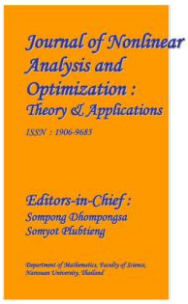
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 13, Issue. 2 : 2022
ISSN : **1906-9685**



SRAM EMULATED TCAMS FOR ERROR DETECTION AND CORRECTION

S Ranjith Kumar,V Naresh Kumar, Assistant Professor Department Of ECE Sree Chaitanya College of Engineering, Karimnagar

ABSTRACT:
TCAMs are commonly found in integrated circuits made expressly for networking applications, either as integrated intellectual property blocks or as standalone units. Nevertheless, field-programmable gate arrays (FPGAs) do not have TCAM blocks. Still, most FPGA suppliers offer SDN development kits, and FPGAs' versatility makes them attractive for SDN applications. To implement TCAM functionality, you must simulate TCAMs using the logic blocks found in the FPGA. In recent years, numerous approaches to emulate TCAMs using FPGAs have been proposed. Some of them make TCAMs by utilizing the large number of memory blocks present in modern FPGAs. The possibility of soft errors damaging the bits that are preserved is one drawback of using memory. Memories can be safeguarded by using error correction codes, which require additional memory bits for each word, or parity checks, which can spot flaws. This short discusses the security of the memories used to imitate TCAMs.

1. INTRODUCTION:

Multiple Cell Upsets (MCUs) are like a single event that induces several bits in an integrated circuit to fail at the same time. It affects mostly Static Random Access Memory (SRAM). The MCUs occur due to radiation particle striking the memory and the neutrons penetrate into the SRAM memory. Due to this, electron hole pair generation will take place resulting in an accumulation of the charges in the memory. When the charges exceed the critical charge limit, then it can flip the logical state in the memory [1]. It is stated that neutron irradiation reduced the single event latch-up and the sensitivity of CMOS SRAM [2] Soft errors are a major concern for modern electronic circuits and,in particular, for memories [1]. A soft error can change the contents of the bits stored in a memory and cause a system failure. The soft error rate in terrestrial applications is low. For example, in [2], it was estimated that for a 65-nm static random access memory (SRAM) memory, the bit error rate was on the order of 10−9 errors pe year. That would translate to only one error per year for a system that uses 1 Gbit of memory. However, even such a low error rate is a big concern for critical applications such as communication networks on which the network elements such as routers have to provide a high level of reliability and availability. Therefore, soft errors are an important issue when designing routers or other network elements, and manufacturers take them into account and incorporate error mitigation techniques [3], [4]. For example, error detection and correction codes are commonly used to protect memories [5]. A parity bit can be added to each memory word to detect single-bit errors, or a single-error correction (SEC) code can be used to correct them. These codes require additional bits per word thus, increasing the memory size and also some logic to write and read from the memory. For example, for a 16-bit word, an SEC code requires 5 bits while a parity check requires only one. Ternary content addressable memories (TCAMs) are a special kind of content addressable memories [6] that support do not care bits (commonly denoted as “x”) that match both a zero and a one.TCAMs are widely used in networking applications to perform packet classification [7]. The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receiver can use to check consistency of the delivered message and to picks up data determined to be corrupt. Error detection and correction scheme can be either

