

## Realization of Bloom Filter for Error Free Transmission Applications

Pavani K<sup>1</sup> Raghavaiah B<sup>2</sup>

<sup>1</sup>PG scholar, Department of ECE, CEC, Chirala, AP, India

<sup>2</sup>Associate professor, Department of ECE, CEC, Chirala, AP, India

---

**Abstract:** Error correction techniques play an important role in today's life. To make the things easier a numerous techniques are introduced. One of the efficient techniques is BLOOM FILTER method. Blooming filters are used in numerous applications one of them is communication and networking. Researchers are still in work on bloom filters to enhance their usage in new scenarios. Reliability is the major challenge that which the advanced electronic circuits are facing due to thermal, radiation and manufacturing violations.

In this paper we have shown the blooming filters as the effective error detection and correction in the data set. For detecting and correcting the data this paper allows synergetic reuse of blooming filters. For very high speed searching applications a technique CAM (content addressable memory) is introduced. Associative memory, associative storage and associative array are the synonyms of CAM. For programming in data structures the name associative array is used most. The input data (tag) is compared with sorted data and result is the matching data address. The custom computers like good year STARAN are built with CAM and are designated as associative computers.

**Index terms:** blooming filters, soft errors, error correction.

---

### I. Introduction

The basic structure of blooming filters has also been extended over the years. For example counting blooming filters (CBFs) are introduced to allow removal of elements from the blooming filter. To optimize the transmission over the network, another extension known as compressed bloom filters are proposed. In large data arrays, for error correction bloom filters are used.

Generally these bloom filters are implemented by using electronic circuits. High speed memory elements are required to store the elements of bloom filters and the processing is the duty of processor in the circuitry. The blooming filter set is commonly stored in low speed memory.

Major challenge faced by the blooming filters is to maintain reliability in considering the elementary violations of electronic circuits. The major errors are faced by poor interfaces, thermal violation, and radiation. Different types of migration levels are invoked for obtaining better reliability. Memories play vital role on implementing bloom filters.

Spare row column technique is used to detect and correct the permanent errors in memory elements. Radiation is one of the major reasons of occurring soft errors, while performing the circuit operation these soft errors are able to change the memory cell value. Soft errors don't have the capability of damaging the memory cell which leads to change the cell value permanently. To overcome these soft errors as per word parity bit or the most advanced error correction codes were used since many years. To migrate the errors in electronic circuit we used the bloom filter technique, the bloom filters are mainly used to detect and correct the fault values and arrays in nano memory. In CAM's for detecting and correcting these errors CBF's are proposed. For detecting the CAM errors, the CBF and CAM's are used in parallel. The consistency can be found by examining the results of CAM and CBF's.

The correction procedure is initiated for replacing the correct value in place of infected value, the replacement value is drawn from the copy which is externally stored, and this is done after detection of error. For the detection and correction of errors, the BF's are added explicitly, these are not present in the design. -BF codes are only used for error correction, and the same concept is adopted to BIFF codes. The BF's are explicitly added, i.e.. They are actually not present in the system. This is different than reuse of BF's that are present in the system.

In the computer memory i.e. RAM, here user provides the data and the address location for searching the data in that particular memory location. Here RAM searches the data and displays the memory location. But in the case of CAM, the CAM searches the data in entire the memory of the computer and produces the list of memory locations where the desired data is found. It returns the multiple locations, whereas the RAM returns a single value, where the data is found in given location. The CAM is the hardware embodiment of what in software terms would be called an associative array.

The network processing form introduced an agreement called look aside interface (LA-1 and LA-1B). The look aside interface interfaces the CAM and other network search engines (NES). Integrated device technology, IBM, Broadcom and other companies produced a number of devices that which follows the LA agreement. Serial look aside (CLA) agreement was proposed by OFI on December 11 2007.

