

# Tema 2: Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

1.1 Conceptos básicos

1.2 Clasificación de memorias semiconductoras.

1.3 Memorias de Acceso aleatorio

- Memorias de sólo lectura (ROM).

- Memorias de lectura y escritura (RAM)

- Expansión de memorias.

1.4 Memorias de Acceso Secuencial

## 2. Dispositivos programables: Arquitectura básica de una FPGA.

NO son apuntes de la asignatura, sólo material de apoyo a las clases de teoría

Bibliografía:

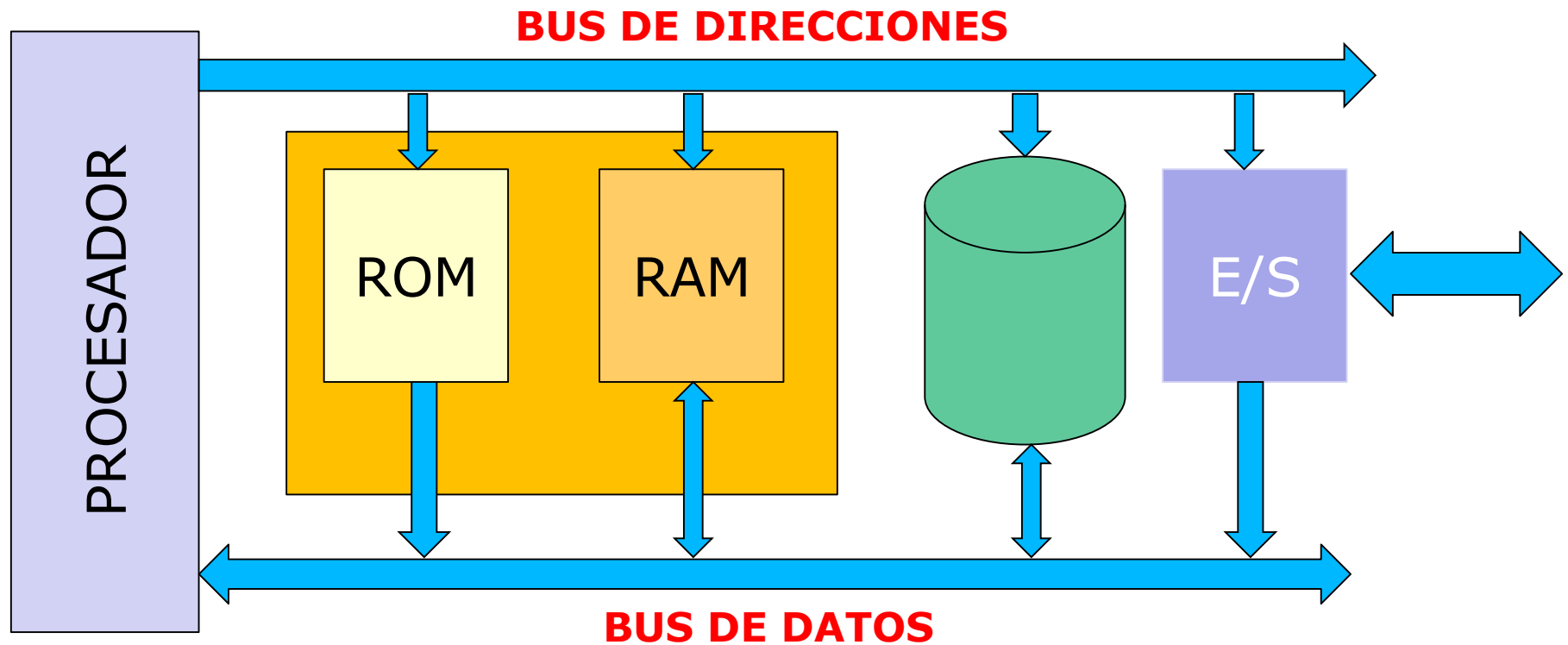
Floyd. **Digital Fundamentals. 10th Ed.** (Ch.10, Ch.11)

# 1. Conceptos básicos

---

- Computador: un sistema digital complejo capaz de ejecutar un programa almacenado en memoria.
- Los computadores necesitan dispositivos de memoria para:
  - almacenar los programas
  - mantener los datos generados durante el procesamiento.
- En informática, normalmente el término *memoria* hace referencia a las memorias RAM y ROM y el término *almacenamiento* hace referencia a los discos y demás dispositivos externos.

# Organización básica de un computador

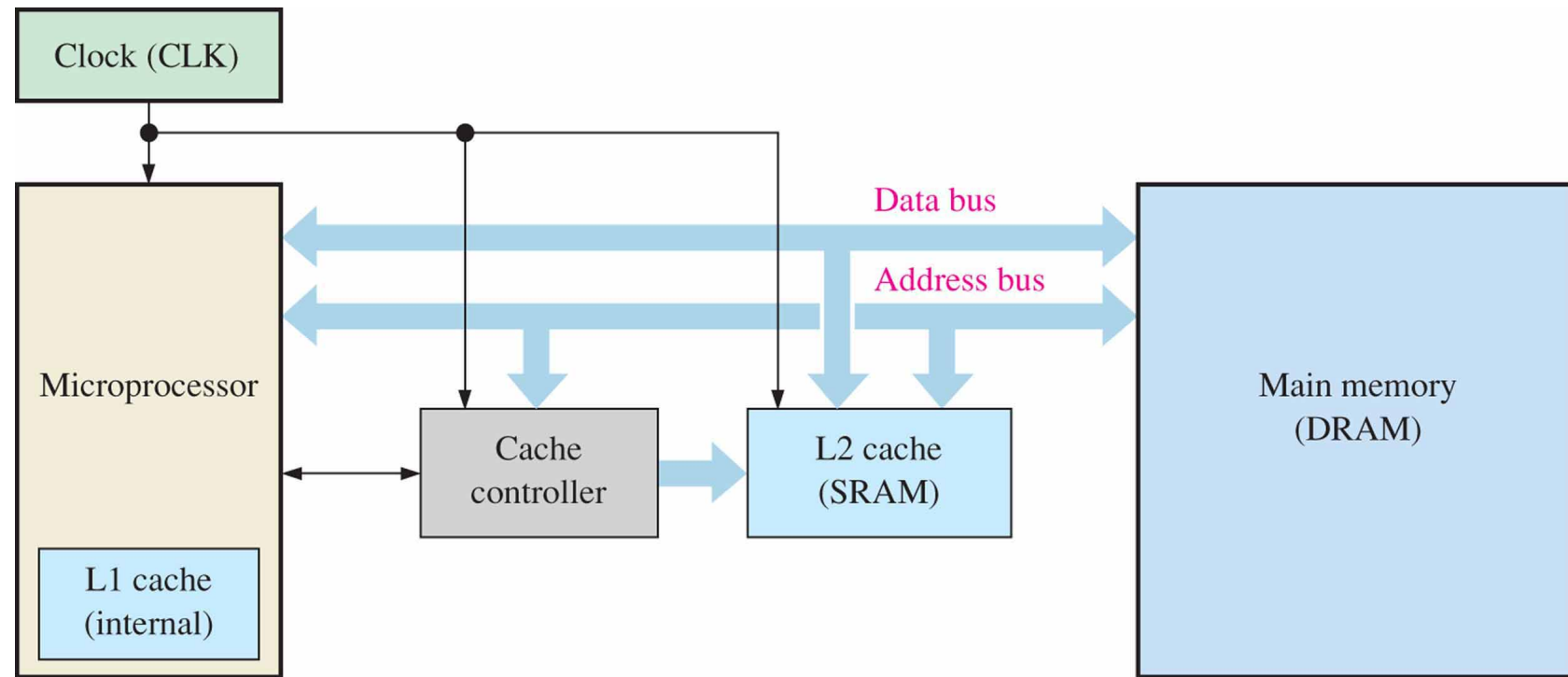


# Jerarquía de memorias

---



# Ejemplo de uso de memorias en un computador



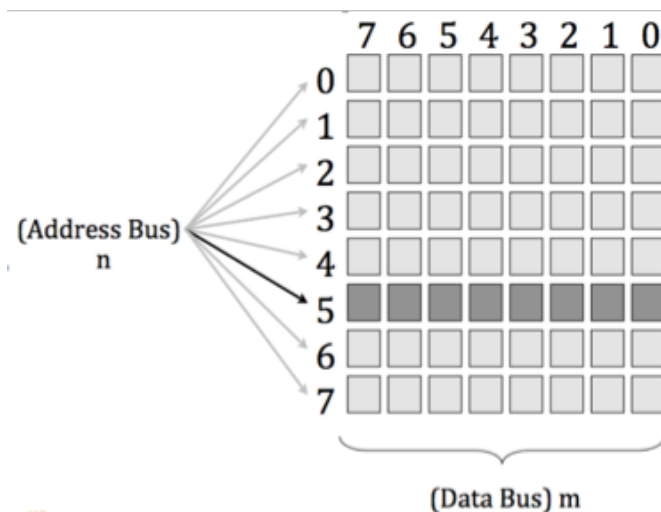
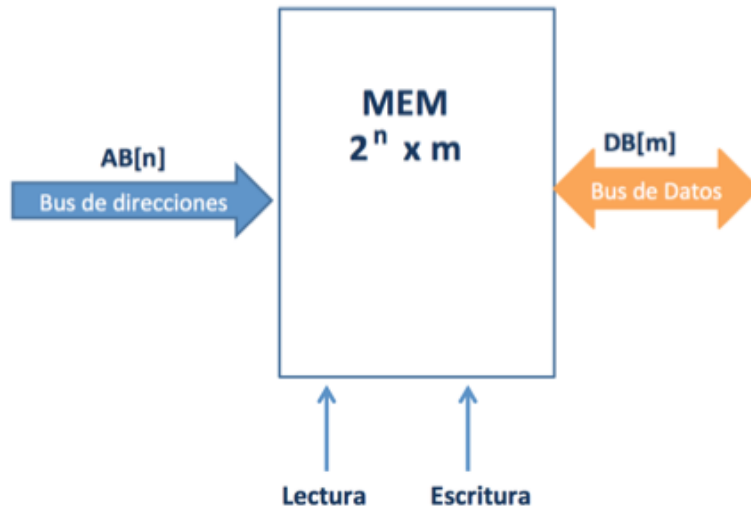
# Matriz básica de memoria

- Las memorias están formadas por **matrices de celdas**.
- En cada celda se almacena 1 bit de información
- Cada **fila** de esa matriz de memoria se denomina **palabra** y representa la información que puede leerse/escribirse en cada acceso a la misma.

	7	6	5	4	3	2	1	0
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

16 palabras de 8 bits

# Direccionamiento y capacidad de las memorias



- La **capacidad** de una memoria es el número total de bits que puede almacenar,  $2^n \times m$ .

- La posición de una palabra en una memoria se denomina **dirección**

# Capacidad de las memorias

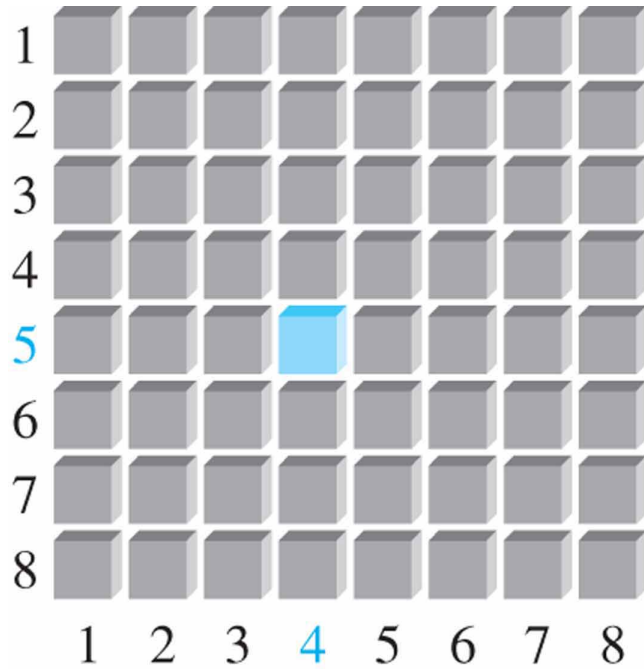
Unidades de información (del byte)			
Sistema Internacional (decimal)		ISO/IEC 80000-13 (binario)	
Múltiplo (símbolo)	SI	Múltiplo (símbolo)	ISO/IEC
kilobyte (kB)	$10^3$	kibibyte (KiB)	$2^{10}$
megabyte (MB)	$10^6$	mebibyte (MiB)	$2^{20}$
gigabyte (GB)	$10^9$	gibibyte (GiB)	$2^{30}$
terabyte (TB)	$10^{12}$	tebibyte (TiB)	$2^{40}$
petabyte (PB)	$10^{15}$	pebibyte (PiB)	$2^{50}$
exabyte (EB)	$10^{18}$	exbibyte (EiB)	$2^{60}$
zettabyte (ZB)	$10^{21}$	zebibyte (ZiB)	$2^{70}$
yottabyte (YB)	$10^{24}$	yobibyte (YiB)	$2^{80}$

Véase también: nibble • byte • sistema octal

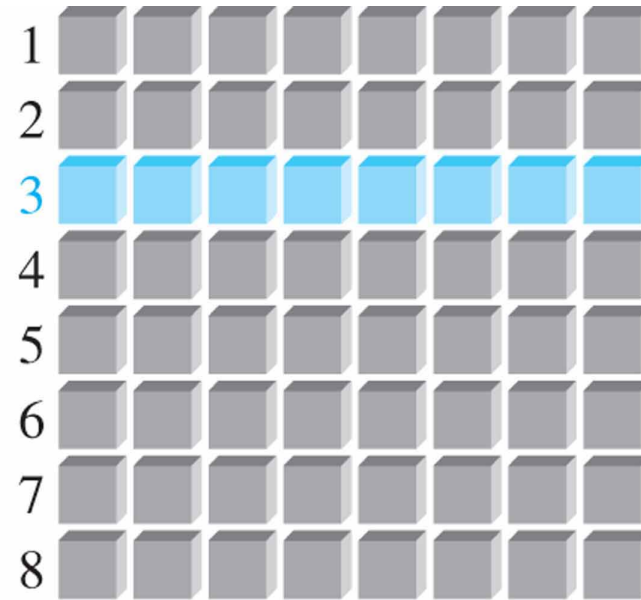
(wikipedia.org)



# Organización bidimensional de una memoria

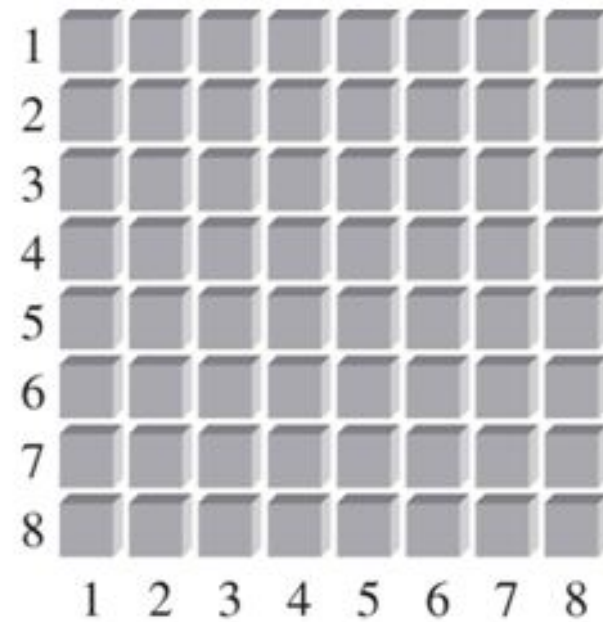


(a) The address of the blue bit is row 5, column 4.

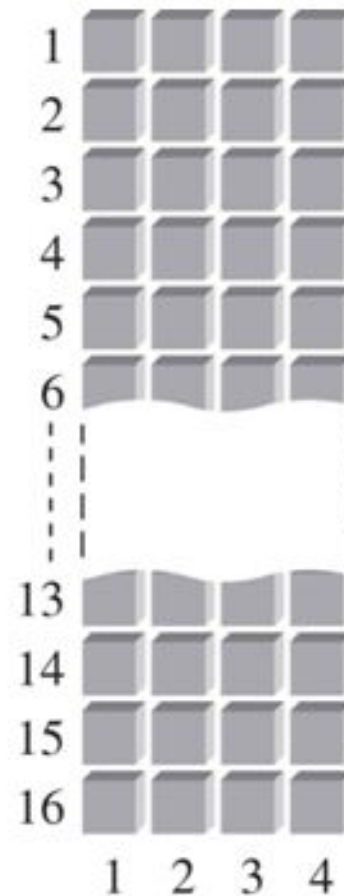


(b) The address of the blue byte is row 3.

# Ejemplo de memorias con la misma capacidad (64 bits) y diferente organización



(a)  $8 \times 8$  array



(b)  $16 \times 4$  array



(c)  $64 \times 1$  array

# Operaciones básicas de las memorias

Al tratarse de dispositivos de almacenamiento, las memorias cuentan con dos operaciones básicas:

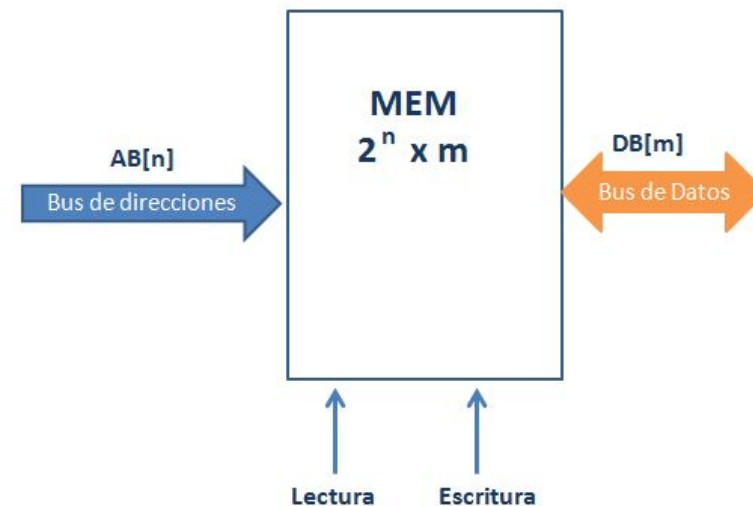
**Escritura** (*write*): permite almacenar una palabra en una determinada dirección de la memoria.

**Lectura** (*read*): permite recuperar la palabra almacenada en una determinada dirección de la memoria.

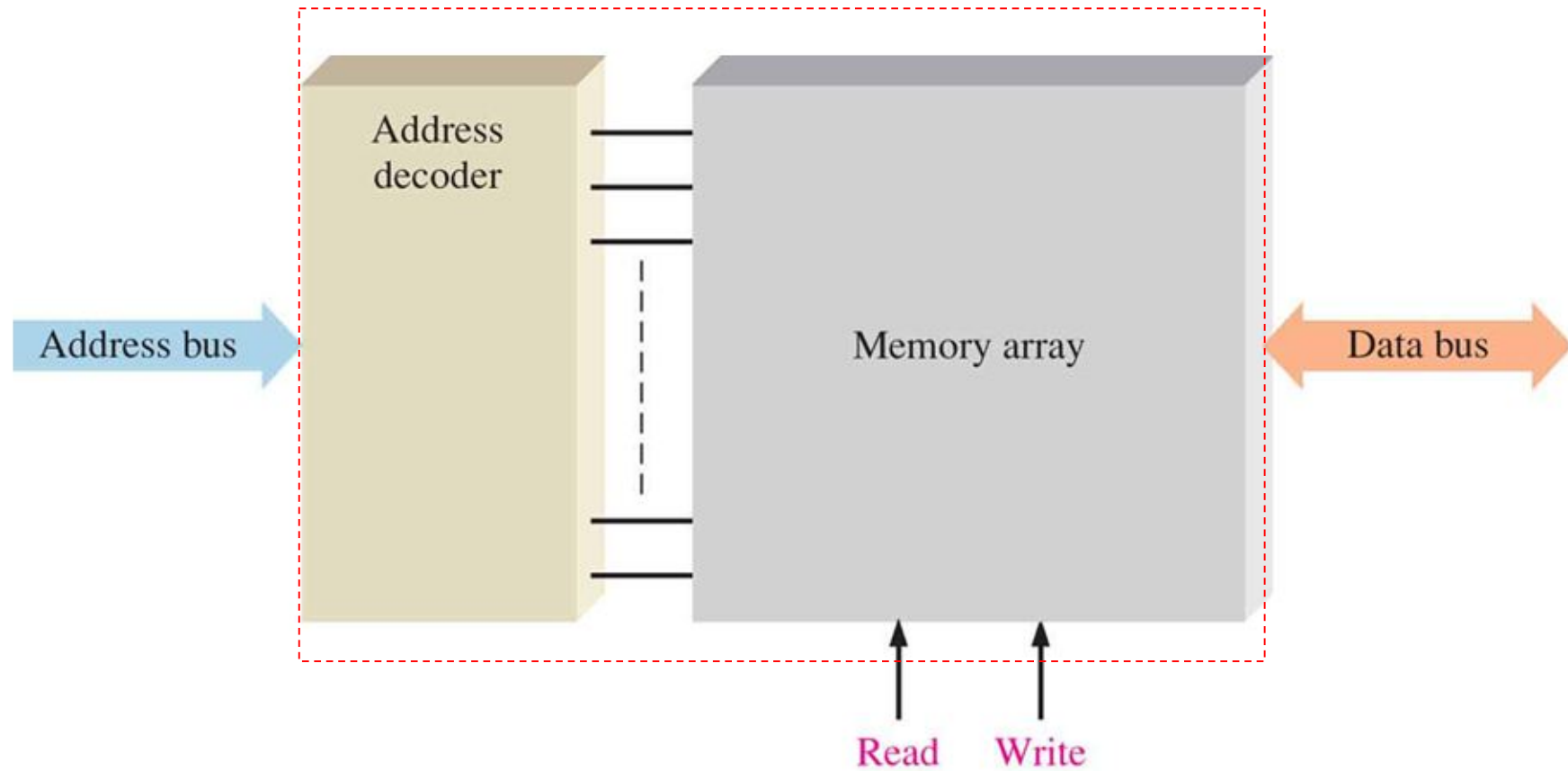
Para la implementación de estas operaciones se necesitan dos buses:

**Bus de direcciones** (*address bus*): para indicar la dirección de lectura/escritura.

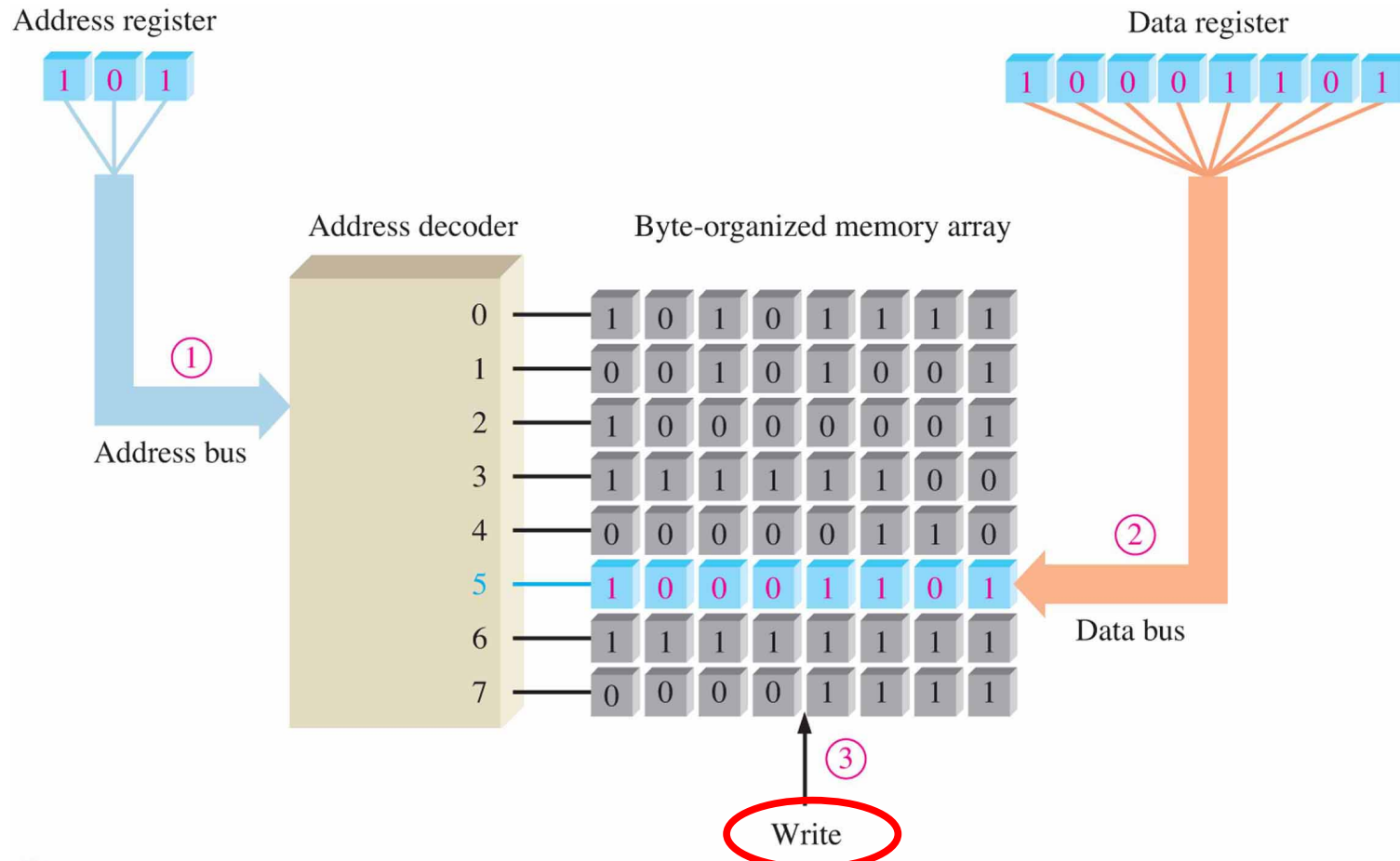
**Bus de datos** (*data bus*): para leer/escribir la palabra en sí.



