

Che cos'è l'informatica?

scienza della rappresentazione e dell'elaborazione automatica dell'informazione

non solo la tecnologia dei calcolatori

definizione da *Association for Computing Machinery* (ACM)

studio sistematico degli algoritmi

*loro teoria
analisi
progetto
efficienza
realizzazione
applicazione*

algoritmo: sequenza di operazioni

precise

comprensibili ⇒ eseguibili da strumento automatico

NB attività informatiche non necessariamente al calcolatore

e.g., progettare analizzare applicazioni

calcolatore soltanto uno strumento di calcolo potente

⇒ permette gestione quantità di informazioni altrimenti intrattabili

Il concetto di algoritmo

sequenza di passi

definiti con precisione \Rightarrow comprensibile a esecutore meccanico

portano a realizzazione di un compito / soluzione di un problema

Esempi,

istruzioni montaggio di elettrodomestico

calcolo massimo comune divisore di numeri naturali

uso di terminale Bancomat

legame fra algoritmi e calcolatori (elettronici)

\Rightarrow calcolatori elettronici come esecutori di algoritmi

algoritmi descritti tramite programmi

sequenze di istruzioni scritte in opportuno linguaggio

esperto informatico produce algoritmi

capisce sequenza di passi per soluzione problema

codifica in programmi

due proprietà essenziali degli algoritmi

correttezza

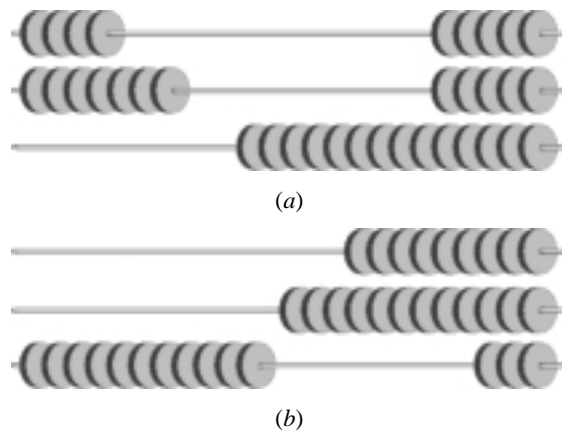
efficienza

corretto \equiv **risolve il compito senza errori o difetti**

efficiente \equiv **usa risorse in modo minimale o almeno ragionevole**

Esempio: Somma di due numeri con un pallottoliere

primo addendo sulla prima riga, secondo sulla seconda riga, risultato sulla terza



Algoritmo:

1. sposta pallina da sinistra a destra nella prima riga, al contempo da destra a sinistra nella terza
2. ripeti operazione precedente fino a svuotare parte sinistra prima riga.
3. sposta pallina da sinistra a destra nella seconda riga, al contempo da destra a sinistra nella terza
4. ripeti operazione precedente fino a svuotare parte sinistra seconda riga.

Esempio: Gestione di una biblioteca

piccola biblioteca con scaffali, su cui disposti i libri.

Ogni libro posto in posizione fissa, con due coordinate: scaffale e posizione nello scaffale

biblioteca dotata di schedario

Ogni scheda contiene, nell'ordine:

- **cognome e nome autore**
(se più autori, riportati tutti cognomi e nomi nello stesso ordine della copertina)
- **titolo del libro**
- **data di pubblicazione**
- **numero dello scaffale in cui si trova**
- **numero d'ordine della posizione nello scaffale**

esempio di scheda

AUTORE / 1: GHEZZI CARLO, JAZAYERI MEHDI.
TITOLO: PROGRAMMING LANGUAGE CONCEPTS. 1981.
SCAFFALE 35 POSIZIONE 21

schede ordine alfabetico in base ad autore e titolo

a pari primo autore dal secondo, etc; libri con stessi autori ordinati in base al titolo

semplice algoritmo per accedere a libro, conoscendone autore e titolo

- 1. ricerca la scheda del libro nello schedario**
- 2. trovata la scheda, segna su un foglietto numero scaffale e posizione del libro**
- 3. ricerca scaffale indicato**
- 4. individuato lo scaffale, accedi alla posizione del libro**

il primo passo dell'algoritmo è il più complesso: precisiamolo

La più semplice tecnica di ricerca nello schedario:

- 1. esamina la prima scheda dello schedario**
- 2. Se autore e titolo coincidono con quelli cercati
allora ricerca conclusa con successo
altrimenti passa a scheda successiva**
- 3. continua di scheda in scheda
se trovata quella cercata ⇒ ricerca conclusa con successo
ee esaurite schede ⇒ ricerca conclusa con insuccesso**

tecnica di ricerca molto semplice ma inefficiente

Algoritmo alternativo, vicino quanto si fa in pratica:

- 1. prendi scheda centrale dello schedario**
- 2. se è la scheda cercata la ricerca ha termine con successo**
- 3. altrimenti, se la scheda cercata segue alfabeticamente quella esaminata,
continua la ricerca nella seconda metà dello schedario
altrimenti continua la ricerca nella prima metà**

questo algoritmo è incompleto: ricerca termina anche se zona di ricerca vuota

passo 1. precisato

- 1. se schedario vuoto
allora termina con insuccesso
altrimenti prendi scheda centrale dello schedario**

seconda tecnica di ricerca è più efficiente della prima

I linguaggi per la programmazione di algoritmi

In soli cinquant'anni informatici hanno creato una babele informatica

pro: opportunità di specializzazione
contro: problemi di comunicazione e compatibilità

agli albori usato linguaggio macchina

≡ insieme dei comandi direttamente eseguibili dalla macchina

seconda metà degli anni cinquanta, il linguaggio si alza di livello
usando traduttori

≡ programmi che traducono linguaggi di alto livello nel linguaggio macchina.

- Fortran (*FORmula TRANslator*) elaborazioni matematiche
- Cobol (*COmmon Business Oriented Language*) applicazioni gestionali

linguaggi basati sui principi della programmazione (capostipite ALGOL 60)

- Pascal, la didattica dell'informatica
- C per costruzione di sistemi informatici
- ADA dal Dipartimento della Difesa (DoD), applicazioni industriali e militari

stile di programmazione recente *orientato agli oggetti (object-oriented)*

≡ corrispondenza fra oggetti dell'applicazione e codifica degli algoritmi

avvento di Internet

⇒ linguaggi per la programmazione di reti di calcolatori, Java

linguaggi di tipo matematico

- LISP, basato sul concetto matematico di *funzione*
- Prolog, basato sul formalismo della *logica matematica*

Architettura dei sistemi informatici

sistema informatico va dal microcalcolatore al grande calcolatore con migliaia di utenti

prima suddivisione, convenzionale ma utile

hardware, i componenti fisici del sistema

software, i programmi eseguiti dal sistema

Hardware

elementi funzionali presenti in ogni calcolatore

- **unità di elaborazione, o processore (CPU, *Central Processing Unit*)**
elabora dati, coordina trasferimento dei dati
esegue i programmi, cioè interpreta ed esegue le loro istruzioni;
- **memoria centrale (RAM *Random Access Memory*)**
memorizzare dati e programmi
capacità limitata
volatile
accesso all'informazione molto rapido
- **memoria secondaria o memoria di massa**
memorizza grandi quantità di dati e programmi
persistente
accesso molto meno rapido della RAM
- **bus di sistema, collega e consente scambio di dati**
- **unità periferiche per comunicare con l'ambiente esterno**
terminali, con tastiera, mouse, video
stampanti
....
- **se ambiente esterno non è un utente umano**
(impianti industriali, robot, strumenti di controllo)
sensori
attuatori

Personal computer

corpo (*box*) contenente unità centrale, memoria centrale di massa;
il box collegato a tastiera, mouse e video

memoria di massa:

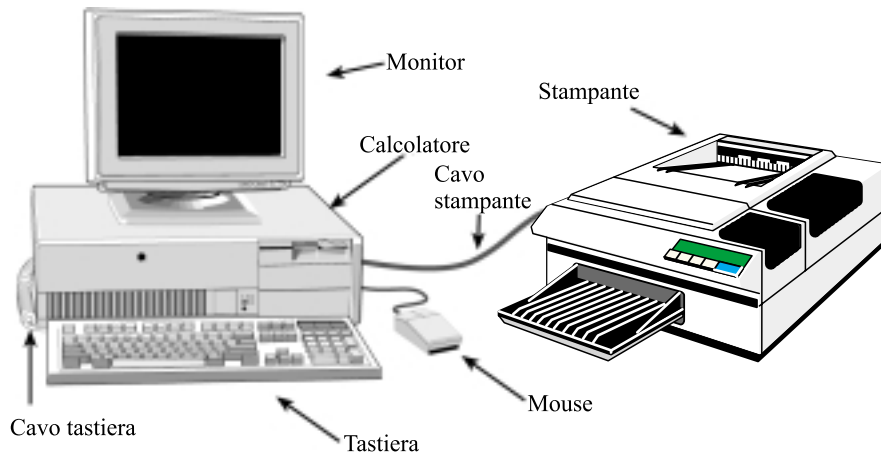
disco fisso (o hard disk),

dischetti (o floppy disk)

