Architettura dei calcolatori

Vittorio Zaccaria

November 8, 2018

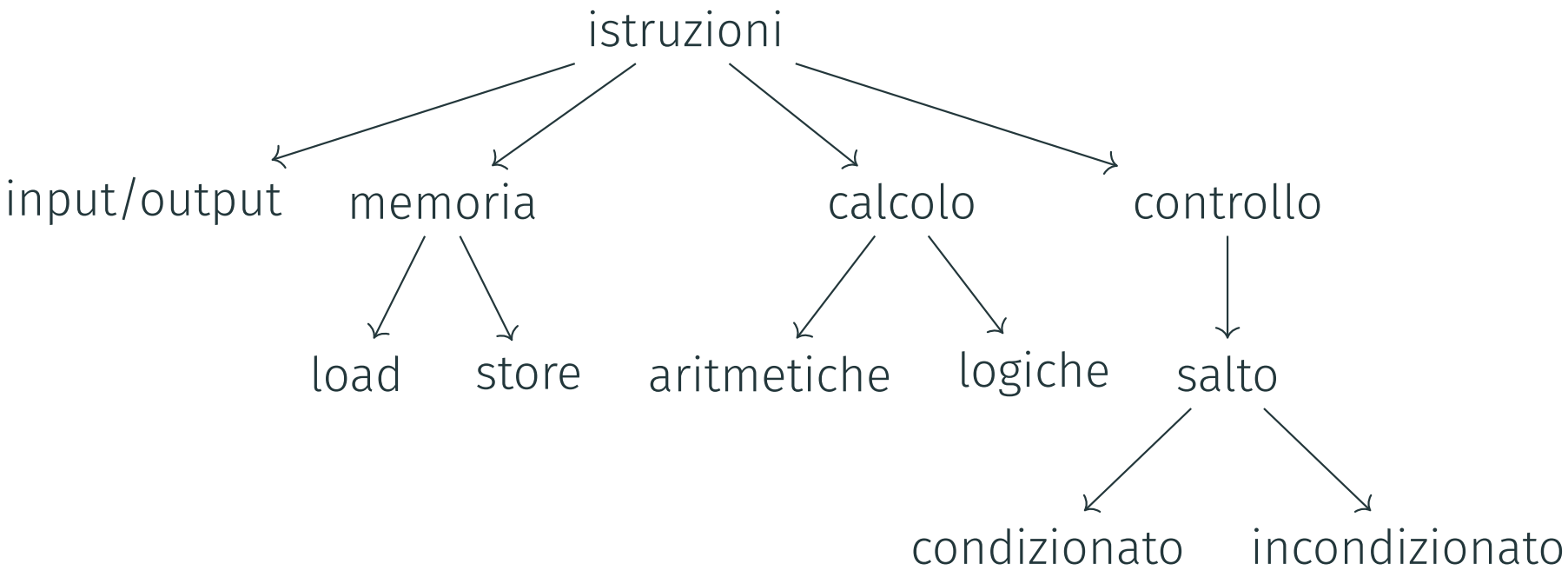
Outline

Architetture dei calcolatori

Ruolo della CPU

- La central processing unit (CPU) esegue una istruzione alla volta.
- Le istruzioni sono posizionate in memoria principale, e devono essere caricate dalla CPU che quindi le decodifica e le esegue.
- Un'istruzione ha quindi una codifica equivalente ad un numero binario (possiamo immaginare che la sua codifica sia l'indice all'interno della tabella delle istruzioni).

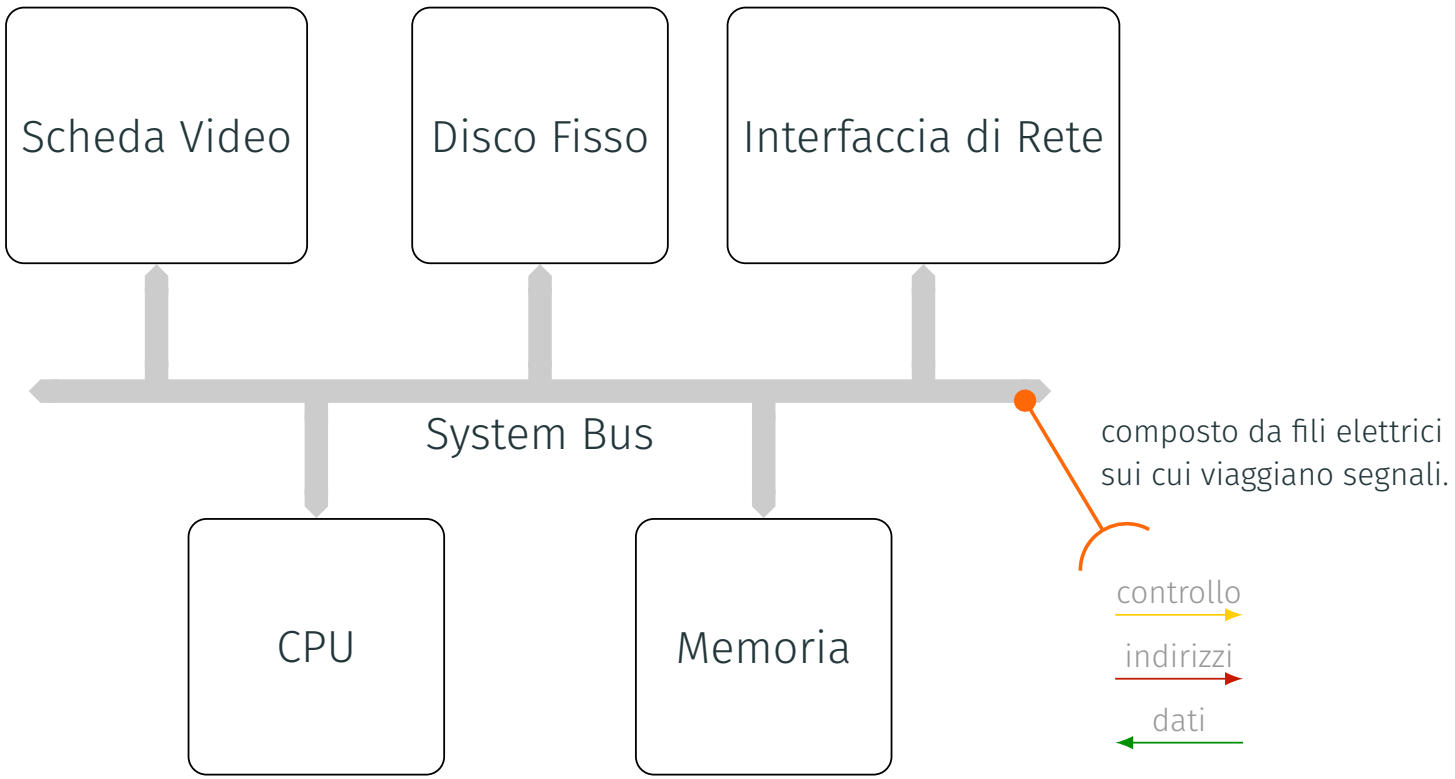
Tassonomia delle istruzioni (semplificata)



Esecuzione istruzioni

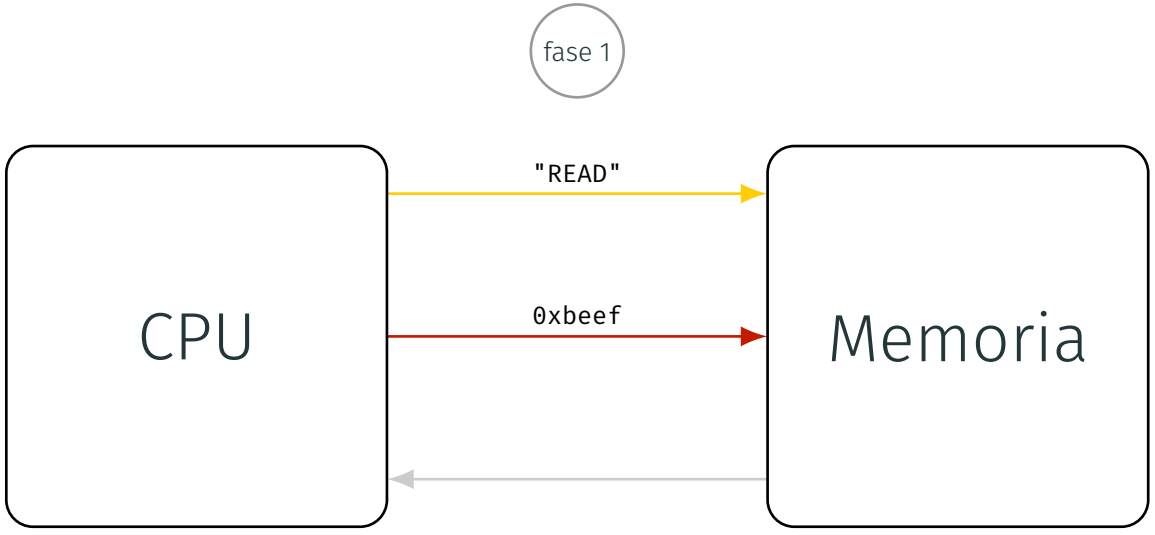
- Da dove il processore fa input/output?
- Da dove il processore fa load/store (carica/salva)?
- Su cosa agiscono le istruzioni aritmetico logiche e i salti?

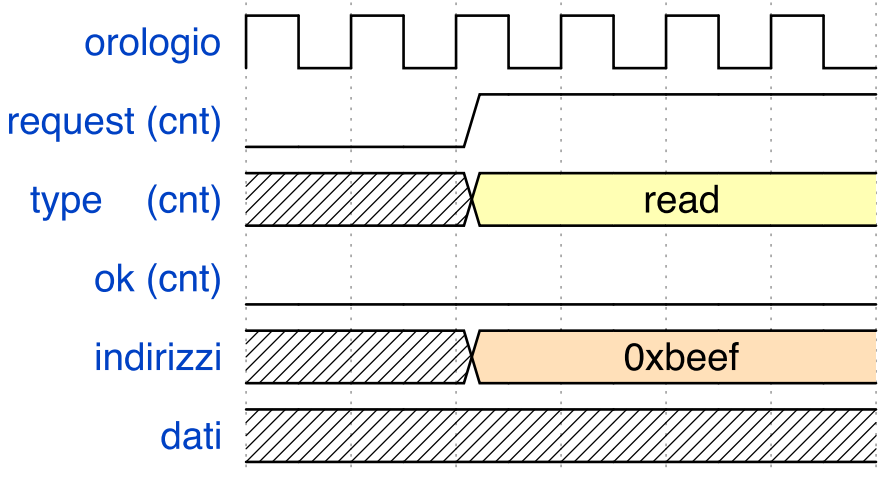
Architettura di Von Neuman



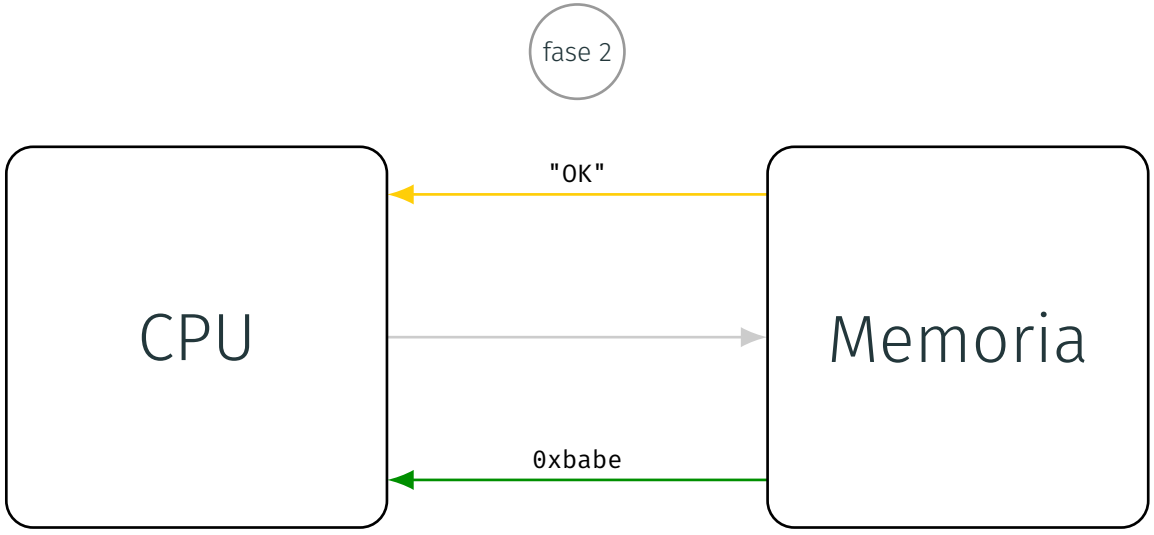
La Memoria Principale

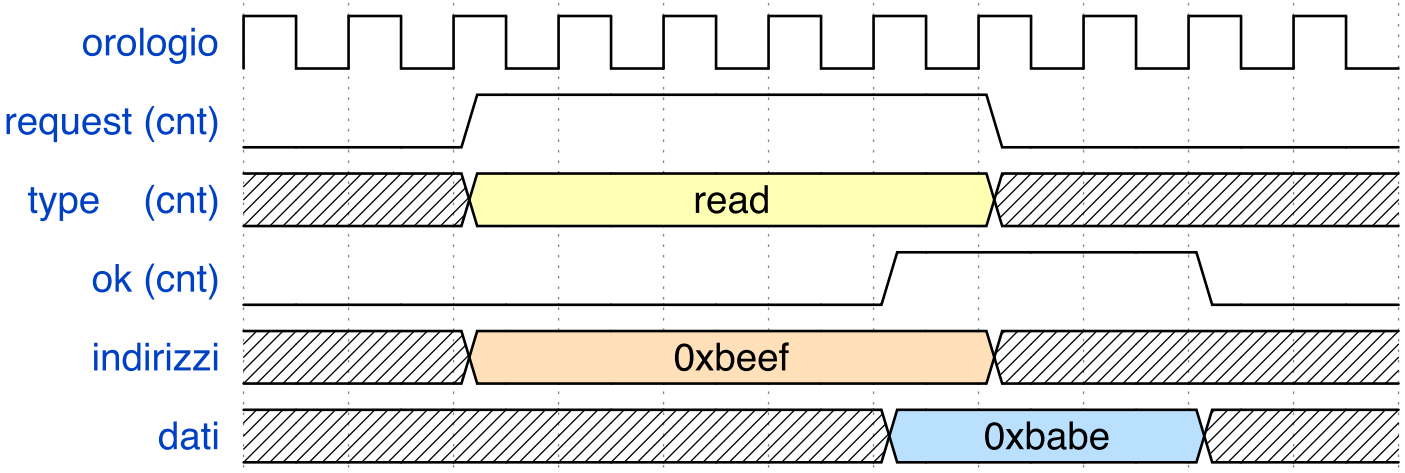
CPU-Memoria Principale/ Lettura



