

---

# Tema 0

## *Introducción a los computadores*

---

# Introducción

- **Definición de computador**

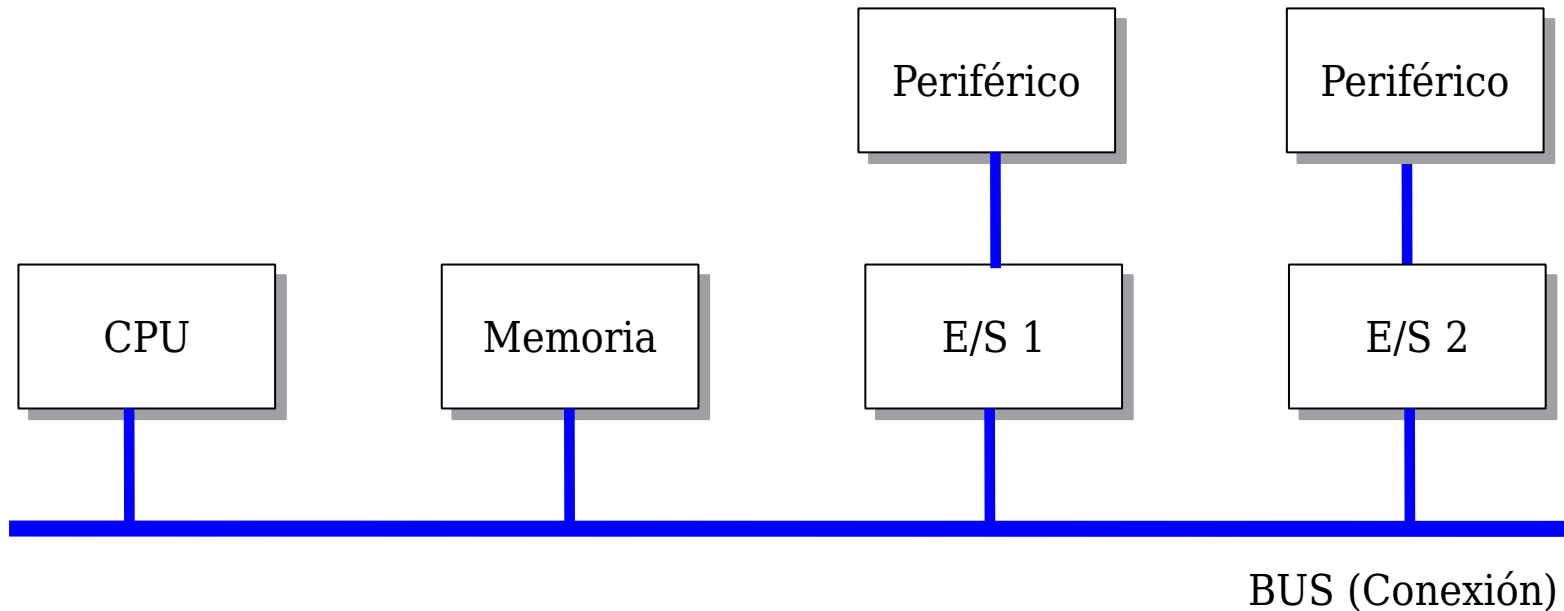
Máquina capaz de realizar de forma automática y en una secuencia programada cierto número de operaciones sobre unos datos suministrados por el operador

- **Características principales**

- Capacidad de cómputo
- Ejecución de un programa
- Alta velocidad de operación
- Alta capacidad de almacenamiento de datos

**Gran rango de aplicaciones**

# Estructura del computador



---

# Estructura del computador

- **CPU** (Unidad central de proceso: **microprocesador**)
  - cerebro del ordenador, ejecuta instrucciones, realiza operaciones lógicas y aritméticas
- **Memoria**
  - almacena datos y programas. Directamente accesible por la CPU
- **Entrada/Salida (E/S, I/O)**
  - comunica la CPU con dispositivos "externos" (periféricos): monitor, teclado, red, modem, discos, etc.
- **Sistema de buses internos**
  - comunica la CPU con la memoria y los módulos de E/S

---

# Función de la CPU

- capta la instrucción que va a ejecutar de la memoria
- cambia el contador de programa para que apunte a la siguiente instrucción
- determina el tipo de instrucción captada
- si la instrucción utiliza datos determina donde están
- almacena los datos en registros internos de la CPU
- ejecuta la instrucción
- almacena los resultados en el sitio adecuado

---

# Estructura de la CPU

- **unidad de control**
  - ✓ controla la secuencia de operaciones realizadas por la CPU
  - ✓ capta las instrucciones contenidas en la memoria principal
  - ✓ analiza el tipo de instrucción y activa las señales necesarias
- **unidad aritmético-lógica**
  - ✓ lleva a cabo las operaciones aritméticas y/o lógicas necesarias para la ejecución de instrucciones
- **registros**
  - ✓ almacenan información dentro de la propia CPU: instrucciones que están siendo ejecutadas, datos que están siendo procesados, direcciones de memoria, ...
- **interconexiones:** comunican a los tres anteriores

---

# Funciones básicas de un computador

- **Procesamiento de datos**
- **Almacenamiento de datos:** temporal o a largo plazo
- **Transferencia de datos:** con periféricos (transferencias E/S), con dispositivos remotos (comunicación)
- **Control:** de recursos del computador, de las diversas unidades funcionales

---

# Hardware/Software

- **Hardware:** conjunto de componentes o sistemas electrónicos o mecánicos que componen el ordenador o sus periféricos.
- **Software:** programas destinados a ser ejecutados por la CPU del ordenador. Son cargados en la memoria principal para su ejecución.
- **Firmware:** programas grabados en memoria de sólo lectura. Suelen ir incluidos con el hardware "de serie".



---

# Memoria: Generalidades

- La memoria digital siempre almacena información **binaria**
- La memoria es una **unidad funcional** donde se escriben o leen palabras binarias:
  - ✓ Programas
  - ✓ Datos
- Características:
  - ✓ Anchura del bus de direcciones
  - ✓ Anchura del bus de datos
  - ✓ Frecuencia de uso
  - ✓ Velocidad
  - ✓ Permanencia de los datos
  - ✓ Capacidad
  - ✓ Otros

# Capacidad y organización

- **Capacidad:**

- ✓ Cantidad de bits que almacenan
- ✓ Ejemplos: 16 Gbits, 64 Kbits, 64 KB (1 B = 1 Byte = 8 bits)

- **Organización:**

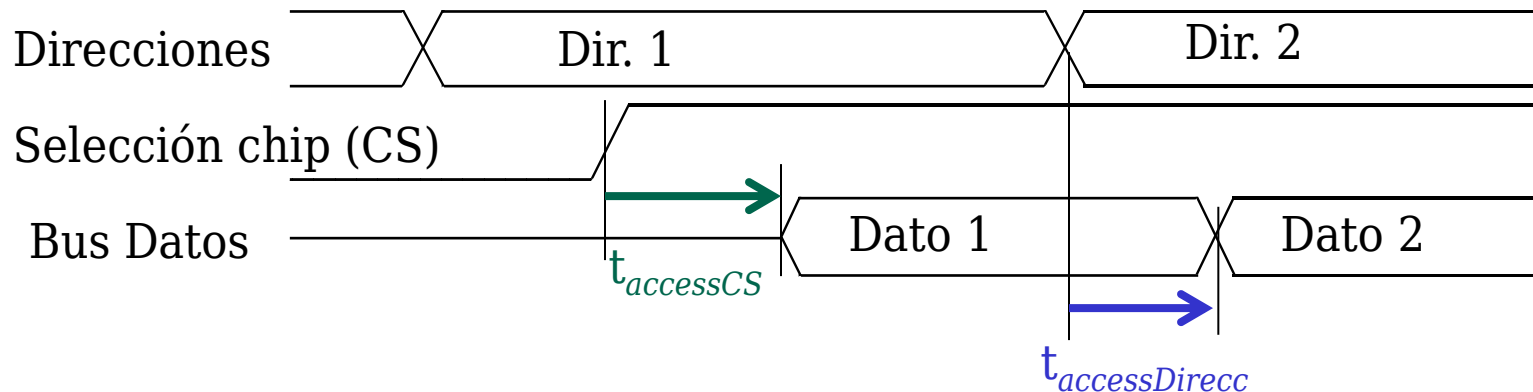
- ✓ Es la estructura o forma en que está guardada o se accede a la información
- ✓ Ejemplos:
  - ✓ Por paquetes, sectores, bloques,... Son grupos de cientos o miles de bits.
  - ✓ Por palabras: dirección de la palabra y anchura de la palabra.  
64 Kbits: 8Kdireccionesx8bit;16Kdireccionesx4bits;  
4 Kdireccionesx16bits

