INVESTIGATIONS ON IMPLEMENTATION OF TERNARY CONTENT ADDRESSABLE MEMORY ARCHITECTURE IN SPARTAN 3E FPGA

Anju Lakshmi. S.[1], Dr. David Solomon George [2],

[1]*Mtech scholar, Department of ECE, GECI, Kerala, India*
[2]*Associate Professor, Department of ECE, GECI, Kerala, India*

---***---

Abstract – *The Field Programmable Gate Array (FPGA) implementation of Ternary Content Addressable Memory (TCAM) is a demanding area of research to address the requirements of data base querying systems and high speed networking. Major investigation area in the Content Addressable Memory (CAM) architecture design is the performance metrics such as area, power and latency in the context of miniaturization, high speed and low power requirements of electronic gadget market. Z-TCAM is one of the latest popular architectures available in FPGA based TCAM. This paper is an investigation on implementation of Z-TCAM architecture in a low cost FPGA platform. The chosen hardware implementation platform is Digilent Basys2 board which works well with all versions of Xilinx Integrated Synthesis Environment (ISE) tool and free Web Pack. The architecture under study is implemented in SPARTAN 3E FPGA to obtain hardware utilization of 12.26%, latency of 3110.55 nS and power consumption of 45.16 mW.*

***Key Words*: FPGA, CAM, BCAM, TCAM, Z-TCAM, Hybrid Partitioning**

1. INTRODUCTION

The era of high speed computation demands very high speed computer memories. CAM is a kind of computer memory used in fast look up application [1]. CAM is widely used in applications that demand very fast memory access like internet router and switches.

The specialty of CAM is that it searches memory using a content rather than address [2]. The CAM behavior is opposite to that of RAM. The search operation outcome is address of memory location where the content matches is found with input data. The search operation ensures constant search time.

There are two major classification of CAM; Binary Content Addressable Memory (BCAM) and Ternary Content Addressable Memory (TCAM). The former supports only two logic states logic'0' , logic'1' while the later supports an additional logic state 'X' which is always considered as matched condition regardless of input state. TCAM is superior in performance than BCAM due to its ability to store three states. Typical TCAM cell structure comprises of storage circuit along with a dedicated comparison circuitry.

There are several limitations for TCAM technology when compared to SRAM technology such as lower density of integration due to presence of dedicated comparison circuitry along with each cell which in turn increase the complexity of architecture and increases cost per bit [3]. The matching circuitry increases access time of TCAM due to presence of additional logic and massive parallelism creates loading effects [4]. TCAM testing is time consuming [5]. CAM technology faced less commercial competition hence TCAM doesn't have as much varieties as that available for SRAM.

TCAM finds wide application in the area of networking; it is an important component of network routers which decides the operating speed of the network [6],[7],[8]. Other major applications are data compression, real time pattern matching in virus detection, gene pattern matching is bio informatics, translation look aside buffer (TLB) in microprocessor, image processing, intrusion detection systems, database management and artificial intelligence. Field Programmable Gate Arrays are best suited for implementing real time systems due to the ability of reconfiguration, quick prototyping and supports massive parallel operations [9]. Hence FPGA based TCAM implementation is a worthy area of research

SRAM is widely available in various varieties which avoid huge licensing and royalty cost. The CAM technology still has limited pattern capacity because this technology is evolving very slowly. Only a slight increase in bit capacity and slight decrease in cost is being achieved in recent researches in this area. Hence the CAM technology is a challenging research area which demands more optimization in chip area, power dissipation, bit capacity, access time, architectural complexity etc.

The Paper is organized in 5 sections. Literature survey is briefed in section 2.The Z-TCAM architecture and algorithms are introduced in section 3. Section 4 presents the actual implementation results of the Z-TCAM architecture. Finally, section 5 concludes the investigation results.

2. LITERATURE SURVEY

An extensive look through on the history of CAM is being done up to the recent advancement in CAM technology. Some of the significant mile stones in typical CAM technology and SRAM based CAM technology, which is the latest architectural development of the area under study are reviewed below.

