

PARTE 1:

Introducción a los Sistemas Operativos y a su relación con la Arquitectura de Computadores

Sección 2. Electrónica del ordenador.

Puertas lógicas

Son los componentes básicos de los circuitos digitales. Ej.: NOT

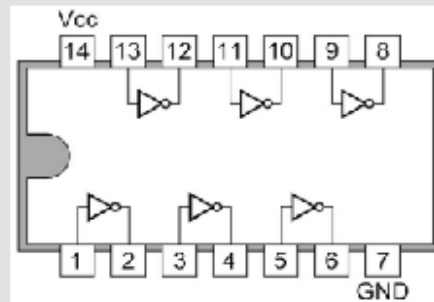
Tabla de verdad:

Entrada	Salida
0	1
1	0

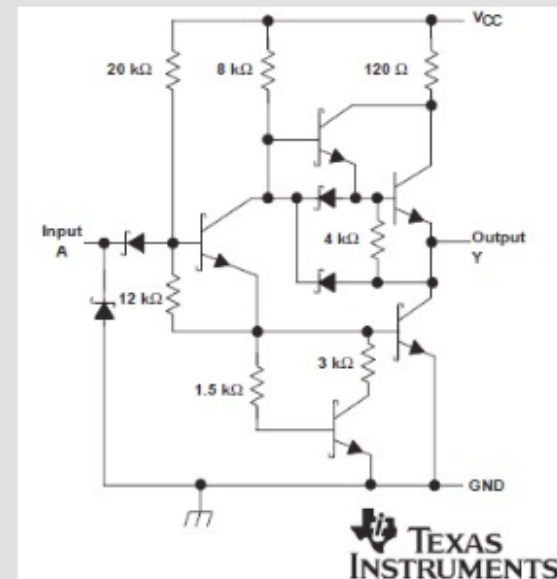
Símbolo:



Chip: 74LS04







Así está hecha:




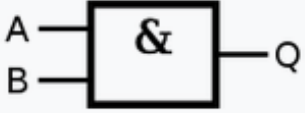

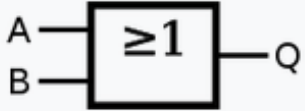
switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	SN54S04 SN74S04			UNIT
				MIN	TYP	MAX	
t_{PLH}	A	Y	$R_L = 280\ \Omega$, $C_L = 15\text{ pF}$		3	4.5	ns
t_{PHL}					3	5	
t_{PLH}	A	Y	$R_L = 280\ \Omega$, $C_L = 50\text{ pF}$		4.5		ns
t_{PHL}					5		


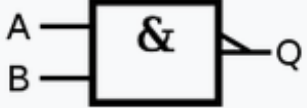

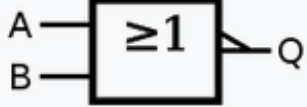
Puertas lógicas: Buffer, No

Type	Distinctive shape (IEEE Std 91/91a-1991)	Rectangular shape (IEEE Std 91/91a-1991) (IEC 60617-12:1997)	Boolean algebra between A & B	Truth table								
Buffer			A	<table border="1"> <thead> <tr> <th>INPUT</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Q</td> </tr> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	INPUT	OUTPUT	A	Q	0	0	1	1
INPUT	OUTPUT											
A	Q											
0	0											
1	1											
NOT (inverter)			\bar{A} or $\neg A$	<table border="1"> <thead> <tr> <th>INPUT</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Q</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	INPUT	OUTPUT	A	Q	0	1	1	0
INPUT	OUTPUT											
A	Q											
0	1											
1	0											


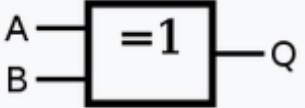

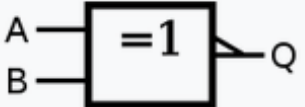
Puertas lógicas: Y, O

<p>AND</p>			<p>$A \cdot B$ or $A \wedge B$</p>	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	Q	0	0	0	0	1	0	1	0	0	1	1	1
INPUT		OUTPUT																				
A	B	Q																				
0	0	0																				
0	1	0																				
1	0	0																				
1	1	1																				
<p>OR</p>			<p>$A + B$ or $A \vee B$</p>	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	1
INPUT		OUTPUT																				
A	B	Q																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	1																				

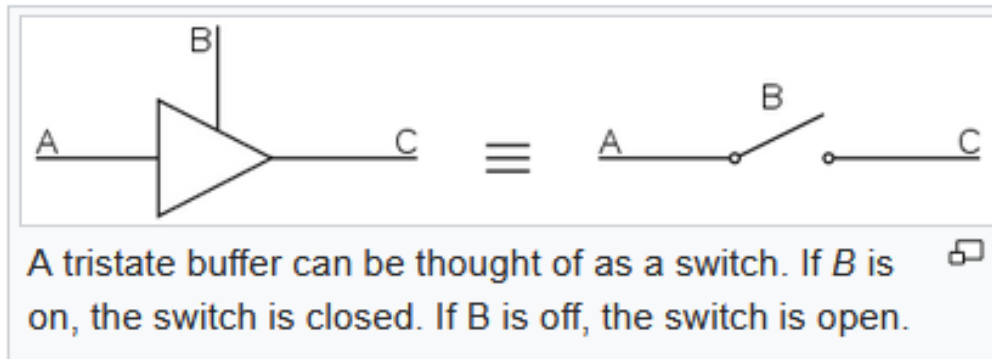
Puertas lógicas: No-y, No-o

<p>NAND</p>			$\overline{A \cdot B}$ or $A \uparrow B$	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	Q	0	0	1	0	1	1	1	0	1	1	1	0
INPUT		OUTPUT																				
A	B	Q																				
0	0	1																				
0	1	1																				
1	0	1																				
1	1	0																				
<p>NOR</p>			$\overline{A + B}$ or $A \downarrow B$	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	0
INPUT		OUTPUT																				
A	B	Q																				
0	0	1																				
0	1	0																				
1	0	0																				
1	1	0																				

Puertas lógicas: O-exclusivo, No-oexclusivo

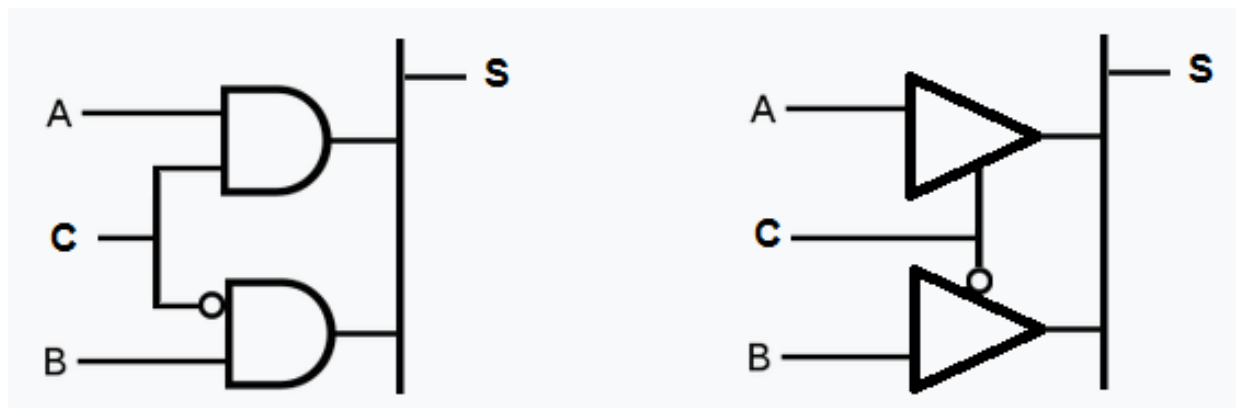
<p>XOR</p>			$A \oplus B$ or $A \vee B$	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	0
INPUT		OUTPUT																				
A	B	Q																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	0																				
<p>XNOR</p>			$\overline{A \oplus B}$ or $A \odot B$	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	1
INPUT		OUTPUT																				
A	B	Q																				
0	0	1																				
0	1	0																				
1	0	0																				
1	1	1																				

Puertas lógicas: buffer triestado



INPUT		OUTPUT
A	B	C
0	0	Z (high impedance)
1	0	Z (high impedance)
0	1	0
1	1	1

¿Son equivalentes estos dos circuitos?



Sistemas combinacionales

Sus salidas solo dependen de las entradas en un momento dado (no depende de entradas anteriores).

