

# FPGA based Decision Tree Classification Accelerator for Networking Applications

<sup>1</sup>Nagesh. R, <sup>2</sup>Vani A.

<sup>1</sup>Assistant Professor, Department of Electronics and Communication, Government Sri Krishna Rajendra Silver Jubilee Technological Institute, Bengaluru, Karnataka, 560 001, India

<sup>2</sup> Research Scholar, Department of Electronics and Communication Engineering, BMS College of Engineering, Bengaluru, Karnataka, 560 019, India

**Abstract :** Packet Classification is used by networking applications to sort packets into flows by comparing their header with a list of rules such that packets will be placed in the flow based on matched rule. With advance in technology and larger storage capacity available in computers, large amount of data needs to be processed leading to larger classification times. Decision tree Classification (DTC) is a known classifier for packet classification. In this paper, a hardware accelerator for DTC implementation is proposed which consists of parallel nodes independently processing data from a streaming source.

**IndexTerms - Decision Tree Classification (DTC), Header Fields, Ethernet, TCP/IP, HyperCuts Algorithm.**

## I. INTRODUCTION

The widespread increase in internet usage is due to ease of implementation in various devices such as desktops, smart phones, tablets and so on. The increasing demand for the internet has posed a serious problem towards processing a large amount of data. This is accompanied with high power consumption, large resource utilization and longer execution time. Network processors are used for packet fragmentation and reassembly, encryption forwarding and classification. With such a large amount of data to be processed, network processors are put under a constant stress. Ramping up the clock speed or increasing the number of processing cores is done with the intention of increasing processing capacity. However, this will not benefit because this in a way increases the power consumption[1-5].

The use of hardware accelerators dedicated to the most computational part of a network processor helps in reducing power consumption and also increases processing capability. This is because hardware accelerators use less number of transistors than other general purpose processors and they process more data at a lower clock speed because they are dedicated for specific tasks[6,7].

The paper deals with the design and implementation of a packet classification hardware accelerator also known as a classifier[8] which is energy efficient.

Decision Tree Classification is a widely used classification technique and is used to categorize streaming data packets into various categories. The classification of data packets into various subgroups or categories or classes involves two processes. The first process deals with constructing a decision tree model which consists of root nodes, internal nodes and leaf nodes[9]. The internal node has a splitting decision and a splitting attribute and results in further branching while the leaf node has no further branching and hence has a specific category classification. The second process is classification which involves application of decision tree model to the incoming data packets to predict their respective class. This process is recursive and continues till the depth of the tree is reached and all the packets are classified into their respective classes.

A HyperCuts Algorithm is used to split an internal node into leaf nodes depending upon the splitting attribute and splitting index. The data packets traverse from the root node down to leaf nodes or internal nodes according to the rules defined along the path to each node. Finally, the packets are in their respective nodes whose rule matches with their packet header field.

## II. BACK GROUND

A number of hardware implementations of the decision tree are reported in the literature [10], [11]. Single stage hardware accelerators over multi stage classifiers have always posed a limitation to the performance because no new dataset can be fed at the input till the previous set of data packets is classified. This poses a restraint on the throughput also and stresses the need to have multi stage DTC engines which can function and process data in a parallel fashion.

An advanced approach as proposed in [12] was to have equivalence between a decision tree and a threshold network since the signal has to pass through two stages irrespective of the depth of the decision tree. Many hardware accelerators known in the literature use a large amount of hardware resources.

Hardware implementation of data mining algorithms have been reported in the past research work. Baker and Prasanna [13] have designed, implemented and accelerated Apriori algorithm, a popular association rule mining technique.

Many software implementations of DTC [16], [17] have been proposed which use complex data structures for splitting and redistribution processes. These implementations mainly focus on having a parallelized DTC.

Chrysos et al. [21] presented data mining on the web so as to perform classification and mining of huge amounts of e-data by implementation of data mining algorithms on a modern FPGA to accelerate certain CPU intensive data classification schemes. Also a parallel scheme was exploited at the decision variable level and implemented it on a high performance reconfigurable pattern [22].

The objective of this paper is to classify data packets efficiently according to a pre-defined set of rules. Also it aims at increasing the throughput and minimizing the resource utilization by employing DTC engines in parallel. The power consumption is reduced by implementing a simple decision tree structure thus decreasing its hardware complexity and this becomes evident by the reduced number of flip flops used compared to ones used in the previous approaches.