# Estructura interna de una memoria

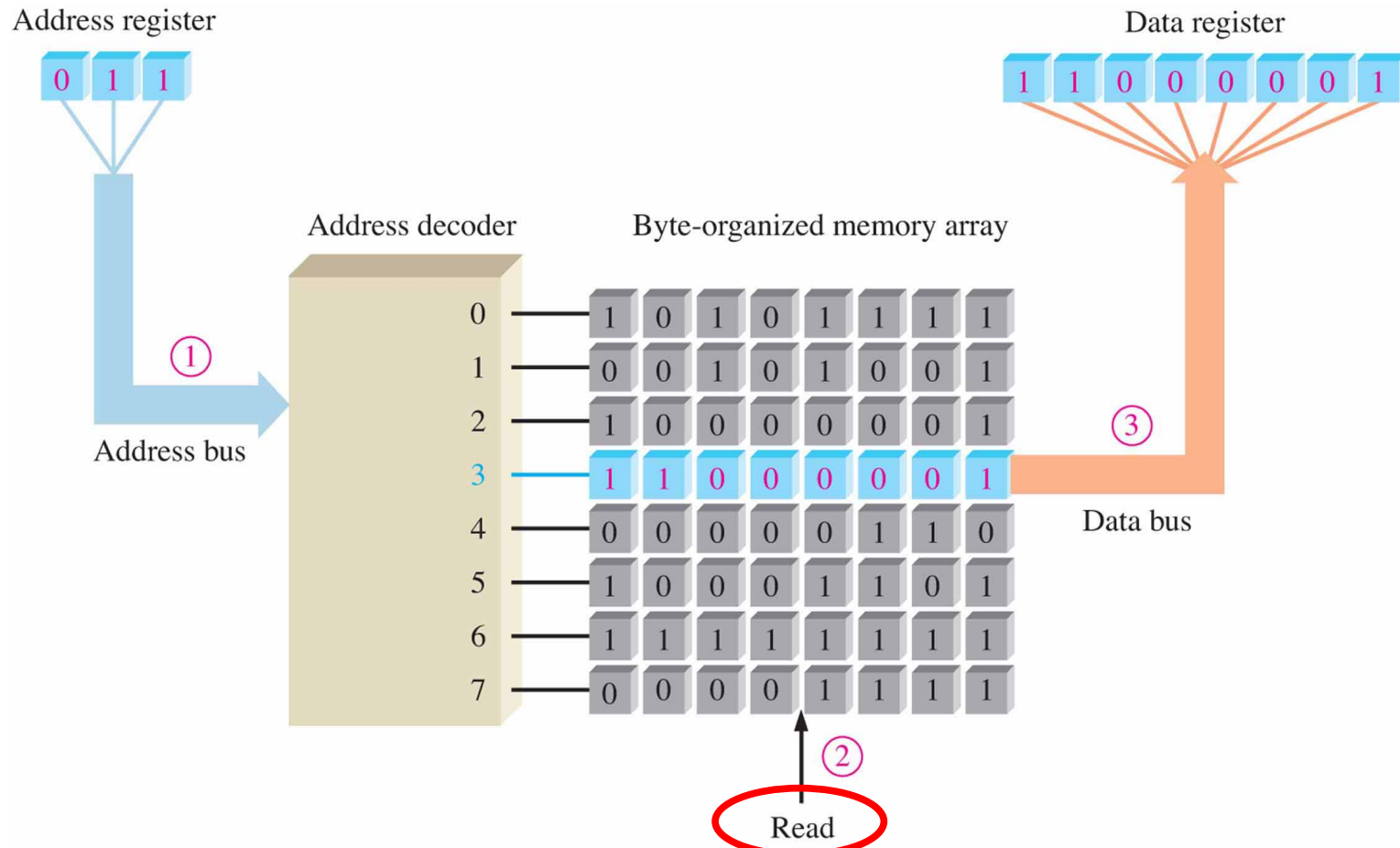


# Escritura en memoria



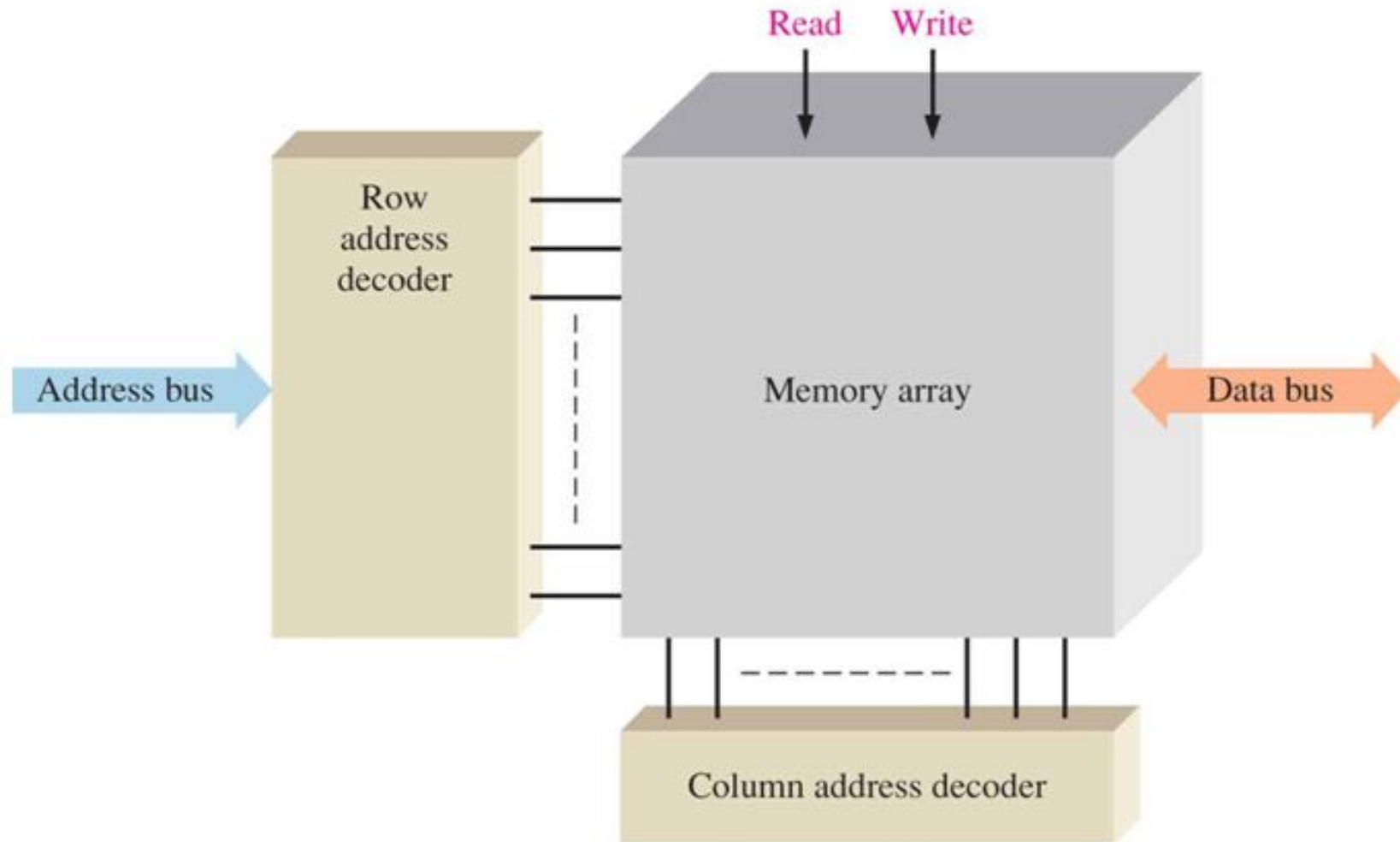
- ① Address code 101 is placed on the address bus and address 5 is selected.
- ② Data byte is placed on the data bus.
- ③ Write command causes the data byte to be stored in address 5, replacing previous data.

# Lectura de memoria

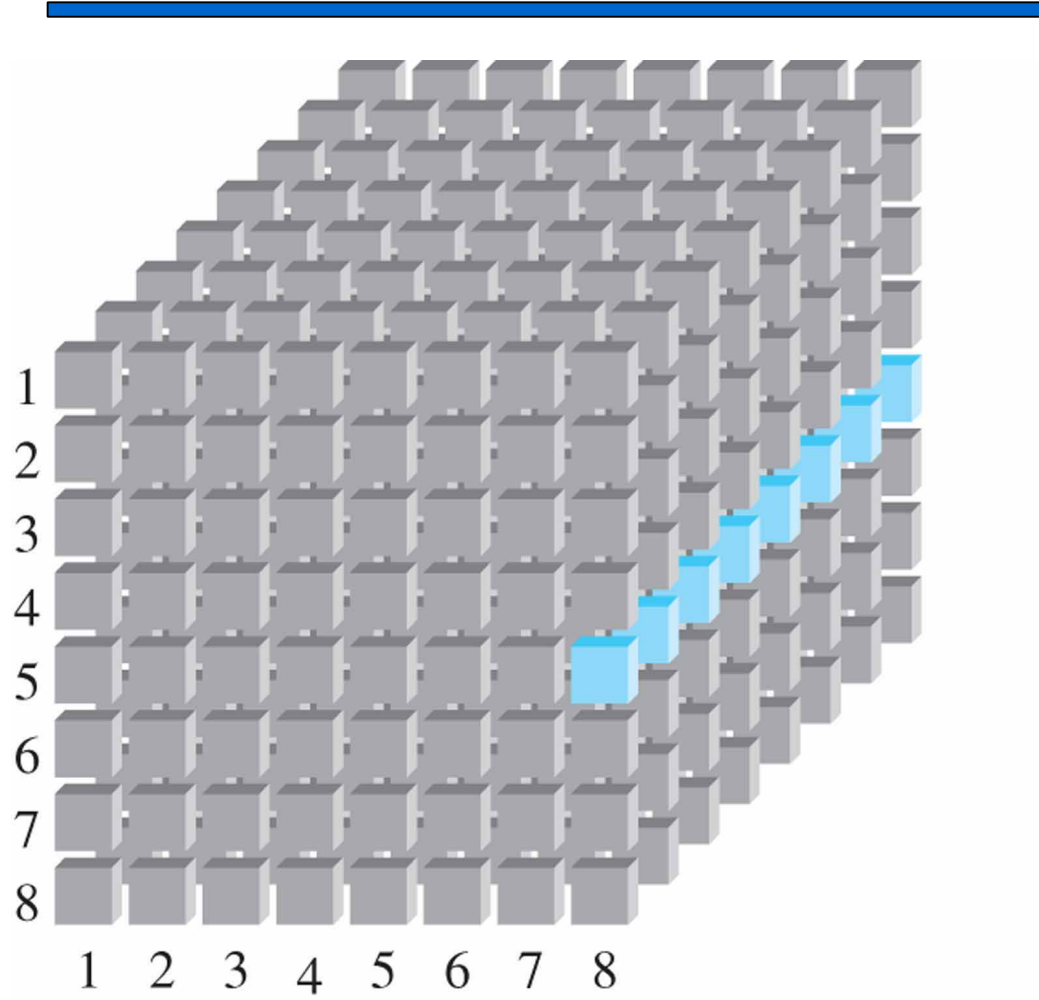


- ① Address code 011 is placed on the address bus and address 3 is selected.
- ② Read command is applied.
- ③ The contents of address 3 is placed on the data bus and shifted into data register. The contents of address 3 is not erased by the read operation.

# Organización tridimensional de una memoria



# Organización tridimensional de una memoria



The address of the blue byte is row 5, column 8.





# Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

### 1.1 Conceptos básicos

### *1.2 Clasificación de memorias semiconductoras.*

### 1.3 Memorias de Acceso aleatorio

- Memorias de sólo lectura (ROM).
- Memorias de lectura y escritura (RAM)
- Expansión de memorias.

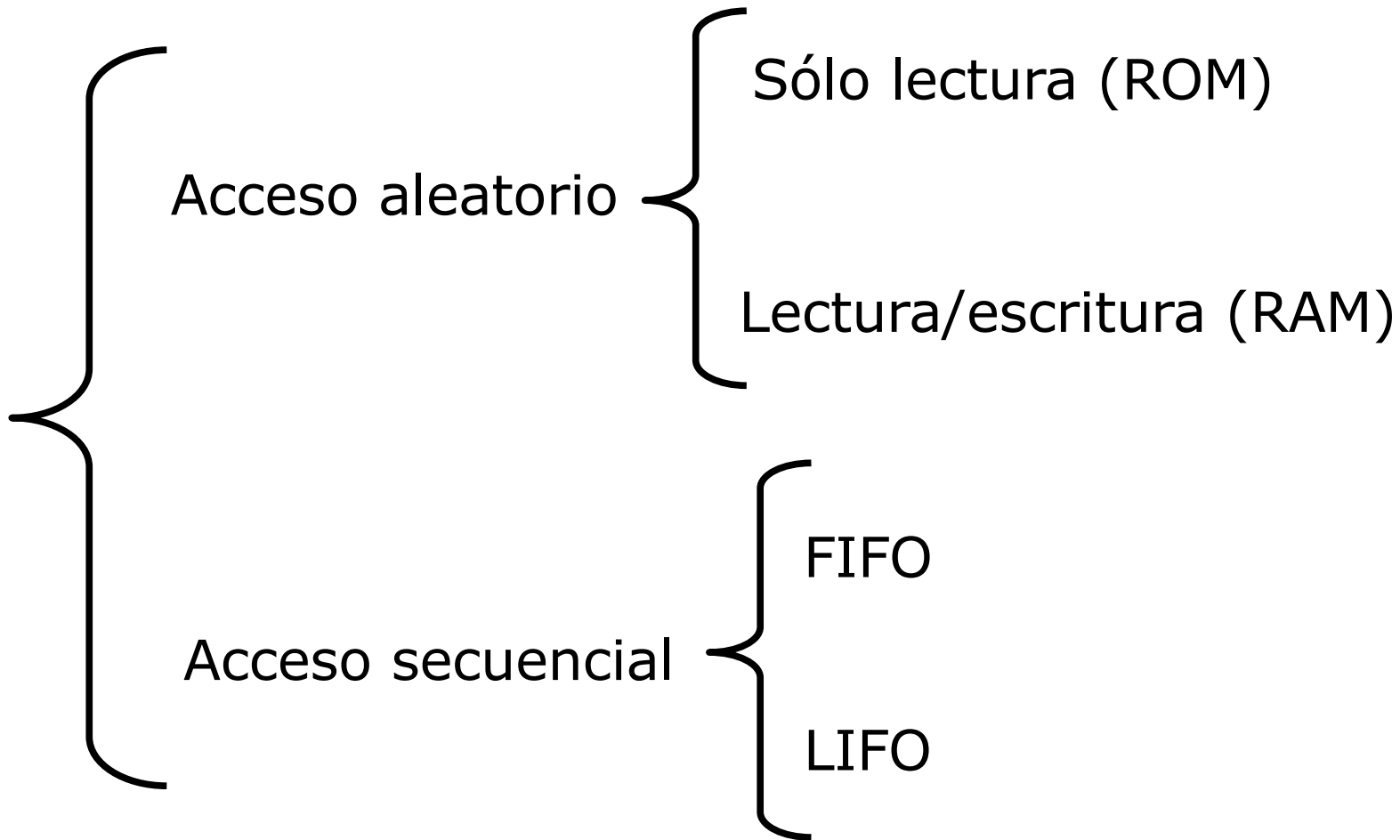
### 1.4 Memorias de Acceso Secuencial

### 1.5 Otros tipos especiales de memorias.

## 2. Dispositivos programables: Arquitectura básica de una FPGA.

# Clasificación de memorias semiconductoras

---



# Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

1.1 Conceptos básicos

1.2 Clasificación de memorias semiconductoras.

*1.3 Memorias de Acceso aleatorio*

- *Memorias de sólo lectura (ROM).*

- Memorias de lectura y escritura (RAM)

- Expansión de memorias.

1.4 Memorias de Acceso Secuencial

1.5 Otros tipos especiales de memorias.

## 2. Dispositivos programables: Arquitectura básica de una FPGA.

# Las memorias RAM y ROM

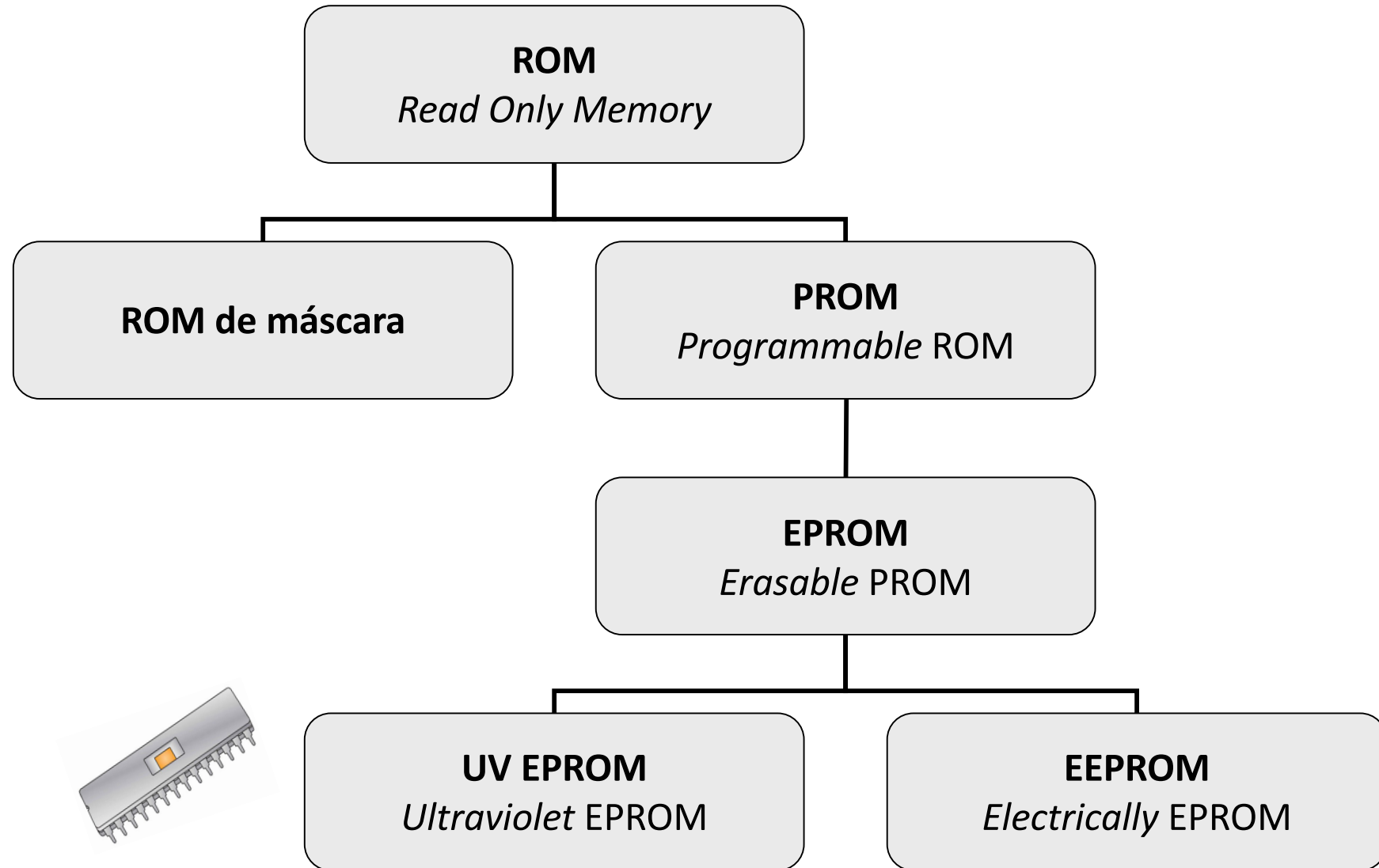
---

Las dos principales categorías de memorias semiconductoras de acceso aleatorio son:

**ROM** (*Read-Only Memory*): sólo poseen capacidad de lectura y son no volátiles.

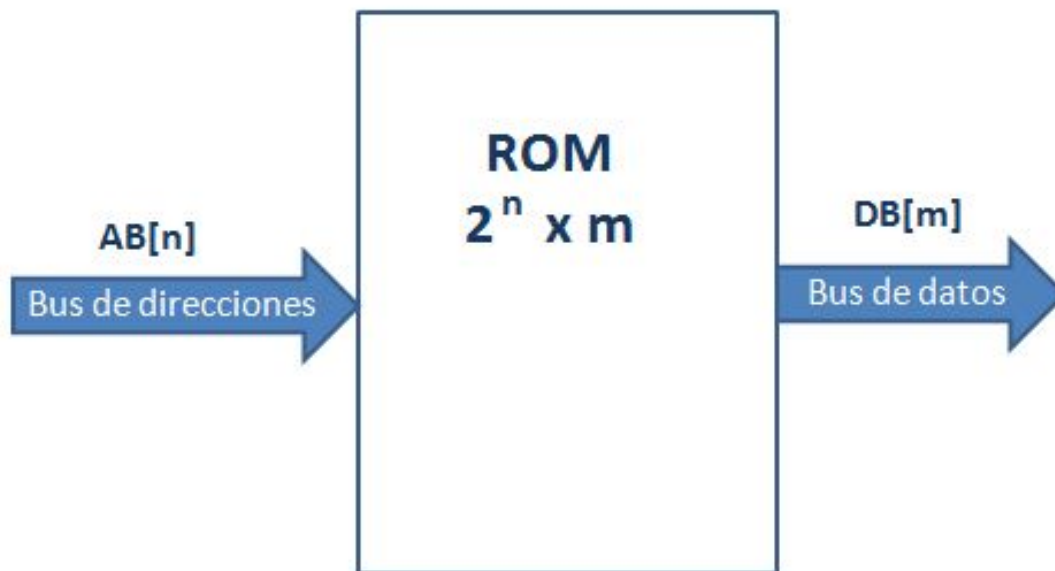
**RAM** (*Random-Access Memory*): poseen capacidad de lectura y escritura y son volátiles.

# Familia de memorias ROM



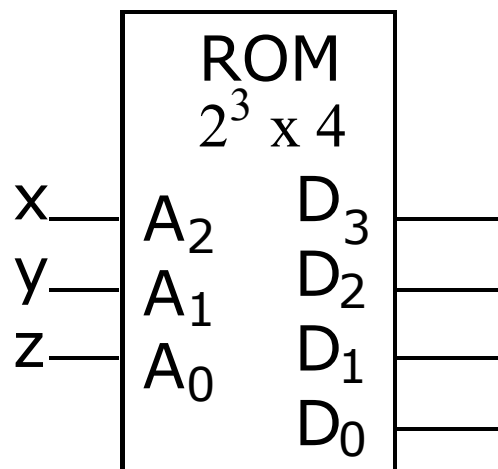
# Memorias ROM

- Una ROM permite almacenar  $2^n$  datos de  $m$  bits
- No volátiles
- Se graba durante la fabricación (ROM) o en laboratorio (EPROM, EEPROM...)



- AB[n]: Address BUS  
 $[A_{n-1}:A_0]$
- DB[m]: Data BUS  
 $[D_{m-1}:D_0]$

# Ejemplo: ROM $2^3 \times 4$ (8 posiciones de 4 bits)

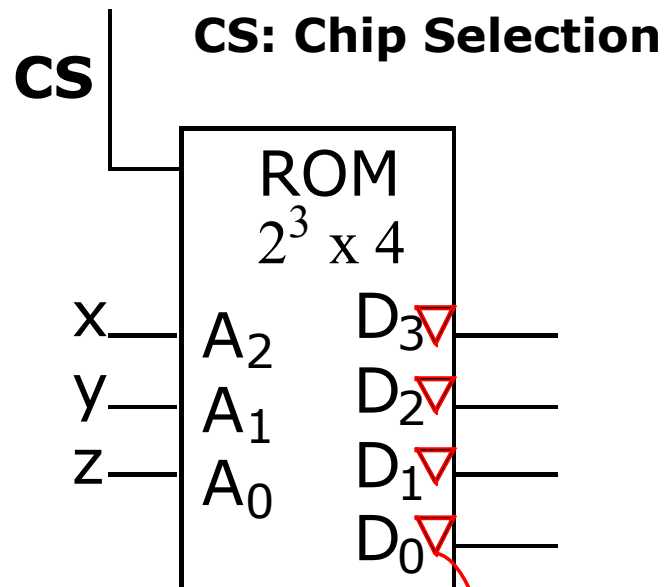


A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	1	1
0	0	1	1	0	0	0
0	1	0	1	0	1	0
0	1	1	1	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

En cada columna se realiza una función (cuando CS=1).

P. ej.  $D_0 = \Sigma (m_0, m_3, m_4, m_5) = \Pi (M_1, M_2, M_6, M_7)$

# Ejemplo: ROM $2^3 \times 4$ (8 posiciones de 4 bits)



CS	A2	A1	A0	D3	D2	D1	D0
0	-	-	-	HI	HI	HI	HI
1	0	0	0	0	0	1	1
1	0	0	1	1	0	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	0	1	1	1
1	1	0	1	0	1	1	1
1	1	1	0	1	1	0	0
1	1	1	1	0	1	0	0

indica capacidad tri-state

En cada columna se realiza una función (cuando CS=1).

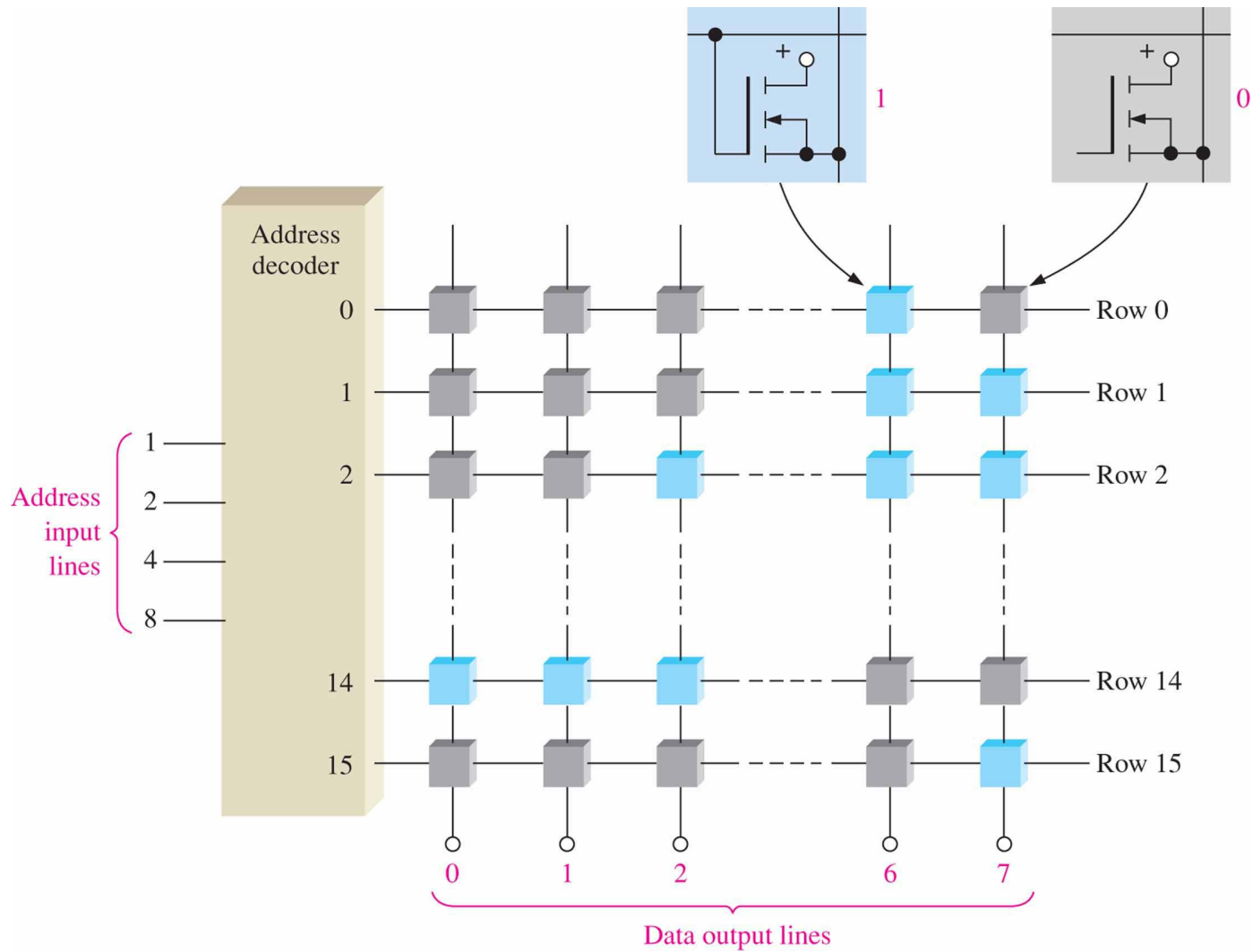
P. ej.  $D_0 = \Sigma (m_0, m_3, m_4, m_5) = \Pi (M_1, M_2, M_6, M_7)$

