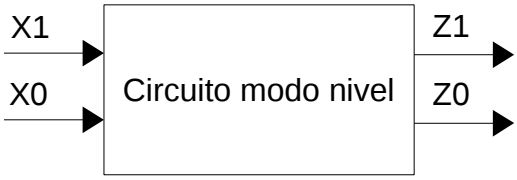
- Mínimo de aprobación: **un problema y un ejercicio**
- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas
- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

Problema 1

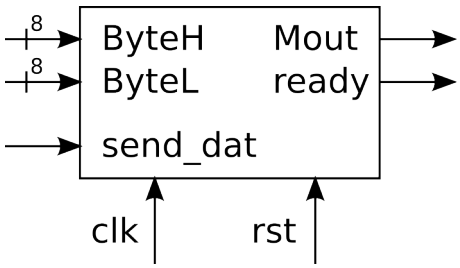
Diseñar un circuito secuencial modo nivel tal que la salida **Zi** vale 1 si y solo si la entrada **Xi** fue la última que cambió de valor.



Se pide diseño completo del circuito libre de carreras y espurios.

Problema 2

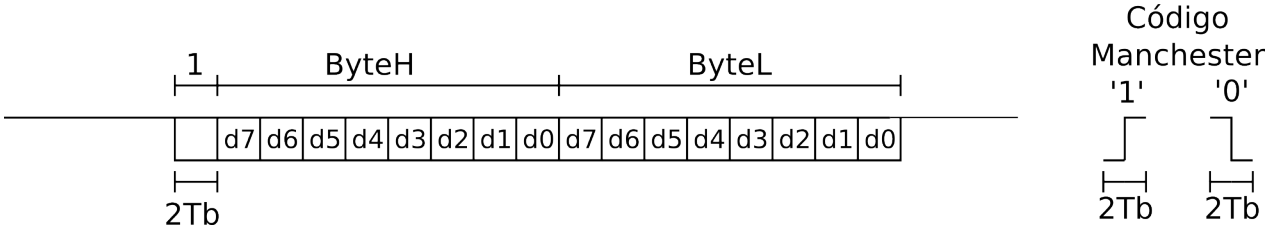
Se desea diseñar un módulo que reciba 2 bytes y los envíe en forma serie codificando cada uno de los bits en código Manchester.



El módulo tiene 2 entradas **ByteH** y **ByteL** de 8 bits por donde se reciben los bytes a transmitir. También cuenta con una entrada de un bit **send_data** en la cual se recibirá un pulso de 1Tclk que indica que hay datos en **ByteH** y **ByteL** y se debe comenzar la transmisión.
ByteH y **ByteL** tienen datos válidos solo durante el momento que **send_data** vale 1.

Se deberá generar una salida **ready** que valga 1 mientras no se esté transmitiendo y una salida de un bit **Mout** por donde se envían los bits.

La transmisión consiste en enviar: un bit en uno, seguido de los 8 bits de **ByteH**, comenzando por el bit más significativo, luego los 8 bits de **ByteL**. Los bits tienen una duración de 2Tb y se envían en código Manchester (incluido el primer 1), el cual consiste en representar un 1 como medio tiempo de bit en 0 y medio tiempo de bit en 1, y un 0 como medio tiempo de bit en 1 y medio en 0.



Mientras no se transmite nada, la salida **Mout** debe valer 1.

Dado que el reloj del sistema es mucho más rápido que 1/Tb, se cuenta con el bloque TimeBase cuya salida **pulso_Tb** genera pulsos de 1Tclk cada un tiempo Tb luego que se activa la entrada **start**. Mientras que **start** vale 0 no se generan pulsos. Deberá agregarse al módulo a diseñar las entradas y salidas necesarias para que pueda utilizar el bloque TimeBase.

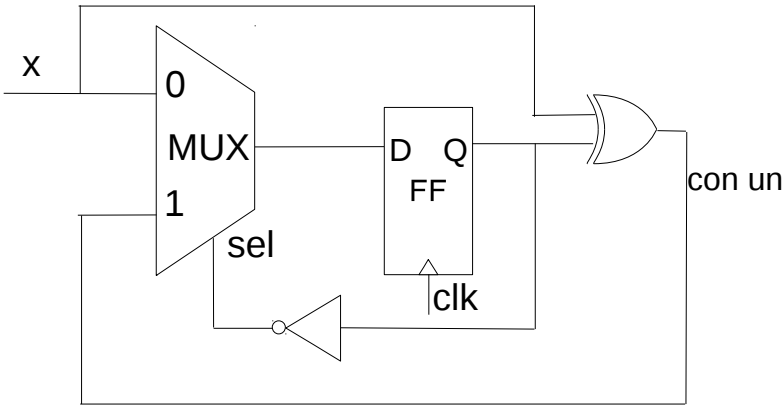
Se pide diseñar el circuito RTL detallado anteriormente (secuencia, bloque de control y bloque de datos).

Nota: suponer que no vendrá un pedido de enviar nuevos datos (**send_data**=1) mientras se este transmitiendo (**ready** = 0)

Ejercicio 1
Dados los siguientes números: A = 37, B = 126, C = -73
a) Cuál es la mínima cantidad de bits para poder representarlos en código complemento a 2.
b) Hallar sus expresiones en código complemento a 2 con la cantidad de bits encontrada en (a).
c) Realizar las siguientes operaciones indicando si el resultado es válido o no y porqué.
X = A + B, Y = A – C, Z = B + C

Ejercicio 2
a) Hallar la frecuencia máxima de funcionamiento del circuito, justificar con un diagrama de tiempos.

b) Hallar el intervalo en el cual la entrada X debe permanecer constante, justificar con un diagrama de tiempos.



Datos:

- tnotmin < tnot < tnotmax (tiempo de propagación de las compuertas NOT)
- tmuxmin < tmux < tmuxmax (tiempo de propagación del MUX entre datos y salida)
- tselmin < tsel < tselmax (tiempo de propagación del MUX entre sel y salida)
- txormin < txor < txomrax (tiempo de propagación del EXOR)
- tsu (tiempo de setup del FF)
- thmin < th < thmax (tiempo de hold del FF)
- tdmin < td < tdmax (tiempo de retardo del FF)