DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

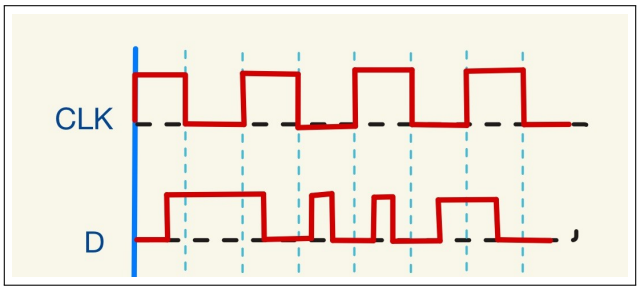
IIC2343 - Arquitectura de Computadores (I/2026)

Ayudantía 3

Ayudantes: Alberto Maturana (alberto.maturana@uc.cl), Mario Rojas (mario.denzel@estudiante.uc.cl)

Pregunta 1: Flip-Flops y Latches

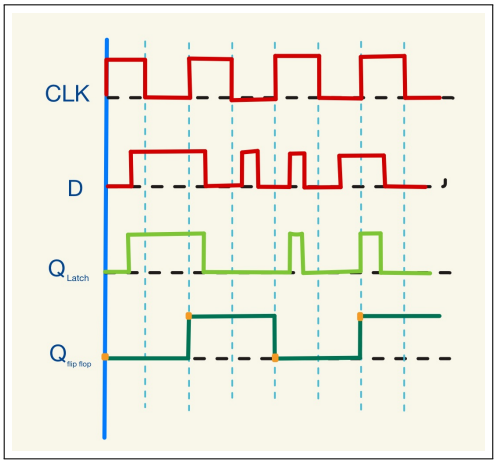
a) Dadas las siguientes señales del Clock y entrada D, determine los valores de salida para un Flip-Flop D y un latch D.



b) Luego, utilizando Flip-Flop tipo D, diseñe un circuito que cuente secuencialmente 2 bits, incrementándose en cada señal de subida.

Solución:

a)

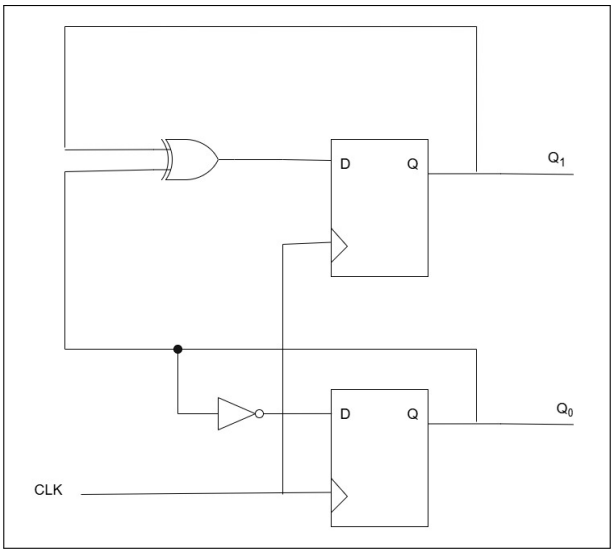


b) Para realizar esto, podemos analizar la tabla de verdad del circuito en los flancos de subida que pretendemos construir para determinar como conectar los Flip-Flop que usemos.

Q_1^t	Q_0^t	D_1	D_0	Q_1^{t+1}	Q_0^{t+1}
0	0	0	1	0	1
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0

Podemos notar que el bit menos significativo va cambiando en cada flanco de subida, por lo tanto una compuerta NOT es ideal para el comportamiento. Por otra parte, el bit más significativo sigue el comportamiento de una compuerta XOR con las entradas siendo ambos bits.

Usando esta información, podemos construir el siguiente circuito:



Pregunta 2: Diseño de circuitos a partir de una situación

Se desea diseñar un sistema digital que permita simular presencia en una vivienda activando cada 4 horas de manera alternada dos luces:

- L1: luz de la cocina
- L2: luz del comedor

Funcionamiento del sistema:

- El sistema puede estar encendido o apagado. Este apagado/encendido del sistema se llevará a cabo mediante una señal T, la cual indicará el estado del sistema ($T = 0$ apagado, $T = 1$ encendido).
- Cuando el sistema esté encendido, este debe **alternar el estado de las luces**, es decir, si por ejemplo la luz de la cocina está encendida y la del comedor apagada, en el siguiente flanco de subida la primera debe estar apagada y la segunda encendida. Nunca se dará la situación en que ambas estén encendidas o apagadas.
- Cuando el sistema esté apagado, este debe **mantener el estado actual de las luces**.

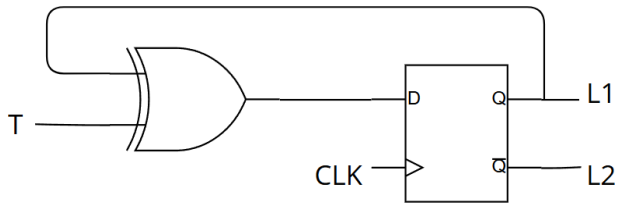
Importante: Siempre habrá una luz encendida y la otra apagada. No pueden estar ambas luces encendidas o apagadas simultáneamente.

Solución: Primero, construimos una tabla que represente los diferentes estados de nuestro sistema:

Q^t	T	Q^{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

- La entrada Q^t representa el estado actual de las luces, donde
  - $Q^t = 0$ indica que la luz de la cocina está encendida.
  - $Q^t = 1$ indica que la luz del comedor está encendida.
- La entrada T indica si el sistema está encendido ($T = 1$) o apagado ($T = 0$).
- La entrada Q^{t+1} representa el siguiente estado de las luces.
  - Si el sistema está encendido ($T = 1$) tenemos que $Q^{t+1} = \neg Q^t$, es decir, se alterna el estado anterior.
  - Si el sistema está apagado ($T = 0$) tenemos que $Q^{t+1} = Q^t$, es decir, se mantiene el estado anterior.

Conforme a lo anterior, podemos notar que el comportamiento de Q^{t+1} se puede modelar como $Q^{t+1} = Q^t \oplus T$ \$. Luego, haciendo uso de circuitos lógicos vistos en clases se llega al siguiente diagrama:



Solución: *Esta es otra forma de resolver el mismo problema. Ahora lo haremos con dos flip-flops*

- Q_0: representa la luz de la cocina (L1)
- Q_1: representa la luz del comedor (L2)

Dado que siempre debe haber exactamente una luz encendida, los estados válidos del sistema son:

$$Q_1Q_0 = 01 \quad \text{y} \quad 10$$

Funcionamiento

- Si $T = 1$ (sistema encendido), el sistema alterna las luces:

$$01 \leftrightarrow 10$$

  Por lo tanto:

$$Q_0^{t+1} = Q_1, \quad Q_1^{t+1} = Q_0$$

- Si $T = 0$ (sistema apagado), el sistema mantiene el estado:

$$Q_0^{t+1} = Q_0, \quad Q_1^{t+1} = Q_1$$

Tabla de verdad

T	Q_1	Q_0	D_1	D_0	Q_1^{t+1}	Q_0^{t+1}
0	0	1	0	1	0	1
0	1	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	0	1	0	1

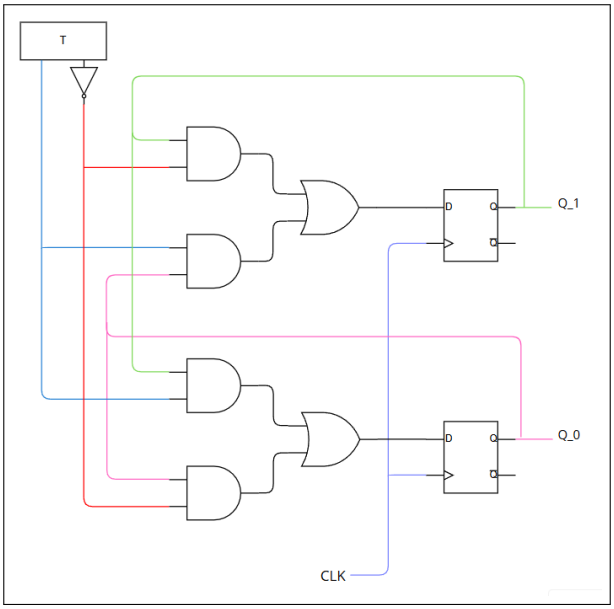
Ecuaciones de entrada

Recordando que en un Flip-Flop tipo D se cumple que $D = Q^{t+1}$, se obtienen:

$$D_0 = (T \wedge Q_1) \vee (\overline{T} \wedge Q_0)$$

$$D_1 = (T \wedge Q_0) \vee (\overline{T} \wedge Q_1)$$

Solución Circuito Lógico

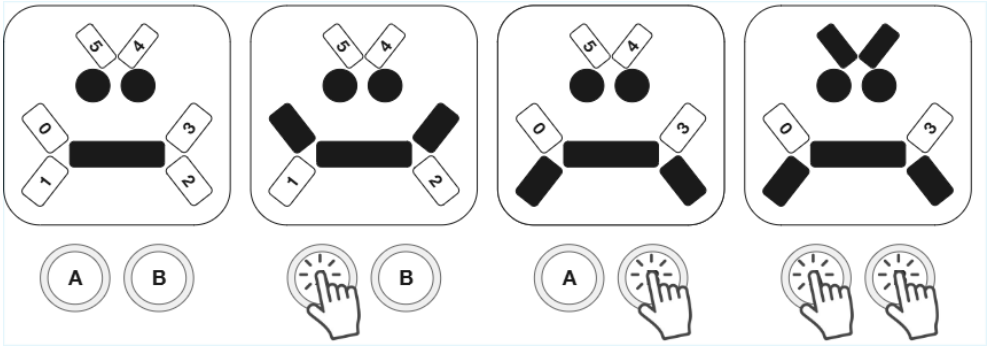


Observación

Los estados 00 y 11 no son válidos para el sistema. Sin embargo, si el circuito parte en un estado válido, permanecerá siempre dentro del conjunto de estados permitidos.

Pregunta 3: Almacenamiento (I1-2025-2)

Su docente, con el fin de comunicar sus emociones sin interrumpir la clase, decide usar un *display* de 6 segmentos para representar su estado de ánimo en torno a la participación del curso:



Como se observa en la figura, este funciona a partir de dos botones A y B: Si presiona solo el botón A, el *display* mostrará que el docente **está contento**; si presiona solo el botón B, el *display* mostrará que el docente **está triste**; y si presiona A y B de forma simultánea, esto señalará que el docente **está molesto**.

A partir de este contexto:
Diseñe, utilizando **solo** compuertas lógicas, *enablers*, multiplexores y flip-flops, un contador secuencial de 2 bits que incrementa o decrementa en cada flanco de subida a partir de las señales A y B antes descritas bajo las siguientes condiciones:

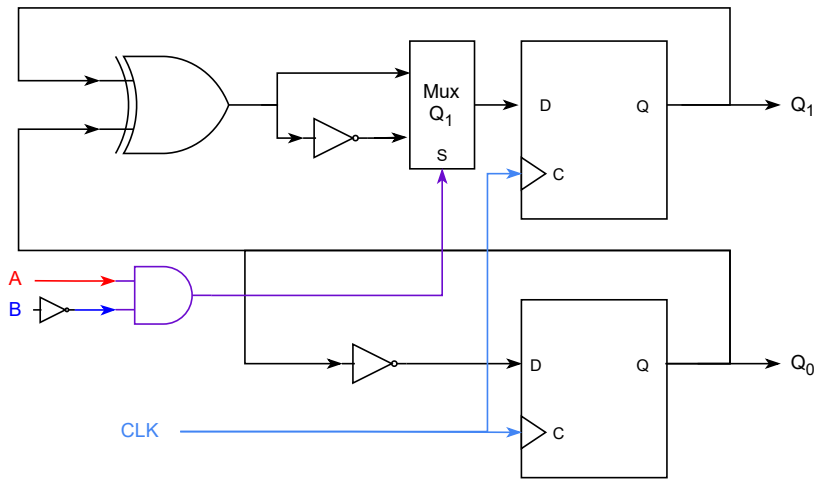
- Incrementa en una unidad si el docente está contento con la participación.
- Decrementa en una unidad si el docente está triste **o** molesto con la participación.
- Mantiene su valor en otro caso.

Puede asumir que su contador hace *overflow* en caso de incrementar estando en su máximo valor o decrementar estando en su mínimo valor.

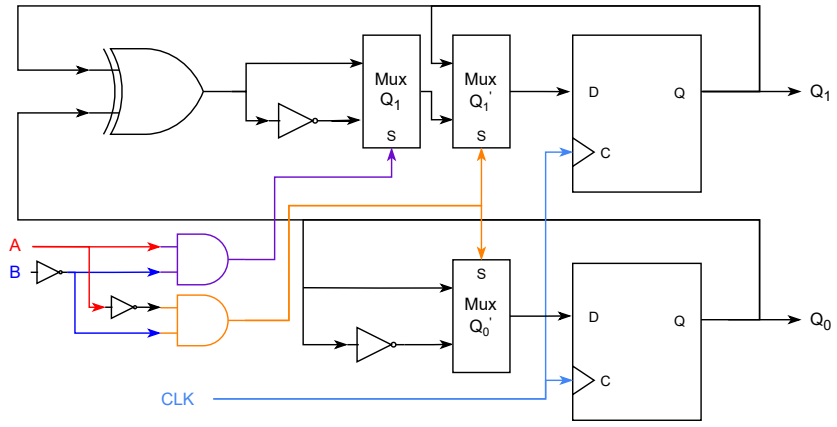
Solución: Si denotamos por $Q_1^t Q_0^t$ el estado del contador en un ciclo t de la señal de control `CLK`; $(Q_1^{t+1}Q_0^{t+1})^+$ su estado en el próximo ciclo si incrementa; y $(Q_1^{t+1}Q_0^{t+1})^-$ si decrementa, se puede elaborar la siguiente tabla de verdad:

Q_1^t	Q_0^t	$(Q_1^{t+1})^+$	$(Q_0^{t+1})^+$	$(Q_1^{t+1})^-$	$(Q_0^{t+1})^-$
0	0	0	1	1	1
0	1	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	0

Se deduce que $(Q_0^{t+1})^+ = (Q_0^{t+1})^- = \overline{Q_0^t}$, $(Q_1^{t+1})^+ = Q_0^t \oplus Q_1^t$ y $(Q_1^{t+1})^- = \overline{(Q_0^t \oplus Q_1^t)}$. Es decir, Q_0 será igual a `NOT` Q_0 independiente de que el contador incremente o decremente, mientras que Q_1 será igual a Q_0 `XOR` Q_1 en caso de incrementar y Q_0 `XNOR` Q_1 en caso de decrementar. Haciendo uso de flip-flops, multiplexores y las compuertas antes señaladas, se llega al siguiente diagrama:



Se añade el multiplexor `Mux Q1` para seleccionar el valor final de Q_1 dependiendo de si el docente está contento ($A = 1, B = 0$) o no. Esta elección es arbitraria y se pudo haber realizado la selección con la condición de decremento. Ahora, esta es una solución intermedia porque no contempla el caso en el que $A = 0$ y $B = 0$, donde se debe mantener el estado del contador y no decrementarlo como lo haría la implementación actual. Para ello, agregaremos un multiplexor para cada señal Q_i que seleccionará su valor sin modificar para la combinación de A y B antes señalada:



Se añaden los multiplexores `Mux Q0'` y `Mux Q1'` para seleccionar los valores originales de Q_0 y Q_1 respectivamente si se cumple que $A = 0$ y $B = 0$. En otro caso, se almacenará el valor actualizado dependiendo de si el contador incrementa o decrementa.

Es importante notar que se evaluará la correctitud del contador diseñado, no que sea igual al de esta pauta. Se otorga **1 pto.** por diseñar correctamente un contador de 2 bits incremental, **1 pto.** por diseñar correctamente un contador de 2 bits decremental, y **1 pto.** por actualizar su valor correctamente a partir de las señales A y B. Por cada criterio, se otorga la mitad del puntaje si existe, como máximo, un error de implementación.

Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

