A Survey on Various Types of Content Address Memory

Ashit Samdur[1],Vaibhavdeep Sahare[2], Nilesh Mohota[3], Rahul Langde[4]
[1]PG Research Student, [2]PG Research Student, [3]Associate Professor, [4]Assistant Professor
[1234]J D College Of Engineering and Management, Nagpur, India.

Abstract: We overview late improvements in the design of extensive limit content-addressable memory (CAM). A CAM is a memory that executes the query table capacity in a solitary clock cycle utilizing committed examination circuitry. CAMs are particularly mainstream in organize switches for parcel sending and bundle characterization, however they are additionally helpful in an assortment of different applications that require rapid table query. The primary CAM-design challenge is to diminish power utilization related with the expansive measure of parallel dynamic circuitry, without yielding velocity or memory thickness. In this paper, we review CAM-design strategies at the circuit level and at the structural level. At the circuit level, we review low-power coordinate line sensing systems and hunt line driving methodologies. At the engineering level we review three strategies for decreasing power utilization.

Keyword: Content Addressable Memory(CAM), Matchline Pipelining, Matchline Sensing, NAND Cell, NOR Cell, Review, Searchline Power.

I. INTRODUCTION

CONTENT-ADDRESSABLE memory (CAM) looks at input seek information against a table of put away information, and returns the address of the coordinating information [1]– [5]. CAMs have a solitary clock cycle throughput making them quicker than other equipment and programming based inquiry frameworks. CAMs can be utilized as a part of a wide assortment of utilizations requiring high pursuit speeds. These applications incorporate parametric bend extraction, Hough change, Huffman coding/interpreting, Lempel–Ziv pressure, and picture coding. The essential business use of CAMs today is to arrange and forward Internet convention (IP) parcels in organize switches. In systems like the Internet, a message such an as email or a Web page is exchanged by first separating the message into little information parcels of a couple of hundred bytes, and, at that point, sending every datum bundle exclusively through the system. These parcels are directed from the source, through the middle of the road hubs of the system (called switches), and reassembled at the goal to duplicate the first message. The capacity of a switch is to look at the goal deliver of a parcel to every conceivable course, so as to pick the proper one. A CAM is a decent decision for actualizing this query activity because of its quick hunt capacity.

For switches that serve the internet center, the routing query task should be performed on information that lands on optical interfaces working at a couple of 100 GB/s. Thus, it is basic that the routing table queries for these switches be done in equipment, in parallel, as opposed to in programming. The circuit of decision for equipment switches is Ternary Content Addressable Memory (TCAM) [1], [2], [3]. TCAMs in a perfect world store a whole routing table, and play out a concurrent correlation for all routing sections against the goal address of the bundle being directed. A TCAM is a variation of a store (which is additionally alluded to as a CAM [4]), with the additional capacity to ignore a subset of address bits while per-framing the query. The address bits that are neglected amid a routing table query activity compare to the cover bits of the routing table section (which have 0 esteems).

