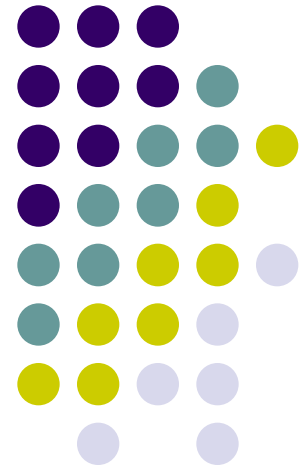
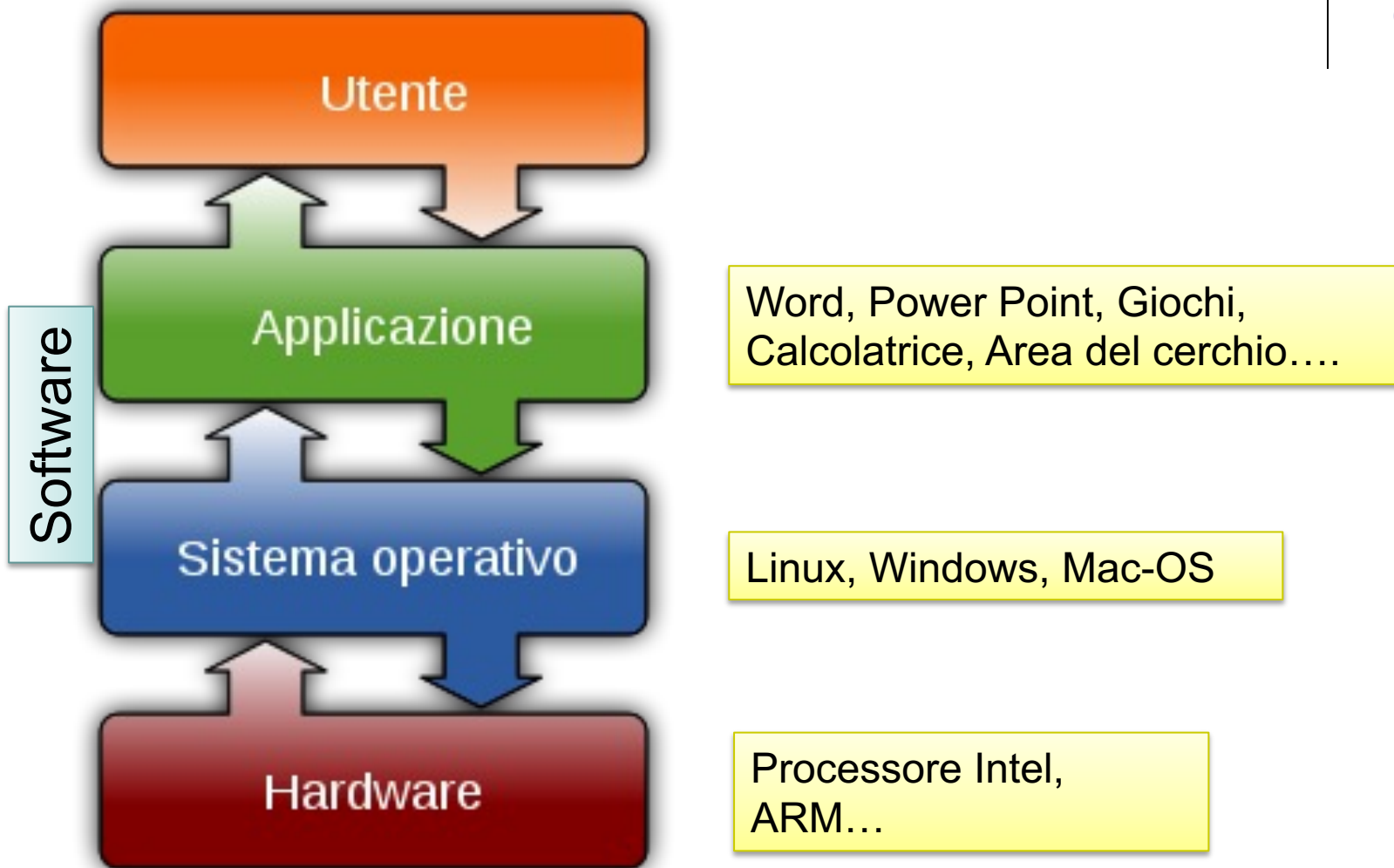


