

uc3m

Universidad
Carlos III
de Madrid



Departamento
Tecnología
Electrónica

Fundamentos en Ingeniería Electrónica

Grado en Ingeniería Electrónica Industrial y Automática, Tecnologías Industriales, Ingeniería Mecánica, Ingeniería de la Energía

Sesión 19: Aplicaciones de circuitos digitales (II). Memorias

Índice

- Terminología, parámetros básicos y tipos.
- Direccionamiento
- Memorias ROM, PROM y EPROM
- Extensión de memorias
- EJEMPLO DE EXTENSION DE MEMORIA
- Síntesis de funciones lógicas con ROM

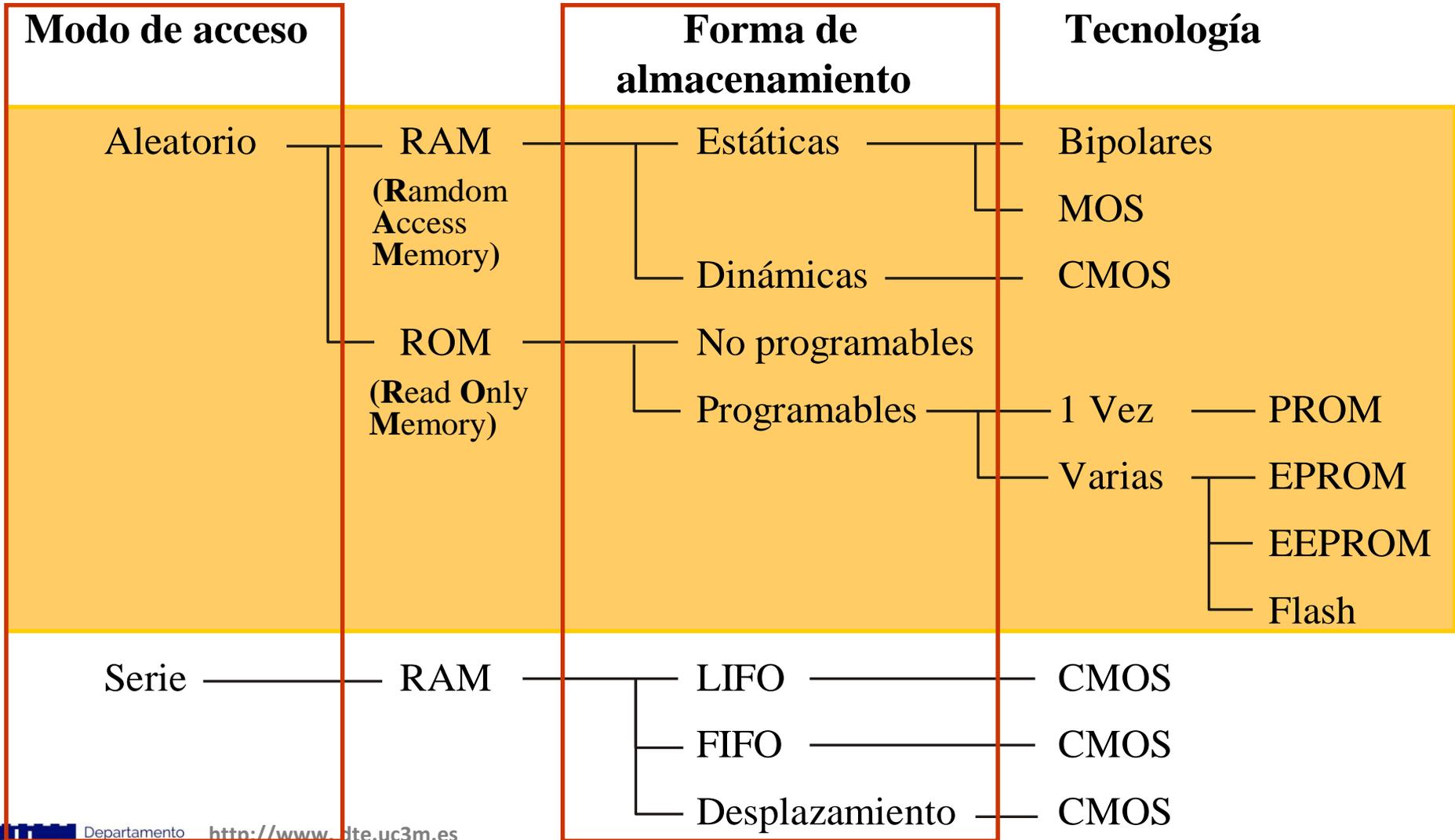
Memorias: Terminología básica

- Elementos de memorización: biestables, registros.
Objetivo: control.
 - La memoria es una estructura de mayor tamaño.
Objetivo: almacenamiento de valores (datos, instrucciones). Su programación está ligada a la tabla de verdad del problema
-
- Bit Elemento mínimo de memorización. Implementado en la célula básica de memoria.
 - Byte Agrupación de 8 bits
 - Word Agrupación de 16 bits
 - Double Word Agrupación de 32 bits
 - Kbyte Equivale a 2^{10} bytes = 1024 bytes
 - Mbyte Equivale a 2^{20} bytes = $1024 \cdot 1024 = 1.048.576$ bytes
 - Gbyte Equivale a 2^{30} bytes = $1024 \cdot 1024 \cdot 1024$ bytes

Memorias: Parámetros básicos

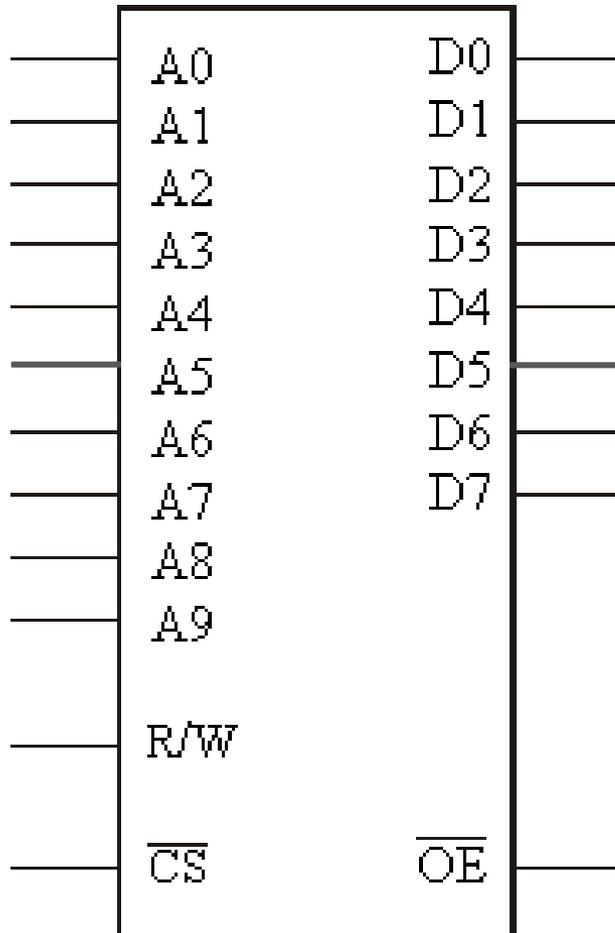
- Soporte físico: Magnéticas, ópticas, sobre semiconductor
- Velocidad
 - Medida considerando el tiempo de acceso, o el tiempo de ciclo
- Consumo: El producto: Retardo x Consumo = Constante
- Capacidad
 - Almacenamiento medido en bits o en Bytes
- Volatilidad
 - Es volátil, si retirando la alimentación se pierde la información almacenada.
- Modo de acceso
 - Secuencial: como en las cintas
 - Aleatorio: el tiempo para acceder a una determinada información es independiente de su ubicación en la memoria
- Coste
 - Precio por unidad de almacenamiento. Es un factor limitante para su utilización, dado su carácter masivo.

