



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Architettura degli Elaboratori 20 – Parallelismo e cache multiple

Prof. Andrea Sterbini – sterbini@di.uniroma1.it



Argomenti

- Come progettare il controllore di una cache
- Sistemi multi-processore e cache multiple in parallelo
- Coerenza dei dati e protocolli di *snooping*

► Problema:

- **In un sistema multi-processore si usano cache multiple in parallelo**
- **Processi diversi possono contemporaneamente accedere/modificare gli stessi dati**
- **Bisogna mantenere «coerenti» le informazioni tra i diversi processi**

Argomenti

- Come progettare il controllore di una cache
- Sistemi multi-processore e cache multiple in parallelo
- Coerenza dei dati e protocolli di *snooping*

► Problema:

- **In un sistema multi-processore si usano cache multiple in parallelo**
- **Processi diversi possono contemporaneamente accedere/modificare gli stessi dati**
- **Bisogna mantenere «coerenti» le informazioni tra i diversi processi**

► IDEA:

- | Collegare tra loro le cache in modo che ciascuna «tenga d'occhio» cosa fanno le altre
(oppure usare un meccanismo centralizzato di coordinamento (directory))

Argomenti

- Come progettare il controllore di una cache
- Sistemi multi-processore e cache multiple in parallelo
- Coerenza dei dati e protocolli di *snooping*

► Problema:

- **In un sistema multi-processore si usano cache multiple in parallelo**
- **Processi diversi possono contemporaneamente accedere/modificare gli stessi dati**
- **Bisogna mantenere «coerenti» le informazioni tra i diversi processi**

► IDEA:

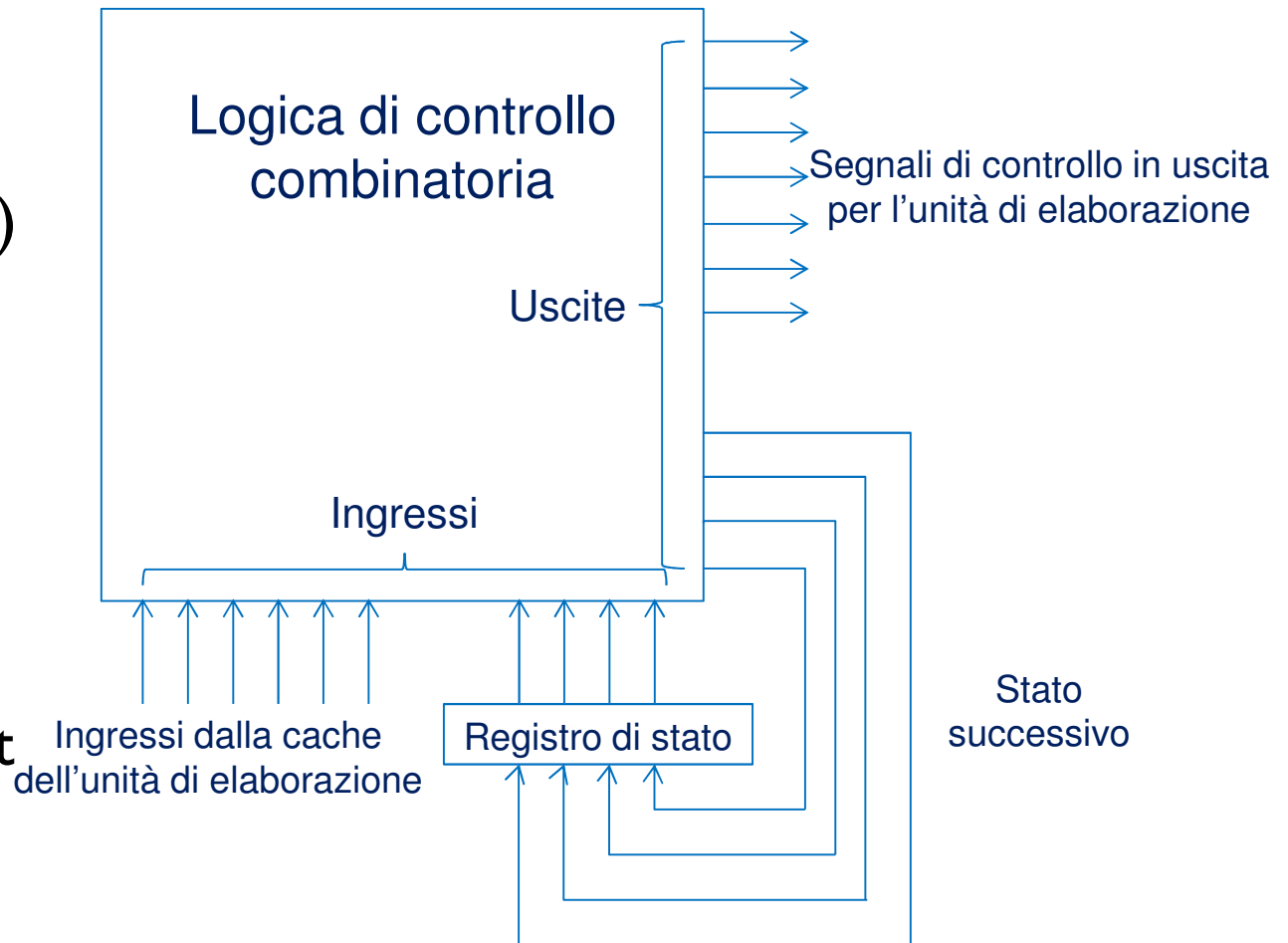
- 1 Collegare tra loro le cache in modo che ciascuna «tenga d'occhio» cosa fanno le altre
(oppure usare un meccanismo centralizzato di coordinamento (directory))
- 2 Permettere la replicazione e migrazione di informazioni tra le cache

