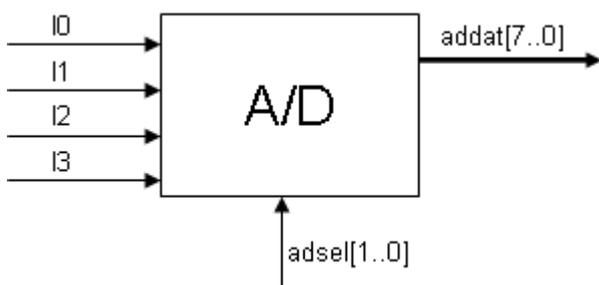


<ul style="list-style-type: none"> - Nombre y CI en cada hoja - Numere las hojas - Indique el total de hojas en la primera - Utilice solo un lado de las hojas 	<ul style="list-style-type: none"> - Incluya un solo problema por hoja - Sea prolijo - Aprobación: mínimo UN problema
--	--

PROBLEMA 1

Se desea diseñar un adquirente de datos basado en el procesador Z80 utilizando el convertor A/D de 4 canales de la figura. Este tiene 4 entradas analógicas (**i0..i3**) y una salida **addat[8]** de 8 bits que en todo momento presenta el valor convertido de la entrada analógica **i** seleccionada por la entrada de selección **adssel[2]** de dos bits.

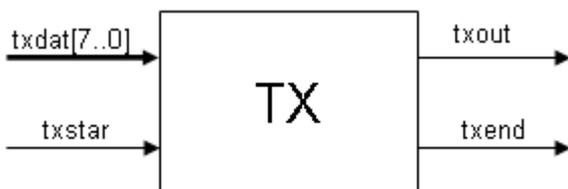


Una vez por minuto el sistema debe adquirir muestras de uno de los canales durante un segundo, procesarlas mediante una subrutina que se supondrá dada y transmitir por un puerto de salida el nro de minuto y el resultado del procesamiento. En el siguiente minuto el proceso debe repetirse para el siguiente canal. Luego de cuatro minutos se vuelve al canal inicial.

Las señales se adquieren a 100 muestras por segundo. Para ello se cuenta con una señal **TIC** periódica de 100Hz que se utilizará para generar una interrupción periódica al procesador. La subrutina de atención a la interrupción debe llevar la cuenta de tics y de segundos, y durante el intervalo de adquisición (durante todo el último segundo de cada minuto) debe adquirir UN dato en cada interrupción y almacenarlo en una tabla de 100 lugares. Además debe informar al programa principal, por un mecanismo a diseñar, cada vez que se completa un intervalo de adquisición y queda llena la tabla con las 100 muestras adquiridas.

El programa principal debe realizar las siguientes tareas cada vez que se completa un intervalo de adquisición (al final de cada minuto):

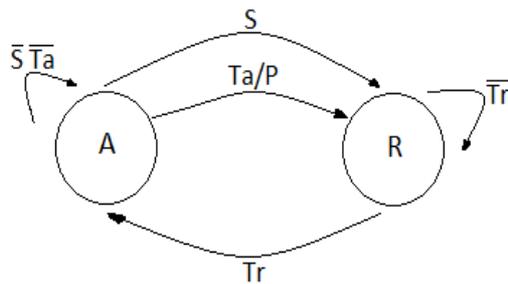
- Procesar los datos invocando la subrutina PROCESAR. Esta recibe en el par de registros HL la dirección de comienzo de los datos a procesar y devuelve en el par de registros DE el valor RESULTADO calculado a partir de las 100 muestras.
- Llevar la cuenta de los minutos transcurridos desde un reset en la variable MINUTO de 8 bits. Esta cuenta se lleva módulo 256.
- Transmitir por el dispositivo TX que se describe más adelante los siguientes tres bytes: MINUTO, byte alto de RESULTADO, byte bajo de RESULTADO en ese orden.
- Manejar la entrada **adssel** del convertor A/D para que adquiera el canal adecuado en el siguiente minuto.



El dispositivo TX comienza a transmitir los datos presentes en su entrada **txdat[8]** cuando recibe un flanco de subida en su entrada **txstart** e indica con un pulso a cero en su salida **txend** cuando completa la transmisión y queda listo para recibir un nuevo dato.

Se pide:

- a) Todo el hardware (memoria, convertor A/D, dispositivo TX e interrupción periódica con la señal TIC).
- b) Rutina de atención a la interrupción.
- c) Programa principal
- d) Inicialización y directivas de reserva de memoria para todas las variables utilizadas.

PROBLEMA 2

El diagrama de estados de la figura describe el funcionamiento de un marcapasos sencillo. La entrada **S** sensa el latido normal del corazón y la salida **P** lo estimula artificialmente en caso de ser necesario.

Al ingresar al estado **A** (Alerta) se arranca un temporizador para medir un tiempo **Ta**.

Si antes que transcurra el tiempo **Ta** la entrada **S** pasa a 1, entonces el corazón latió

espontáneamente y se pasa al estado **R** sin dar un estímulo en **P**.

Si por el contrario finalizado el tiempo **Ta** la entrada **S** no ha subido, entonces se da un estímulo artificial mediante un pulso a 0 de corta duración en la salida **P** y se pasa al estado **R**. El resto del tiempo **P** = 1.

En cualquiera de los dos casos se permanece en el estado **R** durante un tiempo **Tr** y se retorna al estado **A**. Observar que mientras se está en el estado **R** se ignoran las conmutaciones en la entrada **S**. A los efectos del problema se supondrá que cuando se retorna al estado **A** la entrada **S** ya está estable en cero hasta el nuevo latido.

Los valores de los intervalos **Ta** y **Tr** se expresan en múltiplos de **T1** = 10ms y se configuran en dos variables de 8 bits (**val_ta** y **val_tr**).

Estas variables pueden ser actualizadas por un dispositivo externo a través del siguiente protocolo:

- un flanco de subida en la entrada **prog** indica que hay nuevos valores en las entradas **port_ta** y **port_tr**. Los valores de ambas entradas deben memorizarse con el flanco de **prog** ya que solamente son válidos en ese instante.
- la salida **busy** debe subir con el flanco de **prog** y bajar una vez que los nuevos valores fueron almacenados por el sistema en las variables **val_ta** y **val_tr**.

Se debe diseñar un sistema con microprocesador Z80 que implemente el funcionamiento descrito. Se trabajará por interrupciones para implementar el diagrama de estados mientras que el programa principal se encargará de modificar los valores de los temporizadores cuando le sea solicitado.

Se cuenta con un CTC para generar interrupciones en los flancos de subida de **S** (un canal en modo contador con valor inicial 1) y para generar interrupciones al finalizar los tiempos **Ta** y **Tr**. Para esto último se sugiere utilizar dos canales, uno para medir intervalos de duración $T1 = 10\text{ms}$ y otro para contar intervalos $T1$. La frecuencia de reloj es $f_{ck} = 32\text{kHz}$ ($T_{ck} = 31.25\ \mu\text{seg.}$), observar que $T1 = 10\text{ms} = 16 \times 20 \times T_{ck}$.

Se pide:

- Todo el hardware.
- Programa principal que maneje la actualización de los valores de **Ta** y **Tr**.
- Rutinas de atención a interrupción por flanco de subida en **S** y por fin de temporizador que implementen el diagrama de estados de la figura.
- Rutina de reset que inicialice **todo el sistema** poniéndolo en el estado **A** y salte al comienzo del programa principal. Incluir las directivas de reserva de memoria necesarias para todas las variables utilizadas. Los valores iniciales de los temporizadores serán **VAL_TA_INI** = 70 y **VAL_TR_INI** = 30.