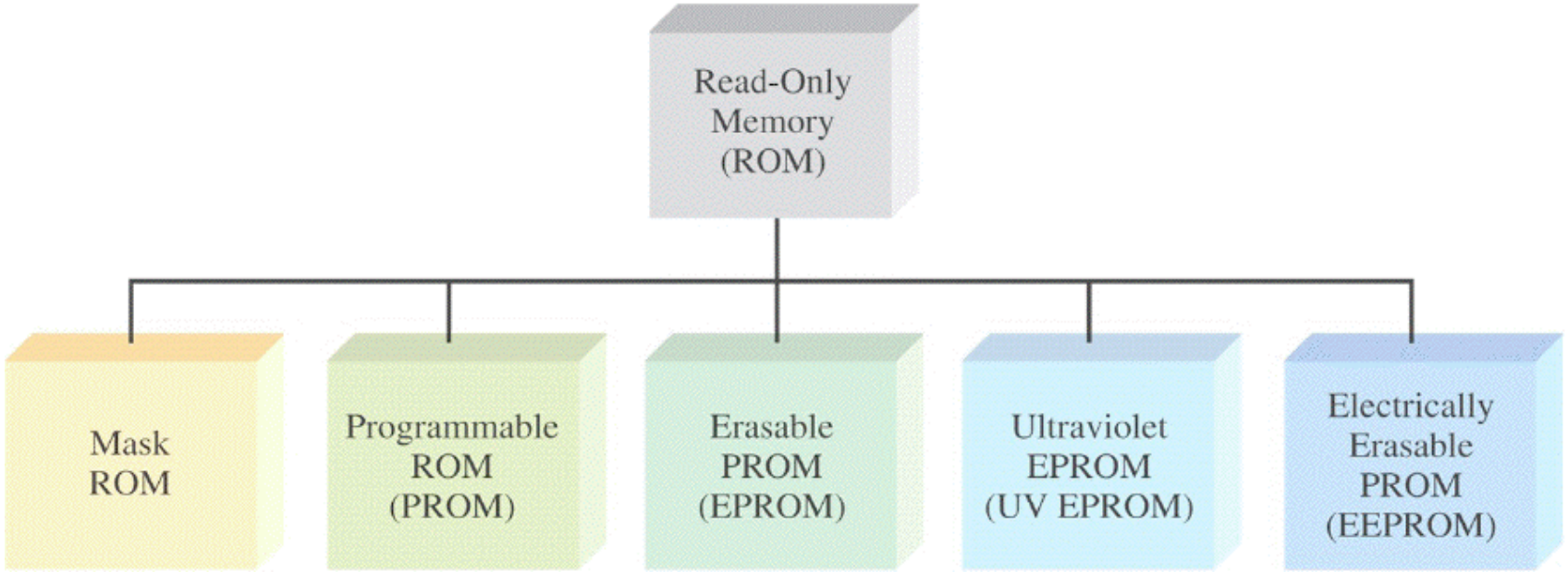
TEMA 2

Memorias ROM, PROM, EPROM y E2PROM. Memorias FLASH. Memorias FIFO y LIFO. Diseño de circuitos lógicos empleando PROMs. Test de memorias

Memorias ROM

- **Memorias de almacenamiento permanente**
- **Memorias de sólo lectura**



Memorias ROM

ROM ("Read-Only Memory")

- **Almacenamiento de datos durante el proceso de fabricación**
- **Almacenamiento por máscara**
- **MOS o bipolar (BJT)**

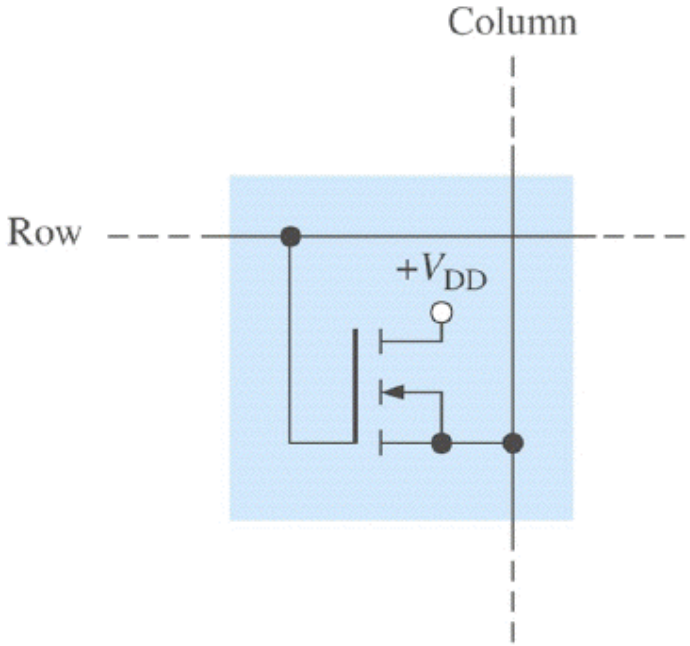
PROM ("Programmable ROM")

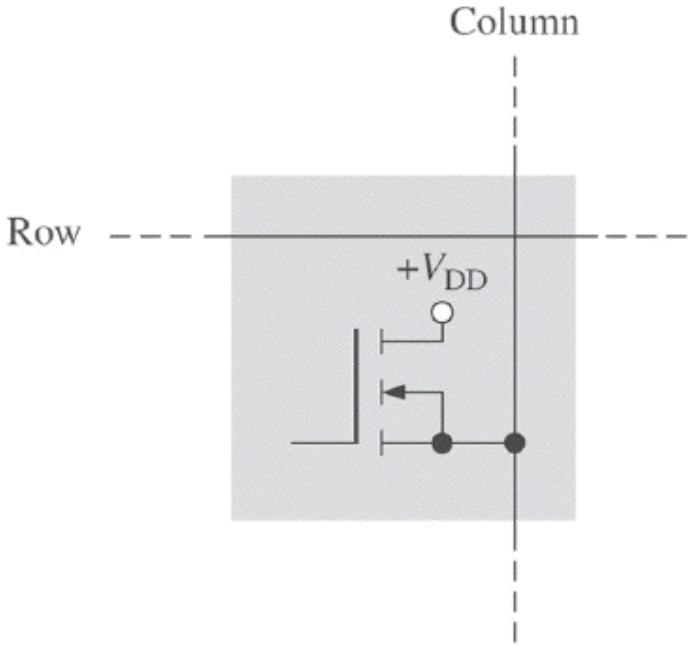
- **Programable por el usuario con ayuda de equipos especializados**
- **Almacenamiento eléctrico**
- **MOS o bipolar (BJT)**

EPROM ("Erasable PROM")

- **Programable por el usuario con ayuda de equipos especializados**
- **Almacenamiento eléctrico**
- **MOS**
- **Borrables**
  - **UV EPROM**
    - **Programable eléctricamente**
    - **Borrable por exposición a luz ultravioleta**
  - **EEPROM o E^2PROM "Electrically Erasable PROM")**
    - **Programable y borrable eléctricamente ("in situ")**

Celda ROM básica





Almacenamiento de un '1' lógico

Almacenamiento de un '0' lógico

Si existe la conexión de "Gate"

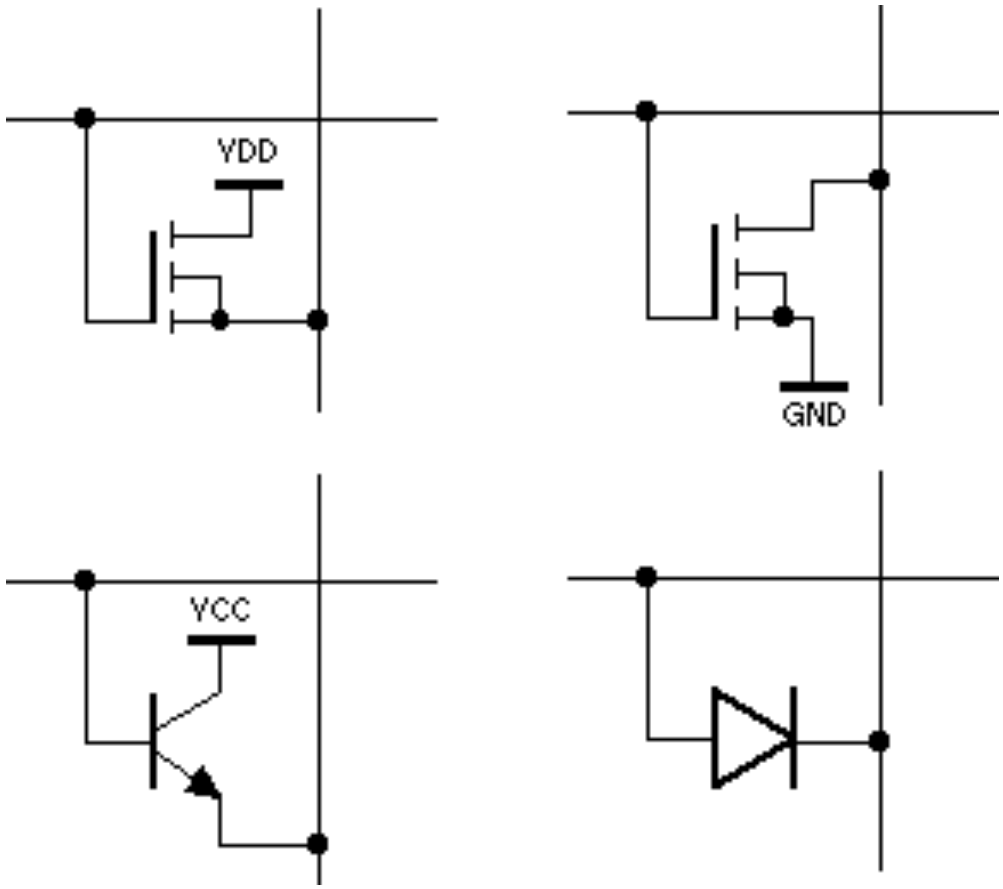
Cuando la línea de fila toma nivel H, el transistor se satura y la línea de columna se pone a nivel H

Cuando la línea de fila toma nivel L, el transistor se corta y la línea de columna se pone a nivel L

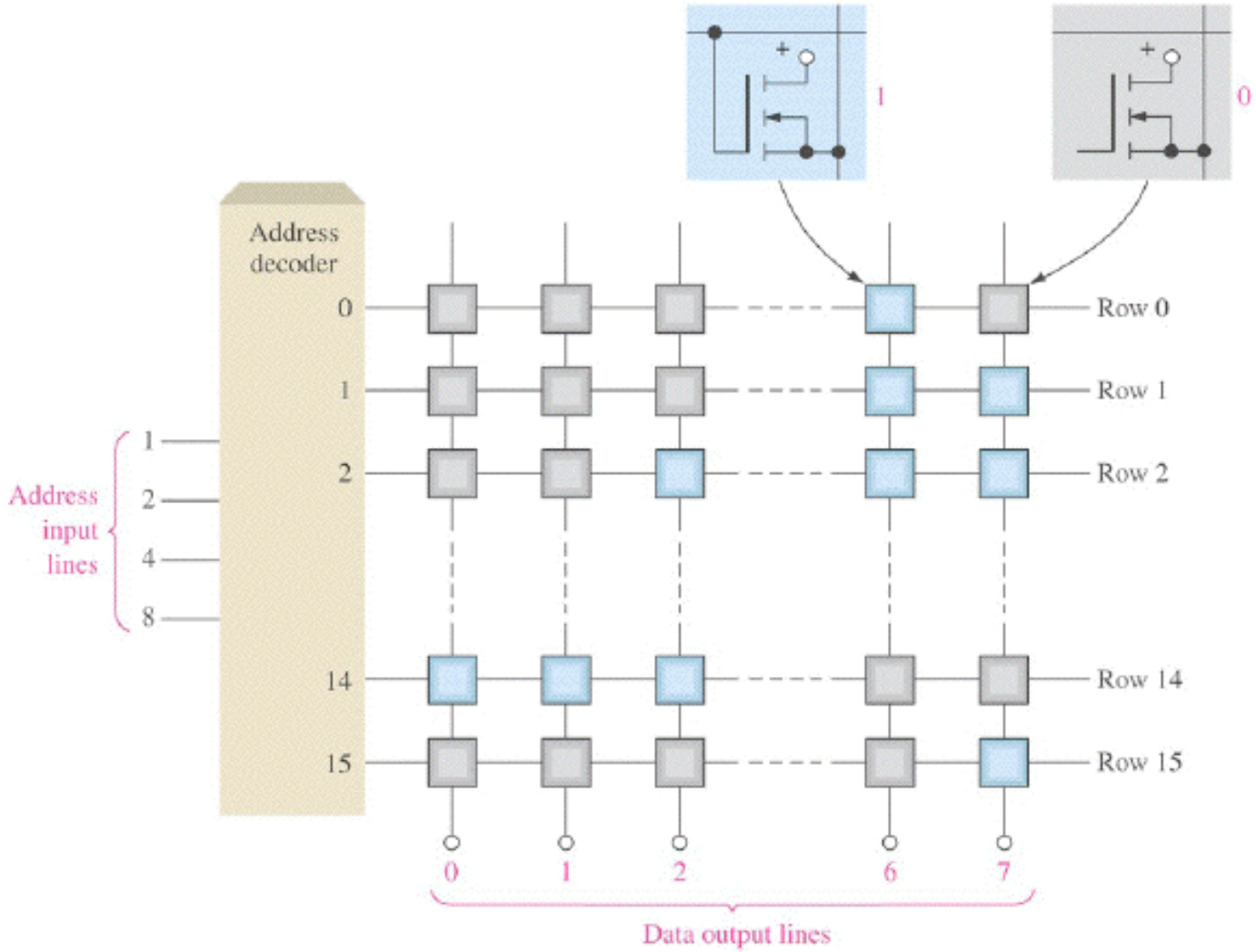
Si no existe la conexión de "Gate"

El transistor no conduce y la línea de columna se pone a nivel L

Tipos de celdas



Estructura ROM básica



16 x 8

Aplicación al diseño digital

Ejemplo

Implementar de un conversor de Código NBCD a Código de Gray

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

$\Leftrightarrow$

$$G_3 = \sum(8,9) + \sum_{\phi}(10,11,12,13,14,15)$$

$$G_2 = \sum(4,5,6,7,8,9) + \sum_{\phi}(10,11,12,13,14,15)$$

$$G_1 = \sum(2,3,4,5) + \sum_{\phi}(10,11,12,13,14,15)$$

$$G_0 = \sum(1,2,5,6,9) + \sum_{\phi}(10,11,12,13,14,15)$$

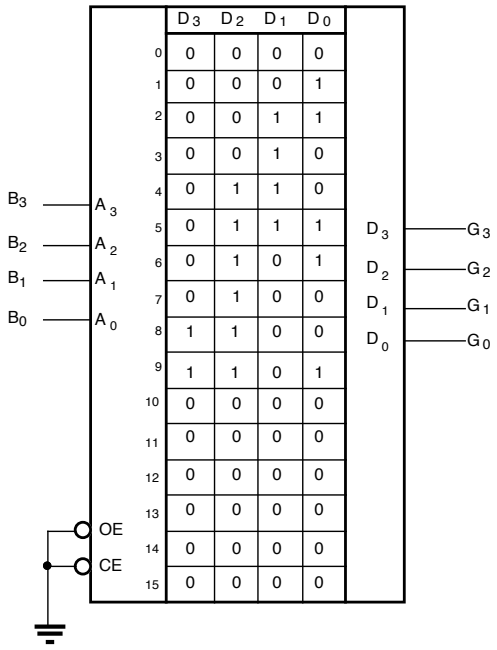
Aplicación al diseño digital (2)

$$G_3 = \sum(8,9) + \sum_{\phi}(10,11,12,13,14,15)$$

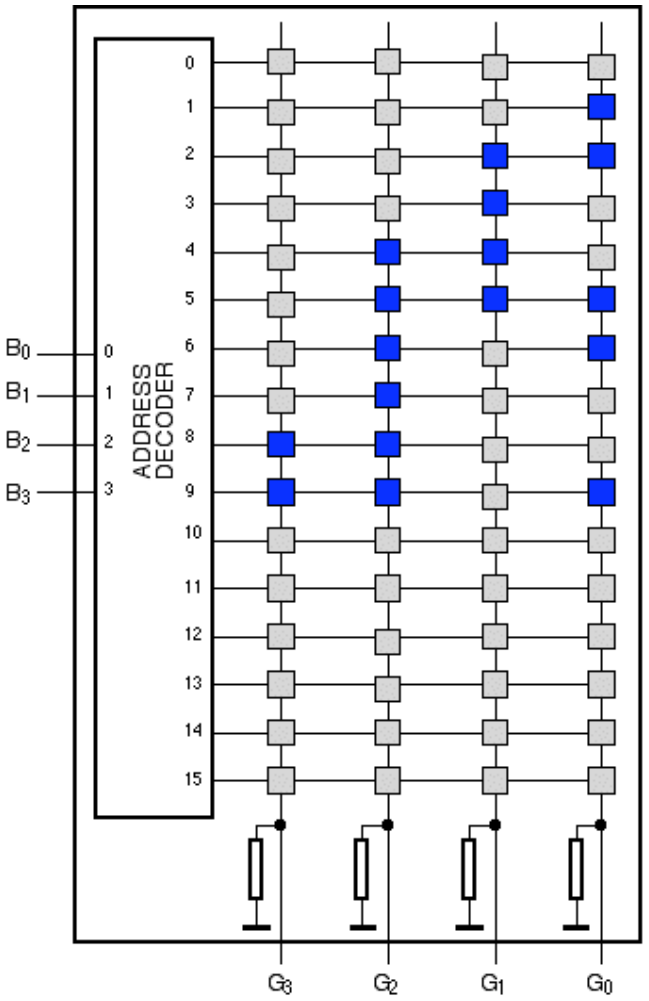
$$G_2 = \sum(4,5,6,7,8,9) + \sum_{\phi}(10,11,12,13,14,15)$$

