IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
Research Paper **© 2012 IJFANS. All Rights Reserved**, **Journal Volume 11, Iss 12 2022**

ADDRESSABLE SECONDARY MEMORY

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

[#2]DOOSA KAVITHA, ***Asst. Professor,***

Department of Electronics Communication Engineering,

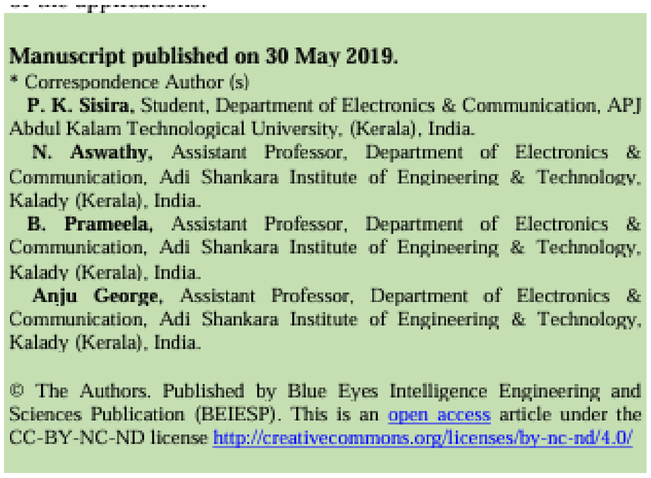
Sree Chaitanya Institute of Technological Sciences, Karimnagar, Ts.

Abstract: Memory technology is required for software that enables for quick searches. Computers use Content Addressable Memory (CAM), a specific type of memory, for searches. CAM allows you to retrieve stored data based on its content rather than its physical location. Ternary CAM (TCAM), an upgraded type of CAM, can hold unnecessary data. TCAM is particularly strong in router-based networking applications. An Energy Efficient TCAM (EE-TCAM) is one that consumes the least amount of power, according to considerable research into TCAM design methodologies. Because just one SRAM row is accessed at a time during search operations, EE-TCAM uses less power than other SRAM-based TCAM systems. Other systems, on the other hand, dedicate the entire SRAM to searches. After partitioning the TCAM table, a pre-classifier is constructed. This study's major purpose is to create an EE-TCAM for the Zybo7000 platform using the Vivado design suite and the Verilog HDL language. A full functional study is performed to explore the power, latency, and resource needs of a 6*6 EE-TCAM. The statistics show that EE-TCAM uses very little energy and has very little latency..

***Keywords*:** Memory, TCAM, EE TCAM, Pre-classifier

1. INTRODUCTION

This storage media is also known as content-addressable memory (or CAM). By reading the data and broadcasting the corresponding address, CAM offers access to data without the requirement for an explicit address. It compares the data that was previously saved with the data that is now in memory to see if they are the same. If the data word was found, CAM will send the coordinates to the requester. CAM, like random-access memory (RAM), can be read and written to. In addition to these features, CAM offers real-time search capabilities. Computer-aided manufacturing (CAM) is used extensively in networking, neuromorphic associative memory, reconfigurable computing, analytics, text mining, multimedia, and other domains. CAM encompasses both binary and ternary content accessible memory (BCAM and TCAM, respectively). Because it saves both 1s and 0s, BCAM uses an exact match on the table key. TCAM can deal with incomplete matches even if it just saves 1s and 0s since it can store a third "don't care" (x) bit. TCAM, like BCAM, allows you to search for information in a variety of ways. TCAM is widely used because of these benefits.


Manuscript published on 30 May 2019.
* Correspondence Author (s)
P. K. Sisira, Student, Department of Electronics & Communication, APJ Abdul Kalam Technological University, (Kerala), India.
N. Aswathy, Assistant Professor, Department of Electronics & Communication, Adi Shankara Institute of Engineering & Technology, Kalady (Kerala), India.
B. Prameela, Assistant Professor, Department of Electronics & Communication, Adi Shankara Institute of Engineering & Technology, Kalady (Kerala), India.
Anju George, Assistant Professor, Department of Electronics & Communication, Adi Shankara Institute of Engineering & Technology, Kalady (Kerala), India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
Research Paper **© 2012 IJFANS. All Rights Reserved**, **Journal Volume 11, Iss 12 2022**

