



FONDAMENTI DI INFORMATICA

Lezione n. 11

- **ARCHITETTURA INTERNA**
- **ARCHITETTURA ESTERNA**
- **CODICE MACCHINA**
- **MODI DI INDIRIZZAMENTO**
- **ARCHITETTURE A PIU' INDIRIZZI**

In questa lezione verranno introdotti i concetti di base relativi alla architettura interna di una CPU e al linguaggio macchina (o linguaggio assembler).



ARCHITETTURA INTERNA E ESTERNA

ARCHITETTURA INTERNA: Struttura interna della CPU.
Migliore compromesso possibile fra le prestazioni e i costi avendo come vincolo la tecnologia.

ARCHITETTURA ESTERNA: Come il processore è visto da chi lo deve programmare. Insieme delle istruzioni, dei registri, dei modi di indirizzamento e dei tipi di dato ammessi dalle istruzioni.

Attraverso livelli di interpretazione si realizzano le funzioni definite all'esterno:

- **Linguaggio macchina (o assembler).**
- **Linguaggio di microprogramma.**
- **Comandi alla parte operativa.**



LINGUAGGIO MACCHINA

Il linguaggio macchina o assembler di una CPU definisce:

- Le operazioni possibili.**
- Le risorse possibili e la loro utilizzazione.**

La maggior parte delle istruzioni è del tipo:

$$X_1 \leftarrow f(X_1, X_2, \dots, X_n)$$

con n eguale a 1, 2 o (raramente) 3.

Funzioni complesse sono realizzate sfruttando i livelli di interpretazione senza modificare l'architettura interna.

L'architettura interna è direttamente influenzata dalle caratteristiche sintattiche dell'architettura esterna (tipi di dato, ...).



LINGUAGGIO MACCHINA

Ogni istruzione è definita da:

- **Codice macchina:** $\Rightarrow 10010101$
- **Codice mnemonico:**

MOV A,B \Rightarrow "A \Leftarrow B"

Corrispondenza 1 a 1 tra i due codici.

Ogni istruzione in linguaggio macchina deve definire:

- **Operazione da svolgere.**
- **Operandi coinvolti.**
- **Posizione dell'istruzione successiva.**



LINGUAGGIO MACCHINA

ISTRUZIONE SUCCESSIVA

- **Le istruzioni sono eseguite in sequenza.**
- **L'indicazione relativa all'istruzione successiva è spesso implicita.**
- **Il PC (Program Counter) memorizza l'indirizzo della istruzione da eseguire.**
- **Per alterare la sequenza sono introdotte istruzioni di salto:**
(PC \Leftarrow X).



LINGUAGGIO MACCHINA

CODICE MACCHINA

In generale il codice macchina è suddiviso in campi:



CODICE MNEMONICO



Il codice mnemonico rispecchia la struttura del codice macchina.

MODI DI INDIRIZZAMENTO

Ogni operando è associato a un dato di cui occorre conoscere la localizzazione attraverso il suo INDIRIZZO.

Modi diversi per indicare la posizione:

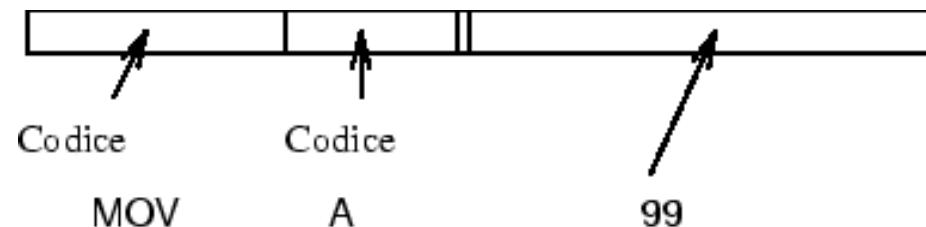
- **IMMEDIATO**
- **DIRETTO**
- **INDIRETTO**

MODO IMMEDIATO

Il dato è contenuto nel codice macchina.

