

**JURNAL PRAKTIKUM  
RANGKAIAN LOGIKA  
DIGITAL MODUL I  
IMPLEMENTASI PROSESSOR SEDERHANA I**



**Nama : Fresinendi**  
**No. BP : 2211512019**  
**Hari/Tanggal : Jumat/ 25 Oktober 2024**  
**Shift III**  
**Asisten : 1. Mutiara Hikmah**  
**2. Lola Dwi Putri**

**LABORATORIUM SISTEM DIGITAL DAN ARSITEKTUR  
KOMPUTER DEPARTEMEN TEKNIK KOMPUTER  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
PADANG  
2024**

## 1. Percobaan 1 : Bahasa Assembly dan Bahasa Mesin

$$x = (3 + 4 + 2) - (6 + 1)$$

addi R0,3. R0 = 3

addi R0,4. R0 = R0 + 4 (3 + 4) addi

R0,2. R0 = R0 + 2 (3 + 4 + 2)

addi R1,6. R1 = 6

addi R1,1. R1 = R1 + 1 (6 + 1)

sub R0,R1. R0 = R0(3 + 4 + 2) – R1(6 + 1)

Bahasa Assembly :

100000011

100000100

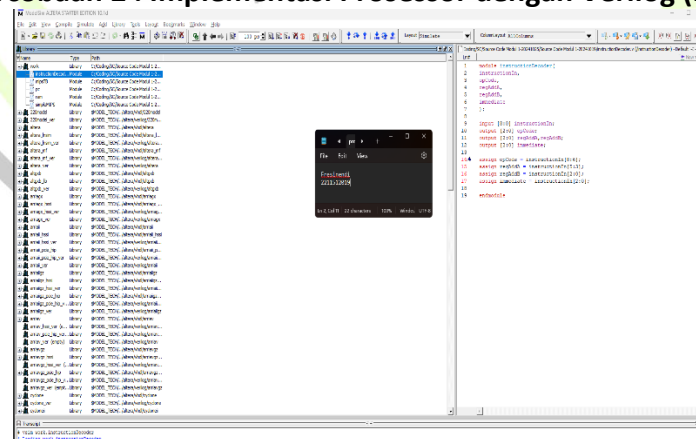
100000010

100001110

100001001

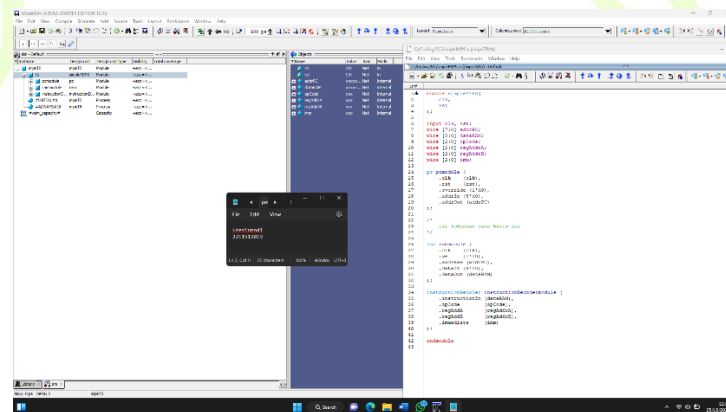
001000001

## 2. Percobaan 2 : Implementasi Prosessor dengan Verilog (IF dan ID)

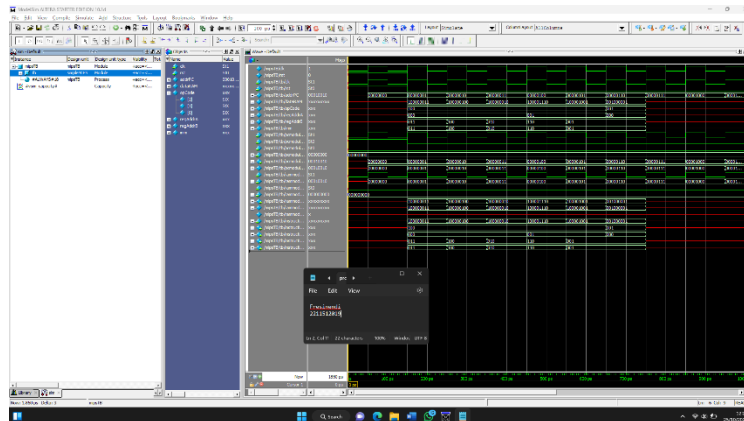


Gambar 2.1 Memasukkan Source Code untuk di Compile

1. Modul pc.v bertugas menyimpan dan menghasilkan alamat instruksi yang akan dijalankan CPU. Sinyal pentingnya meliputi clock (clk) untuk memperbarui alamat setiap kali ada transisi naik, reset (rst) untuk mengembalikan alamat ke 0, dan override untuk menimpa alamat.
2. Modul RAM.v adalah memori blok yang menyimpan data dan mendukung operasi baca/tulis berdasarkan alamat. Dengan sinyal clock (clk), alamat (address), data input (dataIn), dan Write Enable (we), modul ini mengatur proses baca-tulis memori.
3. Modul instructionDecoder.v berfungsi mengurai instruksi biner pada CPU menjadi bagian-bagian seperti opcode, alamat register, dan nilai immediate.



Gambar 2.2 Menambahkan kode pada simpleMIPS.v



Gambar 2.3 Wave Sinyal

Pertanyaan :

1. Jelaskan proses instruction fetch dan instruction decoding secara rinci Instruction Fetch dan Instruction Decoding

