

I processori Intel

Tutta la famiglia dei processori Intel (x86) si basa ed e' compatibile con il primo processore di questo tipo: l'**8086**.

L'8086 e' un processore a **16 bit** quindi i suoi registri potranno contenere al massimo 16 bit di informazione.

I processore piu' moderni quali 80486, Pentium e Pentium II sono processori a **32 bit**.

Intel rilascia a meta del 2000 anche il Merced a 64 bit, ma fino ad allora produce solo processori con registri a 32 bit.

Quindi per poter conoscere come funziona un processore Pentium e' necessario partire dal suo bis-bis-nonno: 8086.

La struttura interna dell'8086 si presenta grossomodo cosi:

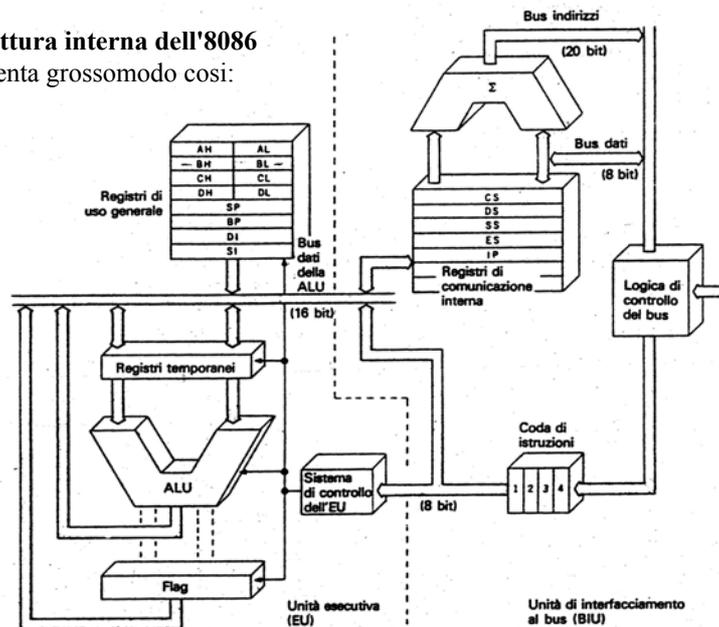
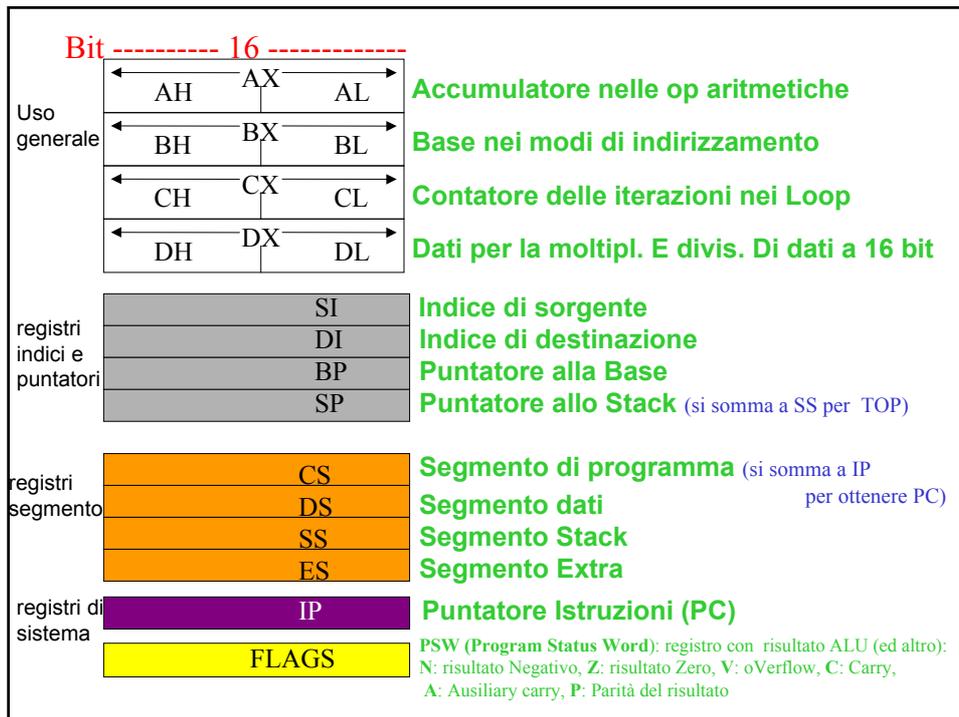


