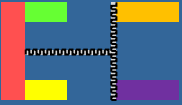


# TEMA 3.

## UNIDAD DE MEMORIA

dtic

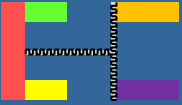




## Índice

- ① Conceptos básicos
  - ② Definiciones
  - ③ Características de las memorias
  - ④ Jerarquía de las memorias
- ① Tipos de memorias
- ① Memoria principal del computador. Memorias semiconductoras
- ① Diseño de mapas de memoria
- ① Otras memorias





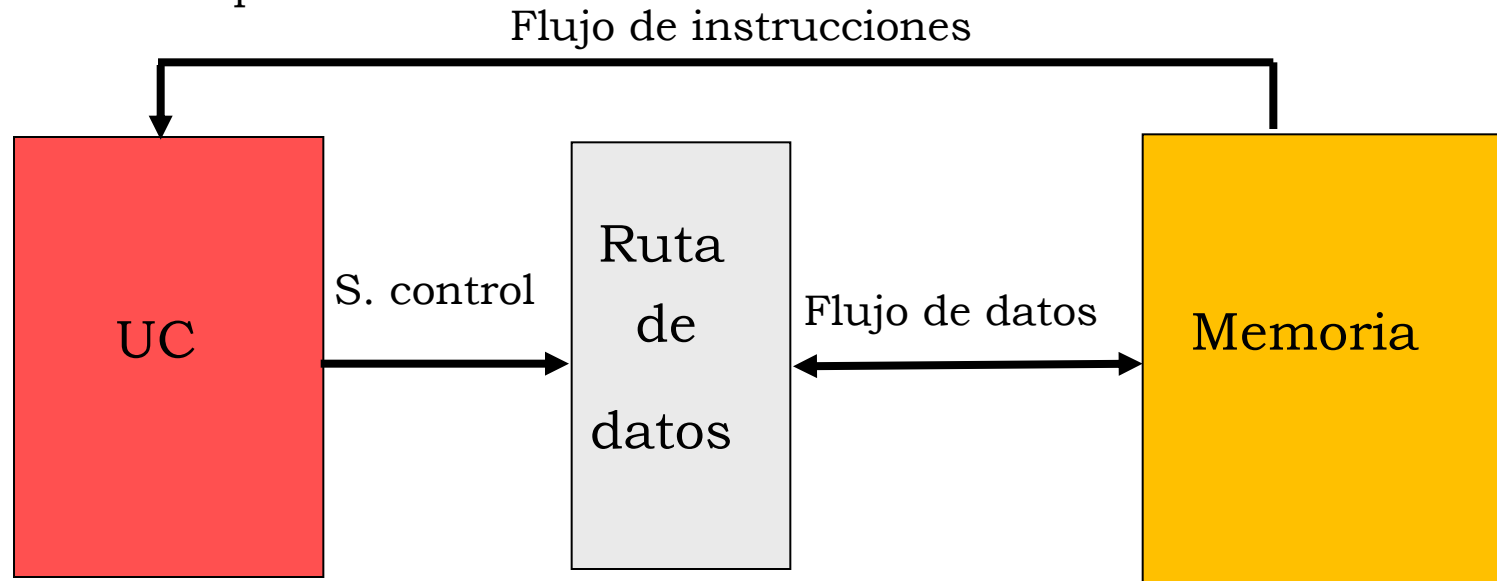
## Conceptos básicos

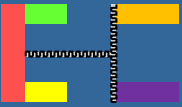
- ③ La memoria contiene los programas que se ejecutan en el computador y los datos sobre los que trabajan dichos programas.
- ③ La memoria es un elemento sencillo, sin embargo, presenta una gran diversidad de tipos, tecnologías, estructuras, prestaciones y costes.
- ③ Un computador dispone de una jerarquía de elementos de memoria donde algunos están localizados internamente en la CPU y otros externamente.

# COMUNICACIÓN CON LA MEMORIA

## Conceptos básicos

- En un monoprocesador tipo von Neumann, sólo existen dos caminos de transferencia de la información desde la **memoria**:
  - Uno bidireccional de datos
  - Otro unidireccional de instrucciones hacia la UC para su interpretación.

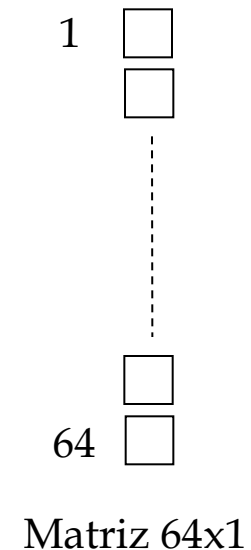
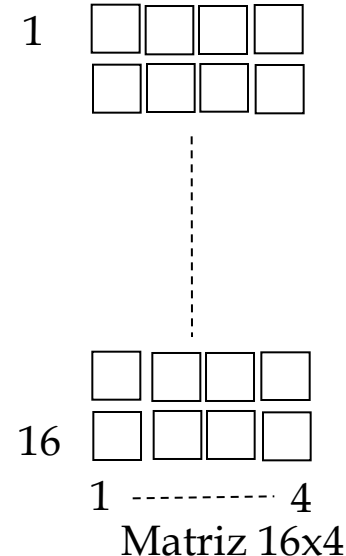
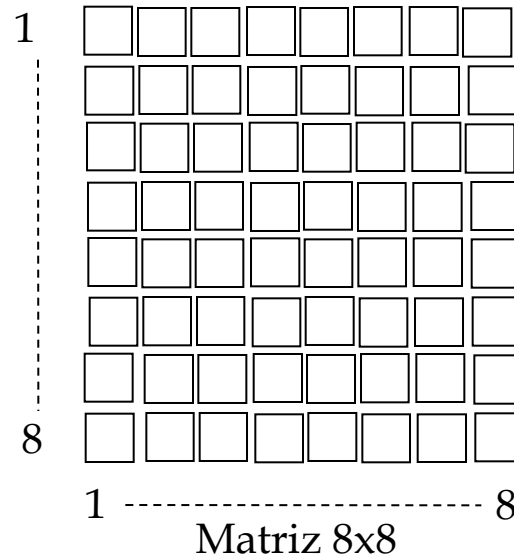


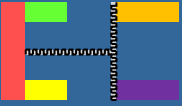


# DEFINICIONES

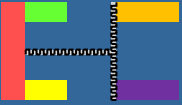
## Conceptos básicos

- ⊙ Cada elemento de almacenamiento se denomina celda y puede contener 1 ó 0.
- ⊙ Su organización puede ser muy diversa.





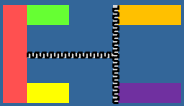
- ③ La posición de la unidad de datos se llama dirección.
- ③ Una memoria se identifica por el número de palabras que puede almacenar, 32Kxpalabras, significa  $32 \times 1024$  posiciones de memoria de un ancho **palabra**.
- ③ Si palabra es igual a 8 entonces decimos 32Kx8



- 🎯 Las dos operaciones básicas de las memorias son:

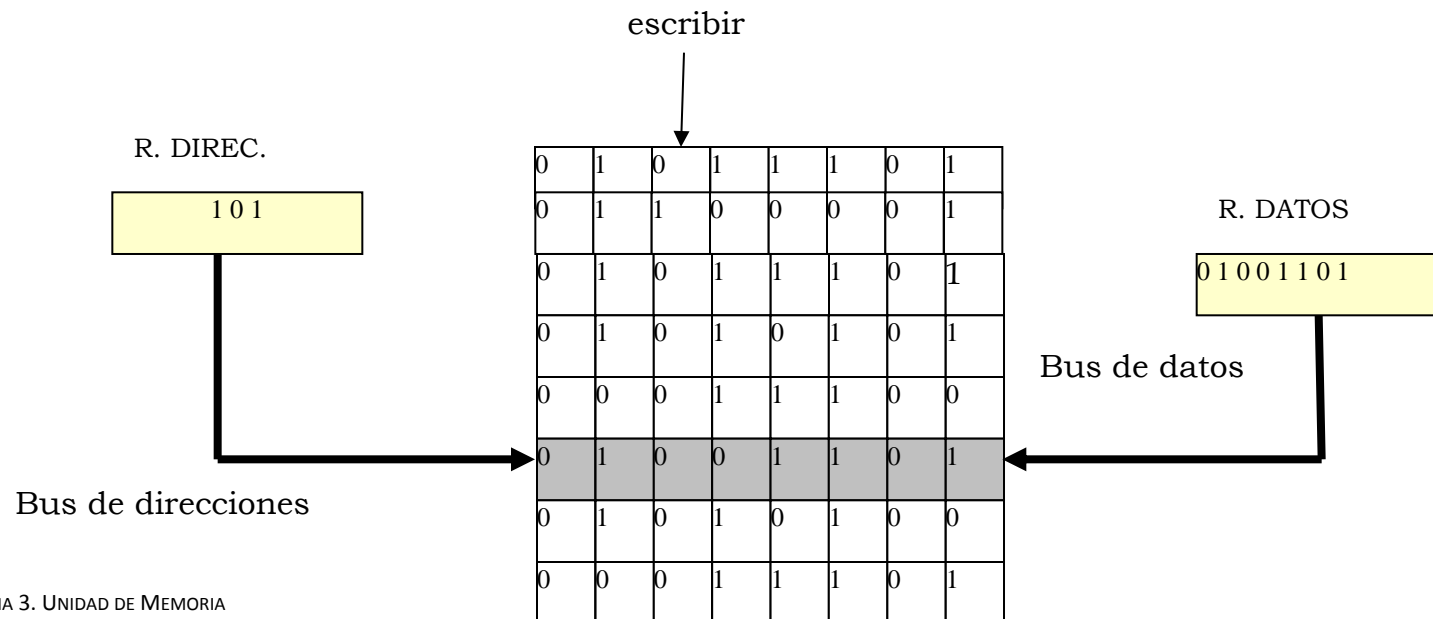
**Lectura.** Cargar en el registro, llamado habitualmente de datos, el contenido de la posición de memoria dada por el bus de direcciones.

**Escritura.** Almacenar el contenido del registro de datos, en la posición de memoria dada por la dirección especificada por el bus de direcciones.

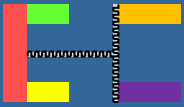


## Operación de escritura

1. colocar la dirección en el registro de direcciones
2. Colocar el dato en el registro de datos
3. Activar la orden de escritura

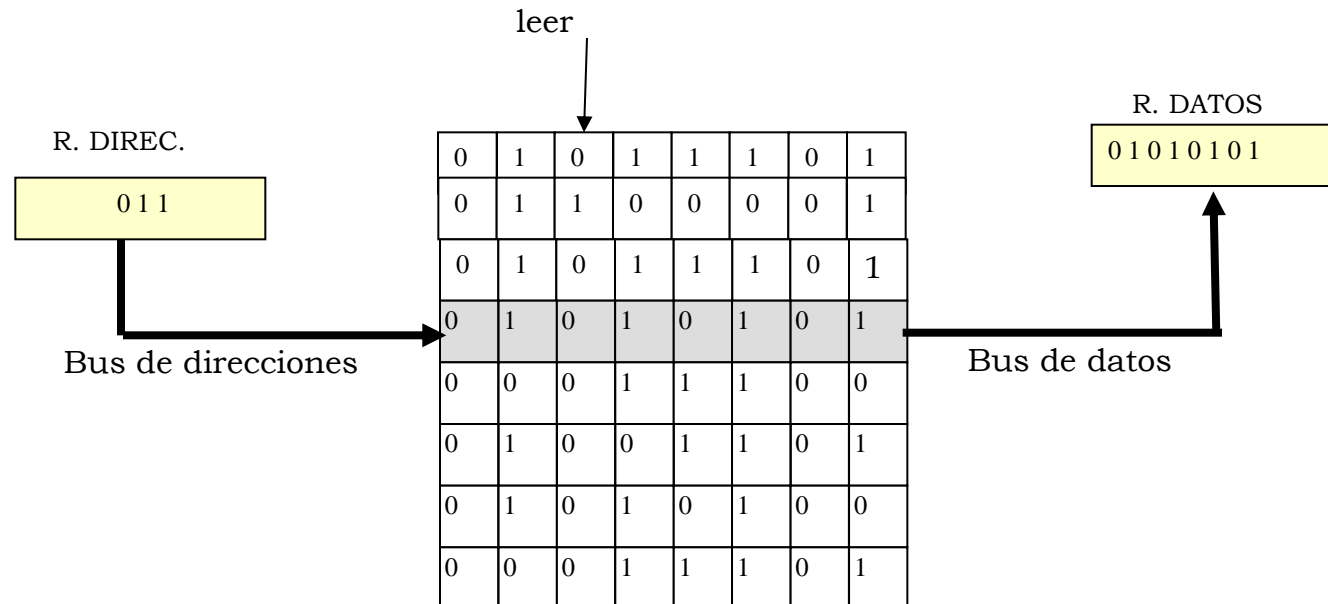






## Operación de lectura

1. colocar la dirección en el registro de direcciones.
2. Activar la orden de lectura.



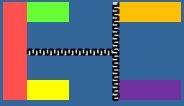


# REQUISITOS DE LAS MEMORIAS

## Conceptos básicos

Un sistema de memoria debe disponer de los siguientes elementos:

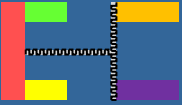
- ① **Medio o soporte.** Deberá disponer de un elemento donde se almacenen estados diferentes que codifiquen la información.
- ① **Transductor.** Es un elemento que permite convertir una energía en otra, es decir, transformar magnitudes físicas a eléctricas (sensor) o magnitudes eléctricas a físicas (actuador). Memoria estática y memoria dinámica.
- ① **Mecanismo de Direccionamiento.** Deberá disponer de un procedimiento para leer y escribir información en el lugar y tiempo deseado.



# UNIDAD DE TRANSFERENCIA

## Conceptos básicos

- ⊙ Unidad de transferencias 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.