Tipos de memorias



Direccionamiento: Entradas/Salidas

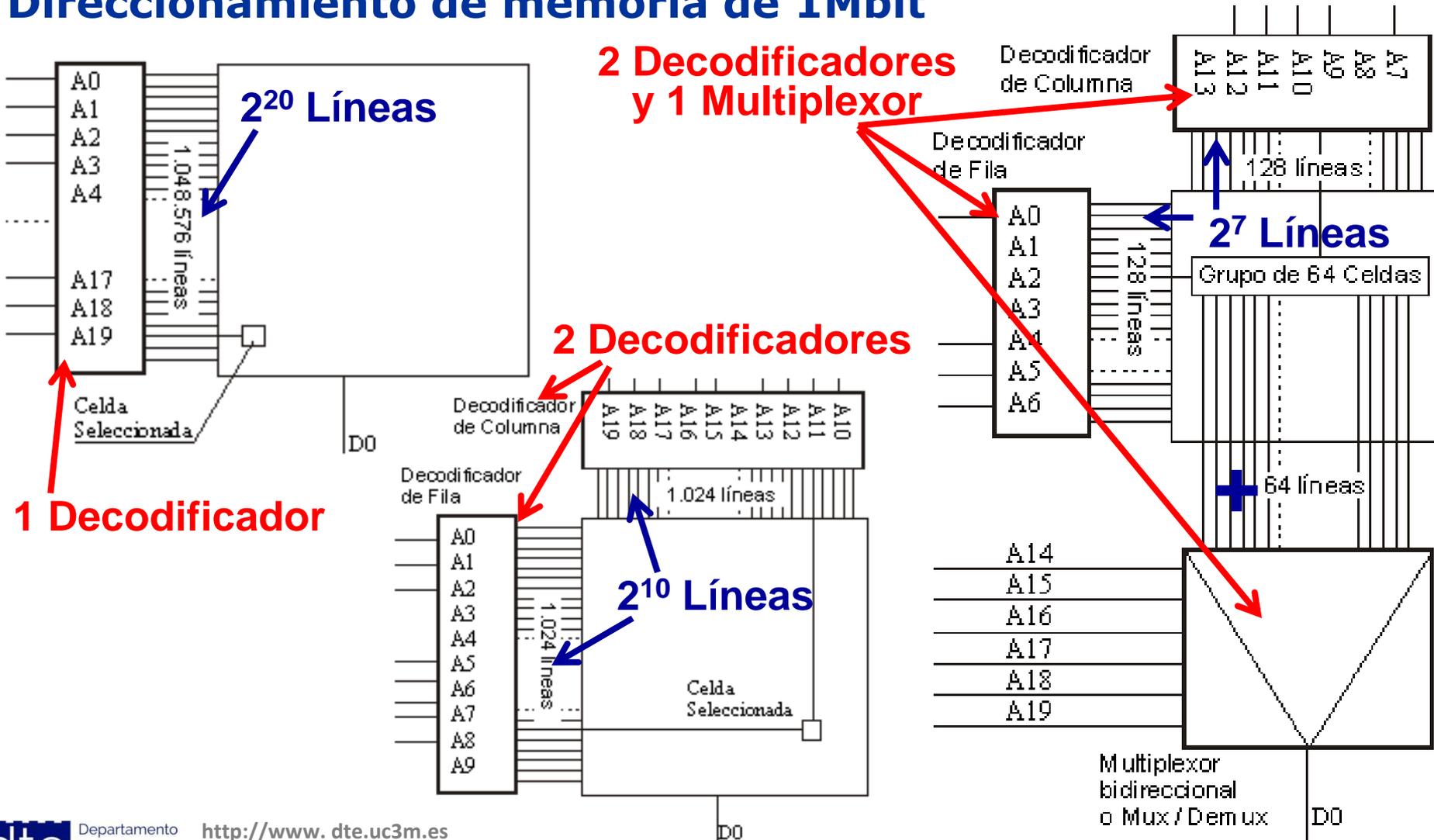
Símbolo de una memoria de 1Kbyte



- Capacidad: 1024 palabras de 8 bits. 8192 bits, o 1K x 8 (nomenclatura)
- A0 a A9: entradas de dirección (address). Se necesitan 10 señales ($2^{10} = 1024$).
- D0 a D7: salidas de datos en lectura, o entradas de datos en escritura. Son, por tanto, bidireccionales, con salida triestado.
- R/W (Read/Write): Entrada que selecciona tipo de operación (lectura o escritura)
- /OE (Output Enable): Controla los buffers triestado de la salida de datos.
- /CS (Chip Select): Deshabilita el chip completo de forma que si no está seleccionada, se ignora cualquier otra señal

