Gestión de la memoria

Ing. Alejandro C. Rodríguez Costello
(pubdigitalix@gmail.com)

“Funes no solo recordaba cada hoja de cada árbol de cada monte, sino cada una de las veces que había percibido o imaginado.”
Jorge Luis Borges

Objetivos

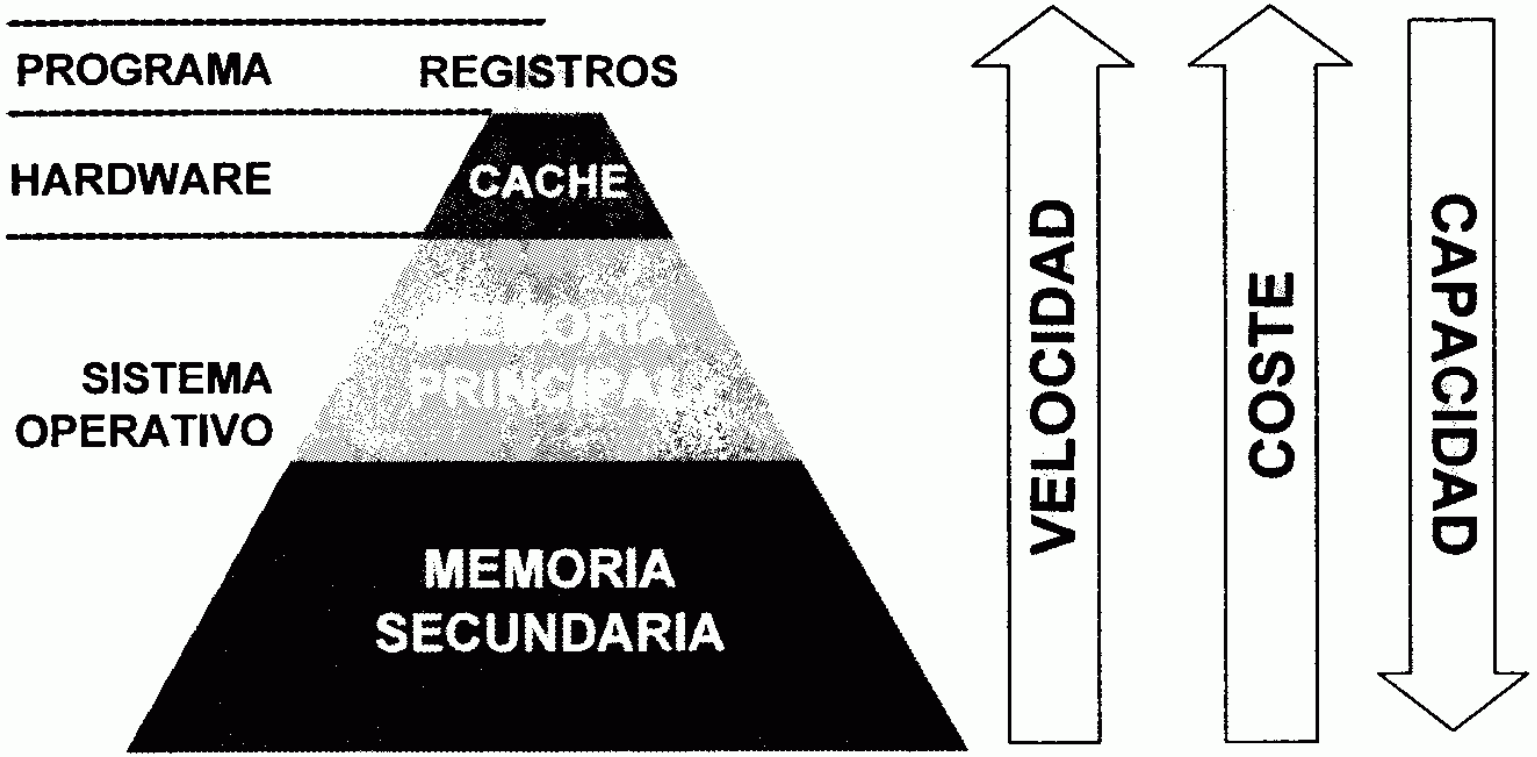
- Definir los tipos de memorias, caracterizarlas y establecer sus principales usos.
- Entender los conceptos relativos a memoria cache.
- Observar tipos y políticas de cache, presentando sus algoritmos que son comunes a otras disciplinas.
- Presentar la influencia que tienen dichos conceptos, sobre las arquitecturas y las ejecuciones de los programas.
- Introducir el concepto de memoria virtual, administración y posibles mejoras que se pueden hacer para elevar sus prestaciones.

Introducción

- El sistema de memoria almacena los programas y datos que requiere la CPU
- Sus prestaciones marcan las del ordenador completo: es un cuello de botella.
- Criterios en el diseño de las memorias
  - ✓ **Coste por bit**
  - ✓ **Velocidad**
  - ✓ **Capacidad**
  - ✓ **Consumo**
  - ✓ **Fiabilidad**
- Estos criterios son incompatibles entre si, con lo que se llega a una solución de compromiso.
- La **JERARQUÍA DE MEMORIA**

Introducción

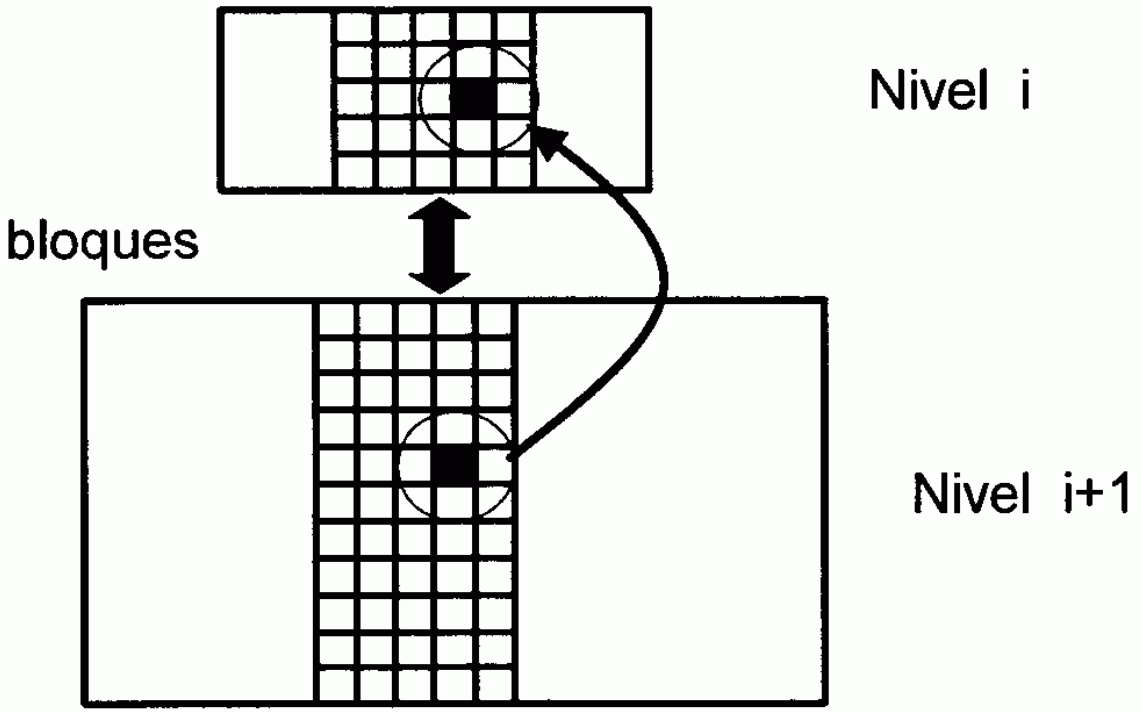
- **Jerarquía de memoria**



Introducción

- **Jerarquía de memoria**

  La información fluye de un nivel inferior a uno superior según va siendo necesaria



Introducción

- **Principio de localidad de las referencias**
  - **Localidad temporal:** Alta probabilidad de volver a hacer referencia a un mismo dato en un corto espacio de tiempo.
  - **Localidad espacial:** Alta probabilidad de que la siguiente referencia sea a un dato almacenado en una posición cercana a la anterior.

    Ejemplo sencillo: Ejecución de una rutina
- **Propiedades de una jeraquía de memoria**
  - Todos los datos contenidos en un nivel se encuentran en un nivel superior.
  - Las copias de un mismo dato en diferentes niveles deben ser coherentes.

Características de la memoria

- Las memorias se caracterizan por su:
  - Ubicación:
    - ✓ Interna: Conectada directamente a la CPU.
      Ejemplo: memoria principal (**MP**)
    - ✓ Externa: Conectada a la CPU a través de E/S.
      Ejemplo: disco rígido
  - Capacidad:
    - ✓ En las memorias externas se expresa en bytes.
    - ✓ En las internas se expresa en bytes o palabras.
  - Unidad de transferencia:
    - ✓ Palabras para la memoria interna.
    - ✓ Bloques para la memoria externa.

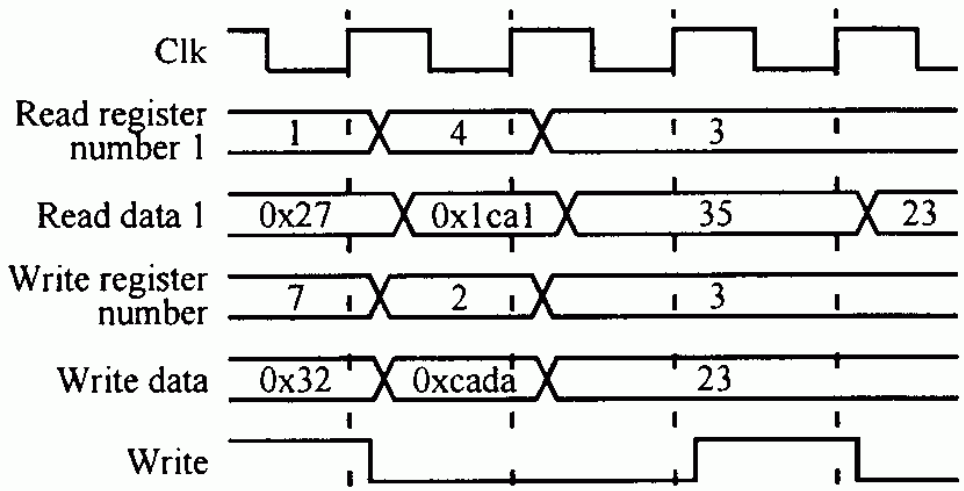
    Recuerde que: La memoria interna se organiza en palabras pero, la unidad direccionable suele ser el byte.

Características de la memoria

- Las memorias se caracterizan por su:
  - Método de acceso:
    - ✓ Secuencial. Cintas.
    - ✓ Directo. Discos.
    - ✓ Aleatorio. Memoria principal y cache.
    - ✓ Asociativo. Tablas de etiquetas para cache.
  - Prestaciones:
    - ✓ Tiempo de acceso (T_a).
      - Variable en secuencial y directo.
      - Fijo en las aleatorias.
    - ✓ Tiempo de ciclo (T_c): se define para las memorias de acceso aleatorio como el tiempo mínimo entre dos accesos consecutivos.
    - ✓ Velocidad de transferencia: se define como $1/T_c$ para las aleatorias.

Memoria de acceso aleatorio

- **Tipos de memoria de acceso aleatorio:**
  - ✓ Registros internos.
  - ✓ SRAM (cache).
  - ✓ DRAM (memoria principal).
  - ✓ ROM (memoria no volátil). BIOS, etc.
- **Banco de registros**



Banco de registros. Temporización.

