

Índice
Objetivos
Conocimientos y materiales necesarios
1. Conexión del teclado
2. Muestreo periódico
3. Códigos ASCII
4. Ejercicios adicionales

Entrada por muestreo. El teclado

Área de Arquitectura y Tecnología de Computadores – Versión 1.0.422, 23/03/2020

Objetivos

En esta sesión se pretende que el alumno se familiarice con uno de los mecanismos clásicos de sincronización de las operaciones de E/S, el denominado *muestreo periódico*. Para ello se desarrollarán diversos ejemplos utilizando la interfaz del teclado del computador teórico.

Conocimientos y materiales necesarios

Para poder realizar esta sesión el alumno debe:

- Comprender el funcionamiento básico del computador teórico así como de su ensamblador.
- Repasar la teoría sobre el funcionamiento del teclado y de su interfaz, prestando especial atención a la funcionalidad de los dos registros básicos de la interfaz de teclado: el de registro de datos y el registro de estado y control.
- Repasar la teoría relativa a entrada mediante muestreo periódico.
- Tener el archivo `5-1pantalla.sim` de la sesión anterior.

Durante la sesión se plantearán una serie de preguntas que deben responderse en el correspondiente [cuestionario](#) del Campus Virtual. El cuestionario se puede abrir en otra pestaña del navegador pinchando en el enlace mientras se mantiene pulsada la tecla `Ctrl`.

1. Conexión del teclado

En esta práctica se van a desarrollar ejemplos de entrada mediante muestreo periódico a través del teclado, así que lo primero que tendremos que hacer es conectarle un teclado a nuestro computador teórico. Para ello:



- Abre el simulador.
- Carga el archivo `5-1pantalla.sim` que has creado en la sesión anterior. Recuerda que en este archivo se guardaba un estado del simulador en el que habías dejado sin módulos de memoria 16K a partir de la dirección `C000h` y habías situado la interfaz de una pantalla a partir de la dirección `F000h`, ocupando 121 posiciones del espacio direccionable.
- Haz clic en el dibujo etiquetado con **ENTRADA/SALIDA** en la esquina inferior derecha de la ventana principal del simulador.
- Escoge la opción **Conectar Teclado**.
- Te aparecerá una ventana en la que tendrás que introducir las características del teclado que quieres conectar. En el lugar reservado para el nombre escribe **Teclado1**. Como dirección base (es decir, la dirección a partir de la cual se mapearán los dos registros del teclado) introduce **F700h**. Tendrás que rellenar también tres campos relativos a interrupciones aunque ahora no los necesitamos porque vamos a trabajar con entrada mediante muestreo periódico y no mediante interrupciones. Como vector de interrupción introduce 1, como prioridad 0 y deja sin marcar la casilla que indica si se generan interrupciones.
- Guarda el estado actual del simulador del computador en un archivo llamado **5-2Teclado.sim**. Te servirá para acceder rápidamente a un computador con una pantalla y un teclado conectados.

2. Muestreo periódico

En este apartado vamos a realizar un ejemplo de programación del teclado utilizando la técnica de sincronización denominada *muestreo periódico*. Para ello debemos conocer el funcionamiento del registro de estado y control del teclado. Este registro se mapea en la segunda dirección de la interfaz de teclado. Para el teclado que hemos conectado anteriormente, ¿en qué dirección estará mapeado? Responde en el [cuestionario](#): pregunta 1.

El registro de estado y control del teclado funciona de la siguiente forma: cuando se produce una pulsación de teclado, el bit de peso 8 de este registro se pone a 1. Este es el hecho que vamos a utilizar para hacer muestreo periódico sobre este registro. Así, aplicando esta técnica de sincronización podemos detectar cuándo se produce una pulsación en el teclado: simplemente tendremos que estar leyendo el registro de estado y control y, si hay un 1 en el bit 8, sabremos que se ha pulsado una tecla.