# 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:



---

# Coste y velocidad

- **Coste:** Es lo que cuesta almacenar un bit
  - Desde los comienzos se está abaratando continuamente
  - Depende mucho del tipo de dispositivo
  - Entre los más baratos (los de mayor capacidad, cintas) y los más caros (biestables o registros específicos) hay varios órdenes de magnitud
- **Velocidad:** Varía enormemente
- **Coste y velocidad** suelen variar a la vez y en contra de la capacidad: los más rápidos son los más caros y con menor capacidad

---

# Clasificación según el soporte

Cada bit se almacena en una *celda*, según sea esta hay distintos tipos de memoria

- **Semiconductoras:**
  - La celda es un *condensador* o uno o varios *transistores*.
  - Ejemplos: Biestables, Registros, ROM (*Read Only Memory*), RAM (*Random Access Memory*), Flash, CCD (*Charge Coupled Device*), ...
- **Magnéticas:**
  - La celda es un elemento que se magnetiza por campos EM.
  - Ejemplos: Discos, Cintas, Ferritas, ...
- **Ópticas (magneto-ópticas):**
  - La celda se calienta por láser y cambia sus propiedades físicas; se lee por láser.
  - Ejemplos: CDRoms, DVDs, ...
- **Otras:** Tarjetas perforadas, etc.

---

# Clasificación según el modo de acceso

- **Memorias de acceso aleatorio:**
  - El tiempo de acceso a una palabra no depende de su dirección
  - Ejemplos: RAM y ROM
- **Memorias de acceso secuencial:**
  - El tiempo de acceso depende de la posición física porque para acceder a una palabra hay que pasar antes por otras
  - Ejemplos: Disco, Cinta, CCD, LIFO (*Last In First Out*), FIFO (*First In First Out*)

---

# Clasificación según las operaciones

- **Memorias de sólo lectura:** En tiempo de operación normal sólo se pueden leer los datos ya almacenados en memoria.  
Ejemplos: ROM, CD-ROM, tarjetas,
- Tipos de lectura:
  - ✓ Lectura destructiva (DRO: *Destructive Read Out*): se pierde el dato leído
  - ✓ Lectura no destructiva (NDRO: *No DRO*): el dato se conserva almacenado tras leerlo.
- **Memorias de lectura y escritura:** Se puede modificar el contenido en tiempo de operación  
Ejemplos: **RWM**: *Read Write Memory*): disco, RAM, ferrita

---

# Clasificación según el mantenimiento de la información en ausencia de alimentación

- **Memoria volátil:**
  - ✓ Pierden los datos almacenados si no hay alimentación
  - ✓ Ejemplos de memoria volátil: Registros, RAM
- **Memoria no volátil:**
  - ✓ Ejemplos de memoria no volátil: ROM, DVD, Flash



---

# Clasificación según el mantenimiento de la información en presencia de alimentación

✓ En las RWM semiconductoras:

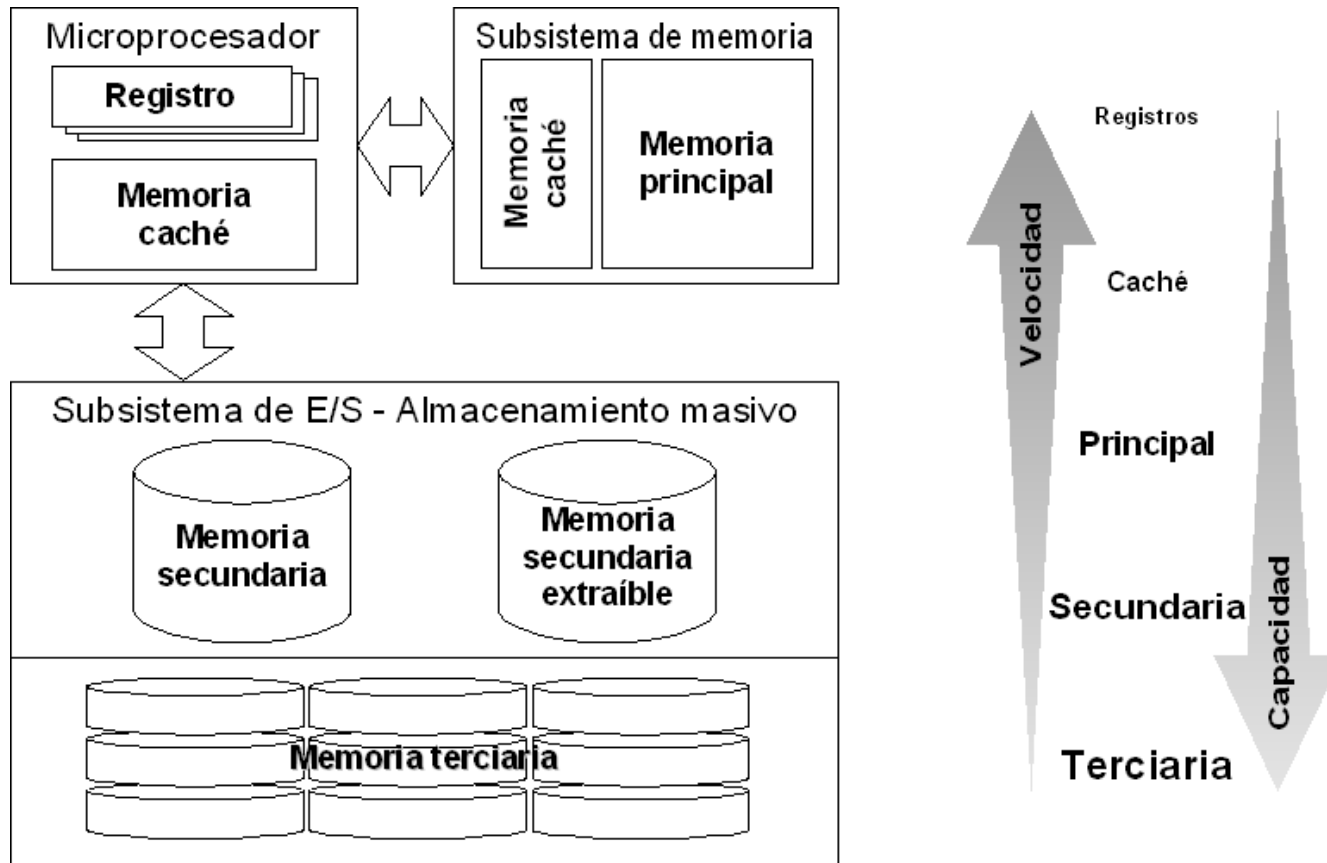
- **Estática** (SRAM: *Static RAM*): El bit se mantiene en el tiempo
- **Dinámica** (DRAM: *Dynamic RAM*): El bit se pierde en el tiempo. Para evitarlo, se realiza una operación de refresco.

