

TEMA 4 : LA MEMORIA

• *Introducción.*

Este tema versará sobre la memoria y los distintos tipos existentes. Antes de empezar debemos conocer qué representa en la Estructura de Computadores la memoria:

- 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 en el propio computador y otros están localizados externamente.

• *Definiciones y conceptos.*

A continuación veremos con detenimiento las definiciones y los conceptos necesarios para la mejor comprensión del tema que nos ocupa.

2.1. 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) que implica memoria estática y dinámica.
- Mecanismo de direccionamiento. Deberá disponer de un procedimiento para L/E de información en el lugar y tiempo deseado.

2.2. Características de las memorias.

Las características de las memorias que veremos a continuación son:

-
- Localización.
- Capacidad.
- Unidad de transferencia.
- Método de acceso.
- Velocidad.
- Dispositivo físico.
- Aspectos físicos
- Organización.

2.2.1. Localización.

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

- Memoria interna del procesador. Memoria de alta velocidad utilizada de forma temporal, por ejemplo, *en el banco de registros*. Es muy rápida.
- Memoria interna. Es la Memoria Principal y es más rápida que la secundaria, por ejemplo, *se ubica donde están los programas para ser ejecutados*.
- Memoria externa. Es la Memoria Secundaria y es más lenta que la principal; se emplea para almacenar grandes cantidades de información, por ejemplo, *unidades zip, CD's y DVD's*.

2.2.2. Capacidad.

La capacidad es la cantidad de información que puede almacenar el sistema de memoria y se mide en múltiplos de bit.

IMPORTANTE

- **Nota 1, sobre Byte:** 1B significa 1 byte.
- 1K x 8b equivale a 8K bits ! memoria de 1K posiciones por 8 bits de ancho de palabra.

Los distintos múltiplos y unidades de capacidad son los siguientes:

1 bit 1 Mb = 1024 Kb = 220 bits

1 nibble = 4 bits 1 Gb = 1024 Mb = 230 bits

1 byte = 1 octeto = 8 bits 1 Tb = 1024 Gb = 240 bits

1 Kb = 1024 bits = 210 bits

2.2.3. Unidad de transferencia.

Es igual al número de líneas de datos de entrada y salida del módulo de memoria, por ejemplo, *8 en MaNoTaS*. Esta descripción lleva consigo una serie de 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. Depende del número de líneas con el que esté trabajando en la memoria, por ejemplo, *8 en MaNoTaS*.
- Unidad direccionable. Es el tamaño mínimo que podemos direccionar en la memoria. El tamaño puede coincidir con el ancho de palabra como, por ejemplo, *en la MaNoTaS con 8 bits*, pero también puede NO coincidir como ocurre, por ejemplo, *en el 8086*.
- Unidad de transferencia. Para la memoria principal es el número de bits que se leen o escriben en memoria a la vez, lo máximo que puedo leer. Por ejemplo, *en MaNoTaS es de 8 bits*.

2.2.4. Método acceso.

Es la forma de localizar la información en memoria. Existen diversos tipos:

- Acceso secuencial (SAM: Sequential Access Memory). Se emplea por ser de gran capacidad de almacenamiento, por ejemplo, *cintas de audio*. Es una memoria lenta.
- Acceso directo (DAM: Direct Access Memory). Es una memoria en la que primero se produce un acceso directo y luego uno secuencial.

- Acceso aleatorio (RAM: Random Access Memory). Igual tiempo de acceso menos directo siempre.
- Acceso asociativo (CAM: Content Addressable Memory). Modo de acceso por contenido y funciona de la siguiente forma: se busca en toda la memoria al mismo tiempo y ,cuando se encuentra lo que buscamos, se da la dirección donde se ha encontrado como, por ejemplo, *en redes neuronales*.

2.2.5. Velocidad.

Para medir el rendimiento se utilizan tres parámetros:

- Tiempo de Acceso (TA). Según el tipo de memoria que se trate tendremos dos tipos de TA:
 - Si se trata de una aleatoria como la 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 (es decir, desde que doy la dirección hasta que L/E en memoria).
 - Si es otra memoria como la CAM o la SAM: tiempo que se emplea en situar el mecanismo de L/E en la posición deseada, es decir, tiempo que tarda en el registro.
- Tiempo de Ciclo de memoria (TC). Tiempo que transcurre desde que se da la orden de una operación de L/E hasta que se puede dar otra orden de L/E porque sino NO le da tiempo a responder y crearía un tiempo muerto que nos interesa que sea lo menor posible.

Gráficamente el TA y el TC se pueden representar cómo funcionan a través del tiempo de la siguiente figura:

TC

tiempo

TA

petición lectura información disponible próxima petición

- Velocidad de Transferencia (VT). Es la velocidad a la que se pueden transferir datos a, o desde, una unidad de memoria. Según el tipo de memoria existen dos casos de velocidad:
 - En el caso de acceso aleatorio : $VT = 1 / TC$
 - En el caso de acceso de no aleatorio:

$TN = TA + (N / VT)$, con la siguiente nomenclatura:

- TN : Tiempo medio de L/E de N bits, es decir, tiempo de disponibilidad de datos
- TA : Tiempo de acceso.
- N: número de bits.
- VT : Velocidad de transferencia (bits / segundo).

2.2.5. Dispositivo físico.

Los sistemas de memorias empleados en los computadores utilizan diferentes dispositivos físicos. Antiguamente se empleaba la memoria de ferrita que era una memoria de lectura destructiva y muy lenta ya que tenían que volver a escribir lo que se leía, sino serían muy rápidas.

