
Arquitectura de Sistemas Informáticos

Memorias: Registros, SRAM, ROM, FLASH

**Departamento de Electrónica
I.E.S. Albert Einstein**

Memorias, usos y tipos

- **Usos:**

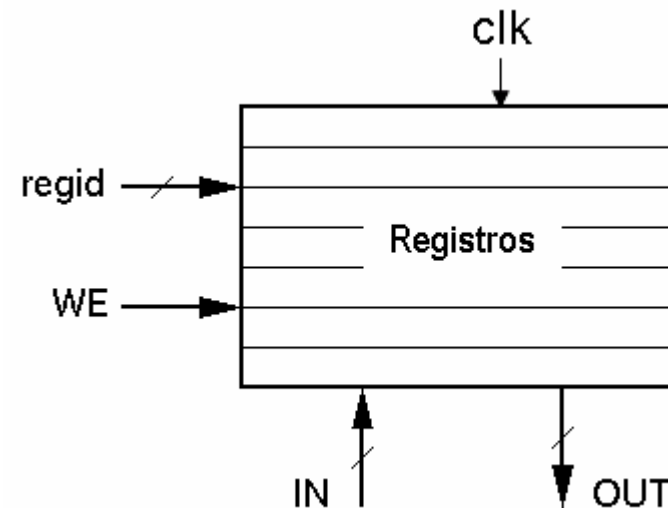
Mantener una cantidad importantes de 'estados' registrados.

- Almacenamiento de datos y programas
- Registos de propósito general
- Buffers
- Tablas de datos
- Memoria sistemas informáticos
- Etc.

- **Tipos:**

- RAM - Random Access memory
- ROM - Read Only Memory
- EPROM, FLASH - Electrically Programmable Read Only Memory

- **Ejemplo RAM: Registros de un microprocesador**



regid = identificador registro
(dirección dato en memoria)

tamaño(regid) = $\log_2(\text{n}^\circ \text{ de reg})$

WE = write enable (permitir escritura)

p.ej: 256 posiciones requieren 8 líneas de direcciones

Definiciones

Interfaz de memoria para Acceso a Datos

- **Asíncrono (sin reloj):**

Un cambio en la dirección produce que aparezcan los datos en el bus

- **Síncrono (con reloj):**

Un cambio de dirección, seguido del flanco/nivel del reloj (CLK) produce la aparición del dato o provoca una operación de escritura.

A veces se puede combinar operaciones de escritura síncronas y de lectura asíncronas.

- **Volátil:**

Se pierden los datos/estados si se desconecta la energía eléctrica.

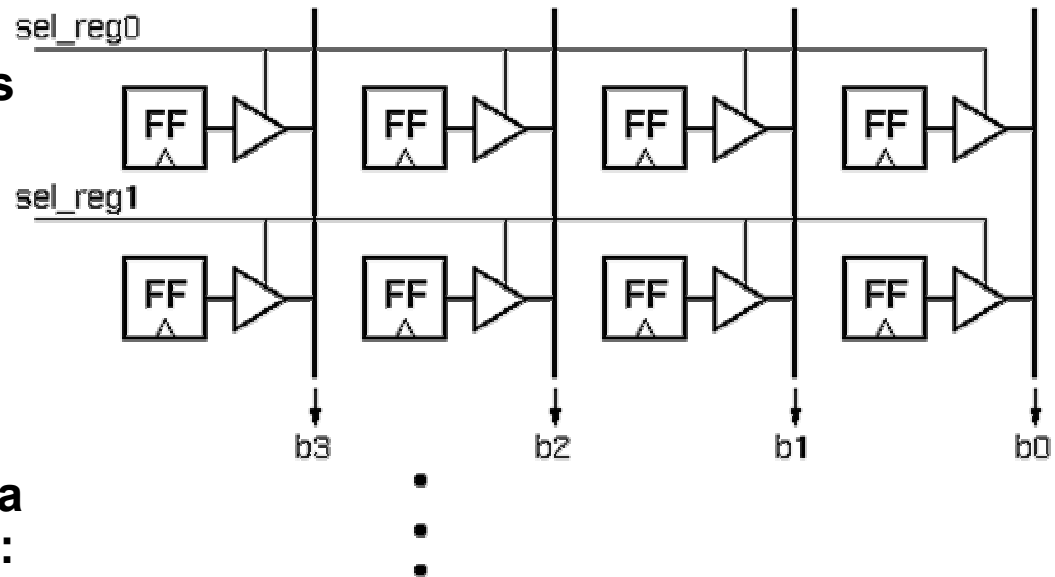
- **No volátil:**

Retiene los datos/estados a pesar de retirar la alimentación eléctrica.

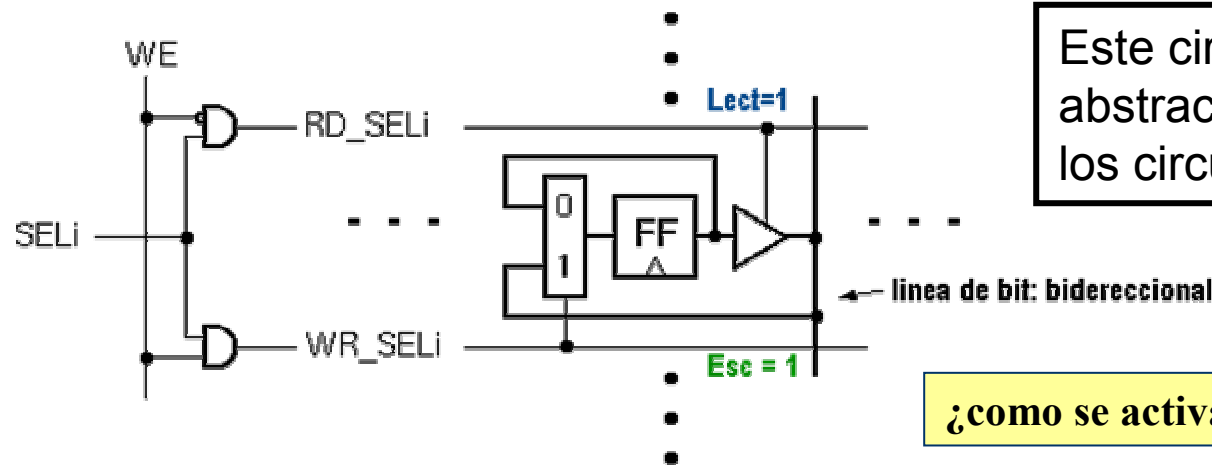
Memoria: Grupo de células básicas

- Para operaciones de lectura, los registros son equivalentes a un array 2-D de flip-flops con salidas tristate en cada uno:

- Se agrupan por ‘palabras’.
- Se activa una línea (palabra).



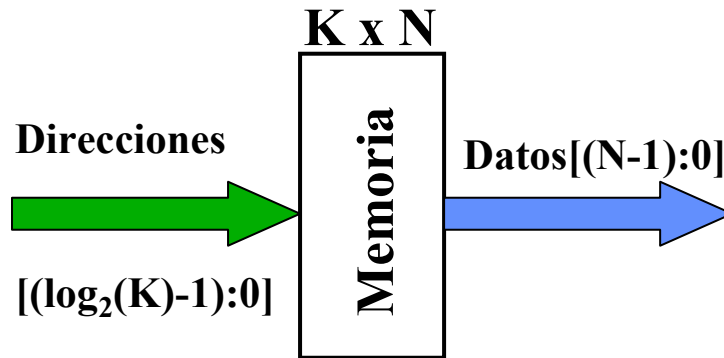
- Unidad con lógica de escritura añadida (si se puede escribir):



Este circuito es una abstracción funcional de los circuitos reales.

¿como se activa "SEL" con una dirección?

Memoria: Grupo de células básicas



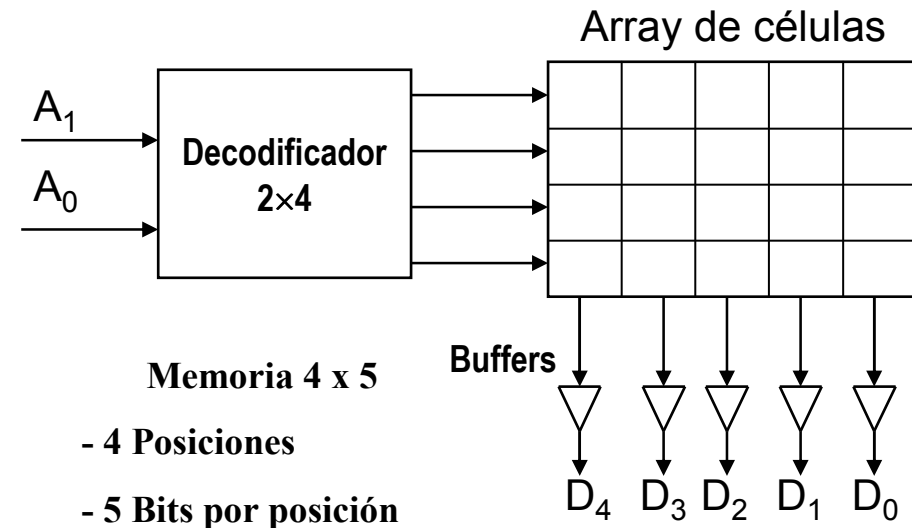
K 'casillas', N bits por 'casilla' (posición)

El **bus de direcciones** tiene $\log_2(K)$ líneas, el **bus de datos** tiene N líneas.