## II. Cam Architecture

RAM, when used for searching operation, it only works on one particular location which is assigned to it. Where as in CAM, it works on the entire memory Which gives an efficient and accurate of the searching memory locations. CAM are designed for operating on entire memory instead of particular location as RAM, hence CAM's are faster than RAM's in virtual applications. But these CAM's are somewhat cost effective than RAM's. RAM has simple storage elements and individual arranged in parallel, CAM should pose its own comparison circuit for detecting the similar element in the memory. At the same time the outputs which are matched with the data word must be combined to yield a complete data word match signal. The additional circuit increases the components and effects cost. As every comparison circuit works at every clock cycle, CAM also influences power requirements. In situations where the other comparison techniques fail, CAM's are replaced for that task. General purpose IC system is one of the successful implementation of CAM's.

These CAM are also used in computer networking devices. As an example, when data reaches for the switch to one of its ports, it updates the internal MAC address of internal tables and port that which receives data. Switch checks the MAC address and the port that which has to receive data. The switch latency is much greater when compared with CAM's. Hence the CAM's or probably used to maintain these MAC address tables.

Ternary CAM's are also used in network routers, in which address has two parts as network address and host address. Network address varies in size based on the subnet mask configuration. Host address occupies the remaining bits. The subnet task is to differentiate the network address bits and host address bits. Routing is done by following the routing table, subnet mask. The router performs the logical AND operation for the network mask addresses and the results are compared with the addresses of network. If both are equal then forwarded the corresponding data. The look up process is very efficient on using a ternary CAM. The mask and compression and mask are done by CAM hardware.

The proposed method is based on the observation CBF, along with that it allows fast membership check in the given array. It is also one of the ways to represent redundant element in the set. This redundancy is used in error correction and detection.

For exploring this idea, on implementing the CBF the element are stored in slow memory and the CBF is stored in fast memory. In practice the element of CBF is stored in DRAM and the CBF is stored in cache memory. CBF is frequently used and it needs a fast accesses time for maximizing the time. Whereas the element set is only used for appending, removing and reading the data, hence time is not an issue here. When the entire element set is stored in the slow memory. There is a minimum chance of deletion data element from the set. Therefore, the false negatives issue in CBFs discussed in is not a concern in our case.

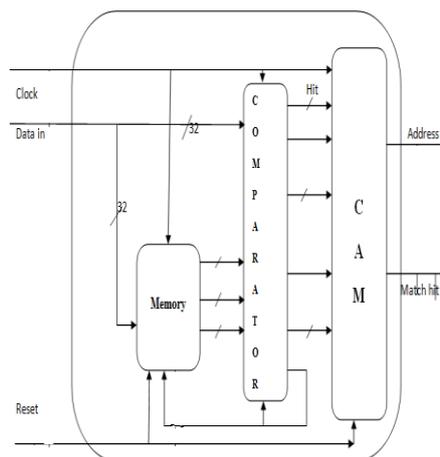


Fig 1:Architecture of the BLOOM Filter

Along with these, soft errors are the events that occur in rare, hence the error occurring time large. Therefore the occurrence of error is terms of days and weeks and comenly associated that errors are isolated.

Our intention is to implmenting this CBF design is to achieve the error correction for sigle bit errors. That is make use of CBF's to work against the errors instead of using the ECC t the memories.

### III. Search Procedure of Cam

First step is to obtain error correction for detected errors. This is done by parity check method while assessing the DRAM or the cache memory. Scrubbing is done periodically on memories for the early detection of errors. When the error is encountered it is cleared by the CBF and the reconstruction circuit generates the element by using the reconstruction circuit. If the error is occurred in data, then it is more complex, then it is divided into phrases and that is described as following. CAM are hardware search engines which are much faster than the static algorithmic approaches. The CAM architecture consists of memory along with that a comparator which compares. The operation should be completed within a single clock cycle. In this paper the introduced CAM architecture and its circuits are first describing the applications for address lookup in Internet routers. Then we describe implementation of this lookup function with CAM.

We have two types of CAM's as binary which will have two possible cases as (0,1) binary symbols. Ternary CAM's also uses don't cares along with the binary digits zero and one. Hence the digit set of ternary CAM consists of (0,1,X).

The comparison and storage circuitry is included in the core cells. In the figure the search lines are searched vertically across the array indicating whether the search data matches the row word.