Il controllore di una cache

- ▶ Per progettare un controllore di cache possiamo usare un Automa a Stati Finiti (FSA)

- ▶ **Dobbiamo definire:**

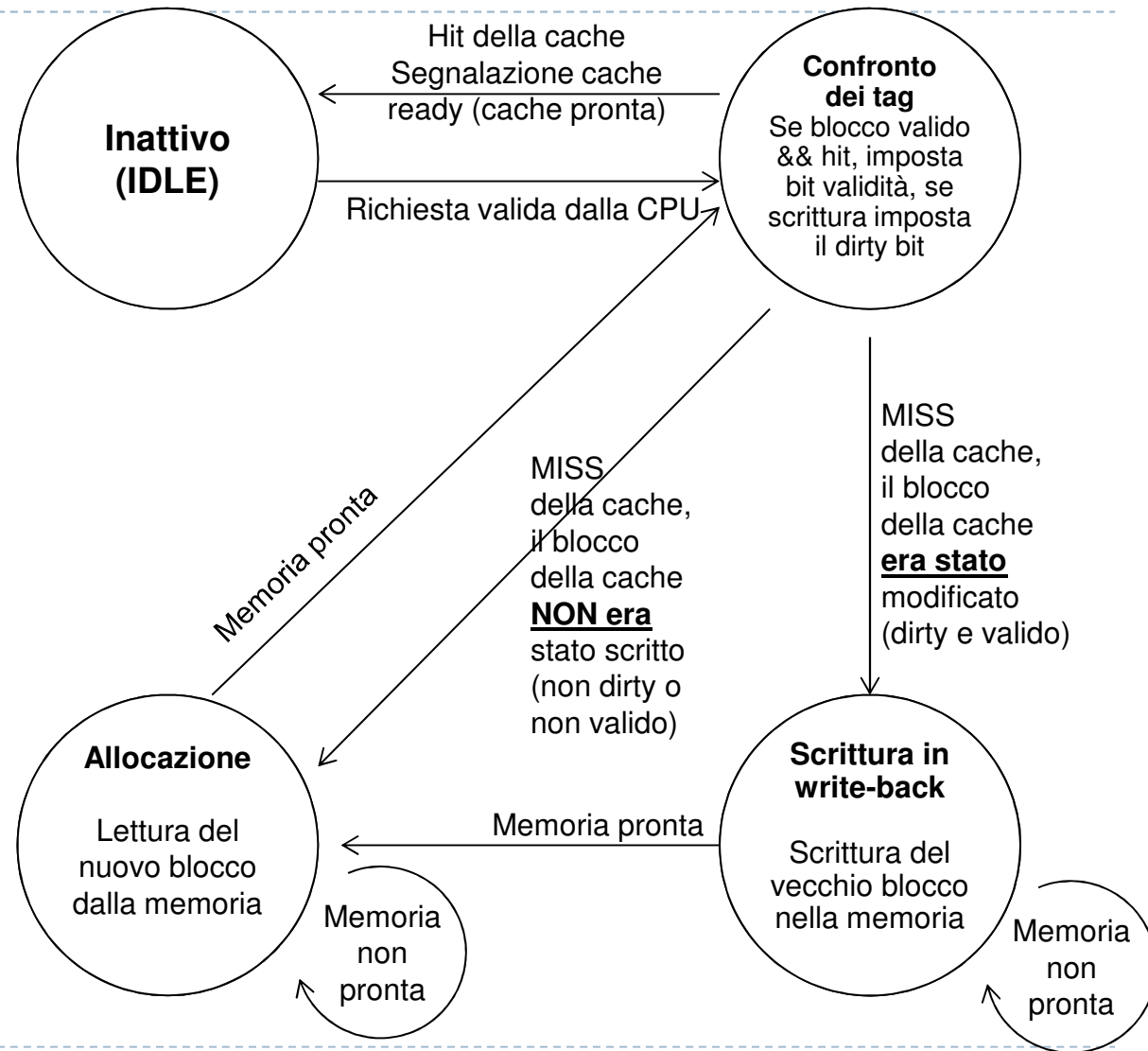
- Gli stati possibili
- I possibili input
- Le transizioni
- I segnali di output
- Come codificare stati e segnali



Il FSA del controllore di cache

► **Sono necessari 4 stati:**

- Attesa della richiesta (**idle**)
- Ricerca del blocco in cache (**confronto dei tag**)
- Lettura del blocco dalla memoria (**allocazione**)
- Scrittura del blocco in memoria (**write-back**)



