FPGA Based Architecture for High Performance SRAM Based TCAM for Search Operations

Lekshmipriya S.¹, Suby Varghese²

¹P G Scholar, VLSI and Embedded Systems, Department of ECE, T K M Institute of Technology, Kollam, India

²Assistant Professor, Department of ECE, T K M Institute of Technology, Kollam, India

Abstract: Ternary Content Addressable memory is a type of memory that allows the memory to be searched by content rather than by address. It performs high speed lookup operations within a single clock cycle. But when compared to RAM technology the conventional TCAM circuitry has certain limitations such as low access time, low storage capacity, circuit complexity and high cost. So we can use the benefits of SRAM by configuring it to behave like TCAM. The project focuses on a memory architecture based on the hybrid partitioning concept which emulates the TCAM (Ternary Content Addressable Memory) functionality with SRAM. The hybrid partitioned SRAM based TCAM logically dissects conventional TCAM table in a hybrid way (row-wise and column-wise) into TCAM sub tables, which are then processed to map on their corresponding memory units and match address is produced. The 64*32 hybrid partitioned SRAM based TCAM can be used in networking applications such as packet switching and packet classification in ATM communication systems.

Keywords: Field Programmable Gate Array (FPGA), Hybrid Partitioning (HP), memory architecture, Priority Encoder(PE), Static Random Access Memory (SRAM)-based TCAM, Ternary Content Addressable Memory (TCAM).

1. Introduction

Content Addressable Memory (CAM) is a type of memory that provides access to stored data by contents rather than by an address and outputs the match address. CAM compares the search key with all the stored words in parallel and returns the address of the best match. But the CAM can store only two values (0 and 1). Since the TCAM can store don't care state(x), which can be matched to both 0 and 1, causes multiple match and provides high speed lookup operations. It has wide range of applications network routers such as packet forwarding in ATM switches, network intrusion-detection systems etc. In network systems, TCAM is used to compare the destination address of incoming packet against the stored address and forward the packet to the corresponding output port. It also finds applications in real-time pattern matching in virus-detection, gene pattern matching and image processing.

Although CAM technology performs lookup operation within a single clock cycle, yet it has some shortcomings, when compared to RAM technology. The TCAM comparison circuitry introduces more area, power and complexity to the TCAM architecture due to the massive parallelism. The extra logic and capacitive loading increases the access time and make the TCAM testing a time consuming process. RAM is cheaper than CAM and CAM technology does not evolve as fast as RAM. The CAM has very limited pattern capacity and denser than RAM technology. Hence the TCAM is not subjected to the intense commercial competition found in the RAM market.

This work mainly emphasis on an alternative to TCAM, which is constructed from SRAM. The SRAM based TCAM is based on the hybrid partitioning concept, in which the conventional TCAM table is divided along vertically and horizontally to form TCAM sub-tables and each sub-table is mapped to its corresponding memory unit. The SRAM based TCAM emulates TCAM functionality with SRAM and can implement in Field Programmable Gate Array. The FPGA technology has become an attractive option for implementing real-time network processing engines due to its reconfigure ability and parallelism to achieve speed and high performance. So by utilizing SRAM and FPGA, the SRAM based TCAM on FPGA is a good choice in networking applications.

2. Related Works

Basically CAM is classified into two types—binary CAM (BiCAM) and ternary CAM (TCAM). BiCAM can store two logical states—0 and 1, whereas TCAM can store an extra state, don't care state (x), in addition to 0 and 1. To store x, an extra bit is required. The input search word is compared with the stored word on bit-by-bit basis and input word is compared with all the stored words in parallel, it makes CAM to provide a high-speed constant-time search. The priority encoder (PE) selects a single match in case of multiple matches in TCAM.

It found very limited work on RAM-based CAMs. RAMbased CAM uses hashing technique, with inborn disadvantages —collisions and bucket overflow. The number of stored elements has a great impact on its performance; with the increase in number of stored elements, the performance of the method becomes gracefully degradable. Furthermore, the method emulates BiCAM not the TCAM. The method in [8] also uses hashing technique to emulate the TCAM functionality with RAM. Being based on hashing technique, it also suffers from collisions and bucket overflow. The RAM-based CAM in [8] may have further limitations. First of all, the performance depends on the actual record distribution and how records are accessed. If many records have been placed in an overflow area due to collisions, a lookup may not finish until many buckets are examined and also causes duplication in multiple buckets that leads to an increased capacity requirement. On the other hand, if the search key contains don't care bits which are taken by the hash function, then multiple buckets must be accessed that results in performance degradation. The essence of TCAM is that it provides a deterministic search performance but hash-based CAMs do not provide deterministic performance because of potential collisions and Bucket overflow. The performance of the methods in [8] greatly depends on hash function. Hash function minimizes the probability of collision. SRAM-based pipelined CAMs also take multiple clock cycles to accomplish a search operation and the memory utilization is also not efficient.