Actualmente los tipos más usados son:

- Para la memoria principal se utilizan memorias semiconductoras.
- Como memoria secundaria, ya que responden a la necesidad de almacenar grandes cantidades de información, se emplean:
 - Memorias magnéticas, cintas, discos, etc.
 - Memorias ópticas, utilizadas.
 - Memorias magneto-ópticas.

2.2.6. Aspectos físicos.

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

- Alterabilidad. Esta propiedad hace referencia a la posibilidad de alterar el contenido de una memoria, las hay de lectura sólo o de lectura / escritura. Memorias ROM (Read Only Memory) y RWM (Read Writable Memory).
- Permanencia de la información. Relacionado con la duración de la información almacenada en memoria:
- Lectura destructiva. Memorias de lectura destructiva DRO (Destructive Read Out) y memorias de lectura no destructiva NDRO (Non Destructive Read Out).
- 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 SRAM (Static Random Access Memory) si la información que contiene no varía con el tiempo. Una memoria es dinámica DRAM (Dynamic Random Access Memory) si la información almacenada se va perdiendo conforme transcurre el tiempo; para que no se pierda el contenido habrá que recargar o refrescar la información.

2.2.7. 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 21/2D.
- Organización 3D.
- **Organización 2D.**

RAM de 2^m palabras de ancho de n bits cada una, la matriz de celdas está formada por 2^m filas (número de posiciones de la memoria interna que van de la cero a la 2^m-1 en el bus de direcciones) y n columnas (número de bits en el registro de la memoria o ancho de palabra). Se trata de una organización lineal que se emplea en memorias de poca capacidad y gran rapidez de acceso

n

Registro de datos a escribir

..... E

Decodificador

Registro de

direcciones

..... 2 m

palabras

.....

..... L

CS Registro de datos leídos

n

Si tuviéramos, por ejemplo, $1K \times 8$ de memoria tendríamos 1024 posiciones que son 210 y va de la 0 ... 1023, o bien, de la 0000h ... 1FFFFh. Si aumento el ancho de palabra a más de 8 bits entonces ampliaría la superficie de silicio de la memoria.

• Organización 21/2D.

Utiliza dos decodificadores con $m/2$ entradas y $2m/2$ salidas. El bus de direcciones de la organización 2D se divide en dos buses encaminados a dos decodificadores que van de 0 ... $[(m/2) - 1]$ y de $[m/2] \dots m - 1$ y donde coincidan las líneas de salidas de los decodificadores estará el bit que busco, por ejemplo, *bus de direcciones de 16 pasa a dos de 8 bits que implican 28 (256) posiciones*. Esta organización se emplea en memorias de un sólo bit como unidad de transferencia pues sólo coincide un bit entre los dos decodificadores y precisa menor tamaño para la memoria que la de organización de tipo 2D por lo que requiere menor número de puertas lógicas.

Registro de direcciones que van a decodificadores

(se divide en dos que suman $m = X + Y$ entradas)

Decodificador Y

Y

CS Registro

datos a

escribir

Decodificador

X

X

n

.....

E

L

.....

Registro de datos leídos

n

• **Organización 3D.**

Es similar a la organización 2D 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, por lo que sólo se habilitarán los bits que estén en el mismo plano; por ejemplo, *si tenemos 16 bits tendremos 16 planos*. Tiene un diseño más complicado y es una organización para memorias de acceso lento. Son chips de mayor densidad de silicio y, por lo tanto, más anchos para así poder ampliar el ancho de palabra más fácilmente.

2.3. Jerarquía de las memorias.

Una serie de parámetros fundamentales caracterizan los tipos de memorias del computador:

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

La configuración ideal de una memoria sería: memoria rápida, de 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. Ésta podría representarse en la siguiente pirámide:

+

Coste

Memoria

Velocidad Interna

Nº accesos

—

—

Memoria

+

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 la localidad de referencia depende de la frecuencia de accesos.

- **Memoria principal semiconductora.**

A continuación veremos todos los tipos de memoria semiconductora (todas las RAM) y sus correspondientes diseños.

3.1. Tipos de memorias.

Las memorias de semiconductores son todas RAM y el método de borrado determina a la memoria.

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 suelen emplearlas cuando producen componentes de forma masiva.
 - **PROM (Programmable Read Only Memory).** El proceso de escritura se lleva a cabo eléctricamente y puede realizarlo el suministrador o el cliente con posteridad 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.
- Memorias de sobre todo lectura (Read-Mostly Memory):
 - **EPROM (Erasable Programmable 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 (Electrically Erasable Programmable Read Only Memory).** Se borra mediante corriente eléctrica de forma selectiva a nivel de byte. Es más cara que la 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.
- Memorias de L/E: RAM (Random Access Memory). Al igual que las anteriores son de acceso aleatorio. Los principales tipos de RAM son:
 - **DRAM (Dynamic Random Access Memory).** Los datos se almacenan de forma similar a la carga de un condensador. Debido a que tiende a descargarse es necesario refrescarlas periódicamente; este proceso de refresco se realiza de la siguiente forma: cada cierto tiempo se borra y se reescribe la información mientras que no se produzca una orden de lectura y escritura. Son más simples y más baratas que las memorias SRAM.
 - **SRAM (Static Random Access Memory).** Los datos se almacenan formando biestables, por lo que

NO necesita refresco. Son más rápidas que las DRAM y más caras.

A continuación tenemos una tabla resumen:

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

3.2. Diseño.

En el diseño de la memoria hay que tener en cuenta dos aspectos importantes:

-
- Chip de memoria.
- Mapa de memoria.

