Elaborazione dell'informazione

- Una elaborazione è una trasformazione di dati
- In generale si può indicare che una elaborazione è la trasformazione:

$$Y=F(X)$$

- dove:
  - X è l'insieme di dati iniziali o di "ingresso"
  - Y è l'insieme di dati finali o di "uscita"
  - F è una *regola* che fa corrispondere Y ad X
- La decomposizione dell'elaborazione in azioni componenti ci porta a costruire l'*algoritmo di calcolo*, detto anche *processo di elaborazione*

Algoritmo

- *L' algoritmo è una **sequenza finita** di azioni elaborative* (o di "passi di elaborazione") *che risolve **automaticamente** un problema*
  - la semplicità o la complessità di una azione elaborativa dipende dall'ESECUTORE
  - la trasformazione F più che una regola, in generale, è un procedimento costituito da un insieme di azioni elaborative che vanno eseguite secondo un definito ordine ...

Algoritmo (2)

- *L' algoritmo è una **sequenza finita** di azioni elaborative* (o di "passi di elaborazione") che risolve ***automaticamente*** un problema
  - **Sequenza**: le azioni elaborative vengono eseguite una alla volta secondo un ordine stabilito
  - **Finita**: con riferimento allo spazio e al tempo;
    spazio: numero finito di azioni elaborative (passi)
    tempo: ogni azione elaborativa si realizza in un tempo finito
  - **Automaticamente**:
    - esiste una macchina che può eseguirlo (esecutore)
    - la macchina una volta avviata è in grado di evolvere da sola
  - il procedimento descritto dall'algoritmo è DETERMINISTICO: la sequenza è rigidamente fissata e niente è lasciato al caso

Formalizzazione di un algoritmo

- Richiede che venga definito:
  - L'insieme di tutte le possibili azioni elaborative
  - Una macchina in grado di eseguirle
  - Un *linguaggio* attraverso il quale descrivere ciascuna delle azioni e la sequenza di esse

Linguaggio e programma

- Un algoritmo viene descritto tramite un LINGUAGGIO
- Il linguaggio di descrizione di un algoritmo deve essere comprensibile all'esecutore, ovvero alla macchina automatica che lo esegue
- Il linguaggio deve essere 'non ambiguo', ovvero ciascuna frase costruita con esso deve evocare un'unica azione elaborativa
- Un PROGRAMMA è la descrizione formalizzata di un algoritmo, espressa in un linguaggio di programmazione
- Un PROGRAMMA è una sequenza di 'frasi' (*istruzioni*) ciascuna esprimente operazioni che l'esecutore può comprendere ed eseguire
- Il PROGRAMMA che deve essere eseguito è memorizzato nella memoria dell'esecutore

Macchine per l'elaborazione

- Macchine diverse potrebbero avere ***diversa capacità di risolvere problemi***
- Se neanche la macchina "più potente" riesce a risolvere un dato problema, esso **potrebbe essere *non risolubile***

Gerarchia di macchine

- macchine base (combinatorie)
- macchine (automi) a stati finiti
- ….
- **macchina di Turing**

Macchina base

- E' definita dalla tripla <**I**,O,mfn> dove:
  - **I** = insieme finito dei **simboli di ingresso**
  - **O** = insieme finito dei **simboli di uscita**
  - **mfn: I→O** *(funzione di macchina)*

- Esempio: le *porte logiche* e le *funzioni* in genere

Macchina base (2)

- *Risolvere problemi* con la macchina base comporta ***enumerare in modo esplicito*** *tutte le possibili configurazioni* d'ingresso, e indicare in corrispondenza il valore di uscita
- Limite computazionale: è un dispositivo *puramente combinatorio*, quindi ***inadatto a risolvere problemi che richiedono una memoria interna*** (riconoscimento di sequenze, somme di numeri forniti in successione, ecc.)

Automa a Stati Finiti (ASF)

- E’ una prima astrazione di macchina “dotata di memoria” che esegue algoritmi
- Introduce il concetto fondamentale di **“STATO**”che informalmente può essere definito come una particolare condizione della macchina, in conseguenza del quale la macchina reagisce con una determinata “uscita” ad un determinato “ingresso”
- Poiché l’uscita dipende anche dallo *stato*, l’ASF è un automa intrinsecamente dotato di una *memoria interna* che può quindi *influenzare le risposte date dall’automa* anche a parità di dati d’ingresso
- Esempio: riconoscitore di sequenza

