

Tema 16: Memorias de semiconductor (1)

Objetivos:

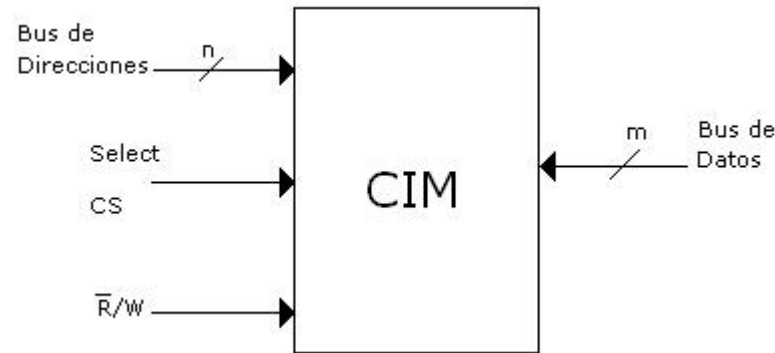
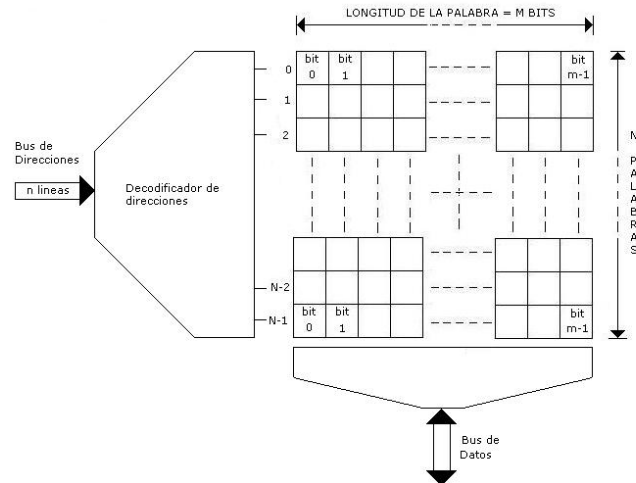
- Introducción.
- Características generales de los CIM.
- Estructura de la celda básica de Memoria.
- Organización interna.
 - Organización 2D.
 - Organización 3D.
- Diseño de bloques de Memoria.
- Conexión entre memoria y bus del sistema.
- Mapa de memoria principal.

INTRODUCCIÓN

- En los primeros computadores, la memoria principal utilizaba una matriz pequeña de bobinas ferromagnéticas conocidas como núcleos de ferrita.
- La aparición de la microelectrónica ha desplazado este tipo de memorias de núcleo
- A finales de lo 60 se comenzó la fabricación de las primeras memorias de semiconductor, utilizando como componentes básicos los Circuitos Integrados de Memoria (CIM).
- Con el avance de las técnicas de integración ha sido posible fabricar circuitos integrados de memoria de elevada capacidad con una velocidad compatible con los procesadores y a unos costes razonables.

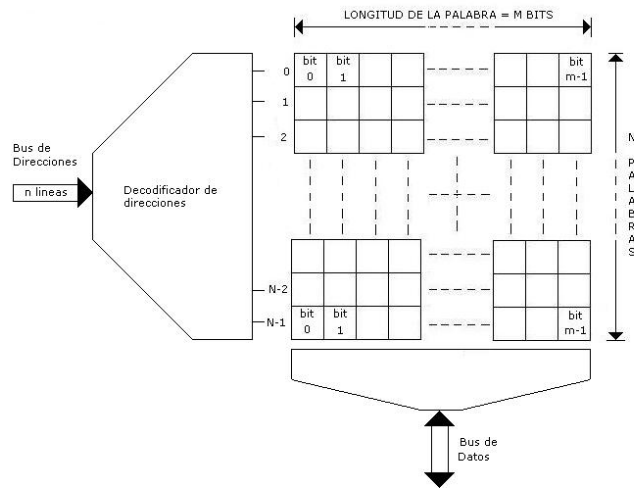
Características de los CIM

- Un CIM se organiza en forma de matriz de $N \times m$ celdas elementales en las que podemos almacenar N palabras de m bits cada una de ellas.

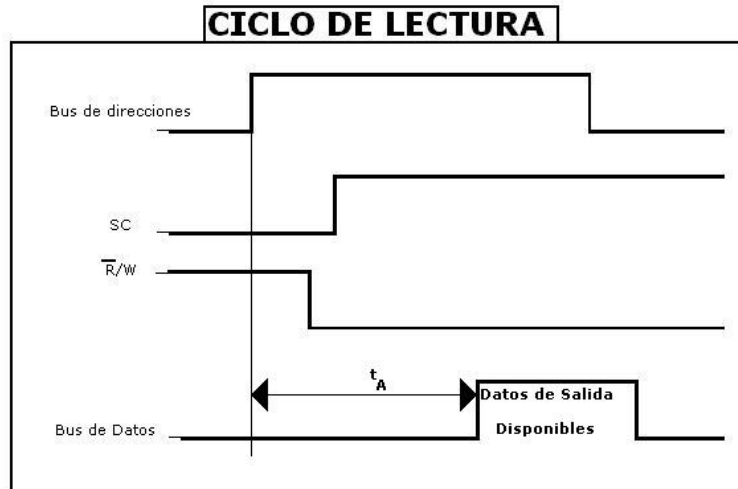


- El número de líneas del bus de direcciones (n) deberá cumplir $2^n = N$ siendo N el número de palabras del CIM.
- El número de líneas del bus de datos se corresponde con la longitud de la palabra o divisor de ésta.
- La línea R/W sirve para seleccionar entre una operación de lectura u otra de escritura.
- La entrada CS (Chip Select) sirve para habilitar o no el CIM.

