

Tema 3

Subsistema de memoria en
sistemas empuotrados

Subsistema de memoria en sistemas empuotrados. (3+1 horas)

3.1 Espacio de memoria

3.2 Memoria Cache

3.3 Memoria principal

3.4 Memoria auxiliar o de almacenamiento

3.5 Acceso directo a memoria

3.6 Rendimiento y memoria

3.1 Espacio de memoria

- El objetivo de organizar la memoria de un ordenador como una jerarquía de memorias es aprovechar el principio de localidad.
- **Consideración:** las memorias más rápidas son las más caras por bit y por tanto suelen ser más pequeñas.

Tabla: Tiempos de acceso y precios de las distintas tecnologías de memoria más comunes (año 2009).

Tecnología	Tiempo de acceso típico	€ por MB
SRAM	1 ns	20 €
SDRAM	5 ns	0,01 €
Disco magnético	8.500.000 ns	0,0001 €

- **Objetivo:** mucha memoria + tecnología barata + alta velocidad

Jerarquía de memoria

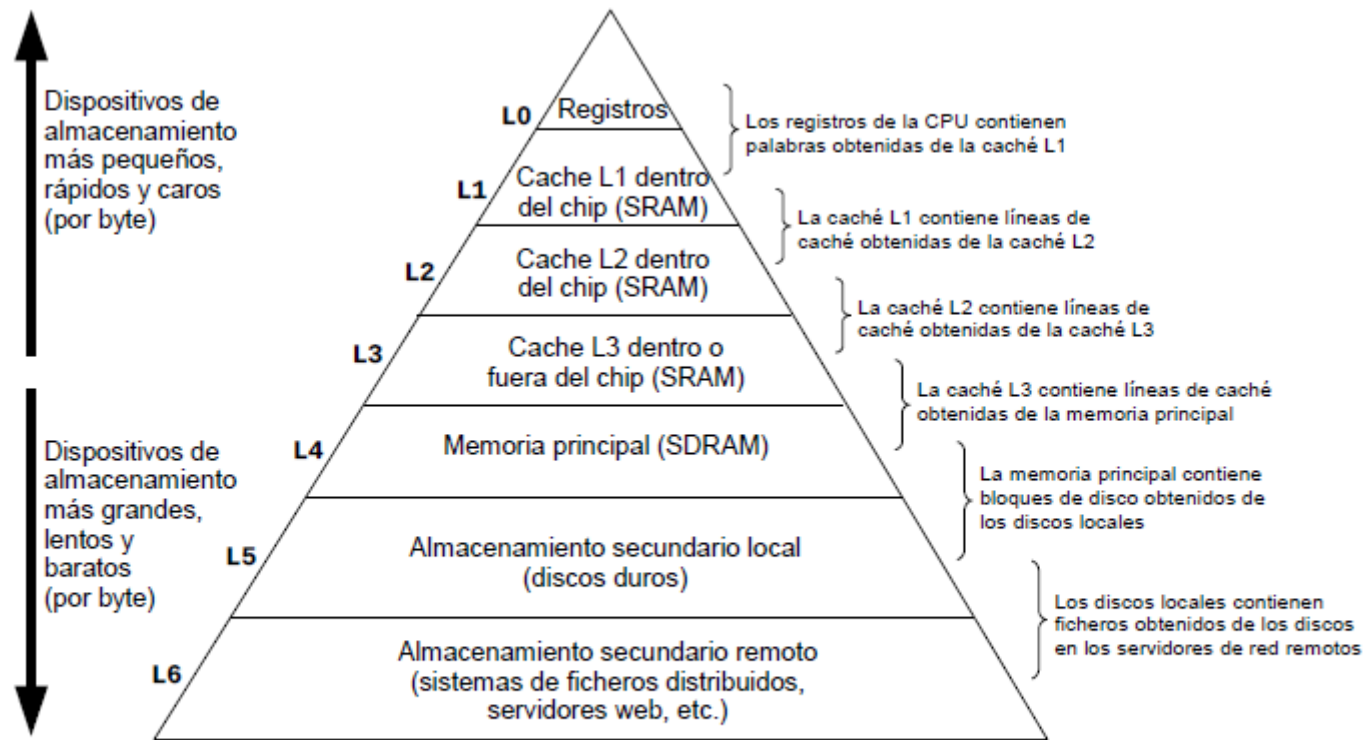


Figura: Estructura básica de la jerarquía de memoria.

Subsistema de memoria en sistemas empuotrados. (3+1 horas)

3.1 Espacio de memoria

3.2 Memoria Cache

3.3 Memoria principal

3.4 Memoria auxiliar o de almacenamiento

3.5 Acceso directo a memoria

3.6 Rendimiento y memoria

3.2 Memoria Cache

- Correspondencia directa

Cada dirección de memoria se corresponde con una única posible posición en la caché

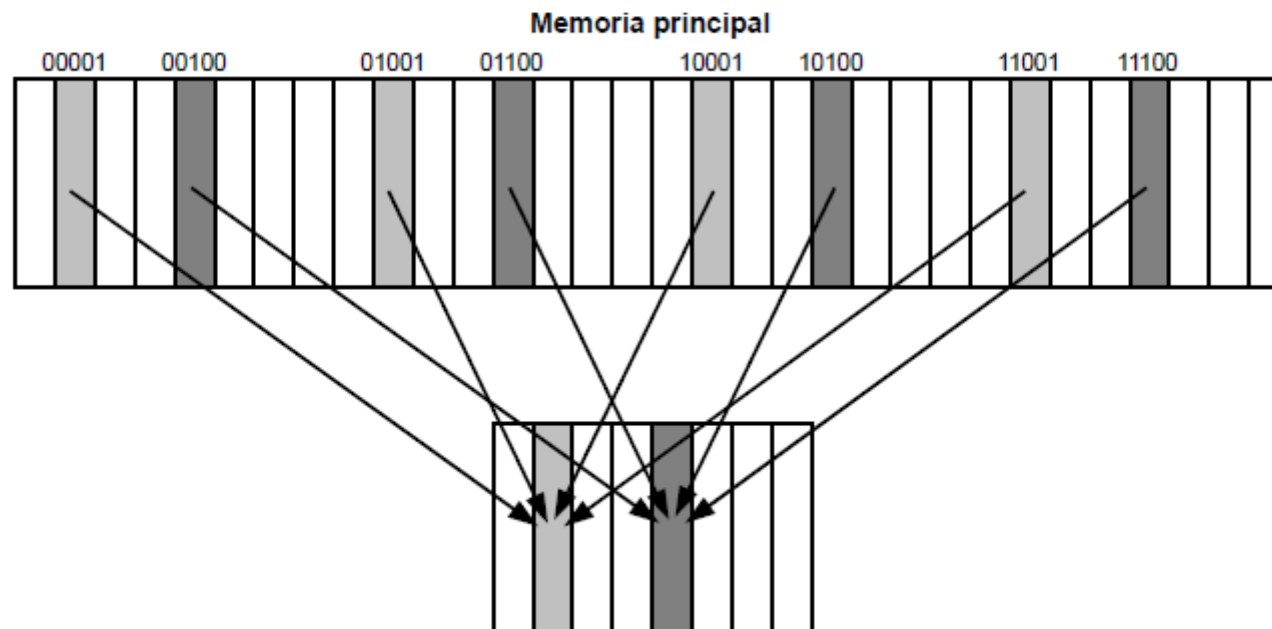
posición en caché = (dirección de bloque en memoria) MODULO (nº de posiciones de la caché)

Bloques (líneas) monopalabra (bloque = 1 palabra)

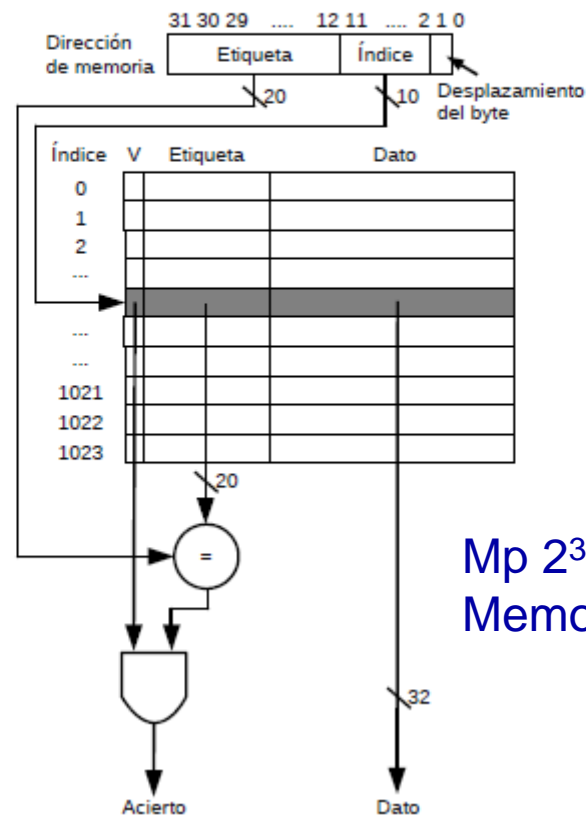
posición en caché = (dirección de palabra en memoria) MODULO (nº de posiciones de la caché)

Cache directa de bloques monopalabra

- Ejemplo: caché de correspondencia directa con 8 posiciones. Memoria principal de 32 palabras.
- Una palabra de dirección **X** se lleva a caché a la posición **X MOD 8**: los 3 ($\log_2 8$) bits de menor peso servirán de índice de la caché.



