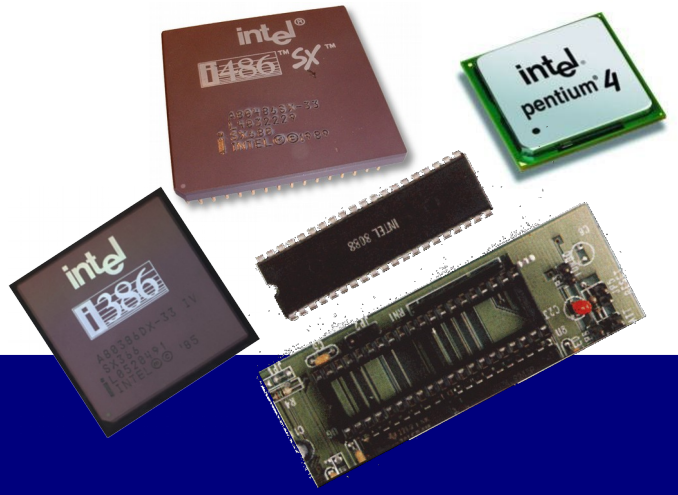


L'Architettura Intel

Architettura dei Calcolatori Elettronici

Un po' di storia...

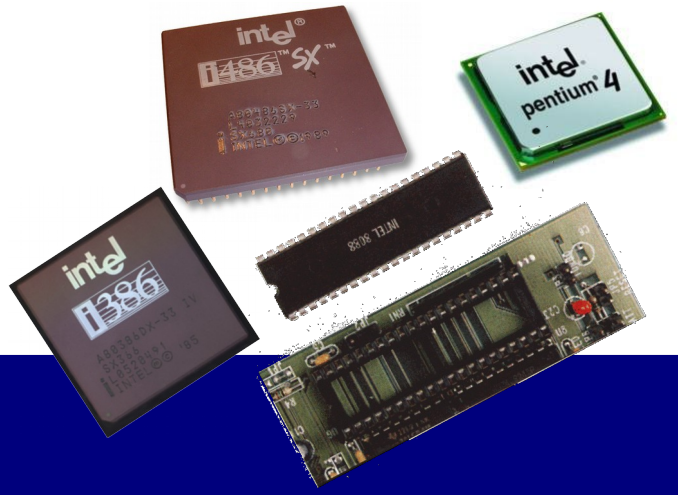
- 1978 - INTEL 8086
  - microprocessore a **16 bit**
  - **1 MB (2^{20}B)** di memoria indirizzabile
  - segmentazione della memoria
- 1979 - INTEL 8088
  - bus dati esterno a **8 bit**
- 1982 - INTEL 80286
  - microprocessore a **16 bit**
  - **16 MB (2^{24}B)** di memoria indirizzabile
  - memoria virtuale
  - protected mode (non è possibile accedere a locazioni di memoria gestite da altri programmi)
- 1985 - INTEL 80386
  - microprocessore a **32 bit → 4 GB (2^{32}B)** di memoria indirizzabile
  - paginazione
  - supporto al multitasking

Un po' di storia...

- 1991 INTEL 80486
  - microprocessore a **32 bit**
  - pipeline singola
  - coprocessore matematico integrato
- 1993 - Pentium
  - microprocessore a **32 bit**
  - introduzione di tecniche superscalari (esecuzione parallela di due istruzioni)
  - cache primaria e secondaria con bus dedicato
- 1995 - Pentium Pro
  - microprocessore a **32 bit**
  - miglioramento delle tecniche superscalari (es.predizione dei salti)
- 1997 - Pentium II
  - microprocessori a **32 bit**
  - incorpora nuove istruzioni in virgola mobile per elaborare in modo efficiente di video, audio e grafici.

Un po' di storia...

- 1999 - Pentium III
  - microprocessore a **32 bit**
  - incorpora nuove istruzioni per supportare il software grafico in 3D
  - cache secondaria e CPU nello stesso integrato
- 2000 - Pentium 4
  - microprocessore a **32 bit**
  - Ulteriori istruzioni in virgola mobile
  - Miglioramenti per la multimedialità
  - Hyper-threading
- 2001 - Itanium
  - microprocessore a **64 bit**
  - architettura IA-64
- 2002 – Itanium2
  - microprocessore a **64 bit**
  - architettura IA-64
  - miglioramento del precedente modello