Características de los CIM

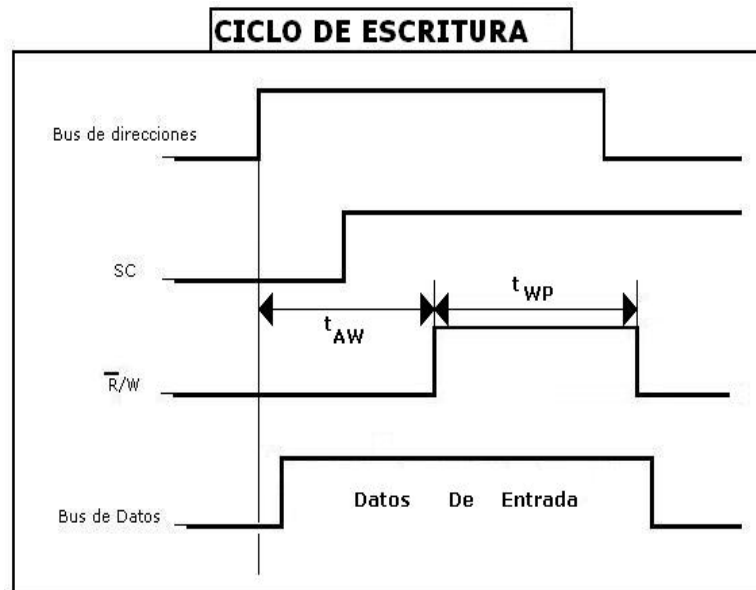
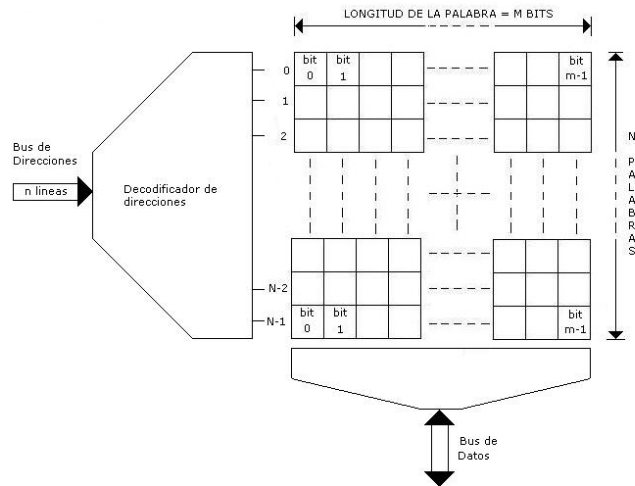


- Un ciclo de lectura comienza colocando la dirección en el bus de direcciones, tras ello se debe activar la línea SC.
- La línea R/W debe estar a cero para indicar que se desea hacer una lectura.
- t_A instantes después de colocar la dirección en el bus de direcciones, tenemos los datos disponibles en el bus de datos



- En un ciclo de escritura, los datos a escribir se envían al bus de datos aproximadamente al mismo tiempo que se coloca la dirección, y transcurridos t_{AW} segundos la línea R/W debe pasar a 1 debiendo permanecer al menos t_{WP} (Anchura de pulso de escritura)

Características de los CIM



- En un ciclo de escritura, los datos a escribir se envían al bus de datos aproximadamente al mismo tiempo que se coloca la dirección, y transcurridos t_{AW} segundos la línea R/W debe pasar a 1 debiendo permanecer al menos t_{WP} (Anchura de pulso de escritura).

- Para garantizar que los datos han sido escritos se deben respetar estos tiempos sin cambiar los datos del bus de datos.

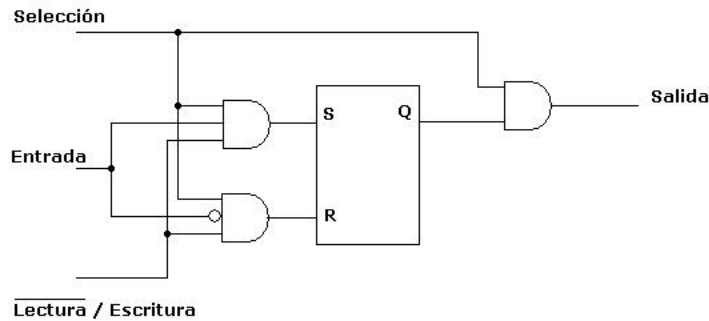
- El tiempo de ciclo de escritura:

$$t_W = t_{AW} + t_{WP}$$

Estructura de la celda básica de Memoria

- La celda básica de un CIM permite almacenar un bit (0 ó 1) pudiéndose fabricar con diferentes tecnologías.
- La celda básica debe permitir ser escrita como mínimo una vez, así como ser leída tantas veces como se desee.
- Las celdas básicas de una memoria ROM son simples conmutadores fabricados con semiconductores y que están permanentemente cerrados o abiertos, generando un 1 o un cero, estos conmutadores se abren o cierran una sola vez y permanecen en ese estado de forma indefinida.
- La celda básica de una memoria RAM es más compleja ya que se debe poder leer/escribir en cualquier momento, como ejemplo:

Estructura de la celda básica de Memoria



OPERACIÓN DE ESCRITURA :

- Para realizar una escritura en la celda básica, es necesario colocar el 0 ó 1 a escribir en la línea “*entrada*” junto con un 1 en la entrada Lectura/escritura además de habilitar la celda con otro 1 en la entrada de selección.

