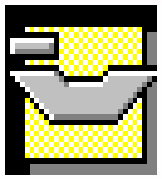


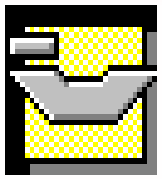
# ARQUITECTURAS DE COMPUTADORES

## 2º CURSO INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

## TEMA 4 - MEMORIA


## JOSÉ GARCÍA RODRÍGUEZ
## JOSÉ ANTONIO SERRA PÉREZ

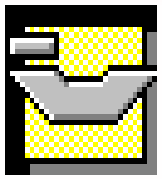

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE





# La Memoria

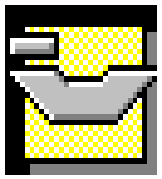
- Introducción
- Definiciones y conceptos
- Memoria principal semiconductora
- Memoria cache
- Memoria asociativa
- Memoria compartida





# Introducción

- La memoria contiene los programas que se ejecutan en el computador y los datos sobre los que trabajan dichos programas.
- La memoria es un elemento sencillo, sin embargo, presenta una gran diversidad de tipos, tecnologías, estructuras, prestaciones y costes.
- Un computador dispone de una jerarquía de elementos de memoria donde algunos están localizados internamente al propio computador y otros localizados externamente.



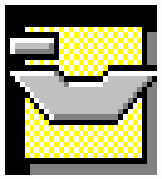
Definiciones
y
conceptos

# Requisitos de las memorias

Un sistema de memoria debe disponer de los siguientes elementos:

- **Medio o soporte**. Deberá disponer de un elemento donde se almacenen estados diferentes que codifiquen la información.
- **Transductor**. Es un elemento que permite convertir una energía en otra, es decir, transformar magnitudes físicas a eléctricas (sensor) o magnitudes eléctricas a físicas (actuador). Memoria estática y memoria dinámica.
- **Mecanismo de Direccionamiento**. Deberá disponer de un procedimiento para leer y escribir información en el lugar y tiempo deseado.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

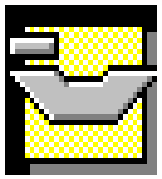
# Características de las memorias

## Localización

Dependiendo de donde esté ubicada físicamente la memoria se distinguen tres tipos:

- Memoria interna al procesador. Memoria de alta velocidad utilizada de forma temporal.
- Memoria interna (Memoria Principal).
- Memoria externa (Memoria Secundaria).

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

# Características de las memorias

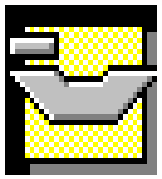
## Capacidad

Cantidad de información que puede almacenar el sistema de memoria.

La capacidad de la memoria se mide en múltiplos de unidades de bit.

| | |
|---|---|
| 1 bit | 1 Mb = 1024 Kb = $2^{20}$ bits |
| 1 nibble = 4 bits | 1 Gb = 1024 Mb = $2^{30}$ bits |
| 1 byte = 1 octeto = 8 bits | 1 Tb = 1024 Gb = $2^{40}$ bits |
| 1 Kb = 1024 bits = $2^{10}$ bits | |

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

# Características de las memorias

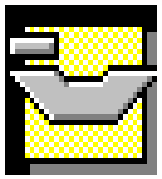
## Unidad de transferencia

Es igual al número de líneas de datos de entrada y salida del módulo de memoria.

Conceptos asociados:

- Palabra. El tamaño de la palabra es generalmente igual al número de bits utilizados para representar un número entero y la longitud de una instrucción.
- Unidad direccionable. Es el tamaño mínimo que podemos direccionar la memoria.
- Unidad de transferencia. Para la memoria principal es el número de bits que se leen o escriben en memoria a la vez.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

# Características de las memorias

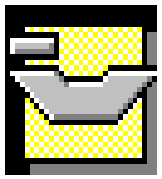
## Método de acceso

Forma de localizar la información en memoria.

Tipos:

- Acceso secuencial (SAM: Sequential Access Memory).
- Acceso directo (DAM: Direct Access Memory).
- Acceso aleatorio (RAM: Random Access Memory).
- Acceso asociativo (CAM: Content Addressable Memory).

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

# Características de las memorias

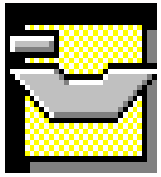
## Velocidad

Para medir el rendimiento se utilizan tres parámetros:

- Tiempo de acceso ($T_A$)

  RAM: tiempo que transcurre desde el instante en el que se presenta una dirección a la memoria hasta que el dato, o ha sido memorizado, o está disponible para su uso.

  Otra: tiempo que se emplea en situar el mecanismo de lectura/escritura en la posición deseada.

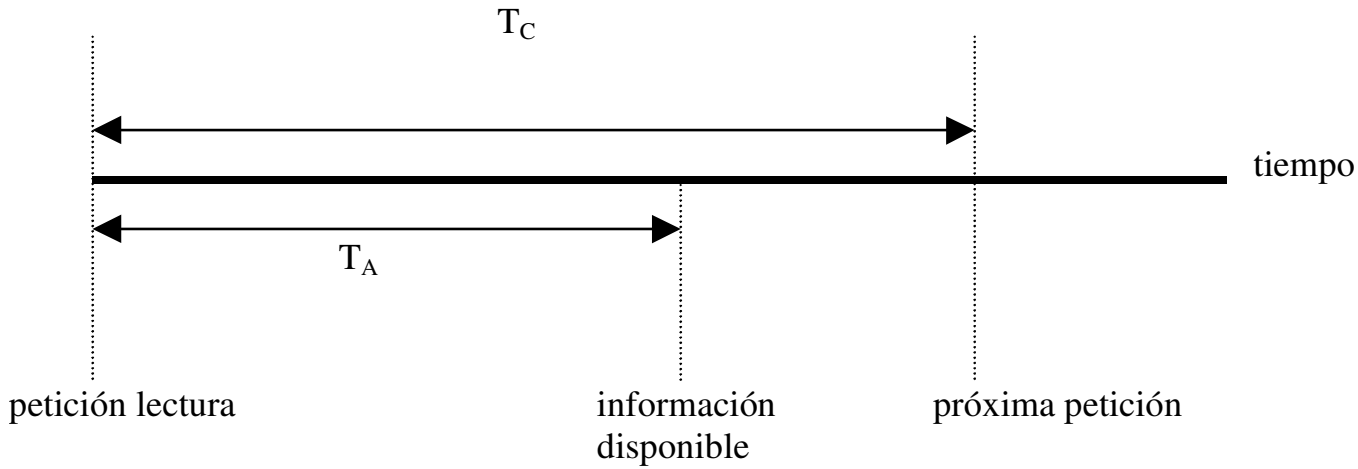


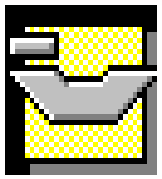
Definiciones
y
conceptos

# Características de las memorias

- Tiempo de ciclo de memoria ($T_C$)

  Tiempo que transcurre desde que se da la orden de una operación de lectura/escritura hasta que se pueda dar otra orden de lectura/escritura.





Definiciones
y
conceptos

# Características de las memorias

- Velocidad de transferencia ($V_T$).

  Es la velocidad a la que se pueden transferir datos a, o desde, una unidad de memoria.

  En el caso de acceso aleatorio $$V_T = \frac{1}{T_C}$$

  En el caso de acceso no aleatorio $$T_N = T_A + \frac{N}{V_T}$$

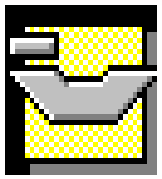
  $T_N$ Tiempo medio de lectura/escritura de N bits

  $T_A$ Tiempo de acceso

  N Número de bits

  $V_T$ Velocidad de transferencia (bits/segundo)

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

# Características de las memorias

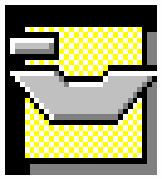
- **Dispositivo físico**

  Los sistemas de memorias empleados en los computadores utilizan diferentes dispositivos físicos.

  Los tipos más usados son:

- Para la memoria principal se utilizan memorias semiconductoras
- Como memoria secundaria se emplean:
  - Memorias magnéticas, discos, cintas, etc.
  - Memorias ópticas, utilizadas.
  - Memorias magneto-opticas.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

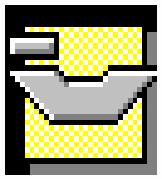
# Características de las memorias

## Aspectos físicos

Las principales características físicas a tener en cuenta para trabajar con determinados tipos de memorias son:

- Alterabilidad. Esta propiedad hace referencia a la posibilidad de alterar el contenido de una memoria. Memorias ROM y RWM.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE

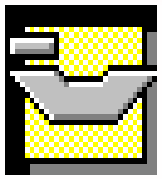


Definiciones
y
conceptos

# Características de las memorias

- Permanencia de la información. Relacionado con la duración de la información almacenada en memoria:
  - Lectura destructiva. Memorias de lectura destructiva (DRO: Destructive ReadOut) y memorias de lectura no destructiva NDRO (No Destructive ReadOut).
  - Volatilidad. Esta característica hace referencia a la posible destrucción de la información almacenada en un cierto dispositivo de memoria cuando se produce un corte en el suministro eléctrico. Memorias volátiles y no volátiles.
  - Almacenamiento estático/dinámico. Una memoria es estática si la información que contiene no varía con el tiempo. Una memoria es dinámica si la información almacenada se va perdiendo con forme transcurre el tiempo. Para que no se pierda el contenido habrá que recargar o refrescar la información. Memoria SRAM (Static RAM) y Memoria DRAM (Dynamic RAM).

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

# Características de las memorias

## Organización

Hace referencia a la disposición física de los bits para formar palabras.

La organización depende del tipo de memoria que se trate.

Para una memoria semiconductora distinguimos tres tipos de organización:

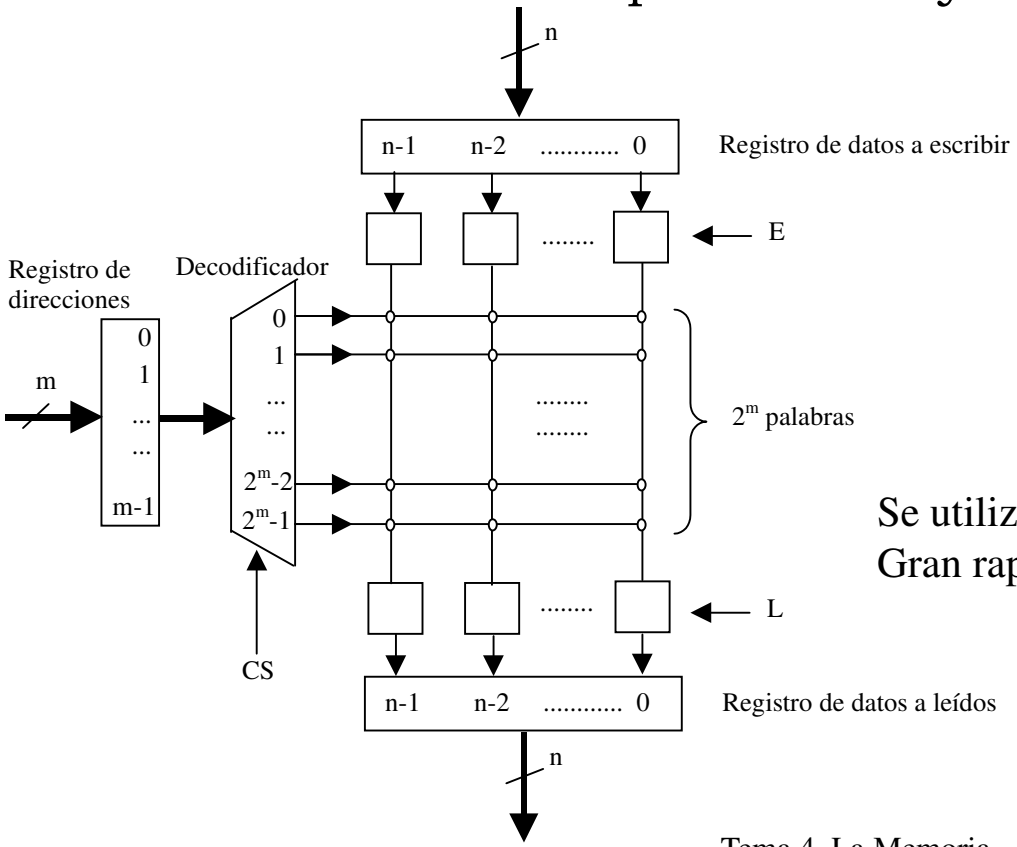
- Organización 2D
- Organización 2½D
- Organización 3D

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE

# Características de las memorias

- Organización 2D

  RAM de $2^m$ palabras de n bits cada una, la matriz de celdas está formada por $2^m$ filas y n columnas.

