A Mismatch-Dependent Power Allocation Technique for Match-Line Sensing in Content-Addressable Memories

Igor Arsovski, *Student Member, IEEE,* and Ali Sheikholeslami, *Senior Member, IEEE*

***Abstract*—In the conventional content-addressable memory (CAM), equal power is consumed to determine if a stored word is matched to a search word or mismatched, independent of the number of mismatched bits. This paper presents a match-line (ML) sensing scheme that allocates less power to match decisions involving a larger number of mismatched bits. Since the majority of CAM words are mismatched, this scheme results in a significant CAM power reduction. The proposed ML sensing scheme is implemented in a 256 $\times$ 144-bit ternary CAM for a 0.13-μm 1.2-V CMOS logic process. For a 2-ns search time on a 144-bit word, the proposed scheme saves 60% of the power consumed by the conventional sensing scheme.**

***Index Terms*—Associative memory, content-addressable memory (CAM), current sensing, high speed, low power, match-line sensing, mismatch dependent, neural network, pattern matching, string matching.**

I. Introduction

CONTENT-ADDRESSABLE memory (CAM) searches for matching data by content and returns the address at which the matching data is found. CAMs are used extensively today in applications such as network address translation, pattern recognition, and data compression. In these applications, there is a steady demand for CAMs with higher density and higher search speed, but at constant power. Currently, commercial CAMs are limited to 18 Mb of storage and 100 million searches per second on a 144-bit search word, at typically 5 W per CAM chip. Compared to the conventional memories of similar size, CAMs consume considerably larger power. This is partly due to the fully parallel nature of the search operation, in which a search word is compared in parallel against every stored word in the entire CAM array. Several techniques [1]–[6] have been developed to reduce the power consumption in CAMs, which we review in the remainder of this section. However, in all these techniques, the power is distributed uniformly among all the compare decisions, independent of the number of mismatched bits in a stored word. We propose [7] distributing power based on the difficulty of the match decisions, where match decisions involving a larger number of mismatched bits consume less power compared to other decisions. This results in 60% power reduction without compromising the search speed.

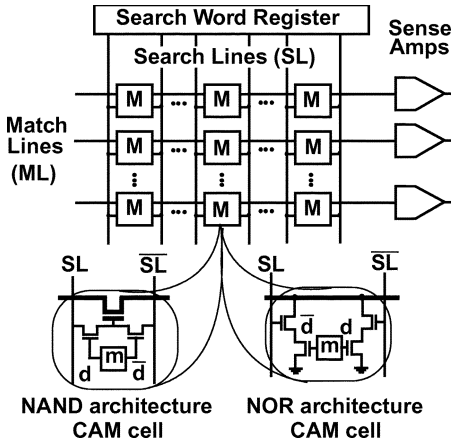


Fig. 1. Simplified CAM architecture illustrating two types of CAM cells.

Fig. 1 shows the basic block diagram of a CAM, consisting of an array of storage elements, a search-word register, and a column of sense amplifiers. Each row of the array stores one word (144 bits in this work) and has one associated match line (ML). The ML is used to signal whether the stored word matches or mismatches the search word. The search word is supplied on search lines (SLs) and compared bitwise against each stored word. As a result of this parallel comparison, the voltage on the corresponding ML changes (in a mismatch case) or does not change (in a match case). The major portion of CAM power is consumed during this parallel comparison, where all of the highly capacitive MLs are charged and discharged in every cycle.

One way to reduce CAM power is to reduce the switching capacitance on the MLs by using the NAND ML architecture [1]. This architecture consists of a number of NAND-type CAM cells (shown in Fig. 1) connected in series to create a long pass transistor network. In case of a match, a signal driven from one end of the ML propagates to the other end. In case of a mismatch, the signal is stopped by the first mismatched CAM cell, as it turns off its corresponding pass transistor. Since on average most MLs are mismatched, the signal is stopped within the first two pass transistors, reducing the switching activity of the ML and saving power [2]. On the other hand, the NAND ML architecture suffers from unacceptably long search delays that grow quadratically with the number of CAM cells in series [2]. To achieve higher speed, a NOR ML architecture is preferred.

Manuscript received April 18, 2003; revised June 30, 2003. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: ali@eecg.utoronto.ca).

Digital Object Identifier 10.1109/JSSC.2003.818139

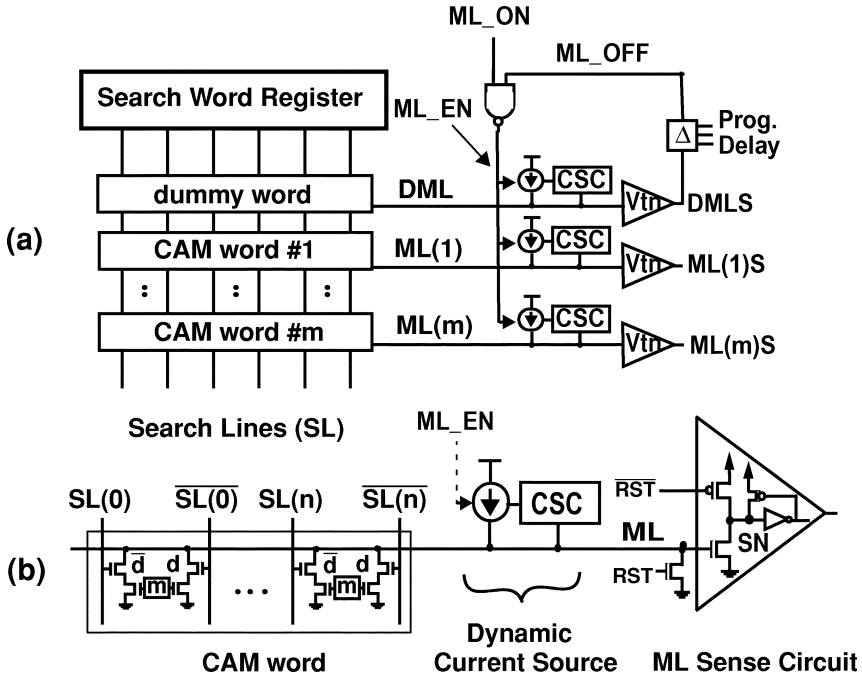


Fig. 2. Proposed ML sensing scheme. (a) General architecture. (b) Detailed schematic of a single CAM word.