2.1 CAM Technology

The concept of CAM was invented by Dudley Allen Buck in 1955. The major interface definition for CAM and other Network Search Elements was specified in an interoperability agreement called Look-Aside interface (LA1 and LA1B) developed by Network Processing Forum, which later merged with Optical Internet Working Forum. Some of the prominent CAM technologies excluded from the category of RAM based CAM technology is referred briefly.

Kostas Pagiamtzis et al. proposed, Using Cache to reduce power in Content-Addressable Memories (CAMs) [10]. The method is similar to that of cache in processor system. Avoid search in large and high power CAM. Matched result from cache saves power. Effectiveness of this technique depends on cache hit rate. Use of these caches requires major modifications to the memory structure and hierarchy to fit the design and also require more area.

Low power design of pre-computation based content-addressable memory is proposed by S.J. Ruan et.al. [11]. Pre Computation is done using parameter extractor. Additional bits are derived from stored word and these bits are used in initial search before going to search main word. This design ensures low-power, low cost, low-voltage, high-reliability than its predecessors. It requires considerable modification to the memory architecture in order to achieve high performance. Special memory cell design is required for reducing power consumption. As the input bit length increases, the delay of the ones-count parameter extractor also increases significantly.

2.2 RAM based Implementations of CAM

Hashing is the transformation of string of characters into shorter fixed length value or key that represents the original string. Hence it is a faster way of indexing and retrieving items in a data base. P. Mahoney et al. proposed Performance Characterization for the Implementation of Content Addressable Memories Based on Parallel Hashing Memories [12]. Where quick search operation is achieved by the simple arithmetic function called hash function on key. Look Up involves a single memory access followed by a parallel key matching operation. This technique assures low-latency, high-bandwidth search performance, large capacity, low power consumption. It has higher performance than other techniques, especially when large databases are used and have area and power savings .The disadvantages of this technique are collisions and bucket overflow. The performance of this technique greatly depends on the number of stored elements. Usually the performance degrades with size. It emulates BCAM, not TCAM.

S. Cho et al. designed CA-RAM: A High-Performance Memory Substrate for Search-Intensive Applications [13]. It is a direct hardware implementation of hashing technique using a conventional high-density memory and a number of match logic blocks. Searchable records are pre-classified and stored in CA-RAM at a location determined by a hash function, defined on their search key. Once a database has been built, look up operation typically involves a single memory access followed by a parallel key matching operation. CA-RAM capitalizes on dense SRAM and DRAM designs. It achieves comparable search performance while occupying much smaller area and consuming significantly less power because the bit-density of CA-RAM is much higher than that of CAM, nearly five times if DRAM is used in the implementation. CA-RAM achieves lower latency and higher search bandwidth compared with software techniques, while eliminating undesirable cases such as cache pollution. It can be easily plugged into the memory hierarchy as a regular address-based RAM. The presence of don't care bit in stored key results in its duplication in multiple buckets which increases the required size. If search key contain don't care bits it cause increase in search lookup in multiple buckets which in turn causes performance degradation.

W. Jiang et al. proposed, Parallel IP Lookup using Multiple SRAM based Pipelines [14]. A partition based parallel IP lookup engine using multiple SRAM-based linear pipelines is depicted along with two-level mapping scheme to balance the memory distribution over multiple linear pipelines as well as across all pipeline stages. To balance the load among pipelines a flow pre-caching scheme is developed. Payload exchange approach exploits the pipeline delay which is used to maintain the intra-flow packet order. It can achieve a high throughput of up to 10 billion packets per second. The disadvantage of algorithmic search solutions is that it takes multiple clock cycles

M. Somasundaram acquired Patent in Memory and power efficient mechanism for fast table lookup [15]. In this work he combined RAM and CAM to emulate CAM functionality. Distinguishing bits in CAM entry is used to partition TCAM table. Using completely random data for partitioning is a very tedious and time consuming job. This method uses TCAM as a part of the overall architecture, hence inherits the intrinsic TCAM disadvantages in the overall architecture.

3. Z-TCAM ARCHITECTURE

Z-TCAM is a popular novel memory architecture proposed by Zahid Ullah et.al. . It is one of the TCAM architectures successfully implemented in FPGA. In this architecture TCAM functionality is emulated using SRAM thus the typical shortcoming of TCAM is avoided and the benefits of SRAM is inherited in Z-TCAM architecture.