systematic or non-systematic. In a systematic scheme, the transmitter sends the unique data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only the error detection is required, a receiver can simple apply the same algorithm to the received data bits and compare its output with the received check bits; if the values do not match, an error has occurred at some point throughout the transmission. Error-correcting codes are regularly used in lower-layer communication, As well as for reliable storage in media such as CDs, DVDs, hard disks and RAM. Static RAM based Field-Programmable Gate Arrays (FPGAs) are most widely used in variety of applications mainly due to short time-to market time, flexibility, high density and cost-efficiency. SRAM-based FPGA stores logic cells configuration data in the static memory organized as an array of latches. FPGA is used for designing complex digital circuits. Power consumption is also reduced by using SRAM. The power consumption of SRAM varies widely depending on how frequently it is accessed; it can be as powerhungry as dynamic RAM, when used at high frequencies, and some ICs can consume many watts at full bandwidth. On the other hand, static RAM used at a somewhat slower pace, such as in applications with moderate clocked microprocessors, draws very little power and can have nearly negligible power consumption when sitting idle – in the region of a few micro watts. Several techniques have been implemented to manage power consumption of SRAMbased memory structures. FPGA device customizable by SRAM consists of an array of programmable logic blocks interconnected by a programmable routing network and I/O blocks. SRAM-based FPGA devices are becoming popular because of their high performance, reduced development cost and re programmability. FPGAs based on a nanometer technology with denser integration schemes. Memories are one of the most widely used elements in electronic systems. Radiation in the environment seriously affects the functionality of a circuit. A single-event upset (SEU) occurs when a charged particle, present in the environment, hits the silicon of a circuit introducing an error in the system. Such errors in FPGA device affects the functionality of the mapped design also called as Soft errors. A soft error will not damage a system's hardware; the only damage is to the data that is being processed in the memory. To address this issue, Built-in Current Sensors (BICS) have recently been applied in conjunction with Single Error Correction/Double Error Detection (SECDED) codes to protect memories from MBUs. But by using those methods only SEU could be corrected. For both the detection and correction of errors, a generic scrubbing scheme to reconstruct the erroneous configuration frame based on the concept of Erasure coding algorithm is introduced in this paper. In this type of Erasure coding algorithm, MBUs are detected by using the interleaving distance which is further classified into horizontal and vertical parity. The results obtained have shown that this approach have a lower delay overhead over other codes. Built-in current sensors (BICS) are implemented to assist with single-error correction and double-error detection codes to provide protection against MBUs. However, this technique can only correct two errors in a word. In this paper, novel matrix code based on divide symbol is implemented to provide enhanced memory reliability. The implemented matrix code utilizes algorithm (integer addition and integer subtraction) to identify errors. Besides, the erasure codes is implemented to minimize the area overhead of extra circuits (encoder and decoder) without disturbing the whole encoding and decoding processes.

2. LITERATURE SURVEY:

In [9], a technique was proposed to reduce power consumption of matchlines in content addressable memories (CAMs) called selective precharge technique. In selective precharge technique, the matchline is divided into two segments. Firstly, the searching operation is performed in the first segment in which first few bits of a word i.e. a small subset of CAM cells are searched. If there is a matching of data in the first segment only then searching of remaining bits in the second segment will be activated. In [8], an architecture was proposed having low-power, low-cost, and high-reliability features called as fully parallel precomputation-based content addressable memory (PBCAM). This design is based on a precomputation skill that saves power consumption of the CAM by reducing number of comparisons in the second part of the comparison process. In this design, one’s count approach is used for precomputation. Hence, a one‟s count parameter extractor was designed using a chain of full adders but it increases delay as data bit length increases. In [7], a

technique was proposed to reduce power consumption of matchlines in content addressable memories (CAMs) called pipelining technique. In this technique, the search operation is pipelined by breaking the match-lines into many segments. Since most stored words do not match in their first segments, the search operation is aborted for subsequent segments. Hence, power gets reduced. The power savings of the pipelined MLs is a result of activating only a small portion of the matchline segments. In [5], a new approach for PBCAM known as a Block-XOR approach was proposed to improve the efficiency of low power precomputation-based CAM (PBCAM) proposed in [8].

3. EXISTING TECHNIQUE:

TCAM IMPLEMENTATIONS

There are two main alternatives to implement TCAMs on FPGAs. sThe first one is to use the FPGA logic resources and flip-flops to implement the TCAM cells and match lines. The second is to use the block memories inside the FPGA [13]. In the first alternative, the bits of the rules are stored in flip-flops. As discussed before, each bit can take three possible values: 0, 1, and x. For example, a flip-flop can be used to store if the bit is 0 or 1 and another flip-flop that acts as a mask and is set when the bit is do not care [12]. Then, the programmable logic can be used to implement the comparison against the key. This alternative uses many resources per rule and, therefore, cannot be used to implement large TCAMs with tens of thousands of rules of more than 100 bits that operate at high speed. The second alternative is based on the use of the embedded memories available in the FPGA. To do so, the key is divided into smaller blocks of b bits. Then, a rule can be emulated using a 1-bit memory of $2b$ positions for each block. When searching for a key, all the memories are accessed using the corresponding key bits and if all the positions read have a one, a match is detected. In general, k rules can be implemented by using a $k - bi\ t$ memory of $2b$ positions for each block. This is best illustrated with an example.

