

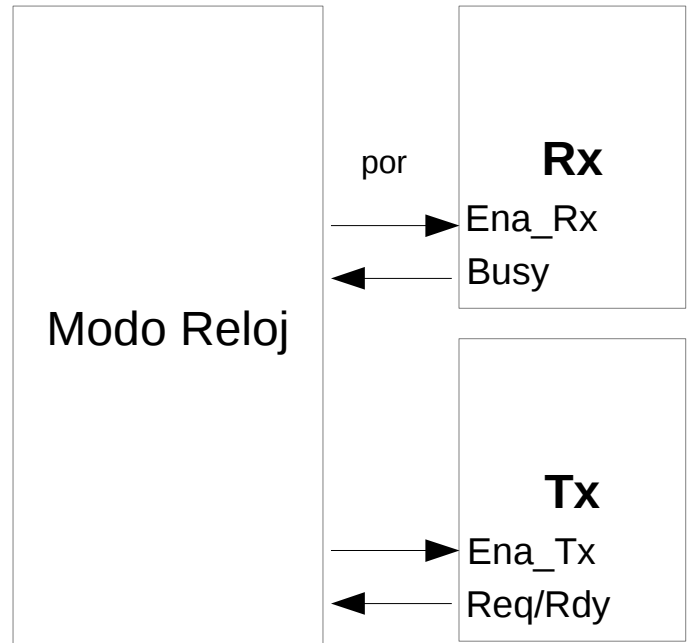
- Mínimo de aprobación: un problema y un ejercicio
- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas
- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

Problema 1

Se desea diseñar un sistema modo reloj para arbitrar el funcionamiento de dos bloques, un receptor de mensajes por radio (**Rx**) y un transmisor de mensajes por radio (**Tx**), los cuales no pueden funcionar simultáneamente por requerimientos energéticos.

Rx cuenta con una entrada de habilitación síncrona **Ena_Rx** activa por nivel alto y una salida **Busy** asíncrona que indica con un nivel alto si se está recibiendo un mensaje.

Tx cuenta con una entrada de habilitación síncrona **Ena_Tx** activa por nivel alto y una salida **Req/Rdy** síncrona por la cual, con pulsos de nivel alto, de un período de reloj de duración, solicita ser habilitado para transmitir mensajes y una vez que fue habilitado, con un pulso de iguales características, informa que finalizó la transmisión del mensaje.

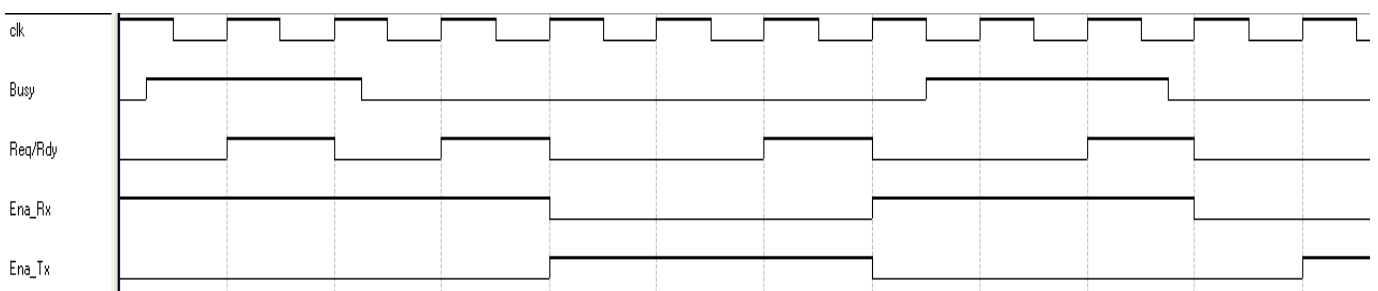


Rx tiene prioridad de funcionamiento, inicialmente comienza habilitado y su funcionamiento no puede ser interrumpido si se encuentra recibiendo un mensaje.

Las solicitudes de **Tx** deben ser ignoradas si se está recibiendo un mensaje (**Busy** =1) en **Rx**. Solo en el caso en que en un mismo período de reloj se finaliza de recibir un mensaje en **Rx** y **Tx** realiza una solicitud, se debe dejar ambos bloques deshabilitados por un período de reloj y luego habilitar **Tx**, en cualquier otro caso que se acepte una solicitud de **Tx** la habilitación de este y la deshabilitación de **Rx** deben ser simultáneas. Una vez habilitado **Tx** se espera a recibir otro pulso por **Req/Rdy** indicando el fin de la transmisión, a partir del cual se deshabilita **Tx** y se vuelve a habilitar **Rx**.

