IOSR Journal of Engineering
Apr. 2012, Vol. 2(4) pp: 803-809

A VHDL Implementation of Low Power Virus Detection Processor

D.Sridhar[1], P. Manohar Rao[2]
[1](Electronics and Communication Engineering, SVIET/ JNTUK, INDIA)
[2](Electronics and Communication Engineering, M.P.R.M.Polytechnic/ SBTET,A.P, INDIA)

ABSTRACT
Nowadays, mobile handsets combine the functionality of mobile phones and PDAs. Unfortunately, mobile handsets development process has been driven by market demand, focusing on new features and neglecting security. So, it is imperative to study the existing challenges that facing the mobile handsets threat containment process, and the different techniques and methodologies that used to face those challenges and contain the mobile handsets malwares. A TCAM-basedvirus-detection unit provides high throughput, but also challengesfor low power and low cost. In this paper, an adaptively dividable edual-port BiTCAM (unifying binary and ternary CAMs) isproposed to achieve a high-throughput, low-power, and low-costvirus-detection processor for mobile devices. The proposed dual-port BiTCAM is realized with the dual-port AND-type match-line scheme which is composed of dual-port dynamicAND gates. The dual-port designs reduce power consumptionand increase storage efficiency due to shared storage spaces.In addition, the dividable BiTCAM provides high flexibility for regularly updating the virus-database

Keywords: ***Mobile, malware,security, malicious programs,virus detection,Associative memories,cojntent addressable Memory.***

I. INTRODUCTION

In this paper, we present Content addressable memory (CAM) can be the data stored in the memory. Due to the parallel operations, the CAM is useful for applications that demand high data-search speed, such as data base access [1], image processing [2], and IP address lookup [3]. 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.

On-demand scanning (ODS)• On-access scanning (OAS) • Memory consumption• Registry scanning • Battery life • Boot time, • Script scan URL exclusions interface • Added capacity for user-defined unwanted programs, A large number of mobile devices are now part of everyday use. These include cell phones, smartphones and PDAs (Personal Digital Assistants). The functionality and applications offered by current day mobile devices are beginning to rival those offered by a traditional PC. These mobile devices are usually have some form of connectivity (e.g., GSM, GPRS, Bluetooth, WiFi). These devices have vulnerabilities like PCs, but also have some peculiarities of their own. Worms and viruses, and other malicious software have been released that exploit vulnerabilities in some of these devices. These malware can cause harm or annoyance to the users of the mobile devices.

Over the past few years, there has been a substantial increase in the number of malware that have been written for mobile devices. As per [2], there exist at least 31 families and 170 variants of known mobile malware. Statistics [2] have shown that at least 10 Trojans are released every week. Even though it took computer viruses twenty years to evolve, their mobile device counterparts have evolved in just a span of two years. To understand the threat that is involved, we first present the comparison of the environment for PC-based and mobile device malware. 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. Intuitively, we can use CAM-based designs to achieve higher search speed. However, traditional CAM designs can not satisfy the other requirements, including low cost, low power, and high programmability, to realize a good SoC integration. Therefore we developed a virus-detection processor for mobile applications, wherein the key component is an adaptively dividable dual-port BiTCAM (unifying binary and ternary CAMs) [9]

Associative memory

A memory that is capable of determining whether a given datum – the *search word* – is contained in one of its addresses or locations. This may be accomplished by a number of mechanisms. In some cases parallel combinational logic is applied at each word in the memory and a test is made simultaneously for coincidence with the search word. In other cases the search word and all of the words in the memory are shifted serially in synchronism; a single bit of the search word is then compared to the same bit of all of the memory words using as many single-bit coincidence circuits as there are words in the memory. Amplifications of the associative memory technique allow for masking the search word or requiring only a "close" match as opposed to an exact match. Small parallel associative memories are used in cache memory and virtual memory mapping applications.since parallel operations on many words are expensive (in hardware), a variety of stratagems are used to approximate associative memory operation without actually carrying out the full test described here. One of these uses hashing to generate a "best guess" for a conventional address followed by a test of the contents of that address. A data-storage device in which a location is identified by its informational content rather than by names, addresses, or relative positions, and from which the data may be retrieved. Also known as associative storage. Recalling a previously experienced item by thinking

of something that is linked with it, thus invoking the association.

Content-addressable memories

Content-addressable memories (CAMs) 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.

