Design of High Performance Packet Classification Architecture for Communication Networks

Ausaf Umar Khan[1], Dr. Manish Chawhan[2], Dr. Yogesh Suryawanshi[3] and Sandeep kakde[4]

[1]*Department of Electronics and Telecommunication, Anjuman College of Engineering and Technology, Nagpur, India.*

[2]*Department of Electronics and Telecommunication Engineering, YCCE, Nagpur, India.*

[3]*Department of Electronics and Communication Engineering, DMIETR, Wardha, India.*

[4]*Department of Electronics Engineering, YCCE, Nagpur, India*

aukhan@anjumanengg.edu.in

***Abstract*— Packet classification is a crucial technique for secure communication and networking. Security tools and internet services use packet classification technique which involves checking of packets against predefined rules stored in a classifier. The performance of the available software solutions of classification is not desirable and efficient for wire speed processing in high speed networks. Ternary Content Addressable Memory (TCAM), Bit-Vector (BV), field split bit vector (FSBV) and StrideBV algorithm are hardware based packet classification algorithms. In this paper, simple and memory efficient approach for packet classification has been proposed using Xnor gate instead of using lookup tables called XnorBV approach. Packet header fields of Internet protocol (IP) addresses and protocol layer are classified using Xnor gate against predefined ruleset which also support ternary bit pattern of '1', '0' and '*' while port numbers of packet header support range match by comparing port numbers against lower bound and upper bound. The proposed parallel pipelined architecture can sustain a high throughput of +100 Gbps and low latency. The proposed method is memory efficient than other existing techniques, also supports prefix, range and exact match without use of range to prefix conversion. Also proposed XnorBV architecture is independent of ruleset feature and supports multiple dimension classification.**

***Index Terms*— Firewall; Network Intrusion Detection System; Packet Classification; Quality Of Services.**

I. INTRODUCTION

A sequence of packets coming from the source system to a destination system is popularly label as traffic flow or packet flow and a sequence of packets from particular source to a particular destination is called a flow. A flow can be identified by using technique called packet classification which categorizes the incoming packets into different flow by inspecting values of header fields of packets within a certain time [1]. For identification and arranging packets into different flow, each incoming packet is checked against a set of rule [2], if an incoming packet is matched with any rule of a rule-set then only it is accepted otherwise rejected. After categorizing incoming packets into different classes, each flow can be processed differently to differentiate the services suggested for the user. Each application and service requested by the user requires packets of same class. Packet classification technique helps to provide respective packets to respective services efficiently using predefined rule-set. Also, various services like firewalls, Virtual private network, network security, policy-based-routing, traffic shaping and quality of services incorporated the packet classification technique to detect threats and to prevent unauthorized access to the network [3][4]. Due to these manifold advantages of packet classification technique in modern communication, packet classification has become an integrated part of all type of intrusion detection systems, firewalls, internet routers and virtual private networks[5].

Software solutions are available to perform classification of packets but they are insufficient for high speed network applications [4]. In software tools, classification is generally done by checking only port numbers or IP addresses or protocol layer. Performance of software solutions which support inspection of multiple fields is not desirable for wire speed processing. For wire-speed processing and secure networking, hardware solutions are desirable and classification of packets can be done by checking all fields of packet header. In hardware packet classification solution, multiple fields of an incoming packet are checked against each rule of a rule-set. A size of ruleset may vary from hundred to thousand rules. The challenge and difficulty for hardware implementation of packet classification system is memory requirement to store large number of rules [2]. Generally, rules are stored using on-chip memory resources of field programmable logic array (FPGA) but because of limited on-chip memory resources, storing of a large number of rules is the problem [1]. For packet classification, rules are stored in a decreasing order of their priority in a ruleset and action is taken according to their priority. Figure 1 depicted below shows a standard 5-tuple packet header having destination and source Internet Protocol (IP) address field, destination and source port number field and the protocol field [3]. For different combination of values of the fields require different matches like prefix match for destination and source Internet Protocol address field, range match for destination and source port field and exact match for protocol field.

Source IP address	Destination IP address	Source port	Destination port	Protocol

Figure 1: Standard 5-tuple packet header

Considering the fact that packet classification system is the central part of various security tools and applications over internet and computer systems [6]. Various packet classification methods have been proposed to perform classification of packets just because of special computational method and certain limitations most of the existing technique may not be suitable for hardware implementation.