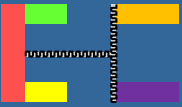


- ⊙ Organización interna de la memoria

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

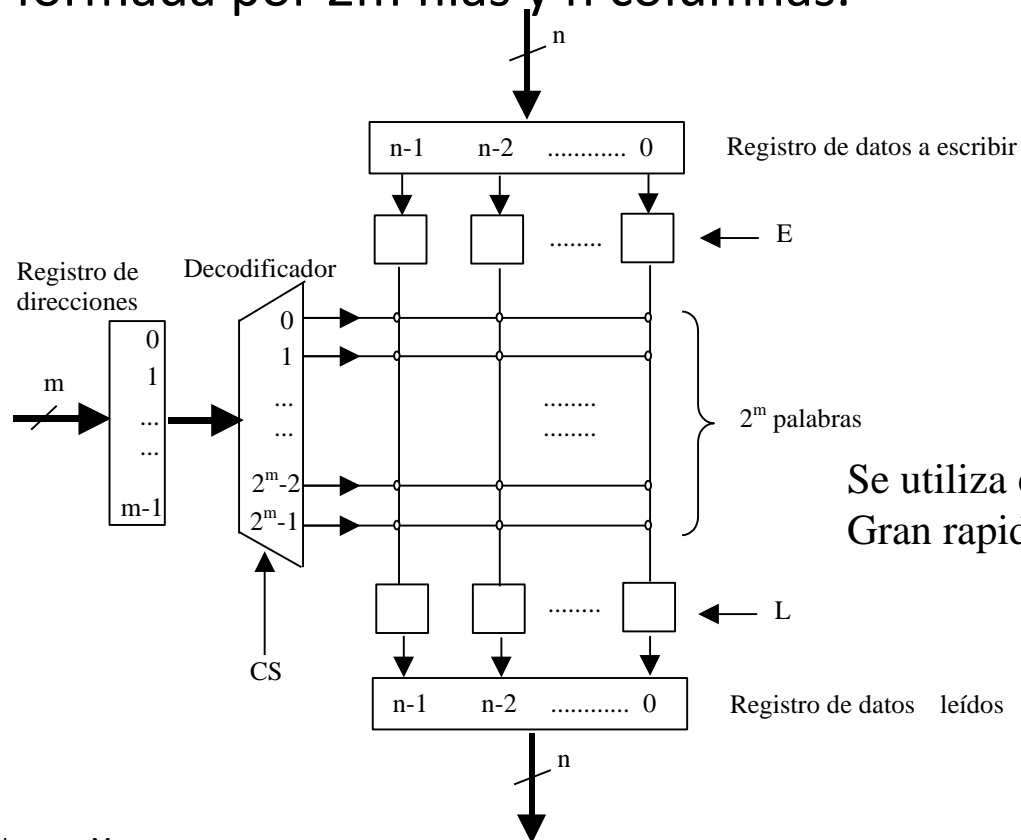
- ⊙ Para una memoria semiconductora distinguimos tres tipos de organización:

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

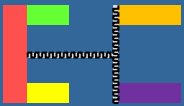


## Organización 2D

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

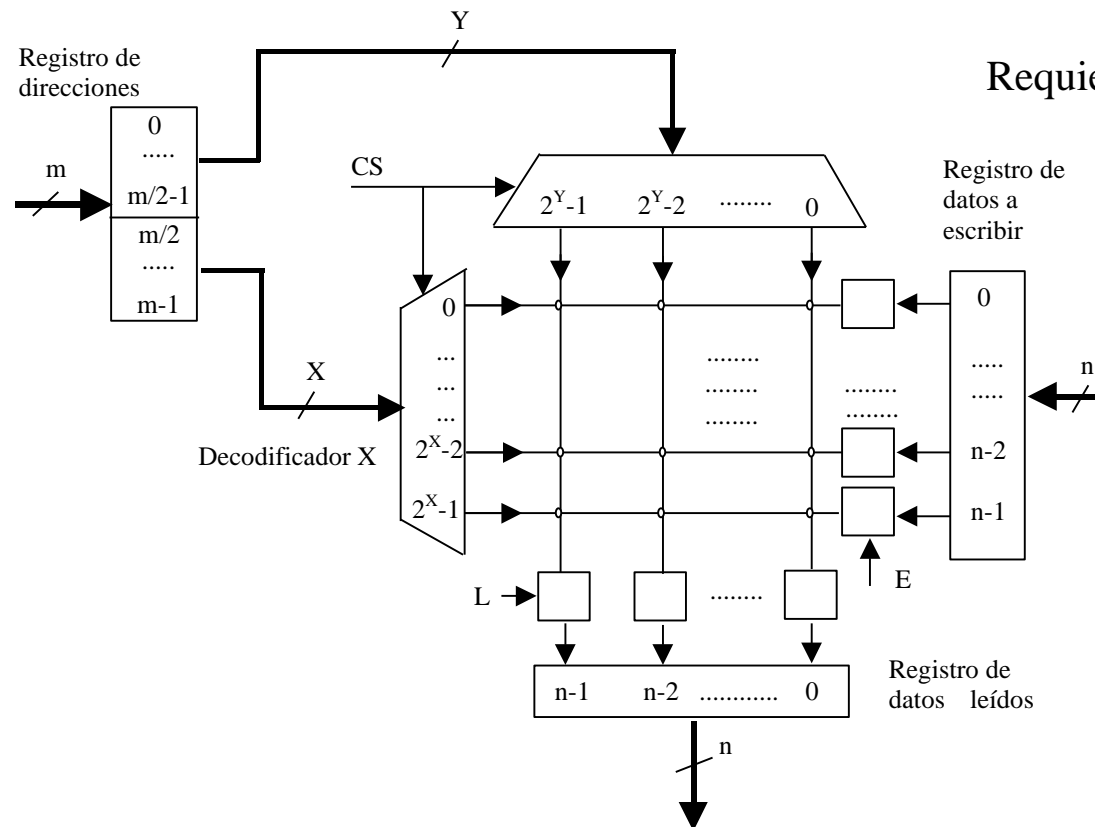


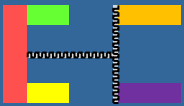
Se utiliza en memorias de capacidad reducida.  
Gran rapidez de acceso.



## Organización 2½D

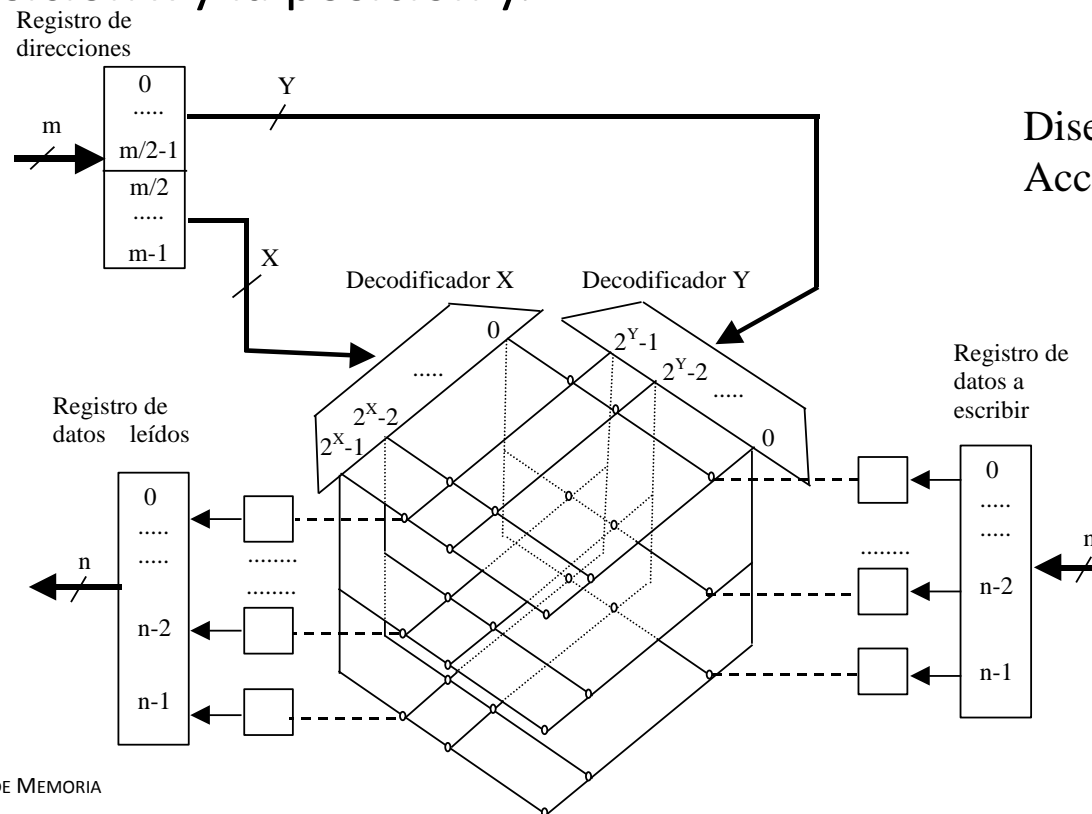
Utiliza dos decodificadores con  $m/2$  entradas y  $2m/2$  salidas.



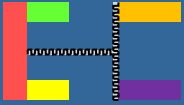


## Organización 3D

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



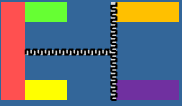
# CARACTERÍSTICAS DE LAS MEMORIAS

## Conceptos básicos

Existen tres características fundamentales que definen el funcionamiento de la memoria:

- ⦿ Capacidad de almacenamiento.
- ⦿ Velocidad o tiempo de acceso.
- ⦿ Coste por bit.





## Conceptos básicos

Es la cantidad de información binaria que puede almacenar, y depende del número de posiciones y de la longitud de cada una.

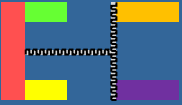
En la matriz anterior se puede ver como:

8x8 bytes.

16x4 nibbles.

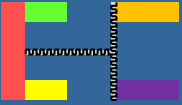
64 bits.

Una memoria de 16Mbytes tiene la misma capacidad que:  
8Mx16, 4Mx32, 128Mbits.



## Conceptos básicos

- ⊙ Ejemplo: ¿Cual de los siguientes modelos de memoria tiene mayor capacidad?
- ⊙ 16KBytes
- ⊙ 128Kx1
- ⊙ 8Kx32
- ⊙ 32Kx4



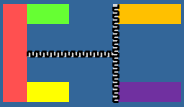
Para medir el rendimiento se utilizan tres parámetros:

⊙ 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.

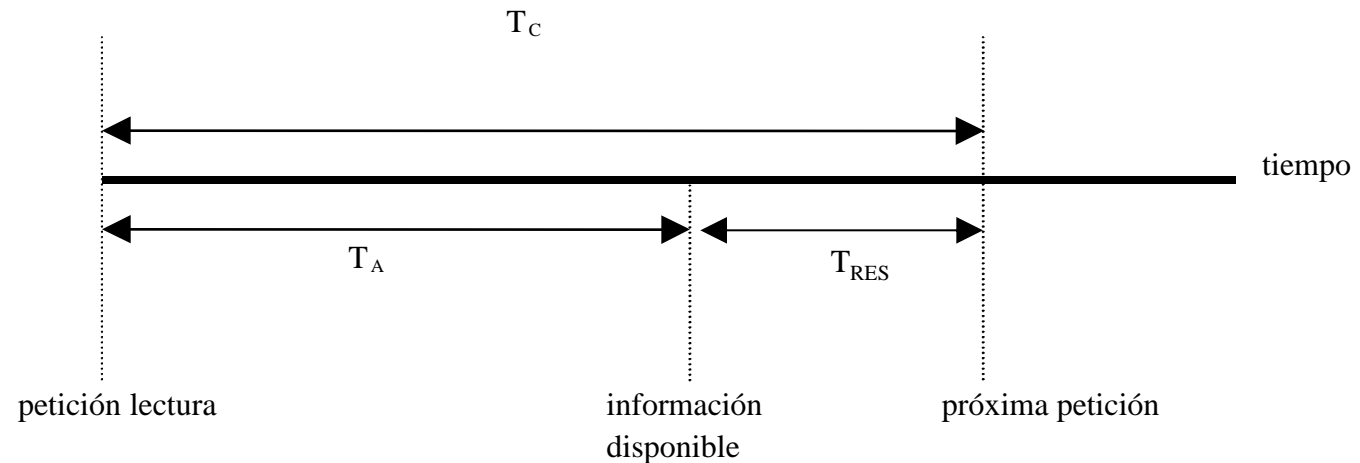
**Para memorias secuenciales:**

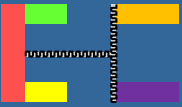
Tiempo que se emplea en situar el mecanismo de lectura/escritura en la posición deseada.



## ⊙ 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.





## Conceptos básicos

### 🎯 Velocidad de transferencia ( $V_T$ ).

Es el número de acceso a posiciones de memoria que se puede hacer en un segundo.

En el caso de acceso aleatorio

$$V_T = \frac{1}{T_C}$$

En el caso de acceso no aleatorio

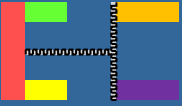
$$T_N = T_A + \frac{N}{V_T}$$

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

$T_A$  Tiempo de acceso

$N$  Número de bits

$V_T$  Velocidad de transferencia (bits/segundo)

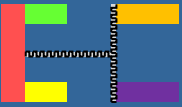


## 🎯 Ejemplo1:

Calcular el número de lecturas que se pueden realizar en una memoria DRAM en 2s, sabiendo que su tiempo de acceso es de 70ns y su tiempo de restauración de 30ns.

$$T_C = T_A + T_{RES} = 70 + 30 = 100\text{ns}$$

$$N^{\circ}lecturas / s = \frac{2}{100 \cdot 10^{-9} s} = 0,02 \cdot 10^9 = 20 \cdot 10^6 \text{ lecturas} / s$$

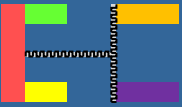


## 🎯 Ejemplo2:

Un disco óptico tiene un tiempo de ciclo de 2ms. Si cada posición consta de 32 bits. Calcular el número bytes que pueden ser leídos o escritos en 6s

$$V_T = \frac{1}{T_C} = \frac{1}{2 \cdot 10^{-3}} = 500 \text{ palabras/s}$$

Como una palabra tiene 4 bytes;  $V_T = 4 \times 500 = 2000 \text{ bytes/s}$



Es el precio que se paga por cada unidad mínima de información, es decir, por bit, en un determinado dispositivo de memoria.

