

# TEMA 3.

## UNIDAD DE MEMORIA

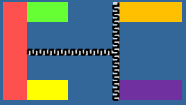
dtic





## Índice

- ⊙ Conceptos básicos
  - ⊙ Definiciones
  - ⊙ Características de las memorias
  - ⊙ Jerarquía de las memorias
- ⊙ Tipos de memorias
- ⊙ Memoria Caché
- ⊙ Diseño de mapas de memoria



# CONCEPTOS BÁSICOS: DEFINICIONES

## Conceptos básicos

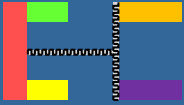
- ⊙ La memoria es necesaria para albergar los programas que se ejecutan en el computador y los datos sobre los que trabajan dichos programas.
- ⊙ La memoria presenta una gran diversidad de tipos, tecnologías, estructuras, prestaciones y costes.
- ⊙ Un solo tipo NO ES SUFICIENTE para cubrir necesidades → Jerarquía de memoria.
- ⊙ Operaciones: Lectura y/o Escritura.
- ⊙ Al menos, necesitan un mecanismo de direccionamiento (bus de direcciones) y otro para el movimiento de datos (bus de datos).
- ⊙ Un computador dispone de una jerarquía de elementos de memoria cuya localización es diversa: algunos están localizados internamente en la CPU (p. ej. caché), otros externamente (p. ej. RAM), etc.



# CONCEPTOS BÁSICOS: CARACTERÍSTICAS

## Conceptos básicos

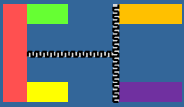
- ⊙ **Ubicación:**
  - ⊙ Procesador
  - ⊙ Interna (principal)
  - ⊙ Externa (secundaria)
- ⊙ **Capacidad:**
  - ⊙ Tamaño de palabra
  - ⊙ Número de palabras
- ⊙ **Unidad de transferencia:**
  - ⊙ Palabra
  - ⊙ Bloque
- ⊙ **Método de acceso:**
  - ⊙ Secuencial: El acceso se realiza en secuencia lineal (p. ej. cintas)
  - ⊙ Directo: Acceso a vecindad + lectura secuencial (p. ej. unidades de disco)
  - ⊙ Aleatorio: Tiempo de acceso constante e indepen. del anterior (RAM)
  - ⊙ Asociativo: Acceso por contenido (p. ej. algunas caché)



# CONCEPTOS BÁSICOS: CARACTERÍSTICAS

## Conceptos básicos

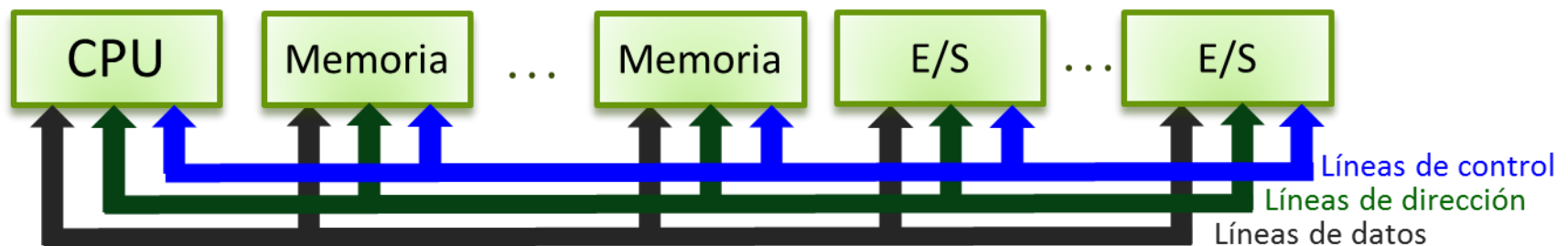
- ⊙ **Rendimiento:**
  - ⊙ Tiempo de acceso
  - ⊙ Tiempo de ciclo
  - ⊙ Velocidad de transferencia
- ⊙ **Tecnología:**
  - ⊙ Semiconductor
  - ⊙ Soporte magnético
  - ⊙ Soporte óptico
  - ⊙ Magneto-óptico
- ⊙ **Persistencia:**
  - ⊙ Volátil/no volátil (ej. RAM/ROM)
  - ⊙ Borrable/no borrable (ej. PROM/ROM)
- ⊙ **Interfaz con el exterior:**
  - ⊙ Síncronas: necesitan señal de reloj para su funcionamiento
  - ⊙ Asíncronas: no necesitan señal de reloj
- ⊙ **Organización:** Disposición o estructura física en bits para formar palabras



# CONCEPTOS BÁSICOS: COMUNICACIÓN. BUSES

## 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.

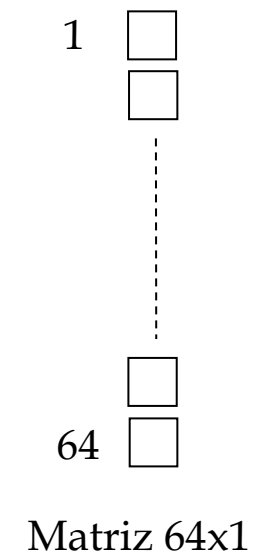
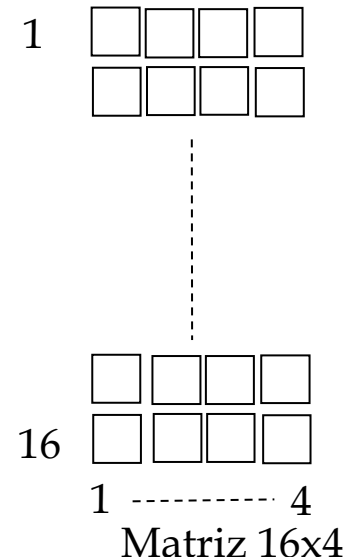
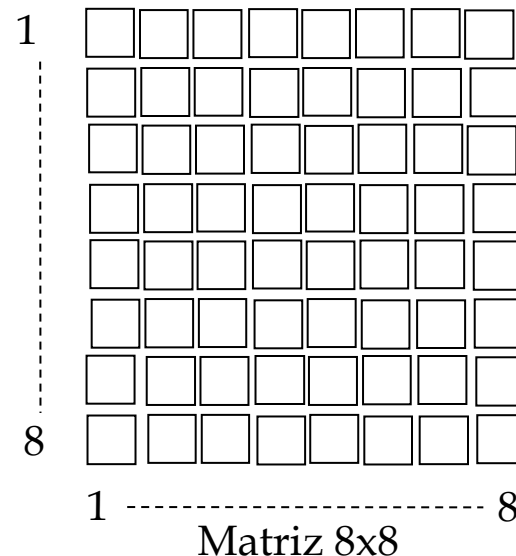


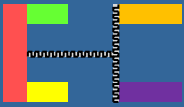


# CONCEPTOS BÁSICOS: ORGANIZACIÓN

## Conceptos básicos

- ☉ Cada elemento de almacenamiento se denomina celda y puede contener 1 ó 0 (*un bit*).
- ☉ Su organización puede ser muy diversa.





# CONCEPTOS BÁSICOS: ORGANIZACIÓN

## Conceptos básicos

- ② **Unidad de transferencia:** es el número de bits que se leen o escriben en memoria **a la vez**.
- ② **Palabra.** El tamaño de la palabra es la unidad básica de organización de la memoria. Suele coincidir con el número de bits de representación de los números y de las instrucciones.
- ② **Unidad direccionable.** Es el tamaño mínimo en que podemos direccionar la memoria, normalmente coincide con la **palabra**. Sin embargo, en ocasiones se permite direccionar a niveles inferiores (p. ej. bytes).

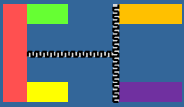




# CONCEPTOS BÁSICOS: ORGANIZACIÓN

## Conceptos básicos

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

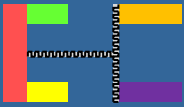


# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

El diseño de la memoria está condicionado por los siguientes aspectos:

- ⦿ Capacidad de almacenamiento.
- ⦿ Tiempo de acceso
- ⦿ Velocidad de transferencia.
- ⦿ Coste por bit.



# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

**CAPACIDAD:** Es la cantidad de información binaria que puede almacenar y depende del número de posiciones y de la longitud de cada una (p. ej. long. de palabra).

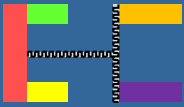
En la matriz del ejemplo de la transparencia 7, se puede ver como:

8 x 8 (cada posición = byte).

16 x 4 (cada posición = nibble).

64 x 1 (cada posición = bit).

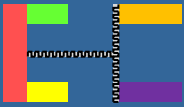
Una memoria de 16Mbytes tiene la misma capacidad que:  
8Mx16, 4Mx32, 128Mbits.



# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

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



# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

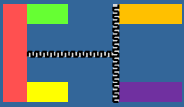
## Conceptos básicos

Para medir el **rendimiento** se utilizan tres parámetros:

🎯 **Tiempo de acceso ( $T_A$ ):** depende del tipo de acceso →

*Aleatorio:* 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.

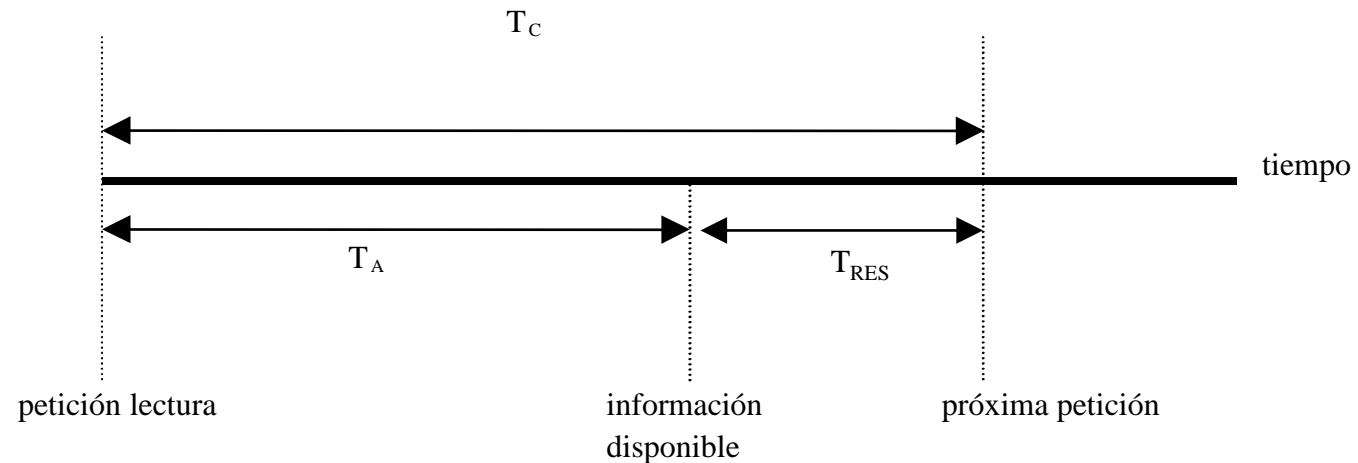
*Resto:* tiempo que se emplea en situar el mecanismo de lectura/escritura en la posición deseada.

