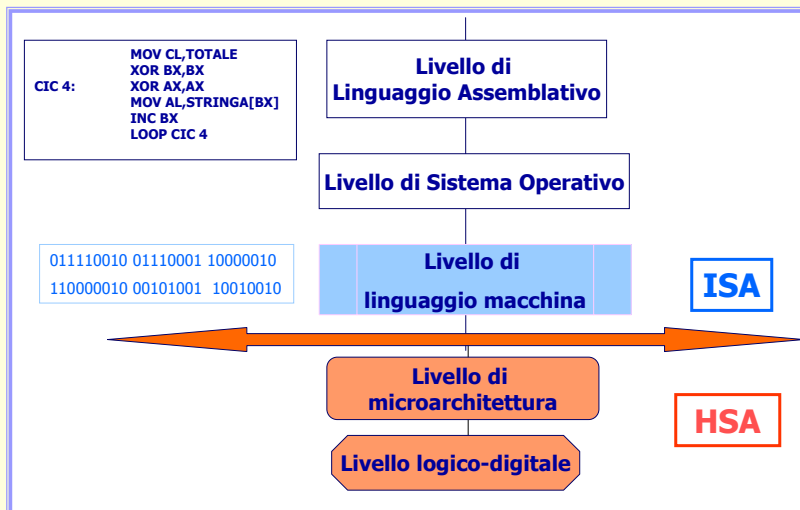


Livello di Linguaggio Macchina

❖ ISA (Instruction Set Architecture)

- Tipi di dati
- Insiemi di registri
- Insiemi di istruzioni

Livello di linguaggio macchina

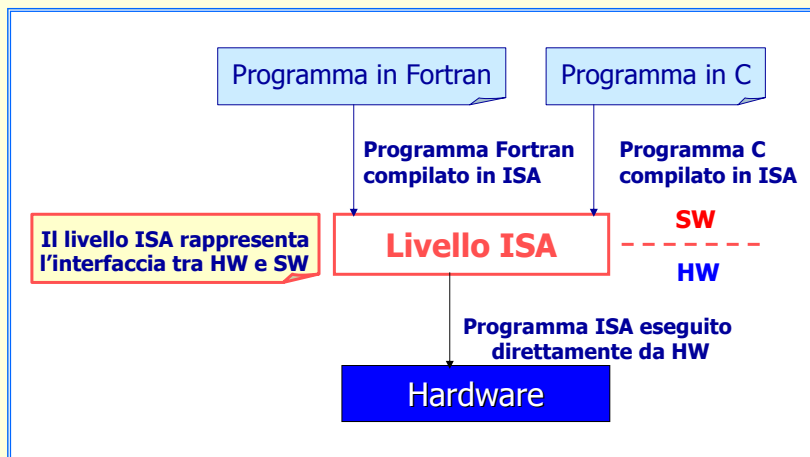


Livello di linguaggio macchina

□ Instruction Set Architecture (ISA)

- ❖ Le caratteristiche computazionali di un elaboratore dipendono in gran parte dal livello ISA.
 - Un elaboratore è in genere visto in termini di ISA: programmatori e compilatori non vedono differenze tra macchine aventi la stessa ISA ma differenti implementazioni.
- ❖ Famiglia di elaboratori
 - Insieme di implementazioni della stessa ISA.
 - IBM PC → elaboratori con stessa ISA ma con implementazioni differenti.

ISA



ISA

- ❖ Per definire il livello ISA occorre specificare come il programmatore a livello di **linguaggio macchina** interagisce con il livello HW.

- ❖ **Parametri che definiscono il livello ISA**

- **Tipi di istruzioni**
 - Quante e quali istruzioni sono disponibili.
- **Tipi di dati**
 - I vari tipi di dati sui quali vengono eseguite le operazioni.
- **Registri**
 - I registri della CPU referenziabili da parte delle istruzioni, e il loro uso.
- **Modello di memoria**
 - Quale è la organizzazione della memoria.
- **Riferimento ai dati**
 - I modi per specificare gli indirizzi degli operandi.

ISA

- ❖ **Concetti generali**

- Concetti indipendenti dalla macchina specifica e applicabili a tutti i tipi di elaboratori.



ISA

1. **Tipi di dati**
 - a. Rappresentazione dei dati
 - b. Precisione dei dati
 - c. Riferimento ai dati (tipo di accesso)

2. **Insiemi di registri**
 - a. Registri dedicati
 - b. Registri operazionali

3. **Insiemi di istruzioni**
 - a. Modalità di indirizzamento
 - b. Tipi di istruzioni

1. Tipi di dati

- **Dato**
 - ❖ Informazione codificata in modo da poter essere trattata dall'elaboratore.
 - ❖ I dati sono definiti da "**valore**", "**indirizzo**", "**tipo**"

- **Tipi di dati (Datatype)**
 - ❖ Insieme di valori e operazioni definite su essi.
 - Datatype Integer: set di interi con le operazioni +, -, *, /;
 - Datatype String: sequenza di caratteri alfabetici con operazioni di concatenazione, selezione,.....

