A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T Bit Cell Enabling Logic-in-Memory

Supreet Jeloka, *Student Member, IEEE*, Naveen Bharathwaj Akesh, Dennis Sylvester, *Fellow, IEEE*, and David Blaauw, *Fellow, IEEE*

***Abstract*—Conventional content addressable memory (BCAM and TCAM) uses specialized 10T/16T bit cells that are significantly larger than 6T SRAM cells. A new BCAM/TCAM is proposed that can operate with standard push-rule 6T SRAM cells, reducing array area by 2–5× and allowing reconfiguration of the SRAM as a CAM. In this way, chip area and overall capacitance can be reduced, leading to higher energy efficiency for search operations. In addition, the configurable memory can perform bit-wise logical operations: "AND" and "NOR" on two or more words stored within the array. Thus, the configurable memory with CAM and logical function capability can be used to off-load specific computational operations to the memory, improving system performance and efficiency. Using a 6T 28 nm FDSOI SRAM bit cell, the 64×64 (4 kb) BCAM achieves 370 MHz at 1 V and consumes 0.6 fJ/search/bit. A logical operation between two 64 bit words achieves 787 MHz at 1 V.**

***Index Terms*—Computation-in-memory, configurable memory, content addressable memory (CAM), reconfigurable sense amplifier, SRAM.**

I. Introduction

A CONTENT addressable memory (CAM) compares its search input data with every word stored in the memory, and returns the address location of matching words. A binary CAM (BCAM) looks for an exact match, while a ternary CAM (TCAM) can have "don't care" bits in the memory, and therefore TCAM words can match multiple search strings.

CAMs are very useful wherever a lookup table is involved. CAMs can perform a parallel search operation across multiple data and consequently boost system performance. This parallel multidata search makes CAM an indispensable component for high-associativity caches, translation look-aside buffers [1], and register-renaming [2]. Lookup tables are also the main function of IP router tables, as shown in Fig. 1, and therefore CAMs are the major component of many router chips [3], [4].

Manuscript received September 08, 2015; revised December 24, 2015; accepted December 26, 2015. Date of publication February 08, 2016; date of current version March 29, 2016. This paper was approved by Guest Editor Masato Motomura. This work was supported in part by the NSF and in part by the DARPA.

S. Jeloka, D. Sylvester, and D. Blaauw are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: sjeloka@umich.edu).

N. B. Akesh is with Oracle, Santa Clara, CA 95054 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2016.2515510

Despite CAM being an important building block, it tends to use large bit cells. The main reason is that foundries typically focus on density and power of SRAM arrays and only make push-rule bit cells for SRAMs. In addition, CAMs require highly specialized bit cells with 10 transistors for a BCAM [4], [5], or even 16 transistors for a TCAM [4], as shown in Fig. 2. Hence, in practice, nonpush-rule CAM bit cells are several times larger [5]–[8] than dense push-rule 6T SRAM [9], [10] and this results in large CAM arrays.

The main motivation for our proposed solution is to improve CAM density [11], [12]. For this, a new CAM structure is proposed that uses a traditional push-rule 6T SRAM bit cell, which results in as much as $4\times$ improvement [13] in array density over conventional CAMs. In this way, chip area and overall capacitance can be reduced, leading to higher energy efficiency for search operations.

In addition to CAM functionality, the configurable memory also provides the ability to perform bit-wise logical operations between two or more data words stored in the memory. By performing the operation within the memory array, a system using the proposed solution will be more energy efficient due to reduced data movement. Performing logical operations in memory also frees up the ALU for more involved calculations, and hence boosts performance [14]–[17]. The configurable SRAM with both CAM and logic functions can therefore be used in accelerators in both ASICs and general purpose design.

II. Conventional CAM Design

A conventional CAM is organized to have its words stored row-wise. The search string is applied in the vertical direction, which is same as the bit-lines, whereas the match lines run horizontally like the word-lines, as shown in Fig. 3. The match-line sense amplifiers (SAs) at the end of the match-lines provide the match or mismatch result for each row.

A word is said to match the search string if each bit of the word matches every bit of the search string. To accomplish the bit-wise comparison, each bit cell has a storage part and a dynamic XNOR part. The bit-wise XNORs are wire ANDed on the match lines, and the match result is obtained at the output of the SAs. In many lookup applications, multiple matches are required, but if a single address is required the results can also be priority encoded.

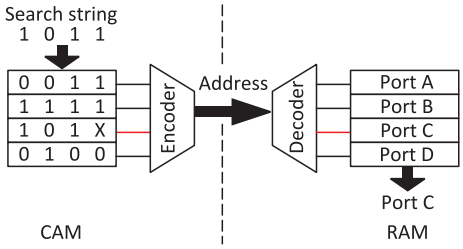

Fig. 1. CAM—a major component of IP router tables [4].

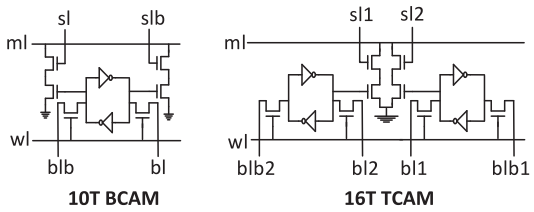

Fig. 2. Conventional bit cell design for BCAM and TCAM, respectively.

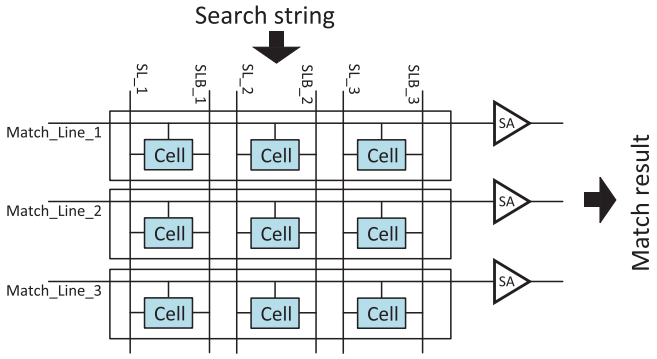

Fig. 3. Conventional CAM array organization.

As shown in Fig. 2, a conventional 10 transistor (10T) BCAM bit cell is composed of a 6T SRAM-like storage component, and a 4T XOR component to determine the bit-wise match. A TCAM can store 0, 1, or X, where "X" implies that it matches with both a "0" and a "1" of the search key. As such, it requires double the storage, resulting in a 16T cell. The high transistor count of BCAM/TCAM cells, coupled with the fact that foundries do not typically support "push-rule" CAM cells, results in a CAM array with 2–5$\times$ larger area than a corresponding SRAM; this significantly impacts chip area as well as power and performance. Certain TCAM cells are built-up using push-rule [3], [18] 8T bit cells. From the layout shown in [18], the 16T TCAM bit cell uses two 8T cells, which is estimated to be $1.35\times$ the size of the proposed TCAM composed of two 6T cells.

III. Overview of Proposed Configurable Memory Circuit

A reconfigurable CAM circuit based on a conventional, push-rule 6T SRAM bit cell that improves array density by as much as $4\times$ is proposed. The approach hinges on storing the words column-wise and using the standard bit-lines to perform a matching operation. Fig. 4 shows the configurable memory. The word-lines are reused to apply the search string in the horizontal direction, and the bit-lines are also reused to read-out the match result.

A configurability feature allows on-the-fly mode switching among BCAM, TCAM, and SRAM operation. In this way, an SRAM memory can be reconfigured to a CAM upon demand to accelerate parallel search-like applications. SRAM mode is still used conventionally with address on word-lines, words stored row-wise, and data-out on the bit-lines. As a result of using standard push-rule 6T cells, the bit density for the proposed memory array is about four times higher than other conventional BCAMs after normalizing for technology.

The configurable memory can also perform bit-wise logical operations: "AND" and "NOR" on two or more words stored row-wise within the array. Thus, the configurable memory with CAM functionality and logical function capability can be used to off-load specific computational operations to the memory in order to improve system performance and energy efficiency.

The proposed memory is energy efficient and configurable. Using a 6T 28 nm FDSOI SRAM bit cell, the 64×64 (4 kb) BCAM achieves 370 MHz at 1 V and consumes 0.6 fJ/search/bit, as shown in Fig. 18(a), energy minimum point, while the TCAM achieves the same performance at 0.74 fJ/search/bit, as shown in Fig. 19(a). A logical operation between two 64 bit words achieves 787 MHz at 1 V. Some tradeoffs are required to be made for the proposed memory configurability. The proposed memory sacrifices speed in CAM mode compared to an SRAM, for area and energy improvement over a conventional CAM. Also, the reconfigurability overhead causes this solution to be 7% larger than a conventional SRAM due to the additional peripherals.

