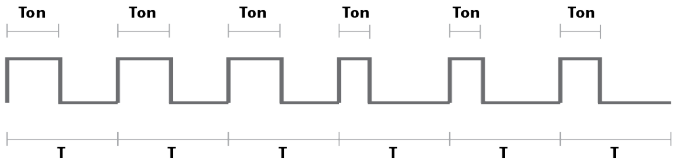
• Nombre y CI en cada hoja • Numere las hojas, indique el total en la primera • Utilice sólo un lado de las hojas	• Incluya un solo problema por hoja • **Sea prolijo** • **Aprobación:** mínimo UN problema

PROBLEMA 1

Se desea diseñar una fuente de corriente pulsada para leds, con la capacidad de protegerlos contra sobre temperatura.

Los leds son alimentados con una fuente que genera pulsos de corriente a una cadencia constante T de 65,280 ms. Para variar la intensidad de luz que generan, se modifica el tiempo en que el pulso esta en 1 (Ton) con respecto al tiempo en 0.



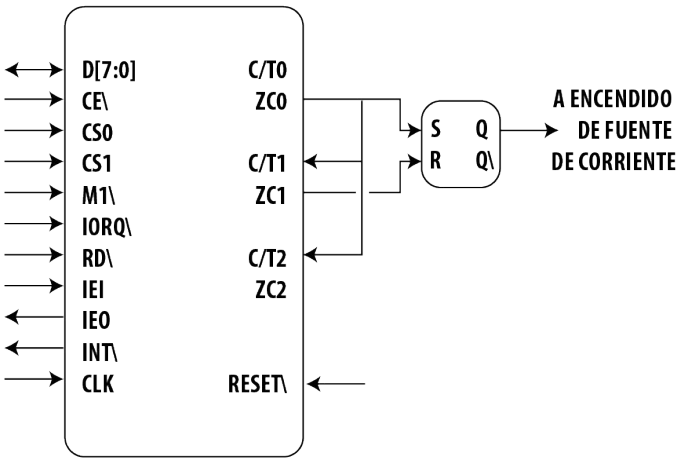
Un flanco de subida en la entrada NEW_VALUE indica que hay un nuevo valor de Ton en la entrada VALUE de 8 bits. Este estará expresado en múltiplos de 256us y se asegura que nunca será mayor a 240.

Cuanto más ancho son los pulsos en 1 (Ton), más luz emitirán pero también calentarán más. La fuente deberá proteger los leds en caso de que se supere la temperatura UMBRAL.
La temperatura de los leds está siempre disponible en la entrada TEMP de 8 bits y debe muestrearse cada 250 períodos T del tren de pulsos. En caso de que la temperatura supere el valor UMBRAL, los siguientes pulsos deberán utilizar Ton/2 hasta que se vuelva a muestrear TEMP~~Ton~~. Si al cabo de este tiempo la temperatura sigue siendo superior, se mantiene el régimen de trabajo de Ton/2, en caso contrario se vuelve a trabajar con Ton.
Tener en cuenta que se mantiene la premisa de que Ton puede ser actualizada en cualquier período.

El hardware para el manejo de la fuente de corriente esta dado y es el siguiente:

Como se ve, la ZC del canal 0 del CTC es utilizada para:

- poner a 1 el FF que habilita la fuente de corriente
- para disparar el canal 1 que deberá contar Ton. De esta forma se logra que ambos canales estén sincronizados.
- para alimentar el canal 2 que deberá llevar la cuenta del intervalo en el que se chequea la temperatura

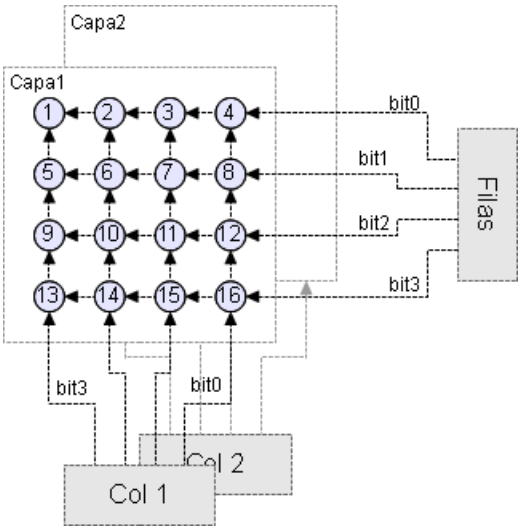


El reloj del sistema es de 1MHz. Notar que 1/1MHz * 256 = 256us y que 65,280 ms = 256us * 255