# 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

HDD: Hard Disk Drive. RAM: Random Access Memory. ROM: Read Only Memory; FIFO: First-In First-Out. DVD: Digital Versatile Disc

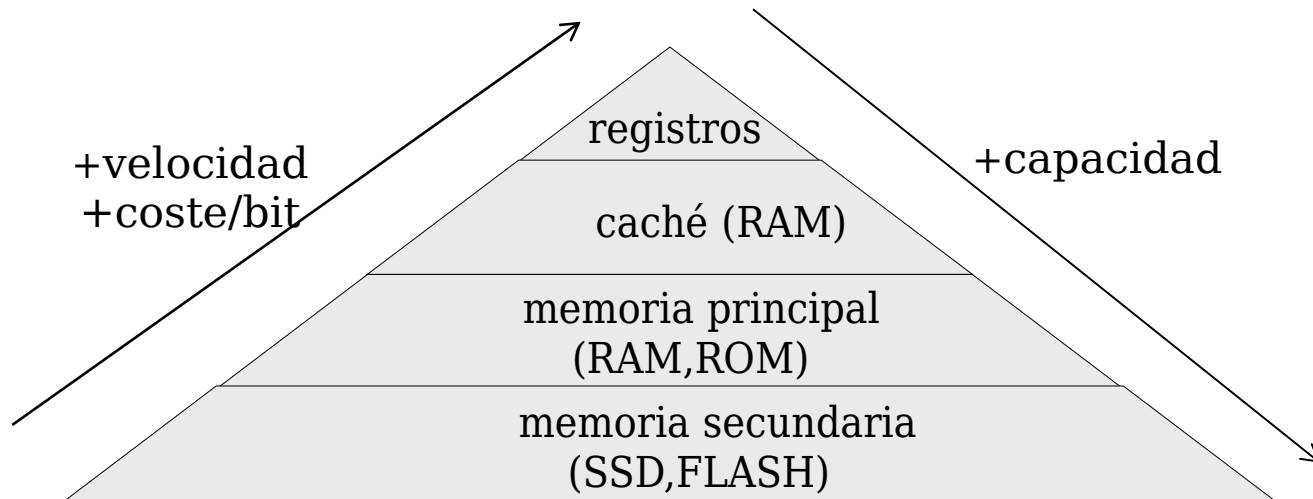
# Memorias. Jerarquías



[Figura extraída de Díaz et al.]

# Memorias semiconductoras:

- Las encontramos en: memoria caché y memoria principal (mayoritariamente volátiles)
- Actualmente también como memoria secundaria (no volátiles)

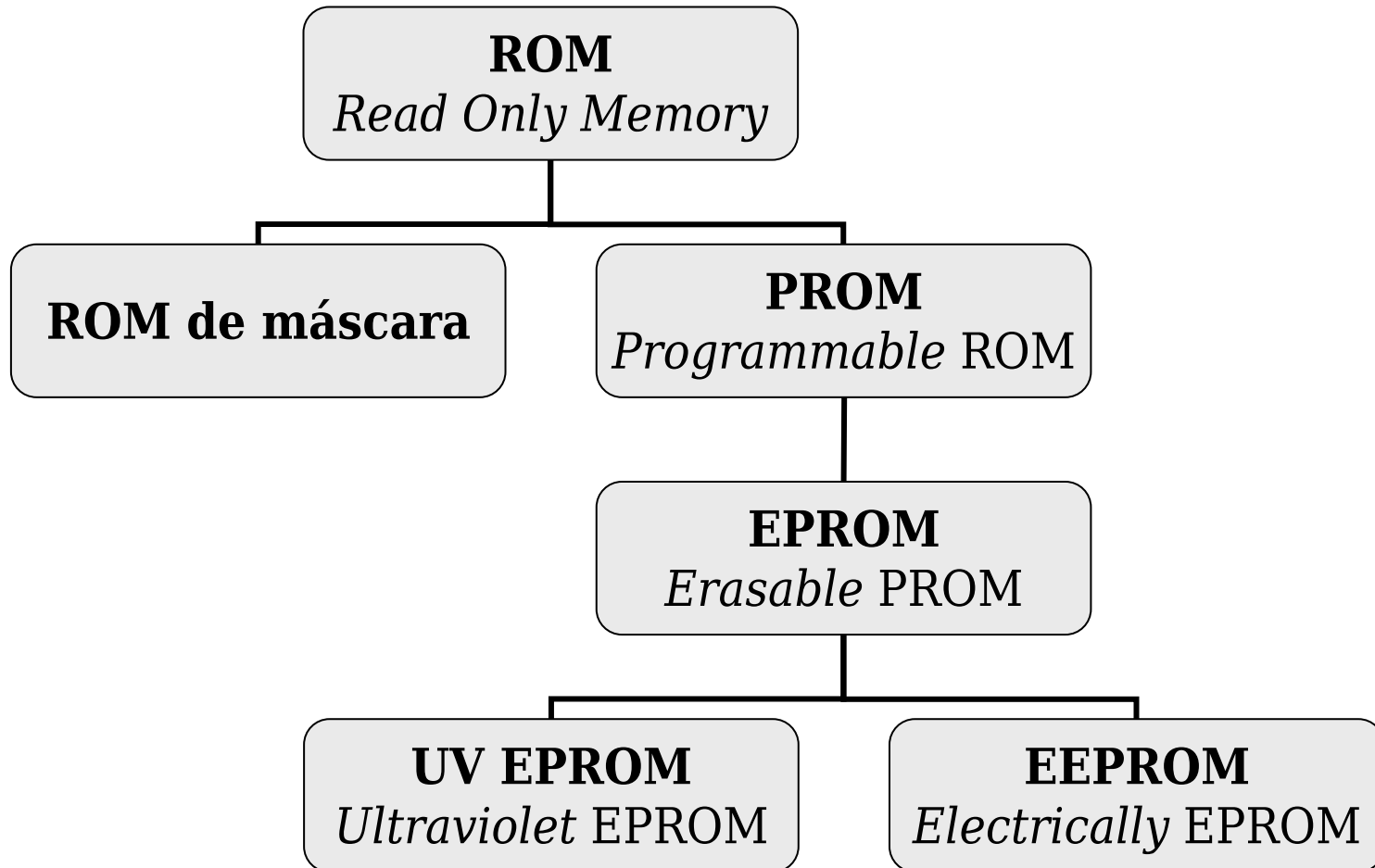


---

# Memorias semiconductoras:

- Son muy rápidas y se dividen en dos categorías:
  - ROM (Read Only Memory):  
Son de solo lectura. Algunas variantes son PROM, EPROM, EEPROM.
  - RAM (Random Access Memory):  
Son de lectura/escritura y volátiles.  
Se subdividen en estáticas (SRAM) y dinámicas (DRAM).

# Familia de memorias ROM

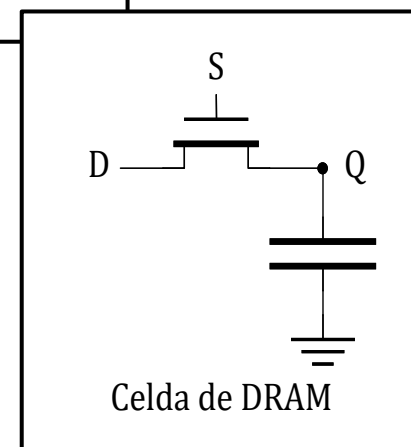
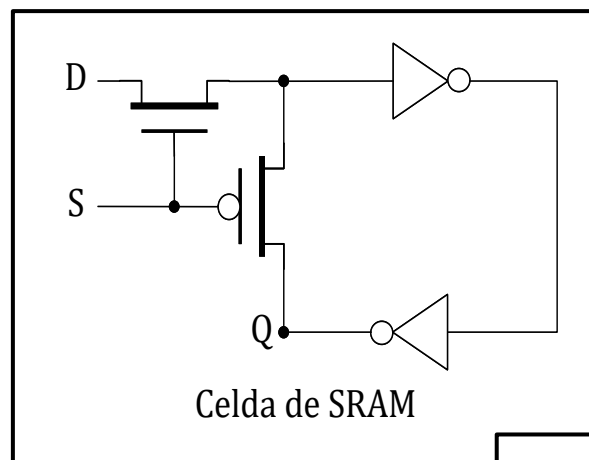


# Tecnologías de RAM

- 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).