- De esta forma, si entrada = 1, estamos poniendo (S=1 R=0), por lo que Q=1.
- Si entrada = 0, estamos poniendo (S=0 R=1), por lo que Q=0.

• OPERACIÓN DE LECTURA :

- Para realizar una lectura de la celda básica, colocaremos con un 0 en la entrada Lectura/escritura además de habilitar la celda con otro 1 en la entrada de selección.
- Así, la salida Q del biestable pasa a la salida de la celda de memoria, y además, las entradas S,R están aisladas de la entrada de la celda gracias al 0 que coloca la entrada lectura/escritura en las puertas NAND de la entrada

Estructura de la celda básica de Memoria

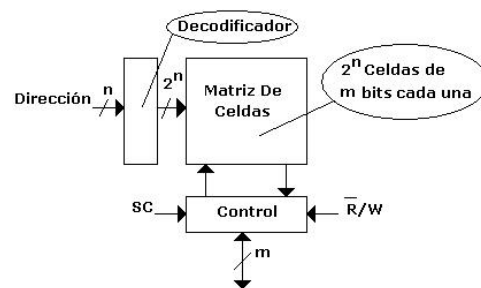
- La celda básica mostrada es del tipo estática, ya que los datos almacenados se mantienen mientras no se interrumpa la alimentación.
- También existen memoria RAM dinámicas, en las que los datos almacenados se desvanecen de forma gradual, debido a que están basadas en capacidades que van perdiendo su carga debido a fugas.
- Para evitarlo necesitan ser refrescadas, para el refresco no es necesario volver a escribir el dato.
- La principal ventaja de las DRAM son su sencillez y alta capacidad

Organización Interna

- Cada CIM está formado por una matriz de celdas básicas de memoria.
- La organización interna de un CIM se puede realizar de diferentes formas.
- Dos posibles formas de organizar internamente las celdas básicas pueden ser:
 - Organización 2D.
 - Organización 3D.

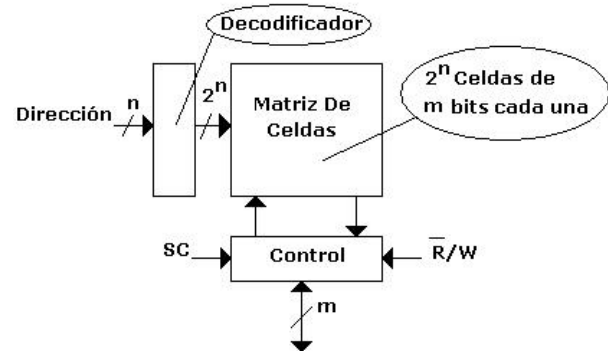
• Organización 2D

- Es el método más simple pero de coste más elevado.
- La organización 2D también es conocida como organización lineal.
- Este tipo de organización consiste en una matriz de celdas básicas.



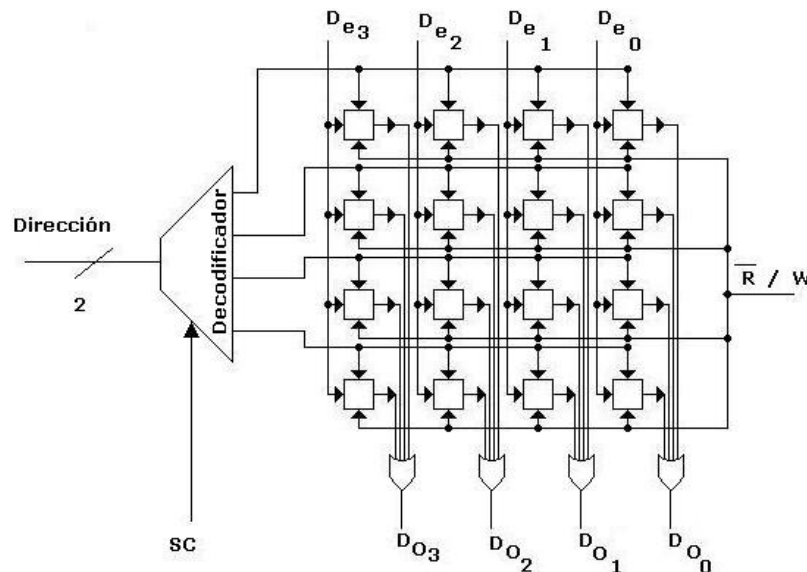
- En la figura disponemos de una memoria de 2^n palabras de m bits cada una de ellas, es decir, una matriz de 2^n filas por m columnas de celdas básicas.

Organización Interna



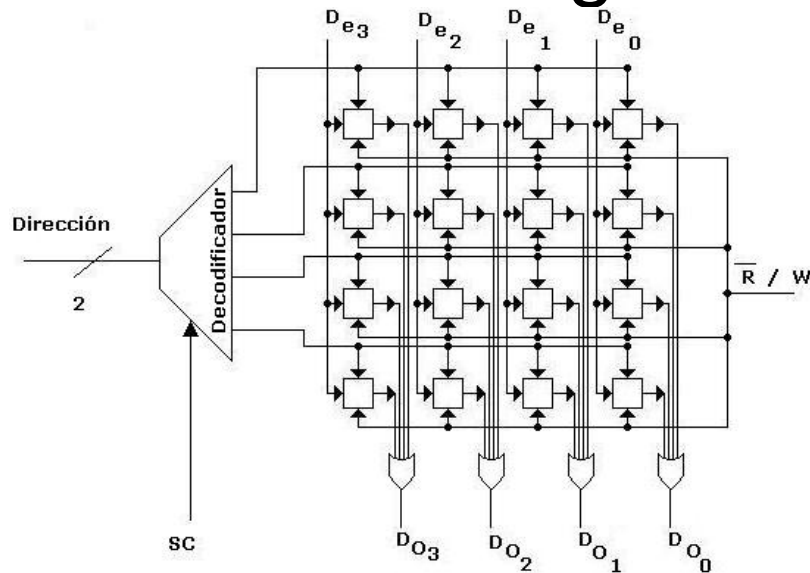
- Para seleccionar la fila de la matriz a la que deseamos acceder, llegará la indicación en forma de dirección, esta dirección será decodificada (Decodificador) y seleccionada la línea correspondiente de la matriz de celdas.