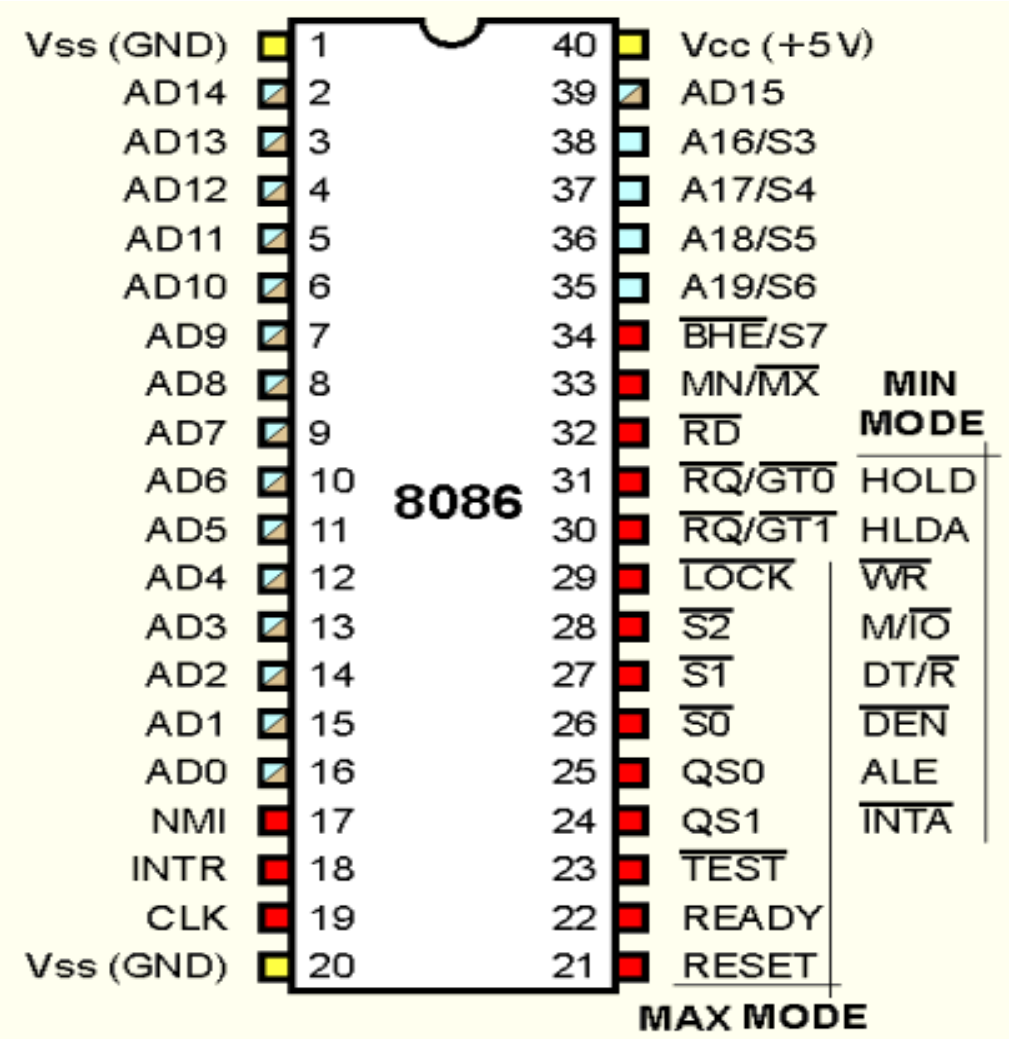


Il Microprocessore Intel 8086

Architettura dei Calcolatori Elettronici

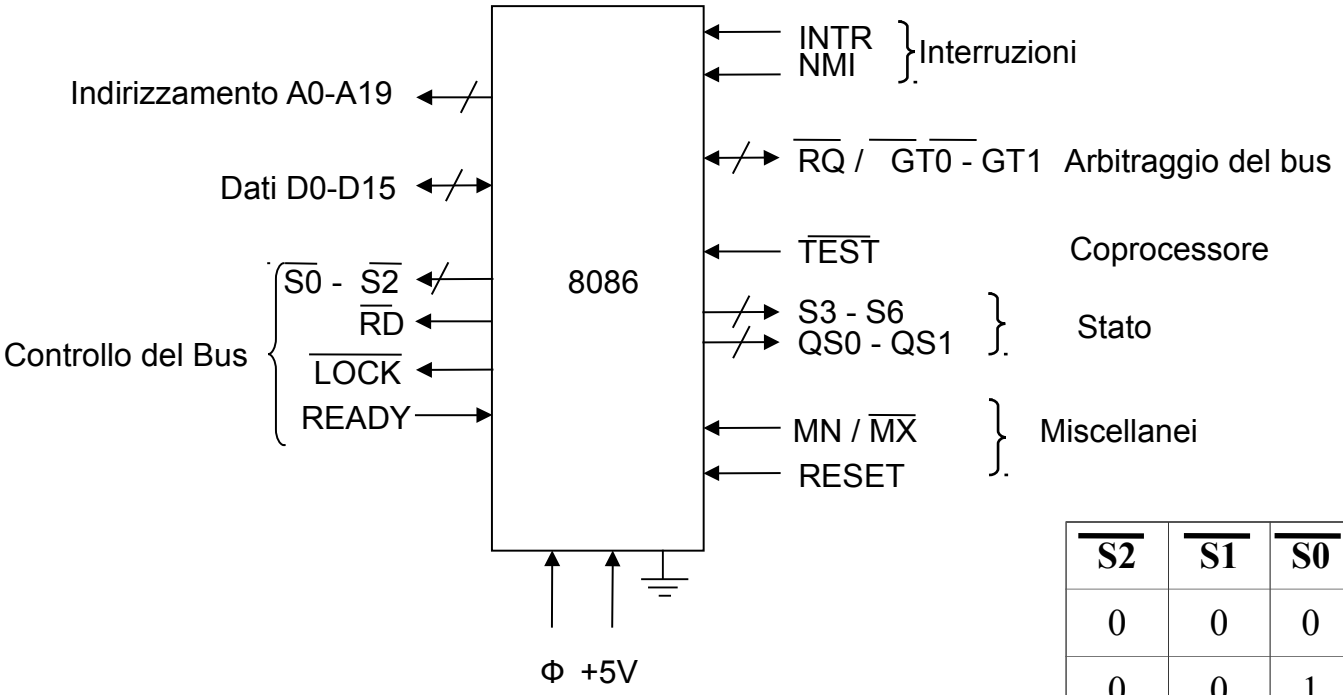
A cura di:
ing. Rosa Anna Micillo

Il pinout fisico



- microprocessore a 16 bit
- 40 piedini
- massima frequenza di clock (1°versione) 5MHz
- numero medio di cicli di clock per istruzione - CPI 15
- $t_{esec} = 15/5MHz = 3\mu s$
  → **0,33 MIPS**

Il pinout logico



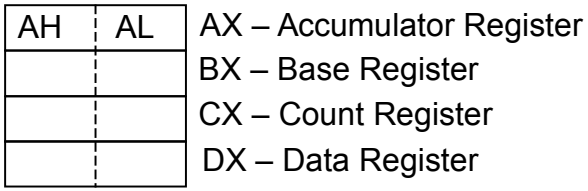
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	**Tipo di Cilco di Bus**
0	0	0	Conferma di interruzione
0	0	1	Lettura da porta I/O
0	1	0	Scrittura da porta I/O
0	1	1	Stop
1	0	0	Acceso del codice
1	0	1	lettura di memoria
1	1	0	scrittura di memoria
1	1	1	bus libero

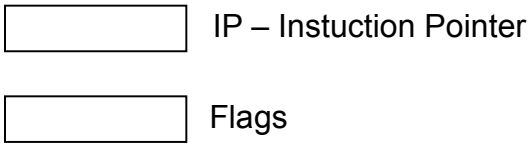
Il modello di programmazione

Parallelismo interno di 16 bit

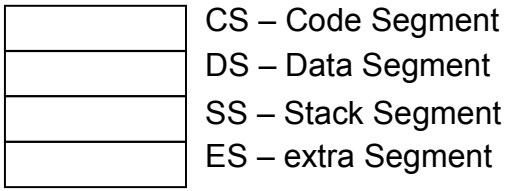
- 4 registri dati utilizzabili anche come registri a 8 bit
- 4 registri puntatori
- 4 registri di segmento
- registro IP
- registro di stato

