Lógica Digital

Circuitos Secuenciales

Organización del Computador I
Departamento de Computación - FCEyN
UBA

5 de setiembre del 2017

Agenda

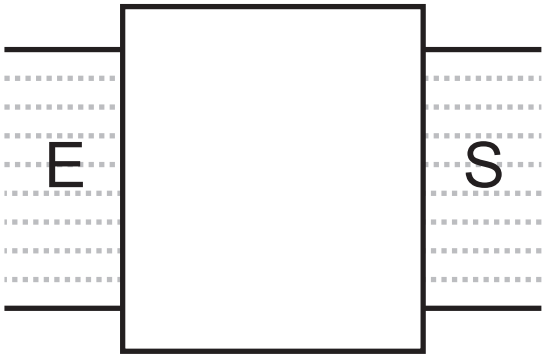
1. Repaso
2. Introducción
3. Flip-Flops
4. Ejercicios

¿Qué deberíamos saber hasta ahora?

- Operadores y funciones booleanas.
- Reducciones utilizando identidades.
- Dada una tabla de verdad poder escribir su función booleana.
- Graficar circuitos lógicos.
- Circuitos combinatorios.

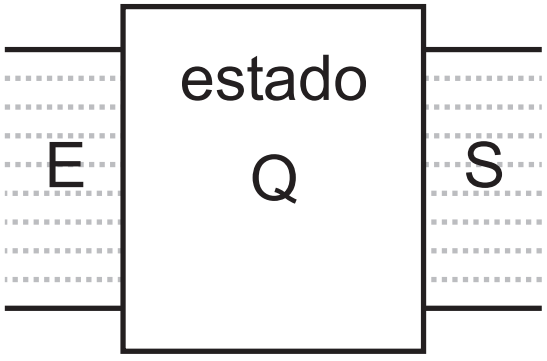
Introducción

Circuitos Combinacionales



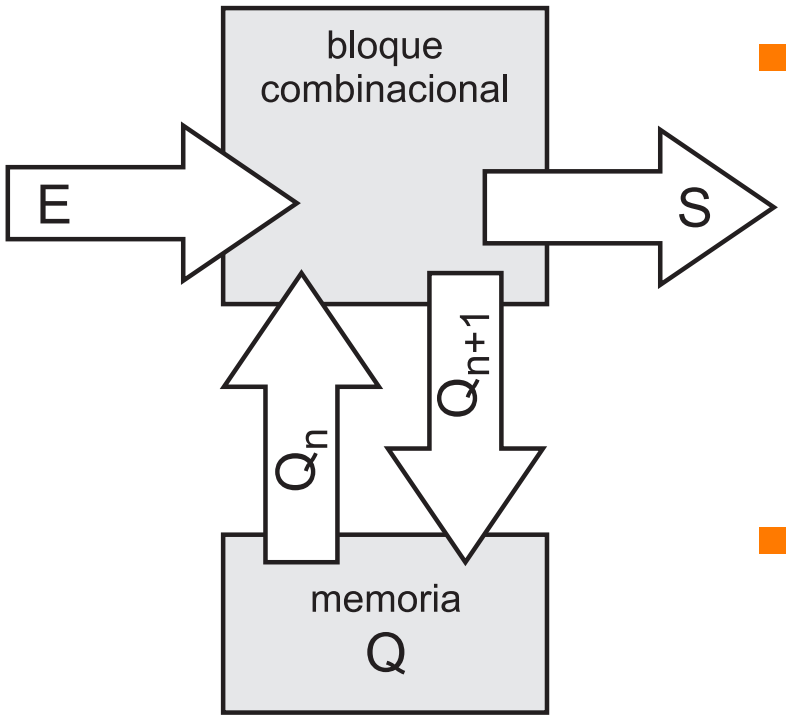
La salida esta determinada únicamente por la entrada del circuito

Circuitos Secuenciales



La salida esta determinada por la entrada y el *estado* del circuito

Circuitos Secuenciales



- Cualquier circuito secuencial, se puede separar en dos partes:
  1. un *bloque combinacional*
  2. un *bloque con memoria*
- La memoria almacena bits que determinan el estado del circuito

- Las entradas del circuito combinacional son las entradas (E) junto con las salidas de la memoria (Q_n)
- El bloque combinacional genera la salida del circuito (S) y el nuevo estado del mismo (Q_{n+1})

Flip-Flops

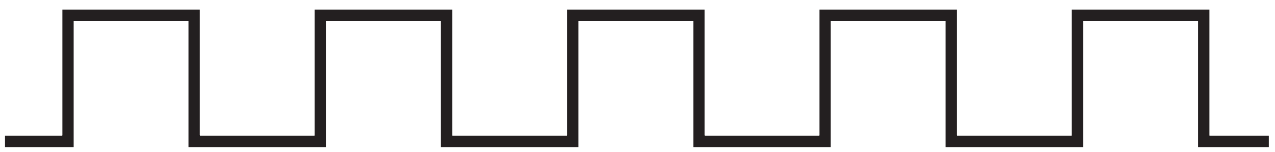
Introducción

- Un FF es un dispositivo capaz de almacenar un bit.
- Utilizan el principio de la retroalimentación.
- Esta característica es utilizada en Electrónica Digital para memorizar resultados.
- El paso de un estado a otro se realiza variando las entradas.
- Según el tipo de entradas pueden dividirse en:
  - Asincrónicos: Solo tienen entradas de control y pueden cambiar de estado en cualquier momento.
  - Sincrónicos: Además de las entradas de control posee una entrada de sincronismo o de reloj. El sistema solo puede cambiar en los instantes de sincronismo.

