

CLASE 1

EL MODELO DE VON NEUMANN (INTRODUCCIÓN) NIVELES DE ABSTRACCIÓN

Intervalo 15'

MEMORIAS: DISEÑO Y ORGANIZACIÓN

El modelo de von Neumann (1945)



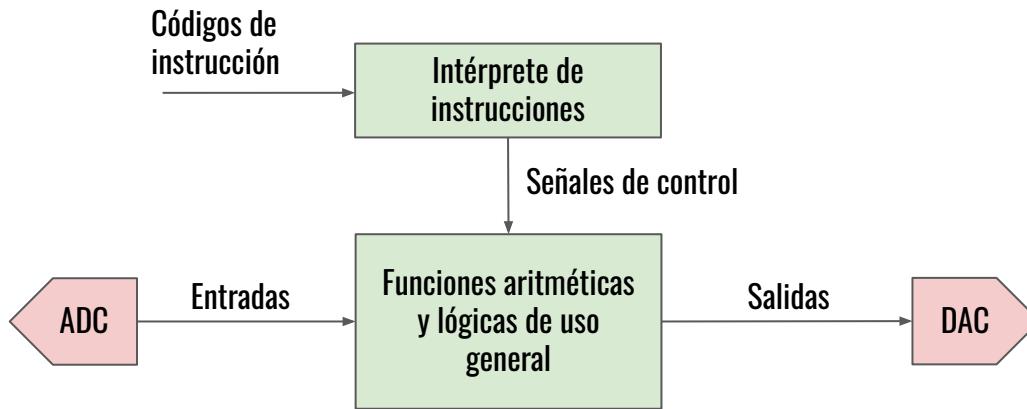
La aproximación por software versus la aproximación por hardware

El modelo de von Neumann (1945)



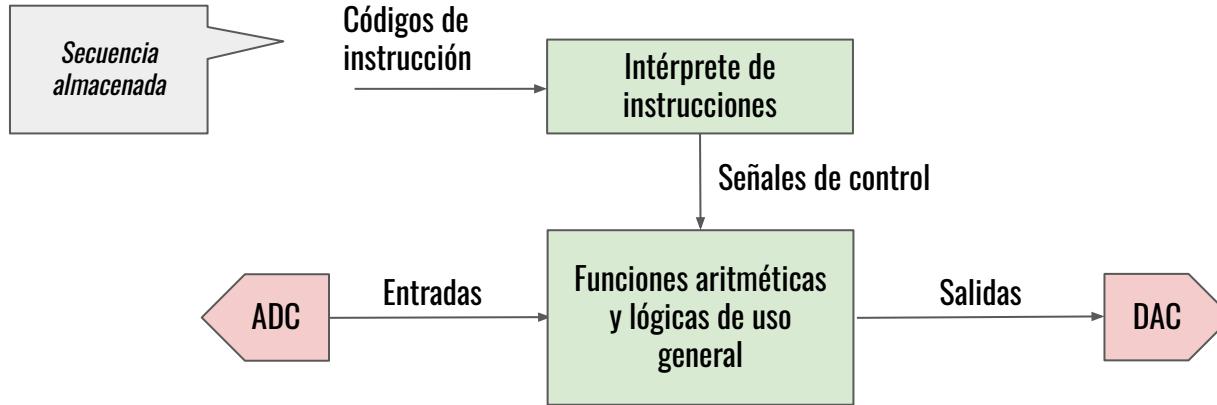
La aproximación por software versus la aproximación por hardware

El modelo de von Neumann (1945)



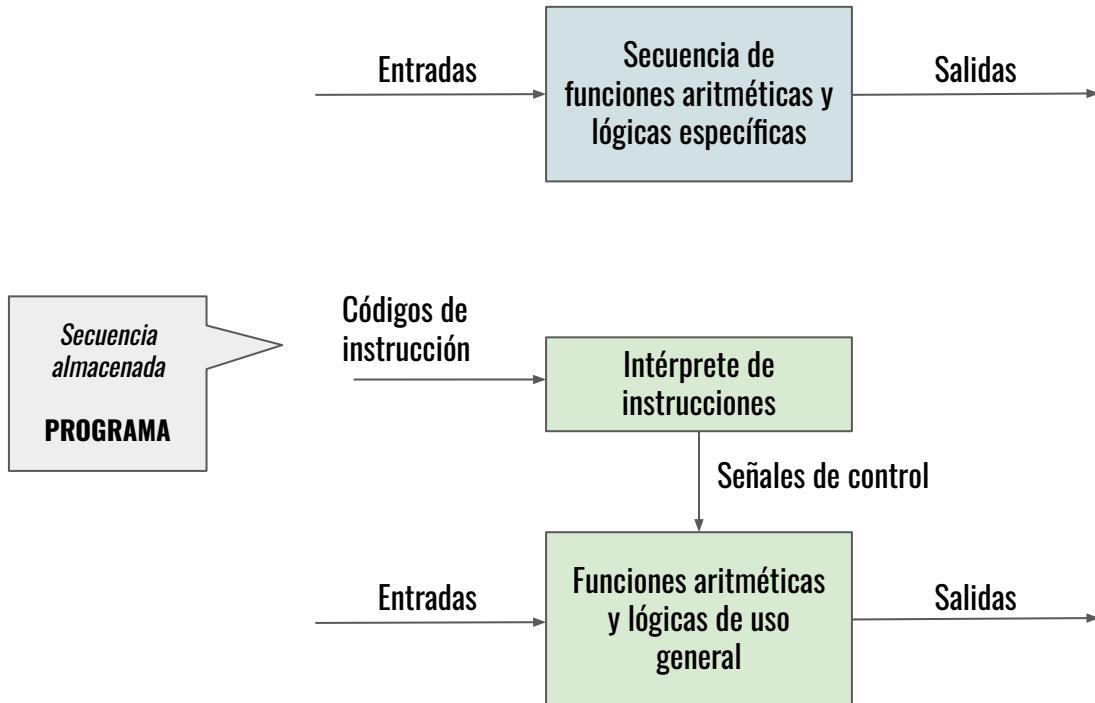
La aproximación por software versus la aproximación por hardware

El modelo de von Neumann (1945)



La aproximación por software versus la aproximación por hardware

El modelo de von Neumann (1945)



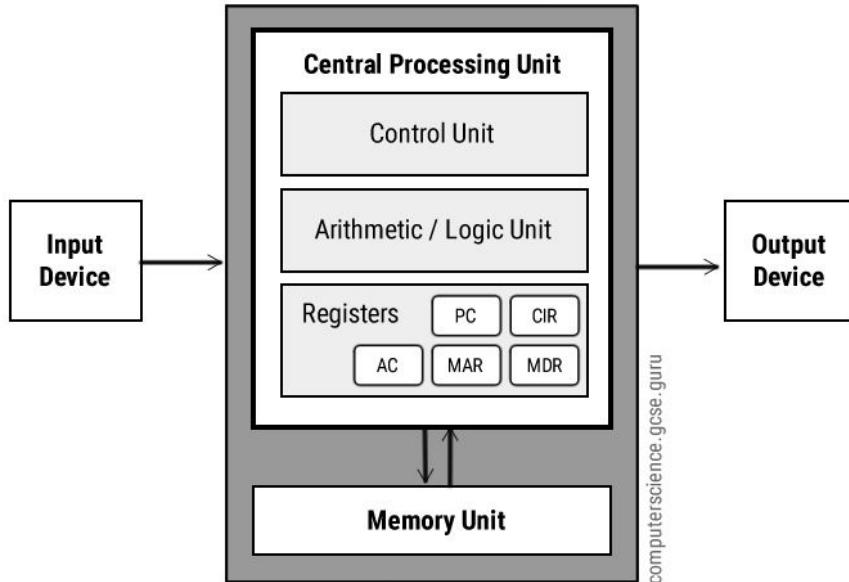
AGREGA VENTAJAS ADICIONALES A LOS SISTEMAS DIGITALES

Insensibilidad al entorno (por ejemplo temperatura).
Insensibilidad a la tolerancia de los componentes.
Inmunidad al ruido.
Predictibilidad y comportamiento repetitivo (como consecuencia de las dos propiedades anteriores).
Pueden resolverse, en forma económica, problemas que en el campo analógico serían muy complicados (por ejemplo filtros adaptivos).

- + Reprogramabilidad
- + Tamaño constante

La aproximación por software versus la aproximación por hardware

El modelo de von Neumann (1945)



Diferentes diseños gráficos,
nosotros tendremos el
nuestro...

Elementos

Unidad central de procesamiento (CPU)

- Ejecución secuencial
- Datapath:
 - ALU (unidad aritmético-lógica)
 - Registros
- Unidad de control

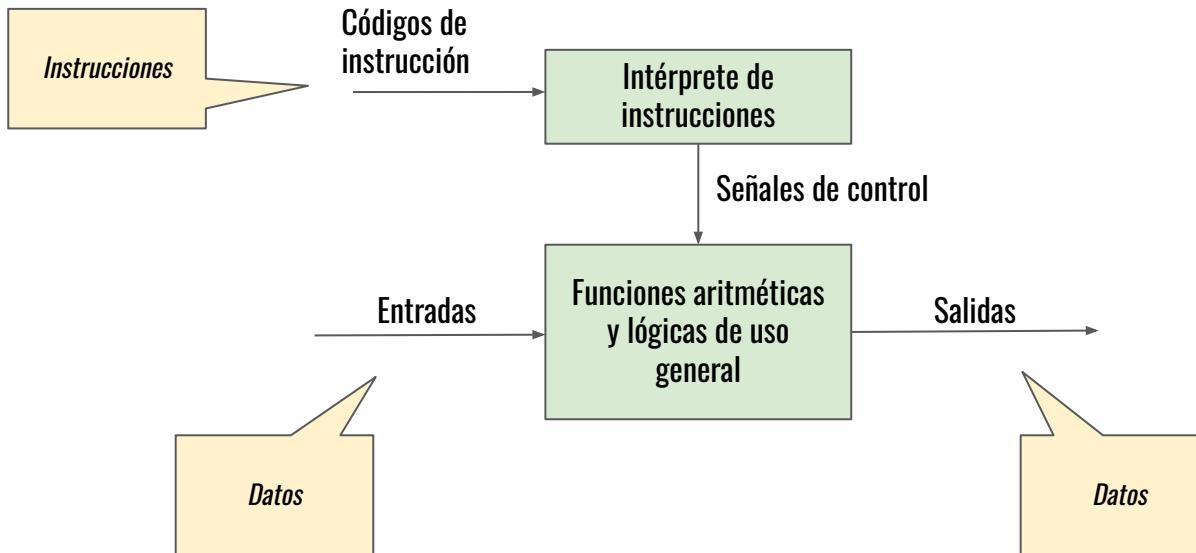
Unidad de memoria

Almacenamiento de las **instrucciones** y los **datos** en una **única memoria** (el mismo espacio lógico).