- La salida la formarán m puertas de tipo OR de 2^n entradas cada una de ellas



- En la figura vemos una memoria de 4 posiciones (direccionable con 2 bits).
- Cada dirección consta de 4 bits de información.
- Existe una entrada SC para habilitar la memoria.

Organización Interna



- Al decodificador le llegarán las 2 líneas $(A_1 A_0) = (00 \ 01 \ 10 \ 11)$. Para cada una de estas combinaciones activará una y sólo una de sus 4 salidas.
- Dicha activación llega a las cuatro celdas básicas de una de las filas de la matriz activándola.
- Dependiendo de si es una operación de lectura o escritura, el dato a escribir entrará desde $D_{e3} \ D_{e2} \ D_{e1} \ D_{e0}$, o el dato a leer saldrá de $D_{o3} \ D_{o2} \ D_{o1} \ D_{o0}$.

• La organización 2D es una organización rápida ya que el único retardo que presenta es el de la decodificación así como el de las propias celdas básicas, por ello, en principio es un tipo de organización válido para cualquier número de filas y columnas.

Organización Interna

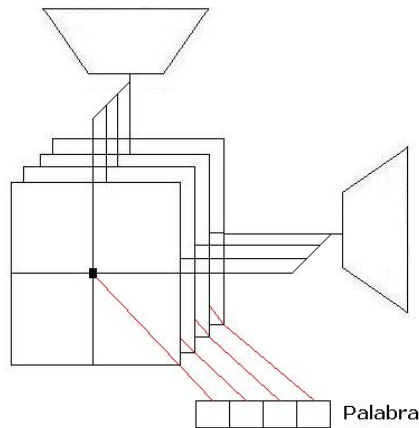
- Pero el aumento del tamaño de la matriz conlleva:
 - Mayor complejidad del Decodificador (Por cada n entradas necesito 2^n puertas AND).
 - Mayor cableado en el número de líneas de dirección y/o de datos lo que hace necesarias puertas OR de gran número de entradas
- Además, normalmente el número de direcciones es mucho mayor que el tamaño de la palabra, ello da lugar a matrices estrechas y largas difíciles de integrar,
- Debido a estas desventajas, veremos la organización 3D.

Organización 3D

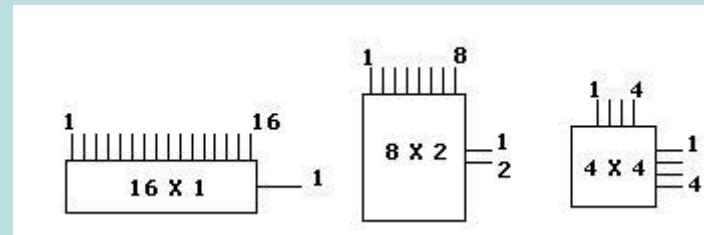
- En la organización 2D se ha visto que todos los bits de una misma palabra estarán localizados dentro de un mismo integrado.
- En la organización 3D esto no tiene porque ocurrir, pudiendo llegar a almacenar hasta un solo bit de cada palabra en un integrado diferente.

Organización Interna

- La principal diferencia consiste en dividir el decodificador único del que dispone la organización 2D, en dos decodificadores.
- Dichos decodificadores seleccionarán una celda de la matriz, que junto con otras “m” celdas que ocupen su misma posición en matrices paralelas, formarán la palabra a la que se desea acceder.

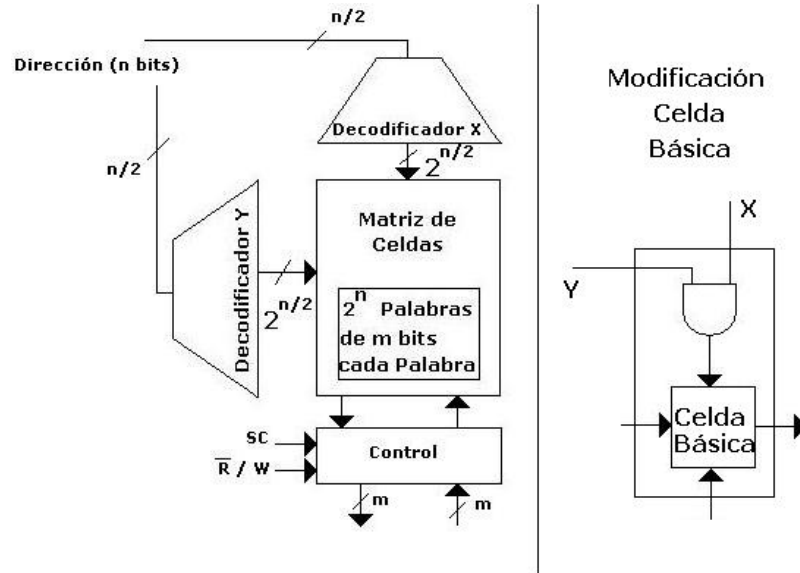


- Si consideramos una matriz de 16 celdas, podemos hacer las siguientes divisiones:



- Evidentemente la mejor configuración es la de 4X4 que es en la que ambos decodificadores son más simples.

