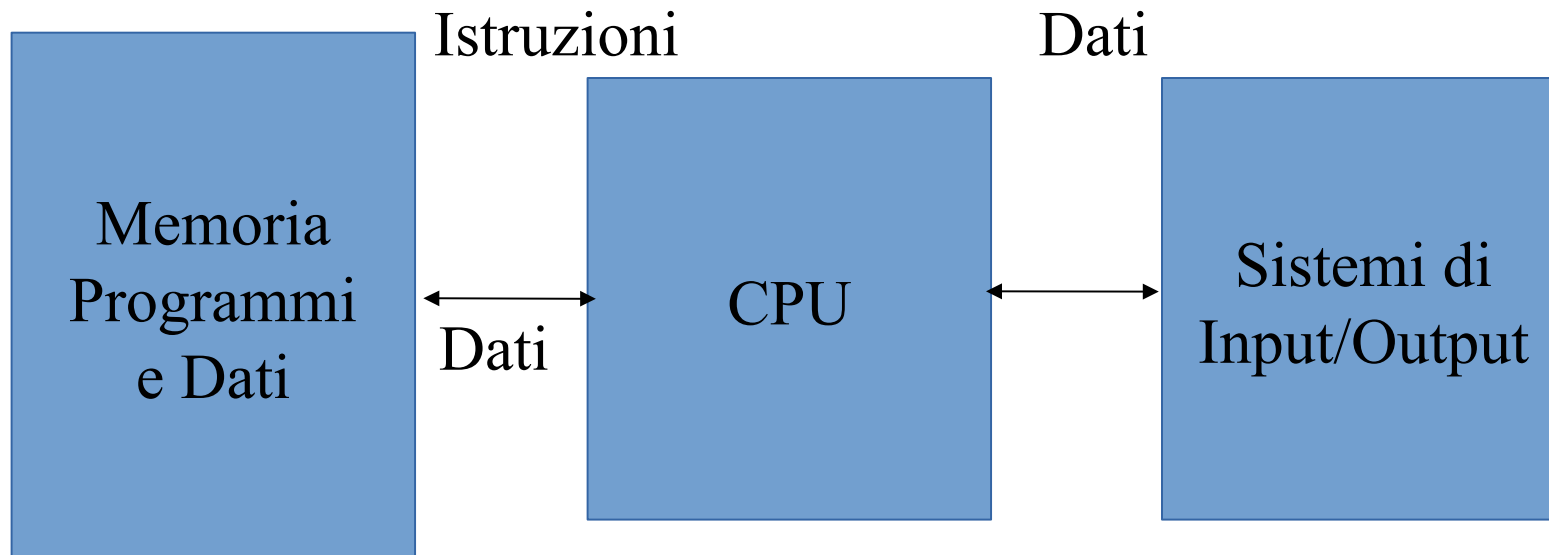


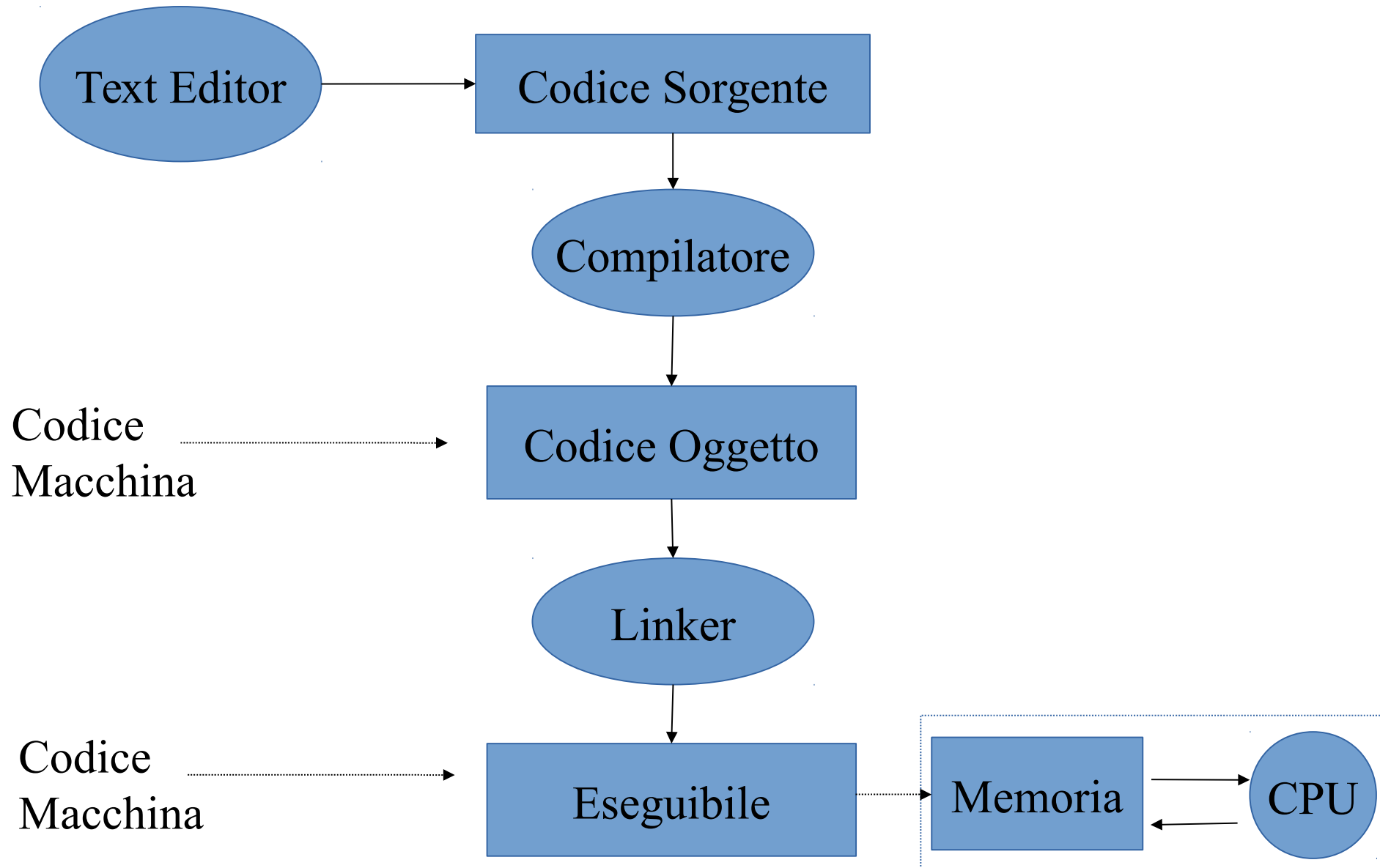
Il Processore

Modulo di Calcolatori Elettronici