II. Problem In Packet Classification

Important issue of packet classification architecture is Power consumption. As throughputs of trillions of bits per second achieved by routers, power consumption becomes an increasingly critical concern. Power efficiency depends on number of rules used to classify incoming packet. This is one of aspect used for evaluation of power efficiency of packet classification system. The power consumed by the router to drive away the extremely large heat created by the router components extensively assist to the operating costs [8]. The power consumption in search engines is becoming an increasingly important evaluation parameter because each port of routers contains packet classification devices and router lookup [4].

Memory requirement is another important issue of packet classification. Nowadays, researchers aim to find out solutions for large ruleset. Method of classification and number of rules stored in classifier is related to amount of memory required. Due to limited resources available on FPGA, memory has become very important issue of hardware solution to support large number of rules [9].

Speed and pliability in specifications are the issues in packet classification devices. In packet classification process, packets are categorized based on a set of predefined rules also called as packet filters. Rules or filters define patterns that are to be matched against incoming packets for arranging packets for different flows [6] [10]. Packet filters or rules specify possible values for each field of a standard 5–tuple packet header [8] [11]. The address fields of a packet header are often used prefixes to define the addresses, although in address fields arbitrary bit masks are acceptable in a classifier or ruleset and this feature is widely used in real filter sets. Rules or Filters specify a range value for port -fields of packet header for matching incoming packets. Protocols can be in two ways either exact value or as a wildcard. Values specified by bit masks are allowed in some system for protocol field of incoming packet, even if it's not clear how convenient that feature is [8][12].

III. Background And Related Work

Methods which are efficient and desirable for hardware implementation can be broadly classified into two approaches decision-tree based approach and decomposition based approach [13] [14]. In decomposition based approach, classification of packets is done in two phase: In first phase, independent searches are performed on each field of packets, while in second phase: results from the first phase are combined [15]. Decomposition based algorithms are suitable for hardware implementation can sustain high throughput at low latency. Bit Vector (BV), Aggregated Bit Vector (ABV), Bit Vector- Ternary Content Addressable Memory (BV-TCAM), Field-Split Bit Vector (FSBV) [16], Crossproducting, Recursive Flow Classification (RFC) and StrideBV [17] are the some example of decomposition based approach. Bit Vector- Ternary Content Addressable Memory (BV-TCAM) algorithm and StrideBV algorithm support all matches and are scalable to large number of rule in a rule-set.

Ternary Content Addressable Memory (TCAM) is the desirable hardware solution because of its simple management and speed. To check all fields at a time and at high speed Ternary Content Addressable Memory (TCAM) based search engine is used [18] [19]. Extension of Ternary Content Addressable Memory (Ternary Content Addressable Memory (TCAM) approach is Bit Vector-Ternary Content Addressable Memory (BV-TCAM) uses Ternary Content Addressable Memory (TCAM) approach and Bit-Vector approach to support prefix, range and exact match. Bit Vector- Ternary Content Addressable Memory (BV-TCAM) approach is used to increase throughput and to compress data representation. This approach is generally used in network intrusion detection systems where report of multi matches at gigabits link rate is necessary. In packet classification, from multi match only single match of highest priority is reported for further processing due to routing problems [2]. In Bit Vector- Ternary Content Addressable Memory (BV-TCAM) approach, IP addresses and protocol layer of header are matched using Bit-Vector approach and port numbers are matched using TCAM approach in parallel and results are ANDing to get final output. This approach supports multi match without use of range to prefix conversion [2] [20].

Bit vector algorithm is desirable and widely used algorithm for hardware implementation of packet classification. Figure 2 shows the bit vector algorithm where bit value '1' indicates matching of incoming packet against a set of rule while bit value '0' indicates the mismatch of incoming packet against a predefined ruleset. In Bit-Vector (BV) algorithm, rules are arranged in a ruleset based on their priority. Generally to avoid complexity in assigning a priority to each rule, rules are arranged in decreasing order to their priority. Bit-vector is simple and has low computational complexity on hardware. For multi field packet classification, each field generates bit vector and then the bit vector of each fields are ANDing together to get final bit vector indicating the status of an incoming packet against a ruleset as shown in Figure 2.