Ejemplo:

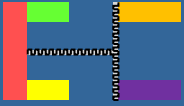
Por un módulo de memoria DRAM de 128MBytes se ha pagado 60€, y por un CD de 700MB se ha pagado 0,60€. Calcular la relación que existe entre ambos dispositivos con referencia al parámetro coste/bit

$$\text{DRAM} \quad \text{Coste} / \text{bit} = \frac{6000c\text{€}}{128M\text{Bytes}} = 5,58794 \cdot 10^{-6} c\text{€} / \text{bit}$$

$$\text{CD} \quad \text{Coste} / \text{bit} = \frac{60c\text{€}}{700M\text{bytes}} = 1,02179 \cdot 10^{-10} c\text{€} / \text{bit}$$

La memoria DRAM es 54687,75 veces más cara que el CD





# JERARQUÍA DE LAS MEMORIAS

## Conceptos básicos

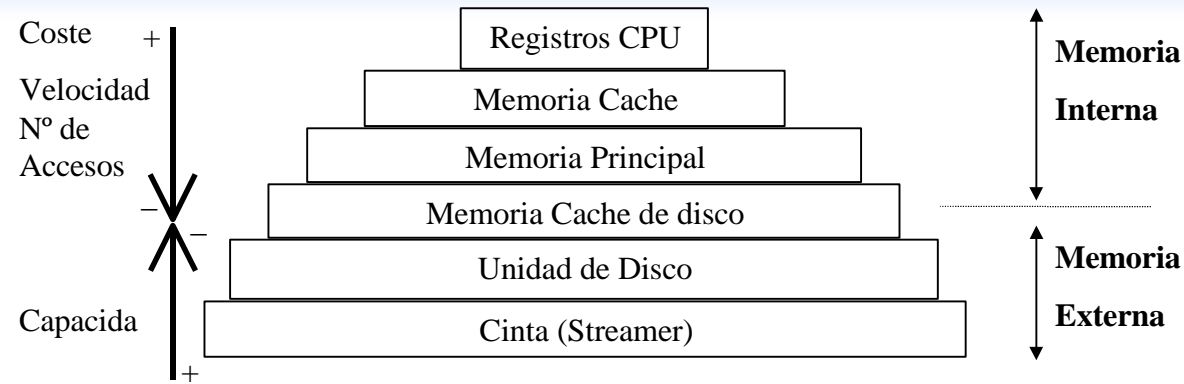
- ⊙ La configuración de memoria ideal es:
  - ⊙ Memoria muy rápida
  - ⊙ Gran capacidad
  - ⊙ Poco coste

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

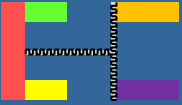


# JERARQUÍA DE LAS MEMORIAS

## Conceptos básicos



- 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 (temporal/espacial) depende de la frecuencia de accesos.

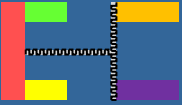


# TIPOS DE MEMORIAS

## Tipos de memorias

Las memorias se pueden clasificar atendiendo a diversos parámetros:

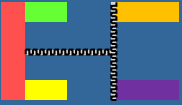
- ⦿ Ubicación
- ⦿ Formas de acceso a la información
- ⦿ Permanencia de la información
- ⦿ Tipo de operaciones
- ⦿ Interfaz con el exterior



## Tipos de memorias

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).

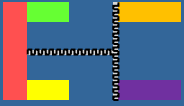


## Tipos de memorias

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).



# PERMANENCIA DE LA INFORMACIÓN

## Tipos de memorias

Relacionado con la duración de la información almacenada en memoria:

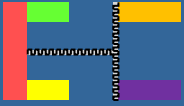
- ⦿ Lectura destructiva. Memorias de lectura destructiva (DRO: Destructive ReadOut) y memorias de lectura no destructiva NDRO (No Destructive ReadOut).
- ⦿ Volatilidad. Esta característica hace referencia a la posible destrucción de la información almacenada en un cierto dispositivo de memoria cuando 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).



# TIPOS DE OPERACIONES

## Tipos de memorias

- ② **Memorias de lectura-escritura** (Read-Write Memory). Memorias que permiten leer y escribir.
- ② **Memorias de sólo lectura** (Read Only Memory). Memorias que únicamente permiten leer.



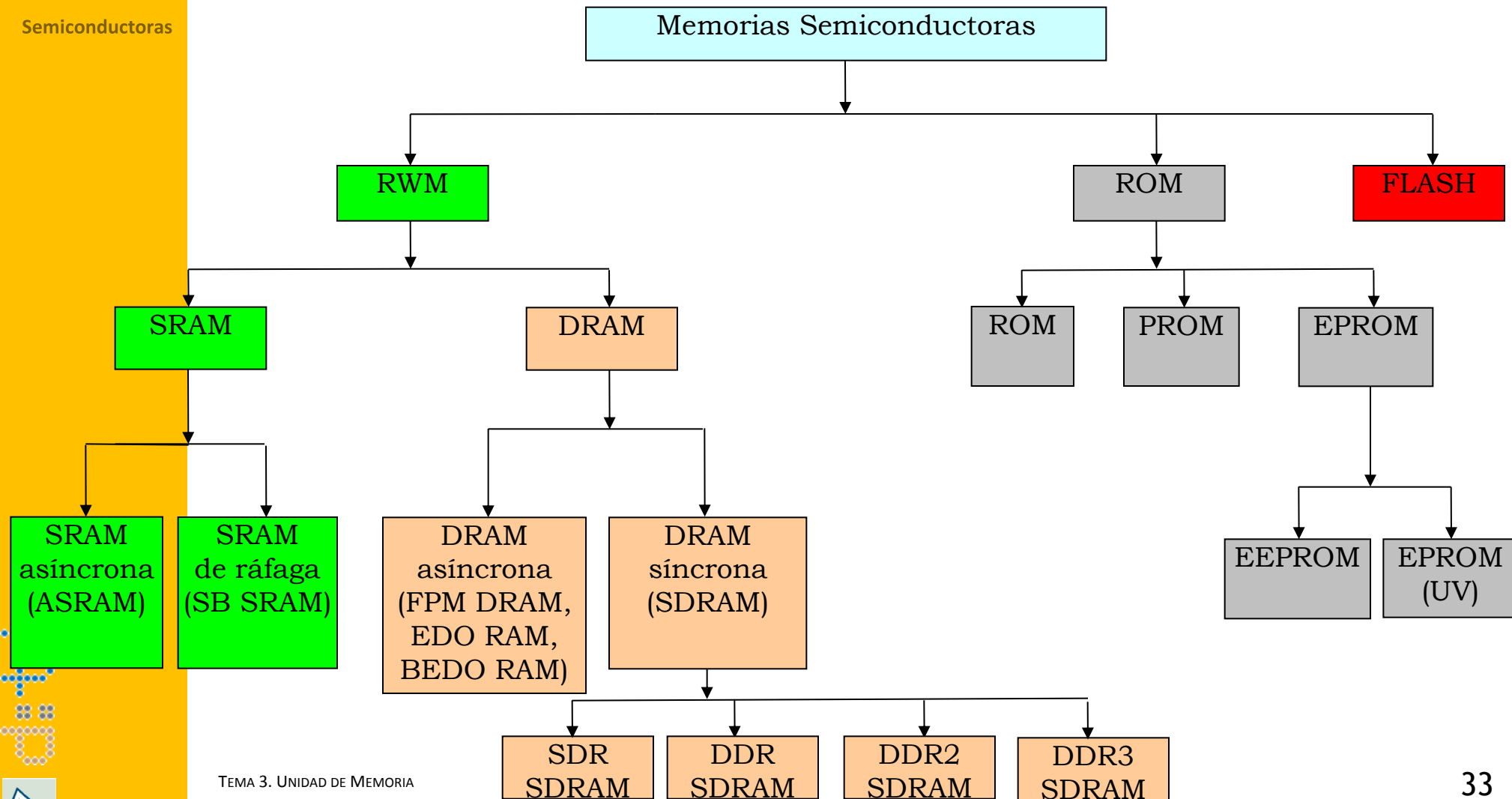
## Tipos de memorias

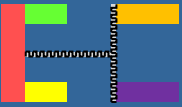
- ① **Síncronas.** Necesitan una señal de reloj ((clk)) que sincroniza la comunicación con el mundo exterior, es decir, las direcciones, datos y señales de control se almacenan en registros intermedios en el flanco activo de la señal del reloj (ascendente o descendente)
- ① **Asíncronas.** No necesitan la señal de reloj para su funcionamiento, es decir, las operaciones de lectura o escritura se realizan independientemente del reloj.



# MEMORIAS SEMICONDUCTORAS

Memorias  
Semiconductoras

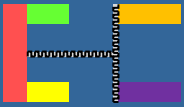




# READ ONLY MEMORY

Memorias  
Semiconductoras

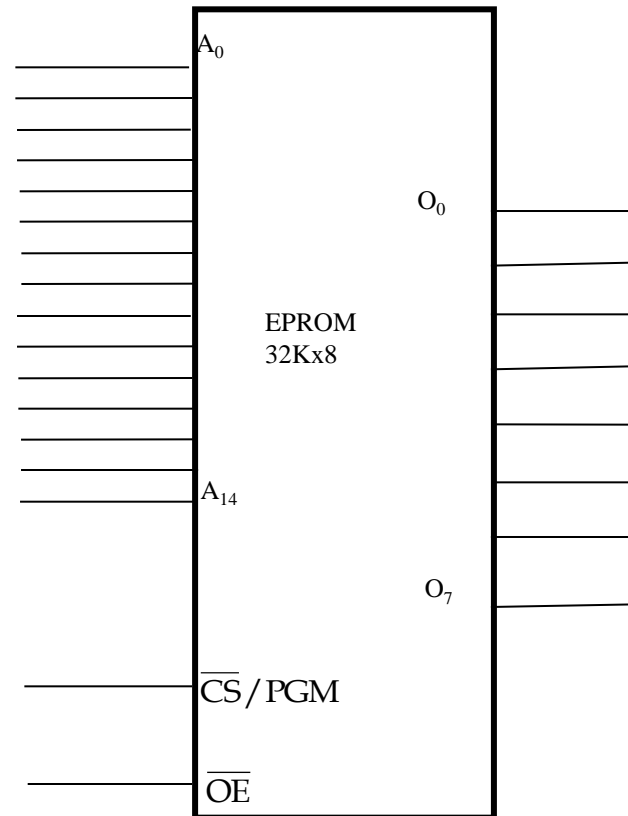
- ⊙ ROM (Read Only Memory). Se graba durante el proceso de fabricación y se suelen utilizar en microprogramación de sistemas, en subrutinas de bibliotecas de uso frecuente. También se llaman ROM de máscara
- ⊙ PROM (Programmable Read Only Memory). Son básicamente como una ROM de máscara. El proceso de 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.
  - ⊙ EPROM (Erasable Programmable Read Only Memory). Es una PROM borrrable mediante rayos ultravioleta que eliminan todo su contenido (UV EPROM)
  - ⊙ EEPROM (Electrically Erasable Programmable Read Only Memory). Se borra mediante corriente eléctrica de forma selectiva a nivel de byte.

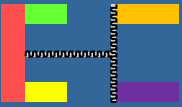


# READ ONLY MEMORY

Memorias

Semiconductoras



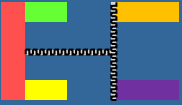


# RANDOM ACCESS MEMORY

Memorias  
Semiconductoras

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

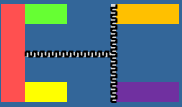
- ⊙ RAM estática (SRAM). El elemento de almacenamiento es un biestable.
- ⊙ SRAM asíncrona. Su funcionamiento no está ajustado a un reloj. En la operación de lectura la palabra direccionada se encuentra en la salida; en la escritura la palabra se almacena en la dirección indicada
- ⊙ SRAM síncrona de ráfaga. Su funcionamiento se ajusta al reloj del sistema, y en su interior los registros se encuentran sincronizados con el reloj.



# RANDOM ACCESS MEMORY

Memorias  
Semiconductoras

- ◎ RAM dinámica (DRAM)
  - ◎ El elemento de almacenamiento es un condensador. Debido a que tiende a descargarse es necesario refrescarlas periódicamente.
  - ◎ Son más lentas que las SRAM
  - ◎ Mayor densidad de integración -> Mayor capacidad que SRAM



## ⊙ Tipos de DRAM

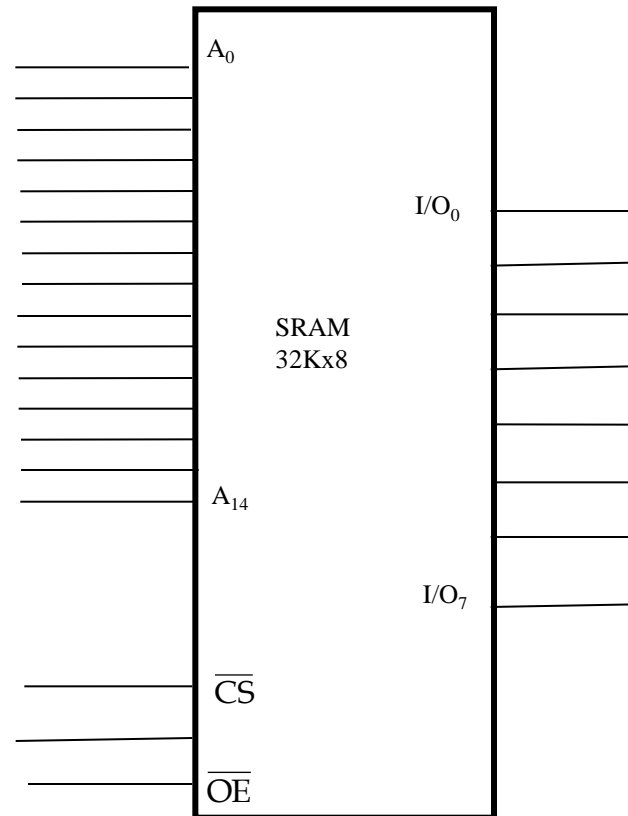
- ⊙ Asíncronas: FPM DRAM, EDO DRAM, BEDO DRAM.
- ⊙ Síncronas (opera a la velocidad del bus sin imponer estados de espera) SDRAM (Synchronous DRAM)
  - SDR SDRAM (Single Data Rate DRAM)
  - DDR SDRAM (Double Data Rate DRAM). Transfiere la información tanto en el flanco de bajada como en el flanco de subida.
  - DDR2 SDRAM. Durante cada ciclo de reloj se realicen cuatro transferencias.
  - DDR3 SDRAM. Menos consumo, más frecuencia
  - DDR4 SDRAM.....
  - QDR SDRAM (Quad Data Rate DRAM)

