Journal for Educators, Teachers and Trainers
The PROFESIOLab electronic, peer-reviewed, open-access Magazine

ISSN 1989-9572

DOI:10.47750/jett.2020.12.04.027

IMPROVING SECURITY IN TERNARY-BASED SOFT ERROR RESILIENT SRAM CONTENT ADDRESSABLE MEMORY USING CHECKSUMS

\

[1]*Devarakonda Chandana,*
[2]*Lavanya Repaka,*
[3]*Uppunutula Venu,*
[4]*Kothapalli Aravind,*
[5]*Madishetty Nikilesh Kumar*

Journal for Educators, Teachers and Trainers, Vol.12 (4)

https://jett.labosfor.com/

Date of Reception: 28 April 2021

Date of Revision: 27 Aug 2021

Date of Acceptance: 18 September 2021

***Devarakonda Chandana, Lavanya Repaka, Uppunutula Venu, Kothapalli Aravind, Madishetty Nikilesh Kuma* (2021). IMPROVING SECURITY IN TERNARY-BASED SOFT ERROR RESILIENT SRAM CONTENT ADDRESSABLE MEMORY USING CHECKSUMS. Journal for Educators, Teachers and Trainers, Vol.12(4).208-214.**

Journal for Educators, Teachers and Trainers
The PROFESIOLab electronic, peer-reviewed, open-access Magazine

Journal for Educators, Teachers and Trainers, Vol. 12(4)
ISSN1989 –9572
https://jett.labosfor.com/

IMPROVING SECURITY IN TERNARY-BASED SOFT ERROR RESILIENT SRAM CONTENT ADDRESSABLE MEMORY USING CHECKSUMS

[1]*Devarakonda Chandana,*[2]*Lavanya Repaka,*[3]*Uppunutula Venu,*[4]*Kothapalli Aravind,* [5]*Madishetty Nikilesh Kumar*

[3]*Associate Professor,* [12]*Assistant Professor,*[45]*Students*
Department of ECE
Vaagdevi College of Engineering, Warangal, Telangana

ABSTRACT
Open flow and packet classification applications in Software Defined Networking (SDN) (FPGAs) employ Ternary content addressable memory (TCAM) based on static random access (SRAM) on field programmable gate arrays. SRAM-based TCAMs will therefore be crucial to all FPGA-based systems as they will improve search speed, shorten critical path time, and lessen soft error vulnerability. In conclusion, SRAM-based TCAM will save money and time by identifying single bit parity flaws with little crucial route overhead. This method achieves low reaction time error correction by using a binary encoded TCAM table-maintained SRAM based TCAM for updates. The intended work for this article will demonstrate SRAM-based TCAM with error detection and correction in the 1024x40 size, along with enhanced security through the use of the Checksum technique. Ultimately, this project demonstrated performance in terms of area, latency, and power after being constructed in Verilog HDL and synthesised on a Xilinx Vertex 5 FPGA.

KEYWORDS: Examples of addressable memory include soft errors, FPGAs (field programmable gate arrays), TCAM (ternary content addressable memory), SRAM (static random-access memory), and addressable memory.

I. INTRODUCTION

Charged particles in memory cells corrupt data, resulting in soft mistakes. Soft errors are turning into a big problem for computer systems that need to be very reliable. Soft mistakes are starting to become an issue in in-memory systems. Charged particles are created when radiation particles collide with semiconductors. A single byte of error can bring down the system. Routers and other high-reliability network applications depend on TCAM, which is used to build SDNs. Since it is necessary for several crucial functions including routing, packet forwarding, and packet categorisation, TCAM is among the most crucial parts of networking equipment. Because it requires more thought when building routers or other network components that can withstand soft mistakes, soft error avoidance is a difficult challenge in TCAM to solve. Field programmable gate arrays (FPGAs) are integrated circuits that, as the name suggests, may be modified by a customer or designer after it has been manufactured. A hardware description language (HDL), similar to that used for application-specific integrated circuits (ASIC), is used to specify the FPGA configuration. To carry out intricate digital calculations, modern FPGAs are equipped with a large number of logic

gates and RAM blocks. The contents of the specified embedded memory can be used to define the TCAM function in the majority of SRAM-based FPGA implementations. A false match or mismatch may arise from an erroneous match address brought on by a temporary malfunction, block RAM (BRAM), or scattered RAM. Therefore, in order for lookups to provide accurate matching data, the impacted SRAM word should be changed whenever a soft error occurs. Conversely, SRAM-based TCAM systems are challenging to maintain without improving search performance or key path latency. This brief paper illustrates how to protect SRAM-enabled TCAMs without reducing search performance using an inexpensive, constrained, and simple-to-integrate approach. Regression testing is simple, requires little effort and logic, and relies on sufficient testing. To remove soft errors, the proposed error-correction method updates the duplicate binary-encoded TCAM table stored in SRAM-based TCAM devices. The suggested failure system maintains a high level of searching performance while operating in the background, enabling several searches to be conducted simultaneously. The proposed error-correction technique produces a flawless TCAM design for lookups since it has a short reaction time during the whole (almost) transaction duration.