Direcccionamiento: Acceso

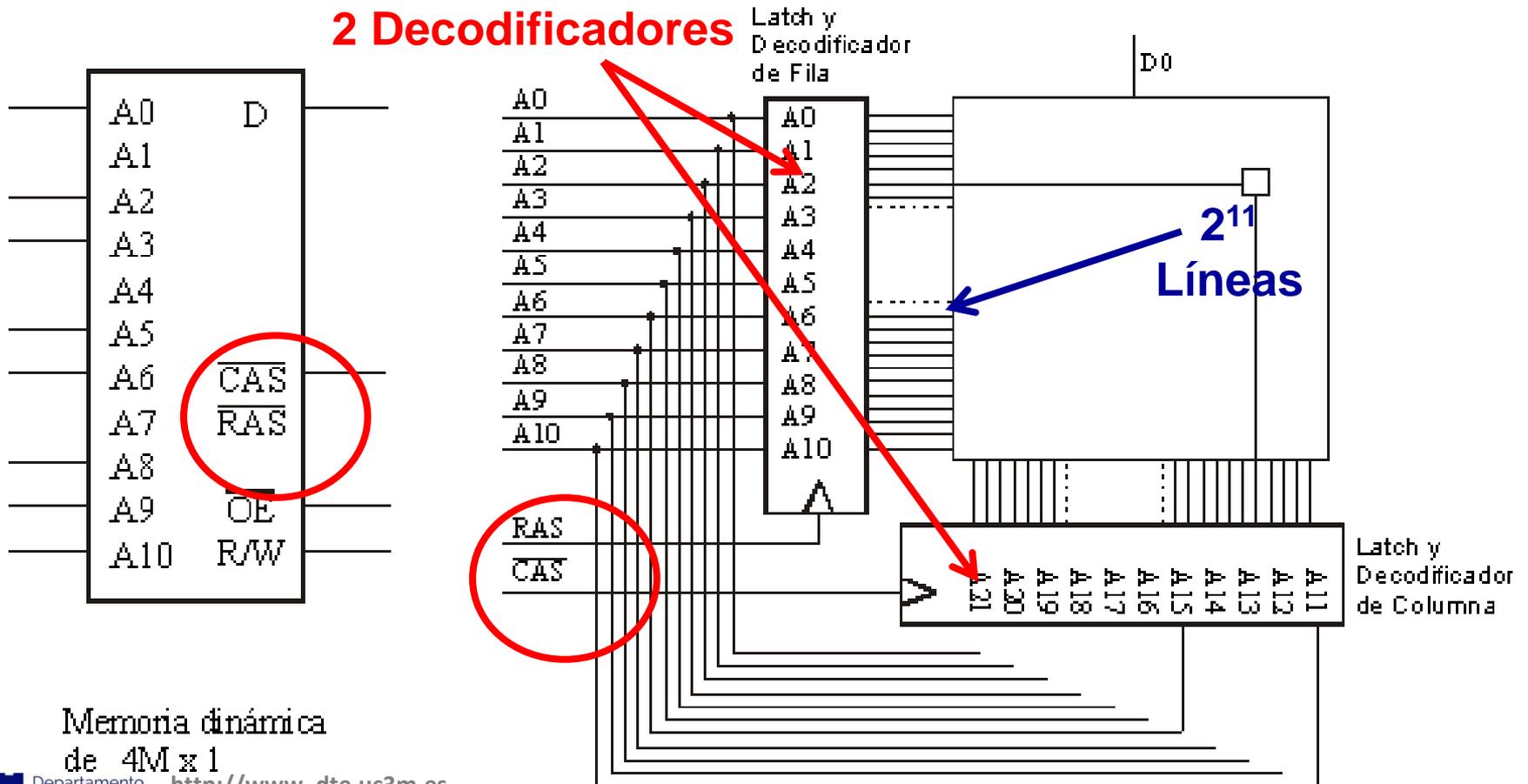
Direcccionamiento de memoria de 1Mbit



Direccionamiento: Acceso

Direccionamiento memoria RAM dinámica 4Mbits

- 22 Líneas direcciones → 11 Líneas direcciones + /CAS + /RAS

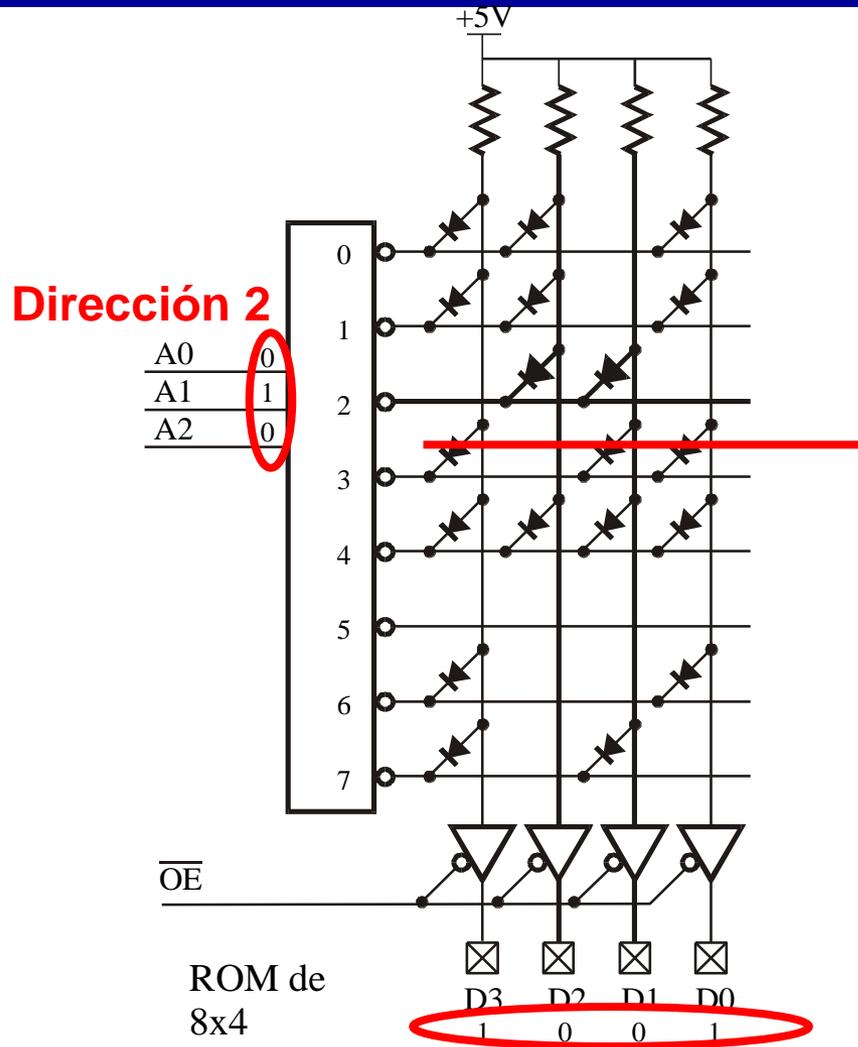


Memorias: ROM, PROM y EPROM

- **ROM : Read Only Memory**
 - Grabadas en fábrica (diodo / no diodo)
 - Altos volúmenes de producción
- **PROM: Programmable Read Only Memory**
 - Menores volúmenes de producción
 - Programables por el usuario
 - Se pueden grabar solo una vez (Diodo + fusible)
- **EPROM: Erasable PROM**
 - Transistor MOS de puerta flotante. Inyección de cargas controladamente.
 - Las cargas se pueden liberar aplicando una luz ultravioleta al silicio (ventana de cuarzo)
- **EEPROM (Electrically EPROM), o E2PROM**
 - Borrada por métodos eléctricos
 - En circuito o fuera de él
- **FLASH**
 - Grabación y borrado bastante rápidos
 - Asimilable a una memoria RAM no volátil

