# High Performance and Energy Efficient SRAM-based Architecture for TCAM

Mahalakshmi R, Gayathri. M
Department of VLSI Design, K. Ramakrishnan, College of Technology, India

**Article Info**
Article history:
Received 13 February 2015
Received in revised form
20 February 2015
Accepted 28 February 2015
Available online 6 March 2015

**Keywords**
Content Addressable Memory (CAM),
Ternary Content Addressable Memory (T-CAM),
Static Random Access Memory (SRAM),
Parameter Extractor

**Abstract**

Content addressable memory (CAM) is a type of solid-state memory in which data is provided as input and an operation is performed to returns the address as output. Unlike other memories (SRAM, DRAM), it performs the search operation in parallel on complete location at once, offered significant reduction in searching time. A ternary content addressable memory (TCAM) is a specialized CAM design for fast table lookups and adds masking capabilities by storing and searching a third "don't care" state. Hence, it is attractive for applications such as packet forwarding and classification in network routers. But TCAMs have some limitations such as low storage density, relatively slow access time, low scalability, complex circuitry, and are very expensive, when compared with static random access memories (SRAMs). Consequently, in order to make use of SRAM benefits in TCAMs and conquer the limitations of TCAM, SRAM can be configured with additional logic to behave like ternary CAM memory. This can be achieved by means of hybrid partitioning the TCAM table and mapping it into corresponding SRAM memory location. In proposed method, a Parameter Extractor is used in modified architecture. The purpose of Parameter Extractor is to minimize the number of comparisons during search operation. As the number of comparison operation is reduced, the power consumption of memory operation is reduced considerably, thus providing an efficient architecture.

## 1. Introduction

Content addressable memory is a hardware search engine that can perform faster than software memories for search intensive application. In CAM memory user provides data search the entire location in parallel and returns the matched address in a single clock cycle [1]. Hence it performs fast and reduces the search time for large database. Due to its fast lookup nature, CAM involved in wide applications such as Communication networks, Databases, Data compression and Bridges and routers. In a CAM table we can store and search only 0's and 1's [1]. Hence, its utility is limited to exact-match SEARCH operations.

In general, CAM has three operation modes: READ, WRITE, and COMPARE. In which 'COMPARE' operation is considered as more important than 'READ' and 'WRITE' operations. Fig. 1 shows the conventional CAM design consists of CAM cell with built-in storage registers for storing a search data and output encoder [2]. In which search line drivers used to drive an input data to each search lines (SL and ~SL) and that are searched agenized the stored data in CAM cell. In each rows CAM cells are used to store single bit of data word and CAM cells of each rows are bind with same Match Line (ML)[3]. Each ML's have their own match line sense amplifiers (MLSA) to detect all CAM cells in ML match with searched data. Thus sense amplifier indicates match or mismatch condition during SEARCH operation. The output of CAM is obtained from CAM priority encoder which selects one result from more than one results of matched address.

A CAM operation start with loading of data onto search data registers and all ML's are pre-charge to high state. After that search operation is begins, the search line driver broadcast the search word onto different search lines and each search data compares against the stored data in CAM cell. In which matched ML preserve in high state and even a single bit mismatched ML's are discharge to ground. Finally MLSA sense the matched ML's and then encoder maps the ML of the matching location to its encoded address.

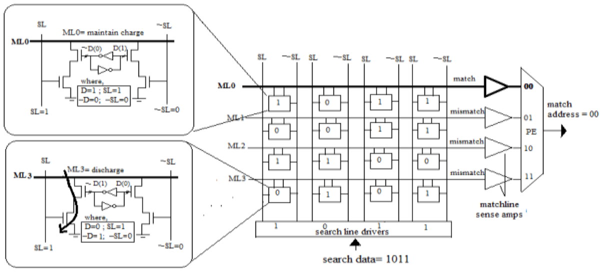

**Fig: 1.** Block Diagram of a Conventional CAM

In each CAM cell cross coupled inverters implement the bit storage nodes D and ~D shown in Fig.1.where comparison operation between complementary stored bit D (and ~D) and complementary search data on complementary search line SL (and ~SL) implemented by NOR cell using four transistors shown in Fig. 1. In which mismatched D and SL (shown in Fig. 1 ML3) activate pulldown path from match line cause discharge of ML to ground indicate mismatch condition. While for matched condition no path was implemented from ML to ground and maintain ML's at high state (shown in Fig. 1 ML0).