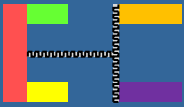


# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

- 🎯 **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: CAPACIDAD Y RENDIMIENTO

## 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:

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

$V_T \rightarrow$  Velocidad de transferencia

$T_C \rightarrow$  Tiempo de ciclo de memoria

En el caso de acceso no aleatorio:

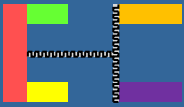
$T_N \rightarrow$  Tiempo medio de lectura/escritura de N bits

$T_A \rightarrow$  Tiempo de acceso

$N \rightarrow$  Número de bits

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

$V_T \rightarrow$  Velocidad de transferencia (bits/segundo)



# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

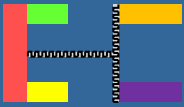
### 🎯 Ejemplo1:

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.

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

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





# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

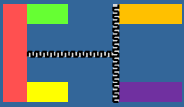
### 🎯 Ejemplo2:

Una memoria 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;  $V_T \text{ (en bytes)} = 4 * 500 = 2000 \text{ bytes/s}$

Solución:  $6 * 2000 = 12000 \text{ bytes}$



# CONCEPTOS BÁSICOS: CAPACIDAD Y RENDIMIENTO

## Conceptos básicos

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

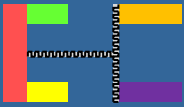
Ejemplo:

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

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

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

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

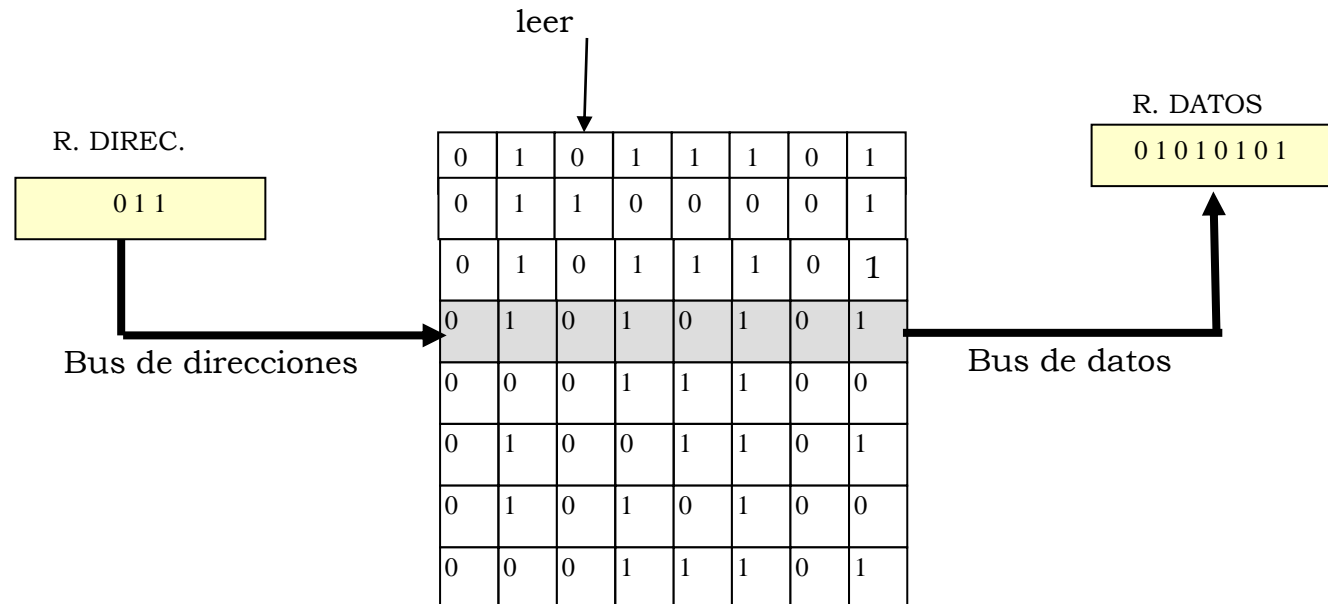


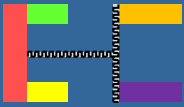
# CONCEPTOS BÁSICOS: OPERACIONES

## Conceptos básicos

### LECTURA

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



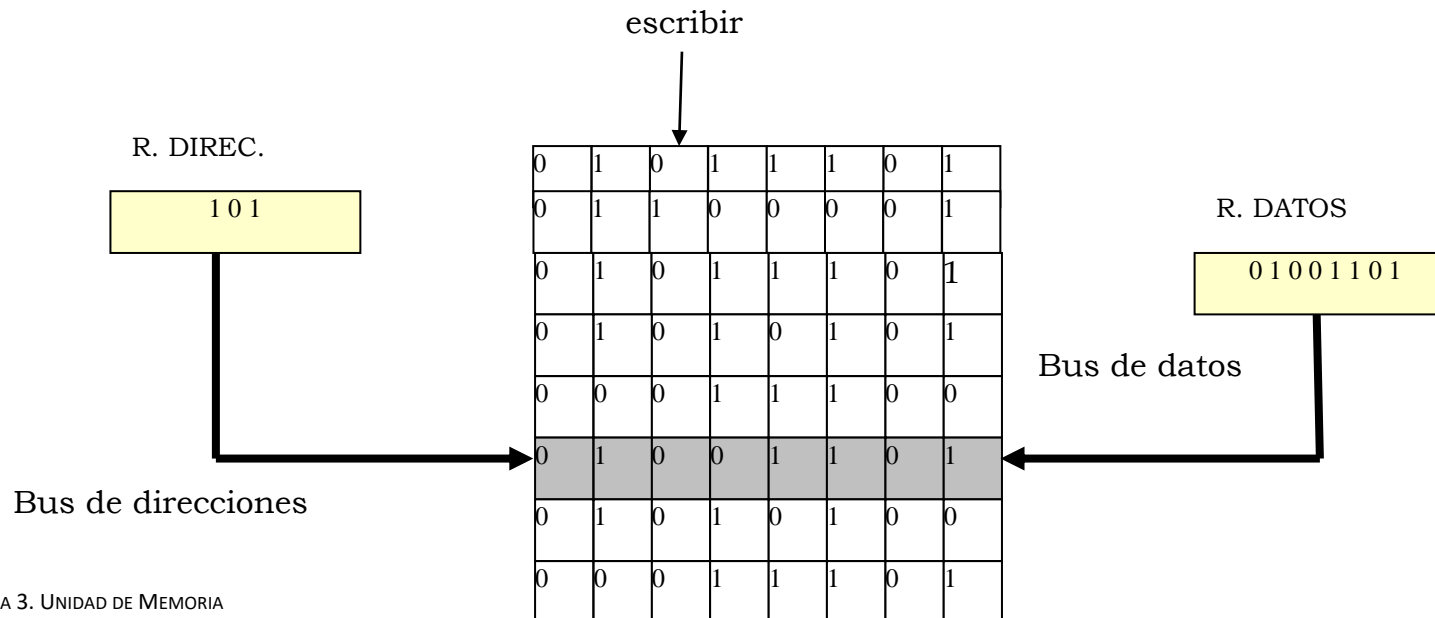


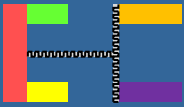
# CONCEPTOS BÁSICOS: OPERACIONES

## Conceptos básicos

### 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



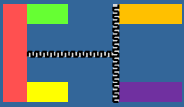


# CONCEPTOS BÁSICOS: JERARQUÍA DE LAS MEMORIAS

## Conceptos básicos

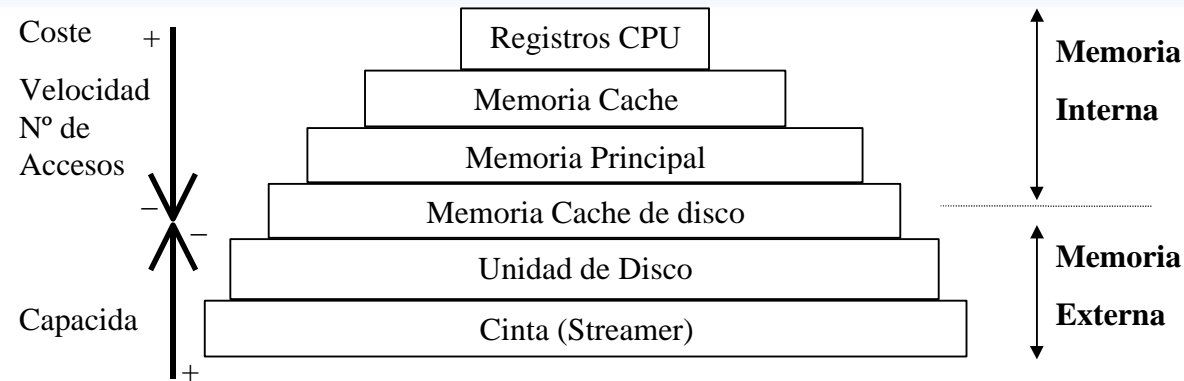
- ⊙ La configuración de memoria ideal es:
  - ⊙ Memoria muy rápida
  - ⊙ Gran capacidad
  - ⊙ Poco coste

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



# CONCEPTOS BÁSICOS: 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.

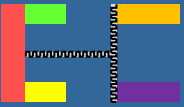


# TIPOS DE MEMORIAS

## Tipos de memorias

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



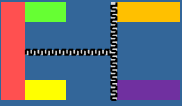


# READ ONLY MEMORY

Memorias  
Semiconductoras

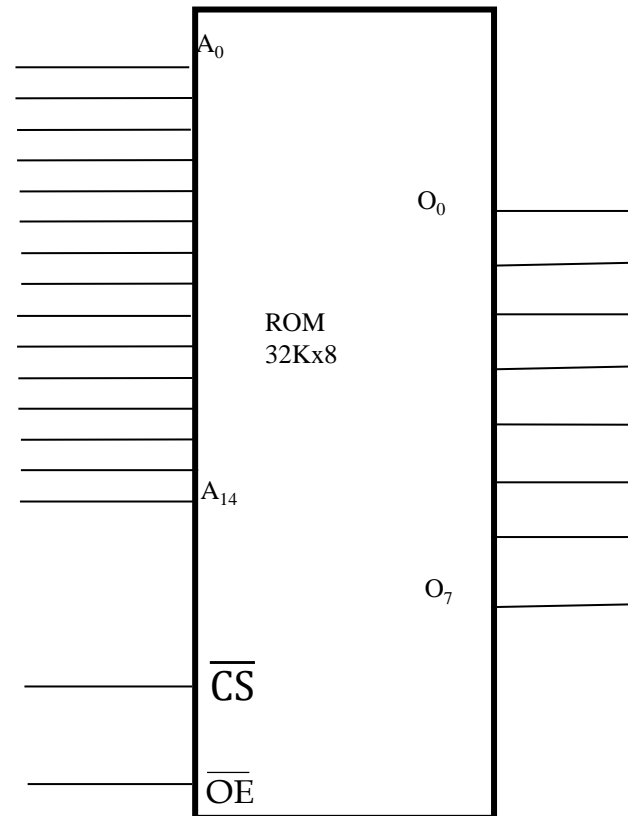
- ⦿ ROM (Read Only Memory). Se graba durante el proceso de fabricación y se suelen utilizar en microprogramación de sistemas, en subrutinas de bibliotecas de uso frecuente.
- ⦿ PROM (Programable Read Only Memory). Son básicamente como una ROM. El proceso de escritura se lleva a cabo eléctricamente y puede realizarlo el suministrador o el cliente con posterioridad a la fabricación del chip original. 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
- ⦿ EEPROM (Electrically Erasable Programmable Read Only Memory). Se borra mediante corriente eléctrica.

