



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

Morse Code to ASCII Decoder

KELOMPOK 4

Arkaan Pasya Seplitara	2406408073
Danish Al Fayyadh Sunarta	2406416951
Raihan Muhammad Nafis Al-Kautsar	2406413451
Qais Ismail	2406487090

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, kami dapat menyelesaikan Laporan Tugas Akhir mata kuliah Perancangan Sistem Digital ini yang berjudul "Morse Code to ASCII Decoder".

Laporan ini disusun sebagai dokumentasi dari proses perancangan, implementasi, dan pengujian sistem digital yang mampu menerjemahkan kode Morse menjadi karakter ASCII menggunakan bahasa deskripsi perangkat keras VHDL. Proyek ini menggabungkan konsep Finite State Machine (FSM), Shift Register, dan logika digital untuk menerjemahkan input morse code dalam biner menjadi output dalam karakter ASCII.

Kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan demi perbaikan di masa mendatang.

Depok, December 7, 2025

Kelompok 4

DAFTAR ISI

BAB 1 : PENDAHULUAN

- 1.1 LATAR BELAKANG
- 1.2 DESKRIPSI PROYEK
- 1.3 TUJUAN
- 1.4 PERAN DAN TANGGUNG JAWAB

BAB 2 : IMPLEMENTASI

- 2.1 ALAT
- 2.2 IMPLEMENTASI

BAB 3: TESTING DAN ANALISIS

- 3.1 TESTING
- 3.2 HASIL
- 3.3 ANALISIS

BAB 4 : KESIMPULAN

REFERENSI

APPENDICES

Appendix A: Project Schematic

Appendix B: Dokumentasi

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Morse code merupakan salah satu metode telekomunikasi tua dan bertahan cukup lama, yang meng-encode karakter teks menjadi kode standar yang terdiri dari dua sinyal yang memiliki durasi berbeda, yaitu dots dan dashes. Meskipun awalnya dirancang untuk operator manusia yang menerjemahkan sinyal ini berdasarkan pola dan pendengaran, proses pengotomasi decoding ke dalam sistem digital memiliki tantangan tersendiri. Berbeda dengan manusia, circuit digital tidak dapat mengetahui irama; mereka membutuhkan waktu yang dapat dikuantifikasi dan juga presisi untuk membedakan berbagai komponen pesan tersebut.

Kesulitan utama dalam merancang digital Morse decoder terletak pada signal timing dan synchronization. Arus data biner yang berkelanjutan 0 dan 1 masuk ke dalam sistem, di mana logika '1' mewakili naiknya sinyal dan '0' mewakili tidak ada sinyal. Sistem harus mengukur durasi setiap pulse tinggi untuk memutuskan apakah itu sebuah dot atau dash, serta mengukur setiap gap rendah untuk menentukan apakah itu pemisah antar bagian huruf, satu huruf utuh, atau kata baru yang lengkap. Tanpa mekanisme yang kuat untuk menghitung durasi ini terhadap threshold tertentu, sistem tidak akan dapat mem-parse aliran data yang masuk menjadi informasi yang benar.

Untuk mengatasi hal ini, entity Morse Decoder yang dikembangkan dalam proyek ini beroperasi sebagai synchronous Finite State Machine (FSM). Sistem ini memanfaatkan clock internal yang bertindak sebagai stopwatch, menghitung siklus untuk mengkategorikan pulse yang masuk. Dengan melakukan shifting pada dots dan dashes yang terdeteksi ke dalam sebuah buffer, mesin ini menyusun digital signature unik untuk setiap urutan yang diterima. Hal ini memungkinkan sistem untuk menjembatani kesenjangan antara sinyal serial berbasis waktu dan teks ASCII parallel yang dapat dibaca, sehingga proses penerjemahan dapat diotomatisasi dengan kecepatan dan akurasi tinggi.

1.2 DESKRIPSI PROYEK

Proyek "Morse to ASCII Decoder" ini dirancang untuk menerima input biner ('1' dan '0') dengan cara kerja sebagai berikut:

1. Mengidentifikasi input morse: mengukur durasi sinyal input '1' untuk membedakan antara simbol morse dot atau dash
2. Mengidentifikasi jeda: mengukur durasi sinyal input '0' lalu mengidentifikasinya sebagai jeda antar simbol morse, huruf, atau kata.
3. Menyimpan input: menyimpan urutan dot dan dash ke dalam sebuah memori buffer
4. Decoding: menerjemahkan kode morse dengan mencocokkan pola dot dan dash yang tersimpan pada memori buffer dengan tabel morse to ASCII, lalu men-generate file .txt yang berisi output ASCII yang dapat dibaca.