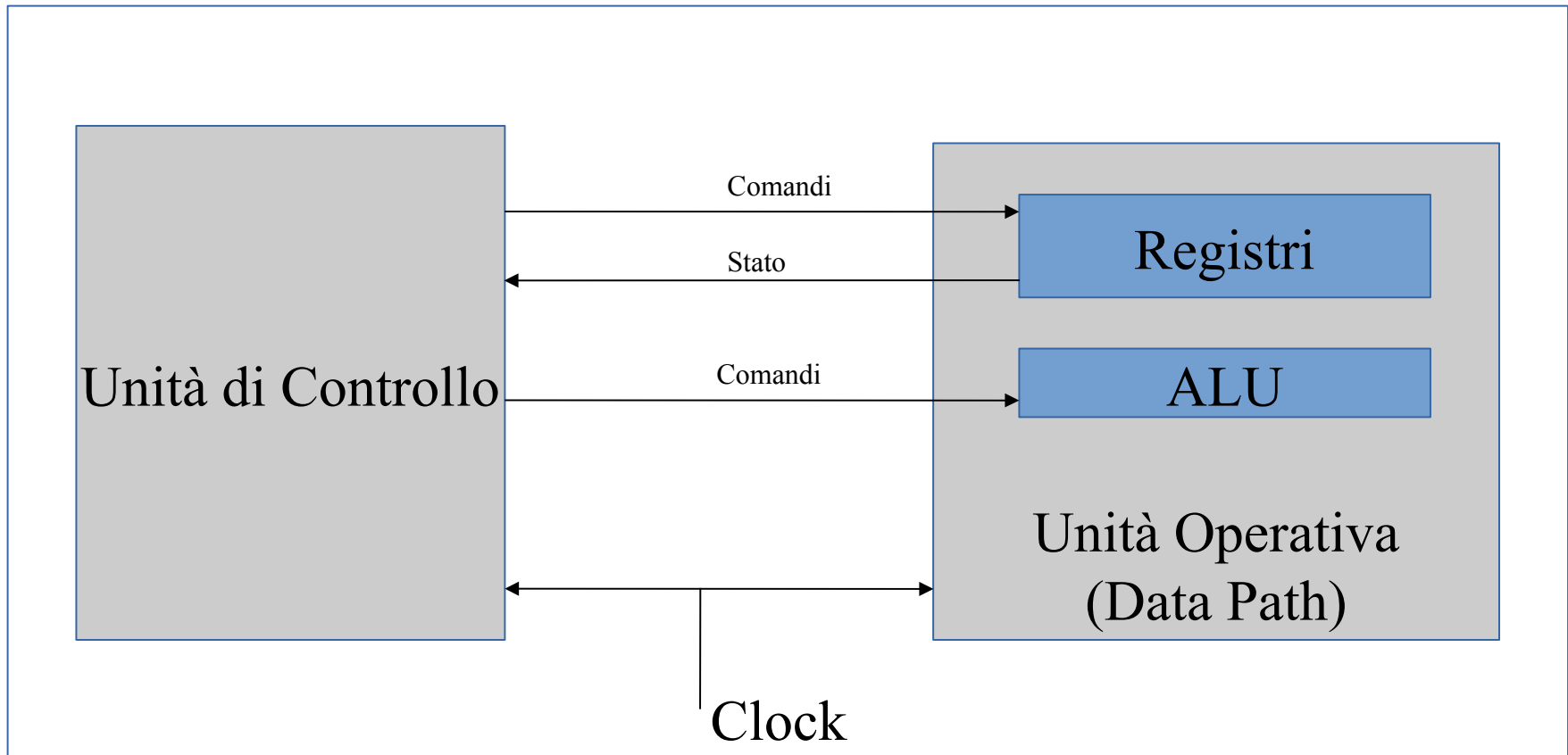
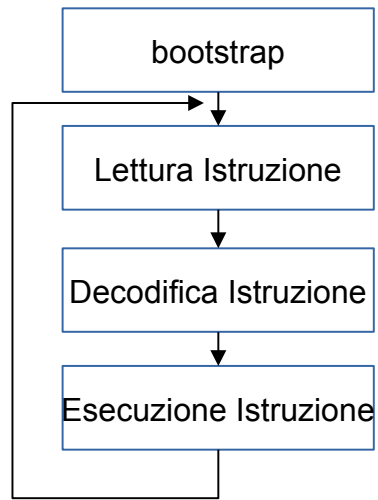
Modello Von Neuman