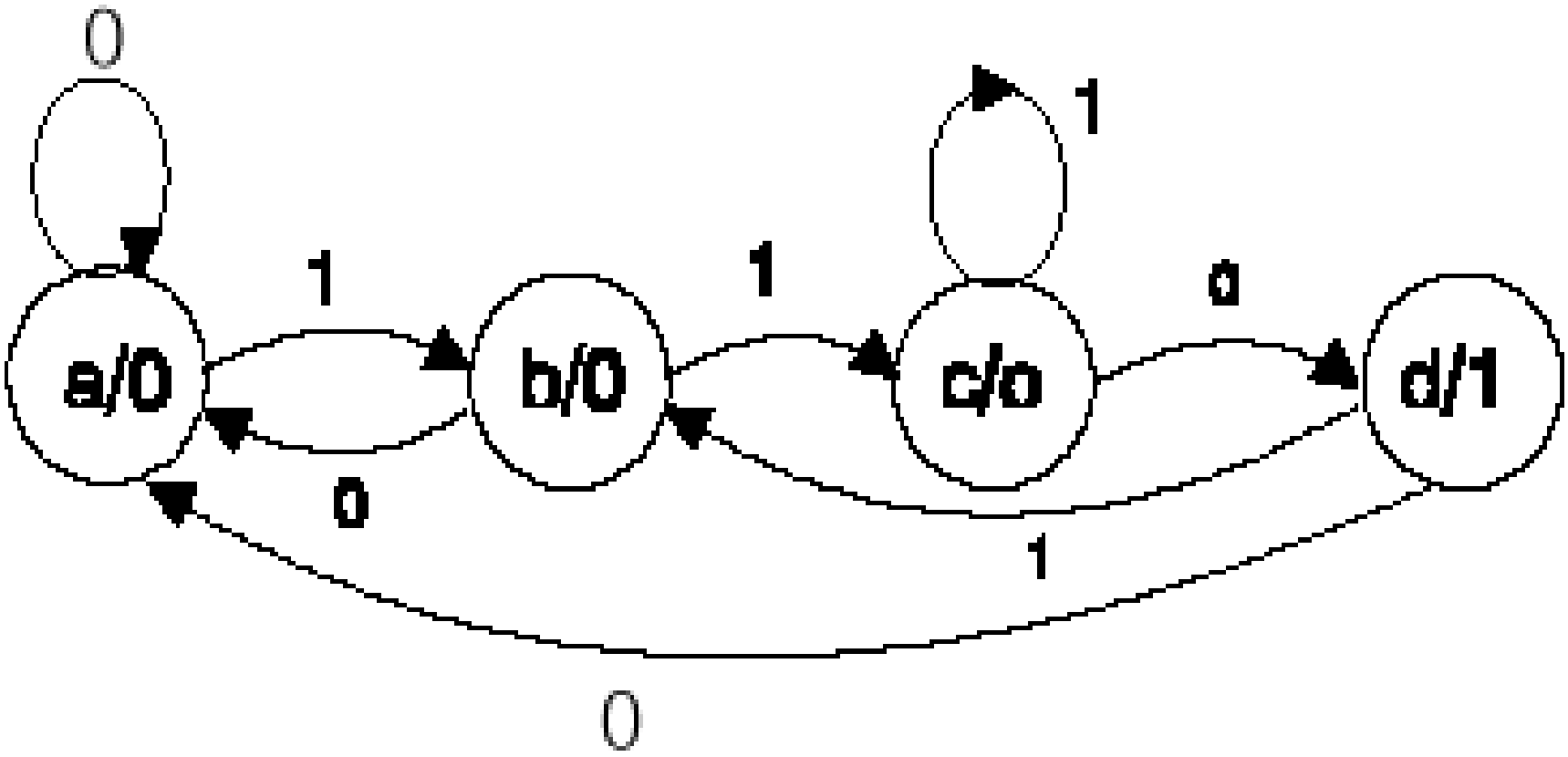
Modello di Automa a Stati Finiti

- Un ASF è una quintupla <Q, I, U, t, w> dove:
  - Q: insieme finito di stati interni — q: $q \in Q$
  - I: insieme finito di ingressi — i: $i \in I$
  - U: insieme finito di uscite — $u \in U$
  - t: funzione di transizione — $t: Q \times I \rightarrow Q$
  - w: funzione di uscita — $w: Q \times I \rightarrow U$
- Problema della tempificazione: in corrispondenza di quali eventi avvengono le transizioni di stato ?

Rappresentazione grafica di un ASF

- E’ possibile rappresentare graficamente un ASF mediante un ***grafo*** detto *diagramma degli stati*
- *Stato: rappresentato da un nodo (cerchio)*
- *Transizione: rappresentata da un arco orientato (freccia)*
- *Ciascun arco viene etichettato con l’ingresso che causa la transizione e la conseguente uscita, separati da un simbolo (/)*
- *Se l’uscita non è specificata, può essere indicata con il simbolo “-”*

Riconoscitore della sequenza 110



ASF come macchina elaboratrice

- Un ASF è una "macchina elaboratrice" con
  - un insieme finito di stati
    - possibili configurazioni interne
    - o modalità ingresso/uscita
  - input, sequenza (finita) di simboli
  - output, sequenza (finita) di simboli
- Elaborazione
  - si parte da uno stato iniziale
  - si consuma un simbolo di input
  - sulla base di input e stato corrente si produce un simbolo di output e ci si muove su un nuovo stato
  - si procede finchè non si raggiunge uno stato finale
- Quale "programmazione"?
  - le funzioni t,w

Limiti di un ASF

- **Gli ASF hanno un potere espressivo limitato.** Con gli ASF si riesce a "realizzare":
  - semplici trasformazioni di sequenze
  - riconoscitori di linguaggi regolari