We survey recent developments in the design of large-capacity content-addressable memory (CAM). 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[4]. In this paper, we review CAM- design techniques at the circuit level and at the architectural level. At the circuit level, we review low-power match line sensing techniques and search line driving approaches. At the architectural level we review three methods for reducing power consumption.

However, these designs are not suitable for mobile devices mainly because of two drawbacks. First, due to their hardware limitation, they are aimed at data matching with only a few thousands of network patterns in SNORT [7] (an open source network intrusion prevention system). They are also not scalable to perform any antivirus scanning, since the number of virus patterns is one order larger than SNORT. The ClamAV [8] (an open source antivirus software) releases more than 20,000 virus patterns and that number is still increasing. Second, these designs store all the virus patterns in the on-chip memory for achieving high throughput. 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. Intuitively, we can use CAM-based designs to achieve higher search speed. However, traditional CAM designs can not satisfy the other requirements, including low cost, low power, and high programmability,to realize a good SoC integration.Therefore we developed a virus-detection processor for mobile applications, wherein the key component is an adaptively dividable dual-port BiTCAM (unifying binary and ternary CAMs) [9].

Mobile viruses

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. The virus only becomes active if the recipient accepts the file and then installs it. Once installed, the virus displays the word "Caribe" on the device's display. Each time an infected phone is turned on, the virus launches itself and scans the area for other devices to send itself to. The scanning process is likely to drain the phone's batteries. Cabir can be thought of as a hybrid virus/worm: its mode of distribution qualifies it as a network worm, but it requires user interaction like a traditional virus.

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. This additional state is typically implemented by adding a mask bit ("care"or "don't care" bit) to every memory cell. The combination of both binary and ternary is known as Bit CAM.

II.VIRUS-DETECTION PROCESSOR

The key idea of our proposed virus-detection processor 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. 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 as shown in Fig1(a)). Only important filtering signatures and skip data are stored on the chip. In order to reduce the on-chip memory, the filtering engine operates only on the fixed amount of the memory, including a 16-KB TCAM and a 8.5-KB SRAM.These filtering data are extracted from the entire virus database by pre-processing the 30K virus patterns released from the ClamAV. Fig. 1(b) shows the operation principle of the virus-detection processor. 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 in Fig. 1(c) and (d).

Conventional Process

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 (so-called *no-plane*). 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 datum is matched with an entry of *no-plane*, the input string will be skipped according to the shift count stored in the shift SRAM

Yes plane Structure

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, as shown in Fig. 1(d). Then we 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 **exactly-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.

The *yes-plane* TCAM[16] to reduce more exact comparisons. 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 exactly-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. The off-chip memory needs roughly 8MB to store the entire 2MB virus patterns of the ClamAV .

Our idea is to merge these two single-port TCAMs into a single rectangular dual-port TCAM and concurrently match with the whole prefix. To achieve this goal we need a dual-port TCAM and two SRAMs as shown in the right part of FIG, with a division line inserted in the dual-port TCAM array to separate the *no-plane* entries and the *yes-plane* entries. With the proposed dual-port TCAM, the ternary cells storing "X" terms can be minimized, and consequently both the total memory capacity and the power consumption are reduced It includes two single-port TCAMs and two SRAMs. One TCAM serves as the *no-plane*