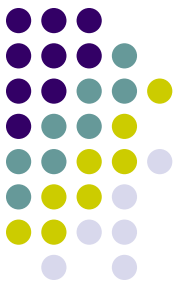
Fondamenti di Informatica

Elementi di Architettura *(parte 1)*



Livelli di un elaboratore

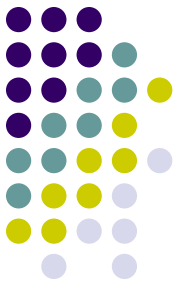




Architettura hardware

- Con il termine *architettura* si intende l'organizzazione dell'elaboratore in parti (o componenti, o unità) e l'interconnessione tra di esse
- Nel seguito ci riferiremo in particolare all'architettura di un personal computer, ma esistono diverse categorie di calcolatori che differiscono per:
 - **dimensioni**
 - **potenza di calcolo**
 - **numero di utilizzatori**
 - **tipologia di utilizzo**
 - **costo**

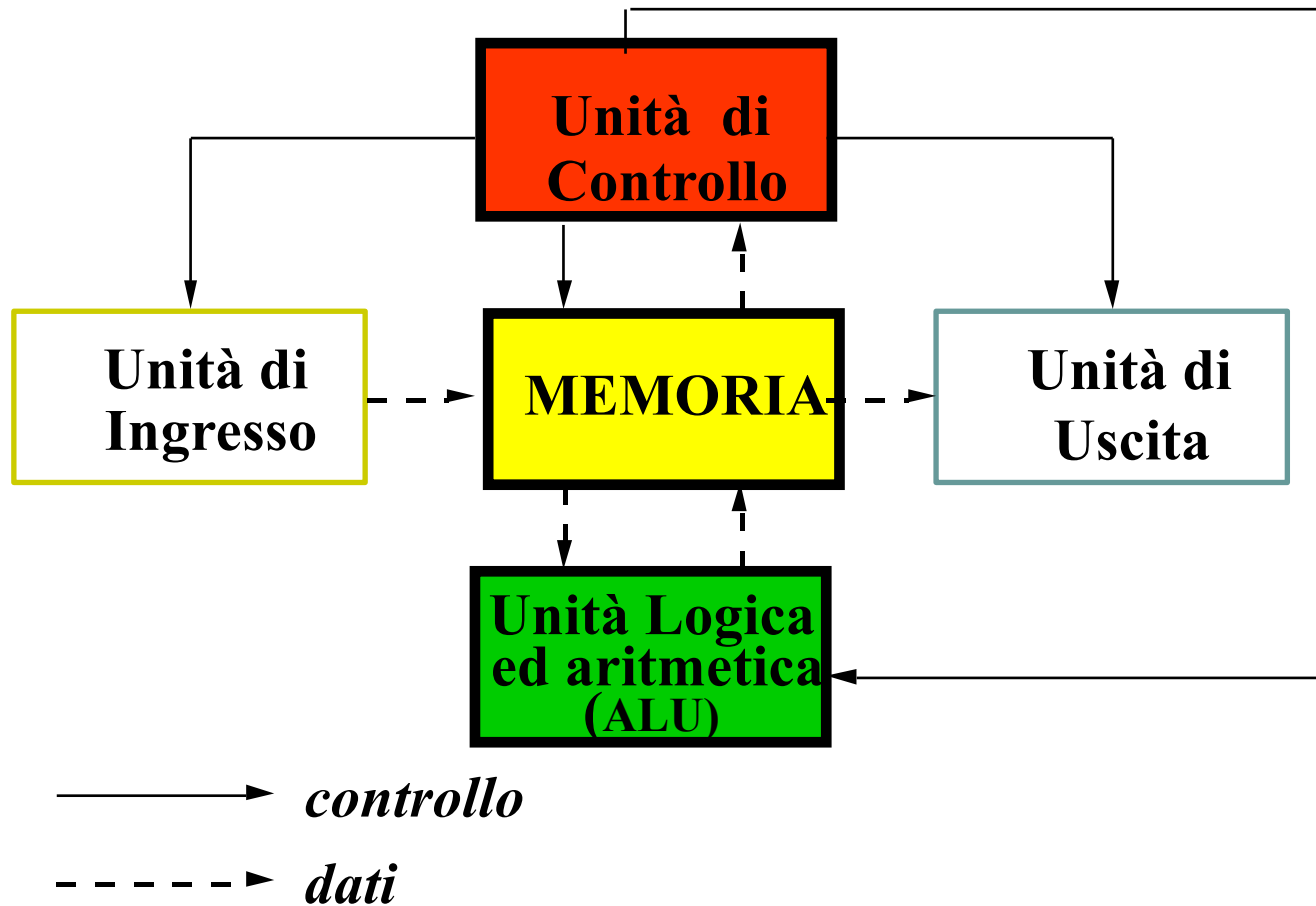
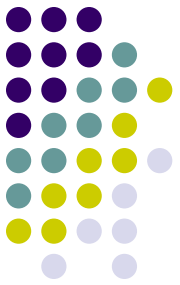
Modello Architeturale