### III. PACKET CLASSIFICATION ARCHITECTURE

The architecture of data packets classification is as shown in Fig. 1

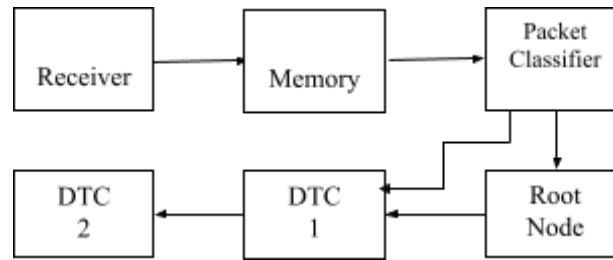


Fig. 1 Block Diagram of Packet Classification Architecture

Fig. 1 shows the block diagram of packet classification architecture consisting of a receiver, memory, packet classifier, root node, DTC 1st Stage and 2nd Stage.

#### 3.1 Receiver

The receiver receives input data packets. The packets are of 8 bits and the receiver stores the received packets in a temporary register. With each packet received, the byte counter is incremented. The counter stops incrementing ones the receiver receives 608 bits of data which is sent as the output of the receiver. The receiver also evaluates the data packets for the presence of any errors in the packet. The receiver thus outputs a valid data output and an error output

#### 3.2 Memory

The output of the receiver is fed as the input to the memory. The memory address is of 4 bits and hence it contains 16 memory locations. Each of the memory locations stores 608 bits of data. Data is written into the memory when the write enable is set and it is read from the memory when the write enable control signal is reset.

#### 3.3 Packet Classification

The packet classifier receives 608 bits of input data. The received packet is divided into Ethernet header, TCP header and IP header. Out of 608 bits of data, 208 bits are assigned as Ethernet header output, 160 bits as IP header output and 160 bits as TCP header output. The byte counter is 7 bytes. When the byte count reaches 67, the first 544 bits are assigned to a temporary register. When the byte count reaches 71 bytes, each of the temporary registers, Ethernet header, TCP header and IP header has 32 bits added to its range of values. The byte counter has its final counting value of 75. This occurs when load is active. When load becomes disabled, each of the temporary registers will be of 0 bits and data out is 32 bits. Thus a packet classifier outputs the data of size 32 bits for an input data of 608 bits.

#### 3.4 Root Node

The root node is the primary node from where the data packet starts traversing down the decision tree. The branching of the root node into leaf nodes and internal nodes depends upon the source and destination IP address. The value of the index is decided by the 16 bit source and destination IP address. Consider the source IP address, sip [3:0] < 4 and for this condition destination IP address, dip may have 4 conditions. For dip [3:0] < 4, index value is equal to 0. For dip [3:0] < 8, index value is equal to 1 and so on.

#### 3.5 Decision Tree Classifier (DTC)

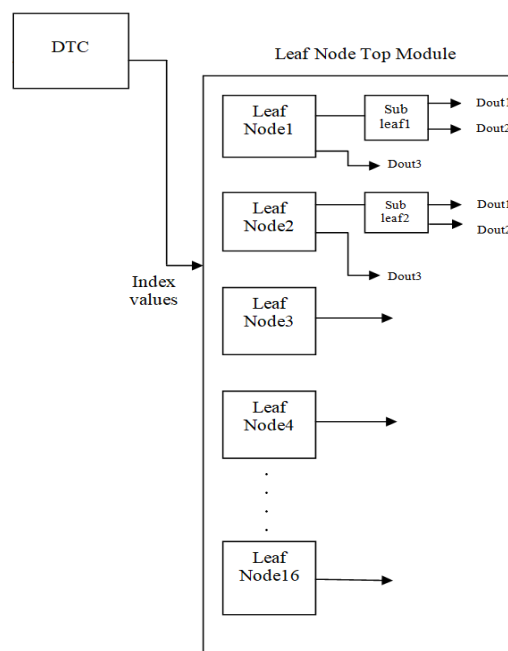


Fig. 2 Decision Tree Classifier

Fig. 2 shows the decision tree classifier. The index values of the DTC are fed into the leaf node top module which consists of 16 leaf nodes. The leaf node 1 and leaf node 2 are cut into sub leaf nodes depending upon the source port values. If the 16 bit source port is equal to 1, branching into the sub leaf node occurs and if source port is equal to 2, no branching occurs and dout3 will be the output. The branching of the sub leaf nodes depends upon the 16 bit destination port number and 8 bit protocol number. Depending upon these two values either dout2 or dout3 output is obtained.

For the remaining 14 leaf nodes, the output data depends only on the 16 bit source port number. In case the packet header matches with the port number, the output data follows the input. Else no output is obtained.

### 3.6 Decision Tree Classifier (DTC) Stages

A 2 stage parallel decision tree classifier is designed and implemented. This increases the execution speed and performance. Since the 2 stages are parallelized, a defect in any one stage would not affect the working of the other stage. Also the resource utilization and power consumption is minimal.

