



Low-power Hybrid CAM for High speed route Lookup Engines

Mr. K. SURESH KUMAR, Dr. Y. RAJASREE RAO, Dr. K. MANJUNATHACHARI

ECE, SSJ Engineering College, Hyderabad, India¹

ECE, SRIDEVI WOMEN'S Engg College, Hyderabad, India²

GITAM University, Hyderabad campus, India³

Abstract: Content-addressable memory (CAM) is a hardware table that can compare the search data with all the stored data in parallel. Due to the parallel comparison feature where a large amount of transistors are active on each lookup, however, the power consumption of CAM is usually considerable. This paper presents a hybrid-type CAM design which aims to combine the performance advantage of the NOR-type CAM with the power efficiency of the NAND-type CAM. In our design, a CAM word is divided into two segments, and then all the CAM cells are decoupled from the match line. By minimizing both the match line capacitances and switching activities, our design can largely reduce the power consumption of CAM. The experimental results show that the hybrid-type CAM can reduce the search energy consumption by roughly 89% compared to the traditional NOR-type CAM. Because the hybrid-type CAM provides a fast pull-down path to speed up the lightweight match line discharge, the search performance of our design is even better than that of the traditional NOR-type CAM.

Keywords: Content-addressable memory (CAM), hybrid-type CAM, NAND-type CAM, NOR-type CAM.

I. INTRODUCTION

A Content Addressable Memory (CAM) compares input search data against a table of stored data, and returns the address of the matching data. CAMs have a single clock cycle throughput making them faster than other hardware- and software-based search systems. CAMs can be used in a wide variety of applications requiring high search speeds. These applications include parametric curve extraction, Hough transformation, Huffman coding/decoding, Lempel-Ziv compression, and image coding. The primary commercial application of CAMs today is to classify and forward Internet protocol (IP) packets in network routers. In networks like the Internet, a message such as an e-mail or a Web page is transferred by first breaking up the message into small data packets of a few hundred bytes, and, then, sending each data packet individually through the network. These packets are routed from the source, through intermediate nodes of the network (called routers), and reassembled at the destination to reproduce the original message. The function of a router is to compare the destination address of a packet to all possible routes, in order to choose the appropriate one. A CAM is a good choice for implementing this lookup operation due to its fast search capability.

However, the speed of a CAM comes at the cost of increased silicon area and power consumption, two design parameters that designers strive to reduce. As CAM applications grow, demanding larger CAM sizes, the power problem is further exacerbated. Reducing power consumption, without sacrificing speed or area, is the main thread of recent research in large-capacity CAMs. In this paper, we survey developments in the CAM area at two levels: circuits and architectures. Before providing an outline of this paper at the end of this section, we first briefly introduce the operation of CAM and also describe the CAM application of packet forwarding. Fig. 1 shows a simplified block diagram of a CAM. The input to the system is the *search word* that is broadcast onto the *searchlines* to the table of stored data. The number of bits in a CAM word is usually large, with existing implementations ranging from 36 to 144 bits. A typical CAM employs a table size ranging between a few hundred entries to 32K entries, corresponding to an address space ranging from 7 bits to 15 bits. Each stored word has a *matchline* that indicates whether the search word and stored word are identical (the match case) or are different (a mismatch case, or miss). The matchlines are fed to an encoder that generates a binary *match location* corresponding to the matchline that is in the match state. An encoder is used in systems where only a single match is expected. In CAM applications where more than one word may match, a priority encoder is used instead of a simple encoder. A priority encoder selects the highest priority matching location to map to the match result, with words in lower address locations receiving higher priority. In addition, there is often a *hit* signal (not shown in the figure) that flags the case in which there is no matching location in the CAM. The overall function of a CAM is to take a search word and return the matching memory location. One can think of this operation as a fully programmable arbitrary mapping of the large space of the input search word to the smaller space of the output match location.

