

Organización del computador

Lógica digital

Jerarquía de máquina

Nivel 6	Usuario	Programa ejecutables
Nivel 5	Lenguaje de alto nivel	C++, Java, Python, etc.
Nivel 4	Lenguaje ensamblador	Assembly code
Nivel 3	Software del sistema	Sistema operativo, bibliotecas, etc.
Nivel 2	Lenguaje de máquina	Instruction Set Architecture (ISA)
Nivel 1	Unidad de control	Microcódigo / hardware
Nivel 0	Lógica digital	Circuitos, compuertas, memorias



- > Cada nivel funciona como una máquina abstracta que oculta la capa anterior
- > Cada nivel es capaz de resolver determinado tipo de problemas a partir de comprender un tipo de instrucciones específico
- > La capa inferior es utilizada como servicio

Lógica digital

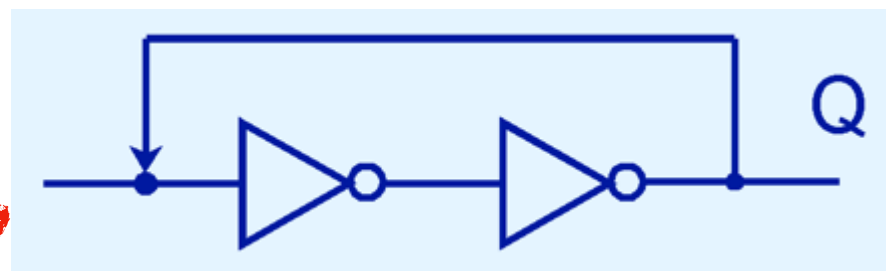
- > Los circuitos operan con dos valores eléctricos
- > Pueden ser interpretados como 1 ó 0 y pensarlos como información
- > Pueden ser interpretados como verdadero o falso y pensarlos como valores lógicos
- > Los circuitos combinatorios tienen un comportamiento funcional, dada cierta entrada arrojan una salida determinada pero no permiten el almacenamiento de valores

Flip-flops

- ➤ Computar está vinculado a la posibilidad de almacenar; aun cuando se desee aplicar una función booleana, será necesario almacenar los valores de entrada y dónde dejar el resultado
- ➤ Más aun si pretendemos implementar un modelo de cómputo que organiza operaciones secuencialmente (por ejemplo, multiplicación como sumas sucesivas)
- ➤ Los **flip-flops** son circuitos secuenciales que permiten almacenar el estado de un bit, es decir, guardar un valor y sirven a los efectos de construcción de memorias
- ➤ Se denominan flip-flop porque conservan su valor hasta que se lo reemplaza por uno diferente

Realimentación

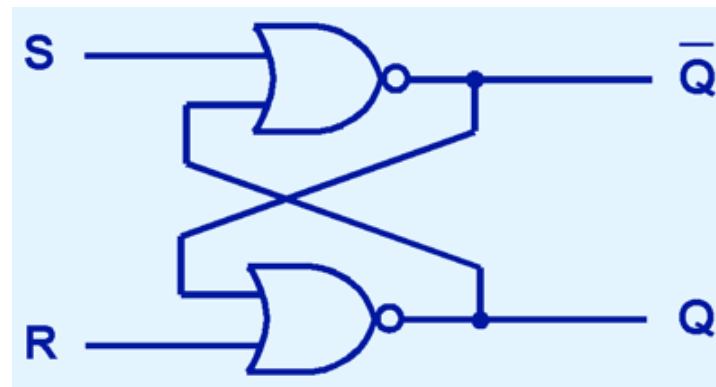
- > Un concepto clave en la implementación de flip-flops es el de realimentación
- > Conectar una salida de un circuito a una entrada del mismo se denomina **realimentación**



- > Si la entrada del circuito es 0, en la salida siempre observaremos 0 y si es 1 entonces la salida siempre será 1

Flip-flop RS

- > Uno de los circuitos secuenciales más conocidos es el **Flip-flop RS**
- > Se denomina RS por Reset/Set

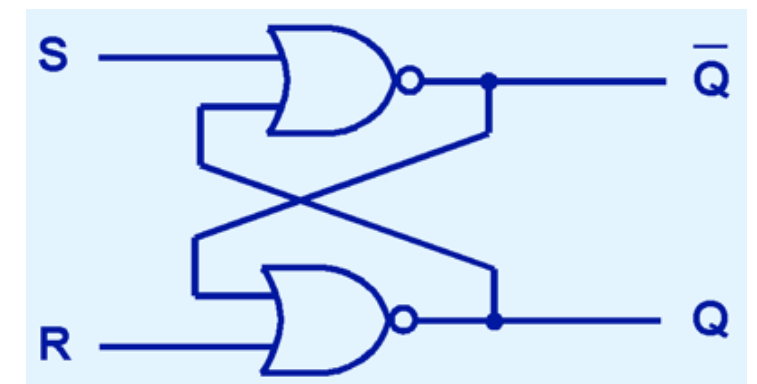


- > Los Flip-flop RS que solo cuentan con líneas de control se denominan asincrónicos puesto que el estado puede ser modificada cada vez que cambia R o S

Flip-flop RS

- A los efectos de su análisis, los circuitos secuenciales son pensados con una entrada adicional pues la salida se observa en un instante determinado de tiempo $Q(t+1)$ y depende de su estado en el instante anterior $Q(t)$
- Adicionalmente, el t es determinado por otra entrada denominada clock
- Debido a la indefinición provocada cuando $S = 1$ y $R = 1$ este flip-flop se denomina inestable

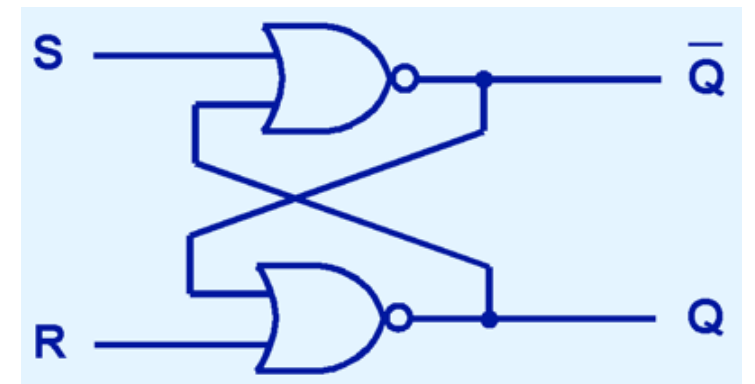
Present State			Next State
S	R	$Q(t)$	$Q(t+1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined



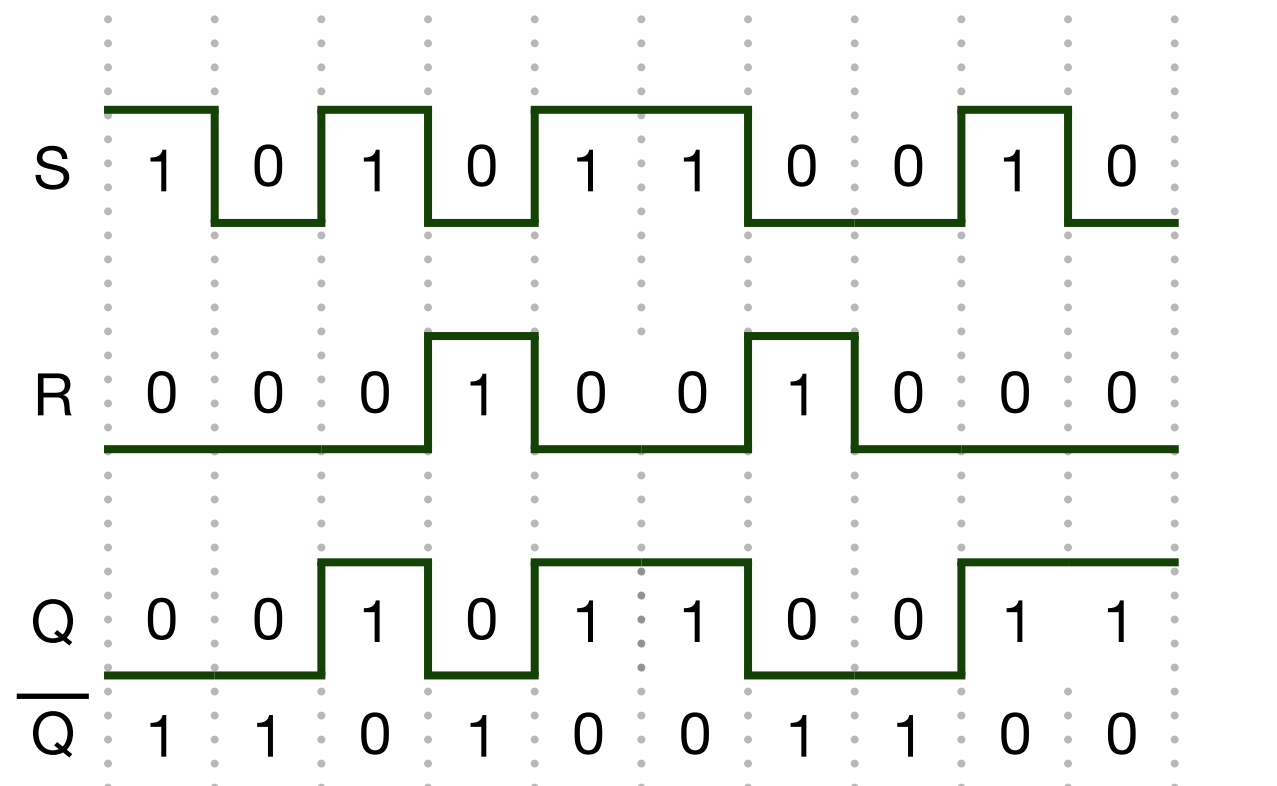
Flip-flop RS

- A los efectos de su análisis, los circuitos secuenciales son pensados con una entrada adicional pues la salida se observa en un instante determinado de tiempo $Q(t+1)$ y depende de su estado en el instante anterior $Q(t)$
- Adicionalmente, el t es determinado por otra entrada denominada clock
- Debido a la indefinición provocada cuando $S = 1$ y $R = 1$ este flip-flop se denomina inestable

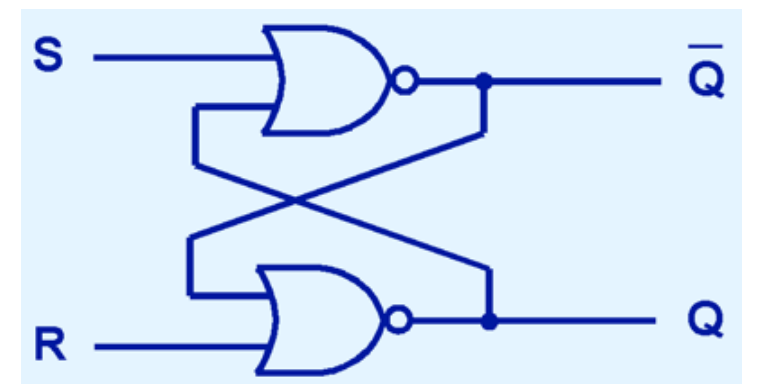
S	R	$Q(t+1)$
0	0	$Q(t)$ (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined



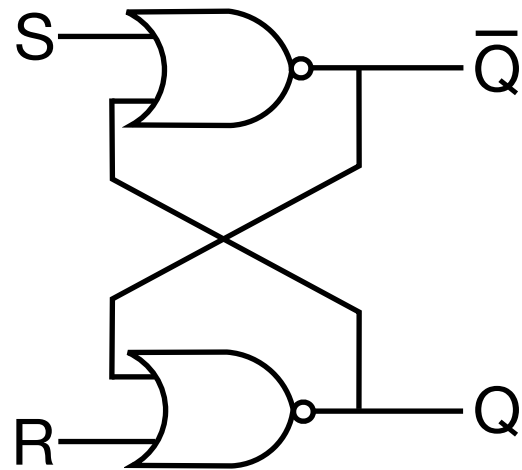
Flip-flop RS



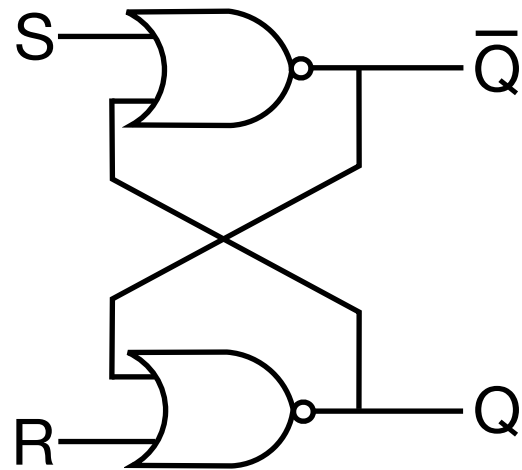
S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined



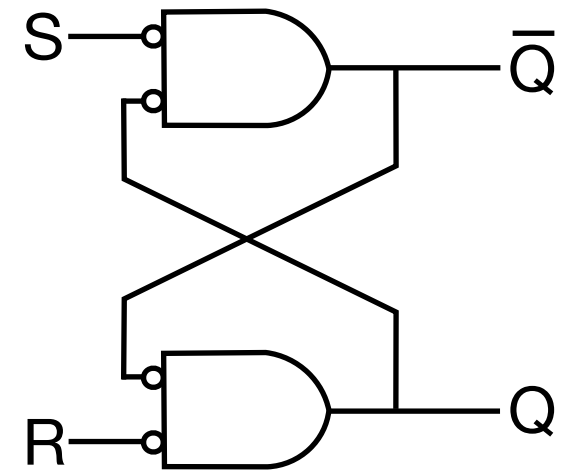
Flip-flop RS (NAND)



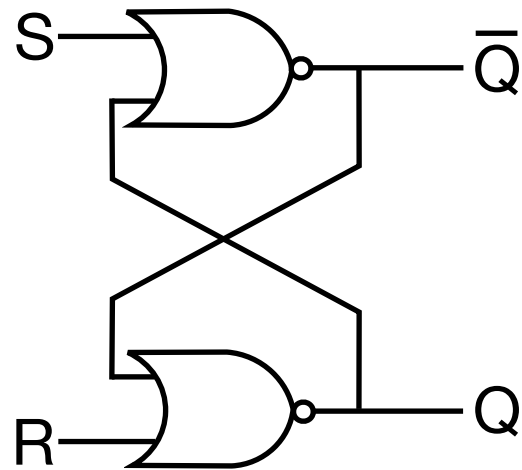
Flip-flop RS (NAND)



$$\overline{S + Q(t)} = \bar{S} \cdot \overline{Q(t)}$$
$$\overline{R + Q(t)} = \bar{R} \cdot \overline{Q(t)}$$

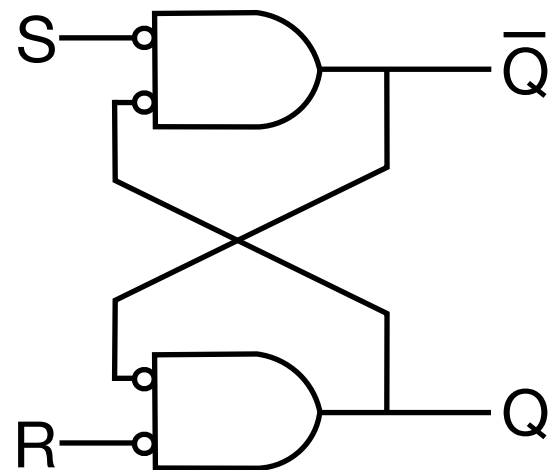
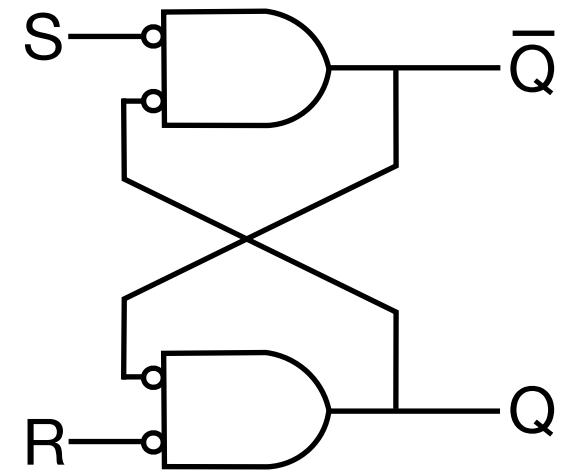


Flip-flop RS (NAND)

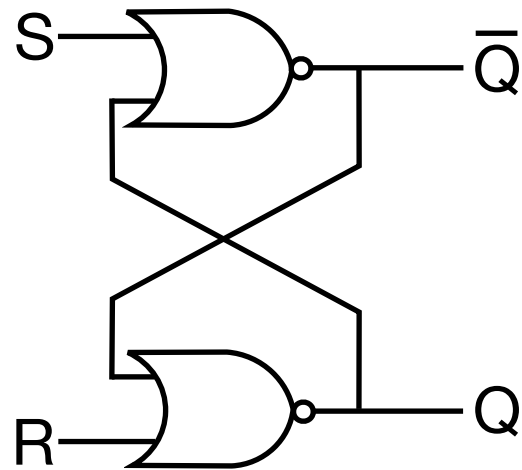


$$\overline{S + Q(t)} = \bar{S} \cdot \overline{Q(t)}$$

$$\overline{R + Q(t)} = \bar{R} \cdot \overline{Q(t)}$$

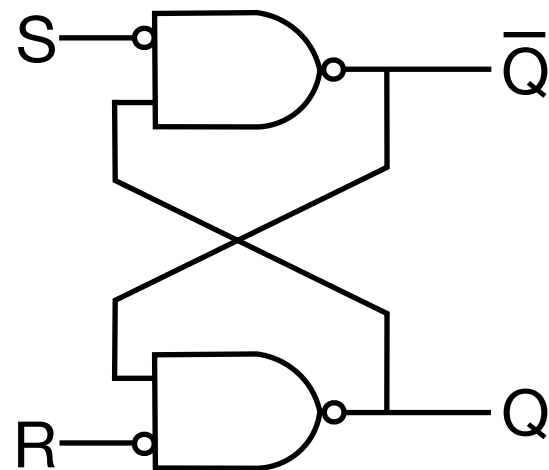
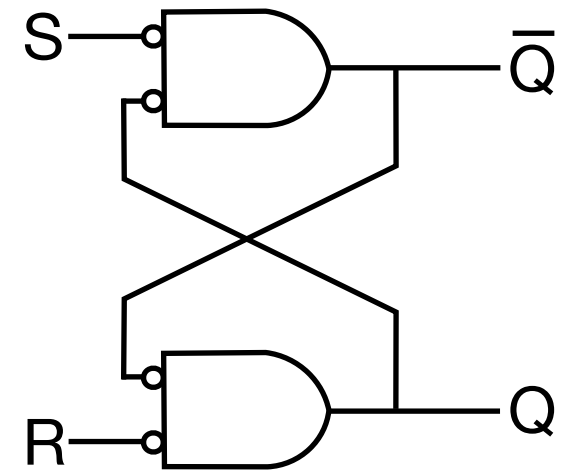


Flip-flop RS (NAND)



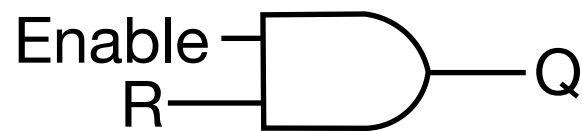
$$\overline{S + Q(t)} = \bar{S} \cdot \overline{Q(t)}$$

$$\overline{R + Q(t)} = \bar{R} \cdot \overline{Q(t)}$$



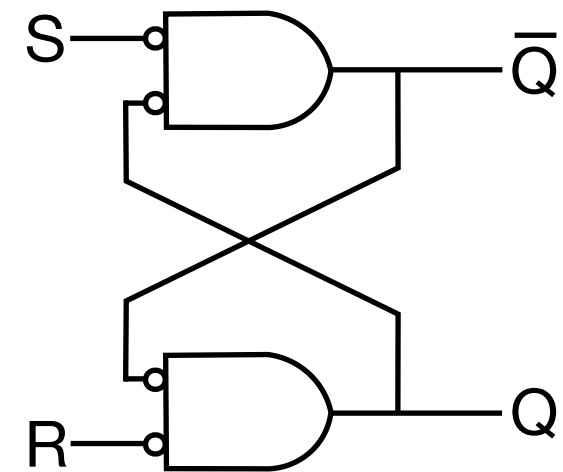
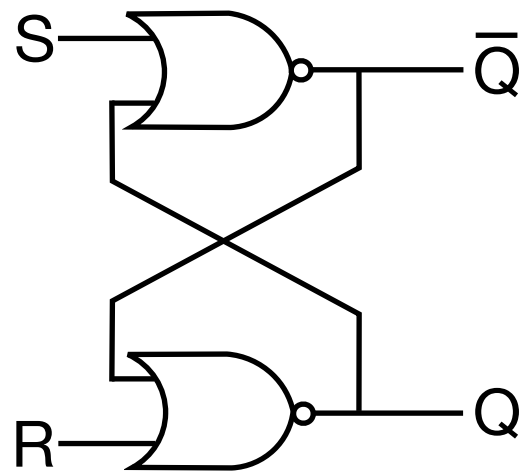
Habilitación de datos (Enable)

- Es común que los circuitos requieran una línea de habilitación de datos que permita determinar cuándo leer / grabar información a pesar del valor presente en la entrada
- Las compuertas AND proveen un mecanismo sencillo para implementar una señal de Enable