In general, CAM classified into two category of: Binary CAM (BCAM) - which is same as conventional

**Corresponding Author,**
**E-mail address:**

CAM, that can stores 0's and 1's and Ternary CAM (TCAM) - which is considered as a specialized CAM design for rapid table lookups, which can store three states of − 0, 1, and x, where x is don't care state (either 0 or 1). Because of this additional state 'x' TCAM provides the additional cost of flexibility to the search. For example, TCAM stored word "110xx1" can match with any of four search words of "110001", "110011", "110101", "110111". The high speed lookup nature of TCAM perform vital role in several application such as networking equipments (high performance routers and switches) to increase the speed of route lookup and packet classification, packet forwarding.

Even though the attractive features of TCAM, some problems degrade the functional usage of TCAM. The main problems discovered are power consumption, complex circuitry, low yield and low scalability [5]. The parallel searching nature of TCAM leads to high power consumption in large size chip and high power consumption degrades chip reliability and affects the chip package cost. For example, an 18–Mb TCAM running at 250 million searches per second (MSPS) consumes 15 W. Many researchers have been proposed several low power techniques for TCAM power consumption reduction. At architectural level, Pagiamtzis et al. proposed Cache-CAM (C-CAM) to reduce the power consumption relative to cache-hit rate but for large hit rate have to increase the cache size and it will be increase the cache power[6]. Lin et al. proposed ones-count pre computation-based CAM (PB-CAM) achieves low-power, low cost, low-voltage, and high-reliability features but requires special memory cell design for reducing power consumption [8].

In this paper, Static Random Access Memory (SRAM) architecture is used to reduce the power consumption of TCAM [7]. In general, compare with random access memories (RAMs), TCAMs have certain limitations such as low storage density, relatively slow access time, low scalability, complex circuitry, and are very expensive. Thus, to use the benefits of SRAM in TCAM, it is configured with additional logic blocks to perform TCAM functionality. Hence in this method, a benefit SRAM is utilized and disadvantage of TCAM is defeated. Thus, RAM based TCAM architecture can offers comparable search performance, scalability, and lower cost than classical TCAM devices, provided that SRAM devices are denser, cheaper, and operate faster than TCAM devices.

The remainder of this paper is organized as follows. In section II, discusses about various literatures regarding the different methods to enhance TCAM performance. In section III, presents the concept of SRAM based TCAM scheme, its comparison with parity bit based approach and its simulation results. In section IV, discusses about the proposed pre-computation based SRAM based TCAM architecture. In section V, concludes the work by describing various observations and scope of future work.

## 2. Literature Review

Many techniques are proposed for reduction of power consumption in CAM. One of the technique is C-CAM architecture [1] concept is similar to that of cache in processor system, where it is used to hold frequently accessed CAM data. During a search operation the matched result from cache saves power by avoiding a search in the large and higher-power CAM. Using cache with CAM offers increase in performance and decrease in power consumption depends on the cache hit rate. The implementation of C-CAM depends upon the cache size and the cache hit rate. The CAM power decreases with the cache size, due to the increased cache hit rate and thus less frequent access to the CAM. The cache-CAM architecture provides moderate level power consumption reduction in high density CAM but use of these caches requires major modifications to the memory structure and hierarchy to fit the design. Even though in reduction of power due to small number of memory access, the area utilization of memory is large.

Another architecture proposed is pre computation method [2], additional bits are derived from stored word and these bits are used in initial search before going to search main word. If the initial search fails, then it terminates the main word search and thus saves power. The major problem arises for long data word is that complex computation scheme uses large number of full adders and need extra silicon area to store additional bit. This method achieves low-power, lowcost, low-voltage, and high-reliability features. Although this method improves the efficiency of PB-CAMs, it requires considerable modification to the memory architecture in order to achieve high performance. Therefore, it is beyond the capacity of the general CAM design. Moreover, the disadvantage of the ones count PB-CAM system is that it adopts a special memory cell design for reducing power consumption. On the contrary, as the input bit length increases, the delay of the ones-count parameter extractor also increases significantly.

