ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Advance Architecture of T-CAM: Z-TCAM

Annaram Sowjanya
M.Tech VLSI,
Department of ECE,
CMR Institute of Technology.

B S Priyanka Kumari
Assistant Professor,
Department of ECE,
CMR Institute of Technology.

K.Narmada
Assistant Professor,
Department of ECE,
CMR Institute of Technology.

ABSTRACT:

Ternary content-addressable memory (TCAM) is often used in high speed search intensive applications such as ATM switch, IP filters. Hence, currently ZTCAM, is introduced which emulates the TCAM functionality with SRAM. It has some drawbacks such as low scalability, low storage density, slow access time and high cost. But this paper proposes a novel memory architecture of existing Z-TCAM using SRAM. Hence the area, delay and power are reduced by using SRAM. The detailed implementation results and power measurements for each design have been reported thoroughly.

General Terms:

Memory Search, Low power, VLSI.

Keywords:

Ternary Content Addressable Memory, Spin Transfer Torque RAM, Hybrid Partitioning.

1. INTRODUCTION:

CAM stands for Content Addressable Memory which is a special type of memory used by Cisco switches. In the case of ordinary RAM the IOS uses a memory address to get the data stored at this memory location, while with CAM the IOS does the inverse. It uses the data and the CAM returns the address where the data is stored. Also the CAM is considered to be faster than the RAM since the CAM searches the entire memory in one operation. CAM tables provide only two results: 0 (true) or 1 (false). [8] TCAM stands for Ternary Content Addressable Memory is the capability extension of CAM which can match a third state, which is any value. This makes TCAM a very important component of Cisco Layer 3 switches and modern routers, since they can store their routing table in the TCAMs, allowing for very fast lookups, which is considerably better than routing tables stored in ordinary RAM. TCAM is a specialized CAM designed for rapid table lookups [8].

TCAM A cell has two static random access memory (SRAM) cells and a comparison circuitry and provides three state: 0, 1, and x where x is a don't care state. The x state is always regarded as matched irrespective of the input bit. TCAM provides single clock lookup with constant search time which makes it suitable for applications such as network routers,data compression, real-time pattern matching in virus-detection, and image processing. In paper [1] TCAM is designed using SRAM which is called as Z-TCAM ,because even though the TCAM table provides lookup of entire table in single clock it has various disadvantages when compared to SRAM. TCAM cells, comparator's circuitry in add complexity to the TCAM architecture. The access time of TCAM, is 3.3 times longer than the SRAM access time due to the massive parallelism [5]. Complex integration of memory and logic also makes TCAM testing very time consuming [3]. The cost of TCAM is also about 30 times more per bit of storage than SRAM [6]. But, the parameters such as area, delay and power can be further reduced by using STT RAM instead of SRAM [1] which has been proved in this work.

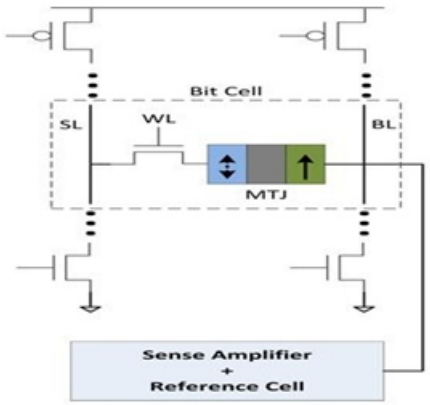


Fig 1: Circuit diagram of STT RAM

Spin transfer torque random access memory (STT-RAM) is an emerging memory technology that promises to deliver the benefits of current non-volatile memories (speed, high density) with the added benefit of being non-volatile and offering no leakage power from the storage element. The detailed architecture and its implementation were explained in the below sections. Critical implementation techniques in low power VLSI design have been discussed in [21], [22] and [23].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Table 1. Traditional TCAM table and its hybrid parameters:

ADDR	SW1	SW2	Layer
0	00	11	1
1	01	01	
2	0x	11	2
3	11	1X	

Table 2. Z-TCAM example: Data Mapping:

ADD	0	1	2	3
VM11	1	1	0	0
VM12	0	1	0	1
VM21	1	1	0	1
VM22	0	0	1	1
OATAM1	0	0	-	-
OATAM1	-	0	-	1
OATAM2	0	1	-	2
OATAM2	-	-	0	1

Table 3. Z-TCAM Original Address Mapping

ADDR	OAT11	OAT12	OAT21	OAT22
	01	01	23	23
0	10	01	10	01
1	01	10	10	11
2	00	00	01	00
3	00	00	00	00

2. HYBRID PARTITIONING OF TCAM:

Hybrid partitioning (HP) is the combination of vertical partitioning and horizontal partitioning of the conventional TCAM table. An example of HP is given in Table 1[1]. HP partitions conventional TCAM table vertically (column wise) and horizontally (row wise) into TCAM sub tables, which are then processed to be stored in their corresponding memory units.

This processing (data mapping) has been explained in Section 4.1 with an example (Table 2) to demonstrate the layer architecture of Z-TCAM. Vertical partitioning (VP) implies that a TCAM word of C bits is partitioned into N subwords; each subword is of w bits. The use of VP is to decrease memory size as much as possible. Horizontal partitioning (HrP) divides each vertical partition using the original address range of conventional TCAM table into L horizontal partitions. HrP cannot be used alone as it is area, power, and cost hungry but is used to create layers. HP results in a total of L × N hybrid partitions.The dimensions of each hybrid partition are K × w where K is a subset from original addresses and w is the number of bits in a sub word. Hybrid partitions spanning the same addresses are in the same layer. For example, HP21 and HP22 span the same address range and are in layer 2.

3. ARCHITECTURE OF Z-TCAM 3.1 Overall Architecture:

The overall architecture of Z-TCAM is depicted in Fig. 2 where each layer represents the architecture shown in Fig.3. It has L layers and a CAM priority encoder (CPE). Each layer outputs a potential match address (PMA). The PMAs are fed to CPE, which selects match address (MA) among PMAs.

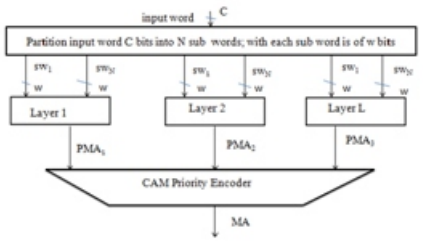


Fig 2: Architecture of Z-TCAM. (sw: subword, C: # of bits in the input word, PMA: potential match address, and MA: match address).

3.2 Layer Architecture:

This paper briefs the design with two layers: Layer 1 and Layer 2.Layer architecture is shown in Fig. 3. It contains 2 validation memories (VMs), 1-bit AND operation, 2 original address table address memories (OATAMs), 2 original address tables (OATs),K-bit AND operation, and a layer priority encoder (LPE).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3.2.1 Validation Memory:

Size of each VM is 2w × 1 bits where w represents the number of bits in each sub word and 2w shows the number of rows. A sub word of w bits implies that it has total combinations of 2w where each combination represents a sub word. For example, if w is of 4 bits, then it means that there are total of 24 = 16 combinations. This explanation is also related to OATAM and OAT. Each sub word acts as an address to VM. If the memory location be invoked by a sub word is high, it means that the input sub word is present, otherwise absent.

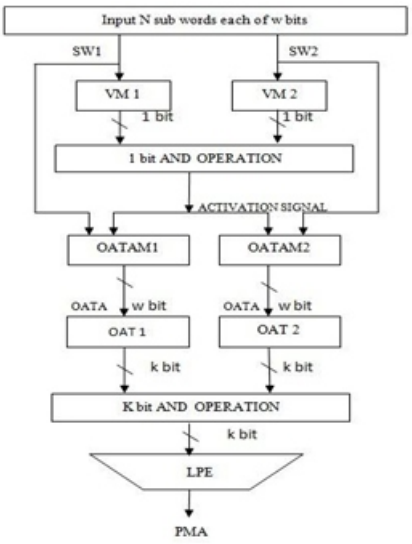


Fig 3: Architecture of a layer of Z-TCAM. (sw: subword, VM: validation memory, OATAM: original address table address memory, OAT: original address table, and LPE: layer priority encoder).

