



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)

Volume 3, Issue 1, January 2014

# Dynamic Based Reconfigurable Content Addressable Memory for FastString Matching

N.Manonmani<sup>1</sup>, K.Suman<sup>2</sup>, C.Udhayakumar<sup>3</sup>

Dept of ECE, Sri Eshwar College of Engineering, Kinathukadavu, Coimbatore, India<sup>1</sup>

Dept of ECE, PSG College of Technology, Peelamedu, Coimbatore, India<sup>2</sup>

Assit Prof Dept of ECE, Sri Eshwar College of Engineering, Kinathukadavu, Coimbatore, India<sup>3</sup>

*Abstract- A Content-Addressable Memory (CAM) stores data in a similar fashion to a conventional RAM. However, "reading" the CAM involves providing input data to be matched, then searching the CAM for a match so that the address of the match can be output. Networking has typical application in software application must typically be compared to high speed network traffic, and string matching it becomes a bottleneck. For that purpose a Variable Word-Width CAM for fast string matching has been designed and implemented in an FPGA. It is used to increase the speed of searching operation and string matching, it upsurges the speed 8 times faster. The CAM design has been simulated with Model Technology ModelSim 6.3f, and synthesized by Xilinx ISE 9.1. It was then loaded into a FPGA.*

*Index Terms- Field Programmable Gate Array, Content Addressable memory, Intrusion Detection system.*

## I. INTRODUCTION

Content Addressable Memories or CAMs are a class of parallel pattern matching circuits. These circuits function like standard memory circuits and may be used to store binary data. Dissimilar to standard memory circuits, however, a powerful match mode is also available. This match mode authorities all of the data in the CAM device to be searched in parallel [1]. The speed of today's networks is of such a kind that a general purpose CPU must struggle to process the network data. The CPU must also have incomes left for other application processes. The amount of processing required on network data is increasing due to the need for intrusion detection, cryptographic processing and more [2].

Here the work deal with a part of an Intrusion Detection System (IDS), Snort, the string matcher will be implemented in hardware. The main involvement of this paper is to implement a Variable Word-length Content Addressable Memory (CAM) in FPGA for string matching. In order to design a system for an FPGA, actual CAM design in VHDL by taking advantage of all programming abilities that is common to any programming language.

## II. BACK ROUND

### A. Content Addressable Memory

CAM and Random Access Memory (RAM) circuits are similar in structure, but it different to RAM, in which the stored data are identified by means of unique address assigned to each data word, CAM words are identified by their content. The device are written or read the data [3] shown in fig. 1. In addition to functioning as a standard memory device, parallel search or match done by CAM. The entire memory array can be searched in parallel using hardware. In this match mode, each memory cell in the array is accessed in parallel and compared to some value. Match is found in memory location, a match signal is generated. CAM is very useful in applications such the search operations are to be performed.

CAM is two types' binary CAM and ternary CAM. Binary CAM stores '0' and '1', ternary CAM stores '0', '1', and 'X'

### 1. Return Value

The entries in a CAM have two parts. It has more important field in searching operation, which is the part of the entry that is matched with the search pattern. The CAM entries also contain a return field; during the read operation the information is returned. This contains either related information or an index. In some cases, one is not only able to write to the search field, but also to the return field, so that the return value can be programmed per entry.

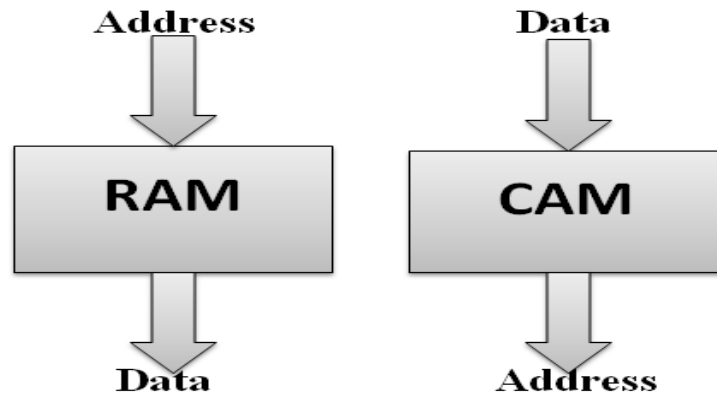


Fig. 1 RAM versus CAM functionality

**2. Prioritizing**

Since the entries stored in the CAM may contain ‘don’t cares’, there is a possibility that two or more entries give a match at the same time.