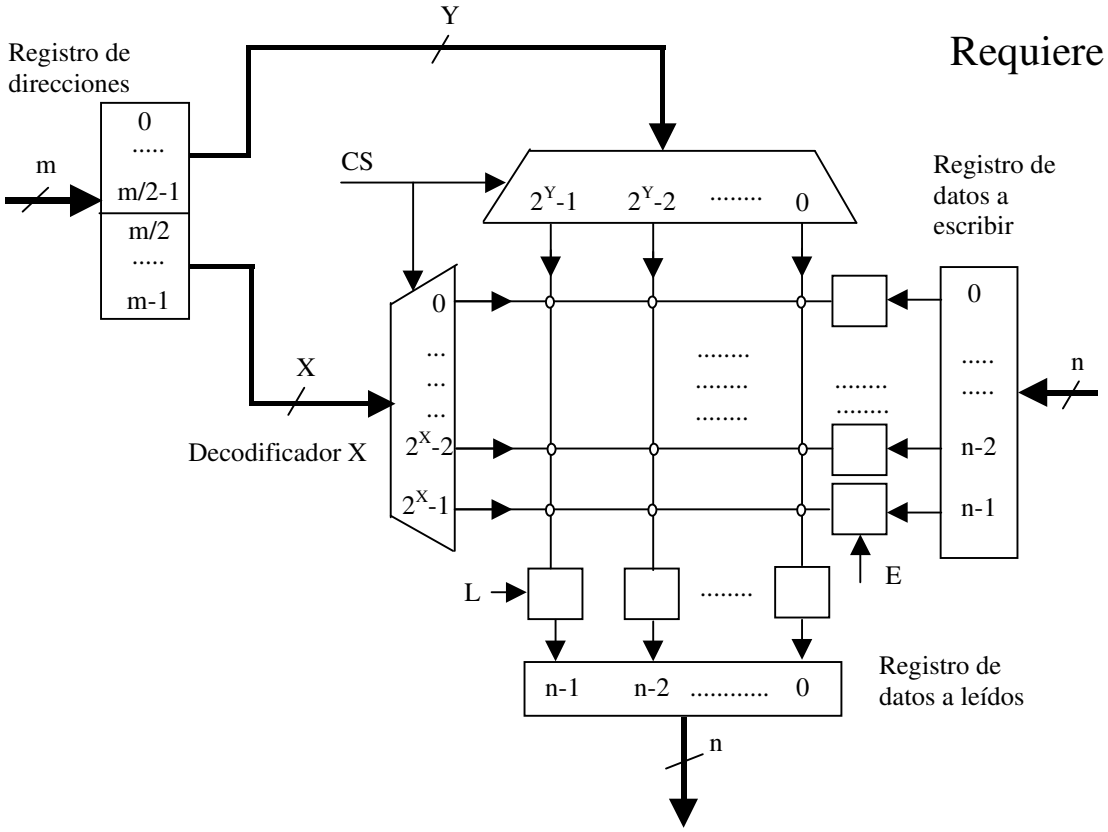



Se utiliza en memorias de capacidad reducida.
Gran rapidez de acceso.

# Características de las memorias

- Organización 2½D

  Utiliza dos decodificadores con m/2 entradas y $2^{m/2}$ salidas.

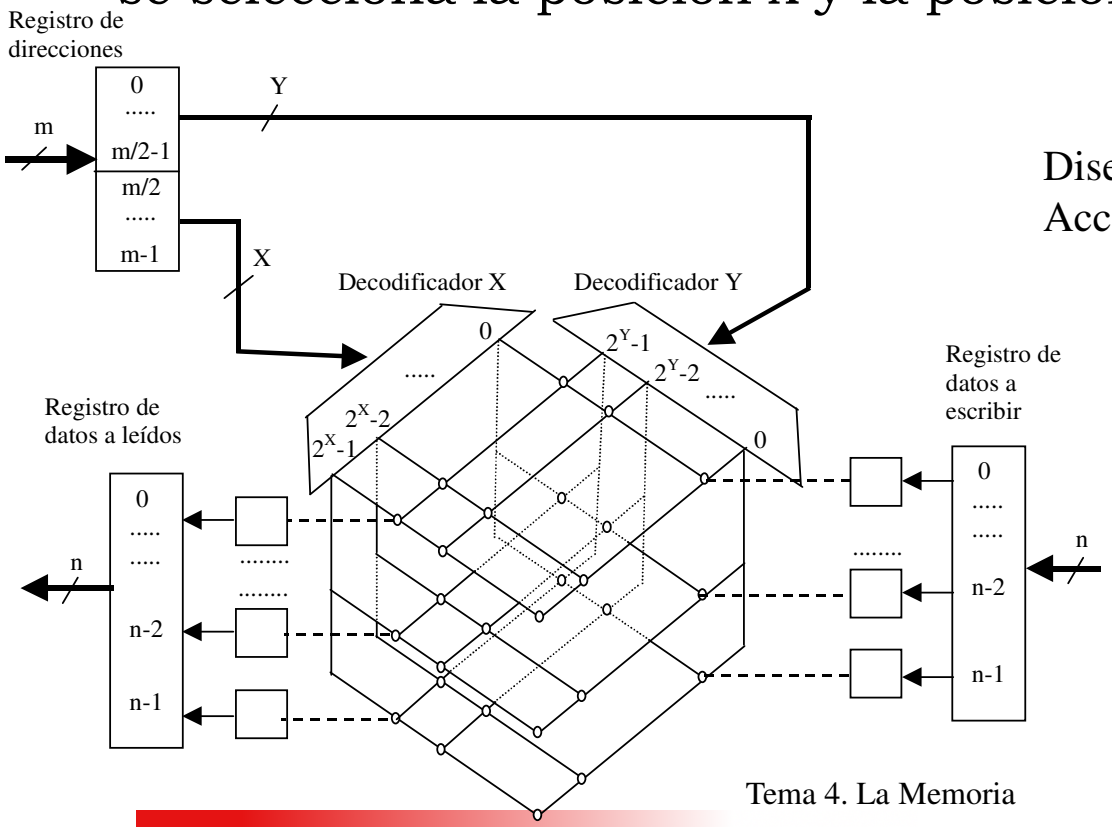



Requiere menos puertas lógicas.

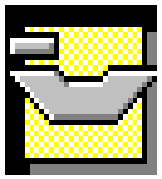
# Características de las memorias

- Organización 3D

  Es similar a la organización 2½D pero la palabra de n bits se almacena en n planos y dentro de cada plano se selecciona la posición x y la posición y.




Diseño más complicado
Acceso más lento



Definiciones
y
conceptos

# Jerarquía de las memorias

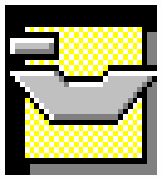
Parámetros fundamentales que caracterizan los tipos de memorias del computador:

- Coste.
- Velocidad. La memoria no debería provocar estados de espera al procesador.
- Capacidad.

La configuración ideal: memoria rápida, gran capacidad y poco coste.

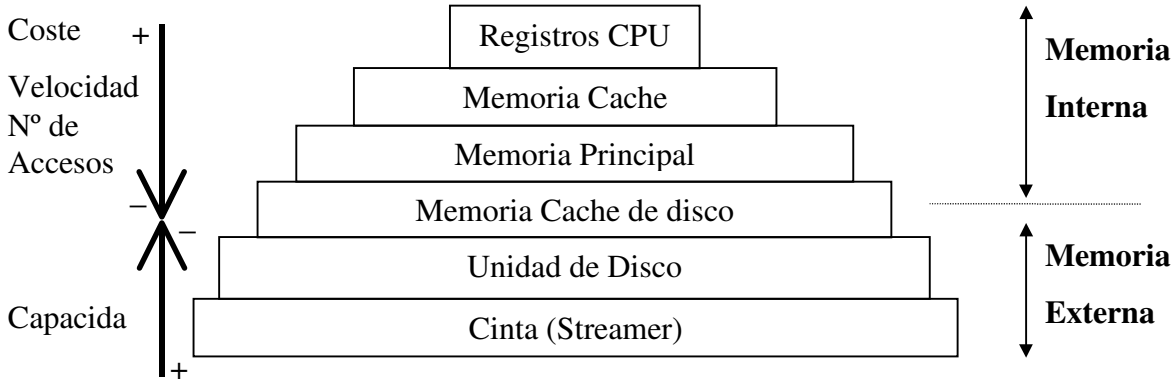
No hay que utilizar un solo tipo de memoria, sino emplear diferentes tipos de memoria, es decir, utilizar una **jerarquía de memoria**.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Definiciones
y
conceptos

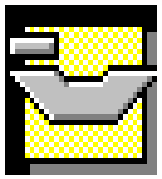
# Jerarquía de las memorias




Si bajamos hacia los niveles inferiores de la jerarquía ocurre que:

- El coste por unidad de información (bit) disminuye.
- La capacidad aumenta.
- El tiempo de acceso aumenta.
- La frecuencia de accesos a la memoria por parte de la CPU disminuye.

El **principio de localidad de referencia** depende de la frecuencia de accesos.

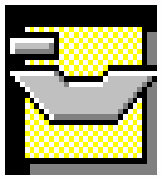


Memoria principal semiconductora

# Tipos de memorias

Dependiendo del tipo de operación que se puede realizar en una memoria distinguimos los siguientes tipos:

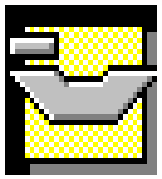
- Memorias de sólo lectura
  - **ROM (Read Only Memory).** Se suelen utilizar en microprogramación de sistemas, en subrutinas de bibliotecas de uso frecuente, etc. Los fabricantes suele emplearla cuando producen componentes de forma masiva.
  - **PROM (Programable Read Only Memory).** El proceso de escritura se lleva a cabo eléctricamente y puede realizarlo el suministrador o el cliente con posterioridad a la fabricación del chip original, a diferencia de la ROM que se graba cuando se fabrica. La memoria PROM permite una sola grabación y es más cara que la ROM.





# Tipos de memorias

- Memorias de sobre todo lectura (Read-Mostly Memory)
  - **EPROM (Erasable Programable Read Only Memory).** Mediante corriente eléctrica permite su escritura varias veces. Sin embargo, mediante rayos ultravioleta se elimina todo su contenido. Este tipo de memoria es más cara que la memoria PROM.
  - **EEPROM (Electricaly Erasable Programable Read Only Memory).** Se borra mediante corriente eléctrica de forma selectiva a nivel de byte. Es más cara que la memoria EPROM.
  - **Memoria Flash**. Denominada así por la velocidad con la que puede reprogramarse. Utiliza borrado eléctrico selectivo a nivel de bloque de bytes. Son más baratas que las memorias EEPROM.