IV. Configurable Memory: CAM Circuit Implementation

This section describes in detail how to obtain CAM operation and logic operations with SRAM bit cells. This section first describes the proposed bit cell and builds up from there. Although the proposed bit cell is 6T push-rule, to obtain the CAM operation the word-line is separated into word-line-right (WLR) and word-line-left (WLL) (Fig. 4). This creates two independent access transistors but incur no area penalty since the push-rule layers are kept intact (i.e., only DRC-compliant metallization changes are made).

The key to performing a parallel search with this bit-cell is to store words column-wise (vertically) while placing the search data on the word-lines rather than the bit-lines as in a conventional BCAM.

A. BCAM Search Operation

This section explains BCAM search with an example on a simplified 4×4 array. In Fig. 5, the search-data is applied to WLRs (the bit-line side access transistors) and search-data-bar to WLLs (the bit-line-bar side access transistors). In the match case, both BL and BLB stay at the precharged high value. If there is a mismatch, BL, BLB, or both discharge. To detect this,

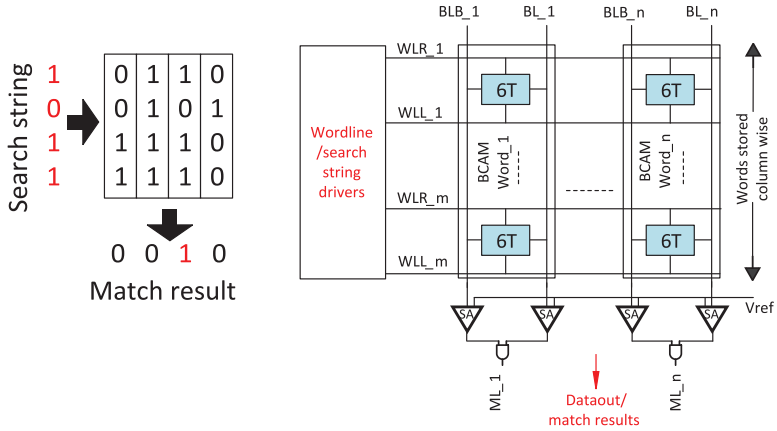

Fig. 4. Proposed CAM array organization.

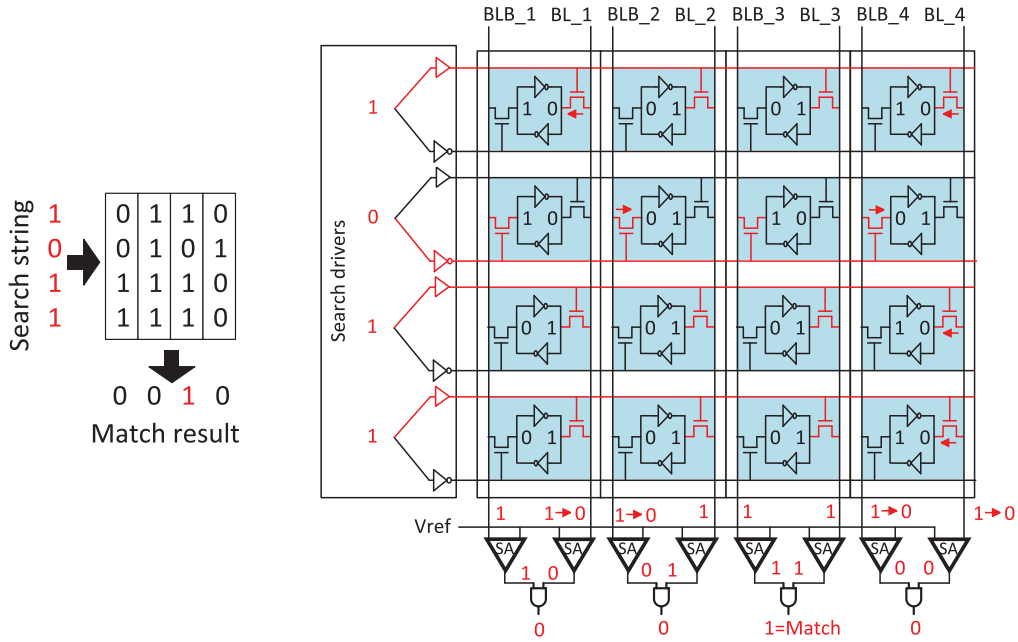

Fig. 5. BCAM search example. Only column 3 is a match. Transistors in red (gray in gray scale) are enabled.

BL and BLB are sensed separately using two single-ended SAs that are logically ANDed to indicate a match in the column.

The CAM operation will happen in parallel for all the columns of the array. The first column has a 0 in place of the 1 in the search string; therefore, it has a mismatch. As indicated by the red arrow in Fig. 5, the "0" on the top bit will start pulling the precharged bit-line down. This will make the bit-line SA to read a "0." Hence, the AND of the two SAs outputs a 0 indicating a mismatch, as expected. The timing waveform for BCAM search operation is shown in Fig. 6. The second cycle shows BL getting discharged, and hence OUT senses 0, indicating a mismatch. In the match case, both OUT and OUTB stay high.

The second column in Fig. 5 has a 1 in place of the 0 in the search string. The 0 on the second bit will start pulling the bit-line-bar down. The ANDing is a 0, indicating a mismatch.

Notice that the proposed memory always indicates the mismatch for a 1 in the search string, on the bit-line SA, whereas it indicates a mismatch for a 0 on the bit-line-bar SA. The third column is a match, as all bits are the same in the search string and the stored word. As seen in Fig. 5, all the access transistors that are enabled have a 1 on both source and drain. Therefore, both bit-line and bit-line-bar stay high and the output at the AND gate is a 1, implying a match.

The array thus performs a similar operation as a conventional BCAM. The bit-wise XNOR of the data is performed at the access transistors and they are then wire-ANDed at the bit-line SAs.

Section IV-B describes the unconventional two SAs per column which is actually designed as a single, reconfigurable amplifier.

B. Reconfigurable Sense Amplifier Design

The cross-couple of a conventional voltage differential SA is split into two parallel cross-couples, as shown in Fig. 7. During the CAM mode, it is required to individually sense both bit-line and bit-line-bar. The upper cross-couple compares bit-line-bar against a reference voltage vref, while the lower one compares

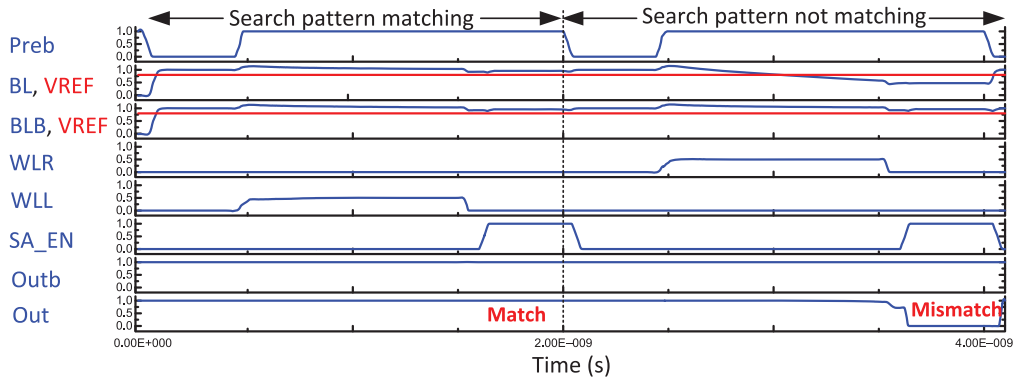

Fig. 6. Timing waveforms for BCAM search. Shows a "match" where both OUT and OUTB are high and a "mismatch" case. The waveform is for typical process corner at room temperature, with $\mathrm{Vdd} = 1$ V, vref $= 0.8$ V, and $\mathrm{Vdd\_Lo} = 0.5$ V.

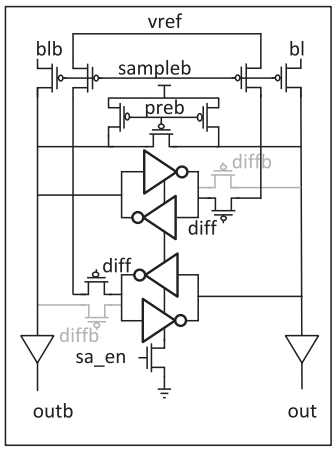

Fig. 7. Reconfigurable sense amplifier: two-single-ended amplifiers in CAM and logic modes; differential mode for SRAM.

bit-line against vref. During the SRAM mode, the faster differential mode is used between the bit-line and the bit-line-bar. In SRAM mode, both the cross-couples are tied together in parallel, effectively leading to the same strength differential SA that had been split. Hence, the two SAs per column obtained for the CAM operation are designed using the same area as that of a standard amplifier for SRAM. Fig. 8 shows the SPICE simulation waveform for the reconfigurable SA in the single-ended mode. In this figure, BLB falls below vref; therefore, OUTB senses a 0 when "SA_EN" is asserted. This reference voltage vref, used for single-ended sensing mode, is brought in as an additional supply for this chip.