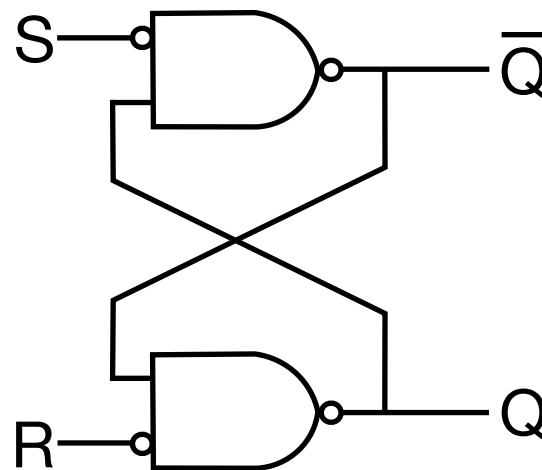


R	Enable	Q
X	0	0
0	1	0
1		1

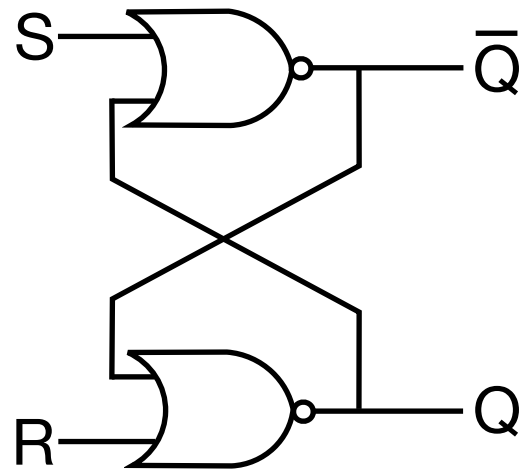
Flip-flop RS (NAND+Enable)



$$\overline{S + Q(t)} = \bar{S} \cdot \overline{Q(t)}$$
$$\overline{R + Q(t)} = \bar{R} \cdot \overline{Q(t)}$$



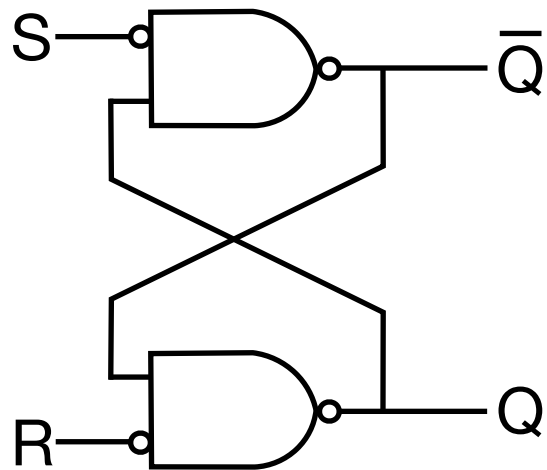
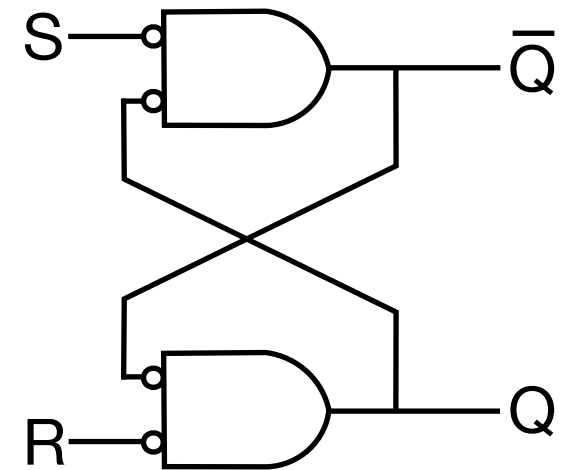
Flip-flop RS (NAND+Enable)



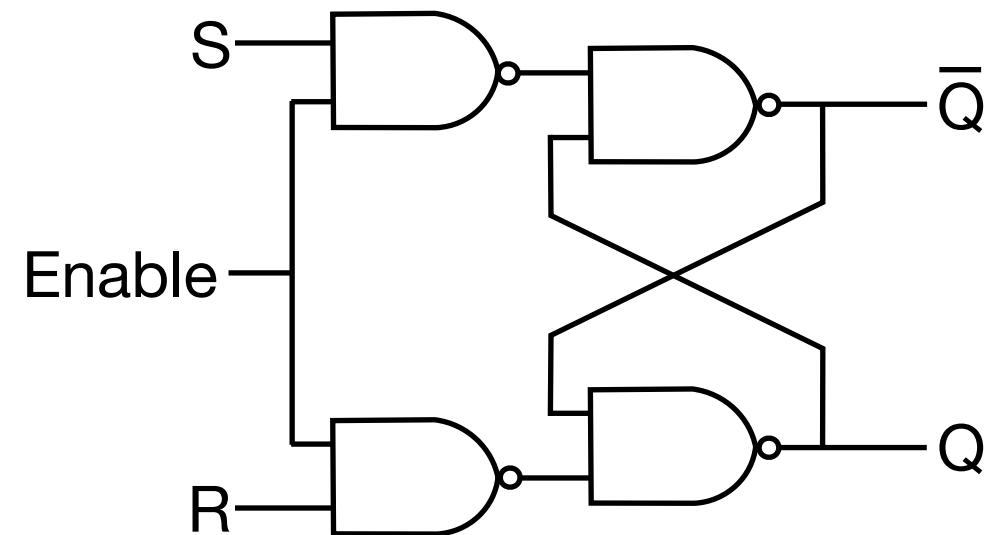
De Morgan

$$\overline{S + Q(t)} = \bar{S} \cdot \overline{Q(t)}$$

$$\overline{R + Q(t)} = \bar{R} \cdot \overline{Q(t)}$$

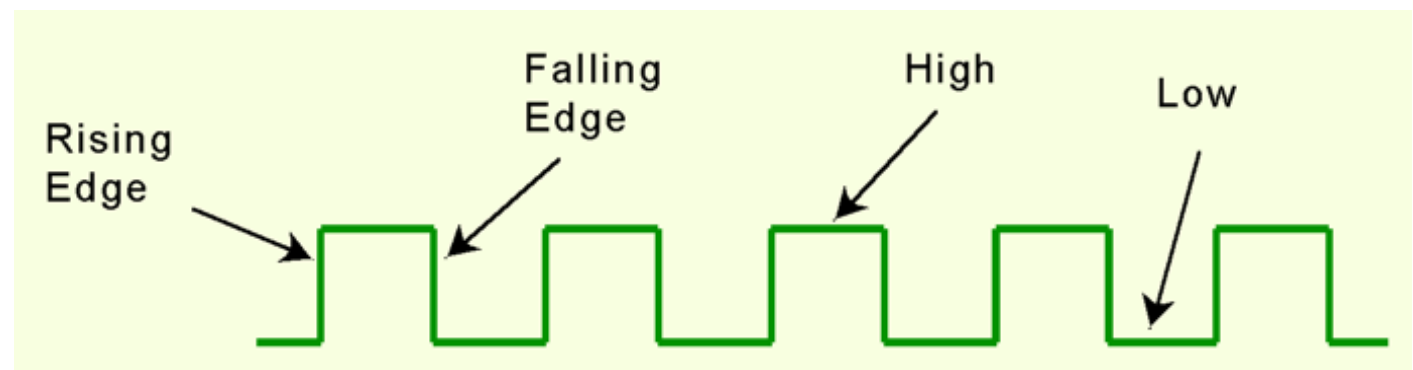


Enable

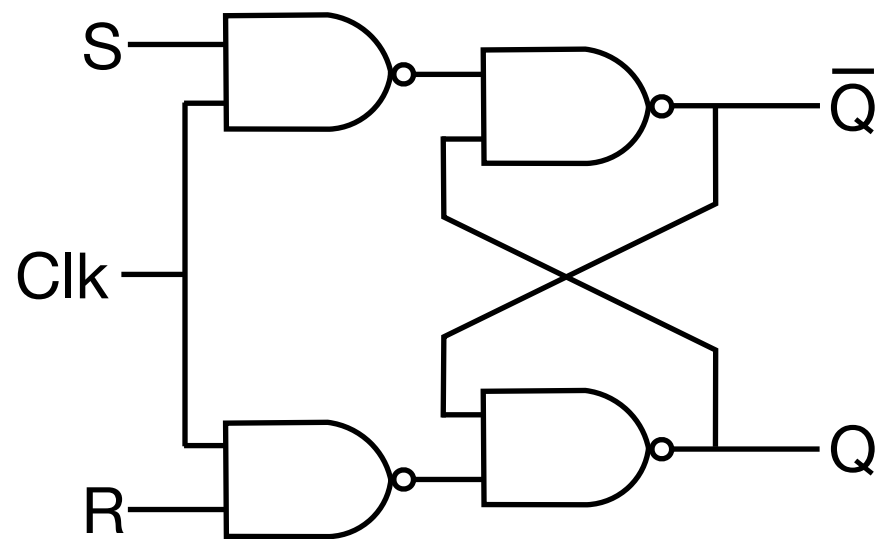


Circuitos sincrónicos

- > Los circuitos sincrónicos funcionan a partir de modificar su estado según lo indica una base de tiempo generada por un reloj
- > Las salidas de un circuito secuencial dependen de las entradas y del estado anterior de este, por lo que es necesario ordenar los eventos de observación
- > Un reloj es un circuito capaz de producir una señal oscilante con frecuencia uniforme
- > Los cambios de estado en los circuitos se producen usando la señal del reloj, ya sea usando sus flancos o sus niveles



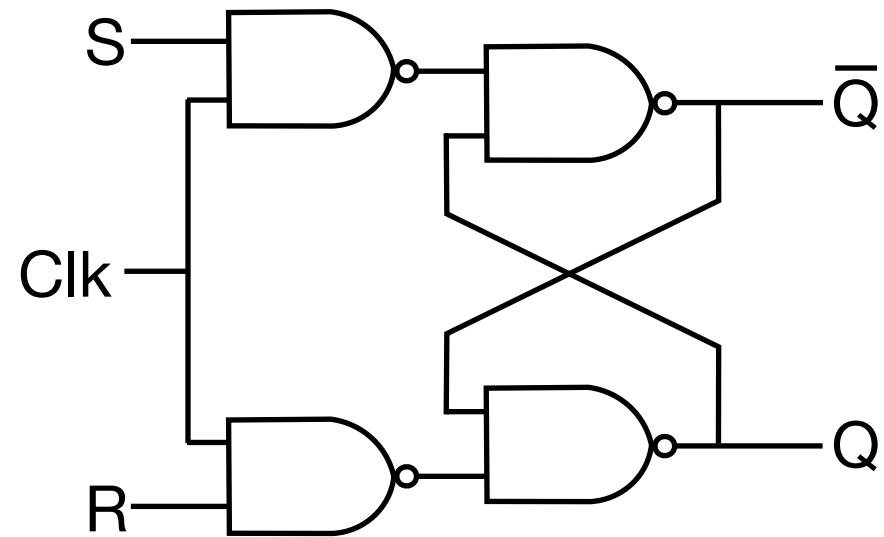
Flip-flop RS (Sincrónico)



