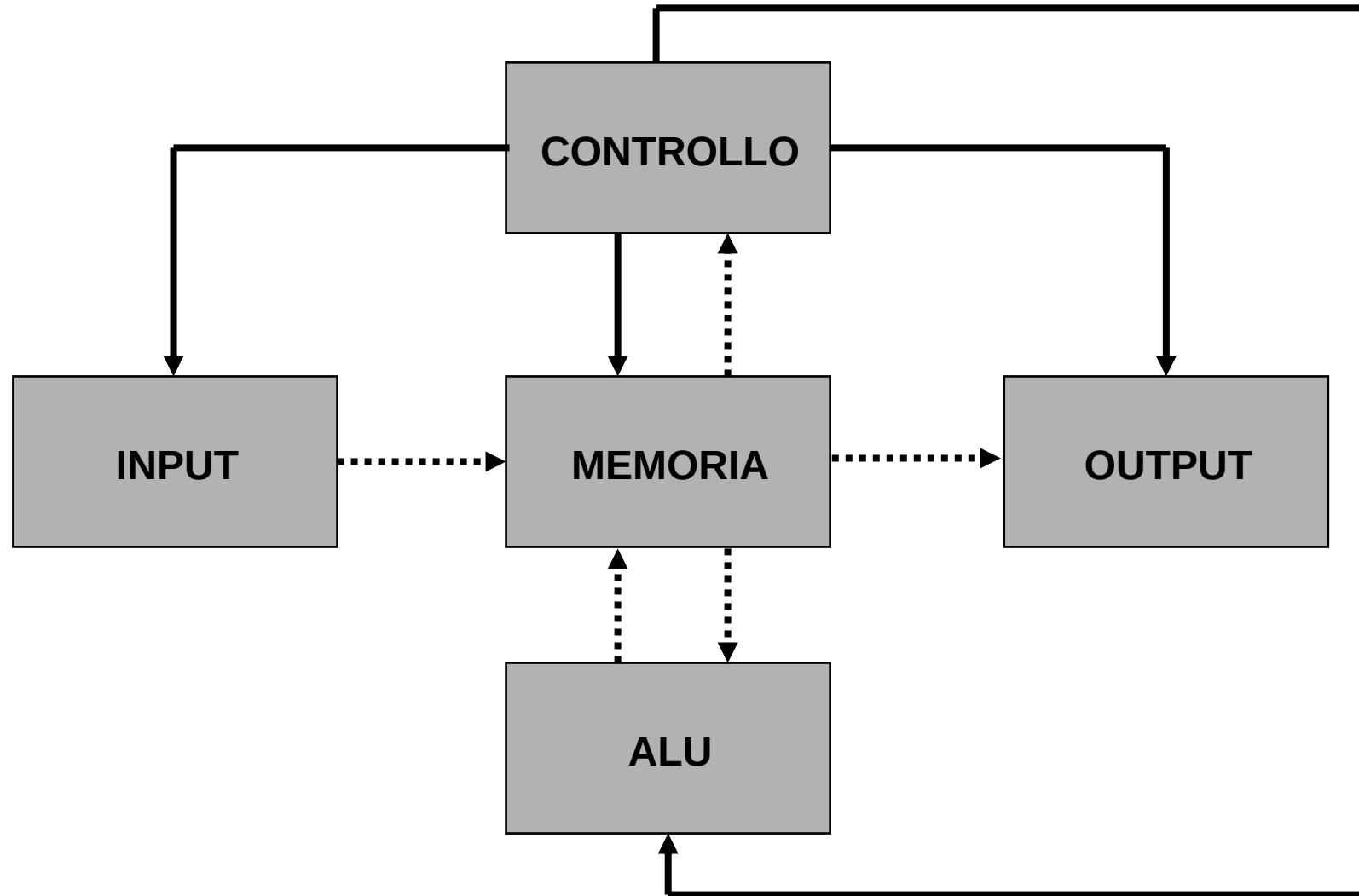

Architettura dei Calcolatori

Corso di Calcolatori Elettronici I

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II

Modello di Von Neumann



Modello di Von Neumann

Flussi:

- **Controllo**
- **Dati**

Fasi:

- **Caricamento**
- **Esecuzione**

Programma:

- **Sequenza di istruzioni**

Istruzione:

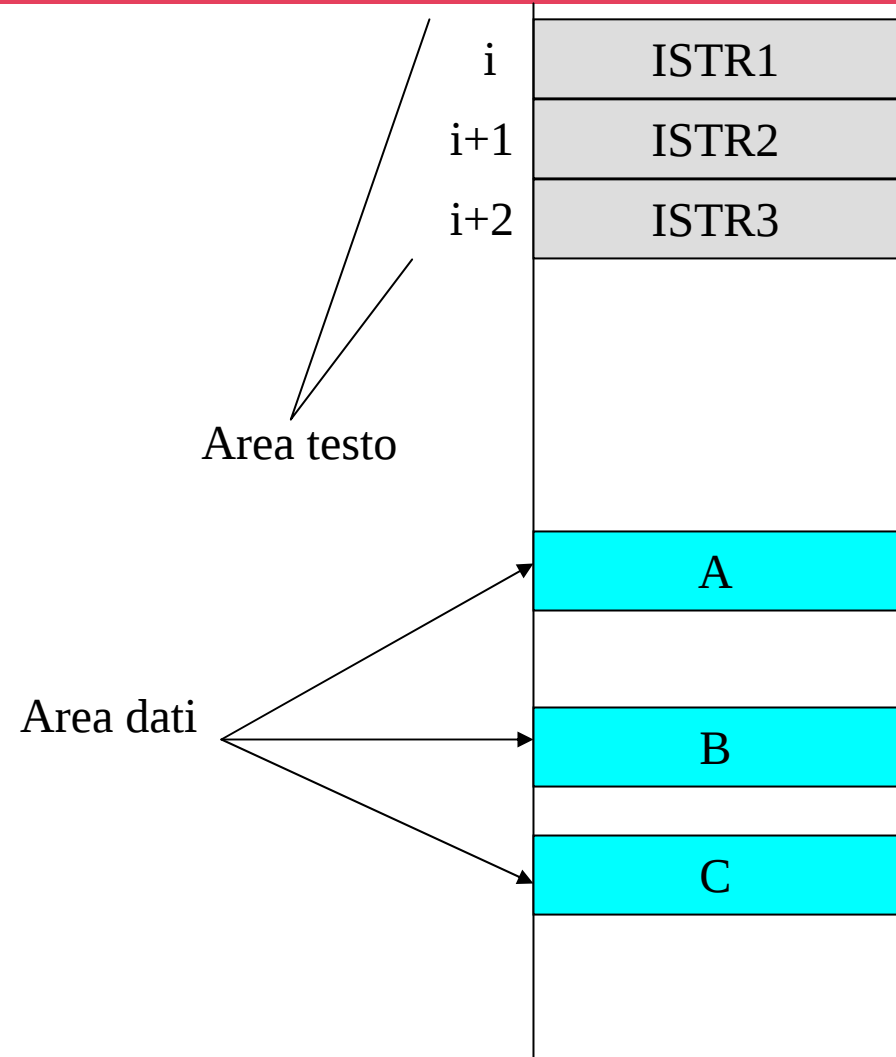
- **Azione elaborativa propria del sistema calcolatore, espressa in un linguaggio prefissato**

Unità di controllo

➤ Opera in un ciclo infinito:

1. Prelievo
2. Esecuzione

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



Catteristiche della Macchina di Von Neumann

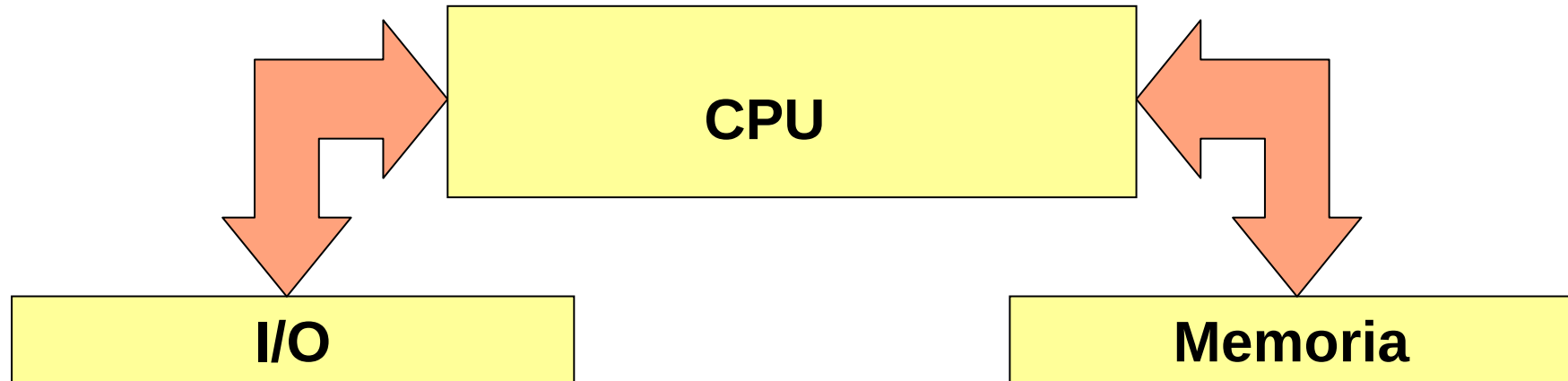
- Elaborazione sequenziale
 - » le istruzioni vengono caricate ed eseguite una *dopo* l'altra
- Linguaggio di programmazione imperativo
 - » le istruzioni indicano al processore *cosa* fare, ovvero quali sono le specifiche operazioni da eseguire sui dati
- Memoria comune contenente dati e programma
 - » le istruzioni ed i dati su cui queste lavorano sono contenuti in un'unica memoria a cui il processore accede
 - » un modello leggermente diverso, detto *Harvard*, prevede memorie fisicamente distinte per dati ed istruzioni

Macchina di Von Neumann: alcune definizioni

- Stato dei dati
 - » (A, V) con: $A = \{a_1, a_2, \dots, a_n\}$, $V = \{v_1, v_2, \dots, v_n\}$
- 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

Sottosistemi di un Calcolatore Elettronico

- Processore o CPU (*Central Processing Unit*)
- memoria centrale
- sottosistema di input/output (I/O)



