Tabla 4.2. Elementos de diseño de la caché.

Tamaño de caché	**Política de escritura**
Función de correspondencia	Escritura inmediata
Directa	Postescritura
Asociativa	Escritura única
Asociativa por conjuntos	**Tamaño de línea**
Algoritmo de sustitución	**Número de cachés**
Utilizado menos recientemente (LRU)	Uno o dos niveles
Primero en entrar-primero en salir (FIFO)	Unificada o partida
Utilizado menos frecuentemente (LFU)	
Aleatorio	

TAMAÑO DE CACHÉ

El primer elemento, el tamaño de caché, ya ha sido tratado. Nos gustaría que el tamaño fuera lo suficientemente pequeño como para que el coste total medio por bit se aproxime al de la memoria principal sola, y que fuera lo suficientemente grande como para que el tiempo de acceso medio total sea próximo al de la caché sola. Hay otras muchas motivaciones para minimizar el tamaño de la caché. Cuanto más grande es, mayor es el número de puertas implicadas en direccionar la caché. El resultado es que cachés grandes tienden a ser ligeramente más lentas que las pequeñas (incluso estando fabricadas con la misma tecnología de circuito integrado y con la misma ubicación en el chip o en la tarjeta de circuito impreso). El tamaño de caché está también limitado por las superficies disponibles de chip y de tarjeta. Como las prestaciones de la caché son muy sensibles al tipo de tarea, es imposible predecir un tamaño «óptimo». La Tabla 4.3 lista los tamaños de caché de diversos procesadores antiguos y modernos.

FUNCIÓN DE CORRESPONDENCIA

Ya que hay menos líneas de caché que bloques de memoria principal, se necesita un algoritmo que haga corresponder bloques de memoria principal a líneas de caché. Además, se requiere algún medio para determinar qué bloque de memoria principal ocupa actualmente una línea dada de caché. La elección de la función de correspondencia determina cómo se organiza la caché. Pueden utilizarse tres técnicas: directa, asociativa, y asociativa por conjuntos. Examinamos a continuación cada una de ellas. En cada caso veremos la estructura general y un ejemplo concreto.

Ejemplo 4.2. Para los tres casos, el ejemplo incluye los siguientes elementos:

- La caché puede almacenar 64 KB.
- Los datos se transfieren entre la memoria principal y la caché en bloques de 4 bytes. Esto significa que la caché está organizada en $16K = 2^{14}$ líneas de 4 bytes cada una.
- La memoria principal es de 16MB, con cada byte directamente direccionable mediante una dirección de 24 bits ($2^{24} = 16M$). Así pues, al objeto de realizar la correspondencia, podemos considerar que la memoria principal consta de 4M bloques de 4 bytes cada uno.

Tabla 4.3. Tamaños de caché de algunos procesadores.

Procesador	Tipo	Año de introducción	Caché L1[a]	Caché L2	Caché L3
IBM 360/55	Gran computador	1968	16 a 32 KB	—	—
PDP-11/70	Minicomputador	1975	1 KB	—	—
VAX 11/780	Minicomputador	1978	16 KB	—	—
IBM 3033	Gran computador	1978	64 KB	—	—
IBM 3090	Gran computador	1985	128 a 26 KB	—	—
Interl 80486	PC	1989	8 KB	—	—
Pentium	PC	1993	8 KB/8 KB	256 a 512 KB	—
PowerPC 601	PC	1993	32 KB	—	—
PowerPC 620	PC	1996	32 KB/32 KB	—	—
PowerPC 64	PC/servidor	1999	32 KB/32 KB	256 KM 1 MB	2 MB
IBM S/390 G4	Gran computador	1997	32 KB	256 KB	2 MB
IBM S/390 G6	Gran computador	1999	256 KB	8 MB	—
Pentium 4	PC/servidor	2000	8 KB/8 KB	256 KB	—
IBM SP	Servidor de gama alta/Supercomputador	2000	64 KB/32 KB	8 MB	—
CRAY MTA[b]	Supercomputador	2000	8 KB	2 MB	—
Itanium	PC/Servidor	2001	16 KB/16 KB	96 KB	4 MB
SGI Origin 2001	Servidor de gama alta	2001	32 KB/32 KB	4 MB	—
Itanium 2	PC/Servidor	2002	32 KB	256 KB	6 MB
IBM POWER5	Servidor de gama alta	2003	64 KB	1,9 MB	36 MB
CRAY XD-1	Supercomputador	2004	64 KB/64 KB	1 MB	—

[a] Dos valores separados por una barra inclinada (/) hacen referencia a las cachés de instrucciones y de datos.
[b] Ambas cachés son de instrucciones; no caché de datos.