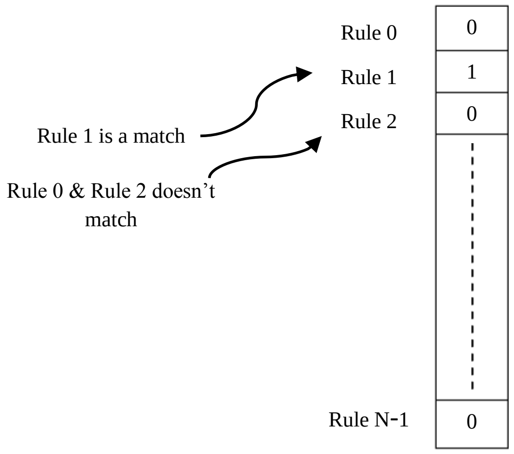
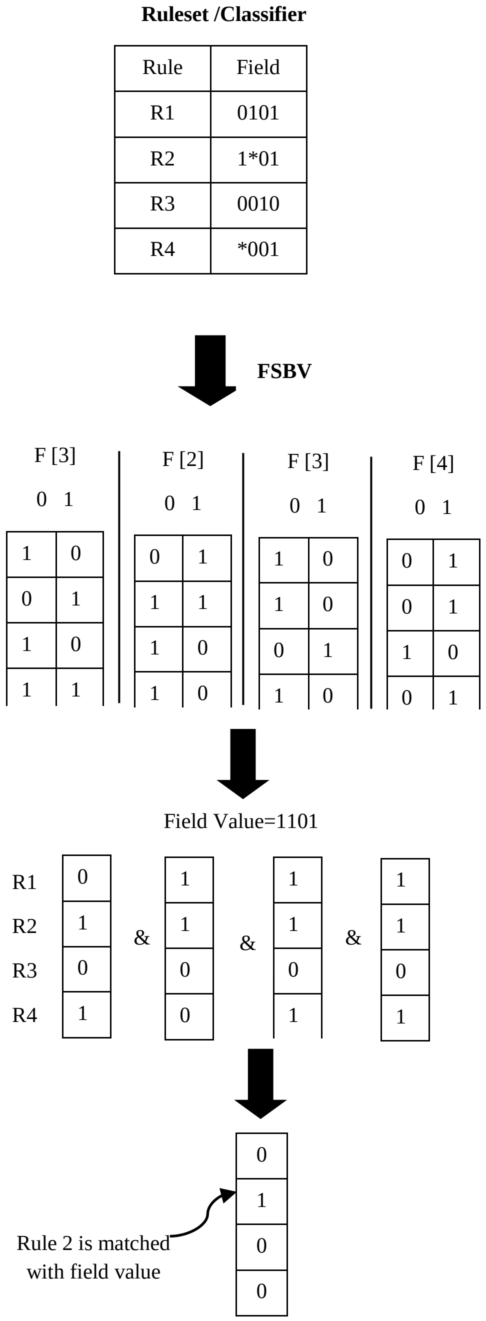


Figure 2: Basic of Bit-Vector Algorithm

Field split bit vector (FSBV) and StrideBV algorithm are the extension of bit-vector algorithm. In Field split bit vector (FSBV) algorithm, large pipelined stages are used to perform classification of an incoming packet against a predefined ruleset or filters. In Field split bit vector (FSBV), header field of W bits decompose into W subfields. W subfields generate bit vectors which are ANDing together to produce one bit vector. Each value position of a bit-vector after ANDing operation indicates the status of incoming packet against a predefined rule as shown in figure 3.



Ruleset /Classifier

Rule	Field
R1	0101
R2	1*01
R3	0010
R4	*001

Figure 3: FSBV Algorithm for Packet Classification

In StrideBV algorithm, pipelined stages are used to perform classification of incoming packets. At each pipelined stages, a stride memory is used to store look up table, stride (size k) of an incoming packet corresponds to memory location of the stride memory and extract one N bit-vector where N represents number of rule [17]. This N bit-vector indicates the statue of the stride of an incoming packet against a set of rule. For 5-tuple packet header of 104 bits and stride size of k=4, 26 pipelining stages are required to get matched result. At each stage of pipelining, N bit-vector of current stage ANDing with previous N bit-vector to produce resultant N-bit vector. The resultant N-bit vector of previous stage is ANDing with current stage and so on. The final stage N-bit vector represents the status of incoming packet against a set of rule and is given to priority encoder to extract highest priority matched rule. A pipelined priority encoder stage is used to extract highest priority rule and to decrease latency of the system. StrideBV can sustain high throughput of +100 Gbps at the cost of memory and latency. StrideBV is rule-set feature independent approach and also eliminates the use of range to prefix conversion for port numbers. An example of strideBV algorithm is shown in Figure 4. For the same ruleset and field value =1101, the strideBV algorithms for stride size k=2 can be performed as shown in Figure 4. For stride size k, the memory has depth of 2^k and width of N. In StrideBV, classification is done in w/k stages, where w number of bits in field and k is stride size.