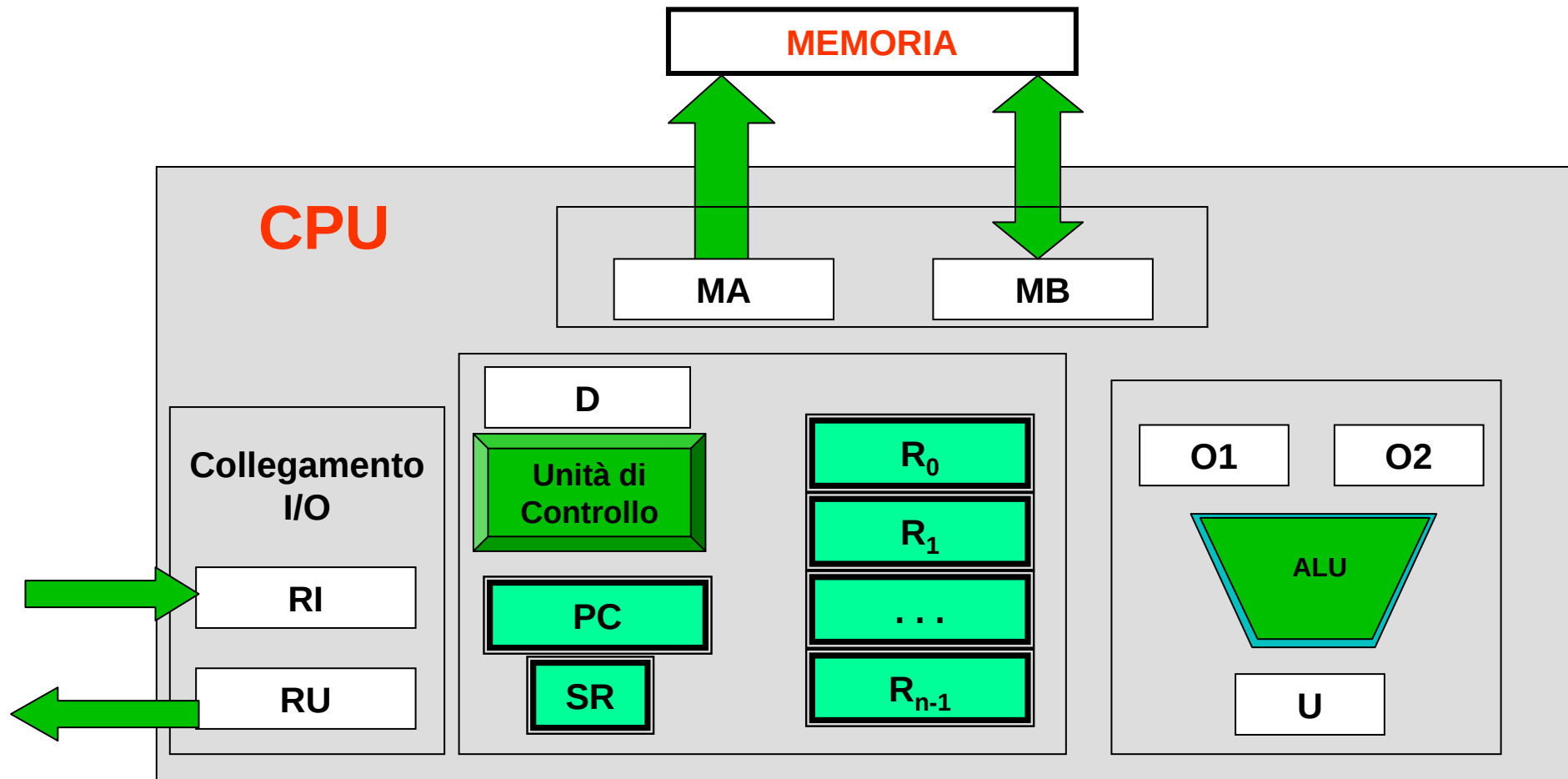
Il processore

- Il processore, o CPU, rappresenta l'Unità di Controllo del modello di Von Neumann
- 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
- Le istruzioni sono prelevate una ad una dalla memoria
- I *dati* di ingresso e di uscita dell'elaborazione e la sequenza di istruzioni da eseguire (programma) sono immagazzinati nella memoria centrale

Componenti fondamentali della CPU

- Registro **Program Counter** (PC) o Prossima Istruzione
- **Instruction Register** (IR) o Registro di decodifica (D)
- Registri di uso generale $R_0, \dots, R_{n-1}$
- Registro di stato (SR)
- Collegamento con la memoria
 - » Registro Memory Address (MA)
 - » Registro Memory Buffer (MB)
- Collegamento con i dispositivi di input/output (I/O)
 - » Registro di input (RI) e Registro di output (RO)
- Unità aritmetico/logica (ALU)
 - » Registri di appoggio per la ALU (O1, O2, U)
- Unità di controllo

Modello architetturale di un processore



Processore a registri generali

- Il processore dispone di un set di registri $R_0, R_1, \dots, R_{N-1}$ utilizzabili indifferentemente
- Le istruzioni che operano su registri sono più veloci di quelle che operano su locazioni di memoria
- Il programmatore può utilizzare i registri del processore per memorizzare i dati di uso più frequente (concetto di gerarchia di memorie)
- Istruzioni con operandi registri:
 $[R_0] + [R_1] \rightarrow R_1$
- Istruzioni con operandi memoria-registri:
 $[R_0] + M[1000] \rightarrow R_0$ *memory-to-register*
 $M[1000] + [R_1] \rightarrow M[1000]$ *register-to-memory*

Funzioni dei registri di macchina

- Indirizzo dell'istruzione corrente (PC)
- Transito dati (qualunque registro generale)
- Accumulazione di risultati
 - » es: $R0 := \text{NOT } R0$ oppure $R0 := R0 + R1$
- Indirizzamento
- Indicatori o flag (Registro di Stato)
- Altre funzioni speciali

