

## Lezione 18-09-08

### **Architettura x86-32 (IA-32) (vedi slide da 51 a 54)**

L'evoluzione delle architetture da 16 a 32 bit (e quindi anche a 64) ha permesso una maggiore capacità di indirizzamento della memoria. Se si pensa che con 16 bit si potevano indirizzare  $2^{16}$  byte (cioè 64 KB) adesso con 32 bit si ottiene una capacità di 4 GB (e con 64 bit è addirittura difficile anche rappresentarla numericamente, ma supera di molto l'ordine dei terabyte!).

Il Pentium è un processore con architettura nativa a 32 bit (che in seguito è stata anche estesa alla versione a 64 bit) che permette un indirizzamento anche a 16 bit rendendosi compatibile con i processori precedenti.

Opera in due modi:

- Protected mode: realizzazione dell'architettura a 32 bit
- Real mode: emulazione dell'architettura a 16 bit.

L'emulazione del modo reale e cioè dell'ambiente DOS avviene in due modi:

- Modo reale con esecuzione delle istruzioni a 16 bit
- Modo protetto tramite il VM86, cioè con l'utilizzo di una macchina virtuale che opera emulando il comportamento logico del DOS.

## **Registri (vedi slide da 55 a 57)**

I registri possono essere suddivisi in 4 gruppi:

- Registri di dato
- Registri puntatore/di indirizzo
- Registri di segmento
- Registri di controllo.

## **Registri di dato (vedi slide 58)**

Sono AX (Accumulator Register), BX (Base Register), CX (Count Register) e DX (Data Register). Sono utilizzati per memorizzare operandi e risultato delle operazioni. Possono essere utilizzati come registri da 16 bit oppure come coppie di registri da 8 bit.

BX può anche essere utilizzato nel calcolo di indirizzi.

CX viene anche utilizzato come contatore da talune istruzioni.

DX contiene l'indirizzo di I/O in alcune istruzioni di I/O.

## **Registri puntatore (vedi slide 59)**

Sono IP (Instruction Pointer), SP (Stack Pointer), BP (Base Pointer), SI (Source Index) e DI (Destination Index).

IP contiene il puntatore alla prima istruzione da eseguire. Non può comparire esplicitamente come operando di una istruzione.

SP contiene il puntatore alla testa dello stack.

BP viene utilizzato come base per fare accesso all'interno dello stack.

SI e DI vengono utilizzati come registri indice.

## **Registri di segmento (vedi slide 60)**

Sono CS, DS, ES e SS.

Vengono utilizzati per costruire gli indirizzi fisici con i quali fare accesso in memoria.

Contengono i puntatori all'inizio dei segmenti di codice, di dato, di dato supplementare e di stack, rispettivamente.

## **Process Status Word (PSW) (vedi slide 61)**

È composta da 16 bit, ma solo 9 di questi sono usati. Ogni bit corrisponde ad un flag.

I flag si dividono in:

- flag di condizione
- flag di controllo.

## **Flag di condizione (vedi slide 62)**

Vengono automaticamente scritti al termine di varie operazioni:

- SF (Sign Flag): coincide con il MSB del risultato dopo una operazione aritmetica
- ZF (Zero Flag): vale 1 se il risultato è nullo, 0 altrimenti
- PF (Parity Flag): vale 1 se il numero di 1 negli 8 bit meno significativi del risultato è pari, 0 altrimenti
- CF (Carry Flag): dopo le operazioni aritmetiche vale 1 se c'è stato riporto (somma) o prestito (sottrazione); altre istruzioni ne fanno un uso particolare
- AF (Auxiliary Carry Flag): usato nell'aritmetica BCD; vale 1 se c'è stato riporto (somma) o prestito (sottrazione) dal bit 3
- OF (Overflow Flag): vale 1 se l'ultima istruzione ha prodotto un overflow.