$$G_2 = \sum(2,3,4,5) + \sum_{\phi}(10,11,12,13,14,15)$$

$$G_2 = \sum(1,2,5,6,9) + \sum_{\phi}(10,11,12,13,14,15)$$

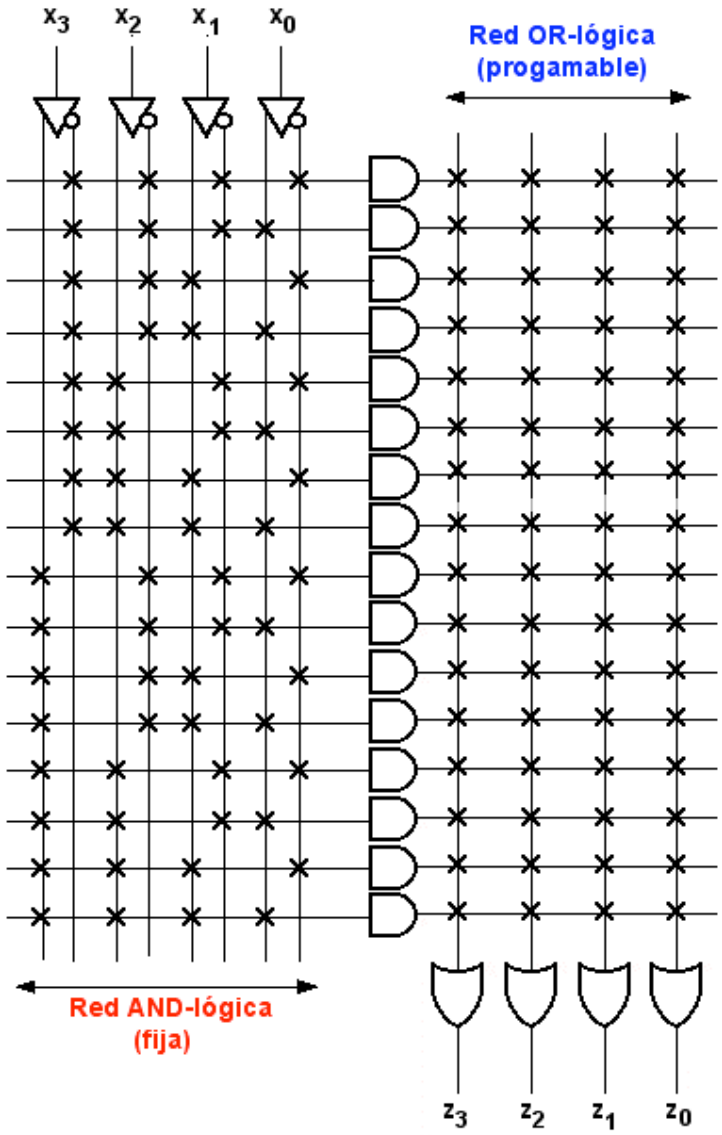


	D_3	D_2	D_1	D_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0



ROM: Estructura Conceptual

ROM 16 x 4



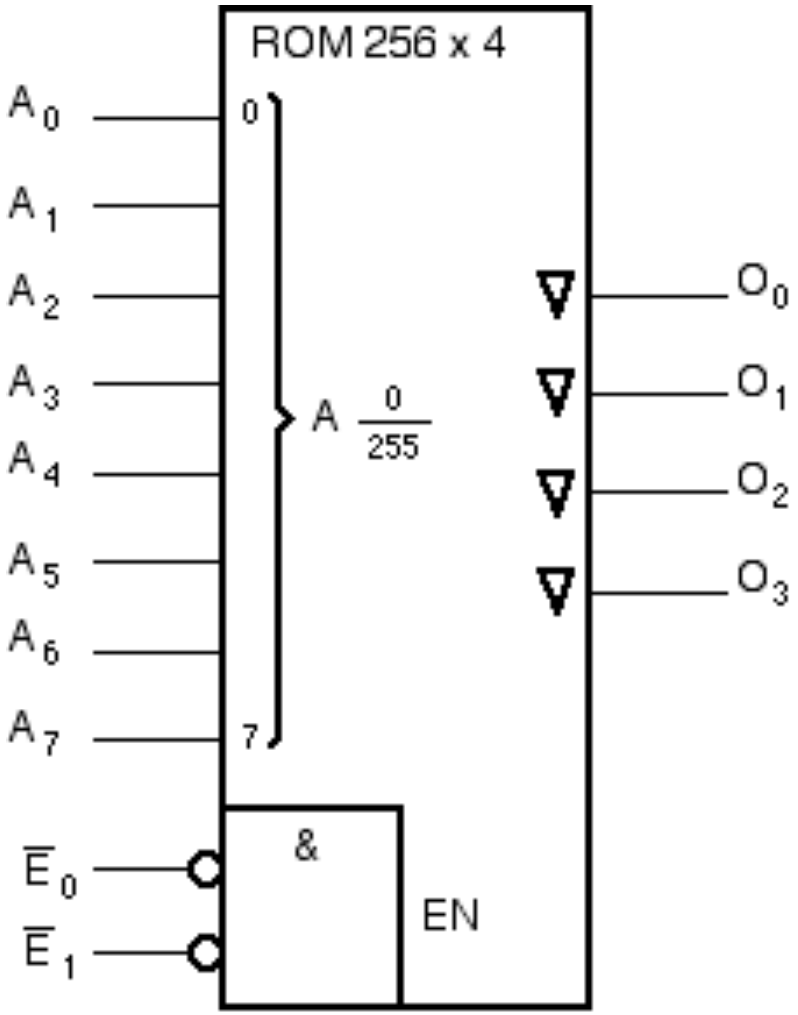
Hasta

4 funciones booleanas

de hasta

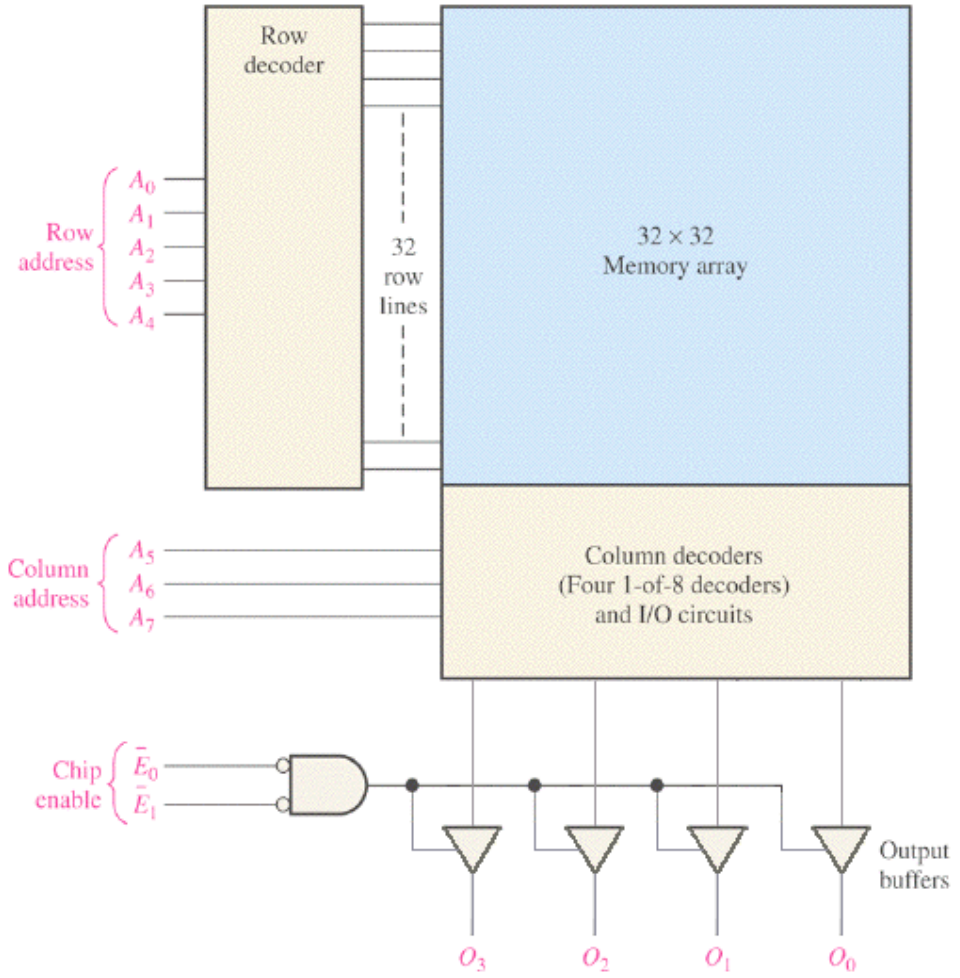
4 variables

ROM: Representación simbólica



ROM 256 x 4

ROM: Organización interna

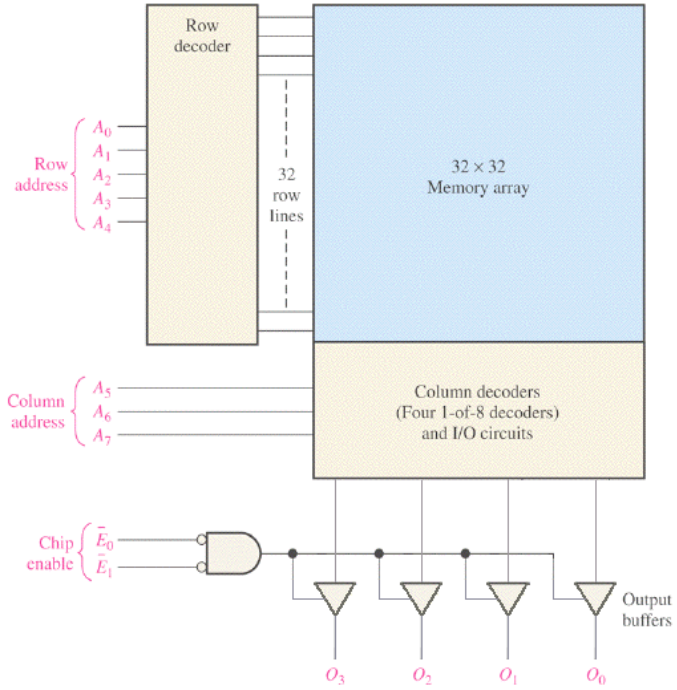


Matriz
32 x (8 x 4)

ROM 256 x 4

ROM: Organización interna (2)

Ejercicio Propuesto



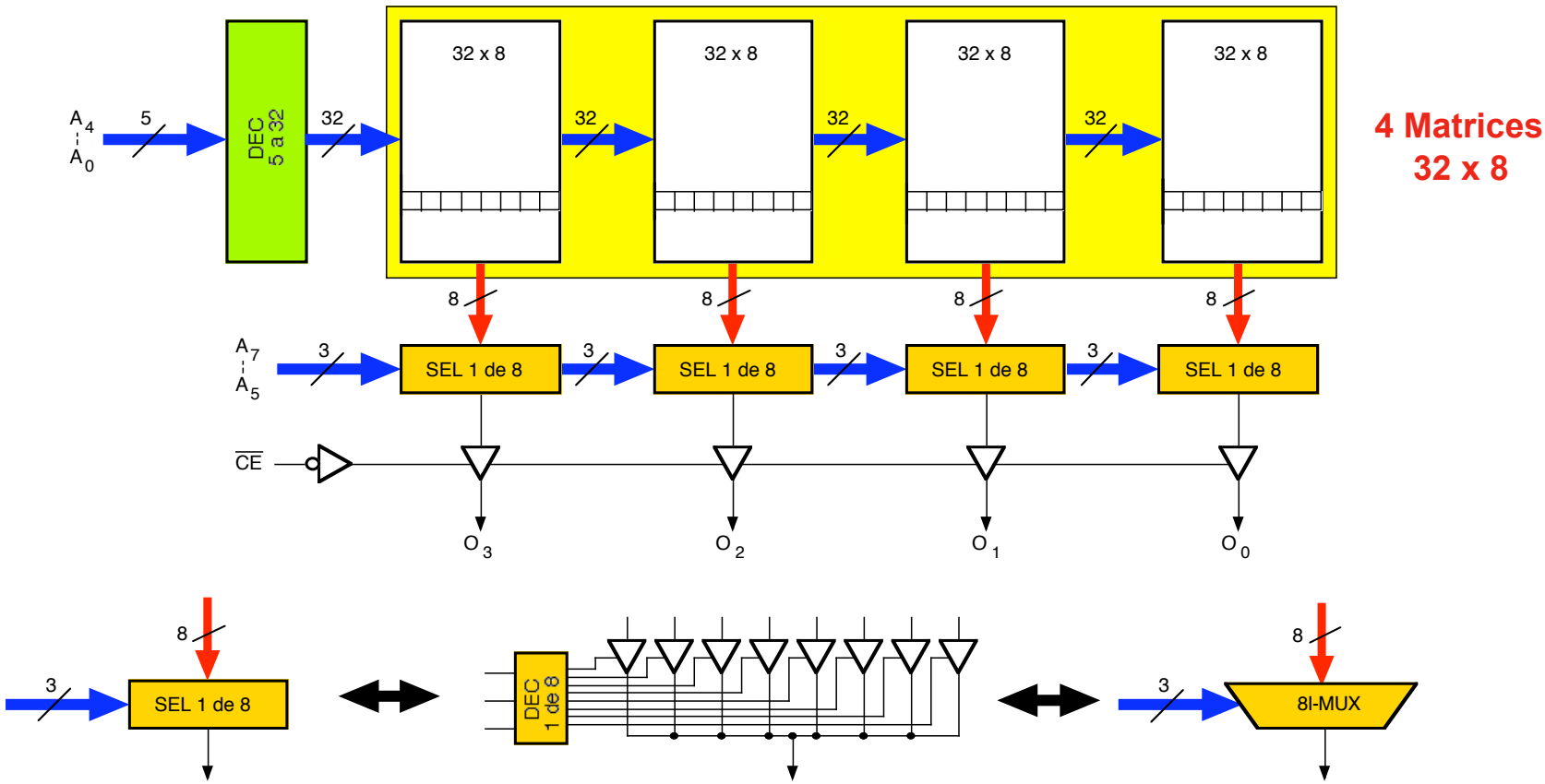
Matriz
32 x (8 x 4)

ROM 256 x 4

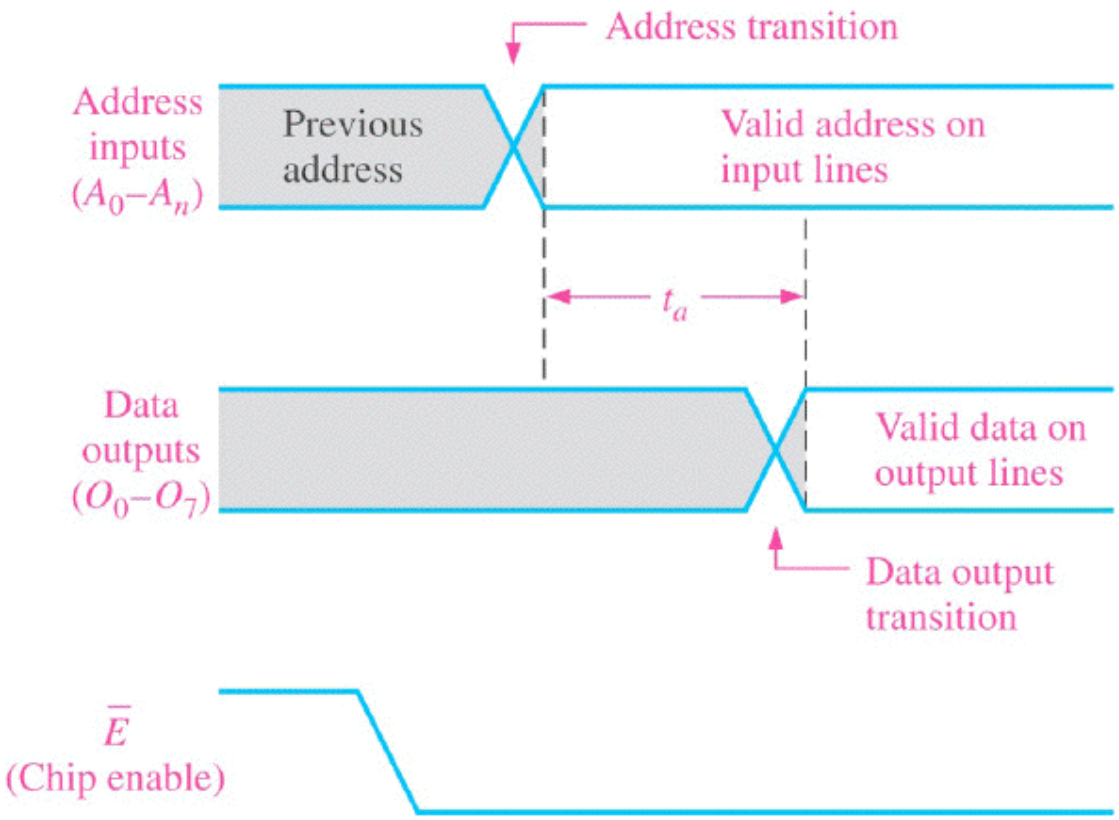
Mostrar una posible estructura interna para la memoria de la figura.

ROM: Organización interna (3)

Ejercicio Propuesto

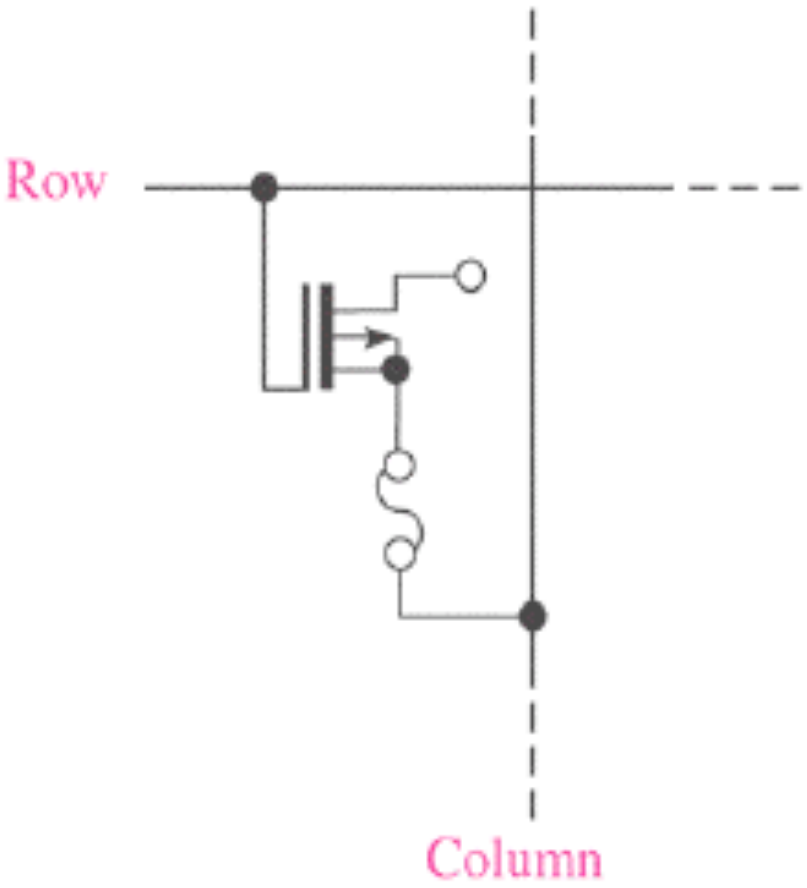


ROM: Lectura de datos



Tiempo de acceso desde dirección

PROM: Celda básica



Fusible intacto: almacena un '1' lógico
Fusible fundido: almacena un '0' lógico

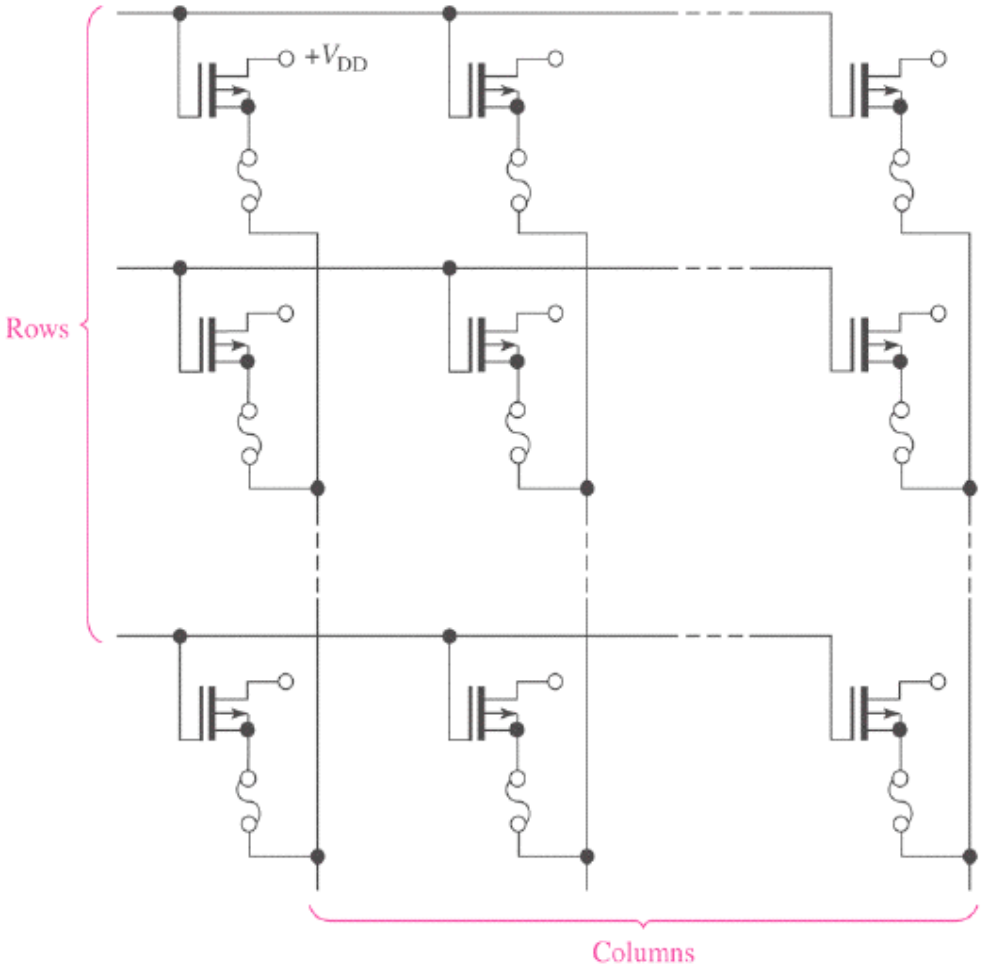
Modo de programación:
Se selecciona una celda y se provoca una fuerte inyección de corriente