Scelta architetturale: i tipi di dati che sono supportati dall' HW.

1. Tipi di Dati

- ❖ Ad ogni livello dell'elaboratore (linguaggio) corrisponde un insieme diverso di dati.
 - I dati a livello di linguaggio macchina sono diversi dai dati a livelli superiori.
 - Dato "indirizzo di memoria"
 - Essenziale a livello di linguaggio macchina
 - Assente in linguaggi ad alto livello
 - Dato "record"
 - Assente in linguaggio macchina
 - Presente in linguaggio ad alto livello

1.a Rappresentazione di Dati

- **Rappresentazione di un dato**
 - ❖ Ogni dato è rappresentato mediante **unità di informazione** (Byte, word, doubleword, quad-word).
- ❖ **Tipi di dati**
 - Tipi di dati numerici
 - Numeri interi o a virgola fissa: Codice BCD (Binary Coded Decimal), Sistema binario.
 - Numeri floating point: rappresentazione con notazione scientifica.
 - Tipi di dati non numerici: caratteri.
 - Codice ASCII (American Standard Code for Interchange Information) in PC e minicomputers;
 - Codice EBCDIC (Extended Binary Codes Decimal Interchange Code) in Mainframe IBM.
 - Tipi di dati logici

1.b Precisione di Dati

□ Precisione (range)

- ❖ Numero di bit utilizzati per rappresentare un tipo di dati.
 - Valore massimo e minimo rappresentabile.

- Single precision
 - Una word per rappresentare un dato.
- Double precision
 - Due word per rappresentare un dato.

Tecniche per generare precisione variabile o aumentare il range del tipo di dati

1.c Riferimento ai Dati

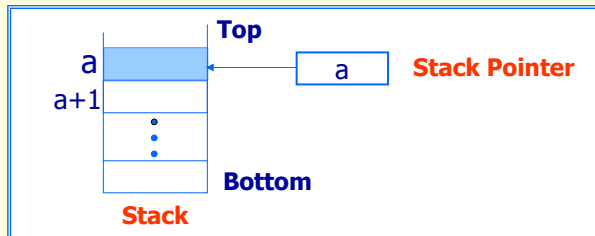
□ Riferimento a un dato

- ❖ Specificazione della struttura fisica che supporta il dato.
 - Strutture fisiche
 1. **Memoria**
 - **Riferimento esplicito** all'indirizzo di una locazione di memoria.
 2. **Registro**
 - **Riferimento esplicito** al numero di registro.
 3. **Stack**
 - **Riferimento implicito** ad un registro (stack pointer) che contiene l'address del top dello stack

1.c Riferimento ai Dati

❖ Stack

- Insieme di registri con organizzazione **LIFO** (Last In First Out).



ISA

1. Tipi di dati

- a. Rappresentazione dei dati
- b. Precisione dei dati
- c. Riferimento ai dati (tipo di accesso)

2. Insiemi di registri

- a. Registri dedicati
- b. Registri operazionali

3. Insiemi di istruzioni

- a. Modalità di indirizzamento
- b. Tipi di istruzioni

2. Insiemi di registri

1. Registri non operazionali (dedicati)

- ❖ Registri che non possono essere manipolati dal programmatore.
 - PC (Program Counter), IR (Instruction Register), MAR (Memory Address Register), PSW (Program Status Word).....

2. Registri operazionali (generali)

- ❖ Registri controllati dal programmatore per conservare e manipolare dati (operandi e risultati).
 - Tutte le operazioni sono eseguite sui dati contenuti nei registri interni al processore.
- ❖ Classificazione delle architetture in base al numero di registri operazionali in essa presenti.
 - **Architettura ad accumulatore**
 - **Architettura a stack**
 - **Architettura a registri**

2.b Registri Operazionali

Architettura ad accumulatore

- ❖ Macchina che usa un singolo registro operazionale detto **accumulatore** per conservare il risultato di una operazione.
 - Il riferimento all'accumulatore è implicito.
 - Un operando è nell'accumulatore.
- Procedure per accedere esplicitamente ai dati
 - **Load** (caricamento del dato da memoria ad accumulatore)
 - **Store** (caricamento del dato da accumulatore a memoria)

$C=A+B \rightarrow$	Load A
	Add B
	Store C

2.b Registri Operazionali

□ Architettura a stack

- ❖ Macchina che usa lo **stack** per conservare gli operandi e il risultato di un'operazione.
 - Il riferimento agli operandi è implicito.
- Procedure per accedere esplicitamente ai dati.
 - **Push** : Caricamento del dato dalla memoria allo stack.
 - **Pop** : caricamento del dato dallo stack alla memoria.

