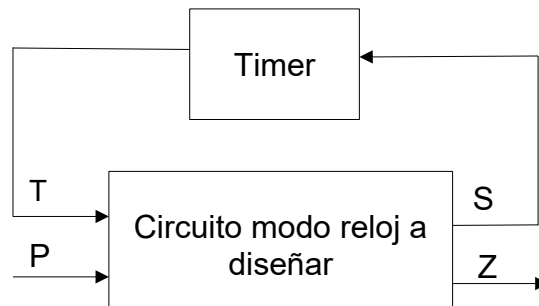


- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas
- Incluya un solo problema por hoja
- Deben estar numeradas y la primer hoja debe decir el total de hojas. Sea prolijo

Problema 1 (30 puntos)

Se desea diseñar un circuito modo reloj para eliminar rebotes de un pulsador industrial. Para esto, se cuenta con un timer, el cuál empezará a contar con un flanco de subida en S, y pondrá su salida T=1 durante 200ms. Pasados 200ms el timer expira poniendo T en 0.

Si mientras está contando recibe un flanco de bajada en S, eso provocará que T pase de 1 a 0 sin esperar los 200ms. En ambos casos, una vez que T volvió a cero el timer queda pronto para esperar un nuevo flanco de subida y comenzar otra nueva cuenta. La señal T está sincronizada con el reloj del sistema.



Al presionar el pulsador $P=1$ se deberá activar $S=1$ en forma inmediata y esperar que transcurran los 200ms ($T=1$). Cuando esto ocurra, se debe evaluar el estado de la señal P, si ésta sigue en uno se confirmará que el pulsador fue presionado activando $Z=1$ en el próximo flanco de reloj; en caso contrario Z se mantiene en cero. La salida del circuito deberá mantenerse en uno mientras el pulsador esté apretado.

La transición de botón apretado a no apretado debe validarse en forma similar, considerando para la entrada P los niveles opuestos a los considerados para la validación de la transición de no apretado a apretado.

Ejercicio 1 (7 puntos)

Dados $A=10110010$ y $B=01011101$

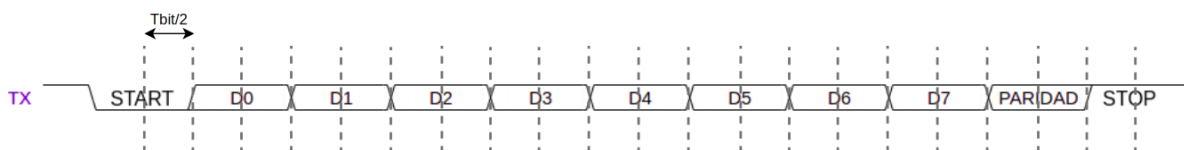
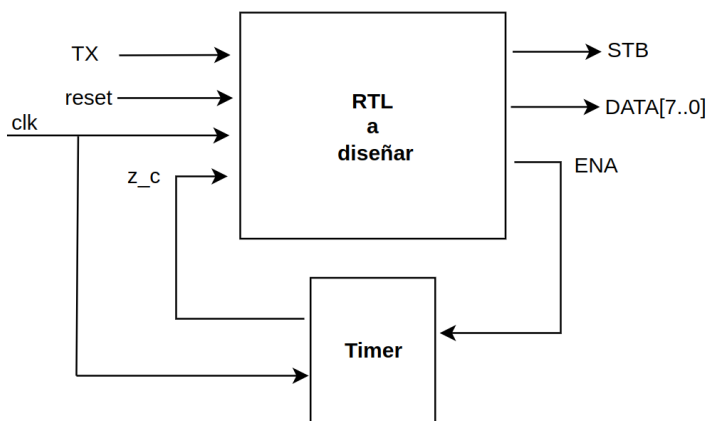
a) Realizar las operaciones $A+B$ y $A-B$ si A y B son números representados en complemento a dos indicando si hubo o no overflow, justificar.

b) Tomando A como parte entera, expresada en magnitud y signo, y B como parte fraccionaria de un número en punto fijo, convertirlo a punto flotante indicando el error de conversión. Utilizar la representación en punto flotante: 1 bit de signo, 5 bits para el exponente y 10 bits para la mantisa: $N=(-1)^s * 2^{(e-15)} * 1,f$

Problema 2 (30 puntos)

Se desea diseñar un circuito RTL que permita convertir un dato de entrada en un protocolo serie mediante la entrada **TX**, en un dato de salida paralelo por la salida **DATA[7..0]**.

El protocolo serie envía la información de la siguiente forma: 1 bit de arranque en 0 (START), 8 bit de datos comenzando por el LSB, 1 bit de paridad impar (PARIDAD) (no se va a chequear la paridad) y por último 1 bit de parada (STOP). La duración de cada bit es $T_{bit} >> T_{clk}$ del sistema.



Se cuenta con un bloque Timer el cual mide tiempos de $T_{bit}/2$. Funciona de la siguiente manera; cuando su señal de habilitación **ENA** pasa a 1, cada $T_{bit}/2$, genera en su salida **ZC** un pulso en alto de $1T_{clk}$.

El circuito a diseñar debe recibir los datos serie recibidos por **TX**. Para ello, una vez detectada la bajada de START inmediatamente se deberá habilitar el timer y utilizar correctamente los pulsos en ZC para recibir los datos.

Una vez recibido el último dato de información D7, se debe poner el dato recibido en la salida **DATA[7..0]** y subir **STB** a 1 durante $1T_{clk}$.

El sistema RTL debe quedar pronto para recibir un nuevo dato, tener en cuenta que tanto el D7 como el bit de paridad pueden valer 0 y no deben confundirse con un bit de start.

Se pide la descripción RTL del circuito y sus bloques de control y de datos.

Ejercicio 2 (8 puntos)

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

- a) Encontrar una asignación sin carreras en la cual no haya que agregar una variable de estado.
- b) Asignar salidas evitando espurios.

	00	01	11	10	00	01	11	10
a	a	b	d	a	0			1
b	a	b	b	c		1	0	
c	a	c	d	c		1		1
d	d	b	d	a	0		1	