An Efficient TCAM Design using Multicascading Technique

Kevin Francis[1], **Nilima S. Warade**[2]

[1]Student, Dept. of Electronics and Telecommunication Engineering, AISSMS Institute of Information Technology, Maharashtra, India
[2]Professor, Dept. of Electronics and Telecommunication Engineering, AISSMS Institute of Information Technology, Maharashtra, India

---***---

Abstract - *Ternary content-addressable memories are an essential part of network routers. The space requirements of TCAM applications are increasing every day. Current solutions of TCAMs are affected by inefficiency in the storage space. The multicascading technique used for SRAM in this design achieves an effective storage use. Existing SRAM designs diminish the effect of the addition in the conventional TCAM pattern width from an sharp increase in memory utilization to a gradual one by using a cascaded configuration of block RAMs (BRAMs). But the BRAMs on the even the most advanced FPGAs have a limit in terms of minimum depth, which in turn affects the storage efficiency for the TCAM bits. Our proposed design circumvents this limit by mapping the divisions in the conventional TCAM table to sub blocks, which are not very deep. of the configured BRAMs, thus attaining an efficient memory design. The proposed design uses the configured two ported BRAMs of the design as a multiple port memory using the unique technique called as multicascading. This technique is implemented by clocking the block with a higher internal clock which is a multiple of the system clock, to access the various sub blocks of the block RAM in a single system cycle. Our design implemented in Xilinx ISE achieves better memory utilization, lower delay as well as lower power consumption with the increase in memory implementation size.*

Key Words: Field-programmable gate array (FPGA), memory architecture, static random-access memory (SRAM)-based ternary content-addressable memory (TCAM)

1. INTRODUCTION

Ternary content addressable memory (TCAM) collates an input word parallelly with the data stored in the entire memory, and presents the result as the address of the word which was matched. TCAM accumulates the data in three states instead of the standard two: 0, 1, and X (which is the "don't care" state). Conventional TCAMs are constructed by default in application specific integrated circuits (ASICs), and these provide fast search performance in a deterministic period. TCAM is commonly used in the design of high performance search algorithms and it has several applications in various fields such as networking, AI, information compression, signal detection, pattern recognition, image processing, and to increase the search performance of various primitives in a given database [1], [2]. Many IoT and big data analysis devices use TCAM in the form of a filter to store signature patterns, and also ti accomplish a significant increase in power efficiency by minimizing the transmission of faulty data wirelessly to cloud servers [3], [4]. FPGAs use SRAM to deploy TCAM by putting the content of the TCAM table into the SRAM block. Each specific TCAM pattern is represented by a single SRAM word, and the SRAM block stores the content of the entire data of the TCAM table. The increasing number of TCAM bits leads to a sharp increase in memory usage. The design mitigates this increase in memory usage by cascading many SRAM blocks.

A summary of the main contributions of this paper is presented in the following.

1) A unique technique called multicascading which results in more efficient utilization of memory.
2) The proposed design presents a TCAM that is modular and highly scalable.
3) The reduction in routing due to using a smaller amount of BRAMs leads to the design being very practical for high storage capacity. The unique optimization methods of cascading the blocks of SRAM in the design reduces the general complexity of the AND operations.

The proposed design is implemented using Xilinx ISE. A comparison of the proposed design and current designs is provided with respect to memory usage and other parameters. The proposed design results in up to 2 times better memory utilization.

2. MATERIALS AND METHODS

In recent times, memory is one of the most important features for the storage and retrieval of data. However, a large portion of device is used for data transmission memory in recent devices [5], [6], therefore TCAMs are used to mitigate this issue. In the current types of TCAM design have a limitation in terms of the minimum depth of the block RAM, which severely reduces the storage efficiency of the ternary CAM bits. Our proposed design will mitigate this issue by mapping the current TCAM design method to an SRAM with the unique technique of multicascading, which configures or arranges a number of sub blocks in one TCAM with the simple two port configuration. Here, this paper will present a multicascaded two port SRAM design in different limitations such as Case I = 512x28 (N=4), Case II = 512x32 (N=2), Case III = 1024x140 (N=4), Case IV = 2048x280 (N=4). This work will be implemented in Xilinx ISE using Verilog HDL and theoretically prove the better performance in terms of area, delay and power.

