

# Il processore Intel 8086

Il **processore Intel 8086**, introdotto sul mercato nel giugno 1978 è stato il primo microprocessore a **16 bit**; esso fu utilizzato nel 1981 per il Personal Computer IBM XT.

È il progenitore della famiglia di processori Intel (x86), che continuano ad essere compatibili con esso (cioè i programmi realizzati per l'8086 continuano a funzionare sui processori successivi, fino al Pentium 4 uscito come prima versione nel 2000).



## SINTESI DELLE CARATTERISTICHE DEL PROCESSORE 8086

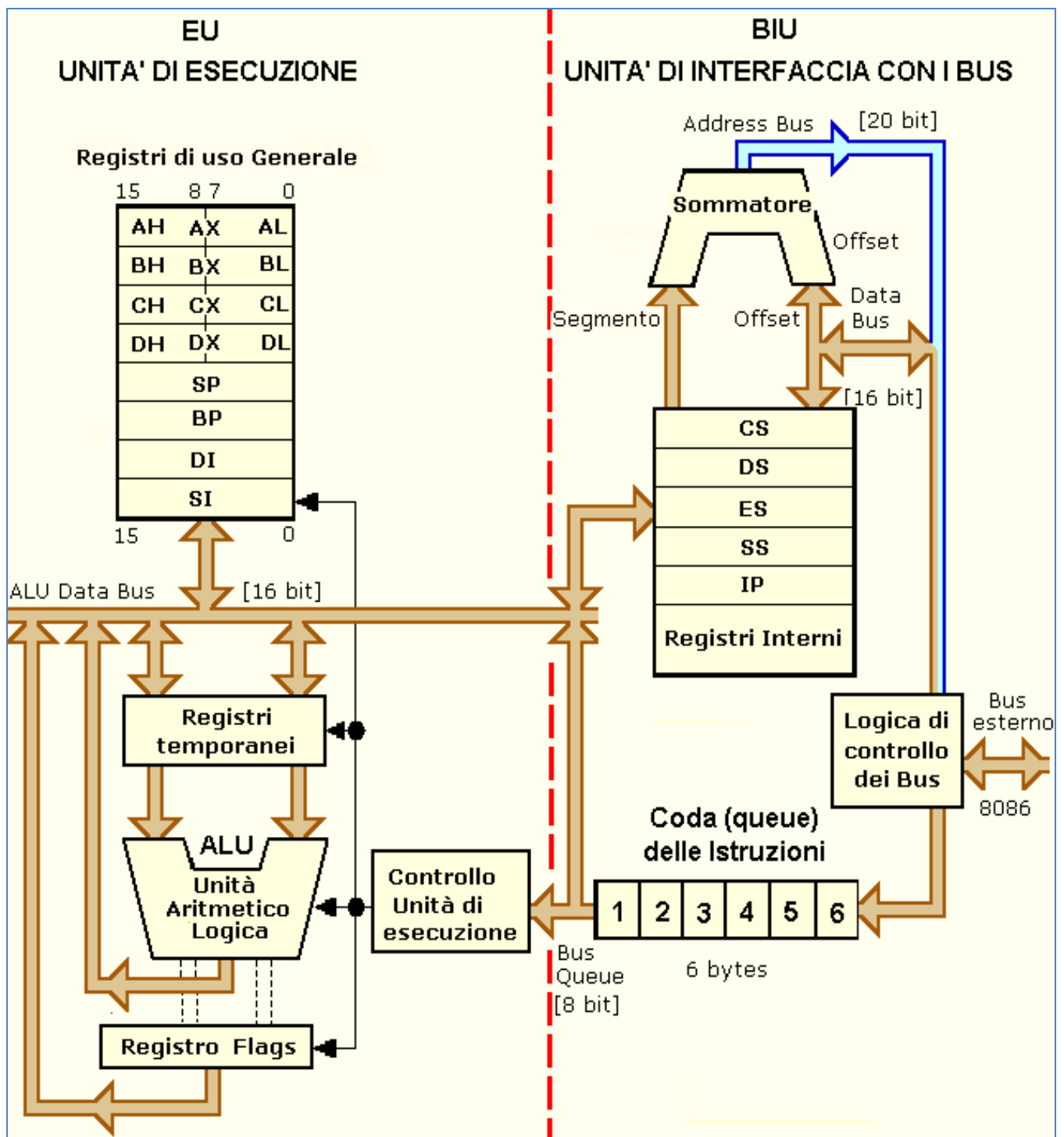
- frequenza originale del clock 4,77 MHz (primi PC IBM), fino a 10 MHz;
- 14 registri a 16 bit;
- bus dati a 16 bit;
- bus indirizzi a 20 bit;
- indirizzamento di 1 MB di memoria (organizzazione della memoria a segmenti di 64 KB);
- dimensione delle celle di memoria: 1 byte;
- contenitore a 40 pin disposti su due linee da 20 ciascuna;
- set di 70 istruzioni di base: operazioni aritmetiche (binarie e BCD) su numeri con e senza segno a 8 e 16 bit, manipolazione di stringhe, operazioni logiche, istruzioni di salto, istruzioni di rotazione e di shift ...;
- può supportare il coprocessore matematico 8087 per le operazioni in virgola mobile (floating point).

