



Architettura di un calcolatore



Indice

- La macchina di Von Neumann
 - Memoria
 - CPU
 - Bus
 - Interfacce
- Esempio
 - L'algoritmo
 - Il programma
 - Fasi di esecuzione di un'istruzione

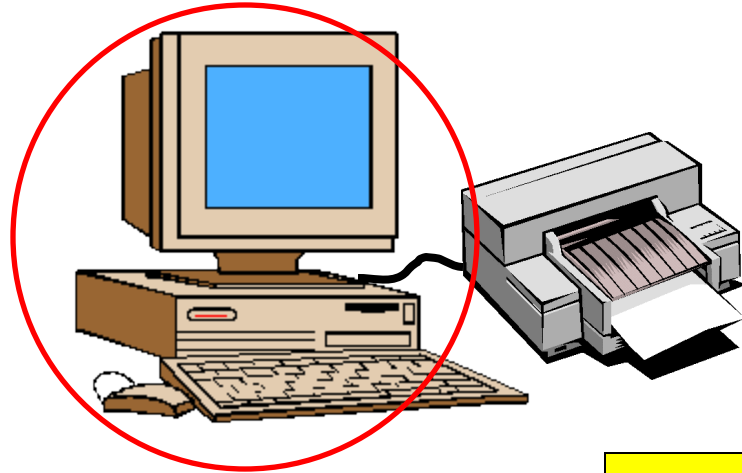


La macchina di Von Neumann(1)

- **Unità di Elaborazione** (CPU- Central Processing Unit)
contiene dispositivi elettronici per acquisire interpretare ed eseguire le istruzioni del programma
- **Memoria centrale** (RAM- Random Access Memory)
contiene informazioni necessarie all'esecuzione di un programma: istruzioni e dati
- **Bus di sistema**
opera il collegamento tra gli elementi funzionali
- **Interfacce**
forniscono il collegamento verso le periferiche che permettono lo scambio di informazioni tra elaboratore e mondo esterno (I/O)

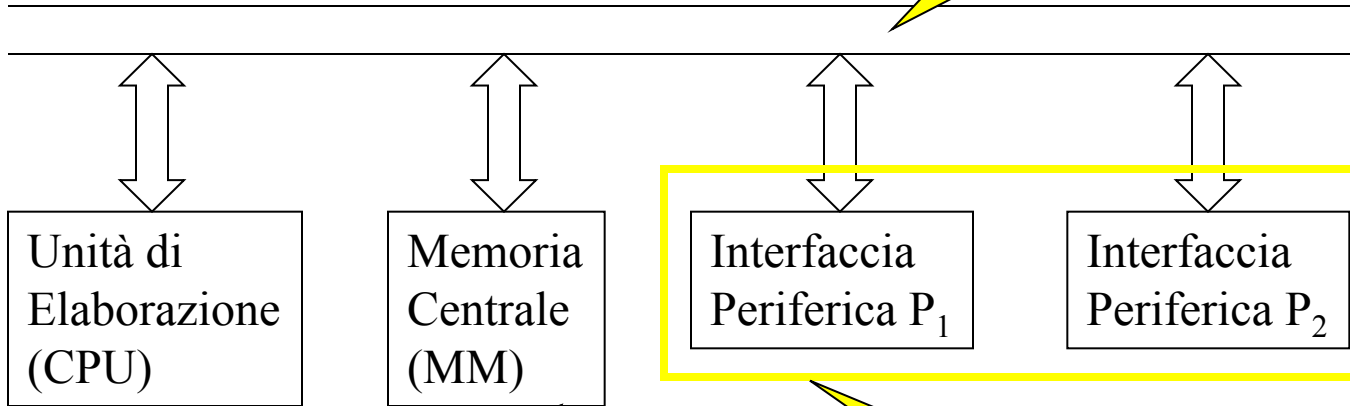


La macchina di Von Neumann(2)



Collegamento

Bus di sistema



Esecuzione istruzioni Memoria di lavoro

Memoria di massa, stampante, terminale...



Funzionamento

- La **CPU** *estrae*, *decodifica* ed *esegue* istruzioni(elaborazione o trasferimento di informazione) della **memoria**
- Il **bus di sistema** effettua il collegamento logico tra gli elementi funzionali in funzione del trasferimento in atto
- Le fasi di elaborazione si susseguono in modo *sincrono* rispetto alla scansione temporale imposta da un **clock di sistema**
- L'**unità di controllo** della CPU coordina durante ogni intervallo di tempo l'esecuzione temporale delle funzioni da svolgere nella CPU o negli altri elementi funzionali

LIMITE: esecuzione in sequenza



Codifica di dati e istruzioni

- Tutti i dati e le istruzioni di programma vengono codificate in forma **binaria**: sequenza finita di 1 e 0
 - **bit(binary digit)**: - più piccola unità di informazione memorizzabile o elaborabile
 - corrisponde allo stato di un dispositivo fisico
 - **byte**: 8 bit
- Codifica dati: Codifica binaria dell'informazione
- Codifica istruzioni(elaborazione o trasferimento):
 - **codice operativo**: specifica l'operazione da compiere
 - **operandi**: specifica dove la macchina può trovare i dati da elaborare o trasferire.