Figura 3.6 8086/8088 con EU e BIU



Microprocessori per il PC

8080 (1974)

- CPU a 8 bit, 6000 tr., 2MHz, 64KB di memoria indirizzabile;
- simile al PDP-8, ma **prodotto di massa**.

8086 (1978) e 8088 (1979)

- CPU a 8 bit, 29000 tr., 5-10MHz, **1MB** memoria indirizzabile;
- bus a 16 bit (8086) o a 8 bit (8088);
- scelta da IBM come CPU per il **PC IBM** originale.

80286 (1982)

- 16 bit, 134'000 tr., 8-12MHz, **16MB** memoria indirizzabile;
- **instruction set** di base **uguale** a **8086** e **8088**;
- organizzazione di memoria differente, ma complessa per garantire la **compatibilità** con i vecchi chip.

I processori Intel a 32 bit

80386 (1985)

- **32 bit**, 275'000 tr., 16-33MHz, **4GB** memoria indirizzabile;
- ~ **compatibile** con quanto prodotto dall'8080 in poi.

80486 (1989)

- 32 bit, 1.2M tr., 25-100MHz, 4GB memoria indirizzabile;
- organizzazione a **pipeline**;
- comprende un'unità **floating point** e **8KB** di **cache** su chip.

L'evoluzione delle CPU Intel: ... dal Pentium a oggi .../1

Pentium (1993-1996), il primo marchio registrato!!

- 3.1M tr., 60-233MHz, processo da 0.8 a 0.35 μ .
- Bus: **32 bit interno** – **64 bit esterno**
- **Cache** di primo livello **integrata** di **8+8 KB**
- **Dual pipelined** superscalar design + dynamic branch prediction

Pentium Pro (1995-1997)

- 5.5M tr. + 15.5M tr. cache II liv., processo 0.5-0.35 μ .
- esegue fino a **cinque istruzioni contemporaneamente**
- dynamic execution (branch prediction, data flow analysis, speculative exec.);
- **superpipeline** (14 stadi) con esecuzione "out-of-order";
- istruzioni CISC tradotte in μ -op **RISC**
- memoria **cache a due livelli**
- nel chip della CPU: **8KB** per istruzioni più **8KB** per dati
- nel package (ma non nel chip): **256 KB** comuni a istruzioni e dati.
- nuova interfaccia (Socket 8 di 387 pin);

ISA del Pentium II

Compatibilità con architetture della famiglia 808x (*real mode* e *virtual 8086 mode*)

Protected mode (vero Pentium II) prevede 4 livelli di privilegi controllati da bit della PSW

- **Livello 0 – kernel mode (sistema operativo)**
- **Livello 3 – user mode (programmi applicativi)**
- **Livelli 1 e 2 intermedi, raramente usati**

Enorme spazio di indirizzamento

- 2^{14} (=16.384) segmenti
- 2^{32} (indirizzi da 0 a $2^{32}-1$) indirizzi per ogni segmento

La maggior parte dei sistemi operativi usa un solo segmento di 2^{32} byte raggruppati in parole da 4 byte

Registri del Pentium II

4 registri da 32 bit di uso generale
(EAX, EBX, ECX, EDX),
ciascuno utilizzabile anche come registri da 16 e 8 bit
(es. AX 16 bit, AH+AL 8 bit)

4 registri da 32 bit usati come puntatori

- ESI – Source
- EDI – Destination
- EBP – Base Pointer (analogo a LV di IJVM)
- ESP – Stack Pointer

