

# Tema 7. Entrada/Salida

## Ejercicios

### 1 Mapeo de interfaces

Una interfaz de vídeo similar a la vista en clase, se conecta al CT mapeada a partir de la dirección A000h. ¿Cuántas líneas de entrada tendrá el circuito de activación asociada a dicha interfaz? ¿Y cuántas líneas de datos?

### 2 Mapeo de interfaces

Se ha conectado un periférico de tipo Pantalla (mapea 128 posiciones, 8x15) como el visto en clase al CT. La interfaz de E/S asociada a dicho periférico ocupa las posiciones de memoria más altas del espacio de direcciones. ¿En qué posición de memoria se encuentra mapeada la primera celda de la segunda fila de dicha pantalla?

### 3 Mapeo de interfaces

En el espacio de direcciones del CT se han mapeado, uno a continuación del otro, una interfaz de tipo Teclado (mapea 2 elementos) y una interfaz de tipo Pantalla (mapea 8x15 posiciones de memoria y a continuación el registro de control), ambos como los vistos en clase. Sabiendo que el registro de control asociado a la pantalla está mapeado en la dirección de memoria AB78h, dibujar el circuito de activación asociado al Teclado. Etiquetar claramente todas sus entradas.

### 4 Muestreo periódico

Se ha conectado al Computador Elemental un periférico cuya interfaz de E/S tiene 4 registros asociados, mapeados a partir de la dirección 2004h. Los tres primeros son registros de datos (reg. resultado, reg. dato1 y reg. dato2) y el último es un registro de estado/control (reg. E/C). El bit de

peso 0 del registro de E/C se pone a 1 si está disponible el primer dato, el bit de peso 1 se activa si está disponible el segundo dato, y el bit de peso 15 se pone a 1 en caso de que se produzca algún error.

Se muestra a continuación un fragmento de un programa de E/S programada, en el que se sale del bucle de sincronización bien porque se produzca un error, o bien porque algunos de los datos (dato1 y dato2) estén disponibles.

```
...
.CODIGO

--- HUECO 1 ---

BucleSincro:
    mov r1, [r0]
    and r2, r1, r3
    brz BucleSincro

; Comprobar si error o datos disponibles
movl ????, 0 ; - Hueco 2a -
???? r2, r1, r3 ; - Hueco 2b -
???? fuera ; - Hueco 2c -

; Tratamiento de los datos disponibles
dec r0
mov r2, [r0]
dec r0
mov r1, [r0]
add r1, r1, r2
dec r0
mov [r0], r1

fuera:
    fin
```

4.1 ¿Cuántas líneas de entrada tendrá el circuito de activación asociado a este periférico?

4.2 ¿Qué instrucciones faltan en ---HUECO 1---

- 4.3 Completar las instrucciones marcadas como **Hueco 2a**, **Hueco 2b** y **Hueco 2c**, sustituyendo los interrogantes por lo que sea necesario para que el código finalice, en caso de que se haya producido un error, o trate los datos disponibles en caso contrario

- 5.3 Suponiendo que en R3 está la dirección del registro de control, escribir las instrucciones necesarias para abrir la puerta.

### 5 Muestreo periódico

Se ha conectado al CT un dispositivo lector de tarjetas de acceso que permite controlar la apertura de puertas a edificios. La interfaz de este dispositivo está compuesta por tres registros, todos de 16 bits: un registro de estado, uno de control y uno de datos, mapeados en este orden a partir de la dirección de memoria ABCDh. Cuando se pasa una tarjeta por el dispositivo lector, éste pone un 1 en el bit de peso 2 de su registro de estado, y escribe el número de tarjeta en el registro de datos. Cuando la CPU lee el registro de datos, se pone a 0 el bit de peso 2 del registro de estado. Cuando la CPU escribe un 1 en el bit de peso 1 del registro de control y un 0 en el resto de los bits, el dispositivo abre la puerta. Si la CPU escribe en 1 en el bit de peso 0 del registro de control, entonces la puerta no se abrirá.

- 5.1 Escribe el fragmento de código que compruebe si se ha pasado una tarjeta por el dispositivo.

- 5.2 Suponiendo que se sabe que se ha pasado una tarjeta, escribir las instrucciones necesarias para leer el número de la tarjeta y dejar el resultado en el registro R5.

## 6 Interrupciones

Se ha escrito un programa para el Computador Teórico que, antes de entrar en la ejecución de un bucle infinito, instala en memoria dos rutinas de interrupción, una asociada a un periférico de tipo Teclado y otra asociada a un periférico de tipo Luces como los vistos en clase. Se sabe que el hueco marcado en el código como Hueco ❶ equivale a 9 instrucciones, el hueco marcado como Hueco ❷ a 18 instrucciones, y el hueco marcado como Hueco ❸ a 27 instrucciones. Se incluye a continuación el esqueleto básico de dicho programa, sin especificar el detalle de cada una de las rutinas.

```
ORIGEN 1000h
INICIO primera
.PILA 10
.DATOS
posicionPantallaVALOR 0
colorCaracter VALOR 3900h

.CODIGO

primera:
; Instalacion de rutinas en memoria
--- Hueco ❶

STI
JMP -1

PROCEDIMIENTO RutLuces
--- Hueco ❷

IRET
FINP

PROCEDIMIENTO RutTeclado
--- Hueco ❸

IRET
FINP
FIN
```

