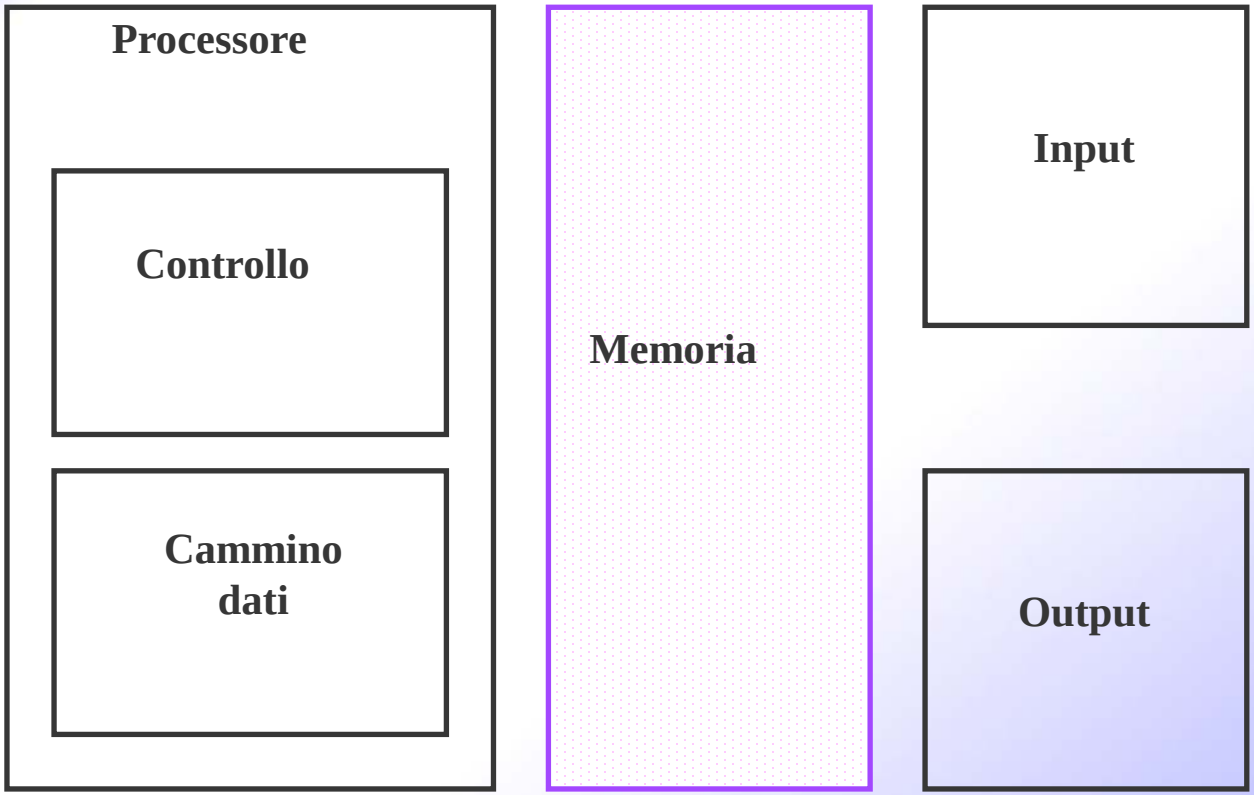
Capacità e velocità: sfruttare la gerarchia delle memorie

Il punto

- Le cinque componenti classiche di un computer



Memorie

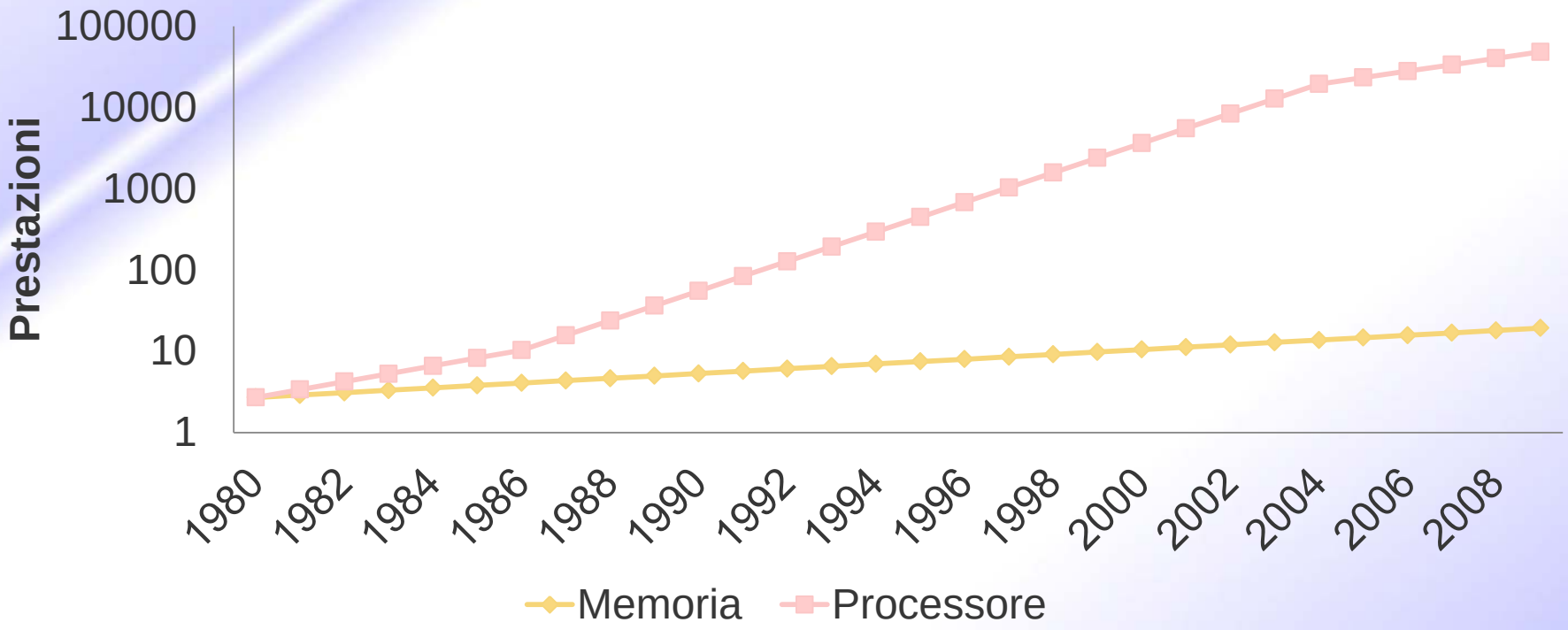
- SRAM (static random access memory):
  - valore memorizzato su una coppia di circuiti invertitori
  - molto veloce ma richiede più spazio
    - tempo di accesso: 0,5-5ns
    - costo per GByte: 4.000-10.000 $
- DRAM (dynamic random access memory):
  - valore memorizzato come carica di un condensatore
  - molto piccola ma più lenta
    - tempo di accesso: 50-70ns
    - costo per GByte: 100-200 $
- Dischi magnetici
  - molto grande ma molto lenta
    - tempo di accesso: 5.000.000-20.000.000ns
    - costo per GByte: 0,5- 2 $

Evoluzione delle DRAM

- 1980 - 1998 capacità quadruplicata ogni 3 anni.
- 1998 - 2006 capacità raddoppiata ogni 2 anni
- Oggi il ritmo di crescita sembra rallentare ulteriormente.
- Il periodo di ciclo si è ridotto di 4 volte in 26 anni

Anno di Introduzione	Dimensione del Chip	Periodo di Ciclo
1980	64K bit	250 ns
1983	256K bit	220 ns
1986	1M bit	190 ns
1989	4M bit	165 ns
1992	16M bit	120 ns
1996	64M bit	110 ns
1998	128M bit	100 ns
2000	256M bit	90 ns
2002	512M bit	80 ns
2004	1G bit	70 ns
2006	2G bit	60 ns

