Low Power SCN CAM Based on Parity

Ashidha E. B[1] , Elsa Jose Palatty[2]

[1]*PG Student Met's School of Engineering Department of Electronics and Communication Engineering*

[2]*Asst. Professor Met's School of Engineering Department of Electronics and Communication Engineering*

Thrissur, Kerala, India

Abstract— we propose a low power content addressable memory (CAM) employing a new algorithm. This is based on a parity based SCN. The proposed idea is based on the recently developed parity based on SCN-CAM. The parity bit based CAM design is consist of the original data segment and an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, i.e., odd or even number of "1"s. The obtained parity bit is placed directly to the corresponding word. Thus the new architecture has the same interface as the conventional CAM with one extra bit.

Keywords—Sparse clustered network(SCN),Content addressable memory (CAM).

I. INTRODUCTION

Content-addressable memory (CAM) is a special type of computer memory used in certain very-high-speed searching applications. It is also known as associative memory, associative storage, or associative array, although the last term is more often used for a programming data structure. It compares input search data (tag) against a table of stored data, and returns the address of matching data[1] (or in the case of associative memory, the matching data. Several custom computers, like the Goodyear STARAN, were built to implement CAM, and were designated associative computers.

Unlike standard computer memory (random access memory or RAM) in which the user supplies a memory address and the RAM returns the data word stored at that address, a CAM is designed such that the user supplies a data word and the CAM searches its entire memory to see if that data word is stored anywhere in it. If the data word is found, the CAM returns a list of one or more storage addresses where the word was found (and in some architectures, it also returns the contents of that storage address, or other associated pieces of data)[2]. Thus, a CAM is the hardware embodiment of what in software terms would be called an associative array. Because a CAM is designed to search its entire memory in a single operation, it is much faster than RAM in virtually all search applications. There are cost disadvantages to CAM however. Unlike a RAM chip, which has simple storage cells, each individual memory bit in a fully parallel CAM must have its own associated comparison circuit to detect a match between the stored bit and the input bit.

Additionally, match outputs from each cell in the data word must be combined to yield a complete data word match signal. The additional circuitry increases the physical size of the CAM chip which increases manufacturing cost. The extra circuitry also increases power dissipation since every comparison circuit is active on every clock cycle. Consequently, CAM is only used in specialized applications where searching speed cannot be accomplished using a less costly method. One successful early implementation was a General Purpose Associative Processor IC and System.

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 1 shows a block diagram of a simplified 4 x 5 bit ternary CAM with a NOR-based architecture. 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 match lines run horizontally across the array and indicate whether the search data matches the row's word. An activated match line indicates a match and a deactivated match line indicates a non-match, called a mismatch in the CAM literature. The match lines are inputs to an encoder that generates the address corresponding to the match location.

The parity bit based CAM design is consist of the original data segment and an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, i.e., odd or even number of "1"s. The obtained parity bit is placed directly to the corresponding word. Thus the new architecture has the same interface as the conventional CAM with one extra bit. During the search operation, there is only one single stage as in conventional CAM. Hence, the use of this parity bits improve the power performance. In the case of a matched in the data segment the parity bits of the search and the stored word is the same, thus the overall word returns a match. When 1 mismatch occurs in the data segment, numbers of "1"s in the stored and search word must be different by 1. As a result, the corresponding parity bits are different. Therefore now we have two mismatches (one from the parity bit and one from the data

bits). If there are two mismatches in the data segment the parity bits are the same and overall we have two mismatches. With more mismatches, we can ignore these cases as they are not crucial cases. Since the driving capability of the 2-mismatch word is twice as strong as that of the 1-mismatch word, the proposed design greatly improves the search speed.

II. CAM REVIEW

A Content Addressable Memory (CAM) is a type of memory that can be accessed using its contents rather than an explicit address. In order to access a particular entry in such memories, a search data word is compared against previously stored entries in parallel to find a match. Each stored entry is associated with a tag that is used in the comparison process. Once a search data word is applied to the input of a CAM, the matching data word is retrieved within a single clock cycle if it exists due to the frequent and parallel search operations, CAMs consume a significant amount of energy.