registri dati

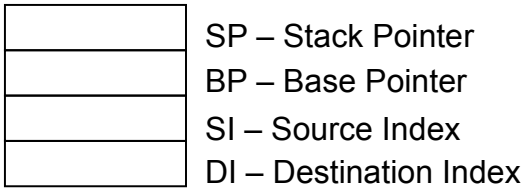




registri di segmento

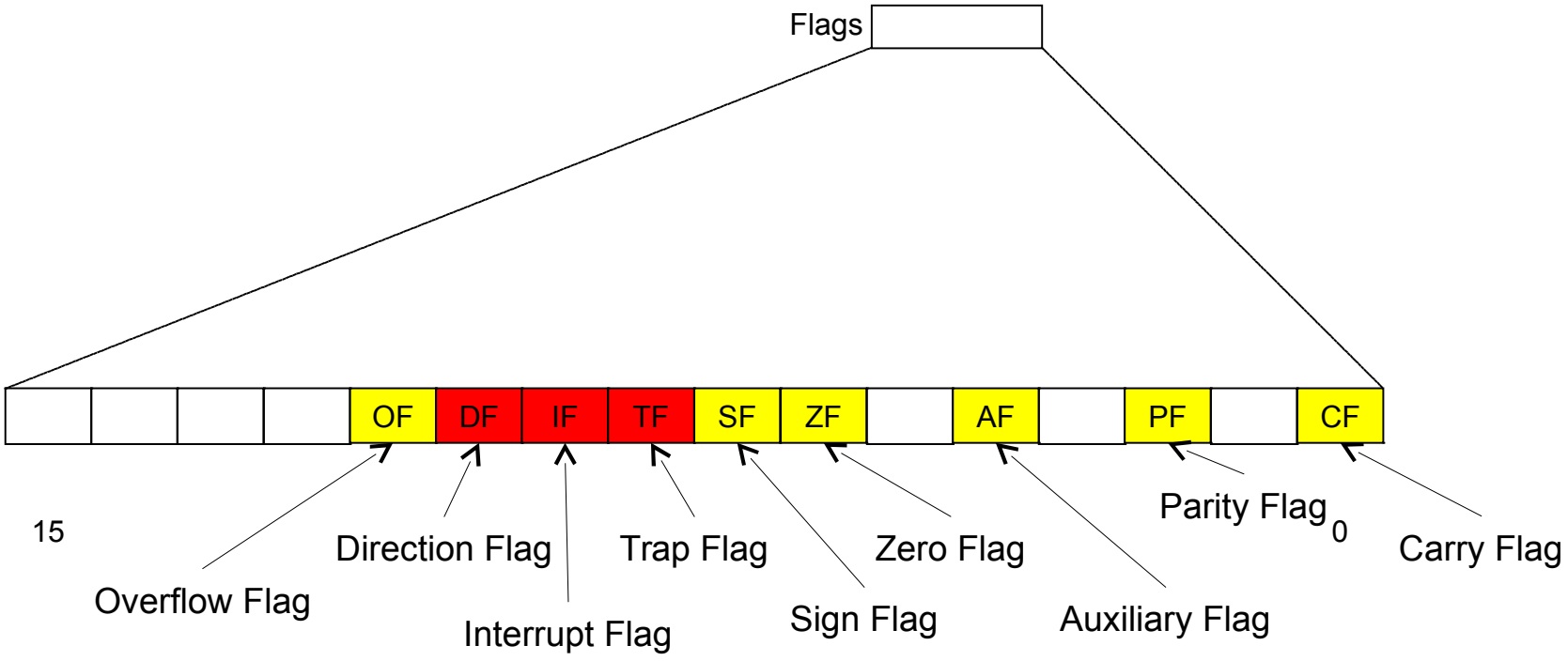


puntatori



Il modello di programmazione

- registro di stato
  - 3 flags di controllo
  - 6 flags di stato



Le istruzioni

- 100 istruzioni assembler
  - a ciascuna istruzione assembler posso corrispondere più istruzioni macchina
    - es. **MOV** corrispondono 28 istruzioni macchina
  - ciascuna istruzione assembler ammette solo alcune modalità di indirizzamento
  - alcune istruzioni possono utilizzare solo determinati registri
- Assemblatore trasforma ciascuna istruzione assembler nel codice opportuno

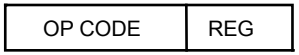
Le istruzioni

- Processore CISC (*Complex Instruction Set Computer*)
- Struttura dei codici e dei formati complicata
  - Lunghezza variabile: da 1 a 6 Byte
  - Lunghezza dipende
    - dalla istruzione
    - dalla modalità di indirizzamento
    - dalla grandezza (in bit) di ciascun operando
- Fase di Decodifica dell'istruzione complessa

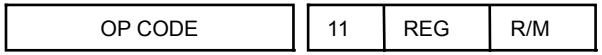
Possibili formati delle istruzioni

Operandi impliciti

Modalità Registri

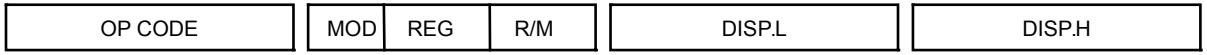


Registro-Registro

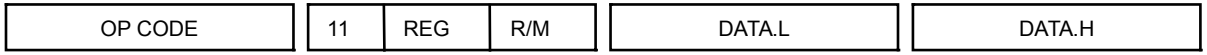


Registro a/da memoria senza scostamento

Registro a/da memoria con scostamento (se è usato un dato a 16 bit)



Operando immediato a registro (se è usato un dato a 16 bit)



Operando immediato a memoria con scostamento a 16 bit (se è usato un dato a 16 bit)