**Cuando CS no activado, salidas en ALTA IMPEDANCIA**

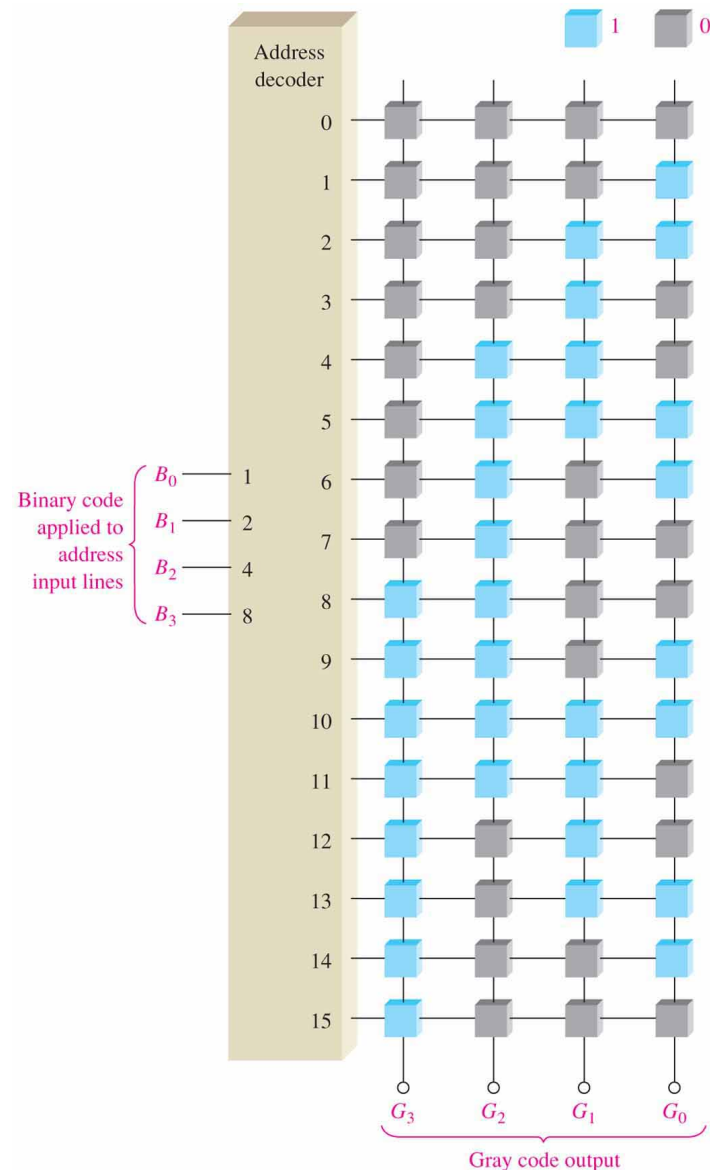


# Organización interna de una ROM

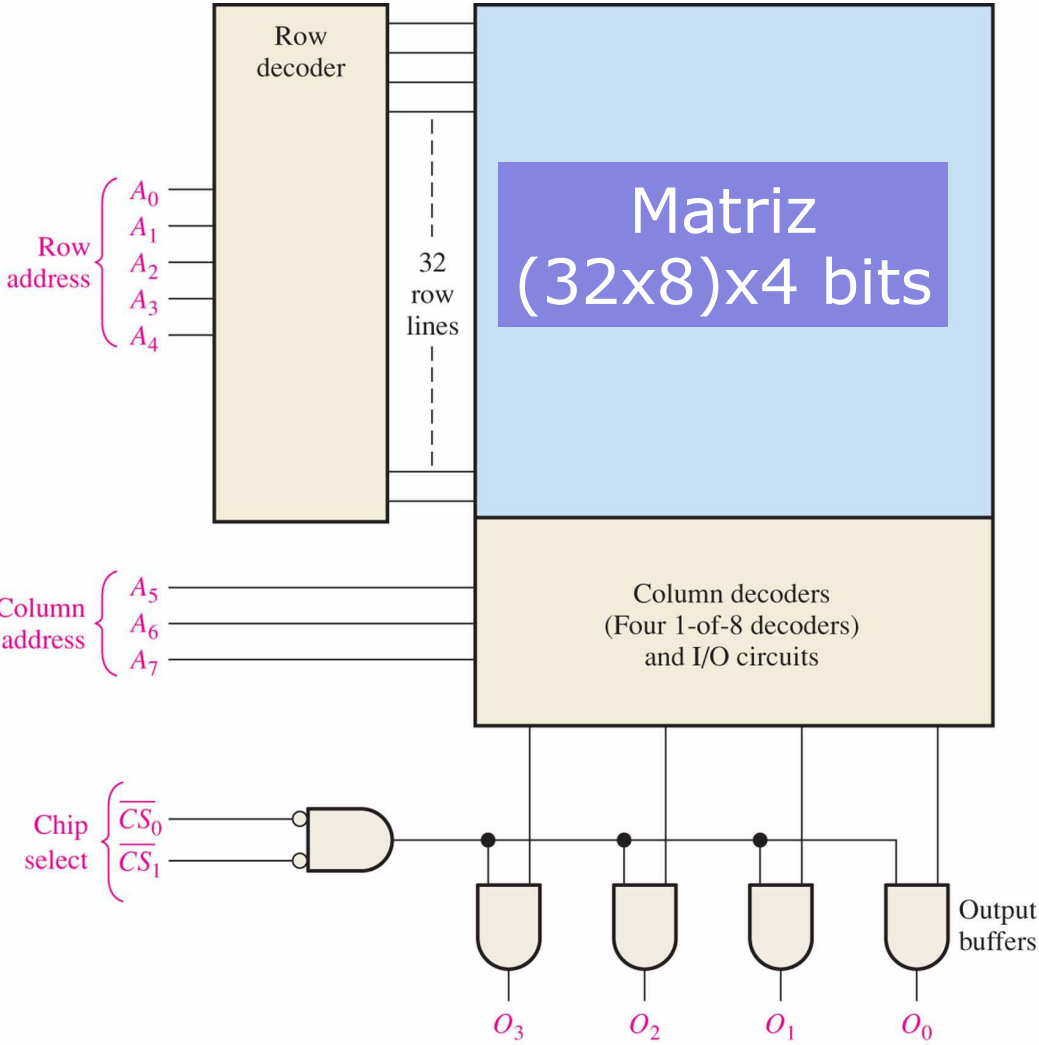
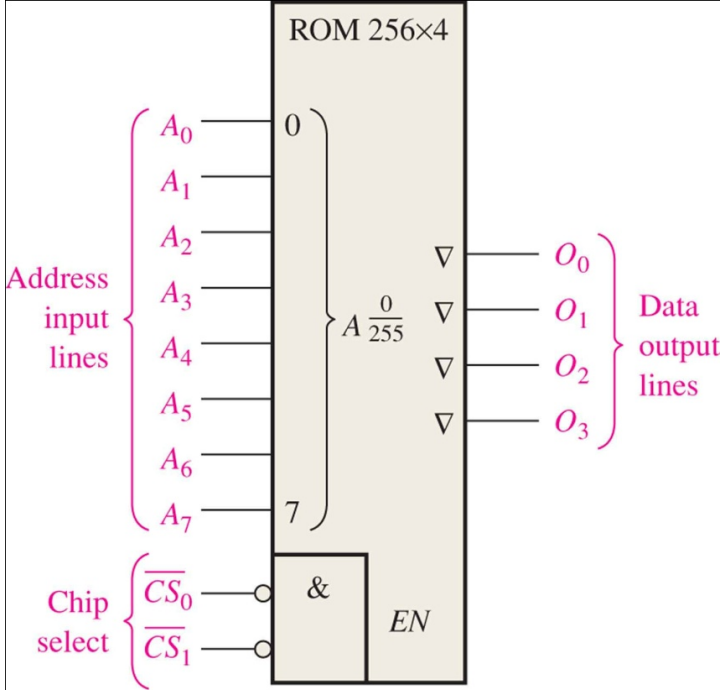
## Ej.: ROM 16x8



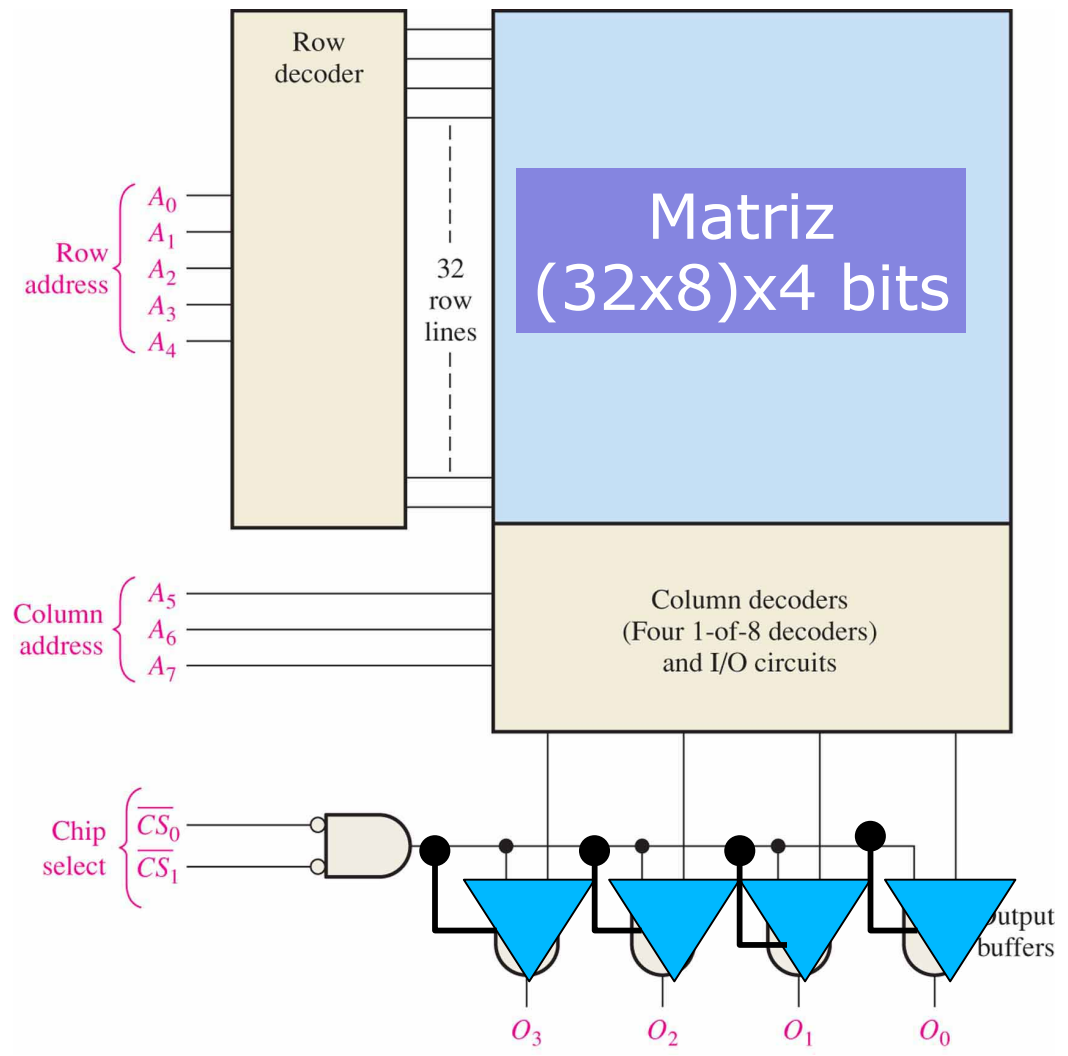
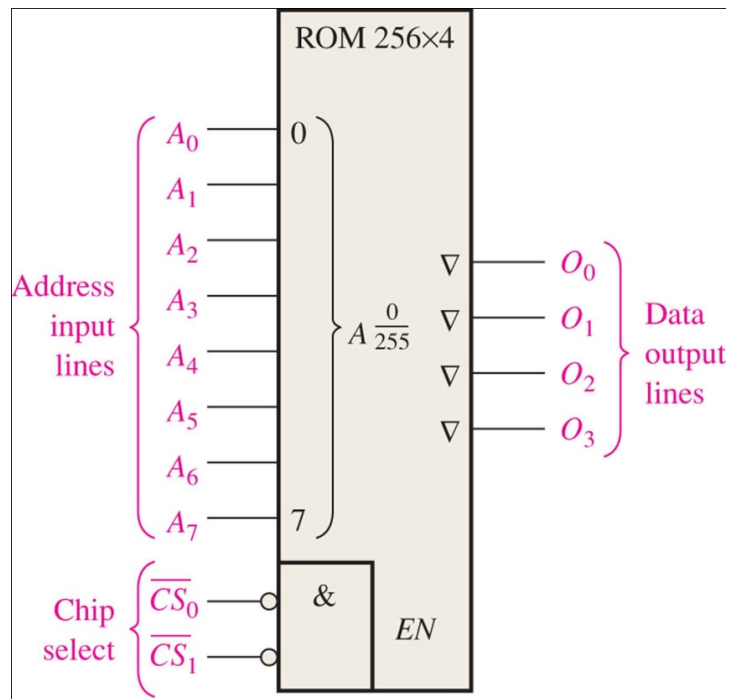
# Ejemplo de uso de la ROM: Convertidor de código Binario natural a Gray



# Ejemplo de organización tridimensional: ROM 256x4

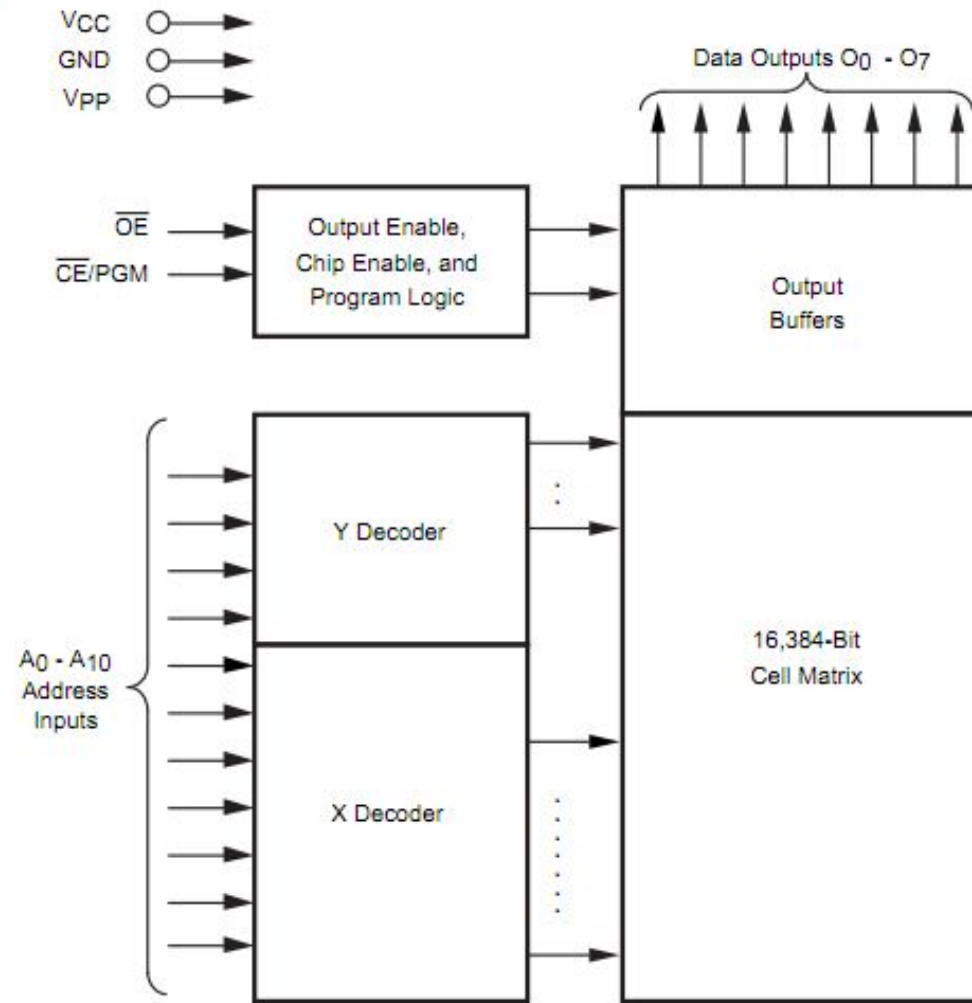


# Ejemplo de organización tridimensional: ROM 256x4 (con buffer tri-state en las salidas)



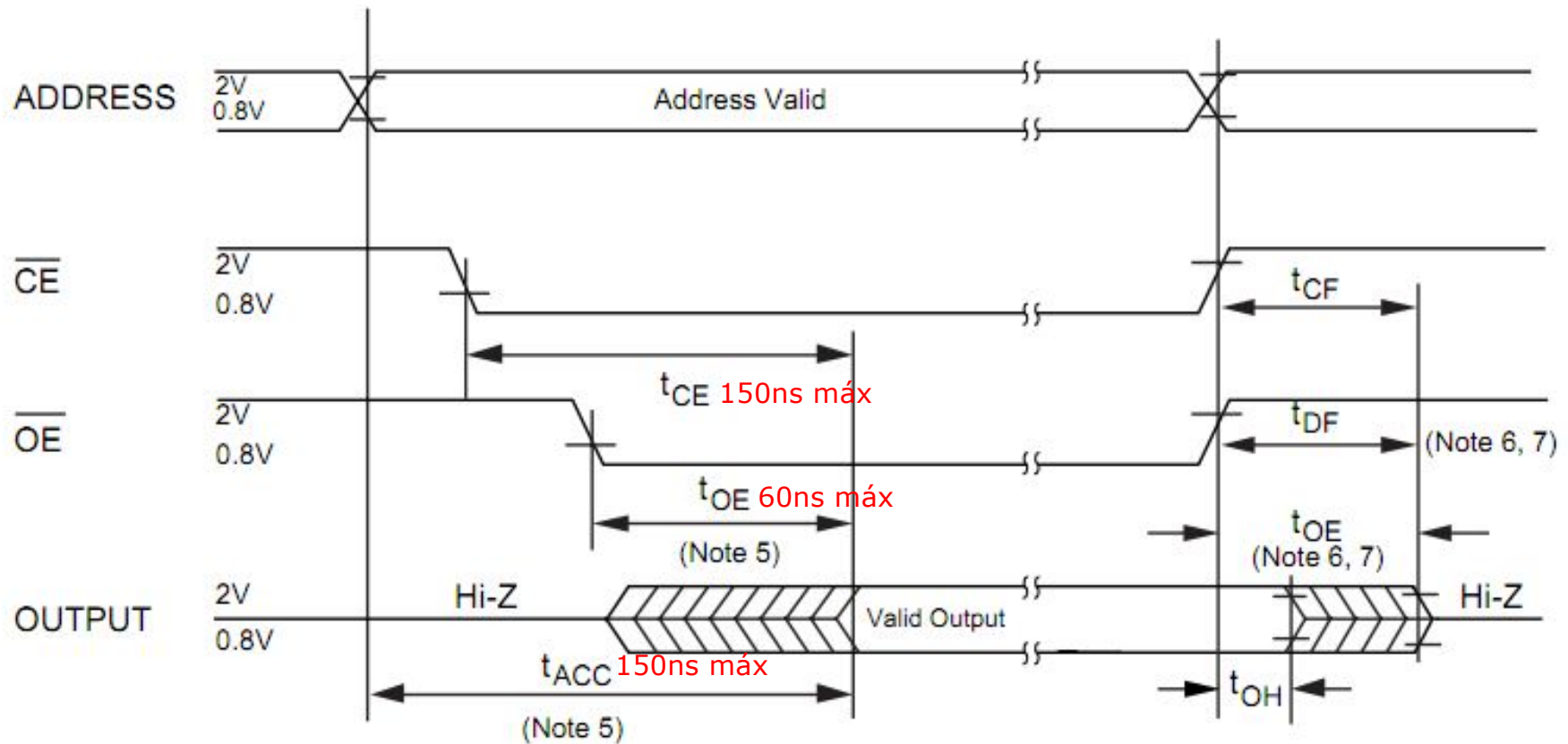
# Ejemplo de una EPROM comercial NMC27C16B 2048x8 Bit EPROM (de Fairchild)

Block Diagram



# Tiempos de acceso de la EPROM comercial NMC27C16B 2048x8 Bit EPROM (de Fairchild)

## AC Waveforms (Note 2) (Note 9)



# Tiempos de acceso de la EPROM comercial NMC27C16B 2048x8 Bit EPROM (de Fairchild)

Symbol	Parameter	Conditions	NMC27C16B				Units
			Q150		Q200, QE200		
			Min	Max	Min	Max	
$t_{ACC}$	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$		150		200	ns
$t_{CE}$	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$		150		200	ns
$t_{OE}$	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$		60		60	ns
$t_{DF}$	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$	0	50	0	60	ns
$t_{CF}$	$\overline{CE}$ High to Output Float	$\overline{OE} = V_{IL}$	0	50	0	60	ns
$t_{OH}$	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ , Whichever Occurred First	$\overline{OE} = \overline{CE} = V_{IL}$	0		0		ns

# Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

1.1 Conceptos básicos

1.2 Clasificación de memorias semiconductoras.

1.3 Memorias de Acceso aleatorio

- Memorias de sólo lectura (ROM).

- *Memorias de lectura y escritura (RAM)*

- Expansión de memorias.

1.4 Memorias de Acceso Secuencial

1.5 Otros tipos especiales de memorias.

## 2. Dispositivos programables: Arquitectura básica de una FPGA.



# Memorias RAM

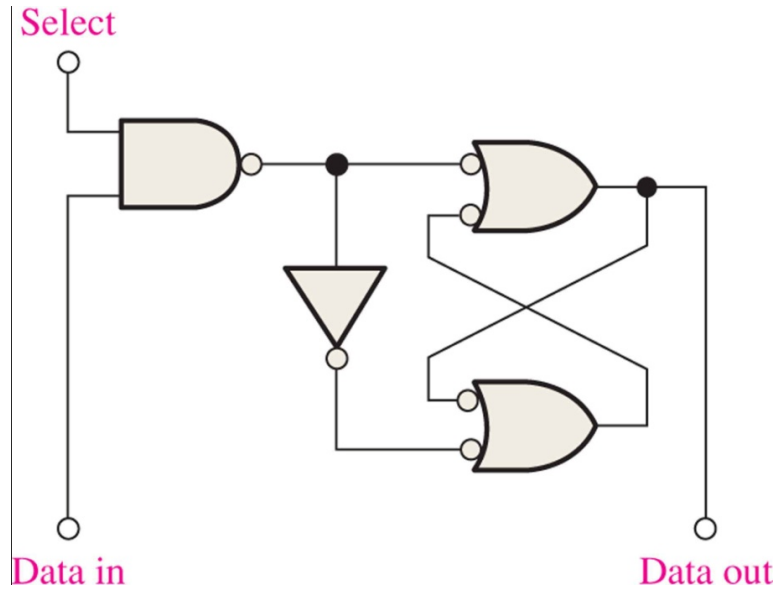
---

## ROM

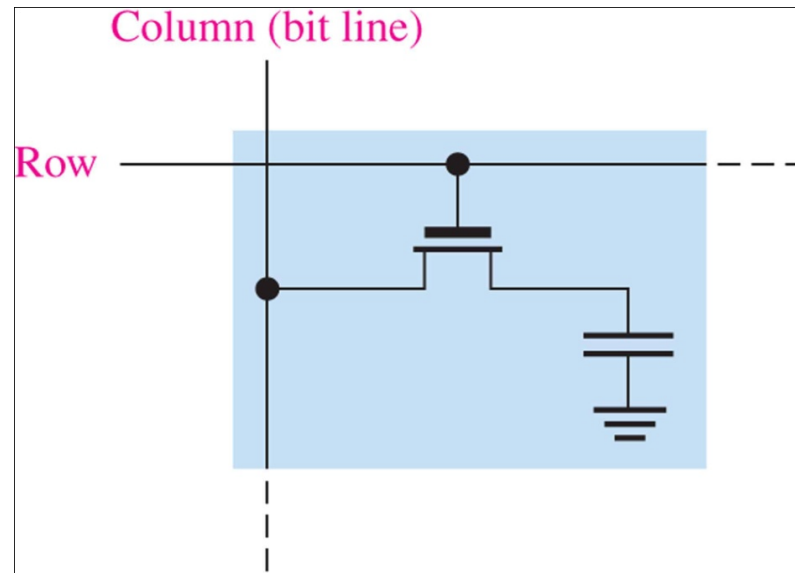
(Read Only Memory: sólo lectura, no volátiles)

- **RAM** (Random Access Memory):
  - Lectura/escritura, volátiles
  - Acceso aleatorio
- 2 tipos principales:
  - **Estáticas (SRAM)** (*Static* RAM): mantienen el dato almacenado mientras haya alimentación (con biestables) (más rápida).
  - **Dinámicas DRAM** (*Dynamic* RAM): requieren el “refrescar” periódicamente la información (“recargar” condensadores)(más económica).

# Celdas RAM



Celda SRAM

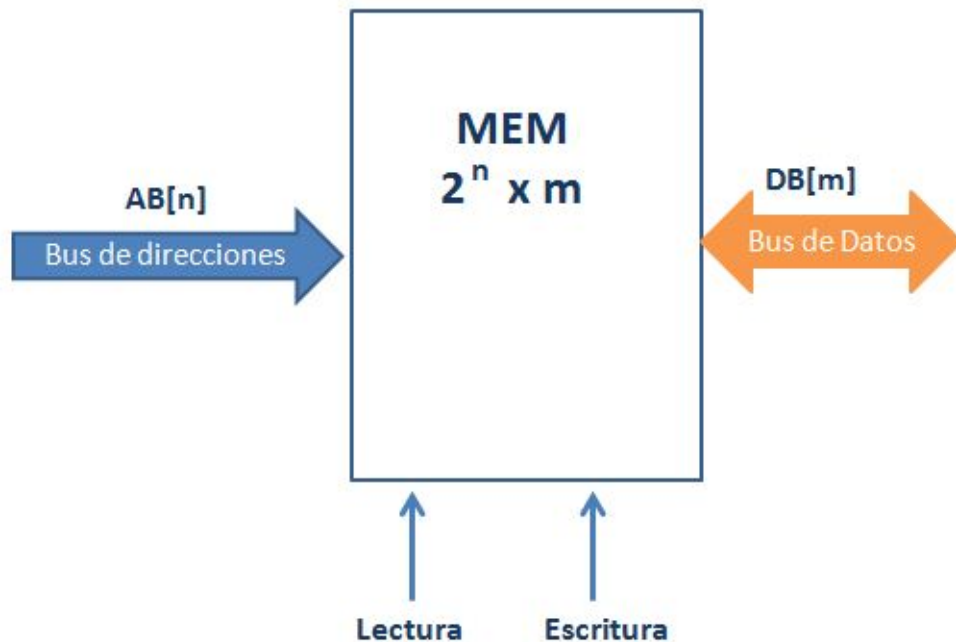


Celda DRAM

# Descripción de memoria RAM

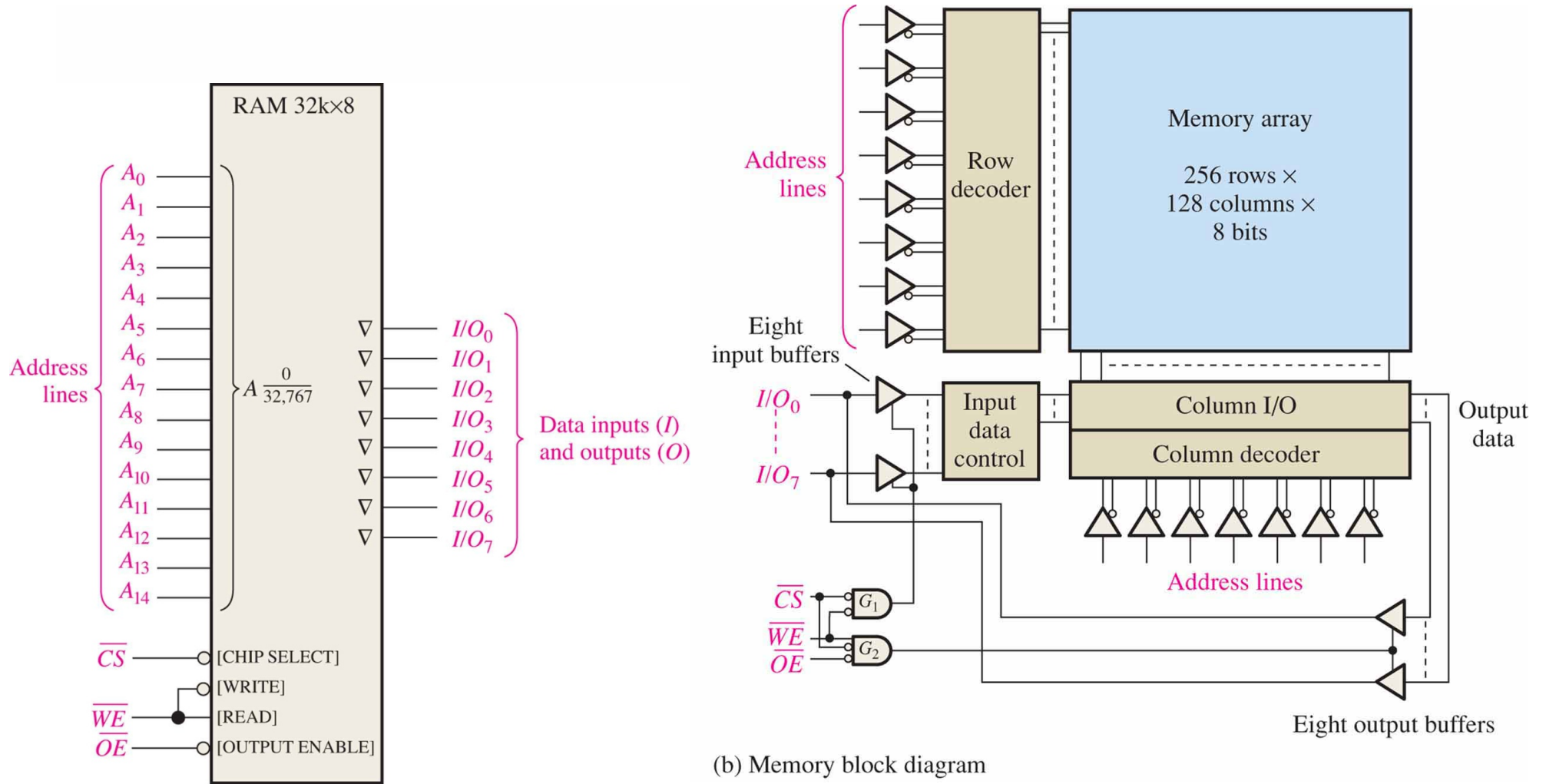
Puede tener las líneas de datos **unidireccionales** (entradas y salidas separadas) o **bidireccionales**.