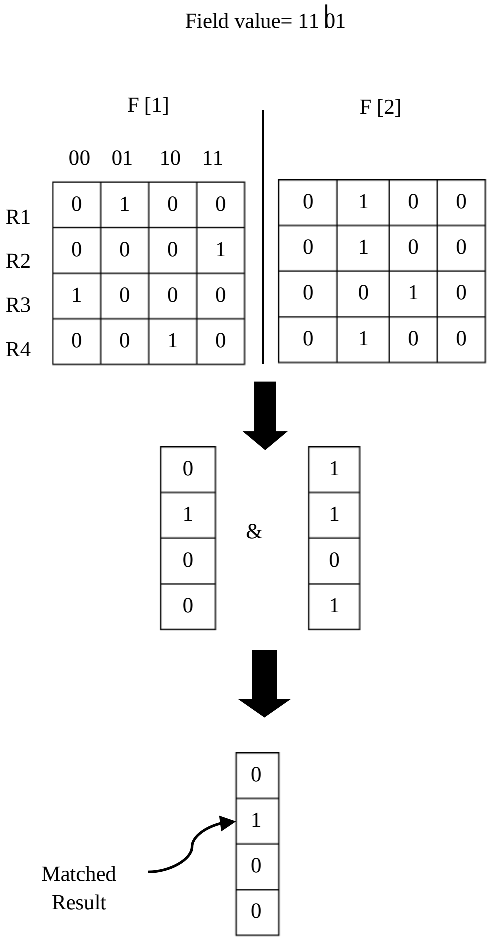
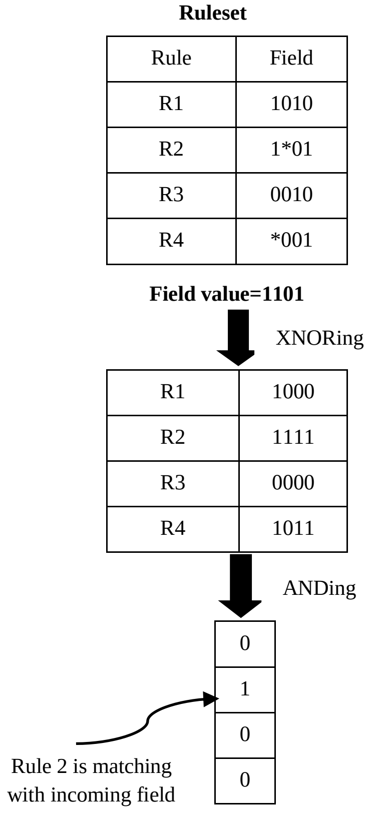


Figure 4: StrideBV Algorithm

IV. PROPOSED XNOR METHOD AND ARCHITECTURE

In this work, classification of each field or tuple of incoming packet is done using XnorBV method instead of using look-up tables. In XnorBV, an Xnor gate is used as a basic comparator for comparing incoming packet with rule of a ruleset. Use of Xnor gate makes the architecture simple and efficient for wide variety of communication network involves packet filtering or packet classification. Using XnorBV algorithm, the proposed design achieves good results on same operating frequency of 300MHz. in XnorBV algorithm, each field of a packet header generates a bit vector which will be ANDing with bit vector generated by others' field to get final output bit-vector. A final bit-vector is given to priority encoder module to fetch higher priority matched rule. In the proposed method, checking of each bit of a field against each bit of a rule stored in a ruleset is done using XNORing operation. Using behavioral modeling of VHDL, designed system supports ternary bit format of '1', '0' and '*' (wildcard entry). The proposed XnorBV method of packet classification is illustrated in figure 5, with the same ruleset and field value=1101 as that of Field split bit vector (FSBV) and StrideBV method of packet classification. After XNORing operation, each bit of obtained output after XNORing is ANDing to get one bit which indicates the status of a rule for incoming packet field [5]. A 5-tuple standard packet header having five fields which are source Internet Protocol (IP) address, destination Internet Protocol (IP) address, source port number, destination port number and protocol layer. In this paper, the classification of IP address fields and protocol field are performed using XnorBV method. Proposed XnorBV module supports prefix and exact match for Internet Protocol (IP) addresses and protocol layer respectively.

.