Thus, VM validates the input sub word, if it is present. In Tabel 2 VM 11 , VM 12 belongs to Layer 1 and VM 21 , VM 22 belongs to Layer 2. For example, Table 2 shows that sub words 00 and 01 are mapped in VM 11 , sub words 01 and 11 are mapped to VM 12,etc.This states that memory locations 00 and 01 should be high in VM11 and the remaining memory locations are set to low because their corresponding sub words do not exist.

3.2.2 1-bit AND Operation:

It ANDs the output of all VMs. The output of 1-bit AND operation decides the continuation of a search operation. If the result of 1-bit AND operation is high, then it permits the continuation of a search operation, otherwise mismatch occurs in the corresponding layer.

3.2.3 Original Address Table Address Memory:

Each OATAM is of 2w × w bits where 2w is the number of rows and each row has w bits. In OATAM, an address is stored at the memory location indexed by a subword and that address is then used to invoke a row from its corresponding OAT. If a subword in VM is mapped, then a corresponding address is also stored in OATAM at a memory location accessed by the sub word. For example, Table II shows OATAM21 where addresses are stored at the memory locations 00, 01, and 11. The output of OATAM is called as OATA. Hyphen "-" indicates that the corresponding memory location has no data because the corresponding sub word for the memory location is not present in VM.STTRAM is used to realize OATAM and OAT.

3.2.4 Original Address Table:

Dimensions of OAT are 2w× K where w is the number of bits in a subword, 2w represents number of rows, and K is the number of bits in each row where each bit represents an original address. Here K is a subset of original addresses from conventional TCAM table. It is OAT, which considers the storage of original addresses. An example of OAT is given in Table III, where 1 shows the presence of a subword at an original address.

3.2.5 K-bit AND Operation:

It ANDs bit-by-bit the read out K-bit rows from all OATs and forwards the result to LPE.

3.2.6 Layer Priority Encoder:

Because we emulate TCAM and multiple matches may occur in TCAM [17], the LPE selects PMA among the outputs of K-bit AND operation.

4. Z-TCAM OPERATIONS 4.1 Data Mapping Operation:

Classical TCAM table is logically partitioned into hybrid partitions. Each hybrid partition is then expanded into a binary version. Thus, we first expand x into states 0 and 1 to be stored in SRAM.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For example, if we have a TCAM word of 010x, then it is expanded into 0100 and 0101. Each subword, acting as an address, is applied to its corresponding VM and a logic "1" is written at that memory location. The same subword is also applied to its respective OATAM and w bits data are written at that memory location. During search, these w bits data act as an address to the OAT. The K bits data are also written at the memory location in OAT determined by its corresponding OATA. Thus, in this way, all hybrid partitions are mapped.

A subword in a hybrid partition can be present at multiple locations. So, it is mapped in its corresponding VM and its original address is/are mapped to its/their corresponding bit(s) in its respective OAT. Since a single bit in OAT represents an original address, only those memory locations in VMs and address positions/ original addresses in OATs are high, which are mapped while remaining memory locations and address positions are set to low in VMs and OATs, respectively.

Example of data mapping is shown in Table 2. We use Table 1 to be mapped to Z-TCAM. We take N = 2, L = 2, K = 2, and w = 2. After necessary processing, HP11, HP12, HP21, and HP22 are mapped to their corresponding memory units. In the example, we map hybrid partitions of layer 2 to their corresponding memory units. Hybrid partitions of layer 1 can be easily mapped in similar way.

Algorithm 1: Pseudo code for searching in a layer of Z-TCAM
INPUT : N sub-words
OUTPUT : PMA

1:Apply N sub-words

2:Apply all sub-words simultaneously to their VMs

3:Read all VMs concurrently

4:if all VMs validate their corresponding sub-words then

5:Sustain search operation

6:a. Read all OATAMs in parallel

7:b. Read all OATs simultaneously

8:c. AND bitwise all K –bits rows

9:d. Select PMA/mismatch occurs

10: else

11:Mismatch occurs

12: end if

Algorithm 2: Pseudo code for searching in Z-TCAM
INPUT : Search Key
OUTPUT : MA

1:Apply search key

2:Divide search key into N sub word

3:All layers use algorithm1 in parallel

4:Select MA among PMAs/mismatch Occurs.