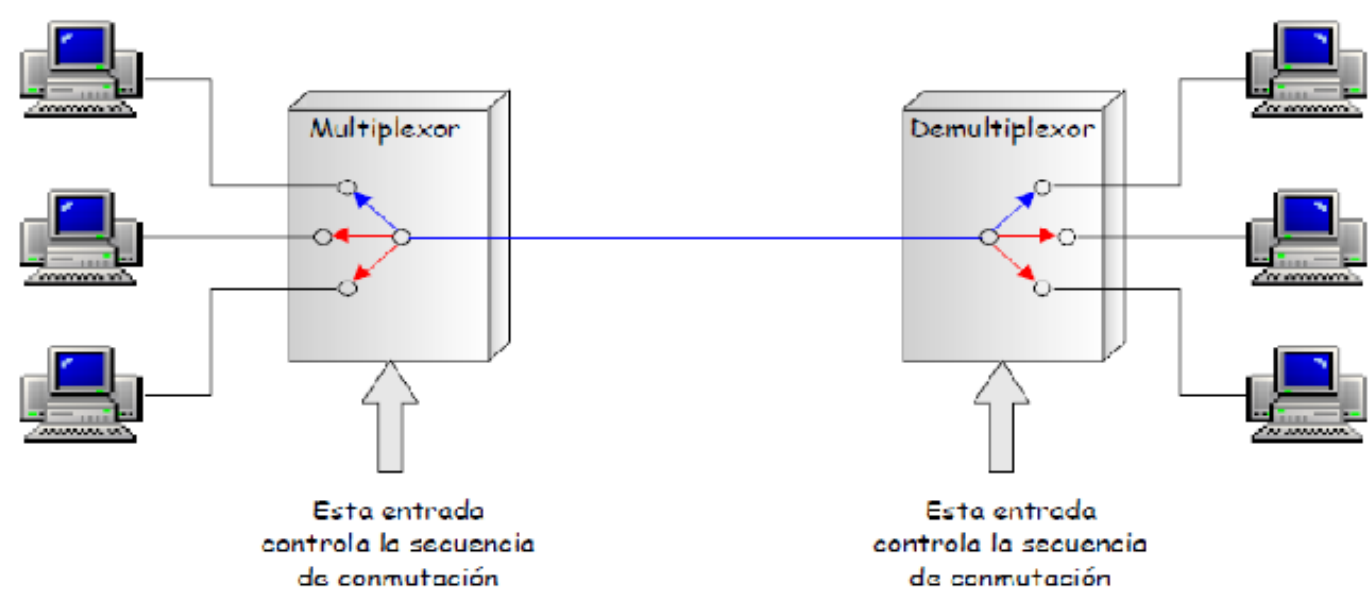
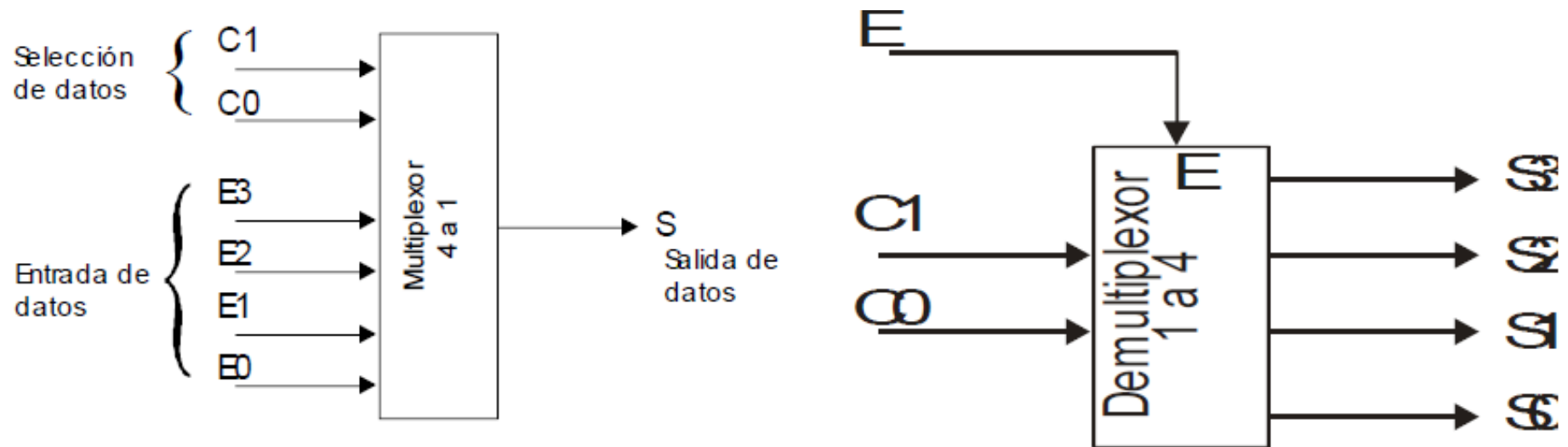
Ejemplos de sistemas combinacionales:

- Lógicos: Multiplexor, demultiplexor, codificador, decodificador, desplazamiento lógico, etc.
- Aritméticos: Desplazamiento aritmético, comparador*, sumador, etc.
- Lógico-aritméticos: Unidad lógico aritmética (ULA).

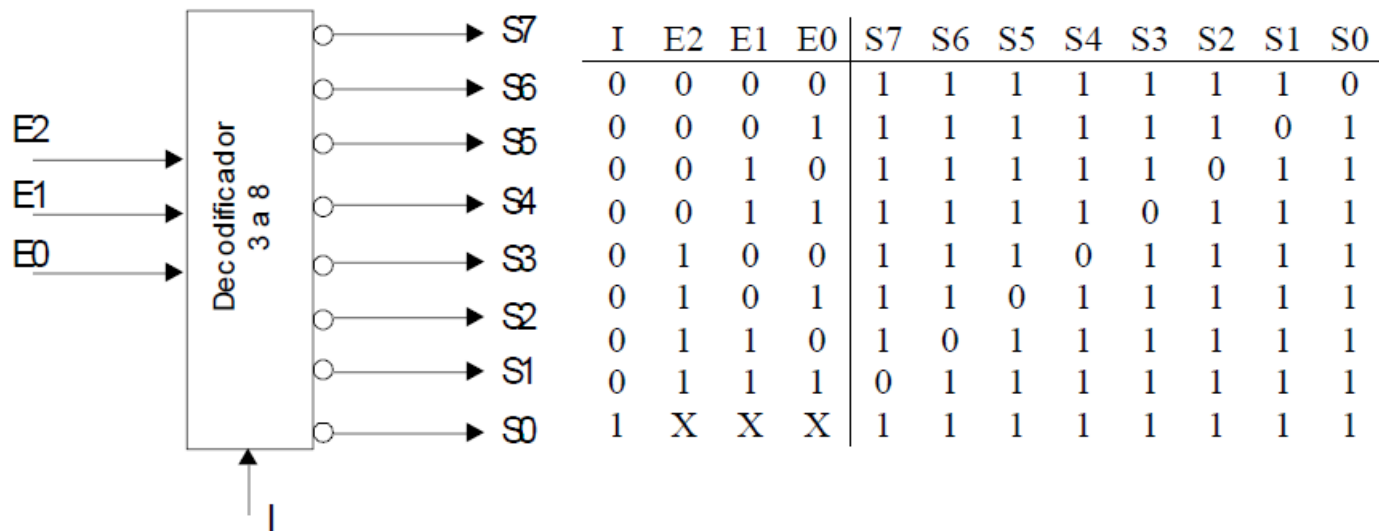
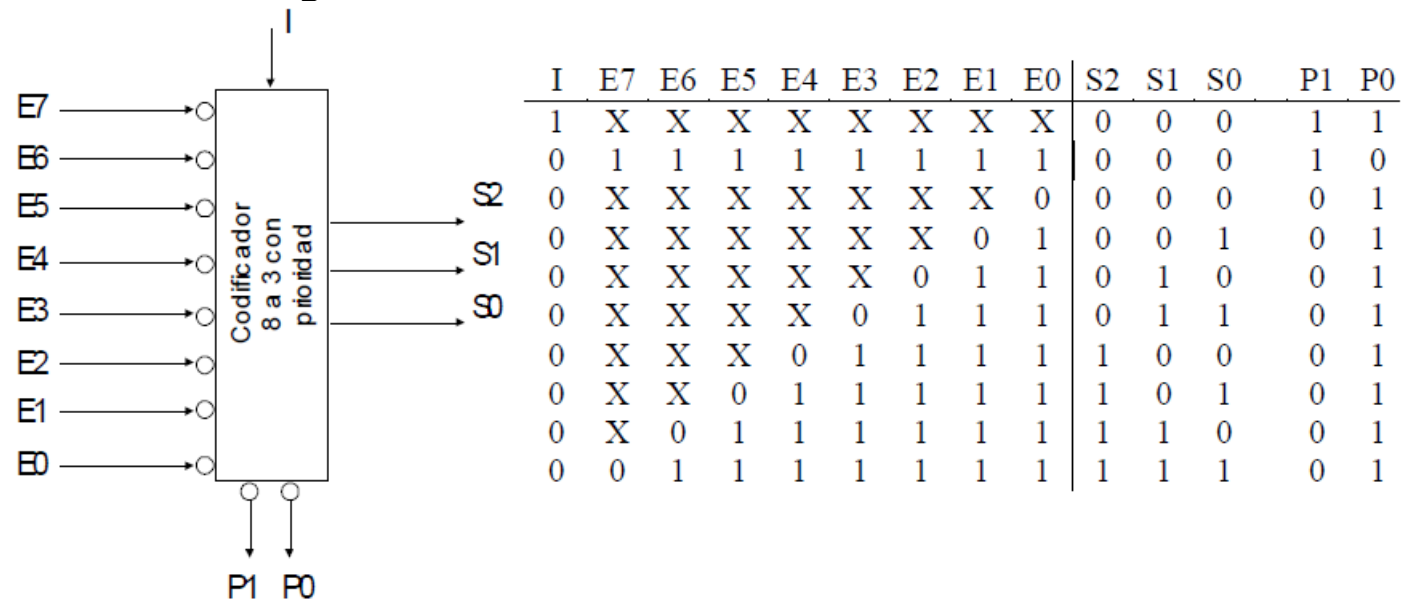
Las entradas pueden ser de 1 o más bits (por ejemplo, un comparador de 8 bits).

* Si el comparador solo indica igualdad, se puede clasificar como lógico.

