Università degli Studi di Torino
Scuola di Amministrazione Aziendale

Corso di laurea in
Management dell'informazione e della comunicazione aziendale

Corso di
Informatica generale (parte teorica)

Diego Magro

ARGOMENTI DI QUESTO GRUPPO DI LUCIDI

- **Architettura hardware dei calcolatori:**
  - macchina di von Neumann
  - memoria principale
  - CPU
  - bus
  - controller
  - memoria di massa
  - dispositivi di input/output
  - scheda madre
  - alcune evoluzioni della macchina di von Neumann (cenni)

ARCHITETTURA HARDWARE DEI CALCOLATORI

INTRODUZIONE

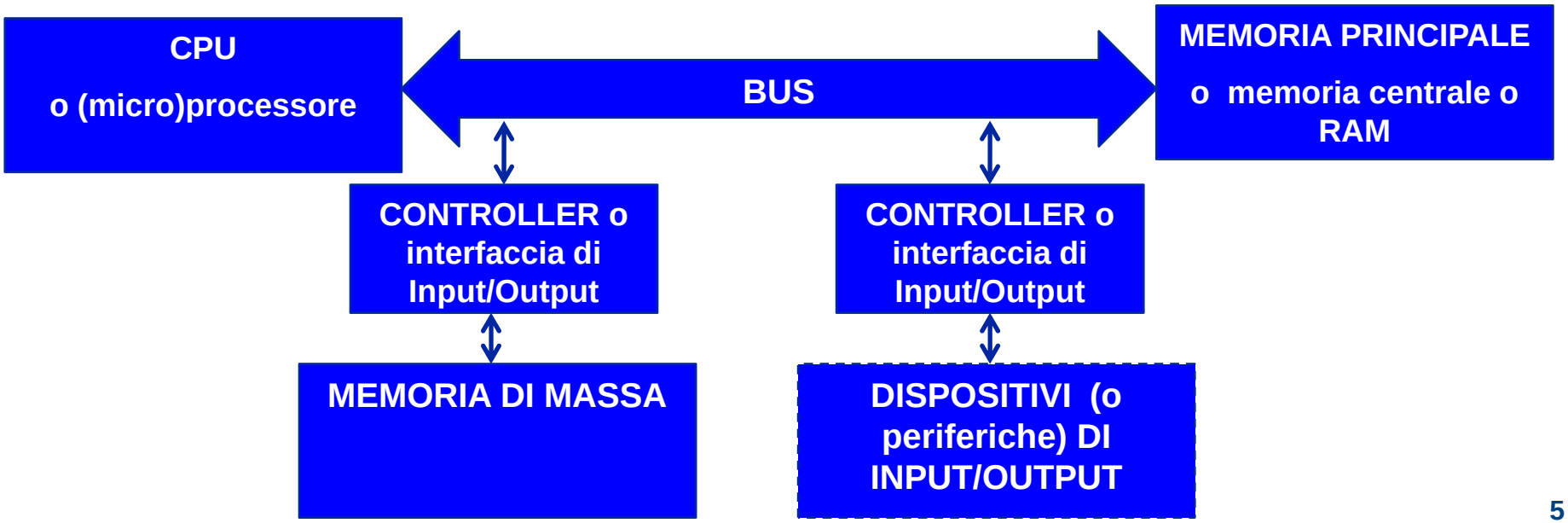
- **Dal punto di vista hardware, i moderni calcolatori sono organizzati secondo un modello chiamato "Macchina di von Neumann" (cui si è già accennato in precedenza)**

- **Descrivere l'architettura hardware dei calcolatori significa innanzi tutto descrivere le componenti della macchina di von Neumann e le interazioni fra queste componenti**

- **Descrivere la macchina di von Neumann ci consente di capire i fondamenti del funzionamento hardware dei calcolatori: occorre però tener presente che l'architettura dei moderni calcolatori, pur avendo sempre come base la macchina di von Neumann, presenta spesso delle differenze rispetto al modello base, introdotte prevalentemente per ragioni di efficienza: nonostante ciò, i fondamenti restano validi**

MACCHINA DI VON NEUMANN (1)

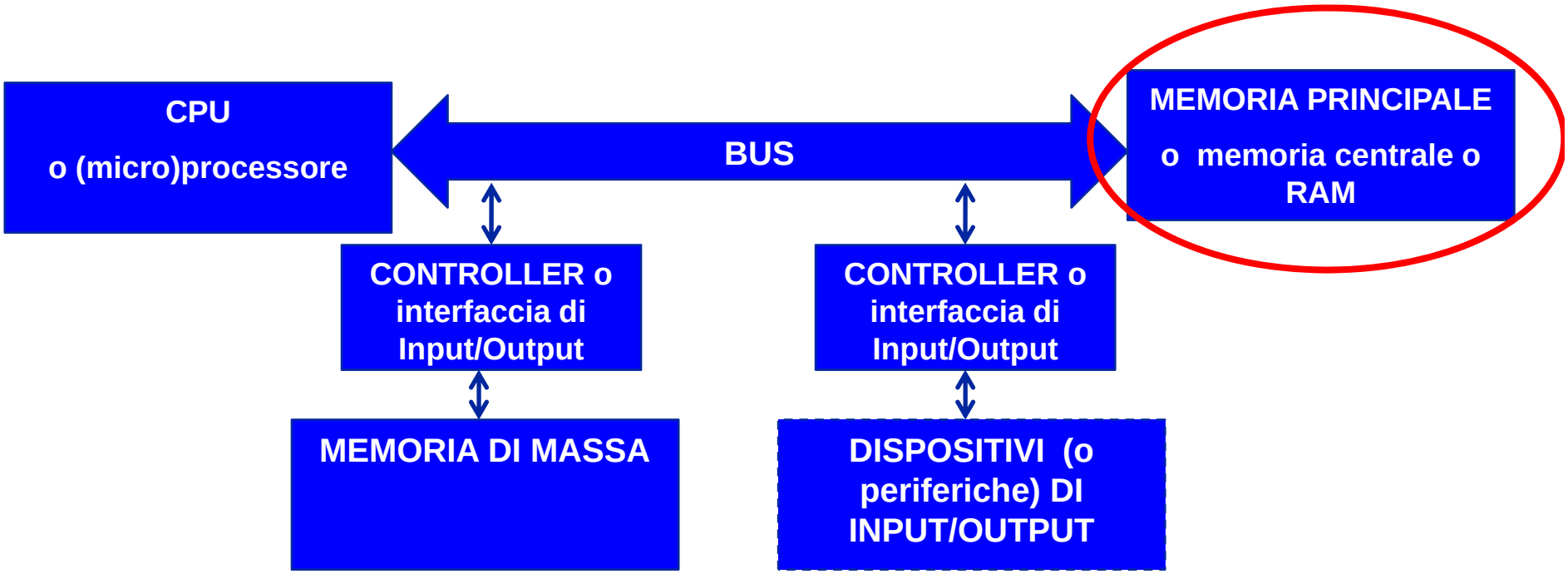
- **Componenti della macchina di von Neumann:**
  1. CPU (o processore, o microprocessore)
  2. memoria principale (o memoria centrale o RAM)
  3. periferiche (o periferiche di Input/Output o dispositivi di Input/Output) fra le quali, un ruolo particolare è giocato dai dispositivi per la memoria di massa (detta in alcuni casi memoria secondaria)
  4. controller (o interfacce di Input/Output)
  5. bus



MACCHINA DI VON NEUMANN (2)

- Come detto, un calcolatore esegue programmi (che usano, elaborano e/o producono dati). I programmi sono costituiti da istruzioni
- La CPU esegue istruzioni e coordina l'attività delle componenti hardware
- La memoria principale contiene (temporaneamente) le istruzioni che la CPU esegue e i dati che la CPU usa ed, eventualmente, dati che quest'ultima elabora o produce nell'esecuzione delle istruzioni
- Le periferiche di Input/Output consentono la comunicazione del calcolatore con l'ambiente esterno: immissione di informazione dall'ambiente esterno (periferiche di input), presentazione di informazione dal calcolatore verso l'ambiente esterno (periferiche di output)
- La memoria di massa memorizza informazioni in maniera persistente. Affinché la CPU possa accedere a queste informazioni, occorre che esse siano precedentemente copiate in memoria principale
- I controller sono le interfacce fra il calcolatore e le periferiche (mediano lo scambio di informazioni fra CPU e memoria principale da un lato e le periferiche dall'altro)
- Il bus collega fra loro le varie componenti ed è il mezzo attraverso il quale esse comunicano

MEMORIA PRINCIPALE (1)



MEMORIA PRINCIPALE (2)

- **Detta anche “memoria centrale” e “RAM (Random Access Memory)”**
- **Assieme alla CPU è una delle due componenti cardine di un calcolatore**
- **Permette di memorizzare le informazioni (dati e istruzioni) in modo che esse siano direttamente disponibili per la CPU**
- **Da un punto di vista fisico, è una sequenza di componenti elettroniche ciascuna delle quali può assumere uno di due stati possibili. Opportuni circuiti elettronici possono far assumere a tali componenti, di volta in volta, uno o l’altro di questi stati e, sempre per mezzo di circuiti elettronici, è possibile riconoscere e distinguere i due stati possibili.**
- **Se ad uno dei due stati associamo lo “0” e all’altro l’ “1”, possiamo dire che, da un punto di vista logico, ciascuna delle suddette componenti della memoria principale è un bit (nel senso che è una realizzazione fisica, per mezzo di dispositivi elettronici, del bit), pertanto:**
- **la memoria principale è una sequenza di bit**

MEMORIA PRINCIPALE (3)

- **Siccome ogni informazione può essere codificata in forma digitale come sequenza di valori di bit (come abbiamo già detto), deduciamo che la memoria principale può contenere qualunque tipo di informazione**
- **Da un punto di vista logico, essa è ulteriormente strutturata: i bit sono aggregati a gruppi, tipicamente di 8, 16, 32 o 64 elementi.**
- **Ogni gruppo di 8, 16, 32, o 64 bit è detto cella o locazione di memoria e può memorizzare il valore di 1 byte, 2 byte, 4 byte, 8 byte, rispettivamente**
- **➔ la memoria principale è pensabile come una sequenza di celle (o locazioni)**

MEMORIA PRINCIPALE (4)

- **Le operazioni che possono essere eseguite sulla memoria principale sono due:**
  1. Lettura di qualche informazione in essa contenuta
  2. Scrittura in essa di qualche informazione

- **La cella è l'unità minima di accesso alla memoria (nel senso che ogni operazione di scrittura scrive sempre almeno una cella e ogni operazione di lettura legge sempre almeno una cella)**

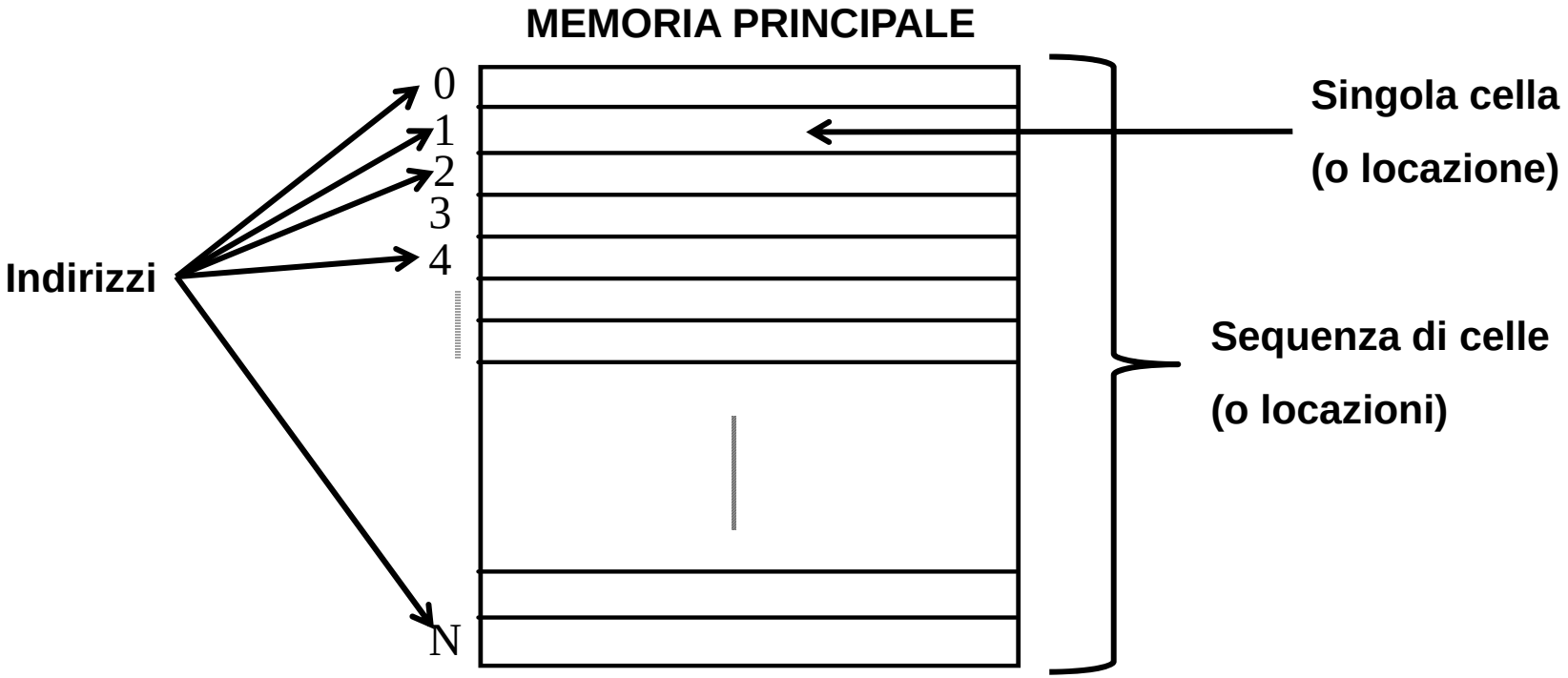
MEMORIA PRINCIPALE (5)

- **La memoria principale è volatile, cioè essa mantiene le informazioni in essa immagazzinate solo finché è alimentata elettricamente. L'interruzione dell'alimentazione elettrica comporta la perdita di ogni informazione in memoria principale.**

- **La memoria principale è "piuttosto veloce":**
  - essa consente un accesso diretto alle singole celle (cioè, ogni operazione di lettura o di scrittura di una certa cella è fatta accedendo direttamente a quest'ultima e non è necessario accedere anche alle celle che la precedono nella sequenza)
  - la tecnologia su cui è basata consente tempi di risposta piuttosto rapidi (cioè, le operazioni di lettura e di scrittura sono veloci, anche rapportati alla grande velocità con cui opera la CPU) e indipendenti dalle particolari celle lette o scritte e dalla loro posizione nella sequenza di celle che costituisce la memoria principale

MEMORIA PRINCIPALE (6)

- La posizione di ciascuna cella nella sequenza è detta **indirizzo della cella** → *Concetto semplice, ma importante!*
- L'indirizzo di una cella consente di identificare la cella ed è espresso con un numero intero non negativo (si comincia da zero!)



MEMORIA PRINCIPALE (7)

- **Essendo numeri interi non negativi (quindi non richiedendo la rappresentazione esplicita del segno), gli indirizzi delle celle sono rappresentati nel calcolatore direttamente nella loro notazione binaria (come numeri interi senza segno, appunto)**

- **Il processore può eseguire direttamente operazioni di lettura o di scrittura in memoria principale**

- **L'unico modo che il processore ha per indicare una cella (cosa che deve fare per ogni operazione di lettura o di scrittura che esso esegue) è attraverso il suo indirizzo**

MEMORIA PRINCIPALE (8)

- **In ogni calcolatore, il numero di bit disponibili per la specifica degli indirizzi di memoria principale è fisso[1] ed esso determina, quindi, il massimo numero di locazioni di memoria di cui il calcolatore può disporre**

- **Ad esempio, avendo a disposizione 32 bit per la specifica degli indirizzi, è possibile rappresentare fino a 2^{32} indirizzi distinti, ma non di più. Pertanto, un tale calcolatore può disporre al più di 2^{32} celle di memoria. Se ogni cella di memoria è di un singolo byte, questo significa che la massima quantità di memoria possibile in un tale calcolatore è 2^{32} byte (4 GB).**

- **Con 64 bit a disposizione, si riesce a specificare un numero molto più grande di indirizzi (2^{64}). Pertanto, un tale calcolatore può disporre al più di 2^{64} celle di memoria. Se ogni cella di memoria è di un singolo byte, questo significa che la massima quantità di memoria possibile in un tale calcolatore è 2^{64} byte (circa 16.000 PB)**

- **Gli indirizzi a 32 o 64 bit sono quelli effettivamente disponibili in molti calcolatori**

1 Come vedremo, esso è determinato dalla dimensione del bus indirizzi

MEMORIA PRINCIPALE (9)

- **La capacità è una caratteristica importante della memoria principale**
- **Con la progressiva diminuzione del costo della tecnologia, la quantità di memoria disponibile nei calcolatori è aumentata nel corso degli anni ed è tuttora in aumento**
- **Per fare un esempio, per gli attuali PC, è comune avere memorie principali che vanno da 2 a 4 GB[1]**
- **Entro un certo limite, è possibile espandere la memoria di un calcolatore, aggiungendovi nuovi elementi (schede) di memoria**

