THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

FPGA packet filter with Ethernet MAC and web server using a RISC-V softcore processor

Project Proposal

Matthew Gilpin

45801600

Semester 1, 2023

The University of Queensland

School of Information Technology and Electrical Engineering

List of Abbreviations

Abbreviations	
IoT	Internet of Things
FPGA	Field Programmable Gate Array
pf	Packet Filter
MAC	Medium Access Control
ISA	Instruction Set Architecture
ASIC	Application Specific Integrated Circuit
SoC	System on Chip
TRL	Technology Readiness Level

Contents

List of Figures

List of Tables

Introduction

This chapter provides the necessary background and reasoning behind the proposed project.

1.1 Background

In a technology age of growing numbers of cyber attacks and record number of connected devices, it's paramount to ensure these devices operate safely and securely. The Australian Cyber Security Center (ACSC) received in excess of 76,000 cybercrime reports and growing in the 2021-22 financial year [1]. The growing trend of Internet of Things (IoT) will provide more opportunity for black hats (malicious attackers). IHS Markit estimates 125 billion IoT devices will be connected by 2030 [2].

To cope with the increase in IoT devices, a common shift to edge computing has evolved in favour over the traditionally more centralised cloud computing architecture. The core principals behind the paradigm is to move the data processing closer geographically to its origin to not only decrease the central load, but to improve latency [3]. Due to the distributed load, smaller and more efficient computers can be used at the edge/perimeter of these networks [3]. Just like any other computer connected to the broader network, these edge networks also need to be protected from bad actors.

Literature review

Some of the concepts behind the proposed project, such as an Ethernet MAC or RISC-V processor are not new, there is a variety of previous works in these areas. This part of the proposal will explore the prior work related to the project.

2.1 Field Programmable Gate Arrays (FPGA)

First introduced by Xilinx in 1984, field programmable gate arrays (FPGAs) allowed for custom logic designs to be recognised without the need for expensive application specific integrated circuits (ASICs) [4].

2.2 RISC-V processor

RISC-V, a reduced instruction set computer architecture, is an open and royalty free instruction set architecture (ISA). As a result, a plethora of soft-core processors have been made. The specific core proposed in this project is the 'NEORV32' RISC-V processor. It's a highly configurable microcontroller like system on chip (SoC) written purely in VHDL.

Due to the open nature of the RISC-V architecture, many designers have made their own System on Chips (SoC) and created their own implementations. RISC-V themselves have published a list[1] of different RISC-V implementations that have a unique architecture ID. The majority of these are either written for application specific integrated circuits (ASICs), written in a hardware description language other than VHDL or has poor documentation with the exception of the *NEORV32 RISC-V* softcore processor. This SoC written in vendor agnostic VHDL and importantly has a considerable amount of documentation.

Being a softcore processor, there is control over which modules are implemented and are not. Some basic features of the *NEORV32 RISC-V* include UART, SPI, and GPIO interfaces [5]. The datasheet [5] also mentions that it supports a *'Wishbone b4'* external bus interface.

A Wishbone B4 (refered to as just 'wishbone') interconnection is designed specifically to connect peices of hardware together on a System-on-Chip (SoC) [6]. In the NEORV32, It allows for external hardware modules to be memory mapped into the the 32bit address space on the processor [5].

2.3 Ethernet MAC

First introduced in 1983 [7], the IEEE 802.3 standardardised a technology, Ethernet, to interconnect devices. There have been many attempts at creating hardware for Ethernet MACs.

[1] See: https://github.com/riscv/riscv-isa-manual/blob/master/marchid.md

The IEEE 802.3 standard [7], more commonly known by the name of 'Ethernet' defines the *'Medium Access Control'* (MAC) protocol amongst other things for two or more devices to communicate over a network. This standard is just one part in the layered network models such as the OSI model and TCP/IP models.

The decision in creating a custom MAC might be considered as interesting given the range of pre-existing Intellectual Property (IP) cores for Ethernet on FPGAs. The issue with the pre-existing solutions only have a single output to connect to something like a softcore processor. To create a firewall, the network traffic would need to pass through the processor. To decrease latency, a second interface can be added to the MAC to allow traffic to flow through a hardware-based firewall. This is analogous to the direct memory access (DMA) controller on most modern microprocessors.

2.4 Packet Filter Firewall

Usually, the first line of defence against bad actors, it is a vital component in a computer network and can become vastly complex. There are several types of Firewalls such as packet filters (pf), stateful packet firewalls and application firewalls [8]. Firewalls can also perform other tasks and employ other techniques to secure a network, however, in this project the most basic pf-style firewall will be implemented. Packet filters are considered as stateless and traditionally only filter on the fields in the headers in the network (layer 2) and transport (layer 3) layers [8]. Such fields include IP addresses, port numbers and protocol type.

More advanced firewalls can perform deep packet inspection and explore the contents of the higher layers to better evaluate a packets true intention. While there is provision to add this functionality on an FPGA based firewall, this will not be explored in this project due to its significant increase in complexity.

2.5 Web servers

... The LwIP library is a popular lightweight TCP/IP stack which has been investigated in a plethora of reaserch papers and projects.

Recently, FreeRTOS have published their *FreeRTOS-Plus-TCP* library which aims to provide ... over the LwIP stack.

Topic Definition

3.1 Topic

While there is no single solution that will fully protect an edge network, a common and effective way to reduce the unauthorised/unwanted network traffic is by simply filtering out the potentially malicious packets. While this may seem overly complex, in reality a few simple rules can be followed to decide on whether to forward or deny/drop/block packets from entering or exiting a network. These packet filters (pf) are a type of firewall that do not follow any complex rules and keep state between packets or use deep packet inspection to check the contents of the payload to ensure it's not malicious.