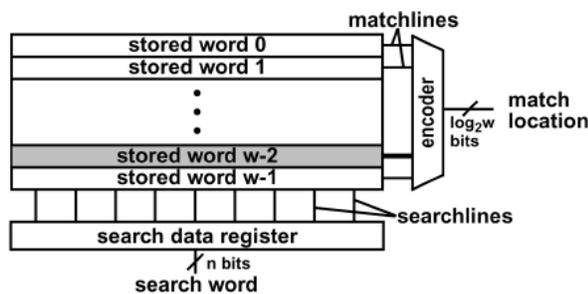


Fig. 1. Conceptual view of a content-addressable memory containing words.

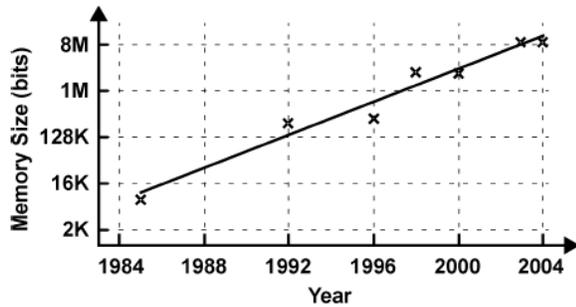


Fig. 2. Plot of CAM capacity (log scale) versus year of publication.

The operation of a CAM is like that of the tag portion of a fully associative cache. The tag portion of a cache compares its input, which is an address, to all addresses stored in the tag memory. In the case of a match, a single matchline goes high, indicating the location of a match. Unlike CAMs, caches do not use priority encoders since only a single match occurs; instead, the matchline directly activates a read of the data portion of the cache associated with the matching tag. Many circuits are common to both CAMs and caches; however, we focus on large capacity CAMs rather than on fully associative caches, which target smaller capacity and higher speed. Today's largest commercially available single-chip CAMs are 18 Mbit implementations, although the largest CAMs reported in the literature are 9 Mbit in size. As a rule of thumb, the largest available CAM chip is usually about half the size of the largest available SRAM chip. This rule of thumb comes from the fact that a typical CAM cell consists of two SRAM cells, as we will see shortly. Fig. 2 plots (on a logarithmic scale) the capacity of published CAM chips versus time from 1985 to 2004, revealing an exponential growth rate typical of semiconductor memory circuits and the factor-of-two relationship between SRAM and CAM.

II CONTENT-ADDRESSABLE MEMORY

There are two basic forms of CAM: binary and ternary. Binary CAMs support storage and searching of binary bits, zero or one (0,1). Ternary CAMs support storing of zero, one, or *don't care* bit (0,1,X). Ternary CAMs are presently the dominant CAM since longest-prefix routing is the Internet standard. Figure 3 shows a block diagram of a simplified 4 x 5 bit ternary CAM with a NOR-based architecture. The CAM contains the routing table. from Fig 4 to illustrate how a CAM implements address lookup. The CAM core cells are arranged into four horizontal words, each five bits long. Core cells contain both storage and comparison circuitry. The *search lines* run vertically in the figure and broadcast the *search data* to the CAM cells. The *matchlines* run horizontally across the array and indicate whether the search data matches the row's word. An activated matchline indicates a match and a deactivated matchline indicates a non-match, called a *mismatch* in the CAM literature. The matchlines are inputs to an encoder that generates the address corresponding to the match location.

