



Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Esercizi Architetture Avanzate dei Calcolatori - CACHE -

Silvia Lovergine

Politecnico di Milano

A.A. 2011-2012



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

1 Struttura di Cache

2 Problemi Essenziali per le Gerarchie di Memoria

3 Soluzioni di Cache

4 Esercizio 1

5 Esercizio 2

6 Esercizio 3

7 Esercizio 4

8 Esercizio 5



Struttura di Cache

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

BLOCCO DATI	TAG	D	V	
				Linea 1
				Linea 2
				Linea 3
...

Ogni linea di cache contiene, in aggiunta al blocco di dati:

- Tag - identifica univocamente un blocco in cache
- D (dirty bit) - quando è settato ad 1, indica che il blocco in cache è stato modificato, e viceversa (facoltativo)
- V (validity bit) - quando è settato ad 1, indica che il blocco in cache è valido, e viceversa



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

1 Struttura di Cache

2 Problemi Essenziali per le Gerarchie di Memoria

3 Soluzioni di Cache

4 Esercizio 1

5 Esercizio 2

6 Esercizio 3

7 Esercizio 4

8 Esercizio 5



Problemi Essenziali

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- 1 **Dove (block placement)** - dove caricare un blocco proveniente da un livello gerarchico inferiore?
- 2 **Come (block identification)** - come individuare un blocco in un livello gerarchico superiore?
- 3 **Quale (block replacement)** - quale blocco sostituire in caso di miss per fare posto ad un blocco del livello gerarchico sottostante? (FIFO, LRU, RANDOM)
- 4 **Politica di Scrittura (write policy)** - come gestire le modifiche dei blocchi? (write back, write through)



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

1 Struttura di Cache

2 Problemi Essenziali per le Gerarchie di Memoria

3 Soluzioni di Cache

4 Esercizio 1

5 Esercizio 2

6 Esercizio 3

7 Esercizio 4

8 Esercizio 5



Cache ad Indirizzamento Diretto

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Ciascun blocco di RAM va mappato su un preciso blocco di cache
- Struttura dell'indirizzo di RAM:





Cache ad Indirizzamento Diretto

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

■ DOVE (block placement)

- (indirizzo blocco) MOD (numero blocchi in cache)
- Esempio: dove caricare il blocco di RAM $(12)_{10}$ se la cache è composta da 8 blocchi? $\rightarrow (12)_{10} \text{ MOD } (8)_{10} = (4)_{10} \rightarrow$ il blocco $(12)_{10}$ di RAM va mappato sul blocco $(4)_{10}$ di cache
- Equivale a considerare il campo **INDICE** dell'indirizzo in RAM

■ COME (block identification)

- Dato l'indirizzo in RAM, si considera il campo **INDICE**, che indicherà il blocco di cache in cui cercare il dato. Se in quel blocco il *tag* è uguale al tag dell'indirizzo in RAM e il *validity bit* è settato ad 1, allora il dato è presente in cache ed è valido.
[Oss: il dato va cercato in una sola linea di cache!]



Cache Completamente Associativa

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Ciascun blocco di RAM può essere mappato in qualsiasi blocco di cache
- Struttura dell'indirizzo di RAM:





Cache Completamente Associativa

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- **DOVE (block placement)**

- In qualsiasi blocco di cache

- **COME (block identification)**

- Dato l'indirizzo in RAM, si considera il campo **TAG**, che indicherà il blocco di RAM da cercare in cache. Tale campo deve essere confrontato con l'omonimo campo in tutte le linee di cache. Se si trova un matching sul campo TAG e il *validity bit* è settato ad 1, allora il dato è presente in cache ed è valido.

[Oss: il dato va cercato in tutte le linee di cache!]



Cache Set Associativa

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Ciascun blocco di RAM va mappato in uno qualsiasi dei blocchi di un preciso set nella cache
- Struttura dell'indirizzo di RAM:





Cache Set Associativa

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

■ DOVE (block placement)

■ (indirizzo blocco) MOD (numero set in cache)

■ Esempio: dove caricare il blocco di RAM $(12)_{10}$ se la cache è composta da 4 set, ciascuno contenente 2 blocchi?