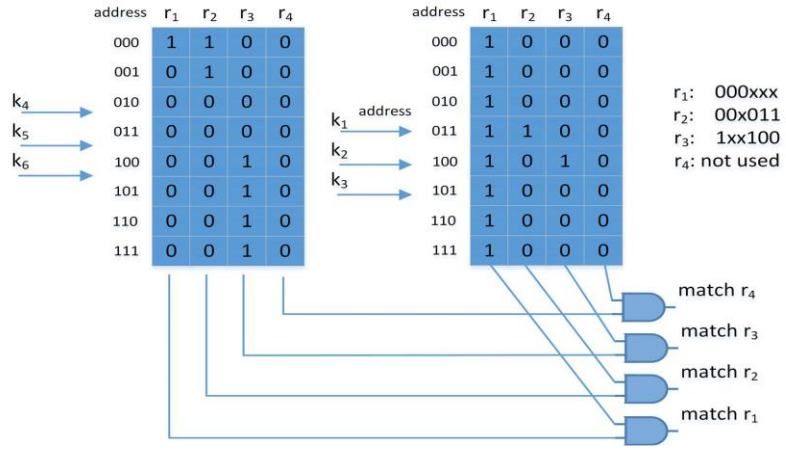


Fig. 1. Example of a TCAM with 6-bit keys and four rules emulated using two SRAMs.

Let us consider a key of 6 bits that is divided into two blocks of 3 bits. Then, a TCAM with four rules can be implemented as shown in Fig. 1. It can be observed that the memories have 23 = 8 positions and a width of 4 bits. The leftmost memory is accessed using the upper 3 bits of the key and the other with the lower three. Those bits are used to determine the address of the position read from the memory. The rules stored in each bit are also shown in Fig. 1. Let us consider a search for key: 000011. We would access the first position (adder 000) on the leftmost memory reading 1100 and the four position (address 011) on the other memory reading 1100. After performing AND there would be a match only for rules $r1$ and $r2$. Looking now at the rules, it can be observed that rules that are not used ($r4$) have zeros in all the memories and positions. For the rest of the rules, the number of ones in a given memory depends on the number of

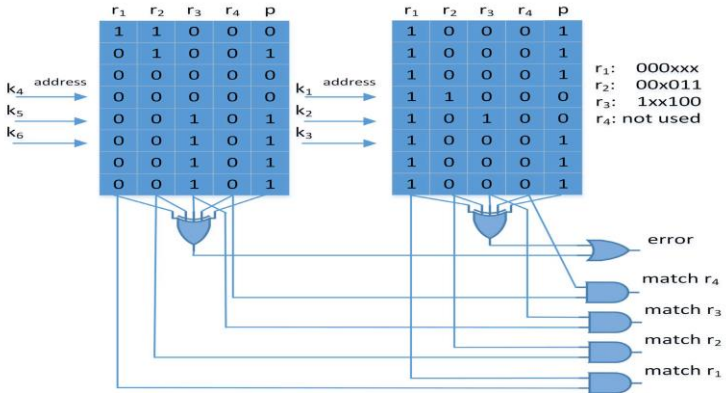


Fig. 2. Parity protected TCAM with 6-bit keys and four rules emulated using

DRAWBACKS:

Major drawback lies in hardware implementation. Because of more density requirement for error detection and correction , need more hard ware components for designing.

Some more complexity is presented in software implementation.

4. PROPOSED TECHNIQUE:

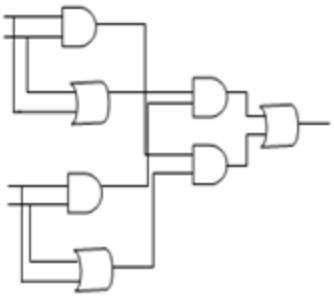


Fig: 4 bit majority

In Boolean logic, the majority function (also called the median operator) is a function from *n* inputs to one output. The value of the operation is false when *n*/2 or more arguments are false, and true otherwise. Alternatively, representing true values as 1 and false values as 0