Compact Disk-Read Only Memory (CD-ROM)

Digital Video (Versatile) Disk (DVD)

nella memoria di massa informazione organizzata in file



altri sistemi informatici

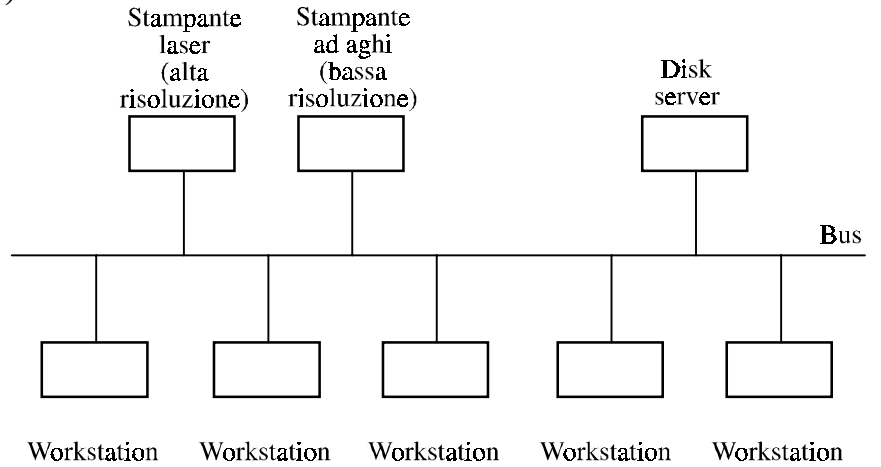
per livelli crescenti di complessità, prestazioni e prezzo

- **personal computer**
- **workstation (stazioni di lavoro).**
- **mainframe**

capacità di elaborazione cresce accrescendo memoria, processori, terminali e periferiche a una singola macchina

oppure componendo calcolatori in reti di calcolatori

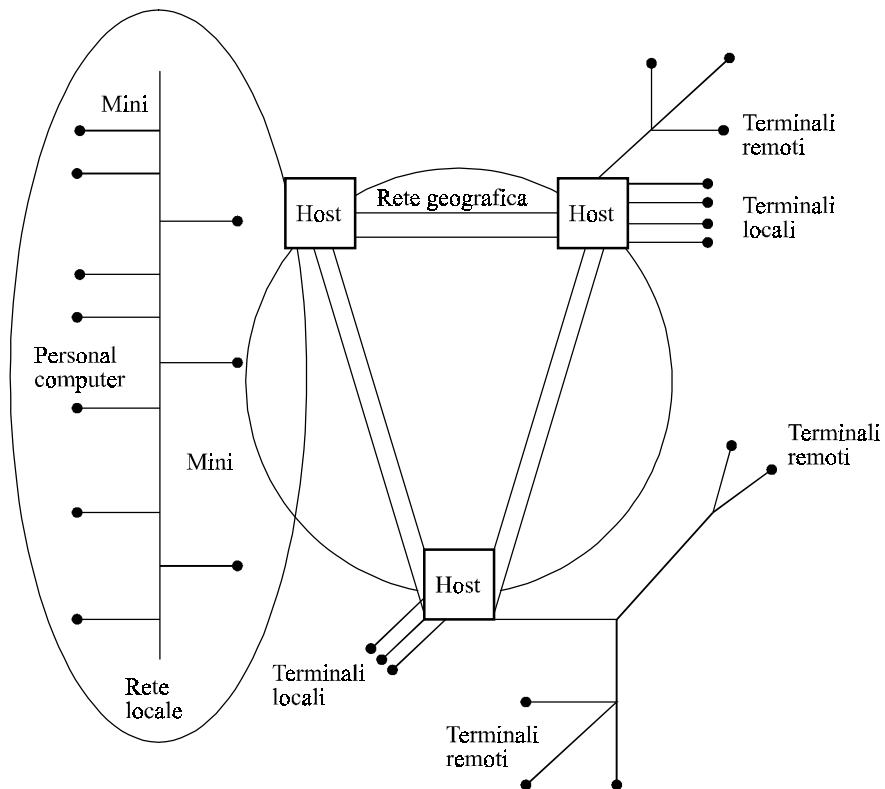
reti locali (LAN, Local Area Network)



reti geografiche (WAN, Wide Area Network).

