

## Banked Approach of Low Power Design of Pre-Computation Based Content Addressable Memory

Rafeekha M. J <sup>1</sup>, V. Lakshmi Narasimhan<sup>2</sup>

<sup>1</sup>PG Scholar, Department of Electrical and Electronics Engineering, Hindusthan College of Engineering and Technology, Coimbatore 641032, Tamilnadu, India

<sup>2</sup>Assistant Professor, Department of Electrical and Electronics Engineering, Hindusthan College of Engineering and Technology, Coimbatore 641032, Tamilnadu, India

### ABSTRACT

Content-addressable memory (CAM) or associative memory is used in applications that require large amount of data transfer in less time. It is a storage device that is addressed by its contents. It is able to perform look-up table function in a single clock-cycle. They are mainly used in network routers for packet forwarding and packet classification. But the parallel comparison technique used in CAM reduces search time, but it increases power consumption. The main challenge in the design of CAM is the reduction in power consumption. This paper presents a banked approach to improve the efficiency of low power precomputation-based CAM (PB-CAM). It is simulated using modelsim. The experimental results show that the Banked PB-CAM system can achieve greater power reduction without the need for a special CAM cell design. This implies that approach is more flexible and adaptive for general designs.

**Keywords:** Associative memory, Content-addressable memory (CAM), low-power, PB-CAM, precomputation.

### 1. 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. Fig 1 shows the comparison between a traditional memory and a content addressable memory. In a traditional memory input is the address of the memory location that we are interested and output will be the content of that address. In CAM it is the reverse. A content-addressable memory is a critical device used for applications involving asynchronous transfer mode (ATM), communication networks, LAN bridges/switches, databases, lookup tables, and tag directories, due to its high-speed data search capability. A CAM is a functional memory with a large amount of stored data that simultaneously compares the input search data with the stored data. The vast number of comparison operations required by CAMs consumes a large amount of power [1].

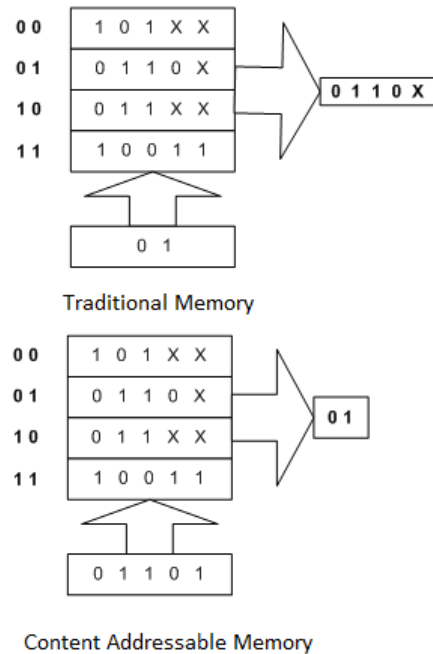


Fig 1: Comparison between traditional memory and content addressable memory.

Since CAM is an outgrowth of Random Access Memory (RAM) technology, in order to understand CAM, it helps to contrast it with RAM. A RAM is an integrated circuit that stores data temporarily. Data is stored in a RAM at a particular location, called an address. In a RAM, the user supplies the address, and gets back the data. The number of address line limits the depth of a memory using RAM, but the width of the memory can be extended as far as desired. With CAM, the user supplies the data and gets back the address. The CAM searches through the memory in one clock cycle and returns the address where the data is found. The CAM can be preloaded at device startup and also be rewritten during device operation. Because the CAM does not need address lines to find data, the depth of a memory system using CAM can be extended as far as desired, but the width is limited by the physical size of the memory.

CAM can be used to accelerate any application requiring fast searches of data-base, lists, or patterns, such as in image or voice recognition, or computer and communication designs. For this reason, CAM is used in applications where search time is very critical and must be very short. For example, the search key could be the IP

address of a network user, and the associated information could be user's access privileges and his location on the network. If the search key presented to the CAM is present in the CAM's table, the CAM indicates a 'match' and returns the associated information, which is the user's privilege. A CAM can thus operate as a data parallel or Single Instruction/Multiple Data (SIMD) processor [2].

