

¿Dónde estamos parados?

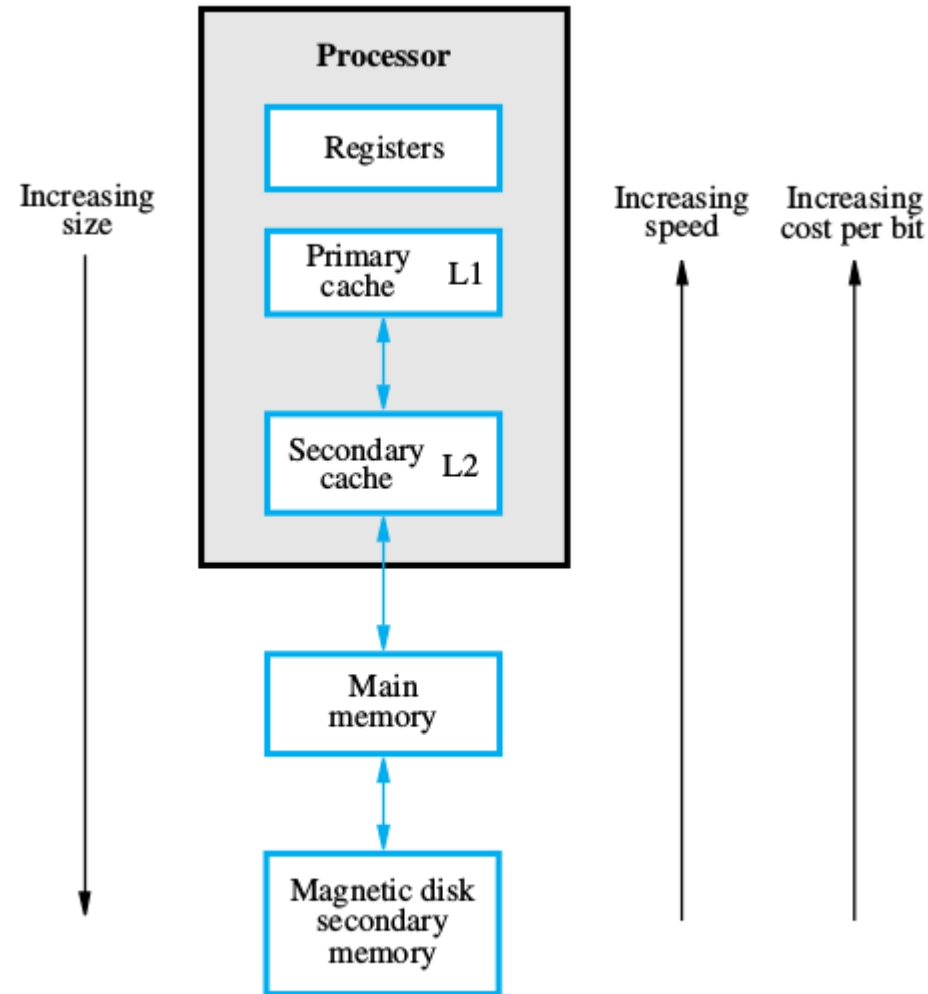


Figure 8.14 Memory hierarchy.