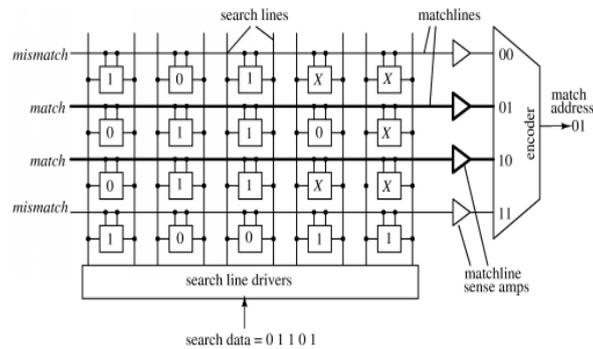
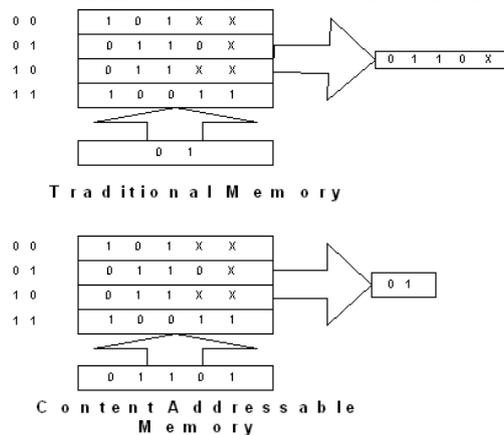


Fig 2: Internal process of the search procedure

- Read operation in traditional memory:  
Input is address location of the content that we are interested in it. In the output the content of that corresponding address should be present.
- In CAM it is the reverse:  
Input is associated with something stored in the memory. Output is location where the associated content is stored.

Fig3 is the architecture of traditional and CAM memory



CAM is used as search engine. That is.. if want to search an element in the given set of data elements, previously we are using the searching or sorting techniques but after proposing the CAM , this replaces the usage of those techniques for searching and sorting. The CAM search operation begins with the pre-searching of all the data, putting them all temporarily in the match state. The Next is to search line drivers broadcast the search data, 01101 in the figure, onto the search lines. Then the CAM core cell search for the required element in whole data. Similar data cell can't identify, but dissimilar elements can be drawn down and replaced. The

aggregate mismatch lines are pulled down for every line set. In the figure, the two middle matched lines will stay at their positions, indicating a match, the other matched lines that are grounded indicates the mismatch. This match address is given for RAM as input address that contains a list of output ports as depicted in Fig. this CAM/RAM system is a complete information 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.

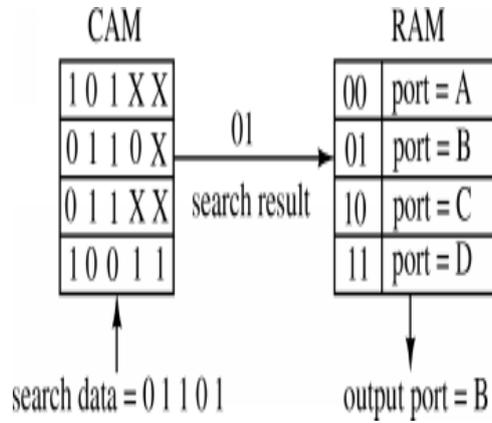
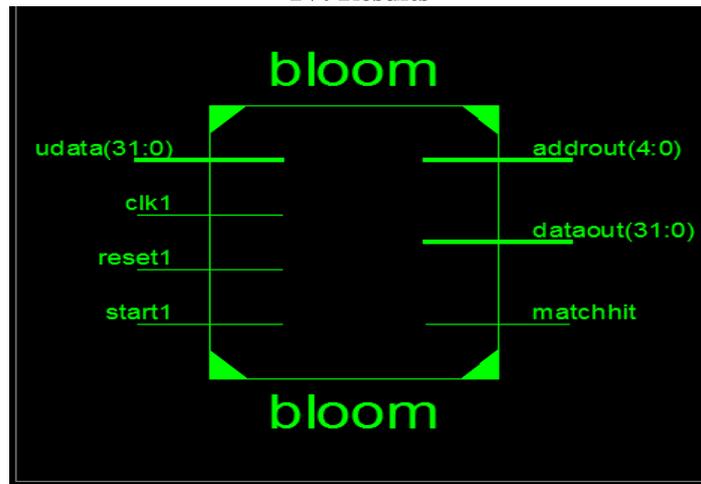