calcolatori di grande dimensione (*host computer*)

terminali collegati direttamente o tramite linee telefoniche (*modem*)



Software

- software *di base* per gestione dell'elaboratore
- software *applicativo* per specifiche esigenze applicative

sistema operativo

nei sistemi semplici

interpretare ed eseguire dei comandi elementari
organizzare la struttura della memoria di massa

nei sistemi complessi (multi utente)

coordinare uso del calcolatore agli utenti
unità di elaborazione
memoria centrale
mostrare a ciascun utente un sistema di elaborazione “dedicato”

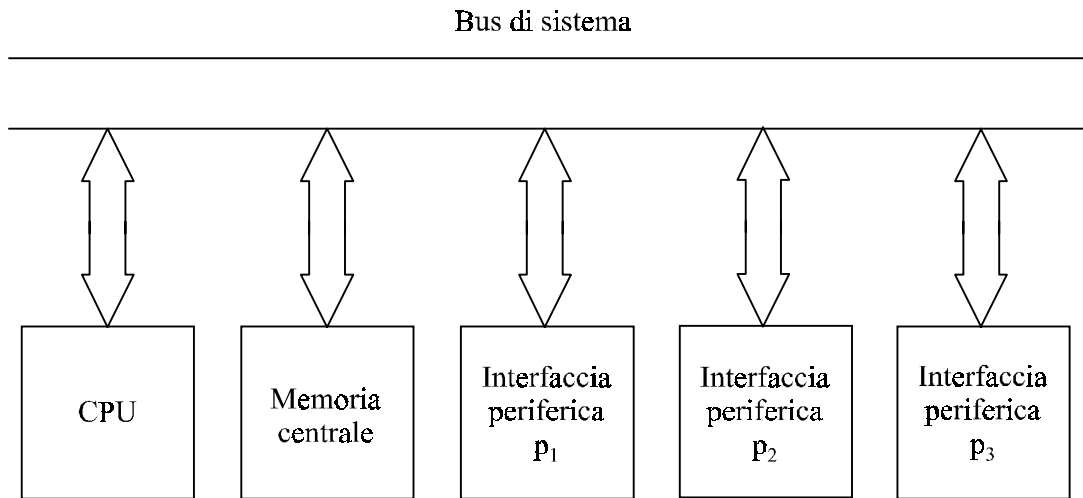
Ambiente di programmazione

- editor costruisce il file che contiene il programma
- compilatore traduce il programma sorgente in un *programma oggetto* eseguibile dall'elaboratore
- linker collega vari programmi oggetto in un unico *programma eseguibile*

Architettura di un calcolatore

non c'è nulla di “magico” nell'esecuzione di un programma

modello della *macchina di von Neumann*



le periferiche includono
memoria di massa
terminali (video e tastiera)
stampanti, e altri dispositivi esterni

unità di elaborazione coordina le varie attività
estrae istruzioni dalla memoria
decodifica
esegue

istruzioni: operazioni di *elaborazione*
di *trasferimento* (tramite bus di sistema)

istruzioni divise in
codice operativo operazione da compiere
operandi dati elaborati o trasferiti

elaborazioni evolvono in modo sincrono con *orologio (clock) di sistema*

Codifica dei dati e delle istruzioni di programma

codifica in forma binaria, con sequenze finite di 1 e di 0

minima unità di informazione bit (*binary digit*)

byte, otto bit, può avere 2^8 differenti configurazioni

(00000000, 00000001, 00000010, ..., 11111111)

calcolatore tratta diversi tipi di dati tutti codificati in forma binaria

Esempi

- Otto bit *numeri naturali* da 0 a 255 ($= 2^8 - 1$)

0 \Leftrightarrow 00000000, 8 \Leftrightarrow 00001000; 255 \Leftrightarrow 11111111 etc.

