Journal of Advanced Research in Technology and Management Sciences
Volume: 01 Issue: 03 ISSN: 2582-3078 November 2019
Available online at: http://www.jartms.org

High SpeedPacket Classification Using XnorBV

[1]Anil Kumar Reddy Kannapu, [2] M.Adiseshaiah
[1] M.Tech (VLSI-SD),Dept. Of ECE
[2] M.Adiseshaiah, M.Tech, Ph.D, Assistant Professor, Dept. Of ECE
[1,2]Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India

Abstract—

Packet classification is a critical technology for secure communication and networking. Security tools and Internet services use packet sorting technology that includes checking packets against predefined rules stored in a workbook. Fulfill software solutions available for classifying unwanted and efficient processing of wired speed in high-speed networks. Ternary Content Addressable Memory (TCAM), Bit-Vector (BV), Field Sharing Bit Vector (FSBV). In this article, we recommend a simple and efficient memory method for classifying packets on the Xnor portal, instead of the lookup tables called the XnorBV approach. The packet header fields of the IP addresses and the protocol layer are categorized by the Xnor gateway against a predefined set of rules that also support bitrates '1', '0' and '*', while the port numbers in the packet header support range match the port numbers with minimum and maximum values. The proposed XnorBV architecture is independent of the ruleset properties and supports multidimensional classification.

I. INTRODUCTION

A series of packets arriving from a source system to a destination system is generally referred to as traffic or packet flow, and a series of packets from a given source to a particular destination is called a flow. Flow can be identified by a method called packet classification, which classifies incoming packets into another process by examining the packet header fields at a given time. To select and

The order of the packets is in a different process, each incoming packet is checked against a set of rules, if we match the inbound packet with a rule of a set of rules, we accept only those. Otherwise they were denied. After

incoming packages are classified into different categories, each process can be handled separately to differentiate the services offered to the user. Packages of the same category must be used for all applications and services requested by the user. Packet classification technology effectively provides the appropriate packets of affected services with a predefined set of rules. Many services, such as firewalls, VPN, network security, policy-based routing, traffic configuration, and quality of service, have been integrated into packet classification technology to detect threats and prevent unauthorized access to the network. Because of these many benefits of packet-classing technology, packet-classing has become an integral part of all types of intrusion detection systems - firewalls - Internet routers and VPNs in modern communications.

Software solutions are available for classifying to packets, from which they are not sufficient for high speed network applications. In To software tools, classification is usually done only by checking to port numbers, IP addresses or to protocol layers. In order to take into account the speed of the To cable, it is desirable to execute software solutions that support multiple area controls.

Sometimes hardware speeds are desirable for managing secure networks, and packets can be renounced by checking all packet headers. In a hardware packet classification solution, multiple fields of an incoming packet are checked with each set of rules. To rule set size can be validated from 100 to thousands. Also, the challenge of implementing a packet classification system hardware is the amount

of memory needed to store many rules. They sent rules to to memory sources using programmable logic arrays (FPGAs), from the chip's limited memory resources to the problem of storing many rules. When classifying To packets, the rules are stored in descending order of priority, and the measures are ranked in order. Figure 1 below shows a standard 5-part packet header, which includes the IP address of the cell to force is the cell and the number port to the protocol field. More sets of fields require different matching, for example, the source also prefixes the source IP address field, matches the source and celport field ranges, and some protocol fields.

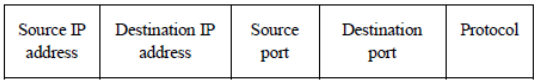


Figure 1: Standard 5-tuple packet header

II.PACKET CLASSIFICATION

The important issue in the structure of the package classification is the power consumption. Because trillions of bits per second are produced by routers, power consumption becomes a critical concern. Energy efficiency depends on the number of rules used to classify the incoming beam. This is one aspect used to evaluate the energy efficiency of a packet classification system. The energy that the router consumes to keep out the very high heat generated by the router components greatly helps in operating costs. The power consumption of search engines has become an increasingly important evaluation parameter because each router port contains packet rating and router search devices.

