "NanoSim User Guide," Jun. 2004.

[79] J. Wang and C. Huang, "High-Performance Encoder With Priority Lookahead," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 10, pp.1511-1514, Oct. 2000.

[80] M. J. Akhbarizadeh, M. Nourani, D. S. Vijayasarathi, and P. T. Balsara, "A nonredundant ternary CAM circuit for network search engines," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 3, pp. 268-278, Mar. 2006.

[81] Cypress Semiconductor, "CYNSE70064A network search engine family data sheet," [Online], Nov. 9, 2004, Available: http://www.cypress.com.

[82] SiberCore Technologies, "SiberCAM Ultra-2M SCT2000C Large-capacity Content Addressable Memory Data Sheet," [Online], May 5, 2002, Available: http://www.sibercore.com.

Glossary

ATM	Asynchronous transfer mode
BE	Block enable in multiple match resolver
BIST	Built-in-self-test
BL	Bit line
BLSA	Bit line sense amplifier for READ operation
CAM	Content addressable memory
CIDR	Classless inter-domain routing
CIMDC	Charge-injection match detection circuit
C_{ML}	Match line capacitance
CMOS	Complementary metal-oxide-semiconductor circuit/technology
CR-MLSA	Current-race match line sense amplifier
DBL	Dummy bit line for differential READ operation
DFT	Design for testability
DIBL	Drain induced barrier lowering in scaled MOS transistors
DML	Dummy match line for timing generation in match line sensing
DNW	Deep n-well in CMOS bulk technology
DRAM	Dynamic random access memory
EDP	Energy delay product
E_{ML}	Energy consumed in match line sensing
FN	Fowler-Nordheim tunneling
GND	Ground
GSL	Global search line in hierarchical search line architecture
IC	Integrated circuit
I_{GOFF}	NMOS gate leakage when the transistor is OFF
I_{GON}	NMOS gate leakage when the transistor is ON
I_{GOFFP}	PMOS gate leakage when the transistor is OFF
I_{GONP}	PMOS gate leakage when the transistor is ON
I_{ML0}	Match line to ground current under match condition
I_{MLk}	Match line to ground current under k-bit mismatch condition

I_{OFF}	Match line to ground leakage through a pull down path
I_{ON}	Match line to ground 'ON' current through a pull down path
IP	Internet protocol
IPv4	Internet protocol version 4
IPv6	Internet protocol version 6
I_{SN}	Subthreshold leakage through an NMOS transistor
I_{SP}	Subthreshold leakage through a PMOS transistor
LA	Look ahead signal in multiple match resolver
LSB	Least significant bit
LSL	Local search line in hierarchical search line architecture
LZ	Lempel Ziv data compression algorithms
MAE	Match address encoder
Mb	Mega bit
MD-MLSA	Mismatch dependent match line sense amplifier
MiM	Metal-insulator-metal capacitance
ML	Match line
ML_0	Match line with match condition (zero mismatches)
ML_1	Match line with 1-bit mismatch
ML_k	Match line with k-bit mismatch ($k \geq 1$)
ML1	First segment of a segmented match line
ML2	Second segment of a segmented match line
MLSA	Match line sense amplifier
MMR	Multiple match resolver
MSB	Most significant bit
Msps	Million searches per second
NC-TCAM	NMOS-coupled ternary content addressable memory
NMOS	N-channel metal-oxide-semiconductor transistor
NPU	Network processing unit
NSE	Network search engine
OC-48	Optical carrier network line with transmission speeds of up to 2.5Gbps
OC-192	Optical carrier network line with transmission speeds of up to 10Gbps

OC-768	Optical carrier network line with transmission speeds of up to 40Gbps
PCB	Printed circuit board
PC-TCAM	PMOS-coupled ternary content addressable memory
PE	Priority encoder
P_{ML1}	Probability of finding a mismatch in ML1 of a segmented match line
P_{ML2}	Probability of finding a mismatch in ML2 of a segmented match line
PMOS	N-channel metal-oxide-semiconductor transistor
PTM	Predictive technology model
QoS	Quality of service
RAM	Random access memory
SL	Search line
SoC	System on a chip
SOI	Silicon on insulator
SPICE	Simulation Program with Integrated Circuits Emphasis
SRAM	Static random access memory
TCAM	Ternary content addressable memory
TLB	Translation lookaside buffer
V_{DD}	Positive power supply voltage
V_{DD_CELL}	Positive power supply voltage of the TCAM storage portion/cell
WL	Word line
5T-SRAM	Static random access memory cell made of 5 transistors
6T-SRAM	Static random access memory cell made of 6 transistors