

---

# Tema9

## Memorias y dispositivos programables

# Contenidos

---

## Memorias:

- Concepto y jerarquía de memorias.
- Principios de las memorias semiconductoras.
- Memorias de acceso aleatorio (RAM).
- Memorias de sólo lectura (ROM).
- Expansión de memorias.
- Tipos especiales de memorias.

Dispositivos programables: Arquitectura básica de una FPGA.

---

# Memorias y dispositivos programables

Concepto y jerarquía de memorias

# Concepto de memoria

---

- Una computadora requiere del almacenamiento permanente de una gran cantidad de datos binarios.
- Los sistemas basados en microprocesador necesitan dispositivos de memoria con objeto de almacenar los programas y 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.

# Clasificación de memorias

Característica		Cinta	HDD	RAM	ROM	FIFO	DVD
Tecnología	Magnética	X	X				
	Semiconductor			X	X	X	
	Óptica						X
Acceso	Aleatorio			X	X		
	Secuencial	X	X			X	X
Volatilidad	Volátil			X		X	
	No volátil	X	X		X		X

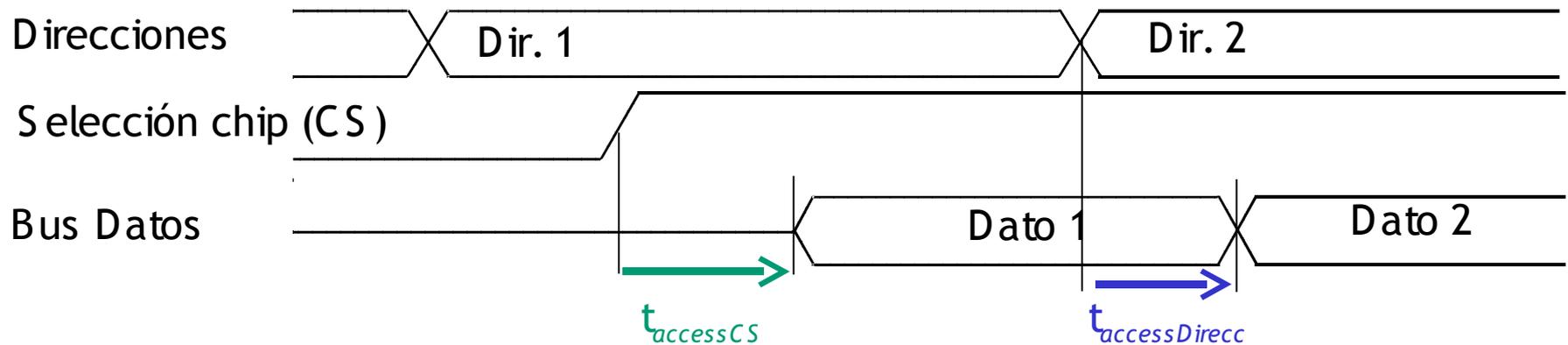
# Clasificación de memorias

## Velocidad y tiempo de acceso

**Velocidad de acceso:** Cantidad de bits que se transfieren por segundo (bits por segundo)

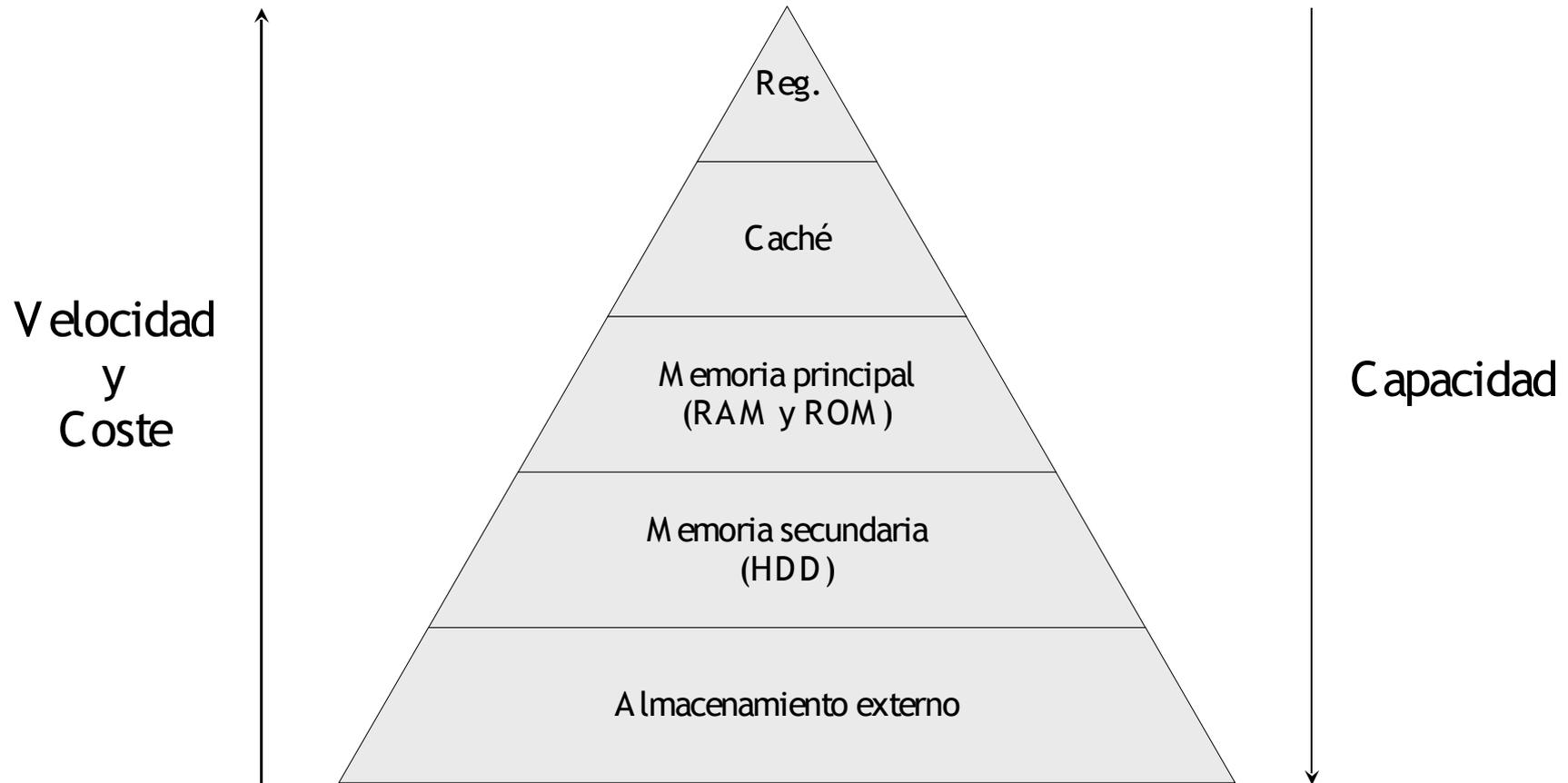
**Tiempo de acceso ( $t_{access}$ ):** Intervalo de tiempo que transcurre entre la orden de acceso y el acceso al dato.

Ejemplo de acceso de lectura a ROM o RAM:



# Jerarquía de memorias

---



---

# Memorias y dispositivos programables

Principios de las memorias semiconductoras

# Matriz de memoria básica

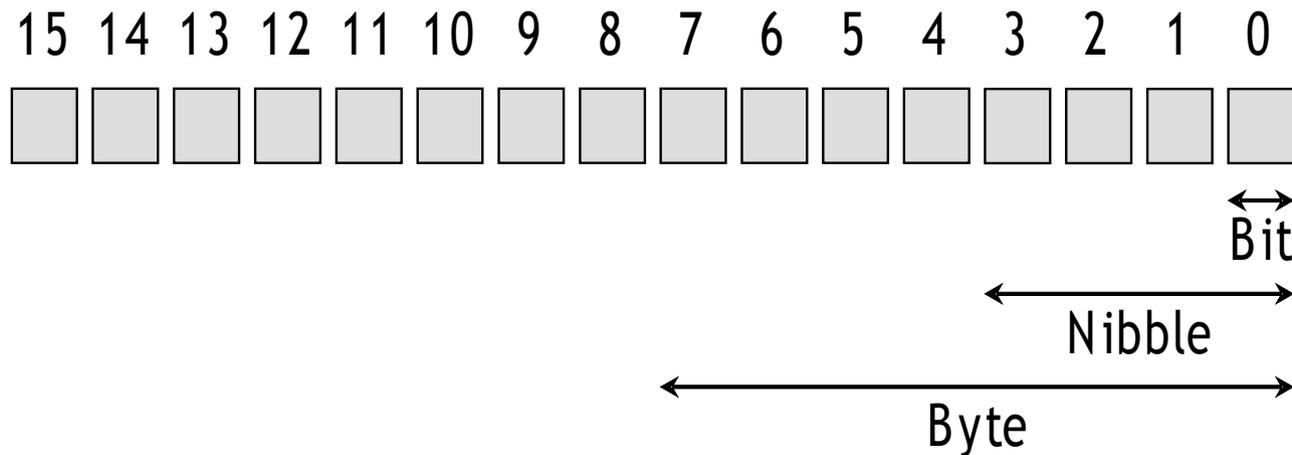
- Cada elemento de almacenamiento de una memoria puede almacenar 1 bit y se denomina **celda**.
- Las memorias están formadas por **matrices de celdas**.
- Cada **fila** de esa matriz de memoria se denomina **palabra** y representa la información que puede leerse/escribirse en cada acceso a la misma.

