

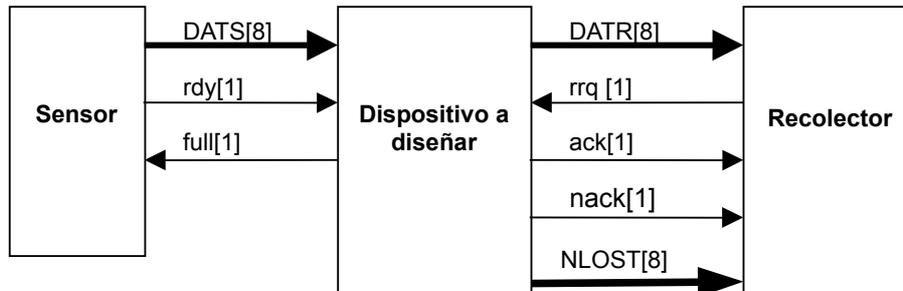
- Nombre y CI en cada hoja
- Numere las hojas, indique el total en la primera
- Utilice sólo un lado de las hojas

- Incluya un solo problema por hoja
- **Sea prolijo**
- **Aprobación:** mínimo UN problema

PROBLEMA 1

Se tiene un sistema basado en Z80 como el que se muestra en la figura.

El dispositivo sensor genera datos de 8 bits a tasa constante, mientras que el dispositivo recolector los solicita a una tasa variable. El sistema a diseñar actúa como colchón adaptando la diferencia entre las tasas de producción y recolección de datos de ambos dispositivos.



Los datos se almacenan temporalmente mediante una cola FIFO implementada en software.

Para poner un dato en la cola se deberá invocar a la subrutina **PutCola** que recibe como parámetro en el registro A el dato a escribir. Si tiene éxito para escribir el dato devuelve el flag Z prendido. Si en cambio la cola está llena, entonces no escribe el dato y devuelve el flag Z apagado.

Para obtener el dato que está primero en la cola se deberá invocar a la subrutina **GetCola**. Si la cola no está vacía, GetCola saca el primer dato de la cola y lo devuelve en el registro A con el flag Z prendido. Si en cambio no hay datos para leer en la cola la subrutina devuelve el flag Z apagado y el contenido del acumulador es desconocido.

Cada vez que el dispositivo sensor tiene un dato disponible lo pone en su salida **DATS** y da un flanco de subida en la señal **rdy** (los datos están garantizados solamente en el flanco de subida de **rdy**). El programa principal estará dedicado a leer los datos del dispositivo sensor y escribirlos en la cola FIFO.

En todo momento la salida **full** debe indicar si el último dato leído del sensor fue escrito con éxito en la cola FIFO (**full** = 0) o si por el contrario fue descartado por estar la cola llena (**full**=1). Esta salida debe mantener su valor entre los instantes en que se intenta escribir en la cola. Además cada vez que se descarta un dato debe incrementarse en uno la salida **NLOST** de 8 bits, hasta un valor máximo de 255. Por encima de ese valor no se lleva más la cuenta de los datos descartados y se mantiene el valor 255 en **NLOST**.

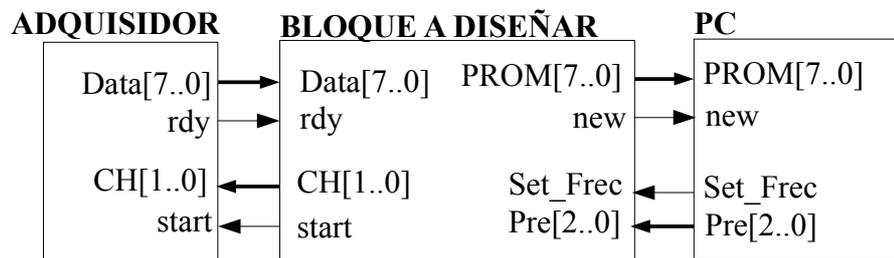
El dispositivo recolector solicita un nuevo dato dando un pulso a "0" de un período de reloj en la señal **rrq**. Esto debe provocar una interrupción en el sistema. La rutina de atención a la interrupción intenta leer un nuevo dato de la cola invocando GetCola. Si tiene éxito debe escribir el dato en el puerto de salida **DATR** y avisar al dispositivo recolector llevando a 1 la señal **ack**. Si por el contrario la cola está vacía se debe informar llevando a 1 la señal **nack**. Ambas señales (**ack** y **nack**) deben volver a cero instantáneamente con el próximo pulso de **rrq**.

