

# Lógica Digital - Secuenciales

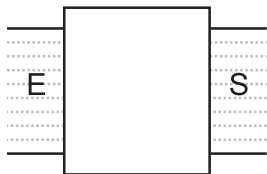
## Organización del Computador I

Departamento de Computación, FCEyN, Universidad de Buenos Aires

22 de septiembre de 2020

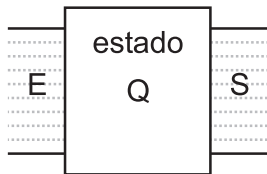
# Introducción

## Circuitos Combinacionales



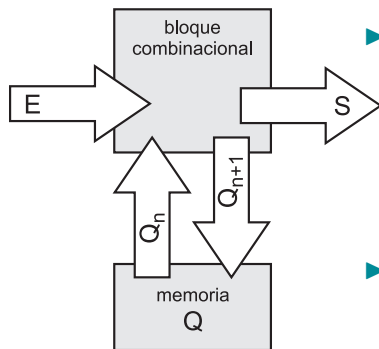
La salida esta determinada únicamente por la entrada del circuito

## Circuitos Secuenciales



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

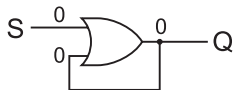
# Circuitos Secuenciales



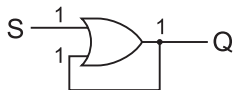
- ▶ Cualquier circuito secuencial, se puede separar en dos partes:
  1. un *bloque combinacional*
  2. un *bloque con memoria*
- ▶ La memoria almacena *bits* que determinan el estado del circuito
- ▶ Las entradas del circuito combinacional son las entradas ( $E$ ) junto con las salidas de la memoria ( $Q_n$ )
- ▶ El bloque combinacional genera la salida del circuito ( $S$ ) y el nuevo estado del mismo ( $Q_{n+1}$ )

## ¿Cómo almacenar un bit?

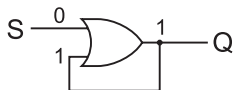
Proponemos el siguiente circuito:



- Supongamos que inicialmente  $S = 0$  y  $Q = 0$

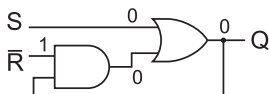


- Si cambiamos el valor  $S = 1$ , luego que el circuito se *estabiliza*, la salida valdrá  $Q = 1$

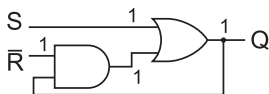


- Si ahora cambiamos  $S = 0$ , obtendremos como resultado que el valor de salida  $Q$  continuará en 1

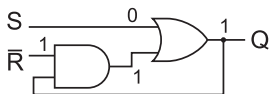
## ¿Cómo construir un *biestable*?



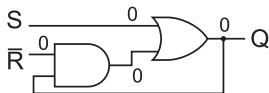
- Supongamos que inicialmente  $S = 0$ ,  $\bar{R} = 1$  y  $Q = 0$



- Si  $S = 1$  y  $\bar{R} = 1$  entonces la salida  $Q$  valdrá 1



- Si luego  $S = 0$  y  $\bar{R} = 1$  entonces la salida  $Q$  continuará en 1

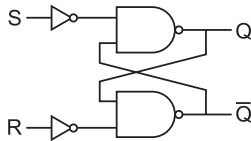
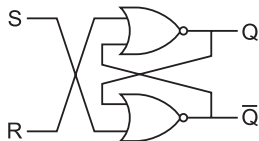


- Ahora si  $S = 0$  y  $\bar{R} = 0$  entonces la salida  $Q$  cambiará a 0

# 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.

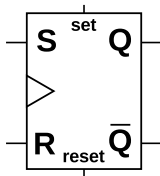
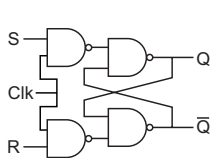


S	R	$Q_{t+1}$
0	0	$Q_t$
0	1	0
1	0	1
1	1	X

# 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.

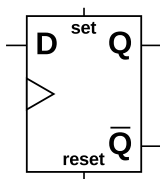
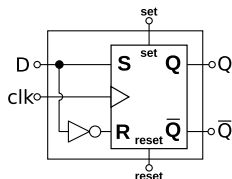


S	R	clk	$Q_{t+1}$
0	0	1	$Q_t$
0	1	1	0
1	0	1	1
1	1	1	X
-	-	0	$Q_t$

# 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.



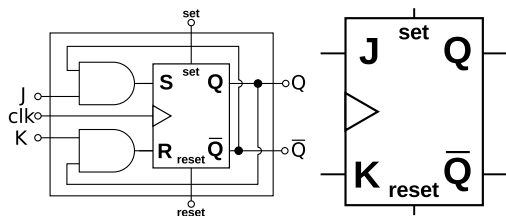
D	clk	$Q_{t+1}$
0	1	0
1	1	1
-	0	$Q_t$



# Sincrónicos: Flip-Flop JK

## Características

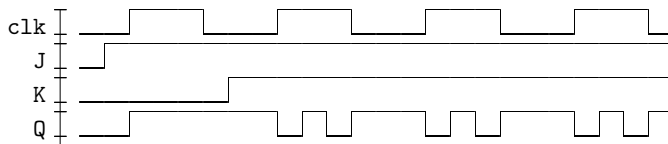
- ▶ Sus entradas son J y K en honor a Jack Kilby.
- ▶ Similar al RS, pero con el estado  $J=1, K=1$  definido (Toggle)



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	$Q_t$

# Sincrónicos: Flip-Flop JK

- **Pregunta:** ¿Qué pasa realmente en el estado del flip-flop JK en el caso que ambos sean 1?
- Supongamos activación por nivel alto.



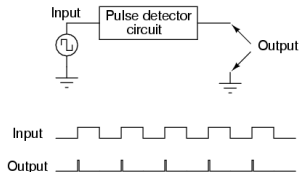
- ¿Qué pasó?
- Race around condition. ¿Cómo lo solucionamos?<sup>1</sup>

---

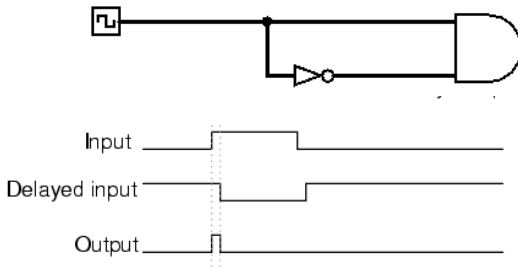
<sup>1</sup>Race around visto desde un osciloscopio

# Sincrónicos: Flip-Flop JK

Activación por flanco ascendente:



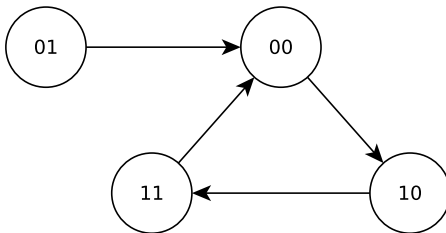
Aprovechando el *delay* de propagación, proponemos la siguiente implementación para el **pulse detector**



# Ejercicio 1

Implementar un registro 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

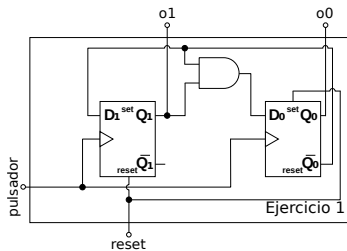
En este caso, dado un estado  $t$  definido por el valor de  $Q_1$  y  $Q_0$  podemos ver cuáles serán los próximos valores a almacenar:

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

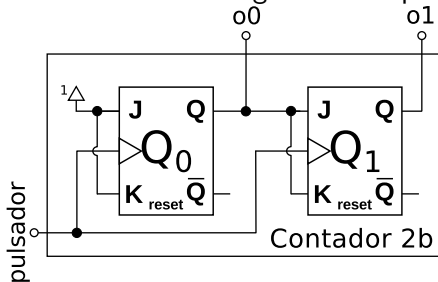
$$\begin{aligned}D_0 &= (Q_1 \cdot \bar{Q}_0) \\D_1 &= (\bar{Q}_1 \cdot \bar{Q}_0) + (Q_1 \cdot \bar{Q}_0) \\&= (\bar{Q}_1 + Q_1) \cdot \bar{Q}_0 \\&= 1 \cdot \bar{Q}_0 \\&= \bar{Q}_0\end{aligned}$$

Así se obtiene el siguiente circuito:



## Ejercicio 2

Analizar los estados del siguiente componente:

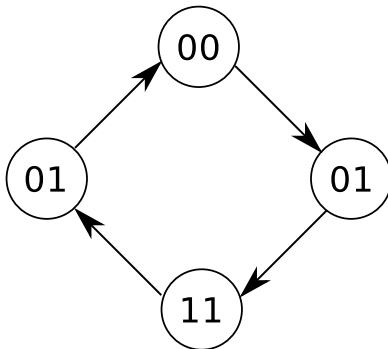


Solución:

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

## Ejercicio 3

Implementar un registro 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

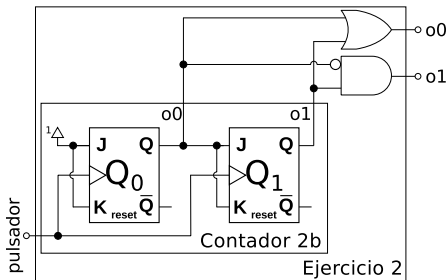
$Q_1$	$Q_0 \rightarrow o_1$	$o_0$
0	$0 \rightarrow 0$	0
0	$1 \rightarrow 0$	1
1	$0 \rightarrow 1$	1
1	$1 \rightarrow 0$	1

Con lo cual podemos decir que:

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

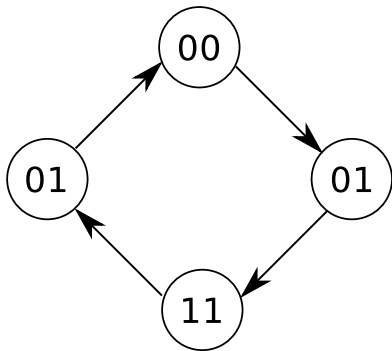
$$o_1 = Q_1 \cdot \bar{Q}_0 \quad \text{por suma de productos}$$

## Solución - Ejercicio 3

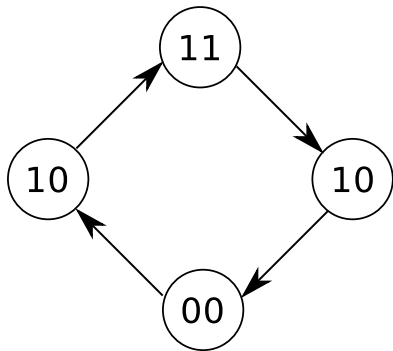


## Ejercicio 3- bis

Implementar un registro 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:

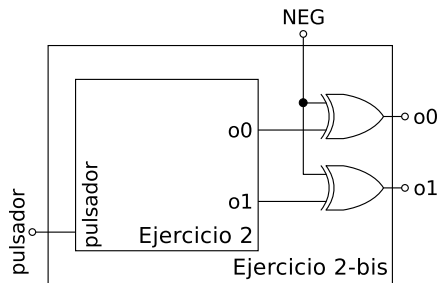


Si NEG vale 0



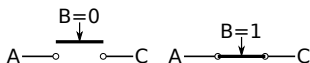
Si NEG vale 1

## Solución - Ejercicio 3

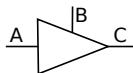


# Componentes de Tres Estados

## Noción Eléctrica



## Símbolo



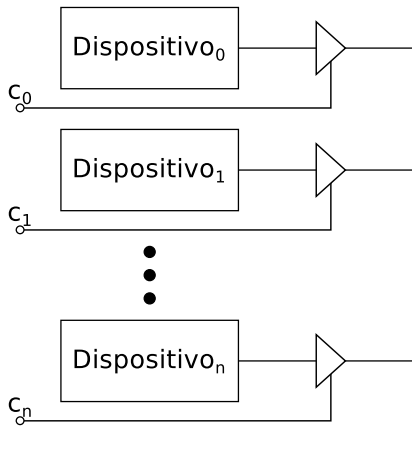
## Tabla de Verdad

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



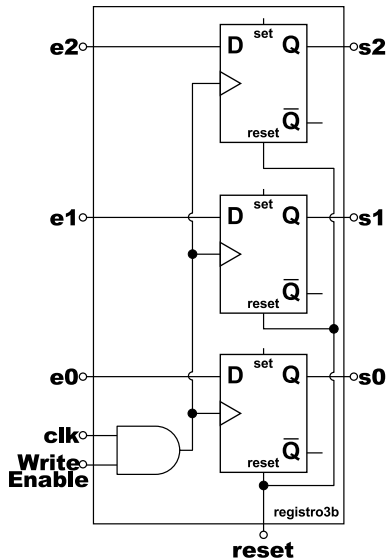
**IMPORTANTE:** Sólo deben ser usados a la salida de componentes para permitirles conectarse a un medio compartido (bus).

## Ejercicio 4

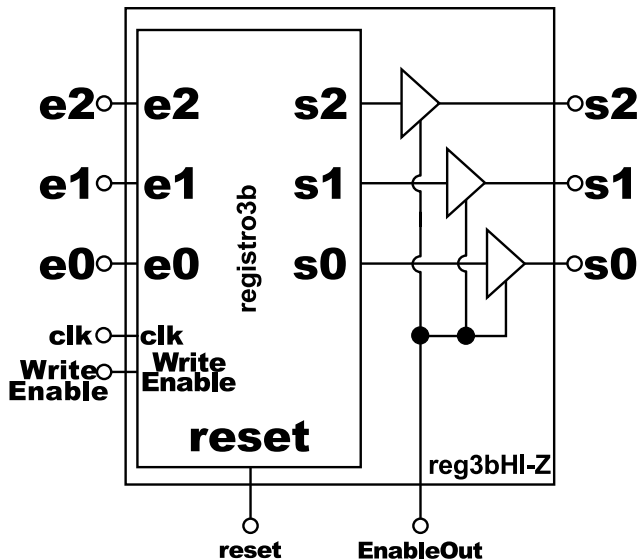
- a) Diseñar un registro de 3 *bits*. El mismo debe contar con 3 entradas  $e_0, \dots, e_2$  para ingresar el dato a almacenar, 3 salidas  $s_0, \dots, s_2$  para ver el dato almacenado y las señales de control CLK, RESET y WRITEENABLE.
- b) Modificar el diseño anterior agregándole componentes de 3 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



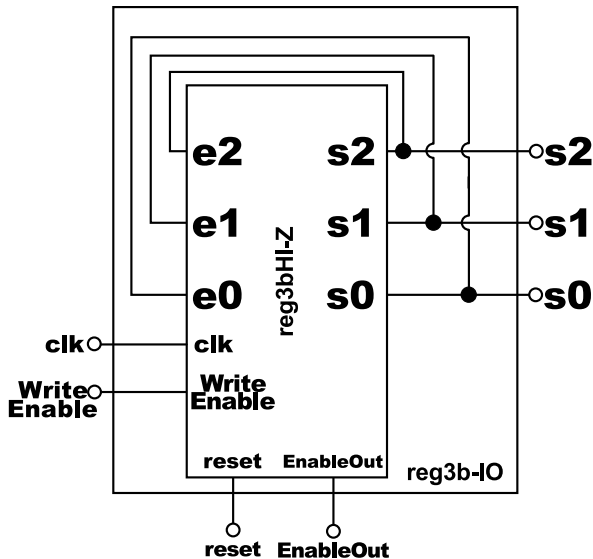
## Solución - Ejercicio 4.a



## Solución - Ejercicio 4.b

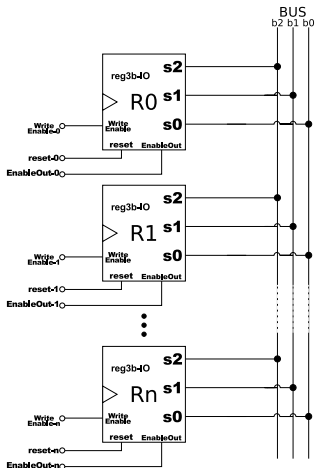


## Solución - Ejercicio 4.c



## Ejercicio 5

- Realizar el esquema de interconexión de  $n$  registros como el diseño
- 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
- ▶ ...clk....
- ▶ 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
- ▶ El próximo martes (29 de septiembre) vamos a hacer el segundo taller obligatorio: **Secuenciales**.

# ¿Preguntas?