

ETHERNET FRAME VALIDATOR

Group PA04



GROUP PA04

Christian Hadiwijaya 2306161952

Fido Wahyu Choirulinsan 2306250674

Muhammad Hilmy Mahardika 2306267006

Raddieff Ezra Satrio Andaru 2306250693

[Home](#)[About](#)[Contact](#)

Ethernet Frame Validator

Proyek Validator Frame Ethernet kami bertujuan untuk memvalidasi frame Ethernet dengan memeriksa komponen penting seperti CRC, tipe, panjang, alamat MAC, payload, dan preamble. Untuk memastikan bahwa data yang diterima sesuai dengan spesifikasi dan bebas dari kesalahan, sistem ini terdiri dari beberapa modul VHDL. Karena pentingnya keandalan data yang dikirim dalam sistem komunikasi jaringan modern, proses validasi diperlukan untuk memastikan bahwa data tidak rusak dan tetap utuh selama transmisi. Sistem berbasis VHDL yang dibangun dalam proyek ini akan memeriksa elemen penting frame seperti preamble, alamat MAC, panjang frame, payload, dan integritas data dengan menggunakan CRC.

[Read More](#)

[Home](#)[About](#)[Contact](#)

Tujuan

**Memahami Struktur
Frame Ethernet**

**Membangun Sistem
Penerimaan Data
Secara Serial**

**Implementasi Buffer
untuk Pemrosesan
Data**

**Menggunakan Metode
CRC-32 untuk Validasi
Checksum**

**Menjamin Validasi
yang Akurat**

**Mengintegrasikan
Perangkat Keras dan
Lunak**

[Home](#)[About](#)[Contact](#)

Implementation

EQUIPMENT

- **Modelsim**
- **Quartus**
- **Visual Studio Code**
- **Git**

Desain

- **Dataflow:** Untuk menghitung CRC.
- **Behavioral:** Untuk validasi panjang dan tipe frame.
- **Structural:** Untuk mengintegrasikan semua modul.
- **Finite State Machine (FSM):**
Digunakan pada modul preamble-check

[Home](#)[About](#)[Contact](#)

Ethernet Frame Parts

Preamble

Start of Frame Delimiter
(SFD)

Alamat MAC

Tipe/Panjang Payload

Payload

CRC



Preamble

Preamble adalah bagian dari frame Ethernet yang terdiri dari 7 byte yang berisi pola bit 10101010. Preamble digunakan untuk memberikan waktu bagi perangkat jaringan untuk melakukan sinkronisasi bit. Pada proyek ini, preamble akan diperiksa apakah terdiri dari pola bit 10101010 atau tidak.

- Jika data yang diterima (data_in) cocok dengan Preamble Pattern, status sistem berubah ke PREAMBLE_CHECK.
- preamble_count diinisialisasi untuk menghitung jumlah preamble yang valid, dan sinyal preamble_detected diaktifkan ('1'), menandakan preamble telah ditemukan

```
CONSTANT PREAMBLE_PATTERN : STD_LOGIC_VECTOR(7 DOWNTO 0) := "10101010";
CONSTANT SFD_PATTERN : STD_LOGIC_VECTOR(7 DOWNTO 0) := "10101011";

IF data_in = PREAMBLE_PATTERN THEN
    current_state <= PREAMBLE_CHECK;
    preamble_count <= 1;
    preamble_detected <= '1';
    sfd_detected <= '0';
END IF;
```



Start of Frame Delimiter (SFD)

SFD adalah bagian dari frame Ethernet yang terdiri dari 1 byte yang berisi pola bit 10101011. SFD digunakan untuk menandai akhir dari preamble dan awal dari alamat MAC penerima

- Start of Frame Delimiter (SFD) adalah byte khusus 10101011 yang menunjukkan awal data frame.
- Jika data_in cocok dengan SFD Pattern, status sistem berubah ke SFD_CHECK, dan sinyal sfd_detected diaktifkan ('1'), menandakan SFD telah ditemukan.

```
IF data_in = SFD_PATTERN THEN
    current_state <= SFD_CHECK;
    sfd_detected <= '1';
END IF;
```