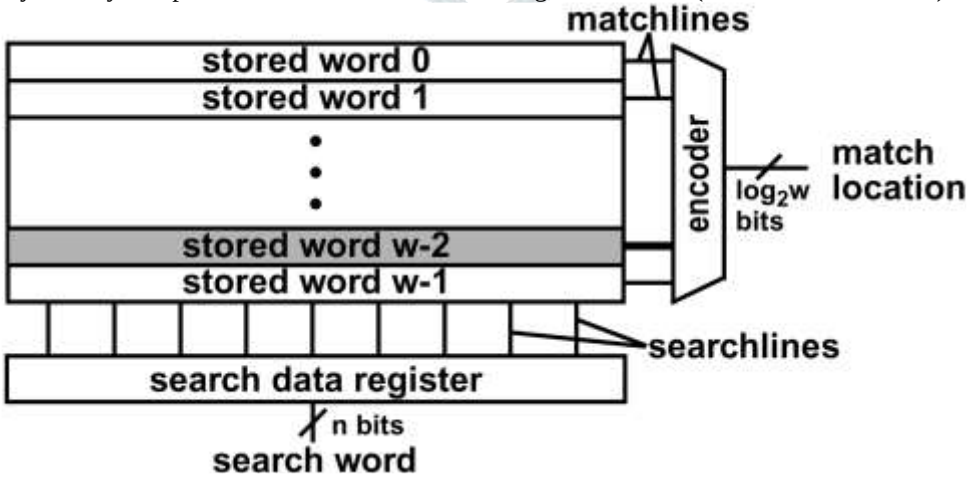

Fig. 1. Conceptual view of a content-addressable memory containing *w* words.

Generally, TCAMs are executed utilizing CMOS ICs, in which each TCAM cell requires 17 transistors, and each SRAM cell (which stores the interface or next bounce port) requires 6 transistors. In our paper, we understand each TCAM cell utilizing 2 flash transistors, and each "SRAM" cell (we allude to it as a port cell) requires 1 flash transistor. As a result, our flash-based TCAM (FTCAM) piece [5] is altogether more thick than the customary CMOS-based TCAM obstruct (by a factor of around 7.9X), with a query postpone which is 2.5X bigger, and a power utilization which is 1.64X lower than the CMOS-based TCAM.

In any case, the speed of a CAM comes at the cost of expanded silicon territory and power utilization, two design parameters that designers endeavour to lessen. As CAM applications develop, requesting bigger CAM sizes, the power issue is additionally exacerbated. Diminishing power utilization, without yielding velocity or region, is the fundamental string of late research in extensive limit CAMs. In this paper, we review improvements in the CAM zone at two levels: circuits and models. Before giving a blueprint of this paper toward the finish of this segment, we first quickly present the task of CAM and furthermore depict the CAM utilization of parcel sending. Fig. 1 demonstrates an improved piece outline of a CAM. The contribution to the framework is the hunt word that is communicated onto the search lines to the table of put away information. The quantity of bits in a CAM word is generally extensive, with existing executions running from 36 to 144 bits. A run of the mill CAM utilizes a table size going between a couples of hundred passages to 32K sections, relating to an address space extending from 7 bits to 15 bits. Each put away word has a match line that shows whether the pursuit word and put away word are indistinguishable (the match case) or are unique (a befuddle case, or miss). The match lines are nourished to an encoder that produces a paired match area comparing to the machine that is in the match state. An encoder is utilized as a part of frameworks where just a solitary match is normal. In CAM applications where in excess of single word may coordinate, a need encoder is utilized rather than a basic encoder. A need encoder chooses the most elevated need coordinating area to guide to the match result, with words in bring down address areas accepting higher need. Furthermore, there is regularly a hit flag (not appeared in the figure) that banners the case in which there is no coordinating area in the CAM. The general capacity of a CAM is to take a pursuit word and restore the coordinating memory area. One can think about this task as a completely programmable subjective mapping of the huge space of the info look word to the littler space of the yield coordinate area.

The activity of a CAM resembles that of the label part of a completely cooperative store. The label bit of a reserve looks at its information, which is an address, to all locations put away in the label memory. On account of match, a solitary match line goes high, demonstrating the area of a match. Not at all like CAMs, reserves don't utilize need encoders since just a solitary match happens; rather, the matchline specifically initiates a read of the information segment of the store related with the coordinating tag. Numerous circuits are regular to the two CAMs and stores; in any case, we center around large capacity CAMs as opposed to on completely cooperative reserves, which target littler limit and higher speed.

The present biggest financially accessible single-chip CAMs are 18 Mbit executions, in spite of the fact that the biggest CAMs announced in the writing are 9 Mbit in measure [6], [7]. As a general guideline, the biggest accessible CAM chip is as a rule about a large portion of the measure of the biggest accessible SRAM chip. This dependable guideline originates from the way that a run of the mill CAM cell comprises of two SRAM cells, as we will see in no time.