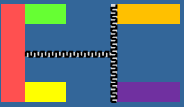




# READ ONLY MEMORY

Memorias  
Semiconductoras





# RANDOM ACCESS MEMORY

Memorias  
Semiconductoras

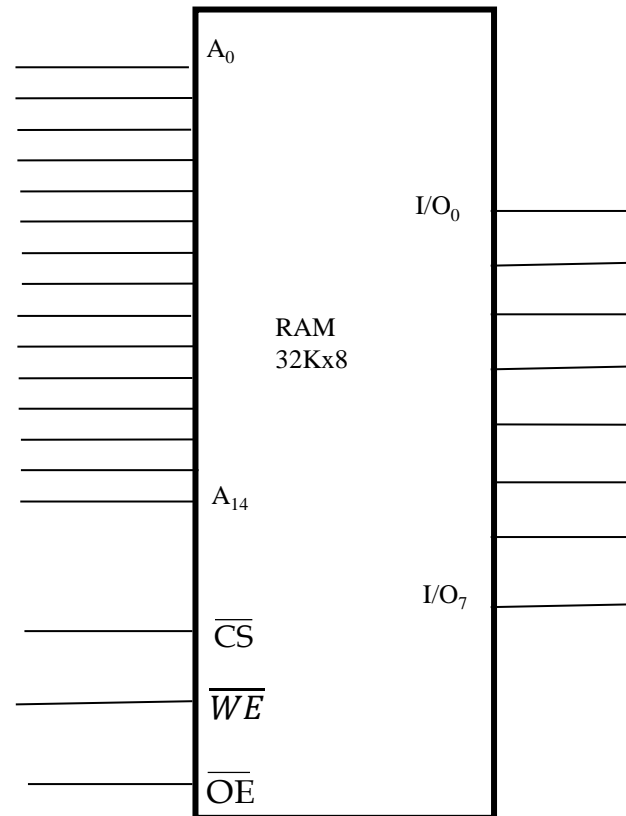
RWM (Read-Write Memory) o RAM (Random Access Memory). Al igual que las ROM, son de acceso aleatorio y volátiles. Los principales tipo de memorias RAM son:

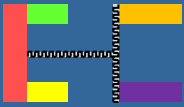
- ⊙ RAM estática (SRAM). El elemento de almacenamiento es un biestable.
- ⊙ RAM dinámica (DRAM). El elemento de almacenamiento es un condensador. Debido a que tiende a descargarse es necesario refrescarlas periódicamente.
  - ⊙ Son más lentas que las SRAM
  - ⊙ Mayor densidad de integración -> Mayor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



# RANDOM ACCESS MEMORY

Memorias  
Semiconductoras



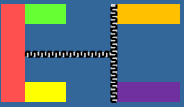


# CONCEPTOS BÁSICOS: TIPOS DE MEMORIAS SEMICONDUCTORAS

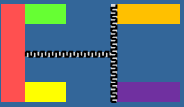
Memorias  
Semiconductoras

## Tabla resumen

Tipo	Clase	Borrado	Escritura	Volatilidad
RAM	Lectura/Escritura	Eléctricamente por bytes	Eléctricamente	Volátil
ROM	Sólo lectura	No	Eléctricamente	No Volátil
PROM		Luz violeta, chip completo		
EEPROM	Sobre todo lectura	Eléctricamente		



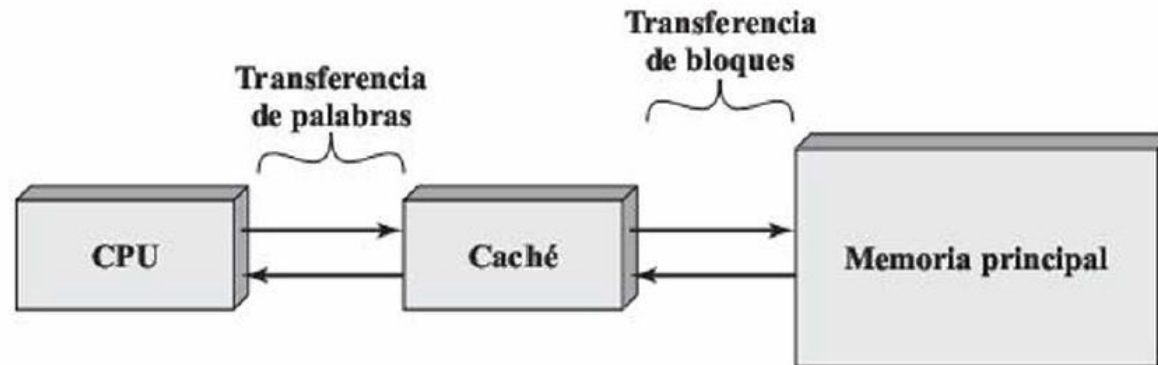
- ⊙ En la organización jerárquica, la **memoria caché** pertenece a la memoria interna del computador (como la RAM), pero **mucho más rápida y cercana al procesador**.
- ⊙ La memoria caché es una memoria muy rápida (y cara) que se suele dividir en niveles: L1, L2, L3.
- ⊙ La estrategia a seguir:
  - ⊙ Organizar los datos y los programas en memoria de forma que las palabras necesarias estén en la memoria más rápida y cercana al procesador.
  - ⊙ Retener copias de las palabras utilizadas recientemente, para no tenerlas que volver a cargar de la DRAM.
  - ⊙ Diseñar adecuadamente la caché para aumentar la tasa de aciertos de palabras en la caché, por parte del procesador

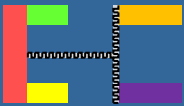


# MEMORIA CACHÉ

## Otras memorias

- ⊙ La CPU lee palabras de memoria, pero la cache se carga con bloques de memoria principal que incluyen la palabra necesaria.
- ⊙ Aprovechamos el fenómeno de la **localidad de las referencias**.

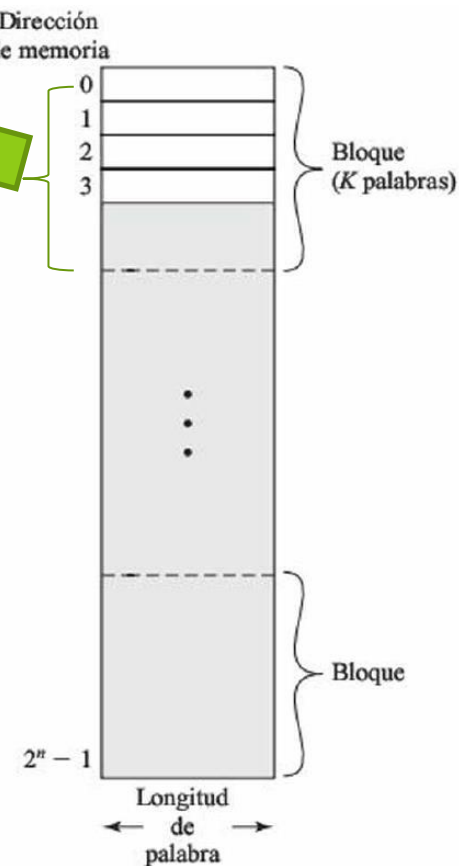
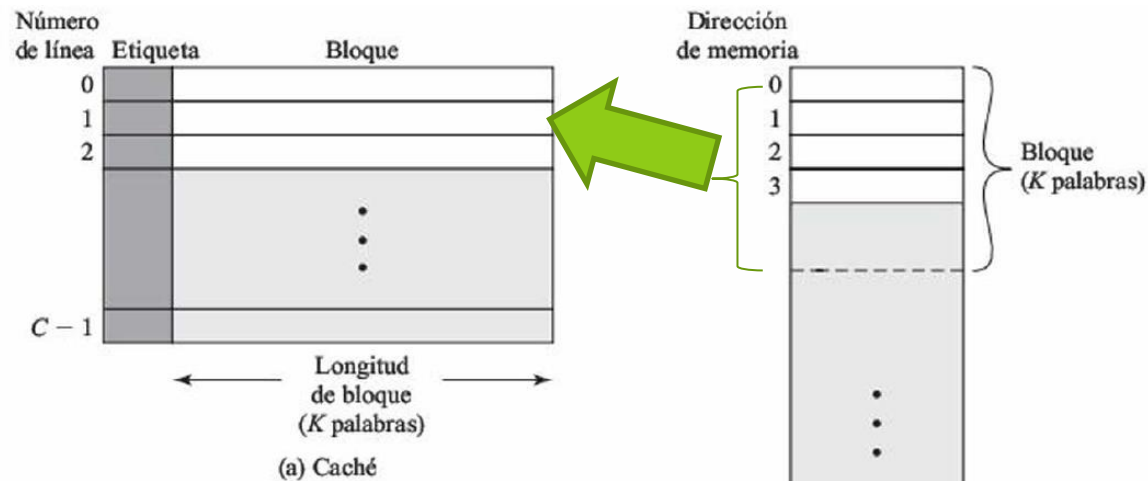




