
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

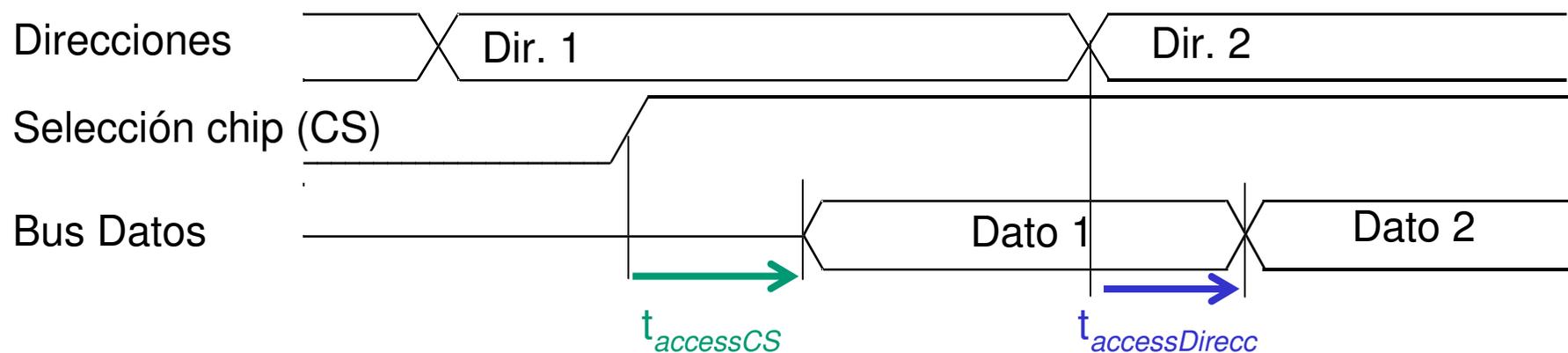
Clasificación de memorias

Velocidad y tiempo de acceso

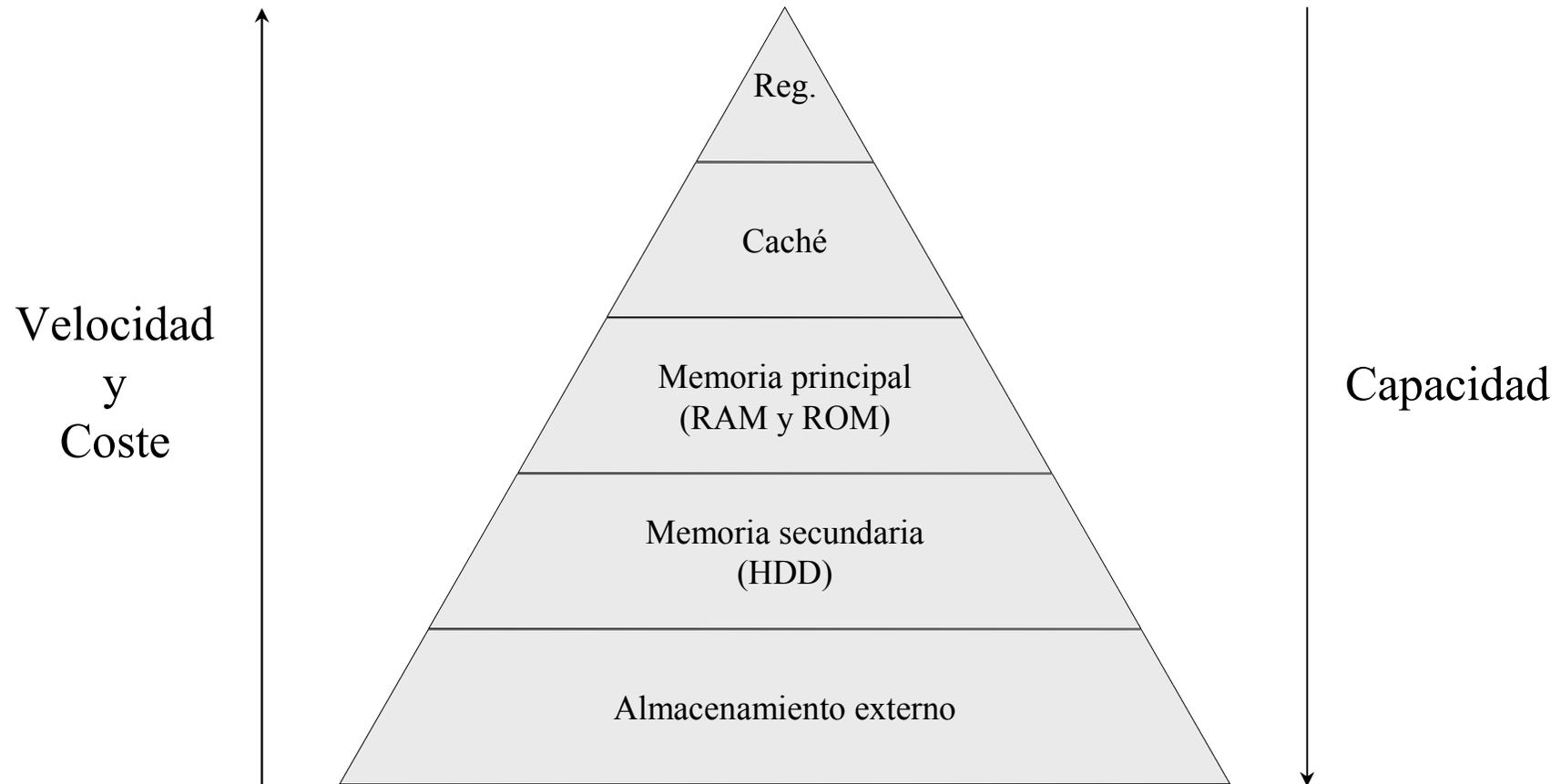
Velocidad de acceso: Cantidad de bits que se transfieren por segundo (bits por segundo)

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

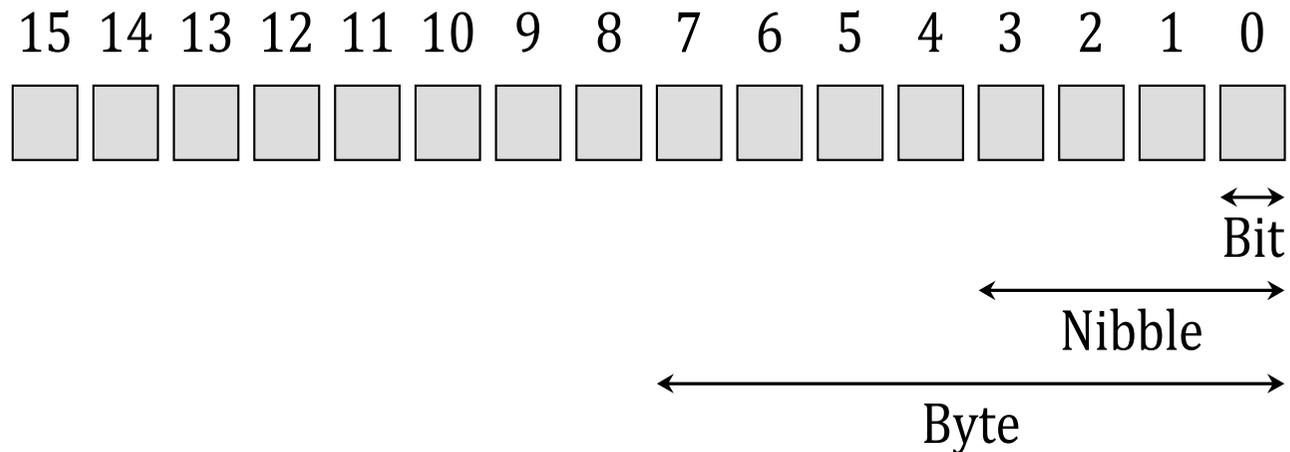
Matriz de memoria básica

- 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.

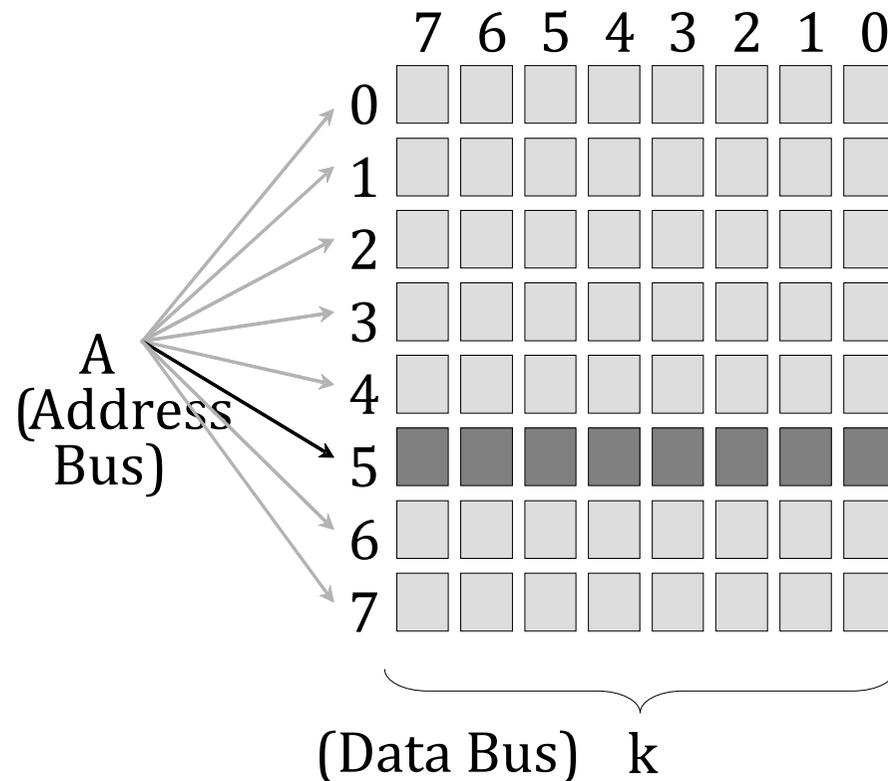
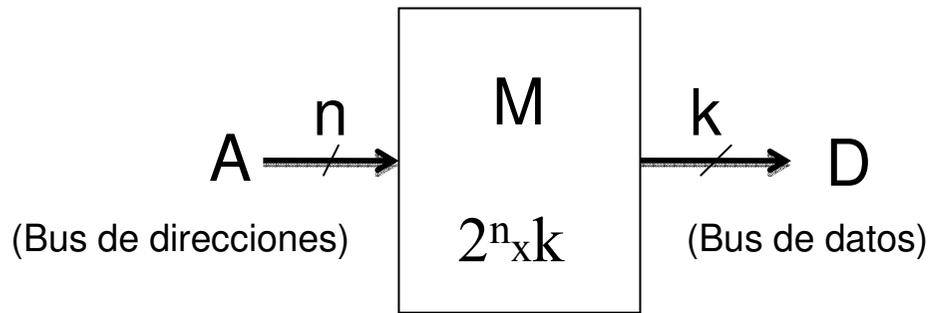
	7	6	5	4	3	2	1	0
0	<input type="checkbox"/>							
1	<input type="checkbox"/>							
2	<input type="checkbox"/>							
3	<input type="checkbox"/>							
4	<input type="checkbox"/>							
5	<input type="checkbox"/>							
6	<input type="checkbox"/>							
7	<input type="checkbox"/>							
8	<input type="checkbox"/>							
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:

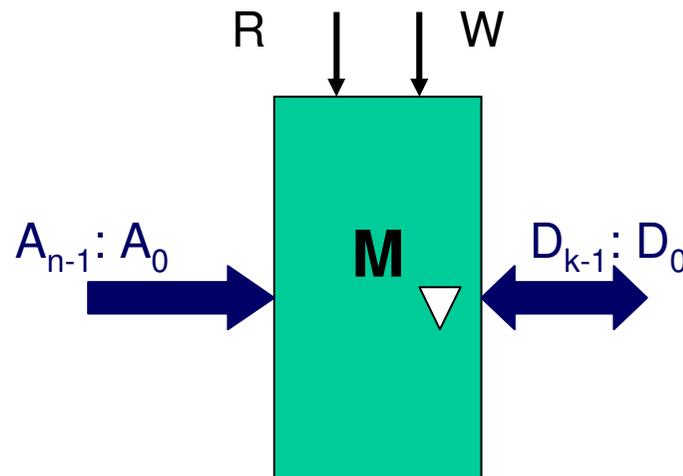
Escritura (*write*): permite almacenar una palabra en una determinada dirección de la memoria.

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.