Mecanismo de entrada-salida

En principio, disponemos de todos los elementos de hardware necesarios para diseñar la CPU.

El modelo de von Neumann (1945)

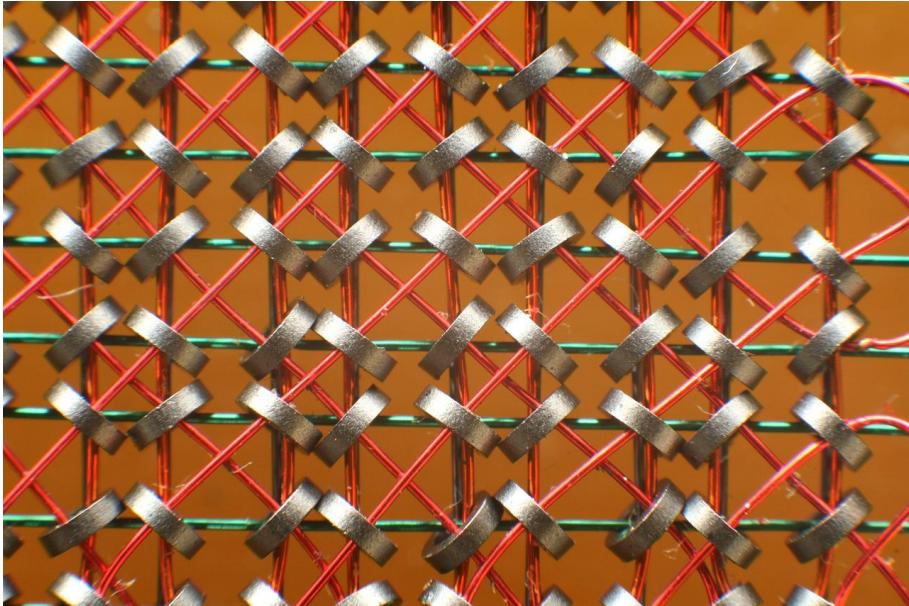
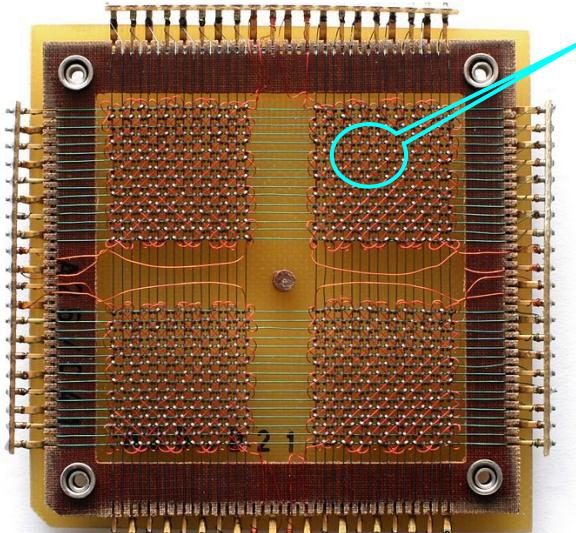


IMPORTANTE

No confundir las **instrucciones** y los **datos**.

En el modelo de von Neumann ambos están almacenados en una **misma memoria**.

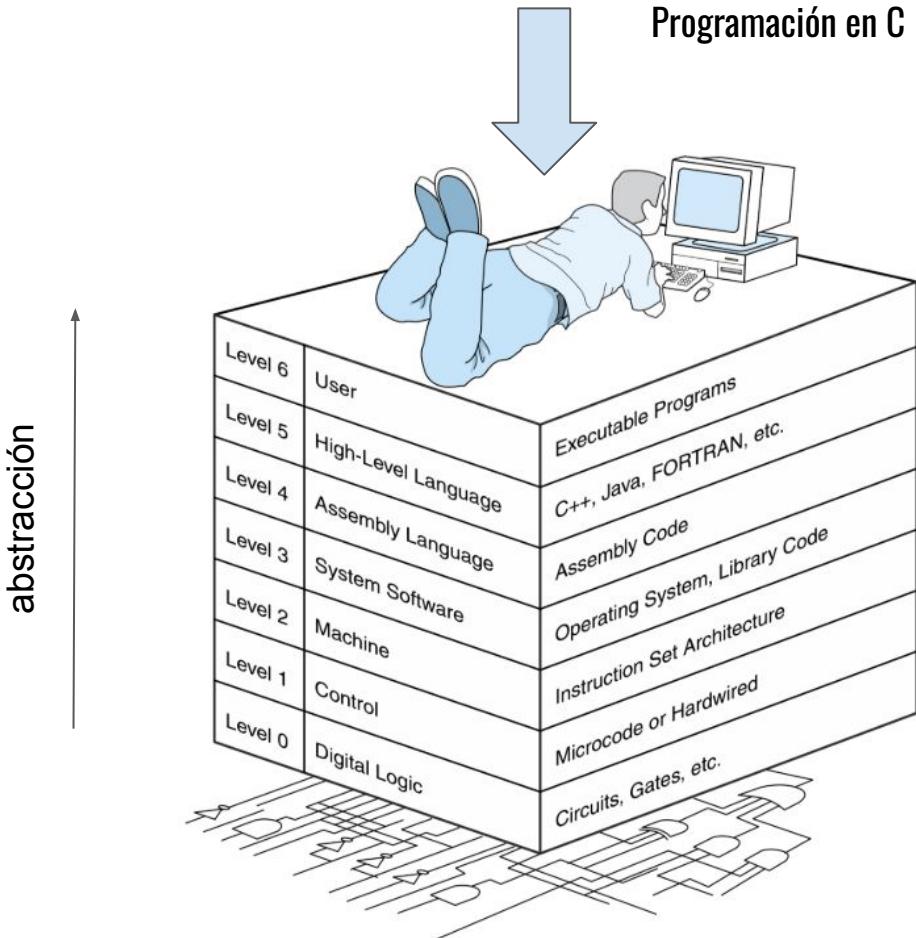
micro SD
128 GB



Magnetic Ferrite Core Memory 32x32

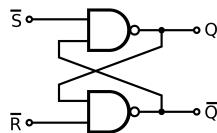
A 32 x 32 core memory plane storing 1024 bits (or 128 bytes) of data. The small black rings at the intersections of the grid wires, organised in four squares, are the ferrite cores.

Niveles de abstracción ADVERTENCIA

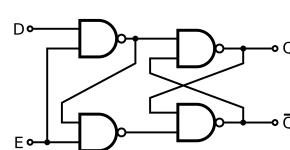


Estos niveles están poco detallados, sobre todo en las capas inferiores. Subniveles.

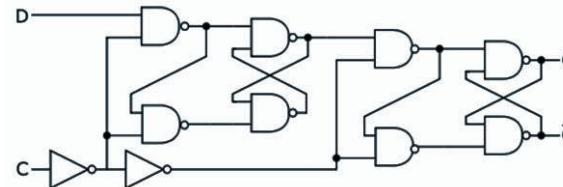
Nivel de abstracción 0: El elemento de almacenamiento Transistor / Compuerta / Latch-FlipFlop / Circuitos básicos



COMPUERTA

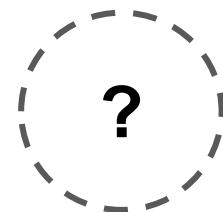
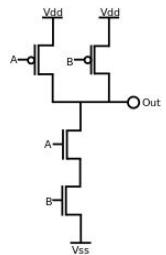


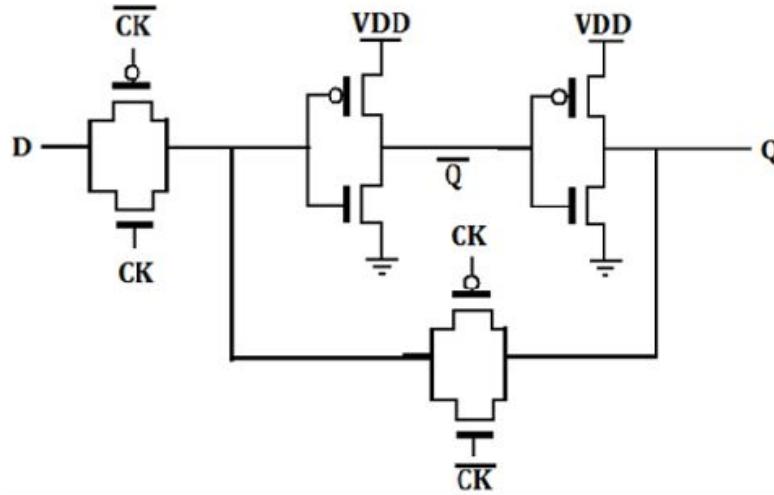
LATCH



GATED D LATCH

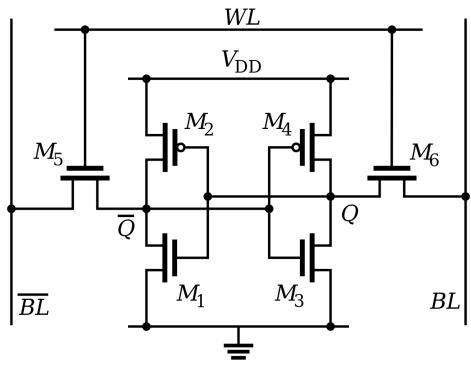
D FLIP-FLOP



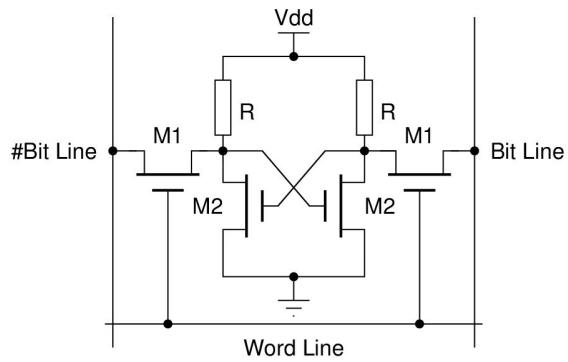


CMOS D Latch Implementation

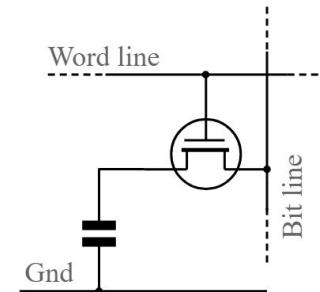
https://www.tutorialspoint.com/vlsi_design/vlsi_design_sequential_mos_logic_circuits.htm



