International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

ANALYSIS OF "BLOOM FILTER" USING CAM BASED ALGORITHM

K.Pradeep *
A.Ramesh **

Abstract

Content-addressable memory (CAM) employing a new algorithm for associativity between the input tag and the corresponding address of the output data. The proposed architecture is based on a recently developed sparse clustered network using binary connections that on-average eliminates most of the parallel comparisons per- formed during a search. Given an input tag, the proposed architecture computes a few possibilities for the location of the matched tag and performs the comparisons on them to locate a single valid match.Bloom filters (BFs) provide a fast and efficient way to check whether a given element belongs to a set. Reliability is becoming a challenge for advanced electronic circuits as the number of errors due to manufacturing variations, radiation, and reduced noise margins increase as technology scales. In this brief, it is shown that BFs can be used to detect and correct errors in their associated data set. This allows a synergetic reuse of existing BFs to also detect and correct errors. The proposed scheme 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 (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.

Keywords:

Bloom filters (BFs),
Error correction,
Soft errors.

Author correspondence:

* Doctorate Program, Linguistics Program Studies, Udayana University Denpasar, Bali-Indonesia (9 pt)
** STIMIK STIKOM-Bali, Renon, Depasar, Bali-Indonesia

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

PRADEEP KONDAPALLI [1] ANTARAKONDA RAMESH [2]
Associate Professor PG Scholar
Department of ECE, BABA INSTITUTE OF TECHNOLOGY & SCIENCES, VISAKHAPATNAM.

1. Introduction

The basic structure of BFs has been extended over the years. For example, counting BFs (CBFs) were introduced to allow removal of elements from the BF. To optimize the transmission over the network, another extension known as compressed Bloom filters was proposed. Recently Bloom filter (Biff) codes that are based on BFs have been proposed to perform error correction in large data sets.In most cases, BFs are implemented using electronic circuits. The contents of a BF are commonly stored in a high speed memory and required processing is done in a processor or in dedicated circuitry. The set used to construct the BF is commonly stored in a lower speed memory.Errors caused by interferences, radiation, and other effects become more common. Therefore, mitigation techniques are used at different levels to ensure that the circuits continue to operate reliably. For BF implementation, memories are the critical elements.For memories, permanent errors and defects are commonly corrected using spare rows and columns. However, soft errors caused for example by radiation can affect any memory cell changing its value during circuit operation. Soft errors do not produce damage to the memory device that continues to operate correctly but has the wrong value in the affected cell. To deal with soft errors, the use of a per word parity bit or more advanced error correction codes (ECCs) has been common in memories for many years .The BFs have also been proposed to mitigate errors in electronic circuits. For example, in a BF is used to identify the faulty words in a nano memory. In, the use of a CBF is proposed to detect and correct errors in content addressable memories (CAMs).In this case, the CBF is used in parallel with a CAM and the objective is to detect errors in the CAM entries. This is done by checking the results of the CAM and the CBF to ensure that they are consistent. The rest of this paper is organized as follows. Section II describes briefly on CAM. In Section III, overview of BF. In Section IV, the proposed Scheme is introduced. Section V simulation results. Finally, conclusions are in Section VI.

2.Cam Review

The CAM-array is divided into several equally sized sub-blocks, which can be activated independently. For a previously trained network and given an input tag, the classifier only uses a small portion of the tag and predicts very few sub-blocks of the CAM to be activated. Once the sub-blocks are activated, the tag is compared against the few entries in them while keeping the rest deactivated and thus lowers the dynamic energy dissipation.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

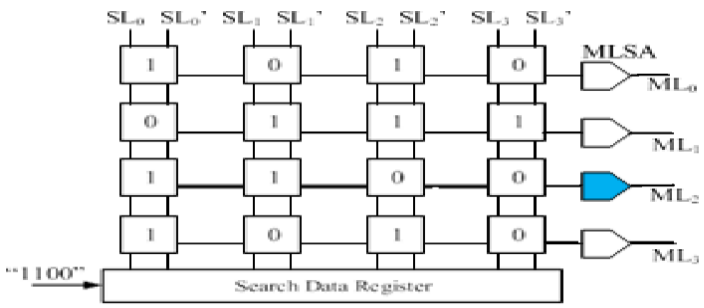


Fig. 1. Simple example of a 4 × 4 CAM array consisting of the CAM cells, MLs, sense amplifiers, and differential SLs.

