

# Calcolatori Elettronici A

a.a. 2008/2009

## **Memoria cache: Esercizi**

*Massimiliano Giacomini*

# Esercizio: miss della cache e collocazione dei blocchi nella cache

- Sia data la seguente **sequenza di indirizzi** a cui si intende fare accesso, espressi come indirizzi **di parola**:  
1, 4, 8, 5, 33, 66, 32, 56, 9, 11, 4, 43, 88, 6, 32
- Sia data una **cache con blocchi di 4 parole** e una **dimensione** totale di **32 parole**
- Determiniamo se ciascuno degli accessi è un **hit** o un **miss** assumendo che la cache sia inizialmente vuota, nei 3 casi seguenti:
  - 1) cache a corrispondenza diretta
  - 2) cache totalmente associativa, ipotizzando una sostituzione LRU
  - 3) cache set-associativa a 2 vie, ipotizzando una sostituzione LRU
- Assumendo indirizzi a **16 bit**, determiniamo anche il numero di bit riservati all'etichetta (tag), al blocco (o insieme) e alla parola

## Calcolo degli indirizzi di blocco

- Gli indirizzi a cui dobbiamo fare accesso sono espressi come indirizzi di parola
- Siccome, in ogni caso, dobbiamo caricare nella cache un blocco di 4 parole, trasformiamo gli **indirizzi di parola** in **indirizzi di blocco**
- Usiamo la seguente equazione

$$\text{indirizzo di blocco} = \left\lfloor \frac{\text{indirizzo di parola}}{\text{ampiezza di un blocco}} \right\rfloor = \left\lfloor \frac{\text{indirizzo di parola}}{4} \right\rfloor$$

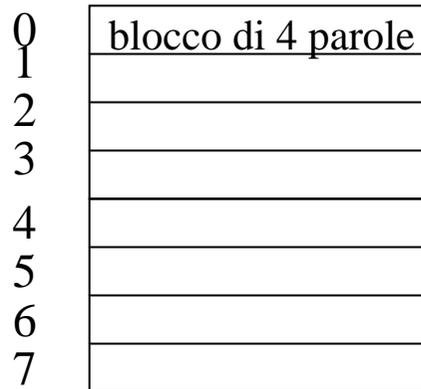
- E quindi troviamo le seguenti corrispondenze

indirizzo di parola	1, 4, 8, 5, 33, 66, 32, 56, 9, 11, 4, 43, 88, 6, 32
	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
indirizzo di blocco	0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

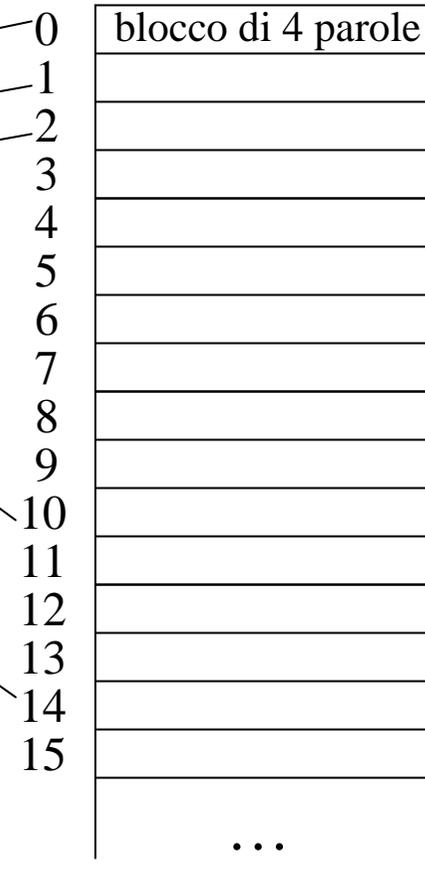
# Caso 1: cache a corrispondenza diretta

Indirizzi di blocco      0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8  
 Blocco della cache    0, 1, 2, 1, 0, 0, 0, 6, 2, 2, 1, 2, 6, 1, 0

**Cache**



**Memoria**



Un blocco della memoria può andare in un solo blocco della cache



## Caso 1: cache a corrispondenza diretta

Indirizzi di blocco            0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