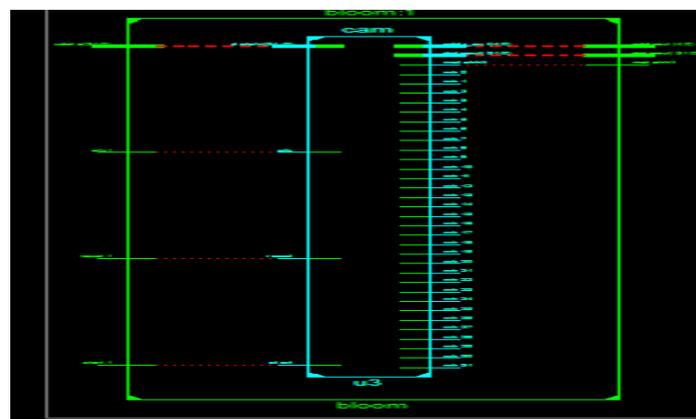
Fig 4: Routing Table

- The input to the system is the *search word*.
- Encoder specifies the match location.
- If multiple matches, a priority encoder selects the first match.
- *Hit signal* specifies if there is no match.

#### IV. Results



RTL SCHEMATIC



INTERNAL SCHEMATIC



WAVE FORMS

### V. Conclusion

In this brief, a new application of BFs has been proposed. The idea is to use the BFs in existing applications to also detect and correct errors in their associated element set. In particular, it is shown that CBFs can be used to correct errors in the associated element set. This enables a cost efficient solution to mitigate soft errors in applications which use CBFs. The configuration considered in this brief is that of a memory protected with a per word parity bit for which it is demonstrated that the CBF can be used to achieve single bit error correction. This paper shows how existing CBFs can be used to achieve error correction in addition to perform their traditional membership checking function.

### References

- [1]. B. Bloom, "Space/time tradeoffs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.
- [2]. A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," in *Proc. 40th Annu. Allerton Conf.*, Oct. 2002, pp. 636–646.
- [3]. A. Moshovos, G. Memik, B. Falsafi, and A. Choudhary, "Jetty: Filtering snoops for reduced energy consumption in SMP servers," in *Proc. Annu. Int. Conf. High-Perform. Comput. Archit.*, Feb. 2001, pp. 85–96.
- [4]. C. Fay et al., "Bigtable: A distributed storage system for structured data," *ACM TOCS*, vol. 26, no. 2, pp. 1–4, 2008.
- [5]. F. Bonomi, M. Mitzenmacher, R. Panigrahy, S. Singh, and G. Varghese, "An improved construction for counting bloom filters," in *Proc. 14th Annu. ESA*, 2006, pp. 1–12.
- [6]. M. Mitzenmacher, "Compressed bloom filters," in *Proc. 12th Annu. ACM Symp. PODC*, 2001, pp. 144–150.
- [7]. M. Mitzenmacher and G. Varghese, "Biff (Bloom Filter) codes: Fast error correction for large data sets," in *Proc. IEEE ISIT*, Jun. 2012, pp. 1–32.
- [8]. S. Elham, A. Moshovos, and A. Veneris, "L-CBF: A low-power, fast counting Bloom filter architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 6, pp. 628–638, Jun. 2008.
- [9]. T. Kocak and I. Kaya, "Low-power bloom filter architecture for deep packet inspection," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 210–212, Mar. 2006.
- [10]. S. Dharmapurikar, H. Song, J. Turner, and J. W. Lockwood, "Fast hash table lookup using extended bloom filter: An aid to network processing," in *Proc. ACM/SIGCOMM*, 2005, pp. 181–192.