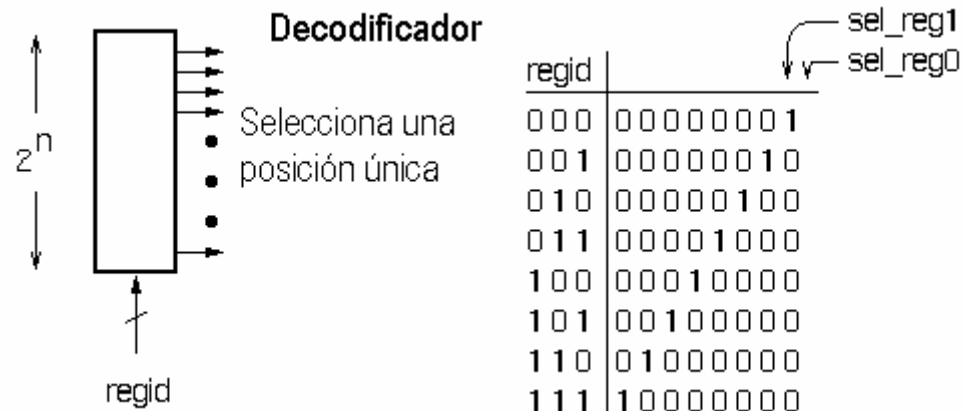
Memoria $n \times m$ (n posiciones – m bits)
 $\log_2 n$ – líneas direcciones "address lines"
m – líneas entrada/salidas "data lines"

Esto es común para todos los tipos de memorias existentes:

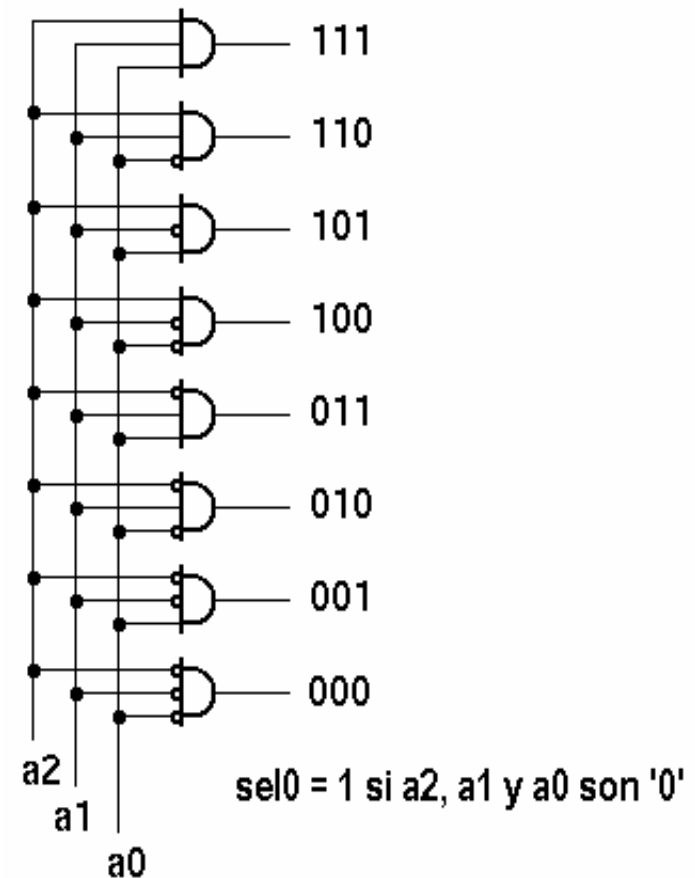
-ROM, RAM, EEPROM, FLASH, etc.



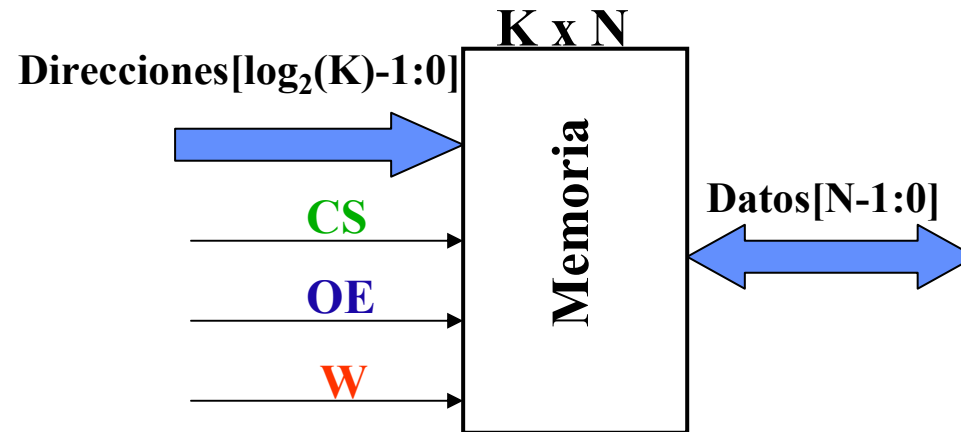
Decodificador de direcciones (address)



- La función del decodificador de direcciones es generar la selección para una dirección única de registro interno de la memoria.
 - Binario -> una dirección un dato (palabra)
 - Simplificado (demultiplexor)
- La salida se usa para seleccionar la fila.
- Existen diferentes circuitos para esta función.
- Estos circuitos están dentro de la memoria.



Memorias: Lineas de control



Chip Select – Debe ser activado para que la memoria responda a las operaciones, sean de lectura o de escritura. Si está negado, el bus de datos está en estado de ‘**alta impedancia**’.

OE – (Output Enable) Activa para operaciones de lectura. Si se niega, el bus de datos está en ‘**alta impedancia**’.

W – Activa para operaciones de escritura, las líneas de datos se convierten en entradas de la memoria.

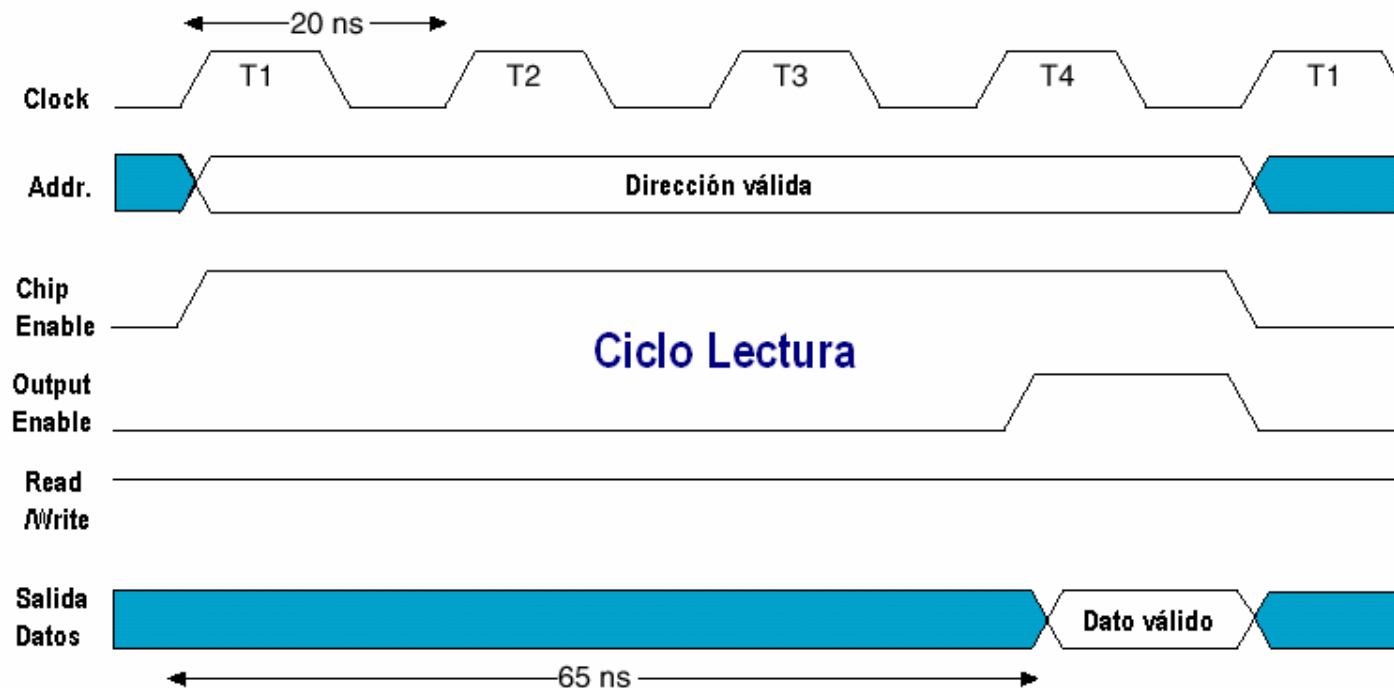
Lineas de Control de Chip Memoria

Chip select CS	Read/Write R/ \overline{W}	Operación Memoria
0	×	Ninguna
1	0	Escribir a la posición seleccionada
1	1	Leer de la posición seleccionada

Memoria: Operación de lectura

- **Lectura asíncrona**

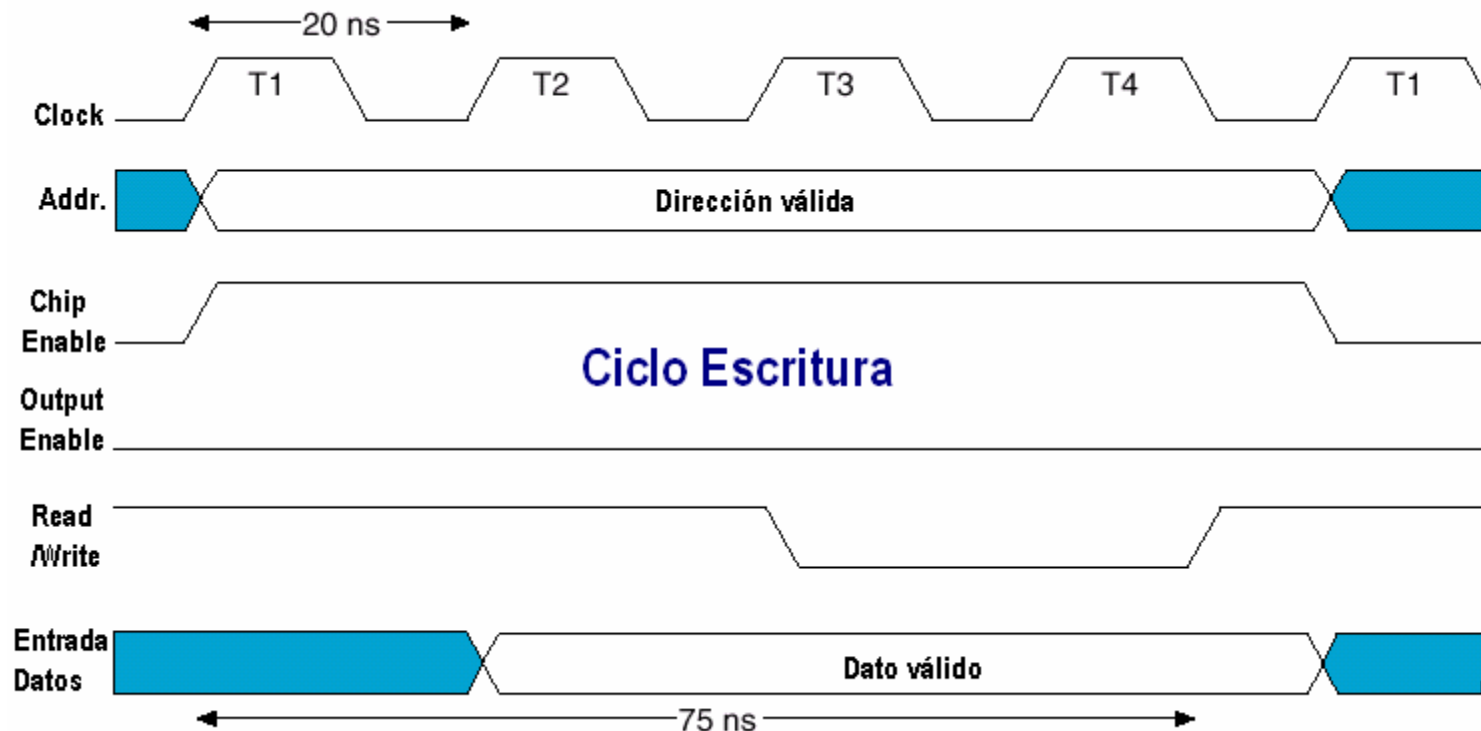
- Selección de chip (CE)
- Establecimiento de dirección (Addr).
- Al activar la salida, datos en salida de forma inmediata.
- Independiente del reloj.