Table 4.Example of a Search Operation in Layer 2 of Z-TCAM:

STEPS	ACTIVITY
1	Sub – word 1 = 00 Sub – word2 = 11
2	Sub – word1 is applied to VM21 Sub – word1 is applied to VM22
3	Read out bit from VM21 = 1 Read out bit from VM22 = 1
4	Have all VMs validated their corresponding sub – words?
5	Yes ,so sustain search operation
6	Read out data from OATAM21 = 0 Read out data from OATAM22 = 1
7	Read out data from OAT21 = 10 Read out data from OAT22 = 11
8	K – Bit AND operation result = 10
9	PMA = 2

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4.2 Search Operation:

4.2.1 Searching in a Layer of Z-TCAM:

Algorithm 1 describes searching in a layer of Z-TCAM. N subwords are concurrently applied to a layer. The subwords then read out their corresponding memory locations from their respective VMs.If all VMs validate their corresponding sub words (equivalent to 1-bit AND operation in Fig. 2), then searching will continue, otherwise mismatch occurs in the layer. Upon validation of all subwords, the sub words read out their respective memory locations from their corresponding OATAMs concurrently and output their corresponding OATAs. All OATAs then read out K-bit rows from their corresponding OATs simultaneously, which are then bitwise ANDed. LPE selects PMA from the result of the K-bit AND operation. Example of a search operation in layer 2 is shown in Table 4, following Algorithm 1. Memory blocks in Table 2 need to be searched.

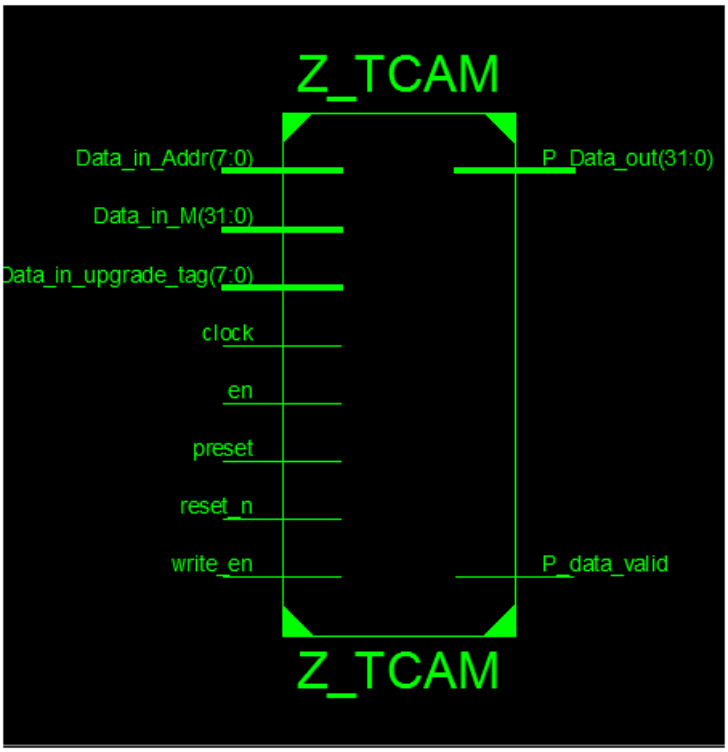
4.2.2 Searching in Z-TCAM:

Search operation in the proposed TCAM occurs concurrently in all layers, which follows Algorithm 2. Search key is applied to Z-TCAM, which is then divided into N subwords. After searching, PMAs are available from all layers. CPE selects MA among PMAs; otherwise a mismatch of the input word occurs. Table 5 provides overall search operation in Z-TCAM, which follows Algorithm 2.[1].