**CELDA CMOS SRAM DE 6
TRANSISTORES**

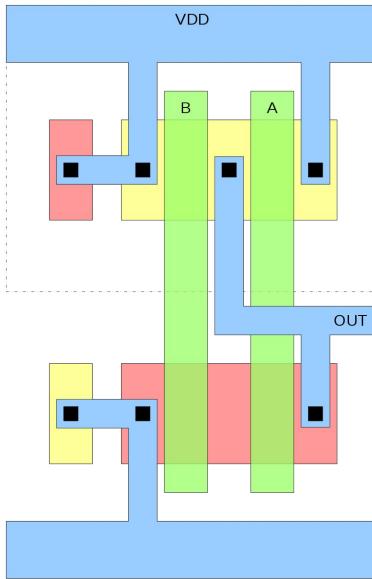
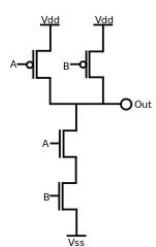


**CELDA CMOS SRAM DE 4
TRANSISTORES (y 2 R)**



CELDA DRAM

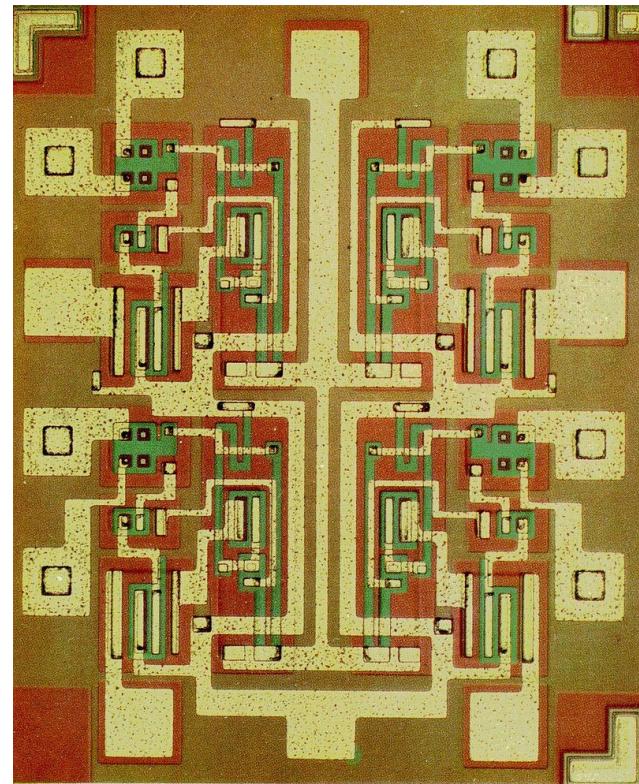
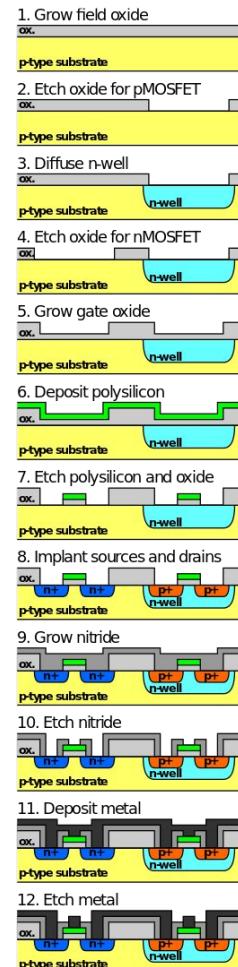
Function	Transistor count	Function	Transistor count
NOT	2	MUX 2-input	8
Buffer	4	MUX 4-input	24
NAND 2-input	4	MUX 2-input with TG	6
NOR 2-input	4	MUX 4-input with TG	18
AND 2-input	6	1-bit Adder full	28
OR 2-input	6	1-bit Adder–subtractor	48
NAND 3-input	6	Latch, D gated	8
NOR 3-input	6	Flip-flop, edge triggered D with reset	12
XOR 2-input	6	8-bit multiplier	3,000
XNOR 2-input	8	16-bit multiplier	9,000
		32-bit multiplier	21,000



METAL1	N DIFFUSION
POLY	P DIFFUSION
CONTACT	N-WELL

Disposición física de una compuerta NAND

Proceso de fabricación CMOS simplificado



Quad NAND CMOS 4011

Tecnología del proceso de fabricación

Procesador	Número de transistores	Fecha de lanzamiento	Proceso de fabricación	Área
Intel 4004	2,300	1971	10 µm	12 mm ²
Core i7	3,000,000,000	2014	14 nm	400 mm ²



Más de 6 órdenes de magnitud !

- 10 µm – 1971
- 6 µm – 1974
- 3 µm – 1977
- 1.5 µm – 1981
- 1 µm – 1984
- 800 nm – 1987
- 600 nm – 1990
- 350 nm – 1994
- 250 nm – 1996
- 180 nm – 1999
- 130 nm – 2001
- 90 nm – 2003
- 65 nm – 2005
- 45 nm – 2007
- 32 nm – 2009
- 22 nm – 2012
- 14 nm – 2014
- 10 nm – 2016
- 7 nm – 2018
- 5 nm – 2019
- 3 nm – ~2021

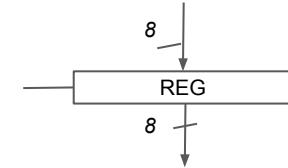
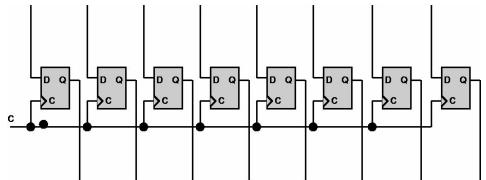
Ley de Moore
el número de transistores se duplica cada dos años - El costo se duplica cada cuatro años

Nivel de abstracción 0+: La unidad funcional ALU/Registro

REGISTRO DE 8 BITS

8 latches/f-f con el mismo enable/clock

Almacena 1 número/palabra de 8 bits
Ancho (wide) 8, 16, 32, 64. Etc.

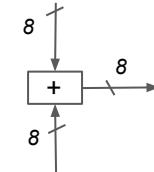
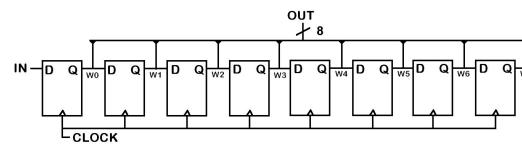


REGISTRO DE DESPLAZAMIENTO CONTADOR

8 latches/f-f en cascada

Ancho (wide) 8, 16, 32, 64. Etc.

Ver: barrel shifter



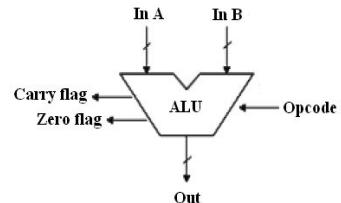
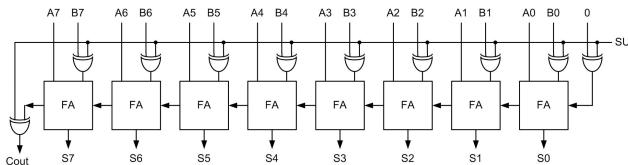
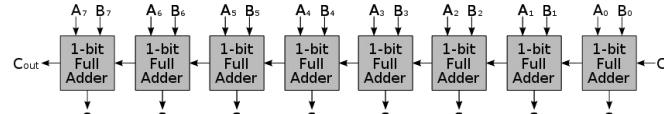
SUMADOR DE 8 BITS

8 full adders en paralelo

Suma 2 números/palabras de 8 bits

Ancho (wide) 8, 16, 32, 64. Etc.

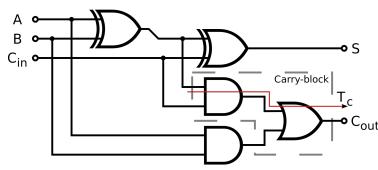
PRACTICA: predictor de carry



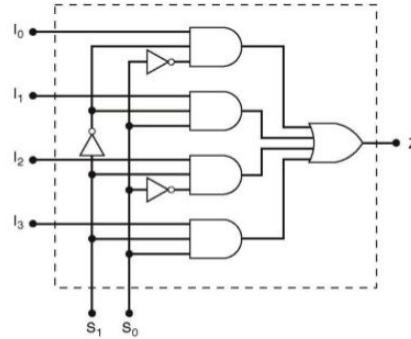
ALU DE 8 BITS

Diferentes operaciones entre 2 números/palabras de 8 bits

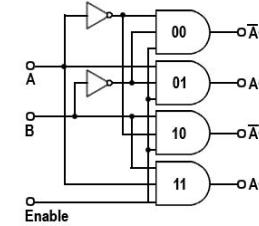
Ancho (wide) 8, 16, 32, 64. etc.



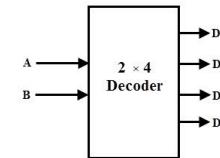
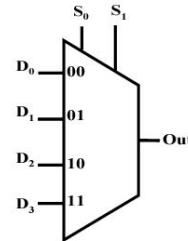
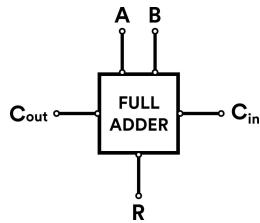
SUMADOR DE UN BIT CON
CARRY (FULL ADDER)