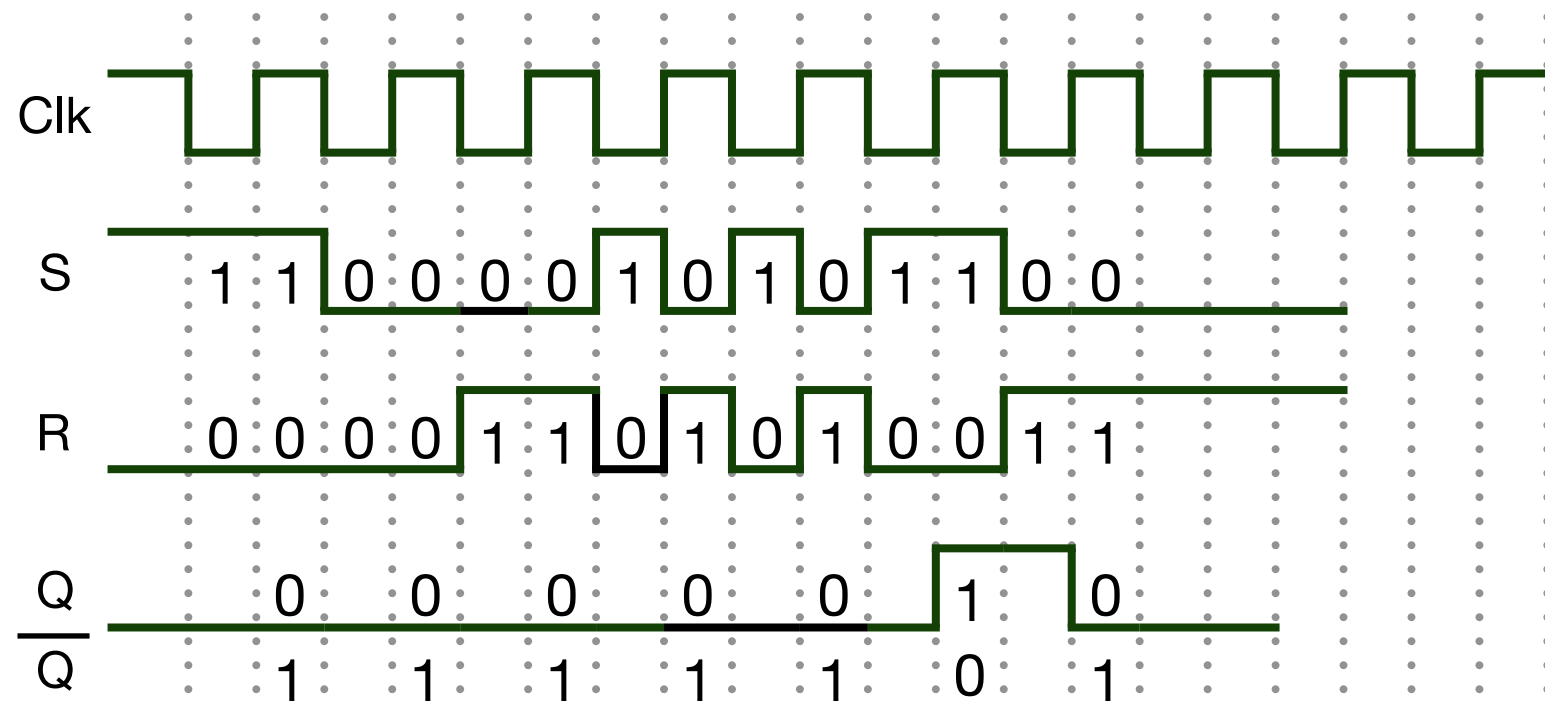
S	R	Clk	Q(t+1)
X	X	0	Q(t)
0	0	1	Q(t)
0	1	1	0
1	0	1	1
1	1	1	Undefined

- Los flip-flop sincrónicos utilizan la línea de habilitación de datos para determinar en qué momento es posible modificar el estado del circuito secuencial.

Flip-flop RS (Sincrónico)



S	R	Clk	Q(t+1)
X	X	0	Q(t)
0	0	1	Q(t)
0	1	1	0
1	0	1	1
1	1	1	Undefined



Flip-flop D

S	R	Clk	Q(t+1)
X	X	0	Q(t)
0	0	1	Q(t)
0	1	1	0
1	0	1	1
1	1	1	Undefined

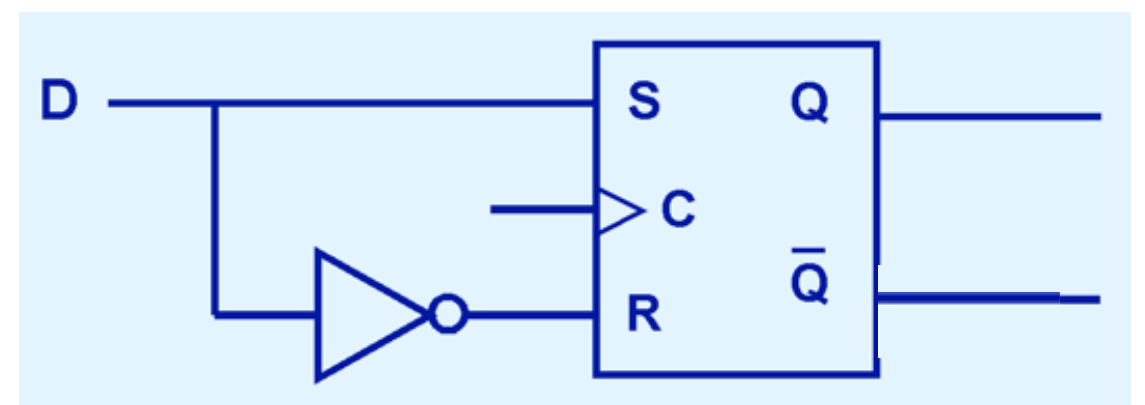
Flip-flop D

S	R	Clk	Q(t+1)
X	X	0	Q(t)
0	0	1	Q(t)
0	1	1	0
1	0	1	1
1	1	1	Undefined

Flip-flop D

S	R	Clk	Q(t+1)
X	X	0	Q(t)
0	0	1	Q(t)
0	1	1	0
1	0	1	1
1	1	1	Undefined

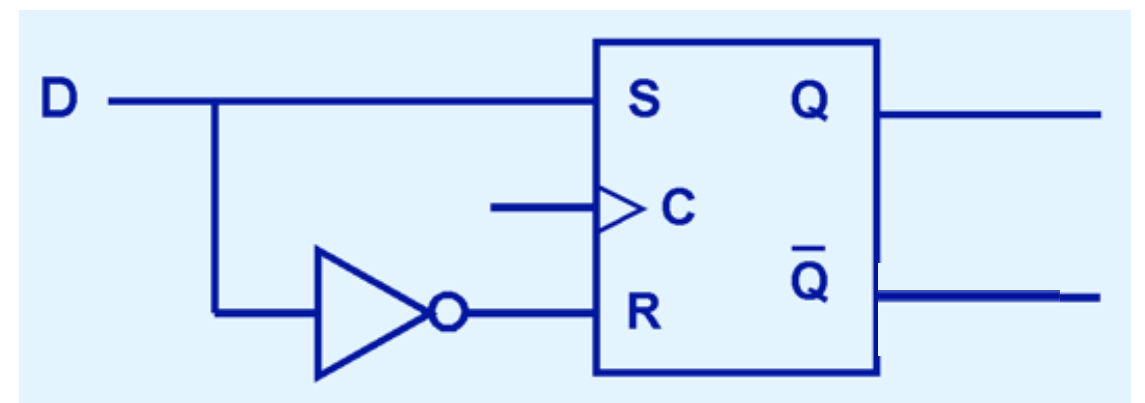
D	Q(t+1)
0	0
1	1



Flip-flop D

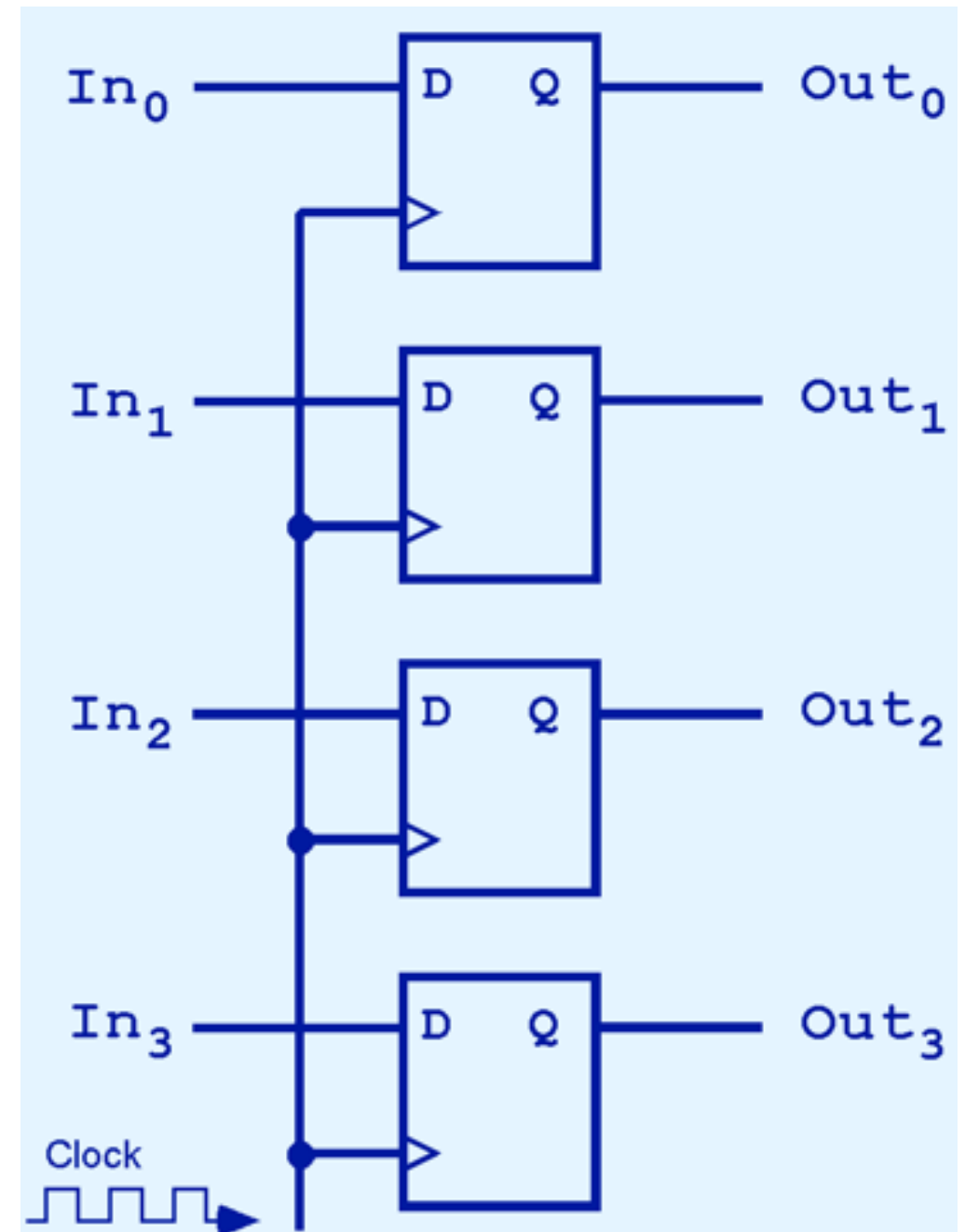
- > Elimina el estado inestable a partir de vincular las entrada para que sean mutuamente inversas
- > El circuito retiene el valor presente en la entrada al momento en el que la señal Clk habilita los datos. Solo volverá a cambiar al siguiente pulso
- > El **flip-flop D** es el circuito fundamental en la construcción de 1 bit de memoria estática (tecnología usada para implementar registros)

