



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 Issue: IX Month of publication: September 2020

DOI: <https://doi.org/10.22214/ijraset.2020.31320>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Performance Improvement in SRAM Emulated TCAM

B. Dharani¹, M. Durga², S. Keshavavardhini³, N. Indhuja⁴, Dr. N. Shanmugasundaram⁵
⁵Professor, ^{1,2,3,4}ECE Department, Sri Eshwar College of Engineering, Coimbatore - 641 202.

Abstract: *Static Random Access Memory (SRAM) emulated Ternary Content Addressable Memory (TCAM) is less memory efficient due to the less capacity of the physical address in SRAM. Here, SRAM is divided into many memory blocks for storing the address of the search word matched against the array of cells. Since there is large number of transistors in the TCAM, it is expensive but has reduced storage efficiency. It is implemented in Field Programmable Gate Array (FPGA). TCAMs play a major role in routing of networks. TCAM is a hardware device used for accelerating the speed of the packet forwarding. But this may be affected by single bit or multiple bit errors. In brief, the memory is safely protected and contents are safely searched by emulating it without corrupting the stored bits by the use of three dimensional parity rule based search algorithm.*

Keywords: *Single-bit errors, SRAM emulated TCAM, search speed, performance improvement*

I. INTRODUCTION

CAM is a specific device for various applications involving Asynchronous Transfer Mode (ATM), communication networks, LAN bridges/switches, databases, lookup tables and tag directories, due to its high-speed data search capability. Content addressable memory is a chip that provides fast table lookups, most notably in network routers and switches. SRAM is not used for high-speed search operation but CAM is used as an associative memory performs all operations of SRAM with high speed and accuracy. It also plays a major role in network devices, image processing etc.

Ternary Content Addressable memories (TCAMs) compare the search data with the already stored input data and send the address as the output in parallel searching process and it also uses don't care term(x) either '1' or '0'. TCAM operates with high power consumption. The data cells stores binary states and the mask cells store don't care state. Emulation of SRAM in TCAM is done using logic blocks in FPGA. But this emulation causes various errors and it can be corrected using extra memory bits. Various errors occur in SRAM emulation namely soft errors and firm (hard) errors. The external sources cause the data to be changed in soft errors and makes system gets corrupted. . It can be remedied by cold booting the computer. It will not damage the system's hardware; the only damage is to the data being processed. This may be a single bit or multiple bit errors. The soft errors may cause severe problem in networking applications. TCAMs are easily affected by soft errors and protecting it is a serious problem since error correction codes for this emulation is more tedious. So, the proposed algorithm definitely create improvement in the performance of the SRAM emulated TCAM. Logic blocks of FPGA are needed to emulate TCAM cell.

CAM works in a single clock-cycle. CAM constantly generates empty information and also an empty address for use if it is going to bring a new missing block into cache and store the corresponding tag into the CAM. If multiple locations of the CAM are empty, the address of the lowest addressed empty location can be produced. This is to be deterministic instead of being random in choosing an empty address as logic to make it random is more expensive. CAM sums up all the memory algorithms for fast searching of the required unique word. In the architecture of CAM, the row cells are connected as parallel or series to make the match line architecture. Every TCAM cell consists of two SRAM cells to store the don't care bit. It is used in lookup table calculation operation. The input data is mapped to the entire stored data and then the output is obtained in the form of address location. Here the priority encoder decides the perfect match using highest priority. SRAM emulated TCAM means TCAM contents are addressed using SRAM cell. It results in marvellous growth in the usage of memory cells.

The searchword is broadcasted into searchline path and the stored word is mapped to a matchline and this is send to the priority encoder which generates match locations. In this paper, we survey the performance of TCAM when emulated using SRAM. The experimental result proved that the proposed design significantly improves the performance of the system.

II. RELATED WORK

Various schemes are proposed to improve the performance of the searching operation in the SRAM emulated TCAM. A CAM is a memory that implements the lookup-table function in a single clock cycle using dedicated comparison circuitry [1].Block RAMs and distributed RAMs are used in emulating TCAM. The proposed solution operates the configured simple dual-port BRAMs of the design as multiported SRAM using the multipumping technique, by clocking them with a higher internal clock frequency to access the sub-blocks of the BRAM in one system cycle [8].The higher power consumption is reduced using [4].