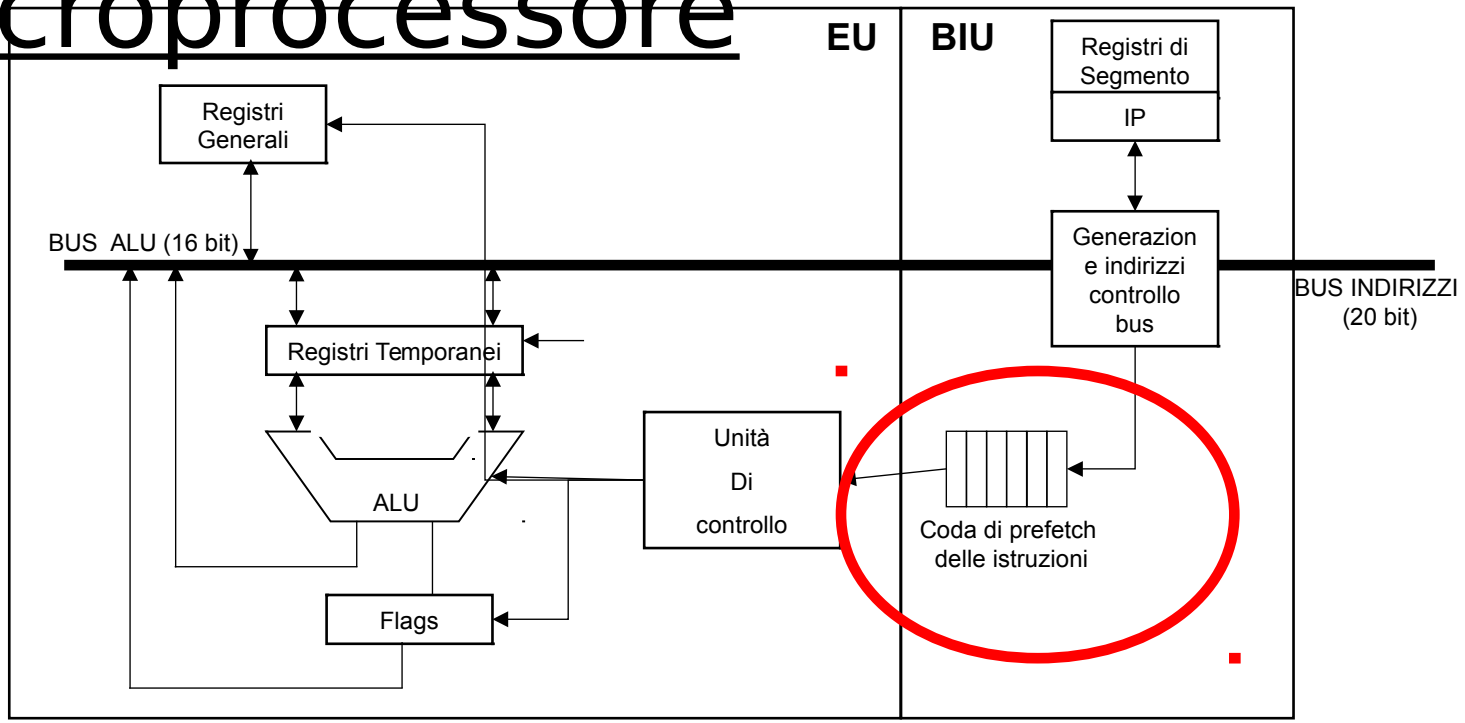
LEGENDA

MOD – modo
REG – registro
R/M – registro o mem
DATA – dato immedia
DISP – scostamento
.L – parte bassa
.H – parte alta

Possibili formati delle istruzioni

- Oltre al byte contenente il codice dell'operazione si può avere:
  - Nessun byte aggiuntivo
  - 2 byte di EA (Effective Address)
  - 1 oppure 2 byte per
    - Operando immediato
    - Scostamento
    - Scostamento e operando immediato di 1 oppure 2 byte
  - 2 byte per lo scostamento e 2 byte per l'indirizzo base del segmento

Struttura interna del microprocessore



Suddivisa in due parti che operano in parallelo (salvo le dovute sincronizzazioni):

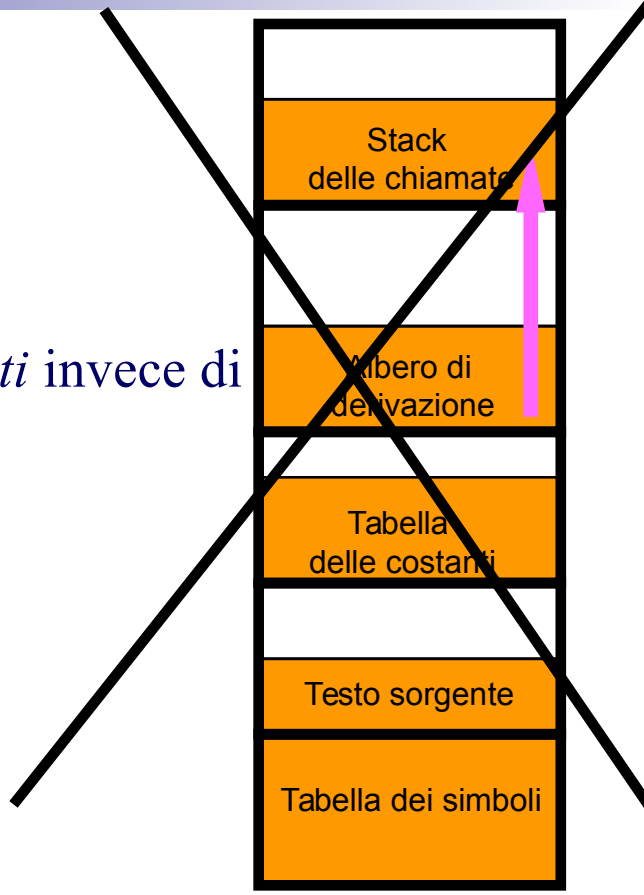
- ***Execution Unit*** per la decodifica e l'esecuzione delle operazioni
- ***Bus Interface Unit*** per creare gli indirizzi fisici ed accedere alla memoria

Pipeling Elementare

Segmentazione

- Esempio: compilatore
- visione della memoria classica monodimensionale
- Soluzione: avere più spazi di indirizzamento *separati* invece di averne uno solo.

visione della memoria *bidimensionale*



Segmentazione

- ***...segmento***
  - unità logica nota al programmatore
  - sequenza lineare di locazioni di memoria comprese da 0 a MAX → segmenti diversi possono avere lunghezze diverse
  - la lunghezza può variare durante l'esecuzione
- i segmenti possono crescere/decrescere indipendentemente dagli altri
- indirizzo è formato da due parti
  - numero del segmento
  - indirizzo all'interno del segmento
- alcuni vantaggi
  - facilita la condivisione di dati e procedure tra i processi
  - semplifica la gestione delle strutture dati che crescono/decrescono velocemente

Organizzazione della memoria

- 1 **MB** di memoria indirizzabile logicamente suddiviso in ***segmenti***

- I segmenti …
  - hanno una dimensione massima pari a **64 KB** (2^{16} Byte)
  - devono iniziare in una locazione della memoria fisica il cui indirizzo sia multiplo di 16
  - possono essere adiacenti, disgiunti, parzialmente o totalmente sovrapposti
    → una locazione di memoria può appartenere anche a più segmenti

- Obiettivo: indirizzare uno spazio di memoria di **1 MB** utilizzando registri di segmento di 16 bit (anziché 20)
  → risparmio di silicio

- I registri segmento identificano i segmenti correnti di codice (CS), di stack (SS) di dati (DS) e l'extra segmento corrente (ES)

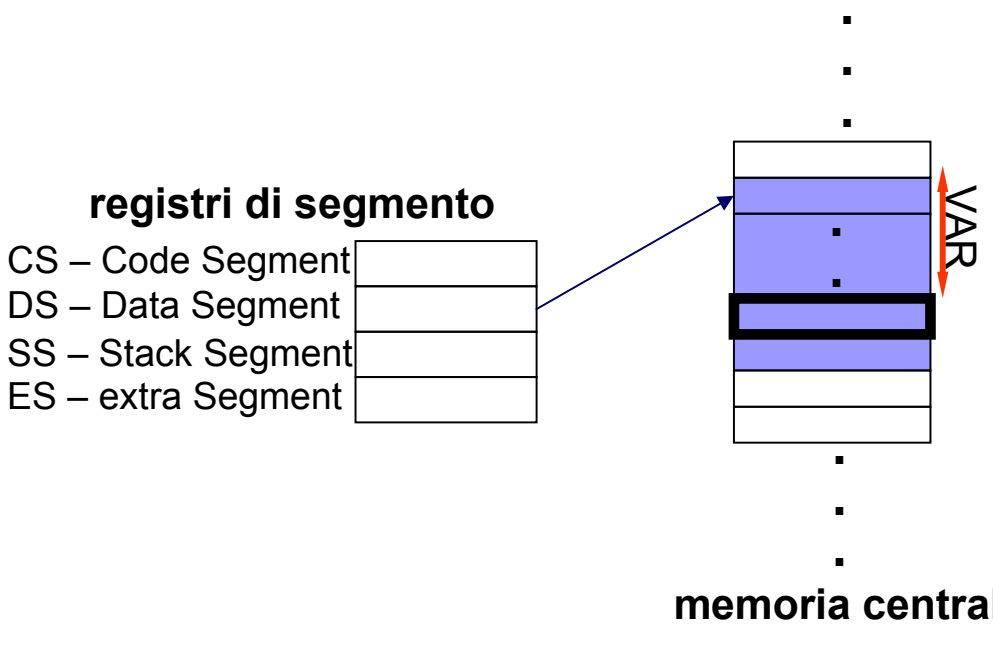
Uso dei registri di segmento

- Sono utilizzati per ricostruire l'indirizzo fisico del "dato"

