Dynamic Based Reconfigurable Content Addressable Memory for Fast String Matching

N. Manonmani¹, K. Suman², C. Udhayakumar³
Dept of ECE, Sri Eshwar College of Engineering, Kinathukadavu, Coimbatore, India
Dept of ECE, PSG College of Technology, Peelamedu, Coimbatore, India
Assit Prof Dept of ECE, Sri Eshwar College of Engineering, Kinathukadavu, Coimbatore, India

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 authorizes 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. BACKGROUND

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.
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

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.
The structure of the VHDL code of the variable length CAM is much like the fixed length CAM and is given in fig. 3. The main differences are that two multiplexers were added for each match line (entity Mux4) and flip flops were replaced by shift registers.

**IV. DYNAMIC RECONFIGURATION**

Reconfigurable memories are widely used in the routers and firewalls. In order to increase the speed and flexibility we go for reconfigurable memories. Full custom Content Addressable Memories that are fast and can store a large amount of data. By implementing the CAM in FPGA, matching is done in hardware which is faster than doing it in software.

**Advantages**

The CAM functionality can therefore be integrated with other logic on the same chip. Implementing a CAM in FPGA technology gives the possibility to add extra features.

**V. INTRUSION DETECTION SYSTEM**

An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also react to anomalous or malicious traffic of network searching speed. IDS on exact string (content) matching. String matching based on software has not been able to keep up with high network speeds, and hardware solutions are needed. Content Addressable Memory (CAM) used for improving the performance of IDS systems. As the string matcher is often the main bottleneck of an IDS, a variable length CAM is used to avoid this problem. And it is used for increasing the speed of searching and matching. This design clearly offers an improvement to IDS. It performs 8 times faster than before used of CAM in IDS as shown in table 1.

**VI. MAKING A STRING MATCHER**

CAM has been applied for string matching in this work. Using of CAM the data to be matched as a Byte Stream. In parallel, the data is stored in the CAM; the string is compared to all. If a match is found, it is indicated by the Match bit. The Match Address reports the “address” of the string that matched in the CAM. When match is found, exact string matching is performed.
A. Shift Register
In the fixed length CAM, primitive SRL16 was used to implement a LUT whose inputs are not swapped. 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: \((8*A3) + (4*A2) + (2*A1) + A0\).

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.

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.
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.
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 1: Speed for Searching and Matching

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

### 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


