

Corso di Calcolatori Elettronici I

A.A. 2011-2012

Modi di indirizzamento avanzati

Lezione 26

Prof. Antonio Pescapè

Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica



Utilità di ulteriori modi di indirizzamento

- Quando un programma deve accedere a dati in memoria può usare:
 - Indirizzamento assoluto: Es. MOVE \$8100,D0
 - Indirizzamento indiretto: Es. MOVE (A1),D0
 - Vantaggio del modo indiretto: l'indirizzo è determinato a runtime, e la stessa istruzione (ad es. all'interno di un ciclo) può operare su dati posti in locazioni diverse
 - Ci sono situazioni in cui un solo grado di libertà attraverso un registro An non è sufficiente
-

Utilità di ulteriori modi di indirizzamento (cont.)

- Ci sono situazioni in cui un solo grado di libertà attraverso un registro A_n non è sufficiente
 - Esempi
 - Accedere agli elementi di una matrice $A(i,j)$
 - Accedere ai campi di un array di record
 - Accedere ai campi di un record la cui posizione è determinata a tempo di esecuzione
 - Es. record di attivazione di una subroutine
 - Soluzione:
 - Metodi di indirizzamento che costruiscono l'EA mediante due o più componenti
-

Indexed addressing

- In generale, l'Indexed Addressing combina due componenti mediante somma, per formare l'EA
 - Il primo componente è detto *base address* ed è specificato come parte dell'istruzione (come nell'absolute addressing)
 - Il secondo componente è detto *index register* e contiene il valore da sommare al base address per ottenere l'EA
 - È adatto per accedere ai valori di array e di tabelle
 - Il processore MC68000 non supporta esplicitamente l'Indexed Addressing.
-

Based Addressing

- Based Addressing è esattamente l'inverso dell'Indexed Addressing, in quanto combina due componenti mediante somma, per formare l'EA, ma:
 - Il primo componente è detto *displacement* ed è specificato come parte dell'istruzione (come nell'absolute addressing)
 - Il secondo componente è detto *base address* ed è contenuto in un registro
 - È adatto per accedere a campi di record di cui si conosca la posizione relativa ad assembly time, ma non quella iniziale
 - Il processore MC68000 supporta il Based Addressing attraverso il modo ***Indirect with displacement*** d16(An)
 - Es. MOVE .L 6 (A0) , D2
-

Based Indexed

- Based Indexed Addressing combina due componenti mediante somma, per formare l'EA, ma:
 - Il primo componente è detto registro base e contiene il *base address*
 - Il secondo componente è detto registro indice e contiene il *displacement*
 - Consente di calcolare a run time sia la posizione iniziale che quella relativa di tabelle ed array
 - Il processore MC68000 supporta lo Short Based Indexed ed il Long Based Indexed
 - Anche detti Indirect with displacement and index
-

Relative Addressing

- “Relative” indica che il calcolo dell’indirizzo è relativo al Program Counter (PC)
 - Questi modi di indirizzamento calcolano l’indirizzo effettivo come la somma di un displacement fisso specificato nell’istruzione e del valore corrente del PC
 - Fanno spesso uso di displacement piccoli, di 8 o 16 bit, per specificare indirizzi vicini all’istruzione corrente, anziché ricorrere a indirizzi assoluti di 32 bit
 - Il 68000 non consente di utilizzare questi modi di indirizzamento per specificare operandi che potrebbero essere modificati
-

Relative Indexed Addressing

- Variante del Relative
 - Funziona come il Based Indexed, ma il base register è sostituito dal PC
 - Può essere usato per saltare ad aree di memoria read-only, contenenti dati o istruzioni
-