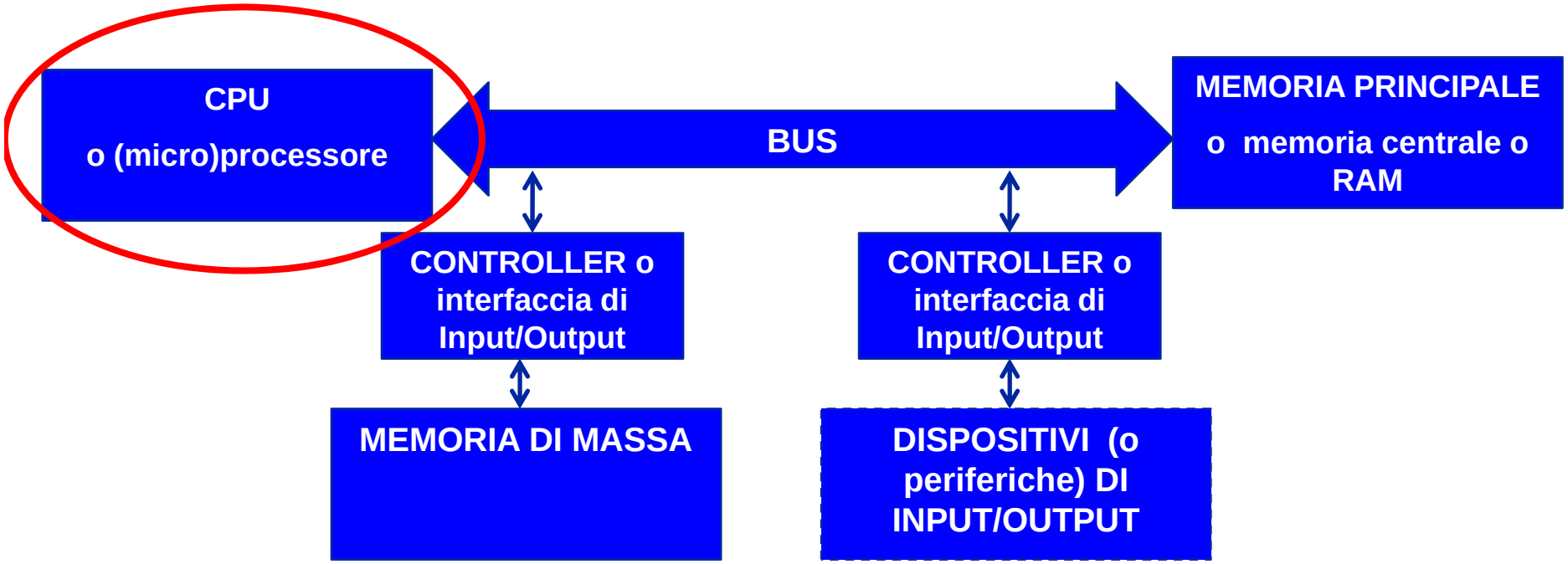
1 Dato soggetto a rapida obsolescenza

MEMORIA PRINCIPALE (10)

- **Moduli di memoria RAM**

Immagine tratta da Wikimedia Commons

CPU (1)


CPU
o (micro)processore
BUS
MEMORIA PRINCIPALE
o memoria centrale o RAM
CONTROLLER o interfaccia di Input/Output
CONTROLLER o interfaccia di Input/Output
MEMORIA DI MASSA
DISPOSITIVI (o periferiche) DI INPUT/OUTPUT

CPU (2)

- **CPU = Central Processing Unit (Unità centrale di elaborazione), detta anche “processore” o “microprocessore”**
- **Assieme alla memoria principale, è una delle due componenti cardine di un calcolatore**
- **Esegue le istruzioni che compongono i programmi e coordina le attività delle varie componenti hardware**

CPU (3)
[Linguaggio macchina]

- **L'insieme delle istruzioni che una CPU è in grado di eseguire direttamente è detto linguaggio macchina**

- **Il linguaggio macchina può variare da una CPU ad un'altra, ma in ogni particolare CPU esso è prefissato (cioè è una caratteristica immutabile della CPU stessa)**

CPU (4)
[Linguaggio macchina]

- **Ogni istruzione di un qualunque linguaggio macchina specifica un compito “piuttoso semplice” (come sommare due numeri e memorizzare il risultato in un certo posto, oppure trasferire “all’interno del processore” il contenuto di una cella di memoria, ecc.)**

- **...non esistono, nel linguaggio macchina, istruzioni che specificano “compiti complessi”, come ad esempio risolvere un’equazione di secondo grado (e nemmeno di primo grado!)**

- **Un “compito complesso”, per poter essere eseguito da una CPU, deve essere descritto nei termini di una sequenza (magari anche “molto lunga”...) di istruzioni “semplici” (appartenenti al linguaggio macchina della CPU)**

CPU (5)
[Linguaggio macchina]

- **Vi sono 3 tipologie di istruzioni macchina:**
  1. Istruzioni aritmetico/logiche
     - Istruzioni aritmetiche: ad esempio, somma si due numeri e salvataggio del risultato in un registro del processore[1]
     - Istruzioni logiche: ad esempio, calcolo di semplici operazioni fra sequenze di valori binari (AND, OR, XOR,...[2]) e salvataggio del risultato in un registro del processore

  2. Istruzioni di trasferimento dati: ad esempio copia del contenuto di una locazione di memoria principale in un registro del processore o, viceversa, copia del contenuto di un registro del processore in una locazione di memoria principale

  3. Istruzioni di controllo: le istruzioni vengono eseguite da processore una dopo l'altra, secondo la sequenza in cui esse appaiono nel programma, a meno che il processore non incontri una "istruzione di controllo" che gli imponga (eventualmente in presenza di certe condizioni) di "saltare" un gruppo di istruzioni e di riprendere l'esecuzione dall'istruzione che si trova in una certa locazione di memoria. Ad esempio, possono esservi istruzioni del tipo "se il contenuto di un certo registro è diverso da zero, allora salta ad una certa locazione"

1 Come vedremo, il processore dispone di una propria memoria interna, distinta dalla memoria principale e costituita da una sequenza di locazioni chiamate "registri"

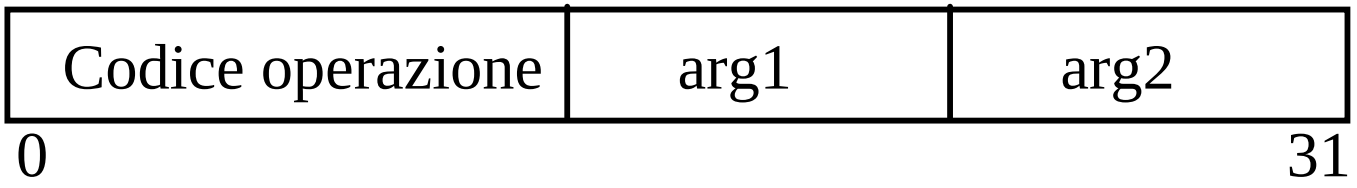
2 Ai fini di questo corso, non importa sapere in cosa consistano esattamente queste operazioni, ma solo che le "istruzioni logiche", nonostante quello che il nome potrebbe far pensare, rappresentano anch'esse semplici operazioni

CPU (6)
[Linguaggio macchina]

- **Come ogni altra informazione all'interno di un calcolatore, anche le istruzioni macchina sono codificate come sequenze di valori di bit**

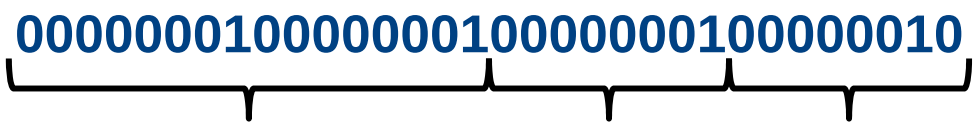
- **Il formato delle istruzioni varia da un linguaggio macchina ad un altro e, all'interno di uno stesso linguaggio macchina, può variare da un'istruzione all'altra**

- **Il seguente è un esempio di possibile formato e rappresenta un'istruzione codificata in 32 bit, in cui una prima sequenza di bit denota l'operazione (ad esempio "somma"), una seconda sequenza di bit denota sia il luogo in cui si trova il primo operando che la destinazione del risultato (ad esempio un registro del processore) e una terza sequenza di bit denota il luogo in cui si trova il secondo operando (ad esempio un registro del processore) :**



CPU (7)
[Linguaggio macchina]

- **Esempio (inventato!) di una particolare istruzione macchina avente il suddetto formato:**



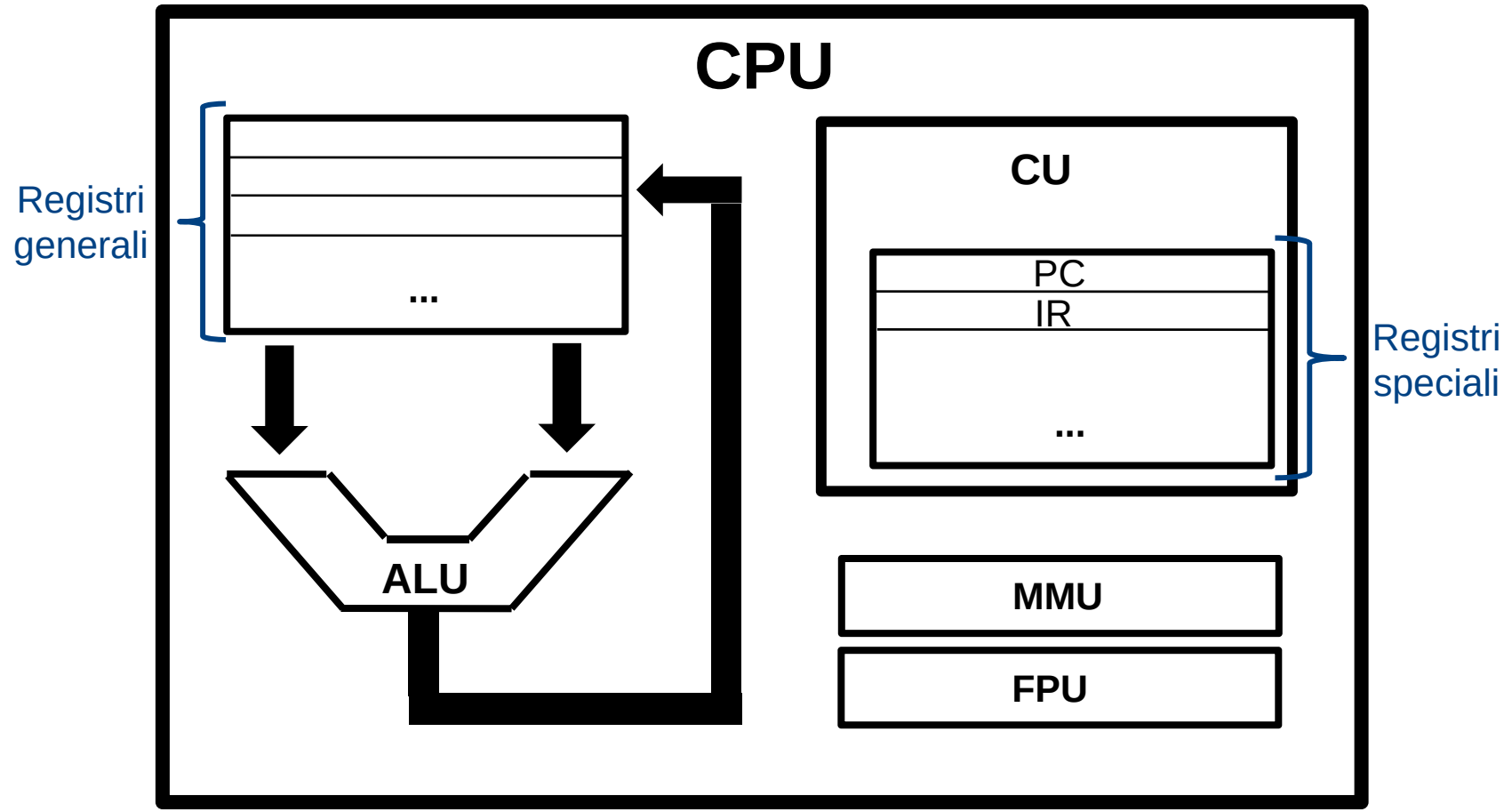
somma **registro 1** **registro 2**

= “somma il contenuto del registro 1 al contenuto del registro 2 e salva il risultato nel registro 1”

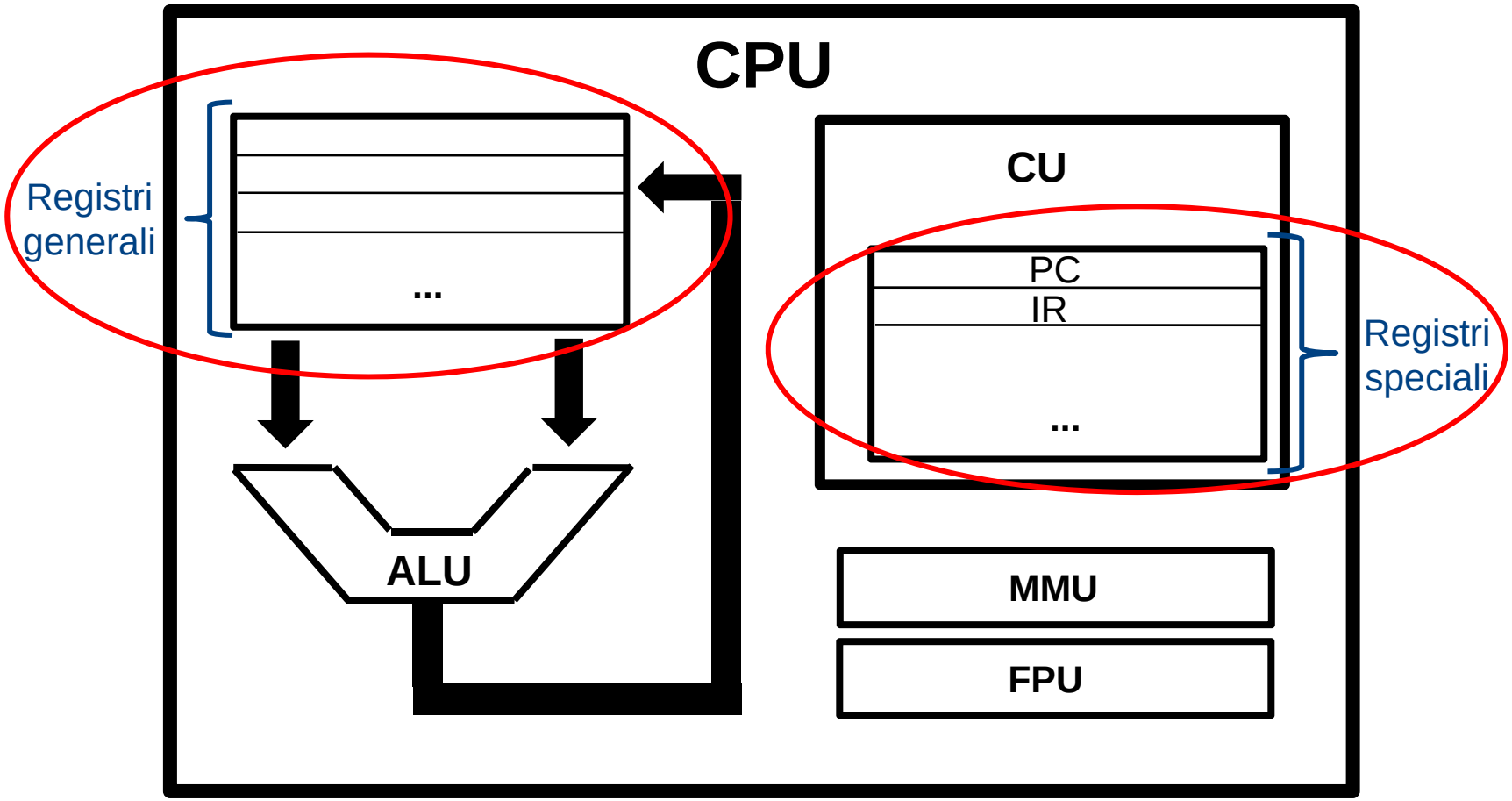
...ovviamente, la corrispondenza tra sequenze di bit e operazioni è decisa da chi progetta e crea un particolare linguaggio macchina...in questo caso, ho deciso, del tutto arbitrariamente, che “00000001000000001” significa “somma”...

CPU (8)
[Anatomia (semplificata) del processore]

- **Anatomia del processore (principali componenti di una CPU):**



CPU (9)
[Anatomia (semplificata) del processore: i registri]



CPU (10)
[Anatomia (semplificata) del processore: i registri]

- **All'interno del processore vi è una memoria, distinta dalla memoria principale (quest'ultima NON è parte del processore), di dimensioni ridotte (molto più piccola della memoria principale) e molto veloce (molto più veloce della memoria principale), costituita da una sequenza di locazioni chiamate "registri"**

- **Concettualmente, la struttura di un registro è del tutto analoga a quella di una locazione della memoria principale: ogni registro è costituito da una sequenza di dispositivi elettronici, ciascuno dei quali realizza un bit**

- **La differenza principale tra un registro e una locazione di memoria risiede nella velocità: l'accesso ad un registro da parte di una componente del processore è molto più veloce dell'accesso ad una locazione di memoria da parte del processore**

CPU (11)
[Anatomia (semplificata) del processore: i registri]