Correspondencia directa. La técnica más sencilla, denominada correspondencia directa, consiste en hacer corresponder cada bloque de memoria principal a solo una línea posible de caché. La Figura 4.7 ilustra el mecanismo general. La correspondencia se expresa como:

$$i = j \text{ módulo } m$$

donde

i = número de línea de caché

j = número de bloque de memoria principal

m = número de líneas en la caché

La función de correspondencia se implementa fácilmente utilizando la dirección. Desde el punto de vista del acceso a caché, cada dirección de memoria principal puede verse como dividida en tres

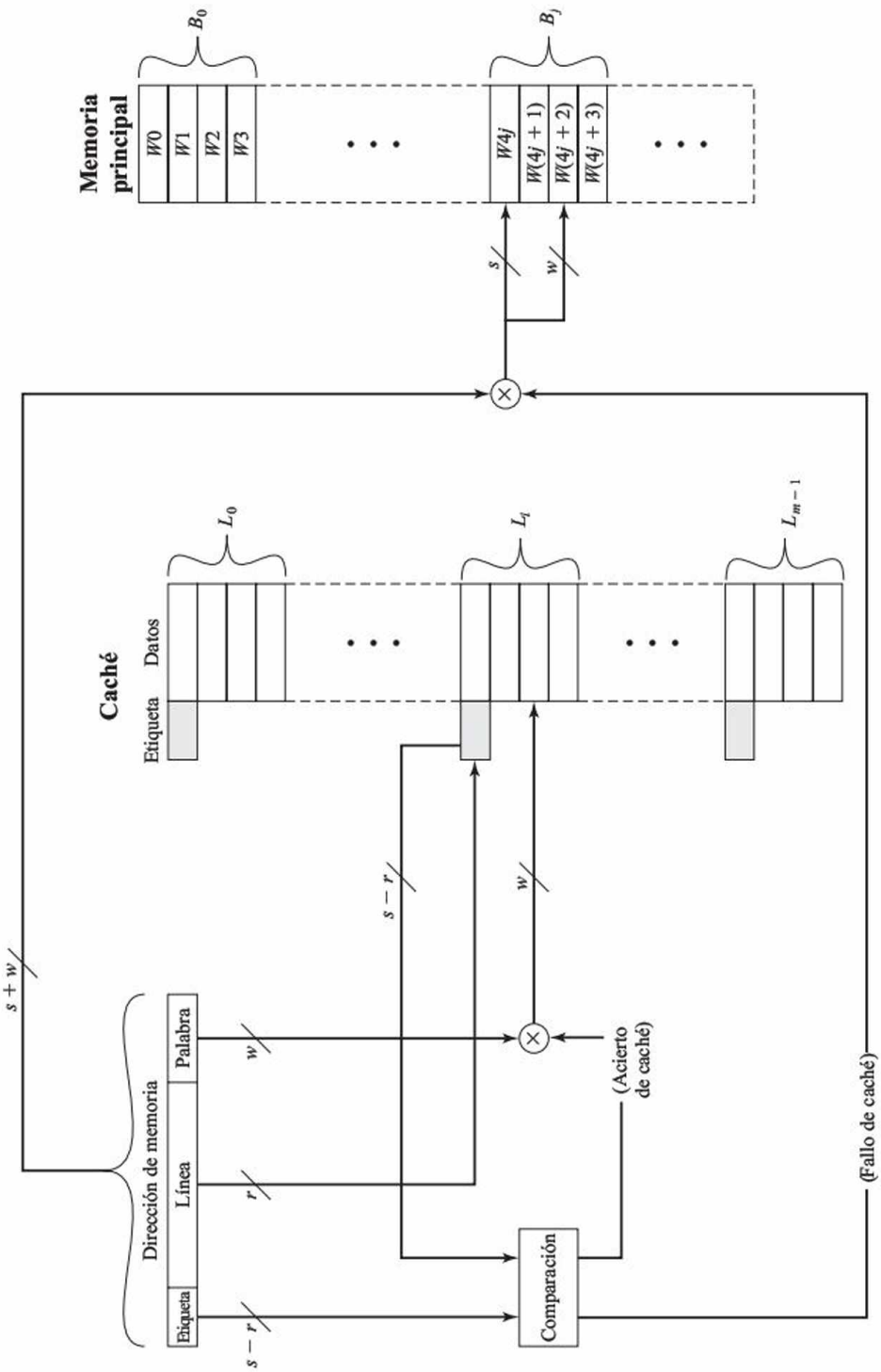


Figura 4.7. Organización de caché con correspondencia directa [HWAN93].

campos. Los w bits menos significativos identifican cada palabra dentro de un bloque de memoria principal; en la mayoría de las máquinas actuales, el direccionamiento es a nivel de bytes. Los s bits restantes especifican uno de los 2^s bloques de la memoria principal. La lógica de la caché interpreta estos s bits como una etiqueta de $s - r$ bits (parte más significativa) y un campo de línea de r bits. Este último campo identifica una de las $m = 2^r$ líneas de la caché.

