IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

Error Tolerant and Performance Enhanced TCAM for Network Applications

Dr.S.Kavitha[1], S.Sreekalpa[2]

Professor, Department of ECE, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India[1]

PG Scholar, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India[2]

Abstract: Content Addressable Memory (CAM) is a type of solid-state memory in which data are accessed by their contents rather than physic al locations. Ternary Content Addressable Memories (TCAMs) are special memories which are widely used in high-speed network applications such as routers, firewalls, and network address translators. In high-reliability network applications such as aerospace and defense systems, soft-error tolerant TCAMs are indispensable to prevent data corruption or faults caused by radiation. The proposed work uses a novel soft-error tolerant TCAM for multiple-bit-flip errors using partial keys and parity based logic for search time reduction. The proposed TCAM corrects multiple bit-flip errors and enhances the tolerance of the TCAM against soft errors. TCAM detects multiple-bit-flip errors by the generated X-keys using the X look-up. If the keys match the different locations, then a soft error is suspected and TCAM refreshes the TCAM words by using the backup ECC-SRAM. The proposed CAM also uses parity an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, i.e., odd or even number of "1"s. This additional parity bit reduces the sensing delay and boosts the driving strength of the 1-mismatch case by half. The hardware overhead of the proposed TCAM is small due to the use of a single TCAM. The parity based TCAM can be easily implemented and is useful for fault-tolerant packet classifiers.

Keywords: TCAM, Multi-bit errors, ATM, SRAM, Parity

I. INTRODUCTION

A CAM is a memory device used in applications requiring fast searches of a database, list, or pattern. Image or voice systems, computer and communication systems are all users of CAM. CAMs have a performance advantage over other memory search algorithms [1]. This is due to the simultaneous comparison of the desired information against the entire list of pre-stored entries. Especially for networking devices, requirements are necessary to filter, translate (encode-decode), forward, and deliver data to diverse destinations such as servers, computers, smart phones, and smart wearable devices that require real-time processing. One of the most important storage and forwarding components in the network devices is a Ternary Content Addressable Memory (TCAM).

Generally, soft errors are caused by ionizing particles, such as alpha particles, protons, heavy ions, neutrons, etc; in some cases, these ionizing particles are generated by radioactive atoms [2]. Several works investigated the effects of scaling down the size of transistors. A work previously shows that moving from the 130 nm process to the 22 nm process increases the soft error rates by up to 7 times. Furthermore, soft errors tend to cause more serious problems in low-power devices [3]. Some works analyze soft errors on combinational and sequential circuits. In [4], a model to quantify Soft Error Rates (SER) of combinational circuits in the changes of technology was presented by considering the effects of CMOS device scaling and by increasing depth of processor pipelines. However, previous hardware and software methods still suffer from severe drawbacks. Firstly, hardware methods are very expensive to implement since they have to modify the TCAM circuits. Secondly, for software methods, researchers are still looking for more efficient way to tackle the soft-error problem in TCAMs. A TCAM with the optimized scrubbing interval is proposed against soft errors. However, this scheme ignores the fact that some keys are more frequently used than others in the TCAM. This scheme can lead to more faults when the frequent keys hit the upset word caused by a soft error. A TCAM checker is proposed that considers frequent keys through comparing the matched words caused by a soft error. However, this scheme uses two TCAMs, which doubles the hardware overhead and the power dissipation.

The rest of the paper is organized as follows: Section II defines the basics of CAM; Section III explains Performance enhanced TCAM; Section IV describes the proposed Error Tolerant TCAM; Section V shows evaluation results and Section VI concludes the paper.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

II. CAM AND ITS PROPERTIES

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. Figure 1 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 matchlines are inputs to an encoder that generates the address corresponding to the match location.

