

Nama : Daffa Althaf Ridovani

NIM : 03041282227097

Kelas : TKK 2022 / Indralaya

## QUIZ I MIKROPROSESOR

### 1. Sebutkan bagian-bagian yang terdapat pada mikroprosesor

Mikroprosesor adalah komponen utama dalam komputer yang berfungsi sebagai otak untuk menjalankan instruksi dan mengontrol operasi sistem. Mikroprosesor terdiri dari beberapa bagian penting yang bekerja bersama untuk memproses data dan menjalankan program. Berikut adalah bagian-bagian utama mikroprosesor beserta penjelasannya:

#### 1. ALU (Arithmetic Logic Unit)

ALU bertanggung jawab untuk melakukan operasi aritmetika (seperti penjumlahan, pengurangan, perkalian, dan pembagian) dan operasi logika (seperti AND, OR, NOT, XOR).

**Cara Kerja:** ALU mengambil data dari register, melakukan operasi yang diperlukan, dan mengembalikan hasilnya ke register atau memori.

#### 2. CU (Control Unit)

CU mengontrol aliran data dan instruksi di dalam mikroprosesor. Ini mengatur bagaimana instruksi diambil dari memori, di-decode, dan dijalankan.

**Cara Kerja:** CU menghasilkan sinyal kontrol yang mengatur operasi ALU, register, dan komponen lainnya.

### 3. Register

*Fungsi:* Register adalah memori kecil yang sangat cepat yang digunakan untuk menyimpan data sementara dan alamat memori yang sedang diproses oleh mikroprosesor.

*Jenis Register:*

- **Accumulator:** Menyimpan hasil operasi ALU.
- **Program Counter (PC):** Menyimpan alamat instruksi berikutnya yang akan dijalankan.
- **Instruction Register (IR):** Menyimpan instruksi yang sedang diproses.
- **Memory Address Register (MAR):** Menyimpan alamat memori yang akan diakses.
- **Memory Data Register (MDR):** Menyimpan data yang akan ditulis atau dibaca dari memori.

### 4. Cache Memory

Cache memory adalah memori kecil yang sangat cepat yang digunakan untuk menyimpan salinan data yang sering diakses dari memori utama. Ini membantu mempercepat akses data dan mengurangi latensi.

*Jenis Cache:*

- **L1 Cache:** Cache level pertama, terintegrasi di dalam mikroprosesor.
- **L2 Cache:** Cache level kedua, biasanya lebih besar dari L1 tetapi sedikit lebih lambat.
- **L3 Cache:** Cache level ketiga, digunakan dalam beberapa mikroprosesor untuk meningkatkan kinerja.

## 5. Bus

Bus adalah jalur komunikasi yang menghubungkan berbagai komponen mikroprosesor dengan komponen lain di sistem komputer.

**Jenis Bus:**

- **Data Bus:** Mentransfer data antara mikroprosesor dan memori atau perangkat I/O.
- **Address Bus:** Mentransfer alamat memori atau perangkat I/O yang akan diakses.
- **Control Bus:** Mentransfer sinyal kontrol yang mengatur operasi sistem.

## 6. Clock

Clock menghasilkan sinyal waktu yang digunakan untuk menyinkronkan operasi mikroprosesor. Setiap operasi dilakukan dalam siklus clock.

**Cara Kerja:** Frekuensi clock menentukan kecepatan mikroprosesor, diukur dalam Hertz (Hz). Semakin tinggi frekuensi, semakin cepat mikroprosesor dapat menjalankan instruksi.

## 7. Instruction Decoder

Instruction decoder mengubah instruksi biner yang diambil dari memori menjadi sinyal kontrol yang dapat dimengerti oleh ALU dan komponen lainnya.

**Cara Kerja:** Setelah instruksi diambil dari memori, decoder akan memecahnya menjadi operasi yang lebih kecil yang dapat dijalankan oleh mikroprosesor.

## 8. Floating Point Unit (FPU)

**Fungsi:** FPU adalah unit khusus yang menangani operasi aritmetika bilangan floating-point (bilangan desimal). Ini sangat penting untuk

aplikasi yang memerlukan presisi tinggi, seperti grafik 3D dan simulasi ilmiah.

**Cara Kerja:** FPU bekerja secara paralel dengan ALU untuk mempercepat pemrosesan operasi bilangan floating-point.

#### 9. Pipeline

Pipeline adalah teknik yang digunakan untuk meningkatkan efisiensi mikroprosesor dengan memungkinkan beberapa instruksi diproses secara bersamaan dalam tahapan yang berbeda.

