Journal of Science and Technology Excellence
ISSN: **3108-3102**| Volume 2 Issue 1 January-March 2026
www.jstejournal.com **DOI**: https://doi.org/10.5281/zenodo.19147188 **JSTE-06**

FAULT-TOLERANT AND SECURE TERNARY CONTENT ADDRESSABLE MEMORY FOR HIGH-RELIABILITY APPLICATIONS

[#1]Dr. M. GANESH, *Professor,*
[#2]PENCHALA RAMU, *B.Tech Student,*
[#3]THADALA SINDU PRIYA, *B.Tech Student,*
[#4]PUSALA SAI KRISHNA, *B.Tech Student,*
[#5]OJJA RAVIKUMAR, *B.Tech Student,*
Department of Electronics and Communication Engineering,
TRINITY COLLEGE OF ENGINEERING AND TECHNOLOGY, PEDDAPALLY, TG.

Abstract: Ternary content addressable memory, also known as TCAM, is a vital component of systems that have a high level of reliability. This is due to the fact that it is extremely secure and can tolerate errors. These applications include, but are not limited to, real-time data processing settings, aviation electronics, cybersecurity infrastructure, and networking systems. Because the data is retrieved according to its content rather than its location in memory, TCAM makes it feasible to perform searching operations in parallel that are extremely fast. As a result of this, it is an excellent option for applications that require accurate pattern matching and retrieval immediately. Traditional TCAM systems, on the other hand, have a number of significant shortcomings, such as the potential for data manipulation and unauthorized access, the chance of human error, and the possibility of hardware failure. Additionally, they have the ability to consume an excessive quantity of power throughout their operation.

Keywords: *SRAM (static random-access memory), TCAM (ternary content addressable memory), FPGAs (field programmable gate arrays), Soft errors, and Addressable Memory are all examples of addressable memory.*

1. INTRODUCTION

Data contained in memory cells may be secretly corrupted by charged particles. Silent issues are becoming more prevalent in highly dependable computer systems. The in-memory technology appears to be having some issues. Semiconductors emit charged particles when exposed to radiation. If even one link fails, the entire system will not function. Without TCAM, high-availability networking systems and routers are unable to produce SDN. The TCAM, one of the most noticeable pieces of networking gear, is responsible for tasks like packet forwarding, routing, and classification. Since network components like routers must be designed with an emphasis on soft defect control, soft error avoidance is a challenging TCAM problem to tackle.

Field programmable gate arrays (FPGAs) allow designers and users to make modifications to the device even after it has been constructed due to its programmability. The HDL programming language is used to define the FPGA's parameters. The same term is also used to define application-specific integrated circuits, or ASICs. Current FPGAs can conduct incredibly complicated digital calculations because of their vast network of logic circuits and

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/

RAM blocks. Data saved in the on-board memory of SRAM-based FPGA devices is frequently used to demonstrate TCAM's capabilities. Incorrect match addresses may result from block RAM (BRAM), distributed RAM, or temporary problems. This could lead to an incorrect identification or discrepancy. To ensure that lookups yield precise matching data, the impacted SRAM word must be modified each time a soft error occurs. For SRAM-based TCAM systems, lowering search or critical route time is particularly difficult.

This brief study outlines a quick, easy, and inexpensive method to postpone the search while maintaining the security of SRAM-enabled TCAMs. Regression testing is frequently disregarded because of its ease of use, dependence on multiple tests, and lack of planning or preparation needs. By using a duplicate of the binary-coded TCAM database in SRAM-based TCAM devices, the proposed method compensates for small inaccuracies. Due to its high degree of searching speed, the proposed failed system may conduct a large number of requests even when it is operating in the background. The fast response time of the proposed error-correction technique for the majority of the transaction period leads to an acceptable TCAM configuration for lookups.

2. ARCHITECTURE OF THE PROPOSED ER-TCAM

The ER-TCAM is shown in full in Figure 1. An ECV CPU, an AGU, a blunder read/write controller, and SRAM for storing the TCAM database tables are the parts of the system. With each repetition, a new set of log2D bits is produced by the MOD-D counter. We read the log2D bits from the bottom of the complement to determine which SRAM word is in the sub-block. The beginning of the pertinent SRAM sub-block is indicated by key components of the SRAM ID. The binary-coded integers in the pertinent TCAM database segment are now visible to AGU. Every cycle, a match bit is produced by comparing the TCAM words to the C-bit pattern after they have been read. The ECV can be detected once this bit has been collected for D clock cycles. The ECV can be written over the damaged SRAM word to indicate student achievement once the read/write controller sends an enable signal.