6 registri da 16 bit per l'uso dei segmenti
(importanti quando la capacità di indirizzamento era limitata)

1 registro da 32 bit come Program Counter
(detto Instruction Pointer)

1 registro da 32 bit per la PSW

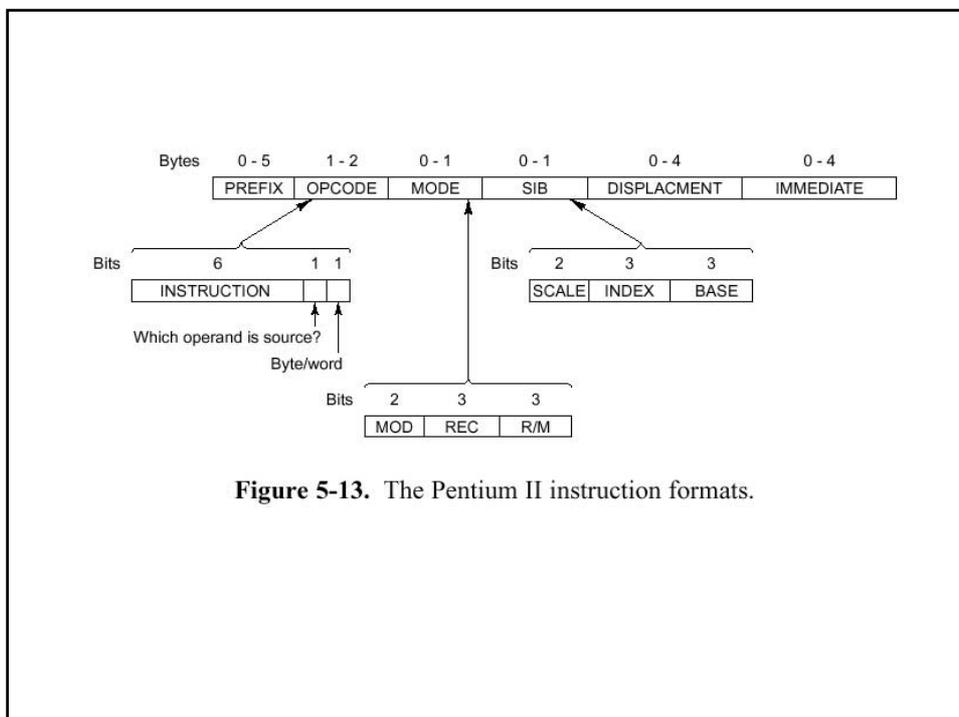
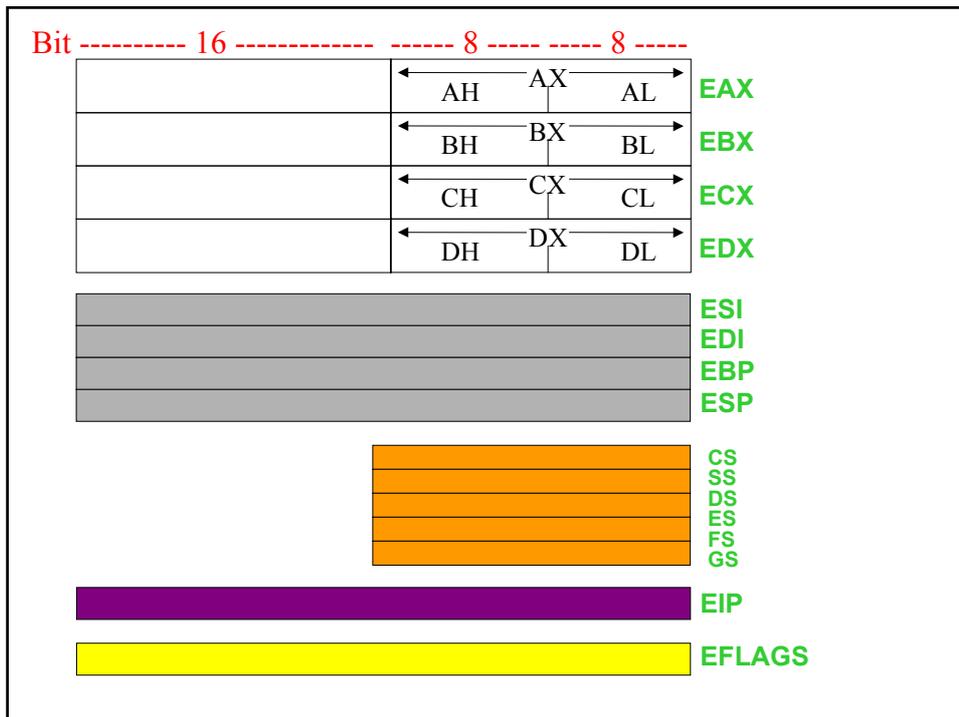