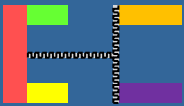
# MEMORIA CACHÉ

## Otras memorias

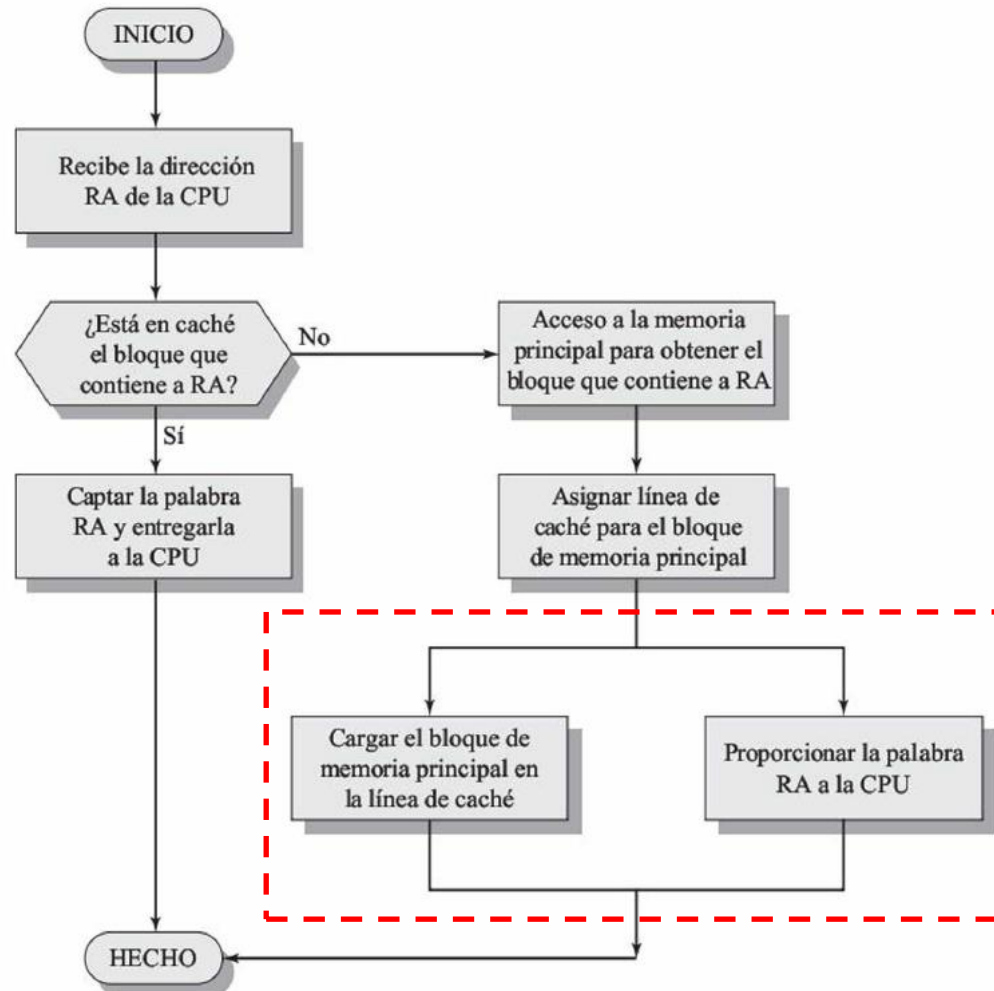
- La CPU lee palabras de memoria, pero la cache se carga con bloques de memoria principal que incluyen la palabra necesaria.
- Aprovechamos el fenómeno de la **localidad de las referencias**.



(b) Memoria principal



## Operación de LECTURA de caché:





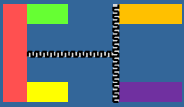


# MEMORIA CACHÉ: CARACTERÍSTICAS

Otras  
memorias

Las características principales de las memorias caché:

- ⊙ Tamaño de caché y tamaño de línea
- ⊙ Función de correspondencia
- ⊙ Algoritmo de sustitución
- ⊙ Número de cachés
- ⊙ Estrategia de escritura
- ⊙ Tiempo de acceso



# MEMORIA CACHÉ: CARACTERÍSTICAS

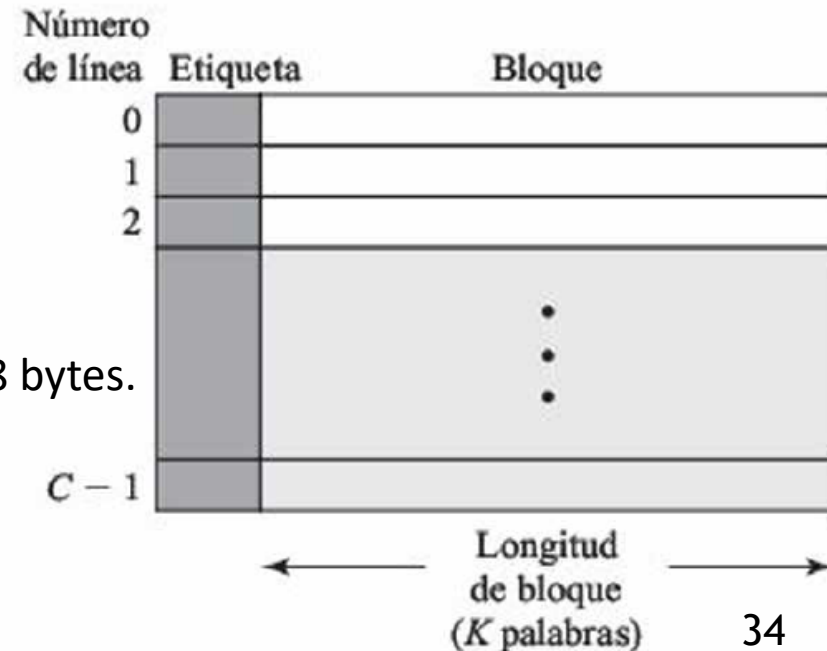
## Otras memorias

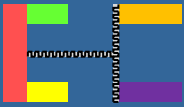
### Tamaño de caché:

- ⦿ No hay un tamaño óptimo, debe caber en el chip o circuito impreso.
- ⦿ Cuanto mayor, más lenta. Hay que lograr equilibrio.
- ⦿ Entre 16KB y 64KB las L1. En algunos computadores hay para datos y para instrucciones.

### Tamaño de línea de caché:

- ⦿ No hay óptimo, depende de la localidad.
- ⦿ Estudios indican entre 8 y 64 bytes.
- ⦿ Computadores altas prestaciones 64 a 128 bytes.



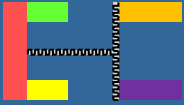


# MEMORIA CACHÉ: CARACTERÍSTICAS

Otras  
memorias

**Función de correspondencia:** Algoritmos que hacen corresponder los *bloques de memoria* principal a las *líneas de caché*.

- ◎ **Correspondencia directa:** Cada bloque de memoria principal se asocia con una línea posible de caché →  $\text{línea\_caché} = \text{bloque} \bmod \text{Núm\_Líneas}$ .
  - ◎ *Fácil de implementar, pero referencias seguidas a dos bloques asignados a la misma línea producen fallos de caché continuos.*
- ◎ **Correspondencia asociativa:** Cada bloque de memoria principal puede cargarse en cualquier línea. Se basa en **etiquetas** que identifican unívocamente cada bloque de memoria principal.
  - ◎ *Resuelven el problema anterior, pero se complica la circuitería de control para la búsqueda de la etiqueta en paralelo para todas las líneas.*

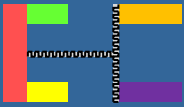


# MEMORIA CACHÉ: CARACTERÍSTICAS

Otras  
memorias

**Algoritmo de Sustitución:** Cómo actuar cuando se llena la caché. Sólo tiene sentido para el caso de correspondencia asociativa.

- ⊙ Sustituir la línea "*menos recientemente usada*" (LRU).
- ⊙ Sustituir la línea según "FIFO".
- ⊙ Sustituir la línea "*usada menos frecuentemente*".
- ⊙ Sustituir la línea aleatoriamente: *Los resultados no son demasiado inferiores como pudiera pensarse.*



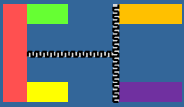
# MEMORIA CACHÉ: CARACTERÍSTICAS

Otras  
memorias

**Número de cachés:** Originalmente un sistema tenía una única caché. En la actualidad se usan múltiples cachés.

Aspectos de diseño asociados al número de cachés:

- ◎ **Cachés multinivel:** Se establecen varios niveles de caché como L1 (*on-chip*) y L2 (*off-chip* deben usar bus externo a RAM). En la actualidad, hay hasta nivel L3 y, con la mejora en la disponibilidad de espacio en el propio chip, todas ellas pueden implementarse *on-chip*.
- ◎ **Cachés unificadas frente a separadas:** Originalmente la caché se diseñaron unificadas. En la actualidad, especialmente en máquinas superescalares, se tiende a diseños que separan la caché de **datos** de la de **instrucciones**.
  - ◎ *Tienen interés en procesamiento segmentado, de forma que se saca partido al paralelismo y a la captación de instrucciones anticipadas.*
  - ◎ *Evita que las necesidades de caché de instrucciones y datos simultáneamente, no degrade las prestaciones.*



# MEMORIA CACHÉ: CARACTERÍSTICAS

## Estrategia de escritura

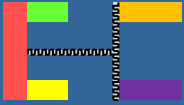
### ⊙ Escritura inmediata o directa (write through)

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

### ⊙ Post-escritura (write back)

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

## Problema de coherencia de los datos



# MEMORIA CACHÉ: CARACTERÍSTICAS

## Tiempo de acceso a memoria con Memoria Caché

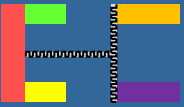
$$T_{\text{acceso}} = N_a * T_c + N_f * T_p$$

$N_a$  es el número de referencias con acierto

$N_f$  es el número de referencias con fallo

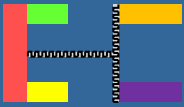
$T_c$  es el tiempo de acceso a una palabra de Memoria Caché

$T_p$  es el tiempo de acceso a un bloque de Memoria Principal



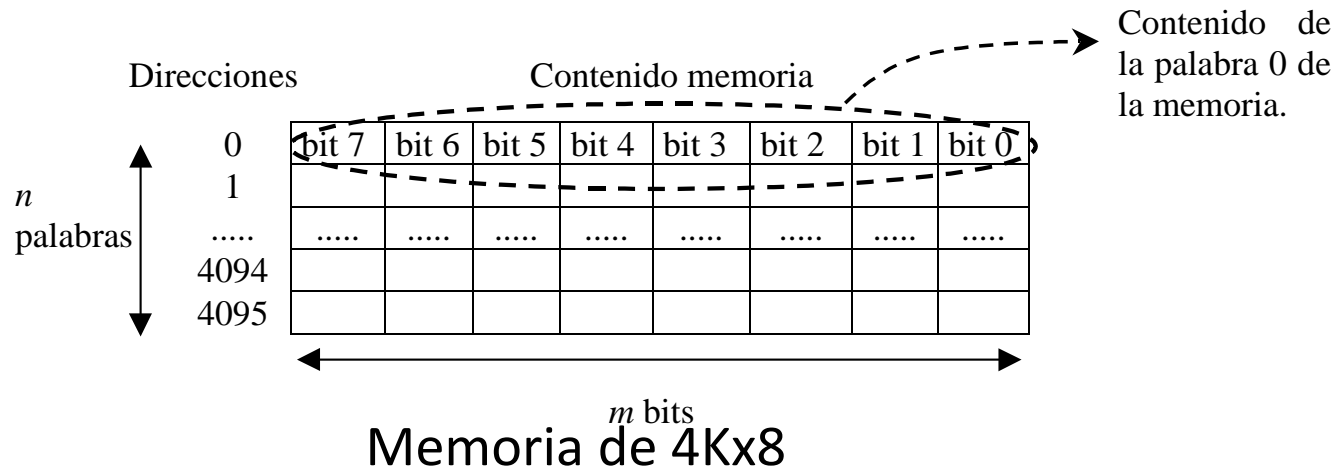
- ⊙ Diseño de la unidad de memoria principal
  - ⊙ Mapa de memoria. Determina la memoria direccionable y el tipo de memoria.
  - ⊙ Diseño de la organización de los chips de memoria y su interconexión
  - ⊙ Diseño de la interfaz hardware procesador-memoria

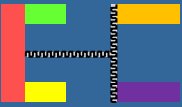




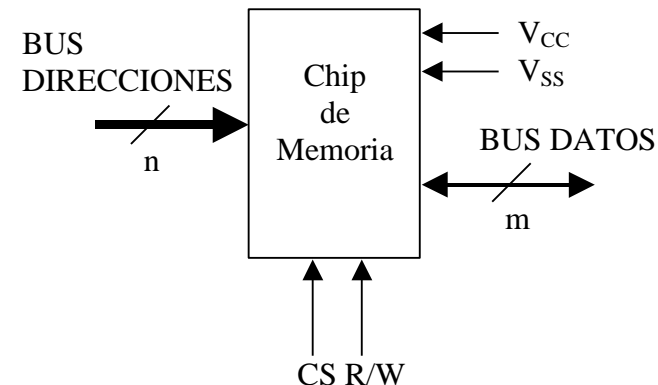
## Chip de memoria

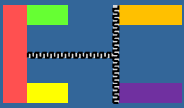
Se organiza internamente como una matriz de celdas de memoria de  $n \times m$ , donde  $n$  es el número de palabras que puede almacenar el chip de memoria y  $m$  es el número de bits por palabra.





