ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

Improved Design of Self Timed Low Power Asynchronous TCAM with Reassigned Parallel Search Mechanism for Higher Data Read and Write Operation

P.Sathish,Dr.C.GaneshBabu

ABSTRACT:

This paper introduces the idea that most mismatches can be found by searching few bits of the search word of content addressable memories. A word circuit was divided into two sections that are sequentially scanned or pipelined. The second section is idle in most of match lines. It is very fast to search last two bits compared to searching all the bits. In this circuit each word circuit is controlled by a locally generated timing signal rather than a global signal. A 128X64-bit CAM is designed in this project and simulation is done using Xilinx Tool. TCAM to be used to find a same mismatch that can be found by searching few bits. This TCAMis used in designing an IP Filter. The proposed asynchronous TCAM operates 5.98 times faster than a synchronous CAM with14.2% reduced power dissipation. The post-layout proposed CAMrecognizes 385-ps cycle delay time and 0.773 fJ/bit/search and is also calculated underreformedturning conditions.

Index Terms—Asynchronous circuits, associative memory, NAND-type CAM, pre-computation,potential match address (PMA), Ternary Content-Addressable Memory (TCAM).

I INTRODUCTION

Most memory devices store and recover data by addressing detailed memory locations. As a result, this path frequentlydevelops the restrictive factor for systems that rely on wild memory accesses.

Manuscript received April, 2016.
Sathish.p, ME VLSI desisgn, Bannari Amman Institute of Technology, Erode, India.
Dr.C.Ganeshbabu, Professor, Department of Electronics and Communication Engineering ,Anna University/ Bannari Amman Institute of Technology Erode, India.

The time compulsory to find an item stored in memory can be reduced meaningfully if the item can be recognized for access by its content rather than by its address. A memory that is retrieved in this technique is called content-addressable memory or CAM. It delivers a presentationbenefit over other memory exploration algorithms, such as binary or tree-based hunts or look-aside tag buffers, by associating the desired information in contradiction of the entire list of pre-stored entries simultaneously, often subsequent in an order-of-magnitude reduction in the search time. Typically a CAM is defined as a functional memory with a huge amount of stored data that simultaneously compares the input search data with the stored data. Once same data are found, their addresses are repaid as output as shown in Figure 1.1.

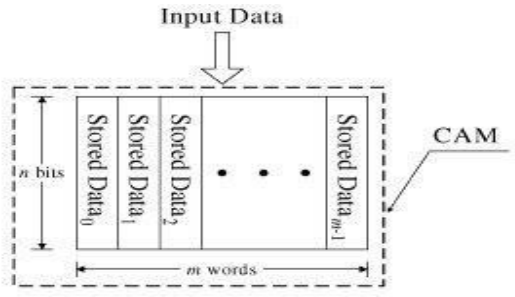

Figure 1.1 Conventional CAM

Differenttypical computer memory (random access memory or RAM) in which the user supplies a memory address and the RAM proceeds the data word stored at that address, a CAM is designed such that the user supplies a data word and the CAM explorations its entire memory to see if that data word is stored wherever in it.If the data word is originate, the CAM takings a list of one or more storage addresses where the word was initiate (and in some architecture, it also returns the data word, or other related pieces of

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

data). Thus, a CAM is the hardware byword of what in software terms would be called an associative array.

II SELF TIMED CAM

Content Addressable Memory (CAM) is a superior type of computer memory used in certain very-high-quicknesspointed applications. It is also identified as associative memory, associative storage, or associative array, while the last term is more frequently used for a programming data structure. It relates input search data (tag) against a table of stored data, and takings the address of equivalent data (or in the case of associative memory, the matching data). Several tradition computers, like the Goodyear STARAN, were built to implement CAM, and were designated associative computers. Each CAM cell has its individual comparison circuitry along with storage size. CAMs can be separated into two types built on its bit storage capacity: Binary CAM (BiCAM) and Ternary CAM (TCAM).