3.2.1. Chip de memoria.

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

Un ejemplo:

Memoria de $4K \times 8$! $4K$ palabras cuyo ancho de palabra es de 8 bits. La memoria tiene en total:

- $4K = 4 \cdot 2^{10} = 4 \cdot 1024 = 4096$ palabras = n palabras que son 4096 direcciones o posiciones de memoria y
- 8 bits = m bits de ancho de palabra.

Gráficamente el chip de memoria, tanto el chip de la memoria del ejemplo como cualquier otro chip, es como se muestra en la figura:

Contenido de la

palabra 0 de

Direcciones Contenido memoria la memoria

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
...	...	...	...	...	...	...	...

m bits

Memoria de 4K x 8

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 un ancho de palabra de 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 la lectura OE (Output Enable).
- CS (Chip Selection) o CE (Chip Enable). Selecciona el chip de memoria al cual voy a acceder.
- V_{CC}. Alimentación del chip.
- V_{SS}. Conexión a tierra.

La pastilla tiene la siguiente configuración:

 V_{CC} V_{SS}

n

BUS DATOS

BUS DIRECCIONES

m

CS R/W

Para el correcto funcionamiento de la memoria, que se rige por una tabla de verdad, es necesario incorporar una circuitería adicional como son decodificadores, multiplexores, buffers, etc. En las siguientes figuras se muestran estos bloques combinatoriales así como la tabla de verdad de la memoria:

Entradas			Salida
CE	WE	OE	Chip
H	X	X	Z
L	H	L	Lectura
L	L	X	Escritura
L	H	H	Power Down / Z

TABLA DE VERDAD

CE

_____

WE

OE

.....

A12

A11 D7

.....

..... D0

A5

A4

.....

....

A0

CIRCUITO DE LA MEMORIA

Para implementarla debemos saber:

- ¿Cuántas líneas necesito para el bus de direcciones? Determina la capacidad total de direccionamiento del microprocesador y son m líneas y que tendré las posiciones de memoria desde 0 hasta $2^m - 1$ codificadas en hexadecimal.
- ¿Cuántas líneas necesito para el bus de datos? Determina la capacidad para leer o escribir datos y es n líneas, lo que implica que puedo leer o escribir 2^n bits y que se hará un número de veces. ____
- ¿Cuántas líneas necesito para realizar la L/E? R/W es una línea única normalmente por lo que con un 0 se escribirá y con un 1 se leerá. La memoria puede funcionar al revés o incluso que exista una línea para escribir y otra independiente para leer.

3.2.2. Mapa de memoria.

Es el espacio que puede direccionar un computador. Con el mapa de memoria se indica al procesador dónde empieza y dónde acaba la memoria así como el tipo de memoria empleado. Para saberlo en la memoria he de saber por el chip:

- ¿Cuántas líneas necesito para el bus de direcciones? Determina la capacidad total de direccionamiento del microprocesador y son m líneas y que tendré las posiciones de memoria desde 0 hasta $2^m - 1$ codificadas en hexadecimal.

Un ejemplo:

Computador con memoria de 64K x 8 ! 64K palabras cuyo ancho de palabra es de 8 bits. La memoria tiene en total:

- $64K = 64 \cdot 2^{10} = 2^6 \cdot 2^{10} = 2^{16}$ palabras ! $n = 16$ bits para el bus de direcciones.
- $2n = 216$ posiciones de memoria, codificadas en hexadecimal, que van desde la 0 hasta la $216 - 1$ bits ,o bien en hexadecimal desde la 0000h hasta la FFFFh.

La implementación física 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, o bien, 1024 posiciones de memoria cada una con una longitud o ancho de palabra de 8 bits.

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. Así, por ejemplo, supongamos que queremos diseñar una memoria de 64Kbytes ($n = 8$) y sólo 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 tal y como se muestra en la figura siguiente:

BUS

DIRECC.

CS

—

WE

Si quisiéramos una capacidad de cK palabras y dispusiéramos de chips de zK, necesitaremos c / z chips para conseguir la capacidad deseada. Así, por ejemplo, supongamos que queremos diseñar una memoria con 64Kbytes ($c = 64$) y sólo disponemos de chips de 32Kx8 ($z = 8$), entonces necesitaremos 2 chips ($64 / 32$). Cuando la línea A15 está a 1 habilita el chip superior, mientras que cuando está a 0 habilita el chip inferior. En esta interconexión veremos que hay 2 filas y 1 columna de chips tal y como se muestra en la figura siguiente:

BUS

DATOS

BUS DIRECC. 16

—

WE

3.3. Ejemplos.

Ahora vamos a ver distintos ejemplos de diseño de memorias.

3.3.1. Ejemplo1.

Si quisiéramos diseñar una memoria principal de 128K palabras:

- *¿Cuántos chips de memoria de 32Kx8 necesitaremos si suponemos que la palabra es de 8bits?*
- *¿Cuántos chips de memoria de 64Kx4 necesitaremos si suponemos que la palabra es de 8bits?*

Solución:

- *¿Cuántos chips de memoria de 32Kx8 necesitaremos si suponemos que la palabra es de 8bits?*

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 entonces NO necesitaremos más.