Fusión de fusible metálico
Inyección de corriente elevada: Rotura por calentamiento

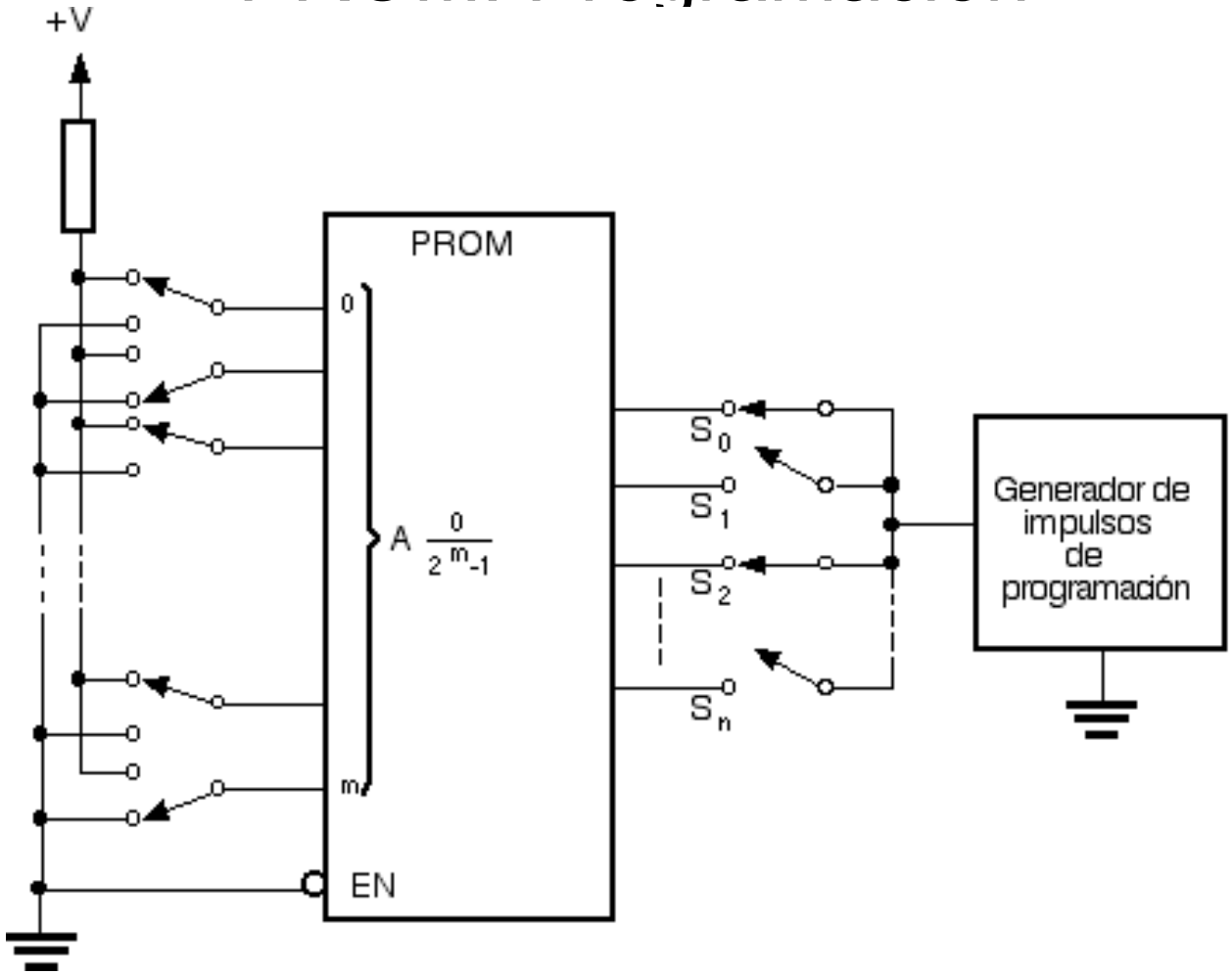
Rotura de fusible polisilicio
Inyección de corriente elevada: Rotura y oxidación por calentamiento

Migración inducida por avalancha
Diodos en oposición: Aplicando un fuerte voltaje se provoca la avalancha de uno de los diodos. La migración de iones de aluminio cortocircuita la unión (~antifusible)

PROM: Matriz interna MOS



PROM: Programación



Se programan los ‘0s’

PROM: Programación (2)

Programación automática empleando un equipo Programador de PROM

- ◊ **Identifica la PROM**
- ◊ **Genera la secuencia de señales necesaria, con la temporización requerida para la programación, según el patrón de '0s' y '1s' establecido**
- ◊ **Aplica la secuencia de programación**
- ◊ **Tras la programación de cada bit verifica el contenido de cada celda recién programada**

Existen pequeños equipos controlados por ordenador, adecuados para la producción en pequeña escala

- ◊ **Válidos para muchos tipos de PROMs, EPROMs, …**

PROM borrable

Tipos básicos: **UV EPROM**
EEPROM

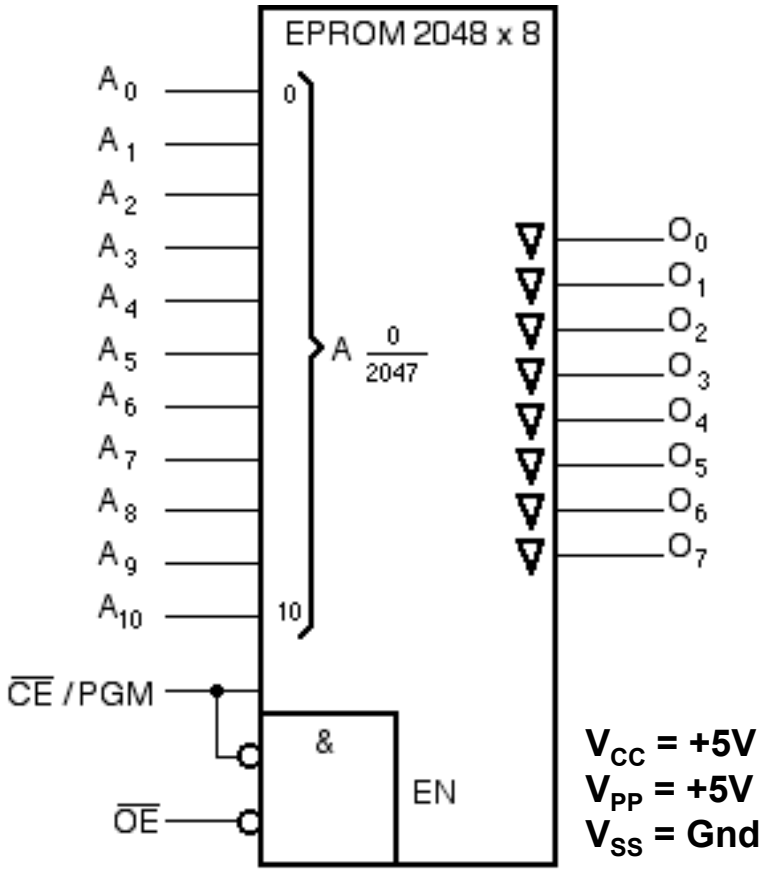
◊ **Matriz de transistores NMOS con compuerta aislada o flotante ("Gate" sin conexión eléctrica)**

◊ **La grabación consiste en cargar la compuerta aislada, de tal modo que cuando se aplique el voltaje de lectura a la "Gate" el transistor no conduzca. Se graban los '0s'**

◊ **UV EPROM: El borrado se realiza mediante exposición a radiación ultavioleta "ex situ". Disponen de un ventana de cuarzo "ad hoc". OT PROM: Usualme una UV EPROM con la capacidad de borrado inhibida**

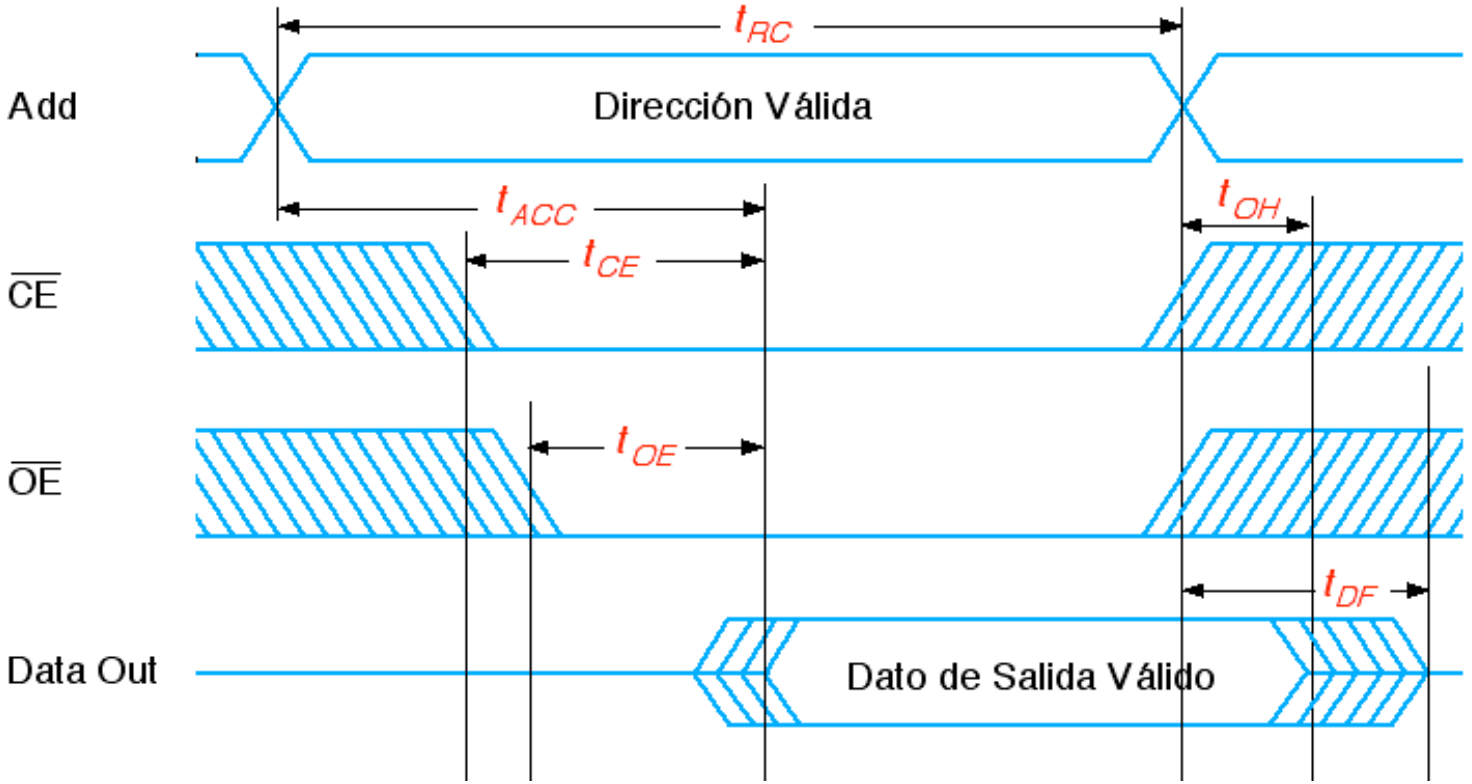
◊ **EEPROM o E^2PROM: El borrado se realiza eléctricamente "in situ". Requieren circuitería específica para borrado y programación. Celda más compleja que la de la UV EPROM (baja densidad)**

EPROM: Representación simbólica



- **CE*/PGM: Habilitación del chip y programación. Sólo UV PROM (WE* en EEPROM)**
- **V_{PP}: Patilla para aplicación de la tensión de programación**

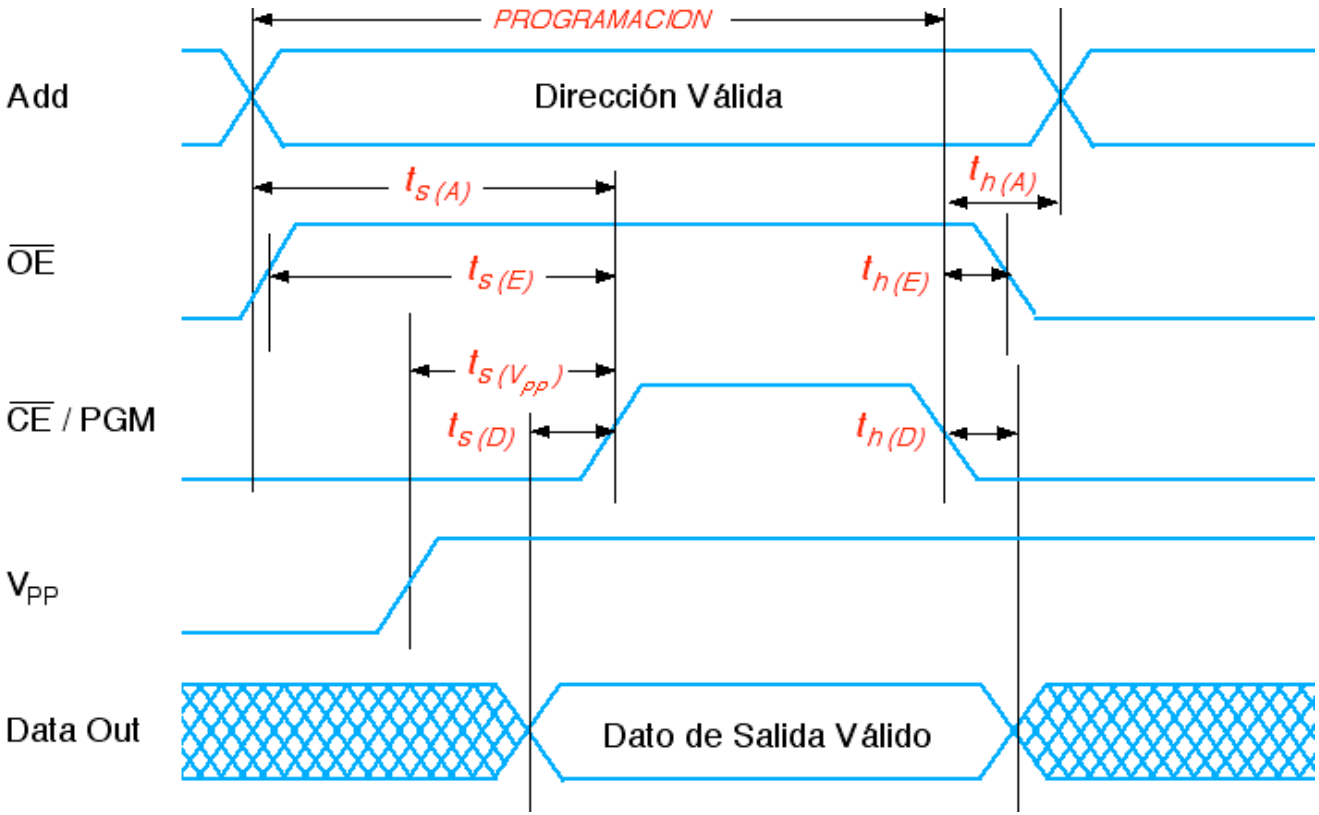
EPROM: Ciclo de lectura



AMD 27C256

- **UV PROM y EEPROM**

EPROM: Ciclo de programación



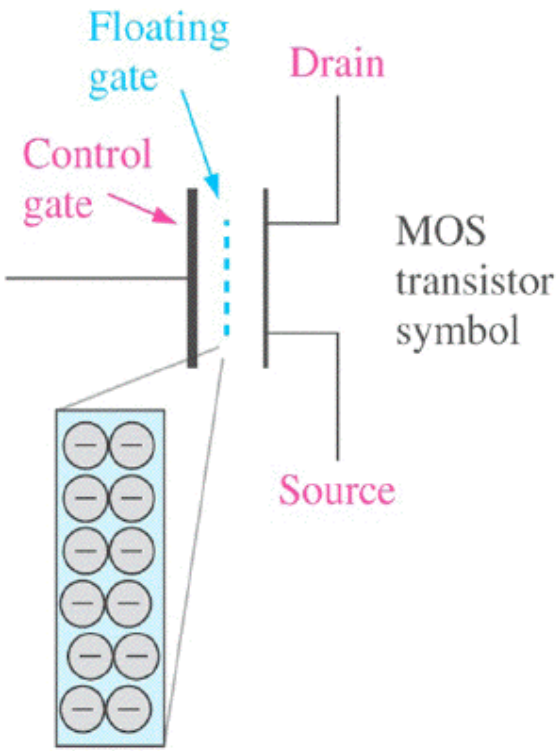
- **UV PROM**

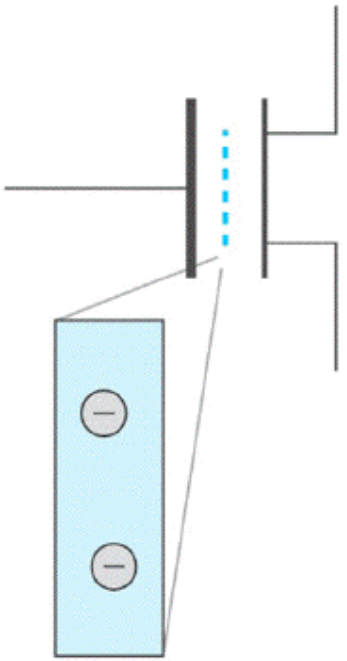
Anchura pulso PGM: 10-60 ms
V_{PP}: 12,5 V (UV EPROM Algoritmos rápidos)
El ciclo de programación de una EEPROM es similar al de una SRAM, y no requiere voltajes superiores al de alimentación

