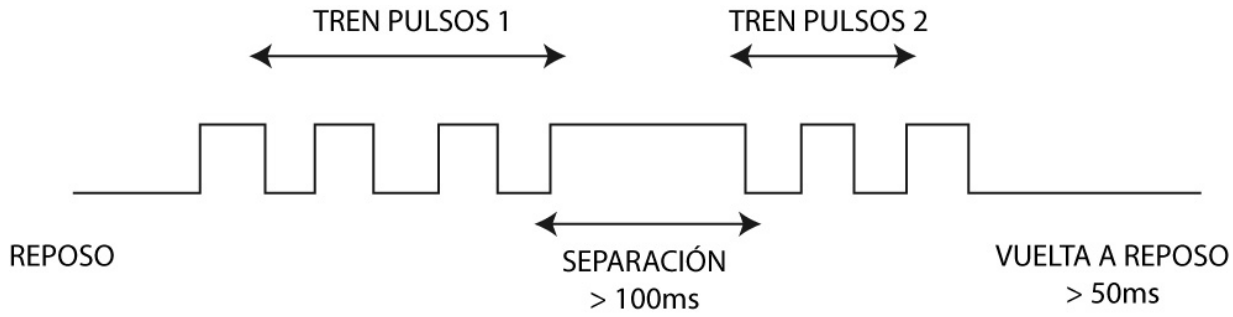
- Nombre y CI en cada hoja - Numere las hojas - Indique el total de hojas en la primera - Utilice solo un lado de las hojas	- Incluya un solo problema por hoja - **Sea prolijo** - **Aprobación:** mínimo UN problema

PROBLEMA 1

Se desea dotar a un sistema Z80 de un mecanismo que permita detectar y contar trenes de pulsos.

Los pulsos llegan por la entrada LINEA cuyo valor en estado de reposo es 0. Cuando LINEA pasa a nivel alto, debe esperarse a que llegue el primer flanco de bajada para comenzar la cuenta de los pulsos.

Cada pulso en nivel bajo debe durar menos de 50ms, sino se entiende como que la línea volvió a reposo. Períodos en nivel alto mayores a 100ms se entienden como separación entre 2 trenes de pulsos.



Para contar el tiempo se deberá programar un CTC para que interrumpa cada 1ms y una rutina de atención a interrupciones actualizará el valor de una variable en memoria que será leída por la rutina encargada de contar los pulsos. La frecuencia del reloj del sistema es 4MHz

La cuenta de los pulsos se realizará con la subrutina CUENTA_PULSO que será invocada cuando LINEA sale de reposo (pasa a 1). CUENTA_PULSO retornará devolviendo en el registro C la cantidad de pulsos contados y en B un código que indica el motivo de retorno: 00h la línea volvió a reposo, 01h finalizó el tren de pulsos y se detectó un separador.

La subrutina debe esperar la primer bajada a cero (sin importar cuánto demore) y a partir de ese momento contar los pulsos como se explicó más arriba.

a. El sistema ya cuenta con 32K de ROM y 32K de RAM y diversos puertos de entrada y salida entre las direcciones 00h y 7Fh. Solo se pide el puerto de entrada para leer el valor de LINEA y la conexión al CTC.

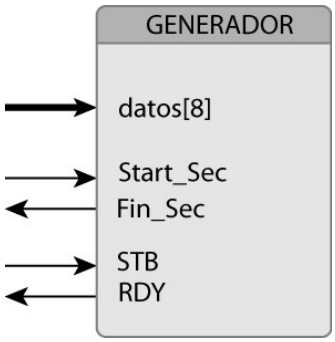
b. Inicialización del sistema luego de un reset, incluyendo el stack, CTC y variables necesarias. También debe inicializarse todo el sistema para trabajar en modo 2 de interrupciones, están disponibles los primeros 4 lugares de la tabla de interrupciones para ser utilizados para el CTC. Debe terminar con un salto a la dirección PPAL.

c. Rutina de atención a la interrupción y subrutina CUENTA_PULSO

PROBLEMA 2

Para testear un chip se le envía una serie de datos conocidos y se evalúa si las respuestas son las esperadas. En una prueba completa de un dispositivo hay varias secuencias que se repiten muchas veces, llamadas “secuencias frecuentes” y otras no tan usuales. La forma de realizar esta prueba es indicarle a un dispositivo GENERADOR qué secuencia frecuente debe enviar al chip, o sino leer datos de memoria y pasárselos al chip bajo prueba, palabra a palabra, con cierto handshake.