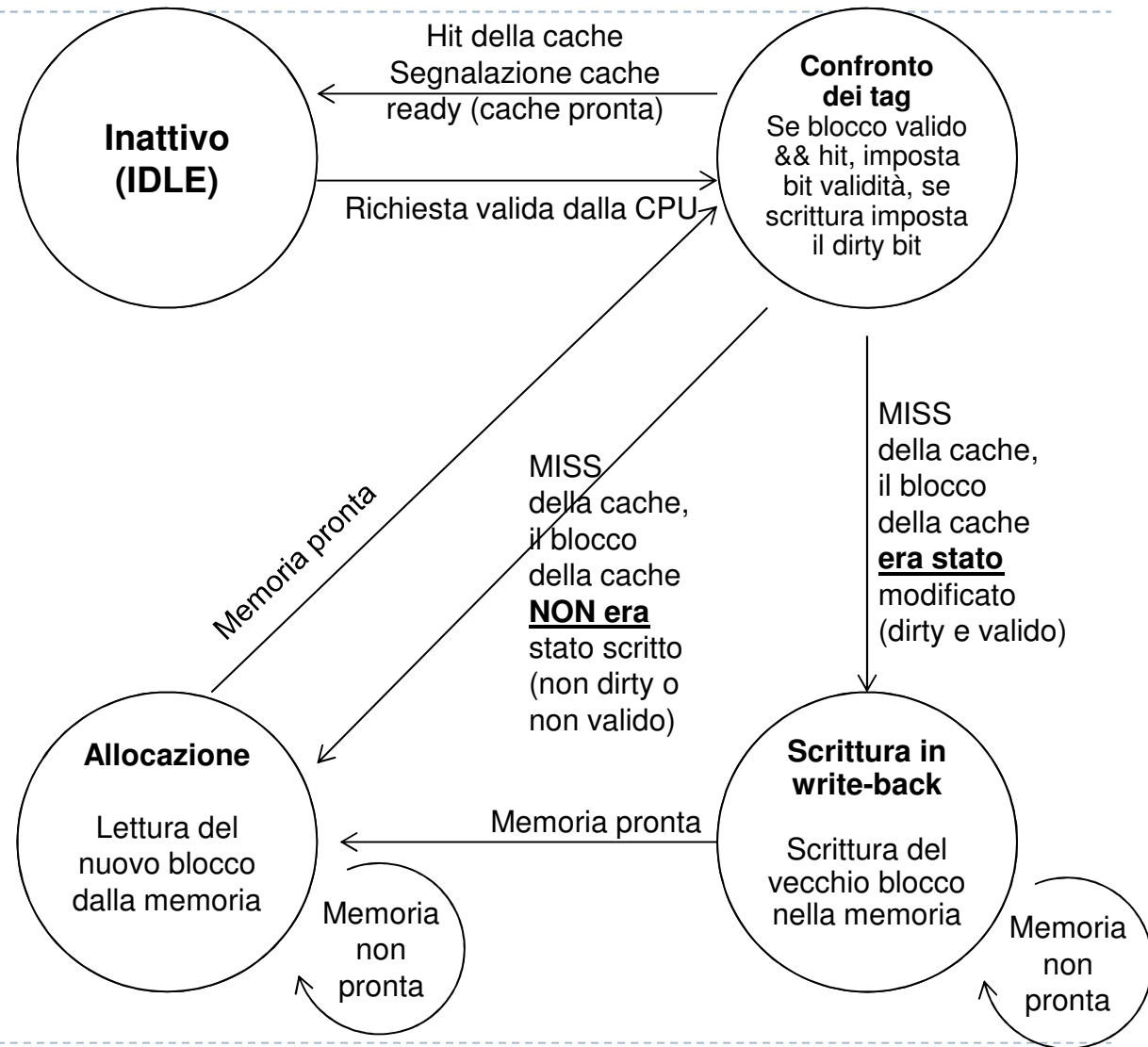
Il FSA del controllore di cache

► Sono necessari 4 stati:

- Attesa della richiesta (**idle**)
- Ricerca del blocco in cache (**confronto dei tag**)
- Lettura del blocco dalla memoria (**allocazione**)
- Scrittura del blocco in memoria (**write-back**)

