

Taller 10 - Interrupciones en 8086

Objetivos

Aprender y comprender los detalles de la arquitectura 8086 cuando trabajamos con interrupciones.

Introducción

Las interrupciones son cambios en el flujo de ejecución no causados por otros programas, sino por otros eventos, usualmente relacionados con la entrada/salida. Una vez finalizada la interrupción el flujo de ejecución vuelve al punto donde se detuvo el programa interrumpido.

Responder las siguientes preguntas

- ¿Qué contexto es salvado por el hardware de x86 al atenderse una interrupción?
- ¿Qué contexto debería salvar el programador? En qué situaciones no es necesario?
- ¿Qué sucede si se produce una interrupción durante la ejecución de un manejador de interrupciones (handler)? Justifique su respuesta.

Interrupciones y Puertos de E/S

La empresa Fit-Flop le ha encomendado la programación de un microcontrolador para su nueva línea de caminadores domésticos (*tipo D*). El caminador se compone de una cinta potenciada por un motor, varios botones para controlar el funcionamiento y un display donde se despliega información, por ejemplo el tiempo de caminata o la velocidad alcanzada.

El caminador dispone de 3 botones para controlar la velocidad. Al presionar cada botón, la velocidad cambia a 11, 5 o 0 km/h, respectivamente. Los 3 botones son accesibles en los bits 2 al 0 del puerto de entrada/salida de solo lectura de un byte, BOTONES. El contenido del puerto es eliminado una vez leído. Al presionar un botón se activa la rutina de interrupción `hay_boton()`. Se asume que como máximo un único bit de BOTONES puede estar encendido en cada lectura.

El motor se enciende al escribir un 1 en el bit menos significativo del byte de entrada/salida de solo escritura MOTOR y se apaga colocando un 0.

El caminador posee un display que debe mostrar el tiempo transcurrido desde que el motor se puso en funcionamiento y la velocidad de movimiento de la cinta. El display se debe actualizar una vez por segundo, escribiendo en dos puertos de entrada/salida de solo escritura de 16 bits, DISPLAY_HI y DISPLAY_LO, con el siguiente formato:

- El bit 14 de DISPLAY_HI indica si el motor está en funcionamiento
- En los bits 13...7 de DISPLAY_HI se indica la velocidad de la cinta durante el último segundo, en décimas de km/h (por ejemplo, una velocidad de 9,5km/h se indica como 95)
- El bit 15 y los bits 6...0 de DISPLAY_HI no se utilizan
- En DISPLAY_LO se indica el tiempo transcurrido en segundos

Para medir la velocidad de funcionamiento, la cinta dispone de conectores cada 10 cm, los cuales al pasar frente a un sensor (incorporado en la cara interna de la cinta), disparan la interrupción `cinta()`.

Se dispone de un timer que interrumpe con frecuencia de 1hz, invocando a la rutina `tiempo()`.

Nota: Se puede utilizar la siguiente aproximación: 1km/h \approx 30cm/s

Nota2: Si el motor está encendido y el boton presionado es 11 o 5, el motor va a intentar por

si solo llegar a la velocidad solicitada por el usuario.

Nota3: El usuario puede querer usar el caminador sin encender el motor, por lo que se debe calcular e imprimir la velocidad en todo momento. El tiempo de uso solamente cuando el motor pasa de apagado a prendido.

Se pide

- a) Implementar todas las rutinas necesarias para controlar el sistema. (Recuerde el taller 9 dedicado a interrupciones).
- b) Compilar todas las rutinas de la parte anterior a assembler 8086.
- c) Escriba la secuencia de código que permita instalar un manejador de interrupciones en 8086. ¿Qué cuidados deben tenerse para que se instale correctamente?