TCAM IMPLEMENTATION WITH HYBRID PARTITION AS Z- TCAM

M.Siva Lakshmi Reddy, M.Tech DECS, Department of ECE, JNTUA Anantapur.

Abstract: Ternary content addressable memory (TCAM) is a memory with some special characteristics. TCAM performs high speed parallel search operations and the operation done in single clock cycle. But TCAM having some limitations as compared with SRAM, which are low storage density, circuit complexity and slow access time. So, further we can move to TCAM with hybrid partition, as Z-TCAM. This paper proposes TCAM functionality with SRAM. Here hybrid partition of stored data in memory blocks is more important. Hybrid partition is main reason of shrinking the size of the memory and latency time. The language used for verifying proposed implementation is Verilog /VHDL.

Keywords: SRAM , TCAM , parallel search operation , Z-TCAM , hybrid partition , latency time.

1. INTRODUCTION

Content addressable memory (CAM) is one type of memory. The process of searching in CAM is reverse to that of RAM. In CAM, the input key is given as content rather than address and output return as address of the location. After that we can retrieve the required data from this address. Where as in RAM the input as address of location and output as required data. CAM's are of two types. One, binary CAM as Bi-CAM and simply called as CAM. And other one is ternary content addressable memory. In CAM, we can use search key and stored data is in the form of 0's and 1's. So, CAM's are accessible only to the exact match of search key. The process is repeat up to the matching whenever mismatch is occurring. Because of this latency time, power increased. So, we can move to ternary content addressable memory (TCAM). In this, the stored data and search key are in the form of 0's, 1's and X's. Here X is known as mask or don't care. It is used as either 0 or 1. So, TCAM is accessible for partially matched search key. Because of this latency time is reduced. But size of the memory is not changed. As compared with SRAM storage density and access time are not efficient. So further we can move to TCAM with hybrid partition. In this paper, hybrid partition is useful to reduce the size of the memory, searching time, power. The constant time search of TCAM makes it a suitable in different applications such as network routers, data compression and IP filter. It has been successfully implemented on Xilinx vertex-7 FPGA and Xilinx X-power analyzer by Verilog /VHDL code.

2. RELATED WORK

We can understand briefly about CAM's in this section. The method proposed in [3] using hashing technique to implement CAM functionality. But this technique has drawback of bucket overflow i.e. the operation may not finished up to the many operations are done. The number of stored bits in a word has a great impact on its performance. The method in [4] also uses hash technique to emulate TCAM functionality with SRAM. Because of hash function, it is suffers with overflow. The method in [6] is TCAM with SRAM and some additional circuitry for comparison. So circuit complexity is increased. And the size of the memory is increased by increasing number of bits i.e. exponentially (2^{nob}). The size of the memory is depends on 2^{nob} and searching time also increased.

The method in [1] is TCAM with hybrid partition. Because of hybrid partition the size of the memory and access time are shrinking. So, it is appropriate method for TCAM functionality.

3. HYBRID PARTITION OF CONVENTIONAL TCAM TABLE

The vertical and horizontal divisions are combined as called as hybrid partition. We can divide stored data of TCAM table as different partitions. This process is depicted in table 1. Each partition of bits referred as sub words. These sub words are stored in memory blocks. Vertical partition: implies that a TCAM word of 'L' bits is partitioned into 'P' sub words. Each sub word of 'b' bits. Vertical partition is used in Z-TCAM to decrease the size of the memory as much as possible. Horizontal partition: It is used in creating the layers. Because of this, we can reduce the searching time. Why because, all the layers are searched parallel. If we create more layers, circuit requirement is increasing. So, implementation is expensive and cost.

The integration of vertical and horizontal partition provides a very good solution to make hybrid partition as to reduce size of the memory and power consumption.

CONVENTIONAL TCAM TABLE AND ITS HYBRID PARTITION

Address	Subword1 (sw1)	Subword2 (sw2)	Layer
0	00	0X	
	HP11	HP12	1
1	10	00	
2	0X	11	
	HP21	HP22	2
3	01	1X	

TABLE I

4. PROPOSED SYSTEM ARCHITECTURE

I. ARCHITECTURE OF Z-TCAM:

The overall architecture of proposed system is illustrated in figure 1. In this each layer associated with architecture shown in figure 2. It is consider for only two layers. It has two layers and CAM priority encoder (CPE). In this all sub words are fed to the every layer of Z-TCAM. The output of each layer is written as potential match address (PMA). The PMA's are input to the CPE, which selects matched address (MA) from different PMA's.

