Faculty of Engineering
Department of Computer Engineering

M.Sc.Thesis

Title of the Thesis:
Implementation and Mapping of Packet Classification Algorithm on Parameterizable Reconfigurable VLIW Architecture

Supervisor:
Dr. Mahmood Ahmadi

By:
Ehsan Zadkhosh

March 2013

Faculty of Engineering
Department of Computer Engineering

M.Sc. Thesis

Title of the Thesis
Implementation and Mapping of Packet Classification Algorithm on Parameterizable Reconfigurable VLIW Architecture

By:
Ehsan Zadkhosh

Evaluated and approved by thesis committee as:

Supervisor: **Assistant professor, Mahmood Ahmadi**

Internal Examiner: **Assistant professor, Jahanshah Kaboodian**

External Examiner: **Assistant professor, Arash Ahmadi**

March 2013

Acknowledgements

First of all, I would like to thank my supervisor, Mahmood Ahmadi for his substantial guidance and support in this thesis.

I also want to express my deepest gratitude to my parents for their support during my studies. I would also like to apologize to them for being so far away during these years. I hope to make it up to them in the future.

Finally I would like to thank my sister Nasrin, my uncle Naser, my friends for their support, showing interest in my work, and getting bothered with helping me, especially to Navid Behboodian, Alireza Ahmadi and Reza Falamarzi. Also Thijs van As for his permission of using his thesis as my main reference.

Ehsan Zadkhosh (ezadkhosh@IEEE.org)

Razi University, Kermanshah.

March 2013.

Dedicated to my parents,
for their love and support.

Abstract

Packet classification as the main task of network devices is the process of classifying different packets into different flows (classes). Recently, high speed progress in network applications and enormous increment in the number of network users caused exponential increment in network traffic. These complex applications and heavier traffic have emerged demands on more powerful devices. Furthermore, due to dynamic nature of network traffic, totally scalable methods for packet classification and implementation platforms that are able to cope with this dynamicity are needed. FPGAs are efficient candidates as they contain hardware programmable logic units that can be reconfigured based on the application requirements. *ρ*–VEX is a reconfigurable soft-core VLIW processor in an FPGA and it is able to use the inherit Instruction Level Parallelism (ILP) of applications. *ρ*–VEX has all the requirements of processing network applications.

For the purpose of scalability and economical reasonability we gave more attention to packet classification algorithms that have taken advantage of Bloom filter. Tuple pruning using Bloom filter for packet classification completely suits our goals. In this thesis we are going to prove the rationale behind implementing Bloom filter based packet classification algorithms on a reconfigurable and parameterizable VLIW soft-core processor, called *ρ*–VEX. As *ρ*–VEX is an open source, reconfigurable, flexible and powerful VLIW processor and Bloom filter is a parallel and space efficient data structure our way of classification is fast enough and reasonable for new demands, moreover this will be done in more economical environment.

We changed ρ–VEX to an embedded processor for one of the most important network applications, packet classification. The results have been depicted the huge performance gap between GPP processors and ρ–VEX. The customized ρ–VEX provided on average approximately 8x speedup.

Keywords: ρ-VEX, CRC, packet classification, Bloom filter, FPGA.