Memorias: Ejemplo ROM

Ejemplo: ROM 8 x 4: Decodificador + Diodos

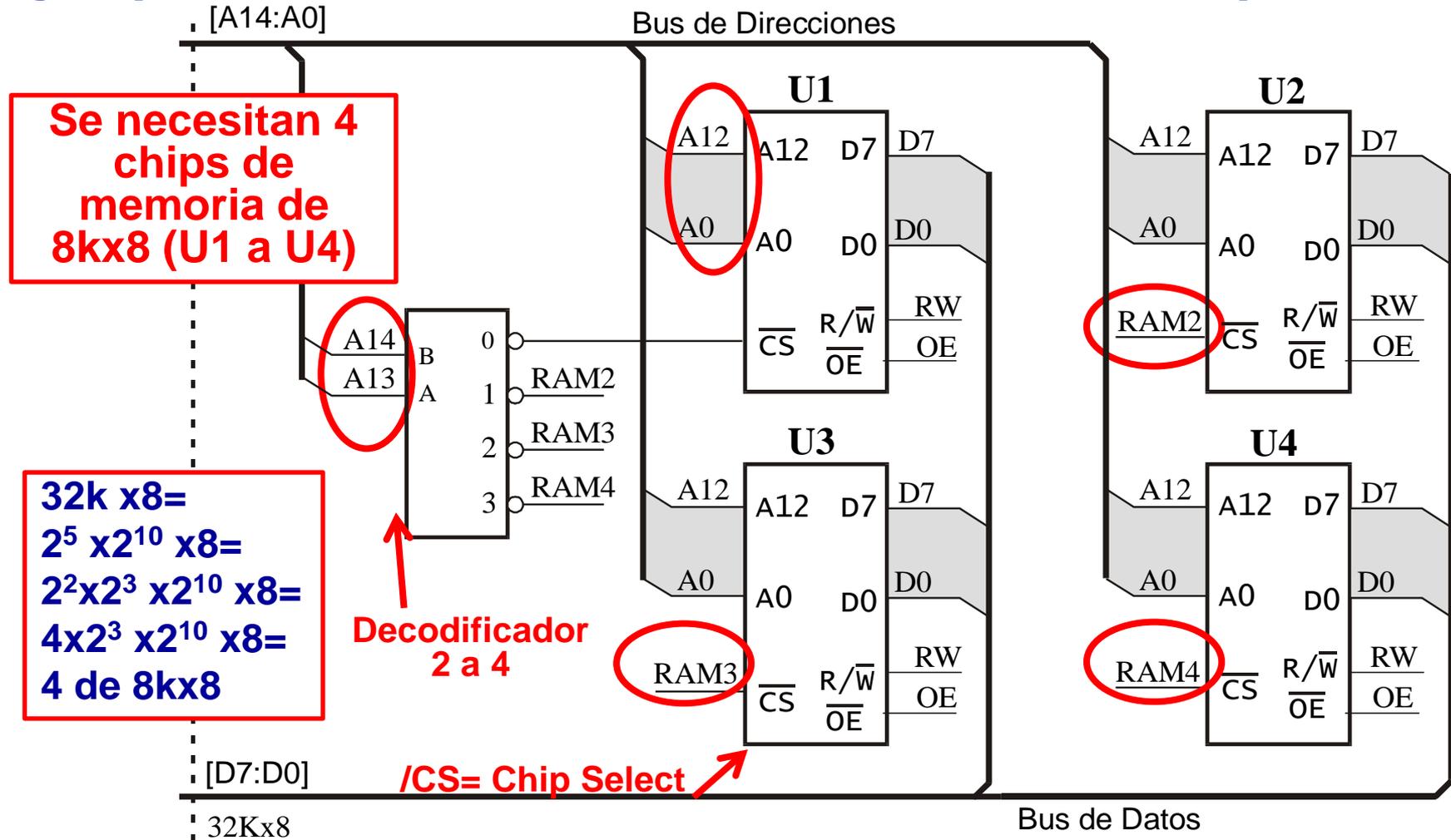


<i>Dirección</i>	<i>Dato</i>
0	2
1	2
2	9
3	4
4	0
5	F
6	6
7	5

Salida = 9

Extensión de memoria (I)

Ejemplo: Diseñar una memoria de 32kx8, con chips de 8kx8



32Kx8

Bus de Datos

Extensión de memoria (I)

Ejemplo (continuación): Mapa de memoria

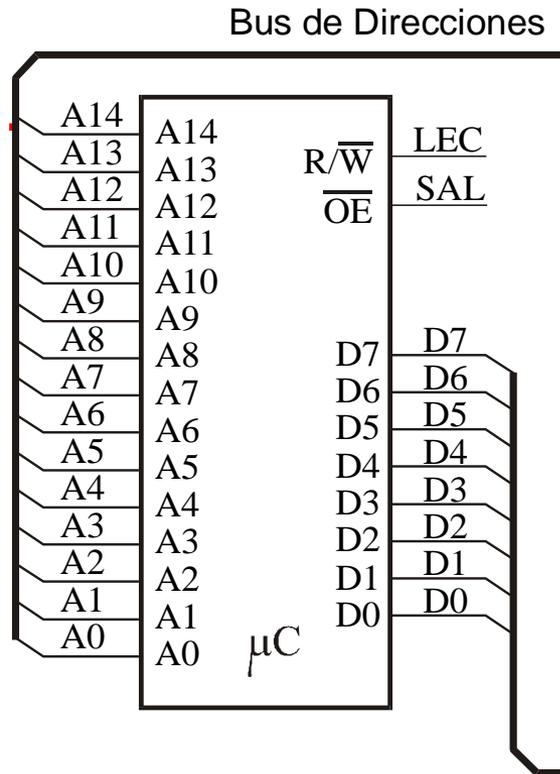
A14 y A13 se utilizan para seleccionar cada uno de los 4 chips

A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Dirección	CHIP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 h	U1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF h	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 h	U2
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF h	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 h	U3
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF h	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000 h	U4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFF h	

Extensión de memoria (I)

Diseñar una memoria para un microprocesador, de 32Kx8, con chips de 16Kx8. Realizar el conexionado entre la memoria y el microprocesador, señalando las líneas bidireccionales.

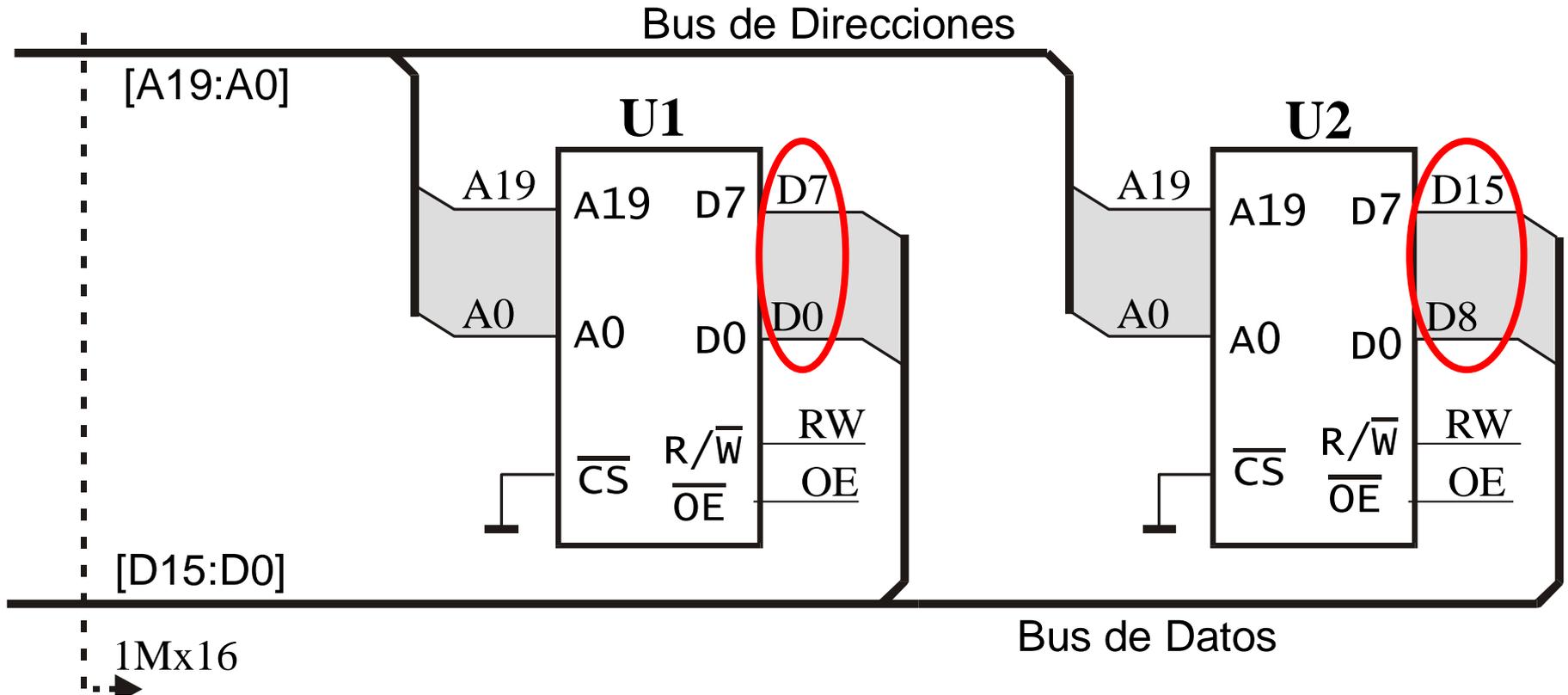
Ejercicio para casa ...



