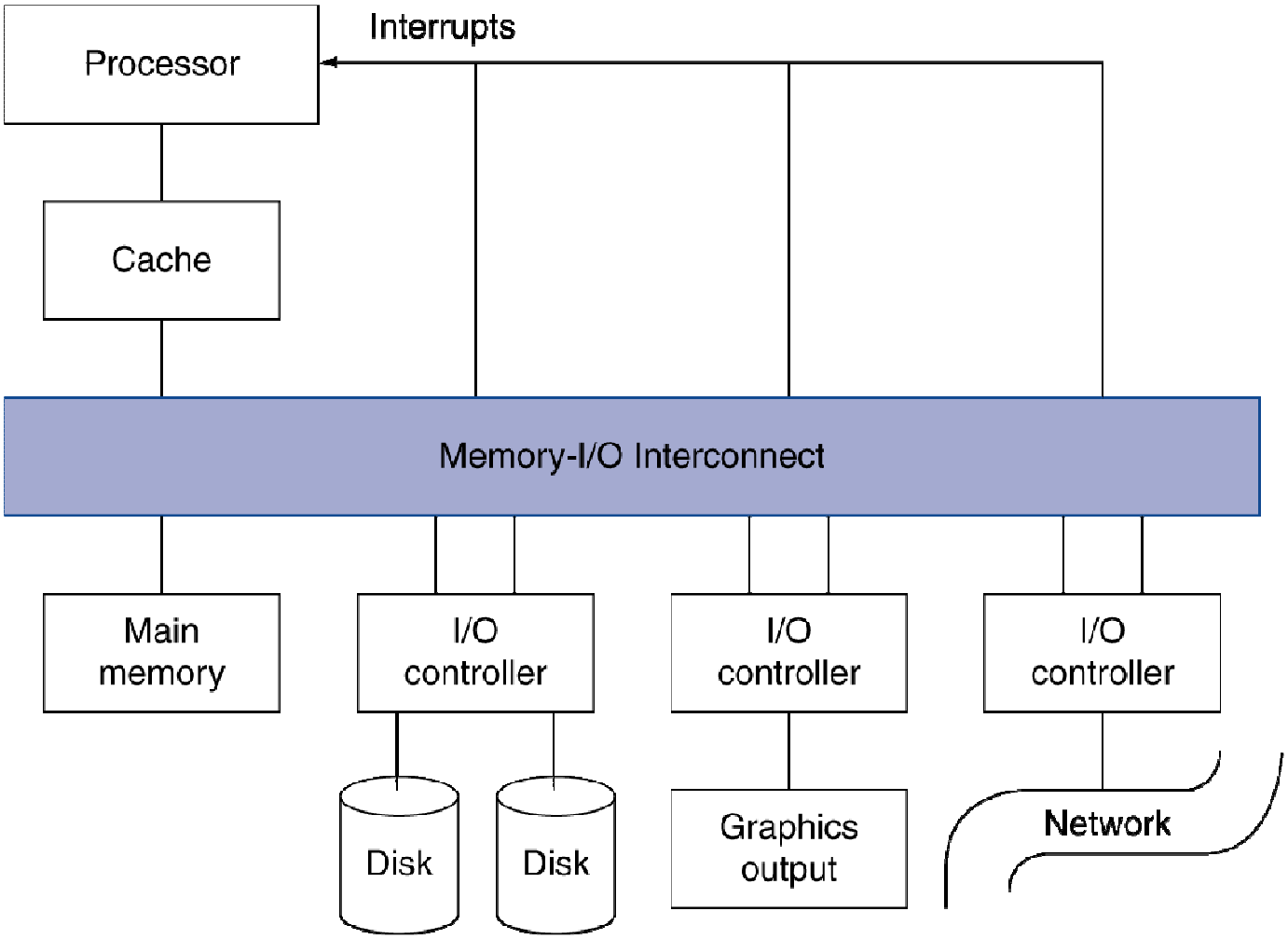
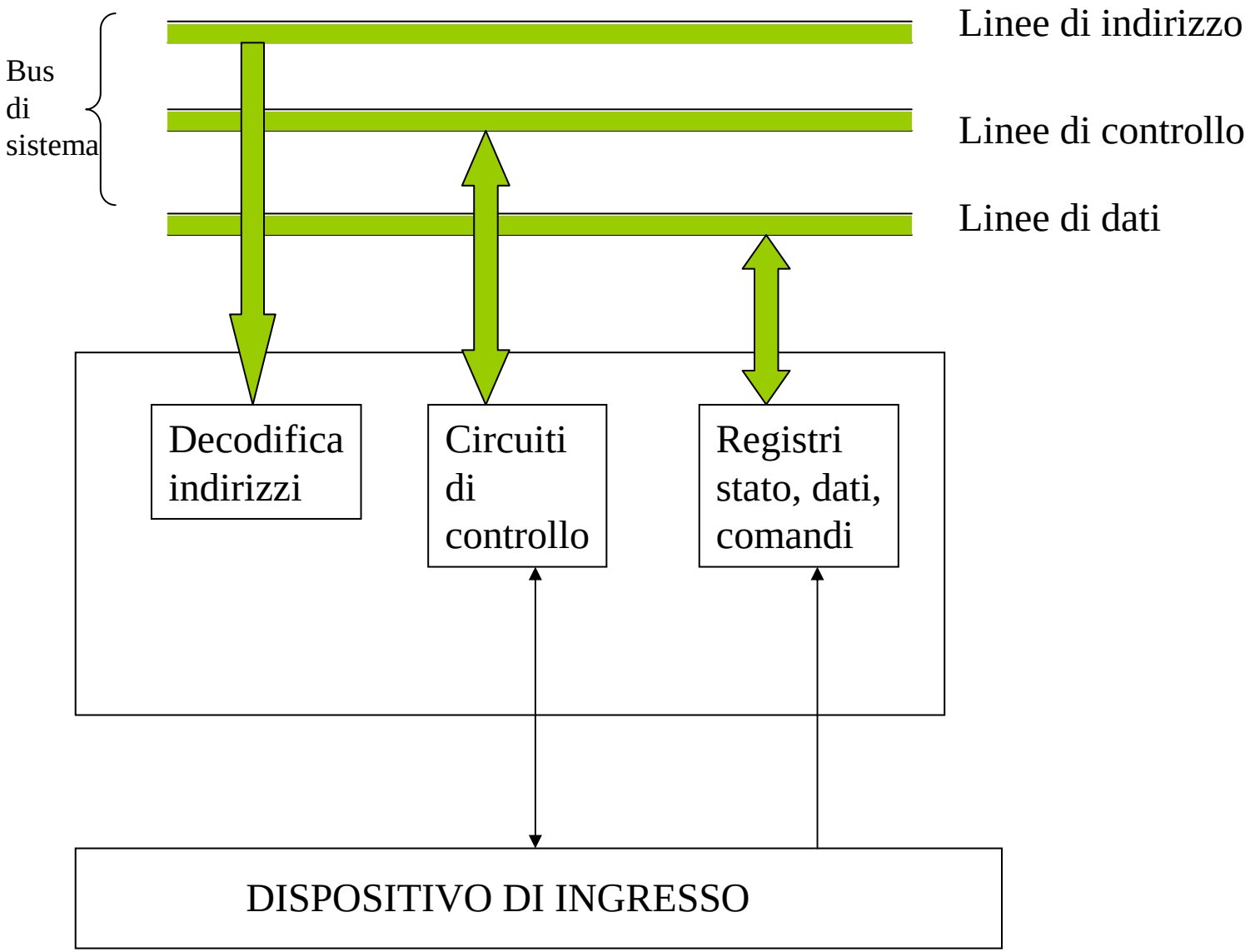
Calcolatori Elettronici A
a.a. 2008/2009