---

# Memorias semiconductoras:

## Operaciones básicas y su selección

- **Operaciones básicas:**

- ✓ no-operación (NOP) o no-selección del dispositivo:  $M \leftarrow M$
- ✓ Lectura: Se accede al dato almacenado en una determinada dirección de la memoria:  $D = M(A)$ .
- ✓ Escritura: Se almacena el dato de entrada en una determinada dirección de la memoria :  $M(A) \leftarrow D$ .

- **Entradas de selección:** ejemplos:

- $R$  y  $W$ :  $RW=00$  para NOP;  $RW=10$  para  $R$ ;  $RW=01$  para  $W$ .
- $CS$  y  $R/W'$ :  $CS=0$  para NOP;  $CS=1$  y  $R/W'=1$  para  $R$  y  $R/W'=0$  para  $W$ .
- Otras:  $EW$ : *Enable write*,  $EO$ : *Enable output*,  $Ck$ : *reloj*...



# Memorias semiconductoras de acceso aleatorio:

## Matriz de memoria básica

- Las memorias están formadas por **matrices de celdas**. **Celda:** Cada elemento que puede almacenar 1 bit.

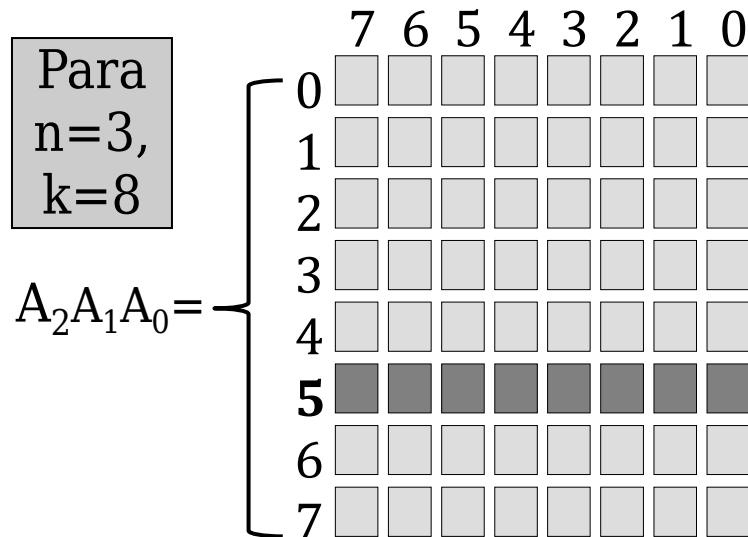
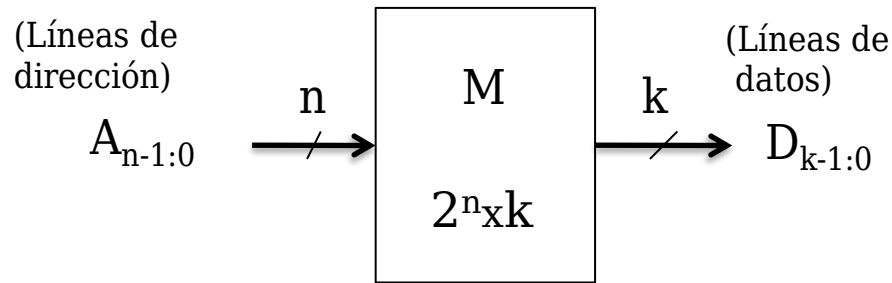
- **Palabra:** Cada **fila** de la matriz. Se le asocia una dirección (*address*). En este caso hay 16. P.ej., palabra 13 (o \$D)

- **Anchura:** Número de bits del dato, en este caso 8. Es la información que puede leerse o escribirse en cada acceso.

- **Capacidad:** Producto del número de palabras por la anchura. En este caso:  $16 \cdot 8 = 128$  bits

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

# Mem. semiconductoras: Líneas de dirección y de datos



- **Líneas de dirección:** Son las “ $n$ ” entradas que dan la posición de la palabra:  $A_{n-1:0} \Rightarrow 2^n$  direcciones.

- **Líneas de datos:** Son las “ $k$ ” salidas (o entradas) que dan acceso al dato almacenado:  $D_{k-1:0}$

- **Capacidad:**  $2^n \times k$  (bits)

- **Acceso aleatorio:** Se accede a todas las palabras en tiempo parecido

- **Ejemplo:**

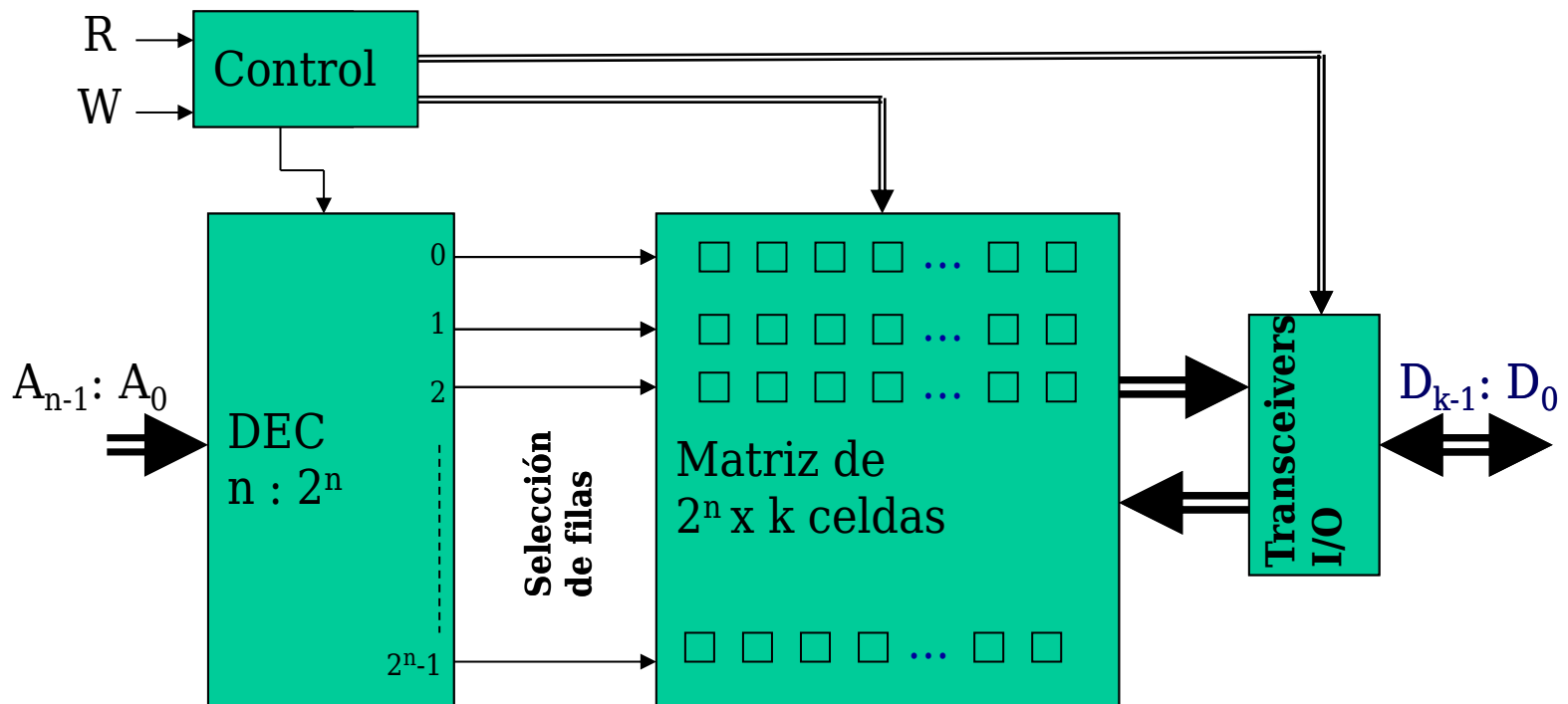
Capacidad:  $2^3 \times 8 \text{ bits} = 8 \times 8 \text{ bits} = 64 \text{ bits} = 8 \text{ B}$