Content Addressable Memory (CAM) or associative memory, is a storage device, which can be addressed by its own contents. Each bit of CAM storage includes comparison logic. A data value input to the CAM is simultaneously compared with all the stored data. The match result is the corresponding address. A CAM operates as a data parallel processor. CAMs can be used to design Asynchronous Transfer Mode (ATM) switches. Implementing CAM in ATM applications are specifically described in this application note. As a reference, the application note XAPP201 "An Overview of Multiple CAM Designs in Virtex™ Devices" presents diverse approaches to implement CAM in other designs.

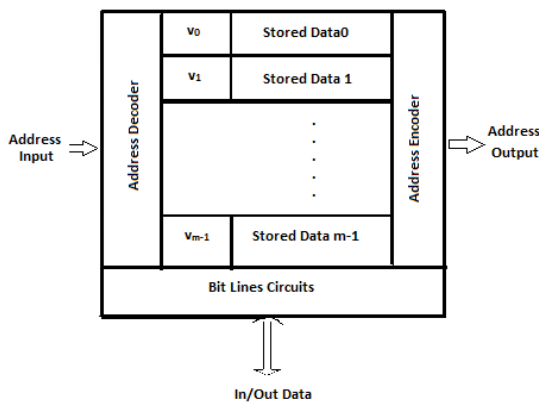


Fig 2: General CAM Architecture

Fig 2 shows the general CAM architecture. It consists of data memory with valid bit field, address decoder, and address priority encoder. The valid bit field indicates the availability of stored data. In the data searching operation, the input data is sent into CAM to compare with all valid data stored in CAM simultaneously, and an address from among those matches of comparison is sent to the output. In this architecture, the CAM circuit performs large amount of comparison operations to identify all valid data stored in CAM during each data searching operation. This comparison consumes most of the total CAM power. In the past decade, much research on energy reduction has focused on the circuit and technology domains provide a comprehensive survey on CAM designs from circuit to architectural levels. Several works on reducing CAM power consumption have focused on reducing match-line power [3].

Although there has been progress in this area in recent years, the power consumption of CAMs is still high compared with RAMs of similar size. At the same time, research in associative cache system design for power efficiency at the architectural level continues to increase. The

filter cache and location cache techniques can effectively reduce the power dissipation by adding a very small cache. However, the use of these caches requires major modifications to the memory structure and hierarchy to fit the design.

Pagiartzis *et al.* proposed a cache-CAM (C-CAM) that reduces power consumption relative to the cache hit rate. Lin *et al.* presented a ones-count pre computation-based CAM (PB-CAM) that achieves low-power, low cost, low-voltage, and high-reliability features. Although Cheng further improved the efficiency of PB-CAMs, the approach proposed requires considerable modification to the memory architecture to achieve high performance [8]. Therefore, it is beyond the scope 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, which is only applicable to the ones count parameter extractor.

This paper deals with banked approach for reducing comparison operations in the second part for the PB-CAM. This approach employs a brand new parameter extractor, which can better reduce the comparison operations required than the ones-count approach. Hence this reduces the power consumption by reducing comparison operations. The BANKED APPROACH that has been presented is suitable for applications demanding a large size of the storage device while the performance is still required. Architecture employs the precomputed parameter to perform a power-aware ordering of the data.

## 2. System Architecture

The memory organization of the PB-CAM architecture proposed by Lin *et al.*, which consists of data memory, parameter memory, and parameter extractor, where  $k \ll n$ . To reduce massive comparison operations for data searches, the operation is divided into two parts. In the first part, the parameter extractor extracts a parameter from the input data, which is then compared to parameters stored in parallel in the parameter memory. If no match is returned in the first part, it means that the input data mismatch the data related to the stored parameter. Otherwise, the data related to those stored parameters have to be compared in the second part [10].