Resumiendo:

- Longitud de las direcciones = $(s + w)$ bits
- Número de unidades direccionables = 2^{s+w} palabras o bytes
- Tamaño de bloque = tamaño de línea = 2^w palabras o bytes
- Número de bloques en memoria principal = $\frac{2^{s+w}}{2^w} = 2^s$
- Número de líneas en caché = $m = 2^r$
- Tamaño de la etiqueta = $(s - r)$ bits

El resultado es que se hacen corresponder bloques de memoria principal a líneas de caché de la siguiente manera:

Línea de caché	Bloques de memoria principal asignados
0	$0, m, 2m, \ldots, 2^s - m$
1	$1, m + 1, 2m + 1, \ldots, 2^s - m + 1$
.	.
$m - 1$	$m - 1, 2m - 1, 3m - 1, \ldots, 2^s - 1$

Por tanto, el uso de una parte de la dirección como número de línea proporciona una correspondencia o asignación única de cada bloque de memoria principal en la caché. Cuando un bloque es realmente escrito en la línea que tiene asignada, es necesario etiquetarlo para distinguirlo del resto de los bloques que pueden introducirse en dicha línea. Para ello se emplean los s-r bits más significativos.

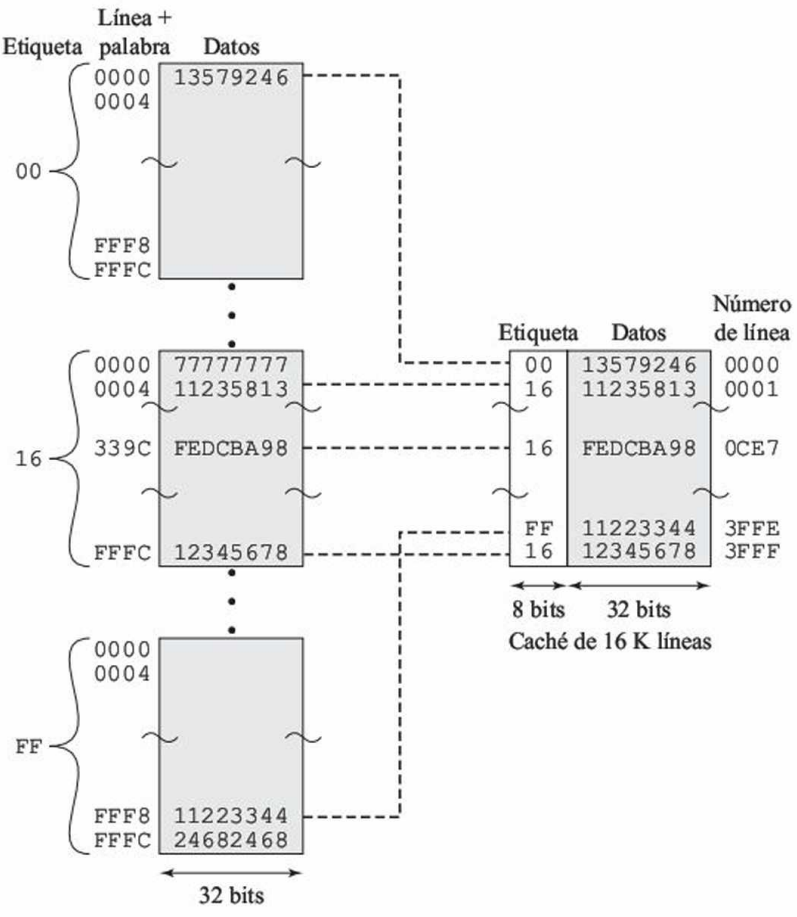
Ejemplo 4.2a. La Figura 4.8 muestra nuestro ejemplo de sistema utilizando correspondencia directa.[4] En el ejemplo: $m = 16\text{K} = 2^{14}$, $i = j$ módulo 2^{14}. La asignación sería:

Línea de caché	Dirección de memoria de comienzo de bloque
0	000000, 010000, ..., FF0000
1	000004, 010004, ..., FF0004
⋮	⋮
$2^{14} - 1$	00FFFC, 01FFFC, ..., FFFFFC

[4] En esta y en figuras posteriores, las direcciones y valores de memoria se expresan en notación hexadecimal. El Apéndice A contiene un resumen de los sistemas de numeración (decimal, binario, hexadecimal).