Acceso: Palabra /dirección

$A_2 A_1 A_0 = 1 \ 0 \ 1 = 5_{(10)}$

# Estructura interna (RAM y ROM)

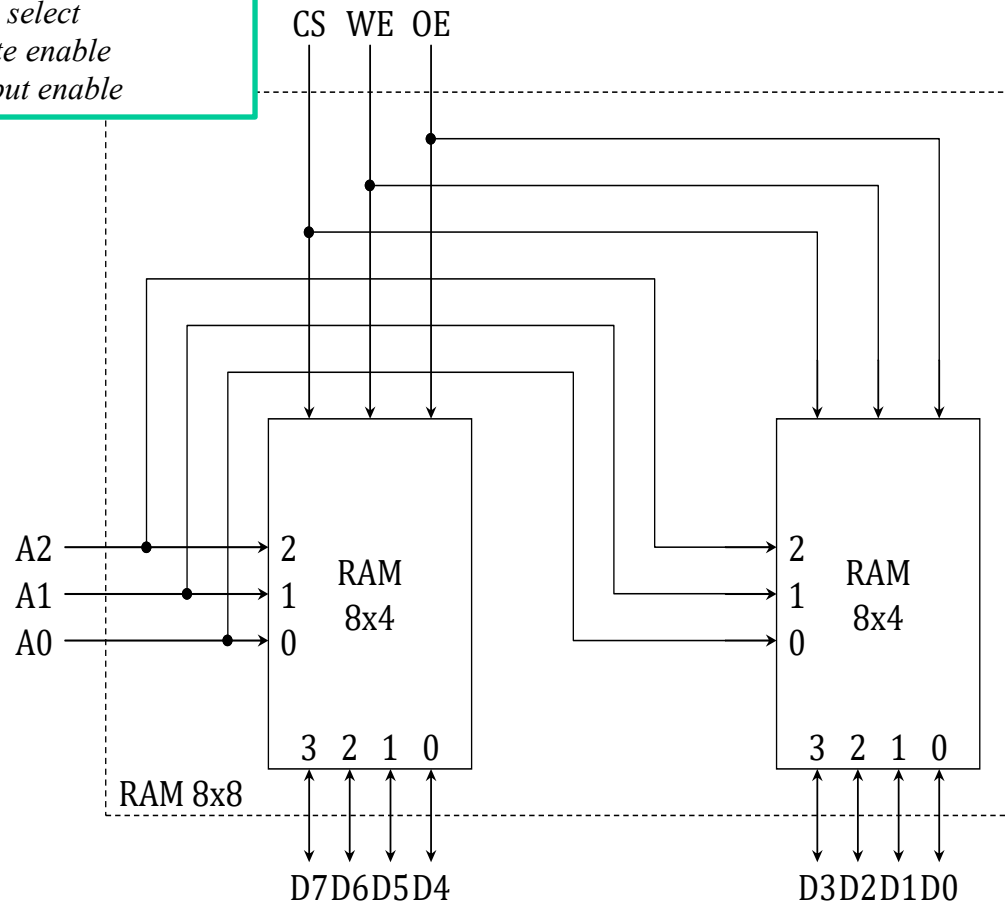
## Estructura interna básica (RAM) $2^n \times k$ :



# Expansión en memorias: doble ancho de palabra

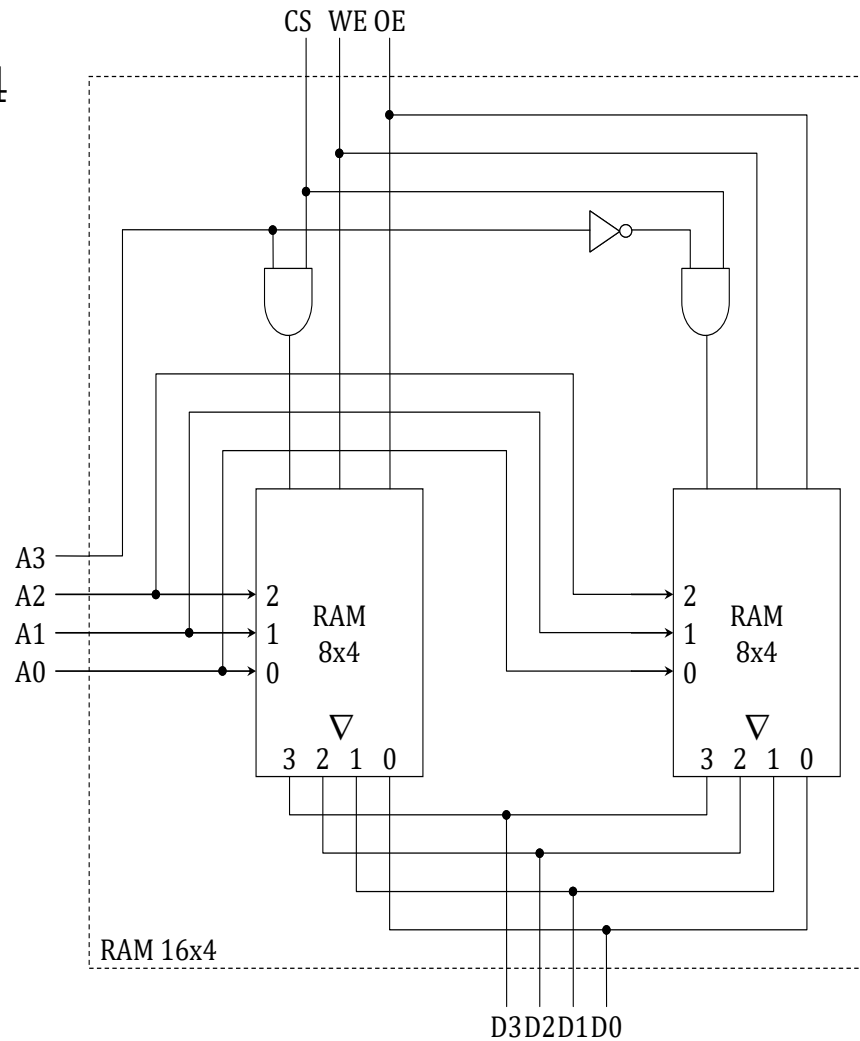
Conseguir una RAM  $2^3 \times 8$  con dos RAM  $2^3 \times 4$

CS: Chip select  
WE: Write enable  
OE: Output enable



# Expansión de número de palabras en memorias RAM

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



# Memoria principal:

## Organización de mapas de memoria

### Objetivo:

Adaptar la CPU a los dispositivos físicos de memoria (RAMs y ROMs) que se utilizarán.