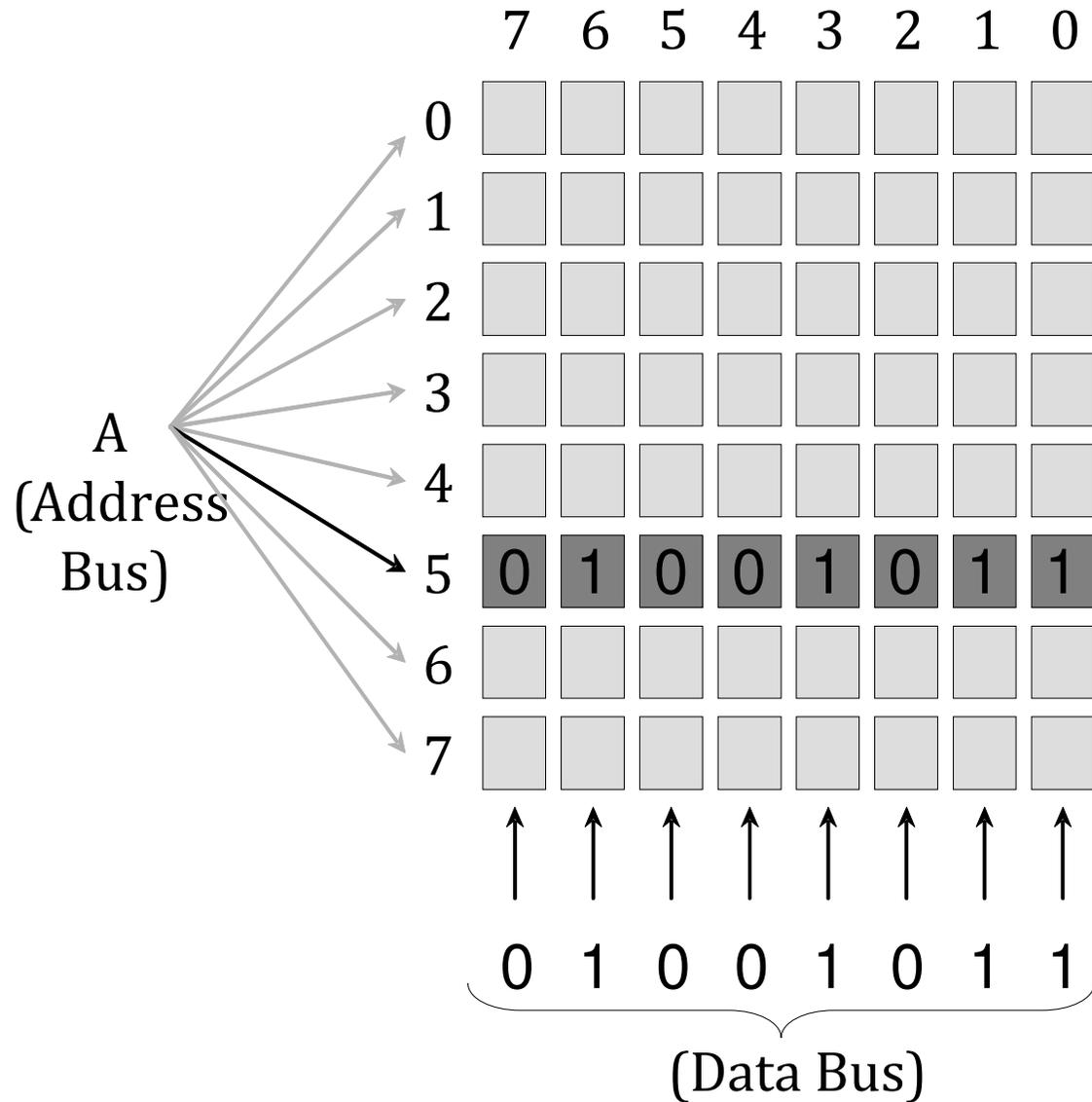


Operación de escritura

Data Bus \longrightarrow M(n)

En el ejemplo:

01001011 \longrightarrow M(101)

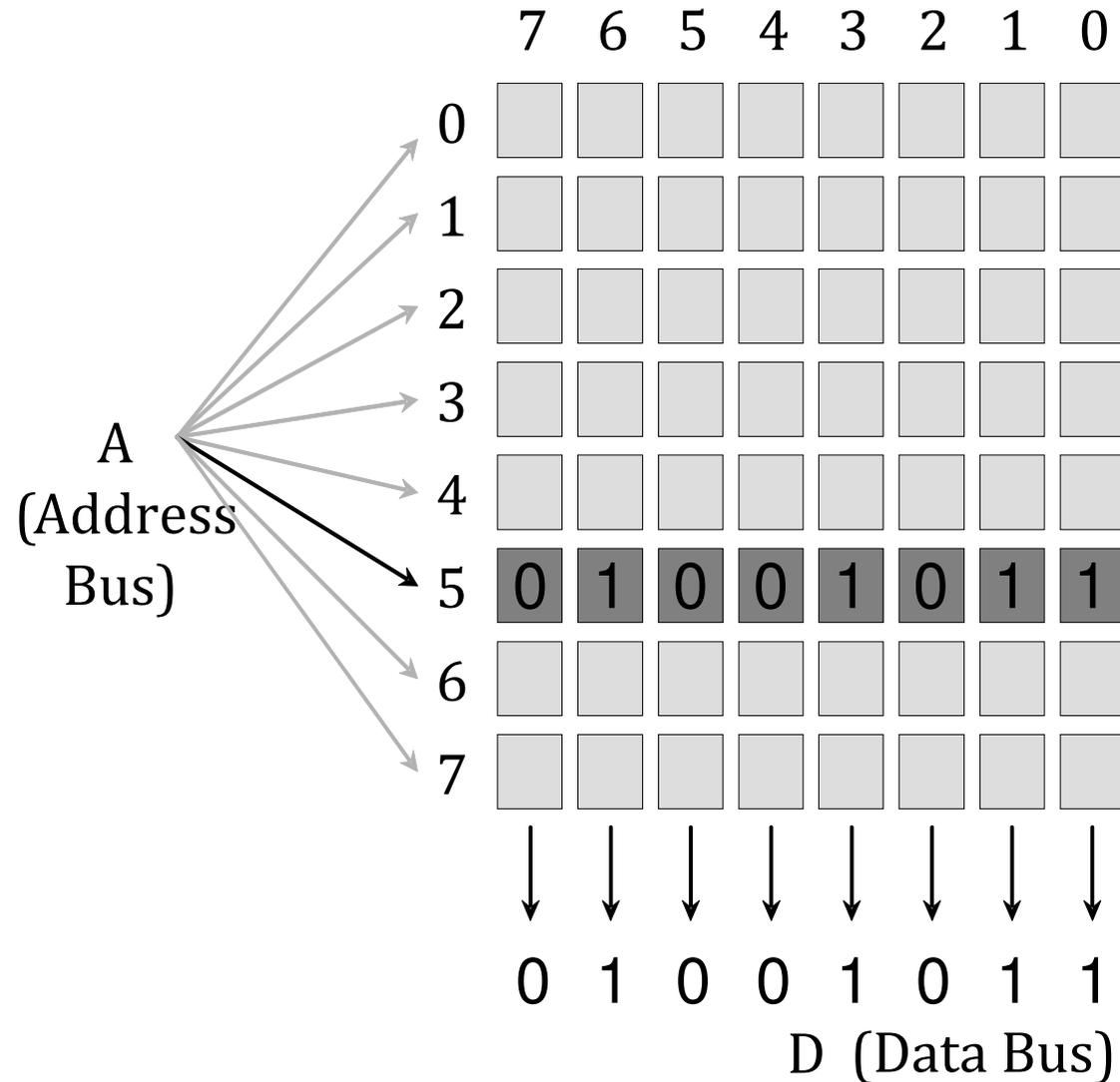


Operación de lectura

$D = [M(n)]$

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

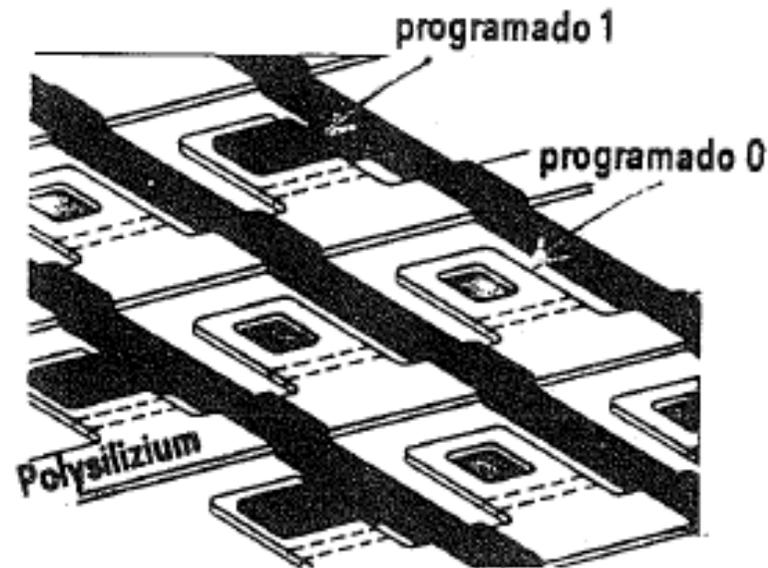
Memorias de sólo lectura (ROM)

Tecnologías de la ROM

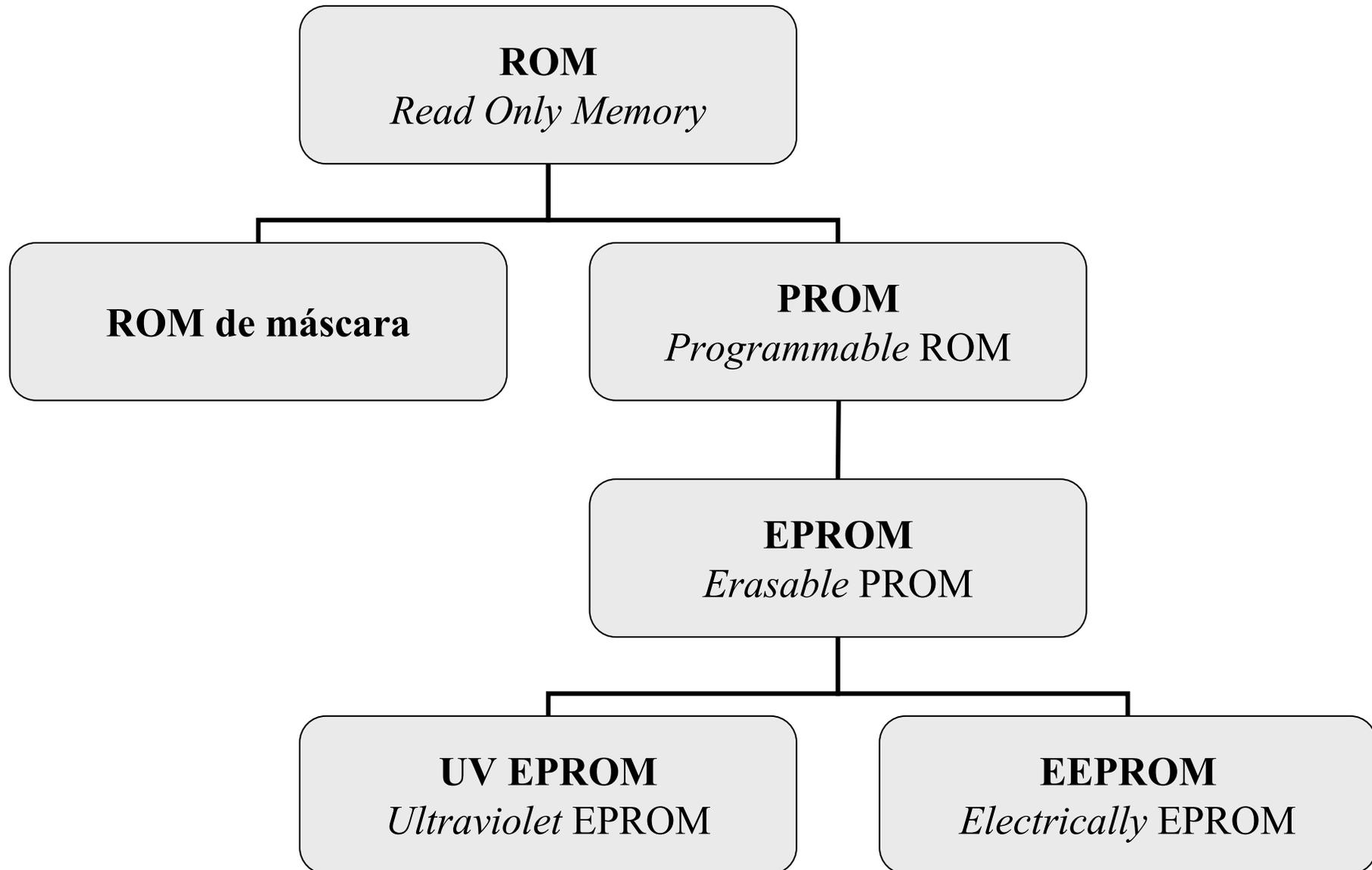
- TTL y MOS
- Con metalización
- Con fusibles

