



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

DSP FIR Lowpass Filter

GROUP P-25

Ahmad Fariz Khairi	2306211370
Christover Angelo Lasut	2306220343
Muhammad Raihan Mustofa	2306161946
Ryan Adidaru Excel Barnabi	2306266994

PREFACE

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan proyek akhir ini dengan baik dan tepat waktu. Laporan ini disusun sebagai bagian dari pemenuhan tugas mata kuliah Perancangan Sistem Digital, dengan judul “FIR Lowpass Filter”. Proyek ini bertujuan untuk merancang, mengimplementasikan, dan mensimulasikan sebuah filter digital jenis Finite Impulse Response (FIR) Lowpass Filter. Dengan menggunakan pendekatan berbasis perangkat keras, kami mengembangkan pemahaman tentang pemrosesan sinyal digital sekaligus meningkatkan keterampilan dalam mendesain sistem digital menggunakan bahasa deskripsi perangkat keras VHDL.

Kami menyadari bahwa keberhasilan penyusunan laporan ini tidak terlepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu, kami ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

- Dosen dan Asisten Mata Kuliah Perancangan Sistem Digital, atas bimbingan dan arahannya selama proses pengerjaan proyek ini.
- Rekan Satu Tim, atas kerja sama dan kontribusi aktif dalam setiap tahap proyek, mulai dari desain hingga validasi.
- Keluarga dan Sahabat, atas dukungan moral dan motivasi yang diberikan selama kami menyelesaikan proyek ini.

Kami menyadari bahwa laporan ini masih memiliki kekurangan dan keterbatasan, baik dari segi isi maupun penyajiannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pihak untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca, khususnya dalam bidang pemrosesan sinyal digital dan desain sistem berbasis perangkat keras.

Depok, December 8, 2024

Group P-25

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

- 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

Dalam dunia modern yang semakin berkembang, teknologi pemrosesan sinyal digital atau Digital Signal Processing (DSP) menjadi salah satu fondasi penting dalam berbagai aplikasi, seperti komunikasi, audio, radar, dan sistem kendali. DSP memungkinkan manipulasi sinyal secara efisien untuk meningkatkan kualitas informasi yang disampaikan. Salah satu komponen utama dalam DSP adalah filter digital, yang berfungsi untuk memisahkan atau menghilangkan komponen sinyal yang tidak diinginkan.

Filter digital terbagi menjadi dua kategori utama, yaitu Infinite Impulse Response (IIR) dan Finite Impulse Response (FIR). Filter FIR, yang menjadi fokus dari proyek ini, memiliki sejumlah keunggulan, seperti stabilitas yang terjamin, respons fase linear, dan kemudahan implementasi menggunakan algoritma tertentu. FIR Lowpass Filter digunakan untuk menyaring sinyal frekuensi tinggi, sehingga hanya sinyal frekuensi rendah yang diizinkan melewati filter.

Dalam proyek ini, bahasa deskripsi perangkat keras VHDL (Very High-Speed Integrated Circuit Hardware Description Language) digunakan untuk mendesain FIR Lowpass Filter secara digital. Penggunaan VHDL memungkinkan pengembangan desain yang modular dan fleksibel, sehingga memudahkan proses implementasi pada perangkat keras seperti FPGA (Field Programmable Gate Array).

Penerapan FIR Lowpass Filter ini tidak hanya memberikan pengalaman teknis dalam mendesain dan mensimulasikan filter, tetapi juga memperkenalkan mahasiswa pada teknik pemrograman berbasis perangkat keras. Dengan pendekatan ini, proyek diharapkan dapat menjadi pembelajaran yang bermanfaat dan relevan bagi pengembangan keahlian di bidang pemrosesan sinyal.