C. BCAM Write Operation

One way to write the CAM is to use the SRAM mode and write the transpose of the required CAM data row-wise. But this implies doing a bulk write of CAM data, which might be acceptable for applications where the lookup table has static data while the search string changes. However, for a general CAM-based lookup, it is required to update specific data elements. To write data column-wise into the CAM, as required for parallel search, a two-cycle write scheme is proposed for BCAM mode. A column-decoder is added to select the column to be written.

To write column-wise, the data is applied to the word-lines instead of the conventional bit-lines, as shown in Fig. 9. Column 3, marked in orange, is the column-under-write. The column-wise write takes two cycles, wherein all the "1"s are written in cycle 1 and all the "0"s are written in cycle 2. In cycle 1, only the word-lines for those bit positions are enabled where a "1" has to be written. The word-lines are under-driven, and additionally, the cross-couple voltage of the column-under-write is also lowered to $\mathrm{Vdd\_Lo}$ as seen in Fig. 9 by the orange cross-couples in column 3. This allows the third column to be written even with low word-line voltages. The other columns are protected from data corruption, by keeping their cross-couple voltage high. Also the bit-line and bit-line-bar are driven strongly only for the column-under-write. Thus, the first cycle only writes all the 1s in the column-under-write.

Similarly, in the second cycle, the 0s are written. For this, data-bar is applied on the word-lines. When writing a 0, the 1s already written in the column should not be corrupted. Therefore, the $\mathrm{Vdd\_Lo}$ should not go below the retention voltage. The constraint for $\mathrm{Vdd\_Lo}$ is thus two sided—it should be less than the Vdd_disturb and more than the retention voltage. The timing waveform for BCAM write operation is shown in Fig. 10. The first cycle shows $\mathrm{Bit_x}$ (bit at row index "x" in the column-under-write) being written with a "1" followed by $\mathrm{Bit_y}$ in the same column being written with a "0" in the second cycle. While $\mathrm{Bit_y}$ is being written, $\mathrm{Bit_x}$ holds its data at $\mathrm{Vdd\_Lo}$.

In addition, if data are written in "bulk," the extra write cycle can be avoided by first writing zero into the entire array in one cycle and then only writing the "1" bits in the data to each of the columns.

D. BCAM Search Robustness

The robustness and the probability of data corruption in a BCAM are discussed in this section. Unlike the SRAM, multiple word-lines are enabled in the array for the CAM operation.

The bit cell encircled in Fig. 11 matches the search string but the data in the column as a whole does not, and hence the bit-line will discharge. As a result, this matching bit cell has a write-like condition, a pseudowrite, where the BL is falling, and the access transistor is ON. But this disturb is not very strong because of two reasons. First, the search disturb is only

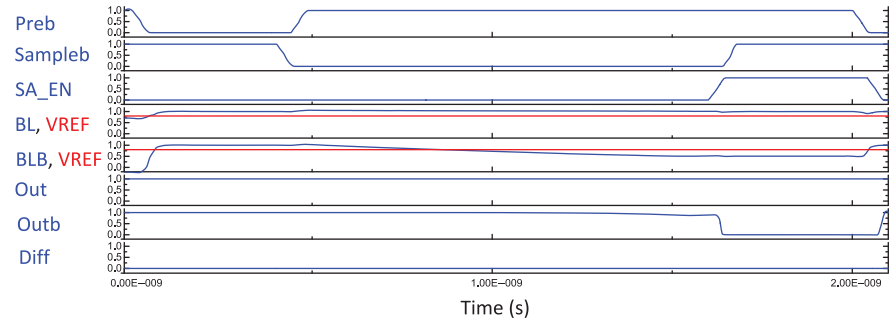

Fig. 8. Spice waveform for reconfigurable SA. "Diff" = 0, therefore, it is in two-single ended amplifier mode. The waveform is for typical process corner at room temperature, with Vdd $= 1$ V, vref $= 0.8$ V, and Vdd_Lo $= 0.5$ V.

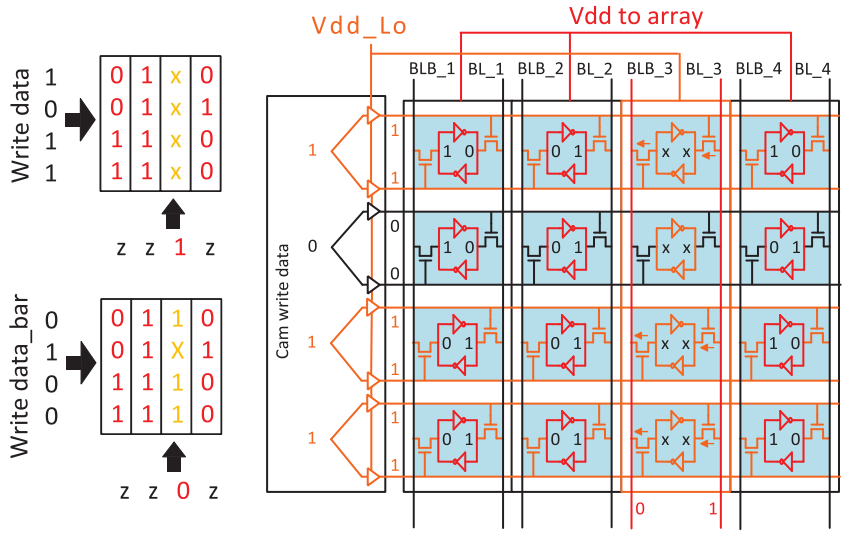

Fig. 9. BCAM column-wise write. In this example, column 3 is being written. Orange lines (light gray) are Vdd_Lo, while red (dark gray) lines are nominal Vdd.

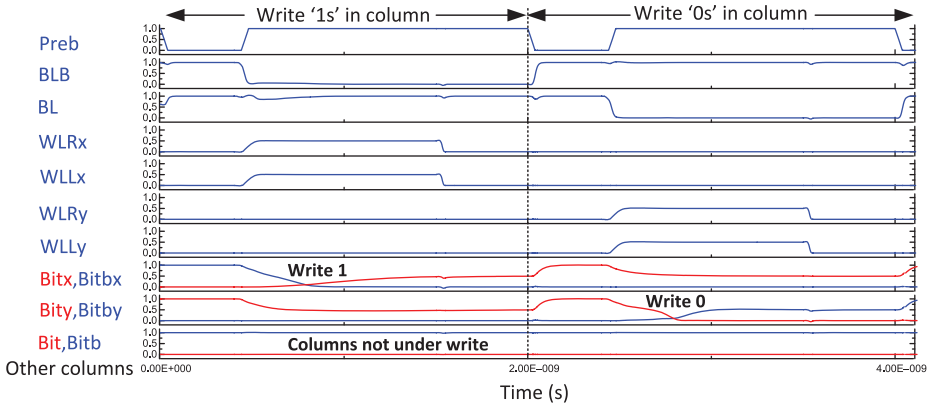

Fig. 10. Timing waveforms for two-cycle BCAM write. First cycle writes all the "1s" in the column, whereas the second cycle writes all the "0s." Notice that other columns not under write have their bit cells at full Vdd. The waveform is for typical process corner at room temperature, with Vdd $= 1$ V and Vdd_Lo $= 0.5$ V.

single-ended as just one access transistor is ON for the cell. Second, the falling bit-line is well above 0.

However, the bit-line voltage is data-dependent. A column with multiple mismatches with "1"s on the search string can have BL closer to 0. Thus, the data in the bit cell might still flip under sufficient process variation.

To solve this search disturb, it is required to weaken the access transistors, and make the storage cross-couple stronger. But for this, the layout cannot be changed, as the SRAM mode and the push-rule cell should not be affected. Therefore, a different voltage on the word-line drivers is used as an assist technique. The word-lines are under-driven, while the power lines supplying the cross-couple in the columns are kept high at Vdd.

The word-line under drive and cell boosting prevents data corruption during the search and write operations. By using Vdd_Lo for both write and search assist, only one additional

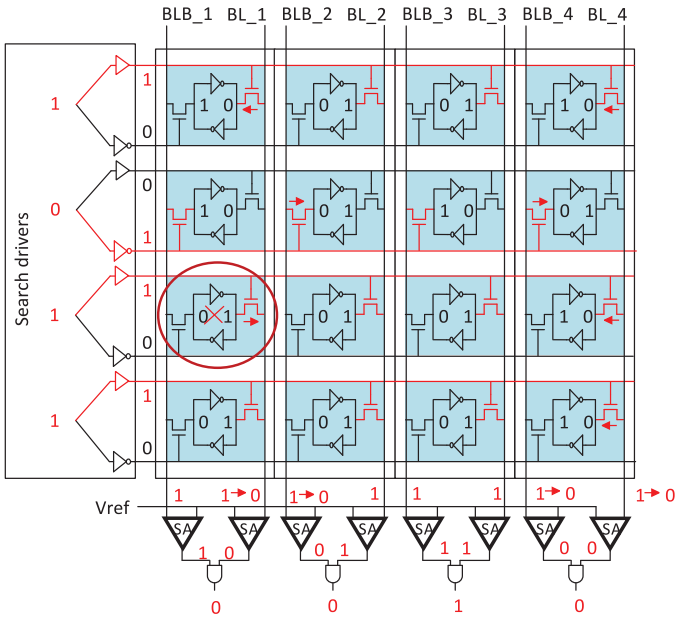

Fig. 11. BCAM search disturb: pseudowrite condition on encircled bit cell.

supply voltage is needed for the configurable memory. Fig. 12 shows a Monte Carlo analysis of the write and search disturbs. The disturb margin in write is smaller than search, as both access transistors are ON during write, but with the assist techniques used we still get a mean noise margin of 263 mV at 27 °C.