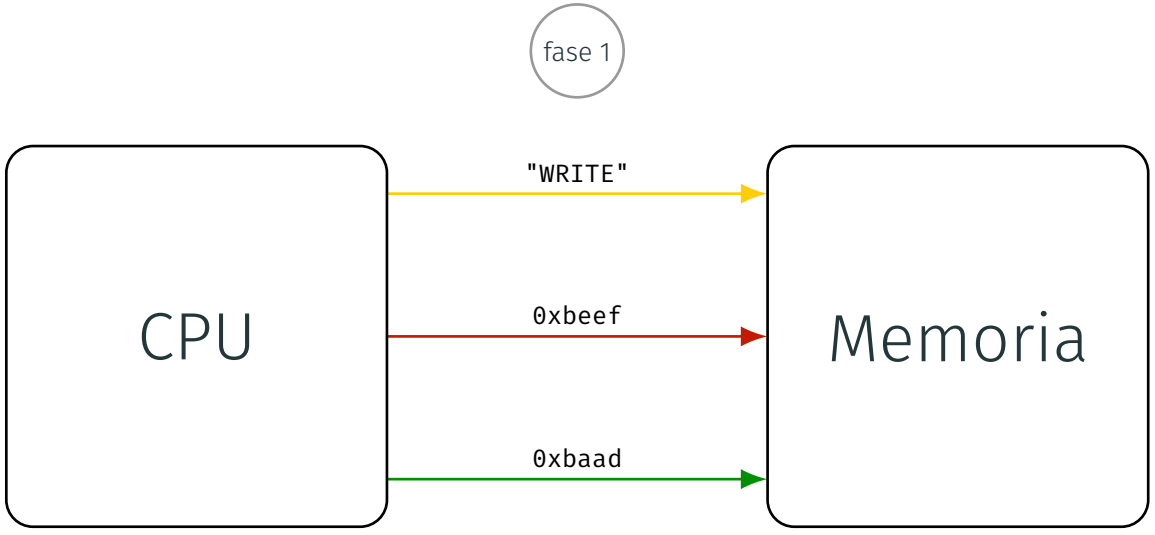


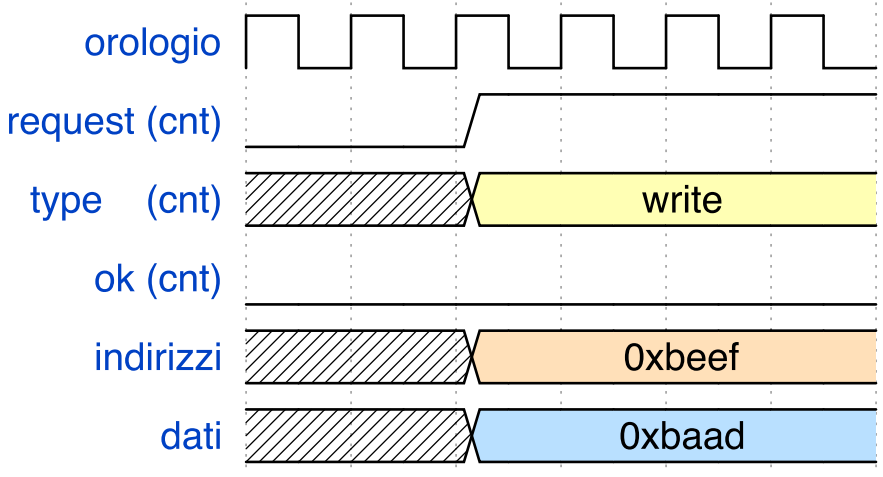
CPU-Memoria Principale/ Lettura



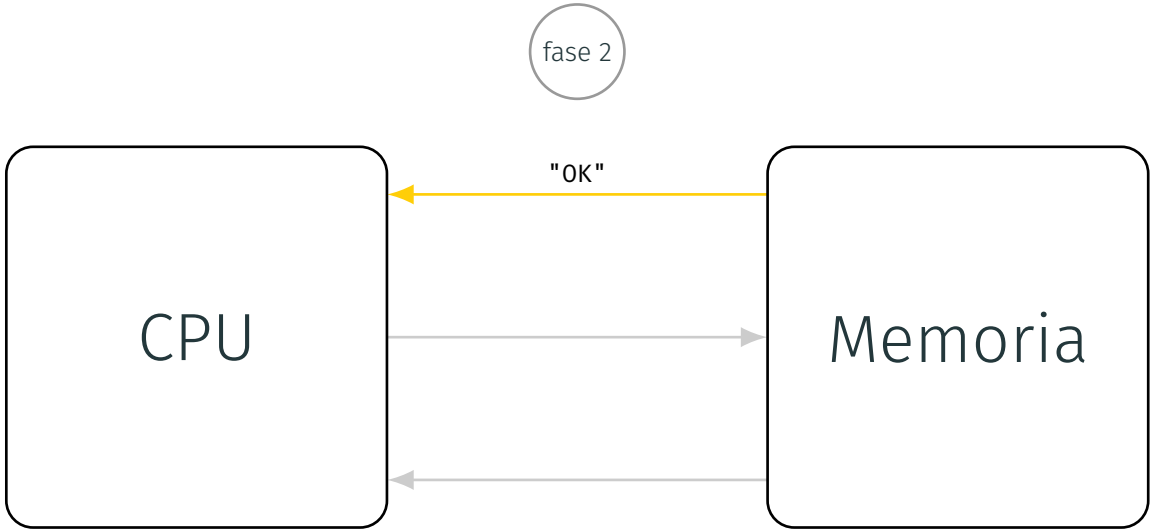


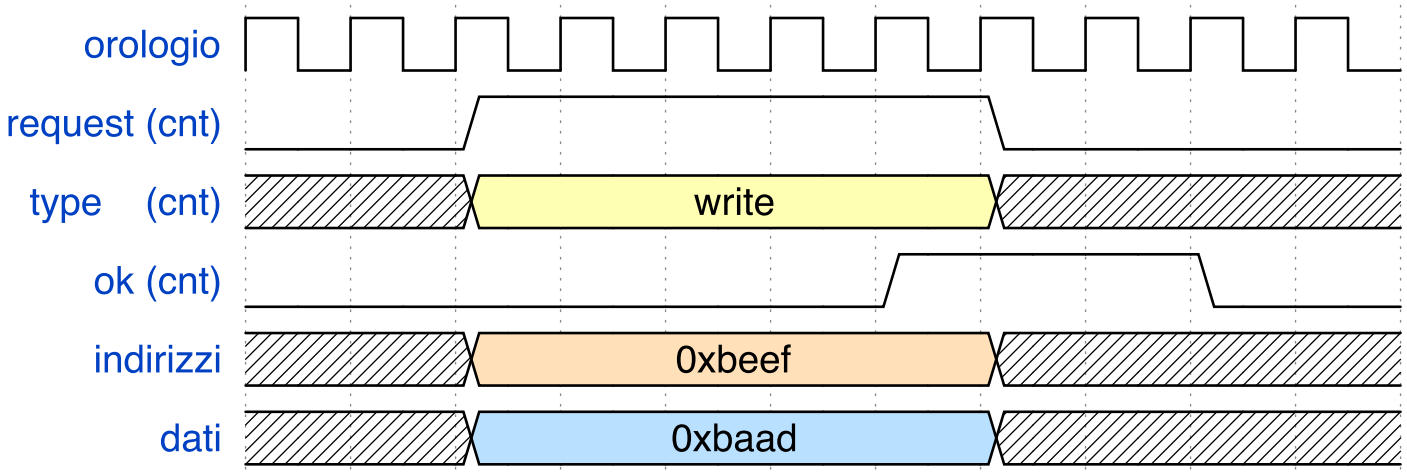
CPU-Memoria Principale/ Scrittura