- *¿Cuántos chips de memoria de 64Kx4 necesitaremos si suponemos que la palabra es de 8bits?*

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

3.3.2. Ejemplo2.

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

- *Debemos obtener el número de bits del bus de direcciones.*
- *Averiguar el número de bits que se necesitan para direccionar el chip de memoria que vamos a emplear.*
- *Calcular el número de chips que necesitamos.*
- *Obtener el número de bits del bus de direcciones que permita seleccionar los chips de memoria.*
- *Dibujar el diagrama de conexiones de la memoria junto con la lógica de selección.*

Solución:

- *Debemos obtener el número de bits del bus de direcciones.*

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

- *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, A15 A14 A13 ... A1 A0. Hay que hacer notar que el resto de bits se quedan sin utilizar de momento ya que pueden ser utilizados más tarde para sucesivas ampliaciones.

- *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 faltan otros 64Kx16, es decir, 16 chips más. Por tanto necesitaremos 32 chips de 64Kx16 para almacenar 128Kx16.

- *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 A16 para seleccionar los chips de memoria. El resto de direcciones se utilizarán para futuras ampliaciones de memoria del computador.

Gráficamente se puede ver el número de bits del bus de direcciones que permite seleccionar los chips de memoria:

20 bits 0 00000 Fila 0: 16

.... chips de

A19 A18 A17 A16 A15 . . . A1 A0 216 -1 0FFFF 64Kx1

216 10000

Selección

Chip 217 -1 1FFFF Fila 1: 16

Direccionamiento chips de

Interno 64Kx1

220 -1 FFFFF

Mapa de Memoria:

A19	A18	A17	A16	A15	A14	A13	A12	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
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
0	0	0	1	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	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

- Dibujar el diagrama de conexiones de la memoria junto con la lógica de selección.

Para el diseño del diagrama de conexiones de la memoria junto con la lógica de control hemos empleado distintos bloques:

- Para la C.P.U.:
- Para la memoria:

Luego el diseño del diagrama de conexiones de la memoria junto con la lógica de control sería el siguiente:

Bus Direcciones

A19...A0 Bus Datos

D15...D0

—

A16 WE

A15...A0

A15...A0 A15...A0

D0 D0

A15...A0 A15...A0

D1 D1

.....

.....

A15...A0 A15...A0

D15 D15

Fila 1 Fila 2

• **Memoria cache.**

A continuación veremos la necesidad de la existencia de la memoria cache.

4.1. Concepto.

La memoria de la CPU es más rápida que la 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.

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

Transferencia Transferencia

de palabras de bloques

La memoria principal de 2^n palabras está organizada en M bloques de longitud fija (K palabras / bloque) donde $M = 2^n / K$ bloques. En esta figura se muestra la configuración de la memoria:

0

1

Bloque 0

K - 1

.....

Direcciones

.....

Bloque M - 1

2n - 1

La memoria cache está dividida en C líneas o particiones de K palabras siendo $C \ll M$. En esta figura se muestra la configuración de la memoria:

Etiqueta Bloque

0

1

.....

Líneas

.....

C - 1

K palabras

Para acabar este apartado conceptual de la memoria cache debemos tener en cuenta el siguiente concepto: tasa de acierto.

• **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. La tasa de acierto se calcula mediante la siguiente expresión:

nº aciertos

Tasa de acierto =

nº referencias

4.2. Parámetros de diseño.

La memoria cache surge como consecuencia del aumento de la velocidad del sistema. A la hora de diseñar la memoria cache debemos de tener en cuenta los siguientes parámetros:

-
- Tamaño de la memoria cache.
- Tamaño del bloque.
- Número de caches.
- Contenido de la cache.
- Estrategia de escritura.
- Función de correspondencia.
- Correspondencia directa.
- Correspondencia asociativa.
- Correspondencia asociativa por conjuntos.
-

4.2.1. Tamaño de la memoria cache.

Este parámetro 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 principal.

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

4.2.2. Tamaño del bloque.

Cuando se va aumentando el tamaño del bloque, a partir de valores muy pequeños, la tasa de acierto inicialmente aumentará; pero a partir de un cierto tamaño del bloque la tasa de acierto comienza a disminuir. Surgen entonces 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 entre 4 y 8 unidades direccionables.

4.2.3. Número de caches.

Este parámetro plantea dos tipos de cache distintos:

- Cache interna. Nivel 1. Físicamente está ubicada en el mismo chip que el procesador (CPU) y aumenta la velocidad de procesamiento. Luego los accesos a esta cache se efectúan muy rápido. La capacidad de esta cache es bastante pequeña.
- Cache externa. Nivel 2. Físicamente está ubicada fuera del chip del procesador (CPU) por lo que será más lenta que la cache de nivel 1 pero seguirá siendo más rápida que la memoria principal. Al estar fuera (se sitúa entre el procesador y la memoria principal) la capacidad podrá ser mayor que la cache de nivel 1. Su uso se está extendiendo de tal forma que cada dispositivo se fabrica con su propia cache.

4.2.4. Contenido de la cache.

Lo ideal es que la memoria cache contenga instrucciones y datos y actualmente se emplean en diseños segmentados.

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 más 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.

4.2.5. 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, es decir, si el bloque es un bloque limpio o un bloque modificado, también llamado sucio. Los datos se modifican en la cache al ejecutar el programa. Si el bloque no fue modificado no hay problema pero sino, entonces existe un problema en la escritura; esto nos lleva a dos tipos de escritura:

- 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 asegura que los contenidos sean siempre válidos. El principal inconveniente es que genera mucho tráfico con la memoria principal y puede originar un cuello de botella.
- Post-escritura (write back). Las escrituras se realizan sólo en la memoria cache. Asociada a cada línea 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 la memoria principal, mientras que si está a cero 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 su uso en multiprocesadores.

Para acabar con este parámetro decir que conlleva un problema de coherencia de datos.

