



# Architettura degli Elaboratori I - B

## Esercitazione

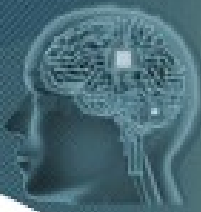
**Processore ARM, Cache e Memoria**

Daniel Riccio/Alberto Aloisio  
Università di Napoli, Federico II

5 aprile 2017



# Le informazioni generali



**Architettura degli Elaboratori I Mod. B. – Prova di valutazione – 03/05/2017**

Candidato:..... Matricola:..... Firma:.....



# Domande a risposta Multipla

## DOMANDE A RISPOSTA MULTIPLA

Si risponda alle seguenti domande riportando sul quadratino al lato di ciascuna opzione una V se si ritiene la risposta corretta ed una F se si ritiene che la risposta sia sbagliata, le risposte lasciate in bianco sono considerate errate (Punti 10):

### Processore ARM

- ☐ Le memorie sono più veloci delle CPU
- ☐ Le memorie sono organizzate in una gerarchia
- ☐ Il posizionamento determina la posizione di una linea in cache
- ☐ Il tempo medio di accesso in cache è dato da  $t = m \cdot t_c + h \cdot t_p$
- ☐ In una cache a indirizzamento diretto le linee di memoria hanno posizioni fisse
- ☐ La cache completamente associative implementano algoritmi di sostituzione
- ☐ La MMU traduce indirizzi virtuali in fisici
- ☐ L'indirizzo fisico è diviso in Directory, Table e Offset
- ☐ La RAM è divisa in pagine
- ☐ La memoria virtuale fa uso di un processo di swapping
- ☐ L'overlay è la sovrapposizione di pezzi di codice in memoria
- ☐ I programmi sono letti dal disco in RAM dal DMA



---

## ESERCIZI

---

**Domanda 1 (Punti 5):** Si consideri una memoria cache set-associativa da 2048 parole, suddivisa in 128 blocchi da 16 locazioni ciascuna, in cui si sceglie di avere 4 blocchi per insieme, ossia 32 insiemi. Sia data la parola 1011010101111101, in quale insieme della cache essa sarà inserita e con quale etichetta?

# Le cache set-associative



Esiste una combinazione delle due tecniche di indirizzamento. I blocchi della cache sono raggruppati in insiemi (*set*), ed il meccanismo di associazione è congegnato in modo che **ciascun blocco della memoria principale è associato ad un particolare insieme della cache ma può risiedere in una qualsiasi posizione all'interno di esso.**

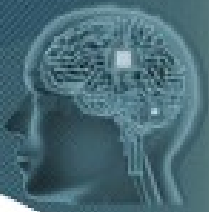
**Avere più possibilità di posizionamento di un blocco in un insieme riduce la possibilità di dover attivare meccanismi di sostituzione di blocchi nella cache, anche quando essa non è completa, e riduce il numero di etichette tra cui effettuare la ricerca associativa.**

# Le cache set-associative



Esiste una combinazione delle due tecniche di indirizzamento. I blocchi della cache sono raggruppati in insiemi (*set*), ed il meccanismo di associazione è congegnato in modo che **ciascun blocco della memoria principale è associato ad un particolare insieme della cache ma può risiedere in una qualsiasi posizione all'interno di esso.**

**Avere più possibilità di posizionamento di un blocco in un insieme riduce la possibilità di dover attivare meccanismi di sostituzione di blocchi nella cache, anche quando essa non è completa, e riduce il numero di etichette tra cui effettuare la ricerca associativa.**



# Le cache set-associative (continua)

Con la solita cache da **2048 parole**, suddivisa in **128 blocchi** da **16 locazioni** ciascuna, se si sceglie di avere **4 blocchi per insieme**, si hanno **32 insiemi**.

Se la memoria principale ha i soliti **4096 blocchi** (64k locazioni) essi sono individuabili attraverso i **12 bit** più significativi di indirizzo, comuni alle 16 parole di ciascun blocco.

Ad ogni insieme della cache corrispondono, allora, **128 blocchi** della memoria principale ma solo fino ad **un massimo di 4** di essi possono trovare effettivamente posto contemporaneamente nella cache.



