



UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

DEPARTAMENTO DE ELECTRÓNICA

CURSO DE INFORMÁTICA II

INTERRUPCIONES PROYECTO DE INVESTIGACION

por:

SEBASTIAN BONILLA CRUZ

Medellín, Antioquia, Colombia
Junio, 2020

¿Qué son las interrupciones?

Las interrupciones son recursos o mecanismos que tiene el microprocesador para responder a eventos externos, permitiendo suspender el programa principal, para ejecutar una subrutina de servicio de interrupción (ISR por sus siglas en inglés Interrupt Service Routines); una vez terminada dicha subrutina, se reanuda la ejecución del programa principal.

Cabe recalcar que anteriormente se intentaba que el proceso de interrupción se hiciera de manera periódica, a esta configuración se le denominó "POLLING". Es un mecanismo de sondeo continuo, el cual consiste en que el mismo microprocesador se encarga de monitorear el estado o información de un evento; el principal inconveniente es que los eventos no pueden ser monitoreados en forma periódica, ya que el microprocesador consume tiempo en ejecutar diversas instrucciones que lo distraen de dicha actividad de sondeo, resultando una técnica ineficiente. La solución a esa problemática fue la incorporación al sistema de un dispositivo controlador de interrupciones para atender el cualquier momento una interrupción (Programmable Interrupt Controller, PIC).

1. HISTORIA

Si bien es coherente hablar de la historia de las interrupciones, ésta está tan ligada a la de los microprocesadores que es imposible buscar una sin tener que incurrir en la otra. El microprocesador surgió de las distintas tecnologías predecesoras, básicamente de la computación y de la tecnología de semiconductores; estas tecnologías se fusionaron a principios de los años 70, produciendo el primer microprocesador. Para 1971 la compañía Intel consiguió por primera vez poner todos los transistores que constituían un procesador sobre un único circuito integrado; naciendo así el primer microprocesador comercial de la historia, el Intel 4004; éste, todos sus predecesores y algunos de sus sucesores implementaron el sistema de interrupción denominado "POLLING" (Explicado anteriormente).

No obstante, debido a las nuevas necesidades y,

por ende, exigencias de la época, los ingenieros encontraron este método ineficiente. De este modo en 1977 sale al mercado el Intel 8085, cuya gran novedad consiste en la implementación del controlador programable de interrupciones (PIC) Intel 8259, permitiendo este la instauración del "Modo disparado por nivel"; método que debido a su eficiencia y eficacia sigue vigente hoy en día.

2. Tipos de interrupciones:

Atendiendo a la fuente que las produce, las interrupciones pueden clasificarse de la siguiente forma:

2.1 Interrupciones por Hardware:

Estas son asíncronas a la ejecución del procesador, es decir, se pueden producir en cualquier momento independientemente de lo que esté haciendo el CPU en ese momento. Las interrupciones de hardware son aquellas interrupciones que se producen como resultado de, por lo general, una operación de E/S. No son producidas por ninguna instrucción de un programa sino por las señales que emiten los dispositivos periféricos para indicarle al procesador que necesitan ser atendidos.

A su vez, considerando que el microprocesador, al recibir una petición, puede decidir aceptarla o no, hay dos tipos de interrupciones de hardware:

2.1.1 Enmascarables:

Bajo el control del software el microprocesador puede no atenderla o ignorarla. Normalmente se usan para la atención de periféricos en general.

2.1.2 No enmascarables:

El procesador no puede evitar atenderla, tiene mayor prioridad que otro tipo de interrupciones y se usa en eventos tipo catástrofe para el sistema.

2.2 Excepciones:

. Las excepciones son un tipo de interrupción sincrónica típicamente causada por una condición

de error en un programa, como por ejemplo una división entre 0 o un acceso inválido a memoria en un proceso de usuario. Normalmente genera un cambio de contexto a modo supervisor para que el sistema operativo atienda el error. Así pues, las excepciones son un mecanismo de protección que permite garantizar la integridad de los datos almacenados tanto en el espacio de usuario como en el espacio kernel. Cuando el Sistema Operativo detecta una excepción intenta solucionarla, pero en caso de no poder simplemente notificará la condición de error a la aplicación/usuario y abortará la misma.