**Inherent priority** exploits the CAMs predictable ordering when reading multiple matched data. In this case, in the order of priority system stores the entries. By using a priority encoder, the top address of the CAM has the highest priority (0) and the bottom address has the lowest priority (127).

**Explicit priority** The inherent priority can be replaced with an explicit priority field added to each CAM word. In case of a multiple match, the entry with the highest explicit priority as stored in the priority field is Returned. The advantage of explicit priority is that updating the CAM becomes easier, since a new entry can always be added at the end. When using inherent prioritizing, new entries are not always added in the end of the CAM and an address has to be reserved by shifting down other entries and updating the memory that is addressed by the CAM. Fixed Length CAM has same amount of area is reserved for all entries. This is referred as fixed length CAM. In Variable Length CAM Area that each entry occupies is variable and depends on the number of ‘don’t cares’. This is referred as Variable length CAM.

**III. CAM DESIGN**

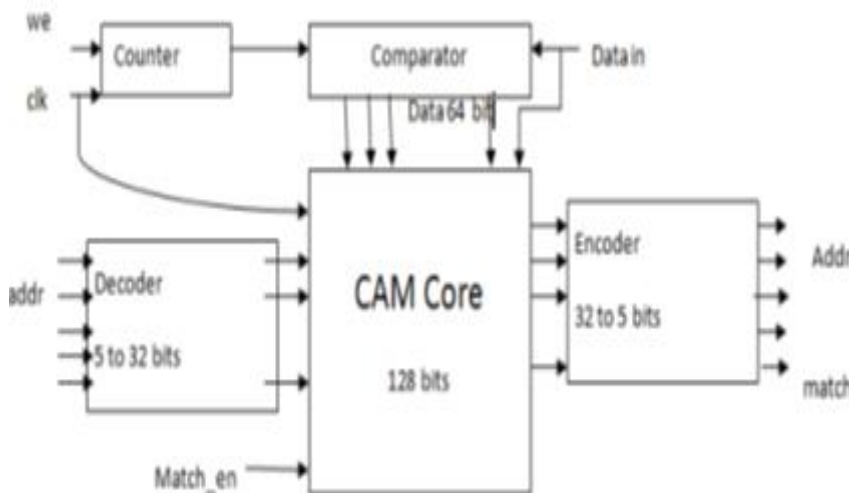


Fig. 2 CAM Design

The designed CAM size is 128 words. It can store 128 words in the match lines and it can compare those words and returns the match and match address. The design is having a priority encoder which decides the priority of match when multiple matches are found in the design it is shown in fig. 2.