MULTIPLEXOR 4:1



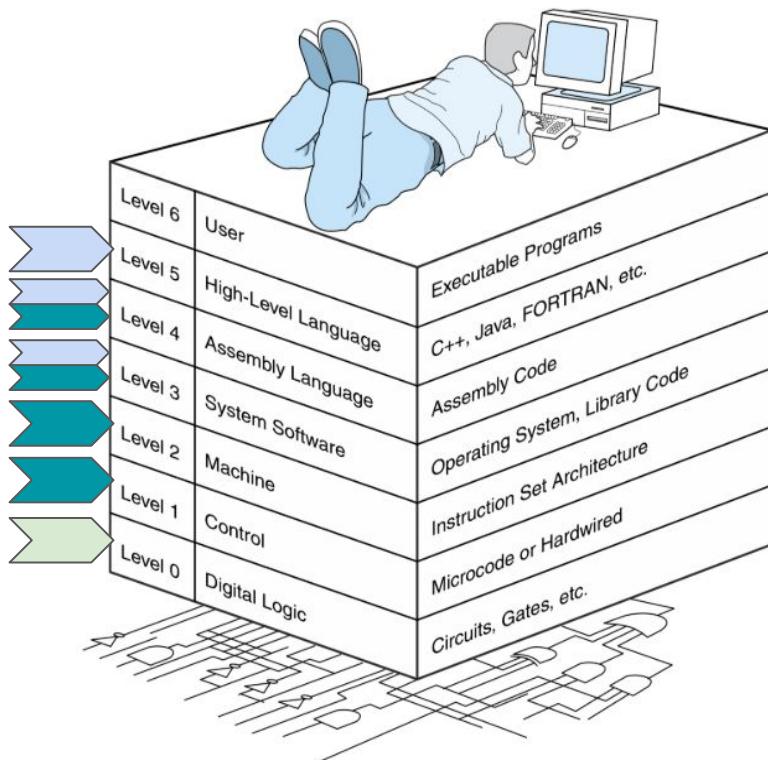
DECODIFICADOR 2:4



Niveles de abstracción ADVERTENCIA

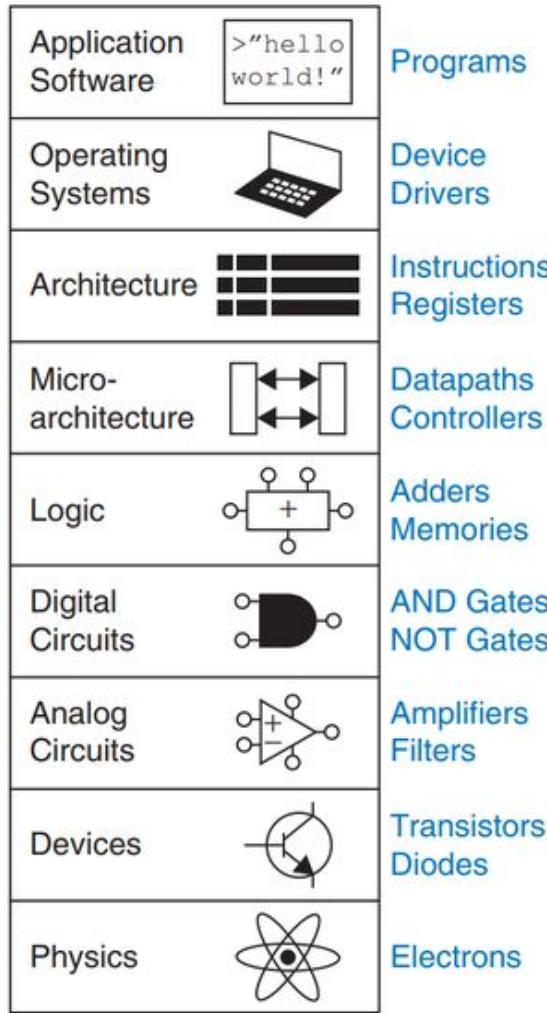
Programación en C

Nivel 0



Nivel 0

Nivel 0	
0-2	La unidad funcional ALU/Registro
0-1	El elemento lógico Compuerta/Flip-flop
0-0	Electrónica MOSFET



Niveles de abstracción

https://wiki.osdev.org/Basic_Theory_of_Computer_Science

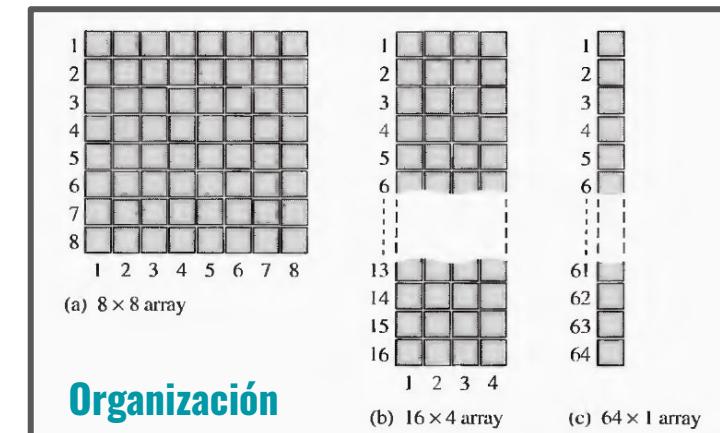
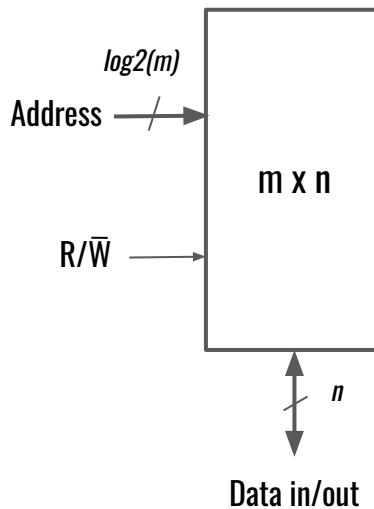
Intervalo
15'

A continuación:
MEMORIAS: DISEÑO Y ORGANIZACIÓN

MEMORIAS: DISEÑO Y ORGANIZACIÓN

Modelo abstracto de una memoria ($m \times n$)

- Arreglo de m celdas consecutivas, cada una capaz de almacenar n bits.
- Cada una de las m celdas tiene una identificación única (**dirección**), por medio de la cual puede ser accedida para almacenar o recuperar información de la misma.
- La dirección está codificada en binario, utilizando $\log_2(m)$ bits.
- Los n bits contenidos en cada celda (**datos**) se leen o escriben simultáneamente.
- Solo una de las m direcciones puede ser accedida por turno, o bien para lectura o para escritura.



UNIDADES

$$k = 10^3 = 1000$$

$$K = 1024 = 2^{10} \text{ (ver Kibibyte KiB)}$$

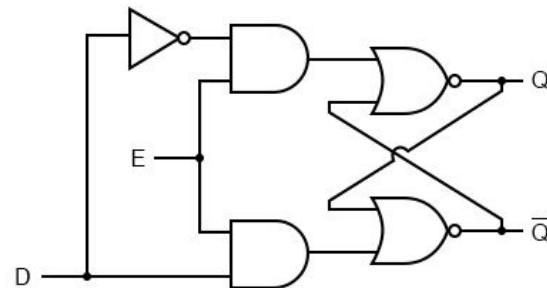
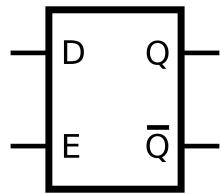
$$Kbit = Kb = 1024 \text{ bits} = K$$

$$Kbyte = KB = 8 \times 2^{10} = 8192 \text{ bits}$$

$$Kword = KW (16, 32, 64, \text{etc.})$$

Elemento de almacenamiento estático

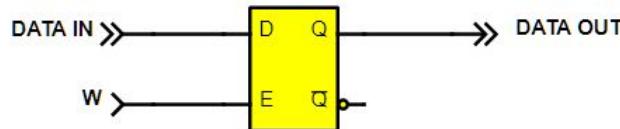
Gated Latch



E	D	Q	\bar{Q}
0	0	latch	latch
0	1	latch	latch
1	0	0	1
1	1	1	0

8 transistores

Elemento de almacenamiento estático

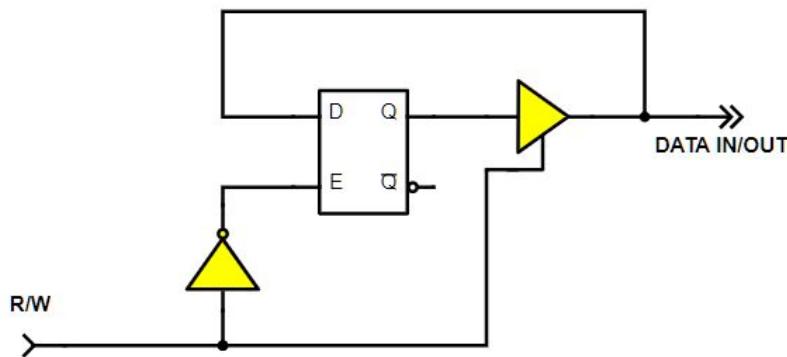


Ejemplo:

Almacenamiento de 4 Kbits usando 4096 flip-flops tipo D (FF-D)

Si quiero tener acceso irrestricto a cada posición de memoria debería disponer 4096 líneas de datos de entrada y 4096 de salida, además de 4096 líneas de control.

Elemento de almacenamiento estático



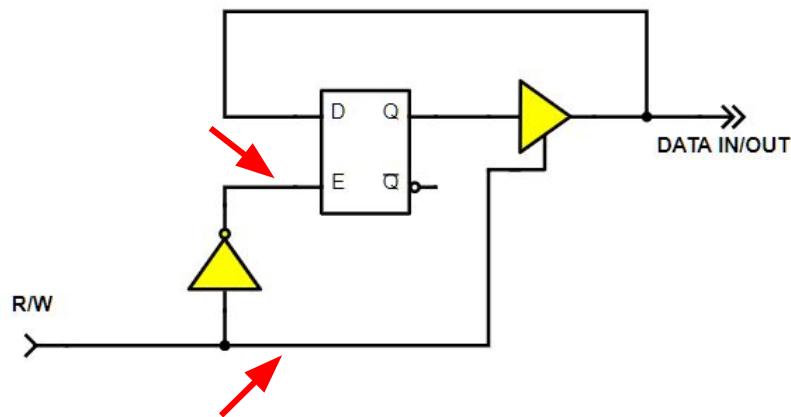
Optimización 1:

Unificación de las líneas de entrada y de salida de datos (o leo o escribo, selección con una línea única que puede ser común para todos los latches).

Una entrada R/~W

Una entrada/salida DATA

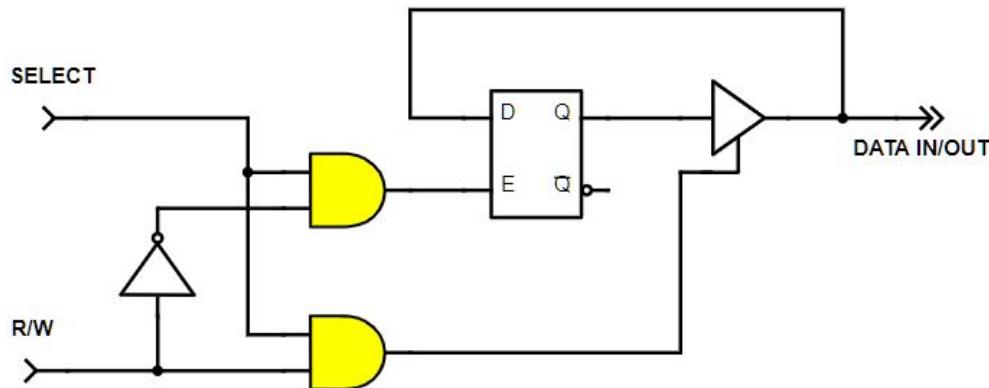
Elemento de almacenamiento estático



Optimización 2:

Se agrega lógica que permita seleccionar el elemento.