Relojes (Clocks)

Introducción

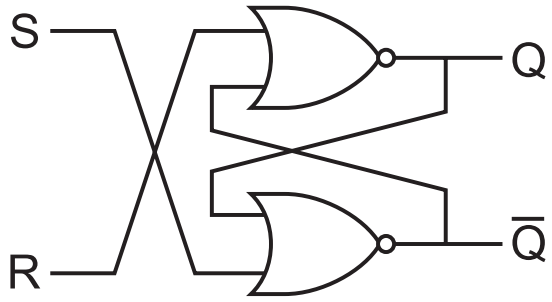
- Un reloj es un circuito que emite una serie de pulsaciones consecutivas con una frecuencia definida.
- Se denomina **Flanco** a la transición del nivel bajo al alto o del nivel alto al bajo.
- El periodo entre dos flancos ascendentes o descendentes se denomina tiempo de ciclo del reloj.
- Recordemos $\textit{Frecuencia} = \frac{1}{T}$

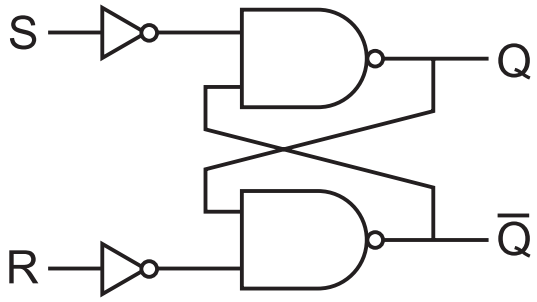


Asincrónicos: Flip-Flop RS (Reset & Set)

Características

- Tiene dos entradas S(et) y R(eset).
- Cuando ambas están en 0 se mantiene el valor de Q.
- Cuando ambas están en 1 el valor de Q se indefine.
- Si sólo S está en 1, el valor de Q cambia a 1.
- Si sólo R está en 1, el valor de Q cambia a 0.



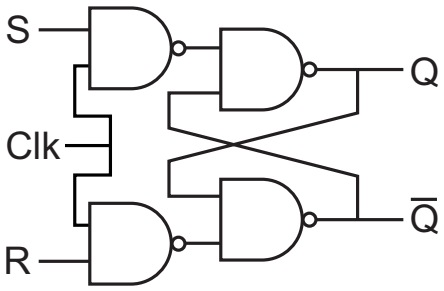


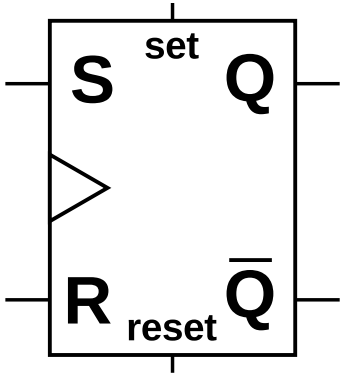
S	R	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	**X**

Sincrónicos: Flip-Flop RS (Reset & Set)

Características

- Es idéntico al Flip-Flop RS asincrónico, pero este sólo se actualiza su estado en el instante de sincronismo que marca el reloj.



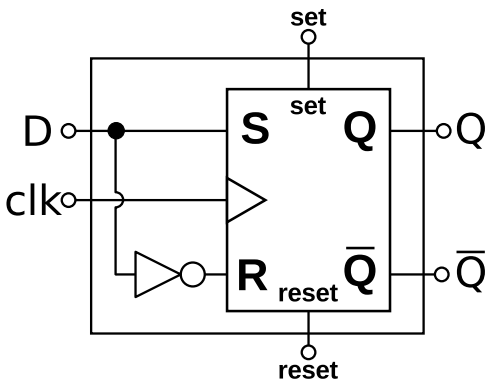


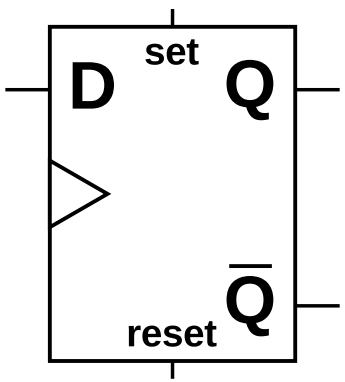
S	R	clk	Q_{t+1}
0	0	1	Q_t
0	1	1	0
1	0	1	1
1	1	1	X
_	_	0	Q_t

Sincrónicos: Flip-Flop D (Delay)

Características

- Posee solo una entrada D.
- La salida Q obtiene el valor de la entrada D cuando hay un pulso de reloj.



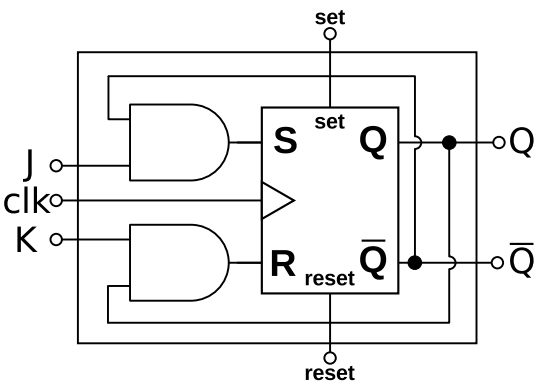


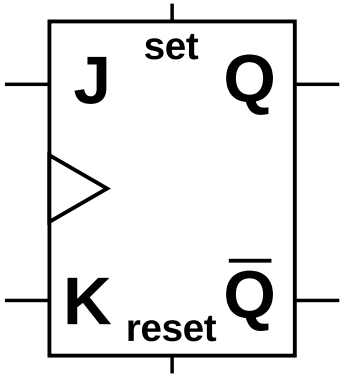
D	clk	Q_{t+1}
0	1	0
1	1	1
–	0	Q_t

Sincrónicos: Flip-Flop JK

Características

- Sus entradas son J y K en honor a Jack Kilby.
- Se considera como el FF universal ya que puede configurarse para obtener los demás FF.