In a conventional CAM array, each entry consists of a tag that, if matched with the input, points to the location of a data word in a static random access memory (SRAM) block. The actual data of interest are stored in the SRAM and a tag is simply a reference to it. Therefore, when it is required to search for the data in the SRAM, it suffices to search for its corresponding tag. Consequently, the tag may be shorter than the SRAM-data and would require fewer bit comparisons.

An example of a typical CAM array, consisting of four entries having 4 bits each, is shown in Fig. 1. A search data register is used to store the input bits. The register applies the search data on the differential SLs, which are shared among the entries. Then, the search data are compared against all of the CAM entries. Each CAM-word is attached to a common match line (ML) among its constituent bits, which indicates, whether or not, they match with the input bits. Since the MLs are highly capacitive, a sense amplifier is typically considered for each ML to increase the performance of the search operation.

I. OVERVIEW OF BFS

A BF is constructed using a set of *k* hash functions to access an array of *m* bits. The hash functions *h*1, *h*2, *hk* map an input element *x* to one of the *m* bits. The following two operations are defined in a BF.

1) *Insertion:* To insert an element *x* in the BF, the bits in the array that correspond to the positions *h*1*(x)*, *h*2*(x)*,…, *hk(x)* are set to one.
2) *Query:* To query for an element *x* in the BF, the bits in the array that correspond to the positions *h*1*(x)*, *h*2*(x)*,…, *hk (x)* are read and if and only if all of them are one, the element isconsidered to be in the BF.

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). Thus, a CAM is the hardware embodiment of what in software terms would be called an associative array. The data word recognition unit was proposed by Dudley Allen Buck in 1955.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

A major interface definition for CAMs and other network search engines (NSEs) was specified in an interoperability agreement called the Look-Aside Interface (LA-1 and LA-1B) developed by the Network Processing Forum, which later merged with the Optical Internetworking Forum (OIF). Numerous devices have been produced by Integrated Device Technology, Cypress Semiconductor, IBM, Broadcom and others to the LA interface agreement. On December 11, 2007, the OIF published the serial look aside (SLA) interface agreement.

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.

Content-addressable memory is often used in computer networking devices. For example, when a network switch receives a data frame from one of its ports, it updates an internal table with the frame's source MAC address and the port it was received on. It then looks up the destination MAC address in the table to determine what port the frame needs to be forwarded to, and sends it out on that port. The MAC address table is usually implemented with a binary CAM so the destination port can be found very quickly, reducing the switch's latency.

Ternary CAMs are often used in network routers, where each address has two parts: the network address, which can vary in size depending on the subnet configuration, and the host address, which occupies the remaining bits. Each subnet has a network mask that specifies which bits of the address are the network address and which bits is the host address. Routing is done by consulting a routing table maintained by the router which contains each known destination network address, the associated network mask, and the information needed to route packets to that destination. Without CAM, the router compares the destination address of the packet to be routed with each entry in the routing table, performing a logical AND with the network mask and comparing it with the network address. If they are equal, the corresponding routing information is used to forward the packet. Using a ternary CAM for the routing table makes the lookup process very efficient. The addresses are stored using "don't care" for the host part of the address, so looking up the destination address in the CAM immediately retrieves the correct routing entry; both the masking and comparison are done by the CAM hardware. This works if (a) the entries are stored in order of decreasing network mask length, and (b) the hardware returns only the first matching entry; thus, the match with the longest network mask (longest prefix match) is used.

II. PROPOSED SCHEME

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

The proposed scheme is based on the observation that a CBF, in addition to a structure that allows fast membership check to an element set, is also in a way a redundant representation of the element set. Therefore, this redundancy could possibly be used for error detection and correction.

To explore this idea, a common implementation of CBFs where the elements of the set are stored in a slow memory and the CBF is stored in a faster memory is considered. In particular, it is assumed that the elements of the set are stored in DRAM while the CBF is stored in a cache . The reasoning behind this is that the CBF is accessed frequently and needs a fast access time to maximize performance, while the elements of the set are only accessed when elements are read, added or removed and therefore the access time is not an issue. It should also be noted that when the entire element set is stored in a slow memory, no incorrect deletions can occur as they would be detected when removing the element from the slow memory. Therefore, the false negatives issue in CBFs discussed in is not a concern in our case.