Organización Interna

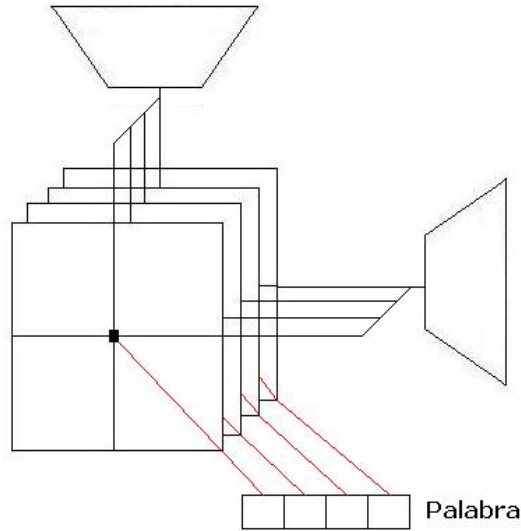


- En la configuración 3D se llama también organización por coincidencia.
- Uno de los decodificadores realiza la selección en X dentro de la matriz y el otro en Y.
- La configuración 3D necesita añadir una puerta NAND en sus celdas básicas

- Para seleccionar un bit de una palabra, la mitad de las líneas de dirección llegan al decodificador Y, que selecciona la fila igual que en 2D, y la otra mitad llegan a X que selecciona la celda básica de memoria a leer o escribir.
- Con lo explicado para la organización 3D, podríamos seleccionar palabras de un solo bit.

Organización Interna

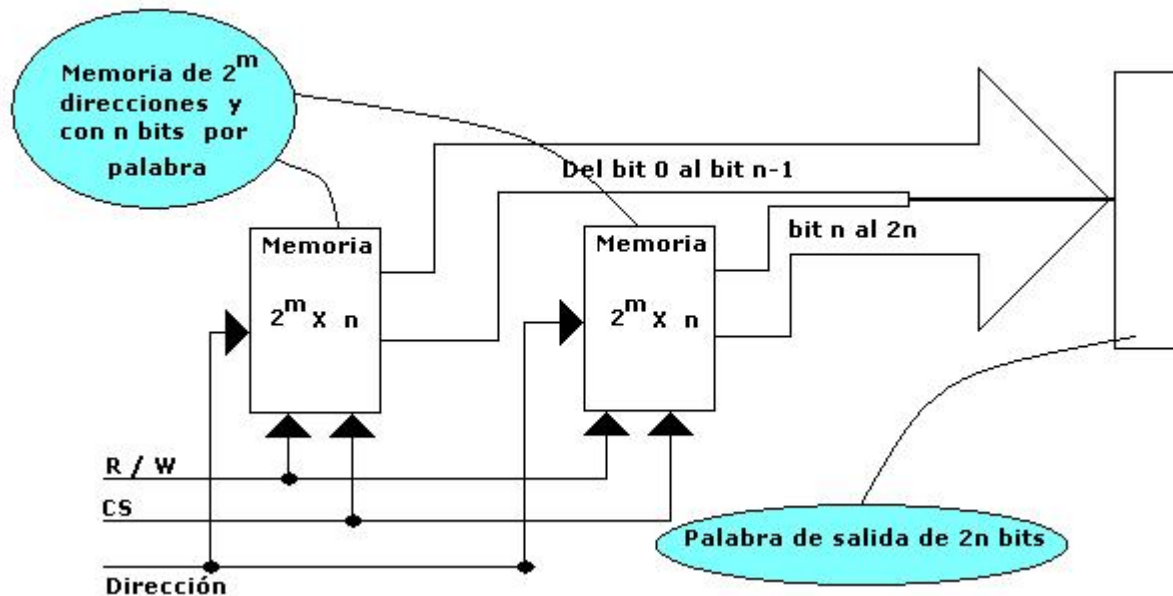
- Para la selección de palabras mayores a un bit, bastaría con repetir la configuración anterior tantas veces como se desee que sea el tamaño de la palabra.



- Con la organización 3D no requiere que las puertas lógicas dispongan de tantas entradas como en la 2D, así como decodificadores de menor tamaño, lo que implica menor número de puertas.
 - Por el contrario, debemos añadir una puerta AND en cada celda básica en la configuración 3D.
- Con la organización 3D no requiere que las puertas lógicas dispongan de tantas entradas como en la 2D, así como decodificadores de menor tamaño, lo que implica menor número de puertas.
 - Por el contrario, debemos añadir una puerta AND en cada celda básica en la configuración 3D.

Diseño de bloques de memoria

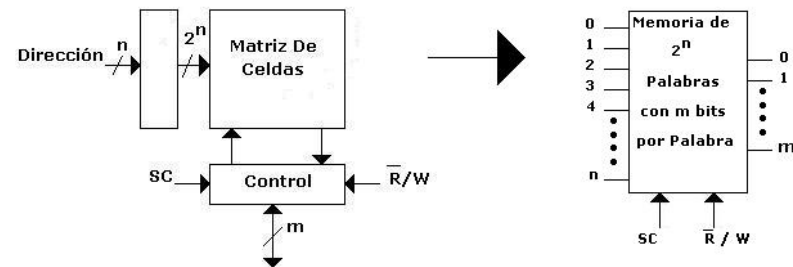
- La ampliación de los bloques de memoria en cualquier sistema persigue dos objetivos:
 - Incrementar el tamaño de las palabras.
 - Incrementar el número de palabras que podemos almacenar.
- El primer apartado es simple, supongamos que disponemos de memorias con un tamaño de palabra de n bits, para conseguir un tamaño doble de palabra bastará con realizar la siguiente interconexión