- Esempio

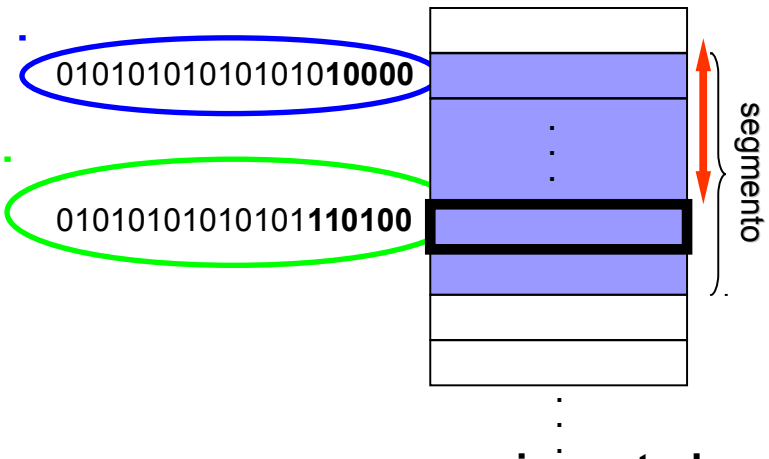
  ADD AX, VAR AX ← M[DS:offset(VAR)]

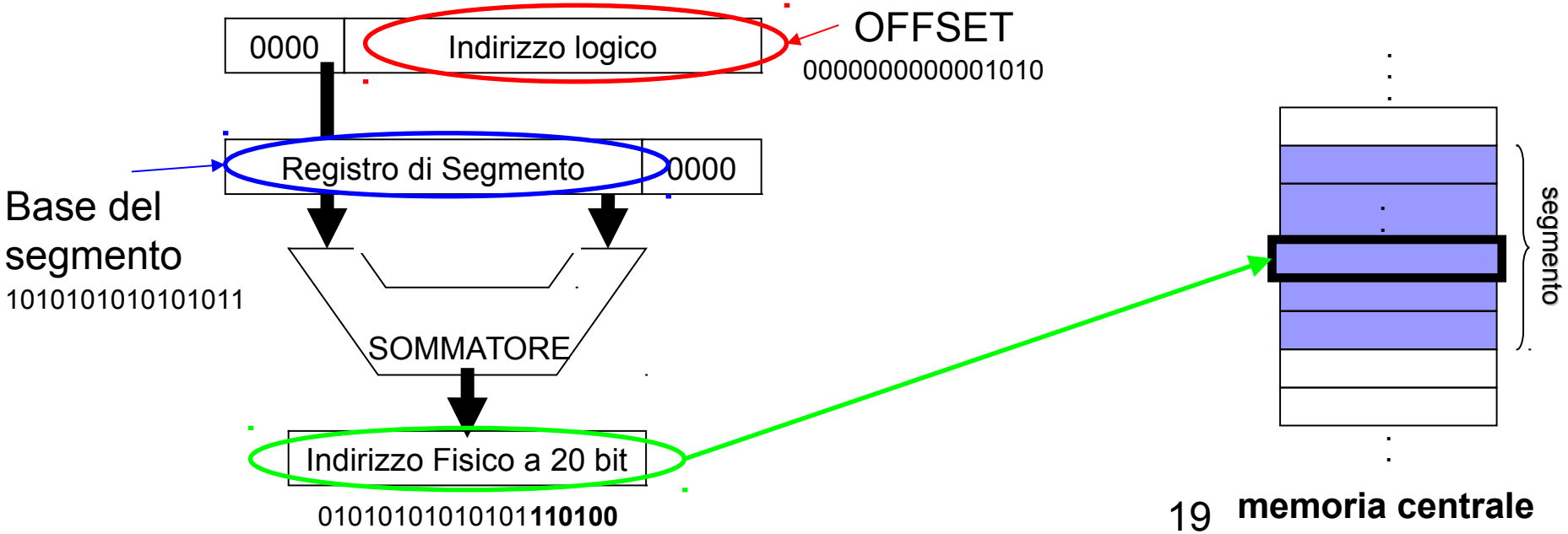
  ADD AX, ES:VAR AX ← M[ES:offset(VAR)]