Memoria FLASH

- **Memoria de lectura / escritura**
- **No volátil**
- **Matriz de transistores NMOS con compuerta aislada o flotante**
- **Cada celda de memoria consiste en un único transistor (alta densidad)**
- **La grabación consiste en cargar la compuerta aislada, de tal modo que cuando se aplique un voltaje de lectura a la "Gate", el transistor no conduzca. Se graban los '0s'**
- **El borrado se realiza eléctricamente "in situ"**
- **Se utilizan en sustitución de unidades de disco duro de baja-media capacidad y en electrónica de consumo portátil**

FLASH: Celda de almacenamiento

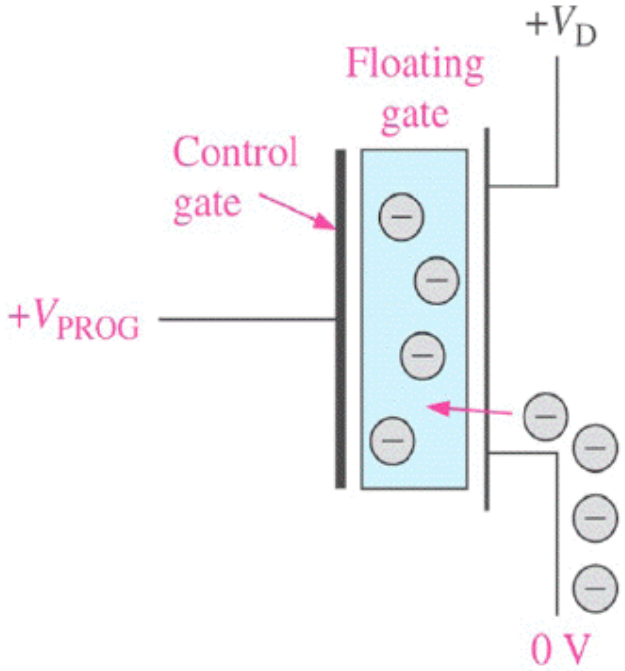




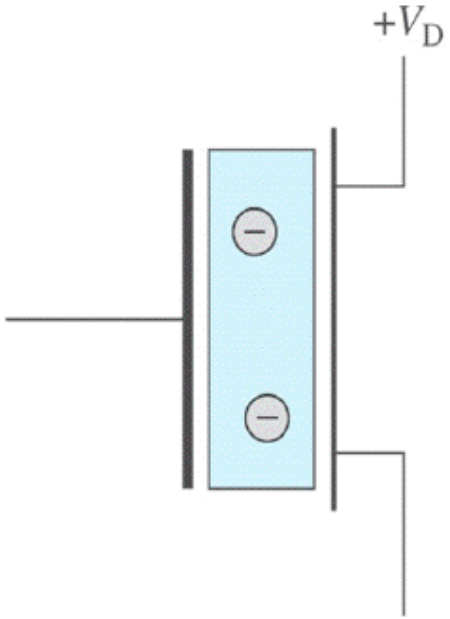
Almacena un ‘0’

Almacena un ‘1’

FLASH: Almacenamiento (escritura)

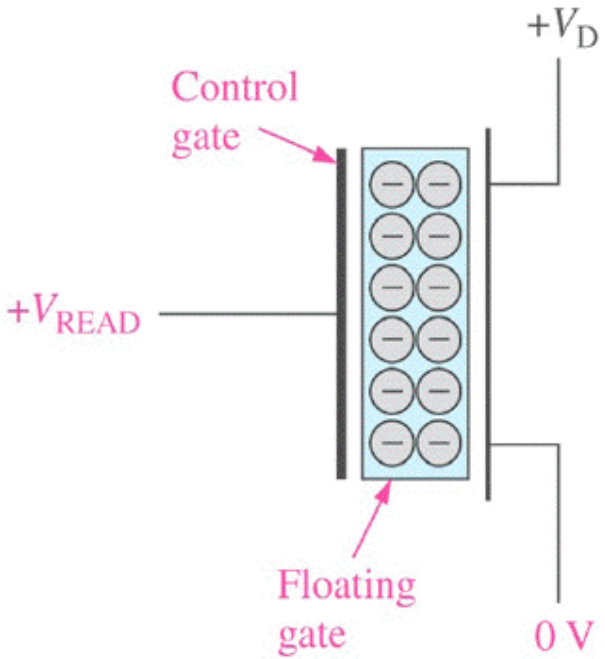


Almacenamiento de un ‘0’ (programación)

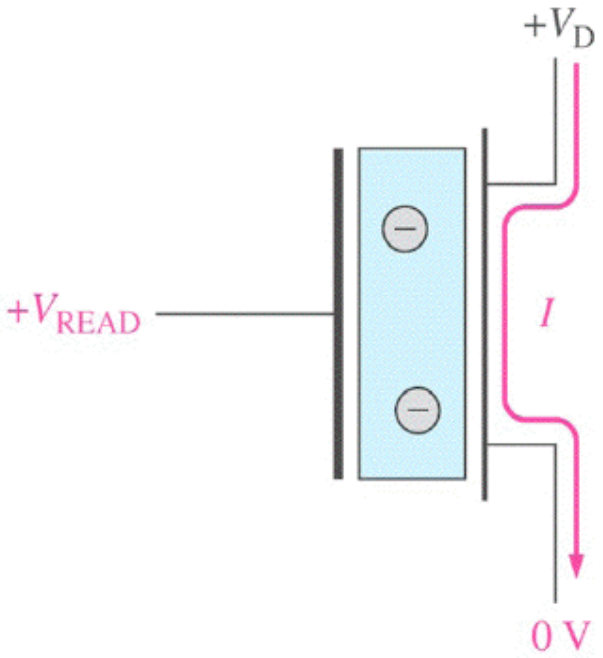


Almacenamiento de un ‘1’ (estado de borrado)

FLASH: Lectura



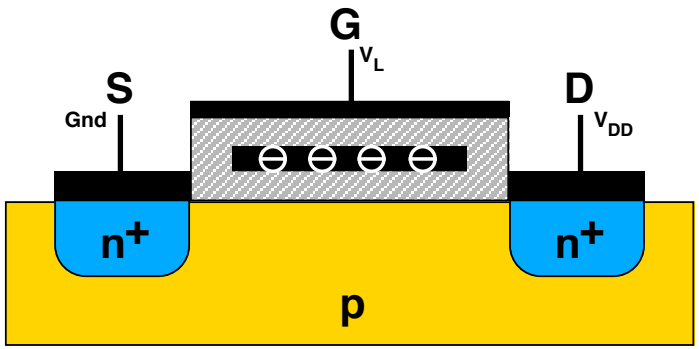
Lectura de un ‘0’

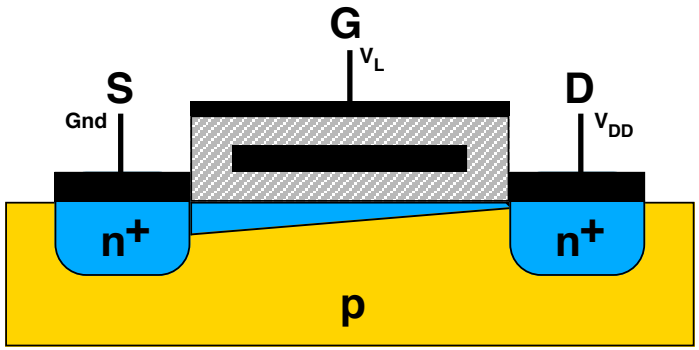


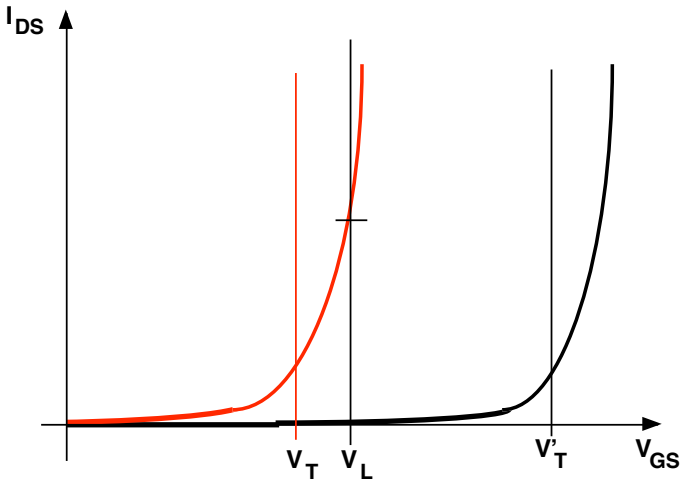
Lectura de un ‘1’

FLASH: Tensión umbral

Almacena un ‘0’

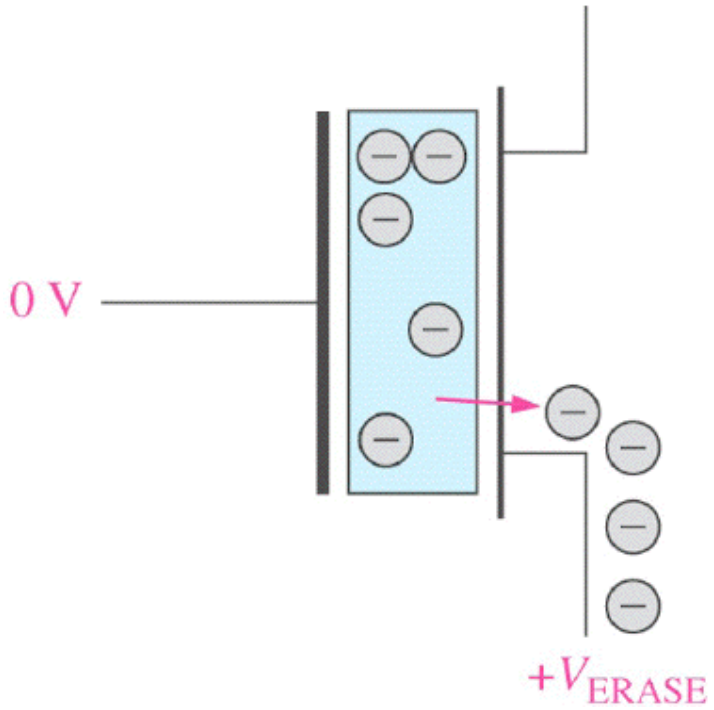






Almacena un ‘1’

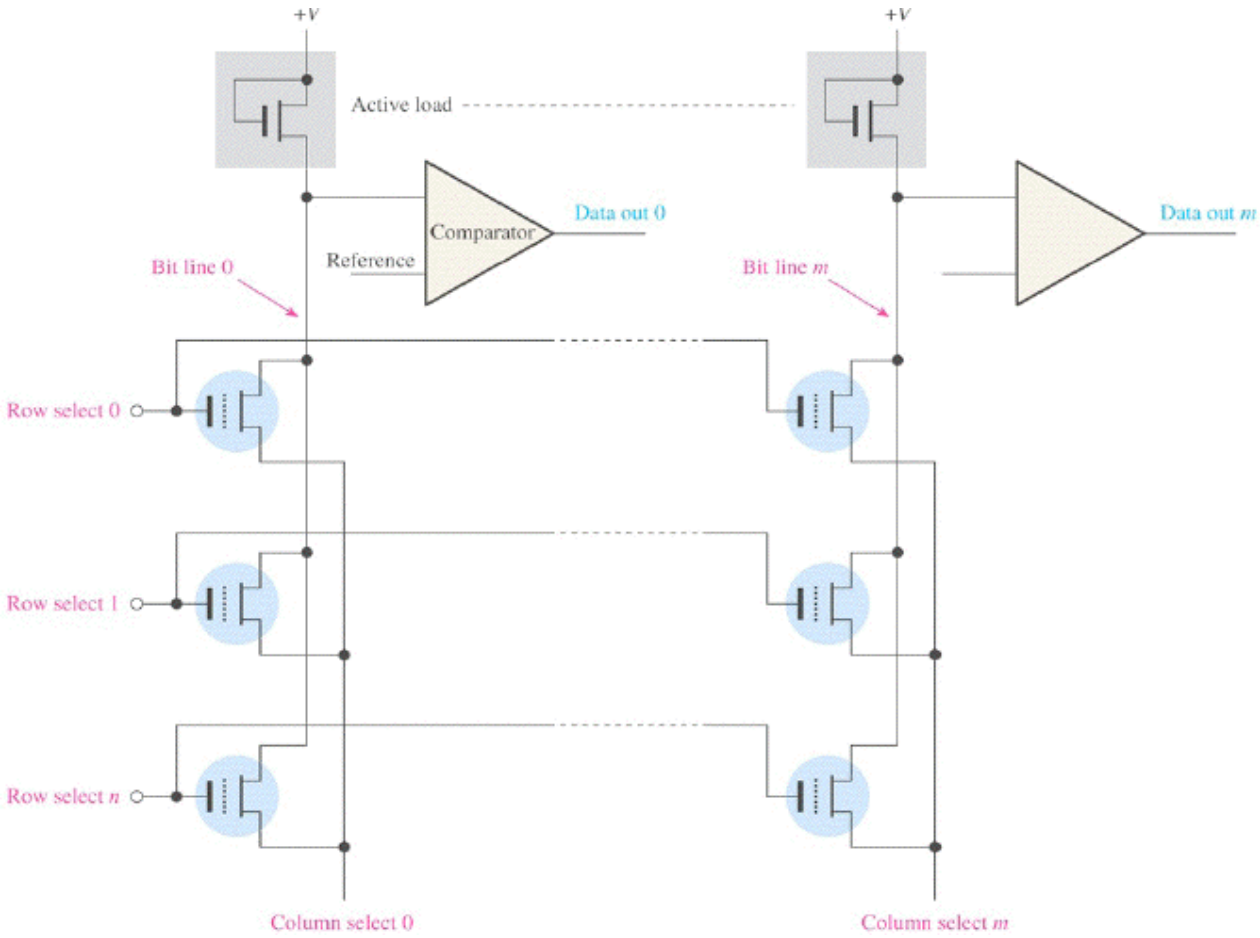
FLASH: Borrado



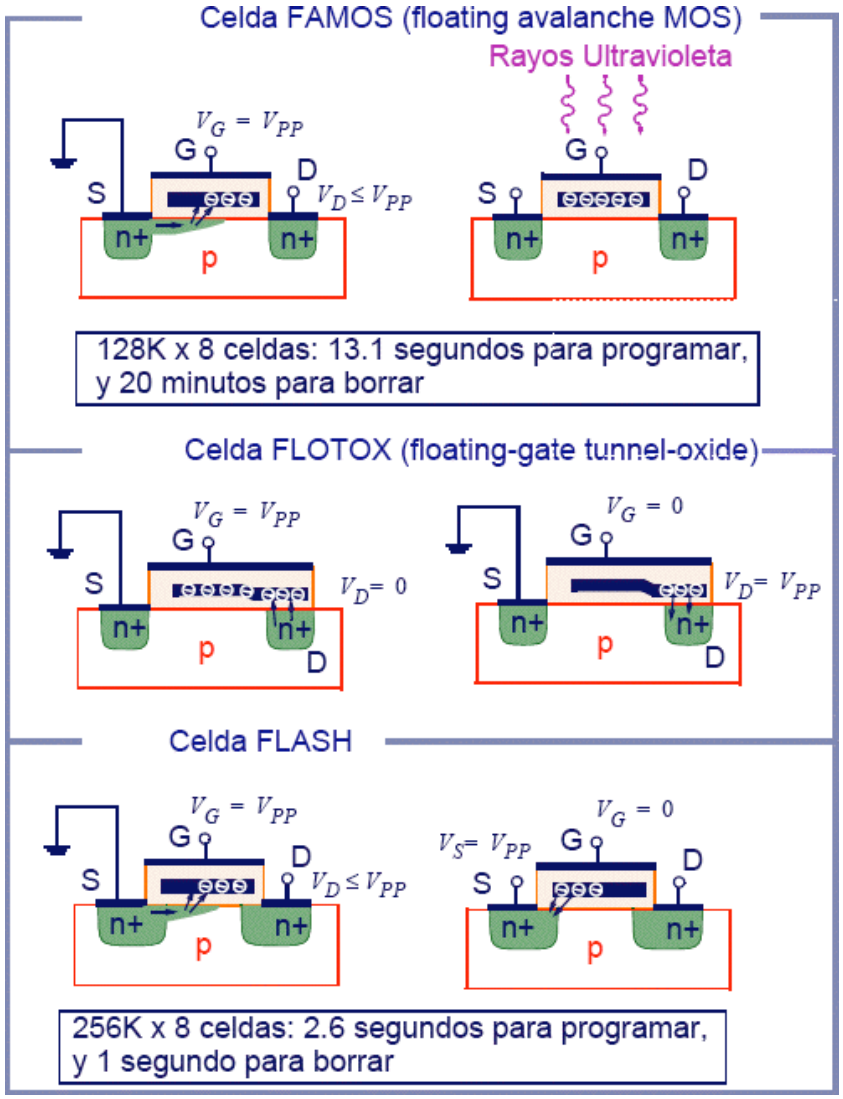
Borrado de un ‘0’

Se almacena un ‘1’