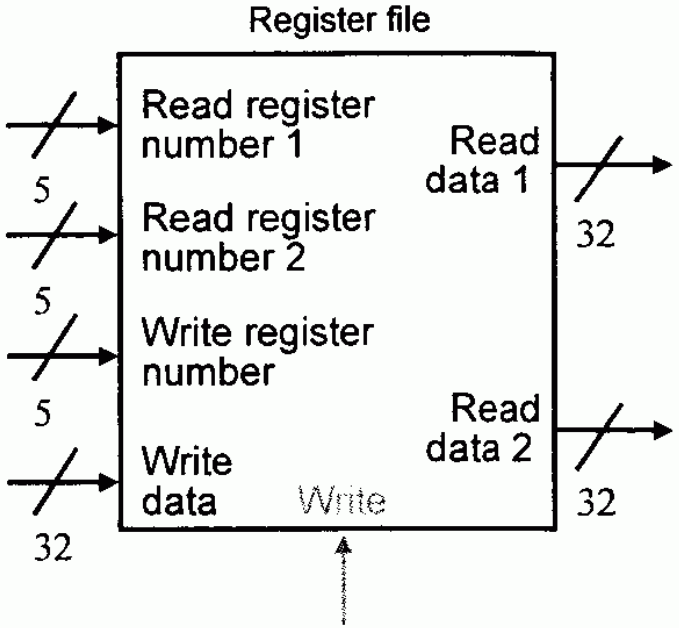
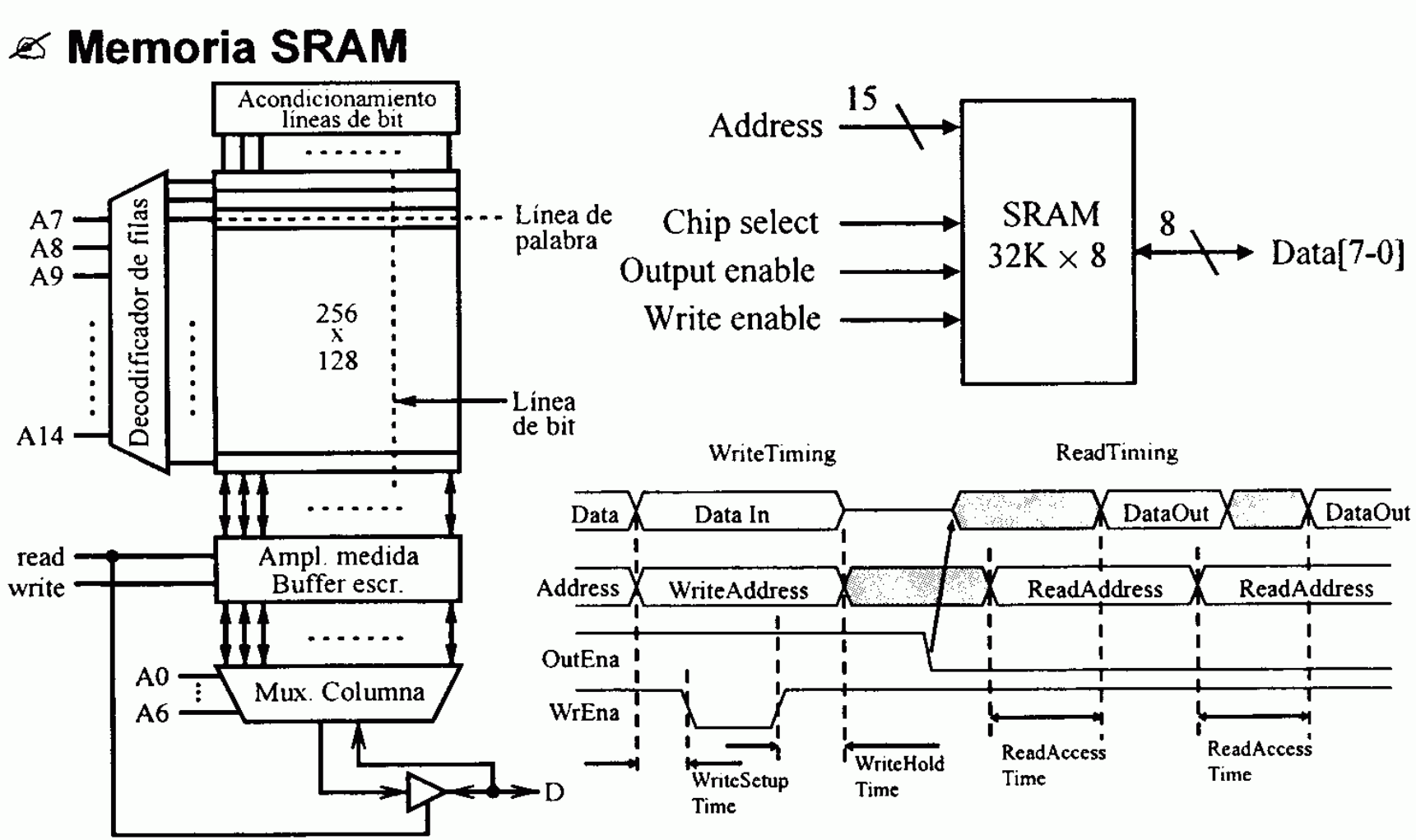
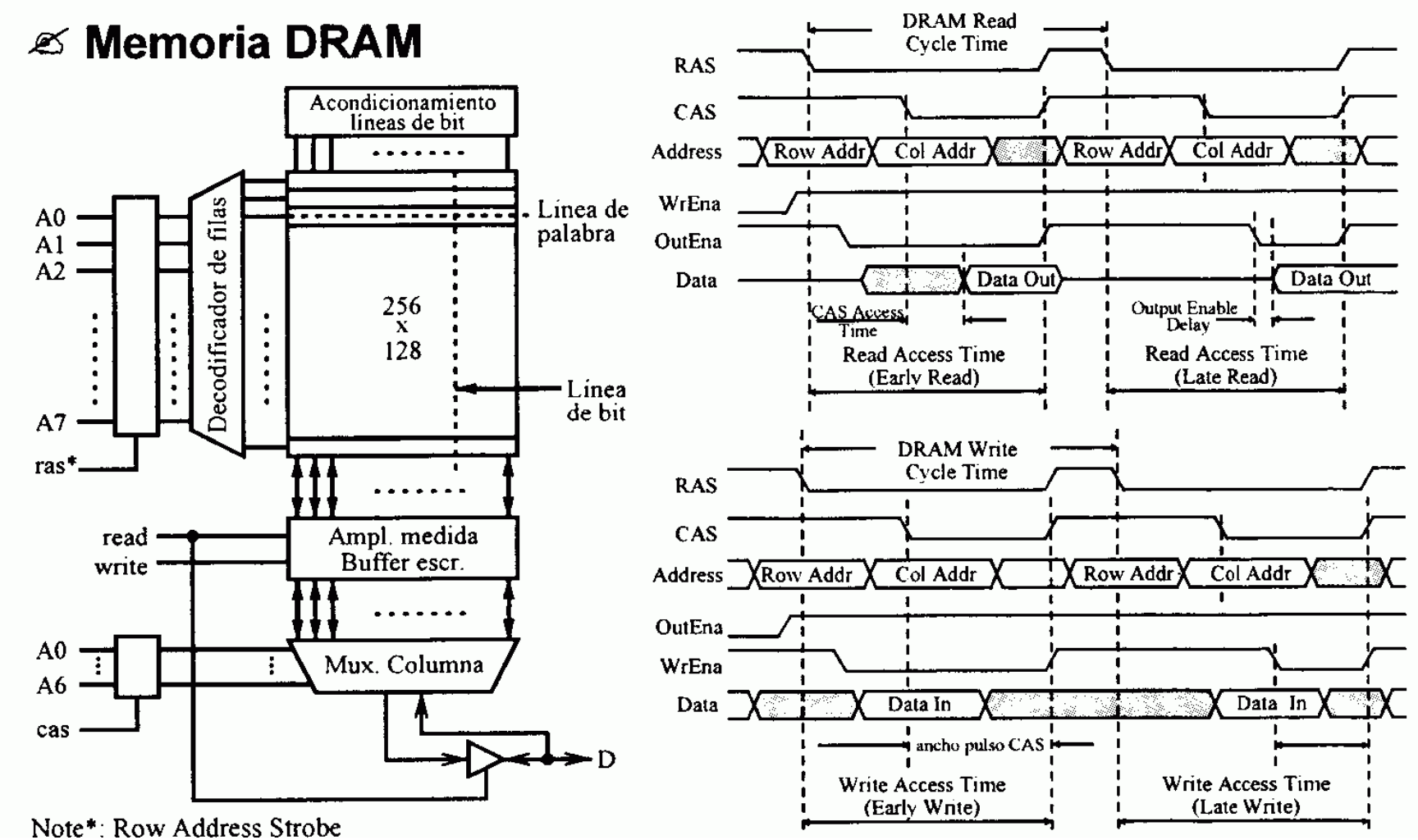


Diagrama.

Memoria de acceso aleatorio

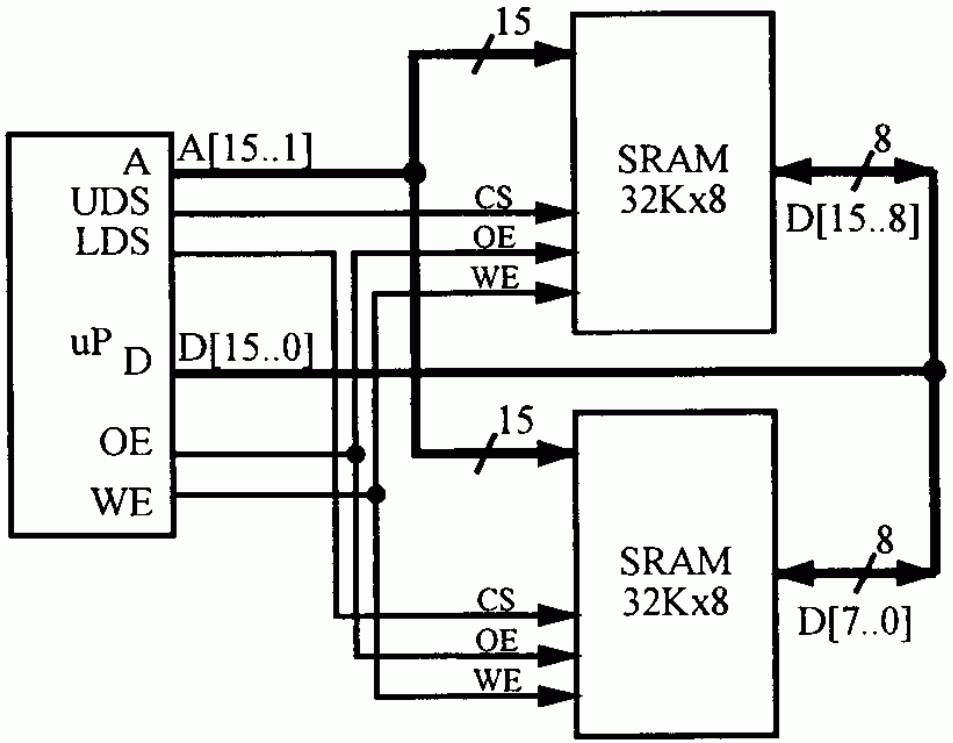


Memoria de acceso aleatorio



Organización de la memoria

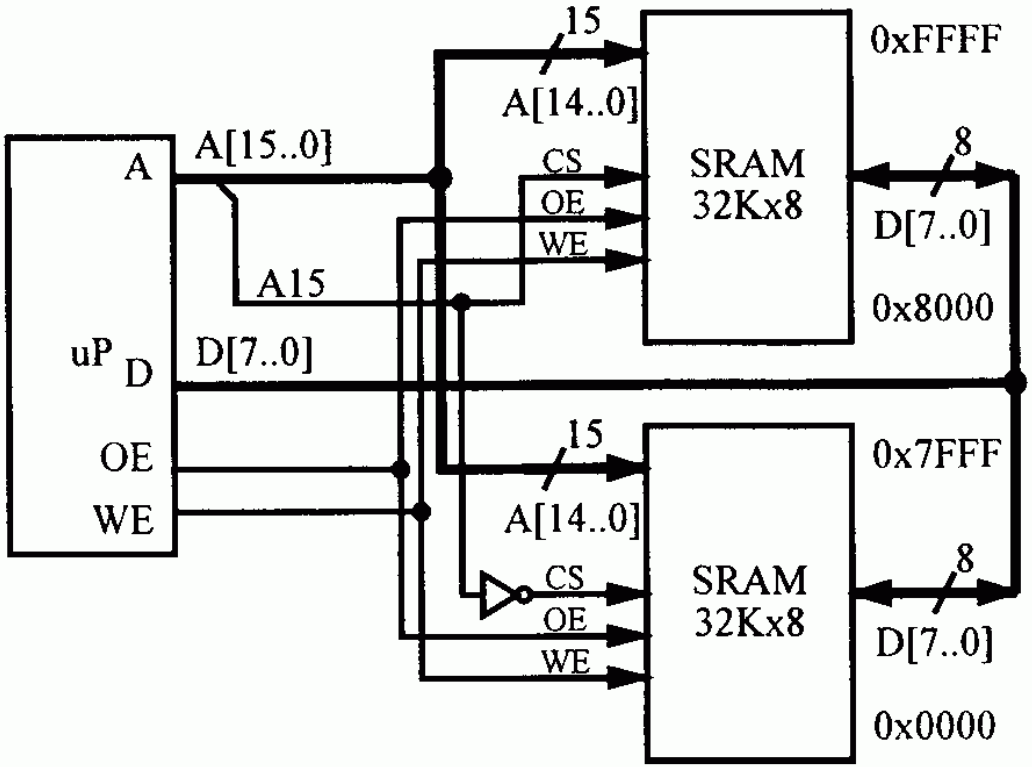
- Sea un sistema microprocesador con 64 kB organizado en palabras de 16 bits, solo se dispone de chips 32 k x 8.



Notas: UDS Upper Data Strobe, LDS Lower Data Strobe, CS Chips Select

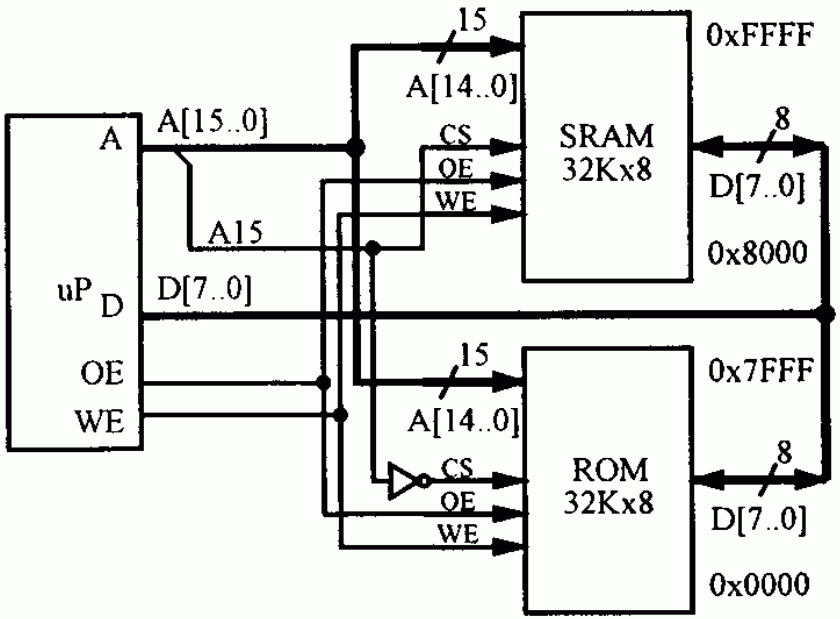
Organización de la memoria

- Sea un sistema microprocesador con 64 kB organizado en palabras de 16 bits, solo se dispone de chips 32 k x 8, pero el bus de datos tiene 8 bits.

