

# Microprocessore 8086

## INDICE

- Intel 8086
- Diagramma PIN
- Diverse modalità di indirizzamento
- Segmentazione della memoria nell'8086
- Registro dei Flag
- Architettura della pipeline nel microprocessore 8086
- Vantaggio del pipelining

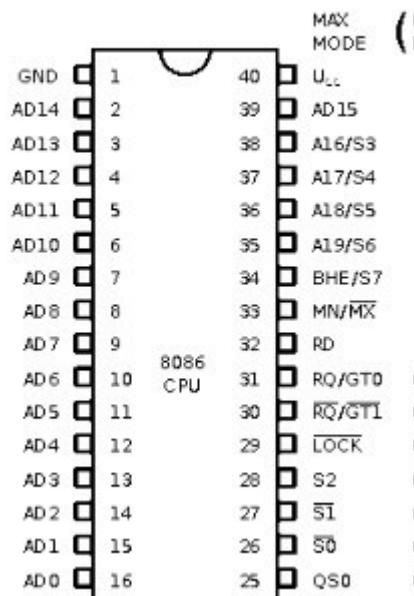
## Cos'è il microprocessore 8086?



## INTEL 8086:

- Microprocessore 8086 inventato per la prima volta da INTEL nel 1976.
- [8086](#) è dotato di microprocessore HMOS basato su canali N a 16 bit.
- Questo ha due modalità; minimo e massimo.
- 8086 ha un totale di venti (20) linee di indirizzo
- 8086 ha sedici (16) linee dati.

## Schema PIN del microprocessore 8086:



## Cos'è la modalità di indirizzamento?

*"La modalità di indirizzamento è il modo per specificare la posizione in memoria di un dato che deve essere gestito da un'istruzione."*

Abbiamo bisogno di diversi tipi di modalità di indirizzamento perché fornisce flessibilità al programmatore per accedere ai dati.

## Quali sono i tipi di modalità di indirizzamento in 8086?

Di seguito vengono illustrati i diversi tipi di modalità di indirizzamento:

### Indirizzamento registro:

L'operando è un registro.

Esempio - `MOV AX, BX`

### Indirizzamento immediato:

L'istruzione stessa comprende gli operandi.

Esempio - `MOV AX, 5000H`

**Indirizzamento diretto:**

L'istruzione specifica l'indirizzo dell'operando.

Esempio - *MOV, AX, 9000H*

**Indirizzamento indicizzato:**

L'operando viene specificato utilizzando uno tra SI e DI come registro indice, insieme a un offset opzionale. L'indirizzo dell'operando viene acquisito sommando le informazioni del registro indice con l'offset, se presente.

Esempio - *MOV AX, [SI]* o *MOV AX, [SI + 1000H]*

**Indirizzamento basato:**

L'operando viene specificato utilizzando uno tra BX e BP come registro di base, insieme a un offset opzionale. L'indirizzo dell'operando viene acquisito sommando le informazioni del registro base con l'offset, se presente.

Esempio - *MOV AX, [BX]* o *MOV AX, [BP + 1000H]*

**Indirizzamento indicizzato in base:**

L'operando viene specificato utilizzando uno tra SI e DI come registro indice e quelli di BX e BP come registro base, insieme a un offset opzionale. L'indirizzo dell'operando viene acquisito sommando le informazioni del registro indice con il contenuto del registro base e l'offset, se presente.

Esempio - *MOV AX, [SI + BX]* o *MOV AX, [DI + BP + 1000H]*

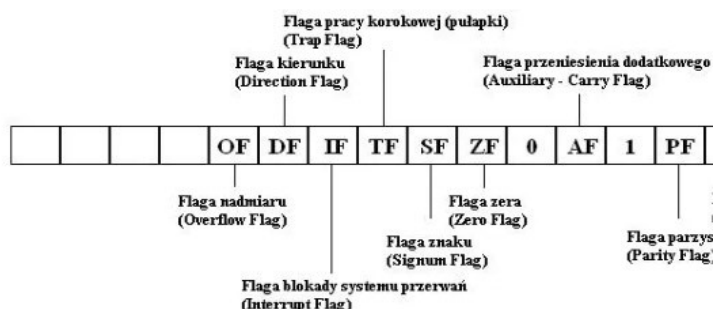
## Segmentazione della memoria nel microprocessore 8086:

Il microprocessore 8086 ha 20 pin di indirizzo, quindi il numero massimo di posizioni di memoria che possono essere collegate a 8086 sono  $2^{20} = 1$  MB di posizione o 16 blocchi di 64 K posizioni. La memoria collegata all'8086 si divide nei seguenti quattro segmenti:

1. **CS - Segmento di memoria del Codice:** Viene utilizzato per memorizzare il codice delle istruzioni di un programma.
2. **DS - Dichiarazione sulla memoria Dati:** Viene utilizzato per memorizzare byte / parole di dati.
3. **ES - Segmento di memoria Extra:** È un segmento aggiuntivo per la memorizzazione dei dati.
4. **SS - Segmento di memoria dello Stack:** Viene utilizzato per memorizzare lo stack di dati utilizzando l'istruzione PUSH / POP.