MOVI A, 99 o

MOV A, #99



MODI DI INDIRIZZAMENTO

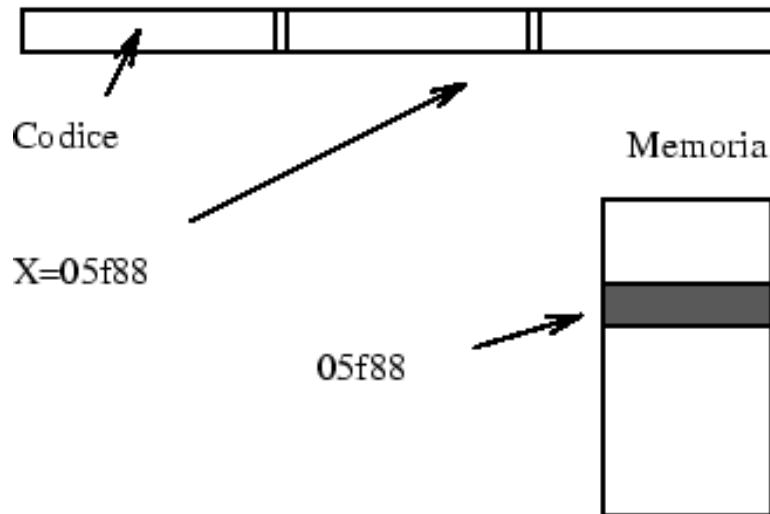
MODO DIRETTO

Il codice macchina contiene l'indirizzo del dato.

MOV A, X

X può essere:

- Indirizzo della cella di memoria contenente il dato.**
- Codice registro interno.**



MODI DI INDIRIZZAMENTO

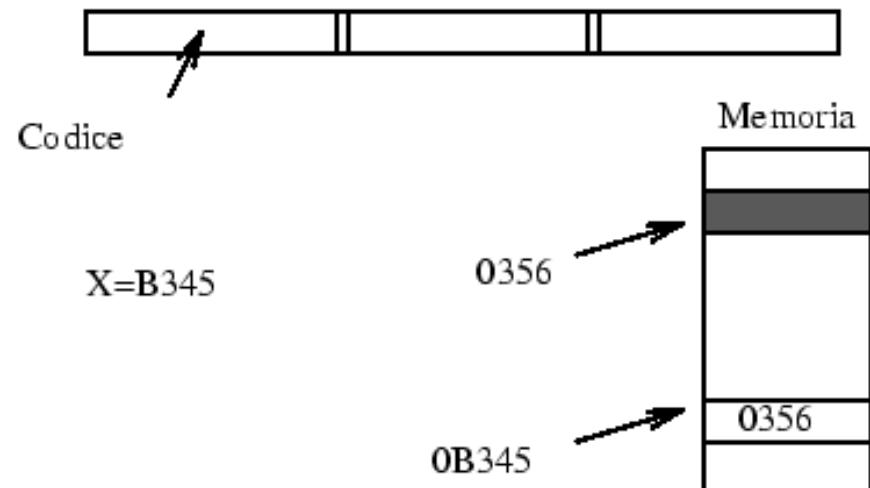
INDIRETTO

Il codice macchina contiene l'indirizzo della locazione di memoria che contiene l'indirizzo del dato:

MOV A, (X)

X può essere:

- Indirizzo della cella di memoria contenente l'indirizzo del dato.**
- Codice registro interno contenente l'indirizzo del dato.**





TIPI DI INDIRIZZO

Il valore dell'indirizzo può essere espresso in modo:

- **ASSOLUTO:**
L'indirizzo completo compare nel campo operando.
Svantaggio: la lunghezza del campo indirizzo genera codice di dimensione elevata.
- **RELATIVO:**
Nel campo operando compare solo lo spiazzamento (scostamento) relativo (differenza rispetto al valore contenuto) al PC.
Lo spiazzamento può essere:
 - **contenuto in un byte,**
 - **un valore negativo.**