Figure 5-13. The Pentium II instruction formats.

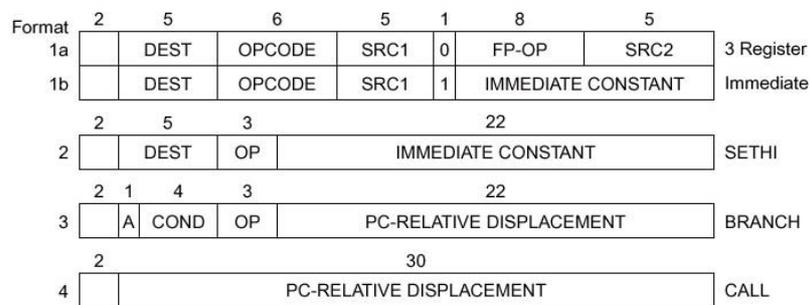


Figure 5-14. The original SPARC instruction formats.

Tipi di istruzioni

Istruzioni su **dati non tipizzati**: determinano operazioni eseguibili su tutti i tipi di dati

- Istr. di trasferimento dati
- Istr. di traslazione e rotazione
- Istr. di I/O

– Istruzioni su **dati tipizzati**: operano su un tipo di dati

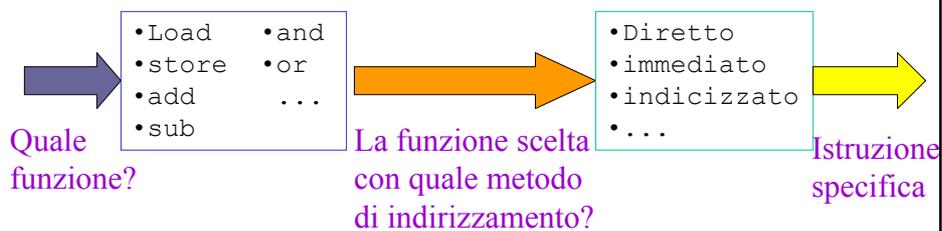
- Istr. aritmetiche
- Istr. logiche
- Istr. di elaborazione stringhe
- Istr. di elaborazione indirizzi
- Istr. di modifica della sequenza di elaborazione

Tipi di istruzioni

Suddivisione funzionale delle istruzioni:

- trasferimento dati
 - shift e rotazione
 - aritmetiche
 - interi binari
 - reali
 - decimali
 - vettori booleani
 - elaborazione indirizzi
 - elaborazione di stringhe
 - modifica sequenza
 - salti condiz. e incondizion.
 - iterazione o loop
 - salti a subroutine
 - salti a procedure
-
- istruzioni su dati tipizzati
 - istruzioni su dati non tipizzati
 - istruzioni general-purpose
 - istruzioni special-purpose