TCAM is emulated using Altera and Xilinx primitive BRAMs in ETCAM. E-TCAM logically divides the classical TCAM table along columns and rows into hybrid TCAM subtables and then maps them to their corresponding memory blocks. During search operation, the memory blocks are accessed by their corresponding subwords of the input word and a match address is produced [7]. TCAM is divided into small TCAMs and implements Z-TCAM and E-TCAM. The functionality of the TCAM uses SRAM and various logic blocks in circuitry. The main function of any memory system is writing and reading the lookup data. Random access memory (RAM) is a volatile memory. It searches lookup table data serially in the memory array, and it requires more clock cycles to search the data. RAM is the inverted form of CAM. Thus, researches concentrate in TCAM designing to minimize power without degrading the performance. The CAM based on system and architectural levels are defined almost very vividly.

Previous works majorly based on throughput improvement and memory utilization and uses lot of SRAM resources. The protection of TCAMs has been proposed in various forms, for example, in the rules replication processor in Bloom filters structure [4]. FPGA is used in implementing any software defined network systems with high speed and accuracy. TCAM functionality can be implemented using logic blocks and memory resources [11]. But it does not contain any blocks of CAM and TCAM. Cuckoo hashing is used in emulating binary CAMs with very low cost [8]. TCAM emulation takes place with the logic and memory resources. So many proposals are experimented.

In few proposals, TCAM memory cells can be implemented with FPGA flip-flops. SRAM memories are embedded in the FPGA [4]-[8] and are preferred and implemented by FPGA users. FPGAs block RAMs (BRAMs) are needed for each single TCAM bit. The distributed RAMs need 6 bits for each TCAM bit. This means it requires large number of memory bits and thus the probability of corruption of data in a system increases. The protection majorly needs additional bits for error correction method using parity check method [8]. It can also be done by using triple modular redundancy that triplicates the flip-flops and has the additional voting logic to correct errors, thus needs large resources. In this brief, the exploitation of TCAM cells is emulated using the stored contents. In particular, when memory protection is with a parity bit to detect single-bit errors, the proposed scheme will be able to correct it also. This makes the proposal provides more efficient performance.

III. METHODOLOGY

A. TCAM implementation using FPGA logic blocks

The two methods for using FPGA are logic resources along with flip flops or block memories for emulation operation. This speeds up the error detection mechanism with exponential growth in the system performance. In the first idea, the bits can be stored in flip-flops and each and every bit can take three possible values: 0, 1, and x. Here one flip-flop can store either '0' or '1' and another one flip-flop can store the don't care bit by masking itself [9]. Most of the programmable logics are used primitively to speed up the process rather than concentrating in the power consumption and its circuitry size and efficiency. ModelSim's architecture allows platform independent compile with the outstanding performance of native compiled code. An easy-to-use graphical user interface enables you to quickly identify and debug problems, aided by dynamically updated windows. For example, selecting a design region in the Structure window automatically updates the Source, Signals, Process, and Variables windows. These cross linked ModelSim windows create a powerful easy-to-use debug environment. Once a problem is found, you can edit, recompile, and re-simulate without leaving the simulator. ModelSim PE fully supports the VHDL and Verilog language standards. You can simulate behavioral, RTL, and gate-level code separately or simultaneously. ModelSim PE also supports all ASIC and FPGA libraries, ensuring accurate timing simulations. The major contribution of this work is reflected on the performance improvement with the lead of finding an error in the array of stored data and corrects it with a greater accuracy and speed. It should also not affect the memory storage efficiency by somehow reducing the power consumption to a little amount that is dissipated in the form of heat in the circuitry basis. The throughput is also a very important factor for the improvement in performance.

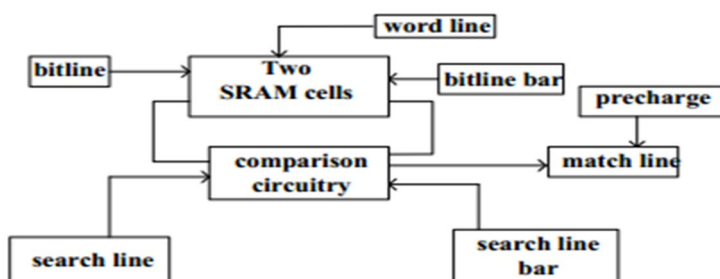


Figure 1 - Block diagram of a TCAM Cell

In VHDL one generally distinguishes between the external view of a module and its internal description. The external view is reflected in the entity declaration, which represents an interface description of a 'black box'. The important part of this interface description consists of signals over which the individual modules communicate with each other. The internal view of a module and, therefore, its functionality is described in the architecture body. This can be achieved in various ways. One possibility is given by coding a behavioral description with a set of concurrent or sequential statements. Another possibility is a structural description, which serves as a base for the hierarchically designed circuit architectures. Naturally, these two kinds of architectures can also be combined. The lowest hierarchy level, however, must consist of behavioral descriptions.