Gap processore/memoria

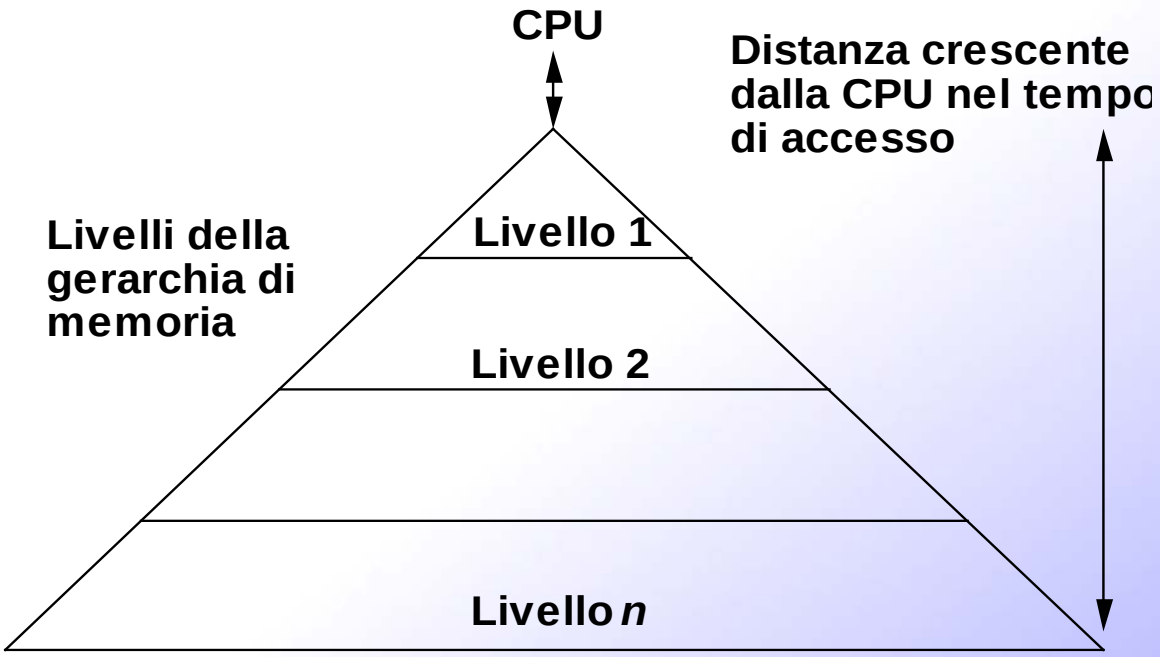


Legge di Moore: *Le prestazioni dei processori, e il numero di transistor ad esso relativo, raddoppiano ogni 18 mesi.*

I progettisti di architetture per calcolatori devono colmare il divario tra i processori e le memorie.

Obiettivo

- Fatto: memorie grandi sono lente, memorie veloci sono piccole
- Scopo: creare una memoria **grande, economica e veloce** (il più delle volte)
- Soluzione: gerarchia di memoria



Principio di località

- Un principio che rende l'uso della gerarchia di memoria una buona idea
- Se si accede ad una locazione di memoria:
  - allora è molto probabile che vi si acceda di nuovo entro breve tempo (*località temporale*)
  - allora è molto probabile che si acceda alle locazioni vicine ad essa entro breve tempo (*località spaziale*)
- Ci concentreremo su due livelli di gerarchia della memoria:
  - superiore ed inferiore

Terminologia

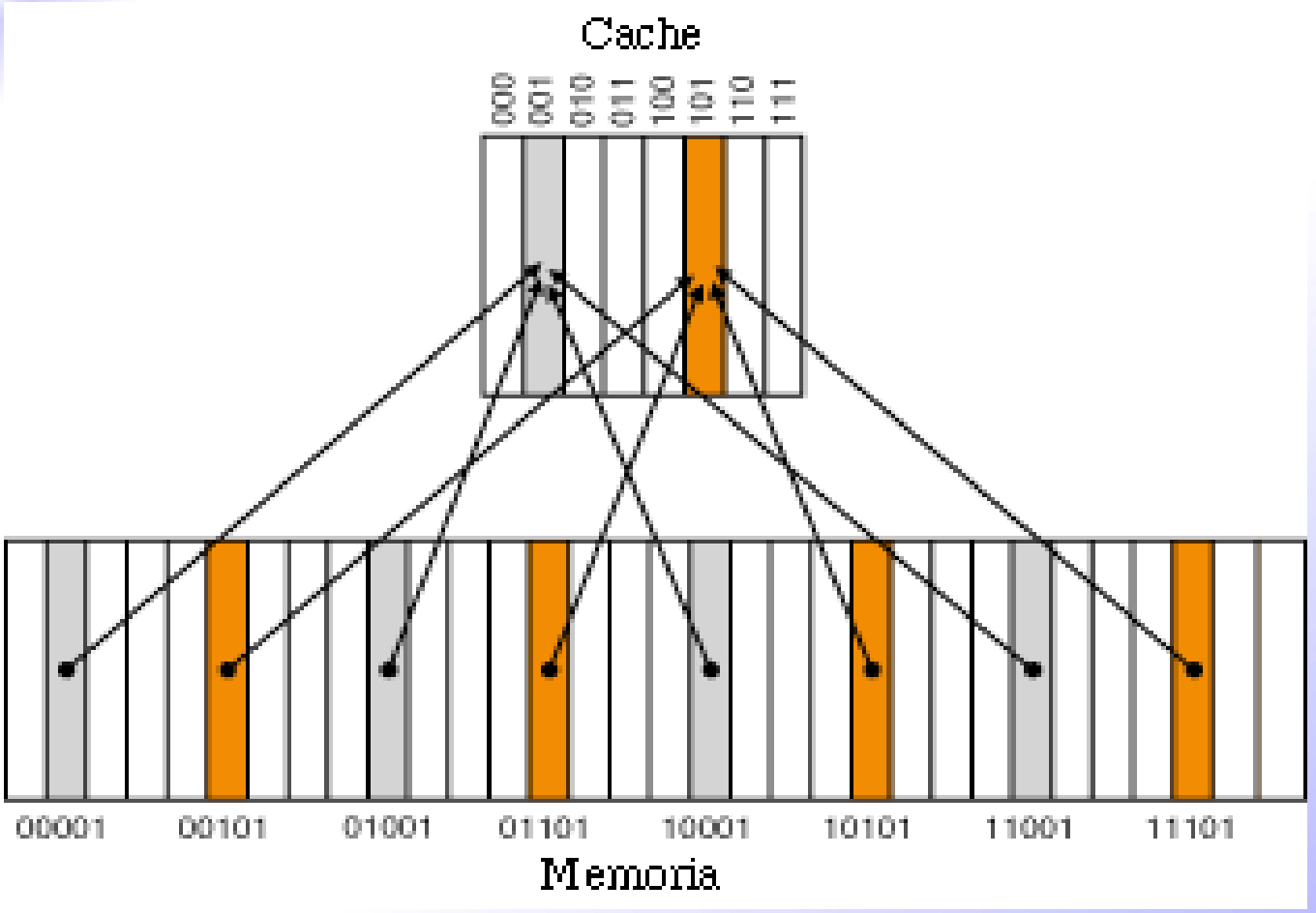
- Blocco: minima unità di dati
- **Successo**: il dato richiesto si trova nel livello superiore
  - tasso di successo = numero di successi / numero di accessi
  - tempo di successo = tempo di accesso alla memoria del livello superiore + tempo per decidere il successo
- **Fallimento**: il dato richiesto **non** si trova nel livello superiore
  - tasso di fallimento = 1 - numero di successi / numero di accessi = 1 – tasso di successo
- penalità di fallimento = tempo per sostituire un blocco nel livello superiore + tempo per restituire il dato al processore

Memoria cache

- Memoria tra la CPU e la memoria principale
  - in generale, ogni memoria che sfrutta il principio di località
  - dal francese: luogo segreto adatto a nascondere
- Due problemi:
  - **Come facciamo a sapere se un dato è in memoria cache?**
  - **Se c'è, dove lo troviamo?**
- Primo esempio:
  - dimensione del blocco = una parola
  - mappa diretta:
    - ad ogni locazione di memoria è associata un'unica posizione nella cache