Se pide:
a) resto del hardware del sistema (puertos, memoria y decodificación de direcciones)
b) programa principal que inicialice todo el sistema y luego monitoree mediante polling NEW_VALUE y actualice Ton. Luego de un reset Ton = 128
c) rutinas de atención a la interrupción que generan los canales 1 y 2 del CTC.

PROBLEMA 2

Se desea implementar el software y el hardware para controlar un “cubo” de leds. El “cubo” consiste en una matriz de leds en tres dimensiones de [4 x 4 x 2] (filas, columnas, capas).



Las señales **Filas, Col1** y **Col2** (hay una señal columna de 4 bits para cada capa) son de 4 bits activas por nivel alto.

Los bits de **Filas** se comparten con las filas de ambas capas. De esta manera si Filas=0001b y las señales columna son: Col1=1000b y Col2=0001b, solo se encenderán los leds 1 de la capa 1 y 4 de la capa 2.

Dir	Mem [7..4]	Mem [3..0]	Fila
tbl_cubo	Col1	Col2	1
tbl_cubo+1	Col1	Col2	2
tbl_cubo+2	Col1	Col2	3
tbl_cubo+3	Col1	Col2	4

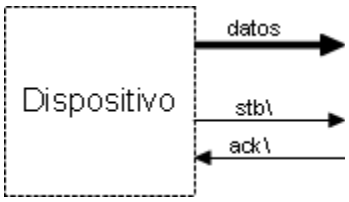
Los datos a mostrar se encuentran en una tabla en memoria de 4 lugares a partir de la dirección **tbl_cubo** y están organizados como se muestra en la tabla.

Los 4 lugares de la tabla contienen los valores que deben tomar las señales Col1 y Col2 para cada una de las 4 filas.

Para cambiar la fila desplegada existe una interrupción periódica **TIC**, siendo esta la única interrupción al sistema. En cada una de las interrupciones se debe desactivar las filas, actualizar el valor de Col1 y Col2 y volver a activar la fila que corresponde. La tabla se recorre en forma cíclica, y en cada interrupción se actualiza el valor de una sola de las filas.

El sistema se deberá conectar a un dispositivo por el cual le llegarán comandos y datos para actualizar la tabla. Este dispositivo avisa que hay un dato disponible en **datos** bajando la señal **stb**, la cual se mantiene hasta que se de un pulso a 0 en la señal **ack**. Los bytes recibidos podrán ser comandos o desplazamientos dentro de la tabla seguidos del byte a guardar.

Comando	Acción
F0h	Apagar todos los leds el cubo
F1h	Reanudar encendido de leds
00h a 03h	Desplazamiento dentro de la tabla a Actualizar (Se recibe el dato en el siguiente byte)
Otros	Ignorar



