

- 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

Ejercicio 1

Se utilizan las siguientes representaciones numéricas, ambas de 2 bytes:

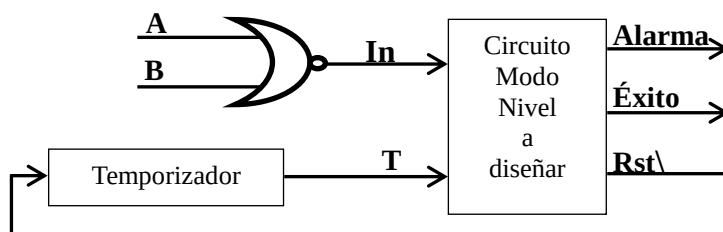
- i) representación en punto fijo: 1 byte parte entera (en magnitud y signo) y 1 byte parte fraccionaria.
 - ii) representación en punto flotante: 1 bit de signo, 5 bits para el exponente y 10 bits para la mantisa. ($N = (-1)^s * 2^{(e-15)} * 1.f$)
- a) Convertir a punto fijo el siguiente número en punto flotante: $s=1$, $e=01101$, $f=11\ 0011\ 0101$, y calcular el error de dicha conversión en caso de haberlo.
 - b) Convertir a punto flotante el siguiente número en punto fijo: parte entera = 83h, parte fraccionaria = 07h, y calcular el error de dicha conversión en caso de haberlo.

Problema 1

Un interruptor de potencia es un dispositivo diseñado para interrumpir la circulación de corriente de cortocircuito cuando se le da la orden. Para saber el estado en que se encuentra dispone de 2 señales, **A** y **B** mediante las cuales se pueden distinguir 3 estados posibles:

- **AB = 10:** Interruptor cerrado (circula corriente)
- **AB = 00:** interruptor en transición (abriendo o cerrando)
- **AB = 01:** interruptor abierto (no circula corriente)
- **AB = 11:** combinación no válida por razones mecánicas.

Se desea diseñar un **circuito secuencial modo nivel** que permita indicar si el tiempo de transición de un interruptor supera un umbral de tiempo dado (**Tu**). Para esto, se cuenta con un temporizador y se lo conecta junto con las señales **A** y **B** al circuito a diseñar como se indica en la figura.



El temporizador funciona de la siguiente manera. Cuando su entrada **Rst** = 0, el temporizador se encuentra en estado de reposo siendo su salida **T** = 0. Cuando **Rst** pasa a 1 comienza a contar el tiempo y **T** sube luego del tiempo **Tu** quedando en nivel alto hasta que se dé un nuevo reset. El tiempo que demora en bajar **T** luego de bajar **Rst** es mucho mayor al tiempo de compuerta.

El circuito debe funcionar de la siguiente manera. Al detectarse un estado de transición, se debe comenzar la cuenta en el temporizador y actuar según lo que ocurra primero:

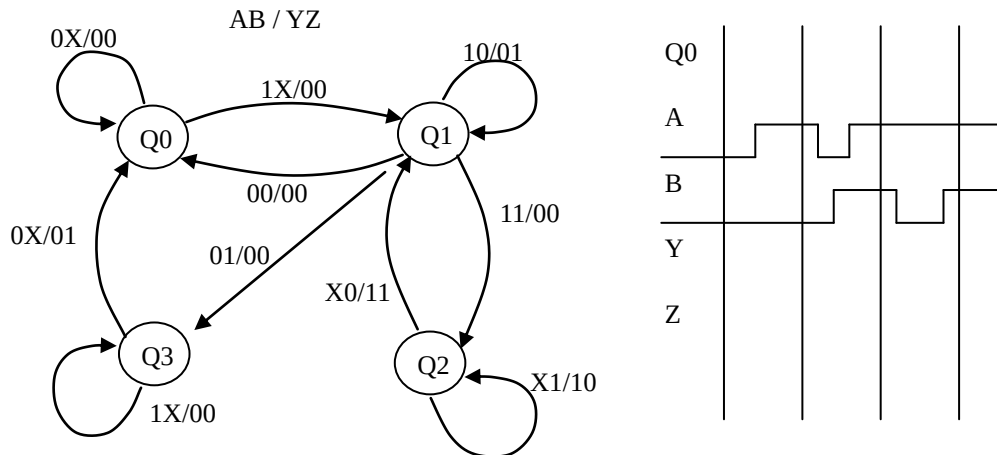
- 1) Si el interruptor finaliza la transición antes del tiempo **Tu**, se debe subir la señal **Éxito**, y debe permanecer así hasta que finalice el tiempo **Tu**. El resto del tiempo **Éxito** = 0.
- 2) Si el temporizador llega al tiempo **Tu** antes de finalizar la transición del interruptor, se debe subir a 1 la señal **Alarma** y deberá permanecer así hasta que el interruptor finalice la transición y **T** baje. El resto del tiempo **Alarma** = 0.

