ENHANCED TCAM FOR IMPROVED NETWORK APPLICATION PERFORMANCE AND ERROR HANDLING

[#1]KAMA KANISHKA, ***Assistant. Professor,***

[#2]KOKKULA SHAILAJA, ***Assistant. Professor,***

Department of Electronics Communication Engineering,

SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR, TS.

Abstract: Content Addressable Memory (CAM) is a type of solid-state memory that prioritizes data content above storage location for data retrieval. TCAMs are employed as specialized memory in high-speed networking equipment such as routers, firewalls, and NATs. Soft-error tolerant TCAMs are required in high-reliability network applications such as military and defense systems to prevent data corruption or errors caused by radiation. To speed up the search, partial keys and parity-based logic are utilized, and a new TCAM is used because it can handle several bit-flip faults. The suggested TCAM makes the TCAM more resilient and corrects various bit-flip problems. TCAM uses the generated X-keys and an X search to detect multiple bit flip issues. If the keys match in more than one spot, a soft error is assumed, and the backup ECC-SRAM is utilized to update the TCAM words. The suggested CAM employs both the genuine data bits and the additional one-bit parity segment. We only get the parity bit to determine how many "1s" are even or odd. Driving intensity is increased by one parity bit for a single mismatch, while sensing delay is cut in half. Because only one TCAM is required, the proposed TCAM has a compact hardware footprint. The ease of implementation of parity-based TCAM aids in the creation of fault-tolerant packet filters.

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

1. INTRODUCTION

A CAM is a type of memory that is utilized by applications that need to iterate quickly across a list, database, or pattern. CAM is utilized in systems that process pictures or voices, such as computer and communication systems. When compared to other memory search methods, CAMs outperform them. At the same time, the saved items list is compared to the desired information. Data must be filtered, translated (encode-decoded), routed, and sent to numerous destinations in real time, including desktops, laptops, cellphones, and smart peripheral devices. Ternary Content Addressable Memory (TCAM) is an important component of modern network devices that stores and transmits data.

Careless errors are typically caused by ionizing particles such as protons, heavy ions, neutrons, and alpha particles. Radioactive atoms can decay into these particles on occasion. The impacts of miniaturization have been extensively researched. According to one recent study, switching from a 130 nm technology to a 22 nm process can raise soft error rates by a factor of seven. Soft faults may generate more serious problems with low-power systems. Soft faults on sequential and combinational circuits have been studied by researchers. In, a method for determining Soft Error Rates (SER) of combinational circuits was suggested that takes into account the effect of increasing the size of CMOS devices and the depth of processor pipelines. Traditional techniques to software and hardware development, on the other hand, have numerous flaws. Because hardware solutions necessitate the modification of TCAM circuits, they are prohibitively expensive to deploy. Researchers are also looking on new software engineering ways to address soft-error in TCAMs. A TCAM with a longer scrubbing time is useful for dealing with minor faults. It makes no difference whether some TCAM keys are used more frequently than others with this method. If the frequently used keys mistakenly strike the wrong word as a result of a little error, then utilizing this method may result in repeated mistakes. To accommodate for frequently hit keys, the suggested TCAM checker evaluates the matched words that arise from a soft error. However, because the system employs two TCAMs, additional hardware and energy are required.

Section II defines and characterizes CAM, Section III discusses Performance Improved TCAM, Section IV examines the suggested Error Tolerant TCAM, Section V illustrates the review's conclusions, and Section VI closes the work.

2. CAM AND ITS PROPERTIES

Binary and ternary CAM structures are the most prevalent. A binary CAM may store and locate bits of information that are either zero or one (0,1). A ternary CAM can hold a bit in the 0/1 or X states. Ternary CAMs are common on the Internet since longest-prefix routing is the current standard. Figure 1 depicts a simplified block design of a ternary CAM with NOR-based architecture. The CAM contains the route table from Table 1 to demonstrate the address lookup method. The CAM core cells are made up of four words, each of which contains five bits. Core cells have both memory and comparison circuits. The ascending and descending search lines in the image send queries to CAM cells. If the array's search results match the query in the row, the lines will be straight. In CAM instructions, a non-match is referred to as such, whereas a lit up matchline denotes a successful pairing. After receiving the matchlines, an encoder generates a position address for the match.