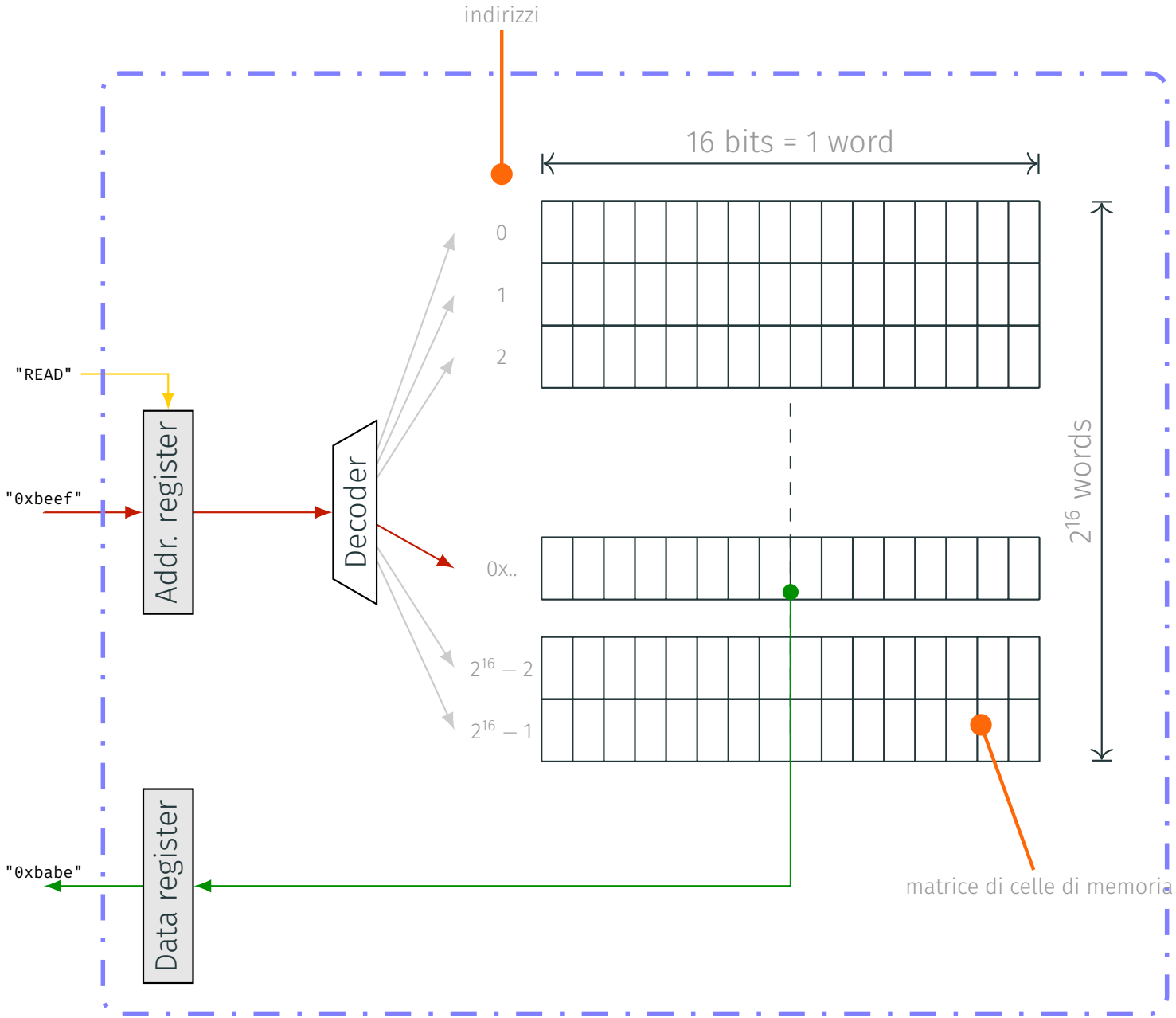


CPU-Memoria Principale/ Scrittura



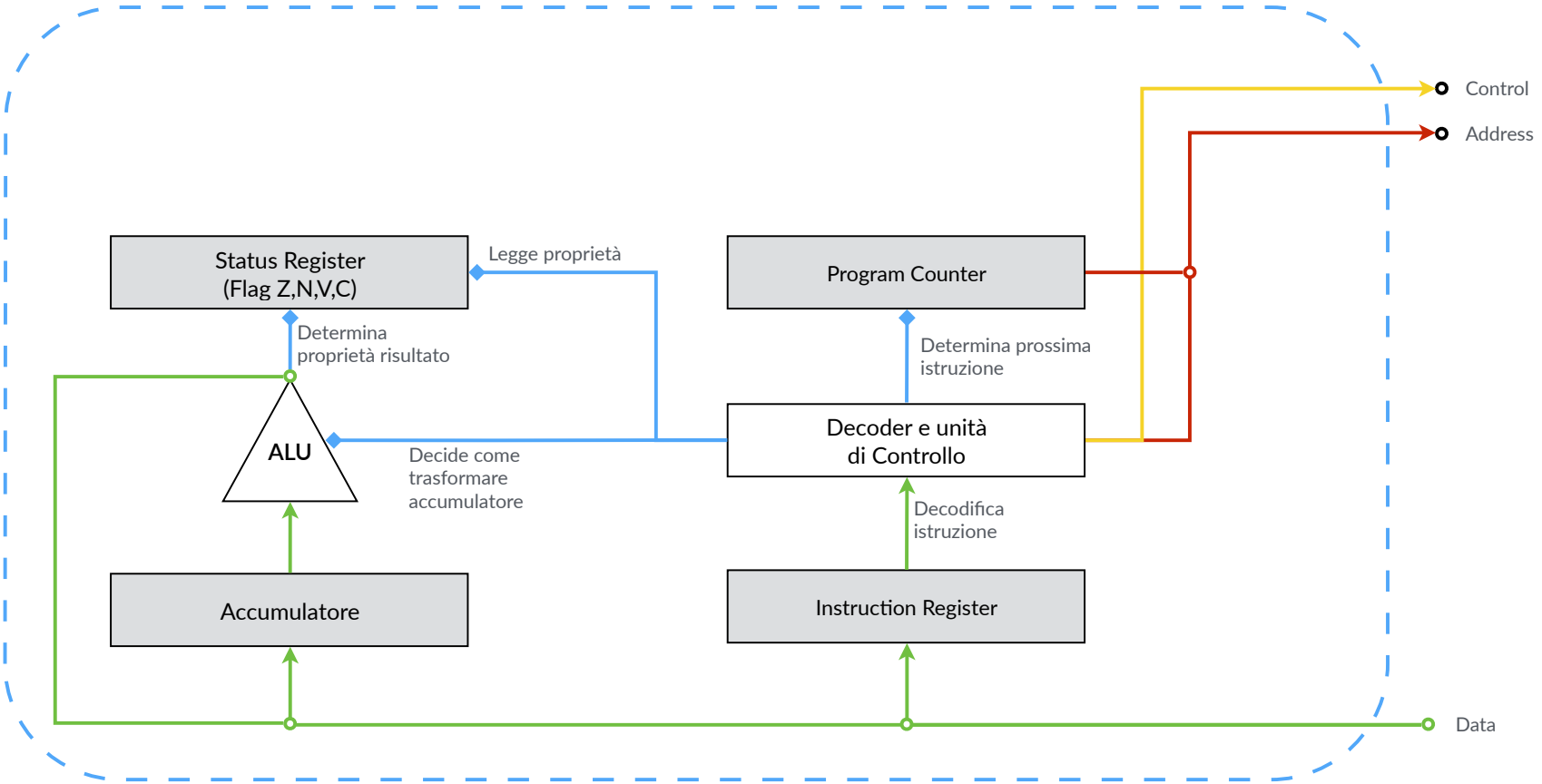


Interno di una memoria principale



La CPU

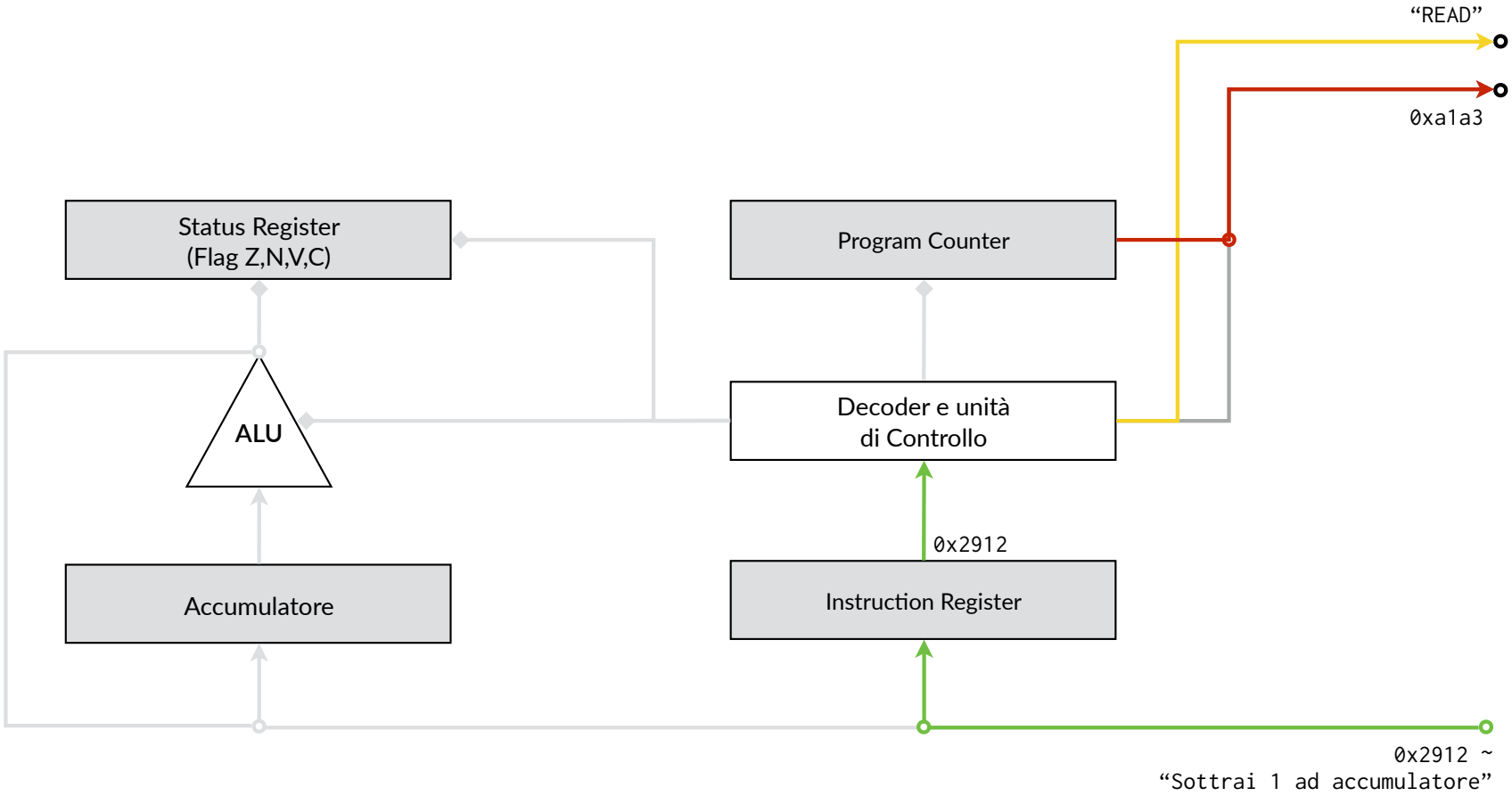
Architettura (Semplificata)