It should be noted that although the first part must access the entire parameter memory, the parameter memory is far smaller than that of the CAM (data memory). Moreover, since comparisons made in the first part have already filtered out the unmatched data, the second part only needs to compare the data that match from the first part. The PB-CAM exploits this characteristic to reduce the comparison operations, thereby saving power. Therefore, the parameter extractor of the PB-CAM is critical, because it determines the number of comparison operations in the second part [12].

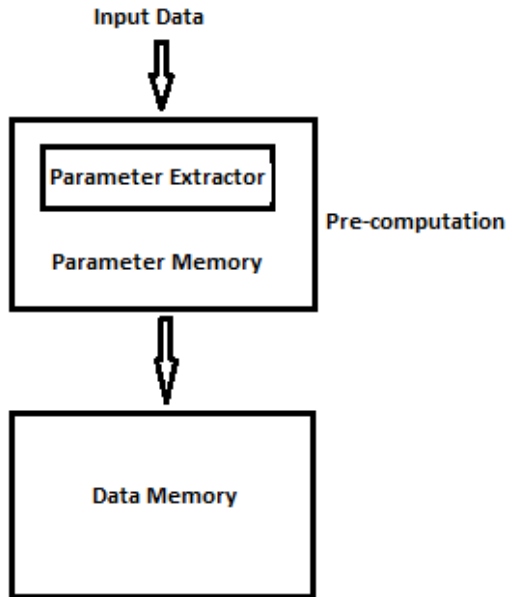


Fig. 3: Basic PB-CAM architecture

**2.1. 1's count PB- CAM**

Recently pre-computation technique has received as one of the most effective approaches for low-power designs. Pre-computation –based CAM (PB-CAM) stores extra information along with data used in the data searching operation to eliminate most of the unnecessary comparison operation, thereby saving power [14].

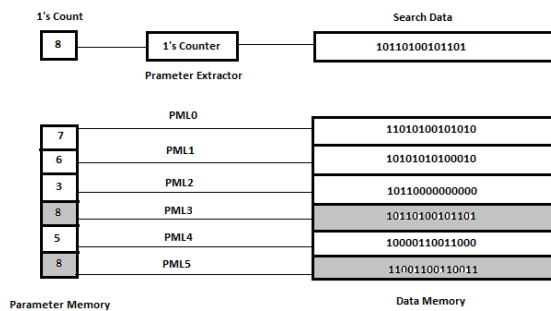


Fig 4: 1's Count Parameter Extraction for a 14 Bit Data

The total number of CAM cell comparisons in 1's count approach is equal to  $m \times \log(n+2) + (m \times n) / (n+1)$ , where  $m$  is the number of words and  $n$  is the number of bits in the word. The traditional CAM architecture has  $(m \times n)$  CAM cell comparisons and it is known that  $m \times (\log(n+2) + 1) \leq (m \times n)$  for  $n > 4$  therefore the PB CAM architecture consumes less comparison power than traditional CAM architecture. The ones count parameter extractor is implemented with many full adders, so it consumes huge power and hardware consumption which not only wastes area but increases delay. And also the cost is high. To overcome this Block-XOR circuit is designed.

**3. Block- XOR Approach**

**3.1. Block-XOR PB-CAM Structure**

In this approach, the input data bit is first partitioned into several blocks, from which an output bit is computed using XOR logic operation for each of these blocks. The output bits are then combined to become the input parameter for the second part of the comparison process. To compare with the ones-count approach, then set the bit length of the parameter to  $d \log(n+2)e$ , where  $n$  is the bit length of the input data. Therefore, the number of blocks is  $dn = \log(n+2)e$  in this approach. Taking the 14-bit input length as an example, the bit length of the parameter is

$$\log(14 + 2) = 4\text{-bit, and the number of blocks is } d14 = \log(14 + 2)e = 4.$$

The selected signal is defined as  $S = P3P2P1P0$ :  
 (2) According to (2), if the parameter is "0000\_1110" ( $S = \setminus 0$ ), the multiplexer will transmit the  $i0$  data as the output. In other words, the parameter does not change. Otherwise, ( $P3P2P1P0 = \setminus 1111$ ,  $S = \setminus 1$ ), the first block of the input data becomes the new parameter, and "1111" can then be used as the valid bit. Note that the case where the first block is "1111" was not considered, because the "1111" block is the parameter.