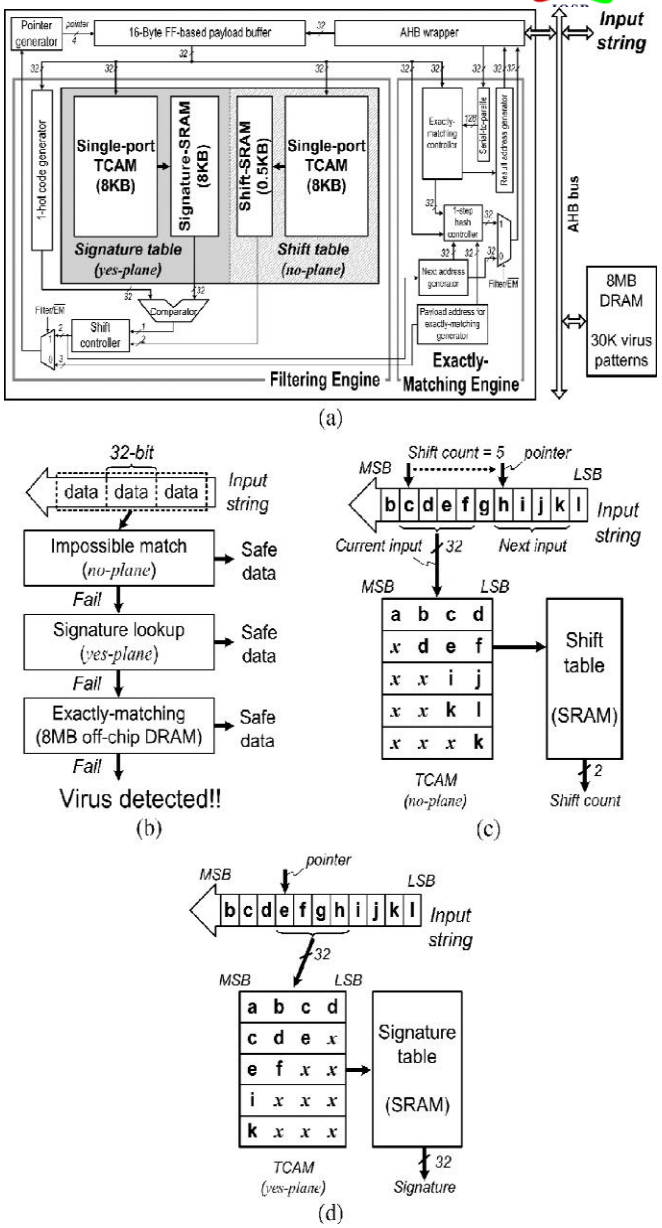

Fig..1.The architecture of the proposed virus-detection processor. (a) System architecture, (b) input data matching flow, (c) no-plane TCAM, and (d) yes-planeTCAM.

Our idea is to merge these two single-port TCAMs into a single rectangular dual-port TCAM and concurrently match with the whole prefix. . To achieve this goal we need a dual-port TCAM and two SRAMs as shown in the right part of FIG, with a division line inserted in the dual-port TCAM array to separate the *no-plane* entries and the *yes-plane*, To achieve this goal we need a dual-port TCAM and two SRAMs as shown in the right part of Fig.3,

with a division line inserted in the dual-port TCAM array to separate the *no-plane* entries and the *yes-plane* the ternary cells storing "X" terms can be minimized, and consequently both the total memory capacity and the power consumption are reduced It includes two single-port TCAMs and two SRAMs. One TCAM serves as the *no-plane*

Look up table (or the shift table) and the other as the yes-plane look-up table (or the signature table). Since tail "don't care" (represented as "X " hereafter) bits are unnecessary in reality, The left part of Fig. illustrates a design example for realizing the memory part of the virus-detection processor. It includes two single-port TCAMs and two SRAMs. One TCAM serves as the no-plane look-up table (or the shift table) and the other as the yes-plane look-up table (or the signature table). Since tail "*don't care*" (represented as "X" hereafter) bits are unnecessary in reality, the prefix of the *yes-plane* will be aligned to the left side of the TCAM array, and the TCAM entries in the *yes-plane* are sorted to form a triangle. Similarly, the *no-plane* TCAM forms another symmetrical triangle. The *no plane* is matched with the bits starting from the LSB and that the *yes-plane* is matched with the MSBs.

The DP-AND gate has two evaluation paths sharing the same pull-down network (PDN) to generate two outputs, *out1* and *out2*. Actually only one output, either *out1* or *out2*, will be determined by the logic evaluation result of the PDN by pre-programming the *set* cell, a 6-T SRAM cell, once the rule set is finalized. In other words, the *set* cell is used to determine the division and should be initialized when the data for the CAM cells are stored. A *set* cell with "0" determines that the CAM cells contributing to the PDN belong to the *yes-plane*. Otherwise, these CAM cells belong to the *no-plane*. If the DP-AND is set to the *yes-plane*, M1 of the dynamic circuit is turned on and the comparison result of the CAM cells will be sent to *out1* through M1. On the other hand, M2 will be turned off for breaking the propagation of the search result and M4 will be turned on.