1.2 PROJECT DESCRIPTION

Proyek ini bertujuan untuk mengimplementasikan FIR Lowpass Filter berbasis VHDL dengan memanfaatkan koefisien filter yang dihasilkan dari ModelSim. Desain filter ini menggunakan struktur delay line untuk menyimpan data masukan sebelumnya, yang kemudian dikombinasikan dengan koefisien filter melalui operasi perkalian dan penjumlahan.

Fitur utama dari desain ini meliputi:

- Koefisien Filter: Menggunakan array yang dideklarasikan dalam VHDL berdasarkan koefisien ModelSim.
- Delay Line: Menggeser nilai masukan untuk digunakan dalam operasi filter.
- Arsitektur Modular: Menggunakan gaya pemrograman terstruktur untuk mengimplementasikan komponen filter.
- Testbench: Digunakan untuk memvalidasi kinerja filter dengan berbagai sinyal masukan.

Proses implementasi mencakup penulisan kode filter, pembuatan testbench, simulasi menggunakan ModelSim, dan analisis hasil untuk memastikan filter bekerja sesuai spesifikasi.

1.3 OBJECTIVES

Proyek ini memiliki beberapa tujuan utama, yaitu:

1. Merancang FIR Lowpass Filter berbasis VHDL yang dapat digunakan untuk menyaring sinyal digital.
2. Mengembangkan pemahaman tentang desain filter digital, mulai dari teori hingga implementasi praktis.
3. Mengaplikasikan konsep delay line dan operasi aritmatika digital untuk menghasilkan keluaran filter yang presisi.
4. Melakukan validasi desain menggunakan testbench untuk memastikan filter bekerja sesuai dengan spesifikasi yang diberikan.
5. Menyediakan solusi berbasis perangkat keras untuk aplikasi penyaringan sinyal digital dalam berbagai bidang.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Documentation & Program	Membantu membuat laporan, PPT, dan kode structural.	Ahmad Fariz Khairi
Documentation & Program	Membantu membuat laporan dan PPT, FSM	Christover Angelo Lasut
Programmer Source Code VHDL	Membantu membuat sistem FIR Lowpass Filter, penggunaan Matlab, dan penetapan koefisien filter.	Muhammad Raihan Mustofa
Debug, Testing & Simulasi	Membantu membuat FSM, Debug FSM, Sintesis dan Dokumentasi	Ryan Adidaru Excel Barnabi

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

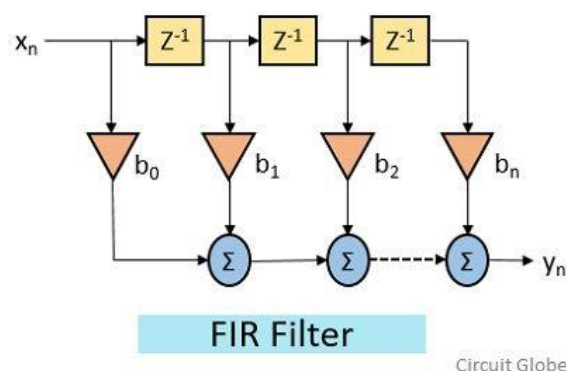
2.1 EQUIPMENT

Terdapat beberapa tools yang digunakan dalam pelaksanaan proyek ini, diantaranya adalah :

- Visual Studio Code
- ModelSim
- Github
- Matlab
- Quartus

2.2 IMPLEMENTATION

FIR (Finite Impulse Response) Lowpass Filter adalah salah satu jenis filter digital yang berfungsi untuk memproses sinyal dengan cara meloloskan komponen frekuensi rendah dan meredam komponen frekuensi tinggi. Filter ini disebut "finite" karena respons impulsnya memiliki durasi terbatas, yang berarti outputnya hanya bergantung pada input saat ini dan sejumlah input sebelumnya tanpa ada feedback dari output. Prinsip kerjanya adalah dengan mengalikan sampel input (saat ini dan sebelumnya) dengan koefisien filter yang telah ditentukan, kemudian menjumlahkan seluruh hasil perkalian tersebut untuk menghasilkan satu sampel output.