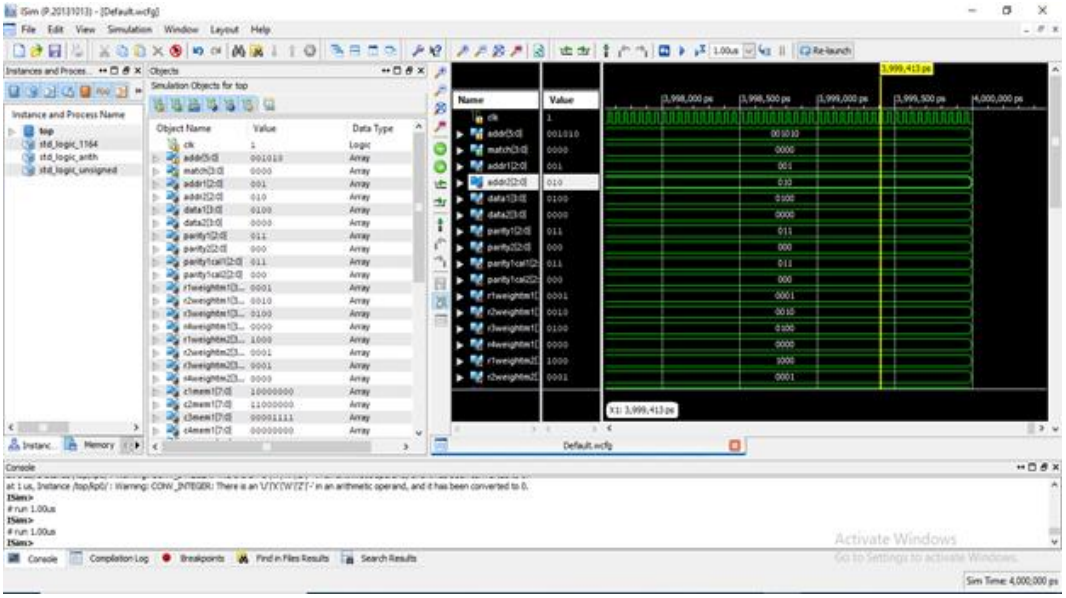
Majority Circuit Implementation: Here we present a compact implementation for the majority gate using Sorting Networks [13]. The majority gate has application in many other error-correcting codes, and this compact implementation can improve many other applications. Majority function of binary digits is simply the median of the digits. A majority gate is a logical gate used in circuit complexity and other applications of Boolean circuits. A majority gate returns true if and only if more than 50% of its inputs are true.

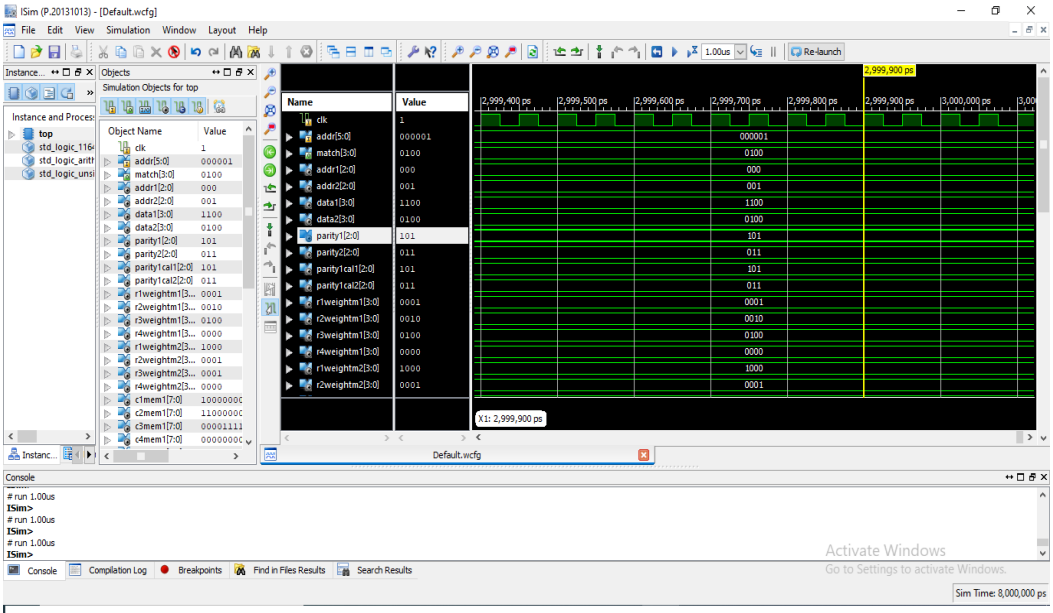
For instance, in a full adder, the carry output is found by applying a majority function to the three inputs, although frequently this part of the adder is broken down into several simpler logical gates.