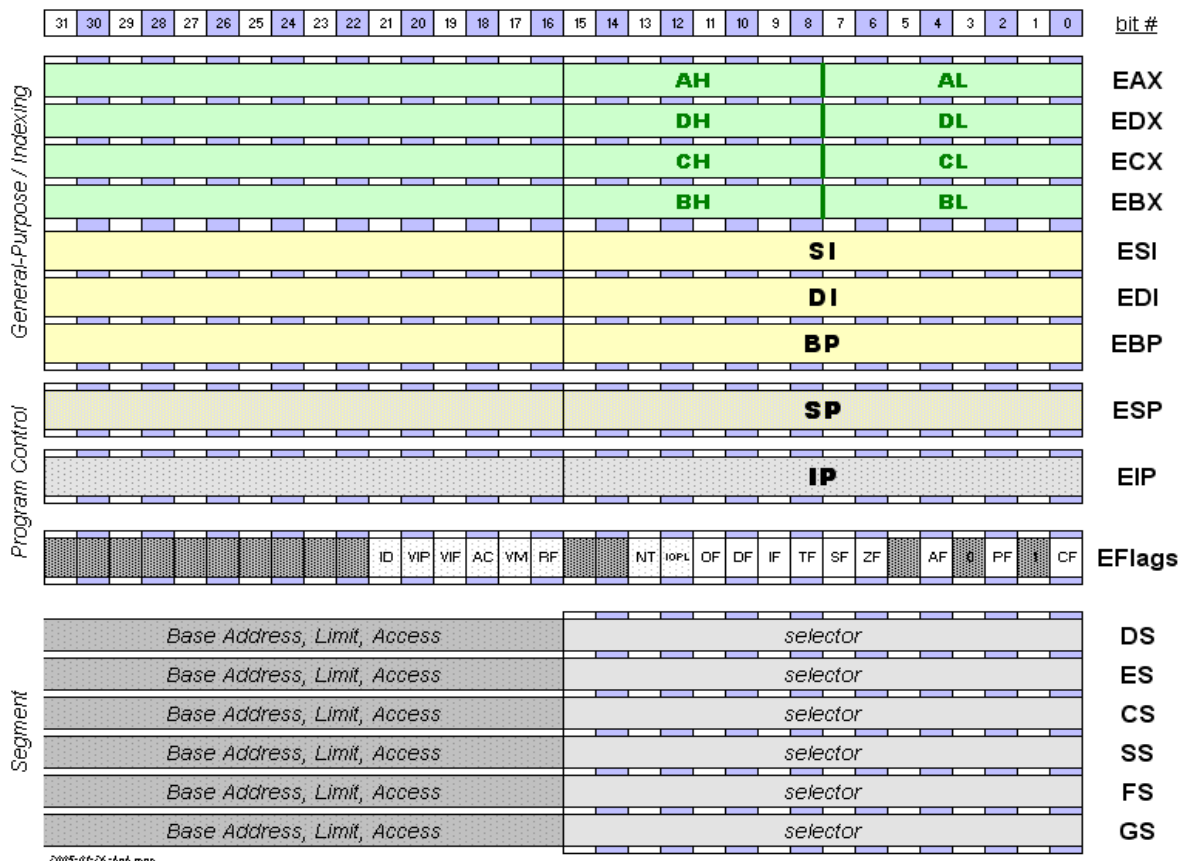
## **Flag di controllo (vedi slide 63)**

Possono venire scritti e manipolati da apposite istruzioni e servono a regolare il funzionamento di talune funzioni del processore:

- DF (Direction Flag): utilizzato dalle istruzioni per la manipolazione delle stringhe; se vale 0 le stringhe vengono manipolate partendo dai caratteri all'indirizzo minore, se vale 1 a partire dall'indirizzo maggiore
- IF (Interrupt Flag): se vale 1, i segnali di Interrupt mascherabili vengono percepiti dalla CPU, altrimenti questi vengono ignorati
- TF (Trap Flag): se vale 1, viene eseguita una trap al termine di ogni istruzione.

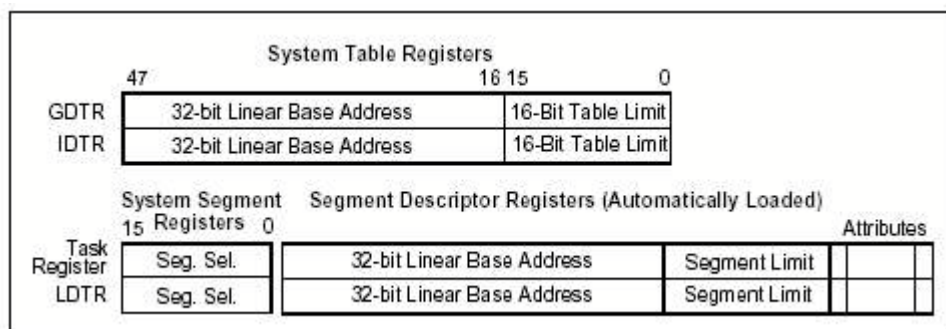
## Registri IA-32 (vedi slide 64)

Ecco il set di registri per l'IA-32. L'immagine seguente riporta i registri a 32 bit che costituiscono un superset dei registri dell'Intel 8086.



Oltre a questi esistono anche:

- Memory Management register: TR, LDTR, IDTR, GDTR



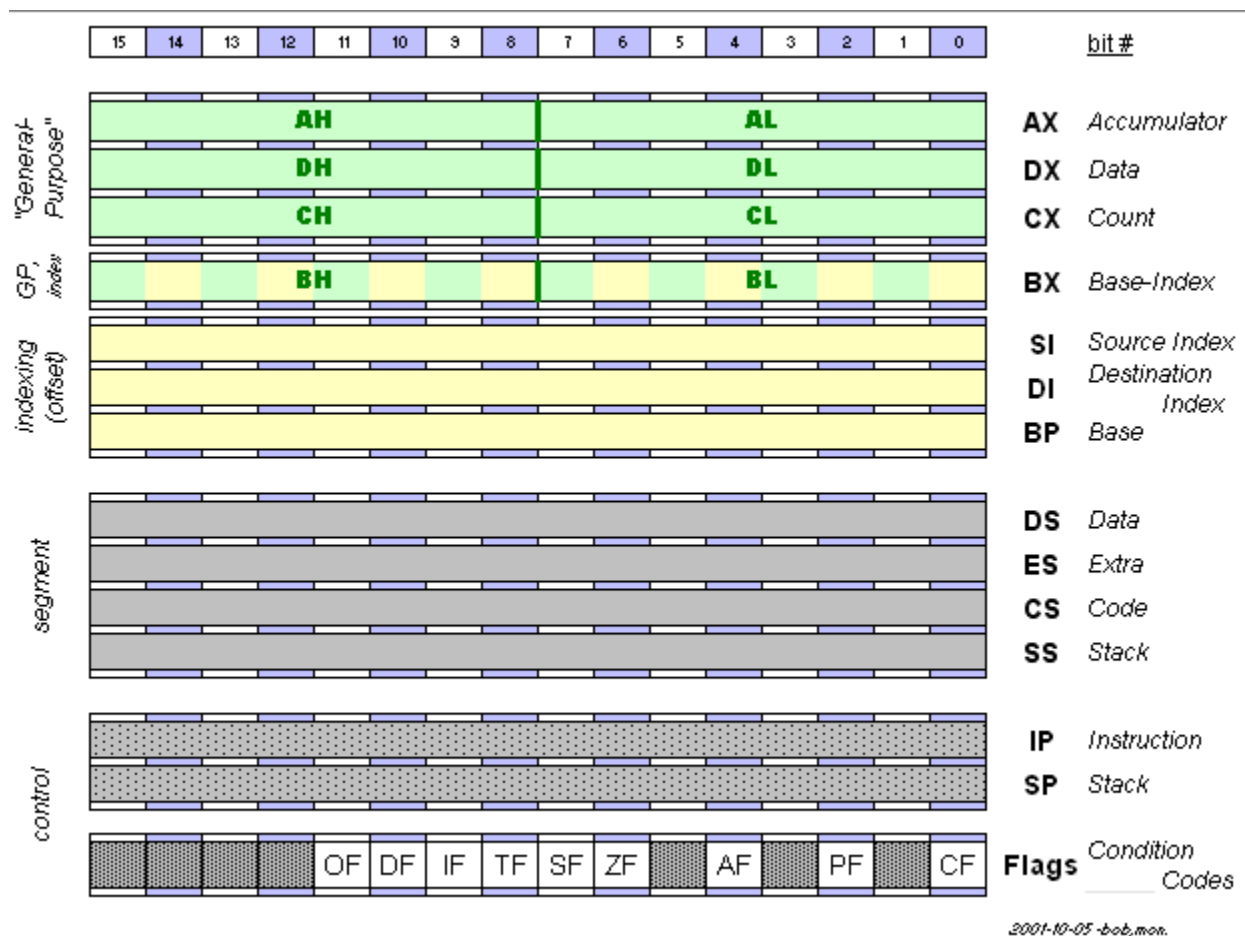
- Control register<sup>a</sup>: CR4, CR3, CR2, CR1, CR0
- Debug register<sup>b</sup>: DR7, DR6, DR5, DR4, DR3, DR2, DR1, DR0
- Test register<sup>c</sup>: TR7, TR6.

<sup>a</sup> [http://en.wikipedia.org/wiki/Control\\_register#Control\\_registers\\_in\\_x86\\_series](http://en.wikipedia.org/wiki/Control_register#Control_registers_in_x86_series)

<sup>b</sup> [http://en.wikipedia.org/wiki/Debug\\_register](http://en.wikipedia.org/wiki/Debug_register)

<sup>c</sup> [http://en.wikipedia.org/wiki/Test\\_register](http://en.wikipedia.org/wiki/Test_register)

Quest'immagine mostra invece il set di registri per l'Intel 8086. Questi registri hanno 16 bit di lunghezza.



## Registri in RM (vedi slide 65)

In modo reale vengono annullati i registri che erano legati al PM, cioè parte dei memory management register, i debug register, i test register e i control register.

## Registri in VM86 (vedi slide 66)

In VM86 vengono annullati i memory management register, i debug register, i control register, i test register e i segment register FS e GS.

Inoltre i restanti registri vengono ridotti a 16 bit in modo da poter emulare l'ambiente DOS utilizzando solo l'istruzione set a 16 bit.