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

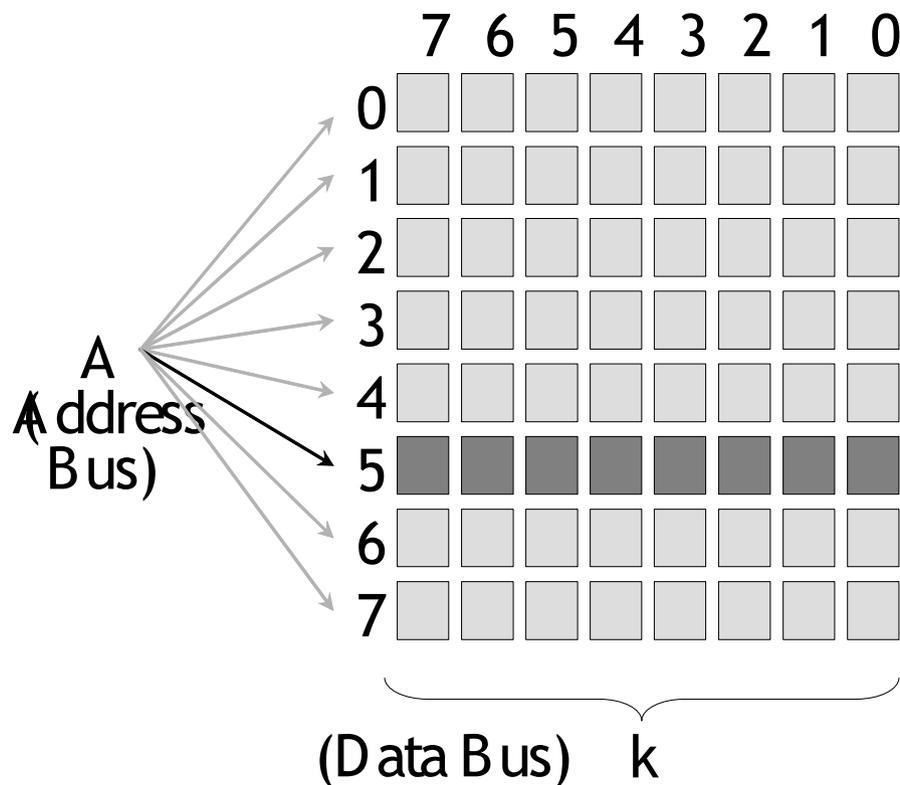
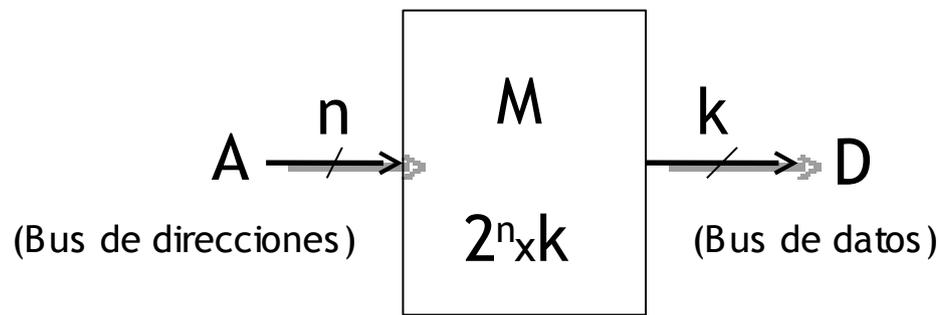
# Unidades de datos binarios

---

- Una palabra de bits puede tener cualquier longitud aunque por defecto suele considerarse de 16 bits.
- En cualquier caso, una palabra de bits puede descomponerse en las siguientes unidades:



# Dirección y capacidad de las memorias



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

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

# Operaciones básicas de las memorias

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

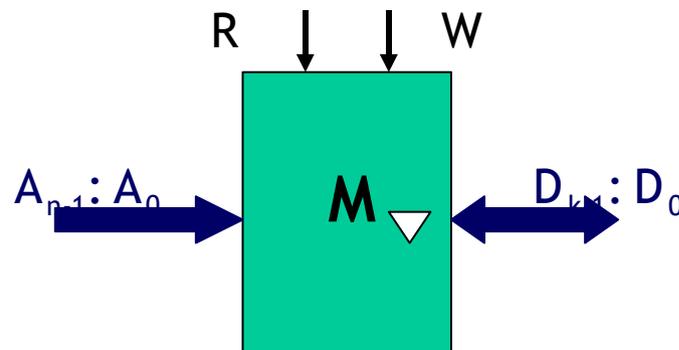
**Escritura** (*write*): permite almacenar una palabra en una determinada dirección de la memoria.

**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.

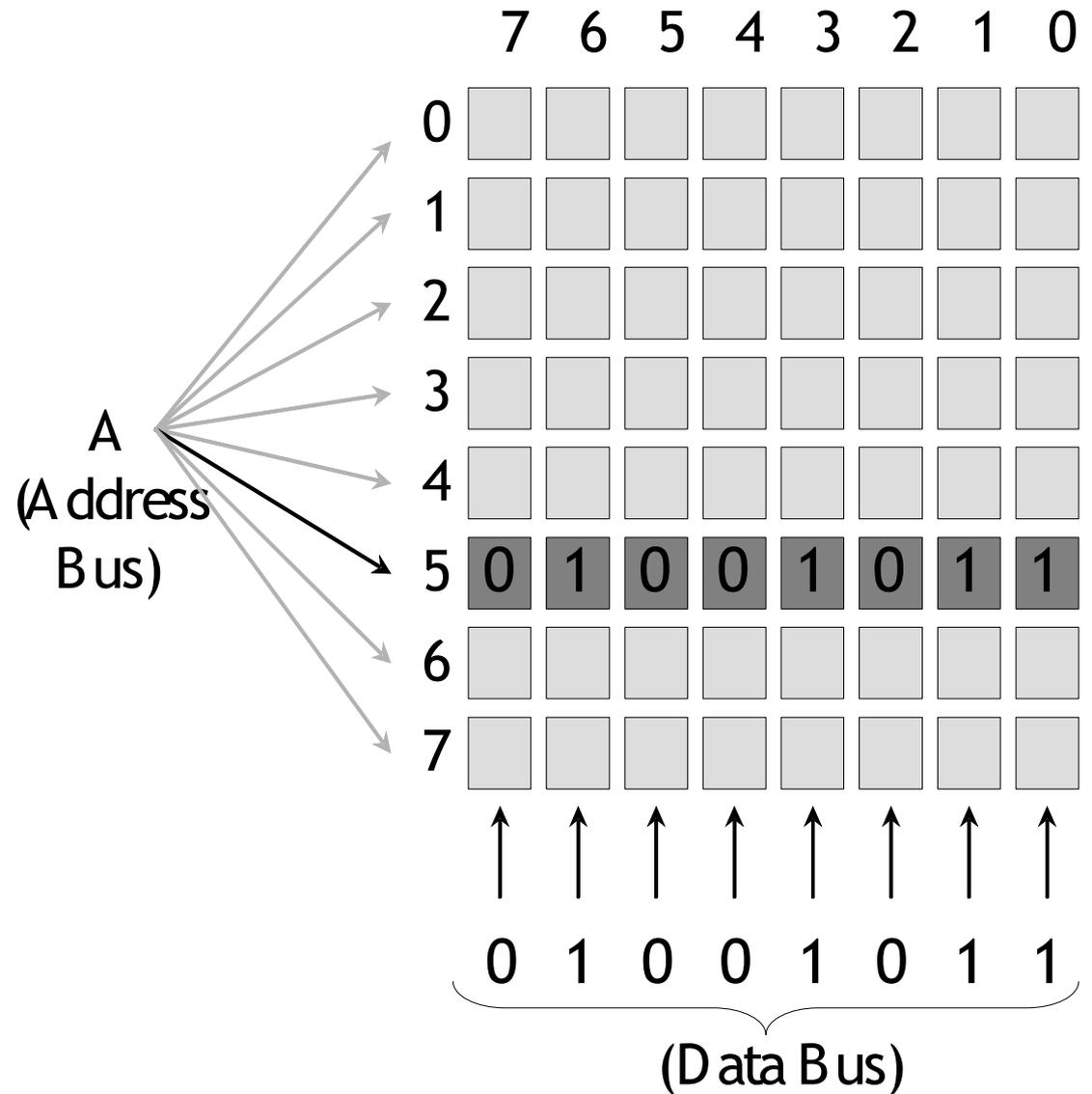


# Operación de escritura

Data Bus  $\longrightarrow$  M(n)

En el ejemplo:

01001011  $\longrightarrow$  M(101)