Rule	Field
R1	1010
R2	1*01
R3	0010
R4	*001

R1	1000
R2	1111
R3	0000
R4	1011

0
1
0
0

Figure5: Proposed XnorBV Algorithm

Figure 6 shows the circuit diagram of proposed XnorBV method of generating bit vectors. A field of 5-tuple incoming packet is checked against N rules of a ruleset. To understand the generation of bit vector using XnorBV method with the help of circuit diagram, let the length of rule and a field of an incoming packet be k bits. Let the first rule of a ruleset is given by R1=$W_{k-1}W_{k-2}$….. W_0 and a field of an incoming packet is given by F1= $T_{k-1}T_{k-2}$….. T_0. Each bit of a rule and a field is XNORing and after completion of XNORing operation, result of k-bits is ANDing to get single bit indicating the matching or mismatching of field with a rule. Same operation is performed for each and every rules of a ruleset of size N to get N-bit vector for the particular field of a packetThe detailed algorithm of generating bit vector and performing packet classification is given 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: $R_n=W_{n\ k-1}\ W_{n\ k-2}\ W_{n\ k-3}$………… $W_{n\ 0}$,

F=$T_{k-1}\ T_{k-2}\ T_{k-3}$ ………….. T_0, where= 1 ………. N ←

```
1: for n ← 1 to N do {Process $R_n$}
2: for k  k-1 to 0 do
3: S[n] [k] = $W_{n\ k-1}$ Xnor $T_{k-1}$
4: end for←
5: for b  0 to k-1 do
6: let Y=1,
7: Y = $S_{n\ b}$ 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 B_1, B_2, B_3, B_4 and B_5 be the bit vector of 5-tuple packet