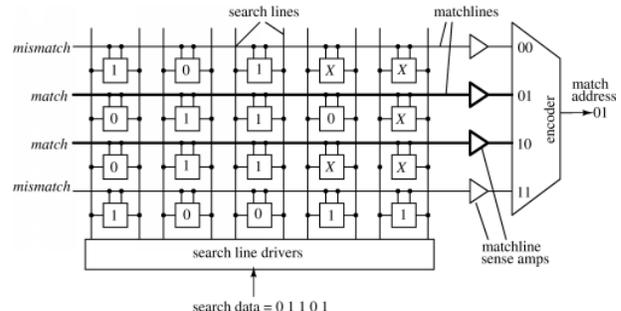


Figure 3: NOR-based CAM architecture

A CAM search operation begins with precharging all matchlines high, putting them all temporarily in the match state. Next, the search line drivers broadcast the search data, 01101 in the figure, onto the search lines. Then each CAM core cell compares its stored bit against the bit on its corresponding search lines. Cells with matching data do not affect the matchline but cells with a mismatch pull down the matchline. Cells storing an X operate as if a match has occurred. The aggregate result is that matchlines are pulled down for any word that has at least one mismatch. All other matchlines remain activated (precharged high). In the figure, the two middle matchlines remain activated, indicating a match, while the other matchlines discharge to ground, indicating a mismatch. Last, the encoder generates the search address location of the matching data. In the example, the encoder selects numerically the smallest numbered matchline of the two activated matchlines, generating the *match address 01*. This match address is used as the input address to a RAM that contains a list of output ports as depicted in Figure 4. This CAM/RAM system is a complete implementation of an address lookup engine. The match address output of the CAM is in fact a pointer used to retrieve associated data from the RAM. In this case the associated data is the output port. The CAM/RAM search can be viewed as a dictionary lookup where the search data is the word to be queried and the RAM contains the word definitions. With this sketch of CAM operation, we now look at the comparison circuitry in the CAM core cells.

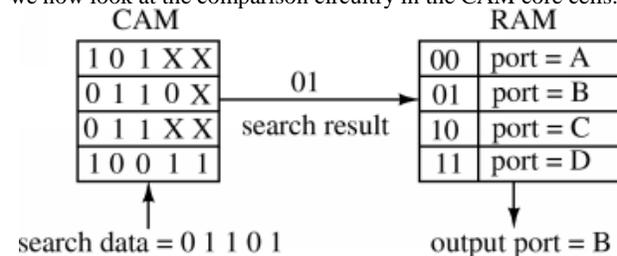


Figure 4: Address lookup with CAM/RAM.

CAM Circuits:

Figure 5(a) displays a conventional SRAM core cell that stores data using positive feedback in back-to-back inverters. Two access transistors connect the bitlines, *bl* and */bl* (we use the prefix / to denote the logical complement in the text and we use an over bar in the figures), to the storage nodes under control of the wordline, *wl*. Data can be read from the cell or written into the cell through the bitlines. We use this differential cell as the storage for building CAM cells. Figure 5(b) depicts a conventional binary CAM (BCAM) cell with the matchline denoted *ml* and the differential search lines denoted *sl* and */sl*. The figure also lists the truth value,



T , stored in the cell based on the values of d and \bar{d} . Read and write access circuitry is omitted for clarity in this figure and subsequent CAM core cell figures. For a binary CAM, we store a single bit differentially. The comparison circuitry attached to the storage cell performs a comparison between the data on the search lines (sl and \bar{sl}) and the data in the binary cell with an XNOR operation ($ml = ! (d \text{ XOR } sl)$). A mismatch in a cell creates a path to ground from the matchline through one of the series transistor pairs. A match of d and sl disconnects the matchline from ground.

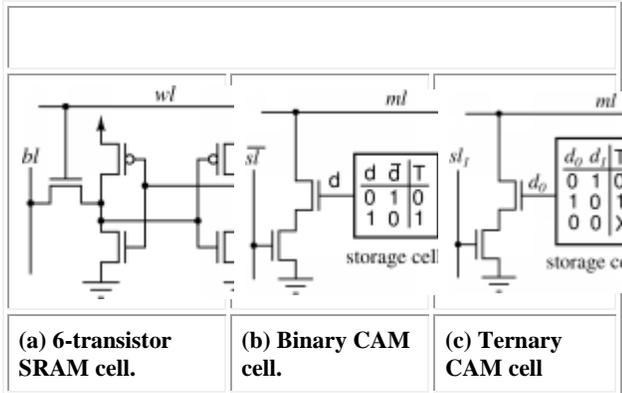


Fig 5. conventional SRAM storage cell, Binary CAM cell, Ternary CAM cell.