$C=A+B \rightarrow$	Push A
	Push B
	Add
	Pop C

2.b Registri Operazionali

□ Architettura a registri general purpose

- ❖ Macchina con **set di registri** che possono essere usati come accumulatori, registri di indirizzo, stack, registri operazionali.
 - Il riferimento ai registri contenenti gli operandi deve essere specificato nella istruzione.
- Procedure per accedere ai dati:
 - **Load** caricamento del dato dalla memoria in un registro;
 - **Store** caricamento del dato da un registro alla memoria.

$C=A+B \rightarrow$	Load R1, A
	Load R2, B
	Add R1, R2
	Store C, R2

2.b Registri Operazionali

- ❖ Notazione **(m,n)** per le architetture a registri general purpose.
 - **m** numero di operandi in memoria
 - **n** numero massimo di operandi in una istruzione

- ❖ Tipi di architetture a registri generali:
 - Registro-Registro (0,3)
 - Registro-Memoria (1,2)
 - Memoria-Memoria (3,3)

2.b Registri Operazionali

- ❑ **Architettura a registri special purpose**
 - ❖ Macchina con set di registri le cui funzioni sono predefinite.

- ❑ **Architettura a registri di uso generale e dedicato**
 - ❖ Processori Intel 80x86 (Personal Computer).

ISA

1. Tipi di dati

- a. Rappresentazione dei dati
- b. Precisione dei dati
- c. Riferimento ai dati (tipo di accesso)

2. Insiemi di registri

- a. Registri dedicati
- b. Registri operazionali

3. Insiemi di istruzioni

- a. Formato delle istruzioni
- b. Modalità di indirizzamento
- c. Tipi di istruzioni

3. Insiemi di istruzioni

□ Set di istruzioni

- ❖ Parte visibile al programmatore e al progettista di compilatori.

❖ Istruzione

- Parola del linguaggio macchina

❖ Insieme delle istruzioni

- Vocabolario del linguaggio macchina

❖ Formato di istruzione

- Sintassi della parola

Vari progetti possibili per l'insieme delle istruzioni

3. Insiemi di istruzioni

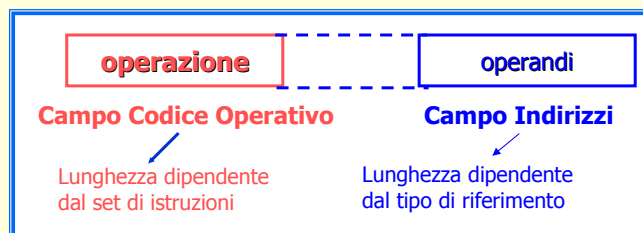
❖ Caratteristiche del set di istruzioni

- **Completezza**
 - Il set di istruzioni deve includere tutte le operazioni di base per i tipi di dati presenti nella ISA.
- **Ortogonalità**
 - Il set di istruzioni deve essere non ridondante, ovvero deve esserci un solo modo per effettuare una operazione.
- **Compatibilità**
 - Una famiglia di elaboratori deve poter utilizzare gli stessi programmi.

3. Insiemi di istruzioni

❑ Formato delle istruzioni

- ❖ Il formato di istruzione si riferisce al **modo in cui è rappresentata una istruzione** e sono codificati il codice operativo e gli operandi.
- ❖ Il formato di istruzione specifica: operazione da eseguire, numero di operandi, tipo e dimensione degli operandi, collocazione degli operandi.

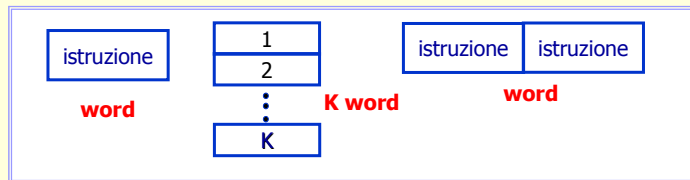


3. Insiemi di istruzioni

- ❖ Il formato dipende dalla natura della istruzione e dalla scelta architetturale.