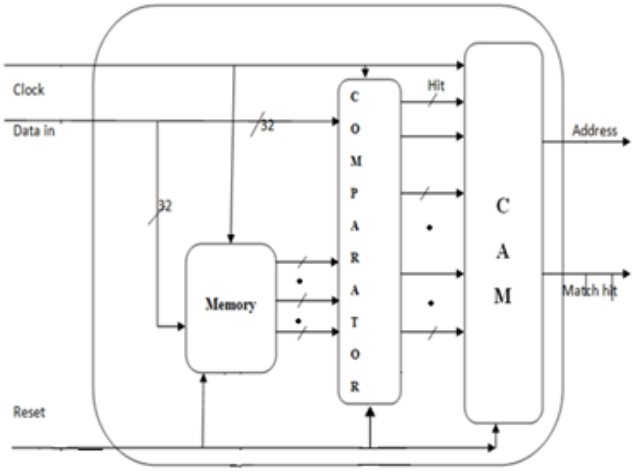


Fig2: CAM Block Diagram Using Comparator

Typically, memories are protected with a per word parity bit or With a single bit error correction code . This is based on the observation that most errors affect a single bit or even if they affect multiple bits, the errors can be spread among different words by the use of interleaving. In addition, soft errors are rare events so that the time between errors is typically large. The arrival rate for terrestrial applications is in the order of at least days or weeks and therefore, it is commonly assumed that errors are isolated. That is, by the time a soft error arrives any previous soft error has been corrected or detected. This is an assumption that is needed, for example, when single bit error correction codes are used.

In the following, one of these two most common protection options is used. In particular, it is assumed that both the DRAM and the cache are protected with a per word parity bit that can detect single errors. As when using single bit error correction codes, it is also assumed that errors are isolated.

The goal for this implementation is to achieve the correction of single bit errors using the CBF. That is, the CBF would enable single bit error correction without incurring in the cost of adding an ECC to the memories.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

The first step to achieve error correction is to detect errors. This is done by checking the parity bit when accessing either the DRAM or the cache. To ensure earlier detection of errors, the use of scrubbing to periodically read the memories could be considered . Once an error is detected, a correction procedure is triggered. If the error occurs in the CBF, it can be corrected by clearing the CBF and reconstructing it using the element set. If the error occurs in the element set, the procedure is more complex and can be divided in two phases that are described in the following sections. The idea is that the simpler and faster procedure is used first and only when it is unable to correct the error, the second more complex error correction procedure is used subsequently.

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 enable 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. 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 remainder of this introdcution assumes you have some familiarity with the operation of transistors and basic ciruit organization of random-access memory (RAM).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. Figure3 shows a block diagram of a simplified 4 x 5 bit ternary CAM with a NOR-based architecture. The CAM contains the routing table from Table 1 to illustrate how a CAM implements address lookup. The CAM core cells are arranged into four horizontal words, each five bits long. Core cells contain both storage and comparison circuitry. The search lines run vertically in the figure and broadcast the search data to the CAM cells. The matchlines run horizontally across the array and indicate whether the search data matches the row's word. An activated matchline indicates a match and a deactivated matchline indicates a non-match, called a mismatch in the CAM literature. The matchlinesare inputs to an encoder that generates the address

corresponding to the match location.

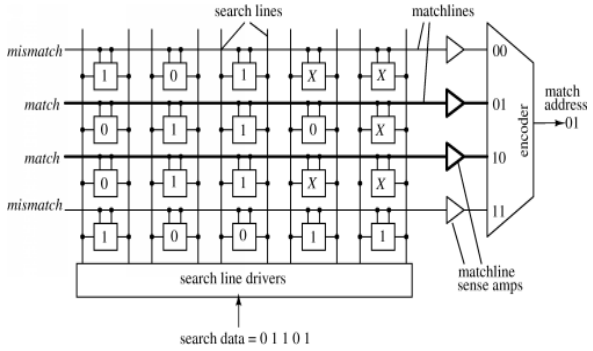


Fig:3 simplified 4 x 5 bit ternary CAM with a NOR-based architecture.