Table.1 Resource Utilization

Rules x Width	Technique	LUTs	BRAMs
64 x 40	Unprotected	129	5
64 x 40	Proposed	219	5
64 x 40	SEC	1218	5
512 x 40	Unprotected	512	38
512 x 40	Proposed	1028	38
512 x 40	SEC	4807	38
1024 x 40	Unprotected	1027	73
1024 x 40	Proposed	2055	73
1024 x 40	SEC	11083	73

The resources utilized are tabulated in the above table. The resources utilized may be LUTs and BRAMs. The delay can also be taken and slices for the resources also noted down. The weights of the memory in a $2b$ positions are quantified for single bit error correction.

Table.2 Memory Weights And Errors Corrected

Weight 0	All errors corrected
Weight 1	Except a bit set to position 1 are corrected
Weight 2	Except a bit set to position 2 are corrected
Weight 4 or more	All errors corrected

B. Algorithm

The algorithm for the proposed model for error correction in SRAM emulated TCAM is given below.

Define Library function

Define entity, port

Define type of memory

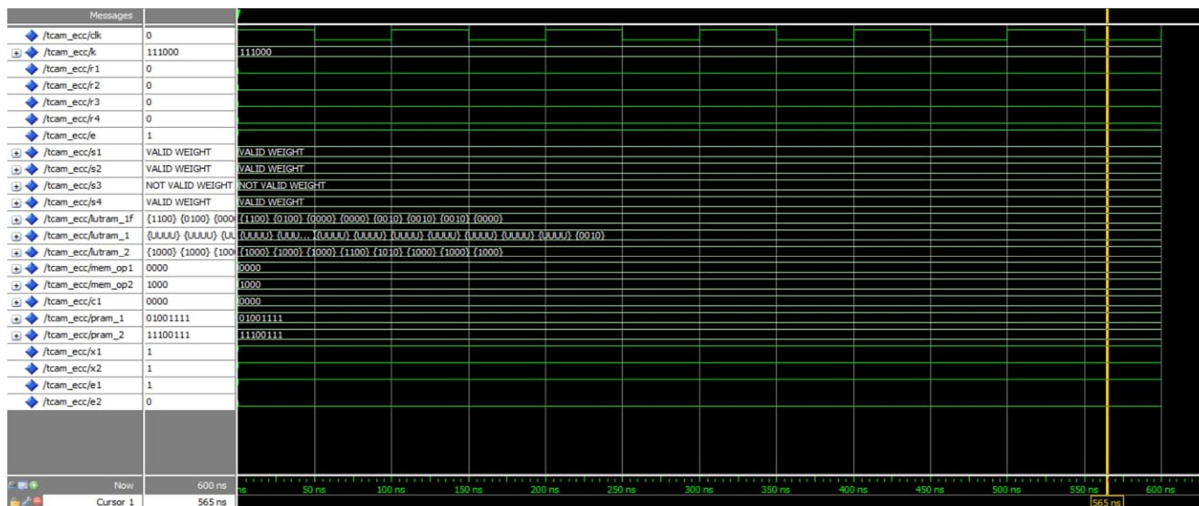
Check whether weight is valid or not.

- 1) *Require:* Parity error detection
 - a) Read memory and compute diagonal weights
 - b) if there is a diagonal with illegal weight then
 - c) Correct that bit
 - d) return error corrected
 - e) end if
 - f) if there is a diagonal with zero weight then
 - g) Read another memory
 - h) Compute the weights of those columns
 - i) if any diagonal has a non zero weight on the other memory then
 - j) Correct that bit
 - k) return error corrected
 - l) end if