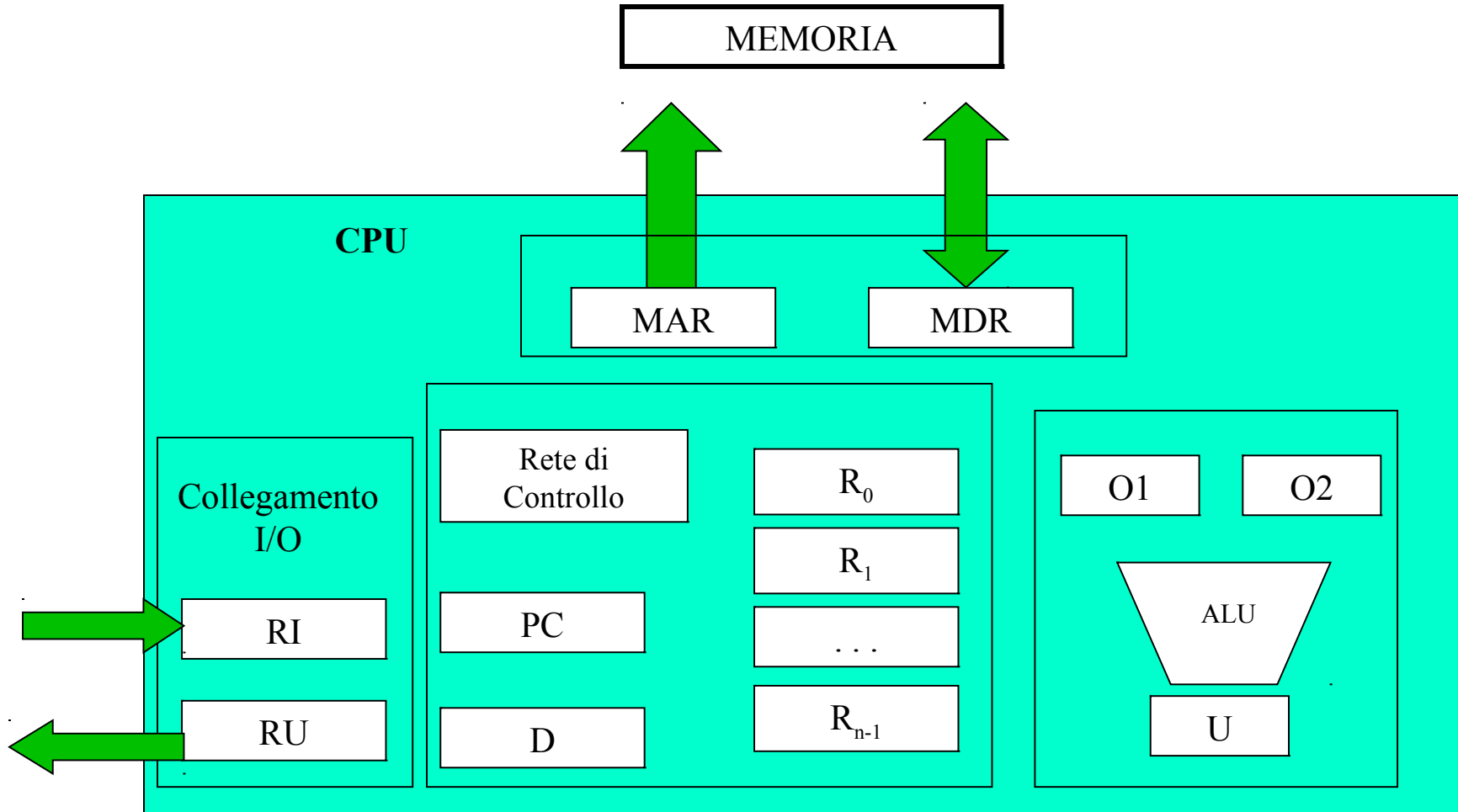
Caricamento in Memoria



CPU



Modello di riferimento di un processore



Elaborazione Ciclo di Von Neumann

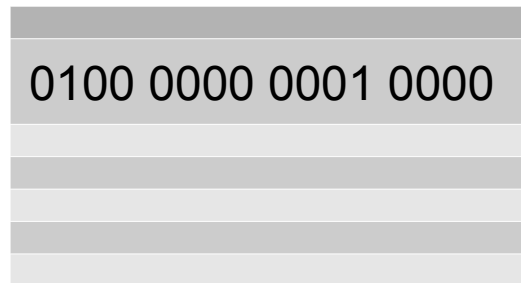
PC

00 000 1000

MAR

00 000 1000

00 000 1000



0100 0000 0001 0000

bootstrap

Lettura Istruzione

Decodifica Istruzione

Esecuzione Istruzione

MDR

0100 0000 0001 0000

IR

0100 0000 0001 0000

Modello di riferimento di un processore

➤ Componenti fondamentali del processore:

» Unità di controllo

- ◆ registro Program Counter (PC) o Prossima Istruzione
- ◆ Instruction Register o registro di decodifica (D)
- ◆ registri di Macchina

» Sezione di Collegamento con la memoria

- ◆ registro degli indirizzi di memoria (MA)
- ◆ registro dei dati della memoria (MB)

» Unità aritmetico/logica (ALU)

» Sezione di Collegamento con Ingresso-Uscita

➤ Il *linguaggio macchina* di un processore è costituito dalla codifica in binario delle istruzioni eseguibili direttamente dall'hardware