Pista: Símbolo del microprocesador con las señales fundamentales

Extensión de memoria (II)

Ejemplo: Diseñar una memoria de 1Mx16 para un procesador de 16 bits de datos, con chips de 1Mx8



Se amplía el bus de datos que está formado por las señales D15 a D0.

Ej. de extensión de memoria

Queremos diseñar la memoria para un microprocesador de 8 bits con un bus de direcciones de 16 bits.

El equipo de software estima que necesita 32K para el programa en ROM. Sabiendo que el microprocesador empleado busca la primera instrucción de programa, al ser conectado, en la dirección FFFFh, diseñar el mapa de memoria necesario.

Los chips disponibles para la fabricación son: RAM de 8Kx8 y ROM de 16Kx8.

Mapa de memoria:

- Indica las direcciones de comienzo y final de cada circuito integrado
- Las direcciones se indican en hexadecimal

Ej. de extensión de memoria

•SOLUCIÓN

•Se necesitan:

- 2 chips de ROM de 16kx8
- y 4 chips de RAM de 8kx8.

•FFFFh -> 16 bits ->64k:

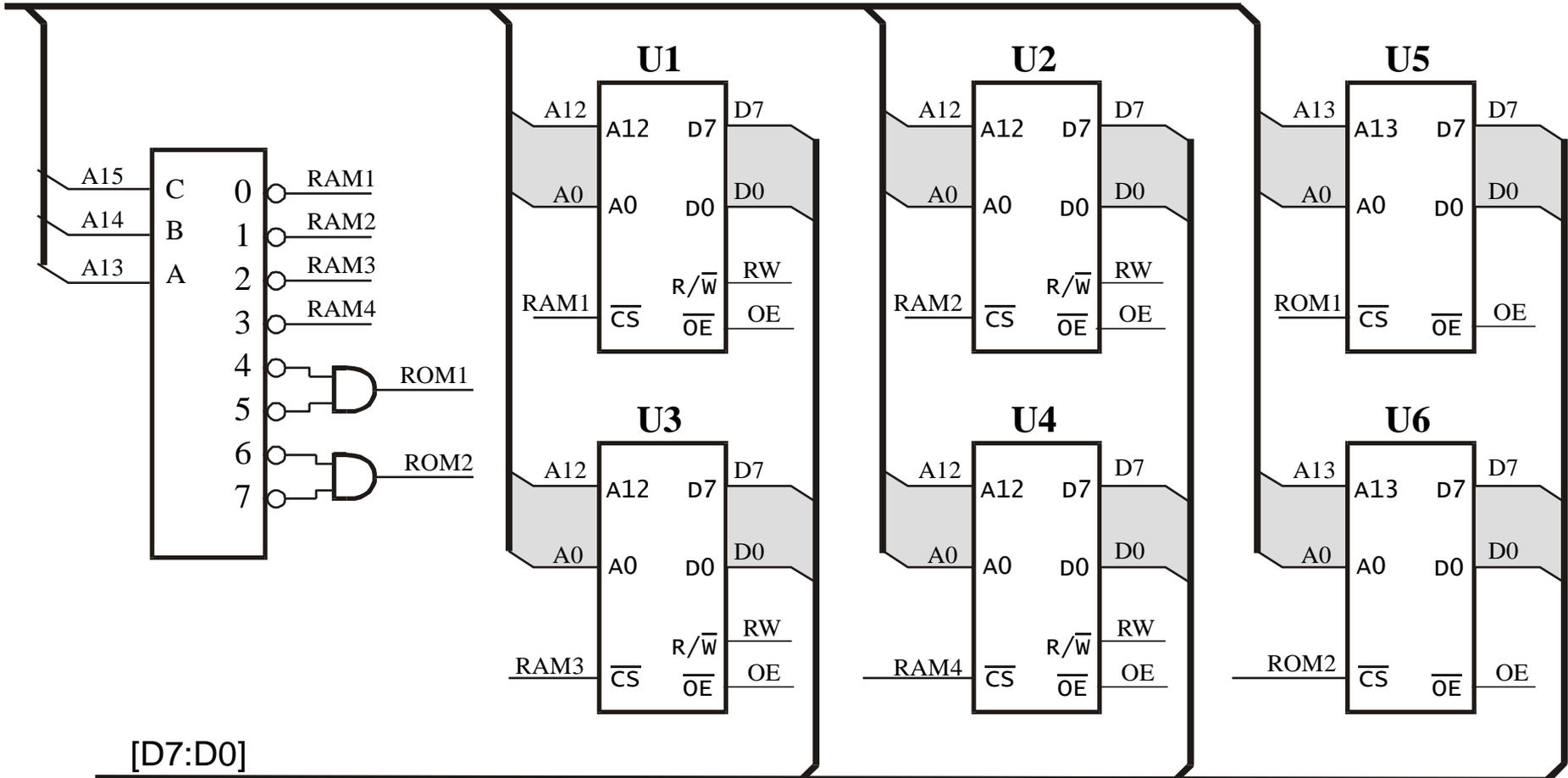
- Dato 32k de ROM, entonces
- 32k RAM (4 chips)

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Dirección	Memoria
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 h	U1, RAM
0	0	0	1... 1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF h	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 h	U2, RAM
0	0	1	1... 1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF h	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 h	U3, RAM
0	1	0	1... 1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF h	
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000 h	U4, RAM
0	1	1	1... 1	1	1	1	1	1	1	1	1	1	1	1	1	7FFF h	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000 h	U5, ROM
1	0	1	1... 1	1	1	1	1	1	1	1	1	1	1	1	1	BFFF h	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000 h	U6, ROM
1	1	1	1... 1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF h	

Ej. de extensión de memoria

•SOLUCIÓN

[A15:A0]



[D7:D0]