**blocco della cache** = (numero blocco memoria) modulo (numero blocchi in cache)

Blocco	1	2	3	4	5	6	7	8
0	<b>mem[0]</b>	mem[0]	mem[0]	mem[0]	<b>mem[8]</b>	<b>mem[16]</b>	<b>mem[8]</b>	mem[8]
1		<b>mem[1]</b>	mem[1]	<b>mem[1]</b>	mem[1]	mem[1]	mem[1]	mem[1]
2			<b>mem[2]</b>	mem[2]	mem[2]	mem[2]	mem[2]	mem[2]
3								
4								
5								
6								<b>mem[14]</b>
7								

miss      miss      miss      hit      miss      miss      miss      miss

## Caso 1: cache a corrispondenza diretta

Indirizzi di blocco                    0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

**blocco della cache** = (numero blocco memoria) modulo (numero blocchi in cache)

Blocco	9	10	11	12	13	14	15
0	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	<b>mem[8]</b>
1	mem[1]	mem[1]	<b>mem[1]</b>	mem[1]	mem[1]	<b>mem[1]</b>	mem[1]
2	<b>mem[2]</b>	<b>mem[2]</b>	mem[2]	<b>mem[10]</b>	mem[10]	mem[10]	mem[2]
3							
4							
5							
6	mem[14]	mem[14]	mem[14]	mem[14]	<b>mem[22]</b>	mem[22]	mem[22]
7							

hit

hit

hit

miss

miss

hit

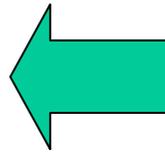
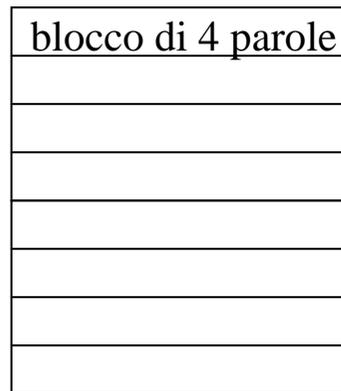
hit

**Risultato totale degli accessi alla cache** = 9 miss e 6 hit

## Caso 2: cache completamente associativa

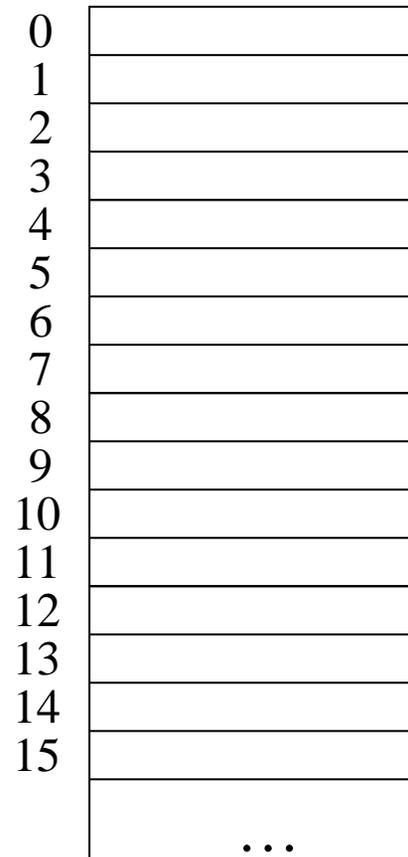
Indirizzi di blocco      0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

*Cache*



Un blocco della memoria può andare  
in qualsiasi blocco della cache

*Memoria*



Indirizzo	Etichetta	Parola
	14	2

## Caso 2: cache completamente associativa

Indirizzi di blocco                    0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

Blocco	1	2	3	4	5	6	7	8
0	<b>mem[0]</b>	mem[0]	mem[0]	mem[0]	mem[0]	mem[0]	mem[0]	mem[0]
1		<b>mem[1]</b>	mem[1]	<b>mem[1]</b>	mem[1]	mem[1]	mem[1]	mem[1]
2			<b>mem[2]</b>	mem[2]	mem[2]	mem[2]	mem[2]	mem[2]
3					<b>mem[8]</b>	mem[8]	<b>mem[8]</b>	mem[8]
4						<b>mem[16]</b>	mem[16]	mem[16]
5								<b>mem[14]</b>
6								
7								

miss      miss      miss      hit      miss      miss      hit      miss

## Caso 2: cache completamente associativa

Indirizzi di blocco                    0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

