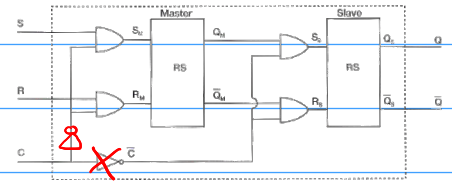


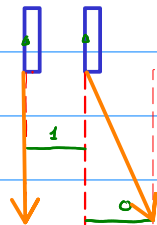
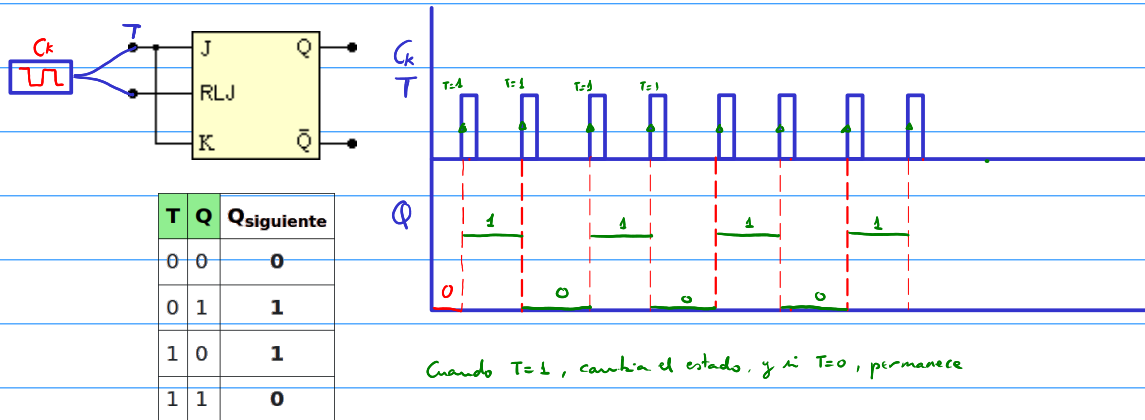
Ejercicio 1

Para construir un biestable RS Sincrono, tipo Master-Slave activo por flanco de subida solo tenemos que invertir el reloj.



Ejercicio 2

Demuestra que un biestable T divide la frecuencia en 2 de una señal de reloj.



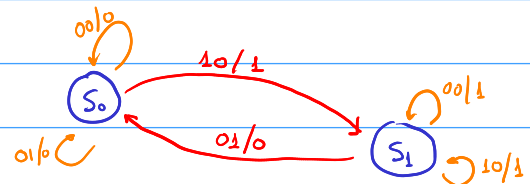
Como se puede observar, la señal DUPLICA su período, lo que significa que divide en dos su frecuencia

Ejercicio 3

- Registro de almacenamiento:
[<https://circuitverse.org/simulator/edit/registrod>]
- Registro de desplazamiento:
[<https://circuitverse.org/simulator/edit/regitroddes>]
- Contador asíncrono:
[https://circuitverse.org/simulator/edit/contador_asincrono-714775e9-8458-41f1-b528-28e86263fc64]
- Contador asíncrono con BCD 7 segmentos:
[https://circuitverse.org/simulator/edit/contador_asincrono_2]

Ejercicio 4 Para resolver este ejercicio usaremos una descripción gráfica o representación de estados llamada "máquina de Mealy". Describo los estados como círculos, y las relaciones entre ellos con flechas. Cada relación, o evolución de la máquina de un estado a otro, se produce con una combinación de entrada y conlleva la activación de las salidas.

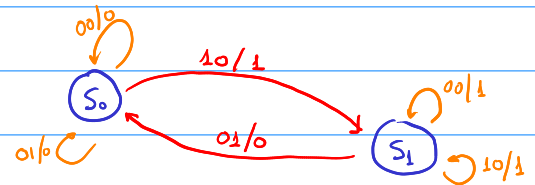
En este ejercicio tenemos dos estados:
bombilla apagada (S_0) y bombilla encendida (S_1).