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

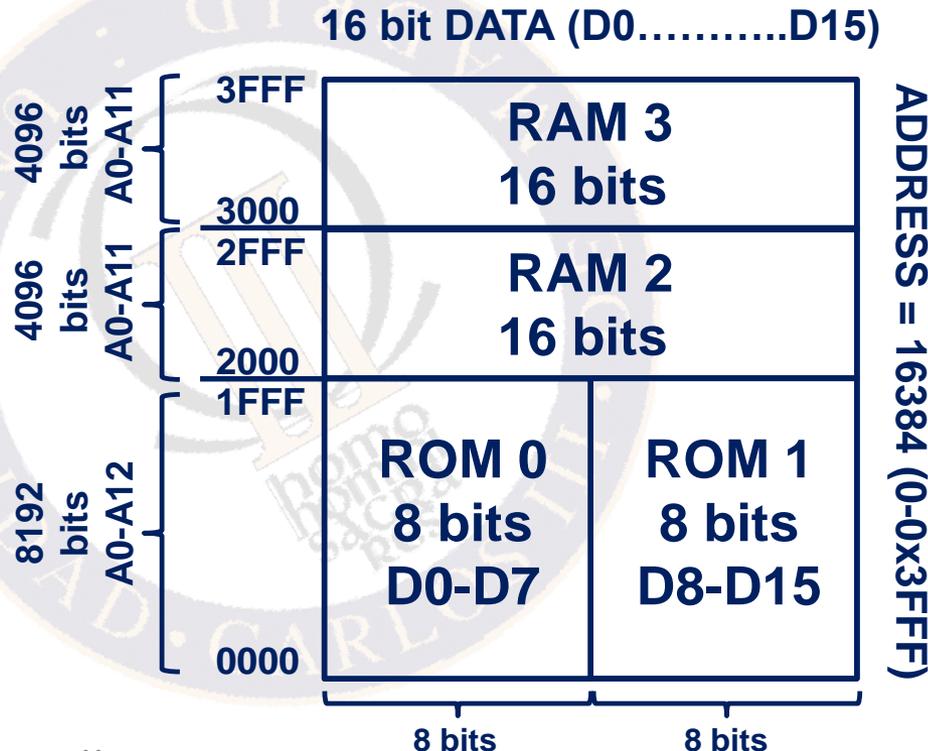
Para el diseño de un controlador industrial se necesita construir un bloque de memoria de **16k x 16** bits, requiriendo que la mitad inferior esté ocupada por memoria **ROM (8k)** para contener el programa de control y el **resto por memoria RAM** para ser utilizada como zona de datos.

Para ello se dispone de chips **ROM de 8k x 8 bits** y **RAM de 4k x 16 bits**.

- ✓ Cada chip ROM dispone de bus de datos, bus de direcciones, \overline{OE} y \overline{CS}
- ✓ Cada chip RAM dispone de bus de datos, bus de direcciones, \overline{OE} , \overline{CS} y R/\overline{W}

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

a) Indique cuántos chips de memoria de cada clase son necesarios y dibuje el mapa de memoria del esquema resultante. Para cada chip deben indicarse claramente las direcciones de memoria inicial y final, y las líneas de datos asignadas.



$$2^{14} = 16384$$

we add 16383 = 0x3FFF

$$2^{13} = 8192$$

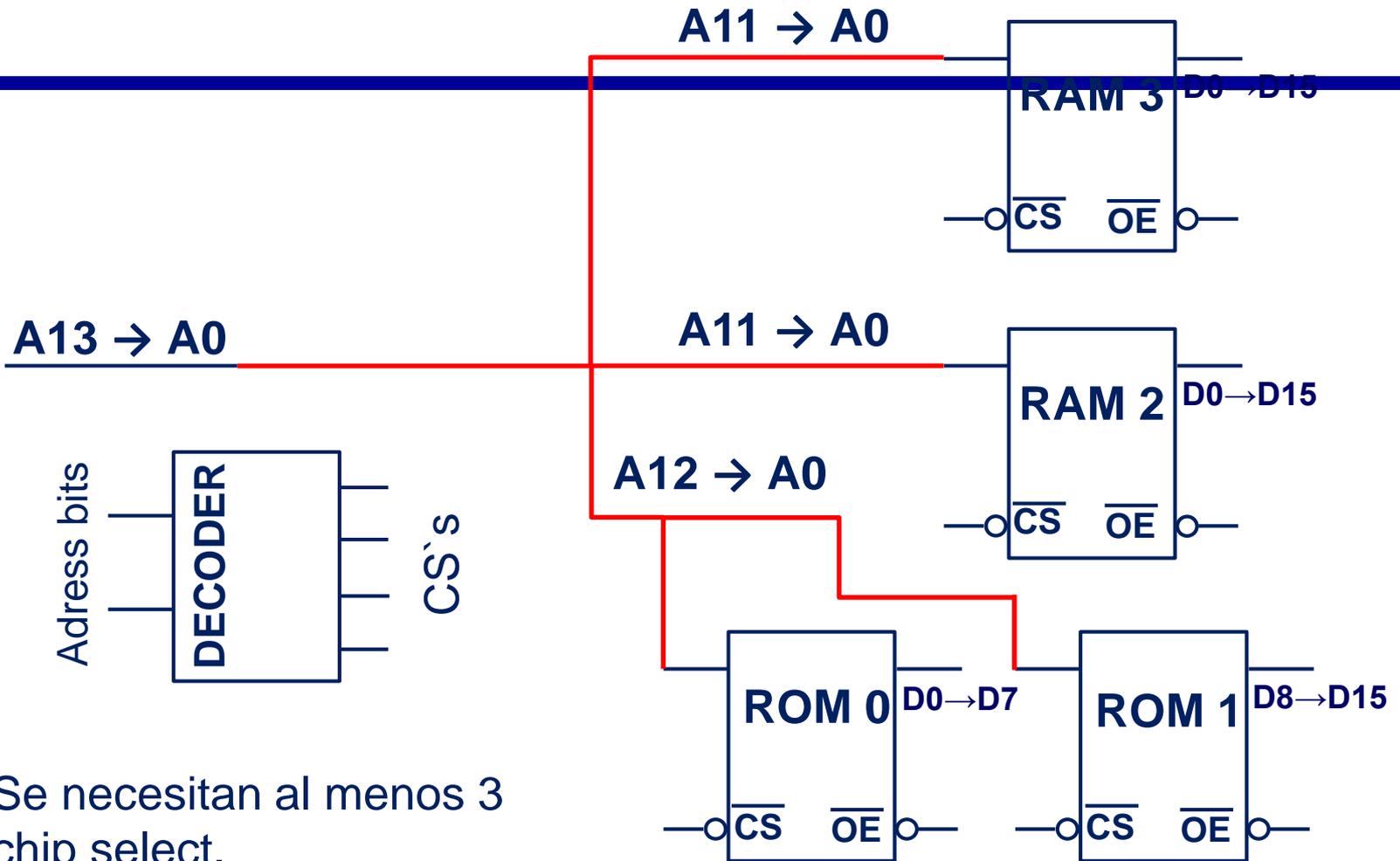
we add 8191 = 0x1FFF

$$2^{12} = 4096$$

we add 4095 = 0x0FFF

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

b) Dibuje un esquema de interconexiones entre todos los elementos necesarios para implementar el bloque de memoria. Debe indicar y etiquetar las líneas de datos, direcciones y control, tanto externas como internas. Suponga que todas las funciones de decodificación internas están agrupadas en un único bloque decodificador que no debe detallar en este momento, sino en el apartado siguiente.



Se necesitan al menos 3 chip select.

¿De dónde se sacan?

De los bits de dirección. ¿Pero cuántos?

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

- Se tienen 3 bloques de direcciones a controlar pero cada uno con una longitud diferente.
- ¿De que tamaño deben ser los bloques a controlar con cada **CS**?
- Se deben generar **CS** para controlar bloques del tamaño del menor bloque de memoria disponible, en este caso es la memoria RAM de 4k direcciones.
- Con esto se tiene que si el total de direcciones a controlar es de 16k y se tiene que el bloque mínimo es de 4k entonces:

$$N^{\circ} \text{ Chip Selects} = \frac{\text{Total memoria a direccionar}}{\text{Menor bloque de memoria}} = \frac{16k}{4k} = 4$$

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

- Nos bastan entonces 2 bit de direcciones para generar los 4 **CS**
- ¿Pero se usan los que ya se tienen o se agregan bits al bus de direcciones?