E. TCAM Mode Operation

TCAM mode will be covered in brief in this section, as it is very similar to BCAM mode in its operation. As the TCAM needs 0, 1, and don't care to be represented, it needs two bits per cell. Consequently, two columns have to be used for each word, as shown in Fig. 13, and hence the capacity is half. To represent X, "01" is used, whereas 0 and 1 are simply 00 and 11, respectively.

In TCAM read, the only difference with BCAM mode is the SAs being observed, as each word spans two columns, as can be seen in Fig. 13. In this mode, two of the four SA outputs that span the two columns constituting a word are ANDed together. A mask bit "X" will not discharge either sensed bit-line or bit-line-bar as it stores a "1" in both positions. Hence, it matches with both 0 and 1 of the search data. In the example in Fig. 13, the top-right bit enclosed in the red box is masked; hence the second word matches "1011." By virtue of the mask bit, the second word would also have matched the search string "0011."

TCAM write is similar to BCAM but takes three cycles. The first two cycles are similar to BCAM, as first "11" is written and then "00" is written. The mask bits "01" are then written in the third cycle by only enabling the word-lines of rows which need to be masked. The adjacent cells are written with 01, by applying the appropriate voltage levels at the bit-lines. This has been shown conceptually in Fig. 13. In Fig. 14, we show the TCAM write operation's timing waveform. $\mathrm{Bit_x}$ and $\mathrm{Bit_{x+1}}$ is written with "11," whereas $\mathrm{Bit_y}$ and $\mathrm{Bit_{y+1}}$ are written with "'00." To write mask in $\mathrm{column\text{-}word_i}$ in $\mathrm{row_z}$, $\mathrm{Bit_z}$ is written as "0" and $\mathrm{Bit_{z+1}}$ as "1" as shown in this figure. Since write is less common in many CAM applications than search, the additional cycles pose less overhead.

V. SRAM Mode and Logic Operations in Memory

A. SRAM Mode Operation

In SRAM mode, the configurable memory works conventionally with both WLR and WLL driven from the address-decoder output. In SRAM mode, reads and writes proceed row-wise using conventional differential signaling and the performance impact from reconfigurability is found to be negligible. By reconfiguring the two single-ended SAs in CAM mode into a single differential SA in SRAM mode, total reconfiguration area overhead is limited to only 7% for the added column decoder.

B. Logic Operations in Memory

The configurable memory can be used to perform certain logical operations between the row-wise stored SRAM words. These logic operations are enabled by reutilizing the circuits used in the CAM modes.

Logic-in-memory here is defined as the feature of performing logical operations within the memory subarray itself, without having to read-out or sense the individual words being operated upon. The term logic-in-memory has been used before in other contexts, such as using memory technology other than CMOS to realize logic on a dedicated memory layer [19], or dedicated logic layer in 3-D DRAMs [20], or logic in the main memory [15], but not within the subarray. The main difference lies in not sensing the individual operands to perform the logic.

Fig. 15 shows an example of a bit-wise "AND" operation between two rows in the array. To perform the "AND" operation, the memory is put in the BCAM search mode. In the BCAM search mode, an input bit of the search string can be masked (denoted by "M" in Fig. 15) by applying a "0" to both the WLR and the WLL. This feature allows the word-lines of two or more rows to be enabled.

In the example shown in Fig. 15, the search string (1, M, 1, M) is applied, which only activates WLR for rows 1 and 3. If any bit in row 1 or row 3 is "0," it will pull-down the precharged bit-line. As all, WLL transistors are disabled, all bit-line-bar lines stay high. Hence, the bit-wise AND of rows 1 and 3 is obtained at the memory output.

More than two words can also be activated by putting more 1s in the search string. The bit-wise "AND" operation can thus be executed for two or more than two words. Table I shows all the logic operations supported by the proposed configurable memory. Similar to the "AND" operation, a NOR operation can be performed by only activating the WLL access transistor and by applying 0s at the search string. A "01" combination activates WLL for row A and WLR for row B, hence it senses the complement of the data in row A on the bit-line-bar SAs, and simultaneously senses the data in row B on the bit-line SAs. These two are then ANDed to produce the result. "10" has the same operation as "'01," but changes the location of

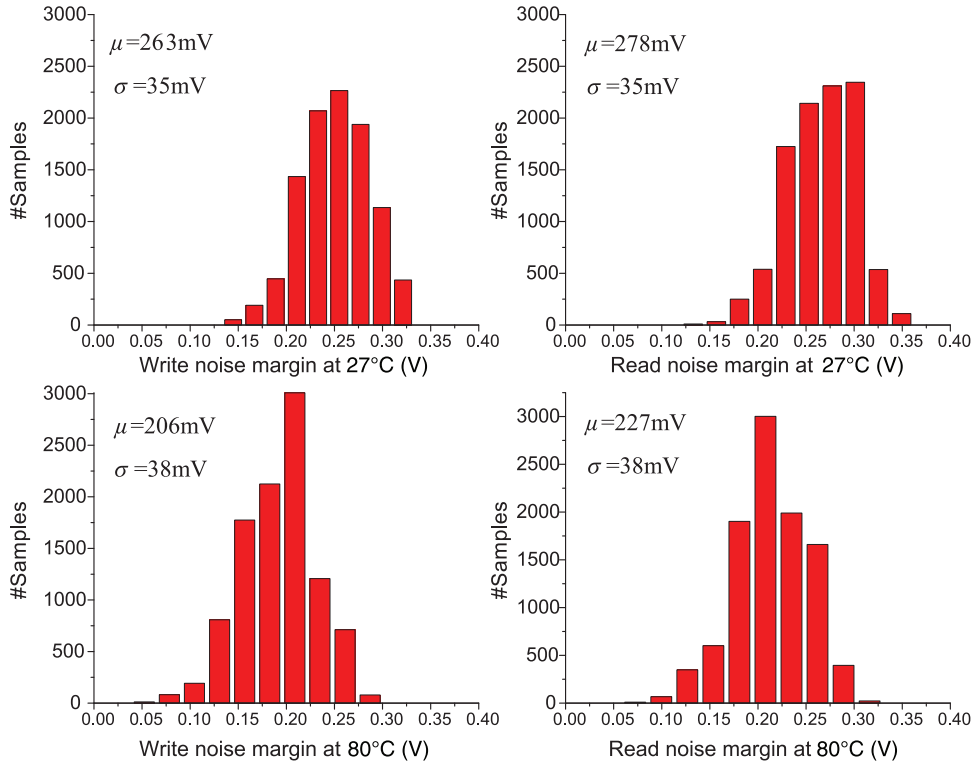

Fig. 12. Monte Carlo simulations for write and search (read) robustness in CAM modes. Monte Carlo simulations performed with global + local mismatch, at both 27°C and 80°C with Vdd = 1 V and Vdd_Lo = 0.5 V.