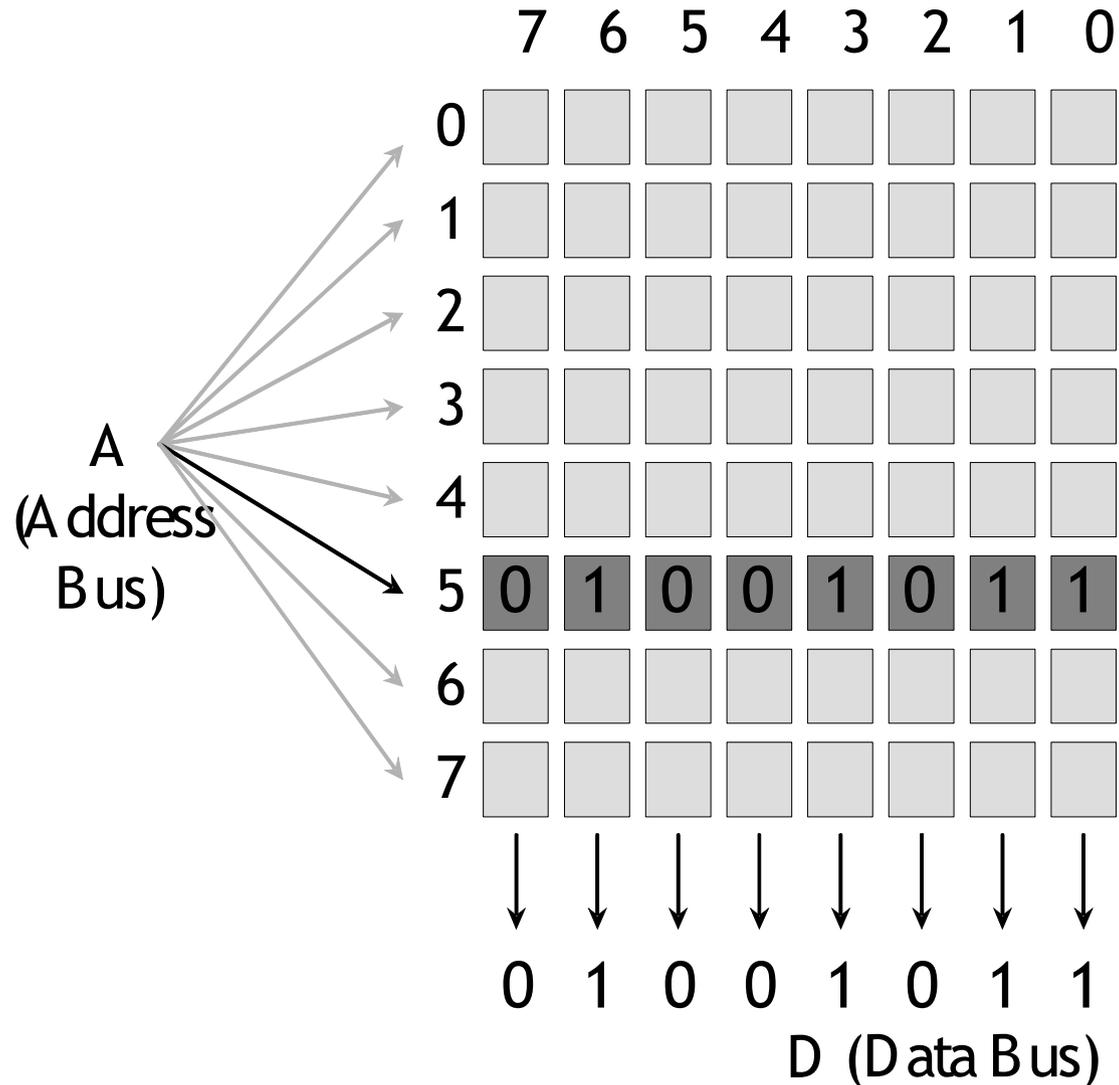


# Operación de lectura

$$D = [M(n)]$$

En el ejemplo:

$$M(101) = 01001011$$



# Las memorias RAM y ROM

---

Las dos principales categorías de memorias semiconductoras 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.

---

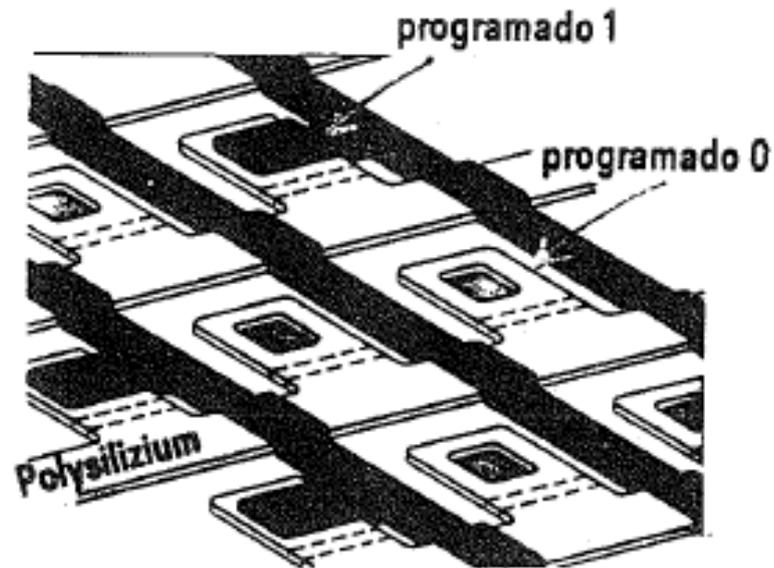
# Memorias y dispositivos programables

Memorias de sólo lectura (ROM)

# Tecnologías de la ROM

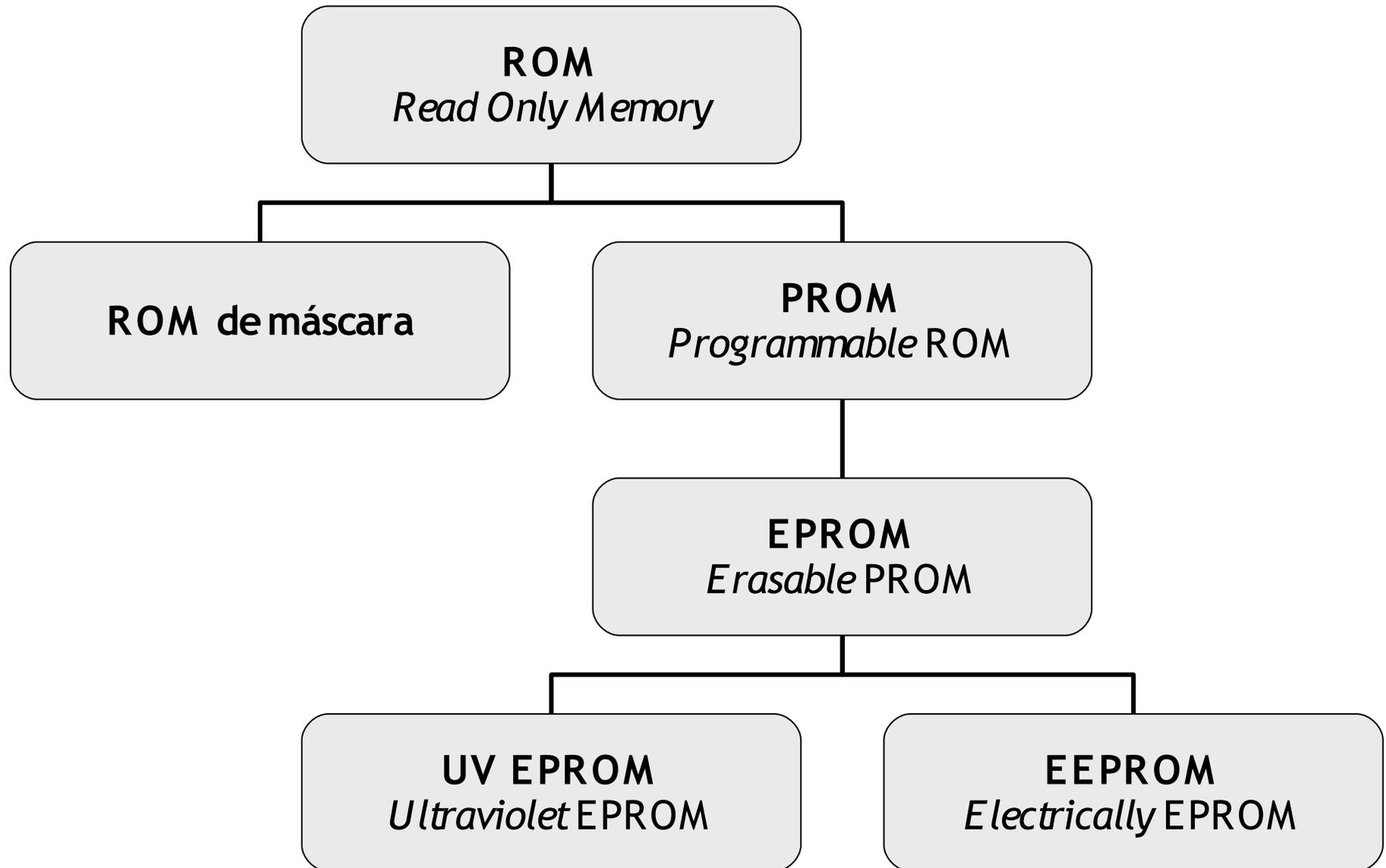
- TTL y MOS
- Con metalización
- Con fusibles