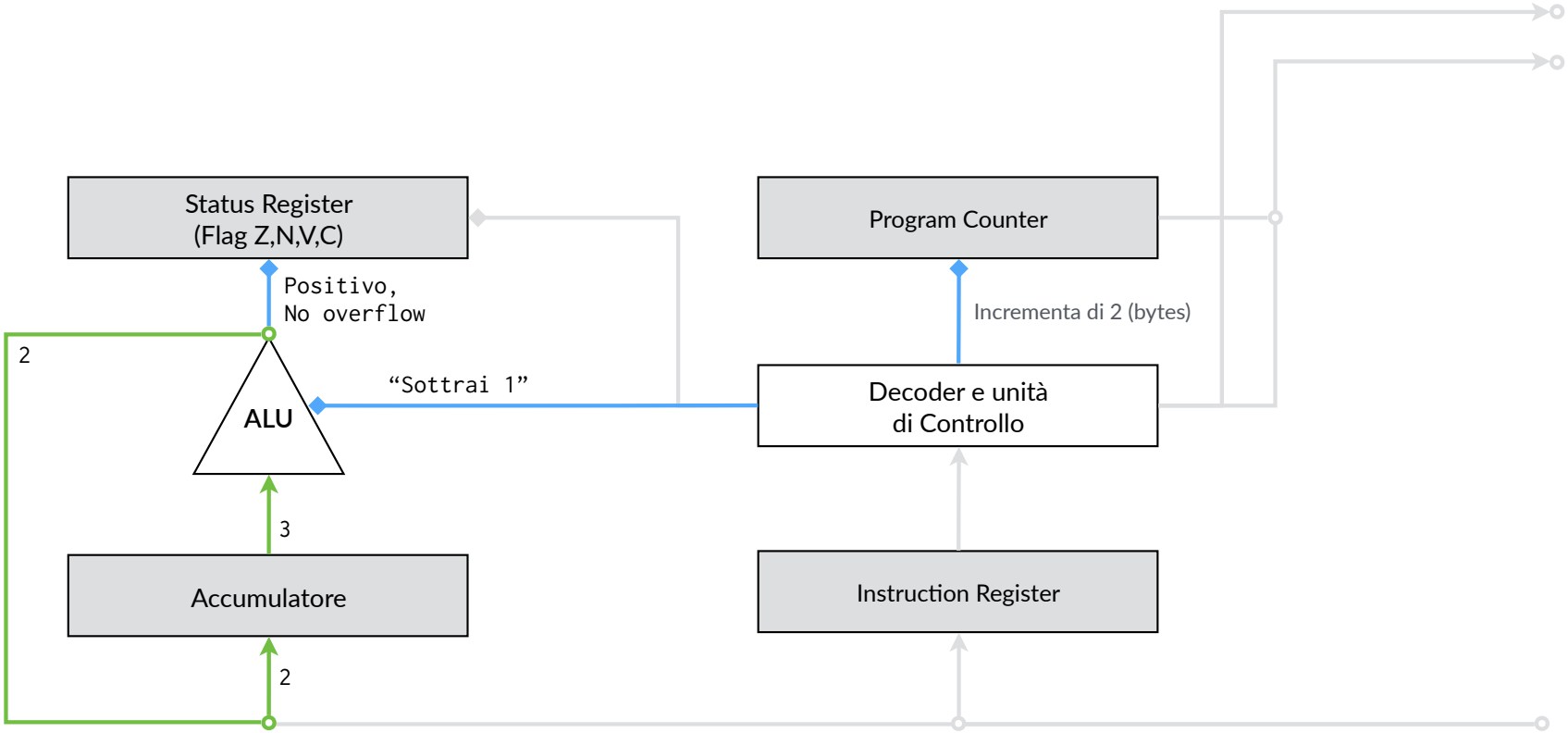
Ciclo fondamentale di un'istruzione

1. Leggi e decodifica un'istruzione
2. Esegui l'istruzione e incrementa (o modifica il program counter).

Architettura (Semplificata) - Leggi e decodifica



Architettura (Semplificata) - Esegui



Le Caches

Qualcosa in più sulla memoria principale

- Ogni bit è rappresentato da uno speciale elemento elettronico chiamato 'capacità'.
- E' una specie di secchiello d'acqua che può essere tutto pieno o tutto vuoto.
- Il secchiello però non è perfetto, infatti perde e deve essere regolarmente rabboccato (ciclo di refresh).
- Per tale motivo questa è chiamata Dynamic Random Access Memory (DRAM).

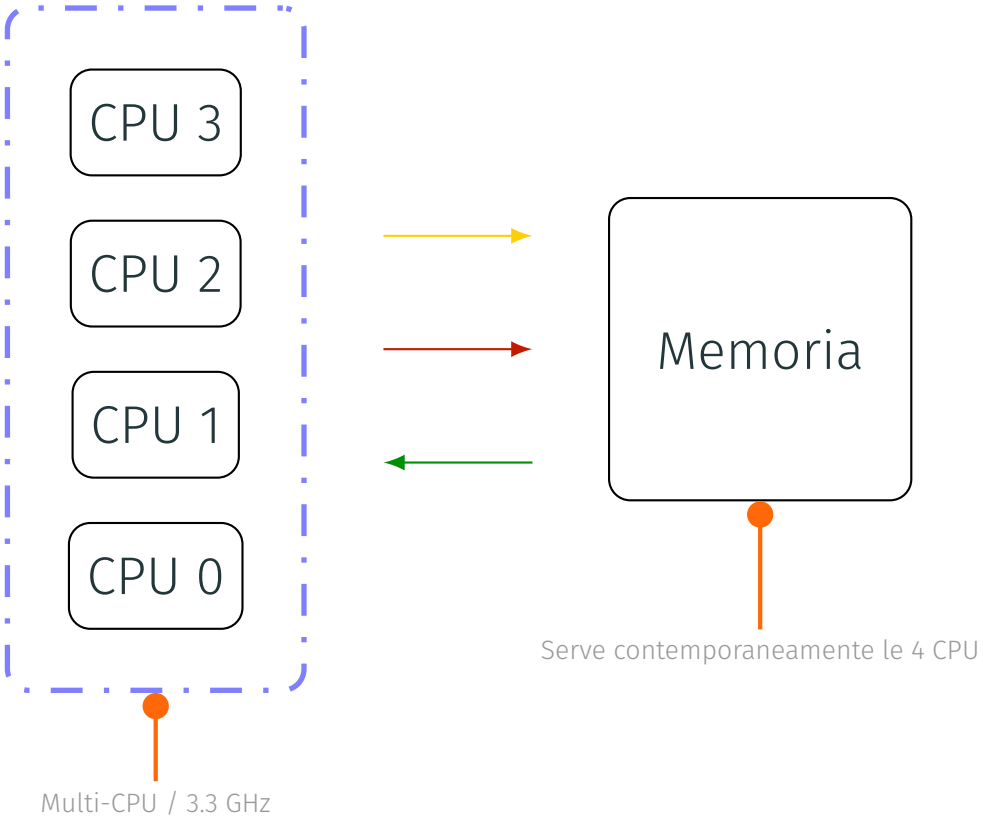
Alcuni dettagli

- Una DRAM costa poco poiché è relativamente semplice da costruire, ma consuma molta potenza elettrica
- La DRAM è evoluta nel tempo:

Caratteristica	FPM DRAM (1998)	DDR4-2400 (2016)
Latenza di accesso	45ns	15ns
Banchi di memoria	1	>1
Clock	no	si
Bus clock	0.022 GHz	1.2 GHz
Velocità di trasferimento	0.17 GB/s	19 GB/s
Eq. scansione di un DVD	29 sec	< 1 sec

Di quanta velocità ha bisogno però la CPU?

Supponiamo di avere una moderna Multi-CPU a 3.3GHz con 4 core. Ogni core esegue calcoli (una istruzione al secondo) su dati a 64 bit, ma tutti condividono la stessa memoria.



Di quanta velocità ha bisogno però la CPU?

La larghezza di banda totale necessaria alla multi-CPU è:

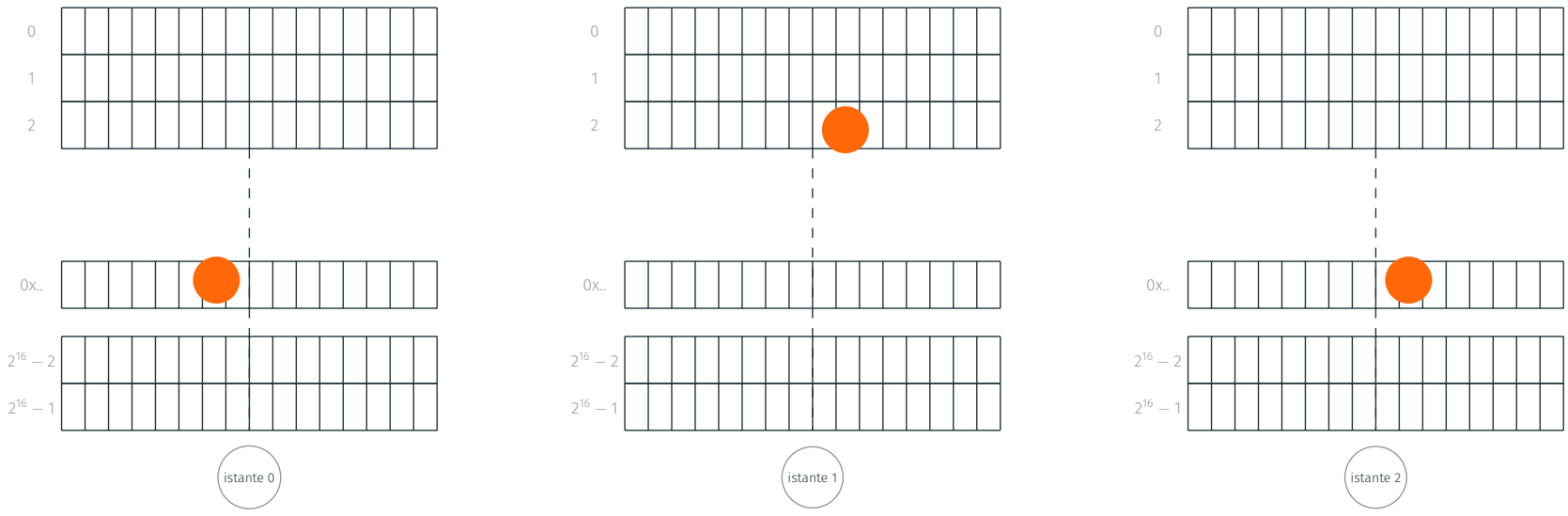
$$3.3 \times 4 \times 8 = 105 \quad GB/s \tag{1}$$

