# Arquitectura de Computadoras

**Clase 2**

**Interrupciones**

# Interrupciones

* Mecanismo mediante el cual se puede interrumpir el procesamiento normal de la CPU.
* Ejecución secuencial de instrucciones de un programa
* Pueden ser de origen interno o externo a la CPU.

**¿Porqué Interrumpir?**

* Por resultado de una ejecución de una instrucción
* Ej: desbordamiento aritmético (“overflow”), división por cero
* Por un temporizador interno del procesador.
* Permite al S.O. realizar ciertas funciones de manera regular.
* Por una operación de E/S.
* Ej: para indicar la finalización normal de una operación.
* Por un fallo de hardware.
* Ej: error de paridad en la memoria, pérdida de energía.

**¿Qué hacer si interrumpen?**

* En casi todos los casos, implica transferir el control a otro programa (el GESTOR) que:
* salve el estado del procesador
* corrija (o responda a) la causa que ocasionó la interrupción
* restaure el estado original del procesador
* retorne a la ejecución normal del programa interrumpido
* ¿continúo ó repito la instrucción interrumpida?

# Jerarquía de interrupciones

Si hay múltiples fuentes que pueden solicitar interrupción se establece cuales son mas importantes. Ser consideran

* No Enmascarables: las que NO pueden ignorarse
* Indican eventos peligrosos o de alta prioridad.
* Enmascarables: pueden ser ignoradas.
* Con instrucciones podemos inhibir la posible solicitud.

# Interrupciones por hardware

* Son las generadas por dispositivos de E/S.
* Son las “verdaderas” interrupciones.
* El sistema de cómputo tiene que manejar estos eventos externos “no planeados” ó “asincrónicos”.
* No están relacionadas con el proceso en ejecución en ese momento.
* Son conocidas como interrupt request.

# Traps/excepciones

* Interrupciones por hardware creadas por el procesador en respuesta a ciertos eventos como:
* Condiciones excepcionales: overflow en ALU de punto flotante.
* Falla de programa: tratar de ejecutar una instrucción no definida.
* Fallas de hardware: error de paridad de memoria.
* Accesos no alineados ó a zonas de memoria protegidos

# Interrupciones por software

Muchos procesadores tienen instrucciones explícitas que afectan al procesador de la misma manera que las interrupciones por hardware.

* Generalmente usadas para hacer llamadas a funciones del SO.
* Esta característica permite que las subrutinas del sistema se carguen en cualquier lugar.
* No requieren conocer la dirección de la rutina en tiempo de ejecución.

# Interrupciones por software (2)

Hay sistemas que no permiten hacer una llamada directa a una dirección de la función del SO, por estar en una zona reservada.

* ¿Qué pasa si no tuviera las int. por software?
* Debería escribir todas las funciones que necesito ó
* Al cargar un programa habría que “mirar” todas las llamadas a funciones del BIOS y SO y reemplazar en el código las direcciones de todas estas funciones invocadas.

# Ciclo de instrucción

* Tres pasos:
* Captación
* Ejecución
* Gestión de interrupciones

INICIO

Captar la

siguiente

instrucción

Ejecutar la

instrucción

Si hay pedido de

interrupción

Interrumpir proceso

**Ciclo de captación**

**Ciclo de ejecución**

PARADA

**I Hab**

**I Deshab**

**Ciclo de interrupción**

# Ciclo de interrupción

* Se comprueba si se ha solicitado alguna interrupción.
* indicada por una señal (flag) de pedido de interrupción.
* Si no hay señal se capta la siguiente instrucción.
* Si hay algún pedido de interrupción pendiente:
* Se suspende la ejecución del programa en curso
* Guarda su contexto (próxima instrucción a ejecutar y el estado del procesador)
* Carga el PC con la dirección de comienzo de una rutina de gestión de interrupción. Se inhiben otras interrupciones.
* Finalizada la rutina de gestión, el procesador retoma la ejecución del programa del usuario en el punto de interrupción.