- a) Si la bombilla está apagada, y pulso A ($AB=10$), se enciende. Pasa al otro estado, y la salida que produce es '1'.
- b) Si está apagada, no hacer nada o pulsar B me dejan en el estado S_0 y no produce salida.
- c) $AB=11$ está prohibida en el enunciado ($AB=00$ $AB=01$)
- d) Si está encendida, y pulso B ($AB=01$), se apaga.
- e) Si está encendida, pulsar A o no hacer nada, la deja encendida.

Apartado A

Luego el diagrama de fases (máq. Mealy) es:



B) Obtener tabla de fases

	AB			
	00	01	11	10
S_0	S_0	S_0	—	S_1
S_1	S_1	S_0	—	S_1

c) Codifico los estados

Así hay dos: $S_0 = 0$
 Así necesito 1 bit $S_1 = 1$

D) tablas de verdad (mapa karnaugh) de los estados y de las salidas.

Q_{t+1}

	AB			
	00	01	11	10
Q_t 0	0	0	X	1
Q_t 1	1	0	X	1

$$Q_{t+1} = A + q \cdot \bar{B}$$

S

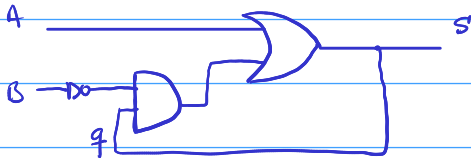
	AB			
	00	01	11	10
Q_t 0	0	0	X	1
Q_t 1	1	0	X	1

$$S = A + q \cdot \bar{B}$$

Apartado 1, implementación en puertas lógicas.

$$q^k = A + q \cdot \bar{B}$$

$$S = q^k$$



<https://circuitverse.org/simulator/edit/pulsadores>

Apartado 2. Como los estados tienen 1 bit, solo necesito 1 biestable JK

A) Construyo una tabla de la verdad, con AB, Q_t y Q_{t+1} , y voy deduciendo las entradas JK, según como lo hace un flip-flop de este tipo.

CLK	J	K	Q_n	Q_{n+1}	Q_{n+1}
1	0	0	0	0	Q_n
1	0	0	1	1	Q_n
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	Q_n'
1	1	1	1	0	Q_n'

Q_t	A	B	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

J	K
0	X
0	X
1	X
X	X
X	0
X	1
X	0
X	X

Por ejemplo:

$$AB=00 \quad Q_t=0 \rightarrow Q_{t+1}=0$$

JK debe de:

1) O dejar el estado como está $JK=00$

2) O forzarlo a cero RESET $JK=01$

Luego $J=0$ y K da igual (X)

Por ejemplo: $AB=10 \quad Q_t=0 \rightarrow Q_{t+1}=1$

JK debe

1) Cambiar el estado $JK=11$

2) Hacer un Set $JK=10$

$J=1 \quad K=X$

Por ejemplo: $AB=00 \quad Q_t=1 \rightarrow Q_{t+1}=1$

1) dejar el estado igual $JK=00$

2) hacer un set $JK=10$

$J=X \quad K=0$

B) Hacemos Karnaugh para J y K

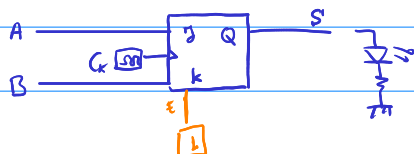
Q	AB			
	00	01	11	10
0	0	0	X	1
1	X	X	X	X

$$J = A$$

Q	AB			
	00	01	11	10
0	X	X	X	X
1	0	1	X	0

$$K = B$$

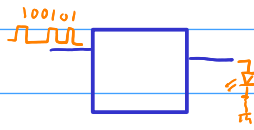
c) Dibujamos el circuito.



<https://circuitverse.org/simulator/edit/pulsadores2>

En el simulador tengo una señal de "Enable" que tengo que poner a 1 para que funcione.