CS

A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Hex	CS	Dispositivo	Tamaño
0	0	0	0	0	0	0	0	0	0	0	0	0			ROM 0 + ROM 1	8k
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓				
0	1	1	1	1	1	1	1	1	1	1	1	1				
1	0	0	0	0	0	0	0	0	0	0	0	0				
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓				
1	1	1	1	1	1	1	1	1	1	1	1	1				
0	0	0	0	0	0	0	0	0	0	0	0	0			RAM 2	4k
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓				
0	1	1	1	1	1	1	1	1	1	1	1	1				
1	0	0	0	0	0	0	0	0	0	0	0	0			RAM 3	4k
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓				
1	1	1	1	1	1	1	1	1	1	1	1	1				

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

Se utilizarán entonces los bits A13 y A12 para generar los 4 Chip Selects

A13	A12	CS0	CS1	CS2	CS3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- ✓ La ROM ocupa 8k (4k+4k) con lo cual se debe poder seleccionar ambos chips con el CS0 o CS1 del decodificador
- ✓ La RAM 2 se seleccionará con CS2
- ✓ La RAM 3 se seleccionará con CS3

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:

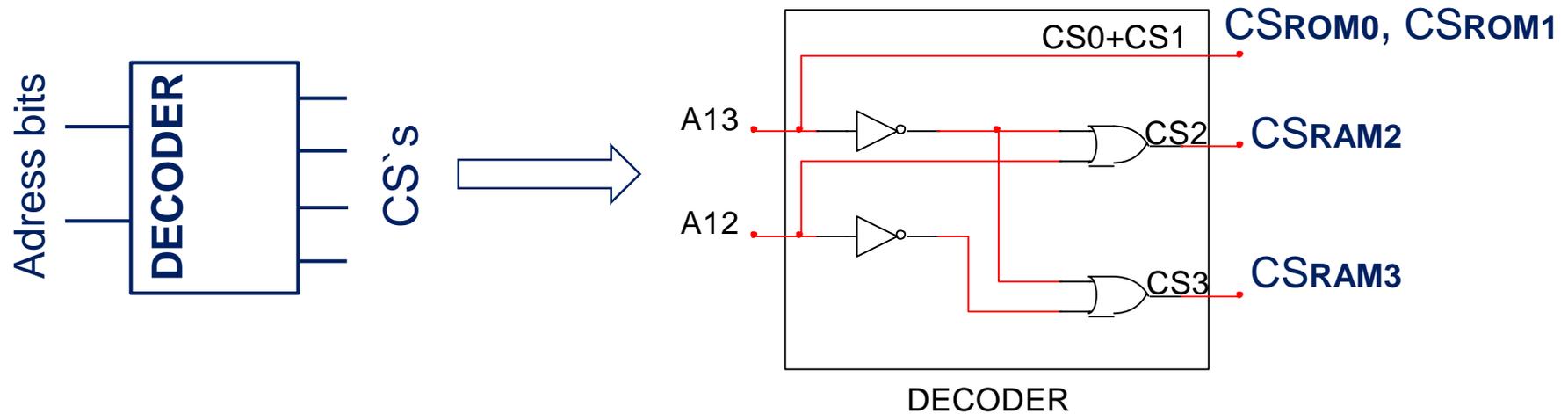
Con lo cual se tiene:

$$CS_{ROM} = CS0 + CS1 = \overline{A13} * \overline{A12} + \overline{A13} * A12 = \overline{A13} \rightarrow \overline{CS_{ROM}} = A13$$

$$CS_{RAM2} = CS2 = A13 * \overline{A12} \rightarrow \overline{CS2} = \overline{A13} + A12$$

$$CS_{RAM3} = CS3 = A13 * A12 \rightarrow \overline{CS3} = \overline{A13} + \overline{A12}$$

PROBLEMA 35 COLECCIÓN DE PROBLEMAS:



Síntesis de funciones lógicas

Implementar las funciones lógicas con una memoria ROM de 256 x 4

$$f_0 = \sum_8 (1, 3, 69, 232)$$

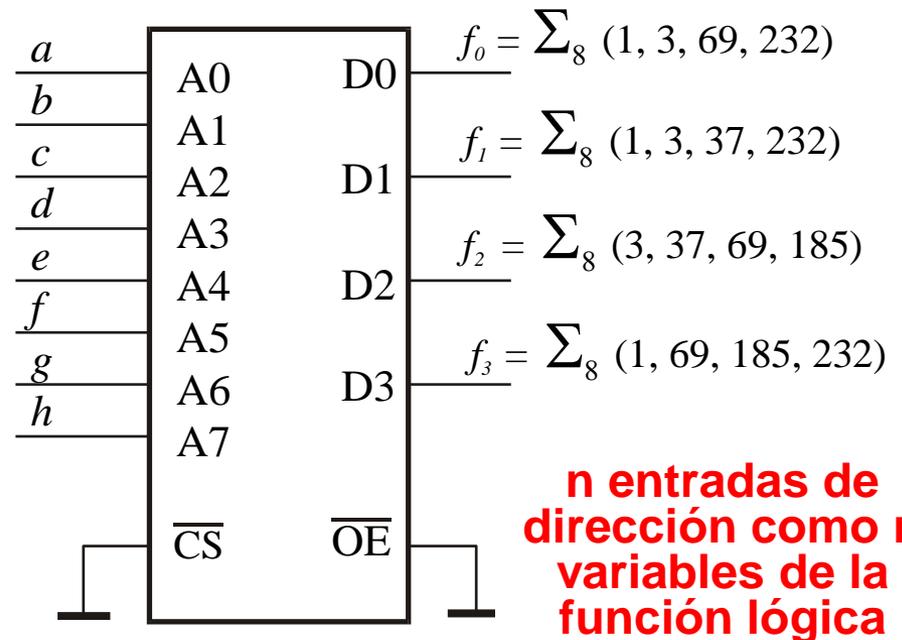
$$f_1 = \sum_8 (1, 3, 37, 232)$$

$$f_2 = \sum_8 (3, 37, 69, 185)$$

$$f_3 = \sum_8 (1, 69, 185, 232)$$

Si todos los bits de la memoria están a cero, antes de ser grabada, indicar qué direcciones hay que grabar, y con qué contenido hexadecimal, (f3 se obtiene con la salida de datos D3, f2 con la salida D2..)

Dirección	Dato				Dato HEX
	D3	D2	D1	D0	
0	0	0	0	0	0
1	1	0	1	1	B
2	0	0	0	0	0
3	0	1	1	1	7
	0	0	0	0	0
37	0	1	1	0	6
	0	0	0	0	0
69	1	1	0	1	D
	0	0	0	0	0
185	1	1	0	0	C
	0	0	0	0	0
232	1	0	1	1	B
255	0	0	0	0	0



f_0

Ej. para trabajar en casa

PREGUNTAS

Se dispone de una memoria PROM de capacidad 8x4 bits. Se desea implementar las funciones lógicas siguientes utilizando la memoria disponible.

$$f_1 = \sum_3 (1, 3, 6) \quad f_2 = \sum_2 (0, 2) \quad f_3 = \sum_3 (4, 7)$$

- ¿Cuántas líneas de direcciones tiene la memoria?
- ¿Es posible implementar la función f_1 con la memoria disponible? ¿Y la función f_2 ? ¿Es posible implementar simultáneamente las tres funciones f_1 , f_2 y f_3 ? Justifique las respuestas describiendo cómo lo haría.