II. RELATED WORK

A. Literature Survey

A decent diagram of existing TCAM methodologies can be found in [1]. Most TCAM executions store routing sections in squares [3], [12], where each piece contains routing passages of a specific veil length. This takes into consideration quick queries, since the IP address would be turned upward in all pieces, and just the match from the square with the most noteworthy match length would be chosen. Another usage [13] permits routing table passages to be put away at any areas in the TCAM, however require two cycles to play out a query. Routing table query in this approach is done in a non-pipelined, two phase way. In the principal stage, the TCAM plays out the query and plays out a bitwise OR of the coordinating sections' covers. This delivers the longest cover, which is nourished back to the TCAM and further compels the first coordinating passages to create the section with the longest prefix. The fundamental downside of this approach is that in bringing down the cost of inclusion, the cost of every query is multiplied.

Most TCAM designs use a need encoder [14] circuit to play out the Longest Prefix Match (LPM). The LPM calculation is generally done in equipment, either utilizing committed equipment [13], or by organizing the routing table passages in a particular request as portrayed in [12].

In [15], the creators examine systems for decreasing power in a TCAM, including 3D stacking and the utilization of programmable vias to spare zone in the port memory. In that capacity, the strategies portrayed are orthogonal to the thoughts we show in this paper. STT-based TCAM circuits were proposed [16], however because of the protection variety of the attractive intersections their

design uses three memory components for each cell, and additionally 3 + 11 CMOS gadgets, and though our FTCAM cell uses just two flash transistors. Our FTCAM utilizes 0.6 fJ/bit/seek while the TCAM in [16] requires 7.1 fJ/bit/look. The query vitality proportion generally tracks the quantity of gadgets per TCAM cell. Memristor-based CAM uses two memory components and three CMOS transistors [17]. Be that as it may, the focal point of their paper isn't on TCAMs, yet rather on the cell design of a memristor-based CAM. The creators of [18] exhibit a resistive TCAM cell, to be coordinated with the virtual memory, influencing the physical deliver to space content-addressable. When all is said in done, the focal point of this paper is at a more elevated amount of deliberation, not at all like our work. In [10], the creators executed a productive TCAM, in which routing table sections are put away in any request, consequently taking out the vast most pessimistic scenario addition cost of average TCAM usage, as portrayed in [12]. What's more, they utilized an efficient Wired-NOR based LPM circuit, whose defer scales logarithmically with n, in this manner enhancing over the direct multifaceted nature (in the extent of the TCAM) of need encoder based circuits. All the above methodologies use a CMOS usage of the TCAM.

There have been a few research endeavours which contemplate the flash gadgets and their utilization in memory. An abbreviated rundown incorporates. These papers report points of interest of flash gadgets and their portrayal. In any case, they don't portray the utilization of flash transistors for TCAM like circuits. To the best of our insight, there has been no earlier work that utilizations flash gadgets to acknowledge TCAM structures.

In our approach, we use flash transistors to understand the TCAM cells. We accept that the TCAM is acknowledged in hinders (with 256 passages for every square). Every section comprises of 256 TCAM bits (consequently supporting IPv6 routing tables), and 512 information bits. In spite of the fact that the focal point of our work is on the design of a TCAM piece, we likewise talk about the engineering of the whole TCAM, examining how routing updates and course folds would be dealt with. The flash-based TCAM piece is thought about as far as format region, deferral and power with a proficient CMOS TCAM design, which was re-actualized in the 45nm PTM innovation for a reasonable examination with our flash-based TCAM square.