Calcolo dell'indirizzo fisico

Base del segmento + OFFSET



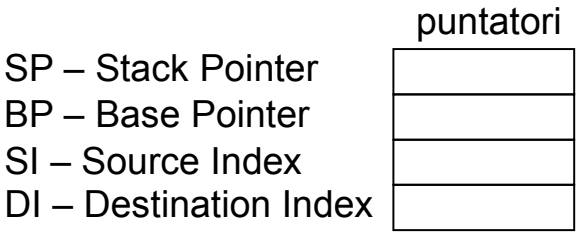


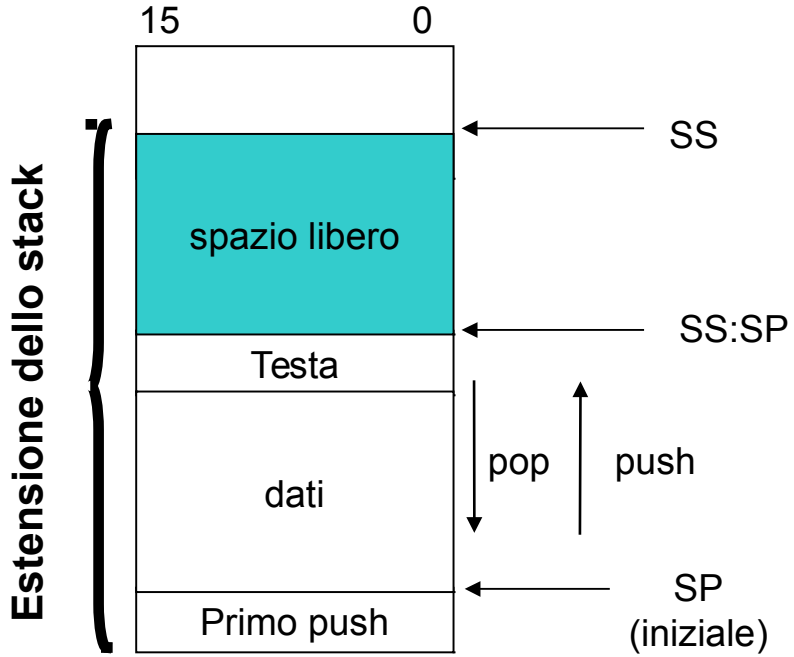
Svantaggi degli indirizzi segmentati

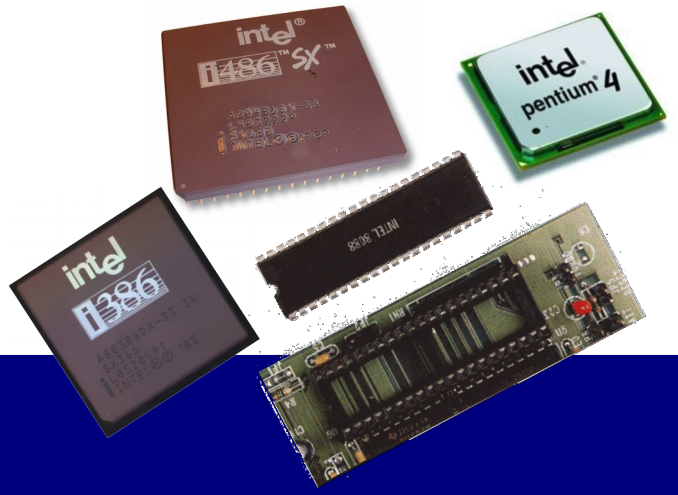
- Un singolo selettore può indirizzare solo 64K di memoria:
  - Se un programma occupa più di 64K, un solo valore del registro CS non è sufficiente per la sua intera esecuzione:
    - Il programma deve essere suddiviso in sezioni (dette, appunto, segmenti) di non più di 64K
    - Quando l'esecuzione passa da un segmento ad un altro, il valore di CS deve essere opportunamente aggiornato
  - Problemi simili si riscontrano con il registro DS (che indirizza il Segmento Dati) in caso di grosse moli di dati
- Ogni byte in memoria non possiede un indirizzo segmentato univoco
  - Es:indirizzo fisico 04808 → può essere riferito come: 047C:0048, 047D:0038, 047E:0028, 047B:0058

Lo stack in memoria centrale

- È gestito dal microprocessore utilizzando
  - il registri di segmento SS
  - i registri puntatori EP e SP
- La CPU può leggere/scrivere parole di 1 o 2 Byte
  - I 2 Byte non devono essere necessariamente allineati
- La durata dell'operazione di accesso in memoria è
  - Un ciclo di clock per istruzioni con indirizzo pari
  - Due cicli di clock per istruzioni con indirizzo dispari





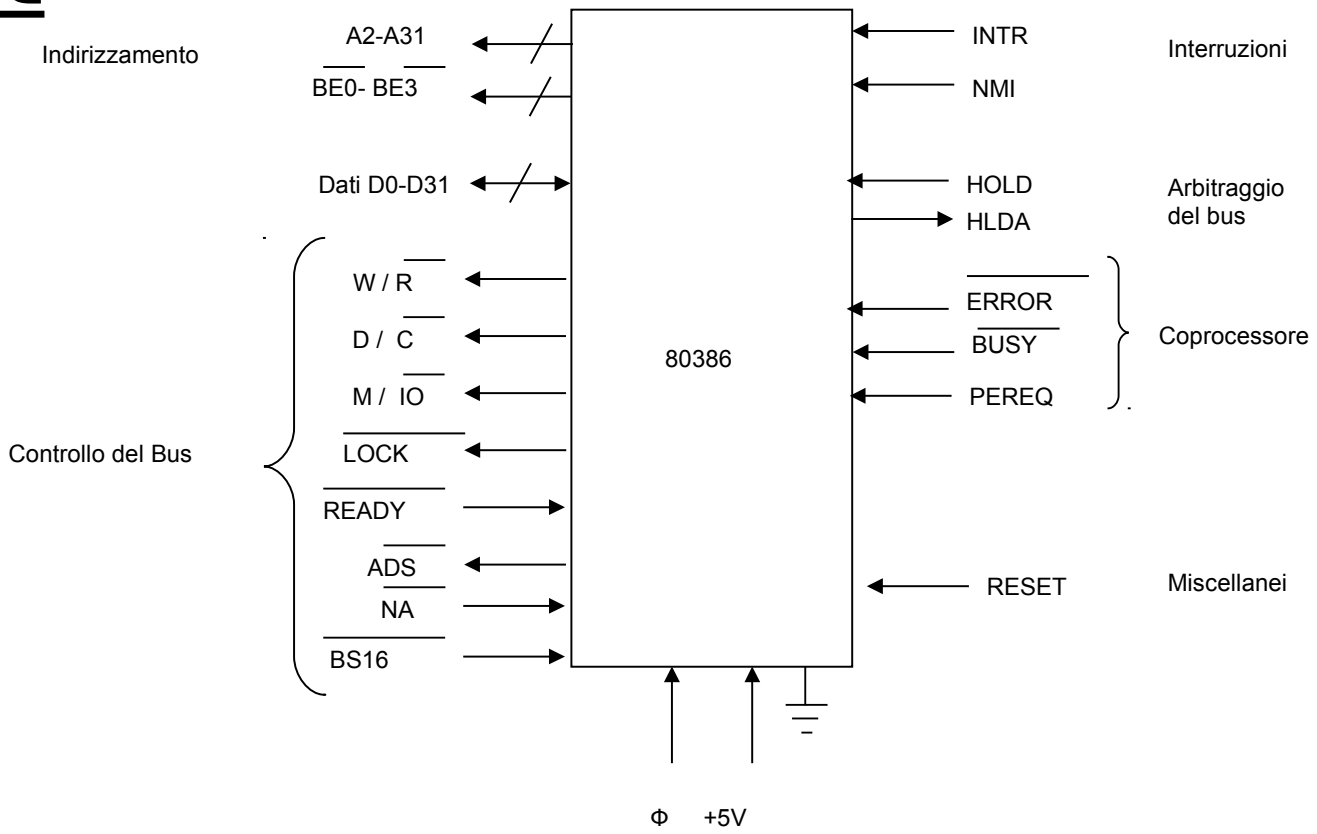


Il Microprocessore 80386

Architettura dei Calcolatori Elettronici

A cura di:
ing. Rosa Anna Micillo

Il pinout



pinout logico

- microprocessore a 32 bit
- 132 piedini
- massima frequenza di clock 16 MHz
- numero medio di cicli di clock per istruzione - CPI 4,4
- $t_{esec} = 4{,}4/16\text{MHz} = 0{,}275\mu s \rightarrow$ **3,6 MIPS**