- Da cosa deriva la mancanza di espressività?
- il numero di stati è fissato a priori non consente di trattare quei problemi che richiedono un "conteggio" di elementi senza limite fissato
- **In altre parole hanno un limite computazionale: sono dispositivi *a memoria finita,* quindi *inadatti a risolvere quei problemi che non consentono di limitare a priori la lunghezza delle sequenze d'ingresso di cui tenere memoria.***
- Inoltre il modello di automa non precisa la natura degli ingressi e delle uscite e quindi non aiuta ad individuare un insieme di possibili azioni elaborative

La Macchina di Turing

- La Macchina di Turing (1936) è un modello fondamentale nella teoria dell'informatica
- Consente di approfondire il concetto di algoritmo
- Ha consentito di ottenere importanti risultati relativamente alla calcolabilità, alla complessità ed alla equivalenza di algoritmi

La Macchina di Turing

- Un particolare automa composto da:
  - Un ideale nastro (memoria), cioè una striscia continua di celle ciascuna delle quali può essere vuota o contenere i simboli di un alfabeto
  - Una testina di lettura-scrittura
- La capacità di compiere le seguenti azione elaborative:
  - Spostamento testina di una posizione (left-right)
  - Scrittura di un simbolo nella cella sotto la testina (sostituendo il contenuto precedente)
- **Un dispositivo di controllo** che in ogni istante si trova in uno tra un numero finito di stati e che per ciascuna coppia (stato-simbolo letto):
  - Cambia stato
  - Esegue una delle azioni elaborative

Elaborazione con la MdT

- si parte da uno stato iniziale
- sul nastro è scritto il valore dell'input
- si legge il simbolo sotto la testina
- sulla base di (input,stato) si ottiene: valore da scrivere sul nastro, nuovo stato, movimento LEFT/RIGHT
- si procede finché non si raggiunge uno stato finale
- Il nastro come luogo dove
  - trasformare l'input in output
  - scrivere valori d'appoggio, risultati intermedi

MdT e Algoritmo

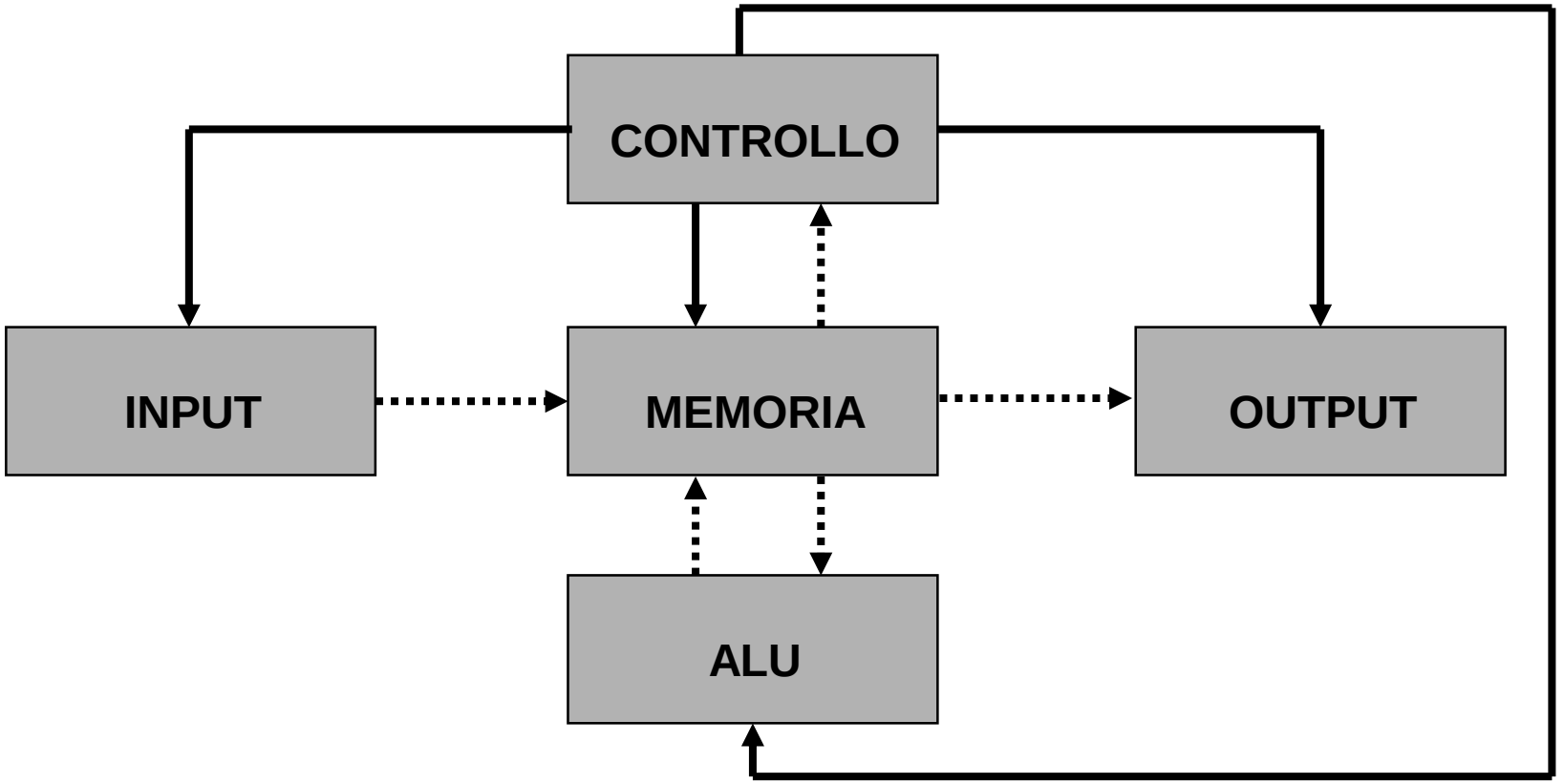
- Una macchina di Turing che si arresti e trasformi un nastro t in uno t' rappresenta l'algoritmo per l'elaborazione Y=F(X), ove X,Y sono rispettivamente **codificati** in t, t'
- Data una funzione Y=F(X) esiste sempre una MdT in grado di calcolarla? La risposta è che ***esistono funzioni non calcolabili***
- **TESI di Church**: Non esiste un formalismo o una macchina concreta che possa calcolare una funzione non calcolabile secondo Turing
  - nessuno è mai riuscito a dimostrare il contrario …
- In altre parole …
  UN ALGORITMO E' CIO' CHE PUO' ESSERE REALIZZATO CON UNA MACCHINA DI TURING

