

**Corso di Calcolatori Elettronici I  
A.A. 2010-2011**

---

**Modi di indirizzamento  
del processore MC68000  
(parte seconda)**

Lezione 23

**Prof. Roberto Canonico**



Università degli Studi di Napoli Federico II  
Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica (allievi A-DE+Q-Z)  
Corso di Laurea in Ingegneria dell'Automazione

---

**Modi di indirizzamento MC68000**

---

- Register Direct
    - Data-register Direct
    - Address-register Direct
  - Immediate (or Literal)
  - Absolute
    - Short (16 bit)
    - Long (32 bit)
  - Address-register Indirect
    - Auto-Increment
    - Auto-Decrement
  - Indexed short
  - Based
  - Based Indexed
    - Short
    - Long
  - Relative
  - Relative Indexed
    - Short
    - Long
-

## Riepilogo modi fondamentali

---

- **Literal (immediate) addressing**
    - usato per costanti che non cambiano
  - **Register direct addressing**
    - usato per variabili mantenute in registri del processore
  - **Direct (absolute) addressing**
    - usato per variabili che risiedono in memoria ad un indirizzo costante (noto a tempo di assemblaggio)
- 

## LEA: Load Effective Address

---

**Operazione:** [An] ← <ea>  
**Sintassi:** LEA <ea>,An  
**Esempio:** LEA table,A3  
**Attributi:** Size = longword

**Descrizione:**

Calcola l'indirizzo effettivo (<ea>) del primo operando, generalmente espresso in forma simbolica, e lo pone nel registro indirizzo specificato dal suo secondo operando. Non influenza i flag di stato:

X N Z V C

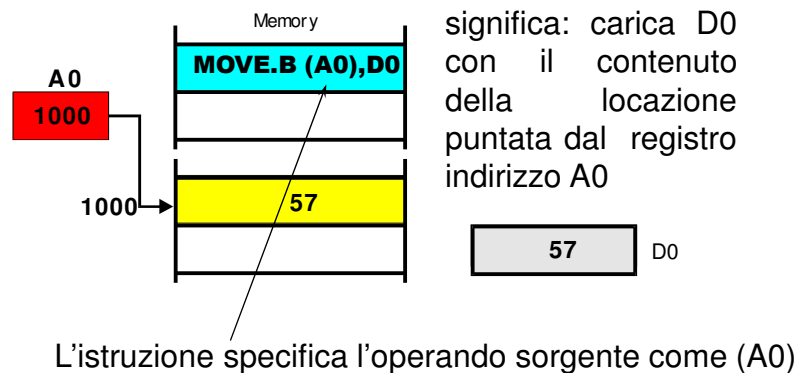
- - - - -

---

## Address Register Indirect Addressing

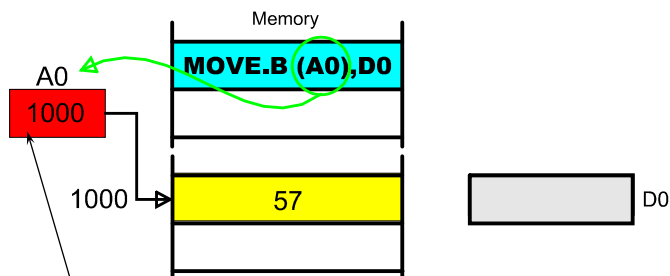
- L'istruzione specifica uno dei registri indirizzo
- Il registro indirizzo specificato contiene l'indirizzo effettivo dell'operando
- Il processore accede all'operando puntato dal registro indirizzo
- Esempio:
  - MOVE.B (A0),D0

## Address Register Indirect: funzionamento



## Address Register Indirect: funzionamento

---

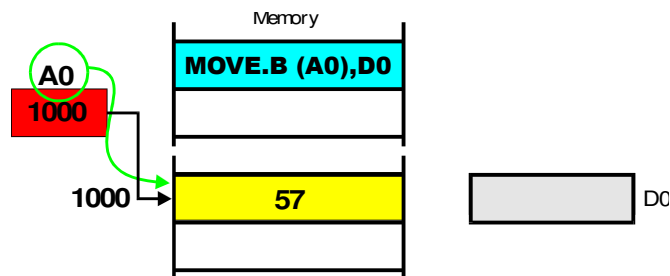


Il registro indirizzo nell'istruzione  
specifica un registro indirizzo che  
contiene l'indirizzo dell'operando