The NOR architecture consists of CAM cells that are connected in parallel, instead of in series. Each CAM cell includes a four-transistor bit-compare circuit that is used to compare a search bit on the SL to a stored bit in the CAM cell. The compare circuits of all CAM cells in the same row are wire NORed to a corresponding ML and represent a single CAM word. When the search data is applied to the SLs, the bit-compare circuit in each CAM cell compares each search bit to its corresponding stored bit. A CAM cell storing a matching bit will isolate the ML from GND, while the one with a mismatched bit creates a path to GND through its bit-compare circuit. If all the bits in a stored word are identical to those of the search word, the ML has no path to GND, and remains in the high-impedance state. On the other hand, if there is one or more bit mismatches, one or more paths to GND are created, and the ML impedance is reduced accordingly. To use this architecture, the ML sensing circuits need to distinguish MLs with high impedence from MLs with low impedence. Conventionally, this ML sensing has been performed by precharging all MLs to $V_{\rm DD}$ and then applying the search data on the SLs. Matches (MLs with high impedence) remain at $V_{\rm DD}$ while mismatches (MLs with low impedence) discharge to GND. This sensing method achieves a higher search speed than the NAND sensing method, but at the price of higher power consumption, since all MLs are charged to $V_{\rm DD}$ and then discharged to GND in every cycle (except for the few MLs that are matched). In addition, the SL pairs contribute to the dynamic power consumption as one of the two SLs in a pair is always cycled between GND and $V_{\rm DD}$.

To reduce power while maintaining speed, several sensing techniques have been developed around the NOR ML architecture. One technique is to limit the voltage swing on the MLs [3], [4] to a value less than $V_{\rm DD}$, hence, reducing the ML portion of the dynamic power consumption. Another technique is to minimize the switching activity of the SLs [5], hence reducing the SL portion of the dynamic power consumption. In our previous work [6], we proposed a sensing scheme that limits the voltage swing on the MLs to $V_{\rm DD}/2$. Also, by precharging the MLs to GND (instead of to $V_{\rm DD}$), we eliminated the need for SL reset, hence, reducing the SL power consumption. In this work, we propose a new sensing scheme that distributes power nonuniformly to MLs, with MLs containing larger number of mismatched bits consuming less power. The overall effect of this technique is a 60% power reduction compared to the conventional NOR architecture [8], and 40% compared to our previous work [6].

The rest of this paper is organized as follows. Section II presents the method of operation and circuit implementation of the proposed ML sensing scheme. Section III analyzes this scheme and presents simulation results. Section IV presents the test chip architecture and Section V discusses further details of the implementation.

II. Mismatch-Dependent Power Allocation Technique for ML Sensing

A general architecture for the proposed ML sensing scheme is shown in Fig. 2(a). The CAM array consists of a search-word register, which holds an n-bit search word, and m memory rows that store the CAM entries being searched. Also included in the array is a dummy row, which is designed to mimic a matched word. As explained later, this row is used to provide a timing signal to other rows during the search operation.

To allow a simultaneous comparison between the search data and all entries of the stored data, the search-word register uses SLs to broadcast the search data to all the memory rows in the array. The search data is then compared, in parallel, against all row entries, resulting in matches and mismatches. Fig. 2(b) shows a detailed circuit diagram of a single row and the method

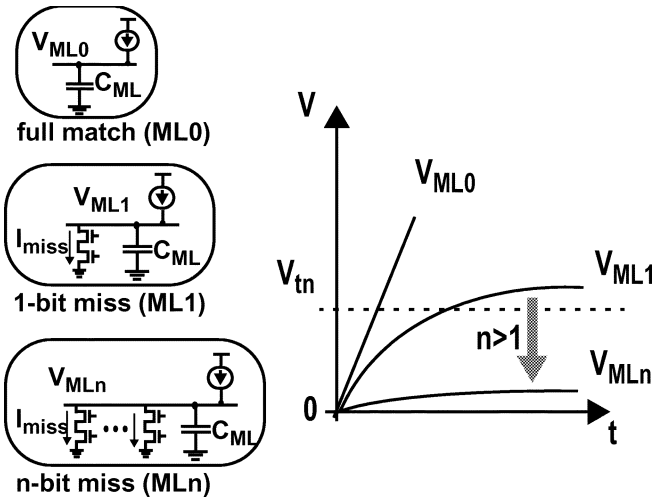


Fig. 3. Voltage on an ML0 (V_{ML0}) ramps faster than the voltage on an ML1 (V_{ML1}). The voltage on MLn ($V_{\mathrm{ML}n}$) stays close to GND.

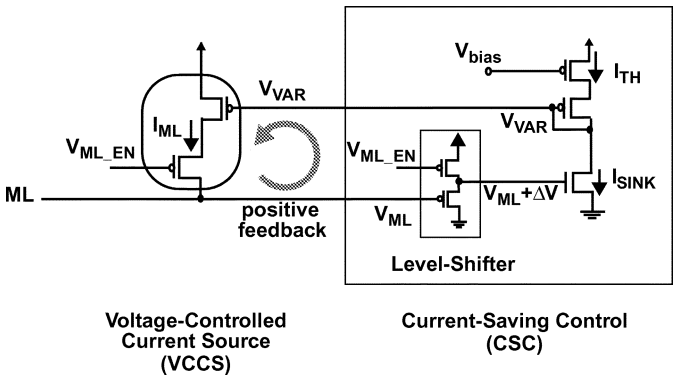


Fig. 4. Circuit implementation of the dynamic current source, consisting of a voltage-controlled current source and a current-saving control block. The CSC block monitors V_{ML} to supply less I_{ML} to mismatched MLs.

by which the comparison is performed. In this paper, we use ML0 to refer to an ML with no mismatches in the row (i.e., a fully matched ML) and ML1 to refer to an ML with only one-bit mismatch in the row. In general, we denote MLn to refer to an ML with n-bit misses in a single row. In contrast, we will use ML(m) to refer to an ML corresponding to row m in the array.

To differentiate between a match (ML0) and a mismatch (MLn, where $n > 0$), all MLs are charged by identical current sources causing their voltages to race toward a sense-voltage threshold. Assuming all current sources are identical and constant, the high-impedance ML0 will develop a voltage higher than that of an ML1, while the low-impedance MLn will stay close to GND. The sense circuitry detects this difference in voltage level, and differentiates between an ML0 and an MLn ($n > 0$). Since the voltage level of an MLn (where n is large) stays close to GND over the entire cycle, it would be power efficient to cut the current supplied to it shortly after this is realized. To save current, we have included a current-saving control (CSC) block on each ML (see Fig. 2) to monitor the voltage development on an ML and accordingly reduce the charging current as it becomes evident with time if an ML is mismatched. We will discuss the detailed circuit implementation of this block later in this section. We now describe in detail the search operation by referring to Fig. 2.

The search operation is performed in two steps. Prior to a search, the RST signal resets all MLs to GND, precharges all sense nodes (SN in each of the ML sense circuits) to V_{DD}, and supplies the new search data on the SLs. Current sources attached to each ML are then enabled, via ML_EN, and start charging all MLs with identical currents. Fig. 3 shows how the voltage on an ML ramps depending on the number of mismatches. Since an ML0 has no path to GND, it ramps faster than any MLn ($n > 0$). This current race continues until the voltage on an ML0 crosses the voltage threshold of the ML sense circuit, discharging the SN node through the nMOS device and signalling a match. At this point, all current sources are turned off with a "shut-off" signal from the dummy row, preventing even the closest mismatch (i.e., ML1) from reaching the sense threshold. To generate this signal, a dummy row is designed to always act as a match, independent of the search data. The dummy ML (DML) ramps past the sense threshold at the same time as an ML0 and signals a match with DMLS. By the time this shut-off signal reaches all the sense circuits, any ML0 has crossed the sense threshold and signalled a match, while any MLn ($n > 0$) has stayed below the sense threshold and signalled a miss. To account for process variations between different MLs, a programmable delay is placed in the path of this global shut-off signal. By shutting off the current supplied to all MLs, this scheme reduces the ML voltage swing, and by doing so decreases the ML power consumption [6].