# LEZIONE 18 Settembre 2008

## IA – 32 (IA – 16) IL PROCESSORE PENTIUM

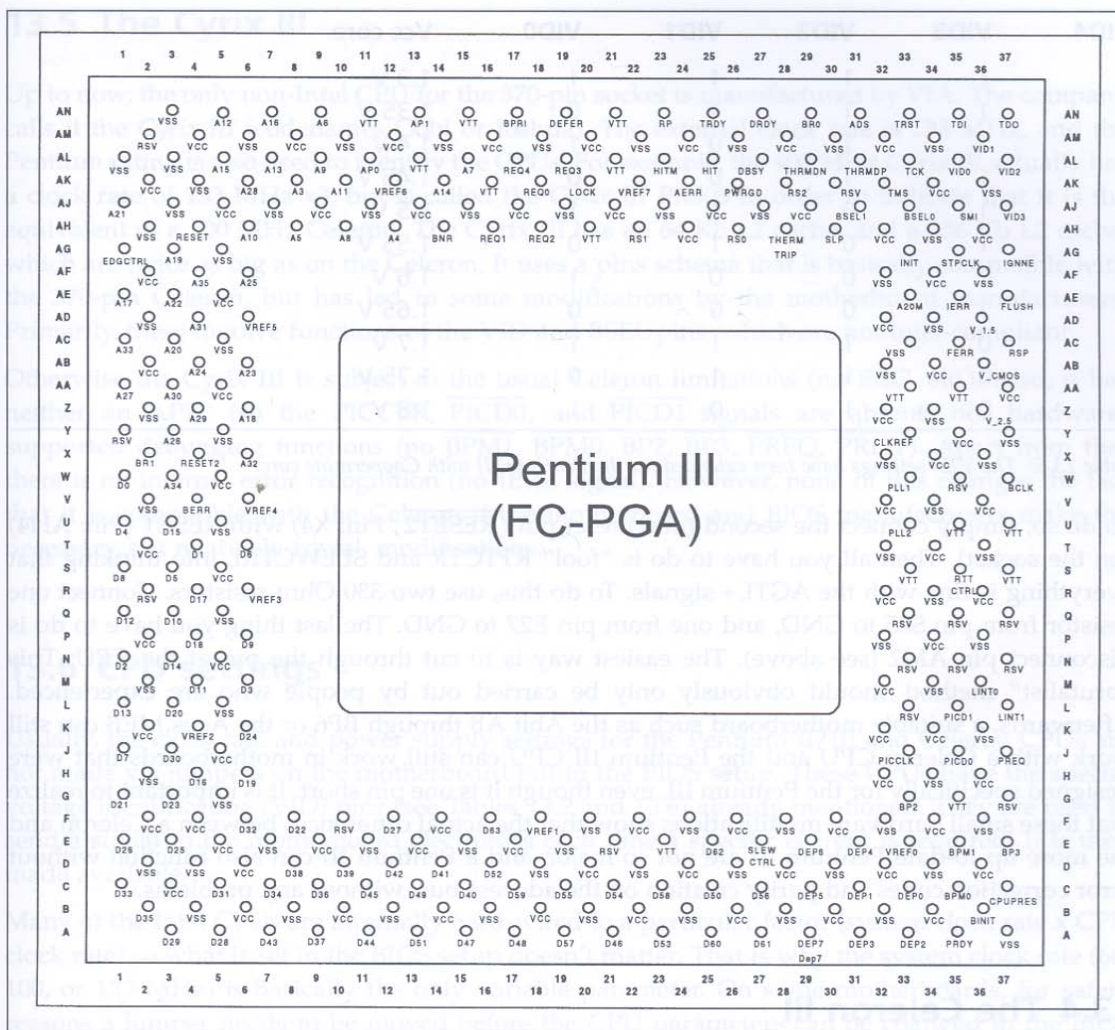
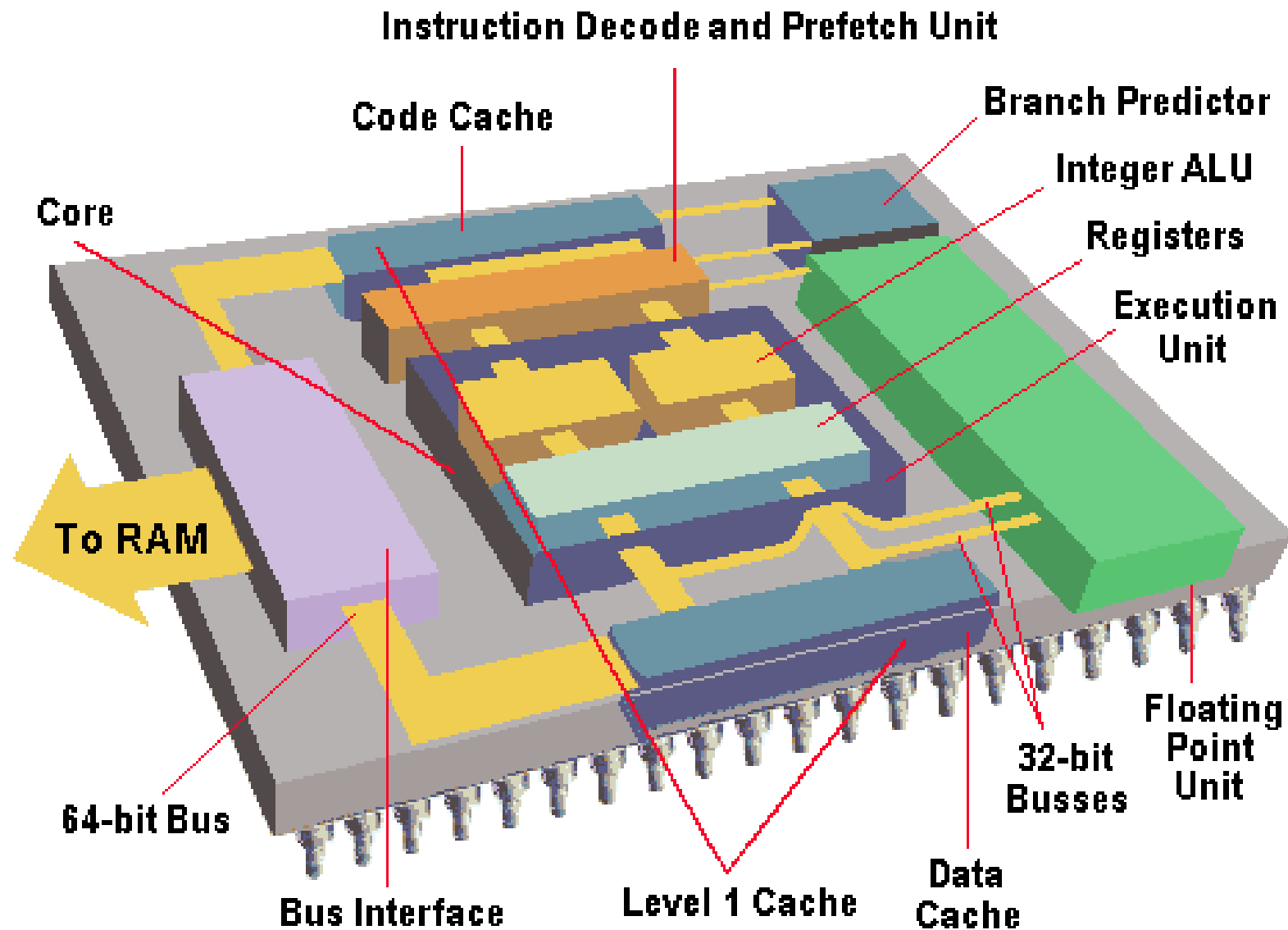