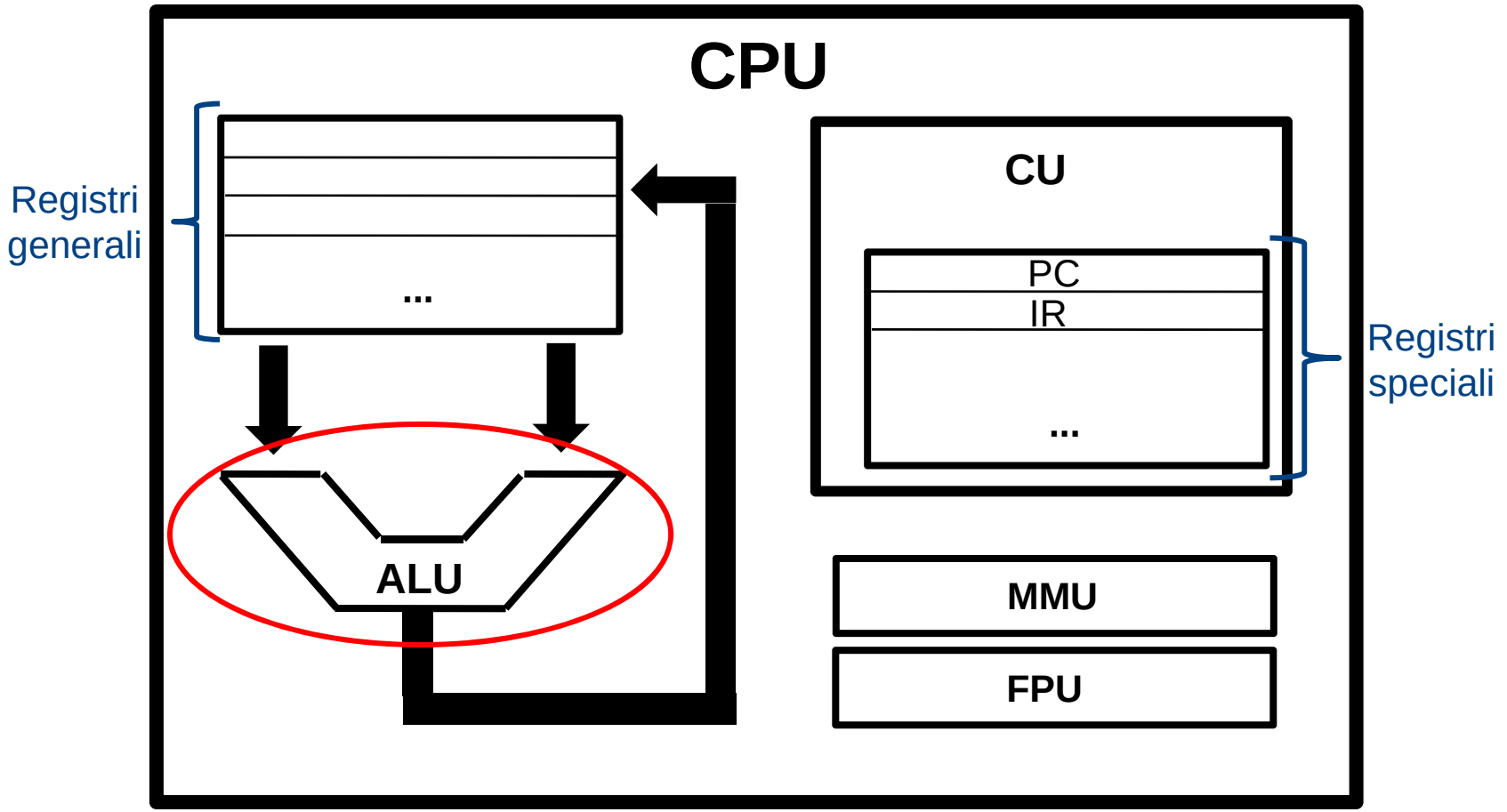
- **I registri sono utilizzati per memorizzare temporaneamente istruzioni e dati usati o prodotti dal processore nel corso dell'elaborazione**

- **Vi sono due categorie di registri:**
  1. Registri generali: sono usati per memorizzare temporaneamente dati usati o prodotti dalla CPU nel corso dell'elaborazione

  2. Registri speciali: sono usati dalla CU (Control Unit o Unità di controllo) per scopi particolari. Fra questi, ricordiamo i due principali:
     a) IR (Instruction Register o Registro istruzioni): in esso viene memorizzata l'istruzione macchina corrente (cioè quella che il processore sta eseguendo)
     b) PC (Program Counter): in esso viene memorizzato l'indirizzo della locazione di memoria principale[1] che contiene l'istruzione macchina che dovrà essere eseguita al termine dell'esecuzione di quella corrente (in alcune architetture, contiene l'indirizzo dell'istruzione corrente, ma il concetto non cambia)

1 Nei sistemi con memoria virtuale, l'indirizzo contenuto è quello virtuale, come vedremo in seguito

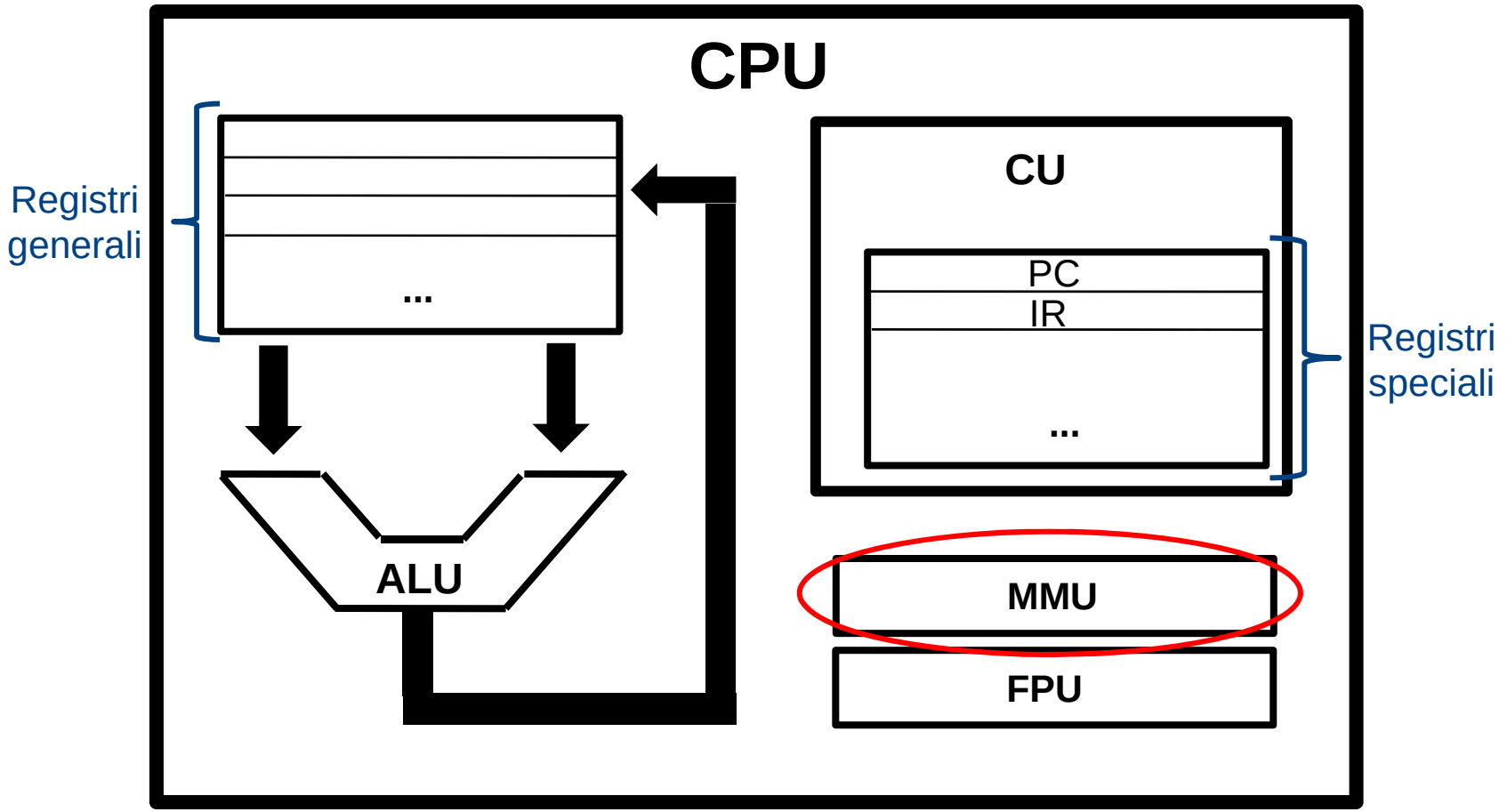
CPU (12)
[Anatomia (semplificata) del processore: la ALU]



CPU (13)
[Anatomia (semplificata) del processore: la ALU]

- **ALU = Arithmetic Logic Unit (Unità logico-aritmetica)**
- **E' costituita da un insieme di circuiti elettronici in grado di svolgere operazioni aritmetiche (es. somma di due numeri) e logiche (es. AND di due sequenze di valori binari)**
- **Legge gli operandi nei registri generali e memorizza il risultato in un registro generale**

CPU (14)
[Anatomia (semplificata) del processore: la MMU]



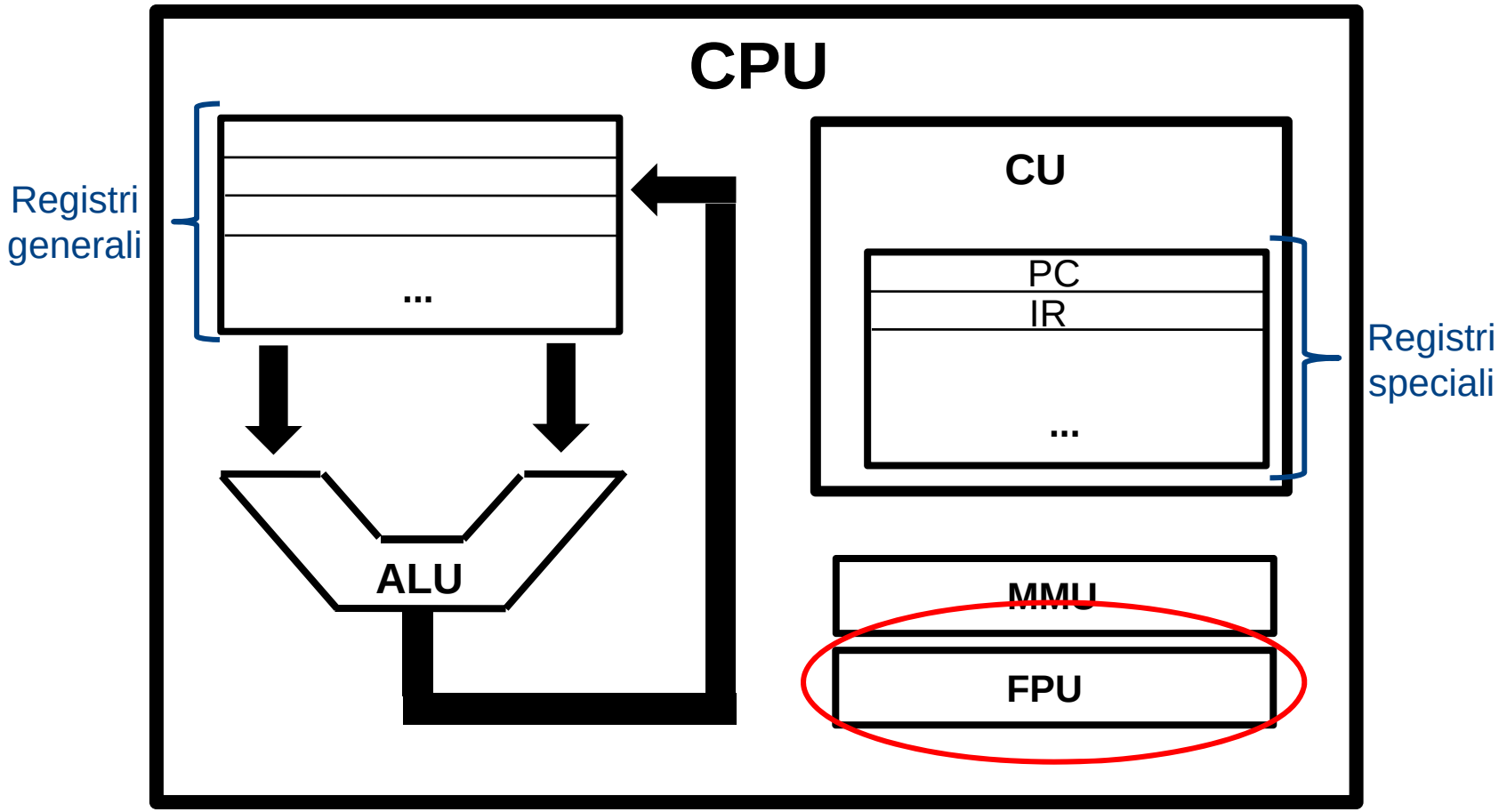
CPU (15)
[Anatomia (semplificata) del processore: la MMU]

- **MMU = Memory Management Unit (Unità di gestione della memoria)**

- **E' costituita da un insieme di circuiti elettronici che gestiscono (molti aspetti del)l'interazione del processore con la memoria principale. In particolare, la MMU si occupa anche dell'indirizzamento della memoria, supportando meccanismi di protezione (ad esempio, controlla che le istruzioni di un programma utente non richiedano l'accesso a locazioni di memoria principale proibite a tale programma), traducendo gli *indirizzi virtuali* in *indirizzi fisici*[1] (compito che, come vedremo, è essenziale nella realizzazione della cosiddetta "memoria virtuale"), ecc.**

1 Approfondiremo in seguito questi concetti

CPU (16)
[Anatomia (semplificata) del processore: la FPU]

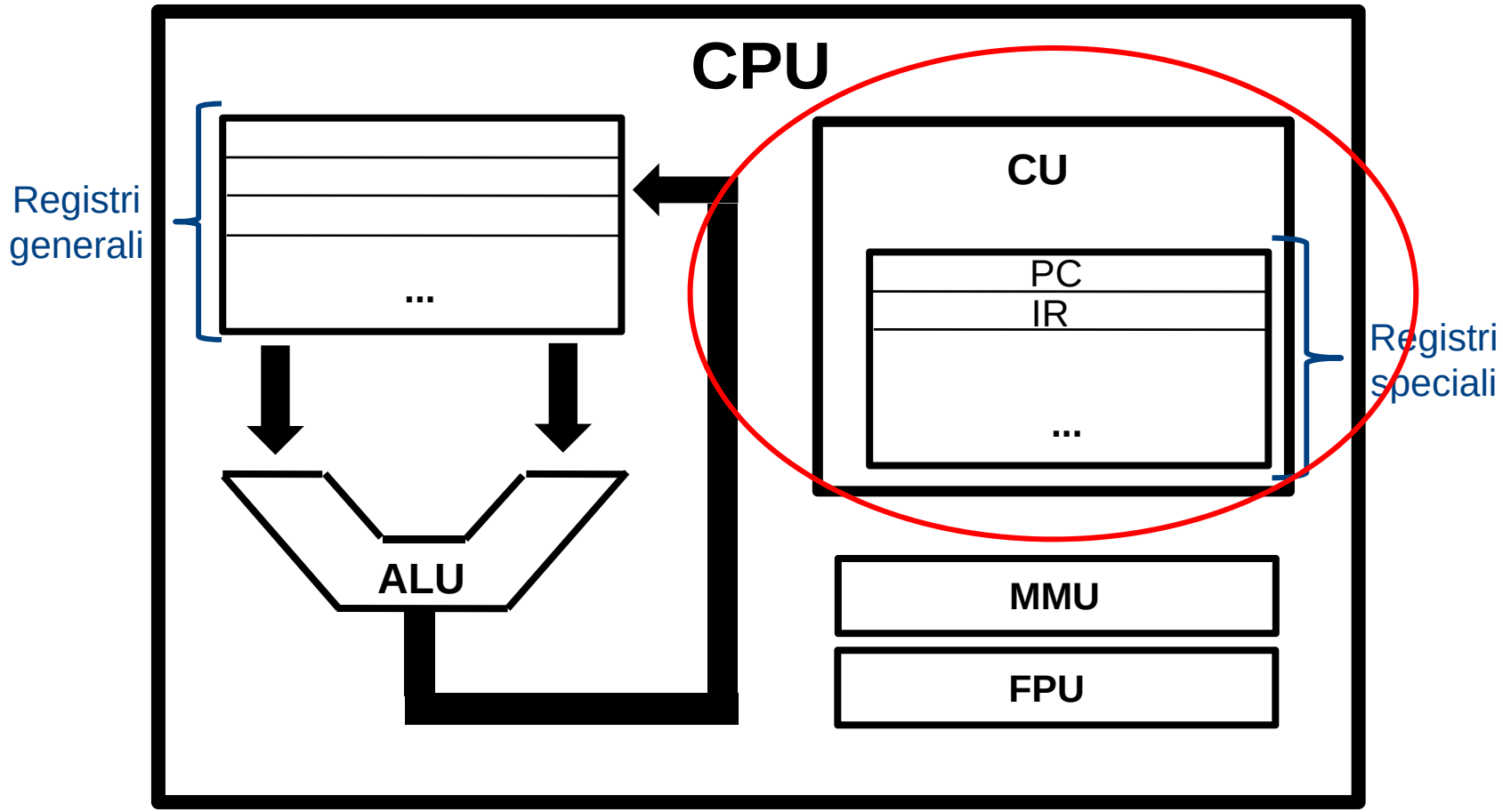


CPU (17)
[Anatomia (semplificata) del processore: la FPU]

- **FPU = Floating Point Unit (Unità di calcolo in virgola mobile)**

- **E' costituita da un insieme di circuiti elettronici in grado di svolgere operazioni che coinvolgono numeri rappresentati in virgola mobile (floating point)**

CPU (18)
[Anatomia (semplificata) del processore: la CU]



CPU (19)
[Anatomia (semplificata) del processore: la CU]

- **CU = Control Unit (Unità di controllo)**

- **Costituita anch'essa da un insieme di circuiti elettronici, costituisce la componente di coordinamento delle varie componenti della CPU e di controllo delle varie attività**

- **Recupera dalla memoria principale l'istruzione da eseguire, la decodifica e ne coordina l'esecuzione, attivando le opportune componenti del processore (con i rispettivi input), controllando che l'esito di ciascuna attività da queste svolte non abbia prodotto errori; coordina il trasferimento dei dati da e verso la memoria principale, ...**

CPU (20)
[Il ciclo fetch-decode-execute]

- **Dopo aver esaminato le funzionalità delle singole componenti, vediamo come esse interagiscono nel funzionamento complessivo del processore**
- **La principale attività del processore è costituita dal cosiddetto ciclo di fetch-decode-execute**
- **Il processore esegue ciclicamente un insieme di azioni che, inizialmente, possiamo approssimativamente descrivere così:**

1. **Fetch**: lettura dalla memoria principale della prossima istruzione da eseguire
2. **Decode**: decodifica dell'istruzione letta al passo 1
3. **Execute**: esecuzione dell'istruzione letta al passo 1

CPU (21)
[Il ciclo fetch-decode-execute]

- **Raffiniamo la descrizione del passo di Fetch, specificando il ruolo che in esso hanno i due registri speciali PC (Program Counter) e IR (Instruction Register)...per semplicità, assumiamo che ogni istruzione macchina occupi una sola locazione di memoria principale:**

1. **Fetch**:
   a) La CU attiva i meccanismi di lettura dalla memoria principale del contenuto della locazione il cui indirizzo è scritto nel PC (tale locazione è quella che contiene la prossima istruzione da eseguire)
   b) Il contenuto della locazione di cui al punto a) viene copiato nell'IR
   c) La CU incrementa di 1 il contenuto del PC
2. **Decode**: decodifica dell'istruzione letta al passo 1
3. **Execute**: esecuzione dell'istruzione letta al passo 1

CPU (21)
[Il ciclo fetch-decode-execute]

La Control Unit usa il contenuto del Program Counter per sapere in quale locazione di memoria principale si trova la prossima istruzione da eseguire. Dopo che tale istruzione è copiata nell'Instruction Register, il contenuto del Program Counter è incrementato di 1 e quindi contiene l'indirizzo della locazione immediatamente successiva a quella il cui contenuto è appena stato copiato nell'Instruction Register: infatti, l'istruzione successiva si troverà *molto probabilmente* in quella locazione

- **Ra … anno i du… Re… a sola loc…**

1. **Fetch**:
   a) La CU attiva i meccanismi di lettura dalla memoria principale del contenuto della locazione il cui indirizzo è scritto nel PC (tale locazione è quella che contiene la prossima istruzione da eseguire)
   b) Il contenuto della locazione di cui al punto a) viene copiato nell'IR
   c) La CU incrementa di 1 il contenuto del PC
2. **Decode**: decodifica dell'istruzione letta al passo 1
3. **Execute**: esecuzione dell'istruzione letta al passo 1

CPU (21)
[Il ciclo fetch-decode-execute]

La CPU non ha occhi per leggere e non ha mani per scrivere, eppure, si dice che essa "legge" e "scrive" in memoria principale... "leggere" una locazione di memoria principale, per il processore, significa copiarne il contenuto in un suo registro. Analogamente, "scrivere" un dato in una locazione di memoria principale, per il processore, significa copiare tale dato (contenuto in un suo registro) nella locazione.