$$y[n] = \sum_{k=0}^M b_k x[n - k]$$

Fig 1. Schematic

Dalam implementasinya, FIR filter menggunakan struktur delay line untuk menyimpan sampel-sampel input sebelumnya, dan serangkaian koefisien (tap) yang menentukan karakteristik filter seperti frekuensi cutoff dan ripple. Koefisien-koefisien ini biasanya dirancang menggunakan tool matematika seperti MATLAB dan seringkali memiliki sifat simetris untuk menghasilkan respons fase linear. Semakin banyak jumlah koefisien (tap) yang digunakan, semakin tajam karakteristik filter yang dapat dicapai, namun hal ini juga meningkatkan kebutuhan komputasi dan sumber daya hardware.

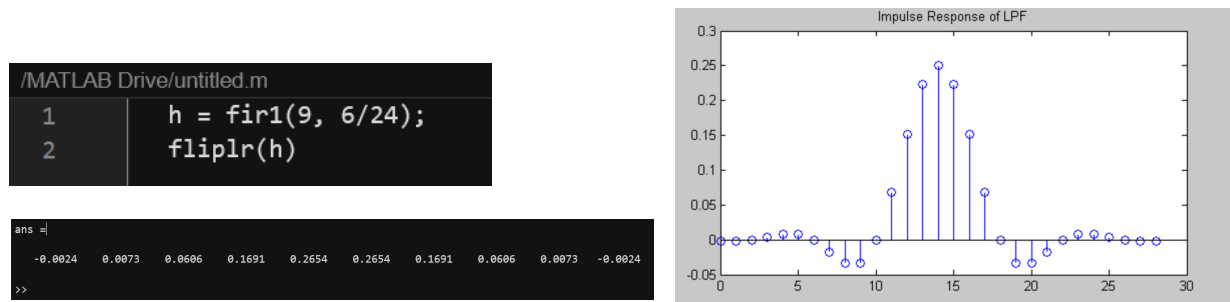


Fig 2. Matlab and Impulse Response

Filter yang dirancang menggunakan arsitektur Direct Form dengan 9 koefisien filter (array 0-8) yang dirancang untuk memfilter sinyal di atas 6 kHz. Koefisien filter didefinisikan sebagai konstanta yang diskalakan dengan 2^{17} untuk meningkatkan presisi perhitungan fixed-point. Nilai koefisien simetris dan berpusat pada 0.2654 (koefisien tengah).

Implementasi menggunakan delay line sepanjang 9 sampel untuk menyimpan sampel input sebelumnya. Pada setiap siklus clock, sampel baru dimasukkan ke delay line[0] dan sampel-sampel sebelumnya digeser. Operasi filtering dilakukan dengan mengalikan setiap sampel dalam delay line dengan koefisien filter yang sesuai, kemudian menjumlahkan seluruh hasil perkalian tersebut menggunakan accumulator.

Proses perhitungan dilakukan secara sekuensial dalam proses yang dipicu oleh clock. Ketika reset aktif, semua register dan delay line dikosongkan. Pada setiap rising edge dari clock, filter melakukan pergeseran sampel dalam delay line, memasukkan sampel input baru, melakukan perkalian sampel dengan koefisien, mengakumulasikan hasil perkalian, dan menyesuaikan ke format output 16-bit dengan shift right. Filter dirancang untuk memproses sinyal input 16-bit dan menghasilkan output 16-bit, dengan perhitungan internal menggunakan lebar bit yang lebih besar (35-36 bit) untuk menghindari overflow.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Strategi pengujian melibatkan pembuatan sinyal yang menggabungkan dua gelombang sinus dengan frekuensi berbeda: gelombang sinus frekuensi rendah 1 kHz dan gelombang sinus frekuensi tinggi 15 kHz. Dengan sengaja memperkenalkan komponen frekuensi yang berbeda, tujuannya adalah untuk memvalidasi kemampuan filter dalam mereduksi sinyal frekuensi tinggi sambil mempertahankan sinyal frekuensi rendah. Sampling ditetapkan pada 100 kHz, memberikan resolusi yang cukup untuk menangkap karakteristik sinyal dan menunjukkan perilaku filter yang selektif terhadap frekuensi.