- *numeri interi*: primo bit del byte per il segno

un byte codifica interi fra -127 ($-(2^{(8-1)} - 1)$) e 127 ($2^{(8-1)} - 1$).

- *numeri reali* approssimati come numeri razionali;

notazione in virgola fissa: codifica separata di parte intera e frazionaria

8.345 \Leftrightarrow primo byte (8) 00001000
secondo byte (0.345) 01011000 ($\Rightarrow 2^{-2} + 2^{-4} + 2^{-5}$)

- *caratteri*: codifica ASCII (*American Standard Code for Information Interchange*)

impiega sette bit (rappresenta 128 caratteri)

“A” \Leftrightarrow 01000001

“;” \Leftrightarrow 00111011

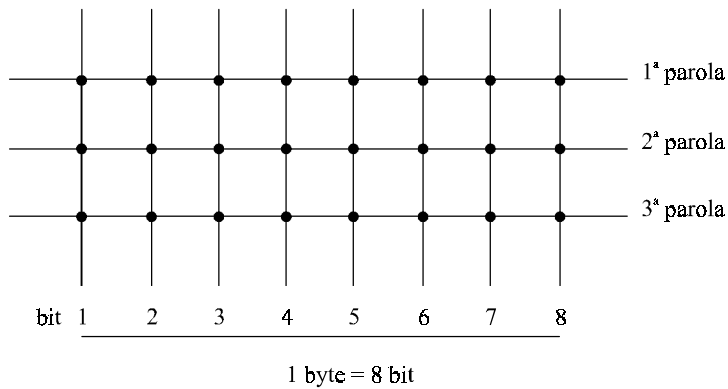
La memoria centrale

memoria centrale contiene dati e programmi

sequenza di celle di memoria, o *parole (word)*

parole tutte della stessa capienza, dimensione varia con il tipo di macchina

parole contengono un numero intero di byte (8, 16, 32, 64 ... bit)



ogni cella di memoria *indirizzata*

selezionata col suo indirizzo \equiv posizione relativa

indirizzamento della memoria tramite registro indirizzi

registro indirizzi ha k bit \Rightarrow indirizzabili 2^k celle (indirizzi tra 0 e $2^k - 1$)

\Rightarrow dimensione della memoria in genere potenza di 2

registro indirizzi di 10 bit $\Rightarrow 2^{10} = 1024$ celle

1024 approssimato a "kilo" e si parla di kiloparole

registro indirizzi di 20 bit $\Rightarrow 2^{20} = 1048576$ celle

1048576 approssimato a "mega" e si parla di megaparole

così via con 30 (giga) 40 (tera), 50 (peta) etc.

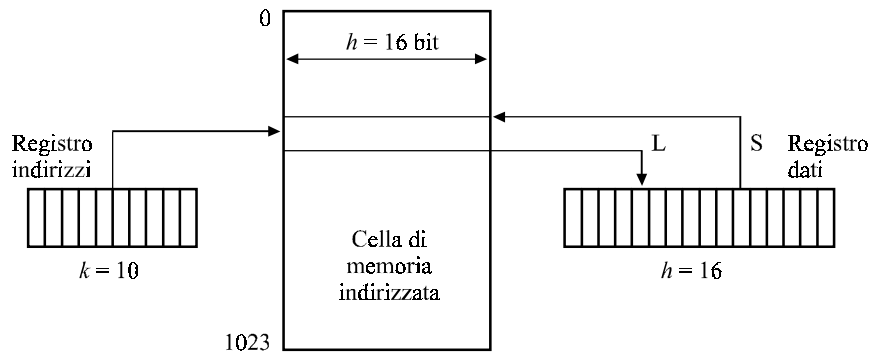
NB: capacità memoria usualmente misurata in byte, non parole
perchè dimensione parola dipende da tipo di macchina

su memoria due operazioni: lettura e scrittura

attraverso *registro dati*, lungo come parola di memoria

lettura carica (*load*) registro dati con una parola di memoria
scrittura copia (*store*) registro dati in una cella di memoria

lettura e scrittura controllate da CPU e tramite bus



bus di sistema diviso in

- bus dati trasmette valori da elaborare o elaborati tra CPU e altri
- bus indirizzi trasmette indirizzi tra CPU e registro indirizzi
- bus controlli trasferisce informazione di controllo tra CPU e altri

Esempio: operazione di *lettura* dalla memoria centrale

CPU:

- carica indirizzo in registro indirizzi e trasmette a memoria centrale tramite bus indirizzi
- comanda alla memoria operazione di lettura tramite bus controlli.