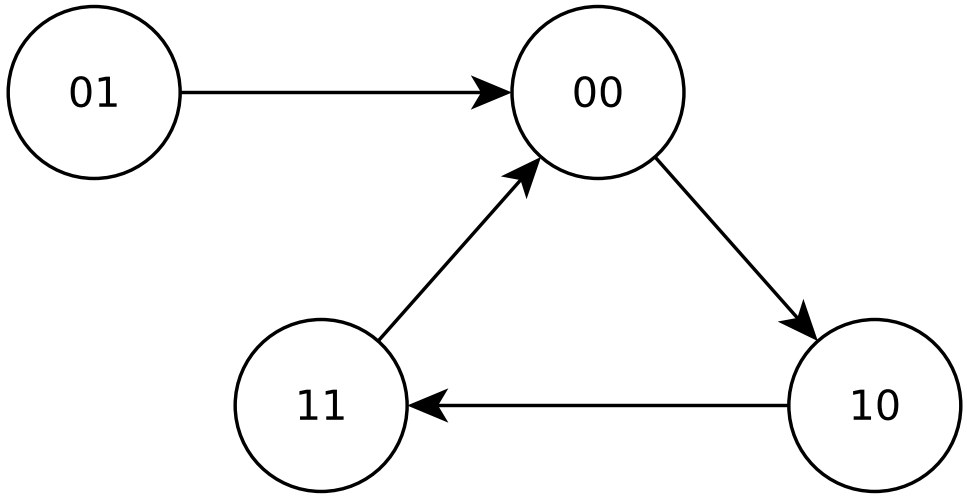




J	K	clk	$Q(t+1)$
0	0	1	Q_t
0	1	1	0
1	0	1	1
1	1	1	$\bar{Q}_t$
_	_	0	Q_t

Ejercicio 1

Implementar un registro contador de dos *bits* que siga los siguientes estados y que cada cambio se produzca al apretar un pulsador. Usando flip-flops D y compuertas básicas a elección. Nos piden además que el componente a desarrollar cuente con una entrada de Reset.



Solución - Ejercicio 1

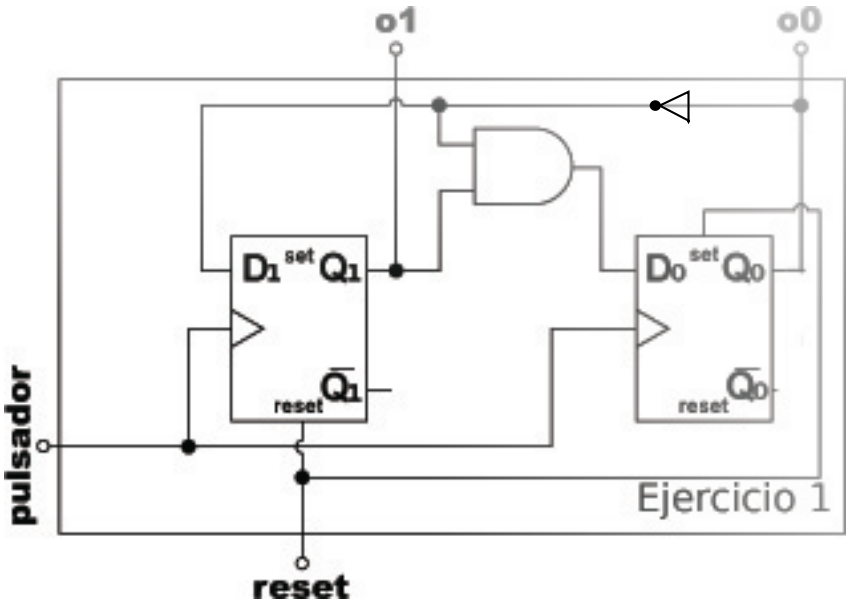
En este caso, dado un estado t definido por el valor de Q_1 y Q_0 podemos ver cuáles serán los próximos valores a almacenar:

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	1	0	0
0	0	1	0
1	0	1	1
1	1	0	0

¿qué valores deberían tener D_1 y D_0 para obtener los valores deseados en el tiempo t+1, es decir, de $Q_1(t+1)$ y $Q_0(t+1)$?
Usando que el flip-flop D define su próximo valor en referencia a lo que tiene en la entrada D, vemos que la suma de productos nos define los valores de D:

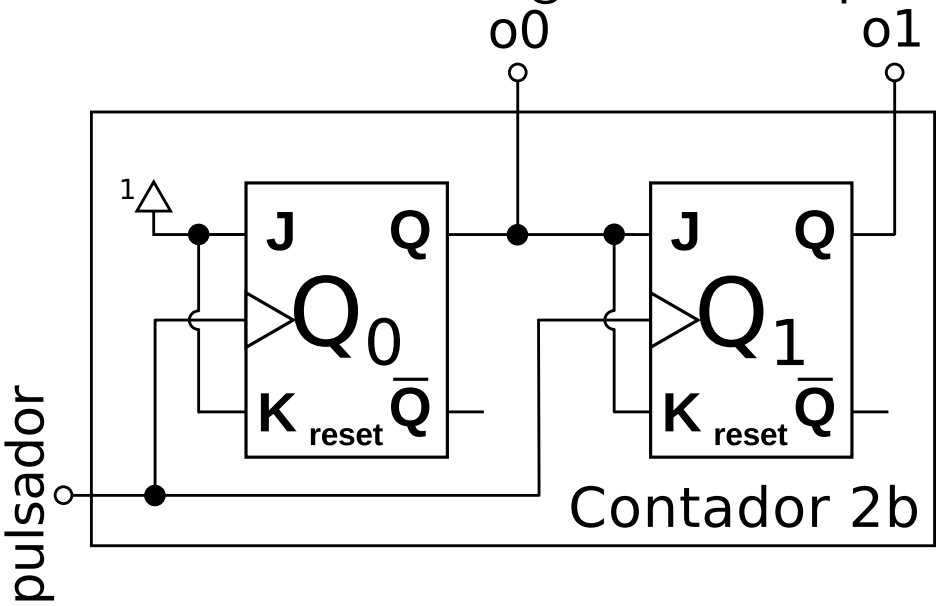
$$
\begin{aligned}
D_0 &= (Q_1.\bar{Q}_0) \\
D_1 &= (\bar{Q}_1.\bar{Q}_0) + (Q_1.\bar{Q}_0) \\
&= (\bar{Q}_1 + Q_1).\bar{Q}_0 \\
&= 1.\bar{Q}_0 \\
&= \bar{Q}_0
\end{aligned}
$$