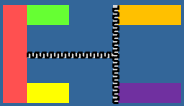




# RANDOM ACCESS MEMORY

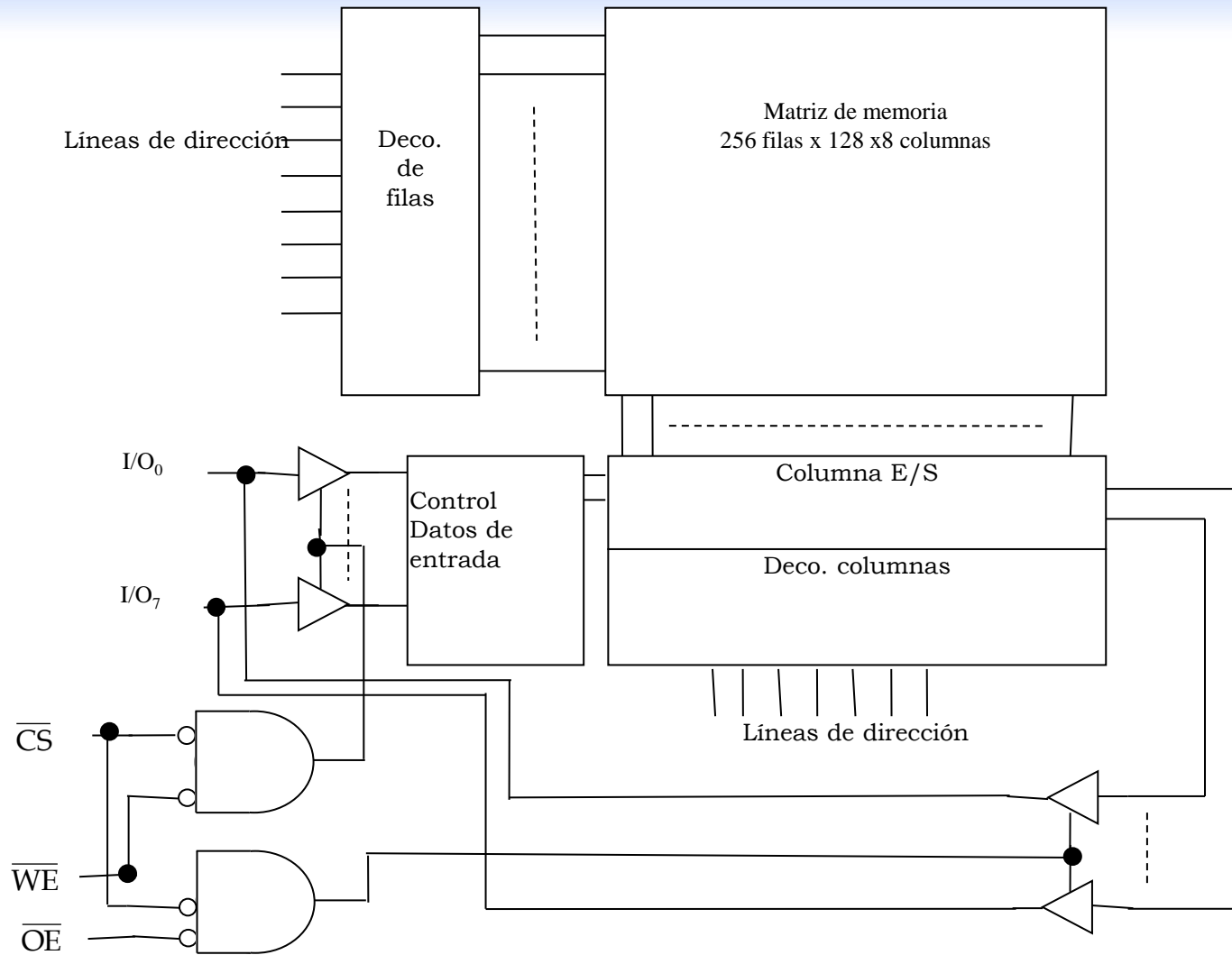
Memorias  
Semiconductoras



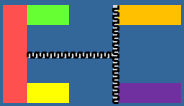


# RANDOM ACCESS MEMORY

Memorias  
Semiconductoras

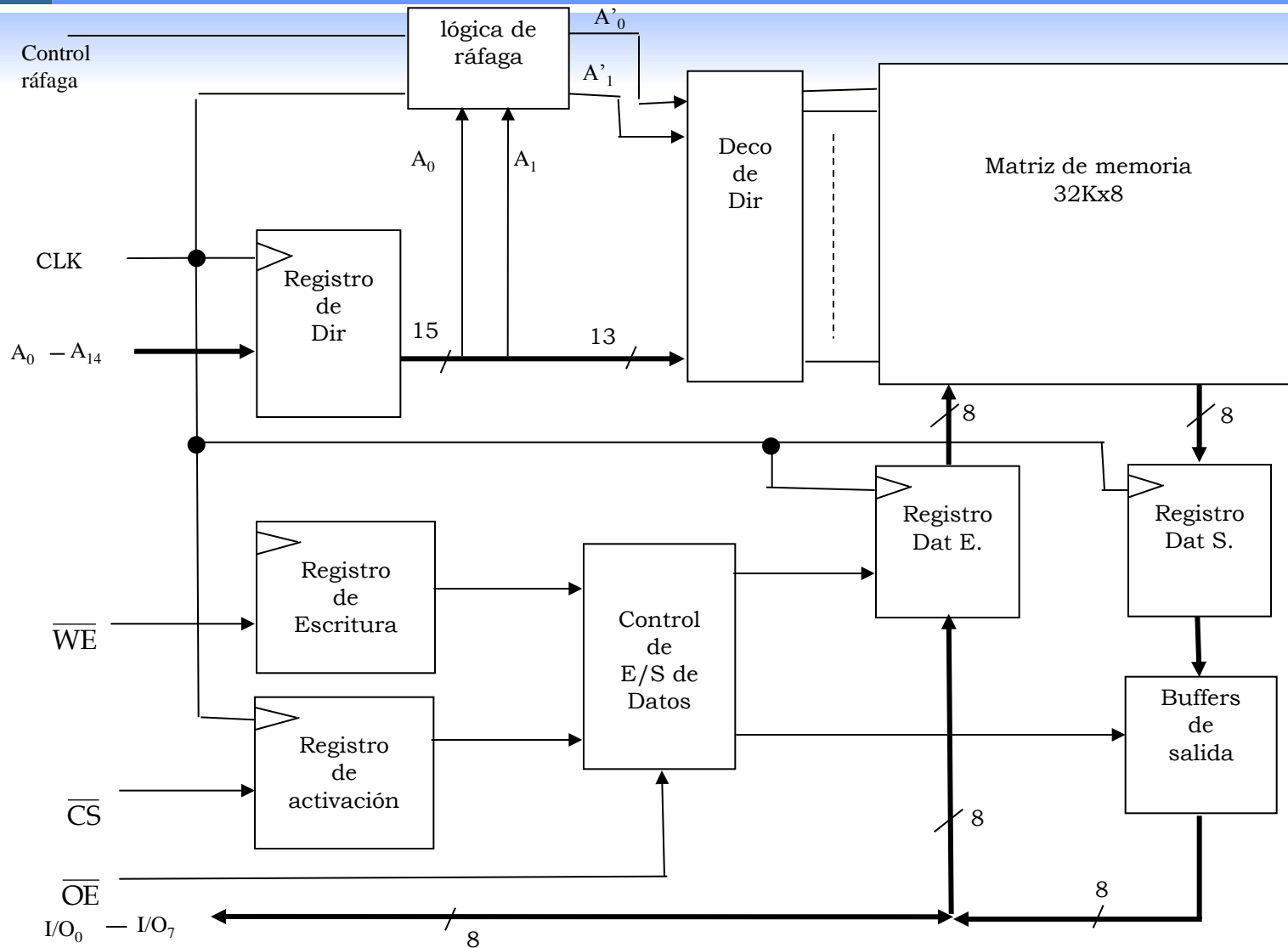


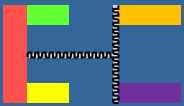




# RANDOM ACCESS MEMORY

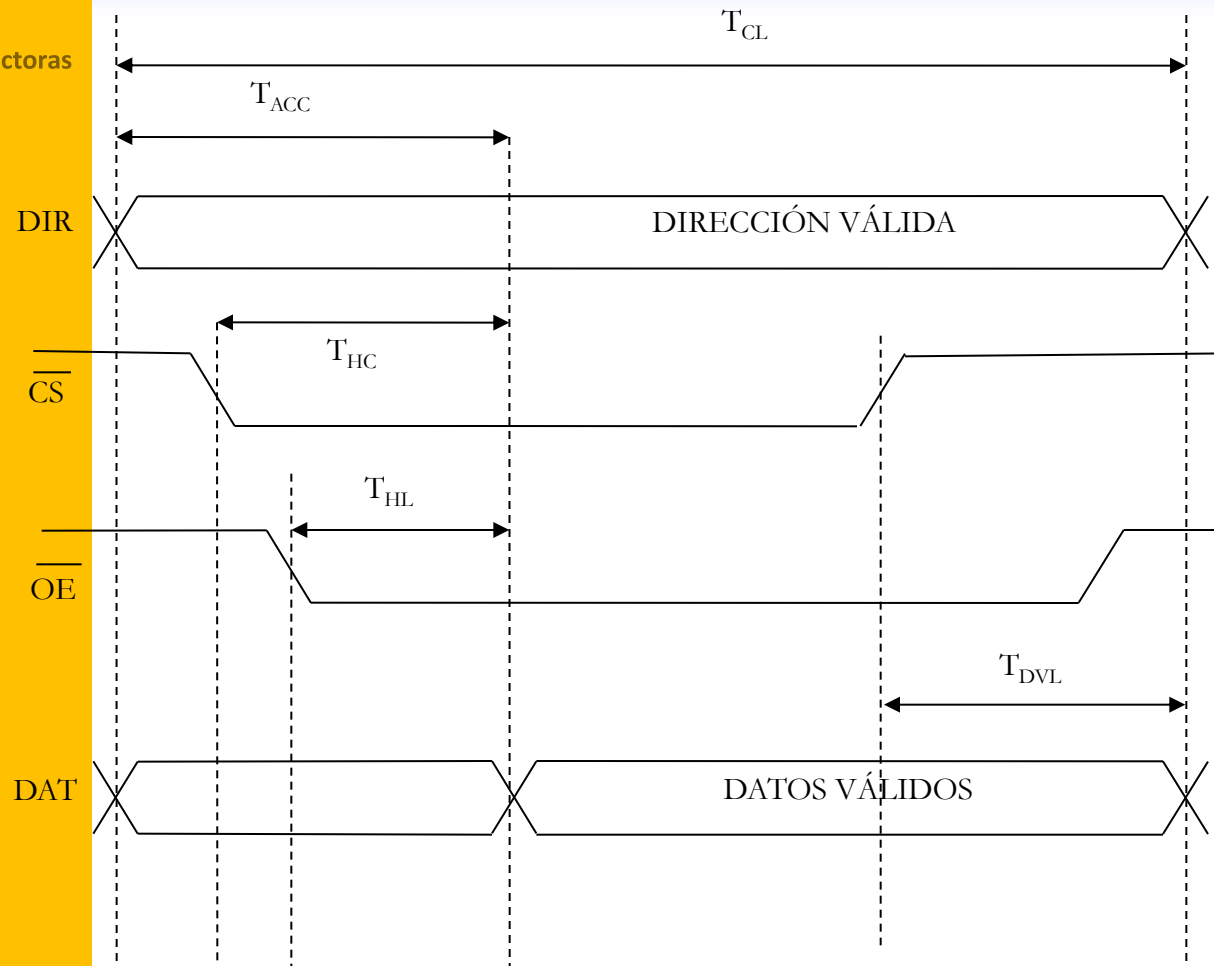
Memorias  
Semiconductoras





# OPERACIÓN DE LECTURA

## Memorias Semiconductoras



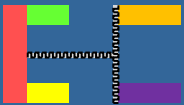
$T_{CL}$  Tiempo de ciclo de lectura memoria

$T_{ACC}$  Tiempo de acceso de acceso. Tiempo que transcurre desde el principio de una dirección válida hasta que los datos válidos aparecen en el B. Dat

$T_{HC}$  Tiempo de acceso de habilitación de chip. Se mide desde que cambia CS hasta que aparecen datos válidos