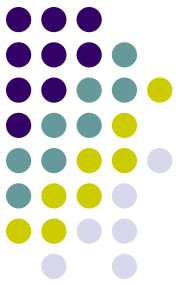
- Esistono diverse categorie di calcolatori che differiscono per dimensioni, potenza di calcolo, numero di utilizzatori, tipologia di utilizzo, costo...
- Il **modello** base per la maggior parte dei calcolatori (con alcune significative eccezioni) è quello di **Von Neumann**

Un **modello** è una rappresentazione astratta di una entità (oggetto, fenomeno, comportamento, sistema...) oggetto di studio e di analisi. Esistono diversi tipi di modelli, che si differenziano anche in base agli strumenti utilizzati per descriverli (ad esempio modelli matematici, modelli simulativi, modelli operazionali, ...)

Il modello di Von Neumann



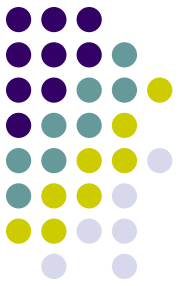
- È uno **schema di principio** rappresentativo dei tradizionali computer che prende il nome da John Von Neumann che lo propose nel 1945.



Unità nel modello di VN

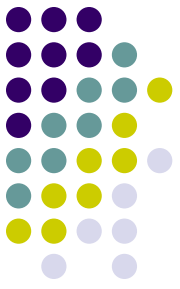
- Unità di Ingresso per l'acquisizione dei dati e dei **programmi** e per il loro trasferimento in memoria
- Unità di Memoria per la registrazione sia dei dati che delle istruzioni del programma
- Unità di Controllo presiede a tutte le operazioni, interpreta le istruzioni prelevate dalla memoria e ne 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

Programma memorizzato



- Il modello di Von Neumann si basa sul concetto di **programma memorizzato**:
 - la macchina immagazzina nella propria memoria **le istruzioni** per il suo funzionamento e **i dati su cui lavorare**
 - Ciò conferisce al calcolatore ampia flessibilità operativa.
- In linea generale il funzionamento interno di qualsiasi odierno computer si può ricondurre al modello di Von Neumann.

Ruolo delle Unità



- L'Unità di controllo coordina l'esecuzione delle operazioni fondamentali;
- La ALU effettua le operazioni logico-aritmetiche
- La *memoria* contiene il programma che descrive le operazioni da eseguire e i dati su cui il programma stesso opera;
- I dispositivi di *input* e *output* sono le interfacce della Unità di controllo nei confronti del mondo esterno

L'elaborazione secondo il modello di Von Neumann



- In LINEA di PRINCIPIO:
 - 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*)





Esempio

- Dati due numeri interi **a** e **b**, calcolare il loro prodotto
- (**res** = **a** * **b**). I valori di **a** e **b** devono essere letti da tastiera e il risultato viene stampato a video.
- Ipotizziamo che l'unica operazione che l'ALU esegue sia l'**addizione**.

leggi da tastiera il valore di **a**

leggi da tastiera il valore di **b**

Calcola il risultato **res**:

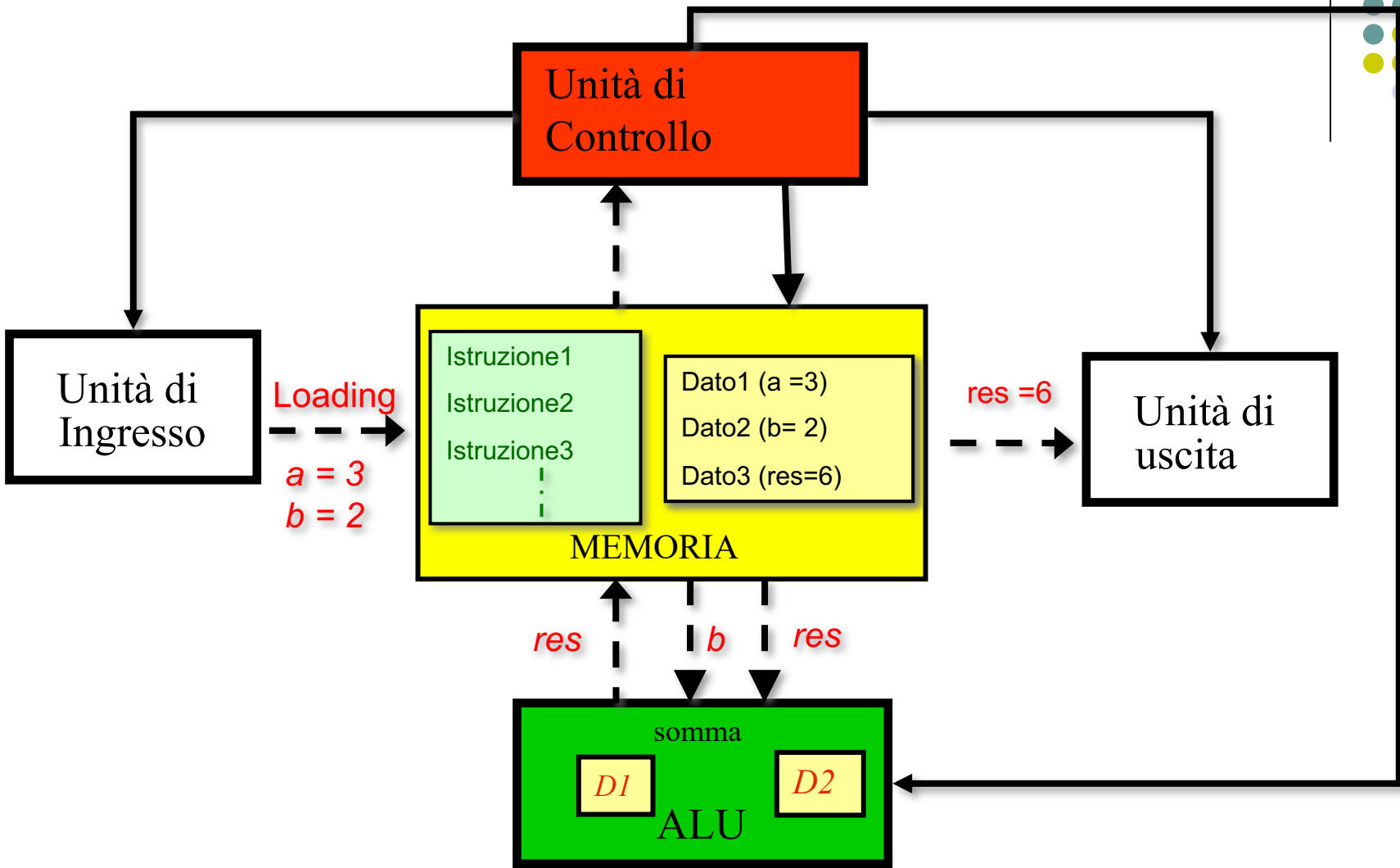
$$res = \underbrace{b + b + b \dots + b}_{a \text{ volte}}$$

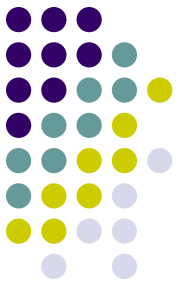


```

stampa «Inserire il valore di a:»
leggi(a)
stampa «Inserire il valore di b:»
leggi(b)
//il risultato inizialmente è zero
res = 0;
fino a quando (a>0) fai le
seguenti operazioni:
{
    res = res + b;
    a = a - 1;
}
stampa (res)
  
```

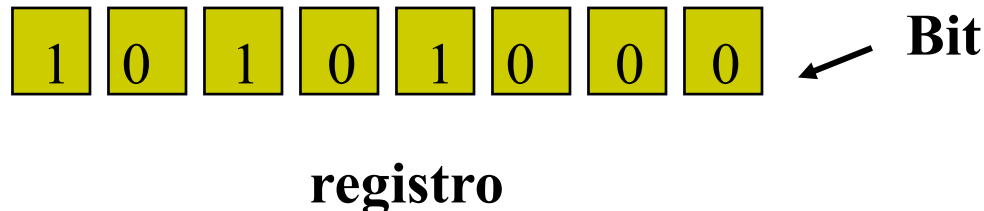
stampa a video il risultato **res**





Registri

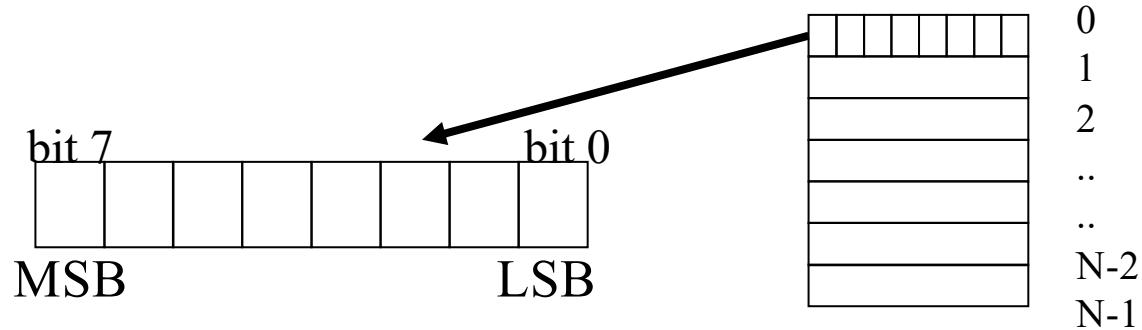
- Il calcolatore opera su informazioni rappresentate in registri
- Il registro è individuato da un “nome”
- E' un organo fisico bi-stabile atto memorizzare 2 stati distinti
- Il valore di ogni cella di un registro (Bit) è mantenuto da un circuito logico detto flip-flop