Cache a mappa diretta

- Associazione: indirizzo modulo il numero di blocchi nella cache



Accesso alla cache

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 22: Il dato non è presente in cache.
FALLIMENTO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

Accesso alla cache

Gestione del FALLIMENTO e scrittura del dato in cache

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 26: Il dato non è presente in cache.
FALLIMENTO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Gestione del FALLIMENTO e scrittura del dato in cache

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	S	11	Mem[26]
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 22: Il dato è presente in cache.
SUCCESSO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	S	11	Mem[26]
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 26: Il dato è presente in cache.
SUCCESSO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	S	11	Mem[26]
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 16: Il dato non è presente in cache.
FALLIMENTO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000

Cache

Indice	V	Etic.	Dati
000	N		
001	N		
010	S	11	Mem[26]
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Gestione del FALLIMENTO e scrittura del dato in cache

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	11	Mem[26]
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 3: Il dato non è presente in cache.
FALLIMENTO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	11	Mem[26]
011	N		
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Gestione del FALLIMENTO e scrittura del dato in cache

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	11	Mem[26]
011	S	00	Mem[3]
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 16: Il dato è presente in cache.
SUCCESSO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011
16	10000	S	000

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	11	Mem[26]
011	S	00	Mem[3]
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 18: c'è un dato in cache al posto con indice 010 ma l'etichetta non coincide, non è il dato giusto: FALLIMENTO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011
16	10000	S	000
18	10010	F	010

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	11	Mem[26]
011	S	00	Mem[3]
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Gestione del FALLIMENTO e scrittura del dato in cache

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011
16	10000	S	000
18	10010	F	010

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	10	Mem[18]
011	S	00	Mem[3]
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Richiesta di lettura indirizzo 26: c'è un dato in cache alla posto con indice 010 ma l'etichetta non coincide, non è il dato giusto: FALLIMENTO

Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011
16	10000	S	000
18	10010	F	010
26	11010	F	010

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	10	Mem[18]
011	S	00	Mem[3]
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accesso alla cache

Gestione del FALLIMENTO e scrittura del dato in cache

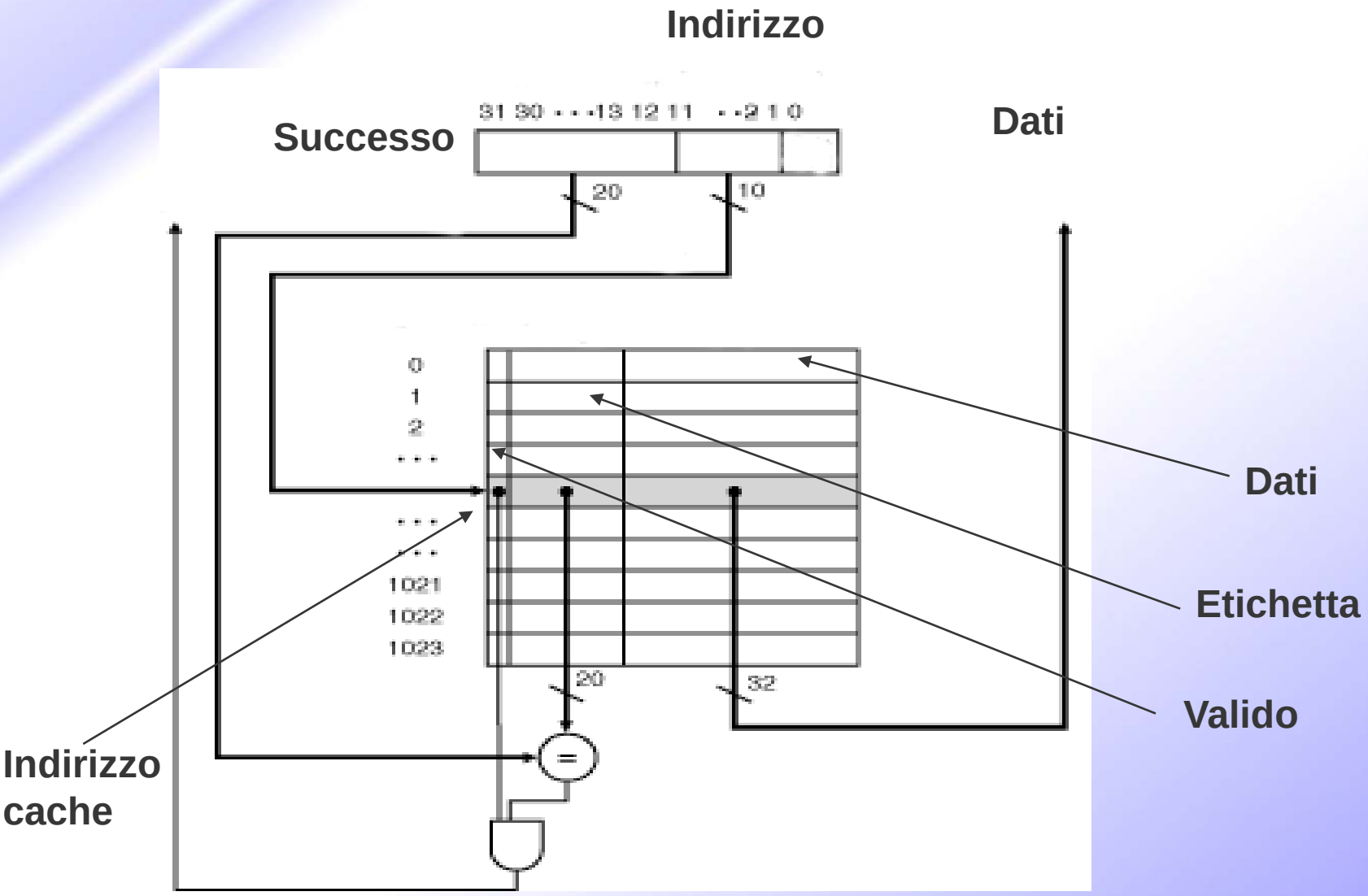
Seq. indirizzi

Ind.	Ind. bin.	F/S	Blocco
22	10110	F	110
26	11010	F	010
22	10110	S	110
26	11010	S	010
16	10000	F	000
3	00011	F	011
16	10000	S	000
18	10010	F	010
26	11010	F	010

Cache

Indice	V	Etic.	Dati
000	S	10	Mem[16]
001	N		
010	S	11	Mem[26]
011	S	00	Mem[3]
100	N		
101	N		
110	S	10	Mem[22]
111	N		