Figure 12.4: Dinout for 270 pin socket version of the Dentium III CDI I





## IA – 32 (IA – 16) IL PROCESSORE PENTIUM

Il Pentium è un processore con architettura nativa a 32 bit (registri dati e indirizzi) esteso nelle ultime versioni a 64 bit di indirizzamento.

Per compatibilità con i precedenti processori intel permette di realizzare anche una architettura a 16 bit.

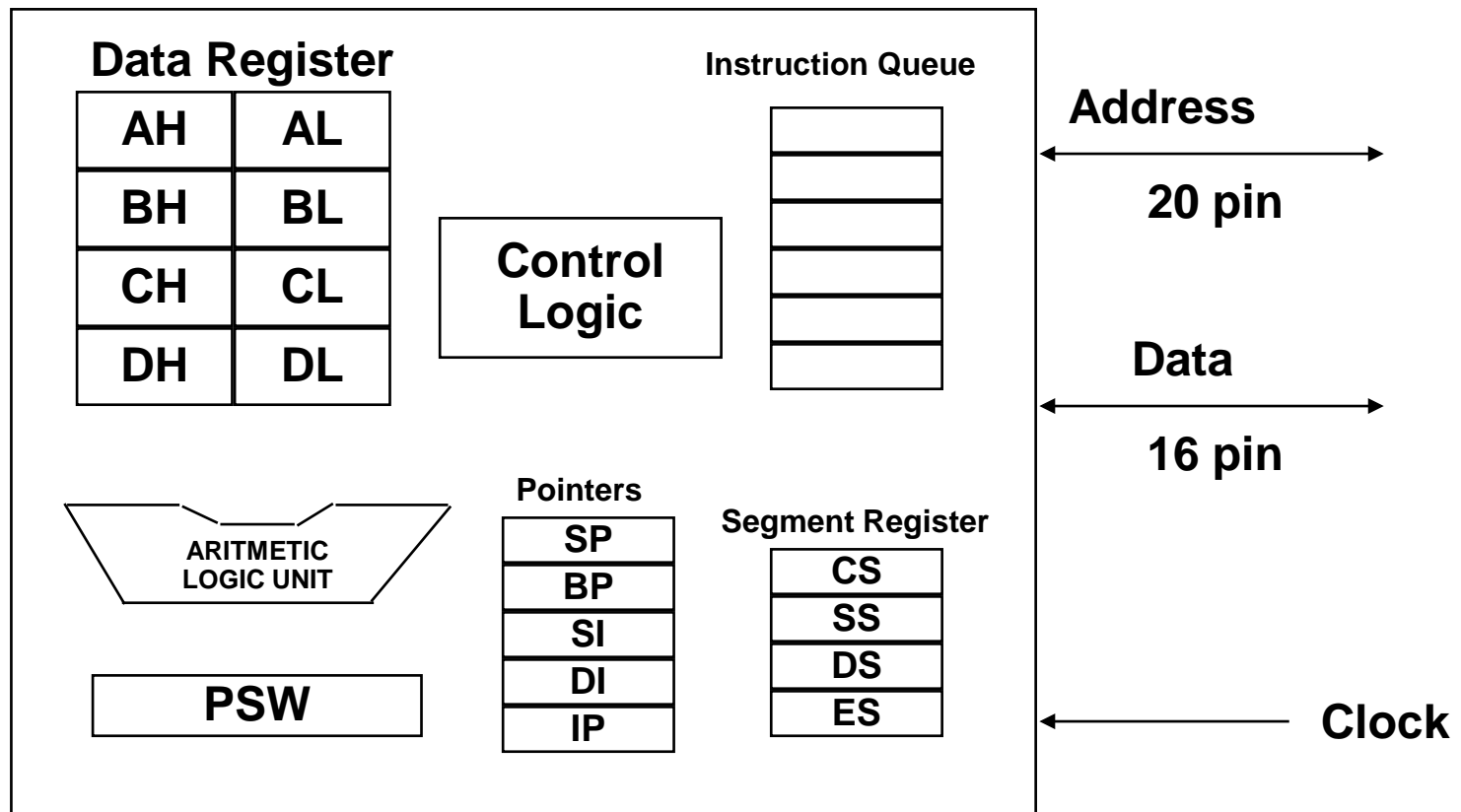
Opera in due modi fondamentali:

1. **Modo reale**, che realizza di fatto l'emulazione hw dello 8086(+), seppur in modo esteso (**avendo la visibilità sui registri a 32/64 bit**); tale modo è attivo al reset ed è pertanto il modo nativo del BIOS nei PC
2. **Modo protetto**, che realizza l'architettura a 32 bit e che è il modo base di Windows MSW e di Linux