However, the partition of two CAM entries should not be fixed because the virus-detection application demands the flexibility of updating the contents of the look-up tables. To satisfy this design requirement, the division line should be adaptively adjusted according to different rule sets

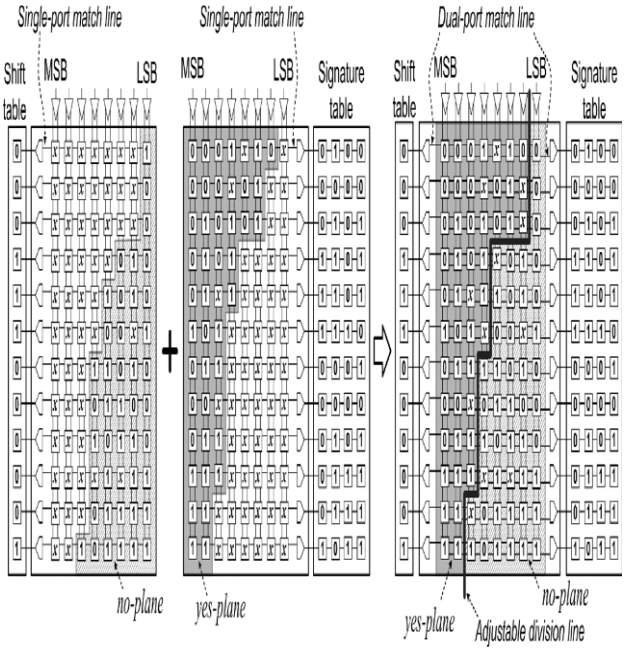

Fig .2.The design Concept of the proposed dual-port TCAM.

The proposed dual-port match-line scheme reduces the transistor count and savings in power consumption[12] compared to the single-port match-line scheme. The design of the adjustable division line provides high flexibility for updating virus databases and the resultant combination is shown clearly in the Fig.3.

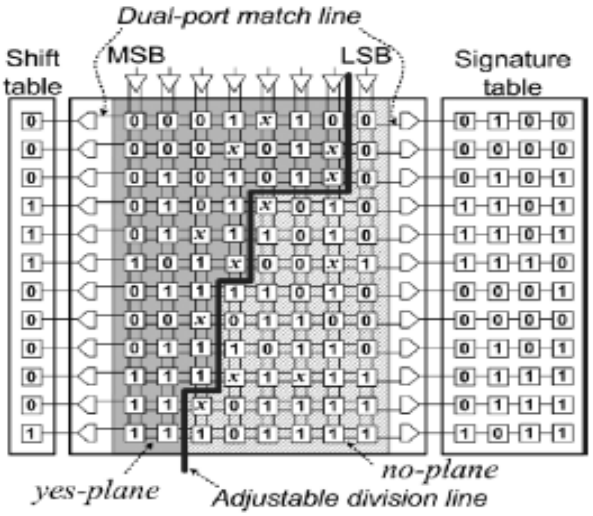

Fig .3. Dual Port TCAM.

According to the characteristics of the virus patterns in ClamAV, we obtain some rules for constructing the CAM'scontents.

1) The data width of the *no-plane* is at most 24 bits. 2) The data width of the *yes-plane* is between 8 and 32 bits, and the bit width of the binary data in the *yes-plane* is set to 8.

3) The number of entries in the *no-plane* is less than 2048. Therefore, the total bit width of those entries not used by the *no-plane* can be fully used to store data of the *yes-plane*.
4) If the summation of the bit width of the *yes-plane* and that of the *no-plane* is larger than 32 for a particular entry to be inserted into the CAM array, either the data for the *yesplane* or that for the *no-plane* will be cut short, depending on which solution cause less sacrifice in the filtering rate.

The Complete Memory Block

The complete memory block is shown in Fig. 4.The memory block is divided into 4 memory banks. Each bank contains one 512X 32b BiTCAM,one 512X 2b shift SRAM, and one 512X32b signature SRAM.
The *yes-plane* and *no-plane* triangles are located on the left and the right part BiTCAMbankrespectively. The two SRAMs and the BiTCAM share one address decoder for reducing the hardware cost. Owing to the circuit structure, the search operation of each match-line of the *yes-plane* goes from the left-most CAM cell to the right, and the outputs from the right port of the BiTCAM should be used as the word-lines for the signature-SRAM. The match result of each match-line is OR'ed with the corresponding output of the address decoder to become the word-line of the signature-SRAM. Similarly, the search operation of the *no-plane* begins from the right-most CAM cells, and the outputs from the left port of the

BiTCAM should be used as the word-lines for the shift-SRAM. The match results are also OR'ed with the corresponding output of the address decoder to become the word-lines of the shift SRAM on the left. Hit-detection circuits are added to determine any un-match cases. The symbol "SC" in Fig. 3 stands for the conventional 6T SRAM cell, while "B" and "T" represent 4-bit *d*ual-*p*ort dynamic *A*ND (DP-AND) gates for binary and ternary CAM cells, respectively. Each 32b match-line circuit is constructed with eight DP-AND gates.