Jerarquías de memoria

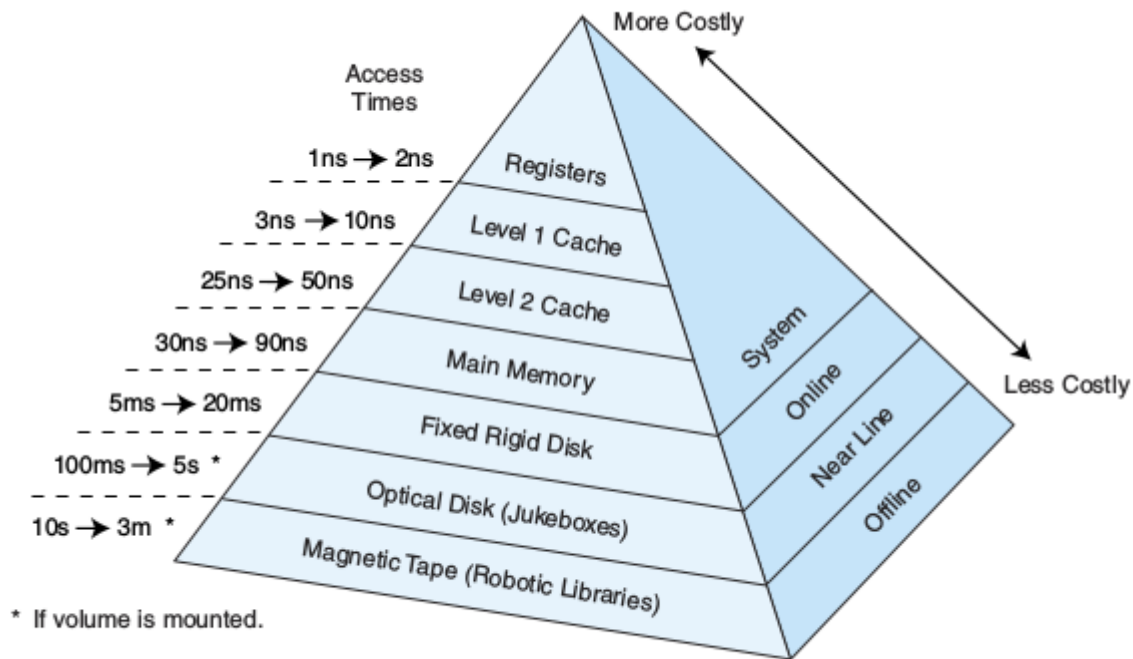


FIGURE 6.1 The Memory Hierarchy

Hit: el dato requerido reside en el nivel de memoria.

Miss: el dato no reside.

Hit rate: El porcentaje de accesos a memoria encontrados (hit) en un determinado nivel.

Miss rate: El porcentaje de accesos a memoria no encontrados (miss) en un determinado nivel.

Jerarquías de memoria

Principio de localidad: si la dirección de memoria X es accedida en tiempo t , existe una alta probabilidad que la dirección $X+1$ sea accedida en el tiempo futuro

- * **Localidad temporal:** Los datos recientemente accedidos tienden a ser accedidos en el futuro cercano.
- * **Localidad espacial:** Los accesos tienden a ser agrupados en el espacio de direcciones (por ejemplo, arreglos en ciclos).
- * **Localidad secuencial:** Las instrucciones tienden a ser accedidas secuencialmente.

Memoria Cache

CAM: *content addressable memory*. Accedida por contenido (no por direcciones)

hay que buscar dónde puede estar el dato que necesito!

Cache schema mappings
(organizaciones de cache)

Cache schema mappings (organizaciones de cache)

Supongamos el siguiente conjunto de instrucciones, ¿cómo funciona esto de la cache?:

EIP —▶ 01100
 01101
 01110
 11100

Además supongamos:

- Tenemos una cantidad total de memoria principal de 32 palabras (re poquito).
- Tenemos una memoria cache que puede almacenar 16 palabras.
(bastante! Comparado con la principal)

Primero imaginemos estas memorias, comprender qué representan los bits de las direcciones, cómo se busca en la cache
y cómo se conformaría una **dirección de memoria principal** y una de **memoria cache**

Cache schema mappings (organizaciones de cache)

Memoria Cache

0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

$2^4 = 16$ palabras diferentes

Memoria Principal

	00000
	00001
	00010
	00011
	00100
	00101
	00110
	00111
	01000
	01001
	01010
	01011
	01100
	01101
	01110
	01111
	10000
	10001
	10010
	10011
	10100
	10101
	10110
	10111
	11000
	11001
	11010
	11011
	11100
	11101
	11110
	11111

$2^5 = 32$ palabras diferentes

Agrupar!: la memoria principal y la cache son particionadas en bloques del mismo tamaño

Organización Directa

Memoria Cache

0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

16 palabras diferentes

Bloque: grupo de palabras (igual tamaño en principal y cache)