ROM
Módulos programados
con máscaras
por el fabricante;

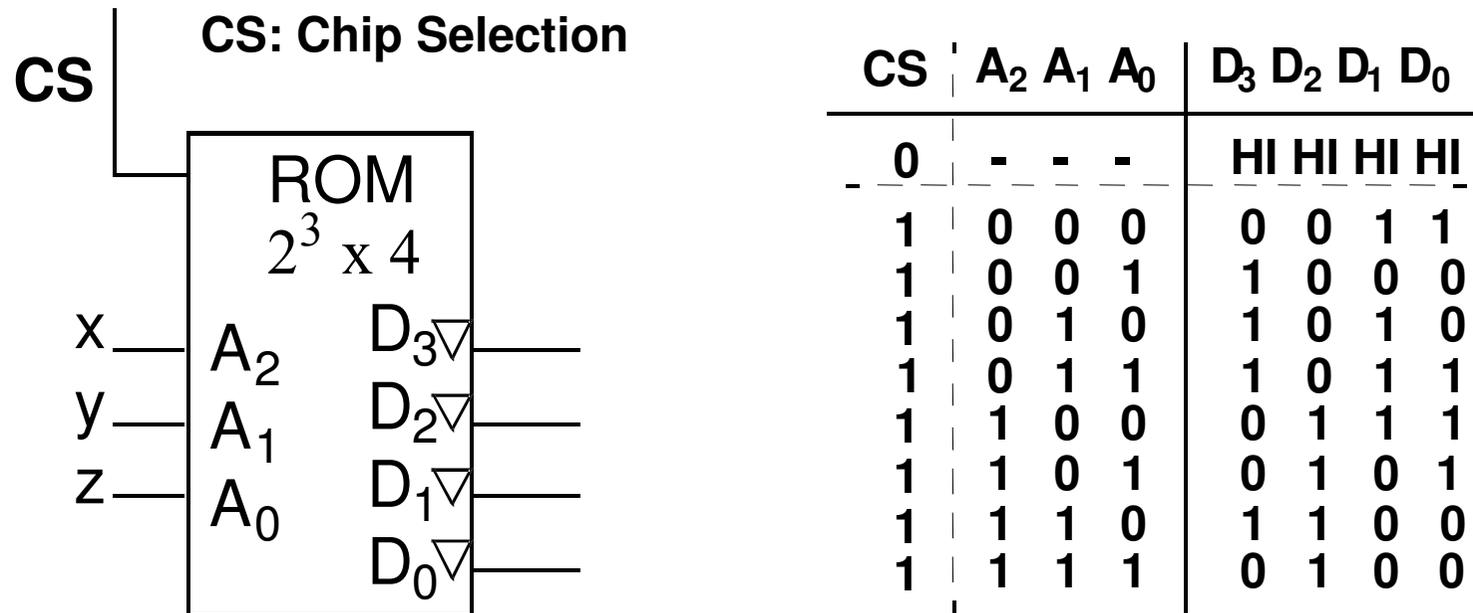
sin posibilidad
de modificar la información



Familia de memorias ROM



Memoria ROM

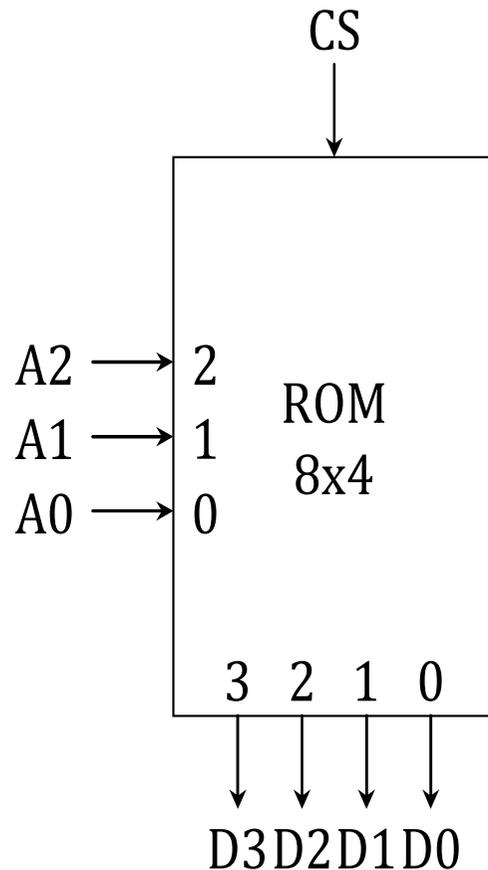


En cada columna se realiza una función (cuando CS=1). P. ej.