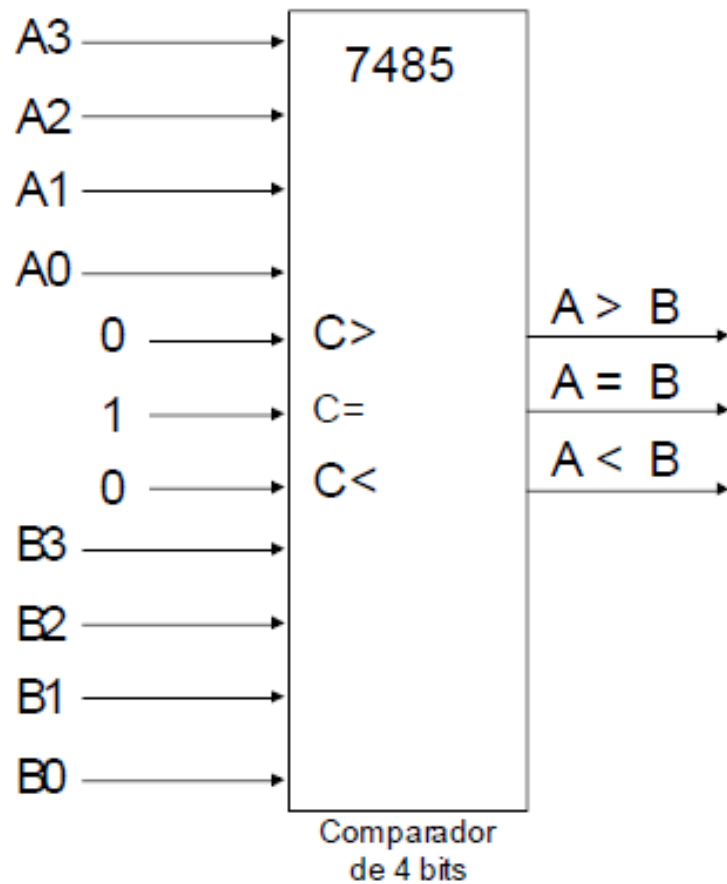
Multiplexor y demultiplexor



Codificador y decodificador

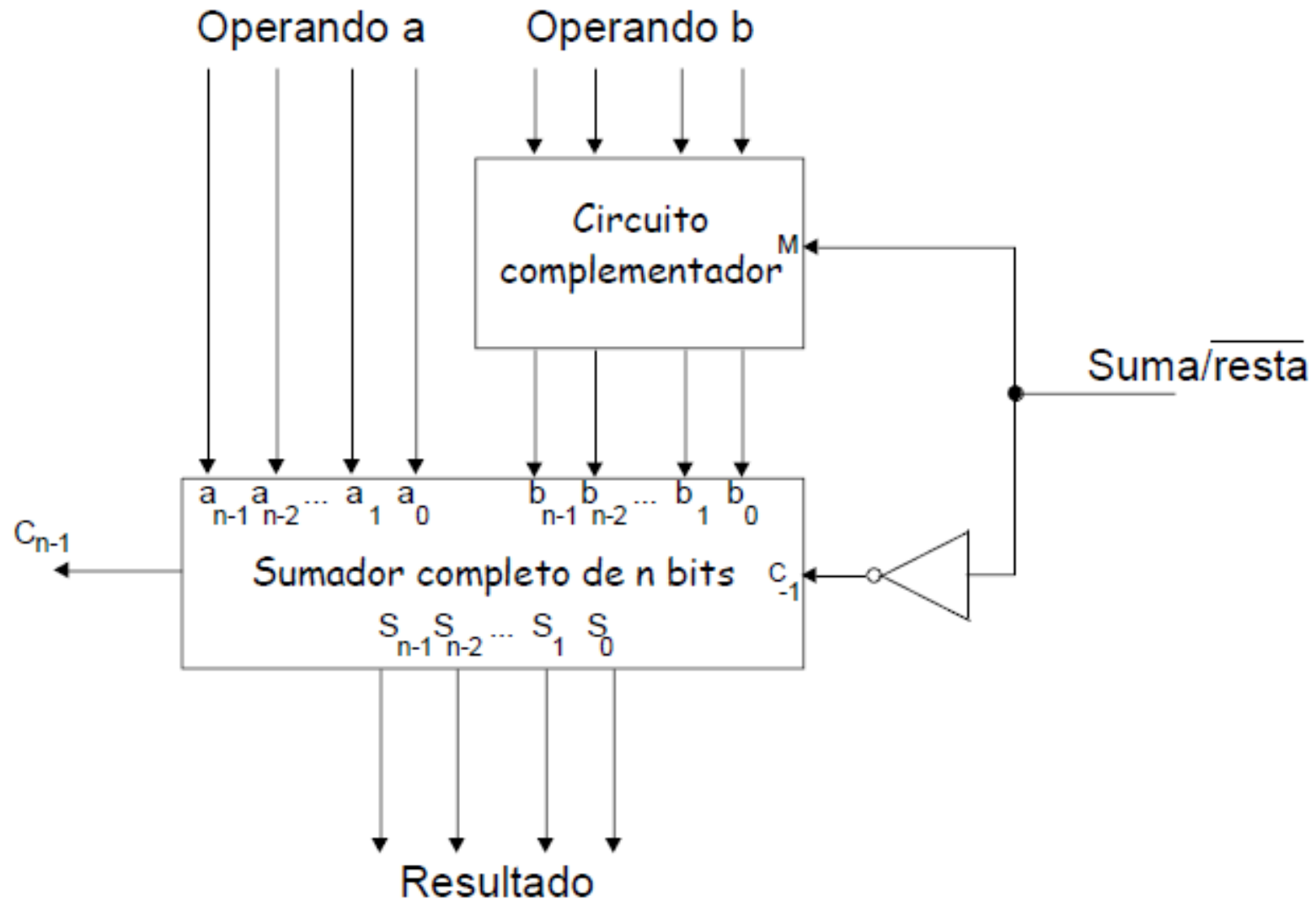


Comparador



Ay B	C>	C=	C<	A>B	A=B	A<B
A > B	X	X	X	1	0	0
A < B	X	X	X	0	0	1
A = B	0	0	1	0	0	1
A = B	0	1	0	0	1	0
A = B	1	0	0	1	0	0

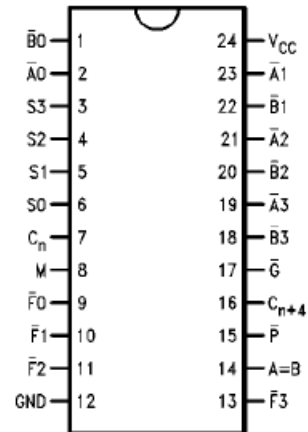
Sumador/restador



ULA (o ALU Arithmetic Logic Unit)

Ej.: 74LS181 (Fairchild Datasheet)

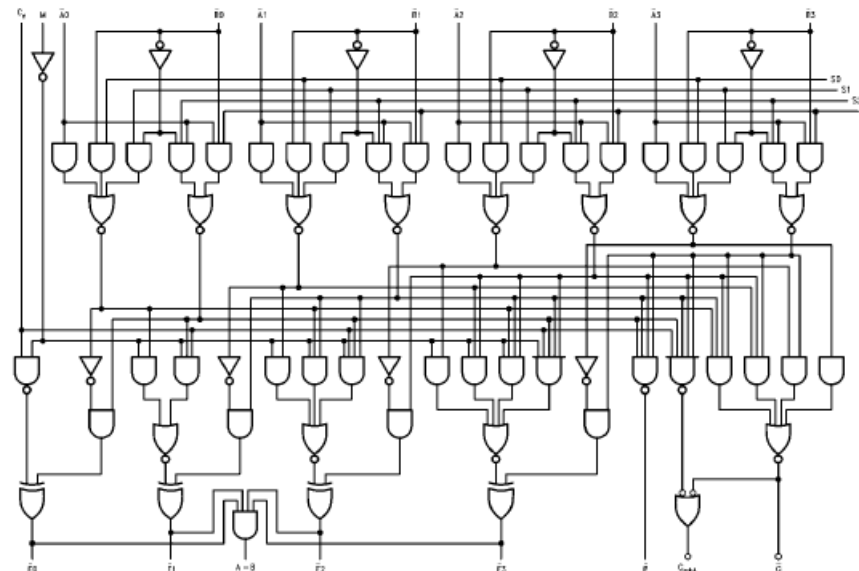
Connection Diagram



Pin Descriptions

Pin Names	Description
$\bar{A}0-\bar{A}3$	Operand Inputs (Active LOW)
$\bar{B}0-\bar{B}3$	Operand Inputs (Active LOW)
S0-S3	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\bar{F}0-\bar{F}3$	Function Outputs (Active LOW)
A = B	Comparator Output
\bar{G}	Carry Generate Output (Active LOW)
\bar{P}	Carry Propagate Output (Active LOW)
C_{n+4}	Carry Output

Logic Diagram



Ej.: 74LS181 (continuación)