Macchina di Turing

- **Risolvere un problema con la MdT** richiede quindi di:
  - **definire** una opportuna ***rappresentazione dei dati iniziali sul nastro***
  - **definire la *parte di controllo***, in modo da **rendere disponibile sul nastro**, ***alla fine*****,** la ***rappresentazione della soluzione***
- In altre parole l'algoritmo è "cablato" nel dispositivo di controllo
- Una volta definita la parte di controllo, **la MdT è capace di risolvere un dato problema (risolubile)** ma così facendo, **essa è *specifica per quel problema***

Modello di Von Neumann

Modello concettuale di macchina per l'elaborazione delle informazioni a cui si ispirano i calcolatori elettronici



Il modello di macchina di Von Neumann

- **Unità di Ingresso per l'acquisizione dei dati e dei programmi e per il loro trasferimento in memoria**
- **Unità di Memoria per la registrazione sia dei dati che delle istruzioni del programma**
- **Unità di Controllo presiede a tutte le operazioni, interpreta le istruzioni prelevate dalla memoria e ne guida l'esecuzione inviando appositi segnali alle altre unità**
- **Unità Aritmetico-Logica (ALU), dedicata alla esecuzione delle operazioni aritmetiche e logiche**
- **Unità di Uscita per il trasferimento all'esterno dei risultati presenti in memoria**

Modello di Von Neumann: caratteristiche (1)

- Elaborazione sequenziale
- Linguaggio imperativo
- Memoria comune per Dati ed Istruzioni
- Macchina a programma registrato

Modello di Von Neumann: caratteristiche (2)

- Il processore (o unità di controllo) nel modello di Von Neumann è la macchina che esegue il processo di elaborazione
- Il processo (algoritmo) è formalmente descritto mediante un *programma*, registrato nella memoria dell'elaboratore
- Un programma è formalmente descritto mediante un *linguaggio* noto al processore

L’elaborazione nella macchina di Von Neumann

- Il programma ed i dati su cui esso opera devono trovarsi in memoria
- Il programma viene prelevato dall’unità di ingresso e registrato in memoria (*fase di caricamento o loading*)
- La lettura dei dati su cui opera è una delle operazioni prevista dal programma, pertanto la registrazione in memoria dei dati avviene nella fase successiva
- Le azioni elaborative descritte dal programma vengono eseguite dall’unità di controllo con l’ausilio delle altre unità (*fase di esecuzione o running*)

