

INTERRUPCIONES



ARREOLA VASQUEZ JESUS ALBERTO

T/M

8.-B

ING. MECATRONICA

UPZMG

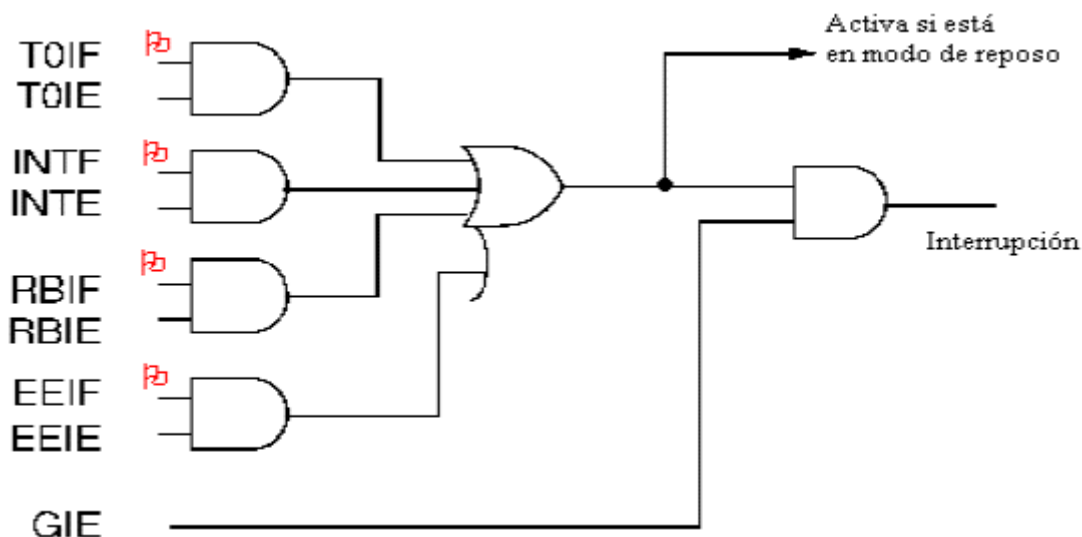
Interrupción

El manejo de interrupciones es una técnica de programación, basada en un mecanismo automático en el hardware del microcontrolador, que permite dar atención a algún periférico interno o externo, únicamente en el momento en que ésta se requiera. Una interrupción es en realidad una llamada a una subrutina, pero iniciada por el hardware del propio periférico y no por la instrucción "CALL". La interrupción puede ocurrir en cualquier instante.

Una interrupción consiste en un mecanismo que provoca la alteración del orden lógico de ejecución de instrucciones como respuesta a un evento externo, generado por el hardware de entrada/salida en forma asincrónica al programa que está siendo ejecutado y fuera de su control. En forma alternativa se puede decir que:

- Una interrupción consiste en un mecanismo que le permite al hardware la invocación de una rutina fuera del control del programa que está siendo ejecutado.

Se menciona que es asincrónico y fuera de su control, lo que significa que el programa que está siendo ejecutado no tiene control sobre el momento en que este mecanismo se dispara, ni sobre la propia ocurrencia del fenómeno. Es decir que, en particular, no hay sincronismo entre la ejecución del programa y la, eventual, invocación a esa rutina "especial" disparada por la interrupción.



Pedido de Interrupción

El mecanismo de interrupción comienza con el pedido de interrupción ("interrupt request") generado por un controlador de entrada/salida. Este pedido se genera a raíz de alguna condición detectada por el controlador (ej: dispone de un dato para ser leído por la CPU, terminó de ejecutar la lectura del sector de disco solicitada, hay una condición de error en el byte recibido por la línea de comunicaciones, etc, etc). Las condiciones que generan el pedido varían de acuerdo al tipo de controlador que se trate y de acuerdo a como esté configurado el mismo.