Registri del processore

- I registri del processore sono classificabili in due categorie:
 - » *registri interni*, non direttamente visibili al programmatore e necessari al funzionamento del processore
 - ◆ MAR, MDR, RI, RO, PC, O1, O2, U, D,...
 - » *registri di macchina* (definiscono il *modello di programmazione*) ulteriormente classificabili in:
 - registri generali
 - registri speciali

Tipi di dato del processore

- Il processore tratta diversi tipi di informazioni così come definito dal suo linguaggio macchina:
 - » Istruzioni;
 - » Indirizzi di memoria, da cui prelevare o immettere istruzioni
 - » Dati da elaborare (tipi aritmetici, logici,...)
- Tutti i registri di un processore sono caratterizzati da una propria lunghezza:
 - » MA e PC contengono dati di tipo indirizzo;
 - » I registri di interfaccia con l'ALU dati di tipo numerico (o altro tipo da elaborare)

Il modello di processore

- Il modello presentato rappresenta uno schema di riferimento
- Implementazioni reali si distinguono dal modello generale, allontanandosi dalla linearità concettuale del modello presentato:
 - » Una serie diversa o comunque più ricca di registri generali
 - » I registri di macchina possono essere delle locazioni di memoria
 - » La sezione di I/O può essere completamente assente ed essere implementata mediante chip appositi
 - » Nel caso di unità aritmetiche molto semplici è possibile prevedere l'aggiunta di appositi chip per l'estensione delle prestazioni della ALU

Parallelismo dei trasferimenti

- Un processore trasferisce i dati al suo interno con un parallelismo che è di norma coincidente con la lunghezza dei registri corrispondenti.
- Uno dei parametri caratteristici per classificare un processore è il numero k di bit di cui sono composti i registri dati (parallelismo dei trasferimenti interni)
- Processori a k bit: 8, 16, 32, 64,...

Parallelismo dei trasferimenti

➤ Si dice parola-macchina di un processore un suo dato o una sua istruzione:

» *Architetture RISC*

- *Istruzioni di uguale lunghezza*
- *Semplice UC (rete combinatoria)*
- *Ridotto set di istruzioni*
- *Istruzioni dedicate per lo scambio di memoria*

» *Architetture CISC:*

- *Istruzioni di lunghezza diversa*
- *UC complessa (microprogrammata)*
- *Set di istruzioni complesso*
- *Diversi modi di indirizzamento*

Funzione dei Registri

- **Memoria dell'istruzione corrente:** PC
- **Transito dei dati** (memoria centrale ↔ registri, registri ↔ registri): move, load, store.
- **Registri accumulatore:** Accumulo di risultati di operazioni aritmetiche o logiche (es: $R := (op) R$ oppure $R := R (op) X$)
- **Indirizzamento:** funzione necessaria ad individuare l'indirizzo di un operando
- **Stack e puntatori a stack:** filosofia LIFO, last-in-first-out
- **Indicatori o flag:** informazioni binarie che memorizzano un evento o una eccezione. Sono posizionate per effetto dell'esecuzione di una istruzione e, a loro volta, possono condizionare l'evoluzione di altre istruzioni (indicatore di dato nullo, dato negativo, dato positivo, di overflow, underflow, riporto, di errori di programmi in termini di riferimento ad indirizzi o a codici illeciti).

Funzione dei Registri

- **Stato del processore:** meccanismo di coordinamento tra programmi utente e S.O e supporto alla multiprogrammazione.
 - » *Stato supervisore:* opera il sistema operativo e tutte le istruzioni sono eseguite dal processore
 - » *Stato utente:* opera un programma di utente e le istruzioni privilegiate non sono eseguibili
- **Funzioni speciali per la gestione del sistema** (tipiche dei singoli processori) :
 - » Ausilio all'indirizzamento
 - » Trattamento delle interruzioni
 - » Conteggio per istruzioni “a blocchi”
 - » Registri “limiti di memoria”

Tipologia dei Registri

- **Nessun processore reale corrisponde completamente al modello a registri generali**
- **I registri di macchina sono organizzati in blocchi:**
 - » Organizzato come una memoria indirizzabile (blocco R, registro R_i)
 - » Organizzato come registri autonomi (ogni registro individuato da un nome)
- **Un medesimo registro può essere preposto all'espletamento di più funzioni:**
 - » Concentrazione di più funzioni nei registri vs. Specializzazione dei registri
 - ◆ Concentrando più funzioni in un registro si ha una maggiore complessità interna, e nel contempo una maggiore flessibilità d'uso
 - ◆ Specializzando i registri è necessario specializzare i codici operativi per la stessa operazione richiesta a banche distinte

Tipologia dei Registri

- **I registri indirizzabili possono essere indipendenti dalla memoria oppure costituire logicamente le sue prime locazioni:**
 - » Quando i registri appartengono allo spazio indirizzabile di memoria si può fare riferimento ad essi anche attraverso un indirizzo di memoria e sono definite tutte le operazioni definite su locazioni di memoria (prestazioni, gerarchia di memoria).
- **Registro PC:**
 - » Talora ritenuto registro interno, talora ritenuto registro di macchina
- **Registri accumulatori:**
 - » Un registro, un banco di registri, due registri
 - » Spesso specializzati per tipo di dato
- **Registri Indice:**
 - » Svolgono funzioni di indirizzamento direttamente controllabili dall'utente.
- **Registri ausiliari di indirizzamento:**
 - » In genere manovrati dal sistema operativo e svolgono funzioni di indirizzamento connesse alla gestione del sistema.