Memory requirements are another important issue of package classification. Nowadays, researchers aim to find solutions to large rules. The classification method and the number of rules stored in the workbook are related to the amount of memory required. Due to limited resources available on FPGA, memory has become a very important issue for hardware resolution to support a large number of rules.

Speed and flexibility in specifications are problems with packet classification devices. In the package classification process, packages are grouped based on a set of predefined rules, also called package filters. Rules or filters specify patterns that must be matched with incoming packets to arrange packets for different flows. Packet filters or rules define possible values for each field with a standard 5-crown packet head. Address fields in the package header often use prefixes to specify addresses, although arbitrary bit masks in address fields are acceptable in a workbook or set of rules, and this feature is widely used in real filter sets. Rules or filters specify a range value for packet header port fields to match incoming packets. Protocols can be in two ways either the exact value or as a wildcard character. The values specified by bit masks are allowed in some system for the incoming packet protocol field, even if it is not clear how appropriate this feature is.

III.RELATED WORK

Efficient and desirable ways of implementing instruments on a large scale can be categorized into two decision-tree and decomposition-based approaches. In the decomposition-based approach, the packages are categorized in two phases: in the first stage, independent searches are performed in each area of the packages, while in the second stage: the results of the first stage are combined. Hydrolysis-based algorithms are suitable for hardware implementation, and can maintain high throughput at low latency time. Example: bit bus (BV), aggregated bit bus (ABV), memory addressable bit-vector content bit (BV-TCAM), discrete field bit bus (FSBV), cross-production, RFC and StrideBV approach Decomposition-based. The addressable memory algorithm (BV-TCAM) and StrideBV content vectors support all types of matches and are scalable to a large number of rules in a rule set.

Ternary Content Addressable Memory (TCAM) is the desirable solution for devices due to its simple management and speed. To verify all fields simultaneously and at high speed, a search engine based on Ternary Contentable Memory (TCAM) is used. TCAM extension (TCAM) is a bit-vector-adjustable content content (BV-TCAM) that uses the TCAM method of content and vector bit approaches to support prefix, range, and exact match. For vector-oriented routing (BV-TCAM) to increase productivity and data compression pressure, this approach is generally used in intrusion detection systems where multiple matches are reported at a Gigabit correlation rate. to Further processing due to routing issues. In a vector-content-tri-memory (BV-TCAM) policy, IP addresses and the header protocol layer are matched using a bit vector policy, and port numbers are matched using the TCAM policy in parallel and ANDING results for final output. This approach supports multiple matching without using ange to convert the prefix.

The bit vector algorithm is a desirable algorithm that is widely used to implement packet classification devices. Figure 2 illustrates the bit vector algorithm, where the value of bit '1' indicates the incoming packet match against a set of base, and the value of bit '0' indicates that the incoming packet does not match a predefined set of rules. In the Bit-Vector (BV) algorithm, rules are arranged in a set of rules based on their priority. Generally to avoid the complexity of prioritizing each rule, the rules are arranged in descending order of priority. The vector bit is simple and has a low computational complexity on hardware. For the classification of multiple field packets, each field generates a bit bus, and the vector of each field goes together to obtain a final bit bus that indicates the status of the incoming packet against a set of rules as shown in Fig. 2.