To achieve further power savings on the ML, the current sources of this scheme have been designed to dynamically allocate less current to MLs with more mismatches. Since the mismatch level of an ML is not known prior to sensing, a small amount of energy is spent for an initial assessment of the state of each ML. To do this, the dynamic current sources start by supplying small identical currents to all MLs. This current develops an ML voltage (V_{ML}) which indicates the probable state of each ML. For example, for a given current, an ML0 develops a higher voltage compared to an ML1, since an ML0 does not leak any charge to GND. On the other hand, an MLn (where n is large) sinks its charge to GND and remains close to GND (as seen in Fig. 3). This scheme uses this V_{ML} to allocate more current to probable matches (MLs with high V_{ML}), and cut current to large mismatches (MLs with V_{ML} close to GND). By allocating less current to MLs with a lower V_{ML}, the dynamic current source effectively allocates less power to mismatches, thus saving power.

Fig. 4 shows the circuit-level implementation of the dynamic current source. The circuit consists of two blocks: a voltage-controlled current source (VCCS) and a CSC block which generates the control voltage. The VCCS is made up of two pMOS devices: one that is used as an enable switch and the other as a variable I_{ML} control, through the VAR node voltage (V_{VAR}). V_{VAR}, in turn, is generated in the CSC block, by comparing I_{SINK} to I_{TH}. I_{SINK} is a current proportional to V_{ML}, while I_{TH} is a constant current set by V_{bias} (discussed in Section VI). A fixed V_{ML} translates into a constant V_{VAR} and a corresponding constant I_{ML}. As V_{ML} rises from GND, it increases I_{SINK}, low-

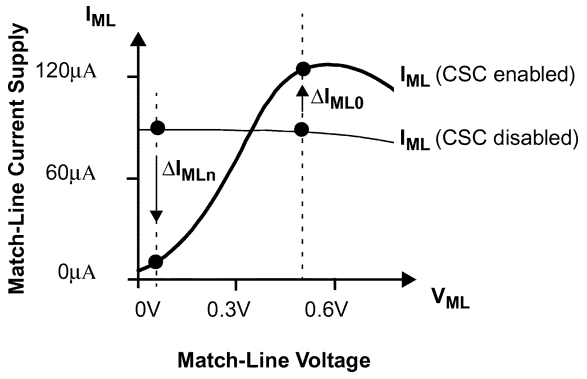


Fig. 5. Current supplied to an ML as a function of $V_{\rm ML}$. With the CSC block disabled, mismatches and matches receive similar $I_{\rm ML}$. With CSC enabled, large mismatches (MLn) receive less current than matches.

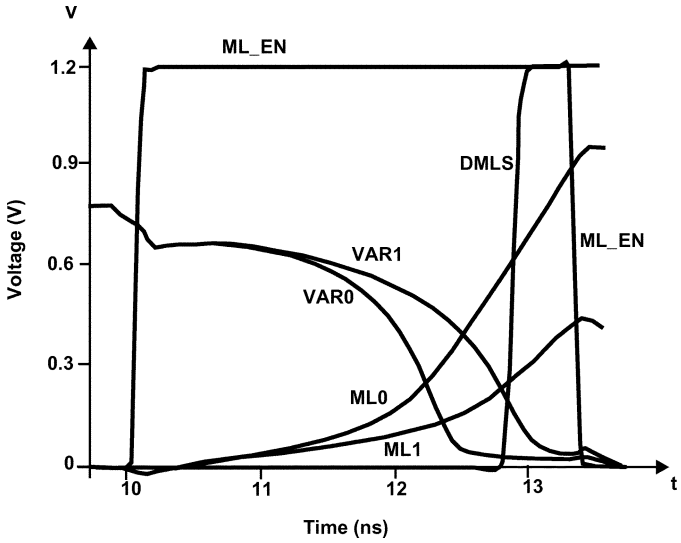


Fig. 6. Simulation results of power-optimized design showing voltage development on an ML0 (fully matched) and an ML1 (one-bit miss) along with their corresponding precharge-high VAR voltages ($V_{\rm VAR0}$ and $V_{\rm VAR1}$).

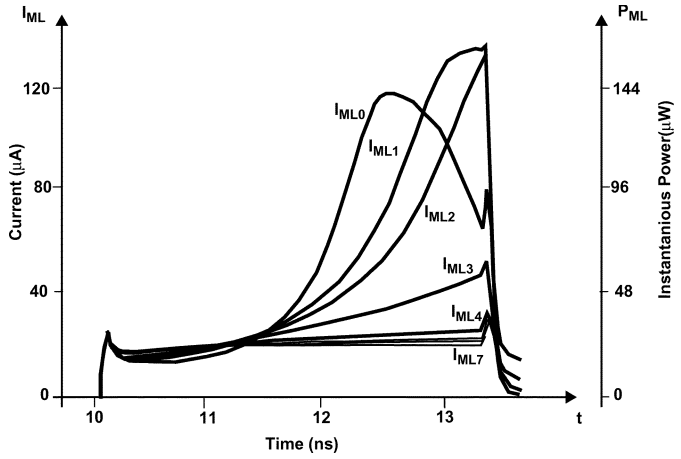


Fig. 7. Simulation results of power-optimized design showing current supplied to an ML0, ML1,…, ML7 (MLn is an ML with n-bit miss). MLs with larger misses receive less current.

ering $V_{\rm VAR}$ and hence, increasing $I_{\rm ML}$. This increase in $I_{\rm ML}$, in turn, causes a further increase in $V_{\rm ML}$, creating a positive feedback loop. To maintain this positive feedback at ML voltages lower than the sense threshold ($V_{\rm tn}$), a level-shifting circuit is included to eliminate the dead band from GND to $V_{\rm tn}$. This circuit simply shifts $V_{\rm ML}$ up by a voltage slightly larger than $V_{\rm tn}$ and ensures the nMOS transistor controlling $I_{\rm SINK}$ is slightly "on" even when the ML voltage is at GND level.