La memoria centrale – MM (1)

- Accoglie il materiale di lavoro di un elaboratore: dati e programmi
- E' un "passaggio obbligato" per l'informazione da elaborare (le operazioni comportano I/O)
- E' **volatile** e non *permanente* (come invece è la memoria di massa)
- Concettualmente: sequenza di celle di memoria, ciascuna contenente una **parola**: sequenza di bit grande a seconda del calcolatore (≥ 32 bit)
- Tecnologicamente:
 - dispositivi a semiconduttori;
 - "tabella" che ha per righe le varie celle e per colonne, in numero pari alla lunghezza della parola, ciascun bit di memoria
 - Informazione memorizzata come stato di tensione (1 alto – 0 basso) nelle posizioni di memoria agli incroci tra righe e colonne



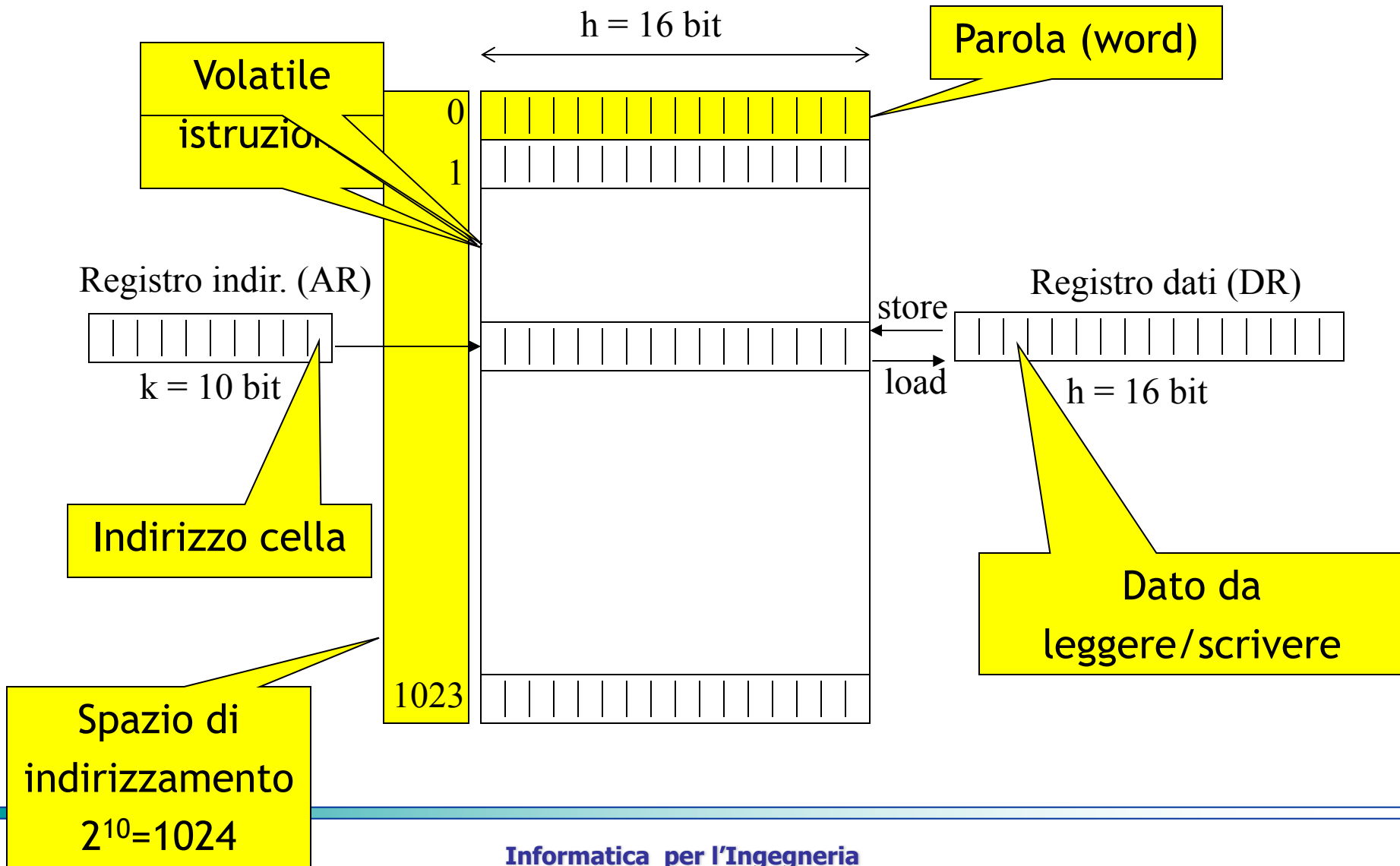
La memoria centrale – MM (2)

Operazioni eseguibili da un elaboratore (controllate dalla CPU e coinvolgenti il bus):

- **Indirizzamento** della MM(selezione di una particolare cella):
 - Indirizzo come posizione relativa rispetto alla prima cella(posizione 0)
 - Uso del **registro** (dispositivo elettronico capace di memorizzare una sequenza di bit) **indirizzi (AR)** della CPU
- **Lettura** dalla MM:
carica(*load*) il **registro dati(DR)** con la parola di memoria contenuta nella cella indirizzata
- **Scrittura** in MM:
deposita(*store*) il contenuto del **registro dati(DR)** nella parola di memoria indirizzata



La memoria centrale (MM)