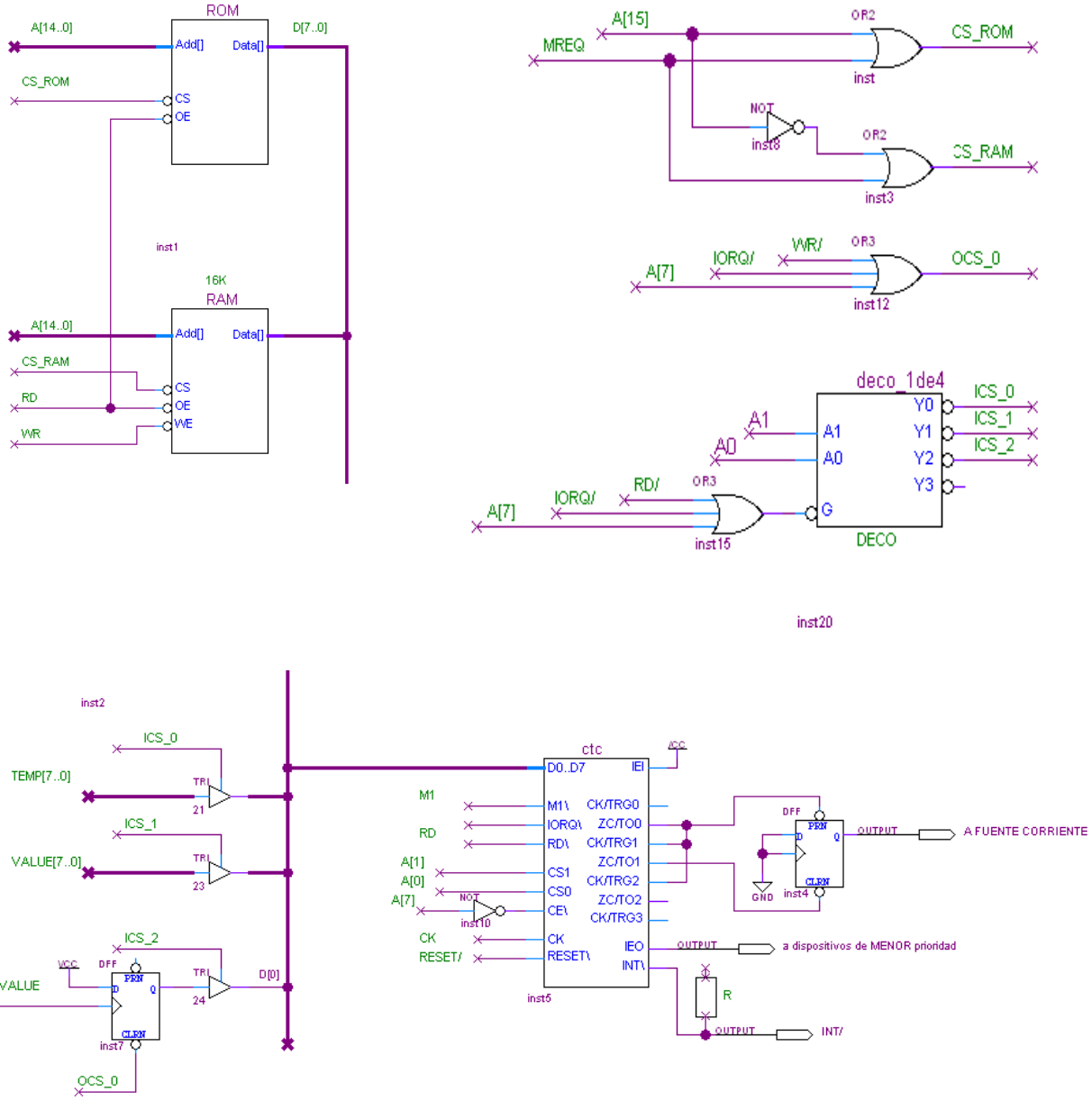
Se pide:

a) Hardware completo del sistema (puertos, memoria y decodificación). Tener en cuenta que hasta que no llegue la siguiente interrupción, se deben mantener los datos en el cubo.
b) Rutina de atención a la interrupción que maneja el cubo.
c) Programa principal, que inicializa el sistema y luego recibe los comandos y actualiza la tabla.

** La tabla debe inicializarse con todos los leds encendidos.

PROBLEMA 1 – Solución

a) Hardware
- memoria: 32K + 32K
- CTC conectado como se indica en la letra.
  . Canal 0 modo timer no interrumpe.
  . Canal 1 modo timer, trigger hw, interrumpe para reprogramar canal 1.
  . Canal 2 modo counter, interrumpe para muestrar temp y setear bandera.
- puertos entrada: VALUE[8], NEW_VALUE con FF, TEMP[8]