GESTIONE DELL'INPUT/OUTPUT
(cenni)

Massimiliano Giacomin


Processor
Interrupts
Cache
Memory-I/O Interconnect
Main memory
I/O controller
I/O controller
I/O controller
Disk
Disk
Graphics output
Network

Dispositivo di ingresso e interfaccia



- Dispositivi di I/O gestiti dai controller hardware
  - Trasferiscono dati da/a dispositivo
  - Sincronizzano operazioni con software
- Command register
  - Per determinare operazioni del dispositivo
- Status registers
  - Indicano eventi particolari, errori, operazioni in corso
- Data registers
  - Write: trasferimento dati al dispositivo
  - Read: trasferimento dati dal dispositivo

PROBLEMA

- Come far sì che un indirizzo venga riconosciuto dal dispositivo e non dalla memoria? [ovvero: spazio indirizzamento memoria vs. I/O]

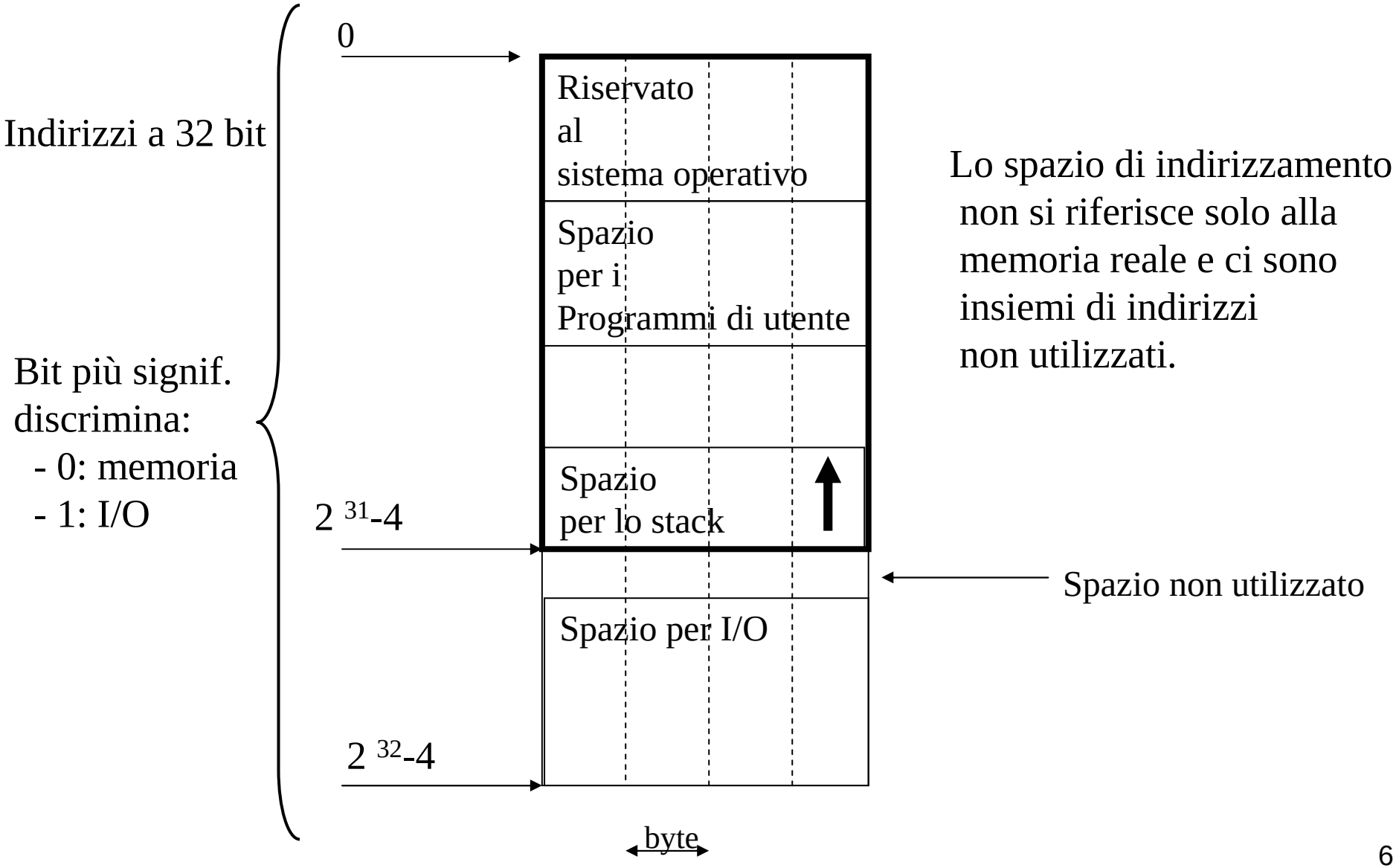
Due soluzioni:

1. **Memory mapped**:
   - I registri dei controller hanno indirizzi che appartengono allo spazio indirizzi di memoria.
   - I registri dei controller sono indirizzati come se fossero locazioni di memoria: uso le istruzioni per il normale trasferimento tra registri e memoria per trasferire dati dai buffer delle periferiche ai registri di CPU (e memoria centrale) o viceversa.
   - La memoria "ignora" indirizzi che non corrispondono a proprie locazioni
2. **Istruzioni speciali di I/O**:
   - Generano opportuni segnali sulle linee di controllo

     $\Rightarrow$ spazio di indirizzi di I/O distinto dallo spazio indirizzi di memoria