Memoria principal semiconductora

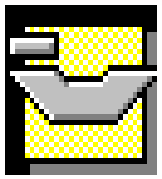
# Tipos de memorias

- Memorias de Lectura/Escritura

  **RAM (Random Access Memory).** Al igual que las anteriores, es de acceso aleatorio. Los principales tipo de memorias RAM son:

  - RAM dinámica (DRAM). Los datos se almacenan de forma similar a la carga de un condensador. Debido a que tiende a descargarse es necesario refrescarlas periódicamente. Son más simples y más baratas que las memorias SRAM.
  - RAM estática (SRAM). Los datos se almacenan formando biestables, por lo que no necesita refresco. Son más rápidas que las memorias DRAM y más caras.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Memoria principal semiconductora

# Tipos de memorias

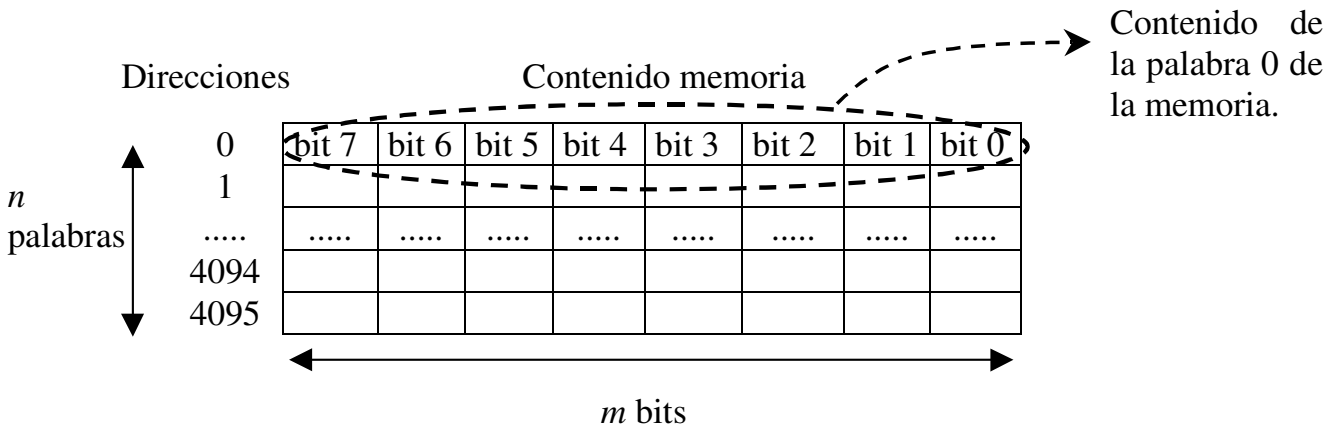
## Tabla resumen

| Tipo | Clase | Borrado | Escritura | Volatilidad |
|---|---|---|---|---|
| RAM | Lectura/Escritura | Eléctricamente por bytes | Eléctricamente | Volátil |
| ROM | Sólo lectura | No | Mediante máscaras | No Volátil |
| PROM | | | Eléctricamente | |
| EPROM | Sobre todo lectura | Luz violeta, chip completo | | |
| FLASH | | Eléctricamente por bloques | | |
| EEPROM | | Eléctricamente por byte | | |

# Diseño

## Chip de memoria

Se organiza internamente como una matriz de celdas de memoria de nxm, donde *n* es el número de palabras que puede almacenar el chip de memoria y *m* es el número de bits por palabra.

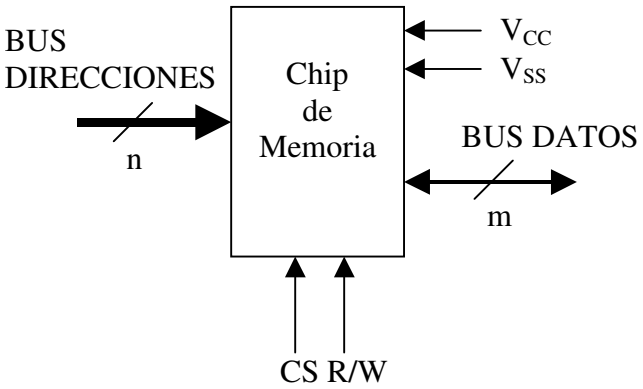


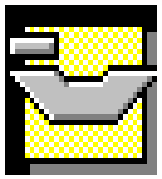

Memoria de 4Kx8

# Diseño

La interconexión de un chip de memoria se realiza a través de sus patillas:

- n patillas para el bus de direcciones, donde se podrá direccionar $2^n$ palabras.
- m patillas para el bus de datos indicando que en cada acceso se trabajará con m bits.
- W/R (Write/Read). Esta patilla indica el tipo de operación a realizar: lectura o escritura. También existen chips que disponen de una patilla para escritura WE (Write Enable) y otra para lectura OE (Output Enable).
- CS (Chip Selection) o CE (Chip Enable). Selecciona el chip de memoria al cual hay que acceder.
- $V_{CC}$. Alimentación del chip.
- $V_{SS}$. Conexión a tierra.

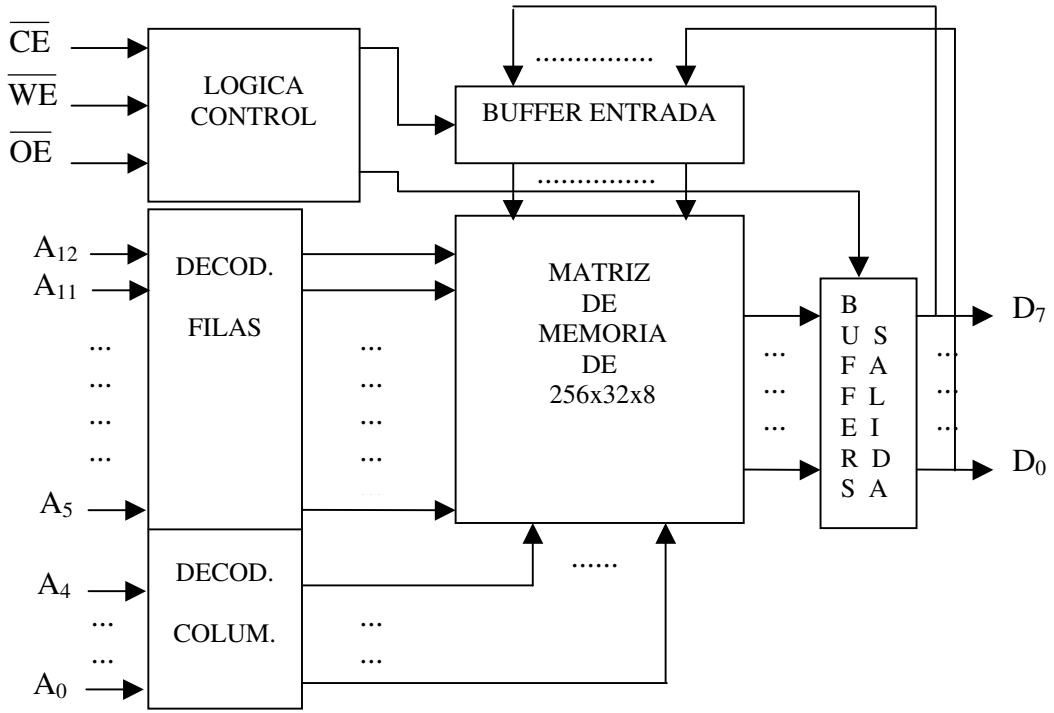


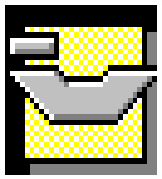


Memoria principal semiconductora

# Diseño

Para el correcto funcionamiento de la memoria es necesario incorporar una circuitería adicional como son decodificadores, multiplexores, buffers, etc.



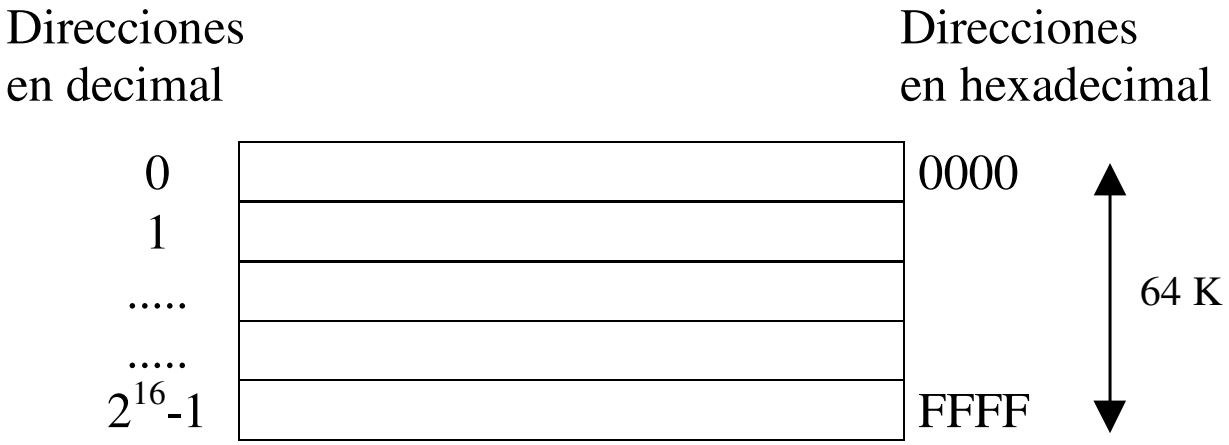


Memoria principal semiconductora

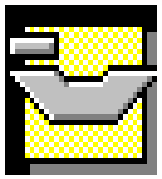
# Diseño

## Mapa de memoria

Espacio que puede direccionar un computador.




Ejemplo de un computador con bus de 16 bits.





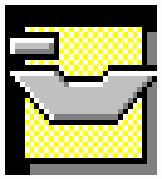
# Diseño

La implementación fisica del mapa de memoria se realiza utilizando uno o varios chips de memoria.

En el mercado se encuentran diferentes configuraciones de chips de memoria:

zKx1, zKx4, zKx8, zKx16, zKx32, zMx1, zMx4, zMx8, zMx16, zMx32, etc. donde z es un múltiplo de 2.

Así, por ejemplo, un chip de 1Kx8 indica que puede almacenar 1024 palabras de 8 bits cada una.



Memoria principal semiconductora

# Diseño

**Ejemplo 1:**

Si quisiéramos diseñar una memoria principal de 128 Kpalabras.

(1) ¿Cuántos chips de memoria de 32Kx8 necesitaremos si suponemos que la palabra es de 8 bits?.

(2) ¿Cuántos chips de memoria de 64Kx4 necesitaremos si suponemos que la palabra es de 8 bits?.

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE

# Diseño

**Solución 1:**

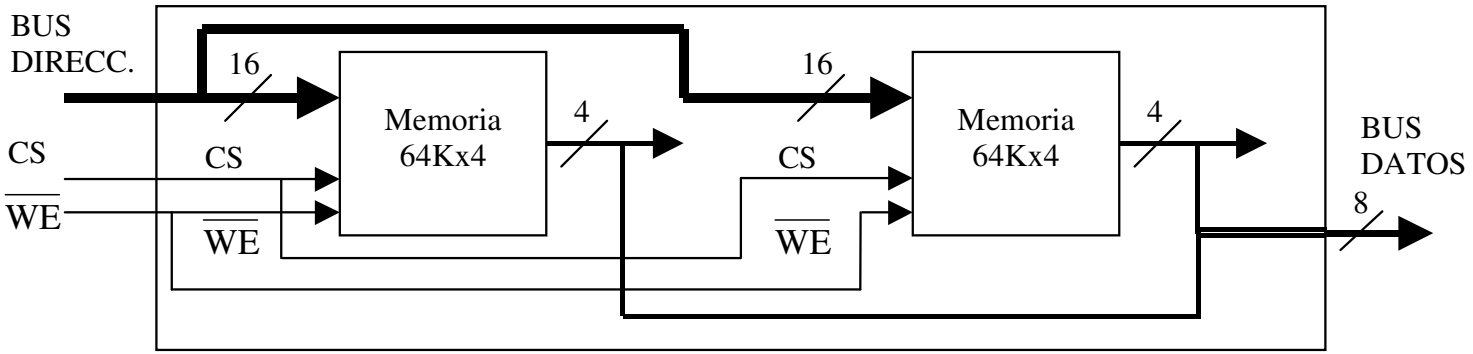
(1) Necesitamos direccionar 128K a partir de 32K luego necesitaremos 4 chips. Como el tamaño de la palabra es igual al contenido de cada dirección del chip no necesitaremos más.

