ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Design of Content Addressable Memory (BCAM/TCAM) using Push-Rule 6T SARAM Cell

N.Venkatesh
M.Tech,
Dept of ECE (VLSI),
BVC College of Engineering,
Rajahmundry- 533294. A.P.

Mr G.Sravan Kumar, M.Tech
Associate Professor,
Dept of ECE,
BVC College of Engineering,
Rajahmundry- 533294. A.P.

Abstract:

Content addressable memory (CAM) is a special type of memory which can do search operation in a single clock cycle. CAM is the hardware for parallel lookup/search. The parallel search scheme promises a high-speed search operation but at the cost of high power consumption. 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. In this way, chip area and overall power consumption 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. The proposed novel 6T SRAM based BCAM/TCAM memory can be designed on DSCH and Microwind 3.5 using 65nm rule.

Index Terms:

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

I. INTRODUCTION:

Software-based search algorithms are widely used. When used in the applications requiring high-speed search, e.g., IP routing, image processing, data compression, data management, and so on [1]–[4], they are slow and reduce the system speed. Thus, software-based search algorithms are unsuitable for high-speed applications. Content addressable memory (CAM) features hardware-based parallel search operation suitable for high-speed applications. Like other memories, CAM stores the data in its data banks. CAM feeds the search data, performs the search, and outputs the match address [5] if any. 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].

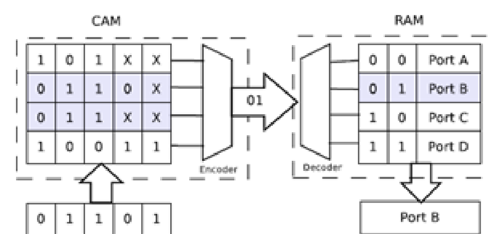

Fig. 1. CAM—a major component of IP router tables [4].

Despite CAM being an important building block, it tends to use large bit cells.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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.

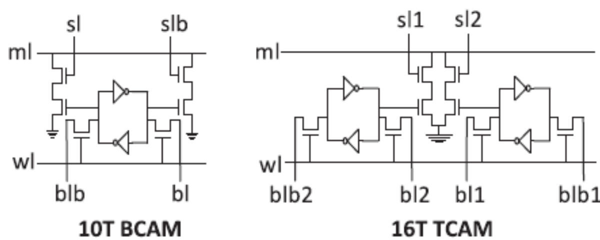

Fig. 2. Conventional bit cell design for BCAM and TCAM, respectively.

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× 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. LITERATURE SURVEY:

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.

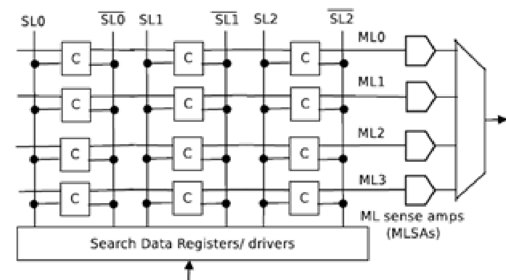

Fig. 3. Conventional CAM array organization.

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× 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× the size of the proposed TCAM composed of two 6T cells.

III. PROPOSED CAM MEMORIES:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A reconfigurable CAM circuit based on a conventional, pushrule 6T SRAM bit cell that improves array density by as much as 4× 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.

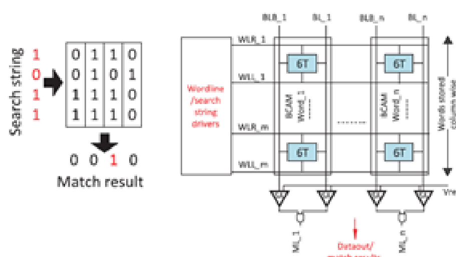

Fig. 4. Proposed CAM array organization.

IV. CONFIGURABLE MEMORY: BCAM/TCAM

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

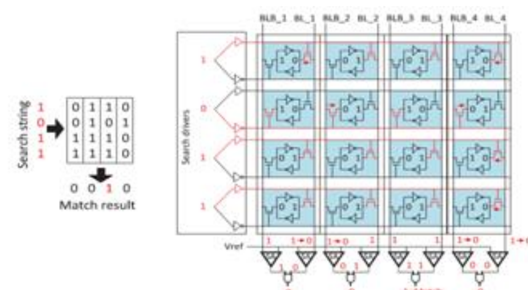

Fig. 5. BCAM search example. Only column 3 is a match. Transistors in red (gray in gray scale) are enabled.