Esempio di spazio di indirizzamento

con I/O memory mapped

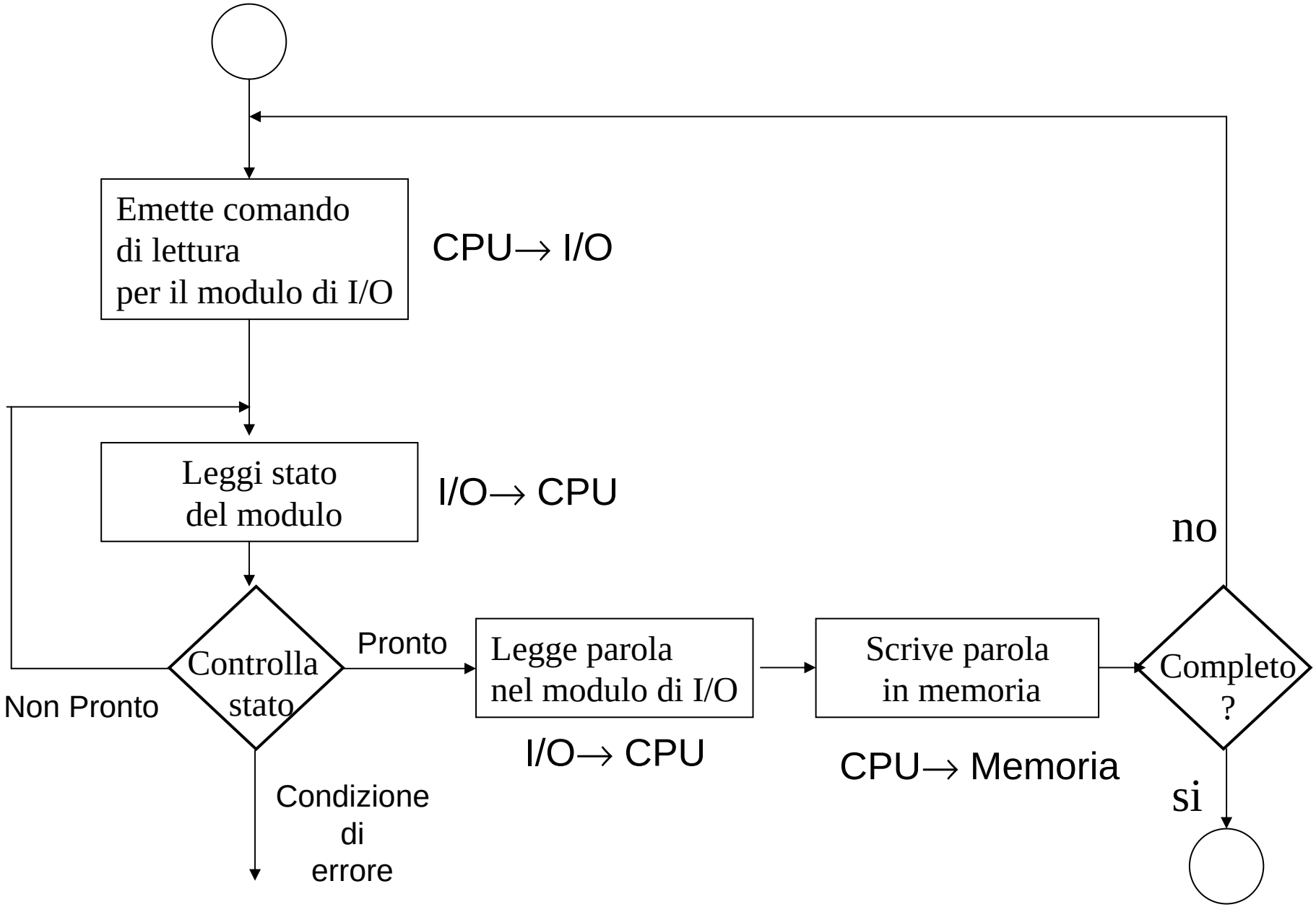


Lo spazio di indirizzamento non si riferisce solo alla memoria reale e ci sono insiemi di indirizzi non utilizzati.