B.Ternary CAMs

The NOR and NAND cells that have been displayed are parallel CAM cells. Such cells store either a rationale "0" or a rationale "1". Ternary cells, what's more, store a "X" esteem. The "X" esteem is a couldn't care less, that speaks to both "0" and "1", permitting a special case operation.Wildcard task implies that a "X" esteem put away in a cell causes a match paying little mind to the information bit. As talked about before, this is a component utilized as a part of bundle sending in Internet switches.

	Stored		Search Bit	
Stored Value	D	$\overline{D}$		
0	0	1	0	1
1	1	0	1	0
X	1	1	0	0

Table I Ternary Encoding For Nor Cell

A ternary image can be encoded into two bits as indicated by Table I. We speak to these two bits as D and D'. Note that in spite of the fact that the D and D' are not really correlative, we keep up the integral documentation for consistency with the twofold CAM cell. Since two bits can speak to 4 conceivable states, yet ternary stockpiling requires just three states, we forbid the state where D and D' are both zero. To store a ternary incentive in a NOR cell, we include a moment SRAM cell, as appeared in Fig. 2. One piece, D, interfaces with the left pull-down way and the other piece, associates with the privilege pull-down way, making the pull-down ways freely controlled.We store a "X" by setting bothD and equivalent to rationale "1", which debilitates both pull-down ways and powers the cell to coordinate in any case in the inputs.We store a rationale "1" by setting D=1 and D' = 0 store a rationale "0" by setting D=0 and D' =1. Notwithstanding putting away a "X", the cell permits scanning for a "X" by setting both SL and to rationale "0". This is an outer couldn't care less that powers a match of somewhat paying little heed to the put away piece. Despite the fact that putting away a "X" is conceivable just in ternary CAMs, an outer "X" image conceivable in both twofold and ternary CAMs. In situations where ternary task is required however just twofold CAMs are accessible, it is conceivable to imitate ternary activity utilizing two paired cells for each ternary image.

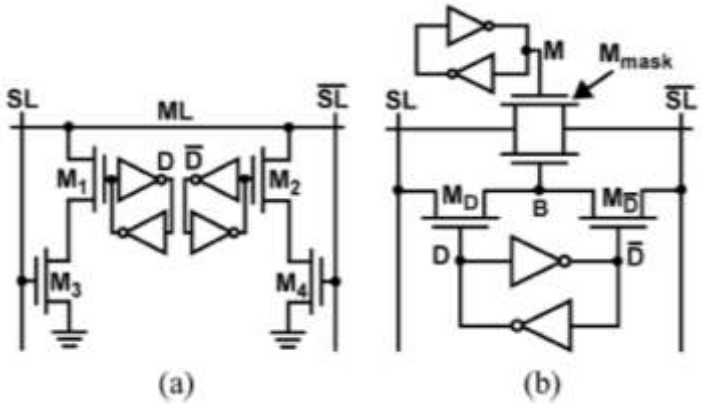

Figure 2 Ternary core cells for (a) NOR-type CAM and (b) NAND-type CAM

As an adjustment to the ternary NOR cell of Fig. 2, Roth et al. [7] propose executing the pull-down transistors utilizing pMOS gadgets and supplementing the rationale levels of the searchlines and match lines as needs be. Utilizing pMOS transistors (rather than nMOS transistors) for the examination circuitry takes into account a smaller design, because of decreasing the quantity of spacing's of p-dispersions to n-disseminations in the cell. Notwithstanding expanded thickness, the littler territory of the cell decreases wiring capacitance and in this manner diminishes power utilization. The tradeoff that outcomes from utilizing least size pMOS transistors, instead of least size nMOS transistors, is that the pull-down way will have a higher proportionate protection, backing off the pursuit activity.

Assist minor alterations to CAM cells incorporate blending parts of the NAND and NOR cells, utilizing dynamic-edge strategies in silicon-on-separator (SOI) forms, and substituting the rationale level of the pull-down way to ground in the NOR cell.