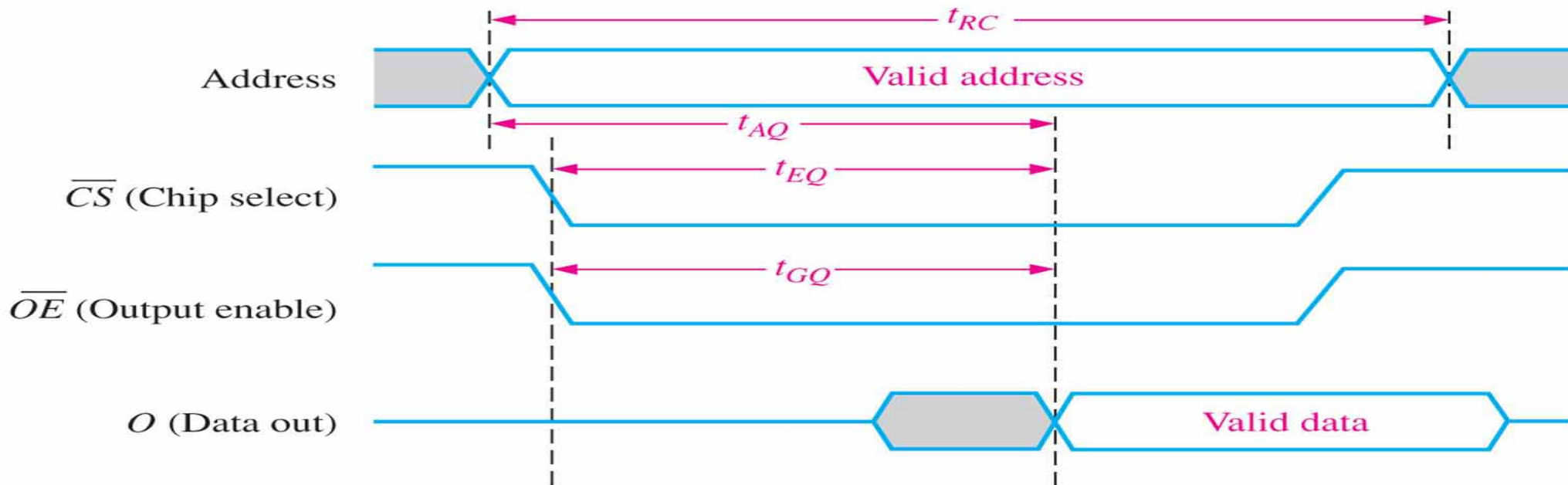
Descripción de RAM  $2^n \times m$  con líneas de datos bidireccionales:



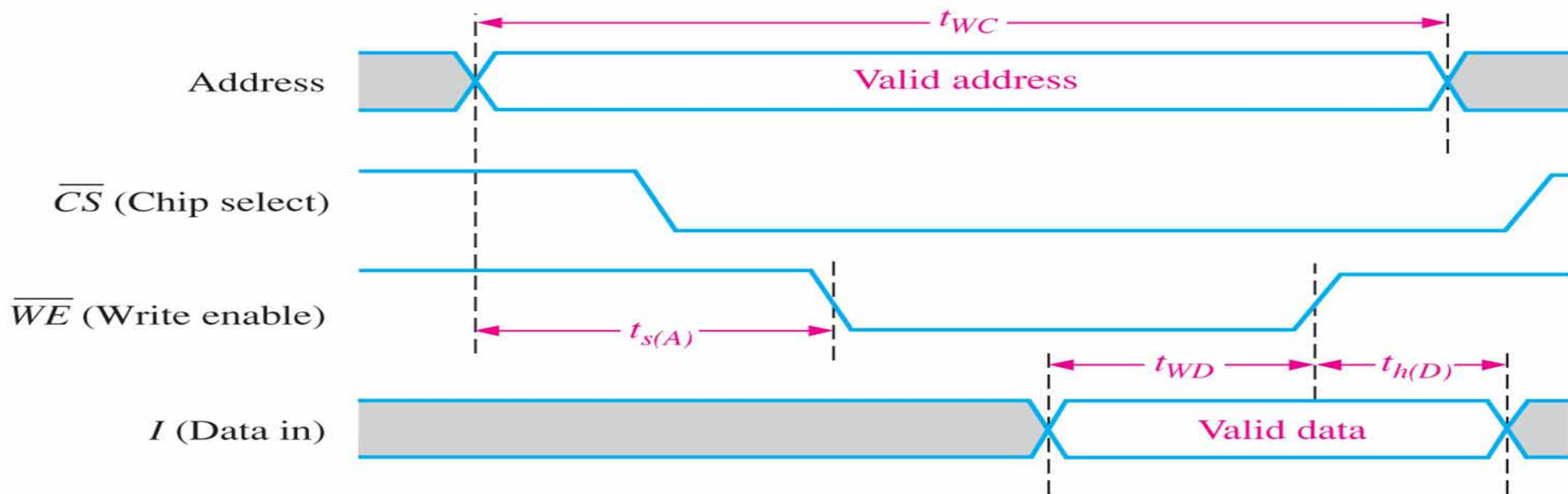
R	W	M ←	D =
0	0	M ← M	HI
0	1	M(A) ← D	[D in]
1	0	M ← M	D = M(A)
1	1	Prohibido	

# Organización interna RAM (32kx8)





(a) Read cycle ( $\overline{WE}$  HIGH)



(b) Write cycle ( $\overline{WE}$  LOW)

# Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

1.1 Conceptos básicos

1.2 Clasificación de memorias semiconductoras.

1.3 Memorias de Acceso aleatorio

- Memorias de sólo lectura (ROM).

- Memorias de lectura y escritura (RAM)

- *Expansión de memorias (Asociación).*

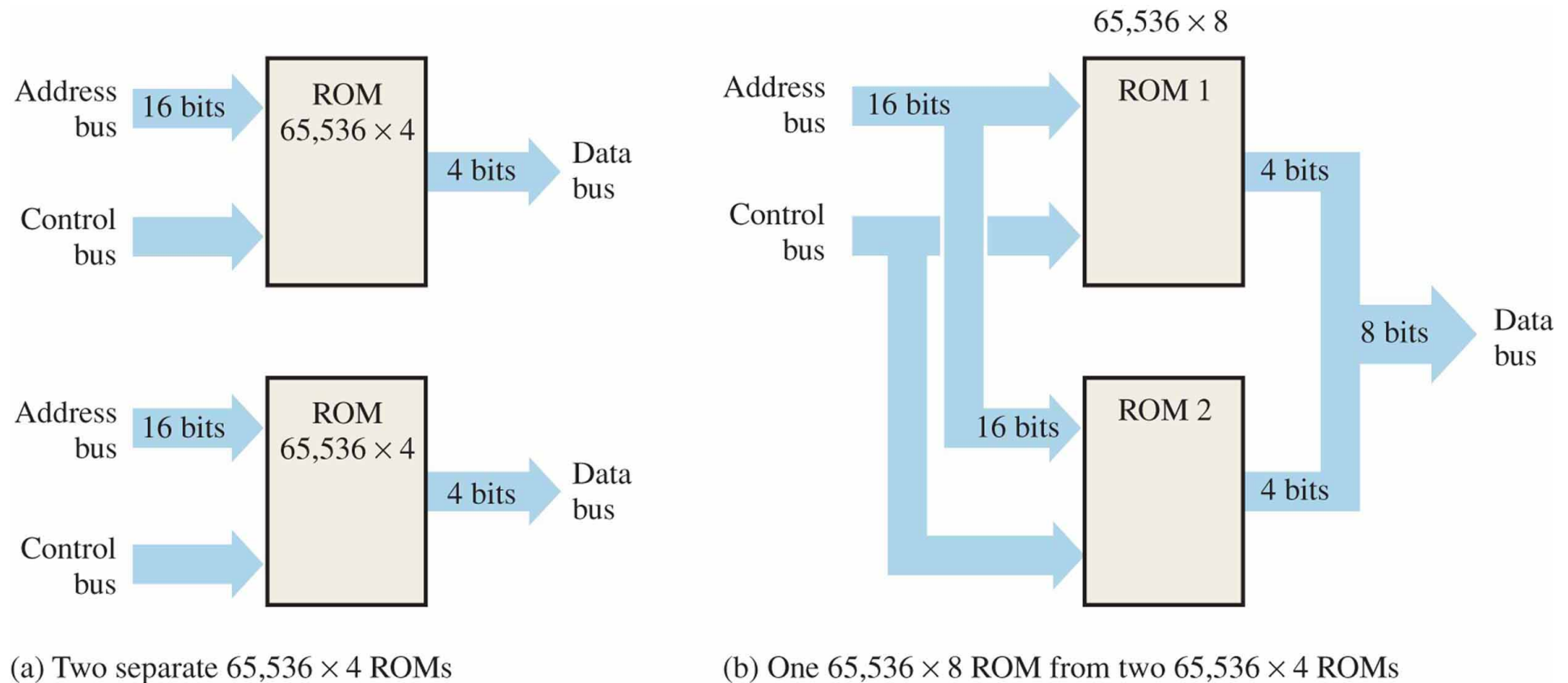
1.4 Memorias de Acceso Secuencial

1.5 Otros tipos especiales de memorias.

## 2. Dispositivos programables: Arquitectura básica de una FPGA.

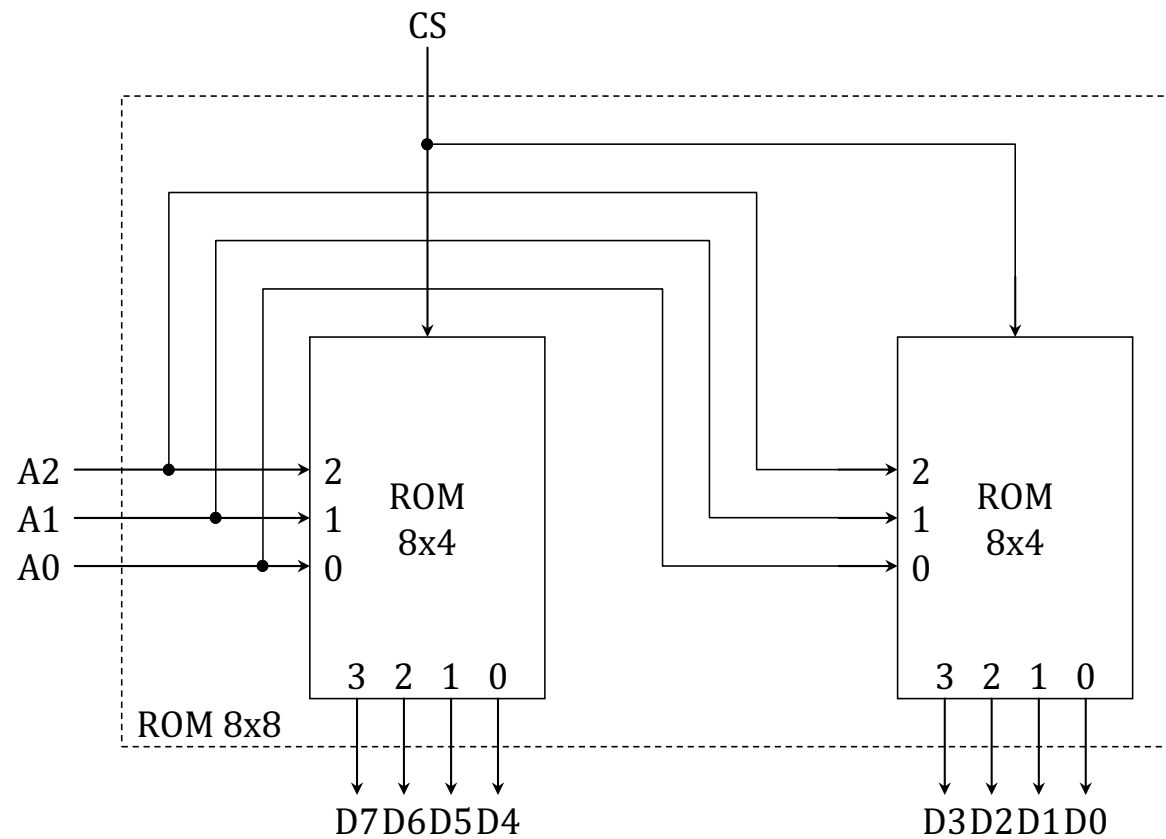
# Expansión del ancho de palabra en memorias ROM

(Cómo conseguir más bits en cada posición de memoria)



# Expansión del "ancho de palabra" en memorias ROM

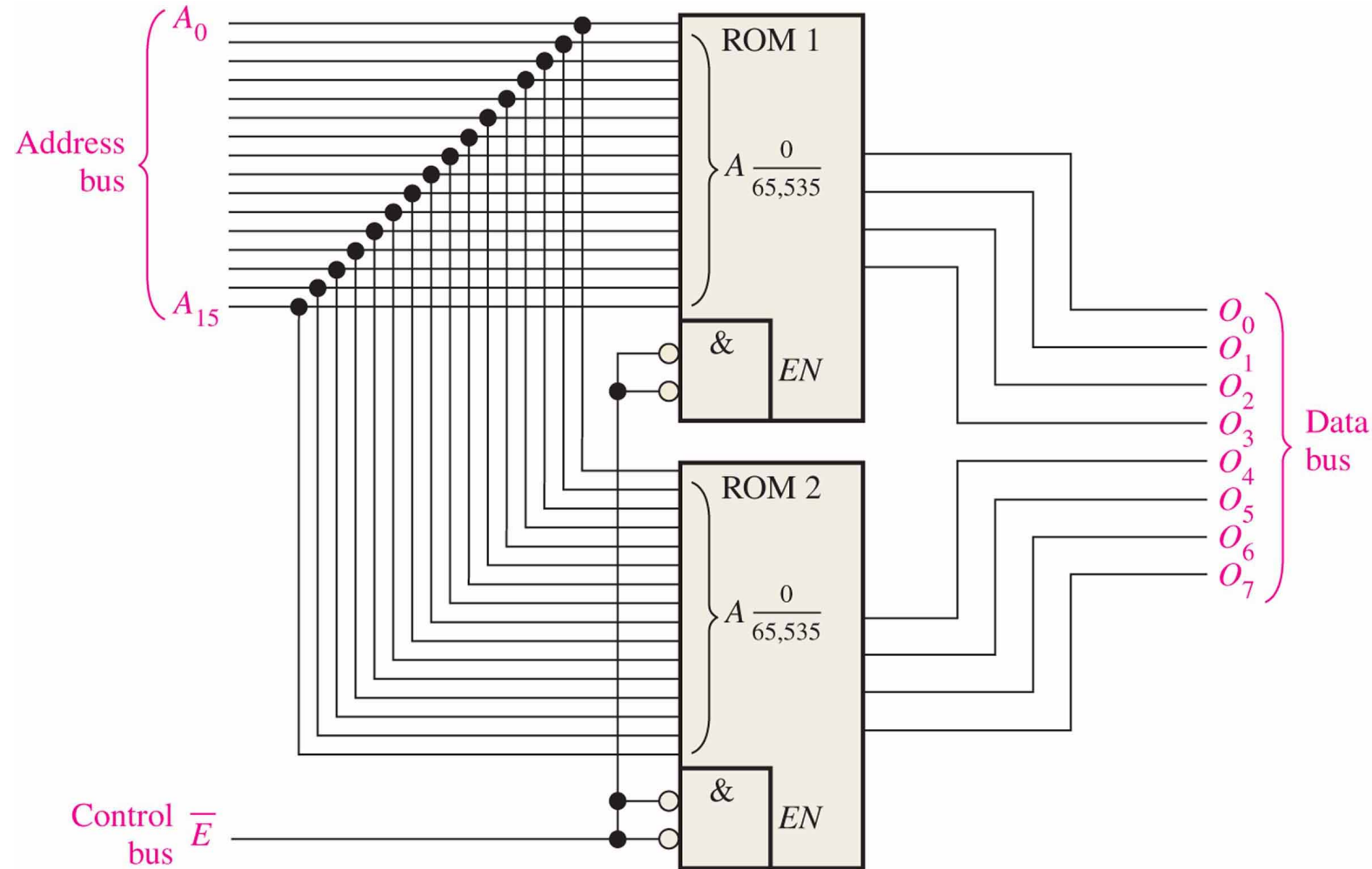
Realizar una ROM  $2^3 \times 8$  con dos ROM  $2^3 \times 4$