Function Table

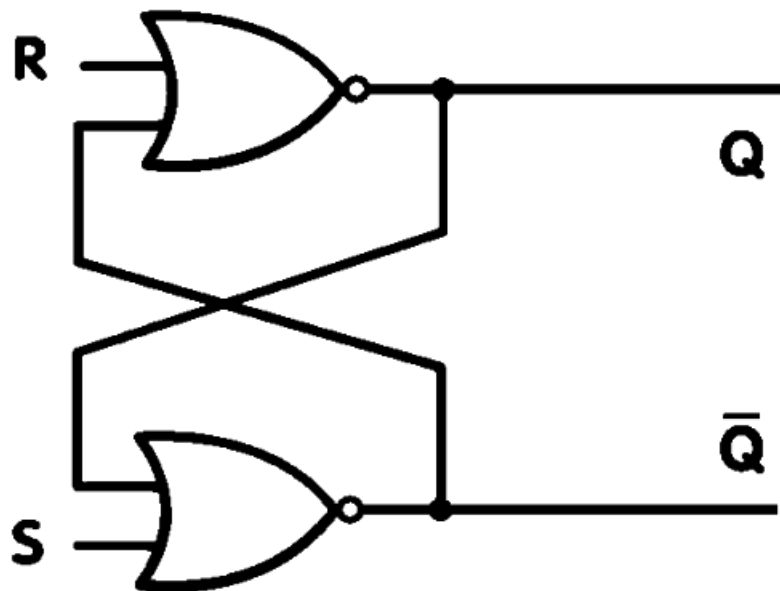
Mode Select Inputs				Active LOW Operands & F_n Outputs		Active HIGH Operands & F_n Outputs	
S3	S2	S1	S0	Logic	Arithmetic (Note 2)	Logic	Arithmetic (Note 2)
				(M = H)	(M = L) ($C_n = L$)	(M = H)	(M = L) ($C_n = H$)
L	L	L	L	\bar{A}	A minus 1	\bar{A}	A
L	L	L	H	\overline{AB}	AB minus 1	$\bar{A} + \bar{B}$	A + B
L	L	H	L	$\bar{A} + \bar{B}$	$A\bar{B}$ minus 1	$\bar{A} B$	A + \bar{B}
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\bar{A} + \bar{B}$	A plus ($A + \bar{B}$)	\overline{AB}	A plus \overline{AB}
L	H	L	H	\bar{B}	AB plus ($A + \bar{B}$)	\bar{B}	(A + B) plus \overline{AB}
L	H	H	L	$\bar{A} \oplus \bar{B}$	A minus B minus 1	$A \oplus B$	A minus B minus 1
L	H	H	H	$A + \bar{B}$	$A + \bar{B}$	\overline{AB}	AB minus 1
H	L	L	L	$\bar{A} B$	A plus (A + B)	$\bar{A} + B$	A plus AB
H	L	L	H	$A \oplus B$	A plus B	$\bar{A} \oplus \bar{B}$	A plus B
H	L	H	L	B	\overline{AB} plus (A + B)	B	(A + \bar{B}) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logic 0	A plus A (Note 1)	Logic 1	A plus A (Note 1)
H	H	L	H	\overline{AB}	AB plus A	$A + \bar{B}$	(A + B) plus A
H	H	H	L	AB	\overline{AB} minus A	A + B	(A + \bar{B}) plus A
H	H	H	H	A	A	A	A minus 1

Note 1: Each bit is shifted to the next most significant position.

Note 2: Arithmetic operations expressed in 2s complement notation.

Sistemas secuenciales

Sus salidas dependen tanto de las entradas como del estado interno producido por entradas anteriores (tiene “memoria”). Ej.: Cerrojo SR (SR Latch)



$S \Rightarrow Set, R \Rightarrow Reset$

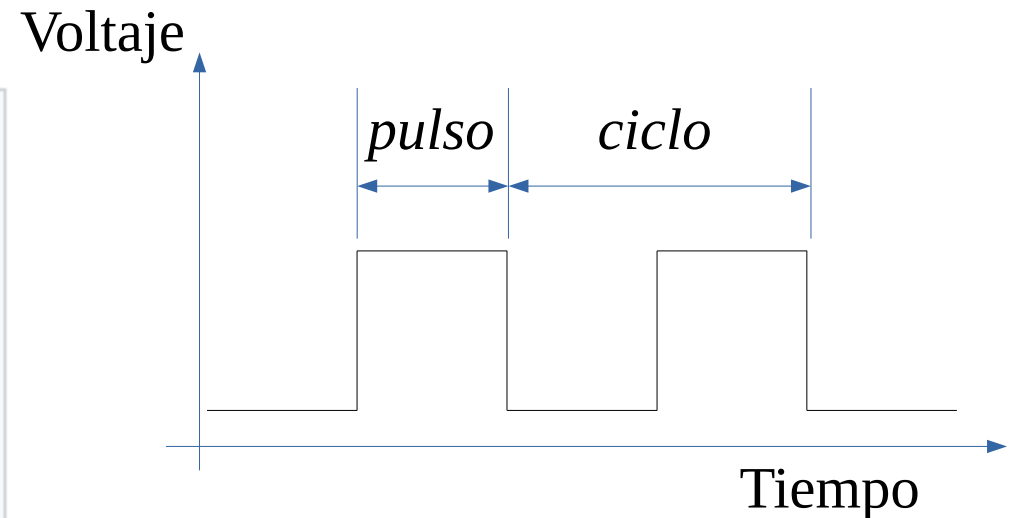
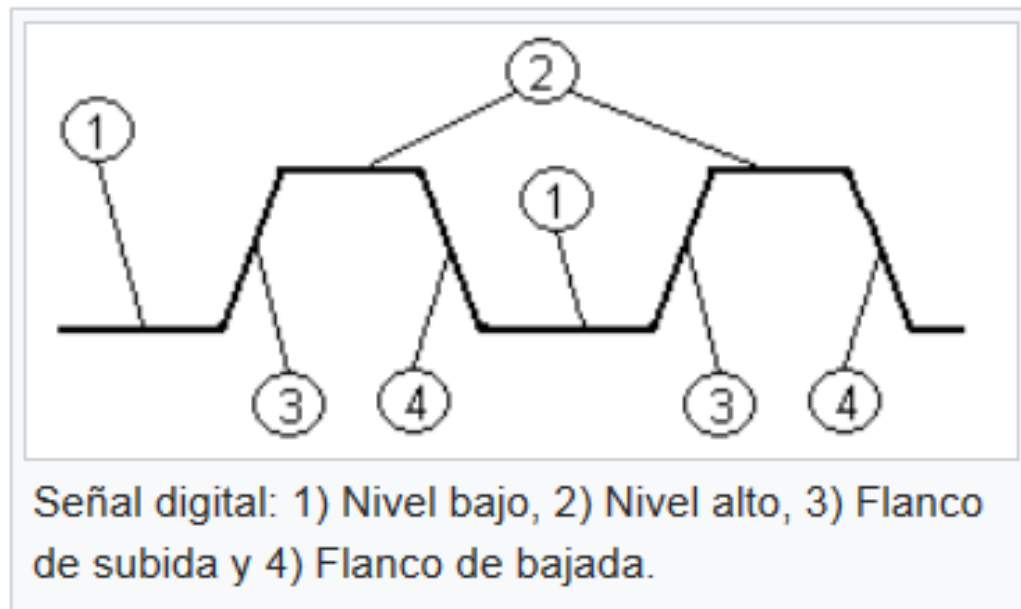
Characteristic table			
S	R	Q_{next}	Action
0	0	Q	Hold state
0	1	0	Reset
1	0	1	Set
1	1	X	Not allowed

recuerda:

INPUT		OUTPUT
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

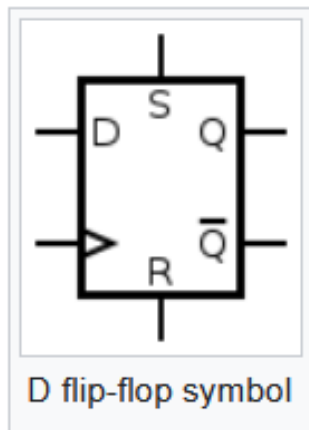
Señal de reloj

Los sistemas secuenciales pueden llegar a ser muy complejos. Se facilita su diseño si se utiliza una señal de reloj que indica los momentos en que los elementos de memoria guardan el estado (sistemas síncronos)



Biastable (*flip flop*) D

Ejemplo elemento síncrono: biastable D

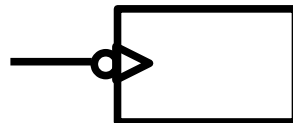


Inputs				
S	R	Clock	D	Q _{next}
0	0	Rising edge	0	0
0	0	Rising edge	1	1
0	0	Non-rising	X	Q

Inputs				Outputs	
S	R	D	>	Q	Q̄
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	X	1	1

$D \Rightarrow Data$, $> \Rightarrow Clock$

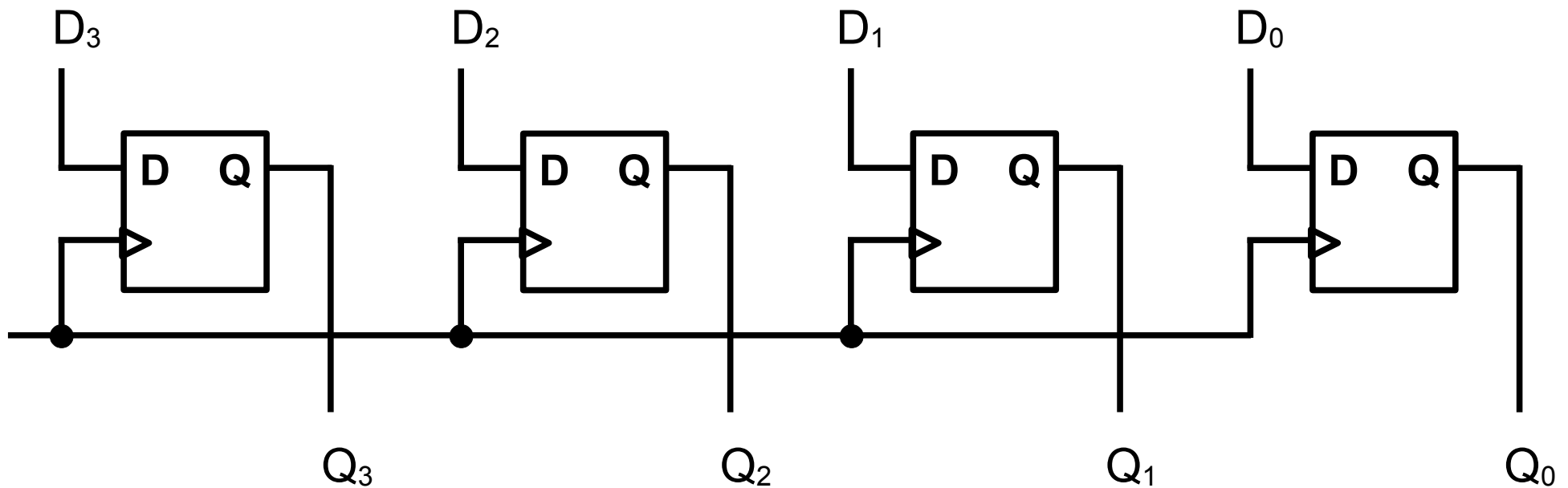
Nota: Para indicar activo por flanco de bajada (*falling edge/negative edge*) se usa también $>$ pero negado:



Registro

Un biestable almacena un solo bit, pero se pueden combinar varios para formar palabras mayores. Este conjunto recibe el nombre de registro.