Packet filters are considered as stateless and traditionally only filter based on the fields in the headers at the network (layer 2) and transport (layer 3) layers [8]. Such fields include IP addresses, port numbers and protocol type.

The proposed project consists of making a hardware implementation of a pf with custom Ethernet Media Access Controllers (MAC) connected to a hardware based filtering block which is all controlled by a RISC-V softcore processor. This will then also have a web interface so that a user can configure the rules for the pf.

3.2 System Overview

3.3 Aims

The aims of the proposed FPGA Ethernet controller and web interface on a RISC-V processor are:

- Increase security to edge IoT networks,
- Increase the power efficiency for wire-speed firewalls, and
- Decrease the latency for packet filter firewalls.

3.4 Establishing Exclusions

While the proposed project will reduce the likelihood of network based attacks it is not a *'one size fits all'* solution. By the nature of the IoT and edge network ecosystem, there are a myriad of different attack vectors where not all of them will be detectable at the network level.

The proposed project will **not**

- Protect against all attacks,

- Be able to protect against all IoT devices, or
- Perform routing

3.5 Performance Indicators

3.6 Required Equiptment

While the hardware design will be developed in such a way that it is vendor agnostic, to test the design a Digilent Nexys A7-100T developement board will be used. Importantly, this board has a RJ45 connector and LAN8720 RMII interface chip which allows for a regular fast ethernet connection to be directly connected to the FPGA board. An additional LAN8720 ETH board from Waveshare is also required to obtain the secondary interface.

To validate the functionality and effectiveness of the design, it will be compared with a Raspberry Pi Compute Module 4 (CM4) with a Waveshare CM4-DUAL-ETH-MINI daughterboard which contains two 1GbE interfaces. This will act as a baseline.

3.7 Technology Readiness Level

One method for estimating the degree of maturity for a technical project is by using the *Technology Readiness Levels (TRL)* benchmark. Within this, there are 9 levels each indicating a different phases of a design. This project is intended to reach a TRL of 6. These six different levels and their relevance to this project are as follows.

1. **TRL 1: Basic Research** - The concepts are researched and a base understanding of the system is gathered,
2. **TRL 2: Applied Research** - Detailed research is conducted into each part of the project,
3. **TRL 3: Proof of Concept** - A cutdown version of the final project protoype that highlights the core functionality or subsystems working,
4. **TRL 4: Lab Testing of Prototype** - Prototype of the core design with majority of the functionality working,
5. **TRL 5: Testing of Integrated system** - Refined prototype that works as intended but may be incomplete, and
6. **TRL 6: Prototype System Verified** - Comparison to pre-existing solutions and verification.

Timeline and Plan

This section of the report details the plan and timeline of the proposed project. It also details the necessary risk assessment.

4.1 Milestones

The TRL benchmark provides a breakdown of the phases of developement, the milestones table 4.2 below highlights the different design stages and tasks throughout the project. The expected durations of these are also presented.

Task	Details	Duration
Create MAC	Create custom Layer 2 Ethernet hardware based on the IEEE 802.3 standard	3-4 Weeks
Wishbone Interface	Connect the Ethernet MAC to the NEORV32 RISC-V Processor using the wishbone interface and access it via software	2 Weels
Webserver	Create and Get the webserver working on the NEORV32 Processor. Web page should be accessed from another computer	5-6 Weeks
Firewall Hardware	Create the hardware between 2 Ethernet MACs to filter out packets based on rules	3-4 Weeks
Integration with software	Add functionality to the server to be able to configure the firewall rules	1 Week
Measure and Compare	Compare to pre-existing solutions	1 Week

Table 4.1: Milestones for the proposed project

4.2 Project Risk Assessment

The majority of the work compeleted in the proposed project is digital and poses little risk outside of the standard office sitting.

Risk	Severity	Likelyhood	Mitigation
Licensing	Minor	Moderate	Avoid software/hardware that requires a specific license.
Data loss	Catastrophic	Unlikely	Ensure all items are backed-up to the cloud and use services such as GitHub where appropriate. Employ a 3 2 1 backup strategy
Hardware Failure	Moderate	Unlikely	Double check all connections to the FPGA board before powering. Reduce excessive handling where necessary to minimise risk of damaging the equiptment
Illness	High	Likely	Take breaks periodically to avoid being over-worked, and take necessary recovery steps if sick.
Missed Deadlines	Major	Likely	Ensure plans are followed and compelete tasks as soon as possible. If behind, spend extra time on project to catch up.

Table 4.2: Risk assessment of proposed project

Bibliography

[1] A. C. S. Center, "Acsc annual cyber threat report, july 2021 to june 2022," Nov 2022.

[2] IHS, "The internet of things: a movement, not a market," tech. rep., IHS Markit, 2017.

[3] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi, "Edge computing perspectives: Architectures, technologies, and open security issues," in *2019 IEEE International Conference on Edge Computing (EDGE)*, pp. 116–123, IEEE, 2019.

[4] S. M. Trimberger, "Three ages of fpgas: A retrospective on the first thirty years of fpga technology," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 318–331, 2015.

[5] Stephan Nolting (M.Sc.), *The NEORV32 RISC-V Processor*, 2023. v1.8.1-r17-gd1b295de.

[6] "Wishbone B4 SoC Interconnection," standard, OpenCores, Dec. 2010.

[7] "IEEE Standard for Ethernet," standard, The Institute of Electrical and Electronics Engineers, Inc., New York, USA, Dec. 2012.

[8] E. W. Fulp, "Chapter e74 - firewalls," in *Computer and Information Security Handbook*, pp. e219–e237, Elsevier Inc, third edition ed., 2017.