Ej. para trabajar en casa

PREGUNTAS

c) Si se pudieran implementar las tres funciones f_1 , f_2 y f_3 , simultáneamente, rellene la tabla adjunta con los datos que deberían grabarse en cada una de las posiciones de la memoria. Suponga que utiliza las líneas D_1 , D_2 y D_3 , respectivamente. Exprese los datos en binario y en hexadecimal.

Dirección (HEX)	Datos (BINARIO)				Datos (HEX)
	D_3	D_2	D_1	D_0	
0					
1					
2					
3					
4					
5					
6					
7					

Ej. para trabajar en casa

SOLUCIÓN

Como la memoria es de capacidad 8x4 bits, quiere decir que posee 8 posiciones de memoria en las cuales los datos son de 4 bits. Para direccionar 8 posiciones de memoria se necesitan 3 líneas de dirección ($2^3 = 8$).

$$f_1 = \sum_3 (1, 3, 6) \quad f_2 = \sum_2 (0, 2) \quad f_3 = \sum_3 (4, 7)$$

La función f_1 es de 3 variables por lo que se puede implementar considerando las 3 líneas de dirección de la memoria como las 3 variables de entrada. Por ejemplo, se puede tomar como terminal de implementación de f_1 , la línea de datos D_1 .

La función f_2 es de 2 variables por lo que se puede implementar considerando sólo 2 líneas de dirección de la memoria como las 2 variables de entrada. Por ejemplo, se puede tomar como terminal de implementación de f_2 , la línea de datos D_2 .

Las funciones f_1 , f_2 y f_3 se pueden implementar simultáneamente porque la memoria posee 4 líneas de datos, pudiéndose implementar hasta 4 funciones diferentes, una por cada línea de datos.

Ej. para trabajar en casa

SOLUCIÓN

Dirección (HEX)	Líneas Dirección $A_2A_1A_0$	Datos (BINARIO)				Datos (HEX)
		D_3	D_2	D_1	D_0	
0	0 0 0	0	1	0	0	4
1	0 0 1	0	0	1	0	2
2	0 1 0	0	1	0	0	4
3	0 1 1	0	0	1	0	2
4	1 0 0	1	1	0	0	C
5	1 0 1	0	0	0	0	0
6	1 1 0	0	1	1	0	6
7	1 1 1	1	0	0	0	8

f_3 f_2 f_1

Una posible implementación es la incluida en la tabla, donde se ha asignado al bit de datos D_0 siempre el valor '0'.

Ej. para trabajar en casa

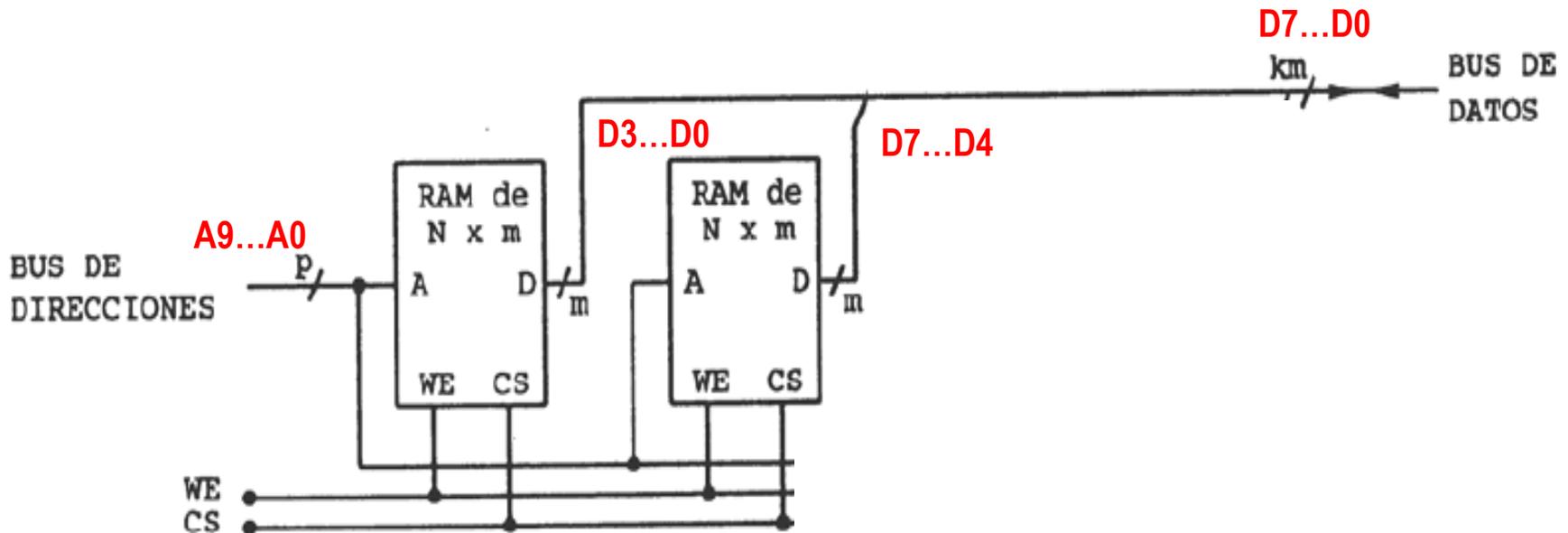
SOLUCIÓN

Diseñe una memoria de 1kx8. Se dispone de memorias de 1kx4.

1kx8 → 1024 palabras de 8 bits → 10 bits bus direccionamiento, A9...A0

1kx4 → 1024 palabras de 4 bits → 10 bits bus direccionamiento, A9...A0

Las salidas de datos de ambas memorias se integran en bus de datos de 8 bits, D7...D0, necesitamos 2 memorias



Ej. para trabajar en casa

PREGUNTAS

- Diseñe una memoria de 4kx4. Se dispone de memorias de 1kx4.
- Diseñe una memoria de 1kx8. Se dispone de memorias de 1kx4.

Ej. para trabajar en casa

SOLUCIÓN

Diseñe una memoria de 4kx4. Se dispone de memorias de 1kx4.

Se dispone de integrados de 1k de capacidad, que se direccionan con:

$$2^{10} = 1024 \rightarrow 10 \text{ líneas de direcciones}$$

Estas líneas de direcciones se numeran de la más significativa a la menos significativa como: A9, A8, A7, etc. hasta la A0.

Como se pretende direccionar una memoria de 4k palabras de capacidad, se necesitan:

$$4 \cdot 2^{10} = 2^2 \cdot 2^{10} = 2^{12} \rightarrow 12 \text{ líneas de direcciones}$$

Es decir, se necesitan 4 memorias de capacidad 2^{10} (1k) y además es preciso añadir dos líneas nuevas de direcciones, (respecto de las 10 líneas para integrados de 1k), que llamamos A11 y A10.

Estas dos líneas nuevas de dirección, se pueden utilizar para distinguir cual de las 4 memorias de 1k se esta direccionando. Una forma sencilla de hacerlo es utilizar un decodificador. El decodificador permite seleccionar cada uno de los integrados, de forma que únicamente uno de ellos este activo en cada instante.

En este caso se utiliza un decodificador 2 a 4, con salidas activas a nivel bajo. Las entradas del decodificador se conectan a las líneas A11 y A10 y sus salidas decodificadas a las entradas de selección de cada chip de memoria RAM.

Ej. para trabajar en casa

SOLUCIÓN

A11	A10	RAM seleccionada
0	0	Chip 0
0	1	Chip 1
1	0	Chip 2
1	1	Chip 3