Accedere alla cache



Numero di bit di cache

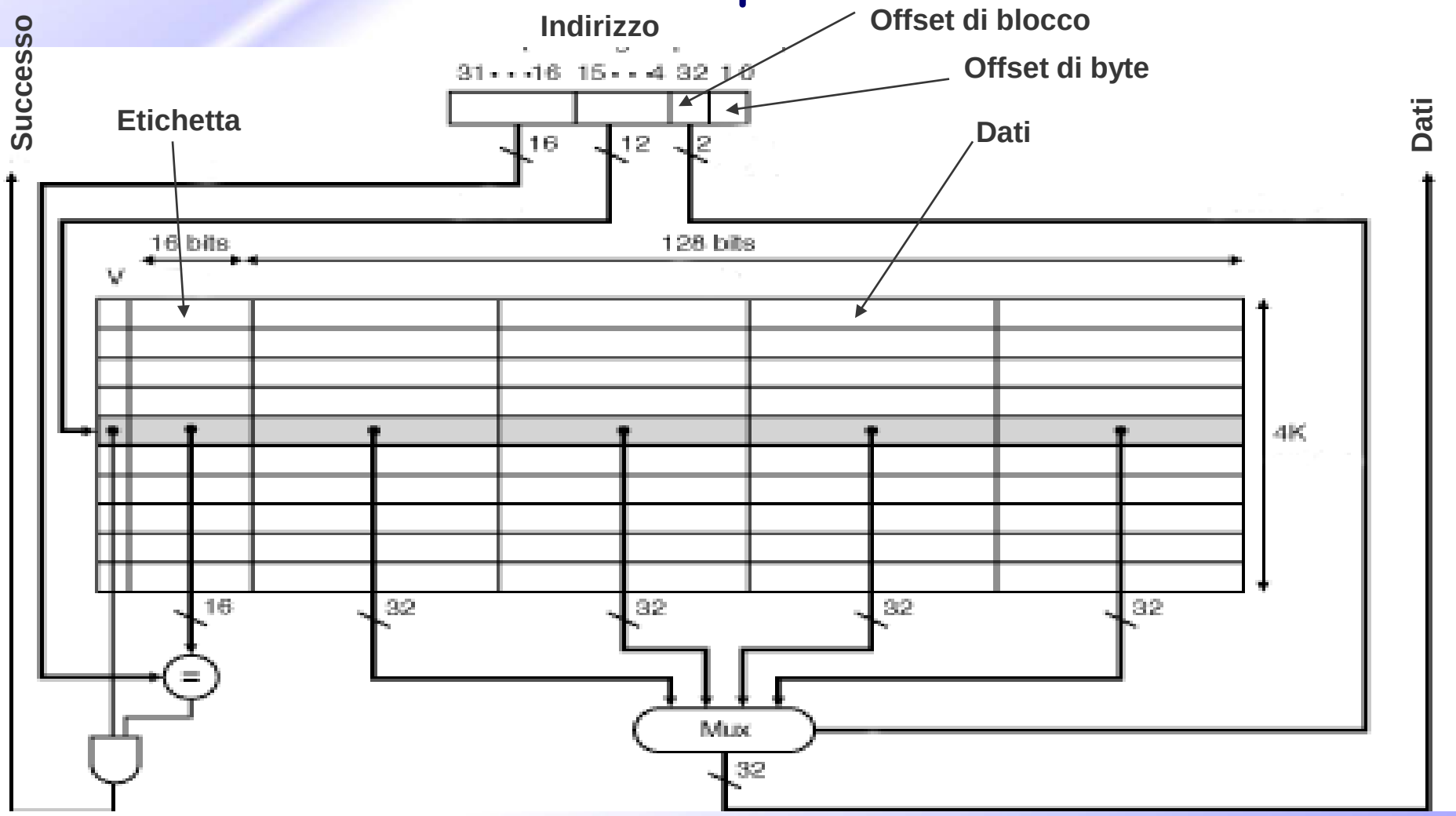
- Assumiamo:
  - indirizzo a m bit
  - 2^n parole (o blocchi) di m bit in cache
- Campo etichetta:
  - m-(n+2) bit
- Numero di bit totale:
  - 2^n x (dim. blocco+dim.etichetta+dim.valido) = 2^n x (m+(m-(n+2))+1) = 2^n x(2m-n-1)

Successi e fallimenti

- Successi di lettura
  - questo è quello che vogliamo
- Fallimenti di lettura
  - CPU in stallo (cioe' ferma!!), carica il blocco dalla memoria, invia alla cache, riparte
- Successo di scrittura:
  - *write-through*: sostituire il dato in cache ed in memoria
  - *write buffer*: scrivere il dato in una memoria tampone in attesa che venga scritto in memoria
  - *write-back*: scrivere il dato solo nella cache (scarica la cache successivamente quando il blocco va sostituito)
- Fallimenti di scrittura:
  - legge il blocco in memoria, invia alla cache, quindi scrive il dato

Blocchi di più parole

➢ Sfruttare la località spaziale



Calcolo indirizzo di cache

- **Indirizzo di blocco:** $\left\lfloor \frac{Ind.\ byte_{mem}}{byte\ \ per\ blocco} \right\rfloor$

Blocco cache: $(Ind.\ di\ blocco)\ \mathrm{mod}\ (n°\ \ blocchi\ in\ cache)$

- Il Blocco cache include tutti gli indirizzi di memoria compresi tra

$$\left\lfloor \frac{Ind.\ byte_{mem}}{byte\ \ per\ blocco} \right\rfloor \cdot byte\ \ per\ blocco \qquad \text{e}$$

$$\left\lfloor \frac{Ind.\ byte_{mem}}{byte\ \ per\ blocco} \right\rfloor \cdot byte\ \ per\ blocco + (byte\ \ per\ blocco - 1)$$

- **Offset di blocco:** $\left\lfloor (Ind.\ byte_{mem}) \bmod (byte\ \ per\ blocco) \Big/ 4 \right\rfloor$

Esempio

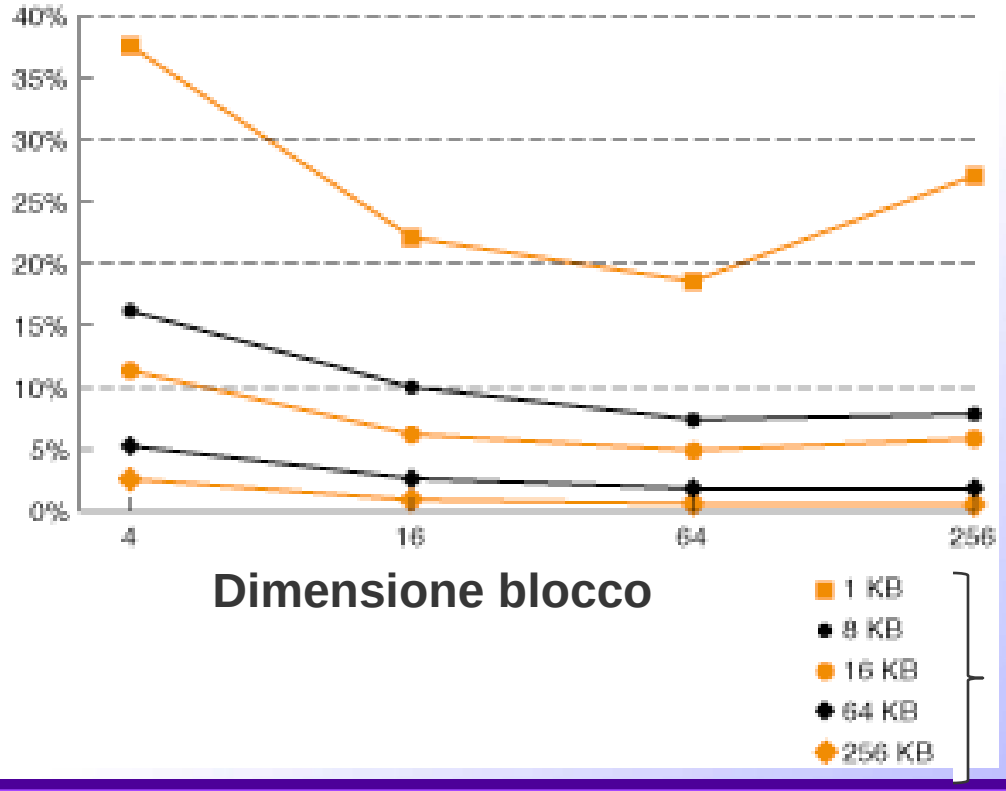
- Cache con 64 blocchi, 16 byte per blocco
- Indirizzo byte in memoria 1208: 10010111000
- Indirizzo di blocco: $\lfloor 1208/16 \rfloor = 75$
- Blocco cache: $10010111000 >> 4 = 1001011$

  $75 \bmod 64 = 11 \quad 1[001011] = 1011$
- Il blocco 11 include tutti gli indirizzi compresi tra 1200 e 1215
- Offset di blocco: $\lfloor (1208 \bmod 16)/4 \rfloor = \lfloor 8/4 \rfloor = 2$