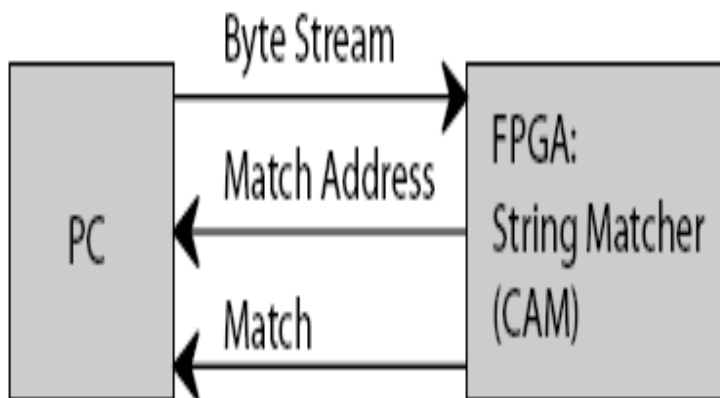


Fig. 4 CAM Applied in a String Matching System

**A. Shift Register**

In the fixed length CAM, primitive SRL16 was used to implement a LUT whose inputs are notswapped. In the variable length CAM this primitive is instantiated to be used as a shift register as well. A description of SRL16 is given in Fig. 5. The data (D) is loaded into the first bit of the shift register and during subsequent Low-to-High clock transitions data is shifted to the next bit position as new data is loaded. The data appears on the Q outputs when the shift register length determined by the address inputs is reached. The length of the shift register can be changed dynamically and is equal to:  $(8 \cdot A3) + (4 \cdot A2) + (2 \cdot A1) + A0$ .

In the VHDL code, the length is initialized at 1 by driving '0' on all address inputs. To change the length during operation, some address inputs need to drive '1' and this is done by 'disconnecting' these inputs. This way, these inputs are not driven and a pull up causes them to become high.

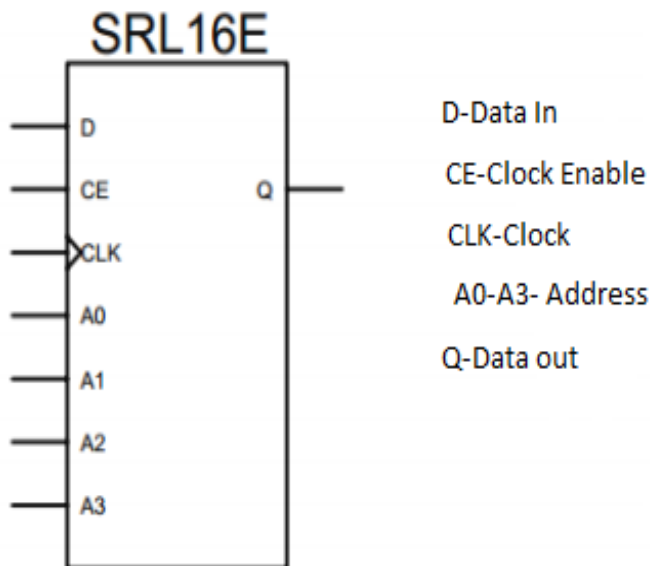


Fig. 5 structure of SRL16E

**B. CAM Read Mode**

Fig. 6 illustrates CAM in read mode. In shift register each location is connected to all matching units in the corresponding row as indicated by horizontal lines. The CAM has a latency of two clock cycles. When the output registers of the encoder are updated match enable goes high, For each clock cycle we giving a new data to the shift register, for each clock cycle we will get a new valid output.