In questo caso, il processore "legge" la prossima istruzione da eseguire, nel senso che essa è copiata nell'Instruction Register

- **Ra… …anno i du… Re… …a sola loc…**

1. **Fetch**:
   a) La CU attiva i meccanismi di lettura dalla memoria principale del contenuto della locazione il cui indirizzo è scritto nel PC (tale locazione è quella che contiene la prossima istruzione da eseguire)
   b) Il contenuto della locazione di cui al punto a) viene copiato nell'IR
   c) La CU incrementa di 1 il contenuto del PC
2. **Decode**: decodifica dell'istruzione letta al passo 1
3. **Execute**: esecuzione dell'istruzione letta al passo 1

CPU (22)
[Il ciclo fetch-decode-execute]

- **Raffiniamo la descrizione degli altri due passi (Decode e Execute):**
  1. **Fetch**:
     a) La CU attiva i meccanismi di lettura dalla memoria principale del contenuto della locazione il cui indirizzo è scritto nel PC (tale locazione è quella che contiene la prossima istruzione da eseguire)
     b) Il contenuto della locazione di cui al punto a) viene copiato nell'IR
     c) La CU incrementa di 1 il contenuto del PC
  2. **Decode**: La CU decodifica l'istruzione contenuta nell'IR e identifica le azioni da compiere per eseguirla
  3. **Execute**:
     a) Se l'istruzione è aritmetico-logica, la CU predispone gli input per la ALU (o, eventualmente, la FPU), le specifica quale operazione calcolare e dove inserire il risultato. La ALU (o la FPU) esegue il proprio compito, poi la CU controlla che tutto sia andato a buon fine (se non è così, intraprende le azioni necessarie)
     b) Se l'istruzione è di trasferimento dati, la CU attiva i meccanismi di accesso alla memoria principale (per leggere o scrivere), specificando l'indirizzo della locazione a cui accedere, l'eventuale dato da scrivere, l'eventuale registro in cui copiare il dato letto, ecc.
     c) Se l'istruzione è di controllo, la CU attiva i meccanismi per la verifica dell'eventuale condizione di salto e, se questa è verificata, scrive nel PC l'indirizzo della locazione di memoria principale che contiene l'istruzione a cui bisogna saltare

CPU (23)
[Esempio di esecuzione di un semplice frammento di programma]

- **Per agevolare la comprensione del programma da parte dei lettori umani, scriviamo le istruzioni in un linguaggio mnemonico, anziché come sequenze di valori binari (ma nel calcolatore esse saranno comunque rappresentate come sequenze di valori binari)**

LOAD R0 5000 ➔ "copia nel registro R0 il contenuto della locazione di memoria il cui indirizzo è 5000"

LOAD R1 5001 ➔ "copia nel registro R1 il contenuto della locazione di memoria il cui indirizzo è 5001"

ADD R0 R1 ➔ "somma i contenuti di R0 e R1 e metti il risultato in R0"

STORE R0 5002 ➔ "copia il contenuto di R0 nella locazione 5002"

JUMPNZ R0 2009 ➔ "se il contenuto di R0 non è 0, salta all'istruzione che si trova nella locazione 2009"

LOAD R0 5003 ➔ "copia in R0 il contenuto della locazione 5003"

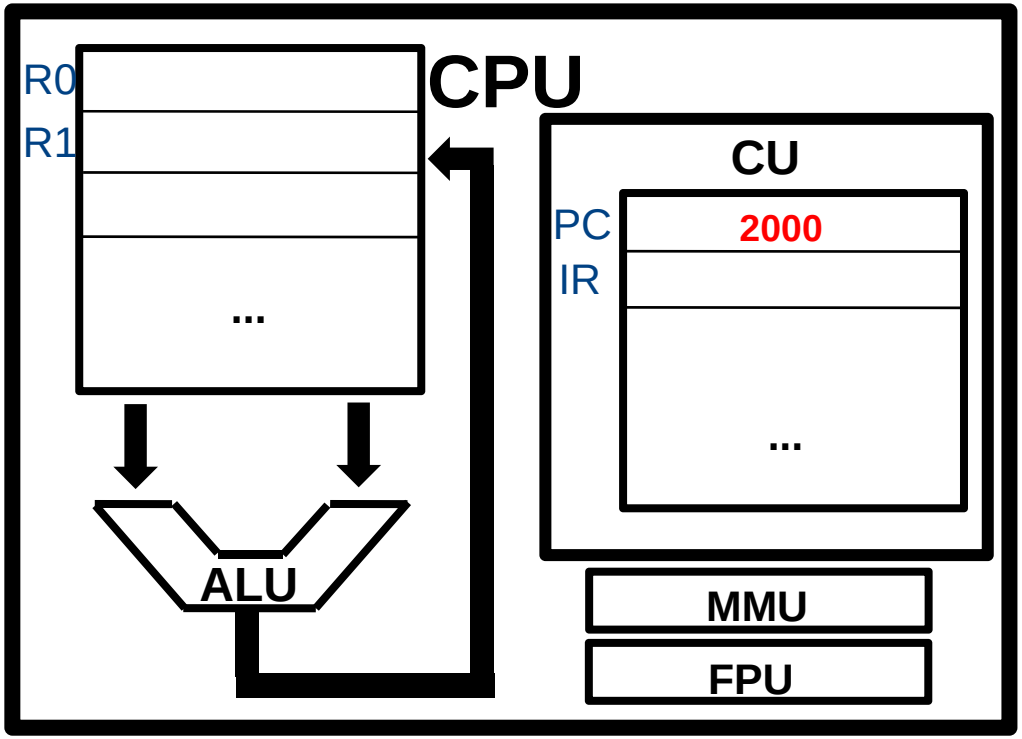
LOAD R1 5004 ➔ "copia in R1 il contenuto della locazione 5004"

ADD R0 R1 ➔ "somma i contenuti di R0 e R1 e metti il risultato in R0"

STORE R0 5002 ➔ "copia il contenuto di R0 nella locazione 5002"

HALT ➔ "termina"

CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]

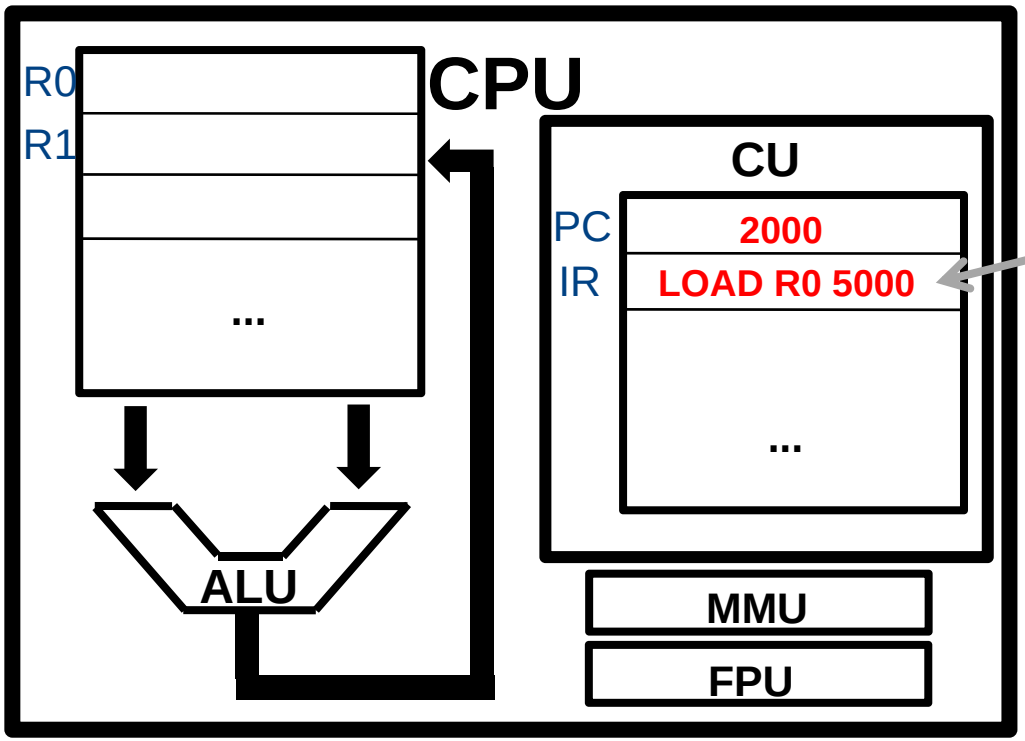


Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

Per migliorare la comprensibilità da parte dei lettori umani, rappresentiamo le istruzioni in un formato mnemonico e i valori numerici in notazione decimale

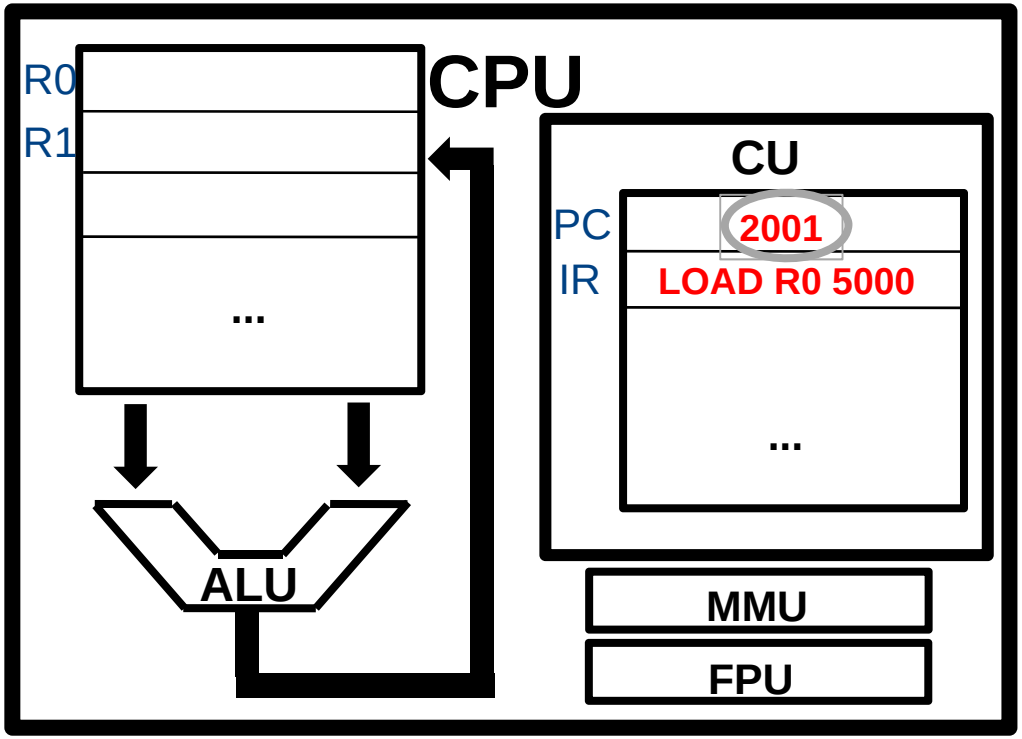
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

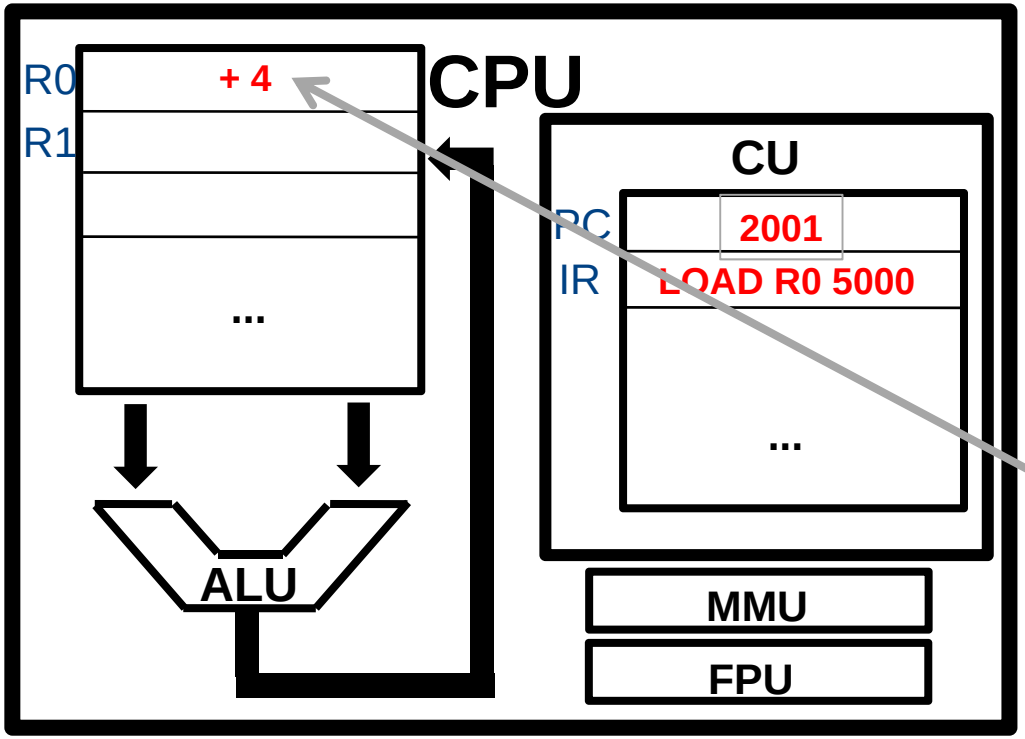
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

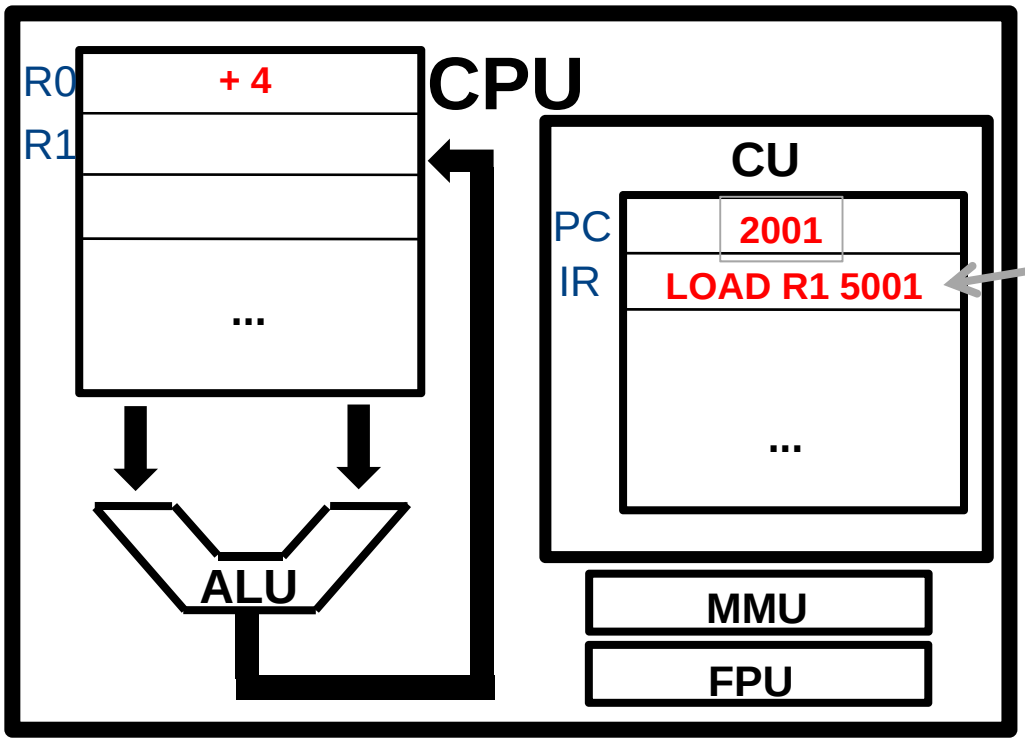
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