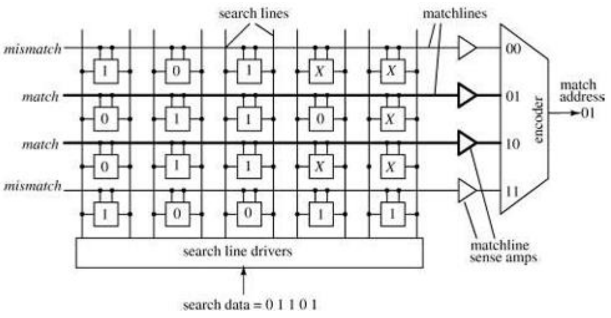

Fig.1. a 4x5 TCAM bit pattern simplified

Figure 2(a) depicts a core cell from a traditional SRAM. To store information, it uses a back-to-back circuit design with positive feedback. With the help of two access transistors, the word line (wl) manages the storage nodes connected to the bit lines (bl and /bl). Using bit lines, information in a cell may be read and written to. This specialized cell is the most fundamental component of the CAM cell. A typical binary CAM (BCAM) cell is depicted in Figure 2(b), which consists of two sets of search lines (sl and /sl) and a single set of match lines (ml). T, the truth value derived from d and d, is also visible in the image and is saved in the cell. For the sake of simplicity, the read and write access circuitry are not shown in the images of the CAM core cell below. When storing a binary CAM, we only need to change the storage of one bit. The storage cell's comparison circuits use an XNOR operation (ml) to compare the information on the search lines (sl and /sl) with the information in the binary cell.

That's shorthand for "(sl AND d)". When cells are mismatched, one of the transistor pairs in series becomes a ground path from the matchline. When d and sl are paired, the ground is separated from the match line.

Tertiary CAM (TCAM) cells are depicted in Figure 2(c). The TCAM cell stores the binary CAM as well as the X state, which requires two bits of storage on its own. If there is a don't care in a search result cell, that cell is a match.

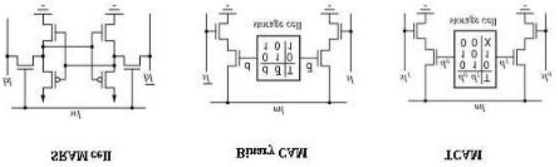

Fig.2. (a) SRAM Cell (b) Binary-CAM (c) TCAM

The figure shows that when d0 = d1 = 0, the TCAM cell stores X. Because it is never used, the meaning of the state d0=d1=1 is unknown. In computer networking equipment, memory that can be addressed based on its contents is often employed. When a data frame arrives at a network switch from one of its ports, the switch saves the source MAC address as well as other port information in an internal table. The destination MAC address is looked up in a table to identify which port should receive the frame. The frame is then sent across that interface. To reduce transition latency and speed up the process of locating the required target port, the MAC address table is often constructed using a binary CAM.

Ternary CAMs are commonly used in network routers. Every IP address has two parts: the host address, which uses the remaining bits, and the network address, the size of which is defined by the underlying subnet's setup. The network mask for a specific subnet is used to identify both the host and network components of an IP address. The router consults its own internal routing table when selecting where to send data packets. This routing table includes the destination network address, network mask, and path to it. The router verifies the provided address in the absence of a CAM.

First, the network mask and network address are combined using an AND operator. The outgoing message is then compared to each entry in the routing table. If they match, the packet is routed using the routing information supplied. Looking for a destination is a quick and easy task when the route table is organized as a ternary CAM. Because the CAM hardware handles masking and comparison, all you need to do is store addresses with "don't care" for the host part of the address and then look up the target address to recover the correct routing record.

3. PERFORMANCE ENHANCED TCAM

Fig. The parity-bit CAM structure is displayed. There are three bits in total, two of which are the data segment and one of which is a segment built from the data bits. We only get the parity bit to determine how many "1s" are even or odd. The parity bit is quickly added to the word. The new CAM has the same basic structure as the previous one, with one bit added. The search technique, like classical CAM, consists of only one stage.

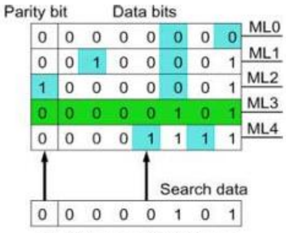

Fig.3. Parity based CAM Memory

Fig.3. Memorization based on parity

The inclusion of a parity bit and the shortening of the sensing time enhance the driving strength by 50% in the worst-case scenario of a 1-mismatch. Because the search word and the encoded word share the same parity bits, when a match in a fragment of data is found, the complete word also yields a match. When there is a 1 difference in the data segments, there must also be a 1 difference in the number of "1"s between the saved and search phrases. This causes a change in the corresponding parity numbers. As a result, the data and parity bits are currently misaligned. There will be two mismatches if there are two inconsistencies in the data part since the parity bits will be identical. Even if further variations are discovered, they are not required to be considered. It is enough for the sense enhancer to tell the difference between the matched and 2-mismatch instances. A 2-mismatch word has double the impact of a 1-mismatch word. This implies that the recommended arrangement considerably reduces search time.