```
1: for n ← 1 to N do {bit-wise AND}
2: V= $B_{1\ n}$ AND $B_{2\ n}$ AND $B_{3\ n}$ AND $B_{4\ n}$ AND $B_{5\ n}$
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: $V_m$ ← V { $V_m$ Output of Priority Encoder}
```

To support range match for port numbers, comparison of a field value against the lower bound and upper bound of a rule is done. Figure 7 shows the range module to perform range match for port numbers of a packet. To make designed architecture for supporting range match lower bound and upper bound has to be defined for each rule of a ruleset and method of performing range match is illustrated in Figure 7. A ruleset with lower and upper bound is depicted in Figure 7 with field value = 1000. A field value is to compare against lower bound, if field value is greater or equal to lower bound then it gives '1' otherwise '0' similarly if a field value is lower than or equal to upper bound then it gives '1' otherwise '0'. Bit values obtained after comparing field value against lower bound and upper bound are ANDing to get one bit which indicating that field value is in between lower bound and upper bound. Range search module is used for source port number and destination port number each of 16 bits for various applications. The proposed architecture supports prefix match for IP addresses, range match for port numbers and exact match for protocol field. Also, it is independent of ruleset feature and supports multiple dimension classification.

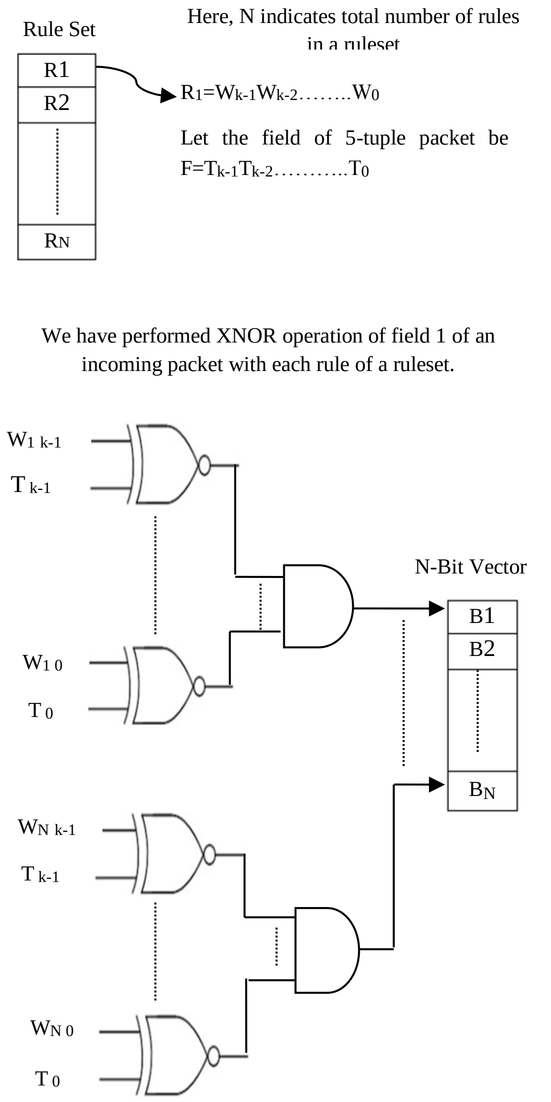
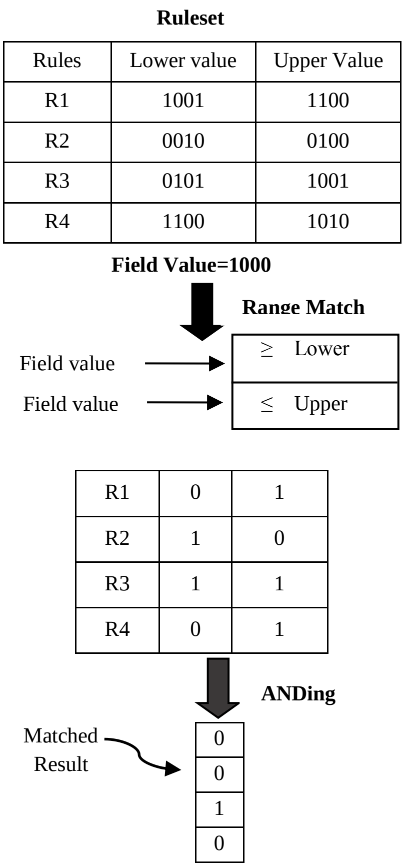

Rule Set
R1
R2
RN
Here, N indicates total number of rules in a ruleset
R1=Wk-1Wk-2……..W0
Let the field of 5-tuple packet be F=Tk-1Tk-2………..T0
We have performed XNOR operation of field 1 of an incoming packet with each rule of a ruleset.
W1 k-1
T k-1
W1 0
T 0
WN k-1
T k-1
WN 0
T 0
N-Bit Vector
B1
B2
BN

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

The complete architecture for packet classification supporting prefix, range and exact match is depicted in Figure 8. Rules are arranged in ruleset in a decreasing order of their priority. The architecture shown in Figure 8 performs the classification of complete packet header of 104 bits with multi-match packet classification feature. The storage of rules for each tuple is done separately and checked each respective tuple or field of an incoming packet against a respective rule-set. A five tuple packet header gives five N-bit vectors; each N-bit vector indicates the status of that tuple against predefined rules in a ruleset. After getting partial results from the classification process of each tuple of the packet, the results of five tuple are undergo ANDing operation to get final bit vector indicating match or mismatch of the packet against the rules of a ruleset. For IP addresses and protocol layer, XnorBV module is used to perform prefix as well as exact match. An XnorBV module can support ternary bit format ‘0’, ‘1’ and ‘*’ (wildcard entry). For port numbers of a packet, range module is used to generate bit-vector. In this way, the proposed architecture can perform prefix, range, and exact match. Priority encoder is used to decide the highest priority rule from the final bit vector and select the rule for further operation.


Ruleset

Rules	Lower value	Upper Value
R1	1001	1100
R2	0010	0100
R3	0101	1001
R4	1100	1010

Field Value=1000

R1	0	1
R2	1	0
R3	1	1
R4	0	1

Figure 7: Range Search Module for Range Match

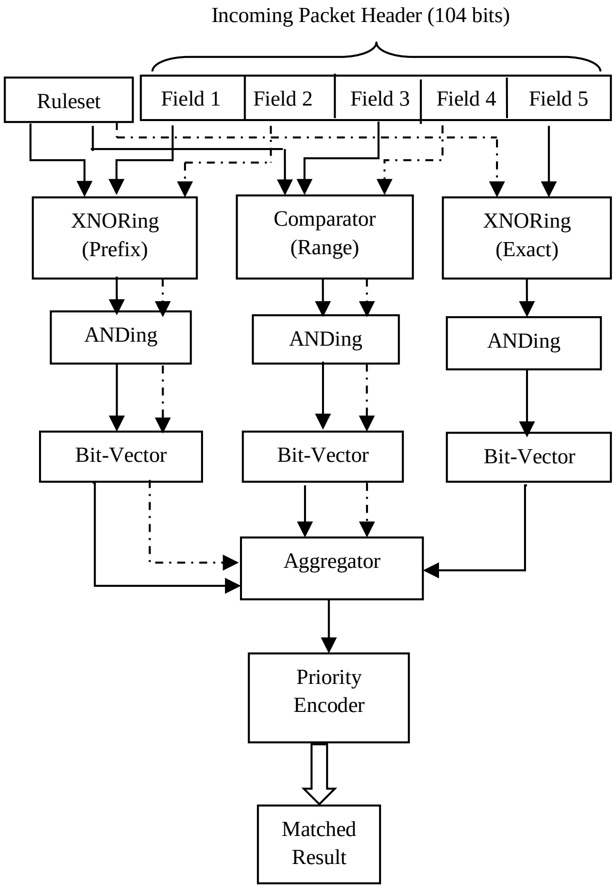


Figure 8: Proposed Architecture for Packet Classification

V. PERFORMANCE EVALUATION

The VHSIC hardware descriptive language (VHDL) is used to design the architecture on Xilinx ISE design 13.1 suite. Design utilization summary of the architecture for different size of ruleset on Virtex 6 family with target device XC6VLX760 device is tabulated in Table.1. Number of slice registers, number of LUTS and number of bonded input output pins require for ruleset of size 32 rules, 64 rules, 128 rules, 256 rules and 512 rules are given in table 1. Using our method, large ruleset can also support using on-chip resources. Since on-chip resources of FPGA is limited and design should be resources efficient.