**Cara Kerja:** Setiap tahap dalam pipeline menangani bagian tertentu dari instruksi, sehingga mikroprosesor dapat menjalankan lebih banyak instruksi dalam waktu yang lebih singkat.

#### 10. Multicore

**Fungsi:** Mikroprosesor modern sering memiliki beberapa core (inti) yang memungkinkan eksekusi paralel dari beberapa instruksi secara bersamaan.

**Cara Kerja:** Setiap core memiliki ALU, CU, dan register sendiri, sehingga dapat menjalankan program secara independen. Ini meningkatkan kinerja secara signifikan, terutama untuk aplikasi multithreaded.

## 2. Bagaimana tahapan cara kerja mikroprosesor

Mikroprosesor bekerja melalui serangkaian tahapan yang dikenal sebagai siklus instruksi (instruction cycle). Setiap instruksi yang diberikan kepada mikroprosesor diproses melalui tahapan-tahapan ini secara berurutan. Berikut adalah tahapan cara kerja mikroprosesor:

### 1. Fetch (Pengambilan Instruksi)

*Tujuan:* Mengambil instruksi dari memori.

*Proses:*

- *Program Counter (PC)* menyimpan alamat memori dari instruksi berikutnya yang akan dijalankan.
  - Mikroprosesor mengirim alamat ini ke memori melalui *Address Bus*.
  - Instruksi yang sesuai diambil dari memori dan dikirim ke mikroprosesor melalui *Data Bus*.
  - Instruksi tersebut disimpan di *Instruction Register (IR)*.
  - *Program Counter (PC)* kemudian diperbarui untuk menunjuk ke alamat instruksi berikutnya.
- 

### 2. Decode (Dekode Instruksi)

*Tujuan:* Menerjemahkan instruksi menjadi sinyal kontrol yang dapat dimengerti oleh mikroprosesor.

*Proses:*

- Instruksi yang disimpan di *Instruction Register (IR)* diterjemahkan oleh *Instruction Decoder*.
  - Instruksi dipecah menjadi operasi yang lebih kecil (*opcode*) dan data yang diperlukan (*operand*).
  - Sinyal kontrol dihasilkan untuk mengatur komponen lain seperti *ALU*, register, dan bus.
- 

### 3. Execute (Eksekusi Instruksi)

*Tujuan: Menjalankan instruksi yang telah didekode.*

*Proses:*

- *Control Unit (CU)* mengirim sinyal kontrol ke komponen yang relevan.
  - Jika instruksi melibatkan operasi aritmetika atau logika, **ALU** akan melakukan perhitungan.
  - Jika instruksi melibatkan akses memori, mikroprosesor akan membaca atau menulis data ke memori.
  - Hasil eksekusi disimpan di register atau memori, tergantung pada jenis instruksi.
- 

#### **4. Store (Penyimpanan Hasil)**

*Tujuan: Menyimpan hasil eksekusi instruksi.*

*Proses:*

- Hasil operasi dari ALU atau data yang dibaca dari memori disimpan di register yang sesuai (misalnya, **Accumulator**).
  - Jika instruksi melibatkan penulisan data ke memori, hasil tersebut dikirim ke memori melalui **Data Bus**.
- 

#### **5. Repeat (Pengulangan)**

- *Tujuan: Mengulangi siklus untuk instruksi berikutnya.*

• *Proses:*

- **Program Counter (PC)** telah diperbarui selama tahap **Fetch** untuk menunjuk ke instruksi berikutnya.

- Siklus instruksi diulang dari tahap **Fetch** hingga **Store** untuk setiap instruksi berikutnya.

---

#### Diagram Siklus Instruksi:

1. **Fetch** → 2. **Decode** → 3. **Execute** → 4. **Store** → (Kembali ke **Fetch**)

---

#### Contoh Sederhana:

Misalkan mikroprosesor menjalankan instruksi **ADD A, B** (menambahkan nilai di register A dan B):

1. **Fetch**: Instruksi **ADD A, B** diambil dari memori dan disimpan di **IR**.
2. **Decode**: Instruksi didekode menjadi operasi penjumlahan dan menentukan bahwa data diambil dari register A dan B.
3. **Execute**: **ALU** melakukan penjumlahan nilai dari register A dan B.
4. **Store**: Hasil penjumlahan disimpan di register A (atau register lain, tergantung desain mikroprosesor).