Tipologia dei Registri

➤ **Registri generali:**

- » Sono quei registri organizzati in banchi che svolgono due o più funzioni (ad es. accumulatore e indirizzamento)

➤ **Registri di transito**

➤ **Registri stack pointer:**

- » Per implementare una memoria a stack (rari)
- » Preposti alla gestione di stack implementati in memoria

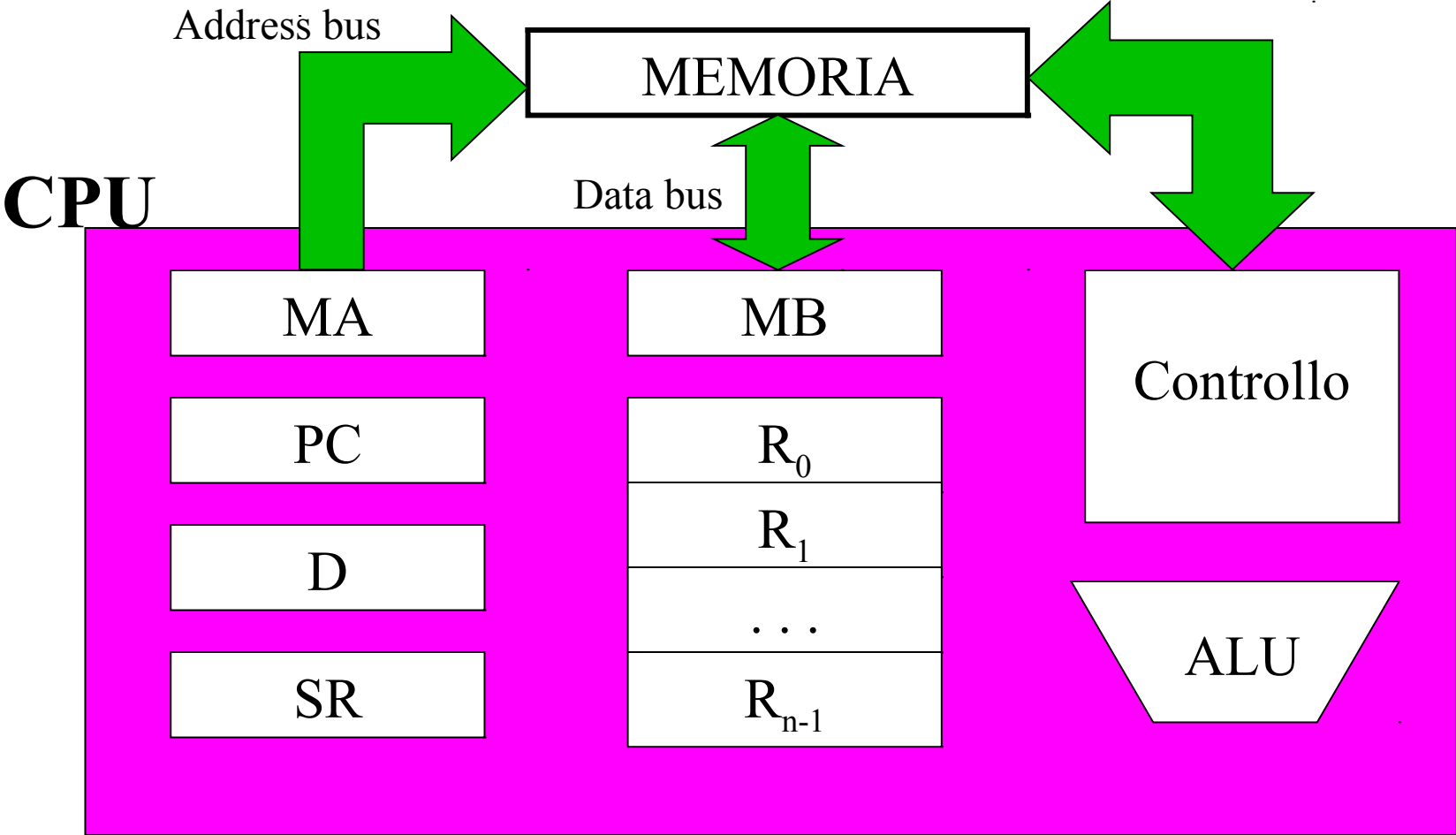
➤ **Flag e Codice di condizione (registro di stato):**

- » Accessibili direttamente da programma. Alcuni sono :
 - ◆ Z : bit zero
 - ◆ N : bit negativo
 - ◆ P : bit positivo
 - ◆ V : bit overflow
 - ◆ C : bit riporto
 - ◆ P : bit parità
- » Spesso alcuni o tutti sono accorpati in un unico registro detto CC (Codice Condizione) che quindi rappresenta lo stato del programma in esecuzione.