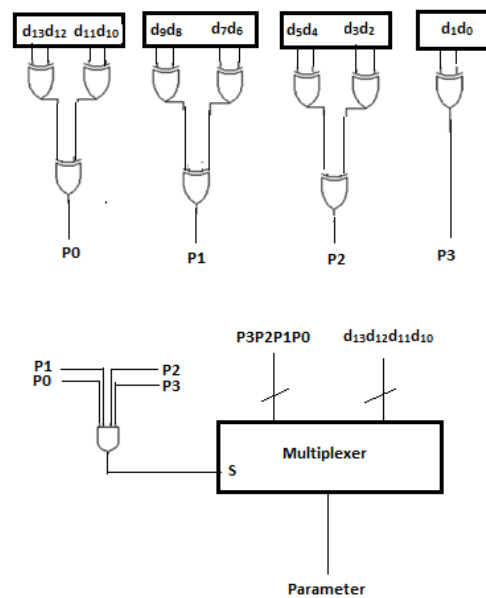


Fig. 5: Structure of Block-XOR approach with valid bit.

**3.2. Mathematical Analysis**

The concept of Block-XOR approach is to uniformly distribute the parameter over the input data. By the rule of product, the number of input data that results in the same parameter (without valid bit) is  $8 \times 8 \times 8 \times 2 = 1024$ . Consequently, the average probability can be determined as  $1024 = (1024 \times 16) \times 100\% = 6:25\%$ . Accordingly, the maximum number of comparison operations is 1024 for each parameter in the second part. Obviously, the concept of Block-XOR approach can reduce the comparison operations, hence minimize power consumption (i.e., with valid bit).

When the parameter is “1111”, the new parameter is provided by the first block with an output bit of “1” so that the number of input data for those parameters is  $1024 + (1024 \times 8) = 1152$ , and the average probability is  $(1152 = (1024 \times 7 + 1152 \times 8)) \times 100\% = 7.03\%$ . Block-XOR PB-CAM results in at least 850 fewer comparison operations in 82% of the cases. In other words, in most cases, the Block-XOR PB-CAM required far [8]. The longer the input bit length is, the fewer the number of comparison operations required (i.e., power reduction). Therefore, the Block- XOR PB-CAM is more suitable for wide-input CAM Applications. In addition, the Block-XOR parameter extractor can compute parameter bits in parallel with three XOR gate delays for any input bit length, hence short constant delay. On the contrary, as the input bit length increases, the delay of the ones-count parameter extractor will increase significantly.

#### 4. Banked Approach

Banked architecture using Block XOR will be one of the most effective techniques to reduce power without compromising hardware. The block-XOR will reduce hardware and power when compare to ones-count. Banked architecture will reduce most of the dissipation power with negligible hardware complexity. Most of the work is carried out by parameter extractor in this architecture hence total work of this banked architecture is reduced and saves energy during searching operations.

One of the main disadvantages in PB-CAM is pseudo power dissipation, which is due to wastage of energy in the match lines during search operation. The first modification is spitted into independent banks with equal no of words per bank. Once work operation is selected search operation is done locally. So that remaining bank will be disabled and it saves power P LSBS used to select the bank and MSB s stored in parameter memory. So that memory is reduced to 5 to 3 bits. Use of banked structure reduces the complexity of logic usage (decoding & encoding matching lines) [9]. Due to the banked implementation of the memory, the operation of the architecture is restricted to just one bank every cycle.