Testbench mengimplementasikan pendekatan stimulus sistematis, menciptakan sinyal input yang berubah seiring waktu selama 2000 siklus clock, yang memungkinkan pengamatan detail respons filter. Parameter seperti amplitudo dan frekuensi dikendalikan dengan gelombang sinus 1 kHz ditetapkan pada amplitudo 80% dan gelombang sinus 15 kHz pada amplitudo 20%. Dengan melacak secara terpisah sinyal input asli, komponen 1 kHz, dan komponen 15 kHz, testbench menyediakan mekanisme untuk menganalisis kemampuan frekuensi filter, transformasi sinyal, dan efektivitas filtering keseluruhan.

3.2 RESULT

Grafik di bawah menunjukkan bagaimana filter lowpass mampu meredam sinyal frekuensi tinggi (15 kHz) sementara mempertahankan sinyal frekuensi rendah (1 kHz). Hal ini terlihat dari penurunan amplitudo pada sinyal 15 kHz di keluaran filter, sementara sinyal 1 kHz masih dipertahankan. Ini menunjukkan bahwa filter lowpass dapat bekerja sesuai dengan rancangan, yaitu melewatkan frekuensi rendah dan meredam frekuensi tinggi.

Gambar ini menunjukkan sinyal input yang terdiri dari kombinasi dua gelombang sine, yaitu gelombang 1 kHz dengan amplitudo 80% dan gelombang 15 kHz dengan amplitudo 20%. Sinyal ini digunakan untuk menguji kinerja filter lowpass yang dirancang.

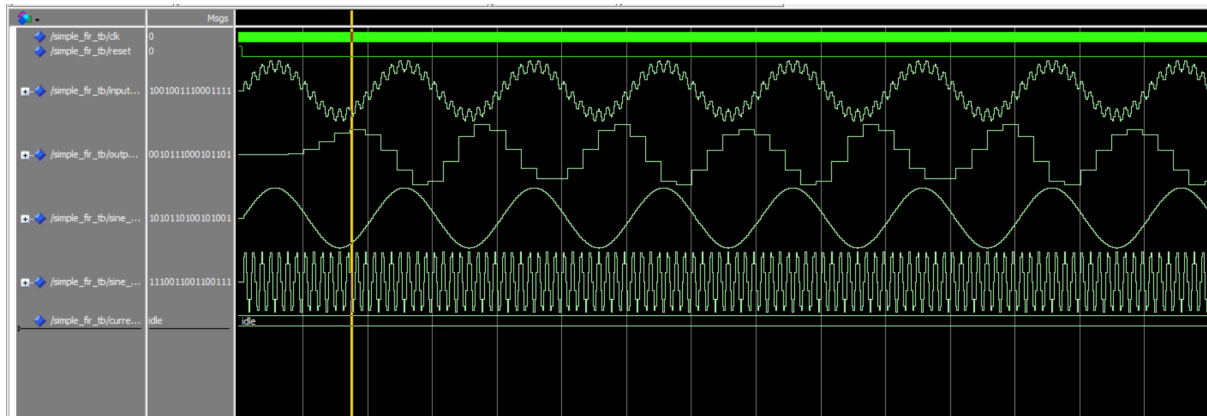


Fig 2. Testing Result

3.3 ANALYSIS

Proyek ini bertujuan untuk mengimplementasikan filter FIR sederhana menggunakan VHDL, di mana pengolahan sinyal dilakukan melalui finite state machine (FSM). Filter dirancang dengan empat state utama: idle, load, compute, dan output_result. Dalam simulasi, ditemukan bahwa FSM selalu berada di state idle, yang menunjukkan adanya kesalahan logika atau implementasi pada desain sistem. Hal ini menjadi fokus utama untuk analisis lebih lanjut guna menemukan penyebab dan memberikan solusi perbaikan.