(+) in realtà l'emulazione dell'architettura 8086/ambiente DOS, avviene in due modi:

- Modo reale eseguendo solo il set di istruzioni a 16 bit
- In modo protetto mediante l'attivazione di un processo specifico denominato VM86, in cui, quando lanciato il task il sistema opera emulando il comportamento logico dell'ambiente 8086/DS

# Modello Architetture in modo reale (8086)

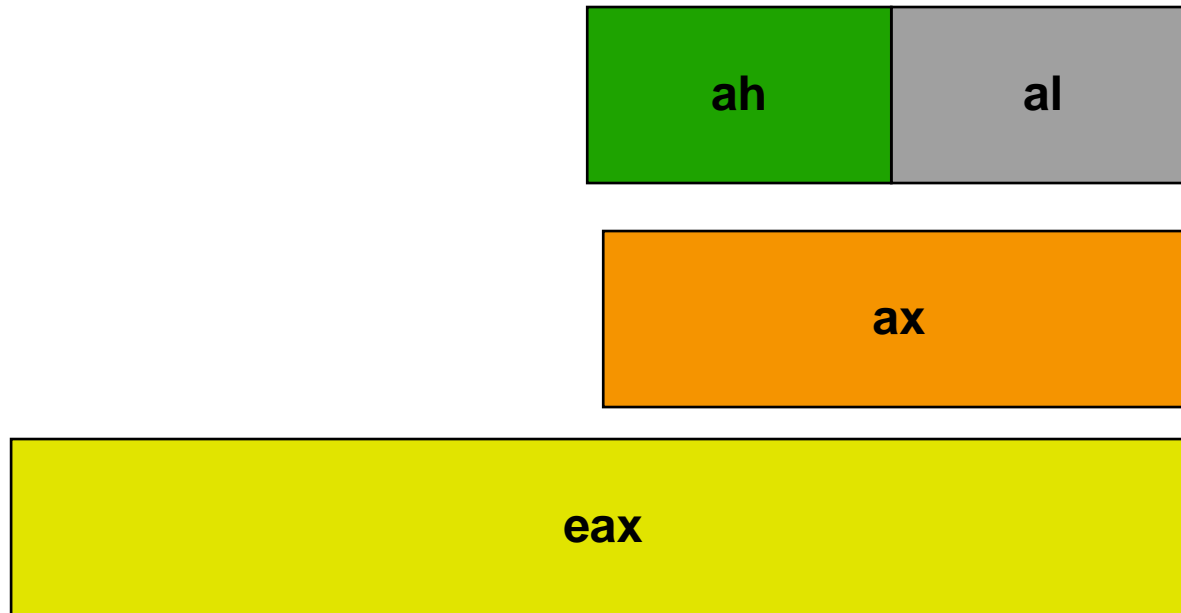


# Registri

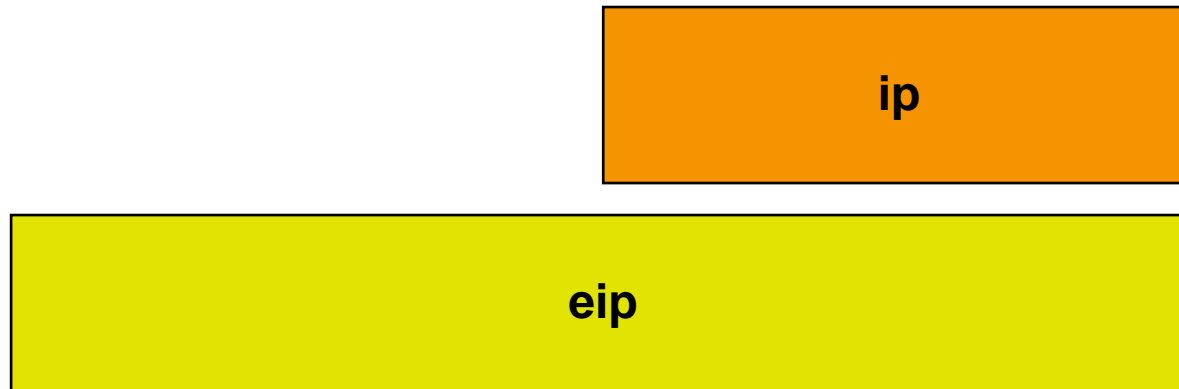
**Possono essere suddivisi in 4 gruppi:**

- **registri di dato**
- **registri puntatore/di indirizzo**
- **registri di segmento**
- **Registri di controllo.**

# I registri dati



# I registri di controllo e di indirizzo



Indirizzamento:  $2^{32} = 4 \text{ GB}$ ;  $2^{16} = 64\text{KB}$

# Registri di Dato

Sono *AX (Accumulator Register)*, *BX (Base Register)*, *CX (Count Register)* e *DX (Data Register)*.

Sono utilizzati per memorizzare operandi e risultato delle operazioni.

Possono essere utilizzati come registri da 16 bit oppure come coppie di registri da 8 bit.

*BX* può anche essere utilizzato nel calcolo di indirizzi.

*CX* viene anche utilizzato come contatore da talune istruzioni.

*DX* contiene l'indirizzo di I/O in alcune istruzioni di I/O.

# Registri Puntatore

Sono IP, SP, BP, SI e DI.