3.1 Overall Architecture of Z-TCAM

The overall architecture of Z-TCAM is shown is shown in Fig 1. It comprises of 'L' number of layers and a priority encoder. A layer contains data that shares same address range during hybrid partitioning [16]. The Priority encoder is used to resolve priority. The input to the Z-TCAM architecture is a search word of 'C' bit word length, which is divided into 'N' SUB Words (SWs) each of 'w' bit word length for the sake of mapping to SRAM units. If the search word is matched with any data in any layer the corresponding layer generates a Potential Match Address (PMA) which is the address location where the matching data is stored. In the case multiple matches, a single output is obtained by resolving priority. In this architecture higher priority is assigned to the lowest address. The final output after resolving priority is considered as match address.

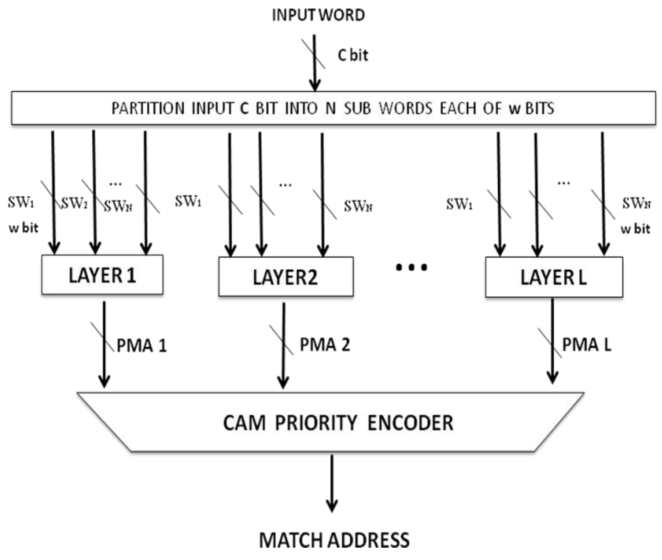


Fig -1: Overall architecture of Z-TCAM. [3]

3.2 Layer Architecture of Z-TCAM

Every layer in Z-TCAM has similar architecture. The layer architecture is shown in Fig 2. Data spanning a fixed address range is mapped to a layer. Layer architecture comprises of three groups of memories namely Validation Memory (VM), Original Address Table Address Memory (OATAM) and Original Address Table (OAT), two kinds of ANDing operations and a Layer Priority Encoder (LPE).

The VM validates whether the input SW is present in the layer. VM is of size $2^w \times 1$ where 'w' is the number of bits in a SW and 2^w is the number of rows in a VM. A SW of w bits implies that it has total combinations of 2^w where each combination represents a sub word. For example, if w is of 8 bits, then it means that there are total of 2^8=256 combinations. This explanation is applicable to OATAM and OAT also. Each SW acts as an address to VM. If the memory location corresponding to a SW is high, it means that the input SW is present, otherwise absent.

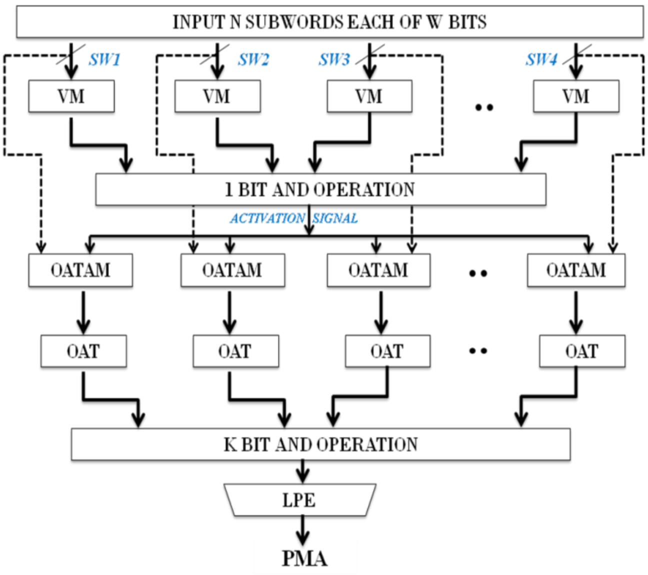


Fig -2: Layer architecture of Z-TCAM. [3]

The 1-bit AND Operation block ANDs the output of all VMs. If all the N SWs are present in their corresponding VM, the output of 1-bit AND operation will be high and this signal act as an activation signal to successive layers of memory. The status of activation signal indicates whether the search word is present in a layer. Thus the output of 1-bit AND operation decides the continuation of a search operation.