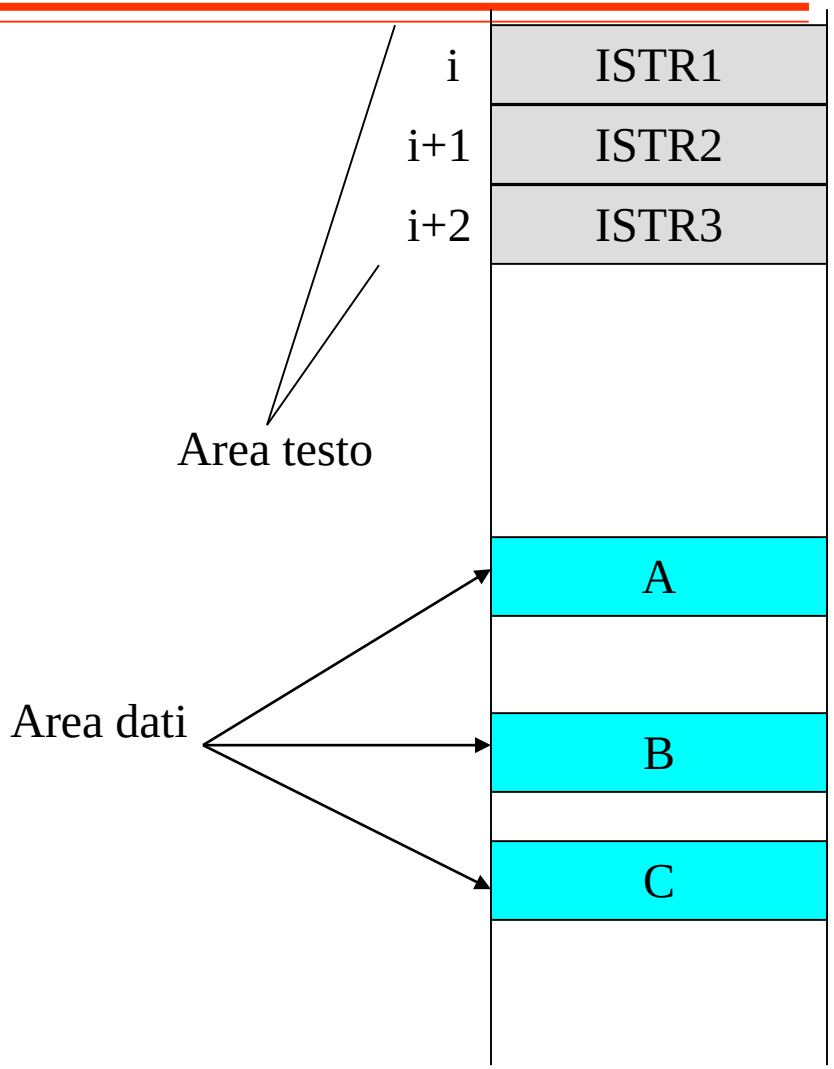
Elaborazione nella Macchina di VN

- Stato dei dati
  - (A,V) con: $A=\{a_1,a_2,\ldots,a_n\}$, $V=\{v_1,v_2,\ldots,v_n\}$
- Azione elaborativa
  - Alterazione dello stato dei dati
- Assegnazione
  - Azione elaborativa che dà un valore ad un attributo
- Processore
  - Macchina che esegue il processo di elaborazione
- Programma
  - Rappresentazione di un processo di elaborazione

Unità di controllo

- Opera in un ciclo infinito:
  1. Prelievo
  2. Esecuzione

```
while
(TRUE){
    Fetch;
    Execute
 }
```

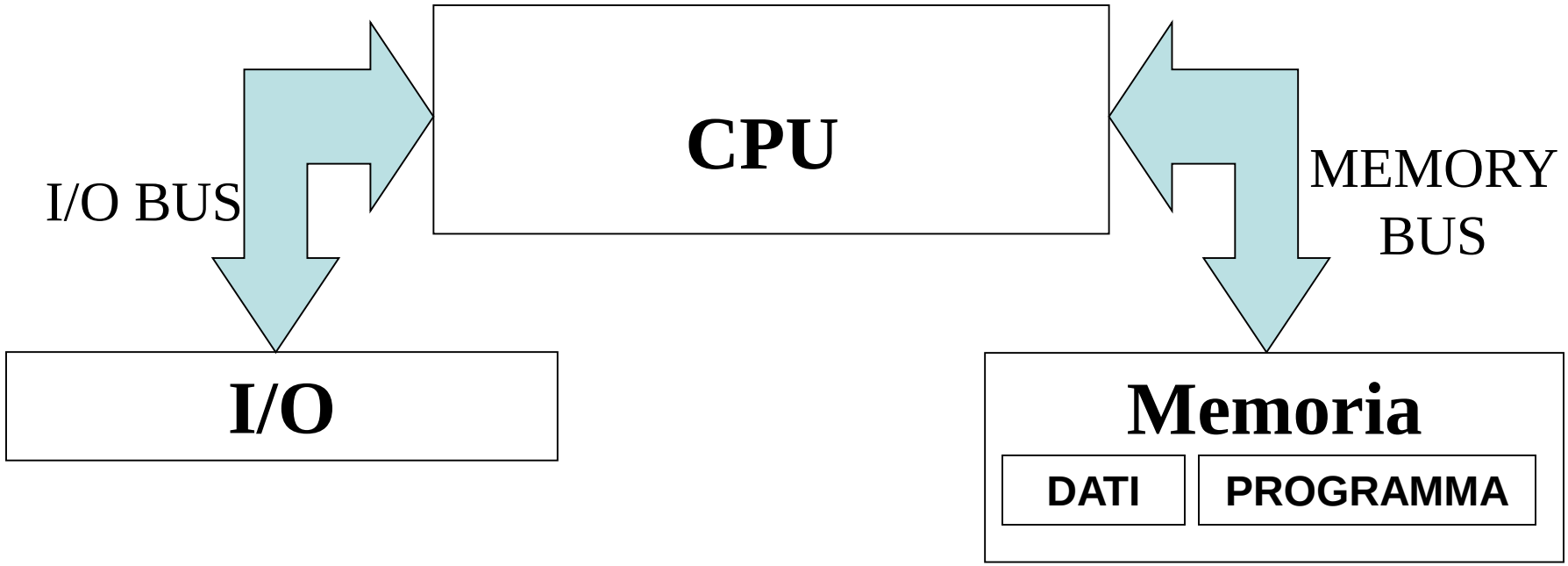


Il calcolatore elettronico

- Un *calcolatore elettronico* è un sistema per l'elaborazione delle informazioni realizzato secondo il modello di von Neumann e dotato delle seguenti caratteristiche:
  - è un sistema *numerico*
  - è un sistema *automatico*
  - è un sistema *a programma registrato*
  - è realizzato mediante circuiti *elettronici*

