



**FUNDAMENTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

ETHERNET FRAME PARSER

GROUP AP-08

Darmawan Hanif	2206829175
Fabio Rabbani Prasetyo	2206829490
Fathia Zulfa Alfajr	2206030501
Kevin Naufal Aryanto	2206062850

PREFACE

Puji syukur kami panjatkan ke Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia-Nya sehingga kami dapat menyelesaikan proyek "Ethernet Frame Parser." Proyek ini kami buat dalam rangka memenuhi tugas akhir dari mata kuliah Perancangan Sistem Digital Tahun Ajaran 2023/2024.

Kami juga ingin berterima kasih sebesar-besarnya atas dukungan dari seluruh pihak yang terlibat selama pelaksanaan proyek ini. Tidak lupa, rasa terima kasih kami juga sampaikan khususnya kepada para asisten laboratorium yang telah memberikan arahan serta meluangkan waktu untuk membimbing kami dalam langkah-langkah pengembangan proyek ini.

Terakhir, kami juga menyampaikan permohonan maaf apabila terdapat kekurangan atau keterbatasan dalam pelaksanaan proyek ini. Kami menyadari bahwa setiap proyek memiliki tantangan dan aspek yang dapat diperbaiki, sehingga kami menghargai setiap masukan dan kritik konstruktif terhadap proyek yang sudah kami buat.

Depok, December 22, 2023

Group AP-08

TABLE OF CONTENTS

CHAPTER 1: INRODUCTION

- 1.1 Background
- 1.2 Project Description
- 1.3 Objectives
- 1.4 Roles and Responsibilities

CHAPTER 2: IMPLEMENTATION

- 2.1 Equipment
- 2.2 Implementation

CHAPTER 3: TESTING AND ANALYSIS

- 3.1 Testing
- 3.2 Result
- 3.3 Analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

- Appendix A: Project Schematic
- Appendix B: Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Melalui mata kuliah Jaringan Komputer, kami menyadari bahwa jaringan Ethernet merupakan salah satu prinsip fundamental dalam proses *data transmission*. Pemahaman komprehensif terhadap struktur data frame Ethernet menjadi hal yang cukup krusial dalam pengimplementasian sistem jaringan komputer. Proyek "Ethernet Frame Parser" ini bertujuan untuk merancang serta mengimplementasikan parser berbasis bahasa VHDL untuk menganalisis dan mengurai frame Ethernet sesuai standar yang berlaku sesuai apa yang sudah kami pelajari.

Kami juga menyadari bahwasanya salah satu permasalahan umum yang sering dihadapi oleh masyarakat secara general adalah keamanan informasi dan privasi. Dalam konteks ini, kami menyaksikan munculnya insiden-insiden serupa terkait pencurian data maupun akses yang tidak sah ke *private system*. Penanganan data sensitif di dalam jaringan, termasuk MAC *address* dalam jaringan Ethernet, menjadi krusial dalam melindungi integritas dan keamanan informasi. Maka dari itu, fitur berupa *intrusion verification* kami tambahkan dalam proyek kami dalam rangka menunjang *concern* kami terhadap permasalahan tersebut. Dengan begitu, proyek kami dibuat dalam rangka menciptakan langkah proaktif dalam memahami, mengoptimalkan, serta menciptakan lingkungan yang aman pada proses *data transmission* dalam jaringan Ethernet.

1.2 PROJECT DESCRIPTION

Proyek ini adalah bentuk pengembangan Ethernet Frame Parser menggunakan VHDL, dimana parser sendiri merupakan komponen kritis dalam penganalisisan ethernet frame dalam konteks jaringan. Fokus utama dari proyek ini ialah menginterpretasi suatu ethernet frame sehingga kita dapat mengurai informasi mengenai MAC address dari destination MAC maupun source MAC, EtherType, Frame Check Sequence (FCS), serta *payload data* dalam frame tersebut. Dalam merealisasikan hal tersebut, kami juga mengadopsi metode *state machine* untuk membaca, memproses, dan mengekstrak informasi relevan dari setiap

Ethernet *frame* yang diterima. Kami menyusun *state machine* dalam lima kondisi utama, yaitu IDLE, PARSE, FETCH, DETECT, dan COMPLETE, dimana kelima kondisi ini akan memastikan alur pemrosesan interpretasi *frame* yang sistematis.

Selain itu, kami juga menambahkan fitur *intrusion verification* yang berguna untuk memeriksa apakah MAC *address* yang terdapat dalam frame merupakan alamat yang diizinkan atau bukan. Apabila MAC *address* dalam *frame* tidak sesuai dengan alamat yang diizinkan, maka akan memicu respons yang direalisasikan dengan pengaktifan *intrusion_detected_flag*, karena sistem menganggapnya sebagai sebuah potensi serangan siber. *Intrusion_detected_flag* kemudian akan memberikan indikasi adanya potensi pelanggaran keamanan.