OATAM stores the address of OAT memory locations to which data is being mapped. Each OATAM is of $2^w \times w$ bits where the number of row present is 2^w and each row has w bits. In OATAM, an address is stored at the memory location indexed by a SW and that address is then used to invoke a row from its corresponding OAT. If a SW in VM is mapped, then a corresponding address is also stored in OATAM at a memory location accessed by the SW. The output of OATAM is the input to OAT. A Hyphen symbol indicates that the corresponding memory location has no data because the corresponding SW for the memory location is not present in VM.

OAT is the last layer memory mapping to original TCAM table. Dimensions of OAT are $2^w \times k$ where w is the number of bits in a SW, 2^w represents number of rows. k is the number of bits in each row where each bit represents an original address. Also k is a subset of original addresses from conventional TCAM table. The K-bit AND Operation block ANDs bit-by-bit the read out K-bit word from all OATs and forwards the result to priority encoder. In case of multiple matches, the LPE selects high priority PMAs among the outputs of K-bit AND operation.

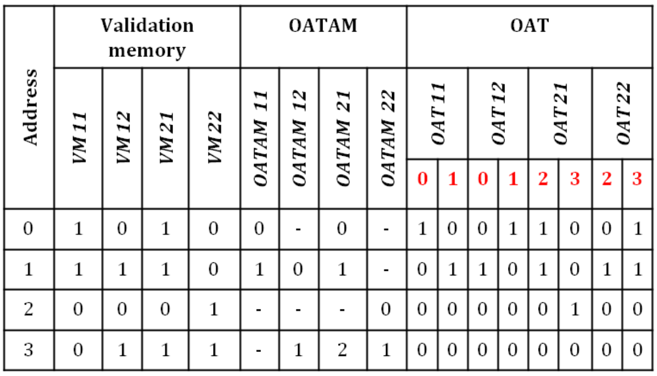
3.3 Data Mapping Technique

An example of hybrid partitioned TCAM table is depicted in Table 1. Mapping of the same to corresponding memories in each layer is shown in Table 2. Layer 1 spans two addresses 0,1each of which stores exactly one data word. HP11 and HP12 are the two sub tables belonging to layer1 of which HP11 contains SW '00'in address location 0 and'01' in address location 1.HP12 contains SW '11'in address location 0, SW '01'in address location1. Layer 2 spans addresses 2, 3, both of them seems to store two data words due to the presence of don't care state. HP21 and HP22 are the sub tables of layer 2. HP21 contains SWs '00', '01' in address location 2 and SW '11' in address location 3. Similarly, HP22 contains SW 11 in address location 2 and SWs '10' and '11' in address location 3.

Table-1: Traditional Hybrid Partitioned TCAM Table [3]

Address	Ternary Data		Layer
0	HP 11 00	HP 12 11	1
2	HP 21 0X	HP 22 11	2

Table -2: Data Mapping to Layer Architecture [3]



Address	Validation memory				OATAM				OAT							
	VM 11	VM 12	VM 21	VM 22	OATAM 11	OATAM 12	OATAM 21	OATAM 22	OAT 11		OAT 12		OAT21		OAT22	
									0	1	0	1	2	3	2	3
0	1	0	1	0	0	-	0	-	1	0	0	1	1	0	0	1
1	1	1	1	0	1	0	1	-	0	1	1	0	1	0	1	1
2	0	0	0	1	-	-	-	0	0	0	0	0	0	1	0	0
3	0	1	1	1	-	1	2	1	0	0	0	0	0	0	0	0

3.4 Z-TCAM Algorithms

Algorithm 1:

Pseudo code for searching in a layer of Z-TCAM [3]

INPUT: N sub words

OUTPUT: PMA (Potential Match Address)

```
Apply N sub words
Apply all SWs simultaneously to their VMs
Read all VMs concurrently
If all VMs validate their corresponding sub words
then
Sustain search operation
    a. Read all OATAMs in parallel
    b. Read all OATs in simultaneously
    c. AND  bit-wise all K-bit rows
    d. Select PMA/mismatch occurs
else
        Mismatch occurs
end if
```

Algorithm 2:

Pseudo code for searching in Z-TCAM [3]

INPUT: Search key

OUTPUT: MA (Match Address)

```
Apply search key
Divide search key into N sub words
All layers use algorithm 1 in parallel
Select MA among PMA/mismatch occurs
```

4. IMPLEMENTATION RESULTS

All the implementations are modelled using Verilog HDL coding in Xilinx ISE Design Suite 14.2. Simulations are obtained from ISim simulator. Power analysis is done using Xilinx Power Analyser (XPA) and Xilinx Power Estimator (XPE) software. The architecture is mapped to target board using Plan Ahead tool.

The RTL schematic of the latest SRAM based TCAM implementation in SPARTAN 3E FPGA is shown below. In Fig 3, layer 1 is elaborated to show the internal blocks such as VM, ONE-BIT AND, OAT, OATAM, K-BIT AND LPE. Layer 2 is represented as a block only. The outputs of both layers are fed to CPE which generates MA of a search word.

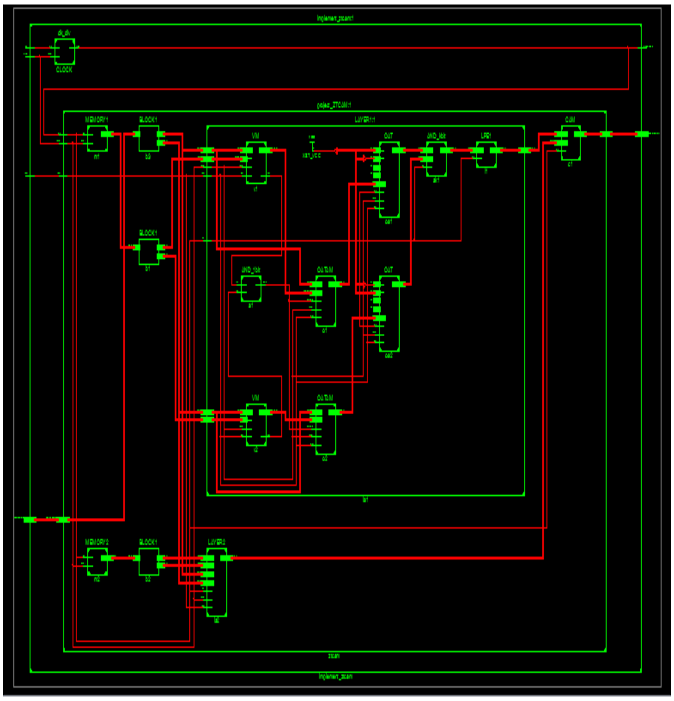


Fig -3: RTL schematic of Z-TCAM

The technology schematic of Z-TCAM is shown in Fig 4 is a pictorial representation of pre-optimized Z-TCAM architecture in terms of symbols of logical units such as buffers, LUTs, multiplexers, digital gates, flip flops etc.