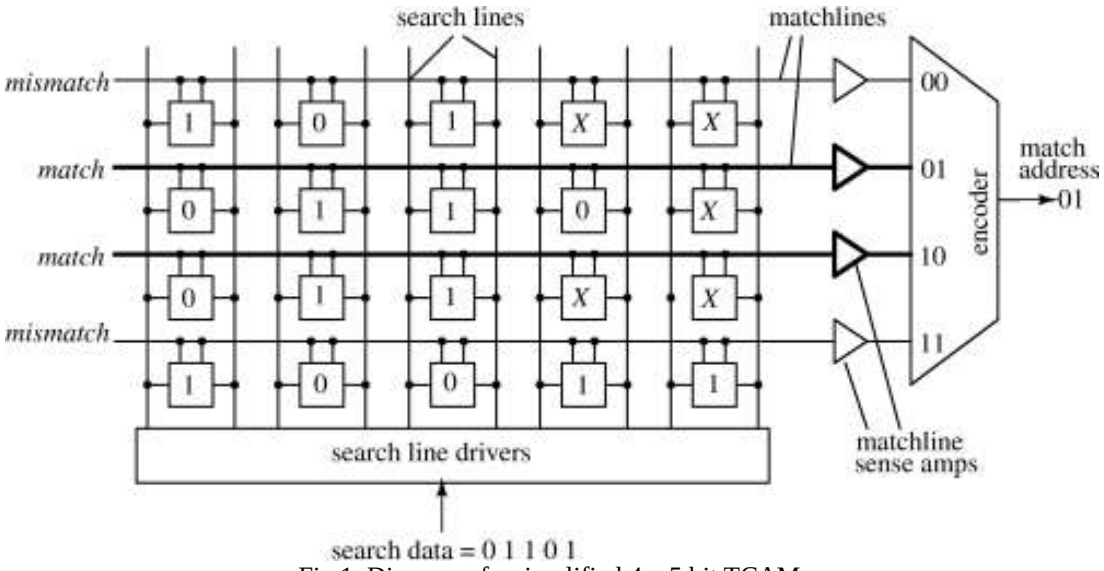

Fig.1. Diagram of a simplified 4 x 5 bit TCAM

Figure 2(a) displays a conventional SRAM core cell that stores data using positive feedback in back-to-back inverters. Two access transistors connect the bitlines, bl and /bl, to the storage nodes under control of the wordline, wl. Data can be read from the cell or written into the cell through the bitlines. We use this differential cell as the storage for building CAM cells. Figure 2(b) depicts a conventional binary CAM (BCAM) cell with the matchline denoted ml and the differential search lines denoted sl and /sl. The figure also lists the truth value, T, stored in the cell based on the values of d and d. Read and write access circuitry is omitted for clarity in this figure and subsequent CAM core cell figures. For a binary CAM, we store a single bit differentially. The comparison circuitry attached to the storage cell performs a comparison between the data on the search lines (sl and /sl) and the data in the binary cell with an XNOR operation (ml = ! (d XOR sl)). A mismatch in a cell creates a path to ground from the matchline through one of the series transistor pairs. A match of d and sl disconnects the matchline from ground.

Figure 2(c) shows a ternary CAM (TCAM) cell. The TCAM cell stores an extra state compared to the binary CAM, the don't care state, labeled X, which necessitates two independent bits of storage. When a don't care is stored in the cell, a match occurs for that bit regardless of the search data.