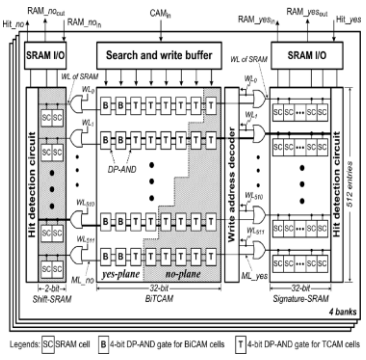

Fig. 4.Memory organization in the virus-detection processor.

III. SIMULATION AND SYNTHESIS RESULTS AND ANALYSIS

Generally in our virus detection processor for detecting virus patterns no plane and yes plane structures are used. In no plane structure we most regular virus patterns are defined And in yes plane structure, next regular virus patterns are defined.The input data of virus detection processor does matching operation .If the input data is not matched with the predefined virus patterns in the conventional method, (proposed virus detection processor) then it gives as the no virus detected and the simulation result of convetional method is shown in the Fig 5.

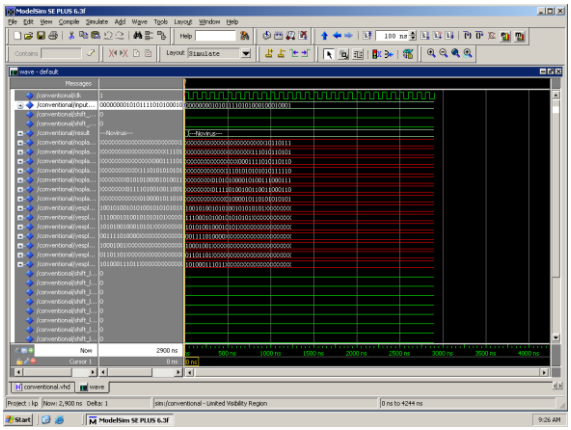

Fig .5. Simulation Result of conventional method(proposed virus detection processor) when no virus case.

The input data of virus detection processor is performing matching operation if the input data is not matched with the predefined virus patterns in the conventional method (no plane and yes plae are defined seaparately) then it gives as the virus detected and the simulation result of convetional(no plane and yes plae are defined seaparately) method is shown in the Fig 6.

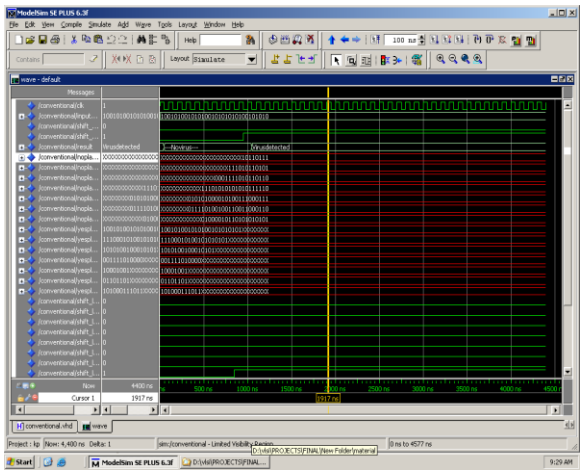

Fig .6.Simulation Result of conventional method(proposed virus detection processor)when virus detected

By using the Proposed virus detection processor to reduce the areaof the chip and power consumption of the virus detection processor. The input data is not matched with the predefined virus patterns in the conventional_exact method [proposed method] (dual port TCAM). when we using the proposed method if the input data doesn't contain any virus information the processor can be represented as no virus, The simulation result of proposed method is shown in the Fig 7.

If the input data is not matched with the predefined virus patterns in the conventional_exact method[proposed method] (dual port TCAM) if the input data doesn't contain any virus information the processor can be represented as virus detected the simulation result of proposed method is shown in the Fig 8.