- Read operation in traditional memory:

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

- ❑ Input is address location of the content that we are interested in it.
- ❑ Output is the content of that address.

➢ In CAM it is the reverse:

- ❑ Input is associated with something stored in the memory.
- ❑ Output is location where the associated content is stored.

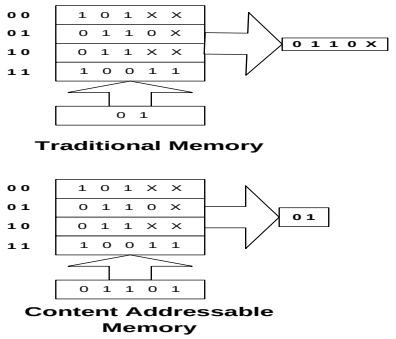


Fig:4 Read operations in Traditional Memory and CAM

CAM can be used as a search engine. We want to find matching contents in a database or Table. A CAM search operation begins with precharging all matchlines high, putting them all temporarily in the match state. Next, the search line drivers broadcast the search data, 01101 in the figure, onto the search lines. Then each CAM core cell compares its stored bit against the bit on its corresponding search lines. Cells with matching data do not affect the matchline but cells with a mismatch pull down the matchline. Cells storing an X operate as if a match has occurred. The aggregate result is that matchlines are pulled down for any word that has at least one mismatch. All other matchlines remain activated (precharged high). In the figure, the two middle matchlines remain activated, indicating a match, while the other matchlines discharge to ground, indicating a mismatch. Last, the encoder generates the search address location of the matching data. In the example, the encoder selects numerically the smallest numbered matchline of the two activated matchlines, generating the match address 01. This match address is used as the input address to a RAM that contains a list of output ports as depicted in Figure 5. This CAM/RAM system is a complete implementation of address lookup engine. The match address output of the CAM is in fact a pointer used to retrieve associated data from the RAM. In this case the associated data is the output port. The CAM/RAM search can be viewed as a dictionary lookup where the search data is the word to be queried and the RAM contains the word definitions. With this sketch of CAM operation, we now look at the comparison circuitry in the CAM core cells.

Example Routing Table

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

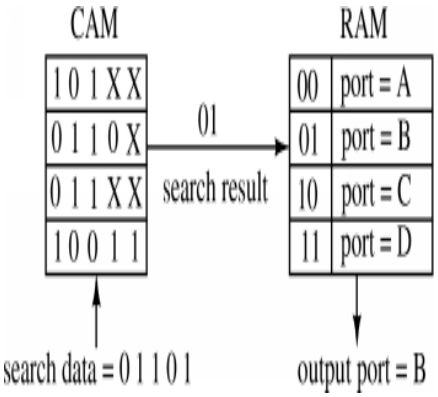
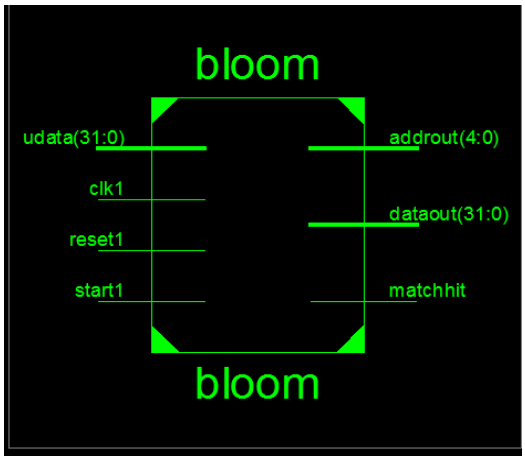


Fig:5 Routing Table for CAM/RAM

- 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.