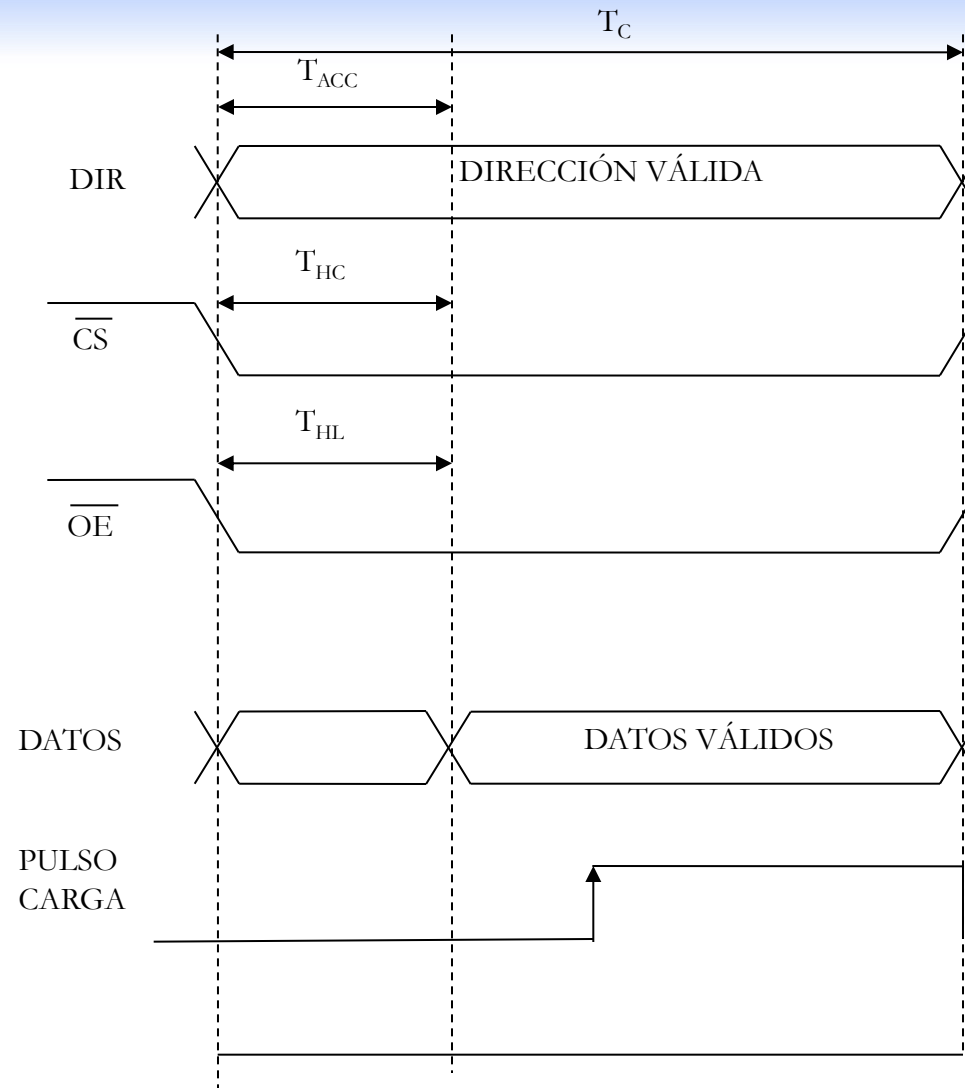
$T_{HL}$  Tiempo de acceso de habilitación de lectura. Tiempo que transcurre desde que se aplica OE, hasta que aparecen datos válidos

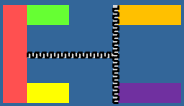
$T_{DVL}$  Tiempo en el que los datos se mantienen válidos aunque se haya deshabilitado el chip



# OPERACIÓN DE LECTURA

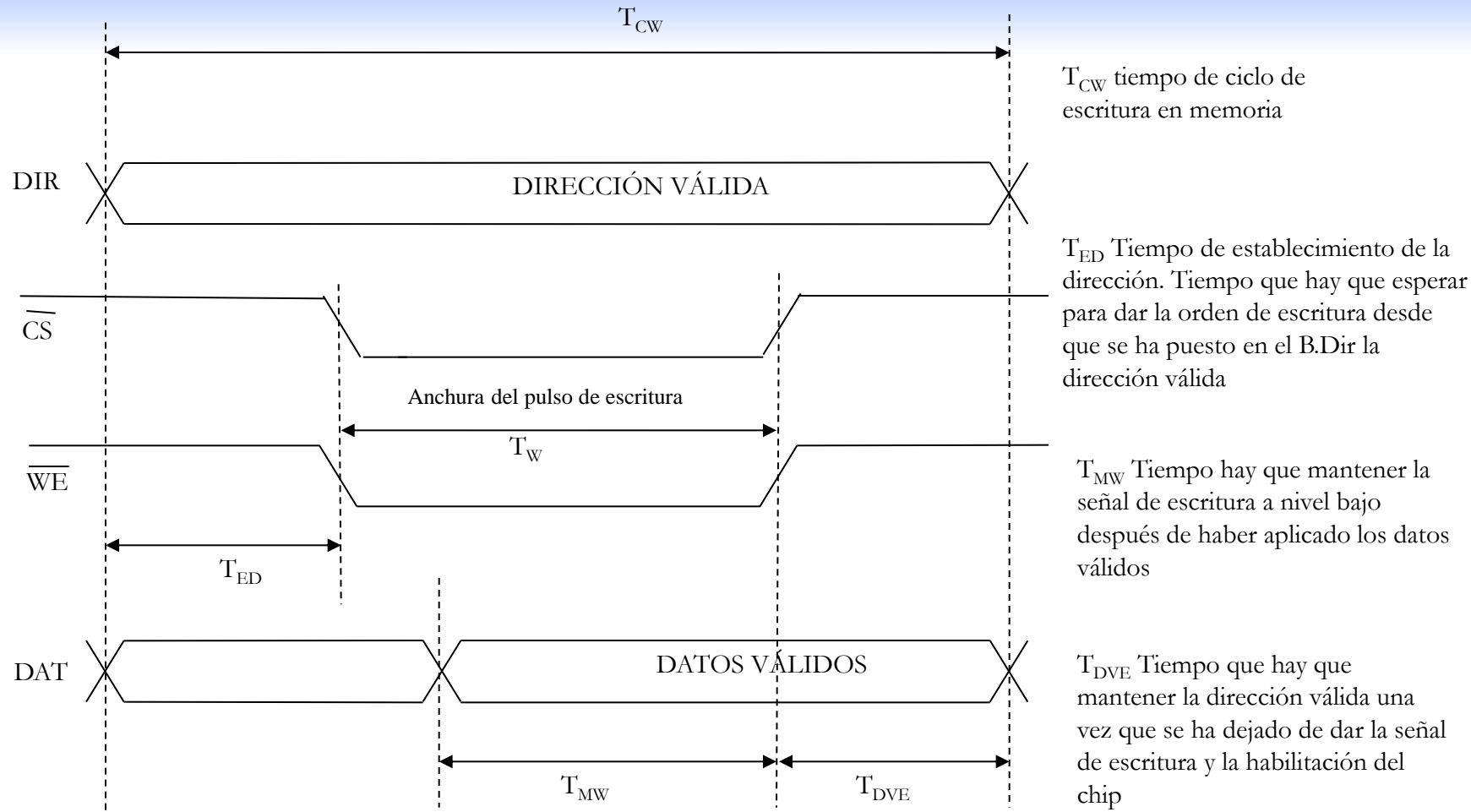
Memorias  
Semiconductoras

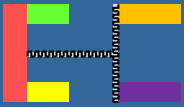




# OPERACIÓN DE ESCRITURA

Memorias  
Semiconductoras



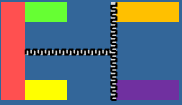


# TIPOS DE MEMORIAS

Memorias  
Semiconductoras

## Tabla resumen

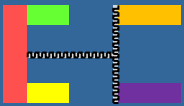
Tipo	Clase	Borrado	Escritura	Volatilidad
RAM	Lectura/Escritura	Eléctricamente por bytes	Eléctricamente	Volátil
ROM	Sólo lectura	No	Mediante máscaras	No Volátil
PROM			Eléctricamente	
EPROM	Luz violeta, chip completo			
FLASH	Eléctricamente por bloques			
EEPROM	Eléctricamente por byte			



# INTRODUCCIÓN AL DISEÑO

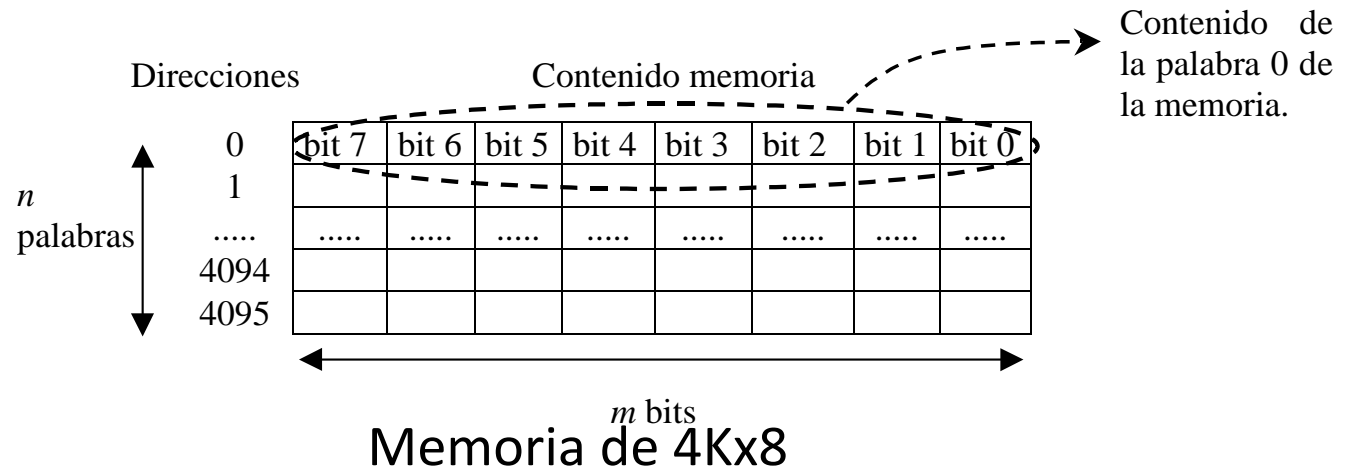
Diseño de  
mapas de  
memoria

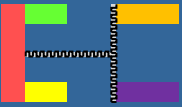
- ◎ Diseño de la unidad de memoria principal
  - ◎ Mapa de memoria. Determina la memoria direccionable y el tipo de memoria.
  - ◎ Diseño de la organización de los chips de memoria y su interconexión
  - ◎ Diseño de la interfaz hardware procesador-memoria



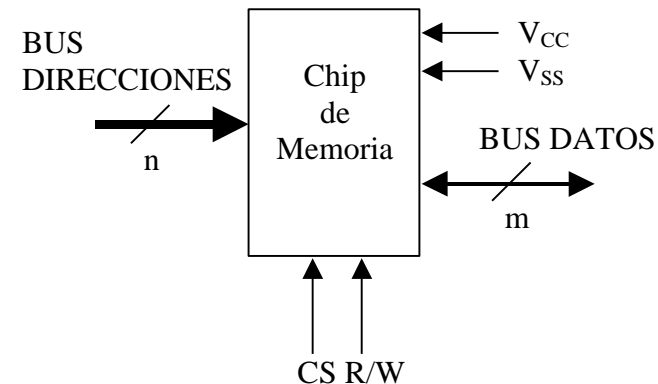
## 🎯 Chip de memoria

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.

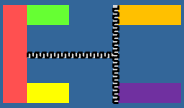




- La interconexión de un chip de memoria se realiza a través de sus patillas:
  - $n$  patillas para el bus de direcciones, donde se podrá direccionar  $2^n$  palabras.
  - $m$  patillas para el bus de datos indicando que en cada acceso se trabajará con  $m$  bits.
  - W/R (Write/Read). Esta patilla 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.
  - VCC. Alimentación del chip.
  - VSS. Conexión a tierra.



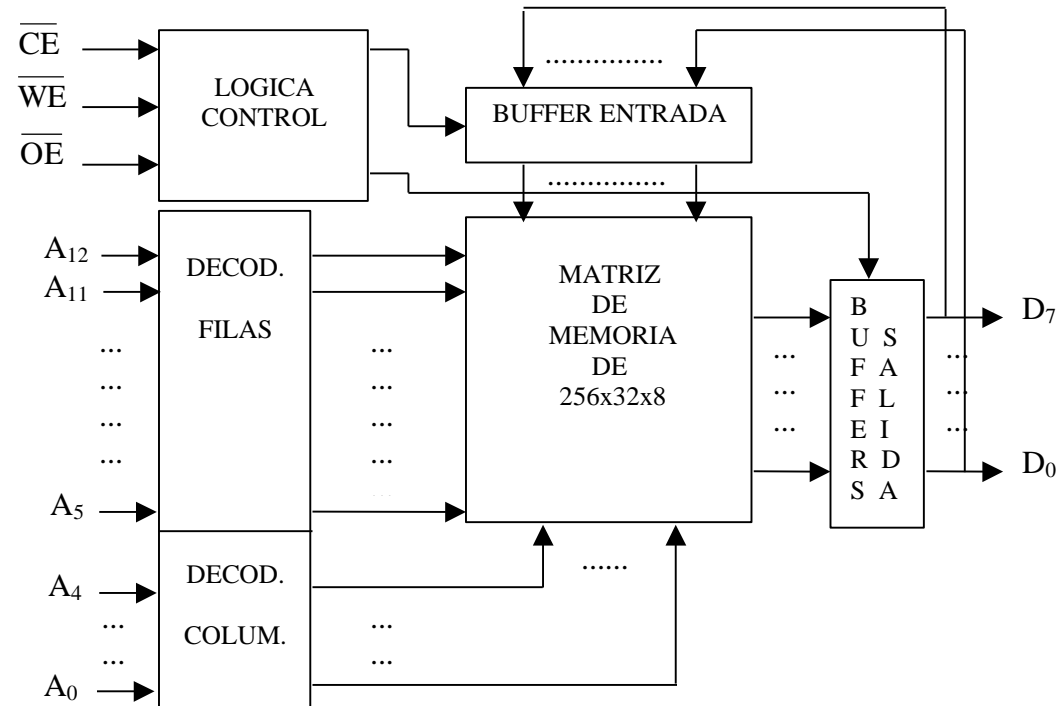




# INTRODUCCIÓN AL DISEÑO

## Diseño de mapas de memoria

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





## Mapa de memoria

Espacio que puede direccionar un computador.

Direcciones  
en decimal

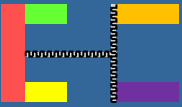
0  
1  
....  
....  
 $2^{16}-1$


Direcciones  
en hexadecimal

0000  
  
  
  
FFFF

▲  
64 K  
▼

Ejemplo de un computador con bus de 16 bits.



# MAPA DE MEMORIA

## Diseño de mapas de memoria

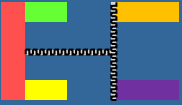
- ⊙ La implementación física del mapa de memoria se realiza utilizando uno o varios chips de memoria.
- ⊙ En el mercado se encuentran diferentes configuraciones de chips de memoria:
  - ⊙  $zK \times 1$ ,  $zK \times 4$ ,  $zK \times 8$ ,  $zK \times 16$ ,  $zK \times 32$ ,  $zM \times 1$ ,  $zM \times 4$ ,  $zM \times 8$ ,  $zM \times 16$ ,  $zM \times 32$ , etc. donde  $z$  es un múltiplo de 2.
  - ⊙ Así, por ejemplo, un chip de  $1K \times 8$  indica que puede almacenar 1024 palabras de 8 bits cada una.