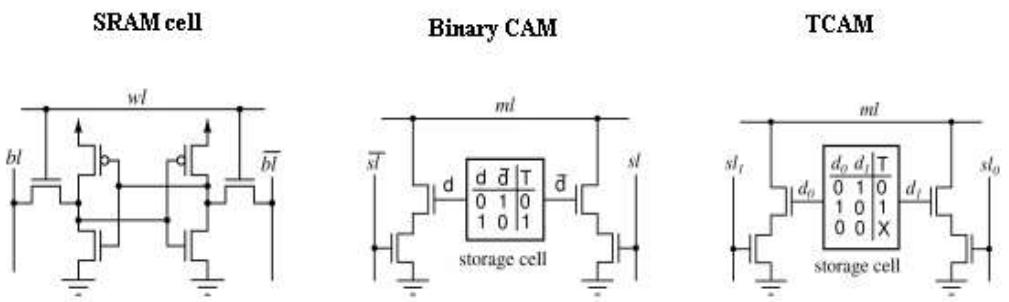

Fig.2. (a) SRAM Cell (b) Binary-CAM (c) TCAM

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

The figure shows that the TCAM cell stores X when $d_0 = d_1 = 0$. The state $d_0 = d_1 = 1$ is undefined and is never used.
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.

III. PERFORMANCE ENHANCED TCAM

The parity bit based CAM design is shown in Fig. 3 consisting of the original data segment and an extra one-bit segment, derived from the actual data bits. We only obtain the parity bit, i.e., odd or even number of "1"s. The obtained parity bit is placed directly to the corresponding word. Thus the new architecture has the same interface as the conventional CAM with one extra bit. During the search operation, there is only one single stage as in conventional CAM.

Fig.3. Parity based CAM Memory

However, this additional parity bit reduces the sensing delay and boosts the driving strength of the 1-mismatch case (which is the worst case) by half. In the case of a matched in the data segment, the parity bits of the search and the stored word is the same, thus the overall word returns a match. When 1 mismatch occurs in the data segment numbers of "1"s in the stored and search word must be different by 1. As a result, the corresponding parity bits are different. Therefore now we have two mismatches (one from the parity bit and one from the data bits). If there are two mismatches in the data segment the parity bits are the same and overall we have two mismatches. With more mismatches, we can ignore these cases as they are not crucial cases. The sense amplifier now only has to identify between the 2-mismatch cases and the matched cases. Since the driving capability of the 2-mismatch word is twice as strong as that of the 1-mismatch word, the proposed design greatly improves the search speed.

IV. THE PROPOSED ERROR TOLERANT TCAM

Fig.4. shows the proposed error tolerant TCAM and it consists of: 1) A novel scheme, called Parity-TCAM, is proposed that can detect and correct multiple-bit-flip soft errors in a TCAM. 2) The parity-TCAM requires one additional cell in each location. 3) The proposed design uses only one TCAM. 4) The soft-error tolerance of TCAM outperforms existing schemes.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

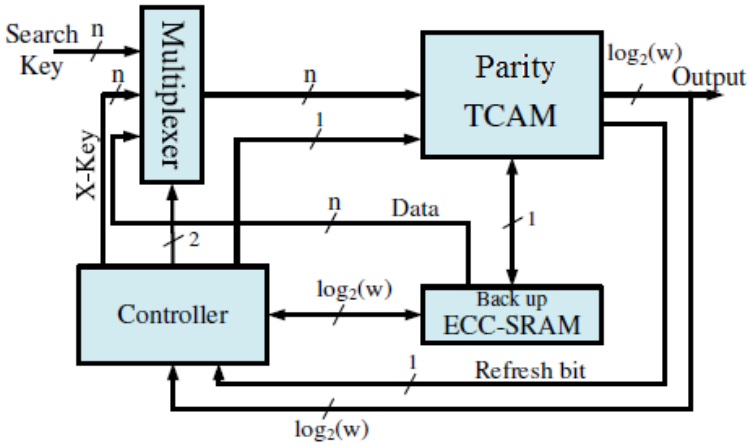

Fig.4. Proposed Error Tolerant TCAM

The proposed scheme consists of several blocks: a Parity TCAM, a multiplexer, a controller, as well as a backup ECC-SRAM. The Parity-TCAM has two modes: the normal mode and the test mode. In normal mode, the TCAM looks up a search key in parallel. In test mode, the X-TCAM generates partial don't-care keys (Xkeys) and detects soft errors. It works sequentially as follows: First, the TCAM starts by applying several X-keys to the TCAM. Don't-care bits are inserted in the X-keys using X look-up memory, and they are expected to match the words that contain soft errors. After applying the X-keys to the TCAM, parity-TCAM records all the returned indices. After that, if the indices are equal, then no soft error is detected; otherwise a soft error is detected.