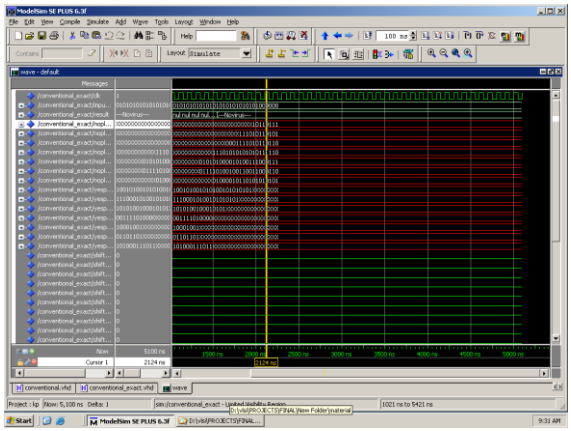

Fig .7.Simulation Result of conventional_exact method [proposed method] (dual port TCAM)when no virus detected.

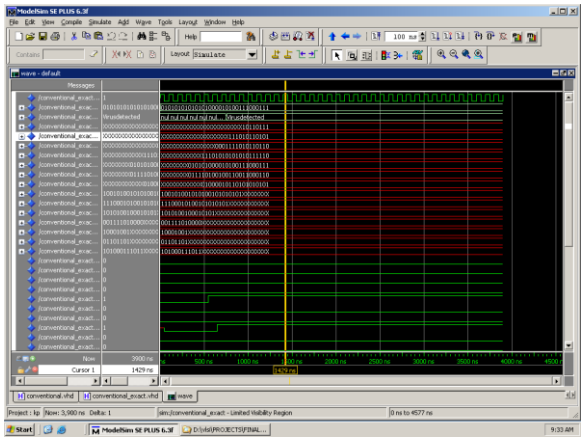

Fig.8.Simulation Result of conventional_exact method [proposed method] (dual port TCAM) when virus detected

.

The design summary of conventional method) shown in the Fig 9.

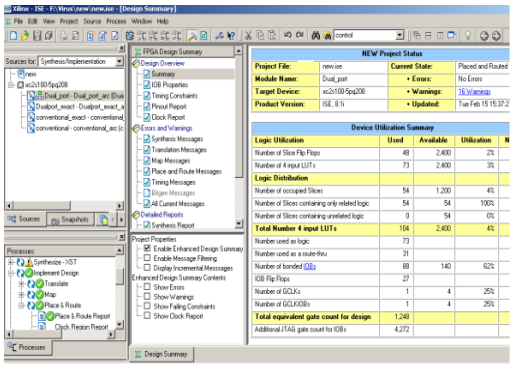

Fig .9.Synthesized design summary of conventional method(proposed virus detection processor).

The power report of conventional method(proposed virus detection processor)shown in the Fig.10

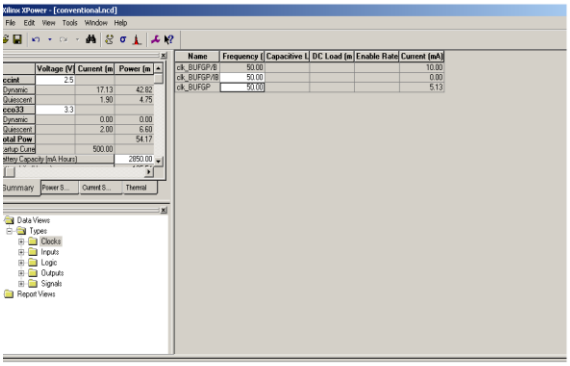

Fig . 10. power report of conventional method (proposed virus detection processor).

By using Xilinx –ISE The design summary of Dual port method (dual port TCAM) is shown in the Fig 11.

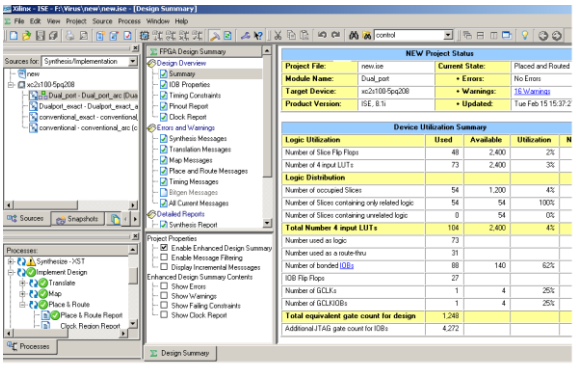

Fig 11:Synthesized design summary of dual port method(dual port TCAM).

