

Cache, da “direct” a “mapped”

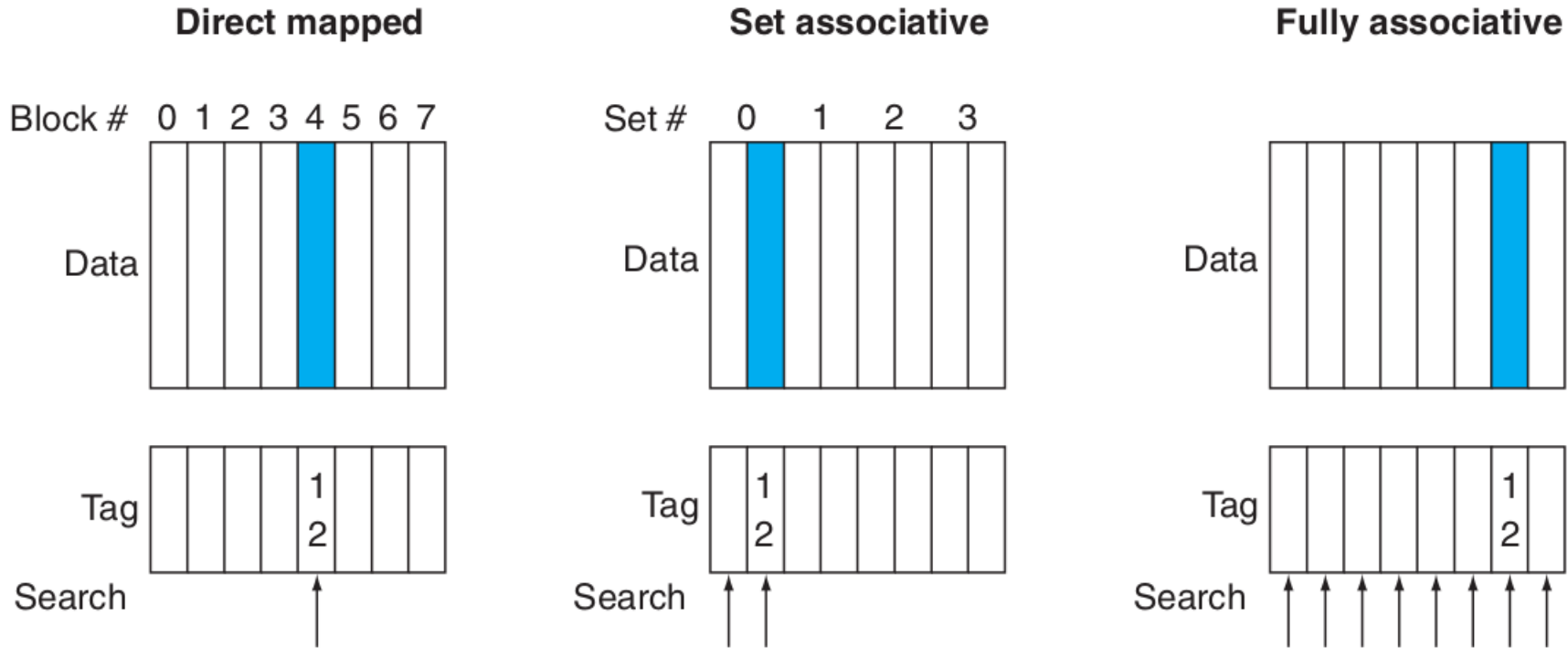


FIGURE 5.14 The location of a memory block whose address is 12 in a cache with eight blocks varies for direct-mapped, set-associative, and fully associative placement. In direct-mapped placement, there is only one cache block where memory block 12 can be found, and that block is given by $(12 \bmod 8) = 4$. In a two-way set-associative cache, there would be four sets, and memory block 12 must be in set $(12 \bmod 4) = 0$; the memory block could be in either element of the set. In a fully associative placement, the memory block for block address 12 can appear in any of the eight cache blocks.