Table of Contents

Content **page**

List of Figures

Figure **page**

List of Tables

Table **page**

List of Acronyms

ACL Access Control List

ASIC Application Specific Integrated Circuit

ASIP Application-Specific Instruction Processors

BMR Best Matching Rule

CISC Complex Instruction Set Computer

CPI Clock Per Instruction

CPU Central Processing Unit

CRC Cyclic Redundancy Check

DFT Discrete Fourier Transform

DSP Digital Signal Processing

DWDM Dense Wavelength Division Multiplexing

EAPR Early Access Partial Reconfiguration

EDA Electronic Design Automation

FIS-Tree Fat Inverted Segment Tree

FPGA Field programmable Gate arrays

FU Functional Units

FW Firewall

GPP General Purpose Processor

ICP Internet Cache Protocol

IDE Integrated Development Environment

ILP Instruction Level Parallelism

IPC Instruction Per Clock

IPC IP Chain

ISA Instruction Set Architecture

NAP Network Access Point

NAT Network Address Translation

NP Network Processors

OC-CRC Our One Clock CRC

QoS Quality of Service

RFC Recursive Flow Classification

RISC Reduced Instruction Set Computer

SIMD Single Instruction Multiple Data

SoC System on Chip

SRAM Static Random Access Memory

TCAM Ternary Content addressable Memories

TSP Tuple Space Pruning

TTM Time to Market

URL Uniform Resource Locator

VEX VLIW Example

VLIW Very Long Instruction Word

VPN Virtual Private Network

Chapter 1
Introduction

1.1 Introduction

Internet at its beginning was only text emailing and webs but it encountered to drastic changes. On one hand the addition of network applications and on the other hand enormous increment in the number of network users causes more traffic growth. The new traffic means many new added operations on each network packet. The result is more processing requirements per packet in comparison to early network devices. For this complex applications and heavier traffic than early days of networks and because of the natural dynamicity of network traffic and bandwidth, demands on more powerful devices, which are able to handle this new situation, have been emerged. The rest of this chapter is organized as follows.

Problem statement is presented in section 1.2. Subsequently, our proposed solution is described in section 1.3. Project goals are identified in Section 1.4. Section 1.5 concludes this chapter with an overview of this thesis organization.

1.2 Problem Statement

The demand for computing power in consumer electronics is increasing at a very high rate. High performance (embedded) computers is needed to cope with all modern application, so computer systems with a single general-purpose Central Processing Unit (CPU) are replacing with computer systems of multiple general-purpose CPUs, or a combination of CPUs and application-specific processing units. This high speed progress of applications that are related to network caused exponential increment in network traffic. Internet at its beginning was only text emailing and webs but it encountered to drastic changes. On one hand the addition of applications such as distributed databases, video-conferencing, multiplayer gaming, etc. and on the other hand enormous increment in the number of network users causes more traffic growth. The new traffic means many new added operations on each network packet. The result is more processing requirements per packet in comparison to early network devices such as firewall, antivirus and intrusion detection / prevention applications. For this complex applications and heavier traffic than early days of networks, demands on more powerful devices, which are able to handle this new situation, have been emerged. Due to dynamic nature of network traffic, we need implementation platforms that are able to cope with this dynamicity.

1.3 Proposed Solution

Field programmable Gate arrays (FPGAs) could be efficient candidates as they contain hardware programmable logic units that can be reconfigured based on the application requirements, sometimes FPGAs referred to as reconfigurable platforms. Application Specific Integrated Circuit (ASIC) is the other perfect candidate, but considering time to market as an important parameter in hardware production, ASIC is less appropriate platform than FPGA due to the fact that FPGA does not have time to market.

FPGAs have become a widely used tool for rapid prototyping, providing software-like flexibility and hardware-like performance. To exploit Instruction Level Parallelism (ILP), a

Very Long Instruction Word (VLIW) processor can be utilized to increase the performance beyond a single issue-width processor. Single issue-width processors can only take advantage of temporal parallelism (by utilizing pipelining) while VLIW architectures can additionally take advantage of the spatial parallelism by utilizing multiple Functional Units (FUs) to execute several operations simultaneously. In addition, VLIW processors are simpler in design when compared to their more complex Reduced Instruction Set Computer (RISC) counterparts. FPGAs are constantly improving and provide a technology platform that allows fast and complex reconfigurable designs. In many cases, the utilization of FPGAs implies a large reduction in development costs or an enormous speedup of the implemented algorithm. Applications in the multimedia and network domain happen to contain a lot of ILP, because they typically consist of many independent repetitive calculations. VLIW processors exploit ILP by means of a compiler that is completely aware of the target processor architecture. When a VLIW processor is implemented in an ASIC, its issue-width is fixed at design time. Therefore, the issue-width of the processor cannot be adjusted after fabrication to suit a different set of applications. A soft-core VLIW processor can be implemented in an FPGA, and its organization can be changed easily by loading a new bit-stream. Design-time reconfigurability requires full configuration of the whole FPGA if a certain parameter of the FPGA-implemented processor is to be changed. In literature, all of the available soft-core VLIW processors only provide some form of design-time and not run-time reconfigurability. In order to change the issue-width or any parameter of a processor, a new full bit-stream is to be downloaded to configure the whole FPGA and all other circuits in the FPGA have to be stopped.