One of the advantages of this banked structure is the reduction of the dynamic power consumption as the charge in the bit lines is limited to one bank (the driven line is simplified to the bit line of the accessed bank of the memory). This behavior is also shown by the parameter lines and also has a positive influence in the memory speed. The complexity of the logic shared for the banks (buffers, priority encoders, and address decoders) is reduced when the bank approach is applied. This simplification saves area, power consumption and improves the delay of these devices.

Banked architecture is based on a parameter pre-computation-based architecture [12] (PB-CAM from now on); however, able to reduce the parameter word’s size with respect to [17], decreasing in this way the logic complexity, area, and power consumption related to this parameter. Moreover, the energy savings obtained with the proposed banked architecture improves the previous implementations

of similar technologies and also improve the scalability capabilities. This architecture shows good results in terms of area and dynamic power consumption [13, 14]. This paper presents an improved architecture with a novel hardware mechanism to reduce the static power consumption and increase the dynamic energy savings with new experimental results and a deeper analysis of the consequences of applying leakage reduction techniques over CAM memories.

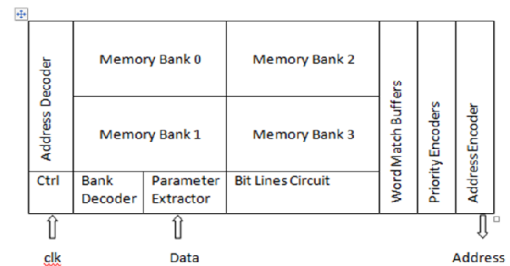


Fig. 6: Structure of Banked approach.

#### 5. Simulation Result

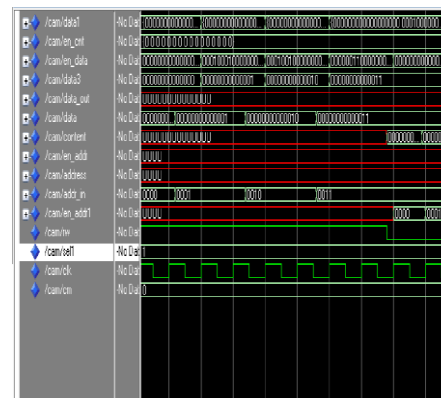


Fig 7. Simulation result of banked PB-CAM

#### 6. Conclusion

The CAM plays a very critical role in many applications, where power consumption is the major limiting factor. The integration levels achieved by current technology process have turned the area and performance factors into secondary

actors. Search based applications with high performance constraints demand efficient implementations of content addressable memories to cover the constraint.

The B.ANKED APPROACH that has been presented is suitable for applications demanding a large size of the storage device while the performance is still required The BLOCK-XOR was found to have an IO utilization of 27% where as in the BANKED approach this has been reduced to 6%, which thereby increases the overall performance.

Table1. Comparison between block-XOR approach and banked approach

|                 | Banked Approach |           | Block- XOR Approach |           |
|-----------------|-----------------|-----------|---------------------|-----------|
|                 | Used            | available | used                | Available |
| IOS utilization | 14              | 232       | 64                  | 232       |
| Utilisation %   | 6               |           | 27                  |           |

The power comparison table of the three approaches is shown in table2. The conventional PB-CAM has an average power consumption of 266.84mW, the Block-XOR PB-CAM has 146.48 mW and the BANKED PB-CAM has 26.79 mW.

Table 2. Power comparison between three approaches.

|                             | PB-CAM   | BLOCK XOR PB-CAM | BANKED PB-CAM |
|-----------------------------|----------|------------------|---------------|
| Technology                  | 0.35µm   | 0.35µm           | 0.35µm        |
| Configuration               | 2048×32  | 2048×32          | 2048×32       |
| Search delay                | 25ns     | 15ns             | 7.5ns         |
| Average power               | 266.84mW | 146.48mW         | 26.79mW       |
| Average power reduction (%) |          | 45               | 81.7          |