---

## Address Register Indirect: funzionamento

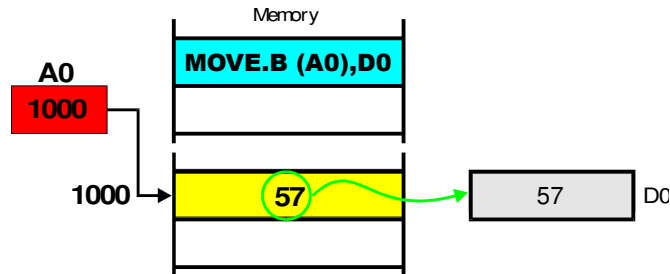
---



Il registro indirizzo è usato per accedere  
all'operando in memoria

---

## Address Register Indirect: funzionamento



Alla fine, il contenuto della locazione puntata da A0 viene copiato nel registro dati

## Auto-post-increment

- L'istruzione specifica uno dei registri indirizzo, usando la modalità Address Register Indirect.
- Se il modo di indirizzamento è specificato come  $(An)_+$ , il contenuto del registro indirizzo è incrementato di una quantità pari alla dimensione dell'operando *dopo l'uso* ("post-incremento")
- Esempio:
  - `MOVE.W (A0)+, D0`      Usa A0 per la MOVE e poi gli aggiunge 2 (2 poiché l'accesso è di tipo `.W = 2 byte`). Di fatto, l'istruzione esegue un pop in D0 dallo stack puntato da A0

## Auto-pre-decrement

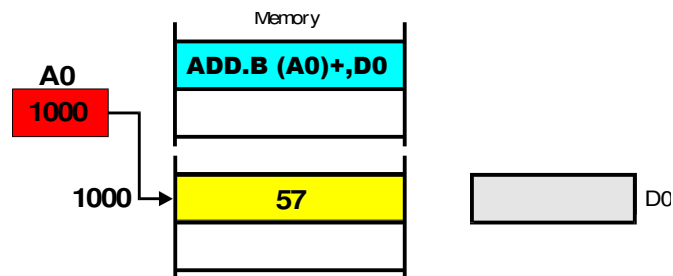
- L'istruzione specifica uno dei registri indirizzo
- Se il modo di indirizzamento è specificato come  $-(A_n)$ , il contenuto del registro indirizzo è decrementato di una quantità pari alla dimensione dell'operando *prima dell'uso* ("pre-decremento")

- **Esempio:**

- `MOVE.W D0,-(A0)`

Sottrae 2 ad A0 e poi lo usa per la MOVE (2 poiché l'accesso è di tipo .W = 2 byte). Di fatto, l'istruzione esegue un push di D0 sullo stack puntato da A0.

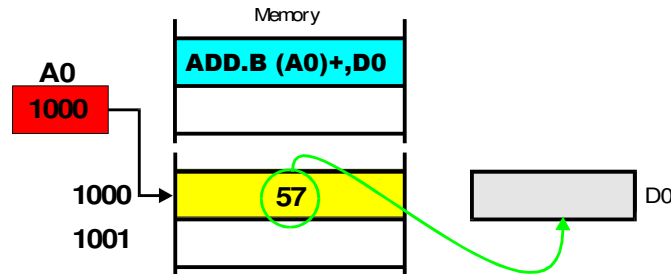
## Auto-post-increment: funzionamento



Il registro indirizzo contiene 1000  
ovvero "punta" alla locazione 1000

## Auto-post-increment: funzionamento

---

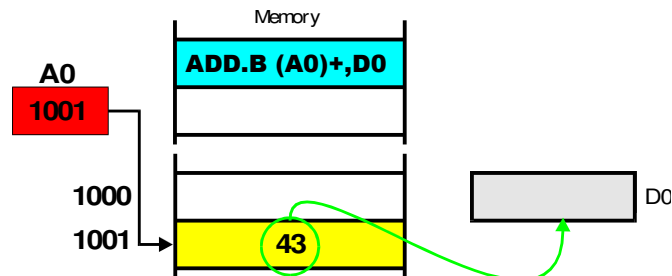


Il registro A0 viene usato per accedere alla locazione di memoria 1000 e il contenuto di questa locazione (57) viene sommato a D0

---

## Auto-post-increment: funzionamento

---



Dopo che l'istruzione è stata eseguita, il contenuto di A0 viene incrementato, per puntare alla locazione successiva

---