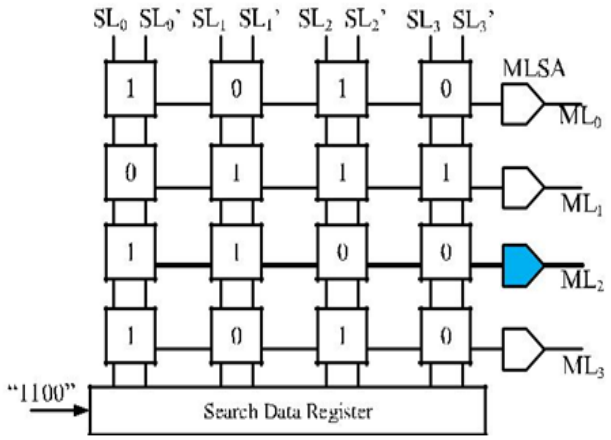

Figure 1. Simple example of a 4x4 CAM

Although dynamic CMOS circuit techniques can result in low-power and low-cost CAMs, these designs can suffer from low noise margins, charge sharing, and other problems.

Content-addressable memories (CAMs) are hardware search engines that are much faster than algorithmic approaches for search-intensive applications. CAMs are composed of conventional semiconductor memory (usually SRAM) with added comparison circuitry that enables a search operation to complete in a single clock cycle. The two most common search-intensive tasks that use CAMs are packet forwarding and packet classification in Internet routers. I introduce CAM architecture and circuits by first describing the application of address lookup in Internet routers. Then we describe how to implement this lookup function with CAM.

The figure 1. Shows the typical CAM. A search data word is given which is of length four. This is applied to search data register. This will store the input bits. These search datas are applied to each search lines SLs, a common match line ML is connected. MLs are highly capacitive and a sense amplifier is connected to each ML in order to increase the performance. Usually Binary CAMs are used.

III. RELATED WORK

Energy reduction of CAM can be done in cell level. In all the papers before they have done using circuit. The paper just before this was SCN based on CAM [1]. Here the data bit is entirely divided in to set of 3 or 4. This is given to CAM block. CAM blocks will check its availability in SRAM and give corresponding address. But it is time consuming and has power loss.