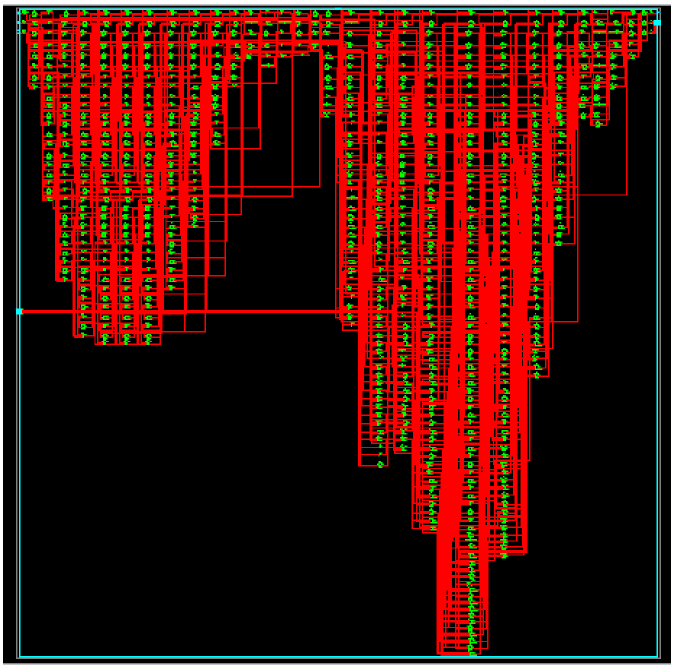


Fig -4: Technology schematic of Z-TCAM

The functional simulation result of Z-TCAM implementation in SPARTAN 3E performed with Xilinx ISim HDL simulator is shown in Fig 5.The search word 1111 is stored in layer 2, address location '11'.The output is delivered after 4 clock cycles so the system has a latency of 4 clock cycles. Afterwards an output is generated in every clock cycle which means the system has single cycle throughput.

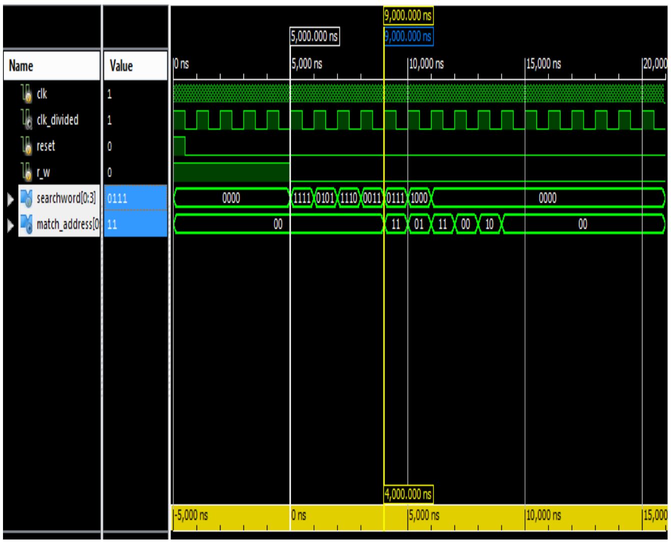


Fig -5: Functional Simulation of Z-TCAM

Timing simulation shown in Fig 6 accounts the latency associated with the implementation of the Z-TCAM architecture in actual hardware. Ruler in the bottom of simulation window shows that the output corresponding to the input provided in 0 nano Second time unit is delivered on 3110.556 nano Seconds which mark the latency of existing architecture.

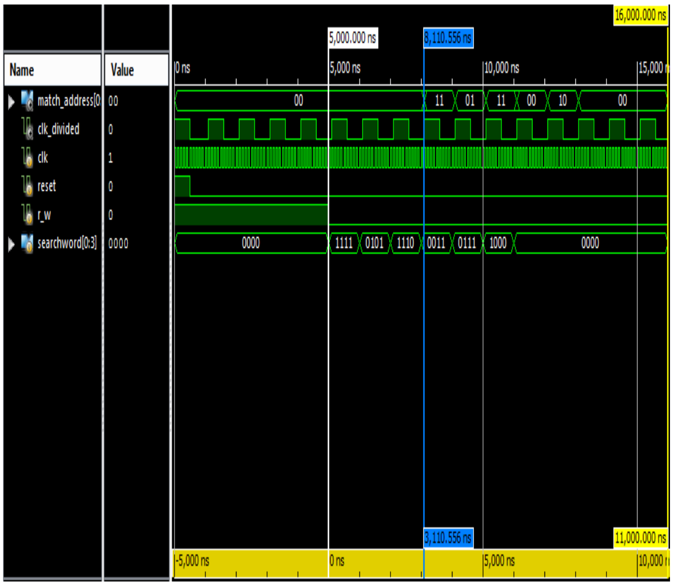


Fig -6: Timing Simulation of Z-TCAM

The design summary of Z-TCAM architectures is given in Table 3.