III. CONCLUSION

In this paper, we have overviewed CAM circuits and designs, with an accentuation on high-limit CAM. To begin with, we roused our talk by demonstrating how CAMs can be connected to bundle sending in arrange switches. At the circuit level, we have reviewed the two essential CMOS cells, to be specific the NOR cell and the NAND cell. We have additionally indicated how the cells are joined in a matchline structure to shape a CAM word. We have investigated the ordinary precharge-high plan for sensing the matchline, and also a few varieties that spare matchline power including low-swing sensing, the present race plot, particular precharge, pipelining, and current sparing plan.

REFERENCES

[1] J. McAuley, P. Francis, "Fast routing table lookup using CAMs", Proc. IEEE INFOCOM, pp. 1382-1391, Mar./Apr. 1993.
[2] T. B. Pei, C. Zukowski, "VLSI implementation of routing tables: Tries and CAMs", Proc. IEEE INFOCOM, vol. 2, pp. 515-524, 1991.
[3] J. Wade, C. Sodini, "A ternary content addressable search engine", IEEE J. Solid-State Circuits, vol. 24, no. 4, pp. 1003-1013, Aug. 1989.
[4] J. M. Rabaey, A. Chandrakasan, B. Nikolic, Digital Integrated Circuits, Englewood Cliffs, NJ, USA:Prentice-Hall, 2003..
[5] V. Fedorov, M. Abusultan, S. Khatri, "An area-efficient ternary cam design using floating gate transistors", Proc. 32nd IEEE Int. Conf. Comput. Des., pp. 55-60, Oct. 2014.
[6] G. Kasai,Y. Takarabe, K. Furumi, and M.Yoneda, "200 MHz/200 MSPS 3.2 W at 1.5 V Vdd, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme," in Proc. IEEE Custom Integrated Circuits Conf. (CICC), 2003, pp. 387–390.
[7] A. Roth, D. Foss, R. McKenzie, and D. Perry, "Advanced ternary CAM circuits on 0.13m logic process technology," in Proc. IEEE Custom Integrated Circuits Conf. (CICC), 2004, pp. 465–468.
[8] Gamache, Z. Pfeffer, S. Khatri, "A fast ternary CAM design for IP networking applications", Proc. IEEE Int. Conf. Comput. Commun. Netw., pp. 434-439, 2003.
[9] K. Pagiamtzis, 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.
[10] D. Shah, P. Gupta, "Fast updating algorithms for TCAMs", IEEE Micro, vol. 21, no. 1, pp. 36-47, Jan./Feb. 2001.
[11] M. Kobayashi, T. Murase, A. Kuriyama, "A longest prefix match search engine for multi-gigabit IP processing", Proc. IEEE Int. Conf. Commun., vol. 3, pp. 1360-1364, 2000.

[12] J. Wakerly, Digital Design Principles and Practices, Englewood Cliffs, NJ, USA:Prentice-Hall, 1990.
[13] M. Lin, J. Luo, Y. Ma, "A low-power monolithically stacked 3D-TCAM", Proc. IEEE Int. Symp. Circuits Syst., pp. 3318-3321, 2008.
[14] W. Xu, T. Zhang, Y. Chen, "Design of spin-torque transfer magnetoresistive RAM and CAM/TCAM with high sensing and search speed", IEEE Trans. Very Large Scale Integration Syst., vol. 18, no. 1, pp. 66-74, Jan. 2010.
[15] K. Eshraghian, K.-R. Cho, O. Kavehei, S.-K. Kang, D. Abbott, S.-M.S. Kang, "Memristor MOS content addressable memory (MCAM): Hybrid architecture for future high performance search engines", IEEE Trans. Very Large Scale Integration Syst., vol. 19, no. 8, pp. 1407-1417, Aug. 2011.
[16] Q. Guo, X. Guo, Y. Bai, E. Ipek, "A resistive TCAM accelerator for data-intensive computing", Proc. 44th Annu. IEEE/ACM Int. Symp. Microarchit., pp. 339-350, 2011.