➤ **Stato del processore:**

- » Memorizza tipicamente lo stato del processore:
 - ◆ Program counter
 - ◆ Registro CC
 - ◆ Ed altre informazioni per la gestione del sistema

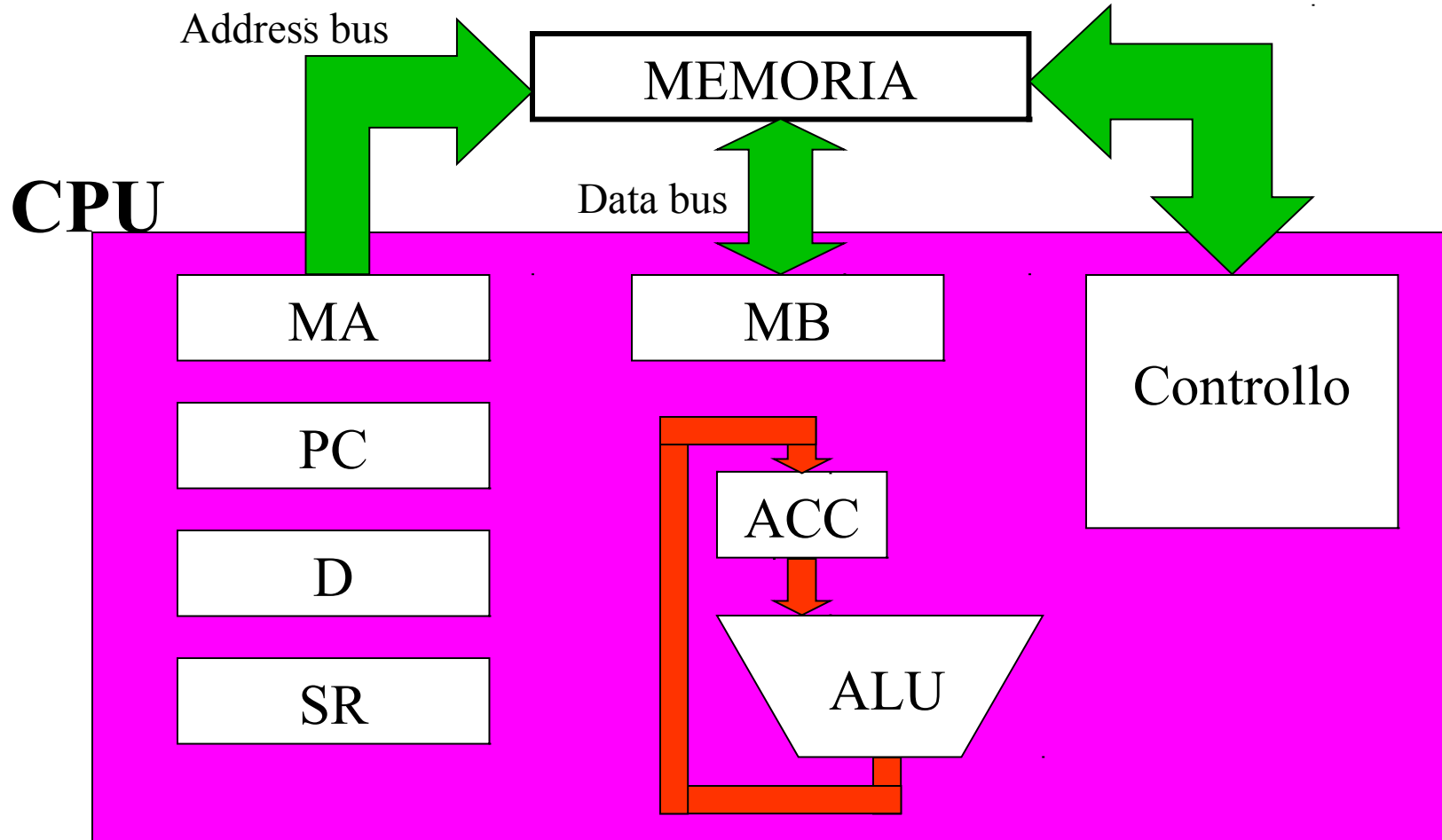
Architettura del processore: modello a registri generali (1)



Architettura del processore: modello a registri generali (2)

- In un processore a registri generali, il processore dispone di un set di registri $REG_0, REG_1, \dots, REG_{N-1}$ 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:
 $[REG_x] + [REG_y] \rightarrow [REG_x]$
- Istruzioni con operandi memoria-registri:
 $[REG_x] + [MEM_y] \rightarrow [REG_x]$ *memory-to-register*
 $[MEM_x] + [REG_y] \rightarrow [MEM_x]$ *register-to-memory*