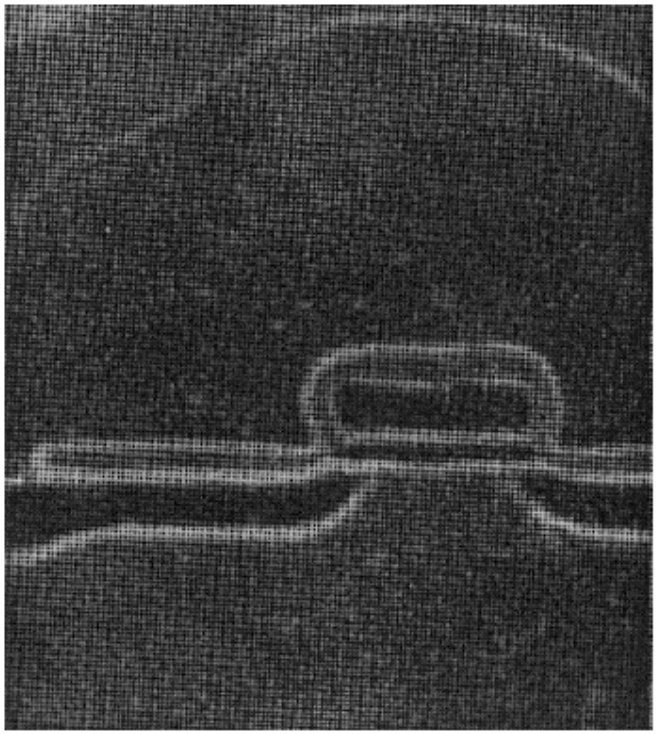
FLASH: Matriz básica



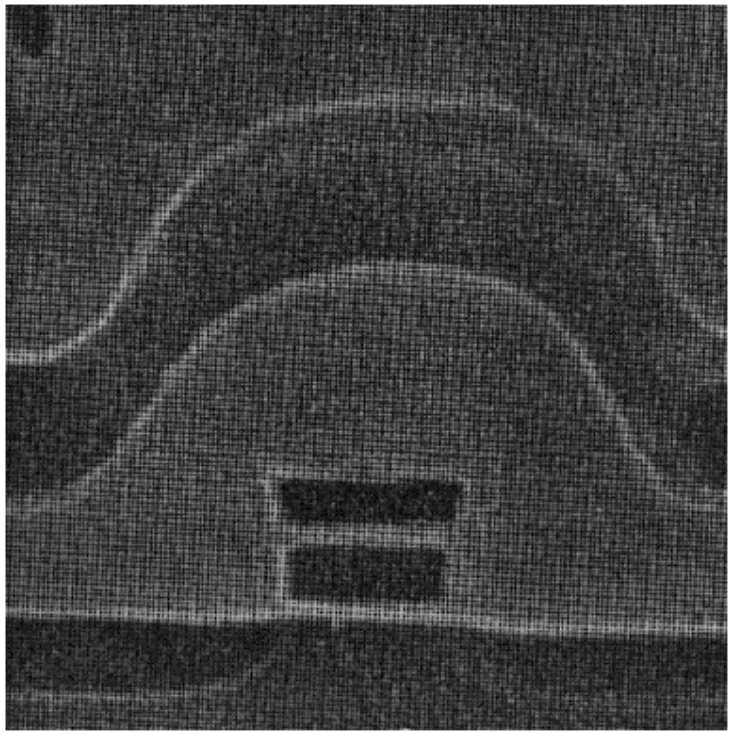
FLASH: Comparativa de celdas EPROM



Comparativa de celdas EPROM



FLASH

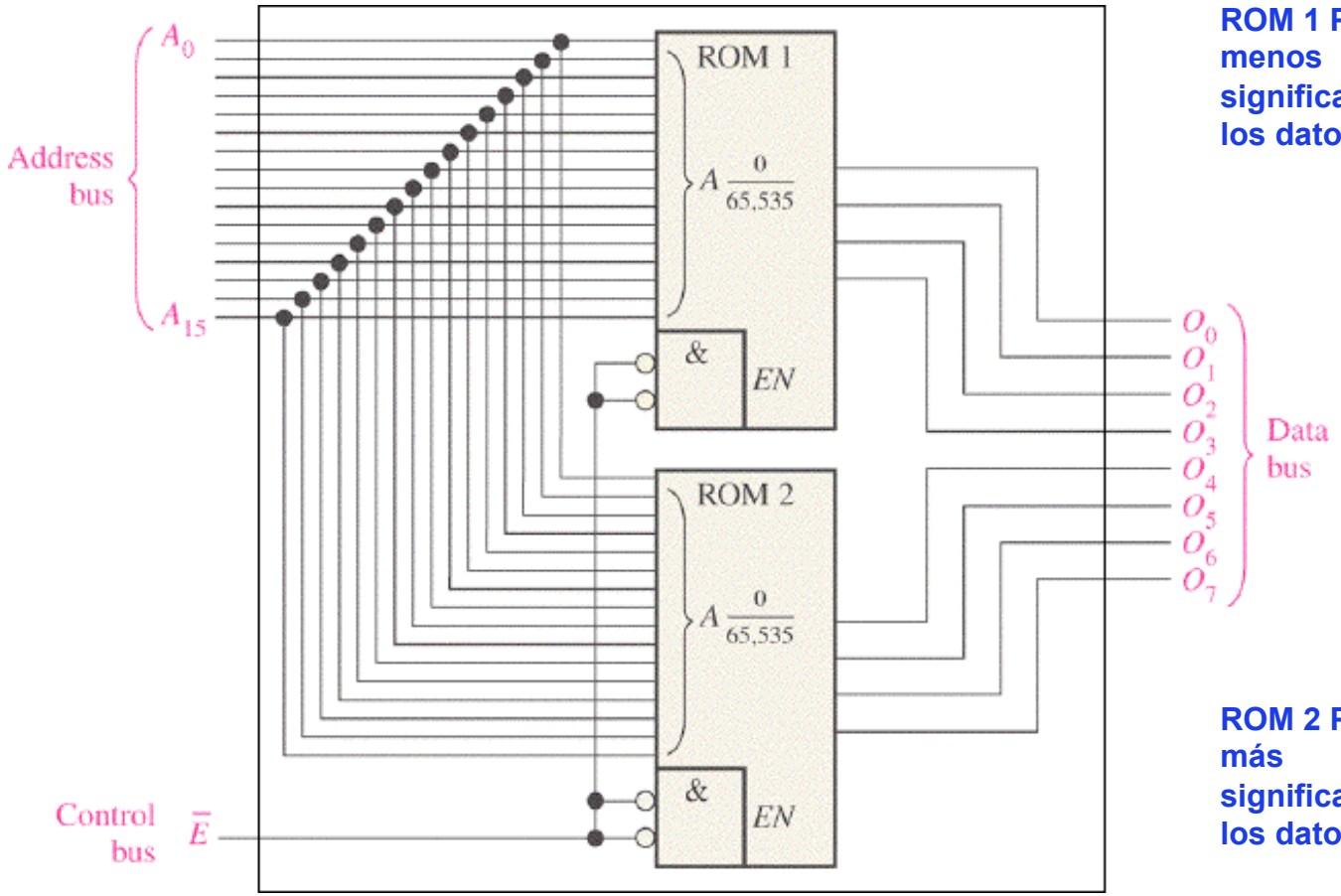


EPROM

Intel

Expansión de memorias

Extensión de la longitud de palabra



ROM 1 Parte menos significativa de los datos

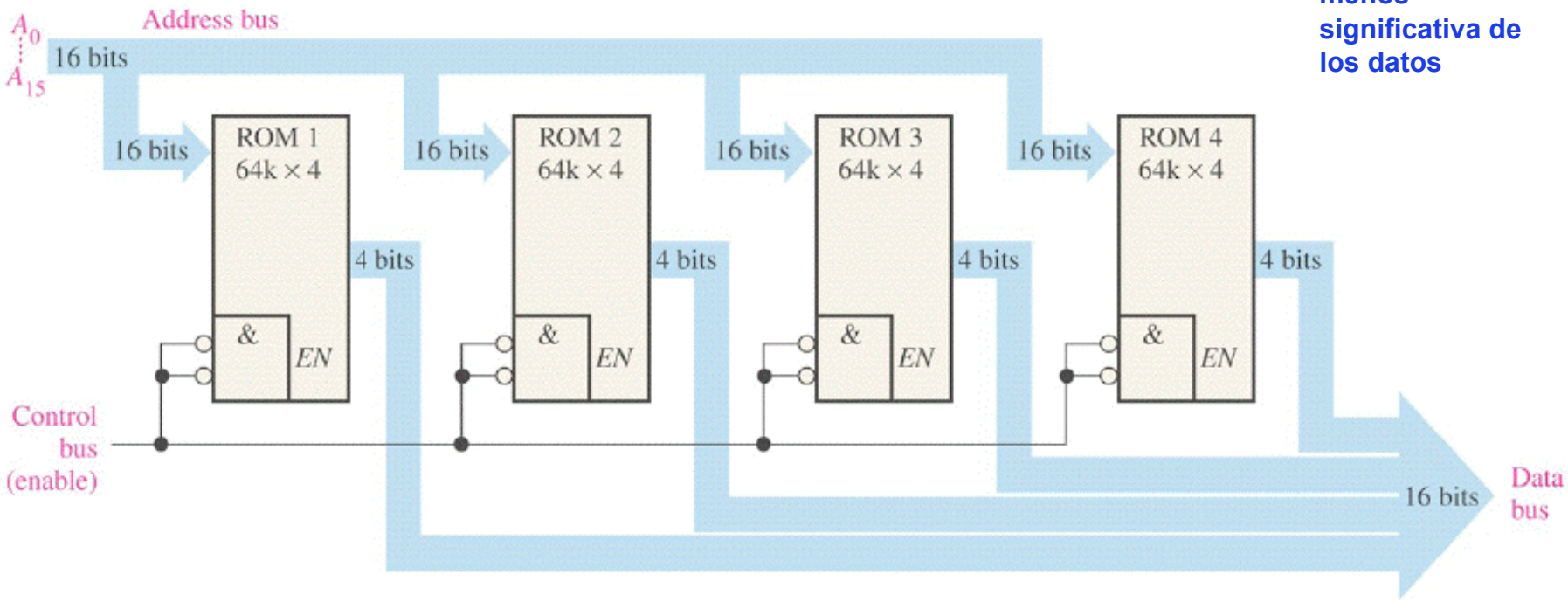
ROM 2 Parte más significativa de los datos

2 x (64K x 4) => 1 x (64K x 8)

Expansión de memorias (2)

Extensión de la longitud de palabra

ROM 1 Parte menos significativa de los datos

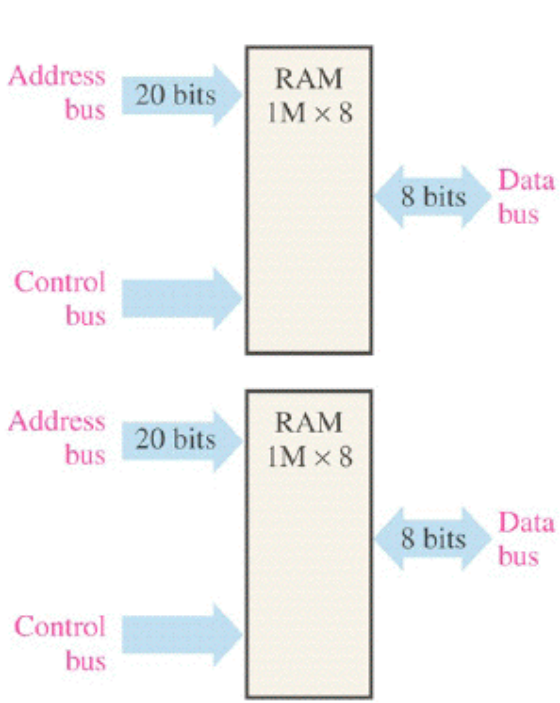


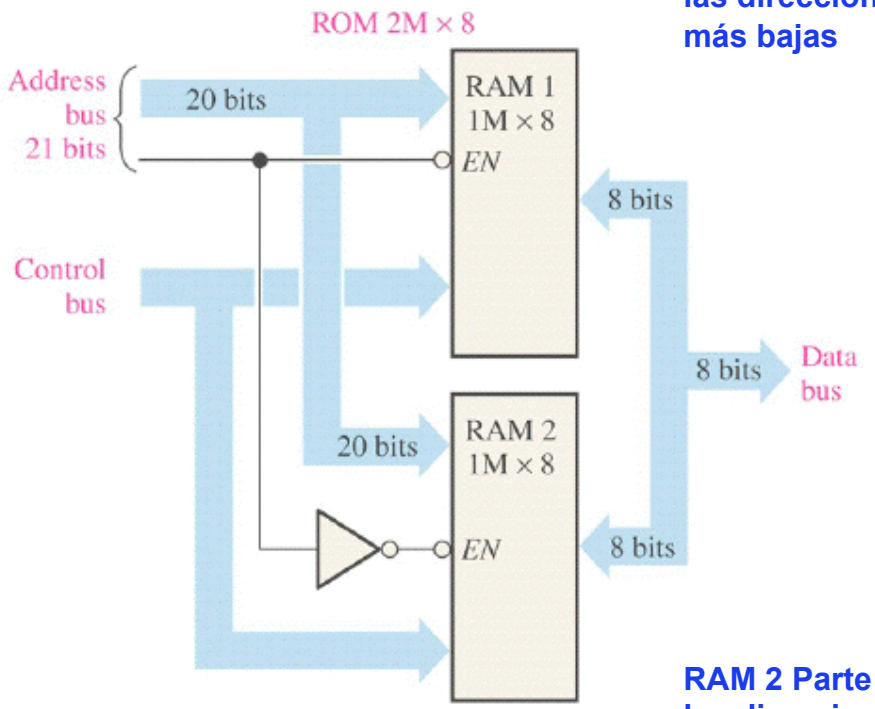
4 x (64K x 4) => 1 x (64K x 16)

ROM 4 Parte más significativa de los datos

Expansión de memorias (3)

Extensión de la capacidad de palabras





RAM 1 Parte de las direcciones más bajas

RAM 2 Parte de las direcciones más altas

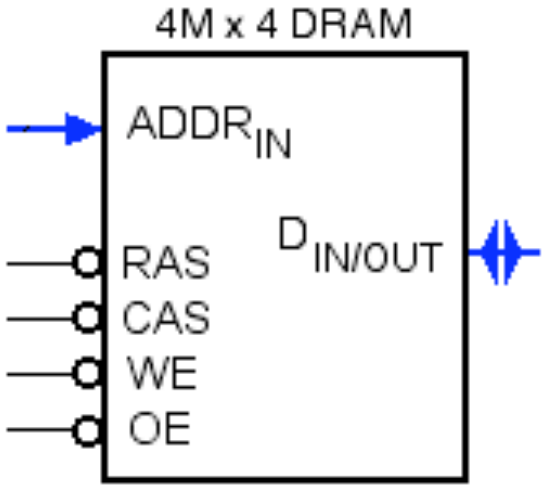
2 x (1M x 8) => 1 x (2M x 8)

Si entradas y salidas son separadas: interconectar los buses de entrada
interconectar los buses de salida

Expansión de memorias (4)

Extensión de la longitud de palabra y de la capacidad de palabras

Ejercicio



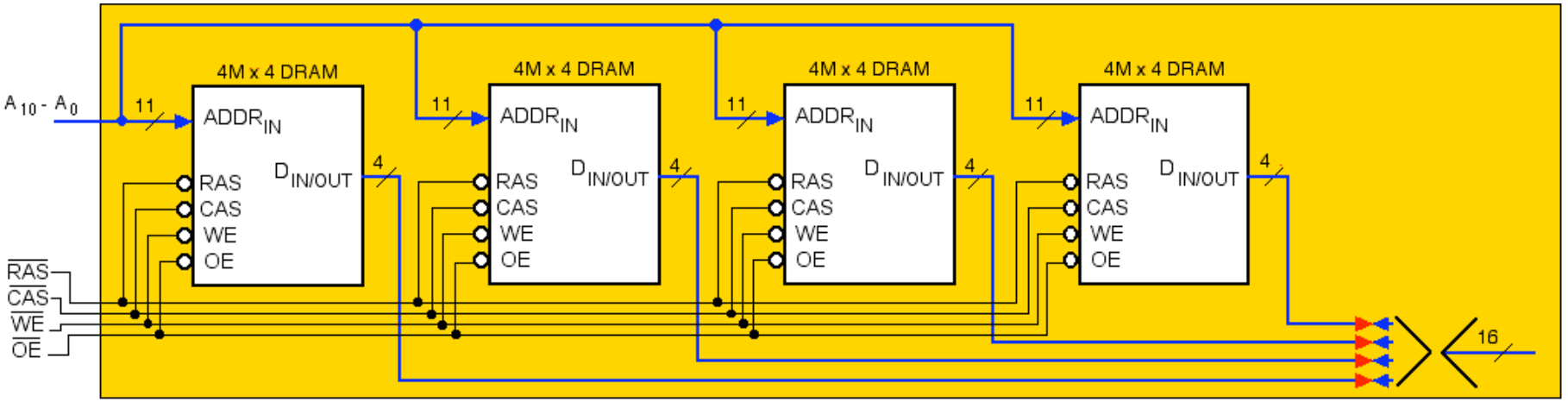
hynix **GM71C(S)17403C/CL**

Se dispone de módulos DRAM de 4M x 4 bits cuyo símbolo recoge la figura. Mostrar una posible implementación de una memoria de 4M palabras de 16 bits, con las capacidades de lectura, escritura y refresco "Sólo-RAS*" que permite el módulo de memoria original.

Expansión de memorias (5)

Extensión de la longitud de palabra y de la capacidad de palabras

Ejercicio



Expansión de memorias (6)

Extensión de la longitud de palabra y de la capacidad de palabras

Ejercicio propuesto

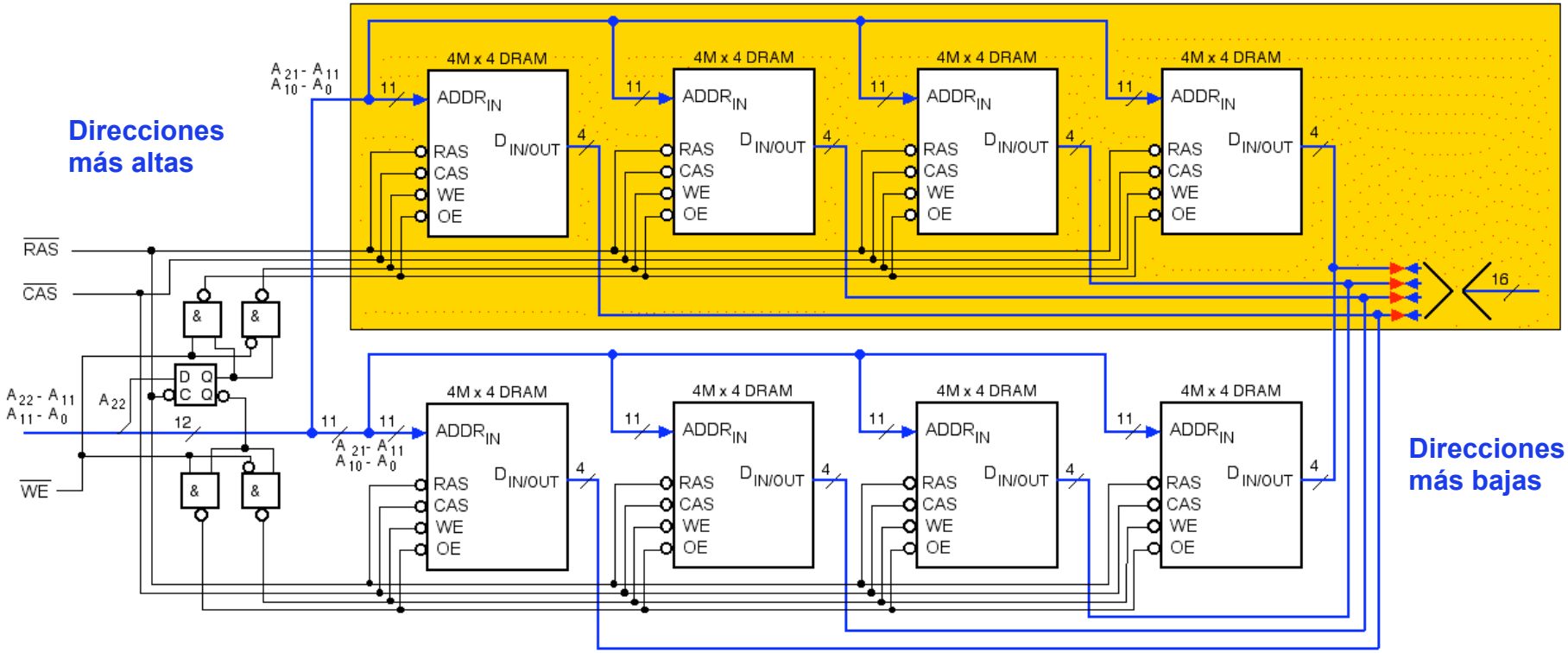
Utilizando los módulos DRAM de 4M x 4 bits anteriores, indicar y analizar una posible implementación de una memoria de 8M palabras de 16 bits, con las capacidades de lectura, escritura y refresco "Sólo-RAS*"que permite el módulo de memoria original. Utilícense si resulta necesario dispositivos SSI/MSI adicionales.