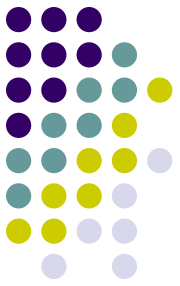
La memoria centrale



- La memoria centrale di un computer è organizzata come un **insieme numerato** di registri di lunghezza m , dette **locazioni**
- Gli m bit di una locazione sono accessibili dal processore (in lettura/scrittura) mediante un'unica operazione (questo parametro indica il **parallelismo** della memoria)
- Ogni locazione è individuata da un **indirizzo**, cioè un intero compreso tra 0 e $N-1$. L'insieme di locazioni indirizzabili è detto **SPAZIO DI INDIRIZZAMENTO**
- La memoria centrale è **ad accesso casuale** (RAM) perciò il tempo di accesso non dipende dalla posizione del dato (ovvero dall'indirizzo).
- La memoria centrale è **VOLATILE**



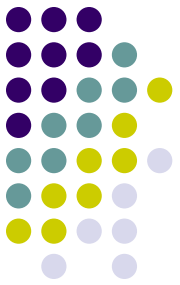
Unità di controllo



I circuiti del calcolatore sono predisposti per l'esecuzione di alcune operazioni elementari, ma attendono di volta in volta appositi segnali di abilitazione per essere attivati

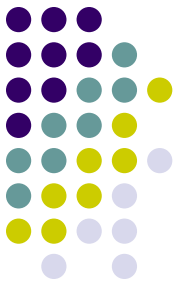
L'unità di controllo accede alla memoria, preleva le istruzioni da eseguire, le interpreta e le esegue con l'aiuto della ALU (algoritmo del processore)

Unità Aritmetico Logica (ALU)

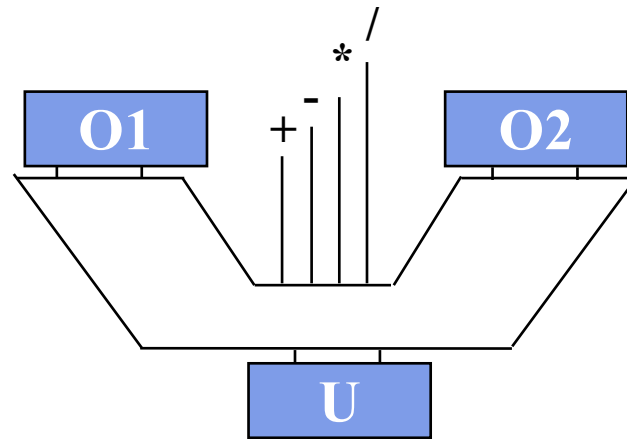


- E' il dispositivo capace di eseguire le operazioni aritmetiche fondamentali, le operazioni logiche e di shift
- L'Unità di controllo trasmette alla ALU gli operandi, cioè i dati su cui deve eseguire una operazione, e un comando che indica quale operazione effettuare
- Il suo funzionamento è realizzato mediante appositi circuiti (*circuiti di commutazione logica*)

Unità Aritmetico Logica (ALU)



- Gli operandi vengono forniti alla ALU sui suoi registri di ingresso
- La ALU esegue l'operazione e pone il risultato sul registro risultato, e inoltre invia informazioni sulle condizioni dell'operazione (flag del registro di stato)



IL PROCESSORE

(Central Processing Unit: CPU)

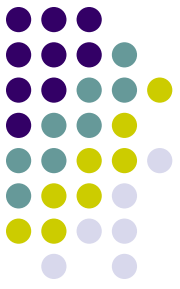


Rispetto al modello di von Neumann, i microprocessori integrano nell'unità di controllo la ALU e ulteriori registri, chiamati **registri del processore**.