Memoria: Operación de escritura

- **Escritura síncrona**

- Se establece la dirección de escritura en el bus de direcciones.
- Se pone el dato a grabar.
- Se activa la orden 'escritura' y en el siguiente ciclo de reloj se produce la escritura.



Tipos de memoria

- **Volátil:**

- Random Access Memory (RAM), escritura y lectura:
 - » SRAM “estática“, no necesita ‘refrescar’ valores.
 - » DRAM “dinámica“, necesita operaciones de ‘refresco’.

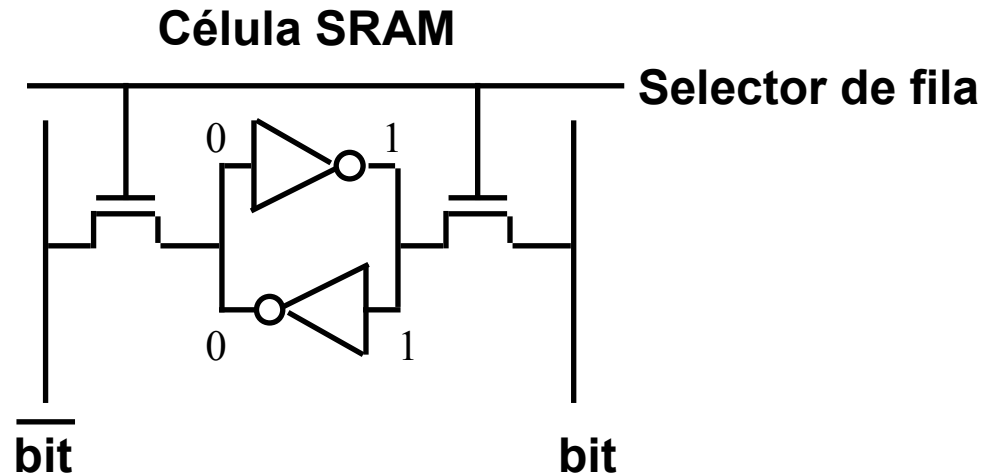
La información se pierde al desconectar la alimentación.

- **No-volátil:**

- Read Only Memory (ROM), solo lectura:
 - » Máscara ROM “mask programmable”
 - » EPROM “electrically programmable”
 - » EEPROM “erasable electrically programmable”
 - » FLASH memory - similar a EEPROM con programador integrado en el chip. * **Permite escrituras de memoria.**

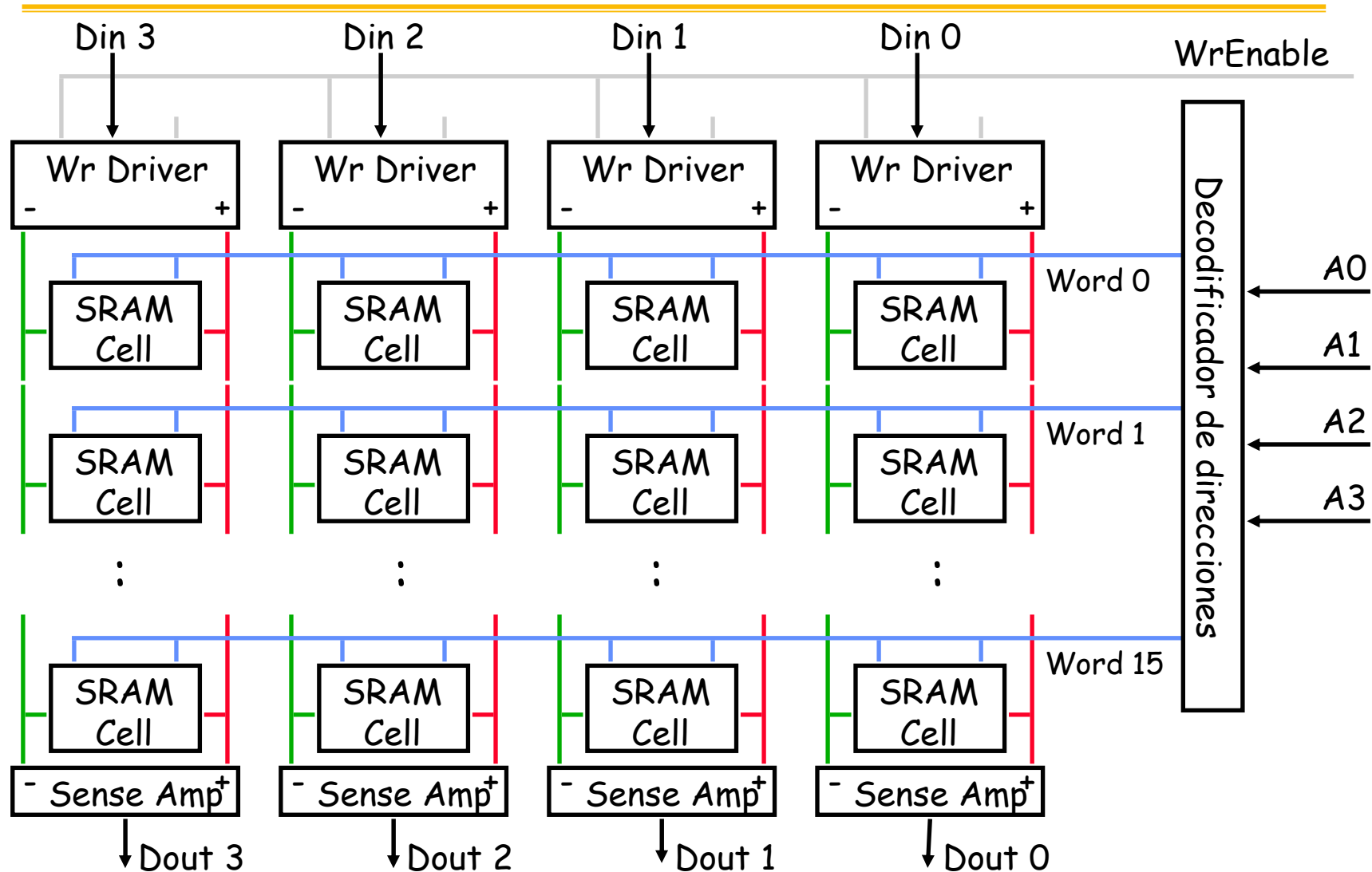
La información no se pierde al desconectar la alimentación.

Célula de RAM estática



- **Lectura:**
 - 1. Seleccionar fila
 - 2. La célula de memoria pone una línea a 'low' y la otra a 'high'
 - 3. Salida sobre bit y $\overline{\text{bit}}$
- **Escritura:**
 - 1. Ajustar líneas bit (p.e. $\overline{\text{bit}}=1$, bit=0)
 - 2. Seleccionar fila
- **¿Como funciona?**
 - Cuando una línea de bit es 'low', fuerza la salida a 'high'; esta operación cambiará el estado al seleccionar columna, si el nuevo estado es diferente, sino, quedaría igual.

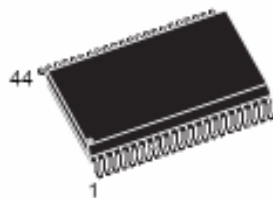
Ejemplo de SRAM: 16-'palabras' x 4-bits



Ejemplo: ST microelectronics M68AW256M

FEATURES SUMMARY

- SUPPLY VOLTAGE: 2.7 to 3.6V
- 256K x 16 bits SRAM with OUTPUT ENABLE
- EQUAL CYCLE and ACCESS TIME: 55ns, 70ns
- SINGLE BYTE READ/WRITE
- LOW STANDBY CURRENT
- LOW V_{CC} DATA RETENTION: 1.5V
- TRI-STATE COMMON I/O
- AUTOMATIC POWER DOWN
- TSOP44, and TFBGA48 PACKAGES
 - Compliant with Lead-Free Soldering Processes
 - Standard or Lead-Free Option



TSOP44 Type II (ND)