# L'algoritmo di posizionamento ( 1 / 2 )

- I 128 blocchi della memoria 0, 32, 64, 96, 128, ..., 4064 puntano all'insieme 0 della cache, in una qualsiasi delle quattro posizioni.
- I blocchi 1, 33, 65, 97, 129, ... 4065, sono associati all'insieme 1 della cache e così via per gli insiemi 2, 3 4 e fino a quello 31.
- Quando un blocco viene caricato nella cache, i 5 bit meno significativi dei 12 bit dell'indirizzo individuano l'insieme della cache dove esso deve essere posizionato; in un blocco vuoto.
- I successivi 7 bit più significativi dell'indirizzo vengono trascritti nel registro della cache associato al blocco, e le 16 parole di programma memorizzate nelle locazioni.



# L'algoritmo di posizionamento ( 2 / 2 )

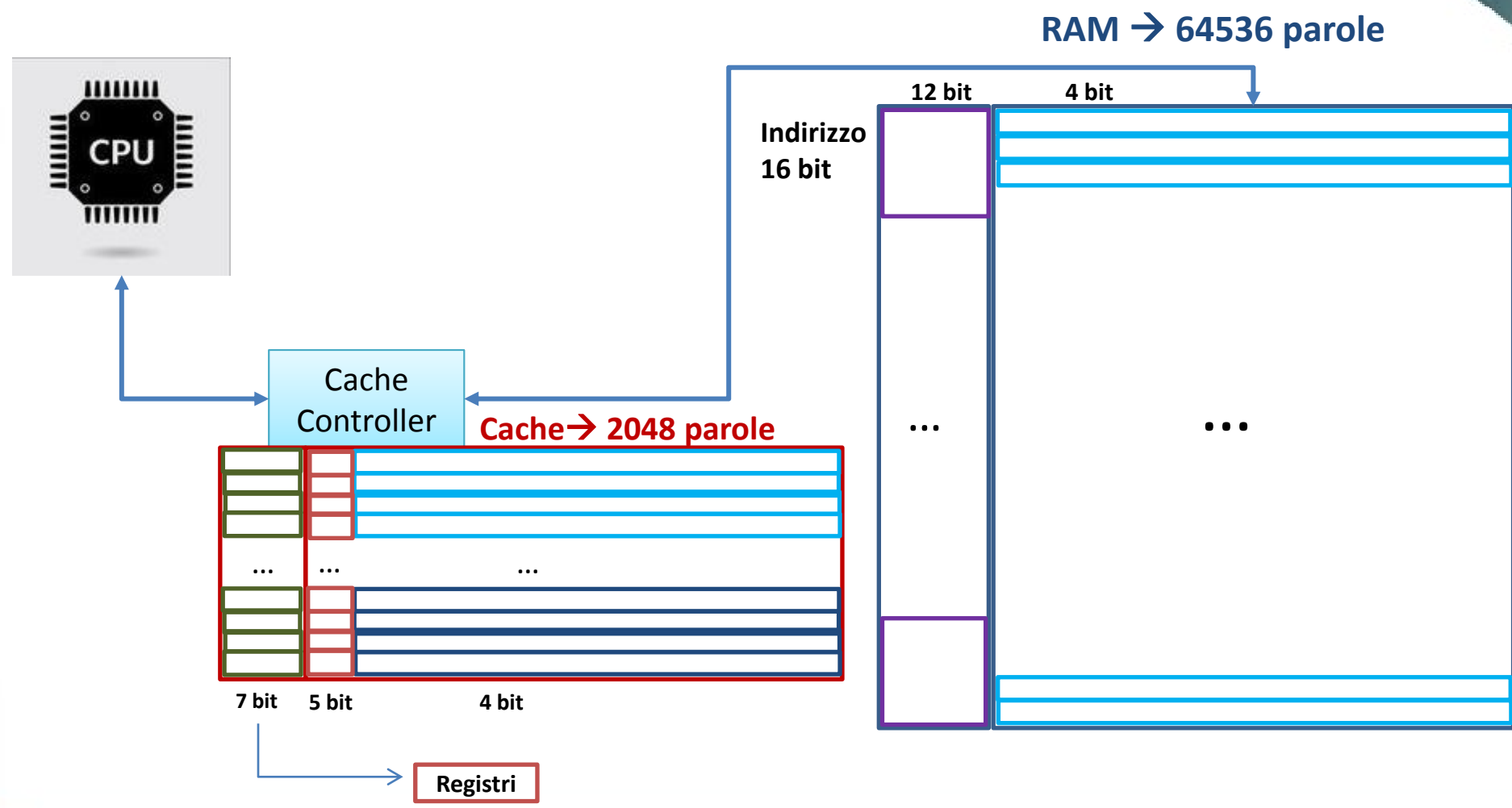


1. Quando si effettua un accesso, i 4 bit meno significativi dell'indirizzo sono inizialmente ignorati, serviranno successivamente per individuare su quale delle 16 parole del blocco si deve operare.
2. I successivi 5 bit (*campo set*) indicano l'insieme dove cercare, mentre i 7 bit più significativi indicano il blocco cercato.
3. Questi bit debbono, quindi, essere comparati associativamente con le etichette dei quattro blocchi dell'insieme, per controllare se il blocco desiderato è presente o meno nell'insieme.

Questa ricerca associativa su solo 4 etichette è certamente molto più semplice e più veloce che in una memoria completamente associativa.



# Nelle puntate precedenti...





# Qualche esempio numerico

Se, per esempio, si vuole trasferire il blocco 2323 della memoria principale, il suo indirizzo è: 100100010011XXXX. Esso andrà nell'insieme 19 della cache con l'etichetta 72.

Viceversa, se si vuole accedere alla locazione 32034 (810A<sup>hex</sup>) che ha indirizzo 1000000100001010<sup>b</sup>, si ignorano inizialmente i 4 bit meno significativi e si va ad esaminare il contenuto dei 4 blocchi dell'insieme 16 della cache, dove potremmo trovare, per esempio, il blocco 4072, il blocco 16 e come terzo blocco il 2064, cui appartiene la locazione cercata.