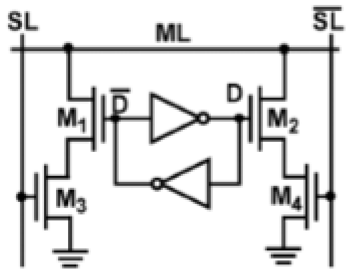


Fig 1. A BCAM Cell

CAM cells are in charge of both bit storage and bit comparison. Bits are stored in SRAM cells, which are built up of cross-coupled circuits. Four NMOS transistors are used for bit comparison. TCAM cells are arrayed in two dimensions. Search lines (SL) and match lines (ML) can be used to connect cells in the same row and column. In Figures 1 and 2, two examples of CAM cells are shown: BCAM and TCAM. M1, M2, M3, and M4 NMOS transistors are used in the comparing circuit. The CAM cell nodes D and Dbar are employed as bit storage. Below the match line (ML) are the search lines (SL and SLbar). The input bit will influence both SL and SLbar. One SRAM cell is enough to store a bit in a BCAM cell, but two are needed to store a ternary bit in a TCAM cell. A BCAM cell normally has 10 transistors, whereas a TCAM cell typically has 16.

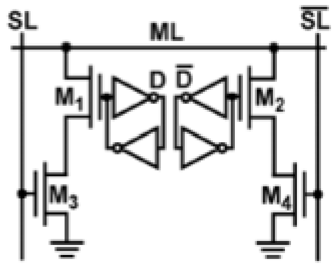


Fig 2. A TCAM Cell

A TCAM arrangement is shown in Figure 3. Before a search, all MLs are raised up to VDD and all SLs are permitted to drop to the ground. ML will not change as long as all cells in the same ML have a matching condition at VDD. All cells that have a matching ML will send it to Gnd. The primary encoder is given the MLs.

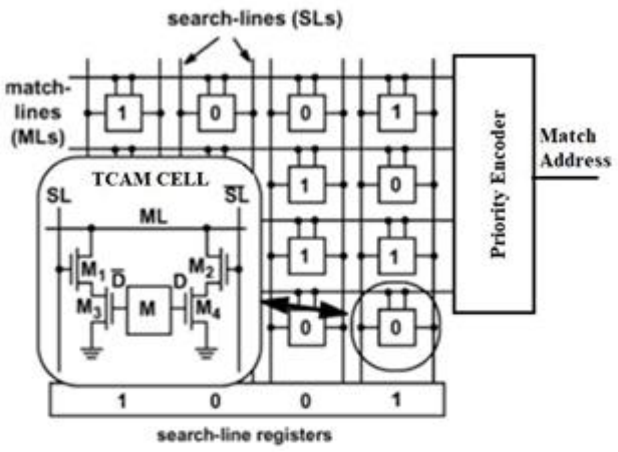


Fig 3. A TCAM Array

Static RAM outperforms TCAM in some cases. Its small size and limited memory hinder it from doing predictable jobs, therefore it is slow. To solve these challenges, Z. Ullah [3] developed a new way of building TCAMs utilizing SRAM. Unlike traditional TCAMs, which use xor or NOR gates for comparison, SRAM-based TCAMs use random-access memory.

2. BACKGROUND STUDY

A search of the literature on SRAM-based TCAM indicates a dearth of articles on the subject. The parallel hashing memory architecture presented in [4] is an alternative to content-addressable memory. It is simple to simulate CAM procedures using hash tables kept in parallel in RAM. Performance degrades when the failure chance of this memory system increases before it reaches full capacity. When more RAM is allocated, a container overflow is unavoidable. Sanggyeun Cho suggested a low-power FPGA-based CAM that searches SRAM blocks hierarchically. If the SRAM portion finds a match, the search is ended, conserving energy. However, in the worst-case situation, a significant amount of energy is consumed, which is a big problem in digital design.

The Xilinx application note addresses two separate FPGA applications. The first employs Xilinx BRAM space, while the second employs

16-bit shift registers. TCAMs based on shift registers are suitable for small devices.

These SRAM-based TCAMs have faster throughput, higher memory utilization, support for huge bit patterns, and consistent one-word search performance. SRAM is used in a wide range of TCAM applications. Z. Ullah offered four different SRAM-based TCAM alternatives. They perform comparable functions and are organized hierarchically.

The first TCAM to use static random access memory was the Hybrid Partitioned (HP) TCAM. HP-TCAM divides the conventional TCAM table in half to form TCAM sub-tables that are linked to SRAM memory. To accomplish its search, the HP-TCAM employs two SRAMs followed by an AND operation. HP-TCAM performs poorly when compared to Z-TCAM. (E)-TCAM, unlike Z-TCAM and HP-TCAM, does not have an infinite number of levels. E-TCAM, in addition to being faster than the original two TCAM systems, employs SRAM. UE-TCAM is a low-power memory technology comprised of two layers of TCAM memory based on SRAM. It outperforms HP-TCAM, Z-TCAM, and E-TCAM in terms of performance, resource usage, power consumption, and latency.