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

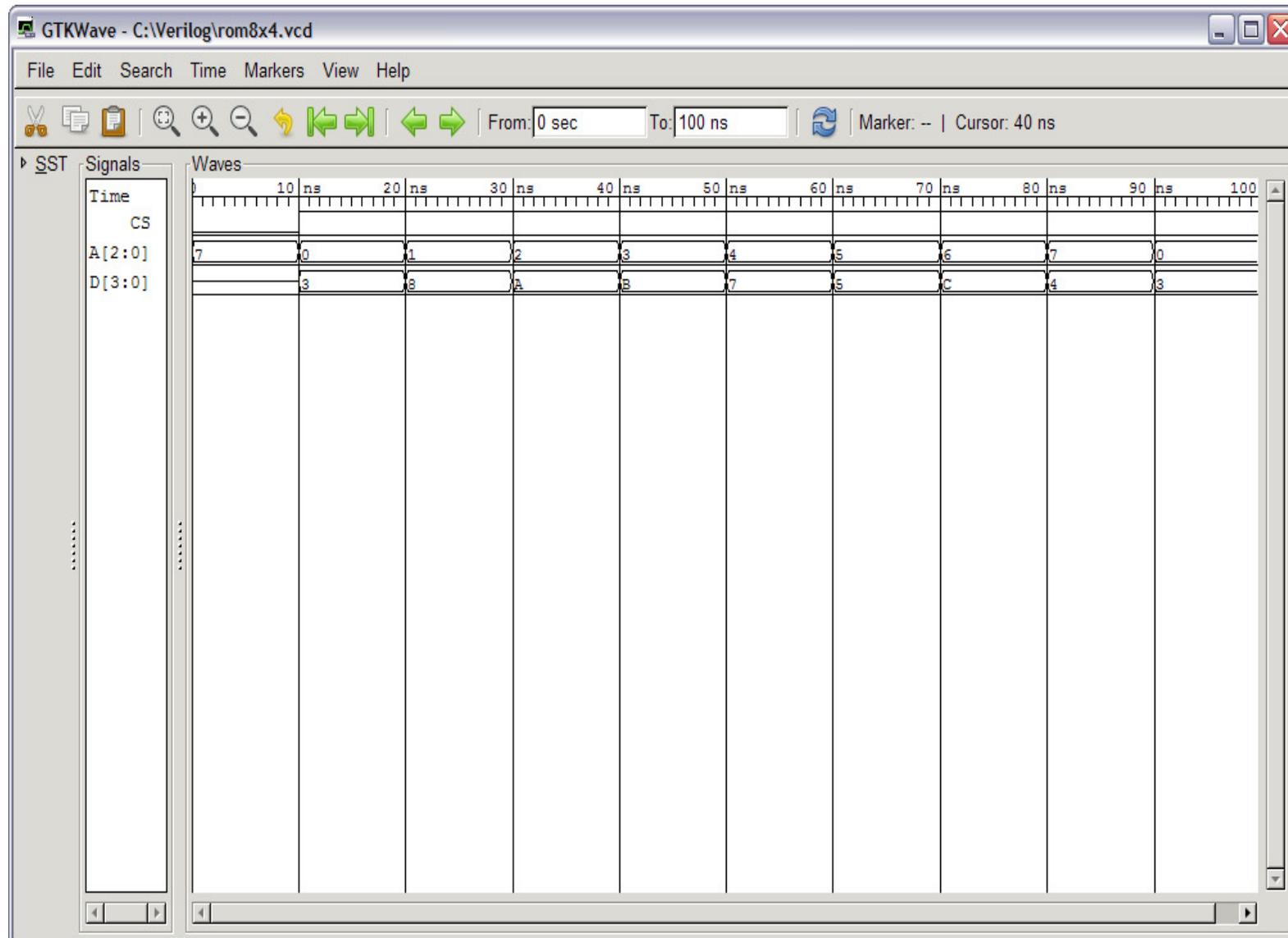
Memoria ROM

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;  
    endmodule // rom8x4
```

Operación de memoria ROM



Memorias y dispositivos programables

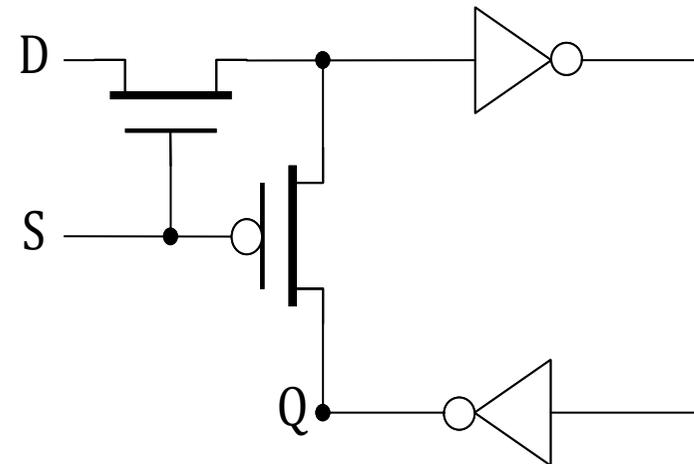
Memorias de acceso aleatorio (RAM)

Familia de memorias RAM

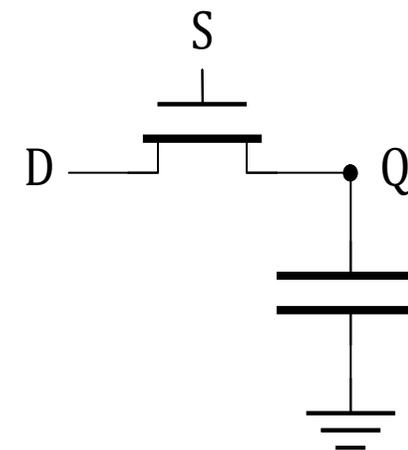
• Las dos categorías principales de memorias RAM son:

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

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



Celda de SRAM

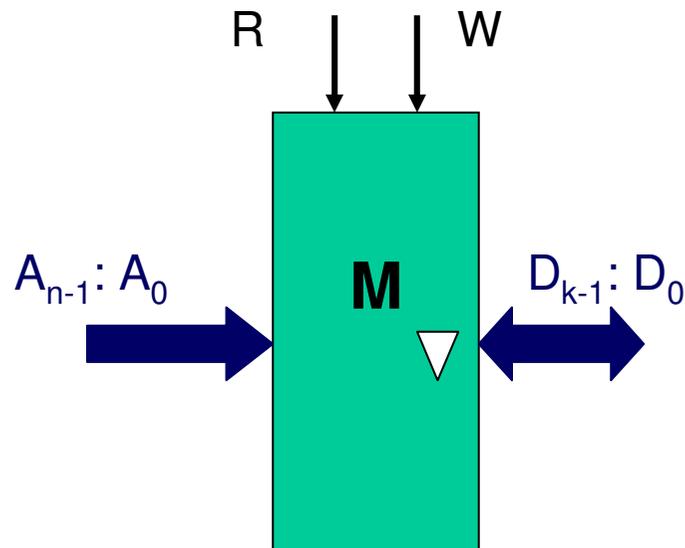


Celda de DRAM

Descripción de memoria RAM

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

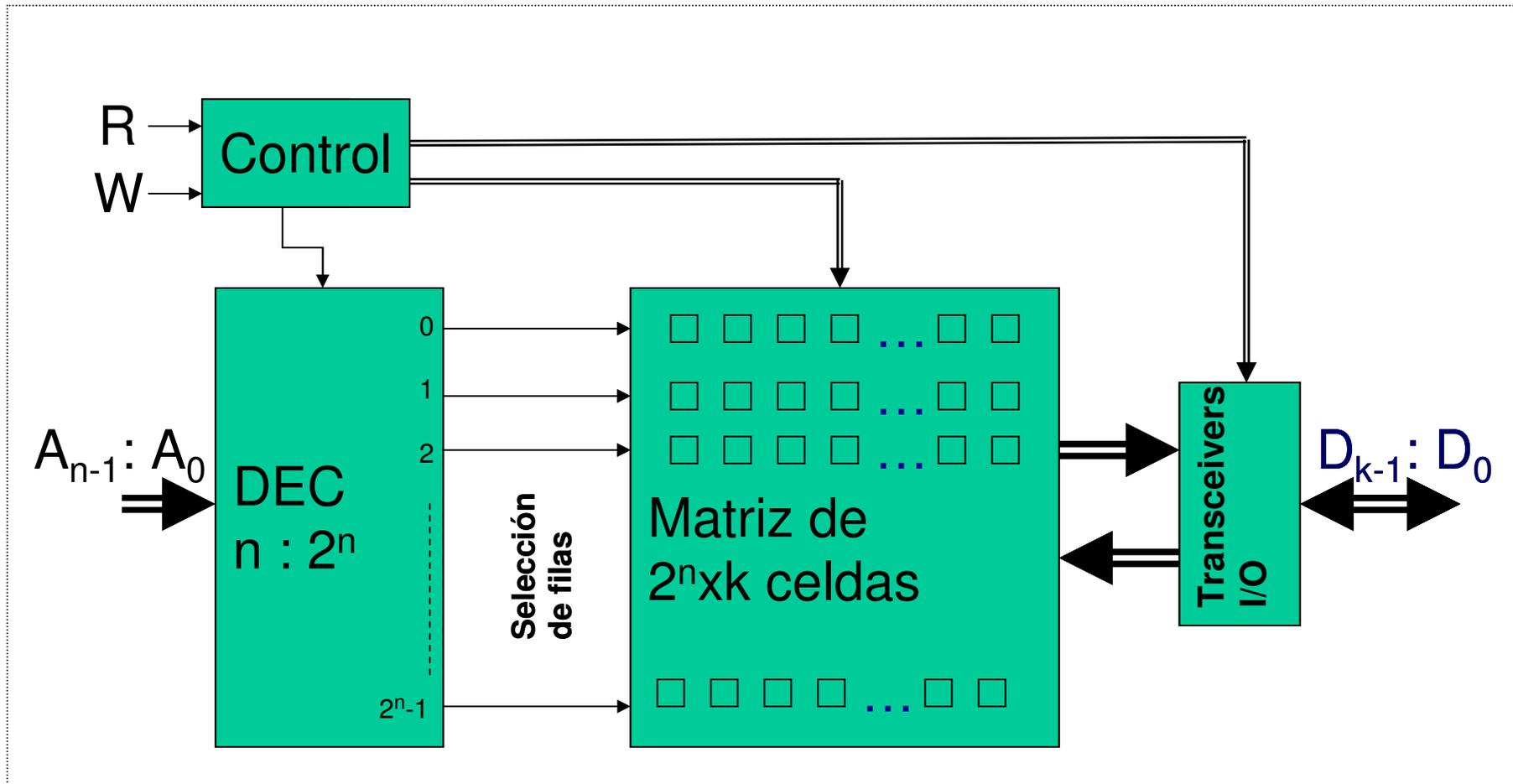
Descripción de RAM $2^n \times k$ con líneas de datos bidireccionales:



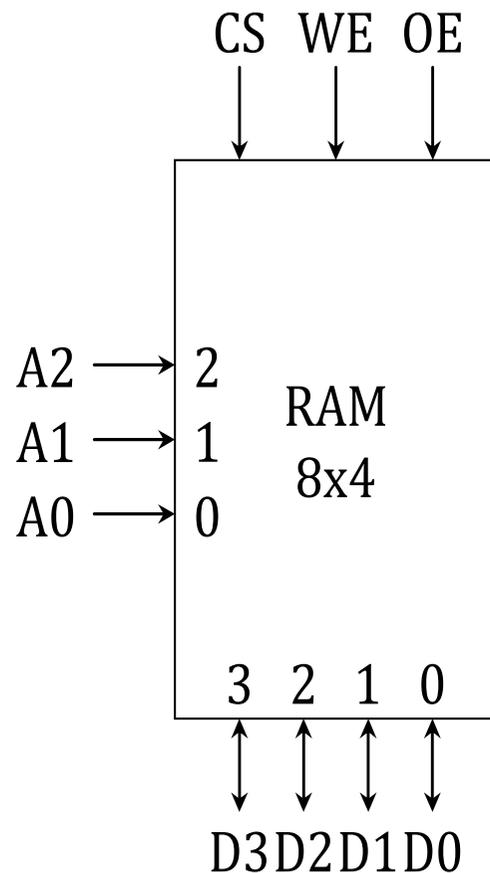
RW	$M \leftarrow$	$D =$
00	$M \leftarrow M$	HI
01	$M(A) \leftarrow D$	[D in]
10	$M \leftarrow M$	$D = M(A)$
11	Prohibido	

Diseño de memoria RAM

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



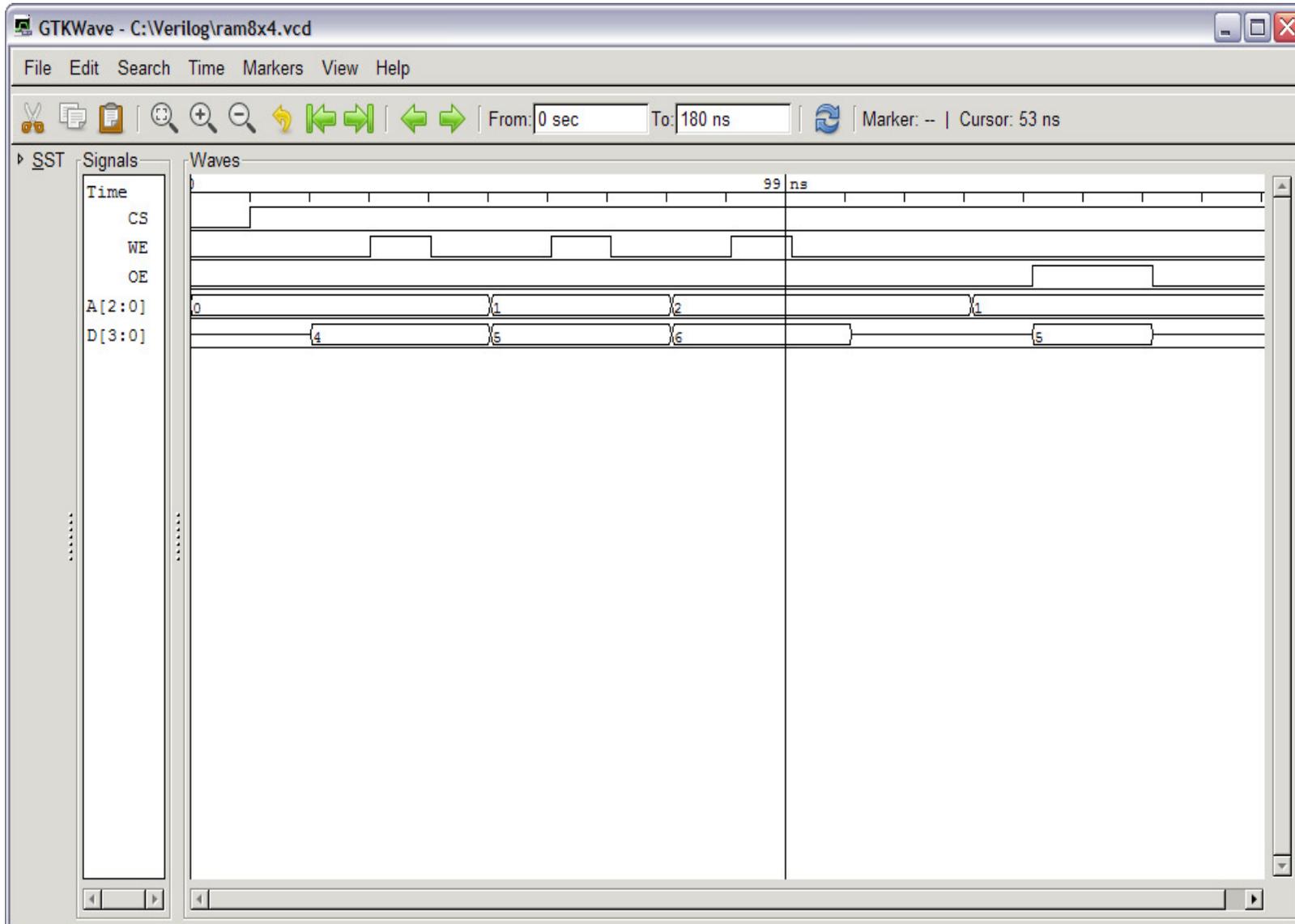
Descripción Verilog de RAM



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

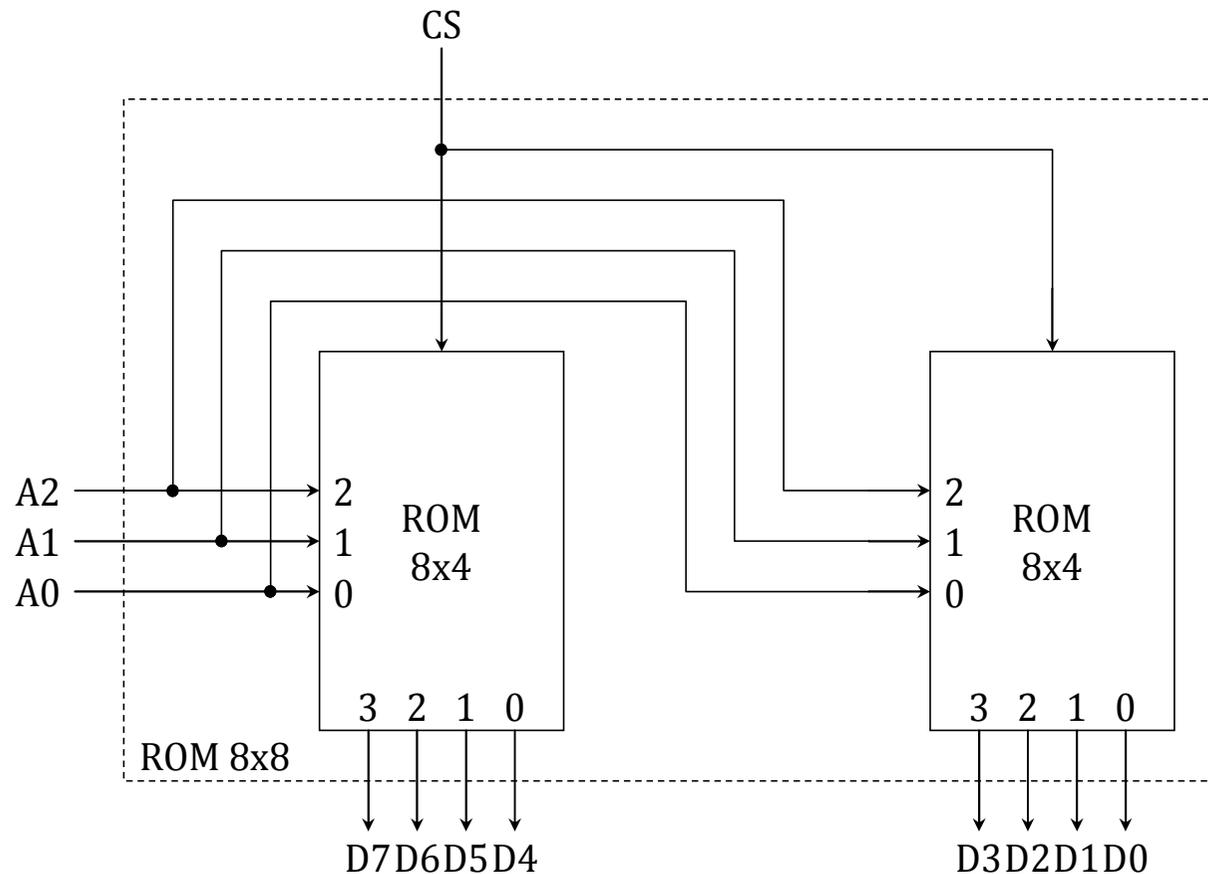


Memorias y dispositivos programables

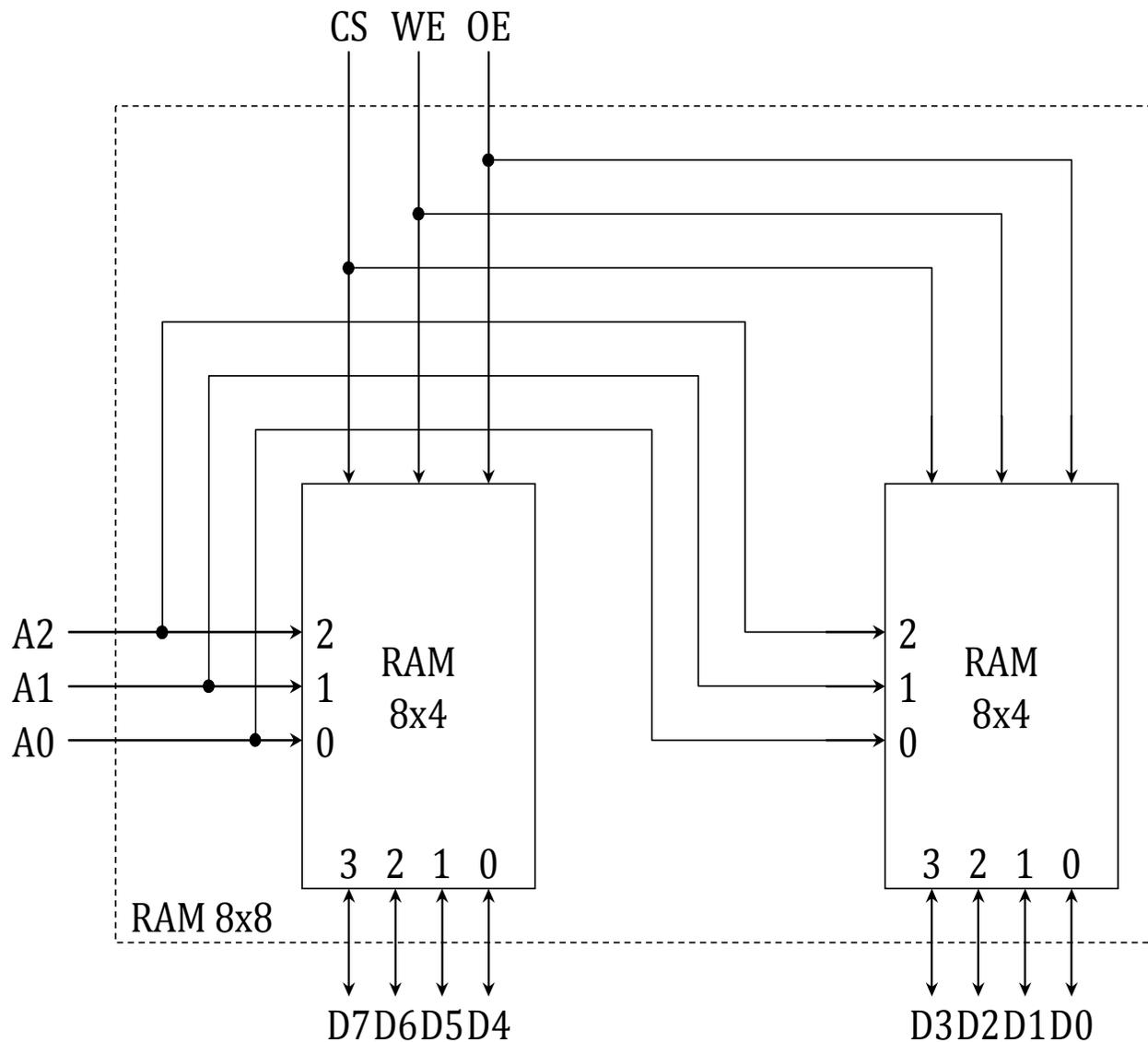
Expansión de memorias

Expansión de longitud de palabra en memorias ROM

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

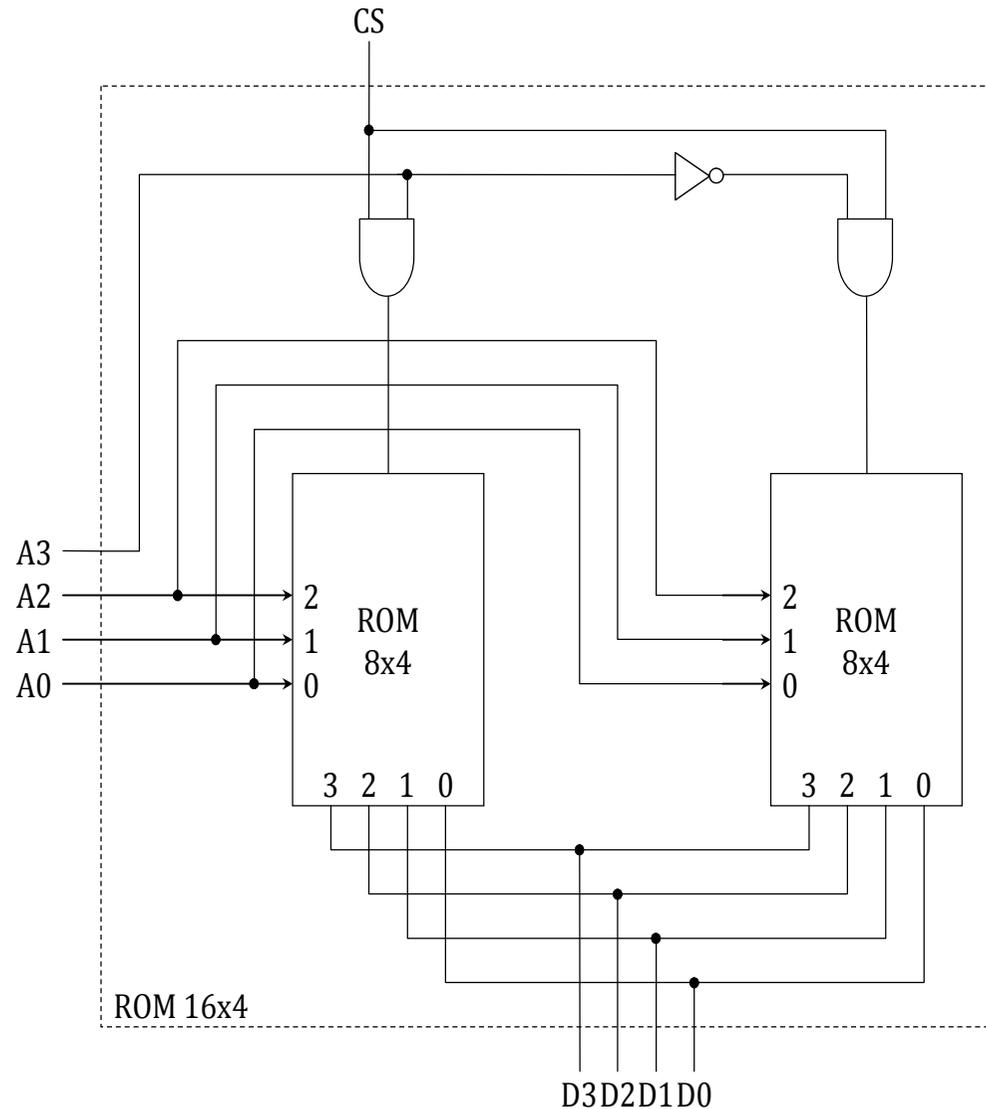