- 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.

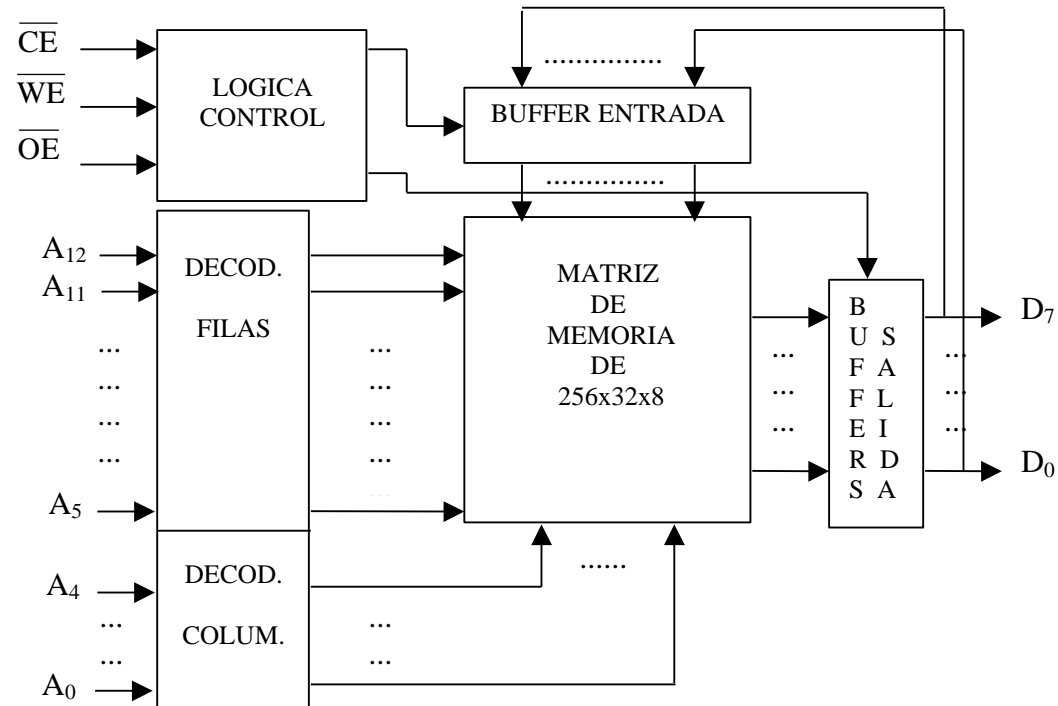


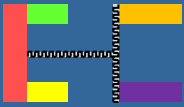


# INTRODUCCIÓN AL DISEÑO

## Diseño de mapas de memoria

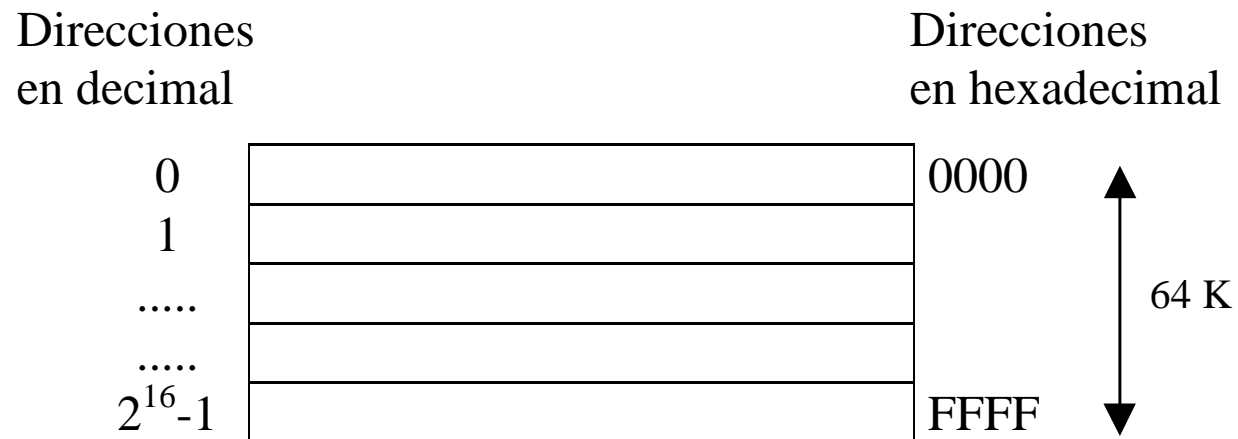
- Para el correcto funcionamiento de la memoria es necesario incorporar una circuitería adicional como son decodificadores, multiplexores, buffers, etc.



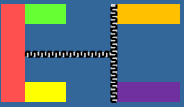


## 🎯 Mapa de memoria

Espacio que puede direccionar un computador.



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

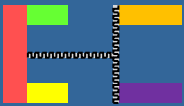
Diseño de  
mapas de  
memoria

- ◎ 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.

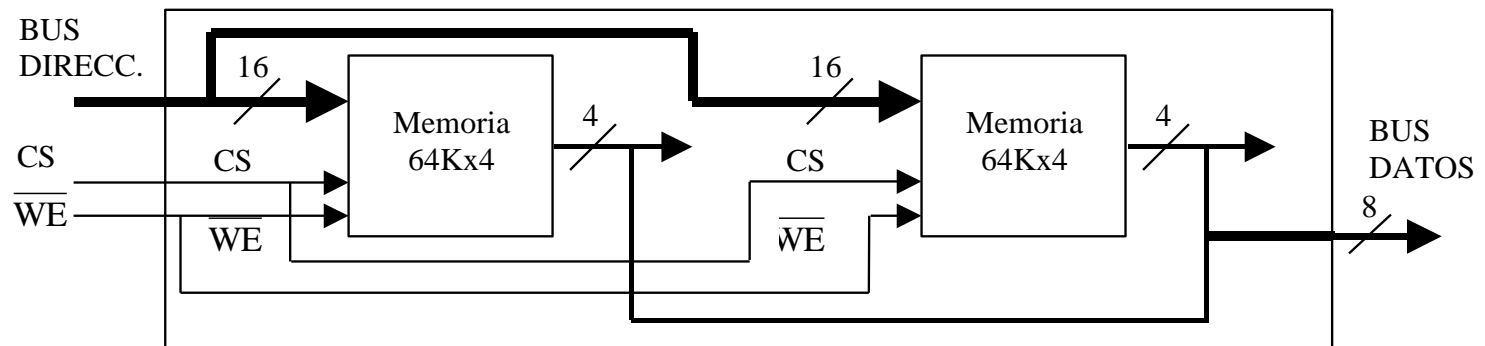


# DISEÑO - EJEMPLO 1

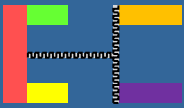
## Diseño de mapas de memoria

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.





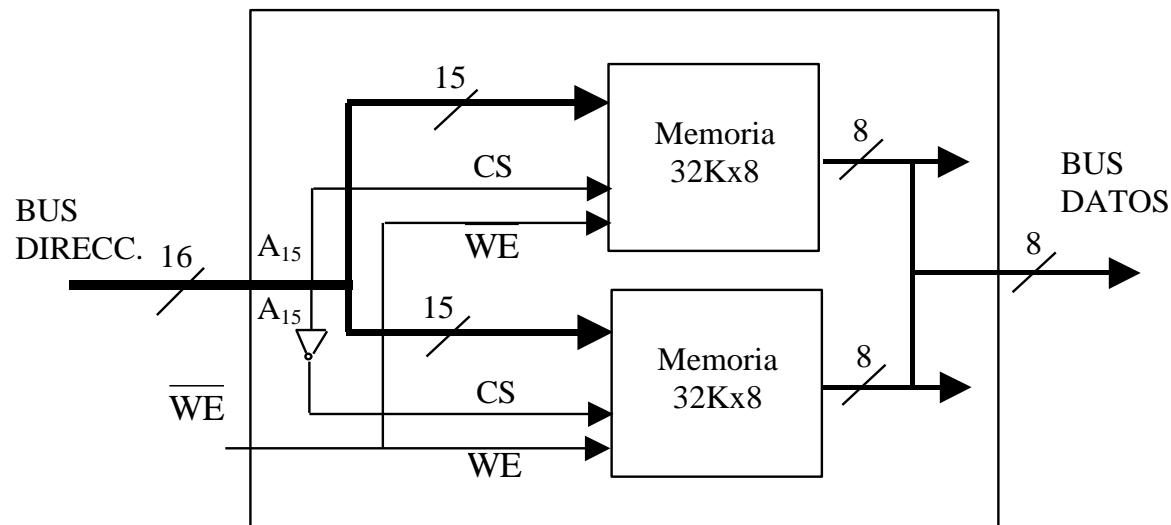


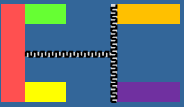
# DISEÑO - EJEMPLO 1

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.



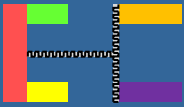


## DISEÑO - EJEMPLO 2

### 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 ( $1M = 2^{20}$ ).

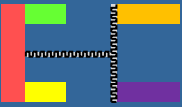
- (2) Averiguar el número de bits que se necesitan para direccionar el chip de memoria que vamos a emplear.

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



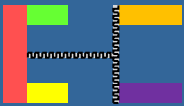
(3) Calcular el número de chips que necesitamos.

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.

