Network Switch Design on Field-Programmable Gate Arrays

CS351 Computer Systems Engineering Project

Specification

Benji Levine
1611586

Superviser: Dr Suhaib Fahmy

Department of Computer Science
University of Warwick

2018-19

Contents

1 Glossary

The following acronyms are used throughout the specification.

- **FPGA**: Field-Programmable Gate Array
- **ASIC**: Application-Specific Integrated Circuit
- **LUT**: Look-Up Table
- **MAC Address**: Media Access Control Address
- **OSI model**: Open Systems Interconnection model
- **TCP SYN flooding**: Media Access Control Address
- **PCIe**: Peripheral Component Interconnect Express
- **SRAM**: Standard Random-Access Memory
- **DRAM**: Dynamic Random-Access Memory

2 Problem Statement

Networking is an important area of Computing, and as network sizes and average complexity of projects have increased, the need for network connections with very low latency has also increased. For systems such as Automated Trading Systems [1], a small delay in completing a trade can result in a significant difference in profit. The faster trades can be processed, the greater the potential for increased revenue. Most conventional network switches will have a latency in the range of "hundreds of nanoseconds to tens of microseconds" [2]. In networks with a large amount of data flowing through a network switch every second, reducing this latency could result in a significant improvement to the speed of the network.

Using FPGAs to conduct switching logic can result in a switch with a very high throughput, since this switching logic will be implemented in hardware. The LUTs on an FPGA can be used to store the MAC addresses of the devices connected to the network switch. This allows an FPGA based switch to theoretically determine the appropriate port to direct a packet to, in a single clock cycle. Unlike ASICs, FPGAs can be reconfigured on the fly, so MAC address tables can be dynamically updated while still being implemented in hardware. For all of these reasons, FPGA-based network switches have the potential to create networks with a much higher throughput than traditional networking hardware, and investigation into FPGA-based switches may reveal further opportunity for other FPGA-based networking hardware, such as routers or firewalls.

3 Objectives

This project will consist of four stages: Research, Implementation, Testing, and Evaluation, and the objectives have been separated as such.

3.1 Research

- Research networking concepts (such as the OSI network model)
- Research methods of measuring latency of a network switch
- Research the NetFPGA platform [3]
- Research the packet switching language P4 [4]

3.2 Implementation

- Implement a packet analyser on a NetFPGA
- Implement a packet switcher on a NetFPGA

3.3 Testing

- Test throughput and latency of switching packets using the NetFPGA packet switcher
- Test throughput and latency of switching packets using a conventional network switch

3.4 Evaluation

- Compare performance of the NetFPGA packet switcher to a conventional network switch
- Write up performance comparison

4 Requirements

In order to measure the success of this project and to clearly define the work to be done, the following requirements have been written to support the objectives above. Since the research stage and initial implementation stage will be used to confirm the direction for the remainder project, these requirements are subject to change. They indicate a feasible path through the project which should return interesting valuable results. They have been written using the MoSCoW method [5], which is a prioritisation technique that helps write clearly defined requirements.

4.1 Functional Requirements

These requirements define the technical detail of the system produced over the course of the project, as well as the data to be analysed for the final report.

F1: The system **must** be able to analyse packets at layer 2 of the OSI network model

F2: The system **must** be able to send packets to the correct port based on MAC addresses

F3: The system **must** be able to store MAC address tables

F4: The system **must** be able to dynamically update MAC address tables

F5: The average latency of the system **must** be measured

F6: The throughput of the system **must** be measured

F7: The system **should** be able to analyse packets at layer 3 of the OSI network model

F8: The system **could** be able to analyse packets at layer 7 of the OSI network model

F9: The system **could** be able to detect basic network attacks (such as TCP SYN Flooding [6])

F10: The system **could** implement features of a basic firewall (such as packet filtering [7])

4.2 Non-Functional Requirements

These requirements describe the general operation of the system produced, as well as how the project should be developed

NF1: The system **should** be scalable

NF2: The system **should** be efficient

NF3: All code for the system **should** be well-documented and maintainable

5 Project Management

5.1 Methods

This project will use an Agile methodology so that it can adapt to changes which arise during the project. Since the research and implementation stages of the project will contribute to confirming its direction, this flexibilty is important. In addition, git [8] will be used to track changes in both written documents and code developed, such as P4 or Verilog code. Repositories will be set up using git for the different areas of the project, and these repositories will be stored primarily on an online GitHub [9] server that will be backed up regularly. The Gantt chart in Figure 1 has been constructed to show an outline of the project timetable, and is intentionally flexible for the changes that will take place.

The NetFPGA [3] platform is intended to be used for this project. It is an "open source hardware and software platform designed for research and teaching" [10]. Of the four hardware plaforms available, the NetFPGA 1G [11] is the most suitable for this project, since its hardware is most compatible with general purpose consumer hardware. The other three platforms could be used to create faster circuits than the NetFPGA 1G, however these may require additional hardware, which is not currently available for this project. This platform is based around a *Xilinx Virtex-II Pro 50* [12] which contains 53,136 logic cells and 4kb block RAM. In addition, the NetFPGA 1G contains four Gigabit Ethernet networking ports, 4.5MB of SRAM, and 64MB of DDR2 DRAM. It has a standard PCI form factor, and so is compatible with most consumer motherboards.