Costruzione di una istruzione



1. Trasferimento Dati

- Permettono il trasferimento di informazione **da** una locazione di memoria (*sorgente*) **ad** un'altra (*destinazione*)
- Effettuano in realtà una duplicazione dei dati
- **LOAD** da locazione di memoria a registro
- **STORE** da registro a locazione di memoria
- **MOVE** da locazione ad altra locazione
- **SWAP** scambio tra due locazioni (XCHG)
- **I/O**
 - IN** da PORTA a locazione
 - OUT** da locazione a PORTA
 - INVECT** da PORTA a locazione contigua
 - OUTVECT** da locazione contigua a PORTA

ESEMPI:

INTEL 8088

- MOV AH,0
- XCHG AL,AH
- LEA
- MOV AX, DX

Ultra SPARC II

- LDSBADDR, DST
- LDUBADDR, DST
- LDSHADDR, DST
- LDSWADDR, DST

8088 - Pentium II trasferimento dati

Sono le istruzioni usate + frequentemente.
Permettono di spostare dati fra registri e fra registro e memoria, ma non fra memoria e memoria.

Le più comuni sono:

- **MOV DST, SRC** copia SRC in DST
- **PUSH SRC** push da SRC a stack
- **POP DST** pop da stack a DST
- **XCHG DS1, DS2** scambia DS1 e DS2
- **LEA DST, SRC** carica indirizzo di SRC in DST
- **IN** da PORTA a locazione
- **OUT** da locazione a PORTA

MOV DST, SRC

copia SRC in DST

I tipi di OPERANDI sono:

- Memoria / Accumulatore
- Accumulatore / Memoria
- Registro Segmento / Memoria o Registro Segmento / Registro
- Memoria / Registro Segmento o Registro / Registro Segmento
- Registro / Registro o Registro/Memoria o Memoria / Registro
- Registro / Valore
- Memoria / Valore

Sorgente Destinazione

7 tipi di istruzione MOV a seconda del tipo di destinazione e di origine

ESEMPIO

- MOV ALFA, AX
- MOV AX, DS: dato
- MOV DS, AX
- MOV CX, DX
- MOV AX, VET[SI]
- MOV VET[DI], DX
- MOV AX, 10
- MOV VET[DI], 100

Se la destinazione è un registro, il bit 'd' dell'OPCODE vale 1

Se il tipo è una parola, il bit 'w' dell'OPCODE vale 1

XCHG DS1, DS2

scambia DS1 con DS2

ESEMPIO

- XCHG AX, BX
- XCHG CX, AX
- XCHG BL, AL
- XCHG BX, ALFA
- XCHG ALFA, CX

I registri segmento non possono essere operandi della XCHG

Esistono due tipi di XCHG a seconda del tipo di operandi:

- registro/accumulatore
- memoria/registro o registro/registro

PUSH SRC

decrementa SP di 2 e trasferisce una parola da SRC alla locazione indicata da SP

ESEMPIO

- PUSH AX
- PUSH ES
- PUSH VET[DI]
- PUSH BETA

3 tipi di PUSH a seconda del tipo di origine:
registro, registro segmento, memoria

POP DST

trasferisce una parola dallo stack a DST e incrementa SP di 2

ESEMPIO

- POP CX
- POP DS
- POP ALFA[BX]
- POP ALFA

3 tipi di POP a seconda del tipo di destinazione

Istruzioni di I/O

Tali istruzioni sono legate alla modalità di indirizzamento dei dispositivi di I/O adottata nel sistema

- ***Memory-Mapped I/O*** : lo spazio di indirizzamento è condiviso dalla memoria e dai dispositivi di I/O ai quali è riservato un sottoinsieme degli indirizzi di memoria (non esistono istruzioni di I/O)
- ***Istruzioni di I/O*** : istruzioni speciali per i trasferimenti di I/O che consentano l'accesso ai registri dati dei dispositivi periferici

IN trasferisce un BYTE o una parola da una PORTA di INPUT al registro AL (o AX)

IN AX, PORTAW
IN AL, PORTAB
IN AX, DX

OUT trasferisce un BYTE o una parola dal registro AL (o AX) ad una PORTA di OUTPUT

OUT PORTAW, AX
OUT PORTAB, AL
OUT DX, AX

La porta può essere indicata da un valore fisso (tra 0 e 255) o variabile, contenuto nel registro DX, che permette di accedere a 64K porte.