2.3 Interrupciones por Software:

Las interrupciones por software, también denominadas “llamadas al sistema”, son aquellas generadas por un programa mientras este está ejecutándose. También se definen como un mecanismo de comunicación entre un proceso (que se ejecuta en modo usuario) y el sistema operativo (que se ejecuta en modo supervisor). El proceso emplea las interrupciones por software para notificar al sistema operativo que requiere de su intervención. Cabe recalcar que tienen más prioridad que las interrupciones de hardware.

Este tipo de interrupciones es la forma más importante que tendrán los programas de ejecutar funciones especiales del DOS (Disk Operating System) o del BIOS (Basic Input Output System). Estas funciones tienen un número de interrupción de software asociada, por lo que debemos saber qué es lo que hace cada interrupción para conseguir el efecto deseado.

Las interrupciones son principalmente subrutinas del BIOS o el DOS que pueden ser llamadas por un programa, su función es controlar el hardware, servir de contacto entre los programas y las funciones del BIOS y del DOS.

Las interrupciones de software pueden ser activadas directamente por el ensamblador invocando al número de interrupción deseada con la instrucción INT.

Este tipo de interrupciones podemos separarlas en dos categorías: Las interrupciones del sistema operativo DOS y las interrupciones del BIOS.

La diferencia entre ambas es que las interrupciones del sistema operativo son más fáciles de usar, pero también son más lentas, ya que estas interrupciones hacen uso del BIOS para lograr su cometido, en cambio, las interrupciones del BIOS son mucho más rápidas, pero tienen la desventaja que, como son parte del hardware, son muy específicas y pueden variar dependiendo incluso de la marca del fabricante del circuito.

La elección del tipo de interrupción a utilizar dependerá únicamente de las características que le quiera dar a su programa: velocidad (utilizando del BIOS) o portabilidad (utilizando las del DOS).

3. Implementación:

3.1 Hardware:

Cuando se genera una interrupción el microprocesador termina la ejecución de la instrucción en curso, salva el estado de registro y banderas, así como la dirección de memoria del contador de programa, entonces salta a la dirección de memoria donde está almacenada la rutina de servicio de interrupción ISR para ejecutar dicha rutina que tiene como objetivo atender el dispositivo que generó la interrupción. Una vez que ha finalizado la rutina de interrupción, el microprocesador restaura el estado que había guardado y retorna al programa principal a partir de la siguiente instrucción donde se quedó el contador del programa.

3.2 Software:

En general las interrupciones de software actúan de la siguiente manera:

- Un programa que se venía ejecutando luego de su instrucción I5, llama al Sistema Operativo, por ejemplo, para leer un archivo de disco (cuando un programa necesita un dato exterior, se detiene y pasa a cumplir con las tareas de recoger ese dato).

- A tal efecto, luego de I5 existe en el programa, la instrucción de código de máquina CD21, simbolizada INT 21 en Assembler, que realiza el requerimiento del paso 1. Puesto que no puede seguir la ejecución de la instrucción I6 y siguientes del programa hasta que no se haya leído el disco y esté en memoria principal dicho archivo, virtualmente el programa se ha interrumpido, siendo, además, que luego de INT 21, las instrucciones que se ejecutarán no serán del programa, sino del Sistema Operativo. Se detiene el programa y ordena en este caso mediante INT21 (interrupción predefinida) que recoge el dato solicitado, para poder seguir el programa que la ordenó).

- La ejecución de INT 21 permite hallar la subrutina del Sistema Operativo.

- Se ejecuta la subrutina del Sistema Operativo que prepara la lectura del disco.

- Luego de ejecutarse la subrutina del Sistema Operativo, y una vez que se haya leído el disco y verificado que la lectura es correcta, el Sistema Operativo ordenará reanudar la ejecución del programa auto interrumpido en espera.