# DISEÑO - EJEMPLO 1

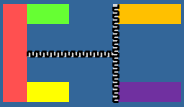
Diseño de  
mapas de  
memoria

- ◎ 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?.



## **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.

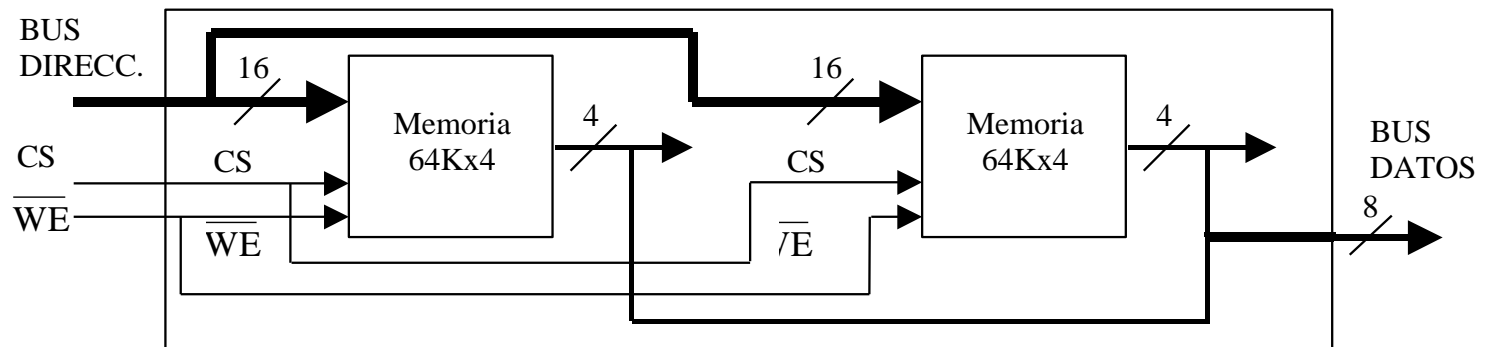


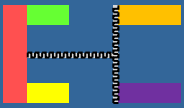
# DISEÑO - EJEMPLO 1

Diseño de  
mapas de  
memoria

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.



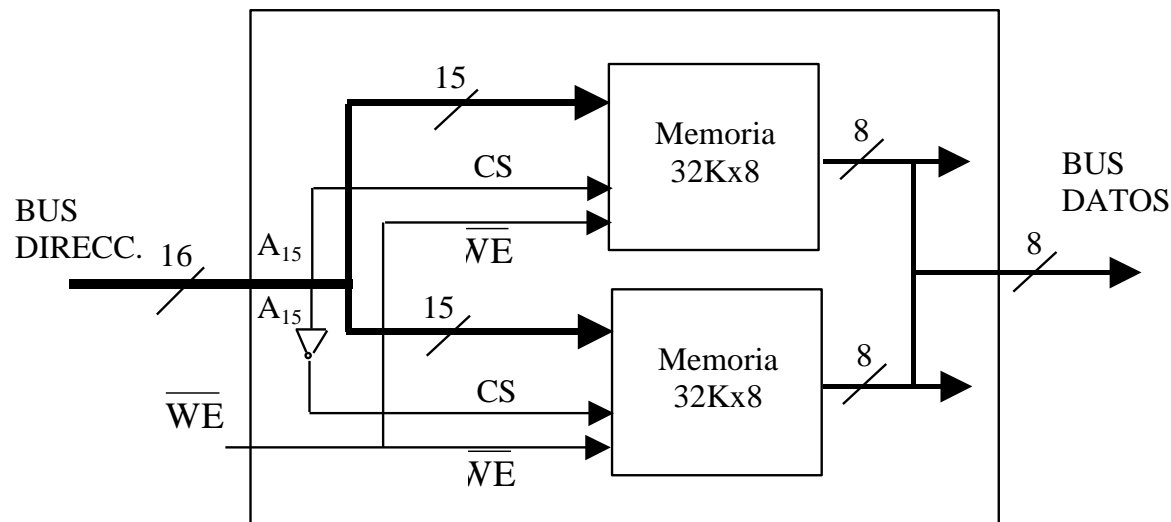


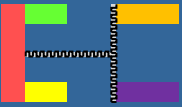
# DISEÑO - EJEMPLO 1

Diseño de  
mapas de  
memoria

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.



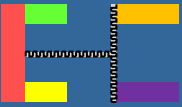


## DISEÑO - EJEMPLO 2

### Diseño de mapas de memoria

- ⊙ 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.



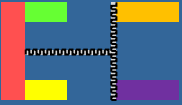


- (1) Debemos obtener el número de bits del bus de direcciones.

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

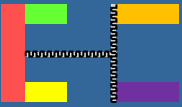
- (2) Averiguar el número de bits que se necesitan para direccionar el chip de memoria que vamos a emplear.

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,  $A_{15}A_{14}...A_1A_0$ .



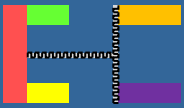
(3) Calcular el número de chips que necesitamos.

Como queremos  $128K \times 16$  necesitaremos 16 chips para obtener una palabra al completo (16 bits). Con estos primeros 16 bits tenemos  $64K \times 16$ , por lo que nos falta otros  $64K \times 16$  más, es decir, 16 chips más. Por tanto, necesitaremos 32 chips de  $64K \times 1$  para almacenar  $128K \times 16$ .



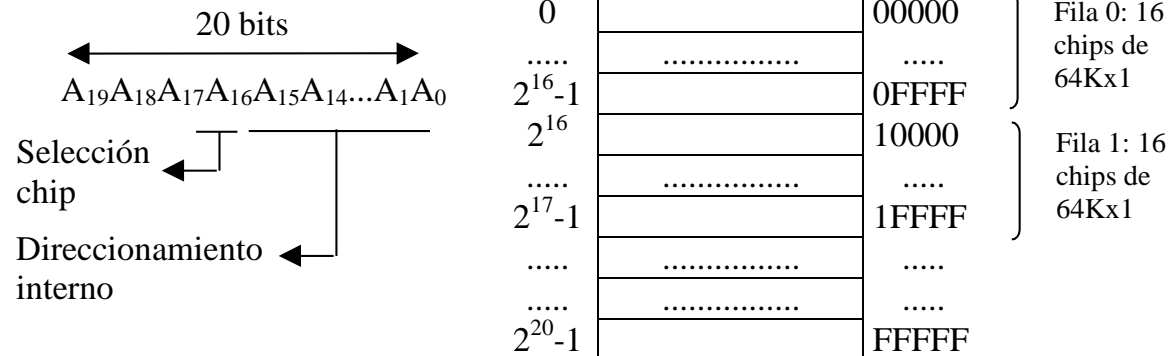
(4) Obtener el número de bits del bus de direcciones que permita seleccionar los chips de memoria.

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.



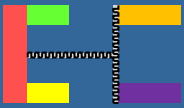
## DISEÑO - EJEMPLO 2

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



Mapa de Memoria

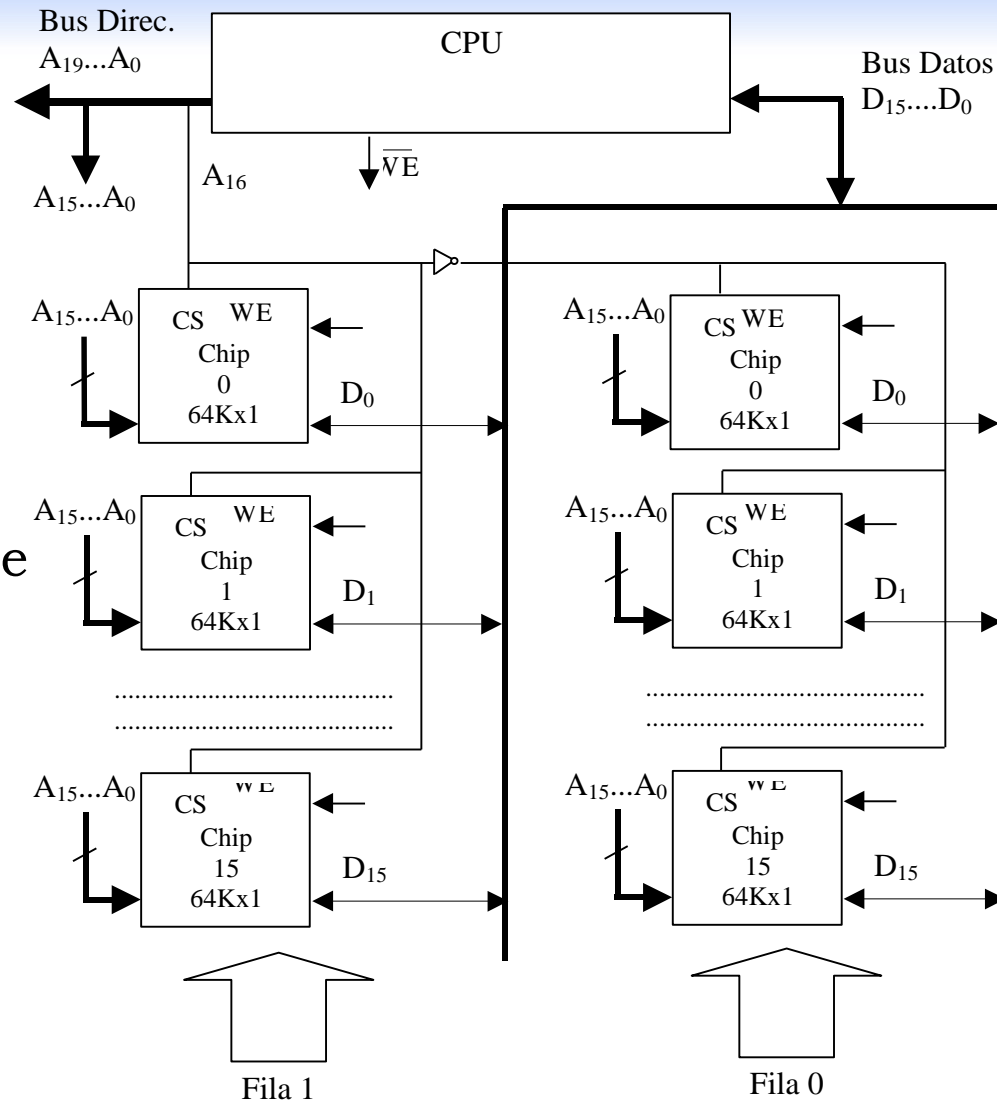
A <sub>19</sub>	A <sub>18</sub>	A <sub>17</sub>	A <sub>16</sub>	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Fila 0: 16 chips de 64Kx1
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	Fila 1: 16 chips de 64Kx1
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	

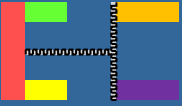


## DISEÑO - EJEMPLO 2

Diseño de  
mapas de  
memoria

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

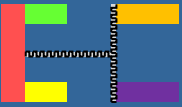




## DISEÑO - EJEMPLO 3

### Diseño de mapas de memoria

- Se dispone de dos módulos de memoria de 256Kx8 de capacidad cada uno, que poseen las señales de control  $\overline{CS}$  para la habilitación y  $R/\overline{W}$  para la de lectura/escritura. Estos módulos se necesitan ubicar en un mapa de memoria de un procesador de 20 líneas en el bus de direcciones y 8 en el de datos. El primer módulo hay que colocarlo al principio del mapa, es decir, en las primeras 256K posiciones, mientras que el segundo hay que colocarlo en las últimas 256K posiciones. Diseñar el esquema de conexionado correspondiente.



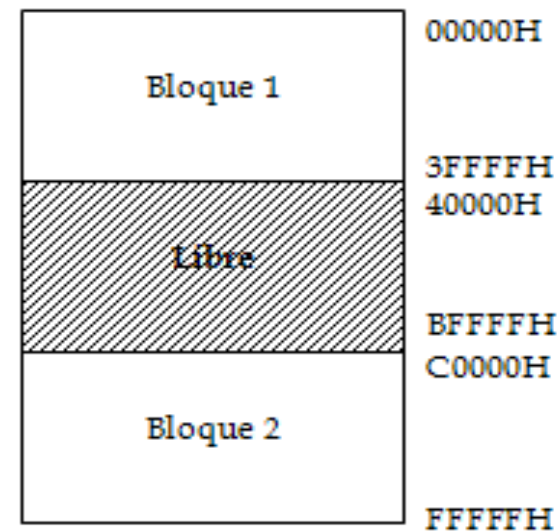
## DISEÑO - EJEMPLO 3

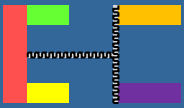
### Diseño de mapas de memoria

El enunciado del problema nos proporciona la siguiente información:

1. Que el procesador es capaz de direccionar  $2^{20} = 1\text{M}$  posiciones de 8 bits, es decir,  $1\text{M} \times 8$
2. Que el procesador posee 20 líneas de direcciones, A0 a A19
3. Que las memorias por ser de 256K necesitan 18 líneas de direcciones,  
 $2^x = 256\text{K} \rightarrow x=18$ , es decir, de A0 a A17

El mapa de memoria pedido es





## DISEÑO - EJEMPLO 3

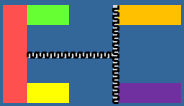
### Diseño de mapas de memoria

- Con un decodificador de 2 a 4 activo por cero tenemos el sistema resuelto. Las entradas al decodificador serán A19A18 y las salidas que debemos conectar a los dos bloques serán la salida cero para el bloque 1 y la salida 3 para el bloque 2

A <sub>19</sub>	A <sub>18</sub>	A <sub>17</sub>	A <sub>16</sub>	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Bloque	Dirección	Cap.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00000H	256K
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		3FFFFH	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LIBRE	40000H	256K
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		7FFFFH	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LIBRE	80000H	256K
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		BFFFFH	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	C0000H	256K
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		FFFFFH	

- Por lo tanto el esquema eléctrico queda como sigue:

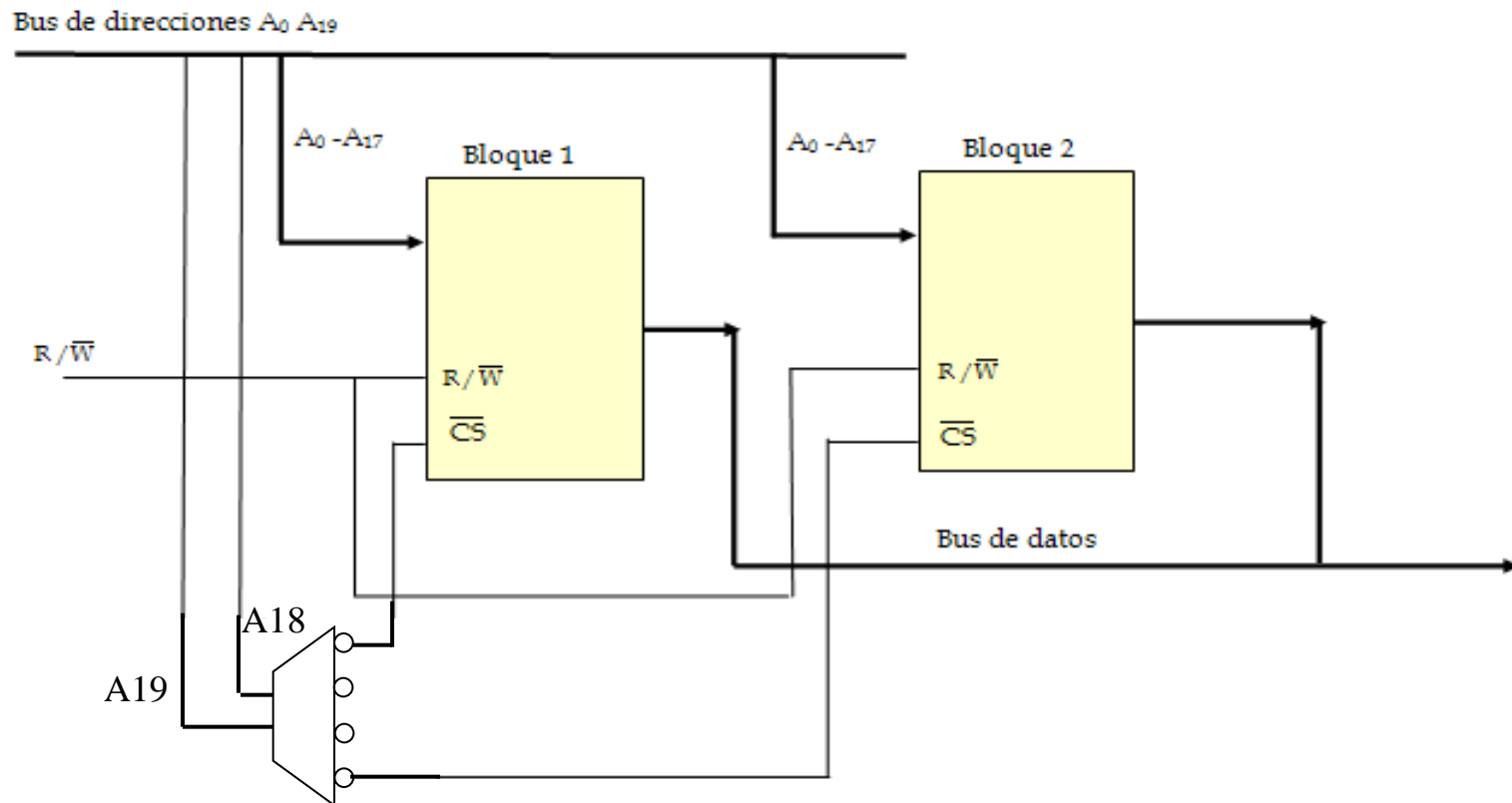


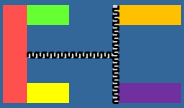


# DISEÑO - EJEMPLO 3

Diseño de  
mapas de  
memoria

## Esquema eléctrico





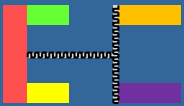
## DISEÑO - EJEMPLO 3

### Diseño de mapas de memoria

- ⊙ Necesitamos las líneas A0 a A17 para direccionar las memorias, por lo tanto con las dos restantes podemos realizar la decodificación, teniendo en cuenta que las memorias se activan por cero

A <sub>19</sub>	A <sub>18</sub>	A <sub>17</sub>	A <sub>16</sub>	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Bloque	Dirección	Cap.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00000H	256K
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		3FFFFH	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LIBRE	40000H	256K
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		7FFFFH	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LIBRE	80000H	256K
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		BFFFFH	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	C0000H	256K
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		FFFFFH	

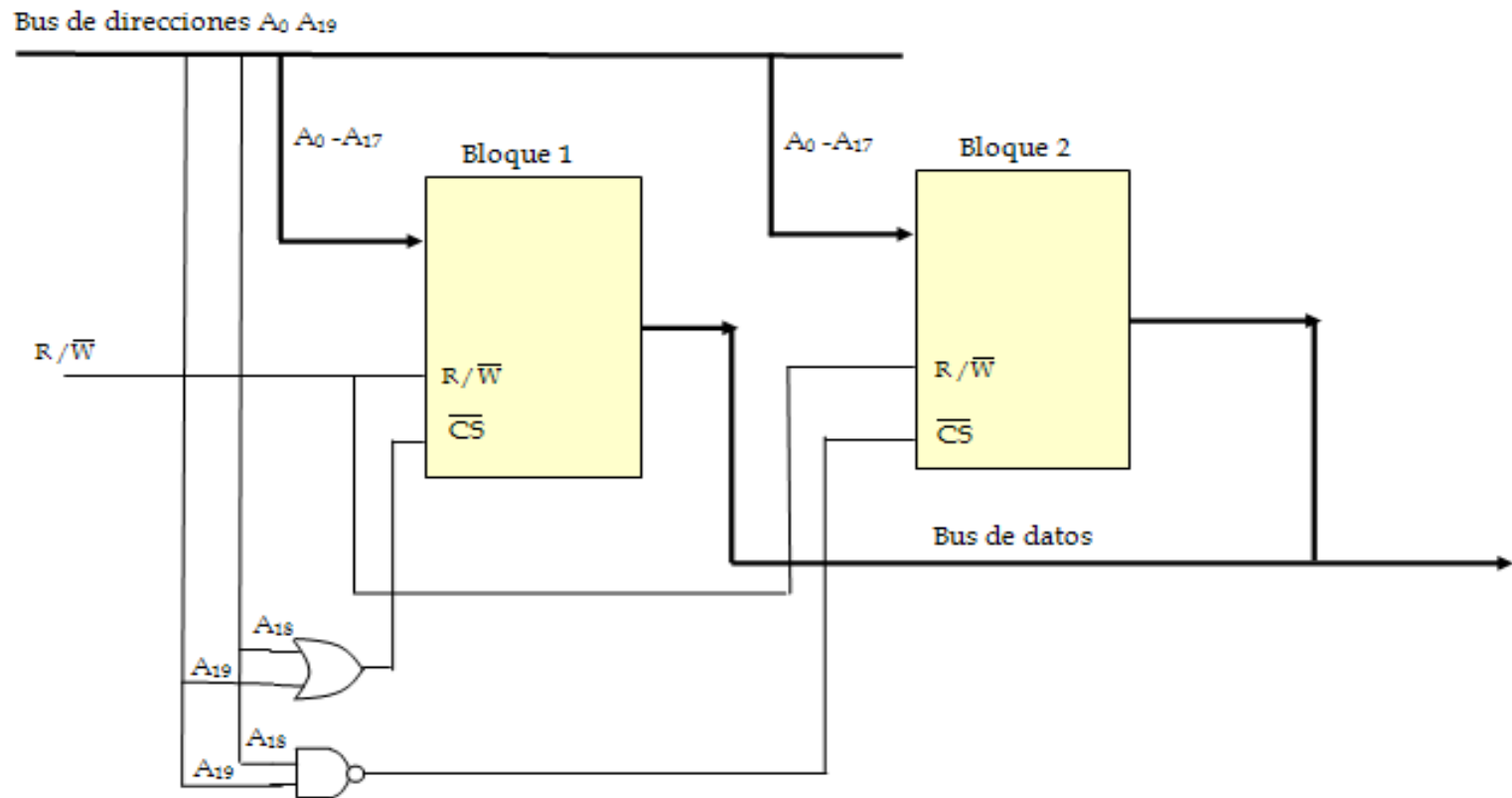
- ⊙ El bloque 1 se debe activar cuando los bits A18 y A19 sean cero, quiere esto decir que su línea  $\overline{CS}_1$  se debe poner a cero en dicho caso, se consigue mediante la función  $\overline{CS}_1 = A_{18} + A_{19}$   
sin embargo el bloque 2, se controla mediante la función:  $\overline{CS}_2 = \overline{A_{18}} \cdot \overline{A_{19}}$

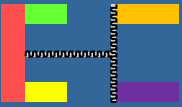


# DISEÑO - EJEMPLO 3

Diseño de  
mapas de  
memoria

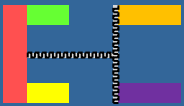
## Esquema eléctrico



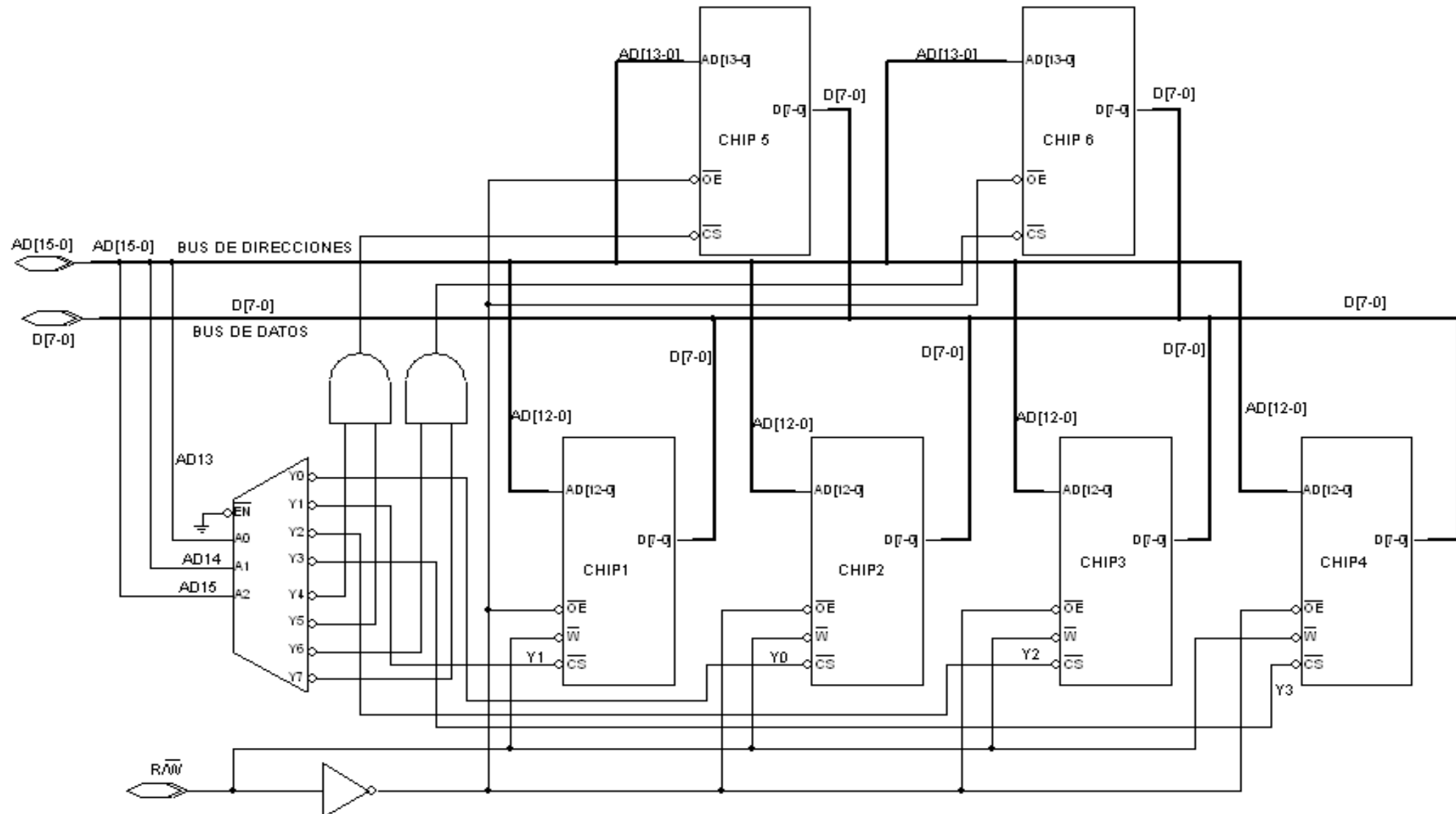


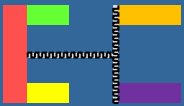
Ha caído en nuestras manos el esquema de la memoria que utiliza el nuevo sistema de codificación de la información patentado por MaNoTas S.A., pero necesitamos averiguar las siguientes cosas:

- a) Capacidad de cada memoria medida en Kbits y en Kbytes.
- b) Direcciones de inicio y final de los chip 1 al 6.
- c) Tipo de memoria utilizada en cada chip.
- d) Tabla de funcionamiento de cada tipo de memoria.
- e) Modificar el diseño para que se puedan unificar las memorias en dos: una que sustituya a los chips 5 y 6 que empiece en la dirección 0000H y la otra que sustituya a los chips 1, 2, 3 y 4 y que empiece en la dirección 8000H.



# DISEÑO - EJEMPLO 4



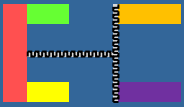


## DISEÑO - EJEMPLO 4

### Diseño de mapas de memoria

- Para establecer las direcciones de inicio y final de cada memoria, conviene realizar una tabla que reúna las siguientes columnas: las líneas del bus de direcciones, separadas por el número de bits de cada memoria, la salida del decodificador que se activa, la dirección de inicio y final, la capacidad y el tipo de memoria.