## IV. EXPERIMENTAL RESULTS

The proposed architecture is implemented on Spartan 3 FPGA development board. The synthesis tool that is used is Xilinx 14.7 and the Modelsim tool is 6.3f. The fig. 3 shows the simulation of the architecture.

### 4.1 Comparison of Resource Utilization

After performing synthesis of Decision Tree Classification architecture, the synthesis report is obtained which is tabulated and the results are compared with those obtained in the previous works.

Table 4.1: Synthesis Report of DTC Architecture

Serial No.	Resource Utilization of DTC Stages		
	Implementation	LUTs	Flip Flops
1	1 Stage	33	19
2	4 Stages	132	78
3	8 Stages	2718	2900
4	Whole Design	5121	5608

Table 1 shows the synthesis report of Decision Tree Classification Architecture. These results are compared with the readings of the previous work. The synthesis report of the previous work is shown in Table 2

Table 4.2: Synthesis Report [1]

Implementation	LUTs	FF
DT Engine 1 Stage	62	96
DT Engine 4 Stage	240	332
DT Engine 8 Parallel instances of 4 Stage	2952	3100
Whole Design	6442	5336

On comparing Table I and Table II, it becomes evident that the proposed architecture uses less number of flip flops and LUTs than what is used in [1].

## 4.2 Simulation

The simulation of the architecture is performed using Modelsim 6.3f. The simulation waveform is shown in Fig. 3

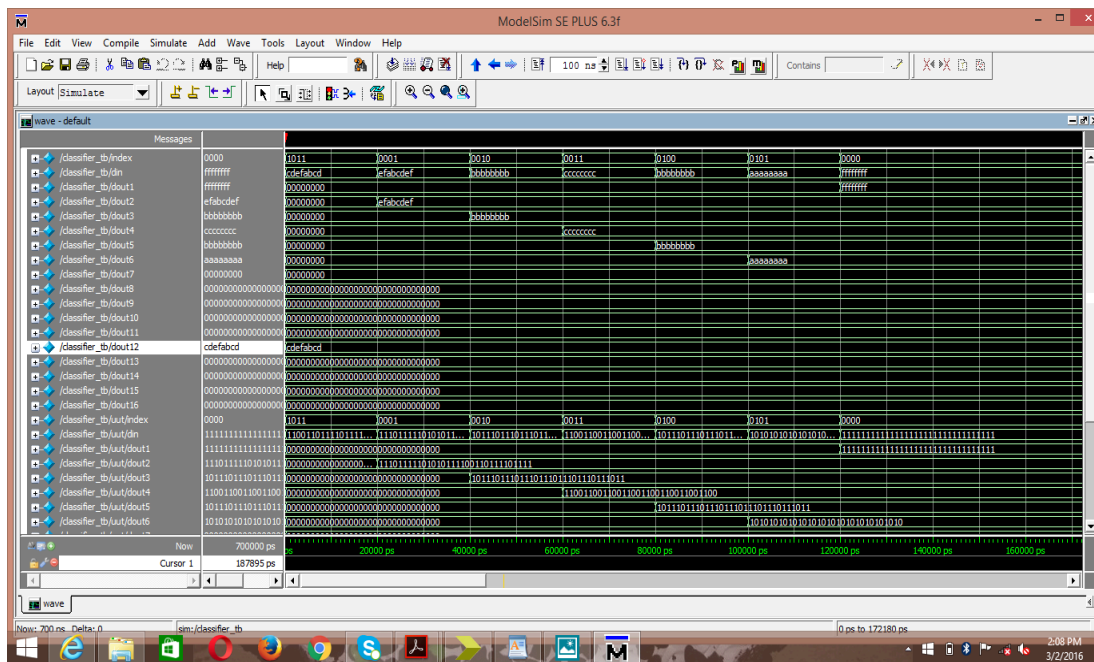


Fig. 3 Simulation

## V. CONCLUSION

The proposed architecture has the advantage of being highly scalable and exhibits high levels of parallelism. The performance of the architecture is not affected when large amount of input data is fed into the system and is independent of the number of splitting attributes used for branching. Higher levels of parallelism can be obtained by increasing the number of parallel pipelines. Even the depth of the decision tree can be increased by increasing the number of parallel pipelines.

The resource utilization of the decision tree classification hardware accelerator is minimal and hence the power consumption is also reduced. Contrary to the previous implementations, the parallelism of different decision tree engines and the use of temporary registers to minimize the clock cycles are focused upon.