1.3 OBJECTIVES

Tujuan dari proyek ini adalah sebagai berikut:

1. Pemenuhan tugas akhir dalam mata kuliah Perancangan Sistem Digital
2. Pengimplementasian Pemrograman VHDL
3. Perancangan Ethernet Frame Parser yang dapat menyediakan informasi terkait MAC *address*, EtherType, FCS, dan *payload data* dalam suatu *frame*.
4. Pengaplikasian sistem *intrusion verification* dalam Ethernet Frame Parser yang dapat menunjang keamanan jaringan.

1.4 ROLES AND RESPONSIBILITIES

Peran dan tanggung jawab dari setiap anggota kelompok adalah sebagai berikut:

Roles	Responsibilities	Person
Role 1	Darmawan Hanif	Person 1
Role 2	Fabio Rabbani Prasetyo	Person 2
Role 3	Fathia Zulfa Alfajr	Person 3
Role 4	Kevin Naufal Aryanto	Person 4

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

Tools yang akan digunakan dalam proyek ini adalah sebagai berikut:

- Visual Studio Code
- ModelSim
- QuartusPrime

2.2 IMPLEMENTATION

Pada pengerjaan Proyek Akhir Perancangan Sistem Digital ini, kami mengimplementasikan konsep-konsep dasar yang sudah kami pelajari, yaitu:

Behavioural Style

Kode kami menggunakan *behavioral style* dengan mendefinisikan *state machine* yang mengubah nilai variabel dan sinyal berdasarkan kondisi tertentu. Aspek-aspek yang menunjukkan implementasi *behavioural style* dalam kode kami adalah:

- Penggunaan *process* sebagai deskripsi perilaku,
- Penggunaan *state machine*,
- Penggunaan *case statement* untuk kontrol alur,
- Adanya perubahan *state* pada kondisi *rising edge clock*,
- Penggunaan Variabel dan Sinyal,
- Implementasi *intrusion detection* sebagai bagian dari *behaviour*.

Finite State Machine

Dalam kode kami, *finite state machine* digunakan untuk mengontrol alur kerja modul Ethernet Frame Parser. FSM membantu menggambarkan dan mengatur bagaimana modul berperilaku dalam merespons input tertentu. Aspek-aspek yang menunjukkan pengimplementasian FSM dalam kode kami adalah:

- **Penggunaan tipe data 'StateType' untuk mendefinisikan keadaan.** Dalam konteks kode kami, kami mendefinisikan lima keadaan, yaitu: IDLE, PARSE, FETCH, DETECT, serta COMPLETE.
- **Penggunaan logika untuk setiap keadaan.** Setiap keadaan memiliki logika kontrol yang berbeda untuk mengatur perilaku sistem. Contohnya, keadaan PARSE_HEADER digunakan untuk menguraikan header dan melakukan *intrusion verification*.
- Adanya pergantian keadaan berdasarkan input dan keadaan saat ini.
- Adanya pergantian keadaan pada *rising edge clock*.

Concurrent Circuit Design

Kode kami juga menggunakan *concurrent circuit design*, dimana hal tersebut dapat dilihat dari penggunaan proses dan penggunaan sinyal yang dieksekusi secara paralel untuk mewakili keadaan yang berbeda. Aspek-aspek yang menunjukkan pengimplementasian *concurrent circuit design* dalam kode kami adalah:

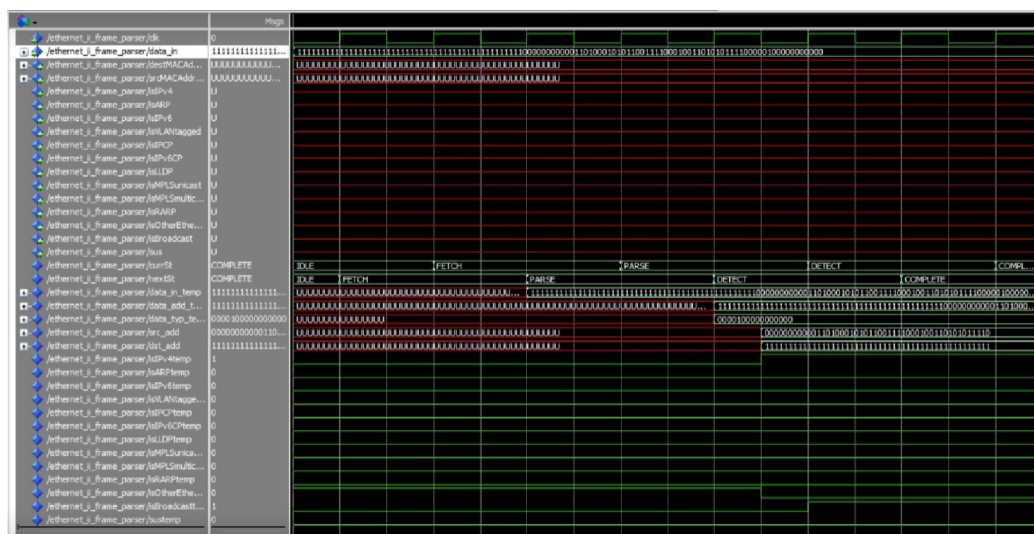
- **Penggunaan proses untuk mendeskripsikan logika sirkuit yang berjalan secara paralel.** Setiap perubahan keadaan dalam state machine terjadi dalam proses yang sama, yang diaktifkan pada sinyal *rising edge clock*.
- **Penggunaan sinyal digunakan untuk merepresentasikan nilai-nilai yang dapat berubah secara paralel.** Contohnya, sinyal 'state' digunakan untuk menyimpan keadaan *state machine* yang dapat berubah secara independen dari input lainnya. Selain itu, terdapat penggunaan sinyal-sinyal lain seperti *dest_mac_reg*, *src_mac_reg*, dan sebagainya yang menyimpan nilai-nilai yang dapat berubah secara paralel.
- Penggunaan sinyal *intrusion_detected_flag* untuk menyimpan hasil deteksi intrusi yang dapat berubah secara paralel tergantung pada kondisi yang diuji.

TESTING AND ANALYSIS

3.1 TESTING

Cara kerja dari kode VHDL "Ethernet_II_Frame_Parser" ini adalah kode ini beroperasi sebagai *state machine* yang dikontrol oleh *rising edge clock*. Pada awalnya, *state machine* berada dalam *state* 'IDLE.' Ketika input *data_in* terdeteksi bukan nol, maka *state machine* beralih ke *state* 'FETCH.' Selanjutnya, *data frame* Ethernet II disalin ke *data_in_temp* untuk proses lebih lanjut. Pada *state* 'PARSE,' *MAC address* dan *EtherType* dipisahkan dari *data frame* dan disimpan dalam '*data_add_temp*' dan '*data_typ_temp*.' *State machine* kemudian memasuki *state* 'DETECT,' di mana modul *Intrusion Detection System* (IDS) menentukan apakah *frame* tersebut mengandung informasi yang mencurigakan, seperti siaran *broadcast* atau protokol ARP/RARP atau tidak. Hasil deteksi kemudian akan disimpan dalam sinyal '*sus_temp*.' Pada akhirnya, pada *state* 'COMPLETE,' *output* disesuaikan dengan hasil *parsing* dan deteksi yang disimpan dalam berbagai sinyal. Setelah itu, *state machine* akan kembali ke *state* 'IDLE' dan siap untuk mengolah *frame* Ethernet II berikutnya pada *rising edge clock* berikutnya. Dengan *approach* ini, kode berhasil mengimplementasikan *parser* yang efisien dalam menganalisis struktur *frame* Ethernet II. Selanjutnya, *testing* akan dilakukan pada *ModelSim simulation* yang hasilnya akan ditampilkan dalam 3.2.

3.2 RESULT



3.3 ANALYSIS

Kode Ethernet_II_Frame_Parser merupakan implementasi *parser* untuk frame Ethernet II menggunakan VHDL. Dengan menerapkan *state machine* dan komponen-komponen seperti 'MAC_Address_Parser', 'EtherType_Parser', dan 'IDS', modul ini dapat mengurai data input yang mewakili informasi terkait frame Ethernet II.

Proses *parsing* dimulai saat mendeteksi sinyal 'data_in' yang bukan nol. Input berupa frame Ethernet II dengan panjang bit sebesar 12144. Data bit ini mencakup *header*, *payload*, dan CRC. Dalam siklus *clock* yang ditentukan oleh sinyal clk, MAC *address* dipisahkan menggunakan 'MAC_Address_Parser', jenis protokol Ethernet diidentifikasi menggunakan 'EtherType_Parser', dan *intrusion* dideteksi menggunakan IDS. Hasil output dari setiap komponen digunakan untuk membentuk output akhir, seperti 'payloadAndCRC', 'destMACAddress', 'srcMACAddress', dan status protokol seperti 'isIPv4', 'isARP', dll.