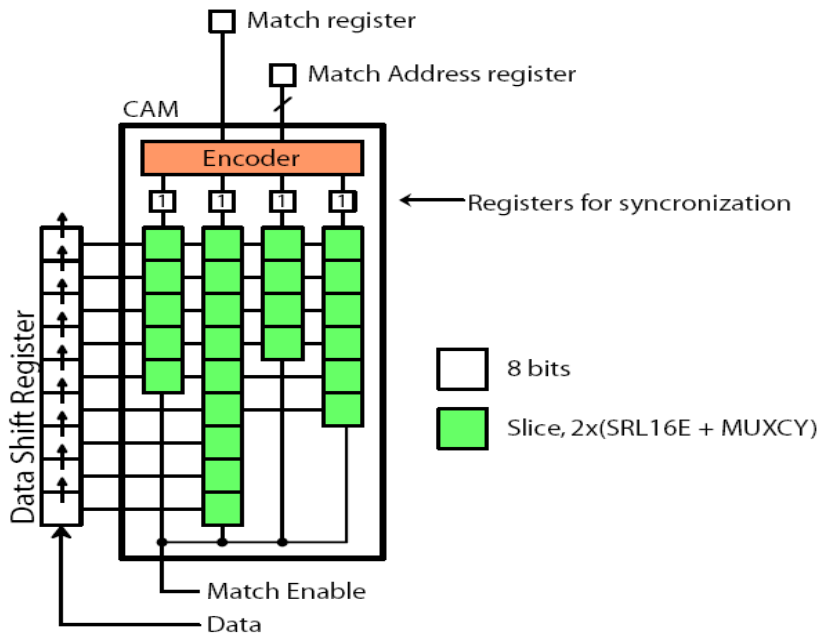


Fig. 6 CAM Read Mode

**C. CAM Write Mode**

Fig. 7 gives an overview of a CAM in write mode. Each comparator is connected to all matching units (SRL16Es) in the corresponding row. At a time only one word can be written. Decoder selects the word. the decoder passes on the Write Enable signal —for the chosen word output is given by the counter.

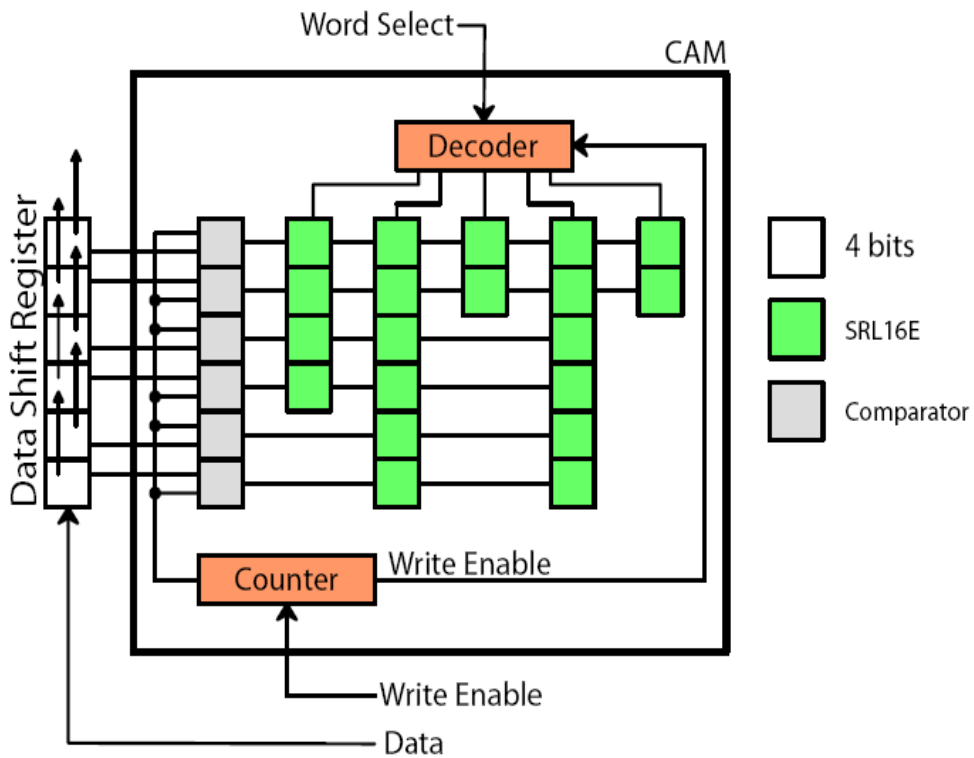


Fig. 7 CAM Write Mode Overview



ISSN: 2319-5967

ISO 9001:2008 Certified

International Journal of Engineering Science and Innovative Technology (IJESIT)  
Volume 3, Issue 1, January 2014

