Interrupciones - PIC

* [Finalidad de las interrupciones](#_1j89jxj2wu19). [Para que se utiliza un controlador de interrupciones.](#_867bvr4v4i01)
* [¿Qué es una interrupción?](#_tbdqcqupp45f) [Describa cómo funcionan.](#_1j89jxj2wu19) [¿Cómo se utiliza un controlador de interrupción?](#_8nl323fvzc0f)
* [Explique el mecanismo de interrupción.](#_kisixem98ifr) [Describa las características y el funcionamiento de un pic](#_cxvkvt1kf876)
* ¿Cómo es la estructura de un módulo de E/S? [Describa las características del controlador de interrupciones PIC.](#_cxvkvt1kf876)

# ¿Qué son las interrupciones?

Mecanismo mediante el que otros módulos (E/S, memoria) pueden interrumpir el procesamiento normal de la CPU.

Son eventos que indican que existe una condición en algún lugar del sistema, o del programa en ejecución, que requiere la atención del procesador. Generalmente resultan en una transferencia forzada del flujo de ejecución hacia una rutina denominada “manejador de interrupciones”. Las interrupciones se asocian normalmente a eventos de hardware, mientras que las excepciones se producen cuando se detectan ciertas condiciones durante la ejecución, como división por cero, fallos de página, violaciones de segmento, etc.

# ¿Para qué son las interrupciones? ¿Cómo funcionan?

Las interrupciones proporcionan una forma de mejorar la eficiencia del procesador. Los dispositivos externos suelen ser mucho más lentos que el procesador y esperar por las instrucciones de E/S sería un desperdicio de procesador. Con las interrupciones, el procesador puede dedicarse a ejecutar otras instrucciones mientras una operación de E/S está en curso. La operación de E/S se realiza concurrentemente con ĺa ejecución de instrucciones del programa de usuario.

Cuando el dispositivo externo está listo para aceptar más datos del procesador, el módulo de E/S de este dispositivo externo envía una señal de petición de interrupción al procesador. Éste suspende la operación del programa que estaba ejecutando y salta a un programa llamado “gestor de interrupción” que da servicio al dispositivo en cuestión y luego continúa con el programa original.

Una interrupción es una *interrupción* en la secuencia normal de funcionamiento. Cuando la misma se completa, la ejecución sigue: el procesador y el SO son los responsables de detener el programa de usuario y después permitir que prosiga desde el mismo punto.

# Mecanismo de interrupción

Se añade al ciclo de instrucción un ciclo de interrupción donde el procesador comprueba si se generó alguna interrupción. Si no hay señal de interrupción se avanza con la instrucción siguiente y si hay señal:

1. Suspende la ejecución del programa en curso y guarda su contexto (el contenido actual del PC y demás)
2. Carga el PC con la dirección de comienzo de una rutina de gestión de interrupción

El mecanismo para el manejo de interrupciones y excepciones en la arquitectura IA-32 permite que éstas sean manipuladas de manera transparente a los programas de aplicación y al mismo Sistema Operativo. Cuando se genera una interrupción, el procedimiento en ejecución se suspende automáticamente mientras el procesador ejecuta el manejador correspondiente; cuando esta operación se termina, el procesador reanuda la ejecución de la tarea interrumpida. La reanudación del proceso sucede sin pérdida de la continuidad del programa, a menos que el retorno no sea posible o que el evento haya causado la terminación del programa.

# Tipos de interrupciones

Las interrupciones pueden ser de dos orígenes:

* **Internas** (generadas por software):
  + Generalmente usadas para hacer llamadas a funciones del SO (esto permite que las subrutinas del sistema se carguen en cualquier lugar)
  + No requieren conocer la dirección de la rutina en tiempo de ejecución
* 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
* **Externas a la CPU** (generadas por hardware):
  + Generadas por dispositivos de E/S
  + Son las “verdaderas” interrupciones
  + El sistema de cómputo tiene que manejar estos eventos externos “no planeados” o “asincrónicos”
  + No están relacionadas con el proceso en ejecución en ese momento
  + Son conocidas como interrupt request

# Clases de interrupciones

* **Programa**: Por el resultado de una ejecución de una instrucción (Ej: Overflow, división por 0, intento de ejecución de una instrucción inexistente, intento de acceso fuera del espacio de memoria permitido para el usuario)
* **Temporización**: Clock interno del procesador que permite al SO realizar ciertas funciones de manera regular
* **E/S**: Por una operación de E/S (Ej: indicar la finalización normal de una operación o avisar condiciones de error)
* **Fallo de HW** (Ej: Error de paridad en la memoria, pérdida de energía, etc)