BiCAM has the ability to store two logical values (0 and 1) whereas TCAM can store 3 states (0, 1, and X; with X is a don't care state). Thus, in TCAM, a deposited key of 100X will equal any of the two search keys 1000 and 1001. TCAM contains of cells, which are organized in the form of a two dimensional array. Cells in the similar row (TCAM word) share a match line (ML) and cells in the similar column segment search lines (SLs).

A conventional TCAM array is shown in Figure4.1, which shows a 4x3TCAMarray containing of 4 rows (TCAM words) and 3 columns. Each TCAM word has its consistent ML and cells in the same column are connected to its consistent SLs. For a search operation, all the MLs must be pre-charged to $V_{DD}$and SLs must be discharged to Gnd in a conservativeTCAM before the search key is to be applied. Then, the search key is practical to all SLs in parallel. If a match occurs, ML breaks at its pre-charged value; then it discharges to Gnd through the contrast circuitry of the TCAM cell. An ML stays at V_{DD}only, if all the cells sharing the same *ML* have a match condition. Otherwise, a mismatch in any cell distribution the same ML will discharge that ML to Gnd. MLs are served to a priority encoder (PE).

Traditional TCAM model includes search lines, match lines, memory cells, and priority encoder. In BiCAM, a single match can arise, so a simple encoder is used. But in TCAM, an input word may be matched at several memory locations. Thus, a priority encoder (PE) is used to choice a memory location having highest importance; conventionally, lower physical address has the main priority.

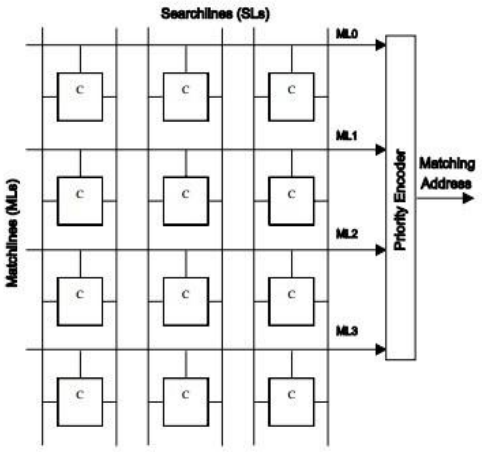

Figure:2.1 Traditional TCAM

This address is then used to raise an entry from the related SRAM. Figure 4.2 shows an example of search operation in standard TCAM. An input word 0111 is applied to TCAM. Two memory locations match: 1 and 2; with PE selecting the upper entry and producing the match location 1, which is then used to index its consistent entry from the associated RAM.

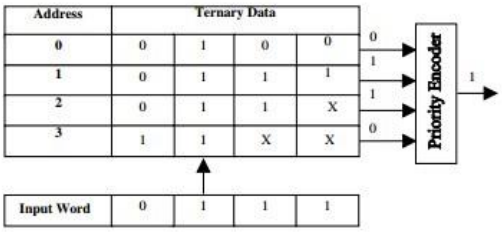

Figure:2.2 Search Operation In Classical TCAM

A.SINGLE CLOCK CYCLE:

The CAM searches done the memory in one clock cycle and revenues the Address where the data is found. The CAM can be preloaded at device start-up and also be redrafted during device operation. Since the CAM does not necessity address lines to find data, the complexity of a memory system using CAM can be extended as far as wanted, but the width is limited by the physical size of the memory.

CAM can be used to quicken any application demanding fast searches of database, lists, or designs, such as in image or voice

recognition, or computer and Communication projects. For this reason, CAM is used in applications where hunt time is very critical and must be very short.