Processore

Control Unit (CU)



- La CU è preposta all'interpretazione delle singole istruzioni e all'attivazione di tutti i meccanismi necessari ad eseguire le singole istruzioni:
 - preleva ogni istruzione dalla memoria centrale e la decodifica
 - preleva dalla memoria i dati che servono all'esecuzione dell'istruzione
 - esegue infine l'istruzione quando tutti i dati sono disponibili
- Al termine dell'esecuzione di una istruzione la CU procede al prelievo dalla memoria della successiva istruzione, in maniera sequenziale
 - L'esecuzione di una nuova istruzione inizia solo se la precedente è stata portata a termine.

ALU



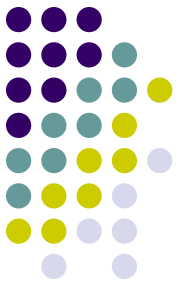
- A seconda dei processori l'ALU può essere molto complessa: nei sistemi attuali l'ALU viene affiancata da **processori dedicati alle operazioni sui numeri in virgola mobile** detti **(co)processori matematici**.
- Per i risultati intermedi di una elaborazione la CU può servirsi dei registri interni piuttosto che dei registri della memoria, perché questi ultimi hanno tempi di accesso inferiori.

Collegamento tra le Unità



- Le unità sono collegate tra loro tramite bus
- Un bus è un insieme di linee in grado di trasmettere un segnale elettrico.
- Attraverso un bus le unità si scambiano dati o istruzioni
- Il bus che collega la ALU alla Unità di controllo è detto **bus interno**
- Il bus che collega la memoria con le altre parti
- del sistema è detto **bus esterno (o bus di sistema)**

Registri del processore



- la lunghezza in bit dei registri della CPU viene anche indicato come numero di bit del processore
- Processori a 8 bit (anni '70/'80)
- Processori a 16 bit (anni '80)
- Processori a 32 bit (anni '90-inizio anni 2000)
- Processori a 64 bit (anni 2000 ad oggi)

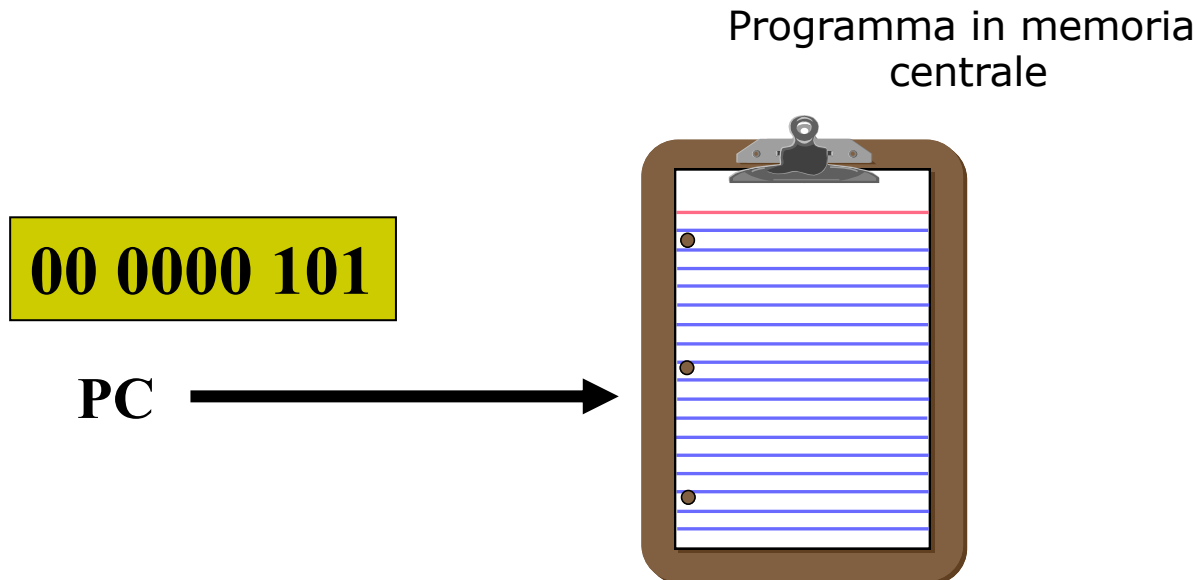
Registri sempre presenti:

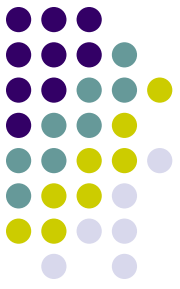
- Registri di comunicazione con la memoria
- Registri per l'interpretazione delle istruzioni
- Registro di Stato
- Accumulatore (deposito di dati da parte dell'ALU)



Registri per l'interpretazione delle istruzioni

- **PC o PI (Program Counter o registro Prossima Istruzione)** contiene l'indirizzo in memoria della prossima istruzione da eseguire
- **IR (Instruction Register)** contiene una copia dell'istruzione da eseguire
 - denominato anche D o RD: registro decodifica