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

discharged, and hence OUT senses 0, indicating a mismatch. In the match case, both OUT and OUTB stay high.

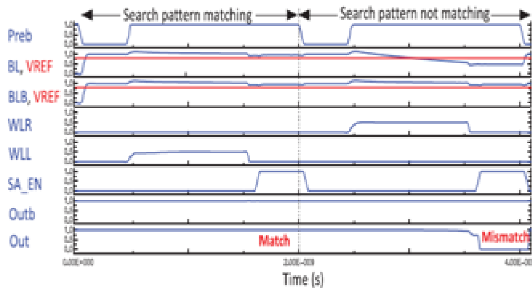

Fig. 6. Timing waveforms for BCAM search. Shows a "match" where both OUT and OUTB are high and a "mismatch" case.

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 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.

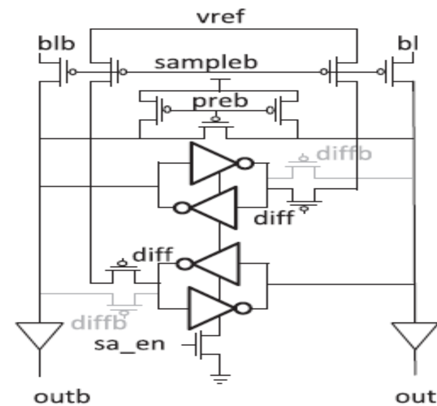

Fig. 7. Reconfigurable sense amplifier: two-single-ended amplifiers in CAM and logic modes; differential mode for SRAM.

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.

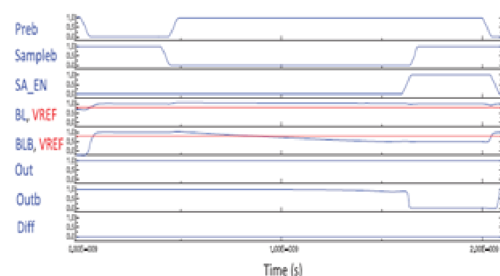

Fig. 8. Spice waveform for reconfigurable SA. "Diff" = 0, therefore, it is in two-single ended amplifier mode.

B. CAM Write Operation

One way to write the CAM is to use the SRAM mode and write the transpose of the required CAM data row-

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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.

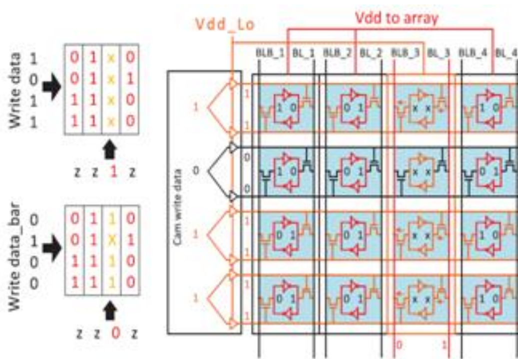

Fig. 9. BCAM column-wise write.In this example, column 3 is being written. Orange lines (light gray) are Vdd_Lo, while red (dark gray) lines are nominal Vdd.

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 columnwise 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 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 crosscouple 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.

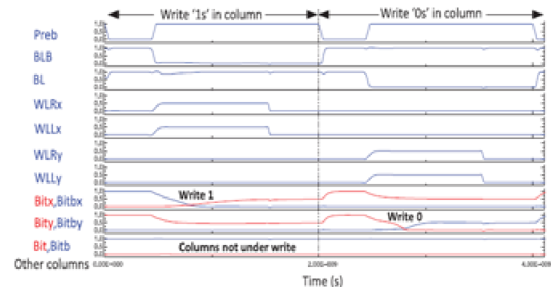

Fig. 10. Timing waveforms for two-cycle BCAM write. First cycle writes all the "1s" in the column, whereas the second cycle writes all the "0s." Notice that other columns not under write have their bit cells at full Vdd.

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 Vdd_Lo should not go below the retention voltage. The constraint for 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 Bitx (bit at row index "x" in the column-under-write) being written with a "1" followed by Bity in the same column being written with a "0" in the second cycle. While Bity is being written, Bitx holds its data at 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.

C. 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 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 supply voltage is needed for the configurable memory.