TFBGA48 (ZH)
6 x 8mm



TFBGA48 (ZB)
7 x 8mm

10-9-2007

Figure 2. Logic Diagram

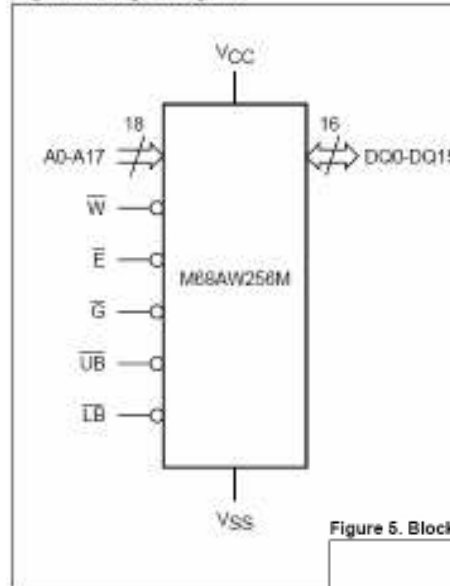
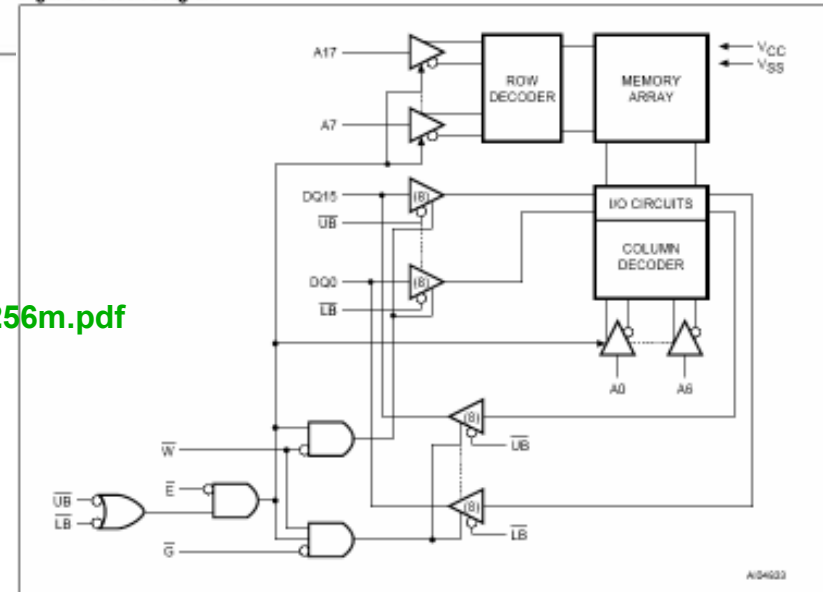


Table 1. Signal Names

A0-A17	Address Inputs
DQ0-DQ15	Data Input/Output
\overline{E}	Chip Enable
\overline{G}	Output Enable
\overline{W}	Write Enable
\overline{UB}	Upper Byte Enable Input
\overline{LB}	Lower Byte Enable Input
V_{CC}	Supply Voltage
V_{SS}	Ground
NC	Not Connected Internally
DU	Don't Use as Internally Connected

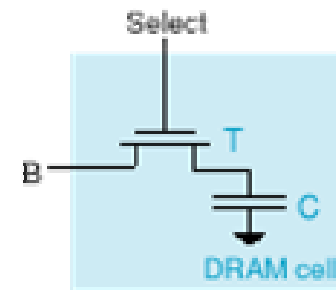
Figure 5. Block Diagram




<http://www.st.com/stonline/products/literature/ds/7996/m68aw256m.pdf>

DRAM Memoria RAM dinámica

- **SRAM – Sistemas de almacenamiento tipo ‘básculas’**
 - ¿Más rápidas? - Si, acceso a dato más rápido.
 - Más caras por unidad de información (al ser más complejas)
 - Capacidades de almacenamiento menores
- **DRAM – Guarda el dato en forma de ‘carga’ de un condensador**
 - La información desaparece tras un período de tiempo.
 - Debe ser ‘refrescada’, o lo que es lo mismo, reescrita / recargada
 - Condensador que retiene carga
 - Un transistor actua como ‘puerta’
 - Sin carga, almacena un ‘0’, con carga ‘1’
 - Se lee cerrando la ‘puerta’
 - La lectura es ‘destruktiva’
 - » Cuando una célula se lee, pierde carga
 - » Debe ser restaurada después de una lectura
 - Refresco
 - » También hay pérdidas de carga por fugas
 - » La carga debe ser restaurada periódicamente



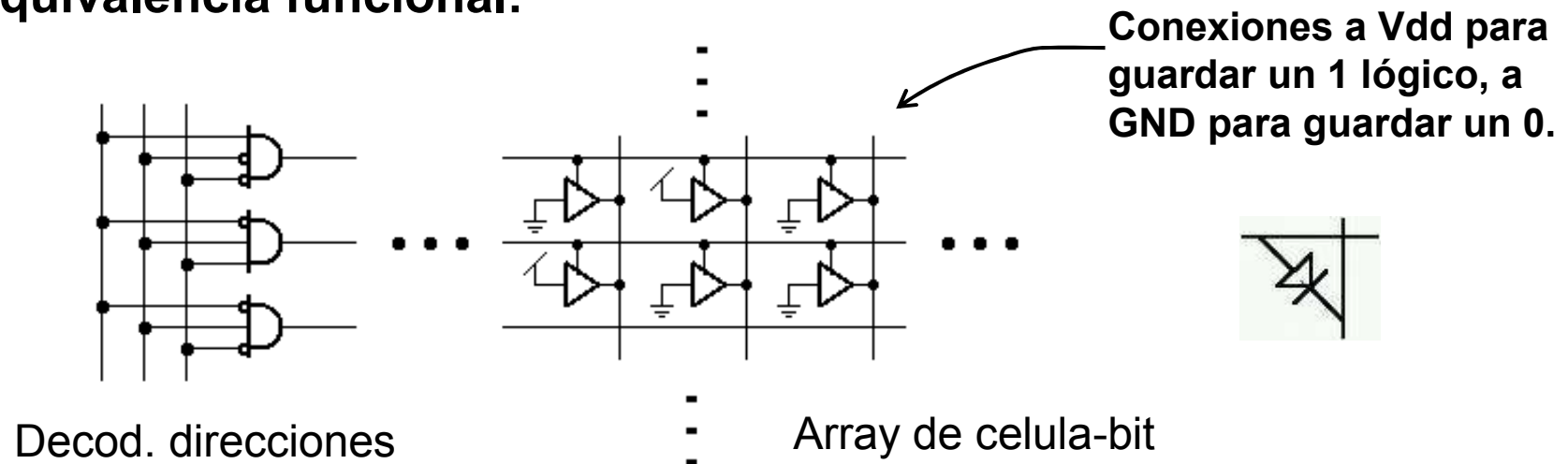
**Tamaño muy pequeño
de célula o unidad de
memoria**

- 



Read Only Memory (ROM)

- Forma simplificada de memoria. No se puede escribir.
- Equivalencia funcional:



- Se programan en fabricación según petición a fabricante.
- Otras son OTP (One Time Programming), solo se pueden programar una vez y se hace 'destruyendo' electricamente las uniones a '1' o '0' y de esta forma se establece el dato fijo.

Memoria no volátil

Se usa para guardar datos y código 'fijo' (p.e. BIOS), tablas de datos (p.e nuevos estados en salidas de FSM), valores que no sufren muchos cambios y deseamos que permanezcan después de 'apagados' (flash, eeprom.)

- **ROM de máscara**

- Datos, programas, tablas, etc, fijas y permanentes.
- Se fabrica grabada por encargo o se escribe una vez.

- **EPROM, EEPROM (erasable programmable) y FLASH**

- Requiere procesos especiales para escritura
- Número limitado de grabaciones, entre 10.000 y 100.000
- La escritura es mucho más lenta que la RAM.
- Las EPROM usan sistemas de programación especiales (tensión, tiempos).
- La lectura si es un proceso rápido.
- La reescritura en caso de Flash puede requerir procesos especiales y claves que hacen más complicadas estas tareas.
 - » Las EPROM requieren borrado mediante luz ultravioleta (sustituidas por EEPROM y Flash).
 - » Las EEPROM de borran con tensiones especiales.



Memoria no volátil

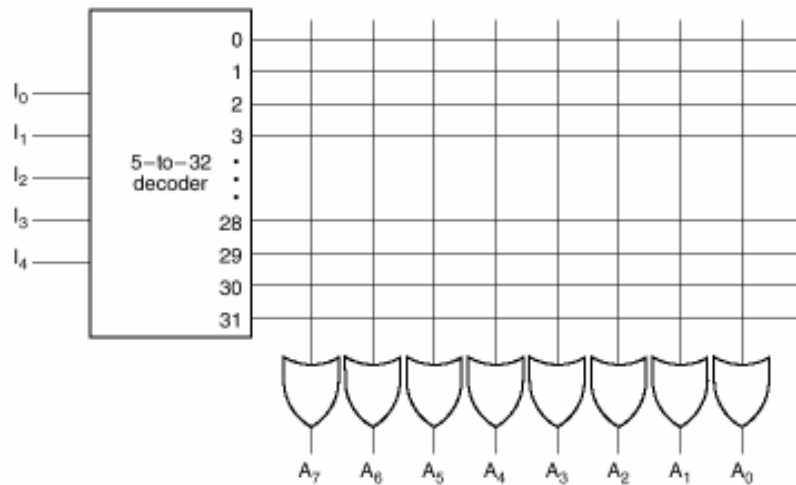
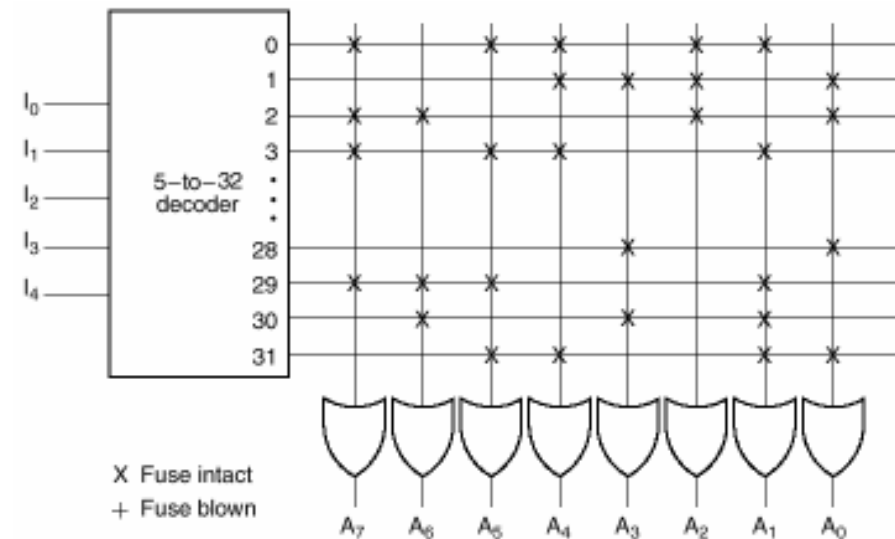


