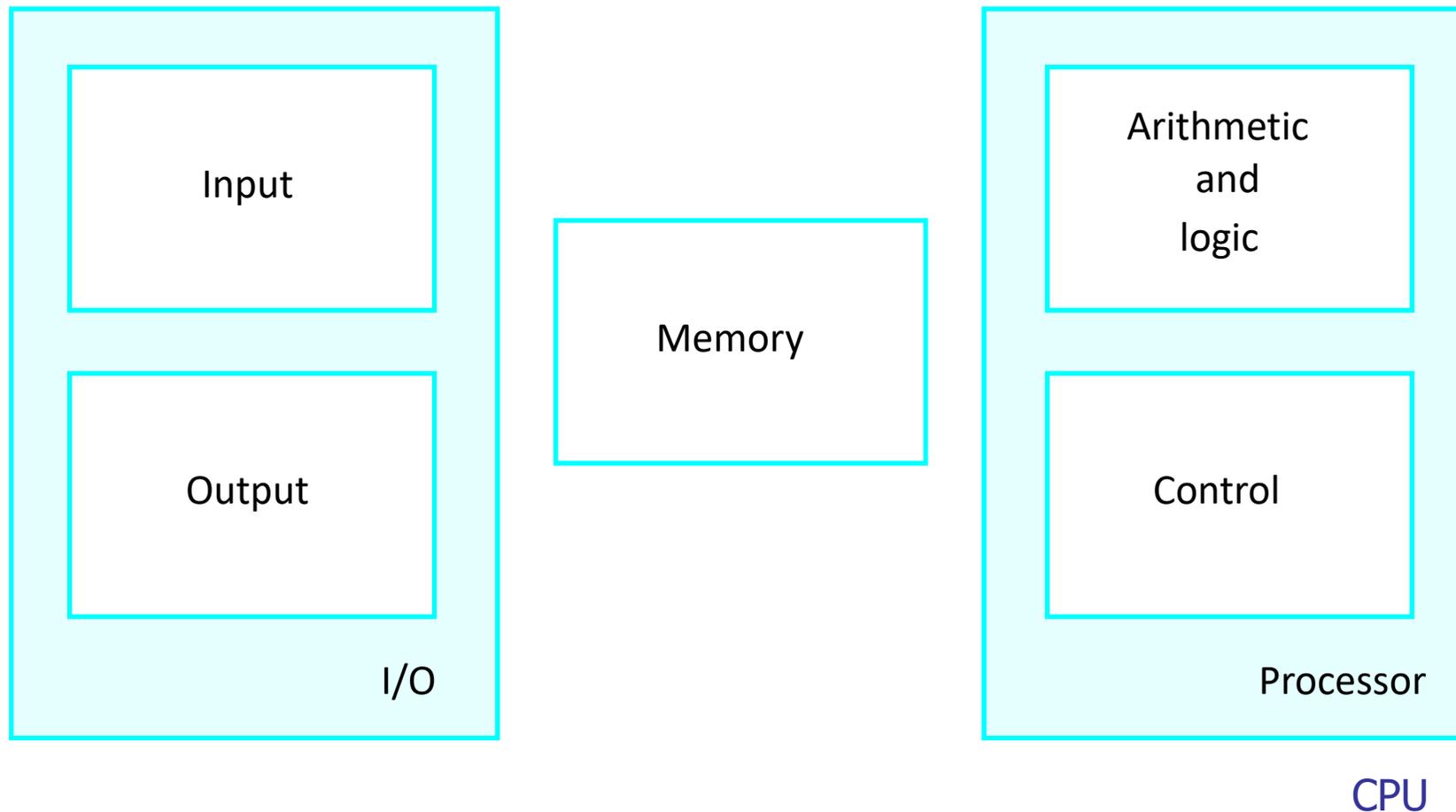




# Unità funzionali di un sistema a Microprocessore

## *Il processore*

# Unità funzionali



@@@ È un modello concettuale, i.e. non corrisponde alla reale implementazione fisica

# Unità di I/O

---

- Input: Typical devices – keyboard, touchpad, mouse, microphone, camera, communication lines, the Internet
- Output: Typical devices – text and graphics displays, printers

# Il processore

---

- È in grado di eseguire un set di azioni elaborative elementari (**istruzioni**) più o meno complesse
- Una istruzione ha una lunghezza tipicamente multipla della lunghezza di parola
- Una sequenza di istruzioni costituisce un **programma**
- Un programma, ed i dati su cui esso opera, deve essere in memoria principale per poter essere eseguito