(2) Para poder direccionar las 128K necesitaremos 2 chips. Con esos dos chips tenemos una memoria de 128Kx4 por lo que necesitaremos además otros 2 chips más para conseguir una memoria de 128Kx8.

# Diseño

Si quisiéramos diseñar una memoria de n bits y dispusiéramos de chips de t bits necesitaremos n/t chips en paralelo para alcanzar el ancho de palabra deseado.

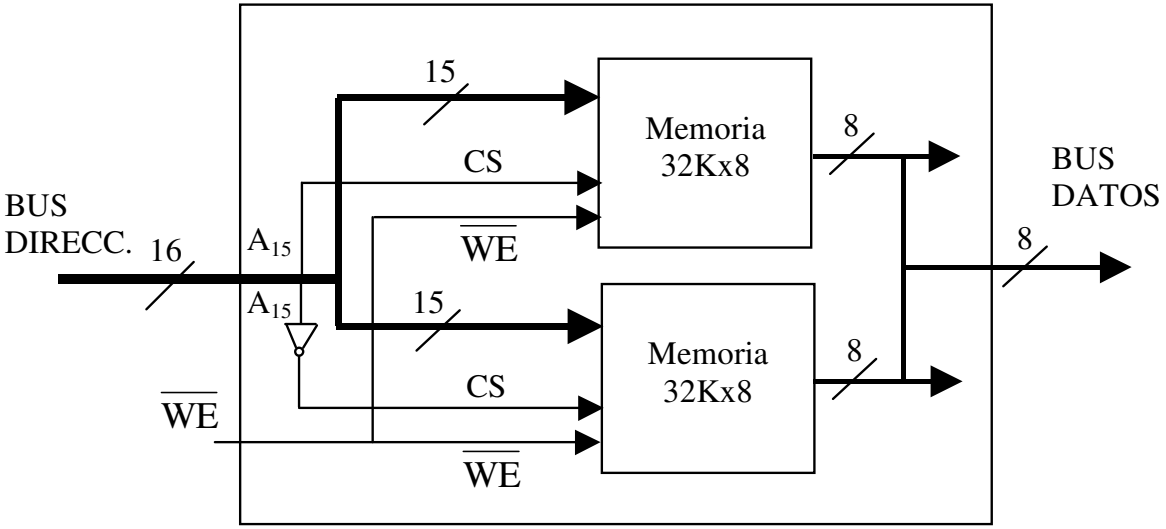
Ejemplo: Supongamos que queremos diseñar una memoria de 64 Kbytes (n=8) y disponemos de chips de 64Kx4 (t=4), entonces necesitaremos 2 chips (8/4). Además, podemos ver que hay 1 fila y 2 columnas de chips.

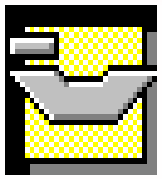


# Diseño

Si quisiéramos una capacidad de cK palabras y disponemos de chips de zK, necesitaremos c/z chips para conseguir la capacidad deseada

Ejemplo: Queremos diseñar una memoria con 64Kbytes y disponemos de chips de 32Kx8, entonces necesitaremos 2 chips. Cuando la línea $A_{15}$ está a 1 habilita el chip superior, mientras que cuando está a 0 habilita el chip inferior. En esta interconexión vemos que hay 2 filas y 1 columna de chips.







# Diseño

**Ejemplo 2:**

Vamos a obtener el mapa de memoria y el diagrama de conexiones de la memoria de un computador de 16 bits que permite direccionar 1Mpalabra y tiene 128Kpalabras instaladas a partir de chips de 64Kx1.

(1) Debemos obtener el número de bits del bus de direcciones.

(2) Averiguar el número de bits que se necesitan para direccionar el chip de memoria que vamos a emplear.

(3) Calcular el número de chips que necesitamos.

(4) Obtener el número de bits del bus de direcciones que permita seleccionar los chips de memoria.

(5) Dibujar el diagrama de conexiones de la memoria junto con la lógica de selección.

# Diseño

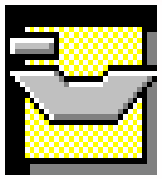
**Solución 2:**

(1) Debemos obtener el número de bits del bus de direcciones.

Como nos indican que puede direccionar 1Mpalabra, vemos que el bus es de 20 bits ($1M = 2^{20}$).

(2) Averiguar el número de bits que se necesitan para direccionar el chip de memoria que vamos a emplear.

Al ser el chip de memoria de 64K, necesitaremos 16 bits ($64K=2^{16}$). Los bits que emplearemos para direccionar el chip de memoria son los de menor peso, luego en este caso, $A_{15}A_{14}...A_1A_0$.





# Diseño

**Solución 2:**

(3) Calcular el número de chips que necesitamos.

Como queremos 128Kx16 necesitaremos 16 chips para obtener una palabra al completo (16 bits). Con estos primeros 16 bits tenemos 64Kx16, por lo que nos falta otros 64Kx16 más, es decir, 16 chips más. Por tanto, necesitaremos 32 chips de 64Kx1 para almacenar 128Kx16.

# Diseño

**Solución 2:**

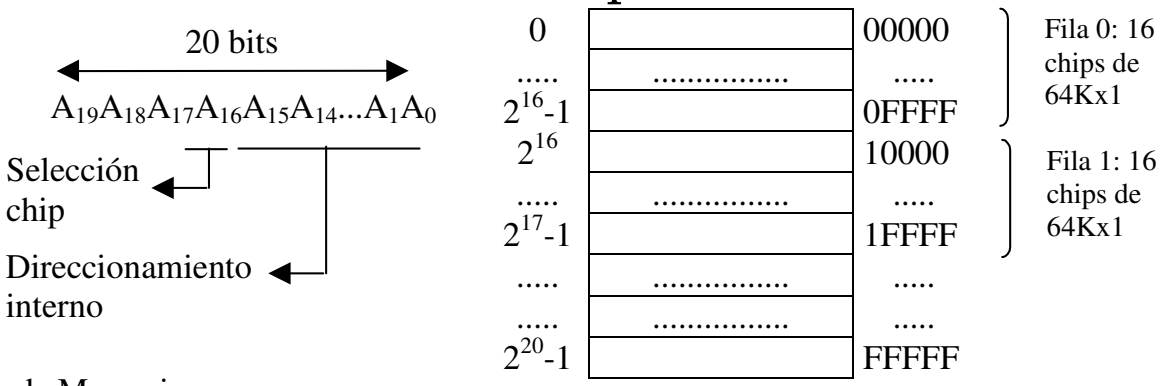
(4) Obtener el número de bits del bus de direcciones que permita seleccionar los chips de memoria.

Como tenemos 2 filas de 16 chips cada una, necesitaremos 1 bit para diferenciar una fila de otra. Por tanto utilizaremos el bit $A_{16}$ para seleccionar los chips de memoria. El resto de direcciones se utilizarán para futuras ampliaciones de memoria del computador.

# Diseño

## Solución 2:

(4) Obtener el número de bits del bus de direcciones que permita seleccionar los chips de memoria.




Mapa de Memoria

| $A_{19}$ | $A_{18}$ | $A_{17}$ | $A_{16}$ | $A_{15}$ | $A_{14}$ | $A_{13}$ | $A_{12}$ | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

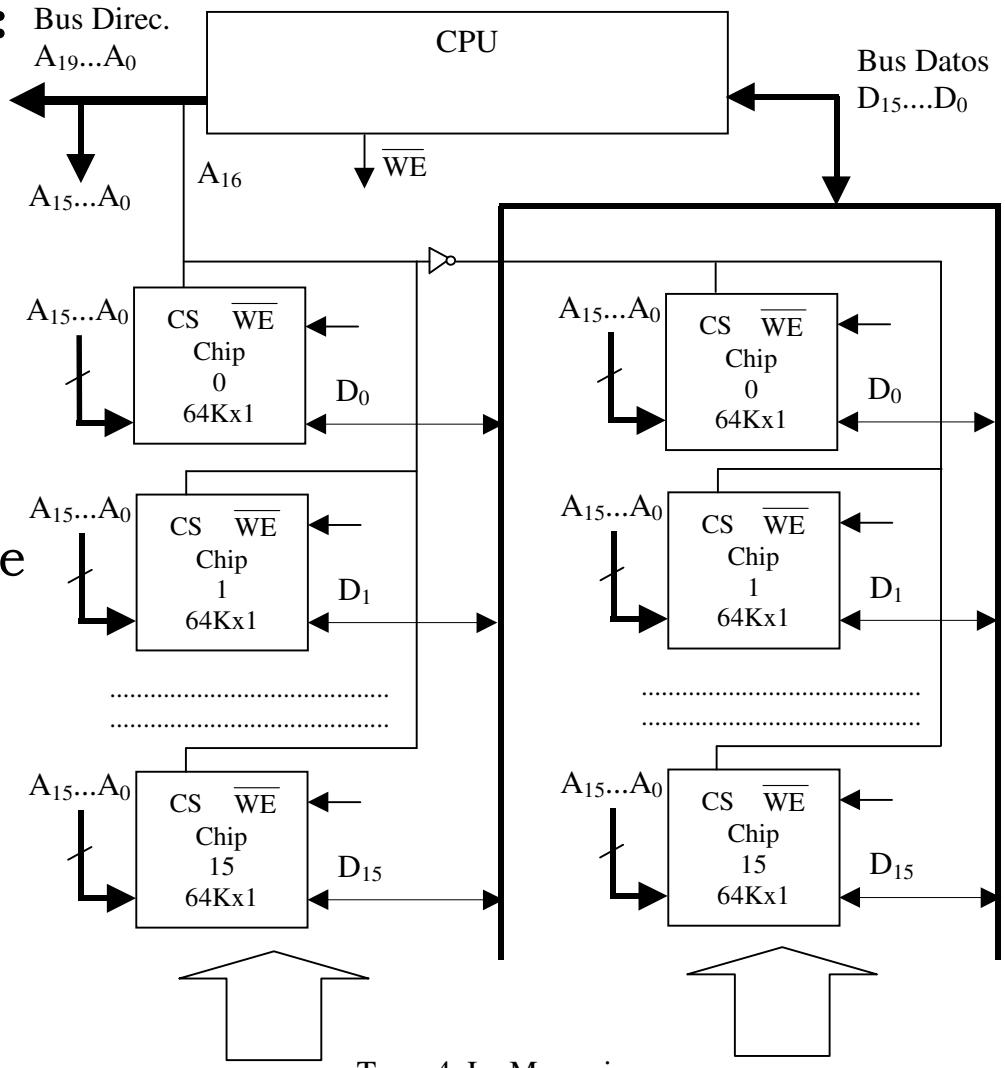
Fila 0: 16 chips de 64Kx1

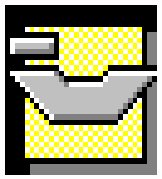
Fila 1: 16 chips de 64Kx1

# Diseño

**Solución 2:**

(5) Dibujar el diagrama de conexiones de la memoria junto con la lógica de selección.