Molto superiore a 19 GB/s (DDR4). Conclusione: la memoria principale, da sola, non è adeguata alle velocità richieste dai (multi-)processori moderni

Come risolvere il problema?

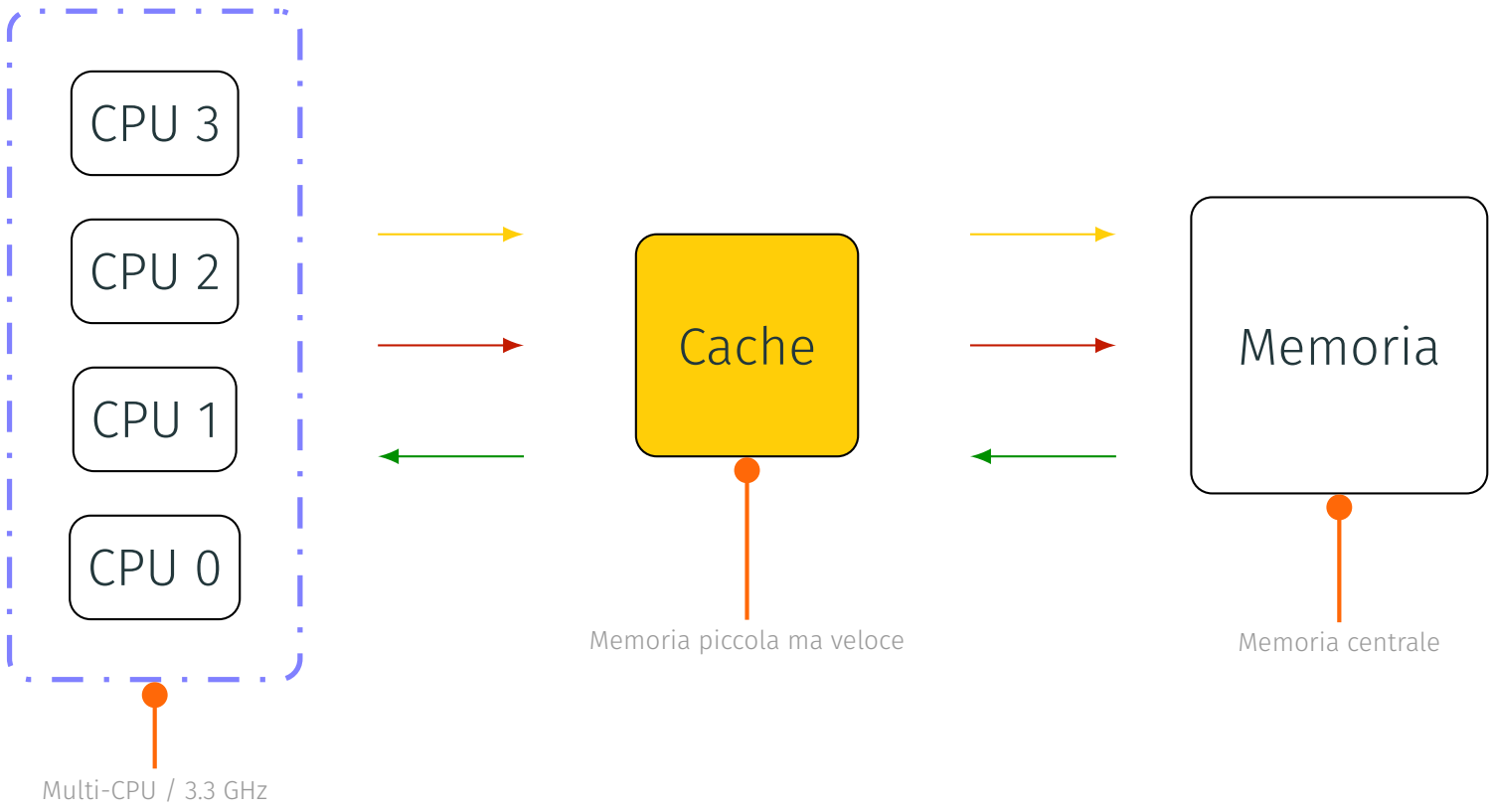
Dobbiamo considerare alcune osservazioni sul comportamento dei programmi nell'accesso alla memoria. Gli accessi seguono il cosiddetto principio di località.

Principio di località



Se un dato viene utilizzato in un dato istante, è probabile che dati posizionati in celle di memoria adiacenti vengano anch'essi richiesti entro breve.

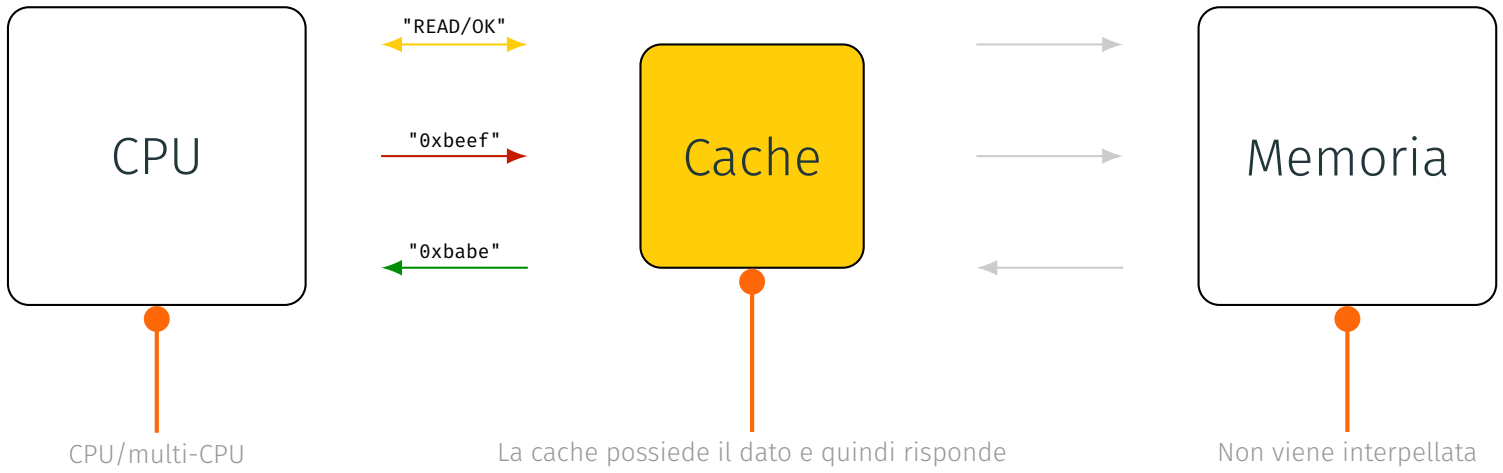
Come sfruttare questo principio?



