Hardware based String Matching Algorithms: A Survey

Gulfishan Firdose Ahmed
Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal-462051

Nilay Khare
Department of Computer Science and Engineering, Maulana Azad National Institute of Technology Bhopal-462051

ABSTRACT
There are various string matching Algorithms which are software based but some are hardware based. The main factor of string matching algorithm is depending on searching efficiency. In this paper we have discussed about the hardware based string matching algorithms such as Brute Force, KMP, and Aho-Corasicks with their applications. There are different types of string matching algorithms which are software based solution and hardware based solution. Since software-based solutions are slower and less efficient, now a day, so the hardware-based solutions are highly preferred. Hardware based approaches are more efficient in terms of speed, memory size and power consumption than software based approaches. Hardware based solutions has great importance in the real life applications. This paper focus on the hardware based solutions and describes the hardware based implementation of string matching algorithms such as Brute Force, KMP and Aho- Corasicks.

General Terms
String matching, Algorithms

Keywords
Exact and Approximate String Matching, Brute Force, KMP, Aho-Corasick ,BWT etc

1. INTRODUCTION
The problem of string matching deals for finding whether a pattern occurs in the long text or not. The string matching algorithms can be divided into two major types, such as exact string matching and approximate string matching. There are mainly two techniques of string matching one is exact matching (Brute Force , Boyer Moore, Aho-Corasicks ,Needleman Wunsch, Smith Waterman, KMP,BMH etc) and other is approximate matching (Shift-OR).These algorithms are character based string matching algorithms. String matching, generally including exact string matching and regular expression matching is used in many applications. Exact string matching approaches can be further divided into software-based solutions and hardware based solutions. Since software-based solutions are slower and less efficient, now a day so the hardware-based solutions are highly preferred. The hardware based string matching algorithms is mainly depends on area of chip and power consumption [1].There are many algorithms which are implemented in both software and hardware. Software based algorithms are Brute Force Algorithm, KMP, Aho-Corasick, Boyer Moore , Boyer Moore Horspool, Shift-OR etc, but some are implemented in hardware such as Brute Force ,KMP algorithm and Aho-Corasick algorithm.The Brute Force algorithm is the basic algorithm that was used earlier and the complexity is $O\ (m * n)$. After Brute Force Algorithm, Morris Pratt Algorithm came that increase searching time complexity, then KMP Algorithm came depends on Morris Pratt it reduces the false mismatches and improve the searching time complexity. These algorithms are hardware based to improve the searching efficiency but these all are character based algorithms. Due to some limitation in software based solution these algorithms are implemented to improve the performance and searching efficiency.. The main factor of string matching algorithm is depends on the searching time complexity. This paper, focus only those algorithms which are hardware based implemented and to discuss the drawbacks of existing both software and hardware based algorithms. Hardware based approaches are more efficient in terms of speed, memory size and power consumption than software based approaches.

2. LITERATURE SURVEY AND RELATED WORK

2.1 Hardware Implementation of Brute Force Algorithms
The very first and basic method of string matching is the Brute Force algorithms. The brute force algorithm is the basic algorithms to solve the real world applications. This algorithm consists for checking all possibilities from the text, and check occurrences of the pattern starts there or not, after each attempt the pattern is shifted by exactly one position to the right. The searching time complexity of Brute force Algorithm is $O\ (m * n)$. There is no requirement of Pre-processing Phase and additional constant space is needed for the pattern and the text [1]. At the application layer, the need of packet payload inspection layer is improved as there is increase in the complexity of network environment. String matching, is used to detects malicious network attacks by a set of rules, hence it is very much important for network intrusion detection systems. The Brute Force Algorithm is used to detect malicious node at network and host levels [2].

The hardware architecture of Brute Force algorithm consists of two components.

1. Process Element: In Process element n^2 comparators are needed for matching n characters per step. When process width is increased linearly, the number of comparators is increased exponentially. To solve the hardware cost problem, the process element which uses n comparators with process width n.The figure 1, show architecture of process element.