To see the effect of the CSC block, Fig. 5 shows $I_{\rm ML}$ as a function of $V_{\rm ML}$ for both the CSC block disabled and enabled. With CSC disabled and $V_{\rm VAR}$ being constant, $I_{\rm ML}$ is almost independent of $V_{\rm ML}$, staying constant over the entire sensing region (from GND to approx $V_{\rm DD}/2$). Thus, large mismatches that stay close to GND receive the same amount of current as full matches which develop a much higher $V_{\rm ML}$. On the other hand, with CSC enabled, $I_{\rm ML}$ starts out small when $V_{\rm ML}$ is close to GND, but rises above $I_{\rm ML}$ (CSC disabled) rapidly as $V_{\rm ML}$ increases. Thus, large mismatches, which stay close to GND, receive very small current, while matches, which ramp faster, receive more current as their $V_{\rm ML}$ increases. The current savings associated with enabling the CSC block are revealed in Fig. 5 by the reduction in the current provided to large mismatches ($\Delta I_{\rm MLn}$). Since statistically most MLs are largely mismatched, this reduction in current translates to overall current reduction. Furthermore, since $I_{\rm ML}$ (CSC enabled) supplies a similar charge over the search cycle as $I_{\rm ML}$ (CSC disabled), it achieves a similar search speed, with considerable power savings.

The dynamic current source described above is used to implement two different designs: one optimized for power consumption and another optimized for speed. In the power-optimized design, the VAR node is precharged to $V_{\rm DD}$–$V_{\rm tn}$, causing current sources to initially provide small currents to all MLs. In the speed-optimized design, the VAR node is precharged to GND, causing large initial current to all MLs. By varying the initial current, the scheme varies the initial charge contribution, and thereby the initial $V_{\rm ML}$. Higher $V_{\rm ML}$ increases the current sunk by the mismatched MLs and allows faster differentiation between an ML0 and an ML1. However, a higher $V_{\rm ML}$ also requires a larger energy investment by supplying larger currents. Simulation results of the two designs are presented in the next section.

III. Simulation Results and Analysis

The proper operations of both the power-optimized and the speed-optimized designs are verified by performing extensive HSPICE simulations. All simulations presented in this section compare an ML0 against an ML1 (the hardest-to-detect mismatch) on a 144-bit CAM word.

A. Design Optimized for Power Consumption

For accurate comparison of the ML voltages, the initial currents to all the MLs must be identical. For minimum power consumption, the initial currents are set close to zero by precharging all of the VAR nodes to $V_{\rm DD}$–$V_{\rm tn}$. As the search progresses, $I_{\rm SINK}$ pulls the VAR node below $V_{\rm DD}$–$|V_{\rm tp}|$ and starts supplying small identical currents to all MLs.

Figs. 6 and 7 show HSPICE simulation results for this design. Fig. 6 compares the voltage of an ML0 against the voltage on an ML1 for a 144-bit CAM word. In less than 1.5 ns after

the current sources have been enabled, the voltages on VAR0 and VAR1, corresponding to an ML0 and an ML1, respectively, begin to differ, helping an ML0 to receive more current and rise faster than an ML1. In less than 3 ns, the DML, whose voltage development is identical to that of the an ML0, triggers its ML sense circuit and initiates the shut-off sequence with DMLS. After a small delay, the ML_OFF signal causes ML_EN to drop, turning off all current sources and latching the search results. At this point, the voltage difference between an ML0 and an ML1 is beyond 300 mV. This is larger than any threshold-voltage mismatch in the ML sense circuits, ensuring correct search results.

To see the power-savings of this scheme, Fig. 7 compares the current supplied to different ML cases with time. This figure reveals that an ML0 receives more current, adding to the sensing speed, while an MLn ($n > 0$) receive less current, saving power. The current saving increases monotonically with the number of mismatches and reaches its maximum current saving for an ML7. Larger mismatches keep the ML very close to GND, and receive the same current from the dynamic current source. The dropoff in I_{ML0} occurring 2.5 ns into the cycle is caused by the voltage on ML0 which approaches V_{DD}, reducing the pMOS drain–source voltage. This current excursion has little effect on sensing since the voltage on ML0 has already crossed the sense threshold. For a 3.5-ns search time, the charge supplied to a 7-bit miss is 62% less than for the full match. Since statistically most 144-bit CAM words will have more than 7-bit mismatches, this charge saving translates to 62% reduction in ML energy per search when compared to our previous work [6], and 74% reduction when compared to the conventional precharge-high scheme [8].

B. Design Optimized for Search Speed

To increase search speed, the initial current supplied to the MLs is increased by precharging the VAR node to GND. This precharge level causes a spike of maximum I_{ML} which over a short time deposits large equal charge to all MLs. This develops a larger V_{ML} than the power-optimized design and helps speed up the differentiation between an ML0 and an ML1.

Figs. 8 and 9 show HSPICE simulation results for the current saving ML sensing scheme with the VAR node initially precharged to GND. Fig. 8 compares the voltages developing on the MLs (ML0 and ML1) and their respective VAR node voltages (VAR0 and VAR1). By precharging the VAR nodes to GND, the search cycle is decreased from 3.5 ns to 2 ns. In less than 0.5 ns, the voltage on VAR0 separates from that of VAR1, helping an ML0 to receive more current and rise faster than an ML1. In less than 2 ns, the voltage difference between an ML0 and an ML1 reaches beyond 200 mV. This is far larger than any threshold-voltage mismatch of the ML sense circuit and ensures correct sensing. Fig. 9 compares the current supplied to the MLs during the search operation. After the initial spike of identical current to all MLs, the amount of current supplied to each ML case varies with the number of mismatches on an ML. Similar to the power-optimized design, an ML0 receives the most current while MLn ($n > 0$) receives diminished current, reaching a near minimum for the ML7 case. In this mode, an MLn ($n > 7$) receives 48% less charge than an ML0. Therefore, to achieve a 2-ns match time, the energy saving between an ML7 and an ML0 has been lowered from 62% to 48%. Thus, in this version of the proposed scheme, higher speed is achieved at the cost of increasing the energy per search.

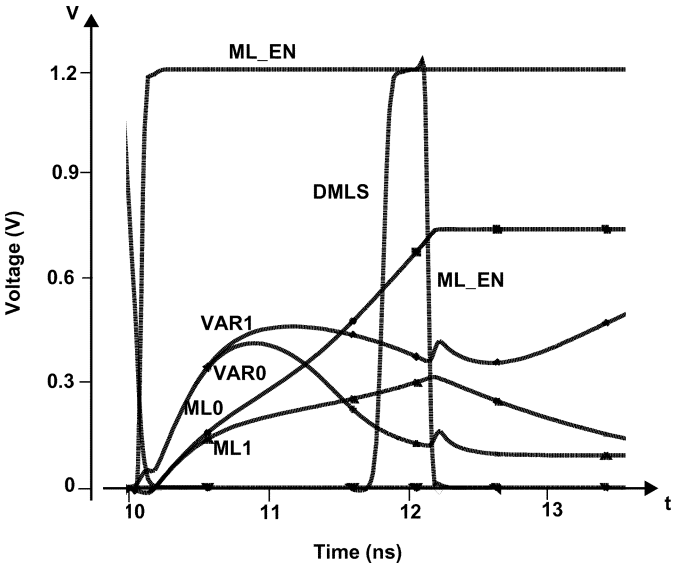


Fig. 8. Simulation results of speed-optimized design showing voltage development on an ML0 (fully matched) and an ML1 (one-bit miss) along with their corresponding precharge-low VAR voltages (V_{VAR0} and V_{VAR1}).