Possiamo usare una memoria piccola ma veloce (SRAM) che salvi solo i dati più frequentemente usati.

Una cache riesce a servire in maniera adeguata il processore.

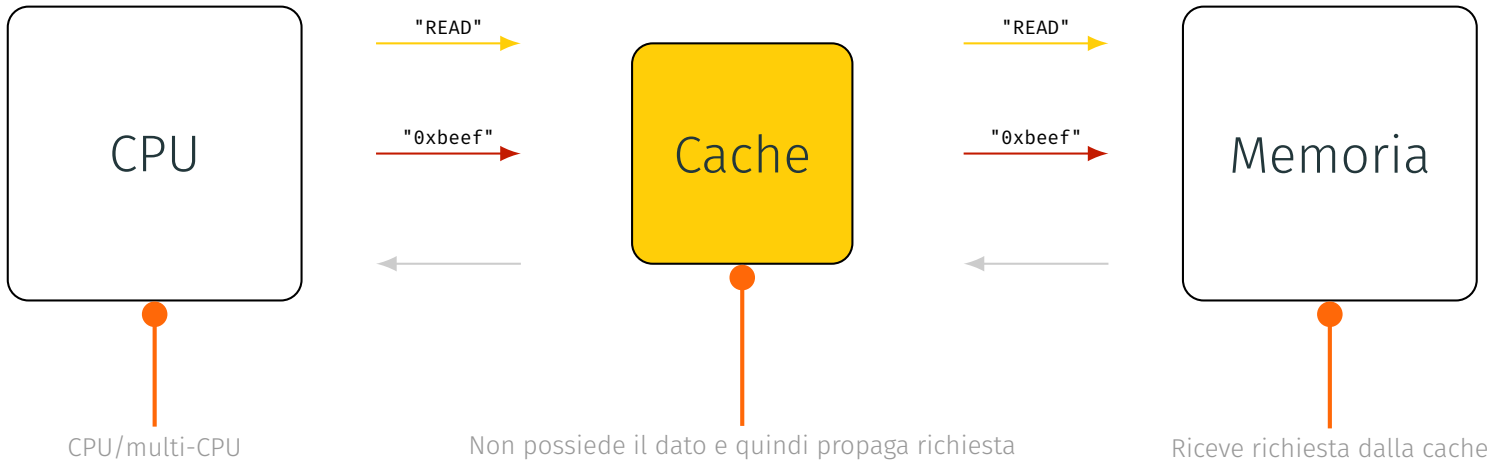
Hit rate in lettura



La CPU chiede il dato richiesto alla cache. Se la cache lo possiede, si ha un hit. In questo caso si parla di:

- Hit time T_h, tempo fra la richiesta e la risposta
- Hit rate R_h, percentuale delle richieste che hanno successo in cache.

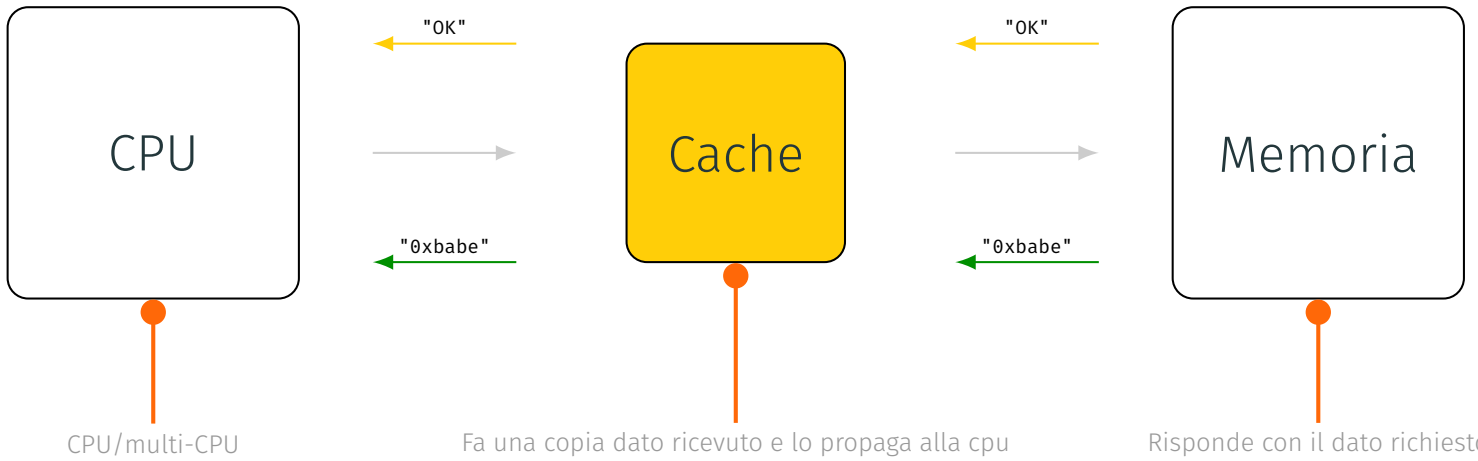
Miss rate in lettura



Se la cache non lo possiede, si parla di miss. I primi passi per risolverlo sono:

1. La richiesta viene propagata alla memoria principale
2. La memoria principale risponde alla cache con il dato

Miss rate in lettura (2)



Successivamente:

1. La cache copia il dato al suo interno (scartandone altri)
2. La cache risponde alla CPU

Il tempo totale è chiamato miss time T_m, mentre il miss rate $R_m = (1 - R_h)$ è la percentuale delle richieste che non hanno successo in cache.

Tempo medio di accesso

Il tempo medio di accesso misura la performance media degli accessi alla memoria. E' calcolato tramite una media pesata delle metriche introdotte precedentemente.

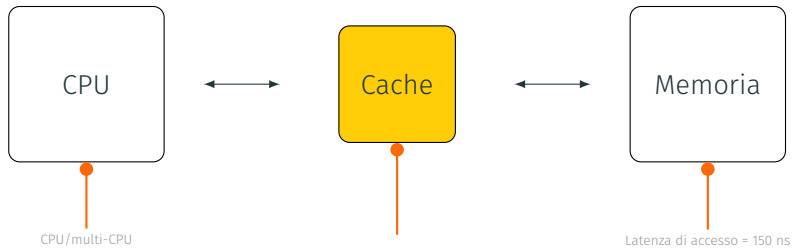
$$T = T_h \times R_h + (1 - R_h) \times T_m \tag{2}$$

Quale dei due sistemi è migliore?

Sistema A:

Provvisto solo di memoria centrale, con latenza di accesso di 150ns.

Sistema B:



Provvisto di cache con i seguenti parametri

R_h	0.7
T_m	300ns
T_h	10ns

Quale dei due sistemi è migliore?

- Tempo medio di accesso di A: 150 ns
- Tempo medio di accesso di B:

$$T = 0.7 \times 10 + (1 - 0.7) \times 300 = 97ns \tag{3}$$

- Vince B (tempo di accesso inferiore)!