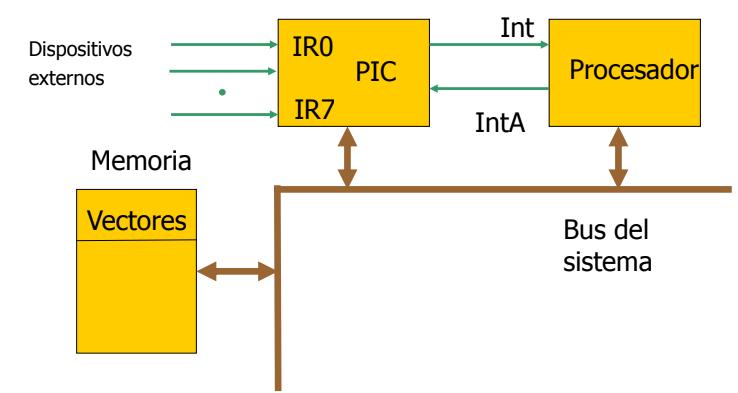
# Controlador de interrupciones (PIC)

## ¿Qué es?

El dispositivo controlador programable de interrupciones (PIC) es un chip que sirve si el procesador tiene una única entrada de pedido de interrupciones o si tenemos varios productores de interrupciones.

Es un dispositivo usado para combinar varias fuentes de interrupciones sobre una o más líneas del CPU. Se maneja con prioridades (las de número mas bajo se atienden primero) y tiene un conjunto de registros internos:

* IRR (Interrupt Request Register): Registro de petición de interrupción, indica con bit en 1 las interrupciones demandadas hasta el momento.
* ISR (In Service Register): Registro de interrupción en servicio, indica con bit en 1 cuál es la interrupción que está siendo atendida.
* IMR (Interrupt Mask Register): Registro de maścara de interrupciones, permite el enmascaramiento selectivo de cada una de las entradas de interrupción, indicando con bit en 1. Tras un reset los bits de este registro quedarán en 0. Indica cuáles deben ser ignoradas.
* EOI (End of Interruption): Fin de interrupción. Como consecuencia, se pone en 0 el bit del ISR correspondiente.
* INT0...INT7: 8 registros, donde carga el valor del vector de interrupción correspondiente



## Tareas realizadas por el PIC

* Puesto que existen muchos dispositivos que pueden solicitar interrupciones, el PIC debe priorizarlas cuando existen varias IRQ’s simultáneas
* Después de enviar una solicitud de interrupción, debe enviar un número de interrupción (número de vector) cuando el procesador indica que está listo para atender la petición
* Mantiene un registro de que se está procesando una interrupción: cuando esto sucede, no envía más peticiones al procesador hasta que este le responde con una señal de EOI (End Of Interrupt), indicando que la rutina de servicio precedente ha terminado o puede aceptar otra interrupción
* Puede enmascarar de forma selectiva cualquiera de las 8 IRQ’s que tiene conectadas

## ¿Cómo funciona?

El controlador de interrupciones puede manejar hasta ocho peticiones de interrupción independientes al mismo tiempo, numeradas de la 0 (INT0) a la 7 (INT7), de las cuales seleccionará una única para presentarla a la entrada de interrupción INT de la CPU.

Si más de una petición de interrupción se producen exactamente al mismo tiempo entonces el PIC las pasa a la CPU en un orden de prioridad, donde la petición por la entrada 0 tiene la prioridad más alta y la de la 7 la menor.

En el ciclo de interrupción, el procesador comprueba si se ha generado alguna interrupción (indicada por una señal -flag- de pedido de interrupción). Si no hay señal, el procesador continúa con el ciclo de captación (capta la instrucción siguiente) y si hay alguna interrupción pendiente:

* Suspende la ejecución del programa en curso y guarda su contexto (dirección de la siguiente instrucción a ejecutar -el contenido del PC- y el estado del procesador)
* Carga el PC con la dirección de comienzo de una rutina de gestión de interrupción
* El procesador prosigue con el ciclo de captación y accede a la primera instrucción del programa de gestión de interrupción que dará servicio a la interrupción.
* Al completarse la rutina de gestión de interrupción, el procesador prosigue la ejecución del programa de usuario en el punto que se interrumpió

Con el uso de interrupciones, el procesador puede dedicarse a ejecutar otras instrucciones mientras una operación de E/S está en curso. El procesador no tiene que comprobar repetidamente el estado del módulo hasta que el mismo pueda transmitir o recibir datos, siendo los dispositivos mucho más lentos que el procesador

**Fuentes**:

* [Clase 02](https://drive.google.com/open?id=1gZ4atzH_eyJJo4ZBYKbkyCwPUHj_sfFU)
* [Capítulo 3.2 - Funcionamiento del computador - Interrupciones](https://drive.google.com/file/d/1boDA36iwQ-FJyWqwt7HZveSPT6yY9gzi/view?usp=sharing) (Stalling 5ta ed. Pág 58 y 60)
* [“7.4 - Controlador de interrupciones (PIC)”](https://drive.google.com/open?id=1GYK-Y0ys8Ui4-XsGMS--ITlN4-Hf_PGi) (Manual MSX88, Pág 35)
* [“Interrupciones en la Arq IA32”](https://drive.google.com/open?id=1O2ixsspxyS8AEY7KlDWXXkG1A05KkRsC) (PDF complementario)