Offset: identifica una palabra dentro de un bloque

Tag: resto de la dirección

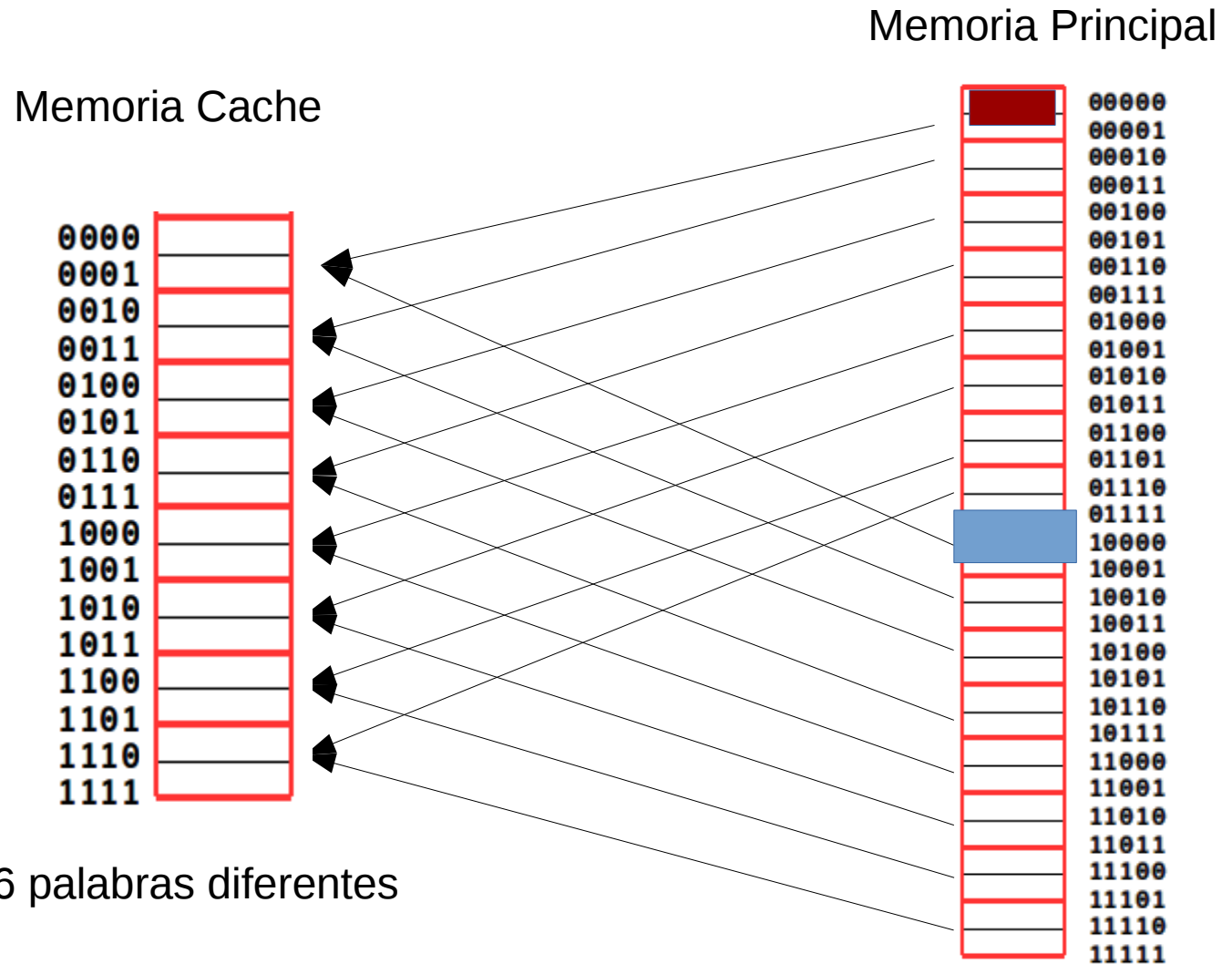
Memoria Principal

	00000
	00001
	00010
	00011
	00100
	00101
	00110
	00111
	01000
	01001
	01010
	01011
	01100
	01101
	01110
	01111
	10000
	10001
	10010
	10011
	10100
	10101
	10110
	10111
	11000
	11001
	11010
	11011
	11100
	11101
	11110
	11111

32 palabras diferentes

Nos interesan los bloques de Cache

Organización Directa



Bloque: grupo de palabras (igual tamaño en principal y cache)

