ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

# A Virus Detection Process for Embedded Network Security with Low Power & Area Efficiency

Jagadeesh.C[1], Kalaiselvan.I[2], Karthik.B[3], Mohamed Wafiq.S[4], R.Poovendran[5]

UG Scholar, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India [1,2,3,4]

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India [5]

**ABSTRACT**: In this project, an adaptively dividable dual-port BiTCAM (unifying binary and ternary CAMs) and Bloom filter is designed and area, power and accuracy is being compared. The dual-port BiTCAM is realized with the dual-port AND-type match-line scheme which is composed of dual-port dynamic AND gates. The dual-port designs reduce power consumption and increase storage efficiency due to shared storage spaces. Bloom Filter algorithm is being executed for the advanced applications so that this two methods is being compared and then executed. Network security for mobile devices is in high demand because of the increasing virus count. Since mobile devices have limited CPU power, dedicated hardware is required, which will be yield by our two techniques is essential to provide sufficient virus detection performance.

**KEYWORDS**: Ternary, Dual port, Virus, Unifying, CAM, Dynamic Power, Exactly Matching Engine, Bloom Filter.

## I. INTRODUCTION

Mobile malware is malicious software that targets mobile phones or wireless-enabled Personal digital assistants (PDA), by causing the collapse of the system and loss or leakage of confidential information. As wireless phones and PDA networks have become more and more common and have grown in complexity, it has become increasingly difficult to ensure their safety and security against electronic attacks in the form of viruses or other malware. Cell phone malware were initially demonstrated by Brazilian software engineer Marcos Velasco. He created a virus that could be used by anyone in order to educate the public of the threat. The first known mobile virus, "Timofonica", originated in Spain and was identified by antivirus labs in Russia and Finland in June 2000. "Timofonica" sent SMS messages to GSM mobile phones that read (in Spanish) "Information for you: Telefónica is fooling you." These messages were sent through the Internet SMS gate of the MoviStar mobile operator. In June 2004, it was discovered that a company called Ojam had engineered an anti-piracy Trojan virus in older versions of its mobile phone game, Mosquito. This virus sent SMS text messages to the company without the user's knowledge. Although this malware was removed from the game's more recent versions, it still exists in older, unlicensed versions, and these may still be distributed on file-sharing networks and free software download web sites.

In July 2004, computer hobbyists released a proof-of-concept mobile virus Cabir that replicates and spreads itself on Bluetooth wireless networks and infects mobile phones running the Symbian OS. In March 2005, it was reported that a computer worm called Commwarrior-A had been infecting Symbian series 60 mobile phones. This specific worm replicated itself through the phone's Multimedia Messaging Service (MMS), sending copies of itself to other phone owners listed in the phone user's address book. Although the worm is not considered harmful, experts agree that it heralded a new age of electronic attacks on mobile phones. In August 2010, Kaspersky Lab reported a trojan designated Trojan-SMS, Android OS, Fake Player. This was the first malicious program classified as a Trojan SMS that affects smartphones running on Google's Android operating system, and which had already infected a number of mobile devices, sending SMS messages to premium rate numbers without the owner's knowledge or consent, and accumulating huge bills.Currently, various antivirus software

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

companies like Trend Micro, AVG, avast!, Comodo, Kaspersky Lab, PSafe, and Softwin are working to adapt their programs to the mobile operating systems that are most at risk. Meanwhile, operating system developers try to curb the spread of infections with quality control checks on software and content offered through their digital application distribution Plat forms, such as Google Play or Apple's App Store. Recent studies however show that mobile antivirus programs are ineffective due to the rapid evolution of mobile malware.