IV.XNORBV ALGORITHM

In this work, each field or group of incoming packets is compiled using the XnorBV method instead of using lookup tables. In XnorBV, the Xnor gateway is used as the base comparator to compare the incoming packet with the rule set rule. The use of Xnor gate makes architecture simple and effective for a wide range of network communications involving packet filtering or packet classification. Using the XnorBV algorithm, the proposed design achieves good results on the same 300 MHz operating frequency. In the XnorBV algorithm, each field of the packet header generates a little vector which will be ANDing with a bit vector created by the Others field to obtain the final output bit vector. A final bit bus of the cryptographic unit is given priority to fetch a higher priority match rule. In the suggested method, each bit of a field is verified against each bit of a rule that is stored in a rule set by using an XNORing process. Using VERILOG behavioral modeling, the designer system supports the triple bit format of "1", "0" and "*" (wildcard input). Figure 5 illustrates the proposed XnorBV method for packet classification, with the same set of rules and field value = 1101 as with the field division bit vector (FSBV) and StrideBV method for packet classification. After the XNORing process, each bit of output obtained after XNORing is ANDing for one part indicating the base state of the incoming packet field [5]. A standard five-fold packet header that contains five fields: source IP address, destination IP address, source port number, destination port number, and protocol layer. In this paper, the classification of the IP address fields and the protocol field is performed by using the XnorBV method. The proposed XnorBV module supports the prefix and exact match of Internet Protocol (IP) addresses and protocol layer respectively.

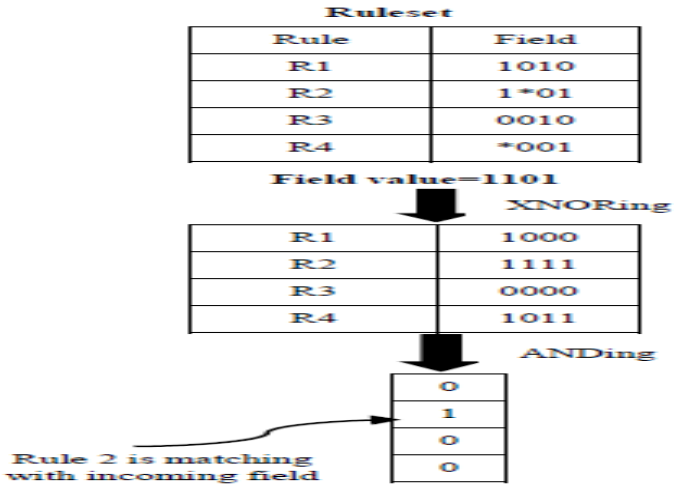


Figure5: Proposed XnorBV Algorithm

Figure 6 illustrates the circuit diagram of the proposed XnorBV method for generating bit vectors. The 5-tuple inbound packet field is checked against the N rules for a set of rules. To understand the bit vector generation using the XnorBV method with the help of the circuit diagram, let the base length and the incoming packet field k kits. Let the first rule of a rule set presented by R1 = Wk-1Wk-2 ... .. W0 and the incoming packet field given by F1 = Tk-1Tk-2… .. T0. Each part of a rule and a field is XNORing and after the XNORing process is finished, the result of the k-bit is ANDing to get one bit that indicates whether or not the field matches a rule. The same process is performed for each rule of a set of size N to obtain an N-bit bus for a given domain of the packet. Detailed algorithm to configure the bit vector and perform the packet classification below.

Algorithm 1: "Bit Vector Generation for each field of a packet using XnorBV method

Require: N rules each of which is represented as a K-bit ternary string of a field of packet: Rn=Wn k-1 Wn k-2 Wn k-3……. Wn 0, F=Tk-1 Tk-2 Tk-3 …….. T0, where= 1 …….. N

1: for n 1 to N do {Process Rn}
2: for k k-1 to 0 do
3: S[n] [k] = Wn k-1 Xnor Tk-1
4: end for
5: for b 0 to k-1 do
6: let Y=1,
7: Y = Sn b AND Y
8: end for

Algorithm 2: Packet Classification using XnorBV

Require: let the B be bit vector after comparing the incoming packet with a set of rules.