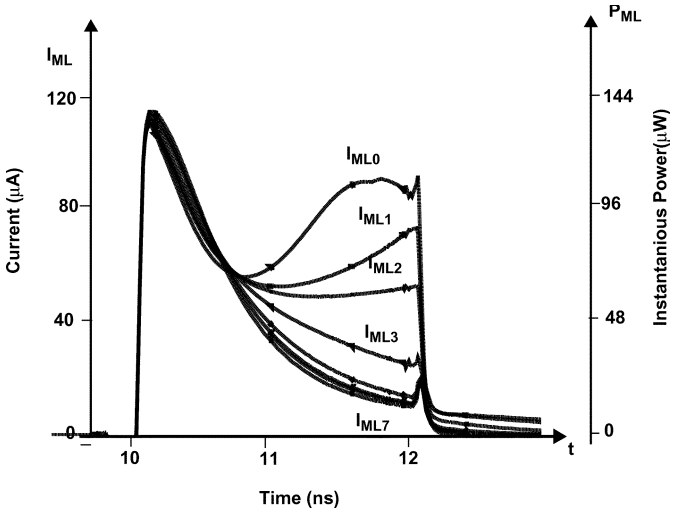


Fig. 9. Simulation results of speed-optimized design showing current supplied to an ML0, ML1,…, ML7 (MLn is an ML with n-bit miss). MLs with larger misses receive less current.

To compare the energy per search in the proposed scheme versus those of the conventional precharge-high [8] and the current-race schemes [6], we have simulated all three schemes in an array of 256 rows by 144 bits. For a fair comparison, the simulation is performed for similar search speed and reports the energy/bit/search spent on the MLs and the SLs. Fig. 10 shows the ML energy/bit/search spent by the three schemes as functions of the number of mismatches on an ML, while Fig. 11 summarizes the simulations and compares the average ML and the SL energy/bit/search spent by the three schemes.

Fig. 10 confirms that both the conventional and current-race schemes spend uniform ML energy/bit/search regardless of the number of mismatches. On the other hand, the proposed sensing

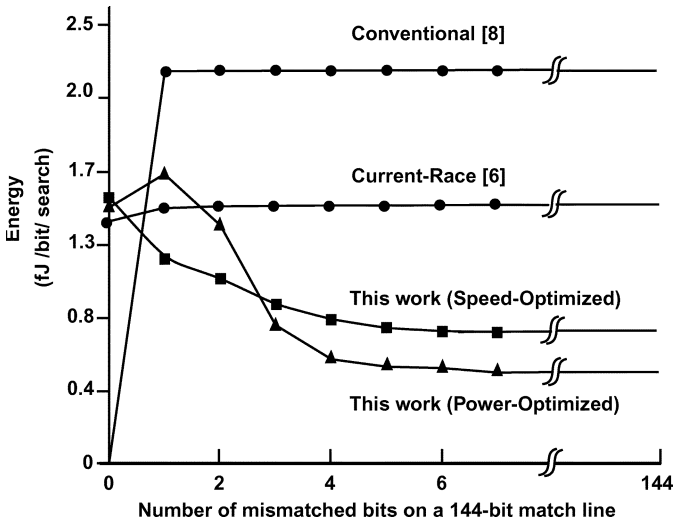


Fig. 10. Energy comparison of three ML sensing schemes: energy/bit/search as a function of number of mismatched bits on a 144-bit CAM word.

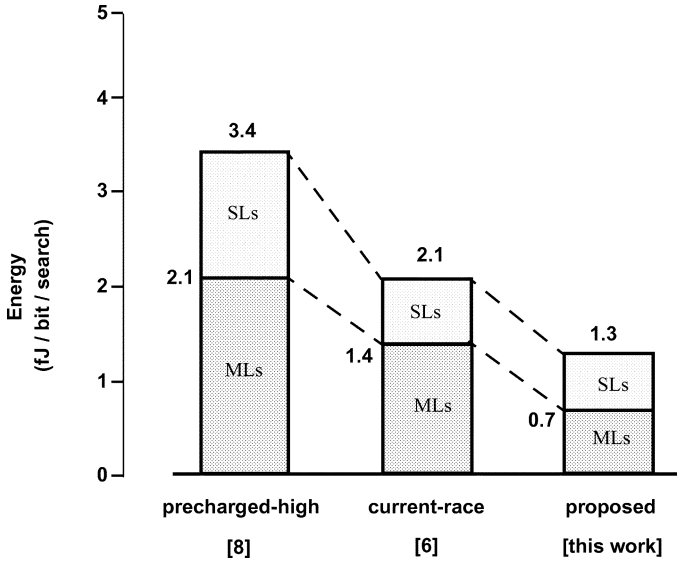


Fig. 11. Energy-per-search comparison of three ML sensing schemes.

scheme spends energy that is similar to that of the current-race scheme when an ML is fully matched, while spending significantly less when an ML is more than 7-bit mismatched. As summarized in Fig. 11, the proposed ML sensing scheme reduces the average ML energy/bit/search portion by 66% when compared to the precharge-high scheme, and by 48% when compared to the current-race ML sensing scheme. In addition, by precharging the MLs to GND instead of $V_{\rm DD}$, both current-based schemes eliminate the need for SL reset, also reducing the SL energy/search portion by 50%. The total energy/search savings of the proposed scheme stands at 60% when compared to the precharge-high scheme and by 40% when compared to the current-race sensing scheme. The energy overhead of the dynamic current source is only 2% of the total energy spent on a search of a single 144-bit word.

C. Process Variation Analysis and Worst Case Simulations

When dealing with a positive feedback ML sense amplifier, an obvious concern is sensitivity to process variations between the DML, which generates the timing, and other MLs which execute the word comparison. This section summarizes our study of the robustness of this scheme to process variations. Fig. 12(a) shows a simplified model for sensing an ML0 or an ML1 on a typical CAM row. The model consists of a current source $I_{\rm ML}$, a capacitor $C_{\rm ML}$ which models the parasitic capacitance on the ML, and a resistor R, which depends on the ML case. In the case of an ML0, there is no path to GND, thus, $R = \infty$. In the case of an ML1, there is a single path to GND through a mismatched bit-compare circuit, and therefore $R = R_{\rm miss}$. In this figure, $R_{\rm miss}$ represents the triode resistance of a single bit-compare circuit. As Fig. 12(b) shows, for a constant $I_{\rm ML}$, the voltage on both cases ramps until it reaches its final state, $V_{\rm DD}$ for an ML0 and $I_{\rm ML} \times R_{\rm miss}$ for an ML1. To ensure that an ML1 stays below the sense threshold, a conservative design could size the current source such that $I_{\rm ML} \times R_{\rm miss}$ is less than the lowest sense threshold for all process variations.

