**Introduction to the 8086 Microprocessor**

• **The 8086 microprocessor is an 8-bit/16-bit microprocessor** designed by Intel and is the first member of the x86 family.

• Its architecture is based on a **complex instruction set computer (CISC)**, meaning it supports a wide range of instructions.

• It has a **20-bit address bus**, allowing it to address up to **1 MB of memory**, and a 16-bit data bus.

• Unlike a microcontroller, the 8086 is a microprocessor with all the functions of a CPU but **cannot be used stand-alone as it has no internal memory (RAM/ROM) or peripherals**.

**Core Architecture and Pipelining**

• The internal architecture is divided into two main units: the **Bus Interface Unit (BIU)** and the **Execution Unit (EU)**.

• A key advantage of the 8086 is its **support for pipelining**, where the BIU fetches the next instruction while the EU executes the current one.

• This parallel process of fetching, decoding, and executing instructions increases the overall speed of the microprocessor.

**Memory Segmentation**

• The 8086 uses a **segmented memory architecture**, dividing the 1MB of addressable memory into segments to increase fetching and execution speed.

• A 20-bit physical address is generated using a 16-bit segment address and a 16-bit offset address.

• The physical address is calculated using the formula: **Physical Address = Segment Address x 10H + Offset Address**.

• The 8086 works with four 64KB segments at a time within the 1MB memory space.

**The Bus Interface Unit (BIU)**

• The BIU **provides the interface to external memory and I/O devices** and is responsible for fetching instructions, transferring data, and generating physical addresses.

• It contains the **Segment Registers (CS, DS, SS, ES)**, the Instruction Pointer (IP), an address generation circuit, and a 6-byte pre-fetch queue.

• The **6-byte pre-fetch queue** stores instructions before they are executed, which is essential for pipelining. This queue is flushed whenever a branch or jump instruction occurs.

**The Execution Unit (EU)**

• The EU **fetches instructions from the BIU's queue, decodes them, and executes them**.

• Its main components include the **Arithmetic Logic Unit (ALU)**, general-purpose registers, special-purpose registers, and the flag register.

• It contains four 16-bit general-purpose registers: **AX (accumulator)**, **BX (base address register)**, **CX (count register)**, and **DX (data register)**.

• The 16-bit **Flag/Status Register** has 9 flags (6 status flags and 3 control flags) that recognize or change the state of the microprocessor after operations.

**Control and Decode Units**

• The **Decode Unit** translates machine code instructions from the prefetch queue into micro-operations that the EU can execute.

• The **Control Unit** manages the overall operation of the microprocessor, acting as the central coordinator that directs the flow of instructions and data between the different components.

A diagram of a block diagram

AI-generated content may be incorrect.

**Execution Flow in the 8086 Architecture**

The execution process in the 8086 is not sequential like in older processors; instead, it uses a parallel process called **pipelining**, which is enabled by its dual-unit architecture (BIU and EU).

1. **Physical Address Calculation**: The process begins in the **Bus Interface Unit (BIU)**. The BIU's address generation circuit calculates the 20-bit physical address of the instruction to be fetched. This is done by combining a 16-bit segment address (from a segment register like CS) with a 16-bit offset address (from the Instruction Pointer, IP) using the formula: Physical Address = (Segment Address x 10H) + Offset Address.

2. **Instruction Fetch & Queuing**: Once the address is calculated, the BIU fetches the instruction from that memory location. The fetched instruction is then stored in a **6-byte pre-fetch queue**, which operates on a First-In, First-Out (FIFO) basis. The size of the instruction determines how much space it occupies in the queue.

3. **Decoding**: The **Execution Unit (EU)** pulls the next instruction from the front of the BIU's pre-fetch queue. The instruction is then sent to the control system, where the **Decode Unit** translates the machine code into micro-operations that the EU can understand. The control system then sends signals throughout the microprocessor, indicating which operation to perform and what data to use.

4. **Execution**: Following the decoded instructions, the EU fetches the required data from the **General-Purpose Registers (GPRs)** like AX, BX, etc.. This data is then processed by the **Arithmetic Logic Unit (ALU)**, which performs the specified calculation (e.g., ADD, SUB, MUL).

5. **Update Status Flags**: After the ALU completes its operation, the values in the **Flag Register** are updated dynamically to reflect the result (e.g., setting the Zero Flag if the result is zero).

6. **Parallel Operation (Pipelining)**: **Crucially, while the EU is busy decoding and executing an instruction (Steps 3-5), the BIU does not sit idle**. It continuously fetches subsequent instructions from memory and fills the pre-fetch queue, ensuring that the EU always has instructions ready for execution. This parallel fetching, decoding, and execution process is the essence of pipelining and is the primary reason for the architectural division into the BIU and EU.

This pipelined approach significantly increases the overall execution speed compared to the sequential fetch-decode-execute cycle of older processors like the 8085. However, this queue system fails during branch or jump instructions, which require the queue to be flushed and refilled from a new memory address

**Advantages of the 8086 Architecture**

• The segmented memory architecture allows the 8086 to **address up to 1 MB of memory** with a 16-bit data bus.

• It has a **powerful instruction set**, with many instructions capable of performing multiple operations at once.

• The presence of two main units (BIU and EU) allows for the **efficient execution of instructions** through pipelining.

**Disadvantages of the 8086 Architecture**

• The architecture is **complex and can be difficult to program**, particularly for those unfamiliar with assembly language and segmented memory management.

• Compared to modern processors, it has **limited performance** due to a slower clock speed and can only address a maximum of 1 MB of memory.

• It **lacks built-in features** common in modern microprocessors, such as hardware floating-point support and virtual memory management