The Resource Efficient SRAM based TCAM (REST) is demonstrated, which uses resources similar to those shown. REST's Virtual Block approach is still quite effective in reducing memory use. Because there are fewer CPUs, power consumption is reduced. The Scalable and Modular Architecture of the Scalable TCAM presented in can be improved in a variety of ways. It consumes a lot of power despite its high output. Despite its use of multi-port SRAM, which allows for several pumps, the TCAM system consumes a significant amount of energy.

These three components will be found in all TCAMs that employ SRAM. The TCAM database is divided into rows first. Vertical partitioning, the second type of partitioning, divides the TCAM database using columns. Finally, the TCAM table is divided in two, first along the columns and then along the rows. These dividers are complicated and require extra room.

Table.I Comparison of Different SRAM based TCAMs

Architecture	Size	FPGA	Speed MHz	Throughput Mbit/s	Power mW
HP-TCAM [10]	512*36	Virtex 6	35	4.2	188
Z-TCAM [11]	512*36	Virtex 6	118	5.6	109
E-TCAM [12]	512*36	Virtex 6	159	5.8	91
UE-TCAM [13]	512*36	Virtex 6	164	7.1	78
REST [14]	72*28	Virtex 6	35	1.4	161
Scalable TCAM [15]	1024*150	Virtex 6	20.4	3221	4587
EE-TCAM [16]	512*36	Virtex 6	336	11.6	33.7

The Energy Efficient TCAM (EE-TCAM), as indicated in Table I, is the most effective TCAM employing SRAM. Z.Ullah intends to release EE-TCAM in 2018. This method lowers the consequences of data search partitioning by employing a pre-classifier-based strategy. This building employs two unique architectural styles. The TCAM table is divided into many subtables first. TCAM sub-tables are converted into SRAM blocks at the next level of the architecture's SRAM implementation. To search for each incoming TCAM word, all TCAMs activate the whole SRAM memory. This system consumes less power than others since it uses SRAM memory for lookups selectively rather than the complete memory.

3. ARCHITECTURE AND CLASSIFICATION OF EE-TCAM

The fundamental goal of this research is to create a 6*6 EE-TCAM using a pre-classifier-based approach. This setting only activates one SRAM row per incoming TCAM word. The TCAM database is divided into sectors by bits

ARCHITECTURE

The architecture represented in Figure 4 is made up of SRAM cells and a pre-classifier block. Reading a W-bit TCAM word and decoding it into V-bit subwords. Each row of SRAM cells is assigned a name based on its component words, which are subsequently read. The outputs of each row's SRAM units are bit by bit ANDed to form matched words, which are then given to the row's priority encoder. At the output, a multiplexer is used to identify which priority encoder is active.

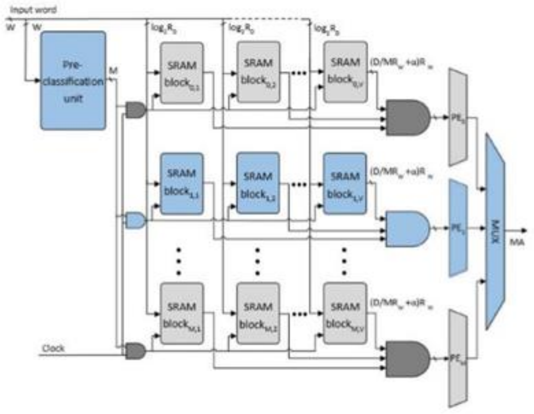


Fig 4. Architecture of EE-TCAM

The 6*6 TCAM necessitates four rows of 24 SRAMs. Each cell has two SRAMs. These several SRAMs were derived from Table II's 6x6 TCAM table.

Table.II A 6*6 TCAM TABLE

Addresses	TCAM Words
0	001001
1	11x100
2	x10010
3	100x11
4	0x01x1
5	x10001

Memory mapping is used to store the TCAM words from Table 2 in SRAM bytes. We use the b1 and b3 bits to organize the words into four tables. Memory mapping and partitioning are extensively discussed in the section on classification.

