

Las interrupciones en el 8086

Santiago Roatta

8 de marzo de 2010

Digital III
FCEIA - UNR

Índice

- 1 ¿Qué es una interrupción?
- 2 Las interrupciones en el 8086
 - Las interrupciones enmascarables
 - Las interrupciones no enmascarables
 - El papel de la pila
- 3 Un circuito
- 4 Preguntas

¿Qué es una interrupción?

Definición

Es un evento que interrumpe la ejecución del programa en ejecución para pasar a ejecutar una rutina denominada servicio de interrupción

Se compone de cuatro etapas

¿Qué es una interrupción?

Definición

Es un evento que interrumpe la ejecución del programa en ejecución para pasar a ejecutar una rutina denominada servicio de interrupción

Se compone de cuatro etapas

- Solicitud

¿Qué es una interrupción?

Definición

Es un evento que interrumpe la ejecución del programa en ejecución para pasar a ejecutar una rutina denominada servicio de interrupción

Se compone de cuatro etapas

- Solicitud
- Reconocimiento

¿Qué es una interrupción?

Definición

Es un evento que interrumpe la ejecución del programa en ejecución para pasar a ejecutar una rutina denominada servicio de interrupción

Se compone de cuatro etapas

- Solicitud
- Reconocimiento
- Atención

¿Qué es una interrupción?

Definición

Es un evento que interrumpe la ejecución del programa en ejecución para pasar a ejecutar una rutina denominada servicio de interrupción

Se compone de cuatro etapas

- Solicitud
- Reconocimiento
- Atención
- Retorno

Solicitud de interrupción

¿Cómo se solicita una interrupción?

El proceso de interrupción comienza por una solicitud de interrupción que se cursa al procesador a través de una señal asincrónica de entrada. El 8086 tiene dos entradas de interrupción: *INTR* y la *NMI*

Solicitud de interrupción

¿Todas las solicitudes son atendidas?

En el 8086 existe un bit interno del micro (IF: interrupt enable flag bit) que se ubica en el Flag Register. Ese bit actúa como máscara de las interrupciones enmascarables. Si la máscara se encuentra activada (IF=0) la solicitud es ignorada. Si por el contrario IF=1, el pedido es aceptado y se dará inicio a el ciclo de reconocimiento. La existencia o no de este bit, separa las interrupciones en dos grupos: enmascarables y no enmascarables. El 8086 posee dos entradas para tal fin: *INTR* (enmascarable) y la *NMI* (no enmascarable)

Solicitud de interrupción

Interrupción enmascarable

La solicitud se realiza a través de la entrada *INTR* (Interrupt Request). *INTR* solicita una interrupción de *tipo* variable que se determina en el ciclo de reconocimiento. *INTR* es una entrada activa por nivel alto, que se muestrea durante el último ciclo de reloj de cada instrucción. En ese momento se determina si el micro debe continuar ejecutando la próxima instrucción o debe ingresar en el ciclo de reconocimiento de interrupción. Para que el pedido sea reconocido, *INTR* debe permanecer en alto (al menos) durante el último ciclo de reloj de la ejecución de la instrucción en curso. Luego *INTR* puede bajar en cualquier momento después del flanco descendente del primer pulso de *INTA*, dentro de la secuencia de reconocimiento.

Solicitud de interrupción

Interrupción no enmascarable

La solicitud se realiza a través de la entrada *NMI* (Non Maskable Interrupt) que solicita una interrupción de *tipo* fijo igual a 2, por lo tanto no es necesario que se realice un ciclo de *INTA*. *NMI* es una entrada asincrónica activa por flanco ascendente. El flanco ascendente en la entrada de *NMI*, se lachea en el interior del micro hasta que la instrucción en curso finaliza. No es enmascarable por lo que su atención es inevitable. Esto hace que, independiente de la estructura que se monte para las interrupciones enmascarables, puede pensarse que, *NMI* es siempre la interrupción más prioritaria ya que puede interrumpir la ejecución de cualquier servicio generado por *IRQ*.

Solicitud de interrupción

¿**INTR** es activa por flanco o por nivel?

Cuando comienza la ejecución de la rutina de atención de la interrupción, *IF* debe pasar a 0 al menos hasta que la línea *INTR* pase a bajo. Si esto no fuera así, solo llegaría a ejecutarse la primera instrucción de esta rutina y el procesador comenzaría nuevamente con otro ciclo de reconocimiento. Conceptualmente: las interrupciones activas por nivel deben ser enmascarables.

Solicitud de interrupción

¿NMI es activa por flanco o por nivel?

Como *NMI* es no enmascarable, no podría ser activa por nivel, ya que sería imposible impedir que se repita el comienzo de la atención cuando se ejecuta la primera instrucción de la rutina de servicio. Conceptualmente: las interrupciones no enmascarables deben ser activas por flanco.

Ciclo de Reconocimiento

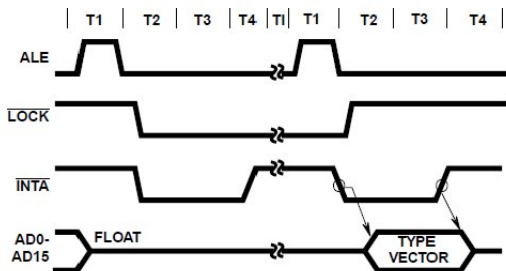
¿Cómo sabe el procesador que tipo de interrupción se produce?