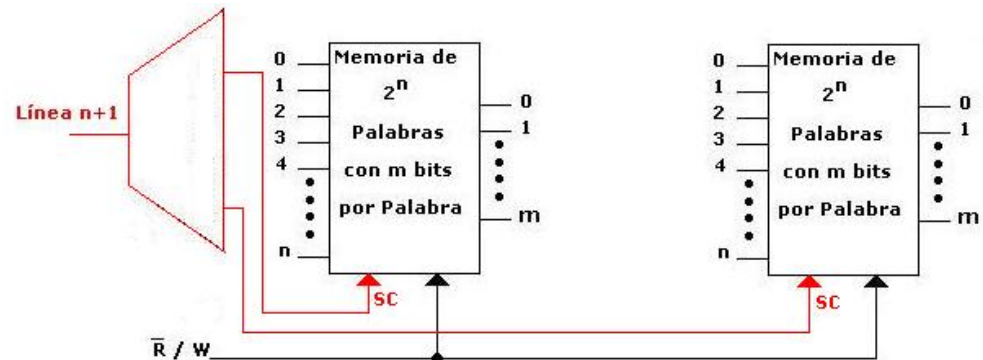
- De la misma forma podemos seguir aumentando el número de bits de la palabra añadiendo más módulos.

Diseño de bloques de memoria

- Incrementar el número de palabras que podemos almacenar supone aumentar el número de líneas de dirección y ello supone la necesidad de añadir decodificadores al sistema.
- Partiendo de una memoria de 2^n palabras, tendrá n líneas de entrada al decodificador



- Para aumentar al doble el número de posiciones de memoria direccionables, bastaría con añadir una línea en el bus de direcciones haciendo la siguiente interconexión



Diseño de bloques de memoria

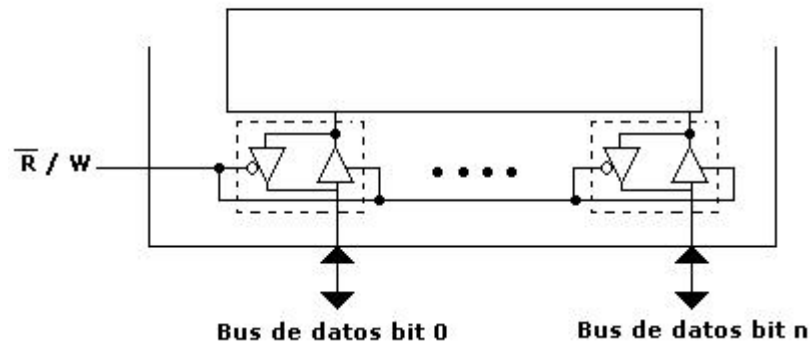
- En general, si añado K líneas de dirección, necesitaremos un decodificador mayor, pudiendo duplicar las posiciones de memoria por cada línea que aumente su bus de direcciones.
- Las dos técnicas vistas pueden usarse de forma simultánea aumentando tanto la longitud de la palabra manejada, como la cantidad de palabras direccionables.

Conexionado entre memoria y bus del sistema

- La memoria dentro de un computador debe estar en permanente comunicación con el procesador.
- Esta conexión la realizaremos a través del bus del sistema que constará por una parte del bus de Datos, el bus de direcciones y el bus de control.
 - En el caso del BUS DE DIRECCIONES, será la CPU la que indique en sus líneas la dirección de la palabra almacenada en la memoria a la que quiere acceder.
 - Con el BUS DE CONTROL, será la CPU la que indique si el acceso se quiere hacer para leer o escribir.
 - A través del BUS DE DATOS, la CPU enviará el dato a escribir hacia la CPU, o leerá el dato de la memoria, dependiendo de si se quiere realizar una operación de lectura o escritura.
- Tanto el caso del Bus de direcciones como el de control será de carácter unidireccional, ya que dicha información partirá siempre de la CPU hacia los módulos de memoria.

Conexión entre memoria y bus del sistema

- El caso del Bus de Datos es más complejo, ya que a veces el dato viajará desde la CPU a la memoria, y en otras ocasiones será al revés, por lo que dicho bus debe ser de carácter bidireccional.



Mapa de memoria Principal

- Se llama mapa de memoria a todas las posibles direcciones de memoria a las que se puede acceder un procesador a través de los buses de dirección, datos y control.
- El número de posiciones de memoria direccionables, depende principalmente del número de líneas del que consta el bus de direcciones, así :

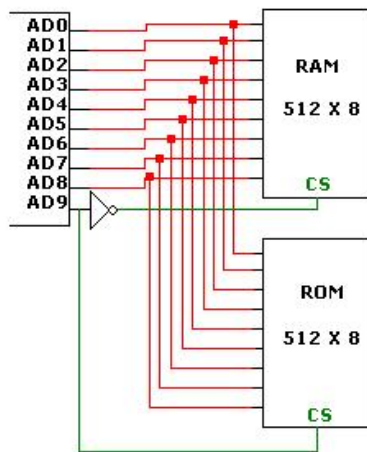
Líneas	Direcciones	Líneas	Direcciones
1	2	11	2048
2	4	12	4096
3	8	13	8192
4	16	14	16384
5	32	15	32768
6	64	16	65536
7	128	17	131072
8	256	18	262144
9	512	19	524288
10	1024	20	1048576

- Dentro del mapa de memoria accesible por una CPU deben estar tanto la memoria a la que éste debe acceder para la correcta ejecución de sus programas como los periféricos con los que intercambiará información con el exterior.