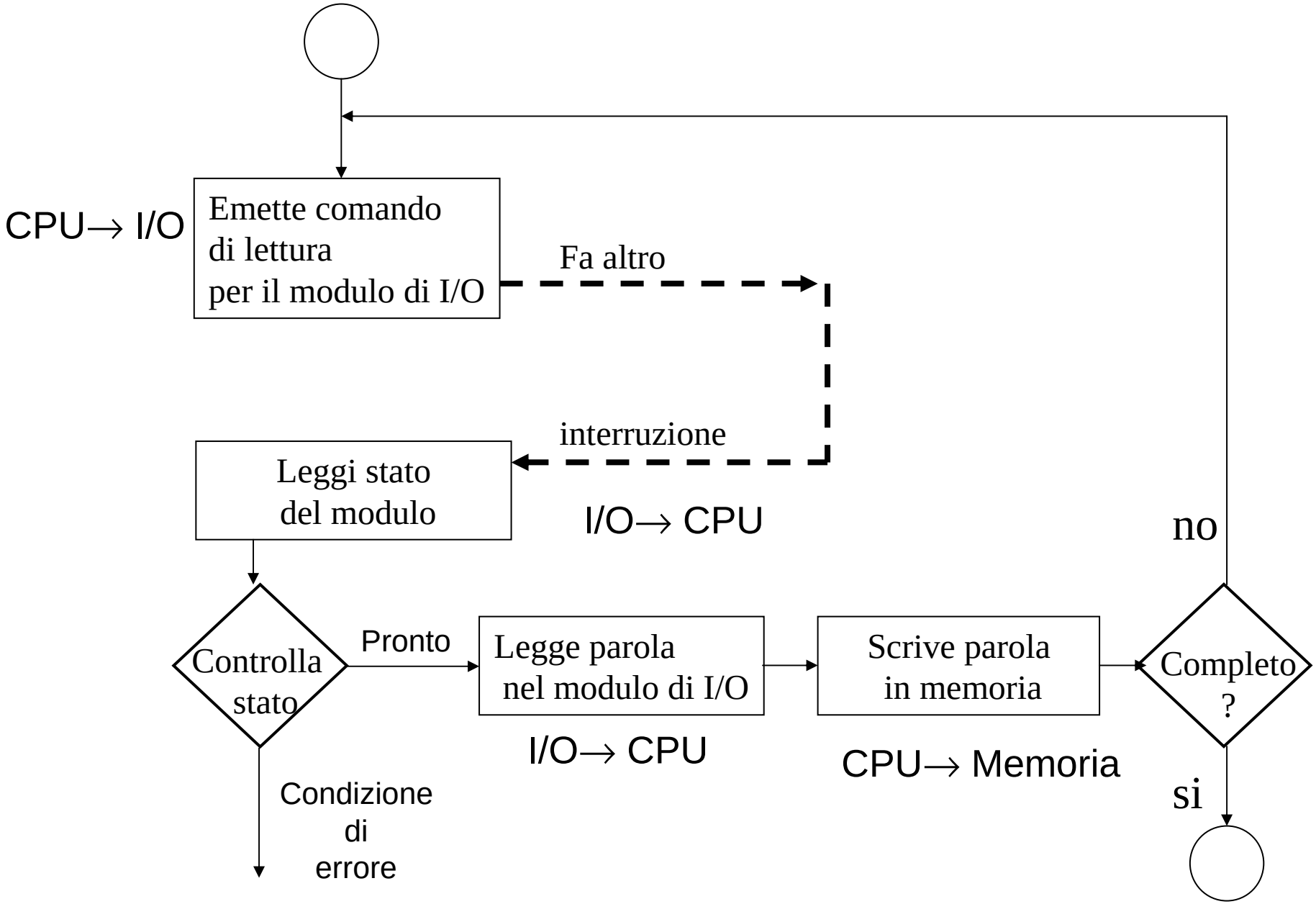
TRE MODALITA’ PER LA SINCRONIZZAZIONE

- CONTROLLO DI PROGRAMMA
- INTERRUPT