A. NOR-Type CAM

Traditionally, there are two types of CAM designs. As shown in Fig. 6, one is NOR-type and the other is NAND-type. In NOR-type CAM design, the CAM cell is usually XOR-type, and the pull-down transistors of each CAM cell are arranged in NOR type. During precharge phase, the match line is initially precharged to high. For a CAM word, if one or more cells are mismatched, the match line would be discharged to 0. Only when all cells are matched, i.e., the search data are identical to the stored data, the match line can retain logic high as in the precharge phase. Because the pull-down path is very short, in case of a mismatch the match line is discharged to 0 quickly. Thus, the NOR-type CAM provides the best search performance. Note that the pull-down transistors arranged in NOR type is beneficial for search performance, but they contribute a lot of drain capacitances to the match line. This results in more power dissipated in match line switching. Because in many applications most of the CAM words are mismatched, a large number of match line switches would consume a significant dynamic power. For example, in the CAM tag used in the TLB or cache memory, at most one word is matched on each lookup, which implies that almost all the match lines would be discharged to 0, and then be charged to high before the next search. Consequently, the NOR-type CAM is power inefficient, although it can provide the best performance.

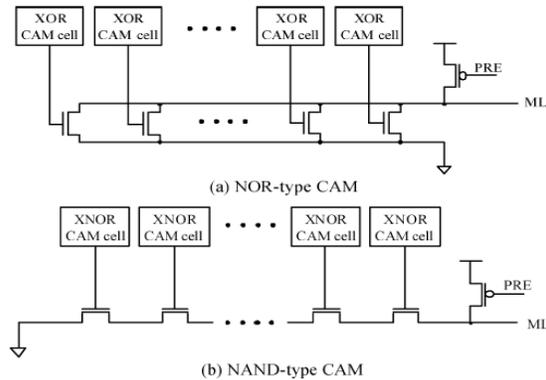


Fig. 6. Traditional CAM designs.

B. NAND-Type CAM

In contrast to the NOR-type CAM, the NAND-type CAM aims to reduce the power dissipated in search operation, in which the CAM cell is implemented as XNOR-type instead of XOR-type, and the pull-down transistors of each CAM cell in the same word are arranged in NAND type, as shown in Fig. 6(b). The match line is initially precharged to high, and discharged to 0 only when all CAM cells are matched. Because the load capacitance of match line is small and only one match line is discharged to 0 during a search, the power consumption is minimal. However, the pull-down path is too long, such that the match line discharge is very slow in case of a match. Thus, the NAND-type CAM trades the performance degradation for a large power saving.

III. IMPLEMENTATION ISSUES

Depending on the application, user can adjust the length of SEG₁. If the length of a CAM word is bits and the length of SEG₁ is bits, then the length of SEG₂ would be n-x bits. In the SEG₁, because all the pull-down transistors are arranged in serial mode (i.e., NAND-type block), and they are on the critical path to discharge the match line, the length of SEG₁ is a powerful lever on the functionality, performance and power efficiency in our design.

A. SEG₁ Length Versus Race Condition

From Fig.7, we note that the speed of M1 discharge depends on the length of SEG₁. This implies that there is a possible race condition problem in case 2, i.e., SEG₁ is matched & SEG₂ is mismatched: a) If the M1 discharge is fast enough, then the tail transistor N3 would be turned on quickly to discharge M2, such that N4 transistor is turned off quickly to prevent the matchline from discharging; therefore, the logic high level of match line can be retained correctly; and b) in the other case, if the M1 discharge is too slow to prolong the on time of N4 transistor, then the match line would be discharged unexpectedly; if the voltage level of match line is too low, then it is a false match. To prevent the incorrect match incurred by the race condition, we add a PMOS transistor, N4, to provide the level-restore capability. Once the M2 node is discharged to 0, regardless of discharge speed, N4 transistor would be turned on to supply the lost charge. Consequently, our design provides the immunity from the potential race condition



problem. This effect can be observed from Fig. 5, in which there is a small pulse marked with the circle. The lost charge would be supplied quickly.

B. SEG_1 Length Versus Charge Sharing

If the length of SEG_1 is too long, the *charge sharing* problem would possibly occur when SEG_1 is mismatched and SEG_2 is matched. As shown in Fig. 6, the worst case is that all the pull-down transistors are turned on but the most left one. In this case, the charge of **M1** node would be shared among the intermediate nodes, i_0-i_5 , such that the voltage level of **M1** node is decreased. Because SEG_2 is matched, **N4** is turned on to discharge the match line. If the voltage level of match line is too low, then it results in a false match.