In the speed-optimized design, we size the current source by taking into account the ML current, the *RC* behavior of the ML, and the minimum delay from the DML match to the global shut-off signal (ML_OFF). Sizing the current source this way achieves high speed while ensuring that an ML1 does not signal a match under all process variations. Process variation analysis of a speed-optimized design is described below.

Due to the head start associated with high initial currents, the speed-optimized version of the proposed ML sensing scheme is more susceptible to process variations. To verify the worst case analysis of this scheme, two problem scenarios are identified. The first case compares the development of a typical DML against a slow ML0, and the second one compares a typical DML against a fast ML1. The potential problem with the first case is that the slow ML0 might not reach the sense threshold before DML sends its global shut-off signal. In contrast, the potential problem with the second case is that DML will send this signal too late, causing an ML1 to be detected as an ML0.

The first problem case can be easily handled by increasing the programmable delay on the fast DML to allow the voltage on the typical ML0 to cross the sense threshold before the global shut-off signal arrives. To accommodate the second problem case, the dynamic current source is sized such that the largest process variations between the current sources will never cause an ML1 to be sensed as an ML0. The worst case ML models along with a full simulation of the speed-optimized version of the proposed scheme are shown in Fig. 13. Fig. 13(a) shows the DML model consisting of the typical $C_{\rm ML}$, along with the nominal sizes for both its enable and variable control pMOS devices, and Fig. 13(b) shows the model for its fast-process ML1 (ML1f), which has a 20% reduced $C_{\rm ML}$ and 20% size increase in its pMOS transistors. Fig. 13(c) illustrates the voltage developed on both cases and compares the results. As expected, the 20% larger $I_{\rm ML}$ charging a 20% smaller $C_{\rm ML}$ causes an ML1f to ramp faster than DML initially. However, as the voltage on an ML1f increases, its bit-compare circuit sinks more of the provided current and slows down its ramping. In less than 1.5 ns

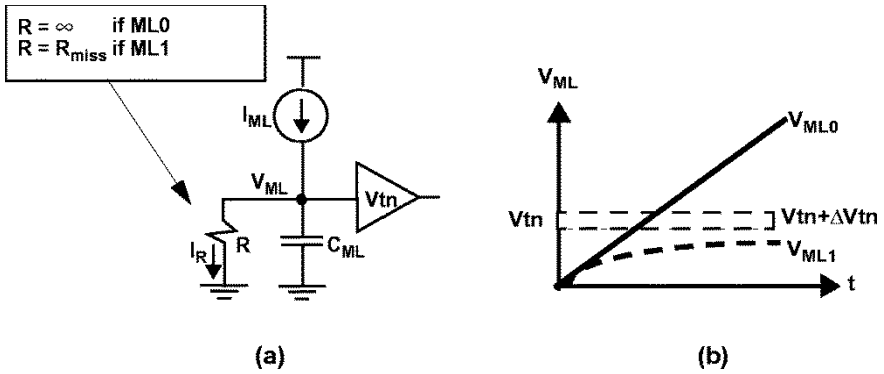


Fig. 12. (a) An ML0 is modeled as a single capacitor ($C_{\rm ML}$) while an ML1 is modeled as $C_{\rm ML}$ in parallel with a resistor ($R_{\rm miss}$). (b) Voltage development on an ML0 and an ML1. To ensure that $V_{\rm ML1}$ does not rise above $V_{\rm tn}$, $I_{\rm ML}$ is set to less than ($V_{\rm tn}/R_{\rm miss}$) for all process corners.

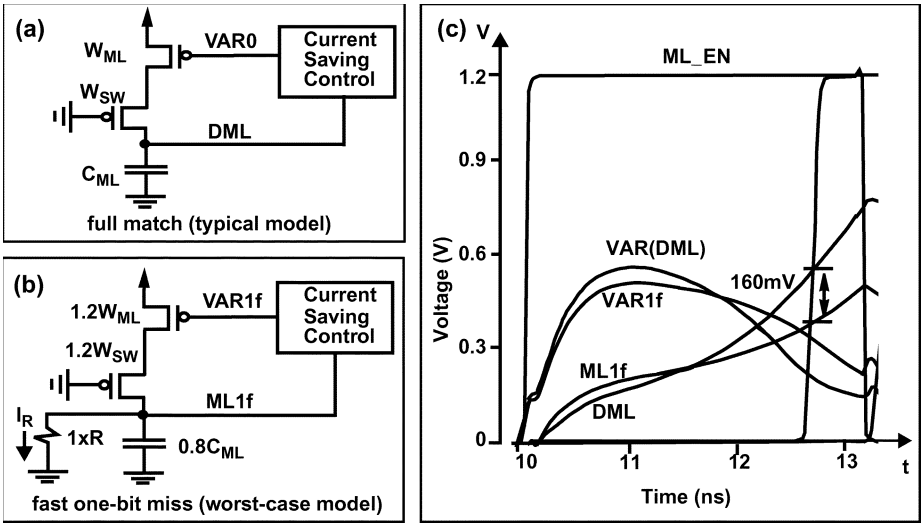


Fig. 13. Worst case analysis showing: (a) model of a typical ML0; (b) model of a fast ML1 (ML1f); (c) worst case simulation results. The fast ML1 initially ramps faster than an ML0, but later slows down due to $R_{\rm miss}$ and allows 160-mV sense margin at sensing time.

after the ML_EN signal, the voltage on an ML1 catches up, and by 2.5 ns, the voltage difference between the two allows for a 160 mV sense margin. Therefore, even for a large process variation the proposed ML sensing scheme produces correct search results. To accommodate process variations of this magnitude, the $V_{\rm bias}$ on the current-saving control block (shown in Fig. 4) is decreased, resulting in a 50% longer search cycle. This modification of the $V_{\rm bias}$ is discussed in Section VI.

This process-variation simulation also illustrates that the search results cannot be solely based on the initial assessment of the mismatch state of the ML. The voltage crossover between an ML0 and an ML1f, which occurs 1.5 ns into the search cycle, suggests that if either ML or the VAR voltage was sensed based on the initial assessment, the search result would have been incorrect. On the other hand, sensing over the entire region reduces the effects of ML noise and process variations and produces correct search results.

IV. Testchip Architecture

The layout of the proposed scheme is shown in Fig. 14. The test chip consists of four memory arrays that are sensed by four different ML sensing blocks (MLSBs).