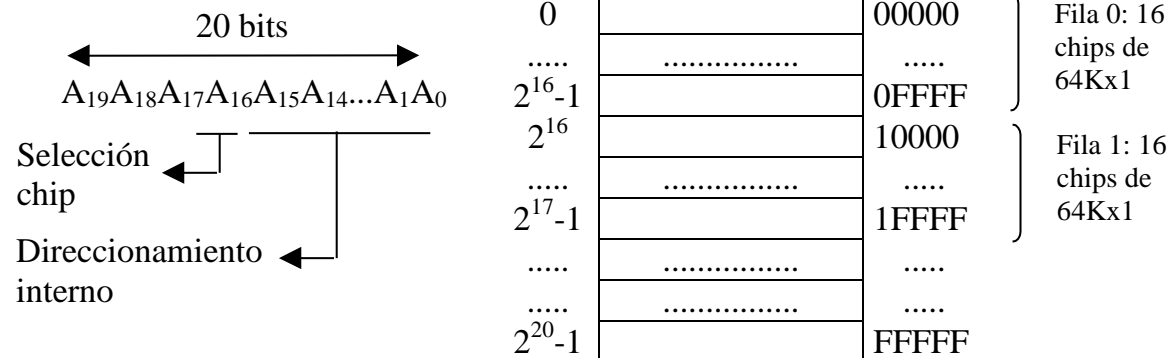
Como tenemos 2 filas de 16 chips cada una, necesitaremos 1 bit para diferenciar una fila de otra. Por tanto utilizaremos el bit  $A_{16}$  para seleccionar los chips de memoria. El resto de direcciones se utilizarán para futuras ampliaciones de memoria del computador.



## DISEÑO - EJEMPLO 2

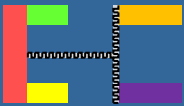
### Diseño de mapas de memoria

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



Mapa de Memoria

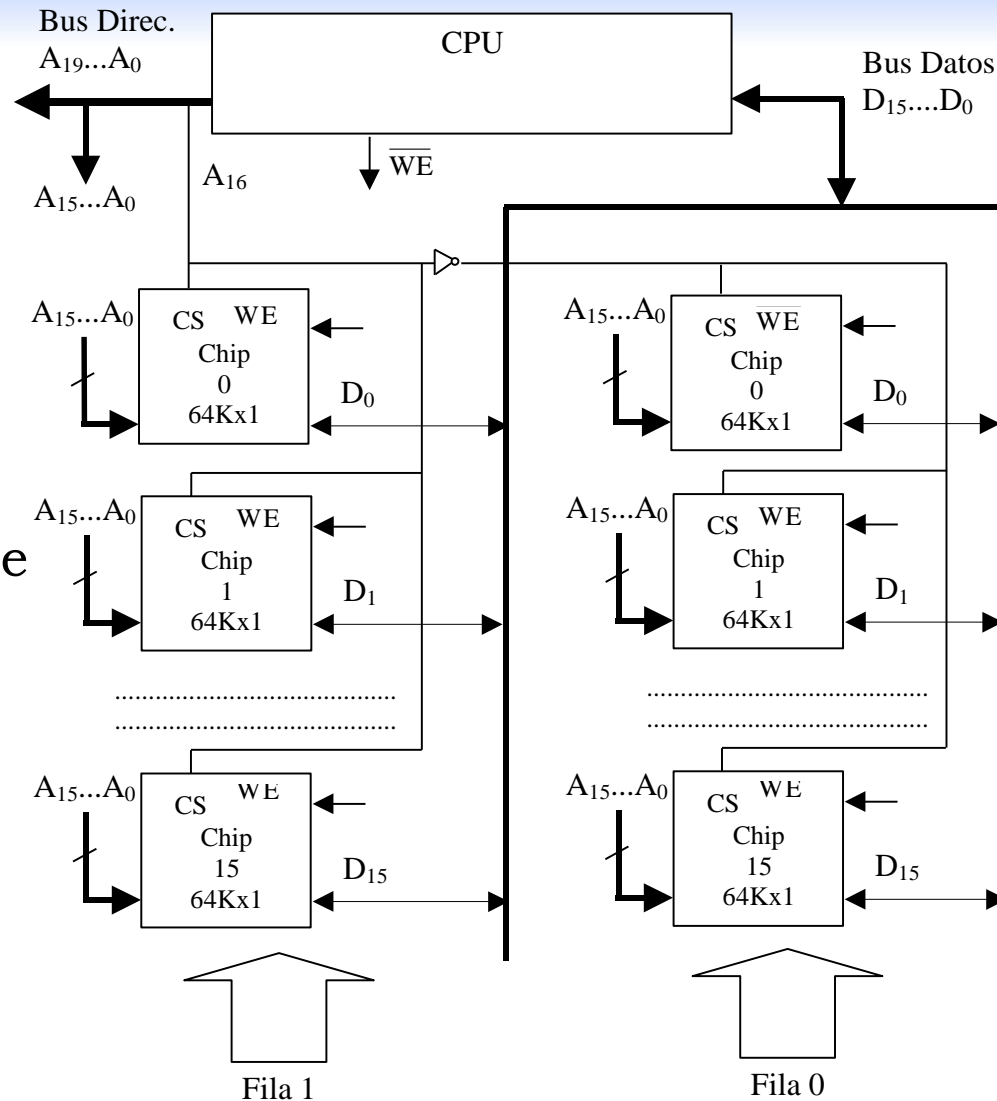
$A_{19}$	$A_{18}$	$A_{17}$	$A_{16}$	$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Fila 0: 16 chips de 64Kx1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	0	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	



## DISEÑO - EJEMPLO 2

Diseño de  
mapas de  
memoria

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



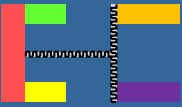


## DISEÑO - EJEMPLO 3

### 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.





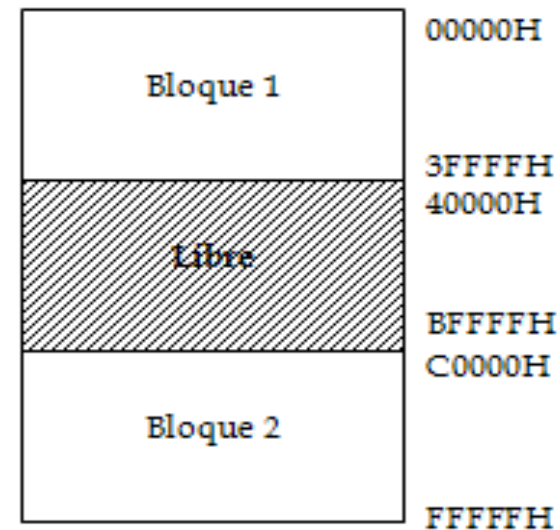
## DISEÑO - EJEMPLO 3

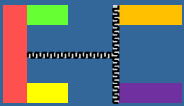
### Diseño de mapas de memoria

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





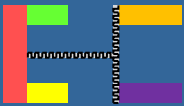
## DISEÑO - EJEMPLO 3

### 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	

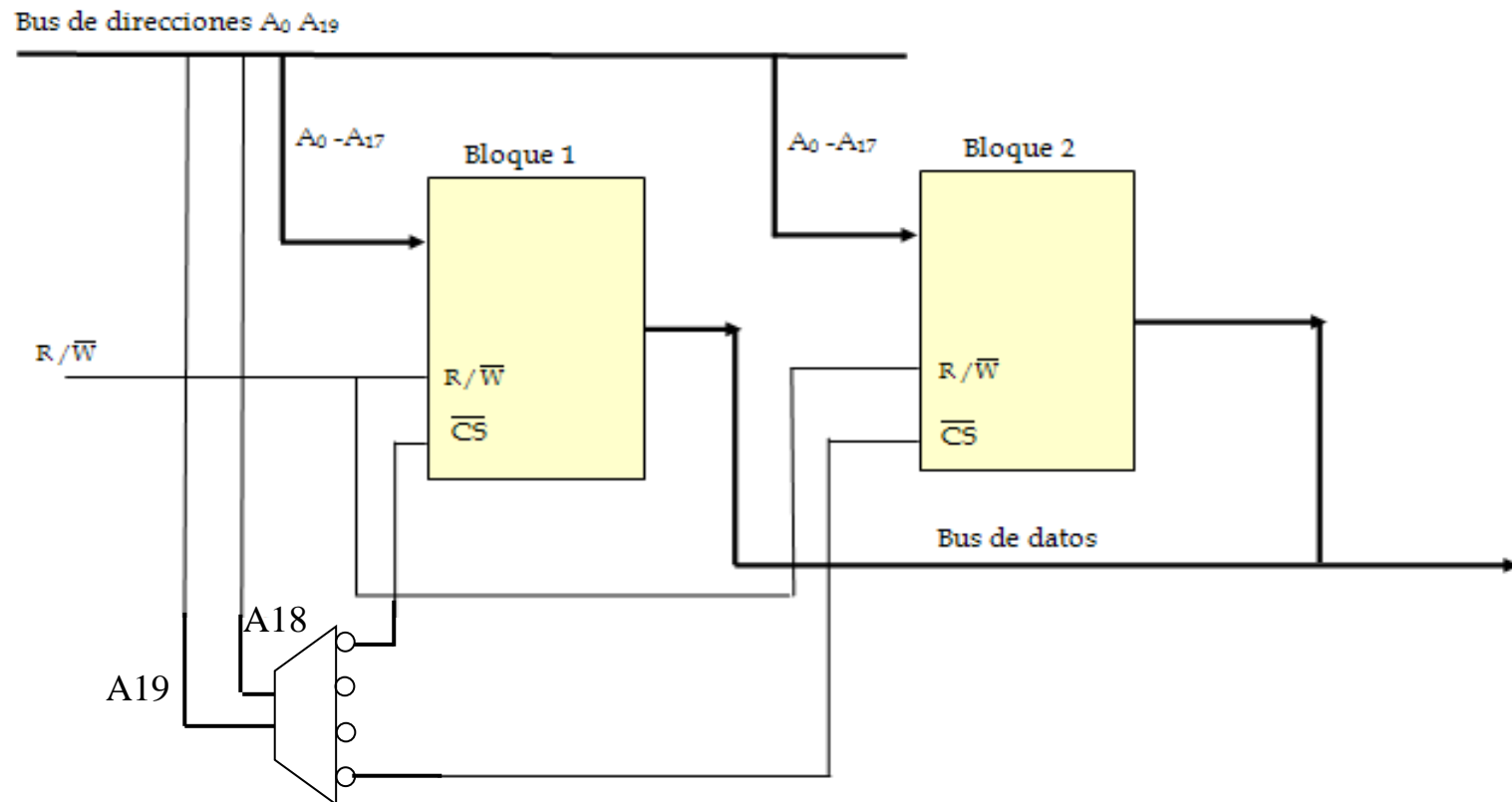
- Por lo tanto el esquema eléctrico queda como sigue:

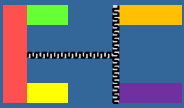


# DISEÑO - EJEMPLO 3

Diseño de  
mapas de  
memoria

- Esquema eléctrico (realizado con decodificador)