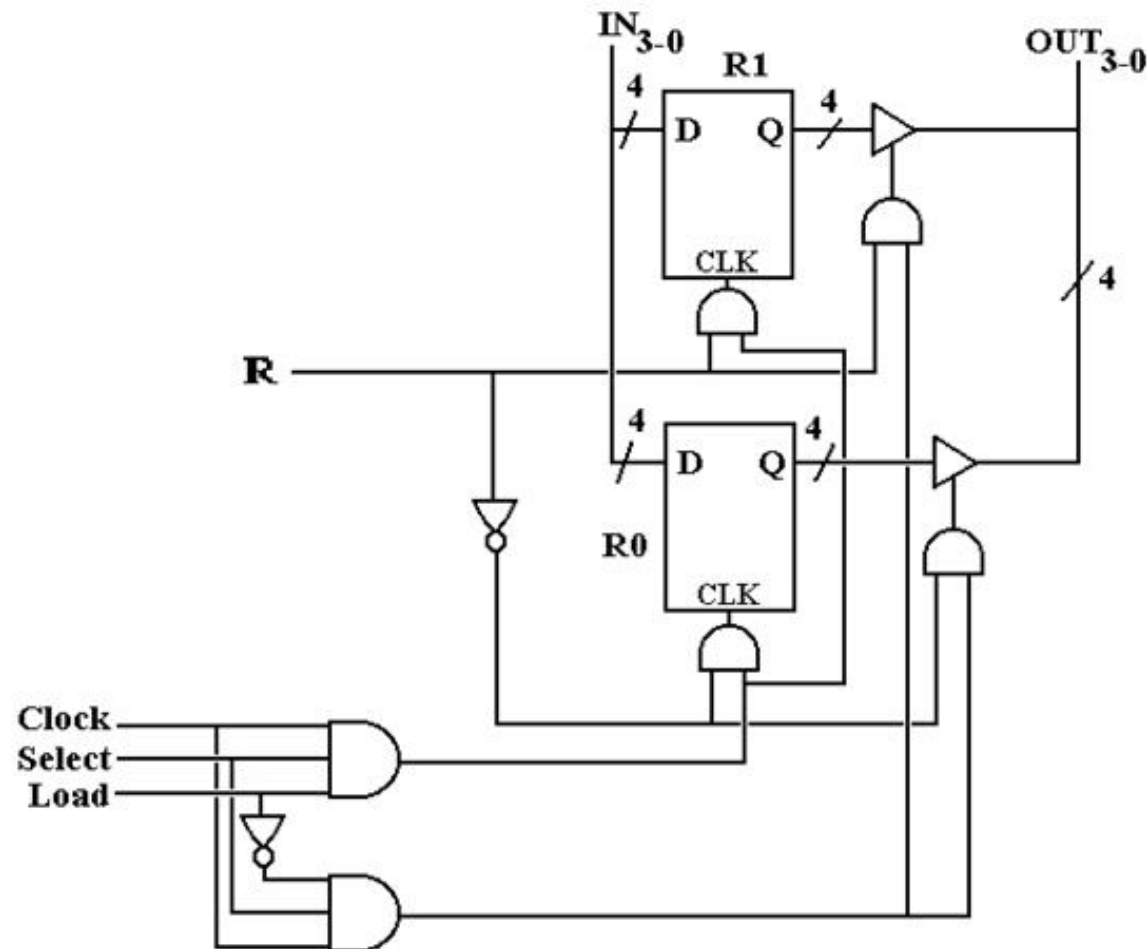
La CPU contiene registros para varios usos.



Banco de registros (1/2)

Varios registros se pueden agrupar formando un banco.

Ej.: Banco dos registros de 4 bits (http://www.edwardbosworth.com/My5155Textbook_HTM/MyText5155_Ch06_V06.htm)



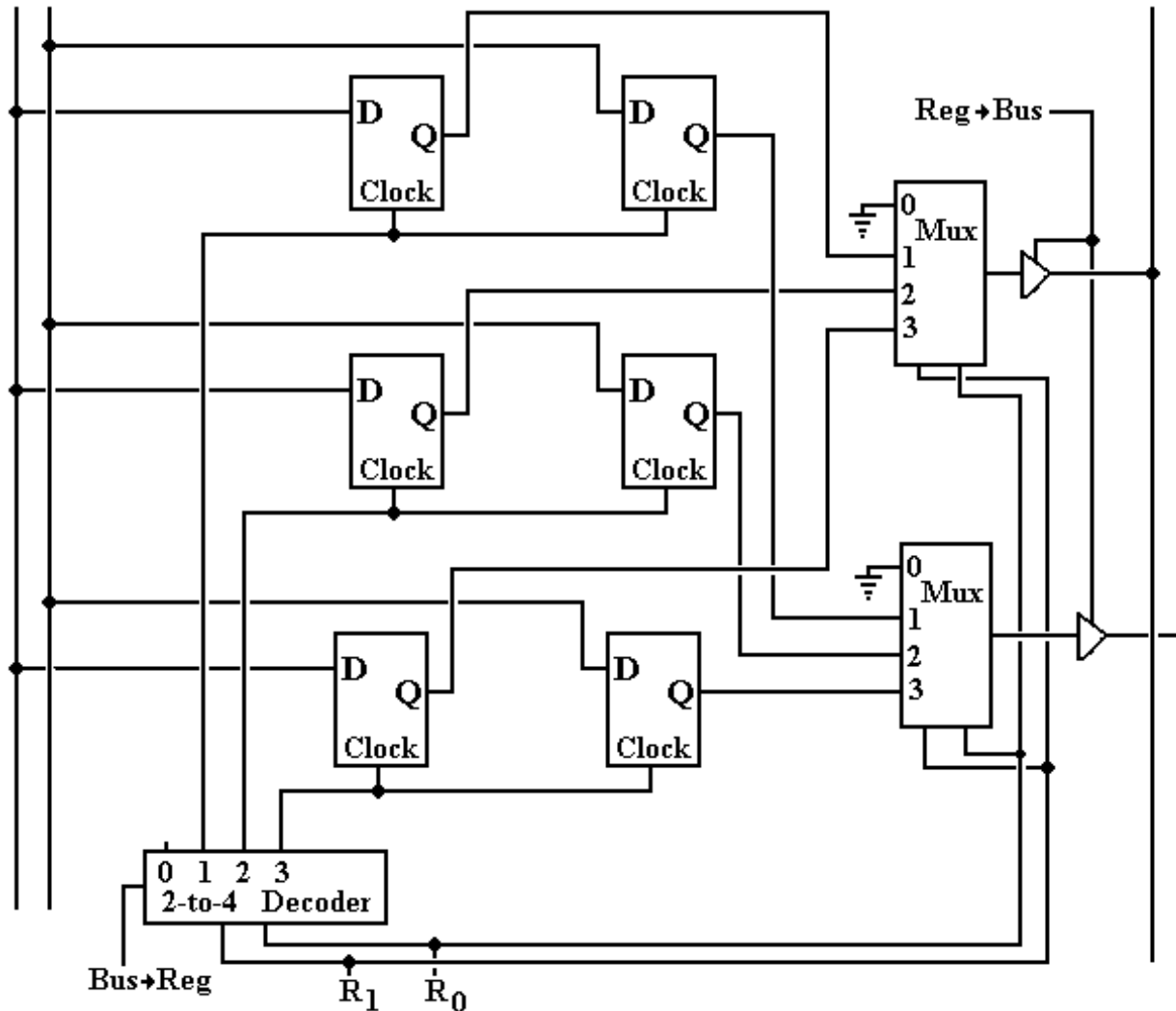
R => selección
registro (R0 o R1).

Select => “CS”/*chip
select*.

Load => carga
(grabación) registro.

Banco de registros (2/2)

Ej. “4” registros de 2 bits (el reg. 0 está cableado a 0).



$Bus \rightarrow Reg \Rightarrow$
grabación

$Reg \rightarrow Bus \Rightarrow$
lectura

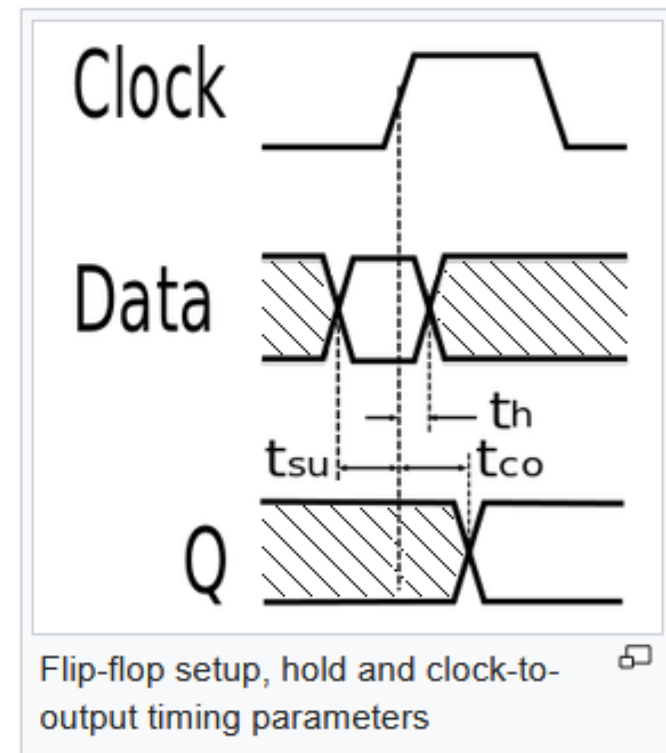
Diagrama de tiempos

Los sistemas no cambian de estado instantáneamente.