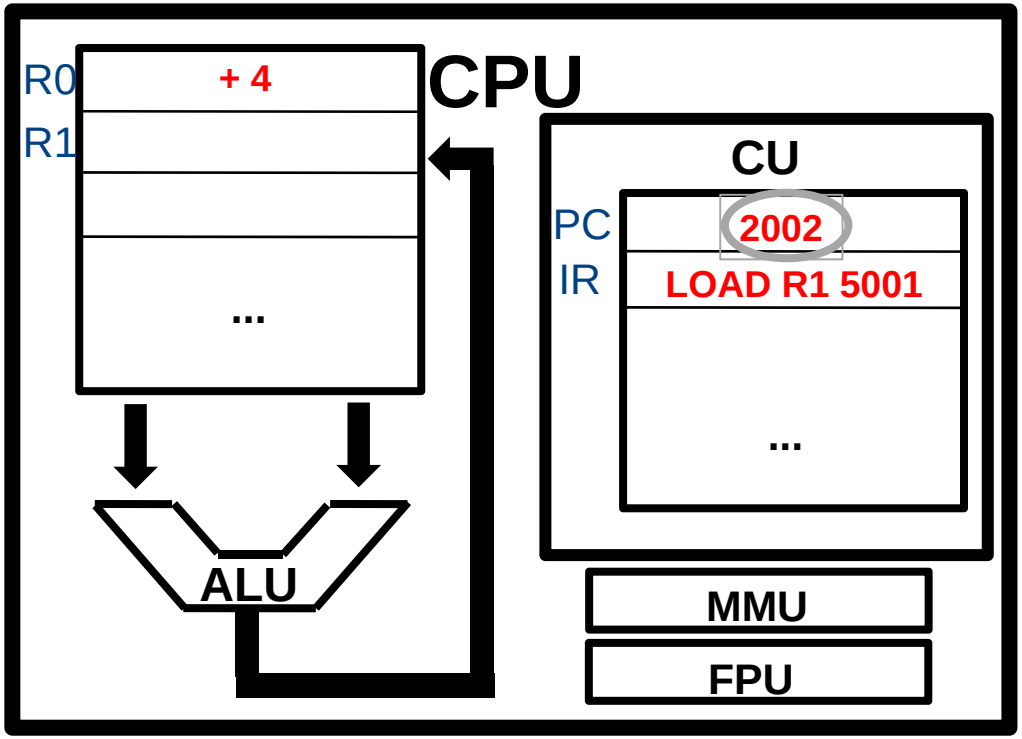
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

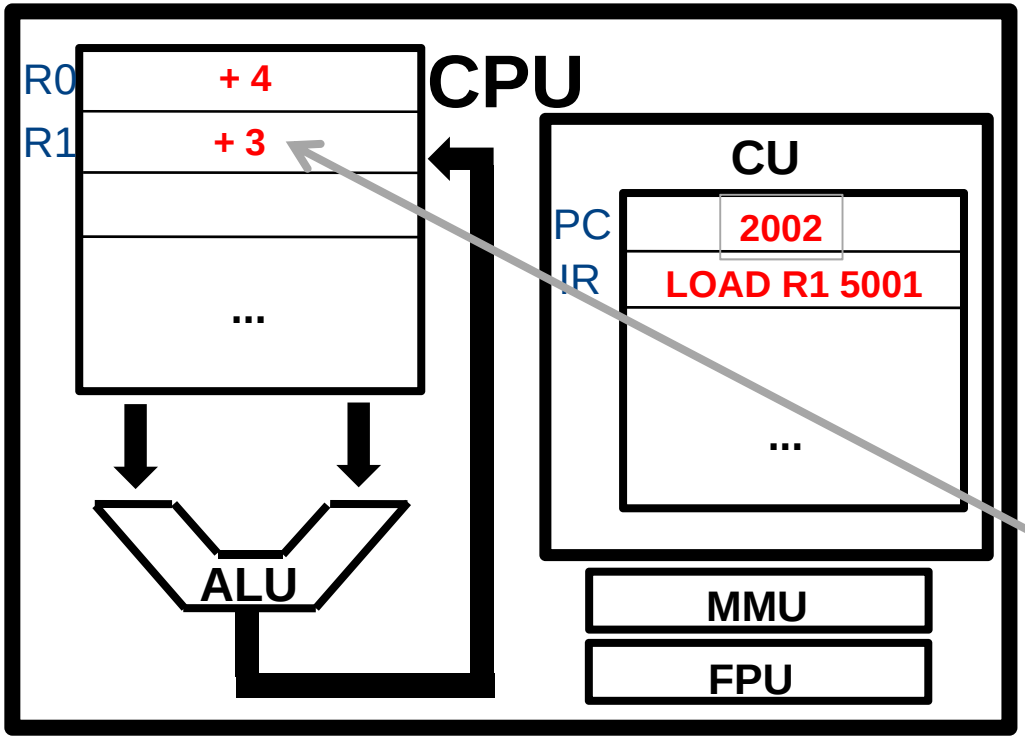
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

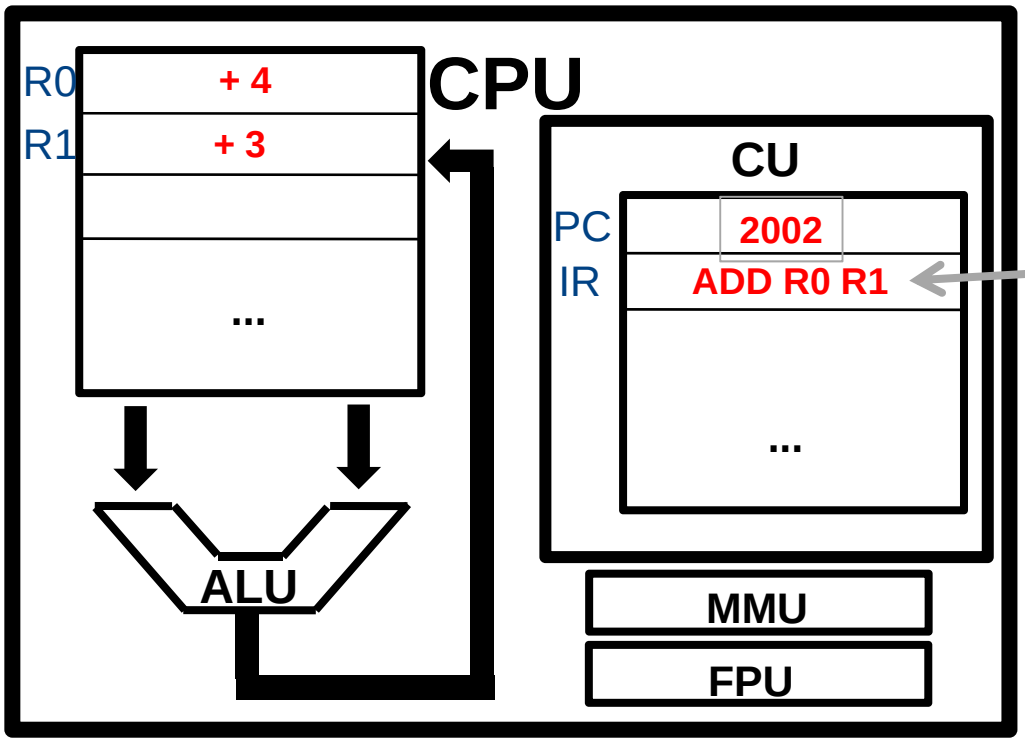
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

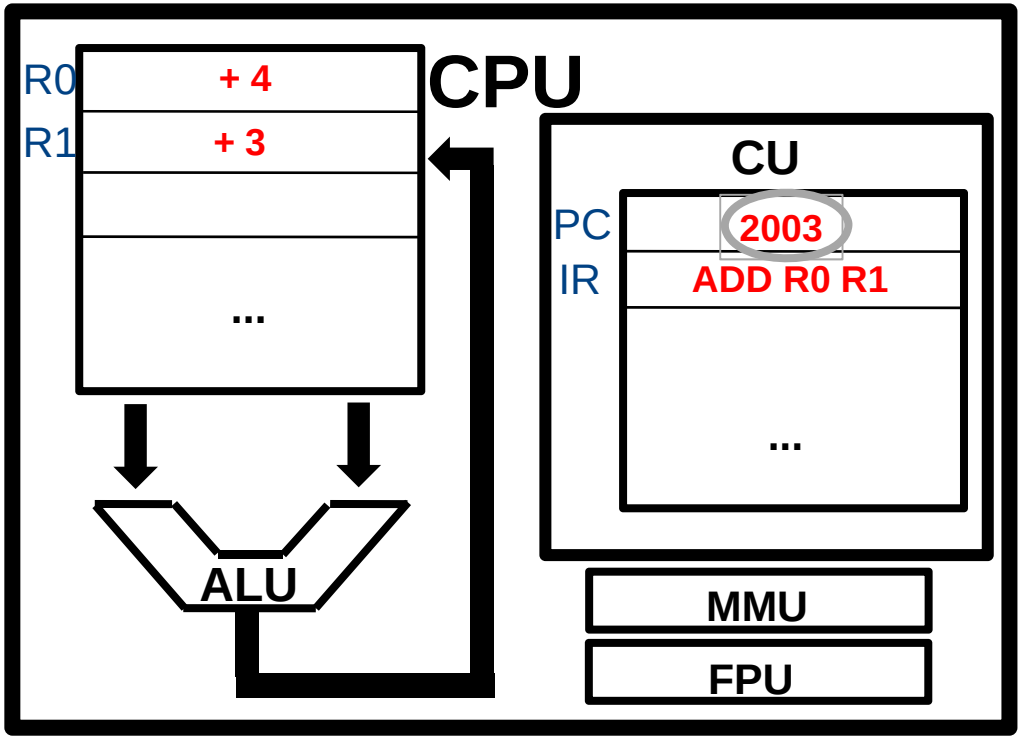
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

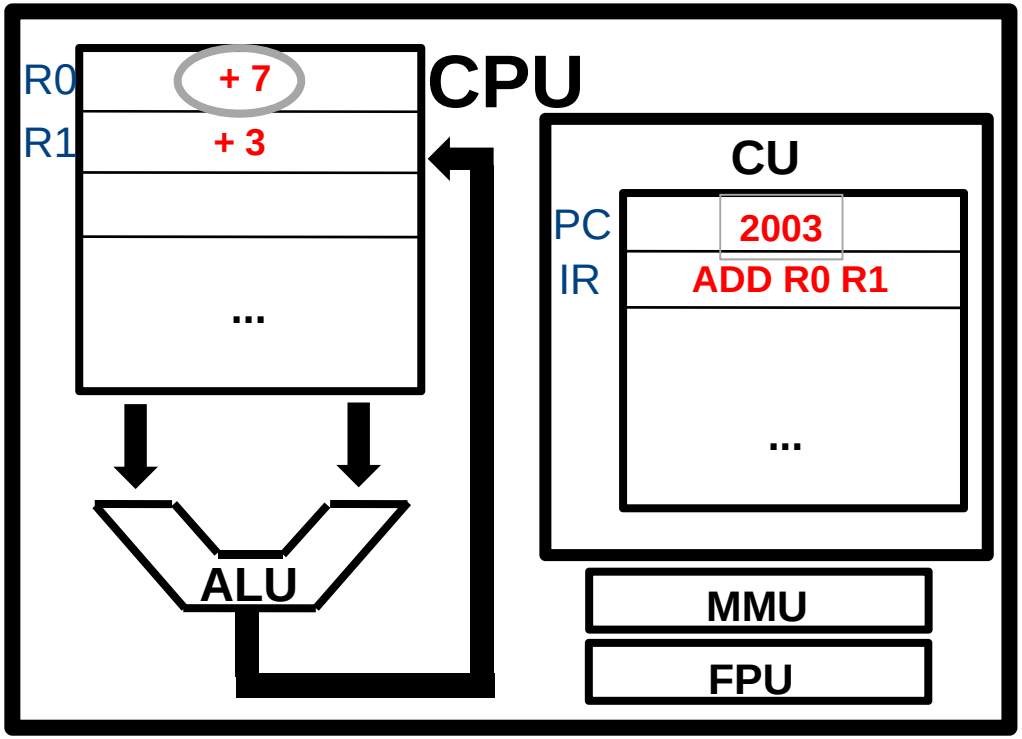
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

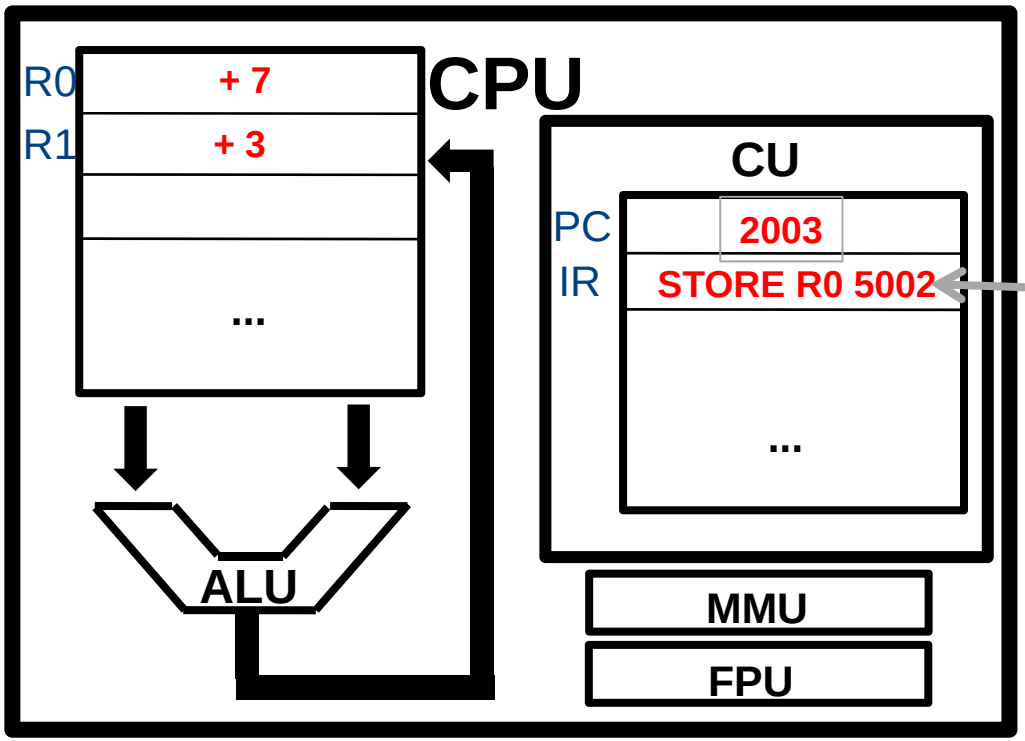
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

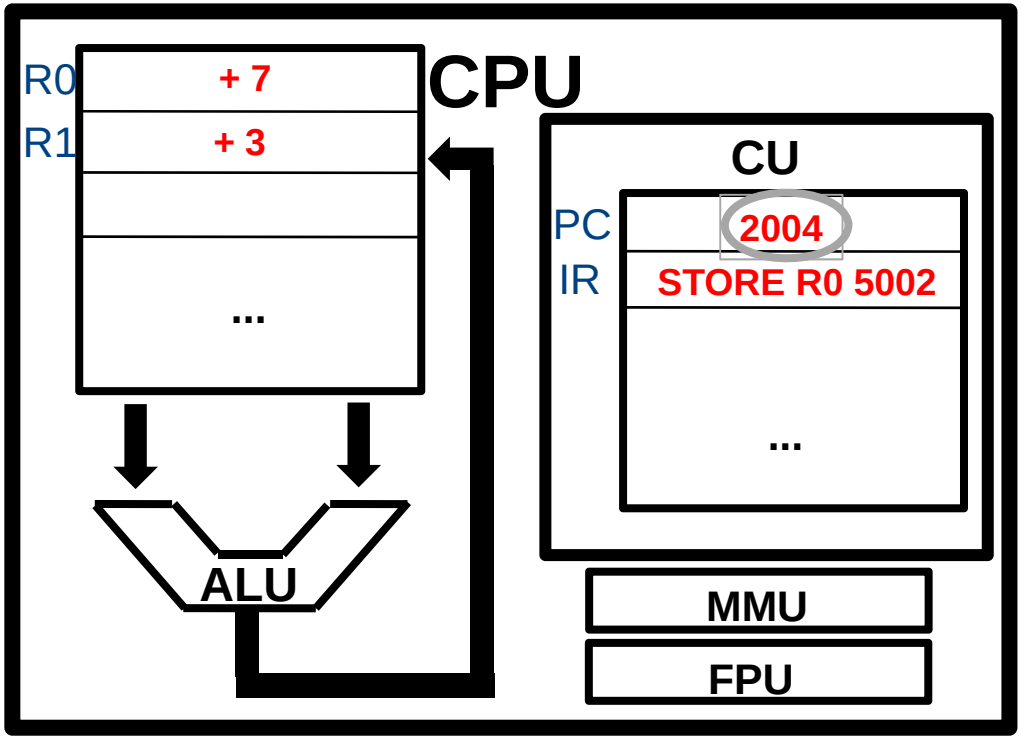
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

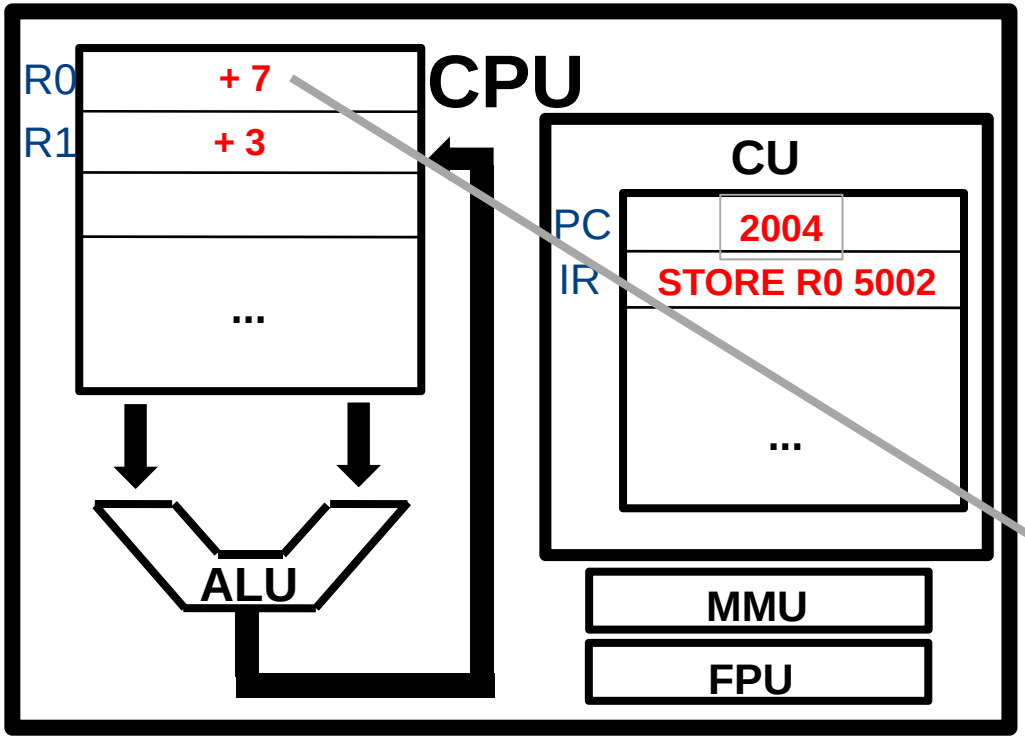
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	
5003	+ 6
5004	- 9
	...