- **Bit de validez:** indica si una posición de caché contiene datos válidos.



1 palabra 4 bytes

M_p 2^{32} bytes
 Memoria Cache 2^{12} bytes

- Continuando con el ejemplo de la memoria principal de 32 palabras y la caché de 8 palabras:

Tabla: Secuencia de ocho peticiones de acceso a memoria a una caché de correspondencia directa de 8 posiciones de un byte cada una.

Peticiones	Dir) ₁₀	Dir) ₂	Posición caché	Etiqueta caché	Acierto o Fallo
1º	22	10110	110	10	Fallo
2º	26	11010	010	11	Fallo
3º	22	10110	110	10	Acierto
4º	26	11010	010	11	Acierto
5º	16	10000	000	10	Fallo
6º	4	00100	100	00	Fallo
7º	16	10000	000	10	Acierto
8º	18	10010	010	10	Fallo

Tabla: Detalles de los contenidos de una caché durante ocho peticiones de palabras después de cada petición que provoca fallo de caché (1/3).

Índice	V	Etiqueta	Dato	Índice	V	Etiqueta	Dato
000	N			000	N		
001	N			001	N		
010	N			010	N		
011	N			011	N		
100	N			100	N		
101	N			101	N		
110	N			110	S	10	M(10110)
111	N			111	N		
a) Estado inicial de la caché después de arrancar				b) Después de tratar el fallo de la dirección de memoria (10110)			

Tabla: Detalles de los contenidos de una caché durante ocho peticiones de palabras después de cada petición que provoca fallo de caché (2/3).

Índice	V	Etiqueta	Dato	Índice	V	Etiqueta	Dato
000	N			000	S	10	M(10000)
001	N			001	N		
010	S	11	M(11010)	010	S	11	M(11010)
011	N			011	N		
100	N			100	N		
101	N			101	N		
110	S	10	M(10110)	110	S	10	M(10110)
111	N			111	N		
c) Después de tratar el fallo de la dirección de memoria (11010)				d) Después de tratar el fallo de la dirección de memoria (10000)			

Tabla: Detalles de los contenidos de una caché durante ocho peticiones de palabras después de cada petición que provoca fallo de caché (3/3).

Índice	V	Etiqueta	Dato	Índice	V	Etiqueta	Dato
000	S	10	M(10000)	000	S	10	M(10000)
001	N			001	N		
010	S	11	M(11010)	010	S	10	M(10010)
011	N			011	N		
100	S	00	M(00100)	100	S	00	M(00100)
101	N			101	N		
110	S	10	M(10110)	110	S	10	M(10110)
111	N			111	N		
e) Después de tratar el fallo de la dirección de memoria (00100)				f) Después de tratar el fallo de la dirección de memoria (10010)			

Tamaño total de una caché (en bits) = n° de posiciones \times
(n° bits control + tamaño etiqueta + tamaño bloque)

donde, para una caché de correspondencia directa con bloques monopalabra:

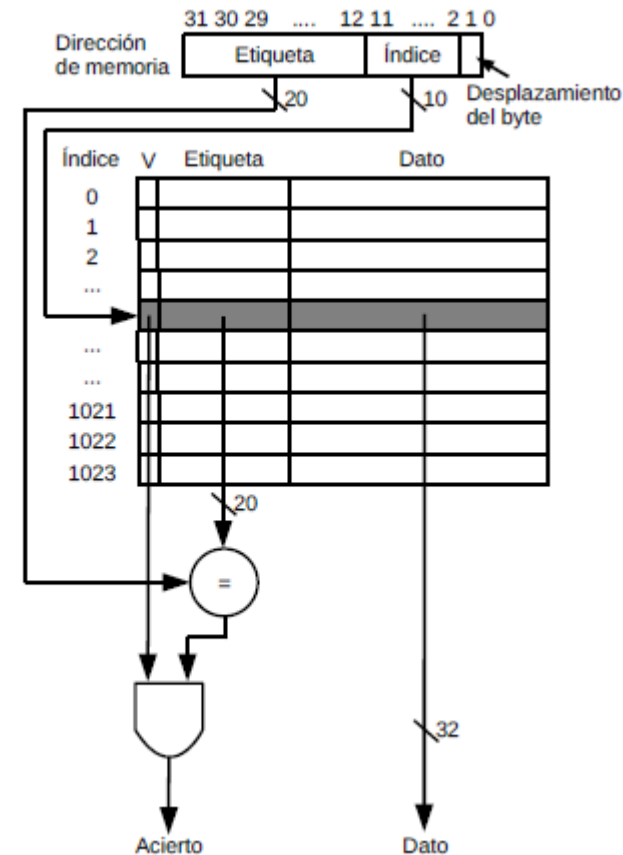
- Tamaño bloque = (n° de palabras por bloque) \times
(n° de bits por palabra) = $1 \times$ (n° de bits por palabra).
- Tamaño etiqueta = (tamaño dirección) – (n° bits para índice) –
(n° bits para el desplazamiento de byte).
- N° bits para índice = \log_2 (n° de posiciones).
- N° bits para el desplazamiento de byte =
 \log_2 (n° de bytes por palabra).

- Tamaño total de una caché, según el ejemplo:

- N° bits para desplazamiento de byte = $\log_2 (4) = 2$ bits.
- N° bits para índice = $\log_2 (2^{10}) = 10$ bits.
- Tamaño etiqueta = $(32 - 10 - 2) = 20$ bits.
- Tamaño bloque = 1×32 bits.

- Por lo tanto:

$$N^{\circ} \text{ total de bits} = 2^{10} \times (1 + 20 + 32) = 54272 \text{ bits} = 6784 \text{ bytes.}$$



Correspondencia directa con bloques multipalabra

Tras un fallo, se traerá a caché la palabra buscada y un grupo de palabras adyacentes a ella y que, por tanto, tendrán una alta probabilidad de ser referenciadas en breve.

- Posición en caché = (dirección de bloque en memoria) MODULO (número de bloques de la caché)
- Dirección de bloque en memoria = (dirección de palabra en memoria) DIV (número de palabras por bloque)

Correspondencia directa con bloques multipalabra

Dir. de bloque	Dir. de palabra	Dir. de byte	Byte de memoria
0000	00000	0000000	
		0000001	
		0000010	
		0000011	
	00001	0000100	
		0000101	
		0000110	
		0000111	
0001	00010	0001000	
		0001001	
		0001010	
		0001011	
	00011	0001100	
		0001101	
		0001110	
		0001111	
...

- direcciones de 7 bits,
- palabras de 4 bytes (32 bits)
- bloques de 2 palabras (64 bits).

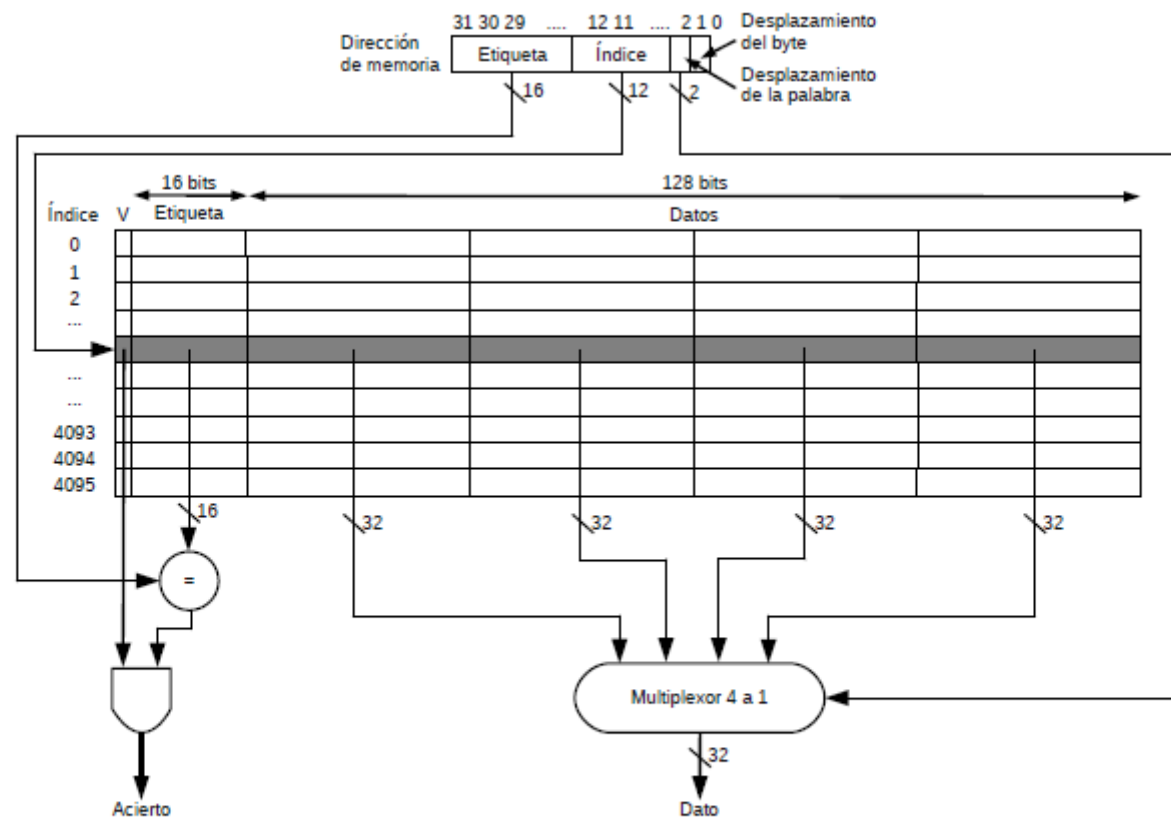


Figura: Memoria caché de 64 KB con bloques de cuatro palabras y palabras de 32 bits.