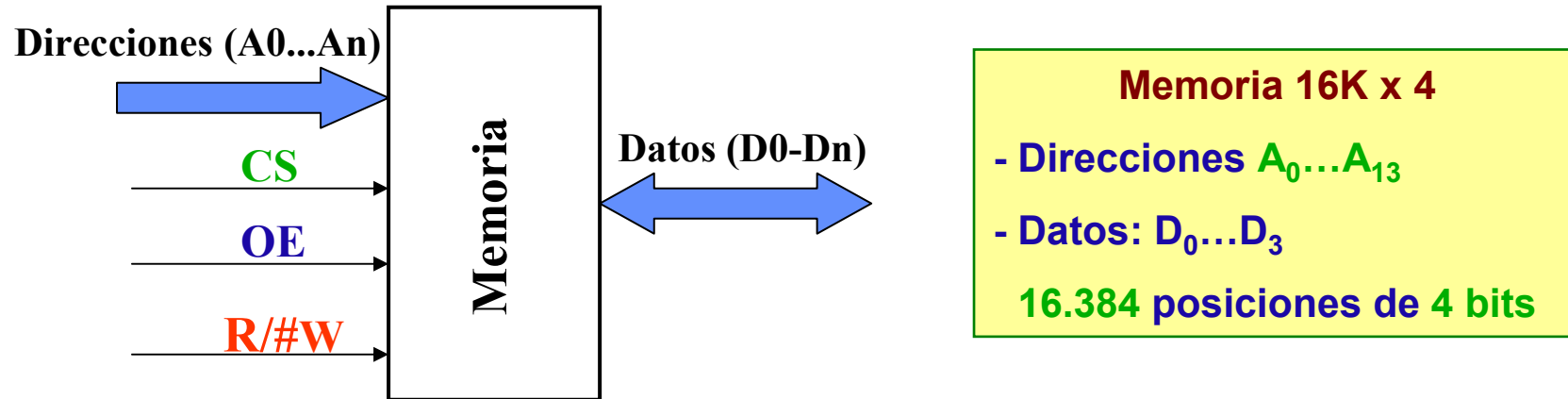
Tabla de la verdad ROM (parte)

Inputs					Outputs							
I ₄	I ₃	I ₂	I ₁	I ₀	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		.							.			
		.							.			
		.							.			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Ejemplo de programación mediante uniones o fusibles en una PROM, una vez que se 'graba', no se puede borrar la información. Las 'x' corresponden a fusibles intactos (continuidad - 1) y los cruces no tienen continuidad (0), al seleccionar la fila.



Memorias: Uso e incorporación a circuitos

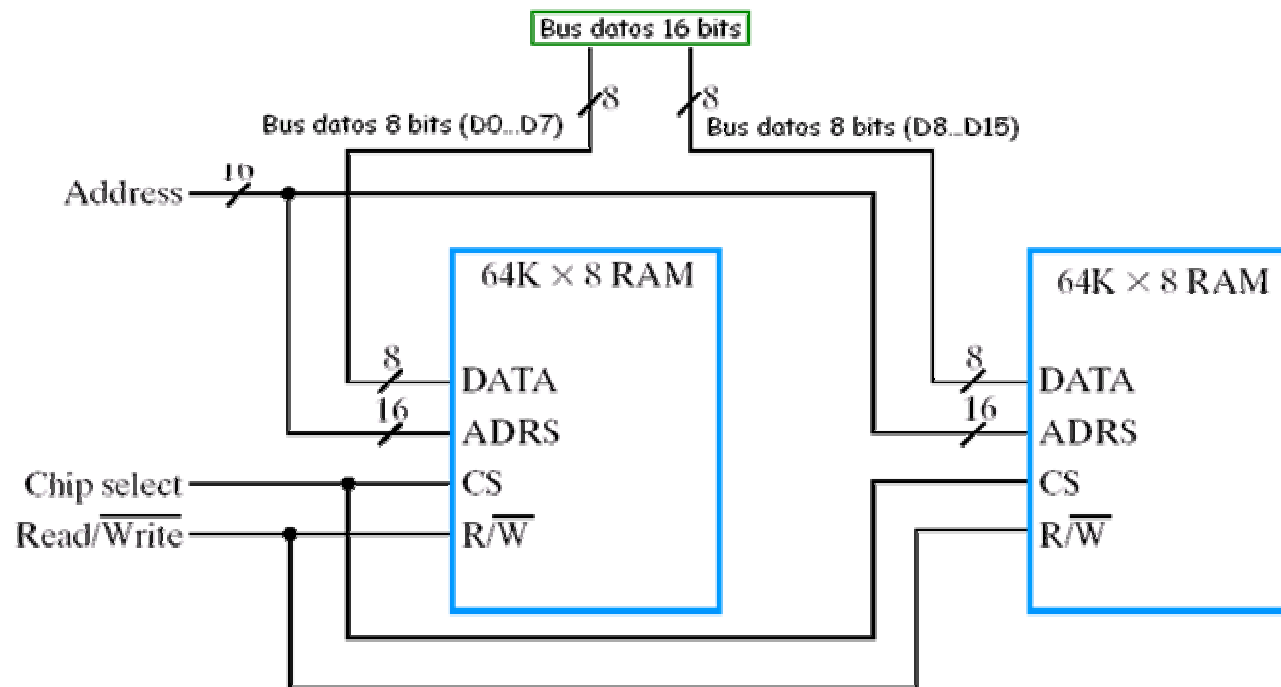


- Las memorias tienen **lineas de direcciones** para indicarnos la '**posición**' del dato a leer o a escribir. La **capacidad** de la memoria en 'posiciones' es 2^n . Por ejemplo, si una memoria tiene 16 lines de direcciones tendría 2^{16} posiciones, o sea 65536. Son **unidireccionales**.
- Las **lineas de datos** nos ofrecen los **datos leídos** o en ellas ponemos los **datos a grabar**, siempre respecto a una 'posición'. El número de lines nos indica la **cantidad de bits** que podemos almacenar por 'posición'. Por ejemplo, 8 lines de datos (D₀-D₇) nos indicaría que en cada posición puedo almacenar 8 bits. Son **bidireccionales**.
- El **resto de las lines, son de control**, pueden variar, pero casi siempre existe CS (chip select) para activar el chip. Si se puede escribir (RAM), existirá también una linea para indicar cuando la operación que queremos realizar es de lectura o de escritura (R/#W).
- **En cualquier caso se consultará el 'datasheet' del chip para ver toda su funcionalidad.**

Ampliando las necesidades de memoria

¿ Que ocurre si necesitamos un ancho de datos mayor ?

- Si necesitamos 16 bits de ancho en lugar de 8 bits de un chip determinado, podemos agrupar varios chips como uno más ancho.
- Las direcciones y líneas de control son comunes a los dos chips.



Memoria 64K x 16

- 2 Chips 64K x 8

- Direcciones:

- $A_0 \dots A_{15}$ comunes

- Datos:

- Chip 1 -> $D_0 \dots D_7$

- Chip 2 -> $D_8 \dots D_{15}$

- Líneas de control:

- Chip select

- Read / #Write

Comunes a todos los chips

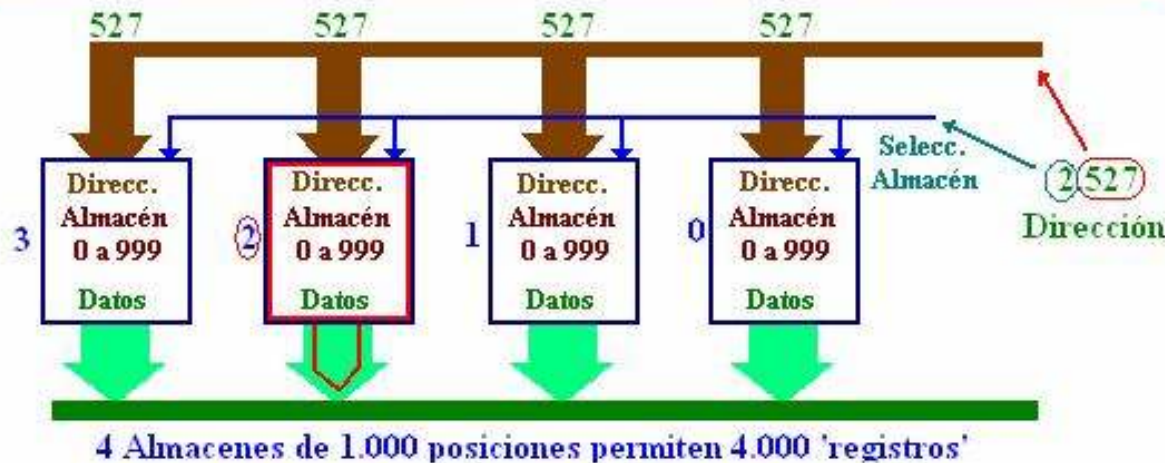
Chips en PARALELO

Ampliando las necesidades de memoria

¿ Que ocurre si necesitamos mayor cantidad de datos?

- Si necesitamos más cantidad de datos, sin modificar el ancho de palabra (bits datos), podemos agrupar varios chips para ocupar más 'posiciones de memoria'.
- Las líneas de datos son comunes, las de direcciones suelen serlo y las de control determinan el '**mapa**' de memoria final.

Ejemplo aumento de 'memoria' de almacenamiento (almacenes)



Almacén 1000 x 4

- 4 Almacenes 1000 pos.
- Direcciones:
- N_0, N_1, N_2 comunes
- Salidas: Comunes
- Líneas de control:
- N_3 (dígito miles)
- Selecc Almacén :
- Determina almacén activado
- En el ejemplo, los tres dígitos de menor valor (527) establecen la posición que vamos a ver en el almacén, pero solo el dígito de mayor valor determina de que almacén cogeremos el dato (2).