5. Z-TCAM IMPLEMENTATION AND RESULTS:

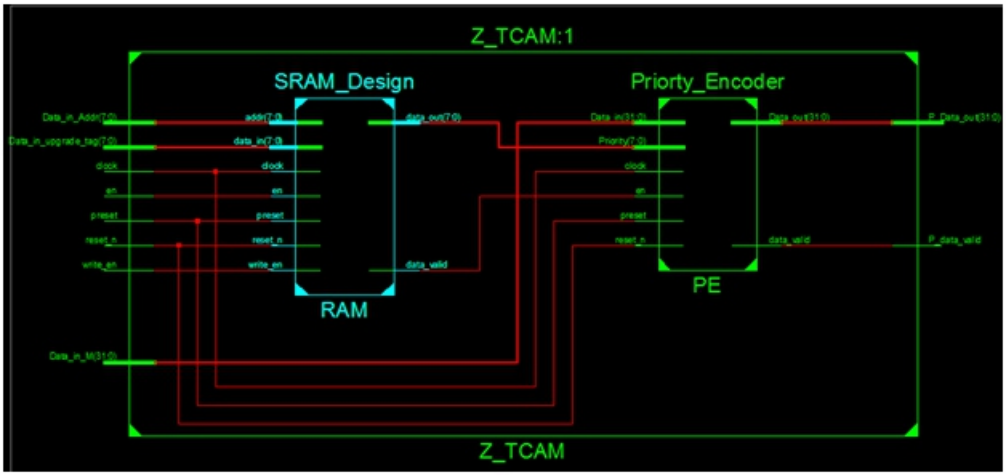
5.1 Delay Analysis:

The delay analysis has been done in xilinx 14.2 for STTRAM based Z-TCAM and the obtained output is shown in fig5.