# *Tipi di istruzione*

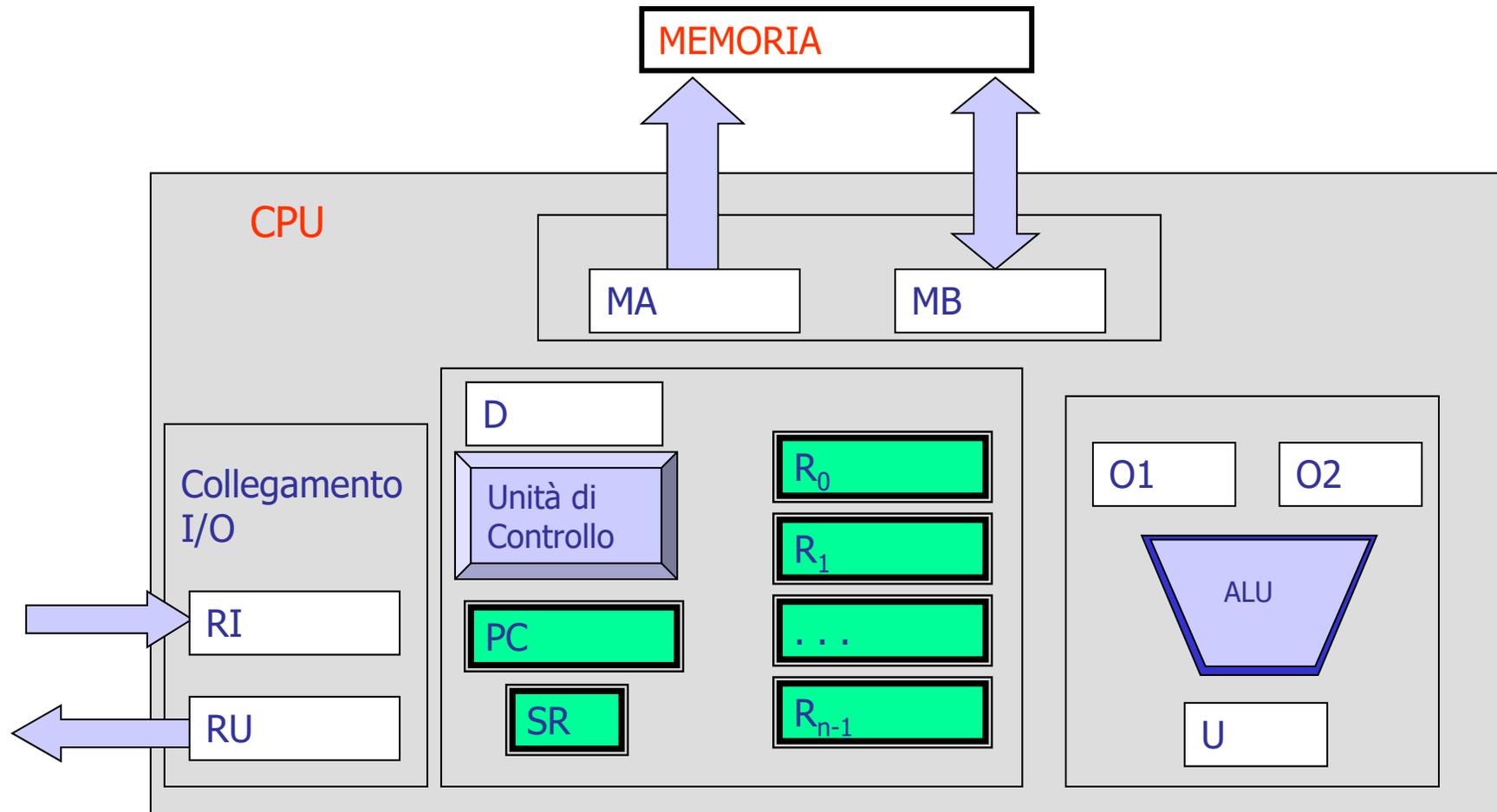
---

Tre tipi principali

- **Load** – sposta un dato dalla memoria (o un input) al processore
- **Store** – sposta un dato da un registro del processore alla memoria (o un dispositivo di output)
- **Operate** – esegue una operazione aritmetica o logica su dati presenti nei registri del processore