Dirección del dato solicitado por el procesador			
32 bits			
Dirección de palabra			Desplazamiento de byte
30 bits			2 bits
Dirección de bloque		Desplazamiento de palabra	Desplazamiento de byte
28 bits		2bits	2 bits
Etiqueta	Posición en caché	Desplazamiento de palabra	Desplazamiento de byte
16 bits	12 bits	2bits	2 bits

Memoria cache asociativa por conjuntos

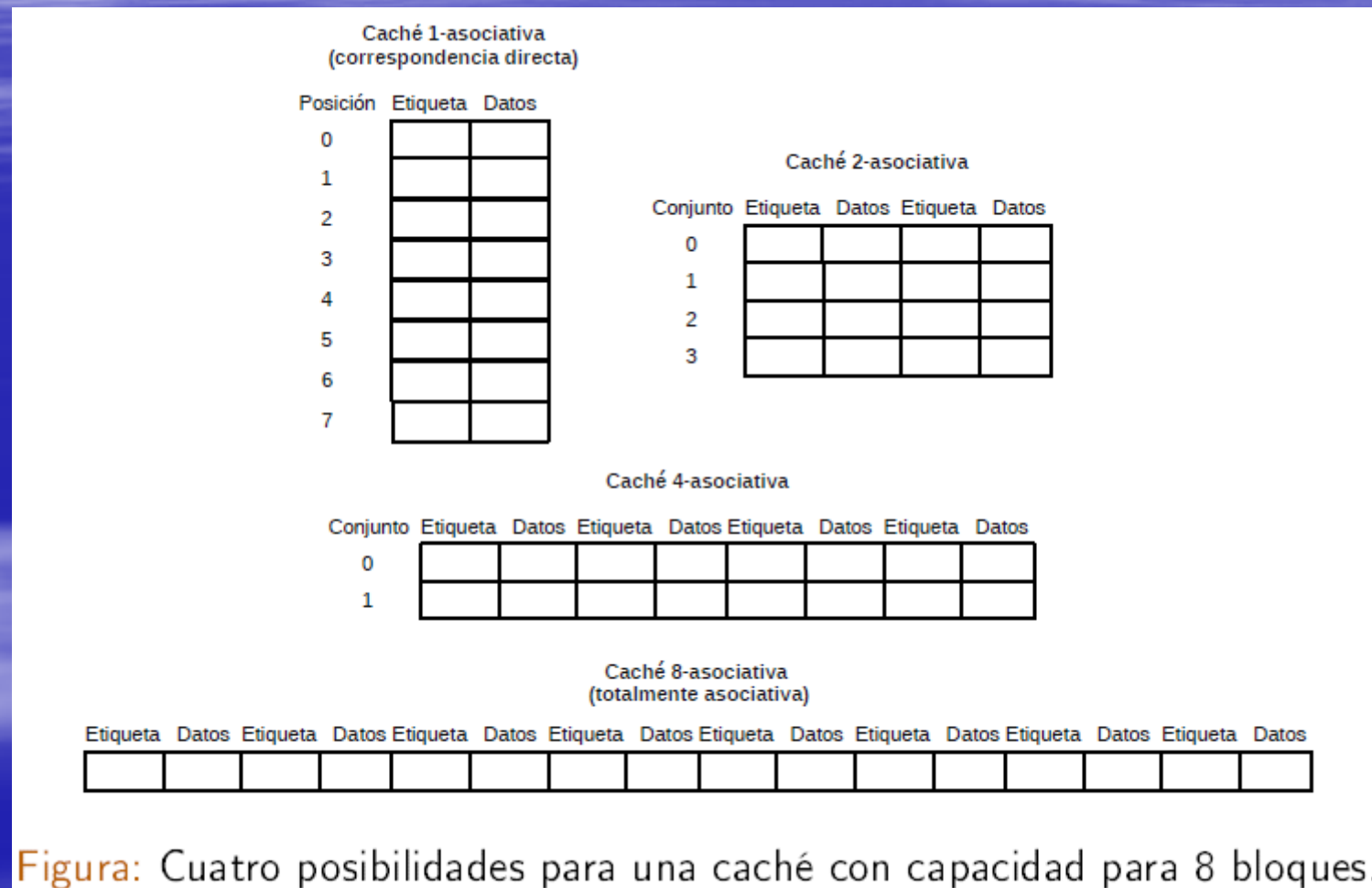


Tabla: Evolución del contenido de una memoria caché de correspondencia directa de 4 posiciones.

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Pos 0	Pos 1	Pos 2	Pos 3

Tabla: Evolución del contenido de una memoria caché de correspondencia directa de 4 posiciones.

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Pos 0	Pos 1	Pos 2	Pos 3
0	0000	00	Fallo	M[0]			
8	1000	00	Fallo	M[8]			

Tabla: Evolución del contenido de una memoria caché de correspondencia directa de 4 posiciones.

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Pos 0	Pos 1	Pos 2	Pos 3
0	0000	00	Fallo	M[0]			
8	1000	00	Fallo	M[8]			
0	0000	00	Fallo	M[0]			

Tabla: Evolución del contenido de una memoria caché de correspondencia directa de 4 posiciones.

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Pos 0	Pos 1	Pos 2	Pos 3
0	0000	00	Fallo	M[0]			
8	1000	00	Fallo	M[8]			
0	0000	00	Fallo	M[0]			
6	0110	10	Fallo	M[0]		M[6]	

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Pos 0	Pos 1	Pos 2	Pos 3
0	0000	00	Fallo	M[0]			
8	1000	00	Fallo	M[8]			
0	0000	00	Fallo	M[0]			
6	0110	10	Fallo	M[0]		M[6]	
8	1000	00	Fallo	M[8]		M[6]	

Tabla: Evolución del contenido de una memoria caché asociativa con 2 conjuntos de 2 bloques.

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Conjunto 0		Conjunto 1	

Tabla: Evolución del contenido de una memoria caché asociativa con 2 conjuntos de 2 bloques.

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Conjunto 0		Conjunto 1	
0	0000	0	Fallo	M[0]			

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Conjunto 0		Conjunto 1	
0	0000	0	Fallo	M[0]			
8	1000	0	Fallo	M[0]	M[8]		

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Conjunto 0		Conjunto 1	
0	0000	0	Fallo	M[0]			
8	1000	0	Fallo	M[0]	M[8]		
0	0000	0	Acierto	M[0]	M[8]		

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Conjunto 0		Conjunto 1	
0	0000	0	Fallo	M[0]			
8	1000	0	Fallo	M[0]	M[8]		
0	0000	0	Acierto	M[0]	M[8]		
6	0110	0	Fallo	M[0]	M[6]		

Dir. bloque		Pos. caché	Acierto o fallo	Contenido de la caché			
Base 10	Base 2			Conjunto 0		Conjunto 1	
0	0000	0	Fallo	M[0]			
8	1000	0	Fallo	M[0]	M[8]		
0	0000	0	Acierto	M[0]	M[8]		
6	0110	0	Fallo	M[0]	M[6]		
8	1000	0	Fallo	M[8]	M[6]		

Tabla: Evolución del contenido de una memoria caché totalmente asociativa.