→ $(12)_{10} \text{ MOD } (4)_{10} = (0)_{10}$ → il blocco $(12)_{10}$ di RAM va mappato in uno qualsiasi dei 2 blocchi del set $(0)_{10}$

■ Equivale a considerare il campo **SET** dell'indirizzo in RAM

■ COME (block identification)

■ Dato l'indirizzo in RAM, si considera il campo **SET**, che indicherà il set nella cache in cui cercare il dato.

All'interno del set individuato, bisogna effettuare una ricerca associativa sul campo TAG di tutte le linee di cache in quel set. Se si trova un matching sul campo TAG e il *validity bit* è settato ad 1, allora il dato è presente in cache ed è valido.

[Oss: il dato va cercato in tutti i blocchi di un solo set!]



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- 1 Struttura di Cache
- 2 Problemi Essenziali per le Gerarchie di Memoria
- 3 Soluzioni di Cache
- 4 Esercizio 1**
- 5 Esercizio 2
- 6 Esercizio 3
- 7 Esercizio 4
- 8 Esercizio 5



Esercizio 1

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Sia data un'architettura composta da una memoria RAM da 1GB, cache da 1MB e blocchi da 512 byte, indirizzata al byte.

- 1 Specificare la struttura dell'indirizzo di memoria nel caso di cache ad indirizzamento diretto, completamente associativa e set associativa a 4 vie.
- 2 Calcolare la dimensione totale della cache nei tre casi indicati al punto 1, sapendo che la cache è stata ottimizzata con l'aggiunta del dirty bit.
- 3 Ipotizzando la cache inizialmente vuota, si consideri il caso in cui la CPU debba caricare in cache per 3 volte di seguito una sequenza di 800KB di dati, memorizzati in modo consecutivo a partire dall'indirizzo 0 di RAM. Sapendo che la politica di sostituzione dei blocchi è LRU, quale delle 3 soluzioni di cache indicate al punto 1 è più conveniente?



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

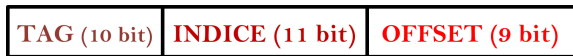
Esercizio 3

Esercizio 4

Esercizio 5

Struttura dell'indirizzo nel caso di cache ad indirizzamento diretto:

- $RAM \rightarrow 1GB = 2^{30} \rightarrow 30$ bit per l'indirizzo
- $CACHE \rightarrow 1MB = 2^{20}$
- $blocco \rightarrow 512byte = 2^9 \rightarrow 9$ bit per l'offset
- $\#blocchi_in_RAM = dim.RAM / dim.Blocco = 2^{30} / 2^9 = 2^{21} \rightarrow 21$ bit per blocco in RAM
- $\#blocchi_in_cache = dim.Cache / dim.Blocco = 2^{20} / 2^9 = 2^{11} \rightarrow 11$ bit per Indice
- $\#bit_tag = \#bit_indirizzo - \#bit_offset - \#bit_indice = 30 - 9 - 11 = 10$



BLOCCO (21 bit)



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

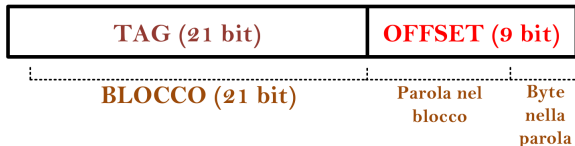
Esercizio 3

Esercizio 4

Esercizio 5

Struttura dell'indirizzo nel caso di cache completamente associativa:

- $RAM \rightarrow 1GB = 2^{30} \rightarrow 30$ bit per l'indirizzo
- $CACHE \rightarrow 1MB = 2^{20}$
- $blocco \rightarrow 512byte = 2^9 \rightarrow 9$ bit per l'offset
- $\#blocchi_in_RAM = dim.RAM / dim.Blocco = 2^{30} / 2^9 = 2^{21} \rightarrow 21$ bit per blocco in RAM
- $\#bit_tag = \#bit_indirizzo - \#bit_offset - \#bit_indice = 30 - 9 = 21$
- Nota: $\#blocchi_in_RAM$ e $\#bit_tag$ coincidono nel caso di cache completamente associativa





Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

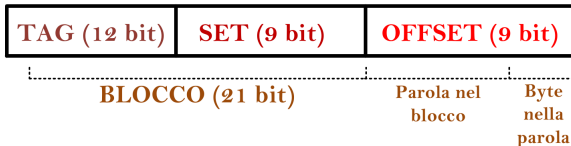
Esercizio 3

Esercizio 4

Esercizio 5

Struttura dell'indirizzo nel caso di cache set associativa a 4 vie:

- $RAM \rightarrow 1GB = 2^{30} \rightarrow 30$ bit per l'indirizzo
- $CACHE \rightarrow 1MB = 2^{20}$
- $blocco \rightarrow 512byte = 2^9 \rightarrow 9$ bit per l'offset
- $\#blocchi_in_RAM = dim.RAM / dim.Blocco = 2^{30} / 2^9 = 2^{21} \rightarrow 21$ bit per blocco in RAM
- $dim.set = \#vie * dim.blocco = 2^2 * 2^9 = 2^{11}$
- $\#set_in_cache = dim.cache / dim.set = 2^{20} / 2^{11} = 2^9 \rightarrow 9$ bit per set in cache
- $\#bit_tag = \#bit_indirizzo - \#bit_offset - \#bit_set = 30 - 9 - 9 = 12$





Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Cache ad indirizzamento diretto:

$$\text{dim.tot.} = \# \text{blocchi} * (\text{dim.Blocco} + \text{dim.Tag} + 2\text{bit}) =$$

//2 bit per validity e dirty

$$= 2^{11} * (2^9 \text{byte} + 10\text{bit} + 2\text{bit}) =$$

$$= 2^{20} \text{byte} + (2^{11} \text{byte} * 12\text{bit}) =$$

$$= 1\text{MB} + 2^{11} * (2^3 \text{bit} + 2^2 \text{bit}) =$$

$$= 1\text{MB} + 2^{14} \text{bit} + 2^{13} \text{bit} =$$

$$= 1\text{MB} + 2^{11} \text{byte} + 2^{10} \text{byte} =$$

$$= 1\text{MB} + 2\text{KB} + 1\text{KB} = 1024\text{KB} = 1,003\text{MB}$$



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Cache completamente associativa:

$$\begin{aligned} \text{dim.tot.} &= \# \text{blocchi} * (\text{dim.Blocco} + \text{dim.Tag} + 2\text{bit}) = \\ &= 2^{11} * (2^9 \text{byte} + 21\text{bit} + 2\text{bit}) = \\ &= 2^{20} \text{byte} + (2^{11} * 23\text{bit}) = \\ &= 1\text{MB} + 2^{11} * (2^4 \text{bit} + 2^2 \text{bit} + 2\text{bit} + 1\text{bit}) = \\ &= 1\text{MB} + 2^{15} \text{bit} + 2^{13} \text{bit} + 2^{12} \text{bit} + 2^{11} \text{bit} = \\ &= 1\text{MB} + 2^{12} \text{byte} + 2^{10} \text{byte} + 2^9 \text{byte} + 2^8 \text{byte} = \\ &= 1\text{MB} + 4\text{KB} + 1\text{KB} + 512\text{byte} + 256\text{byte} = \\ &= 1005,768\text{KB} \approx 1,006\text{MB} \end{aligned}$$



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Cache set associativa a 4 vie:

$$\begin{aligned} \text{dim.tot.} &= \# \text{blocchi} * (\text{dim.Blocco} + \text{dim.Tag} + 2\text{bit}) = \\ &= 2^{11} * (2^9 \text{byte} + 12\text{bit} + 2\text{bit}) = \\ &= 2^{20} \text{byte} + (2^{11} * 14\text{bit}) = \\ &= 1\text{MB} + 2^{11} * (2^3 \text{bit} + 2^2 \text{bit} + 2\text{bit}) = \\ &= 1\text{MB} + 2^{14} \text{bit} + 2^{13} \text{bit} + 2^{12} \text{bit} = \\ &= 1\text{MB} + 2^{11} \text{byte} + 2^{10} \text{byte} + 2^9 \text{byte} = \\ &= 1\text{MB} + 2\text{KB} + 1\text{KB} + 512\text{byte} = \\ &= 1003,512\text{KB} \approx 1,035\text{MB} \end{aligned}$$