U.S. patent [7] claims that a CAM of size up to 2^{W} words \times *w*-bit can be emulated with 2^{*w*} memory bits. If CAM is of size $2^{W} \times W$ bits, then it implies that the number of addresses and the total possible combinations of W bits are same. For example, for 4 bits word, the total combinations are $2^4 = 16$, and according to the patent, to store 2^4 CAM words, 2^4 addresses are required. Thus, 2^4 bits are needed to emulate the CAM functionally with RAM, where each TCAM word is mapped to its corresponding memory bit. For the method to be applicable, entries in TCAM table must be in ascending order and are then mapped to their corresponding memory bits. The approach in the patent uses a concept of last index (LI), which also takes (W+1) memory bits. The 2^{w} of memory bits is grouped into 2^{w-b} rows. Each row has two parts- Data part and LI part. The w-b most significant bits of the input word make an address, which selects a row in 2^{w-b} rows. The b low-order bits access a particular bit in the selected row. If the accessed bit is high, then the input word is present; otherwise, absent. HP-TCAM [3] discusses the same concept as in [7], which emulates the TCAM functionality with SRAM. The paper presents SRAM based TCAM based on Hybrid Partitioning concept (HP). tables where "n" represents vertical partitions and "m" represents horizontal partitions of each vertical partition.

'n' vertical partitions



Figure 1: Conceptual view of Hybrid partitioning (HP)

Figure 1 illustrates a conceptual view of HP. The Horizontal partitioning (HrP) part of HP divides the entire TCAM table into L layers by using the original address range of conventional TCAM tables. The function of vertical partitioning (VP) part of HP is to divide each of the layers into N vertical partitions (columns). In each layer, the input TCAM word is divided into N sub words, where each sub word is of w bits. Thus, HP results in a total of L X N hybrid partitions. The dimension of each hybrid partition is K X w, where K is a subset of original addresses and w is the number of bits in a sub word. All TCAM sub-tables are then processed to be stored in their corresponding SRAM memory block. As the number of bits in a TCAM word increases, the memory requirements increases exponentially. Hence the vertical partitioned TCAM table is further divided into hybrid partitions to decrease the memory requirements as much as possible.

3.2 Architecture of SRAM based TCAM

The hybrid partitioning concept implies that a TCAM table is divided into L layers (horizontal partitions) and each layer is divided into N vertical partitions. In each layer the input search word is divided into N sub words to generate the corresponding potential match address (PMA) and given to the CAM priority encoder to generate the Match address (MA). It performs LXN hybrid partitioning. The work focuses on 64X32 hybrid partitioning.

3. Methodology

3.1 Hybrid Partitioning

Hybrid partitioning is a concept, which divides the TCAM table along columns (vertical partitions) and rows (horizontal partitions) that results in TCAM sub-tables. Each TCAM sub table is termed as hybrid partition and the collective partitioning scheme (vertical and horizontal) is called HP (Hybrid Partitioning). Hybrid partitioning logically dissects the TCAM table into m X n number of TCAM sub

3.2.1 Overall Architecture

The memory architecture for SRAM based TCAM is depicted in Figure 2. The overall architecture consists of L layers and a CAM priority encoder. The input search word is

Volume 4 Issue 2, February 2015 <u>www.ijsr.net</u> Licensed Under Creative Commons Attribution CC BY

International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

given to all the L layers simultaneously. Each layer performs vertical partitioning to generate Potential match address (PMA) for the corresponding sub word. Then all the PMAs are fed to CAM priority encoder (CPE), which selects Match address (MA) among PMAs. The PMA is the Potential Match Address corresponds to each layer.

