

- 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

Con la meta de analizar el desempeño de un sistema T80 funcionando en modo2 se busca cuantificar que tan frecuentemente una interrupción de prioridad alta (IRQ\_HI, atendida por la rutina ISR\_HI) interrumpe a otra de prioridad más baja (IRQ\_LO, rutina ISR\_LO). Ambas interrupciones IRQ\_HI e IRQ\_LO cuentan con sus respectivos controladores que se conectan siguiendo el protocolo daisy chain habitual para manejo de prioridades. Para cuantificar qué tanto se solapan ambas interrupciones se tomarán muestras periódicas (cada NUM períodos de reloj) de los estados de los controladores de interrupción, pero solamente en el intervalo comprendido desde el comienzo de la ejecución de ISR\_LO hasta detectarse su fin (det. fin en la figura).



Este muestreo periódico se realizará en una interrupción ISR\_TIMER generada con un Timer programable que interrumpe a través de NMI. En cada ejecución de ISR\_TIMER (líneas verticales en la figura) se incrementarán un par de contadores de 8 bits, LO\_COUNT y HI\_COUNT, si le correspondiente controlador de interrupciones está "en servicio". Se puede suponer que la duración de ISR\_TIMER es despreciable con respecto a las de ISR\_HI e ISR\_LO y que el tiempo entre interrupciones del Timer es suficientemente corto para asegurar que se producen varias interrupciones del Timer durante la ejecución de las rutinas de interrupción bajo estudio.

El Timer debe de programarse y conectarse en forma apropiada al CI de IRQ\_LO de forma que comience la cuenta al momento en que este entra en servicio. Se considerará que la rutina ISR\_LO habrá terminado la primera vez que se detecte que su controlador ya no está en estado "en servicio", momento en que el Timer deberá dejar de interrumpir hasta que comience el siguiente intervalo con un nuevo comienzo de ejecución de ISR\_LO.

Además, un flanco de subida en la entrada REPORT solicitará reportar los valores LO\_COUNT y HI\_COUNT a través de dos puertos de salida LO\_COUNT\_port[8] y HI\_COUNT\_port[8]. El sistema señalará la presencia de estos nuevos valores con un flanco de bajada en la salida READY. Una vez escritos en los puertos de salida, ambos contadores serán llevados a cero para recomenzar el proceso de medición.

Se pide:

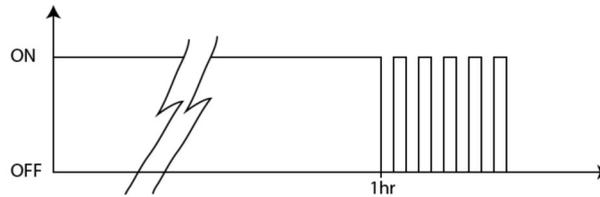
- Todo el hardware necesario para las funciones pedidas, incluyendo el conexionado de los controladores de interrupciones, el Timer y los puertos. El sistema ya cuenta con RAM (32KB), ROM (32KB) y otros puertos que dejan libre solo los primeros 16 lugares de I/O
- Rutina de atención a interrupción del timer ISR\_TIMER.
- Rutina HANDLE\_REPORT. Es invocada desde el loop del programa principal; debe manejar REPORT y READY y, cuando corresponda escribir en puertos de salida los contadores y resetearlos. Retorna inmediatamente si no hay un pedido de reporte pendiente.
- Directivas de organización de memoria, reserva de variables y código de inicialización para el correcto funcionamiento de las funciones pedidas. Este código debe terminar con la instrucción JP main.

Notas:

- $NUM = 9 \times 64 = 576$  períodos de reloj entre interrupciones del Timer
- IRQ\_HI e IRQ\_LO están asociadas a los primeros dos vectores de la tabla.
- ISR\_HI e ISR\_LO ya están implementadas; no existe posibilidad de modificarlas.
- La inicialización debe invocar la rutina existente init\_gral que inicializa al resto del sistema.

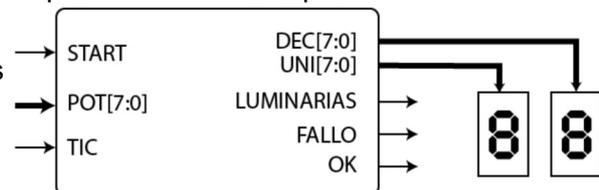
## PROBLEMA 2

Una fabrica de luminarias led debe diseñar un equipo para automatizar el ensayo al fin de fabricación. Este ensayo consiste en encender la luminaria durante 1 hora y luego realizar un ciclo de 5 encendidos y apagados (0.25 minutos en cada estado).



En el banco de ensayo se montan las luminarias y se pulsa un botón generando un flanco a 1 en la señal de entrada START. En ese momento se deberá alimentar las luminarias poniendo a 1 la salida LUMINARIAS y medir la potencia inicial consumida por ellas, la cual se comparará con la potencia monitoreada a lo largo del ensayo. Si en algún momento la potencia medida es menor al 90% de la inicial, se pondrá a 1 la salida FALLO, se apagarán las luminarias (LUMINARIAS = 0) y se dará por terminado el ensayo. La potencia se mide con un vatímetro conectado a la entrada POT de 8 bits; el valor de potencia instantánea esta disponible todo el tiempo.

Durante el ensayo se deberá mostrar en un display con 2 dígitos 7 segmentos los minutos transcurridos desde el comienzo del ensayo. Como base de tiempo se utiliza una señal TIC con flancos de subida cada 15 segundos (=1 minuto / 4).



El software se debe estructurar en un programa principal y una rutina de atención a la interrupción generada por los flancos de TIC.

La rutina de atención a la interrupción es la encargada de llevar la cuenta del tiempo transcurrido, desplegar los minutos en los displays (notar que minutos = cantidad de tics transcurridos / 4), y realizar el ciclado on/off cuando llegue el momento.

El programa principal inicializa todo lo necesario luego de un reset y espera por el flanco en la señal de START. Un vez recibida la orden de comenzar, debe encender las luminarias, medir la potencia inicial, habilitar las interrupciones y quedar en un bucle monitoreando la potencia y esperando para informar las condiciones de fin de ensayo.

Durante los ciclos on/off al final del ensayo se debe tener la precaución de no monitorear la potencia cuando las luminarias están apagadas. Si se llega al final del test sin detectar una falla debe encenderse la salida OK y dejar apagadas las luminarias.

Una vez comenzado, el ensayo no se detendrá hasta que se de una falla o se llegue al final, momento en el cual el sistema volverá a esperar un pulso en START para comenzar todo nuevamente.

Se cuenta con rutinas auxiliares ya implementadas:

RUT\_PERCENT: recibe en A y B los parámetros y devuelve en A el B por ciento del número recibido en A ( $A \leftarrow A * B / 100$ )

RUT\_BINA7SEG: recibe un numero binario positivo menor que 99 en el acumulador y devuelve en B, C las Decenas y Unidades codificados en 7 segmentos.

Se pide:

- todo el Hardware del sistema utilizando un Z80. Solo se dispone de un chip de 16KB de ROM y uno de 4KB de RAM. No deben quedar fantasmas en la decodificación de memoria. Notar que no se conoce la duración de START, por lo que debe detectarse el flanco de subida.
- rutina de atención a la interrupción.
- programa principal y reservas de memoria para variables.