

<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 quiere dotar a un sistema basado en Z80 de un periférico BOTON, con un comportamiento similar al del botón principal de un ratón. El nivel de reposo de la señal BOTON es alto, por lo que un flanco de bajada indica que se ha presionado el botón y un flanco de subida que se lo ha liberado. Partiendo de un estado de reposo con el botón liberado, se desea diferenciar entre secuencias de click y de doble click como sigue:

- **click:** se presiona el botón, se libera el botón y transcurren Tdoble ms desde la liberación sin un nuevo evento.
- **doble click:** se presiona el botón, se libera el botón y antes de Tdoble ms vuelve a presionarse el botón.

La funcionalidad del botón se debe agregar a través de interrupciones. El sistema ya cuenta con 32K de ROM, 32K de RAM y otros puertos que ocupan la mitad inferior del espacio de E/S. Se debe incorporar un CTC, un puerto de entrada de 8 bits para leer la constante TICS dada por llaves de configuración del sistema ($T_{doble} = 256 * TICS * T_{ck}$) y dos salidas SIMPLE y DOBLE de un bit cada una para indicar cuál fue la última secuencia detectada.

Cuando se valida la detección de una secuencia deberá activarse a nivel alto la salida que corresponda (SIMPLE o DOBLE) hasta que se vuelva a presionar el botón y comience la detección de la siguiente secuencia.

El CTC se utilizará para generar interrupciones al presionarse el botón (programándolo como contador con cte=1) y para generar interrupciones al vencer el tiempo Tdoble. Para esto último se sugiere programar un canal del CTC en modo timer arrancando el flanco de subida de la señal BOTON.

Se pide:

- a) Hardware a agregar al sistema.
- b) Rutinas de atención a cada una de las interrupciones utilizadas. En ningún caso una rutina de atención a la interrupción debe quedar esperando por un evento del ratón. Cuando se detecta una secuencia se debe de activar la salida (SIMPLE o DOBLE) e invocar las subrutinas SIMPLE_RUT y DOBLE_RUT que se supondrán ya implementadas.
- c) Escribir la subrutina **init_boton** que es invocada por la inicialización del sistema como se muestra en el recuadro. Esta subrutina debe inicializar todo lo necesario (CTC, puertos, variables, entradas en la tabla de interrupciones) para que el botón funcione correctamente luego de un reset. Otros dispositivos ya ocupan las 3 primeras entradas de la tabla de interrupciones.

```

org 0
ini:

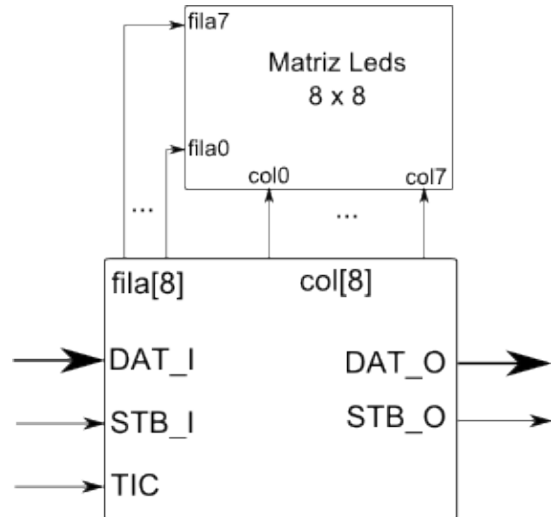
ld sp, 0
im 2
ld a, 80h
ld i, a
call init_sistema
call init_boton
ei
jp principal
    
```

Escribir las directivas de reserva de memoria para todas las variables utilizadas. Se cuenta con memoria libre entre 9000h y A000h.

PROBLEMA 2

Se dispone de matrices de leds de 8 filas por 8 columnas. El funcionamiento de la matriz es el siguiente: para encender el led en la posición *i, j* de la matriz, se debe poner simultáneamente un 0 lógico en las entradas **fila[i]** y **col[j]**.

La ilusión de que todos los leds permanecen encendidos o apagados en forma continua se logra seleccionando una columna por vez y poniendo los datos de las filas correspondientes a esa columna. La columna seleccionada se cambia cada **T_col** ms recorriendo toda la matriz. Haciendo esto cíclicamente la secuencia se repite cada $8 \times T_{col}$ ms y cada columna está seleccionada 1/8 del tiempo.



Se quiere diseñar un controlador para la matriz de leds basado en un microprocesador Z80. La imagen a desplegar inicialmente se carga durante el reset en una tabla de 8 bytes en RAM a partir de la dirección BASE, en la cual se almacenará en forma consecutiva el valor de cada columna (la columna 0 en la dirección BASE, la columna 1 en la dirección BASE+1, etc.).

Es posible ordenar al controlador que modifique la imagen desplegada insertando un nuevo valor para la columna 0 y desplazando lo que inicialmente se desplegaba en las columnas 0..6 que debe pasar a ser desplegado en las columnas 1..7. Para eso debe ponerse el valor de la nueva columna 0 en la entrada **DAT_I[8]** del controlador y dar un pulso a cero en la entrada **STB_I** para avisar al sistema que hay un nuevo valor en la entrada.

Además de actualizar su tabla interna, el controlador debe poner en las salidas **DAT_O[8]** el valor que se estaba desplegando en la columna 7 y dar un pulso a cero en la salida **STB_O** de manera que conectando N controladores en cascada se puede formar una matriz de 8 filas x 8N columnas.

Se puede suponer que la entrada **DAT_I** se mantiene estable hasta que se inserte una nueva columna. Lo mismo debe garantizarse para la salida **DAT_O**.

Se dispone de una señal **TIC** que es una onda cuadrada de período **T_col** ms.

Se debe diseñar completamente el controlador de la matriz. Para eso se pide:

- a) Agregar todo el Hardware, incluyendo memorias, necesario para manejar la matriz de leds y generar interrupciones cada **T_col** ms a partir de la señal **TIC**, trabajando en modo 1 de interrupciones.
- b) Escribir una subrutina **INS_COL** que reciba en el acumulador el nuevo valor para la columna 0, modifique la tabla en forma adecuada y devuelva en el acumulador el valor anterior de la columna 7. Salvo A y F, el resto de los registros deberán preservarse.
- c) Escribir la rutina de atención a la interrupción que actualice el estado de los leds a partir del contenido de la tabla. Si se desea puede utilizarse la subrutina **BIN2ONEHOT** que recibe un byte en el acumulador y devuelve otro con un cero en el bit indicado por el valor de los 3 bits menos significativos.
- d) Programa principal que quede en loop infinito esperando la indicación de **STB_I** para actualizar la tabla. Debe leer el nuevo valor de la columna, invocar **INS_COL** y manejar **DAT_O** y **STB_O**.
- e) Directivas de reserva de memoria y código de inicialización para que todo lo anterior funcione correctamente a partir de un RESET. Inicialmente los leds deberán estar todos apagados.