*** Faltan dos en las señales que manejan PRN y CLRN ya que las salidas ZC/TO son activas en nivel alto.

b)

```
CTC0_CW  equ 0010 0111  ;canal 0: sin interrupciones, timer, prescaler 256,
                        ; No importa, Trigger con cte, sigue constante,
                        ; software reset, palabra de control
CTC0_CTE  equ 255       ; cte canal 0 = 255. 256uS*255 =65.280ms
CTC1_CW   equ 1011 1111  ;canal 1: con interrupciones, timer, prescaler 256,
                        ; Rising edge, Trigger con pulso, sigue constante,
                        ; reset, palabra de control
CTC1_CTE equ 128        ; luego de reset Ton = 128
CTC2_CW  equ 1101 0111  ;canal 2: con interrupciones, counter, No importa,
                        ; Rising edge, No importa, sigue constante,
                        ; software reset, palabra de control
CTC2_CTE equ 250
CTC_VW   equ 00h        ; palabra de control de CTC2=04h
UMBRAL   equ            ; constante a especificar

org 8000h
Ton: db
Temp_ok: db  ; 0ffh si ok, 00h si se debe usar Ton/2

org 4000h
tabla_int:   dw                     ; CTC0 (00h)
             dw rutint_canal1       ; CTC1 (02h)
             dw rutint_contador     ; CTC2 (04h)

org 0000h
  ; iniciliazo SP
  ld SP, 0000h
  ; tabla e interrupciones modo 2
  ld A, tabla_int/256
  ld I, A
  IM 2
  ld A, CTC0_CW
  out (CTC0), A
  ld A, CTC0_CTE
  out (CTC0), A
  ld A, CTC1_CW
  out (CTC1), A
  ld A, CTC1_CTE
  out (CTC1), A
  ld A, CTC2_CW
  out (CTC2), A
  ld A, CTC2_CTE
  out (CTC2), A
  ld A, CTC_VW
  out (CTC0), A
  ; inicializo puertos y variables
  in A,(NEW_VALUE)  ; para borra FF que indica NEW_VALUE
  ld A, 128
  ld (Ton), A
  ld A, 0FFh
  ld (Temp_ok), A
; loop de polling de nuevo Ton
; en caso de un nuevo Ton, actualiza variable. La rutina de atención a
; CTC1 se encarga de actualizar el CTC con el nuevo valor
loop_Ton:   in A,(NEW_VALUE)
  bit 0,A
  jp Z, loop_Ton
  out (CLRFF), A
  in A, (VALUE)
  ld (Ton), A
  jp loop_Ton
```