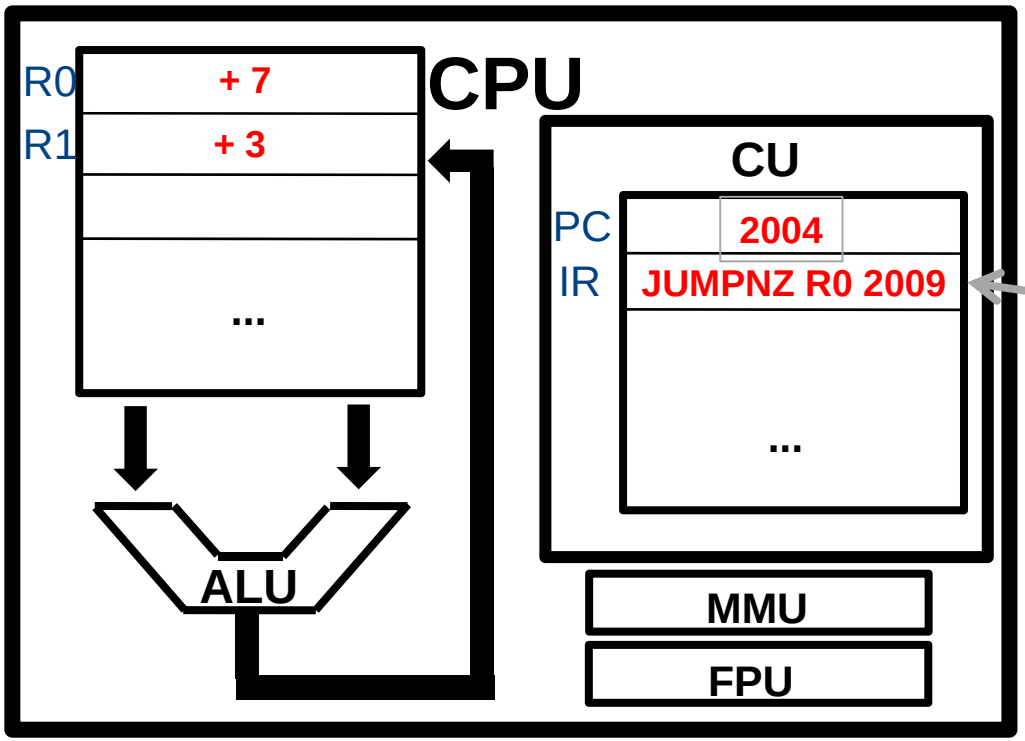
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

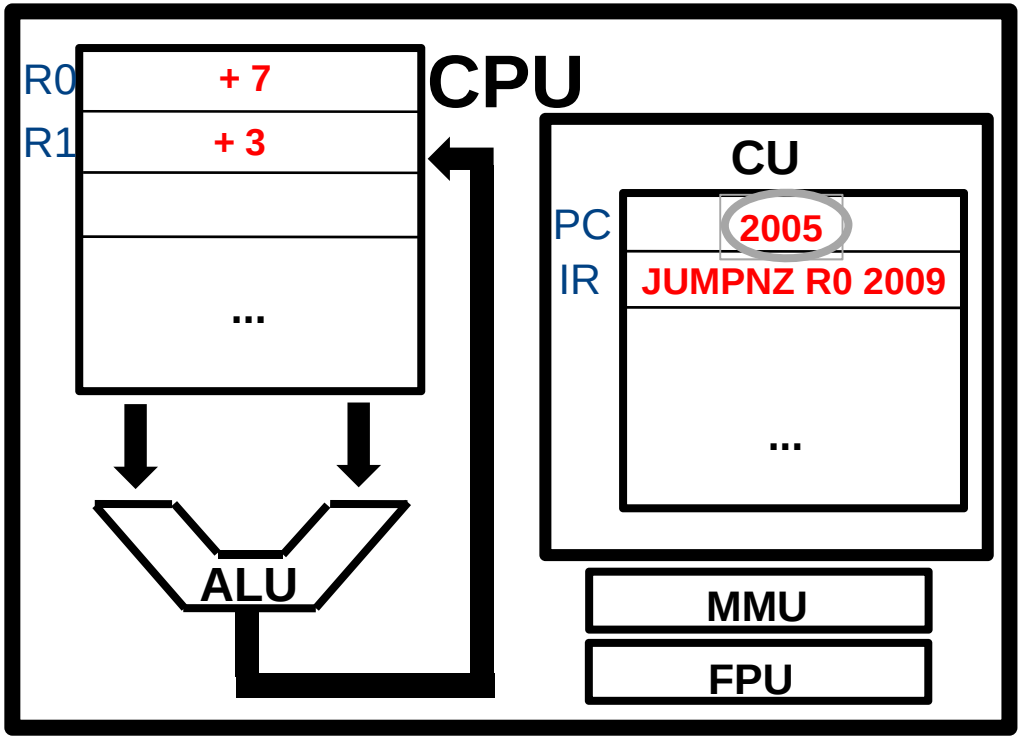
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

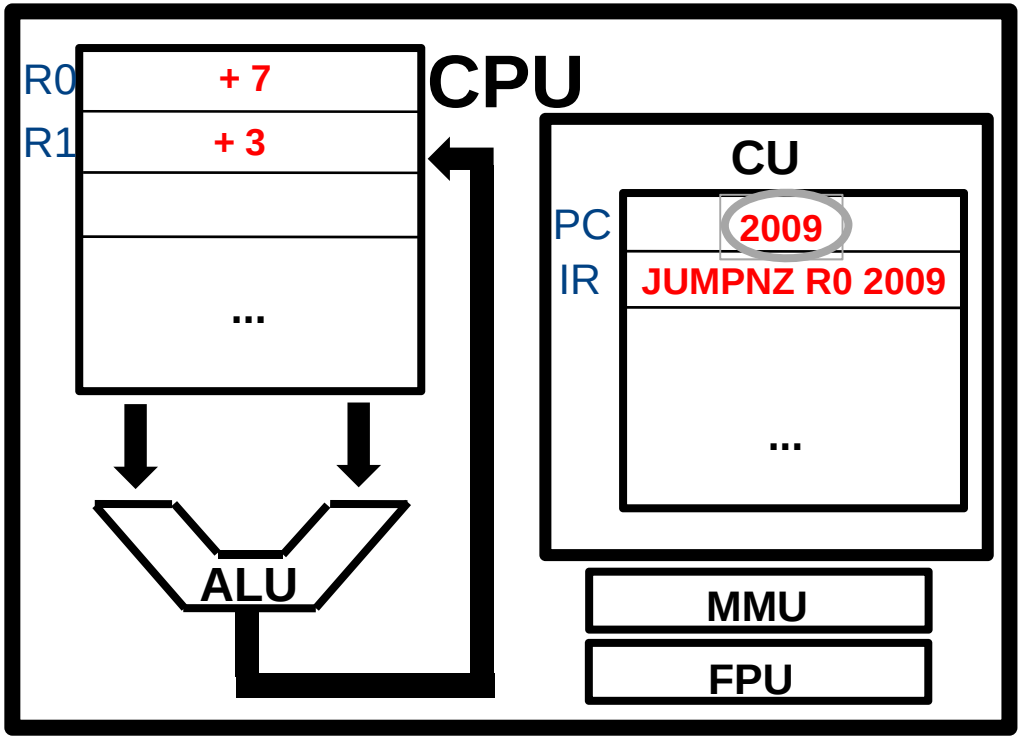
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

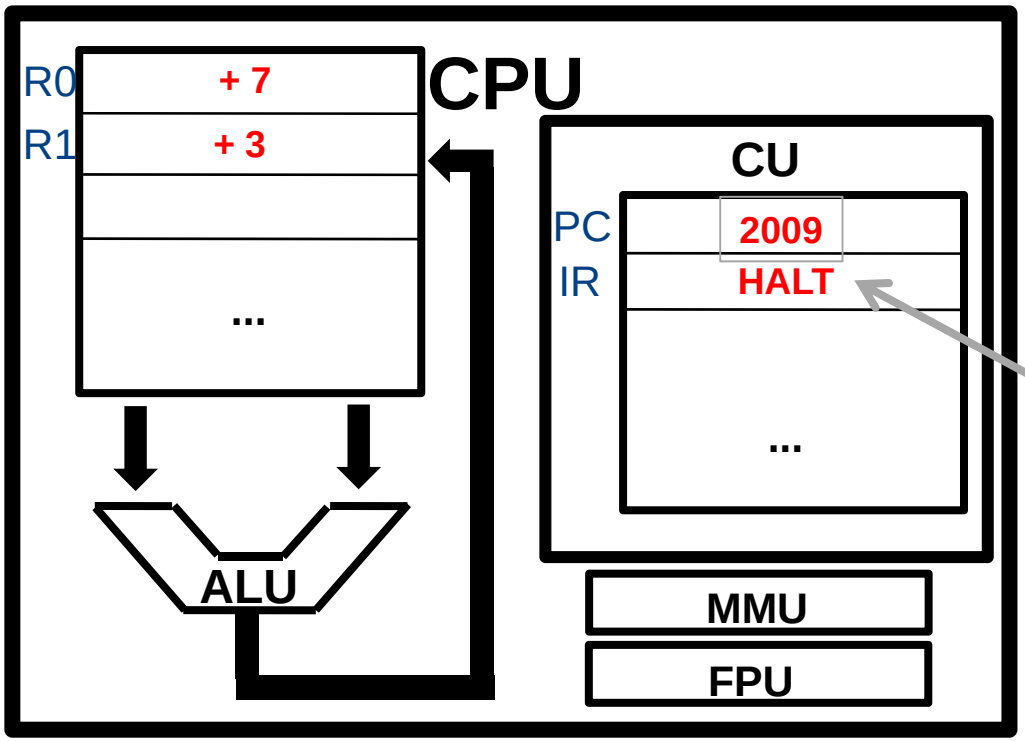
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

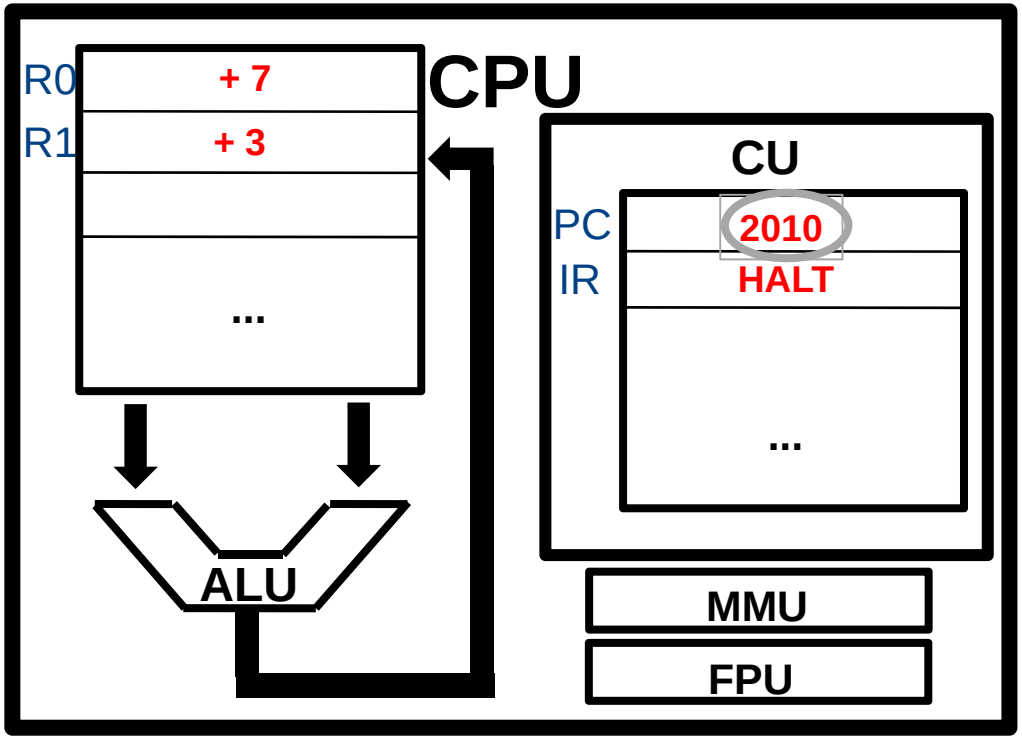
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

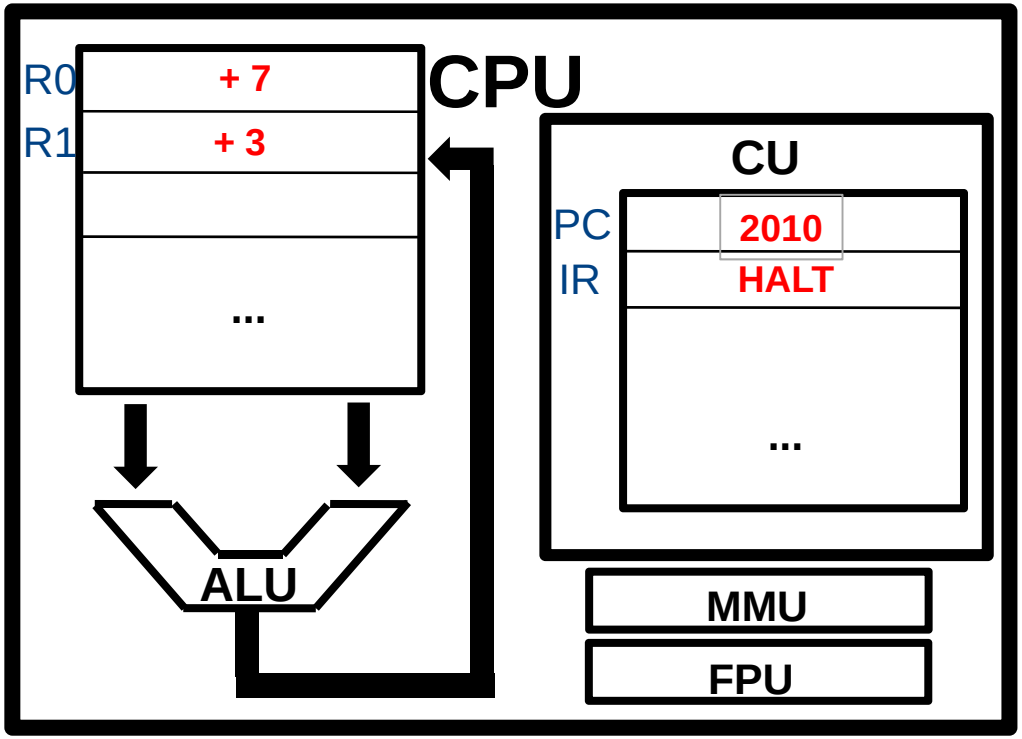
CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

CPU (24)
[Esempio di esecuzione di un semplice frammento di programma]



Memoria principale

	...
2000	LOAD R0 5000
2001	LOAD R1 5001
2002	ADD R0 R1
2003	STORE R0 5002
2004	JUMPNZ R0 2009
2005	LOAD R0 5003
2006	LOAD R1 5004
2007	ADD R0 R1
2008	STORE R0 5002
2009	HALT
	...
5000	+ 4
5001	+ 3
5002	+ 7
5003	+ 6
5004	- 9
	...

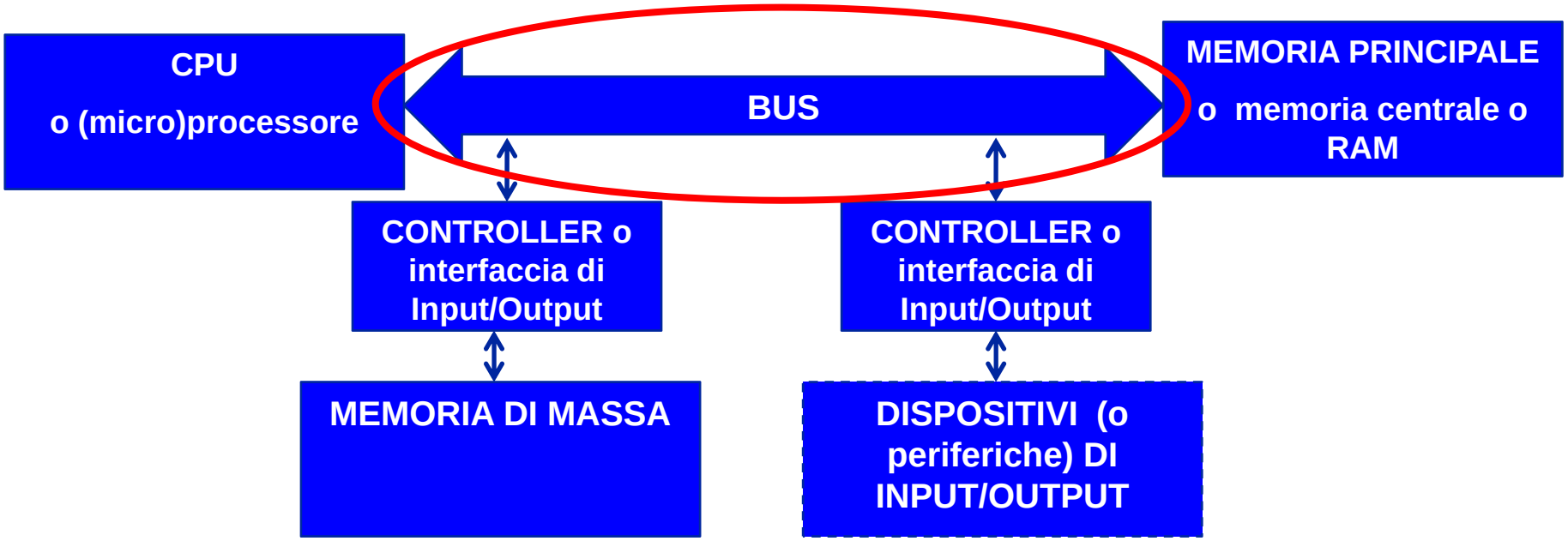
FINE ESECUZIONE!!!!