Table-3: Hardware utilization summary

Device Utilisation Summary of Z-TCAM			
Logic Resources	**Used**	**Available**	**Utilization**
Number of Slices	161	960	16.77%
Number of Slice Registers	164	1920	8.54%
Number of 4 Input LUTs	263	1920	13.69%
Number of Bonded IOBs	10	83	12.04%
Number Of BUFGMUXs	4	24	16.66%
TOTAL	602	4907	12.26%

Table-4: Performance Analysis

PARAMETERS	Z-TCAM IMPLEMENTATION ON SPARTAN 3E
Area	12.26%
Latency	3110.556 nS
Power Consumption	45.16 mW

Table 4 shows the results obtained from various analytical tools. Area is obtained from design summary analysis, latency is obtained from ISim simulator and power consumption is obtained from XPA and XPE tools.

5. CONCLUSIONS

The popular SRAM based TCAM architecture namely Z-TCAM is implemented in Xilinx Spartan 3E FPGA at a clock frequency of 50MHz. The hardware simulation is done using Xilinx ISE Design Suite 14.2. The investigation was done on performance matrices such as hardware utilization, latency and power consumption. The system implementation concluded a hardware utilization of 12.26%, Latency of 3110.556 nS, Power Consumption 45.16 mW.

REFERENCES

[1] H. Lim, J. Seo and Y. Jung, "High speed IP address lookup architecture using hashing," Communications Letters, IEEE, Vol.7, Iss.10, Oct. 2003, pp.502- 504.

[2] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: a tutorial and survey, Solid-State Circuits," IEEE Journal of, vol. 41, no. 3, 2006, pp. 712-727.

[3] Zahid Ullah, Manish K. Jaiswal and Ray C. C. Cheung, "Z-TCAM: An SRAM-based Architecture for TCAM," IEEE transactions on very large scale integration (VLSI) systems, vol. 23, no. 2, Feb. 2015, pp. 402–406.

[4] Md. Atik Foysal, Md. Zahidul Anam, Md. Shoriful, slam Intisar Tahmid, Kartick Mondal, "Performance Analysis of Ternary Content Addressable Memory (TCAM)," Proceedings of 2015 3rd International Conference on Advances in Electrical Engineering, Dec. 2015, pp.17-19 .

[5] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "Design techniques and test methodology for low-power TCAMs," IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 6, 2006, pp. 573–586.

[6] Y. J. Chang, "A high-performance and energy-efficient TCAM design for IP address lookup," IEEE Trans. Circuits Syst. II, vol. 56, no. 6, 2009, pp. 479–483.

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

[8] S. Dharmapurikar, P. Krishnamurthy, and D. Taylor, "Longest prefix matching using bloom filters," IEEE/ACM Trans. Netw., vol. 14, no. 2,2006, pp. 397–409.

[9] Wayne Wolf ,FPGA-Based System Design, 3rd Ed., Pearson, 2014.

[10] Kostas Pagiamtzis and Ali Sheikholeslami, "Using Cache to reduce power in Content-Addressable Memories (CAMs)," IEEE CUSTOM Integrated Circuits Conference, vol. 41, no. 3, Mar. 2006, pp 712-726.

[11] S. J. Ruan, C.Y. Wu, J. Y. Hsieh (2008) "Low power design of pre-computation based content-addressable memory", IEEE Transactions Very Large Scale Integration (VLSI) Systems, vol. 16, no. 3, Mar. 2008, pp 331-335.

[12] P. Mahoney, Y. Savaria, G. Bois, and P. Plante, "Performance Characterization for the Implementation of Content Addressable Memories Based on Parallel Hashing Memories," The 3rd, International IEEE-NEWCAS Conference, 2005, pp. 223–226

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

[14] W. Jiang and V. Prasanna, "Parallel IP lookup using multiple SRAM based pipelines," Proc. IEEE International Symposium in Parallel and Distributed Processing, 2008, pp. 1–14.

[15] M. Somasundaram, "Memory and power efficient mechanism for fast table lookup," U.S. Patent 20060253648, November 9, 2006.

[16] Z. UIlah, K. ligon, and S. Baeg, "Hybrid partitioned SRAM-based ternary content addressable memory," Circuits and Systems 1: Regular Papers, IEEE Transactions on, vol. 59, no. 12, Dec. 2012, pp. 2969-2979.