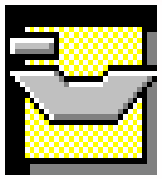




Memoria cache

# Concepto

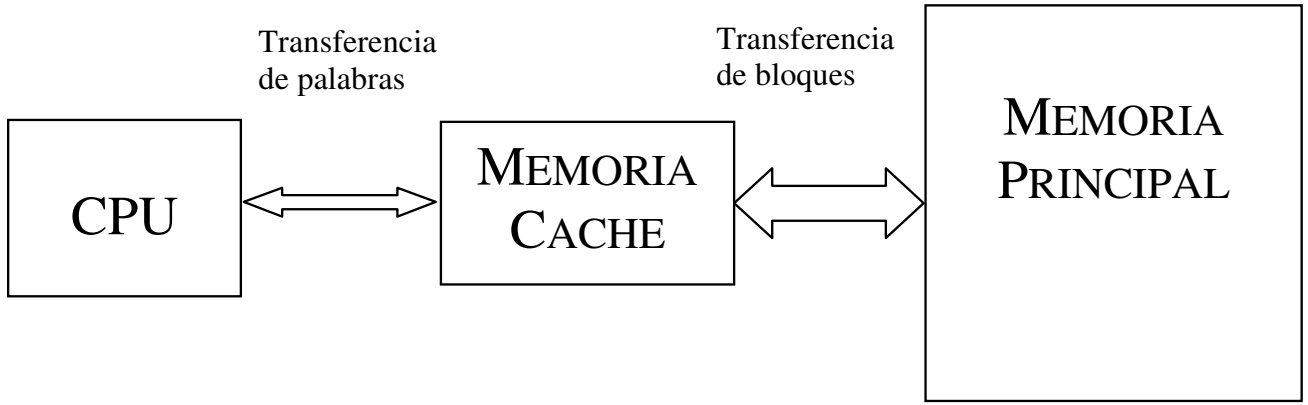
- La memoria de la CPU es más rápida que la memoria principal.
- Lo ideal sería que la memoria principal fuera de la misma tecnología que los registros de la CPU, pero debido a su alto coste se tiende a soluciones intermedias.
- Una solución sería aprovecharnos del principio de localidad y colocar una memoria muy rápida entre la CPU y la memoria principal de tal forma que la CPU acceda más veces a esa memoria que a memoria principal.
- Esta memoria muy rápida deberá ser pequeña para que los costes no sean excesivos. A esta memoria se le denomina **memoria cache**.





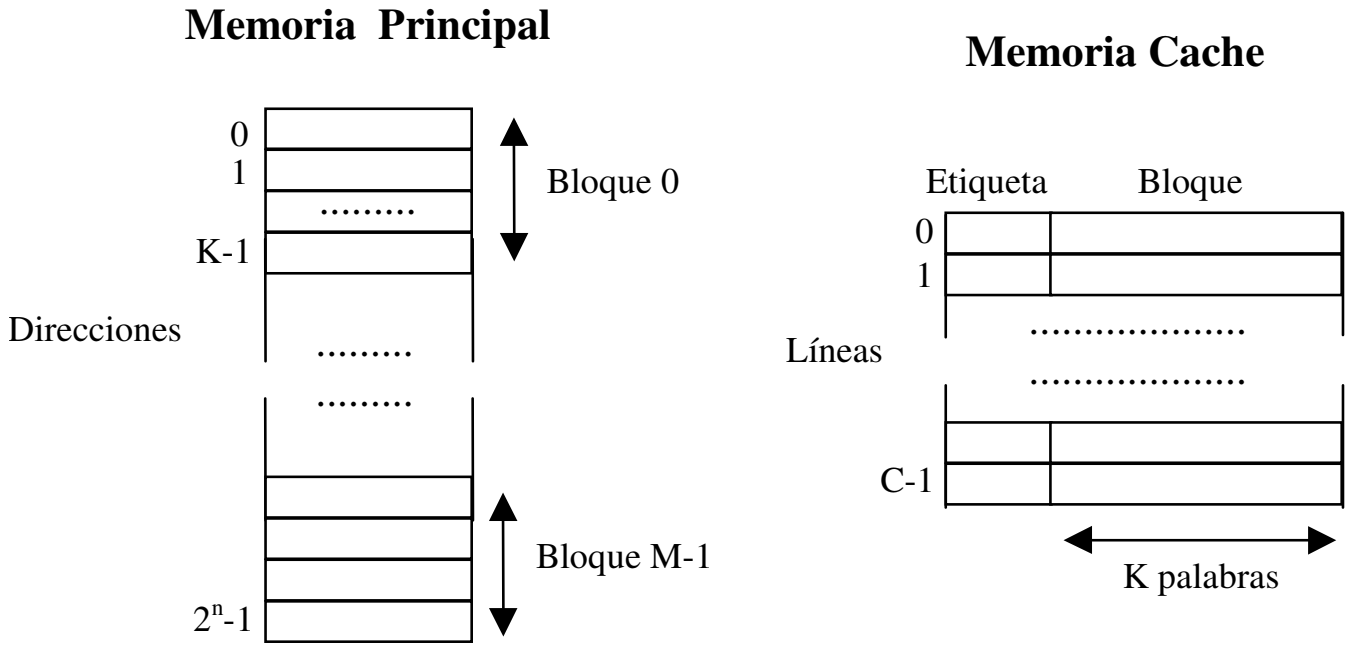
# Concepto

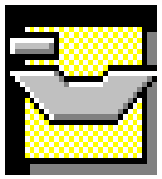
El funcionamiento de la memoria cache se basa en la transferencia de partes (bloques) de la memoria principal y la memoria cache y de la transferencia de palabras entre memoria cache y la CPU.



# Concepto

- La memoria principal de $2^n$ palabras está organizada en M bloques de longitud fija (K palabras/bloque) donde $M=2^n/K$ bloques.
- La memoria cache está dividida en C líneas o particiones de K palabras (C<<M).





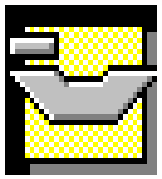


# Concepto

## Tasa de acierto

- Cuando la CPU necesita una palabra de memoria y la encuentra en la memoria cache, se dice que se produce un acierto (hit). Si la palabra no está en la memoria cache se contabiliza un fallo (miss).
- La relación entre el número de aciertos y el número total de referencias a memoria (aciertos+fallos), es la tasa de acierto. En sistemas bien diseñados se suelen conseguir tasas de aciertos de 0.9.

$$Tasa \quad de \quad acierto = \frac{N^{o} \quad aciertos}{N^{o} \quad referencias}$$





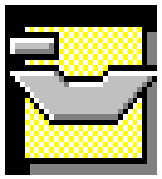
# Parámetros de diseño

## Tamaño de la memoria cache

Plantea un cierto compromiso:

- Debería ser lo suficientemente pequeña como para que el coste medio por bit de información almacenada en la memoria interna del computador estuviese próximo al de la memoria principal.
- Tendría que ser lo suficientemente grande como para que el tiempo de acceso medio total fuese lo más próximo posible al de la memoria cache.

De acuerdo a estudios empíricos se sugiere que el tamaño de una cache esté situado entre 1Kb y 1Mb.





# Parámetros de diseño

## Tamaño del bloque

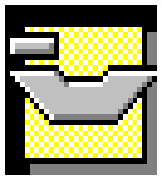
Cuando se va aumentando el tamaño del bloque, a partir de valores muy pequeños la tasa de acierto inicialmente aumentará.

A partir de un cierto tamaño del bloque la tasa de acierto comienza a disminuir.

Surgen dos efectos:

- Cuanto mayor sea el tamaño de los bloques, menos bloques cogerán en la memoria cache y más veces se ejecutará el algoritmo de sustitución de bloques.
- Cuando crece el tamaño de un bloque, cada nueva palabra añadida a ese bloque estará a mayor distancia de la palabra requerida por la CPU, y por tanto es menos probable que sea necesitada a corto plazo.

Se sugiere un tamaño 4-8 unidades direccionables.

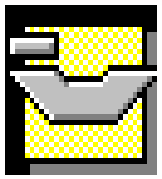




# Parámetros de diseño

## Número de caches

- Cache interna. Nivel 1. Físicamente está ubicada en el mismo chip que el procesador. Los accesos a esta cache se efectúan muy rápido. La capacidad de esta cache es bastante pequeña.
- Cache externa. Nivel 2. Físicamente se ubica fuera del procesador por lo que será más lenta que la cache de nivel 1 pero seguirá siendo más rápida que la memoria principal. Al estar fuera, la capacidad podrá ser mayor que la cache de nivel 1.





# Parámetros de diseño

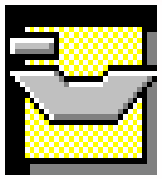
## Contenido de la cache

Una cache que contiene tanto datos como instrucciones presenta las siguientes ventajas:

- Tiene una tasa de aciertos mayor ya que nivela la carga, es decir, si un patrón de ejecución implica muchas captaciones de instrucción que de datos, la cache tenderá a llenarse con instrucciones, y viceversa.
- Sólo se necesita implementar y diseñar una cache, por lo que el coste será más reducido.

Hoy en día, se emplea la ejecución paralela de instrucciones y los diseños pipelining en los que el uso de dos caches da mejores prestaciones.

Ventaja: Elimina la competición entre el procesador de instrucciones y la unidad de ejecución.





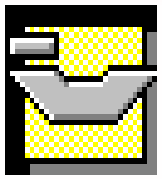
# Parámetros de diseño

## Estrategia de escritura

Antes de que pueda ser reemplazado un bloque que está en la cache, es necesario saber si se ha modificado o no. (Bloque límpio y bloque modificado o sucio).

- Escritura inmediata o directa (write through)

Todas las operaciones de escritura se hacen tanto en la memoria cache como en la memoria principal, lo que se asegura que los contenidos son siempre válidos. El principal inconveniente es que genera mucho tráfico con la memoria principal y puede originar un cuello de botella.





# Parámetros de diseño

## Estrategia de escritura

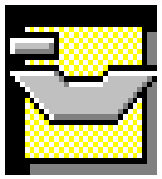
- Post-escritura (write back)

  Las escrituras se realizan sólo en la memoria cache. Asociada a cada línea de la cache existe un bit de modificación. Cuando se escribe en la cache, el bit de modificación se pone a 1. En el caso de reemplazar una línea de la cache, se mira el bit de modificación y si está a 1, se escribirá la línea en memoria principal, mientas que si está a 0 no.

  Inconveniente: se obliga a que los módulos de E/S accedan a memoria principal a través de la memoria cache. Esto complica la circuitería y genera un cuello de botella.

  Ventaja: se utiliza menos ancho de banda de memoria principal haciendo idóneo para su uso en multiprocesadores.

**Problema de coherencia de los datos.**





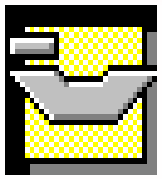
# Parámetros de diseño

## Función de correspondencia

Debido a que existen menos líneas que bloques, se necesita un algoritmo (función) que haga corresponder bloques de memoria principal a líneas de memoria cache.

- Correspondencia directa. El bloque 12 de memoria principal solo podrá almacenarse en la línea 4 (=12 módulo 8).
- Correspondencia asociativa. Se puede almacenar en cualquier línea.
- Correspondencia asociativa por conjuntos. Se puede almacenar en cualquier línea del conjunto 0 (=12 módulo (8/2)).

| Número Línea | Directa | Asociativa | Asociativa por conjuntos | |
|---|---|---|---|---|
| 0 | | | | Conjunto 0 |
| 1 | | | | |
| 2 | | | | Conjunto 1 |
| 3 | | | | |
| 4 | | | | Conjunto 2 |
| 5 | | | | |
| 6 | | | | Conjunto 3 |
| 7 | | | | |

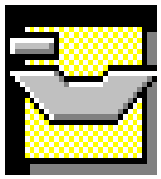




# Parámetros de diseño

## Función de correspondencia

- A continuación vamos a describir las tres técnicas antes enumeradas.
- En cada caso veremos la estructura general y un ejemplo concreto.
- Para los tres casos, vamos a trabajar con un sistema con las siguientes características:
  - El tamaño de la memoria cache es de 4Kb.
  - Los datos se transfieren entre memoria principal y la memoria cache en bloques de 16 bytes (K). Esto nos indica que la cache está organizada en 256 (4096/16) líneas (C).
  - La memoria principal consta de 64Kb, por lo que el bus de direcciones es de 16 bits ($2^{16}$ = 64K). Esto nos indica que en la memoria principal está compuesta por 4096 bloques (M).





