

- 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 desea diseñar un sistema basado en Z80 que implemente el siguiente juego. Cuando se presiona **BOTON** por primera vez el sistema enciende la salida **LJUEGO** = 1. El jugador deberá comenzar a contar el tiempo mentalmente y presionar nuevamente **BOTON** en un instante lo más próximo posible a un tiempo **DELTA** después del encendido de **LJUEGO**. Si transcurrido un tiempo **TOUT** mayor que **DELTA** no ha llegado el segundo pulso de **BOTON**, la jugada termina por timeout.

Si la jugada termina normalmente se debe escribir en la salida **TULT** el tiempo en que se produjo el pulso y detener la cuenta para el timeout.

Si en cambio la jugada termina por timeout debe activarse la salida **LTOUT** = 1.

En ambos casos, al terminar la jugada debe apagarse **LJUEGO** y volver a esperar un pulso en **BOTON** para comenzar una nueva jugada.

LTOUT y **TULT** deben mantener su valor hasta el comienzo de la siguiente jugada, instante en que deben pasar a valer 0.

Para medir los tiempos se utilizará un timer programado para contar décimas de segundo. Los valores son **DELTA** = 100 (10 segundos) y **TOUT** = 200 (20 segundos). **TULT** es de 8 bits.

La lógica del juego se implementará totalmente por interrupciones trabajando en modo 2. Una interrupción se originará por pulsos en **BOTON**. Esta interrupción deberá iniciar o terminar la jugada, dependiendo de si se trata del primer o segundo pulso. La otra interrupción debe ser originada por el timer si transcurre **TOUT** desde el primer pulso sin que llegue el segundo. Hay otros dispositivos que interrumpen ocupando los tres primeros vectores de la tabla de interrupciones.

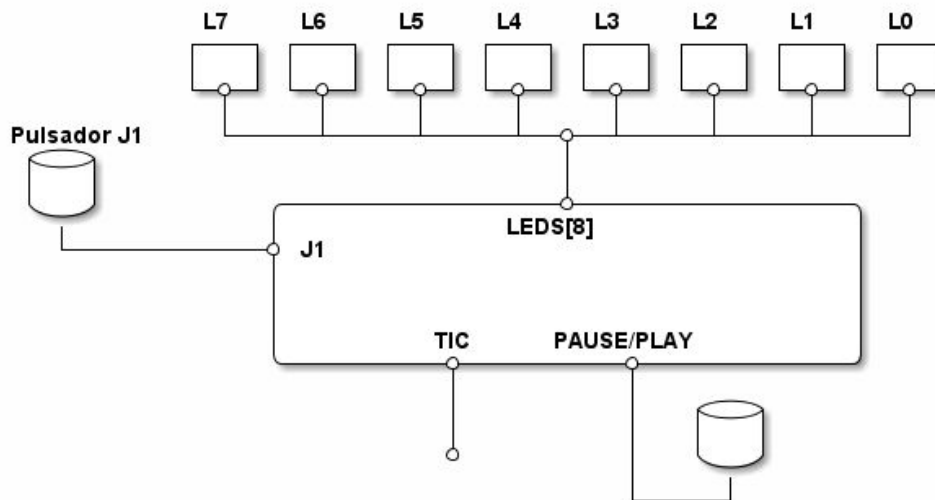
Se pide:

- Hardware. Puede utilizarse o bien un Z80 tradicional con buses tristado y un CTC, o bien un T80 con buses multiplexados y los bloques Timer y Controlador de Interrupciones usados en el curso. Todos los puertos a utilizar deben mapearse en el rango $0xC0-0xFF$ del espacio de E/S ya que el resto está reservado o usado por otros dispositivos. (**Nota:** NO es necesario incluir memorias ROM y RAM ni su decodificación.)
- Rutinas de servicio de interrupción para boton y timeout.
- Inicialización del sistema luego de RESET y directivas al ensamblador. Se debe invocar a la subrutina *init_otros* para inicializar otros dispositivos y terminar con un salto a la direccion PPAL donde comienza el programa principal que se encarga de otras tareas.

$$fck = 327680 \text{ Hz}, 10\text{Hz} = fck / 256 / 128$$

Nota para la solución con CTC: recordar que un canal del CTC programado como contador con cte=1 puede usarse para interrumpir en cada flanco de la entrada.

PROBLEMA 2



Se desea implementar una versión del clásico juego de PONG en una única dimensión y en el cual el jugador juega contra un frontón.

En este juego el jugador debe devolver una pelota representada por un único LED encendido en una tira de 8 LEDs que rebota contra el fin de la tira.

El jugador puede pegarle a la pelota presionando el pulsador **J1** solamente cuando ésta alcanza su ubicación ($L[7] = 1$, encendido). Si el jugador logra pegarle a la pelota, ésta invierte el sentido de su movimiento en dirección al frontón ($L[0]$). En cambio, si el jugador falla en pegarle a la pelota a tiempo, perderá el juego. Cuando la pelota llega al frontón ($L[0]$), invierte el sentido y avanza hacia el jugador.

El jugador también perderá el juego si intenta pegar demasiado temprano (pelota viene hacia él pero aún no alcanzó su ubicación).

El avance de la simulación del juego vendrá dado por una señal periódica externa **TIC** que interrumpirá al sistema. En cada interrupción se debe leer el estado de los pulsadores y actualizar el estado de la simulación:

- Si la pelota está en movimiento hacia el frontón, ésta avanzará a la próxima posición.
- Si la pelota ya está en el frontón, invierte la dirección de su movimiento y avanza a la próxima posición.
- Si la pelota está en la posición del jugador, y éste tiene su pulsador apretado, la pelota invertirá la dirección de su movimiento.
- Si la pelota está en la posición del jugador, y este NO tiene su pulsador apretado, el jugador perderá el juego.
- Si la pelota está en movimiento hacia el jugador, no ha llegado a su posición y tiene su pulsador apretado, el jugador perderá el juego.

Además, el juego debe poder ser pausado y resumido en cualquier momento mediante un flanco de bajada producido por el botón **PAUSE/PLAY**.

Cuando se pierda el juego, se prenderán todos los leds y el sistema quedará en ese estado hasta que se dé un **RESET**.

Luego de un reset la pelota comenzará en el medio de la cancha ($L[4]$), en dirección al frontón y pausado.

Se pide:

- a) Hardware completo del sistema basado en un Z80 con interrupciones en modo 1.
- b) Rutina de atención a la interrupción **TIC** que lleve la simulación del juego y actualice LEDs.
- c) Programa principal capaz de detectar la activación de **PAUSE/PLAY** y pausar o resumir el juego.
- d) Reserva de memoria para variables y código de inicialización luego de un reset.