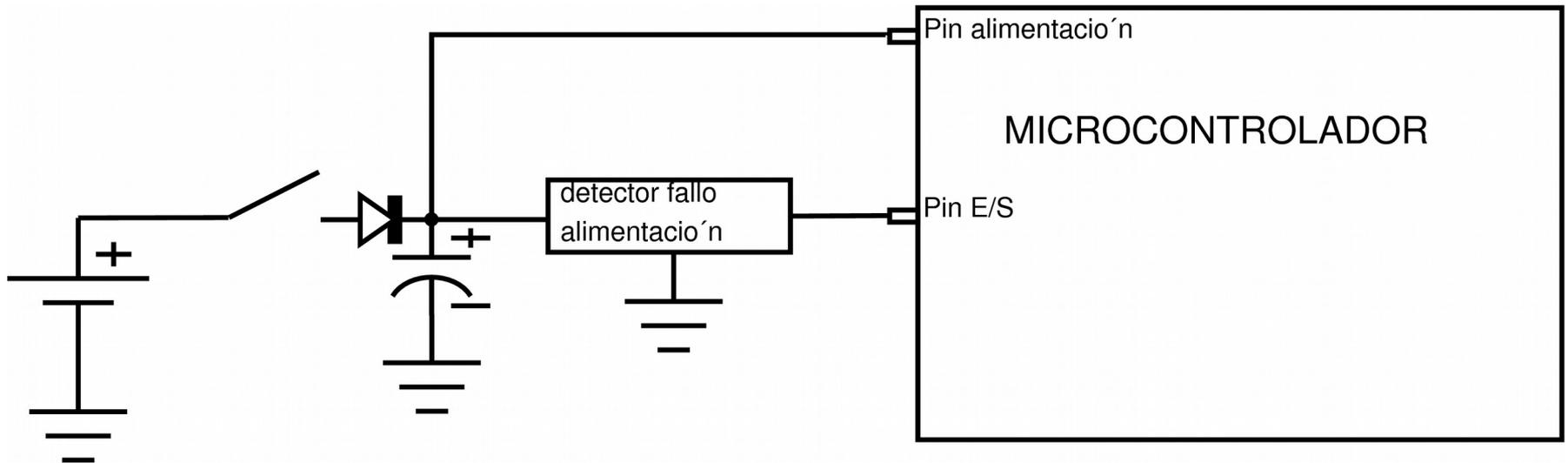


Interrupciones

Ejemplo 1

- ▶ Supongamos que un microcontrolador a 1MHz se usa para implementar un reproductor de música.
- ▶ Cuando se quita la alimentación al sistema el microcontrolador puede detectarlo y permanecer correctamente alimentado durante un tiempo.



Interrupciones

Ejemplo 1

- ▶ Al desconectar la alimentación debe guardarse en memoria no volátil el volumen seleccionado y número de pista actual.
- ▶ El código que realiza esa tarea tarda en ejecutarse 40 microsegundos.
- ▶ El condensador solo garantiza una alimentación adecuada durante 50 microsegundos.

Interrupciones

Ejemplo 1

- ▶ Solución inadecuada: pooling de la señal de E/S
- ▶ Escribir una subrutina que compruebe si la alimentación es correcta y...
 - si es correcta retorna.
 - si no es correcta salva la información y para.
- ▶ Modificar el programa para que llame a la subrutina cada dos o tres instrucciones.
- ▶ Inconvenientes:
 - ▶ Es engorroso de programar y poco elegante.
 - ▶ El programa es $\frac{1}{2}$ o $\frac{1}{3}$ más largo.
 - ▶ El programa es mucho más lento.

Interrupciones

Ejemplo 2

- ▶ Supongamos que un microcontrolador se ha programado para implementar el juego Simón.
- ▶ Se quiere aumentar la funcionalidad del sistema de forma que, simultáneamente, funcione también como joystick USB.
- ▶ Unas 60 veces por segundo debe enviar al PC el estado de los botones y palancas de juego.



Interrupciones

Ejemplo 2

- ▶ Solución inadecuada: pooling de un contador
 - ▶ Inicializar un contador descendente a un valor K de forma que llegue a cero en $1/60$ segundos.
 - ▶ Escribir una subrutina que compruebe el valor del contador y...
 - si positivo, retorna.
 - si es negativo, envía al PC el estado de los botones y palancas, inicializa de nuevo el contador a K y retorna.
 - ▶ Modificar el programa principal para que en el cuerpo de cada bucle (incluyendo los bucles de espera) se llame periódicamente a la subrutina.

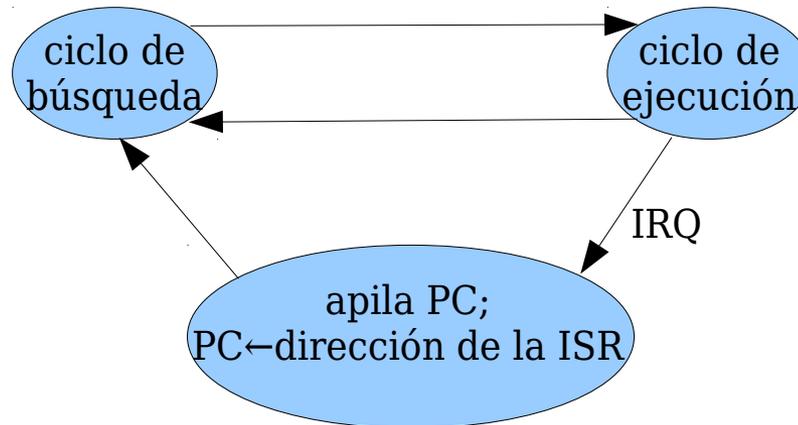
Interrupciones

Solución adecuada: atender IRQs

- ▶ Una petición de interrupción (IRQ) es una activación de una señal de entrada a la unidad de control para indicarle que ha ocurrido un suceso que requiere la ejecución de cierto código.
- ▶ Ese código se denomina manejador de interrupción o rutina de servicio de interrupción (ISR).
- ▶ Un sistema puede tener varias fuentes de IRQ, y normalmente cada una requiere una ISR distinta.
- ▶ Normalmente, cuando una IRQ ocurre, el procesador debe saltar inmediatamente al comienzo de la ISR correspondiente.

Interrupciones

- ▶ El computador simple no implementa interrupciones. Para atenderlas debería seguir este esquema:



- ▶ Normalmente, cuando una ISR termina de ejecutarse, el procesador debe continuar la ejecución por el punto donde se produjo la IRQ. Esto se soluciona con una instrucción de retorno que recupere el PC de la pila.
- ▶ El estado de los registros visibles debe ser el mismo cuando se retorna de la ISR que cuando se produce la IRQ (reestablecimiento del contexto ejecución). De nuevo, esto puede solucionarse usando la pila.

Interrupciones

- ▶ En procesadores con interrupciones vectorizadas, cada fuente de IRQ tiene asociada un identificador, normalmente de entre 4 y 8 bits.
- ▶ ¿Como obtiene el procesador la dirección inicial de la ISR a partir de ese identificador?
 - ▶ Opción 1: Las direcciones de inicio de las ISR son fijas y se obtienen fácilmente a partir del identificador de la IRQ correspondiente. Ejemplo: AVR.
 - ▶ Opción 2: Las direcciones de inicio de las ISR se encuentran en una tabla en memoria denominada tabla de vectores de interrupción. La dirección de la entrada que contiene la dirección de la ISR se obtiene fácilmente a partir del identificador de la IRQ correspondiente. Ejemplo: Motorola 68000.