Obsérvese que no hay dos bloques que se asignen en la misma línea que tengan el mismo número de etiqueta. Así, los bloques 000000, 010000..., FF0000 tienen respectivamente los números de etiqueta 00, 01..., FF.

Refiriéndonos de nuevo a la Figura 4.5, una operación de lectura se lleva a cabo de la siguiente manera. Al sistema de caché se presenta una dirección de 24 bits. El número de línea, de 14 bits, se utiliza como índice para acceder a una línea particular dentro de la caché. Si el número de etiqueta, de 8 bits, coincide con el número de etiqueta almacenado actualmente en esa línea, el número de palabra de 2 bits se utiliza para seleccionar uno de los cuatro bytes de esa línea. Si no, el campo de 22 bits de etiqueta+línea se emplea para captar un bloque de memoria principal. La dirección real que se utiliza para la captación consta de los mencionados 22 bits concatenados con dos bits 0, y se captan 4 bytes a partir del comienzo del bloque.



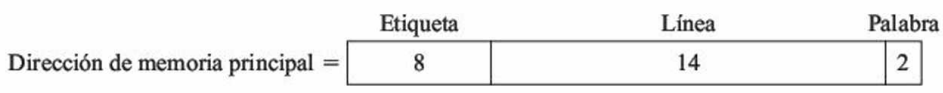


Figura 4.8. Ejemplo de correspondencia directa.

La técnica de correspondencia directa es sencilla y poco costosa de implementar. Su principal desventaja es que hay una posición concreta de caché para cada bloque dado. Por ello, si un programa referencia repetidas veces a palabras de dos bloques diferentes asignados en la misma línea, dichos bloques se estarían intercambiando continuamente en la caché, y la tasa de aciertos sería baja [un fenómeno conocido con el nombre de vapuleo (*thrashing*)].

Correspondencia asociativa. La correspondencia asociativa supera la desventaja de la directa, permitiendo que cada bloque de memoria principal pueda cargarse en cualquier línea de la caché. En este caso, la lógica de control de la caché interpreta una dirección de memoria simplemente como una etiqueta y un campo de palabra. El campo de etiqueta identifica unívocamente un bloque de memoria principal. Para determinar si un bloque está en la caché, su lógica de control debe examinar simultáneamente todas las etiquetas de líneas para buscar una coincidencia. La Figura 4.9 muestra esta lógica. Observe que ningún campo de la dirección corresponde al número de línea, de manera que el número de líneas de la caché no está fijado por el formato de las direcciones. En resumen:

- Longitud de las direcciones = $(s + w)$ bits
- Número de unidades direccionables = 2^{s+w} palabras o bytes
- Tamaño de bloque = tamaño de línea = 2^w palabras o bytes
- Número de bloques en memoria principal = $\frac{2^{s+w}}{2^w} = 2^s$
- Número de líneas en caché = indeterminado
- Tamaño de la etiqueta = s bits

Ejemplo 4.2b. La Figura 4.10 muestra nuestro ejemplo utilizando correspondencia asociativa. Una dirección de memoria principal consta de una etiqueta de 22 bits, más 2 bits que identifican un número de byte. La etiqueta de 22 bits debe almacenarse con el bloque de 32 bits de datos en cada línea de la caché. Obsérvese que son los 22 bits de la izquierda de la dirección (los más significativos) los que forman la etiqueta [5]. De manera que, la dirección de 24 bits 16339C en hexadecimal, contiene la etiqueta de 22 bits 058CE7. Esto se ve fácilmente en notación binaria:

dirección de memoria	0001	0110	0011	0011	1001	1100	(binario)
	1	6	3	3	9	C	(hexadecimal)
etiqueta	00	0101	1000	1100	1110	0111	(binario)
(22 bits de la izda.)	0	5	8	C	E	7	(hexadecimal)

[5] En la Figura 4.10, la etiqueta de 22 bits se representa mediante un número hexadecimal de seis dígitos; de los cuales el dígito más significativo tiene una longitud efectiva de solo dos bits.