Require: let the B1, B2, B3, B4 and B5 be the bit vector of 5-tuple packet

1: for n 1 to N do {bit-wise AND}
2: V= B1 n AND B2 n AND B3 n AND B4 n AND B5 n
3: end for
4: V be the final bit-vector indicating the match of mismatch of packet with against rule of ruleset
4: V is the input to priority encoder to get highest priority matched rule
5: Vm V { Vm Output of Priority Encoder}"

To support range matching for port numbers, the field value is compared with the minimum and maximum rules. Figure 7 shows the band unit to perform band matching for packet port numbers. To make the architecture designed to support a minimum matching range, the upper limit of each rule and the method of performing the range matching are shown in Fig. 7. Figure 7 shows a set of rules with a minimum and a = 1000. Or equal to the minimum, you will give "1" or else "0" similarly if the field value is less than or equal to the upper limit and then "1" otherwise gives "0". The bit values obtained after comparing the field value with the minimum and upper limits are ANDing for one part, which indicates that the field value falls between the minimum and the upper limit. The range lookup module is used for both the source port number and the intended 16-bit port number for many applications. The proposed structure supports IP prefix matching, port matching for port numbers, and exact protocol field matching. Also, it is independent of ruleset feature and supports multi-dimensional classification

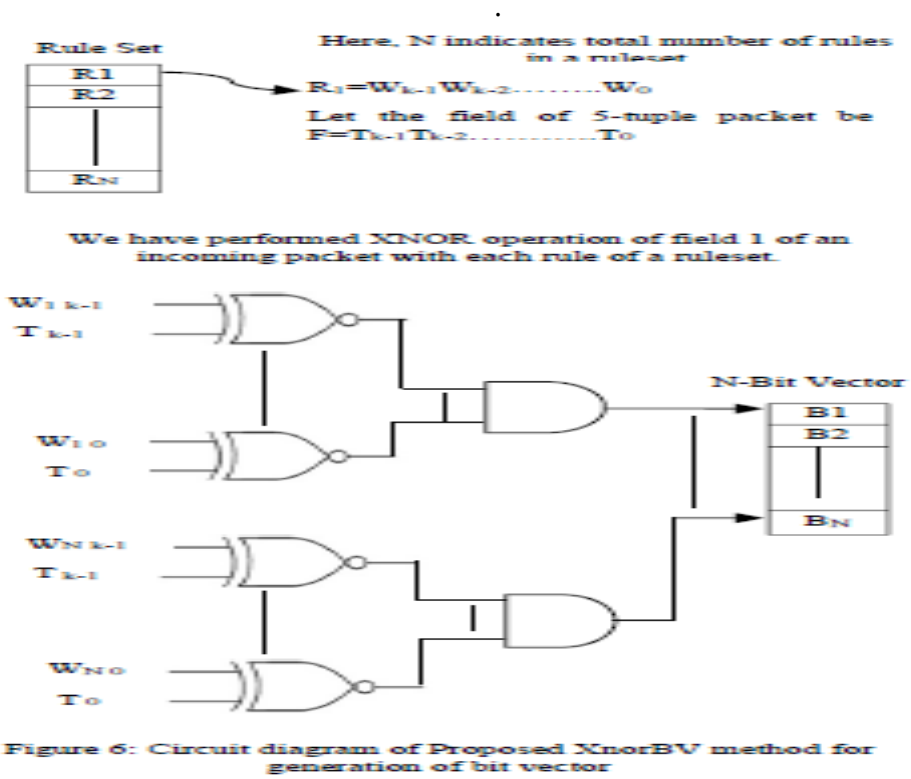


Figure 6: Circuit diagram of Proposed XnorBV method for generation of bit vector

Figure 8. shows the complete structure of packet classification that supports prefix, scope and exact match. The rules are arranged in a set of rules in descending order of priority. The structure shown in Fig. 8 carries out the 104-bit full packet header classification with multi-match packet classification. The rules for each tuple are stored separately and each tuple or the incoming packet field is checked against the respective set of rules. A five-channel packet header provides five N-bit vectors; each N-bit vector represents the state of that group as opposed to predefined rules in the rule set. After obtaining partial results for each group of packets from the assembly process, the results of the five sequences are subjected to an AND to obtain a final bit bus indicating whether the packet matches the rules defined by the rules. For IP addresses and protocol layers, you need to use the XnorBV module to perform a prefix and exact match. XnorBV supports the '0', '1' and '*' (wildcard) triple bit formats. For packet port numbers, the band unit is used to create a bit vector. In this way, the proposed