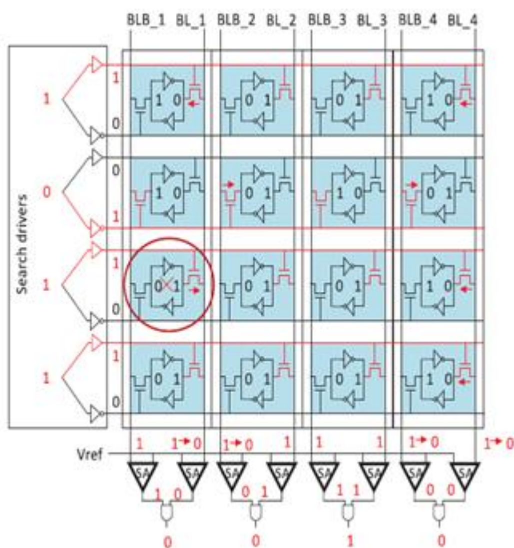

Fig. 11. BCAM search disturb: pseudowrite condition on encircled bit cell.

D. 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. 12, 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. 12. 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 bitline-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. 12, 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."

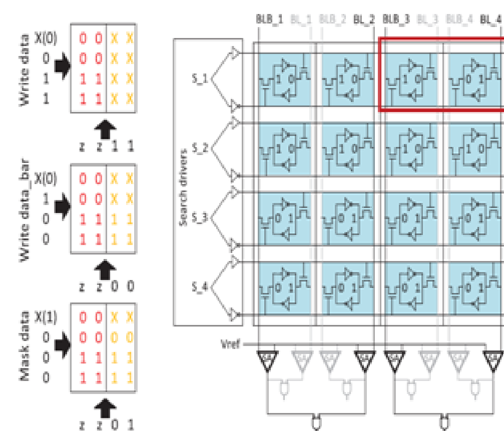

Fig. 12. TCAM mode organization. Two columns comprise a word. The bit cell in the top-right red box is "masked."

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. 12. In Fig. 13, we show the TCAM write operation's timing waveform. Bitx and Bitx+1 is written with "11," whereas Bity and Bity+1 are written with "'00."

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

To write mask in column-wordi in rowz, Bitz is written as “0”. and Bitz+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.

Fig. 13. 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

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. 14. 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.

In the example shown in Fig. 14, 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 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.

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

Note: 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.

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 signifi- cantly; hence the logic mode for two words can run much faster than the BCAM mode. For multi-word logic operations, the 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.

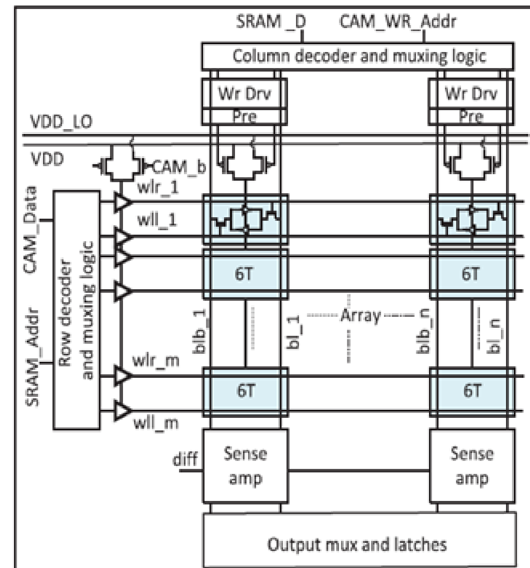

Fig. 15. Configurable memory organization

Fig. 15 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 wordline drivers to switch between Vdd and Vdd_Lo, as shown in Fig. 15. Also, most 6T SRAMs at advanced technology nodes, need some type of read/write assist techniques.

VI. SIMULATION RESULTS:

Along with the existing and proposed CAM design, both BCAM and TCAM using 6T Push Rule SRAM cell, CAMs of size 4 × 4 were implemented in the 65-nm technology node and postlayout simulation was performed to measure their performance using DSCH and Microwind. Following figures shows the DSCH and Microwind simulation result and the layout of the proposed CAM cell.