## Diversi flag nel microprocessore 8086:

1. **S (Flag d segno)** - Impostato quando la risposta del calcolo è negativa.
2. **Z (Zero)** - Impostato quando il calcolo dell'istruzione precedente è zero.
3. **P (Parità)** - Impostato quando il byte inferiore contiene un numero pari di uno.
4. **C (Riporto)** - Quando ci sono riporti nel calcolo.
5. **T (Trappola)** - quando il processore entra in modalità istruzione passo singolo.
6. **I (Interrupt)** - Vengono identificate le interruzioni mascherabili.
7. **D (Direzione)** - Nella manipolazione delle stringhe.
8. **AC (Riporto ausiliario)**
9. **O (Overflow)** - Quando il risultato è maggiore per adattarsi ai registri.



## **Architettura della pipeline nel microprocessore 8086:**

L'idea fondamentale dell'architettura pipeline è quella di suddividere l'elaborazione delle istruzioni di un computer in una serie di fasi indipendenti (come "pre-fetch", "fetch", "decode", "execute" ecc.) Con memorizzazione alla fine del ogni passaggio.

Ciò consente al controllo del computer di indicare la velocità di elaborazione del passaggio più lento, risultando molto più veloce del tempo necessario per eseguire tutti i passaggi in sequenza.

La pipeline indica come ogni passaggio acquisisce informazioni e ogni passaggio è collegato a quello successivo.

In questo ci sono 2 unità separate

- La "Bus Interface Unit" (**BIU**)
- La "Execution Unit" (**UE**).

La BIU esegue tutte le operazioni del bus per l'unità di esecuzione. I dati sono in comunicazione tra la CPU e le memorie e il kit input output su richiesta dall'UE. Durante questo se l'UE è attiva nell'implementazione dei comandi, la BIU "guarda avanti" e porta più istruzioni dalla memoria. In questo modo, nell'8086 viene implementato un tipo di "Fetch-Execute-Pipeline".

## **Alcuni dei vantaggi e degli svantaggi del pipelining**

### **I vantaggi del pipelining sono:**

- Il tempo di ciclo del chip è relativamente inferiore. Il pipelining non riduce al minimo il tempo necessario per completare un'istruzione; piuttosto aumenta la quantità di istruzioni che possono essere elaborate contemporaneamente e riduce il ritardo tra le istruzioni complete.
- Le molteplici fasi senza aumento della pipeline significano che è possibile elaborare più comandi contemporaneamente e il minor ritardo tra i comandi. Ogni microprocessore simulato prioritario prodotto oggi utilizza almeno condutture a due stadi di circa 30-40 stadi.
- Quando si utilizza il pipelining, la CPU ALU è progettata per funzionare velocemente, ma con un design più complicato.
- Il concetto di pipeline migliora le prestazioni all'interno di un core non pipeline di uno stadio di fattore  $n$  e inoltre il codice è impeccabile per l'implementazione della pipeline.
- Le CPU pipeline in generale funzionano a una frequenza di clock molto più alta della RAM e questo migliora le prestazioni complessive del processore.

## **Gli svantaggi del pipelining sono:**

- Questo è un chip non pipeline, più semplice nel design e più economico da fabbricare, implementa solo una singola istruzione alla volta. Ciò evita che le istruzioni sequenziali vengano eseguite simultaneamente.
- Questo tipo di processore ha più latenza delle istruzioni rispetto ad alcuni chip non pipeline. Il funzionamento di un processore pipeline è molto più difficile da prevedere e può variare notevolmente per varie applicazioni.

## **Funzioni del BIU e del microprocessore EU 8086**

### **Unità di esecuzione (UE):**

Una ALU a 16 bit nell'**UE** mantiene lo stato della CPU e il flag di controllo e distribuisce i registri generali e l'operando di istruzione, ecc. Tutti i registri e i datapath dell'**UE** hanno una lunghezza di 16 bit per le comunicazioni interne.

L'**UE** non ha alcun collegamento con il BUS dati. Questo acquisisce indicazioni dalla BIU tramite coda. Allo stesso modo, non appena un'istruzione deve accedere alla memoria o alle periferiche, l'**UE** chiede alla BIU di accedere o di conservare le informazioni. La BIU, allora, trasferisce l'indirizzo a prevede l'ingresso nell'**UE** per l'intero dato.

### **Unità di interfaccia bus (BIU):**

Le **BIU** vengono impiegate in modo diverso per adattarsi alla disposizione, alle caratteristiche delle prestazioni dei vari bus. La BIU implementa tutte le operazioni di bus per l'**UE**.

La dimensione della coda in BIU consente di mantenere l'**UE** fornita con istruzioni precaricate nella maggior parte degli stati senza monopolizzare il bus di sistema. La BIU 8086 normalmente riceve due byte per recupero.