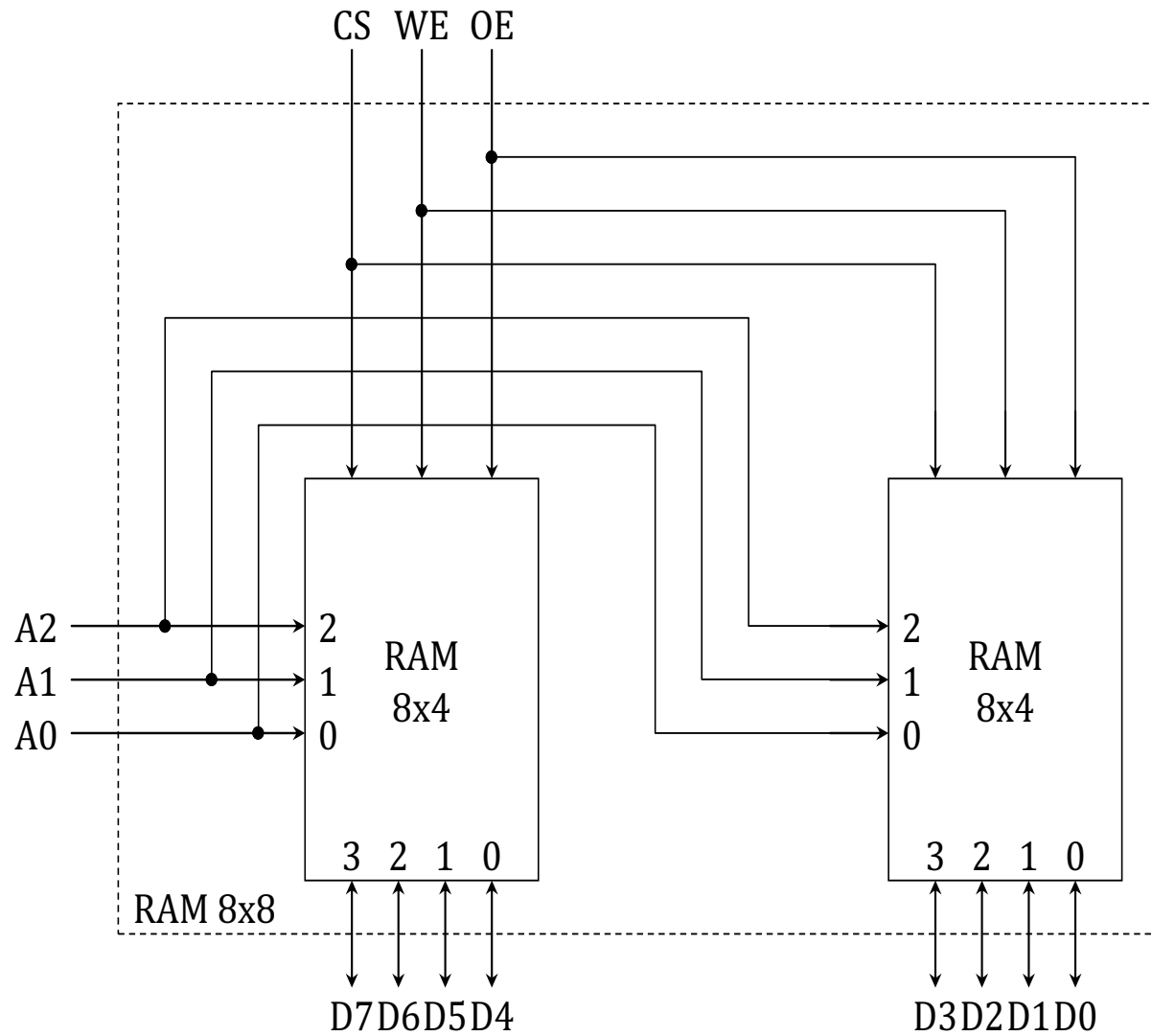


# Expansión del "ancho de palabra" en memorias ROM

Ej. Realizar una ROM 64kx8 con dos ROM 64kx4

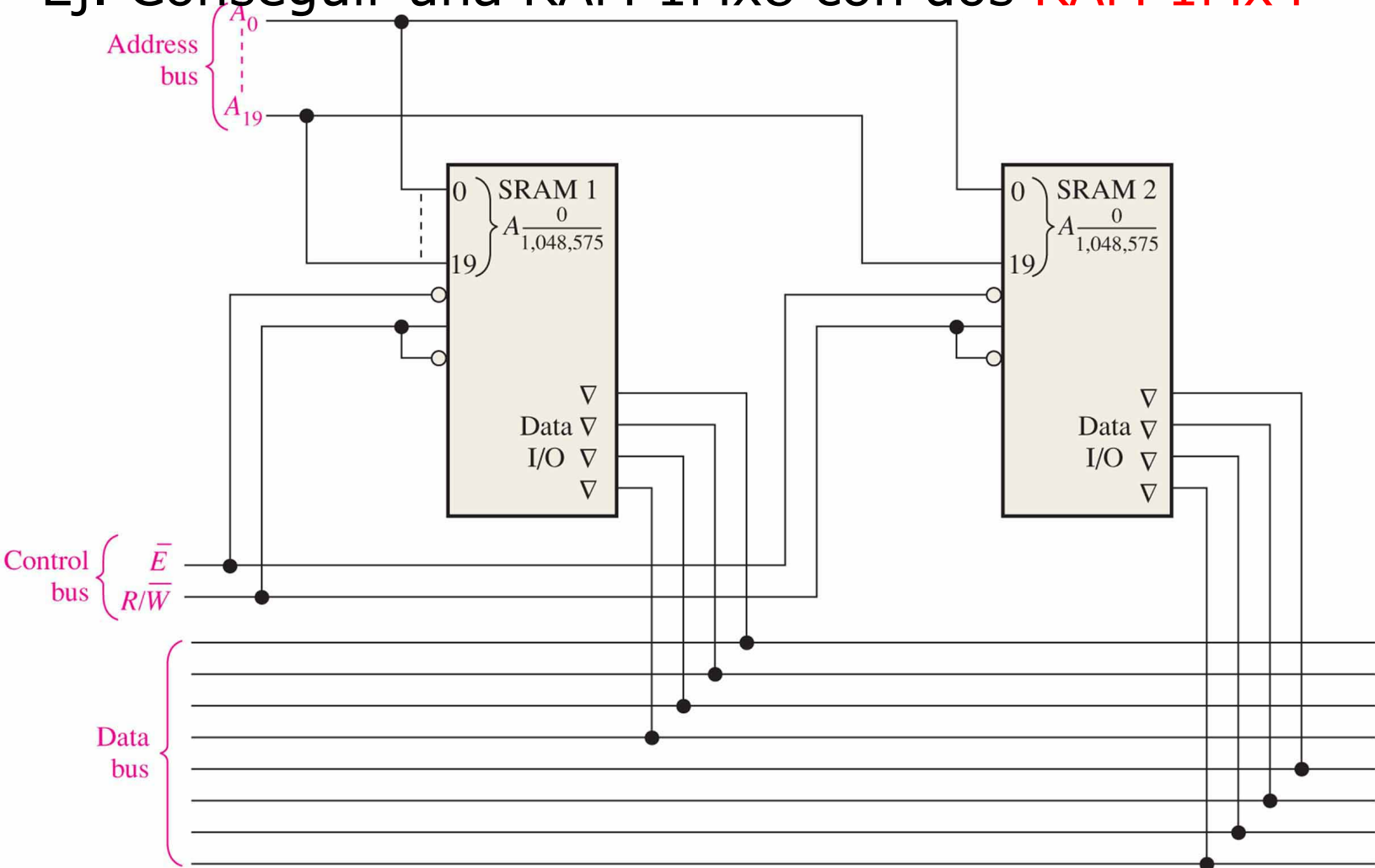


# Expansión de longitud de palabra en memorias RAM

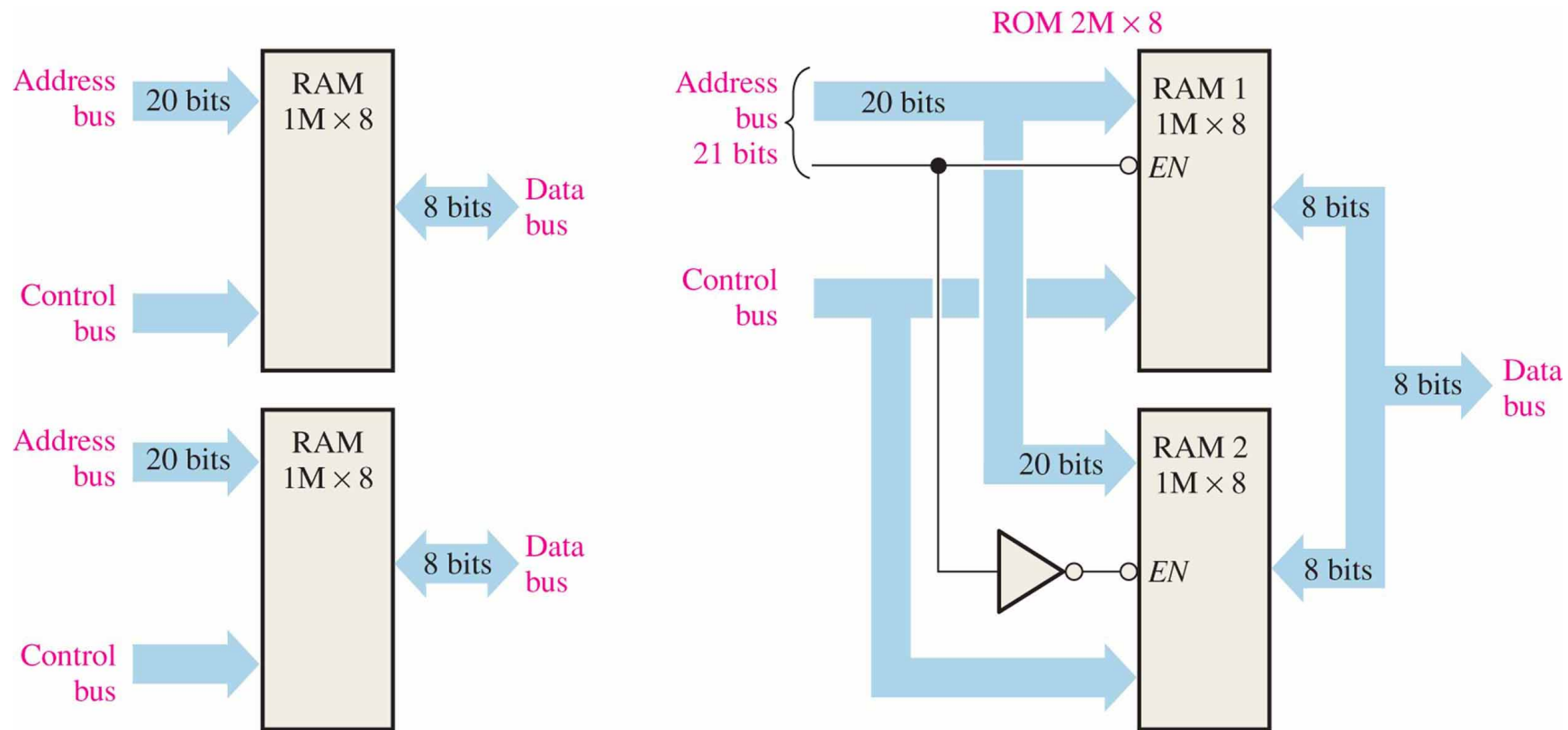


# Expansión del ancho de palabra en memorias RAM

Ej. Conseguir una RAM 1Mx8 con dos RAM 1Mx4



# Expansión de número de palabras en memorias

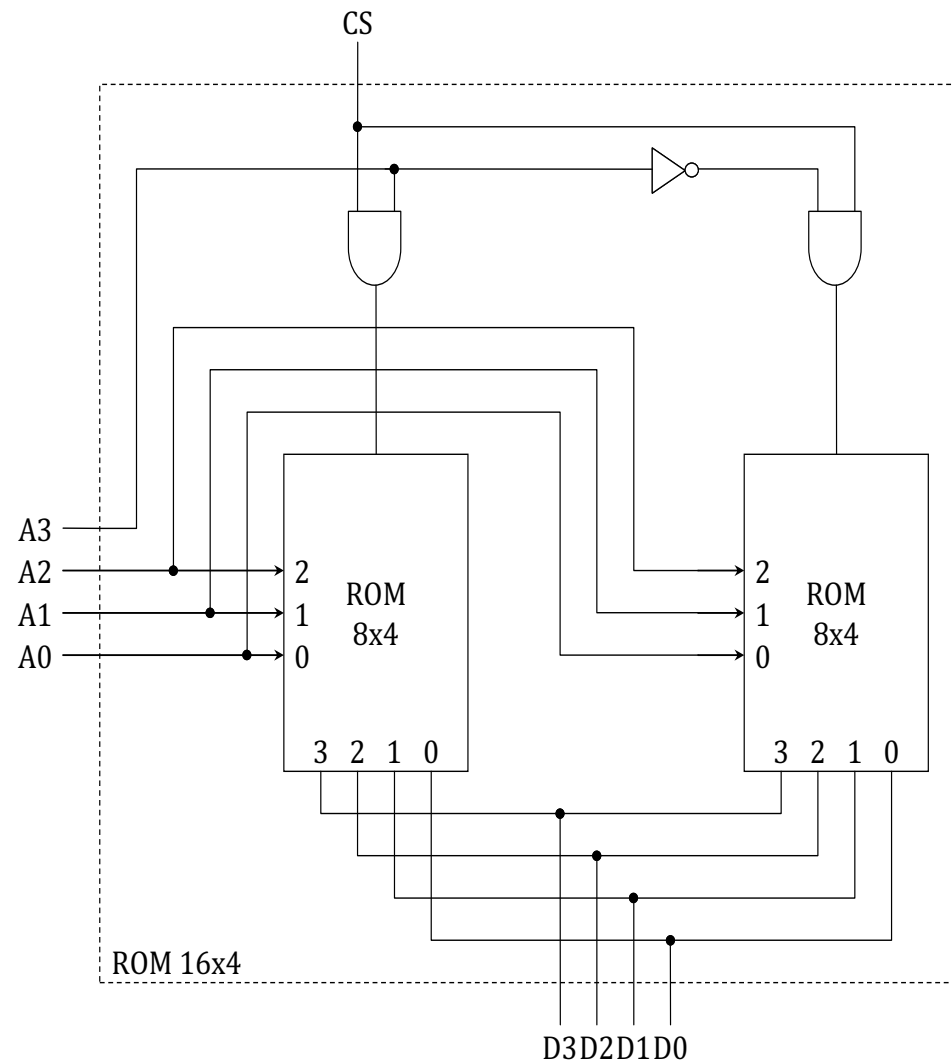


(a) Individual memories each store 1,048,576 8-bit words

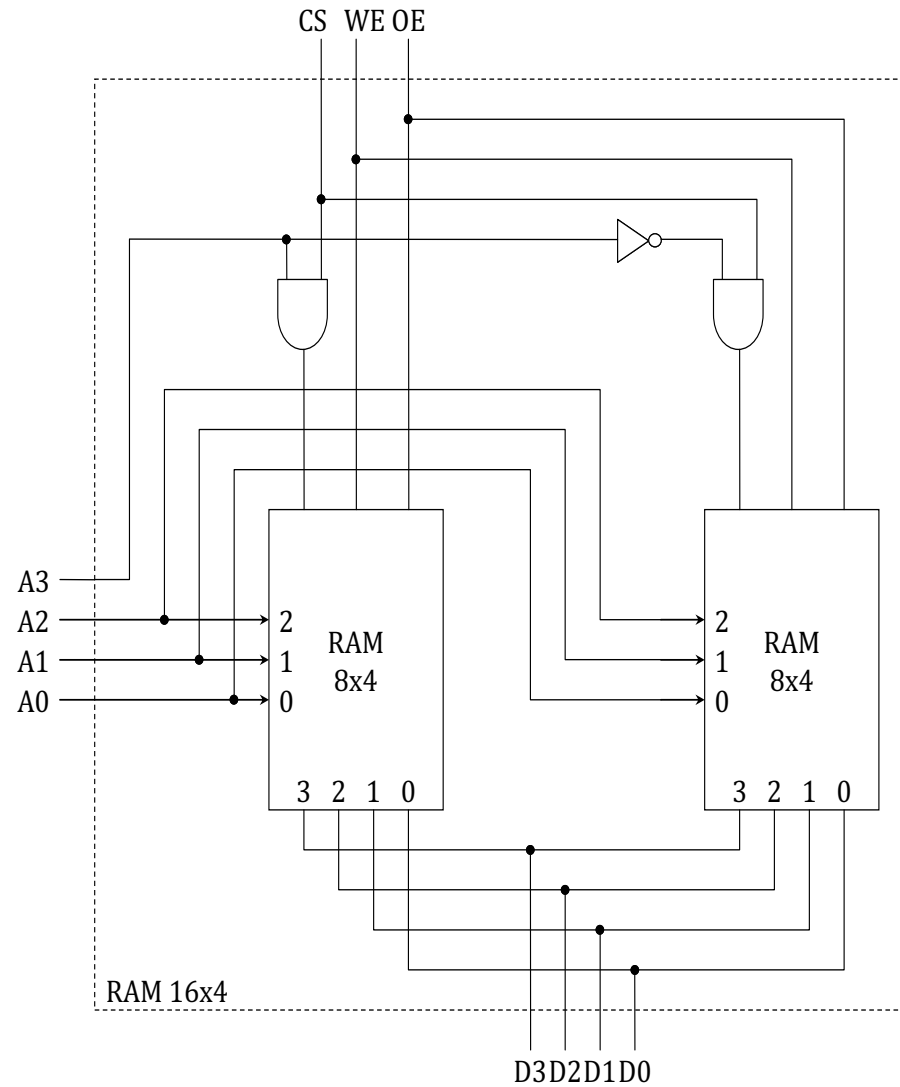
(b) Memories expanded to form a 2M x 8 RAM requiring a 21-bit address bus

# Expansión de número de palabras en memorias ROM

Conseguir una ROM  $2^4 \times 4$  con dos ROM  $2^3 \times 4$

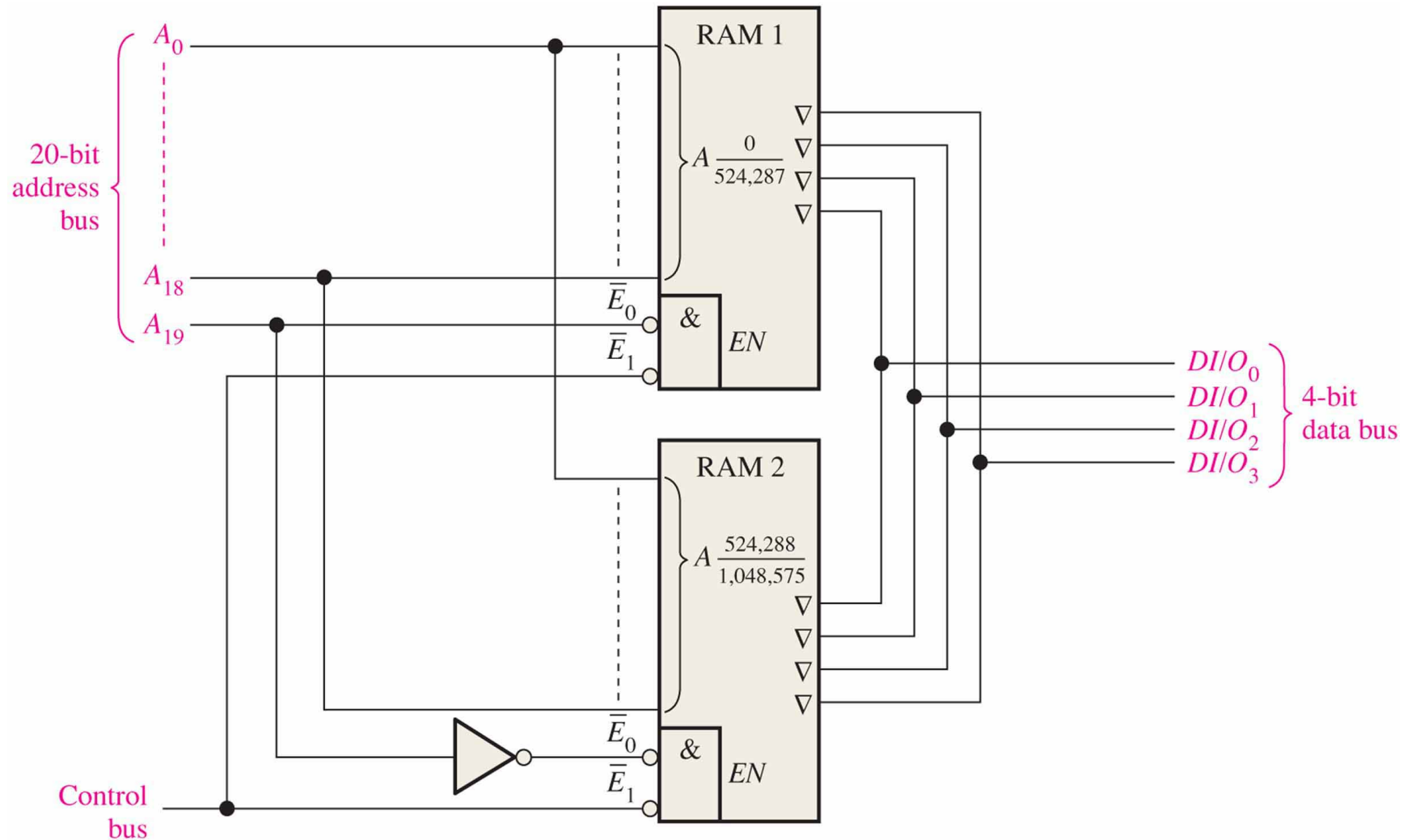


# Ejemplo: expansión de número de palabras en memorias RAM



# Ejemplo de expansión de número de palabras en memorias RAM

Ej.: Conseguir una RAM 1Mx4 con dos RAM 512K x4



# Memorias FLASH

- Son memorias de lectura/escritura, de gran capacidad, no volátiles***

TIPO	No volátil	Alta densidad	Celda de 1 transistor	Escritura "in-system"
Flash	Sí	Sí	Sí	Sí
SRAM	No	No	No	Sí
DRAM	No	Sí	Sí	Sí
ROM	Sí	Sí	Sí	No
EPROM	Sí	Sí	Sí	No
EEPROM	sí	No	No	Sí



# Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

1.1 Conceptos básicos

1.2 Clasificación de memorias semiconductoras.

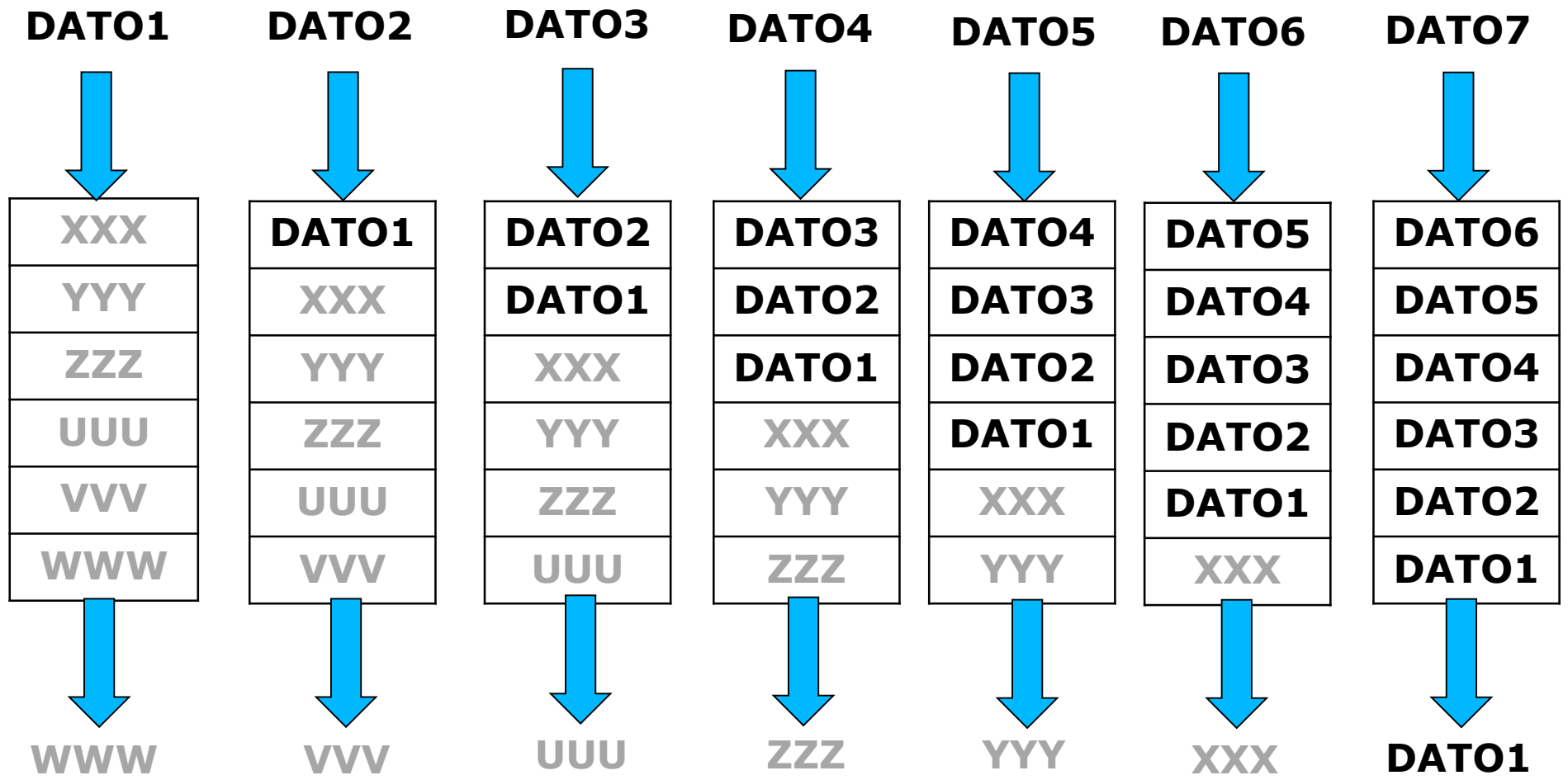
1.3 Memorias de Acceso aleatorio

- Memorias de sólo lectura (ROM).
- Memorias de lectura y escritura (RAM)
- Expansión de memorias.

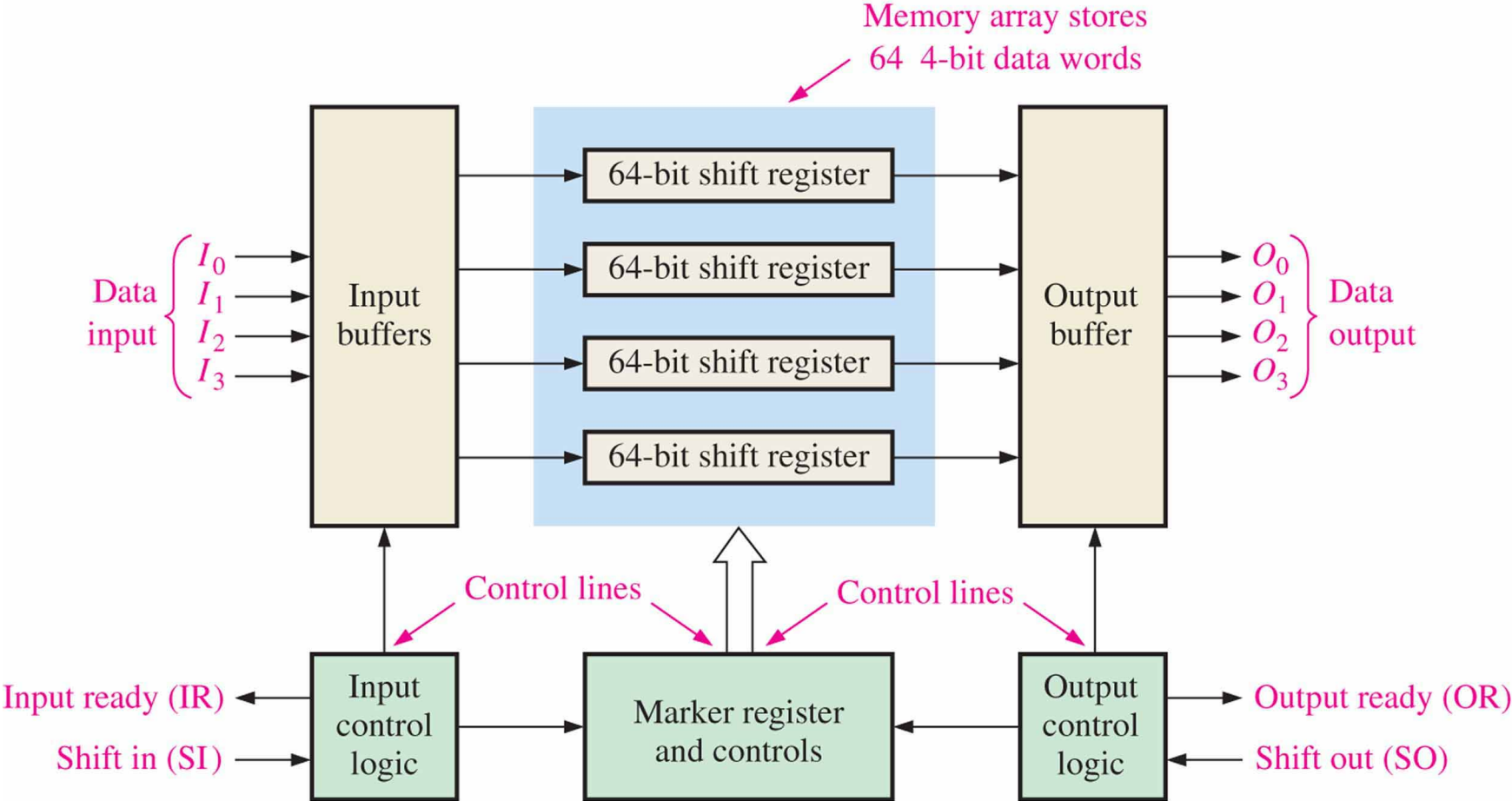
*1.4 Memorias de Acceso Secuencial*

2. Dispositivos programables: Arquitectura básica de una FPGA.

# Memorias FIFO (First In-First Out)



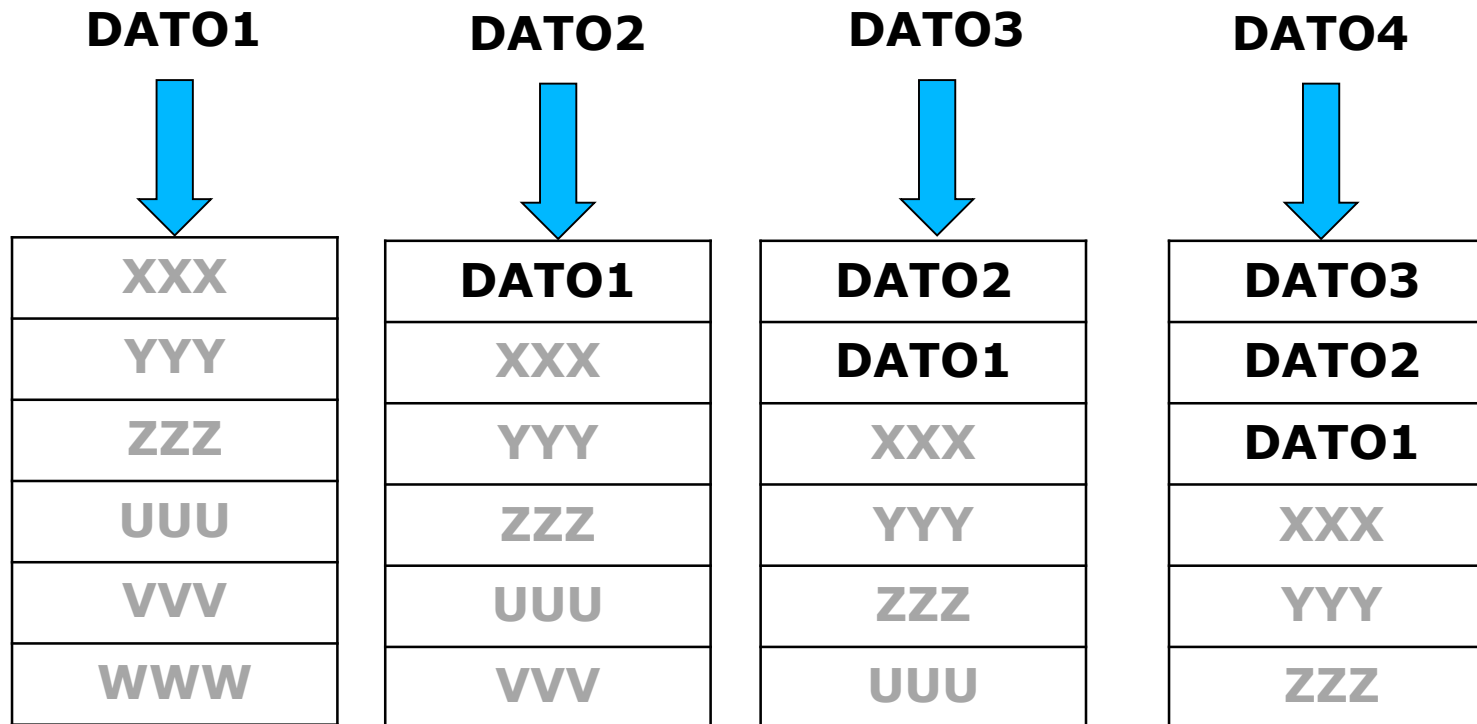
# Ejemplo: Memorias FIFO (First In-First Out 64x4)



# Memorias LIFO (Last In-First Out)

---

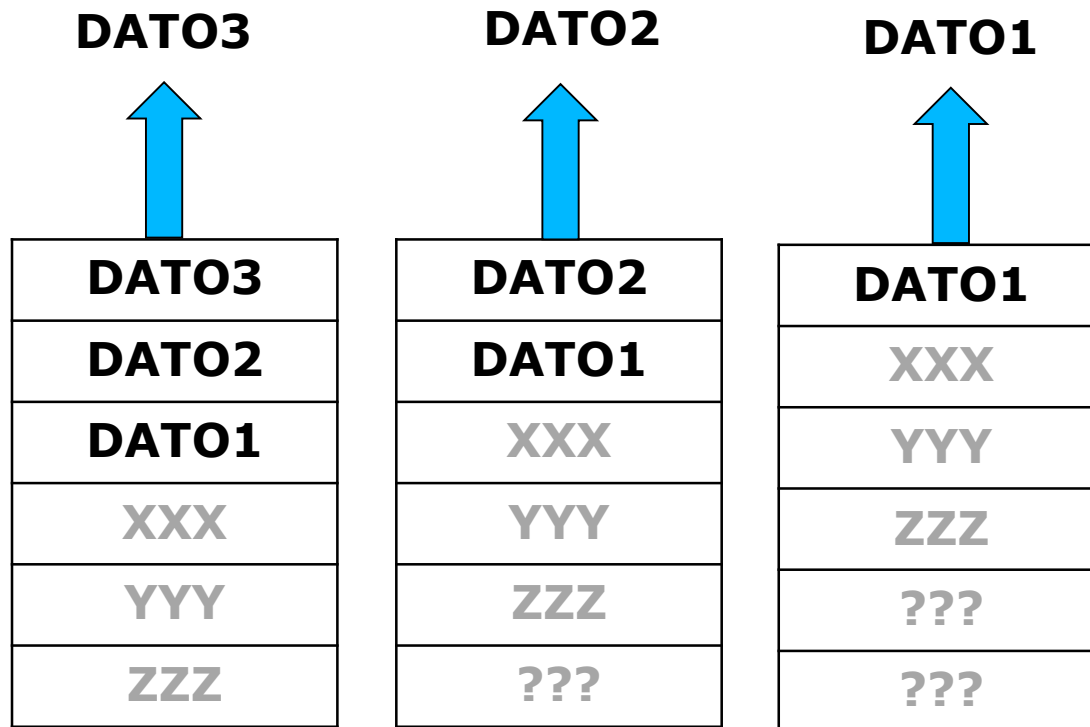
Concepto de pila: (“guardar” datos de la pila)



# Memorias LIFO (Last In-First Out)

---

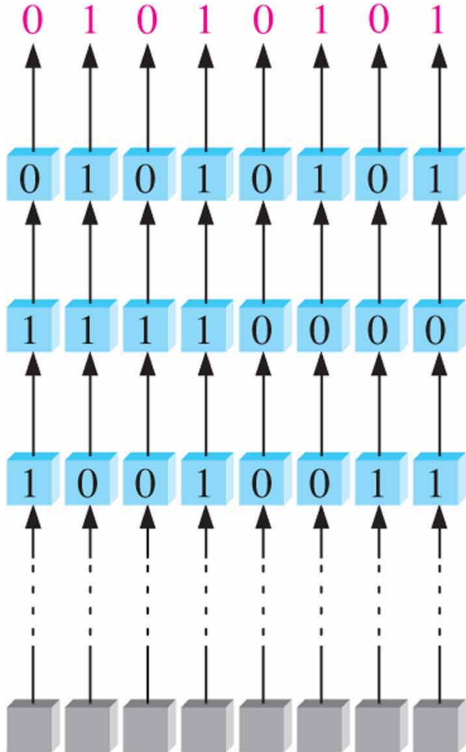
Concepto de pila: ("sacar" datos de la pila)



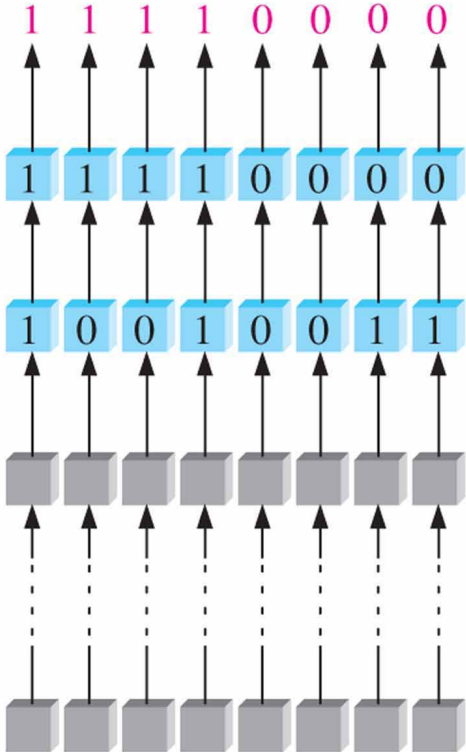


# Leyendo de la Pila

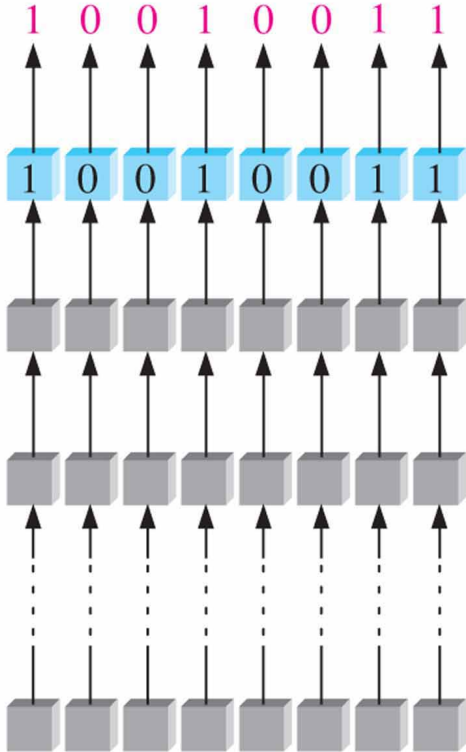
Initially storing 3 data bytes.  
The last byte in is at top-of-stack.



After third byte is pulled from stack, the second byte that was stored pops up to the top-of-stack.



After second byte is pulled from stack, the first byte that was stored pops up to the top-of-stack.



# Memorias y Dispositivos Lógicos Programables

---

## 1. Memorias

1.1 Conceptos básicos

1.2 Clasificación de memorias semiconductoras.

1.3 Memorias de Acceso aleatorio

- Memorias de sólo lectura (ROM).

- Memorias de lectura y escritura (RAM)

- Expansión de memorias.