Alamat MAC

Alamat MAC adalah bagian dari frame Ethernet yang terdiri dari 6 byte yang berisi alamat MAC penerima dan pengirim

- Bagian ini membaca alamat MAC (6 byte) dari frame Ethernet.
- Byte yang diterima dimasukkan ke dalam current_mac secara berurutan.
- Ketika 6 byte telah diterima (`mac_byte_count = 4`), sinyal `mac_read_complete` dan `mac_valid` diaktifkan, menandakan bahwa alamat MAC telah berhasil dibaca.

```
IF data_in_valid = '1' AND enable_mac = '1' THEN
    -- Geser MAC dan masukkan byte baru
    current_mac <= current_mac(39 DOWNTO 0) & data_in;

    -- Setelah 6 byte (MAC lengkap)
IF mac_byte_count = 4 THEN
    mac_read_complete <= '1';
    mac_valid <= '1';
ELSE
    mac_byte_count <= mac_byte_count + 1;
    mac_read_complete <= '0';
END IF;
```



Tipe/Panjang Payload

Tipe/Panjang Payload adalah bagian dari frame Ethernet yang terdiri dari 2 byte yang berisi tipe atau panjang payload yang akan diterima.

- Bagian ini membaca field Length/Type pada frame Ethernet, yang berisi informasi panjang payload atau tipe protokol.
- Data diterima dalam dua byte, yang disusun menjadi current_length_type.
- Setelah dua byte diterima (length_type_complete = '1'), field ini dianggap telah lengkap

```
IF byte_count = 0 THEN
    current_length_type(15 DOWNTO 8) <= data_in;
    byte_count <= 1;
    length_type_complete <= '0';
ELSIF byte_count = 1 THEN
    current_length_type(7 DOWNTO 0) <= data_in;
    byte_count <= 0;
    length_type_complete <= '1';
```



Payload

Payload adalah data yang akan dikirimkan melalui frame Ethernet. Payload memiliki panjang minimal 46 byte dan maksimal 1500 byte. Pada proyek ini, panjang payload bernilai fixed di 46 bit.

- Bagian ini membaca payload, yaitu data utama yang dikirim dalam frame Ethernet.
- Byte diterima satu per satu dan dimasukkan ke dalam internal_payload.
- Jika jumlah byte mencapai batas (payload_byte_count = 5), sinyal payload_complete dan payload_valid diaktifkan, menandakan payload telah selesai diterima.

```
-- Masukkan data ke payload MSB first
internal_payload <= internal_payload(37 DOWNTO 0) & data_in;

-- Setelah mencapai panjang maksimum payload
IF payload_byte_count = 5 THEN
    payload_complete <= '1';
    payload_valid <= '1';
    payload <= internal_payload;
ELSE
    payload <= internal_payload;
    payload_byte_count <= payload_byte_count + 1;
END IF;
```



CRC

CRC adalah bagian dari frame Ethernet yang terdiri dari 4 byte yang berisi hasil CRC-32 dari frame Ethernet yang dikirimkan. CRC digunakan untuk memastikan bahwa data yang diterima tidak mengalami kesalahan. Algoritma CRC yang digunakan di proyek ini secara eksplisit terpisah dan berada di file crc.vhd. Algoritma ini adalah algoritma khusus untuk frame ethernet dengan panjang payload 46 bit.