Dir. bloque		Acierto o fallo	Contenido de la caché			
Base 10	Base 2		Conjunto único			

Dir. bloque		Acierto o fallo	Contenido de la caché			
Base 10	Base 2		Conjunto único			
0	0000	Fallo	M[0]			

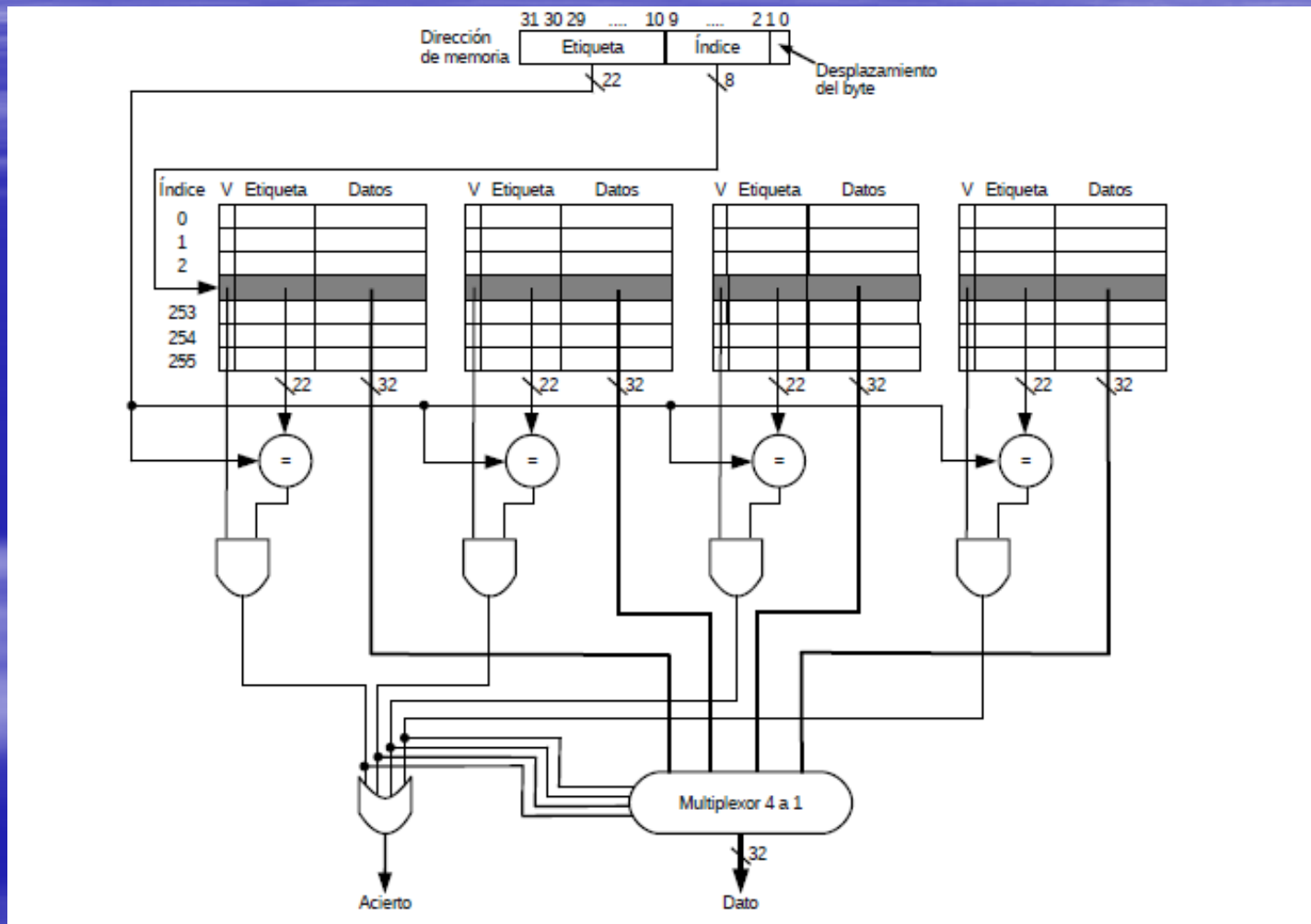
Dir. bloque		Acierto o fallo	Contenido de la caché			
Base 10	Base 2		Conjunto único			
0	0000	Fallo	M[0]			
8	1000	Fallo	M[0]	M[8]		

Dir. bloque		Acierto o fallo	Contenido de la caché			
Base 10	Base 2		Conjunto único			
0	0000	Fallo	M[0]			
8	1000	Fallo	M[0]	M[8]		
0	0000	Acierto	M[0]	M[8]		

Dir. bloque		Acierto o fallo	Contenido de la caché			
Base 10	Base 2		Conjunto único			
0	0000	Fallo	M[0]			
8	1000	Fallo	M[0]	M[8]		
0	0000	Acierto	M[0]	M[8]		
6	0110	Fallo	M[0]	M[8]	M[6]	

Dir. bloque		Acierto o fallo	Contenido de la caché			
Base 10	Base 2		Conjunto único			
0	0000	Fallo	M[0]			
8	1000	Fallo	M[0]	M[8]		
0	0000	Acierto	M[0]	M[8]		
6	0110	Fallo	M[0]	M[8]	M[6]	
8	1000	Acierto	M[0]	M[8]	M[6]	

Construcción de una caché 4asociativa



- Aumentar la asociatividad requiere más comparadores, así como
- más bits para las etiquetas de los bloques de caché.
- Ejemplo: caché de 4096 (2^{12}) bloques monopalabra. Direcciones de 32 bits

	Nº Conjuntos	Nº bits para una etiqueta	Nº total de bits para etiquetas
Corresp. directa	4096 (2^{12})	$30 - 12 = 18$	$4096 \times 18 = 72 \text{ Kb}$
2-asociativa	2048 (2^{11})	$30 - 11 = 19$	$4096 \times 19 = 76 \text{ Kb}$
4-asociativa	1024 (2^{10})	$30 - 10 = 20$	$4096 \times 20 = 80 \text{ Kb}$
Total. asociativa	1	30	$4096 \times 30 = 120 \text{ Kb}$

Subsistema de memoria en sistemas empotrados. (3+1 horas)

3.1 Espacio de memoria

3.2 Memoria Cache

3.3 Memoria principal

3.4 Memoria auxiliar o de almacenamiento

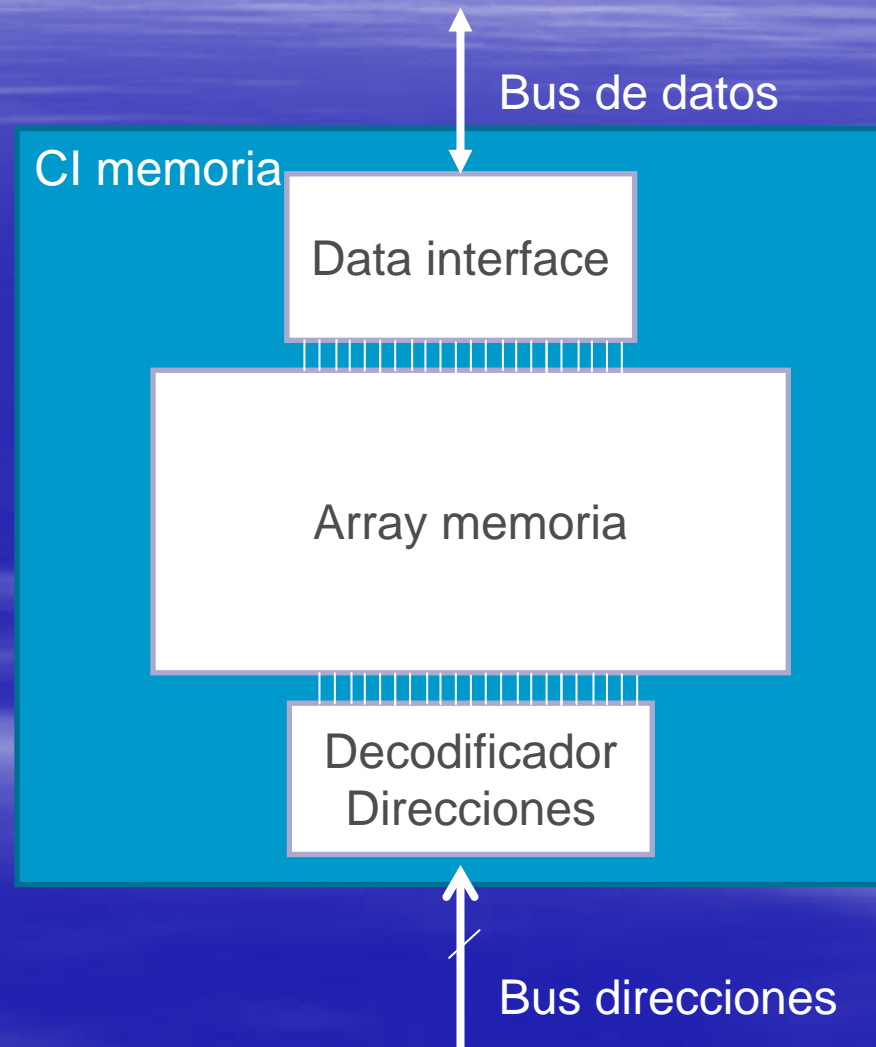
3.5 Acceso directo a memoria

3.6 Rendimiento y memoria

3.3 Memoria Principal

Formada por tres componentes

- El CI de memoria
- El bus de direcciones
- El bus de datos
- El CI de memoria está formado por
 - El array bidimensional de celdas de memoria (memoria física)
 - El decodificador de direcciones
 - El interfaz de datos



3.3 Memoria Principal

- El CI de memoria se conecta a la tarjeta empotrada a través de una variedad de empaquetamientos dependiendo del tipo de memoria:
 - DIP Dual inline packages (un CI)
 - SIMM Single in –line memory modules
 - DIMM Dual in-line memory modules

Los SIMM y los DIMM son minimódulos (PCBs) que contienen varios CIs de memoria
El número de pines puede variar (30, 72, 168, etc.)