Mediante una secuencia de reconocimiento que se genera solo como respuesta a una interrupción proveniente de INTR. Está formado por dos ciclos de INTA, cada uno de ellos muy similar a un ciclo de lectura. En el segundo ciclo de INTA, se lee un byte del bus de datos que identifica el “tipo” de la interrupción. Por tanto, INTA (Interrupt Acknowledge) es la salida de reconocimiento de interrupción y se usa como pulso de lectura del “tipo” en el ciclo de INTA. Es activa en bajo y pasa a este valor durante T₂, T₃ (y T_w) de los ciclos de reconocimiento de interrupción. Su objetivo es avisar al dispositivo solicitante que su pedido de interrupción ha sido aceptado y que debe notificar al micro cuál es el “tipo” de interrupción que pide.

Ciclo de Reconocimiento

¿Cómo sabe el procesador donde comienza la rutina de servicio?

En el primer KByte de memoria están los 256 vectores de interrupción que indican el comienzo de 256 posibles rutinas. Cada vector ocupa cuatro bytes: dos para el segmento y dos para el desplazamiento



¿Qué hace el procesador previo al comienzo de la ejecución de la rutina de servicio de interrupción?

¿Qué hace el procesador previo al comienzo de la ejecución de la rutina de servicio de interrupción?

- PUSH del registro de banderas, el CS y el IP en la pila

¿Qué hace el procesador previo al comienzo de la ejecución de la rutina de servicio de interrupción?

- PUSH del registro de banderas, el CS y el IP en la pila
- Reset del bit de máscara para que, al menos por default, la rutina de ejecución de interrupción no sea interrumpida por interrupciones enmascarables

¿Qué hace el procesador previo al comienzo de la ejecución de la rutina de servicio de interrupción?

- PUSH del registro de banderas, el CS y el IP en la pila
- Reset del bit de máscara para que, al menos por default, la rutina de ejecución de interrupción no sea interrumpida por interrupciones enmascarables
- Carga de los nuevos CS e IP (provenientes de la IVT)

¿Qué hace el procesador previo al comienzo de la ejecución de la rutina de servicio de interrupción?

- PUSH del registro de banderas, el CS y el IP en la pila
- Reset del bit de máscara para que, al menos por default, la rutina de ejecución de interrupción no sea interrumpida por interrupciones enmascarables
- Carga de los nuevos CS e IP (provenientes de la IVT)
- Ejecuta la primera instrucción de la rutina de atención

¿Cuándo finaliza la ejecución de la rutina de servicio de interrupción?

Con la instrucción IRET que hace lo siguiente:

¿Cuándo finaliza la ejecución de la rutina de servicio de interrupción?

Con la instrucción IRET que hace lo siguiente:

- POP de los dos bytes del IP de la pila,

¿Cuándo finaliza la ejecución de la rutina de servicio de interrupción?

Con la instrucción IRET que hace lo siguiente:

- POP de los dos bytes del IP de la pila,
- POP de los dos bytes del CS de la pila

¿Cuándo finaliza la ejecución de la rutina de servicio de interrupción?

Con la instrucción IRET que hace lo siguiente:

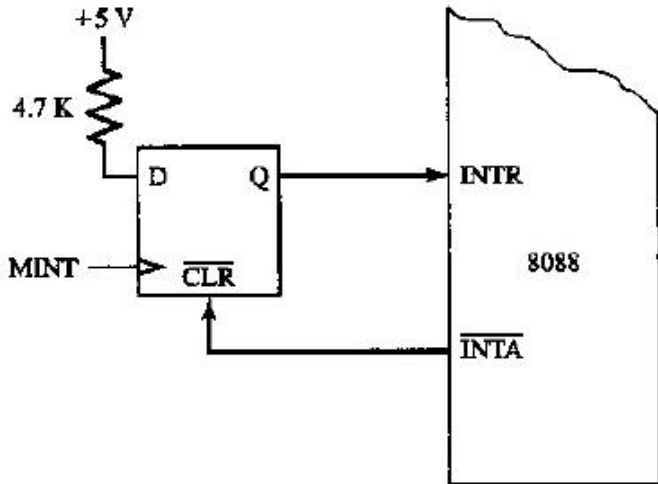
- POP de los dos bytes del IP de la pila,
- POP de los dos bytes del CS de la pila
- y POP de los dos bytes del registro de banderas.

¿Cuándo finaliza la ejecución de la rutina de servicio de interrupción?

Con la instrucción IRET que hace lo siguiente:

- POP de los dos bytes del IP de la pila,
- POP de los dos bytes del CS de la pila
- y POP de los dos bytes del registro de banderas.
- Al recuperar el IP el CS de la pila, luego de ejecutar IRET, el procesador continúa ejecutando la instrucción que se encuentra luego de la última instrucción ejecutada antes de entrar en la rutina de atención de interrupción. Al recuperar el registro de banderas, se repone el estado anterior de IF

El circuito más simple !!



Algunas preguntas:

Algunas preguntas:

- ¿Es realmente eficiente la atención a dispositivos de E/S por interrupciones?

Algunas preguntas:

- ¿Es realmente eficiente la atención a dispositivos de E/S por interrupciones?
- ¿Es posible y/o conveniente anidar interrupciones?

Algunas preguntas:

- ¿Es realmente eficiente la atención a dispositivos de E/S por interrupciones?
- ¿Es posible y/o conveniente anidar interrupciones?
- ¿Es posible calcular el tiempo entre la solicitud y la ejecución del servicio?

Algunas preguntas:

- ¿Es realmente eficiente la atención a dispositivos de E/S por interrupciones?
- ¿Es posible y/o conveniente anidar interrupciones?
- ¿Es posible calcular el tiempo entre la solicitud y la ejecución del servicio?
- ¿Es la única arquitectura posible?

Algunas preguntas:

- ¿Es realmente eficiente la atención a dispositivos de E/S por interrupciones?
- ¿Es posible y/o conveniente anidar interrupciones?
- ¿Es posible calcular el tiempo entre la solicitud y la ejecución del servicio?
- ¿Es la única arquitectura posible?
- ¿Qué es una interrupción por software?