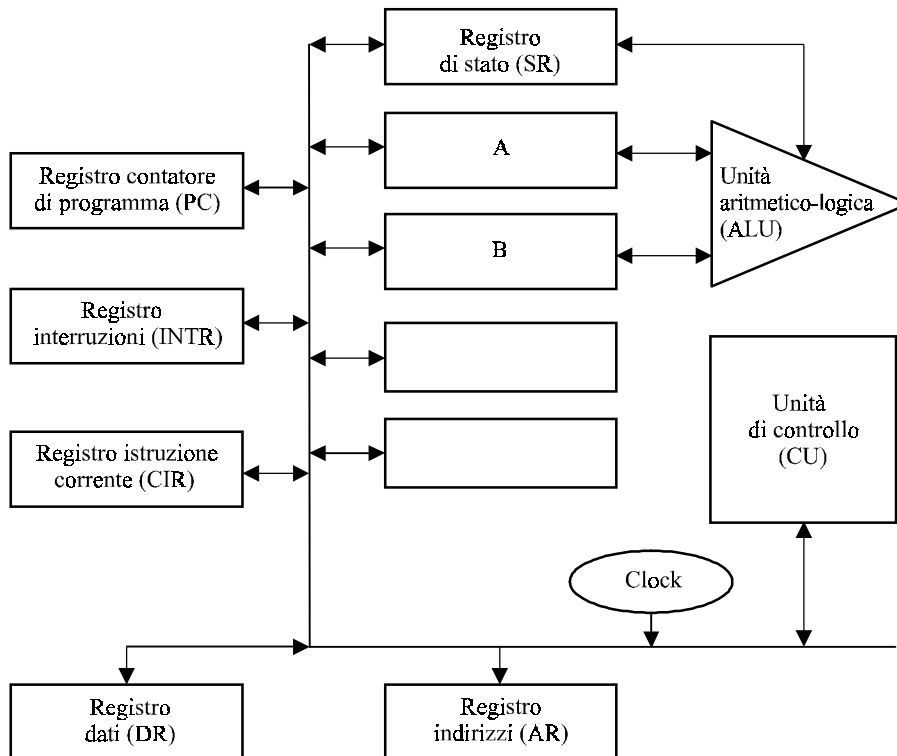
memoria:

- esegue lettura \equiv sposta tramite bus dati parola indirizzata nel registro dati;
- segnala a CPU, tramite il bus controlli, fine operazione

operazione di scrittura svolta in modo simile

L'unità di elaborazione

- **unità di controllo:** prelievo e decodifica istruzioni
invio dei segnali di controllo
- **unità aritmetico-logica**
- **CIR contiene istruzione in esecuzione**



- **PC, Program Counter** indirizzo della prossima istruzione da eseguire;
- **INTR** informazioni sul funzionamento delle periferiche
- **registri A e B** operandi risultato delle elaborazioni dell'ALU;
- **registri di lavoro** analoghi a celle di memoria, ma letti e scritti velocemente durante una singola istruzione.
- **SR** ogni bit indica condizione sul risultato delle operazioni della ALU

Esempi di bit di SR

1. **bit di carry:** presenza di un riporto (*carry-over*);
2. **bit zero:** presenza di valore nullo nel registro A;
3. **bit di segno:** segno del risultato di un'operazione aritmetica;
4. **il bit di overflow:** risultato operazione supera massimo valore memorizzabile (e.g. 2^h se registro A lungo h bit)

Interfacce di ingresso/uscita

consentono collegamento tra elaboratore e periferiche

interfaccia contiene registri per
inviare comandi alla periferica
scambiare dati
controllare il funzionamento della periferica

Esempio di esecuzione dei programmi

calcolare espressione $(a + b) * (c + d)$,
leggendo variabili da dispositivo di ingresso
scrivendo risultato sul dispositivo di uscita.

algoritmo generale:

- 1. leggi valori a, b, c e d da dispositivo di ingresso;**
- 2. somma valori di a e b ;**
- 3. salva risultato parziale in memoria;**
- 4. somma valori di c e d ;**
- 5. moltiplica risultato parziale con quello precedente;**
- 6. scrivi il risultato finale su dispositivo di uscita;**
- 7. arresta l'esecuzione del programma.**

celle numero 16, 17, 18, 19 e 20 per a, b, c e d e risultato

versione algoritmo in forma vicina a macchina di von Neumann

1. poni in memoria centrale, in cella 16, il valore letto; lo stesso per b , c e d nelle celle 17, 18 e 19;
2. esegui l'addizione di a e b :
 - 2.1. copia cella 16 in registro A;
 - 2.2. copia cella 17 in registro B;
 - 2.3. somma i due registri (l'operazione è eseguita dalla ALU);
3. immagazzina risultato (ora in registro A) nella cella 20;
4. esegui l'addizione di c e d :
 - 4.1. copia cella 18 in registro A;
 - 4.2. copia cella 19 in registro B;
 - 4.3. somma i contenuti dei due registri;
5. esegui la moltiplicazione di $(a + b)$ e $(c + d)$:
 - 5.1. copia in registro B cella 20;
 - 5.2. moltiplica il contenuto dei due registri [si osservi che il registro A contiene ancora il valore $(c + d)$];
6. scrivi il risultato sul dispositivo di uscita:
 - 6.1. memorizza registro A, nella cella 20;
 - 6.2. scrivi cella 20 nel registro dati della periferica appropriata.
7. Arresta l'esecuzione del programma.

NB: ci sono operazioni sia di elaborazione sia di trasferimento dati

caricata in memoria centrale la forma binaria del programma

0100000000010000	leggi un valore in ingresso e ponilo nella cella numero 16 (variabile <i>a</i>)
0100000000010001	leggi un valore e ponilo nella cella numero 17 (variabile <i>b</i>)
0100000000010010	leggi un valore e ponilo nella cella numero 18 (variabile <i>c</i>)
0100000000010011	leggi un valore e ponilo nella cella numero 19 (variabile <i>d</i>)
0000000000010000	carica il registro A con il contenuto della cella 16 (valore <i>a</i>)
0001000000010001	carica il registro B con il contenuto della cella 17 (valore <i>b</i>)
0110000000000000	somma i contenuti dei registri A e B
0010000000010100	immagazzina il contenuto del registro A nella cella numero 20 (risultato parziale)
0000000000010010	carica il registro A con il contenuto della cella 18 (valore <i>c</i>)
0010000000010011	carica il registro B con il contenuto della cella 19 (valore <i>d</i>)
0110000000000000	somma i contenuti dei registri A e B
0001000000010100	immagazzina il contenuto del registro A nella cella numero 20 (risultato)
0101000000010100	scrivi in output il contenuto della cella numero 20 (risultato)
1101000000000000	arresta l'esecuzione del programma
.....	spazio per la <i>a</i> (cella 16)
.....	spazio per la <i>b</i> (cella 17)
.....	spazio per la <i>c</i> (cella 18)
.....	spazio per la <i>d</i> (cella 19)
.....	spazio per il risultato (cella 20)

(a)

0100000000010000	cella numero 0
0100000000010001	cella numero 1
0100000000010010	cella numero 2
0100000000010011	cella numero 3
0000000000010000	cella numero 4
0001000000010001	cella numero 5
0110000000000000	cella numero 6
0010000000010100	cella numero 7
0000000000010010	cella numero 8
0010000000010011	cella numero 9
0110000000000000	cella numero 10
0001000000010100	cella numero 11
1000000000000000	cella numero 12
0010000000010100	cella numero 13
0101000000010100	cella numero 14
1101000000000000	cella numero 15
	cella numero 16 riservata alla variabile <i>a</i>
	cella numero 17 riservata alla variabile <i>b</i>
	cella numero 18 riservata alla variabile <i>c</i>
	cella numero 19 riservata alla variabile <i>d</i>
	cella numero 20 riservata al risultato
	} celle di memoria libere

(b)

Ogni istruzione eseguita in tre fasi

acquisizione dalla memoria centrale

interpretazione (decodifica)

esecuzione.

acquisizione (fetch), divisa in quattro passi

PC trasferito in AR

lettura dalla memoria centrale:

contenuto della cella indirizzata va in DR tramite il bus di sistema

DR trasferito in CIR.

PC incrementato di 1 (per prossima istruzione)

Prima della fase di acquisizione della prima istruzione PC 00 0000 0000

Fase di acquisizione della prima istruzione

