Pattern Matching Algorithm Using Filter Engine and Exact Matching Engine

*D.Rakesh, **L. Padmalatha

*PG Student in ECE Department, Gudlavalleru Engineering College, India

**Associate professor in ECE Department, Gudlavalleru Engineering College, India

ABSTRACT

Network intrusion detection system[1] is used for detecting the virus or unwanted data in the security fields. Network Security is increasing in many applications. Now a days network accessing is done by mobiles is increased. Network security is increasing for the mobile devices because of virus present in network. But mobiles have less power and small hardware. Here pattern matching is used for detecting virus in mobiles or other network devices. The pattern matching is done by using Aho-corasick algorithm, Boyer moore algorithm or memory architectures. By using these algorithms the throughput comes low, but by using memory based memory architecture [4] the throughput comes high and power also high. So in proposed an BITCAM is used for a high-speed, low-power and low-cost virus-detection processor in devices. The proposed match-line scheme reduces area and complexity. The design of the adjustable division line provides high flexibility for updating all data patterns. The idea of proposed virus-detection is to condense as much information on-chip as possible such that most of input data can be quickly scanned without further inspection. The entire virus scanning is split into two phases: fast on-chip filtering by the filtering engine, and the exactly- matching with some off-chip memory accesses.

Keywords – **Pattern matching, Filter engine, Exactly matching engine, BITCAM, Virus detection.**

1. INTRODUCTION

Content addressable memory (CAM) is used for comparing the input data with data stored in the memory. CAM performs parallel operation so search speed is increasing. Network security systems require a great amount of pattern matching operations to compare the input network packet with the pre-defined rule set for protecting the system from network attacks such as worms and viruses. When dealing with a large number of virus patterns, these designs need a large chip area and significant power due to the enlarged size of the on-chip memory. so CAM-based designs can achieve higher search speed. Here general CAM not satisfy the requirements, so BITCAM is used for fulfilling the requirements. In the proposed architecture filter engine have two planes they are YES plane and NO plane. It as exact matching engine it stored all the data. By using BITCAM in this architecture the throughput is increasing and power is decreasing. When input string is given that string is compared with patterns stored in memory. If pattern is matched the virus is detected. It can apply in the ATM's for pattern matching.

2. RELATED WORK

In previous so many algorithms are used. But mostly Aho-Corasick algorithm is used. It can apply in the finite state machines . It can be easily reconfigurable the hardware usage is more. The idea is in mobiles the network usage is more but some viruses can attack the mobile. But Mobiles have less hardware and less power. So here proposed a method for detecting the virus attack patterns in the mobile by using the BITCAM's. By using this the virus attack pattern can search easily. The hardware usage is less as compare to previous for virus pattern detection detection. So this is applied for detection of virus patterns.

3. MOBILE VIRUS

A mobile phone virus is a computer virus specifically adapted for the cellular environment and designed to spread from one vulnerable phone to another. Although mobile phone virus hoaxes have been around for years, the so called Cabir virus is the first verified example. The virus was created by a group from the Czech Republic and Slovakia called 29a, who sent it to a number of security software companies, including Symantec in the United States and Kapersky Lab in Russia. Cabir is considered a "proof of concept" virus, because it proves that a virus can be written for mobile phones, something that was once doubted.

Cabir was developed for mobile phones running the Symbian and Series 60 software, and using Bluetooth. The virus searches within Bluetooth's range (about 30 meters) for mobile phones running in discoverable mode and sends itself, disguised as a security file, to any vulnerable devices.

4. CONTENT ADDRESSABLE MEMORY

Content-addressable memories (CAMs) [5] are hardware search engines that are much faster than algorithmic approaches for search-intensive applications. CAMs are composed of conventional semiconductor memory (usually SRAM) with added comparison circuitry that enables a search operation to complete in a single clock cycle. The two most common search-intensive tasks that use CAMs are packet forwarding and packet classification in Internet routers.

A CAM is a memory that implements the lookup-table function in a single clock cycle using dedicated comparison circuitry. CAMs are especially popular in network routers for packet forwarding and packet classification, but they are also beneficial in a variety of applications that require high-speed table lookup. The main CAM-design challenge is to reduce power consumption associated with the large amount of parallel active circuitry, without sacrificing speed or memory density. A CAM search operation begins with all match lines high, putting them all temporarily in the match state. Next, the search line drivers broadcast the search data, 10001001 onto the search lines. Then each CAM core cell compares its stored bit against the bit on its corresponding search lines. Cells with matching data do not affect the match line but cells with a mismatch pull down the match line. Cells storing an X operate as if a match has occurred. The aggregate result is that match lines are pulled down for any word that has at least one mismatch. All other match lines remain activated.