1.4 Memorias de Acceso Secuencial

## 2. *Dispositivos programables: Arquitectura básica de una FPGA.*

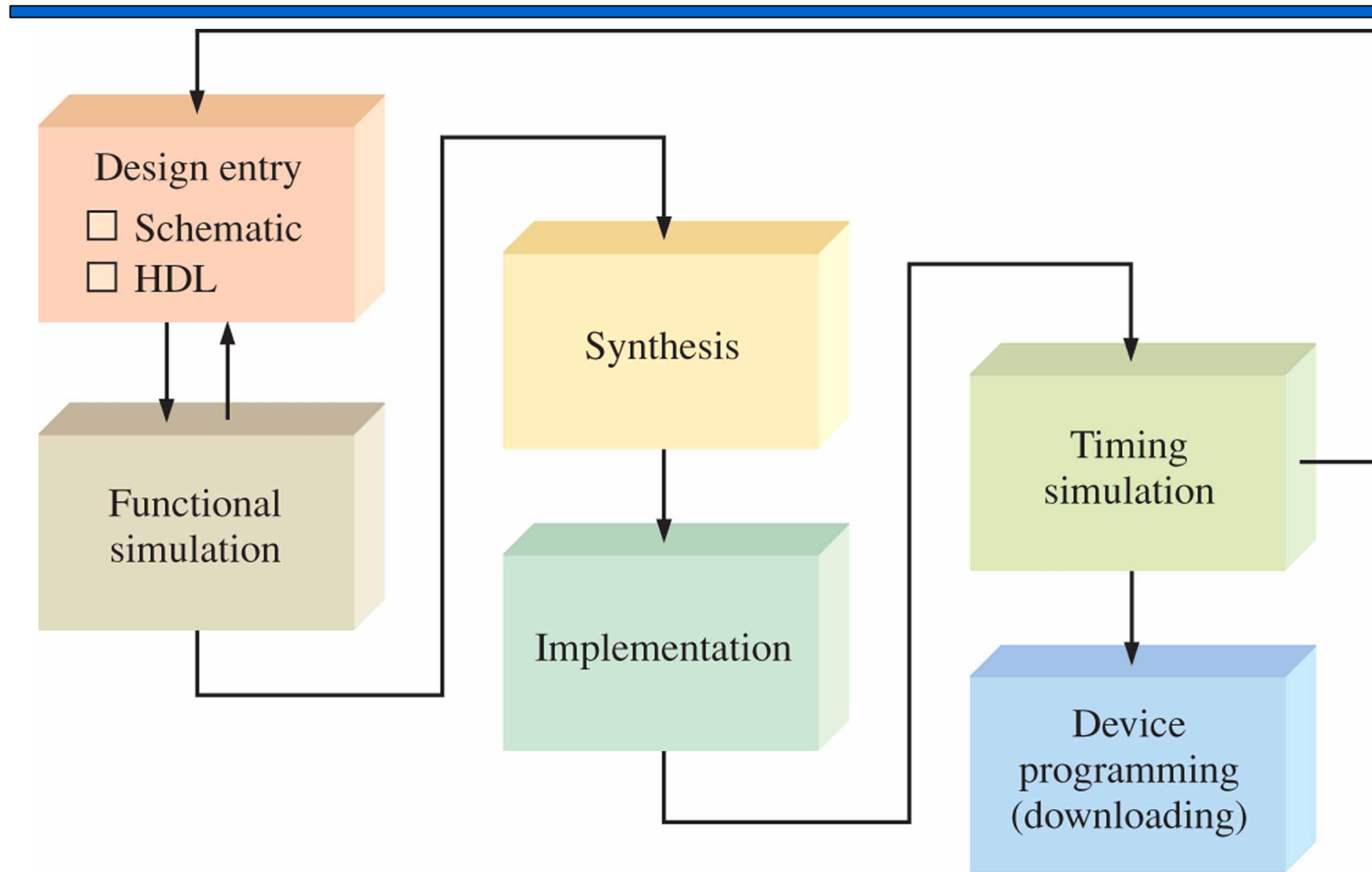


# Dispositivos Lógicos Programables

---

- Para funciones de muchas entradas y muchas salidas, el diseño con biestables y puertas (a partir de K-mapa y tablas de estado) es inviable.
- PLDs: Son dispositivos programables por el usuario, para implementar “muchas funciones” de “muchas variables”.
- Gran variedad:
  - SPLD (Simplex PLD): PAL/GAL, FPLA,
  - CPLD (Complex PLD)
- Actualmente, los más usados son las **FPGA** (Xilinx y Altera)

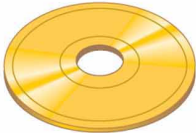
# Ciclo de vida del diseño con PLD



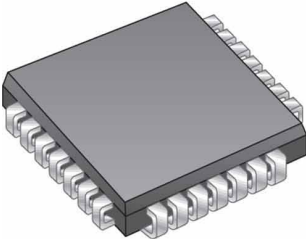
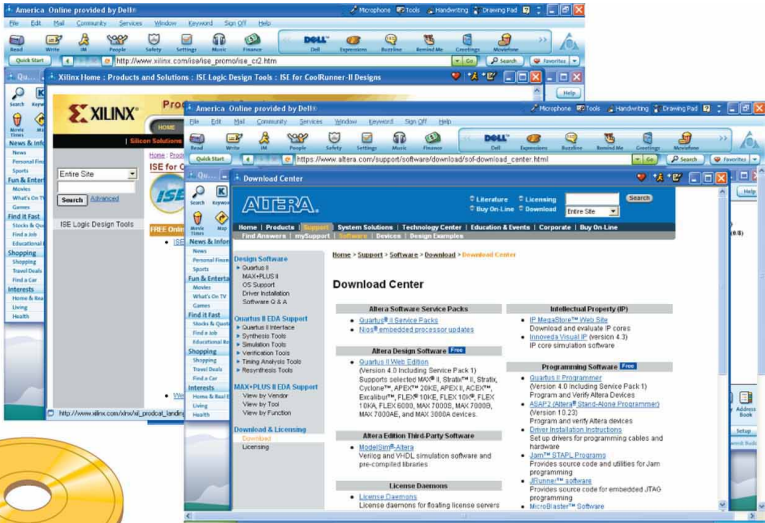
# Entorno de diseño con PLD



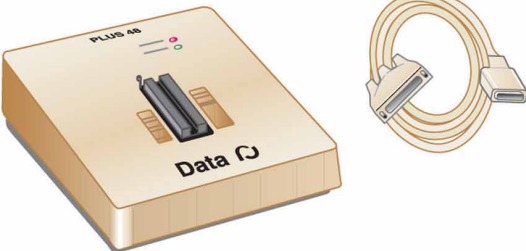
(a) Computer



(b) Software (CD or Website download)

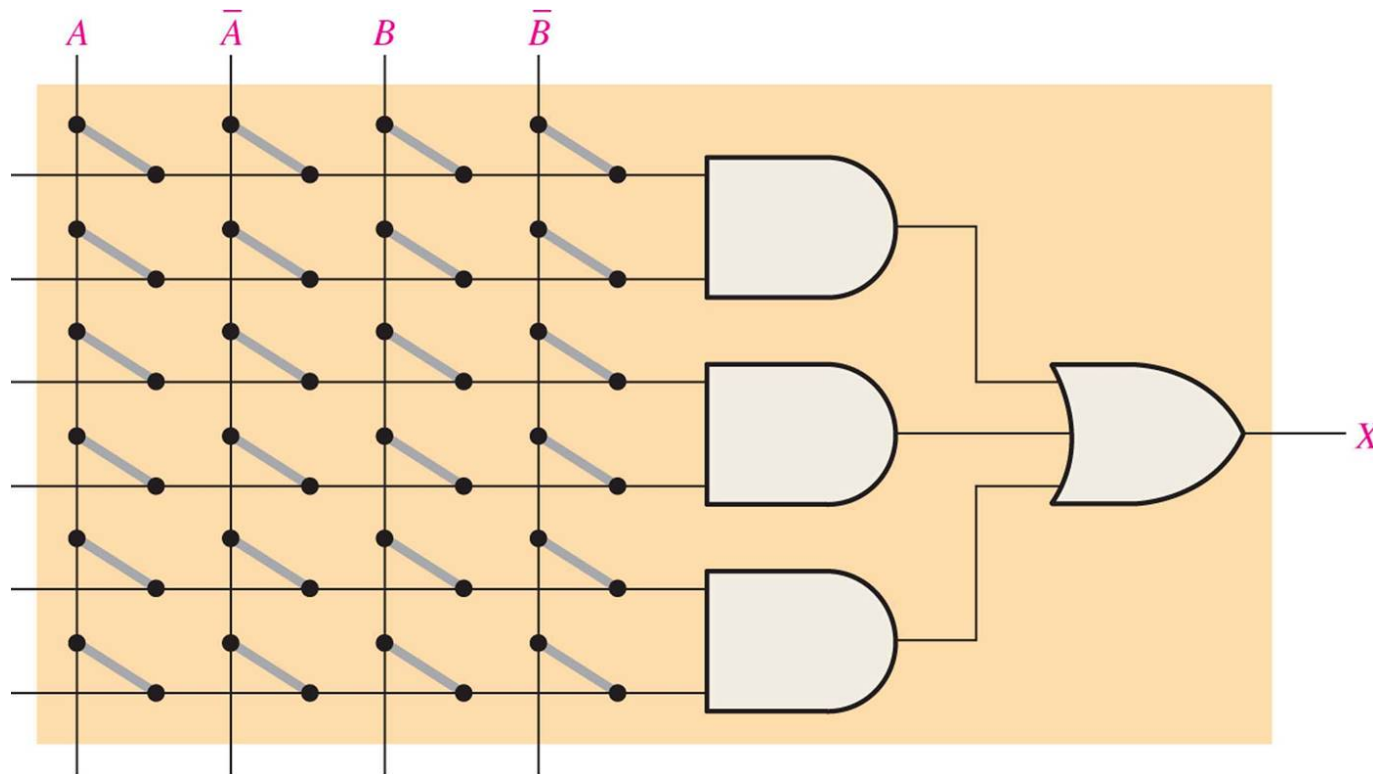


(c) Device



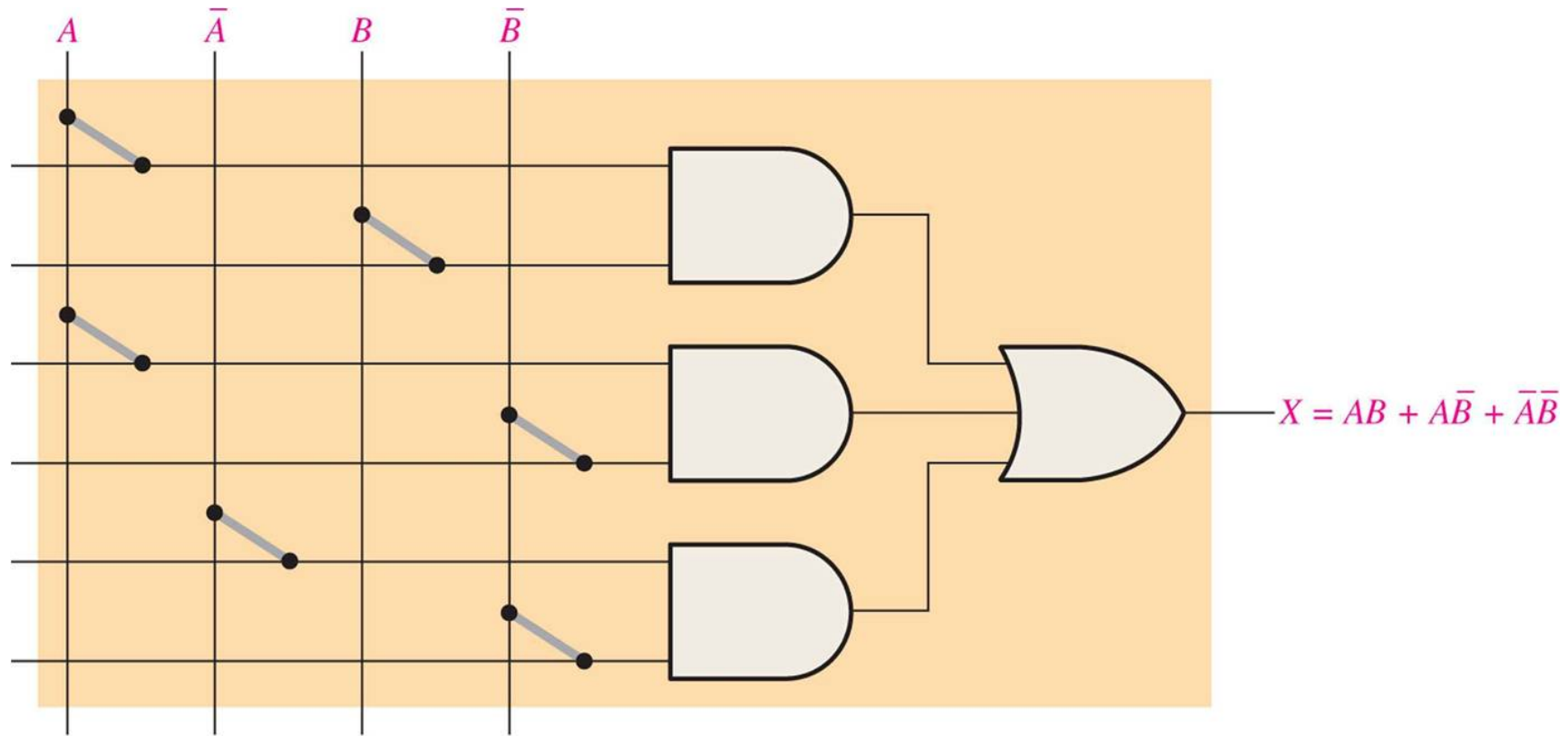
(d) Programming hardware (programming fixture or development board with cable for connection to computer port)

# Estructura básica de una PAL (SPLD)



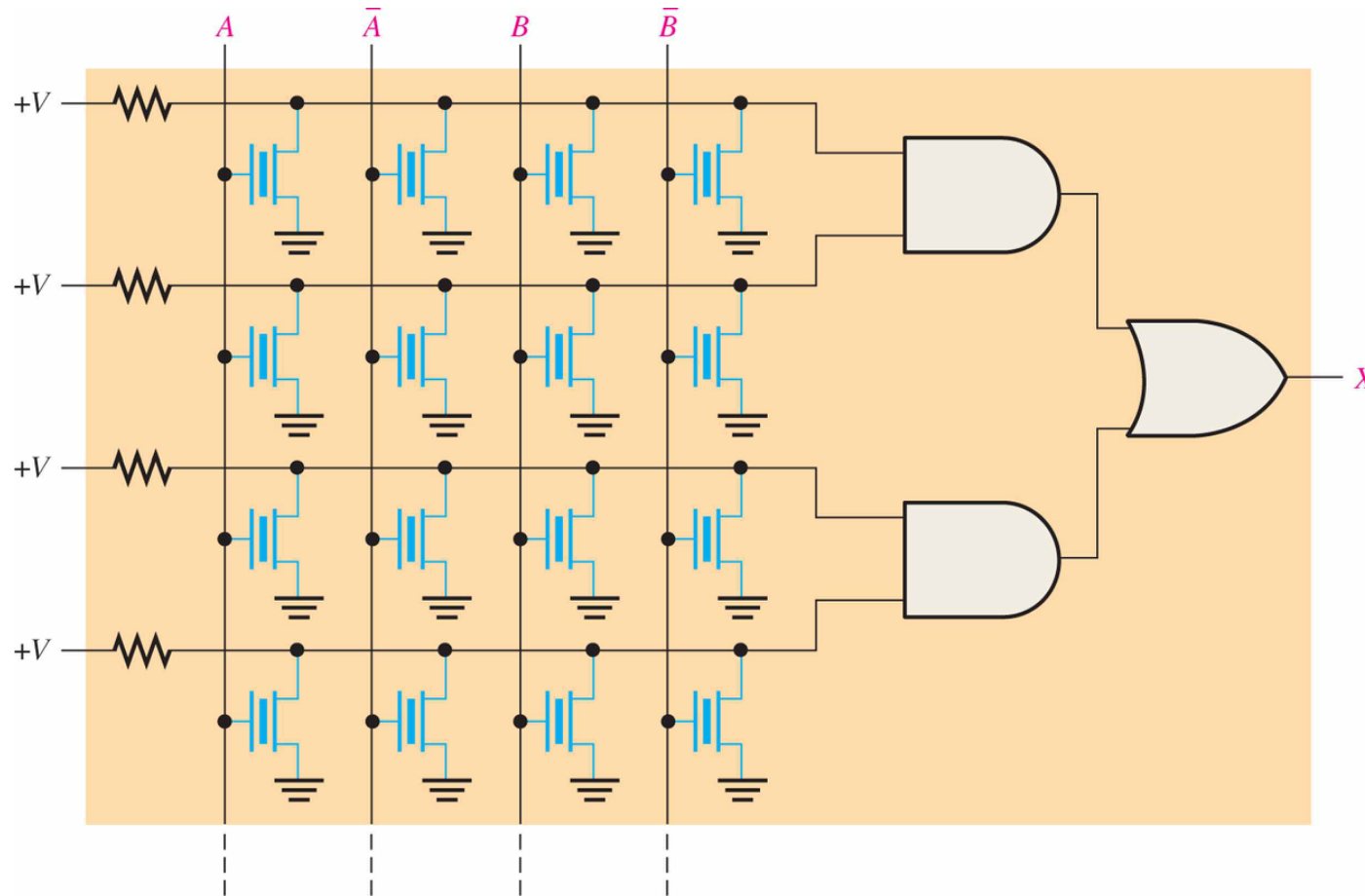
**(lógica de fusibles NO reprogramable)**

# Suma de productos realizada con una PAL (SPLD)



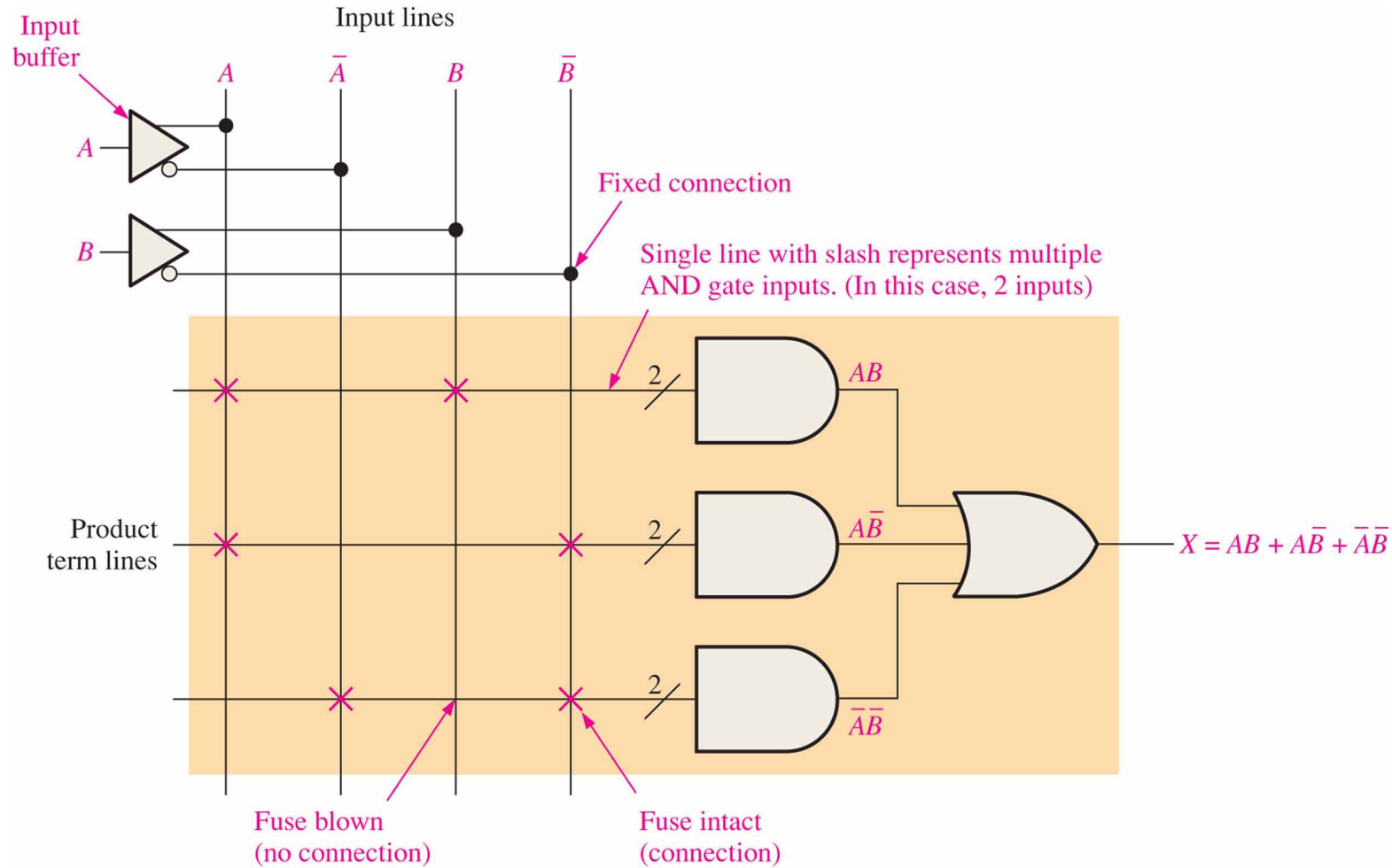
**(lógica de fusibles NO reprogramable)**

# Estructura básica de una GAL (SPLD)

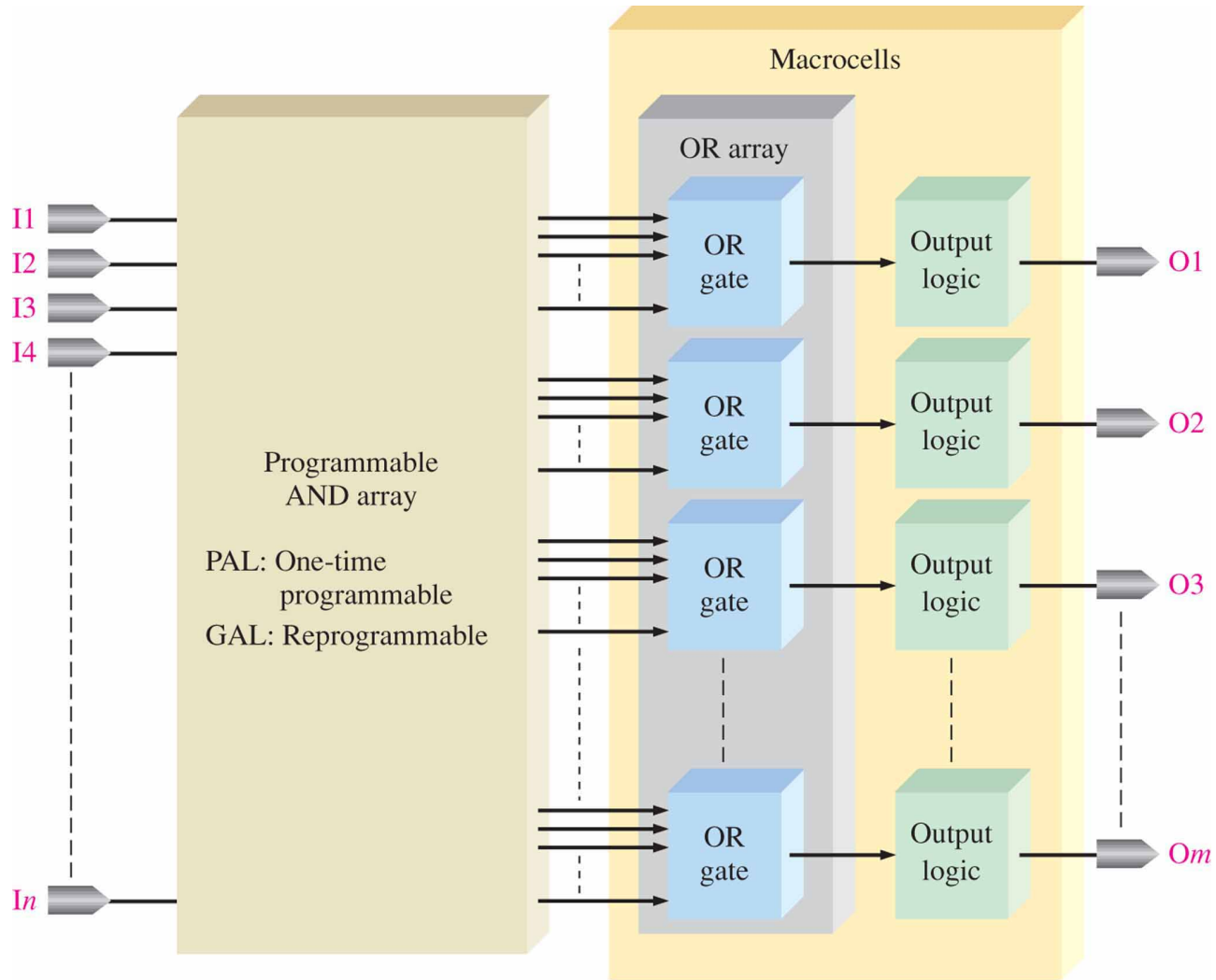


**(SÍ es reprogramable)**

# Notación simplificada PAL/GAL (SPLDs)

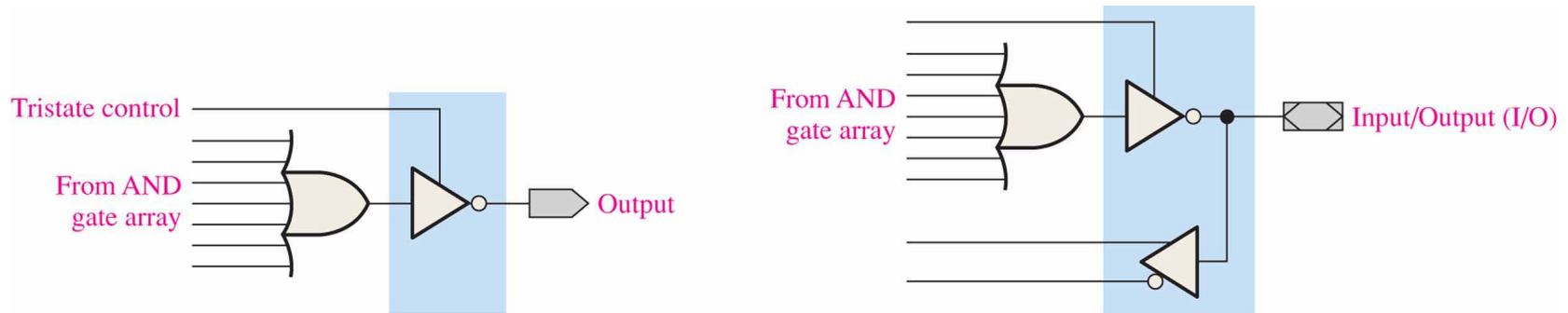


# Arquitectura de una PAL/GAL (SPLDs)



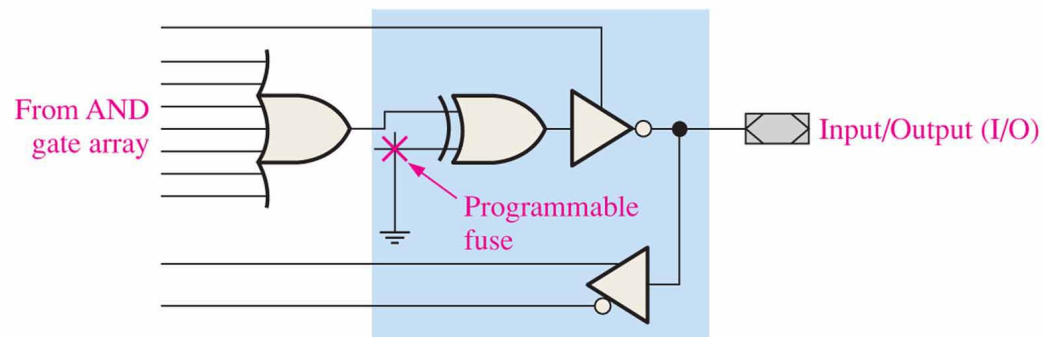


# Lógica de salida de las macroceldas en los dispositivos PAL/GAL (SPLDs)



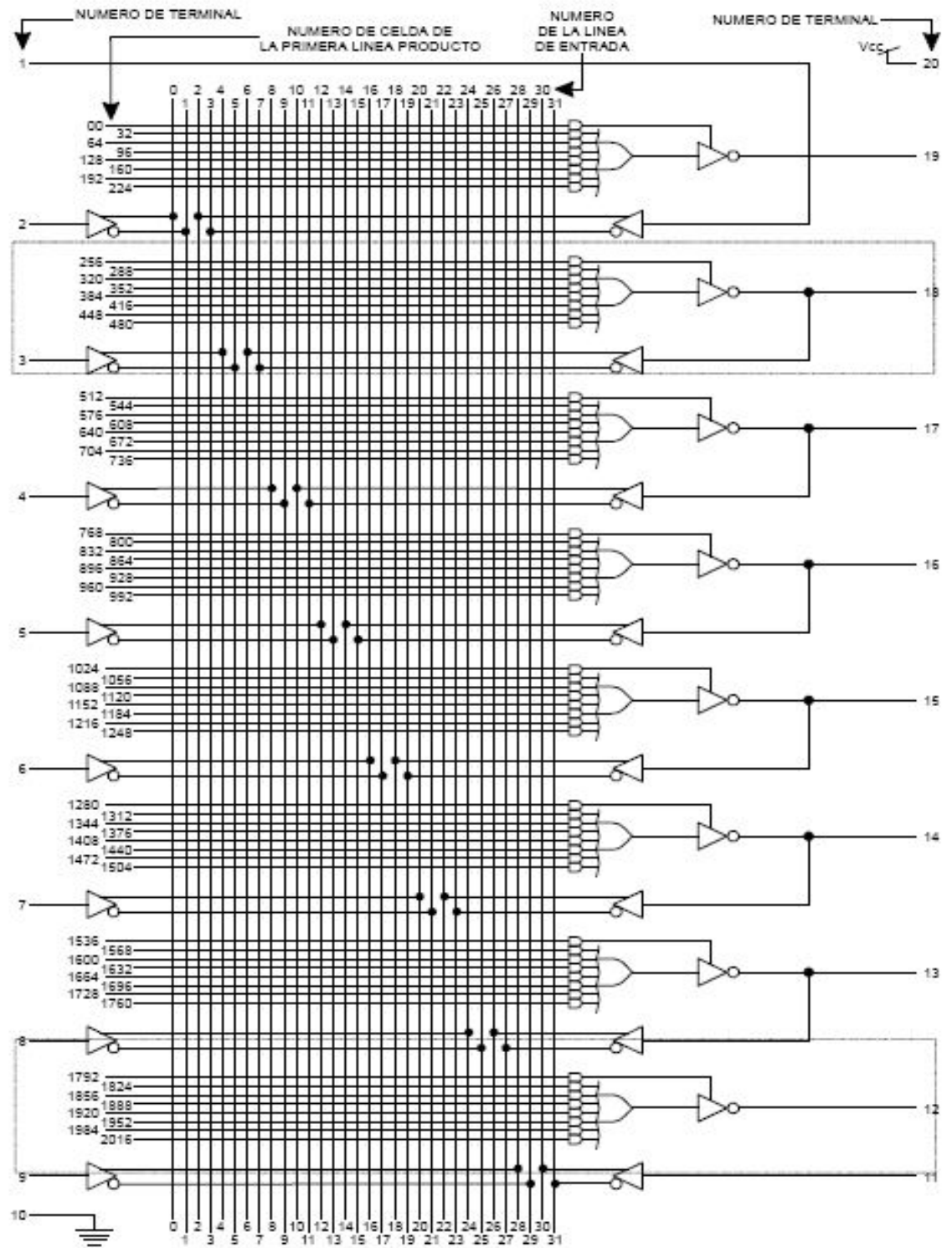
(a) Combinational output (active-LOW). An active-HIGH output would be shown without the bubble on the tristate gate symbol.