MAPA DE ALMACÉN

- 0-999 -> Almacén '0'
- 1.000 a 1.999 -> Almacén '1'
- 2.000 a 2.999 -> Almacén '2'
- 3.000 a 3.999 -> Almacén '3'

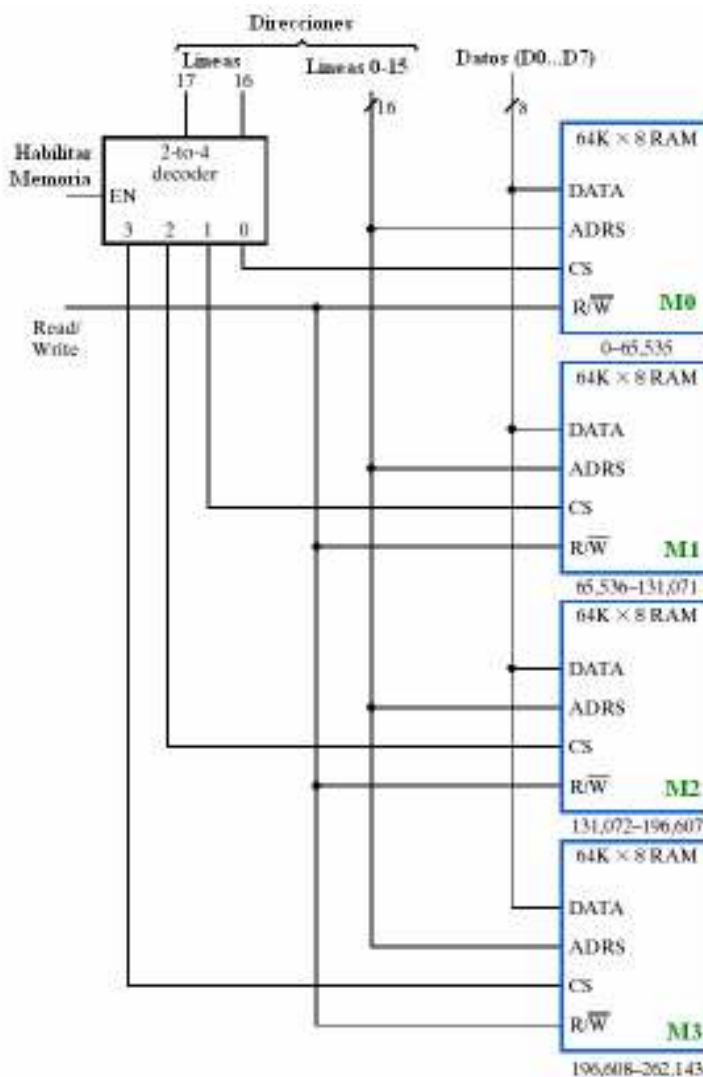
Ampliando las necesidades de memoria

Memoria 256K x 8

- 4 Chips 64K x 8
- Direcciones: $A_0 \dots A_{15}$ comunes
- Datos: Comunes
- Líneas control: $R/\#W$ (común)
- Chip Select :
- Determina que chip se activa
- Un decodificador extrae de las líneas A_{16} y A_{17} la información de que chip tenemos que activar, es como las unidades de millar del ejemplo anterior, en este caso son las 'unidades' o grupos de 64K es decir, 2^{16} por A_0 a A_{15} .

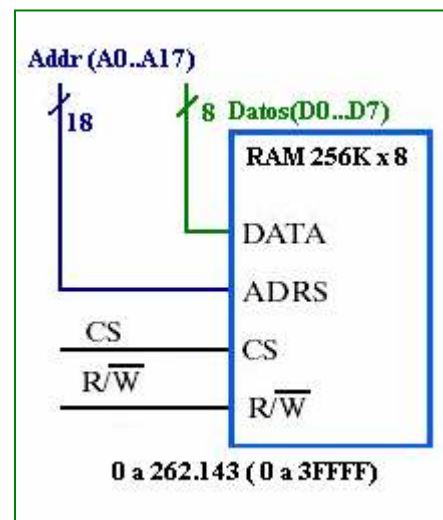
MAPA DE MEMORIA

- 00000 a 0FFFF \rightarrow CS-'0'
- 10000 a 1FFFF \rightarrow CS-'1'
- 20000 a 2FFFF \rightarrow CS-'2'
- 30000 a 3FFFF \rightarrow CS-'3'



Circuito Equivalente

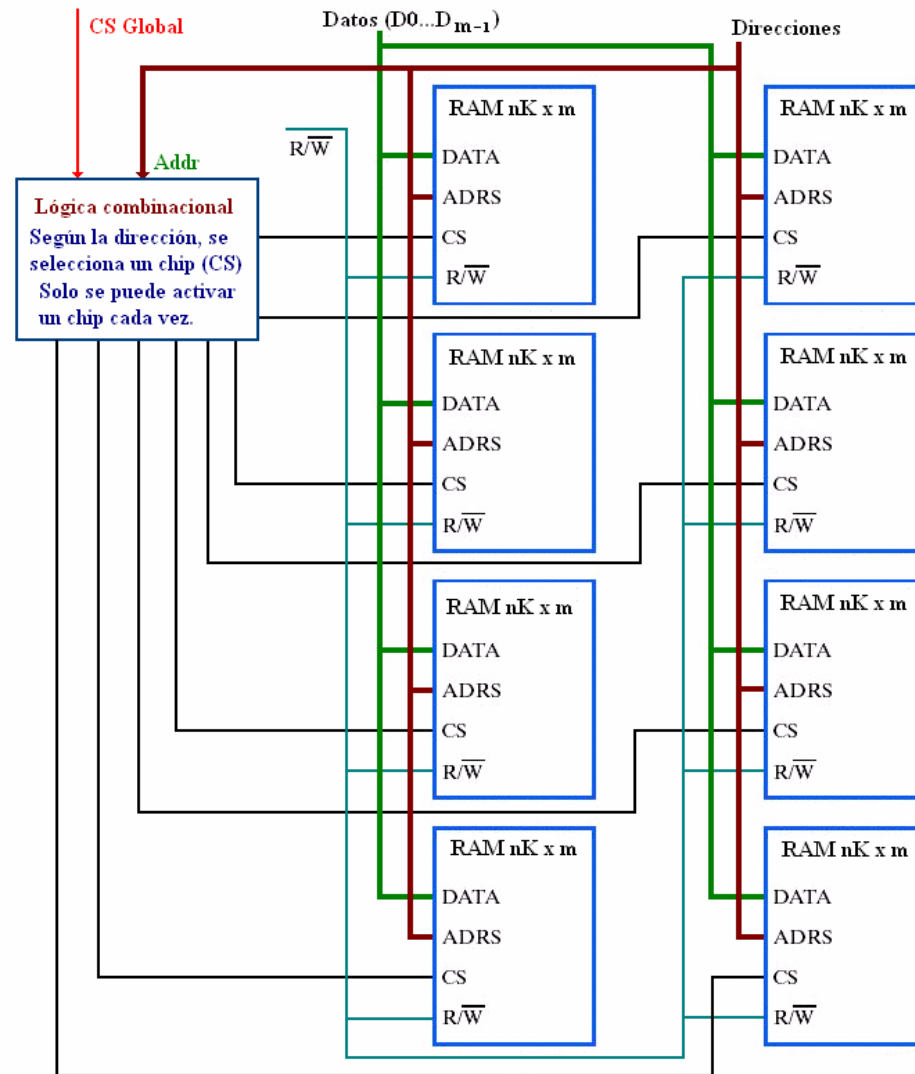
Memoria 256K x 8



Ampliando las necesidades de memoria

Agrupando memorias

- El grupo debe comportarse como una memoria única.
- Las direcciones llegan a todos los chips.
- Las señales R/W van a todos los chips.
- Existe un sistema combinacional que determina, según el mapa de memoria a implantar, que chip debe seleccionar.
- Solo se puede activar un chip cada vez.
- Cada CS tiene una 'fórmula' lógica.
- Suele ser conveniente agrupar las memorias de forma continua, sin dejar huecos.
- Se pueden agrupar RAM, ROM, EPROM, siempre que se respete la funcionalidad.
- Por ejemplo, se puede montar una ROM de 16 K y 3 chips de 16K de RAM, cubriendo un mapa de memoria de 64K.



Ampliando las necesidades de memoria

Ejercicio memorias

- Dibujar esquema del circuito con el siguiente mapa de memoria usando las memorias 2716 y 6116.
- El ancho del bus de datos deseado es de 8 bits.

0x0000	2K (EPROM)
0x07FF	memoria 2716
0x2000	2K (EPROM)
0x27FF	memoria 2716
0xE800	(RAM)
0xFFFF	memoria 6116

MAPA DE MEMORIA

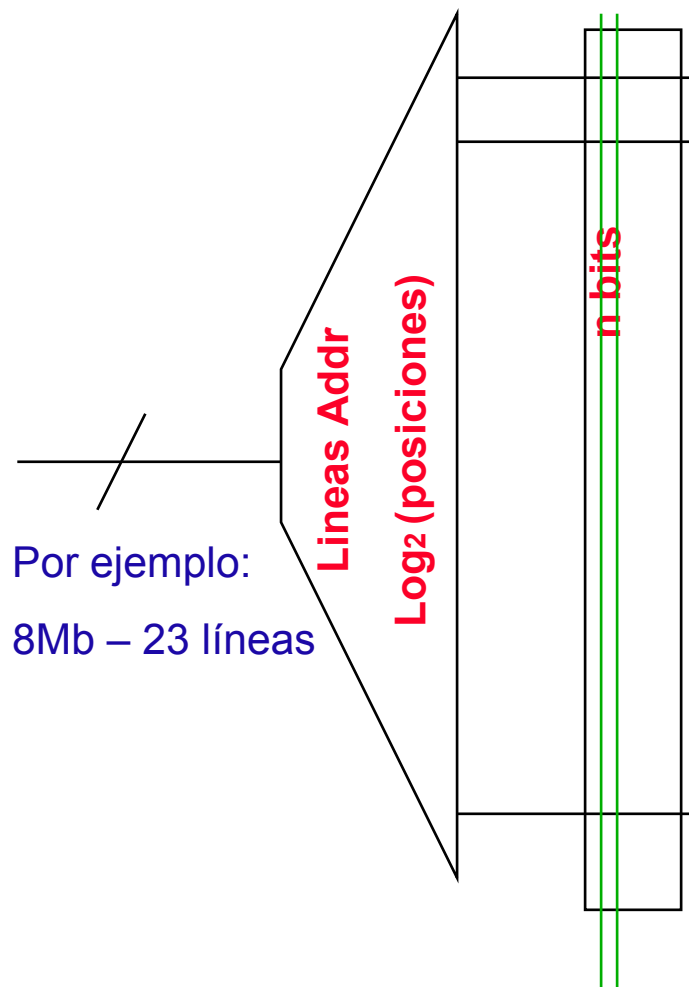
Resolución del problema

- Determinar características (datasheet) de los chips.
- Ver pines a conectar en cada chip.
- Determinar el número de chips a colocar.
- Realizar tabla verdad de CS por chip.
- Determinar esquema de selección de chips.

Es conveniente realizar otros ejercicios con diferentes mapas de memoria y diferentes chips.