IP (*Instruction Pointer*) contiene il puntatore alla prima istruzione da eseguire. IP non può comparire esplicitamente come operando di una istruzione.

SP (*stack pointer*) contiene il puntatore alla testa dello stack.

BP (*Base Pointer*) viene utilizzato come base per fare accesso all'interno dello stack.

SI (*Source Index*) e DI (*Destination Index*) vengono utilizzati come registri indice.

# Registri di Segmento

**Sono CS, DS, ES e SS.**

**Vengono utilizzati per costruire gli indirizzi fisici con i quali fare accesso in memoria.**

**Contengono i puntatori all'inizio dei segmenti di codice, di dato, di dato supplementare e di stack, rispettivamente.**



# Process Status Word (PSW)

È composta da 16 bit, ma solo 9 di questi sono usati. Ogni bit corrisponde ad un flag. I flag si dividono in:

- flag *di condizione*
- flag *di controllo*.

				OF	DF	IF	TF	SF	ZF		AF		PF		CF
--	--	--	--	----	----	----	----	----	----	--	----	--	----	--	----

# Flag di Condizione

Vengono automaticamente scritti al termine di varie operazioni:

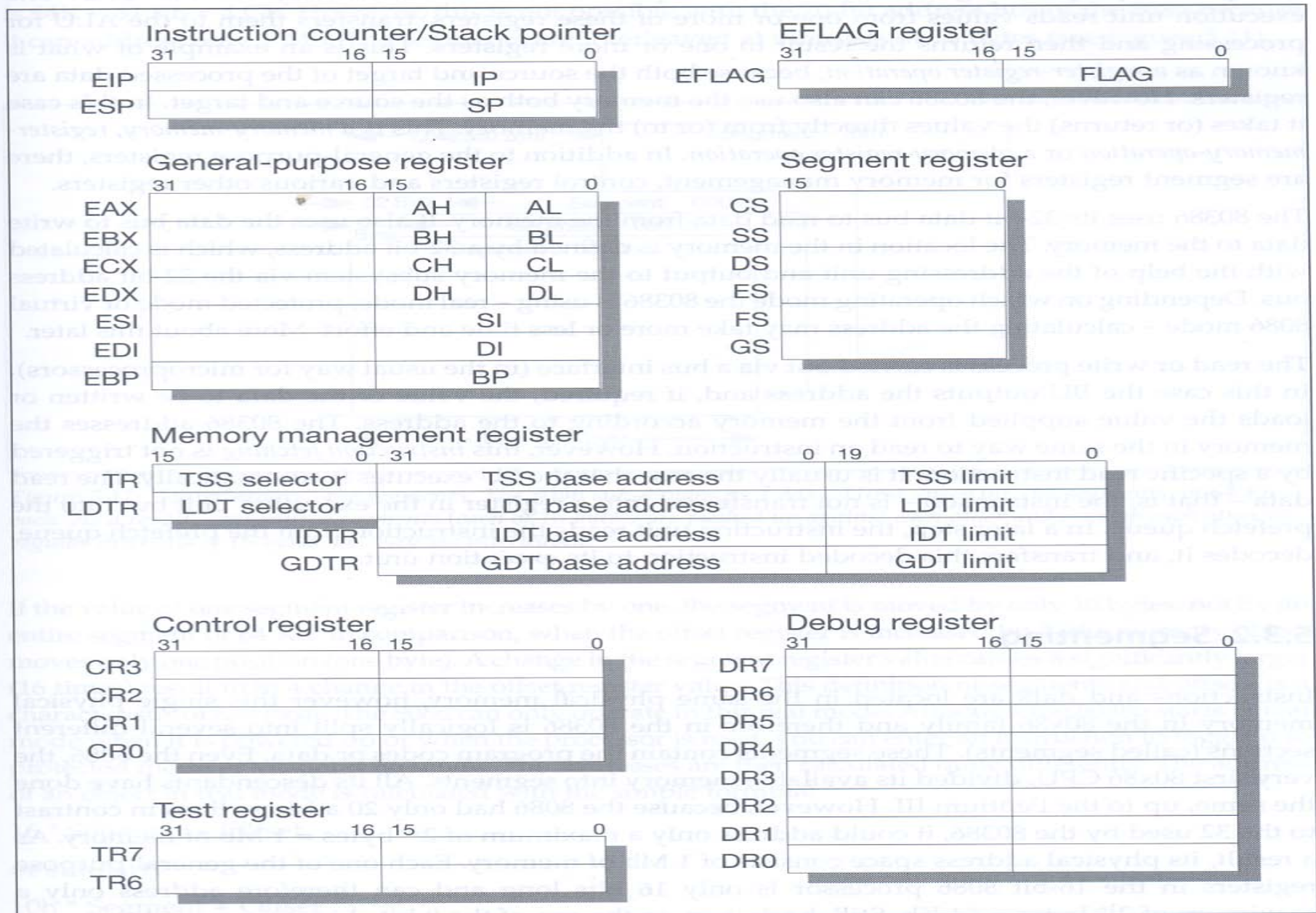
- **SF (*Sign Flag*)**: coincide con il MSB del risultato dopo una operazione aritmetica
- **ZF (*Zero Flag*)**: vale 1 se il risultato è nullo, 0 altrimenti
- **PF (*Parity Flag*)**: vale 1 se il numero di 1 negli 8 bit meno significativi del risultato è pari, 0 altrimenti
- **CF (*Carry Flag*)**: dopo le operazioni aritmetiche vale 1 se c'è stato *riporto* (somma) o *prestito* (sottrazione); altre istruzioni ne fanno un uso particolare
- **AF (*Auxiliary Carry Flag*)**: usato nell'aritmetica BCD; vale 1 se c'è stato *riporto* (somma) o *prestito* (sottrazione) dal bit 3
- **OF (*Overflow Flag*)**: vale 1 se l'ultima istruzione ha prodotto un *overflow*.