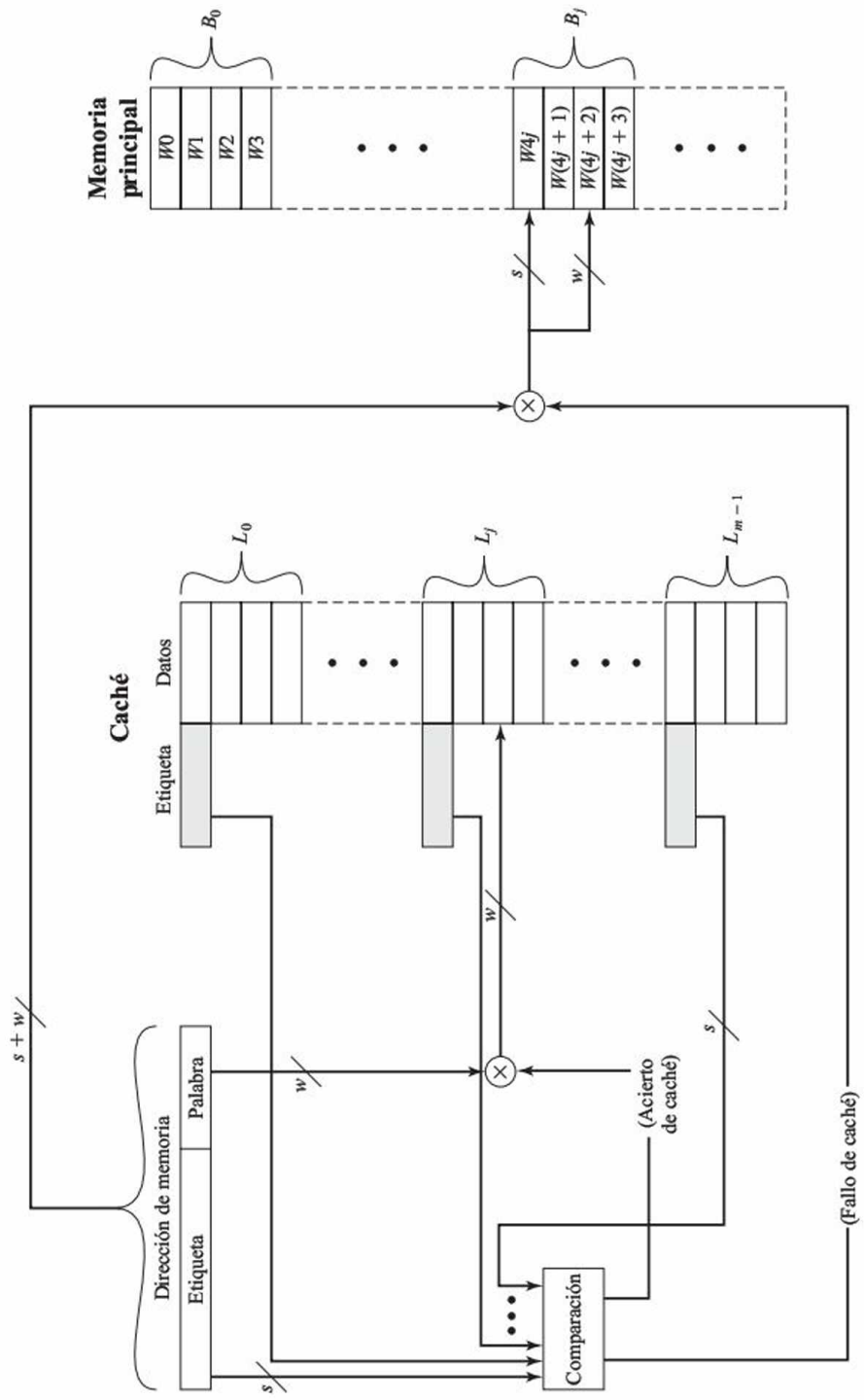


Figura 4.9. Organización de caché totalmente asociativa [HWAN93].