4.2.6. 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. Existen tres tipos de correspondencia por lo que existirán tres tipos función de correspondencia distintas:

- Correspondencia directa. El bloque 12 de memoria principal sólo 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)).

Este ejemplo de cada tipo de correspondencia se ve mejor en la figura siguiente:

Número Línea	Directa	Asociativa	Asociativa por conjuntos	
0				
1				
2				Conjunto 1
3				
4				Conjunto 2
5				
6				Conjunto3
7				

A continuación vamos a describir las tres técnicas antes enumeradas. En cada caso veremos la estructura general y un caso concreto.

Para los tres casos, vamos a trabajar con un sistema con las siguientes características:

- El tamaño de la memoria cache es de 4Kb = 4096 b.
- Los datos se transfieren entre memoria principal y la memoria cache en bloques de 16B (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 (ya que $2^{16} = 64K$). Esto nos indica que la memoria principal está compuesta por 4096 bloques (M).

4.2.7. 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 expresión

$$i = j \text{ módulo } m$$

donde:

- i = número de línea de cache.
- j = número de bloque de la memoria principal.
- m = número de líneas de la memoria cache.

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

s bits w bits

donde:

- w bits = identifica cada palabra dentro de un bloque.
- s bits = identifica el número de bloque.

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 tal y como se muestra en la siguiente figura:

$s + w$

Memoria

Etiqu. Datos Principal

Línea Bloque

0 0 0

$s-r$ r w

s

Línea Bloque

Acierto w i j

w

Fallo

Línea Bloque

$2r-1$ $2s-1$

Ejemplo:

Etiqueta Línea Palabra

Memoria Principal

Dirección Dato Memoria Cache

0000h

0001h

0002h Bloque 0

Etiqueta Datos N° línea

000Fh

· 0 (00)

· 1 (01)

·

F800h 128 (80)

F801h

F802h Bloque 255 (FF)

3968

F80Fh

· 4 bits 16 bytes 8 bits

·

·

FFF0h

FFFDh Bloque

FFFEh 4095

FFFFh

Ventaja: la técnica de correspondencia directa es simple y poco costosa de implementar.

Inconveniente: hay una posición concreta de cache para cada bloque dado.

4.2.8. 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

s bits w bits

donde:

- palabra = identifica cada palabra dentro de un bloque de MP.
- etiqueta = identifica unívocamente un bloque de MP.

Para determinar si un bloque está en la memoria cache, se debe examinar simultáneamente todas las etiquetas de las líneas de memoria cache, si la etiqueta está entonces es un acierto y se indica que la palabra escogida es la que corresponde en la cache, sino entonces es una fallo y la dirección es la que nos dice dónde está en la memoria principal.

Veamos cómo ocurre esto gráficamente:

s + w

s w Memoria

s Etiq. Datos Principal

Línea Bloque

0 0 0

s

s

Línea Bloque

w i j

w

Comparador

simultáneo s

Línea Bloque

$2r - 1$ $2s - 1$

Fallo

Acierto

Ejemplo: *Etiqueta Palabra*

Memoria Principal

Dirección Dato Memoria Cache

0000h

0001h

0002h Bloque 0

Etiqueta Datos N° línea

000Fh

.

.

·.

F800h

F801h

F802h Bloque

3968

F80Fh

· *12 bits 16 bytes*

.

.

FFF0h

FFFDh Bloque

FFFEh 4095

FFFFh

Ventaja: acceso muy rápido.

Inconveniente: necesidad de una circuitería bastante compleja.

4.2.9. Correspondencia asociativa por conjuntos.

Esta técnica es un compromiso que trata de aunar las ventajas de las dos técnicas vistas anteriormente. Memoria cache dividida en T conjuntos de L líneas. Las relaciones que se tienen son:

$$C = T \times L \text{ e } i = j \text{ módulo } T$$

donde:

- i = número de conjunto de MC (memoria cache).
- j = número de bloque de MP.

El bloque B_j puede asociarse a cualquiera de las líneas del conjunto i . La lógica de control de la memoria cache interpreta una dirección de MP con tres campos

s — d bits d bits w bits

donde:

- w bits de menor peso = palabra dentro de un bloque.
- s bits = identifica un bloque de MP.
- d bits = especifica uno de los conjuntos de la MC.
- $s - d$ bits = etiqueta asociada a las líneas del conjunto d bits.

Para saber si una dirección está o no en la memoria cache, lo primero se aplica correspondencia directa y luego correspondencia asociativa. Veamos cómo ocurre esto gráficamente:

$s + w$

$s - d$ d w Memoria

Etiqu. Datos Principal

Conjunto Bloque

0 0 0

$s - d$

s

Conjunto Bloque

$s - d$ w i j

w

Comparador

simultáneo $s - d$

k líneas Conjunto Bloque

$2^d - 1$ $2^s - 1$

Fallo

Acierto

Ejemplo: *Etiqueta Conjunto Palabra*

Memoria Principal

Dirección Dato Memoria Cache

0000h

0001h

0002h Bloque 0

000Fh

.

.

.

0400h

0401h

0402h Bloque

64

040Fh

.

.

.

FFF0h 6 b 16 B 6 b 16 B

FFFDh Bloque

FFFEh 4095

FFFFh

4.2.10. 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 en estos algoritmos. Entre los diferentes algoritmos que se han propuesto destacan los siguientes:

- LRU.
- FIFO.
- LFU.