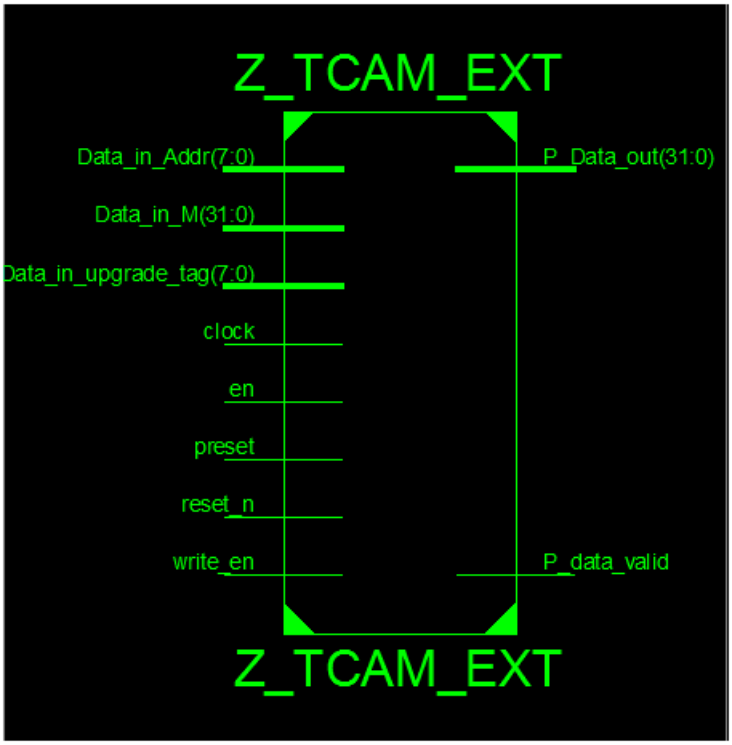


5.1.Z-TCAM block diagram

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

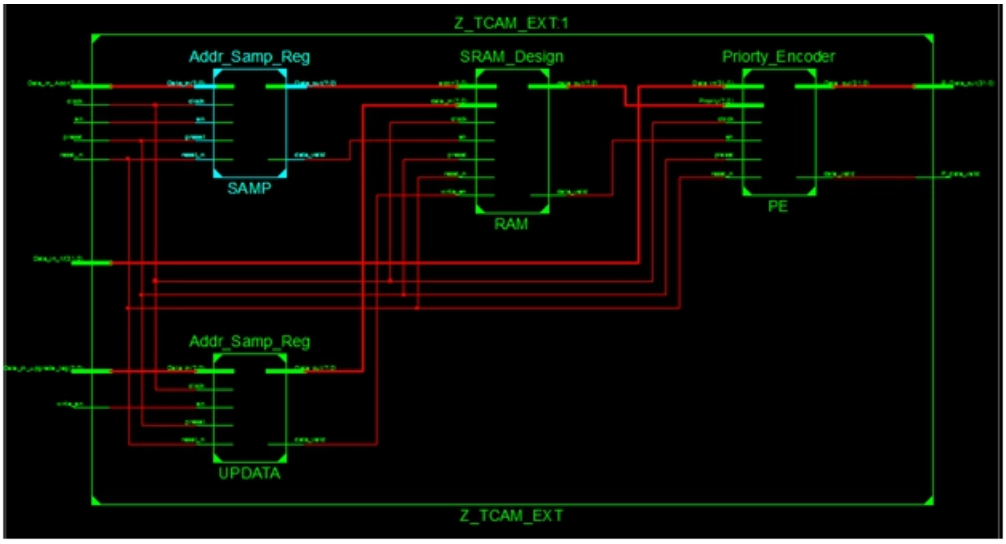


5.2.Z-TCAM PROPOSED METHOD

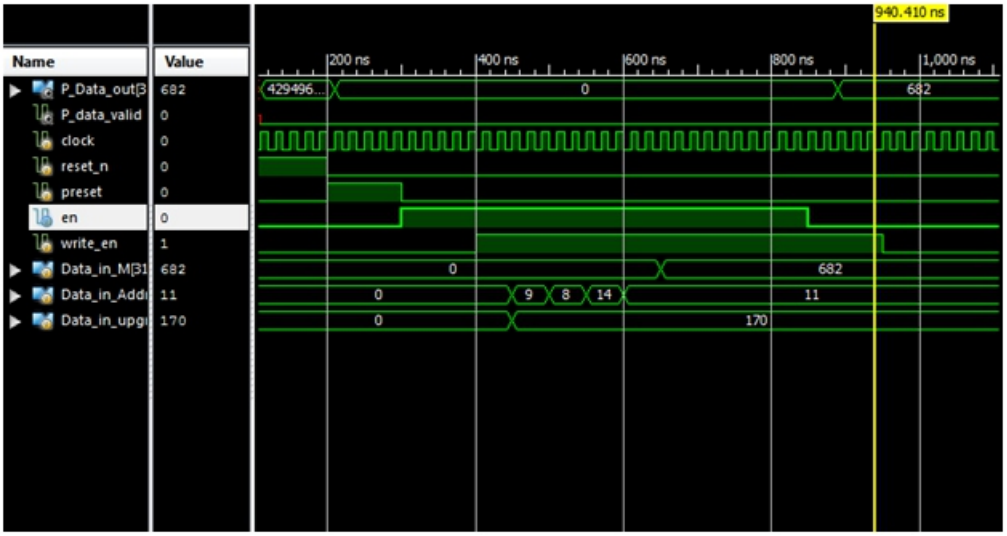


5.3.Z-TCAM ARCHITECTURE

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



5.4.RTL OF SRAM USING Z-TCAM ARCHITECTURE



5.5.Wave form of Z-Tcam architecture

6.Conclusion:

In this brief, we have presented a novel SRAM-based TCAMarchitecture of Z-TCAM. We have implemented two example designs of 512 × 36 and 64 × 32 of Z-TCAM on Xilinx spartan 6 FPGA. FPGA implementation is a big plus for Z-TCAM. Resources utilization, speed, and power consumption for different situations for the example designs on FPGA as well as in ASIC have been tabulated. Z-TCAM also ensures large capacity TCAM whereas this capability is lacked by conventional ones. Moreover, the proposed TCAM has a simpler structure, and very importantly, has a deterministic search performance of one word comparison per clock cycle

7. REFERENCES:

[1] Zahid Ullah, Manish K. Jaiswal, and Ray C. C. Cheung, "Z-TCAM: An SRAM-based Architecture for TCAM," IEEE Transaction on Very Large Scale Integration.(VLSI) Systems,vol.23,no.2, pp.402-406, 2015.