Pada implementasi FSM, state idle berfungsi sebagai titik awal sebelum sistem melanjutkan proses ke state berikutnya. Namun, selama simulasi, FSM tidak bertransisi ke state load seperti yang diharapkan. Salah satu penyebab potensial adalah sinyal reset yang tidak dilepas dengan benar. Jika sinyal reset tetap aktif, sistem akan terus kembali ke state idle pada setiap siklus clock. Selain itu, mekanisme transisi dalam case current_state is harus diperiksa lebih lanjut. Kemungkinan besar, kondisi yang mengontrol transisi dari idle ke load tidak terpenuhi atau tidak diimplementasikan secara benar.

Filter ini menggunakan struktur delay line dan operasi multiply-accumulate (MAC) untuk mengolah sinyal input. Struktur delay line bertugas menyimpan sampel sebelumnya yang digunakan dalam perhitungan MAC. Simulasi mengindikasikan bahwa operasi MAC mungkin tidak berjalan dengan benar, terutama pada variabel mac_count yang digunakan

untuk mengontrol iterasi pengolahan koefisien. Jika variabel ini tidak diinisialisasi atau diperbarui dengan benar, maka akumulasi hasil perhitungan tidak akan berjalan sesuai ekspektasi, sehingga FSM tidak dapat melanjutkan ke state compute atau output_result.

Analisis testbench untuk implementasi FIR Lowpass Filter dimulai dengan state_mapping_pkg, yang memiliki fungsi krusial dalam menerjemahkan state enumerasi menjadi string yang dapat dibaca. Melalui fungsi get_state_name yang diimplementasikan dengan struktur case, paket ini memberikan representasi yang jelas tentang state aktif dalam Finite State Machine (FSM), dengan penambahan when others untuk meningkatkan keamanan dan penanganan input yang tidak valid. Parameter simulasi, yang mencakup konstanta SAMPLE_RATE (100 kHz) dan CLK_PERIOD (10 ns), dirancang untuk mereplikasi kondisi operasional sistem digital nyata, memberikan kerangka waktu yang akurat dalam pengujian filter digital.

Proses stimulus input menjadi elemen kunci dalam testbench, menghasilkan sinyal kompleks yang menggabungkan dua gelombang sinusoidal: frekuensi rendah 1 kHz dengan amplitudo 80% dan frekuensi tinggi 15 kHz dengan amplitudo 20%. Variabel low_freq_amp dan high_freq_amp digunakan untuk menentukan karakteristik amplitudo, sementara sinyal input_signal, sine_1kHz, dan sine_15kHz memungkinkan analisis komprehensif respons filter. Proses pemantauan FSM (monitor) memainkan peran vital dalam observasi, menampilkan state saat ini dan nilai sinyal output pada setiap siklus clock melalui konversi state ke string dan transformasi sinyal output ke bilangan integer. Device Under Test (DUT) yang diimplementasikan sebagai entitas simple_fir mengintegrasikan semua elemen ini, menerima input kompleks dan menghasilkan output yang memungkinkan evaluasi kinerja filter FIR.

CHAPTER 4

CONCLUSION

Proyek implementasi FIR Lowpass Filter merupakan upaya dalam merancang filter digital untuk pemrosesan sinyal menggunakan arsitektur Direct Form dengan sembilan koefisien. Filter dirancang khusus untuk memfilter sinyal di atas 6 kHz, mengimplementasikan fixed-point dengan penskalaan koefisien menggunakan 2^{17} untuk meningkatkan presisi perhitungan. Arsitektur filter menggunakan delay line sepanjang sembilan sampel yang memungkinkan penyimpanan dan pengolahan sampel input sebelumnya, dengan kemampuan memproses sinyal 16-bit melalui perhitungan internal 35-36 bit untuk mencegah potensi overflow. Strategi pengujian dilakukan melalui sinyal kompleks yang menggabungkan dua gelombang sine dengan frekuensi berbeda: gelombang sine frekuensi rendah 1 kHz dan gelombang sine frekuensi tinggi 15 kHz, dengan laju sampling 100 kHz untuk memvalidasi kemampuan filter dalam mereduksi sinyal frekuensi tinggi sambil mempertahankan konten frekuensi rendah.