Memorie RAM e ROM

- RAM (Random Access Memory):
 - memoria ad accesso casuale: indirizzabile in qualunque cella per operazioni di lettura e scrittura
- ROM (Read Only Memory):
 - zone di memoria su cui è impossibile scrivere, inizializzate dal costruttore con dati e programmi che servono al funzionamento del sistema (contenuto protetto)
 - contenuto persistente (usate nei microprocessori presenti in dispositivi di controllo: lavatrici, auto)



ROM

Classificazione in base alla scrittura dei costruttori:

- **EROM**(erasable ROM): cancellabili con raggi ultravioletti e riutilizzabili
- **PROM**(programmable ROM): scritte non in fase di costruzione, ma successivamente, con i programmatori di ROM
- **EPROM**(erasable programmable ROM)

Il software contenuto nelle ROM prende il nome di FIRMWARE per la sua natura a cavallo tra hardware(non modificabilità durante l'esecuzione) e software(programmabilità)



L'unità di elaborazione (CPU)

- Esegue i programmi nella memoria centrale *prelevando, decodificando ed eseguendo* una dopo l'altra le istruzioni
- Contiene gli elementi circuitali che regolano il funzionamento dell'elaboratore:
 - **unità di controllo:**
 - prelievo e decodifica di istruzioni
 - invio segnali di controllo che provocano i trasferimenti o le elaborazioni necessari per l'esecuzione dell'istruzione decodificata
 - **orologio di sistema:**
sincronizza le operazioni rispetto ad una data frequenza
 - **unità aritmetico-logica (ALU, Arithmetic Logic Unit):**
 - realizza le operazioni aritmetiche e logiche eventualmente richieste per l'esecuzione dell'istruzione



I registri della CPU

Registro: elemento di memoria leggibile e scrivibile molto velocemente, utilizzabile per memorizzare risultati parziali o informazioni di controllo

- **Registro dati(DR):** lungo quanto una parola (h bit)
- **Registro Indirizzi(AR) :** lungo k bit
- **Registro Istruzione Corrente (CIR, Current Instruction Register):** lungo h bit, contiene, istante per istante, l'istruzione in esecuzione
- **Contatore di programma(PC, Program Counter):** lungo k bit, contiene l'indirizzo della prossima istruzione del programma in esecuzione
- **Registro interruzioni(INTR, Interrupt Register):** contiene informazioni relative allo stato di funzionamento delle periferiche
- **Registri per operandi e risultati delle elaborazioni aritmetiche-logiche svolte dall'ALU (A e B in figura)**
- **Registri di lavoro:** in numero elevato, analoghi a celle di memoria, contengono dati e istruzioni di uso frequente o risultati intermedi, leggibili e scrivibili velocemente, senza operazioni di lettura e scrittura



ALU

- Capace anche di operazioni molto complesse(solo le 4 operazioni fondamentali negli esempi)
- Messa in azione dall'unità di controllo con un codice operativo
- Si serve dei registri operandi (A e B in figura), caricati con gli operandi prima dell'esecuzione dell'operazione
- Il registro A è caricato con il risultato dell'operazione dopo un certo tempo necessario all'esecuzione dell'operazione, misurato dall'orologio di sistema(nel caso di divisione intera il resto viene messo in B; per altre operazioni contenuto indefinito per B)



ALU- Registro di stato

- SR** (State Register): riporta indicazioni relative al risultato delle operazioni svolte dalla ALU in alcuni suoi bit:
1. **Bit di carry**, indica la presenza di un riporto(carry over)
 2. **Bit zero**, indica la presenza di un valore nullo in A
 3. **Bit di segno**, riporta il segno del risultato dell'operazione aritmetica
 4. **Bit di overflow**, rileva la condizione di overflow(se il risultato dell'ultima operazione supera il valore 2^h , massimo valore rappresentabile in A, se A è lungo h bit)



Il Clock di sistema

- Si occupa di sincronizzare le operazioni:
 - L'esecuzione di una singola istruzione di macchina può prevedere più clock
 - Il clock è tarato in modo che ci sia il tempo di effettuare un'operazione, ma non abbastanza da effettuare quella successiva:
 - L'informazione scorre senza sosta sotto forma di segnali elettrici in continuo movimento
 - Ogni dispositivo (registri, memoria, ecc) ha una **porta** normalmente chiusa, in modo tale da non consentire l'entrata di dati e la sovrascrittura di quelli contenuti
 - Un segnale di controllo proveniente dall'unità di controllo apre la porta del dispositivo su cui si deve effettuare un'operazione; un altro segnale la richiude dopo che l'operazione viene eseguita



L'unità di elaborazione (CPU)

Stato CPU
Flag: C, Z, S, V

Indirizzo prox istruzione

Registro contatore di programma (PC)

Registro interruzioni (INTR)

Registro istruzione corrente (CIR)

Registro di stato (SR)

A
B

Unità aritmetico-logica (ALU)

Registri operandi

Registri generali

Unità di controllo (CU)

Istruzione in elaborazione

Clock

Sincronizzazione

Registro dati (DR)

Registro indirizzi (AR)

Controllo:
-Prelievo
-Decodifica
-Esecuzione

Parola letta/da scrivere in MM

Indirizzo cella MM



Il bus di sistema(1)

- Insieme di connessioni elementari lungo cui viene trasferita informazione(trasmettendo valori di tensione)
- Collegamento aperto(non limitato ad un estremo) su cui si collegano le varie unità funzionali
- In ogni istante di tempo collega un'unità funzionale trasmittente ed una ricevente dati(CPU con memoria o CPU con interfaccia periferica)
- Sotto il controllo della CPU, che seleziona l'interconnessione da attivare e indica l'operazione da compiere: la CPU ha il ruolo di **MASTER**, le altre unità funzionali di **SLAVE**



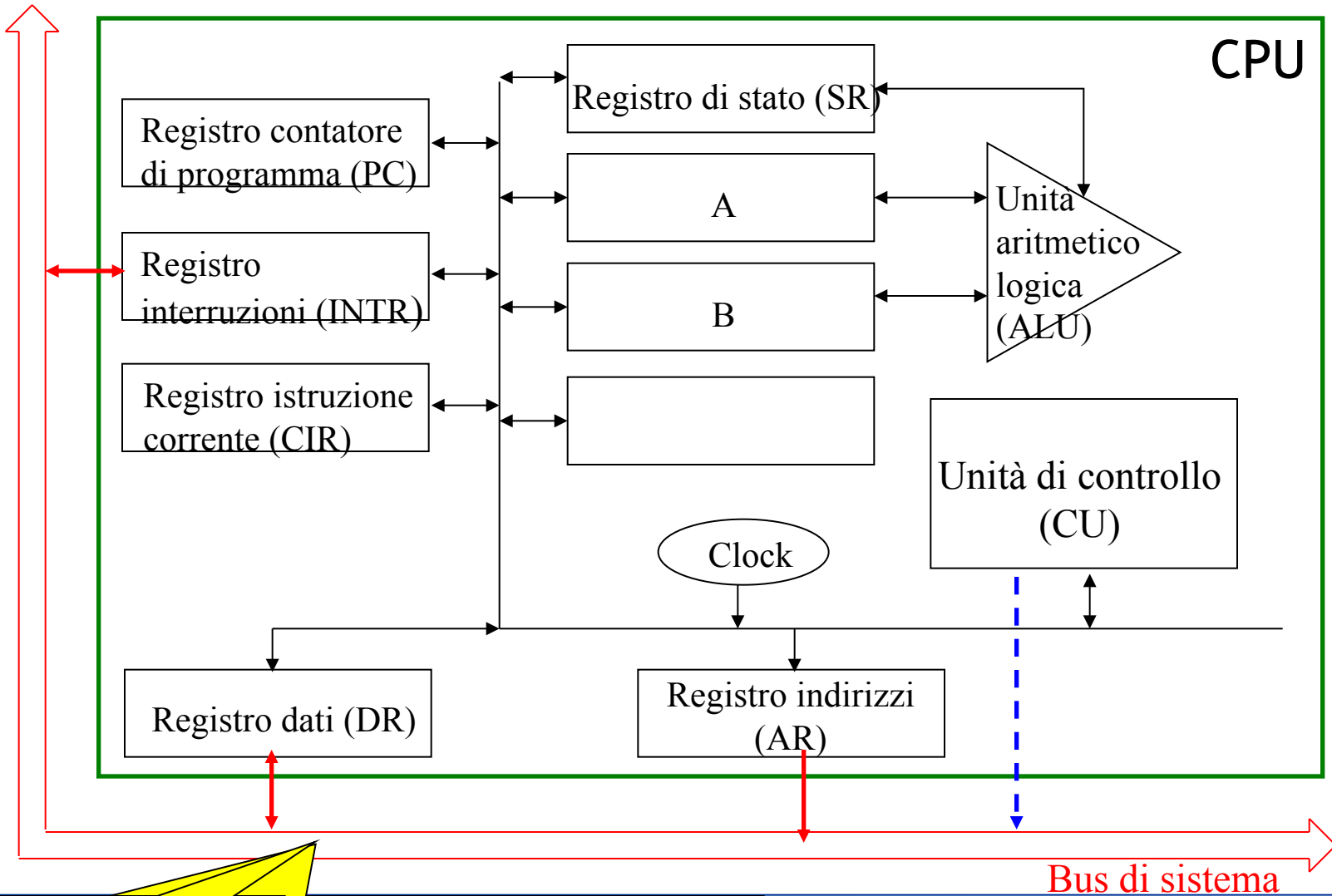
Il Bus di sistema(2)

Classificazione delle linee in base al tipo di informazione trasportata:

- **Bus dati:**
 - trasferisce dati dall'unità master alla slave e viceversa
 - trasferisce in parallelo grazie alla presenza di molte linee
 - in genere trasferisce ogni bit di una parola nello stesso istante, dedicando una linea a ciascuno di essi (se il numero di linee del bus dati non è minore dei bit della parola)
- **Bus indirizzi:** trasferisce indirizzi (per esempio l'indirizzo contenuto nel registro indirizzi alla memoria centrale per l'indirizzamento)
- **Bus controlli:** trasferisce dall'unità master alla slave un codice corrispondente all'istruzione da eseguire e dall'unità slave alla master informazioni sull'espletamento dell'operazione



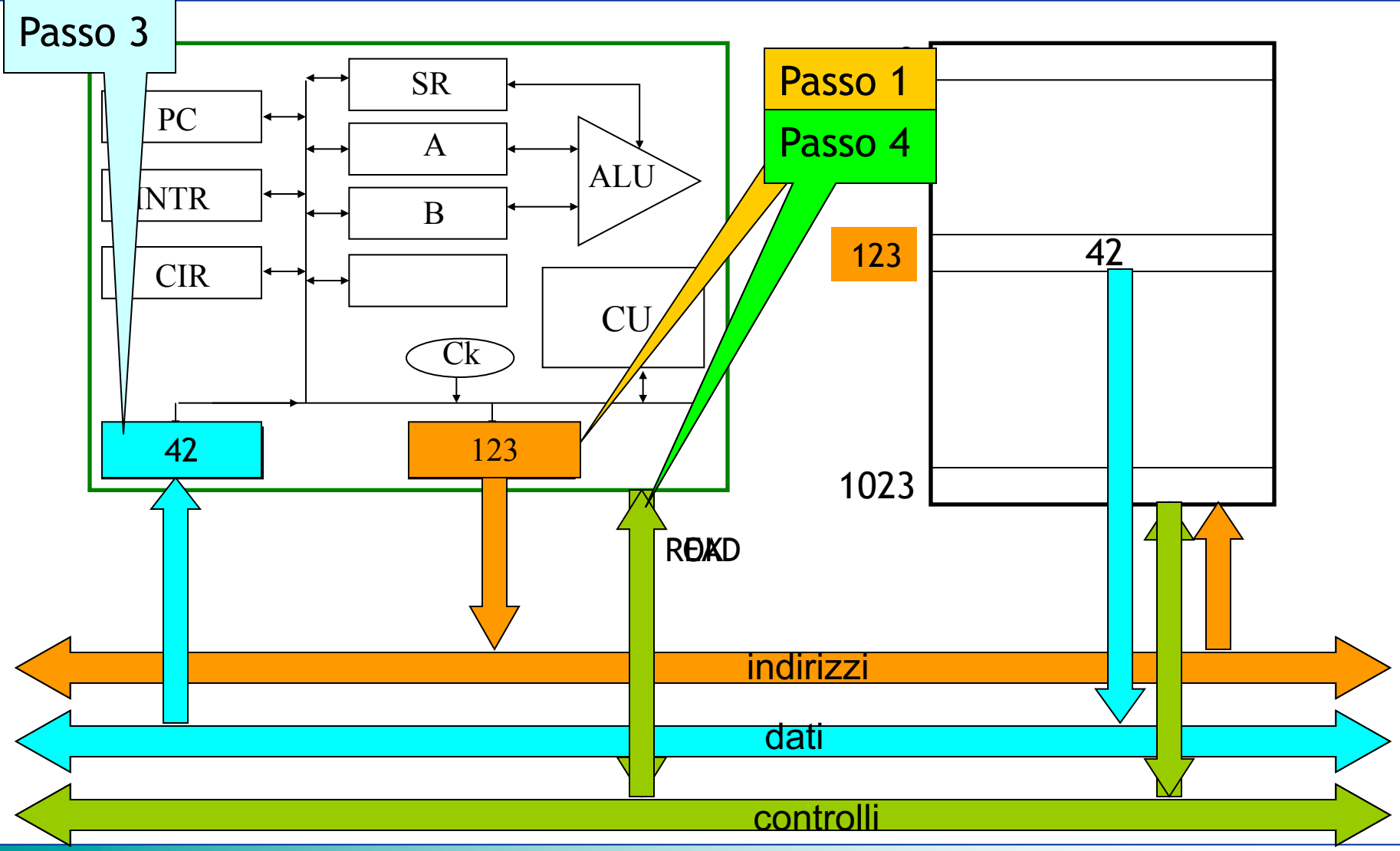
Il bus di sistema



Master/slave indirizzi, Bus controlli

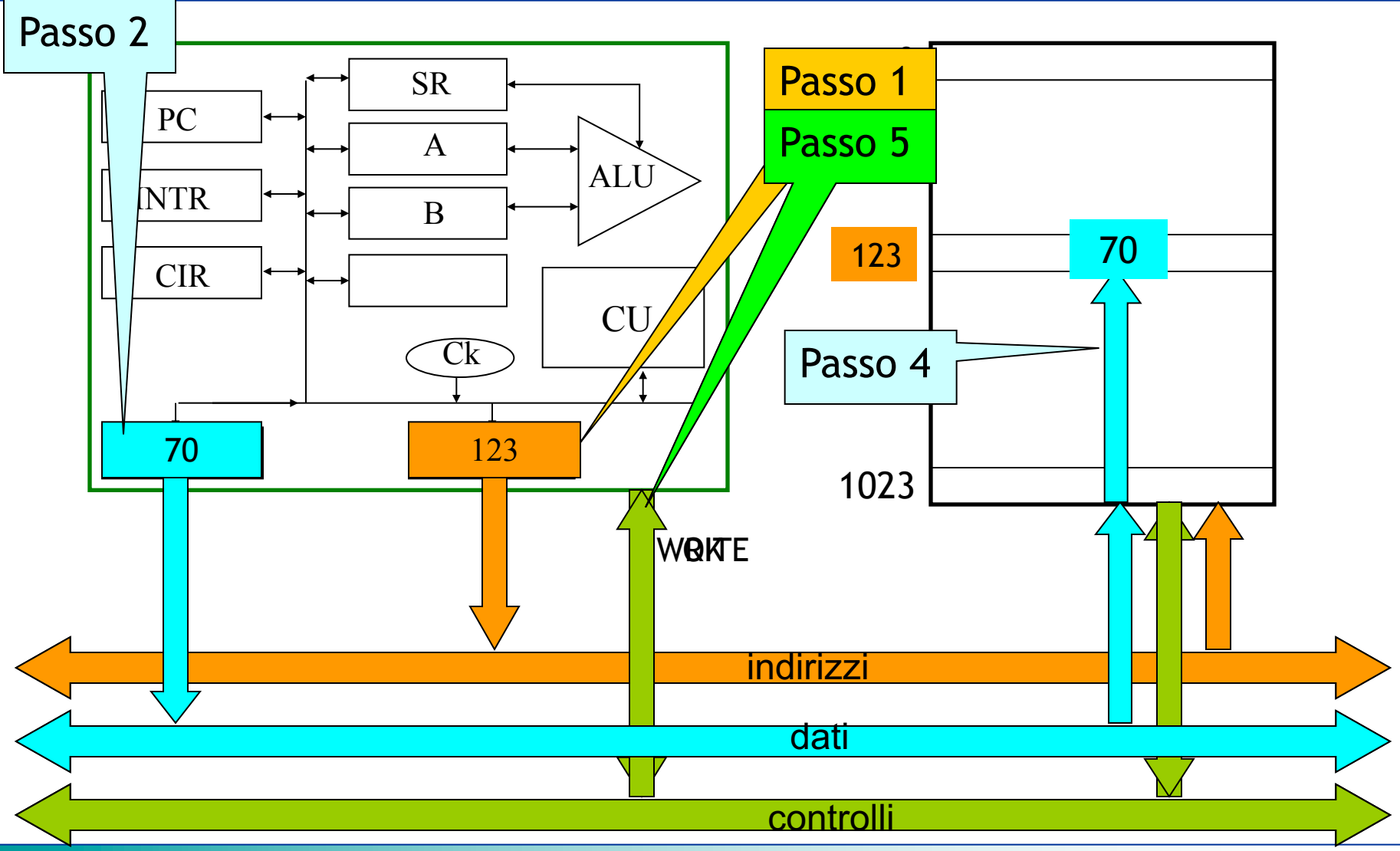


Sequenza di lettura





Sequenza di scrittura





Le interfacce delle periferiche(1)

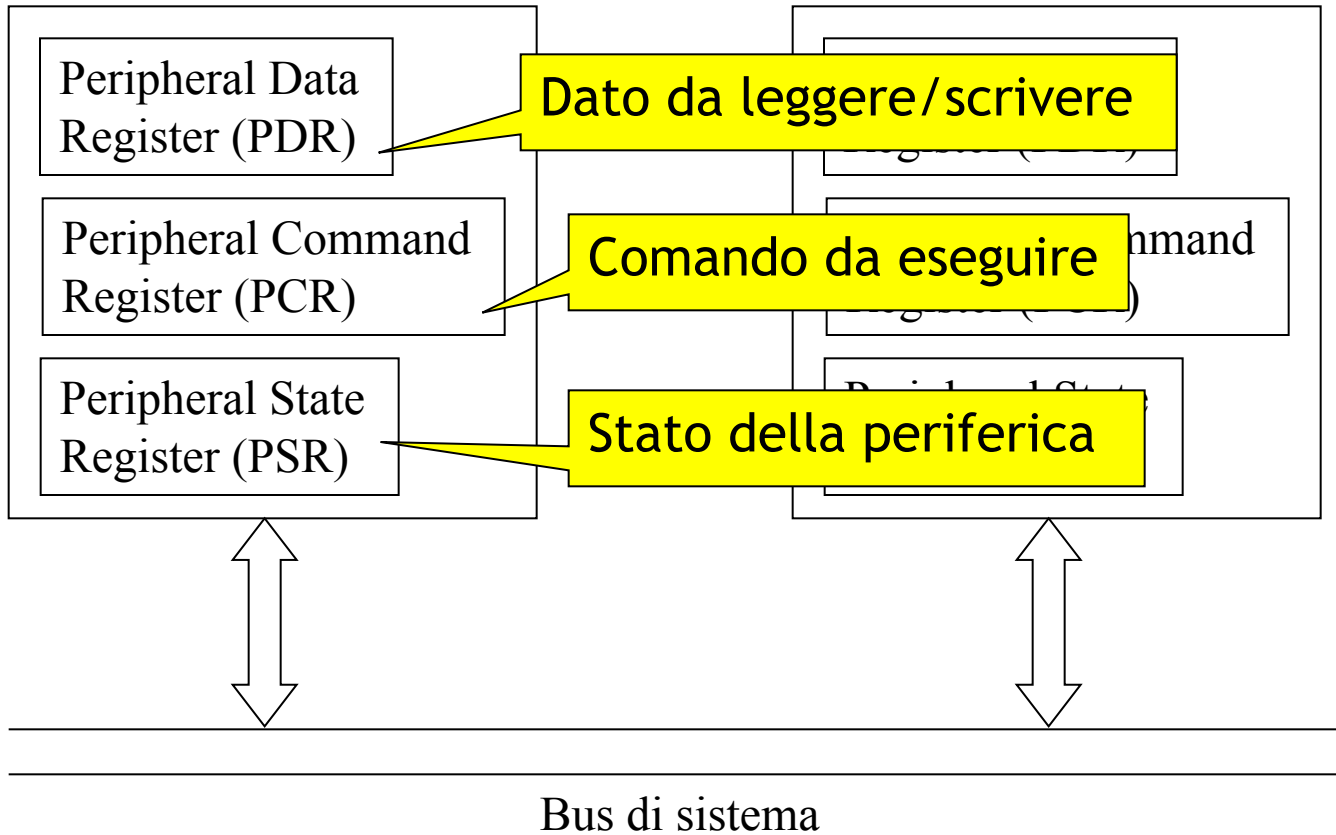
- Costituiscono gli elementi circuitali per il collegamento dell'elaboratore alle periferiche
- Contengono i registri per inviare comandi alla periferica, scambiare dati e controllare il funzionamento
- Molto diverse a seconda del tipo di periferica(alcune sono "intelligenti", dotate di proprie unità di controllo per convertire ed elaborare dati al posto della CPU)
- Contenuto delle interfacce:
 - **Registro dati della periferica**(PDR, Peripheral Data Register): consente lo scambio bidirezionale di dati con la periferica; collegato al bus dati
 - **Registro comando della periferica**(PCR, Peripheral Command Register): contiene il comando che la periferica stessa dovrà eseguire; collegato al bus controlli
 - **Informazione sullo stato** della periferica; può essere trasferita in un **registro di stato della periferica** (PSR, Peripheral Status Register) e letta a comando dalla CPU tramite il bus oppure direttamente collegato alla CPU con circuiti elettrici speciali(in tal caso l'informazione concorre a riempire **INTR**)