Pada tahap pertama siklus eksekusi, yaitu Instruction Fetch, prosesor mengambil instruksi dari memori utama (RAM) dan menyimpannya dalam register. Lokasi instruksi berikutnya ditentukan oleh Program Counter (PC), yang kemudian diperbarui setelah instruksi diambil. Setelah itu, instruksi ditempatkan di Instruction Register (IR) untuk proses selanjutnya. Pada tahap Instruction Decoding, instruksi di IR diuraikan menjadi opcode dan operand. Control Unit (CU) membaca opcode untuk menentukan operasi yang diperlukan, seperti perhitungan, transfer data, atau percabangan. Jika instruksi menggunakan immediate value, nilai tersebut diekstrak dan disiapkan di register tertentu untuk tahap eksekusi.

2. Jumlah Immediate dan Cara Meningkatkan

Jumlah immediate yang didukung bergantung pada panjang field immediate dalam arsitektur instruksi prosesor. Misalnya, jika ada field immediate 16-bit, rentang nilai yang didukung adalah:

Unsigned: 0 hingga 65,535

Signed (2's complement): -32,768 hingga 32,767

Untuk mendukung lebih banyak nilai immediate, beberapa pendekatan dapat digunakan:

- Memperlebar Format Instruksi: Menambah panjang bit instruksi untuk lebih banyak nilai immediate.
- Immediate Extension: Jika field immediate kecil, nilai dapat dipecah dalam beberapa instruksi atau dengan menggunakan register tambahan.
- Literal Pool: Menyimpan nilai immediate besar di memori, lalu menggunakan pengalamatan tidak langsung untuk mengaksesnya.

3. Tugas Immediate Extend

Immediate extend memperbesar nilai immediate dalam instruksi agar cocok dengan operasi selanjutnya. Tugas-tugasnya meliputi:

1. Melakukan sign extension atau zero extension pada nilai immediate berdasarkan tipe signed atau unsigned.
2. Memperluas nilai jika immediate bitnya lebih kecil dari register prosesor (misalnya, immediate 16-bit untuk register 32-bit).
3. Memastikan nilai immediate dapat digunakan di berbagai operasi tanpa masalah ukuran data, seperti operasi aritmatika atau pengalamatan memori.