Elemento de almacenamiento estático



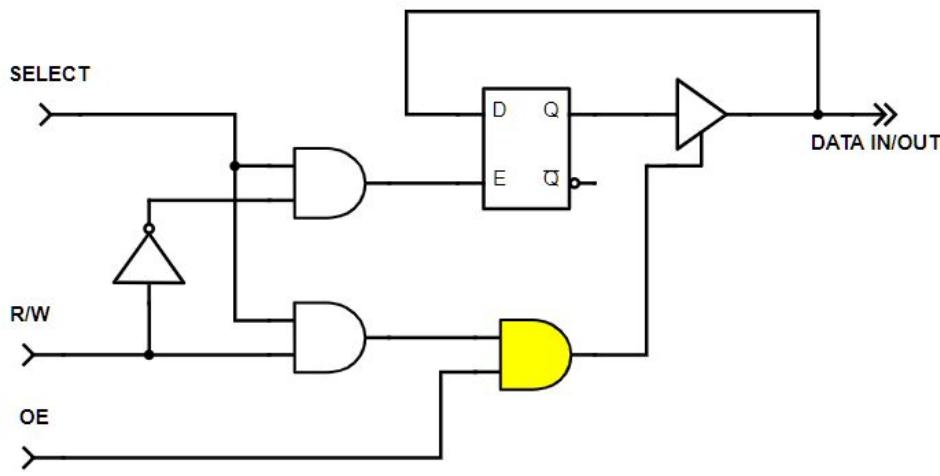
Optimización 2:

Se agrega lógica que permita seleccionar el elemento.

Una entrada adicional: SELECT

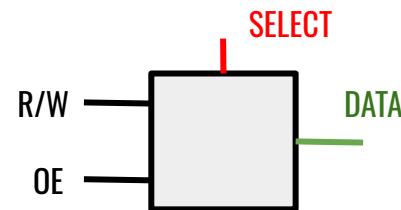
Así puede compartirse la línea de DATA entre varios elementos. Solo toma el control el elemento seleccionado.

Elemento de almacenamiento estático

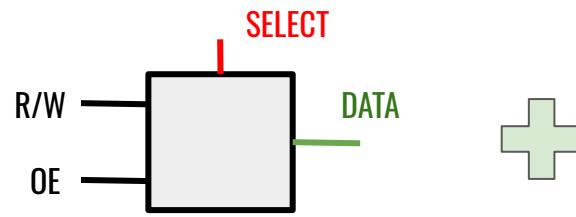


Optimización 3:

Suele agregarse una línea adicional que habilita la salida OE (Output Enable). Útil para la interconexión con buses. No confundir con SELECT.

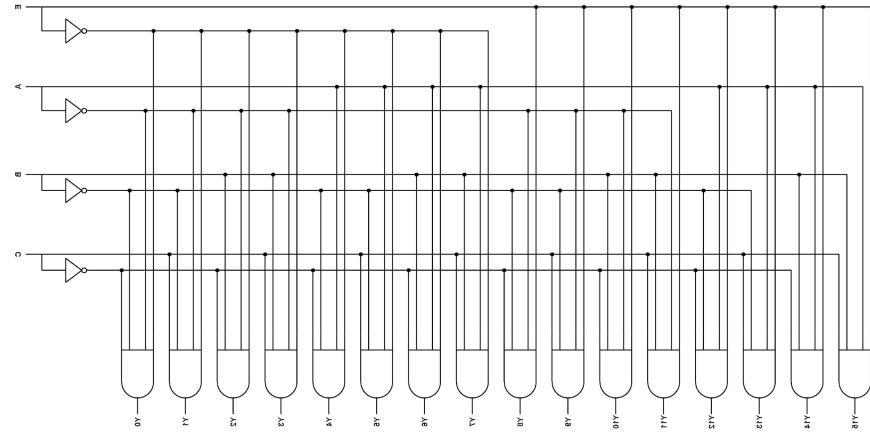


Memoria Organización 16x1



16 elementos

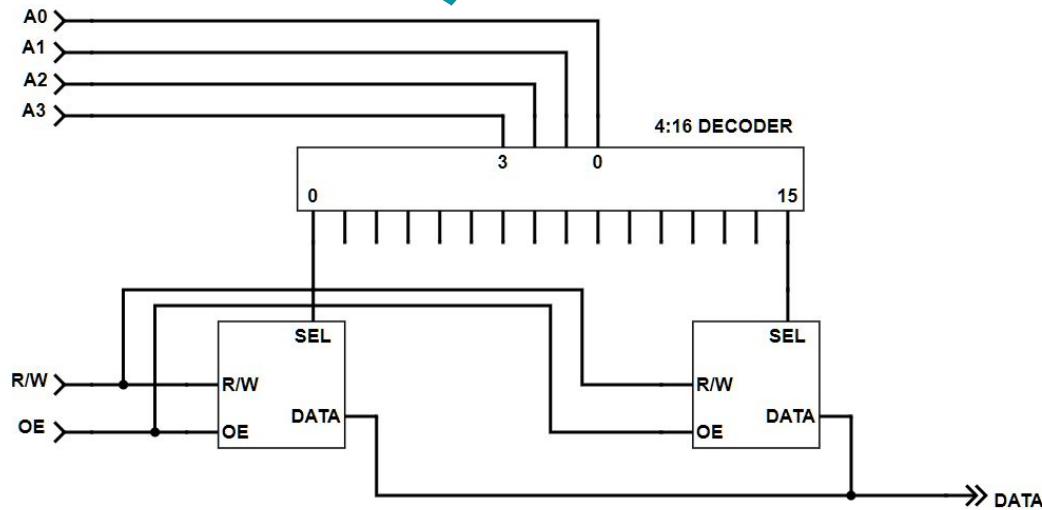
Ejemplo 2: 4Kx1, 12:4096



Decodificador de direcciones 4:16

Organización unidimensional y bidimensional 16x1

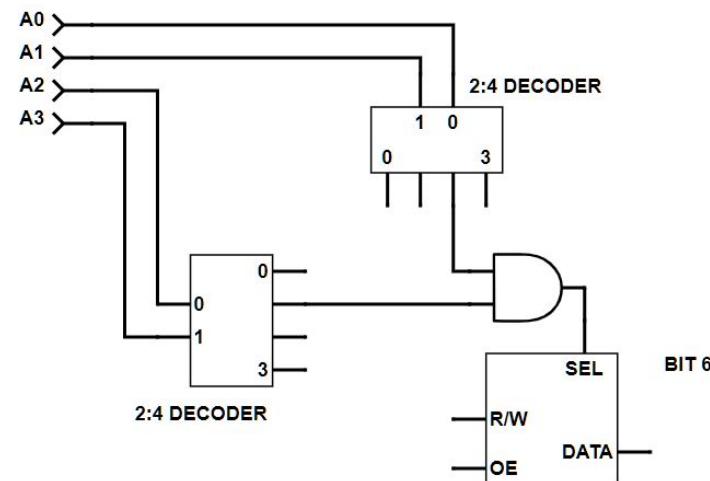
Compromiso: acceder de a una posición por vez.
Todos los elementos en paralelo y seleccionar sólo uno. Una línea de SELECT por cada posición de memoria. Pero aún así éste puede ser un número enorme, por ejemplo en una memoria de 4K x 1 tiene 4096 líneas de SELECT.



Para 4K elementos ($4096=64 \times 64$):

Opción a) requiere un decodificador 12:4096

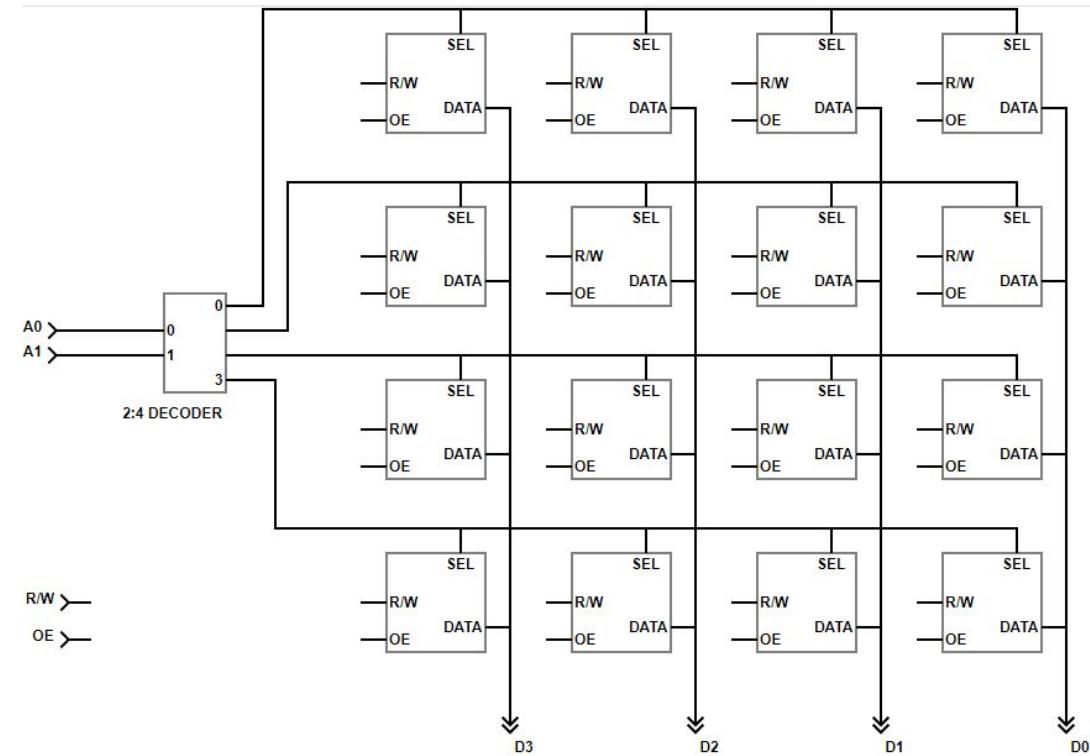
Opción b) requiere dos decodificadores de 6:64 + 4096 compuertas adicionales



Se puede agregar una compuerta más al Select del elemento y arreglarlo en dos dimensiones.

Por ejemplo, una memoria de 4 Kb se resuelve como una matriz de 64 x 64. Resultan $64+64=128$ líneas versus 4096.