  $(1001011[1000]) >> 2 = 1000 >> 2 = 10$

Prestazioni

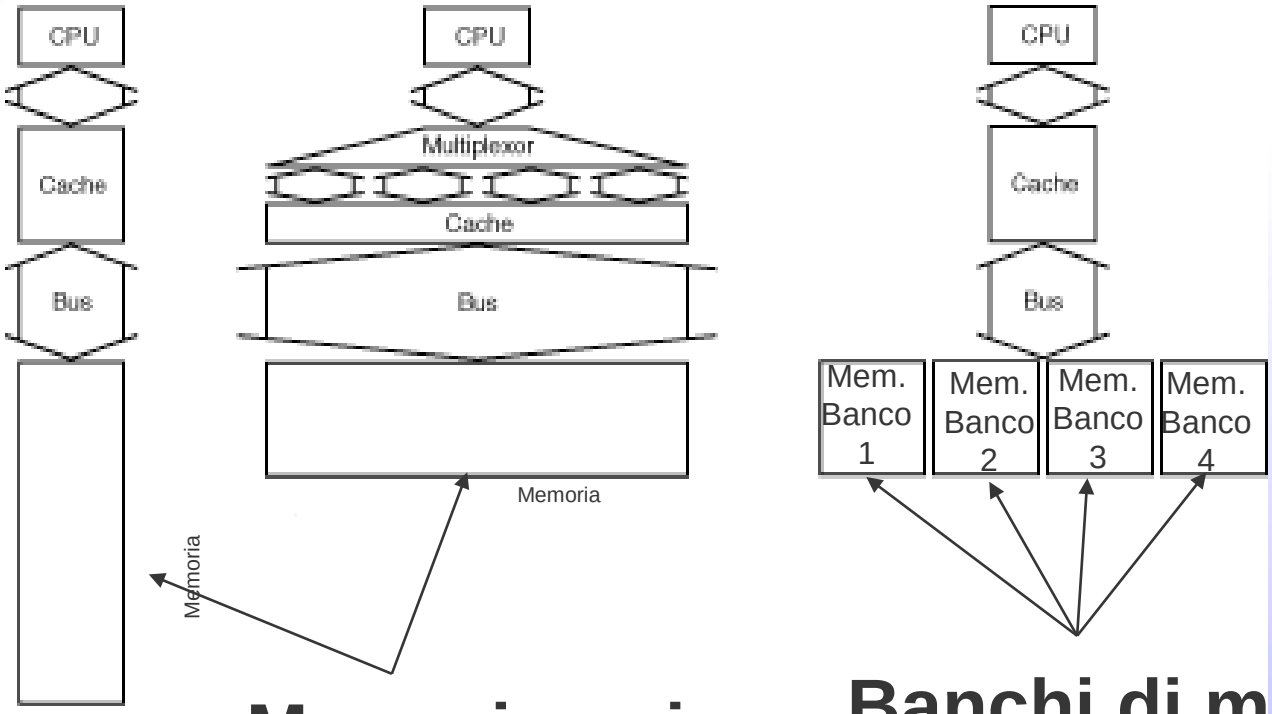
Programma	Dimensione blocco	Tasso di fallimento istruzione	Tasso di fallimento dati	Effettivo tasso di fallimento combinato
gcc	1	6.1%	2.1%	5.4%
	4	2.0%	1.7%	1.9%



Migliorare le prestazioni

Supponiamo di leggere blocchi da 4 parole

Si può rendere più efficiente la cache con blocchi a più parole usando bande maggiori oppure banchi di memoria



Memoria unica **Banchi di memoria**

Confronto di prestazioni

- Assumiamo:
  - 1 ciclo di clock per inviare l'indirizzo
  - 15 cicli di clock per ogni accesso a DRAM
  - 1 ciclo di clock per inviare la parola

blocco di 4 parole

- Memoria unica a banda piccola:
  - 1+4x15+4x1 = 65 cicli (0.25 byte per ciclo)
- Memoria unica a larga banda (due parole):
  - 1+2x15+2x1 = 33 cicli (0.48 byte per ciclo)
- Memoria unica a larga banda (quattro parole):
  - 1+1x15+1x1 = 17 cicli (0.94 byte per ciclo)
- Memoria a banchi:
  - 1+1x15+4x1 = 20 cicli (0.80 byte per ciclo)

Misurare le prestazioni

- Modello semplificato:
- tempoEsecuzione = (cicliEsecuzione + cicliStallo)*tempoCiclo
- cicliStallo=cicliStalloIstr+cicliStalloDati
- cicliStalloIstr=percFallimentoIstr*penalitàFallimento
- cicliStalloDati=percIstrMemoria*percFallimentoDati*penalitàFallimento

- Esempio:
- I=n° istruzioni, CPI=2, penalitàFallimento=40 cicli, fallimentoIstruzioni=2%, percFallimentoDati = 4%, percIstrMemoria=36%
- tempoEsecuzioneIdeale = I x CPI x tempoCiclo
- tempoEsecuzioneReale = I x CPIReale x tempoCiclo

CPIReale = CPI + (0.02 x 40 + 0.36 x 0.04 x 40) = 2+1.36 = 3.36

- rapporto reale/ideale = 3.36/2 = 1.68

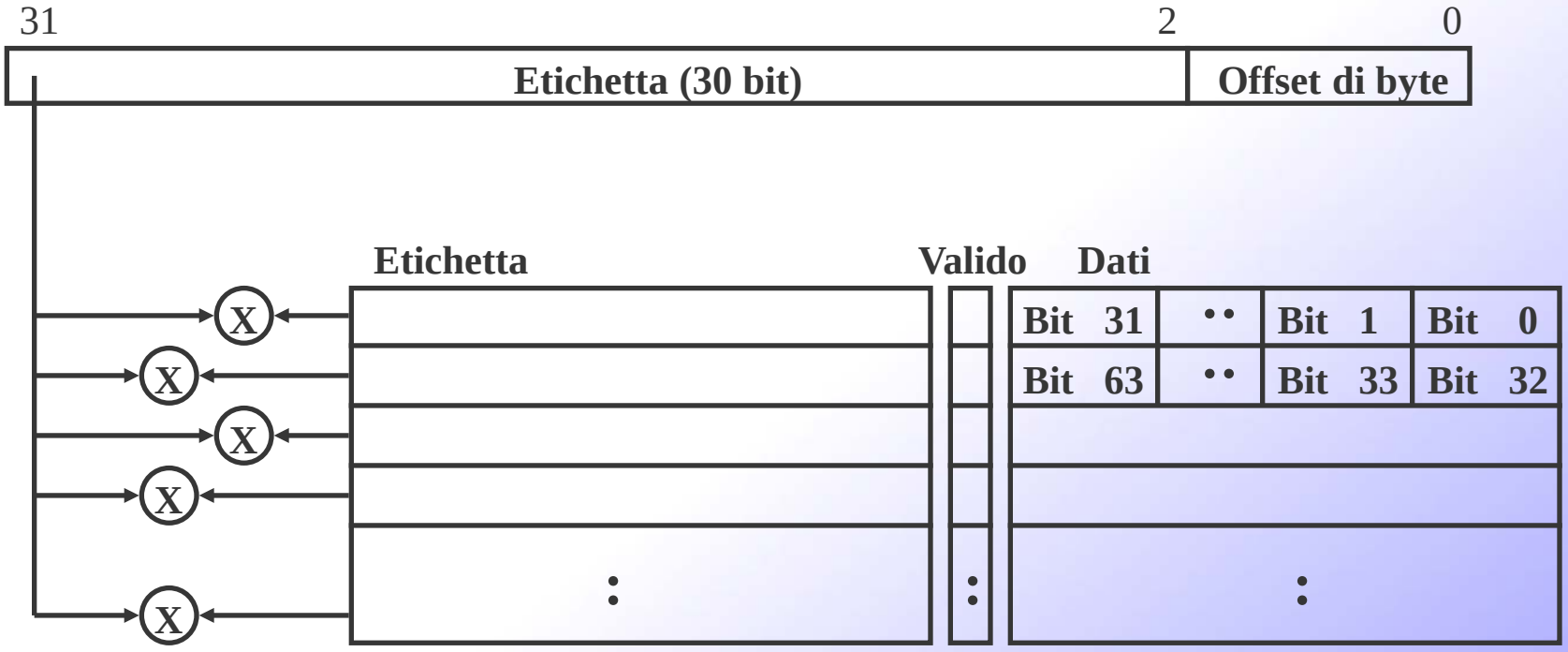
- Due modi di migliorare le prestazioni:
  - diminuire il tasso di fallimento
  - diminuire la penalità di fallimento