Table 1
Design Utilization Summary

Family : Virtex 6 Target Device : XC6VLX760					
Ruleset	32 rules	64 rules	128 rules	256 rules	512 rules
No. of slice register	144	185	231	358	613
No. of slice LUTs	566	1070	2144	4125	7707
No. of fully used LUT-FF pair	34	65	127	255	508
No. of bounded IOBs	111	112	113	114	115
No. of BUFG/ BUFGCTL	1	1	1	1	1

Memory requirement to store large number of rules is the major problem of hardware solutions. Techniques available are not much memory efficient for storing large number of rules due to limited on-chip memory available on FPGA [9] [21]; large number of rules cannot be stored using on-chip memory available and interfacing of external memory to store large ruleset is undesirable for high speed processing network [4]. To overcome memory requirement for supporting large ruleset, a simple and efficient approach of packet classification is proposed called XnorBV. For packet classification of header size of 104 bits, proposed method requires 120 bits; extra 16 bits are required to specify the lower bound or upper bound of range search module. Therefore, the XnorBV method requires 15 bytes to define a rule of a ruleset for a standard 5-tuple packet header.

The Latency of a system is the time required to get output after applying input. In packet classification, the latency is defined as time required for completing one classification process. In XnorBV method, the classification process is performed in three stages. In first step, there is separation each field of an incoming packet to classify against ruleset to generate bit vectors. In second stage, bit-vector of each field generated in first stage i.e. partial results are combined to get final bit vector which indicates the status of rules against incoming packet. Final result obtained in second stage is forward to priority encoder to get single match result from multi match result for further process. Extraction of highest priority matched rule using priority encoder is done in third stage. In this way, the proposed XnorBV method requires three clock cycle to perform classification of one incoming packet. So, the latency of proposed architecture is 3 clock cycles which is also desirable for low latency application.

With the clock frequency of 300 MHz and latency of 3 clock cycle, calculated throughput of the system is 114 Gbps. Total data bits is the sum of all bits processed at each stage of pipelined architecture or design. Whenever, pipelined architecture is used there is an increase in the throughput of the system but at the cost of latency. However, in the proposed XnorBV, latency of the architecture is low and throughput is +100 Gbps meeting the in-card requirement applications where low latency is required.

Power efficiency is one of the major and crucial parameter of VLSI design. Low power requirement is crucial for low power high frequency application devices. Static power of a design is almost constant for any ruleset while dynamic power is varying by varying ruleset. Dynamic Power is 0.36 mW for one rule in a ruleset. For evaluation of performance parameters and to overcome the technology gap, evaluations are done by assuming the operating frequency or clock frequency of 300 MHz of all the existing architecture or method to get real analysis [17]. The memory requirement per rule, throughput at 300 MHz for ruleset size of 512 rules, latency in clock cycle and power efficiency of proposed and other existing techniques are listed in Table 2.

Table 2
Performance Comparison with Existing techniques at 300 MHz

Approach	Memory (bytes/rule)	Latency (clock)	Throughput (Gbps)	Power (mW)
Proposed XnorBV	15	3	114	0.3
StrideBV (k=4) [1]	52	31	135	0.624
DFCL [14]	90	5	19	N/A
BV-TCAM [13]	154	11	75	0.846
TCAM [12]	30	1	115	4.902
Emulated TCAM [15]	24	1	64	N/A

From Table 2, the results show that proposed XnorBV method requires less memory, has low latency high throughput and low power consumption as compare to other existing techniques or methods.

VI. CONCLUSION