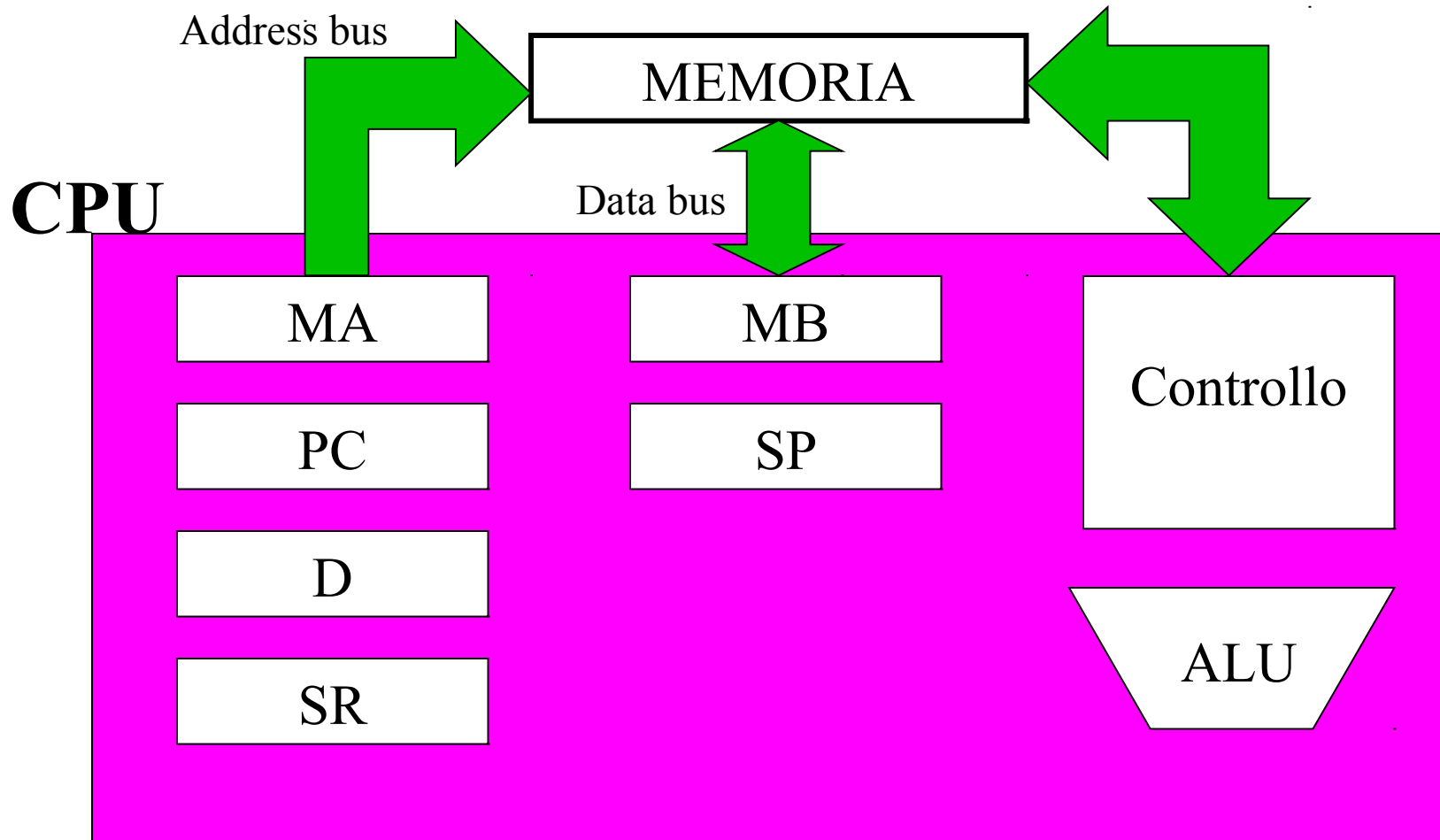
Architettura del processore: modello ad accumulatore (1)



Architettura del processore: modello ad accumulatore (2)

- In un processore ad accumulatore tutte le istruzioni aritmetiche, logiche e di confronto hanno un operando in memoria ed un altro (riferito implicitamente) contenuto in un registro interno del processore detto *accumulatore*
- Esempio: per realizzare $[x]+[y] \rightarrow z$ con una macchina ad accumulatore (es. Motorola 6809) occorre eseguire una sequenza di istruzioni del tipo
 - LDA x [x] → accumulatore
 - ADDA y [y]+[accumulatore] → accumulatore
 - STA z [accumulatore] → z
- Dimensione e velocità di esecuzione dei programmi penalizzate dal fatto che tutte le istruzioni devono indirizzare un dato in memoria

Architettura del processore: modello a stack (1)



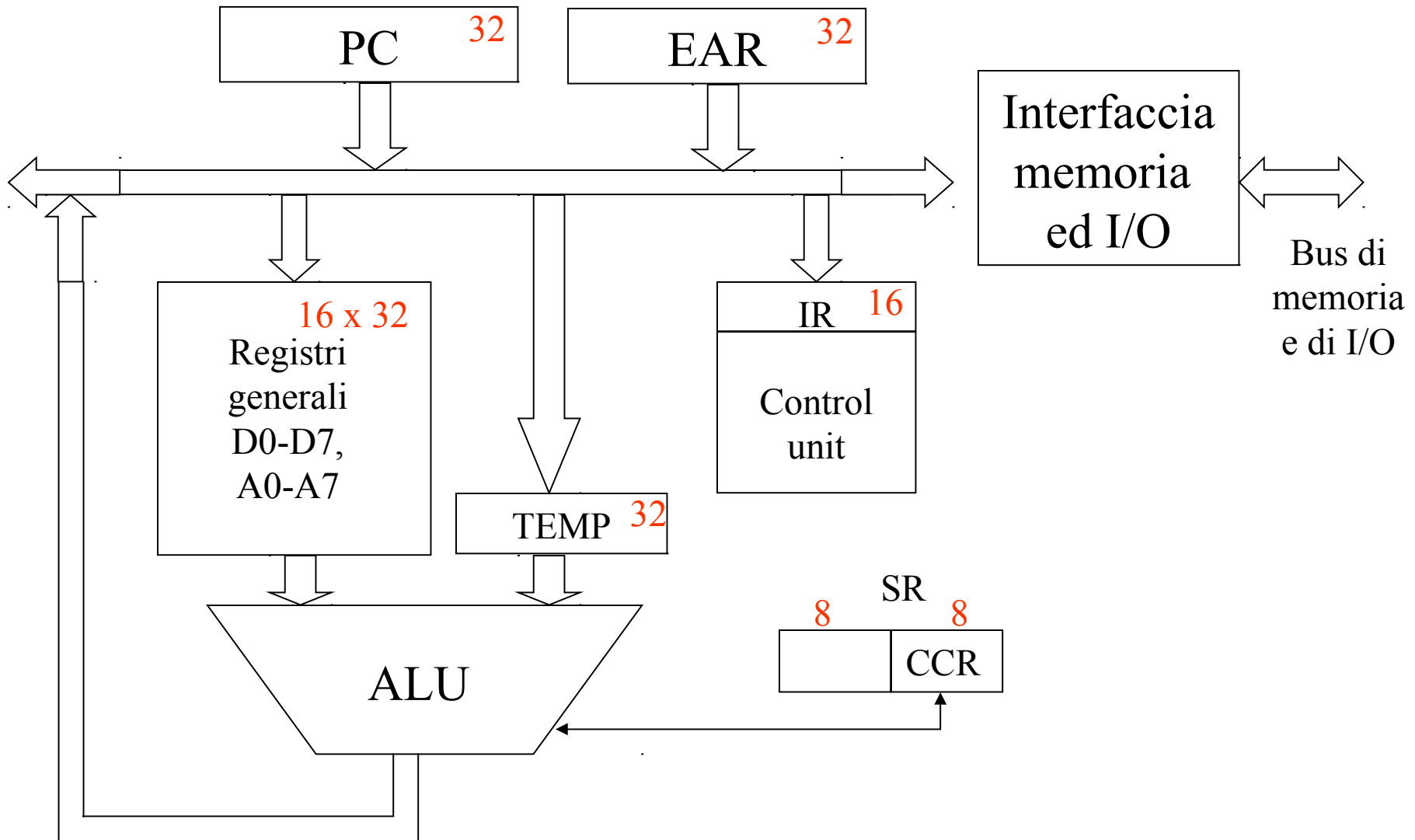
Architettura del processore: modello a stack (2)