Each layer produced a PMA, which represents the address of input sub-word from each of the layers. All are fed to CAM priority Encoder (CPE). The CPE is similar to a conventional priority encoder used in CAM. It selects a Match Address from all the PMAs. Match Address (MA) represents the original address of the input search word. CAM is designed such that, it gives the highest priority data as the output. The lowest bit location has the highest The PMA is the Potential Match Address corresponds to each layer. Each layer produces a PMA, which represents the address of input subword from each of the layers. All are fed to CAM Priority Encoder (CPE). The CPE is similar to a conventional priority encoder used in CAM. It selects a Match Address from all the PMAs. Match Address (MA) represents the original address of the input search word. CAM is designed such that, it gives the highest priority data as the output. The lowest bit location has the highest priority. The PMAs are of 4 bit values, which represent a particular address location at which the input search word is stored. The MA is the final address consists of 4 bits, which represents the original address. The PMAs are of 4 bit values, which represent a particular address location at which the input search word is stored. The MA is the final address consists of 4 bits, which represents the original address corresponding VM, OATAM and OAT. These three constitute a hybrid partition in a layer. It needs SRAM units to map the corresponding sub words and their original addresses. The layer is further equipped with 1-bit AND operation, k-bit AND operation and layer priority encoder, which constitute additional logic.



Figure 2: Overall Architecture

3.2.2. Layer architecture

The layer architecture is shown in the Figure 3.Each layer performs vertical partitioning. The layer architecture consist of

- N validation memories (VMs)
- 1-bit AND operation
- N Original Address Table Address Memory (OATAMs)
- N Original Address Tables (OATs).
- K-bit AND operation

• Layer priority encoder (LPE)





Validation Memory (VMs)

Validation memory has a size of $2^{W} \times 1$ bit as shown in table 1, where w represents the number of bits in each sub word. Since w bits are given as the input to each of the validation memory and all VM s have a total combination of 2^{W} , where each combination represents a sub word. Each sub word acts as the address to the validation memory. The validation memory stores values that indicate the presence or absence of the input sub word in that validation memory. If the memory location invoked by the sub word is high, then it means that the sub word is present at that validation memory. If the memory location invoked by the sub word is set to low, then the sub word does not exist in that VM.

 Table 1: Validation memory

Address	VM1	VM2	VM32
0	0	1	0
1	1	0	0
2	0	1	0
			•
			26
15	0	0	1

Since a layer has N vertical partitions, to accommodate the data in each vertical partition, it required N VMs, N OATAMs, N OATs. Each hybrid partition in a layer has its

1-bit AND Operation

The outputs of all VMs are fed to 1-bit AND block. It ANDs all the VM outputs. The output of 1 bit AND operation decides the continuation of search operation. The result of the 1-bit AND operation is the activation signal. If the activation signal is high, then it allows the continuation of the search operation, otherwise mismatch occurs in the corresponding layer and the search operation will be aborted.

Original Address Table Address Memory (OATAMs)

The size of original Address Table address memory is $2^{w}x w$ bits, where 2^{w} is the number of rows in each of the OATAM and 'w' is the number of bits in each row. Each of the input

Volume 4 Issue 2, February 2015 www.ijsr.net

Paper ID: SUB151610

Licensed Under Creative Commons Attribution CC BY

sub word is also fed to the corresponding Original address table address memory. In OATAM an address is stored at the memory location which is indexed by the sub word and that address is used to invoke the original address from the corresponding

Original Address Table (OAT).

The output of the OATAM is called the Original Address Table Address (OATA), which is fed to the input of the OAT. If a sub word is present in a validation memory, then a corresponding address is also stored in OATAM at a memory location, which is accessed by the sub-word. The OATAM contains both address values and '-'(don't care value). The hyphen ('-') or don't care value indicates that no data is stored at the corresponding memory location because that sub-word is not present in the corresponding validation memory.

K-bit AND Operation

The output from all OATs is fed to K-bit AND block. It performs bit-by-bit AND operation the output of all OATs. Read out the K bit rows from all Original Address Table(OAT) by using their corresponding Original Address Table Address(OATA), then the K bit values from each OAT is ANDed and the result is then forwarded to Layer Priority Encoder(LPE).The LPE selects the match address from all the possible PMAs in the result of K bit AND operation.

Layer Priority Encoder

TCAM supports multiple matches. So multiple matches may also occur in a layer, which are then resolved by Layer Priority Encoder (LPE). The result from K bit AND is fed to the LPE. The LPE selects Potential Match Address (PMA) among the outputs of K bit AND operation. LPE selects the highest priority matching location as PMA.