Registri del processore

➤ **Registri interni**

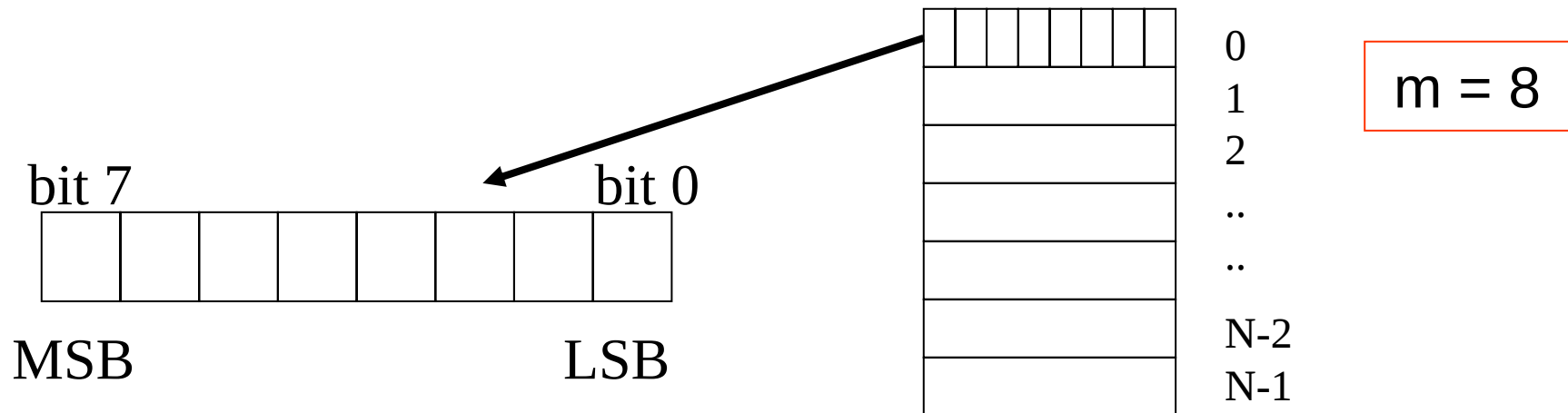
- » Necessari al funzionamento del processore
- » Non direttamente visibili al programmatore (non appartengono al *modello di programmazione*)

➤ **Registri di macchina**

- » Visibili al programmatore (appartengono al *modello di programmazione*)
 - Registri generali
 - Registri speciali

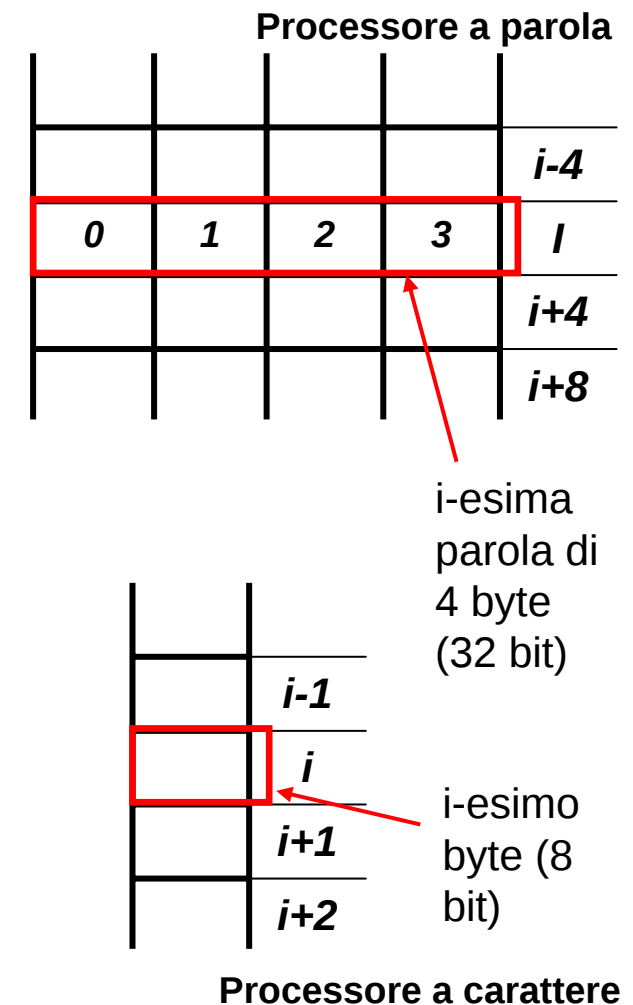
La memoria centrale

- 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 $N = 2^c$
 - » $[0, N-1] = \text{SPAZIO DI INDIRIZZAMENTO}$
- 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