Offset: identifica una palabra dentro de un bloque

Tag: resto de la dirección

32 palabras diferentes

Organización Directa

Memoria Cache

Pregunta: ¿qué tamaño de bloque ven?
(cuántas palabras entran en un bloque)

Respuesta: 2

Pregunta: ¿cuántos bits hacen falta para identificar una palabra en un bloque?
(bits de offset)

Respuesta: 1

0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

16 palabras diferentes

Bloque: grupo de palabras (igual tamaño en principal y cache)
Offset: identifica una palabra dentro de un bloque
Tag: resto de la dirección

Memoria Principal

	00000
	00001
	00010
	00011
	00100
	00101
	00110
	00111
	01000
	01001
	01010
	01011
	01100
	01101
	01110
	01111
	10000
	10001
	10010
	10011
	10100
	10101
	10110
	10111
	11000
	11001
	11010
	11011
	11100
	11101
	11110
	11111

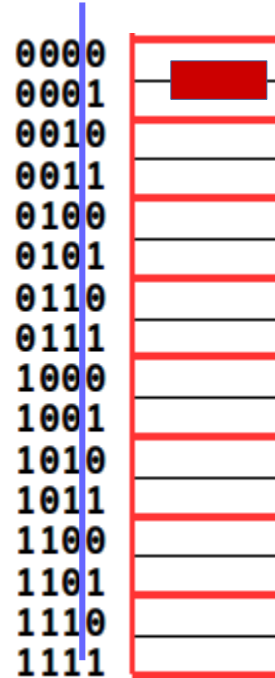
32 palabras diferentes

Nos interesan los bloques de Cache

Organización Directa

Memoria Cache

16 palabras = 8 bloques
2 pal x bloque



16 palabras diferentes

EIP
01100
01101
01110
11100

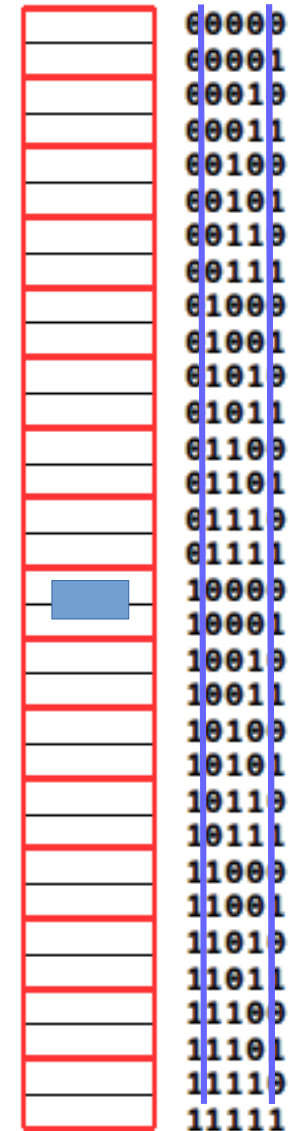
Dirección de
Mem. Cache:

BLOQUE	OFFSET
3	1

4

Dirección de
Mem. Principal:

Memoria Principal



32 palabras diferentes

TAG	BLOQUE	OFFSET
1	3	1

Organización Directa

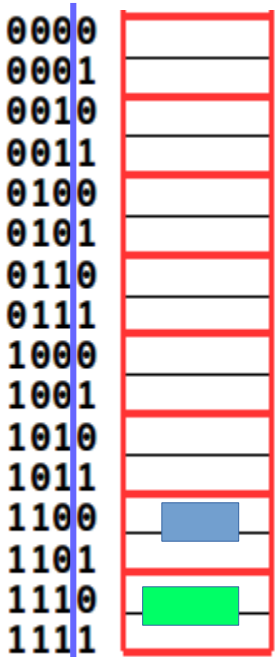
Memoria Principal

Memoria Cache

Tabla auxiliar

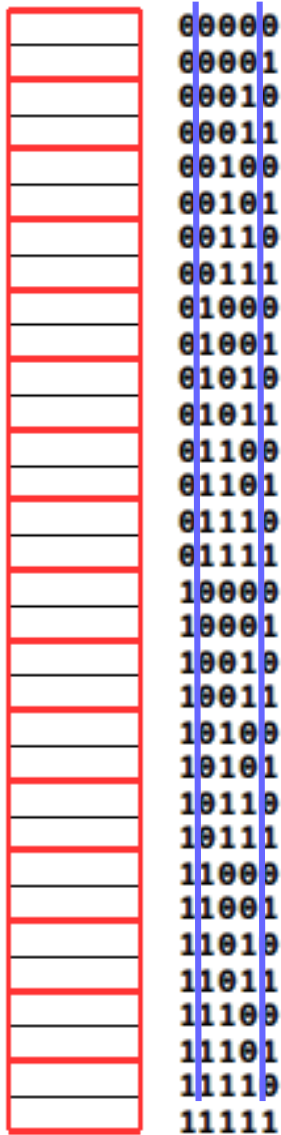
Bloque 3	Tag 1
000	
001	
010	
011	
100	
101	
110	0
111	0

EIP
01100 M
01101 H
01110 M
11100 M



16 palabras diferentes

BLOQUE 3	OFFSET 1
-------------	-------------



32 palabras diferentes

TAG 1	BLOQUE 3	OFFSET 1
----------	-------------	-------------

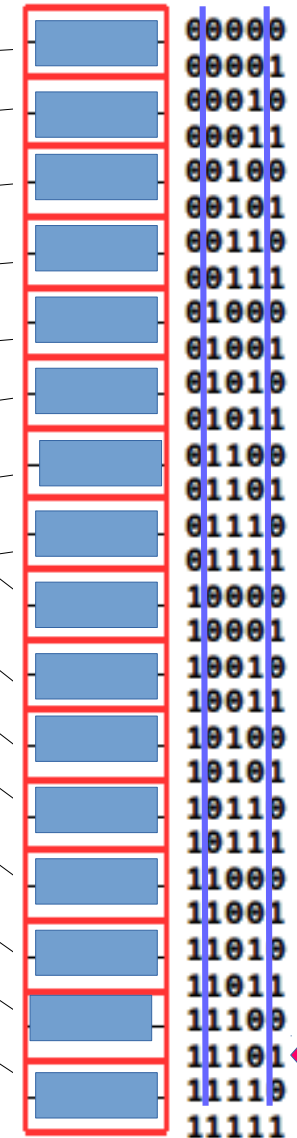
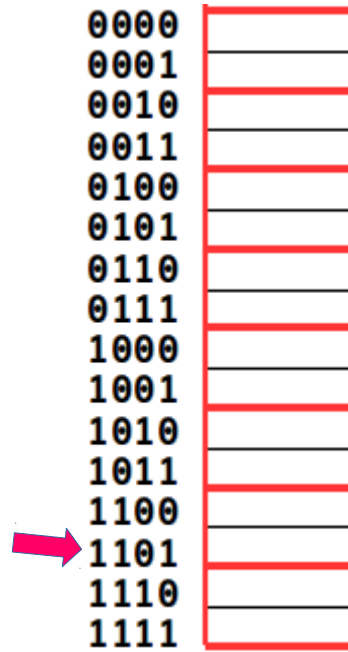
Organización Directa

Memoria Cache

Memoria Principal

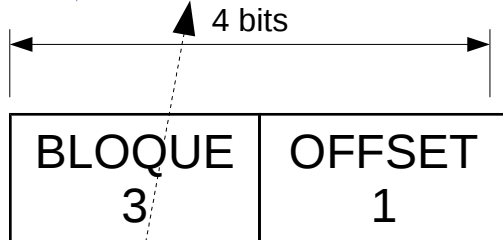
Tabla auxiliar

bloque	tag
000	
001	
010	
011	
100	
101	
110	0
111	



(2)

Dir
Cache

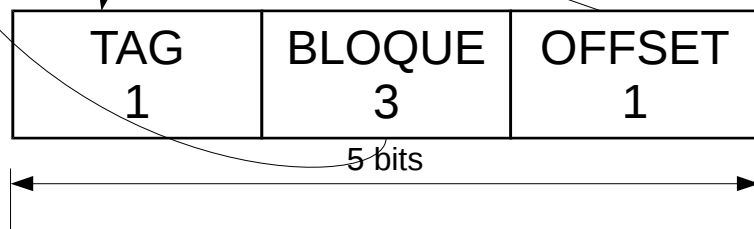


(1)