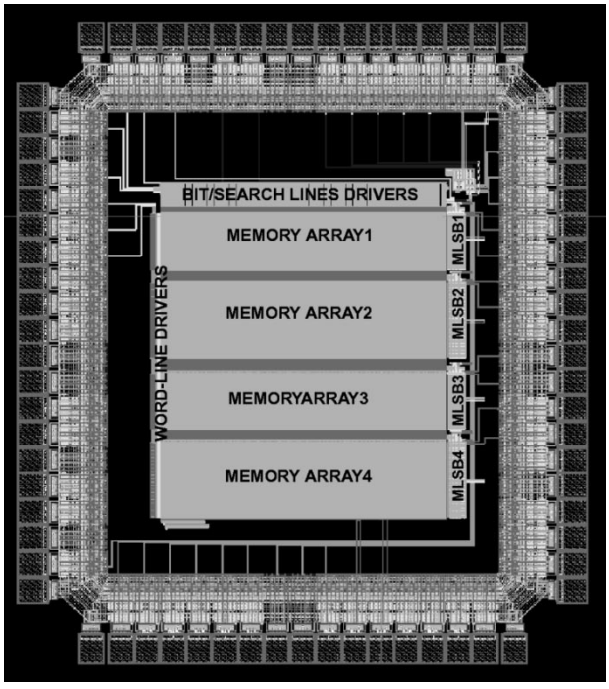


Fig. 14. Chip layout.

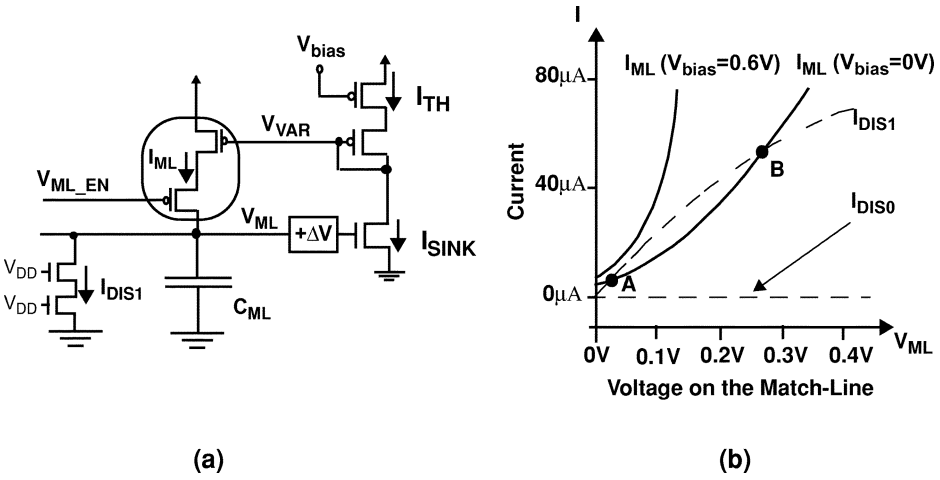


Fig. 15. The effect of current-saving control $V_{\rm bias}$ on $I_{\rm ML}$. (a) Model of CAM word sensing. (b) Simulation results of current development ($I_{\rm ML}$ and $I_{\rm discharge}$) as a function of $V_{\rm ML}$.

Each array consists of 64 words, containing 144 SRAM-based ternary CAM (TCAM) cells. In addition to "0" and "1", TCAM cells allow the storage of "don't cares", which act as wild cards and allow pattern matching between the search and stored data. This operation is particularly attractive for implementing longest prefix match searches in routing tables [9]. The four memory arrays contain four types of TCAM cells that use different storage devices to improve density, but use identical NOR-type bit-compare circuits to maintain search speed. The write operation is performed by shifting data serially into both bitline (BL) and wordline (WL) registers, and then driving this data on the BLs and WLs. The search operation is performed similarly by applying this data on the SLs, and then sensing the MLs with the MLSBs.

Each MLSB implements a sensing scheme that performs the search and stores the search results in its dedicated shift register. To examine the performance of the proposed scheme, MLSB1 and MLSB3 contain the power-optimized and the speed-optimized versions of the proposed scheme. MLSB2 and MLSB4 contain similar versions but with the CSC blocks disabled. To reproduce the power results presented in Section III, each block has dedicated power pins for measurement.

The chip is designed for implementation in a 1.2-V 0.13-μm CMOS process without resorting to special devices. The area overhead associated with using the current-saving blocks is less than 1% of the total design area when the CAM word size is 144 bits.

V. Discussion

This section discusses the effect of the $V_{\rm bias}$ level on circuit speed and sensitivity to process variations. Referring to Fig. 15(a), $V_{\rm bias}$ scales $I_{\rm ML}$ by controlling $I_{\rm TH}$ and $V_{\rm VAR}$. An increase in $V_{\rm bias}$ decreases $I_{\rm TH}$, hence, decreasing the level of $V_{\rm VAR}$ and increasing $I_{\rm ML}$. Thus, higher $V_{\rm bias}$ produces higher $I_{\rm ML}$. This has two effects on circuit performance: increased search speed and reduced sense margin. Higher $I_{\rm ML}$ increases search speed by reducing the time to charge up $C_{\rm ML}$. However, a higher $I_{\rm ML}$ decreases the sensing margin by reducing the effect of the discharge current from an ML1, causing the voltage on an ML1 to ramp at a similar rate to that of an ML0. To achieve high search speed and sufficient sense margin over different process variations, $I_{\rm ML}$ is controlled through $V_{\rm bias}$ externally.

Fig. 15 shows the effect of $V_{\rm bias}$ on ML sensing. Fig. 15(a) shows an ML sensing model, illustrating the current provided to an ML ($I_{\rm ML}$), along with the current sunk from it ($I_{\rm DIS}$). Fig. 15(b) illustrates $I_{\rm DIS}$ for two ML cases along with $I_{\rm ML}$ for two $V_{\rm bias}$ settings. The dashed-line curves represent the current discharged from both an ML0 and an ML1 ($I_{\rm DIS0}$ and $I_{\rm DIS1}$, respectively), while the solid curves represent $I_{\rm ML}$ for two extreme $V_{\rm bias}$ settings ($V_{\rm bias} = 0$ V and 0.6 V). $I_{\rm ML}$ increases $V_{\rm ML}$ while $I_{\rm DIS}$ decreases $V_{\rm ML}$. With both $I_{\rm ML}$ and $I_{\rm DIS}$ being functions of $V_{\rm ML}$, the intersection of these curves show the possible solutions for the $V_{\rm ML}$.

In the case of an ML0, $I_{\rm DIS} = 0$, causing $V_{\rm ML0}$ to ramp at a rate of $(I_{\rm ML}/C_{\rm ML})$ until the global shut-off signal arrives. In the case of an ML1, both $I_{\rm DIS}$ and $I_{\rm ML}$ are nonzero and increase with $V_{\rm ML}$, hence, $V_{\rm ML1}$ ramps slower than $V_{\rm ML0}$, and achieves a reduced final value. As we see next, both the slew rate and the final voltage of an ML1 are dependent on $V_{\rm bias}$.