Proposed method XnorBV architecture using Xilinx ISE 13.1 suite selecting Virtex 6 XC6VLX760 as target device is memory efficient requires 15 byte/rule less than any other existing technique of packet classification. Architecture supports prefix, exact and range match without use of range to prefix conversion and is independent of ruleset feature.

Power efficiency is also improved with power increment in addition of one rule. The proposed architecture can sustain high throughput of +100 Gbps at low latency which is desirable for low latency applications.

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.

[6] Aladdin Abdulhassan and Mahmood Ahmadi, "Parallel Many Fields Packet Classification Technique using R-Tree ," in *Annual Conference on New Trends in Information & Communications Technology Applications-(NTICT'2017)*, 7-9 March 2017.

[7] Safaa O.Al-Mamory and Wesam S.Bhaya; "Taxonomy of Packet Classification algorithms," in *Journal of Babylon University/Pure and Applied.*

[8] Balasaheb S. Agarkar and Uday V. Kulkarni, Ph.D., "A Novel Technique for Fast Parallel Packet Classification," in *International Journal of Computer Applications (0975 – 8887)* Volume 76–No.4, August_2013.

[9] Andreas Fiessler, Sven Hager and Björn Scheuermann, "Flexible Line Speed Network Packet Classification Using Hybrid On-chip Matching Circuits," in *IEEE 18th International Conference on High Performance Switching and Routing (HPSR)*, 18-21 June 2017.

[10] Pankaj Gupta and Nick Mckneown; "Algorithms for packet classification," in IEEE magazine, March/April 2001 pp. 24-32

[11] G. Jedhe, A. Ramamoorthy, and K. Varghese, ''A Scalable High Throughput Firewall in FPGA,'' in *Proc. 16th Int'l Symp. FCCM.* Apr. 2008, pp. 43-52.

[12] Yeim-Kuan Chang and Cheng-Chien Su, "Efficient TCAM Encoding Scheme Packet Classification using Gray Code," *in IEEE GLOBECOM 2007 proceedings @2007 IEEE*.

[13] M. Faezipour and M. Nourani, ''Wire-Speed TCAM-Based Architectures for Multimatch Packet Classification,'' in *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 5-17, Jan. 2009.

[14] D.E. Taylor, ''Survey and Taxonomy of Packet Classification Techniques,'' in *ACM Computing Survey*, vol. 37, no. 3, pp. 238-275, Sept. 2005.

[15] Lu Sun, Hoang Le, Viktor K. Prasanna; "Optimizing Decomposition-based Packet Classification Implementation on FPGAs," in *International Conference on Reconfigurable Computing and FPGAs*; 978-0-7695-4551-6/11 $26.00 © 2011 IEEE; pp. 170-175.

[16] W. Jiang and V. K. Prasanna, "Field-split Parallel Architecture for High Performance Multi match Packet Classification using FPGAs," *in Proc. of the21st Annual Symp. on Parallelism in Algorithms andArch. (SPAA)*, 2009, pp. 188–196.

[17] Thilan Ganegedara, Weirong Jiang, and Viktor K. Prasanna, Fellow, IEEE; "A Scalable and Modular Architecture for High-Performance Packet Classification," in *IEEE Transactions On Parallel And Distributed Systems*, Vol. 25, No. 5, May 2014; 1045-9219 _ 2013 IEEE, pp.1135-1144.

[18] C.R. Meiners, A.X. Liu, and E. Torng, "*Hardware Based Packet Classification for High Speed Internet Routers*," Berlin, Germany: Springer-Verlag, 2010.

[19] Cheng-Liang Hsieh and Ning Weng, "Many-Field Packet Classification for Software-Defined Networking Switches," *in @ACM 2016 , ANCS '16,* March 17-18, 2016, Santa Clara, CA, USA.

[20] H. Song and J.W. Lockwood, ''Efficient Packet Classification for Network Intrusion Detection Using FPGA,'' *in Proc. ACM/SIGDA. 13th Int'l Symp. FPGA*, 2005, pp. 238-245

[21] Hung-Yi Chang, Chia-Tai Chan, Pi-Chung Wang, Chun-Liang Lee; "A Scalable Hardware Solution for Packet Classification," in *ICCS @2004* IEEE.

[22] D. Taylor and J. Turner, ''Scalable Packet Classification Using Distributed Crossproducing of Field Labels,'' in *Proc. 24th Annu. Joint IEEE INFOCOM*, Mar.2005, vol.1, pp.269-280

[23] C.A. Zerbini and J.M. Finochietto, ''Performance Evaluation of Packet Classification on FPGA-Based TCAM Emulation Architectures,'' in *Proc. IEEE GLOBECOM*, 2012, pp. 2766-2771.

[24] Weirong Jiang and Viktor K. Prasanna, "Large-Scale Wire-Speed Packet Classification on FPGAs," in *ACM*, 2009.