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Per stabilire quale delle soluzioni di cache è più conveniente in caso di caricamento di una sequenza di 800KB di dati, ripetuta per 3 volte, bisogna calcolare i cache miss.
- Dato che sono necessari 9 bit per l'offset, ciascun blocco può contenere 2^9 byte (=512 byte), quindi bisognerà caricare $800KB/512byte = 1562.5 \rightarrow 1563$ blocchi dalla RAM alla cache.
- La cache può contenere fino a $1MB/512byte = 1953,125 \rightarrow 1953$ blocchi



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Cache ad indirizzamento diretto:
 - Prima sequenza di caricamento:
 - Tutti i 1563 blocchi causano miss in quanto la cache era inizialmente vuota
 - Nota: tutti i blocchi hanno una precisa posizione in cache, e andranno memorizzati dalla linea 0 alla linea 1562 di cache.
 - Seconda sequenza di caricamento:
 - I 1563 blocchi sono già in cache → nessun miss
 - Terza sequenza di caricamento:
 - I 1563 blocchi sono già in cache → nessun miss

Totale: 1563 miss



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Cache completamente associativa:
 - Prima sequenza di caricamento:
 - Tutti i 1563 blocchi causano miss in quanto la cache era inizialmente vuota. I blocchi possono essere mappati in qualsiasi posizione in cache. Supponendo di memorizzare ciascun blocco di RAM nella prima linea di cache libera, abbiamo 1563 miss “a freddo”.
 - Seconda e terza sequenza di caricamento:
 - I 1563 blocchi sono già in cache → nessun miss

Totale: 1563 miss



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Cache set-associativa a 4 vie:
 - Prima sequenza di caricamento:
 - Tutti i 1563 blocchi causano miss in quanto la cache era inizialmente vuota
 - Nota: i blocchi possono essere mappati in qualsiasi posizione di un preciso set in cache. In cache ci sono $2^9 = 512$ set. Quindi, i blocchi di RAM 0, 512, 1024 e 1536 vanno nel set 0, ..., i blocchi di RAM 26, 538, 1050 e 1562 vanno nel set 26, i blocchi di RAM 27, 539 e 1051 vanno nel set 511, e così via. Sui set dallo 0 al 26 sono mappati 4 blocchi di RAM, mentre dal 27 al 511 sono mappati 3 blocchi di RAM. Supponendo di memorizzare ciascun blocco di RAM nella prima linea di cache libera del set associato, abbiamo 1563 miss a freddo.



Esercizio 1 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Seconda e terza sequenza di caricamento:
 - I 1563 blocchi sono già in cache → nessun miss

Totale: 1563 miss

NOTA: era possibile arrivare alla stessa soluzione osservando che la quantità di dati da caricare in cache (800 KB) è inferiore alla capacità totale della cache, quindi, a prescindere dalla politica di block placing scelta, a partire dalla seconda sequenza di caricamento i dati sono già tutti in cache.

In conclusione, in questo caso specifico le tre soluzioni di cache sono equivalenti in quanto causano lo stesso numero di cache miss.



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

1 Struttura di Cache

2 Problemi Essenziali per le Gerarchie di Memoria

3 Soluzioni di Cache

4 Esercizio 1

5 Esercizio 2

6 Esercizio 3

7 Esercizio 4

8 Esercizio 5



Esercizio 2

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Sia data un'architettura composta da una memoria RAM da 4KB, cache da 512 byte e blocchi da 128 byte. Si consideri la seguente sequenza di richieste di lettura (R) e scrittura (W) della memoria:

R 000000010010

R 000000010011

R 100100010100

W 000011111111

W 111110001010

W 011110010010

W 110000000110



Esercizio 2

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Specificare il contenuto della cache a seguito delle richieste indicate, nei seguenti casi:

- 1 cache ad indirizzamento diretto, politica di sostituzione FIFO e politica di scrittura write back (con dirty bit). Cosa cambierebbe cambiando la politica di sostituzione in questo caso?
- 2 cache completamente associativa, politica di sostituzione LRU e politica di scrittura write back (con dirty bit)
- 3 cache set associativa a 2 vie, politica di sostituzione RANDOM e politica di scrittura write through