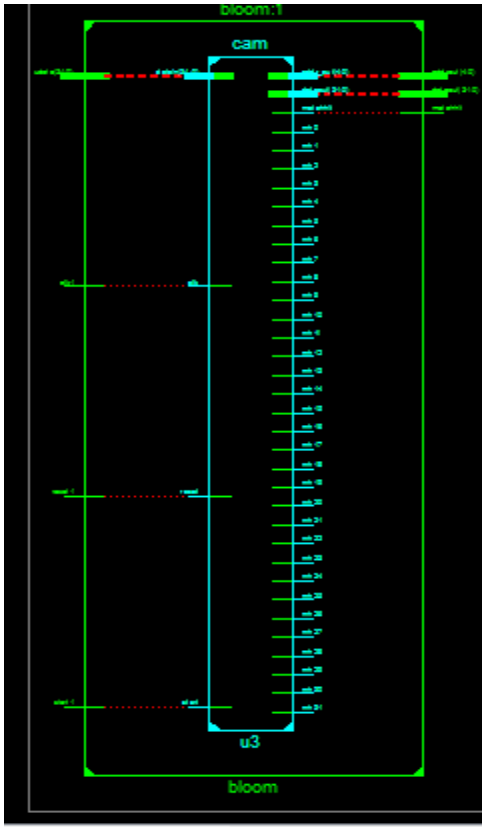
III. RTL SCHEMATIC AND SIMULATION RESULT

RTL : The RTL SCHEMATIC gives the information about the user view of the design.The internal blocks contains the basic gate representaion of the logic.These basic gate realization is purely depend upon the corresponding FPGA selection and the internal database information.



International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

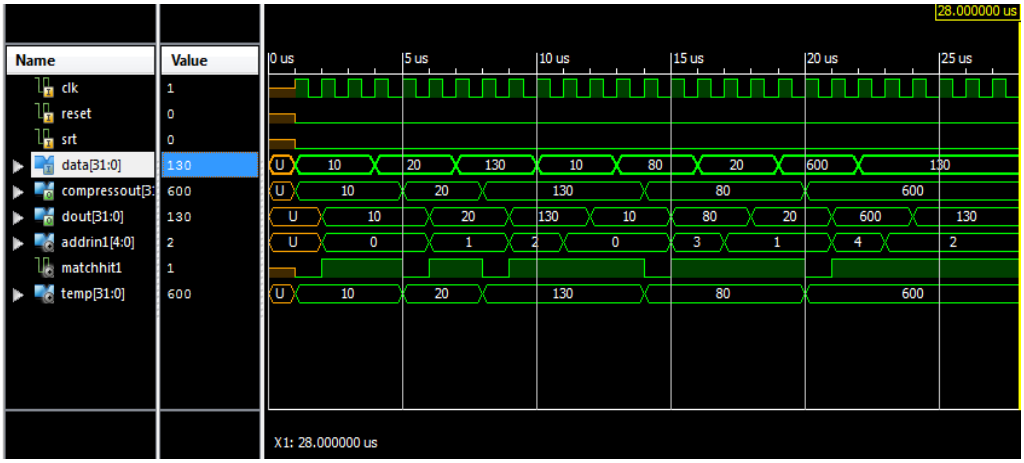
Internal schematic:



SIMULATION:

In the simulations for measuring the energy consumption and the cycle time (/bit/search), on average half of the data bits were assumed to mismatch in case of a word mismatch.The relationship between the dynamic energy consumption of SCN-CAM and the tag length is depicted for various number of entries of the CAM in comparison with the conventional CAMs. The estimated energy consumption is obtained based on the extracted values for energy consumption using HSPICE simulations. As the value of q is increased, the energy consumption is decreased as well since the number of comparisons is reduced but up to a point until the energy consumption of the SCN-based classifier itself would dominate that of the CAM array. Therefore, the energy consumption of the SCN-based classifier is not dependent on the original tag length, and rather on the number of entries in the CAM array.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A



Device Utilization Summary

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	1768	960	184%
Number of Slice Flip Flops	2133	1920	111%
Number of 4 input LUTs	1287	1920	67%
Number of bonded IOBs	99	66	150%
Number of GCLKs	2	24	8%

Typically, memories are protected with a per word parity bit or With a single bit error correction code. This is based on the observation that most errors affect a single bit or even if they affect multiple bits, the errors can be spread among different words by the use of interleaving. In addition, soft errors are rare events so that the time between errors is typically large. The goal for this implementation is to achieve the correction of single bit errors using the CBF. That is, the CBF would enable single bit error correction without incurring in the cost of adding an ECC to the memories.

CONCLUSION

In this application BLOOM FILTERFS has been proposed. The idea is to use the BFs in existing applications to 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

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

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 shows how existing CBFs can be used to achieve error correction in addition to perform their traditional membership checking function

International Journal of Engineering, Science and Mathematics
Vol. xIssue x, Month201x,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind
Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

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.