Organización por palabras 4x4



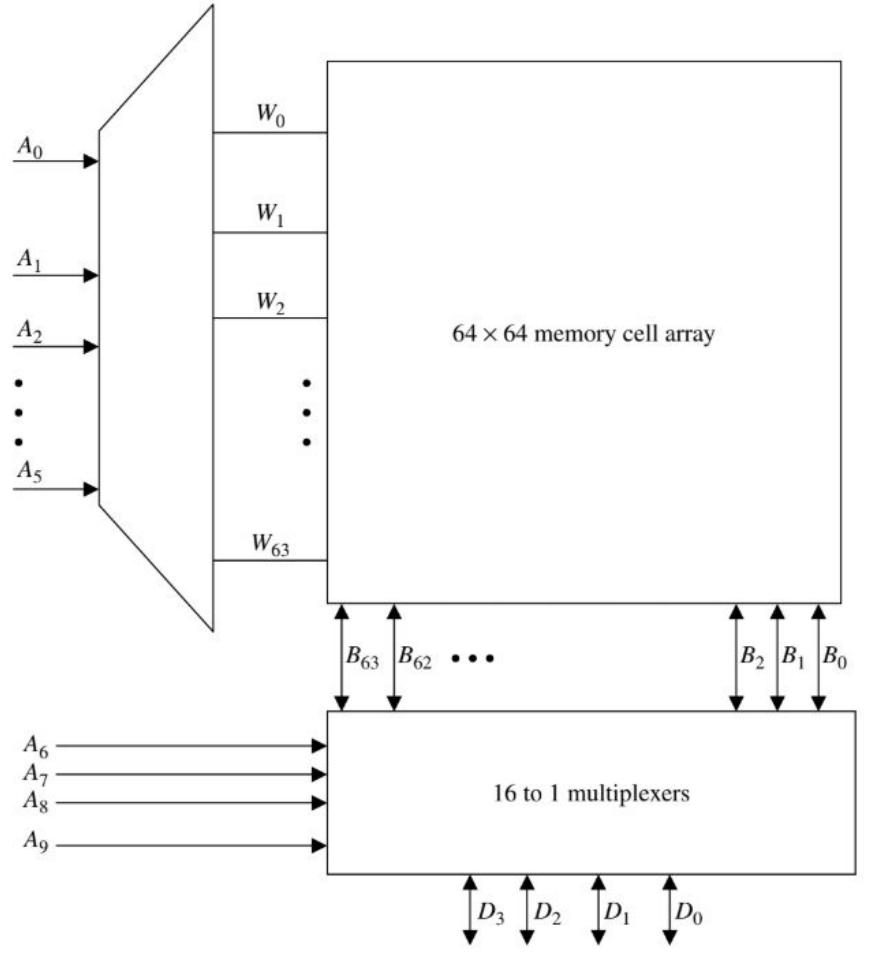
Si no es necesario acceder a cada elemento individualmente, se simplifica el diseño utilizando acceso por palabras (filas/celdas/posiciones): $m \times n$

$4096 \times 1 \rightarrow 512 \times 8$

Requiere un decodificador 9:512

Sigue siendo conveniente organizar la memoria internamente como una matriz de 64×64 :
OPTIMIZACIÓN

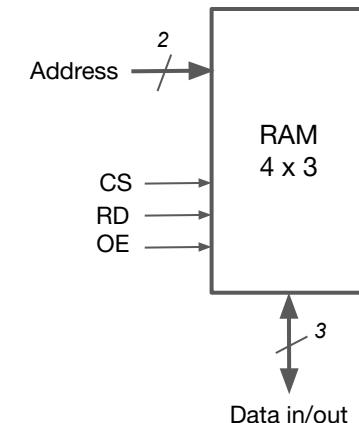
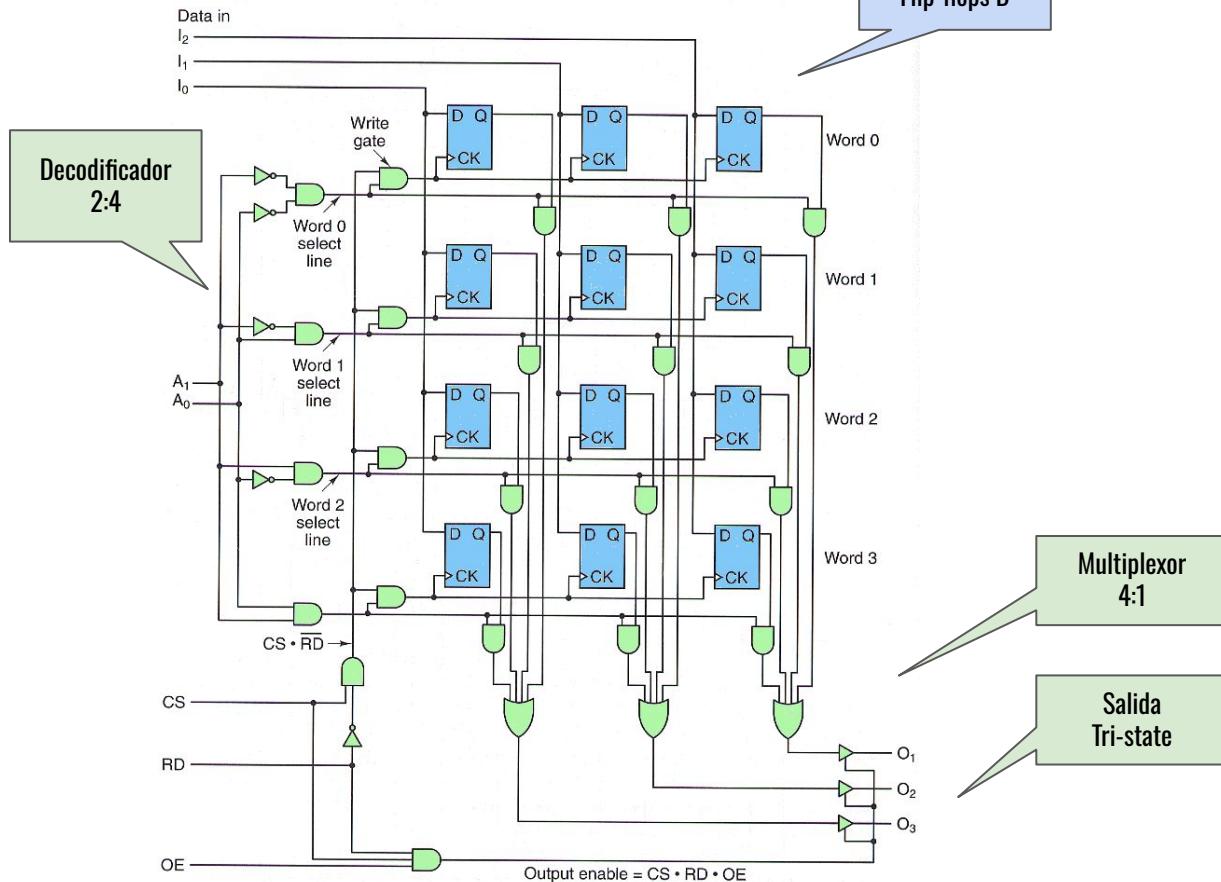
Memoria Optimización



D	n	A
1	4096	12
4	1024	10
8	512	9
16	256	8
64	64	6

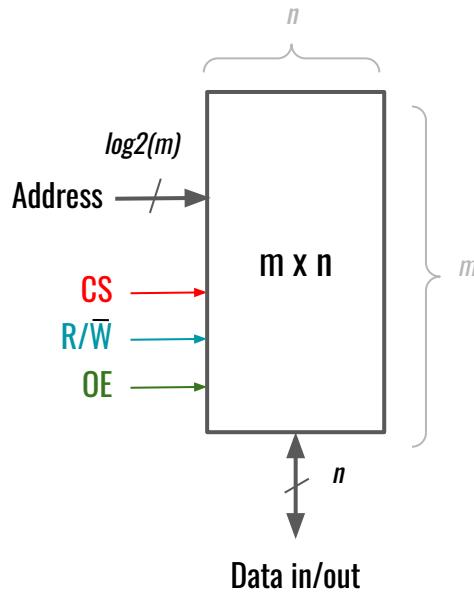
1K x 4
Organización interna eficiente

Ejemplo del libro (L. Null, cap. 3)



Memoria

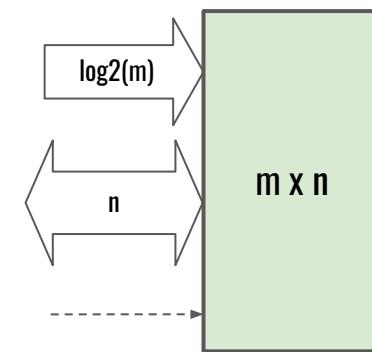
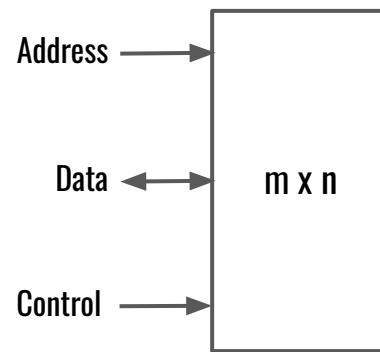
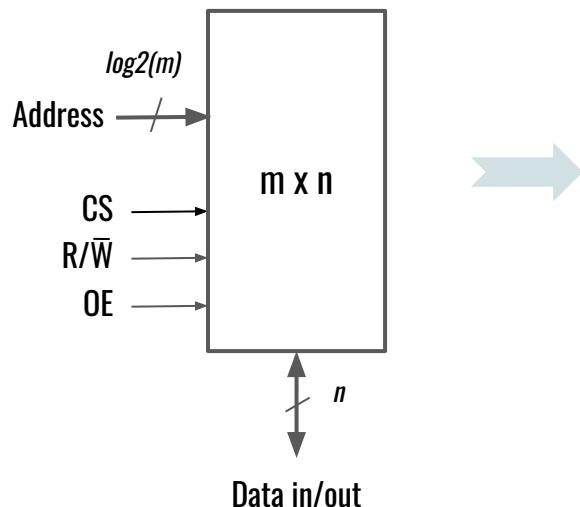
Funcionalidad - Chip Select (CS) Output Enable (OE) Read/Write (R/W)