D	Q (t+1)
0	0
1	1



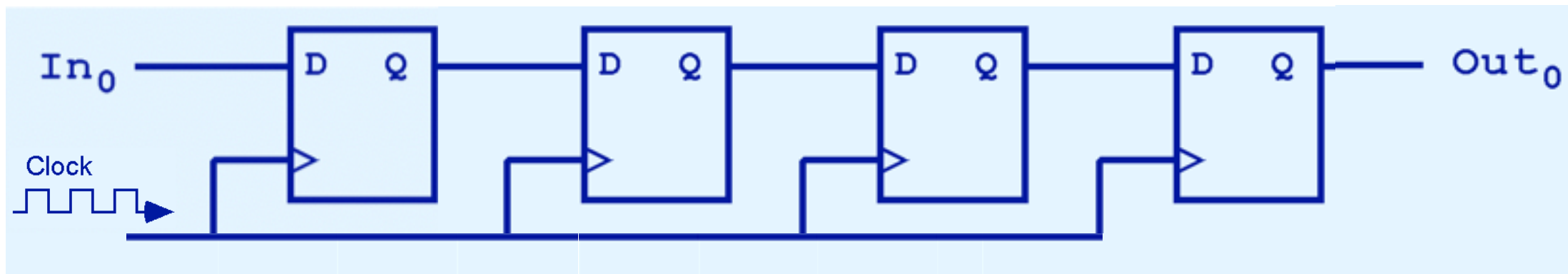
Registros (flip-flop D)

- Registro de 4 bits formado por 4 circuitos flip-flop D



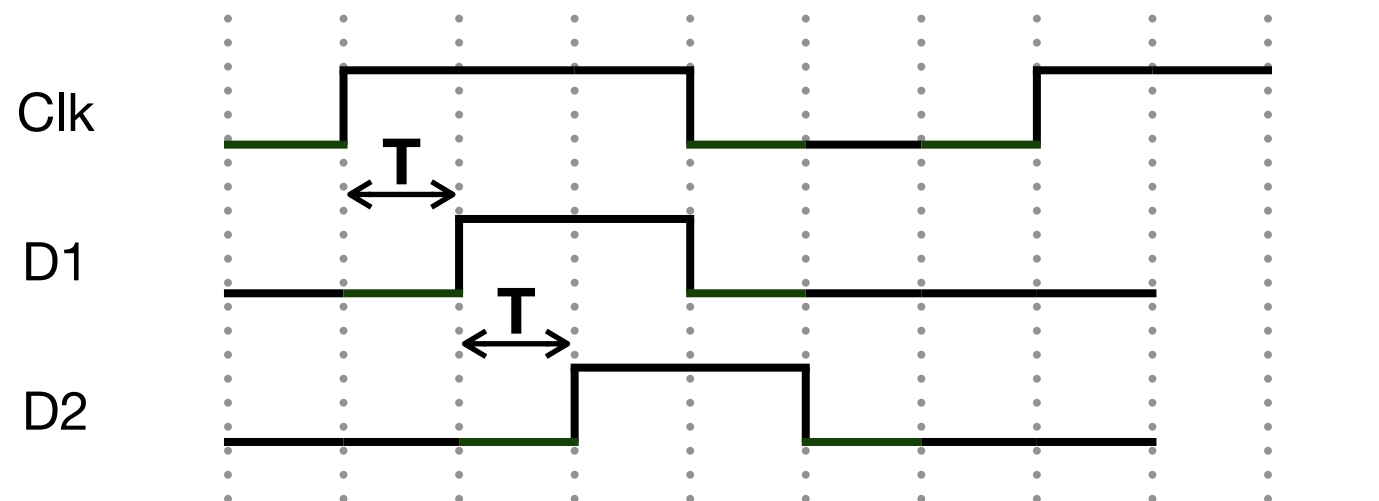
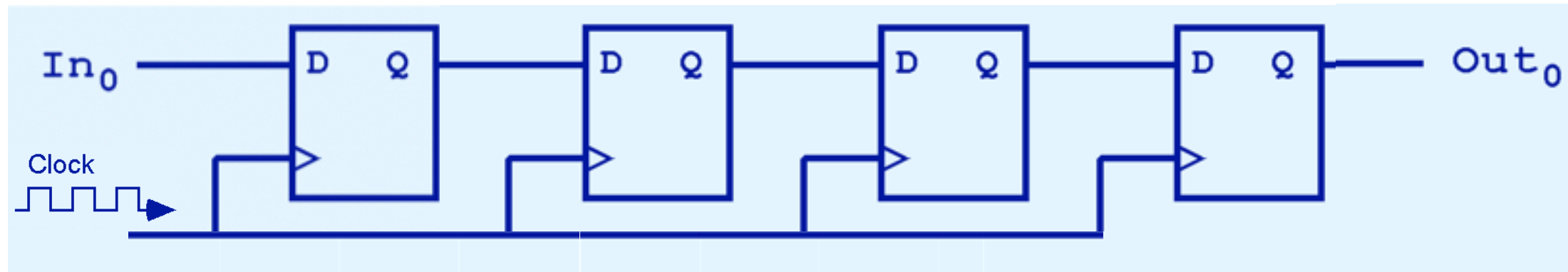
Registros (flip-flop D)

- Registro de desplazamiento de 4 bits formado por 4 circuitos flip-flop D



Registros (flip-flop D)

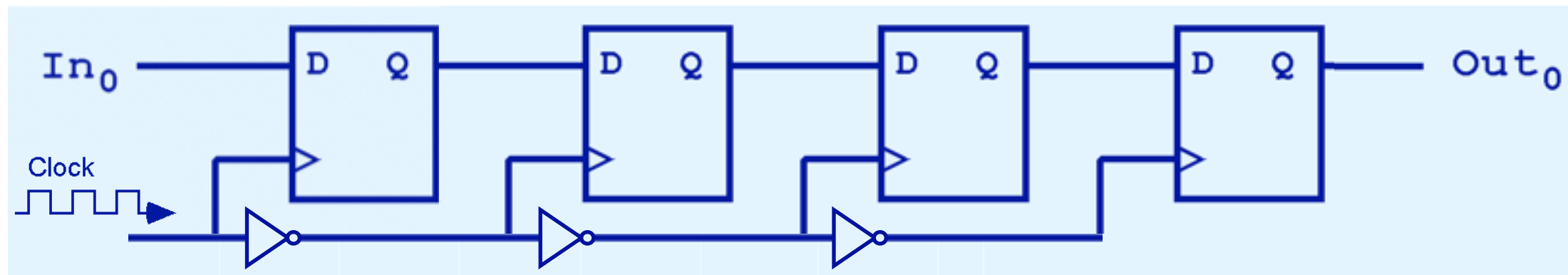
- Registro de desplazamiento de 4 bits formado por 4 circuitos flip-flop D



- Cuando se conectan flip-flop en cascada el tiempo de propagación T , que es el tiempo que tarda un cambio en el valor de la entrada en ser observable en la salida, puede hacer que ambos tomen el mismo valor.

Registros (flip-flop D)

- Registro de desplazamiento de 4 bits formado por 4 circuitos flip-flop D

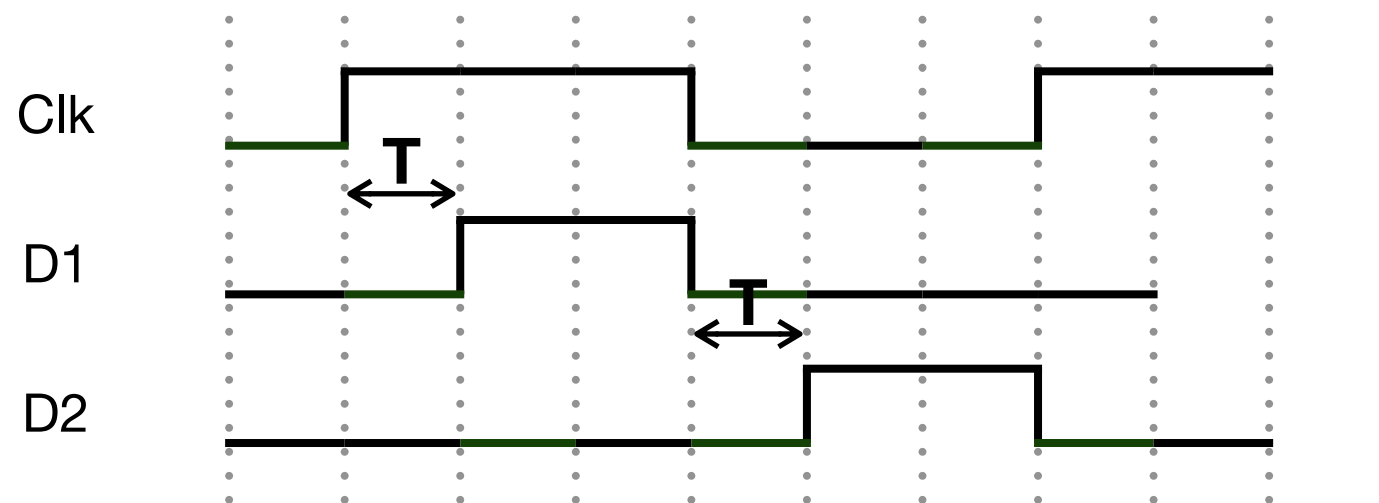
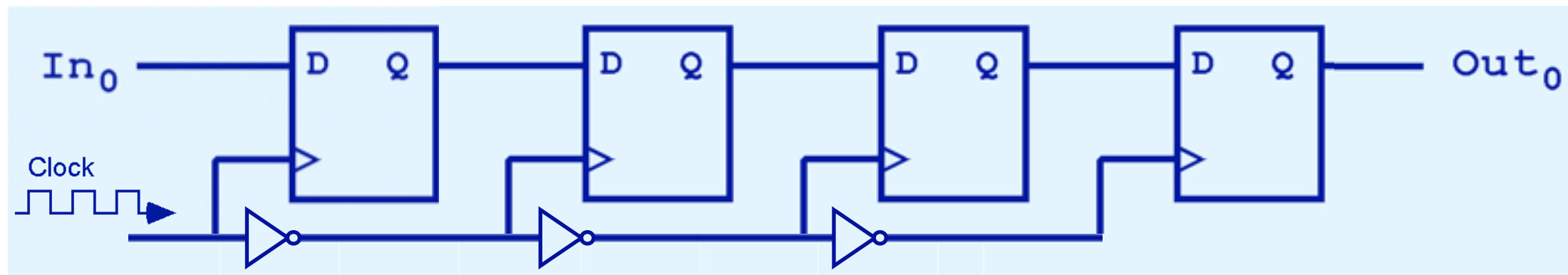


$\longleftrightarrow T$

$\longleftrightarrow T$

Registros (flip-flop D)