- “four way” set

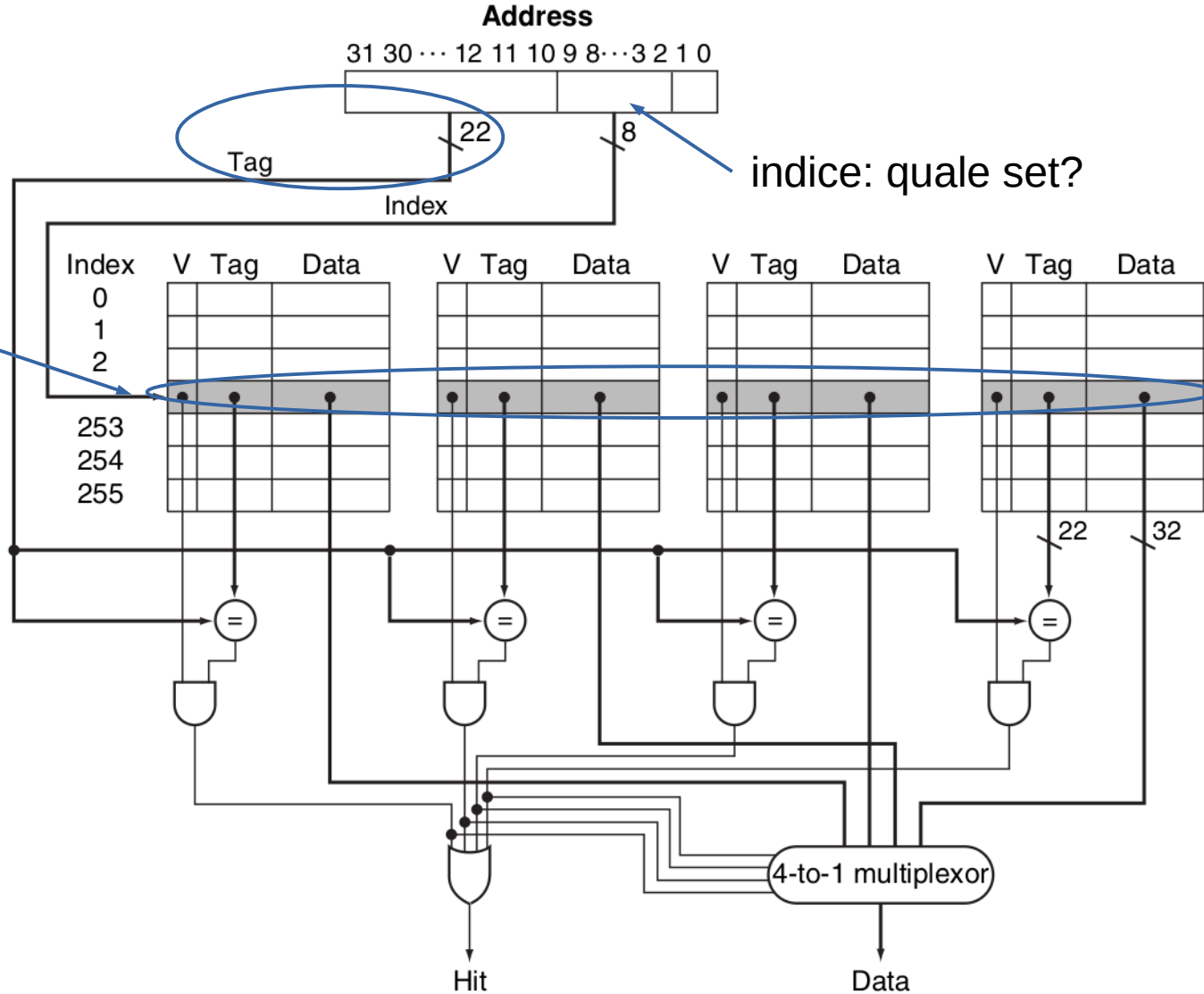


FIGURE 5.18 The implementation of a four-way set-associative cache requires four

Cache:

- stessa dimensione
- diverso utilizzo

ogni blocco è un set (a 1 posto)

One-way set associative
→ (direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

l'intera cache è un singolo set

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

FIGURE 5.15 An eight-block cache configured as direct mapped, two-way set associative, four-way set associative, and fully associative. The total size of the cache in blocks is equal to the

Usare le “parti” di un indirizzo

- Index per scegliere il “set” ($\text{num.set} = 2^{\text{num.bit in index}}$)
(NB: in mappatura diretta ogni blocco è un set a sé stante)
- Tag per riconoscere il blocco tra quelli “del set”
- Offset, per indicare un singolo byte del blocco
(es.: se blocco=word allora 2 bit) ($\text{dim.blocco} = 2^{\text{num.bit in offset}}$)



FIGURE 5.17 The three portions of an address in a set-associative or direct-mapped cache. The **index** is used to select the set, then the tag is used to choose the block by comparison with the blocks in the selected set. The block offset is the address of the desired data within the block.

LRU, ed esempio sul testo

- Dati importanti:
 - Dim.totale
 - Dim.blocchi
 - Organizzazione cache
 - Sequenza accessi (indirizzi di byte o di blocchi RAM?)
- 1 consisting of four one-word blocks.
is two-way set-associative, and the
of misses for each cache organization
addresses: 0, 8, 0, 6, and 8.
-

- Ricordare: **least recently used (LRU)** A replacement scheme in which the block replaced is the one that has been unused for the longest time.

(NB: non è rilevante per la mappatura diretta, non ho “scelta” nel “set”)

Eserc.:

The set-associative cache has two sets (with indices 0 and 1) with two elements per set. Let's first determine to which set each block address maps:

- Indirizzi "di blocco"!
(=word, non di byte)
- Ricavo indice da LSB

Block address	Cache set
0	(0 modulo 2) = 0
6	(6 modulo 2) = 0
8	(8 modulo 2) = 0

- Applico LRU: (miss per "prima load") (miss per "LRU")

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		Set 0	Set 0	Set 1	Set 1
0	miss	Memory[0]			
8	miss	Memory[0]	Memory[8]		
0	hit	Memory[0]	Memory[8]		
6	miss	Memory[0]	Memory[6]		
8	miss	Memory[8]	Memory[6]		