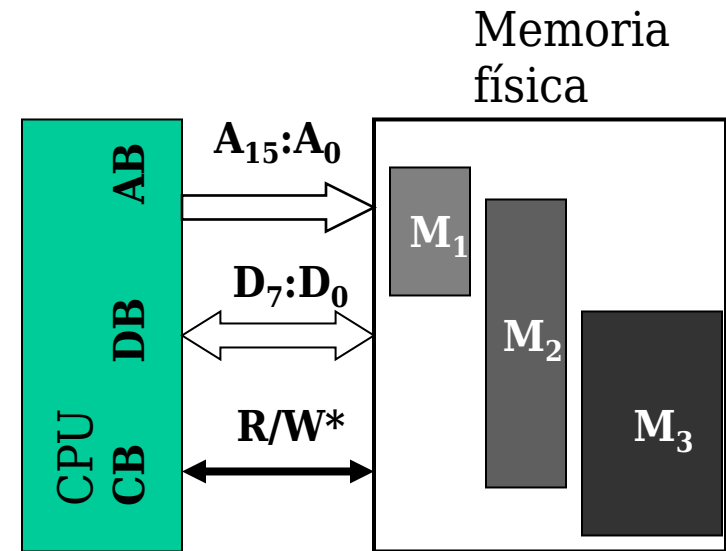
- Hay que adaptar los tres buses:

*Address Bus, AB.* Ej. 16 líneas

*Data Bus, DB.* Ej. 8 líneas

*Control Bus, CB.* Ej. R/W\*

- Cada dispositivo de memoria ( $M_1$ ,  $M_2$ ,  $M_3$ ,...) tiene sus propias líneas de dirección (... $a_1$ ,  $a_0$ ), de datos (... $d_1$ ,  $d_0$ ) y de control (CS, OE; R, W,...)



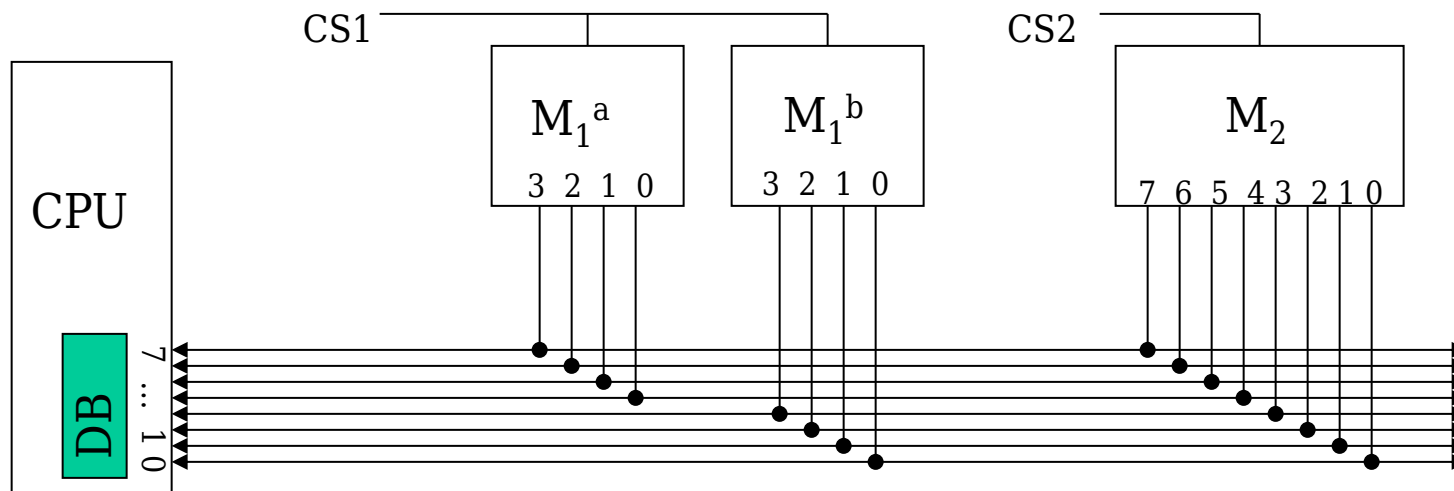
(CS: Chip Select, OE: Output Enable, R: Read, W: Write)

# Adecuación de datos

**Objetivo:** Las memorias suministrarán los datos de DB

**Ejemplo:**  $M_1$  es de 4 bits y  $M_2$  es de 8 bits.

**Solución:** Se asocian 2  $M_1$  ( $M_1^a$  y  $M_1^b$ ) como si fuese una sola  
Se conecta adecuadamente  $M_2$



# Adecuación de direcciones

**Objetivo:** Adaptar las señales de dirección de la CPU con las de los dispositivos de memoria para que:

- la CPU acceda a cualquier palabra de las memorias
- evitar la colisión de éstas en DB (sólo un CS activo como mucho)

**Dirección lógica:** Valor de MAR (y de AB).

Ejemplo:  $A_{15}:A_0 = 1010\ 0000\ 0001\ 1000 = \$ A018$

**Espacio de direcciones:** Es el conjunto de direcciones posibles

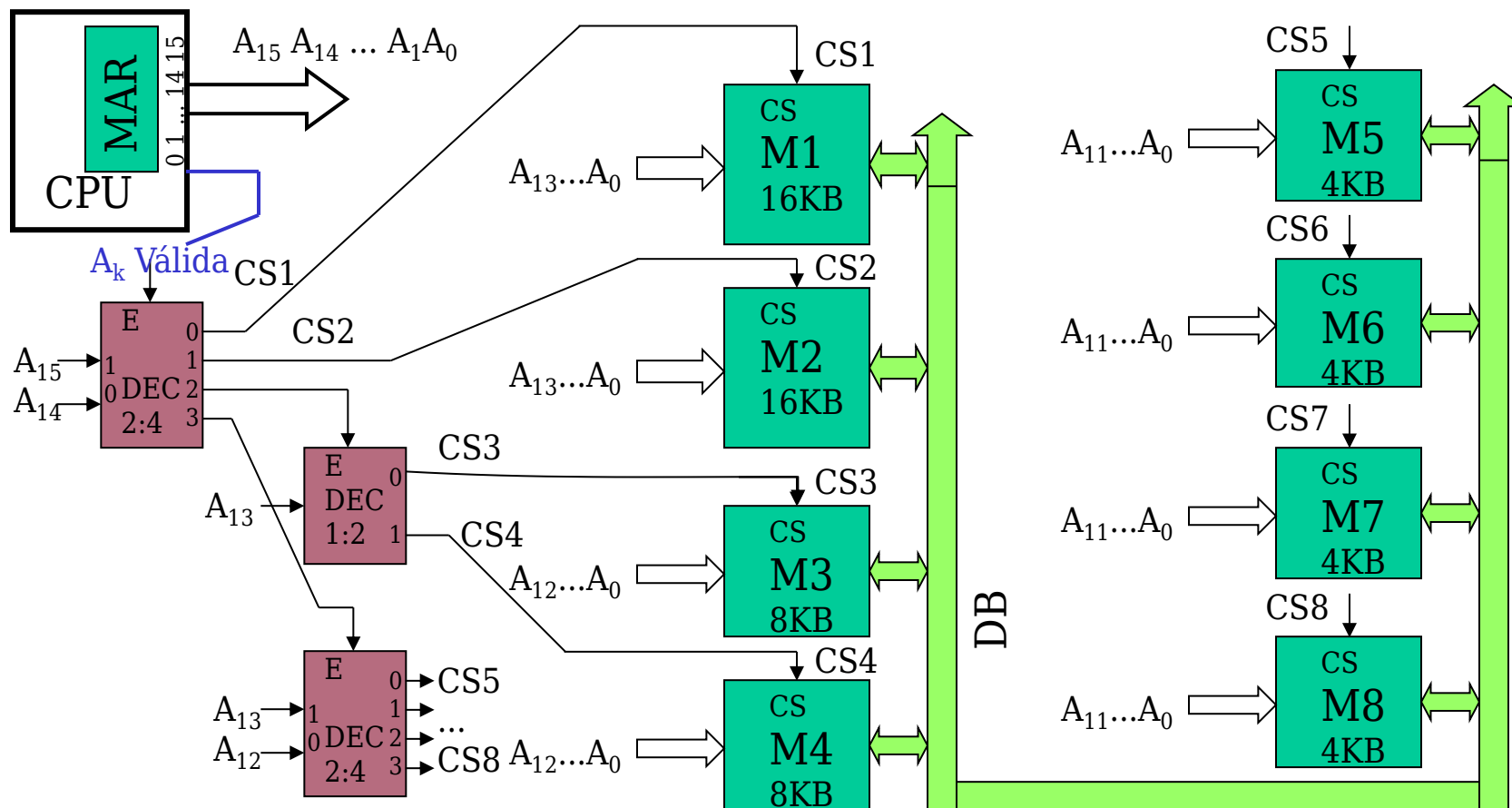
Ejemplo: Hay 64K, desde la \$0000 hasta la \$FFFF