$A \leftarrow (A) + (B)$ operazione diadica
 $AC \leftarrow (AC) + (A)$ operazione monadica

- ❖ La lunghezza d'istruzione è un multiplo o sottomultiplo della lunghezza di parola.
 - ❖ Diversi formati di istruzione → diverse lunghezze di istruzione



3. Insiemi di istruzioni

❖ Lunghezza di istruzione

- Scelta architetturale condizionata dalla dimensione ed organizzazione della memoria, dalla struttura del bus, dalla complessità della CPU.
- Istruzioni brevi rispetto alla larghezza di banda della memoria
 - Minore occupazione di memoria
 - Maggiore velocità di trasferimento e quindi di esecuzione delle istruzioni
 - Difficile la decodifica e la esecuzione concorrente
- Istruzioni lunghe rispetto alla larghezza di banda della memoria
 - Maggior occupazione di memoria
 - Complessità maggiore per l'unità di controllo
 - Tempo di decodifica assente

3. Insiemi di istruzioni

❖ Formato a lunghezza fissa

- Per ogni istruzione una parola sufficientemente lunga da ospitare tutti i formati.

❖ Formato a lunghezza variabile

- Per ogni istruzione una parola di lunghezza variabile.

❖ Formato ortogonale

- Ogni istruzione contiene un campo **modo** per specificare le informazioni.



3. Insiemi di istruzioni

❖ Formato non ortogonale

- Per ogni operazione esistono **diversi codici operativi** a seconda del numero di operandi e della loro lunghezza.

ADDB2 Op1, Op2:	$Op1 \leftarrow (Op1) + (Op2)$
ADDB3 Op1, Op2, Op3	$Op3 \leftarrow (Op1) + (Op2) + (Op3)$
ADDW2 Op1, Op2	$Op1 \leftarrow (Op1) + (Op2)$
B=Byte	W=Word

- In **linguaggio macchina** per una stessa operazione occorre utilizzare istruzioni diverse per diversi tipi di dati.

- In **linguaggio ad alto livello** per una operazione esiste un solo codice operativo.

- L'informazione sul **"tipo"** di dati è attaccata al dato stesso

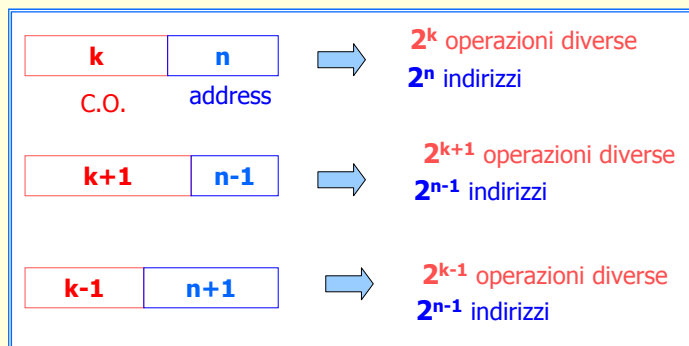
3. Insiemi di istruzioni

❖ Lunghezza del codice operativo



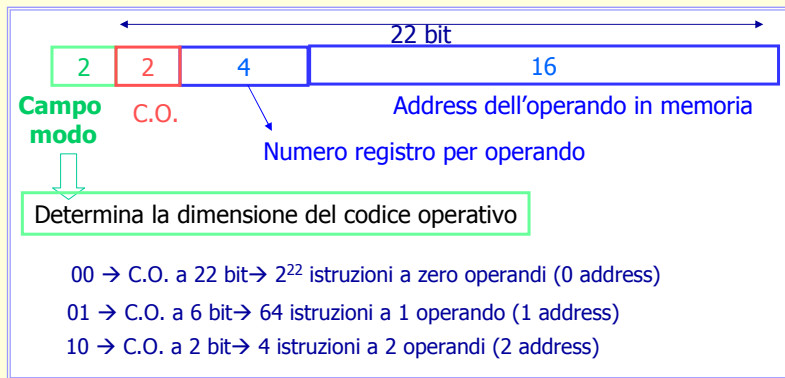
3. Insiemi di istruzioni

❖ Codice operativo a lunghezza variabile in formati di istruzione a lunghezza fissa



3. Insiemi di istruzioni

❖ Codice operativo a lunghezza variabile in formati di istruzione a lunghezza fissa



3. Insiemi di istruzioni

❑ Codice operativo a lunghezza variabile

❖ Codice di Huffman

- Codice dipendente dalla frequenza delle parole che deve rappresentare.
 - Frequenza alta → parola codice breve
 - Frequenza bassa → parola codice lunga

2^3 codici operativi → C.O. a lunghezza fissa $L=3$ bit
 2^3 codici operativi con frequenza f_i → 2^3 parole codice di lunghezza L_i

Codice operativo di lunghezza media $\bar{L} = \sum_{i=1}^{2^3} f_i L_i$