II. ARCHITECTURE OF LAYER

Architecture of layer is depicted in figure 2. The architecture contains validation memories (VM's), single bit AND operation, original address table address memories (OATAM's), original address tables (OAT's), 2-bit AND operation and layer priority encoder (LPE).

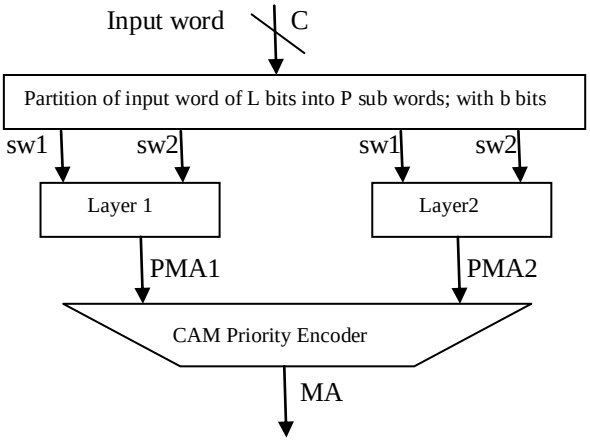

Fig.1. Architecture of Z-TCAM

VALIDATION MEMORY

Validation memory size is 2^b x 1bits. Here b is associated with the number of bits in each sub word and 2^b indicates the number of rows in TCAM table. Here 2^b combinations are available where each combination represents a sub word. For example, if 'b' is of 2 bits, then $2^2 = 4$ combinations are available. Same thing also related to OATAM and OAT. If the sub word present in VM, the output is set to high. Otherwise set to low. For example table 2 shows the sub words 00 and 01 are mapped in VM11 and that location of memories are high. Remaining locations are set to low.

SINGLE BIT AND OPERATION

The outputs of all VM's are fed to AND. Then the single bit AND operation output is presiding whether searching process is sustaining or not. If the output of AND operation is one (1), then the process will be continued. Otherwise the input is mismatch with stored data and the process will be stopped.

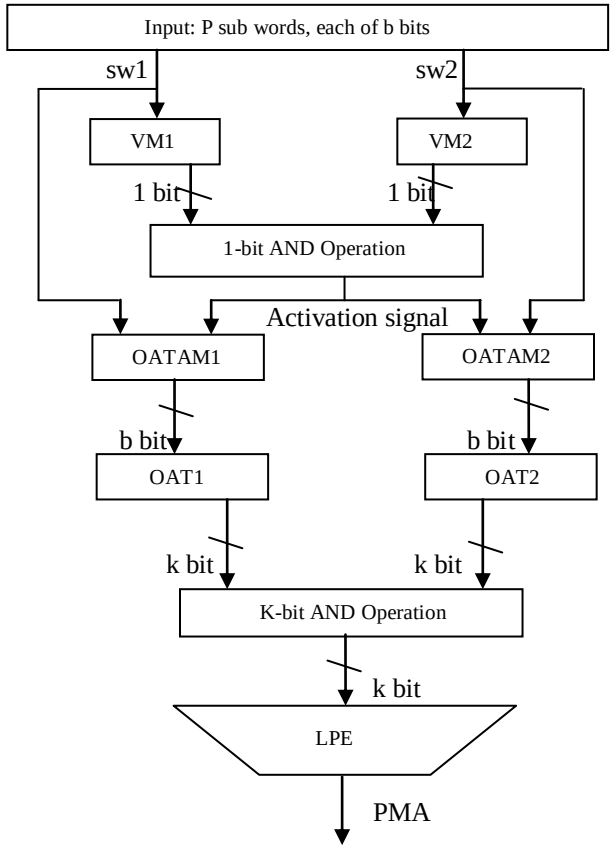

Fig.2. Architecture of layer of Z-TCAM

ORIGINAL ADDRESS TABLE ADDRESS MEMORY

The size of the OATAM is 2^b x b. where 2^b represents number of rows and each row contains 'b' bits. The OATAM is used for the stored data in certain memory location indexed with sub word. If the sub word is mapped in VM, that sub word is passed to the respective OATAM.