Table -1: List of used notations.

Notation	Description
D	Conventional TCAM Depth
W	Conventional TCAM Width
S_D	Configured Memory Depth
S_W	Configured Memory Width
$Log_2(S_D)$	Memory Block Address Bits
N	Sub Blocks in Memory Block/ Multicascading factor
S_D/N	Sub Block Depth
P	TCAM Memory Rows
Q	TCAM Memory Columns

2.1 Multicascaded N Port SRAM

This unique technique called multicascading is used to multiply or increase the ports of a two port SRAM block by a factor N. This is carried out by multiplying the internal clock of the SRAM block to the same factor N of the external clock [7], [8], [9], [10]. As shown in Figure 1 below, the two sides of the SRAM block are using a N to 1 multiplexer to address N read and write lines. The two lines are given access to the two port SRAM block in a circular method using the mod N counter bits.

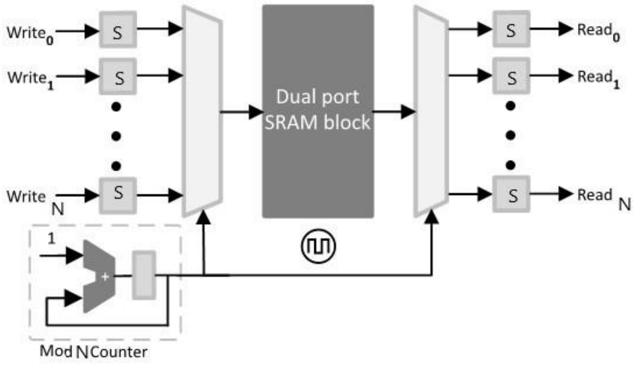


Fig -1: Multicascaded N port memory: the SRAM block is clocked at an N multiple of system clock, enabling N access during a single external clock cycle.

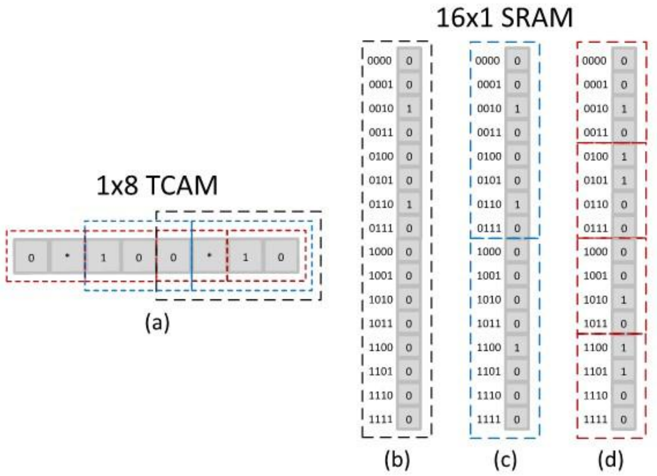
2.2 Fundamental Idea

In this proposed design, the width of the memory is decided by how deep the conventional TCAM table is and hence the width of the conventional table is coded as the address of the memory block.

Figure 2 shows the basic concept of the proposed design which achieves an increase in memory efficiency. Figure 2(a) depicts a D = 1, W = 8 conventional TCAM table, and Figure 2(b) depicts the implementation of the four TCAM bits by utilizing a D = 16, W = 1 memory block. Figure 2(c) depicts the implementation of six bits by using a D = 16, W = 1 memory block, which has been multicascaded twice, each memory sub block having a size of Q = 8, P = 1 which emulates three bits. Figure 2(d) depicts the implementation of eight bits by using a D = 16, W = 1 memory block, which has been multicascaded quadruple times, with each memory sub block having a of size Q = 4, P = 1 which emulates two bits. Thus, higher memory utilization efficiency (lesser physical memory bits are used for every TCAM bit) is achieved by using the unique technique of multicascading as depicted in Figure 2(c) and 2(d) as compared to the design which does not utilize multicascading as depicted in Figure 2(b). Thus, the storage capacity of the memory block for the TCAM bits increases with the technique of multicascading.