# Flag di Controllo

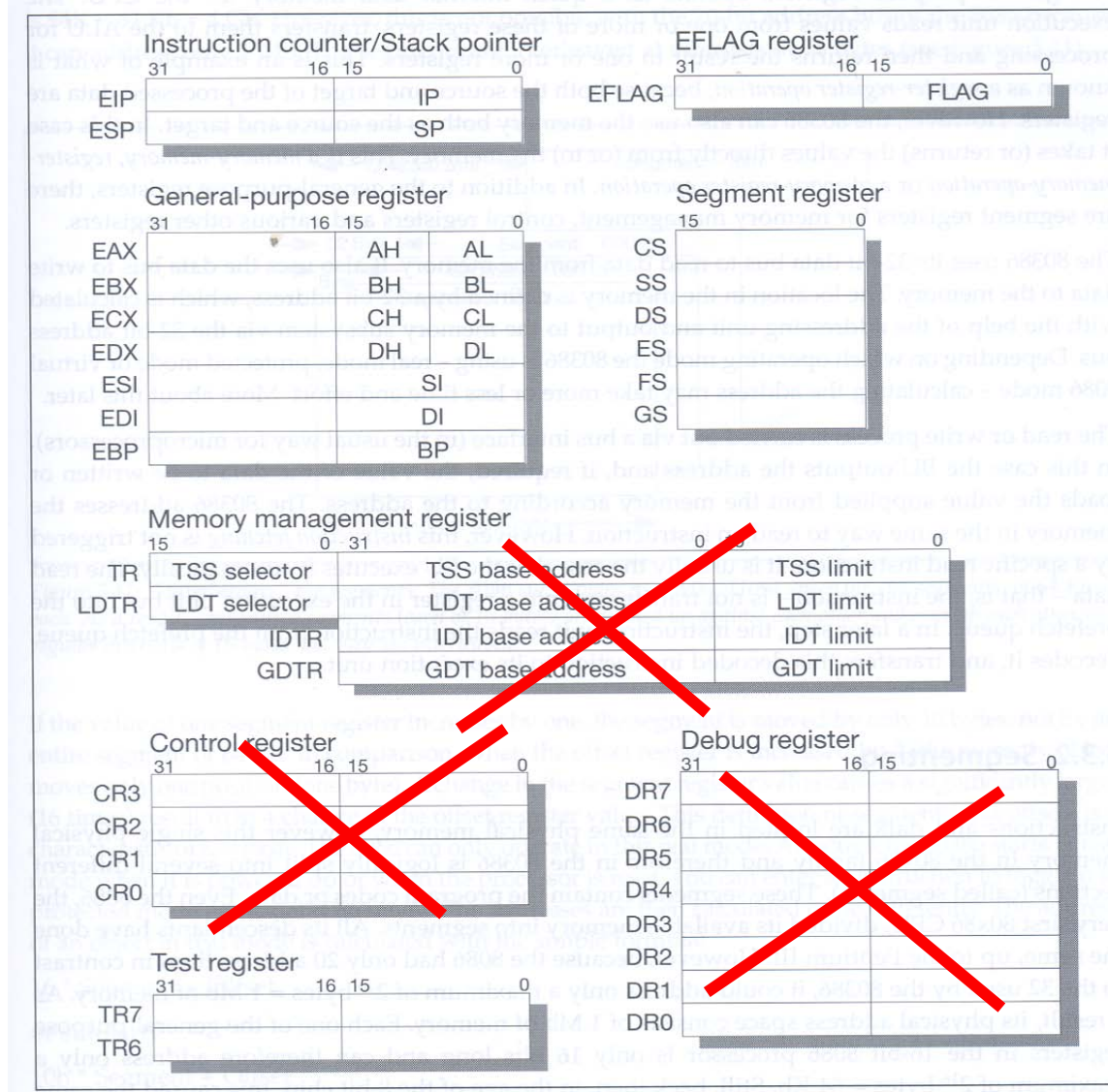
Possono venire scritti e manipolati da apposite istruzioni, e servono a regolare il funzionamento di talune funzioni del processore:

- **DF (*Direction Flag*)**: utilizzato dalle istruzioni per la manipolazione delle stringhe; se vale 0 le stringhe vengono manipolate partendo dai caratteri all'indirizzo minore, se vale 1 a partire dall'indirizzo maggiore
- **IF (*Interrupt Flag*)**: se vale 1, i segnali di Interrupt mascherabili vengono percepiti dalla CPU, altrimenti questi vengono ignorati
- **TF (*Trap Flag*)**: se vale 1, viene eseguita una *trap* al termine di ogni istruzione.

## I REGISTRI IA - 32



# I REGISTRI IN MODO REALE





# I REGISTRI IN MODO VM 86