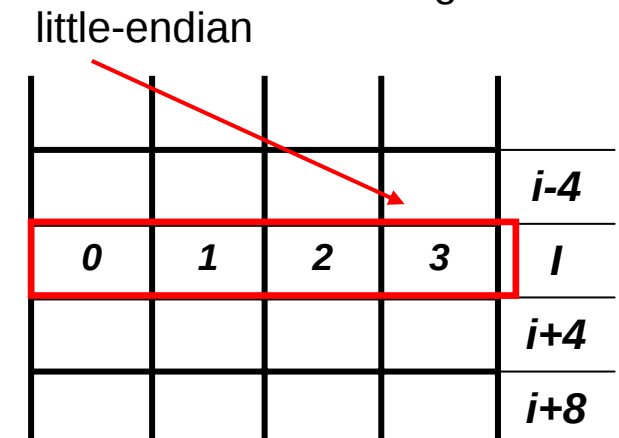
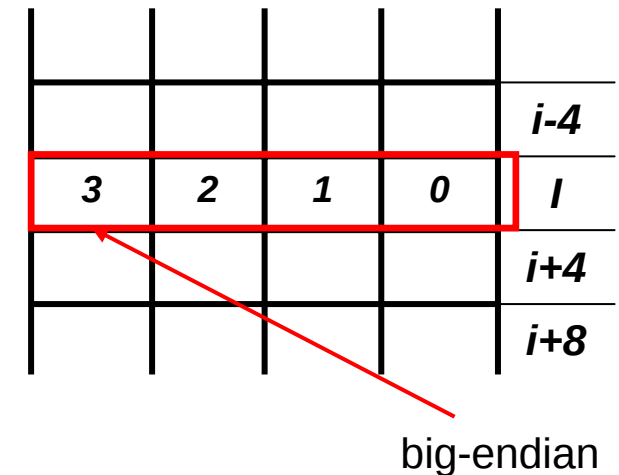
- I processori “a parola” hanno la memoria organizzata in locazioni (parole o *word*) di 16 bit, 32 bit o 64 bit
- I processori “a carattere” accedono alla memoria con un parallelismo di 1 byte (8 bit)
- La maggior parte dei sistemi moderni accede alla memoria con un parallelismo di “parole” da 16, 32 o 64 bit, ma l’unità indirizzabile di memoria (locazione) è ancora il byte (sistemi a memoria *byte-addressable*)
 - » in altre parole, ogni byte ha il suo indirizzo. Parole fatte -ad esempio- di 4 byte hanno indirizzi consecutivi che differiscono di 4



Processori big-endian e little-endian

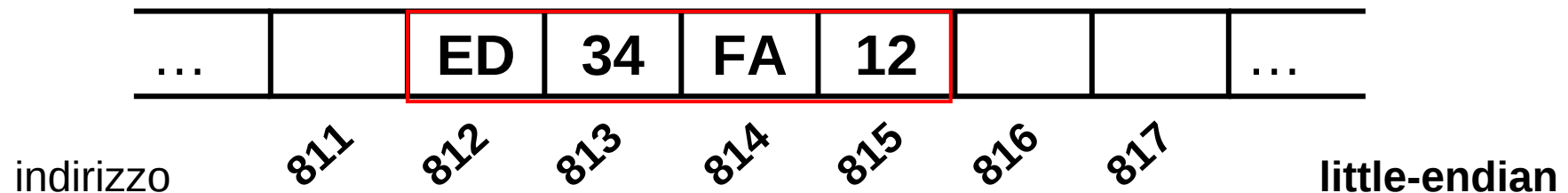
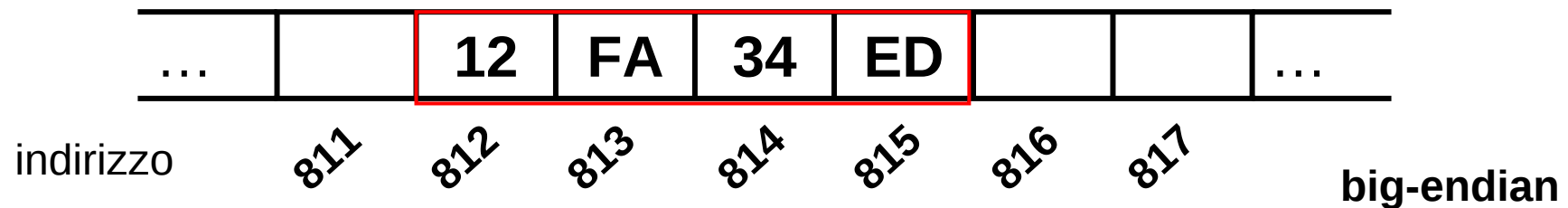
- I processori “a parola” hanno possono organizzare i byte all'interno della parola in diverse maniere
- In un processore *big-endian*, il byte più significativo del numero rappresentato nella parola occupa il primo indirizzo (il più basso) in memoria, mentre quello meno significativo il più alto
- viceversa per i processori *little-endian*