(b) Combinational input/output (active-LOW)

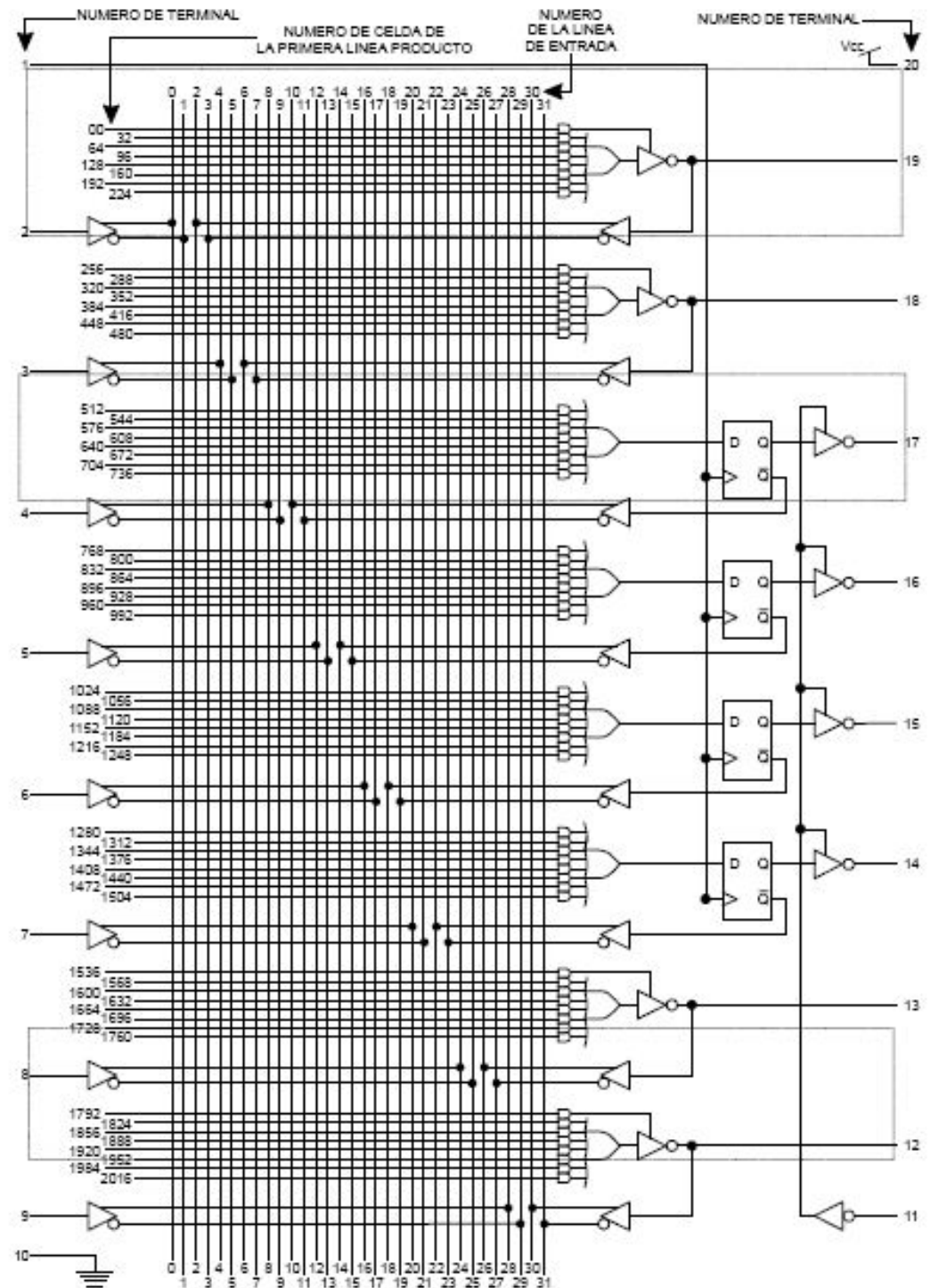


(c) Programmable polarity output

# Ejemplo: PAL16L8

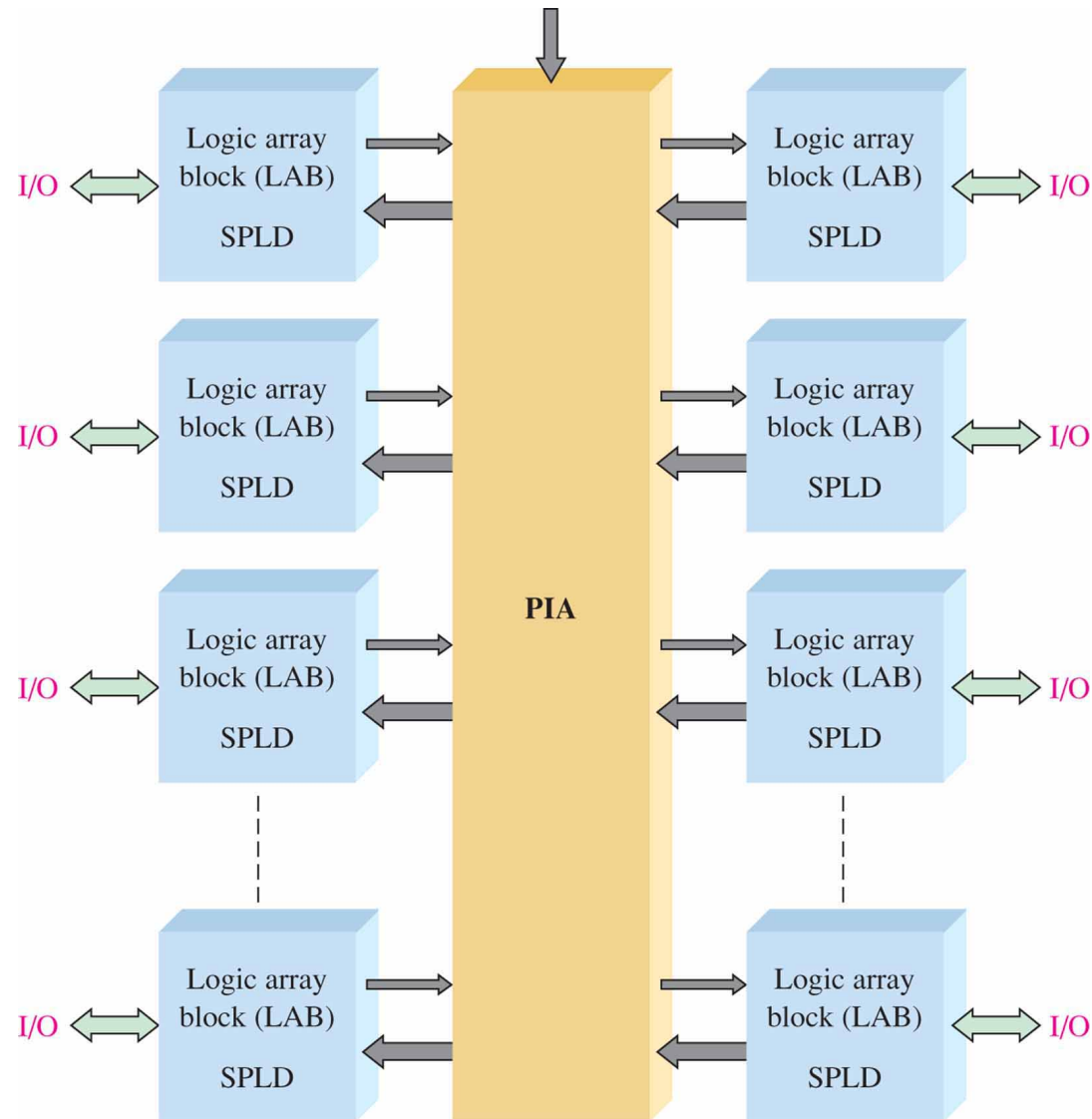


# Ejemplo: PAL16R4

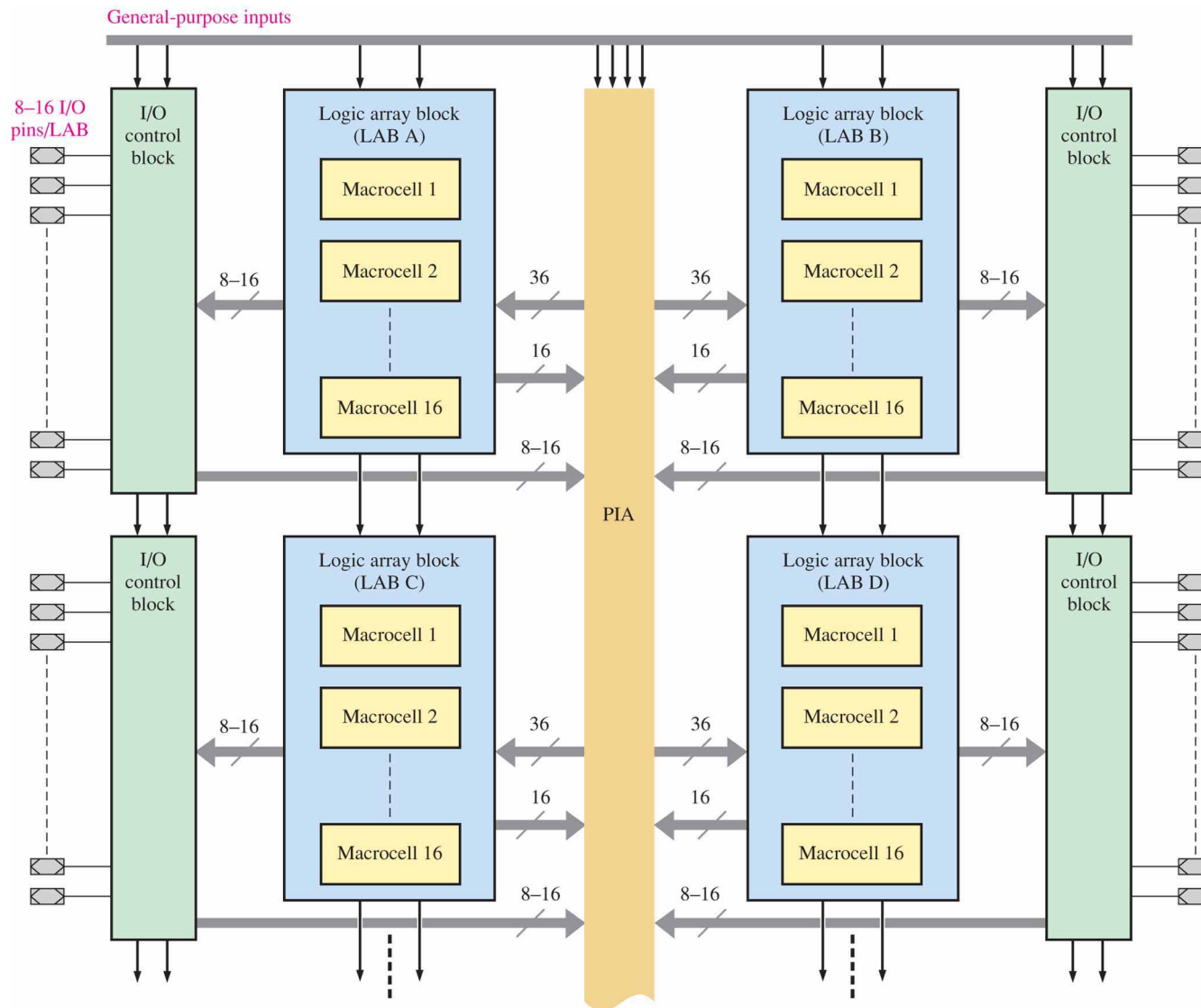


# CPLDs (Complex PLDs)

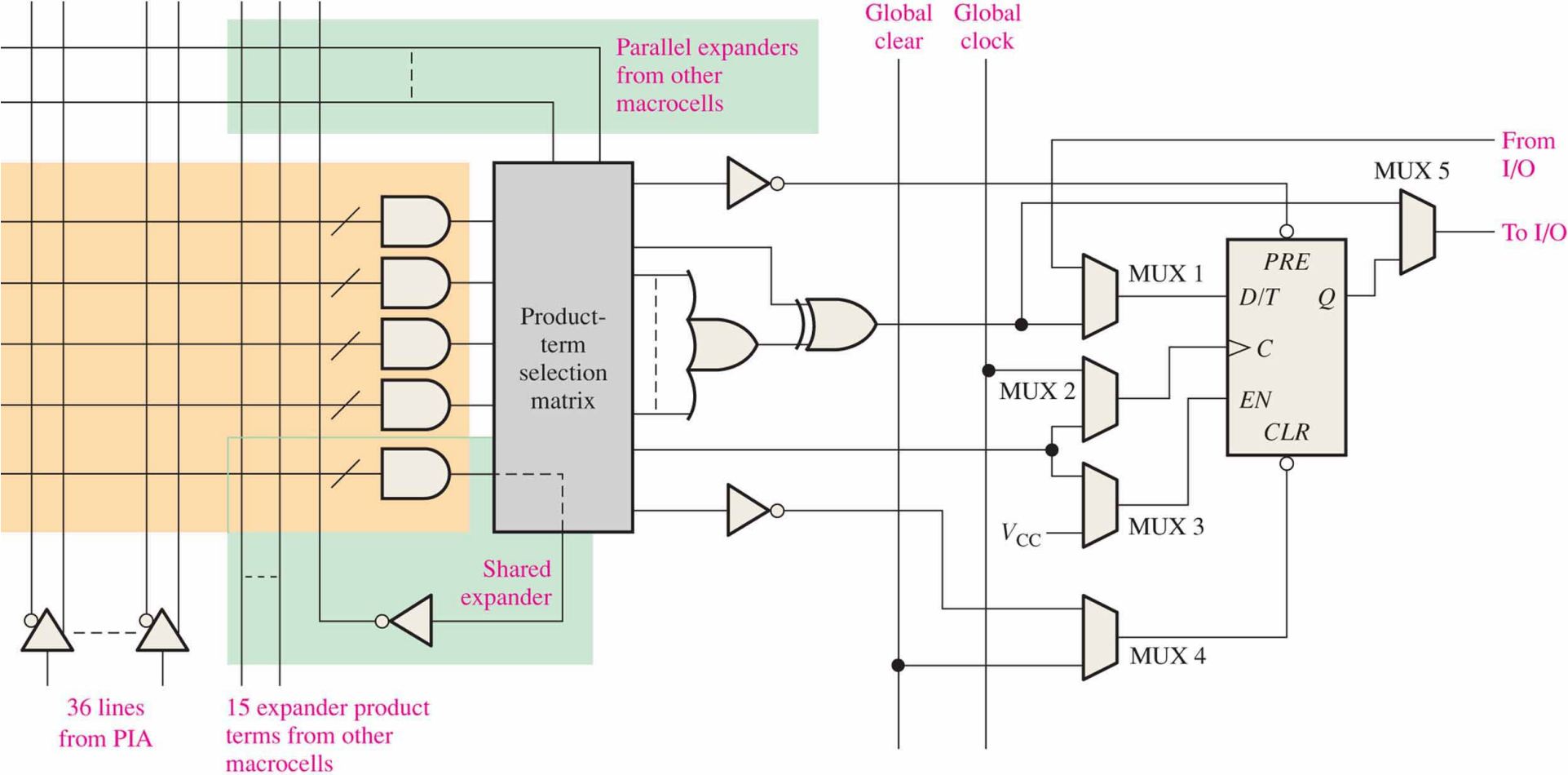
## Arquitectura genérica



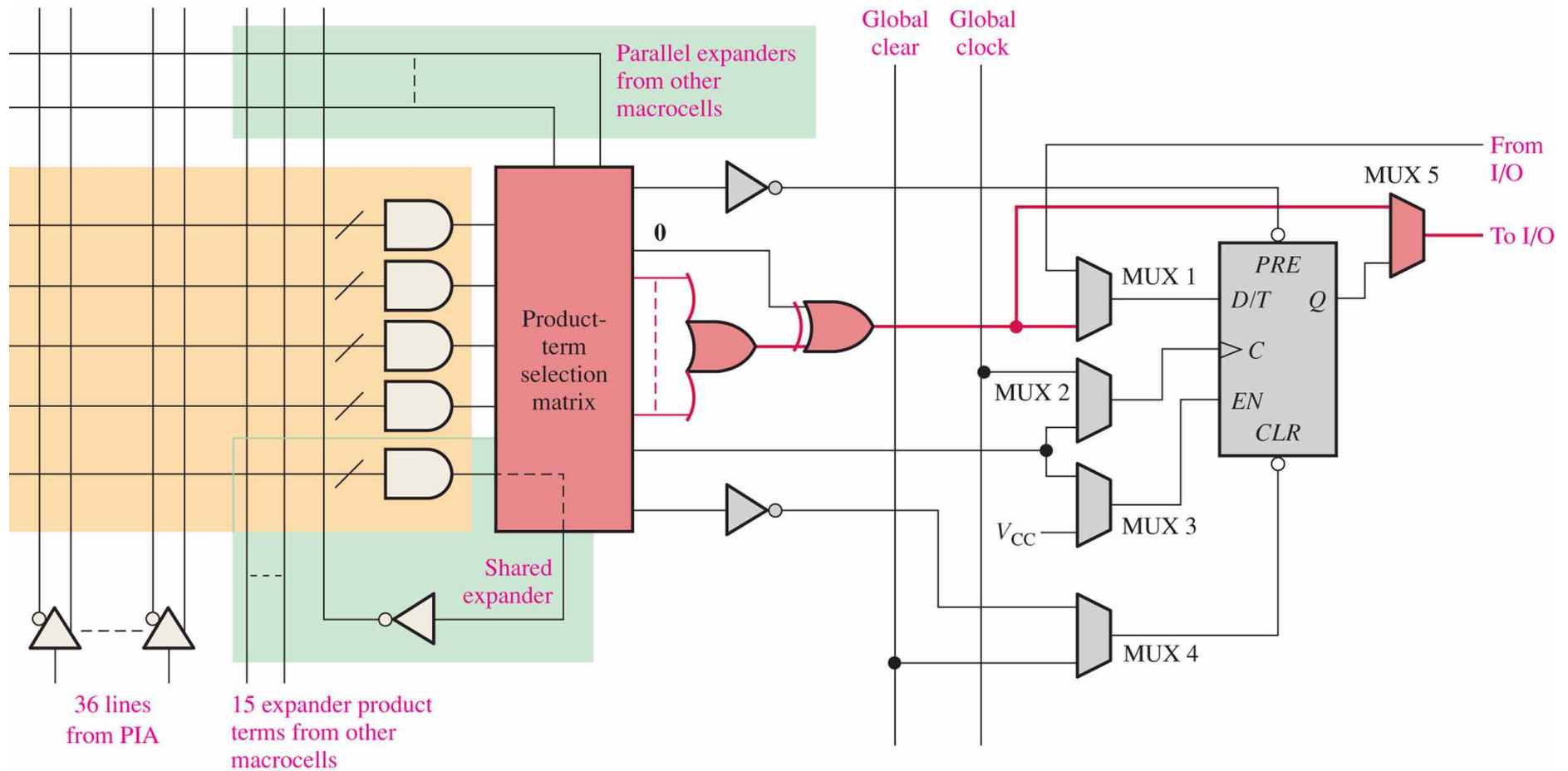
# Ejemplo: Arquitectura CPLD MAX 7000 (Altera)



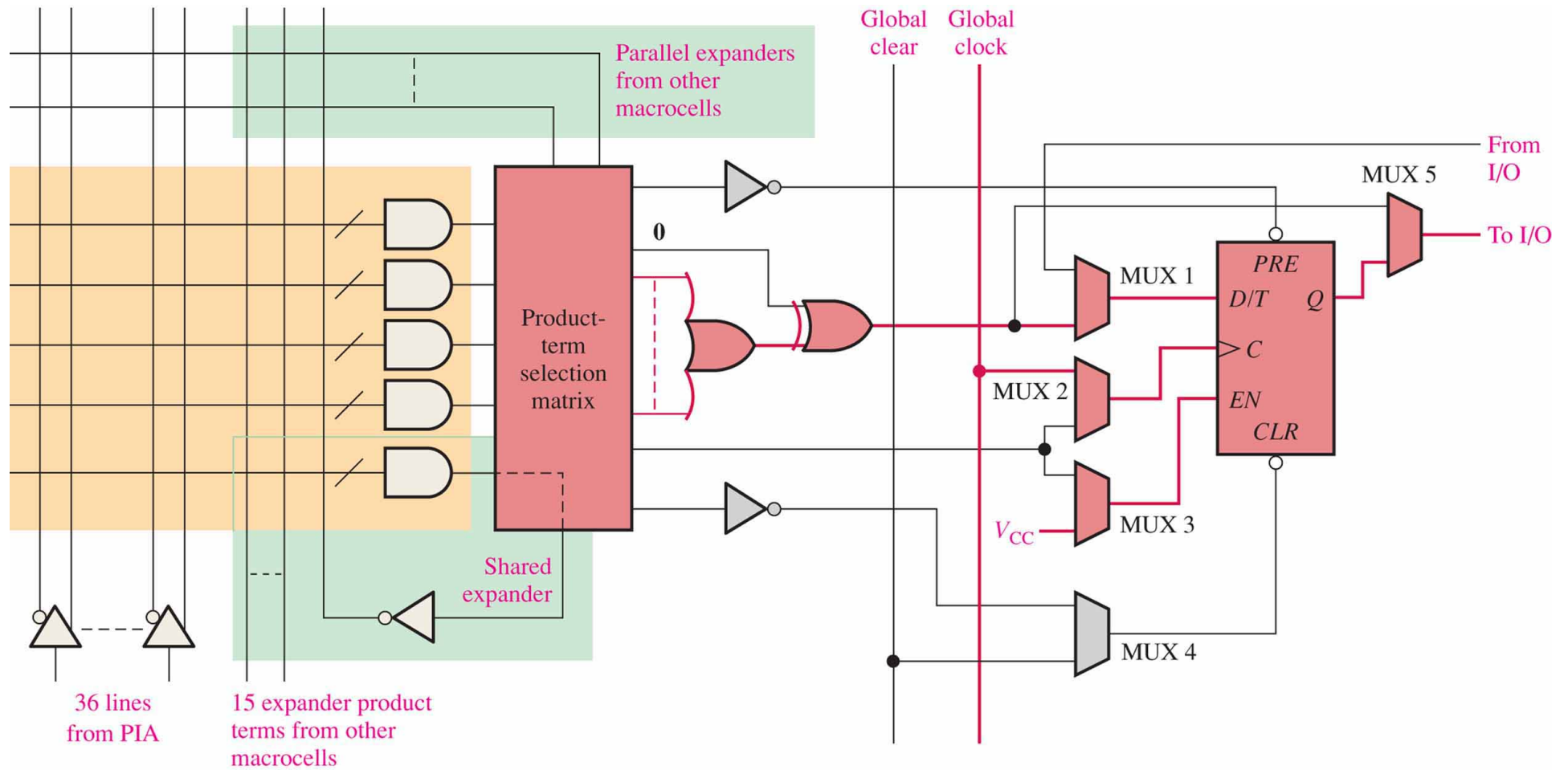
# Ejemplo de macrocelda CPLD: MAX 7000 (Altera)



# Macrocella del CPLD MAX 7000 (Altera) configurada como func. combinacional



# Macrocella del CPLD MAX 7000 (Altera) configurada como func. secuencial





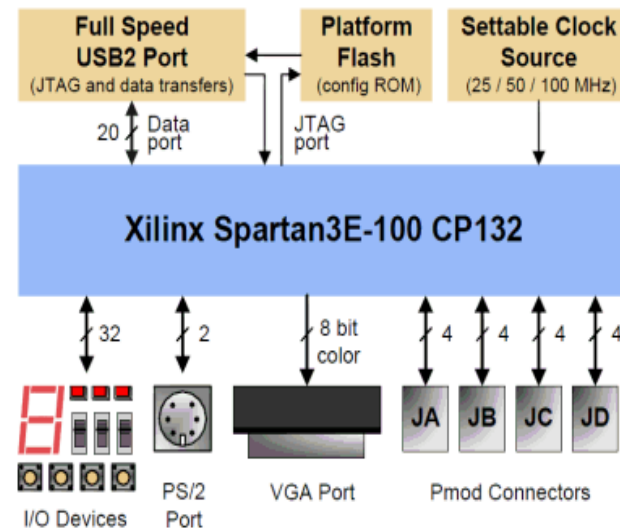
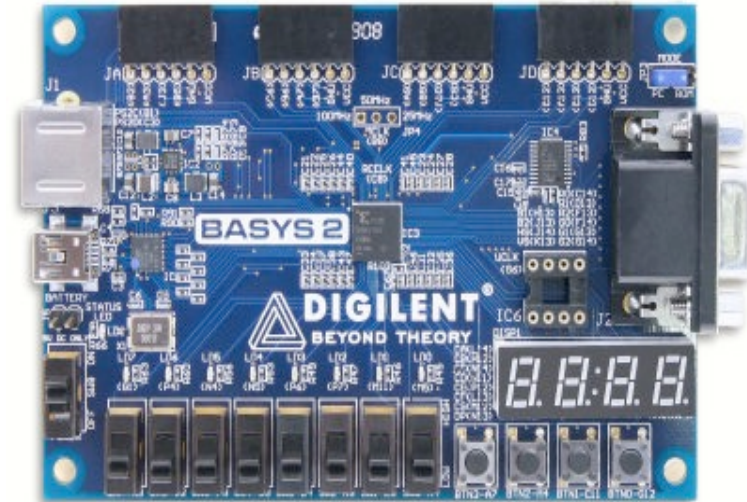
# FPGA (Field-Programmable Gate Array)

- Una FPGA

- Permite implementar “cualquier” circuito digital.

