

Corso di Calcolatori Elettronici I
A.A. 2011-2012

**Caratteristiche e modello di
programmazione del
processore MC68000**

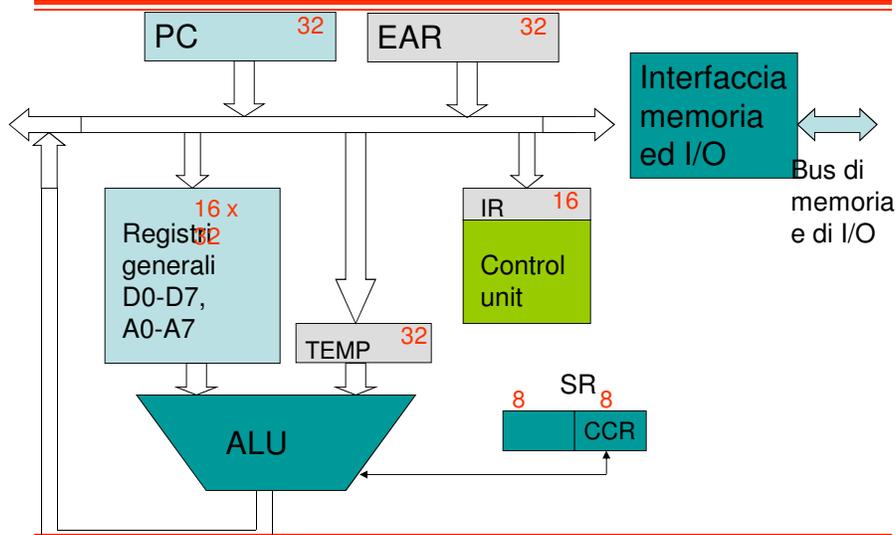
Lezione 14

Prof. Roberto Canonico



Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica (allievi A-DA)
Corso di Laurea in Ingegneria dell'Automazione

**Architettura del processore
MC68000**



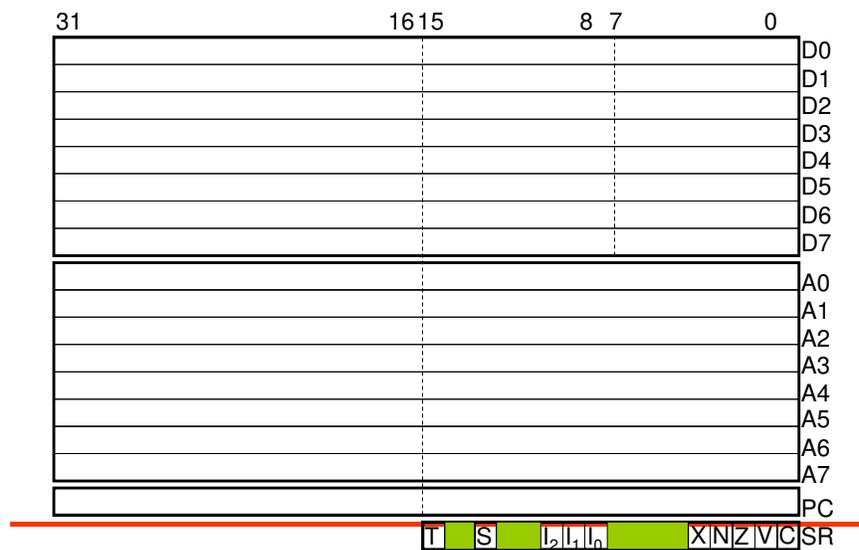
Caratteristiche del processore MC68000

- Dati:
 - All'esterno:
parola di 16 bit (16 pin per i dati)
 - All'interno:
registri di 32 bit
 - Indirizzi:
 - All'esterno:
24 bit (spazio di indirizzamento fisico $2^{24} = 16M$)
 - 512 pagine (2^9) da 32K (2^{15})
 - All'interno:
32 bit
-