B.OPERATION OF CAM:

It starts a liken operation by loading an n bit input hunt word into the hunt data register. The search data are then broadcast into the memory banks over n pairs of balancing search-lines SLs and directly associated with every bit of the stored words using comparison circuits. The search-data word is overloaded into the search-data register. All match-lines are pre-charged to high (impermanent match state).Search line drivers broadcast the search word onto the variance search lines. Each CAM core associates its stored bit against the bit on the matching search-lines. Match words that have at least one lost bit, discharge to ground. Every stored word has a ML that is shared between its bits to take the comparison result. Location of the coordinated word will be identified by an output encoder, as shown in Fig. 4(a).During a pre-charge stage, the MLs are seized at ground voltage level though both SL and ~SL are at VDD.

During assessment stage, complementary search data is transmission to the SLs and~SLS. When disparity occurs in any CAM cell (for example at the first cell of the row D="1"; ~D= "0"; SL= "1"; ~SL= "0"), transistor P3 and P4 will be twisted on, charging up the ML to a greater voltage level. A sense amplifier (MLSA) is used to sense the voltage change on the ML and amplifies it to a complete CMOS voltage output.

If incongruity happens to none of the cells on a row, no charge up path will be designed and the voltage on the ML will continue unchanged, representing a match. Since all accessible words in the CAMs are associated in parallel, result can be gotten in a single clock cycle. Hence, CAMs are quicker than other hardware- and software-based hunt systems.

C.PARITY BIT BASED CAM:

The parity bit based CAM design is exposedin Fig:2.3containing of the original data segment and an extra one-bit segment, resulting from the real data bits. The parity bit, attained is odd or even number of "1"s. The obtained parity bit is located directly to the consistent word and ML3. Thus the new architecture has the matching interface as the conservativeCAM with one extra bit. During the search operation, there is only one single stage as in conservativeCAM. Hence, the use of this parity bits does not recover the power performance.

However, this extra parity bit, in theory, reduces the detecting delay and boosts the driving strength of the 1-disparity case (which is the worst case) by half, as discoursed below .In the case of a coordinated in the data segment (e.g., ML3), the parity bits of the hunt and the stored word is the similar, thus the overall word returns a match. When 1 mismatch happens in the data segment (e.g., ML2), numbers of "1"s in the stored and search word must be dissimilar by 1. As a result, the consistent 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 disparities in the data segment (e.g., ML0, ML1or ML4), the parity bits are the same and overall we have two disparities. With more disparities, we can ignore these cases as they are not critical cases.