Esercizio 2 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- RAM da 4KB \rightarrow 12 bit per indirizzo
- blocchi da 128 byte $= 2^7$ byte \rightarrow 7 bit per offset
- $\#Blocchi_in_cache = dim.Cache / dim.Blocco = 512 / 128 = 2^9 / 2^7 = 2^2 \rightarrow$ 2 bit per l'indice nel caso di cache ad indirizzamento diretto
- $dim.Set = \#vie * dim.Blocco = 2 * 128 = 2 * 2^7 = 2^8$ e $\#_set_in_cache = dim.Cache / dim.Set = 2^9 / 2^8 = 2 \rightarrow$ 1 bit per set nel caso di cache set-associativa a 2 vie
- Struttura indirizzo cache ad indirizzamento diretto:
Tag=3bit, Indice=2bit, Offset=7bit
- Struttura indirizzo cache completamente associativa:
Tag=5bit, Offset=7bit
- Struttura indirizzo cache set-associativa a 2 vie:
Tag=4bit, Set=1bit, Offset=7bit



Esercizio 2 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Cache ad indirizzamento diretto, politica di sostituzione FIFO e politica di scrittura write back

		BLOCCO 0					BLOCCO 1					BLOCCO 2					BLOCCO 3					
Richiesta	E	V	D	TAG	dato	V	D	TAG	dato	V	D	TAG	dato	V	D	TAG	dato	NOTE				
R 000000010010	M	1	0	000	0	0	0	-	-	0	0	-	-	0	0	-	-					
R 000000010011	H	1	0	000	0	0	0	-	-	0	0	-	-	0	0	-	-					
R 100100010100	M	1	0	000	0	0	0	-	-	1	0	100	18	0	0	-	-					
W 000011111111	M	1	0	000	0	1	1	000	1	1	0	100	18	0	0	-	-					
W 111110001010	M	1	0	000	0	1	0	000	1	0	0	100	18	1	1	111	31					
W 011110010010	M	1	0	000	0	1	0	000	1	0	0	100	18	1	1	011	15	Copia blocco 31 in RAM				
W 110000000110	M	1	1	110	24	1	0	000	1	0	0	100	18	1	1	111	15					



Esercizio 2 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Cache completamente associativa, politica di sostituzione LRU
e politica di scrittura write back

		BLOCCO 0					BLOCCO 1				BLOCCO 2				BLOCCO 3				
Richiesta	E	V	D	TAG	dato	V	D	TAG	dato	V	D	TAG	dato	V	D	TAG	dato	NOTE	
R 000000010010	M	1	0	00000	0	0	0	-	-	0	0	-	-	0	0	-	-		
R 000000010011	H	1	0	00000	0	0	0	-	-	0	0	-	-	0	0	-	-		
R 100100010100	M	1	0	00000	0	1	0	10010	18	0	0	-	-	0	0	-	-		
W 000011111111	M	1	0	00000	0	1	0	10010	18	1	1	00001	1	0	0	-	-		
W 111110001010	M	1	0	00000	0	1	0	10010	18	1	1	00001	1	1	1	111	31		
W 011110010010	M	1	1	00000	15	1	0	10010	18	0	0	00001	1	1	1	111	31	(LRU)	
W 110000000110	M	1	1	00000	15	1	1	11000	24	0	0	00001	1	1	1	111	15	(LRU)	



Esercizio 2 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Cache set associativa a 2 vie, politica di sostituzione RANDOM
e politica di scrittura write through

Richiesta	E	V	SET 0						SET 1						NOTE
			BLOCCO 0			BLOCCO 1			BLOCCO 2			BLOCCO 3			
			TAG	dato		V	TAG	dato		V	TAG	dato		V	
R 000000010010	M	1	0000	0	0	-	-	0	-	-	0	-	-		
R 000000010011	H	1	0000	0	0	-	-	0	-	-	0	-	-		
R 100100010100	M	1	0000	0	1	1001	18	0	-	-	0	-	-		
W 000011111111	M	1	0000	0	1	1001	18	1	0000	1	0	-	-	Copia blocco 1 in RAM	
W 111110001010	M	1	0000	0	1	1001	18	1	0000	1	1	1111	31	Copia blocco 31 in RAM	
W 011110010010	M	1	0000	0	1	1001	18	1	0111	15	1	1111	31	Copia blocco 15 in RAM	
W 110000000110	M	1	1100	24	1	1001	18	1	0111	15	1	1111	31	Copia blocco 24 in RAM	