3.3 Phases of SRAM based TCAM

can store an extra value called the don't care value, a hybrid partition may contain a x bit. Since the SRAM cannot store a x bit, the x bit is expand into binary bits 0 and 1.For example if the search word is '001x' then it can be expanded to '0011' and '0010'.These binary words are then stored in SRAM. Each subword act as an address and access a memory location in the corresponding memory blocks (VM,OATAM and OAT). Each sub-word is applied to its corresponding

VM and a logic '1' is written at that memory location and all other memory locations are set to '0'. The same sub-word is also applied to its corresponding Original address table address memory and it store addresses in the memory location, which is indexed by the input sub-word. During the search operation the 'w' bit address is act as the OATA. The k bit values are written at the memory location in OAT, which is invoked by the corresponding Original Address Table Address. Each sub-word in all hybrid partitions is mapped to its corresponding memory location in VM and original address of the sub-word are mapped to their corresponding OAT. The output of the OAT represents an original address.

3.3.2 Data Searching 3.3.2.1 Layer searching



Layer consists of N sub-words. All the N sub-words applied to a layer and read out their corresponding memory locations simultaneously. If output of all the VMs are validate then searching will continue, otherwise mismatch occurs in the layer. If all VMs are validate, then read out all the OATAs from their corresponding OATAMs concurrently to obtain the k-bit row value. Then Read out k-bit rows from their corresponding OATs simultaneously to obtain the address, which are then bit-wise ANDed. LPE selects the potential

3.3.2.2 Overall searching

The operation of SRAM based TCAM has been described in two phases.

match address from the output of k-bit AND operation.

3.3.1. Data Mapping

In hybrid partitioning, the conventional TCAM table is logically partitioned into TCAM sub-tables. Since the TCAM. Search operation performed concurrently in all layers in the SRAM based TCAM. First the search word is applied to SRAM based TCAM, which is then divided into N subwords and perform layer searching. Potential Match Addresses (PMAs) are obtained from all layers and are applied to CAM priority Encoder (CPE).CPE selects a match address (MA) among all PMAs. Otherwise a mismatch occurs.

4. Results and Discussion

The entire layer structure and overall memory architecture are modeled using VHDL in Xilinx ISE Design Suite 12.1 and the simulation of the design is performed using Modelsim SE 6.3 f to verify the functionality of the design. Each of the modules are designed using Structural and behavioral models.



Figure 5: RTL-Schematic of 64 X 32 SRAM based TCAM

🧇 /overallarchitecture/dk	-No Data-					
Ioverallarchitecture/rst 🧇	-No Data-					
📻	-No Data-	000000010010001101000101011100111	1000100110101011	1100110111	1011110000	00
🧇 /overallarchitecture/ma	-No Data-	3	(o			
🖅 🔶 /overallarchitecture/a0	-No Data-	0011	0000			
😐 🤣 /overallarchitecture/a1	-No Data-	0011	(0000	8		
🕞 🧄 /overallarchitecture/a2	-No Data-	0011	0000		<u> </u>	
🖅 🔶 /overallarchitecture/a3	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a4	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a5	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a6	-No Data-	0011	0000			
🖅 /overallarchitecture/a7	-No Data-	0011	0000			
🖅 /overallarchitecture/a8	-No Data-	0011	,0000			
庄 🍫 /overallarchitecture/a9	-No Data-	0011	,0000			
💽 🥠 /overallarchitecture/a10	-No Data-	0011	0000	8		
🖅 🔶 /overallarchitecture/a11	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a12	-No Data-	0011	0000			
💶 🧄 /overallarchitecture/a13	-No Data-	0011	0000			
📻	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a15	-No Data-	0011	0000			
庄 🧄 /overallarchitecture/a16	-No Data-	0011	20000			
😐 🤣 /overallarchitecture/a17	-No Data-	0011	0000			
庄 - 🔷 /overallarchitecture/a18	-No Data-	0011	0000	8		
💶 🧇 /overallarchitecture/a19	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a20	-No Data-	0011	0000			
🖅 🔶 /overallarchitecture/a21	-No Data-	0011	0000			
📻 - 🤣 /overallarchitecture/a22	-No Data-	0011	0000			
🖅 /overallarchitecture/a23	-No Data-	0011	0000	<u> </u>		
庄 🧄 /overallarchitecture/a24	-No Data-	0011	0000	8		
庄 - 🔷 /overallarchitecture/a25	-No Data-	0011	0000	8		
💽	-No Data-	0011	0000	8		