Para almacenar un dato, dicho dato debe estar disponible un tiempo antes de que ocurra el flanco de reloj que activa la grabación y mantenerse durante un tiempo después.

Además hay que esperar otro tiempo desde el evento del flanco para que la salida se estabilice.

Ej.: Carga datos en biestables D



Bus

De forma simplificada, un bus es un conjunto de líneas eléctricas que interconectan elementos.

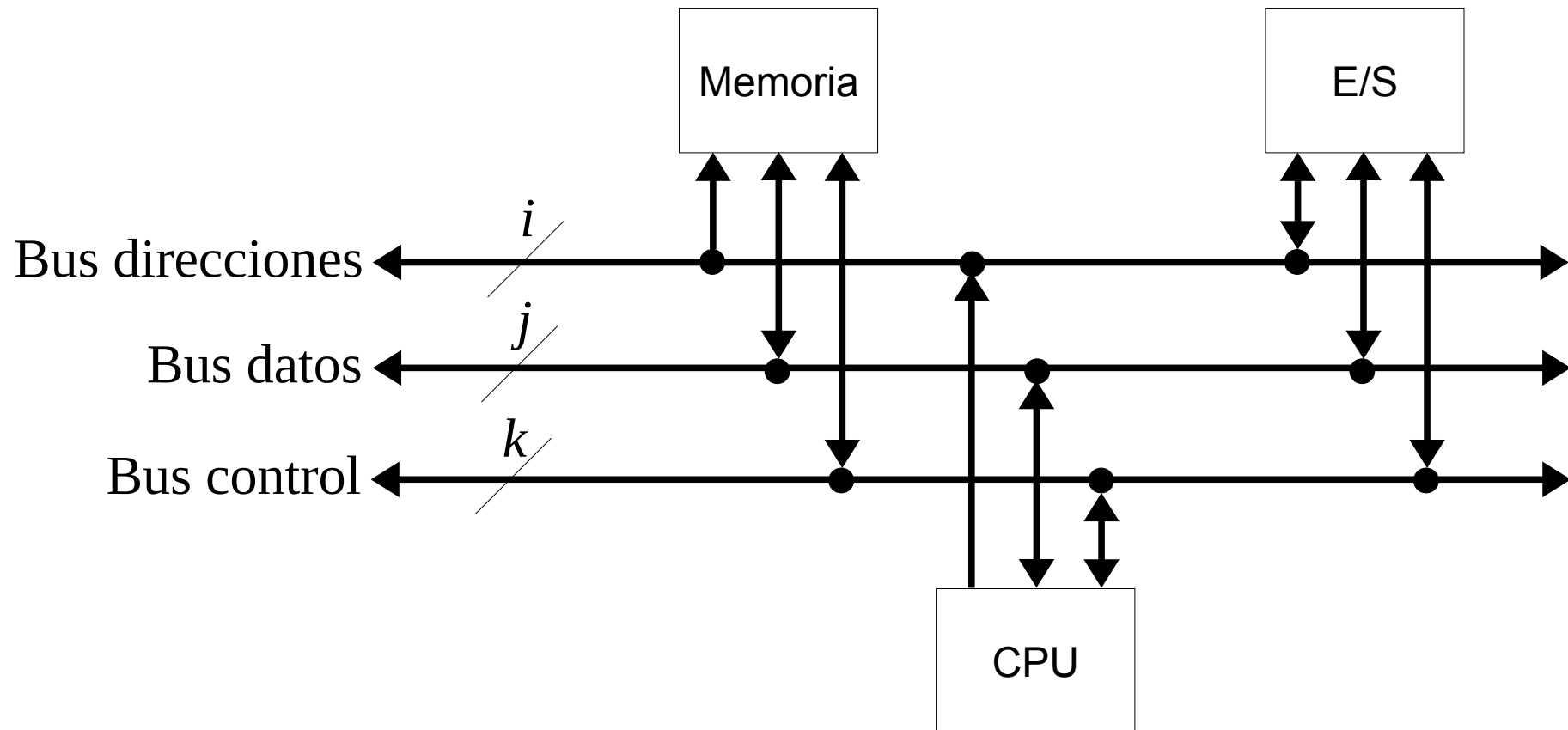
La CPU se conecta al resto de componentes mediante un bus (bus del sistema), que a su vez está compuesto por el bus de direcciones, de datos y de control.

Varios dispositivos se pueden conectar a una misma línea para salida (¡no simultánea!), y una línea de un mismo dispositivo puede servir tanto de salida como de entrada, gracias a las puertas triestado.

La CPU tiene también un bus interno para conectar los módulos que la constituyen.

Bus del sistema (arquitectura von Neumann)

Ej. simplificado de bus (con DMA/Direct Memory Access)



Arquitectura von Neumann vs Harvard

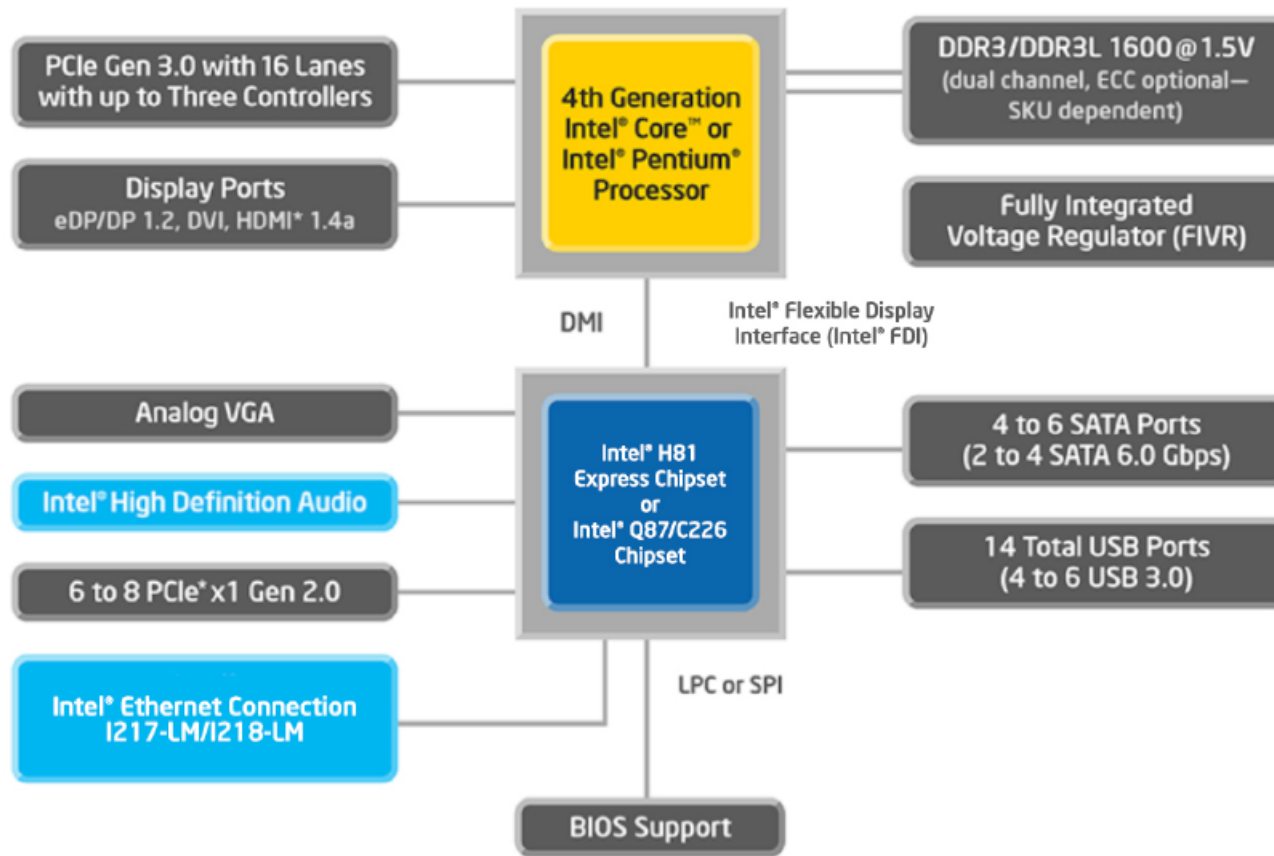
En la arquitectura de von Neumann, en la memoria principal se encuentran tanto datos como programas.

En la arquitectura de Harvard, hay dos memorias con buses independientes: una para el programa y la otra para datos. Pueden tener distinto tamaño de palabra y otras diferencias.

La arquitectura de Harvard permite accesos simultáneos a programa y datos (más rápido) pero a costa de más complejidad.

La mayoría de los ordenadores actuales son arquitectura von Neumann.

¿Bus del sistema o buses?