Figure 5 depicts the unit size of the Pre-classifier. This device has multiplexers and an address decoder. Multiplexers are required due to the size of the EE-TCAM segments.

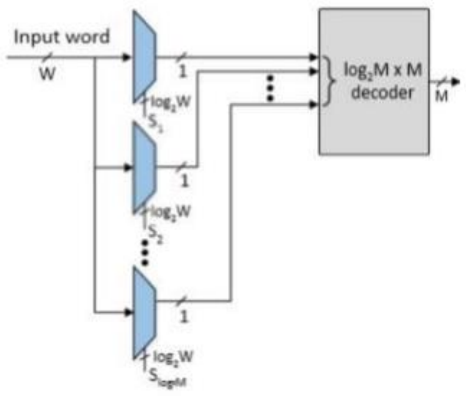


Fig 5. A Pre-Classifier unit

Each of the six possibilities must have three distinct pick lines. Because the pieces are based on b1 and b3, utilize lines 001 and 011. The multiplexer sends data to the 24 encoder.

CLASSIFICATION

Table III shows a dismantled TCAM table made up of 66 distinct cells. Bits b1 and b3 act as the glue that holds the division together. If both b1 and b3 are x before being separated, they will initially appear as 0. TCAM ST0 addresses have b1=0 and b3=0 values. The 01 indicates one address in subgroup ST1, the 10 another, and the 11 yet another. SRAM memory is mapped with integers grouped into groups based on the bits b0, b2, 4, and 5. The partitioned TCAM database is represented by four rows of SRAM blocks, each having two 4*2 RAMs. The memory map shows where each TCAM word is stored in memory. Table 2 shows that each TCAM index corresponds to a certain SRAM. If a RAM-based version is to be able to find in the same way as TCAM can, mapping must also include filling up RAM. The TCAM cell is made up of two SRAMs. To map a bit in the TCAM table, two SRAMs are required. The match vector for the word is presented in a single RAM column.

Table.III Partition based on b1 and b3 bits

Addresses	TCAM words						
	b_0	b_1	b_2	b_3	b_4	b_5	
0	0	0	1	0	0	1	ST_0
3	1	0	0	0	1	1	
3	1	0	0	1	1	1	ST_1
4	0	0	0	1	x	1	
2	x	1	0	0	1	0	ST_2
5	x	1	0	0	0	1	
1	1	1	x	1	0	0	ST_3
4	0	1	0	1	x	1	

A wider input key is required with a larger TCAM. To function in a single RAM, the TCAM requires a larger input key. This key can be used to access the RAM location.

Table.IV Representing bits of TCAM in RAM

The value of the ternary bit	The value stored at	
	RAM[0]	RAM[1]
0	1	0
1	0	1
don't care	1	1

TCAM RAM bits are shown in Table IV. A 1 in RAM [0] corresponds to TCAM table number 0, while a 0 in RAM [1] performs the inverse. According to the 1 in the TCAM database, a 0 and a 1 are kept in RAM. A 1 is stored in RAM's [0] and [1] locations for the TCAM table's x (don't care) entry.

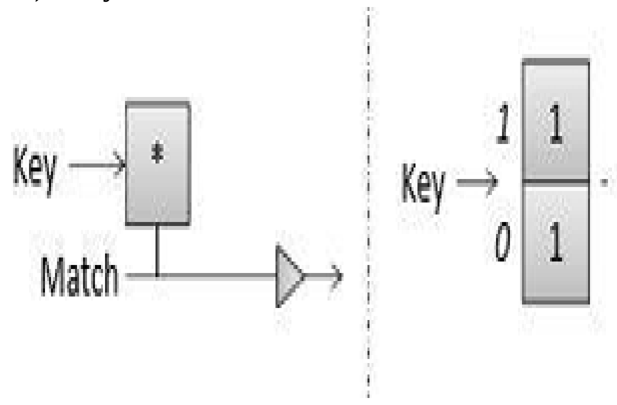


Fig.6 Mapping one bit in RAM

Figure 6 shows a 1 bit key and a 1*1 TCAM. A * indicates the "don't care bit" button. Memory addresses 0 and 1 can be mapped to the same value, 1 (see Table 5), allowing "1" to be stored in both locations.