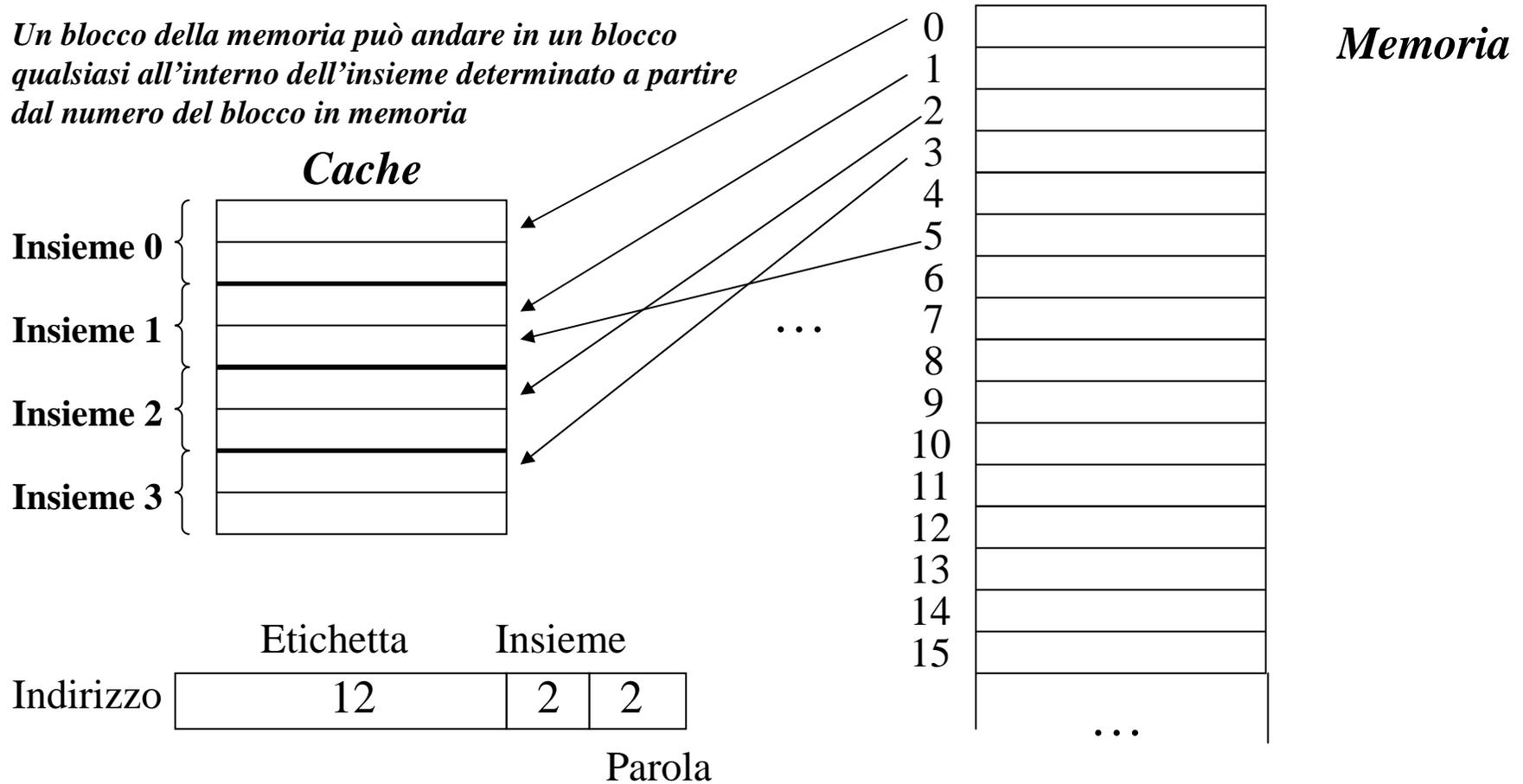
Blocco	9	10	11	12	13	14	15
0	mem[0]	mem[0]	mem[0]	mem[0]	mem[0]	mem[0]	mem[0]
1	mem[1]	mem[1]	<b>mem[1]</b>	mem[1]	mem[1]	<b>mem[1]</b>	mem[1]
2	<b>mem[2]</b>	<b>mem[2]</b>	mem[2]	mem[2]	mem[2]	mem[2]	mem[2]
3	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	<b>mem[8]</b>
4	mem[16]	mem[16]	mem[16]	mem[16]	mem[16]	mem[16]	mem[16]
5	mem[14]	mem[14]	mem[14]	mem[14]	mem[14]	mem[14]	mem[14]
6				<b>mem[10]</b>	mem[10]	mem[10]	mem[10]
7					<b>mem[22]</b>	mem[22]	mem[22]
	hit	hit	hit	miss	miss	hit	hit