4.2.11. Ejemplos de memoria cache.

Los procesadores de Intel han ido incorporando la memoria cache para aumentar su rendimiento. A continuación veremos los tipos de procesador más utilizados por Intel y las características de sus memorias cache:

Procesador Nivel 1 Nivel 2

Inferior 80386 NO NO

80386 NO 16K, 32K, 64K

80486 8K datos / instrucciones 64K, 128K, 256K

Pentium 8K datos y 8K instrucciones 256K, 512K, 1M

Procesador Tipo cache (nivel 1 y 2) Descripción

80486 Cache interna Tamaño de línea de 16 bytes

Organización Asoc. por Conjuntos de 4 vías

Cache externa Tamaño de línea de 32, 64 ó 128 bytes

Organización Asoc. por Conjuntos de 2 vías

Pentium Cache interna Tamaño de línea de 32 bytes

Organización Asoc. por Conjuntos de 2 vías

Cache externa Tamaño de línea de 32, 64 ó 128 bytes

Organización Asoc. por Conjuntos de 2 vías

• Memoria asociativa.

A continuación estudiaremos la memoria asociativa que se caracteriza porque su acceso es por contenido.

5.1. 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 denomina direccionables por contenido: CAM (Content Addressable Memory).

5.2. 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 marca (M) de n bits.

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 argumentado. 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.

La comparación simultánea se realiza bit a bit. El bit A_j ($j = 1, 2, \dots, m$) 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, $M_i = 0$.

Registro argumento

Registro máscara

Registro de Marca

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

IMPORTANTE

Nota 2, sobre Registros de memoria asociativa: Existen 3 registros:

- Registro argumento: lo que quiero averiguar si está en el contenido de la memoria.
- Registro de máscara: podré quedarme con el que quiero averiguar que esté en la memoria. Determina filtrado del argumento a buscar.
- Registro de marca: tiene tantos bits como palabras tenga en memoria y pone un bit donde encuentre el argumento buscado a 1 y 0 donde NO esté. Sirve para saber si en esa palabra (fila de la matriz) está el argumento que busco.

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.

A continuación estudiaremos la memoria compartida.

6.1. Concepto.

La memoria compartida surge por la necesidad de que diferentes dispositivos tengan acceso a una misma unidad de memoria.

Petición

Petición

Espera Memoria

Petición

Espera

.....

.....

Petición

Espera

.....

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, es decir, que arbitra qué elemento en cada momento puede compartir el uso de la memoria. El árbitro se diseña de forma que asigne un tiempo de servicio, en promedio, análogo a todas las 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.

6.2. Memorias de doble puerto.

Surgen como consecuencia de la aparición de la memoria compartida. Las memorias de doble puerto son memorias compartidas que permiten trabajar con dos elementos a la vez. Se basan en duplicar los buses de direcciones, los decodificadores, la selección de chip, la L/E y bits de semáforo para establecer prioridad de las CPU's que acceden a la misma dirección de memoria, por ejemplo, *con VRAM puede acceder la tarjeta gráfica y el monitor a la vez.*

La configuración de las memorias de doble puerto es la siguiente:

Bus Datos Bus Datos

I/OL I/OR

Bus Direcciones Bus Direcciones

AL AR

Bus Control Bus Control

— — — — — — — — — —

CEL, R/WL, OEL, SEML CER, R/WR, OER, SEMR

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

Puerto Izquierdo Puerto Derecho Descripción

I/OL I/OR Bus de datos

AL AR Bus de direcciones

— —

CEL CER Selección de chip

— —
R/WL R/WR Lectura / Escritura

———— ————
OEL OER Habilita lectura

———— ————
SEML SEMR 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.

- **Otras memorias.**

A continuación estudiaremos otros tipos de memoria.

7.1. Introducción.

El rendimiento de cualquier computador se ve afectado principalmente por la frecuencia del bus, que hoy en día puede considerarse de 66 MHz o de 100 MHz. La final de los procesadores se obtiene a partir de un multiplicador interno.

Para no reducir el rendimiento del computador, las memorias deberían ser igual de rápidas que el procesador, pero hoy en día no es así. Por tanto, cada vez que el procesador accede a la memoria tiene que esperar. El uso de las memorias cache aumentan el rendimiento del sistema, pero debido a su alto coste y al espacio que ocupan físicamente no es viable añadir gran cantidad de memoria cache.

A medida que ha evolucionado la tecnología, las memorias cada vez se han fabricado más rápidas para no hacer esperar al procesador. Así, pues, otra alternativa para aumentar el rendimiento del sistema es utilizar memorias principales más rápidas.

7.2. Memorias entrelazadas.

En las memorias entrelazadas los módulos se organizan de tres formas que estudiaremos más adelante:

- Organización con acceso S.
- Organización con acceso C.
- Organización con acceso C/S.

7.2.1. Organización S.

Es una de las configuraciones más sencillas:

- Utiliza entrelazado de orden inferior y aplica los (n – m) bits superiores de la dirección a todos los módulos $M = 2^m$ módulos de memoria simultáneamente en un acceso.
- En un único acceso se obtiene M palabras consecutivas de información procedentes de los M módulos de memoria.
- Los m bits inferiores de la dirección se utilizan para seleccionar la información de un módulo particular.

Este tipo de configuración se denomina de acceso S porque a todos los módulos se acceden simultáneamente tal y como se puede observar en la figura:

Registro Datos

Bus Datos

.....

.....

Bus

$n - m$ Direcciones

El funcionamiento es sencillo:

- La dirección de memoria se divide en dos por tiempo de acceso y tiempo de recuperación; sólo nos quedamos con una palabra del módulo de memoria lo que implica que por cada módulo de memoria tendremos un registro de datos.
- Los módulos de memoria se organizan partiendo de las matrices de bits de las memorias; con los $n - m$ bits más significativos tengo la selección de cada uno de los módulos de memoria, de ellos leo a la vez así como de los registros, pero a la salida sólo tengo una palabra por el multiplexor que he puesto antes del bus de datos.