[2] Philip Asare and Ben Melton, "Towards An Early Design Space Exploration Tool Set for Spin Transfer Torque RAM (STT-RAM) Design," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.

[3]P. Mahoney, Y. Savaria, G. Bois, and P. Plante, "Parallel hashing memories: An alternative to content

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

addressable memories," in Proc. 3rd Int. IEEE-NEWCAS Conf., Jun. 2005, pp. 223–226.

[4] S. Dharmapurikar, P. Krishnamurthy, and D. Taylor, "Longest prefix matching using bloom filters," IEEE/ACM Trans. Netw., Apr. 2006.

[5] D. E. Taylor, "Survey and taxonomy of packet classification techniques," ACM Comput. Surveys, New York, NY, USA: Tech. Rep. WUCSE-2004-24, 2004.

[6] P. Mahoney, Y. Savaria, G. Bois, and P. Plante, "Transactions on high performance embedded architectures and compilers II," in Performance Characterization for the Implementation of Content Addressable Memories Based on Parallel Hashing Memories, P. Stenström, Ed. Berlin, Germany: Springer-Verlag, 2009.

[7]S. V. Kartalopoulos, "RAM-based associative content-addressable memory device, method of operation thereof and ATM communication switching system employing the same," U.S. Patent 6 097 724, Aug. 1, 2000.

[8]https://supportforums.cisco.com/document/60831/cam-content-addressable-memory-vs-tcam-ternary-content-addressable-memory .

[9] W. Jiang and V. Prasanna, "Scalable packet classification on FPGA,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 9, pp. 1668–1680, Sep. 2012.

[10] M. Becchi and P. Crowley, "Efficient regular expression evaluation: Theory to practice," in Proc. 4th ACM/IEEE Symp. Archit. Netw. Commun. Syst., Nov. 2008, pp. 50– 59.

[11] Xilinx, San Jose, CA, USA. Xilinx FPGAs [Online].

[12]W. Jiang and V. K. Prasanna, "Large-scale wire-speed packet classification on FPGAs," in Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays, 2009, pp. 219–228.

[13] W. Jiang and V. Prasanna, "Parallel IP lookup using multiple SRAM based pipelines," in Proc. IEEE Int. Symp. Parallel Distrib. Process., Apr. 2008, pp. 1–14.

[14] S. Cho, J. Martin, R. Xu, M. Hammoud, and R. Melhem, "CA-RAM: A high-performance memory substrate for search-intensive applications,"in Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw., Apr. 2007, pp. 230–241.

[15] M. Somasundaram, "Memory and power efficient mechanism for fast table lookup," U.S. Patent 20 060 253 648, Nov. 2, 2006.

[16] M. Somasundaram, "Circuits to generate a sequential index for an input number in a pre-defined list of numbers," U.S. Patent 7 155 563, Dec. 26, 2006.

[17] 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

[18] Xilinx, San Jose, CA, USA. Xilinx Xpower Analyzer

[19] OSUCells, Stillwater, OK, USA [Online]. Available: http://vlsiarch.ecen.okstate.edu

[20] S.-J. Ruan, C.-Y. Wu, and J.-Y. Hsieh, "Low power design of precomputation-based content-addressable memory," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 3, pp. 331–335, Mar. 2008.

[21] J. Raja, M. Kannan, VLSI implementation of high throughput MIMO OFDM transceiver for 4th Generation systems. ,international journal of Engineering and materials sciences,Vol. 19 ,October 2012, pp. 307-312.

[22] E. Konguvel, J. Raja, M. Kannan, A Low Power VLSI Implementation of 2X2 MIMO OFDM Transceiver with ICI-SC Scheme, International Journal of Computer Applications (0975 – 8887) Volume 77 – No.5, September 2013.

[23] A.Amjadha, E. Konguvel, J. Raja, Design of Multipath Delay Commutator Architecture based FFT Processor for 4th Generation System. International Journal of Computer Applications (0975 – 8887) Volume 89 – No 12, March 2014