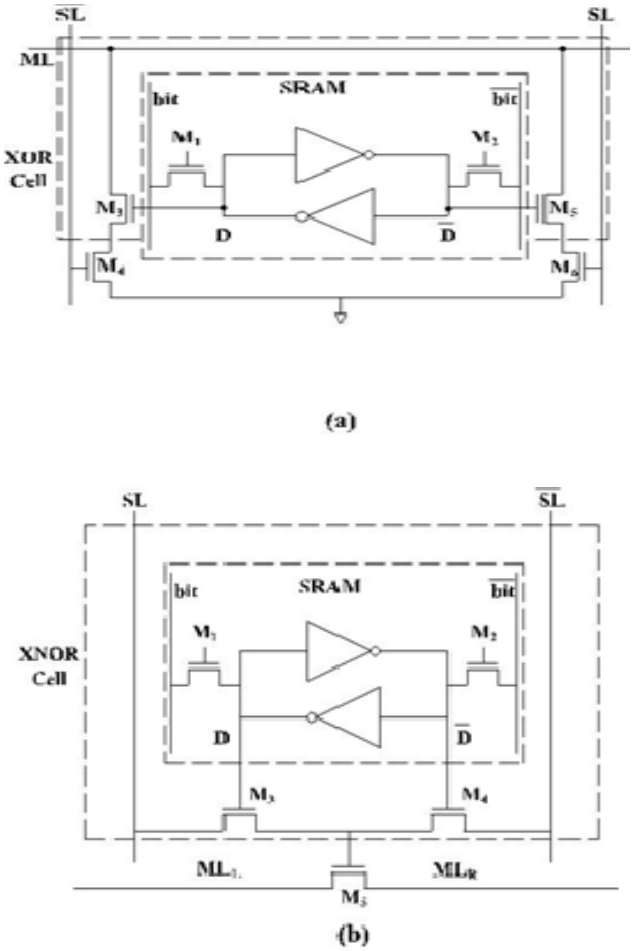

Figure 2. Classical BCAM cell types
(a) 10T NOR. (b)9T NAND

Energy consumption can be reduced in circuit level. This is done by reducing SL energy consumption by disabling pre charge process and reducing ML pre charging. [18], [25]-[28].

But this will increase the delay while adding segments. A high performance NOR type gate and power reducing NAND type gate is used in order to make Hybrid CAM [30]. Here ML is segmented in to two parts. It has 10 transistors and 9 transistors which is used in the conventional CAM. This is explained in fig 2.

In the bank selection architecture [31], [32], the CAM array is divided into equal banks. These banks are activated in terms of added bits. These extra bits are decoded in order to determine which bank is selected. But here much comparison occurs. So time delay increases and power consumption increases. In asynchronous architecture proposed [15] consecutive search data matches; it operates based on delay of matching. But the cycle time drastically increases, so the delay increases. Asynchronous architecture is more susceptible to process variations while comparing synchronous architectures.

IV. PARITY BASED SCN-CAM ALGORITHM

The parity bit based CAM design is consist of the original data segment and an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, i.e., odd or even number of "1"s. The obtained parity bit is placed directly to the corresponding word. Thus the new architecture has the same interface as the conventional CAM with one extra bit. During the search operation, there is only one single stage as in conventional CAM. Hence, the use of this parity bits improve the power performance shown in fig 3. In the case of a matched in the data segment the parity bits of the search and the stored word is the same, thus the overall word returns a match. When 1 mismatch occurs in the data segment, numbers of "1"s in the stored and search word must be different by 1.

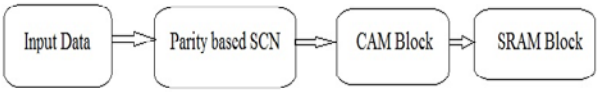

Figure 3. Block diagram of Parity based SCN

As a result, the corresponding parity bits are different. Therefore now we have two mismatches (one from the parity bit and one from the data bits). If there are two mismatches in the data segment the parity bits are the same and overall we have two mismatches. With more mismatches, we can ignore these cases as they are not crucial cases. Since the driving capability of the 2-mismatch word is twice as strong as that of the 1-mismatch word, the proposed design greatly improves the search speed. There are two types of parity. Even parity and odd parity .we keep one extra bit for storing parity. Already one parity will be stored in the CAM memory. This will compared by the parity derived from the input data. The main advantage by comparing SCN based CAM is reduction of time delays and reduction in power.

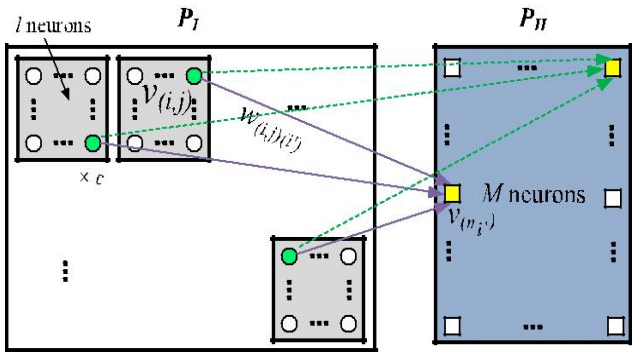

Figure 4. SCN based classifier

Parity logic is added along with the input. Parity is determined depending up on number of ones. First of all we trained the classifier with data bits associated with input data. The SCN-CAM is divided in to several sub blocks. By applying input enable signal is given to classifier. For example, if the input is 16 bit, by adding an even or odd parity the length will be 17. This is given to SCN classifier. Now it is splitted to group of four or three. Then compare it with the stored bits and give the required address.