Así se obtiene el siguiente circuito:



Ejercicio 2

Analizar los estados del siguiente componente:

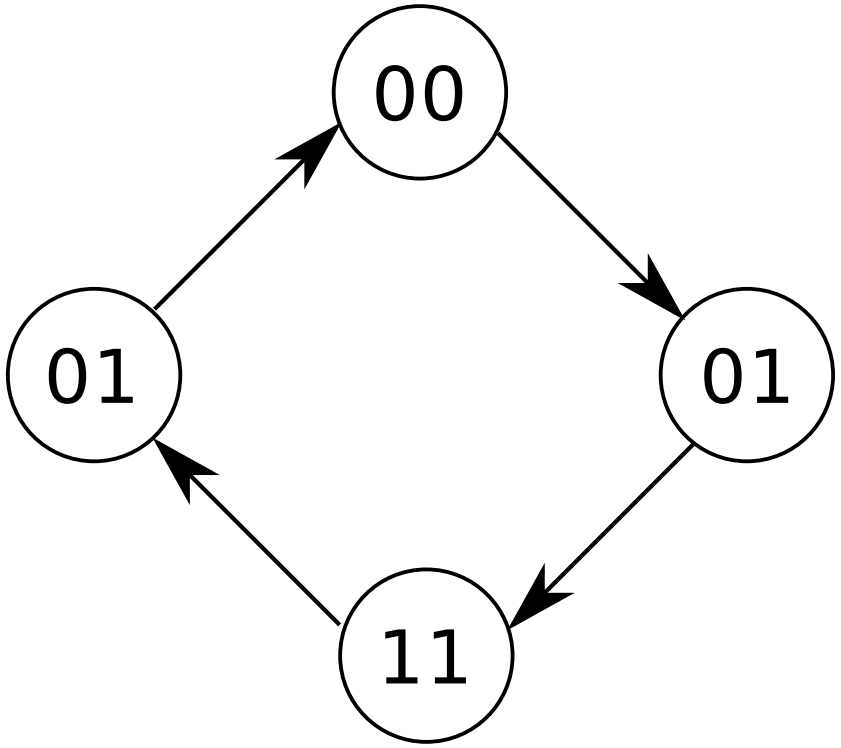


Solución:

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Ejercicio 3

Implementar un registro contador de dos *bits* que siga los siguientes estados y que cada cambio se produzca al apretar un pulsador.



Solución - Ejercicio 3

Realizando un análisis análogo al del ejercicio anterior se obtiene:

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	?	?
1	0	-	-
1	1	0	1

Lo cual no parece funcionar, ya que para el 01 no se puede determinar si es 11 ó 00 y para 10 no hay definido un próximo estado.

Solución - Ejercicio 3

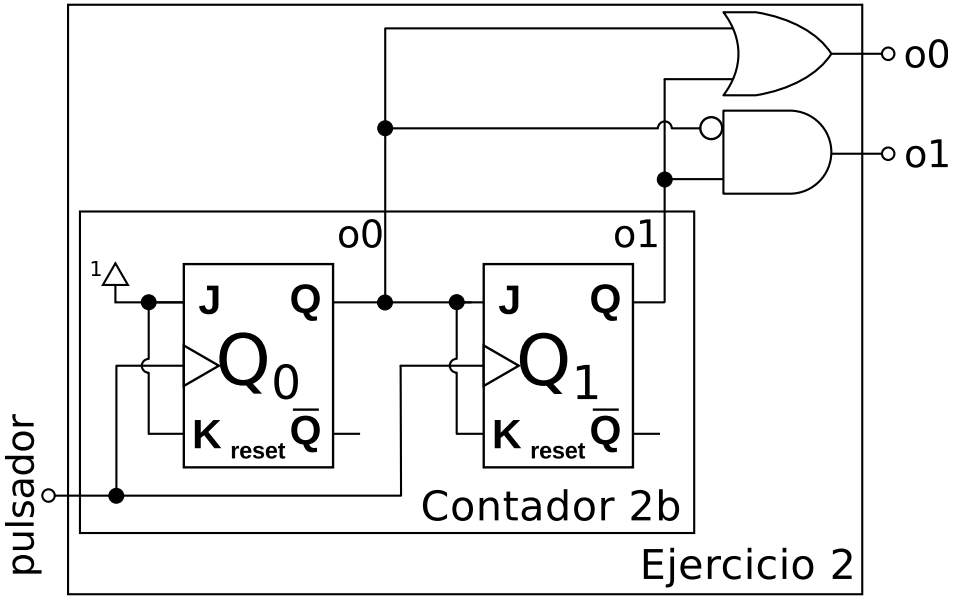
Q_1	Q_0	$\rightarrow$	o_1	o_0
0	0	$\rightarrow$	0	0
0	1	$\rightarrow$	0	1
1	0	$\rightarrow$	1	1
1	1	$\rightarrow$	0	1

Con lo cual podemos decir que:

$$o_0 = Q_1 + Q_0 \quad \text{por producto de sumas}$$

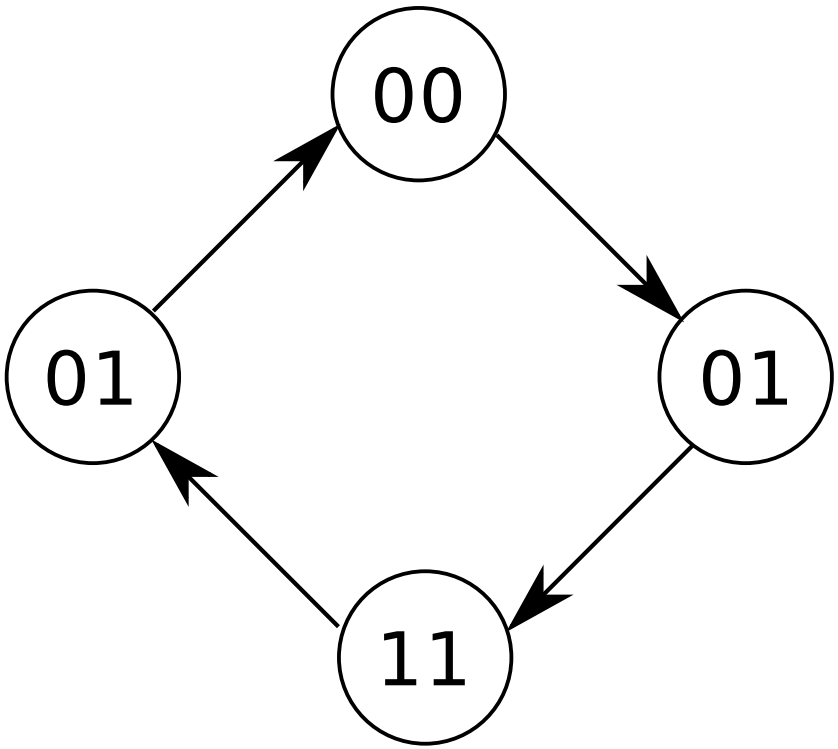
$$o_1 = Q_1 \,.\, \bar{Q_0} \quad \text{por suma de productos}$$