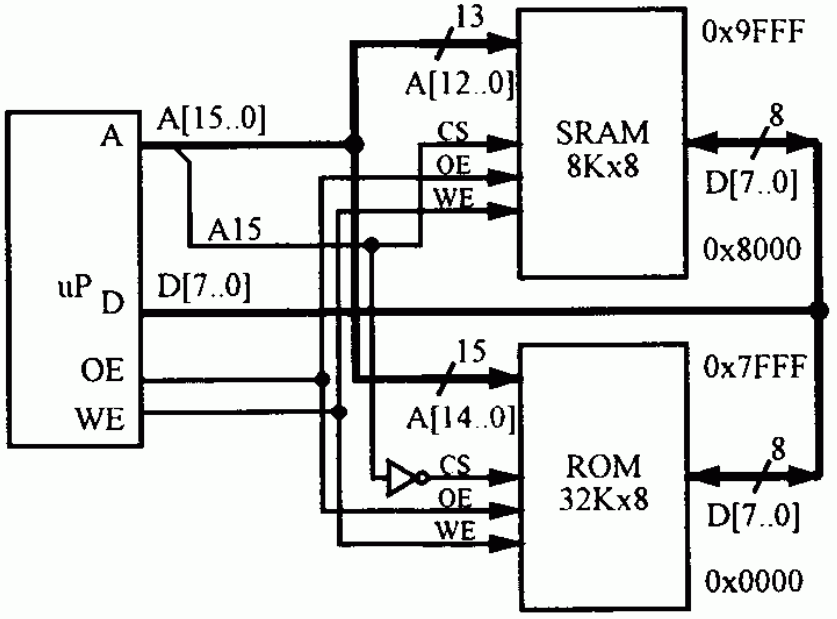


Organización de la memoria

✍ **Otros ejemplos.**



Ejemplo con RAM y ROM



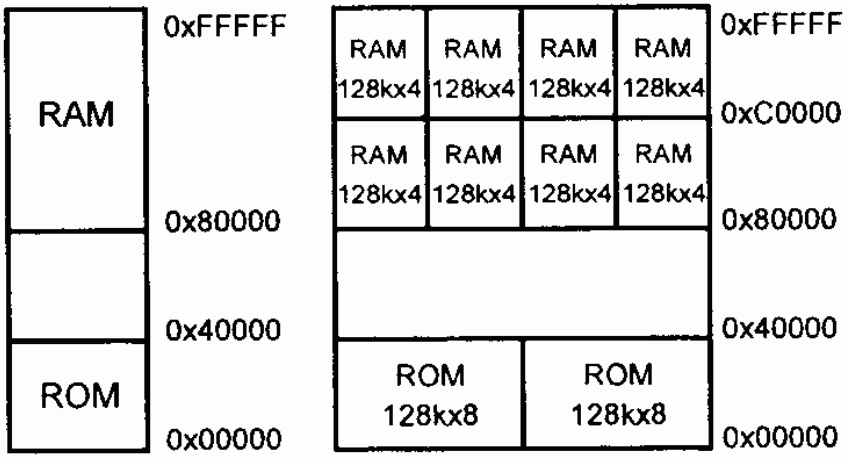
Ejemplo con RAM (ghost) y ROM

En el segundo ejemplo los 8 kB de RAM aparecen repetidos en las direcciones 0xA000-0xBFFF, 0xC000-0xDFFF y finalmente en 0xE000-0xFFFF.
Se dice entonces que dichas posiciones son fantasmas de la memoria situada en 0x8000-0x9FFF.

Organización de la memoria

Ejercicio:

- Se dispone de un microprocesador con un bus de direcciones de 20 bits y un bus de datos de 16 bits:
- ¿Cual es el tamaño máximo de memoria que puede usar?
- Se necesita disponer de una memoria RAM de 512 kB y una ROM de 256 kB. Se dispone de chips de ROM de 128 k x 8 y chips de RAM de 128 k x 4. La memoria ROM ha de situarse en las posiciones bajas del mapa de memoria y la RAM en las posiciones altas.

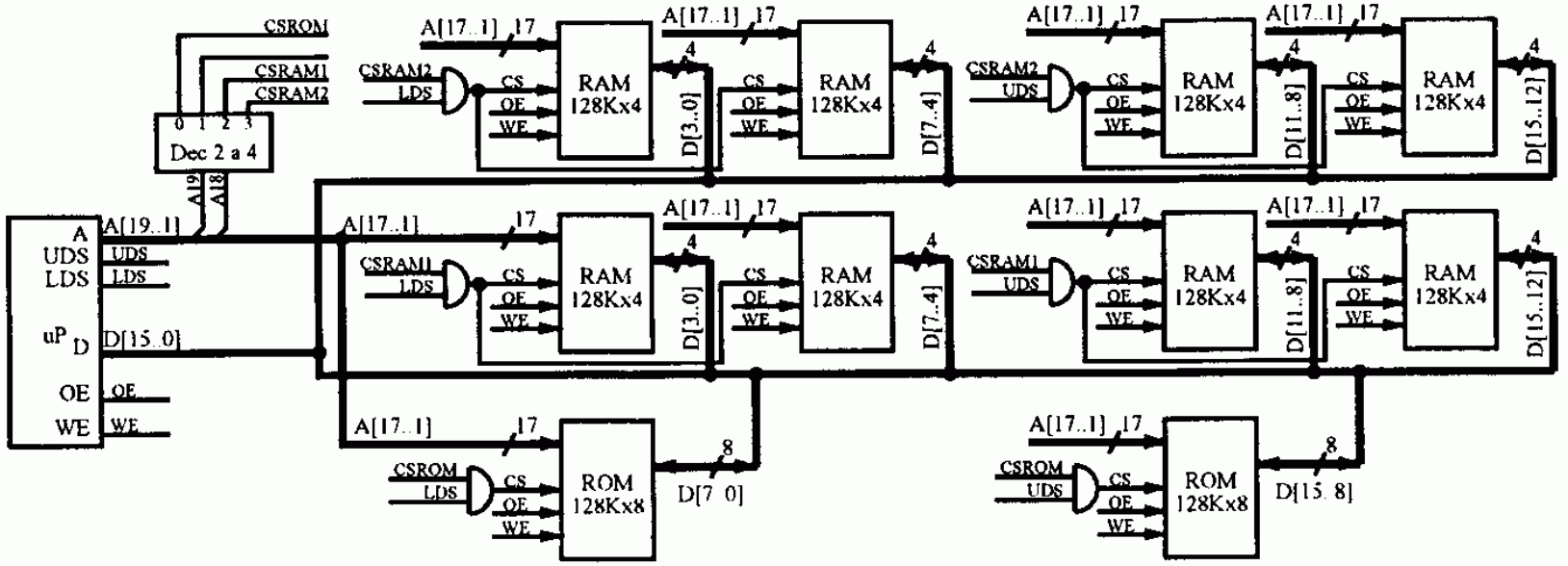


Mapa de memoria:

Dirección	Memoria
00XX XXXX XXXX XXXX XXXX	ROM
01XX XXXX XXXX XXXX XXXX	No usado
10XX XXXX XXXX XXXX XXXX	RAM 1
11XX XXXX XXXX XXXX XXXX	RAM 2

Organización de la memoria

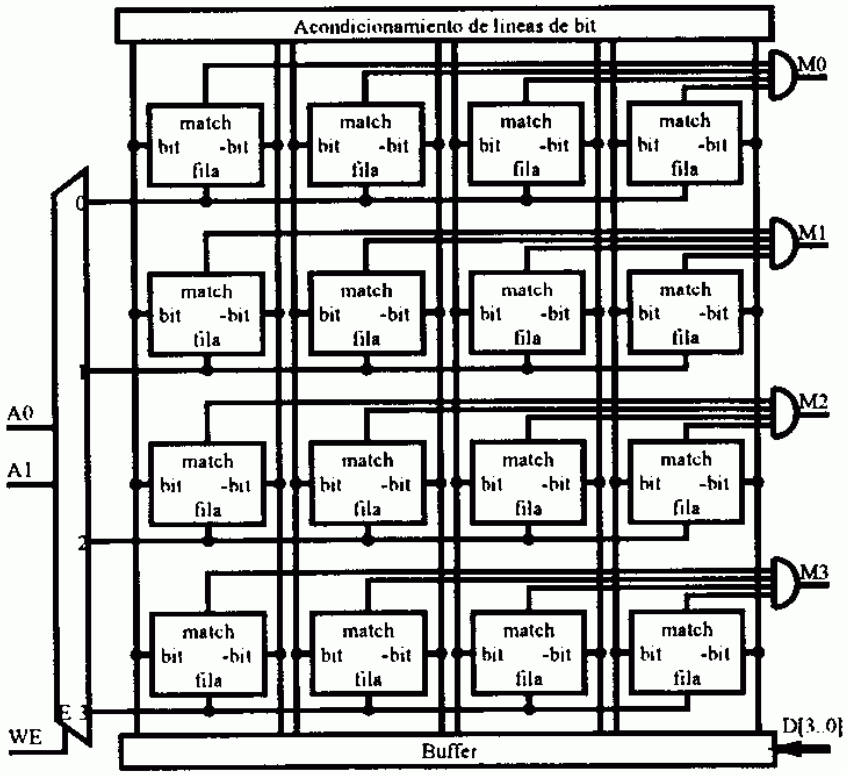
✍ Solución:



Organización de la memoria

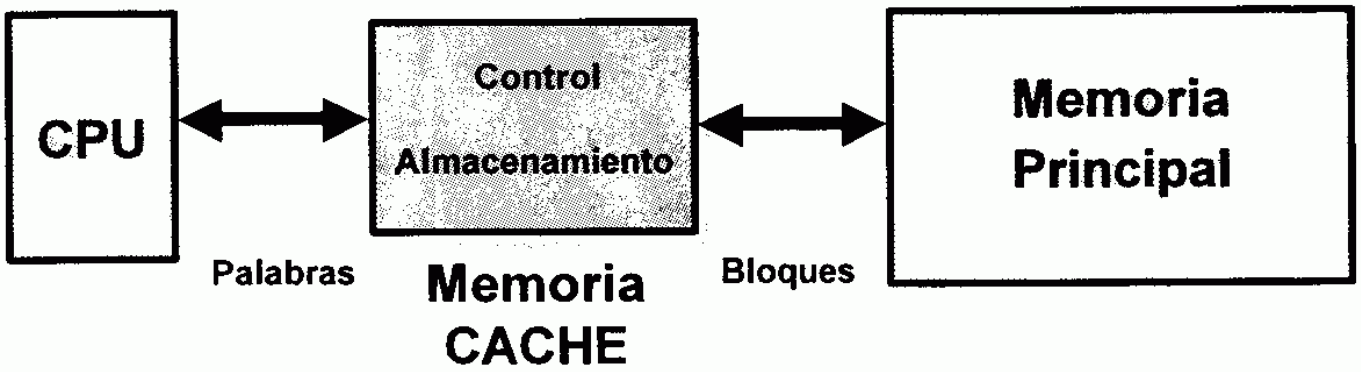
Memoria asociativa (CAM)

Las memorias CAM (*content addressable memory*), permiten conocer, según su tipo, si un dato está almacenado en ella o en qué posiciones está.



Memoria cache

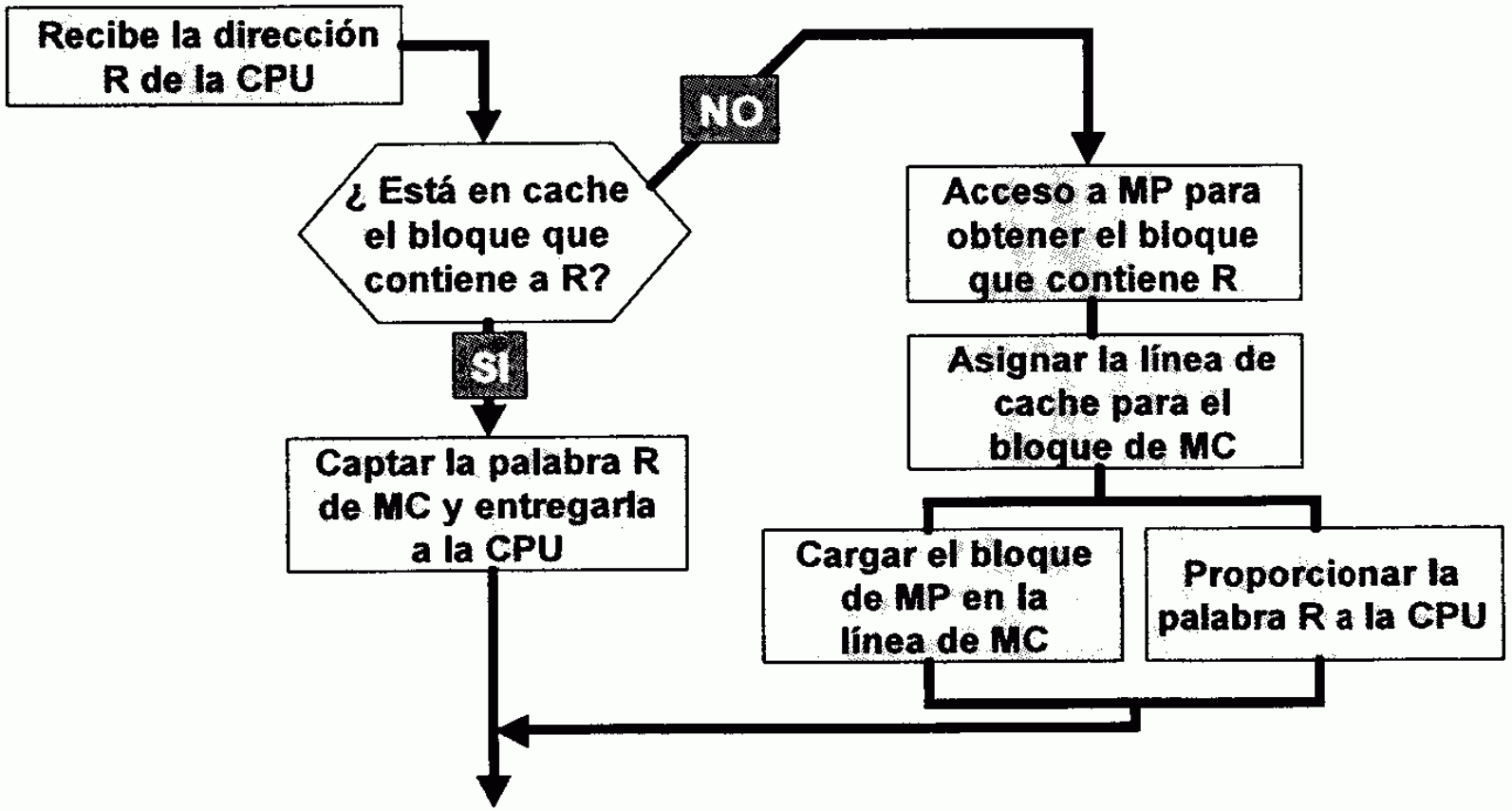
Estructura y funcionamiento



- La unidad de transferencia entre memoria principal (**MP**) y memoria cache (**MC**) es el bloque o línea.
- Cada bloque está constituido por un conjunto de 2^r palabras
- La memoria cache y la memoria central están divididas en líneas o bloques de igual tamaño.