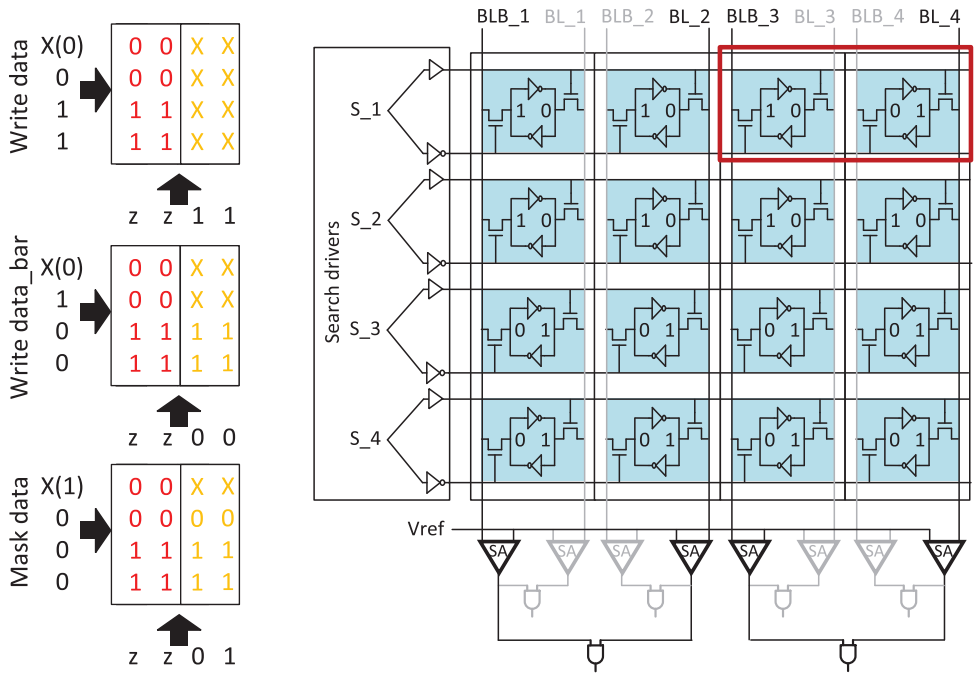

Fig. 13. TCAM mode organization. Two columns comprise a word. The bit cell in the top-right red box is "masked."

the rows activated. A "01" like operation allows two rows to be read out simultaneously, as the configurable memory has two single-ended SAs. Thus, this feature can also be used as a dual read port, where "A_bar" is read on the bit-line-bar SAs and "B" is read on the bit-line SAs.

Similar to the BCAM search robustness issue discussed in Section IV-D above, logic operations also activate multiple word-lines. The BCAM search activates all the word-lines, and hence the probability of data corruption is higher. To prevent data corruption in BCAM mode, Vdd_Lo has to be reduced significantly. During a logic operation on two words, only two bits are fighting in any column. This allows Vdd_Lo to rise significantly; hence the logic mode for two words can run much faster than the BCAM mode. For multi-word logic operations, the

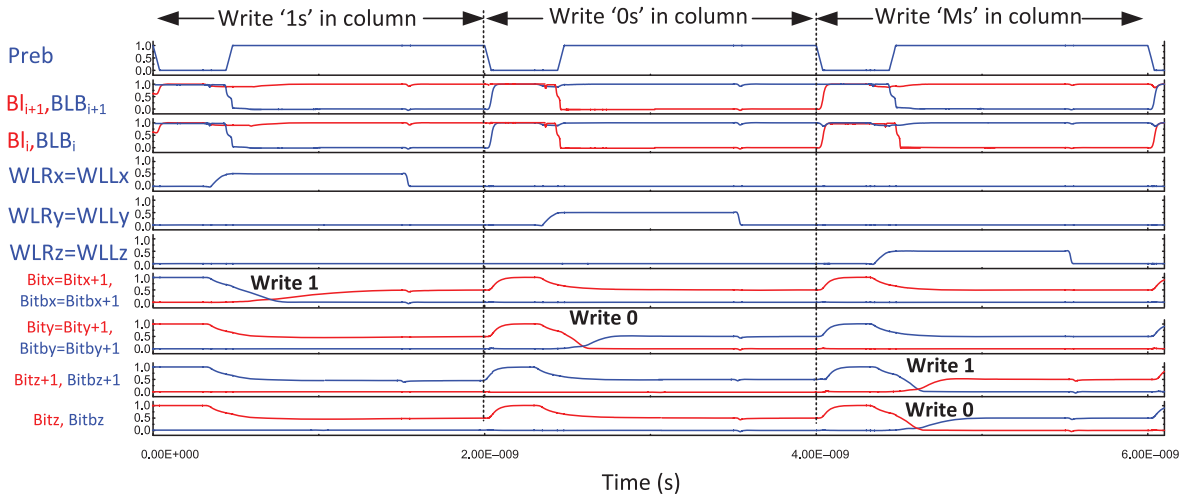

Fig. 14. Timing waveforms for three-cycle TCAM write. First cycle writes all the "11" in the column, the second cycle writes all the "00." The third cycle writes a "01" in the adjacent cells of the bit to be masked. The waveform is for typical process corner at room temperature, with Vdd = 1 V and Vdd_Lo = 0.5 V.

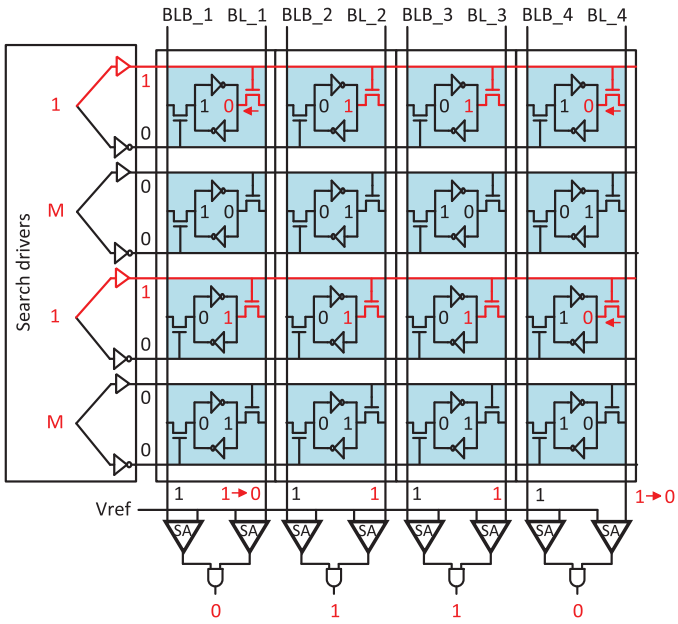

Fig. 15. Logic operations in memory. Enabled two rows (rows 1, 3 while rows 2, 4 are masked, i.e., disabled), to get an "AND" between data stored in the two enabled rows. The logic operations reuse the search circuit for BCAM mode.

TABLE I
LOGIC OPERATIONS IN MEMORY

Search string	Logic in memory	Multi word operation possible
11 (other rows masked)	A AND B	Yes (A&B&C…)
00 (other rows masked)	A NOR B	Yes (NOT(A|B|C…))
01 (other rows masked)	A_Bar AND B	No
10 (other rows masked)	A AND B_BAR	No

Only the words to be operated upon get the search string value while the other words have WLR=WLL=0, i.e., they are masked.

Vdd_Lo reduces with the increase in the number of words that are simultaneously operated upon, and consequently the frequency of operation reduces. If an "AND" or "NOR" operation is performed on all the rows in the array, operation approaches the BCAM frequency and Vdd_Lo value.

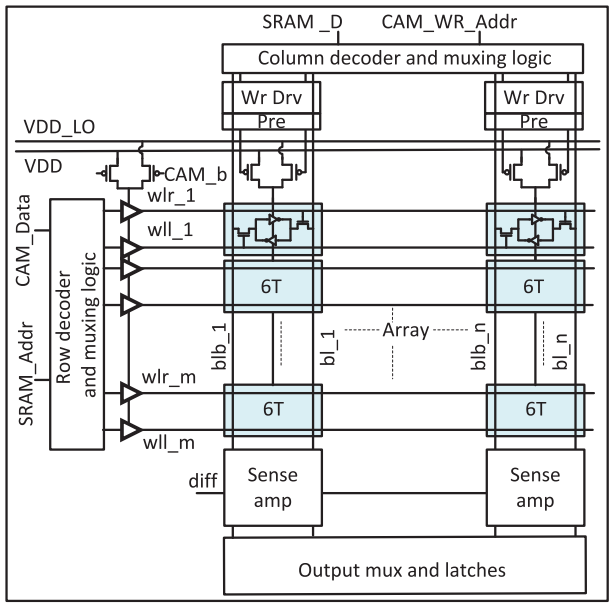

Fig. 16. Configurable memory organization.

Fig. 16 is the overall block diagram of the reconfigurable memory. Notice the additional column decoder at the top. This ensures that only the column-under-write is supplied by low Vdd. The column decoder output also controls the enable of the write drivers. A common header switch is placed for word-line drivers to switch between Vdd and Vdd_Lo, as shown in Fig. 16. Also, most 6T SRAMs at advanced technology nodes, need some type of read/write assist techniques. One common technique used is WL under-drive/over-drive. If this were the case, these assist switches are reutilized for CAM WL under-drive. Table II summarizes the memory driver and SA configuration during different memory modes.