- La ejecución del programa se reanuda.

Nota:

Debido a que en el punto anterior los procesos se han descrito de manera muy general, no se ha profundizado en la importancia que tiene el lenguaje de programación y el hardware con el que se está trabajando. No obstante, en el siguiente apartado se dejará clara esta cuestión.

Importancia del lenguaje de programación y el hardware con el que se trabaja:

Debido a que cada diseño de microprocesador tiene su propio lenguaje ensamblador, es clave entender cuán difícil es traducir determinado lenguaje de programación al respectivo lenguaje ensamblador. Naturalmente, mientras más alto sea el nivel de un lenguaje de programación más tiempo requerirá traducirlo al lenguaje ensamblador y,

por ende, a lenguaje de máquina. Por otro lado, el hardware y su pre programación son determinantes a la hora de establecer por cuantos canales (puertos/pines) se puede atender interrupciones y así mismo, instaurar el nivel de importancia de cada interrupción.

Implementacion en Arduino:

Para la implementacion de interrupciones en arduino se ha creado un circuito en tinkercad. Para acceder a éste se adjunta el siguiente enlace:

URL: <https://www.tinkercad.com/things/0o5QJfjgfa-grand-turing-leelo/editel?tenant=circuits>

Referencias

- [1] ANONIMO *INTERRUPCIONES*, Universidad de las palmas de la gran Canaria, España.
http://sopa.dis.ulpgc.es/ii-dso/leclinux/interrupciones/system_call/system_call.pdf
- [2] RODRIGUEZ, E. *Historia de los microprocesadores*, www.maestrosdelweb.com, (Feb., 2008).
<http://www.maestrosdelweb.com/historia-de-los-microprocesadores/>
- [3] LLAMAS, L. *QUÉ SON Y CÓMO USAR INTERRUPCIONES EN ARDUINO*, (Abril.28, 2016).
<https://www.luisllamas.es/que-son-y-como-usar-interrupciones-en-arduino/>
- [4] MIQUEL, B. *Historia de la informatica* (Mar., 1936).
https://books.google.com.co/books?id=8wwUowhAp_MC&pg=PA19&dq=historia+de+la+informatica&hl=es&sa=X&ved=2ahUKEwiJmqm_tKrQAhXFh0AKHZnxCXIQ6AEwAXoECAIQAg#v=onepage&q=historia%20de%20la%20informatica&f=false
- [5] MARTÍNEZ CANO, H. *Arquitectura de máquinas y computadoras curso II*, Universidad Nacional de Ingeniería, Nicaragua, Managua, (Jul., 2007).
- [6] ANONIMO. *Manejo de Interrupciones*, Universidad Nacional del Mar de Plata - Facultad de Ingeniería.
http://www3.fi.mdp.edu.ar/electrica/opt_archivos/arduino/Manejo_de_Interrupciones.pdf
- [7] CALDERON OLMEDA, C. *Metodo de manejo de interrupciones*.
https://prezi.com/p/znlwml_kst2x/metodos-de-manejo-de-interrupciones/
- [8] MONTEJO, P. *INTERRUPCIONES DE HARDWARE Y SOFTWARE*, [archivo de video], recuperado de <https://www.youtube.com/watch?v=UwrJvU86NCo> (Oct,28. 2012),
- [9] TECNOLOGIA01-ANONIMO *Control y manejo de interrupciones*, UNIVERSIDAD DE NSRIÑO
<https://es.slideshare.net/tecnologia01/control-y-manejo-de-interrupciones> (Dec. 5, 2013).
- [10] E. CERUZZI, P. *Breve historia de la computación*
<https://books.google.com.co/books?id=eBSGDwAAQBAJ&printsec=frontcover&dq=historia+de+la+informatica&hl=es&sa=X&ved=2ahUKEwjDx-j40qrqAhWNneAKHQfwBw8Q6AEwAnoECAUQAg#v=onepage&q&f=false>.