Il calcolatore elettronico (2)

- L'architettura dei calcolatori elettronici è nata ispirandosi al modello generale di Von Neumann
- Un "processore" (CPU) manipola dati registrati in memoria sotto il controllo di un *programma* registrato anch'esso in memoria



Il calcolatore elettronico (2)

- La capacità elaborativa del calcolatore risiede nel **processore**; il processore è in grado di eseguire un set di azioni elaborative elementari più o meno complesse
- Le **istruzioni** sono comandi espliciti che
  - governano il trasferimento di informazioni sia all'interno del calcolatore sia tra il calcolatore e i dispositivi di I/O
  - specificano le operazioni aritmetiche e logiche che devono essere effettuate
- I dati di ingresso e di uscita dell'elaborazione, nonché la stessa sequenza di istruzioni sono immagazzinati nella memoria centrale
- Il processore preleva ed esegue le istruzioni dalla memoria una ad una
- Una sequenza di istruzioni memorizzate nella memoria centrale costituisce un programma

Calcolatore come “Macchina a strati”

• L’organizzazione “a strati” di un sistema complesso è una soluzione tipica dell’Ingegneria (approccio *divide-et-impera*)

Calcolatore come “Macchina a strati” (2)

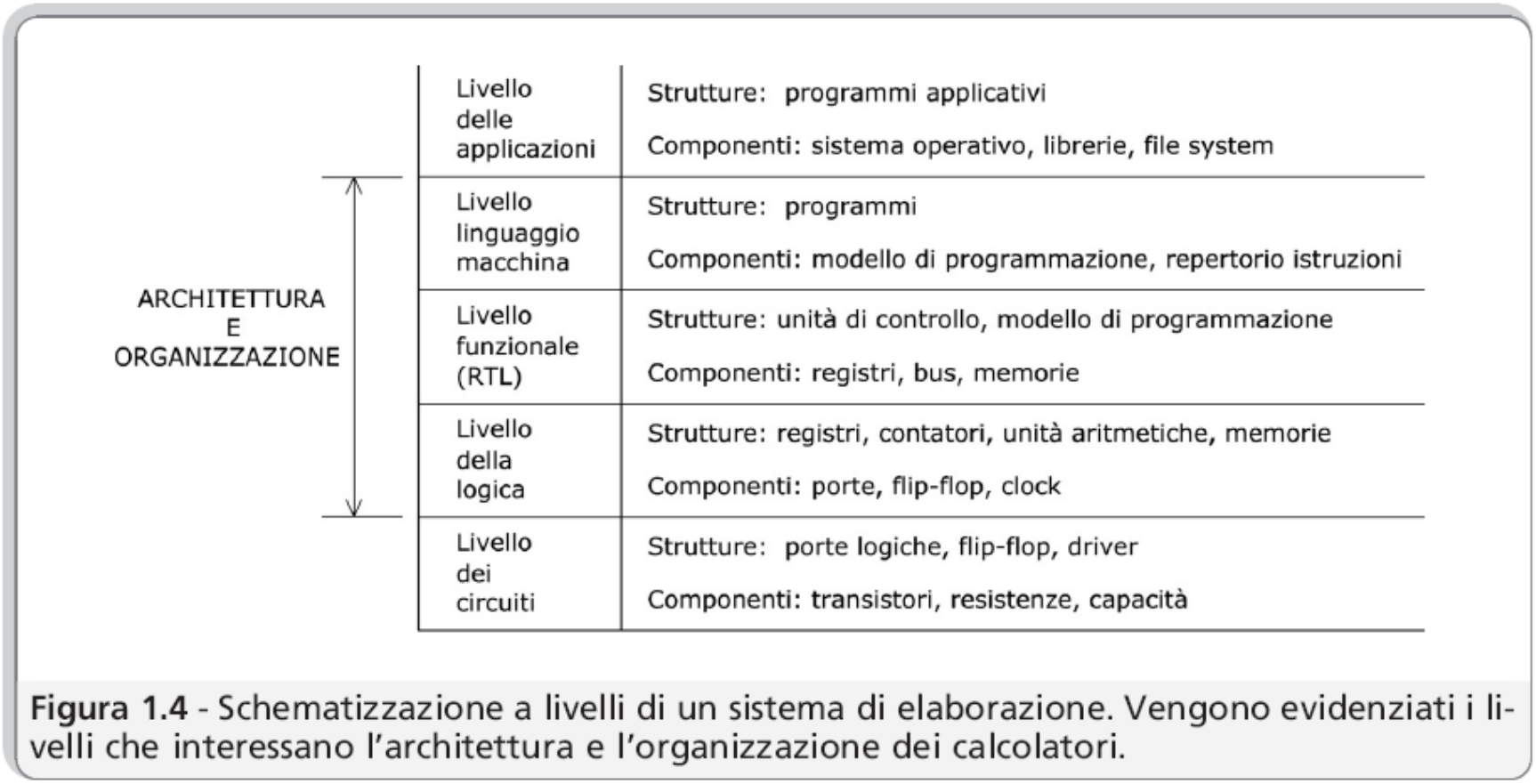
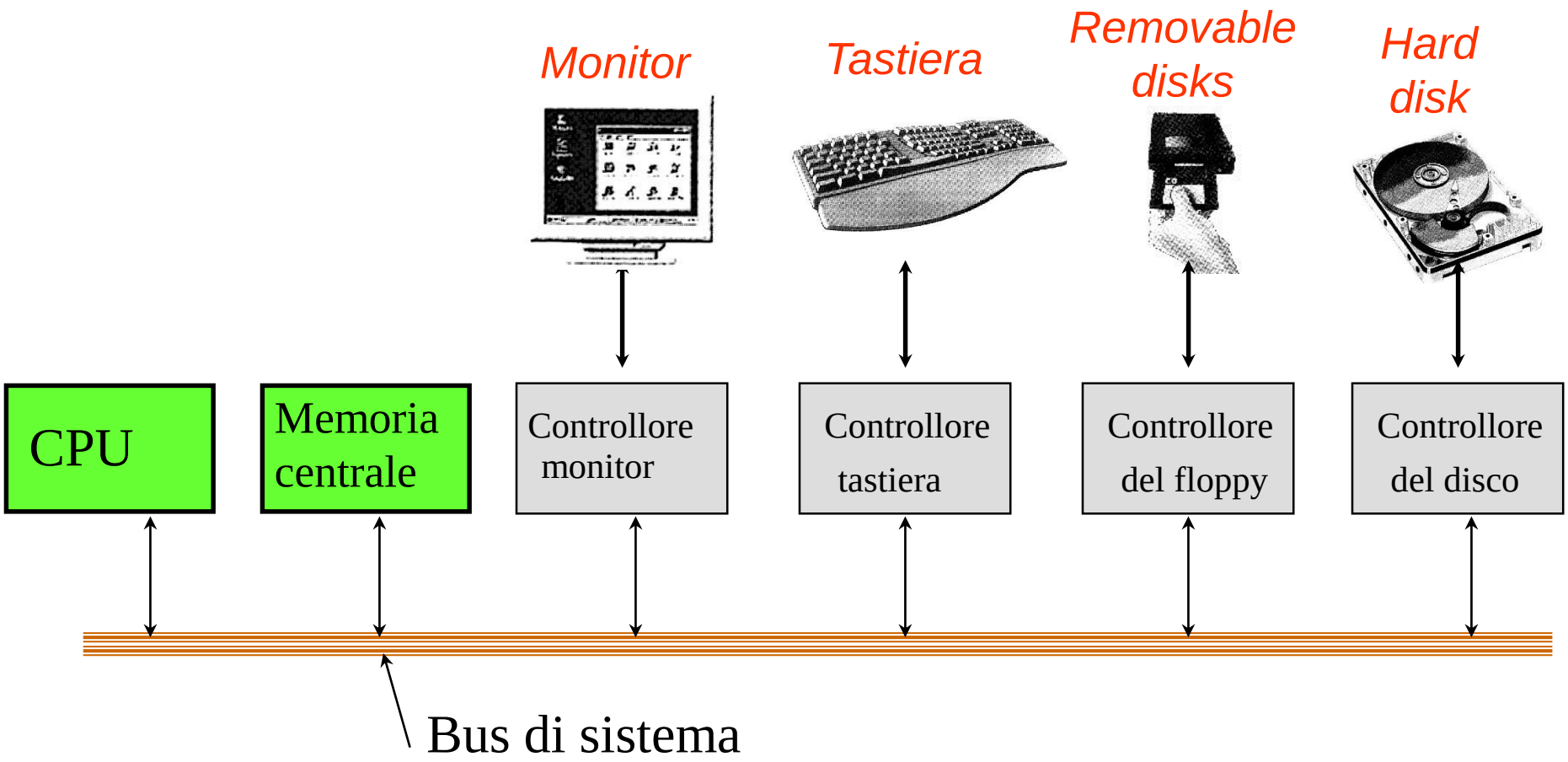


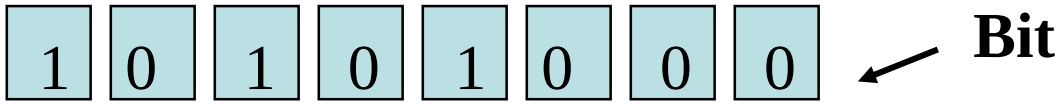
Figura 1.4 - Schematizzazione a livelli di un sistema di elaborazione. Vengono evidenziati i livelli che interessano l’architettura e l’organizzazione dei calcolatori.