indirizzi byte crescenti →



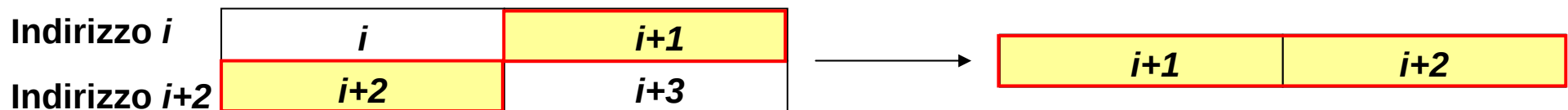
Big-endian e little-endian: un esempio

- Immaginiamo di avere un processore a parola, con parole di 32 bit (4 byte) e voler scrivere in memoria il valore intero (esadecimale) **12FA34ED** all'indirizzo **812**
- Le figure sottostanti illustrano il contenuto della memoria nei due casi big-endian e little-endian



Memoria: parole allineate e non allineate

- 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

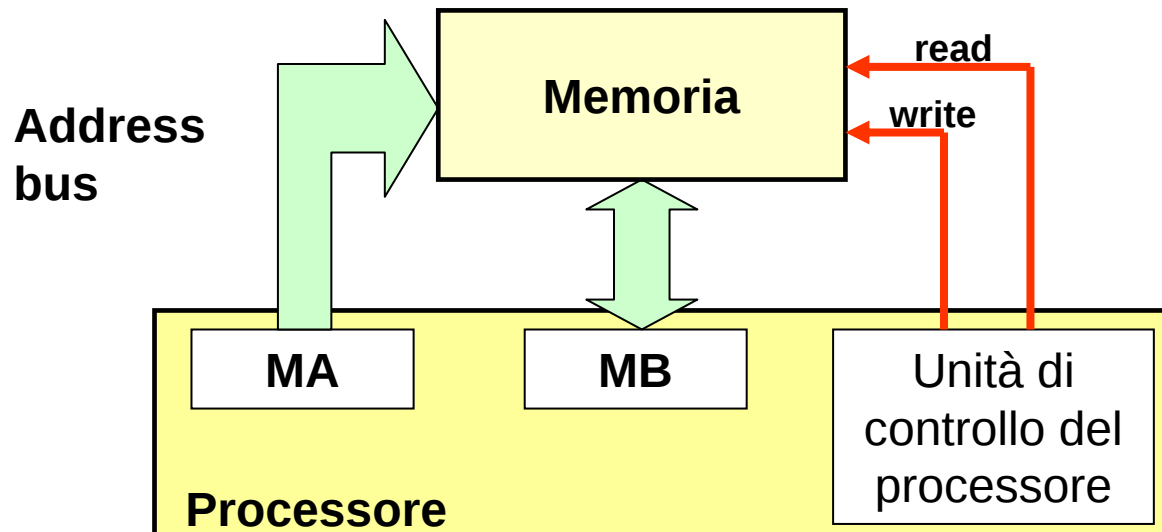


(i pari)