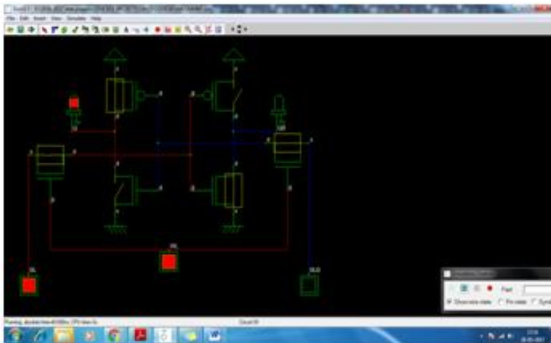
Fig 16: Schematic of 6T SRAM cell

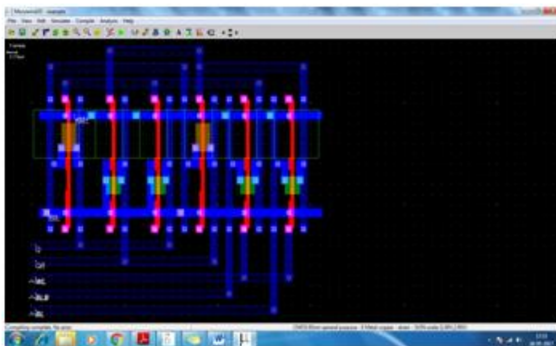
Fig 18: Layout of 6T SRAM cell

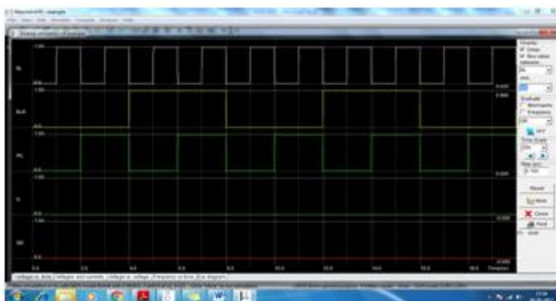
Fig 19: Simulation of Layout of 6T SRAM cell

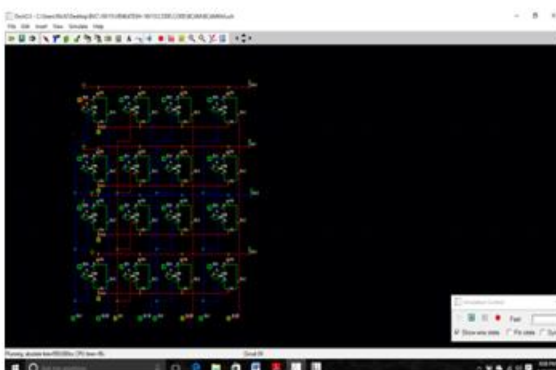
Fig 20: Schematic of Existing 4x4 BCAM array

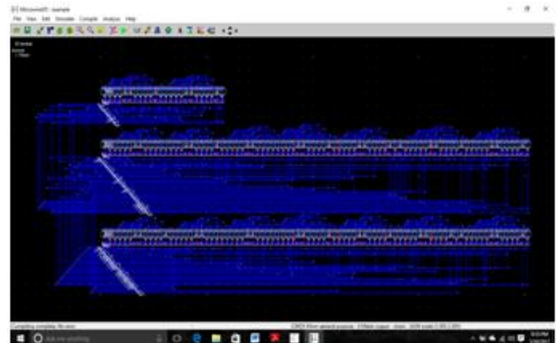
Fig 21: Layout of Existing 4x4 BCAM array

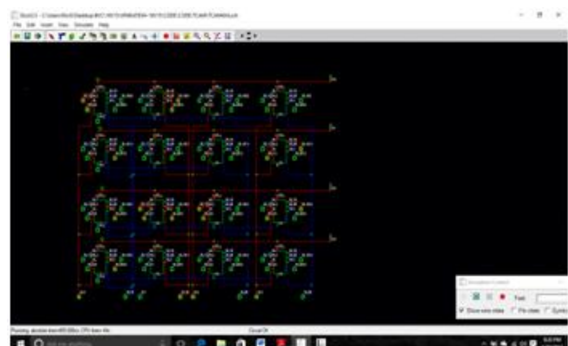
Fig 22: Schematic of Existing 4x4 TCAM array

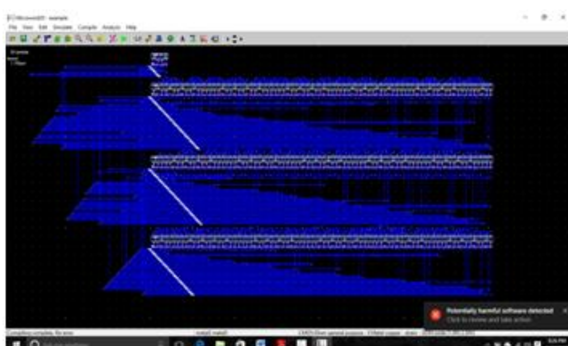
Fig 23: Layout of Existing 4x4 TCAM array