VI. TEST HARNESS

The configurable memory is validated in a 28 nm FDSOI CMOS test chip. An on-chip built-in self-test (BIST) is used

TABLE II
CONFIGURABLE MEMORY—MODE CONFIGURATION TABLE

	SRAM		BCAM			TCAM				Logic-in-memory
	Read	Write	Search	WrCycle 1	WrCycle 2	Search	WriteCycle 1	WriteCycle 2	WriteCycle 3	Read only write using SRAM
wll	Row decoder output		Dbar	D	Dbar	Dbar	D	Dbar	Mask	Dbar and Mask
wlr			D			D				D and Mask
bl_i	Pre/ SA	D_i	Pre/ SA	1	0	Pre/ SA	1	0	0	Pre/ SA
blb_i		$Dbar_i$		0	1		0	1	1	
bl_{i+1}		D_{i+1}		Hi-z	Hi-z		1	0	1	
blb_{i+1}		$Dbar_{i+1}$		Hi-z	Hi-z		0	1	0	
Sense Amp	Differential	–	Two-single ended	–		Two-single ended	–			Two-single ended
Output	SA o/p	–	SA_bl_i and SA_blb_i	–		SA_bl_{i+1} and SA_blb_i	–			SA_bl_i and SA_blb_i

to test the different modes. March test was applied to test the bit cells for fault models like stuck-at, transition, coupling and neighborhood pattern sensitive faults. As 6T bit cells are being used even for CAM, the SRAM tests are able to cover most of the fault models.

For testing the CAM search functionality, walk mode search patterns are used. The most critical search pattern is differentiating between match case and a single-mismatch case. The walk mode uses a search pattern that negates one bit position every cycle. This pattern is able to test a 1 bit mismatch condition for every bit in every column for at-speed functionality. In addition, read disturb faults in CAM modes are tested by creating the worst case column data and corresponding search patterns. The worst disturb scenario is when all bits mismatch except one bit, on the same bit-line. Again we use walk modes to test each bit for worst disturb.

Interleaved write and search operations are used to test the correct at-speed column-wise write for CAM modes. The BIST is able to modify the expected data pattern in a walking mode. In addition, checkerboard patterns are run on the 4 kb array with 64 bit words to check for column-wise write and then read back using SRAM mode reads. Arbitrary data can be searched at-speed using an on-chip FIFO buffer.

VII. MEASURED RESULTS

The configurable memory has been designed in a 28 nm FDSOI CMOS process. Fig. 17 shows the die photo. The dimension of the memory array is 64×64, to make a 4 kb array. The BCAM read and write disturbs are sensitive to column length, and were verified on this chip for a contiguous column length of 64 bits. The word length can be extended further by having multiple banks with an AND tree of match results. With an array area of 724 μm^2 for 4 kb, the bit density of array is $\sim 5.4\ Mb/mm^2$.

For BCAM, both the measured frequency and energy are a function of Vdd and Vdd_Lo. It is found that Vdd_Lo close

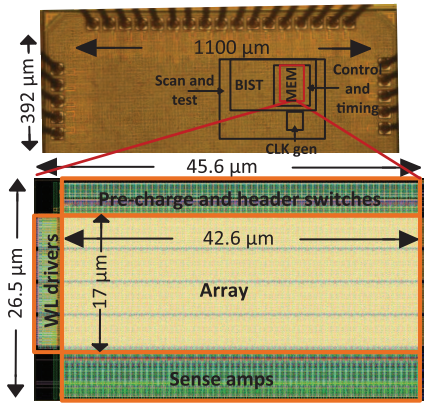

Fig. 17. Die photo and memory layout.

to Vdd-divided-by-2 works well. Vdd_Lo is brought in as an additional supply for this chip. On the *x*-axis of the graph in Fig. 18(b), Vdd and Vdd_Lo are swept, keeping a ratio of 0.5 between them. The black curve shows that the frequency increases with voltage, as expected. At Vdd = 1 V, a maximum frequency of 400 MHz is achieved. The blue energy curve is a bit more complex, as it also depends on the frequency.

To gain more insight into this, Vdd is kept fixed to 1 V nominal and only Vdd_Lo is swept, as shown in Fig. 18(a). As can be seen from this figure, the frequency of BCAM operation is a strong function of Vdd_Lo. Also the energy has a sweet spot. It decreases with voltage up to a certain point, before starting to increase again due to the frequency falling, which incurs higher leakage energy. This energy optimum for BCAM at Vdd = 1 V is measured to be 0.6 fJ per search per bit. The minimum energy point is 0.41 fJ, with a frequency of 70 MHz at Vdd = 0.7 V and Vdd_Lo = 0.375 V.

The TCAM frequency and energy have a similar trend as BCAM. In TCAM mode, the maximum frequency is 417 MHz and the optimum energy is measured to be 0.74 fJ per search per bit, as seen in Fig. 19(a). The energy consumption per bit

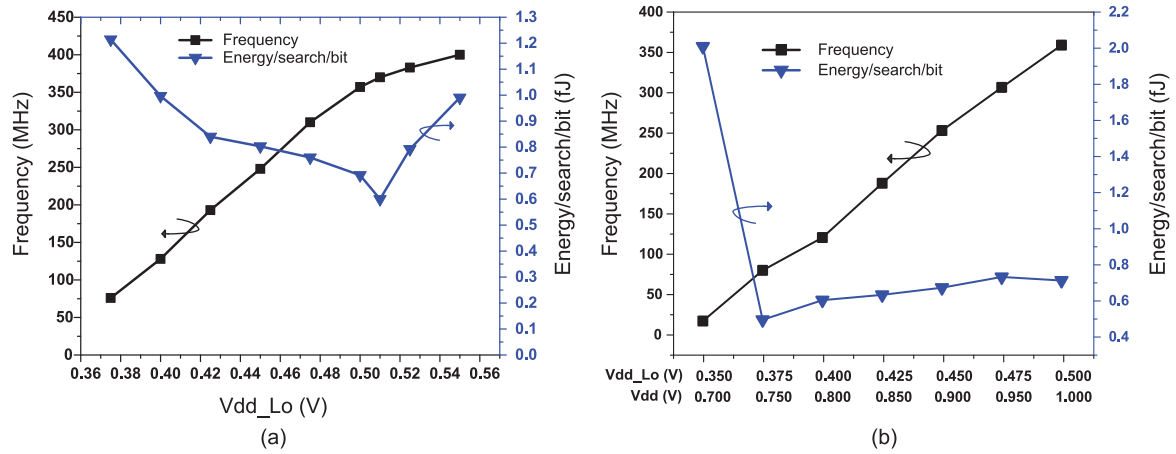

Fig. 18. (a) Measured frequency and energy in BCAM mode against Vdd_Lo, with Vdd = 1 V. (b) Measured frequency and energy in BCAM mode against Vdd, with Vdd_Lo = 0.5*Vdd. All measurements taken at room temperature.

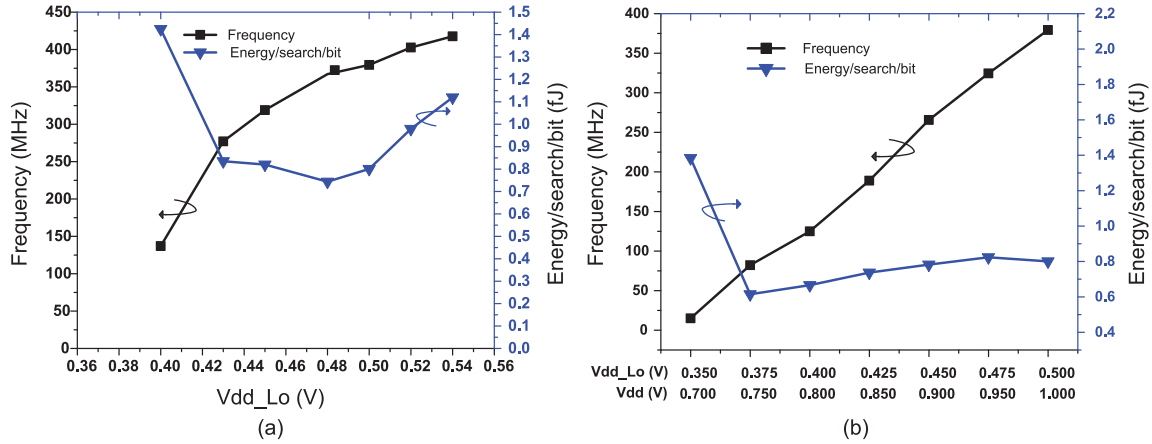

Fig. 19. (a) Measured frequency and energy in TCAM mode against Vdd_Lo, with Vdd = 1 V. (b) Measured frequency and energy in TCAM mode against Vdd, with Vdd_Lo = 0.5*Vdd. All measurements taken at room temperature.