ρ–VEX is a parameterizable and extensible soft-core VLIW design-time reconfigurable processor in an FPGA. In these processor parameters such as type and the number of FUs, supported instructions, memory bandwidth and register file size can vary based on the variation of the applications and available FPGA resources. It supports run-time partial dynamic reconfiguration. After the processor is implemented in an FPGA, it can adjust its issue-width while other circuits in the FPGA keep running. Only a small partial bit-stream is loaded to adjust the issue-slots. Applications/kernels with more fine-grain (instruction level) parallelism such as a Discrete Fourier Transform (DFT) kernel can be run on the larger 4-issue machine for better performance, and applications with more coarse-grain (data level) parallelism such as an encryption algorithm with a specific amount of data can be run on the two 2-issue processors in a Single Instruction Multiple Data (SIMD) fashion for faster execution. It has a freely available tool-chain and a well-defined development framework. This VLIW processor is able to use the inherit ILP of applications. As mentioned ρ–VEX is an embedded reconfigurable and extensible open source VLIW processor, accompanied by a development framework. Its architecture is based on the VEX Instruction Set Architecture (ISA). A software development tool-chain for VEX is made freely available by Hewlett-Packard (HP). As the bandwidth and traffic is increasing and varying a lot in networks, we need reconfigurable processors as a powerful, parameterizable and flexible platform. ρ–VEX has all the requirements of processing network applications.

The main task in network devices is packet classification. Packet classification is the process of classifying different packets into different flows (classes) based on applying rules on their headers. Traditionally, routers have forwarded packets based only on the destination address in the packet. As the Internet begins to be used for commercial applications, service providers would like routers to provide "service differentiation". Routers with a packet classification capability, however, can distinguish traffic based on destination, source, and application type. Such classification allows various forms of

service differentiation. Despite the progress made in the last year on solutions to the packet classification problem, existing routers need to be more powerful and faster due to mentioned increases in networks traffic and bandwidth. Increasingly, however, users are demanding, and some router vendors are providing, a more discriminating form of router forwarding, called Layer 4 switching (service differentiation). Traditional routers do not provide service differentiation because they treat all traffic going to a particular Internet address in the same way. Packet classification enables routers to support various value-added services, such as blocking traffic from insecure sites, giving preferential treatment to premium traffic, routing based on traffic type and source, firewall packet filtering , policy routing , Virtual Private Network (VPN) implementations , traffic billing, and Quality of Service (QoS) applications. In order to provide these services, the router needs to classify the packets into flows according to different criteria. These criteria form rules which are based on L2/L3/L4 fields in the packet header. Routers classify arriving packets by comparing them to a set of predefined rules and finding the highest priority rule or the rule that best matches the packet header fields (the Best Matching Rule (BMR)). A rule consists of a set of fields made up of the IP source prefix, the IP destination prefix, the source port range, the destination port range, and the protocol type and flags. The difficulty of packet classification is in performing multiple field lookups at wire speed for every incoming packet, given that the packet arrival rate can be several million packets per second. For example for a 40 Gigabit per second (OC768) wire speed, given the smallest packet size of 40 bytes in the worst case, the router needs to look up packets at a speed of 125 million packets per second. That, together with other needs in processing a packet, amounts to less than 8 ns per packet lookup. Nowadays, one access to on-chip memory takes 1–5 ns for SRAM and about 10 ns for DRAM. One access to off-chip memory takes 10–20 ns for SRAM and 60–100 ns for DRAM. These figures show that it is highly demanding to develop high-speed packet classification algorithms. It also shows that it is very difficult for serial algorithms to achieve ideal wire speed. The speed of a packet classification algorithm is measured by the times of memory access. Other performance metrics for a packet classification algorithm include memory storage requirements, update speed, and scalability to both the number of the rules and the number of fields included in the rules, among possible others. There are lots of algorithms for packet classification and every one of them has used different methods. Most of them use high bandwidth and a small on-chip memory, while locating the rule database in a slower and higher capacity off-chip memory. Many packet classification algorithms, such as tuple space pruning, cross-producing and coarse-grained tuple space perform a separate lookup on each field to narrow the search space. Hence, these algorithms cause off-chip memory accesses for both the individual field lookups and the final combined lookup.

With the mention increases and dynamicity in traffic and bandwidth we need faster, more efficient and totally scalable algorithms. For the purpose of scalability and economical reasonability, we gave more attention to methods that have taken advantage of Bloom filter. In [9] it is proposed to replace each field lookup with an on-chip Bloom filter and add a new tuple Bloom filter to further reduce unnecessary off-chip memory accesses. This algorithm has very small memory requirements. It scales well from two-dimensional classifiers to high-dimensional ones. In particular, this algorithm is inherently parallel. It is easy to exploit the parallel mechanism in the hardware.