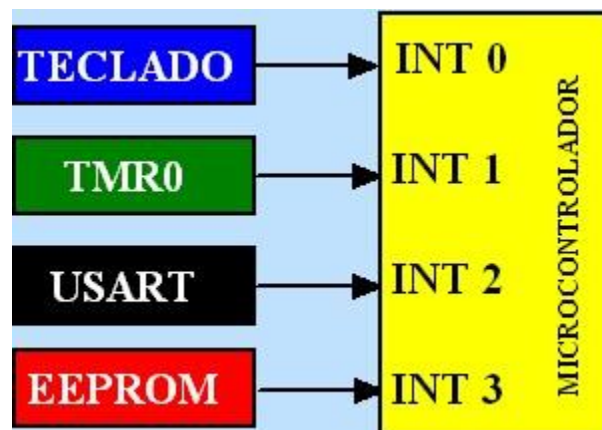
Detección por Nivel

En este caso la CPU reconocerá que hay un pedido de interrupción pendiente mientras su entrada INT esté en el nivel lógico alto ("1"). Esto significa que toda vez que la CPU detecte esa entrada en alto, desencadenará la invocación a la rutina de atención, incluso cuando haya terminado recién de ejecutar una rutina de interrupción.

Detección por Flanco

En este caso lo que interesa a los efectos de determinar que existe un requerimiento de interrupción es la existencia de un cambio de nivel bajo ("0") a alto ("1"), es decir un flanco, en la entrada INT. Es decir que no importa el valor absoluto actual de la entrada INT, lo que se tiene en cuenta es si existió un cambio de 0 a 1.

Si hubo un flanco y aún no ha sido invocada la rutina de atención, la CPU lo hace y da por cumplido el pedido. Luego veremos que esto tiene sus implicancias en el diseño de las rutinas de servicio a las interrupciones.



Atención del Pedido de Interrupción

Cuando la CPU reconoce que hay un pedido de interrupción desencadena el mecanismo de atención a dicha solicitud. Por ahora vamos a considerar que la CPU acepta la interrupción,

dejando para más adelante la consideración de en qué casos esto ocurre. Para atender una solicitud de interrupción, la CPU realiza los siguientes pasos:

- "termina" de ejecutar la instrucción actual. Notemos que utilizamos las comillas para señalar que, en realidad, lo que ocurre es que la CPU consulta si hay un pedido de interrupción al final del ciclo de instrucción, luego de la etapa de "write" y antes del siguiente "fetch".

- salva el valor actual del puntero de instrucción (IP), como forma de poder regresar a ejecutar la siguiente instrucción del programa que fue interrumpido, luego de la ejecución de la rutina de servicio a la interrupción. Es de destacar que las distintas arquitecturas realizan esta actividad de diferentes formas. Podemos distinguir básicamente dos:

- utilizando un stack. Esta vía es la habitual en las arquitecturas que implementan un stack por hardware, como es el caso de los procesadores Intel.

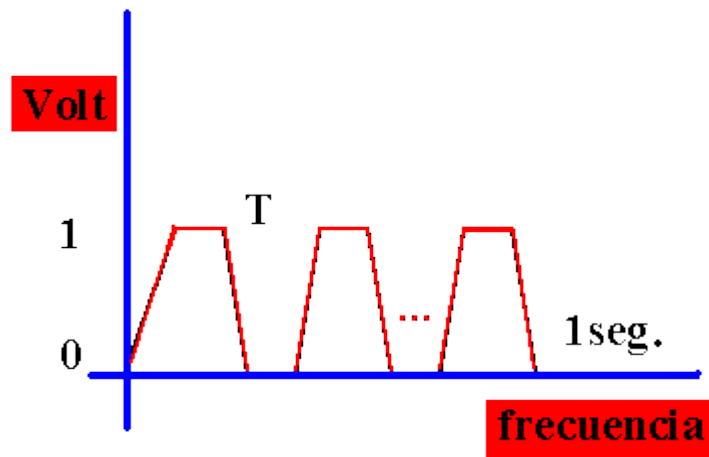
- utilizando un registro. Esta vía es la habitual de las arquitecturas RISC, en particular la SPARC.