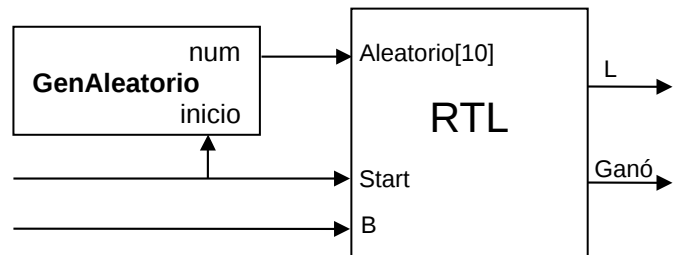
Mientras **Rx** se encuentra deshabilitado ignorará los mensajes que le lleguen por radio (es decir, **Busy** permanecerá en nivel bajo). Se puede suponer que los mensajes llegan separados por los menos por un período de reloj, así como las solicitudes de **Tx**.

Se pide diseñar el circuito modo reloj descrito.



Problema 2

Se desea diseñar un circuito RTL que implemente el siguiente juego. Para comenzar una partida, se oprime el botón **Start** y a continuación se enciende un led **L** durante un tiempo aleatorio. Una vez que se apaga el led **L**, un jugador debe intentar oprimir el botón **B** durante la misma cantidad de tiempo que estuvo encendido el led **L**. Si se consigue esto con una diferencia menor a 1 segundo, se enciende la señal luminosa **Ganó** y se mantendrá encendida hasta se oprima **B** en la próxima partida.



El circuito RTL tendrá las siguientes entradas, las cuales son todas asíncronas:

- Start:** Botón que se utiliza para comenzar el juego. Cuando está oprimido, **Start** está nivel alto. Se garantiza que cada nivel alto dura al menos 1 período de reloj.
- Aleatorio[10]:** Señal proveniente del bloque *GenAleatorio* que indica tiempo en unidades de 1/16 segundos. Cuando **Start** = 1, tiene un valor aleatorio entre 20 (01 0100 0000) y 40 (10 1000 0000) segundos. Cuando **Start** = 0 su valor es 0 (00 0000 0000).
- B:** Botón utilizado por el jugador para ser oprimido el tiempo indicado por *GenAleatorio*. Cuando está oprimido, **B** está en nivel alto.

y las salidas **L** y **Ganó** son señales luminosas que se encienden con nivel alto.

Para dar inicio al juego se oprime el botón **Start** y al soltar el botón, se debe encender el led **L** la cantidad de períodos de reloj que fue indicado por el bloque *GenAleatorio* cuando **Start** = 1. Una vez que se apaga el led **L**, se debe esperar a que el jugador oprima el botón **B**. Cuando el botón **B** es soltado se debe encender la señal luminosa **Ganó** si la diferencia entre la cantidad de flancos en que **B** estuvo oprimido y **Aleatorio[]** es menor que 00 0001 0000. **Ganó** quedará encendida hasta que se oprima **B**, luego que **L** pasó a 0 en la siguiente partida. Luego de un Reset, el valor de **Ganó** no es relevante.

La frecuencia de reloj del sistema es tal que 16 períodos equivalen a 1 segundo.
Se supondrá que el tiempo durante el cual **B** es oprimido es menor a 64 segundos.

Diseñar el circuito RTL, secuencia, bloque de datos y control, que implemente el juego descrito. La solución debe considerar que inmediatamente luego de bajar **B**, puede subir **Start**. Además, si se oprime **Start** o **B** cuando no corresponde, se hará caso omiso.

NOTA: Además de los bloques combinatorios usuales (comparadores, incrementadores etc.) se cuenta con un bloque RESTA_ABS que tiene como salida el valor absoluto de la diferencia de sus entradas.

Ejercicio 1

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

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

	00	01	11	10	00	01	11	10
q0	q2	q0	q0	q3		0	1	
q1	q1	q2	q1	q3	0		0	
q2	q2	q2	q0	q3	0	1		
q3	q2	X	q1	q3				0

Ejercicio 2

Se utilizan las siguientes representaciones numéricas:

Representación en punto fijo: 1 bit de signo, 7 bits de parte entera y 8 bits de parte fraccionaria

Representación en punto flotante de 2 bytes:

s signo 1 bit

e exponente 5 bits

f mantisa 10 bits

$$N = (-1)^s * 2^{e-15} * 1, f$$

- Convertir a punto fijo el siguiente número en punto flotante: s = 0, e = 10000, f = 3F9h; y calcular el error de dicha conversión. Justificar los resultados.
- Convertir a punto flotante el siguiente número en punto fijo: signo= 1, parte entera=0010011b, parte fraccionaria = 10110000b ; y calcular el error de dicha conversión. Justificar los resultados.