## 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] M. Tanaka, H. Miyatake, 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.
- [3] 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.
- [4] 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.
- [5] Y. J. Chang, F. Lai, and S. J. Ruan, "Design and analysis of low power cache using two-level filter scheme," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 4, pp. 568–580, Aug. 2003.
- [6] S. Bhattacharyya, T. Srikanthan, and Vivekanandarajah, "Dynamic filter cache for low power instruction memory hierarchy," in *Proc. Euromicro Symp. Digit. Syst. Des.*, Sep. 2004, pp. 607–610.
- [7] Y. Hu, W. B. Jone, and R. Min, "Location cache: A low-power L2 cache system," in *Proc. Int. Symp. Low Power Electron. Des.*, Apr. 2004, pp. 120–125.
- [8] K. Pagiamtzis and A. Sheikholeslami, "Using cache to reduce power in content-addressable memories (CAMs)," in *Proc. IEEE Custom Circuits Conf.*, Sep. 2005, pp. 369–372.
- [9] J. C. Chang, B. D. Liu, and C. S. Liu, "A low-power precomputation-based fully parallel content-addressable memory," *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 622–654, Apr. 2003.
- [10] K. H. Cheng, S. Y. Jiang, and C. H. Wei, "Static divided word matching line for low-power content addressable memory design," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2004, vol. 2, pp. 23–26.
- [11] Anh Tuan Do, Shoushan Chen, Zhi-Hui Kong, and Kiat Seng Yeo, "A Low power CAM with Efficient Power and Delay Trade-off," *Circuits and Systems, 2011 IEEE International Symposium*, pp. 2573–2576, 2011.
- [12] Echeverria, P.; Ayala, J. L.; Lopez-Vallejo, M.; "Leakage Energy Reduction in Banked Content Addressable Memories", *Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference*, pp. 1196–1199, Dec. 2006.
- [13] Echeverria, P.; Ayala, J. L.; Lopez-Vallejo, M.; "A banked precomputation-based CAM architecture for low-power storage-demanding applications," *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*, pp. 57–60, May 2006.
- [14] Eshraghian, Kyoungrok Cho, Soon Ku Kang, Abbott D., "Mermistor MOS content Addressable Memory: Hybrid architecture for future High Performance Search Engines"; *VLSI Systems IEEE Transactions*, pp. 1407–1417, 2011.
- [15] Jianping Hu, Jinghong Fu, Xiaoyan Luo; "A Single Phase Adiabatic CAM Using Adiabatic CAL Circuits," *Circuits, Communications and Systems 2009 Pacific Asia Conference*, pp. 338–341, 2009.
- [16] Jui-Yuan Hsieh; Shanq-Jang Ruan; "Synthesis and design of parameter extractors for low-power pre-computation-based content-addressable memory using gate-block selection algorithm," *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, pp. 316–321, March 2008.
- [17] Kavitha, V.; Jeeva, S.; "Low power design of pre-computation based content-addressable memory," *Communication and*

- Computational Intelligence(INCOCCI), 2010 International Conference*, pp. 223-228, Dec.2010.
- [18] Mahini Alireza, Bergani, Reza, Fatemah, Seyedah,” “low power TCAM forwarding engine for IP packets” *2007 Military Communications Conference*, pp 1-7,2007.
- [19] Rueichi Shen; Chinhung Peng; Feipei Lai;,” An early design estimation approach to synthesize the low-power pre-computation-based content addressable memory ”. *Open Systems (ICOS), 2011 IEEE Conference*, pp 21 – 25, Sept. 2011.
- [20] Tsung-ShengLai; Chin-HungPeng;FeipeiLai;” Data driven approach for low power precomputation based content Addressable memory“ *Computers & Informatics (ISCI), 2011 IEEE Symposium*, pp328-333, March2011 .
- [21] Yu-Ting Pai; Chia-Han Lee; Shanq-Jang Ruan; Naroska,E; “An improved comparison circuit for low power pre-computation- based content-addressable memory designs“,*Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference*, pp.663-666. Dec.2009.