ORIGINAL ADDRESS TABLE

The size of OAT is 2^b x k. where 'b' is no. of bits in each sub word and 'k' represents bits per row. Here each bit indicated as original address. The output of OATAM as index is fed to the OAT. The original address of is stored memory location of OAT and that memory locations are accessed by index of OATAM.

2-BIT AND OPERATION

The outputs comes from all OAT's are connected to AND gate and all the bits ANDed bit by bit. The out of and operation is also original address. Here each bit indicates original address.

LAYER PRIORITY ENCODER

Here different matches may occur; because of we are using ternary data to emulate TCAM functionality. So, LPE decides which is best match to the given search key from the outputs of 2- bit AND operation.

5. OPERATION OF Z-TCAM

I. MAPPING OF DATA

The conventional TCAM table is divided as hybrid partitions. Each partition is expanded into binary form. So, first expand the mask bits as 0's or 1's before stored in memories. For example, we have ternary data of 00x1 is in partition and then we expand as 0001 and 0011. Then stored into SRAM. Each sub word of hybrid partition acts as an address to the memories and access a certain location of memory in associated memory blocks (VM, OATAM and OAT).

A sub word in a hybrid partition may be present at multiple locations. So, it is mapped in its respective VM and its original address(es) is/are mapped to its/their corresponding bit(s) in its corresponding OAT. Here each bit in OAT indicates an original address, only those memory locations are mapped 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 mapping of data is shown in Table II. We use Table I to be mapped to Z-TCAM. We take P = 2, L = 2, N = 2, and b = 2. After necessary processing, HP11, HP12, HP21, and HP22 are mapped to their respective memory units. In the example, we map hybrid partitions of layer 1 to their respective memory units. Hybrid partitions of layer 2 can be easily mapped in same way.

DATA MAPPING OF LAYER 1 OF Z-TCAM

Address	VM 11 12	OATAM 11 12	Original address 11 12 0 1 0 1
0	1 1	0 0	1 0 1 1
1	1 1	- 1	0 1 1 0
2	0 0	- -	0 0 0 1
3	0 1	1 2	0 0 0 0

TABLE II

The algorithm1 for searching in a layer of ZTCAM is as follows:

INPUT: N sub-words

OUTPUT: PMA

1: Apply P 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 OATs parallel

7: b. AND bit-wise all N-bits rows

8: c. Select PMA/mismatch occurs

9**:** else

10: Mismatch occurs

11: end if

EXAMPLE OF SEARCH OPERATION IN LAYER1 OF Z-TCAM

Steps	Functionality
1	Sub word1 = 00 Sub word2 = 01
2	Sub word1 is fed to VM11 Sub word2 is fed to VM12
3	Sub word1 is validated , VM11 = 1 Sub word2 is validated , VM12 = 1
4	Check all the VM's are validated or not
5	Yes, operation is continued
6	Read out data from OATAM11 = 00 Read out data from OATAM11 = 01
7	The output of OAT11 = 10 The output of OAT12 = 10
8	Result of N-bit and operation is 10
9	PMA = 00(0)

TABLE III

II. SEARCHING OF DATA

SEARCHING OPERATION IN LAYER OF Z-TCAM

Algorithm 1 represents search operation in layer of Z-TCAM. N sub words are applied to Z- TCAM layer at a time. All sub words are read out respective VM's from those memory locations. If all the VM's are validated, the output of single bit AND operation is high. Then only the operation sustained. If any one of the VM is not validated, the output of single bit AND operation is low. Then the operation is stopped and again starts from first with new search key. If all VM's are validated, all the sub words are fed to OATAM and the output is indexes of respective addresses. All indexes are used to read out the original address of the corresponding

OAT's. In OAT outputs, each bit represents original address. The output of OAT containing N- bits. All the OAT outputs are ANDed bitwise. And then LPE selects PMA among the output of N-bit AND operation. Example of searching operation in layer1 is depicted in table III. And table II also to be used for searching. In this searching process, mismatch is occurred in two conditions. They are

1. If any one of the VM is not validated, the output of single bit AND operation is low.
2. And if all the bits are low (none of the bits are high) after N-bit AND operation.