Cache completamente associativa

Nessun indice di cache

Confronta i campi etichetta di tutti gli elementi della cache in parallelo

Esempio:



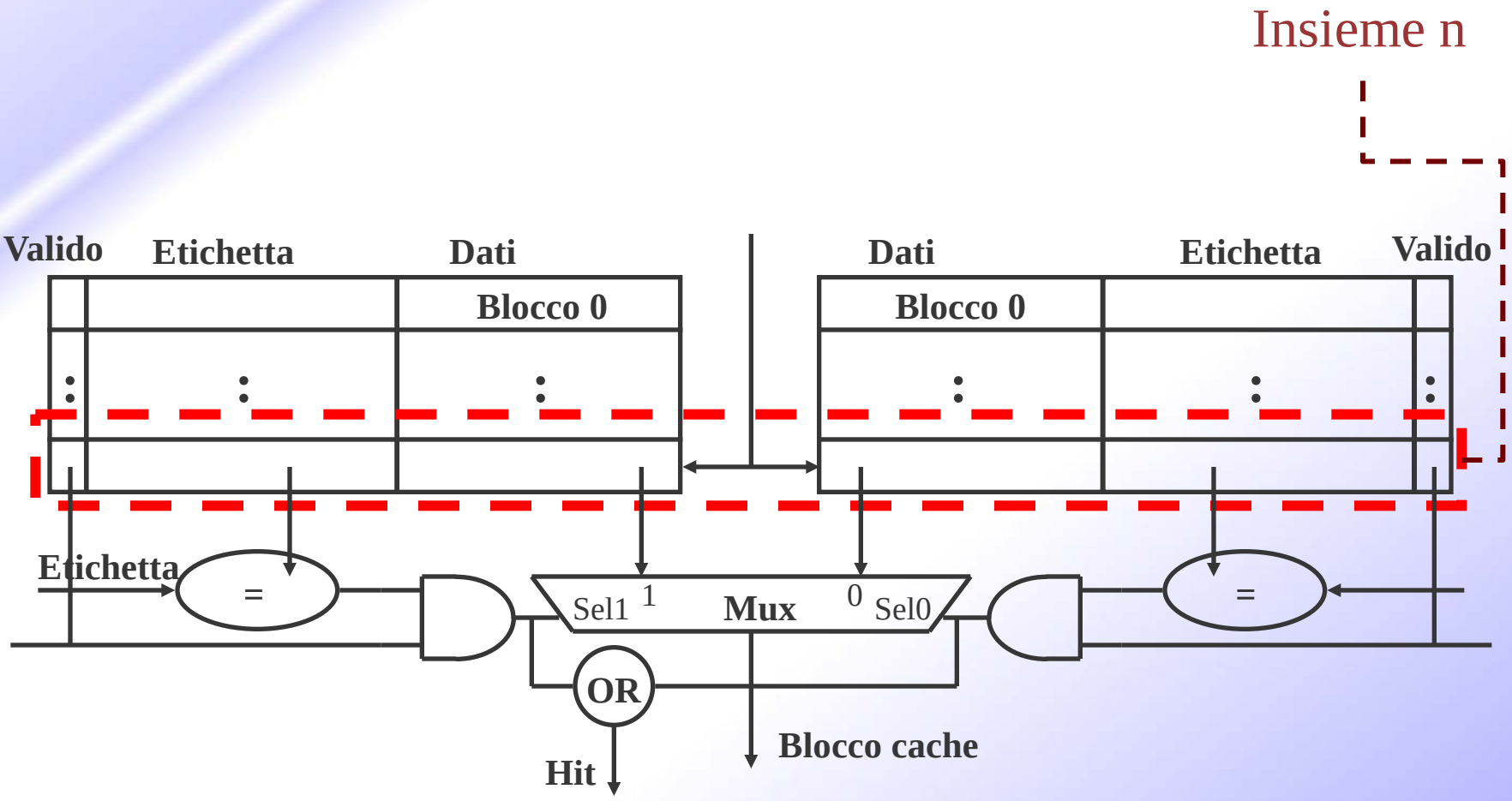
Cache associativa ad N vie

- Insieme di N elementi per indice di blocco

  - Indice di blocco seleziona un insieme di N elementi della cache
  - Le N etichette dell'insieme sono confrontate in parallelo
  - Il dato è selezionato in base al risultato del confronto

$$Ind.\ blocco_{cache} = (ind._{mem}) \bmod (n^{\circ}\ insiemi)$$

Esempio: 2 vie



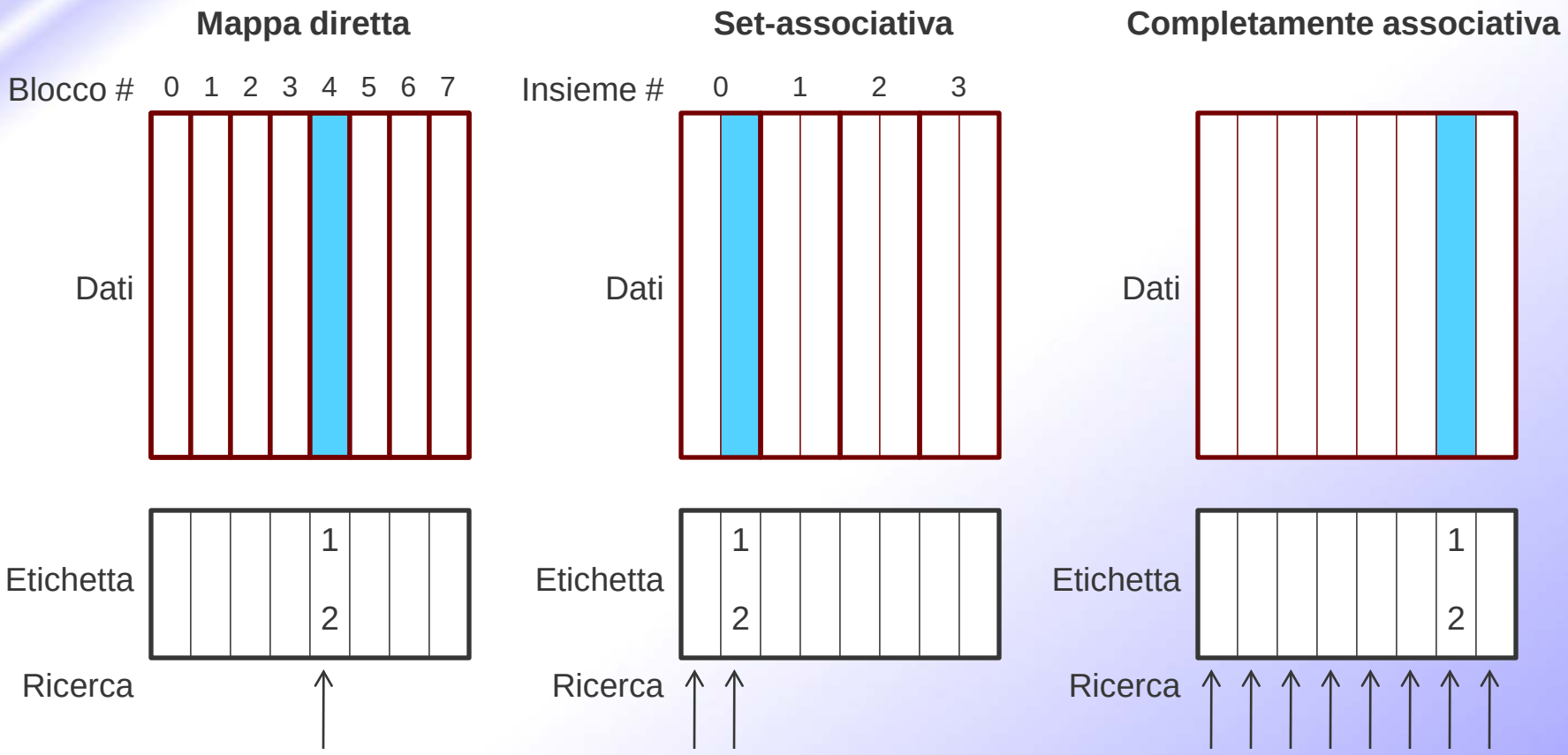
Svantaggi della cache associativa

- Cache associativa ad N vie in confronto a cache a mappa diretta:
  - N comparatori rispetto ad 1
  - Ritardo extra dovuto al multiplexer per i dati
  - Dati arrivano dopo aver deciso il successo ed aver selezionato l'insieme

- In una cache a mappa diretta, il blocco cache è disponibile prima di aver deciso il successo:
  - Possibile assumere che sia un successo e continuare (recuperare dopo se è un fallimento).

