

Problema 1 (8 puntos)

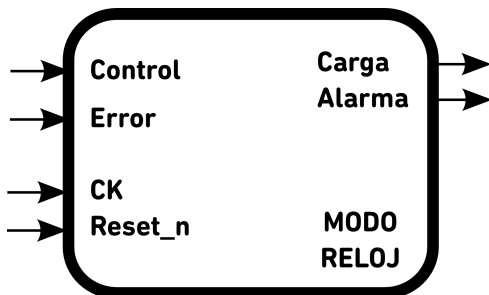
La siguiente tabla corresponde a un circuito secuencial modo nivel.

	00	01	11	10	00	01	11	10
a	c	<u>a</u>	b	X		1		
b	d	a	<u>b</u>	<u>b</u>			0	0
c	<u>c</u>	a	<u>c</u>	b	0		1	
d	<u>d</u>	a	b	<u>d</u>	1			1

- a) Dar una nueva tabla con una asignación de variables de estado sin carreras. Utilizar la mínima cantidad de variables de estado posible.
- b) Asignar salidas a los inestables de modo de no tener espurios. Asignar solamente en donde sea necesario.

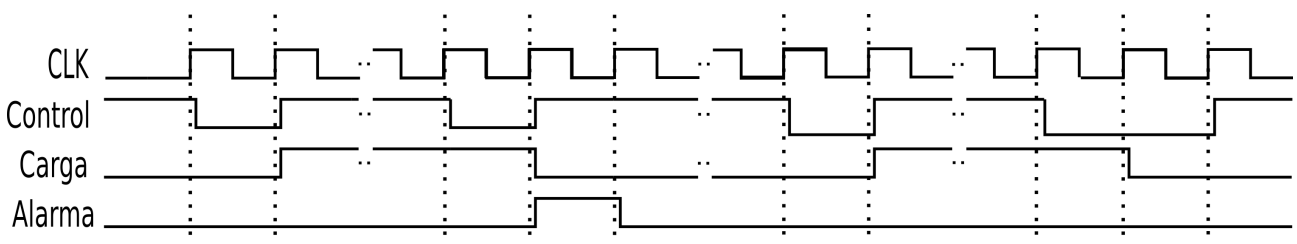
Problema 2 (30 puntos)

Se desea implementar un circuito modo reloj para controlar la carga de una batería. El circuito posee las entradas y salidas que se muestran en la figura. El circuito cuenta con una entrada **Control** que le indica cuando se debe iniciar y detener la carga y una entrada **Error** que se activa si hubo un problema con la carga. **Control** se encuentra sincronizada con el reloj del sistema.



Luego de un reset la carga debe estar deshabilitada, para ello la salida **Carga** debe permanecer en nivel bajo. La señal **Control**, que vale 1 en estado de reposo, indica mediante un pulso a cero de un período de reloj que se debe comenzar la carga. La carga debe habilitarse y deshabilitarse de manera síncrona, salvo en el caso de error que se explica más adelante. La carga termina cuando la señal **Control** baja a cero. Si **Control** baja a cero por dos períodos de reloj quiere decir que la carga finalizó de manera exitosa. En cambio, si **Control** baja a cero solo por un período de reloj quiere decir que la carga fue deshabilitada antes de que finalizara. En este caso la salida **Alarma** debe dar un pulso en alto de 1 período de reloj antes de volver a esperar una nueva carga (Ver diagrama). En cualquiera de los dos casos la señal **Carga** debe bajar a 0 luego del primer flanco activo que **Control** bajó a 0.

finalizó de manera exitosa. En cambio, si **Control** baja a cero solo por un período de reloj quiere decir que la carga fue deshabilitada antes de que finalizara. En este caso la salida **Alarma** debe dar un pulso en alto de 1 período de reloj antes de volver a esperar una nueva carga (Ver diagrama). En cualquiera de los dos casos la señal **Carga** debe bajar a 0 luego del primer flanco activo que **Control** bajó a 0.



Para saber si ocurrió un error durante la carga, se cuenta con la entrada **Error**. Se asume que la señal **Error** puede activarse únicamente mientras la carga está habilitada. Si se activa **Error** (**Error**=1), inmediatamente se debe deshabilitar la carga y la salida **Alarma** debe pasar a 1. En este escenario **Alarma** permanece en 1 y no se tiene en cuenta la entrada **Control**, es decir, es necesario un reset del sistema para volver al funcionamiento normal.