A.ATM protocol

CAMs can be used in Asynchronous Transfer Mode (ATM) switching network components as a translation table. Since ATM networks are connection-oriented, virtual circuits need to be set up across them prior to any data transfer. ATM approximately maps to the three lowest layers of the ISO-OSI reference model: network layer, data link layer, and physical layer. ATM is a core protocol used over the SONET/SDH backbone of the public switched telephone network (PSTN) and Integrated Services Digital Network (ISDN), but its use is declining in favour of all IP. ATM provides functionality that is similar to both circuit switching and packet switching networks: ATM uses asynchronous time-division multiplexing, and encodes data into small, fixed-sized packets (ISO-OSI frames) called cells. This differs from approaches such as the Internet Protocol or Ethernet that use variable sized packets and frames. ATM uses a connection-oriented model in which a virtual circuit must be established between two endpoints before the actual data exchange begins. These virtual circuits may be "permanent", i.e. dedicated connections that are usually preconfigured by the service provider, or "switched", i.e. set up on a per-call basis using signaling and disconnected when the call is terminated. Use of ATM technology was eventually largely superseded by Internet Protocol (IP)-only technology. Wireless and mobile ATM never established a significant foothold.

There are two kinds of ATM virtual circuits: Virtual Path (identified by a virtual path identifier or VPI) and Channel Path (identified by a channel path identifier or VCI). VCI/VPI values are localized; each segment of the total connection has unique VPI/VCI combinations. Whenever an ATM cell travels through a switch, its VPI/VCI value has to be changed into the value used for the next segment of connection. This process is called VPI/VCI translation. Since speed is an important factor in ATM network, the speed at which this translation occurs forms a critical
part of the network's overall performance. CAM can act as an address translator in an ATM switch and perform the VPI/VCI translation very quickly. During the translation process, the CAM takes incoming VPI/VCI values in ATM cell headers and generates addresses that access data in an external RAM (since standard CAM architectures cannot support the required capacity, a CAM/RAM combination enables the realization of multi- megabit translation tables with fully-parallel search capability). VPI/VCI fields from the ATM cell header are compared against a list of current connections stored in the CAM array. As a result of the comparison, CAM generates an address that is used to access an external RAM where VPI/VCI mapping data and other connection information is stored. The ATM controller modifies the cell header using the VPI/VCI data from the RAM, and the cell is sent to the switch.

B. Structure of an ATM cell

An ATM cell consists of a 5-byte header and a 48-byte payload as shown in Fig.5. ATM defines two different cell formats: user–network interface (UNI) and network–network (NNI). Most ATM links use UNI cell format. GFC = The generic flow control (GFC) field is a 4-bit field that was originally added to support the connection of ATM networks to shared access networks such as a distributed queue dual bus (DQDB) ring. The GFC field was designed to give the User-Network Interface (UNI) 4 bits in which to negotiate multiplexing and flow control among the cells of various

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

ATM connections. However, the use and exact values of the GFC field have not been standardized, and the field is always set to 0000.