Organizzazione di un Computer



Registri

- Il calcolatore opera su informazioni rappresentate in registri
- Il registro è individuato da un “nome”
- E’ un organo fisico k-stabile atto memorizzare k stati distinti
- Il valore di ogni cella di un registro (bit) è mantenuto da un circuito logico detto flip-flop



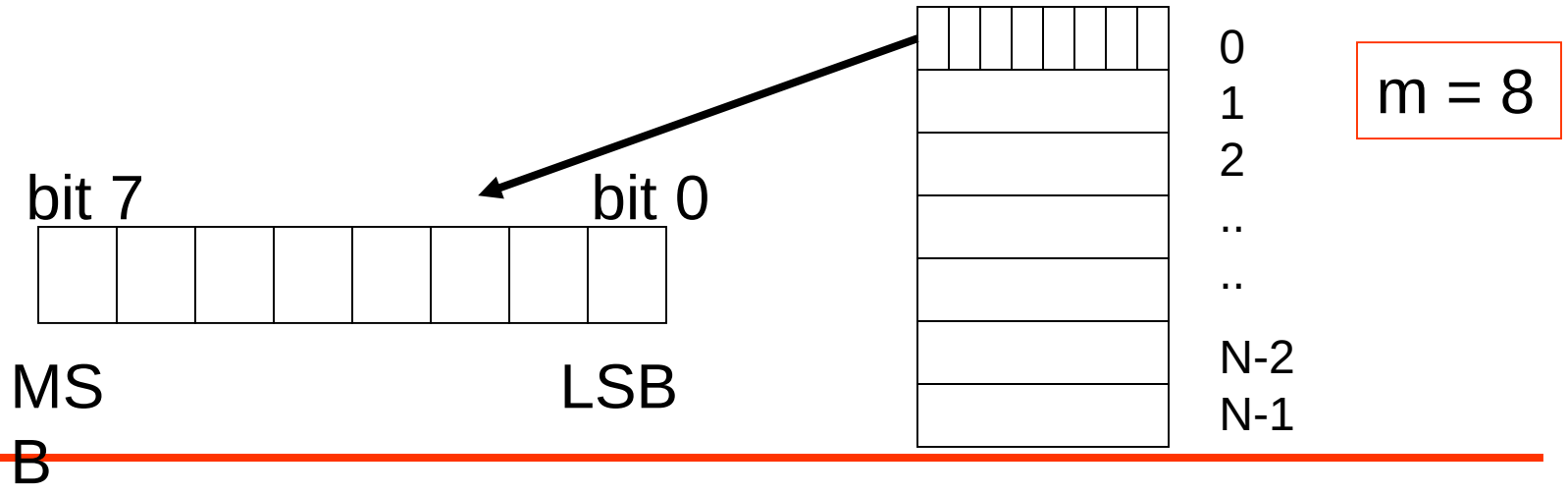
registro

Registri del processore

- **Parallelismo in bit: la lunghezza in bit dei registri della CPU viene anche indicato come numero di bit del processore**
- **Processori a 8 bit (anni ’70/’80)**
  Es. Intel 8080, Zilog Z80, Motorola 6809
- **Processori a 16 bit (anni ’80)**
  Es. Intel 8086/80286, Motorola 68000 (16/32 bit)
- **Processori a 32 bit (anni ’90-inizio anni 2000)**
  Es. Intel 80386, 80486, Pentium, Celeron, Motorola 68030, 68040, PowerPC
- **Processori a 64 bit (anni 2000)**

La memoria centrale

- La memoria centrale di un computer è organizzata come un array di stringhe di bit di lunghezza m, dette *locazioni*
- Gli m bit di una locazione sono accessibili dal processore (in lettura/scrittura) mediante un'unica operazione
- Ogni locazione è individuata da un *indirizzo*, cioè un intero compreso tra 0 e N-1, con $N = 2^c$
  - » [0, N-1] = SPAZIO DI INDIRIZZAMENTO
- La memoria centrale è *ad accesso casuale* (RAM) cioè il tempo di accesso non dipende dalla posizione del dato

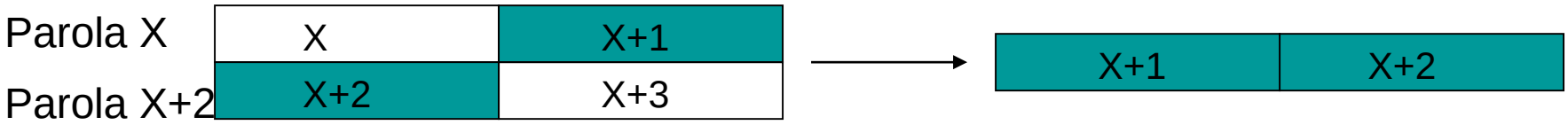


Processori a parola e processori a carattere

- I processori “a parola” hanno la memoria organizzata in locazioni (parole o *word*) di 16 bit, 32 bit o 64 bit
- I processori “a carattere” accedono alla memoria con un parallelismo di 1 byte (8 bit)
- La maggior parte dei sistemi moderni accede alla memoria con un parallelismo di “parole” da 16, 32 o 64 bit, ma l’unità indirizzabile di memoria (locazione) è ancora il byte (sistemi a memoria *byte-addressable*)

Memoria: parole allineate e non

- Per un processore a parola di 16 bit, una *parola* che inizia ad un indirizzo pari si dice “allineata sul limite di parola”
- Tipicamente, un tale processore è in grado di accedere ai due byte che costituiscono una parola allineata mediante una sola operazione di lettura
- Il processore 8086 consente l’utilizzo di parole non allineate, cioè parole che iniziano ad un indirizzo dispari, ma in tal caso sono necessari 2 distinti accessi in memoria
- Il processore 68000 NON consente l’accesso a parole non allineate



(X pari) La parola (X+1) non è allineata sul limite di parola

Interazione processore-memoria