EIP

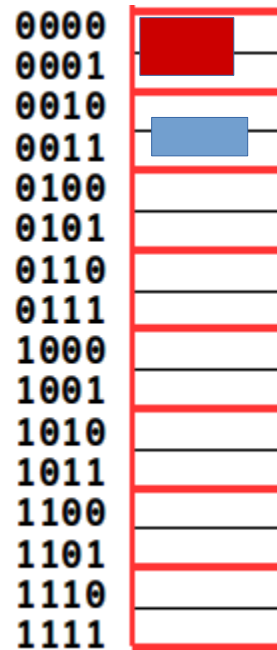
Dir Mem P.	Dir Cache
01100 M	1100
01101 H	1101
11101 M	1101
01101 M	1101
11101 M	1101
01100 M	1100

Dir Mem.
Principal



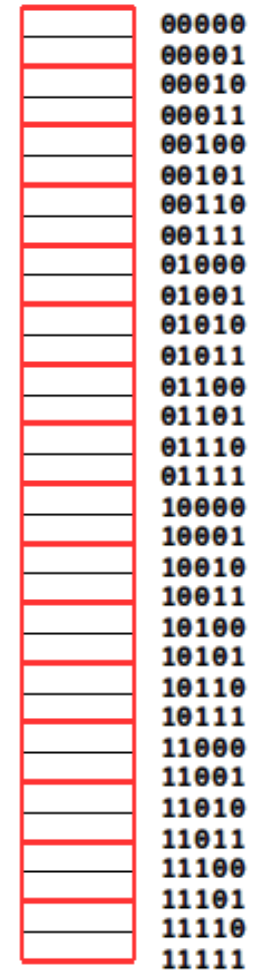
Organización Full asociativa

Memoria Cache



16 palabras diferentes

Memoria Principal



32 palabras diferentes

Bloque: grupo de palabras (igual tamaño en principal y cache)

Offset: identifica una palabra dentro de un bloque

Tag: resto de la dirección

Organización Full Asociativa

Memoria de búsqueda asociativa

Bloque 3	Tag 4
000	
001	
010	
011	
100	
101	
110	
111	

Memoria Cache

0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

16 palabras diferentes

Dirección de Mem. Cache:

BLOQUE 3	OFFSET 1
-------------	-------------

Memoria Principal

	0000
	0001
	0010
	0011
	0100
	0101
	0110
	0111
	1000
	1001
	1010
	1011
	1100
	1101
	1110
	1111
	0000
	0001
	0010
	0011
	0100
	0101
	0110
	0111
	1000
	1001
	1010
	1011
	1100
	1101
	1110
	1111

32 palabras diferentes

Dirección de Mem. Principal:

TAG 4	OFFSET 1
----------	-------------

Organización Full Asociativa

Memoria de búsqueda asociativa

Bloque	Tag
3	4
000	0110
001	1110
010	
011	
100	
101	
110	
111	

4 bits

(3)

Dir Cache

BLOQUE	OFFSET
--------	--------

(1)

Dir Mem. Principal

TAG	OFFSET
4	1

5 bits

Memoria Cache

0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Memoria Principal

	00000
	00001
	00010
	00011
	00100
	00101
	00110
	00111
	01000
	01001
	01010
	01011
	01100
	01101
	01110
	01111
	10000
	10001
	10010
	10011
	10100
	10101
	10110
	10111
	11000
	11001
	11010
	11011
	11100
	11101
	11110
	11111

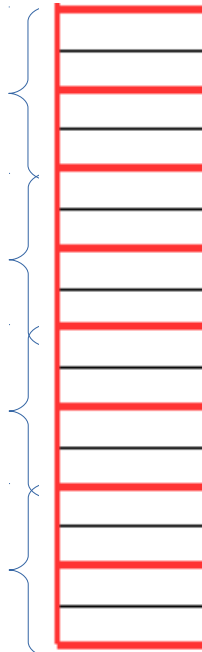
Dir Mem P.	Dir Cache
01100 M	0000
01101 H	0001
11101 M	0011
01101 H	0001
11101 H	0011
01100 H	0000