Le interfacce delle periferiche(2)

Interfaccia periferica 1

Interfaccia periferica 2





Es.: valutazione di espressione

- Vogliamo calcolare il valore dell'espressione:
 $(a+b) \cdot (c+d)$
leggendo i valori delle variabili **a**, **b**, **c**, **d** dal dispositivo di ingresso e scrivendo il risultato della valutazione sul dispositivo di uscita.



Un algoritmo generale

1. Leggi dal dispositivo di ingresso il valore delle variabili **a**, **b**, **c**, **d**
2. Somma il valore di **a** al valore di **b**
3. Salva il risultato parziale ottenuto
4. Somma il valore di **c** al valore di **d**
5. Moltiplica il risultato parziale appena ottenuto con quello precedentemente salvato
6. Scrivi sul dispositivo di uscita il risultato della valutazione complessiva
7. Termina l'esecuzione del programma.



L'algoritmo dettagliato (1)

1. Scrivi nella cella di memoria centrale riservata al valore della variabile **a** il valore letto dal dispositivo di ingresso (disponibile nel registro dati della periferica). Fai la stessa cosa per **b**, **c**, **d**
2. Somma il valore di **a** al valore di **b**
 - 2.1 Copia il contenuto della cella di memoria riservata ad **a** nel registro A
 - 2.2 Copia il contenuto della cella di memoria riservata a **b** nel registro B
 - 2.3 Somma il contenuto dei registri A e B
3. Salva il risultato parziale, contenuto nel registro A, in una cella di memoria predisposta per il risultato (**z**).



L'algoritmo dettagliato (2)

4. Somma il valore di c al valore di d
 - 4.1 Copia il contenuto della cella di memoria riservata a c nel registro A
 - 4.2 Copia il contenuto della cella di memoria riservata a d nel registro B
 - 4.3 Somma il contenuto dei registri A e B
5. Moltiplica il risultato parziale appena ottenuto con quello precedentemente salvato
 - 5.1 Copia il contenuto della cella riservata a z nel registro B (z e B contengono ora $a+b$, mentre A contiene $c+d$)
 - 5.2 Moltiplica il contenuto dei registri A e B.



L'algoritmo dettagliato (3)

6. Scrivi sul dispositivo di uscita il risultato della valutazione complessiva
 - 6.1 Memorizza il risultato appena calcolato (e disponibile nel registro A) nella cella di memoria riservata a z
 - 6.2 Copia il contenuto della cella di memoria riservata a z nel registro dati della periferica di uscita
7. Termina l'esecuzione del programma.



Tipologia delle operazioni svolte

- Operazioni di manipolazione
 - Somma: 2.3, 4.3
 - Moltiplicazione: 5.2
- Operazioni di trasferimento
 - Da periferica-input a MM: 1
 - Da MM a CPU: 2.1, 2.2, 4.1, 4.2, 5.1
 - Da CPU a MM: 3, 6.1
 - Da MM a periferica-output: 6.2