5.2 Timetable

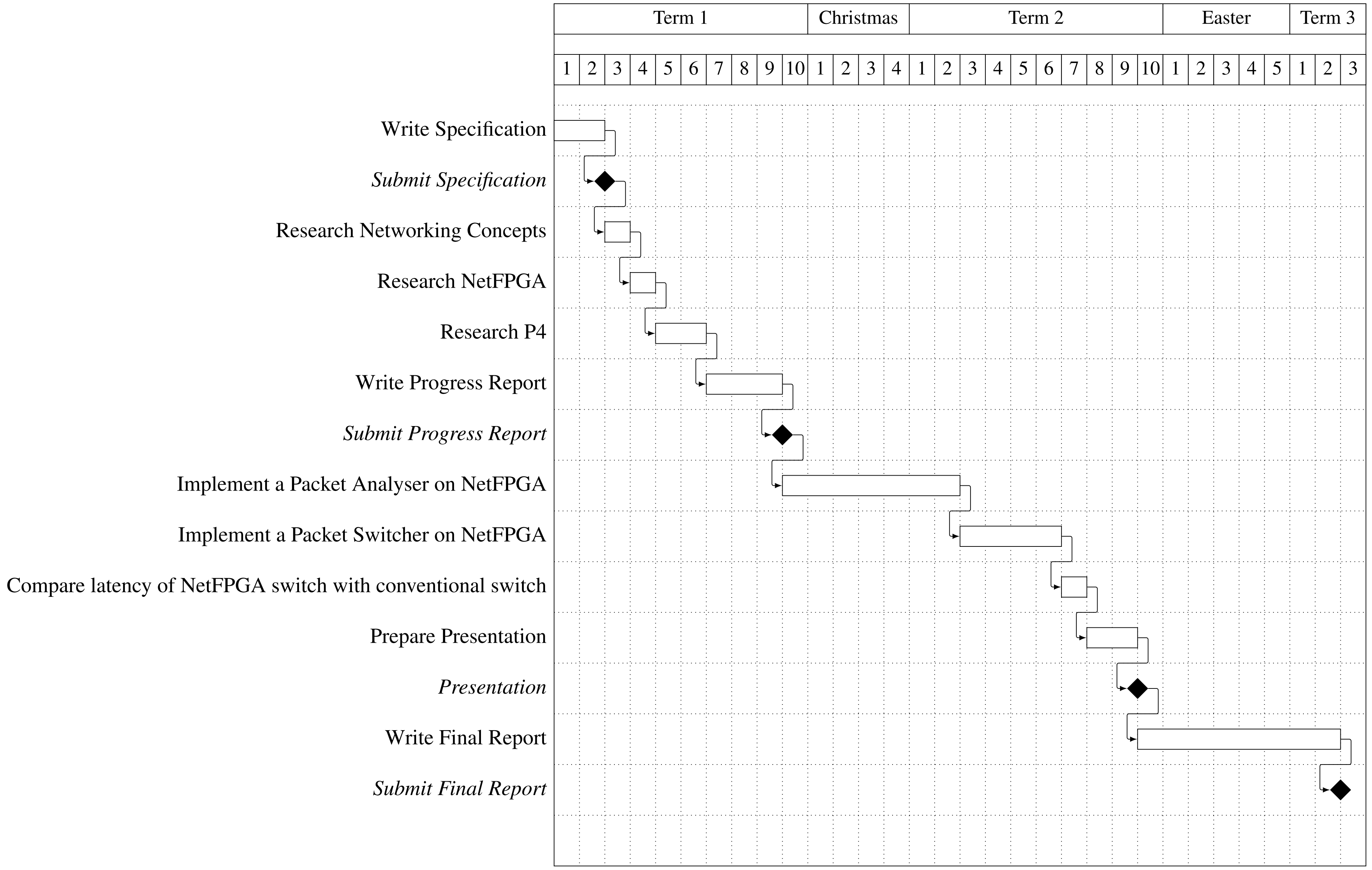

Figure 1: Gantt Chart of Project Timetable

6 Resources

This project will use a number of different resources, including hardware, software, and languages. These are listed below.

- git [8]
  - will be used for version control of all code and documents
- GitHub [9]
  - will be used as an external server to store code and documents, as well as an interface for git
- NetFPGA [3]
  - will be used as the platform on which a switch is developed
- P4 [4]
  - will be used to implement packet switching on the NetFPGA
- LaTeX [13]
  - will be used to write and format all documents
- Atom [14]
  - will be used as an editor to write code

References

[1] J. Folger, "The pros and cons of automated trading systems," *Investopedia*, Aug 2018. Accessed: 2018-10-12.

[2] Plexxi, "Behavior of and Requirements for Internet Firewalls," tech. rep., Jan. 2016.

[3] "Netfpga." `https://netfpga.org/`, Jul 2010. Accessed: 2018-10-08.

[4] "P4 language consortium." `https://p4.org/`, Sep 2014. Accessed: 2018-10-08.

[5] R. M. Sydeek, "Moscow method explained," *Feedough*, Dec 2017. Accessed: 2018-10-12.

[6] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," Tech. Rep. 4987, Aug. 2007.

[7] N. Freed, "Behavior of and Requirements for Internet Firewalls," Tech. Rep. 2979, Oct. 2000.

[8] "git." `https://git-scm.com/`, May 2012. Accessed: 2018-10-08.

[9] "Github." `https://github.com/`, Sep 2009. Accessed: 2018-10-09.

[10] “Netfpga - about.” `https://netfpga.org/site/#/about/`, Jul 2010. Accessed: 2018-10-09.

[11] “Netfpga - netfpga 1g.” `https://netfpga.org/site/#/systems/4netfpga-1g/details/`, Jul 2010. Accessed: 2018-10-09.

[12] Xilinx, *Virtex-II Pro and Virtex-II Pro X Platform FPGAs*, 5 2011. v5.0.

[13] “Latex - a document preparation system.” `https://www.latex-project.org/`, 1983. Accessed: 2018-10-10.

[14] “Atom.” `https://atom.io/`, Jun 2015. Accessed: 2018-10-10.