Memoria cache

Estructura y funcionamiento



Memoria cache

- **Estructura y funcionamiento**
  - **Organización**
    - ✓ **¿cómo se sabe si una posición de MP se encuentra copiada en una posición de MC?**
    - ✓ **¿dónde se copian las posiciones de MP en MC?**
  - **Adquisición**
    - ✓ **¿cuándo se lleva una lineade MP a MC?**
  - **Actualización**
    - ✓ **¿cuándo se actualiza el contenido de MC en MP?**
  - **Reemplazo**
    - ✓ **¿qué línea de cache se desprecia cuando se necesita espacio en MC para otra diferente?**

Memoria cache

- **Organización**
  - Cada línea de cache está compuesta por tres campos:
    - ✓ **Bit de validez**
    - ✓ **Información de etiqueta. Para realizar la correspondencia entre la línea de cache y el bloque de MP que almacena.**
    - ✓ **Datos**

Bit de validez

  - Localización de un bloque de memoria central en una línea de la memoria cache
    - ✓ **Mapeado directo**
    - ✓ **Completamente asociativa**
    - ✓ **Asociativa por conjuntos**

Cache de mapeo directo

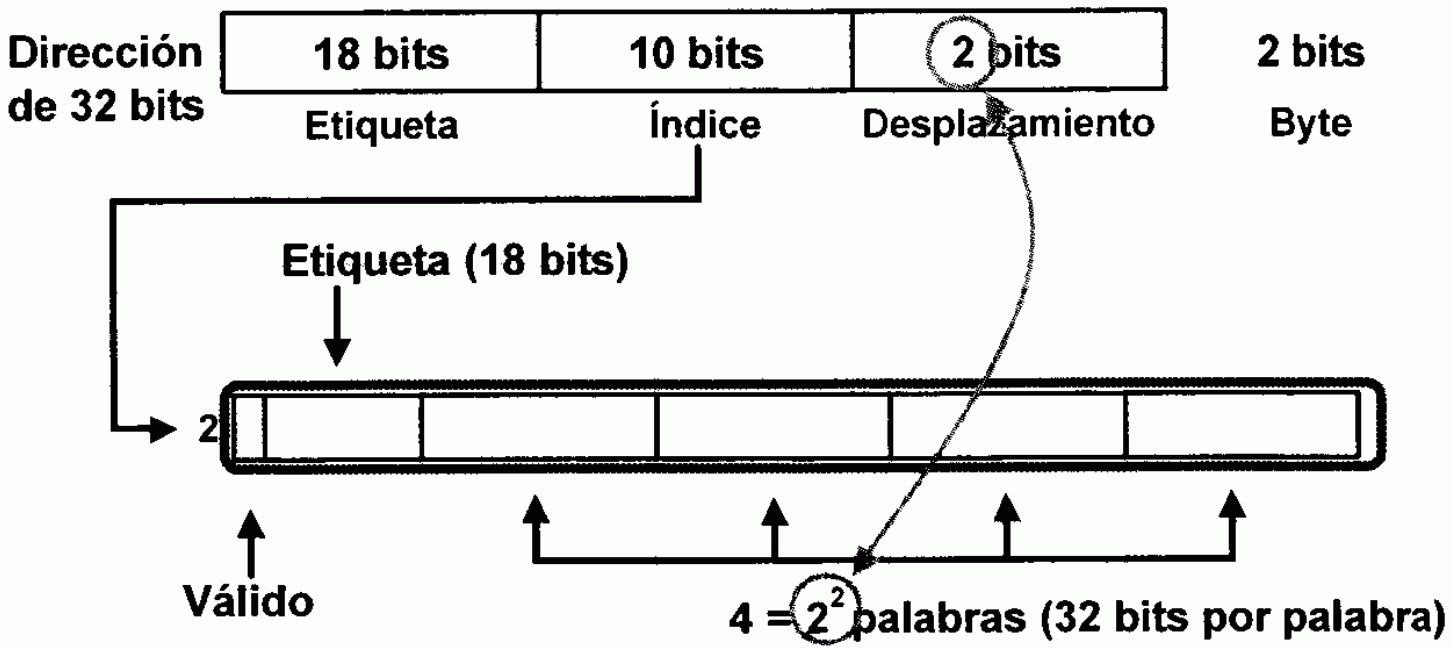
- **Organización**
  - Cada bloque de MP se puede almacenar solo en una determinada línea de MC.
  - Si se tienen **m** líneas de cache

SI SE ELIGE m COMO UNA POTENCIA DE 2

Línea	Bloques de MP asignados Todos los que sus últimos $x=\log_2 m$ bits contienen
0	0+0 (0x0....00) , m+0 , 2m+0 ,..., 2^im+0
1	0+1 (0x0....01) , m+1 , 2m+1 ,..., 2^im+1
...	... ...
m-1	0+m-1 (0xF....FF) , m+(m-1) , 2m+(m-1) ,..., 2^i m+(m-1)

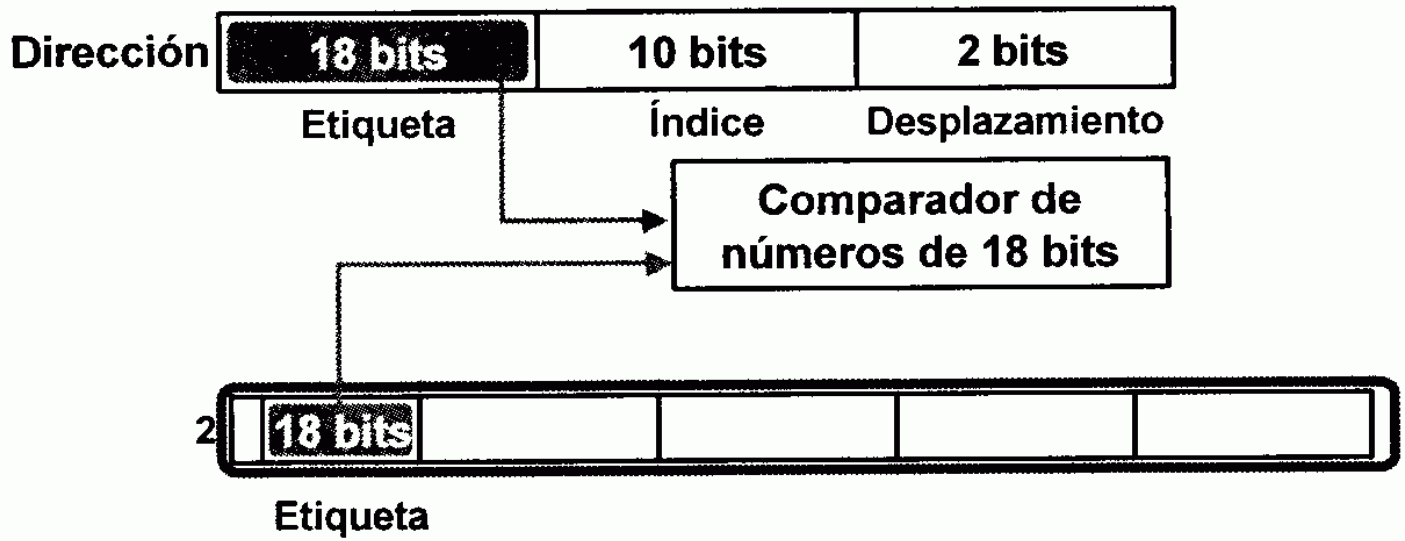
Cache de mapeo directo

- **Organización**
  - ¿Cómo se sabe, dada una dirección de MP en qué línea de MC se encuentra?



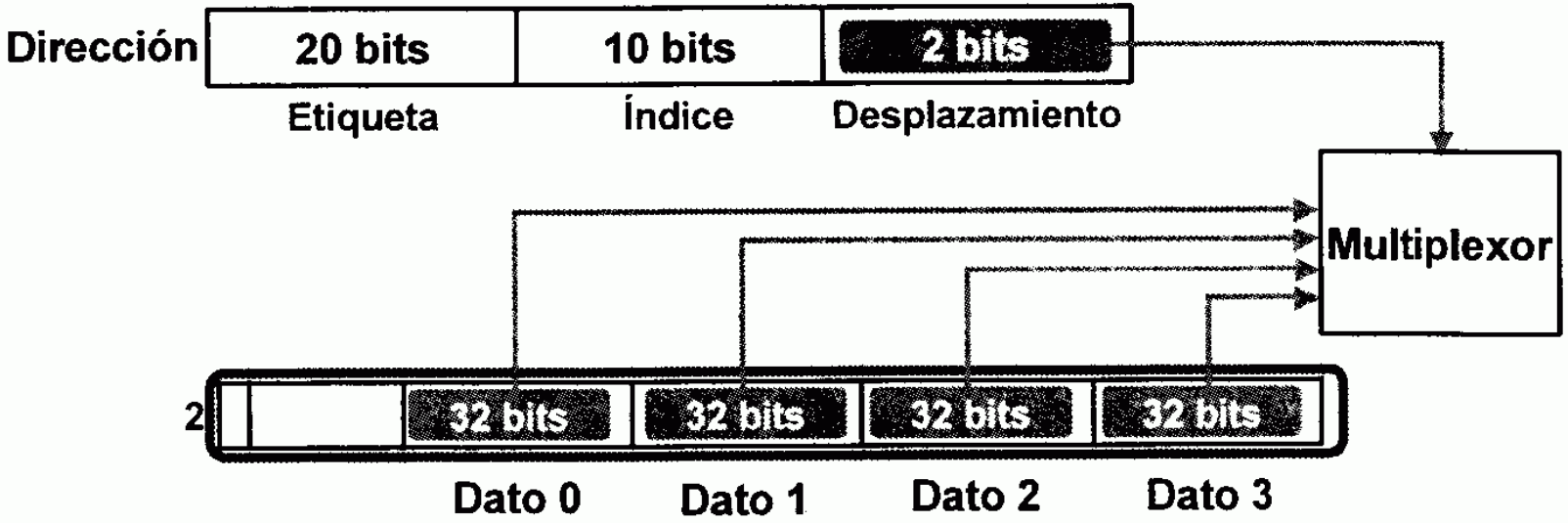
Cache de mapeo directo

- **Organización**
  - ¿Cómo se sabe, dada una dirección de MP en qué línea de MC se encuentra?



Cache de mapeo directo

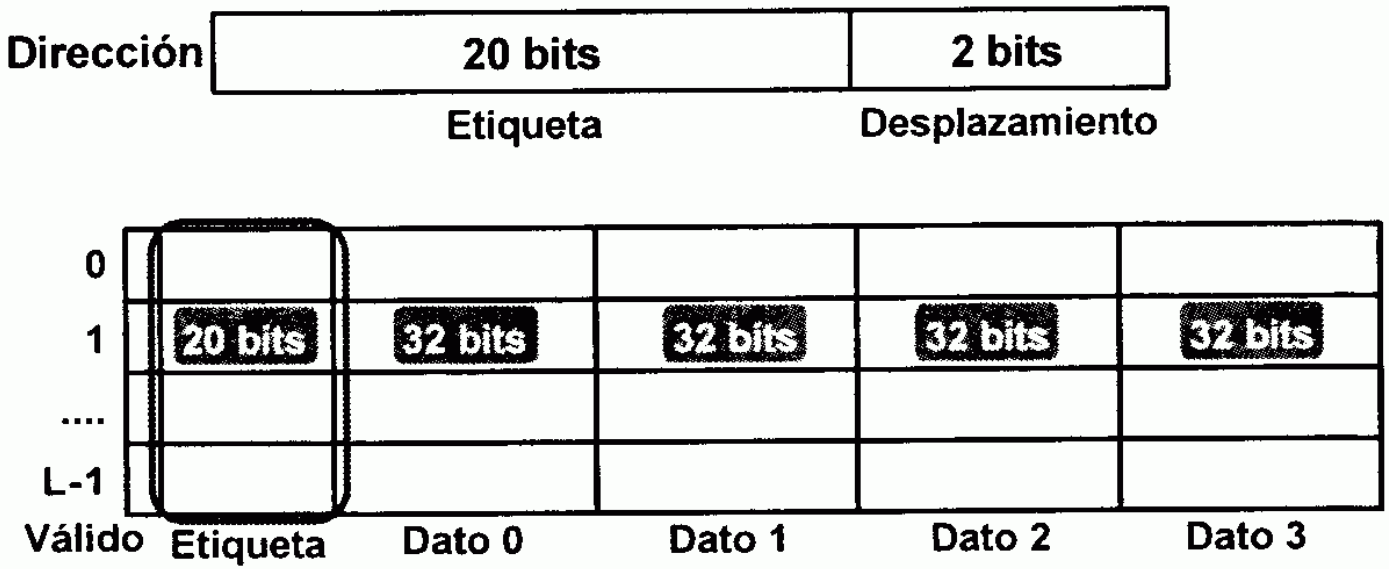
- **Organización**
  - ¿Cómo se sabe, dada una dirección de MP en qué línea de MC se encuentra?