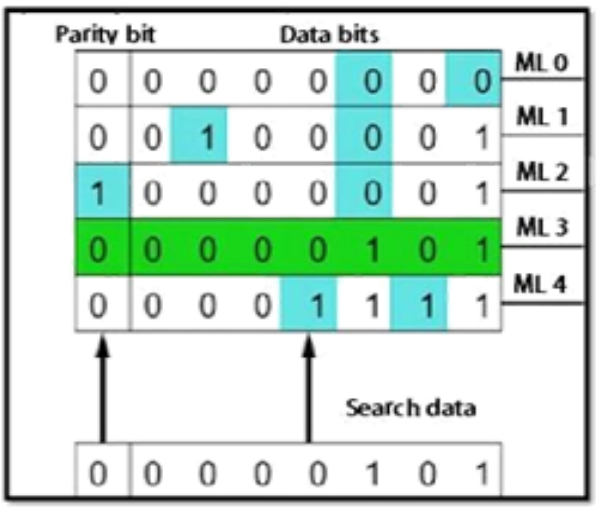

Figure 5. Search Speed Boosting With a Parity Bit

Comparisons can be reduced by using parity based SCN. The fig.4 shows Parity based SCN classifier. This consists of two parts, P1 and p11. The input tag is splitted to q equal parts and training and decoding is done. Each section is of c size. It is calculated using **c=q/log(*l*) .** Each sub block consists of l neurons and p11 consists of M neurons.

The Parity classifier in SCN-CAM architecture generates the compare-enable signal(s) for the CAM sub-blocks attached to it. The architecture of the SCN-based classifier is shown in Fig. 6. It consists of c k-to-l one-hot decoders,

c SRAM modules of size l×M each, M c−input AND gates, M/ζ ζ−input OR gates, M/ζ 2-input NAND gates and a parity bit. Each row of an SRAM module stores the connections from one tag to its corresponding output neuron depends on parity. Each reduced-length tag of length q is thus divided into c sub tags of k bits each, where each sub tag creates the row address of each SRAM module.

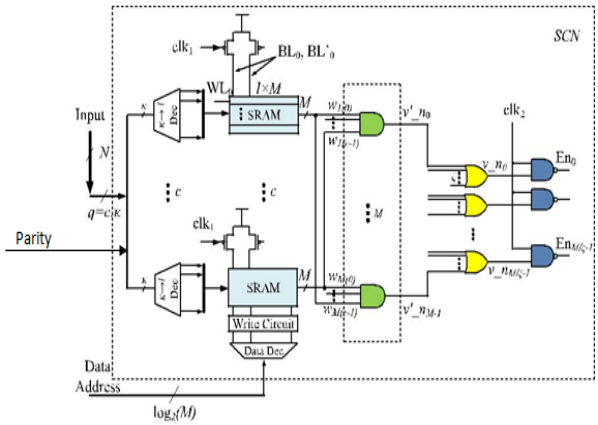

Figure 6. Architecture of Parity based SCN classifier

Training, decoding and updating are the 3 operations done here. During training process, SRAM will store connection value between input data and the output address. The decoding process is implemented using the structure of the SRAM modules, and the c-input AND gates. To update an entry after deletion, the new entry added to the network using the above approach. Finally the modified row is given to the SRAM block.

V. EXPERIMENTAL RESULT

Here VHDL is used as the coding language and it is a hardware description language used in electronic design automation to describe digital mixed signal systems such as field programmable gate arrays and integrated circuits. It is developed by the U.S Department of Defense in order to document the behavior of the ASIC's and can be used as a general purpose parallel programming language. Here in the proposed work simulation can be done using modelsim and comparison can be done using Xilinx. Modelsim is a multi-language HDL simulation environment used for simulation of much hardware description languages such as VHDL, verilog and system C and is developed by Mentor Graphics.