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

1 Struttura di Cache

2 Problemi Essenziali per le Gerarchie di Memoria

3 Soluzioni di Cache

4 Esercizio 1

5 Esercizio 2

6 Esercizio 3

7 Esercizio 4

8 Esercizio 5



Esercizio 3

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Data la seguente tabella:

	L1	L2
SIZE	32 KB	1MB
HitRate	80%	90%
HitTime	1ns	27ns
MissPenalty	10ns	750 ns

- 1 Calcolare il tempo di accesso T_A nel caso di architettura rispettivamente con singolo e doppio livello di cache.
- 2 Quale MissPenalty deve avere L1 perchè il tempo di accesso dell'architettura con singolo livello di cache sia pari a 13ns, lasciando invariati tutti gli altri parametri?
- 3 Quale HitRate deve avere L2 perchè il tempo di accesso dell'architettura con due livelli di cache sia pari a 4 ns, lasciando invariati tutti gli altri parametri?



Esercizio 3 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Con singolo livello di cache:

$$T_A = hit_time_{L1} + miss_rate_{L1} * miss_penalty_{L1} = 1ns + (1 - 0.8) * 10ns = 3ns$$

- Con doppio livello di cache:

$$T_A = hit_time_{L1} + miss_rate_{L1} * (hit_time_{L2} + miss_rate_{L2} * miss_penalty_{L2}) = 1ns + 0.2 * (27ns + 0.1 * 750ns) = 21.4ns$$

- Per rispondere al punto 2 poniamo:

$$T_A = hit_time_{L1} + miss_rate_{L1} * miss_penalty_{L1} = 13ns \rightarrow 1ns + (1 - 0.8) * miss_penalty_{L1} = 13ns \rightarrow miss_penalty_{L1} = 12/0.2 = 60ns$$



Esercizio 3 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Per rispondere al punto 3 poniamo:

$$T_A = hit_time_{L1} + miss_rate_{L1} * (hit_time_{L2} + miss_rate_{L2} * miss_penalty_{L2}) = 4ns \rightarrow$$

$$1ns + 0.2 * (27ns + (1 - hit_rate_{L2}) * 750ns) = 4ns \rightarrow$$

$$27ns + (1 - hit_rate_{L2}) * 750ns = 15 \rightarrow (1 - hit_rate_{L2}) = (15 - 27)/750 \rightarrow hit_rate_{L2} = 1 + 12/750 > 100\%!!!$$

- Quindi non è possibile ottenere un tempo di accesso con doppio livello di cache pari a 4 ns modificando solo l' hit_rate_{L2} .



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

1 Struttura di Cache

2 Problemi Essenziali per le Gerarchie di Memoria

3 Soluzioni di Cache

4 Esercizio 1

5 Esercizio 2

6 Esercizio 3

7 Esercizio 4

8 Esercizio 5



Esercizio 4

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Sia data un'architettura composta da CPU e cache, avente miss penalty pari a 6 cicli di clock, CPI ideale pari a 8.5 cicli di clock e miss rate pari a 11%. Supponendo che per ciascuna istruzione siano necessari 2 accessi in memoria per recuperare i dati, calcolare il CPI reale, lo speedup del sistema ideale rispetto a quello reale, il CPI del sistema privo di cache e lo speedup del sistema privo di cache rispetto a quello dotato di cache.

■ Soluzione:

- $CPI_{REALE} = CPI_{IDEALE} + miss_rate * miss_penalty * \#riferimenti_memoria = 8.5 + 0.11 * 6 * 3 = 10.48$
- $speedup_{ideale_vs_reale} = (CPI_{REALE} * IC * f) / (CPI_{IDEALE} * IC * f) = 10.48 / 8.5 \approx 1.23$
- $CPI_{NO_CACHE} = CPI_{IDEALE} + miss_penalty * \#riferimenti_memoria = 8.5 + 6 * 3 = 26.5$
- $speedup_{nocache_vs_reale} = (CPI_{REALE} * IC * f) / (CPI_{NO_CACHE} * IC * f) = 10.48 / 26.5 \approx 0.39$



Contents

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- 1 Struttura di Cache
- 2 Problemi Essenziali per le Gerarchie di Memoria
- 3 Soluzioni di Cache
- 4 Esercizio 1
- 5 Esercizio 2
- 6 Esercizio 3
- 7 Esercizio 4
- 8 Esercizio 5



Esercizio 5

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

Sia data un'architettura composta da una memoria RAM da 4Mparole, indirizzata a parole, una cache da 1Kparola e blocchi da 64 parole, dove ogni parola è composta da 4 byte.