Many systems have triple modular redundancy; they use the majority function for majority logic decoding to implement error correction.

A major result in circuit complexity asserts that the majority function cannot be computed by AC0 circuits of sub exponential size

5. RESULT





6. CONCLUSION

More complex error correction codes (ECCs) are often used to protect memory and prevent data corruption caused by MCUs; nevertheless, their primary drawback is an increase in delay overhead. Recently, matrix codes (MCs) based on Hamming codes have been proposed for memory protection. This completed project offered a novel per-word DMC to guarantee memory reliability. The protection code employed a decimal way to detect mistakes in order to find and repair more issues. The gathered data showed how much better the developed strategy protected against large MCUs in memory. Additionally, the chosen decimal error detection method is a strong option for CAM MCU detection because it may be combined with BICS to provide a high enough level of immunity.

FUTURE SCOPE:

The research done for this publication has many potential applications in other fields. One important possibility is the development of an error-correcting system with enhanced performance, less overhead delays, lower power needs, and smaller overall area consumption. The investigation can be finished by pipelining the existing codes into a useful format, which will reduce the delay overhead.

Furthermore, the area can be reduced by making adjustments to the adders and other realization-related parts, and the power consideration can be greatly reduced by correctly applying the two previously mentioned ideas.

REFERENCES:

[1] Dejan Georgiev, "Low Power Concept for Content Addressable Memory (CAM) Chip Design," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol.2, Issue 7, July 2013.

[2] S. Jeeva, S. Bharathi and Dr. C. N. Marimuthu, "Low Power Architecture of Banked Pre-Computation Based Content Addressable Memory", International Conference on Computing, Communication and Information Technology (ICCCIT 2012), p.p. 156-160, June 2012.

[3] Rafeekha M. J, V. Lakshmi Narasimhan, "Banked Approach of Low Power Design of Pre-Computation Based Content Addressable Memory", International Journal of Modern Engineering Research (IJMER), Vol. 2, Issue 3, p.p. 1424-1429, May-June 2012.

[4] Subha. M, "The Efficient Architecture Methods for Low Power Content Addressable Memory-Survey", Recent Advances in Networking, VLSI and Signal Processing, p.p. 141-146.

[5] 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, Vol. 16, No. 3, p.p. 331-335, March 2008.

[6] K. Pagiamtzis and A. Sheikholeslami, "ContentAddressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey," IEEE Journal of Solid-State Circuits, Vol. 41, p.p. 712-727, March 2006.

[7] K. Pagiamtzis and A. Sheikholeslami, "A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme," IEEE J. Solid-State Circuits, Vol. 39, No. 9, p.p. 1512–1519, Sep. 2004.

[8] C.-S. Lin, J.-C. Chang, and B.-D. Liu, "A low-power precomputation-based fully parallel content-addressable memory," IEEE J. Solid-State Circuits, Vol. 38, No. 4, p.p. 654–662, Apr. 2003.

[9] C. A. Zukowski and S.-Y. Wang, "Use of selective precharge for low power content-addressable memories," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Vol. 3, p.p. 1788–1791, 1997.

[10]. Yoon Seok Yang , Seung Eun Lee, Wei Wu, Ravi Iyer, Gwan S. Choi, *Low-Power, Resilient Interconnection with Orthogonal Latin Squares*, in IEEE design & test of computers.

[11]. V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, *Generalized parity-check matrices for SEC-DED codes with fixed parity*, in Proc. IEEE On-Line Testing Symp., Jul. 2011, pp. 198–20.

[12]. Jing Guo, Liyi Xiao, Zhigang Mao, and Qiang Zhao ,*Enhanced Memory Reliability against Multiple Cell Upsets Using Decimal Matrix Code*, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, March 2014.

[13]. Pedro Reviriego, Salvatore Pontarelli, Juan Antonio Maestro, and Marco Ottavi, *A Method to Construct Low Delay Single Error Correction Codes for Protecting Data Bits Only*, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems., vol. 32, no. 3, pp. 479 - 483, March 2013