**ROM**  
Módulos programados  
con máscaras  
por el fabricante;  
  
sin posibilidad  
de modificar la información

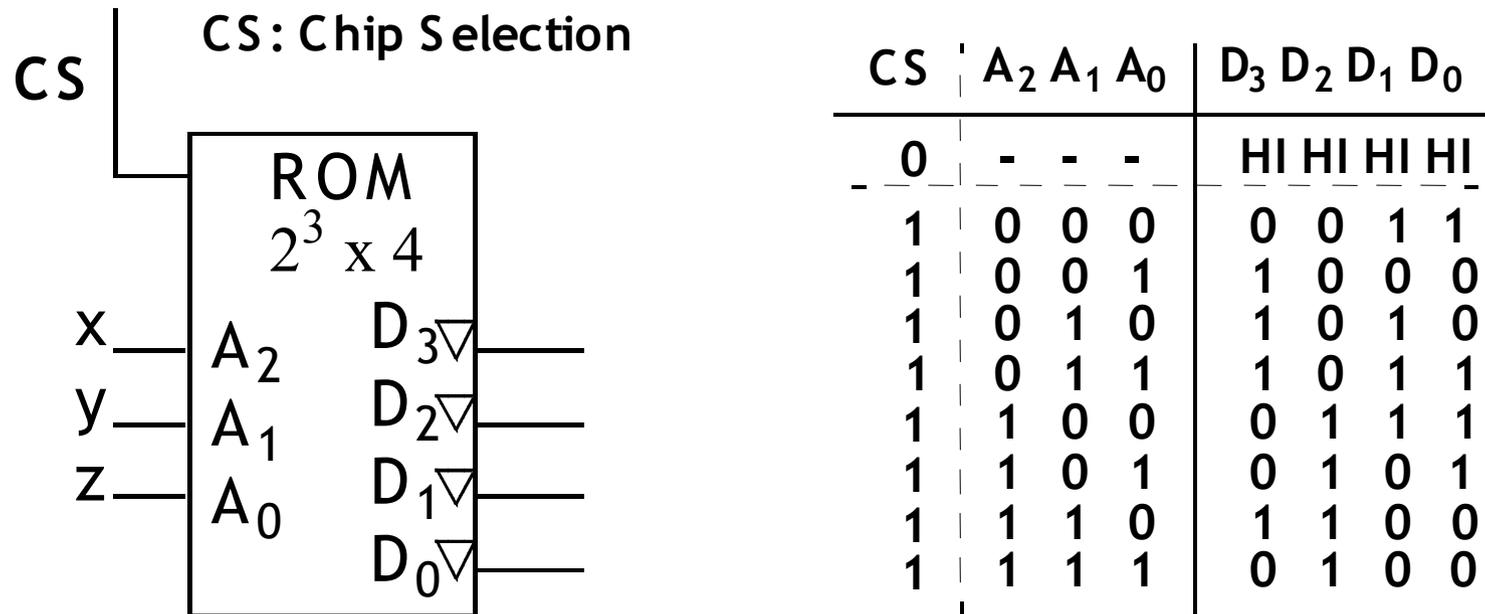


# Familia de memorias ROM

---



# Memoria ROM

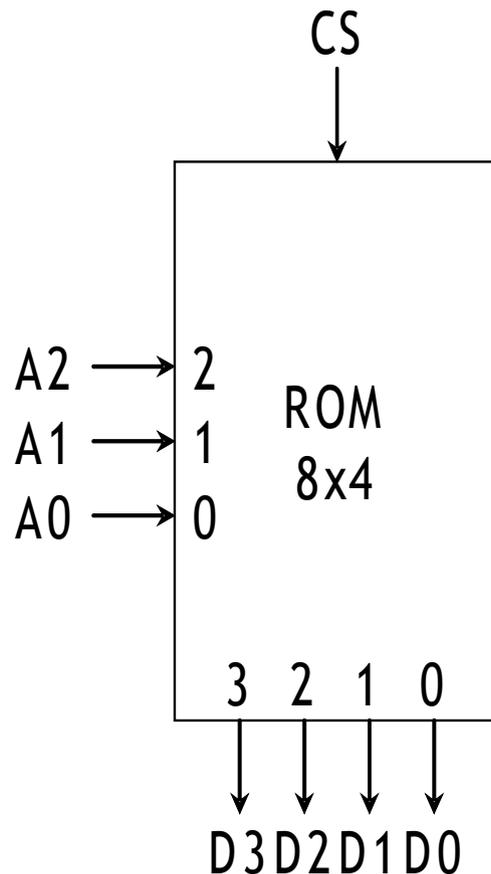


En cada columna se realiza una función (cuando CS=1). P. ej.

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

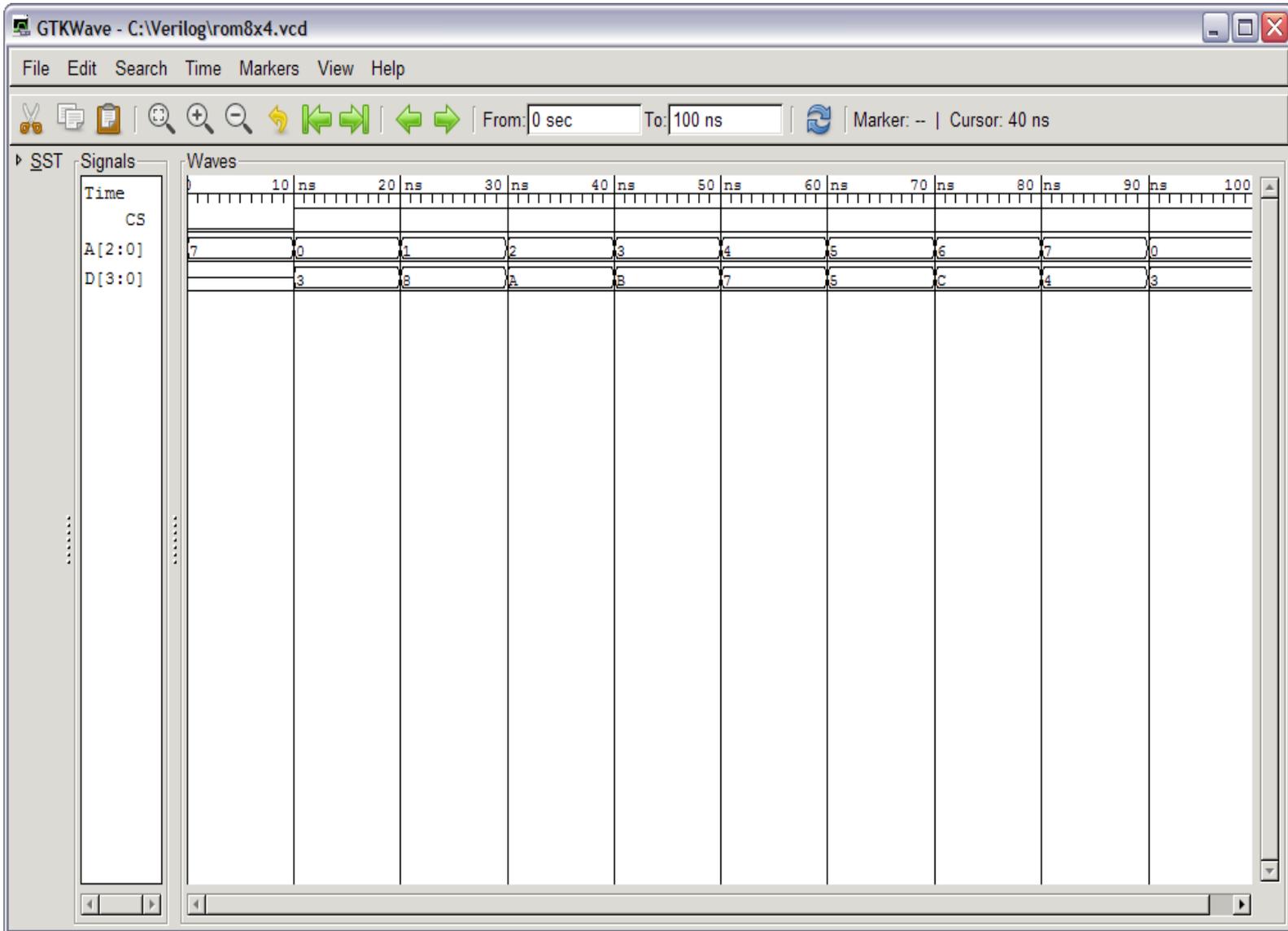
# Memoria ROM

## Descripción de ROM en lenguaje Verilog



```
module rom8x4(  
    input CS,  
    input [2:0] A,  
    output reg [3:0] D  
);  
  
always @(CS, A)  
    if (CS)  
        case (A)  
            0: D = 'h3;  
            1: D = 'h8;  
            2: D = 'hA;  
            3: D = 'hB;  
            4: D = 'h7;  
            5: D = 'h5;  
            6: D = 'hC;  
            default: D = 'h4;  
        endcase  
    else  
        D = 'hZ;  
    end  
endmodule // rom8x4
```

# Operación de memoria ROM



---

# Memorias y dispositivos programables

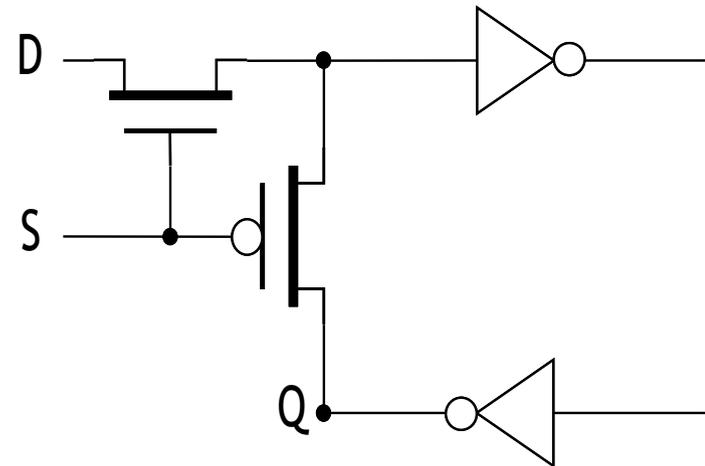
Memorias de acceso aleatorio (RAM)

# Familia de memorias RAM

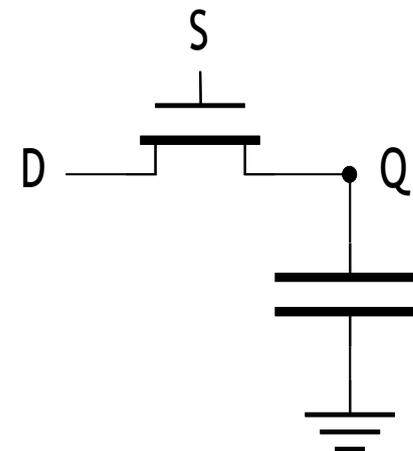
• Las dos categorías principales de memorias RAM son:

- **SRAM** (*Static RAM*):  
realizada con *flip-flops* (más rápida).

