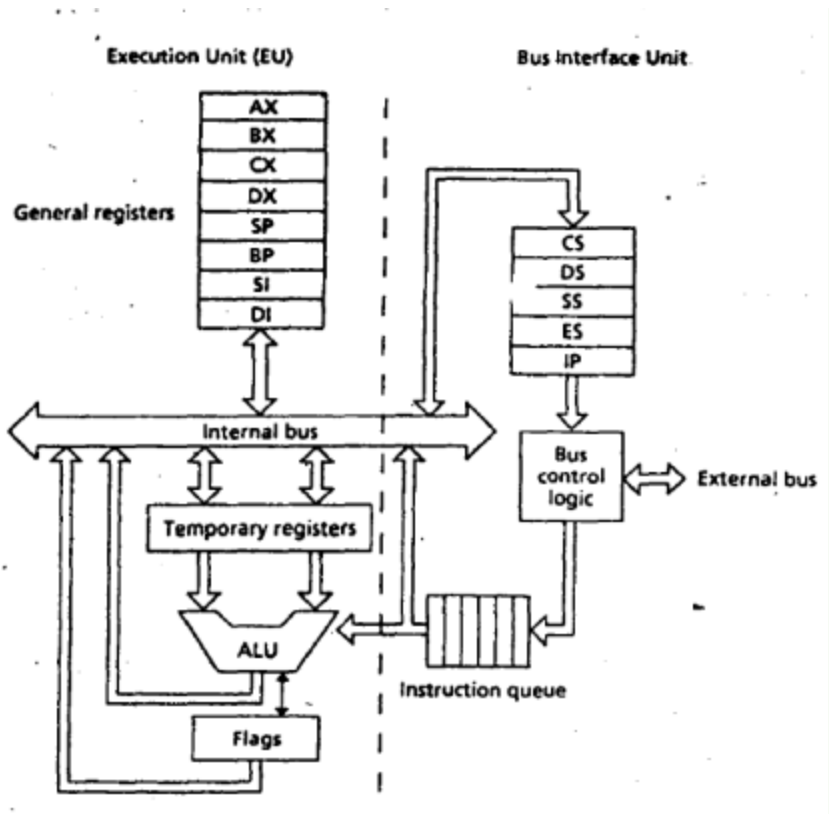




Assembly : Introduction

👤 Author	🖥️ Md. Zarif Ul Alam
🕒 Date Created	@March 1, 2021 10:18 PM
🏷️ Tag	CSE315 Microprocessors Microcontrollers and Embedded Systems

8086 Microprocessor Organization



- Execution Unit (EU)
 - ALU
 - Registers
- Bus Interface Unit (BIU)
 - facilitates communication between the EU and the memory or I/O circuits
 - responsible for transmitting address, data and control signals on the buses

Register

- Data Register
 - `ax` (2 byte): `accumulator`
 - `ah` (1 byte)
 - `al` (1 byte)
 - `bx` : `base`
 - `bh`
 - `bl`
 - `cx` : `count`
 - `ch`
 - `cl`
 - `dx` : `data`

- dh
- dl
- Segment Register
 - `cs` : code
 - `ds` : data
 - `ss` : stack
 - `es`
- Pointer and Index Register
 - `si` : source
 - `di` : destination
 - `sp` : stack
 - `bp` : base
- Flags Register

Code

- **Data Segment initialization**

```
; BASICALLY STARTING ADDRESS IS STORED
MOV AX, @DATA ; AX : ACCUMULATOR DATA REGISTER : preferred register to use in arithmetic logic and transfer operation
MOV DS, AX ; DS : DATA SEGMENT
```

- **INPUT single key from user**

```
MOV AH, 1
INT 21H ; interrupt number
```

- **OUTPUT single character**

```
; DL for 1 byte , DX for 1 word / 2 byte
MOV DL, NUM ; stored in DL for display
MOV AH, 2
INT 21H
```

- **OUTPUT string**

- string must end with a dollar `$`
- LEA : Load Effective Address
 - puts a **copy of the source offset address** into the destination
 - i.e. doesn't store the string in register

```
LEA DX, MSG1 ; DX : USED IN IO and MUL,DIV
MOV AH, 9 ; AH,9 used for character string output
INT 21H;
```

- **return control to DOS**

```
;DOS EXIT
MOV AH, 4CH
INT 21H
```

Q/A

- Why do we need registers ? Why not use memory (like ram) ?

- Ans1 : <https://stackoverflow.com/questions/45581423/why-are-registers-needed-why-not-only-use-memory>

▲

12

▼

🕒

Because RAM is slow. Very slow.

Registers are placed *inside* the CPU, right next to the ALU so signals can travel almost instantly. They're also the **fastest memory type** but they take significant space so we can have only a limited number of them. Increasing the number of registers increases

- die size
- distance needed for signals to travel
- work to save the context when switching between threads
- number of bits in the instruction encoding

Read [If registers are so blazingly fast, why don't we have more of them?](#)

More commonly used data will be placed in caches for faster accessing. In the past caches are very expensive so they're an optional part and can be purchased separately and plug into a socket outside the CPU. Nowadays they're often in the same die with the CPUs. Caches are constructed from SRAM cells which are smaller than register cells but maybe tens or hundreds of times slower.

Main memory will be made from DRAM which needs only one transistor per cell but are thousands of times slower than registers, hence we can't work with only DRAM in a high-performance system. However some embedded system do make use of [register file](#) so registers are also main memory

More information: [Can we have a computer with just registers as memory?](#)

- Ans2 : <https://stackoverflow.com/questions/2360997/assembly-why-are-we-bothering-with-registers>
- If registers are so blazingly fast, why don't we have more of them ?
 - Ans (didn't understand everything) : <https://stackoverflow.com/questions/6079215/if-registers-are-so-blazingly-fast-why-dont-we-have-more-of-them>
- Why memory to memory move is not allowed ?
 - Ans : <https://stackoverflow.com/questions/11953352/why-ia32-does-not-allow-memory-to-memory-mov>

The answer involves a fuller understanding of RAM. Simply stated, RAM can only be in two states, read mode or write mode. If you wish to copy one byte in ram to another location, you *must* have a temporary storage area outside of RAM as you switch from read to write.