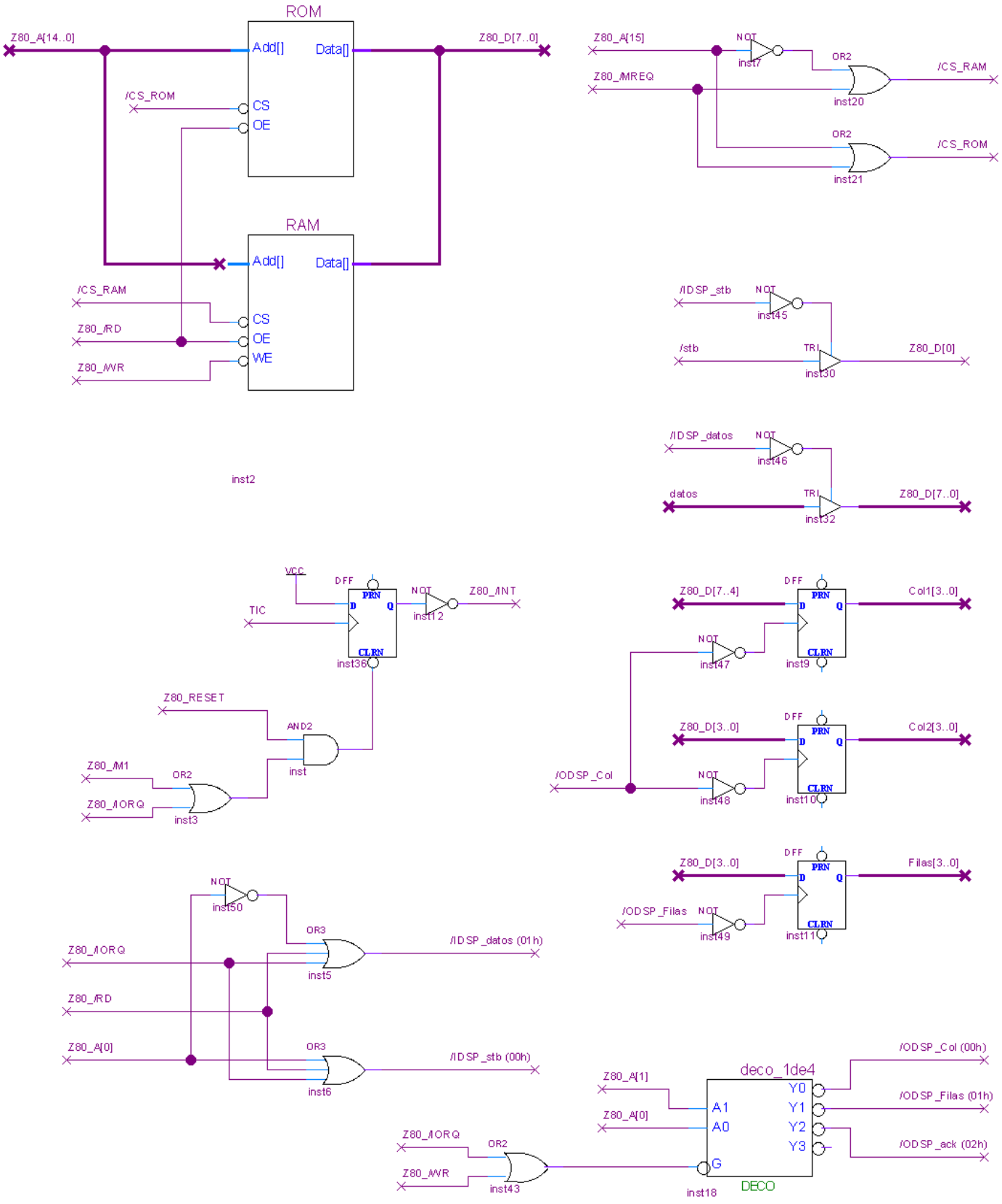
c)

```
org 5000h
rutint_contador:
      ei
      push af
      in A, (TEMP)
      cp UMBRAL
      jp P, temp_mayor_a_umbral
temp_menor_a_umbral:
      ld A, 0FFh
      jp fin
temp_mayor_a_umbral:
      ld A, 00h
fin:        ld (Temp_ok),A
      pop af
      reti

rutint_Ton:
      ; cada vez que el canal 1 termina de contar Ton
      ; debo programarlo para que espere el flanco de trigger para contar
      ei
      push af
      push bc
      ld A, (Temp_ok)
      cp A, 0FFh
      ld A, (Ton)            ; A=Ton
      jp NZ,  reprogramo_ctc ; su TEMP no era > UMBRAL, se reprograma con Ton
      SRL A   ; divido Ton   ; TEMP>UMPRAL se usa Ton/2
reprogramo_ctc:
      ld B,A
      ld A, CTC1_CW
      out (CTC1), A
      ld A,B
      out (CTC1), A
      pop bc
      pop af
      reti

; Puertos
      TEMP        equ 0
      VALUE       equ 1
      NEW_VALUE   equ 2
      CLRFF       equ 0
      CTC0        equ 0x80
      CTC1        equ CTC0+1
      CTC2        equ CTC0+2
```

PROBLEMA 2 – Solución

a) Hardware

32Kb de ROM
32Kb de RAM



b) Rutina de Atención a la Interrupción

```
        ORG     38h             ; Rutina en 38h cuando estoy en MODO1

        PUSH    AF              ; Preservo los registros a usar
        PUSH    HL

        LD      HL,tbl_cubo     ; Inicializo el puntero a la tabla tbl_cubo

        LD      A,00h           ; Apago las filas (activas nivel alto)
        OUT     (io_fil),A

        LD      A,(estado)      ; Leo el estado para saber si tengo que apagar el cubo o no
        CP      01h
        JP      NZ,fin          ; Si estado = 00h hay que apagar el cubo (ya apague las filas, me voy)

        LD      L,(cont_int)    ; Obtengo el contador de interrupciones
        LD      A,(HL)          ; Cargo en A el dato de la tabla tbl_cubo
        OUT     (io_col),A      ; Actualizo el latch de columna

        LD      A,(fila)        ; Obtengo la fila actual
        OUT     (io_fil),A      ; Actualizo el latch de fila

        RLC     A               ; Me muevo a la fila siguiente
        CP      10h             ; Si llegue a 10h = 10000b me pase
        JP      NZ,sigo1        ; Si no llegue a 10h salto

        LD      A,01h           ; Reinicio el dato de filas con 01h = 0001b

sigo1:
        LD      (fila),A        ; Actualizo la variable filas
        LD      L, (cont_int)   ; Obtengo el contador de interrupciones
        INC     L               ; Incremento el contador
        LD      A,L
        CP      04h             ; Si pasaron 04h interrupciones
        JP      NZ,sigo2        ; Si no llegue a 4 interrupciones salto
        LD      A,00h

sigo2:
        LD      (cont_int),A    ; Actualizo el contador de interrupciones

fin:
        POP     HL              ; Recupero los registros preservados
        POP     AF

        EI
        RETI                    ; Retorno de la interrupcion
```