# Parámetros de diseño

## Correspondencia directa

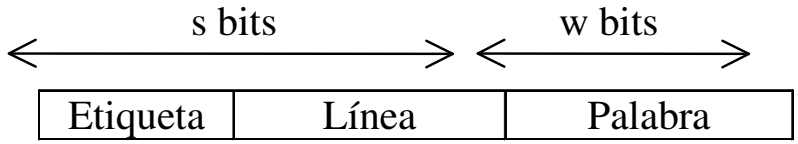
Consiste en hacer corresponder cada bloque de memoria principal a sólo una línea de memoria cache. La función de correspondencia se expresa mediante la siguiente función

Nº de L. de cache = Nº B. M.M módulo Nº de Líneas de M.C.

# Parámetros de diseño

## Correspondencia directa

Cada dirección de la memoria principal puede verse como dividida en tres campos




donde

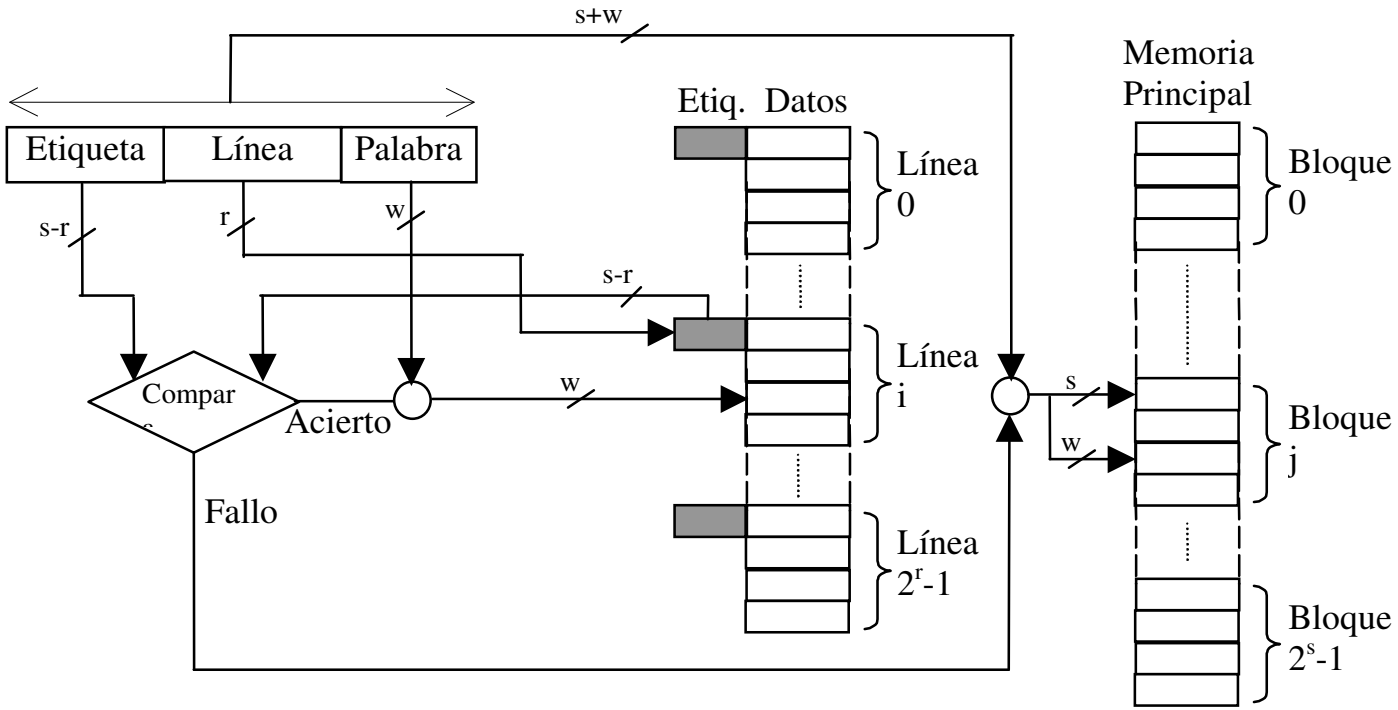
w bits = Identifica cada palabra dentro de un bloque

s bits = Identifica el número de bloque

# Parámetros de diseño

## Correspondencia directa

El uso de una parte de la dirección como número de línea proporciona una asignación única de cada bloque de memoria principal en la cache.

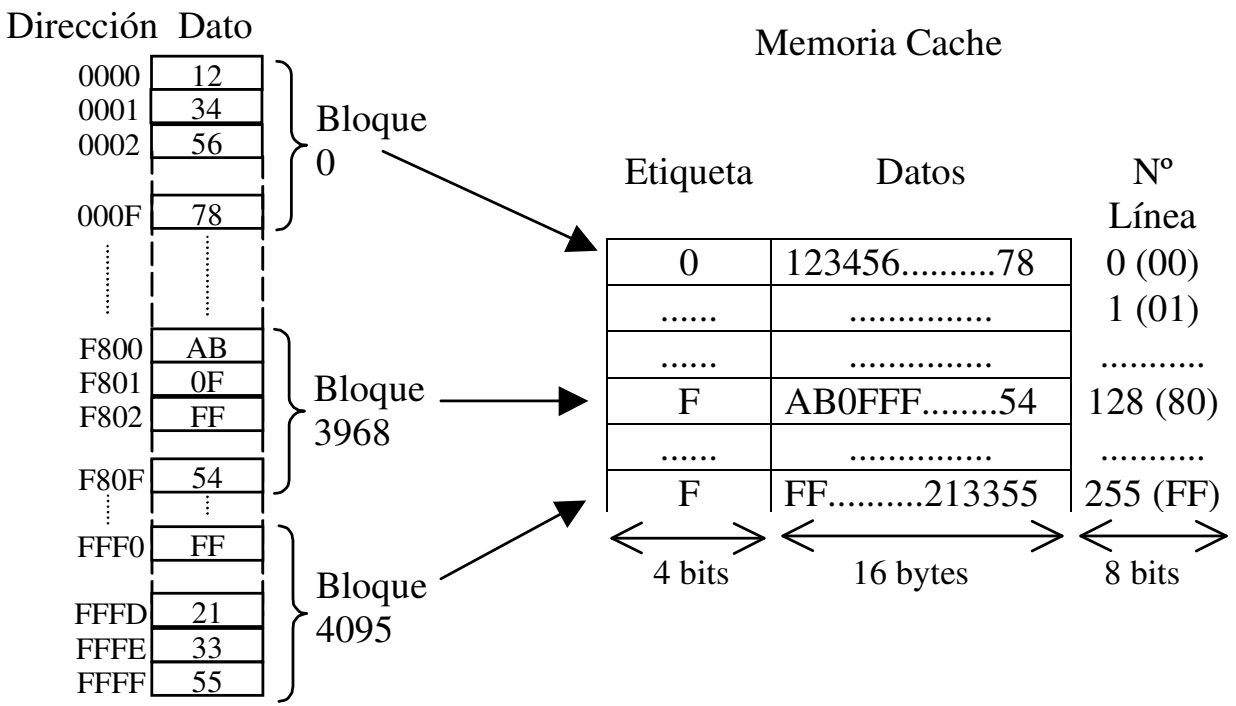


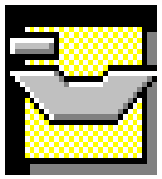
# Parámetros de diseño

## Correspondencia directa

Ejemplo:

| Etiqueta | Línea | Palabra |
|---|---|---|
| 4 bits | 8 bits | 4 bits |







# Parámetros de diseño

## Correspondencia directa

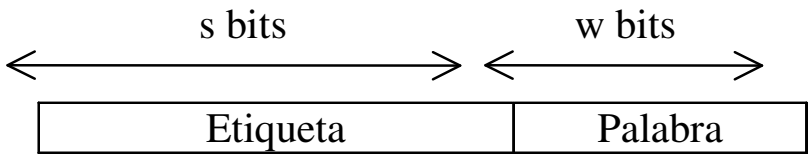
- La técnica de correspondencia directa es simple y poco costosa de implementar.
- Inconveniente: hay una posición concreta de cache para cada bloque dado.

# Parámetros de diseño

## Correspondencia asociativa

Permite que se cargue un bloque de memoria principal en cualquier línea de la memoria cache.

La lógica de control de la memoria cache interpreta una dirección de memoria como una etiqueta y un campo de palabra.




donde

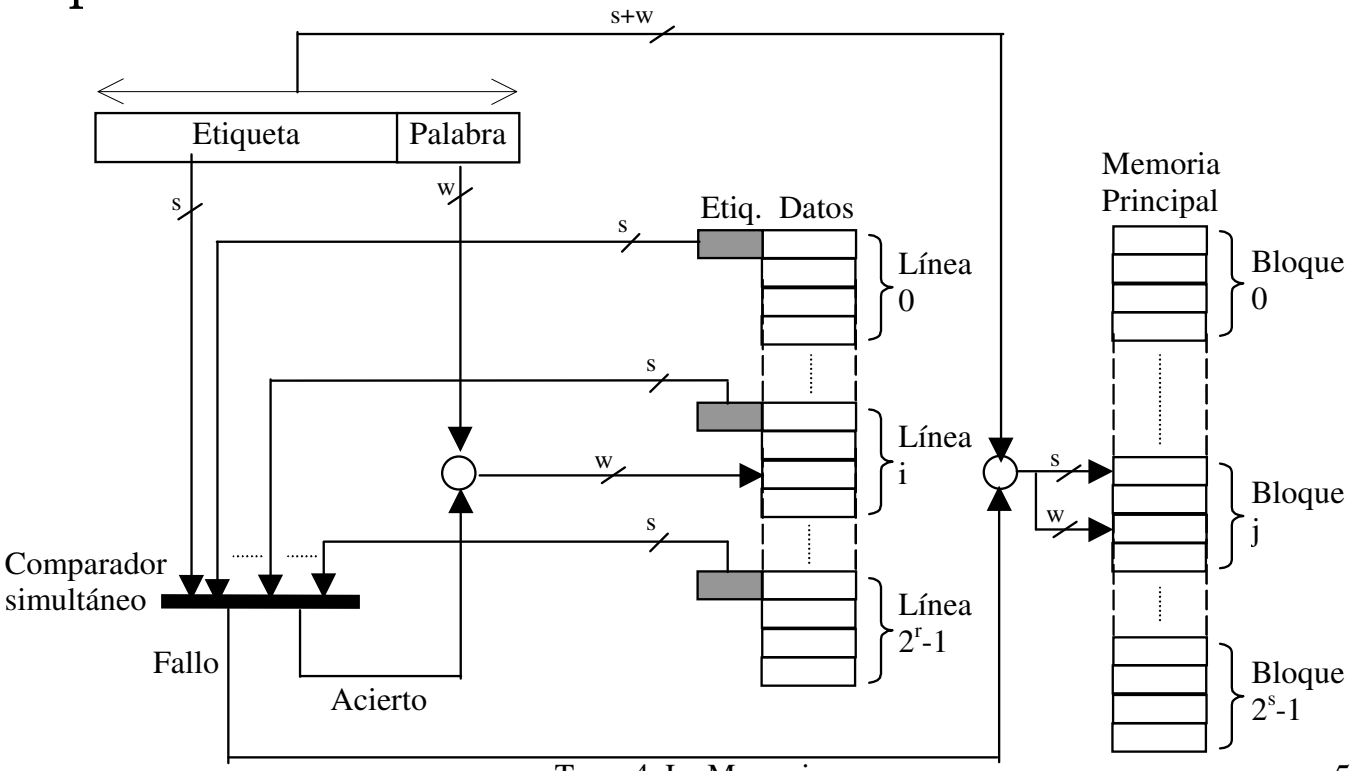
Palabra = Identifica cada palabra dentro de un bloque de MP

Etiqueta = Identifica unívocamente un bloque que MP

# Parámetros de diseño

## Correspondencia asociativa

Para determinar si un bloque está en la memoria cache, se debe examinar simultáneamente todas las etiquetas de las líneas de la memoria cache.