**Dirección (de una palabra) física:** Valor de las entradas de dirección de un dispositivo de memoria.

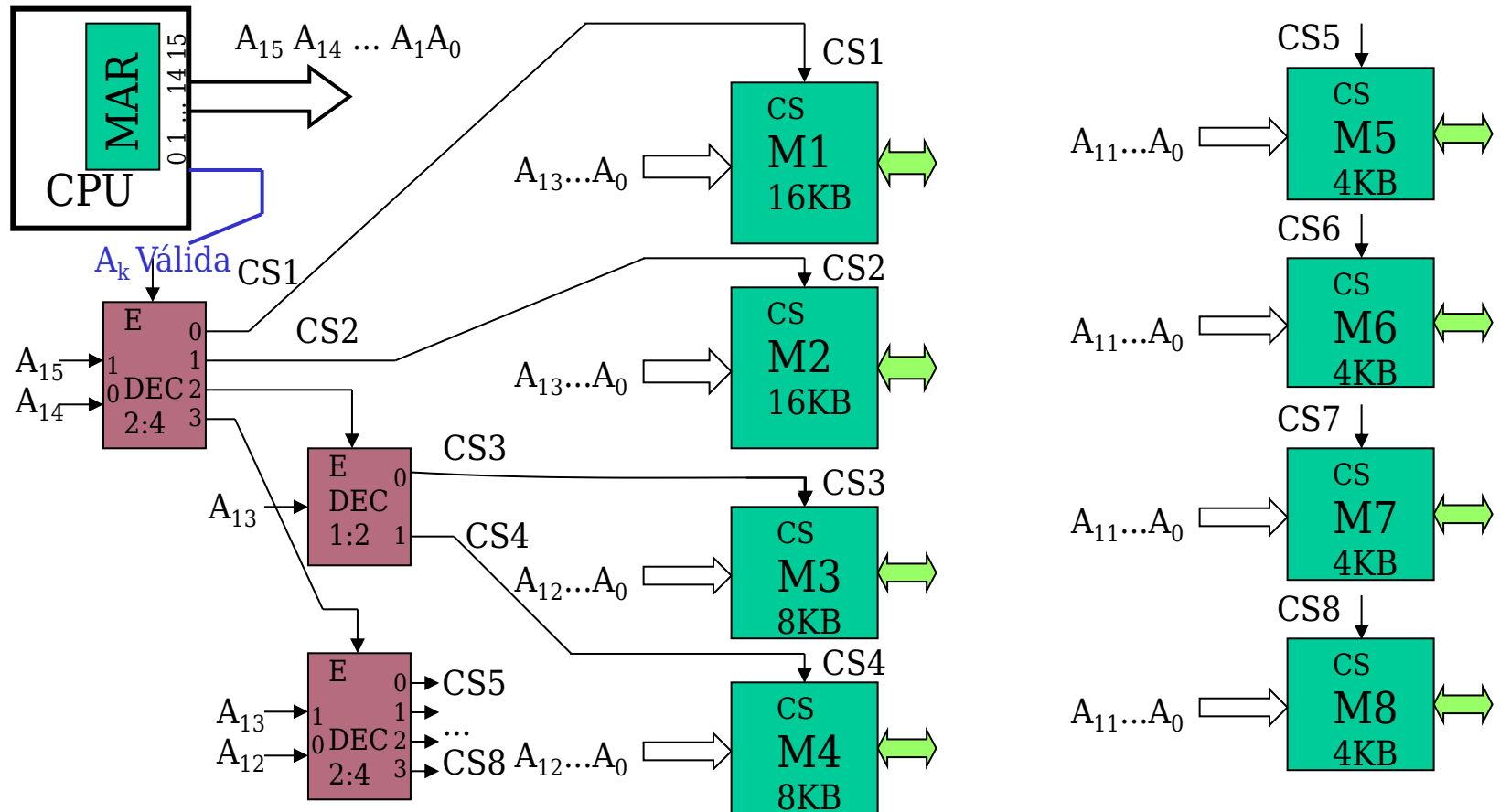
Ejemplo: Para M de 4Kx8,  $a_{11}:a_0 = 0101\ 0011\ 1110 = \$ 53E$