Typical system based on the Intel® Core™ i7 processor

Fuente: "Introduction to Intel Architecture". Intel.

BIOS	Basic Input/Output System; a boot ROM
DDR3	Double Data-Rate v3; a popular DRAM interface standard
DMI	Direct Memory Interface; a video graphics standard
FIVR	Fully Integrated Voltage Regulator
LPC	Low Pin Count; a simple interface to slower I/O devices
PCH	Platform Controller Hub; a companion chip
PCI	Peripheral Component Interconnect; a popular expansion bus
PCIe*	PCI Express*; an upgraded PCI standard
SATA	Serial ATA; a popular disk-interface standard
SPI	Serial Peripheral Interface; simple interface to slower devices

Distintos buses para distintas necesidades de velocidad y capacidad.

Memoria principal de un ordenador

Se denomina también RAM (*random-access memory*).

Es volátil: se pierde cuando se apaga.

Internamente está hecha con condensadores (Dinamic RAM o DRAM), ya que permiten una densidad de almacenamiento mayor, pero es más lenta (necesita refrescarse) que una memoria hecha de biestables (Static RAM o SRAM), que sí se usa para las memorias caché.

Nota: cada variable de los lenguajes de programación se “traduce” a una dirección de memoria a partir de la cual se almacenan los datos que se le asignan.

Mapa de memoria de un ordenador (1/2)

Los ordenadores precisan una “pequeña” cantidad de memoria no volátil que contiene el programa de inicio que chequea el equipo, carga el S.O., etc. Se le llama tradicionalmente BIOS aunque ahora es UEFI (*Unified Extensible Firmware Interface*), mucho más potente.

Cuando la CPU arranca, empieza a ejecutar instrucciones a partir de una dirección de memoria determinada por el tipo de CPU. Esa dirección tiene que estar en memoria no volátil.

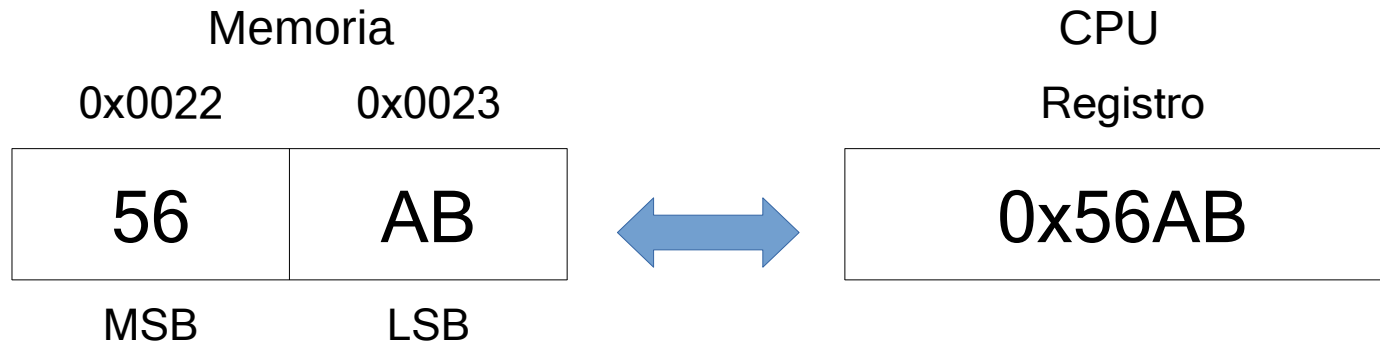
No se tiene que cubrir todo el direccionamiento posible del bus de direcciones con memoria física.

Mapa de memoria de un ordenador (2/2)

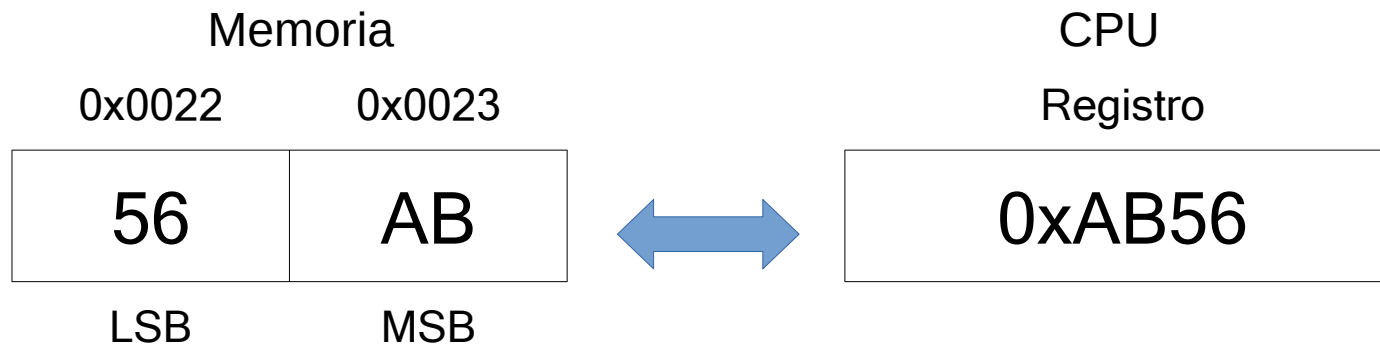
El bus de control puede tener una o más líneas que indican si se trata de acceso a memoria o acceso a E/S. Pero a veces se carece de esa línea o interesa no usarla total o parcialmente y se reserva una parte del direccionamiento que corresponde a la memoria a E/S, lo que se denomina E/S mapeada en memoria.

Formato memoria: ¿*big, little o bi endian*?

Big-Endian: (ej. Motorola 68000)



Little-Endian: (ej. Intel x86-64)



Bi-Endian: ej. PowerPC; S.O. permite especificar tipo *endian* por proceso.

Ejemplo de examen:

Sea el siguiente volcado hexadecimal de bytes una zona de memoria de un programa en ejecución que contiene como datos las primeras potencias de 8.

Nºbyte	Contenido
n	00
n+1	01
n+2	00
n+3	08
n+4	00
n+5	40
n+6	02
n+7	00

Por el volcado podemos decir que la arquitectura del sistema es:

- a) big-endian
- b) little-endian
- c) Nada, porque habría que saber el tamaño de la palabra de memoria.
- d) Nada, porque habría que saber si n es par o impar.

Chip de RAM

Contiene n celdillas de un determinado tamaño de palabra (por ejemplo, 8 bits).

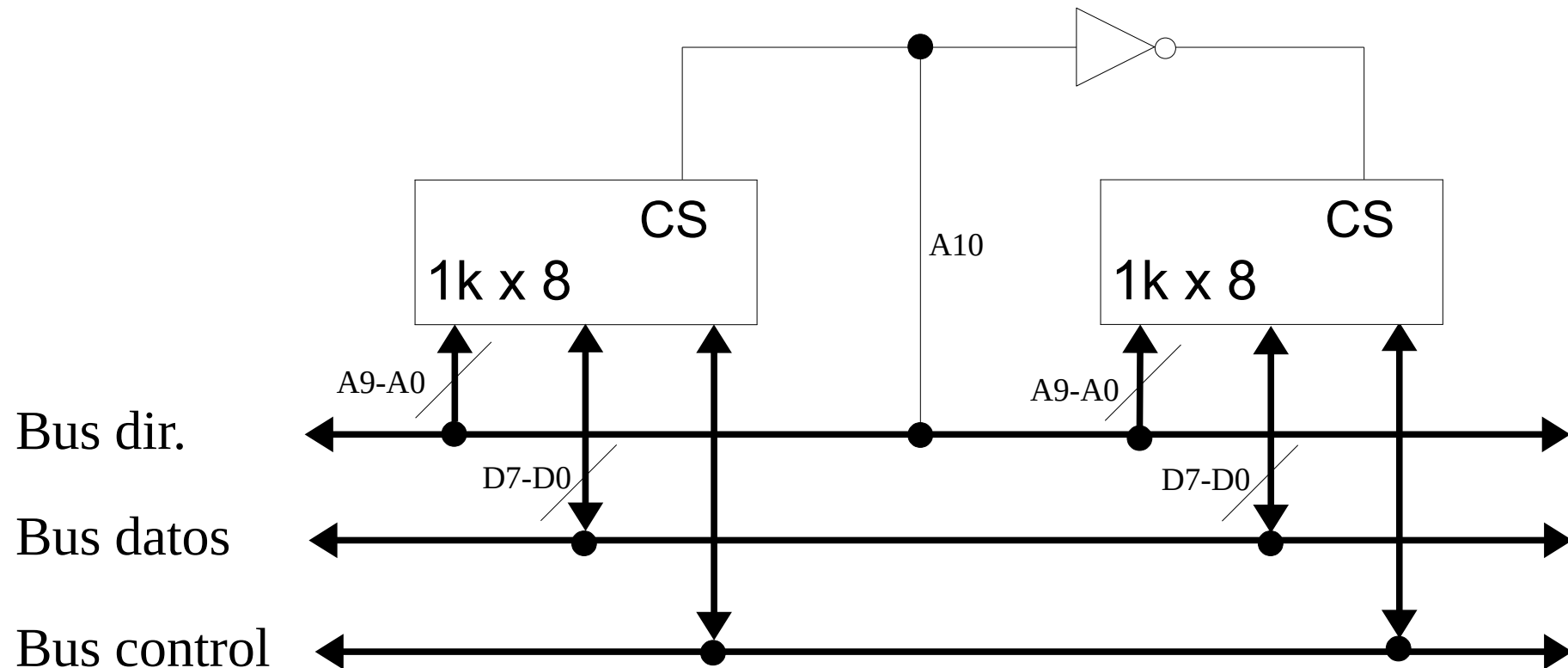
Cada celdilla tiene asignada una dirección (dirección de memoria) que es un número de 0 a $n-1$ (como si la memoria fuera un vector de n elementos).