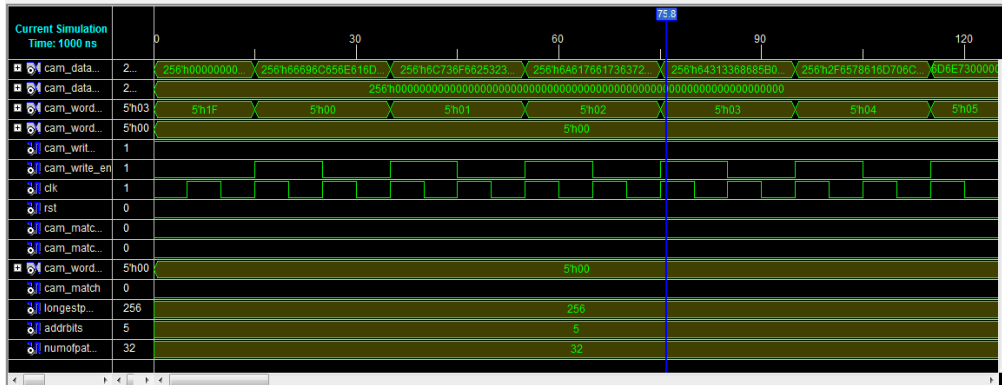


Fig. 8 Simulation Results for CAM

Simulation is done using VHDL codes in Xilinx ISE Project Navigator 9.2. The design is implemented in Virtex2 pro FPGA board from Xilinx. The design includes the basic blocks of CAM. They are of Counter, Comparator, Encoder, Decoder, CAM core and the cam components library. The basic blocks were integrated and it is simulated using VHDL test benches shown in fig. 8.

Table. I Speed for Searching and Matching

With/without using CAM for string matching	Words	Bytes	Reported Max Speed (MHz) Static Timing Report	Max speed of incoming bit stream (Mbit/s)
With out using CAM for string matching	128	1822	100	100
With using CAM for string matching	128	1822	105	840
With using CAM for string matching	256	3601	100	800

### VII. CONCLUSION AND FUTURE WORK

A Variable length CAM has been designed. And this design is used for fast string matching and searching operation in network application. CAM has high flexibility and parallel searching operation. it used to increase the speed of searching operation and string matching. it performs 8 times faster than before using of CAM in IDS.

In future, the JBits can be used to modify the content of the CAM dynamically. Thus this application helps us to handle varying width data in CAMs in run time. Thus a Dynamically reconfigurable Content Addressable Memory can be developed and implemented using Xilinx Virtex 2 pro FPGA.

### REFERENCES

- [1] K. E. Grosspietsch, "Associative processors and memories: a survey," IEEE Micro, vol. 12, no. 3, pp. 12–19, Jun. 1992.
- [2] Peter Bellows et al. GRIP: A Reconfigurable Architecture for Host-Based Gigabit-Rate Packet Processing. FCCM 2002.



ISSN: 2319-5967

ISO 9001:2008 Certified

**International Journal of Engineering Science and Innovative Technology (IJESIT)**

**Volume 3, Issue 1, January 2014**

- [3] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison-Wesley Publishing Company, 1985.
- [4] A Reconfigurable CAM Architecture for Network Search Engines, Nourani, M., Vijayasarathi, D.S., Balsara, P.T. 2006, IEEE.
- [5] Reconfigurable cache memory architecture for integral image and integral histogram applications Po HaoHsu ; Shao-Yi Chien 2011,IEEE.
- [6] Reconfigurable computing using content addressable memory for improved performance and resource usage Paul, S. Bhunia, S. 2008, IEEE.
- [7] A Flexible Design of Network Devices Using Reconfigurable Content Addressable Memory Qutaiba Ali 2011, IEEE.
- [8] Scalable Multi gigabit Pattern Matching for Packet Inspection., IoannisSourdis, Dionisios N. Pnevmatikatos, StamatisVassiliadis 2008,IEEE.
- [9] A Memory-Efficient and Modular Approach for Large-Scale String Pattern Matching. Hoang Le, Viktor K. Prasanna 2013, IEEE.
- [10] Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey. Kostas Pagiamtzis, Ali Sheikholeslami 2006, IEEE.