CPU (25)

- **Alcune CPU**

Immagine tratta da Wikimedia Commons

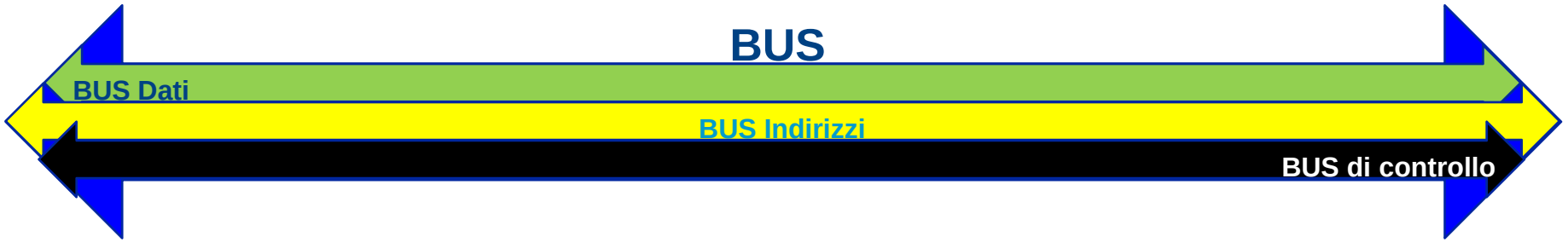
BUS (1)


CPU
o (micro)processore
BUS
MEMORIA PRINCIPALE
o memoria centrale o RAM
CONTROLLER o interfaccia di Input/Output
CONTROLLER o interfaccia di Input/Output
MEMORIA DI MASSA
DISPOSITIVI (o periferiche) DI INPUT/OUTPUT

BUS (2)

- **Collegamento fisico fra le componenti che funge da canale di comunicazione fra di esse**
- **Le informazioni che le componenti si scambiano sono codificate in segnali elettrici trasportati dal bus**
- **In figura è rappresentato un singolo bus a cui sono connesse la CPU, la memoria principale e, attraverso le loro interfacce, anche la memoria di massa e le altre periferiche**
- **E' attraverso il bus che transitano, ad esempio, i dati scambiati fra la CPU e la memoria principale nelle operazioni di lettura o scrittura**

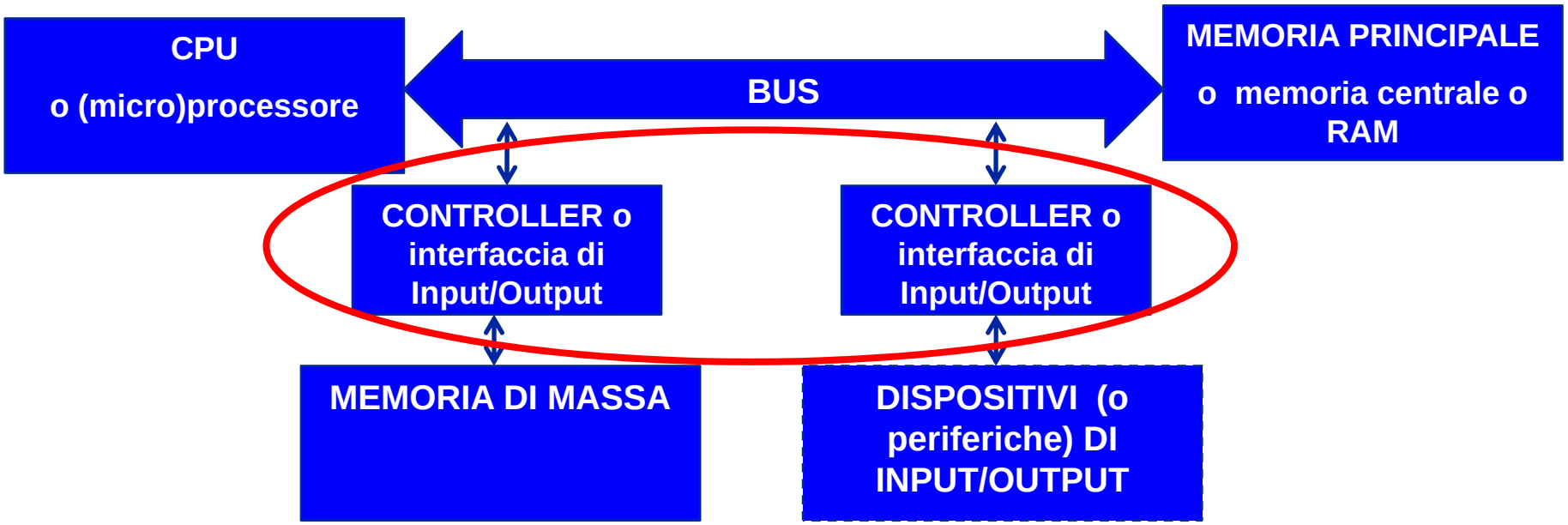
BUS (3)
[Anatomia (semplificata) del bus]



- **Il BUS, nel suo complesso, è strutturato in tre bus specifici:**
  1. bus dati: usato dalle componenti del processore per lo scambio di dati (ad esempio, è attraverso questo bus che transitano i dati letti e scritti dalla CPU in memoria principale)
  2. bus indirizzi: usato dalla CPU per specificare gli indirizzi delle celle di memoria principale in cui essa vuole scrivere o da cui vuole leggere. Usato anche per specificare i dispositivi di I/O con cui comunicare[1]
  3. bus di controllo: usato per inviare segnali di controllo, quali, ad esempio, quelli che specificano che una richiesta di accesso alla memoria principale è una richiesta di lettura (o, analogamente, di scrittura)

1 Come detto, la dimensione del bus indirizzi (cioè il numero di bit che possono transitare in esso per ogni indirizzo) determina il massimo numero di locazioni di memoria indirizzabili

CONTROLLER (1)



CONTROLLER (2)

- **Sono dispositivi elettronici che fungono da interfaccia fra le periferiche (compresi i dispositivi di memoria di massa) e il sistema CPU-memoria principale.**
- **Sono collegati al bus e, pertanto, in grado di comunicare direttamente con il sistema CPU-memoria principale**
- **Attraverso i controller passano le richieste alle periferiche da parte della CPU e le risposte di queste verso sistema CPU-memoria principale**
- **Pilotano direttamente la/le periferica/periferiche a cui sono collegati**
- **Esempi di controller: scheda video, controller del disco, controller USB, ...**

CONTROLLER (3)

- **Un controller USB**

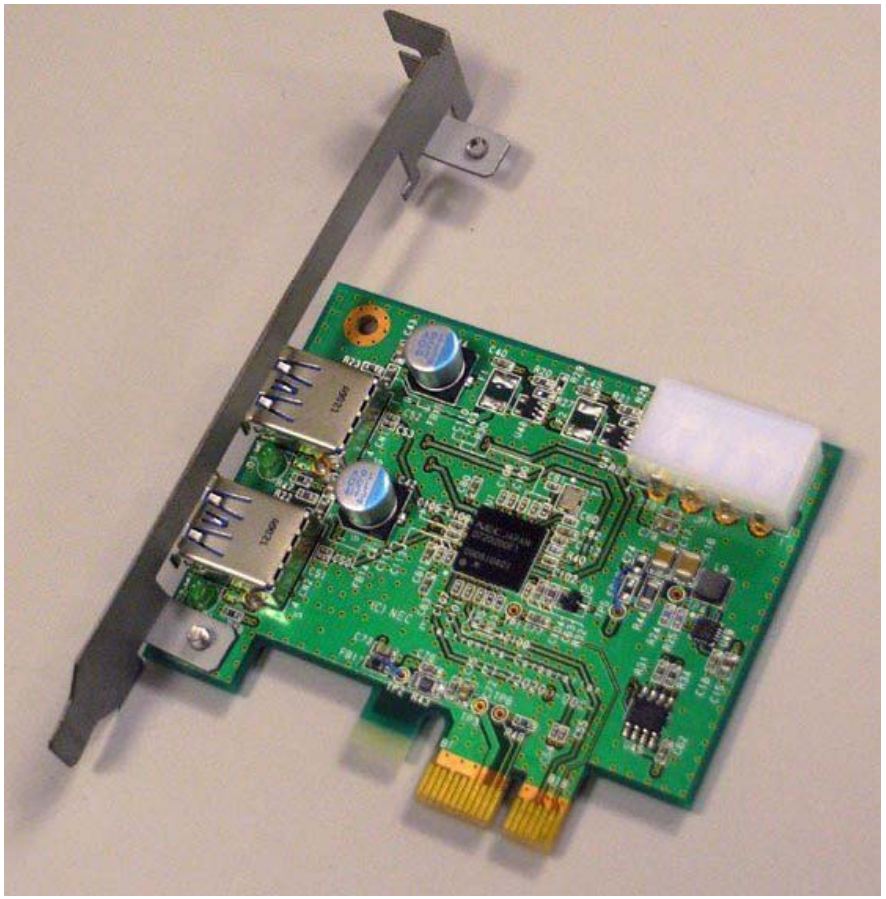
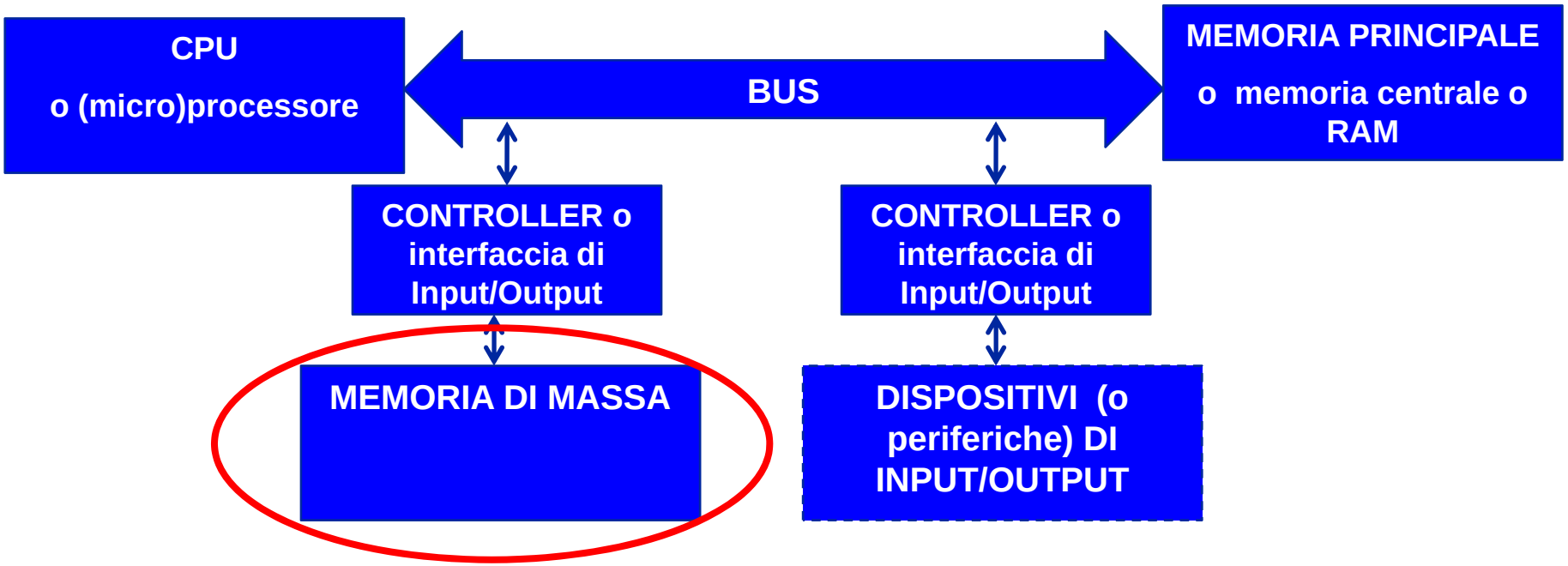


Immagine tratta da un articolo apparso sul sito http://www.dvhardware.net

MEMORIA DI MASSA (1)



MEMORIA DI MASSA (2)

- **Come abbiamo visto, la memoria principale è volatile, cioè è in grado di mantenere le informazioni in essa memorizzate solo fino a che è alimentata elettricamente. Quando viene meno l'alimentazione elettrica, tali informazioni sono perse**

- **Inoltre, nonostante l'evoluzione tecnologica consenta la realizzazione di memorie principali sempre più capaci e con costi sempre più ridotti, la quantità di memoria di cui necessitiamo per immagazzinare informazioni è sempre crescente**

- **La memoria di massa soddisfa le necessità di immagazzinare in maniera persistente (cioè non volatile) grandi quantità di dati a costi relativamente bassi**

MEMORIA DI MASSA (3)

- **Le principali tecnologie per la realizzazione di memorie di massa sono:**

1. tecnologie magnetiche: es. dischi magnetici, hard disk
2. tecnologie ottiche: es. CD, DVD
3. tecnologie elettroniche: es. memorie flash

MEMORIA DI MASSA (4)
[Tecnologie magnetiche]

- **Sono realizzati con queste tecnologie i dischi e i nastri magnetici**

- **Entrambi i tipi di dispositivi sono realizzati da un substrato di materiale non magnetico, ricoperto di un sottile strato di una particolare sostanza magnetica**

- **La particolarità della sostanza magnetica consiste nel fatto che, attraverso una *testina* posta molto in prossimità della superficie magnetica, è possibile produrre delle alterazioni localizzate e persistenti del "verso di magnetizzazione" della sostanza**

MEMORIA DI MASSA (5)
[Tecnologie magnetiche]

- **In base al verso di magnetizzazione, è possibile caratterizzare due possibili stati di magnetizzazione in cui opportune regioni di superficie possono trovarsi in ogni istante**

- **La testina è in grado di indurre un cambiamento da uno stato di magnetizzazione ad un altro (*operazione di scrittura*). L'esito di una tale operazione di scrittura persiste fino a che una successiva operazione di scrittura non cambia nuovamente lo stato di magnetizzazione della regione**

- **La stessa testina che scrive[1] è poi in grado di *leggere*, cioè di riconoscere lo stato di magnetizzazione delle suddette regioni di superficie**

- **➔ associando uno dei due stati di magnetizzazione al simbolo '0' e l'altro al simbolo '1', è possibile sfruttare questa proprietà magnetica di certa materia (assieme al meccanismo di lettura e scrittura) per realizzare fisicamente il bit**

1 Anche se fisicamente le due testine (quella di lettura e quella di scrittura) possono essere distinte, esse sono sempre poste ad una distanza estremamente ridotta l'una dall'altra e, per i nostri scopi, possono essere considerate un'unica testina

MEMORIA DI MASSA (6)
[Tecnologie magnetiche: dischi magnetici]

- **Un disco magnetico è costituito di un supporto di materiale non magnetico a forma di disco (!), ricoperto di sostanza magnetica, solitamente su entrambe le superfici**
- **E' presente una testina di lettura/scrittura per ogni superficie magnetizzata, posta molto vicino alla superficie (pochi nanometri), ma che non viene mai a contatto con questa (a meno di guasti)**
- **La testina è situata ad un'estremità di un braccio meccanico, in grado di spostarla radialmente**
- **Quando è attivo, il disco è in rapida rotazione attorno ad un perno e la testina può leggere o scrivere le regioni di superficie che le passano sotto**

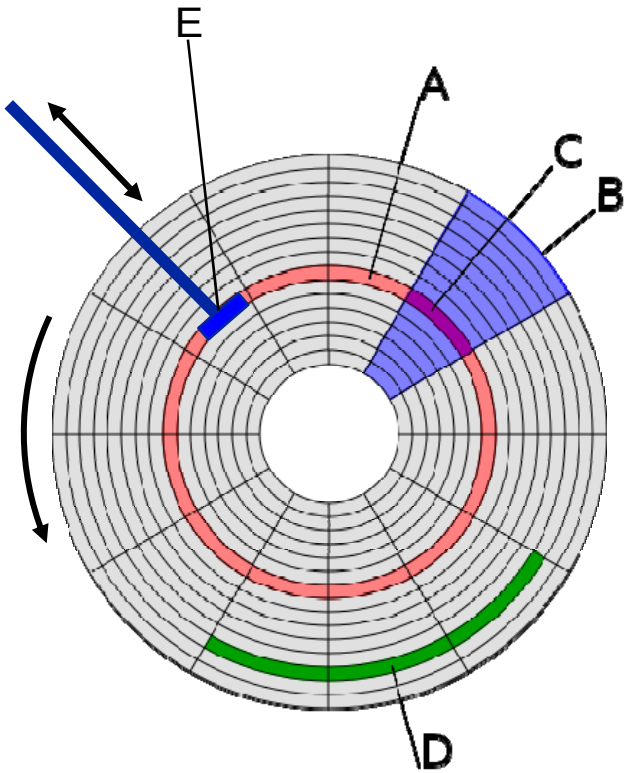
MEMORIA DI MASSA (7)
[Tecnologie magnetiche: dischi magnetici]

- **Le informazioni su disco magnetico sono memorizzate secondo un'organizzazione del disco basata su tracce e settori: ogni superficie magnetizzata del disco è (magneticamente) suddivise in *tracce* circolari concentriche e ogni traccia è suddivisa in *settori*. Le tracce (e analogamente i settori) sono fra loro separate da "spazi di confine" detti *gap*.**

- **Le informazioni sono memorizzate nei settori**

- **Ogni trasferimento di informazioni da un disco al sistema CPU-memoria principale e viceversa trasferisce sempre il contenuto di uno o più settori**

MEMORIA DI MASSA (8)
[Tecnologie magnetiche: dischi magnetici]



A: traccia

B: settore geometrico

C: settore di traccia (può contenere informazioni)

D: per i nostri scopi, non interessa

E: testina

Il processo di creazione delle tracce e dei settori è detto formattazione (a basso livello)

Immagine tratta da Wikimedia Commons

MEMORIA DI MASSA (9)
[Tecnologie magnetiche: dischi magnetici]

- **Quando è attivo, il disco ruota rapidamente, a velocità costante, attorno al proprio perno**

- **Per leggere o scivere un certo settore, il meccanismo deve:**
  1. spostare radialmente la testina fino a farle raggiungere la traccia che contiene il settore
  2. aspettare fino a che la rotazione del disco non porta il settore desiderato sotto la testina
  3. leggere o scrivere le informazioni

- **Il tempo di accesso (in lettura o scrittura) ad un settore consiste di:**
  1. seek time (tempo di ricerca), necessario a spostare la testina sulla traccia voluta
  2. latency time (tempo di latenza), che intercorre dal momento in cui la testina è correttamente posizionata sulla traccia al momento in cui il settore desiderato passa sotto di essa
  3. tempo di trasferimento informazioni