structure creates the prefix, the scope and the exact match. The priority encoder is used to determine the highest priority of the final bit bus rule and to determine the rule for further operation.

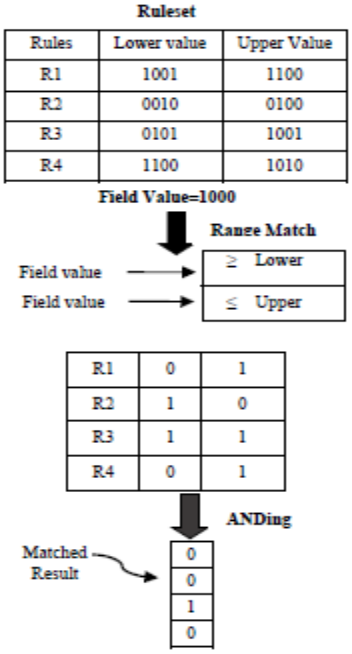


Figure 7: Range Search Module for Range Match

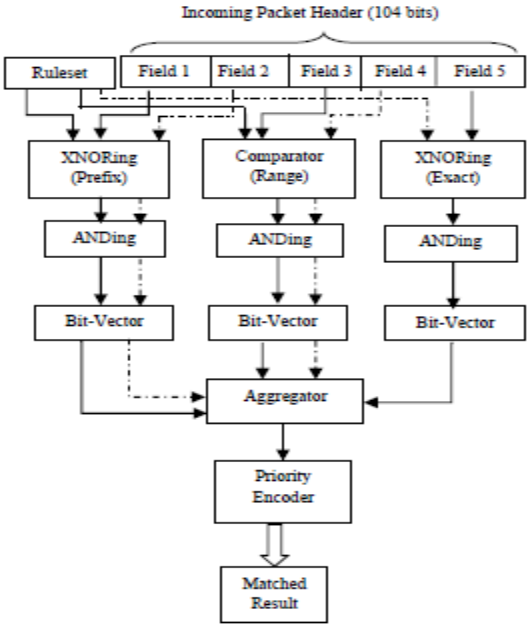


Figure 8: Proposed Architecture for Packet Classification

V.Results:

The Verilog HDL is used to design the architecture on Xilinx ISE design 12.1 suite. Design utilization summary of the architecture for the proposed XNOR BV Algorithm on SPARTAN3E FPGA trainer kit is shown below.

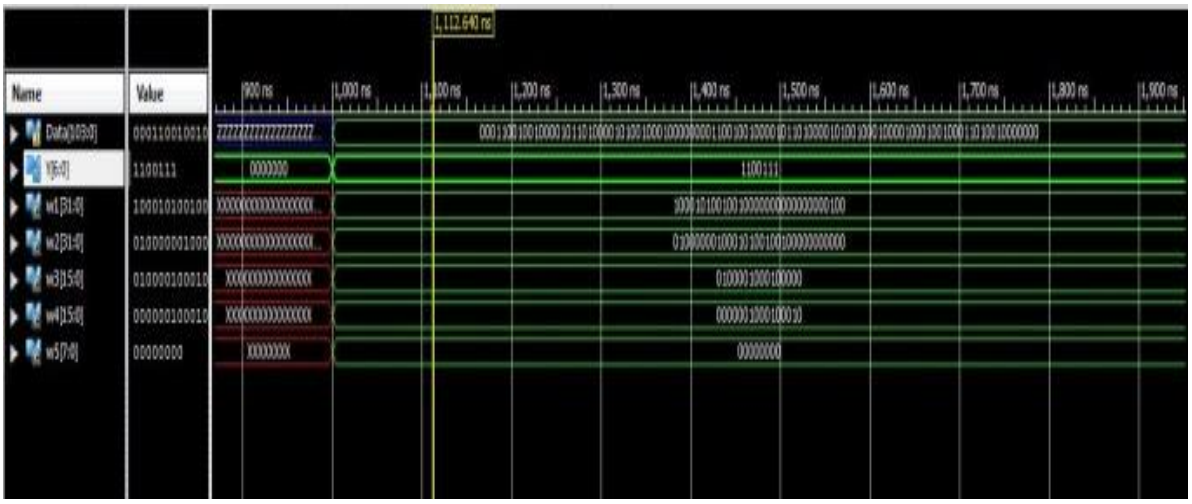
Device utilization summary:

Selected Device: XC3s500efg320-5

Number of Slices: 139 out of 4656 2%
Number of 4 input LUTs: 242 out of 9312 2%
Number of IOs : 111
Number of bonded IOBs: 111 out of 232 47%

The system latency is the time it takes to get output after the input is applied. In packet classification, the transition time is defined as the time required to complete a single classification operation. In the XnorBV method, the classification process is carried out in three stages. In the first step, there is a separation for each field of the incoming packet to classify against a set of rules to create bit vectors. In the second stage, the bit vector is summed for each field created in the first stage, that is, the partial results of obtaining the final bit vector that indicates the status of the rules versus the incoming beam. The final result obtained in the second stage is converted to a priority encoder to obtain a single match result from a multiple match result for further operation. The high priority identical rule is extracted using the Phase 3 priority encryption. In this way, the proposed XnorBV method requires three hours a clock to classify a single incoming beam. Therefore, the transition time of the proposed structure is 3 cycles around the clock which is also desirable to apply low latency.

The below fig depicts the simulation result of proposed algorithm.



VI.Conclusion:

The proposed method of XnorBV architecture using the Xilinx ISE 12.1 set selects XC3s500efg320-5, SPARTAN3E FPGA as the target device is memory-efficient, requiring 15 bytes / base lower than any other packet classification technique. The architecture supports prefix, exact match and scope without using range conversion to prefix and is independent of the rule set feature. The design of high-performance packet classification for networks has also been improved. Energy efficiency is also improved with increased power plus one base. The proposed structure can maintain high throughput at low latency, which is desirable for applications with low latency.

REFERENCES

[1] Andrea Sanny, Thilan Ganegedara, Viktor K. Prasanna; "A Comparison of Ruleset Feature Independent Packet Classification Engines on FPGA," *in 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum*, 978-0-7695- 4979-8/13 $26.00 © 2013 IEEE

[2] T. Ganegedara and V. Prasanna, ''StrideBV: 400G+ Single Chip Packet Classification,'' *in Proc. IEEE Conf. HPSR*, 2012, pp. 1-6.

[3] Mahmood Ahmadi, S. Arash Ostadzadeh, and Stephan Wong; "An Analysis of Rule-Set Databases in Packet Classification," in *18th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC 2007)*, 29-30 November 2007, Veldhoven, The Netherlands.

[4] Nekoo Rafiei Karkvandi, Hassan Asgharian, Amir Kusedghi, Ahmad Akbari, "Hardware Network packet Classifier for High Speed Intrusion Systems," in *International Journal of Engineering and Technology*; Volume 4 No.3, March, 2014.

[5] Ausaf Umar Khan, Yogesh Suryawanshi, Dr. Manish Chawhan, Sandeep Kakde, "Design and Implementation of High performance Architecture for Packet Classification," in *International Conference on Advances in Computer Engineering and Applications,* IMS Engineering College, Ghaziabad, India, page 598-602, IEEE.