- DMA

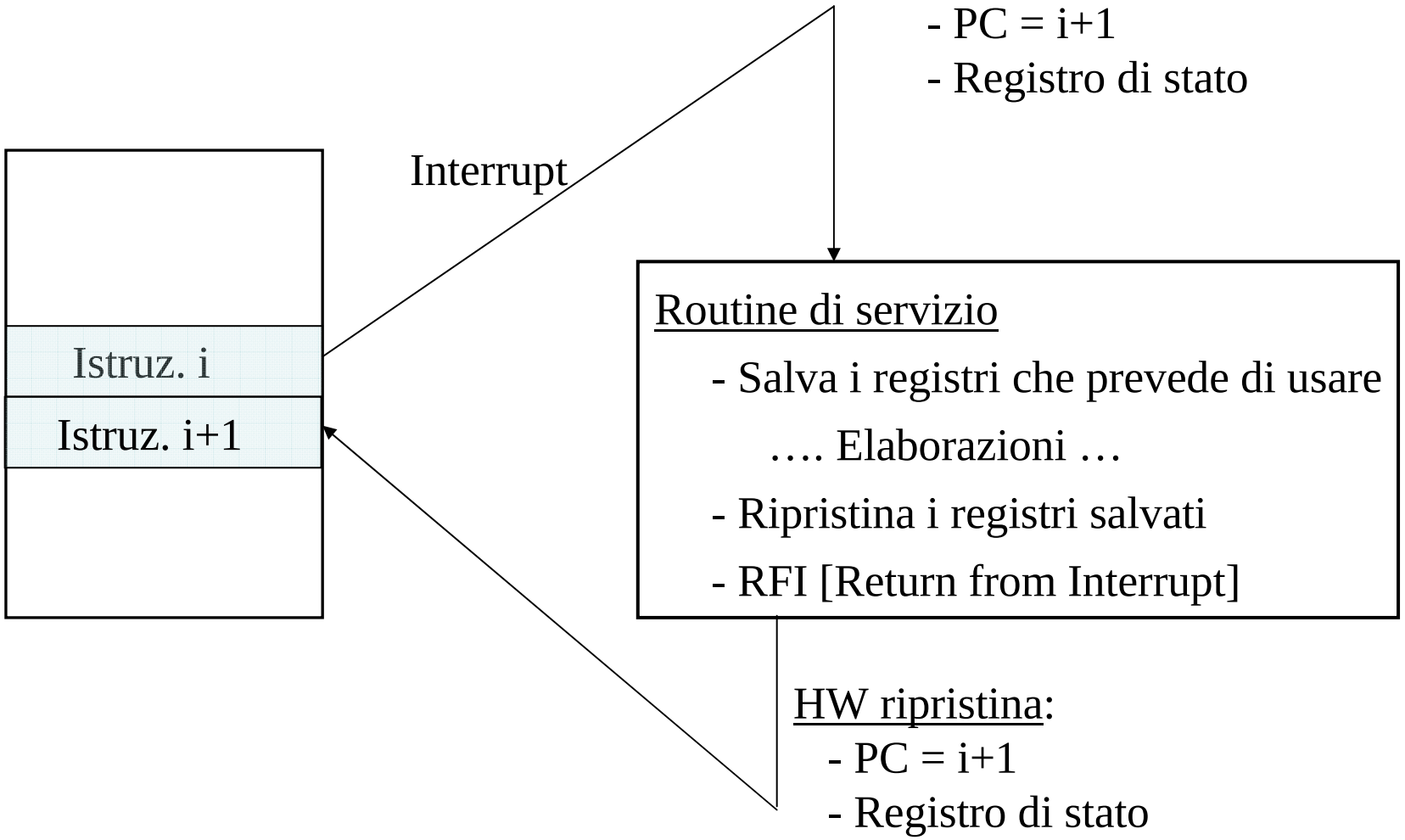
Schema tipo per la lettura di dati con controllo di programma