The ER-TCAM facilitates data retrieval from the base SRAMs of the TCAM to enhance search operations during the error-correction phase. The ER-TCAM reads and writes to these SRAMs in a certain order within a single clock cycle to enable them to function as dual-port RAM. Once the ECV has been calculated and written to the write port of the SRAM, the error correcting mechanism disables the ER-TCAM's search and rescue capabilities.

SRAMs that store a binary-encoded TCAM table have significantly lower error rates than SRAMs that renew TCAM tables due to the magnitude of the soft error. If the binary-encoded TCAM database was kept in ECC, the ER-TCAM might be able to continue using SRAM with less storage and error-correction needs.

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/

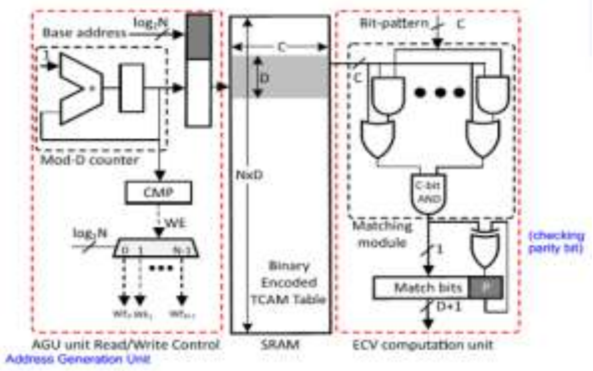


Figure 1: tool proposed for ER-TCAM flaw correction

3. PROPOSED METHODOLOGY

SRAM in an SDN (FPGA) handles both traffic identification and open flow applications. Field engineering arrays are designed by engineers using trinary programmable memory. All FPGA-based devices will require TCAMs based on SRAM in order to reduce the critical path time, search time, and the probability of soft errors. The cost and reaction time of SRAM-based TCAM are reduced by detecting single-bit parity faults using the least expensive required approach. For slow reaction times, the error correction process is made simpler by using a TCAM bar counter SRAM based on binary coded TCAM.

The goal of this study is to demonstrate the feasibility of a 1024x40 SRAM-based TCAM that fixes issues with size, latency, and power consumption. Verilog HDL code and a Xilinx Vertex 5 FPGA were used in its construction. In order to fix errors in SRAM-based TCAMs, redundant data must be maintained. Since a TCAM cell can consist of three SRAM bits, the data stored in SRAMs that can store TCAM is at least 2D C bits in size ("0""00," "1""01," and "x""10").

For usage with later SRAM-based TCAMs, the chip also stores the data from the original TCAM, as was previously described. Consequently, the design-configured data stored in the 2C x D bits of the SRAMs is useless. To deal with intermittent SRAM failures, ER-TCAM implements system-level data redundancy. This core idea is the foundation of ER-TCAM. The TCAM table divisions cannot operate without the two SRAMs seen in the image.

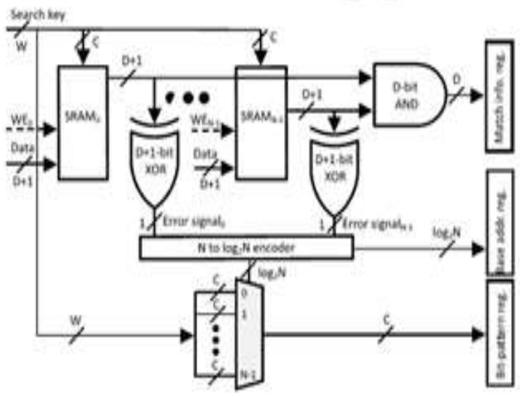


Figure 2: ER-TCAM suggested architecture error detection

To find an SBU, the ER-TCAM adds one bit to each SRAM word, as shown in the picture.

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/

Every SRAM component was first subjected to a flaw testing procedure during lookups. The ER-TCAM looks for further information in the TCAM database to fix misspelled words.
Figure 1 displays the suggested ER-TCAM configuration for error detection. When a search key-based lookup is performed, the SRAM reading bit is EX-ORed to produce an error signal.
To identify which SRAM is malfunctioning, the TCAM system transforms the N error signals from the SRAMs into a log2N-bit error message. The error code and any relevant search-key bit patterns are sent to the error-correction module. For every SRAM failure, the system then generates an error message of length log2N bits. The error code and any pertinent search-key bit patterns are passed to the error-fixing program.