- ✓ Simple y fácil de implementar
- ✓ Posición concreta de cache para cada bloque dado

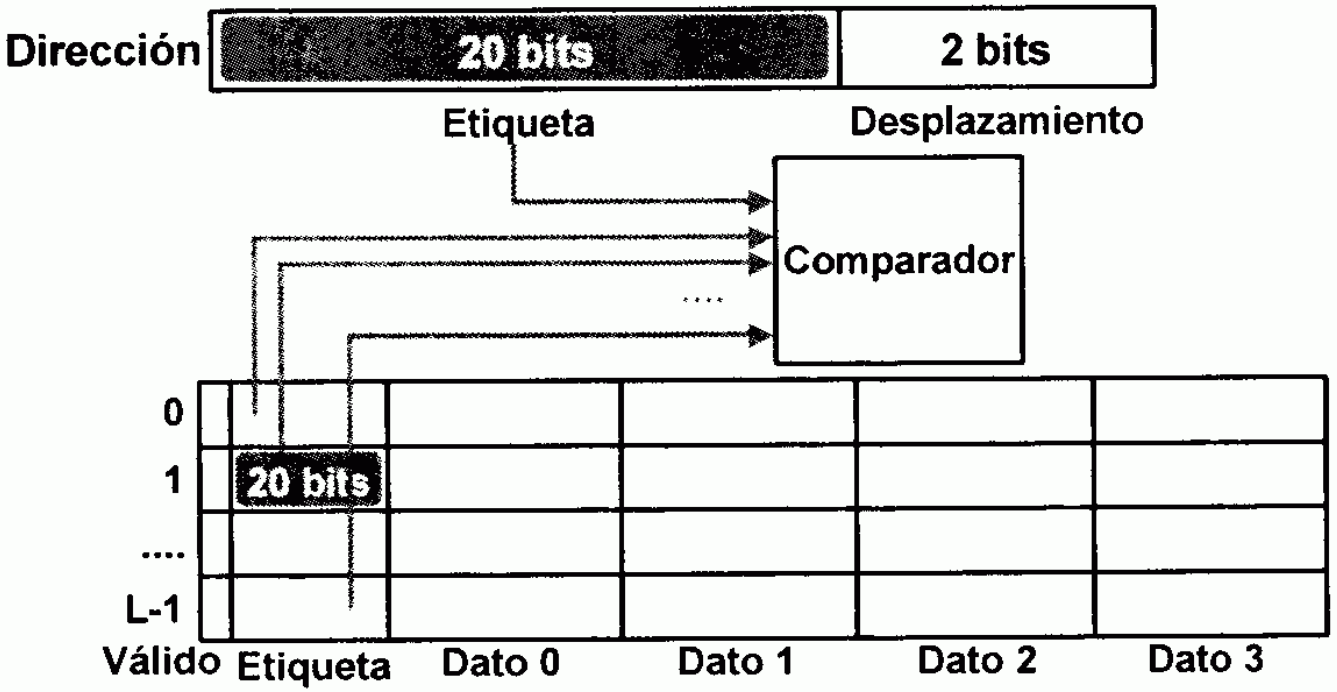
Cache asociativa

- **Organización**
  - Los bloques de MP se pueden almacenar en cualquier línea de MC



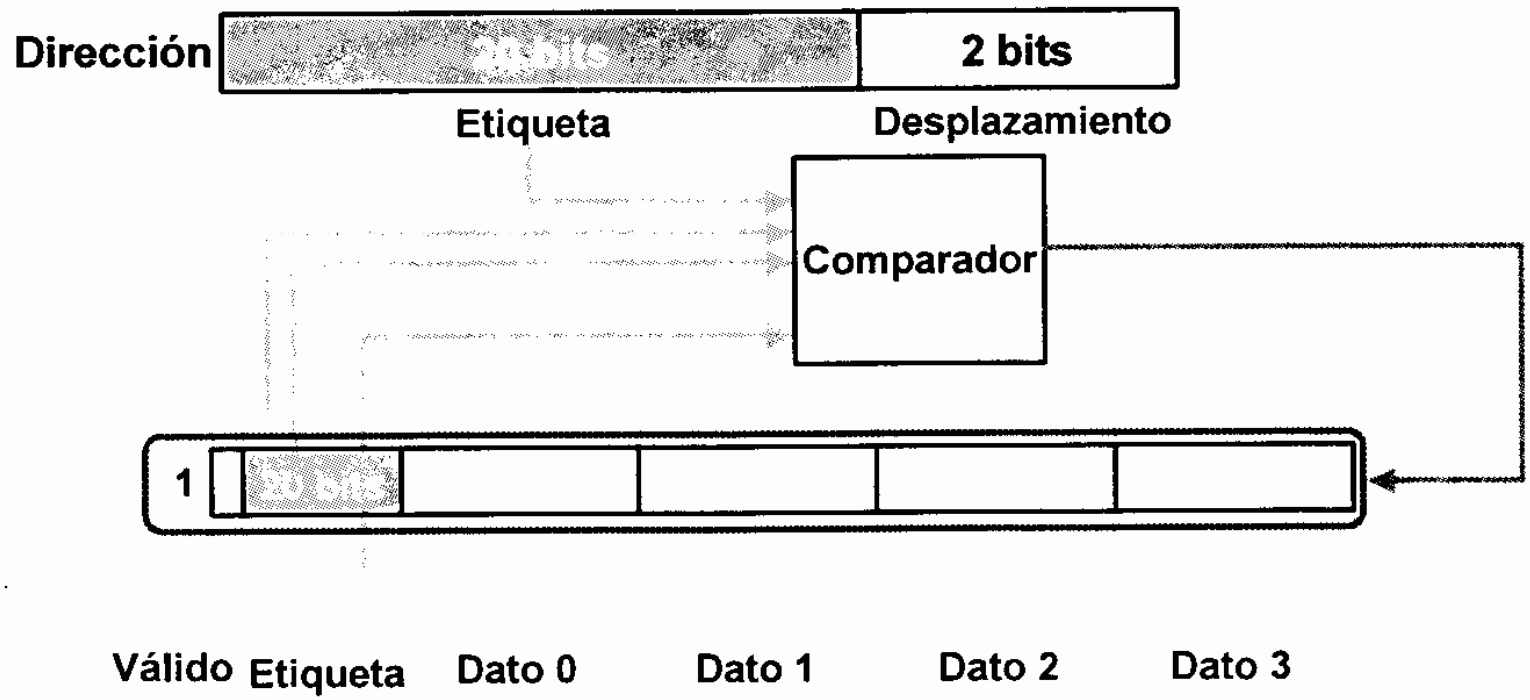
Cache asociativa

- **Organización**
  - ¿Cómo se sabe, dada una dirección de MP en qué línea de MC se encuentra?



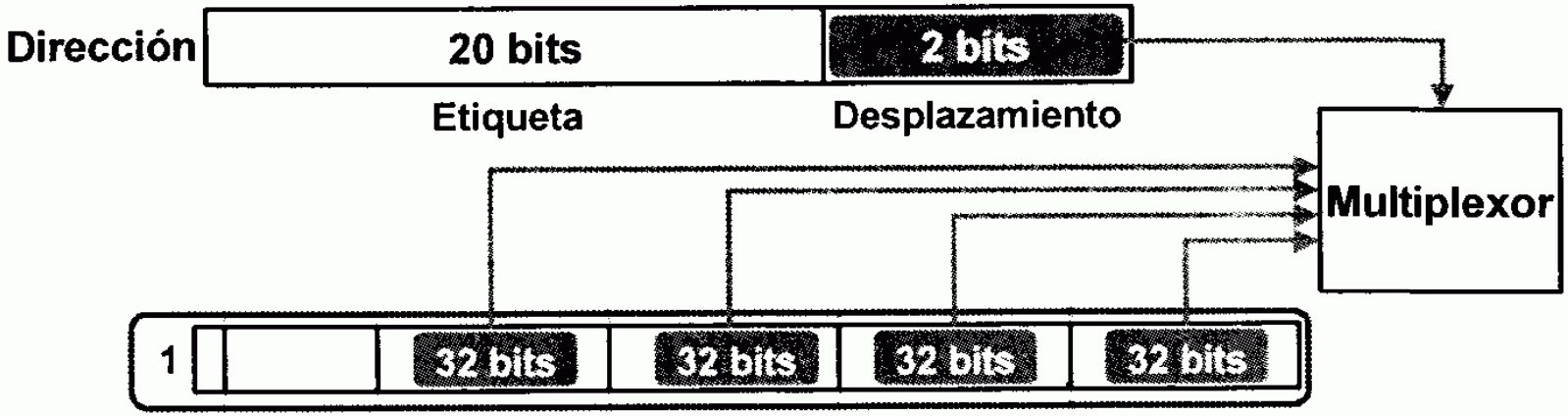
Cache asociativa

- **Organización**
  - ¿Cómo se sabe, dada una dirección de MP en qué línea de MC se encuentra?



Cache asociativa

- **Organización**
  - ¿Cómo se sabe, dada una dirección de MP en qué línea de MC se encuentra?



Válido Etiqueta Dato 0 Dato 1 Dato 2 Dato 3

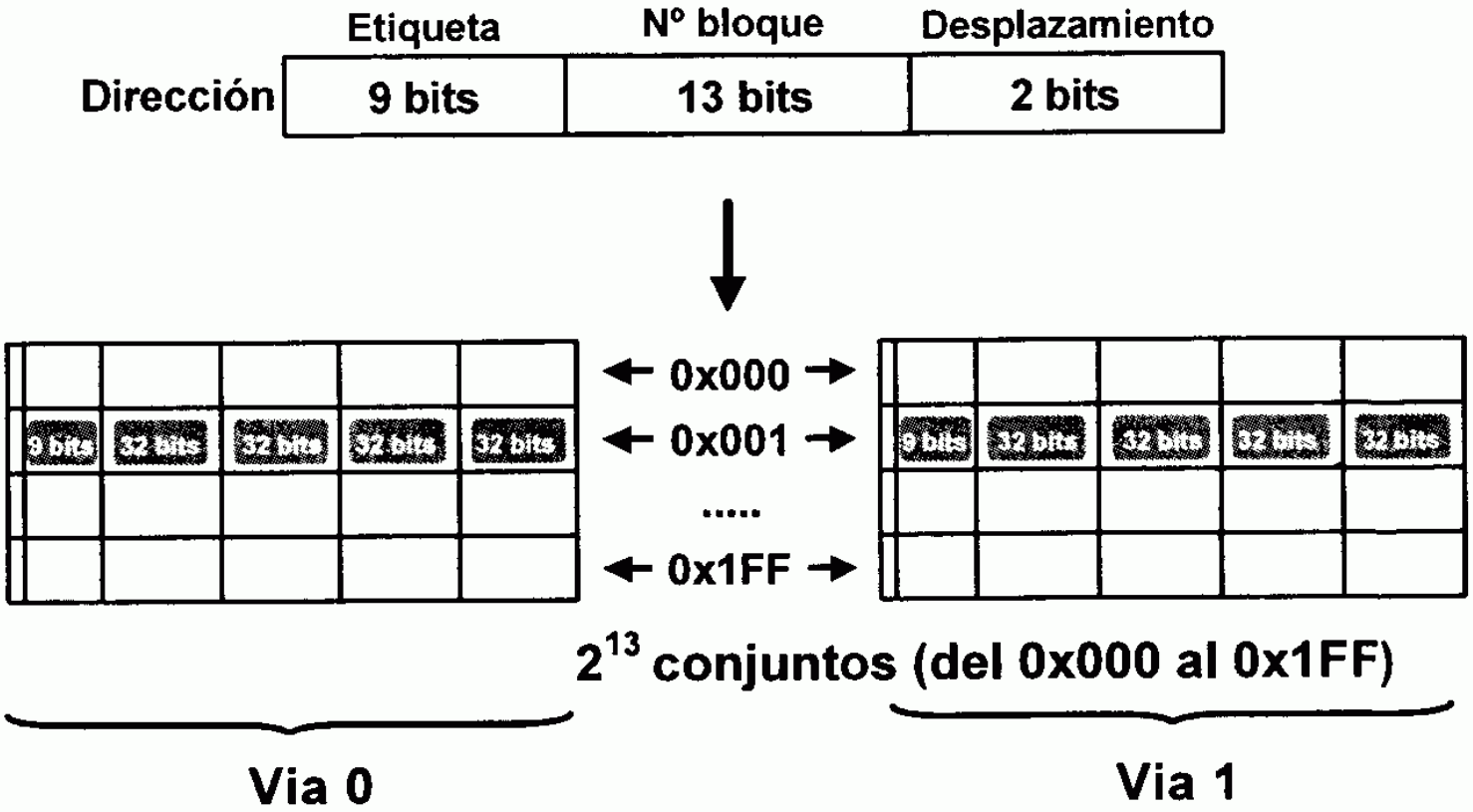
  - Flexibilidad para que cualquier bloque se encuentre en cualquier línea de MC vs circuitería muy compleja para poder examinar en paralelo todas las etiquetas.