Cache a confronto

- Cache contenente 8 blocchi



Esempio: mappa diretta

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 4) = 0
6	(6 mod 4) = 2
8	(8 mod 4) = 0

- Accesso al blocco 0, FALLIMENTO

Blocco in memoria	S/F	Blocco 0	Blocco 1	Blocco 2	Blocco 3
0	F	Mem[0]			

Esempio: mappa diretta

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 4) = 0
6	(6 mod 4) = 2
8	(8 mod 4) = 0

- Accesso al blocco 8, FALLIMENTO

Blocco in memoria	S/F	Blocco 0	Blocco 1	Blocco 2	Blocco 3
0	F	Mem[0]			
8	F	Mem[8]			

Esempio: mappa diretta

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 4) = 0
6	(6 mod 4) = 2
8	(8 mod 4) = 0

- Accesso al blocco 0, FALLIMENTO

Blocco in memoria	S/F	Blocco 0	Blocco 1	Blocco 2	Blocco 3
0	F	Mem[0]			
8	F	Mem[8]			
0	F	Mem[0]			

Esempio: mappa diretta

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 4) = 0
6	(6 mod 4) = 2
8	(8 mod 4) = 0

- Accesso al blocco 6, FALLIMENTO

Blocco in memoria	S/F	Blocco 0	Blocco 1	Blocco 2	Blocco 3
0	F	Mem[0]			
8	F	Mem[8]			
0	F	Mem[0]			
6	F	Mem[0]		Mem[6]	

Esempio: mappa diretta

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 4) = 0
6	(6 mod 4) = 2
8	(8 mod 4) = 0

- Accesso al blocco 8, FALLIMENTO

Blocco in memoria	S/F	Blocco 0	Blocco 1	Blocco 2	Blocco 3
0	F	Mem[0]			
8	F	Mem[8]			
0	F	Mem[0]			
6	F	Mem[0]		Mem[6]	
8	F	Mem[8]		Mem[6]	

Associativa a 2 vie

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 2) = 0
6	(6 mod 2) = 0
8	(8 mod 2) = 0

- Accesso al blocco 0, FALLIMENTO

Blocco in memoria	S/F	Insieme 0		Insieme 1	
0	F	Mem[0]			

Associativa a 2 vie

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 2) = 0
6	(6 mod 2) = 0
8	(8 mod 2) = 0

- Accesso al blocco 8, FALLIMENTO

Blocco in memoria	S/F	Insieme 0		Insieme 1	
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		

Associativa a 2 vie

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 2) = 0
6	(6 mod 2) = 0
8	(8 mod 2) = 0

- Accesso al blocco 0, SUCCESSO

Blocco in memoria	S/F	Insieme 0		Insieme 1	
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		

Associativa a 2 vie

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 2) = 0
6	(6 mod 2) = 0
8	(8 mod 2) = 0

- Accesso al blocco 6, FALLIMENTO

Blocco in memoria	S/F	Insieme 0		Insieme 1	
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[6]		

Associativa a 2 vie

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

Indirizzo del blocco in memoria	Blocco cache
0	(0 mod 2) = 0
6	(6 mod 2) = 0
8	(8 mod 2) = 0

- Accesso al blocco 8, FALLIMENTO

Blocco in memoria	S/F	Insieme 0		Insieme 1	
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[6]		
8	F	Mem[8]	Mem[6]		

Completamente associativa

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

  - Accesso al blocco 0, FALLIMENTO

Blocco da accedere	S/F	Blocco			
0	F	Mem[0]			

Completamente associativa

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8
  - Accesso al blocco 8, FALLIMENTO

Blocco da accedere	S/F	Blocco			
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		

Completamente associativa

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

  - Accesso al blocco 0, SUCCESSO

Blocco da accedere	S/F	Blocco			
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		

Completamente associativa

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

  - Accesso al blocco 6, FALLIMENTO

Blocco da accedere	S/F	Blocco			
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[8]	Mem[6]	

Completamente associativa

- Cache da 4 parole
- Sequenza di accessi: 0,8,0,6,8

  - Accesso al blocco 8, SUCCESSO

Blocco da accedere	S/F	Blocco			
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[8]	Mem[6]	
8	S	Mem[0]	Mem[8]	Mem[6]	

Etichette ed associatività

- Supponiamo:
  - cache con 4K blocchi di dati
  - indirizzi a 32 bit
- Cache a mappa diretta:
  - indice di cache: log(4K)=12 bit
  - etichetta: (32-12) = 20
- Ogni raddoppio dell'associatività (lasciando inalterato il numero complessivo di blocchi) diminuisce i bit dell'indice di 1 ed aumenta i bit dell'etichetta di 1
- Cache a 2 vie:
  - etichetta: 21 (ne servono due per insieme)
- Cache a 4 vie:
  - etichetta: 22 (ne servono quattro per insieme)
- Cache completamente associativa:
  - etichetta: 30 (ne servono 4K per insieme)

Scegliere il blocco da sostituire

- Facile per cache a mappa diretta
- Associativa ad N vie o associativa completa:
  - casuale
  - LRU (Least Recently Used)

Associatività	2 vie		4 vie		8 vie	
Dimensione Cache	LRU	Random	LRU	Random	LRU	Random
16 KB	5,20%	5,70%	4,70%	5,30%	4,40%	5,00%
64 KB	1,90%	2,00%	1,50%	1,70%	1,40%	1,50%
256 KB	1,15%	1,17%	1,13%	1,13%	1,12%	1,12%

Cache a più livelli - 1

- Aggiungere un secondo livello di cache:
  - spesso cache primaria sullo stesso chip del processore
  - usa SRAM per aggiungere un'altra cache sopra la memoria principale (DRAM)
  - penalità di fallimento diminuisce se i dati sono nella cache di secondo livello

Cache a più livelli

- Esempio:
  - CPI = 1, clock = 500Mhz, tasso di fallimento = 5%, penalità di fallimento = 200ns
    - CicliStallo=tasso di fallimento x (clock x penalità di fallimento)
    - CPIReale = CPI + cicliStallo = 1+0.05 x ($500 \cdot 10^6$ x $200 \cdot 10^{-9}$) = 1+0.05 x 100 = 6
  - **Aggiungere cache di secondo livello con tempo di accesso = 20ns e tasso di fallimento = 2%**
    - Penalità di fallimento 1° Liv = 20 ns, Penalità di fallimento 2° Liv = 200 ns
    - CPIReale = CPI+cicliStallo1°Liv+cicliStallo2°Liv = 1+0.05 x 10+0.02x100 = 3.5
- Usando cache a più livelli:
  - **ottimizzare il tempo di decisione nella cache a primo livello**
  - **ottimizzare il tasso di fallimento nella cache a secondo livello**