## EVOLUZIONE DEI PROCESSORI INTEL FINO AL 2006

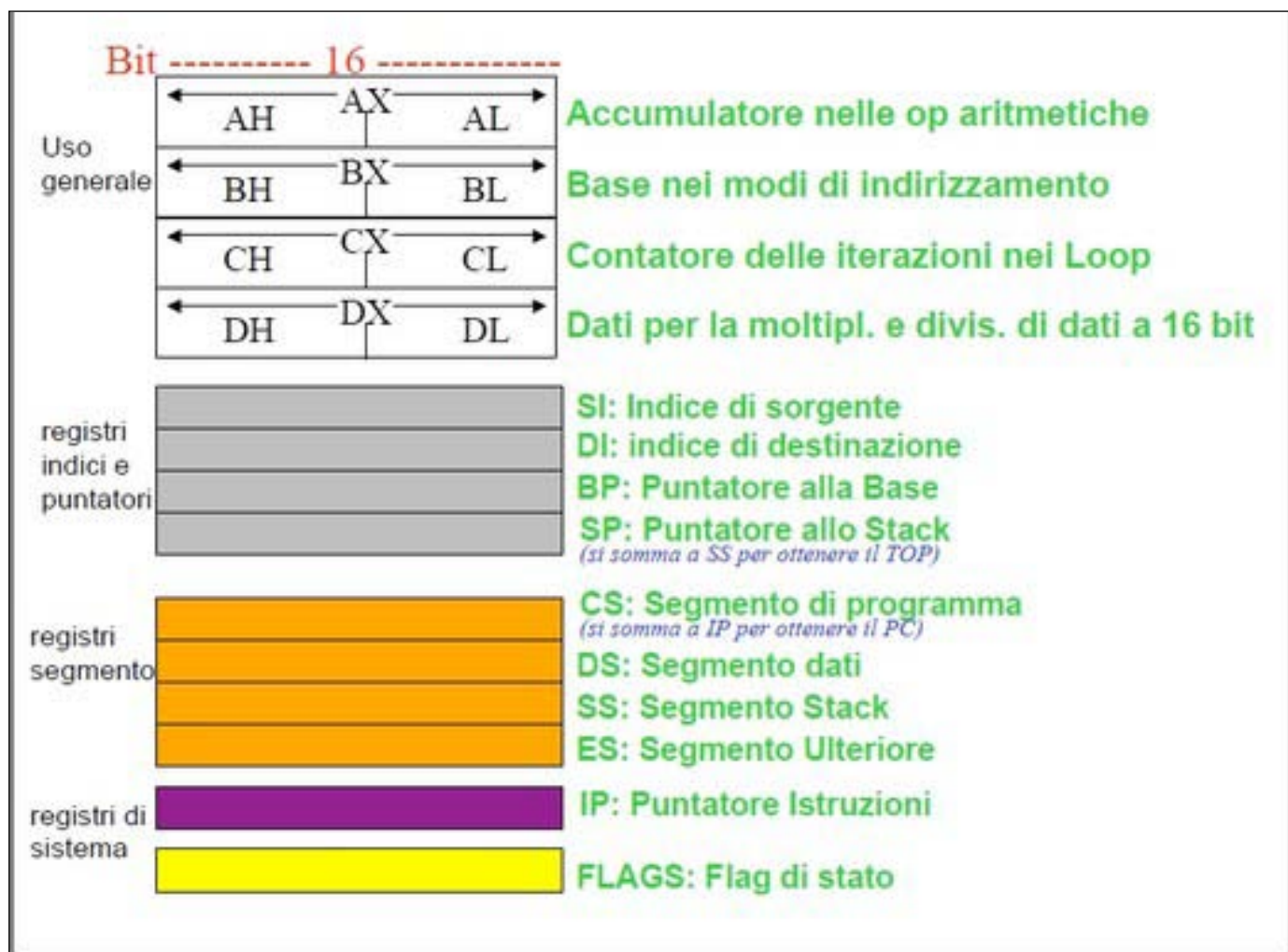
### TABELLA RIASSUNTIVA ARCHITETTURE INTEL