Caratteristiche del processore MC68000

- Parallelismo della memoria:
 - Parole di 16 bit, ognuna costituita da due byte con indirizzi distinti (memoria byte addressable)
 - Convenzioni della memoria:
 - Una parola deve essere allineata ad un indirizzo pari (even boundary)
 - Convenzione big-endian
-

Modello di programmazione del MC68000



Codifica delle istruzioni di un processore in stile RISC

ESEMPIO: una CPU con istruzioni a lunghezza fissa di 32 bit

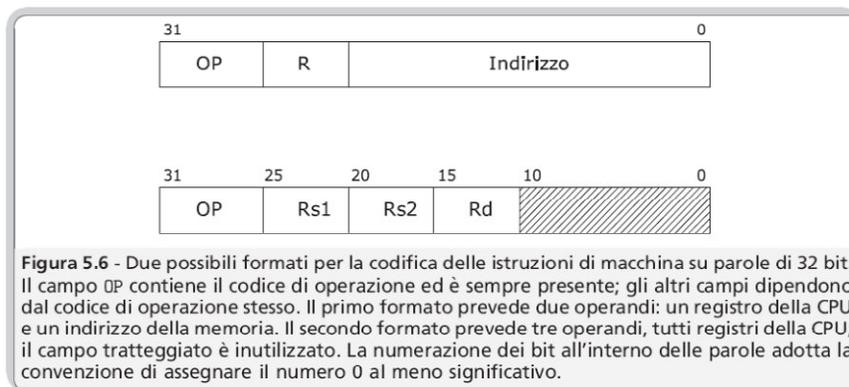


Figura 5.6 - Due possibili formati per la codifica delle istruzioni di macchina su parole di 32 bit. Il campo OP contiene il codice di operazione ed è sempre presente; gli altri campi dipendono dal codice di operazione stesso. Il primo formato prevede due operandi: un registro della CPU e un indirizzo della memoria. Il secondo formato prevede tre operandi, tutti registri della CPU; il campo tratteggiato è inutilizzato. La numerazione dei bit all'interno delle parole adotta la convenzione di assegnare il numero 0 al meno significativo.

LD R1, Var ; R1 ← M[Var]

ADD R1, R2, R3 ; R1 ← R2 + R3

da: G. Bucci. Calcolatori Elettronici – Architettura e organizzazione. © McGraw-Hill, 2009

Sequenze di istruzioni in memoria

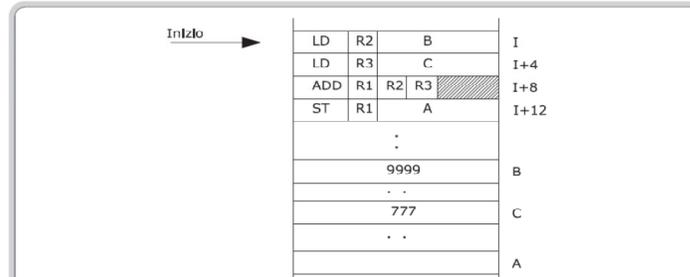


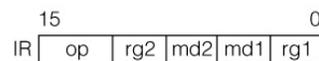
Figura 5.7 - Mappa della memoria con le tre variabili e la sequenza di istruzioni equivalenti allo statement C "a = b+c" per il caso della seconda soluzione. Si è fatta l'ipotesi che la variabile b valga 9999 e che la variabile c valga 777. Dopo l'esecuzione del tratto di codice, nella posizione A si trova il numero 10776, qualunque sia il contenuto precedente della cella, mentre il contenuto delle altre due è immutato.