Organización N-set Asociativa (N= cantidad de bloques por set)

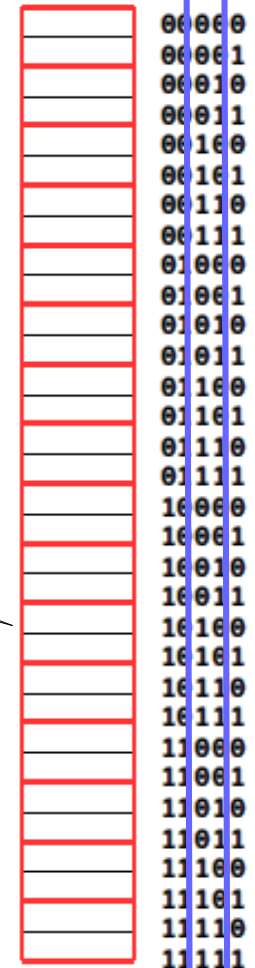
Memoria Principal

Memoria Cache

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111



16 palabras diferentes



32 palabras diferentes

- Agrupo por SET
- Directa hacia cada SET
- En cada set es full asociativa

Bloque: Identifica un bloque en un SET
Offset: identifica una palabra dentro de un BLOQUE
Tag: resto de la dirección de Mem. Principal
Set: grupo de bloques

Organización N-set Asociativa (N= cantidad de bloques por set)

Sea 4-set asociativa

Memoria de búsqueda
Asociativa SET 0

bloque	tag
00	011
01	111
10	
11	

Memoria de búsqueda
Asociativa SET 1

bloque	tag
00	
01	
10	
11	

O bien

(3)

4 bits

Dir
Cache

SET	BLOQUE	OFFSET
1	2	1

(2)

(1)