- **DRAM** (*Dynamic RAM*):  
realizada con condensadores  
(más económica).



Celda de SRAM

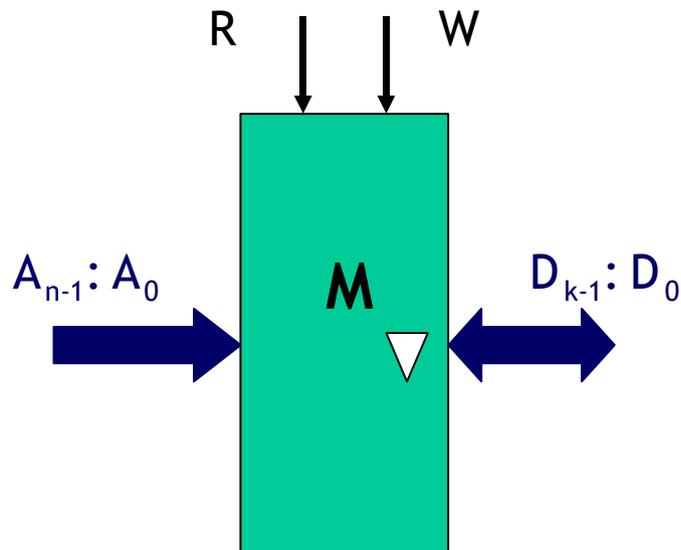


Celda de DRAM

# Descripción de memoria RAM

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

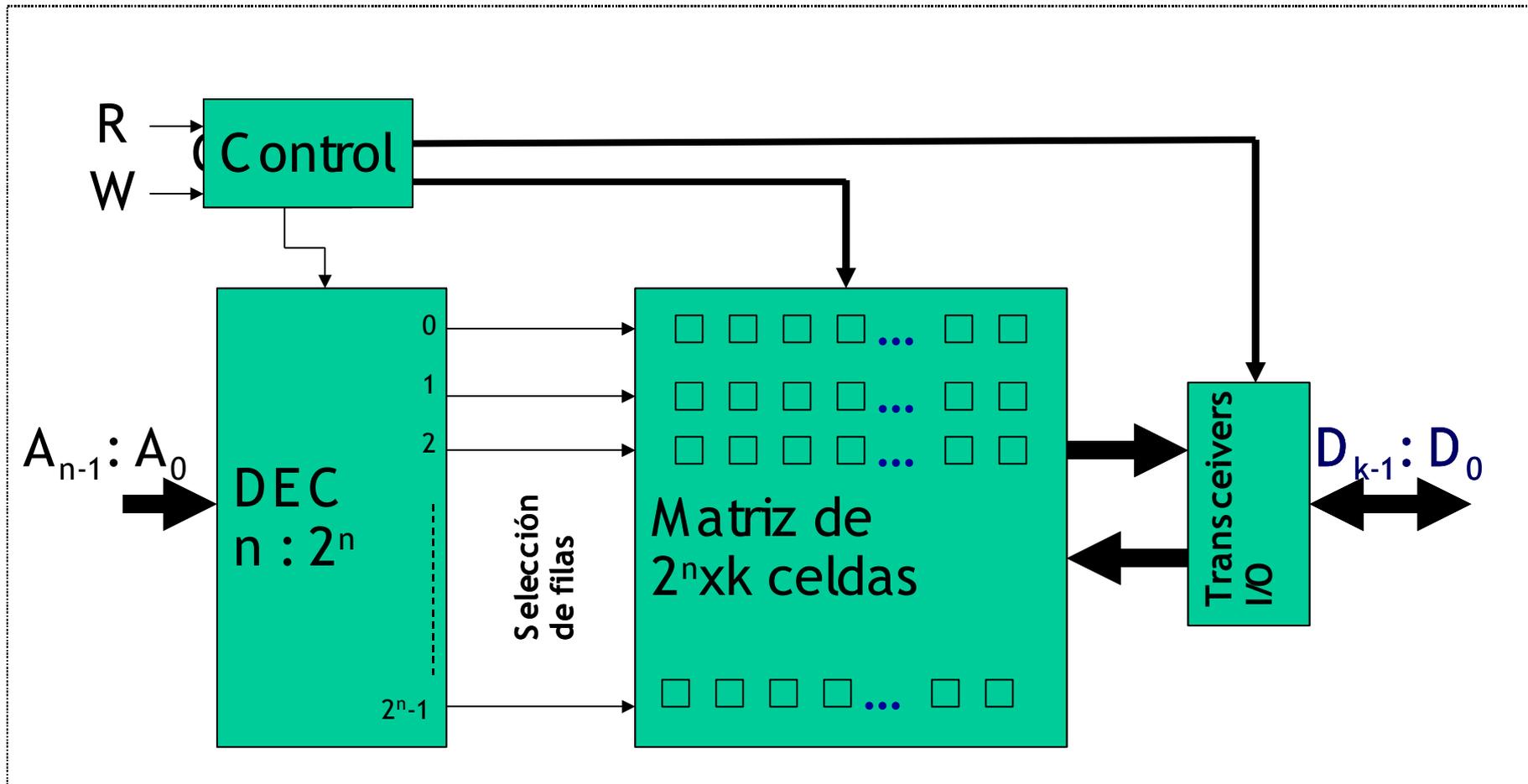
Descripción de RAM  $2^n \times k$  con líneas de datos bidireccionales:



RW	M ←	D =
00	M ← M	HI
01	M(A) ← D	[D in]
10	M ← M	D = M(A)
11	Prohibido	

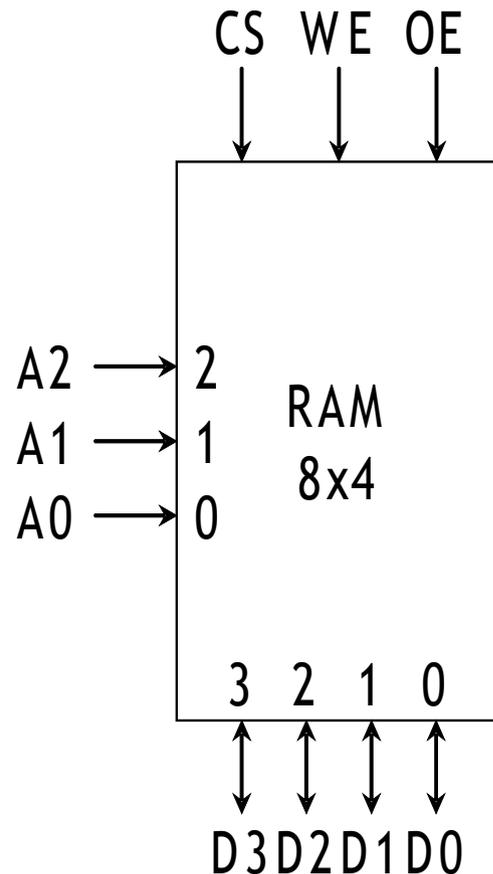
# Diseño de memoria RAM

## RAM $2^n \times k$ : Estructura interna básica



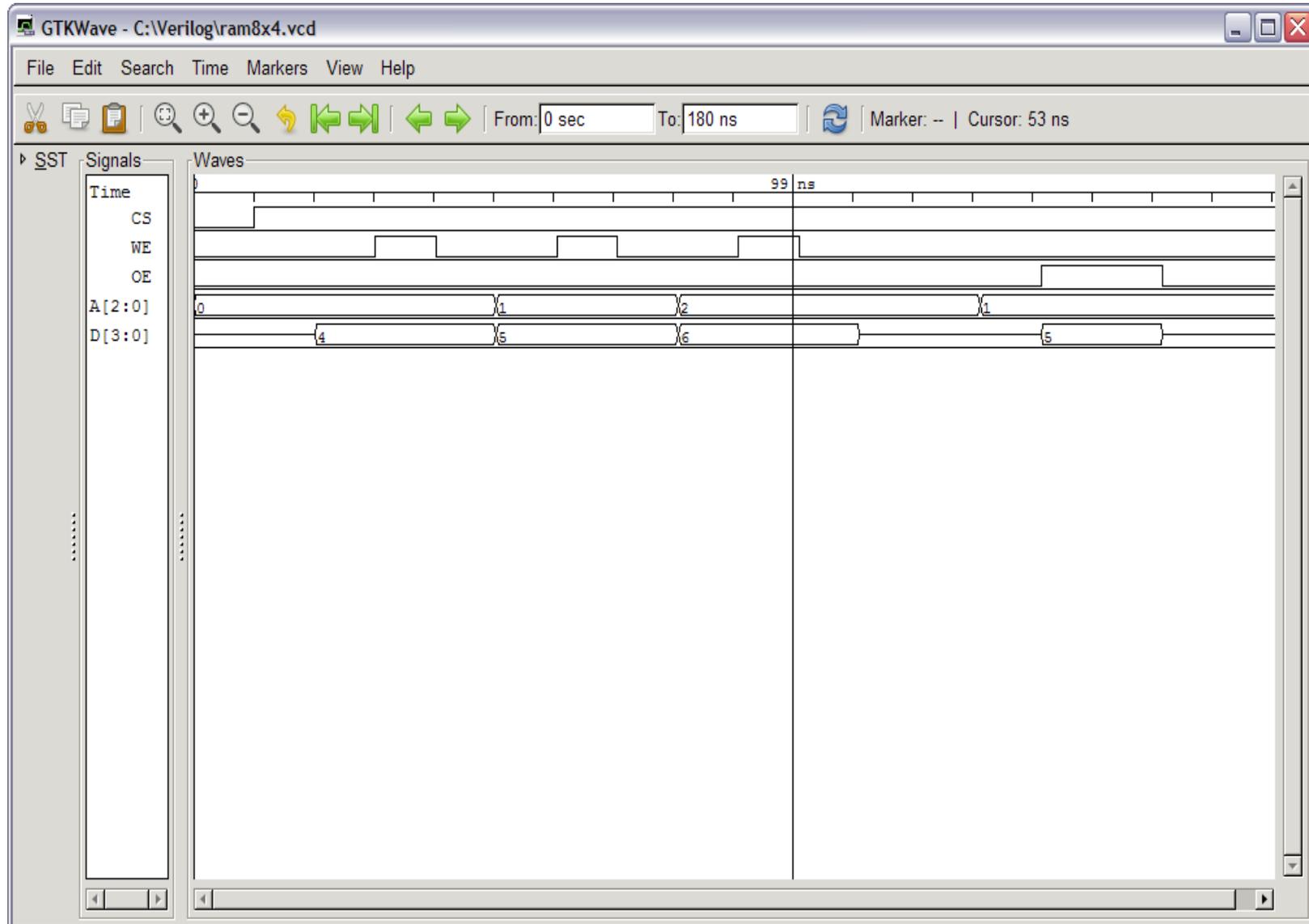
# Descripción Verilog de RAM

## Descripción de RAM en lenguaje Verilog



```
module ram8x4(  
    input CS,  
    input WE,  
    input OE,  
    input [2:0] A,  
    inout [3:0] D  
);  
  
reg [3:0] mem [7:0];  
  
always @(CS, WE, A, D)  
    if (CS && WE)  
        mem[A] = D;  
  
assign D = (CS && !WE && OE) ?  
    mem[A] : 'hZ;  
  
endmodule // ram8x4
```