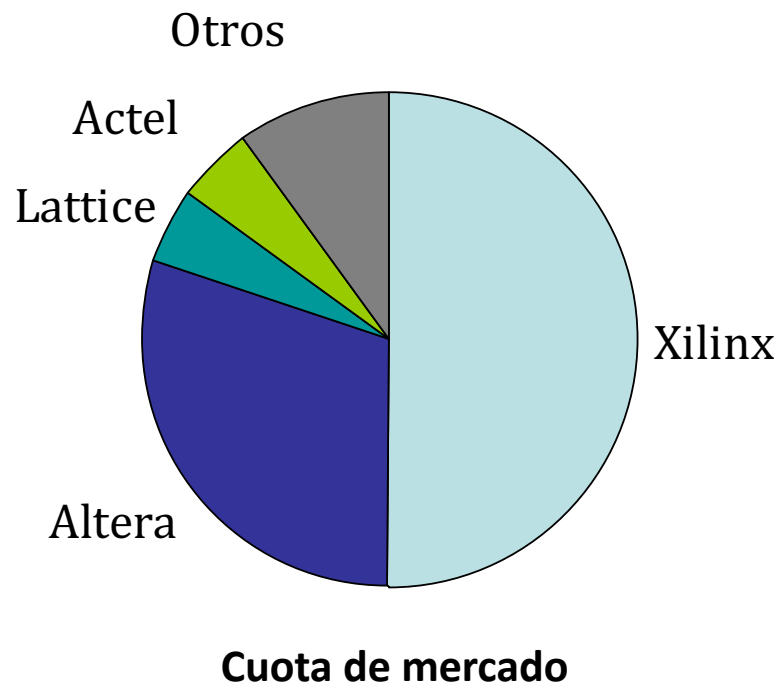
- La única limitación es la cantidad de puertas del circuito.

- El desarrollo se realiza sobre una placa de entrenamiento.



# Principales fabricantes y modelos de FPGA

---



- Xilinx:

- Spartan
- Virtex

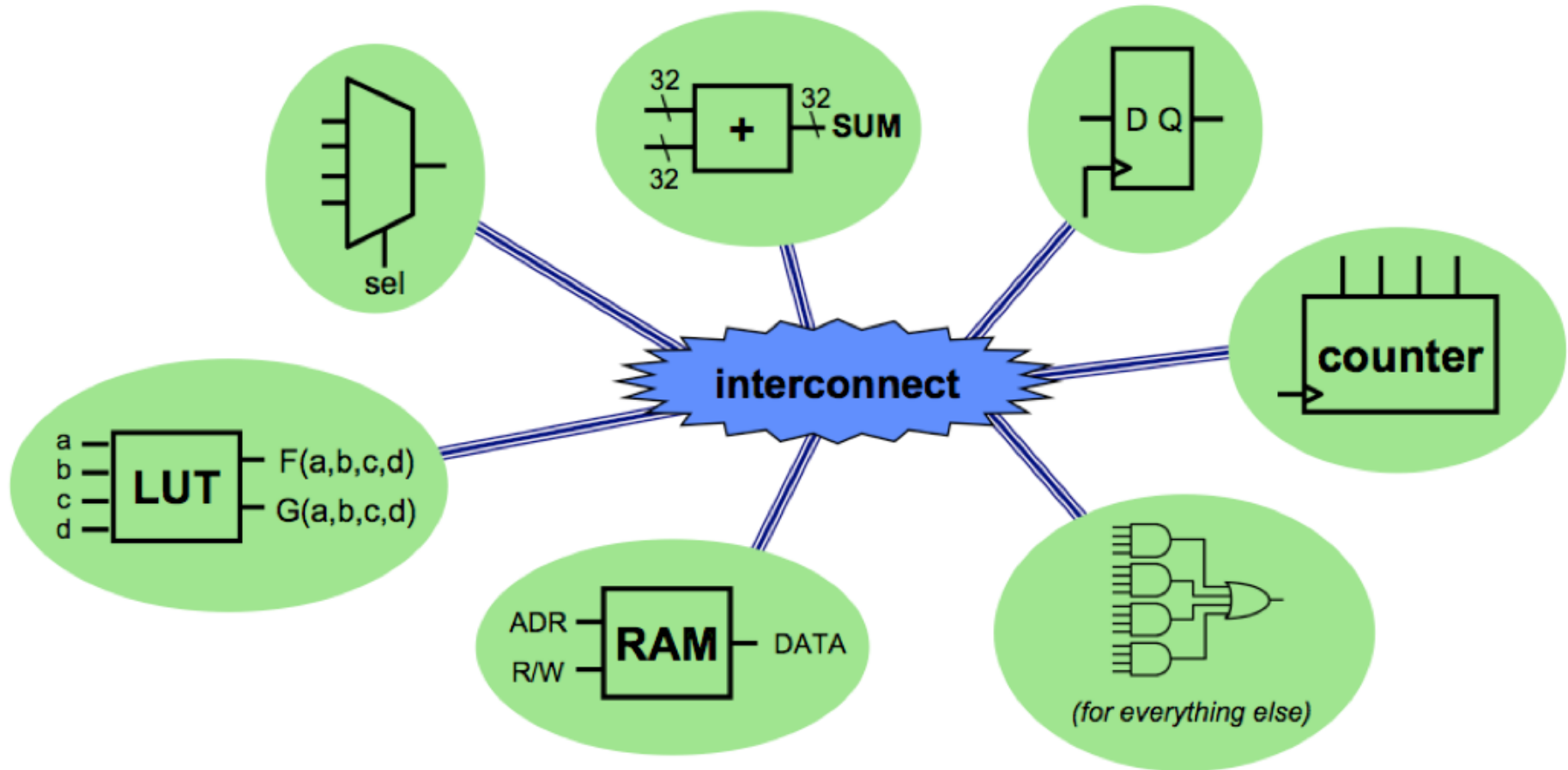
- Altera:

- Arria
- Cyclone
- Stratix

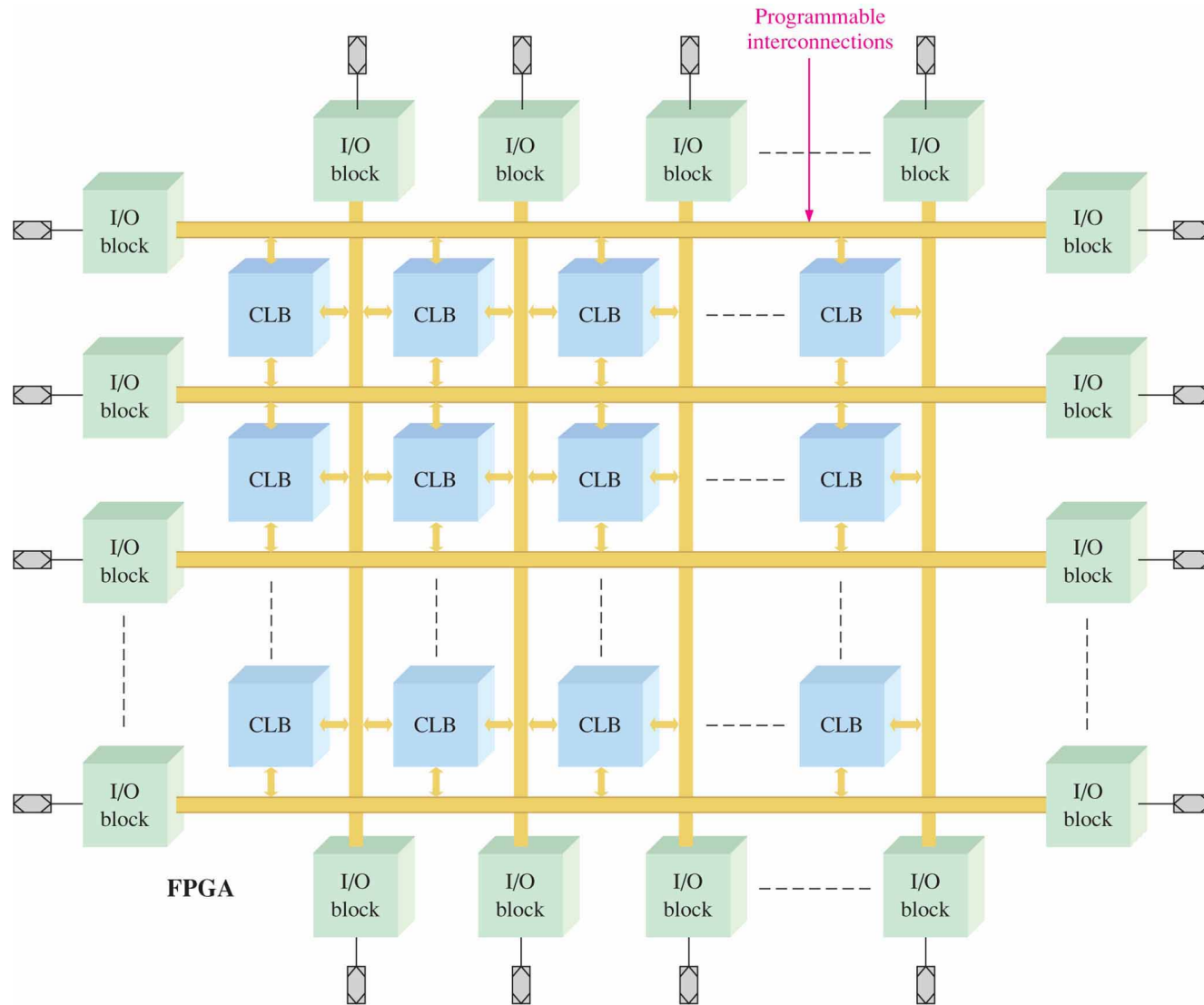
- Actel:

- Igloo
- ProASIC
- SmartFusion

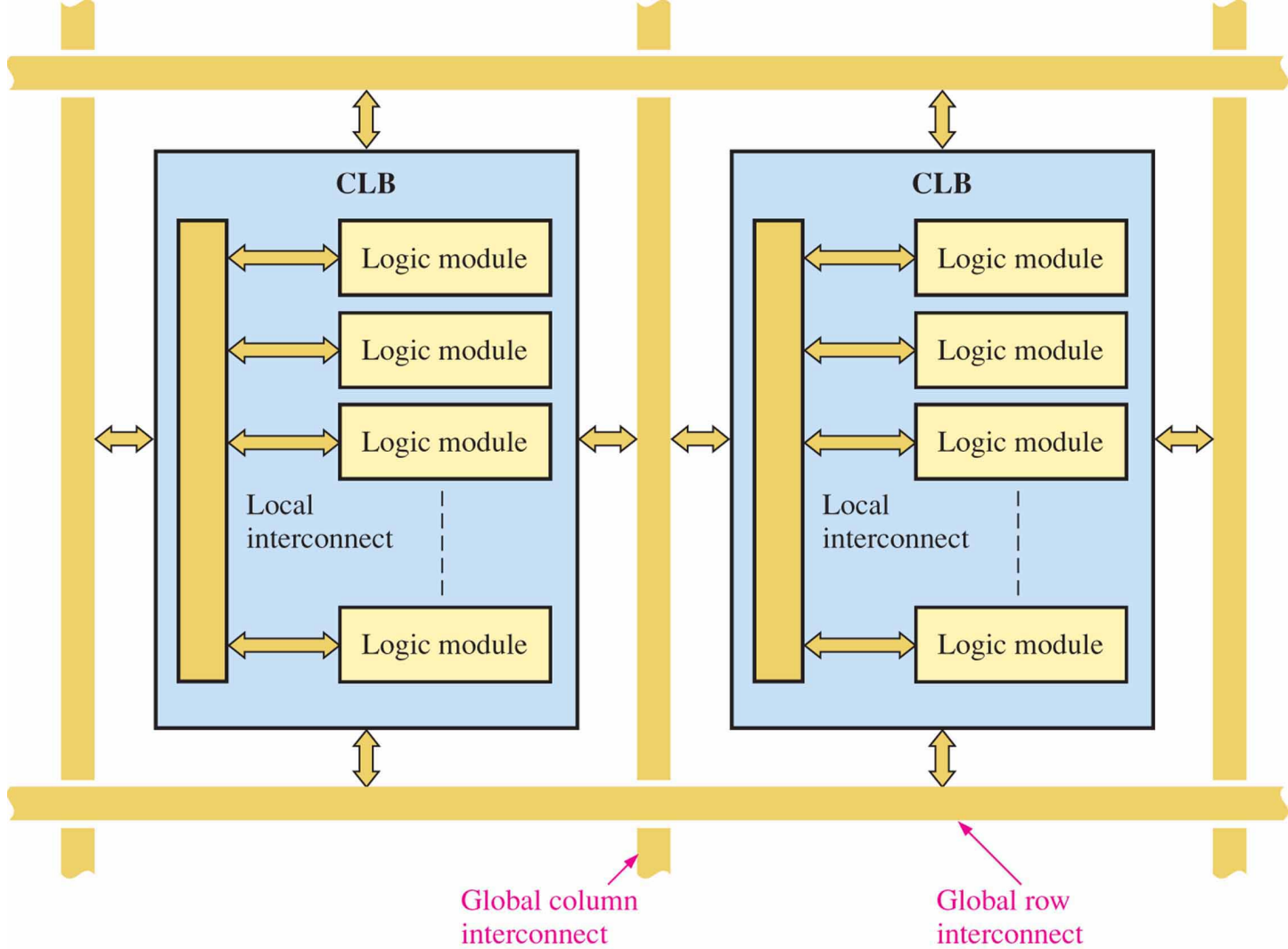
# Concepto de FPGA



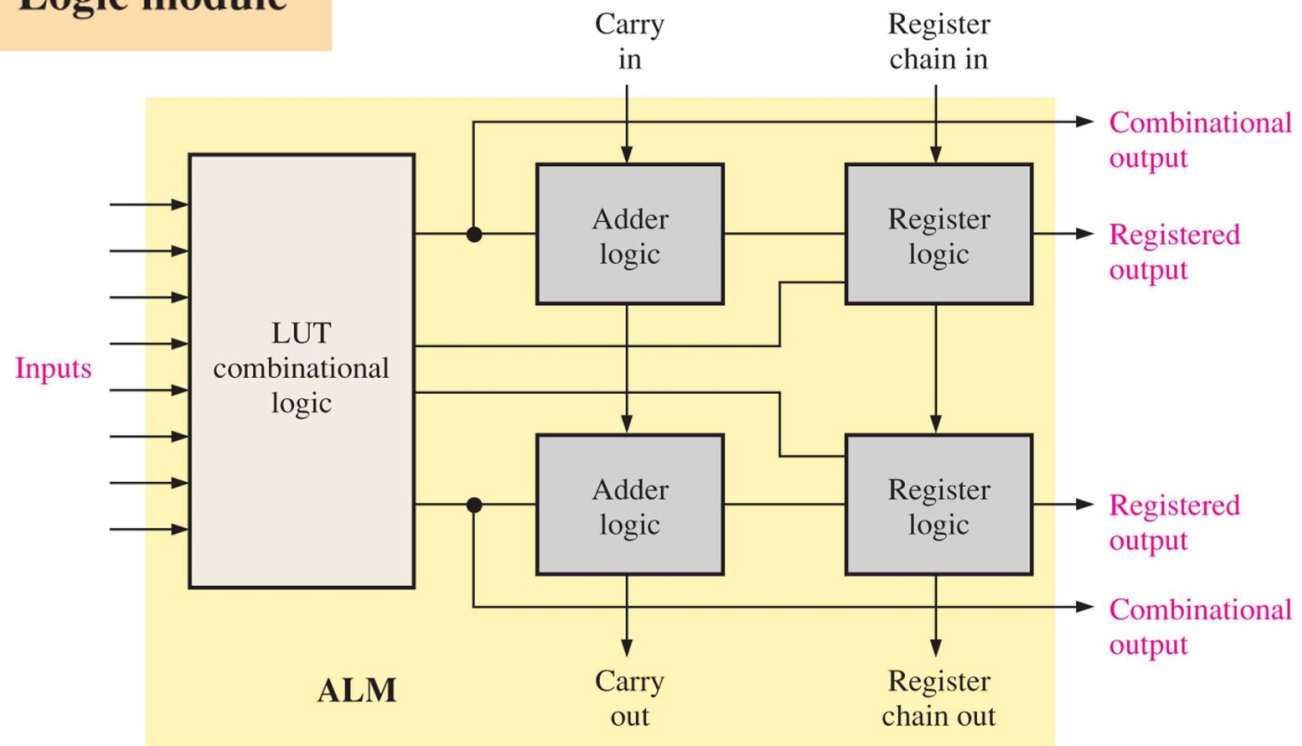
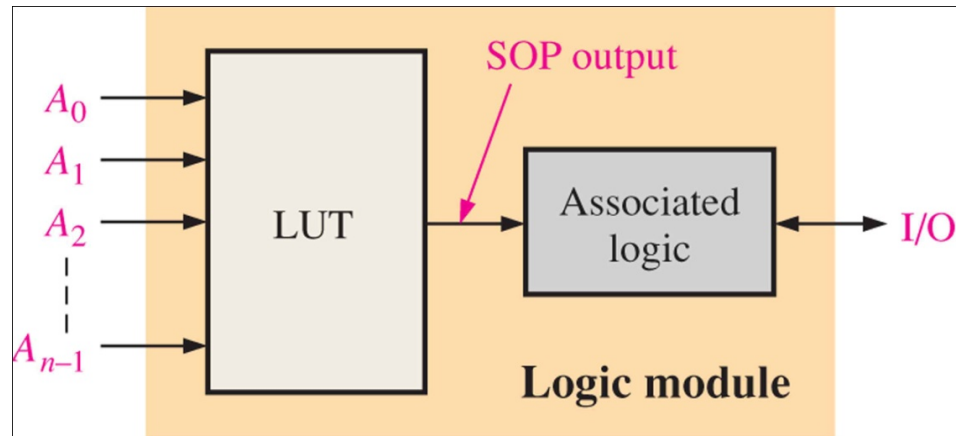
# Arquitectura básica de una FPGA



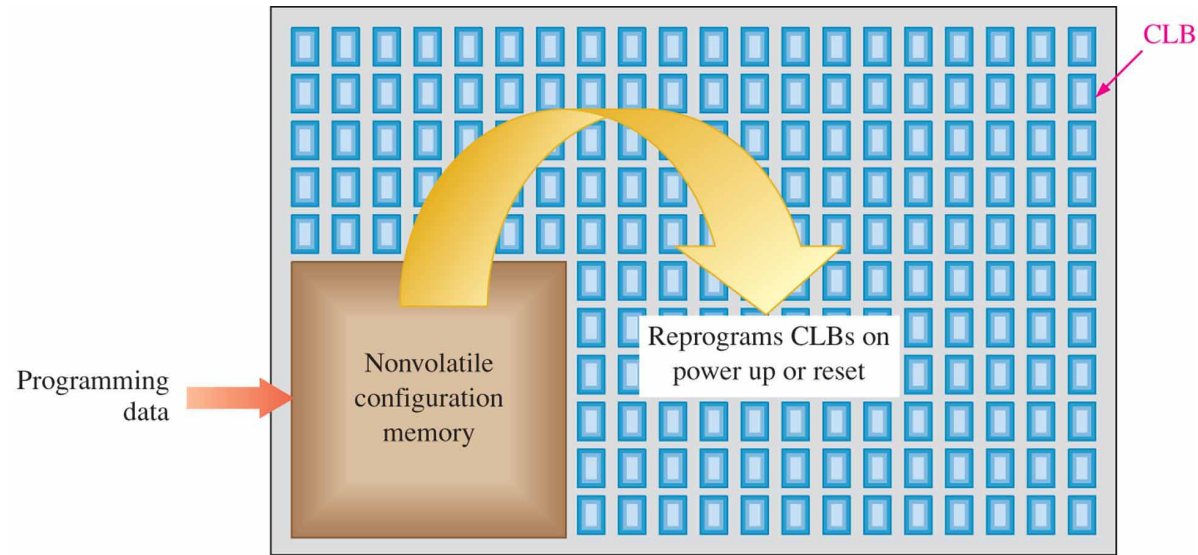
# Estructura de los Bloques Lógicos Configurables (CLB) de una FPGA



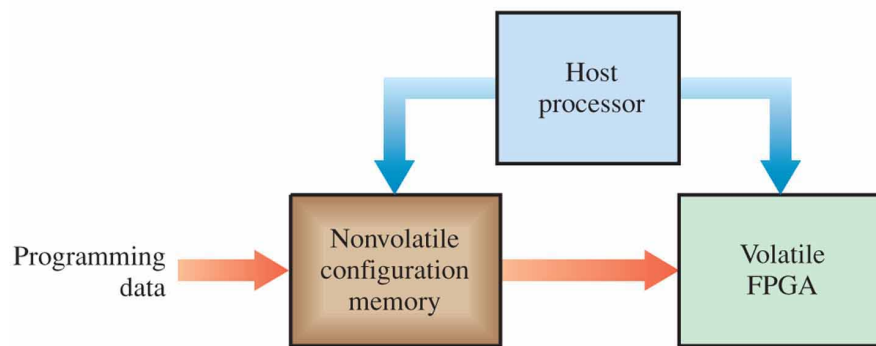
# Estructura de un módulo lógico (incluidos en los CLBs) de una FPGA



# Ejemplos de configuraciones de FPGAs volátiles (LUT SRAM)

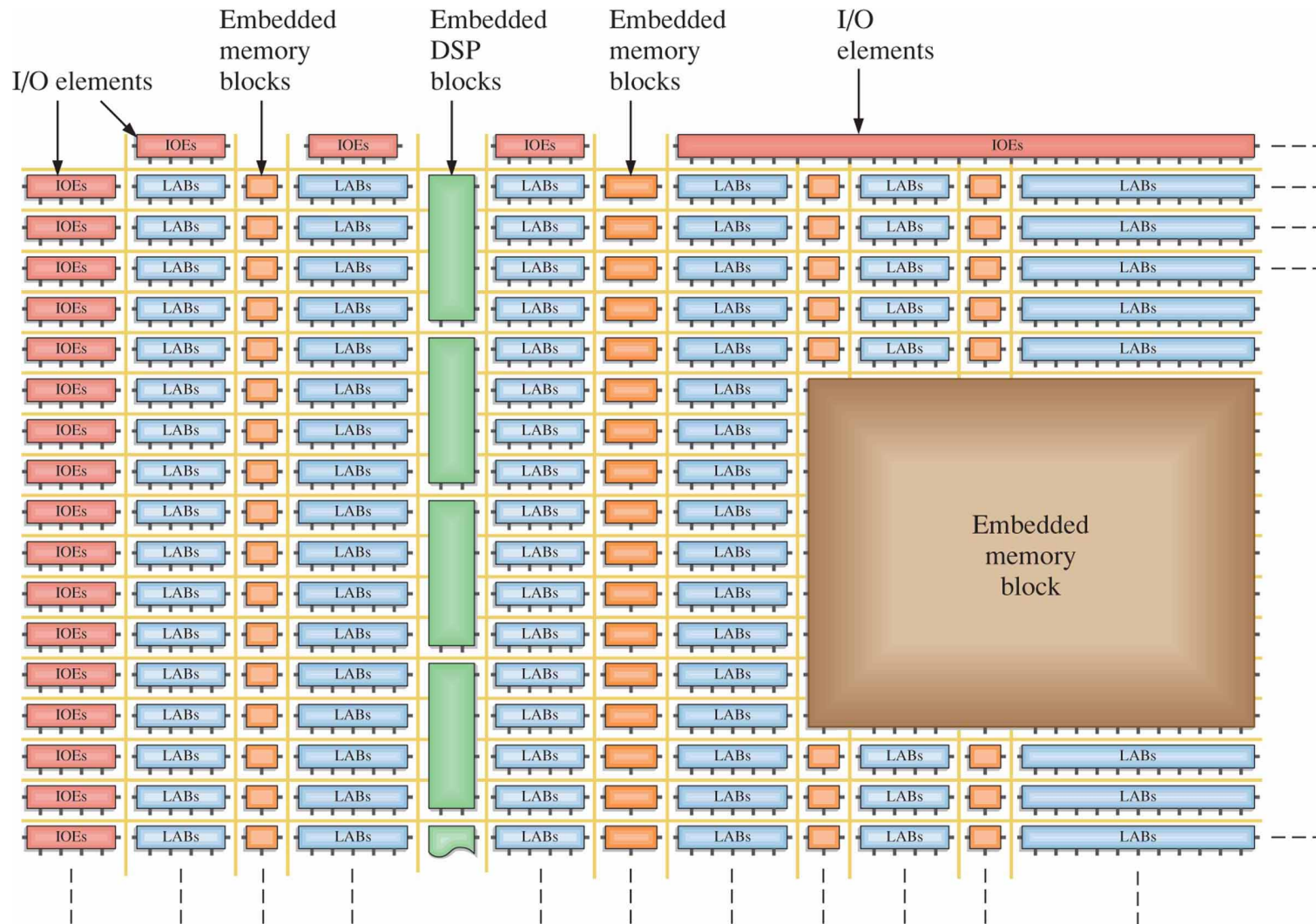


(a) Volatile FPGA with on-the-chip nonvolatile configuration memory



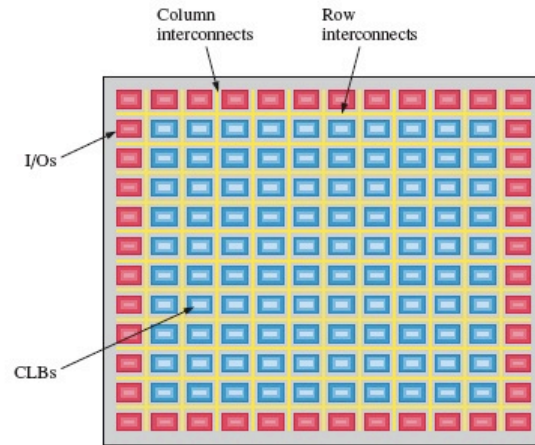
(b) Volatile FPGA with on-board memory and host processor

# Arquitecturas avanzadas de FPGAs

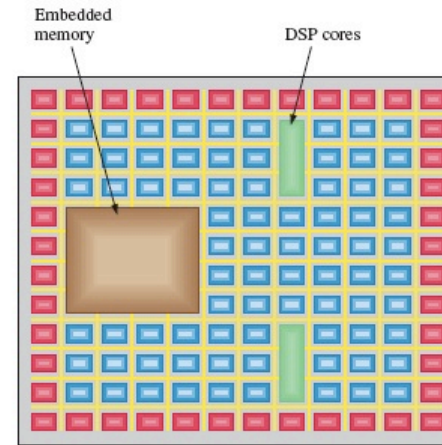




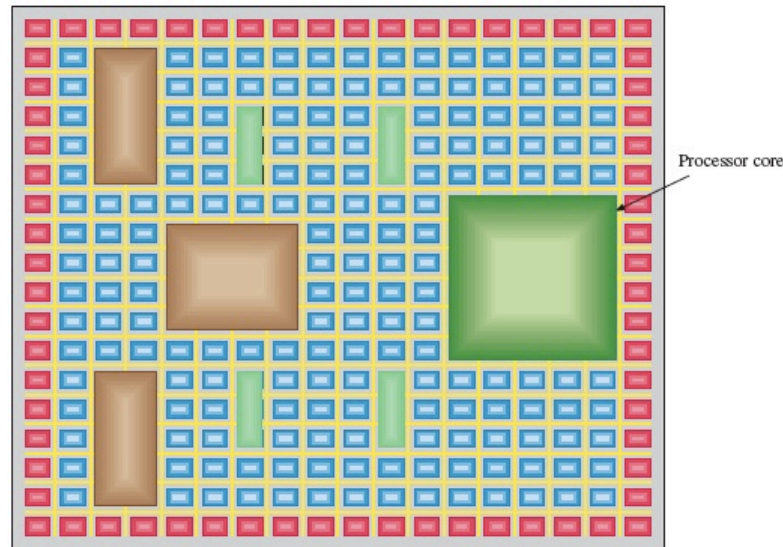
# Arquitecturas avanzadas de FPGAs



(a) FPGA with fully configurable logic



(b) Same size FPGA with embedded memory and IP cores (DSP) results in fewer CLBs and is limited by the perimeter I/Os.



(c) FPGA with more embedded memory, additional DSP cores, and processor core will require a larger physical size at some point.

# Desarrollo de sistemas sobre PLDs

---

- La implementación de sistemas digitales complejos mediante PLDs requiere:
  - Lenguajes de Descripción de Hardware (HDL)
    - Verilog
    - VHDL
  - Herramientas de Desarrollo (SW):
    - Compilación:
      - sintetizan HW a partir de una descripción HDL
    - Simulación
    - Programación del dispositivo
  - Placas de desarrollo (HW)
- En laboratorio usaremos:
  - Placa de desarrollo de FPGAs serie Spartan 3E (Xilinx)
  - Herramientas de desarrollo ISE Verilog