Pero ¿cómo sabemos cuando hay un 1 en el bit de peso 8? Para ello primero tendremos que traer el contenido del registro de estado y control del teclado a un registro de propósito general de la CPU y luego realizar una operación entre ese registro y una **máscara** (una combinación de unos y ceros) que dé cero cuando el bit 8 esté a 0 y distinto de 0 cuando el bit 8 esté a 1. Como máscara vamos a utilizar el valor `0000 0001 0000 0000`. ¿Qué operación necesitaremos? Responde en el [cuestionario](#): pregunta 2. (Piensa una operación de la ALU que dé 0 si el registro de estado y control es, por ejemplo, `0000 0000 0000 0001`, y que dé distinto de 0 si es `0000 0001 0000 0001`.)

Vamos a hacer un programa para ver cómo funciona la entrada por muestreo. El programa consistirá en un bucle infinito en el que se estará comprobando si hay una tecla pulsada. Cuando se detecte la pulsación de una tecla, se imprimirá un asterisco en la pantalla del computador teórico.

El listado, con algunas instrucciones incompletas, sería el siguiente:

```
ORIGEN 300h
.CODIGO
movl r0, ??h      ; Cargar la dirección del registro de estado
movh r0, ??h      ; y control del teclado en R0

movl r1, ??h      ; Cargar la máscara en R1
movh r1, ??h

movl r3, ??h      ; Cargar en R3 la dirección de la primera celda
movh r3, ??h      ; de pantalla

movl r4, '*'      ; Cargar lo necesario para imprimir un asterisco
movh r4, 7h       ; en color blanco sobre fondo negro

bucle:
mov r2, [r?]      ; Leer el registro de estado y control a R2
??? r2, r2, r1     ; Realizar operación con la máscara
br?? bucle        ; Si no se pulsó tecla, volver al principio del bucle

; Si estamos aquí, se pulsó tecla. Imprimir un asterisco
mov [r3], r4       ; Escribir asterisco
inc r3             ; Avanzar a la sgte. posición de pantalla
jmp bucle          ; Volver al bucle a ver si se pulsan más teclas

FIN
```



- Crea un archivo llamado `5-2pulsa.ens`.
- Copia el listado anterior completando los huecos indicados por las interrogaciones para que las instrucciones hagan lo que indican los comentarios. No hace falta que copies dichos comentarios en tu fichero fuente.
- Guarda, ensambla y carga el programa en el simulador.
- Vete ejecutando el programa paso a paso. Comprueba que todos los registros se cargan con los valores adecuados y ejecuta el bucle dos veces, sin haber pulsado ninguna tecla del teclado del computador teórico, para ver que mientras no se pulse ninguna tecla el programa funciona bien (debes ir pulsando a `F8` hasta que se ejecute dos veces la instrucción `mov r2, [r0]`).
- Cuando estés viendo en **IR** la instrucción `mov r2, [r0]` (por lo tanto esta instrucción ya estará ejecutada), pulsa una tecla en el teclado del computador teórico. Fijate que a la derecha de la ventana que representa el teclado se muestra el contenido del buffer de teclado y debe aparecer la tecla que acabas de pulsar.
- Cuando pulses `F8` vas a ejecutar la instrucción `and r2, r1`. ¿Será el resultado de la operación distinto de cero? Responde en el [cuestionario](#): pregunta 3. ¿Por qué? Comprueba si has acertado pulsando `F8`.
- ¿Cuántas veces más deberás pulsar `F8` para que se imprima el asterisco en pantalla? Responde en el [cuestionario](#): pregunta 4. Compruébalo pulsando `F8` hasta que aparezca el asterisco.
- Parece que el programa funciona... Pero vamos a probarlo un poco más. Sigue pulsando `F8` hasta que aparezca de nuevo en el registro **IR** el código máquina de la instrucción `mov r2, [r0]`. Esta instrucción acaba de traer el contenido del registro de control de la interfaz del teclado al registro `r2` de la CPU. Según el valor de ese registro, ¿se ha pulsado alguna tecla? Responde en el [cuestionario](#): pregunta 5. Entonces, ¿qué va a ocurrir? Compruébalo pulsado 3 veces más `F8` (y comprendiendo lo que pasa). ¡Solo has pulsado una vez el teclado del computador teórico y tienes dos asteriscos en pantalla! Algo va mal.