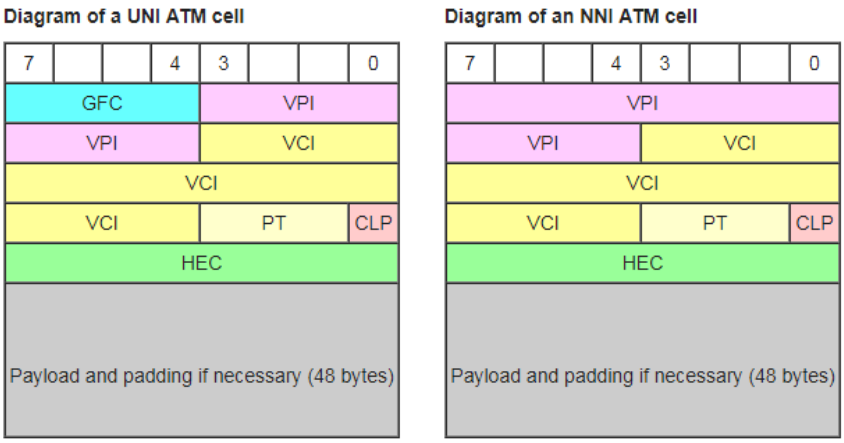

Fig.4. ATM Cell Format

VPI = Virtual path identifier (8 bits UNI, or 12 bits NNI)
VCI = Virtual channel identifier (16 bits)
PT = Payload type (3 bits)
PT bit 3 (msbit): Network management cell. If 0, user data cell and the following apply:
PT bit 2: Explicit forward congestion indication (EFCI); 1 = network congestion experienced.
PT bit 1 (lsbit): ATM user-to-user (AAU) bit. Used by AAL5 to indicate packet boundaries.
CLP = Cell loss priority (1-bit)
HEC = Header error control (8-bit CRC, polynomial = $X^8 + X^2 + X + 1$).

V. EVALUATION RESULTS

Table 1 show that various parameters like area, power and delay of existing and proposed TCAM along with ATM application. Power and performance of the parity enhanced TCAM is better compared with existing designs. The design when applied to ATM applications results in better power reduction with very small area overhead.