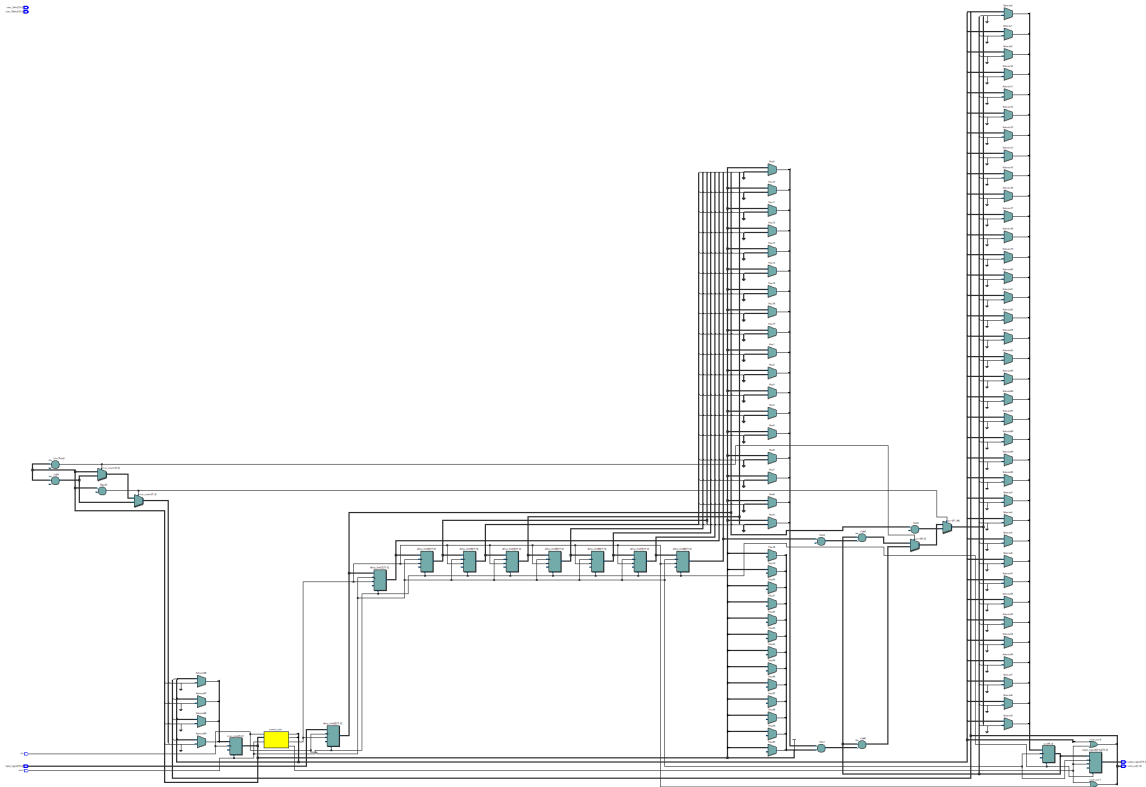
Selama proses implementasi dan simulasi, ditemukan beberapa tantangan signifikan dalam desain Finite State Machine (FSM), di mana sistem cenderung berada pada state idle, yang mengindikasikan potensi kesalahan dalam logika transisi state, mekanisme reset, atau operasi Multiply-Accumulate (MAC). Meskipun mengalami kendala teknis, filter lowpass berhasil menunjukkan kemampuan dasarnya dalam meredam sinyal frekuensi tinggi sambil mempertahankan integritas sinyal frekuensi rendah. Hasil pengujian menggambarkan kemampuan frekuensi filter, dengan penurunan signifikan pada amplitudo sinyal 15 kHz di output filter, sementara sinyal 1 kHz dapat dipertahankan dengan baik. Kesimpulan dari proyek ini menunjukkan bahwa implementasi FIR Lowpass Filter menggunakan VHDL memiliki potensi yang baik, namun memerlukan optimasi lebih lanjut, dengan fokus utama perbaikan mencakup validasi logika fsm, penyempurnaan mekanisme reset, dan pengoptimalan operasi pengolahan sinyal.