A N ported memory block of size $S_D \times S_W$ with a multicascading factor of N implements a conventional table of size $N\log_2(S_D/N) \times S_W$, each memory sub block of size $(S_D/N) \times S_W$ which emulates $\log_2(S_D/N) \times S_W$ table data, as depicted in Figure 3 and 1. Our proposed design achieves an increase in the table bits packing capacity with an rise in the multicascading factor N.

N.



0000	0	0000	0	0000	0
0001	0	0001	0	0001	0
0010	1	0010	1	0010	1
0011	0	0011	0	0011	0
0100	0	0100	0	0100	1
0101	0	0101	0	0101	1
0110	1	0110	1	0110	0
0111	0	0111	0	0111	0
1000	0	1000	0	1000	0
1001	0	1001	0	1001	0
1010	0	1010	0	1010	1
1011	0	1011	0	1011	0
1100	0	1100	1	1100	1
1101	0	1101	0	1101	1
1110	0	1110	0	1110	0
1111	0	1111	0	1111	0

Fig -2: (a) A conventional TCAM of P= 1, Q = 8; (b) D = 16, W= 1 memory block without multicascading technique emulating P=1, Q= 4 table; (c)D= 16, W = 1 memory block with a multicascading factor of N = 2 emulating P = 1, Q = 6 table; (d)D = 16, W= 1 memory block with a multicascading factor of N = 4 emulating P= 1, Q= 8 table.

2.3 Proposed Division of Conventional Table

The conventional TCAM table which is sized D x W is divided into P x Q partitions, such that every partition has N parts of $\log_2(S_D/N)$ x S_W size as shown in Figure 3. Our proposed design uses its configured memory blocks of $S_D \times S_W$ size as N ported SRAM, having N sub blocks of size $(S_D/N) \times S_W$ as depicted in Figure 3.

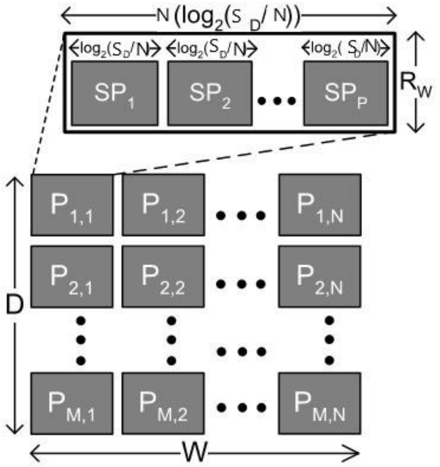


Fig -3: Proposed division of the conventional TCAM table.

Every sub block of the memory stores log2(SD/N) × SW size divisions of the conventional TCAM. Accordingly, the N sub blocks of the N ported memory in our proposed design stores a conventional TCAM division of size Nlog2(SD/N) × SW as depicted in the Figures 3 and 1. Similarly, the P × Q divisions of size Nlog2(SD/N) × SW are mapped to the memory blocks of the P × Q TCAM memory units in the proposed design, as depicted in Figures 3 and 2.

2.4 Basic Architecture of Proposed Memory Design

The basic architecture of our proposed memory design is represented in Figure 4. The memory is connected by two clock signals - Sck that is the system clock, and Nck that is the internal clock of the memory block, both of which are synchronized. Nck is configured such that it is N times faster than Sck. The system clock Sck is used when an inbound TCAM bit is written in the shift register of bit length Y. The $\log_2 N$ counter shown in Figure 1 generates a $\log_2 N$-bit number sequence within N cycles. This counter increases over the N clocks and is set to zero when it is reset. The $\log_2 N$-bits produced by the counter are joined at the end with the $\log_2(S_D/N)$ bits from the shift register. This sequence of bits makes the $\log_2 S_D$-bit address space of the physical memory.