El problema es que mientras no se lea el carácter que hay en el registro de datos del teclado, no se pondrá a 0 el bit de peso 8 del registro de control. Esto lo puedes comprobar además porque en la ventana que representa el teclado sigue apareciendo en la primera posición del buffer la tecla que pulsaste. Vamos a arreglar nuestro programa para que sólo ponga un asterisco por cada letra, pero antes, para que compruebes hasta que punto funciona mal el programa, pulsa `F9` y observa la pantalla. Cuando te canses de ver aparecer asteriscosque no deberían aparecer, pulsa de nuevo a `F9` para detener la simulación.

Vamos a arreglar el problema:



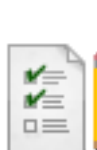
- Edita otra vez el archivo `5-2pulsa.ens`.
- Vamos a añadir las instrucciones necesarias para leer el contenido del registro de datos del teclado. Recuerda que este registro está mapeado en la primera dirección del rango de direcciones asignado para la interfaz de teclado. ¿Cuál es esa dirección? Responde en el [cuestionario](#): pregunta 6. Añade las instrucciones necesarias para inicializar antes del bucle el registro `r6` a ese valor.
- Busca la instrucción donde escribes el asterisco en pantalla. Justo antes de esa instrucción vamos a introducir otra que lea el contenido del registro de datos del teclado. Para ello, debe mover lo que hay en la dirección donde apunta `r6` al registro `r5`.
- Guarda el programa y ensámbalo. Carga en el simulador el archivo `5-2Teclado.sim` para volver a la situación inicial y luego carga el archivo con el programa modificado, es decir, `5-2pulsa.eje`.
- Ejecuta instrucciones hasta que hayas ejecutado el bucle una vez y llegues a la instrucción `brz -3`. Fijate que el programa sigue funcionando mientras no se pulse ninguna tecla del teclado conectado al computador teórico.
- Pulsa una tecla en el teclado del computador teórico.
- Pulsa `F8` hasta que llegues otra vez a la instrucción `brz -3`. La siguiente instrucción que ejecutes debe ser la nueva que has añadido. Antes de pulsar `F8`, fijate en la ventana del teclado en que de momento sigue en el buffer la tecla que has pulsado.
- Ejecuta la instrucción añadida con `F8`. ¿Ha desaparecido la tecla del buffer? Responde en el [cuestionario](#): pregunta 7.
- Pulsa `F8` hasta que aparezca la instrucción de lectura del registro de estado y control, es decir, `mov r2, [r0]`. Mira `r2`. Según el registro de estado y control, ¿hay alguna tecla pulsada? Responde en el [cuestionario](#): pregunta 8.
- Pulsa `F9` y comprueba pulsando teclas del teclado del computador teórico que el programa funciona ahora correctamente, es decir, sólo escribe un asterisco por cada tecla pulsada.

3. Códigos ASCII

En este apartado vamos a complicar un poco el programa realizado en el apartado anterior. No sólo detectaremos que se produce una pulsación de teclado, sino que también examinaremos qué tecla se ha pulsado. Para ello utilizaremos el registro de datos del teclado. Cuando se pulsa una tecla, en este registro queda registrado de dos formas qué tecla se ha pulsado:

- En el byte bajo se almacena el código ASCII de la tecla pulsada.
- En el byte alto se almacena el código SCAN asociado a la tecla.