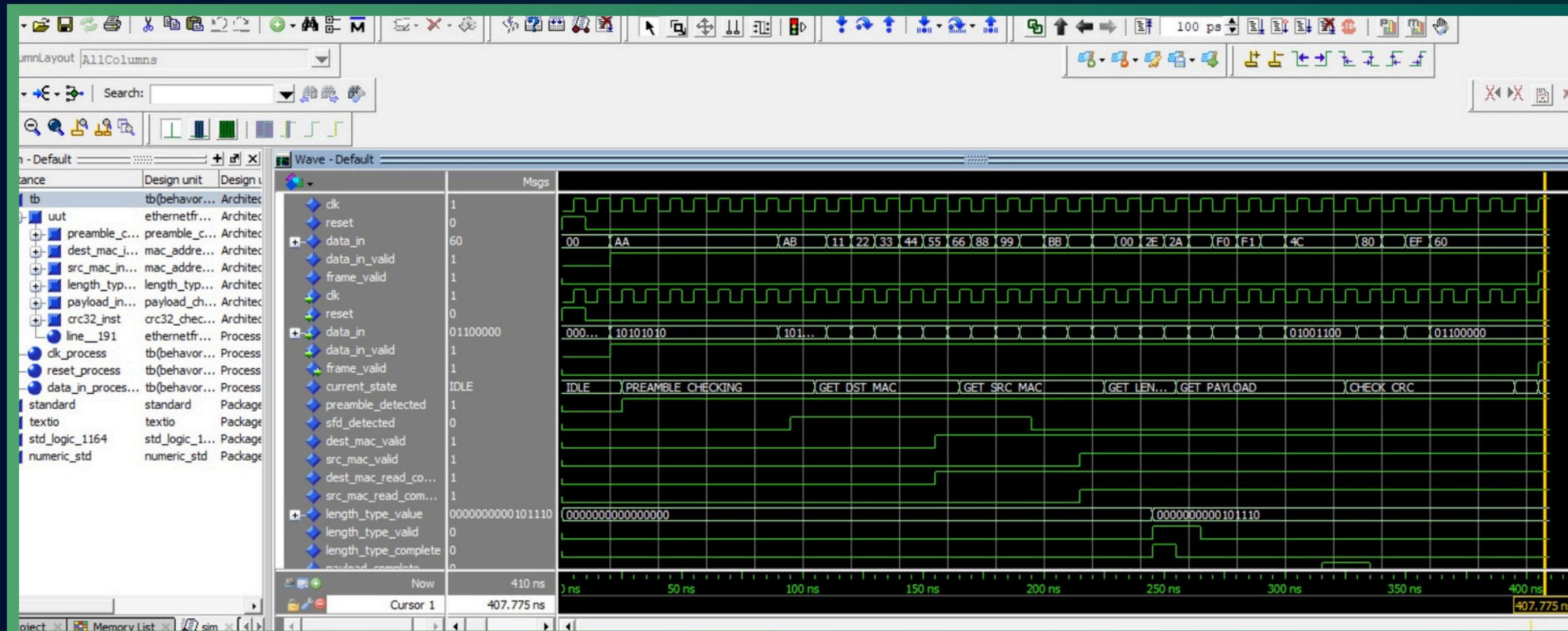
- CRC digunakan untuk memeriksa integritas data frame.
- Pada state CHECK_CRC, nilai CRC yang dihitung (crc_value) dibandingkan dengan nilai CRC yang diterima (input_crc).
 - Jika cocok, sistem pindah ke state VALIDATE_CRC.
 - Jika tidak cocok, sistem kembali ke IDLE, menandakan adanya kesalahan.
- Pada state VALIDATE_CRC, sinyal crc_match dan crc_complete diaktifkan, menandakan CRC valid dan pemeriksaan selesai.

```
WHEN CHECK_CRC =>
    IF input_crc = crc_value THEN
        current_state <= VALIDATE_CRC;
    ELSE
        crc_complete <= '1';
        current_state <= IDLE;
    END IF;

WHEN VALIDATE_CRC =>
    crc_match <= '1';
    crc_complete <= '1';
    current_state <= IDLE;
```



Testing



[Read More](#)