II. TCAM ON FPGAS BASED ON SRAM

TCAM implementations in recent FPGAs are implemented using on-chip SRAM. A 1X1 TCAM could be achieved with such a 2X1 RAM, with existence of the a "0" TCAM authoritarian leadership by saving a "1" at RAM, the quantity of "1" expressed by saving a "1" in SRAM, as well as the "x" state indicated by saving a "1" both at the SRAM & SRAM locations. A C-bit TCAM architecture can be built using a 1-bit SRAM having 2C locations. The words the address of an SRAM represents the C-bit TCAM pattern, while every word of the TCAM database records match/mismatch information against all possible C-bit patterns. A B-bit broad SRAM having 2C locations can be used to build a C-bit wide TCAM table of B words. Because large TCAM bit patterns do not grow well in terms of necessary memory in SRAM-based TCAMs, the researchers split them down into smaller chunks. TCAM's W-bit broad depth D bit patterns are decomposed into shorter C-bit pieces and implemented utilizing As well as 2C D size Order to start up [8], [9]. A simpleSRAM-based TCAM implementation. A 4-word deep TCAM's 4-bit patterns are split into two 4 X 2 divisions, which are subsequently implemented using two 4 X 4 SRAM. Take into consideration the various search key (1001): The first two bits (10) access the third word (1100) of the first SRAM, while the last two bits (01) read the second SRAM's second sentence (1001). To get the ultimate match result (1000), the received SRAM bits are ANDed together, showing a rule R0 match.

TCAM on FPGAs could be accomplished in a variety of ways. Small-size TCAMs are created using the flip flops (FFs) available in the FPGA's logic assets. As previously stated, the majority of existing FPGA-based TCAMs utilize BRAM or distribution RAM. When shallow SRAMs are utilized to design TCAMs, higher memory economy is attained. Because the lowest depth limit of Xilinx BRAM configurations is 512 bits, 9 bits of TCAM are achieved using 29 BRAM bits. As a result, the best configuration sizes for 18- and 36-kb BRAMs are 512 36 and 512 72, correspondingly, with a TCAM piece width of 9 bits. Every SLICEM's four 64-bit LUTRAMs are converted into a 32-6 basic dual-port RAM by the distribution RAM-based TCAMs, providing 30 TCAM bits per SLICEM. This indicates that 30 bits of TCAM are implemented using 256 distribution RAM bits, and a single bit of TCAM is constructed using 8.53 distribution RAM bits. As a result, SRAMs based on distribution RAM should have a 32-bit depth and a 5-bit TCAM chunk width. TCAM must be developed using BRAMs in practice, with LUTRAMs used for other aspects of the system.

In TCAM setups that require more resources, soft errors are much more sensitive. The proportions of BRAMs, LUTRAMs, and slice files on a 28-nm FPGA device are 34 percent, 5%, and 1%, respectively, trying to make SEUs in the tiny population markedly less likely [16]. As a result, BRAM-based TCAM systems are more prone to soft errors than other models (distribution RAM-based and SR-based). Time failures are a common way to express the real-time soft error rate per incidence in the BRAMs of the FPGA device used in the proposed work's tests (FIT). The success rate of Vertex-5 varies between 73 and 9 percent. TCAM words can portray ranges because the wild-card state "x." A 1-bit word can match a number of different terms whenever the ranges of two words overlap. When a match contains several words, the lowest-order word is communicated. TCAM words must be arranged in terms of priority, with greater priority words being stored in limited memory addresses. A change to a TCAM word could cause a re - organization of the TCAM table's terms, as well as a shift in precedence. Incoming words for lookup operations are paused during update operations since they can be accurately matched. SDN and Open Flow, for instance, will necessitate the replacement of network switches on a regular basis in the future. SRAM-based TCAM solutions on FPGAs update a TCAM entrance in the data type TCAM table, necessitating many TCAM entry movements to maintain the order constraint, and are then plotted onto SRAMs to try to mimic the TCAM function. To fix soft faults in SRAMs that imitate the TCAM function, the proposed ER-TCAM utilizes redundant data of stored binary-encoded TCAM material.