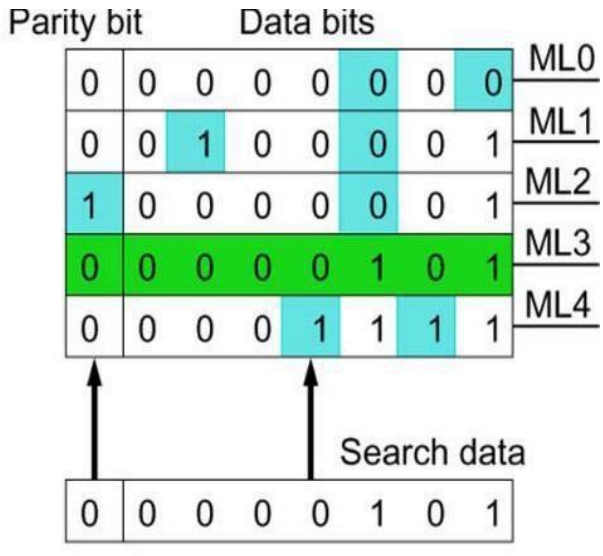

Figure:2.3 Parity-Bit Based CAM

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

III TCAM

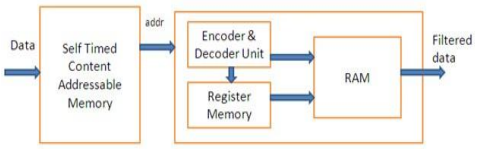

Figure:3.1 TCAM Block diagram

As stated above, route lookup is the major bottleneck in Internet routers. TCAM provides a very Good-looking solution. Several investigators have explored the pros and cons of a TCAM-based route lookup method in the past few years. These comprised both single and parallel TCAM architectures where numerous issues were addressed high memory competence, Cost-effective power dissipation and balanced load among similar TCAMs. Liu et al used both clipping and logic minimizing algorithms to reduce the advancing table sizes, which in tum abridged the memory cost and power consumption of TCAM. Many other studies developed more power-efficient lookup engines helping from a new feature of some TCAMs called "partition-disable". The key idea is to separate the whole routing table into numerous sub-tables or buckets, where all bucket is laid out over single or more TCAM blocks. During a lookup operation, only the block(s) covering the prefaces that match the incoming IP address is (are) triggered instead of all the entries in the original table. In this fashion, TCAM's power consumption can be melodramaticallyreduced. As a key component of such design, generally three kinds of algorithms for dividing the entire routing table were proposed, Le., key-ID based, prefix tree-based and range-based partitioning. The basic idea of key-ID approach is to select some particular bits among the prefixes, and group the prefixes accordingly.

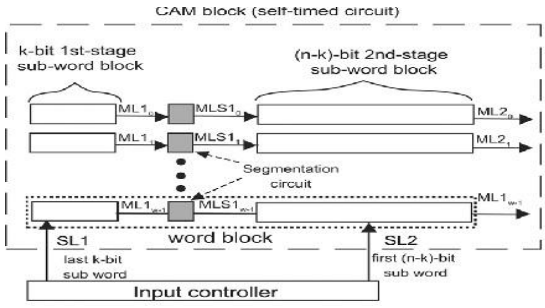

Figure:3.2 TCAM Architecture

Memory Filter

Illustrates the structure of aninnovative lookup engine.It is mostlycollected of an index created by "Max-splitting", a traditional TCAM-based lookup engine and a memory logical function unit joint Lookup engine named Memory Filter. Given an incoming IP packet to be hunted, the IP address is extracted and brought to the index. The index will return a sub-tree number representing the possible competitions. If this sub-tree covers more than n prefixes, we store and lookup this sub-tree in the TCAM. Then, we store and lookup this sub-tree in the memory Filter. Since the Memory Filter lone conducts very simple comparison (IP XOR Prefix =? = Mask) and priority decoding in small scale, it still equals the speed of TCAM when n is small. For IPv4, each prefix needs about 42 bits storage (32-bit IP, 5-bit Mask, and 5-bit next-hop interface). By presenting a 128-bit DDRAM module to work at the equivalent frequency as TCAM, n can easily reach 6. With an even quickerDDR3 memory, higher n can be achieved.

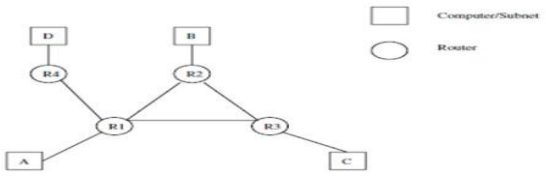

Figure: 3.3 IP Routing network model

Internet Routers have develop a major determining factor in the presentation of the Internet backbone. Routing networks have developedprogressively fast with Gigabit Ethernet (1.0 GB/s), OC-48 (2.4 GB/s), and OC-192 (9.6Gb/s) standards flattering prevalent in current designs. Optic communication standards with speeds up to 40 GB/s are around the angle. For IP networks working at these speeds, wild routing lookup is a critical condition. When an IP packet is being routed from its source to its endpoint, we perform a routing lookup at each router that is encountered in the path of the packet. The routing lookup process at each router includes performing a comparison of the incoming packet endpoint address against each entry in the routing (forwarding) table related with the router. The forwarding table entry for the received packet records the destination address or port for the packet. The packet is forwarded to this port (also mentioned to as its next hop address) on the way to its last destination. The process of IP routing is illustrated by the simplified IP routing network