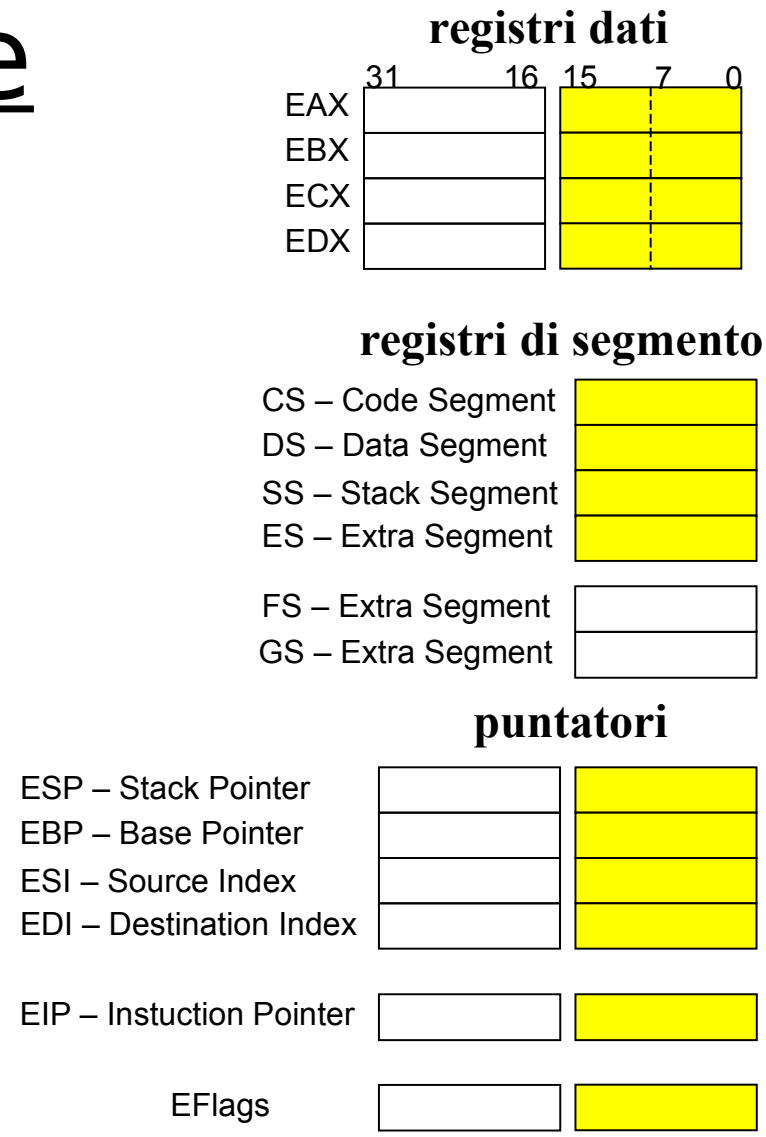
Innovazioni più significative

- Architettura a 32 bit
- Paginazione
- Supporto al Multitasking

Il modello di programmazione

Parallelismo interno di 32 bit

- 4 registri dati
- 4 registri puntatori
- registro IP
- registro di stato
- **6 registri di segmento a 16 bit**
  - FS e GS sono usati per individuare segmenti extra per ridurre il numero di volte che i registri di segmento devono essere caricati



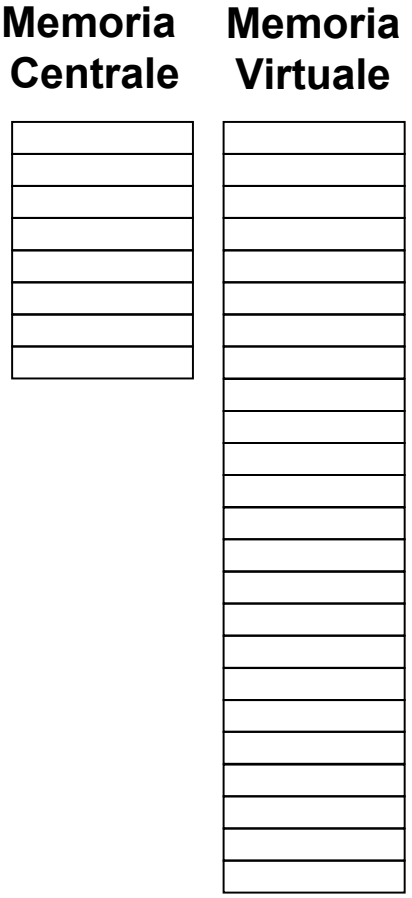
Paginazione

- Separare i concetti di *Spazio di Indirizzamento* e di *Locazioni di Memoria*

- …senza paginazione
  - indirizzo di n bit
  - spazio di indirizzamento 2^n
  - memoria principale contenente 2^n

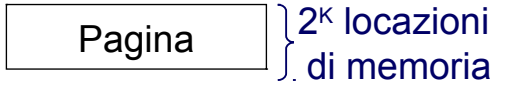
- …con paginazione
  - indirizzo di m bit (con m > n)
  - spazio di indirizzamento $2^m \rightarrow$ ***memoria virtuale***
  - memoria principale contenente 2^n

Si ha l'illusione di avere una memoria centrale più capiente

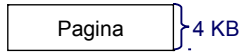


Paginazione

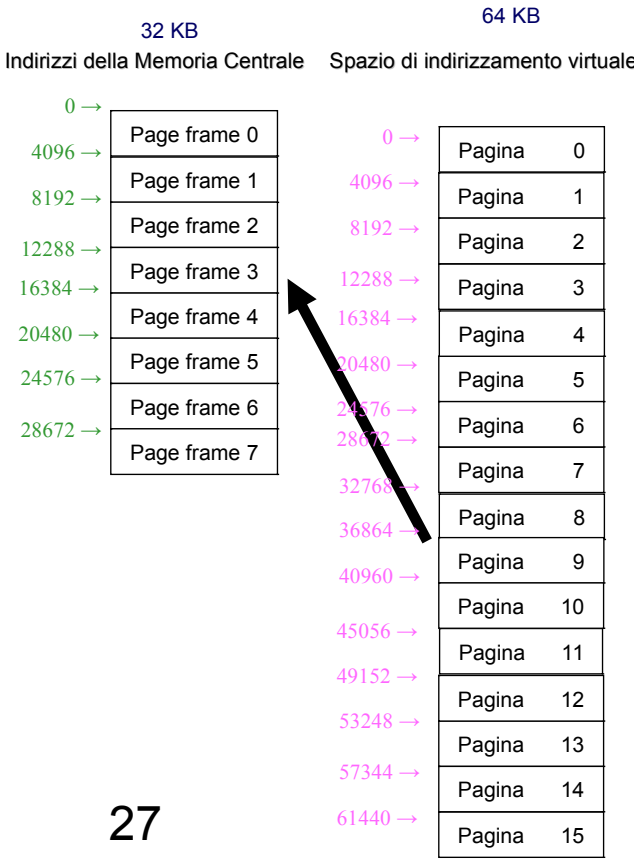
- La memoria virtuale è divisa in pagine
  - una ***pagina*** è un gruppo di 2^k locazioni di memoria