Programa en

ejecución

Gestor de la

interrupción (4)

Vectores de

interrupción

1

-

int

2

-

apilar

PSW

3

-

salto

5

-

desapilar

Pila

PC

**¿cómo**

**trabajan?**

## Diagrama de estados de un ciclo de instrucción con interrupciones

Captación

de instrucción

Captación

de

operando

Almacena

-

miento del

operando

Cálculo de la

dirección de

instrucción

Decodifica

-

ción de la

operación de

la instrucción

Cálculo de

la dirección

del

operando

Operación

con los

datos

Cálculo de

la dirección

del

operando

Comproba

-

ción de

interrupción

Interrupción

Resultados

múltiples

No

interrupción

Cadena

o vector

Instrucción

siguiente

Operandos

múltiples

# Interrupciones múltiples (1)

* Interrupciones inhabilitadas
* El procesador puede y debe ignorar la señal de petición de interrupción si se produce una interrupción en ese momento.
* Si se hubiera generado una interrupción se mantiene pendiente y se examinará luego una vez que se hayan habilitado nuevamente.
* Ocurre una interrupción, se inhabilitan, se gestiona la misma y luego se habilitan otra vez.
* Por lo tanto las interrupciones se manejan en un orden secuencial estricto.

# Interrupciones múltiples (2)

* Definir prioridades
* Una interrupción de prioridad más alta puede interrumpir a un gestor de interrupción de prioridad menor.
* Cuando se ha gestionado la interrupción de prioridad más alta, el procesador vuelve a las interrupciones previas (de menor prioridad).
* Terminadas todas las rutinas de gestión de interrupciones se retoma el programa del usuario.

# Procesamiento de interrupciones secuenciales

Programa de usuario

Gestor de

interrupción X

Gestor de

interrupción Y

# Procesamiento de interrupciones anidadas (priorización)

Gestor de

Programa de usuario

interrupción X

Gestor de

interrupción Y

## Reconocimiento de interrupciones

Interrupciones multinivel

* Cada dispositivo que puede provocar interrupción tiene una entrada física de interrupción conectada a la CPU.
* Es muy sencillo, pero muy caro.

Línea de interrupción única

* Una sola entrada física de pedido de interrupción a la que están conectados todos los dispositivos.
* Se debe “preguntar” a cada dispositivo si ha producido el pedido de interrupción (técnica Polling/encuesta).

### Reconocimiento de interrupciones (2)

Interrupciones vectorizadas

* El dispositivo que quiere interrumpir además de la señal de pedido de interrupción, debe colocar en el bus de datos un identificador (vector).
* Lo coloca el periférico directamente ó
* Controlador de Interrupciones (que se ocupa de todo).

# Escenario de trabajo

* Si el procesador tiene una única entrada de pedido de interrupciones.
* Si tenemos varios productores de interrupciones

¿Cómo lo solucionamos?

* Dispositivo controlador programable de interrupciones **‘PIC’**.

# Conexionado

Dispositivos

externos

Bus del

sistema

Memoria

Vectores

IR0

IR7

Int

IntA

Procesador

PIC

# Interrupciones del MSX88

* Hardware
* Línea INT
* Con respuesta de reconocimiento INTA
* Línea NMI

Procesos de atención son por salto indirecto

* Software
* Instrucción INT xx

Para retorno desde el gestor debe usarse la instrucción IRET

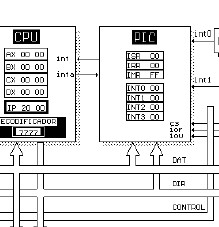
* Proceso de atención vectorizado

# Tabla de vectores de Interrupción

* Es el nexo entre tipo de interrupción (0...255) y el procedimiento designado para atenderla.
* Cada entrada es una doble palabra (4 bytes).
* Dirección del procedimiento que brinda el servicio.
* Ej: 0000yyyy, donde yyyy es la dirección lógica/física.
* Vectores preasignados
* Tipo 0 – finaliza ejecución de programa
* Tipo 3 – punto de parada para depuración/seguimiento
* Tipo 6 – lectura de entrada std. Requiere el uso de BX.
* Tipo 7 – escritura de salida std. Requiere BX y AL.

# Controlador de Interrupciones

Registros internos PIC

* EOI: para comandos
* Para fin de int escribir

20H

* IMR: máscara de int
* enmascara con ‘1’
* IRR: petición de int
* Indica con bit en 1
* ISR: int en servicio
* Indica con bit en 1
* INT0...INT7
* c/u con su vector

# Conexionado y direccionamiento

* Los registros internos del PIC se sitúan a partir de la dirección 20H.
* Son accedidos con operaciones lectura y escritura en el espacio de E/S (IN y OUT).
* Interrupciones hardware asignadas
* INT0 – tecla F10
* INT1 – Timer
* INT2 – Handshake
* INT3 – DMA
* INT4 a INT7 no usadas

# Referencias

* William Stallings, Capítulo 3.
* MSX88, Manual de usuario.

Lecturas recomendadas

* “Interrupciones en la arquitectura INTEL IA32”, Blázquez, J.M. 2004.
* “sobre interrupciones y MSX88”, Quiroga et al., Apunte de. cátedra. 2017