III. PROPOSED METHODOLOGY

Static random access (SRAM) based Ternary programmable memory (TCAM) on field engineeringarray is used in traffic classification and open flow application in (SDN) (FPGAs) (FPGAs). Therefore, SRAM-based TCAMs will play a crucial role in all FPGA-based devices, because they will help reduce soft error susceptibility while also enhancing search speed and minimizing critical path time. In conclusion, by identifying single bit parity faults with little critical route cost, SRAM-based TCAM will minimize the price and response time. This technology leverages a binary coded TCAM bar counter SRAM based TCAM for updating purpose to provide a low reaction speed error correction. This research aims to show SRAM-based TCAM with error correction, including TCAM size of 1024x40. Finally, this project was written in Verilog HDL and implemented on a Xilinx Vertex 5 FPGA, demonstrating performance in terms of size, latency, and energy consumption. Information redundancy is crucial for fault tolerance in SRAM-based TCAMs. So because three components of a TCAM cell could be made up of two SRAM bits, the 2C D bit data stored in SRAMs with TCAM capabilities is derived from D C TCAM material, which is at least 2D C bit information ("0""00," "1""01," and "x""10"). As previously noted, the initial TCAM content is also maintained on-chip for use with succeeding SRAM-based TCAMs, rendering the design-configured SRAMs' 2C X D bit data completely useless. ER-TCAM uses system-level data redundancy to correct soft errors in SRAMs. The main idea behind the ER-TCAM technology. The TCAM table divisions are implemented by the two SRAMs indicated in

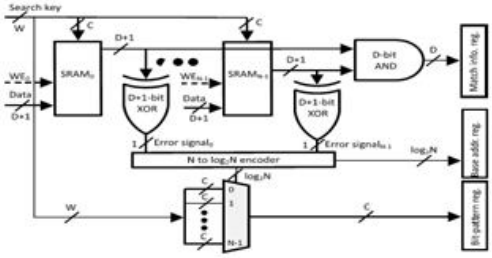

Figure 1: Proposed ER-TCAM architecture for error detection

To recognize an SBU in SRAMs, the ER-TCAM adds a duplicate bit to each SRAM word, as shown in Fig. 1. All bits of SRAMs were checked for errors during lookups. When the ER-TCAM identifies a word mistake, it uses redundant data from the data type TCAM database to rectify it.

The suggested ER-TCAM error-detection architecture is shown in Figure 1. The bit of the SRAM reading is EX-ORed to give an error signal when an inputted search key is used for lookup. The N SRAM fault signals in the TCAM architecture are encoded to create a log2N-bit error message saying that identifies each failing SRAM. The error code is sent to the error-correction module, along with any applicable search-key bit patterns, to generate a log2N-bit error message saying that uniquely identifies each defective SRAM. The error code is sent to the error-correction module, along with any applicable search-key bit patterns.

IV. ARCHITECTURE OF THE PROPOSED ER-TCAM

A SRAM for holding data type TCAM database tables, an ECV computation unit, an address generation unit (AGU), and a gaffe read/write controller are all part of the ER-TCAM design, as shown in Figure 2. The MOD-D counter produces a fresh set of log2D bits every cycle. The SRAM ID's most crucial bits point to the start of the appropriate SRAM sub-block, whereas the comp's lower log2D bits select SRAM word in the sub-block. The AGU has now accessibility to all binary coded bits in the appropriate division of the TCAM database. The read TCAM words are compared to the C-bit pattern to produce a match bit each cycle, that takes D clock cycles to compute and yields the ECV. The read/write controller sends a student's' achievement enable signal to the defective SRAM word, allowing the computed ECV to be written over it.

The ER-TCAM facilitates search operations throughout the error-correction process since SRAMs underlying the TCAM function are available for lookup operations. These SRAMs are configured as a basic dual-port RAM by the ER-TCAM, which reads and writes in succession at the same clock cycle. The error correcting mechanism totally overlaps the search and rescue operations in the ER-TCAM after the ECV is calculated and written via the write port of SRAM. Although a soft error can occur in an SRAM holding a binary encoded TCAM table, its error incidence is modest when contrasted to SRAMs rearising TCAM tables due to its small size. Despite this, the ER-TCAM may preserve SRAM by employing ECC to store the binary encoded TCAM table with minimal memory and error-correction latency costs.