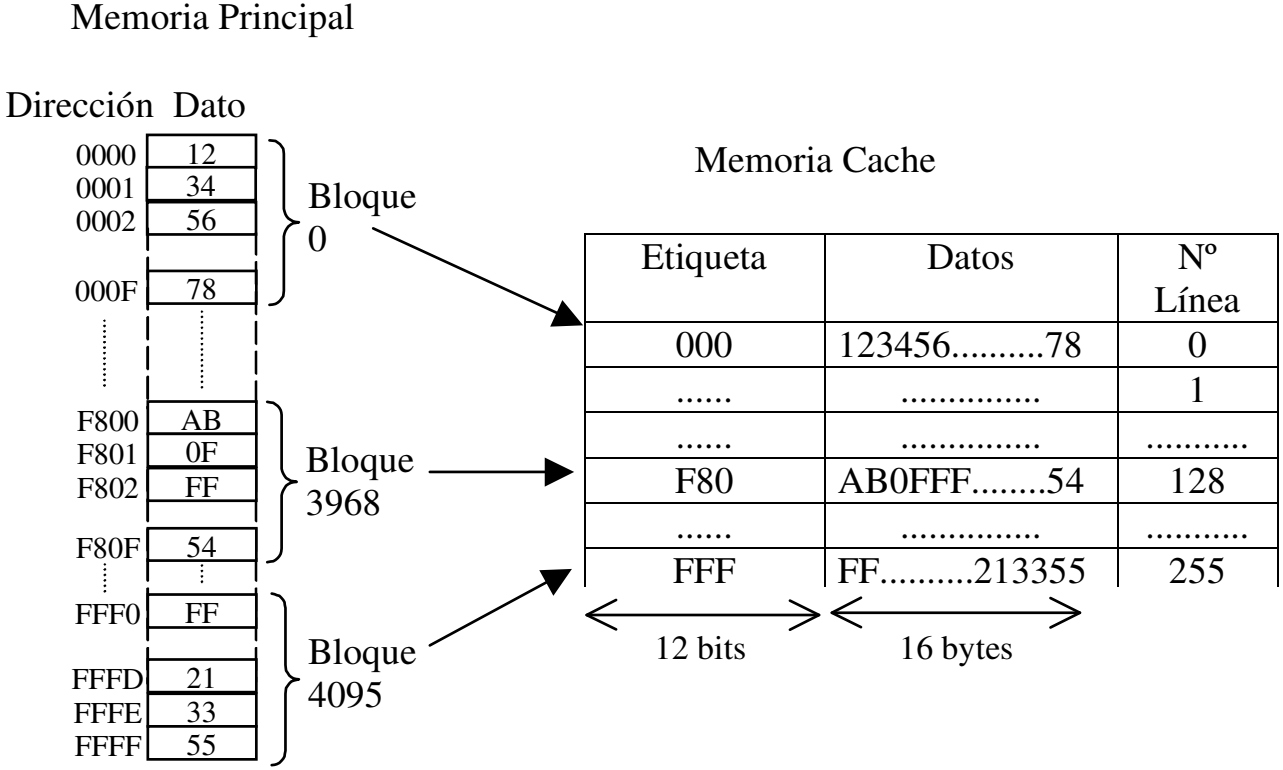


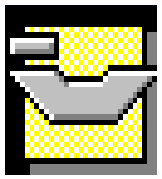
# Parámetros de diseño

## Correspondencia asociativa

Ejemplo:

| Etiqueta | Palabra |
|---|---|
| 12 bits | 4 bits |



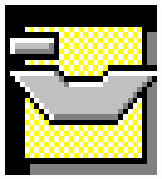




# Parámetros de diseño

## Correspondencia asociativa

- Ventaja: acceso muy rápido.
- Inconveniente: necesidad de una circuitería bastante compleja.





# Parámetros de diseño

## Correspondencia asociativa por conjuntos

Este técnica es un compromiso que trata de aunar las ventajas de las dos técnicas vistas anteriormente.

MC dividida en T conjuntos de L líneas.

Relaciones que se tienen son

$$C = T \times L$$

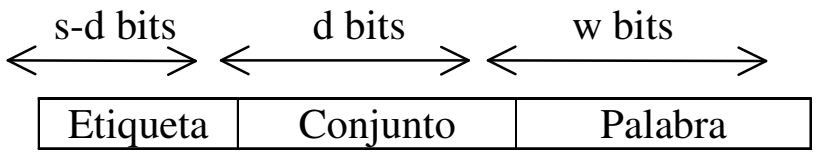
Nº Conjunto M.C.= Nº B MM módulo Nº de C de la MC

El bloque $B_j$ puede asociarse a cualquiera de las líneas del conjunto i.

# Parámetros de diseño

## Correspondencia asociativa por conjuntos

La lógica de control de la cache interpreta una dirección de memoria con tres campos




donde

w bits de menos peso = Palabra dentro de un bloque

s bits = Identifica un bloque de la memoria principal

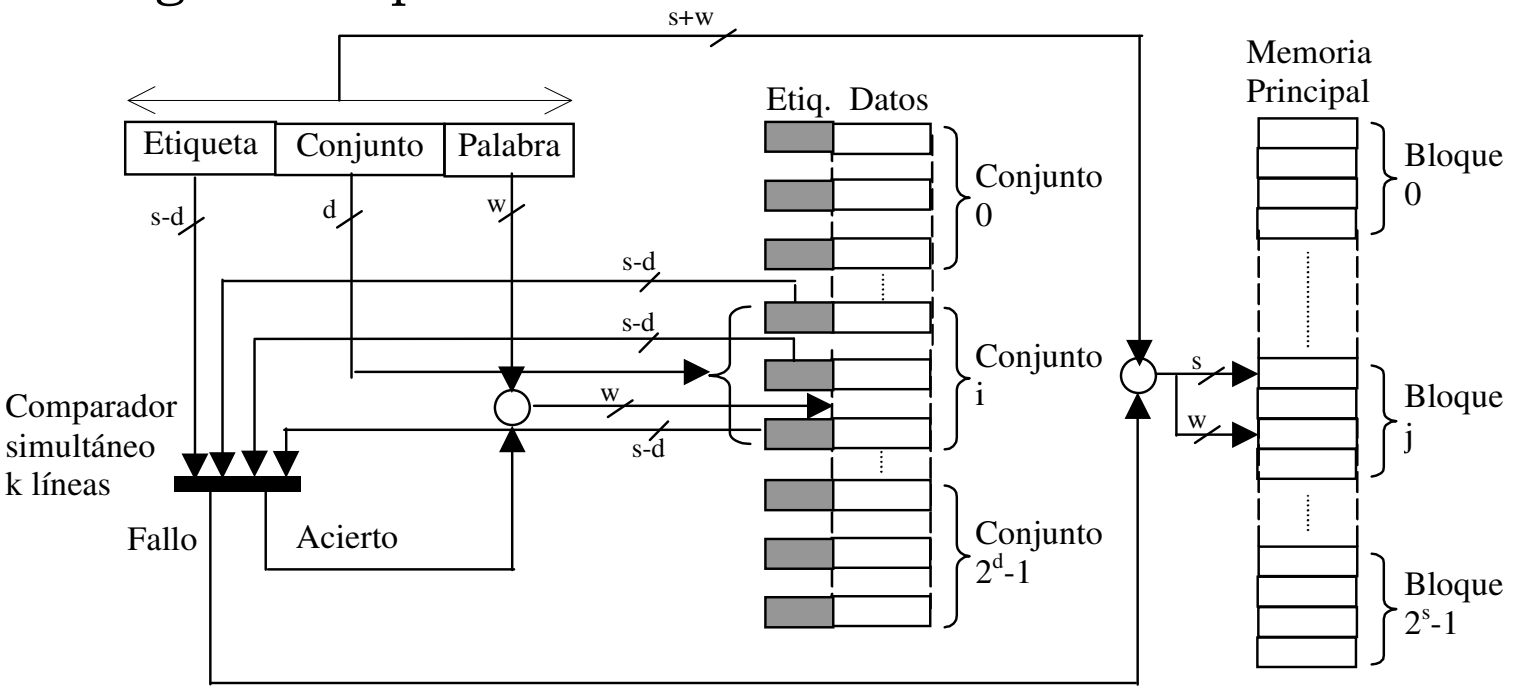
d bits = Especifica uno de los conjuntos de la memoria cache

s-d bits = Etiqueta asociada a las líneas del conjunto d bits

# Parámetros de diseño

## Correspondencia asociativa por conjuntos

Para saber si una dirección está o no en memoria cache, lo primero se aplica correspondencia directa y luego correspondencia asociativa.



# Parámetros de diseño

## Correspondencia asociativa por conjuntos

Ejemplo:

| Etiqueta | Conjunto | Palabra |
|---|---|---|
| 6 bits | 6 bits | 4 bits |

Memoria Principal

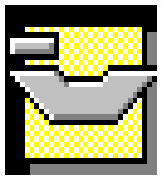
| Dirección | Dato | |
|---|---|---|
| 0000 | 12 | Bloque 0 |
| 0001 | 34 | |
| 0002 | 56 | |
| 000F | 78 | |
| 0400 | AB | Bloque 64 |
| 0401 | 0F | |
| 0402 | FF | |
| 040F | 54 | |
| FFF0 | FF | Bloque 4095 |
| FFFD | 21 | |
| FFFE | 33 | |
| FFFF | 55 | |

Memoria Cache

| Etiqueta | Datos | Nº Conjunto | Etiqueta | Datos |
|---|---|---|---|---|
| 00 | 123456.....78 | 0 | ..... | ..... |
| ..... | ..... | ..... | ..... | ..... |
| ..... | ..... | 64 | ..... | ..... |

| Etiqueta | Datos | Nº Conjunto | Etiqueta | Datos |
|---|---|---|---|---|
| 04 | AB0FFF.....54 | 0 | ..... | ..... |
| ..... | ..... | ..... | ..... | ..... |
| 3F | FF.....213355 | 64 | ..... | ..... |

6 bits — 16 bytes — 6 bits — 16 bytes





# Parámetros de diseño

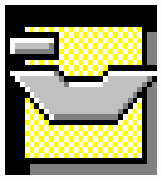
## Algoritmos de sustitución

Cuando un nuevo bloque se transfiere a la memoria cache debe sustituir a uno de los ya existentes si la línea estuviera ocupada.

En el caso de la correspondencia directa la línea no tiene sentido estos algoritmos.