Block-XOR approach [3] reduces the power consumption by reducing the number of comparison operation. The comparison operation reduced by using parameter extractor that extracts the parameter from input data, which is then compared with parameters stored in the parameter memory. If matches occur in first part then data related to those matched parameters will be compared in second part; otherwise no match condition is returned. The design goal of the parameter extractor is to filter out as many unmatched data as possible to minimize the required number of comparison operations in the second part. The first part output is computed from XOR logic operation in each partitioned input data block, and the resultant bits are combined to become the input parameter for the second part of the comparison process. The outcome result is bit '1' for odd number of one's in data word and '0' for even number of ones. It is interesting that XOR- ing can be performed over arbitrary positioned bits instead of adjacent ones. Block-XOR PB-CAM becomes more effective in reducing the number of comparison operations, as the input bit length increases with smaller constant delay. In this scheme only parity of 1s in the data word is checked. Therefore, single bit area is enough to store additional bit and that ease the computation process.

One of the best techniques of hashing [4] offers higher performance than other techniques, especially when a large database is used. In this technique searchable data items (or records) contain two fields: key and data. The goal of search is to discover a record associate with key in database. The quick search operation is achieved by the simple arithmetic function h(.) ( hash function) on key, that offers the direct detection of associated record location in memory. However to build CAM from RAM using hash technique suffers by problems of collisions and bucket overflow. These problems cause increase in search lookup. If don't care bits in stored key used for hashing, need high capacity because of duplication of such key in multiple buckets must occurred. Thus access of multiple buckets causes performance degradation. Furthermore, it emulates binary CAM, not TCAM. Thus, hashing cannot provide deterministic performance owing to potential collisions and it is inefficient in handling wildcard.

The low power technique of parity bit based CAM design [5] is similar to that of pre-computation based CAM design (PB-CAM) where extra one bit segment derived from the stored data bit held along with original data segment. The parity bit can be either odd or even. Thus the new architecture has the same interface as the conventional CAM with one extra bit. The match condition occurs when the parity bit of search and stored word is the same, thus returns a match. The mismatch condition may occur either when parity bit of stored and search bit is different, or when parity bit of search and stored is same but no match found in stored words. Therefore driving capacity of more mismatch word is stronger than that of the 1-mismatch word, the design greatly improves the search speed. This parity bit does not improve the power performance. However, this additional parity bit reduces the sensing delay and boosts the driving strength of the 1-mismatch case (which is the worst case) by half.

Another architecture method of architecture imitates TCAM functionality with SRAM memory [6]. A parity bit was used to boost the searching speed with less power and delay. These parity bits are used to find the matched word and reduce the comparison with the mismatched word, so this method has low power consumption than existing methods. Initially parity bit of each stored word is pre computed along with stored word [9]. When input search word is given, the parity of that input word is calculated, so the input for example is odd parity '1', and then it is compared only with stored word that contains odd parity, so the number of comparison is reduced. This architecture is a low power TCAM with high speed, it support large input patterns with much simpler design structure and shorter propagation delay.

## 3. SRAM based Architecture for TCAM

### A) Hybrid Partitioning of TCAM Table

Hybrid partitioning (HP) is a collective name given to vertical partitioning and horizontal partitioning of the conventional TCAM table. An example of HP is given in Table I HP partitions conventional TCAM table vertically (column wise) and horizontally (row wise) into TCAM subtables, which are then processed to be stored in their corresponding memory units. Vertical partitioning (VP) implies that a TCAM word of C bits is partitioned into N subwords; each subword is of w bits. VP is used in SRAM based TCAM Architecture to decrease memory size as much as possible.