4.1 Binary and Ternary CAM

Binary CAM is the simplest type of CAM which uses data search words comprised entirely of 1s and 0s. Ternary CAM allows a third matching state of "X" or "Don't Care" for one or more bits in the stored data word, thus adding flexibility to the search. For example, a ternary CAM might have a stored word of "10XX0" which will match any of the four search words "10000", "10010", "10100", or "10110". The added search flexibility comes at an additional cost over binary CAM as the internal memory cell must now encode three possible states instead of the two of binary CAM. The combination of both binary and ternary is known as Bit CAM.

5. VIRUS SIGNATURES

A signature is a characteristic byte-pattern that is part of a certain virus or family of viruses. If a virus scanner finds such a pattern in a file, it notifies the user that the file is infected. The user can then delete, or (in some cases) "clean" or "heal" the infected file. Some viruses employ techniques that make detection by means of signatures difficult but probably not impossible. These viruses modify their code on each infection. That is, each infected file contains a different variant of the virus. Most modern antivirus programs try to find virus-patterns inside ordinary programs by scanning them for so-called virus signatures.

5.1 Virus Detection

The design considerations for a virus-detection in devices are analyzed as follow:

The system throughput should reach up to 1 Gbps for supporting real-time virus detection. The scalability of handling more than ten thousands patterns is required for versatile network protection. In addition, the system must be highly flexible to accommodate

the rapidly increasing new virus patterns. Power consumption is the most important design consideration for mobile devices.

The increasing virus pattern will greatly increase the power consumption and the cost of on-chip CAMs. The memory design is critical for dealing with the increasingly large virus database.

5.2 Virus Detection processor

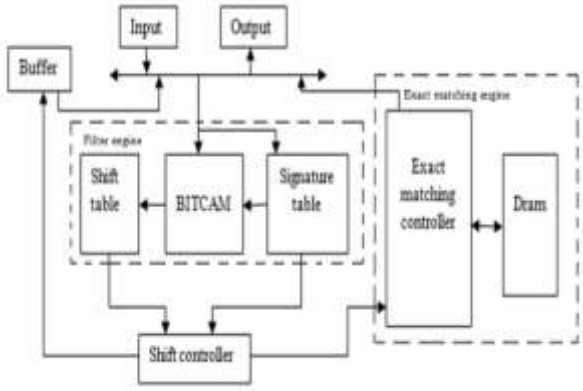

Fig. 1 Architecture of virus detection processor

The above Fig.1 shows architecture of virus detection processor. The proposed virus-detection processor is to condense as much information on-chip as possible. Most of input data can be quickly scanned without further inspection. The entire virus scanning is split into two phases: fast on-chip filtering by the filtering engine, and the exactly- matching with some off-chip memory accesses. The filtering engine have BITCAM lookup tables (named no-plane and yes-plane), which are used to perform the on-chip data-scanning.

A. No-plane Structure

The Filtering engine screens impossible matches by consulting two TCAM lookup tables (named no-plane and yes-plane). which are used to perform two steps of the on-chip data-scanning to obtain a fast shift table. which indicates the impossible matching patterns. By comparing the input datum with the No-plane TCAM from the least significant bit (LSB), the engine first looks up the shift table to perform a quick shift of impossible bytes until locating a possible match. If the input data is matched with an entry of No-plane, the input string will be skipped according to the shift count stored in the shift SRAM.

B. Yes plane

When the comparison of No-plane is missed or if the corresponding shift-count is zero, the filtering engine will enter the second step of virus detection. Then further look up another signature table (called the Yes-plane) to eliminate any false positives by ensuring that the prefix has the same signature. The Filtering engine will skip the input datum if it is mismatched with the data of the Yes-plane. If a possible match is still not ruled out, then the **Exact-matching engine** performs suffix matching by making comparisons with a suffix tree stored in off-chip memory, which can hold a large number of virus patterns.

C. Match Line Scheme in BITCAM

The proposed AND-type match-line scheme [7] can be applied in either the binary CAM (BICAM) or the ternary CAM (TCAM). Here NO plane and YES plane data is merged if any don't care bits present. So it names as BITCAM. The data string is given as input, in that MSB side 4-bits are taken as input to signature table. In signature table first 4-bits are taken as signatures. When the input is matched in signature table the searching operation starts directly on that memory address location. If input string is matched in that memory location the output comes '1'(virus pattern detected) if not matched comes '0'(No Virus pattern detected). The matching is done by AND gates to generate the final matching result. Match line scheme is shown in below Fig.2. If input is not matched it goes to Exact matching engine which can hold a large number of virus patterns. It search the given string in that engine and detect the virus pattern is detected are not.