Mapa de memoria Principal

- Ejemplo: Se dispone de una CPU con 10 líneas en su bus de direcciones y 8 en la de datos. Diseñar el mapeado de memoria para localizar una memoria RAM de 512 X 8 bytes a partir de la dirección 0, y tras ella una ROM de otros 512 X 8 bytes.

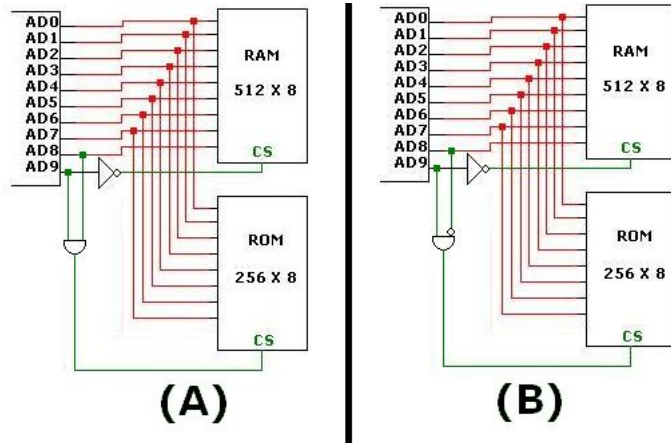
- Con 10 líneas en el Bus de direcciones podemos llegar a direccionar hasta 1 Kb.
- Cada memoria de 512 bytes tendrá 9 líneas de selección.
- La línea sobrante se utilizará para seleccionar entre la RAM y la ROM.



- No se han indicado las líneas R / W que también irán de CPU a las memorias, así como el bus de datos.
- Queda el siguiente mapa de memoria:
 - 000h – 1FFh Memoria RAM.
 - 200h – 3FFh Memoria ROM.

Mapa de memoria Principal

- Este mismo caso pero con la memoria ROM de 256 bytes, podría dar lugar a dos soluciones.



- En la opción A, quedará el siguiente mapa:

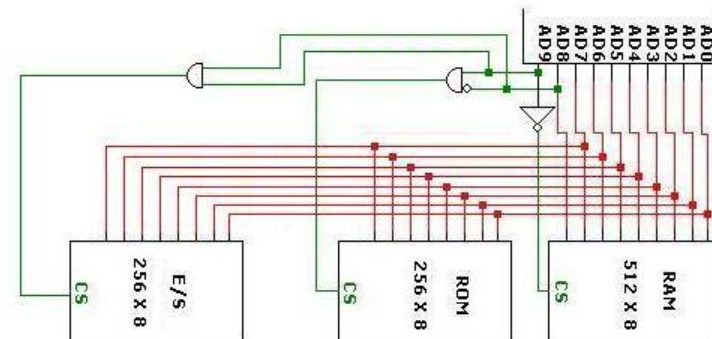
- 000h – 1FFh Memoria RAM.
- 200h – 2FFh Sin Usar.
- 300h – 3FFh Memoria ROM.

- En la opción B, quedará el siguiente mapa:

- 000h – 1FFh Memoria RAM.
- 200h – 2FFh Memoria ROM.
- 300h – 3FFh Sin Usar.

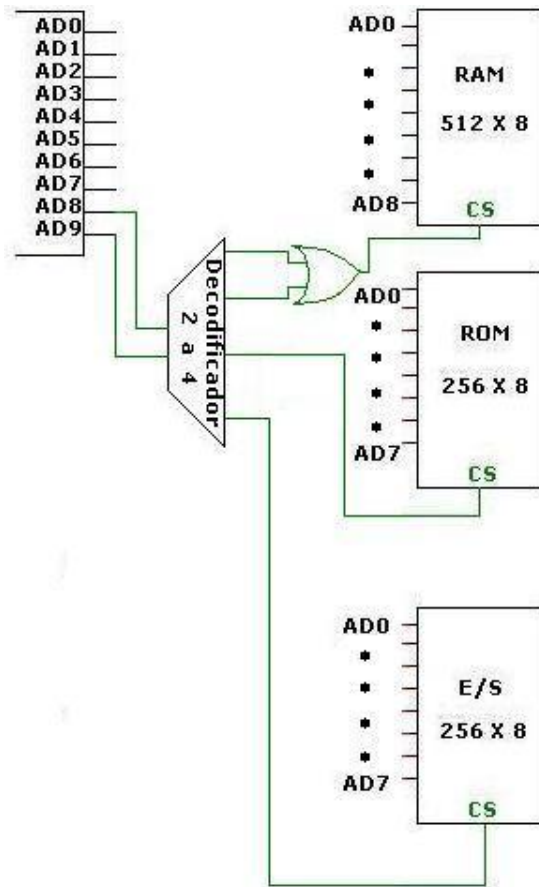
- Si por ejemplo, en el caso B queremos usar el mapa no usado para otra memoria o para periféricos de E/S:

- Análogamente lo podríamos hacer para el caso A.



Mapa de memoria Principal

- Según se aumente las memorias a conectar en un mismo mapa la circuitería se ira haciendo más compleja, por ello merece la pena estudiar la forma de hacer la misma selección con decodificadores.