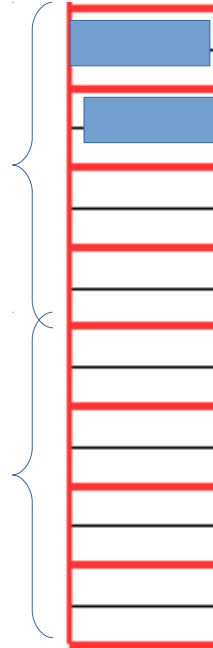
Dir Mem.
Principal

TAG	SET	OFFSET
3	1	1

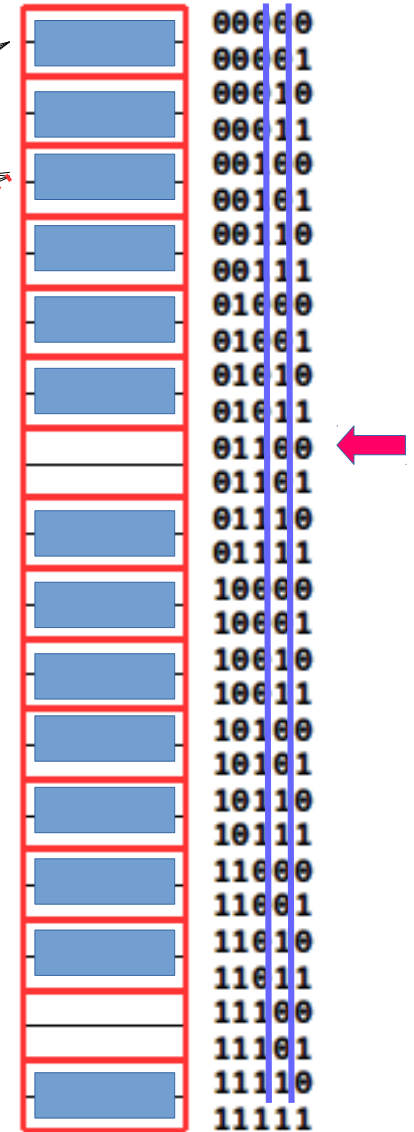
5 bits

Memoria Cache

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111



Memoria Principal

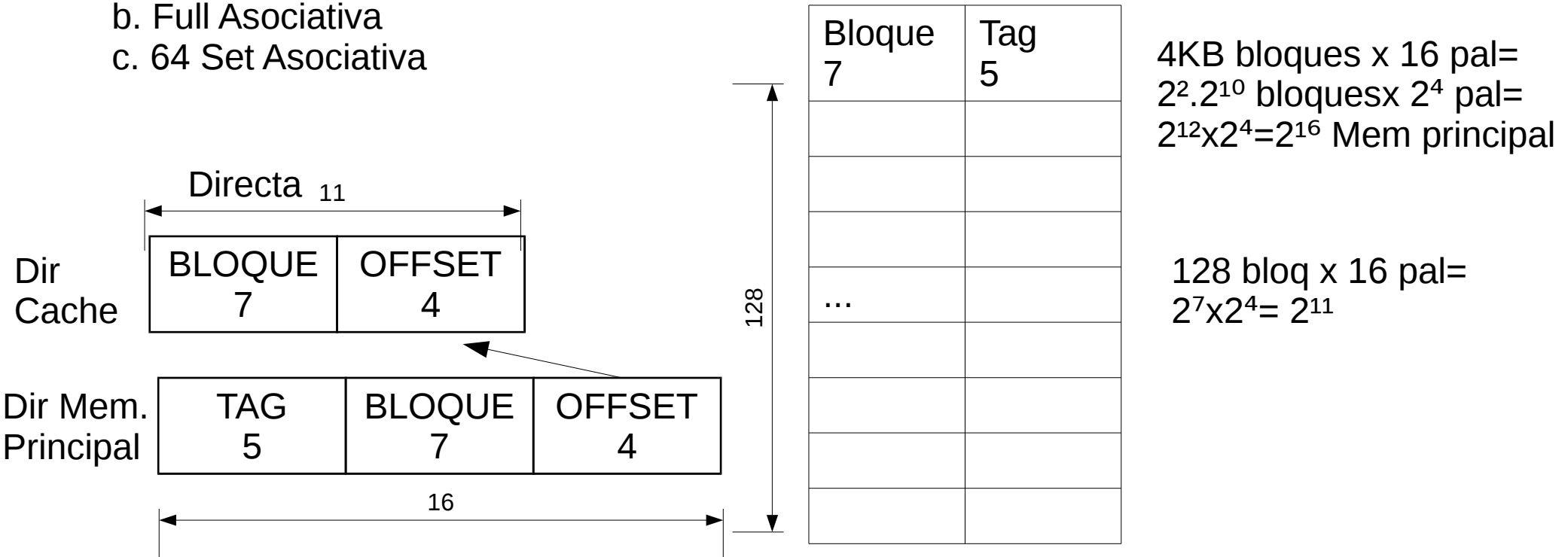


Dir Mem P.			Dir Cache		
tag	set	off	set	bl	off
011	0	0 M	0	00	0
011	0	1 H	0	00	1
111	0	1 M	0	01	1
011	0	1 H	0	00	1
111	0	1 H	0	01	1
011	0	0 H	0	00	0

Ejercicios

Dado un sistema de memoria principal de 4KB bloques de 16 palabras c/u y una memoria caché de 128 bloques, indique cuál sería la decodificación a realizar, si la memoria caché tuviera una organización:

- Directa
- Full Asociativa
- 64 Set Asociativa



Ejercicios

Dado un sistema de memoria principal de 4KB bloques de 16 palabras c/u y una memoria caché de 128 bloques, indique cuál sería la decodificación a realizar, si la memoria caché tuviera una organización:

- Directa
- Full Asociativa
- 64 Set Asociativa

Full Asociativa

BLOQUE	OFFSET
--------	--------

Dir Cache

	TAG	OFFSET
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		

Dir Mem.
Principal

Memoria de búsqueda asociativa

[illegible]

Ejercicios

Dado un sistema de memoria principal de 4KB bloques de 16 palabras c/u y una memoria caché de 128 bloques, indique cuál sería la decodificación a realizar, si la memoria caché tuviera una organización:

- Directa
- Full Asociativa
- 64 Set Asociativa

64 Set Associativa

Dir
Cache

SET	BLOQUE	OFFSET
-----	--------	--------

Dir Mem.
Principal

TAG	SET	OFFSET
-----	-----	--------

Memoria de búsqueda asociativa

[illegible]