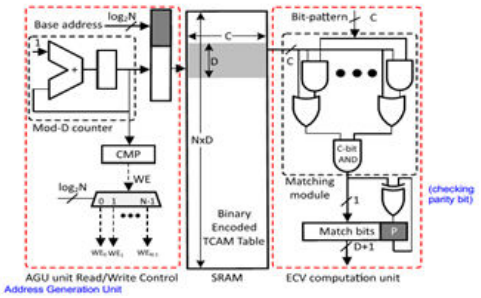

Figure 2: Proposed ER-TCAM error-correction module

V. METHOD FOR CALCULATING THE CHECKSUM:

In detecting faults in upper layer protocols, checksums are thought to be more trustworthy than LRC, VRC, and CRC. Upper layer protocols use it to detect mistakes. A Checksum Producer is used on the Transmitter end of the communication, while a Checksum Checker is used on the Receiver side. The checksum generator separates the information into identical n-bit subunits for the Sender. The average duration of this piece is 16 inches. To bring these components together as a whole, the one's complement method is applied. The n bits that make it up this total amount are as follows: The final step is to complement the newly produced bit. The checksum is calculated and added to the time of the previous data unit before it is delivered to the receiving computer.

The receiver delivers the data and checksum to the checksum checker after receiving the data and checksum. When a data unit is divided into numerous equal-length subunits, the checksum checker adds them all up. The checksum, that is one of these subunits' components, is one of them. Following that, the completed task is commended. The data is error-free when the augmented result is zero. If the dataset shows an error, the result is non-zero, as well as the recipient rejects the input.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

Table:ER-TCAM: A Soft-Error-Resilient SRAM-Based Ternary Content-Addressable Memory for FPGAs Synthesized on Xilinx Vertex-5 FPGA

TCAM Size	1024x40	
	Parity	Checksum
Slice Register	39	39
LUT	96	91
Occupied Slices	51	36
Number of IOBs	71	71
Delay (ns)	2.402	2.292
Power (W)	3.314	3.315

Table: 1. Comparison between different methodologies

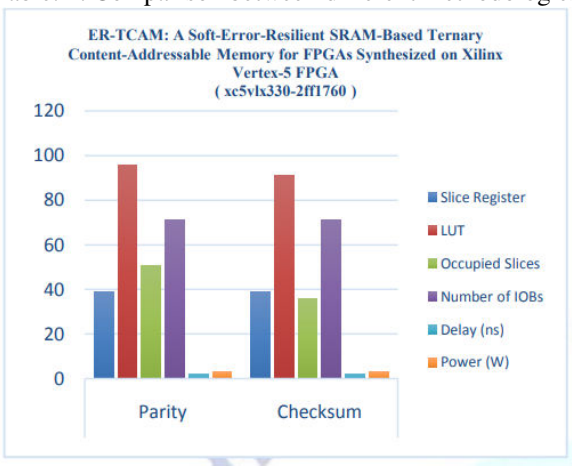

Fig:3. Comparison of Bit Parity and Checksum in Various Characteristics

Error Correction Module: With single-bit parity checking and little latency and processing overhead, error detection is straightforward. The redundant binary-encoded technique is used in the error-correction procedure. In SRAM-based TCAM techniques, a TCAM table is saved for update reasons to correct soft mistakes. While the proposed error-correction system runs in the background, it maintains a high performance, enabling for several search operations to take place at the same time. The results of a simulation for detecting and repairing a single error. The efficiency of the design is evaluated using Verilog code. To test the output of the modified design, we utilized Modalism software to mimic the Verilog code and Xilinx ISE design to synthesize it. The simulation design for BRAM is for a memory capacity of 1024x40. This allows us to compare the efficacy of the Enhanced design to earlier findings.

VII. CONCLUSION

This brief suggests an error-detection method for memory-based and model TCAMs that seeks to utilise duplicate original TCAM information stored on-chip for update purposes in order to safeguard minimum area, critical chain latency, and reaction speed. The suggested approach uses minimal critical route time and logic to identify single-bit parity problems. The suggested error resilience technique, called ER-TCAM, makes use of the binary-encoded TCAM table to fix flaws in SRAM-based TCAMs. The suggested error-correction technique has no effect on data route processing as SRAMs with the TCAM feature are accessible for search operations throughout the background error-correction procedure. With an EDD of 8 ns and a predictable error-correction time of 260 ns, the ER-TCAM was able to do up to 2 billion searches per second on the Vertex-5 FPGA chip. However, the error-correction duration of other current error-correction techniques is incredibly long and nes. The ER-TCAM's EDD is at least twice as high as that of similar devices. Communication networks or other schemes that need SRAM-based TCAMs to expedite particular activities may benefit from the ER-TCAM.