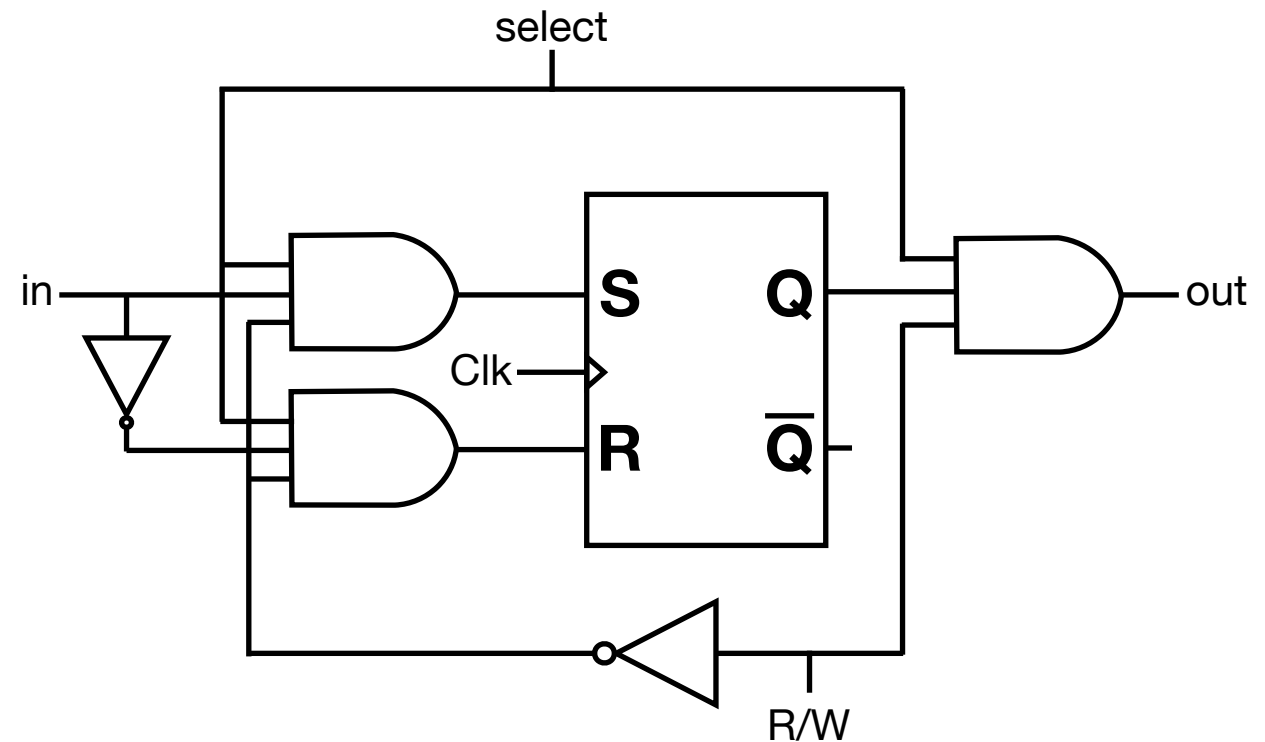
- Registro de desplazamiento de 4 bits formado por 4 circuitos flip-flop D



- Al negar la señal de reloj el cambio en la entrada del siguiente flip-flop solo modificará el estado interno del circuito al siguiente medio ciclo de reloj

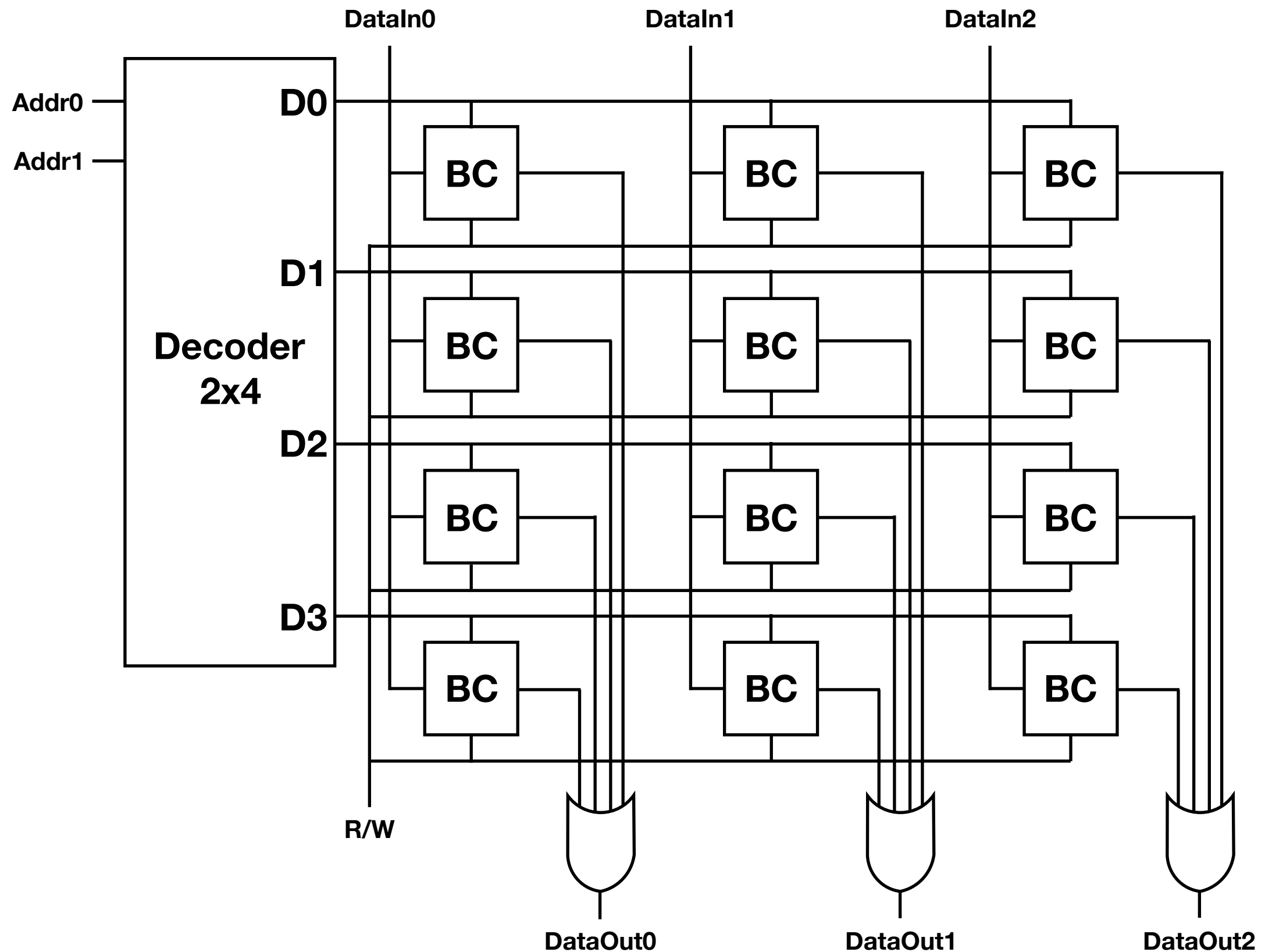
Celda de memoria (BC)

- > **in**: entrada del circuito
- > **out**: salida del circuito
- > **Clk**: señal de reloj para sincronización
- > **select**: señal de habilitación de datos de entrada y salida
- > **R/W**: señal que determina si en el siguiente ciclo de reloj se leerá o escribirá

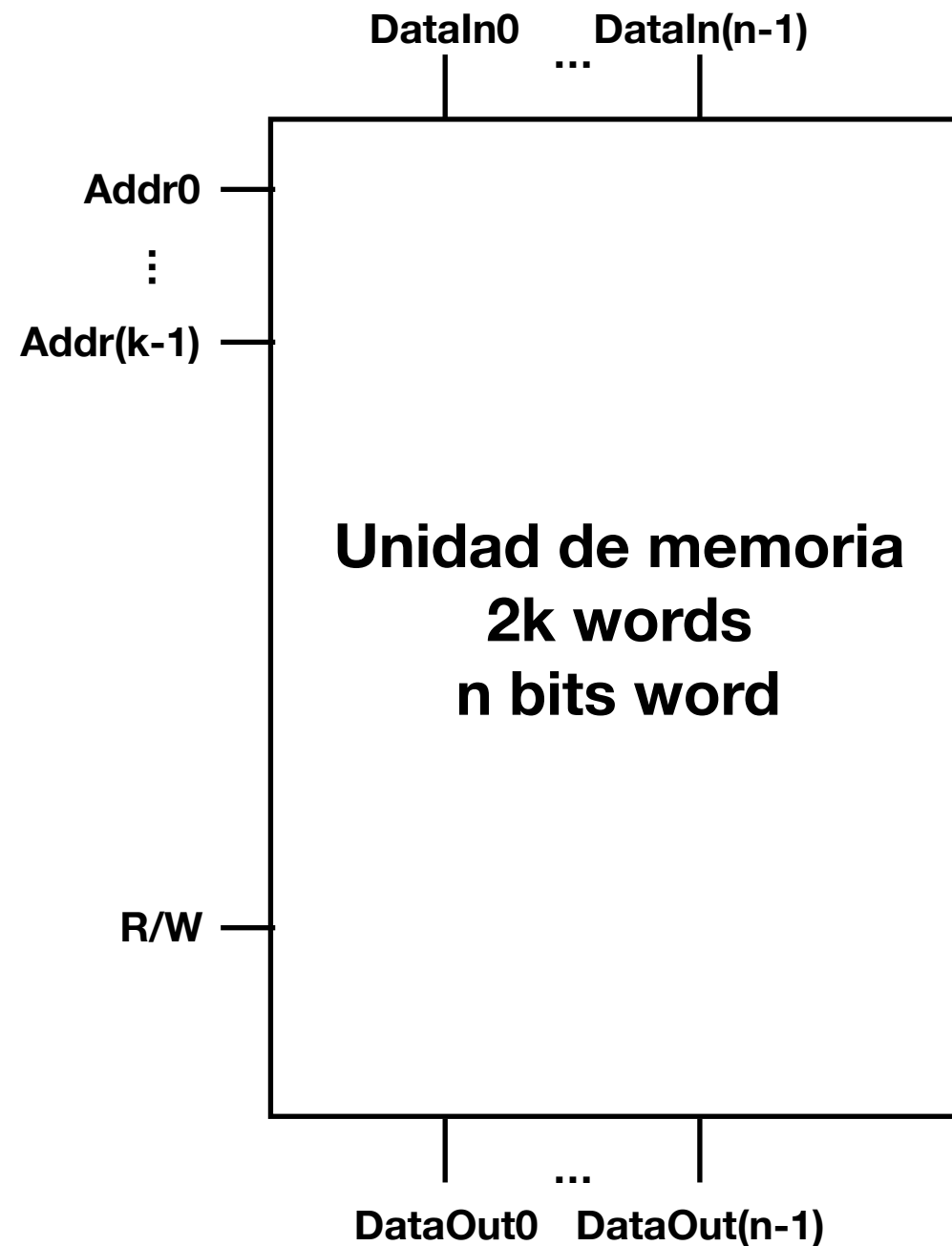


- > Es interesante observar que la entrada ingresa a R negada con respecto a S reproduciendo el comportamiento del flip-flop D