## II.EXISTING SYSTEM

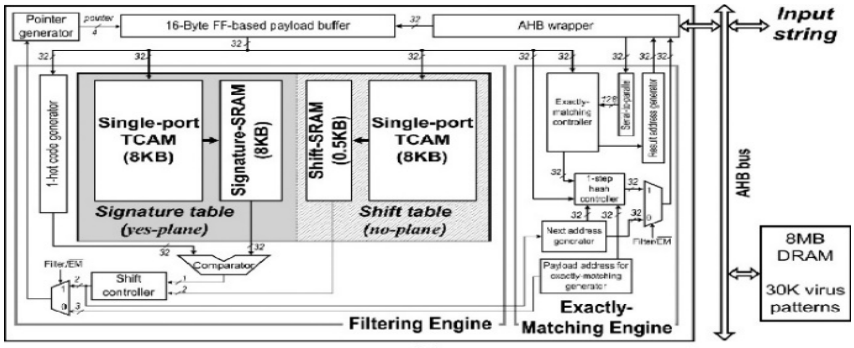



Fig.1 virus detection processor

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 ternary cells storing “X” not only waste hardware cost but also waste energy. 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. In order to reduce power consumption, we must be aware that a TCAM macro consumes power mainly for three parts: the clock driver, match lines, and search lines. Due to the advancement of match-line circuit techniques, the power consumption of both match-line circuit and clock driver have been greatly reduced. According to the data published in the search time and the energy index breakdown normalized at 1.2 V 0.1 m technology are shown. We find that the power consumption of search lines occupies about 54%, 71%, and 82% of the total power consumption of the CAM de-signs respectively. To reduce the power consumption for search lines, this study says that segmented search-line technique for the TCAM macro in the application of IP address lookup. In the following, we will first describe the attributes of TCAM for IP address lookup, and then describe the design of the segmented search-line circuitry. 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. 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.

The ternary cells for storing those leading bytes can be replaced by binary cells for further area and power reduction, since usually a binary cell is simpler than a ternary cell. Thus, it is named as a BiTCAM, since it combines binary and ternary cells. We obtain some rules for constructing the CAM’s contents. 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-

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

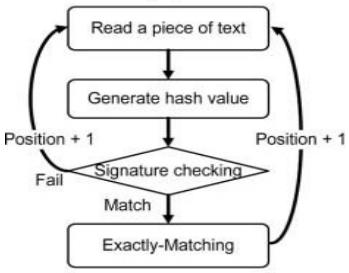
Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

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 yes- plane or that for the no-plane will be cut short, depending on which solution cause less sacrifice in the filtering rate

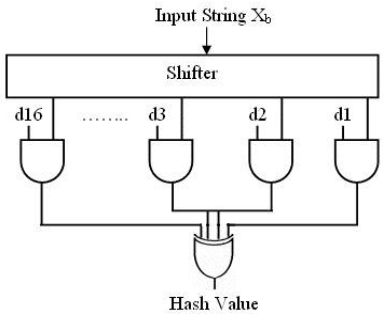
## III. PROPOSED SYSTEM

A Bloom filter is a space-efficient data structure used to test whether an element exists in a given set. This algorithm is composed of different hash functions and a long vector of bits. Initially, all bits are set to 0 at the pre processing stage. To add an element, the Bloom filter hashes the element by these hash functions and gets positions of its vector. The Bloom filter then sets the bits at these positions to 1. The value of a vector that only contains an element is called the signature of an element. To check the membership of a particular element, the Bloom filter hashes this element by the same hash functions at run time, and it also generates positions of the vector.




**ALGORITHM FOR BLOOM FILTER**

It describes a typical flow of pattern matching by Bloom filters. This algorithm fetches the prefix of a pattern from the text and hashes it to generate a signature. Then, this algorithm checks whether the signature exists in the bit vector. If the answer is yes, it shifts the search window to the right by one character for each comparison and repeats the above step to filter out safe data until it finds a candidate position and launches exact-matching.




**PRODUCING HASH VALUE**

**Pseudo-code for hash function**

For each signature:
i. Generate as many random numbers as the bits in the signature
ii. Left shift the signature to get to the specified bit
iii. AND each shifted signature with the random number
iv. XOR all the results of AND

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

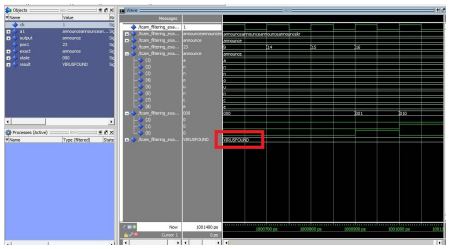
13th-14th March 2017

Organized by

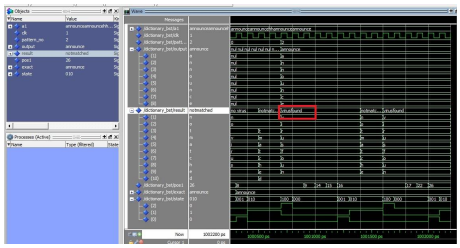
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