# Organización del hardware: Datos



# Organización del hardware: Decodificación



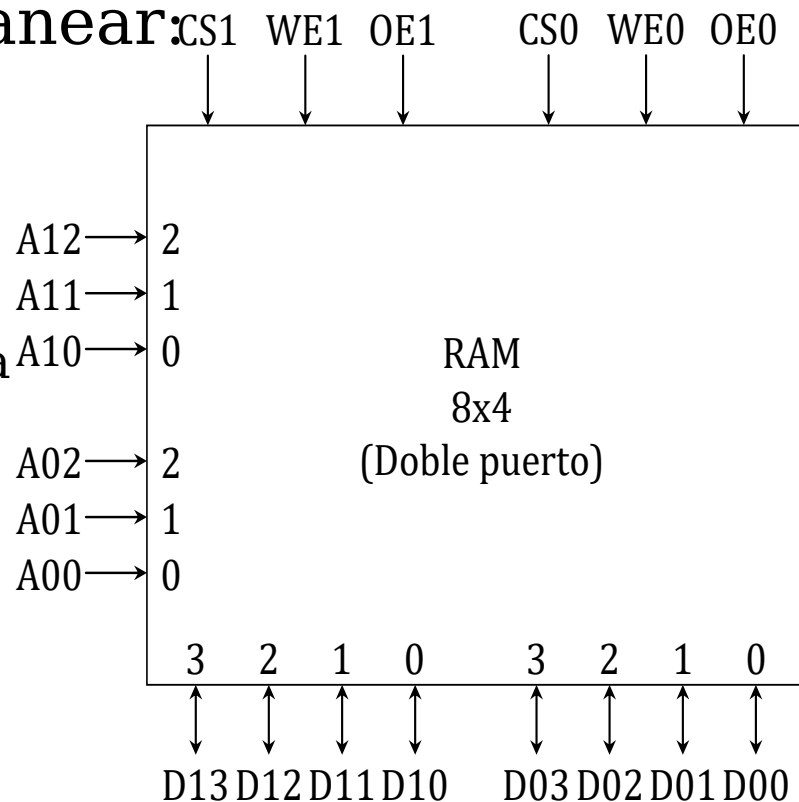
# Memorias de doble puerto

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

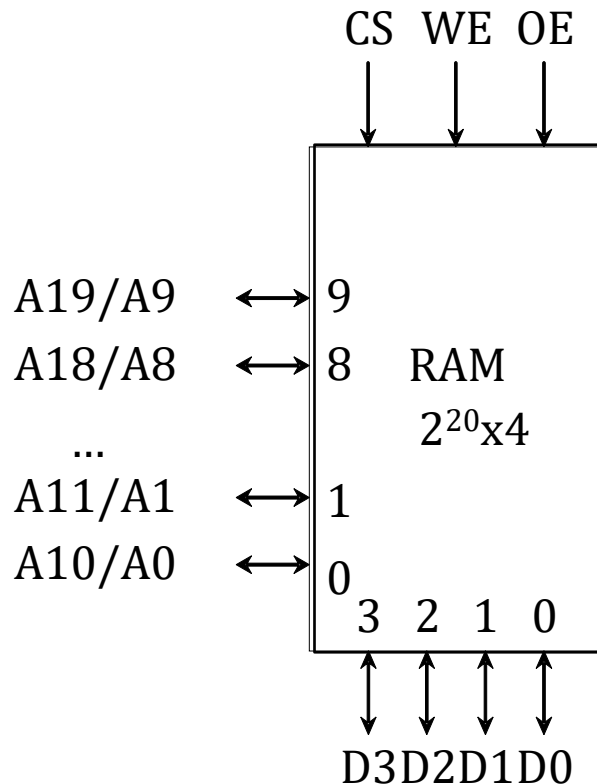
-2 Lecturas

-2 Escrituras

-1 Lectura + 1 Escritura



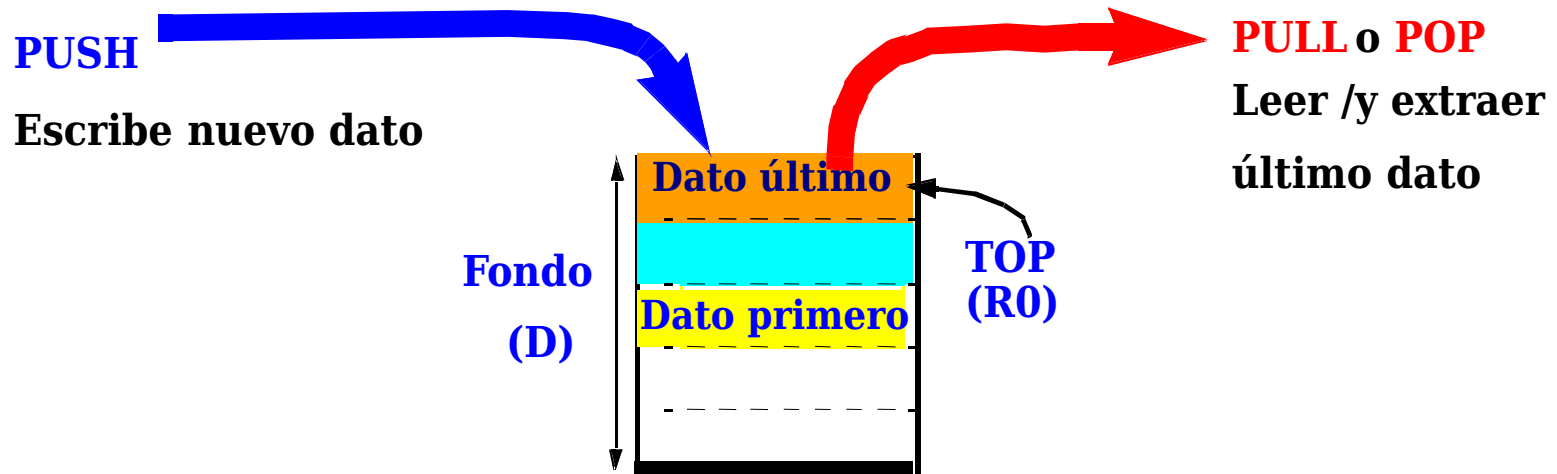
# Memorias con bus de direcciones multiplexado



- 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 LIFO

Memorias LIFO (*Last In-First Out*), también llamadas *stacks* (pilas)



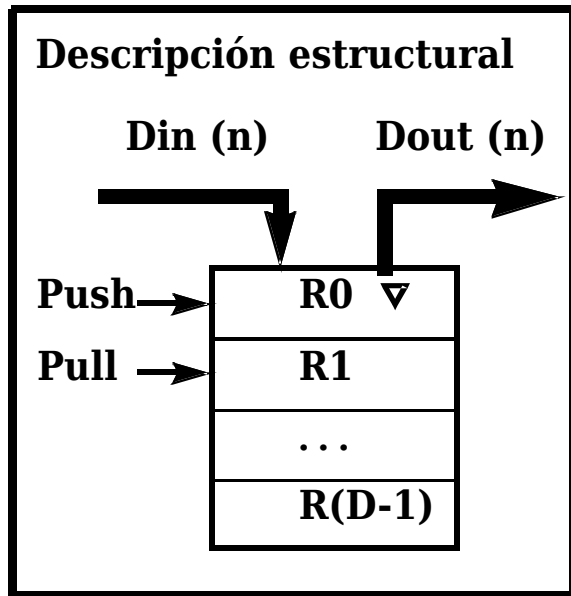
**PILA VACÍA:** Cuando no se ha escrito ningún dato

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

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

# Memorias LIFO

- *Push (Apilar)*: se escribe en la cabecera de la pila (siguiente posición libre).
- *Pull (Desapilar)*: 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	

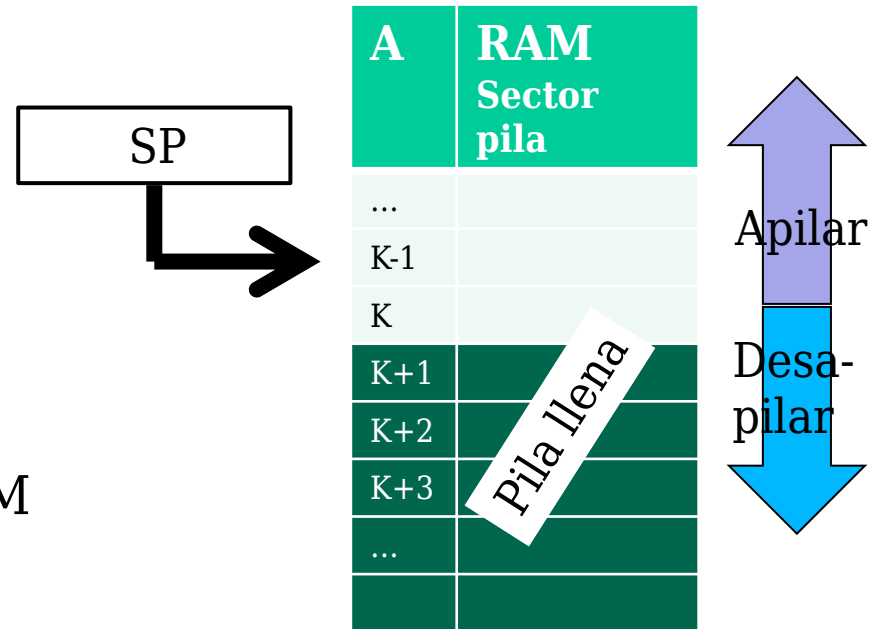
# Memorias LIFO: Realización con punteros

## Fundamentos:

- Datos en un sector de RAM
- El registro TOP está apuntado por SP (*Stack Pointer*)

## Acuerdos a tomar:

- Concretar sector de pila en RAM
- Apilar hacia direcciones menores
- El TOP es la primera palabra vacía



PUSH, Apilar: 1/  $\text{RAM}(\text{SP}) \leftarrow \text{Din}$   
2/  $\text{SP} \leftarrow \text{SP}-1$   
PULL, Desapilar: 1/  $\text{SP} \leftarrow \text{SP}+1$   
2/  $\text{Dout} = \text{RAM}(\text{SP})$

# Memorias FIFO (*First In-First Out*)