Z-TCAM SEARCHING

The operation of searching in Z-TCAM is done in all the layers at a time. This total process is depicted in algorithm2. The search key is applied to proposed Z-TCAM. It is divided into P-sub words. All the sub words are applied to all the layers in Z-TCAM. The outputs of the layers are potential match addresses (PMA). And the CPE gets matched address (MA) from the PMA's. Otherwise, the given search key is not matched. Table IV gives the information of searching process of Z-TCAM to the input key word as 0001.

The algorithm2 for searching in ZTCAM is as follows

Input: search key

Output: MA

1. Apply search key
2. Divide search key into P sub-words
3. All the layers use algorithm1 in parallel
4. Select MA among PMA's/ mismatch occurs

Step	Functionality
1	Input key = 0001
2	Sub word1 = 00 Sub word2 = 01
3	PMA1 = 0(00) PMA2 = ZZ
4	CPE selects as MA = 0

TABLE IV

6. SIMULATION AND IMPLEMENTATION RESULTS

We have implemented Z-TCAM on Xilinx Virtex-7 FPGAs and xpower analyzer. Power and delay of proposed system are shown in table V. and comparison also.

Parameters	TCAM	Z-TCAM
Power(mW)	29.20	27.34
Delay(ns)	9.1	8.2

TABLE V

Simulation result of layer 1, layer 2 and Z-TCAM are shown in following figures 3, 4 and 5.

Fig.3. Simulation of layer1 of Z-TCAM

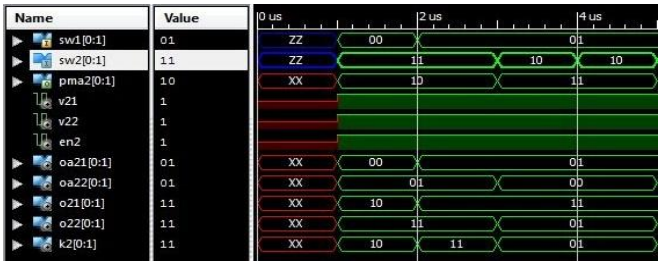

Fig.4.Simulation of layer2 of Z-TCAM

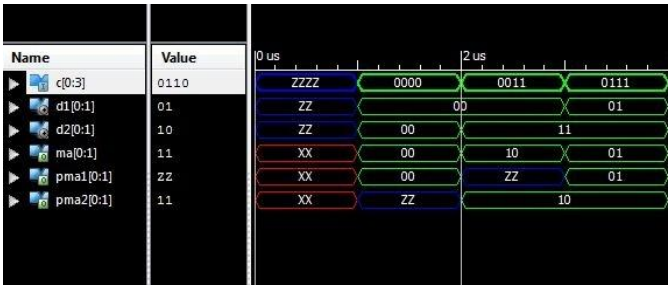

Fig.5. Simulation of overall implementation of Z-TCAM

7. CONCLUSION

The objective to design and implement a TCAM with hybrid partition as ZTCAM is achieved. We are using hybrid partition and its data mapping for better performance. Because of this, we achieved better delay, memory utilization and power consumption. The implementation is implemented on Xilinx vertex7 FPGA.

REFERENCES

1. Zahid Ullah, Manish K. Jaiswal, andRay C. C. Cheung, "Z-TCAM: An SRAM-based Architecture for TCAM," IEEE Transactions on Very Large Scale Integration (vlsi) Systems, vol. 23, no. 2, february 2015.

2. 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.

3. Z. Ullah, K. Ilgon, S. Baeg, Hybrid partitioned SRAM-based ternary content addressable memory. Circuits Syst. I **59**(12), 2969–2979 (2012).

4. Ullah, Zahid, Manish Kumar Jaiswal, and Ray CC Cheung. "E-tcam: An efficient sram- based architecture for tcam." *Circuits, Systems, and Signal Processing* 33.10 (2014): 3123-3144.

5. 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.

6. Xilinx, San Jose, CA, USA. *Xilinx Xpower Analyzer* [Online]. Available: http://www.xilinx.com

7. Somasundaram , Madian. "Memory and power efficient mechanism for fast table lookup." U.S.Patent No.7,162,572. 9 Jan. 2007.

8. 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.

9. L. Frontini, S. Shojaii, A. Stabile, V. Liberali, A new XOR-based content addressable memory architecture. In *19th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, (2012), pp. 701–704.