Horizontal partitioning (HrP) divides each vertical partition using the original address range of conventional TCAM table into L horizontal partitions. HP results in a total of L × N hybrid partitions. The dimensions of each hybrid partition are K × w where K is a subset from original addresses and w is the number of bits in a subword. Hybrid partitions spanning the same addresses are in the same layer. For example, HP21 and HP22 span the same address range and are in layer 2.

**Table: 1.** Traditional TCAM Table & Its Hybrid Partitions (HP)

| Address | Ternary data | | Layer |
|---|---|---|---|
| 0<br>1 | 00<br>01 HP11 | 11<br>01 HP12 | 1 |
| 2<br>3 | 0*x*<br>11<br>HP21 | 11<br>1*x* HP22 | 2 |

**Table: 2.** TCAM Example: Data Mapping

| **Address** | **$VM_{21}$** | **$VM_{22}$** | **OATA $M_{21}$** | **OATA $M_{22}$** | **Original Address** | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | **$OAT_{21}$** | | **$OAT_{22}$** | |
| | | | | | **2** | **3** | **2** | **3** |
| 0 | 1 | 0 | 0 | - | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | - | 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | - | 0 | 0 | 1 | 0 | 0 |
| 3 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 |

### B) Architecture of SRAM Based TCAM Architecture

### 1. Overall Architecture

The overall architecture of SRAM based TCAM is depicted in Fig.2 where each layer represents the architecture shown in Fig.3. It has L layers and a CAM priority encoder (CPE) [9]. Each layer outputs a potential match address (PMA). The PMAs are fed to CPE, which selects match address (MA) among PMAs.

### 2. Layer Architecture

Fig.3 shows the layer architecture of SRAM BASED TCAM ARCHITECTURE. Since the total number of vertical partitions is N in a layer, to accommodate data in N vertical partitions, N validation memories (VMs) and N original address tables (OATs) are required[10]. To achieve the TCAM functionality, the layer architecture is further

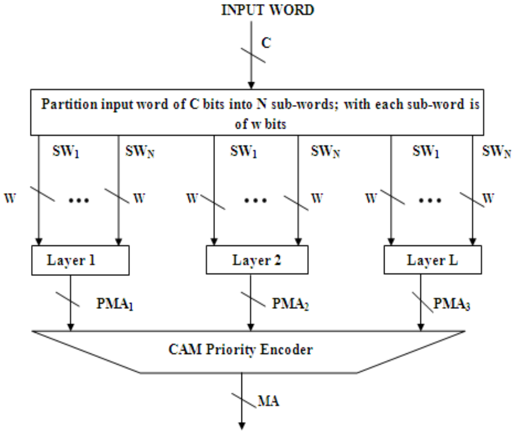

**Fig: 2.** SRAM based TCAM. Architecture (sw: subword, C:# of bits in the input word, PMA: potential match address and MA: match address).

equipped with 1-bit AND operation, K-bit AND operation, and a layer priority encoder (LPE). Each hybrid partition in a layer has its corresponding VM and OAT. For instance, HP11 has its corresponding VM11 and OAT11. The collective functionality of 1-bit ANDing, K-bit ANDing, and priority encoding constitutes additional logic.

Validation Memory-Size of each VM is 2w × 1 bits where w represents the number of bits in each subword and 2w shows the number of rows. A subword of w bits implies that it has total combinations of 2w where each combination represents a subword. For example, if w is of 4 bits, then it means that there are total of 24 = 16 combinations. This explanation is also related to OATAM and OAT. Each subword acts as an address to VM. If the memory location be invoked by a subword is high, it means that the input subword is present, otherwise absent. Thus, VM validates the input subword, if it is present. For example, Table 3.2 shows that subwords 00, 01, and 11 are mapped in VM21. This states that memory locations 00, 01, and 11 should be high in VM21 and the remaining memory Locations are set to low because their corresponding subwords do not exist.