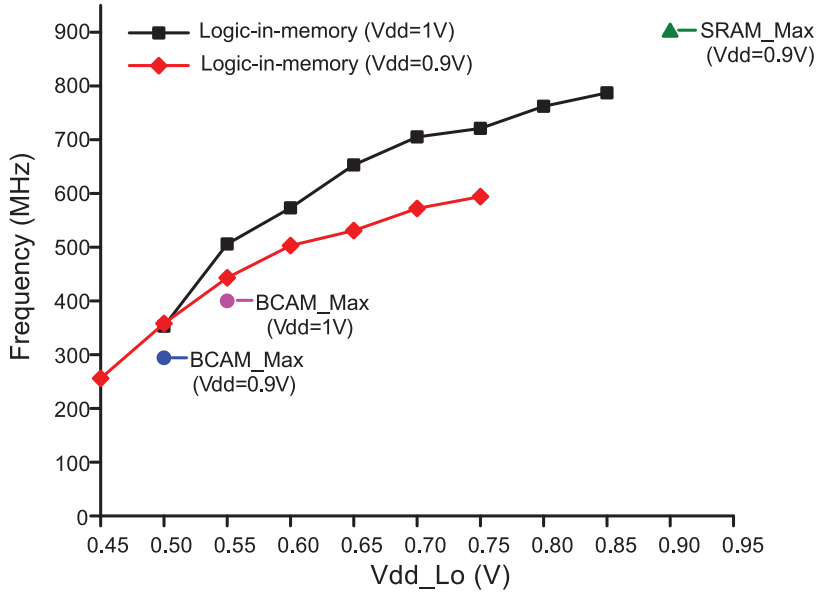

Fig. 22. Measured frequency for logic operation between two words in memory against Vdd_Lo, with fixed Vdd (Vdd = 0.9 V and Vdd = 1 V) at room temperature.

TABLE III
COMPARISON WITH PREVIOUS BCAM WORKS

	This work	Agarwal [1]	Do [5]	Wang [6]	Yang [7]
Technology	**28 nm FDSOI**	32nm	65nm	0.13µm	0.18µm
Transistor/cell	**6T**	11T	10T	9T+Read	13T,14T
Area/cell(µm²)(F²)	**0.152 (194)**	n.a.	3.3 (780)*	20 (1200)	30 (926)*
Energy/search /bit (fJ)	**0.6 (1V) 0.41 (0.75V)**	1.07 (1V) 0.3 (0.5V)	0.77 (1.2V)	1.87 (1V)	2.82 (1.8V)
Frequency (MHz)	**370**	n.a.	500	250	210
Array size	**64×64**	(64×64) ×4 arrays	128×128	128×32	128× 34×4 block
Match-line technique	**2-single ended SA**	Wide AND	NOR	Differential	NAND-NOR
Memory modes	**BCAM / TCAM / SRAM / Logic**	BCAM	BCAM	BCAM	TCAM/ BCAM

*From die-photo.

TABLE IV
COMPARISON WITH PREVIOUS TCAM WORKS

	This work	Arsovski [21]	Nii [3]	Hayashi [18]	Huang [22]
Technology	**28 nm FDSOI**	32nm SOI	28nm	65nm	65nm
Transistor/cell	**12T**	16T	16T	16T	14T
Area/cell(µm²)(F²)	**0.304 (388)**	n.a.	0.625 (797)*	1.69 (400)**	7.05 (1669)
Energy/search /bit (fJ)	**0.74 (1V) 0.61 (0.75V)**	0.58	n.a.	1.98	0.34–0.16 (based on % don't care)
Frequency (MHz)	**370**	1000	400	250	400
Array size	**32×64 (2columns/word)**	128×128(max) variable	(4k × 80) ×64 ×4	(2k × 72) ×32 ×4	256×144
Match-line technique	**2-single ended SA**	Early predict late correct	Differential+ valid bit	Differential low voltage	Butterfly match-line
Memory modes	**BCAM / TCAM / SRAM / Logic**	TCAM	TCAM	TCAM	TCAM

* Bit cell area calculated from density, assuming array efficiency of 40%. Reference [3] cites [18] as its previous work. The array efficiency for [18] is 43%. From details in [3] and [18], we conservatively estimate [3] to have an array efficiency of $\sim 40\%$.

**Scaling trend of push-rule SRAM according to ISSCC trends [23]— $124f^2$ at 65 nm but at 28 nm it is $162f^2$. Also, from the layout figure in [18] the bit cell uses two 6T cells plus the additional 4 transistors. From the layout figure shown, this bit cell is estimated to be 1.35× the size of the proposed TCAM bit cell (two 6T cells).

in TCAM is higher as the total number of bits is half, but in TCAM only half the sense-amplifiers and output latches are used. The minimum energy point is 0.61 fJ, with a frequency of 116 MHz at $\mathrm{Vdd} = 0.75$ V and $\mathrm{Vdd\_Lo} = 0.375$ V, as seen in Fig. 19(b).

In Fig. 20, a shmoo plot is shown with Vdd on the *x*-axis and $\mathrm{Vdd\_Lo}$ on the *y*-axis. As discussed earlier, $\mathrm{Vdd\_Lo}$ has a two-sided constraint. The red tiles are voltage pairs where the BCAM fails, whereas the numbers in the passing green tiles are the operating frequency. If $\mathrm{Vdd\_Lo}$ is high, speed is better, but as the access transistor becomes stronger, the probability of disturb goes up, and hence failures start to be seen in the upper left triangle. The failures below 0.325 $\mathrm{Vdd\_Lo}$ are due to the SA read resolution, i.e., the design cannot reliably resolve the single-mismatch case for every column.

Fig. 21 shows the $\mathrm{Vdd\_Lo}$ operational voltage margin distribution across multiple chips. $\mathrm{Vdd\_Lo\_margin}$ is the voltage range of $\mathrm{Vdd\_Lo}$ over which the CAM is functional. At nominal Vdd, the $\mathrm{Vdd\_Lo\_margin}$ across 10 chips has a mean of 180 mV as shown in the bar graph on the left. Thus, a reasonable margin for CAM operations is available. The mean of the max frequency in BCAM mode across chips is about 365 MHz as shown on the right.

The max frequency in SRAM mode is about 900 MHz at 0.9 V, as it is not affected by $\mathrm{Vdd\_Lo}$ because the word-lines are driven to nominal voltage in SRAM mode. Fig. 22 shows the

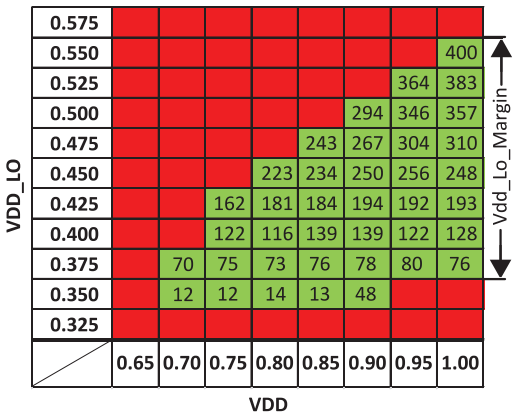

Fig. 20. Measured shmoo plot of Vdd_Lo versus VDD for BCAM. Numbers in box are frequency in MHz. All measurements taken at room temperature.

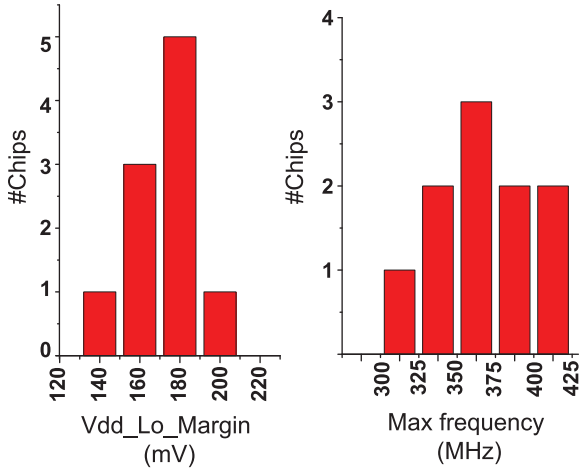

Fig. 21. Measured Vdd_Lo margin and max frequency across 10 chips. The histograms are for following operating conditions: $\mathrm{Vdd} = 1$ V and $\mathrm{Vdd\_Lo} = 0.5$ V at room temperature.

operational frequency for logic operation between two words stored row-wise in the memory. The logic in memory mode is similar to BCAM search operation and hence its frequency is also a function of both Vdd and Vdd_Lo. On the *x*-axis of the graph in Fig. 22, Vdd_Lo is swept, keeping Vdd fixed. As explained in Section V-B, the search disturb is less pronounced in the logic mode than in the BCAM mode where all rows are activated. For two-word logic mode at $\mathrm{Vdd} = 1$ V, the Vdd_Lo can be increased to 0.85 V, allowing it to achieve a maximum frequency of 787 MHz as compared to the BCAM's 400 MHz. Compared to the 900 MHz for SRAM mode at $\mathrm{Vdd} = \mathrm{Vdd\_Lo} = 0.9$ V, the logic mode achieves a maximum frequency of 594 MHz at $\mathrm{Vdd} = 0.9$ V and $\mathrm{Vdd\_Lo} = 0.75$ V. The frequency loss in logic mode compared to the SRAM mode is because of, first, lower word-line voltage and, second, slower SA as logic mode uses single-ended sensing mode.