1.3 TUJUAN

Tujuan dari proyek ini antara lain:

1. Merancang dan mengimplementasikan sistem digital sinkron yang mampu menerjemahkan sinyal kode Morse serial menjadi karakter ASCII paralel menggunakan VHDL.
2. Mengimplementasikan Microprogrammed Control Unit untuk mengelola transisi state dan sinyal kendali secara efisien berdasarkan durasi pulsa input.
3. Memanfaatkan Shift Register dan Counter pada Datapath untuk mengukur waktu sinyal (membedakan dot dan dash) serta menghasilkan signature biner unik untuk proses decoding karakter.
4. Memvalidasi fungsionalitas sistem melalui simulasi menggunakan testbench yang meniru urutan Morse dunia nyata dan menghasilkan file teks yang dapat dibaca (output.txt).

1.4 PERAN DAN TANGGUNG JAWAB

Peran dan tanggung jawab yang diberikan kepada anggota kelompok adalah sebagai berikut:

Peran	Tanggung Jawab	Anggota
Ketua	Membuat backbone, membuat logika fundamental dari decoder, dan membuat testbench	Arkaan Pasya Seplitara
Anggota	Membuat control unit (microprogramming), membuat fungsi output.txt	Qais Ismail
Anggota	Membuat tabel morse to ASCII Membuat README.md	Danish Al Fayyadh Sunarta
Anggota	Membuat konsep awal control unit (microprogramming)	Raihan Muhammad Nafis Al-Kautsar

Table 1. Roles and Responsibilities

BAB 2

IMPLEMENTASI

2.1 ALAT

Alat-alat yang digunakan dalam proyek ini adalah sebagai berikut:

- Xilinx Vivado / Intel Quartus Prime
- ModelSim / Vivado Simulator
- VS Code / Notepad

2.2 IMPLEMENTASI

Implementasi sistem dibagi menjadi dua komponen utama: Datapath (`morse_decoder.vhd`) dan Control Unit (`control_unit.vhd`) dengan arsitektur struktural.

1. **Microprogrammed Control Unit:** Berbeda dengan Finite State Machine konvensional yang hardwired, Control Unit ini dirancang sebagai Microprogrammed FSM. Sistem ini menggunakan ROM internal (`Control_Store`) yang berisi 16 baris *microcode*. Setiap instruksi mikro terdiri dari:
 - Sequence Bits (3 bit): Menentukan logika next state (contoh: `SEQ_JMP_HIGH`, `SEQ_CHK_FALL`).
 - Control Bits (7 bit): Mengontrol sinyal Datapath secara langsung (seperti `ctrl_inc_one` untuk increment counter, `ctrl_clr_all` untuk reset).

Control Unit memantau sinyal `morse_in` dan flags dari Datapath (`flag_char_end`, `flag_space_end`) untuk melakukan transisi antara kondisi IDLE, HIGH (pemrosesan pulsa), dan LOW (pemrosesan jeda).

2. **Datapath dan Logika Decoding:** Datapath menangani pengukuran waktu dan penyimpanan data:
 - Timing Counters: Dua integer counter digunakan untuk mengukur durasi sinyal. `counter_one` menghitung durasi sinyal tinggi (mark), dan `counter_zero` menghitung durasi sinyal rendah (space).

- Komparasi Threshold: Sistem menggunakan konstanta untuk membedakan simbol:
 - THRESH_DASH (2 siklus): Membedakan Dot (<2) dan Dash (≥ 2).
 - THRESH_CHAR_END (3 siklus): Mendeteksi akhir dari sebuah huruf.
 - THRESH_SPACE (7 siklus): Mendeteksi spasi antar kata.
- Input Buffer: Sebuah shift register 10-bit (shiftreg_inb) menyusun pola unik untuk setiap karakter Morse. Saat dot terdeteksi, pola bit tertentu digeser masuk; saat dash terdeteksi, pola bit yang berbeda digeser.
- Tabel ASCII: Proses kombinasional menerjemahkan nilai integer dari shift register menjadi vektor ASCII 8-bit (contoh: pola register bernilai 512 dipetakan menjadi huruf 'E', 768 menjadi 'T').

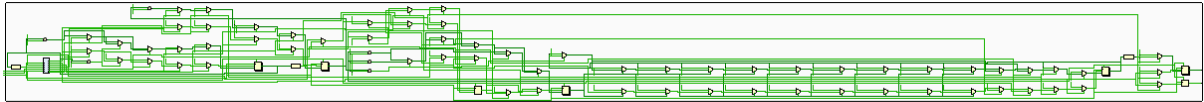


Fig 1. Schematic

BAB 3

TESTING DAN ANALISIS

3.1 TESTING

Pengujian dilakukan menggunakan *testbench* (testbench.vhd) untuk memverifikasi fungsionalitas keseluruhan sistem decoder.