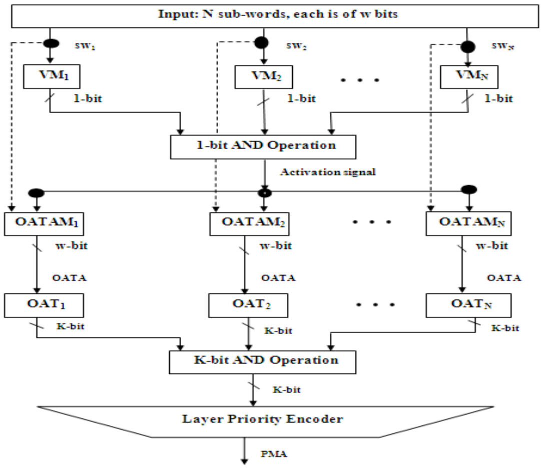

**Fig: 3.** SRAM based TCAM architecture layer. (sw: subword, VM: validation memory, OATAM: original address table address memory, OAT: original address table, and LPE: layer priority encoder).

1-bit AND Operation- It performs the ANDs the output of all VMs. The output of 1-bit AND operation decides the continuation of a search operation. If the result of 1-bit AND operation is high, then it permits the continuation of a search operation, otherwise mismatch occurs in the corresponding layer.

Original Address Table Address Memory (OATAM) - Each OATAM is of 2w × w bits where 2w is the number of rows and each row has w bits. In OATAM, an address is stored at the memory location indexed by a subword and that address is then used to invoke a row from its corresponding OAT. If a subword in VM is mapped, then a corresponding address is also stored in OATAM at a memory location accessed by the subword. For example, Table II shows OATAM21 where addresses are stored at the memory locations 00, 01, and 11. The output of OATAM is called as OATA [11]. Hyphen "-" indicates that the corresponding memory location has no data because the corresponding subword for the memory location is not present in VM.

Original Address Table- Dimensions of OAT are 2w× K where w is the number of bits in a subword, 2w represents number of rows, and K is the number of bits in each row where each bit represents an original address. Here K is a subset of original addresses from conventional TCAM table[12]. It is OAT, which considers the storage of original addresses. An example of OAT is given in Table II.

K-bit AND Operation- The bit-by-bit AND operation is carried out in OAT stored value. Here, it read outs the K-bit rows from all OATs and forwards the result to LPE [14]. The resultant value is used to detect the correct address location of given input data. The AND operation result is indicates the correct address by using the logic 1.

Layer Priority Encoder- When multiple matches occurs in TCAM, and then the particular address location is selected by LPE. The LPE is selects PMA among the outputs of K-bit AND operation.

### C) SRAM based TCAM architecture Operations

Data Mapping Operation- Classical TCAM table is logically partitioned into hybrid partitions. Each hybrid partition is then expanded into a binary version. Thus, we first expand x into states 0 and 1 to be stored in SRAM. For example, if we have a TCAM word of 010x, then it is expanded into 0100 and 0101. Each subword, acting as an address, is applied to its corresponding VM and logic "1" is written at that memory location. The same subword is also applied to its respective OATAM and w bits data are written at that memory location. During search, these w bits data act as an address to the OAT. The K bits data are also written at the memory location in OAT determined by its corresponding OATA. Thus, in this way, all hybrid partitions are mapped.

A subword in a hybrid partition can be present at multiple locations. So, it is mapped in its corresponding VM and its original address (es) is/are mapped to its/their corresponding bit(s) in its respective OAT. Since a single bit in OAT represents an original address, only those memory locations in VMs and address positions/ original addresses in OATs are high, which are mapped while remaining memory locations and address positions are set to low in VMs and OATs, respectively. Example of data mapping is shown in Table II and use Table I to be mapped to SRAM based TCAM architecture. We take N = 2, L = 2, K = 2, and w = 2. After necessary processing, HP11, HP12, HP21, and HP22 are mapped to their corresponding memory units.

Search Operation- Searching in a Layer of SRAM based TCAM architecture, N subwords are concurrently applied to a layer. The subwords then read out their corresponding memory locations from their respective VMs. If all VMs validate their corresponding subwords (equivalent to 1-bit AND operation in Fig. 3), then searching will continue, otherwise mismatch occurs in the layer. Upon validation of all subwords, the subwords read out their respective memory locations from their corresponding OATAMs concurrently and output their corresponding OATAs [15]. All OATAs then read out K-bit rows from their corresponding OATs simultaneously, which are then bitwise ANDed. LPE selects PMA from the result of the K-bit AND operation.