# Operación de memoria RAM



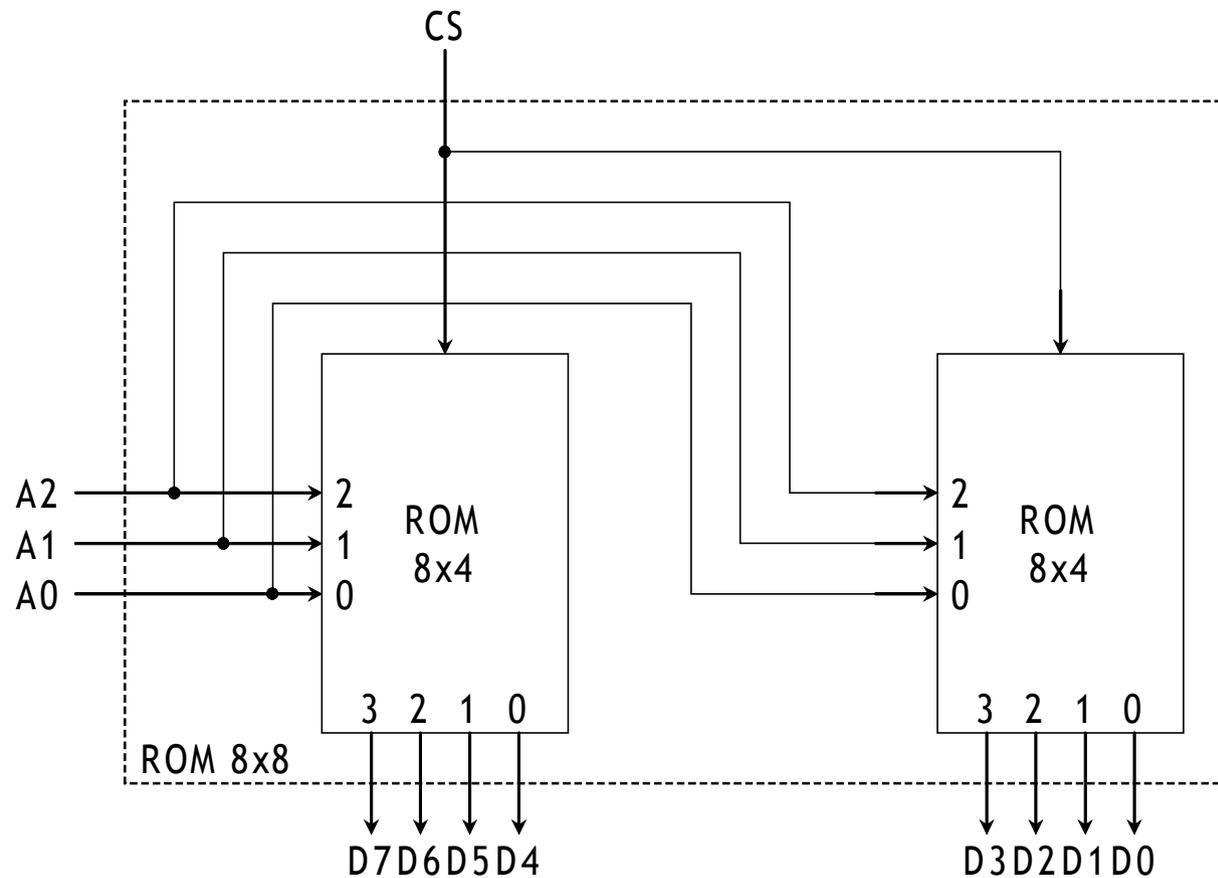
---

# Memorias y dispositivos programables

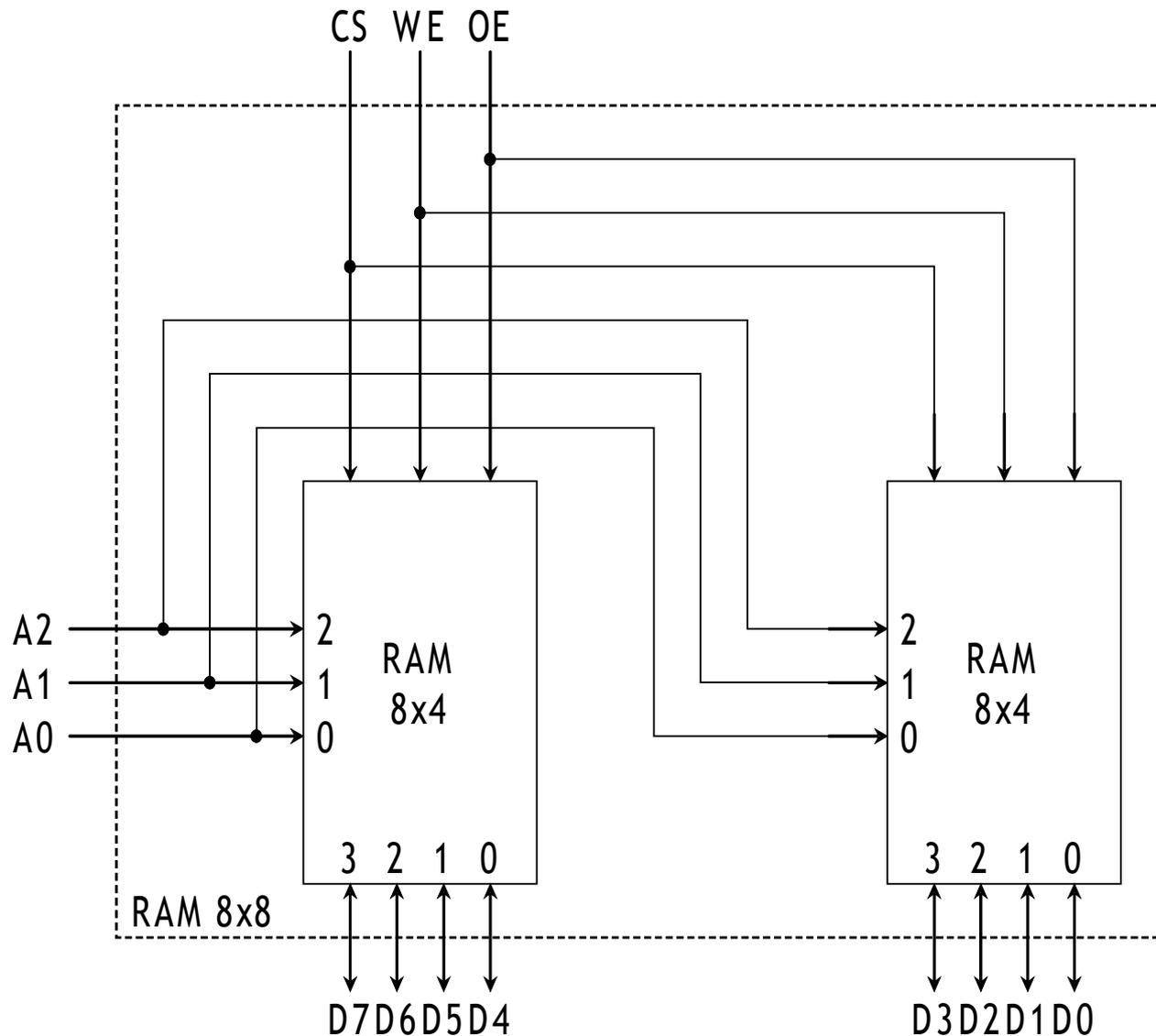
Expansión de memorias

# Expansión de longitud de palabra en memorias ROM

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

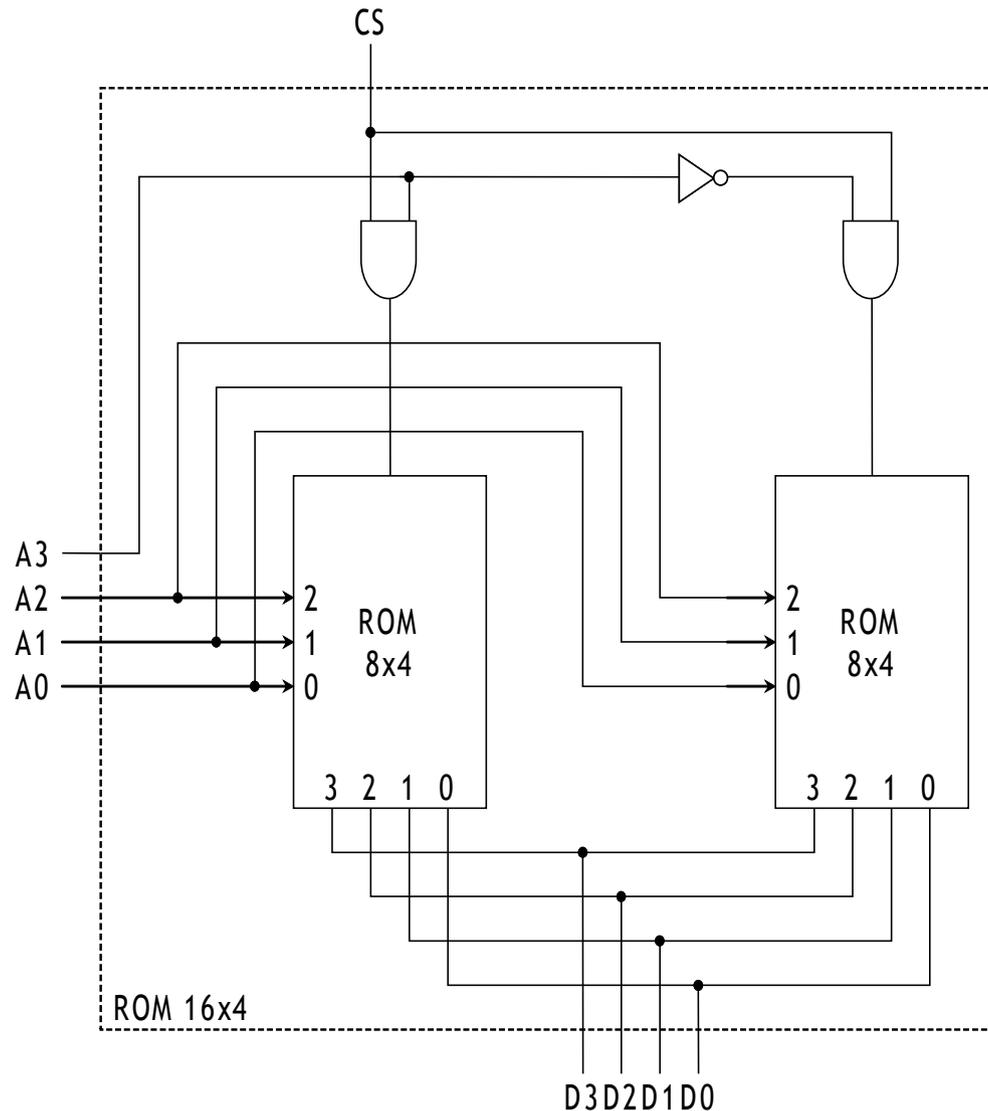


# Expansión de longitud de palabra en memorias RAM

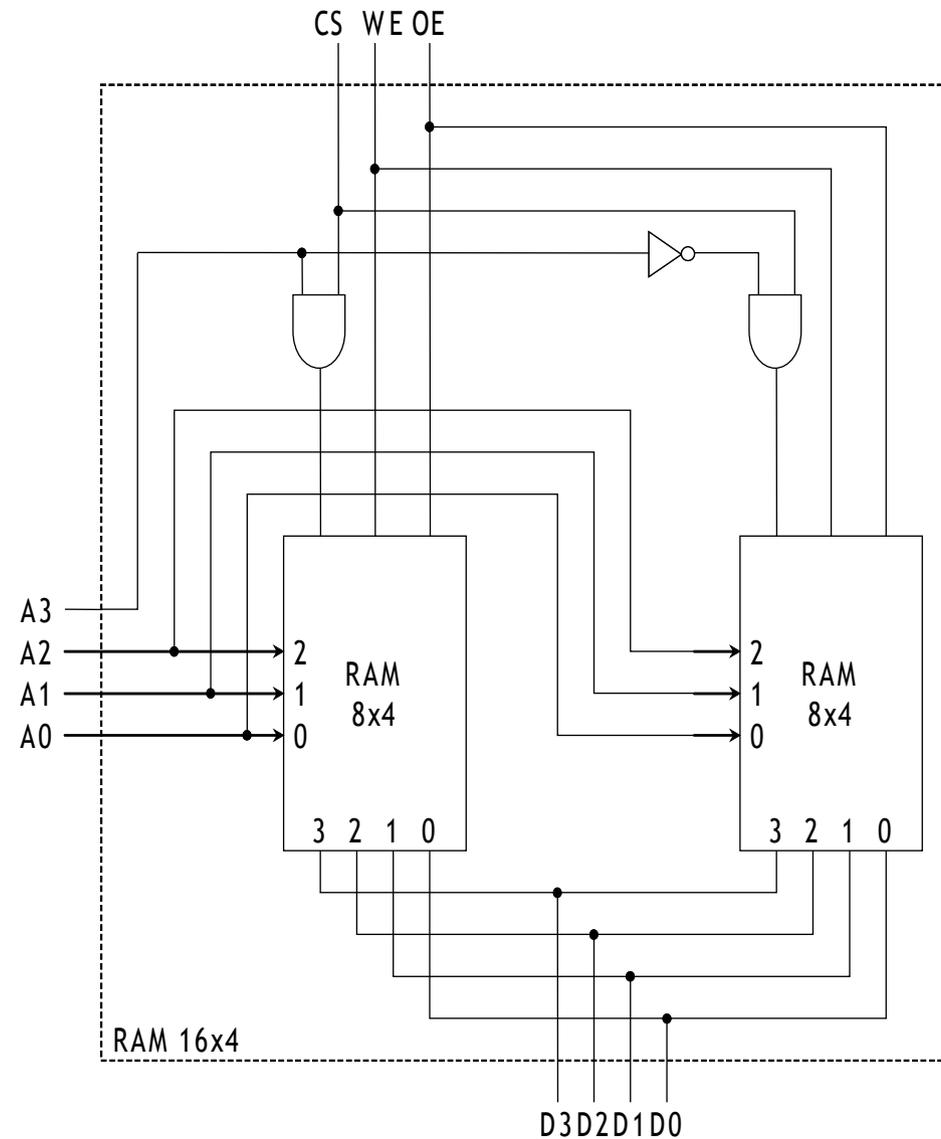


# Expansión de número de palabras en memorias ROM

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



# Expansión de número de palabras en memorias RAM



---

# Memorias y dispositivos programables

Tipos especiales de memorias

# Memorias secuenciales

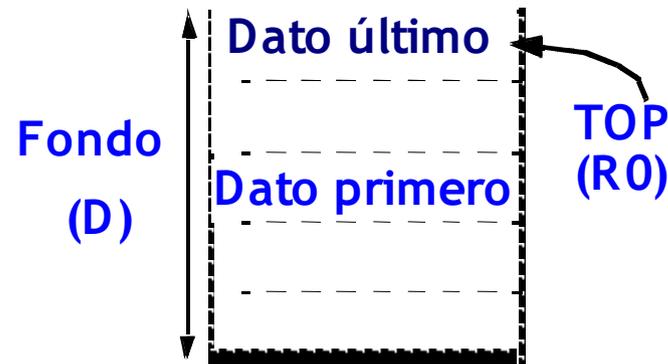
## Memorias LIFO (Last In-First Out)

### PUSH

Escribe nuevo dato

### PULL o POP

Leer /y extraer  
último dato



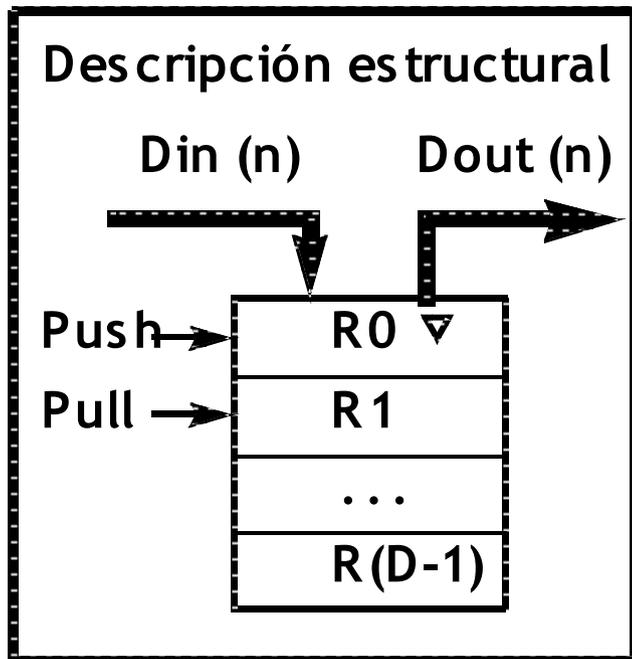
**PILA VACÍA:** Cuando no se ha escrito ningún dato