4. METHODN FOR CALCULATING THE CHECKSUM

A common belief is that checksums are more effective than LRC, VRC, and CRC at identifying errors in higher-layer methods. It increases the effectiveness of high-level procedures in identifying errors. Including a checksum generator in communication receivers is standard procedure. A checksum is verified at the recipient's end. The sender divides the data into identical n-bit chunks using a hash technique. This item is around sixteen inches long.
The one's complement technique is used to combine all of these disparate elements to create a whole. Adding to the newly generated bit is the final step. Prior to transmitting the antecedent data unit to the receiving system, we calculate and apply the checksum The user then provides the checksum verifier with the collected data and the checksum.
The checksum analyzer will add all of the identically sized sections if the data unit is split up into several pieces. One of these components includes the checksum as an essential component. recognizing that a task is perfect after doing it. The accuracy of the data is shown by a corrected result of "0". The result is not zero if the receiver rejects the input due to the dataset's inaccuracy.

5. TCAM ON FPGAS BASED ON SRAM

Modern FPGAs are able to generate TCAM with the use of on-chip SRAM. A "1" is added to SRAM to create a "1" quantity; this 2X1 RAM also maintains a "0" TCAM authoritarian leadership. A "1" is kept in both SRAM and RAM, maintaining a "x" state.
A 1-bit SRAM with 2C locations can be used to set up a C-bit TCAM arrangement, enabling a one-on-one TCAM. For each word, TCAM records all possible C-bit patterns that it matches or does not match. The C-bit TCAM architecture is represented by the address words in an SRAM. A C-bit wide SRAM with 2C locations can be used to generate a B-word TCAM table. SRAM-based TCAMs were separated because they were unable to handle the extended bit patterns that the researchers required. 2C D-sized D-bit structures are used by TCAM.
Sort them in the following order after splitting them into C-bit segments that are W bits wide. only a basic SRAM-based TCAM configuration. Two 4-by-2 portions comprise the 4-bit patterns of a 4-word deep TCAM. Two 4-by-4 SRAMs are utilized to activate them. Note the

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/

index number (1001) of the digits. The second phrase (1001) and the third word (1100) are obtained by bits 0 and 1 when reading from the first SRAM. According to rule R0, a match occurs when all of the SRAM bits are ANDed together, yielding 1000.

There are numerous methods for putting TCAM on FPGAs. Flip flops (FFs) from field-programmable gate array (FPGA) logic units are used to construct miniature TCAMs. It was previously stated that shallow SRAMs are utilized for TCAM manufacture because they are more energy-efficient memory processors, and that the majority of contemporary TCAMs constructed on FPGAs use BRAM, which stands for distribution RAM. The depth of Xilinx BRAM architectures is up to 512 bits. As a result, only 29 BRAM bits can accommodate nine TCAM bits.

Considering a piece width of 9 bits in the TCAM, the optimal configuration sizes for 18kb and 36kb BRAMs are 51236 and 51272, respectively. These TCAMs use distributed RAM and have 30 TCAM bits per SLICEM. To achieve the minimum requirements, the four 64-bit LUTRAMs are all reduced to 32-bit 6-port dual-channel RAMs. 853 bits of distribution RAM are needed for a single bit of TCAM, while 256 bits are needed for 30 bits. SRAMs based on distributed RAM should have a TCAM chunk width of five and a depth of thirty-two bits. While LUTRAMs are utilized for certain system components, TCAM necessitates the development of BRAMs.

Soft faults are simpler to identify when TCAM settings have more resources available. The total space of a 28-nanometer FPGA is composed of 1% BRAMs, 5% LUTRAMs, and 34% slice files. This leads to a large decrease in the number of SEUs. Because of this, BRAM-based TCAM systems are less dependable and more prone to minor errors than their distributed RAM and SR equivalents. The BRAM failure rates of the FPGA device were used to emphasize the project's real-time soft error rate for each event in this scenario.

Vertex-5 has a failure rate of 27%. TCAM words have the wildcard state "x," which means that when two words' ranges overlap, a single-bit word can match up to several phrases.