Expansión de memorias (7)

Extensión de la longitud de palabra y de la capacidad de palabras

		MSB		LSB		
A22	WE	OE	WE	OE	WE	
0	0	1	1	1	0	Escribe en LSB
0	1	1	1	0	1	Lee de LSB
1	0	1	0	1	1	Escribe en MSB
1	1	0	1	1	1	Lee de MSB

Ejercicio propuesto



El bit A_{22} (más significativo de la dirección de filas) se utiliza para deshabilitar las salidas y la operación de escritura del conjunto superior (si es '0') o inferior (si es '1') de módulos

Memoria FIFO

"First In-First Out"

Modo de operación

Conventional shift register

Input	X	X	X	X	Output
0	0	X	X	X	→
1	1	0	X	X	→
1	1	1	0	X	→
0	0	1	1	0	→

FIFO shift register

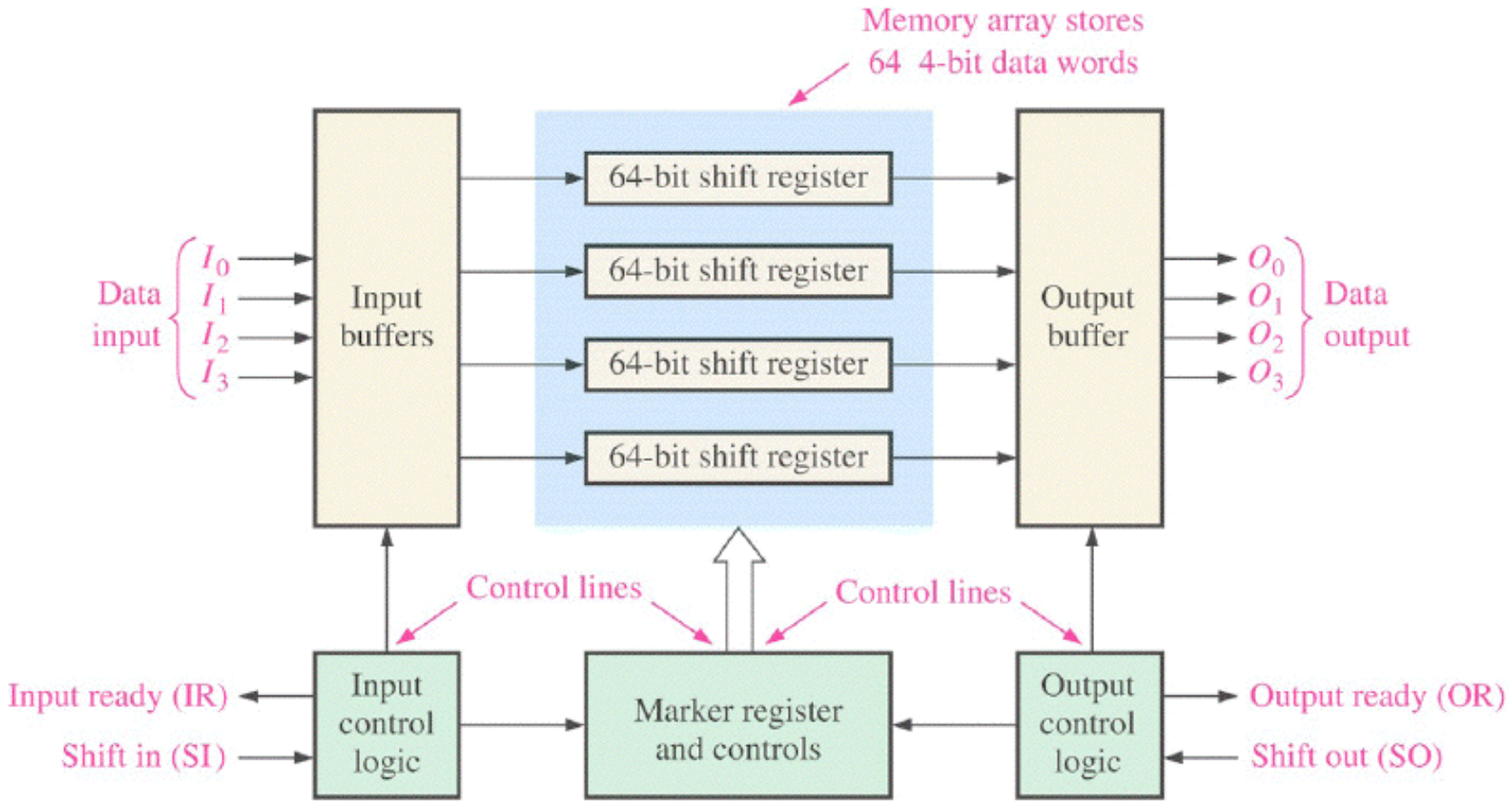
Input	—	—	—	—	Output
0	—	—	—	0	→
1	—	—	1	0	→
1	—	1	1	0	→
0	0	1	1	0	→

El bit se almacena en la posición más próxima a la entrada de dato, desplazando a los bits previamente almacenados

El bit se almacena en la posición vacía más alejada de la entrada de dato. Los datos "caen" hasta la posición indicada por un registro de marca

Memoria FIFO (2)

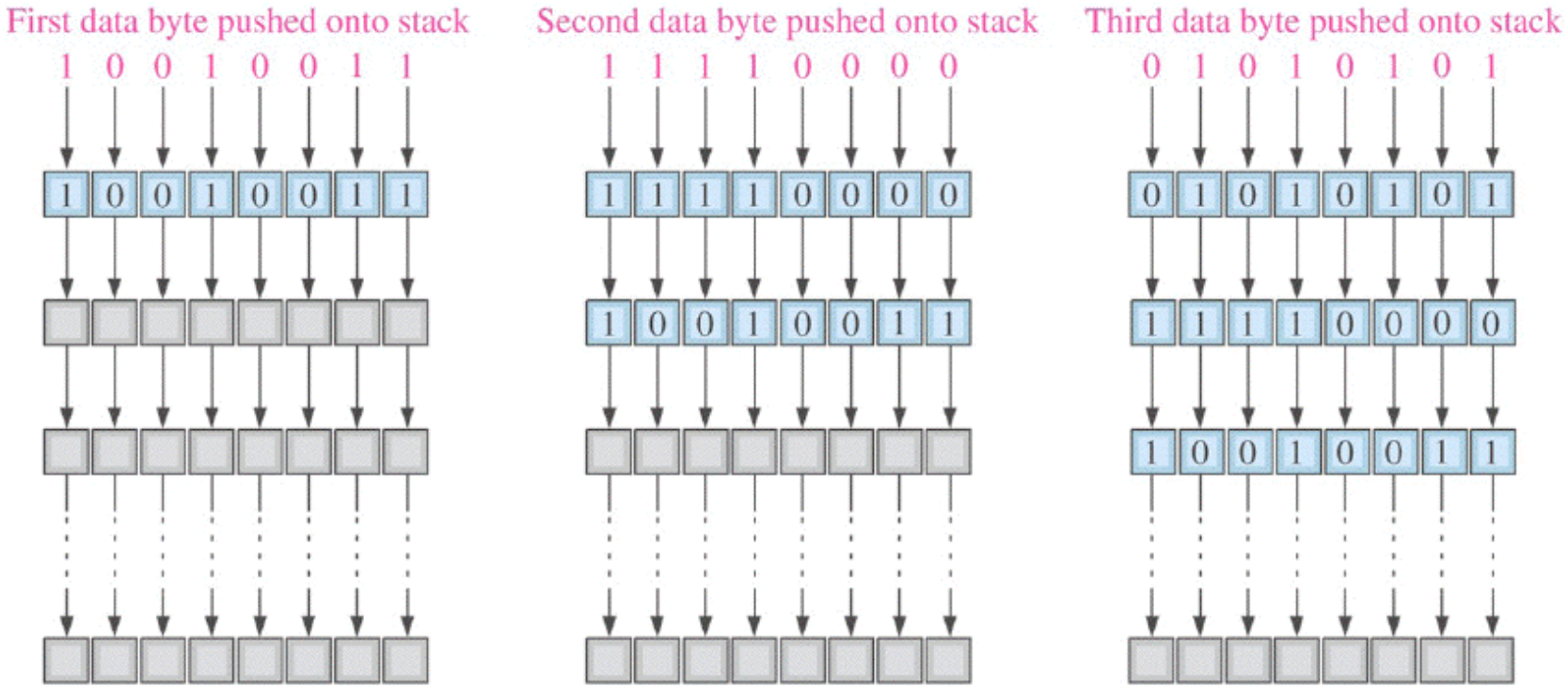
"First In-First Out"



Memoria LIFO

"Last In-First Out"

Almacenamiento de datos (escritura)

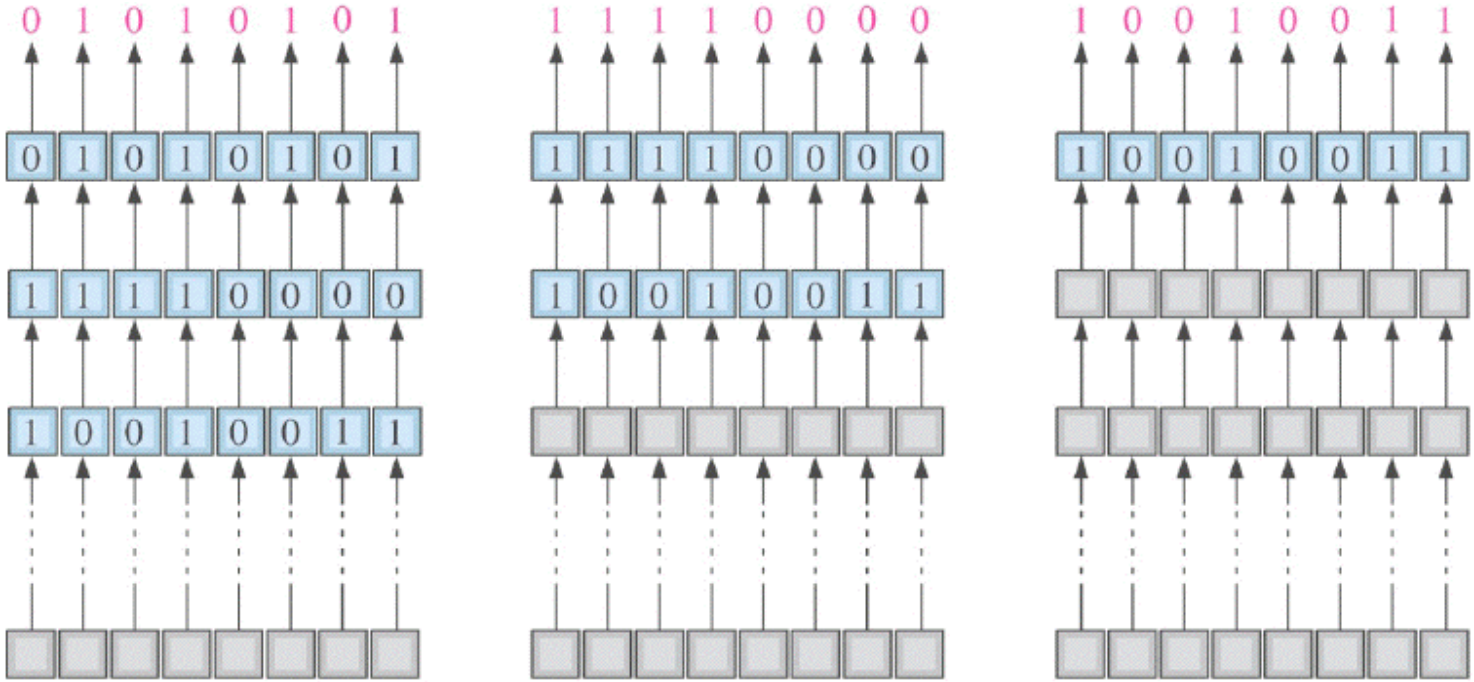


Pila de Registros

Memoria LIFO (2)

"Last In-First Out"

Extracción de datos (lectura)



- **Registros de desplazamiento bidireccionales en paralelo**

Test de Memorias

Verificación de memorias ROM

◊ **Comparación con una memoria de referencia ("Gold unit")**
◊ **Método de la suma de comprobación**

Comparación con una memoria de referencia

Para cada dirección de la memoria:
- **Leer la palabra de la ROM a comprobar y la correspondiente de la ROM de referencia**
- **Comparar ambas palabras**

Inconvenientes:

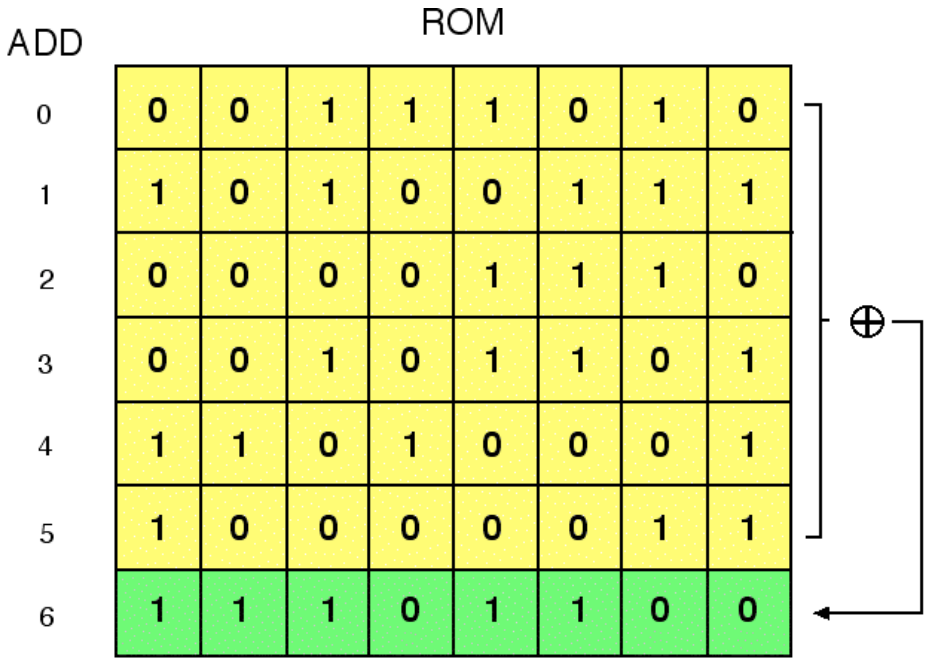
- **Requiere una memoria grabada de referencia idéntica a la que se quiere comprobar**
  - **Para cada memoria del mismo tipo pero con contenido distinto**
  - **Para cada memoria de distinto tipo**
- **Puede fallar la ROM de prueba**
- **No se adecua a la comprobación automática dentro de un sistema digital**

Método de la suma de comprobación

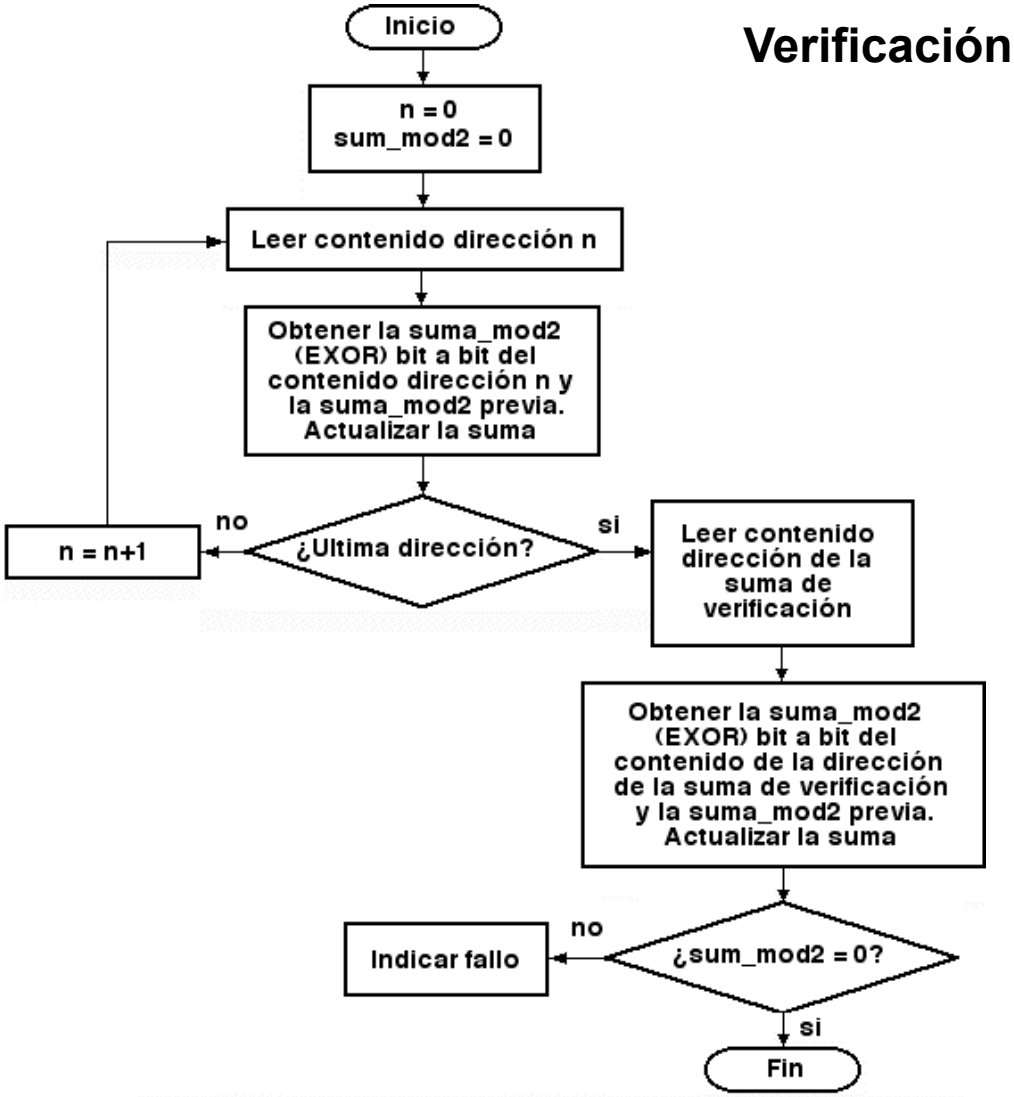
Verificación de memorias ROM

Para cada columna (bits correspondientes de cada palabra):