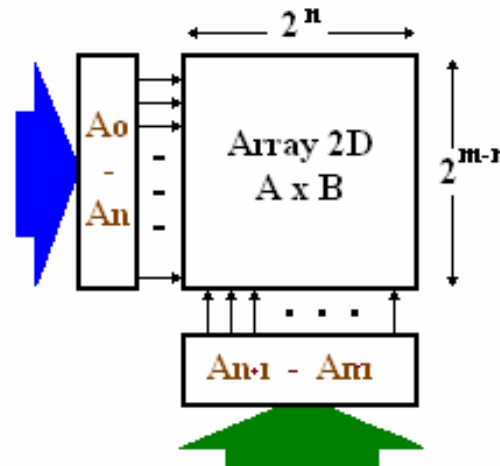
¿Qué pasa cuando n° posiciones es muy grande?



- Decodificador de registro grande y lento
- Líneas de bit largas
 - Carga RC alta, implica menos velocidad de transición en datos.
- La lógica y geometría es muy estrecha y muy larga, por ejemplo 8 x 8.388.608

¡ Hay que hacerlas más anchas y 'cortas' !

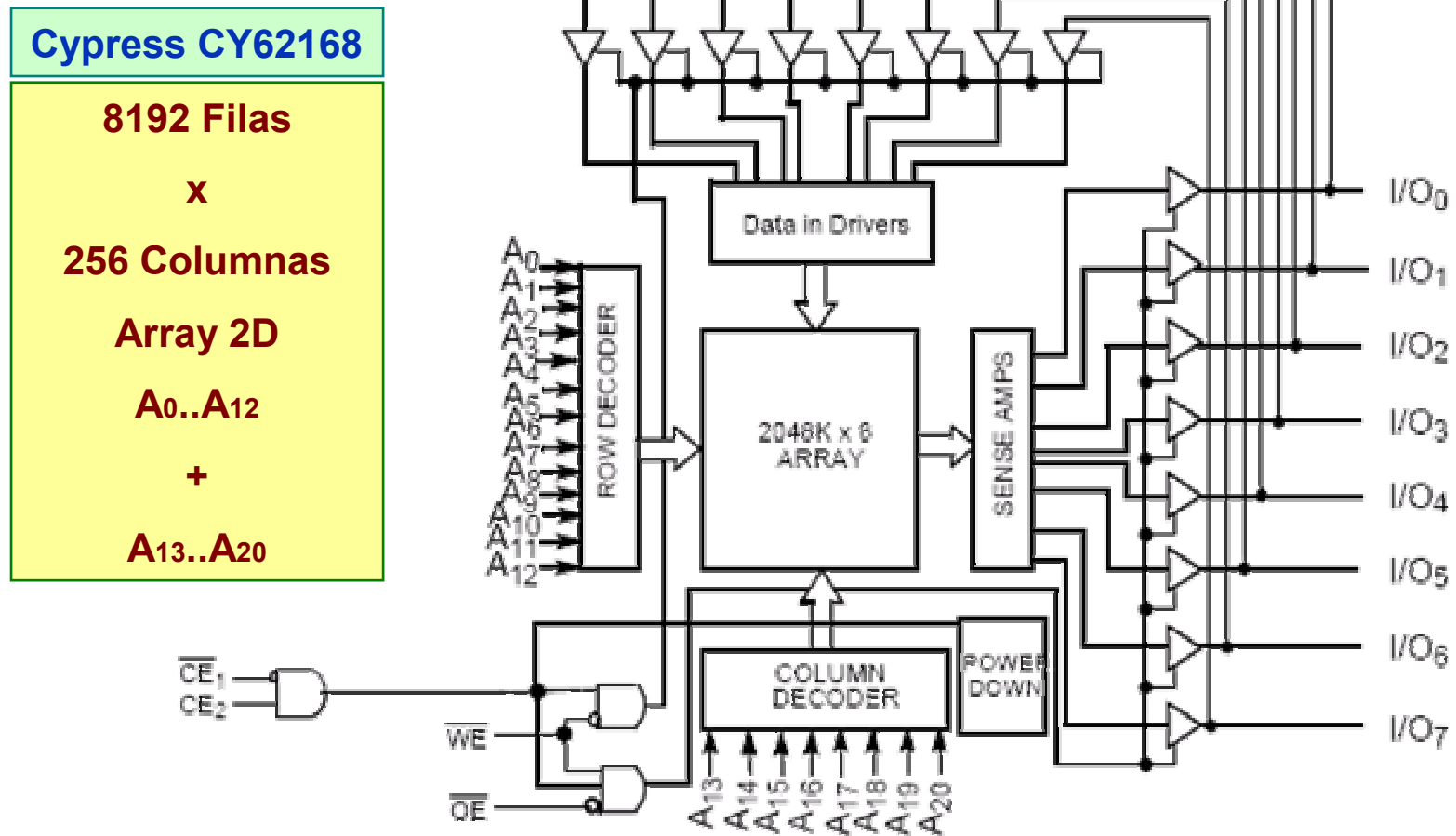
- Para conseguirlo se implementan arrays de células de memoria 2D.



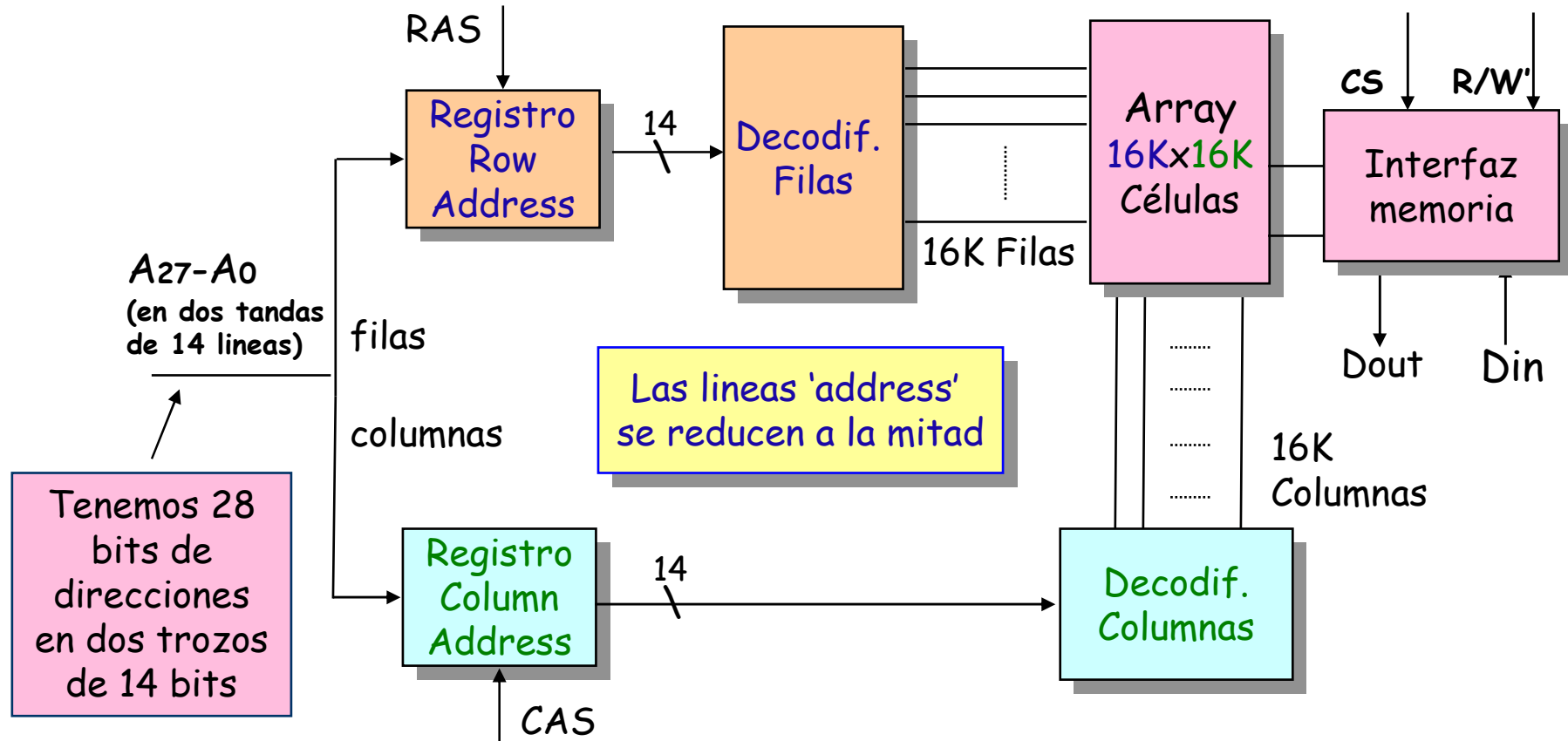
- Geometría mejorada.
- Acceso más rápido.
- Menos RC parásitas.
- 'Pistas' más cortas.
- Velocidad superior.
- Diseño más racional.

Ejemplo: Memoria 2M x 8

Logic Block Diagram



¿ y cuando es mucho más grande ?



RAS = Row Address Strobe , CAS = Column Address Strobe
Con RAS se indica que la primera mitad de la dirección (Row=Fila) está disponible y es correcta, entonces se lee la fila y mientras llega la segunda mitad de la dirección (Column=columna) y cuando está cargada se activa el CAS y se selecciona la columna.