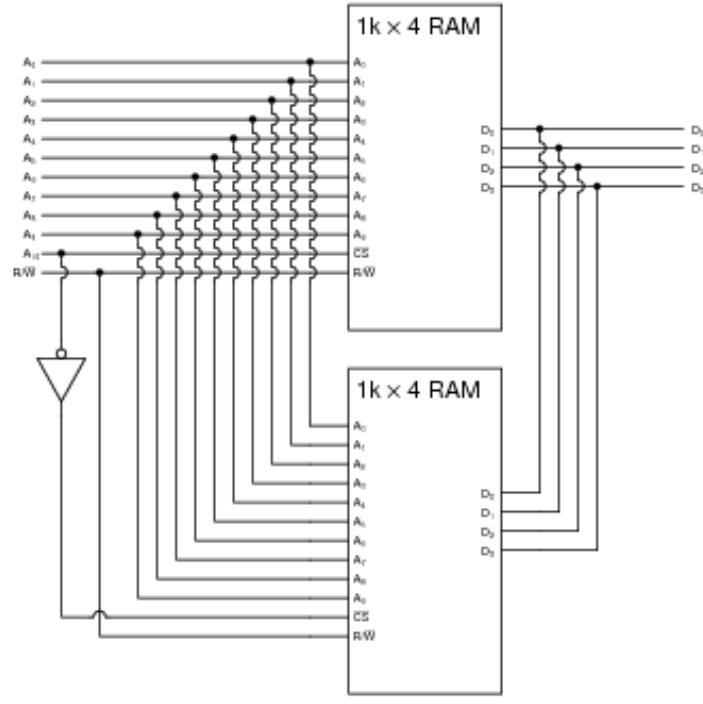
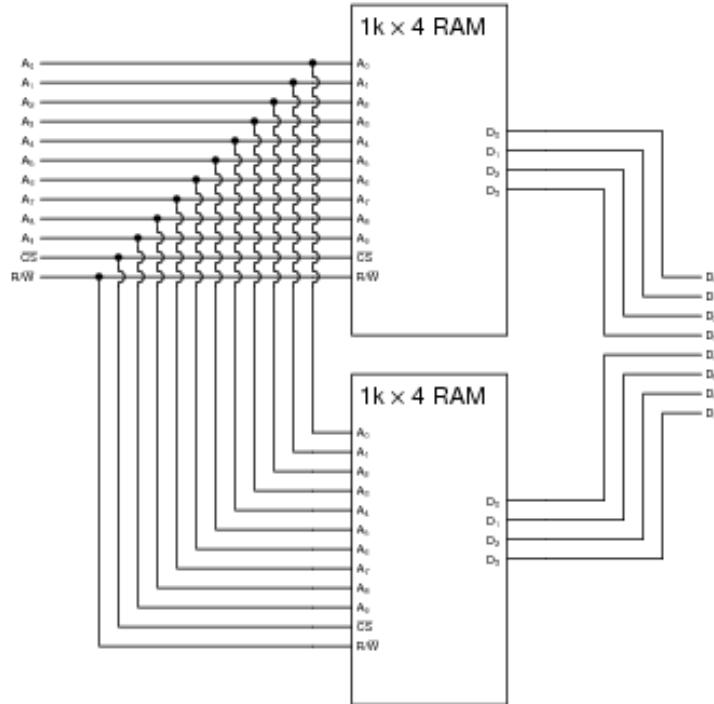
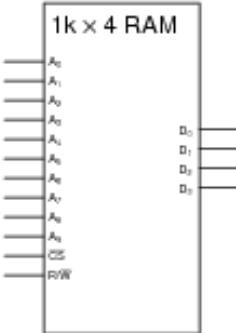
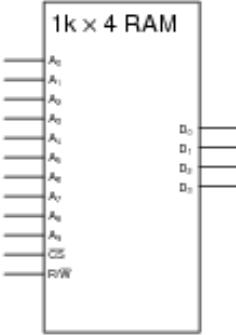
Modelo abstracto de una memoria ($m \times n$)

- Arreglo de **m** celdas consecutivas, cada una capaz de almacenar **n** bits.
- Cada una de las **m** celdas tiene una identificación única (**dirección**), por medio de la cual puede ser accedida para almacenar o recuperar información de la misma.
- La dirección está codificada en binario, utilizando $\log_2(m)$ bits.
- Los **n** bits contenidos en cada celda (**datos**) se leen o escriben simultáneamente.
- Sólo una de las **m** direcciones puede ser accedida por turno, o bien para lectura o para escritura.

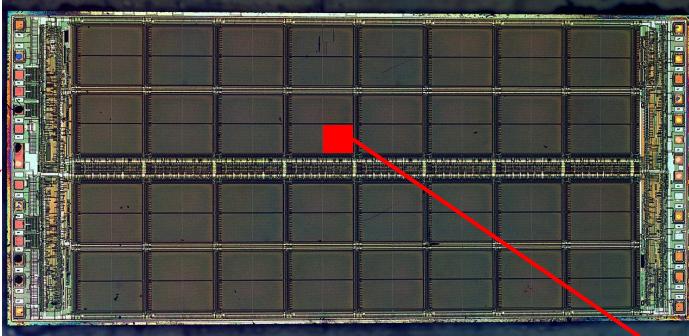
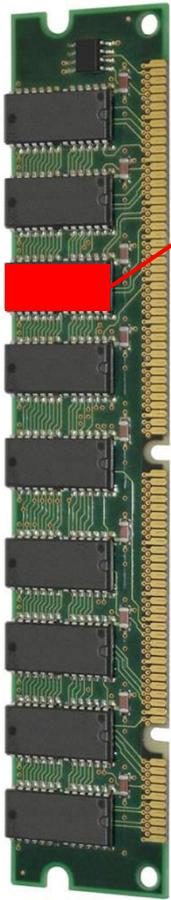
Simplificaciones esquemáticas diversas (diseño gráfico...)



*La clase que viene
vamos a usar ésta*



2 IC de 1Kx4
(ej. 2114 o 7C150)



SIMM 8MB

Micron Technology **MT4C1024** DRAM
8 ic de 1Mx1 organizados como 1Mx8
 $8 \times \{ (128 \times 64) \times 4 \times 8 \times 4 \}$

20/26-Pin SOJ
(DC-1)

D	1	26	Vss
WE	2	25	Q
RAS	3	24	CAS
TF	4	23	NC
NC	5	22	A9*
A0	9	18	A8
A1	10	17	A7
A2	11	16	A6
A3	12	15	A5
Vcc	13	14	A4



Matriz
128 x 64

Tecnología CMOS 1990 (600 nm, 60-80 ns)

Memoria VHDL RAM

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

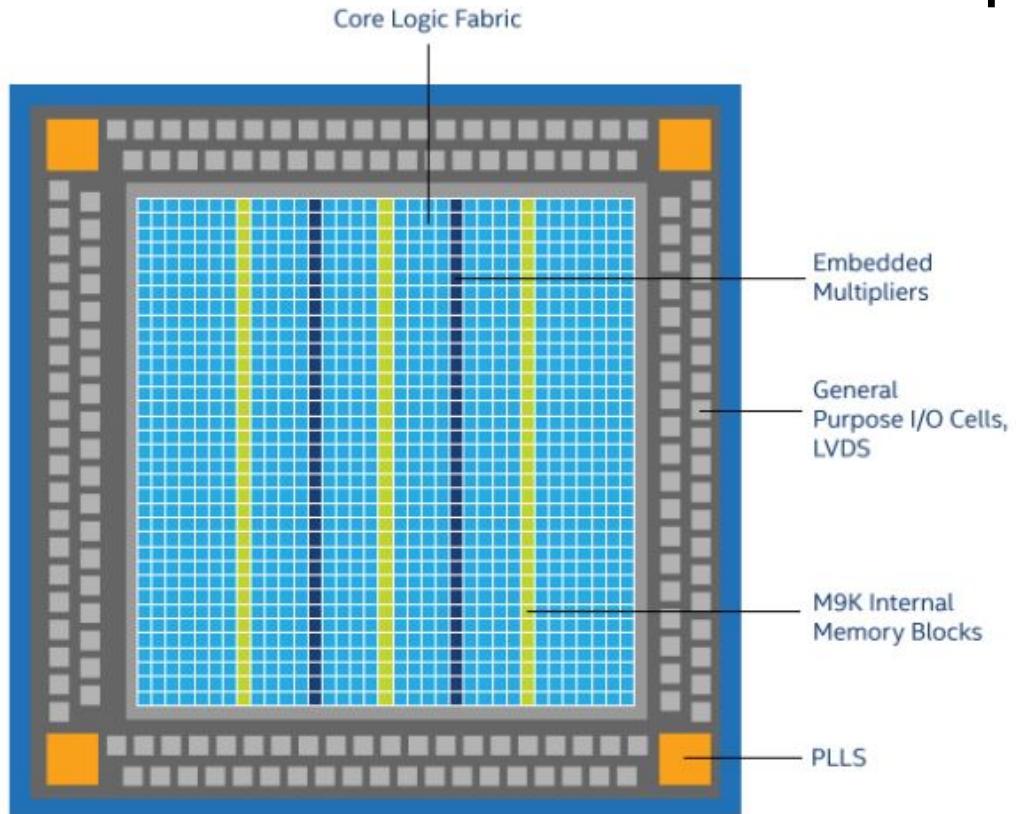
entity ram6116 is
    port(address: in unsigned(7 downto 0);
        data: inout std_logic_vector(7 downto 0);
        WE_b, CS_b, OE_b: in std_ulogic);
end entity ram6116;
```

```
architecture simple_ram of ram6116 is
    type ram_type is array (0 to 2**8-1) of
        std_logic_vector(7 downto 0);
    signal raml: ram_type:= (others => (others => '0'));
begin
    process
    begin
        data <= (others => 'Z'); -- chip is not selected
        if (CS_b = '0') then
            if rising_edge(WE_b) then -- write
                raml(conv_integer(address'delayed)) <= data;
                wait for 0 ns;
            end if;
            if WE_b = '1' and OE_b = '0' then -- read
                data <= raml(conv_integer(address));
            else
                data <= (others => 'Z');
            end if;
        end if;
        wait on WE_b, CS_b, OE_b, address;
    end process; end simple_ram;
```

Memoria VHDL ROM