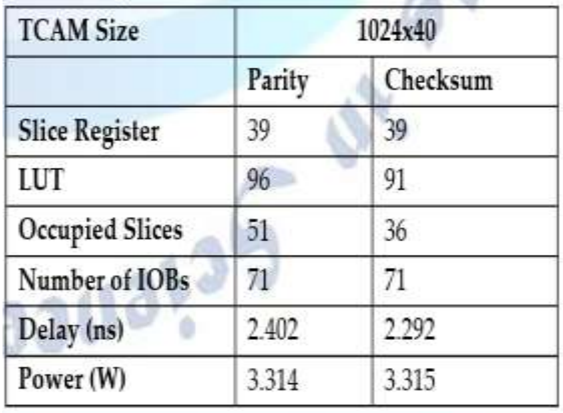
Only the lowest-order word is transmitted when multiple words match. Setting priorities is essential for organizing TCAM concepts. Lower memory addresses are the recipients of priority words. The words in the TCAM table may vary along with the word order when a TCAM word is modified. Update steps complete the lookup procedure after a precise match between the incoming terms is found.

As open flow and software-defined networking become more popular, switch replacements will occur more frequently. SRAM-based TCAM solutions on FPGAs can simulate the TCAM function's behavior by using a TCAM item in the TCAM data table. This must be performed many times in order for the order constraint to be effective. By reusing data from previously stored binary-encoded TCAM material, the proposed ER-TCAM resolves minor problems with reproducing TCAMs in SRAMs.

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/

6. EXPERIMENTAL RESULTS AND ANALYSIS

Table: 1. a comparison of existing techniques is provided.



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

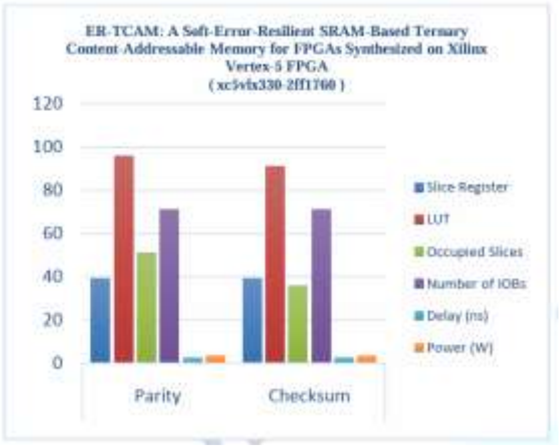


Fig:3. A Comparison of Bit Parity and Checksum Characteristics

A comparison of the bit parity and checksum properties is presented in Figure 3. Single-bit parity testing requires very little computing resources and can be used to rapidly and simply identify errors in error-correcting modules. Binary decoding is frequently used in mistake correction. TCAM techniques store a customizable TCAM table that can tolerate minor inaccuracies in SRAM.

The proposed error-correction approach can process many queries simultaneously while operating in the background in an invisible manner. A single error has been fixed in the simulation results. Verilog code is used to test the design's functionality. To validate the new architecture, we used Modalism for Verilog code simulation and Xilinx ISE for design implementation. Simulation data can be stored in 1024 by 40 bytes using the BRAM architecture. As a result, we can assess how well the Enhanced plan performs in comparison to the current situation.

7. CONCLUSION

According to this study, on-chip copies of the original data should be used to update memory-based and model TCAMs in order to identify and correct any defects. Reactions are now faster due to the reduced latency in critical links and the space savings. The suggested method detects single-bit parity flaws using minimal inference and ideal travel time. Furthermore, the proposed error-correction approach does not interfere with data channel processing. SRAMs

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/

that support TCAMs can carry out search operations during the background error-correction process. The binary-encoded TCAM table is used by the error resilience technology, ER-TCAM, to overcome issues with SRAM-based TCAMs. The Vertex-5 FPGA device's ER-TCAM has a known error-correction time of 260 nanoseconds and an 8 nanosecond EDD, which allows it to conduct up to 2 billion searches per second. Nevertheless, certain error-correction algorithms are not particularly effective and fix errors quite slowly. The EDD of the ER-TCAM is twice as high as that of its predecessors. The ER-TCAM can be used in systems that use SRAM-based TCAMs to speed up specific operations, such as data networks.

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. [2] P. He, W. Zhang,
2. H. Guan, K. Salaminian, and G. Xie, "Partial order theory for fast TCAM updates," IEEE/ACM
3. "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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. Xilinx Product Guide PG190, Ternary Content Addressable Memory (TCAM) Search IP for SD Net V1.0; Xilinx, San Jose, CA, USA, Nov. 2017.
10. 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.
11. 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.

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/