Fig 24: Schematic of Proposed 4x4 BCAM array

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

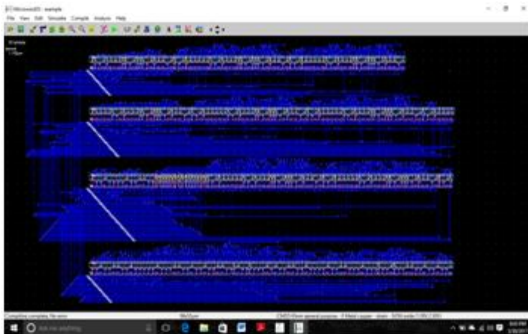
Fig 25: Layout of Proposed 4x4 BCAM array

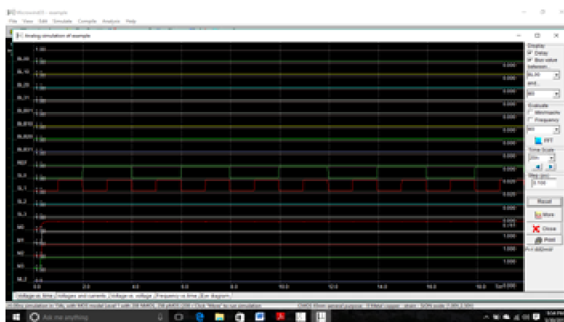
Fig 26: Simulation of Layout of Proposed 4x4 BCAM array

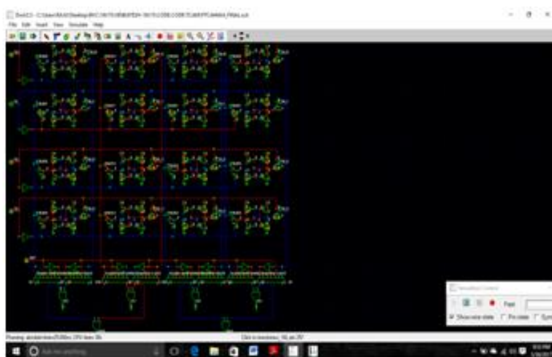
Fig 27: Schematic of Proposed 4x4 TCAM array

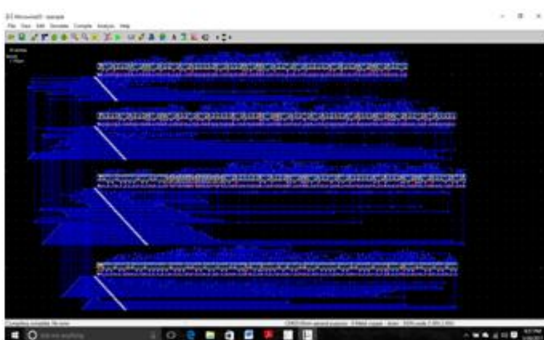
Fig 28: Layout of Proposed 4x4 TCAM array

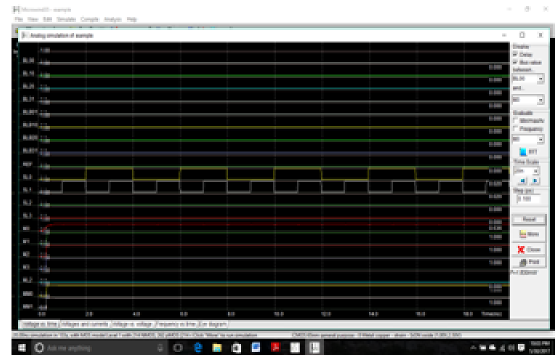
Fig 29 Simulation of Layout of Proposed 4x4 TCAM array

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. In conventional BCAM required 10T and TCAM required 16T due to this area and power consumptions are high comparatively proposed push-rule SRAM 6T based BCAM and TCAM. It also has lower instantaneous power because of less number of transistor counts. 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.

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. SolidState 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 µm2 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 lowvoltage 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 applicationspecific 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 deeptrench 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 SolidState Circuits Mag., vol. 7, no. 1, pp. 14–24, Feb. 2015.