Las secuencias frecuentes se identifican con un número que va del 1 al 127. Existe un dispositivo HW, llamado GENERADOR, capaz de generar estas secuencias. En su entrada **datos[8]** se debe poner el número correspondiente a la secuencia que se quiere reproducir y darle un flanco de subida en su entrada **Start_Sec**. Cuando GENERADOR termina de pasarle toda la secuencia al chip bajo prueba, da un pulso corto en su salida **Fin_Sec**. La señal **Fin_Sec** se utilizará para interrumpir al Z80.
Por otro lado, cuando no hay secuencias frecuentes, el mismo dispositivo GENERADOR, si recibe un pulso en su entrada **STB**, lee el dato presente en su entrada **datos[8]** y lo pasa tal cual al chip bajo prueba, poniendo en 1 su salida **RDY** cuando está listo para la próxima palabra. **RDY** baja a cero con cada pulso en **STB** o de **Start_Sec**.



En ROM, a partir de la dirección 4000h, se encuentra precargado un listado de palabras de 1 o 3 bytes que indican como llevar a cabo el test.
La palabra **00h** está reservada e indica el fin de la tabla. Para el resto de los casos, el bit más significativo indica como interpretarla:

- bit7=0: debemos reproducir una secuencia frecuente
  - los 7 bits menos significativos son el número de secuencia a reproducir.
- bit7=1 debemos pasar datos de memoria uno a uno.
  - los 7 bits menos significativos contienen la cantidad de bytes a enviar.
  - los 2 bytes siguientes contienen la dirección de donde deben leerse estos bytes (primero esta almacenado el byte más significativo de la dirección).

Se deberá escribir la subrutina **RUTIN_TEST** encargada del test que leerá el listado de palabras y procederá como se explica a continuación.
En caso de leer una palabra con bit7=1, se debe enviar uno a uno la cantidad de bytes al GENERADOR, respetando el handshake de **STB** y **RDY**. Una vez enviados todos los bytes, se debe pasar a procesar la siguiente palabra de la tabla que especifica el test.
En caso de leer una palabra con bit7=0, se debe mandatar al GENERADOR para que ejecute la secuencia frecuente y se debe retornar al programa principal. Una vez que el generador termine, se generará una interrupción que deberá volver a llamar a esta subrutina de forma de seguir con el test.
En caso de leer la palabra 00h, se retorna ya que el test finalizó. En la tabla hay al menos un test para realizar.

Se desea implementar el sistema descripto anteriormente, para esto se dispone de un Z80 con 48K de ROM (3 chips de 16k c/u) y 16K de RAM y un dispositivo GENERADOR externo como el descrito.

a. Dibuje todo el HW necesario y su conexionado: Z80, memoria, dispositivo GENERADOR, puertos, etc.
b. Escriba la subrutina RUTIN_TEST y subrutina de atención a la interrupción.
c. Código a ejecutar luego de un reset que:

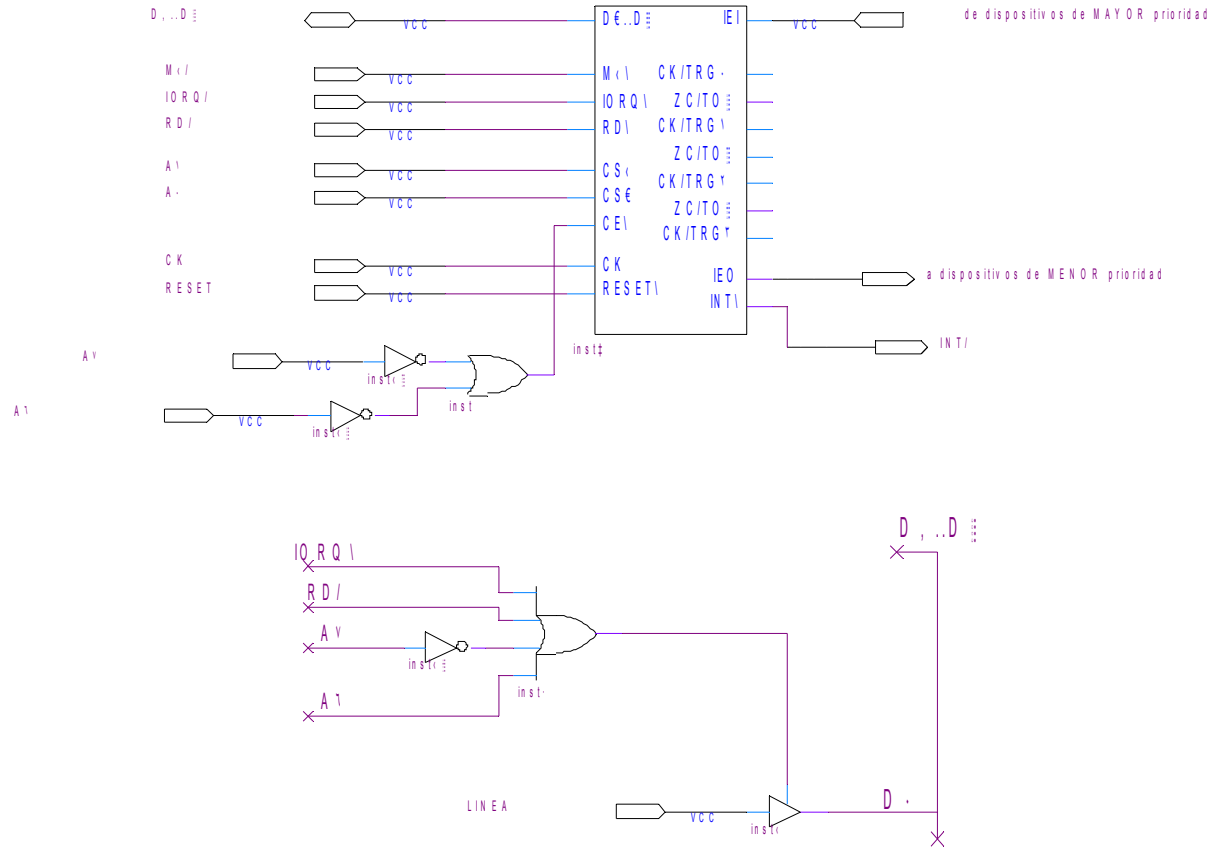
- Inicialice el sistema: inicialización de variables, stack e invoque la subrutina INIT_OTROS() encargada de inicializar el resto del sistema
- Realice la primera invocación a RUTIN_TEST. Al retornar se debe ejecutar JP PPAL.