Il blocco 2064 sarà individuato attraverso una ricerca associativa sulle etichette dei 4 blocchi.



# Posizionamento in cache

**Domanda 1 (Punti 5):** Si consideri una memoria cache set-associativa da **2048** parole, suddivisa in **128** blocchi da **16** locazioni ciascuna, in cui si sceglie di avere **4** blocchi per insieme, ossia **32** insiemi. Sia data la parola **1011010 101111101**, in quale insieme della cache essa sarà inserita e con quale etichetta?

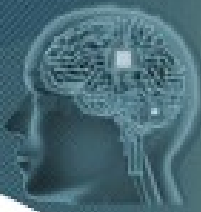
Parola in memoria:

**1011010 101111101**

Parola nel blocco = **1101** = **13**

Insieme = **10111** = **23**

Etichetta = **1011010** = **90**



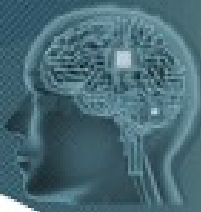
# Posizionamento in cache

**Domanda 2 (Punti 5):** Si consideri una memoria cache set-associativa da **2048** parole, suddivisa in **128** blocchi da **16** locazioni ciascuna, in cui si sceglie di avere **4** blocchi per insieme, ossia **32** insiemi. Sia data la parola **1001000 10011 1101**, in quale insieme della cache essa sarà inserita e con quale etichetta? Analogamente, quali sono insieme ed etichetta se i blocchi per insieme fossero **2**?

1001000 10011 **1101** → Parola nella linea cache = 13

1001000 **10011** → Insieme nella cache = 19

**1001000** → Etichetta = 72



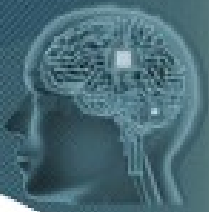
# Posizionamento in cache

**Domanda 2 (Punti 5):** Si consideri una memoria cache set-associativa da **2048** parole, suddivisa in **64** blocchi da **32** locazioni ciascuna, in cui si sceglie di avere **2** blocchi per insieme, ossia **32** insiemi. Sia data la parola **1101001100111111**, in quale insieme della cache essa sarà inserita e con quale etichetta?