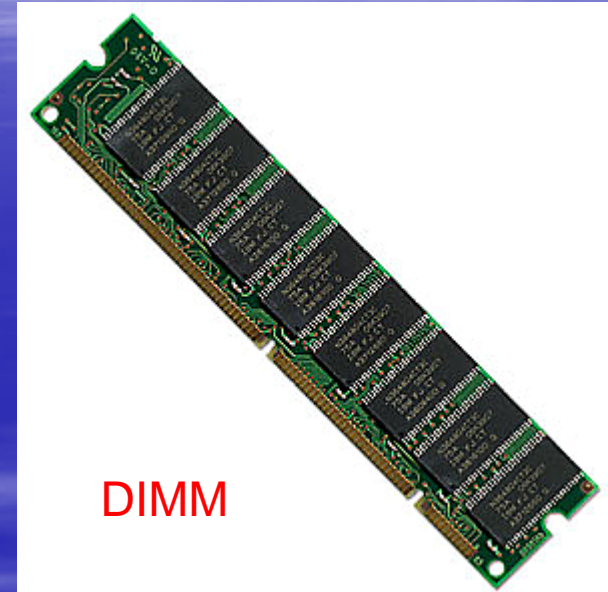
3.3 Memoria principal



DIP



SIMM



DIMM



Zócalo tipo SIMM

Memoria Primaria

- La memoria primaria y secundaria puede dividirse en 2 grupos:
 - Memoria no volátil
 - ROM, Flash
 - NVRAM (Non Volatil RAM)
 - Memoria auxiliar
 - Memoria volátil
 - RAM (acceso aleatorio)

NVRAM es una RAM CMOS de bajo consumo conectada a una batería de litio + un controlador, o bien una RAM CMOS + EEPROM que se actualiza con un pulse de retención (cuando se va a apagar la alimentación)

Memoria Sólo lectura (ROM)

- Se usa para guardar datos de forma permanente (software de arranque)
- Tipos:
 - **Mask ROM (MROM)** Los bits de datos se programan de forma permanente en un microchip en la fábrica. Suelen ser de tecnología MOS o bipolar
 - **One-time Programmable ROM (OTP ROM)** Puede programarse fuera de fábrica mediante un programador de ROM. Utiliza tecnología bipolar. Es como una EPROM que no se puede borrar
 - **Erasable Programmable ROM (EPROM)** puede borrarse varias veces usando onda corta o luz ultravioleta. Tecnología MOS + puerta flotante
 - **Electrically Erasable Programmable ROM (EEPROM)** Se pueden borrar sin usar dispositivos especiales (desde el dispositivo empotrado). Se escribe y borra en bytes. Usan tecnología NMOS + puerta flotante
 - **Flash** variación de la EEPROM que puede borrarse y escribirse en bloques o sectores (grupos de 64Kbytes). Por tanto puede borrarse eléctricamente en el propio dispositivo empotrado. Usa tecnología CMOs
Suelen usarse para reemplazar discos en PDAs, cámaras, teléfonos, etc).

Flash

- Puede utilizarse para:
 - Almacenar el sistema de arranque
 - Almacenar el Sistema Operativo
 - Almacenar aplicaciones e imágenes de librerías
 - Almacenar ficheros

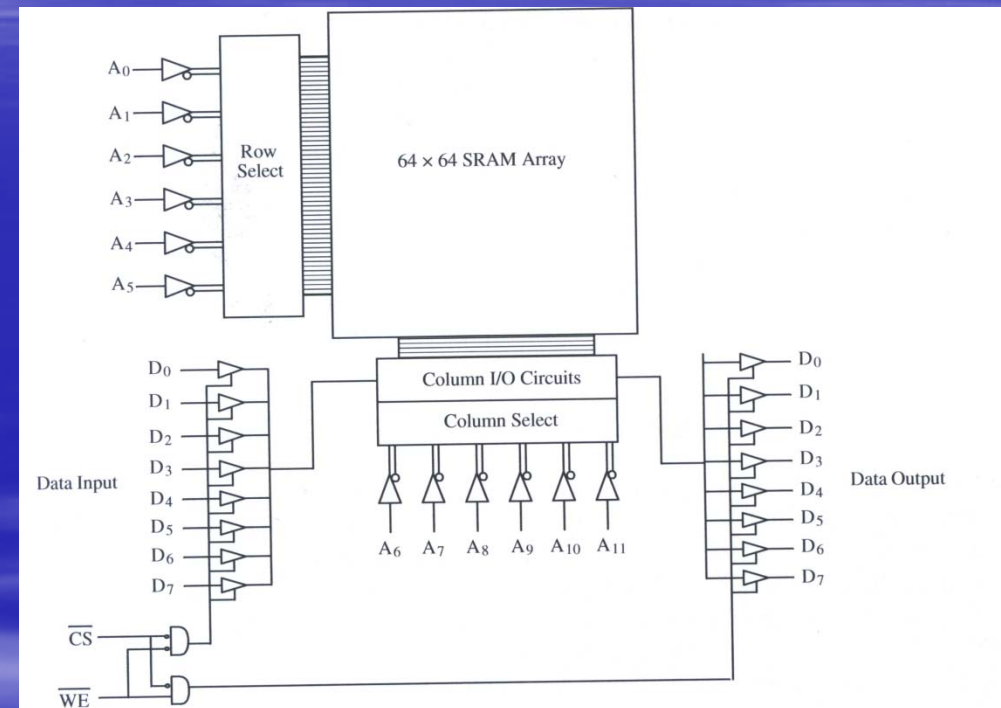
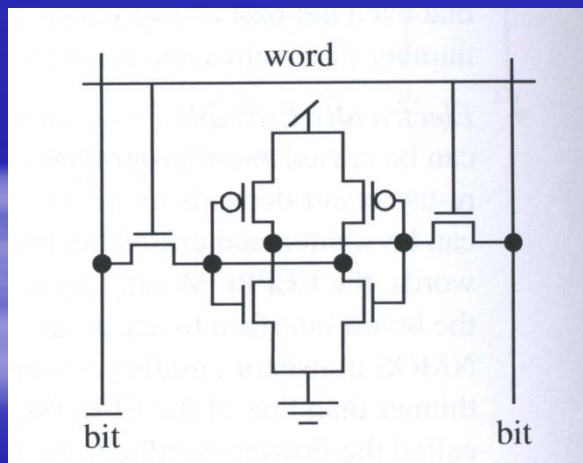
Se diferencian tecnológicamente de los discos en que los sectores son de **64kbytes** en lugar de 512bytes y que un sector hay que borrarlo antes de escribirlo. El tiempo de vida viene limitado por el número de veces que un sector se borra.

Hay 2 tipos de dispositivos:

- 1.- Los **chips de flash** (tecnología NAND o NOR) que suelen llevar un "Common flash interface"
- 2.- Los **Discos flash** para almacenamiento masivo que pueden ser basados en ATA (interfaz de disco estándar IDE) y Flash Lineales (basados en NAND pero con capacidad de arranque gracias a una boot ROM que se reconoce como extensión de la BIOS)

Memoria de acceso aleatorio (RAM)