Se pide:

- a) Diseñar el hardware completo con 32k de ROM y 32K de RAM.
- b) Escribir todo el software necesario para el funcionamiento del sistema a partir de un reset (inicialización, programa principal que lee los datos del sensor y los escribe en la cola FIFO, rutina de atención a la interrupción que lee los datos de la cola FIFO y los suministra al dispositivo recolector).

PROBLEMA 2

Se desea utilizar un microprocesador T80 para implementar un sistema que procese las 4 señales EEG de una determinada zona cerebral. Estas llegan desde un adquisidor y se debe enviar el promedio de lo adquirido a la PC para su posterior estudio.



El **Bloque Adquisidor** realiza la conversión A/D del canal indicado en **CH[1..0]** al recibir un flanco de subida en la entrada **start**. Una vez adquirida y digitalizada la señal correspondiente al canal, el bloque sube a 1 la señal **rdy** por un periodo de reloj, colocando el valor de la conversión en **Data[7..0]**. Este valor está representado en entero sin signo de 8 bits y se sabe que el valor máximo que toma es 0x3F (63).

El **Bloque a Diseñar** debe encargarse periódicamente (a frecuencia **f_adq**) de adquirir los cuatro canales, calcular el promedio, escribirlo en **PROM[7..0]** y dar un pulso en **new** indicando que hay un nuevo dato. La frecuencia **f_adq** puede ser modificada desde el PC cuando el operador lo desee.

El **Bloque PC** además de leer el valor del promedio, se encarga de setear la frecuencia de adquisición. Cada vez que se debe setear la frecuencia, este bloque envía un flanco de subida en **Set_Frec**, colocando el código de la frecuencia a setear en **Pre[2..0]**. Este código toma valores del 1 al 7.

La secuencia de cuatro adquisiciones comienza cuando se genera la interrupción periódica a frecuencia **f_adq.**, esto deberá estar a cargo de un bloque timer. La rutina de atención a esta interrupción debe solicitar la lectura del canal 0 al **Bloque Adquisidor** y luego se deberá generar otra interrupción cuando el pulso en **rdy** indica que finalizó la conversión. La rutina de atención a esta última interrupción, la del adquisidor, debe almacenar el valor adquirido y solicitar la lectura del siguiente canal, salvo en el último en el que se deberá de calcular el promedio de las lecturas y enviarlas al PC.

El programa principal debe consultar por la ocurrencia del flanco de subida de **Set_Frec** y reconfigurar el Timer para que interrumpa a la nueva frecuencia una vez que haya expirado la cuenta anterior. El valor de **Pre[2..0]** se corresponde con el valor del preescaler a configurar en el timer. El valor de la cuenta debe configurarse para que con **Pre[2..0] = 7** la frecuencia **f_adq** sea 3125 Hz. La frecuencia de reloj es fck=50MHz

Se pide:

- Hardware del sistema, se dispone de bloques controladores de interrupciones y bloque timer. Se supondrá la memoria ya implementada con 32K de RAM y 32K de ROM.
- Inicialización del sistema y reservas de memoria. Luego de la inicialización se deberá saltar al loop del programa principal. El valor de la frecuencia **f_adq** por defecto debe ser 3125 Hz.
- Rutinas de atención a interrupciones, siendo la del Timer la de mayor prioridad.
- Programa principal que atienda los pedidos de cambio de frecuencia.