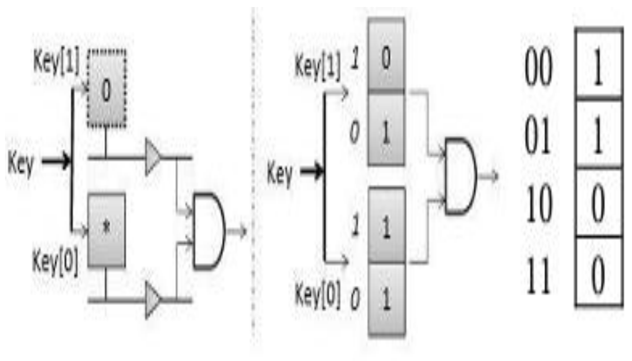


Fig.7 Memory Mapping of two bits

Figure 7 depicts two keys in use with a 1x2 RAM. The unlock code is '0' and the careless bit (*). Each of these keys corresponds to a certain memory value, which is mapped via AND. Figure 8 depicts the transfer of Table 3 numbers to SRAM. Elements b1 and b3 make up the mapping base.

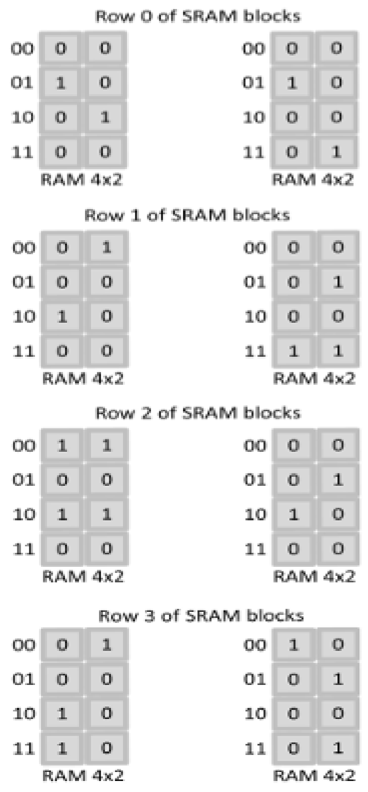


Fig.8 Mapping TCAM sub-tables to SRAM {b1,

b3}

The b0 and b2 bits in Table 4 are mapped to the first 24 RAM cells in each column. Bits b4 and b5 of each subtable are assigned to create the extra RAM. The incoming data is compared to the data in the SRAMs,

4. RESULTS AND PERFORMANCE EVALUATION

The output of a 6*6 EE-TCAM is shown in Figure 9. Pre-classification is used to find the position of previously stored words. A bitwise AND operation is then used to acquire the address that meets the criteria. TCAM table (mem) and partitioned table (mem1-mem4) are displayed. From mat1 to mat8, the variety of available twin port SRAM blocks is shown. The data term to be located is sent into the pre-classifier. The address decoder output from the pre-classifier was transmitted to the SRAM blocks. The search was then launched. The priority encoder is responsible for locating the correct memory address for a given word.

The priority encoder selects one of two addresses for a given memory region that contains the requested word. This is due to the splitter deciding which of the rows' priority encoders to employ. The expander's output is used to calculate the final output (fin_out).

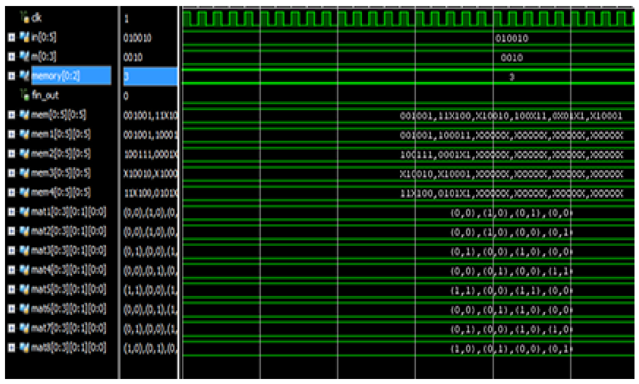

clk 1
in[0:5] 010010
m[0:3] 0010
memory[0:2] 3
fin_out 0
mem[0:5][0:5] 001001,11X100,X10010,100X11,000X1,X10001
mem1[0:5][0:5] 001001,100011,XXXXXX,XXXXXX,XXXXXX,XXXXXX
mem2[0:5][0:5] 100111,0001X1,XXXXXX,XXXXXX,XXXXXX,XXXXXX
mem3[0:5][0:5] X10010,X10001,XXXXXX,XXXXXX,XXXXXX,XXXXXX
mem4[0:5][0:5] 11X100,0101X1,XXXXXX,XXXXXX,XXXXXX,XXXXXX
mat1[0:3][0:1][0:0] (0,0),(1,0),(0,1),(0,0)
mat2[0:3][0:1][0:0] (0,0),(1,0),(0,0),(0,1)
mat3[0:3][0:1][0:0] (0,1),(0,0),(1,0),(0,0)
mat4[0:3][0:1][0:0] (0,0),(0,1),(0,0),(1,1)
mat5[0:3][0:1][0:0] (1,1),(0,0),(1,1),(0,0)
mat6[0:3][0:1][0:0] (0,0),(0,1),(1,0),(0,0)
mat7[0:3][0:1][0:0] (0,1),(0,0),(1,0),(1,0)
mat8[0:3][0:1][0:0] (1,0),(0,1),(0,0),(0,1)