Istruzioni di traslazione (SHIFT) e rotazione

- Istruzioni *monadiche* (istruzioni che hanno un operando e producono un risultato un operando)
- **Traslazione (SHIFT)** : operazione in cui i bit vengono spostati a destra o a sinistra ed i bit che sono posti alla fine del byte o della parola vengono persi. Trasferimento di ogni bit di una word nella posizione successiva (a dx o a sx)
 - **Logical SHIFT**: inserimento di zeri nelle posizioni vuote
 - **Arithmetic SHIFT**: replica del bit del segno (right) o test del bit del segno (left)
- **Rotazione (ROTATE)**: shift a destra o a sinistra con reinserimento del bit espulso

Istruzioni di traslazione (SHIFT) e rotazione

In generale queste istruzioni sono utili per isolare ed accedere ad unità di informazione inferiori alla minima unità indirizzabile (byte o word).

Lo shift può essere usato anche per effettuare operazioni aritmetiche di moltiplicazione e divisione per potenze di 2 su interi positivi:

- shift a sx di k bit di A $\Rightarrow A \cdot 2^k$
- shift a dx di k bit di A $\Rightarrow A : 2^k$

SHL/SHR DST, #

shift logico di DST sx/dx di # bit

inserendo quindi degli 0 nei bit di ordine superiore spostati

SHR AH,1

SHR AX,CL

SHR VARIABILE,3

SAL/SAR DST, #

shift aritmetico di DST sx/dx di # bit

ROL/ROR DST, #

rotazione di DST sx/dx di # bit

ROR AX,1

ROR DATO,2

ROL VET[BX], CL

Istruzioni logiche o Booleane

- Istruzioni *diadiche* che consentono le operazioni logiche AND, OR, NOT, XOR, Nand, NOR
- Operano su **vettori booleani**, aggregati (word o byte) di variabili logiche generalmente non omogenee
- Operano in parallelo su tutti gli elementi del vettore booleano ('bit a bit')

NOT DST rimpiazza DST col Complemento a 1 di DST

ESEMPIO

- NOT VARIABILE

AND DST, SRC

esegue la congiunzione bit per bit dei due operandi e pone il risultato in DST

ESEMPIO

- AND AX, BX
- AND DX, VET[BX]
- AND VARIABILE, AL
- AND AH, 0EH
- AND VARIABILE, 46H

Esistono 3 tipi di AND a seconda degli operandi:

- registro/registro o registro/memoria o memoria/registro
- accumulatore/valore
- memoria/valore o registro/valore

Ad es, AND usato per l'estrazione di bit a parole:

- 32 bit parola di memoria che contiene 4 caratteri rappresentati a 8 bit
- ad es. separare il 2 carattere dagli altri 3 per stamparlo creando una parola che contenga quel carattere negli 8 bit + a dx e 0 nei 24 bit + a sx
- mettiamo in AND la parola di origine con una costante detta **maschera**.
- il risultato e' che i bit indesiderati vengono tutti trasformati in 0.

OR DST, SRC

esegue la disgiunzione bit per bit dei due operandi e pone il risultato in DST

ESEMPIO

- OR AH, BL
- OR VET[DI], BX
- OR AX, DATO_Word
- OR AL, 0EH
- OR DATO_byte, 3DH

Esistono 3 tipi di OR a seconda degli operandi:

- registro/registo o registro/memoria o memoria/registo
- accumulatore/valore
- memoria/valore o registro/valore

Ad es, OR usato per aggiungere bit a parole:

- 32 bit parola di memoria che contiene 4 caratteri rappresentati a 8 bit
- ad es. cambiare gli 8 bit + a dx senza disturbare i 24 bit rimanenti
- prima devo mascherare gli 8 bit indesiderati e poi devo aggiungere con un OR il carattere desiderato.