El contenido de las primeras 254 posiciones de memoria es 0000h, y el de las cuatro posiciones siguientes se especifica en la siguiente tabla:

Dirección	Contenido
00FEh	100Dh
00FFh	1020h
0100h	0000h
0101h	0000h

- 6.1 ¿Cuál es el número del vector de interrupción asociado al periférico Luces? Responder en hexadecimal

- 6.2 Si al inicio del paso de ejecución nº 4 de una determinada instrucción PC vale 103Ch, ¿cuál será el próximo valor que se desapile? Responder en hexadecimal

## 7 Interrupciones

La CPU elemental está ejecutando la instrucción INC R0 cuando recibe una petición de interrupción asociada al vector de interrupción número 4. La interrupción se acepta y en los buses del computador aparece la siguiente secuencia de valores:

SCB	SAB	SDB
WRITE	0C08	0017
WRITE	0C07	0B44
INTA	----	X
READ	X	0444

- 7.1 ¿En qué posición de memoria está almacenado el código máquina de la instrucción INC R0?

- 7.2 ¿Cuál es el valor de X? Responder en hexadecimal.

## 8 Interrupciones

Se ha realizado el siguiente programa en lenguaje ensamblador del CT sin ningún cometido en particular. Se sabe que durante la ejecución del programa sólo se realizan dos peticiones de interrupción, justo en el momento en el que la CPU se encuentra ejecutando por primera vez la instrucción `dec r2`.

```

origen 1000h
inicio ini
.pila 10h
.codigo

procedimiento RutInt
    iret
finp

ini:
    xor r0, r0, r0
    movl r1, bytebajo direccion RutInt
    movh r1, bytealto direccion RutInt
    movl r2, 0
    movh r2, 1

bucle:
    mov [r0], r1
    inc r0
    dec r2
    brnz bucle

    sti
    jmp -1

fin
    
```

8.1 ¿Cuáles serán las siguientes DOS instrucciones que se ejecutarán tras la ejecución de la instrucción `sti`? Responder con los mnemónicos

8.2 ¿Cuál será la dirección de retorno de la rutina de interrupción? Responder en hexadecimal.

8.3 ¿Qué valor habrá almacenado en la dirección de memoria 00FFh durante la ejecución del programa?

## 9 Entrada/Salida

Se han conectado dos periféricos al Computador Teórico: Teclado y Luces. Al conectar el Teclado no se ha habilitado la posibilidad de generar interrupciones, pero sí al conectar Luces. El periférico Luces, cuya interfaz se mapea a partir de la dirección 8000h, tiene asociada una rutina de interrupción denominada `luces`, la cual se encarga de actualizar el estado de los leds de dicho periférico conforme al estado de sus interruptores, de forma que sólo luzcan aquellos leds asociados a interruptores activados.

El código proporcionado se encarga de instalar en memoria la rutina `luces` y de habilitar el tratamiento de interrupciones, quedando a continuación a la espera de que el usuario pulse alguna tecla del periférico Teclado. Cuando eso ocurra, se abandonará el código de E/S programada asociado al teclado, se deshabilitará el tratamiento de interrupciones y se pasará a ejecutar un bucle infinito. Dicho de otra forma, mientras el usuario no pulse ninguna tecla del periférico Teclado estará habilitado el tratamiento de interrupciones, quedando deshabilitado dicho tratamiento en el momento que se detecte que el usuario ha pulsado alguna tecla. A continuación se muestra el listado incompleto del programa.

```

origen 0100h
inicio ini
.pila 10h
.codigo
procedimiento luces
    push r0
    push r1
    movl r0, 80h
    movh r0, 00h

    *** hueco 1 ***

    pop r1
    pop r0
    iret
finp

ini:
    movh r0, ?????
    movl r0, ?????
    movh r1, bytealto direccion luces
    movl r1, bytebajo direccion luces
    mov [r0], r1
    sti
    movh r2, 1
    movl r2, 0
    movh r0, 0FFh
    movl r0, 1

buclesinc:
    mov r3, [r0]

    *** hueco 2 ***

    cli
    jmp -1
fin
    
```

Además, se sabe que todas las posiciones de memoria anteriores a la dirección 00FFh contienen el valor 0000h.

- 9.1 ¿Cuál es el número del vector de interrupción asociado al periférico Luces? Responder en hexadecimal

- 9.2 ¿Qué dos instrucciones faltan en el hueco etiquetado como \*\*\* HUECO 1 \*\*\* ?

- 9.3 ¿Qué dos instrucciones faltan en el hueco etiquetado como \*\*\* HUECO 2 \*\*\* ?

- 9.4 Dibujar el circuito de activación del periférico Teclado. Etiquetar correctamente las entradas.

- 9.5 Sabiendo que se generó una interrupción cuando se estaba ejecutando la primera instrucción del programa, ¿qué instrucción se ejecutará después del IRET? Responder indicando el mnemónico de la instrucción.

- 9.6 Si durante la ejecución del paso 4 de la instrucción IRET se generase una nueva interrupción desde el periférico Luces, ¿cuál sería la dirección de la siguiente instrucción que se ejecuta? Responder en hexadecimal.

- 9.7 Si durante la ejecución de la instrucción C0FFh se generase una interrupción desde el periférico Luces, ¿qué valor tomaría el registro PC tras ejecutar el último paso de esta instrucción? Responder en hexadecimal.