Se suelen asociar formando módulos para aumentar su capacidad, ya sea aumentando el número de direcciones y/o tamaño de palabra.

Aunque la memoria tenga un tamaño de palabra mayor de 8 bits, usualmente se direcciona “byte a byte”. Ej.: memoria 1k palabras de 32bits => 4k direcciones.

Ej.: Asociación chips memoria

Con chips de 1kbyte crear una memoria de 2kbytes



CS => Chip Select

Lecturas/escrituras en RAM (1/2)

La RAM ha ido evolucionando para aumentar su rendimiento y, junto con otras técnicas, como las memorias caché, se intenta disminuir todo lo posible el cuello de botella que supone la diferencia entre la velocidad de la CPU y la del acceso a memoria.

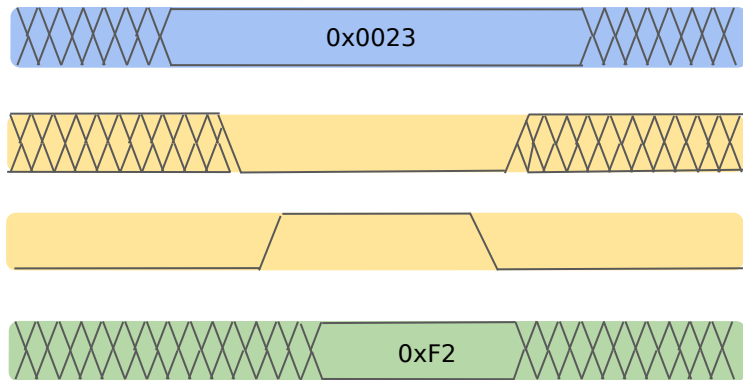
El acceso a las memorias actuales es complejo. Para simplificar, se va a suponer un acceso elemental en el que se utilizan las siguientes líneas del bus de control:

Mem: Indica si acceso a memoria.

$\overline{R/W}$: Indica si es lectura o escritura.

Lecturas/escrituras en RAM (2/2)

Ejemplo lectura



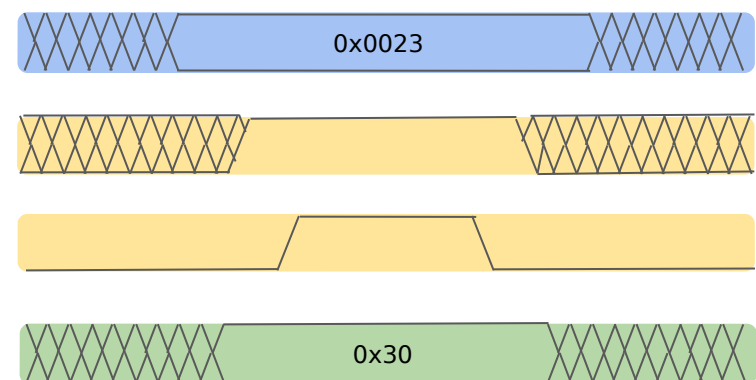
Bus dir.

\bar{R}/W

Mem

Bus datos

Ejemplo escritura



Entrada/Salida

¿Para qué valdría un ordenador sin al menos una entrada? ¿Y si no tiene ninguna salida?

Los dispositivos de entrada/salida sirven para interactuar con personas, herramientas/actuadores, sistemas de almacenamiento (memoria) externo, comunicarse con otros ordenadores, etc.

La velocidad de un dispositivo de E/S puede diferir mucho de la velocidad del procesador (por lentitud o por rapidez).

Componentes de un dispositivo externo

Transductor: Convierte energía de una forma a otra. Por ejemplo, el teclado tienen transductores que convierten la presión en señales eléctricas.

Buffer: Almacén de datos temporal de entrada y/o salida. Por ejemplo, hay teclados que tienen un buffer interno de 16 bytes.

Lógica de control: Gestiona el periférico, acepta instrucciones de control y genera información de estado. Por ejemplo, los teclados tienen un comando para establecer las luces led de los bloqueos de mayúsculas, numérico y scroll.

Generalmente los dispositivos externos se conectan a un módulo de entrada/salida que hace de “puente” entre el bus del sistema y ellos.

Módulo de E/S (1/3)

Las funciones principales de los módulos de E/S son:

- Control y temporización.
- Comunicación con el procesador.
- Comunicación con el dispositivo externo.
- Buffer de datos.
- Detección de errores.

La memoria interna de los módulos de E/S se denomina registros, que pueden ser de **datos** (información procedente de o con destino al dispositivo), de **estado** (lectura; indican el estado del dispositivo) y de **control** (escritura; para el envío de comandos).

Módulo de E/S (2/3)

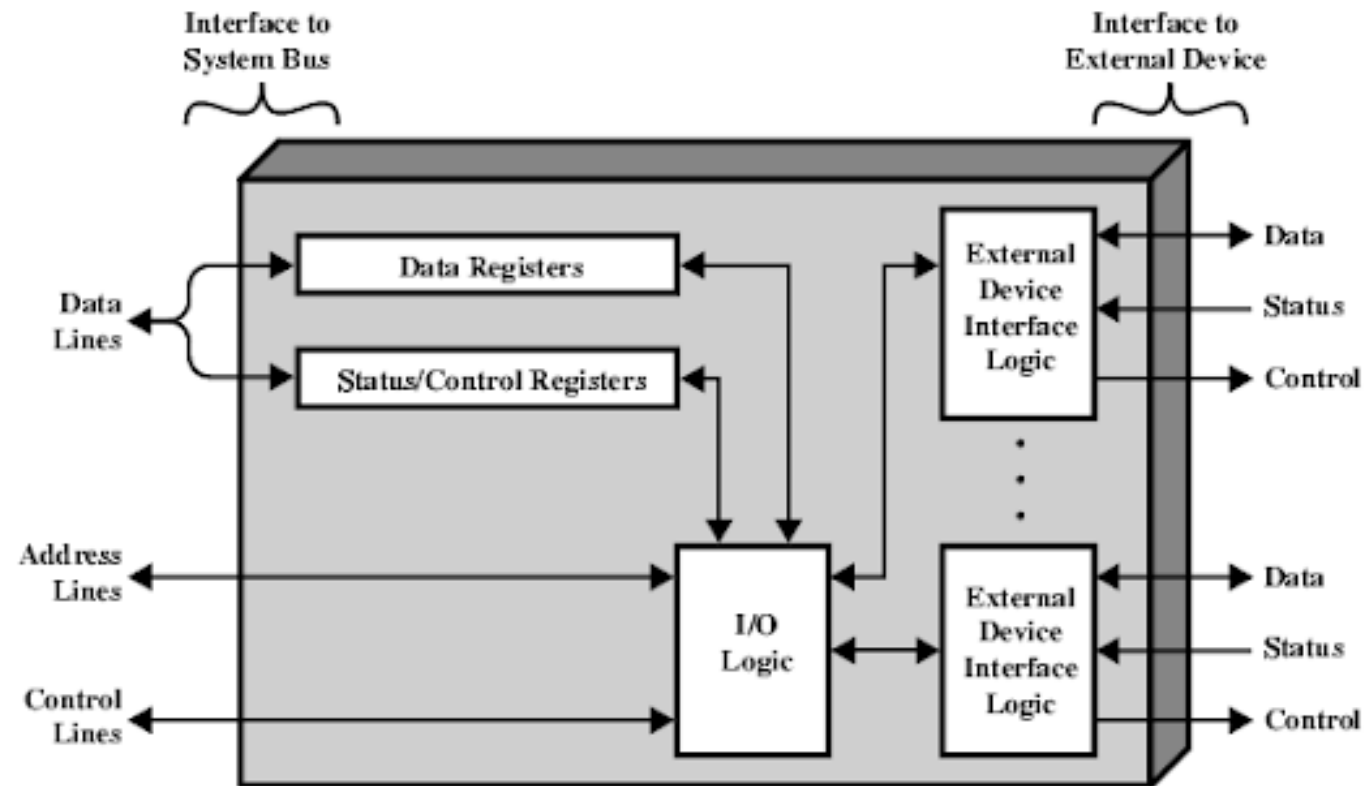


Figure 7.4 Block Diagram of an I/O Module

“Computer Organization and Architecture”. William Stallings

Módulo de E/S (3/3)

El conexionado de los módulos de E/S al bus de sistema (si estamos usando E/S independiente, esto es, no mapeada en memoria) incluye los buses de dirección, datos y varias líneas del bus de control, entre ellas IO (equivalente a Mem, pero en este caso para activar la E/S) y $\overline{R/W}$ (es la misma línea a la que se conecta la memoria).

Técnicas de entrada/salida

Polling: El programa “pregunta” o “pide” al dispositivo y espera la respuesta. ¿Con qué frecuencia? Ocupa recursos de forma innecesaria. Sencillo (no precisa hardware adicional).

Interrupciones: El programa, tras la petición de E/S, continúa. Cuando el dispositivo tiene el resultado, avisa a la CPU mediante una interrupción. Precisa líneas de control específicas para el protocolo de interrupción.

Acceso Directo a Memoria (DMA): Permite la lectura o escritura desde el módulo de E/S a la memoria y viceversa sin intervención de la CPU. Suele precisar de líneas de control específicas para el protocolo de DMA.

Referencias:

- “Textbook for Computer Architecture and Design course”. Edward L. Bosworth. Disponible en http://www.edwardbosworth.com/My5155Textbook/MyText5155_AFrontMatter.htm
- Wikipedia.
- “Computer Organization and Architecture”. William Stallings, 2015. Pearson.