**PILA LLENA:** Cuando están escritos D datos

**PILA OCIOSA:** Cuando no hay Pull ni Push

# Memorias LIFO (Last In-First Out)

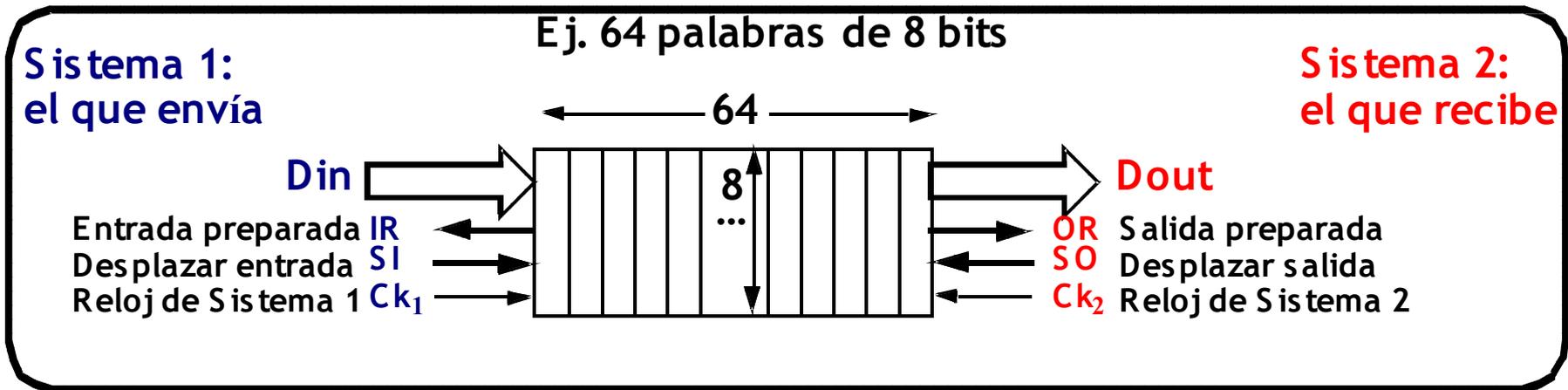
- *Push (Escritura)*: se escribe en la cabecera de la pila (siguiente posición libre).
- *Pull (Lectura)*: se lee el dato más nuevo y se libera la posición.



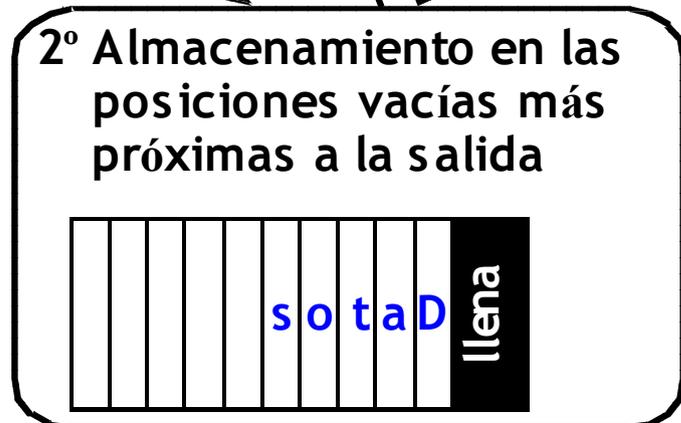
Descripción funcional

Push	Pull	$R_x \leftarrow$	Dout =
0	0	$R_x \leftarrow R_x$	Dout = HI
0	1	$R_x \leftarrow R_{(x+1)}; R_{(D-1)} \leftarrow 0$	Dout = [R0]
1	0	$R_x \leftarrow R_{(x-1)}; R_0 \leftarrow D_{in}$	Dout = HI
1	1	Prohibida	

# Memorias FIFO (First In-First Out)

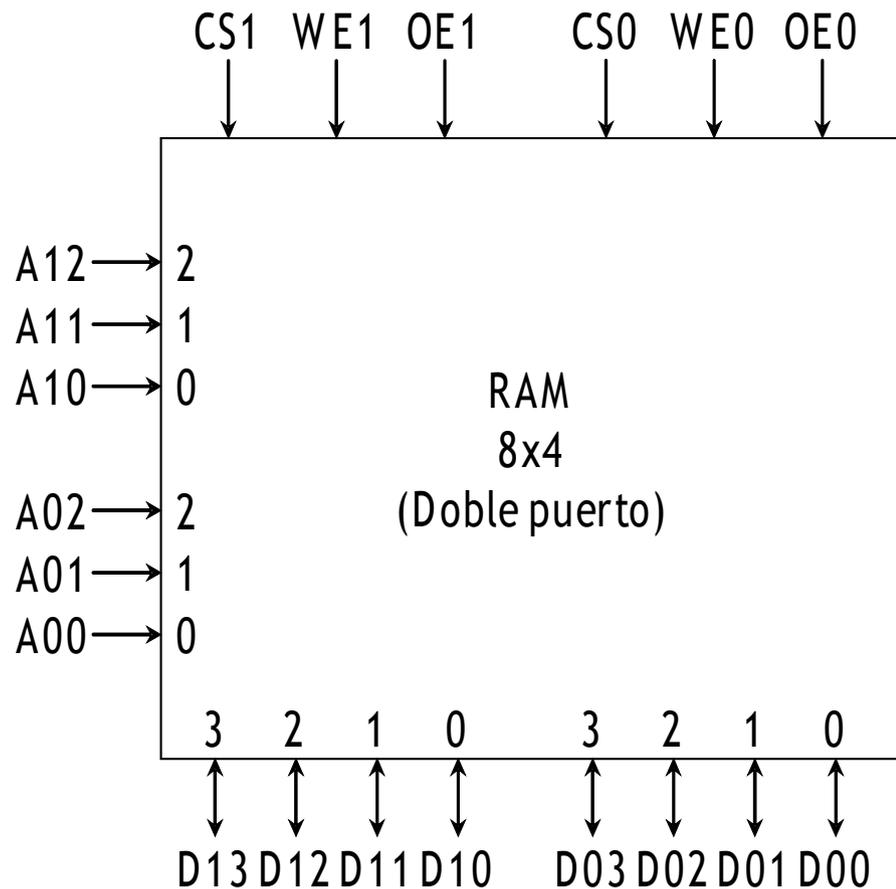


**1º IR activo:**  
Sistema 1 (con Ck<sub>1</sub>)  
SI activo y,  
secuencialmente,  
Din: D, a, t, o, s



**3º OR activo:**  
Sistema 2 (con Ck<sub>2</sub>)  
SO activo y,  
secuencialmente,  
Dout: D, a, t, o, s

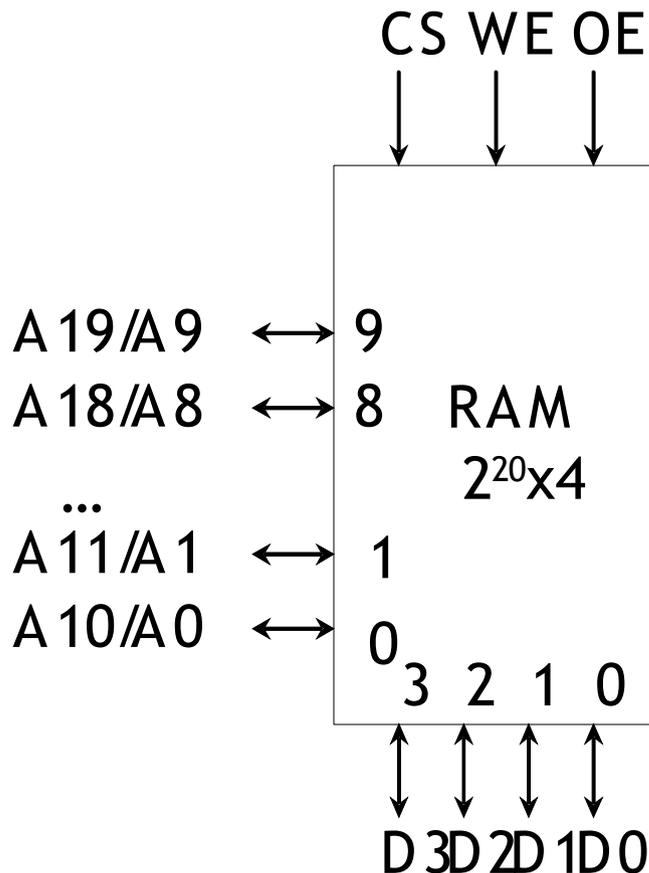
# Memorias de doble puerto



• Cuentan con 2 puertos independientes por lo que permiten simultanear:

- 2 Lecturas
- 2 Escrituras
- 1 Lectura + 1 Escritura

# Memorias con bus de direcciones multiplexado



- Se ahorran líneas de conexión utilizando un bus más estrecho que el necesario para suministrar la dirección.
- Son más lentas ya que hay que suministrar la dirección por partes.

# Memorias NVRAM (*Non-Volatile RAM*)

---

Pueden implementarse siguiendo varias estrategias diferentes:

SRAM + Pila de litio (configuración BIOS )

SRAM + Batería (videoconsolas portátiles)

RAM + EEPROM: ante un pulso de retención, el contenido de la RAM se vuelca en la EEPROM en paralelo (PDA).