

Proyecto de investigación - Interrupciones

Stiven Velásquez López

Junio 2020

1 Las interrupciones en los microprocesadores.

(1) Las interrupciones son un proceso con el que cuentan dispositivos y procesos para captar la atención de la CPU sobre la aparición de alguna situación que requiera atención. De esta manera, los dispositivos pueden ocasionar que la CPU deje la tarea que esta ejecutando y atienda de manera inmediata la interrupción. Cuando se atiende el asunto se continua con el proceso pausado.

2 Historia de las interrupciones.

(5) En los sistemas de comunicación primitivos cuando una aplicación requería que una tecla fuese pulsada, se interrogaba de manera reiterada el teclado esperando que esta tecla fuese presionada, y mientras se esperaba no era posible realizar otras tareas, no existía lo que actualmente es conocido como los sistemas multitarea. El proceso anterior es comúnmente conocido como polling o sondeo, que hace alusión a una acción de consulta reiterada, regularmente hacia un dispositivo de hardware para crear una tarea sincrónica, esto también podría implementarse en dispositivos de software. Esta técnica fue reemplazada por las interrupciones. El polling presentaba el problema de ser bastante ineficiente, el procesador gastaba mucho tiempo en las actividades y consumía muchos recursos para ejecutar estas órdenes. Las interrupciones lograron dar solución a estos inconvenientes y despreocuparse de esta problemática, en este proceso el periférico solo se comunica con el procesador cuando se requiere. El procesador no sondea los aparatos, él solo espera a que le deban comunicar algo, sin la necesidad de consultar de manera constante, solo cuando le interrumpen resuelve el asunto. (6)

3 ¿Qué tipo de interrupciones existen?.

(2)

-Interrupciones por hardware: es una señal eléctrica elaborada por un dispositivo periférico. Con esto se indica al procesador que esta herramienta física requiere ser atendida. El procesador interrumpe la actividad que está

realizando y atiende la situación. Al finalizar la interrupción, el procesador continúa ejecutando la tarea que había suspendido, o inicia una nueva en tal caso que lo requiera. Las interrupciones de este tipo las producen fuentes como una tecla al presionarla y soltarla, otros pueden ser periféricos como la impresora, el disco, el puerto serie, etc. Estas interrupciones se producen por las señales de los periféricos, no son programadas y pueden ocurrir en cualquier instante.

-Interrupciones por software: Estas se generan por los programas haciendo uso de una función particular del lenguaje. Su finalidad es que la CPU ejecute alguna clase de función. Cuando acaba de ejecutarse, se continúan con las instrucciones del programa que produjo la interrupción. Con este tipo de interrupción se ejecutan funciones del DOS (Disk Operating System) o del BIOS (Basic Input Output System). Todas las funciones tienen un número de interrupción asociada. Es importante saber qué hace cada interrupción para conseguir el efecto deseado.

Excepciones: Son interrupciones sincrónicas para atender causas que hacen que un programa presente dificultades mientras se está ejecutando, y el sistema operativo trata este inconveniente, ejemplos claros pueden ser la división entre cero, un acceso inválido a la memoria, entre otros.

4 Implementación de interrupciones: hardware.

(7) En las interrupciones a nivel de hardware lo que se pretende es unir el periférico al procesador del sistema, un caso puntual es tocar un tecla en el teclado del PC, este dispositivo periférico es atendido inmediatamente por el procesador. Los 256 interruptores presentes no pueden llamarse en el mismo instante, para que esto no ocurra es importante que al instalar las tarjetas de expansión un interruptor no se utilice para dos dispositivos periféricos diferentes, ya que en tal caso ocurriría un choque y ningún dispositivo periférico cumpliría su tarea, no hay forma que el sistema logre distinguirlos. Así que los dispositivos que necesitan relacionarse con el procesador por interrupciones tienen que poseer una línea única que permita indicar al CPU cuando requiere de él para efectuar una tarea. Esta línea es conocida como IRQ. Estas líneas van hasta el controlador de interrupciones, el cual es un elemento del hardware que maneja las interrupciones y pertenece al procesador. El controlador de interrupciones habilita o deshabilita las líneas de interrupción para no presentar conflictos. Para llevar a cabo una interrupción a nivel de Hardware se deben cumplir ciertos pasos: 1. Se pausa el proceso que tiene la máquina antes de la interrupción. 2. Es necesario guardar los registros de la última actividad en alguna parte, para que luego de realizar la interrupción se pueda reanudar el proceso. 3. Se atiende el dispositivo que generó la interrupción. 4. Luego de atender la solicitud se restaura el proceso anteriormente ejecutado y se continúa con la ejecución normal del programa.