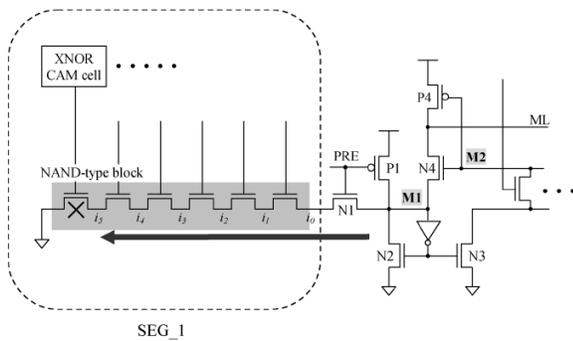


Fig. 7. Example of the charge sharing problem incurred by large SEG_1.

C. SEG_1 Length Versus Power Saving

As described above, short SEG_1 can prevent the charge sharing problem, but it increases the probability of **M1** discharge. Suppose, for example, that the length of SEG_1 is one bit. For a random pattern, the probability of **M1** discharge would be 50% on average, i.e., the probability of tail transistor **N3** turned on is also 50%. Because there are n-1 pull-down transistors in the NOR-type block, the probability of **T3** path connected to the ground would increase largely. It results in a significant power dissipated in the discharge of the **M2** node with large drain capacitances. Ideally, the probability of **M2** discharge is the product of the probability of **T1** path conducting, i.e., (**T1** conducting), and the probability of **T3** path conducting, i.e., (**T3** conducting), as shown in the following equation:

$$P(\mathbf{T}_1 \text{ conducting}) \times P(\mathbf{T}_3 \text{ conducting}) \\ = (1/2)^x \times (1 - (1/2)^{n-x}) = (1/2)^x - (1/2)^n.$$

where n and x are the lengths of the entire word and SEG_1, respectively. In this equation, we assume that the match probability is 0.5 for each CAM cell. In the SEG_2, because all the pull-down transistors are arranged in the NOR type, the **T3** path is disconnected only when they are all turned off. Thus, (**T3** conducting) is equal to $(1 - (1/2)^{n-x})$. Fig. 8 shows the probability of **M2** discharge for various SEG_1 lengths, in which n=32 is assumed. Clearly, the probability of **M2** discharge decreases sharply as the length of SEG_1 is increased. This implies that the

search operation would consume more power when we decrease the length of SEG_1.

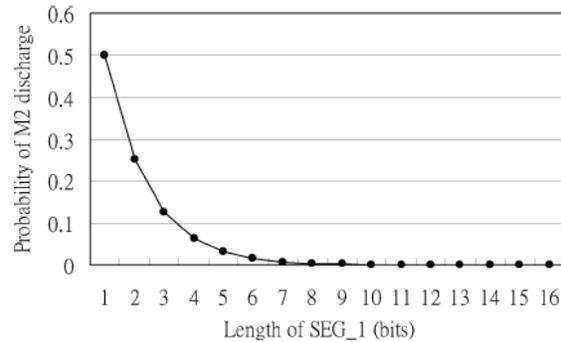


Fig. 8. Probability of M2 discharge for various SEG_1 lengths.

IV. EXPERIMENTAL RESULTS

In this paper, For a substantial comparison, besides the conventional NOR-type and NAND type CAM, we also implement the related designs, including the selective precharge scheme, the static divided word structure, and the SPCAM that operates in serial and parallel mode. They are denoted as SP, SDW, SPCAM_S, and SPCAM_P, respectively. All CAM designs are with size of 128x 32, i.e., 128 words by 32 bits, and the data presented in the following discussion are obtained from the HSPICE post layout simulation.