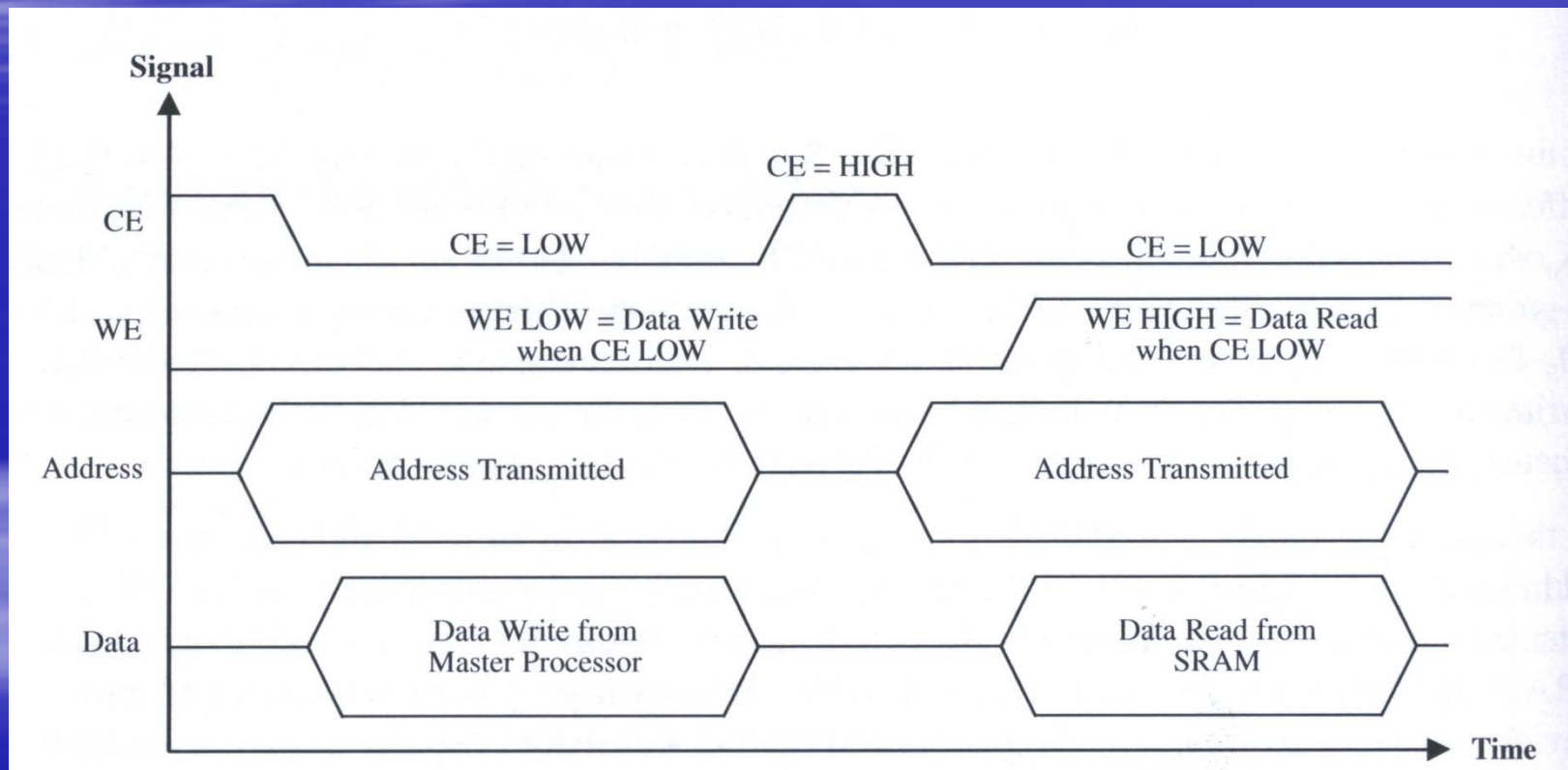
Normalmente es la llamada **Memoria Principal**. Es volátil. Puede ser estática (SRAM) o dinámica (DRAM)



SRAM

Usada en pequeñas cantidades (Cache)

Diagrama de tiempos SRAM



Memoria DRAM

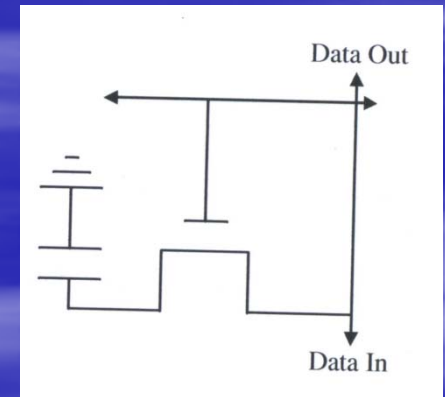
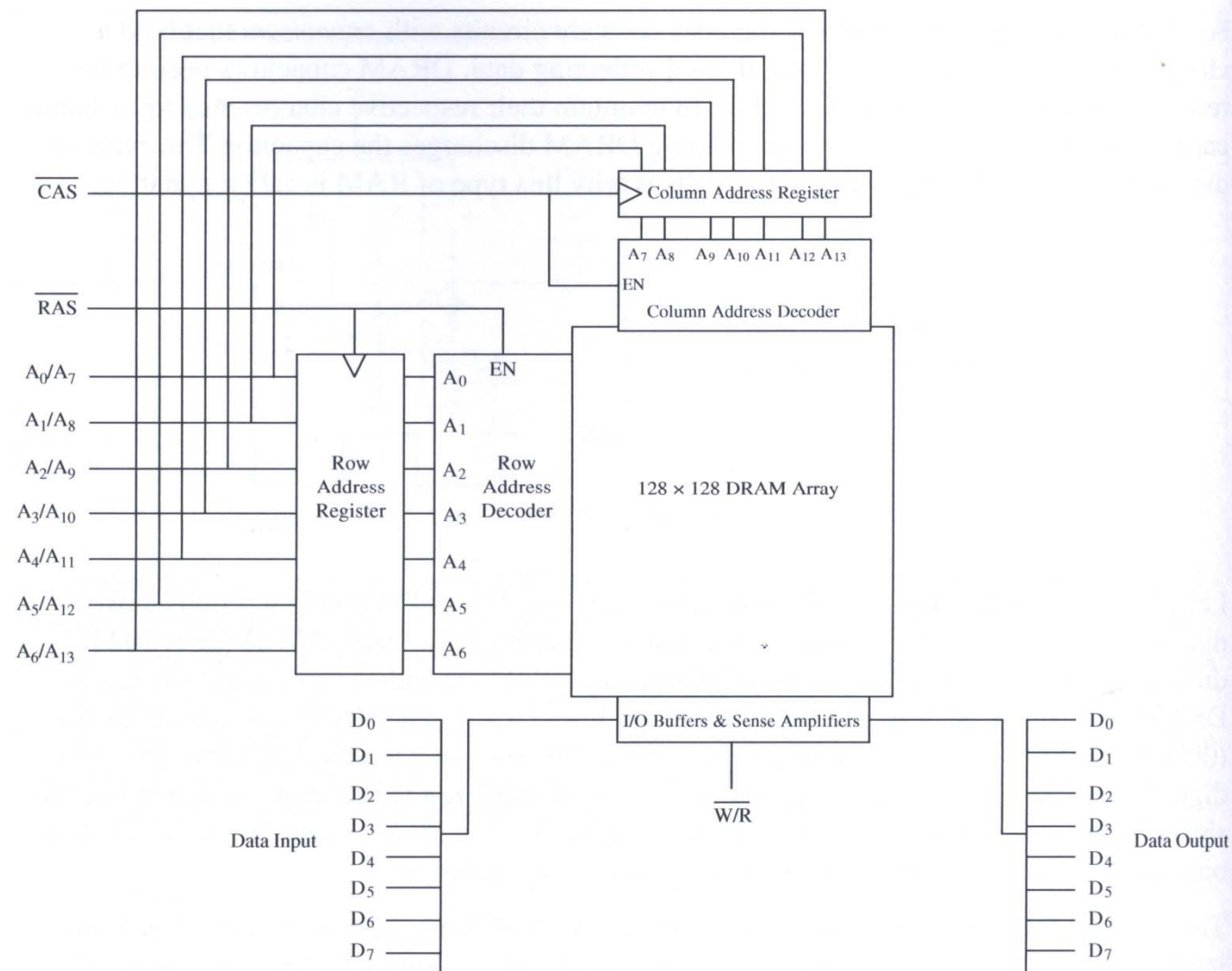


Diagrama de tiempos DRAM (L)