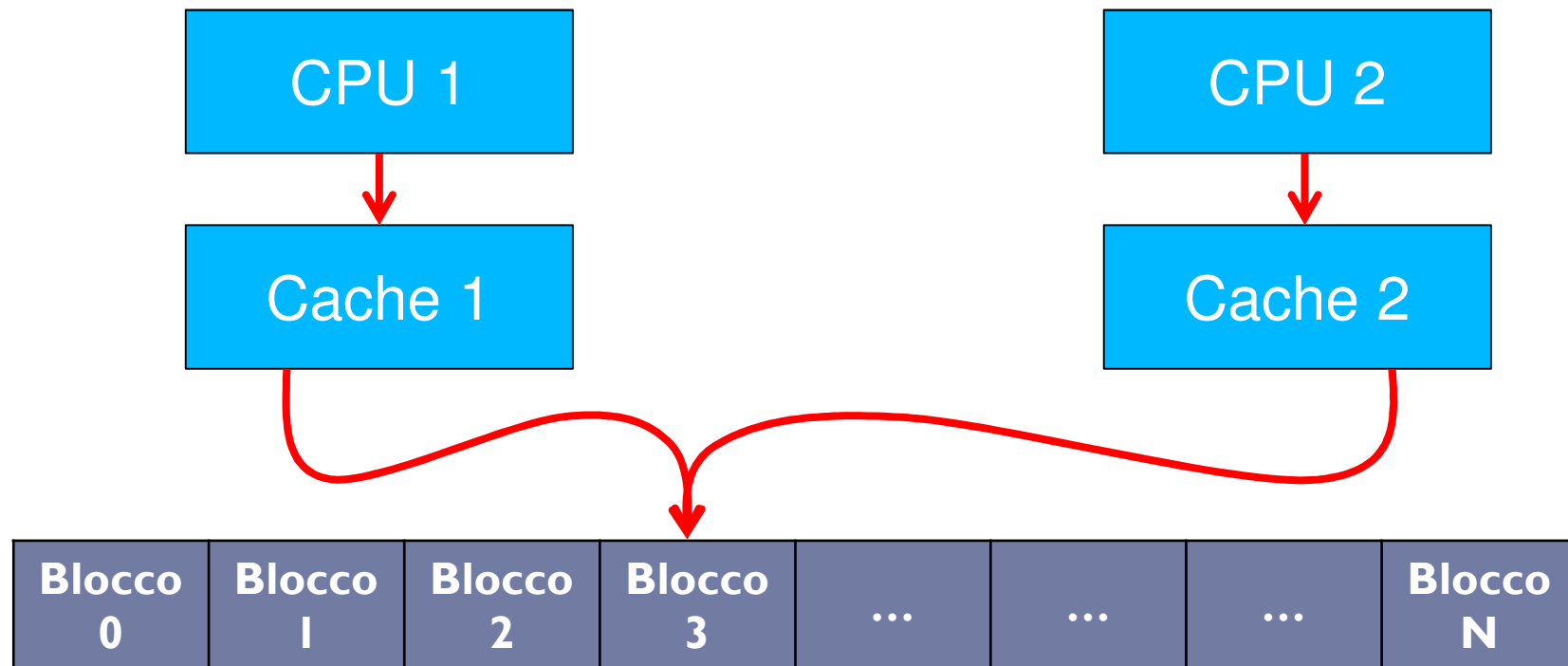
► Miglioramenti:

- Bufferizzare la scrittura per leggere subito il nuovo blocco e non far attendere la CPU
- Separare confronto dei tag dalla scrittura in cache x diminuire il clock



Parallelismo e cache multiple

- ▶ Più processori che condividono la stessa memoria fisica (p.es. nei sistemi multi-core)
- ▶ possono accedere contemporaneamente agli stessi dati tramite cache diverse (quando lo stesso blocco è presente in due copie diverse in due cache separate)



Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0				0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0				0
1	CPU A legge X	0		0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0		0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0		0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	← 0 ←	←	0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	←	0
2	CPU B legge X	0	0	0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	MISS	0
2	CPU B legge X	0	0	0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	MISS	0
2	CPU B legge X	0	0	0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	← 0 ←	← MISS ←	0
2	CPU B legge X	← 0 ←	← 0 ←	0

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	MISS	0
2	CPU B legge X	0	0	0
3	CPU A scrive 1 in X	1	0	1

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	← 0	← MISS	0
2	CPU B legge X	← 0	← 0	← 0
3	CPU A scrive 1 in X	→ 1	0	1

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	← 0 ←	← MISS ←	0
2	CPU B legge X	← 0 ←	← 0 ←	0
3	CPU A scrive 1 in X	→ 1 →	→ 0 →	1

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	MISS	0
2	CPU B legge X	0	0	0
3	CPU A scrive 1 in X	1	0	1
4	CPU B legge X	1	0	1

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	MISS	0
2	CPU B legge X	0	0	0
3	CPU A scrive 1 in X	1	0	1
4	CPU B legge X	1	0	1