Problema #1

Capacidad de direccionamiento: 1KB

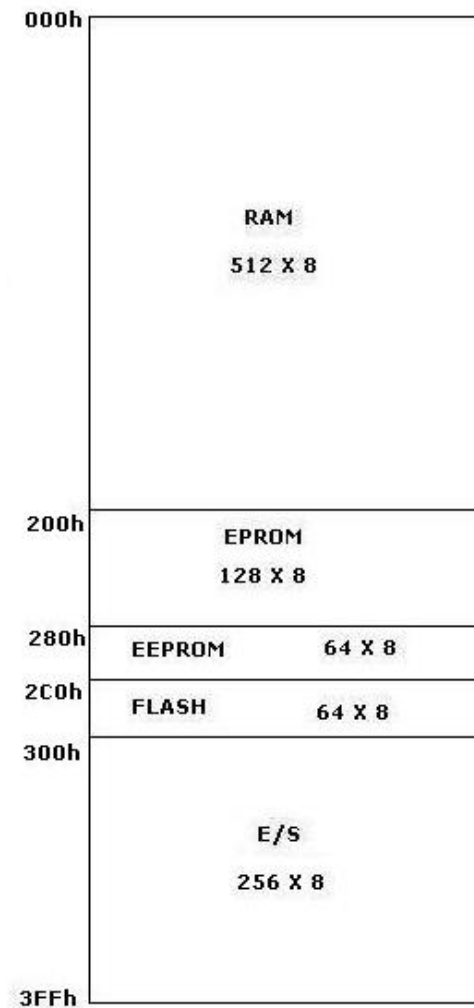
$A_9-A_0=000H$ a $3FFH$

Bloques más pequeños de 64B

A_5-A_0

16 bloques de 64B

A_9-A_6



Mapa de memoria Principal

A ₉	A ₈	A ₇	A ₆	A ₅	A ₀	<u>CS_RAM</u>	<u>CS_EPROM</u>	<u>CS_EEPROM</u>	<u>CS_FLASH</u>	<u>CS_ES</u>
0	0	0	0			1	0	0		
0	0	0	1			1	0	0		
0	0	1	0			1	0	0		
0	0	1	1			1	0	0		
0	1	0	0			1	0	0		
0	1	0	1			1	0	0		
0	1	1	0			1	0	0		
0	1	1	1			1	0	0		
1	0	0	0			0	1	0		
1	0	0	1			0	1	0		
1	0	1	0			0	0	1		
1	0	1	1			0	0	0	1	
1	1	0	0			0	0	0		1
1	1	0	1			0	0	0		1
1	1	1	0			0	0	0		1
1	1	1	1			0	0	0		1

Implementación

<u>CS RAM</u>		$A_7 A_6$			
		00	01	11	10
$A^9 A_8$	00	1	1	1	1
	01	1	1	1	1
	11	0	0	0	0
	10	0	0	0	0

$$CS_RAM = \overline{A^9}$$

Implementación

$$CS_RAM = \overline{A^9}$$

$$CS_EPROM = A^9 \cdot \overline{A^8} \cdot \overline{A^7}$$

$$CS_EEPROM = A^9 \cdot \overline{A^8} \cdot A^7 \cdot \overline{A^6}$$

$$CS_FLASH = A^9 \cdot \overline{A^8} \cdot A^7 \cdot A^6$$

$$CS_ES = A^9 \cdot A^8$$

Problema #2

- Bus de 15 líneas A_{14} - A_0 [32KB]
 - De 0000H a 7FFFH
- ROM de 1KB en dirección 0000H
 - De 0000H a 03FFH
- RAM#1 de 2KB en dirección 1000H
 - De 1000H a 17FFH
- RAM#2 de 8KB en dirección 4000H
 - De 4000H a 5FFFH

Mapa de memoria

		8KB								
		4KB								
		2KB								
				1KB						
		A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₀
ROM	0000H	0	0	0	0	0	0	0	0	0
	03FFH	0	0	0	0	0	1	1	1	1
RAM#1	1000H	0	0	1	0	0	0	0	0	0
	17FFH	0	0	1	0	1	1	1	1	1
RAM#2	4000H	1	0	0	0	0	0	0	0	0
	5FFFH	1	0	1	1	1	1	1	1	1

Implementación

$$ROM = \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}}$$

$$RAM\#1 = \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}}$$

$$RAM\#2 = A_{14} \cdot \overline{A_{13}}$$

[illegible]

¿Simplificación?

		8KB										
		4KB										
		2KB							1KB			
		A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_0
ROM	0000H	0	0	0	0	0	0	0	0	0	0	0
	03FFH	0	0	0	0	0	1	1	1	1	1	1
RAM#1	1000H	0	0	1	0	0	0	0	0	0	0	0
	17FFH	0	0	1	0	1	1	1	1	1	1	1
RAM#2	4000H	1	0	0	0	0	0	0	0	0	0	0
	5FFFH	1	0	1	1	1	1	1	1	1	1	1

$$ROM = \overline{A_{14}} \cdot \overline{A_{12}}$$

$$RAM\#1 = \overline{A_{14}} \cdot A_{12}$$

$$RAM\#2 = A_{14}$$

Mapa

- ¿Si se introduce A_{13} ?

ROM	000 0000 0000 0000 = 0000 _H 000 0011 1111 1111 = 03FF _H
ROM (espejo)	
ROM (espejo)	
ROM (espejo)	
RAM_1	001 0000 0000 0000 = 1000 _H 001 0111 1111 1111 = 17FF _H
RAM_1(espejo)	
ROM (espejo)	
ROM (espejo)	
ROM (espejo)	
ROM (espejo)	
RAM_1(espejo)	
RAM_1(espejo)	
RAM_2	100 0000 0000 0000 = 4000 _H 101 1111 1111 1111 = 5FFF _H
RAM_2 (espejo)	
	111 1111 1111 1111 = 7FFF _H