During the positive edge of Nck, the $\log_2 N$ counter bits points to the beginning of the matching sub block and the shift register's lower $\log_2(S_D/N)$ bits chooses the word in the sub block. In the Nck cycle, the memory words are ANDed and then accumulated in SY-bit register. The word lookup for the Y-bit input word is accomplished in N clock cycles by first reading it via the input line, then ANDing and subsequently accumulating the memory words from every memory sub block. Therefore, the N words in memory which have been ANDed and accumulated are presented as the matched word using Sck.

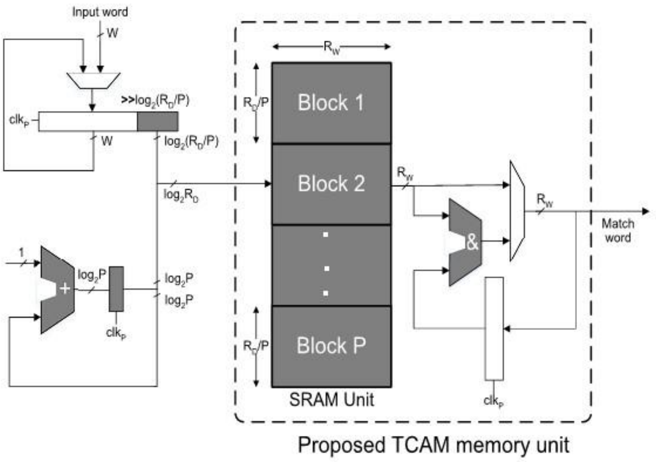


Fig -4: Basic architecture of the proposed memory design.

2.5 Modular Memory Design

The design of high storage capacity memory designs is realized by P x Q cascading units as depicted in Figure 5. The input W-bit memory word is split into P sub words consisting of $N\log_2(S_D/N)$ bits with the range of the bits represented in Figure 5. During the Sck cycle, the resulting sub words are kept in the P shift registers which are of size $N\log_2(S_D/N)$ bits. The N shift registers consisting of $\log_2 S_D$ directories are used for parallely matching P memory units of the Q columns of the design using Nck cycle. The matched SY-bit words from every row of memory units are ANDed bit wise on Sck. Subsequently, the result of the ANDing operation are given to the accompanying priority encoder. Thus, the matched $\log_2 P$ bit address and the resulting information from each encoder unit are given to the global priority encoder unit that performs the task of outputting the match address according to the priority. The proposed memory design catalogues the input word and outputs a matched word in the Sck cycle. The memory unit updates the word in physical memory parallelly. The proposed design has the update latency of S_D/N in the worst case scenario.

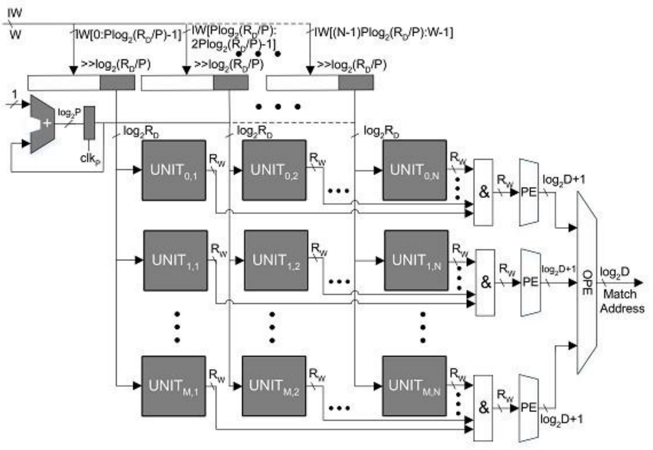


Fig -5: Grouping of the proposed high capacity memory units: (IW: input word, PE: priority encoder, GPE: global priority encoder).