- **Essendo coinvolte parti meccaniche, i tempi di accesso ad un disco magnetico sono molto elevati, se comparati ai tempi di funzionamento del sistema CPU-memoria principale**

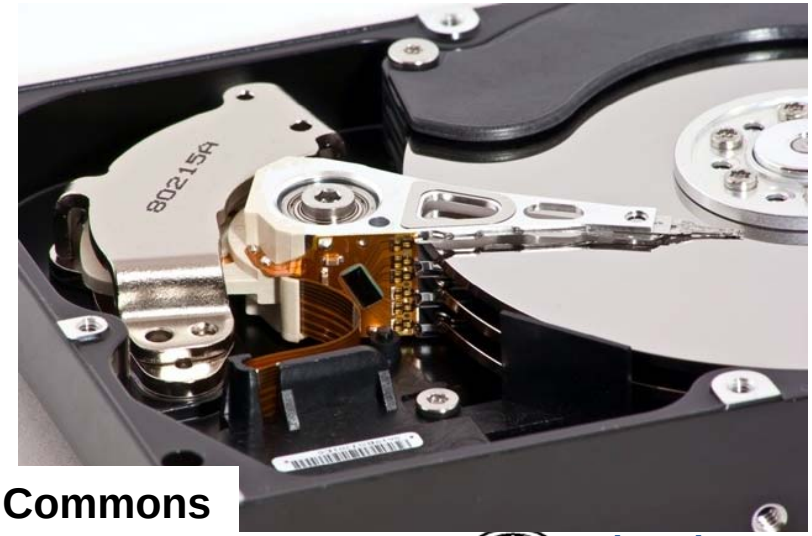
MEMORIA DI MASSA (10)
[Tecnologie magnetiche: dischi magnetici]

- **I principali rappresentanti di questa tecnologia sono attualmente i *dischi rigidi* (*hard disk*)**

- **Un hard disk contiene un certo numero di dischi magnetici impilati in uno stesso perno centrale, con una testina per ogni superficie magnetizzata**

- **Per fare un esempio, negli attuali PC è comune avere un hard disk interno con capacità di 80 GB – 250 GB, ma esistono anche hard disk più capienti (ad es. 1.5 TB)[1]**

1 Dati soggetti a rapida obsolescenza



Immagini tratte da Wikimedia Commons

MEMORIA DI MASSA (11)
[Tecnologie magnetiche: dischi magnetici]

- **Altri rappresentanti di questa tecnologia: i floppy disk**
- **Sono ormai quasi in disuso, soppiantati da memorie flash e dischi ottici**

Immagini tratte da Wikimedia Commons

MEMORIA DI MASSA (12)
[Tecnologie magnetiche: nastri magnetici]

- **Altri prodotti della tecnologia magnetica sono i nastri magnetici**

- **Grazie al loro basso costo e alla loro grande capacità, sono ancora oggi utilizzati, prevalentemente per il backup dei dati (anche se, a questo fine, si utilizzano anche tecnologie ottiche)**

- **Sono dispositivi piuttosto lenti, rispetto ad altri utilizzati nelle memorie di massa**

MEMORIA DI MASSA (13)
[Tecnologie ottiche: CD e DVD]

- **Nei dischi ottici, la memorizzazione delle informazioni e la loro lettura avviene per mezzo di un raggio laser**
- **In fase di scrittura, il raggio laser produce delle "bruciature" in unità di superficie riflettente**
- **La presenza o l'assenza di bruciature in elementi di superficie produce differenze nella riflessione del raggio laser usato per leggere e questa caratteristica consente la realizzazione fisica del bit nei dischi ottici: il raggio di lettura può essere riflesso in due modi possibili, sfruttati per rappresentare i simboli '0' e '1'**
- **I dispositivi in grado di scrivere su dischi ottici sono detti *masterizzatori***
- **A differenza dei dischi magnetici, in quelli ottici i dati sono memorizzati in un'unica traccia che si sviluppa a spirale a partire dal centro del disco (il DVD-RAM fa eccezione ed ha tracce concentriche)**

MEMORIA DI MASSA (14)
[Tecnologie ottiche: CD e DVD]

- **Quando è attivo, il disco ottico è in rapida rotazione attorno ad un perno**
- **Principali rappresentanti di questa tecnologia sono i CD (Compact Disc) e i DVD (Digital Versatile Disc)**
- **I CD hanno una capacità tipica di 600 MB – 700 MB [1]**
- **I DVD sono basati su una tecnologia leggermente diversa da quella dei CD che consente una maggiore capacità. Attualmente hanno tipicamente capacità di 4,7 GB, 8,5 GB, 9,4 GB e 17 GB[1]**

1 Dati soggetti a rapida obsolescenza

MEMORIA DI MASSA (15)
[Tecnologie ottiche: CD e DVD]

- **A seconda delle possibilità di lettura e scrittura offerte, i CD e i DVD si suddividono in varie tipologie:**
  1. CD-ROM e DVD-ROM (ROM = Read Only Memory): informazioni scritte una volta per tutte dal produttore e soltanto leggibili dagli utenti finali. Sono tipicamente usati per la distribuzione di software, prodotti multimediali, ecc.

  2. CD-R, DVD-R, DVD+R (R = Recordable): sono anche detti WORM (Write Once Read Many); sono venduti vergini, possono essere scritti una sola volta e letti molte volte. I DVD di questo tipo possono basarsi su tecnologie leggermente di verse, da qui le due sottocategorie DVD-R e DVD+R

  3. CD-RW, DVD-RW, DVD+RW, DVD-RAM (RW = ReWritable, RAM=Random Access Memory): possono essere scritti e letti più volte

MEMORIA DI MASSA (17)
[Tecnologie ottiche: CD e DVD]

Immagini tratte da Wikimedia Commons

MEMORIA DI MASSA (18)
[Tecnologie ottiche: BD]

- **Altra tipologia di dischi ottici: BD (Blu-ray Disc)**

- **Ha capacità superiori rispetto ai DVD (ma ha costi di produzione maggiori e la sua diffusione è più limitata): attualmente i BD hanno una capacità tipica di circa 25 GB[1]**

- **Fino a poco tempo fa, era in concorrenza con l' HD-DVD (High Definition Digital Versatile Disc). Ora questa tipologia di disco sembra definitivamente abbandonata da suoi principali sostenitori**

1 dato soggetto a rapida obsolescenza

MEMORIA DI MASSA (19)
[Tecnologie elettroniche: memorie flash]

- **Sono memorie persistenti e riscrivibili basate su tecnologia elettronica**
- **A differenza di dischi o nastri magnetici e dischi ottici, nelle memorie flash non vi sono parti meccaniche in movimento (quindi sono più veloci)**
- **Sono soggette ad usura più rapida rispetto ai dischi magnetici (le operazioni di scrittura "usurano" la memoria, anche se la quantità di possibili operazioni di scrittura le rendono comunque adatte a molti scopi, quale quello dello scambio informazioni)**
- **Hanno dimensioni ridotte**

MEMORIA DI MASSA (20)
[Tecnologie elettroniche: memorie flash]

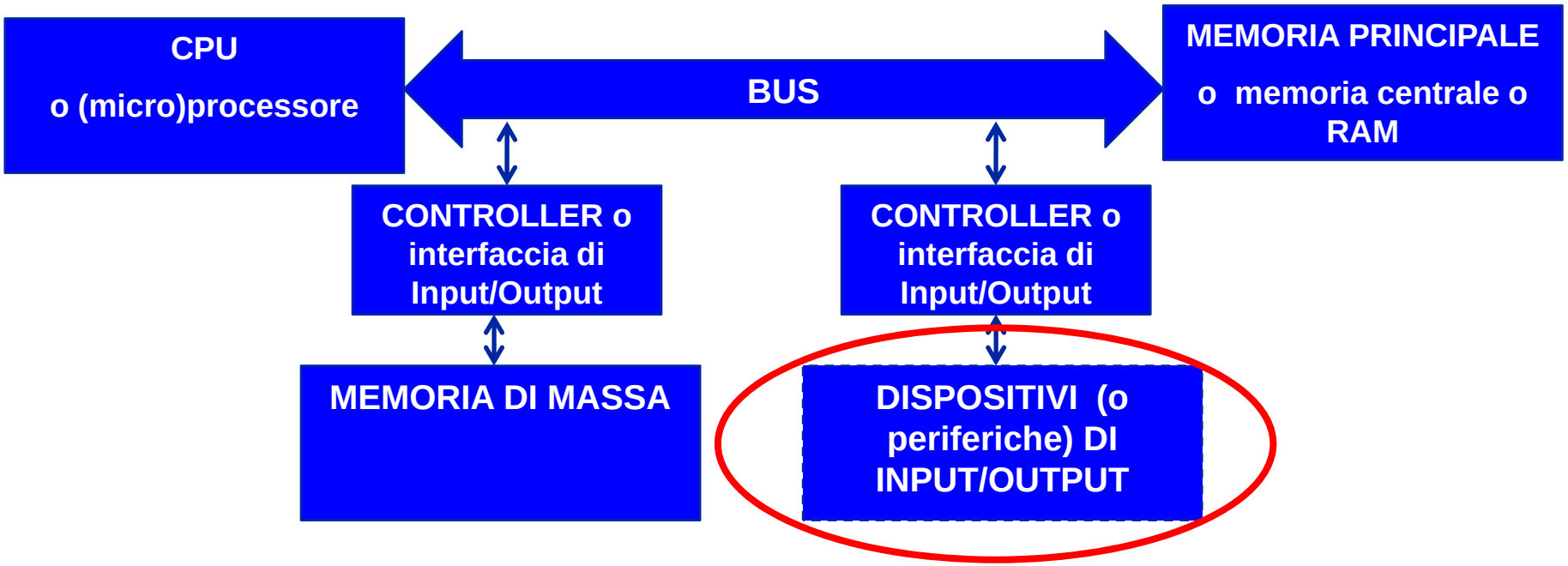
- **Sono attualmente utilizzate in macchine fotografiche digitali, lettori MP3, ecc. e nelle cosiddette “pennine USB” hanno soppiantato i floppy disk come dispositivi per la memorizzazione e lo scambio di dati e programmi**

- **In alcuni portatili sono utilizzati come principale dispositivo di memoria di massa (negli SSD, Solid State Drive o Unità allo stato solidi): sono più veloci e meno soggetti ai guasti dei classici hard disk, ma hanno un costo maggiore, una durata che può essere inferiore, a causa dell’usura prodotta dalle operazioni di scrittura (problemi che sembrano però destinati a soluzione...)**

MEMORIA DI MASSA (21)
[Tecnologie elettroniche: memorie flash]

Immagini tratte da Wikimedia Commons

DISPOSITIVI DI INPUT/OUTPUT (1)



DISPOSITIVI DI INPUT/OUTPUT (2)

- **Consentono l'interazione del calcolatore con l'ambiente esterno**

- **I dispositivi di input consentono l'immissione di informazioni nel calcolatore. Tipici dispositivi di input:**
  1. tastiera
  2. dispositivi di puntamento (mouse, touchpad, pointing stick, ...)
  3. tavolette grafiche
  4. scanner
  5. ...

- **I dispositivi di output consentono la comunicazione di informazioni dal calcolatore verso l'ambiente esterno. Tipici dispositivi di output:**
  1. video
  2. stampante
  3. plotter
  4. altoparlanti
  5. ...

- **Operano in maniera asincrona rispetto al processore, ma devono esservi momenti di sincronizzazione**

DISPOSITIVI DI INPUT/OUTPUT (3)

- **Interrupt: è il principale meccanismo di sincronizzazione fra periferiche e CPU (è mediato dal sistema operativo): mentre una periferica svolge il proprio compito (ad esempio, lettura di informazioni sull'hard disk), il processore può continuare a svolgere i propri (ad esempio esecuzione di istruzioni di un programma utente); quando la periferica necessita di sincronizzarsi con la CPU (ad esempio, quando le informazioni necessarie sono state lette dall'hard disk e sono disponibili per il processore), questa invia sul bus al processore un segnale di interrupt; quando riceve un tale segnale, il processore interrompe quello che stava facendo (ad esempio, l'esecuzione del programma utente) e passa ad eseguire un insieme di istruzioni legate al particolare interrupt che ha ricevuto (ad esempio, istruzioni per il controllo che l'attività della periferica sia terminata senza errori e l'eventuale esecuzione di altre opportune azioni)**

SCHEDA MADRE (1)

- **Molte delle componenti di un calcolatore sono ospitate in circuiti stampati, detti schede**

- **In molti computer attuali, è presente una scheda principale, detta *scheda madre* (in inglese, *motherboard* o *mainboard*) che contiene (fra le altre cose) gli *alloggiamenti per la CPU* e *per i moduli di memoria principale*, i *bus*, vari *controller* in essa integrati, il *firmware* contenente le istruzioni che devono essere eseguite all'avvio del calcolatore, una *piccola batteria* a lunga durata, ecc.**

SCHEDA MADRE (2)

Immagine tratta da http://www.flickr.com/photos, inserita da hekman2007.

L'autore dell'immagine non ha alcuna responsabilità sull'uso di tale immagine in questa presentazione.

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (1)

- **Abbiamo presentato la macchina di von Neumann nelle sue componenti fondamentali**
- **I moderni calcolatori presentano alcune differenze rispetto all'architettura hardware finora descritta**
- **Tali differenze derivano principalmente da modifiche introdotte per ragioni di efficienza**
- **I principi basilari dell'architettura dei calcolatori fin qui descritti restano comunque validi**

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (2)

Per completezza, accenniamo ad alcune delle principali differenze fra l'architettura hardware di base descritta sopra e quella di alcuni moderni calcolatori:

1. **Memoria cache: abbiamo descritto un'architettura in cui vi sono tre tipologie di memorie, che possiamo considerare situate a tre diversi livelli per quanto riguarda la velocità (livelli caratterizzati anche dal costo per quantità unitaria di memoria e dalla capacità):**
   a) i registri del processore (memoria volatile): possiamo considerarli come il livello più alto. Costituiscono la memoria più veloce, più costosa e meno capiente di un calcolatore
   b) la memoria principale (memoria volatile): si situa a livello intermedio per velocità (più lenta dei registri, ma più veloce dei dispositivi di memoria di massa), costo (meno costosa dei registri, ma più costosa dei dispositivi di memoria di massa) e capienza (più capiente dei registri, ma meno della memoria di massa)

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (3)

c) la memoria di massa (non volatile): si situa al livello più basso per quanto riguarda la velocità. E' la meno costosa e la più capiente

- **La memoria principale, anche se molto veloce, opera con tempi superiori a quelli della CPU. I registri del processore sono velocissimi, ma anche molto costosi. Per sfruttare il più possibile la capacità dei moderni processori di lavorare a velocità elevatissime, pur contenendo i costi dell'intero sistema, nei moderni calcolatori sono stati introdotti uno o due ulteriori livelli di memoria (volatile) tra i registri del processore e la memoria principale**

- **Le memorie situate a questi ulteriori livelli prendono il nome di memorie cache**

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (4)

- **Le memorie cache sono realizzate con una tecnologia che ne rende l'accesso più veloce rispetto a quello alla memoria principale**
- **Sono meno costose (ma anche meno veloci!) dei registri del processore**
- **Sono più capienti dei registri, ma meno della memoria principale**
- **Le memorie cache sono pensate per contenere le informazioni che hanno la maggior probabilità di essere utilizzate nell'immediato dalla CPU (e un apposito meccanismo assicura la copia in cache di tali informazioni)**
- **Quando un'informazione deve essere recuperata dalla memoria principale, la richiesta di tale informazione viene fatta anche alle memorie cache, in modo che l'informazione, se presente in cache, possa essere recuperata più velocemente. Se l'informazione non è presente in cache, essa potrà comunque essere recuperata dalla memoria principale (se è presente in questa)**
- **Può anche accadere che un'informazione cercata non sia presente neanche in memoria principale: in tal caso, la ricerca prosegue nel livello più basso della memoria di massa (come vedremo parlando di sistema operativo e di memoria virtuale, un esempio di questa situazione è il "page fault")**

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (5)

2. **Parallelizzazione nell'esecuzione di istruzioni macchina: nel descrivere il ciclo *di fetch-decode-execute*, abbiamo tacitamente assunto che la CPU completi il ciclo per l'istruzione corrente prima di passare al quella successiva. Nei moderni calcolatori non è esattamente così e solitamente la CPU è in grado di "gestire" più istruzioni contemporaneamente. A esempio, siccome l'esecuzione di un'istruzione impegna circuiti diversi da quelli necessari alla lettura delle istruzioni, è possibile che la CPU recuperi un'istruzione successiva (o presunta tale) mentre quella corrente è ancora in fase di esecuzione: in questo modo, la fase di *execute* dell'istruzione corrente è eseguita in parallelo a quella di *fetch* di un'istruzione (molto probabilmente) successiva.[1]**

1 Il parallelismo a livello di esecuzione di istruzioni macchina è generalmente molto più sofisticato di quello accennato qui, ma una sua descrizione dettagliata esula dagli scopi del corso

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (6)

3. **Architetture parallele: esistono in commercio sistemi in cui il parallelismo è realizzato non solo a livello di singolo processore, ma anche duplicando alcune componenti hardware. Esempio rappresentativo sono i sistemi con più CPU, in grado di operare in parallelo. In tali sistemi, le CPU possono condividere una memoria principale (oltre a disporre ciascuna di un'eventuale memoria "principale" privata), oppure possono disporre ciascuna solo di una memoria "principale" privata, in assenza di una memoria principale condivisa**

ALCUNE EVOLUZIONI DELLA MACCHINA DI VON NEUMANN (7)

4. **Maggiore strutturazione del sistema di bus: abbiamo presentato un'architettura in cui CPU, memoria principale e periferiche condividono uno stesso insieme di bus (bus dati, bus indirizzi e bus di controllo). In un tale meccanismo l'insieme condiviso di bus potrebbe diventare il collo di bottiglia dell'intero sistema (nel senso che la condivisione di uno stesso canale di comunicazione da parte di molte componenti può facilmente congestionarsi e rallentare il funzionamento dell'intero sistema) e, inoltre, non tiene conto dei diversi tempi di funzionamento delle varie tipologie di componenti. Per ovviare a questi inconvenienti, nei moderni calcolatori è presente un sistema di bus più sofisticato di quello discusso qui, ma la cui descrizione esula dagli scopi del corso**