A. Performance

In this paper, the metric used to evaluate the CAM performance is the *match delay*, which is defined as the elapsed time from signal PRE is asserted high to the match line discharged to 0 in case of a match. Table I lists the match delay of all CAM designs where the SEG_1 length is varied from 1 to 6 bits. Due to no segmentation, the match delay of the NOR-type and NAND-type CAM are fixed at 0.641 ns and 2.774 ns, respectively. In other words, the match delay of NAND-type CAM is 4.3 times larger than that of NOR-type CAM. As indicated in the background and, the NAND-type CAM is not a feasible solution because of its long match delay. Fig. 9 shows the normalized match delay where the match delay of all CAM designs are normalized to that of the NOR-type CAM. From this figure, we summarize the most important aspects as follows. (1) It is clear that the search performance of our design is better than the other word segmentation techniques. Particularly, only the hybrid-type CAM (with SEG_1 length ≤ 4) has better search performance than the conventional NOR-type CAM. Because the match line is precharged conditionally, the search performance of SP[7], SDW, and SPCAM_S[7] is worse than that of NOR-type CAM. Note that the match delay of SPCAM_P[7] is almost the same as that of NOR-type CAM. This is because in SPCAM_P [7] both SEG_1 and SEG_2 are always active concurrently for high search performance. (2) The SEG_1 length has a significant impact on the search performance for all word segmentation techniques except for SPCAM_S [7] and SPCAM_P [7], in which the SEG_1 is broken into several sets of two bits to limit the number of transistors in series to three. In our design the match delay increases with the length of SEG_1. As shown in Fig. 3, the match line discharge relies on the **M1** discharge that connects **T1** and **T2** paths to the ground; and further, **M1** discharge delay increases with



the number of transistors in the NAND-type block. (3) One interesting observation from this result is that when the SEG_1 length is less than or equal to four bits, the match delay of our design is even shorter than that of the conventional CAM. This is because our design decouples all CAM cells from the match line, such that the match line is lightweight. Once the fast path T2 is connected, it can discharge the lightweight match line quickly. Although the NAND-type block would degrade the match performance slightly, the fast path T2 can compensate for the performance loss. Due to the charge sharing problem, the length of SEG_1 is constrained within four bits. From the detailed data shown in Table II, if the length of SEG_1 is 4, the match delay is 0.609 ns. Compared to the conventional NOR-type CAM, our design can improve the search performance by 5%.

B. Power and Energy

Table I shows the power consumption during a search for all CAM designs where the SEG_1 length is varied from 1 to 6 bits. Clearly, the search power consumption can be reduced sharply when the SEG_1 length is increased except for SPCAM_P [7], in which SEG_1 and SEG_2 are checked concurrently for high search performance. No matter whether SEG_1 is match or not, SEG_2 is always active on every search, such that the search power of SPCAM_P [7] is slightly less than that of NOR-type CAM. In the proposed hybrid-type CAM, the search power consumption is roughly 0.29 mW as the SEG_1 length is 6 bits. Compared to the NOR-type CAM, whose search power consumption is fixed at 3.04mW, our design can reduce roughly 90% of the search power consumption. Besides reducing the search power consumption, increasing SEG_1 length has a large impact on the search performance as revealed in Section IV-A. Consequently, increasing SEG_1 length is a tradeoff between power and performance. For a fair comparison, energy is a suitable metric, which is the product of the match delay (performance) and search power (power). Fig. 9 shows the normalized search energy, in which all energy results are normalized to that of the NOR-type CAM design. From Fig. 9, it is clear that the word segmentation techniques are indeed effective in reducing the energy consumption of CAM except for SPCAM_P [7] where both SEG_1 and SEG_2 are always active for high search performance. Actually, the search energy of both SPCAM_P [7] and NOR-type CAM are almost the same. The major advantage of our design is that it not only largely reduces the search power, but also improves the match delay. In contrast, SP, SDW and SPCAM_S [7] would result in a delay penalty while reducing the search power. Consequently, our design can achieve the most energy improvement compared to the other related techniques. In addition, the energy efficiency of our design is even better than that of the NAND-type CAM which has the lowest search power. From Table I, our design can reduce the energy consumption of NOR-type CAM by 90% as the SEG_1 length is 6 bits. The improvement difference between SEG_1=6 and SEG_1=4 is marginal. However, if the SEG_1 length is larger than 4 bits, due to charge sharing a possible false match does exist in our design. For a reliable system, when the SEG_1 length is 4 bits, our design can reduce the energy consumption of NOR-type and NAND-type CAM by roughly 88% and 40%, respectively.