### 3. Bagaimana tahapan penyimpanan data pada prosesor, berikan contohnya

Penyimpanan data pada prosesor melibatkan beberapa tahapan dan komponen yang bekerja bersama untuk memastikan data dapat diakses dan diproses dengan cepat. Berikut adalah tahapan penyimpanan data pada prosesor beserta contohnya:

#### A. Tahapan Penyimpanan Data pada Prosesor

1. **Pengambilan Data dari Memori Utama (RAM):**

- *Proses:* Data yang diperlukan oleh prosesor diambil dari memori utama (RAM) melalui **Memory Address Register (MAR)** dan **Memory Data Register (MDR)**.
- *Detail:*
  - **MAR** menyimpan alamat memori tempat data berada.
  - **MDR** menyimpan data yang dibaca dari atau ditulis ke memori.
- *Contoh:* Jika prosesor memerlukan data dari alamat memori **0x1000**, **MAR** akan menyimpan **0x1000**, dan data dari alamat tersebut akan dibaca ke **MDR**.

## 2. Penyimpanan Data di Cache Memory:

- *Proses:* Data yang sering diakses disimpan di cache memory untuk mempercepat akses di masa mendatang.
- *Detail:*
  - Cache memory adalah memori kecil yang sangat cepat yang terletak di dalam atau dekat prosesor.
  - Cache memiliki beberapa level (**L1**, **L2**, **L3**), dengan **L1** yang paling cepat dan paling kecil.
- *Contoh:* Jika prosesor mengakses data dari alamat **0x1000** beberapa kali, data tersebut akan disimpan di cache **L1** untuk akses yang lebih cepat.

## 3. Penyimpanan Data di Register:

- *Proses:* Data yang sedang diproses disimpan di register, yang merupakan memori tercepat di dalam prosesor.
- *Detail:*



- Register adalah komponen kecil yang dapat menyimpan data sementara, seperti operand, hasil perhitungan, atau alamat memori.
- Contoh register: **Accumulator** (menyimpan hasil operasi ALU), **Program Counter (PC)** (menyimpan alamat instruksi berikutnya), dan **Instruction Register (IR)** (menyimpan instruksi yang sedang diproses).
- **Contoh:** Jika prosesor melakukan penjumlahan  $A + B$ , nilai  $A$  dan  $B$  akan disimpan di register sebelum diproses oleh ALU.

#### 4. Penyimpanan Hasil Operasi:

- **Proses:** Hasil operasi (misalnya, hasil perhitungan ALU) disimpan kembali di register atau memori.
- **Detail:**
  - Hasil operasi dapat disimpan di register (seperti Accumulator) untuk digunakan dalam operasi berikutnya.
  - Jika hasil tersebut perlu disimpan secara permanen, data akan ditulis kembali ke memori utama (RAM).
- **Contoh:** Setelah prosesor menghitung  $A + B = C$ , hasil  $C$  disimpan di register Accumulator. Jika hasil tersebut perlu disimpan di memori, prosesor akan menuliskannya ke alamat memori yang ditentukan.

#### 5. Penulisan Data ke Memori Utama (RAM):

- **Proses:** Data yang telah diproses atau diubah ditulis kembali ke memori utama.

- **Detail:**
    - Prosesor menggunakan **Memory Address Register (MAR)** untuk menentukan alamat memori tujuan.
    - Data dari register (seperti MDR) ditulis ke alamat memori yang ditentukan.
  - **Contoh:** Jika hasil perhitungan *C* perlu disimpan di alamat memori *0x2000*, prosesor akan menulis nilai *C* ke alamat tersebut.
- 

## **B. Contoh Lengkap Tahapan Penyimpanan Data**

Misalkan prosesor menjalankan instruksi:

**ADD [0x1000], [0x1004], [0x2000]**

Artinya: Tambahkan nilai di alamat *0x1000* dan *0x1004*, lalu simpan hasilnya di alamat *0x2000*.

### **1. Fetch:**

- Instruksi diambil dari memori dan disimpan di **Instruction Register (IR)**.

### **2. Decode:**

- Instruksi didekode menjadi operasi penjumlahan dengan operand di alamat *0x1000* dan *0x1004*, serta tujuan penyimpanan di *0x2000*.

### **3. Pengambilan Data:**

- Prosesor membaca nilai dari alamat *0x1000* dan *0x1004*:
  - Alamat *0x1000* dibaca, nilai 5 disimpan di register sementara.

- Alamat *Ox1004* dibaca, nilai 7 disimpan di register sementara.

#### 4. Eksekusi:

- **ALU** melakukan penjumlahan:  $5 + 7 = 12$ .
- Hasil 12 disimpan di **Accumulator**.

#### 5. Penyimpanan Hasil:

- Prosesor menulis hasil 12 ke alamat memori *Ox2000*:
  - **MAR** diisi dengan alamat *Ox2000*.
  - **MDR** diisi dengan nilai 12.
  - Nilai 12 ditulis ke alamat *Ox2000* di memori utama.