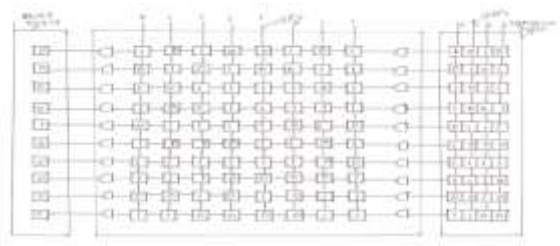

Fig.2 Match lining scheme in Filter engine

6. SIMULATION RESULTS

When the input string pattern is given this pattern is compared with the stored virus patterns. If the string pattern is matched with the virus pattern present in the filter engine the result shows as high or one. The below Fig.3 shows when virus pattern is detected.

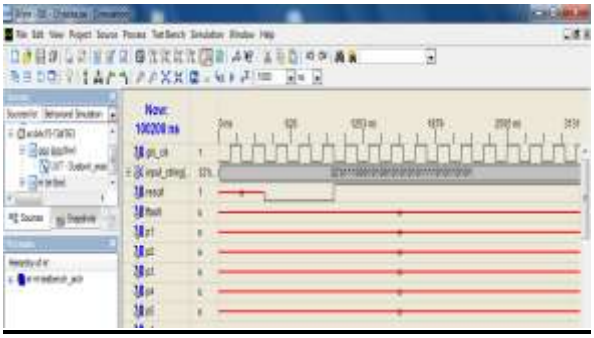

Fig.3 Virus pattern detected in Filter engine

If string pattern matched with the virus pattern present in exact matching engine then final fault becomes high as shown in Fig.4.

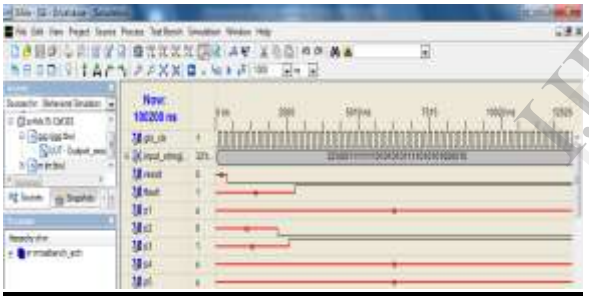

Fig.4 Virus pattern detected in Exact matching engine

When input pattern not matched with any virus patterns the result become low or zero as shown in Fig.5.

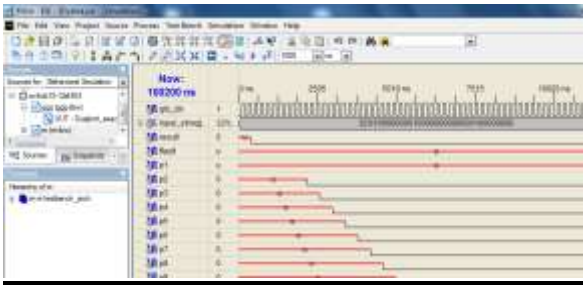

Fig.5 Virus pattern not detected

7. CONCLUSION

By using the proposed BITCAM the throughput becomes high, power consumption is low and cost also reducing for detecting virus patterns presenting in mobiles. Not only in mobiles it can be applicable in ATMs and network security devices and etc. by using BITCAM the power and area of architecture are reducing up to 20%.

REFERENCES

1. Sangkyun, "An efficient TCAM-based implementation of multi-pattern matching using covered state encoding," in IEEE Trans. Comput., vol. 60, pp. 1–9, 2011.

2. Cheng-Hung Lin, Student Member, "Optimization of Pattern Matching Circuits for Regular Expression on FPGA," IEEE, Chih-Tsun Huang, Dec 2007.

3. L. Tan and T. Sherwood, "A high throughput string matching architecture for intrusion detection and prevention," in Proc. IEEE Int. Symp. Computer Architecture, pp. 112–122, 2005.

4. Sarang Dharmapurikar, John Lockwood, "Fast and Scalable Pattern Matching for Content Filtering," Oct 2005.

5. Y. H. Cho and W. H. Mangione -Smith, "A pattern matching coprocessor for network security," in Proc. IEEE Int. Conf. Design Automation, pp. 234–239, 2005.

6. J. S. Wang, H. Y. Li, C. C. Chen, and C. W. Yeh, "An AND-type match-line scheme for energy-efficient content addressable memories," in IEEE Int. Solid-State Circuits Conf. Dig., 2005.

7. K. J. Lin and C. W. Wu, "A low-power CAM design for LZ data compression," in IEEE Trans. Comput., vol. 49, no. 10, pp. 1139–1145, 2000.

8. About ClamAV. [Online]. Available: http://www.clamav.org/.