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Deco	Chip	Dirección	Cap.	Tipo
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>0</sub>	2	0000H	8K	RAM
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1			1FFFH		
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>1</sub>	1	2000H	8K	RAM
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFFH		
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>2</sub>	3	4000H	8K	RAM
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1			5FFFH		
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>3</sub>	4	6000H	8K	RAM
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFFH		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>4</sub>	5	8000H	16K	ROM
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Y <sub>5</sub>		BFFFH		
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>6</sub>	6	C000H	16K	ROM
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Y <sub>7</sub>		FFFFH		

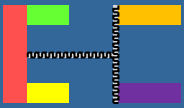


- Las memorias 1, 2, 3 y 4 son de lectura escritura, ya que poseen las líneas y por lo tanto son de las llamadas RAM. Su tabla de funcionamiento es:

$\overline{CS}$	$\overline{WE}$	$\overline{OE}$	E/S
H	X	X	Alta impedancia
L	H	L	Lectura
L	L	X	Escritura

- Las memorias 5 y 6 son sólo de lectura, su patilla para poder leer es la , por lo tanto son de las llamadas ROM. Su tabla de funcionamiento es:

$\overline{CS}$	$\overline{OE}$	E/S
H	X	Alta impedancia
L	L	Lectura



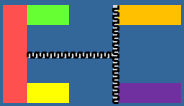
## DISEÑO - EJEMPLO 4

### Diseño de mapas de memoria

- Por último, si queremos unificar las memorias en dos, por un lado una que sea RAM de 32K, y otra que sea ROM de 32K debemos utilizar la tabla del mapa de memoria y tendremos en cuenta que para direccional 32KBytes, necesitamos 15 bits, A0 ,A14 y por lo tanto con el bit que sobra, A15 ,podremos elegir cada una de las dos memorias utilizando un simple inversor y teniendo en cuenta que la ROM debe empezar en la dirección 0000H

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Deco	Chip	Dirección	Cap.	Tipo
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>0</sub>	1	0000H	32K	ROM
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFFH		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y <sub>1</sub>	2	8000H	32K	RAM
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFFH		

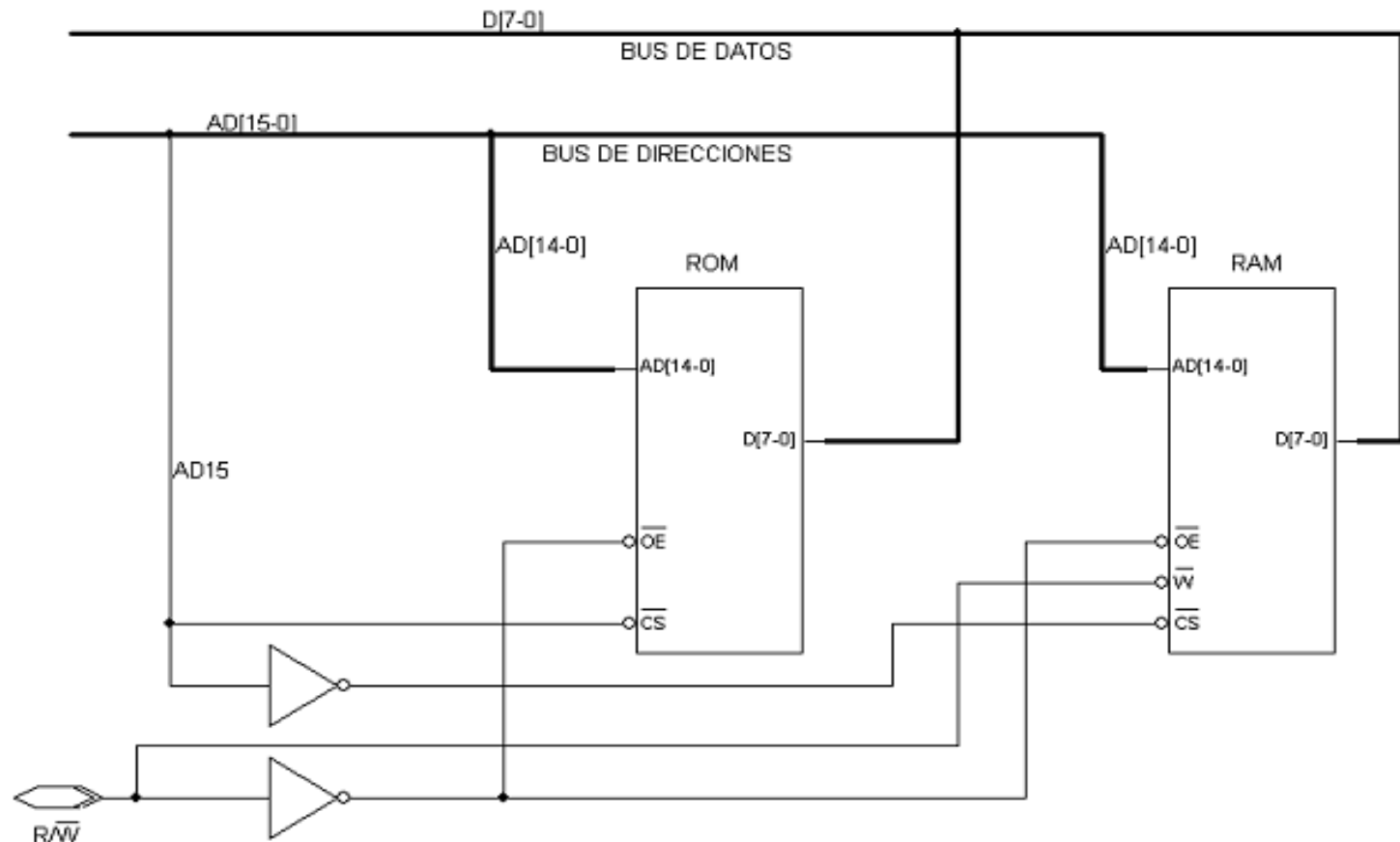




# DISEÑO - EJEMPLO 4

Diseño de  
mapas de  
memoria

🎯 El esquema eléctrico sería el siguiente

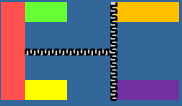




# MEMORIAS ASOCIATIVAS

## Otras memorias

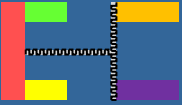
- ⊙ Una memoria asociativa se caracteriza por el hecho de que la posición de memoria a la que se desea acceder se realiza especificando su contenido o parte de él y no por su dirección.
- ⊙ A las memorias asociativas también se les denominan memorias direccionables por contenido (**CAM**: Content Addressable Memory).



# ESTRUCTURA DE UNA CAM

## Otras memorias

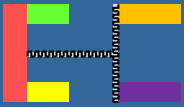
- ⊙ Una memoria asociativa consiste en un conjunto de registros y una matriz de celdas de memoria, con su lógica asociada, organizada en  $n$  palabras con  $m$  bits/palabra.
- ⊙ El conjunto de registros está formado por un registro argumento (A) de  $m$  bits, un registro máscara (K) de  $m$  bits y un registro de marca (M) de  $n$  bits.



# ESTRUCTURA DE UNA CAM

## Otras memorias

- ⊙ 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

## Otras memorias

- 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,  $M_i=0$ .

0	Alicante	0
---	----------	---

Registro argumento

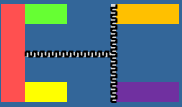
0	1	0
---	---	---

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

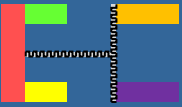
Registro  
de marca



# ESTRUCTURA DE UNA CAM

## Otras memorias

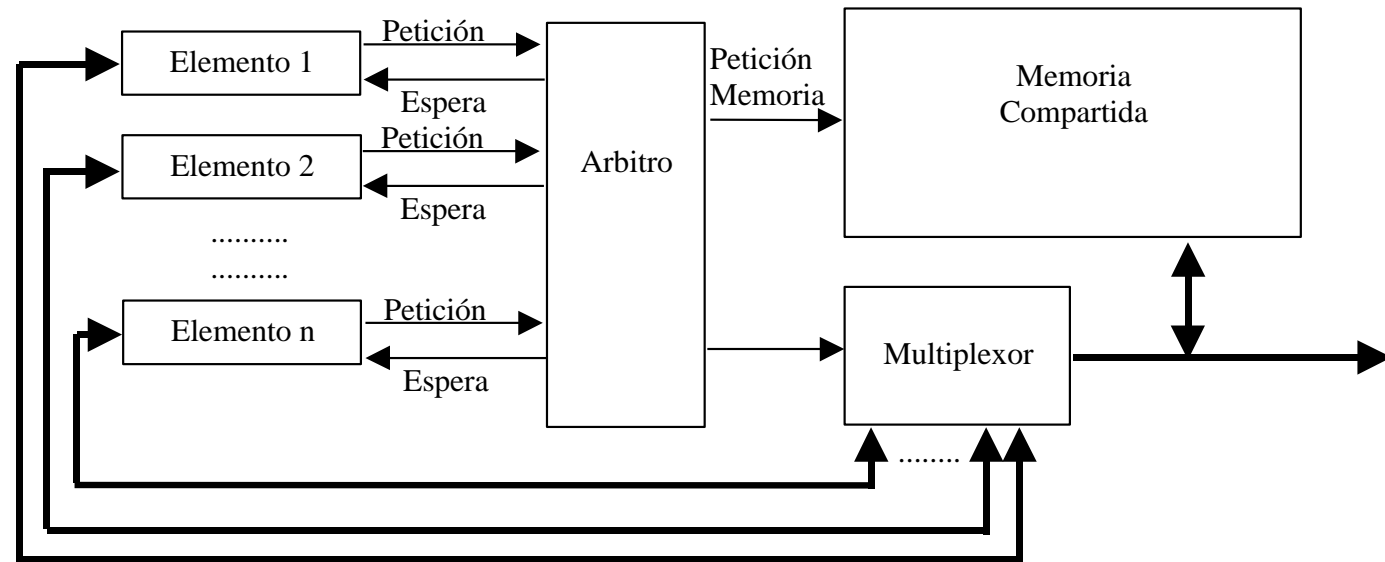
- ⊙ Por regla general, en la mayoría de las aplicaciones la memoria asociativa almacena una tabla que no tiene, para una máscara dada, dos filas iguales.
- ⊙ Las memorias asociativas se utilizan sobre todo con memorias cache de tal forma que la identificación de la etiqueta de cada línea se realice de 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.



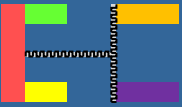
# MEMORIAS COMPARTIDAS

## Otras memorias

- ⊙ 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.



# MEMORIAS COMPARTIDAS

## Otras memorias

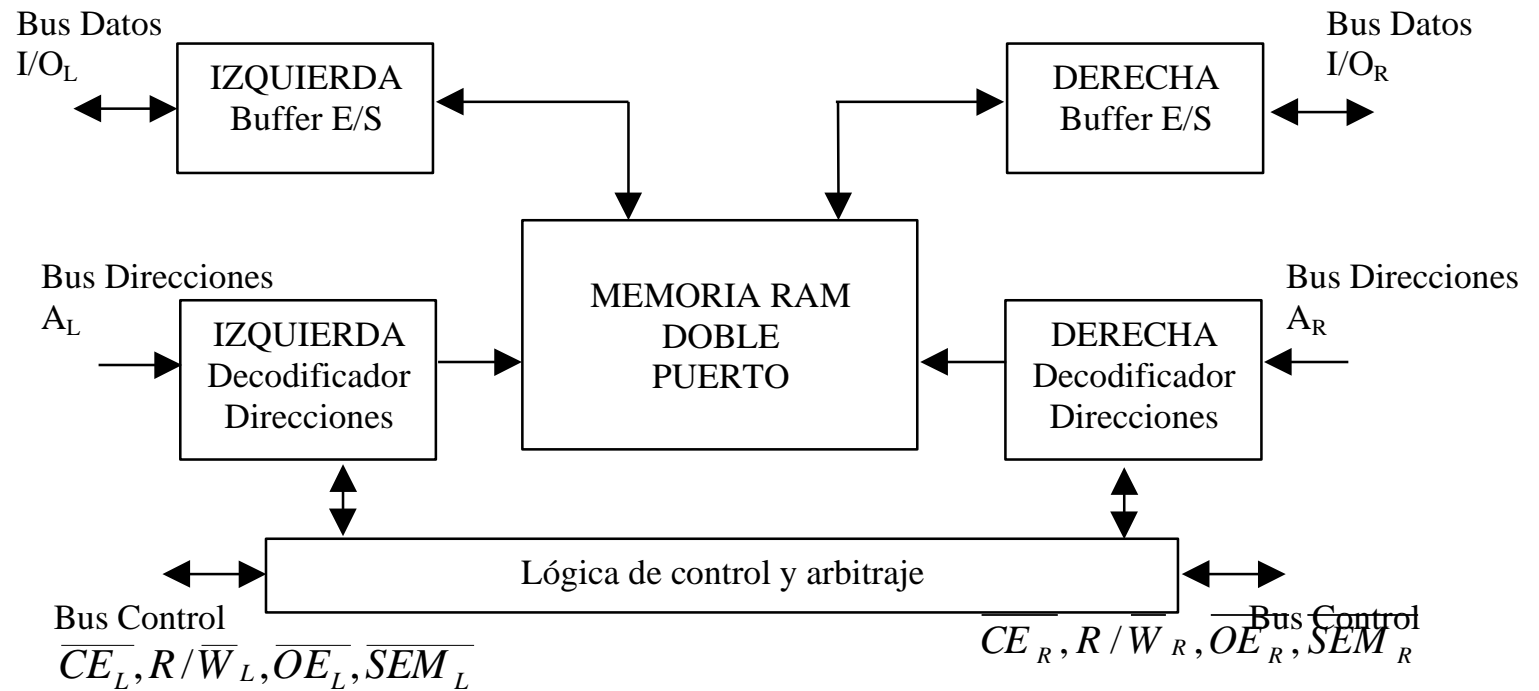
- ⊙ El árbitro se diseña de forma que asigne un tiempo de servicio, en promedio, análogo a todas la unidades que solicitan el recurso.
- ⊙ Existen diferentes estrategias:
  - ⊙ Asignación de la menor prioridad al elemento servido.
  - ⊙ Rotación de prioridades. En un estado cualquiera, el próximo estado se calcula rotando el orden de prioridades actual hasta que el elemento al que se acaba de dar servicio tiene la menor prioridad.



# MC - MEMORIAS DE DOBLE PUERTO

## Otras memorias

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





# MC - MEMORIAS DE DOBLE PUERTO

## Otras memorias

- 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.