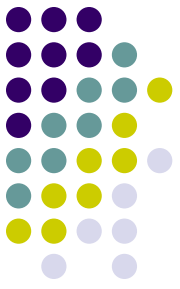




Registro di Stato

- **PSW o CC (Processor Status Word o Condition Code)** mantiene informazioni sullo stato del processore al termine dell'esecuzione dell'istruzione attraverso alcuni indicatori o *flag*, ad esempio:
 - ***N:*** è posto ad 1 se il risultato è negativo
 - ***Z:*** è posto ad 1 se il risultato è zero
 - ***V:*** è posto ad 1 se si è verificato un overflow
 - ***C:*** è posto ad 1 se si è avuto un riporto
 - ***I:*** è posto ad 1 se è presente una richiesta di interruzione

Algoritmo del Processore



Passi fondamentali:

1. **Prelievo dell'istruzione** (Fetch)

Il processore richiede alla memoria l'istruzione il cui indirizzo è in PC. L'istruzione viene copiata nel registro IR.

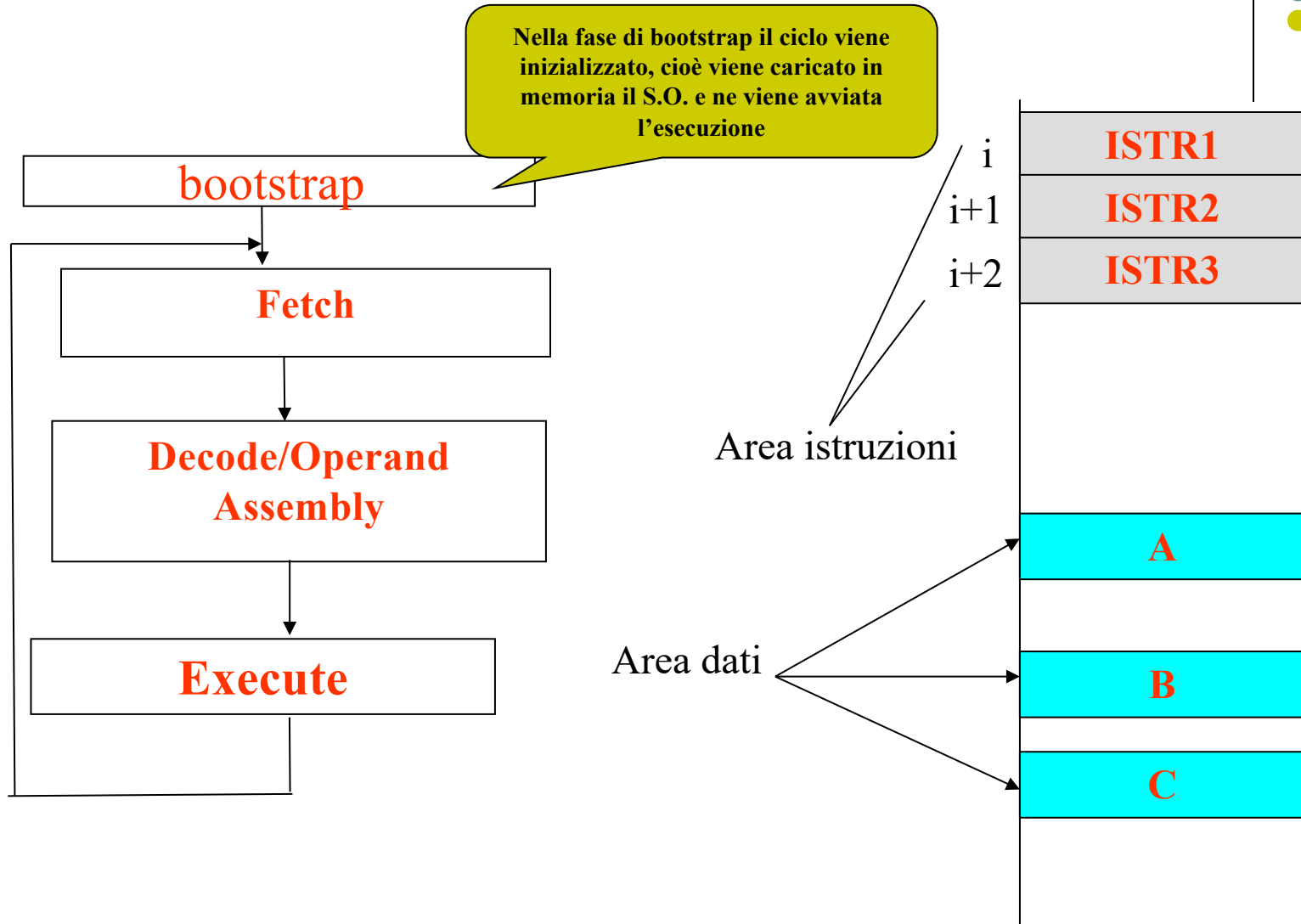
2. **Decodifica/prelievo degli operandi** (Operand Assembly)

L'unità di controllo esamina i bit di IR e ricava il tipo di operazione e i suoi operandi.