- In un processore a stack, non esistono né l'accumulatore né i registri generali, ma solo un registro STACK POINTER (SP) che punta alla cima di una pila
- Le istruzioni (eccetto quelle di trasferimento dati) operano su dati posti in cima allo stack
- Lo stack può essere interno al processore o in memoria
- Istruzioni a zero operandi:
 $[SP] + [SP+1] \rightarrow SP$
- Istruzioni con 1 operando memoria:
 $[SP] \rightarrow [MEM_x]$
 $[MEM_x] \rightarrow SP$
- Più lenta di una macchina a registri generali (se stack in memoria centrale)

Architettura del processore: architetture ortogonali

- Unico banco di registri generali
- Soltanto il registro di stato al di fuori del banco
- Registri speciali, puntatori a stack, PC appartengono al banco di registri generali.

Architettura di un processore reale: MC 68000

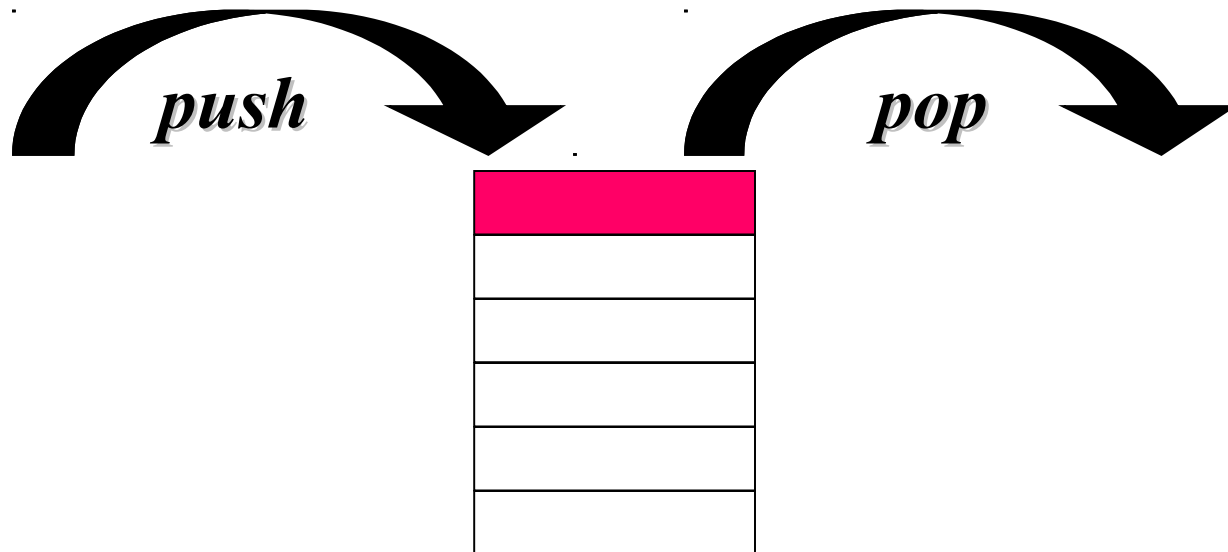


Lo stack

- E' una struttura dati astratta atta a contenere un insieme di valori di un determinato tipo e sulla quale sono definite le funzioni di accesso:
 - » Inserimento o scrittura (push) : memorizza nello stack un elemento che viene posizionato in ultima posizione; l'elemento che si trovava prima in ultima posizione diventa il penultimo, etc.
 - » Lettura o prelievo (pop) : preleva ed elimina dallo stack l'ultimo elemento
- Non serve specificare l'operando.

Lo stack

- Si utilizza laddove è necessario accantonare dei dati per svolgere un compito, al termine del quale i dati stessi devono essere recuperati (gestione di sottoprogrammi, interruzioni, etc).
- Gestione LIFO (Last-In-First-Out).



Lo stack

- Nella pratica si implementa attraverso l'uso di un puntatore in memoria (o in registri dedicati) che indica istante per istante la posizione dell'ultimo elemento dello stack
- Talvolta sono previsti dei controlli
 - » Stack overflow, all'atto del push
 - » Stack empty, all'atto del pop