Con interrupt



GESTIONE INTERRUPT:



GESTIONE DELL'INTERRUPT

- Il dispositivo richiede interruzione con segnale di controllo su bus *interrupt request* [INTR]
- Il processore risponde con *segnale di riscontro* INTA [Interrupt Acknowledge]
- Ricevuto questo segnale, il dispositivo toglie il segnale INTR

A questo punto: il processore deve passare il controllo alla *procedura di servizio dell'interrupt [Interrupt Service Routine ISR]*

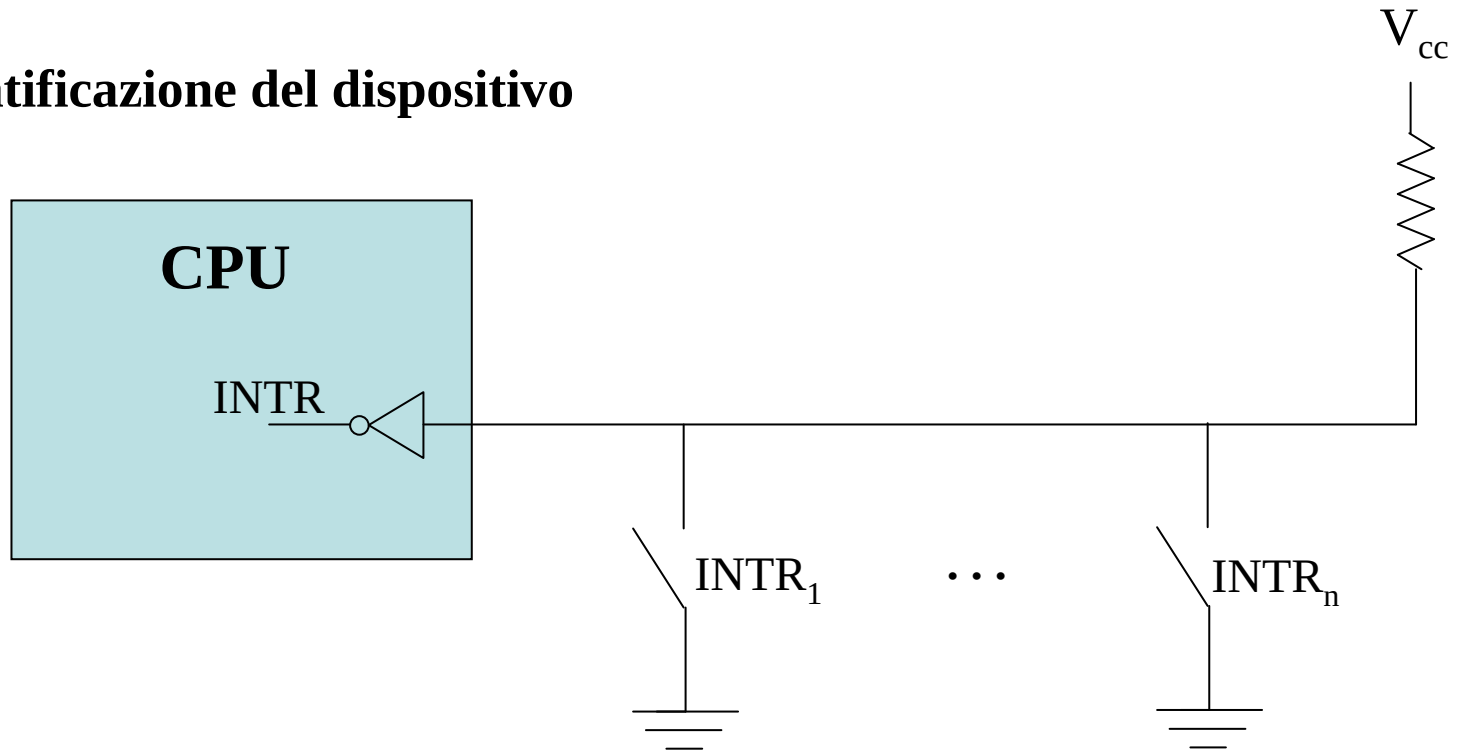
Terminata la procedura: il processore restituisce il controllo alla procedura originaria [istruzione i+1]