Expansión de longitud de palabra en memorias RAM

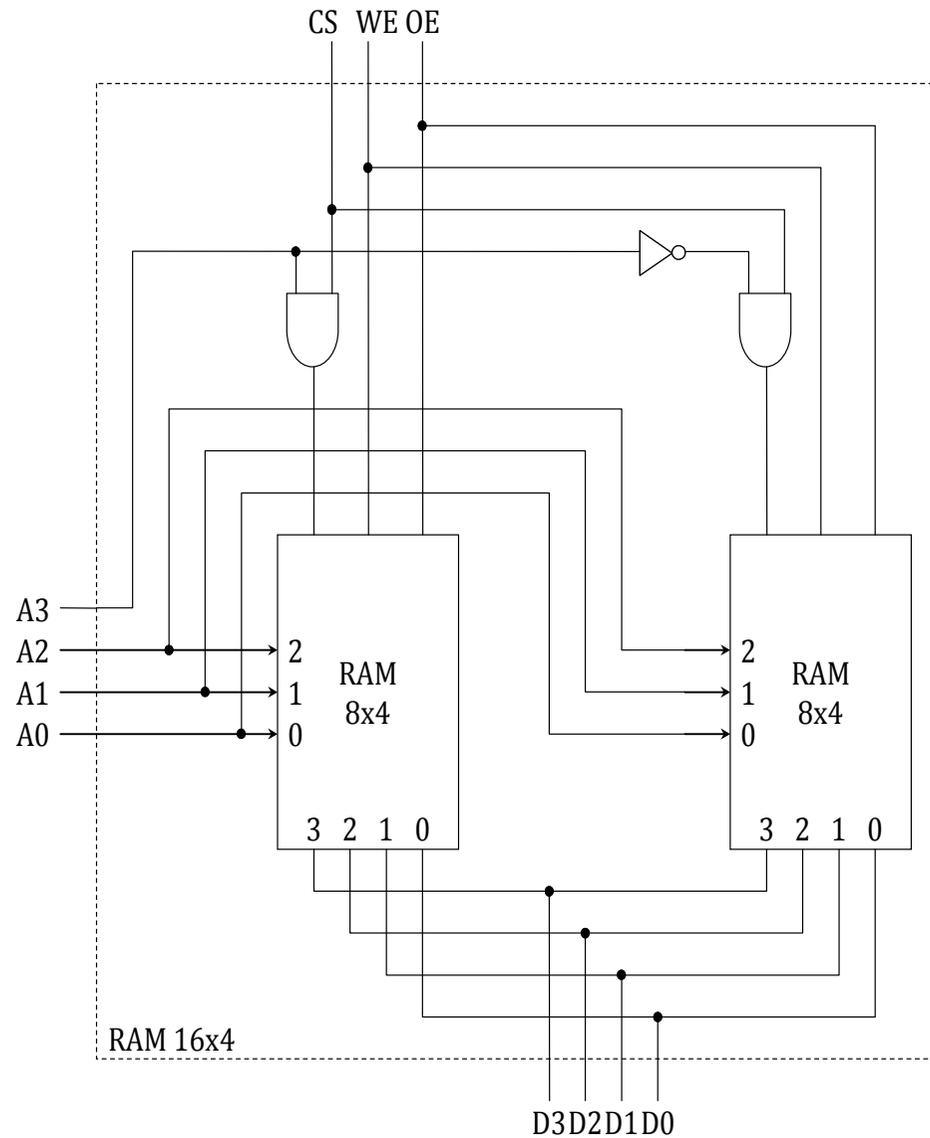


Expansión de número de palabras en memorias ROM

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



Expansión de número de palabras en memorias RAM

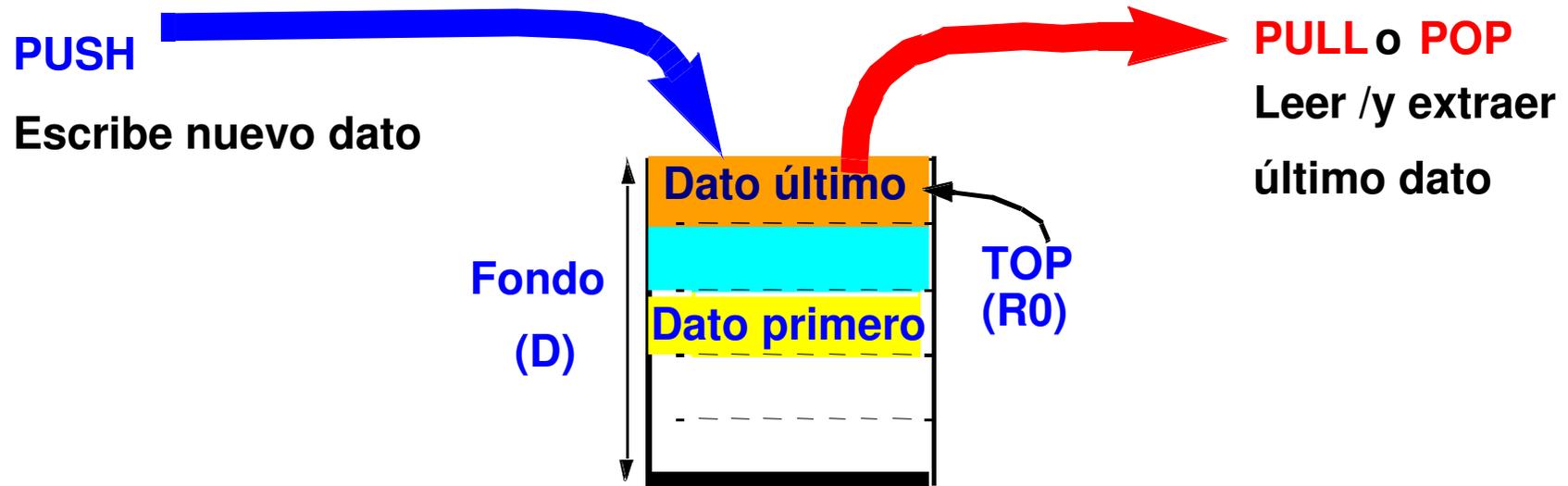


Memorias y dispositivos programables

Tipos especiales de memorias

Memorias secuenciales

Memorias LIFO (Last In-First Out)



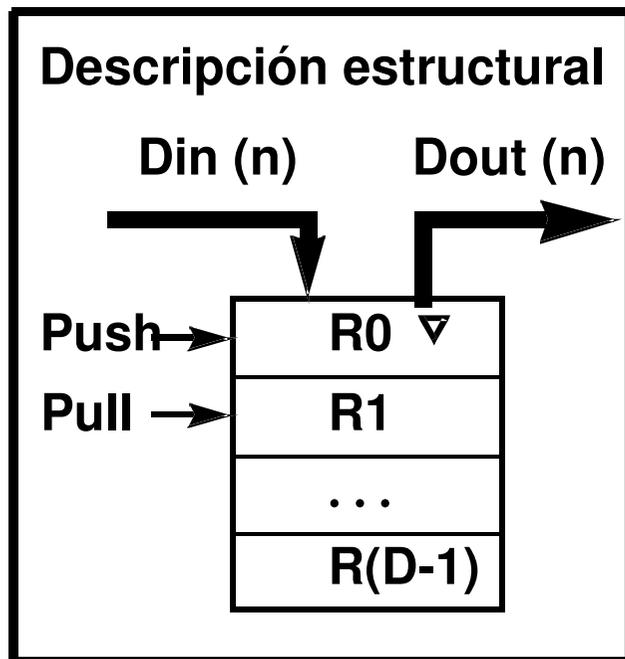
PILA VACÍA: Cuando no se ha escrito ningún dato

PILA LLENA: Cuando están escritos D datos

PILA OCIOSA: Cuando no hay Pull ni Push

Memorias LIFO (Last In-First Out)

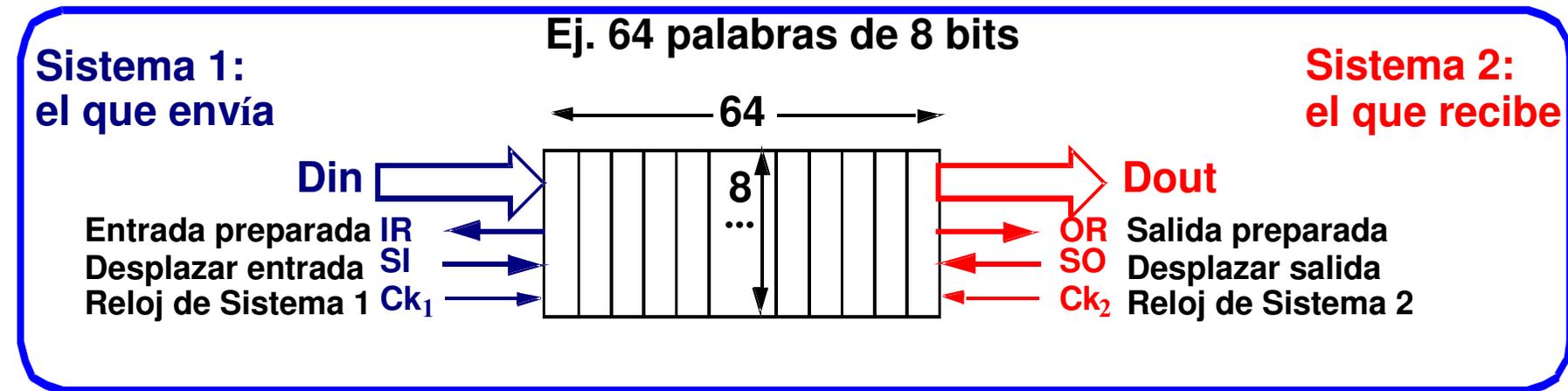
- *Push (Escritura)*: se escribe en la cabecera de la pila (siguiente posición libre).
- *Pull (Lectura)*: se lee el dato más nuevo y se libera la posición.



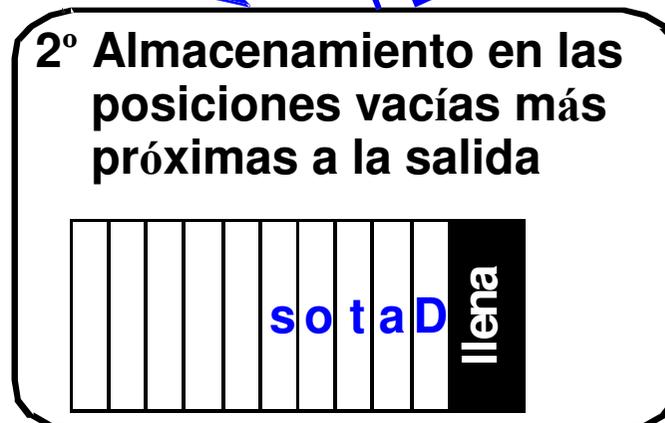
Descripción funcional

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)

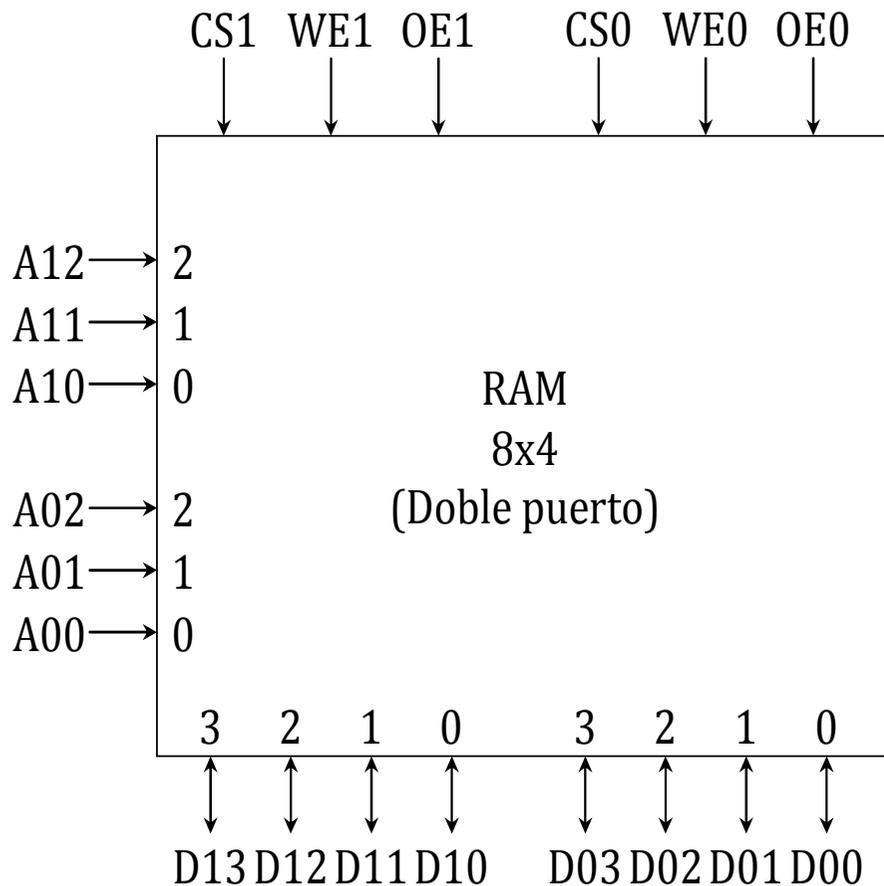


1º IR activo:
Sistema 1 (con Ck₁)
SI activo y,
secuencialmente,
Din: D, a, t, o, s



3º OR activo:
Sistema 2 (con Ck₂)
SO activo y,
secuencialmente,
Dout: D, a, t, o, s

Memorias de doble puerto



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

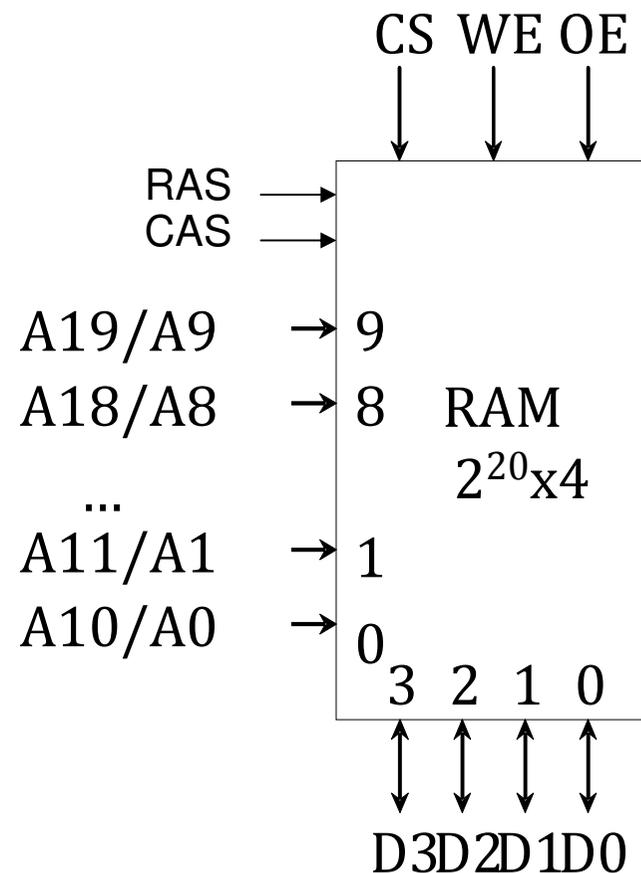
- 2 Lecturas

- 2 Escrituras

- 1 Lectura + 1 Escritura

- Precisan prevenir accesos simultáneos a la misma posición

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:

– Líneas adicionales de RAS y CAS

Memorias NVRAM (*Non-Volatile RAM*)