3. RESULTS AND DISCUSSION

3.1 Memory Utilization

The proposed design realizes a conventional TCAM table of depth D and width W by cascading the memory blocks of $S_D \times S_W$ size. The usage of memory slices is noted in Table 2. As we observe, the utilization of memory is reduced with high sizes of implementation by using a higher N factor.

3.2 Delay

The delay of the search operation is improved even with the increase in the size of memory. This reduction is due to the unique technique of multicascading, which multiplies the internal clock by N to achieve higher system clock.

3.3 Efficiency

The power required by higher memory size is also kept in check and does not increase linearly or exponentially. This is due to the efficient utilization of shift registers and counters in the design.

Table -2: Comparison of various D x W sizes with different multicascade factors N.

	Case I 512x28 (N=4)	Case II 512x32 (N=2)	Case III 1024x140 (N=4)	Case IV 2048x280 (N=4)
No. of Slice Register	388	232	1308	2450
No. of Slice LUT'S	162	101	330	560
No. of Occupied Slices	139	81	408	779
Delay (ns)	3.187	3.264	3.300	3.457
Power (mW)	3.309	3.301	3.322	3.370
Frequency (MHz)	128	256	256	512

4. CONCLUSIONS

The current implementations of TCAM designs have inefficient utilization of memory and correspondingly have higher delay and power consumption with the increase in memory size. The proposed design operates the system clock at a higher frequency of the internal clock and thus allows accessing the entire content of the memory blocks within a single system clock. This results in lower delay in accessing the desired content and lower power consumption while accessing the same, even for higher memory design sizes.

REFERENCES

[1] N. Mohan, W. Fung, D. Wright, and M. Sachdev, “Design techniques and test methodology for low-power TCAMs,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 6, pp. 573– 586, 2006.

[2] B. Agrawal and T. Sherwood, “Ternary CAM power and delay model: Extensions and uses,” IEEE transactions on very large scale integration (VLSI) systems, vol. 16, no. 5, pp. 554–564, 2008.

[3] M.-F. Chang, C.-C. Lin, A. Lee, Y.-N. Chiang, C.-C. Kuo, G.-H. Yang, H.-J. Tsai, T.-F. Chen, and S.-S. Sheu, “A 3t1r nonvolatile tcam using mlc reram for frequent-off instant-on filters in iot and big-data processing,” IEEE Journal of Solid-State Circuits, 2017.

[4] V. C. Ravikumar, R. N. Mahapatra and Laxmi Narayan Bhuyan, "EaseCAM: an energy and storage efficient TCAM-based router architecture for IP lookup," in IEEE Transactions on Computers, vol. 54, no. 5, pp. 521-533, May 2005.

[5] Kai Zheng, Chengchen Hu, Hongbin Liu and Bin Liu, "An ultra high throughput and power efficient TCAM-based IP lookup engine," IEEE INFOCOM 2004, Hong Kong, pp. 1984-1994 vol.3, 2004.

[6] E. Spitznagel, D. Taylor and J. Turner, "Packet classification using extended TCAMs," 11th IEEE International Conference on Network Protocols, 2003. Proceedings., Atlanta, GA, USA, pp. 120-131, 2003.

[7] N. Manjikian, “Design issues for prototype implementation of a pipelined superscalar processor in programmable logic,” in Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on, vol. 1. IEEE, pp. 155–158, 2003.

[8] C. E. LaForest and J. G. Steffan, “Efficient multi-ported memories for FPGAs,” in Proceedings of the 18th annual ACM/SIGDA int. symposium on Field programmable gate arrays. ACM, pp. 41–50, 2010.

[9] C. E. LaForest, “Multi-Ported Memories for FPGAs,”, http://fpgacpu.ca/multiport/index.html.

[10] H. E. Yantir, S. Bayar, and A. Yurdakul, “Efficient implementations of multi-pumped multi-port register files in FPGAs,” in Digital System Design (DSD), 2013 Euromicro Conf. on. IEEE, pp. 185–192, 2013.