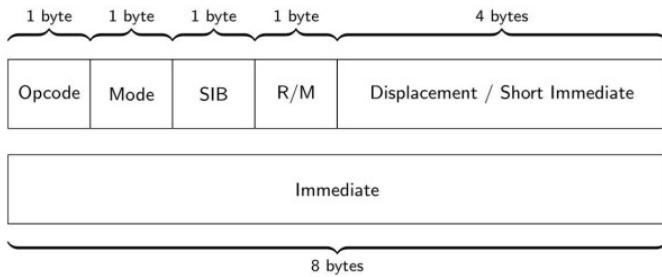


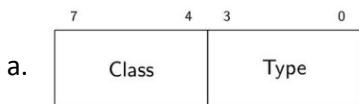
# Lezione 18 Z64 4

mercoledì 8 novembre 2023 15:49

Torniamo al formato istruzioni macchina :

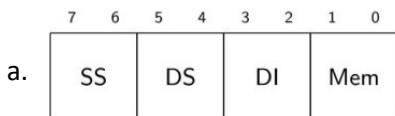


1. **Op-code:**



- b. Dove nella Class ci vanno i 4 bit che servono a specificare il tipo/famiglia delle istruzioni , mentre type specifica la precisa istruzione della famiglia .

2. **Mode:**



- b. SS rappresenta la codifica della dimensione dell'operando sorgente ( 2 bit).  
c. DS rappresenta la codifica della dimensione dell'operando destinazione (2 bit) .  
d. DI ( 2 bit indipendenti ) è un campo di 2 bit che indica o meno la presenza di un displacement o di un dato immediato  
e. MEM indica quali degli operandi (sorgente / destinazione) sono da considerarsi operandi in memoria.

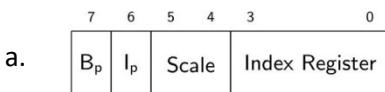
Campo	Valore	Significato
SS	00	La sorgente è un byte
	01	La sorgente è una word
	10	La sorgente è una longword
	11	La sorgente è una quadword
DS	00	La destinazione è un byte
	01	La destinazione è una word
	10	La destinazione è una longword
	11	La destinazione è una quadword
DI	00	Spiazzamento non utilizzato, immediato non presente
	01	Immediato presente
	10	Spiazzamento utilizzato
	11	Spiazzamento utilizzato, immediato presente
Mem	00	Sia la sorgente che la destinazione sono registri
	01	La sorgente è un registro, la destinazione è in memoria
	10	La sorgente è in memoria, la destinazione è un registro
	11	Condizione impossibile (genera un'eccezione a runtime)

- g. Non esiste nessun tipo di configurazione che permette di fare una copia da memoria a memoria . Devo usare quindi un registro intermedio .  
h. Vediamo ora le modalità di indirizzamento dello Z-64 : ricordiamoci che è architettura CISC, ma ci semplifica le cose ( simile ad accesso a vettori) : gli indirizzi non sono numeri di cella ma usa una formula:

$$i. \left[ \begin{array}{c} AX \\ BX \\ CX \\ DX \\ SP \\ BP \\ SI \\ DI \\ R8 \\ R9 \\ R10 \\ R11 \\ R12 \\ R13 \\ R14 \\ R15 \end{array} \right] + \left[ \begin{array}{c} AX \\ BX \\ CX \\ DX \\ SP \\ BP \\ SI \\ DI \\ R8 \\ R9 \\ R10 \\ R11 \\ R12 \\ R13 \\ R14 \\ R15 \end{array} \right] * \left\{ \begin{array}{c} 1 \\ 2 \\ 4 \\ 8 \end{array} \right\} + [spiazzamento]$$

- j. Partiamo dallo spiazzamento : 32 bit (calcolato a tempo scrittura / esecuzione) , il quale viene sommato al resto , mentre per il registro di base (general purpose) viene usato come indirizzo base ( esempio uso registro DX come locazione 0) , infine per quanto lo spiazzamento (come muoverci a celle successive) : in base al tipo di dato si ha la dimensione della cella opportuna ( 1,2,4,8 -> word , long-word, double-word, quad-word). Attenzione al contesto .

### 3. Sib (scale /index/base) :



- b. I primi due bit indicano se la modalità di indirizzamento in memoria usa una base e/o indice  
 c. Se  $I_p=1$  il campo index mantiene la codifica binaria del registro indice. 64 bit e solo scrittura.  
 d. La scala invece vale 1,2,4,8 i quali vengono codificati con 2 bit

i.

1	00
2	01
4	10
8	11

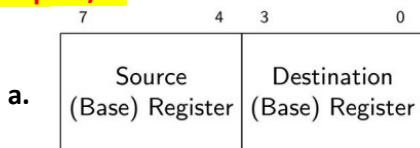
### 4. Codifica registri :

Nome	Codifica	Uso Comune
RAX	0000	Registro Accumulatore
RCX	0001	Registro Contatore
RDX	0010	Registro Dati
RBX	0011	Registro Base
RSP	0100	Stack Pointer
RBP	0101	Base Pointer
RSI	0110	Registro Sorgente
RDI	0111	Registro Destinazione
R8	1000	Registro di uso generale
R9	1001	Registro di uso generale
R10	1010	Registro di uso generale
R11	1011	Registro di uso generale
R12	1100	Registro di uso generale
R13	1101	Registro di uso generale
R14	1110	Registro di uso generale
R15	1111	Registro di uso generale

- b. Da R8 a R15 possono essere usati per qualunque cosa  
 c. RAX si usa come accumulatore , ci salvo le operazioni : registro usato come destinazione.  
 d. RCX si usa come contatore : conta le istruzioni che verranno ciclate .  
 e. RDX : interazione con periferiche I/O.  
 f. RBX: registro base : general purpose  
 g. RSP : stack pointer  
 h. RBP : base pointer  
 i. RSI : sorgente : opera su stringa ( buffer di memoria )

j. RDI : destinazione : opera su stringa ( buffer di memoria )

5. **Campo R/M :**



- a. Source (Base) Register
- b. Sorgente sempre RAX (0000)
- c. Entrambi i campi sono codificati due registri
- d. L'interpretazione di questi due campi dipende sempre dal contesto : dipende dalla configurazione di mem e da quella di Bp
- e. Possibile quindi interpretare registri come semplici registri oppure come registri base.