Solución - Ejercicio 3

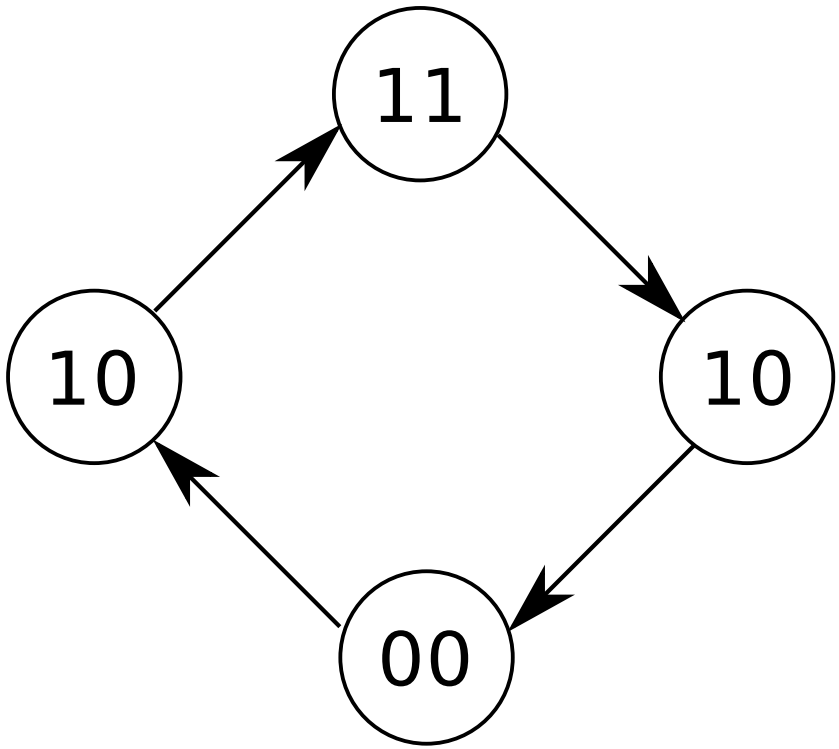


Ejercicio 3- bis

Implementar un registro contador de dos *bits* que siga los siguientes estados y que cada cambio se produzca al apretar un pulsador. Con el agregado de que tenga una entrada llamada NEG que genera los siguientes comportamientos:

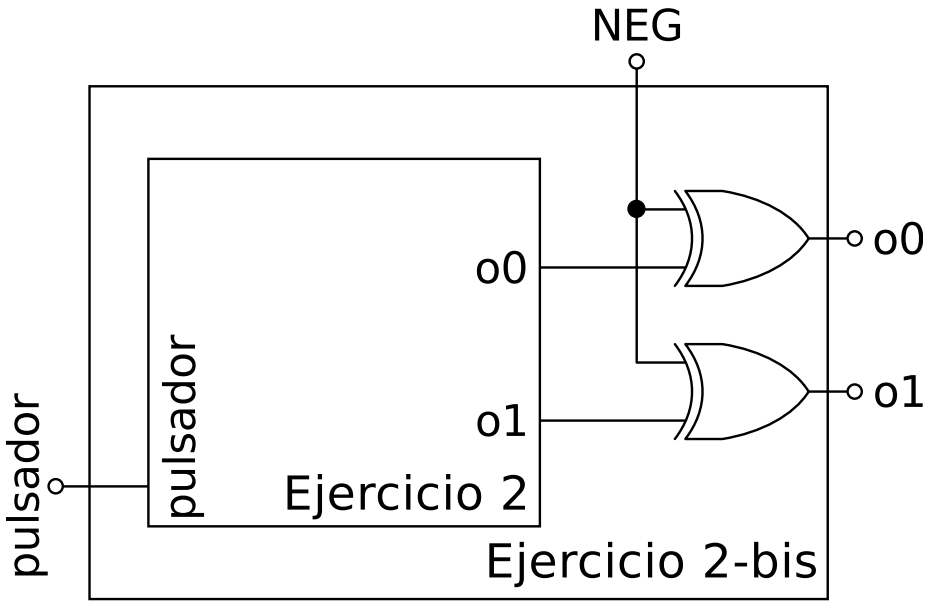


Si NEG vale 0



Si NEG vale 1

Solución - Ejercicio 3



Un cable[1]

- Un cable permite mandar una señal de un bit por él
- Un dispositivo/componente puede escribir un 0 ó un 1
- Si dos dispositivos intentan escribir al mismo tiempo un 0 y un 1, se asume que el valor es basura
- Para la materia, asumiremos que si dos dispositivos escriben a la vez en un cable producen un valor basura
- Un cable puede ser leído por más de un dispositivo a la vez
- Si ningún dispositivo está escribiendo un cable, entonces vale Hi-Z (alta impedancia) -no es ni 1 ni 0
- Si ningún dispositivo está escribiendo un cable, al leerlo se obtiene un valor basura
- Un cable no tiene memoria, no conserva ningún valor si nadie lo está escribiendo