TIPI DI INDIRIZZO

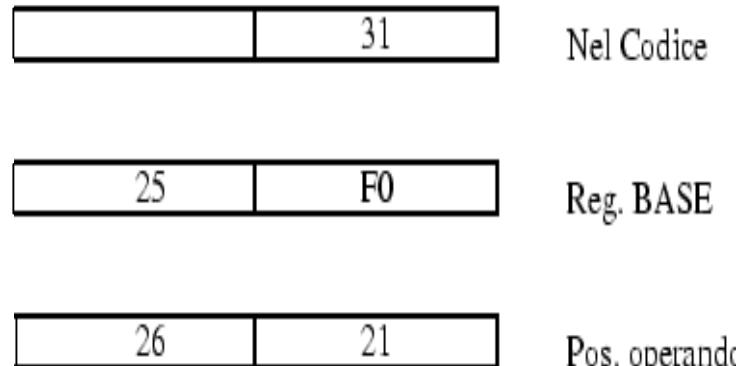
INDIRIZZO COMPOSTO

**Estendendo il concetto di indirizzo relativo, al posto
del PC, si utilizza:**

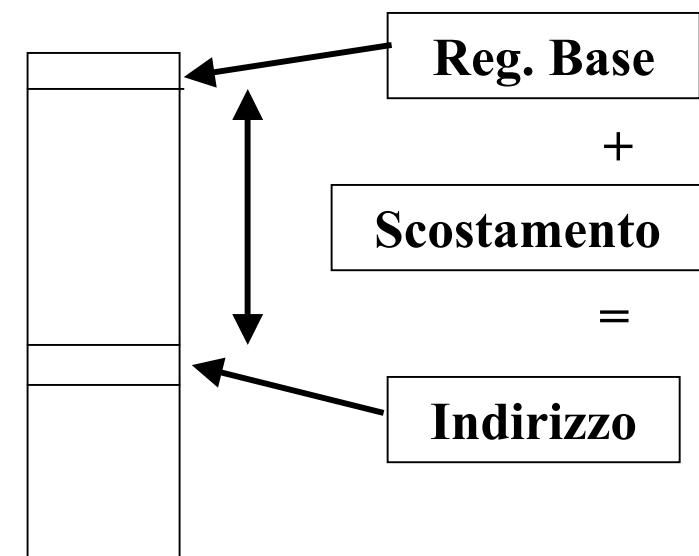
- **Registro Base.**
- **Registro Indice.**

REGISTRO BASE

- **Nel codice operativo compare solo lo spiazzamento rispetto a un valore contenuto nel registro base.**
- **L'indirizzo effettivo si ottiene sommando lo spiazzamento al contenuto del registro base.**
- **Permette al processore di accedere ad una nuova zona di memoria solo cambiando il contenuto del registro base.**



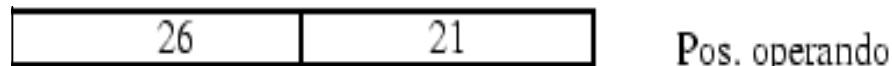
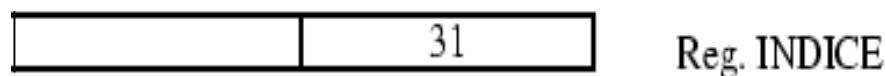
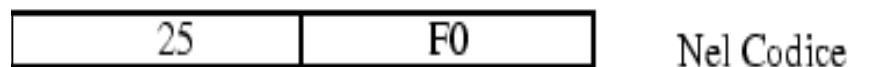
Vettore in memoria





REGISTRO INDICE

- **Nel codice compare l'indirizzo iniziale di un blocco, la posizione all'interno del blocco è individuata mediante un registro.**
- **Il vettore X_0, X_1, \dots, X_N è memorizzato in locazioni consecutive.**
- **Il codice operativo contiene l'indirizzo di X_0 .**
- **Il registro R contiene l'indice i.**





STACK POINTER

MODIFICA PUNTATORI

Un altro modo di indirizzamento:

MOV A,(X)+

**X è un registro che al termine dell'esecuzione
dell'istruzione viene incrementato.**

ALTRI CASI:

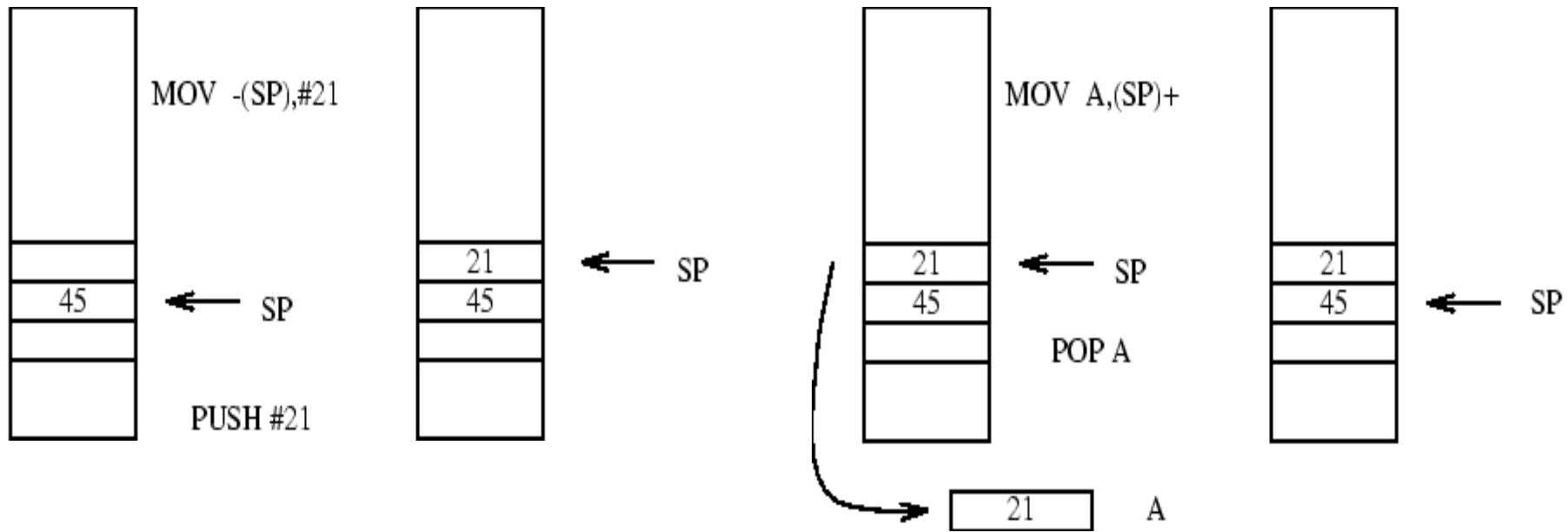
- **(X)+ : post-increment**
- **(X)- : post-decrement**
- **+(X) : pre-increment**
- **-(X) : pre-decrement**



STACK POINTER

PUSH

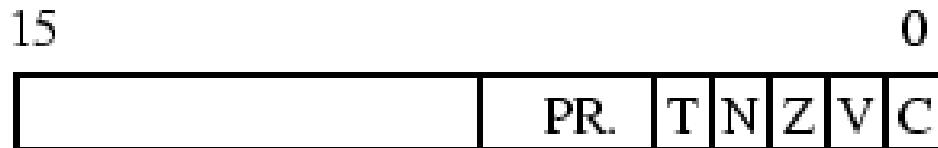
POP





REGISTRO DI STATO

Un caso molto semplice PDP11 (1970)



- **PR** - livello di priorità del processore.
- **T** - modo passo-passo.
- **Z,N,C,V** - risultato zero, negativo, con riporto, overflow.



ARCHITETTURA INTERNA

La gestione efficace di operazioni particolari svolte frequentemente richiede la presenza di funzioni specializzate:

- **Registri dedicati:**
 - **Registro indice**
 - **Registri base**
 - **Registro di stato**
 - **Stack Pointer**
- **Gestione sottoprogrammi e interruzioni.**

ARCHITETTURA ESTESA