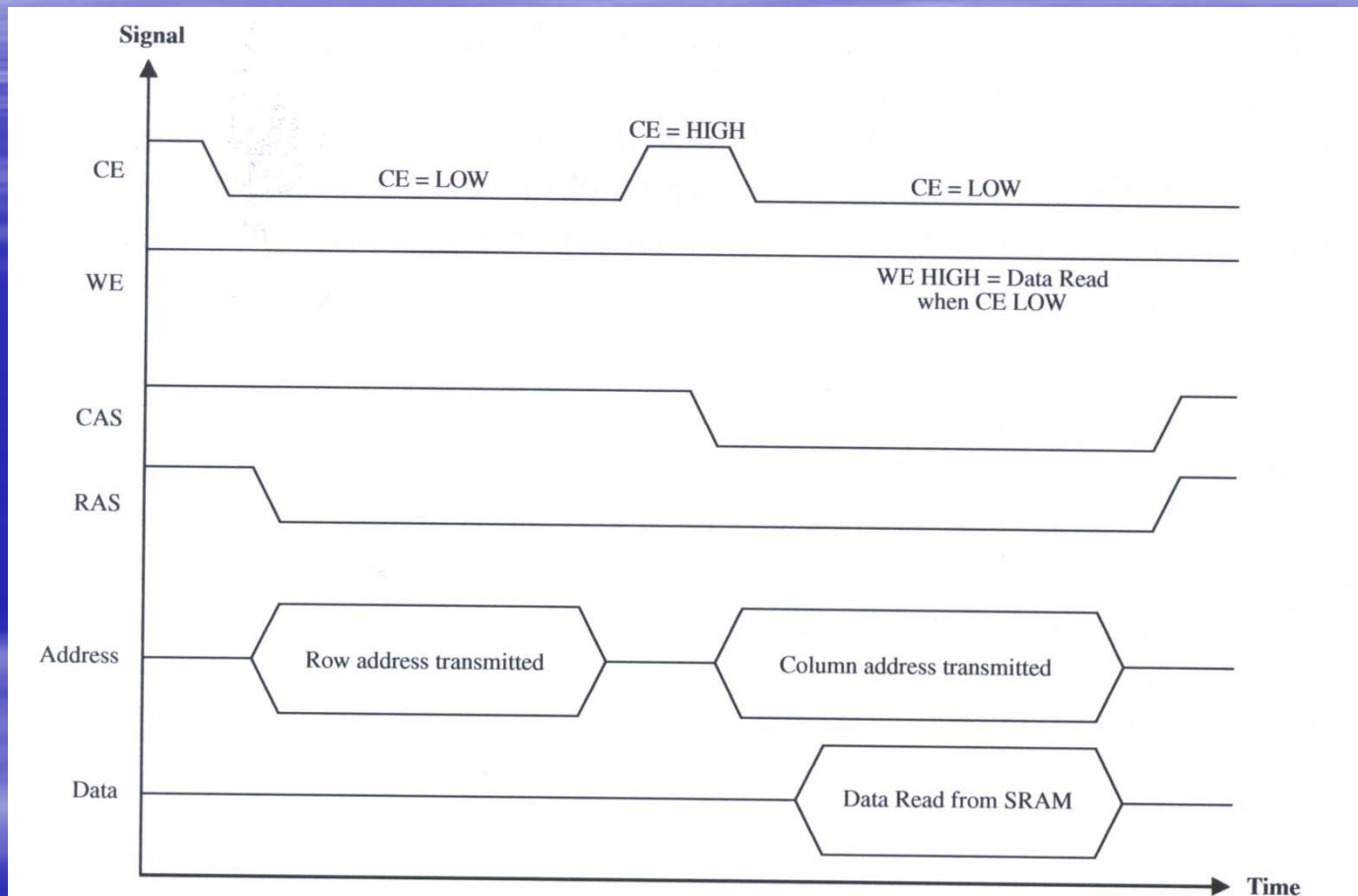
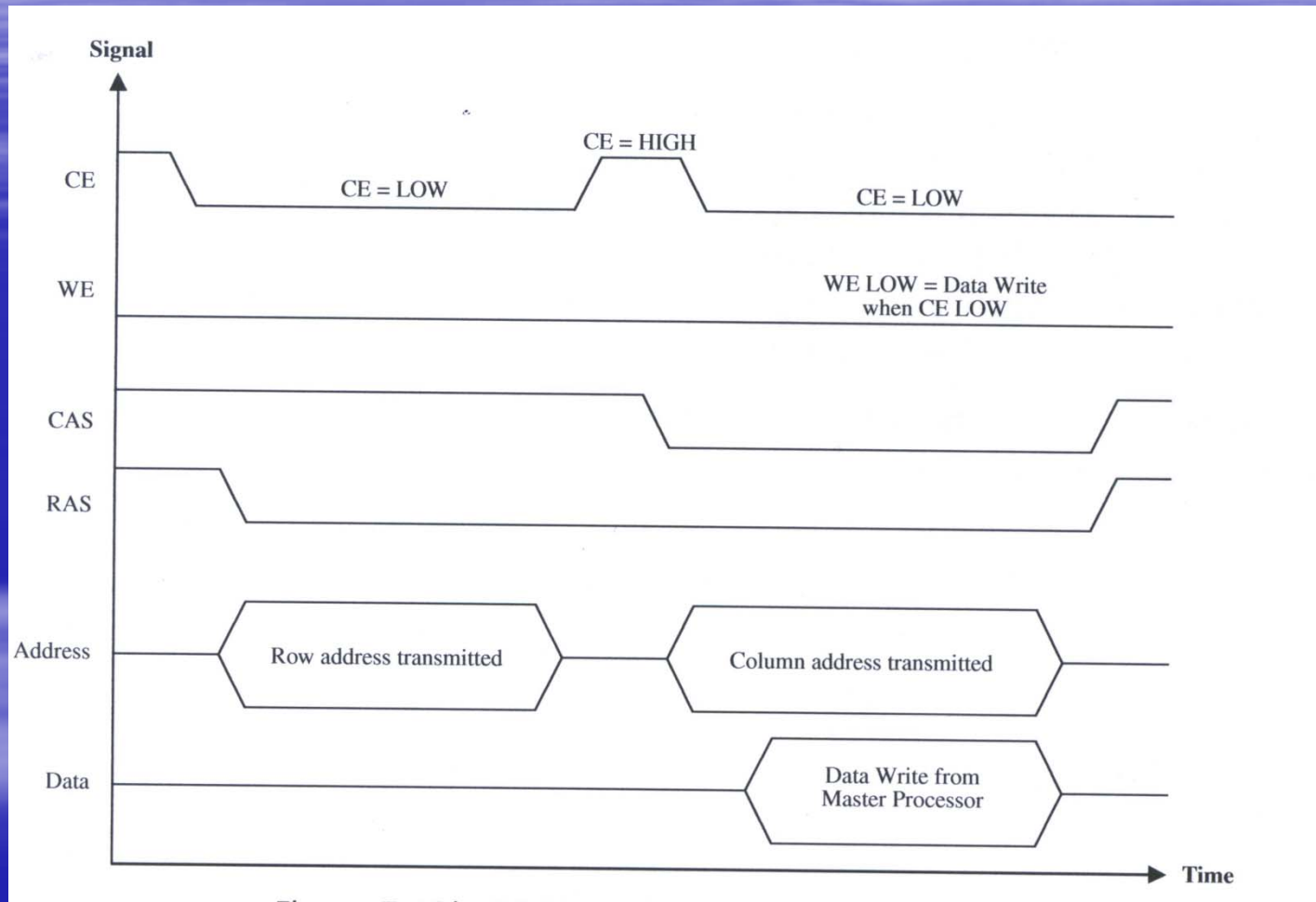


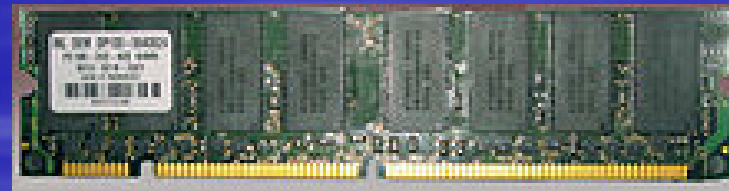
Diagrama de tiempos DRAM (E)



Hay un controlador encargado del refresco
Es más lenta y más barata. Se usa en grandes cantidades

Memoria Síncrona

- Es la memoria externa de mayor rendimiento. Son baratas pq la industria de computadores personales usa esta memoria en módulos DIMM estándar.
- Hay dos tipos:
 - SDRAM
 - DDR-SDRAM



SDRAM

- Las lecturas y las escrituras se sincronizan con el reloj del procesador.
- Necesita refresco.
- Frecuencias 100MHz-133MHz (532 Mbytes/s)

Datos
Dir
DM(1:0)
CLK
SCKE
RAS
CAS
WE

Latencia de CAS es el retardo en ciclos de reloj desde que se da un comando de lectura hasta que el dato está listo. Suele ser de 2-3 ciclos

El proceso de preparar la SDRAM (RAS) puede llevar varios ciclos de reloj, pero una vez la fila está activa se puede leer completa a una dato por ciclo (**burst**)

Double Data Rate (DDR)

- Permite acceso síncrono en ambos flancos de reloj, por lo cual proporciona doble ancho de banda. Requiere su propio controlador.
- A 400MHz incrementa el ancho de banda a 3.2 Gbytes/s (2.5v)
- DDR2 ofrece hasta 6.4 Gbytes/s a 1.8v

Subsistema de memoria en sistemas empuotrados. (3+1 horas)

3.1 Espacio de memoria

3.2 Memoria Cache

3.3 Memoria principal

3.4 Memoria auxiliar o de almacenamiento

3.5 Acceso directo a memoria

3.6 Rendimiento y memoria

3.4 Memoria auxiliar

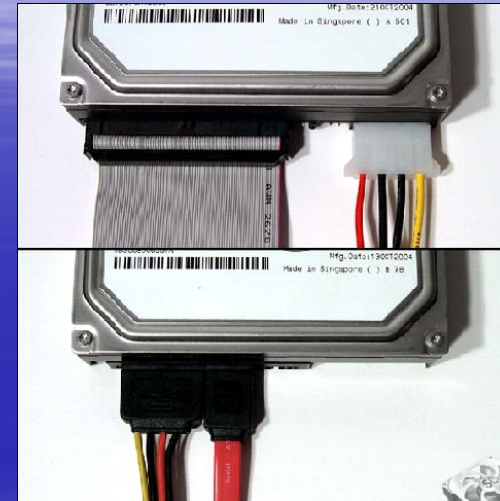
- Sirve para almacenar grandes cantidades de datos durante grandes periodos o indefinidamente
 - **Cintas magnéticas** (acceso secuencial)
 - **Discos duros** formados por varios platos o discos de metal cubiertos de un material magnético. Pueden tener múltiples cabezas soportadas por un brazo. En cada pista los datos se leen secuencialmente
 - **Compact Disc**: no son magnéticos sino ópticos y se leen con láser. Son de 1 escritura salvo que lleven material óptico y magnético (se leen y escriben con láseres manipulados y campos magnéticos).