Ethernet Frame parts

● Analisis

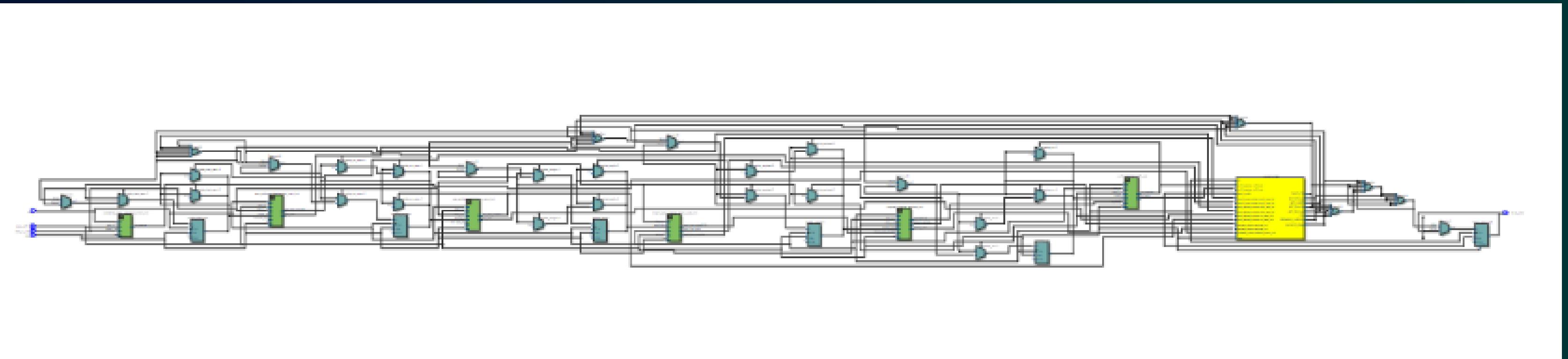
Testing dilakukan dengan menggunakan wave test simulation pada ModelSIM.

Simulation dijalankan pada program test bench dengan input yang sudah di set dari awal dengan contoh input frame yang seharusnya bernilai valid.

Pada pengujian Ethernet Frame Validator, tidak ada anomali atau behavior yang tidak sesuai dari design perancangan awal. Output akhir dari simulasi juga sudah sesuai dengan signal frame_valid bernilai 1 pada akhir simulasi, menandakan bahwa frame yang diuji sudah benar. Jika kita berusaha merubah input frame dan berusaha memberikan frame yang tidak valid, simulasi juga sudah berhasil mengidentifikasinya dengan signal frame_valid tetap bernilai 0 dan state machine kembali ke state IDLE.

[Home](#)[About](#)[Contact](#)

Result



Setelah dilakukan testing dengan wave test dan ternyata sesuai dengan harapan, maka dapat disimpulkan bahwa Ethernet Frame Validator sudah berhasil diimplementasikan pada

FPGA. Langkah selanjutnya adalah melakukan synthesis menggunakan software Quartus Prime. Dapat dilihat pada gambar, bahwa synthesis berhasil dilakukan dan tidak ada error yang muncul. Ini berarti bahwa design tidak hanya sesuai dengan simulasi tetapi juga benar-benar dapat diimplementasikan pada FPGA.

[Home](#)[About](#)[Contact](#)

ANALISIS

Proyek Ethernet Frame Validator ini berhasil dibuat dan diuji untuk memvalidasi frame Ethernet secara serial menggunakan algoritma CRC-32. Sistem yang dirancang mampu menjalankan semua proses validasi dengan baik, mulai dari sinkronisasi frame, pengecekan alamat MAC, panjang frame, payload, hingga perhitungan checksum.

Dari hasil pengujian, sistem terbukti akurat dalam mendeteksi frame yang valid maupun tidak valid. Semua modul bekerja sesuai desain, dan pendekatan modular yang digunakan membuat sistem lebih stabil dan mudah diintegrasikan.

Proyek ini menjadi pengalaman yang baik dalam mengaplikasikan teori yang dipelajari selama perkuliahan. Kami berharap hasil kerja ini bisa berguna sebagai dasar untuk pengembangan sistem jaringan di masa depan dan menjadi referensi untuk proyek serupa.



PARTICIPANT	UNITS SOLD
Audy	11
Bie Oishi	18
Cipio	6
Daniel	14
Gloria	21
Shopkins	25

Home

About

Contact

Thank You For Attention

See You Next