Forma binaria del programma

010000000010000	Leggi un valore dall'input e mettilo nella cella 16 (a)
010000000010001	Leggi un valore dall'input e mettilo nella cella 17 (b)
010000000010010	Leggi un valore dall'input e mettilo nella cella 18 (c)
010000000010011	Leggi un valore dall'input e mettilo nella cella 19 (d)
000000000010000	Carica il contenuto della cella 16 (a) nel registro A
0001000000010001	Carica il contenuto della cella 17 (b) nel registro B
0110000000000000	Somma i registri A e B
0010000000010100	Scarica il contenuto di A nella cella 20 (z) (ris.parziale)
000000000010010	Carica il contenuto della cella 18 (c) nel registro A
0001000000010011	Carica il contenuto della cella 19 (d) nel registro B
0110000000000000	Somma i registri A e B
0001000000010011	Carica il contenuto della cella 20 (z) (ris. parziale) in B
1000000000000000	Moltiplica i registri A e B
0010000000010100	Scarica il contenuto di A nella cella 20 (z) (ris. totale)
0101000000010100	Scrivi il contenuto della cella 20 (z) (ris. totale) in output
1101000000000000	Halt



Programma in memoria centrale

Cella 0	010000000010000
1	010000000010001
2	010000000010010
3	010000000010011
4	000000000010000
5	000100000010001
6	011000000000000
7	001000000010100
8	000000000010010
9	000100000010011
10	011000000000000
11	000100000010011
12	100000000000000
13	001000000010100
14	010100000010100
15	110100000000000
16	
17	
18	
19	
20	

Spazio riservato per **a**

Spazio riservato per **b**

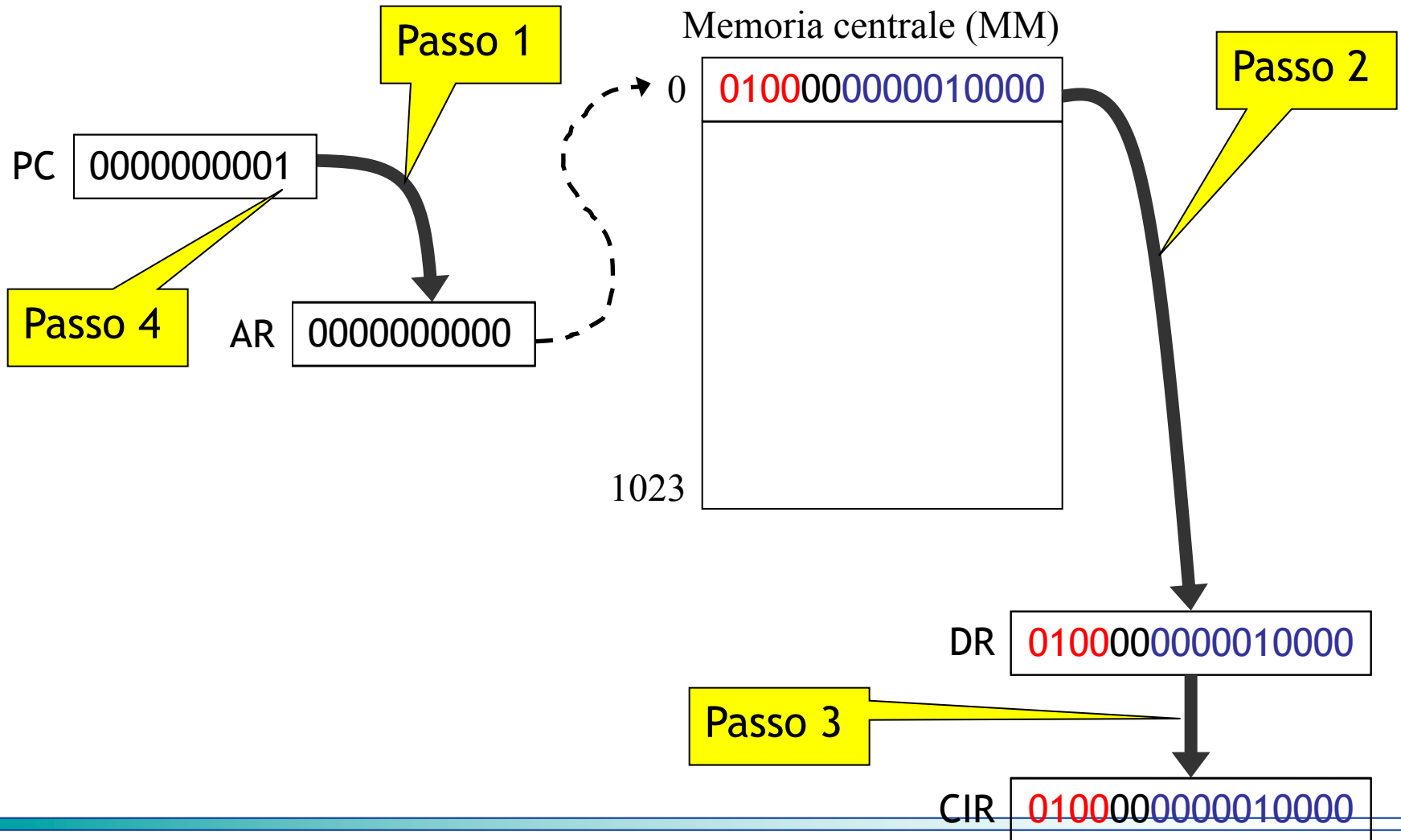
Spazio riservato per **c**

Spazio riservato per **d**

Spazio riservato per **z**



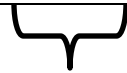
Fase di fetch 1^a istruzione





Fase di interpretazione 1^a istruzione

CIR 0100000000010000



Codice operativo 0100 = leggi da input



Fase di esecuzione 1^a istruzione