Este tipo de configuración es ideal para acceder a un vector de datos o para la búsqueda de instrucciones y operandos secuenciales. También es muy útil en la transferencia de bloques en sistemas con memoria cache.

Inconveniente:

- Acceso a posiciones de memorias no consecutivas.
- Se utilizan todos los módulos para acceder a una posición.

7.2.2. Organización C.

Este tipo de organización tiene las siguientes características:

- Permite acceder a posiciones distintas de los módulos concurrentemente.
- Los m bits de orden inferior se utilizan para seleccionar el módulo y los $n - m$ bits restantes direccionan el elemento deseado dentro del módulo.
- El controlador de memoria se utiliza para mantener una petición que referencie un módulo ocupado. Existen configuraciones en las que el controlador dispone de una única cola FIFO o bien dispone de una cola FIFO por cada módulo de memoria.

Este tipo de configuración se denomina de acceso C porque a todos los módulos se acceden de manera concurrente tal y como se puede observar en la figura:

Bus Datos

$2m$

...

...

Registros de dirección y control

n-m

Bus

Direcciones

n m

.....

Ventaja: para acceder a una posición de memoria ya no se ocupan todos los módulos sino que se ocupa únicamente el módulo que contiene esa dirección y el resto están libres. Para seleccionar el módulo correspondiente se emplea el decodificador.

7.2.3. Organización C/S.

Es una configuración mixta:

- Combina los esquemas con acceso S y acceso C.
- Los módulos se organizan en forma de matriz bidimensional. Si el acceso S es un entrelazado de M vías y el acceso C un entrelazado de L vías, hasta L accesos diferentes a bloques de M palabras consecutivas por estar en progreso simultáneamente.

7.3. Fast Page Mode RAM (FPM RAM).

Este tipo de memoria incorpora un sistema de paginado puesto que considera probable que el próximo dato a acceder esté en la misma columna, ganando en tiempo en caso afirmativo.

Las transferencias de datos desde la memoria se realizan en paquetes de 4 datos denominados ráfagas (burst). El uso de ráfagas elimina los tiempos de establecimiento de dirección y de precarga de líneas de fila y columna posteriores al primer acceso.

El rendimiento de un tipo de memoria se expresa con cuatro números separados por guiones que son los de ciclo de reloj que la memoria necesita para responder. La memoria ideal sería 1-1-1-1 que significa que en cada ciclo de reloj se transfiere un dato. Para una cache del tipo L2 (SRAM) es 2-1-1-2.

La memoria FPM RAM tiene un esquema en el caso más favorable de 5-3-3-3, es decir, cuatro estados de espera en el primer dato y dos en los sucesivos, haciendo un total de 10 por ráfaga. Por tanto, con una memoria FPM de 60 ns alcanzamos un ancho de banda de 28.5 MHz

7.4. Extended Data Output RAM (EDO RAM).

Es una modificación de la FPM RAM. Mientras accede a los datos se prepara la siguiente dirección. La EDO RAM es la memoria más popular por el momento debido a que su velocidad de acceso se incrementa notablemente y a que los fabricantes han tenido que hacer muy pocos cambios respecto de la FPM RAM.

La EDO RAM puede trabajar, en el caso más favorable, con un esquema 5-2-2-2, lo que puede suponer una

mejoría de hasta un 40% en el rendimiento global. Aún así, el ancho de banda alcanzado con memorias de 60 ns, es de 40 MHz, todavía lejos de los 66 MHz del bus de un PC. Algunos PC's permiten trabajar con memoria de 50 ns consiguiendo un ancho de banda de 50 MHz.

Dentro de poco tiempo, los computadores utilizarán un bus de 100 MHz y la memoria EDO ya no tendrá sentido utilizarla ya que, como hemos visto, su ancho de banda llega 50 MHz y reduciría mucho el rendimiento del sistema.

7.5. Burst Extended Data Output RAM.

Esta memoria permite trabajar con un esquema 5-1-1-1 utilizando 50 ns de tiempo de acceso consiguiendo alcanzar los 66 MHz. Estas memorias no han tenido mucho éxito debido a la aparición del bus de 100 MHz y las memorias SDRAM.

7.6. Synchronous DRAM (SDRAM).

La memoria DRAM síncrona o SDRAM a diferencia de las DRAM típicas, que son asíncronas, intercambia datos con el procesador de forma sincronizada con una señal de reloj externa, que opera a la velocidad del bus sin imponer estados de espera.

El núcleo DRAM es mucho más rápido que el de la memoria convencional, llegando a alcanzar velocidades de hasta cuatro veces más. Una arquitectura de doble banco permite entrelazados, de forma que mientras un banco prepara los datos otro los proporciona. Así mismo, la longitud de la ráfaga es programable, permitiendo un diseño flexible en función del sistema que deba soportarlo.

El esquema de trabajo de la SDRAM es de 5-1-1-1. Con memorias de 15 ns se pueden alcanzar los 66 MHz de ancho de banda, mientras que con memorias de 10 ns se pueden alcanzar los 100 MHz. Esta última configuración es la que se impondrá dentro de poco tiempo en los nuevos PC's.

Un ejemplo de SDRAM es la *PC-100 RAM* que cumple determinadas restricciones establecidas por Intel para el correcto funcionamiento con el bus de 100 MHz implementado en su chipset 440BX.