REFERENCES

1. P. He, W. Zhang, H. Guan, K. Salaminian, and G. Xie, "Partial order theory for rapid TCAM updates," IEEE/ACM Trans. Newt., vol. 26, no. 1, pp. 217–230, Feb. 2018.P. He, W. Zhang, H. Guan, K. Salaminian, and G. Xie, "Partial order theory for fast TCAM updates," IEEE/ACM [2]
2. "FAS: Using FPGA to accelerate and secure SDN software switches," Secure. Common. Newt., vol. 2018, Jan. 2018, Art. no. 5650205, W. Fu, T. Li, and Z. Sun.
3. T. Li, H. Liu, and H. Yang, "Design and characterization of SEU hardened circuits for SRAM-based FPGA," IEEE Trans. Very Large-Scale Integral (VLSI) Syst., vol. 27, no. 6, Feb. 2019, pp. 1276–1283.
4. T. Li, H. Yang, H. Zhao, N. Wang, Y. Wei, and Y. Jia, "Investigation into SEU effects and hardening techniques in SRAM based FPGA," in Proc. 17th Eur. Conf. Radiant. Effects Common. Syst. (RADECS), pp. 1–5.
5. A. Ramos, R. G. Toral, P. Reviriego, and J. A. Maestro, "An ALU protection methodology for soft processors on SRAM-based FPGAs," IEEE Trans. Computer., vol. 68, no. 9, March 2019, pp. 1404–1410.
6. S. Pontarelli, P. Reviriego, and J. A. Maestro, "Parallel D-pipeline: A cuckoo hashing implementation for enhanced throughput," IEEE Trans. Computer, vol. 65, no. 1, March 2015, pp. 326–331.
7. P. Reviriego, A. Ullah, and S. Pontarelli, "PR-TCAM: Efficient TCAM emulation on Xilinx FPGAs utilising partial reconfiguration," IEEE Trans. Large-Scale Integral (VLSI) Syst., vol. 27, no. 8, pp. 1952–1956, Mar. 2019.
8. Xilinx Product Guide PG190, Ternary Content Addressable Memory (TCAM) Search IP for SD Net V1.0; Xilinx, San Jose, CA, USA, Nov. 2017.
9. W. Jiang, "Scalable Ternary content addressable memory implementation utilising FPGAs," in Proceedings of the 9th ACM/IEEE Symp. Archit. Newt. Common. Syst., pp. 71–82, 2013.
10. G-AETCAM: Gate-based area-efficient Ternary content-addressable memory on FPGA," IEEE Access, vol. 5, pp. 20785–20790, 2017. M. Irfan and Z. Ullah, "G-AETCAM: Gate-based area-efficient Ternary content-addressable memory on FPGA," IEEE Access, vol. 5, pp. 20785–20790, 2017.
11. P. Reviriego, S. Pontarelli, and A. Ullah, "Error detection and correction in SRAM emulated TCAMs," IEEE Trans. Very Large-Scale Integral (VLSI) Syst., vol. 27, no. 2, February 2019, pp. 486–490.
12. I. Ullah, Z. Ullah, U. Afzaal, and J.-A. Lee, "DURE: An energyand resource-efficient TCAM architecture for FPGAs with dynamic updates," IEEE Trans. Very Large-Scale Integral (VLSI) Syst., vol. 27, no. 6, pp. 1298–1307.
13. "Hybrid partitioned SRAM-based Ternary content addressable memory," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 12, pp. 2969–2979, Dec. 2012.
14. IEEE Access, vol. 6, pp. 19940–19947, 2018. I. Ullah, Z. Ullah, and J.-A. Lee, "Efficient TCAM design based on multipumping-enabled multi ported SRAM on FPGAA," IEEE Access, vol. 6, pp. 19940–19947, 2018.

15. Xilinx is a semiconductor company (2016). User Guide UG474 for 7 Series FPGAs Configurable Logic Blocks (V1.8). On the 25th of June, 2019, I was able to get a hold of some information. [Online]. http://www.xilinx.com is a website where you can learn more about Xilinx.
16. A. M. Keller and M. J. Wirthlin, "Impact of soft mistakes on largescale FPGA cloud computing," in Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, 2019, pp. 272–281.
17. Xilinx, San Jose, CA, USA, 2018. Device Reliability Report.
18. D. Shah and P. Gupta, IEEE Micro, vol. 21, no. 1, pp. 36–47, Jan. 2001, "Fast updating techniques for T
19. CAM."