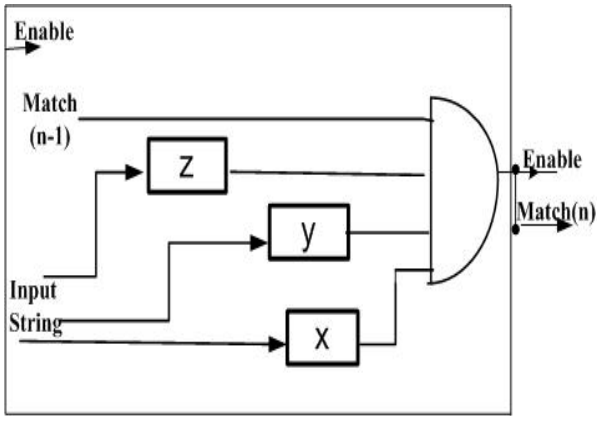

Fig.1: Hardware architecture of Process Element

2. Alignment element: Process element cannot search the pattern in the all the possible input strings. The multi-character matching problem is solved by alignment element. If the suffix of input string is matched to the prefix of the matching target pattern, the alignment element aligns input string. One alignment element is needed to one pattern. The figure 2, show architecture of process alignment. The alignment element handles partial matching of substring in the first process element. Therefore, the required comparators per alignment element increase as the process width increases. By using the process element with any process width, the requirement of the comparators is reduced. The performance of the architecture is nearly equal as the previous work, but this string matching architecture reduces the hardware cost.

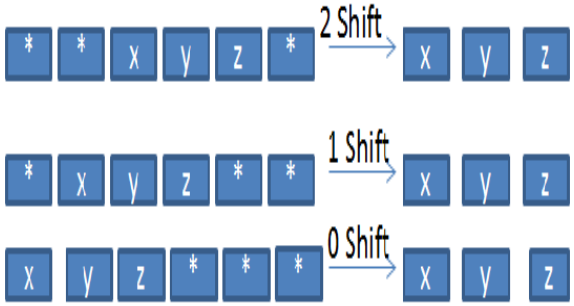

Fig.2: Hardware architecture of Process Alignment

The Brute force algorithm is used in network intrusion detection system. With the helps these both the process elements and alignments architecture brute force algorithm is used in detecting malicious network attacks using set of rules. It reduces the requirements of comparators and it also reduces the hardware cost of the architecture.

2.2 Hardware Implementation of KMP Algorithms

The KMP algorithm was developed after Morris Pratt Algorithm to improve the searching time complexity. KMP algorithm is based on Morris Pratt, here prefix function is computed. The KMP algorithm uses the observation that a mismatch pattern contains sufficient information, for find out the location of the next possible match. It speed up the string search by skipping there examination of previously matched characters [1]. The specific pattern is used by the KMP algorithm to build the prefix function table, using memory-based FSM to reconfigure and speed up the KMP search algorithm. The prefix function is transformed into a state machine and implemented as a reconfigurable Finite State machine (FSM). A dynamically reconfigurable Finite State Machine (FSM) was implemented using on-chip memory and an embedded processor [3, 4].The block diagram of KMP logic is shown in figure 3.

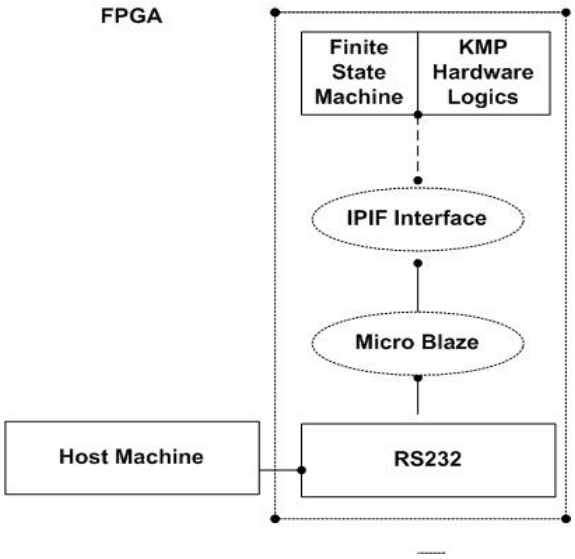

Fig 3: Block Diagram of KMP logics

The FSMs for KMP and KMP search execution logic are modelled as separate VHDL entities. For creation and synthesis of source files, Xilinx ISE 10.1 is used. The most important factors that limit performance improvement in this FPGA-based embedded system are: 1) data transfer rate, and 2) memory bandwidth. The main application of KMP algorithm is used in Cryptographic ciphering Algorithms.

2.3 Hardware Implementation of Aho-Corasicks Algorithm

Aho-Corasick is the one of the earlier algorithm of string matching algorithm. This algorithm is the multi pattern algorithm based on AC automaton. The AC-algorithm gives excellent performance for exact string matching. There are many hardware architectures based on the AC-algorithm for accelerating string matching. This paper described an intuitive and efficient algorithm to construct multi character transition functions that represents a multi- character FSM from an AC-tire. The proposed algorithm iteratively concatenates the 1-character transition functions derived from the AC-trie to construct the multi-character transition functions. By using AC-trie example in figure 4 shows the physical lines represent the goto functions and the dotted-lines represent the failure functions linked to the non-initial states. State 0 is the initial state or the root state of the AC-trie. Every non-initial state has failure functions while the failure functions linked to the initial state are not shown. The non-empty matching outputs are shown beside the corresponding states. Each state in an AC-trie represents a unique string which is the prefix of one of the keywords forming the AC-trie.

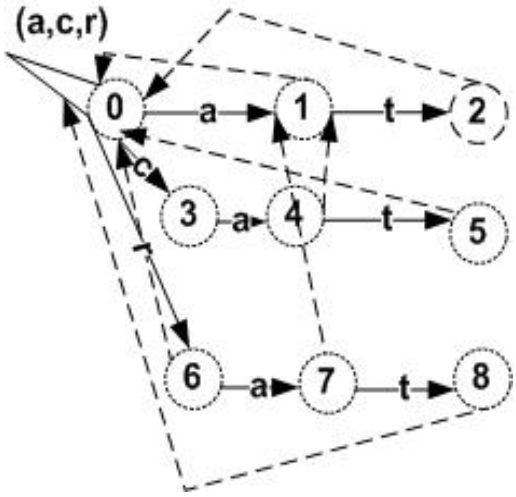

Fig 4: AC-Trie for keywords [at,cat,rat]

The main limitation of the AC-algorithm is a memory exhausted algorithm by improving the efficiency of hardware utilization. The string matching device accepts n-character input denoted by IN_ CHRS and produces n matching outputs from OPT_1 to OPT n corresponding from the first to the last characters of IN_CHRS respectively. The multi-character string matching device includes m matching unit's $n + 1$ priority multiplexers, and $n + 1$ registers. Each matching unit is responsible for executing one n-character transition rule. To evaluate the hardware approaches there are two ways. Firstly, compare the numbers of generated transition rules for different number of keywords and different n-character Finite State Machines. Secondly, implement the architecture on an ASIC device using FPGA synthesis tools to evaluate the utilization of hardware resource and estimate achievable throughput. The hardware architecture for implementing the derived multi-character Finite State Machine consists of multiple matching units where each matching unit is responsible for one multi character transition function [5, 6]. The main application of this Aho-Corasick algorithm in NIDS. This algorithms depend on multiple keywords are matched in high speed.

2.4 FPGA-Based hardware Implementation of the FM-Index

The FM-index is based on the index which contains the BWT of the text to be searched in the form of a set of integer type arrays. The BWT is an algorithm that transforms an input data to another form that is easily compressible and reversible to its original form. The Burrows wheeler Transform is related to suffix array. The indexing is using binary search for searching a pattern from the text. The FM indexing includes two tables for storing the data, I-Table and C-Table. The FM-index does not require performing all character comparisons compared to the brute force approach. The main limit of the FM-index hardware implementation is depends on the size of the memory available on the FPGA [7,8].

3. VARIOUS HARDWARE BASED STRING MATCHING ALGORITHMS COMPARATIVE ANALYSIS

Several data structures have been developed since the seventies, due to the importance of the matching problem based on dozens of algorithms. Based on the analysis we have concluded that the Brute Force algorithm is the very basic algorithm for string matching; it uses process element and process alignment to improve the efficiency of the Brute Force Algorithm but still this is slow to improve the performance by implementing in hardware. KMP is based on the Prefix function to improve the efficiency. Both algorithms are single pattern algorithm. Aho-corasick algorithm is the multiple pattern algorithms which depend on AC Trie. AC trie consist of three functions first one is goto function, second is failure and third one is output functions. It performs Exact String Matching in less time for small keyword set. The main limitation of Aho-Corasick Algorithm when the size of the Trie grows enormously, if the pattern set grows and also time of searching increases.

Table 1. Comparative Analysis of Hardware Based Algorithm

Algorithm	Search Time	Discussion
Brute Force	$O(m * n)$	It uses process element and process alignment
KMP	$O(m + n)$	Based on KMP hardware Logic
Aho-Corasick	$O(n)$	ASIC device for evaluation and the throughput achieve 4.5 Gbps
FM Indexing using BWT	-	It uses comparators, registers, shifters and multiplexors, FPGA

In this paper, there are only four hardware based string matching algorithms are discussed such as Brute Force, KMP, Aho-Corasick and FM indexing FPGA based algorithms, which are implemented in Hardware. Now a day, Aho-Corasicks Algorithms is widely used in solving various real life applications. This algorithm is based on multi pattern string matching and AC prefix trie. These are also depends on various parameters in terms of clock speed, area of chip and power consumption but here we consider hardware requirements and their characteristics. FM indexing using BWT is mainly used in exact string matching algorithms.

4. CONCLUSION

The main focus of this paper, presented several hardware based solutions for exact string matching such as brute force, KMP, Aho-Corasick algorithms with their application. We have concluded that, brute force algorithm is a single pattern algorithm and character based algorithm, that was proposes hardware-efficient string matching architecture. The hardware architecture consists of the alignment elements and the process elements uses n number of comparators. Therefore, the proposed string matching architecture reduces the hardware cost with the same performance. Second one is KMP algorithms, reduces the false mismatches and it is also single pattern and character based algorithm. This was a new approach based on Finite State machine. The test cases data clearly indicates that irrespective of the pattern length, the approach explained achieves maximum design frequency. The design increased the performance of a pattern matching application. Memory size required implementing an FSM

increases with the size of input vector, output vector, and number of bits needed to represent the states. Third one is the Aho-Corasick string matching algorithm, which is a multi-pattern and automata based string matching algorithm. In intrusion detection systems, a hardware based string matching architecture is used to accelerate string matching in various applications. The last one, the FM-index based has a higher valuable throughput than the brute force. Based on these hardware implementation, we have concluded that hardware based approaches are more efficient in terms of speed, memory size and power consumption than software based approaches. Now a days, hardware based string matching architectures are highly preferred in various real life application.

5. REFERENCES

[1] Christen Charras and Thierry Lecroq, "Hand Book of Exact String Matching Algorithms", pp.251-288, Addison-Wesley Publishing Company, 2011.

[2] Seongyong Ahn, Hyejong Hong, Hyunjin Kim, Jin-Ho Ahn, "A Hardware-Efficient Multi-character String Matching Architecture Using Brute-force Algorithm", pp. 464-467,IEEE , ISOCC, 2009.

[3] D.E. Knuth, J.H. Morris and V.R. Pratt, "Fast pattern matching in strings," SIAM Journal of Computing, vol. 6, no. 2, pp. 323–350, June,1977.

[4] Indrawati Gauba, "A Reconfigurable Pattern Matching Hardware Implementation using on chip RAM Based FSM", thesis Boise State University, August, 2010.

[5] Chien-Chi Chen and Sheng-De Wang, "A Multi-character Transition String Matching Architecture Based On Aho-Corasick Algorithm", ICIC International Volume 8, pp. 8367-8386, 2012.

[6] Leena Salmela, J. Tarhio and J. Kytojoki, "Multi-Pattern String Matching with Very Large Pattern Sets", ACM Journal Algorithmic, Volume 11, 2006.

[7] M. Burrows and D.J. Wheeler, "A Block-sorting Lossless Data Compression Algorithm," SRC Research Report, May, 1994.

[8] Edward Fernandez, Walid Najjar and Stefano Lonardi, "String Matching in Hardware using the FM-Index", IEEE International Symposium on Field-Programmable Custom Computing Machines, 2011.