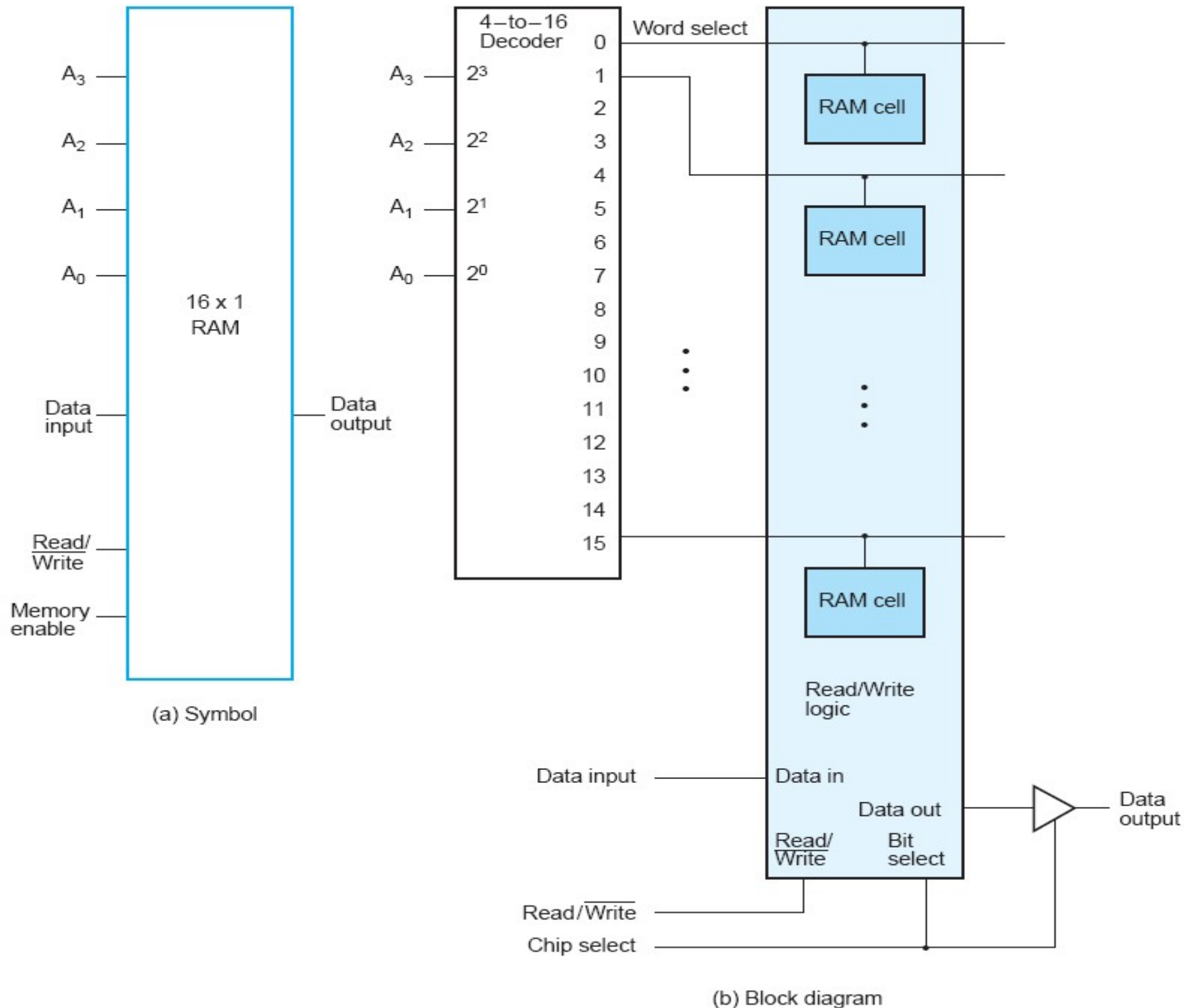
Memoria 4 word x 3 bits



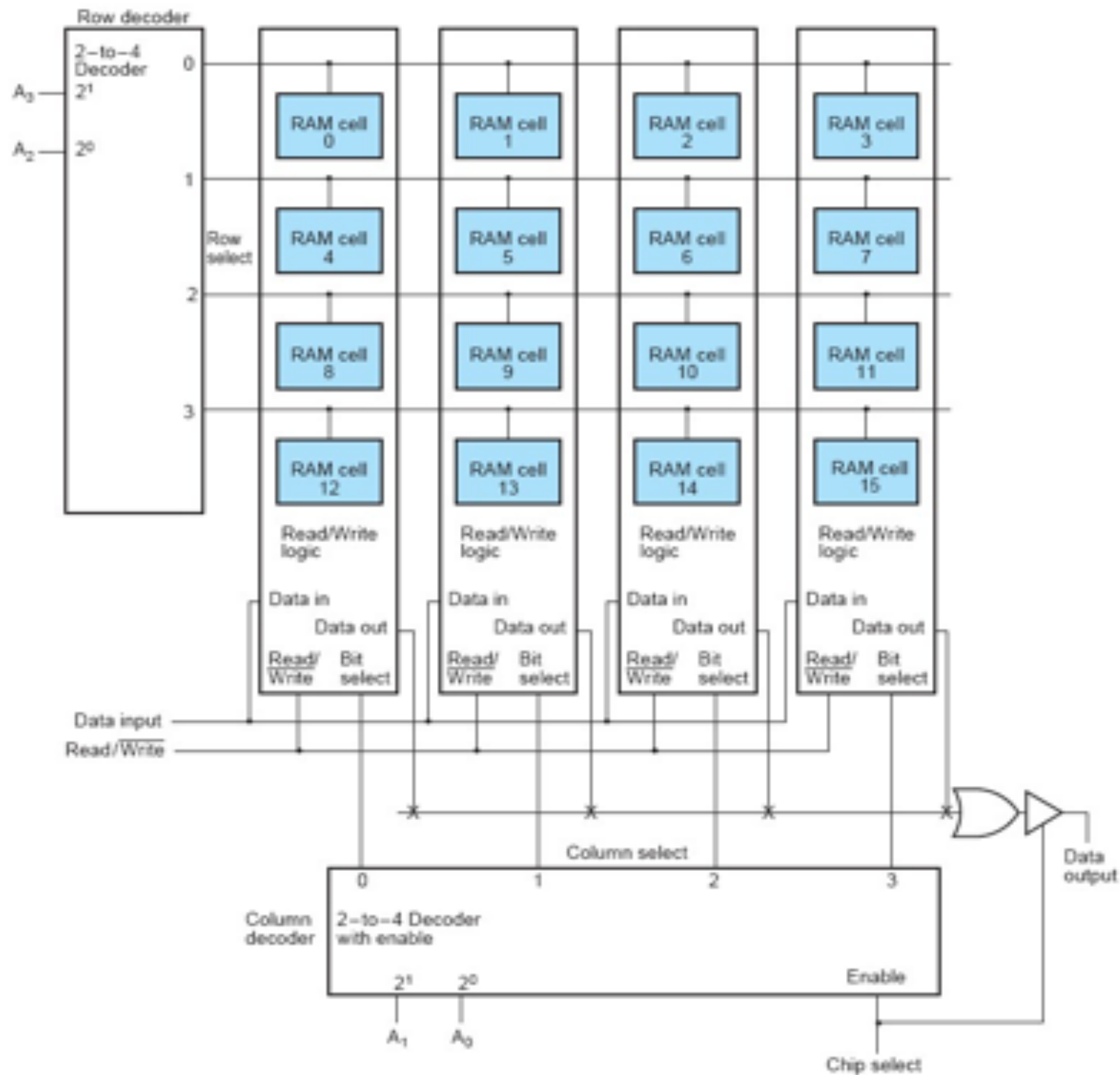
Memoria 2^k word x n bits



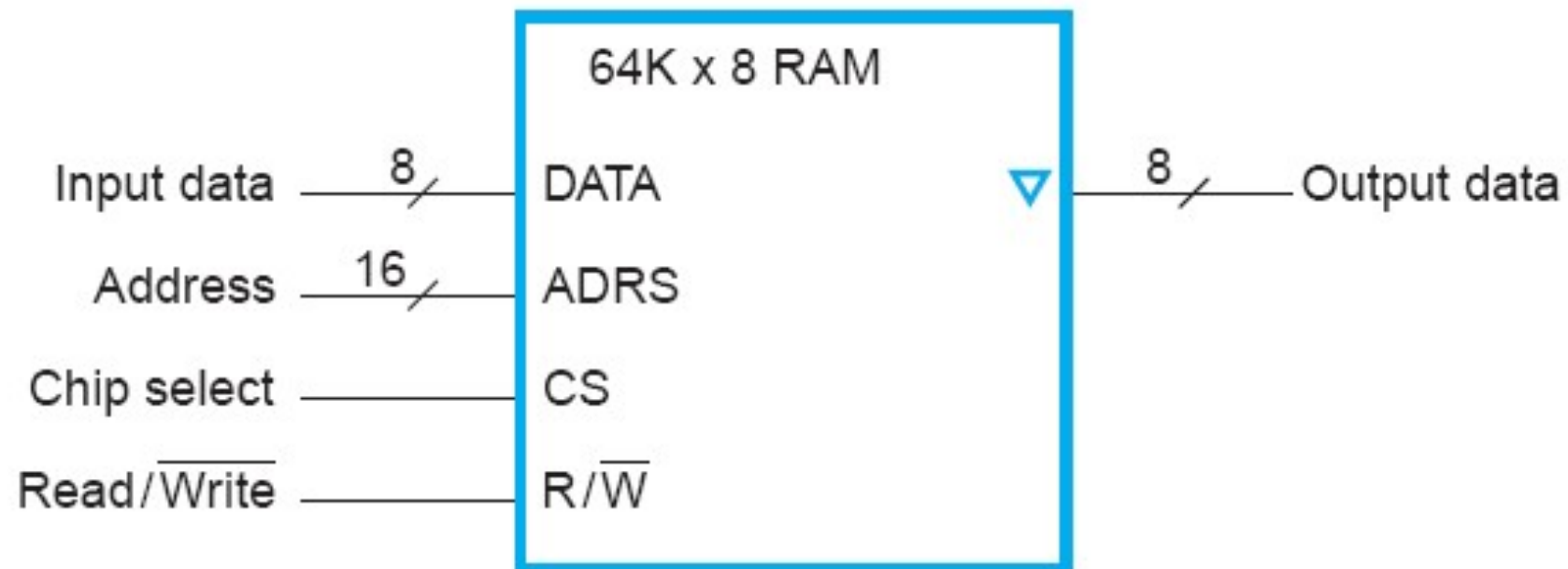
Memoria 16 word x 1 bits



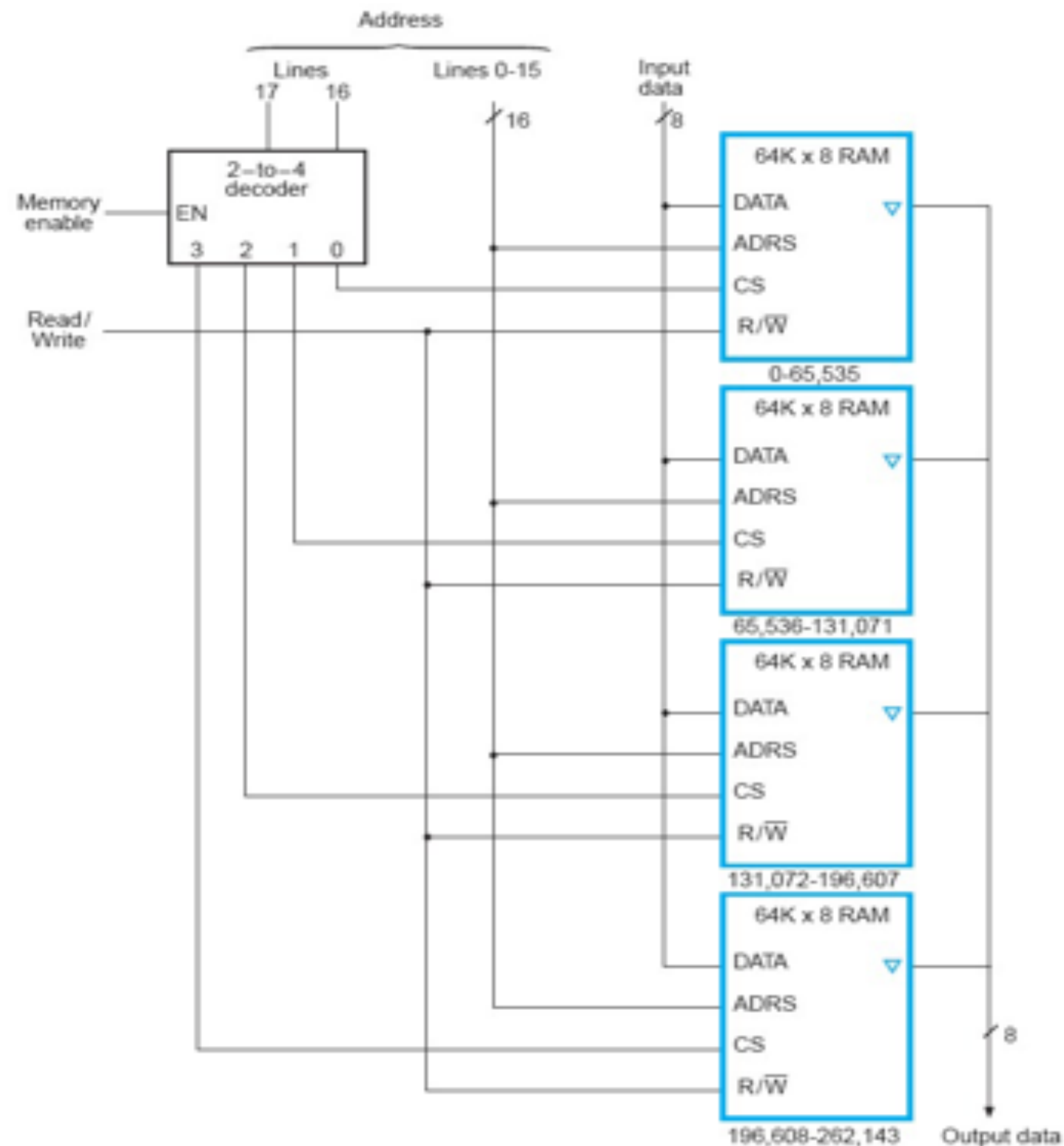
Memoria 16x1 usando memorias de 4x4



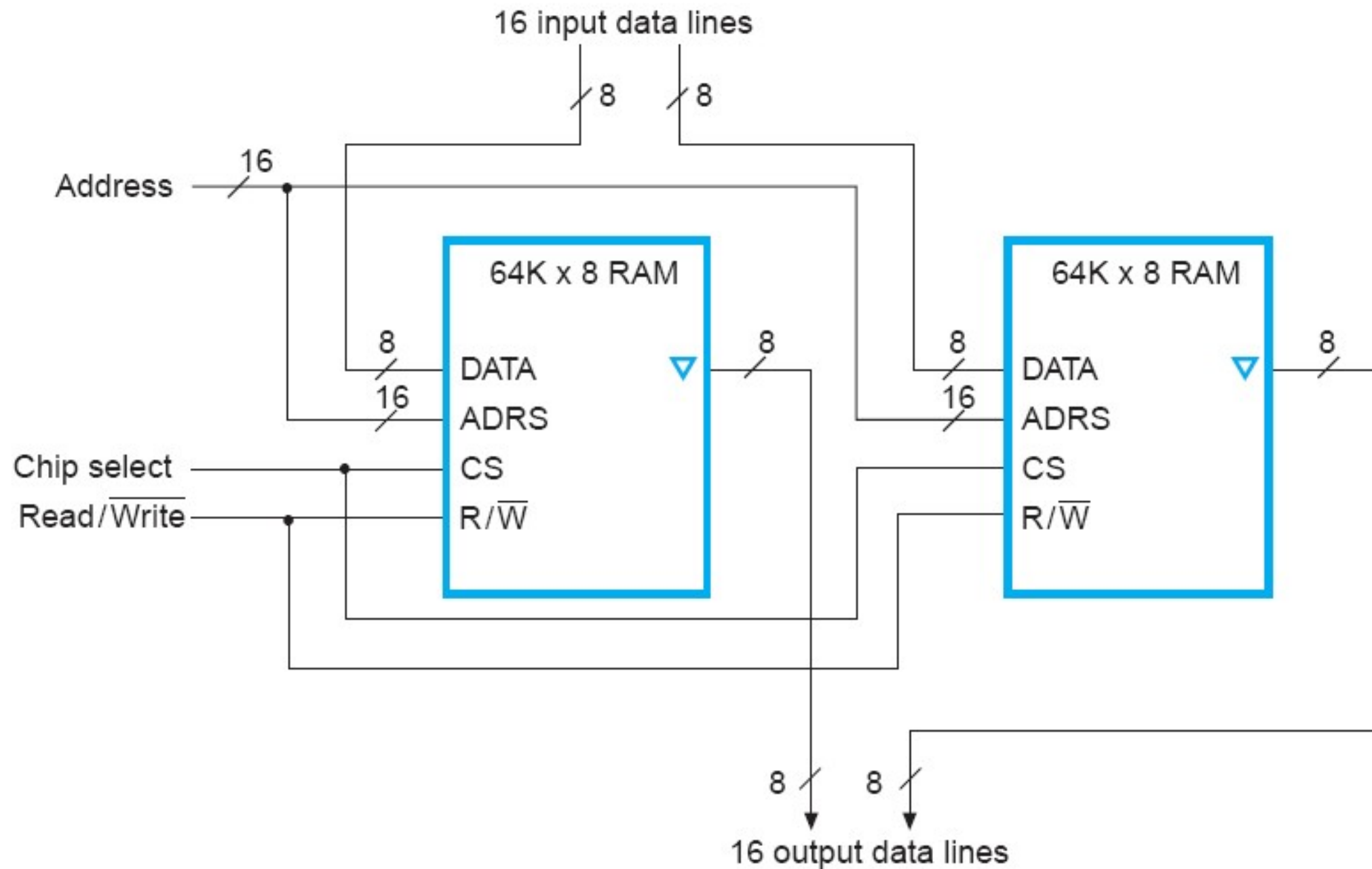
Memoria 64Kx8



Memoria 256Kx8 usando memorias de 64Kx8



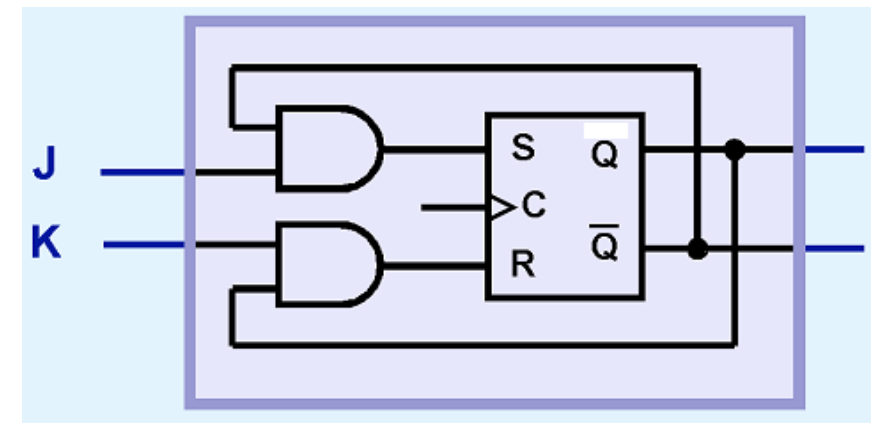
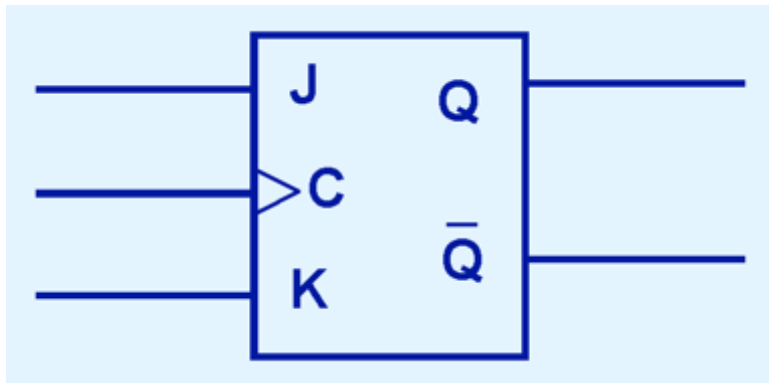
Memoria 64Kx16 usando memorias de 64Kx8



Flip-flop JK

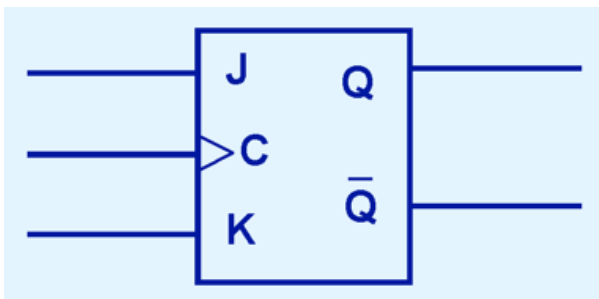
- Una forma alternativa de modificar el flip-flop RS para hacerlo estable es el flip-flop JK (en honor a Jack Kilby)
- La estabilidad se obtiene a partir de realimentar el circuito usando las salidas Q y \bar{Q} para garantizar que las entradas S y R son mutuamente inversas

J	K	$Q(t+1)$
0	0	$Q(t)$ (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\bar{Q}(t)$



Contador usando flip-flop JK

- La señal de enable activa todo el contador
- Al poner en 1 ambas entradas a cada ciclo de reloj se alterna el estado del flip-flop JK
- Cada vez que se tiene un 1 en la salida de un flip-flop JK eso hace que ambas entradas del siguiente sean puestas en 1



J	K	$Q(t+1)$
0	0	$Q(t)$ (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\bar{Q}(t)$