Kemudian, pada tahap *testing*, kami menggunakan input "111111111111111111111111111111111111110000000000011010001010110011110001001101010111100000100000000000", didapat bahwa modul ini memberikan output yang mencerminkan hasil *parsing* dari data input tersebut. Beberapa sinyal output yang akan keluar adalah sebagai berikut:

- destMACAddress adalah 110100010101100111100010011
- srcMACAddress adalah 00000000000000000000000000000000
- isIPv4 adalah 1 (ya)
- isARP adalah 0 (tidak)
- isIPv6 adalah 0 (tidak)
- isVLANtagged adalah 0 (tidak)
- isIPCP adalah 0 (tidak)
- isIPv6CP adalah 0 (tidak)
- isLLDP adalah 0 (tidak)
- isMPLSunicast adalah 0 (tidak)
- isMPLSmulticast adalah 0 (tidak)
- isRARP adalah 0 (tidak)
- isOtherEthertype adalah 0 (tidak)
- isBroadcast adalah 0 (tidak)

- sus adalah 0 (tidak)

CHAPTER 4

CONCLUSION

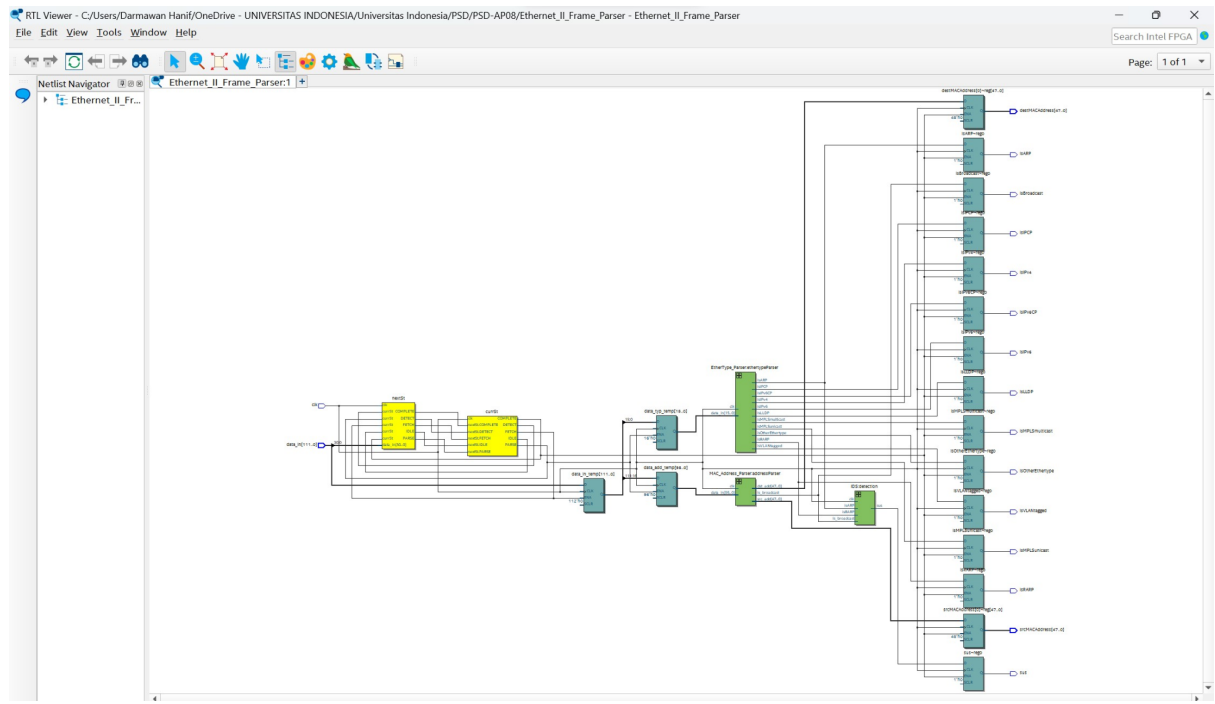
Kode dari proyek Ethernet Frame Parser yang sudah kami ciptakan berfungsi untuk mengidentifikasi jenis sinyal data yang diterima pada alamat MAC, seperti IPv4, ARP, IPv6, VLAN, dan lain-lain. Kode juga mengevaluasi aktivitas abnormal yang berpotensi memicu *intrusion* pada sinyal data tersebut. Secara umum, kode ini dapat digunakan sebagai komponen dari jaringan datar untuk mengidentifikasi jenis sinyal data yang masuk dan memberikan informasi yang relevan untuk mengendalikan jaringan dan menangani situasi abnormal yang terjadi.

REFERENCES

- [1] Reference 1
- [2] Reference 2
- [3] Reference 3
- [4] Reference 4
- [5] Reference 5
- [6] Reference 6
- [7] And so on

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

Put the documentation (photos) during the making of the project