Searching in SRAM based TCAM architecture, Search key is applied to SRAM based TCAM Architecture, which is then divided into N subwords. The input subwords are forward to all layers of SRAM based TCAM architecture in parallel. The search operation is performed in each layer and the resultant PMAs are available from all layers. CPE selects MA among PMAs; otherwise a mismatch of the input word occurs.

**Table: 3.** Example of a Search Operation in SRAM based TCAM Architecture

| **Steps** | **Activity** |
|---|---|
| 1 | Search Key = 0011 |
| 2 | Sub-word$_1$ = 00<br>Sub-word$_2$ = 11 |
| 3 | PMA$_1$ = 0<br>PMA$_2$ = 2 |
| 4 | CAM Priority Encoder Selects Address 0 as MA |

Table III shows the overall search operation in SRAM based TCAM architecture. We use input word 0011 to be searched. Then the input word is partitioned into two subwords of 00&11. The search operation in layers finds the matched layer as Layer 1 and the result of address 0 is obtained from last content addressable memory priority encoder.

## 4. Comparison between SRAM based TCAM architecture & SRAM based TCAM architecture with PARITY BIT

SRAM based TCAM Architecture has higher power dissipation, as the number of comparison operation is more before validating particular search key is present in existing layers or not. So in order to reduce the number of comparison operation several pre-computation based methods like Parity Bit based approach, One's count approach and Block- XOR approach were introduced [15]. Here SRAM based TCAM Architecture is compared with Parity Bit based approach in terms of power, delay and LUTs.

Parity Bit based Approach for SRAM based TCAM Architecture- This architecture has the same interface as the SRAM based TCAM with one extra bit. SRAM based TCAM Architecture with a parity bit design consisting of the original data segment and an extra one bit segment derived from the actual data bits. Parity bit is either odd or even. The extractor is used to find the parity bit value. During the search operation, the matched parity bit value of word is found first. Only the word whose parity bit value matched will be compared with the search word and reduce the comparison with the mismatched word. The new architecture has the same interface as the SRAM based TCAM Architecture with one extra bit. Therefore driving capacity of more mismatch word is stronger than that of the 1-mismatch word, the design greatly improves the search speed. This parity bit does not improve the power performance.

However, this additional parity bit reduces the sensing delay and boosts the driving strength of the 1-mismatch case (which is the worst case) by half. The additional equipment required for parity generation. In this architecture parameter extractor and parameter memory are used to generate and store the additional parity bit of stored and input data. The parameter extractor is used extract the input subwords and find out the parity bit (either 1 or 0). The parameter memory is used to store the stored words parity bit and result from parameter extractor is compared with it to determine the selective data for search. Thus the number of comparison is reduced to sensing delay. In Parity Bit based approach, if any word has don't care condition, then for both parity condition, we have to compare that word. If there is less number of words having don't care, then there will reduction in comparison operation and power. Table V derived to Compares the SRAM based TCAM Architecture and SRAM based TCAM Architecture with a Parity Bit based on parameters LUT, Delay and Power.

**Table: 5.** Comparison between SRAM based TCAM Architecture and SRAM based TCAM Architecture with a Parity Bit

| Case (L,N) | Method | LUTs | Delay (ns) | Power (mW) |
|---|---|---|---|---|
| **64 X 32** | | | | |
| (2,4) | Z-TCAM | 380 | 27.194 | 109.99 |
| | Z-TCAM with Parity | 364 | 16.045 | 110.08 |
| (4,4) | Z-TCAM | 393 | 15.568 | 112.05 |
| | Z-TCAM with Parity | 427 | 18.702 | 112.15 |
| **512 X 36** | | | | |
| (2,4) | Z-TCAM | 1606 | 21.587 | 118.06 |
| | Z-TCAM with Parity | 1654 | 17.647 | 118.15 |
| (4,4) | Z-TCAM | 2140 | 13.336 | 127.47 |
| | Z-TCAM with Parity | 2198 | 15.535 | 127.49 |