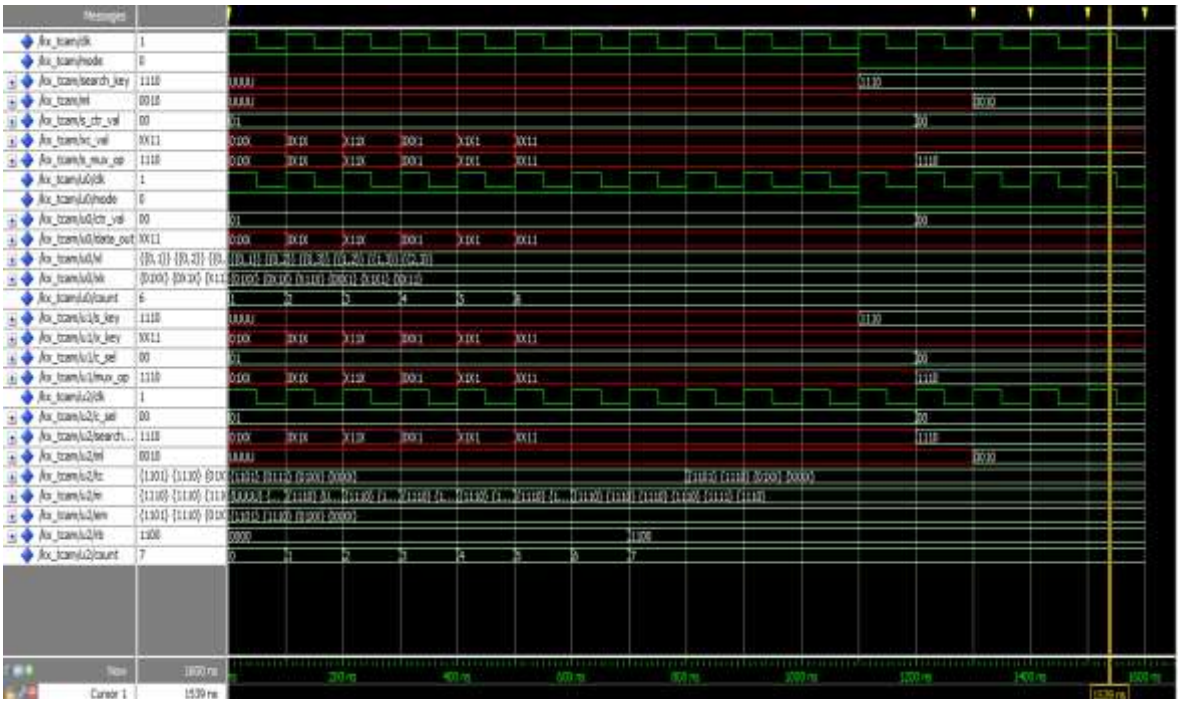

Fig.5 Simulation Result of Proposed Error Tolerant TCAM

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

The proposed simulation of parity TCAM with ATM is shown in Fig.5. During the normal mode search data is checked for its parity and is compared with data matching with its parity bits. In the testing mode X-keys are generated and if error detected then it is refreshed using ECC-SRAM memory.

Table.1. Comparison Results of Error Tolerant TCAM

Parameters	Area(Gate Count)	Power(mW)	Delay(ns)
TCAM	1556	235.90	9.285
Parity TCAM	1576	215.70	9.121
ATM with TCAM	84084	280.14	9.285
ATM with Parity TCAM	84104	262.69	9.121

VI. CONCLUSION

A novel soft-error tolerant Parity TCAM for multiple-bit-flip errors using partial don't-care keys (Xkeys) is proposed. The proposed TCAM corrects up to k-bit flip errors and significantly enhances the tolerance of the TCAM against soft errors, where k is the maximum number of bit flips in a word of a TCAM. Parity bit helps in reducing the search time along with power consumption. TCAM consists of a TCAM, a preprocessed don't-care bit index look-up memory (X look-up), and a backup ECC-SRAM. First, KX-TCAM randomly selects a search key. After that, TCAM detects multiple-bit-flip errors using the generated X-keys by X lookup. When the keys match the different locations, a soft error is detected and KX-TCAM refreshes the TCAM words by using a backup ECC-SRAM. Experimental results show that soft error tolerance of TCAM outperforms other schemes. The proposed design results in 8% power reduction with small overhead due to parity bits.

REFERENCES

[1]. K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE JSSC, vol. 41, No. 3, pp. 712-727, March 2006.
[2]. D. E. Taylor, "Survey and taxonomy of packet classification techniques," ACM Computing Surveys, vol. 37, no. 3, pp. 238-275, Sept. 2005.
[3]. C. Slayman, "Soft errors: Past history and recent discoveries," International Integrated Reliability Workshop Final Report, pp. 25-30, Oct. 2010.
[4]. A. Dixit and A. Wood, "The impact of new technology on soft error rates," IEEE International Reliability Physics Symposium, pp. 5B.4.1-5B.4.7, April 2011.
[5]. E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo and T. Toba, "Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule," IEEE Transactions on Electron Devices, vol. 57, no. 7, pp. 1527-1538, July 2010.
[6]. M. Maniatakos, M. K. Michael, and Y. Makris, "Multiple-bit upset protection in microprocessor memory arrays using vulnerability-based parity optimization and interleaving," IEEE TVLSI, vol. 23, no. 11, pp. 2447-2460, Nov. 2015.
[7]. I.Syafalni, T.Sasao, X.Wen, S.Holst, & K. Miyase, "A soft-error tolerant TCAM using partial don't-care keys," IEEE ETS, pp. 1-2, May 2015.
[8]. I. Syafalni, T. Sasao, and X. Wen, "Multiple-bit-flip detection scheme for a soft-error resilient TCAM," IEEE ISVLSI, pp. 679-684, July 2016.
[9]. I. Syafalni, T. Sasao, and X. Wen, "A soft-error tolerant TCAM for multiple-bit flips using partial don't care keys," IWLS-2015, Mountain View, CA, pp. 11-18, June 2015.
[10]. 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.
[11]. A. T. Do, S. S. Chen, Z. H. Kong, and K. S. Yeo, "A low-power CAM with efficient power and delay trade-off," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2011, pp. 2573–2576.