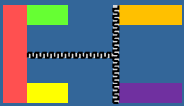
## DISEÑO - EJEMPLO 3

### 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	

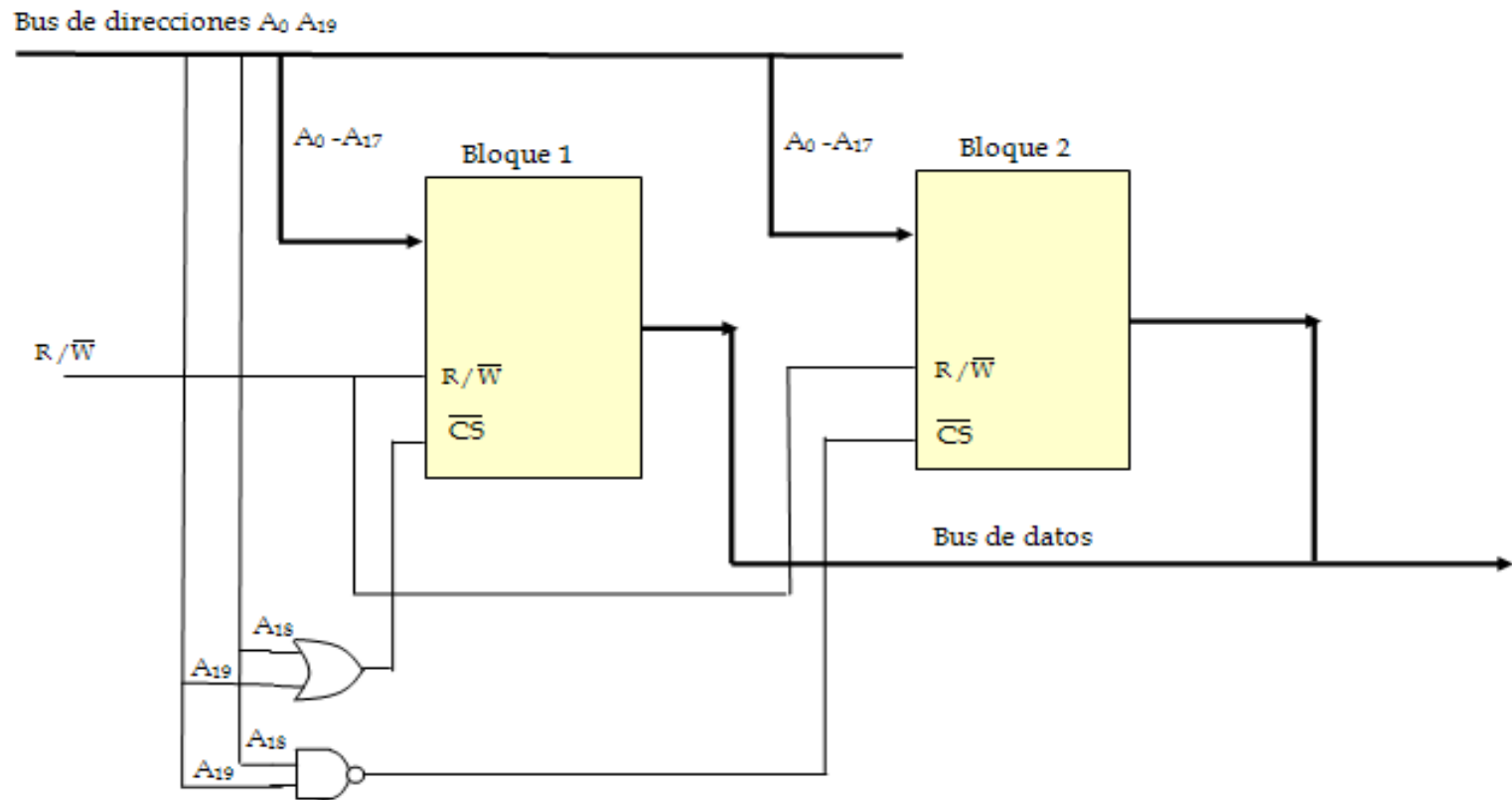
- ⊙ El bloque 1 se debe activar cuando los bits A18 y A19 sean cero, quiere esto decir que su línea  $\overline{CS}_1$  se debe poner a cero en dicho caso, se consigue mediante la función  $\overline{CS}_1 = A_{18} + A_{19}$   
sin embargo el bloque 2, se controla mediante la función:  $\overline{CS}_2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

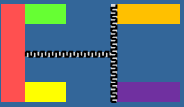


## DISEÑO - EJEMPLO 3

Diseño de  
mapas de  
memoria

- Esquema eléctrico (realizado con puertas lógicas)

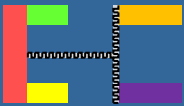




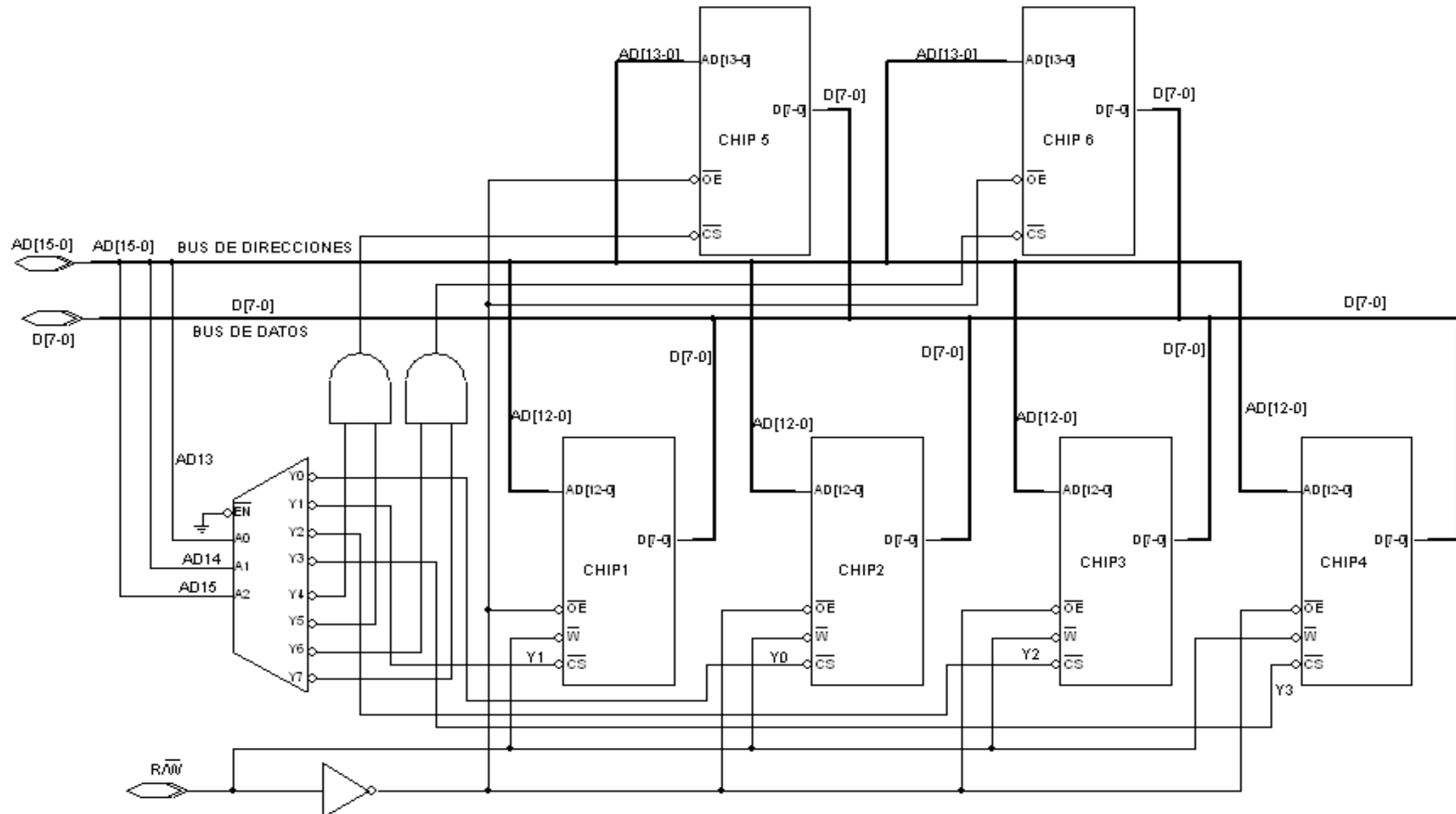
Ha caído en nuestras manos el esquema de la memoria que utiliza el nuevo sistema de codificación de la información patentado por MaNoTas S.A., pero necesitamos averiguar las siguientes cosas:

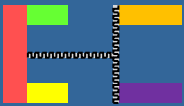
- a) Capacidad de cada memoria medida en Kbits y en Kbytes.
- b) Direcciones de inicio y final de los chip 1 al 6.
- c) Tipo de memoria utilizada en cada chip.
- d) Tabla de funcionamiento de cada tipo de memoria.
- e) Modificar el diseño para que se puedan unificar las memorias en dos: una que sustituya 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





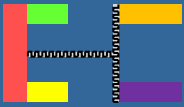
## DISEÑO - EJEMPLO 4

### 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			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			FFFFH		



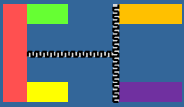


- Las memorias 1, 2, 3 y 4 son de lectura escritura, ya que poseen las líneas **OE** y **W** y, por lo tanto, son de las llamadas RAM. Su tabla de funcionamiento es:

$\overline{CS}$	$\overline{WE}$	$\overline{OE}$	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 **OE**, no tiene **W** y, por tanto, son de las llamadas ROM. Su tabla de funcionamiento es:

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

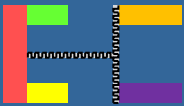


## DISEÑO - EJEMPLO 4

### 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

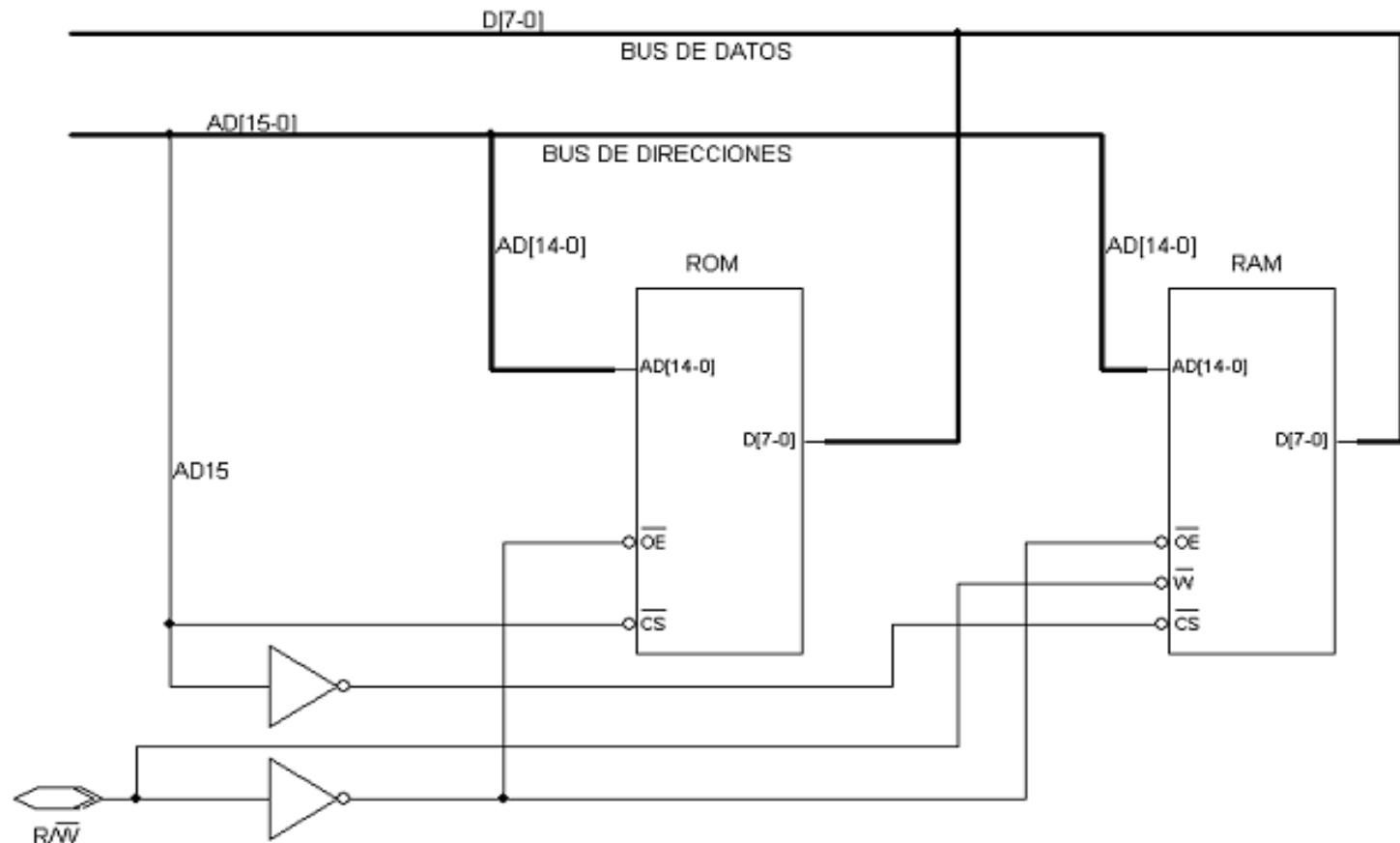
A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Deco	Chip	Dirección	Cap.	Tipo
0	0	0	0	0	0	0	0	0	0	0	0	0	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

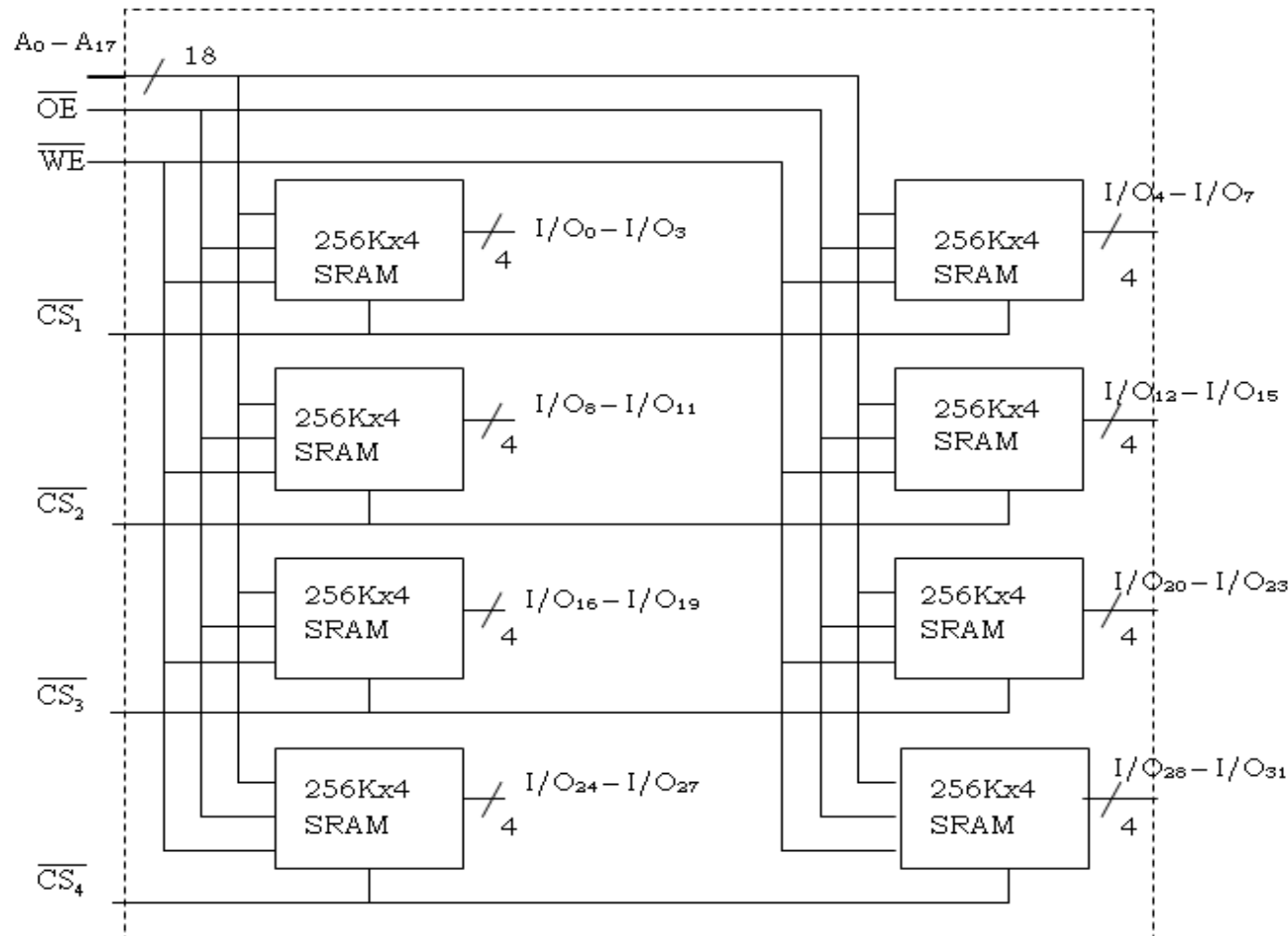
Diseño de  
mapas de  
memoria

- El esquema eléctrico (con puertas lógicas) sería el siguiente



# EJERCICIO 1

**CYM1841A (256KX32)**

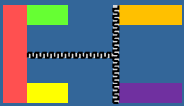


**TABLA DE VERDAD**

$\overline{CS}$	$\overline{WE}$	$\overline{OE}$	E/S	Modo
H	X	X	Alta impedancia	Deseleccionada / <u>Power-Down</u>
L	H	L	Salida	Lectura
L	L	X	Entrada	Escritura
L	H	H	Alta impedancia	<u>Deseleccionada</u>

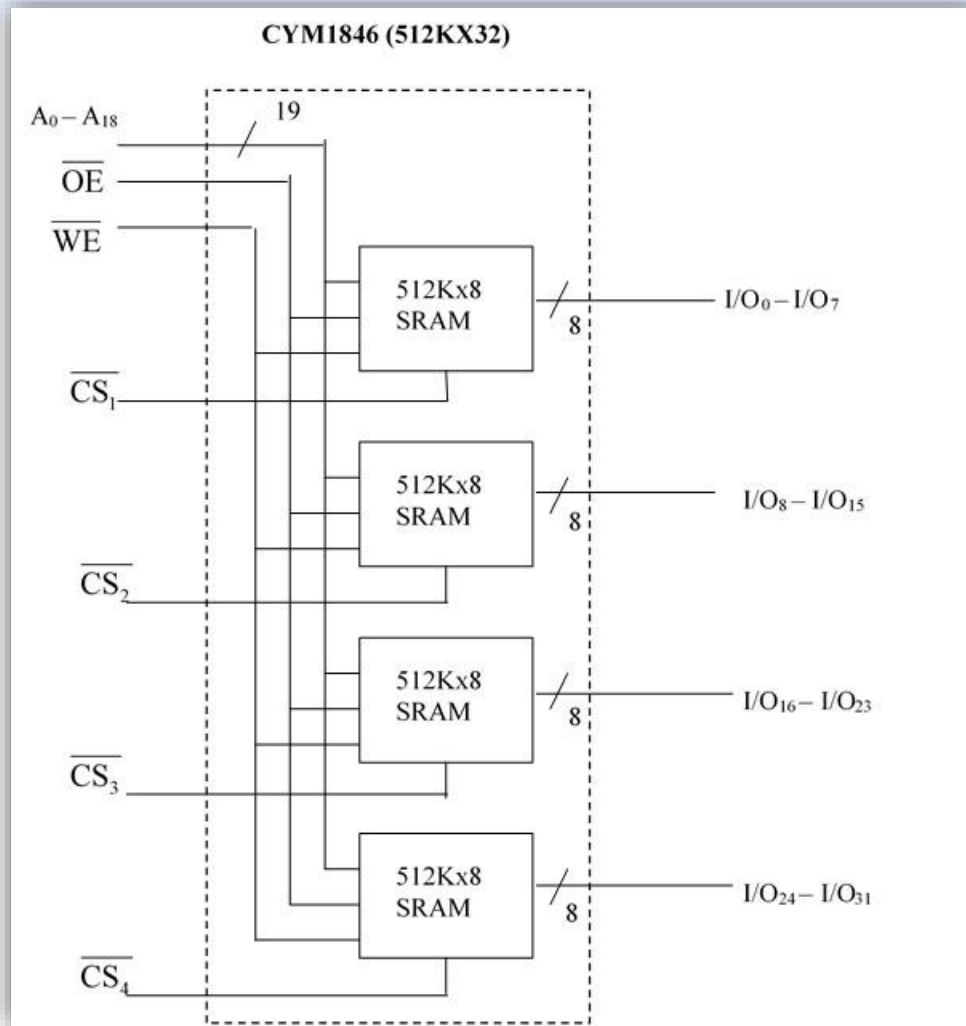
CPU con bus de direcciones de 20 bits y  $R/\overline{W}$

- Mapa de memoria y circuito con decodificadores para 512Kx32 y primera dirección 00000H
- Mapa de memoria y circuito con decodificadores para 1Mx8 y primera dirección 00000H
- Mapa de memoria y circuito con decodificadores para 512Kx16 y comenzando en la dirección 80000H



## EJERCICIO 2

### Ejercicios



CPU con bus de direcciones de 20 bits y  $R/\overline{W}$

- a) Mapa de memoria y circuito con decodificadores para 512Kx32
- b) Mapa de memoria y circuito con decodificadores para 1Mx16
- c) Mapa de memoria y circuito con decodificadores para 512Kx8