Figure 6: Simulation Result of 64 X 32 SRAM based TCAM

The layer architecture consists of validation memory, Original address table address memory, Original address table and layer priority encoder. The coding for the entire layer structure has completed and simulation result is obtained. First the 128 bit input word is applied to the layer structure. In each layer it is divided into 32 sub-words, each with 4 bits. 32 validation memories were designed to store the 32 sub-words, which generate activation signal to indicate whether a match or mismatch occur in that layer. Then OATAM is designed to store the address of the stored data and to generate an Original address table address (OATA). Since doing 64*32 Hybrid Partitioning, it consists of 64 layers and each layer has 32 divisions. Input to the SRAM based TCAM is 128 bit search word

5. Conclusion

The RAM technology is more attractive than CAM technology due to several reasons. So the SRAM can be configured to behave like TCAM. Here the TCAM functionality is emulated with SRAM. The memory architecture is based on the Hybrid Partitioning concept, which dissects the conventional TCAM table into rows and columns and then it is processed to be stored in their corresponding SRAM units. The memory architecture utilizes the benefits of RAM technology along with the feasibility of FPGA. The TCAM takes a 128 bit search word as input. The output is the Match Address(MA), which represents the original address at which the search word is stored. The search operation in TCAM is carried out in two steps. First the input search word is divided into sub-words and each sub-word in all the hybrid partitions maps to its corresponding memory locations in validation memory and original address of each sub-word is mapped to their corresponding Original Address Table (OAT). This is the data mapping phase and next is the searching phase. The validation memory (VMs), Original Address Table Address Memory (OATAMs), Original Address Table (OATs) are constructed from SRAM. The Layer Priority Encoder encodes (LPE) the input 16 bit into a 4 bit value called PMA. CPE selects a matched address from all the PMAs. The entire layer structure is coded using Xilinx 12.1 and is simulated in ModelSim 6.3f.Since the SRAM based TCAM supports large input bit patterns and it can be successfully implemented on FPGA, it is a favorable choice for networking applications. The conventional TCAMS in packet forwarding engines, pattern matching systems and the tag matching in caches can be replaced by the SRAM based TCAMs, which uses the benefits of RAM technology and feasibility of FPGA technology.

References

- [1] ZahidUllah, 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, february 19.
- [2] ZahidUllah and SanghyeonBaeg, "Vertically Partitioned SRAM-Based Ternary Content Addressable Memory" IACSIT International Journal of Engineering and Technology, Vol. 4, No. 6, December 2012.
- [3] ZahidUllah, Kim Ilgon, and SanghyeonBaeg,"Hybrid Partitioned SRAM-Based Ternary Content Addressable Memory", IEEE Transactions on circuits and systems 1: regular papers, vol. 59, no. 12, December 2012.
- [4] P. Mahoney, Y. Savaria, G. Bois, and P. Plante," Transactions on high performanceembedded architectures and compilers II", in Performance Characterization for the Implementation of Content Addressable Memories Based on Parallel Hashing Memories, P. Stenstrm, Ed. Berlin, Germany: Springer-Verlag, 2009, pp. 307325.
- [5] S.J. Ruan, C.-Y.Wu, and J.-Y.Hsieh,"Low power designof pre computation- based content-addressable memory", IEEE Trans. Very Large Scale Integr. (VLSI)Syst., vol. 16, no. 3, pp. 331335, Mar. 2008.
- [6] K.Pagiamtzis, A. Sheikholeslami," Content-addressable memory (CAM) circuits and architecture tutorial and survey", IEEE J. Solid- State Circuits, vol. 41, no. 3, pp.

[7] M. Somasundaram, "Memory and power efficient mechanism for fast table lookup," U.S. Patent 20 060 253

712727, Mar. 2006.

648, Nov. 2, 2006.
[8] P.Mahoney, Y. Savaria, G. Bois, and P. Plante, "Parallel hashing memories: An alternative to content addressable memories", in Proc. 3rd Int. IEEE-NEWCAS Conf., Jun. 2005, pp. 223226.