```
library ieee;
use ieee.std_logic_1164.all;
entity rom16x8 is
    port(address: in integer range 0 to 15;
         data: out std_ulogic_vector(7 downto 0));
end entity;
architecture sevenseg of rom16x8 is
    type rom_array is array (0 to 15) of std_ulogic_vector (7
                           downto 0);
    constant rom: rom_array := ( "11111011", "00010010",
                                 "10011011", "10010011", "01011011", "00111010",
                                 "11111011", "00010010", "10100011", "10011010",
                                 "01111011", "00010010", "10101001", "00110110",
                                 "11011011", "01010010");
begin
    data <= rom(address);
end architecture;
```

Implementación en FPGA





Computer memory types

Volatile

RAM

DRAM (SDRAM) · SRAM

Historical

Williams–Kilburn tube (1946–47)

Delay line memory (1947)

Mellon optical memory (1951) · Selectron tube (1952) · Dekatron · T-RAM (2009) · Z-RAM (2002–2010)

Non-volatile

ROM

Mask ROM · PROM · EPROM · EEPROM · Flash memory

NVRAM

ReRAM

Early stage NVRAM

FeRAM · MRAM · PCM · FeFET memory

Magnetic

Magnetic tape · Hard disk drive

Optical

Optical disc

In development

CBRAM · Racetrack memory · NRAM ·

Millipede memory · FJG RAM

Historical

Paper data storage (1725) · Drum memory (1932) · Magnetic-core memory (1949) · Plated wire memory (1957) · Core rope memory (1960s) ·

Thin-film memory (1962) · Disk pack (1962) · Twistor memory (~1968) · Bubble memory (~1970) · Floppy disk (1971)

RAM-ROM

¿Por qué el término RAM es confuso?

Lectura/escritura - sólo lectura

Volátil - no volátil

Acceso aleatorio - Acceso secuencial

SRAM: RAM estática

DRAM: RAM dinámica

SDRAM: DRAM sincrónica

DDR-SDRAM: SDRAM double data rate

PROM: ROM programable

EPROM: PROM borrible

EEPROM: PROM electricamente borrible

Flash memory: EEPROM por bloques

Tecnologías de memorias (ISDL-12, Floyd ch.14)



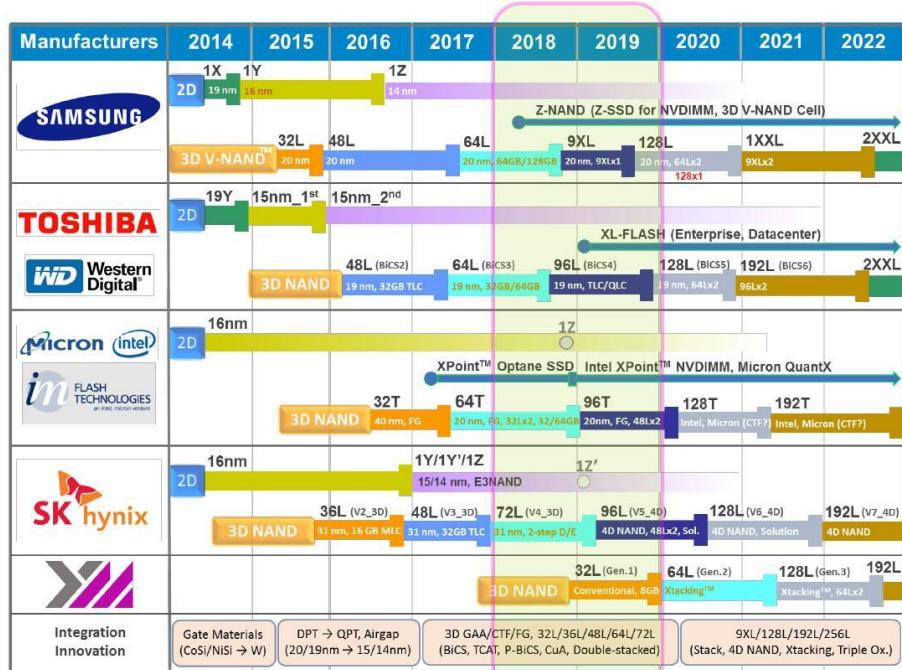
TEMAS PARA MANTENERSE INFORMADOS

Estado del arte de la tecnología de almacenamiento existente (volumen, velocidad, costo)

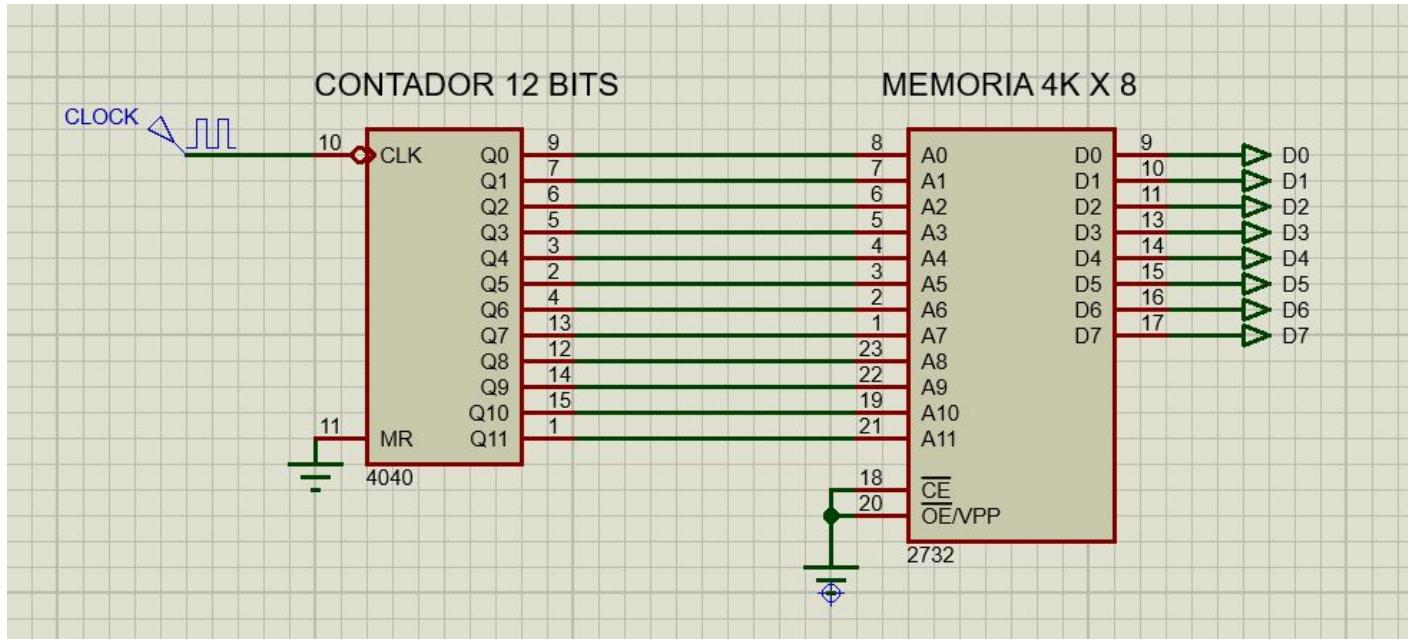
Nuevos tipos de memorias volátiles y no volátiles

Nuevos fabricantes, alianzas y “roadmaps”.

Ejemplo: almacenamiento de archivos personales: diskette (magnético), pendrive (estádo sólido), nube (virtual).



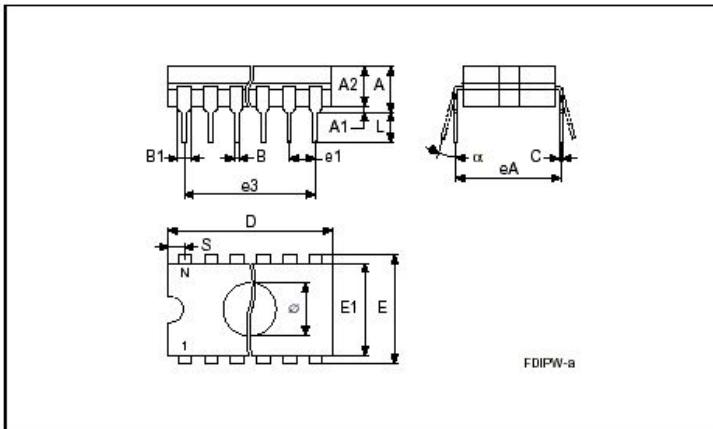
Memorias en Proteus



https://assets.nexperia.com/documents/data-sheet/74HC_HCT4040.pdf

<http://ee.hawaii.edu/~sasaki/EE260/Labs/Datasheets/m2732a.pdf>

Memorias en Proteus



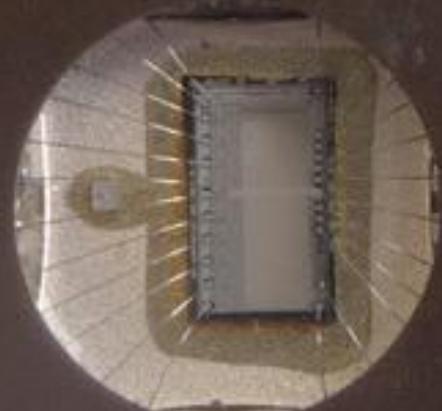
A7	1	VCC
A6	2	A8
A5	3	A9
A4	4	A11
A3	5	GVPP
A2	6	M2732A
A1	7	A10
A0	8	E
Q0	9	Q7
Q1	10	Q6
Q2	11	Q5
VSS	12	Q4
	13	Q3



2732A - 32K (4K x 8)
450ns UV NMOS EPROM

https://www.futurlec.com/Memory/2732_Datasheet.shtml

© 1968 ZTELL 82



Para pensar...

Una memoria ROM ($m \times n$) puede verse como un circuito combinacional con $\log_2(m)$ entradas y n salidas.

Dada la tabla de verdad del circuito combinacional:

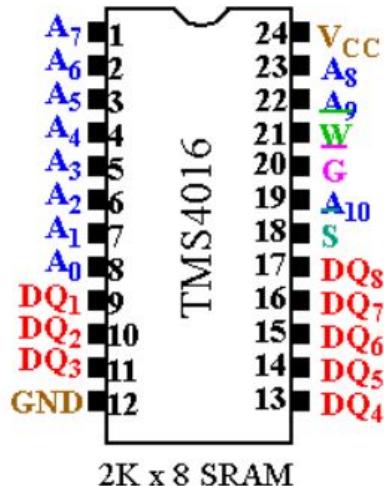
- Las entradas son las direcciones.
- Las salidas son el contenido de la memoria

Ejemplos:

- Una ROM 8x8 puede grabarse tal que se comporte como un Decodificador 3:8. ¿Cuál debería ser el contenido de la memoria? Ver ejemplo Proteus.
- ¿Qué dimensiones debería tener una ROM para que se pueda utilizarse como un sumador de 4 bits con carry? ¿Cuál debería ser el contenido de la memoria?

TI TMS4016 - 2K x 8 SRAM

Datasheet: <http://datasheets.hypertriton.com/TMS4016.pdf>



Pin(s)	Function
A ₀ -A ₁₀	Address
DQ ₀ -DQ ₇	Data In/Data Out
S (CS)	Chip Select
G (OE)	Read Enable
W (WE)	Write Enable

Importante: 2K x 8 es la organización lógica.

Es muy probable que internamente los elementos estén arreglados en una matriz de 128x128

$$\sqrt{2K \cdot 8} = 128$$