Fig 7 shows the simulation result of Parity based SCN-CAM. The parity bit based CAM design is consist of the original data segment and an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, i.e., odd or even number of "1"s. The obtained parity bit is placed directly to the corresponding word. By comparing each input data and stored data output address is obtained.

Figure 7.Simulation result for Parity based SCN-CAM

VI. CONCLUSION

In this paper, the algorithm and the architecture of a low-power CAM are introduced. The proposed architecture (Parity based SCN-CAM) employs an associativity mechanism based on a recently developed family of associative memories based on SCNs. Parity based SCN-CAM is suitable for low-power applications, where frequent and parallel look-up operations are required. It employs an Parity input, SCN-based classifier, which is connected to several independently compare-enabled CAM sub-blocks, some of which are enabled once a tag is presented to the SCN-based classifier. By using independent nodes in the output part of SCN-CAM's training network, simple and fast updates can be achieved without retraining the network entirely. With optimized lengths of the reduced-length tags, Parity based SCN-CAM eliminates most of the comparison operations. It reduces timing delays and increases the performance. It reduces the power consumption.

REFERENCES

[1] Hooman Jarollahi, "Algorithm and Architecture for a Low-Power Content-Addressable Memory Based on Sparse Clustered Networks",IEEETRANS.Very Large Scale Integr.(VLSI),vol.23, April 2015.

[2] Y.-J. Chang and M.-F. Lan, "Two new techniques integrated for energyefficient TLB design," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 1, pp. 13–23, Jan. 2007.

[3] H. Chao, "Next generation routers," Proc. IEEE, vol. 90, no. 9, pp. 1518–1558, Sep. 2002.

[4] N.-F. Huang, W.-E. Chen, J.-Y. Luo, and J.-M. Chen, "Design of multifield IPv6 packet classifiers using ternary CAMs," in Proc. IEEE Global Telecommun. Conf., vol. 3. 2001, pp. 1877–1881.

[5] M. Meribout, T. Ogura, and M. Nakanishi, "On using the CAM concept for parametric curve extraction," IEEE Trans. Image Process., vol. 9, no. 12, pp. 2126–2130, Dec. 2000.

[6] M. Nakanishi and T. Ogura, "A real-time CAM-based Hough transform algorithm and its performance evaluation," in Proc. 13th Int. Conf. Pattern Recognit., vol. 2. Aug. 1996, pp. 516–521.

[7] L.-Y. Liu, J.-F. Wang, R.-J. Wang, and J.-Y. Lee, "CAM-based VLSI architectures for dynamic Huffman coding," IEEE Trans. Consum. Electron., vol. 40, no. 3, pp. 282–289, Aug. 1994.

[8] C.-C. Wang, C.-J. Cheng, T.-F. Chen, and J.-S. Wang, "An adaptively dividable dual-port BiTCAM for virus-detection processors in mobile devices," IEEE J. Solid-State Circuits, vol. 44, no. 5, pp. 1571–1581, May 2009.

[9] B. Wei, R. Tarver, J.-S. Kim, and K. Ng, "A single chip Lempel–Ziv data compressor," in Proc. IEEE ISCAS, May 1993, pp. 1953–1955.

[10] S. Panchanathan and M. Goldberg, "A content-addressable memory architecture for image coding using vector quantization," IEEE Trans. Signal Process., vol. 39, no. 9, pp. 2066–2078, Sep. 1991.

[11] T. Juan, T. Lang, and J. Navarro, "Reducing TLB power requirements," in Proc. Int. Symp. Low Power Electron. Des., Aug. 1997, pp. 196–201.

[12] Y.-J. Chang and Y.-H. Liao, "Hybrid-type CAM design for both power and performance efficiency," IEEETrans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 8, pp. 965–974, Aug. 2008.