- identifica el controlador de E/S que realizó el pedido de interrupción. Veremos este tema en el siguiente punto.

- obtiene la dirección de la rutina de servicio de la interrupción correspondiente al controlador identificado. También hay distintas formas de implementar esto, aunque la mayoría son variaciones sobre el concepto de vector de interrupciones.

Identificación del Controlador que solicita la Interrupción

Es normal que un sistema tenga múltiples controladores de E/S. Por tanto, hay que tener un mecanismo que permita saber cuál controlador de E/S generó un pedido concreto de atención mediante el mecanismo de interrupción. Las soluciones a este problema son variadas y pueden estar basadas en el hardware o en el software.



Identificación por Hardware

Líneas INT/IRQ independientes en la CPU: En este caso el propio CPU tiene múltiples entradas INT (numeradas, por ej: INT0, INT1, INT2, etc.) y la idea es conectar un controlador de E/S a cada una de ellas. La identificación es entonces por hardware y directa: el controlador que está pidiendo la interrupción es el que está conectado a la entrada INTn donde se detecta la solicitud. En este caso hay una dirección de la rutina de servicio a la interrupción por cada línea de pedido disponible.

Mecanismo INT/INTA

En mecanismo fue diseñado por Intel para sus primeros microprocesadores de 8 bits (8080 y 8085) y luego mantenido en las sucesivas familias arquitectónicas que los sucedieron. La idea es que el CPU dispone de una única entrada de pedido de interrupción INT, a la cuál se conectan en modalidad OR-cableado todos los pedidos de interrupción de los distintos controladores de E/S. También dispone de una salida denominada INTA (Interrupt Acknowledge) que le avisa al controlador de E/S que ha sido aceptado su solicitud de interrupción y le indica con esa señal que coloque en el bus de datos su identificación. La CPU entonces realiza una lectura del bus de datos y obtiene el identificador.

Habilitación de Interrupciones

La CPU tiene la capacidad de aceptar o no los pedidos de interrupción de los controladores de E/S. Esta capacidad está implementada de dos maneras, una general y otra selectiva, a las que denominaremos enmascaramiento y deshabilitación, sólo a los efectos de distinguirlas en la denominación.

Prioridades

Existen dos situaciones que pueden llevar al mecanismo de interrupciones a tener que tomar una decisión acerca de cómo proceder:

- hay dos (o más) solicitudes de interrupción simultáneas
- hay una (o más) solicitud de interrupción mientras se está ejecutando una rutina de servicio de una interrupción previa y las interrupciones han sido habilitadas.

Interrupciones Simultáneas

Cuando una CPU o un controlador de interrupciones implementa un mecanismo de prioridades, éste determinará cuál de múltiples solicitudes que ocurran a la misma vez será atendida. Es de notar que el concepto de simultaneidad no implica que los pedidos de interrupción lleguen exactamente en el mismo instante del tiempo, sino que lleguen durante el período entre una verificación y otra por parte de la CPU, la cuál puede estar postergada en el tiempo si las interrupciones están deshabilitadas. El mecanismo de prioridades puede adoptar distintas estrategias:

- prioridad fija
- prioridad configurable
- sin prioridad.

Cuando la prioridad es fija siempre será atendida primero la solicitud que provenga de una línea de pedido que tenga mayor jerarquía. Normalmente se utiliza la numeración de las entradas para fijar su prioridad.

Cuando la prioridad es configurable, la misma se puede cambiar en función de las condiciones del sistema en general. Cuando el sistema es sin prioridad debe implementar un sistema de selección de la solicitud a atender que asegure la equitatividad en selección de las distintas entradas.

REFERENCIAS:

<https://lsi.vc.ehu.eus/pablogn/docencia/manuales/SO/TemasSOuJaen/PLANIFICACIONDE PROCESOS/4y5ElRelojdeInterrupciones.Uso de Prioridades.htm>

<https://www.monografias.com/trabajos37/memoria-e-interrupcion/memoria-e-interrupcion2.shtml>

<http://www.puntoflotante.net/INTERRUPCIONES.htm>