- m) end if
- n) if there are diagonal with weight one then
- o) Read another memory
- p) Compute the weights of those diagonals
- q) if any has zero weight on the other memory then
- r) Correct that bit
- s) return error corrected
- t) end if
- u) end if
- v) if there are diagonal with weight two then
- w) Read the memory
- x) Check the patterns of those diagonals
- y) if any has an illegal pattern then
- z) Correct that bit
- aa) return error corrected
- bb) end if
- cc) end if
- dd) return uncorrected error

IV. SIMULATION RESULTS AND DISCUSSION:

ModelSim PE, our entry-level simulator, offers VHDL, Verilog, or mixed-language simulation. Coupled with the most popular HDL debugging capabilities in the industry. It is known for delivering high performance, ease of use, and outstanding product support. It simulate behavioral, RTL, and gate-level code separately or simultaneously. ModelSim PE fully supports the VHDL and Verilog language standards. You can simulate behavioral, RTL, and gate-level code separately or simultaneously. ModelSim PE also supports all ASIC and FPGA libraries, ensuring accurate timing simulations. Performance improvement SRAM emulated TCAM can be easily simulated in ModelSim with accuracy in results. So, this software is preferred. Version used is ModelSim 6.3f. Inputs are rows(r1,r2,r3,r4), columns, clock(clk). Outputs are weights whether valid or not according to $2b$ positions.



From this waveform, we infer that the SRAM emulated TCAM can produce the error free output by using the diagonal parity algorithm. This enables memory efficient and cost efficient scheme with huge profit in its idea. The inputs assumed are 4 rows, one clock, lutrams, memory. The output inferred from this simulation is the weights of the memory with which they are selected for validity for ensuring the memory efficiency and henceforth leading to performance efficient SRAM emulated TCAM. In this method, the three dimensional parity rule based algorithm is used i.e., row parity, column parity, diagonal parity altogether to check whether there is error in it. If found, it is corrected by calculating number of ones using 1248 rule based algorithm. Let us now quantify the fraction of single-bit error patterns that can be corrected for each weight in a memory of $2b$ positions.

1) Weight zero: all patterns can be corrected. 2) Weight one: all except those that set a bit to one for a position with an address at distance one, this corresponds to $1 - b/2b$. 3) Weight two: all patterns can be corrected except the two that set a position with a one to a zero, this corresponds to $1 - 2/2b$. 4) Weight four or larger: all patterns can be corrected. It can be seen that most of the error patterns are corrected.

V. CONCLUSION AND FUTURE WORK

A performance efficient technique to protect the SRAMs used to emulate TCAMs on FPGAs has been proposed in this work. This technique is used to correct most single-bit error patterns when the memories are protected with a parity bit. This technique will provide improvement in the performance. The idea presented can be extended to other memory configurations. It can be used to detect errors by periodically changing the contents to check their correctness. Memory can be protected with efficient code that can detect several errors and correct it.

REFERENCES

- [1] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey", *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp.712–727, Mar. 2006.
- [2] A. L. Silburt, A. Evans, I. Perryman, S. J. Wen, and D. Alexandrescu, "Design for soft error resiliency in Internet core routers", *IEEE Trans. Nucl. Sci.*, vol. 56, no. 6, pp.3551–3555, Dec. 2009.
- [3] F. Yu, R. H. Katz, and T. V. Lakshman, "Efficient multimatch packet classification and lookup with TCAM", *IEEE Micro*, vol. 25, no. 1, pp. 50–59, Jan./Feb. 2005.
- [4] S. Pontarelli, M. Ottavi, A. Evans, and S. Wen, "Error detection in ternary CAMs using Bloom filters" in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2013, pp.1474–1479.
- [5] 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.
- [6] W. Jiang, "Scalable ternary content addressable memory implementation using FPGAs" in *Proc. ACM ANCS*, San Jose, CA, USA, Oct. 2013, pp.71–82.
- [7] Z. Ullah, M. K. Jaiswal, and R. C. C. Cheung, "E-TCAM: An efficient SRAM-based architecture for TCAM," *Circuits, Syst., Signal Process.*, vol. 33, no. 10, pp.3123–3144, Oct. 2014.
- [8] I. Ullah, Z. Ullah, and J.-A. Lee, "Efficient TCAM design based on multipumping-enabled multiported SRAM on FPGA," *IEEE Access*, vol. 6, pp.19940–19947, 2018.
- [9] A. Ahmed, K. Park, and S. Baeg, "Resource-efficient SRAM-based ternary content addressable memory," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp.1583–1587, April 2017.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)