5 Implementación de interrupciones: software

.(3)

Las interrupciones por software se generan mientras el programa está en plena ejecución. Para llevar a cabo este tipo de interrupción se deben cumplir ciertos pasos: 1. Un programa que se encontraba en ejecución se comunica con el Sistema Operativo, por ejemplo para leer un archivo (Se requiere de un dato exterior, el programa se pausa y pasa a resolver el asunto) 2. Dado que no se puede seguir la ejecución de las instrucciones en el programa hasta que no se lea el disco, y mientras no esté el archivo en la memoria principal, todo el proceso se interrumpe, y ahora las instrucciones que se ejecutan no son del programa que venía ejecutándose, ahora son instrucciones del Sistema Operativo, se ordena una interrupción indefinida, se recoge el dato que se solicitó. 3. Se halla la subrutina del Sistema Operativo y luego se ejecuta para leer el disco. 4. Se realiza la lectura del disco y se verifica una correcta lectura, para luego reanudar la ejecución del programa pausado. Es importante tener en cuenta que a medida que el programa en un microprocesador se hace más tesioso y enredado, trabajar con “assembler” (lenguaje de programación de bajo nivel en el que comúnmente se representan las instrucciones básicas para los microprocesadores, computadores, etc.) en el manejo de las interrupciones es una tarea complicada, es importante buscar alternativas para estas implementaciones, por ejemplos usar el lenguaje C, y así los programas pueden simplificarse y volverse más sencillos y comprensibles. Como parte del hardware, es necesario que el microprocesador, microcontrolador y otros circuitos integrados programables, dispongan de una línea especial que debe ser parte del cúmulo de líneas de control del bus del sistema y que se denominan líneas de petición de interrupción (INT). El módulo de Entrada y salida se comunica con el procesador mediante esta línea y le advierte que esta preparado para hacer la transferencia. El procesador debe contar con un punto de conexión de entrada por donde entraran las interrupciones, y el módulo de entrada y salida debe contar con un punto de conexión para la salida, por donde se generan las interrupciones. Si un sistema informático no cuenta con estas características no es posible realizar interrupciones en los procesos.

6 Ejemplo de interrupción usando Arduino.

(4)

El ejemplo consiste en encender y apagar un led con un retardo de 1 segundo en la función del Loop. Se pone un pulsador en el PIN 2 del microcontrolador y se configura para que funcione por medio de interrupciones. El arduino aumenta un contador y lo muestra en el puerto serial, cada vez que se presione el botón.

En el siguiente link se muestra el proyecto realizado con su respectivo código debidamente comentado:

<https://www.tinkercad.com/things/cIKiNVJVmZ8-implementacion-de-interrupcion-en-arduino>

References

- [1] Santamaria Eduardo. Electronica digital y Microprocesadores. Universidad Pontificia Comillas, 1993.
- [2] V. Ramirez Edward, Weiss Melvyn. Introducci3n a los microcontroladores: equipo y sistemas. Limusa, 1986.
- [3] Fing.edu.uy. 2020. [online] Available at: <https://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teo/arq-teo08.pdf>, [Accessed 1 Julio 2020].
- [4] Ww3.fi.mdp.edu.ar. 2020. [online] Available at: <http://www3.fi.mdp.edu.ar/electrica/optarchivos/arduino/ManejodeInterrupciones.pdf>, [Accessed 28 junio 2020].
- [5] Valvano Jonatan.”Embedded Microcomputer Systems Real Time Interfacing”., CENGAGE Learning, Tercera (Año 2011).
- [6] P. Marwedel.”Embedded System Design”. Springer, Año 2006.
- [7] Fdi.ucm.es. 2020. [online] Available at: <http://www.fdi.ucm.es/profesor/jjruiz/WEB2/Temas/Curso0506/E>, [Accessed 1 July 2020].