Nome	anno	architettura	PIN n.	Transistor n.	Clock CPU MHz	Clock BUS MHz	Abus linee	DBus linee	Core n.	Stadi pipeline n.	Cache livelli	Volt v
8086	1979	x-86	20	29.000	4	10	20	16	1	-	-	5
80286	1982	x-86	68	134.000	12	10	24	16	1	-	-	5
80386	1986	IA-32	68	275.000	33	33	32	32	1	-	-	5
80486	1989	IA-32	168	1.200.000	133	50	32	32	1	-	L1	5
Pentium	1993	IA-32	296	3.100.000	300	66	32	64	1	5	L1	3,3
Pentium Pro	1996	IA-32	387	5.500.000	200	66	36	64	1	10	L1-L2	3,3
Pentium II	1997	IA-32	242	7.500.000	300	100	36	64	1	20	L1-L2	2
Pentium III	1999	IA-32	370	9.500.000	1200	133	36	64	1	20	L1-L2	2
Pentium 4	2000	IA-32	468	42.000.000	3400	400	36	64	1	31	L1-L2-L3	1,5
Itanium	2001	IA-64	610	2.000.000.000	1600	533	64	64	1/2	10	L1-L2-L3	1,4
Pentium D	2005	IA-32	775	125.000.000	3600	400	36	64	2/4	10	L1-L2-L3	1,4
Core 2 Duo	2006	IA-32	775	291.000.000	3000	533	36	64	2	10	L1-L2-L3	1

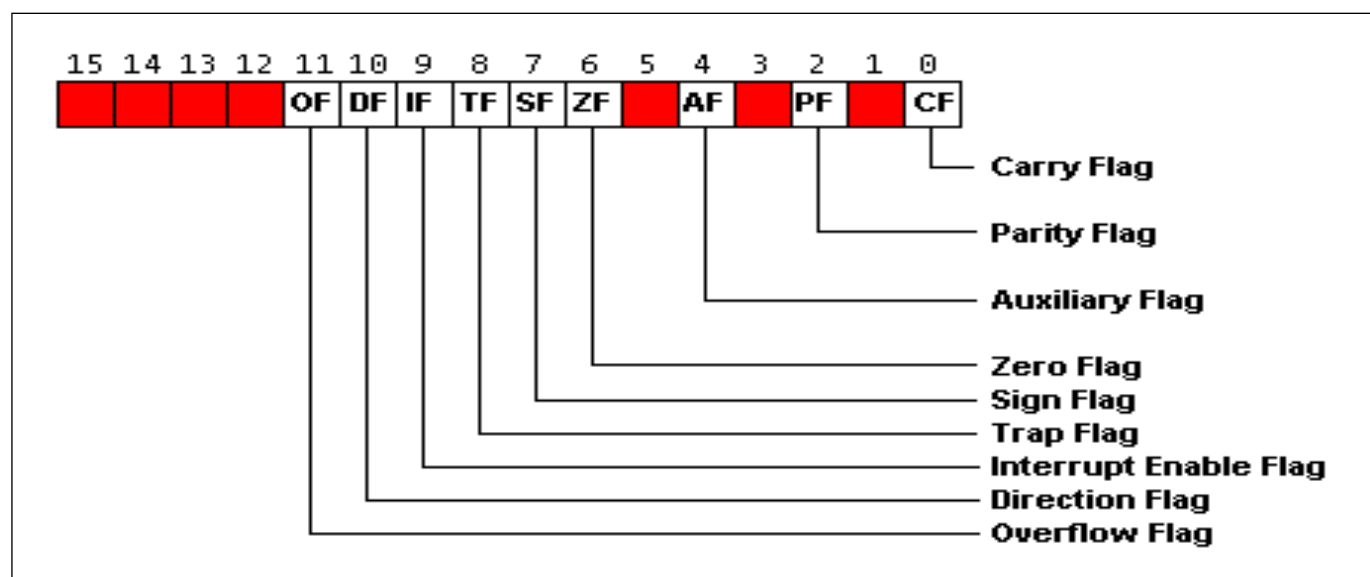
# Architettura interna del processore Intel 8086