Fig.9 Final output of EE-TCAM

The pre-classifier is given the input 010010 in this task. The pre-classifier produces the value 0100. According to research, the input number 010010 is situated in memory slot 3's address space zero.

Table. V Zynq-7000 FPGA resource utilization of 6*6EE-TCAM

Resource	Utilization	Utilization %
LUT	440	0.20
LUTRAM	24	0.03
Flip Flop	237	0.05
IO	6	2.40

To apply the suggested EE-TCAM plans in Zybo-7000 and calculate power, area, and delay, Vivado HLS was utilized. The FPGA resource use factors for the 6*6 EE-TCAM design are shown in Table V. The slice LUT, LUTRAM, Flip Flop, and IO settings are among these options.

Table VI shows the output lag as well as the total power obtained after execution. The power consumption is quite low when compared to other SRAM-based TCAM systems already in use (see Table I for details).

Table. VI Power and Delay

Total Power (mW)	1.03
Dynamic Power (mW)	0.80
Static Power (mW)	0.231
Delay (ns)	4.237

The delay is also reduced when compared to other variations. The system is faster than the TCAMs in Table I, with a speed of 268 megahertz (MHz).

5. CONCLUSION

Superior TCAMs are available in network switches. Even when the routing table grows in size, the demand for fast network speeds can be met. TCAMs based on SRAM are becoming more widespread, however the most majority are inefficient and wasteful. The Energy Efficient TCAM used far less energy than its SRAM-based competitors. Unlike traditional systems, EE-TCAM's pre-classification design only uses one row from each SRAM block during the search process. In this paper, we build a 6*6 TCAM using a Zybo-7000 outfitted with Vivado HLS software and Verilog HDL. The first stage completes memory mapping, builds a pre-classifier, and divides the TCAM table. After the full system has been constructed, the power, latency, and resource use may be calculated. A virtual enclosure may be included in the system's design to further optimize space.

REFERENCES

1. K. Pagiamtzis and A. Sheikholeslami, Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey, IEEE Journal of Solid State Circuits, vol. 41, no. 3, pp.712-727, March 2006.
2. Zahid Ullah, Manish Kumar Jaiswal, Y.C. Chan,and Ray C.C. Cheung, "FPGA Implementation of SRAM-based Ternary Content Addressable Memory", 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum,2012
3. Ullah Zahid, "SRAM-Based Ternary Content Addressable Memory", Doctor of Philosophy City University of Hong Kong, August 2014 .
4. Mahoney, P., Savaria. Y., Bois, G., and Plante, P., "Parallel Hashing Memories: an Alternative to Content Addressable Memories", 3rd International IEEE NEWCAS Conference, June 2005.
5. Madian Somasundaram, "Circuit to Generate a Sequential Index for an Input Number in a Pre-defined List of Numbers", US Patent, No. US 7155563 B1, December 2006.
6. Sangyeun Cho, Martin, J. R, Ruibin Xu, Hammoud, M. H, and Melhem, R, "CA - RAM: A High-Performance Memory Substrate for Search-Intensive Applications", proceedings of Performance Analysis of Systems Software, ISPASS, IEEE International Symposium, April, 2007.
7. Ullah, Zahid. "LH-CAM: Logic-based higher performance binary CAM architecture on FPGA." Embedded Systems Letters 9.2,2017
8. Locke, K. Parameterizable content-addressable memory. In Xilinx Application Note XAPP1151; Xilinx: San Jose, CA, USA, 2011.
9. Zahid Ullah and Sanghyeon Baeg, "Vertically Partitioned SRAM-based Ternary Content Addressable Memory," Elsevier Journal, Procedia Engineering , 2011
10. Ullah, Z. Ilgon, K. Baeg, "Hybrid partitioned SRAM-based ternary content addressable memory." IEEE Trans. Circuits Syst. I Regul., 2012