Pueden implementarse siguiendo varias estrategias diferentes:

SRAM + Pila de litio (configuración BIOS)

SRAM + Batería (videoconsolas portátiles)

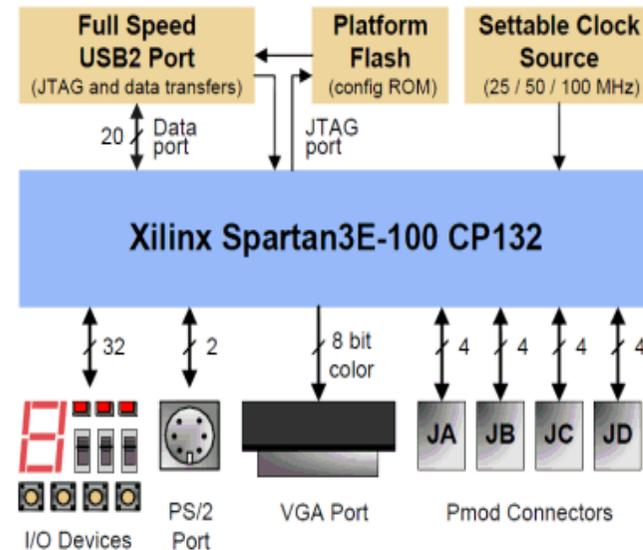
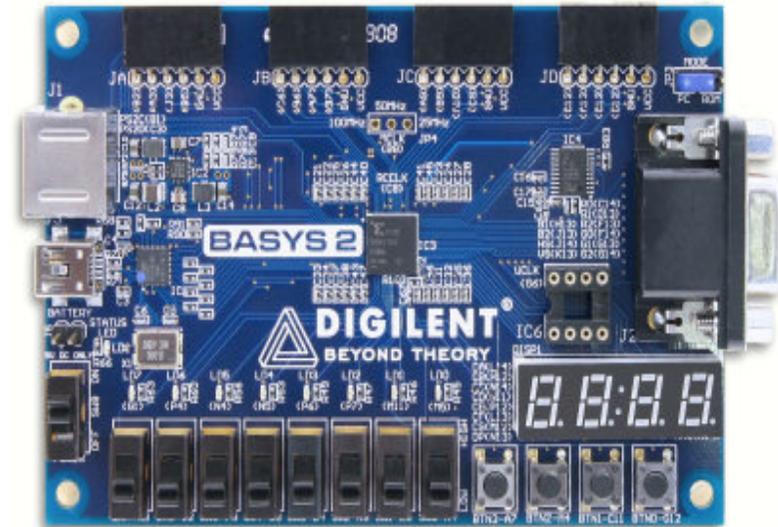
RAM + EEPROM: ante un pulso de retención, el contenido de la RAM se vuelca en la EEPROM en paralelo (PDA).

Memorias y dispositivos programables

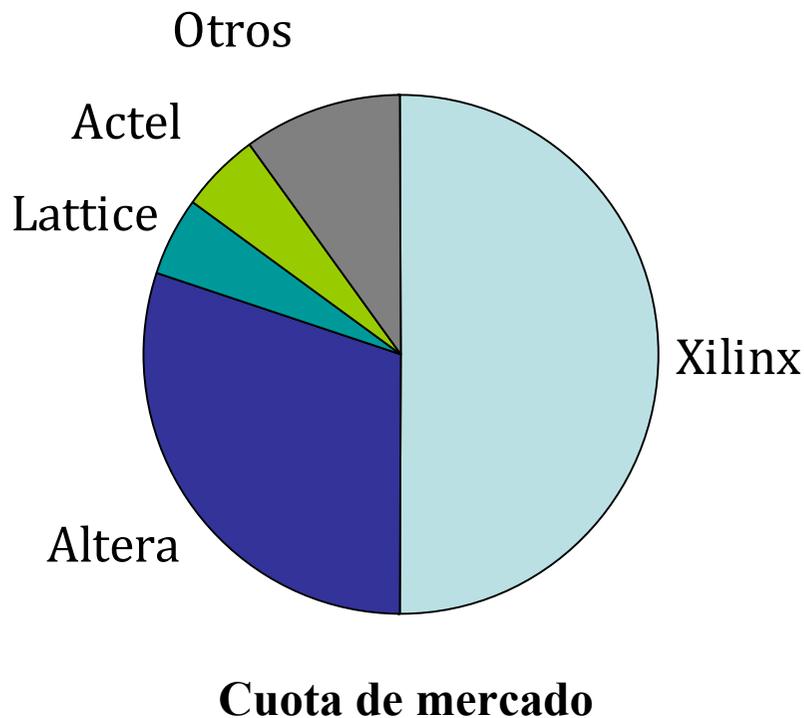
Arquitectura básica de una FPGA

Definición de FPGA

- Una FPGA
(*Field-Programmable Gate Array*)
es un dispositivo programable:
 - 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

Recursos internos de una FPGA

En general, una FPGA contiene los siguientes recursos internos:

Recursos lógicos:

Slices, agrupados en CLB (*Configurable Logic Blocks*).

Memoria BRAM (*Block RAM*).

Multiplicadores empotrados.

Recursos de interconexión:

Interconexión programable.

Bloques de entrada/salida IOB (*Input/Output Blocks*).

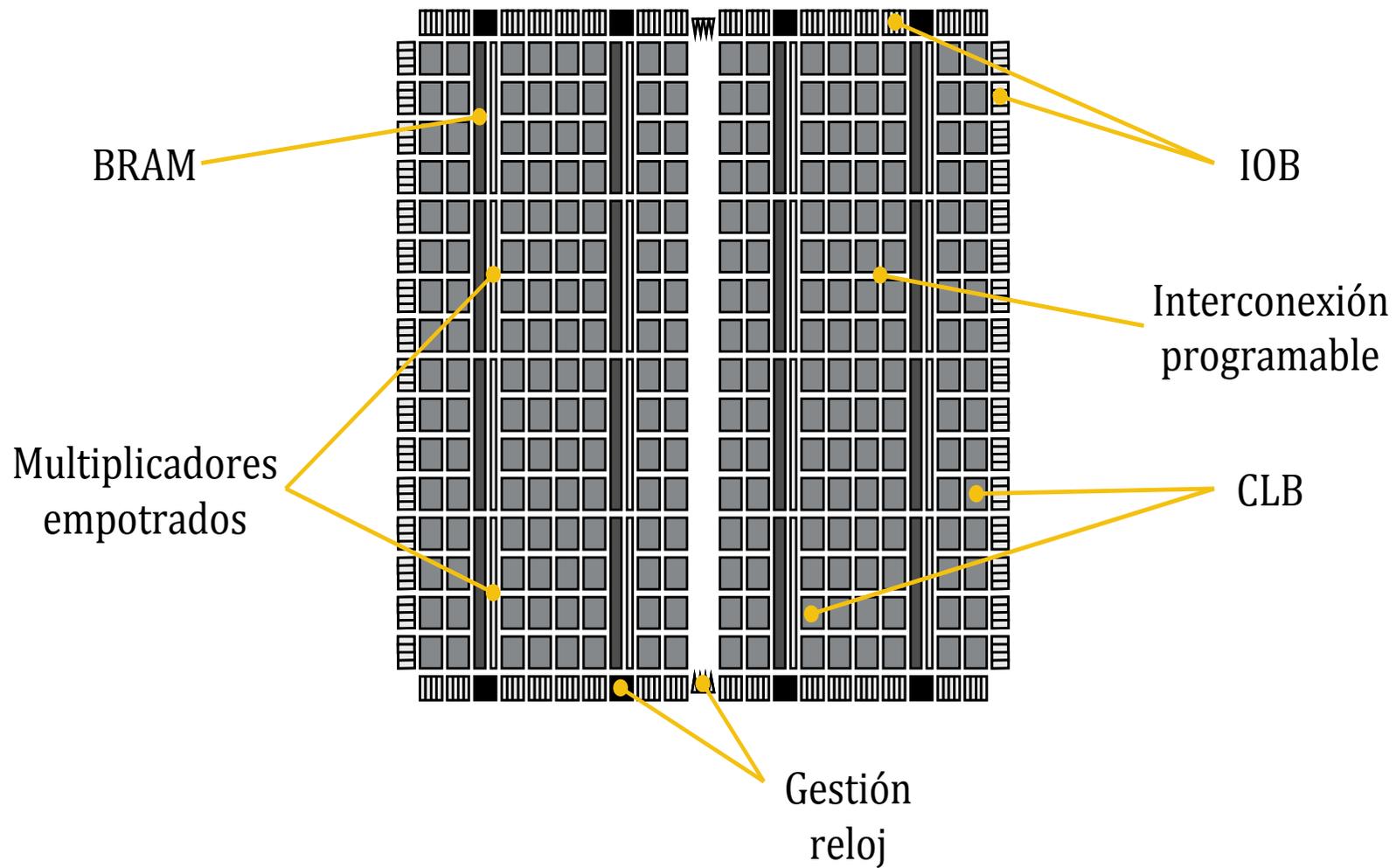
Otros recursos:

Búferes de reloj.

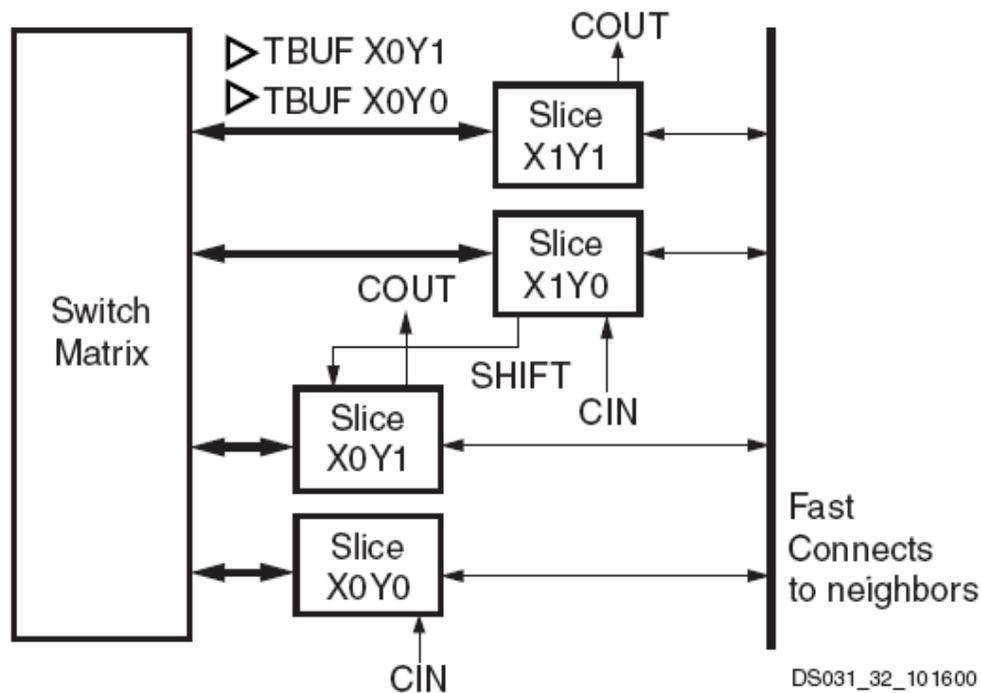
Lógica de escaneo de pines (*boundary scan logic*) normalmente mediante conexión JTAG (*Join Test Action Group*).

En las siguientes diapositivas se ilustra la estructura de la Virtex-II de Xilinx.

Estructura general de la FPGA modelo Virtex-II de Xilinx



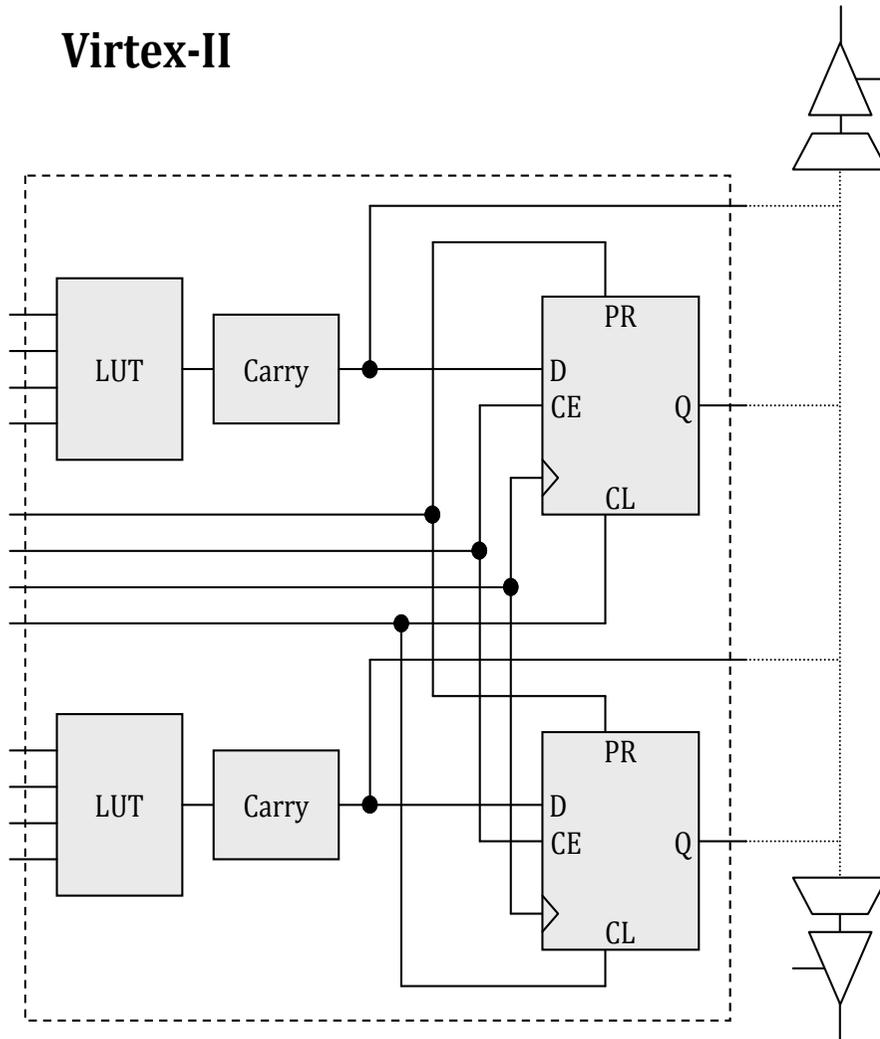
Estructura general de la FPGA modelo Virtex-II de Xilinx



- Cada *CLB* de la Virtex-I permite generar funciones combinacionales y secuenciales.
- Contiene:
 - 4 Slices
 - Conexionado hacia los *CLB*'s vecinos
- Una matriz de conexión permite su conexión con el resto de elementos de la FPGA.

Unidad básica de programación: *slice*

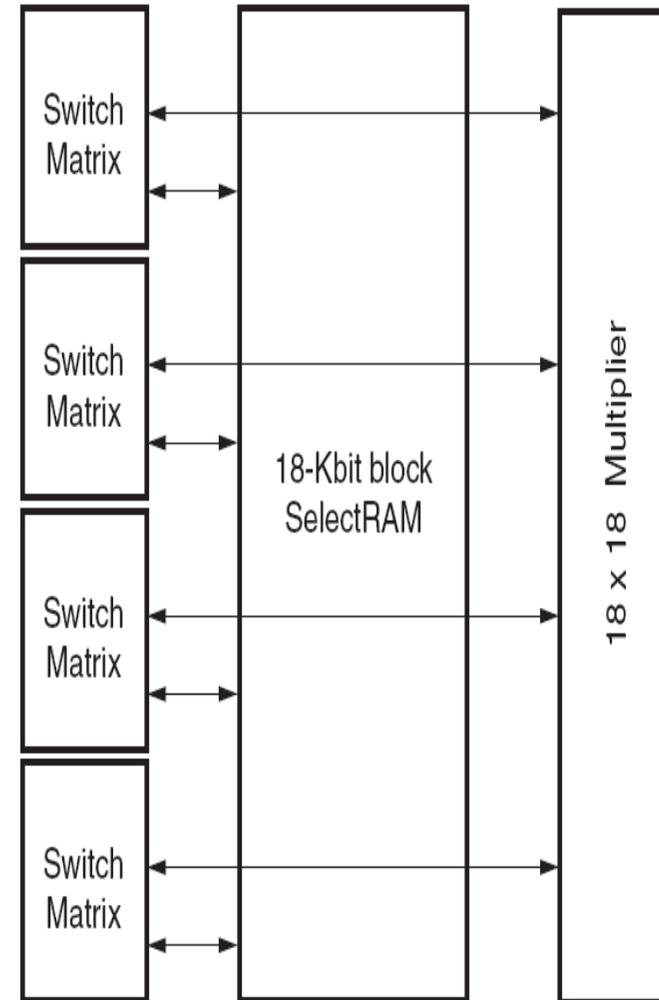
Virtex-II



- Cada *slice* incluye básicamente:
 - 2 biestables D.
 - 2 bloques de *carry*.
 - 2 bloques LUT (*Look-Up Table*).
 - Multiplexores que permiten diferentes configuraciones
- Los bloques LUT
 - Son programables (equivalen a una ROM 16x1) y pueden implementar cualquier función de 4 variables
 - Las dos LUTs del mismo slice se pueden combinar para formar funciones combinacionales de más variables.
- Los biestables D
 - Dispone de Pr y Cl (configurables en modo asíncronos o síncronos)
 - Pueden trabajar en modo latch o en modo registro.

Unidad básica de programación: *BRAM* y *Multiplicador*

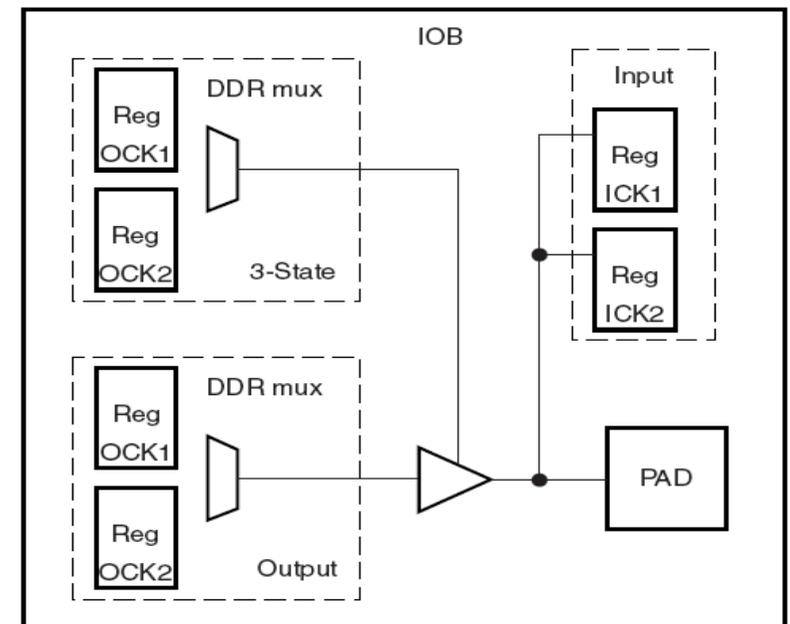
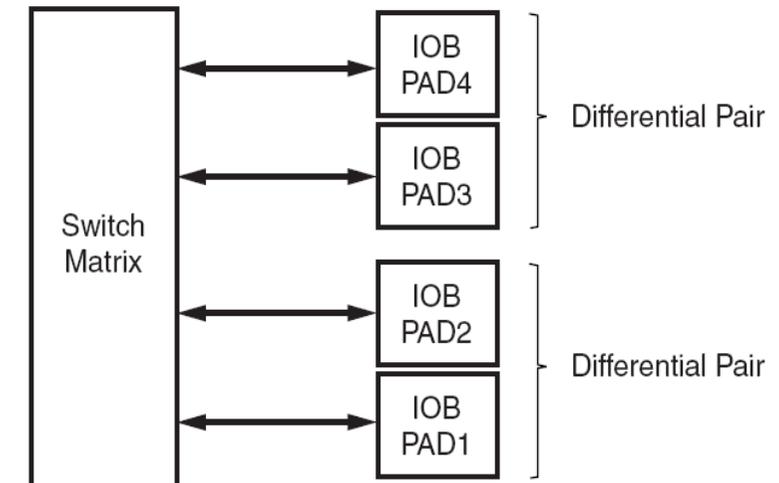
- BRAM 18Kx1 configurable como:
 - Memoria de uno o dos puertos
 - 16kx1, 8kx2, 1kx18,...
 - Conectada a la red general a través de 4 matrices de conexión.
- Multiplicador:
 - 18 x 18 bits
 - Complemento a 2.



Unidad básica de programación: IOB

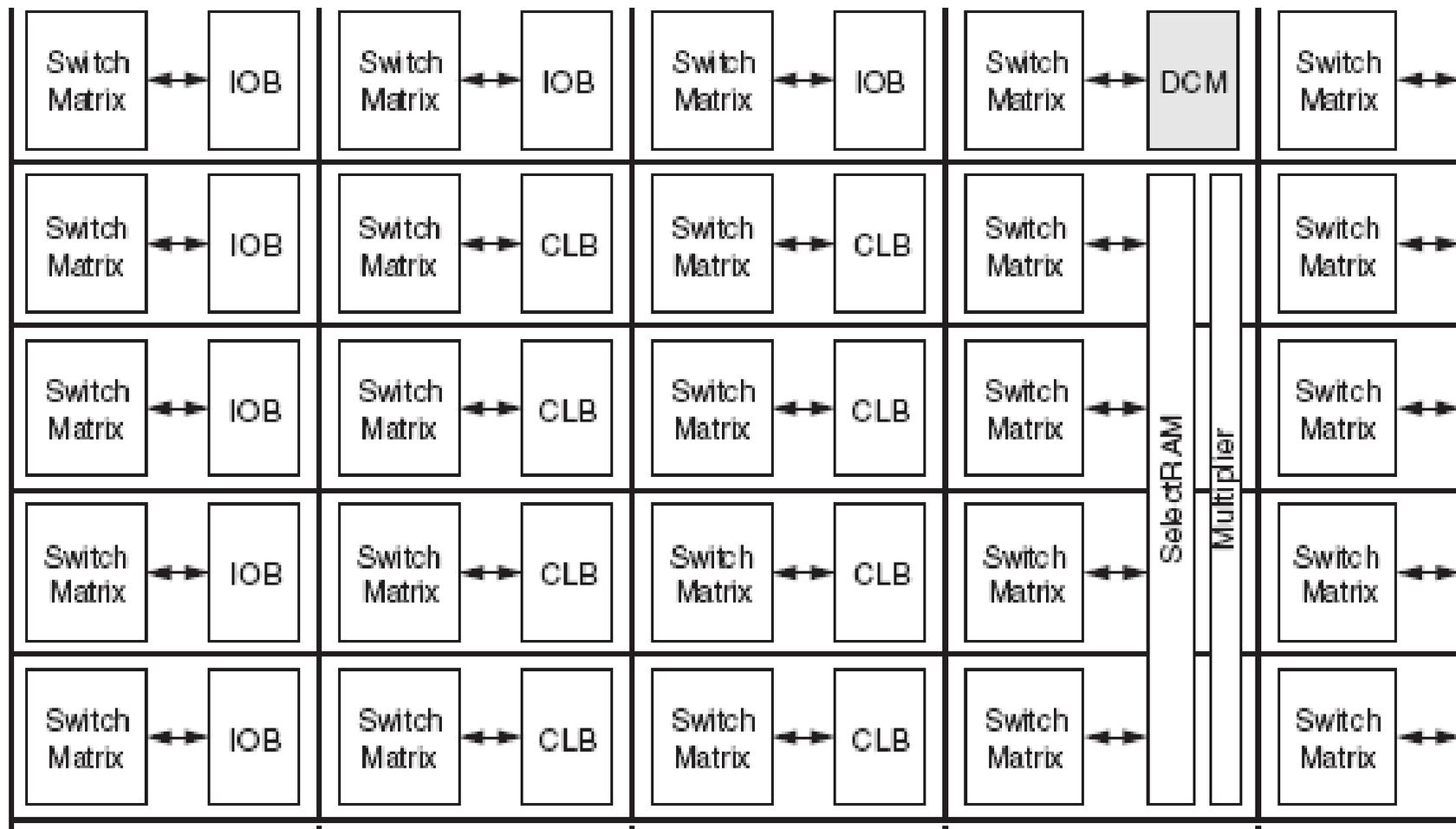
• IOB

- Admiten diferentes tipos de señales:
 - Diferencial (dos pads consecutivos)
 - Single-ended
- Tres partes:
 - Entrada
 - Salida
 - Control de salida (triestado)
- Cada parte tiene dos biestables configurables como latch o registro.
- Cada pin o pad puede configurarse como entrada, salida o bidireccional