Memoria auxiliar

- Discos IDE (Integrated Drive Electronics)
 - El controlador de disco se integra directamente en el propio disco, en lugar de en la tarjeta empotrada
 - El interfaz ATA especifica el modo en que los comandos se pasan al controlador del disco duro, se interpretan y procesan.
 - Método de acceso:
 - CHS Cilindro-Head-Sector
 - LBA Logical Block Addressing (Cada sector tiene un número de identificación)
 - Modos de acceso:
 - **E/S programada**: el procesador planifica, arranca y completa un ciclo de proceso de datos
 - **DMA** Transferencia disco-memoria sin intervención del procesador

Memoria auxiliar

- Otros interfaces de disco duro:
 - **SATA**: Serial ATA serializa el interfaz de datos paralelo del ATA. Utiliza señalización diferencial a velocidades de varios cientos de MB/s
 - **SCSI** (Small Computer System Interface) Mejor rendimiento que el ATA pero más complejo y caro. Permite la expansión a mayor número de dispositivos.
 - **Microdrive**: es un disco duro en miniatura de tecnología magnética.
 - **USB/Firewire** son discos ATA paralelos con un terminal USB 2.0 o Firewire (IEEE 1394)



Subsistema de memoria en sistemas empuotrados. (3+1 horas)

- 3.1 Espacio de memoria
- 3.2 Memoria Cache
- 3.3 Memoria principal
- 3.4 Memoria auxiliar o de almacenamiento
- 3.5 Acceso directo a memoria**
- 3.6 Rendimiento y memoria

3.5 Acceso Directo a memoria

- Son **controladores** que se encargan de realizar las transferencias entre la memoria externa/interna y los periféricos o almacenamiento auxiliar sin intervención del procesador.
- Éste se encarga de configurar el controlador de DMA durante la **inicialización**.
- Al terminar, el controlador produce una **interrupción**.
- Puede realizar transferencias entre los distintos tipos de memoria
- Periféricos que necesitan DMA: video, audio, interfaces de red

3.5 Acceso Directo a Memoria

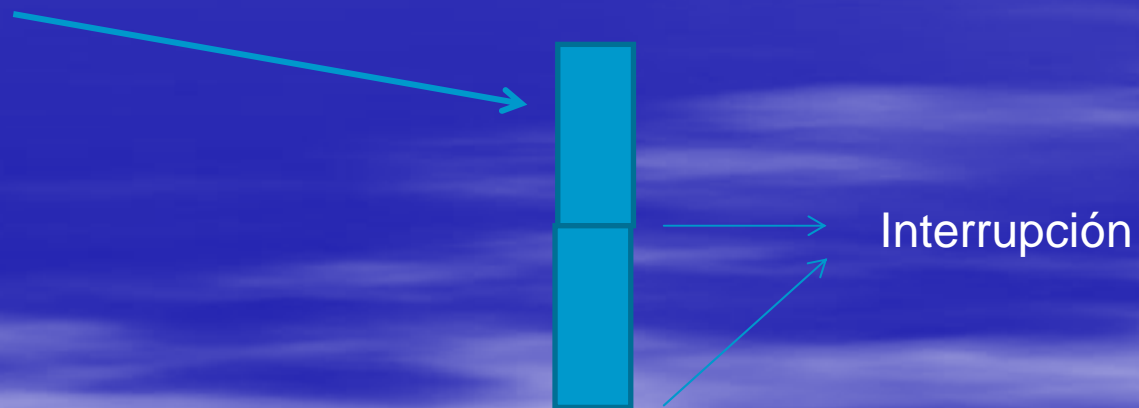
- Cada controlador de DMA puede tener varios canales con una prioridad programable. Arbitraje por prioridades
- Existen un conjunto de FIFOs que actúan de buffer
- Una transferencia DMA implica
 - Fuente
 - Destino
 - N° bloques a transferir X_count
 - N° bytes para incrementar el puntero de dirección después de mover un elemento X_Modify
 - Para transferencias en 2D (con lazos anidados) Y_Count, Y_Modify
- Dos tipos de configuraciones: **modo registro** y **modo descriptor**

3.5 Acceso Directo a Memoria

- **Modo registro** Existen 2 submodos: autobuffer y stop.
 - **Autobuffer**: *Flujo de datos continuo*
cuando se completa la transferencia de un bloque el registro de control automáticamente se carga con los valores originales y el proceso comienza de nuevo.
 - **Stop**: cuando termina se **para** y espera a la siguiente transferencia

Acceso Directo a Memoria

- Ej. Una aplicación opera con 512 muestras de audio, y el codec envía los nuevos datos a la frecuencia de audio. Si se dispone de un doble-buffer para que mientras la DMA opera con uno, el procesador opera con el otro, qué técnica es la más apropiada: el autobuffer
 - Xcount=512
 - Ycount=2
 - Xmodify=4
 - Ymodify=1



3.5 Acceso Directo a Memoria

- **Modo descriptor** Permite encadenar varias secuencias de DMA. Hay 2 submodos: array y lista
- **Modo Array**

Start_Addr [15:0]
Start_Addr[31:16]
DMA_Config
X_Count
X_Modify
Y_Count
Y_Modify

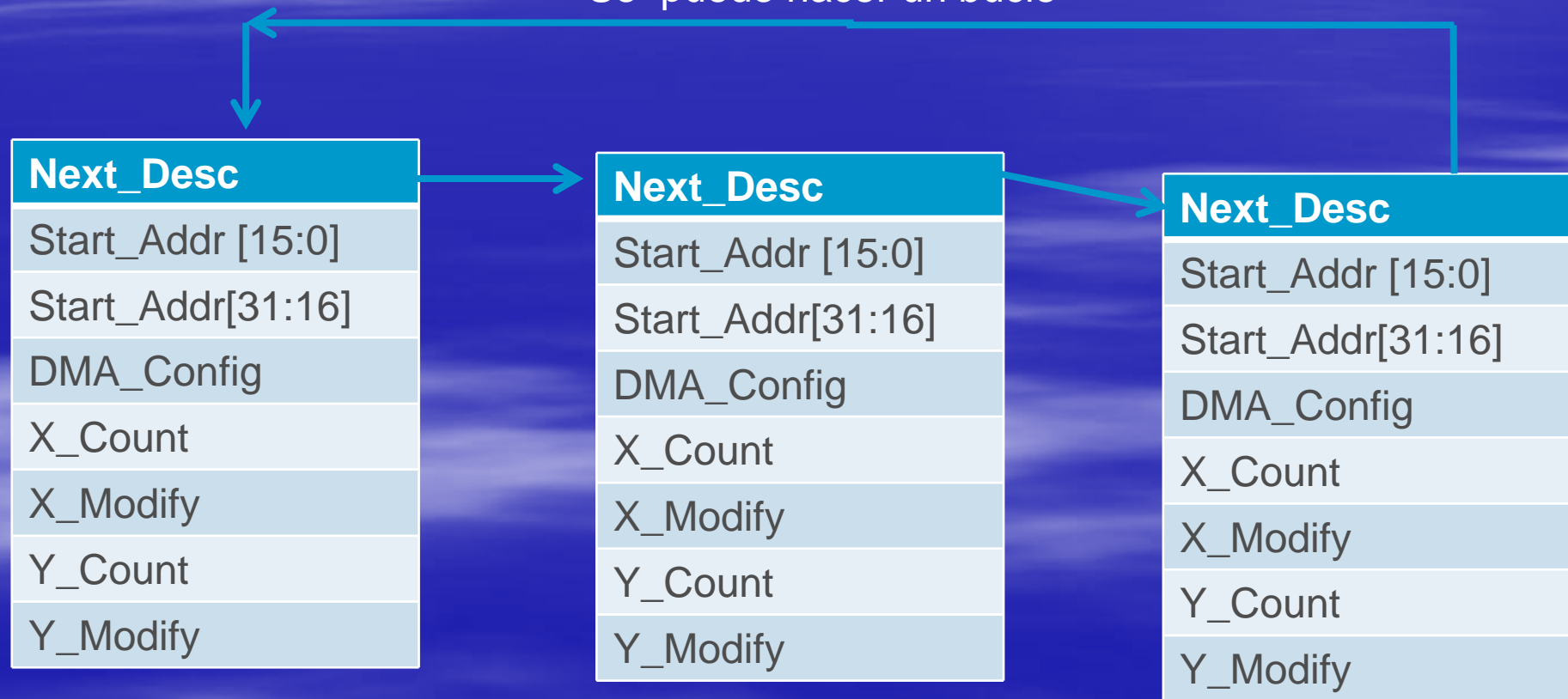
Descriptor bloque 1

Descriptor bloque 2

Acceso Directo a Memoria

■ Modo lista

Se puede hacer un bucle



Subsistema de memoria en sistemas empuotrados. (3+1 horas)

- 3.1 Espacio de memoria
- 3.2 Memoria Cache
- 3.3 Memoria principal
- 3.4 Memoria auxiliar o de almacenamiento
- 3.5 Acceso directo a memoria
- 3.6 Rendimiento y memoria**

3.6 Rendimiento y memoria

- Una de las medidas más comunes de rendimiento del procesador es el ancho de banda (**throughput**). Este puede verse negativamente afectado por la memoria principal si la DRAM utilizada es más lenta que el procesador.
- Soluciones para mejorar el rendimiento:
 - Arquitectura tipo Harvard
 - Usar DRAMs que utilicen señales de bus integradas, para disminuir el tiempo de arbitraje
 - Utilizar una jerarquía de memoria (caches)
 - Utilizar DMA