- La memoria centrale è divisa in page frame
  - un ***page frame*** è un "contenitore" per una pagina



- È necessario tener traccia di quali pagine si trovano nella memoria centrale e in quali *page frame* sono memorizzate



- Esempio
  - pagine di 4KB (2^{12}B)
  - memoria centrale di 32 KB (2^{15}B)
  - memoria virtuale di 64KB (2^{16}B)
  - locazioni di memoria da 32 bit

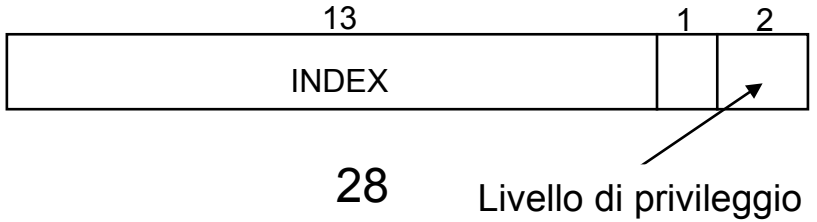
$k = 10$
$n = 13$
$m = 14$

Organizzazione della memoria

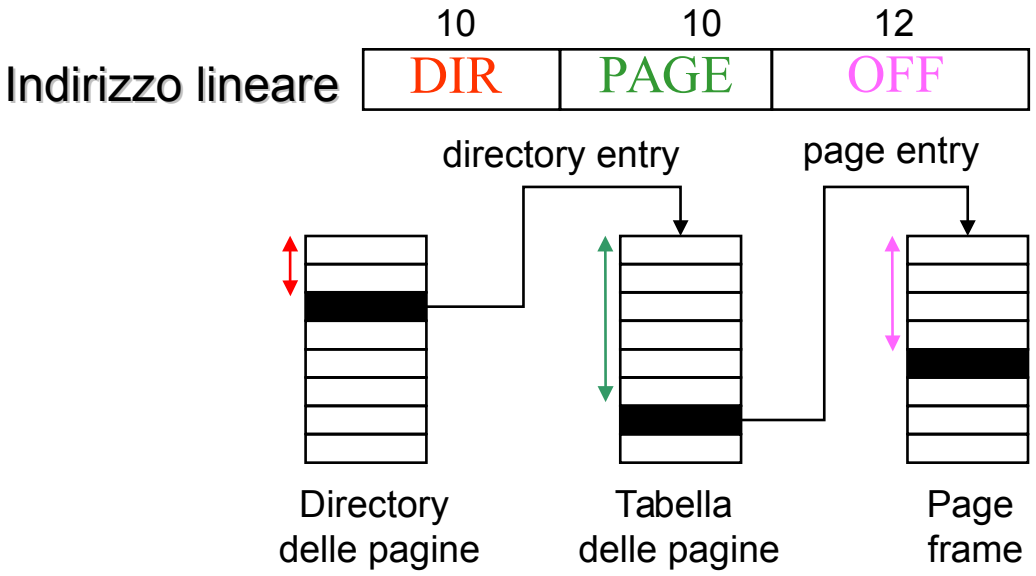
- utilizza sia la *paginazione* che la *segmentazione*

- paginazione
  - semplifica l'utilizzo di S.O. che prevedono la gestione della Memoria Virtuale
  - ***memoria virtuale*** da **4 GB** (2^{32} B)

- segmentazione
  - 16K segmenti indipendenti
  - dimensione massima **4 GB** (2^{32} B)
  - Gestione dei segmenti:
    - Tabella del descrittore locale LDT descrive i segmenti di un programma
    - Tabella del descrittore globale GDT descrive i segmenti di sistema (incluso il S.O.)
  - Per accedere ad 1 segmento bisogna caricare il selettore per quel segmento presente in uno dei registri di segmento
    - lunghezza del selettore 16 bit
    - 2^{13} ovvero 8K segmenti per ciascuna tabella



Calcolo dell'indirizzo fisico



Modalità di Funzionamento

- Modalità reale (Real mode)
- Modalità protetta (Protected mode)
- Modalità “8086 virtuale” (Virtual-8086 mode)

Real Mode

- unica modalità prevista dal 8086/88
- ogni programma può accedere a qualsiasi locazione di memoria
- tutti i processori intel partono in questa modalità perché è l'unica supportata dal BIOS
- non c'è supporto per la memoria virtuale e per il multitasking
- MS-DOS opera nativamente con il processore in tale modalità

Real Mode

- massima memoria indirizzabile di 1MB (2^{20} B)
  - Gli indirizzi vanno da 00000 a FFFFF esadecimale
  - Sono necessari 20 bit per codificare l'indirizzo → problema con i registri a 16 bit dell'8086
  - Soluzione: si utilizzano 2 registri a 16 bit per determinare l'indirizzo
    - Primi 16 bit: il *selettore* contenuto in un registro di segmento
    - Secondi16 bit: l'*offset*
    - Indirizzo fisico è *selettore:offset*
      - *Indirizzo fisico = 16 x selettore +offset*
      - *Es: 047C:0048 → Indirizzo fisico = 047C0 +0048 = 04808*

Protected Mode

- modalità “nativa” di IA-32
- memoria protetta: evita corruzione memoria da parte di altri programmi
- supporto alla memoria virtuale
- memoria indirizzabile: 4 GByte (2^{32})

Protected Mode

- Nel'286 i valori dei selettori sono interpretati diversamente rispetto al real mode:
  - Ogni selettore è indice all'interno di una *descriptor table*, contenete informazioni quali:
    - Presenza o assenza in memoria centrale del segmento specificato
    - Sua locazione in caso di presenza in memoria
    - Permessi di accesso (read-only, read-write, ecc.)
  - I selettori così utilizzati consentono la gestione della memoria virtuale
  - Persiste lo svantaggio legato all'utilizzo di offset a 16 bit che impongono un limite di 64 KB sulla dimensione dei segmenti

Protected Mode

- Il 386 introduce il protected mode a 32 bit
- Gli offset sono estesi a 32 bit → i segmenti possono raggiungere una dimensione di 4 GB
- i segmenti possono essere suddivisi in pagine da 4 KB ciascuna:
  - La memoria virtuale lavora con le pagine
  - In un dato istante è possibile che solo una parte di un dato segmento risieda in memoria.

Virtual-8086 mode

- "Real mode" simulato all'interno del "Protected Mode"
  - le istruzioni che possono causare problemi al sistema non possono essere eseguite → trap al S.O. che le simula
- è possibile eseguire più programmi 8086 protetti tra loro (anche il S.O. ne risulta protetto) → *sistema multitasking* in cui ogni processo esegue un programma 8086