Ejercicio 5

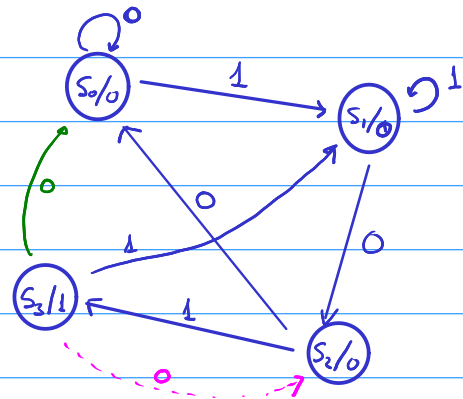


Trabajaremos con una representación denominada "máquina de Moore" en el que cada estado tiene asociado una salida. En la entrada voy recibiendo una serie de datos, los voy leyendo y quiero detectar si me ha recibido la cadena "101". Por ejemplo:

1001010010101
 detecta detecta
 x no solapa. Empieza de 0

Como no solapa, el "0" vuelve al estado S_0

Si solapara, volviera al S_2



B) Table de fases con solape

	X	
	0	1
S_0	S_0	S_1
S_1	S_2	S_1
S_2	S_0	S_3
S_3	S_0	S_1

c) Codificar: Tengo 4 estados, luego necesito 2 bits para codificarlos.

$S_0 = 00$ $S_1 = 11$ $S_2 = 01$ $S_3 = 10$ $S = q_1 q_2$

Hay métodos para optimizar una codificación. Nosotros intentaremos que haga las mínimas transiciones $00 \rightarrow 11$
 $11 \rightarrow 00$

D) Mapa de Karnaugh

	S_0	S_2	S_1	S_3
	00	01	11	10
X 0	S_0 00	S_0 00	S_2 01	S_0 00
X 1	S_1 11	S_3 10	S_1 11	S_1 11

$S_3 = q_1 q_2' = \text{salida.}$

	Q_1			
$q_1 q_2$	00	01	11	10
X 0	0	0	0	0
X 1	1	1	1	1

$Q_1 = x$

	Q_2			
$q_1 q_2$	00	01	11	10
X 0	0	0	1	0
X 1	1	0	1	1

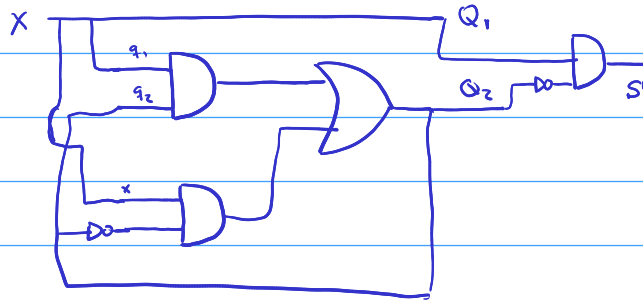
$Q_2 = q_1 q_2 + x q_2'$

Luego un posible circuito es

$$Q_1 = X$$

$$Q_2 = q_1 q_2 + x q_2'$$

$$S = q_1 q_2'$$



Pero normalmente se lleva a cabo con biestables JK.

Lo haremos con biestables JK.

CLK	J	K	Qn	Qn+1	Qn+1'
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	0	1

1er flip-flop

X	q ₁	q ₂	Q ₁	J ₁	K ₁
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	0	X	1
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	1	X
1	1	0	1	X	0
1	1	1	1	X	0

Transiciones

$$\begin{aligned} 0 \rightarrow 0 & \quad JK = 0X \\ 0 \rightarrow 1 & \quad JK = 1X \\ 1 \rightarrow 0 & \quad JK = X1 \\ 1 \rightarrow 1 & \quad JK = X0 \end{aligned}$$

2º flip-flop

X	q ₁	q ₂	Q ₂	J ₂	K ₂
0	0	0	0	0	X
0	0	1	0	X	1
0	1	0	0	0	X
0	1	1	1	X	0
1	0	0	1	1	X
1	0	1	0	X	1
1	1	0	1	1	X
1	1	1	1	X	0

q ₁ q ₂	00	01	11	10
0	0	0	0	0
1	1	1	1	1

q ₁ q ₂	00	01	11	10
0	0	0	1	0
1	1	0	1	1

Obtenemos funciones para J₁, K₁, J₂, K₂

X	q ₁ q ₂	00	01	11	10
0	0	0	0	X	X
1	1	1	1	X	X

$J_1 = X$

	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$K_1 = X'$

X	q ₁ q ₂	00	01	11	10
0	0	0	X	X	0
1	1	1	X	X	1

$J_2 = X$

	00	01	11	10
0	X	1	0	X
1	X	1	0	X

$k_2 = q_1'$

$$S = q_1 q_2'$$