- **Se calcula la suma módulo 2 de todos los bits de cada columna**
- **En una dirección específica, se graba o programa en cada columna el resultado de la citada suma. Se obtiene una palabra de comprobación**



Método de la suma de comprobación



Verificación de memorias ROM

- La probabilidad de que ocurran múltiples errores que impidan la detección por enmascaramiento es muy baja
- Se adecúa a la comprobación automática dentro de un sistema digital

Rutina de comprobación en el software del sistema (sistemas basados en microprocesador); la rutina se ejecuta automáticamente durante el arranque

Verificación de memorias RAM

Verificación básica

- **Escritura de ‘0s’ en todas las celdas**
  Lectura de comprobación
  Escritura de ‘1s’ en todas las celdas
  Lectura de comprobación

Problema: Algunos fallos no se detectan

- **Cortocircuito entre celdas adyacentes**
- **Ruido interno: El contenido de algunas direcciones se modifica debido al cambio de otras direcciones**

Método de prueba con patrón ajedrezado

- **Cargar patrón de ‘1s’ y ‘0s’ alternados**
  Lectura de comprobación
  Se invierte el patrón de ‘1s’ y ‘0s’
  Lectura de comprobación
- **Para cada dirección: Se alterna el patrón de una dirección de memoria**
  Lectura de comprobación del contenido de las restantes direcciones

Método de prueba con patrón ajedrezado

Verificación de memorias RAM

ADD	RAM							
0	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	0
2	0	1	0	1	0	1	0	1
3	1	0	1	0	1	0	1	0
4	0	1	0	1	0	1	0	1
5	1	0	1	0	1	0	1	0
6	0	1	0	1	0	1	0	1
7	1	0	1	0	1	0	1	0
8	0	1	0	1	0	1	0	1
9	1	0	1	0	1	0	1	0
10	0	1	0	1	0	1	0	1
11	1	0	1	0	1	0	1	0

ADD	RAM							
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1
2	1	0	1	0	1	0	1	0
3	0	1	0	1	0	1	0	1
4	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1

- **Detecta cortocircuitos entre celdas adyacentes**
- **Prueba la capacidad de cada celda para almacenar ‘0s’ y ‘1s’**

Método de prueba con patrón ajedrezado (2)

Verificación de memorias RAM

ADD	RAM							
0	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1
2	1	0	1	0	1	0	1	0
3	0	1	0	1	0	1	0	1
4	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1

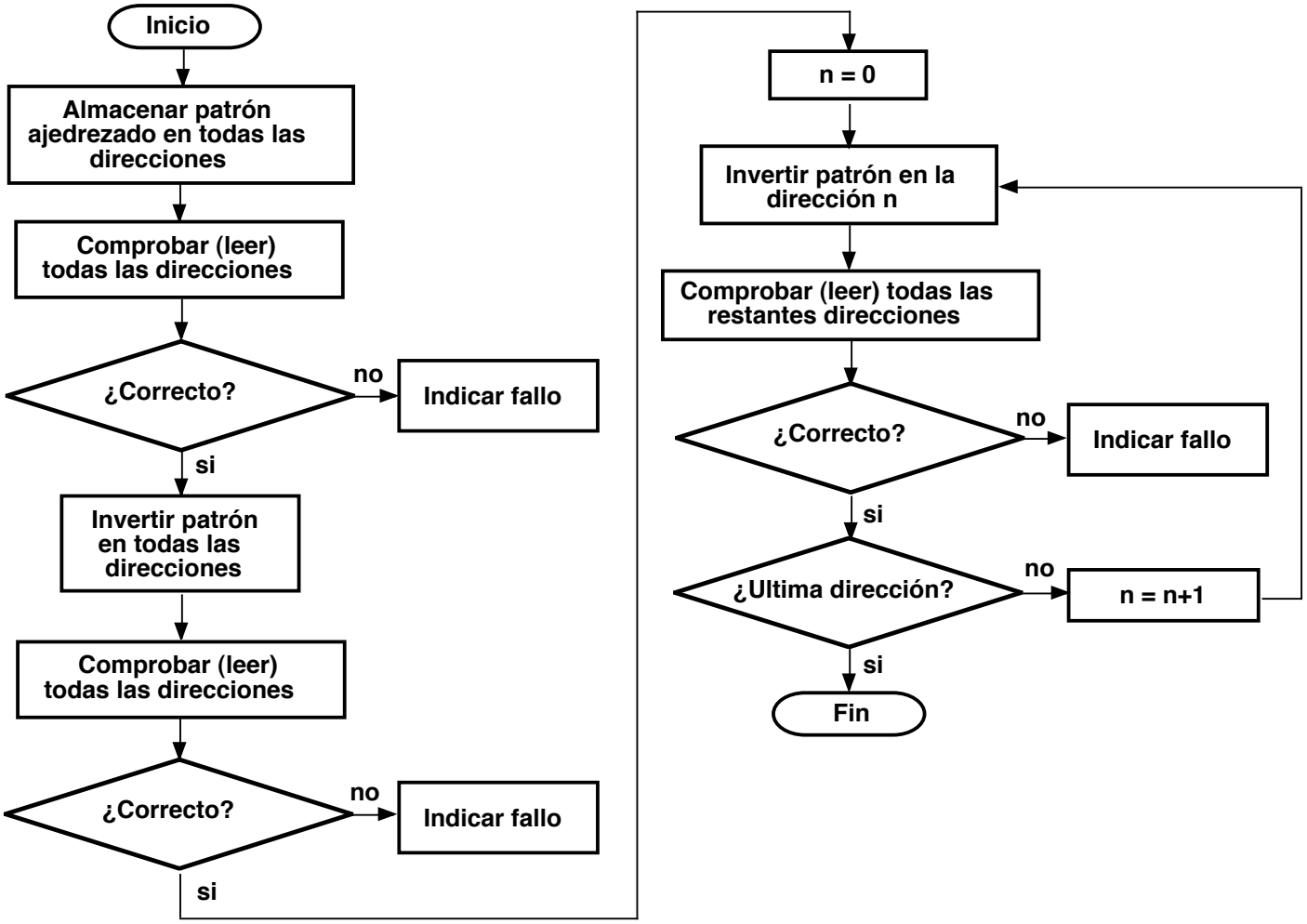
ADD	RAM							
0	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	0
2	1	0	1	0	1	0	1	0
3	0	1	0	1	0	1	0	1
4	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1

...

- **Detecta alteraciones dinámicas en el contenido de una dirección cuando el contenido de otra dirección cambia**
- **Adecuada a la comprobación automática dentro de un sistema digital**
  Rutina de comprobación del sistema (sistemas basados en microprocesador)

Método de prueba con patrón ajedrezado (3)

Verificación de memorias RAM



Mapa de Memoria

Microprocesador:

Bus de direcciones
Bus de datos
Bus de control

La memoria consiste en una hilera o lista ordenada de bytes. Cada byte ocupa una posición determinada (dirección) en la lista.

1 byte <=> 1 dirección lógica

Direcciones físicas:

$2^{\#\text{líneas de direcciones}} = 2^n$

Cada palabra consiste en m bytes

Líneas datos: m bytes (m x 8 bits)

#direcciones lógicas / dirección física = m

Si m = 1: #direcciones físicas = #direcciones lógicas

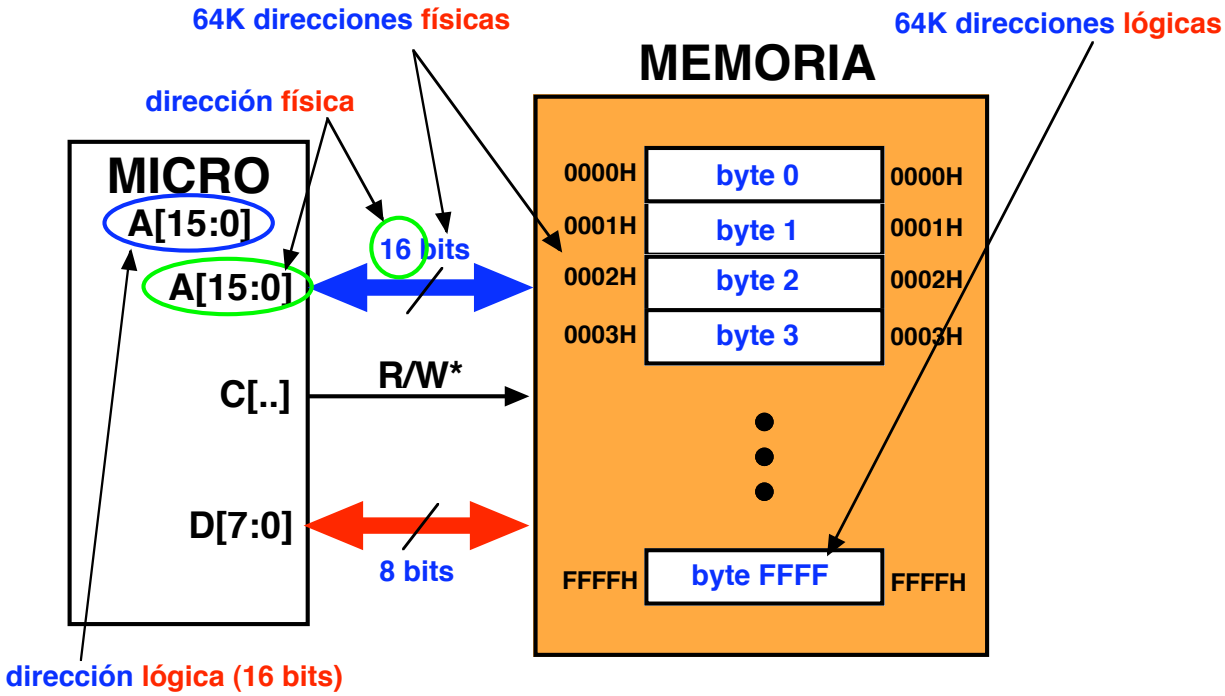
Mapa de Memoria (2)

Direcciones lógicas (Espacio de direccionamiento) :

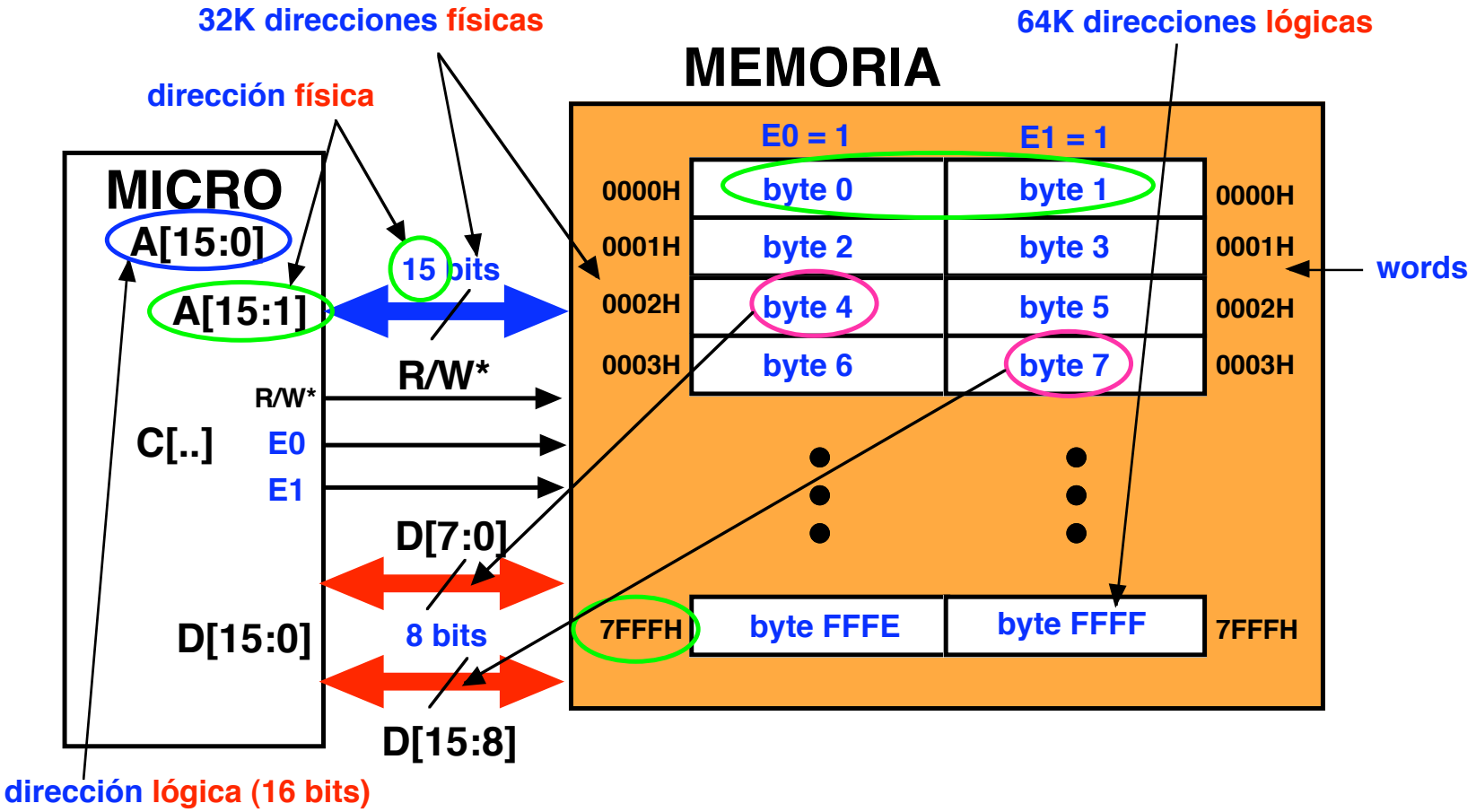
#direcciones físicas x (#direcciones lógicas / dirección física) = 2 $^{\#\text{líneas de direcciones}}$ x m = 2^n x m

Mapa de memoria:

Representa gráficamente los rangos de direcciones lógicas que tienen asignados los bloques de memoria (memoria, periféricos) que constituyen la memoria física



Mapa de Memoria (3)



A[0] = 0 : E0 = 1, E1 = 0 Lectura/Escritura dirección lógica par
A[0] = 1 : E0 = 0, E1 = 1 Lectura/Escritura dirección lógica impar
E0 = 1, E1 = 1 Lectura/Escritura word (dirección lógica debe ser par)

Mapa de Memoria (3)

Ejercicio

Un microprocesador puede direccionar una memoria de 128KB. Aunque la unidad de transferencia es la palabra de 2 bytes, utiliza únicamente un bus de 1 byte para transferencia de datos. El microprocesador se encarga de acceder dos veces a la memoria para leer/ escribir una palabra. La primera palabra se almacena en la dirección de memoria más baja. Se desea distribuir la memoria de la forma:

	Bits más significativos de la dirección			
	A_{n-1}	A_{n-2}	A_{n-3}	A_{n-4}
8K palabras de RAM	0	0	0	X
4K palabras libres	0	0	1	0
4K palabras de RAM	0	0	1	1
8K palabras libres	0	1	0	X
8k palabras de EPROM	0	1	1	X
Sin utilizar	1	X	X	X

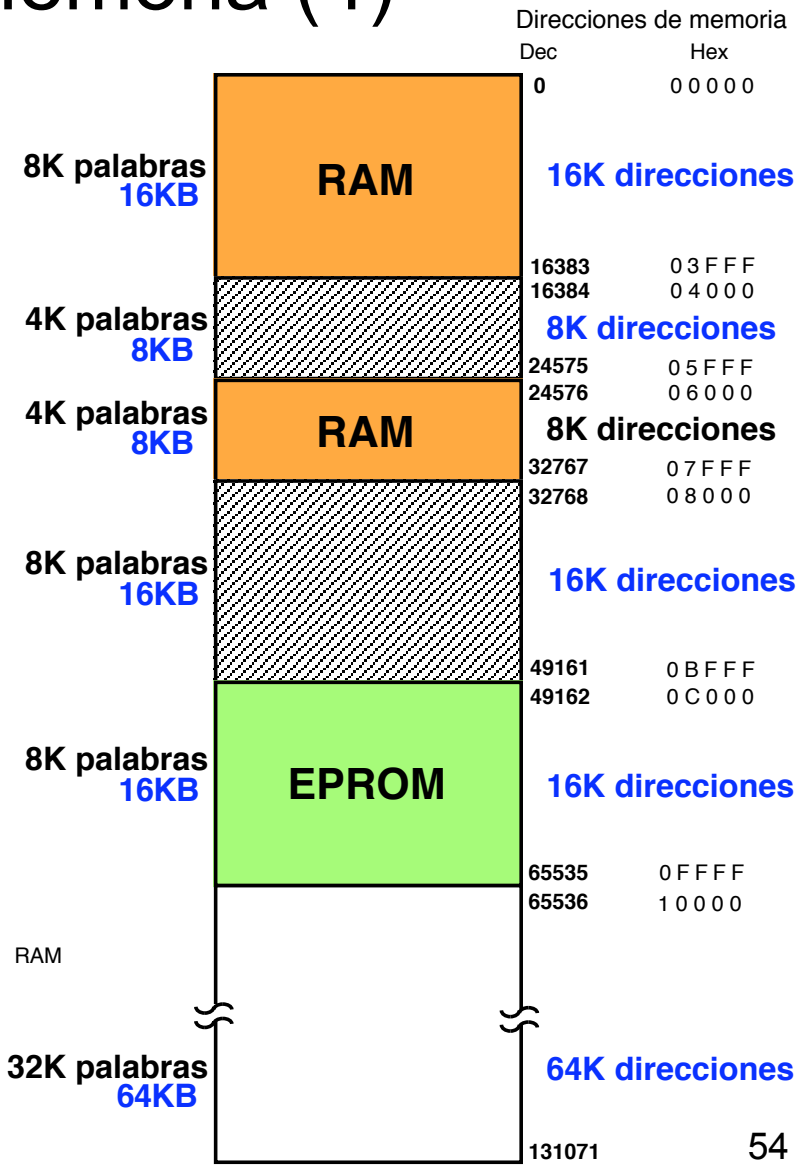
a) Mostrar el Mapa de Memoria

b) Suponiendo que se dispone únicamente de SRAM de 4Kx8, EPROM de 4Kx8 y DECs de 2 a 4 y de 3 a 8, mostrar una posible implementación de la lógica de descodificación y de las interconexiones entre el microprocesador y la memoria.