## TIPOLOGIA DEI REGISTRI del processore INTEL 8086



## REGISTRO dei FLAGS



## REGISTRO dei FLAGS (Intel 8086)

I bit colorati in rosso sono inutilizzati e in ogni caso, si tratta di campi riservati alle future CPU; i manuali del processore Intel **8086** consigliano di evitare la modifica del contenuto di questi bit.

I **9 bit** colorati in **bianco** rappresentano ciascuno un differente **flag** (segnalatore).

Nel caso generale, ciascuno di questi bit indica se, dopo lo svolgimento di un'operazione logico-aritmetica, una determinata condizione si è verificata o meno; se il bit vale **1**, la condizione si è verificata, mentre se il bit vale **0**, la condizione non si è verificata.

- Il flag **CF** o **Carry Flag** (segnalatore di riporto/prestito) vale **1** quando si verifica un riporto dopo un'addizione, o quando si verifica un prestito dopo una sottrazione; in caso contrario, **CF** vale zero. Questo flag può assumere anche altri significati durante operazioni specifiche.
- Il flag **PF** o **Parity Flag** (segnalatore di parità) viene largamente utilizzato per il controllo degli errori durante la trasmissione di dati tra due computer attraverso dispositivi come il modem; quando la CPU esegue determinate istruzioni logico aritmetiche, controlla gli **8** bit meno significativi del risultato prodotto dalle istruzioni stesse e attraverso **PF** indica se questi **8** bit contengono un numero pari o dispari di **1**. Se il numero di **1** è pari la CPU pone **PF=1**; se il numero di **1** è dispari la CPU pone **PF=0**. La ragione per cui **PF** si riferisce solo agli **8** bit meno significativi di un numero è legata al fatto che diversi protocolli di comunicazione tra computer prevedono la suddivisione dei dati trasmessi, in gruppi (pacchetti) di **8** bit; questi protocolli utilizzano per i dati la vecchia codifica **ASCII** a **7** bit che permette di rappresentare  $2^7=128$  simboli differenti. I **7** bit meno significativi di ogni pacchetto contengono un codice **ASCII**, mentre il bit più significativo contiene il cosiddetto **bit di parità**; un bit di parità di valore **1** prende il nome di **parità pari** e indica che il pacchetto è valido solo se i suoi **8** bit hanno un numero pari di **1**. Un bit di parità di valore **0** prende il nome di **parità dispari** e indica che il pacchetto è valido solo se i suoi **8** bit hanno un numero dispari di **1**; questo sistema si basa sul fatto che se la probabilità di inviare un bit sbagliato è bassa, la probabilità di inviare due bit sbagliati diventa enormemente più bassa.
- Il flag **AF** o **Auxiliary Flag** (segnalatore ausiliario) vale **1** quando si verifica un riporto dal nibble basso al nibble alto di un numero a **8** bit, o quando si verifica un prestito dal nibble alto al nibble basso di un numero a **8** bit; come vedremo in un apposito capitolo, **AF** viene utilizzato nell'aritmetica dei numeri in formato **BCD**, cioè in formato **Binary Coded Decimal** (decimale codificato in binario).
- Il flag **ZF** o **Zero Flag** (segnalatore di zero) vale **1** quando il risultato di una operazione è zero; in caso contrario si ha **ZF=0**.
- Il flag **SF** o **Sign Flag** (segnalatore di segno) vale **1** quando il risultato di una operazione ha il bit più significativo (**MSB**) che vale **1**; in caso contrario si ottiene **SF=0**. Come già sappiamo, i numeri interi con segno espressi in complemento a **2** sono negativi se hanno l'**MSB** che vale **1**, mentre sono positivi in caso contrario.