4. THE PROPOSED ERROR TOLERANT TCAM

Fig.4. demonstrates the Parity-TCAM approach to TCAM soft error detection and repair, which can locate and correct defects such as multiple bit flips. 2) For parity-TCAM, an additional cell is required at each location. Third, the proposed strategy requires the use of a single TCAM. 4) TCAM beats competing techniques in terms of soft error tolerance.

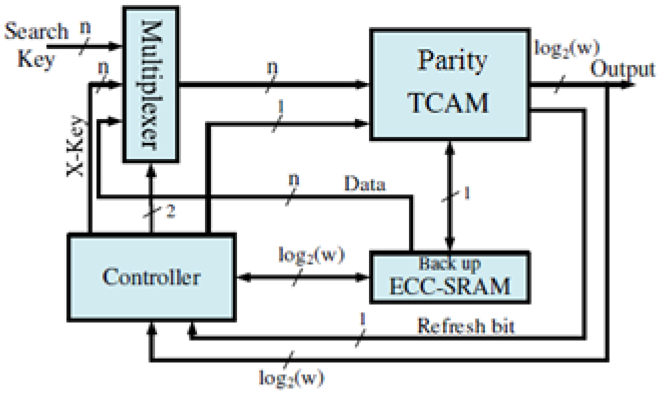

Fig.4. It is suggested that a TCAM that can tolerate errors be used.

There are two modes available: regular and testing. The suggested design also includes a multiplexer, controller, and redundant ECC-SRAM. The TCAM does a parallel search using a specified search key in its default mode. The X-TCAM creates partially careless keys (Xkeys) and checks for soft faults in test mode. The method begins with a sequence of X-keys being pressed on the TCAM. "Don't care" bits are added to the X-keys in the X look-up memory to make them match words with soft errors. When the X-keys on the TCAM are pressed, Parity-TCAM remembers all returned indices. If the two sums differ, this is referred to as a rounding mistake. If this is the case, there is no opportunity for a typographical error.

ATM protocol

CAMs can be used as a translation table by Asynchronous Transfer Mode (ATM) mobile network components. Because ATM networks are connection-based, virtual circuits must be formed across them before any data can be transferred. ATM runs on the ISO/OSI reference model's network layer, data link layer, and physical layer. Asynchronous Transfer Mode (ATM) is largely used in public switched telephone networks (PSTNs) and Integrated Services Digital Networks (ISDNs), however IP is increasingly replacing ATM in most cases. ATM uses asynchronous time-division multiplexing, with data encoded into cells, which are discrete packets of a size set by the International Organization for Standardization (ISO/OSI). ATM networks share several features with both circuit and packet switching networks. Different techniques are used by the Internet Protocol and Ethernet, which use bits and frames of varied sizes. Because ATMs use a connection-oriented approach, a virtual link between two terminals must be established before any information can be exchanged. These virtual circuits can be "permanent," meaning they are always-on, dedicated links pre-created by the service provider, or "switched," meaning they are established for each call by signaling and canceled when the call is complete. Finally, IP-only technology surpassed ATMs as the industry standard. Wi-Fi and smart ATMs have never taken off.

ATM virtual lines are identified by their respective virtual path IDs (VPIs) or channel path identifiers (VCIs). Each connection endpoint has a one-to-one relationship between the VPI and VCI. At each switch hop, the VPI and VCI of an ATM cell must be changed so that it can be used with the proper connection segment. This is referred to as VPI/VCI versioning. Because velocity is an important component of the ATM network, the rate at which this exchange occurs is critical.

the overall efficiency of the network. In an ATM switch, CAM can function as an address converter and rapidly translate VPI/VCI. Because typical CAM architectures lack the requisite capacity, creating multi-megabit translation tables that can search in parallel is only possible with a CAM/RAM combination. CAM translates the VPI/VCI numbers received in ATM cell headers into addresses for accessing data in external RAM during the translation process. It is compared to the VPI/VCI information in the ATM cell header as well as a running total of all active connections in the CAM array. Following the comparison, CAM will provide an address that may be used to access some external RAM containing data on links as well as the mapping between VPI and VCI. The cell is transferred to the switch after the ATM processor combines the VPI/VCI information from RAM into the cell header.