Impatto sugli algoritmi

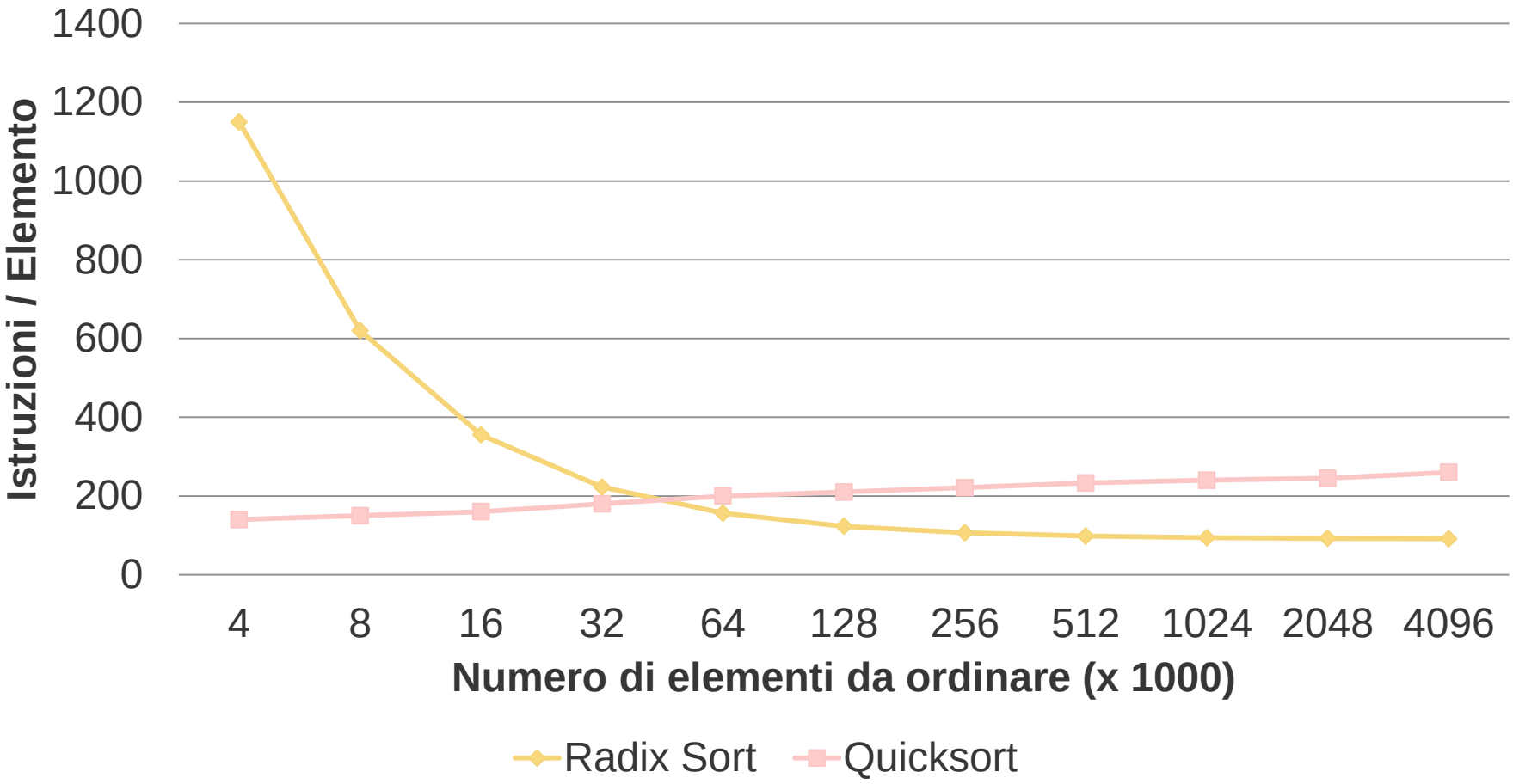
- Quicksort: algoritmo più veloce (basato su confronti) se i dati entrano in memoria principale

- Radix sort: detto anche "tempo lineare" in quanto, con chiavi di lunghezza fissata e radice fissata, per ordinare i dati è sufficiente un numero costante di passi indipendentemente dal numero di chiavi

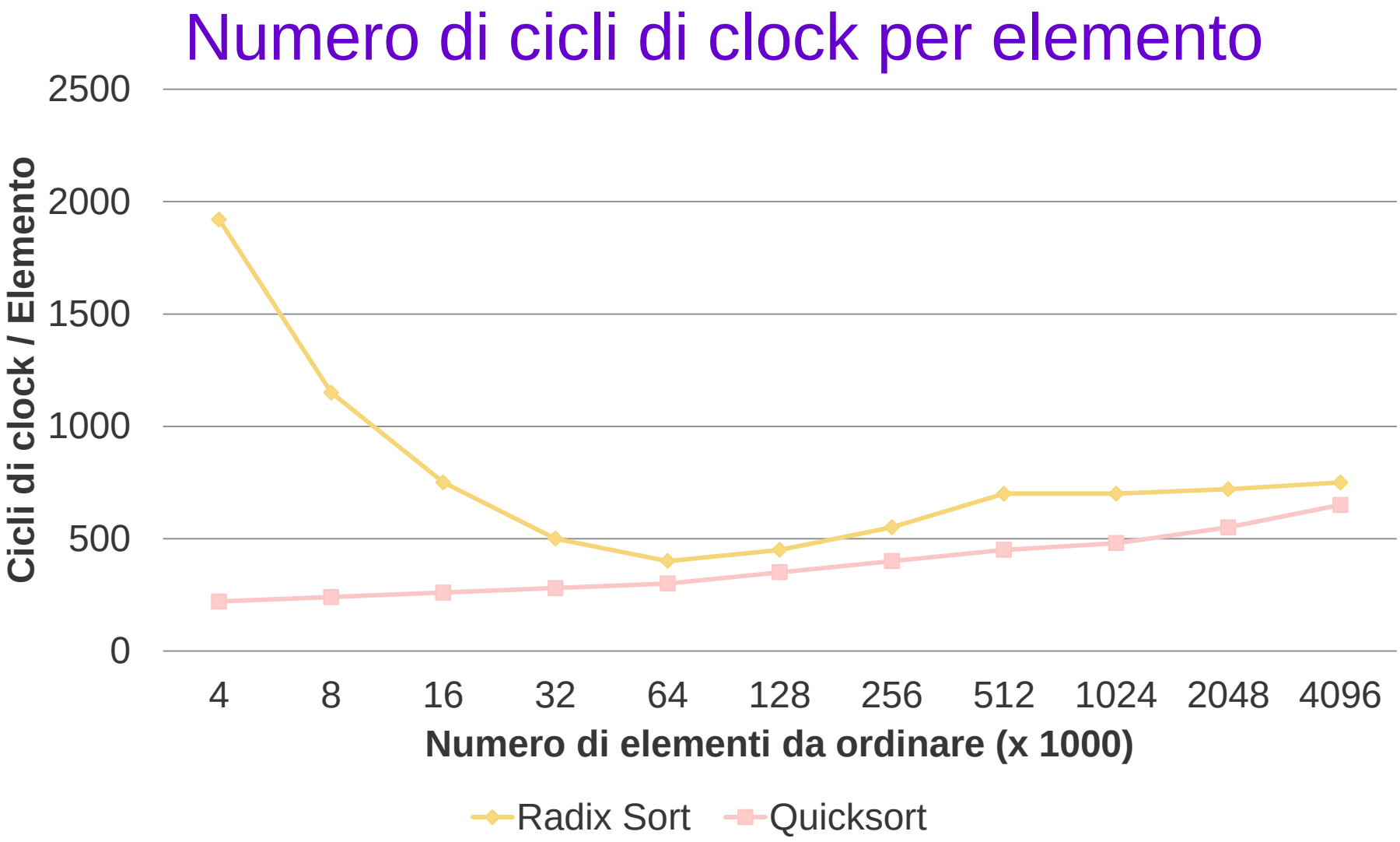
- Su Alphastation 250, blocchi da 32 byte, cache mappa diretta L2 2MB, chiavi da 8 byte, da 4000 a 4000000

Confronto fra Radix Sort e Quicksort

Numero di istruzioni per elemento



Confronto fra Radix Sort e Quicksort



Confronto fra Radix Sort e Quicksort

Numero di fallimenti in cache per elemento