## IV. EXPERIMENTAL RESULTS

### Simulation of Existing



### Simulation of Proposed



### Existing Delay

```
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
  IBUF:I->O           20   0.003   0.618  a1_6__7_IBUF (a1_6__7_IBUF)
  LUT4:I1->O           1   0.061   0.566  Mmux_n0603883 (Mmux_n0603883)
  LUT6:I2->O           1   0.061   0.512  Mmux_n0603885 (Mmux_n0603886)
  LUT5:I0->O           1   0.061   0.512  Mmux_n0603886 (Mmux_n0603887)
  LUT6:I3->O           2   0.061   0.517  Mmux_n0603889 (n0603c7>)
  LUT6:I3->O           9   0.061   0.741  GND_6_o_GND_6_o_AND_10_o11 (GND_6_o_GND_6_o_AND_10_o1)
  LUT6:I0->O           6   0.061   0.594  GND_6_o_GND_6_o_AND_11_o11 (GND_6_o_GND_6_o_AND_11_o1)
  LUT6:I2->O           3   0.061   0.706  GND_6_o_GND_6_o_AND_34_o1 (GND_6_o_GND_6_o_AND_34_o)
  LUT6:I1->O           1   0.061   0.694  Maccum_pos_lut<0>1 (Maccum_pos_lut<0>1)
  LUT6:I1->O           1   0.061   0.426  Maccum_pos_lut<0>2 (Maccum_pos_lut<0>2)
  LUT6:I4->O           1   0.061   0.426  Maccum_pos_lut<0>3 (Maccum_pos_lut<0>3)
  LUT5:I3->O           1   0.061   0.566  Maccum_pos_lut<0>4 (Maccum_pos_lut<0>4)
  LUT6:I2->O           1   0.061   0.426  Maccum_pos_lut<0>5 (Maccum_pos_lut<0>5)
  LUT6:I4->O           4   0.061   0.583  Maccum_pos_lut<0>6 (Maccum_pos_lut<0>)
  LUT6:I2->O           3   0.061   0.524  Maccum_pos_xor<5>121 (Maccum_pos_xor<5>12)
  LUT4:I1->O           1   0.061   0.000  Maccum_pos_xor<5>11 (Result<5>)
  FDE:D               -0.002              pos_5
----------------------------------------  ------------
Total                       9.331ns (0.918ns logic, 8.413ns route)
                                    (9.8% logic, 90.2% route)
```

### Proposed Delay

```
Cell:in->out      fanout   Gate    Net    Logical Name (Net Name)
                           Delay   Delay
----------------------------------------  ------------
  FDE:C->Q             2   0.317   0.699  pos_5_1 (pos_5_1)
  LUT5:I0->O          16   0.061   0.509  Mmux_n1258111 (Mmux_n1258111)
  LUT6:I4->O           1   0.061   0.426  Mmux_n1259662 (Mmux_n1259661)
  LUT6:I4->O           3   0.061   0.578  Mmux_n1259668 (n1259<5>)
  LUT4:I0->O           1   0.061   0.357  GND_6_o_GND_6_o_AND_10_o11_SW0_SW0 (N246)
  LUT6:I5->O          15   0.061   0.435  GND_6_o_GND_6_o_AND_162_o11 (GND_6_o_GND_6_o_AND_162_o1)
  LUT6:I5->O           2   0.061   0.362  Mmux_GND_6_o_PWR_6_o_MUX_95_o143 (Mmux_GND_6_o_PWR_6_o_MUX
  LUT5:I4->O           2   0.061   0.362  Madd_pos[5]_GND_6_o_add_2928_OUT_lut<0>7_SW0_SW3 (N285)
  LUT6:I5->O           1   0.061   0.694  Madd_pos[5]_GND_6_o_add_2928_OUT_lut<0>7_SW0 (N217)
  LUT6:I1->O           4   0.061   0.443  Madd_pos[5]_GND_6_o_add_2928_OUT_lut<0>8 (Madd_pos[5]_GND_
  LUT6:I4->O           3   0.061   0.369  Mmux_GND_6_o_st[2]_mux_3137_OUT631 (Mmux_GND_6_o_st[2]_mux
  LUT6:I5->O           1   0.061   0.357  Mmux_GND_6_o_st[2]_mux_3137_OUT51 (Mmux_GND_6_o_st[2]_mux_
  LUT6:I5->O           2   0.061   0.000  Mmux_GND_6_o_st[2]_mux_3137_OUT61 (GND_6_o_st[2]_mux_3137_
  FDE:D               -0.002              pos_5
----------------------------------------  ------------
Total                       6.642ns (1.049ns logic, 5.593ns route)
                                    (15.8% logic, 84.2% route)
```

### Existing Design Summary

| TCAM_FILTERING_EXACT Project Status (01/31/2017 - 10:22:39) | | | |
|---|---|---|---|
| Project File: | aa.xise | Parser Errors: | No Errors |
| Module Name: | TCAM_FILTERING_EXACT | Implementation State: | Synthesized |
| Target Device: | xc6vlx565t-2ff1924 | •Errors: | No Errors |
| Product Version: | ISE 13.2 | •Warnings: | 305 Warnings (305 new) |
| Design Goal: | Balanced | •Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | •Timing Constraints: | |
| Environment: | System Settings | •Final Timing Score: | |