- La impedancia (o resistencia) de salida se puede controlar digitalmente.
- Cada 2 o 4 pads tiene una matriz de conexión que los conecta al conjunto.
- Permite transferencias al doble de velocidad DDR.



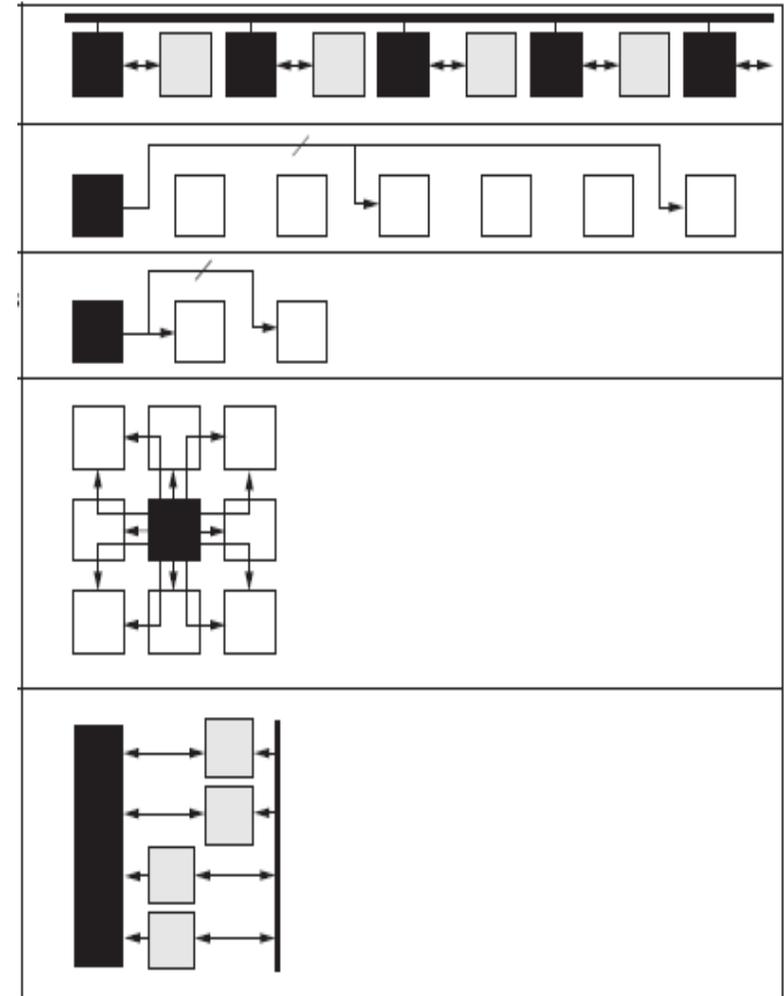
Unidad básica de programación: *Interconexión*

- La mayoría de las señales se envían por la red de líneas horizontales y verticales a la que los diferentes CLB, IOB, etc, tienen acceso a través de la matriz de interconexión programable.



Unidad básica de programación: *Interconexión*

- Long lines
 - Bidireccionales
 - 24 horizontales por cada fila y columna
 - Abarcan toda la FPGA
- Hex lines
 - Unidireccionales
 - 120 por cada fila y columna.
 - Conectan un bloque con su tercero o sexto
- Double lines
 - Unidireccionales
 - 40 por cada fila y columna
 - Conectan un bloque con su contiguo o al siguiente.
- Direct connect lines
 - Conectan un CLB con sus contiguos (incluyendo la diagonal).
 - 16 en total
- Fast connect lines
 - Internas al CLB, salidas de las LUTs a las entradas de otras LUTs



Bibliografía

T. L. Floyd, *Fundamentos de Sistemas Digitales*, 7ª Ed., ISBN 84-205-2994-X, Prentice-Hall, 2002.

S. Díaz, M. C. Romero, A. J. Molina, *Estructura y Tecnología de Computadores. Teoría y problemas*, ISBN 978-84-481-7085-1, McGraw-Hill, 2009.

Xilinx University Program, *FPGA Design Flow Workshop*, 2009.

C. Baena, M. J. Bellido, A. J. Molina, M. P. Parra, M. Valencia, *Problemas de circuitos y sistemas digitales*, ISBN 84-481-0966-X, McGraw-Hill, 2001.