REFERENCES

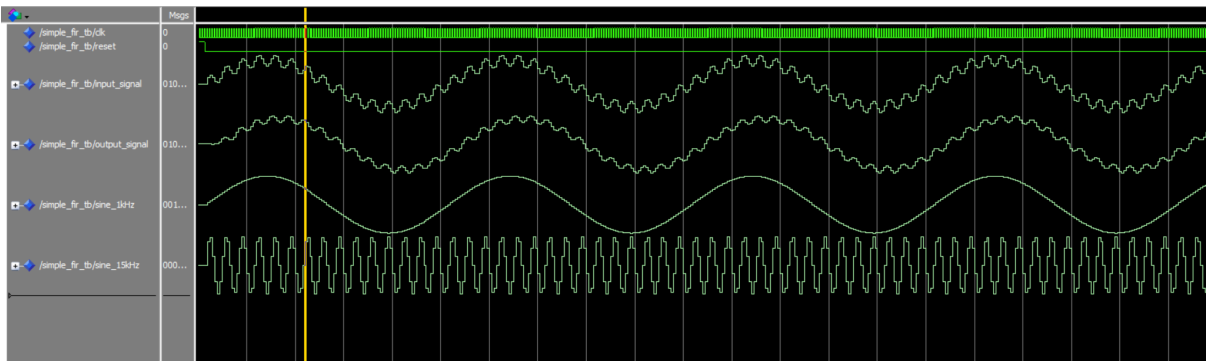
- [1] Circuit Globe, "Difference Between FIR Filter and IIR Filter," *Circuit Globe*, Mar. 24, 2020. <https://circuitglobe.com/difference-between-fir-filter-and-iir-filter.html> (accessed Dec. 08, 2024).
- [2] "FIR Lowpass Filter Example," *Github.io*, 2024. https://arm-software.github.io/CMSIS_5/DSP/html/group_FIRLPF.html (accessed Dec. 08, 2024).
- [3] "Memisahkan Sinyal Yang Tercampur dengan FIR Filter," *Mon Honneur Est Nommée Fidèle*, Feb. 27, 2012. <https://stahlvormund.wordpress.com/2012/02/27/memisahkan-sinyal-yang-tercampur-dengan-fir-filter/> (accessed Dec. 08, 2024).
- [4] "FIR Low-Pass Filter Component," *Qsc.com*, 2024. https://q-syshelp.qsc.com/Content/Schematic_Library/filter_lowpass_fir.htm?TocPath=%7CSystem%20Link%7CUSB%7CUSB%20Input%7CFIR%20Low-pass%20Filter%7C___0 (accessed Dec. 08, 2024).
- [5] Surf-VHDL, "How to Implement FIR Filter in VHDL," *Surf-VHDL*, Nov. 14, 2015. <https://surf-vhdl.com/how-to-implement-fir-filter-in-vhdl/> (accessed Dec. 08, 2024).
- [6] D. Marinov, "Part 2: Finite impulse response (FIR) filters," *VHDLwhiz*, Feb. 10, 2022. <https://vhdlwhiz.com/part-2-finite-impulse-response-fir-filters/> (accessed Dec. 08, 2024).

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation



Raihan Mustofa



 Ahmad Fariz Khairi

✍ Christover Angelo