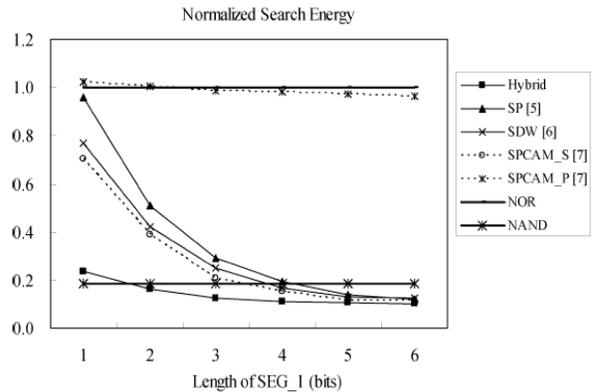


Fig.9 Normalized search energy.

Energy (pJ)	Hybrid	SP [5]	SDW [6]	SPCAM_S [7]	SPCAM_P [7]	NOR	NAND
1	0.4621	1.8653	1.4968	1.3734	1.9913	1.9486	0.3647
2	0.3196	0.9905	0.8259	0.7612	1.9547	1.9486	0.3647
3	0.2435	0.5652	0.4897	0.4091	1.9274	1.9486	0.3647
4	0.2164	0.3766	0.3271	0.2996	1.9160	1.9486	0.3647
5	0.2044	0.2708	0.2490	0.2217	1.8939	1.9486	0.3647
6	0.1979	0.2334	0.2401	0.2286	1.8753	1.9486	0.3647

TABLE I
 SEARCH ENERGY OF ALL CAM DESIGNS.

C. Area Cost

In contrast with the conventional NOR-type CAM, our design costs nine additional transistors which all come from the control circuitry. For a CAM word with 32 bits, the layout size of the conventional NOR-type and our design are 7.54m×131.21 m and 7.54 m ×141.57 m. Note that the height of the proposed CAM is purposely retained the same as the height of the conventional NOR-type CAM, such that both designs have the same power dissipated in the bit line switching. The area overhead is roughly 7.8%. Because the CAM words are part of the entire CAM system, the total CAM area overhead is less than 7.8%.

V. CONCLUSIONS

In this paper, we have developed a hybrid-type CAM design, in which we decouple all the CAM cells from the match line, and provide a fast path to accelerate the search operation. With a marginal area overhead, our design not only largely reduces the search power consumption but also improves the search performance.

REFERENCES

[1] 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.



- [2] K. Pagiamtzis and A. Sheikholeslami, "A low power content-addressable memory (CAM) using pipelined hierarchical search scheme," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1512–1519, Sep. 2004.
- [3] S. P. Mohanty, N. Ranganathan, and S. K. Chappidi, "Peak power minimization through datapath scheduling," in *Proc. IEEE Computer Soc. Annu. Symp. VLSI (ISVLSI)*, Feb. 2003, pp. 121–126.
- [4] Y. J. Chang, F. Lai, and C. L. Yang, "Zero-aware asymmetric SRAM cell for reducing cache power in writing zero," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 12, no. 8, pp. 827–836, Aug. 2004.
- [5] C. S. Lin, J. C. Chang, and B. D. Liu, "A low-power precomputation base fully parallel content addressable memory," *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 654–662, Apr. 2003.
- [6] I. Arsovski and A. Sheikholeslami, "A mismatch-dependent power allocation technique for match-line sensing in content-addressable memories," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1958–1966, Nov. 2003.
- [7] 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, no. 6, pp. 956–968, Jun. 2001.
- [8] C. A. Zukowski and S. Y. Wang, "Use of selective precharge for lowpower content-addressable memories," in *Proc. Int. Symp. Circuits and Syst.*, 1997, pp. 1788–1791.

BIOGRAPHY



K.Suresh Kumar is an Assistant Professor in SSJ Engineering college, Hyderabad. He received his B.Tech degree in ECE from CVR Engineering college, JNTUH. M.E degree in VLSI Design from PSNA college of engineering & technology, ANNA University, Trichy. He was a research scholar in Electronics &

Communication Engineering department, JNTUH. He has more than 4 years of experience in teaching. His current research interest includes in Low-power VLSI design.