Structure of an ATM cell

Figure 1 depicts this. A typical ATM cell has 48 bytes of data followed by a 5-byte preamble. ATM distinguishes between two types of cells: Network-Network Interface (NNI) and User-Network Interface (UNI). ATM lines will frequently use the UNI cell format. The general flow control (GFC) field is a four-bit field designed to aid interoperability between ATM networks and shared access networks such as a distributed queue dual bus (DQDB) ring. The GFC field is made up of four bits that are used by the UNI to manage multiplexing and flow control across the cells of multiple ATM connections. Because no one has stated what the GFC field's purpose or values should be, they are always set to 0000.

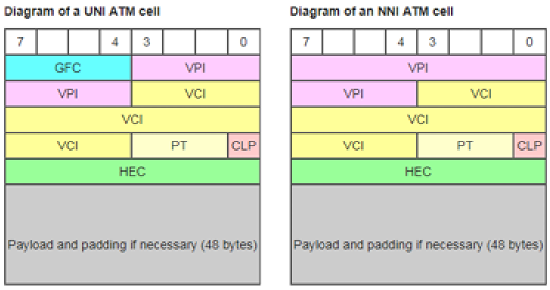
Diagram of a UNI ATM cell
7
4
3
0
GFC
VPI
VPI
VCI
VCI
VCI
PT
CLP
HEC
Payload and padding if necessary (48 bytes)
Diagram of an NNI ATM cell
7
4
3
0
VPI
VPI
VCI
VCI
VCI
PT
CLP
HEC
Payload and padding if necessary (48 bytes)

Fig.4. Tracing an ATM cell's "digital footprint" (8 bits UNI or 12 bits NNI)

The three-bit value PT, which is coupled with the virtual channel identification VCI, identifies the kind of payload.

The third bit (msbit) in a network indicates control cells. If the value in the user data cell is 0, the matching statement is accurate.

A 1 in PT bit 2 indicates network congestion, which stands for explicit forward congestion indication (EFCI). The first AAU bit (lsbit) of an ATM is its PT code. This is how packets are designated at their edges in AAL5. It stands for Cell Loss Priority (1-bit).

HEC is an abbreviation for "header error control," which is equivalent to "polynomial = X8 + X2 + X + 1, 8-bit CRC."

5. EVALUATION RESULTS

Table 1 shows the coverage, power, and delay of many TCAM implementations, both existent and proposed. In terms of power and efficiency, TCAM with higher parity exceeds alternatives. When applied in automated teller machines, the design greatly reduces power usage and takes up almost no additional space.

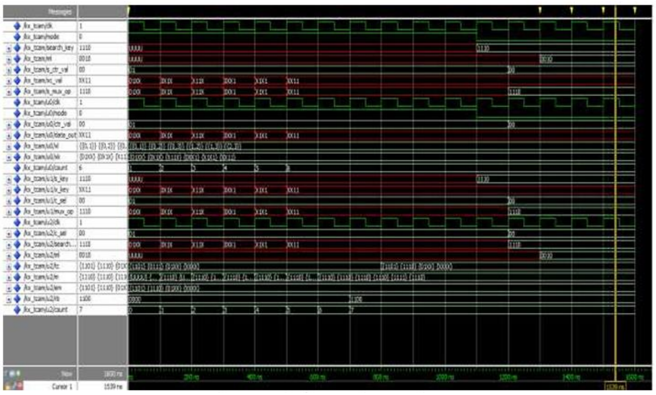

Fig.5 The proposed error-tolerant TCAM simulation results

Fig. Method 5 depicts the proposed method for mimicking parity TCAM utilizing ATM. In standard mode search, data parity is checked, and matching data with the same number of parity bits is compared. X-keys are generated and faults in the ECC-SRAM memory are repaired when in testing mode.

Table.1. TCAM Outcomes Error-Tolerant Comparison

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

6. CONCLUSION

A new Parity TCAM is proposed that can handle soft mistakes and numerous bit-flip errors, and it is predicted that this TCAM will use partial don't-care keys (Xkeys). The suggested TCAM excels at soft error management and can correct up to k-bit flip faults. In this case, k is the maximum number of bit flips allowed in a TCAM word. Parity bits reduce search time as well as energy consumption. TCAM has both primary and secondary ECC-SRAM, as well as a TCAM and an X look-up memory that has been pre-processed for "don't care" bits. KX-TCAM starts by picking a random search key. TCAM uses the X-keys created by X lookup to identify mistakes involving multiple bit flips. When the keys match in several spots, KX-TCAM detects a soft error and makes modifications to the TCAM words using fallback ECC-SRAM. The experimental results show that TCAM surpasses competing systems in terms of soft error tolerance. If the proposed solution were applied, there would be no parity-related expenditures, and energy consumption may be decreased by 8%.

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
4. 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. 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.