## REFERENCES

- [1] Tong, Da, Yun Rock Qu, and Viktor K. Prasanna. "Accelerating decision tree based traffic classification on FPGA and multicore platforms." *IEEE transactions on parallel and distributed systems* 28, no. 11 (2017): 3046-3059.
- [2] Owaida, Muhsen, Hantian Zhang, Ce Zhang, and Gustavo Alonso. "Scalable inference of decision tree ensembles: Flexible design for CPU-FPGA platforms." In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-8. IEEE, 2017.
- [3] Struharik, Rastislav. "Decision tree ensemble hardware accelerators for embedded applications." In *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 101-106. IEEE, 2015.
- [4] Owaida, Muhsen, and Gustavo Alonso. "Application partitioning on FPGA clusters: Inference over decision tree ensembles." In *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 295-2955. IEEE, 2018.
- [5] Lin, Zhe, Wei Zhang, and Sinha Sharad. "Decision tree based hardware power monitoring for run time dynamic power management in FPGA." In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-8. IEEE, 2017.
- [6] Zhao, Shuang, Yipin Sun, and Shuhui Chen. "A discretization method for floating-point number in FPGA-based decision tree accelerator." In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 2698-2703. IEEE, 2018.
- [7] Barba, Jesus, María José Santofimia, Julio Dondo, Fernando Rincon, Julian Caba, and Juan Carlos López. "FPGA acceleration of semantic tree reasoning algorithms." *Journal of Systems Architecture* 61, no. 3-4 (2015): 185-196.
- [8] Nakahara, Hiroki, Akira Jinguji, Simpei Sato, and Tsutomu Sasao. "A random forest using a multi-valued decision diagram on an FPGA." In *2017 IEEE 47th international symposium on multiple-valued logic (ISMVL)*, pp. 266-271. IEEE, 2017.
- [9] Barbareschi, Mario. "Implementing hardware decision tree prediction: a scalable approach." In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 87-92. IEEE, 2016.
- [10] Barbareschi, Mario. "Implementing hardware decision tree prediction: a scalable approach." In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 87-92. IEEE, 2016.
- [11] Shawahna, Ahmad, Sadiq M. Sait, and Aiman El-Maleh. "FPGA-based accelerators of deep learning networks for learning and classification: A review." *IEEE Access* 7 (2018): 7823-7859.
- [12] Lin, Xiang, RD Shawn Blanton, and Donald E. Thomas. "Random forest architectures on FPGA for multiple applications." In *Proceedings of the on Great Lakes Symposium on VLSI 2017*, pp. 415-418. 2017.

- [13] Nakahara, Hiroki, Akira Jinguji, Tomonori Fujii, and Simpei Sato. "An acceleration of a random forest classification using Altera SDK for OpenCL." In 2016 International Conference on Field-Programmable Technology (FPT), pp. 289-292. IEEE, 2016.
- [14] Vranjković, Vuk, and Rastislav Struharik. "Coarse-grained reconfigurable hardware accelerator of machine learning classifiers." In 2016 International conference on systems, signals and image processing (IWSSIP), pp. 1-5. IEEE, 2016.
- [15] Soyulu, Tuncay, Oğuzhan Erdem, Aydin Carus, and Edip S. Güner. "Simple CART based real-time traffic classification engine on FPGAs." In 2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig), pp. 1-8. Ieee, 2017.
- [16] Lin, Zhe, Sharad Sinha, and Wei Zhang. "An ensemble learning approach for in-situ monitoring of FPGA dynamic power." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38, no. 9 (2018): 1661-1674.
- [17] Owaida, Muhsen, and Gustavo Alonso. "Application partitioning on FPGA clusters: Inference over decision tree ensembles." In 2018 28th International Conference on Field Programmable Logic and Applications (FPL), pp. 295-2955. IEEE, 2018.
- [18] Kang, Ming, Sujun K. Gonugondla, Sungmin Lim, and Naresh R. Shanbhag. "A 19.4-nJ/decision, 364-K decisions/s, in-memory random forest multi-class inference accelerator." IEEE Journal of Solid-State Circuits 53, no. 7 (2018): 2126-2135.
- [19] Su, Jiang, Jianxiong Liu, David B. Thomas, and Peter YK Cheung. "Neural network based reinforcement learning acceleration on fpga platforms." ACM SIGARCH Computer Architecture News 44, no. 4 (2017): 68-73.
- [20] Khan, Osama U., and David D. Wentzloff. "Hardware accelerator for probabilistic inference in 65-nm cmos." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24, no. 3 (2015): 837-845.
- [21] Jinguji, Akira, Shimpei Sato, and Hiroki Nakahara. "An FPGA realization of a random forest with k-means clustering using a high-level synthesis design." IEICE TRANSACTIONS on Information and Systems 101, no. 2 (2018): 354-362.
- [22] Nguyen, Xuan-Thuan, Trong-Thuc Hoang, Hong-Thu Nguyen, Katsumi Inoue, and Cong-Kha Pham. "An FPGA-based hardware accelerator for energy-efficient bitmap index creation." IEEE Access 6 (2018): 16046-16059.