7.7. Double Data Rate SDRAM (DDR SDRAM).

Son una mejora de las memorias SDRAM. A diferencia de las anteriores, ésta transfiere la información tanto con el flanco de bajada como en el flanco de subida, con lo que duplica la cantidad de información que puede transferir. De esta forma se puede llegar a 200 MHz de ancho de banda, con lo que se tendría en el bus del PC una tasa de hasta 4.6 GB/sg.

Funciona a velocidades de 100 y 125 MHz; en un futuro esta velocidad puede incluso llegar a triplicarse o cuadruplicarse, con lo que se adaptaría a los nuevos procesadores y buses del tipo AGP 100 MHz x2 y x4.

Un factor a tener en cuenta es que es una arquitectura abierta, por lo que no se tiene que pagar derechos a ningún fabricante. Sin embargo, un factor negativo es el hecho de que el fabricante de procesadores Intel no apoya este tipo de memorias, por lo que no se sabe qué futuro les deparará.

7.8. Synchronous Link DRAM.

Esta memoria posee las ventajas de la SDRAM y la DDR SDRAM. Además se le ha incorporado un protocolo orientado a paquetes para el control/direccionamiento y un sistema de calibrado de tiempos para mantener la compatibilidad con las antiguas memorias.

Permite trabajar a grandes velocidades (3.2 GHz) y hasta 4 GB/sg de transferencia. Se cree que puede ser la memoria a utilizar en los grandes servidores por su alta transferencia de datos.

Para acabar hay que mencionar que este tipo de memoria, al igual que la memoria DDR SDRAM, tampoco ha sido respaldada por Intel.

7.9. *Direct Rambus DRAM.*

Puede producir ráfagas de 2 ns y puede alcanzar tasas de transferencia de 533 MHz, con picos de 1.6 GB/sg. Es el complemento ideal para las tarjetas gráficas AGP, evitando los cuellos de botella en la transferencia entre la tarjeta gráfica y la memoria del sistema. Al usar este tipo de memoria se ha de pagar derechos a Intel y, a diferencia de los dos tipos anteriores, sí está respaldada por Intel.

Tema 4: La Memoria – 93 –

Estructura de Computadores 1 Guardado por José David Balibrea Sánchez

$n-1$ $n-2$ 0

0

1

...

...

...

$m-1$

0

1

...

...

$2m-2$

$2m-1$

$n-1$ $n-2$ 0

0 1

... ...

... $n-2$ $n-1$

0

1

...

...

$2X - 2$

$2X - 1$

0

...

$m/2 - 1$

$m/2$

...

$m - 1$

$n - 1 \ n - 2 \ \dots \ 0$

$2Y - 1 \ 2Y - 2 \ \dots \ 0$

Registros CPU

Memoria Cache

Memoria Principal

Memoria Cache de Disco

Unidad de Disco

Cinta (Streamer)

0

1

....

4094

4095

n palabras

Chip

de

Memoria

LÓGICA

DE CONTROL

DE

CO

DI

FI

CA

FI DO

LA R

S

DECO -DIFICA-DOR

COLUMNAS

BUFFER ENTRADA

MATRIZ

DE

MEMORIA

DE

256x32x8

B

U S

F A

F L

E I

R D

S A

Memoria

64K x 4

Memoria

64K x 4

- 16

- 4

CS

__ 8

WE

BUS

DATOS

Memoria

32K x 8

Memoria

32K x 8

15

8

CS

15 8

8

CS

—

WE

.....

.....

.....

CPU

—

CS WE

Chip 0

64Kx1

CPU

Memoria

Cache

Memoria

Principal

—

CS WE

Chip 2

64Kx1

—

CS WE

Chip 1

64Kx1

—

CS WE

Chip 2

64Kx1

—

CS WE

Chip 1

64Kx1

—

CS WE

Chip 0

64Kx1

CPU

—

CS WE

Chip 0

64Kx1

.....

Etiqueta Línea Palabra

Etiqueta Línea Palabra

Compara

4 bits 8 bits 4 bits

12

34

56

78

.

.

.

AB

0F

FF

54

.

.

.

FF

21

33

55

• 123456.....78

.....

.....

F AB0FFF.....54

.....

F FF.....213355

Etiqueta Palabra

Etiqueta Palabra

Etiqueta Conjunto Palabra

12 bits 4 bits

12

34

56

78

.

.

.

AB

0F

FF

54

.
 .
 .
 FF
 21
 33
 55
 000 123456.....78 0
 1

 F80 AB0FFF.....54 128

 FFF FF.....213355 255
6 bits 6 bits 4 bits
 12
 34
 56
 78
 .
 .
 .
 AB
 0F
 FF
 54
 .
 .

.

FF

21

33

55

Etiqueta Datos N° Conj. Etiqueta Datos

00 123456....78 0

.....

..... 64

Etiqueta Conjunto Palabra

0 Alicante 0

0 1 0

1

0

1

0

1

0

0

1

Elemento 1

Elemento 2

Elemento n

Árbitro

Memoria

Compartida

Multiplexor

IZQUIERDA

Buffer E/S

DERECHA

Buffer E/S

IZQUIERDA

Decodificador

Direcciones

DERECHA

Decodificador

Direcciones

MEMORIA RAM DOBLE PUERTO

Lógica de control y arbitraje

Etiqueta Datos N° Conj. Etiqueta Datos

04 AB0FFF....54 0

.....

3F FF....213355 64

Módulo 0

Módulo $2m - 1$

Módulo 1

Multiplexor

Módulo 0

Módulo $2m - 1$

Módulo 1

Ocupado

Controlador

De

Memoria

Decodificador