Cache asociativa por conjunto

- **Organización**
  - **Solución de compromiso entre la organización de mapeado directo y la completamente asociativa**
  - **La cache se divide en v conjuntos, cada uno con k líneas, cumpliéndose que:**
    - ✓ **m = v x k**
      - **m = número de líneas de cache**
    - ✓ **i = j módulo v**
      - **i es el número de conjunto de cache**
      - **j es el número de bloque de MP**
  - **k-asocitiva o asociativa por conjunto de k vías**
  - **Un determinado bloque de MP se puede cargar en cualquiera de las líneas de cache del conjunto de lineas de cache asociado.**

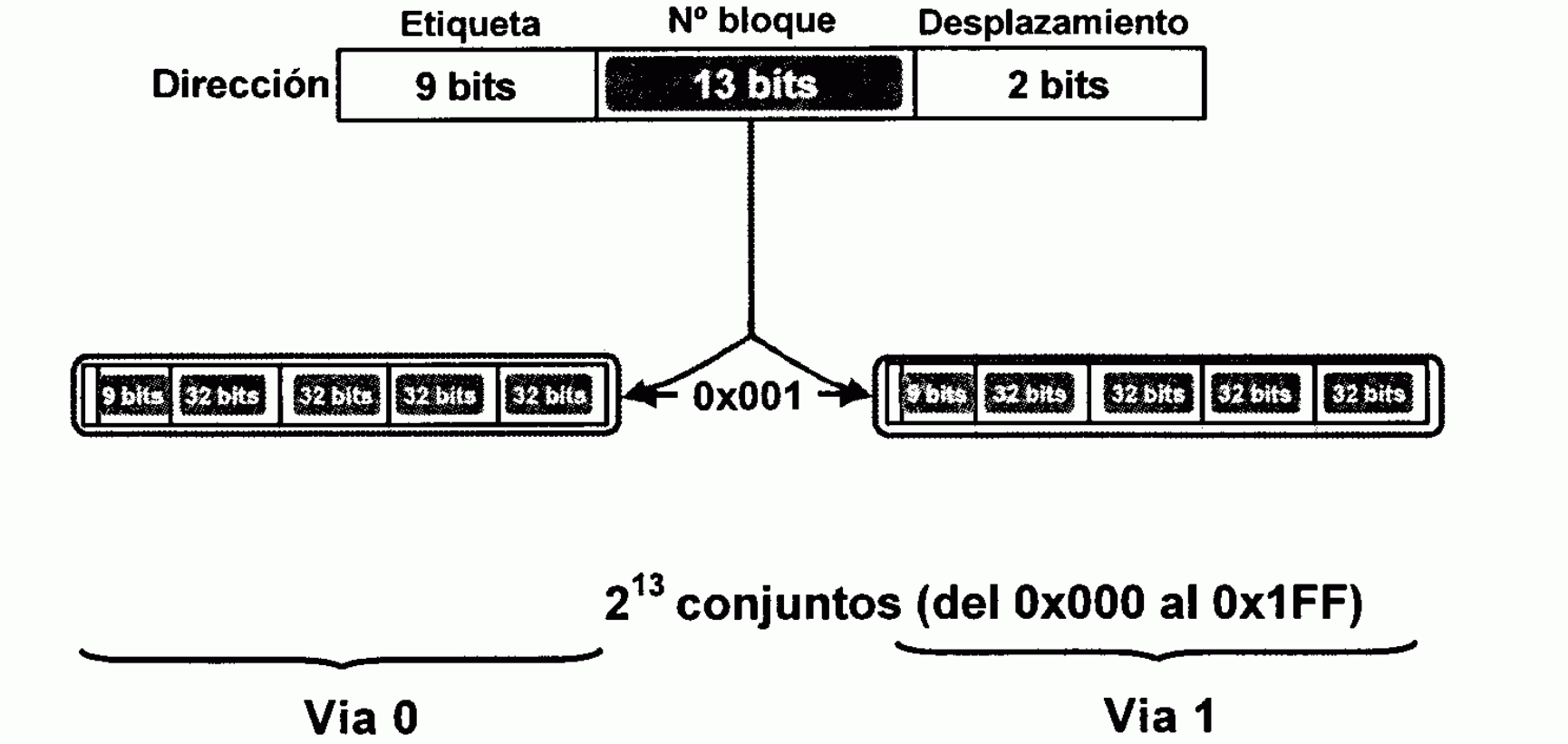
Cache asociativa por conjunto

Organización



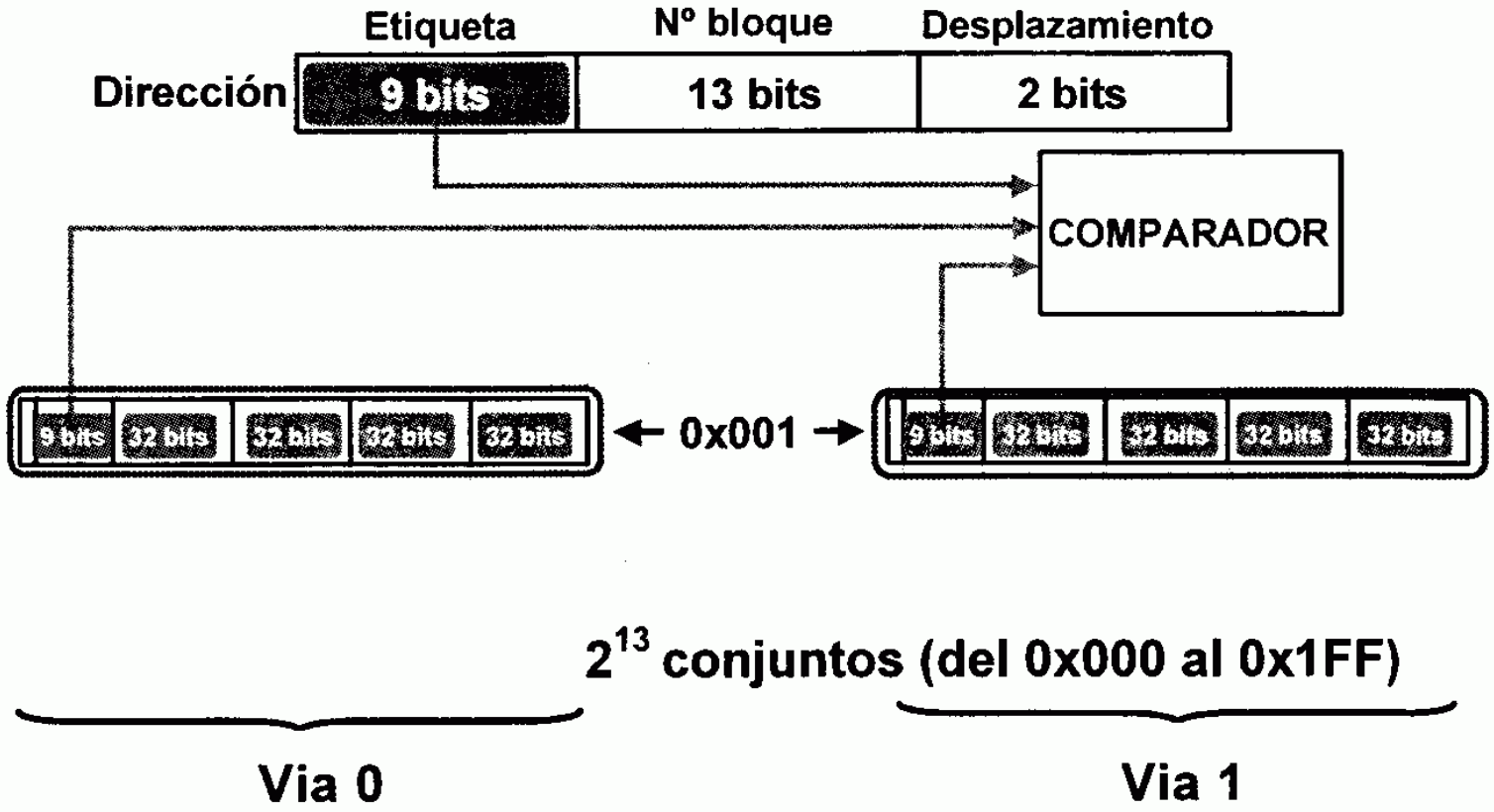
Cache asociativa por conjunto

✍ **Organización**



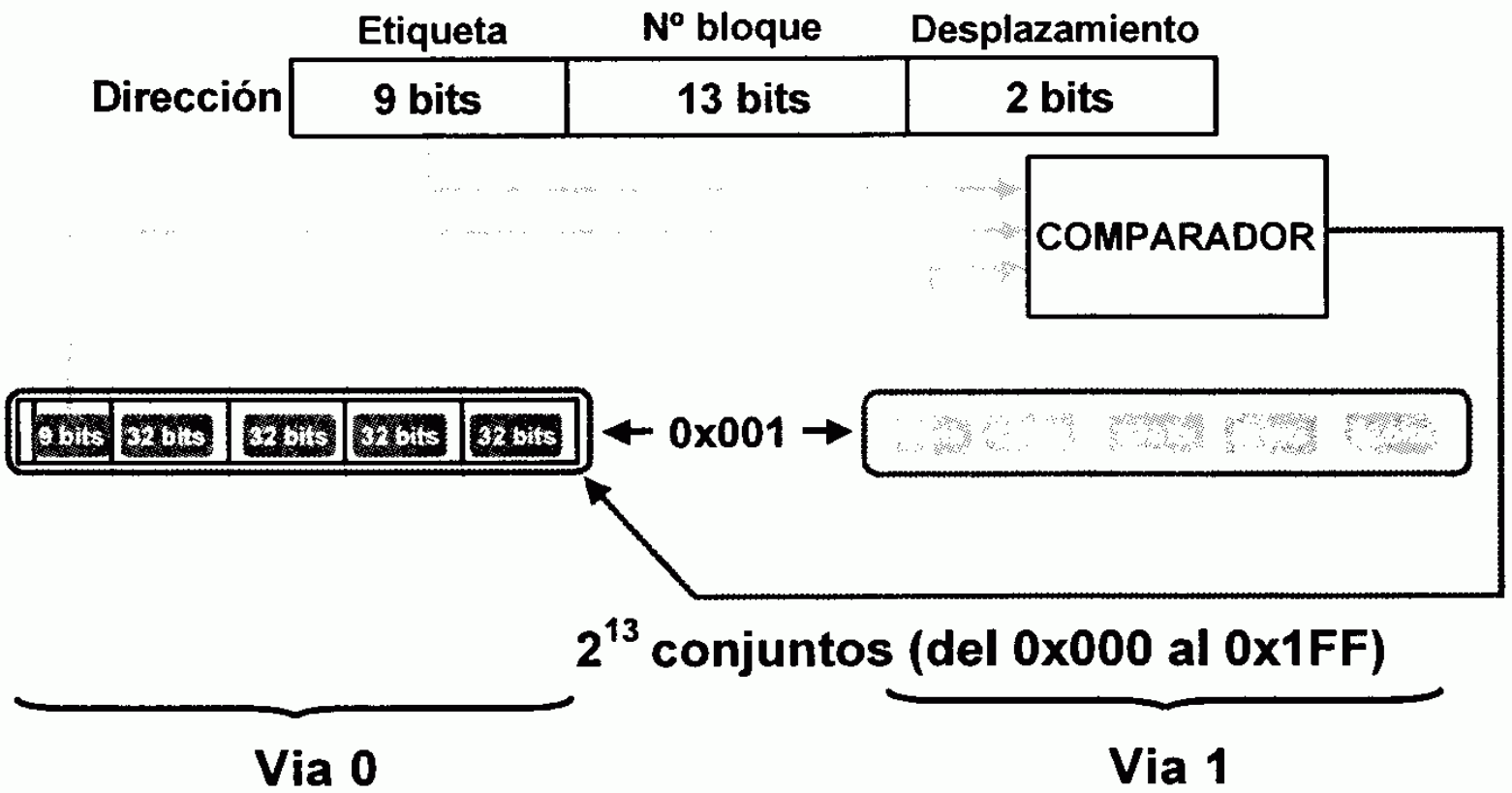
Cache asociativa por conjunto

Organización



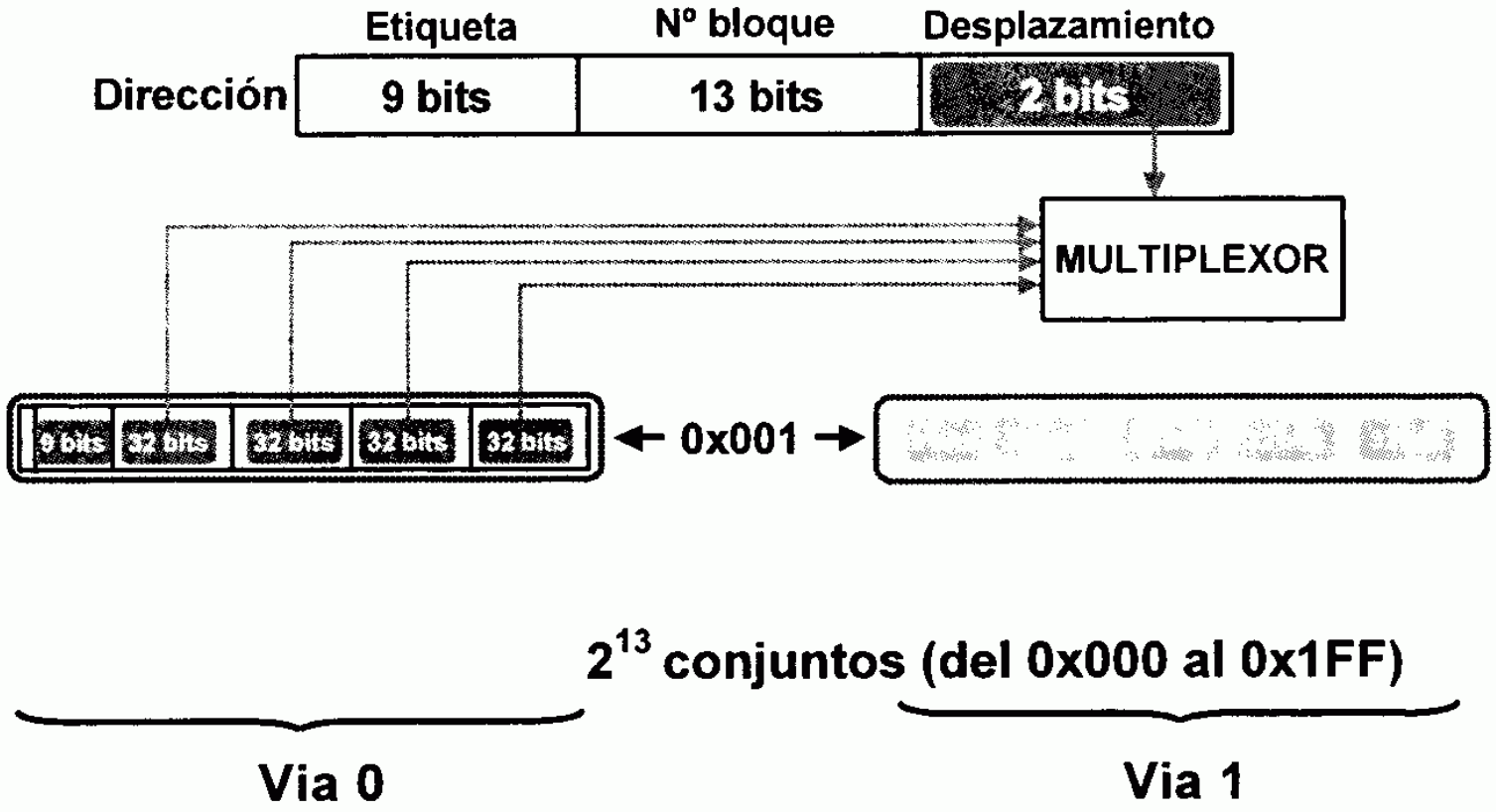
Cache asociativa por conjunto

Organización



Cache asociativa por conjunto

Organización



Funcionamiento de la memoria cache

- **Adquisición**

  ¿Cuándo se lleva un conjunto de MP a una línea de cache?