# Modello architetturale di un processore

## Modello a registri generali



# Componenti fondamentali del processore

---

- 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

# 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

# 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*

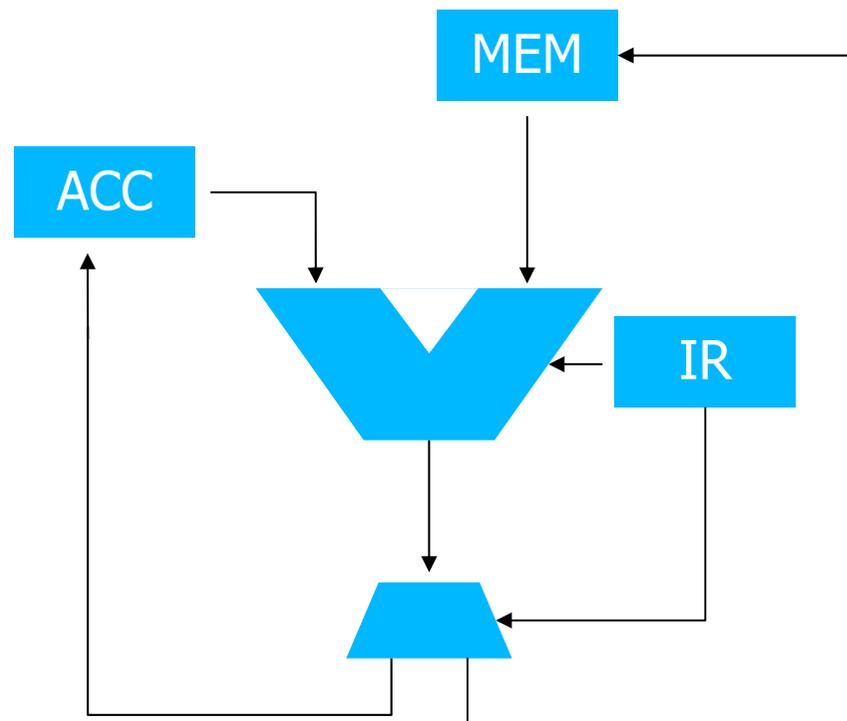
# Funzioni dei registri di macchina

---

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

# Tipologie di CPU alternative: ad accumulatore

- Un operando implicito è sempre l'accumulatore



**$C=A+B$**

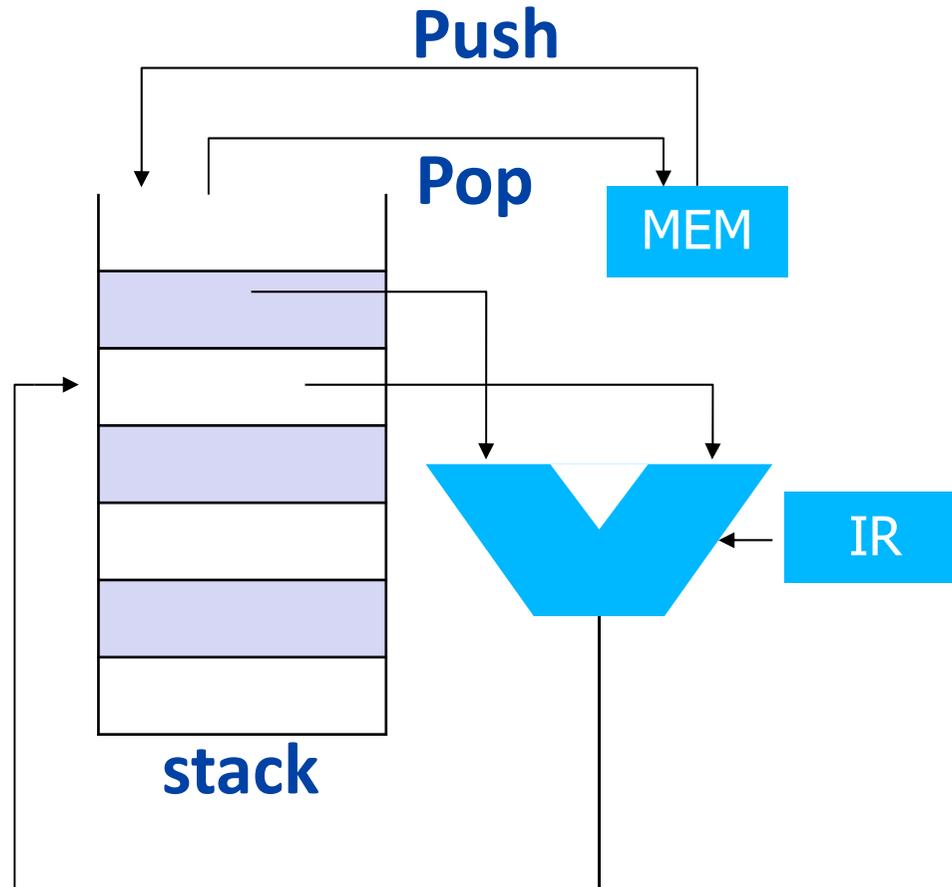
**Load A ; carica A in ACC**

**Add B ;  $ACC \leftarrow ACC+B$**

**Store C ;  $C \leftarrow ACC$**

# Tipologie di CPU alternative: a Stack

- Gli operandi sempre sullo stack



$C=A+B$

Push A ; [SP]  $\leftarrow$  A

Push B ; [SP]  $\leftarrow$  B

ADD ;

Pop C ; [SP]  $\Rightarrow$  C

[SP] = elemento sulla testa dello stack

# Confronto tra i modelli

---

<i>Stack</i>	<i>Accumulatore</i>	<i>Registro-memoria</i>	<i>Registro-registro</i>
Push A	Load A	Load R1, A	Load R1, A
Push B	Add B	Add R3, R1, B	Load R2, B
Add	Store C	Store R3, C	Add R3, R1, R2
Pop C			Store R3, C