Sugerencia: utilizar una variable PUNTERO que se inicializa en 4000h antes de invocar por primera vez a la subrutina RUTIN_TEST. Esta variable apuntará a la próxima palabra a procesar en la tabla

que contiene el test.

SOLUCIÓN PROBLEMA 1

a) Hardware



b) Inicialización

```
LINEA EQU 80h

CTC0 EQU C0h
CTC0_CTE EQU 250 ;1ms = 250ns * 16 * 250
CTC0_CW EQU 100X0111
CTC0_VW EQU 00h

org EN_ROM
TABLA_INT: DW RUT_TIMER

org 8000h
CONT DB
```

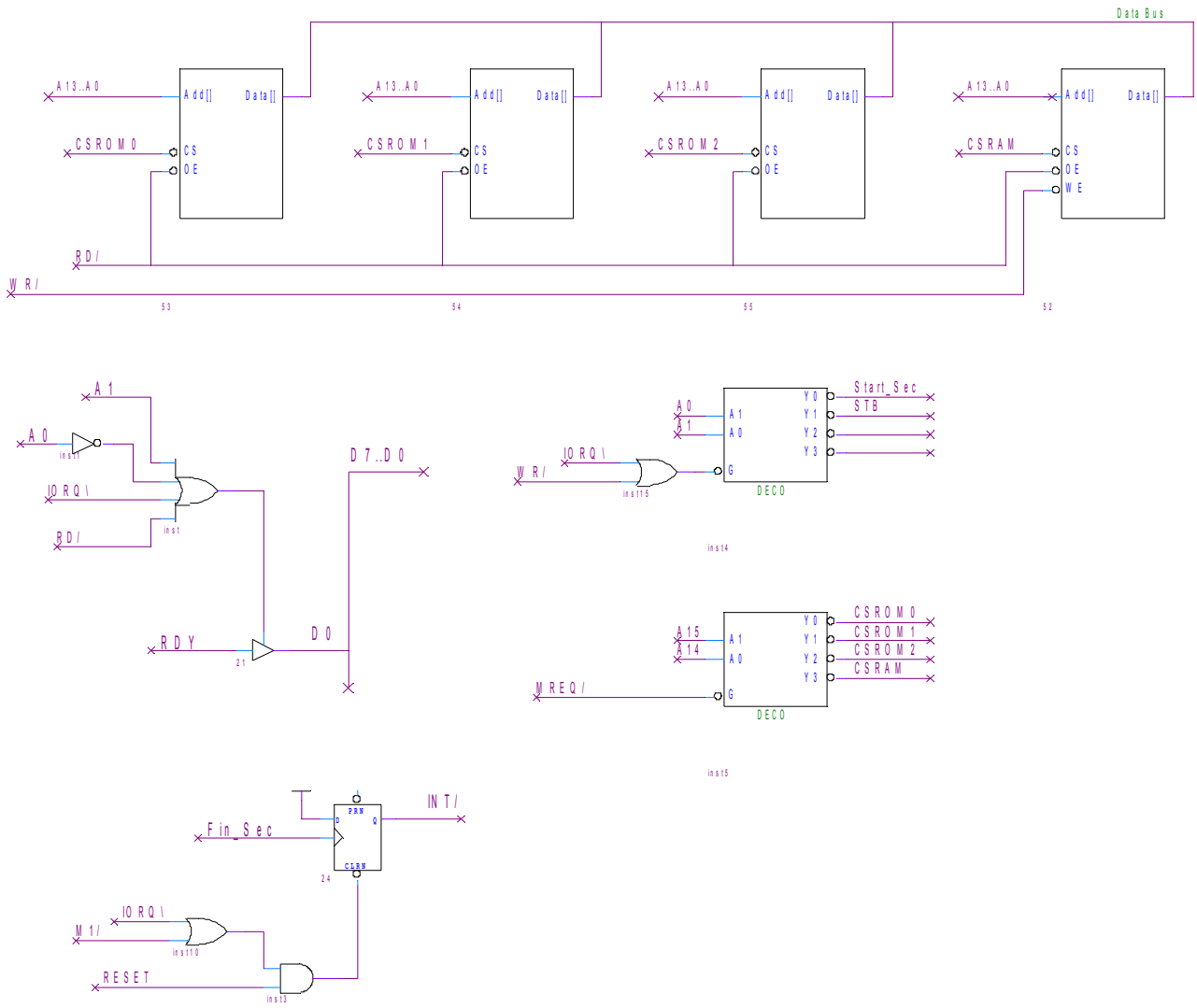
```
org 0000h
ld SP, 0000h
ld A, TABLA_INT/256
ld I, A

ld A, CTC0_VW
out (CTC0), A
ld A, CTC0_CW
out (CTC0), A
ld A, CTC0_CTE
out (CTC0), A

EI

jp Ppal
```