Vamos a modificar el programa anterior para que cada vez que se pulse una tecla, en lugar del asterisco, se imprima el carácter correspondiente a esa tecla. Sigue estos pasos:



- Edita el archivo `5-2pulsa.ens` y guárdalo con el nombre `5-2letra.ens`.
- Busca la instrucción que lee el registro de datos (es la última que añadiste). Después de esa instrucción tendrás en `r5` el contenido del registro de datos. Para imprimir por pantalla la letra correspondiente a esa tecla necesitamos tener en la parte baja el código ASCII de la letra. Este código lo genera la interfaz de teclado, así que ya lo tenemos en el registro `r5`. En la parte alta necesitamos poner el byte de atributo. Añade una instrucción después de ésta que ponga como atributo en `r5` un 7 (carácter blanco sobre fondo negro).
- Ya tenemos en `r5` el atributo y el código ASCII, así que sólo queda escribirlo en pantalla. Sustituye la instrucción `mov [r3], r4`, que escribía siempre un asterisco, por otra que escriba el carácter leído.
- Guarda y ensambla el programa. En el simulador, carga el archivo `5-2Teclado.sim` para que vuelva a la situación inicial. A continuación carga el archivo que acabas de ensamblar.
- Ejecuta el programa con `F9`. Comprueba que funciona pulsando teclas en el teclado del computador teórico. Si no funciona correctamente, ejecuta el programa paso a paso para intentar detectar el origen del problema.

Ahora vamos a complicar un poco más el programa. Vamos a hacer que cuando se pulse la barra espaciadora se vaya a un bucle infinito en el que ya no se compruebe si se pulsan más teclas. Sigue estos pasos:



- Edita el fichero `5-2letra.ens`.
- Busca la inicialización del registro `r4` (está antes del bucle). Esa inicialización es un resto de cuando escribíamos asteriscos. Ahora ya no la necesitamos, pero vamos a utilizar `r4` para contener un valor que nos permita detectar si se pulsa la barra espaciadora. Como valor vamos a usar en la parte baja el código ASCII de la barra espaciadora y en la alta el código SCAN. En hexadecimal sería el valor `3220h`. Modifica las instrucciones de inicialización de `r4` para que carguen este valor.
- Busca en el código la instrucción donde se lee el registro de datos del teclado.
- Justo después de esa instrucción vas a introducir dos nuevas instrucciones para ver si se ha pulsado la barra espaciadora y, en caso de que sea así, saltar fuera del bucle de muestreo.
- La primera instrucción a añadir tiene que comparar (instrucción `cmp`) la tecla que se acaba de leer (está en `r5`) y el valor que pusimos antes en `r4`.
- La siguiente instrucción a introducir tiene que saltar si se produjo un cero, porque en ese caso `r4` y `r5` serían iguales, lo que querría decir que se pulsó la barra espaciadora. Pon como etiqueta del salto **fuera**.
- Sólo nos falta por añadir, justo antes de la directiva **FIN**, un bucle infinito que esté etiquetado con **fuera** para que sea el destino del salto anterior. Añade esta línea antes del **FIN**: `fuera: jmp -1`
- Guarda y ensambla el programa. En el simulador, carga el archivo `5-2Teclado.sim` para que vuelva a la situación inicial. A continuación carga el archivo que acabas de ensamblar.
- Ejecuta el programa pulsando `F9`. Comprueba que funciona pulsando las teclas **A** y **B** del teclado del computador teórico. Si todo va bien, se deberían haber escrito ambos caracteres en pantalla.
- Pulsa ahora la barra espaciadora. Si todo va bien, no se debería escribir en pantalla. Además la CPU debería estar en un bucle infinito ejecutando la instrucción `jmp -1`. Comprueba que ya no se detectan más teclas pulsando la **C** y la **D** y observa cómo quedan acumuladas en el buffer del teclado.

4. Ejercicios adicionales

Estos ejercicios adicionales permiten al alumno reforzar los conocimientos adquiridos durante la sesión práctica.



- Modifica el programa `5-2letra.ens` para que la letra que se imprime en pantalla sea siempre mayúscula, aunque en el teclado no esté pulsado `Shift W`.
- Modifica el programa `5-2letra.ens` para que cuando se pulse la tecla `Q` se salga del programa, sin importar si estaba pulsado `Shift W` o no. Tendrás que comparar con el código SCAN de la tecla, que, al contrario que el código ASCII, no cambia de mayúsculas a minúsculas.