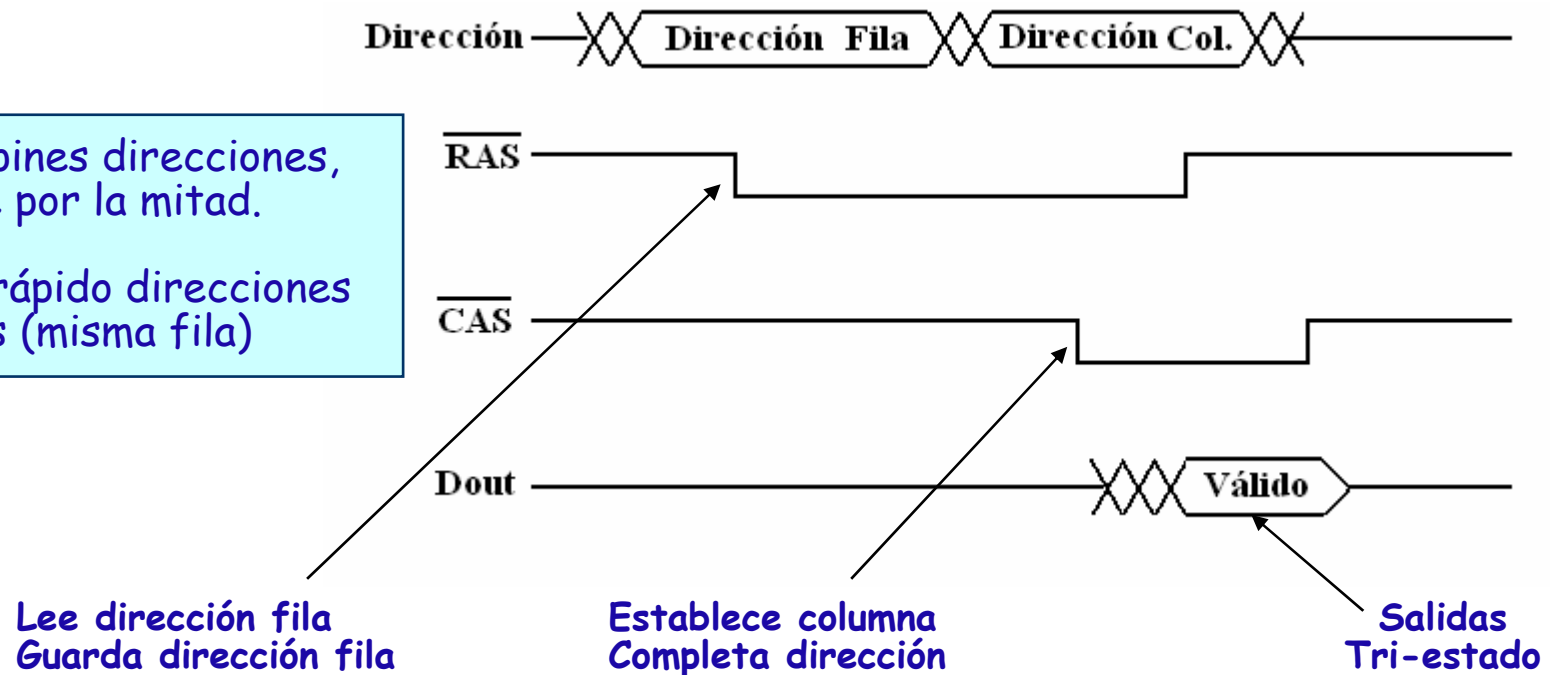
Lectura memoria usando CAS y RAS

Direccionamiento mediante RAS y CAS

- **Direccionamiento en dos ciclos o fases:**
 - Direccionamiento de fila (Row Address)
 - Direccionamiento de columna (Column Address)

Ciclo de Lectura

- Ahorra pines direcciones, las divide por la mitad.
- Acceso rápido direcciones Contiguas (misma fila)



Memorias FLASH

- Son memorias de '**solo lectura**' reescribibles.
- Gran implantación en la actualidad.
- Pen-drives, reproductores MP3, MP4, tarjetas memoria, etc.
- Borrable eléctricamente y programable en sistema (ISP)
- Circuitería en el chip que controlan las tensiones necesarias para borrar y programar los datos
- Los borrados se realizan por sectores (16K - 64K Bytes)
- Las tarjetas SD, Compact flash, ets. Se basan en estas memorias
- La escritura no se realiza igual que la RAM, requiere codificaciones especiales.
- El número de escrituras posibles no es ilimitado y no se garantiza a partir de un número especificado por el fabricante.



Memorias: Características a tener en cuenta

- **Latencia (tiempo de acceso)**
 - El intervalo de tiempo entre el instante en que se solicita la lectura (READ) o se solicite una escritura (WRITE) y el instante en el que la lectura se entrega o se completa la grabación.
- **Tiempo de ciclo**
 - El tiempo transcurrido entre el instante en que se accede a la memoria y el instante en el que se puede acceder de nuevo a la memoria.
- **Ancho de banda (throughput)**
 - Cantidad de datos que pueden ser leídos o escritos en la unidad de tiempo.
 - Inversamente proporcional al tiempo de ciclo

Como mejorar estas características

- **La latencia puede ser reducida:**
 - Seleccionando chips con menores tiempos de acceso
 - Usando memoria caché (se verá en temas posteriores A.S.I.)
- **El Ancho de Banda puede ser incrementado:**
 - Bus de datos más ancho (más chips en paralelo)
 - Más pines de datos por chip seleccionado (memorias con bus datos más ancho)
 - Seleccionando chips más rápidos.

Mapas de memoria

Memoria que es capaz de direccionar un dispositivo que usa memoria microprocesador, microcontrolador, etc.

◆ Distribución de la misma, es decir qué direcciones ocupan los diferentes dispositivos destinados a funciones determinadas con respecto al dispositivo.

◆ Nota: A veces los mapas de memoria incluyen hardware como si fuese una memoria de lectura / escritura.

La especificación del mapa de memoria se puede realizar cómo:

Funcional: ubicación (direcciones) de los elementos (hardware o software) del sistema digital, atendiendo a la función de los mismos. Así se describirán la ubicación de: sectores de programa, posición de datos generales y tablas, registros de interfaz, etc.

Físico: correspondencia entre las direcciones del mapa y los dispositivos físicos en el que se plasman. De acuerdo a él se realizará la conexión entre los diferentes dispositivos, teniendo en cuenta la estructura del bus de direcciones y del bus de datos, la forma de selección de dispositivos, etc.

Mapas de memoria

Mapa de Memoria		
Funcional	Físico	Direcciones
Programa y tablas de datos fijos.	Pastilla ROM de 8Kbytes. Circuito Integrado tipo....	0x0000 0x1FFF
<i>Zona vacía (Ningún chip ocupa estas posiciones)</i>	---	0x2000 0x3FFF
Dispositivos de Entrada/Salida	Registros varios: Circuitos Integrados(registros) Display LCD, de 64 caracteres	0x4000 0x403F
<i>Zona vacía</i>	---	0x4400 0x8FFF
Teclado hexadecimal y registros auxiliares.	Teclado de membrana: modelo	0x9000 0x9003
<i>Zona vacía</i>	---	0x9004 0xBFFF
Variables y tablas temporales(4Kbytes)	Pastilla RAM de 8Kbytes C.I. tipo....	0xC000 0xCFFF
Zona de datos transferencia serie (4Kbytes).		0xD000 0xDFFF

DIRECCIÓN
BASE

Mapas de memoria

Tareas fundamentales:

- ◆ **Diseño del mapa funcional**, donde se decide en que direcciones del mapa se van a colocar las diferentes zonas de memoria y variables del sistema. Puede venir dado por las características del dispositivo.
- ◆ **Diseño del mapa lógico** donde se seleccionan los dispositivos más adecuados para cumplir el mapa de memoria.
- ◆ **Diseño de la lógica de selección** para acceder a la celda o posición de memoria deseada y no a otras. Activar los elementos que intervienen en la operación y seleccionar (CS) el dispositivo adecuado.
 - A partir de una dirección que aparece en el bus, activa la línea de selección del chip al que corresponde dicha dirección.
 - Puede diseñarse empleando puertas lógicas, decodificadores, etc, o empleando dispositivos programables como PLD, PAL, etc.

Mapas de memoria

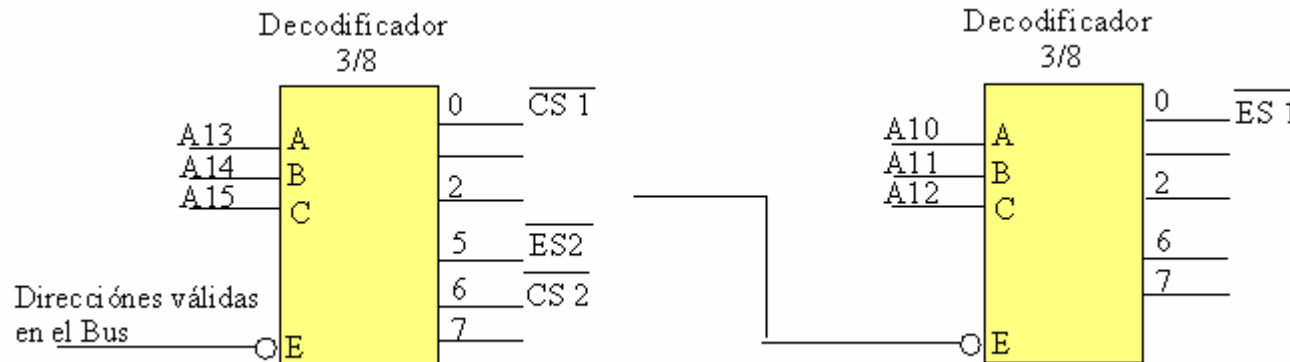
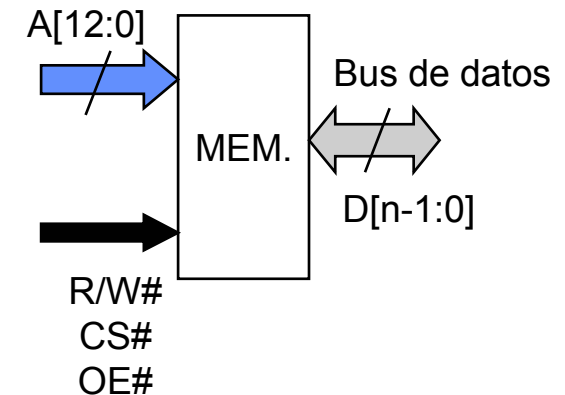
◆ Proceso

Tener en cuenta la **dirección base**

Reconocer la función de los diferentes bits de direcciones.

P.e. para el elemento (pastilla RAM de 8 Kbytes) las líneas A[12..0] se necesitan para seleccionar la posición deseada dentro del chip, y por tanto pueden presentar cualquier código, lo que se representa como "XX...X".

El resto de las líneas A[15..13], para que la dirección presente en el bus se corresponda a este chip, deben presentar el código "1 1 0", *C000 h a DFFF h*



Mapas de memoria

Decodificación completa:

Se emplea este término cuando el acceso a una posición concreta de memoria se puede realizar sólo para una única combinación de los bits del bus de direcciones; esto es: *{una posición física = una dirección lógica}*.

Decodificación incompleta:

Se corresponde a aquella situación en la que el acceso a una posición concreta de memoria se puede producir con varias combinaciones diferentes de los bits del bus de direcciones: *{una posición física = varias direcciones lógicas}*.