While the general solution is geared towards hardware parallelism and high speed memories, a proper combination of existing fast and reconfigurable platforms with scalable and space efficient packet classification algorithm that has fast update time can benefit routers and any applications of packet classification. In this thesis we provide a background of the requirements and the rationale behind implementing Bloom filter based

packet classification algorithms on a reconfigurable and parameterizable VLIW soft-core processor, called ρ–VEX. We also evaluate this mechanism and compare the results with some other platforms. As ρ–VEX is an open source, reconfigurable, flexible and powerful VLIW processor and Bloom filter is a parallel and space efficient data structure, we expect faster and more powerful platform for packet classification with a proper combination of them.

1.4 Project Goals

The main goal of this project is to implement one of the vital tasks of network processing, packet classification, on extensible and reconfigurable VLIW processor. To accomplish this goal we searched among the algorithms of packet classification for the most suitable one. Because of the rapid growth and inherit dynamicity of network traffic, Bloom filter base algorithms have been chosen. Tuple pruning using Bloom filters for packet classification is the best Bloom filter base packet classification algorithm which is accurately fit our goal. The following stages have been identified in order to achieve the main goal of this project:

- For the first stage, ρ–VEX should be implemented on FPGA.
- After that, the C code of the tuple pruning using Bloom filters for packet classification should be written.
- Then the implemented software platform should be mapped on the hardware platform, and the provided results should be compared with other hardware platforms for the acquired throughput, speedup and improvement.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. A background on underlying technologies as network processing, packet classification, Bloom filter and ρ–VEX processor is presented in Chapter 2. The implementation of our platform and required C code is presented in Chapter 3. Chapter 4 discusses the performance results of our packet classification and configuration analysis of p-VEX serving as a configurable and extensible processor. Finally, this thesis is concluded in Chapter 5. The VHDL code of our proposed One Clock CRC is presented in Appendix A. The C codes of our application and the VEX machine model for ρ–VEX can be found in Appendix B and C respectively.

Chapter 2
Background

2.1 Introduction

Recently, high speed of progress in network related applications caused exponential increment in network traffic. Internet at its beginning was only text emailing and webs but it encountered to drastic changes. For this complex and heavier traffic than early days of networks, demands on more powerful devices, which are able to handle this new situation, have been emerged. Due to dynamic nature of network traffic, platforms that are able to cope with this dynamicity are needed. FPGAs could be efficient candidates as they contain hardware programmable logic units that can be reconfigured based on the application requirements, sometimes FPGAs referred to as reconfigurable platforms.

ρ–VEX is a reconfigurable soft-core VLIW processor in an FPGA. It is parametric and parameters such as type and the number of FUs, supported instructions, memory bandwidth and register file size can vary based on the variation of the applications and available FPGA resources. It has a freely available tool-chain and a well-defined development framework. This VLIW processor is able to use the inherit ILP of applications. ρ–VEX has all the requirements of processing network applications. Moreover, the main task in network devices is packet classification. Packet classification is the process of classifying different packets into different flows (classes) based on applying rules on their headers. As the bandwidth and traffic is increasing and varying a lot in networks, reconfigurable processors as a powerful, parameterisable and flexible platform are needed. Among lots of packet classification algorithms, with the mention increases in traffic and bandwidth totally scalable methods are more in demand. For the purpose of scalability and economical reasonability, more attention to methods that have taken advantage of Bloom filter has been paid.

In this Chapter we are going to provide a background and the rationale behind packet classification, Bloom filter, Bloom filter based packet classification algorithms and a reconfigurable and parameterizable VLIW soft-core processor, called ρ–VEX (As it is an open source, reconfigurable, flexible and powerful VLIW processor).

Section 2.2 presents an overview of network processing. Section 2.3 discusses processing platforms. Bloom filter is presented in Section 2.4. Subsequently, application of Bloom filters is presented in Section 2.5. Fundamentals of packet classification are provided in section 2.6. Section 2.7 describes ρ–VEX. Some related work has been discussed in section 2.8. Finally, this chapter is concluded in Section 2.9.

2.2 Network Processing

Network processing is the acted operations on packets by network devices such as switches and routers. Packets have one or more headers that contain information such as source/destination addresses, source/destination port number, protocol type, checksum, expiration timer, etc. The protocol declares the number of headers for a packet. TCP/IP is the network protocol, which is in use in different networks. Every layer of TCP/IP adds its own header to the packet. The most common processing task is packet classification. It enables routers to provide differential services such as routing based on the traffic type and