**Risultato totale degli accessi alla cache = 8 miss e 7 hit**

### Caso 3: cache set-associativa a due vie

Indirizzi di blocco      0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8  
 Indirizzi di insieme      0, 1, 2, 1, 0, 0, 0, 2, 2, 2, 1, 2, 2, 1, 0

*Un blocco della memoria può andare in un blocco qualsiasi all'interno dell'insieme determinato a partire dal numero del blocco in memoria*



### Caso 3: cache set-associativa a due vie

Indirizzi di blocco            0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

**insieme della cache** = (numero blocco memoria) modulo (numero insiemi in cache)

Insieme	1	2	3	4	5	6	7	8
0	<b>mem[0]</b>	mem[0]	mem[0]	mem[0]	mem[0]	<b>mem[16]</b>	mem[16]	mem[16]
					<b>mem[8]</b>	mem[8]	<b>mem[8]</b>	mem[8]
1		<b>mem[1]</b>	mem[1]	<b>mem[1]</b>	mem[1]	mem[1]	mem[1]	mem[1]
2			<b>mem[2]</b>	mem[2]	mem[2]	mem[2]	mem[2]	mem[2]
								<b>mem[14]</b>
3								
	miss	miss	miss	hit	miss	miss	hit	miss

### Caso 3: cache set-associativa a due vie

Indirizzi di blocco            0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

**insieme della cache** = (numero blocco memoria) modulo (numero insiemi in cache)

Insieme	9	10	11	12	13	14	15
0	mem[16]	mem[16]	mem[16]	mem[16]	mem[16]	mem[16]	mem[16]
	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	<b>mem[8]</b>
1	mem[1]	mem[1]	<b>mem[1]</b>	mem[1]	mem[1]	<b>mem[1]</b>	mem[1]
2	<b>mem[2]</b>	<b>mem[2]</b>	mem[2]	mem[2]	<b>mem[22]</b>	mem[22]	mem[2]
	mem[14]	mem[14]	mem[14]	<b>mem[10]</b>	mem[10]	mem[10]	mem[10]
3							

hit            hit            hit            miss            miss            hit            hit

**Risultato totale degli accessi alla cache = 8 miss e 7 hit**

## Esercizio 2

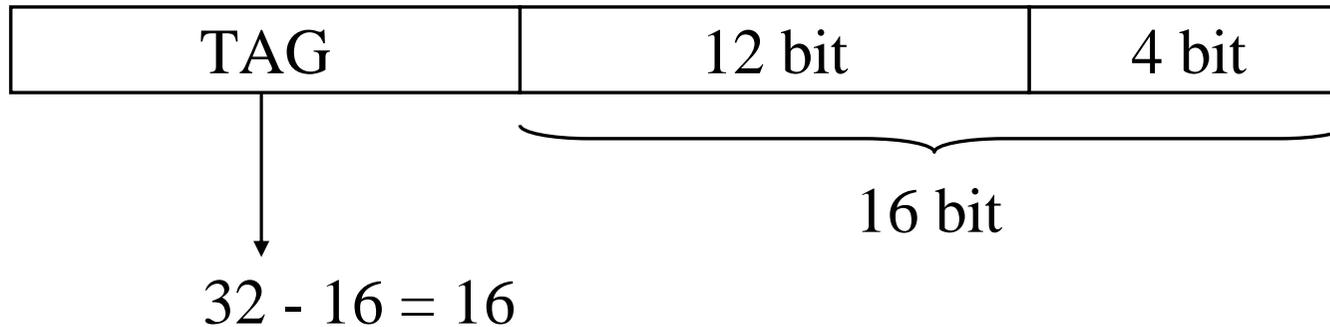
- Sia data una cache con:
  - 4K blocchi
  - ogni blocco ha 4 parole
  - indirizzi di 32 bit
- Trovare il numero totale di bit per i tag nel caso di:
  - cache a corrispondenza diretta
  - cache set associativa a 4 vie
  - cache completamente associativa