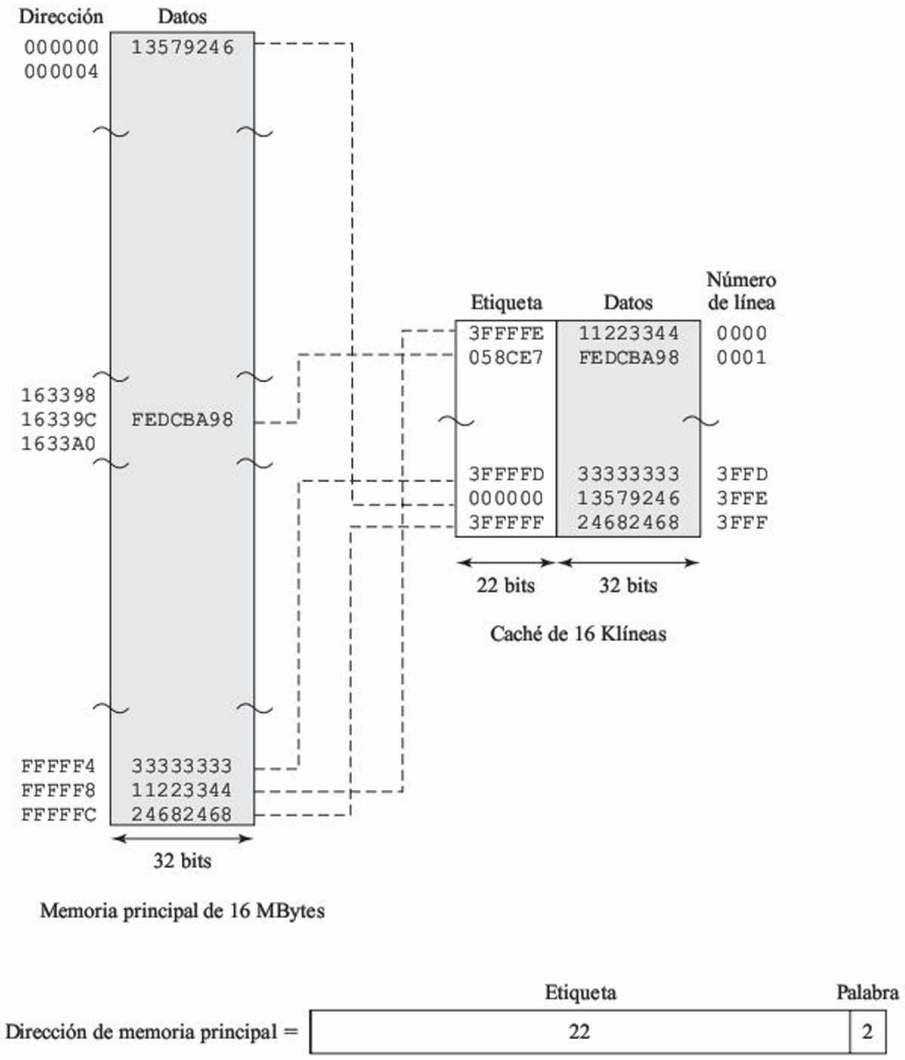


Figura 4.10. Ejemplo de correspondencia asociativa.

Con la correspondencia asociativa hay flexibilidad para que cualquier bloque sea reemplazado cuando se va a escribir uno nuevo en la caché. Los algoritmos de reemplazo o sustitución, discutidos más adelante en esta sección, se diseñan para maximizar la tasa de aciertos. La principal desventaja de la correspondencia asociativa es la compleja circuitería necesaria para examinar en paralelo las etiquetas de todas las líneas de caché.

Correspondencia asociativa por conjuntos. La correspondencia asociativa por conjuntos es una solución de compromiso que recoge lo positivo de las correspondencias directa y asociativa, sin presentar sus desventajas. En este caso, la caché se divide en v conjuntos, cada uno de k líneas. Las relaciones que se tienen son:

$$m = v \times k$$
$$i = j \text{ módulo } v$$

donde

$i =$ número de conjunto de caché
$j =$ número de bloque de memoria principal
$m =$ número de líneas de la caché

En este caso se denomina correspondencia asociativa por conjuntos de k vías. Con la asignación asociativa por conjuntos, el bloque B_j puede asignarse en cualquiera de las k líneas del conjunto i. En este caso, la lógica de control de la caché interpreta una dirección de memoria como tres campos: etiqueta, conjunto y palabra. Los d bits de conjunto especifican uno de entre $v = 2^d$ conjuntos. Los s bits de los campos de etiqueta y de conjunto especifican uno de los 2^s bloques de memoria principal. La Figura 4.11 muestra la lógica de control de la caché. Con la correspondencia totalmente asociativa, la etiqueta en una dirección de memoria es bastante larga y debe compararse con la etiqueta de cada línea en la caché. Con la correspondencia asociativa por conjuntos de k vías, la etiqueta de una dirección de memoria es mucho más corta y se compara solo con las k etiquetas dentro de un mismo conjunto. Resumiendo:

- Longitud de las direcciones $= (s + w)$ bits
- Número de unidades direccionables $= 2^{s+w}$ palabras o bytes
- Tamaño de bloque = tamaño de línea $= 2^w$ palabras o bytes
- Número de bloques en memoria principal $= \frac{2^{s+w}}{2^w} = 2^s$
- Número de líneas en el conjunto $= k$
- Número de conjuntos $= v = 2^d$
- Número de líneas en caché $= kv = k \times 2^d$
- Tamaño de la etiqueta $= (s - d)$ bits

Ejemplo 4.2c. La Figura 4.12 muestra nuestro ejemplo utilizando correspondencia asociativa por conjuntos con dos líneas por cada conjunto, denominada asociativa por conjuntos de dos vías[6]. El número de conjunto, de 13 bits, identifica un único conjunto de dos líneas dentro de la caché. También da el número, módulo 2^{13}, del bloque de memoria principal. Esto determina la asignación de bloques en líneas. Así, los bloques de memoria principal 000000, 008000..., FF8000, se hacen corresponder al conjunto 0 de la caché. Cualquiera de dichos bloques puede cargarse en alguna de las dos líneas del conjunto. Obsérvese que no hay dos bloques que se hagan corresponder al mismo conjunto de la caché que tengan el mismo número de etiqueta. Para una operación de lectura, el número de conjunto, de 13 bits, se utiliza para determinar qué conjunto de dos líneas va a examinarse. Ambas líneas del conjunto se examinan buscando una coincidencia con el número de etiqueta de la dirección a la que se va a acceder.