With $\rm Vbias = 0$ V, $I_{\rm ML}$ starts initially larger than $I_{\rm DIS1}$, causing $V_{\rm ML1}$ to rise from its initial precharge state at GND toward $V_{\rm DD}$. As $V_{\rm ML}$ rises, so does $I_{\rm DIS1}$. This trend continues until $V_{\rm ML}$ reaches point A in Fig. 15(b), where $I_{\rm ML}$ ($V_{\rm bias} = 0$ V) becomes equal to $I_{\rm DIS1}$. Further increase in $V_{\rm ML}$ causes $I_{\rm DIS1}$ to become larger than $I_{\rm ML}$ ($V_{\rm bias} = 0$ V), forcing $V_{\rm ML}$ back to point A. The sensing margin when $V_{\rm bias} = 0$ is equal to the difference in voltage between an ML0 and an ML1 when the search is stopped with the global shut-off signal. Since $V_{\rm ML1}$ stays near GND at point A while $V_{\rm ML0}$ crosses the sense threshold at $V_{\rm tn}$, the sensing margin is close to $V_{\rm tn}$. The other intersection of the two curves (point B) is not a stable solution, as any voltage disturbance on ML forces $V_{\rm ML}$ toward point A or $V_{\rm DD}$.

With $V_{\rm bias} = 0.6$ V, on the other hand, $I_{\rm ML}$ is larger than $I_{\rm DIS1}$ over the entire $V_{\rm ML}$ region, charging an ML1 toward $V_{\rm DD}$

at the rate of $((I_{\mathrm{ML}} - I_{\mathrm{DIS1}})/C_{\mathrm{ML}})$. Since this rate of change is less than that of an ML0, V_{ML1} is always less than V_{ML0}. To guarantee that only V_{ML0} (and not V_{ML1}) passes the sense threshold, the delay of the global shut-off signal must arrive before V_{ML1} reaches V_{tn}.

In addition to search speed and process variation sensitivity, V_{bias} also controls the energy consumed per search. Referring to Fig. 15, increasing V_{bias} increases both the initial current provided to the ML and the slope at which this current rises with V_{ML}. These in turn increase the search speed and the energy per search. Thus, by controlling V_{bias}, this sensing scheme allows a further tradeoff between speed and energy per search. V_{bias} is made adjustable through three programable bits to allow a tradeoff between search speed and energy savings. A side benefit of this is that the chips often discarded as a result of their process variation from the typical can be used in a decreased V_{bias} setting.

VI. Conclusion

The proposed ML sensing scheme allocates power to match decisions based on the number of mismatched bits in each CAM word. With allocating less power to mismatched MLs and with most MLs being in this category, this scheme results in a considerable power reduction. The proposed scheme was implemented in a 265×144-bit TCAM for a 0.13-μm 1.2-V CMOS logic process. For a 2-ns search time on a 144-bit TCAM word, the proposed scheme uses 60% less power compared to the conventional precharge-high NOR scheme [8] and 40% compared to our previous work [6]. In addition, this scheme allows a tradeoff between search speed and energy per search. Finally, the proposed scheme can be used in conjunction with CAM architectures such as selective precharge [5] and preclassification [10].

Acknowledgment

The authors would like to thank T. Chandler, K. Pagiamtzis, and M. van Ierssel for insightful discussions on this work, and Canadian Microelectronics Corporation for test chip fabrication.

References

[1] 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 J. Solid-State Circuits*, vol. 33, pp. 666–676, Apr. 2001.

[2] Y. L. Hsiao, D. H. Wang, and C. W. Jen, "Power modeling and low-power design of content addressable memories," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 2001, pp. 926–929.

[3] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE J. Solid-State Circuits*, vol. 36, pp. 956–968, June 2001.

[4] G. Thirugnanam, N. Vijaykrishnan, and M. J. Irwin, "A novel low power CAM design," in *ASIC/SOC Conf. Proc.*, 2001, pp. 198–202.

[5] C. Zukowski and S. Wang, "Use of selective precharge for low-power content-addressable memories," in *Proc. IEEE Int. Symp. Circuits and Systems*, June 1997, pp. 1788–1791.

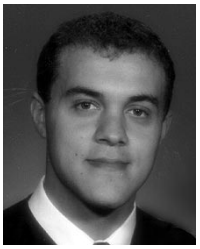
[6] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, vol. 38, pp. 155–158, Jan. 2003.

[7] I. Arsovski and A. Sheikholeslami, "A current-saving match-line sensing scheme for content- addressable memories," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2003, pp. 304–305.

[8] P. Lin and J. 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 J. Solid-State Circuits*, vol. 36, pp. 666–676, Apr. 2001.

[9] M. Kobayashi, T. Murase, and A. Kuriyama, "A longest prefix match search engine for multi-gigabit IP processing," in *IEEE Int. Conf. Communications*, vol. 3, June 2000, pp. 1360–1364.

[10] K. J. Schultz and P. G. Gulak, "Architectures for large capacity CAMs," *Integration: VLSI J.*, vol. 18, pp. 151–171, 1995.



Igor Arsovski (S'00) received the B.Sc. and M.A.Sc. degrees from the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, in 2001 and 2003, respectively.

He was with Mosaid Technologies, Canada, working on content-addressable memories (CAMs) for network applications, during the summer of 2000. He is currently working with IBM Microelectronics on the design of embedded CAMs. His research interests include high-speed low-power CAMs for networking applications, analog circuits, and VLSI memories such as DRAM and SRAM.

Mr. Arsovski recieved the Centennial Thesis Award for the best fourth-year design project from the University of Toronto in 2001 for his work on the design of high-density CAM cells.

Ali Sheikholeslami (S'98–M'99–SM'02) received the B.Sc. degree from Shiraz University, Shiraz, Iran, in 1990 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1994 and 1999, respectively, all in electrical and computer engineering.

In 1999, he joined the the Department of Electrical and Computer Engineering, University of Toronto, where he is currently an Assistant Professor and holds the L. Lau Junior Chair in Electrical and Computer Engineering. His research interests are in the areas of analog and digital integrated circuits, high-speed signaling, VLSI memory design (including SRAM, DRAM, and CAMs), and ferroelectric memories. He has collaborated with industry on various VLSI design projects in the past few years, including work with Nortel, Canada, in 1994, with Mosaid, Canada, since 1996, and with Fujitsu, Japan, since 1998. He is currently supervising three active research groups in the areas of ferroelectric memories, CAMs, and high-speed signaling. He has coauthored several journal and conference papers, and received two U.S. patents on CAMs in 1998 and 1999.

Dr. Sheikholeslami received the Best Professor of the Year Award in 2000 and 2002 by the popular vote of the undergraduate students in the Department of Electrical and Computer Engineering, University of Toronto. He has served on the Memory Subcommittee of the IEEE International Solid-State Circuits Conference (ISSCC) since 2001, and on the Technology Directions Subcommittee of the same conference since 2002. He presented a tutorial on ferroelectric memory design at the ISSCC 2002.