- 1 Strutturare gli indirizzi di memoria nel caso di cache ad indirizzamento diretto, cache completamente associativa e cache set associativa a 2 vie.
- 2 Ipotezzando la cache inizialmente vuota, si consideri il caso in cui la CPU debba caricare in cache 1025 parole memorizzate in modo adiacente nella memoria RAM a partire dall'indirizzo 0, ripetendo la sequenza di caricamento per 5 volte consecutive. Calcolare il numero di cache miss in caso di cache ad indirizzamento diretto e di cache completamente associativa, con politica di sostituzione LRU.



Esercizio 5 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- $\dim.RAM = 4Mparole = 2^{22}parole \rightarrow 22$ bit per l'indirizzo
- $\dim.Cache = 1Kparola = 2^{10}parole$
- $\dim.Blocco = 64parole = 2^6parole \rightarrow 6$ bit per offset
- $\#blocchi_{in_cache} = 2^{10}/2^6 = 2^4 \rightarrow 4$ bit per campo indice in caso di cache ad indirizzamento diretto
- $\dim.Set = 2 * 2^6 = 2^7$
- $\#set_{in_cache} = 2^{10}/2^7 = 2^3 \rightarrow 3$ bit per campo set in caso di cache set-associativa a 2 vie
- Struttura dell'indirizzo in presenza di cache ad indirizzamento diretto: Tag=12 bit, Indice=4 bit, Offset=6 bit
- Struttura dell'indirizzo in presenza di cache completamente associativa: Tag=16 bit, Offset=6 bit
- Struttura dell'indirizzo in presenza di cache set-associativa a 2 vie: Tag=13 bit, Set=3 bit, Offset=6 bit



Esercizio 5 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Bisogna caricare 1025 parole, memorizzate a partire dall'indirizzo 0 di memoria, per 5 volte consecutive.
- $1025 \text{ parole} = 1024 + 1 \text{ parole} = (2^{10} + 1) \text{ parole}$
- Dato che ciascun blocco è grande 2^6 parole:
 $2^{10}/2^6 = 2^4 \rightarrow 2^{10}$ parole stanno in 2^4 blocchi, più un blocco per l'ultima parola.
- Se la cache è ad indirizzamento diretto risulta:
 - 1° ciclo: 1024 miss a freddo + 1 miss per sostituzione (la 1025^a parola prende il posto della prima)
 - 2°, 3°, 4°, 5° ciclo: 1 miss per la prima parola che prende il posto della 1025^a + un miss per la 1025^a parola che prende il posto della prima
 - Totale: $1025 + 2 \cdot 4 = 1033$ miss



Esercizio 5 - Soluzione

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Se la cache è completamente associativa, e la politica di sostituzione è LRU risulta:
 - 1° ciclo: 1024 miss a freddo + 1 miss per sostituzione (la 1025^a parola prende il posto della prima)
 - 2° ciclo: 1 miss per la prima parola che prende il posto della seconda (LRU) + un miss per la seconda parola che prende il posto della terza, e così via → 1025 miss
 - 3°, 4°, 5° ciclo: come il secondo ciclo, partendo rispettivamente dalla terza, quarta e quinta linea di cache.
 - Totale: $1025 \times 5 = 5125$ miss



Esercizio 5 - Osservazioni

Cache

Silvia
Lovergine

Struttura di
Cache

Problemi
Essenziali per
le Gerarchie di
Memoria

Soluzioni di
Cache

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4

Esercizio 5

- Quando la memoria è indirizzata a *word*, la parola è la più piccola unità di memoria indirizzabile. Questa soluzione ha il vantaggio di utilizzare indirizzi di memoria più compatti, che occuperanno meno spazio nella memoria istruzioni.
- Quando invece la memoria è indirizzata al *byte*, l'unità di risoluzione della memoria è il byte. Le parole possono comunque essere indirizzate, ma l'indirizzo di memoria richiede alcuni bit aggiuntivi rispetto al caso precedente. L'approccio più diffuso è quello in cui la dimensione di una parola è un multiplo intero della dimensione del byte.