Problema 3 (7 puntos)

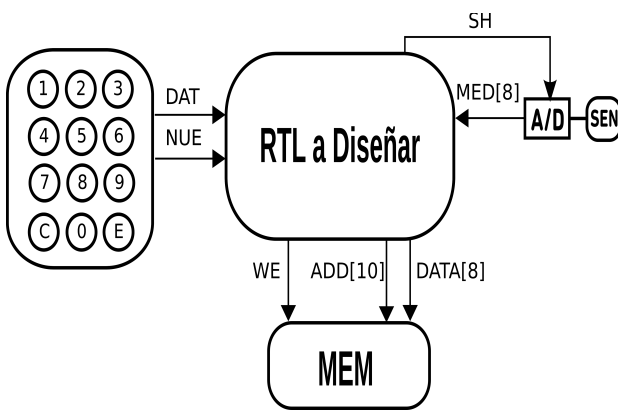
Se utilizan las siguientes representaciones numéricas, ambas de 2 bytes:

1. Representación en punto fijo: 1 byte parte entera (en magnitud y signo) y 1 byte parte fraccionaria.
2. 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: $s=0, e=01110, f=01\ 1001\ 0111$, y calcular el error de dicha conversión en caso de haberlo.

b) Convertir a punto flotante el siguiente número en punto fijo: parte entera = 92h, parte fraccionaria = 15h, y calcular el error de dicha conversión en caso de haberlo.

Problema 4 (30 puntos)

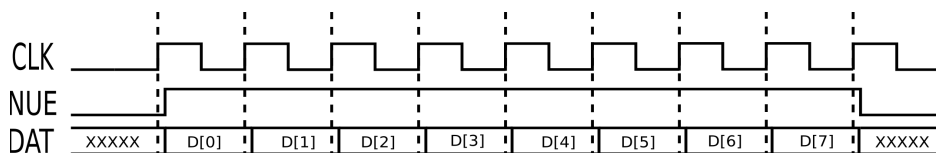


Se desea implementar un circuito RTL que controle un conversor A/D para tomar muestras periódicamente de un sensor y guardarlas en una memoria. El conversor A/D convierte una señal analógica en una digital y funciona de la siguiente manera; cuando su entrada **SH** se encuentra en alto pone en su salida (**MED[8]**) el valor digital de la señal de entrada analógica tomada del sensor. La salida digital del conversor A/D es de 8 bits. El tiempo que demora desde que **SH** sube hasta que **MED[8]** tiene un dato válido es mucho menor a 1 Tclk.

Adicionalmente, se posee una memoria que permite almacenar datos de 8 bits y que posee 1024 lugares. Para escribirla se debe poner en su entrada **DATA[8]** el dato a guardar, en **ADD[10]** el lugar de memoria en el cual se quiere escribir el dato y por último indicar con un pulso de 1 período de reloj en la entrada **WE** para escribir.

El circuito debe tomar muestras del sensor cada un tiempo T_s y guardarlas en lugares consecutivos de memoria. La memoria se debe comenzar a llenar desde la dirección más baja y una vez que se llena, volver a empezar. Se supondrá que hay otro circuito que lee la memoria y utiliza los datos.

El tiempo T_s será configurable entre 10 Tclk y 255 Tclk. Por defecto se debe establecer en 255 Tclk. Para poder modificar este valor se dispone de un teclado como el que se muestra en el diagrama. Cuando un usuario presiona un número (entre 0 y 255) y luego la tecla E, éste envía el número a través de sus salidas **DAT** y **NUE** utilizando el siguiente protocolo serie. La señal **NUE** se mantiene en bajo mientras no se esta enviando ningún dato. Esta señal sube al mismo tiempo que en la señal **DAT** se encuentra el LSB del dato a enviar. A continuación se envían los 8 bits de datos, uno por período de reloj. La señal **NUE** se mantiene en alto durante toda la transmisión de datos.



Cuando se termina de recibir un T_s nuevo, se debe actualizar el tiempo de muestreo del sensor. Si el nuevo T_s es menor al tiempo ya transcurrido, se debe pasar a recibir una muestra. En caso contrario debe continuar esperando hasta completar el nuevo tiempo T_s para tomar la nueva muestra.

Se pide descripción RTL, bloque de datos y control.