

Investigación: Interrupciones en microprocesadores

Cruz Buitrago, Edgar Julian
julian.cruz1@udea.edu.co
Facultad de Ingeniería
Universidad de Antioquia
Medellín, Antioquia

Julio 2020

Resumen

Las interrupciones son una parte vital de la secuenciación de una computadora moderna. Fueron desarrollados para el manejo de excepciones y luego se aplicaron a eventos de entrada y salida. Hoy en día no se concibe ningún sistema que no deba implementar interrupciones para garantizar su buen funcionamiento.

Palabras clave: c++, qt, interrupciones, microprocesadores.

la UNIVAC I (1951) como el primer sistema en implementar la interrupciones ya que contaba con un manejo de excepciones para el desbordamiento aritmético. Más adelante aparece el primer uso de enmascaramiento de interrupción en la IBM 650 (1954), las primeras interrupciones de entrada y salida en la NBS DYSEAC (1954) y la captura de saltos o transferencias como fue llamado en su época en la IBM 704 (1955).[8]

1. Definición

Una interrupción es una señal que hace que un microprocesador trabaje temporalmente en una tarea diferente y luego vuelva a su tarea que tenía en curso.[5] Dicha señal es conocida como una solicitud de interrupción o IRQ por sus siglas en inglés; y son procesadas por un controlador de interrupciones, también llamado rutina de servicio de interrupción, o ISR por sus siglas en inglés.[2] Las interrupciones son de gran utilidad para dar atención a situaciones que requieren un tratamiento diferente o que inciden en el comportamiento de un sistema.

2. Antecedentes

Consultando la historia de las interrupciones nos encontramos con dos sistemas que son referenciados como los primeros en utilizarlas; el primero de ellos la UNIVAC 1103 (1953) el cual es citado en el libro de Bell y Newell, en una línea de tiempo de la computación en la categoría de procesamiento concurrente[1]. A pesar de que en el manual de la UNIVAC 1103[6] no se describen las interrupciones, publicaciones posteriores realizadas por los laboratorios que tuvieron acceso a ella dan evidencia de implementaciones para múltiples dispositivos de entrada y salida; sin embargo otros autores citan a

3. Clasificación

Existen diferentes formas de clasificar las interrupciones, por ejemplo, según su fuente clasificarlas en interrupciones internas o externas, comúnmente categorizadas como de software y de hardware respectivamente[3], pero en materia de microprocesadores un mayor entendimiento se puede dar indicando que la ejecución secuencial de instrucciones en un microprocesador puede ser interrumpida (i) por el mismo procesador, (ii) por una instrucción de interrupción (software) o (iii) por una señal de interrupción de hardware[4]. Cuando ocurre una interrupción, normalmente el procesador efectúa lo siguiente: 1. Los contenidos del contador de programa deben ser almacenados de forma que sean recuperables cuando la CPU pueda volver al punto en el que fue interrumpida; 2. El contador de programa se carga con la dirección de la rutina de servicio de la interrupción; 3. Algunos de los registros de la CPU que se necesiten en la rutina de la interrupción deben ser almacenados temporalmente al principio de la rutina, para ser recuperados al final de ella, este almacenamiento normalmente se hace en el stack; 4. Una vez efectuada la rutina de la interrupción, se restablecen los valores de los registros y del contador de programa y el procesador vuelve otra vez al punto del programa en que fue interrumpido[7].

4. Aplicación en Software

A nivel de lenguajes de programación es muy común el uso de interrupciones y varían según las clases implementadas o la filosofía del lenguaje; por ejemplo en el desarrollo del curso de INFORMATICA II, se fueron implementando diferentes tipos de interrupciones quizás sin ahondar en sus conceptos, acá algunos ejemplos: en la captura por teclado el comando "cin" de la librería estándar solo termina la lectura hasta que la tecla ENTER sea presionada, en las implementaciones realizadas con las interfaces gráficas de QT, utilizamos timers que al cumplir cierto tiempo desencadenaban en otra acción o función. Estos son algunos ejemplos sencillos pero que dan a evidenciar la utilidad de las interrupciones.

5. Ejemplo

Fue implementado el ejemplo de arduino en la siguiente URL:
<https://www.tinkercad.com/things/bDHqkZy8Wdc>.

Referencias

- [1] Allen Bell C. Gordon y Newell. *Computer Structures: Readings and Examples*. 1971. URL: http://gordonbell.azurewebsites.net/computer_structures_-_readings_and_examples/00000065.htm.
- [2] Per Christensson. *Interrupt Definition*. 2018. URL: <https://techterms.com/definition/interrupt>.
- [3] A.P. Godse y D.A. Godse. *Microprocessor and Microcontroller*. Technical Publications, 2010. ISBN: 9788184315417. URL: <https://books.google.com.co/books?id=B0B-JnN6SHwC>.
- [4] N. MATHIVANAN. *PC-BASED INSTRUMENTATION: CONCEPTS AND PRACTICE*. PHI Learning, 2007. ISBN: 9788120330764. URL: <https://books.google.com.co/books?id=0B65rMNQDo8C>.
- [5] *Microprocessor Design*. 2019. URL: https://en.wikibooks.org/w/index.php?title=Microprocessor_Design/Interrupts&oldid=3620027.
- [6] *Notes on the logic of the ERA 1103 Computer System*. 1953. URL: http://www.textfiles.com/bitsavers/pdf/univac/1103/ERA_1103_Jul53.pdf.
- [7] E. Santamaria. *Electrónica digital y microprocesadores*. Ingeniería (Universidad Pontificia Comillas). Universidad Pontificia Comillas, 1993. ISBN: 9788487840333. URL: <https://books.google.com.co/books?id=7dk7YCtu0HMC>.
- [8] Mark Smotherman. *Interrupts*. 2017. URL: <https://people.cs.clemson.edu/~mark/interrupts.html>.