[6] En la Figura 4.12, la etiqueta de nueve bits se representa mediante un número hexadecimal de tres dígitos. El dígito más significativo tiene una longitud efectiva de solo un bit. El campo de conjunto+palabra, de quince bits, de la dirección de memoria principal, está representado en la figura con números de cuatro dígitos hexadecimales; de los cuales, el más significativo tiene una longitud efectiva de solo tres bits.

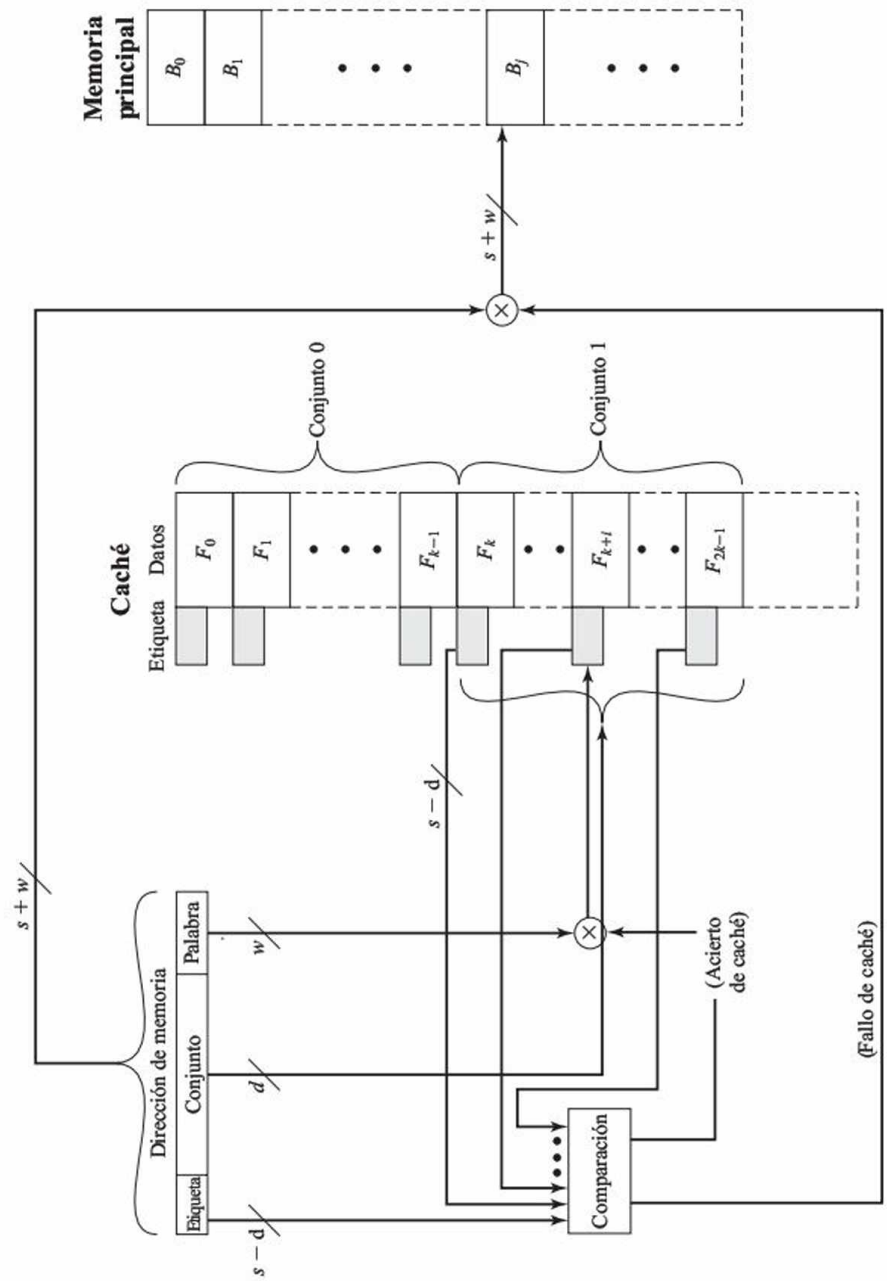


Figura 4.11. Estructura de caché asociativa por conjuntos de *k* vías.