  - **En fallo**
    - ✓ **Se trae la línea sobre la que se quiera leer o escribir**
  - **En fallo de lectura (No-write allocate)**
    - ✓ **Se trae la línea sólo cuando se quiere leer**
  - **En fallo de escritura (write allocate)**
    - ✓ **Se trae la línea cuando se quiere escribir**

Funcionamiento de la memoria cache

- **Actualización**

  ¿Cuándo se actualiza el bloque de MP?

  - **Escritura directa (Write Through)**
    - ✓ **Simultáneamente se escribe en Cache y MP**
    - ✓ **Optimización de la escritura**
      - **Buffer de escritura**
        - **– Evita las paradas de escritura de la CPU**

  - **Post escritura (Write Back)**
    - ✓ **Se actualiza la MP cuando se reemplaza la línea de MC**
    - ✓ **Mecanismo para indicar inconsistencia entre Cache y MP**
      - **Bit extra para indicar línea modificada (M)**

Funcionamiento de la memoria cache

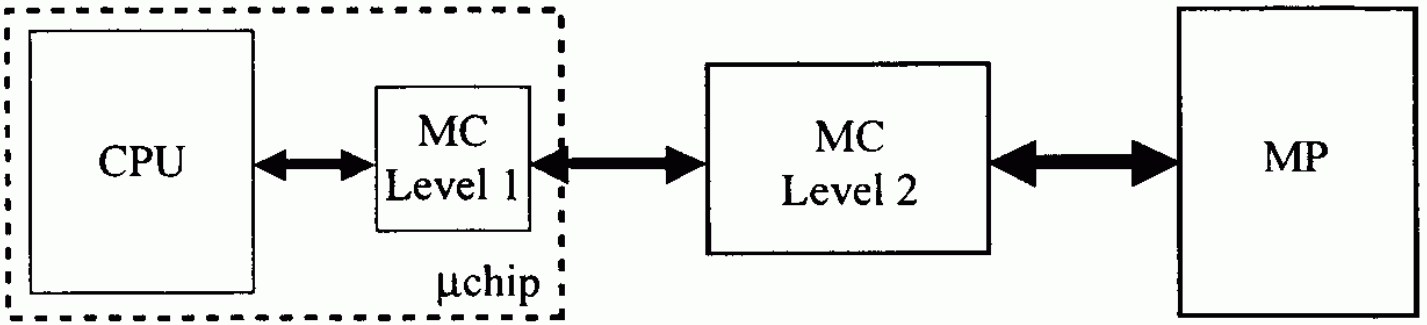
- **Reemplazo**

  ¿Qué línea se desecha cuando hace falta espacio para otra?

  - **Aleatoria (Random)**
    - ✓ **La línea a reemplazar se elige de forma aleatoria**
  - **Menos recientemente utilizada (LRU o pseudo-LRU)**
    - ✓ **Siguiendo el principio de localidad se reemplaza la línea que más tiempo tiene si ser referenciada**
  - **Menos frecuentemente utilizada (LFU)**
    - ✓ **Se reemplaza la línea que haya sido menos referenciada**
  - **FIFO (no se usa normalmente)**
    - ✓ **Se sigue un orden de reemplazo atendiendo al orden en que se han ido utilizando las líneas**

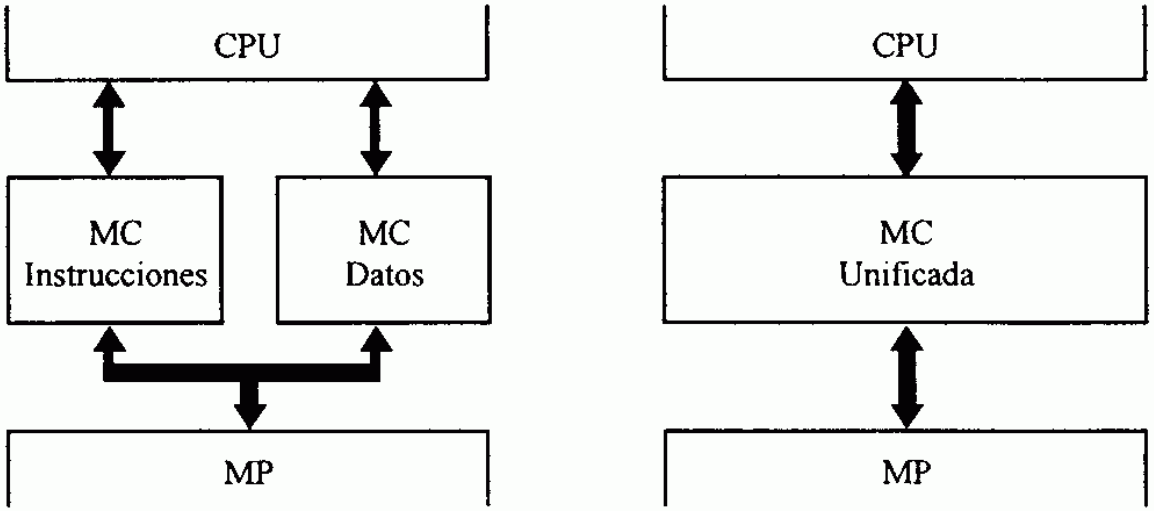
Memoria cache multinivel

- **Esquema básico**
  - La diferencia entre velocidad de CPU y MP crece cada día.
  - Se pueden crear sistemas de memoria con varios niveles de MC que tienen normalmente distintos tamaños y velocidades.
  - Los sistemas actuales vienen equipados con dos niveles de MC:
    - ✓ El primero integrado dentro del propio procesador con tamaño pequeño: Los accesos a la cache tienen un ciclo de reloj igual al de la CPU.
    - ✓ El segundo nivel tiene gran tamaño para conseguir que el número de accesos a MP sea el menor posible.



Memoria cache multinivel

- **Organización de los datos**
  - Las MC pueden almacenar tanto datos como instrucciones.
  - Existen diferentes tipos de MC:
    - ✓ Cache de datos: Sólo almacenan datos.
    - ✓ Cache de instrucciones: Sólo almacenan instrucciones.
    - ✓ Cache unificada o mixta: Simultánemente existen datos e instrucciones.



Rendimiento de la cache

- **Definiciones**
  - $T_{(i)}$: Tiempo de acceso en el nivel i
  - $T_{(i+1)}$: Tiempo de acceso en el nivel i+1
  - α es la tasa de acierto (hit ratio)
  - $T_{ap(i)}$: Tiempo de acceso aparente en el nivel i

$$T_{ap(i)} = \alpha\, T_{(i)} + (1-\alpha)\, T_{fallo(i)}$$

  - $T_{fallo(i)} = T_{penalización(i)} + T_{(i)}$
  - $T_{penalización(i)} = T_{ap(i+1)}$
  - $\theta = (1-\alpha)$ es la tasa de fallo (miss ratio)

$$T_{ap(i)} = T_{(i)} + \theta T_{penalización(i)}$$

Rendimiento de la cache

Definiciones

- A nivel del procesador

$$T_{CPU} = \left[\begin{array}{c}\text{Ciclos de}\\ \text{ejecución}\end{array} + \begin{array}{c}\text{Ciclos de}\\ \text{bloqueo}\end{array}\right] \times \text{Tiempo de ciclo}$$

- **Ciclos de ejecución = I x CPI**

$$\text{Ciclos de bloqueo} = I \times \left[\frac{\text{Referencias a memoria}}{\text{Instrucción}}\right] \times \left[\begin{array}{c}\text{tasa de}\\ \text{fallos}\end{array}\right] \times \left[\begin{array}{c}\text{Penalización}\\ \text{por fallo}\end{array}\right]$$

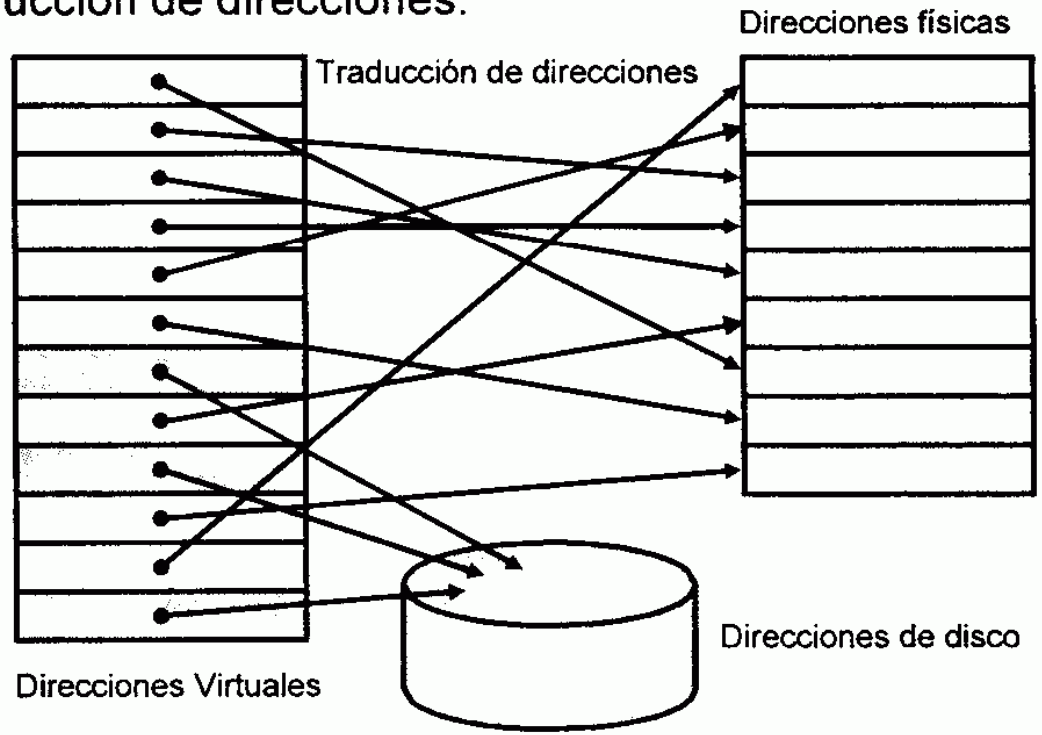
Memoria virtual

- **Conceptos**
  - Un computador emplea memoria virtual cuando el espacio de direcciones que utilizan los programas durante su ejecución es mayor que el espacio de direcciones físicas en MP.
  - El espacio de direcciones virtuales (instrucciones y datos) que maneja un programa se divide en bloques:
    - ✓ En un instante determinado en MP sólo se encuentran unos pocos bloques del programa.
    - ✓ El resto de bloques se mantienen en memoria secundaria (area swap del disco).
    - ✓ Se van trayendo nuevos bloques a la MP según se van necesitando.
  - Objetivos de la memoria virtual:
    - ✓ Permite disponer de un espacio de direcciones superior al real
    - ✓ Permite compartir eficientemente la memoria (entornos multiproceso)

Memoria virtual

Conceptos

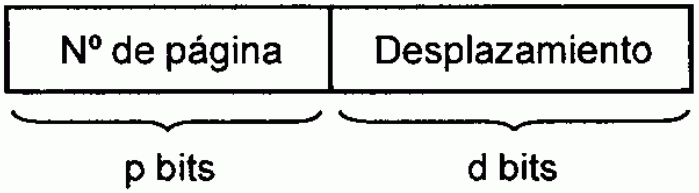
- La CPU produce direcciones virtuales que se traducen en una dirección física que se usa para acceder a MP.
- Según el tipo de traducción de direcciones:
  - ✓ Memoria virtual paginada
  - ✓ Memoria virtual segmentada
  - ✓ Memoria virtual seg/pag



Memoria virtual

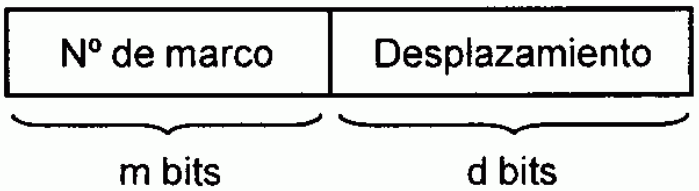
Memoria virtual paginada

- Espacio virtual de direcciones:
  - ✓ Se divide en páginas de tamaño fijo
  - ✓ Dirección virtual:



Tamaño de una página: 2^d palabras
Nº máximo de páginas (por proceso): 2^p páginas

- Espacio físico de direcciones:
  - ✓ Se divide en marcos de página del mismo tamaño que una página
  - ✓ Dirección física:

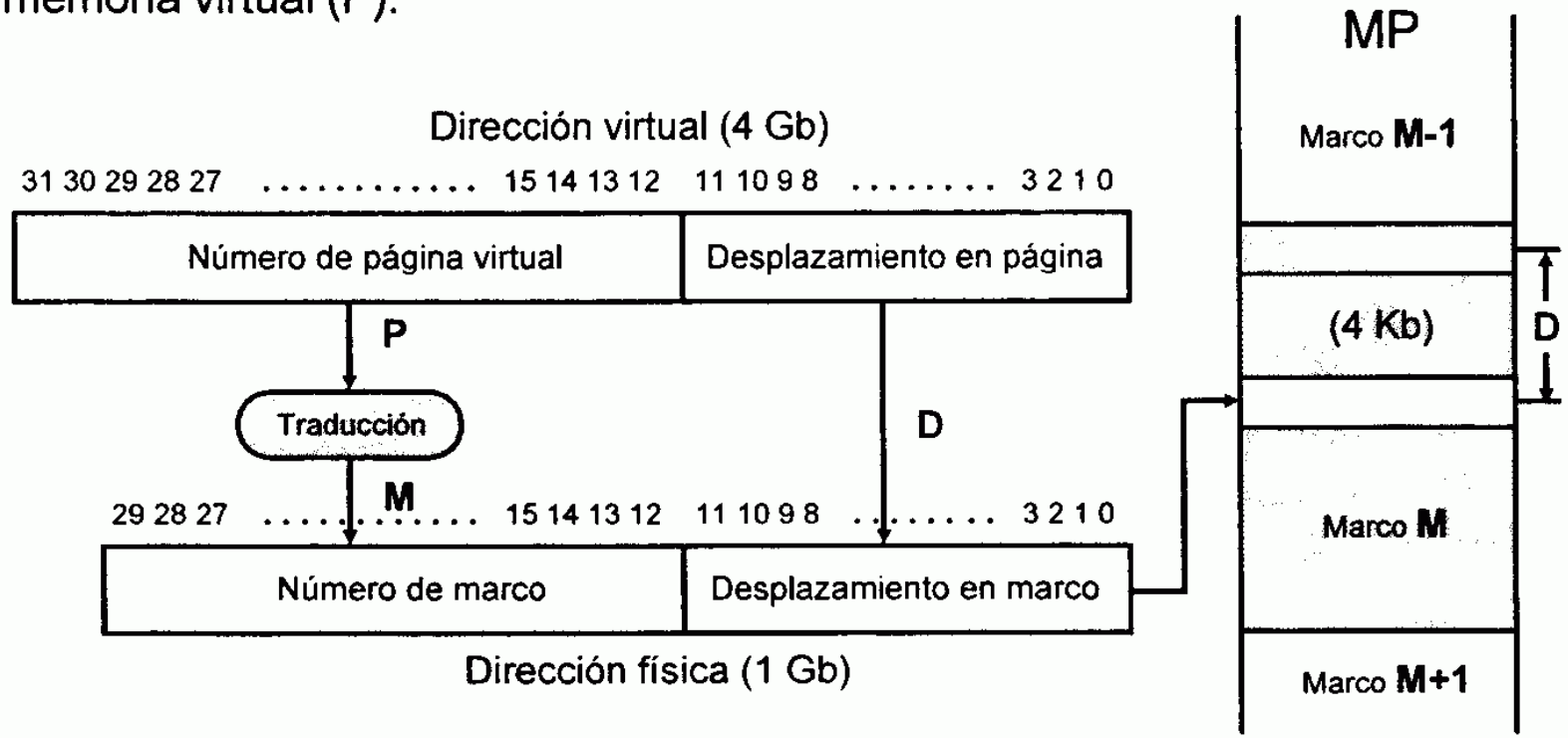


Tamaño de un marco: 2^d palabras
Nº total de marcos en MP: 2^m marcos

Memoria virtual

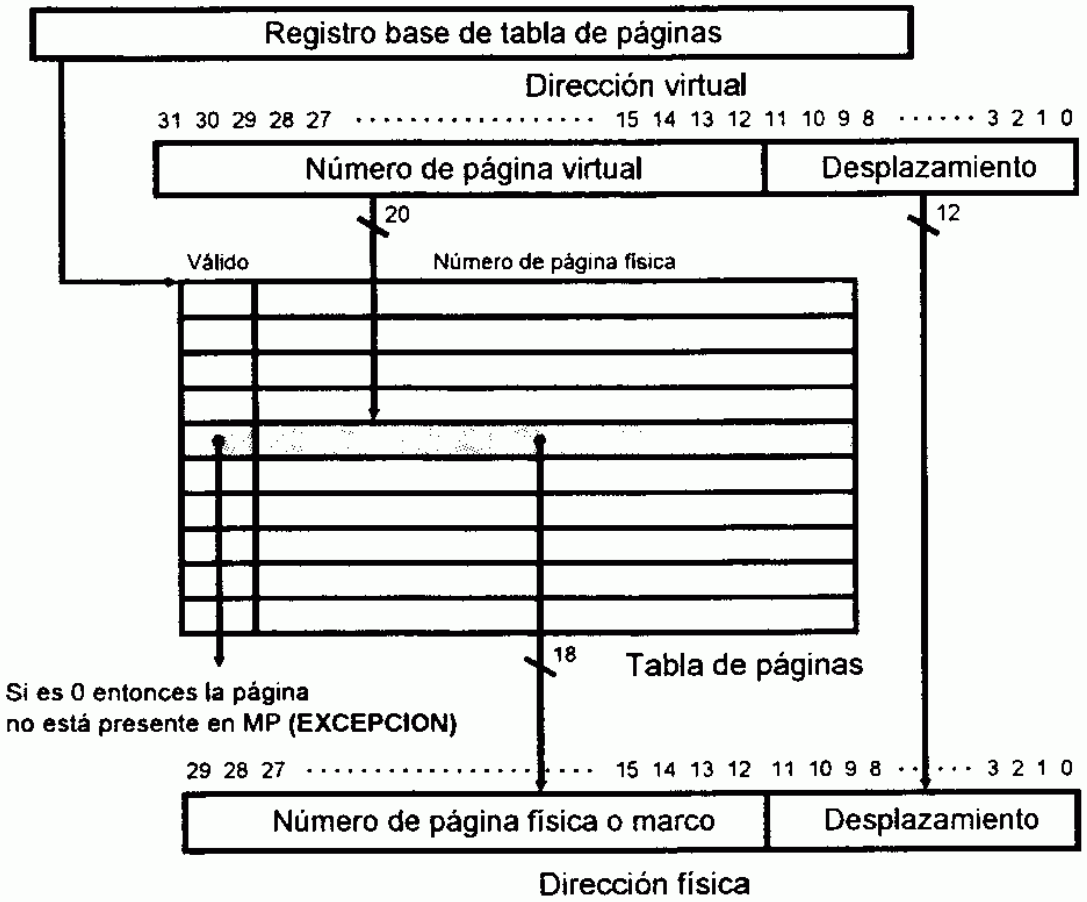
✍ Traducción de direcciones

Es necesario un mecanismo de correspondencia para conocer en qué marco de página (M) de MP está ubicada una determinada página de memoria virtual (P).



Memoria virtual

✍ Tabla de páginas

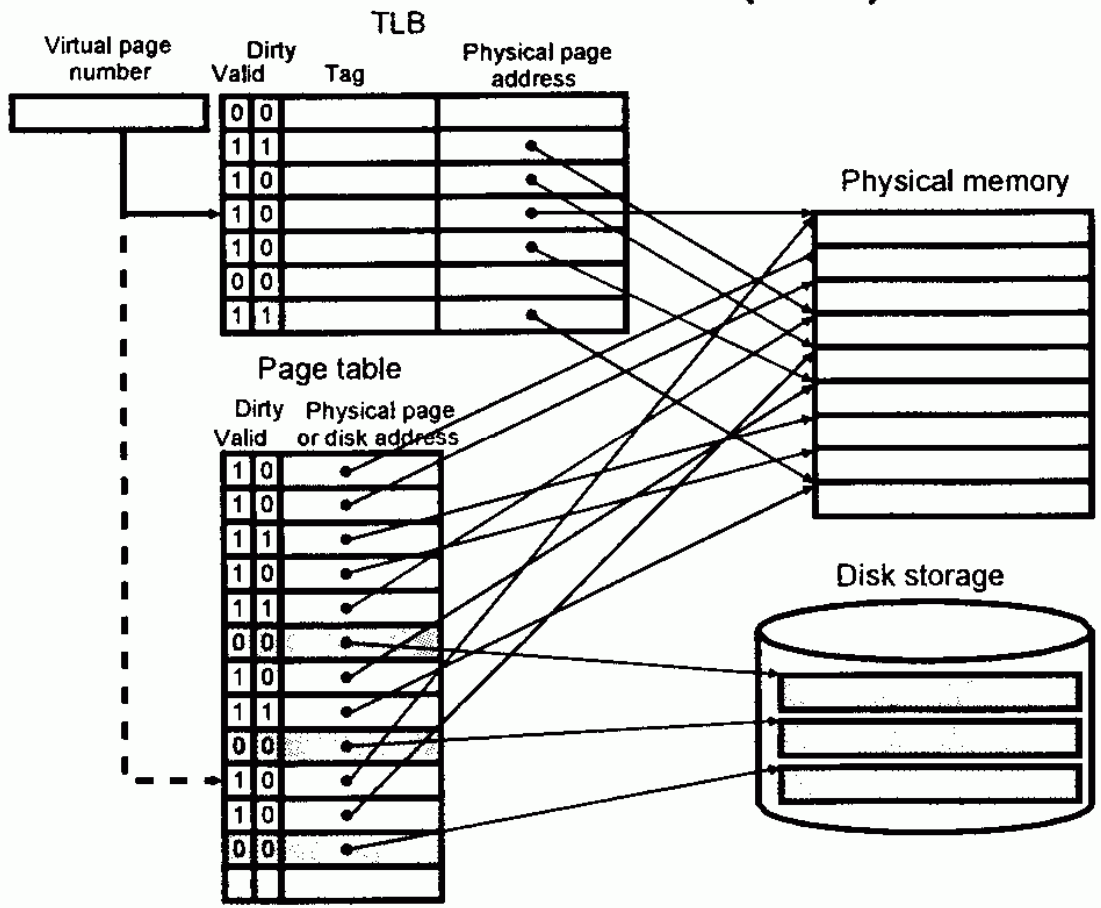


Memoria virtual

- **Tabla de páginas**
  - Existe una tabla de páginas (TP) por cada proceso J.
    - ✓ La TP contiene una entrada por cada posible página del proceso J: 2^p entradas por proceso.
    - ✓ La entrada P-ésima de TP contiene el marco de página M donde está ubicada la página P (si está en MP).
    - ✓ La TP de un determinado proceso J está apuntada por un registro base asignado a ese proceso.
  - Problemas al almacenar la TP en MP:
    - ✓ La TP puede de ser de gran tamaño (4 Mb en el ejemplo).
    - ✓ Se requieren dos accesos a MP por cada referencia: Se duplica el tiempo de acceso.
  - Uso de registro de limites, tablas de hash, varios niveles de tablas, paginar las tablas.
  - Máquinas actuales incorporan una cache especial que guarda las traducciones más recientes: TLB (*table lookaside buffer*).

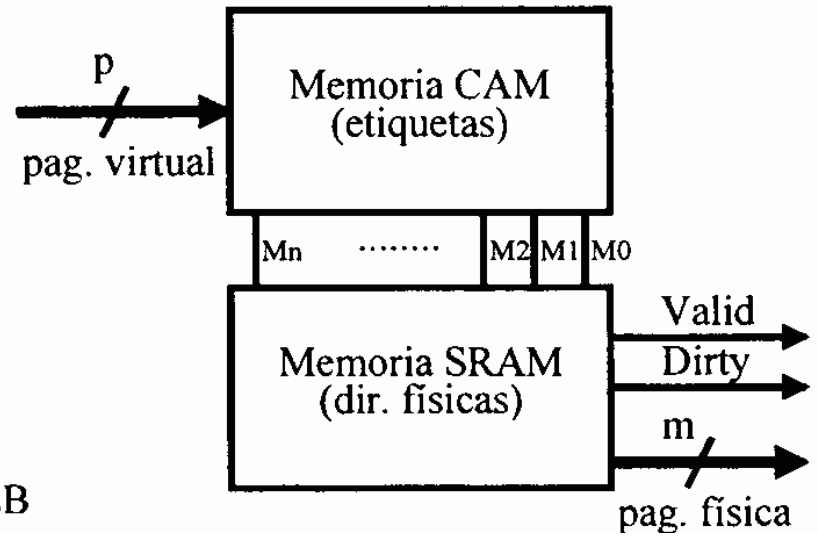
Memoria virtual

✍ Buffer de traducción de direcciones (TLB)



Memoria virtual

- **Buffer de traducción de direcciones (TLB)**
  - Gracias al principio de localidad, las traducciones de direcciones se vuelven a utilizar en breve.
  - El TLB es una memoria cache especial que permite mejorar las prestaciones, al almacenar las traducciones más recientes.
  - Cada entrada en el TLB consta de tres partes:
    - ✓ El número de página virtual.
    - ✓ El número de marco.
    - ✓ Varios bits de control.
  - Implementacion de un TLB:



Nota: n es el número de entradas del TLB

Memoria virtual

- **Funcionamiento**
  - Políticas de emplazamiento: Todas las páginas son de igual tamaño y coinciden en tamaño con los marcos de página.
    - ✓ Una página se puede ubicar en cualquier marco de página libre
  - Políticas de reemplazo de páginas: Cuando no hay ningún marco de página libre se debe reemplazar alguna de las páginas de MP.
    - ✓ Principales algoritmos: Aleatorio, FIFO, LFU, LRU
  - Políticas de actualización de la MS:
    - ✓ Necesario mantener la coherencia entre MP y MS:
      - Cuando se modifica una página en Mp es necesario actualizar esa misma página en MS.
    - ✓ Se emplea la política de actualización de post-escritura:
      - Una página modificada se actualiza en MS sólo cuando se reemplaza.