exposed in Figure 1. Suppose Atransmits a data packet envisioned for B. Typically, rather than burden Ahas prior knowledge of the pathto use in order to reach B, the network calculates a route from A to B in a localized fashion. A simply transmits the data to router R1, permitting R1 to decide how to route the data to B. R1 may now onward the data to R2, while R2 would forward the data to B. Note that in this figure, all router has several incident edges, which are mentioned to as interfaces. So the task of all router is to forward an incoming packet on interface i to an interface j such that the packet makes "progress" in reaching its destination. A router attains this by means of a routing table, which records the outgoing interface for all incoming packet. Of course, since it would be prohibitively luxurious to record the outgoing interface for each destination address, routers store entries in a compact method. This is attained by storing destination addresses as subnet prefixes (henceforth mentioned to as prefixes) along with a consistent subnet mask (henceforth referred to as a mask). A mask records the bits of the address that essential to be considered while performing the lookup. Table5.1 is a minor IP routing table which illustrates this idea. In this table, each entry contains a subnet prefix, a subnet mask, and the next hop address.

IV TCAM Implementation

Our TCAMdesign consists of 168 TCAMblocks, applied in a 13×13 grid as described in Figure 3. Each block contains of 128 entries, implemented1 With 128 entries per block, the layout of all block is approximately square, resulting in the lowest circuit delay.

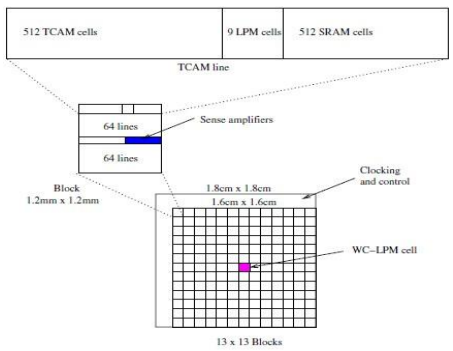

Figure: 4.1 Floor Plan and Cell Arrangement Of Our TCAM

TCAM Cell Design:

Each of the 512 TCAM cells consists of two memory cells which store the prefix and the mask bits. The applied input is likened to the entry bit stored in the TCAM cell. If the mask bit has a stored value of '1', the TCAM cell will perform the comparison between the practical input bit (for that specific cell) and the stored value. The top portion of the cell stores the prefix entry although the bottom portion stores mask data. The preface (mask) is written by asserting W1 (W0) while placing the data to be stored on B and B Bar. A match is performed by first pre-charging the MATCH line using the PCH signal. Next, the data being looked up is asserted on B Bar and B. If the data miss matches with contents of the cell and the stored mask value is '1', then MATCH is pulled low. In all other cases, MATCH stays precharged. The MATCH line is common among all 512 TCAM cells in a row. In other words, whenever the mask value stored in a cell is '0', the cell no lengthier affects the final value of the MATCH signal, as we require.