## Esercizio 2

- Sia data una cache con:
  - 4K blocchi
  - ogni blocco ha 4 parole
  - indirizzi di 32 bit

Corrispondenza diretta:

$$4K = 2^{12} \text{ blocchi}$$



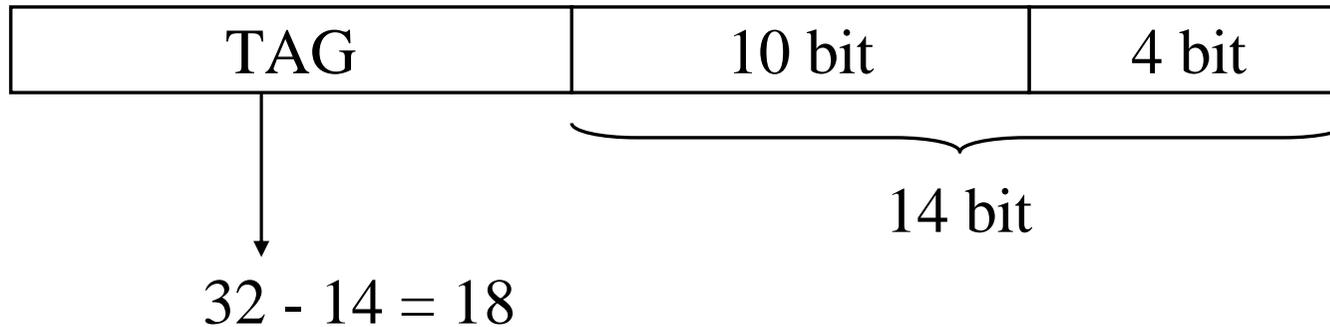
$$\Rightarrow 16 \text{ bit} * 2^{12} = 16 * 4 \text{ kbit} = 64 \text{ kbit}$$

## Esercizio 2

- Sia data una cache con:
  - 4K blocchi
  - ogni blocco ha 4 parole
  - indirizzi di 32 bit

Set associativa a 4 vie:

$$4K = 2^{12} \text{ blocchi} \Rightarrow 2^{12}/4 = 2^{10} \text{ insiemi}$$



$$\Rightarrow 18 \text{ bit} * 2^{12} = 18 * 4 \text{ kbit} = 72 \text{ kbit}$$

## Esercizio 2

- Sia data una cache con:
  - 4K blocchi
  - ogni blocco ha 4 parole
  - indirizzi di 32 bit

Completamente associativa:

$$4K = 2^{12} \text{ blocchi}$$



↓

$$32 - 4 = 28$$

⇒  $28 \text{ bit} * 2^{12} = 28 * 4 \text{ kbit} = 112 \text{ kbit}$

### Esercizio 3

- Si consideri una CPU che esegue tutte le istruzioni in 2 cicli di clock.
- Si assuma il seguente carico di lavoro:
  - Tipo-R            50 %
  - lw                 20 %
  - sw                 16 %
  - salti               14 %
- Si dispone di una cache con le seguenti caratteristiche:
  - $f_{\text{miss}}$  per le istruzioni: 2%
  - $f_{\text{miss}}$  per accesso ai dati: 4%
  - $P_{\text{miss}} = 100$  cicli di clock
- Confrontare le prestazioni tra il caso reale ed il caso di cache ideale senza miss

## Caso reale

$$\begin{aligned} \text{CPI} &= 2 + 0.02 * 100 + 0.36 * 0.04 * 100 = 2 + 2 + 1.44 \\ &= 5.44 \end{aligned}$$

## Caso ideale

$$\text{CPI} = 2$$

⇒ Dato che nei due casi  $T_{\text{clock}}$  non cambia,  
nel caso ideale il processore sarebbe  
 $5.44/2 = 2.72$  volte più veloce

### Esercizio 3 (continua)

- Si consideri una CPU che esegue tutte le istruzioni in 2 cicli di clock.
- Si assuma il seguente carico di lavoro:
  - Tipo-R            50 %
  - lw                 20 %
  - sw                 16 %
  - salti              14 %
- Si dispone di una cache con le seguenti caratteristiche:
  - $f_{\text{miss}}$  per le istruzioni: 2%
  - $f_{\text{miss}}$  per accesso ai dati: 4%
  - $P_{\text{miss}} = 100$  cicli di clock
- Si supponga raddoppiare la frequenza del processore:  
quale miglioramento in termini di prestazione si ottiene?