- **Skenario Input:** Testbench dirancang untuk mengirimkan serangkaian sinyal morse yang merepresentasikan kalimat: *"LIST OF THE THINGS THAT I HATE THE MOST"*.
- **Prosedur Stimulus:** Dibuat prosedur `send_dot`, `send_dash`, dan `send_space` yang secara otomatis mengatur durasi sinyal high dan low sesuai dengan periode clock agar sesuai dengan *threshold* yang ditetapkan pada desain utama.
- **Observasi:** Pengujian dilakukan dengan mengamati *waveform* pada simulator (Vivado/ModelSim) dan memeriksa file output yang dihasilkan.

3.2 HASIL

Berdasarkan simulasi yang dijalankan, sistem berhasil mendeteksi urutan sinyal input dan menerjemahkannya.

1. **Waveform:** Sinyal `valid_out` bernilai '1' setiap kali akhir karakter terdeteksi, dan `ascii_out` menunjukkan nilai Hex yang sesuai dengan huruf yang dikirim.
2. **File Output (output.txt):** File teks berhasil digenerate oleh sistem dengan isi sebagai berikut:

ambang batas. Logika ini memastikan ketahanan terhadap variasi waktu kecil (jitter) selama pulse width tetap berada di atas ambang batas integer yang ditentukan.

2. **Efisiensi FSM Mikroprogram:** Implementasi Microprogrammed terbukti sangat terstruktur. Dengan memisahkan logika transisi state (Sequence bits) dari logika output (Control bits) di dalam ROM, logika Control Unit menjadi sederhana dan mudah dimodifikasi. Jika persyaratan waktu berubah, hanya microcode dalam fungsi `init_rom` yang perlu diperbarui, tanpa perlu menulis ulang keseluruhan arsitektur sistem.
3. **Logika Decoding:** Pendekatan shift-register menciptakan logika hashing, dengan mengubah input berupa bit menjadi integer, yang unik untuk urutan Morse. Misalnya, huruf 'A' (.-) menghasilkan nilai integer spesifik dalam `shiftreg_inb` setelah logika padding diterapkan. Hal ini memungkinkan decoding instan menggunakan pernyataan CASE, yang akan disintesis menjadi logika multiplexer yang efisien atau Look-Up Table (LUT) pada FPGA.
4. **Keterbatasan:** Sistem saat ini tidak bersifat adaptif (*auto-baud rate*). Kecepatan ketukan morse (WPM) diasumsikan konstan dan diketahui sebelumnya (`CLK_PERIOD` yang tetap). Pada aplikasi nyata dengan operator manusia, fitur toleransi waktu yang lebih fleksibel mungkin diperlukan. Selain itu program ini juga tidak mampu menerjemahkan input morse yang berupa angka, hal ini dikarenakan keterbatasan ukuran `shiftreg_inb` yang hanya 10 bit, diperlukan setidaknya 12 bit untuk menyimpan morse angka dalam biner.

BAB 4

KESIMPULAN

Berdasarkan perancangan dan pengujian yang telah dilakukan, dapat disimpulkan bahwa:

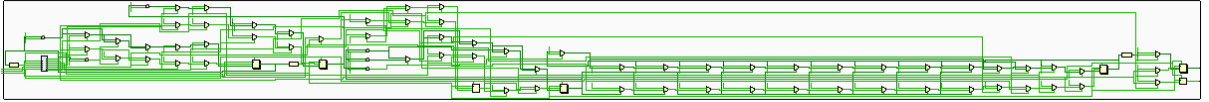
1. Sistem *Morse Code to ASCII Decoder* berhasil diimplementasikan menggunakan VHDL. Mekanisme *Finite State Machine* (FSM) yang dipadukan dengan microprogrammed control unit yang mengatur transisi state pada FSM terbukti efektif dalam memproses aliran data serial biner menjadi data paralel ASCII yang valid.
2. Penggunaan logika pengukuran durasi sinyal (counter) memungkinkan sistem membedakan antara *dot*, *dash*, dan berbagai jenis jeda (*gap*) secara presisi sesuai spesifikasi waktu yang ditetapkan.
3. Fitur penyimpanan output ke dalam file .txt berhasil diaplikasikan, memudahkan verifikasi data yang panjang seperti kalimat penuh, dibandingkan hanya melihat waveform per karakter.
4. Proyek ini menunjukkan bahwa konsep dasar sistem digital seperti Shift Register, Counter, dan FSM dapat digabungkan untuk menyelesaikan masalah telekomunikasi klasik secara digital dan otomatis.

REFERENSI

- [1] IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-2008.
- [2] Volnei A. Pedroni, *Circuit Design with VHDL*, 3rd Edition, MIT Press, 2020.
- [3] International Telecommunication Union, "Morse code for use in the maritime mobile satellite service," ITU-R M.1677-1, 2009.
- [4] J. Bhasker, *A VHDL Primer*, 3rd Edition, Prentice Hall, 1998.
- [5] Xilinx, "Vivado Design Suite User Guide: Logic Simulation," UG900, 2021

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation