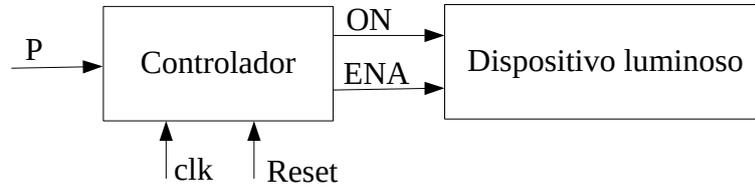


- Mínimo para pasar al oral: un problema
- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas
- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

Problema 1



Se dispone de un dispositivo luminoso (tipo luz de bicicleta) que tiene 3 modos de funcionamiento y se maneja con 2 señales ENA y ON.

Los modos de funcionamiento son 3: luz fija, luz intermitente rápida y luz intermitente lenta.

Cuando se enciende arranca siempre en el modo fijo. Si estando encendido recibe un pulso a 1 en su entrada ENA pasa al siguiente modo. Con ese mecanismo se puede recorrer cíclicamente los distintos modos.

Se quiere diseñar un circuito secuencial modo reloj que sea capaz de manejar el dispositivo luminoso con un único pulsador P y genere las salidas ON y ENA.

La entrada P está normalmente en 1 y cuando se presiona el pulsador baja a 0.

Cada vez que se recibe un pulso "largo" en la entrada P, el controlador debe encender el dispositivo (ON=1) si estaba apagado y apagarlo (ON=0) si estaba encendido.

Estando encendido, cada vez que recibe un pulso "corto" debe poner a 1 la señal ENA por exactamente un período de reloj. ENA debe valer 0 el resto del tiempo. Si vienen pulsos cortos estando apagado el controlador no los tendrá en cuenta, y por lo tanto no generará un pulso en ENA.

Se considera "largo" un pulso que encierre dos o más flancos activos de reloj. La duración mínima de los pulsos en P es igual al período de reloj, por lo que siempre encierra al menos un flanco activo.

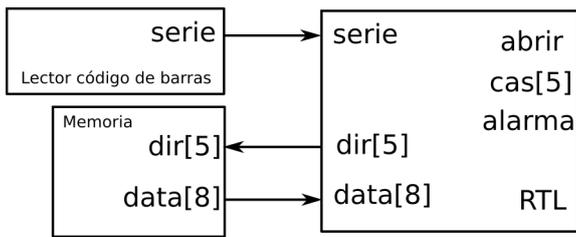
Los cambios en ON, tanto de apagado a encendido como de encendido a apagado, deben suceder simultáneamente con la subida del pulso en P, sin esperar el siguiente flanco activo de reloj.

Luego de Reset el dispositivo debe iniciarse apagado (ON=0).

Diseñar completamente el circuito secuencial modo reloj del controlador.

Nota: La solución queda de más de 4 estados.

Problema 2

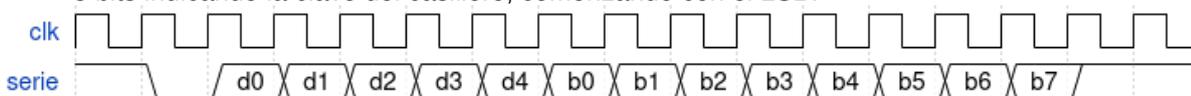


Se desea diseñar una parte del circuito de apertura de casilleros electrónicos para un supermercado. Se pide diseñar solamente la apertura de un casillero mediante un código de barras. El código de barras tiene codificado un número de casillero de 5 bits y una clave de 8 bits. La clave es generada de manera aleatoria al asignar el casillero y almacenada, por otro sistema, en una memoria que guarda en su dirección *i*, la clave asignada al casillero *i*.

El circuito a diseñar es capaz de leer dicha memoria, que en todo momento muestra en su salida **data[8]** el contenido del lugar de memoria que se especifica en su entrada **dir[5]**.

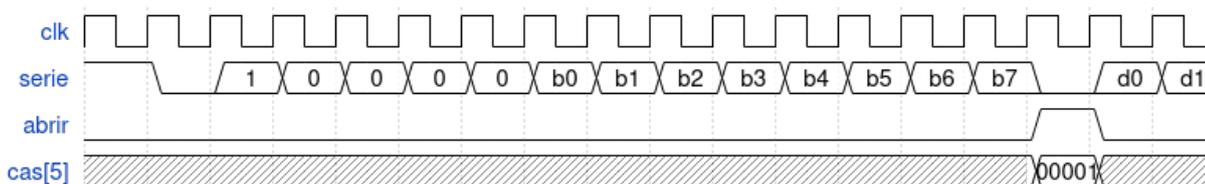
Se cuenta además con un bloque lector de código de barras que cuando se le acerca un código de barras este lo decodifica y lo envía de forma serie mediante su salida **serie**. Esta salida se encuentra sincronizada con el reloj del sistema y se mantiene en nivel alto mientras se encuentra en reposo. Envía la información de la siguiente forma:

- 1 bit de start en 0
- 5 bits indicando el número de casillero, comenzando por el LSB
- 8 bits indicando la clave del casillero, comenzando con el LSB.



El circuito a diseñar deberá leer los datos recibidos del lector de código de barras y verificar que la clave recibida es la misma que se encuentra almacenada en el lugar de memoria correspondiente. De ser así, se debe colocar el número de casillero en la salida **cas[5]** y dar un pulso de un período de reloj en la salida **abrir**. De lo contrario, se debe encender la salida **alarma** y permanecer encendida hasta recibir una clave reservada para el supervisor (0xFF). En este caso, **alarma** se debe encender en el siguiente período de reloj que se recibe b7 y apagarse cuando llega el b7 correspondiente a la clave del supervisor. Estando con **alarma** encendida el sistema deberá seguir funcionando para claves correctas.

El pulso en **abrir** (si corresponde) siempre debe darse en el siguiente período de reloj a recibir b7. Esto se debe a que es posible recibir un bit de start en **serie** a continuación del último bit de dato anterior. Este caso se muestra en la siguiente figura.



Se pide, secuencia, bloque de control y bloque de datos.