In Table III, our design is compared against other more conventional BCAMs, while in Table IV, we compare against other conventional TCAMs. All the conventional BCAMs have a higher transistor count in their bit cells. If the area normalized for technology in F^2 (F being feature size) is compared, the gain is by more than $4\times$. Even the push-rule TCAM bit-cell is $1.35\times$ larger than our proposed TCAM bit cell. The energy efficiency achieved is good at 0.41 fJ/search/bit at 0.75 V for BCAM. Also, configurability is possible between different operating modes.

VIII. Conclusion

A configurable memory with CAM functionality using standard push-rule SRAM 6T bit cells is presented. This memory can be used as an area-energy efficient CAM in search-based applications. It also has lower instantaneous power because of low voltage word-line drive. The memory can also be used to perform certain logic operations between two or more rows. This can be used to off-load computations to the memory, improving system performance.

The proposed configurable memory with logic-in-memory has an energy efficiency of 0.6 fJ/search/bit at 1 V with an array bit density of $\sim 5.4\ \mathrm{Mb/mm^2}$ which is a $4\times$ improvement in array density over conventional BCAMs. This is achieved with only 7% area overhead for configurability over a conventional SRAM. The logic-in-memory operations between two 64 bit words in the configurable memory, achieves 787 MHz at 1 V.

Acknowledgment

The authors would like to thank the STMicroelectronics for IC fabrication.

References

[1] A. Agarwal *et al.*, "A 128×128b high-speed wide-and match-line content addressable memory in 32 nm CMOS," in *Proc. ESSCIRC*, 2011, pp. 83–86.

[2] G. Burda, Y. Kolla, J. Dieffenderfer, and F. Hamdan, "A 45 nm CMOS 13-port 64-word 41b fully associative content-addressable register file," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2010, pp. 286–287.

[3] K. Nii *et al.*, "A 28 nm 400 MHz 4-parallel 1.6 Gsearch/s 80Mb ternary CAM," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 240–241.

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

[5] A. T. Do, C. Yin, K. S. Yeo, and T. T. H. Kim, "Design of a power-efficient CAM using automated background checking scheme for small match line swing," in *Proc. ESSCIRC*, 2013, pp. 209–212.

[6] C.-C. Wang, C.-H. Hsu, C.-C. Huang, and J.-H. Wu, "A self-disabled sensing technique for content-addressable memories," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 57, no. 1, pp. 31–35, Jan. 2010.

[7] B.-D. Yang, Y.-K. Lee, S.-W. Sung, J.-J. Min, J.-M. Oh, and H.-J. Kang, "A low power content addressable memory using low swing search lines," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 58, no. 12, pp. 2849–2858, Dec. 2011.

[8] C.-C. Wang, J.-S. Wang, and C. Yeh, "High-speed and low-power design techniques for TCAM macros," *IEEE J. Solid State Circuits*, vol. 43, no. 2, pp. 530–540, Feb. 2008.

[9] R. Ranica *et al.*, "FDSOI process/design full solutions for ultra low leakage, high speed and low voltage SRAMs," in *Proc. Symp. VLSI Technol. (VLSIT'13)*, 2013, pp. T210–T211.

[10] E. Karl *et al.*, "A 0.6 V 1.5 GHz 84Mb SRAM design in 14 nm FinFET CMOS technology," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2015, pp. 1–3.

[11] J. Li, R. K. Montoye, M. Ishii, and L. Chang, "1 Mb 0.41 μm^2 2T-2R cell nonvolatile TCAM with two-bit encoding and clocked self-referenced sensing," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 896–907, Apr. 2014.

[12] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.
[13] S. Jeloka, N. Akesh, D. Sylvester, and D. Blaauw, "A configurable TCAM/BCAM/SRAM using 28 nm push-rule 6T bit cell," in *Proc. Symp. VLSI Circuits (VLSIC'15)*, 2015, pp. C272–C273.
[14] P. Jain, G. E. Suh, and S. Devadas, "Embedded intelligent SRAM," in *Proc. 40th Ann. Des. Autom. Conf.*, 2003, pp. 869–874.
[15] D. Patterson *et al.*, "A case for intelligent DRAM: IRAM," *IEEE Micro*, vol. 17, no. 2, pp. 33–44, Apr. 1997.
[16] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. J. Dally, and M. Horowitz, "Smart memories: A modular reconfigurable architecture," in *Proc. 27th Int. Symp. Comput. Archit. (ISCA'00)*, 2000, pp. 161–171.
[17] K. Mai *et al.*, "Architecture and circuit techniques for a reconfigurable memory block," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2004, pp. 500–501.
[18] I. Hayashi *et al.*, "A 250-MHz 18-Mb full ternary CAM with low-voltage matchline sensing scheme in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2671–2680, Nov. 2013.
[19] S. Matsunaga *et al.*, "MTJ-based nonvolatile logic-in-memory circuit, future prospects and issues," in *Proc. Conf. Des. Autom. Test Eur. (DATE'09)*, 2009, pp. 433–435.
[20] Q. Zhu *et al.*, "A 3D-stacked logic-in-memory accelerator for application-specific data intensive computing," in *Proc. IEEE 3D Syst. Integr. Conf. (3DIC'13)*, 2013, pp. 1–7.
[21] I. Arsovski, T. Hebig, D. Dobson, and R. Wisort, "A 32 nm 0.58-fJ/bit/search 1-GHz ternary content addressable memory compiler using silicon-aware early-predict late-correct sensing with embedded deep-trench capacitor noise mitigation," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 932–939, Apr. 2013.
[22] P.-T. Huang and W. Hwang, "A 65 nm 0.165 fJ/bit/search 256 144 TCAM macro design for IPv6 lookup tables," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 507–519, Feb. 2011.
[23] S. G. Narendra, L. C. Fujino, and K. Smith, "Through the looking glass? The 2015 edition: Trends in solid-state circuits from ISSCC," *IEEE Solid-State Circuits Mag.*, vol. 7, no. 1, pp. 14–24, Feb. 2015.

Supreet Jeloka (S'15) received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology (NIT), Warangal, India, in 2007, and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2013, where he is currently pursuing the Ph.D. degree.

Prior to joining the University of Michigan, he worked for Freescale, India, as a Senior Design Engineer. His research interests include low-power circuits, memory design, memory-based computing, interconnect fabrics, and hardware security.

Naveen Bharathwaj Akesh received the B.E. degree in electronics and communication engineering from Anna University, Chennai, India, in 2012, and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2014.

He is currently a Hardware Engineer with the Oracle America Inc., Santa Clara, CA, USA. His research interests include energy efficient design for high-performance systems and variability tolerant circuit design.

Dennis Sylvester (S'95–M'00–SM'04–F'11) received the Ph.D. degree in electrical engineering from the University of California, Berkeley, CA, USA, in 1999.

He is a Professor of Electrical Engineering and Computer Science with the University of Michigan, Ann Arbor, MI, USA, and the Director of the Michigan Integrated Circuits Laboratory (MICL), a group of 10 faculty and more than 70 graduate students. He has held Research Staff positions with the Advanced Technology Group of Synopsys, Mountain View, CA, USA; Hewlett-Packard Laboratories, Palo Alto, CA, USA; and Visiting Professorships at the National University of Singapore and Nanyang Technological University, Singapore. He has authored over 375 papers along with one book and several book chapters. He holds 20 U.S. patents. His research interests include the design of millimeter-scale computing systems and energy efficient near-threshold computing.

Dr. Sylvester serves as a Consultant and Technical Advisory Board Member for electronic design automation and semiconductor firms. He Cofounded Ambiq Micro, a fabless semiconductor company developing ultra-low power mixed-signal solutions for compact wireless devices. He also serves on the Technical Program Committee of the IEEE International Solid-State Circuits Conference and previously served on the executive committee of the ACM/IEEE Design Automation Conference. He has served as an Associate Editor for the IEEE TRANSACTIONS ON CAD and the IEEE TRANSACTIONS ON VLSI SYSTEMS and the Guest Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II. He was the recipient of the David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley EECS Department, an NSF CAREER Award, the Beatrice Winner Award at ISSCC, an IBM Faculty Award, an SRC Inventor Recognition Award, and eight Best Paper Awards and nominations. He was also the recipient of the ACM SIGDA Outstanding New Faculty Award and the University of Michigan Henry Russel Award for distinguished scholarship.

David Blaauw (M'94–SM'07–F'12) received the B.S. degree in physics and computer science from Duke University, Durham, NC, USA, in 1986, and the Ph.D. degree in computer science from the University of Illinois, Urbana, IL, USA, in 1991.

Then, he worked with Motorola, Inc., Austin, TX, USA, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been on the faculty with the University of Michigan, Ann Arbor, MI, USA, where he is a Professor. He has authored over 450 papers and holds 40 patents. His research interests include VLSI design with particular emphasis on ultra low power and high performance design.

Dr. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronic and Design. He was also the Technical Program Co-Chair of the ACM/IEEE Design Automation Conference and a member of the ISSCC Technical Program Committee.