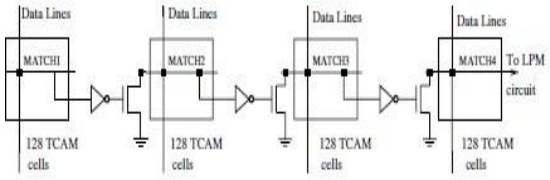

Figure: 4.2 TCAM Row broken into four sections

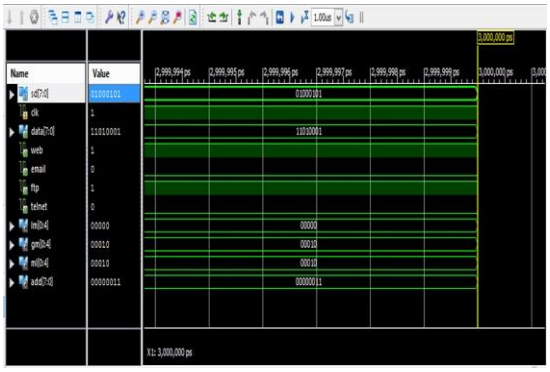

Figure: 4.3 TCAM IP

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

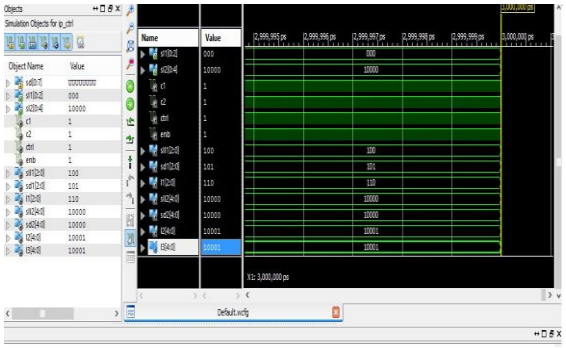

Figure: 4.4Input Controller

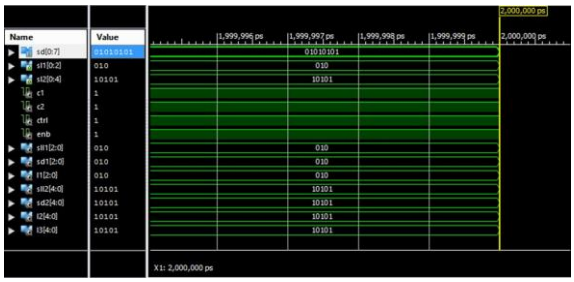

Figure: 4.5 TCAM

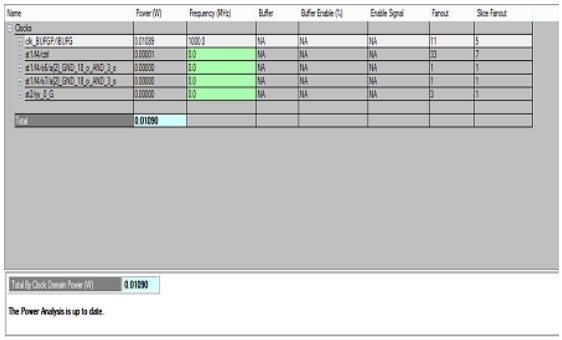

Figure: 4.6 Power analysis

V CONCLUSION

In this paper, a high-throughput low-energy content addressable memory (CAM) based on a rearranged overlapped search mechanism that includes two new approaches, a reordered word-overlapped search (RWOS) and phase-overlapped processing (POP).Using the RWOS scheme at the scheduling level, the proposed CAM operates at a rate based on the short delay of the last few-bit search rather than the long delay of whole-word search. At the circuit level, the POP scheme hides the delay of a pre-charge phase using local control and hence greatly reduces the cycle time compared with a traditional synchronous CAM. These methods are realized using asynchronous circuits, which operate properly under PVT variations. As a design example, a 128X64-bit CAM is designed and simulation using Xilinx Tool. The proposed asynchronous CAM achieves 385-ps cycle time and 0.773 fJ/bit/search that are 47.2% and 25.7% of a recently published high-speed CAM using a swapped CAM cell whose the number of transistors is 11 instead of 9 in a regular CAM cell used in the proposed CAM.

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. 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. Circuit Syst. (ISCAS)*, 2011, pp. 2573–2576.

[3] I. Arsovski and A. Sheikholeslami, "A mismatch-dependent power allocationtechnique for match-line sensing in content-addressable memories," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1958–1966, Nov. 2003.

[4] N. Mohan and M. Sachdev, "Low-leakage storage cells for ternary content addressable memories," *IEEE Trans. Very Large Scale Integr (VLSI) Syst.*, vol. 17, no. 5, pp. 604–612, May 2009.

[5] O.Tyshchenko and A. Sheikholeslami, "Match sensing using matchline stability in content addressable memorys (CAM)," *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 1972–1981, Sep. 2008.

[6] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "A low-power ternary CAM with positive-feedback match-line sense amplifiers,"*IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 3, pp. 566–573,Mar. 2009.