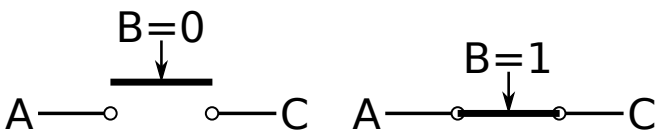
Tabla de verdad

(con dos dispositivos conectados de forma tal que pueden escribir en el cable)

$Disp_1$	$Disp_0$	Valor
0	0	???
0	1	???
0	Z	0
1	0	???
1	1	???
1	Z	1
Z	0	0
Z	1	1
Z	Z	Z

Componentes de Tres Estados

Noción Eléctrica



Símbolo

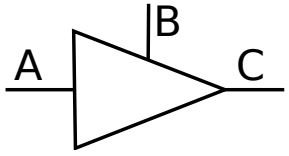


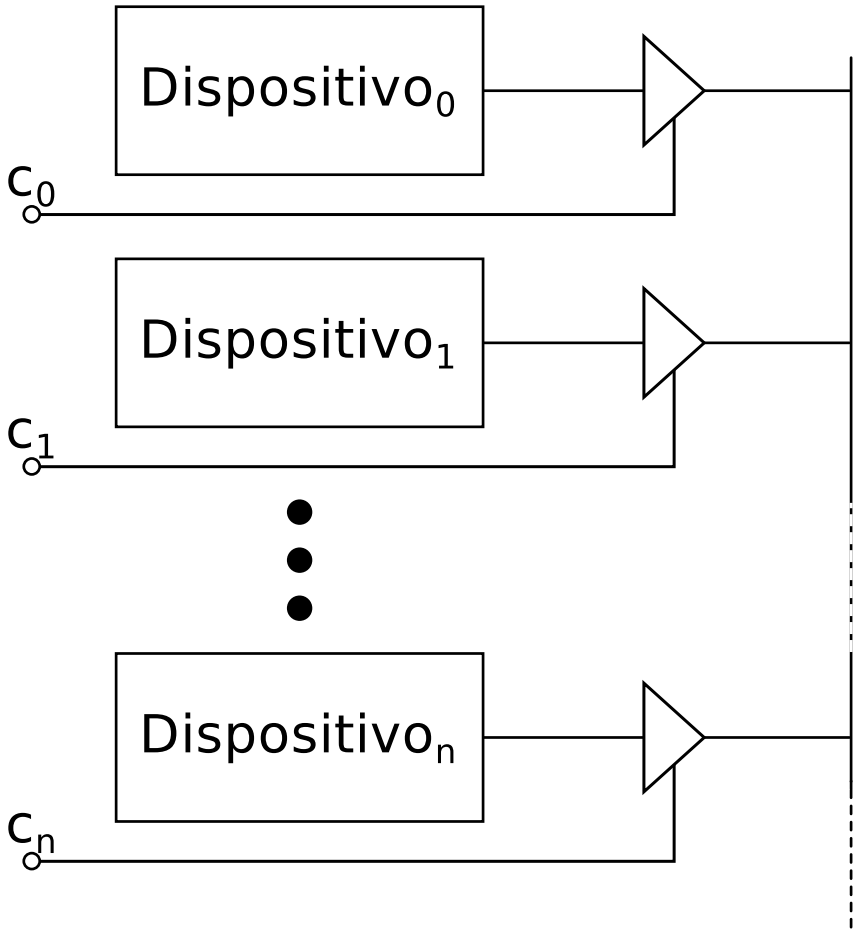
Tabla de Verdad

A	B	C
0	1	0
1	1	1
–	0	Hi-Z

Hi-Z significa "alta impedancia", es decir, que tiene una resistencia alta al pasaje de corriente. Como consecuencia de esto, podemos considerar al pin C como desconectado del circuito.

IMPORTANTE: Sólo deben ser usados a la salida de componentes para permitirles conectarse a un medio compartido (bus).

Componentes de Tres Estados



IMPORTANTE: Sólo deben ser usados a la salida de componentes para permitirles conectarse a un medio compartido (bus).

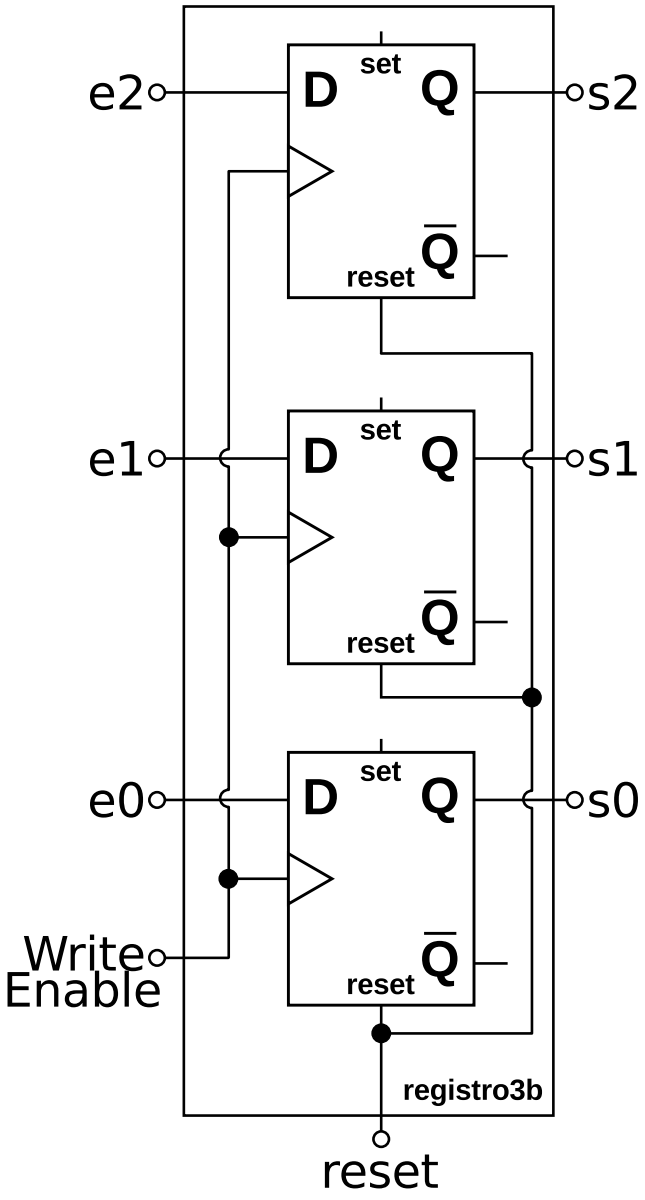
Ejercicio 4

a) Diseñar un registro de 3 *bits*. El mismo debe contar con 3 entradas $e_0, \ldots, e_2$ para ingresar el dato a almacenar, 3 salidas $s_0, \ldots, s_2$ para ver el dato almacenado y las señales de control RESET y WRITEENABLE.

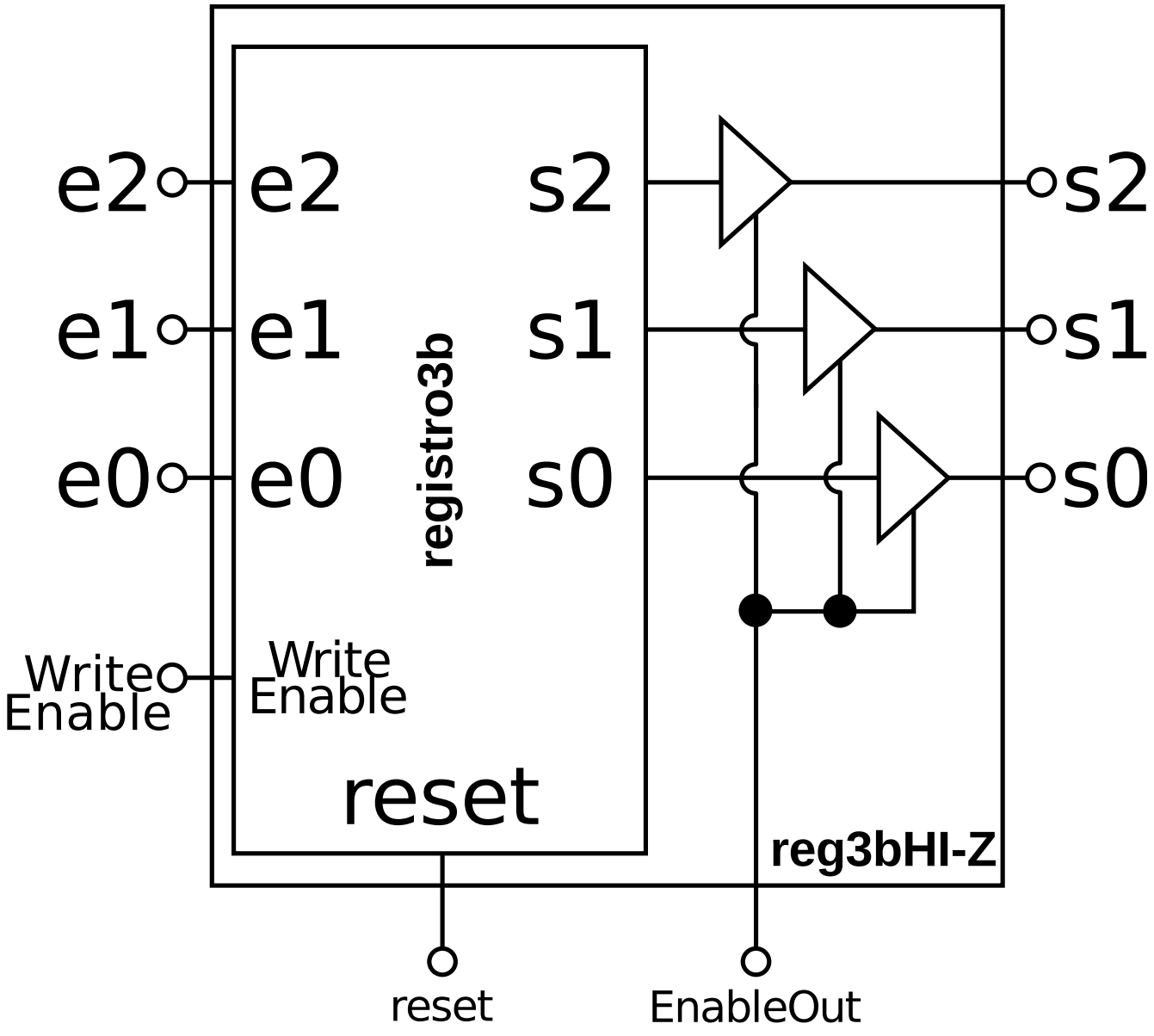
b) Modificar el diseño anterior agregándole componentes de 3 estados para que sólo cuando se active la señal de control ENABLEOUT muestre el dato almacenado.

c) Modificar nuevamente el diseño para que e_i y s_i estén conectadas entre sí al mismo tiempo teniendo en lugar de 3 entradas y 3 salidas, 3 entrada-salidas

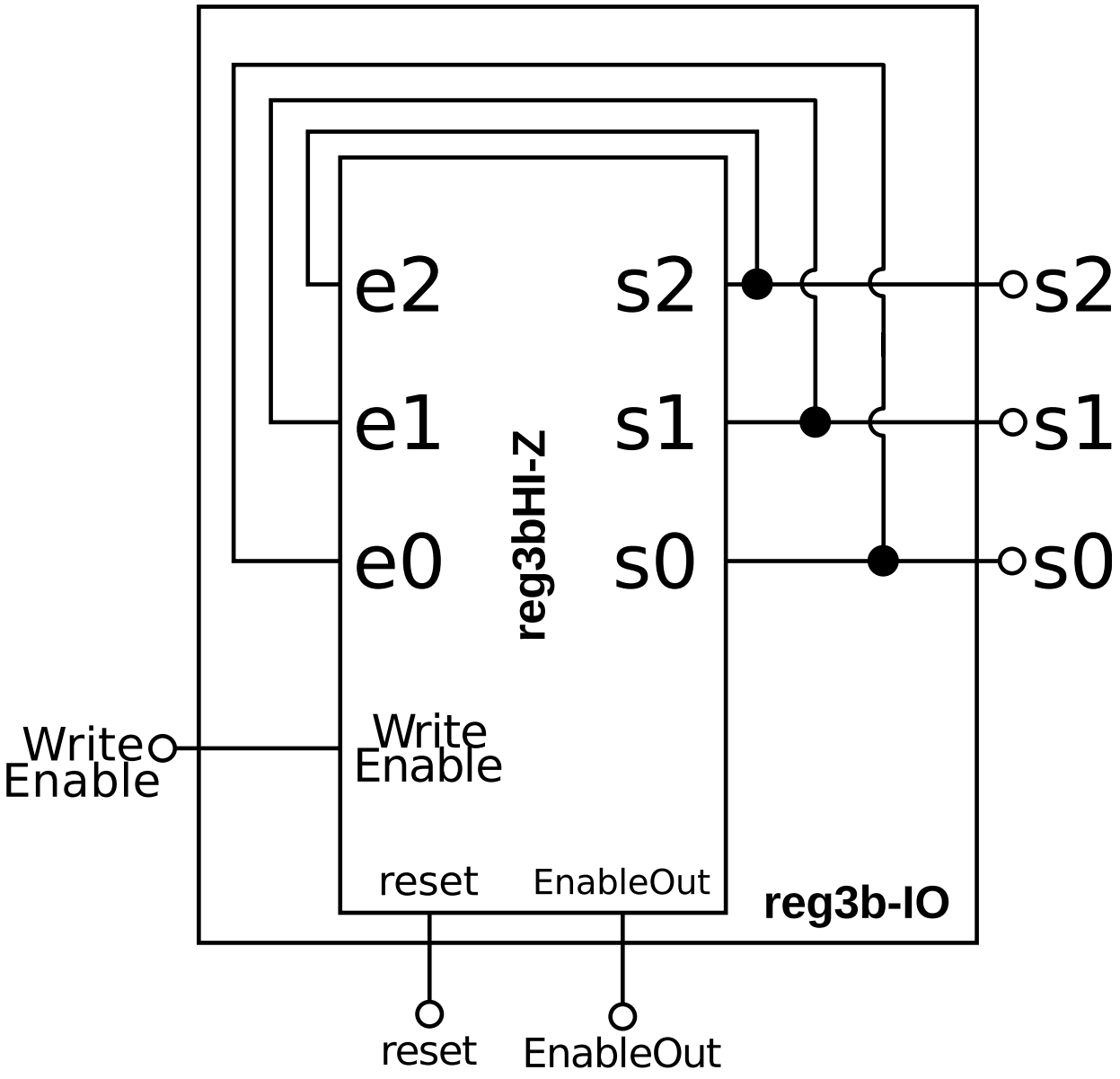
Solución - Ejercicio 4.a



Solución - Ejercicio 4.b



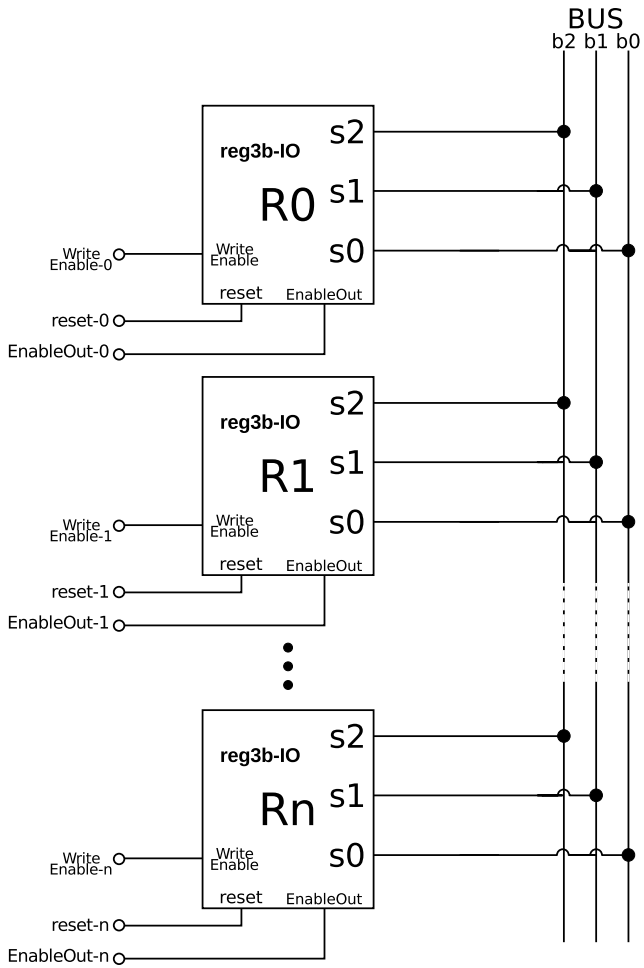
Solución - Ejercicio 4.c



Ejercicio 5

a) Realizar el esquema de interconexión de n registros como el diseñado

b) Dar una secuencia de valores de las señales de control para que se copie el dato del R1 al R0



Señales de control:

R0	R1	...	Rn
WriteEnable-0	WriteEnable-1	...	WriteEnable-n
reset-0	reset-1	...	reset-n
EnableOut-0	EnableOut-1	...	EnableOut-n

Inician todas las señales en 0. Luego se sigue la siguiente secuencia:

- EnableOut-1 $\leftarrow$ 1
- WriteEnable-0 $\leftarrow$ 1
- WriteEnable-0 $\leftarrow$ 0
- EnableOut-1 $\leftarrow$ 0

¿Cómo seguimos?

- Con lo que vimos hoy ya pueden terminar toda la práctica 2 (parte A y B)
- Pueden profundizar más sobre estos temas en *The Essential of Computer Organization (L. Null) - Capítulo 3*