Coerenza delle cache

- ▶ **Informalmente:** «un sistema di memoria è **coerente** se la lettura di un dato restituisce il valore che è stato scritto in quella memoria più di recente»
- ▶ In realtà le cose sono più complesse:
 - La **coerenza** indica quale valore deve essere ritornato da una lettura
 - La **consistenza** indica quando, fatta una scrittura, il dato sarà finalmente disponibile
- ▶ **Sequenza di accessi alla stessa locazione X da due processi diversi con due cache separate**

Passo	Evento	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
0		MISS		0
1	CPU A legge X	0	MISS	0
2	CPU B legge X	0	0	0
3	CPU A scrive 1 in X	1	0	1
4	CPU B legge X	1	0	1

Garantire la coerenza

- ▶ Più precisamente la coerenza è data da:
 - Se P1 legge da X dopo aver scritto in X, e senza che altri scrivano in X, otterrà lo stesso valore
 - Se P2 legge X dopo che P1 ci ha scritto, il valore letto sarà quello scritto da P1 se è passato abbastanza tempo e nessun altro processore ha scritto in X
 - Le scritture nella stessa posizione X sono serializzate, quindi due scritture fatte da processori diversi nella stessa X vengono viste da tutti i processori nello stesso ordine
- ▶ Per garantire la coerenza si usano due meccanismi:
 - **Migrazione dei dati:** un dato può essere trasferito da una cache all'altra
 - **Replicazione dei dati:** quando un dato condiviso viene letto da più processori viene replicato nelle corrispondenti cache
- ▶ **Punto chiave:** lo stato di condivisione dei blocchi nelle diverse cache deve essere tracciato
 - ▶ (meglio se da tutte le cache per usare un algoritmo distribuito)

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
				0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
				0
CPU A legge X	MISS cache per X	0		0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0		0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0 ←		0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	← MISS cache per X	0 ←		0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0		0
CPU B legge X	MISS cache per X	0	0	0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	← MISS cache per X	0	← MISS	0
CPU B legge X	← MISS cache per X	0	0	0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	← MISS cache per X	0	← MISS	0
CPU B legge X	← MISS cache per X	0	0	0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive I in X	Invalidazione di X	I		0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive I in X	Invalidazione di X	I		0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive I in X	Invalidazione di X	I	X	0

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive 1 in X	Invalidazione di X	1	X	0
CPU B legge X	MISS cache per X	1	1	1

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive I in X	Invalidazione di X	I	MISS	0
CPU B legge X	MISS cache per X	I	I	I

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive I in X	Invalidazione di X	I	MISS	0
CPU B legge X	MISS cache per X	I	I	I

Protocollo di «snooping»

- Ogni cache che contiene un blocco ne ha anche lo stato di condivisione nelle altre cache
- Le cache sono collegate da un bus o una rete che permette a ciascuna di osservare («snooping») le altre per aggiornare lo stato di condivisione del blocco
- **E' sufficiente la trasmissione delle MISS da una cache a tutte le altre e le scritture**
 - Per far **migrare il blocco** se necessario o per **invalidare un blocco** divenuto obsoleto
 - **Purtroppo:** la rete di interconnessione è difficilmente «scalabile» → (meccanismi a «directory»)

► Esempio di invalidazione con una cache di tipo write-back

Attività del processore	Attività del bus	Contenuto della cache della CPU A	Contenuto della cache della CPU B	Contenuto della memoria nella locazione X
		MISS		0
CPU A legge X	MISS cache per X	0	MISS	0
CPU B legge X	MISS cache per X	0	0	0
CPU A scrive I in X	Invalidazione di X	I	MISS	0
CPU B legge X	MISS cache per X	I	I	I

Consistenza della memoria

- Una scrittura non è considerata completa finché tutti i processori ne vedono l'effetto
- Un processore non deve cambiare l'ordine delle scritture
 - Ovvero se P1 prima scrive in X e poi in Y, gli altri processori che leggono Y devono poter sicuramente ottenere anche X
- Quindi i processori possono riordinare le letture (in ordine diverso)
- ma devono completare le scritture nell'ordine giusto