11010011001**1111** → Parola nella linea cache = 31

100100**010011** → Insieme nella cache = 19

**100100** → Etichetta = 36



## Incremento di prestazioni con la cache (1/3)

Valutiamo ora l'impatto della cache sulle prestazioni complessive del sistema di calcolo. Sia  $h$  la frequenza di successo,  $M$  la penalità di fallimento, ossia il tempo necessario per accedere alle informazioni nella memoria principale, e  $C$  il tempo per accedere alle informazioni nella memoria cache. Il tempo medio di accesso sperimentato dalla CPU è

$$t_{\text{ave}} = h C + (1 - h) M$$

Consideriamo un *processore veloce* senza cache e con *memoria principale* DRAM, per il quale sono necessari 10 cicli di clock per ogni accesso in lettura alla memoria.

Si supponga ora che allo stesso sistema sia aggiunta una memoria cache che contiene blocchi di otto parole e che la memoria principale sia interallacciata. In questo caso, supponiamo che siano necessari 17 cicli per caricare un blocco nella cache.



## Incremento di prestazioni con la cache (2/3)

Si ipotizzi che il 30% delle istruzioni in un programma tipico effettui un'operazione di scrittura o di lettura, e che la frequenza di successo sia il 95% per le istruzioni e il 90% per i dati.

Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura.

Con le ipotesi fatte, i tempi nei due casi sono:

$$\frac{\text{Senza - cache}}{\text{Con - cache}} = \frac{100 \times 10}{70(0,95 \times 1 + 0,05 \times 17) + 30(0,9 \times 1 + 0,1 \times 17)} = 4,90$$





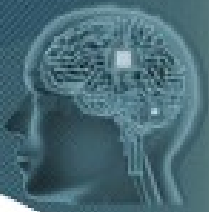
## Incremento di prestazioni con la cache (3/3)

Possiamo anche valutare quanto sia efficace la memoria cache con le prestazioni ipotizzate se confrontata con una cache ideale con una frequenza di successo pari al 100% (in tal caso, tutti gli accessi alla memoria richiedono un solo ciclo di clock).

$$\frac{70(0,95 \times 1 + 0,05 \times 17) + 30(0,9 \times 1 + 0,1 \times 17)}{100} = 2,04$$

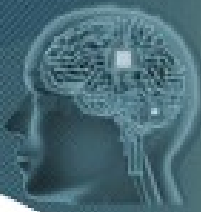
In pratica la cache ipotizzata permette alla CPU di funzionare come se avesse una memoria principale, basata su DRAM, con tempi d'accesso che sono solo il doppio di quelli della cache.

Nell'esempio abbiamo supposto la frequenza di successo diversa per istruzioni e dati. Sebbene frequenze di successo con valori superiori a 0.9 siano ottenibili per entrambi, la frequenza di successo per le istruzioni è di solito maggiore di quella per i dati. Questi valori dipendono dal progetto della cache e dal profilo di accesso a istruzioni e dati che caratterizza il programma in esecuzione.



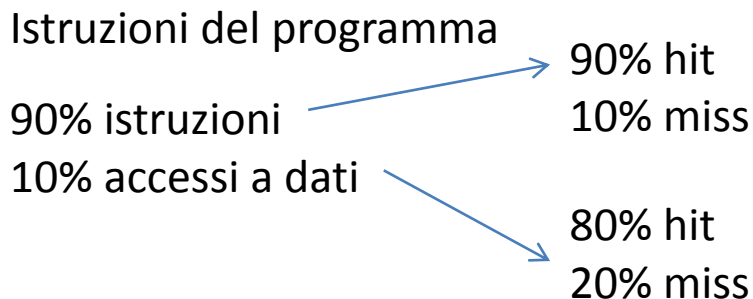
# Esercizi sulle cache

**Domanda 3 (Punti 10):** Si ipotizzi che il 10% delle istruzioni in un programma tipico effettui un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache sia del 90% per le istruzioni e del 80% per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e una sia in cache che in memoria costi 16. Quale sarebbe il guadagno utilizzando la cache?



# Esercizi sulle cache

**Domanda 3 (Punti 10):** Si ipotizzi che il 10% delle istruzioni in un programma tipico effettui un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache sia del 90% per le istruzioni e del 80% per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e una sia in cache che in memoria costi 16. Quale sarebbe il guadagno utilizzando la cache?