c) Programa Principal

```
; ----------------
; Constantes
; ----------------
io_dat  EQU     01h             ; Direccion del puerto de datos
io_stb  EQU     00h             ; Direccion del puerto de la señal stb
io_ack  EQU     02h             ; Direccion del puerto de la señal ack

io_col  EQU     00h             ; Direccion del puerto para las columnas
io_fil  EQU     01h             ; Direccion del puerto para las filas

; -------------------------
; Variables y tabla en RAM
; -------------------------
        ORG     8000h
tbl_cubo:       DS      4       ; Reservo 4 lugares de memoria para la tabla
                                ; lo hago acá para que quede alineado a una dirección terminada en 00h

estado:         DB              ; Bandera que indica si se muestra o no la animacion
cont_int:       DB              ; Variable que cuenta la cantidad de interrupciones
fila:           DB              ; Variable que alamacena la fila a mostrar


; ----------------
; Programa en ROM
; ----------------
        ORG     0000h
        JP      inicio          ; Voy al programa principal

        ORG     0400h           ; Dejo espacio para la rutina de atencion a las INT

inicio:
        LD      SP,0000h        ; Configuro el SP
        LD      A, 0
        LD      (cont_int),A    ; Inicializo el contador de interrupciones
        LD      A, 01h
        LD      (fila),A        ; Inicializo la fila a mostrar
        IM1                     ; Modo de interrupciones
        LD      HL,tbl_cubo
        LD A, 0FFh
        LD B, 4
loop_ini:                       ; Inicializo tabla con todos los leds encendidos
        ld (HL), A
        DJNZ
        LD      HL,tbl_cubo
        EI                      ; Habilito las interrupciones


loop1:  IN      A,(io_stb)      ; Leo la señal stb
        AND     01h             ; Mascara (Uso el bit 0 para leer stb)
        JP      Z,loop1         ; Mientras recibo un 1 espero

        IN      A,(io_dat)      ; Leo el comando recibido
        OUT     (io_ack),A      ; Doy un pulso a 0 en ack, ya recibi el comando
```

```
        CP      F0h             ; Si es F0 hay que detener la animacion
        JP      NZ,sigue1
        LD      A,00h
        LD      (estado),A      ; Pongo la bandera de estado en 00h
       JP       loop1

sigue1:
        CP      F1h             ; Si es F1 hay que reanudar la animacion
        JP      NZ, sigue2
        LD      A,01h

        LD      (estado),A      ; Pongo la bandera de estado en 01h
        JP      loop1

sigue2:
        CP      03h             ; Hago "datos - 03h"
        JP      P,loop1         ; Si es positivo "datos > 03h", ignoro el comando

        LD      L,A             ; Como esta entre 00h y 03h tengo un desplazamiento en A, lo cargo en L

loop2:  IN      A,(io_stb)      ; Leo la señal stb
        AND     01h             ; Mascara (Uso el bit 0 para leer stb)
        JP      Z,loop2         ; Mientras recibo un 1 espero


        IN      A,(io_dat)      ; Recibo el valor a cargar en la tabla
        OUT     (io_ack),A      ; Doy un pulso a 0 en ack, ya recibi el dato
        LD      (HL),A          ; Cargo el dato recibido en la tabla

        JP      loop1           ; Vuelvo al inicio
```