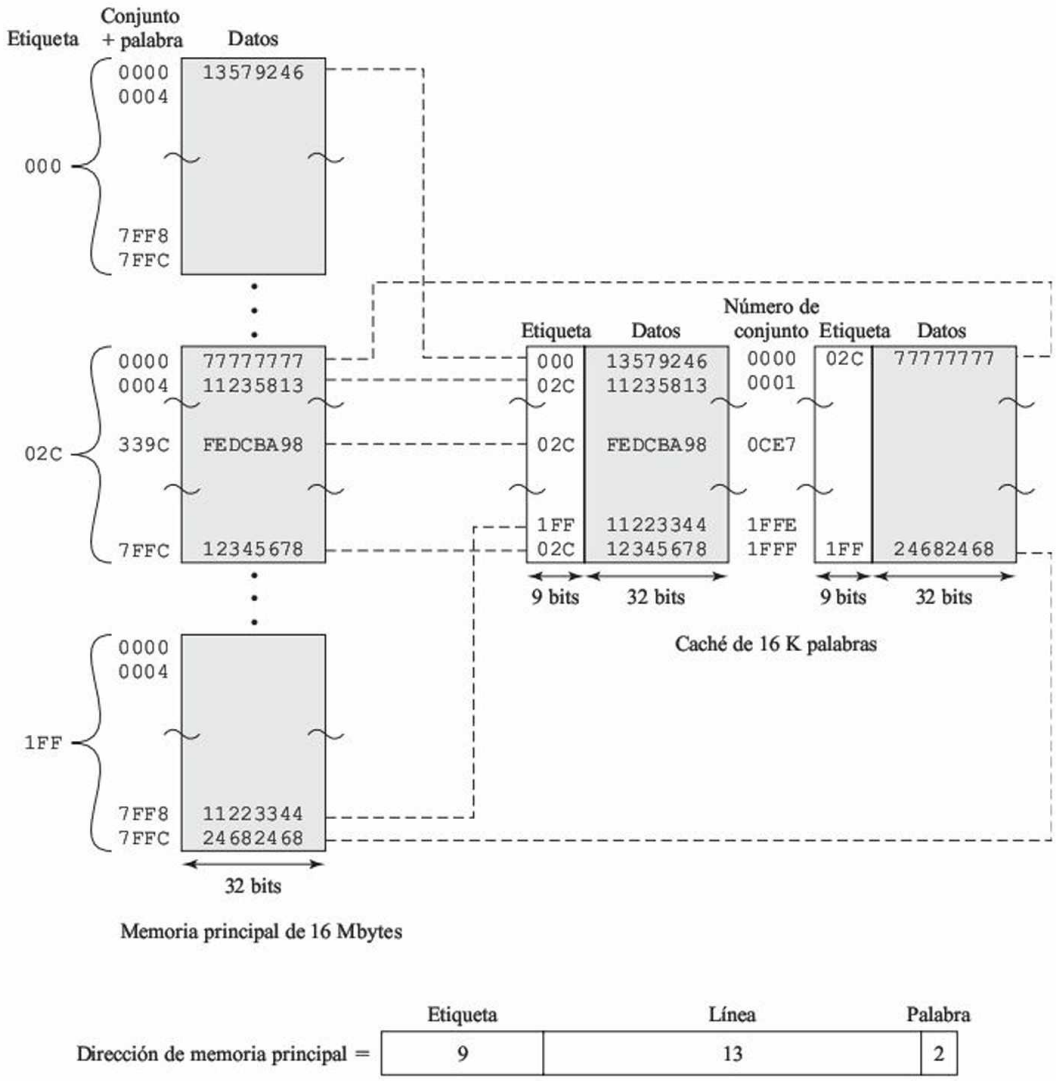


Figura 4.12. Ejemplo de correspondencia asociativa por conjuntos de dos vías.

En el caso extremo de $v = m$, $k = 1$, la técnica asociativa por conjuntos se reduce a la correspondencia directa, y para $v = 1$, $k = m$, se reduce a la totalmente asociativa. El uso de dos líneas por conjunto ($v = m/2$, $k = 2$) es el caso más común, mejorando significativamente la tasa de aciertos respecto de la correspondencia directa. La asociativa por conjuntos de *cuatro vías* ($v = m/4$, $k = 4$) produce una modesta mejora adicional con un coste añadido relativamente pequeño [MAYB84, HILL89]. Un incremento adicional en el número de líneas por conjunto tiene poco efecto.