Entre los diferentes algoritmos que se han propuesto destacan los siguientes:

- LRU
- FIFO
- LFU





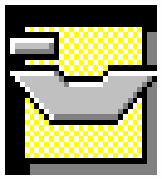
# Parámetros de diseño

## Ejemplos de memorias cache

Los procesadores de Intel han ido incorporando la memoria cache para aumentar su rendimiento.

| Procesador | Nivel 1 | Nivel 2 |
|---|---|---|
| Inferior 80386 | NO | NO |
| 80386 | NO | 16K, 32K, 64K |
| 80846 | 8K datos/instrucciones | 64K, 128K, 256K |
| Pentium | 8K datos y 8K instrucciones | 256K, 512K, 1M |

| Procesador | | Descripción |
|---|---|---|
| 80846 | Cache interna | Tamaño de línea de 16 bytes |
| | | Organización asociativa por conjuntos de 4 vías |
| | Cache externa | Tamaño de línea de 32, 64 ó 128 bytes |
| | | Organización asociativa por conjuntos de 2 vías |
| Pentium | Cache interna | Tamaño de línea de 32 bytes |
| | | Organización asociativa por conjuntos de 2 vías |
| | Cache externa | Tamaño de línea de 32, 64 ó 128 bytes |
| | | Organización asociativa por conjuntos de 2 vías |

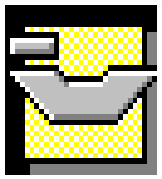




# Concepto

- Una memoria asociativa se caracteriza por el hecho de que la posición de memoria a la que se desea acceder se realiza especificando su contenido o parte de él y no por su dirección.

- A las memorias asociativas también se les denominan memorias direccionables por contenido (CAM: Content Addressable Memory).

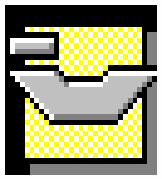




# Estructura de una CAM

- Una memoria asociativa consiste en un conjunto de registros y una matriz de celdas de memoria, con su lógica asociada, organizada en n palabras con m bits/palabra.

- El conjunto de registros está formado por un registro argumento (A) de m bits, un registro máscara (K) de m bits y un registro de marca (M) de n bits.





# Estructura de una CAM

- Cada palabra de la memoria se compara simultáneamente con el contenido del registro argumento, y se pone a 1 el bit del registro de marca asociado a aquellas palabras cuyo contenido coincide con el del registro argumento.

- Al final de este proceso, aquellos bits del registro de marca que están a 1 indican la coincidencia de las correspondientes palabras de la memoria asociativa y del registro de argumento.

# Estructura de una CAM

La comparación simultánea se realiza bit a bit.

El bit $A_j$ (j=1,2,...,m) del registro argumento se compara con todos los bits de la columna j si $K_j$=1.

Si existe coincidencia entre todos los bits $M_i$=1. En caso contrario, $M_i$=0.

| | | |
|---|---|---|
| 0 | Alicante | 0 |

Registro argumento

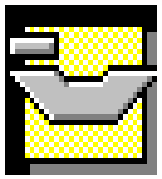
| | | |
|---|---|---|
| 0 | 1 | 0 |

Registro máscara

| | | |
|---|---|---|
| Juan | Alicante | 965254512 |
| Pepe | Elda | 965383456 |
| Ana | Alicante | 965907799 |
| Laura | Elche | 965442233 |
| Pepe | Alicante | 965223344 |
| Paco | Elche | 966664455 |
| Paqui | Petrer | 965375566 |
| Pepi | Alicante | 965286677 |

| |
|---|
| 1 |
| 0 |
| 1 |
| 0 |
| 1 |
| 0 |
| 0 |
| 1 |

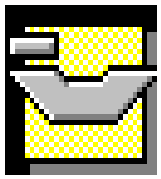
Registro de marca





# Estructura de una CAM

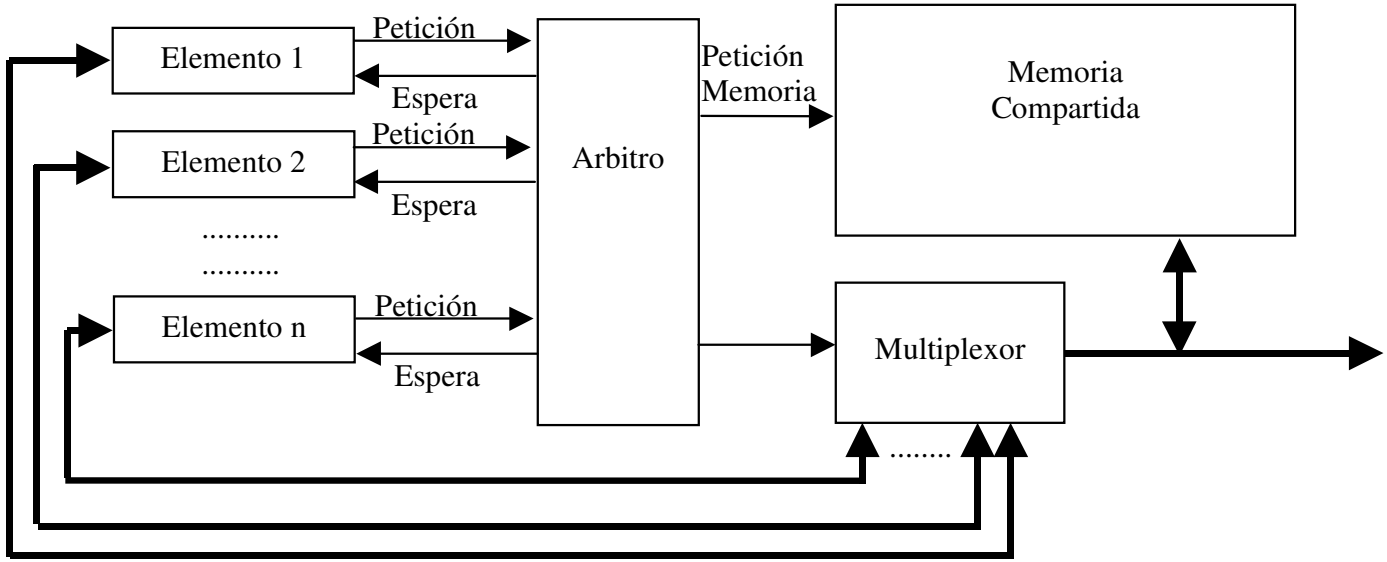
- Por regla general, en la mayoría de las aplicaciones la memoria asociativa almacena una tabla que no tiene, para una máscara dada, dos filas iguales.
- Las memorias asociativas se utilizan sobre todo con memorias cache de tal forma que la identificación de la etiqueta de cada línea se realice de forma simultánea.
- La TAG RAM es un claro ejemplo de memoria asociativa utilizada como parte de memoria cache en los sistemas con Pentium de Intel.



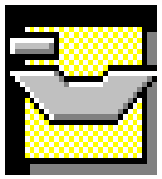
Memoria compartida

# Concepto

Necesidad de que diferentes dispositivos tengan acceso a una misma unidad de memoria.




El árbitro es el elemento encargado de permitir el acceso a la unidad de memoria, en un instante dado, a cada uno de los elementos que solicitan dicho recurso.



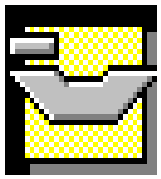


# Concepto

- El árbitro se diseña de forma que asigne un tiempo de servicio, en promedio, análogo a todas la unidades que solicitan el recurso.

  Existen diferentes estrategias:

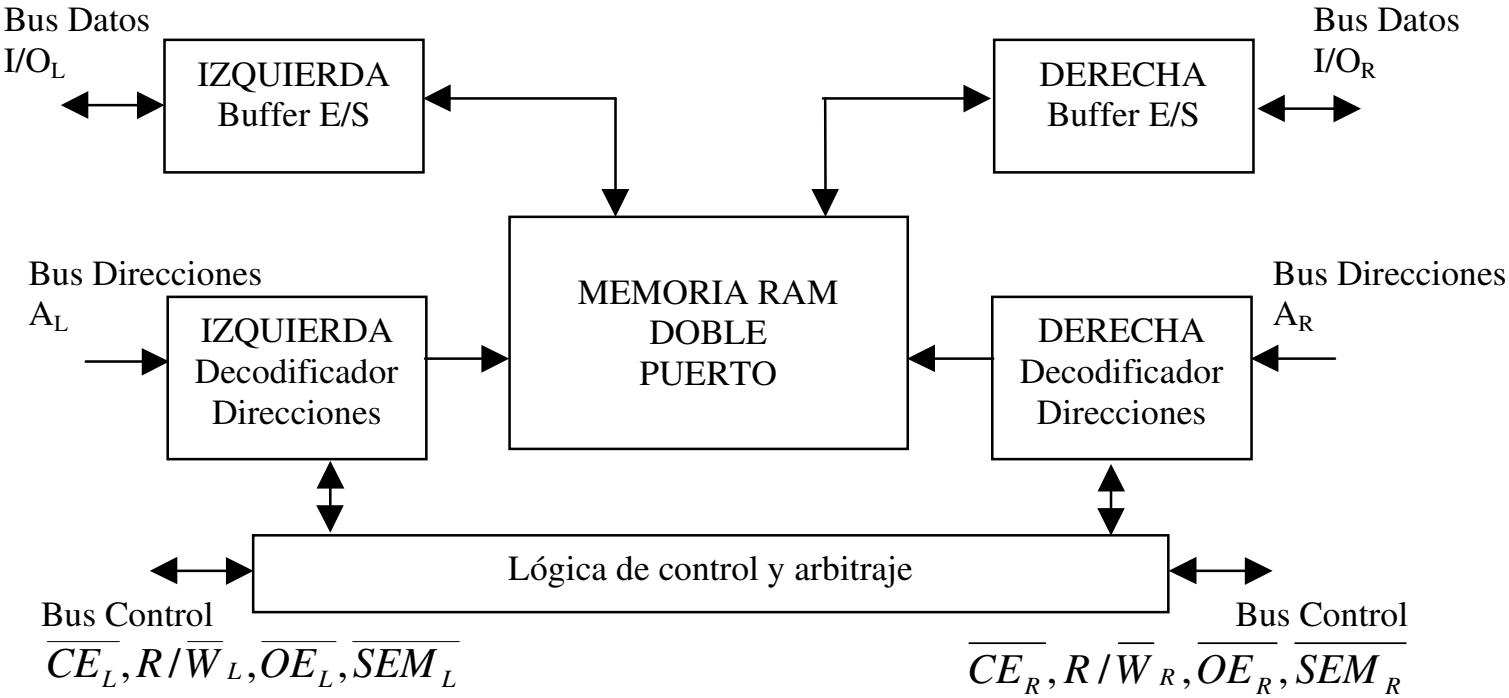
  - Asignación de la menor prioridad al elemento servido.
  - Rotación de prioridades. En un estado cualquiera, el próximo estado se calcula rotando el orden de prioridades actual hasta que el elemento al que se acaba de dar servicio tiene la menor prioridad.

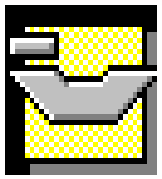


Memoria compartida

# Memorias de doble puerto

Son memorias compartidas que permiten trabajar con dos elementos a la vez. Se basa en duplicar los buses, decodificadores, etc.





Memoria compartida

# Memorias de doble puerto

La memoria de doble puerto tiene prácticamente todos los componentes duplicados (puerto izquierdo (LEFT) y puerto derecho (RIGHT)).

| Puerto Izquierdo | Puerto Derecho | Descripción |
|---|---|---|
| $I/O_L$ | $I/O_R$ | Bus de datos |
| $A_L$ | $A_R$ | Bus de direcciones |
| $\overline{CE_L}$ | $\overline{CE_R}$ | Selección de chip |
| $R/\overline{W}_L$ | $R/\overline{W}_R$ | Lectura/Escritura |
| $\overline{OE_L}$ | $\overline{OE_R}$ | Habilita lectura |
| $\overline{SEM_L}$ | $\overline{SEM_R}$ | Habilita semáforo |

La memoria VRAM es un ejemplo claro de memoria de doble puerto. A ella puede acceder simultáneamente el controlador del monitor y el procesador de la tarjeta gráfica.