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slice Registers | 169 | 708480 | 0% |
| Number of Slice LUTs | 3530 | 354240 | 0% |
| Number of fully used LUT-FF pairs | 168 | 3531 | 4% |
| Number of bonded IOBs | 337 | 640 | 52% |
| Number of BUFG/BUFGCTRLs | 2 | 32 | 6% |

### Proposed Design Summary

| Device Utilization Summary | | | | |
|---|---|---|---|---|
| Slice Logic Utilization | Used | Available | Utilization | Note(s) |
| Number of Slice Registers | 95 | 708,480 | 1% | |
| Number used as Flip Flops | 95 | | | |
| Number used as Latches | 0 | | | |
| Number used as Latch-thrus | 0 | | | |
| Number used as AND/OR logics | 0 | | | |
| Number of Slice LUTs | 2,028 | 354,240 | 1% | |
| Number used as logic | 2,027 | 354,240 | 1% | |
| Number using O6 output only | 1,894 | | | |
| Number using O5 output only | 0 | | | |
| Number using O5 and O6 | 133 | | | |
| Number used as ROM | 0 | | | |
| Number used as Memory | 0 | 101,920 | 0% | |
| Number used exclusively as route-thrus | 1 | | | |
| Number with same-slice register load | 1 | | | |
| Number with same-slice carry load | 0 | | | |
| Number with other load | 0 | | | |

**Synthesis Results**

Family : Spartan 6.

Device : XC6S150

Package : PQ708
Speed : -5.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

**TABLE 1**

| PARAMETER | EXISTING | PROPOSED |
|---|---|---|
| AREA | 7581um | 6912um |
| POWER | 25mW | 19.7Mw |

## V. CONCLUSION

This paper proposed an adaptive Bloom Filter for a high-speed, low-power, Low area and low-cost virus-detection processor in mobile devices. The proposed Bloom Filter algorithm reduces the transistor count and provides savings in power consumption compared to the Dual port BiTCAM. The design of the adjustable division line provides high flexibility for updating virus databases.

## REFERENCES

[1] K.-J. Lin and C.-W. Wu, "A low-power CAM design for LZ data compression," IEEE Trans. Comput., vol. 49, pp.1139–1145, Oct. 2000.
[2] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern- matching using TCAM," in Proc. IEEE ICNP, 2004, pp. 174–183.
[3] T. Ikenaga and T. Ogura, "A fully parallel 1-Mb CAM LSI for real-time pixel-parallel image processing," IEEE J. Solid-State Circuits, vol. 35, No. 4, pp. 536–544, Apr. 2000.
[4] R. Sangireddy and A. K. Somani, "High-speed IP routing with binary decision diagrams based hardware address lookup engine," IEEE J. Sel. Areas Commun., vol. 21, no. 5, pp. 513–521, May 2003
[5] T. Hayashi and T. Miyazaki, "High-speed table lookup engine for IPv6 longest prefix match," in Proc. GLOBECOM'99, 1999, vol. 2, pp. 1586–1571.
[6] N.-F. Huang, W.-E. Chen, J.-Y. Luo, and J.-M. Chen, "Design of multi-field IPv6 packet classifiers using ternary CAMs," in Proc. GLOBECOM 2001, vol. 3, pp. 1877– 1881.
[7] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," IEEE J. Solid-State Circuits, vol. 36, no. 6, pp. 956–968, Jun. 2001.
[8] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content- addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," IEEE J. Solid-State Circuits, vol. 38, no. 1, pp. 155–158, Jan. 2003.
[9] I. Arsovski and A. Sheikholeslami, "A mismatch- dependent power al-location technique for match-line sensing in content-addressable mem- ories," IEEE J. Solid-State Circuits, vol. 38, no. 11, pp. 1958–1966, Nov. 2003.
[10] F. Shafai, K. J. Schultz, G. F. R. Gibson, A. G. Bluschke, and D. E. Somppi, "Fully parallel 30-MHz, 2.5- Mb CAM," IEEE J. Solid-State Circuits, vol. 33, no. 11, pp. 1690–1696, Nov. 1998.
[11] S. Choi, K. Sohn, and H.-J. Yoo, "A 0.7 fJ/bit/search, 2.2 ns search time, hybrid type TCAM architecture," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 254–260, Jan. 2005.