Mapa de Memoria (4)

Direcciones de memoria

A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11} – – – A_0			
					H_2	H_1	H_0	
			0	0	0	0	0	0
			0	1	F	F	F	8K-1
			1	1	F	F	F	16K-1
		1	0	0	0	0	0	16K
		1	0	1	F	F	F	24K-1
		1	1	0	0	0	0	24K
		1	1	1	F	F	F	32K-1
	1	0	0	0	0	0	0	32K
	1	0	0	1	F	F	F	40K-1
	1	0	1	1	F	F	F	48K-1
1	1	0	0	0	0	0	0	48K
1	1	0	0	1	F	F	F	56K-1
1	1	1	1	1	F	F	F	64K-1



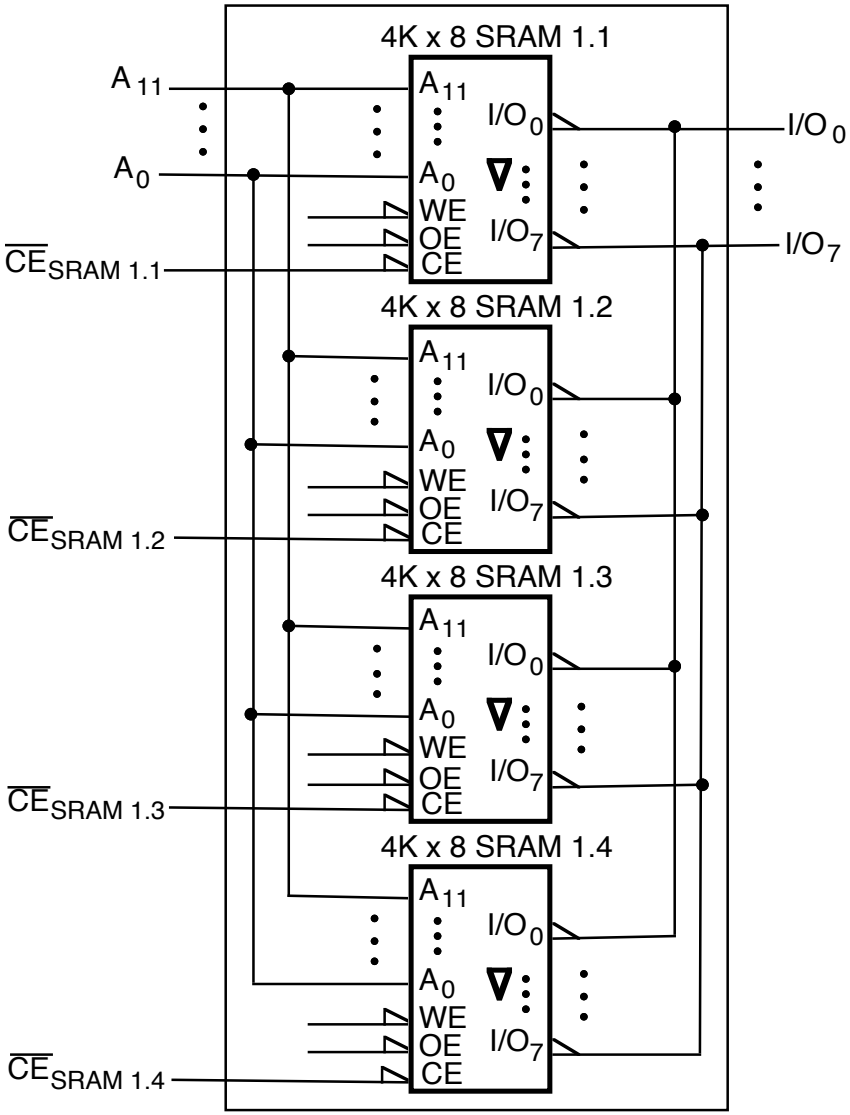
Mapa de Memoria (5)

Direcciones de memoria

A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11} – – – A_0			
					H_2	H_1	H_0	
			0	0	0	0	0	0
			0	1	F	F	F	8K-1
			1	1	F	F	F	16K-1
		1	0	0	0	0	0	16K
		1	0	1	F	F	F	24K-1
		1	1	0	0	0	0	24K
		1	1	1	F	F	F	32K-1
	1	0	0	0	0	0	0	32K
	1	0	0	1	F	F	F	40K-1
	1	0	1	1	F	F	F	48K-1
	1	1	0	0	0	0	0	48K
	1	1	0	1	F	F	F	56K-1
	1	1	1	1	F	F	F	64K-1

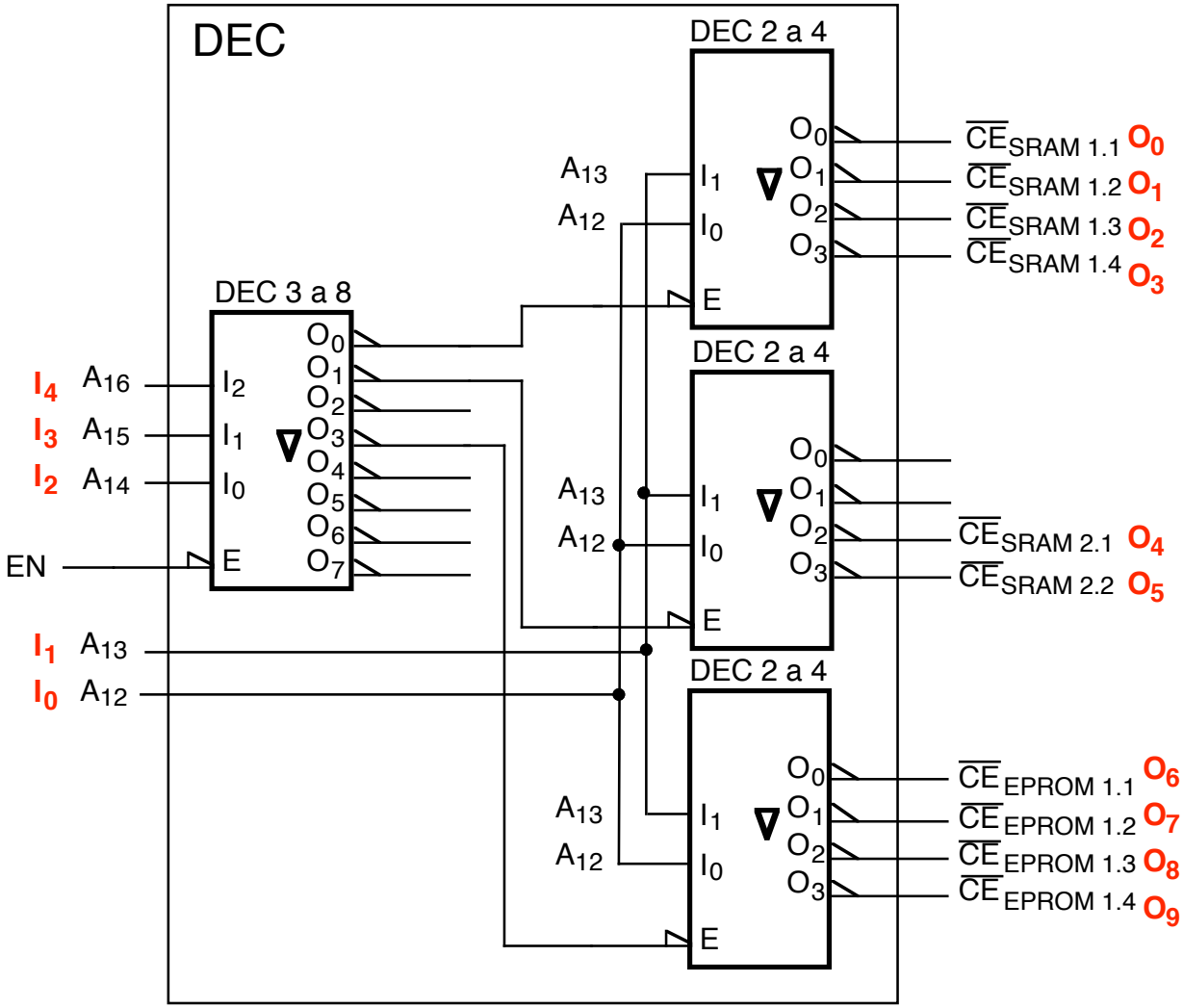
A_{16}	A_{15}	A_{14}	A_{13}	
0	0	0	X	RAM
0	0	1	0	
0	0	1	1	RAM
0	1	0	X	
0	1	1	X	EPROM
1	X	X	X	

Mapa de Memoria (6)



SRAM 16K x 8

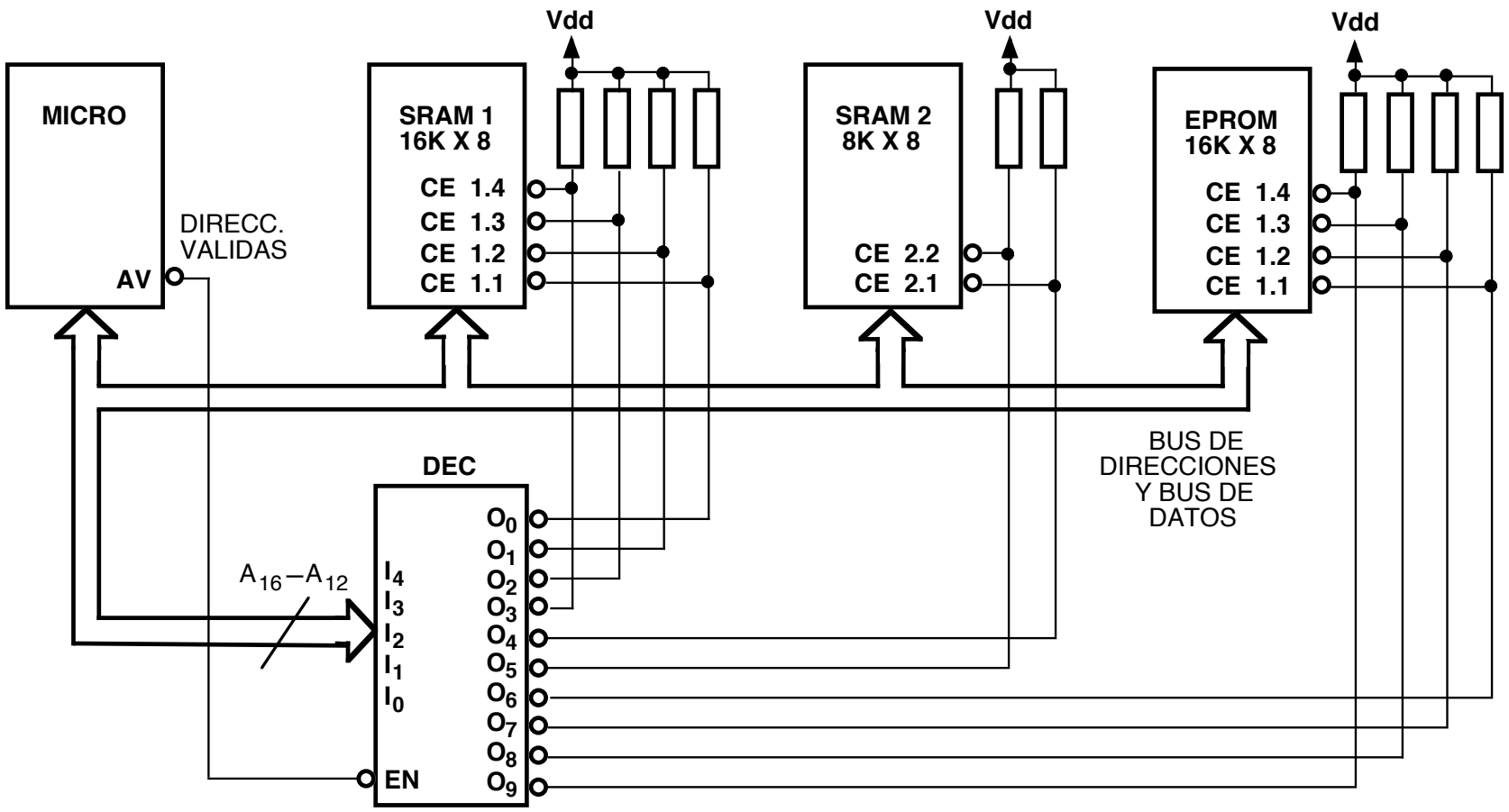
Mapa de Memoria (7)



A_{16}	A_{15}	A_{14}	A_{13}	
0	0	0	X	RAM
0	0	1	0	
0	0	1	1	RAM
0	1	0	X	
0	1	1	X	EPROM
1	X	X	X	

DECODIFICADOR

Mapa de Memoria (8)



Mapa de Memoria (9)

Ejercicio propuesto

Se desea configurar un mapa de memoria de 64Kbytes (0000H-FFFFH) de tal modo que las direcciones 0000-1FFF correspondan a memoria RAM, las direcciones 4000-4FFF a EEPROM, las 7000-7007 a un dispositivo periférico y las F000-FFFF a memoria ROM.
Suponiendo que se dispone de chips de 2K x 8 RAM, 4K x 8 EEPROM, 2K x 8 ROM, y de decodificadores, mostrar el Mapa de Memoria y una posible implementación circuital.

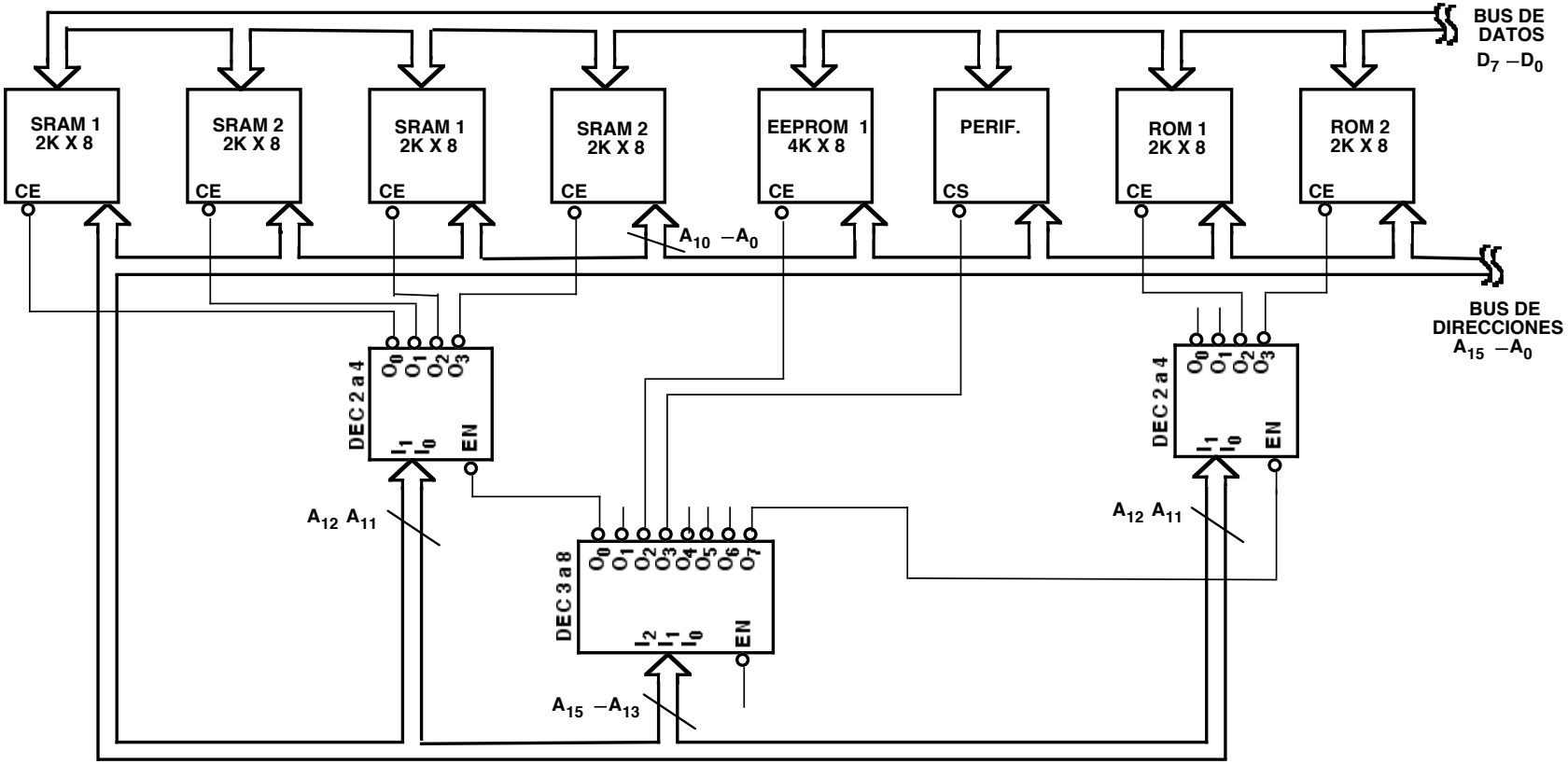
Mapa de Memoria (10)

Ejercicio

Asumiendo un bus de direcciones de 16 bits y un bus de datos de 8 bits:

				15	11	07	03
0000-1FFF:	8KB de RAM	0000-07FF	2KB RAM1	0000	0XXX	XXXX	XXXX
		0800-0FFF	2KB RAM2	0000	1XXX	XXXX	XXXX
		1000-17FF	2KB RAM3	0001	0XXX	XXXX	XXXX
		1800-1FFF	2KB RAM4	0001	1XXX	XXXX	XXXX
2000-3FFF:	8KB libres						
4000-4FFF:	4KB EEPROM	4000-4FFF	4KB EEPROM1	0100	XXXX	XXXX	XXXX
5000-6FFF:	8KB libres						
7000-7007:	8B Periférico	7000-7007	8B Periférico1	0111	0000	0000	0XXX
7008-EFFF:	(32K-8)B (32760B) libres						
F000-FFFF:	4KB ROM	F000-F7FF	2KB ROM1	1111	0XXX	XXXX	XXXX
		F800-FFFF	2KB ROM2	1111	1XXX	XXXX	XXXX

Mapa de Memoria (11)



Mapa de Memoria (ExJ08)

Asumiendo un bus de direcciones de 16 bits y un bus de datos de 8 bits:

				15	11	07	03
0000-0FFF:	4KB libres						
1000-2FFF:	8KB de RAM	1000-17FF	2KB RAM1	0001	0XXX	XXXX	XXXX
		1800-1FFF	2KB RAM2	0001	1XXX	XXXX	XXXX
		2000-27FF	2KB RAM3	0010	0XXX	XXXX	XXXX
		2800-2FFF	2KB RAM4	0010	1XXX	XXXX	XXXX
3000-4FFF:	8KB libres						
5000-5FFF:	4KB EEPROM	5000-5FFF	4KB EEPROM1	0101	XXXX	XXXX	XXXX
6000-7FFF:	8KB libres						
8000-800B:	12B Periférico	8000-8007	8B Periférico1	1000	0000	0000	0XXX
		8008-800B	4B Periférico1	1000	0000	0000	10XX
800C-DFFF:	(24K-12)B (24564B) libres						
E000-EFFF:	4KB ROM	E000-E7FF	2KB ROM1	1110	0XXX	XXXX	XXXX
		E800-EFFF	2KB ROM2	1110	1XXX	XXXX	XXXX

F000-FFFF: 4

Mapa de Memoria (ExJ08 2)