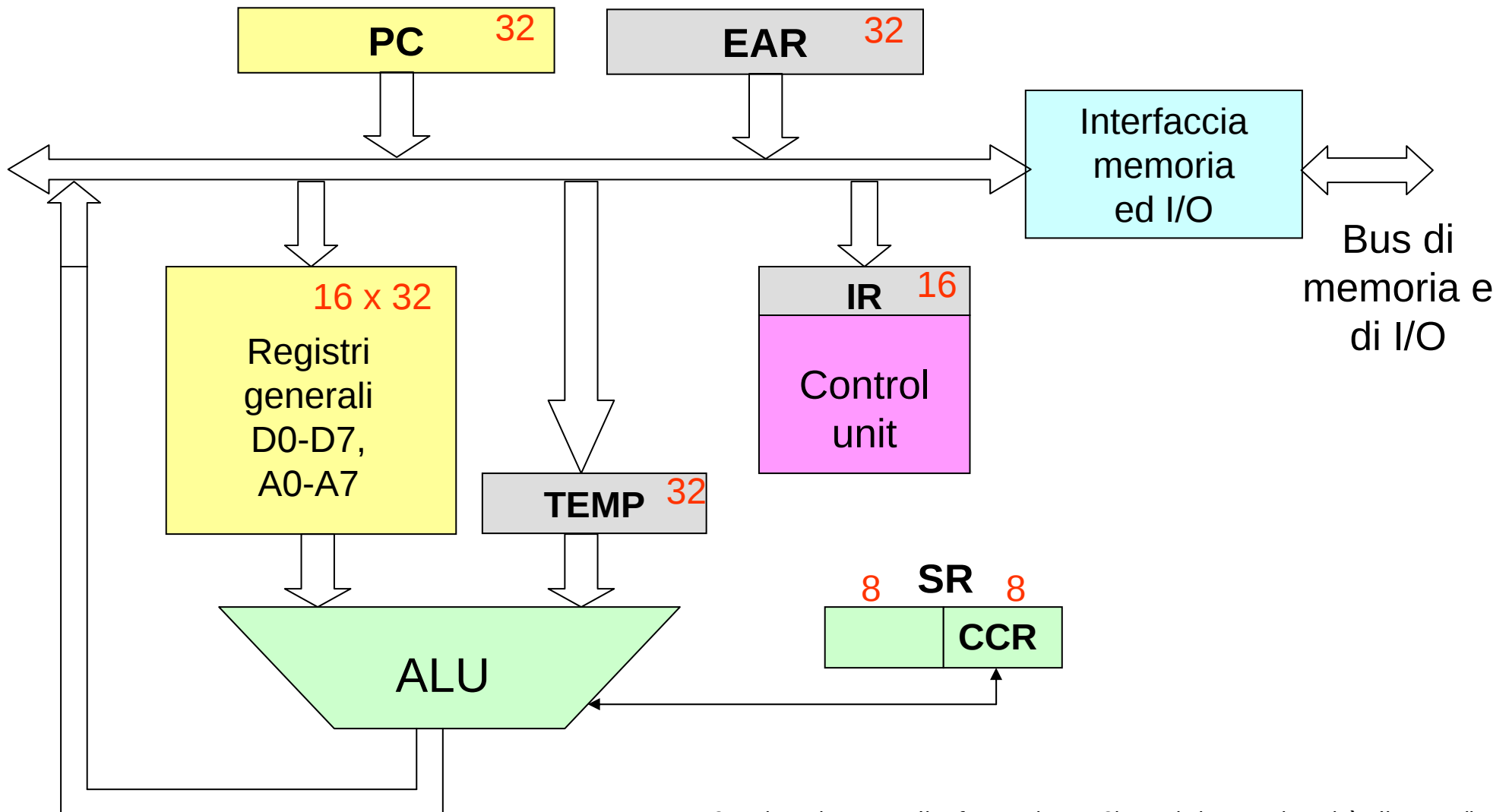
La parola di 16 bit formata dai due byte **ombreggiati** non è allineata sul limite di parola (indirizzo multiplo di 2)

Interazione processore-memoria

- Registro Memory Address (MA)
 - » fornisce alla memoria l'indirizzo del dato da leggere/scrivere
- Registro Memory Buffer (MB)
 - » contiene il dato da scrivere verso, o da leggere *dalla* memoria
- Segnali di controllo: *read / write*
 - » ordinano alla memoria operazioni di lettura oppure di scrittura



Architettura del processore MC 68000



Caratteristiche del processore MC68000 – 1/2

➤ Dati:

» all'esterno:

- ◆ parola di 16 bit (16 pin per i dati)
- ◆ necessita di due letture per acquisire una parola di 32 bit

» all'interno:

- ◆ registri di 32 bit

➤ Indirizzi:

» all'esterno:

- ◆ bus di 24 bit (spazio di indirizzamento fisico $2^{24} = 16\text{M}$)
 - 512 pagine (2^9) da 32K (2^{15})

» all'interno:

- ◆ bus di 32 bit

Caratteristiche del processore MC68000 – 2/2

- Parallelismo della memoria:
 - » Parole di 16 bit, ognuna costituita da due byte con indirizzi distinti (memoria byte addressable)
- Convenzioni della memoria:
 - » Una parola deve essere allineata ad un indirizzo pari (even boundary)
 - » Convenzione big-endian