Arquitectura de Computadores I

Laboratorio 2 (solución): encuesta + pantalla

En una empresa han instalado un sistema de detección automática de incendios. Para ello han utilizado un PC compatible con el Intel 8086. Para detectar fuego se utiliza un sensor cuyo controlador es KFU. Este sensor interrumpe por la línea IRQ3 cuando detecta un incendio, dejando en su registro de datos (dirección `0x381`) información acerca del fuego detectado (en concreto, esta información está codificada en un carácter). A pesar de que la sincronización de este dispositivo puede ser por interrupción, lo vamos a controlar por encuesta (como lo solemos hacer con el teclado). Como sabes, para ello es necesario inhibir las interrupciones de este controlador. Te pedimos que programes en lenguaje C la rutina que inhibe las interrupciones de este controlador: `void InhibirIntKFU();`

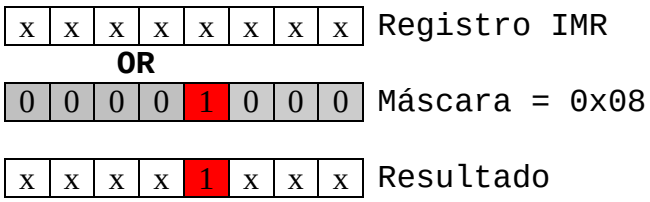
Una vez inhibidas las interrupciones, se necesita programar la rutina que realiza la encuesta de este periférico: `unsigned char EncuestaKFU()`. Esta rutina deberá devolver el código almacenado en el registro de datos del controlador KFU. La gestión de este periférico requiere además una secuencia de `strobe` en el bit 2 del registro de control (dirección `0x382`). También se pide programar en C la rutina de `strobe: void strobe_KFU().`

Hay que sacar por la pantalla la información referente al incendio leída en la rutina anterior. La pantalla está mapeada en memoria a partir de la dirección `C000h` y es una pantalla de 48 filas y 160 columnas. Programa en C la siguiente rutina que escribe un carácter en la fila y columna especificadas como parámetros.

```
void EscribeCar (int fila, int columna, unsigned char car,
                 unsigned char atrib);
```

Solución

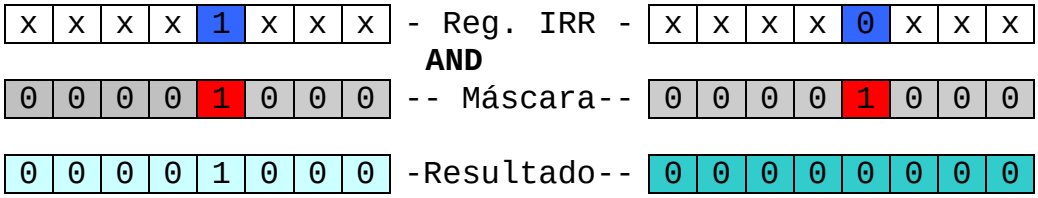
La primera rutina que se pide codificar es la rutina que inhibe las interrupciones del controlador KFU. Para llevar a cabo esta operación se debe escribir un 1 en el bit 3 (debido a que el controlador KFU utiliza la línea de interrupción número 3) del registro máscara (registro IMR) que se encuentra en el controlador de interrupciones. Para modificar únicamente el bit 3 del registro IMR, leemos este registro y realizamos una operación OR entre su valor y la máscara 0x08. El bit 3 del resultado de esta operación tomará el valor 1 (como se puede ver en el esquema siguiente). La operación acabará al escribir ese resultado en el registro IMR. Para asegurar que este conjunto de instrucciones se ejecutan seguidas al principio se deshabilitan todas las interrupciones y al final se vuelven a habilitar.



```
void interrupt Inhibir_Int_KFU()
{
unsigned char val;

DisableInts(); //Deshabilitar interrupciones
val = InPort(REG_IMR_CI8259);//Leer el registro IMR y guardarlo en la variable val
val = val | 0x08; //Operación OR con la máscara 0x08 para que el bit 3 valga 1 en val
OutPort(REG_IMR_CI8259,val); //Escribir la variable val en el registro IMR
EnableInts();//Habilitar las interrupciones
}
```

Teniendo en cuenta que hemos inhibido las interrupciones del dispositivo KFU, cuando KFU interrumpa se activará la línea IRQ3 y como consecuencia se activará el bit 3 del registro IRR pero la interrupción no se enviará el procesador. La encuesta de este dispositivo se realizará en el bit 3 del registro IRR. Para analizar el bit 3 del registro IRR realizamos una operación AND entre este registro y la máscara 0x08. El resultado de esta operación tomará valor 0 mientras el bit 3 del registro IRR valga 0, y tomará un valor distinto de 0 cuando se active el bit 3 del registro IRR (ver esquema siguiente).



Una vez finalizada la encuesta hay que leer el registro de datos del controlador KFU para obtener su valor, realizar la secuencia de `strobe` y devolver el carácter leído.

```
unsigned char Encuesta_KFU()
{
 unsigned char Mi_Irr;
 unsigned char Car;

        //Leer el registro IRR
        OutPort(REG_IRR_CI8259, 0x0A);
        Mi_Irr = InPort(REG_IRR_CI8259);

        //Realizar la encuesta mientras el bit 3 del IRR valga 0
        while ((Mi_Irr & 0x08)==0)
        {
             OutPort(REG_IRR_CI8259, 0x0A);
             Mi_Irr = InPort(REG_IRR_CI8259);
        }
        Car = InPort(0x381); //Leer el registro de datos de KFU
        strobe_KFU();
        return(Car);

}
```

La secuencia de `Strobe` hay que realizarla en el bit 2 del registro de control del controlador KFU. Para ello, primero hay que poner a 1 ese bit (operación OR) y después a 0 (operación AND).

```
void strobe_KFU()
{
 unsigned char val;

      val = InPort(0x382);        //Leer el registro de control del teclado
      val = val | 0x04 ;          //Poner a 1 el bit 2
      OutPort(0x382,val);         //Escribir en el registro de control
      val = val & 0xFD;           //Poner a 0 el bit 2
      OutPort(0x382,val);         //Escribir en el registro de control
}
```

Para poder sacar un carácter por pantalla tenemos que escribir en memoria, a partir del segmento `C000h`, el carácter junto con su atributo. Para escribir esa información en memoria disponemos de la función `EscribeByteFis` que escribe 1 byte en memoria (tanto el carácter como el atributo son de 1 byte). Se debe escribir primero el carácter y luego el atributo en la siguiente dirección. A la hora de calcular el desplazamiento respecto a la dirección base correspondiente al carácter a escribir, hay que tener en cuenta que cada carácter ocupa 2 bytes (1 byte el carácter y otro byte el atributo) y que la pantalla tiene 160 caracteres por fila.

```
void EscribeCar (int fila, int col, unsigned char car,
                 unsigned char atrib)
{
      EscribeByteFis(C000h, ((fila*160)+col)*2, car);
      EscribeByteFis(C000h, (((fila*160)+col)*2)+1, atrib);
}
```