

- 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

Se recibe una secuencia de unos y ceros por medio de las señales:

- in** por esta señal se recibe la secuencia de unos y ceros en forma serial.
on_off pulso de 1 período de reloj que indica con el flanco de subida el comienzo y el fin de la secuencia.

Ambas señales están sincronizadas con el flanco de bajada de reloj.

Se dispone de una memoria RAM de $2^{16} \times 8$ bits síncrona con el flanco de bajada de reloj (ver figura 1). Para escribir un byte en una dirección dada de la RAM, se indica la dirección en la entrada **Dir[]**, el byte a escribir en la entrada **Data[]** y se debe poner en nivel bajo la entrada **we_n**. El byte se escribe en el siguiente flanco de baja de reloj. No se detalla la lectura pues no forma parte del problema.

Se debe diseñar un circuito RTL que almacene en la RAM esta secuencia de la siguiente forma:

1. Luego de un **RESET**, esperar el comienzo de la secuencia (flanco de subida en **on/off**).
2. Escribir en la dirección 00h, "FFh" si la secuencia comienza con "1" ó "00h" en caso contrario.
3. En las direcciones sucesivas se irán almacenando la cantidad de períodos en que **in** permanece en "1" ó en "0" respectivamente. Este valor es siempre menor a 255.
4. Dejar de grabar en la RAM cuando finaliza la secuencia (nuevo flanco de subida en **on_off**).
5. Luego de finalizada la secuencia, se debe indicar esto en la RAM grabando un "00h" en la siguiente dirección de memoria y no realizar nada más hasta la ocurrencia de un nuevo **RESET**.
6. La RAM nunca se llena.

Se pide descripción RTL, bloque de control y de datos.

Ejemplo: Tomando el diagrama de tiempos de la figura 2, el contenido de la RAM deberá ser el de la figura 3.

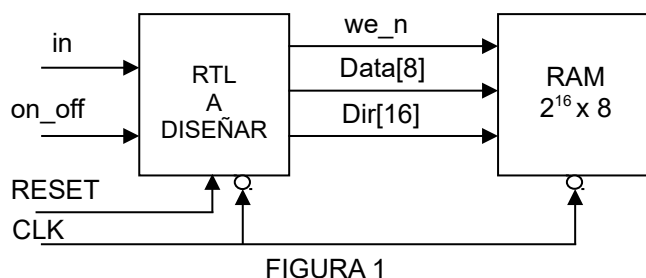


FIGURA 1

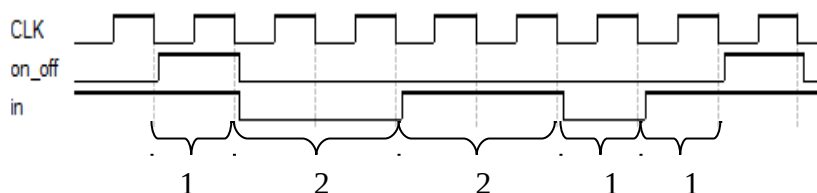


FIGURA 2

DIR	DATO	
0000h	FF h	Comienza en "1"
0001h	01 h	Señal
0002h	02 h	
0003h	02 h	
0004h	01 h	
0005h	01 h	
0006h	00 h	Fin
0007h	¿?	Datos sin grabar
A FFFFh		

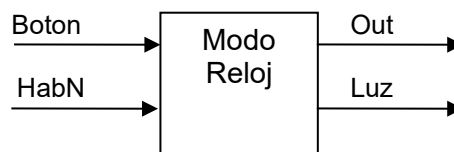
FIGURA 3

Problema 2

Se desea diseñar un circuito modo reloj con las entradas y salidas indicadas en la figura.

Ambas entradas cumplen que:

- son asíncronas
- el tiempo entre transiciones en una misma entrada es mayor a 1 período de reloj.



El funcionamiento del circuito debe ser el siguiente:

- Si la entrada **HabN** = 0 al detectar un cambio de 0 a 1 en la entrada **Botón** se deberá generar un único pulso a '1' de un período de reloj (1 Tclk) de duración en la salida **Out** a partir del siguiente flanco de reloj, sin importar la duración en que **Botón** permanezca en nivel alto. Si la señal **HabN** sube a 1 una vez que comenzó el pulso en la salida **Out** se ignorará hasta tanto se detecte un nuevo cambio de 0 a 1 en la entrada **Botón** y se completará el pulso en curso.
- Si la entrada **HabN** = 1 al detectar un cambio de 0 a 1 en la entrada **Botón** se deberá encender inmediatamente la luz de aviso manteniendo en 1 la salida **Luz** hasta el siguiente flanco luego de la bajada de **HabN**. A partir de ese momento (con **HabN** = 0) se deberá esperar a detectar un nuevo cambio de 0 a 1 en **Botón** para generar el pulso de duración 1 Tclk. Si vuelve a subir **HabN** se ignorará hasta tanto se detecte un nuevo cambio de 0 a 1 en la entrada **Botón**.

Se garantiza que al inicio las 2 entradas son 0.

Se pide diseñar completamente el circuito modo reloj especificado.

Ejercicio 1

Se desea trabajar con representación numérica en punto fijo.

- Determinar la cantidad de bits necesaria para poder representar números en el intervalo $[-3000, +3000]$ y con un error absoluto de 0,02.
- Hallar la representación de los siguientes números y determinar el error absoluto en cada uno de los casos:
 - 1853,148
 - -49,9998

Ejercicio 2

Dada la siguiente tabla de transiciones de un secuencial modo nivel, se pide:

a) Minimizarla.

b) Con la tabla mínima asignar variables de estado eliminando carreras y utilizando la menor cantidad de variables posible.

	00	01	11	10
q0	q2	<u>q0/0</u>	q1	--
q1	--	q0	<u>q1/1</u>	q3
q2	<u>q2/0</u>	q5	q3	q4
q3	q2	q0	<u>q3/0</u>	<u>q3/0</u>
q4	q5	<u>q4/1</u>	q1	<u>q4/1</u>
q5	<u>q5/0</u>	<u>q5/0</u>	q3	--