NB: differenza con procedura: il programma interrotto non "sa" cosa viene modificato – cambia il concetto di "responsabilità",
es. registri temporanei devono essere ripristinati!

⇒ HW deve salvare almeno:
- Program Counter [modificato cedendo il controllo]
- **Registro di stato** del processore [**PS**]

Di solito, al resto pensa la routine di servizio!

Identificazione del dispositivo



$INTR = INTR_1 + INTR_2 + \ldots INTR_n$

<u>Metodo 1</u>: Polling (Interrogazione)

- ISR controlla i bit IRQ in tutte le interfacce dei dispositivi
- Il primo che ha IRQ=1 viene servito

⇒ Facile da realizzare, ma spreca tempo

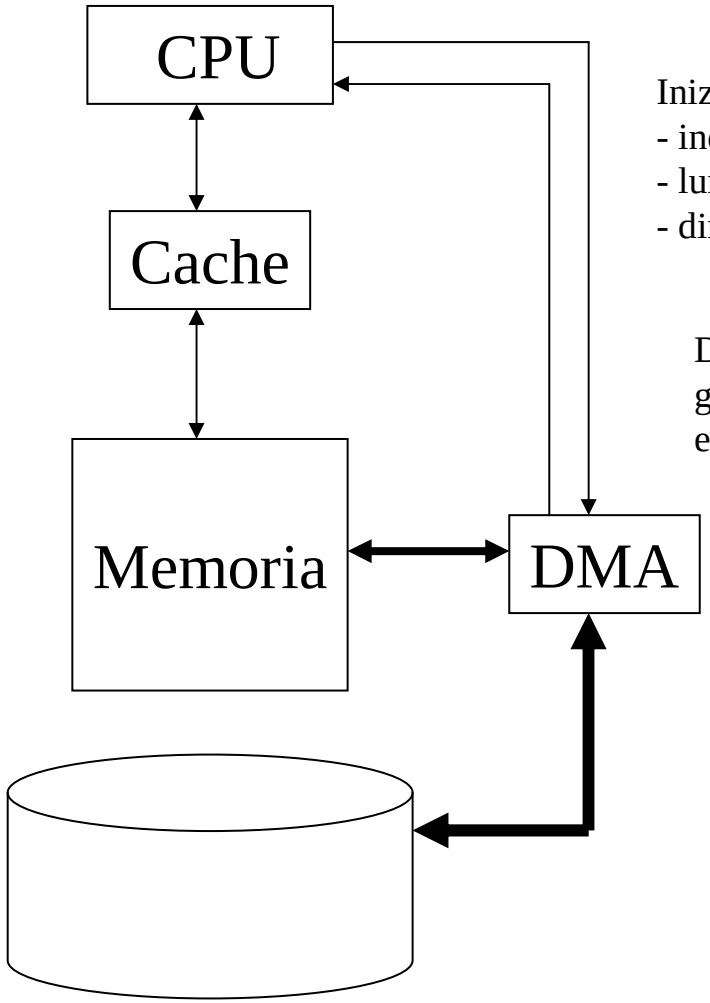
<u>Metodo 2</u>: Interrupt vettorizzati

- dispositivo: INTR
- CPU: INTA [segnala che è pronto a trattare l'interrupt]
- dispositivo: invia un codice di identificazione attraverso linee dati del bus (vettore di identificazione) e interrompe INTR
- CPU accede ad una tabella con corrispondenza codice $\rightarrow$ indirizzo $ISR_{dispositivo}$

NB: finchè INTA non è attivato, il dispositivo non ha il diritto di accedere al bus
[quando arriva INTR, può esserci istruzione in fase di esecuzione, oppure gli interrupt potrebbero essere disabilitati durante una routine di servizio]