- Il flag **OF** o **Overflow Flag** (segnalatore di trabocco) vale **1** quando il risultato di una operazione tra numeri con segno è "troppo positivo" o "troppo negativo"; in sostanza, **OF=1** segnala un overflow di segno, cioè un risultato il cui bit di segno è l'opposto di quello che sarebbe dovuto essere.

I flags appena elencati, vengono modificati direttamente dalla **CPU**; nel **Flags Register** dell'**8086** sono presenti anche altri tre flags che vengono gestiti, invece, dal programmatore e servono a configurare il modo di operare della **CPU**.

- Il flag **TF** o **Trap Flag** (segnalatore di trappola) viene utilizzato da appositi programmi chiamati **debuggers** per scovare eventuali errori in un programma in esecuzione; se **TF** viene posto a **1**, la **CPU** genera una interruzione hardware per ogni istruzione eseguita e ciò permette ai debuggers (che intercettano l'interruzione) di analizzare, istruzione per istruzione, il programma in esecuzione (si parla in questo caso di **esecuzione a passo singolo**). Se non si deve effettuare nessun lavoro di "debugging", si consiglia vivamente di tenere **TF** a **0** in modo da evitare di rallentare il computer; questo flag viene inizializzato a **0** in fase di accensione del **PC**.
- Il flag **IF** o **Interrupt Enable Flag** (segnalatore di interruzioni abilitate) serve ad abilitare o a disabilitare la gestione delle interruzioni hardware che le varie periferiche inviano alla **CPU** per segnalare di voler intervenire; tutte le interruzioni hardware che possono essere abilitate o disabilitate tramite **IF** vengono chiamate **maskable interrupts** (interruzioni mascherabili).

Si può avere la necessità di disabilitare temporaneamente queste interruzioni quando, ad esempio, un programma sta eseguendo un compito piuttosto delicato che richiede la totale assenza di "interferenze" esterne; ponendo **IF=0** le interruzioni mascherabili vengono disabilitate (mascherate), mentre se **IF=1** le interruzioni mascherabili vengono abilitate ed elaborate dalla **CPU**.

Lo stato di **IF** non ha nessun effetto sulle interruzioni software (che vengono inviate esplicitamente dai programmi) e sulle cosiddette **NMI** o **Non Maskable Interrupts** (interruzioni non mascherabili); le **NMI** vengono inviate dall'hardware alla **CPU** nel caso si verificano gravi problemi nel computer come, ad esempio, errori in memoria o nei dati che transitano sui bus. Osservando la Figura 3 e la Figura 6 del precedente capitolo, si può notare nella piedinatura della **8086** e della **80386**, la presenza di un terminale chiamato **NMI**; proprio attraverso questo ingresso, le interruzioni non mascherabili vengono inviate alla **CPU**. Per motivi facilmente comprensibili, si consiglia di tenere **IF** a **0** per il minor tempo possibile; in fase di accensione del **PC**, il flag **IF** viene inizializzato a **1**.

- Il flag **DF** o **Direction Flag** (segnalatore di direzione) viene utilizzato in combinazione con le istruzioni della **CPU** che eseguono trasferimenti di dati e comparazioni varie tra due blocchi di memoria; abbiamo già visto che in questo caso la **CPU** utilizza **DS:SI** come indirizzo sorgente predefinito e **ES:DI** come indirizzo destinazione predefinito. Durante queste operazioni, i puntatori **SI** e **DI** vengono automaticamente aggiornati, cioè incrementati o decrementati; ponendo **DF=0** i puntatori vengono autoincrementati, mentre con **DF=1** i puntatori vengono autodecrementati. In fase di accensione del **PC**, il flag **DF** viene inizializzato a **0**.