Lo statement $a = b + c$ si traduce come:

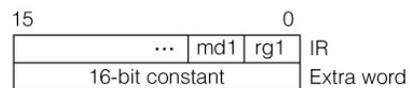
```
LD R2, B           ;B indirizzo a cui è allocata la parola b
LD R3, C           ;C indirizzo a cui è allocata la parola c
ADD R1, R2, R3
ST A, R1           ;A indirizzo a cui è allocata la parola a
```

da: G. Bucci. Calcolatori Elettronici – Architettura e organizzazione. © McGraw-Hill, 2009

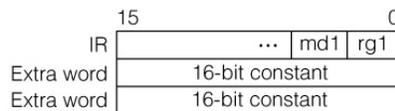
Codifica istruzioni MC68000



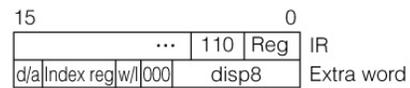
(a) A 1-word move instruction



(b) A 2-word instruction



(c) A 3-word instruction



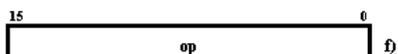
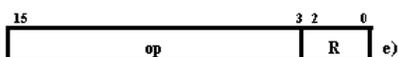
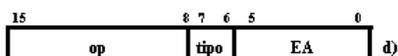
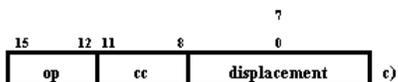
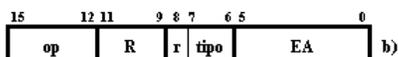
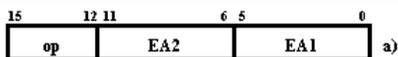
(d) Instruction with indexed address

**Codifica a lunghezza variabile multipla di 2 byte:
opcode word + extra word(s)**

Copyright © 2004 Pearson Prentice Hall, Inc.

Codifica istruzioni MC68000 (2)

Si analizza qui solo la struttura della prima word (16 bit) del codice di una istruzione, detta **OPCODE WORD**



op= codice operativo
R =indirizzo di registro D oppure A
tipo= tipo di operando (B,W,L)
displacement =indirizzo di salto relativo

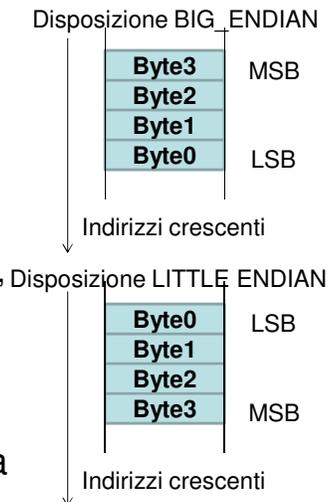
EA = Effective Address (cfr. § 4)
r =il registro è origine o destinazione
cc= codice di condizione

Processori a carattere e processori a parola

- I processori “a carattere” accedono alla memoria con parallelismo 1 byte (8 bit)
 - prime CPU ad accumulatore
- I processori “a parola” hanno la capacità di indirizzare ed accedere la memoria per unità (parole o *word*) di 16 bit, 32 bit o 64 bit
- In questi sistemi (tranne pochissime eccezioni nel passato) l’unità indirizzabile di memoria (*locazione*) è ancora il byte
 - Si parla di sistemi a memoria *byte-addressable*: ogni byte ha il suo indirizzo
- Terminologia Motorola 68000:
 - word = 2 byte, longword = 4 byte

Big-endian e little-endian

- I processori “a parola” possono disporre in memoria i byte che formano una parola in due modi
 - *Big-endian*:
i byte sono disposti in memoria in modo che il più significativo MSB occupi la locazione di memoria di indirizzo minore, e poi via via gli altri, fino a quello meno significativo LSB che è collocato nella locazione di indirizzo maggiore
 - *Little-endian*: disposizione opposta
- Il processore Motorola 68000 usa la convenzione Big Endian



Big-endian e little-endian: un esempio

- Immaginiamo di avere un processore a parola, con parole di 32 bit (4 byte) e voler scrivere in memoria il valore intero (esadecimale) **\$12FA34ED** all'indirizzo **812**
- Le figure sottostanti illustrano il contenuto della memoria nei due casi big-endian e little-endian