3. **Esecuzione dell'istruzione** (Execute)

L'unità di controllo richiede all'ALU di effettuare l'operazione e invia il risultato a un registro o alla memoria.

Algoritmo del processore

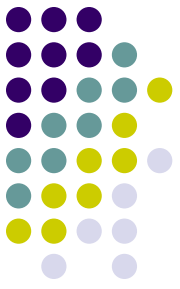


La fase di Boot



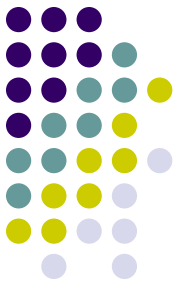
- La **Fase di Boot** predispone la CPU per l'inizio del ciclo, impostando **il registro PI alla prima locazione di memoria che contiene istruzioni** per l'Unità di Controllo.
- Una volta avviato, il ciclo del processore non termina mai e ad ogni istruzione ne segue un'altra sino all'arresto del calcolatore.
 - Perché tutto ciò proceda nel rispetto del modello di Von Neumann, in memoria devono essere presenti istruzioni e dati da elaborare.

Ciclo del Processore: Fase di Fetch



- **La fase fetch:**
 - Preleva dalla memoria l'istruzione posta all'indirizzo indicato nel registro PI, e la porta nel registro IR.
 - La **fase fetch si conclude** con l'aggiornamento del registro PI con l'indirizzo all'istruzione successiva.

Ciclo del Processore: Fase di Decode e Operand Assembly

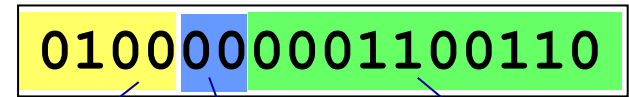


- La istruzione da eseguire deve essere analizzata e decodificata, gli operandi che servono all'esecuzione dell'istruzione devono essere predisposti.
 - L'UC interpreta l'istruzione contenuta nell'IR ovvero:
 - interpreta il **codice operativo dell'istruzione** (cosa fare, e se l'operazione coinvolge l'ALU).
 - Interpreta i **riferimenti ai dati di input ed output** (su quali dati eseguire l'operazione).
 - Il codice istruzione contenuto nell'IR contiene riferimenti ai dati usati nell'operazione.
 - Questa fase può richiedere ulteriori accessi in memoria centrale per preparare gli operandi in ingresso all'ALU.



Formato di una istruzione

- Codifica delle istruzioni:
 - codice operativo
 - operandi
- La maggior parte dei codici operativi prevede 0 o 1 o 2 o (raramente) 3 operandi



codice operativo

operando

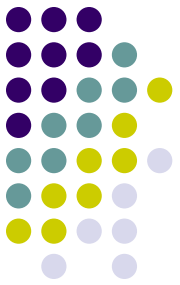
modo di indirizzamento

Ciclo del Processore: Fase di Execute



- La **fase execute** consiste nell'eseguire l'operazione di trasformazione indicata dal codice operativo.
- I risultati, se prodotti, verranno memorizzati negli indirizzi specificati dal codice istruzione indicato nell'IR.

Processori RISC e CISC



- **CISC (Complex Instruction Set Computer)**
 - Il linguaggio macchina ha istruzioni con potenza espressiva prossima a quella dei linguaggi di programmazione di alto livello;
 - Sono caratterizzati quindi da un ampio repertorio di istruzioni:
 - Sviluppo di programmi più semplice;
 - Codici ridondanti;
 - Maggiore complessità costruttiva;
- **RISC (Reduced Instruction Set Computer).**
 - Repertorio costituito da un ridotto ed essenziale insieme di istruzioni al fine di ottenere processori più veloci e di costo ridotto, data la minore complessità del loro progetto;
 - L'obiettivo fondamentale dell'approccio RISC è di disporre di un insieme fondamentale di istruzioni per ridurre al minimo il numero dei cicli di macchina (clock) necessari per loro esecuzione.
 - **Tutte le istruzioni RISC fondamentali hanno la stessa durata (un ciclo macchina), la stessa lunghezza e lo stesso formato**



Riferimenti

- Libro di Teoria «Alle radici dell'informatica» capitolo 2