Considerando modo fundamental, diseñar el circuito modo nivel sin carreras, azares ni espurios.

Nota: **Tu** y los tiempos de transición de un interruptor son del orden de las decenas de milisegundos.

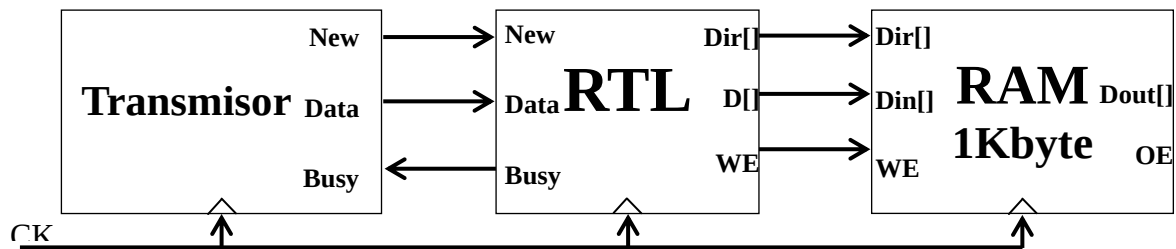
Ejercicio 2

Dado el siguiente diagrama de estados de un circuito secuencial modo reloj: **a)** Dar la tabla de estados. **b)** Completar el estado y las salidas en el diagrama de tiempos. Las líneas verticales representan los flancos activos del reloj.



Problema 2

Se desea implementar un circuito RTL para la recepción de palabras que se transmiten en forma serial y consecutiva.



El transmisor genera un pulso en nivel alto de un período de reloj en la señal **New** para comenzar a enviar palabras, y luego de un tiempo mayor a un período de reloj comienza a transmitir en forma serial cada palabra por su salida **Data**. Las palabras se transmiten de la siguiente forma:

- 1 bit de arranque en nivel bajo.
- 8 bits de datos, **comenzando por el menos significativo**.
- n bits de parada en nivel alto, con $n \geq 0$ (notar que puede NO haber bits de parada).
- cada bit demora 1 período de reloj.
- en reposo **data** = 1

Las palabras recibidas deberán guardarse en la memoria RAM a partir de la dirección 0. Una vez que se llena la memoria debe darse un pulso a 1 de un período de reloj en la señal **Busy**, el cual debe coincidir con el bit 7 de la última palabra recibida.

La memoria tiene una capacidad de 1K x 8 bits, que equivale a 1024 direcciones de 8 bits cada una. Para guardar un dato, se debe poner WE en alto y cuando ocurra un flanco activo de reloj se escribe el dato que se encuentra en la entrada **Din[7..0]** en la dirección indicada en la entrada **Dir[9..0]**.

Al recibir el transmisor la señal **Busy**, este suspenderá el envío de palabras por un tiempo. Durante este tiempo, otro dispositivo se encargará de leer los datos de la memoria.

Luego de esto, el transmisor comenzará a transmitir nuevamente, generando un nuevo pulso en la señal **New** y todo comenzará nuevamente.

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

Nota: No se detalla cómo se realiza la lectura de la memoria ni sobre el dispositivo que la lee pues no forma parte del problema.