Accesso senza cache:  $100 \times 10$

Accesso con cache: costo istruzioni + costo dati

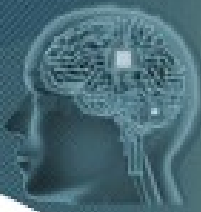
$90(0.9 \times 1 + 0.1 \times 16)$  +  $10(0.8 \times 1 + 0.2 \times 16)$

$$\frac{\text{Senza - cache}}{\text{Con - cache}} = \frac{100 \times 10}{90(0,9 \times 1 + 0,1 \times 16) + 10(0,8 \times 1 + 0,2 \times 16)} = 5.4$$

# Esercizi sulle cache

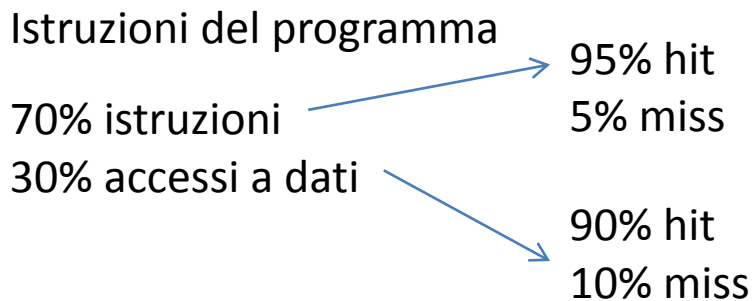


**Domanda 3 (Punti 10):** Si ipotizzi che il 30% delle istruzioni in un programma tipico effettui un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache sia del 95% per le istruzioni e del 90% per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e una sia in cache che in memoria costi 16. Quale sarebbe il guadagno utilizzando la cache?



# Esercizi sulle cache

**Domanda 3 (Punti 10):** Si ipotizzi che il 30% delle istruzioni in un programma tipico effettui un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache sia del 95% per le istruzioni e del 90% per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e una sia in cache che in memoria costi 16. Quale sarebbe il guadagno utilizzando la cache?



Accesso senza cache:  $100 \times 10$

Accesso con cache: costo istruzioni + costo dati

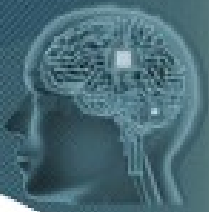
$70(0.95 \times 1 + 0.05 \times 16)$        $30(0.9 \times 1 + 0.1 \times 16)$

$$\frac{\text{Senza - cache}}{\text{Con - cache}} = \frac{100 \times 10}{70(0.95 \times 1 + 0.05 \times 16) + 30(0.9 \times 1 + 0.1 \times 16)} = 6,79$$

# Esercizi sulle cache



**Domanda 1 (Punti 10):** Si ipotizzi che  $\frac{4}{5}$  delle istruzioni in un programma tipico effettuino un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache per le istruzioni sia  $\frac{2}{3}$  di quella per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e che una in entrambe (sia in memoria, che in cache) costi 16. Quale deve essere il tasso di successo per avere un guadagno maggiore di 1?



# Esercizi sulle cache

**Domanda 1 (Punti 10):** Si ipotizzi che  $\frac{4}{5}$  delle istruzioni in un programma tipico effettuino un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache per le istruzioni sia  $\frac{2}{3}$  di quella per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e che una in entrambe (sia in memoria, che in cache) costi 16. Quale deve essere il tasso di successo per avere un guadagno maggiore di 1?

Numero totale di accessi in RAM =  $n$

Lecture/Scritture Dati ( $D$ ) =  $\frac{4}{5} n = 0.8 n$

Istruzioni ( $I$ ) =  $n - \frac{4}{5} n = 0.2 n$

Frequenza di hit per i dati in cache ( $hD$ ) =  $x$

Frequenza di hit per le istruzioni in cache ( $hI$ ) =  $\frac{2}{3} x$

Tempo medio di accesso =  $I \times (hI \times 1 + (1 - hI) \times 16) + D \times (hD \times 1 + (1 - hD) \times 16)$

Tempo medio di accesso =  $0.2n \times (\frac{2}{3}x \times 1 + (1 - \frac{2}{3}x) \times 16) + 0.8n \times (x \times 1 + (1 - x) \times 16)$