## 5. Conclusion

SRAM based architecture for TCAM is presented and its outputs were simulated and verified using Xilinx 13.1 software. It is noted that latency of this architecture is more but it can be easily compromised as throughput is achieved. The power dissipation of SRAM based TCAM Architecture is measured using Xilinx Xpower Analyzer tool, from this report it is revealed that the power dissipation of SRAM based TCAM Architecture is high, as the number of comparison operation is more. So, in order to shrink the power dissipation, parity bit based approach was used along with SRAM based TCAM Architecture and compared, but it fails in certain cases. As TCAM plays a major role nowadays to perform high speed lookup/search operation in many fields like router lookup table, an efficient approach is required to reduce the power consumption, therefore in future to develop using pre-computation based approach of bank selection scheme with compression technique and implement that on FPGA. To validate this work any application like TCAM based network router table or data compressor will be established.

## References

[1] Anh-Tuan Do, Shoushun Chen, Zhi-HuiKong, KiatSeng Yeo, A High Speed Low Power CAM with A Parity Bit and Power- Gated ML sensing, IEEE Transaction Very Large Scale Integration (VLSI) Systems, 21(1), 2013

[2] B. Agrawal, T. Sherwood, Modeling TCAM power for Next Generation Network Devices, IEEE International Symposium Performance Analysis of Systems and Software (ISPASS), 2006

[3] Cho S, Martin J, Xu R, M. Hammoud, R. Melhem, CA-RAM: A high-performance memory substrate for search-intensive applications, IEEE International Symposium Performance Analysis Syst. Softw, 2007, 230-241

[4] D. Georgiev, Low power concept for Content Addressable Memory (CAM) chip Design, International Journal of Advanced Research in Electrical, Electronics & Instrumentation Engineering, 2(7)

[5] Kostas Pagiamtzisand, Ali Sheikholeslami, Using Cache to reduce power in Content-Addressable Memories (CAMs), IEEE CUSTOM Integrated Circuits Conference, 2005

[6] Kostas Pagiamtzis, Ali Sheikhloeslami, Content-addressable memory (CAM) circuits and architectures: A tutorial and suryey, IEEE J. Solid State Circuits, 41(3), 2006, 712-727

[7] P. Mahoney, Y. Savaria, G. Bois, P. Plante, Parallel hashing memories: An alternative to Content Addressable Memories, International IEEE-NEWCAS Conference, 2005, 223-226.

[8] M. M. Remya, E. Amrutha, A high performance area efficient Low Power CAM architecture Design, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), 4(2) 1, 2014, 26-32

[9] S. J. Ruan, C.Y. Wu, J. Y. Hsieh, Low power design of pre-computation based content-addressable memory, IEEE Transactions Very Large Scale Integration (VLSI) Systems, 16(3), 2008, 331-335

[10] S. Pasumarti, L. Kadali, Swathi. Bankapalli, Design of Parameter Extractor in Low Power Precomputation based Content Addressable Memory, International Journal on Computer Science and Engineering (IJCSE) 3(7), 2011

[11] M. Somasundaram, Memory and power efficient mechanism for fast table lookup, In: U.S. Patent 7155563, 2006

[12] T. Thushara, H. S. Jacob, A Low Power Hybrid Partitioning SRAM based TCAM with a Parity Bit, International Journal of Innovative research in Advanced Engineering (IJIRAE), 2014

[13] K. Yuvana, P. Balavenkateswarlu, Mohammed T. Abbas, S. Ashok, Analysis and Design of Low Power Content Addressable Memory (CAM) Cell, International Journal of Emerging Technology and Advanced Engineering, 3(12), 2013

[14] Zahid Ullah, Sanghyeon Baeg, Vertically partitioned SRAM-based Ternary Content Addressable Memory, IACSIT International Journal of Engineering and Technology 4(6), 2012

[15] Zahid Ullah, Manish K. Jaiswal, Ray C. C. Cheung, Z-TCAM: An SRAM based- Architectures for TCAM, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014