The power report of dual port method (proposed method i.e no plane and yes plane are defined combinedly). Shown in the Fig.12.

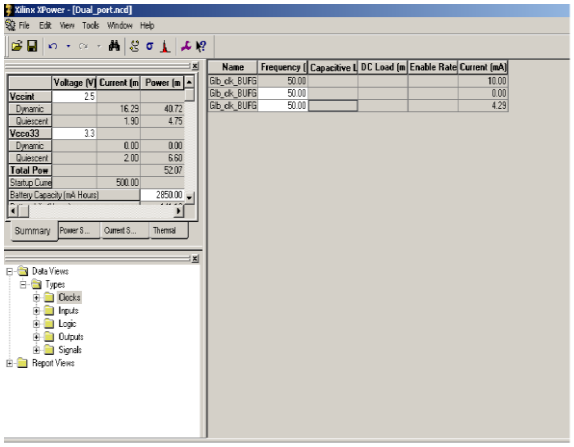

Fig 12: power report of of dual port method(proposed method i.e dual port TCAM).

IV. CONCLUSION

In this paper we describe a novel architecture for pattern matching virus detection processor for network intrusion detection system. The virus detection -processor is RAM-based design which is used to store the more virus patterns to find the virus patterns .the dual port BiT CAM is efficient pattern matching engine is capable of detecting more virus patterns . Since the patterns are programmed into the co-processor with software, the architecture can be used to implement designs in FPGA as well as ASIC We have shown that our pattern filter is capable of yield ing performance that surpasses the most recent FPGA implementations while enabling the users to program it with out having to regenerate and reconfigure the hardware. Such quick configuration may become critical, as the rate of emergence of new attack increase.

V.REFERENCES

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

[2] T. Ikenaga and T. Ogura, "A fully parallel 1-MbCAMLSI for real-time pixel-parallel image processing," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, pp. 536–544, 2000.

[3] N. F. Huang, W. E. Chen, J. Y. Luo, and J. M. Chen, "Design of multifield IPv6 packet classifiers using ternary CAMs," in *Proc. IEEE Int. Conf. Global Telecommunications*, 2001, vol. 3, pp. 1877–1881.

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

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

[6] M. Yadav, A. Venkatachaliah, and P. D. Franzon, "Hardware architecture of a parallel pattern matching engine," in *Proc. IEEE Int. Symp.Circuits and Systems*, 2007, pp. 1369–1372.

[7] Snort Users Manual 2.8.1. [Online]. Available: http://www.snort.org/ docs/snortht manuals/htmanual 281/

[8] About ClamAV. 2008 [Online]. vailable: http://www.clamav.org/about/

[9] C. C. Wang, C. J. Cheng, T. F. Chen, and J. S. Wang, "An adaptively dividable dual-ported BiTCAM for virus detection processors in mobile devices," in *IEEE Int. Solid-State Circuits Conf. Dig.*, 2008, pp.390–391.

[10] S. Wu and U. Manber, "A fast algorithm for multi-pattern searching," Univ. Arizona, Report TR-94-17, 1994.

[11] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[12] C. C. Wang, J. S. Wang, and C. W. Yeh, "High-speed and low-power design techniques for TCAM macros," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 530–540, Feb. 2008.

[13] 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, pp. 464–610.

[14] TSMC 0.13□mLogic 1P8M Salicide CU FSG 1.2V/3.3V Process Documents, Taiwan Semiconductor Manufacturing Co., Ltd..

VI. ABOU THE AUTHOR

D. Sridhar received the M.Tech degree in VLSI SYSTEM DESIGN from Avanthi Institute of Engineering and Technology ,Narsipatnam , B.Tech degree in Electronics and communication Engineering at Gudlavalleru. He has total Teaching Experience (UG and PG) of 5 years.He has guided and co-guided 4 P.G students .His research areas included VLSI system Design ,Digital signal Processing,Embedded Systems.

P.Manohar Rao received the M.Tech degree in VLSI SYSTEM DESIGN from Avanthi Institute of Engineering and Technology ,Narsipatnam ,B.Tech degree in Electronics and communication Engineering,and presently working as a principal in M.P.R.M.polytechnic college area of intrest VLSI system Design.