Tempo medio di accesso =  $16n - 14 n x$



# Esercizi sulle cache

Numero totale di accessi in RAM =  $n$

Lecture/Scritture Dati ( $D$ ) =  $4/5 n = 0.8 n$

Istruzioni ( $I$ ) =  $n - 4/5 n = 0.2 n$

Frequenza di hit per i dati in cache ( $hD$ ) =  $x$

Frequenza di hit per le istruzioni in cache ( $hI$ ) =  $2/3 x$

Tempo medio di accesso =  $I \times (hI \times 1 + (1 - hI) \times 16) + D \times (hD \times 1 + (1 - hD) \times 16)$

Tempo medio di accesso =  $0.2n \times (2/3x \times 1 + (1 - 2/3x) \times 16) + 0.8n \times (x \times 1 + (1 - x) \times 16)$

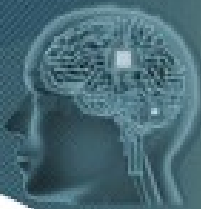
Tempo medio di accesso =  $16n - 14 n x = n (16 - 14 x)$

Tempo di accesso senza cache =  $10 n$

Guadagno =  $10 n / n (16 - 14 x) = 10 / (16 - 14 x)$

Guadagno  $> 1 \implies 10 / (16 - 14 x) > 1 \implies 10 > (16 - 14 x) \implies x > 6 / 14 = 0.42857$





# Esercizi sulle cache

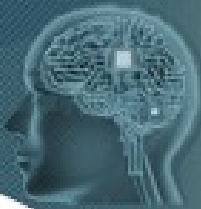
**Domanda 1 (Punti 10):** Si ipotizzi che **4/5** delle istruzioni in un programma tipico effettui una operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache per le istruzioni sia **2/3** di quella per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi **10**, una in cache costi **1** e che una in entrambe (sia in memoria, che in cache) costi **16**. Quale deve essere il tasso di successo per avere un guadagno maggiore di **1**?

Istruzioni del programma  
20% istruzioni  $\rightarrow$   $2/3x$  hit  
 $1 - 2/3x$  miss  
80% accessi a dati  $\rightarrow$   $x$  hit  
 $1 - x$  miss

Accesso senza cache:  $100 \times 10$

Accesso con cache: costo istruzioni + costo dati  
 $20(2/3x \cdot 1 + (1 - 2/3x) \cdot 16) + 80(x \cdot 1 + (1 - x) \cdot 16)$

$$\frac{\text{Senza-cache}}{\text{Con-cache}} = \frac{100 \times 10}{20(2/3x + (1 - 2/3x) \times 16) + 80(x + (1 - x) \times 16)} > 1 \quad \Rightarrow \quad \frac{1000}{1600 - 1400x} > 1 \quad \begin{matrix} x > 0.42857 \\ 2/3x > 0.2857 \end{matrix}$$



# Esercizi sulle cache

**Domanda 3 (Punti 10):** Si ipotizzi che  $\frac{4}{3}$  delle istruzioni in un programma tipico effettuino un'operazione di scrittura o di lettura in RAM, e che la frequenza di successo di lettura in cache per le istruzioni sia  $\frac{1}{2}$  di quella per i dati. Si supponga, inoltre, che la penalità di fallimento sia la stessa per operazioni di scrittura e operazioni di lettura. Si supponga che, in cicli di clock, una lettura diretta in memoria costi 10, una in cache costi 1 e una sia in cache che in memoria costi 16. Quale deve essere il tasso di successo per avere un guadagno maggiore di 1?

Istruzioni del programma

25% istruzioni

75% accessi a dati

$x$  hit

$1-x$  miss

$2x$  hit

$1-2x$  miss

Accesso senza cache:  $100 \times 10$

Accesso con cache: costo istruzioni

+

costo dati

$$25(x \cdot 1 + (1-x) \cdot 16)$$

$$75(2x \cdot 1 + (1-2x) \cdot 16)$$

$$\frac{\text{Senza-cache}}{\text{Con-cache}} = \frac{100 \times 10}{25(x + (1-x) \times 16) + 75(2x + (1-2x) \times 16)} > 1$$



$$x > 0.2286$$

$$2x > 0.4573$$