c) Subrutina cuenta_pulso y rutina de atención a interrupciones.

```
org EN_ROM
CUENTA_PULSO:
  pusf AF
  ld C,0 ; cuenta de pulsos

ESPERO_PULSO
  in A, (LINEA)
  bit 0, A
  jp NZ, ESPERO_PULSO
  ld A, 0
  ld (CONT), A

LINEA_0:
  in A, (LINEA)
  bit 0, A
  jp Z, MIRO_CONT_0
  inc C
  ld A, 0
  ld (CONT), A
  jp LINEA_1

MIRO_CONT_0:
  ld A, (cont)
  cp 50
  jp Z, REPOSO
  jp LINEA_0

LINEA_1:
  in A, (LINEA)
  bit 0, A
  jp NZ, MIRO_CONT_1
  ld A, 0
  ld (CONT), A
  jp LINEA_0

MIRO_CONT_1:
  ld A, (cont)
  cp 100
  jp Z, NUEVO_TREN
  jp LINEA_1

REPOSO:
  ld B, 00h
  jp FIN

NUEVO_TREN:
  ld B, 01h

FIN:
  pop AF
  ret

;--------- RUTINA INTERRUPCIONES
org EN_ROM
RUT_TIMER:
 ei
 push AF
 ld A, (CONT)
 inc A
 ld (CONT), A
 pop AF
 reti
```

SOLUCIÓN PROBLEMA 2

a)



La entrada datos[8] del generador se conecta al bus de datos, las demás señales se conectan como se indica en el circuito.

b)

```
Start_Sec EQU 00h
STB       EQU 01h
RDY       EQU 01h

TABLA_TEST EQU 4000h

org C000h
PUNTERO_TABLA_TEST DW

org ALGUN_LUGAR_ROM
RUTIN_TEST:
   push AF
   push HL
   push BC
   push DE
```

```
LEO_TABLA:
   ld HL, (PUNTERO_TABLA_TEST)
   ld A, (HL)
   cp A, 00
   jp Z, FIN

   bit 7, A
   jp Z, SECUENCIA

PALABRA:
   res 7,A
   ld B,A   ; B = cantidad de bytes a enviar
   inc HL
   ld A, (HL)
   ld D, A
   inc HL
   ld A, (HL)
   ld E, A   ; DE = dirección de comienzo
   inc HL
   ld (PUNTERO_TABLA_TEST), HL
ESPERO_RDY:
   in A, (RDY)
   bit 0, A
   jp Z, ESPERO_RDY
   ld A, (DE)
   out (STB), A
   inc DE
   djnz, ESPERO_RDY
   jp LEO_TABLA

SECUENCIA:
  out (Start_Sec),A
  inc HL
  ld (PUNTERO_TABLA_TEST), hl

FIN:
  pop DE
  pop BC
  pop HL
  pop AF
  RET

org 38h
  call RUTIN_TEST
  ei
  ret

c)
org 0000h
  ld SP, 0000
  ld HL, TABLA_TEST
  ld (PUNTERO_TABLA_TEST), HL
  call INIT_OTROS
  ei
  call RUTIN_TEST
  jp PPAL
```