NB2: sono anche possibili schemi misti
[gruppi di dispositivi riconosciuti con vettori, all'interno di un gruppo uso del polling]

DMA (Direct Memory Access)



Inizio CPU invia
- indirizzo di partenza
- lunghezza del blocco da trasferire
- direzione (I/O)

DMA gestisce trasferimento inviando gli opportuni comandi a dispositivo e memoria [modo “programmato”]

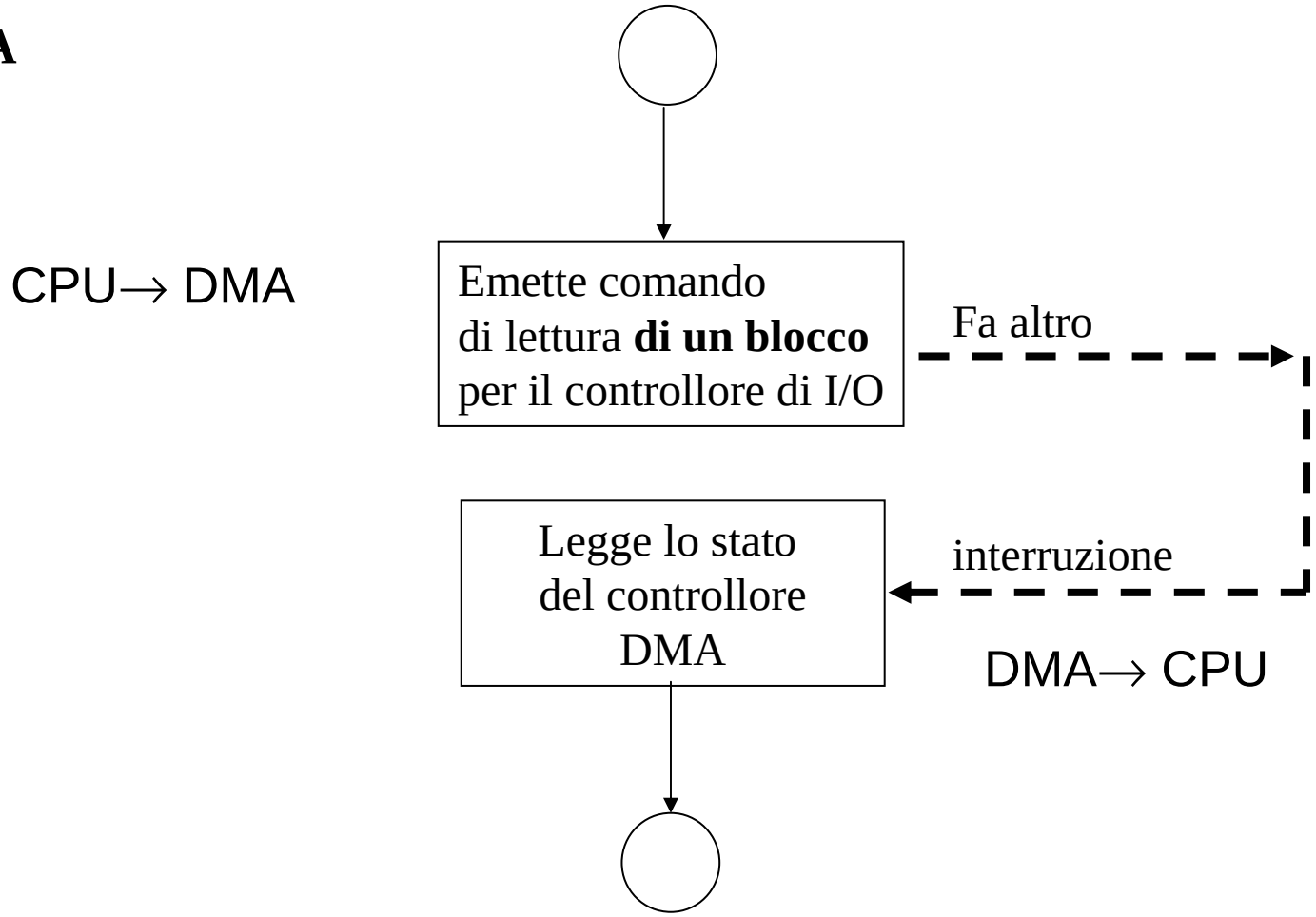
Fine:
DMA invia interrupt per informare fine operazione

Il processore non è impegnato nell’acquisizione del singolo dato

[P.es. salvataggio in memoria, incremento indice, salvataggio e ripristino stato,…]

ma solo all’inizio e alla fine del trasferimento di un blocco!

I/O con DMA



NB: processore dialoga con DMA come con un normale dispositivo I/O;
controllore DMA ha vari registri:

- stato e controllo [bit IRQ, IE, R/W, Done]
- indirizzo di partenza
- contatore di parole