Nel caso precedente:

$$T_{es} = I * 5.44 * T_{clock}$$

Raddoppiando la frequenza:

$P_{miss}$  uguale in termini di tempo  $\Rightarrow$  raddoppia in termini di cicli

$$\begin{aligned} CPI &= 2 + 0.02*200 + 0.36*0.04*200 = 2 + 4 + 2.88 \\ &= 8.88 \end{aligned}$$

$$T_{es} = I * 8,88 * T'_{clock}$$

$$\Rightarrow \frac{5.44*2}{8.88} = 1.23 \text{ volte più veloce}$$

## Commento

Quanto più il processore è veloce ( $T_{\text{clock}}$  o CPI bassi)  
tanto maggiore è l'importanza della cache

## Esercizio 4

Si consideri un processore con  $CPI=1$  e frequenza di 5GHz.

Si dispone di una cache primaria con  $f_{\text{miss}} = 2\%$  e di una memoria DRAM con un tempo di accesso di 100ns.

Determinare quale sarebbe l'incremento di prestazioni aggiungendo una cache secondaria con un tempo di accesso di 5ns e  $f_{\text{miss}} = 25\%$

## Esercizio 4

Si consideri un processore con  $CPI=1$  e frequenza di 5GHz.

Si dispone di una cache primaria con  $f_{\text{miss}} = 2\%$  e di una memoria DRAM con un tempo di accesso di 100ns.

Determinare quale sarebbe l'incremento di prestazioni aggiungendo una cache secondaria con un tempo di accesso di 5ns e  $f_{\text{miss}} = 25\%$

### Solo cache L1

$$T_{\text{clock}} = 1/5 * 10^9 \text{ ns} = 0.2 \text{ ns}$$

$$\Rightarrow P_{\text{miss}} = 100/0.2 = 500 \text{ cicli di clock}$$

$$CPI = 1 + 0.02 * 500 = 11$$

## Esercizio 4

Si consideri un processore con CPI=1 e frequenza di 5GHz.

Si dispone di una cache primaria con  $f_{\text{miss}} = 2\%$  e di una memoria DRAM con un tempo di accesso di 100ns.

Determinare quale sarebbe l'incremento di prestazioni aggiungendo una cache secondaria con un tempo di accesso di 5ns e  $f_{\text{miss}} = 25\%$

### Solo cache L1

$$T_{\text{clock}} = 1/5 * 10^9 \text{ ns} = 0.2 \text{ ns}$$

$$\Rightarrow P_{\text{miss}} = 100/0.2 = 500 \text{ cicli di clock}$$

$$\text{CPI} = 1 + 0.02 * 500 = 11$$

### Cache L1 + L2

Miss in cache primaria, hit in secondaria:  $5/0.2=25$  cicli

$$\text{CPI} = 1 + 0.02 * (0.75 * 25 + 0.25 * 525) = 4$$

## Esercizio 4

Si consideri un processore con CPI=1 e frequenza di 5GHz.

Si dispone di una cache primaria con  $f_{\text{miss}} = 2\%$  e di una memoria DRAM con un tempo di accesso di 100ns.

Determinare quale sarebbe l'incremento di prestazioni aggiungendo una cache secondaria con un tempo di accesso di 5ns e  $f_{\text{miss}} = 25\%$

### Solo cache L1

$$T_{\text{clock}} = 1/5 \cdot 10^9 \text{ ns} = 0.2 \text{ ns}$$

$$\Rightarrow P_{\text{miss}} = 100/0.2 = 500 \text{ cicli di clock}$$

$$\text{CPI} = 1 + 0.02 \cdot 500 = 11$$

11/4=2.8 volte più veloce

### Cache L1 + L2

Miss in cache primaria, hit in secondaria:  $5/0.2=25$  cicli

$$\text{CPI} = 1 + 0.02 \cdot (0.75 \cdot 25 + 0.25 \cdot 525) = 4$$

NB: si noti che nonostante il local miss rate della cache L2 sia del 25%, il global miss rate risultante è  $2\% * 25\% = 0.5\%$