[13] Z. Lei, H. Xu, D. Ikebuchi, H. Amano, T. Sunata, and M. Namiki, "Reducing instruction TLB's leakage power consumption for embedded processors," in Proc. Int. Green Comput. Conf., Aug. 2010, pp. 477–484.

[14] S.-H. Yang, Y.-J. Huang, and J.-F. Li, "A low-power ternary content addressable memory with Pai-Sigma matchlines," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 10, pp. 1909–1913, Oct. 2012.

[15] N. Onizawa, S. Matsunaga, V. C. Gaudet, and T. Hanyu, "Highthroughput low-energy content-addressable memory based on self-timed overlapped search mechanism," in Proc. Int. Symp. Asynchron. Circuits Syst., May 2012, pp. 41–48.

[16] C.-S. Lin, J.-C. Chang, and B.-D. Liu, "A low-power precomputationbased fully parallel content-addressable memory," IEEE J. Solid-State Circuits, vol. 38, no. 4, pp. 654–662, Apr. 2003.

[17] 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.

[18] 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.

[19] V. Gripon and C. Berrou, "Nearly-optimal associative memories based on distributed constant weight codes," in Proc. ITA Workshop, Feb. 2012, pp. 269–273.

[20] H. Jarollahi, N. Onizawa, V. Gripon, and W. J. Gross, "Architecture and implementation of an associative memory using sparse clustered networks," in Proc. IEEE ISCAS, Seoul, South Korea, May 2012, pp. 2901–2904.

[21] H. Jarollahi, N. Onizawa, V. Gripon, and W. J. Gross, "Reducedcomplexity binary-weight-coded associative memories," in Proc. IEEE ICASSP, May 2013, pp. 2523–2527.

[22] H. Jarollahi, N. Onizawa, and W. J. Gross, "Selective decoding in associative memories based on sparse-clustered networks," in Proc. IEEE Global Conf. Signal Inf. Process., Dec. 2013, pp. 1270–1273.

[23] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proc. Nat. Acad. Sci. USA, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[24] H. Jarollahi, V. Gripon, N. Onizawa, and W. J. Gross, "A low-power content-addressable memory based on clustered-sparse networks," in Proc. 24th IEEE Int. Conf. ASAP, Jun. 2013, pp. 305–308.

[25] H. Noda et al., "A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 245–253, Jan. 2005.

[26] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in Proc. IEEE Custom Integr. Circuits Conf., Sep. 2003, pp. 383–386.

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

[28] H. Noda et al., "A 143 MHz 1.1 W 4.5 Mb dynamic TCAM with hierarchical searching and shift redundancy

architecture," in Proc. IEEE ISSCC, vol. 1. Feb. 2004, pp. 208–523.

[29] C. Zukowski and S.-Y. Wang, "Use of selective precharge for lowpower on the match lines of content-addressable memories," in Proc. Int. Workshop Memory Technol., Des. Test., Aug. 1997, pp. 64–68.

[30] J.-S. Wang, H.-Y. Li, C.-C. Chen, and C. Yeh, "An AND-type matchline scheme for energy-efficient content addressable memories," in IEEE ISSCC Dig. Tech. Papers, vol. 1. Feb. 2005, pp. 464–610.

[31] M. Motomura, J. Toyoura, K. Hirata, H. Ooka, H. Yamada, and T. Enomoto, "A 1.2-million transistor, 33-MHz, 20-b dictionary search processor (DISP) ULSI with a 160-kb CAM," IEEE J. Solid-State Circuits, vol. 25, no. 5, pp. 1158–1165, Oct. 1990.

[32] K. Schultz and P. Gulak, "Fully parallel integrated CAM/RAM using preclassification to enable large capacities," IEEE J. Solid-State Circuits, vol. 31, no. 5, pp. 689–699, May 1996. [35] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in Proc. IEEE Custom Integr. Circuits Conf., Sep. 2003, pp. 383–386.