

Arquitectura de Computadoras

Interrupciones



Curso 2020
Prof. Jorge M. Runco

Interrupciones

- La funcionalidad dentro de un sistema de cómputo es implementada por hardware que ayuda al procesador.
- Cada dispositivo opera a su propio ritmo, pero hay necesidad de sincronizar la operación del procesador con estos dispositivos.

Interrupciones (2)

- Una solución: el procesador pregunta a c/dispositivo su estado (si hay dato disponible). Cuando el dato está disponible el procesador puede leerlo y procesarlo. (Polling)

- Desventajas :
 - El procesador malgasta tiempo preguntando continuamente si hay algún dato disponible.
 - Hay que esperar a los dispositivos más lentos

Interrupciones (3)

- Mejor solución : cuando un dispositivo necesita la atención del procesador, envía una señal para avisarle, es decir produce una interrupción. Este mecanismo permite alterar la secuencia normal de procesamiento. Recordar otros mecanismos (jmp, call)

Int 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

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

Int por software

- Muchos procesadores tienen instrucciones explícitas que afectan el estado del procesador en la misma manera que int por hardware. Generalmente usadas para hacer llamadas a funciones del SO. Esta característica permite que las subrutinas del sistema se carguen en cualquier lugar.

Int por software

- No requieren conocer la dirección de la rutina en tiempo de ejecución.
- Son más cortas que Call.
- Hay sistemas que no permiten hacer una llamada directa a una dirección de la función del SO, pues es una zona reservada.
- Se pueden reubicar fácilmente.

¿Qué pasa si no las tuviera ?

- Al cargar un programa habría que “mirar” todas las llamadas al BIOS y SO y reemplazar en el código las direcciones de todas estas funciones invocadas

Interrupciones

Por software

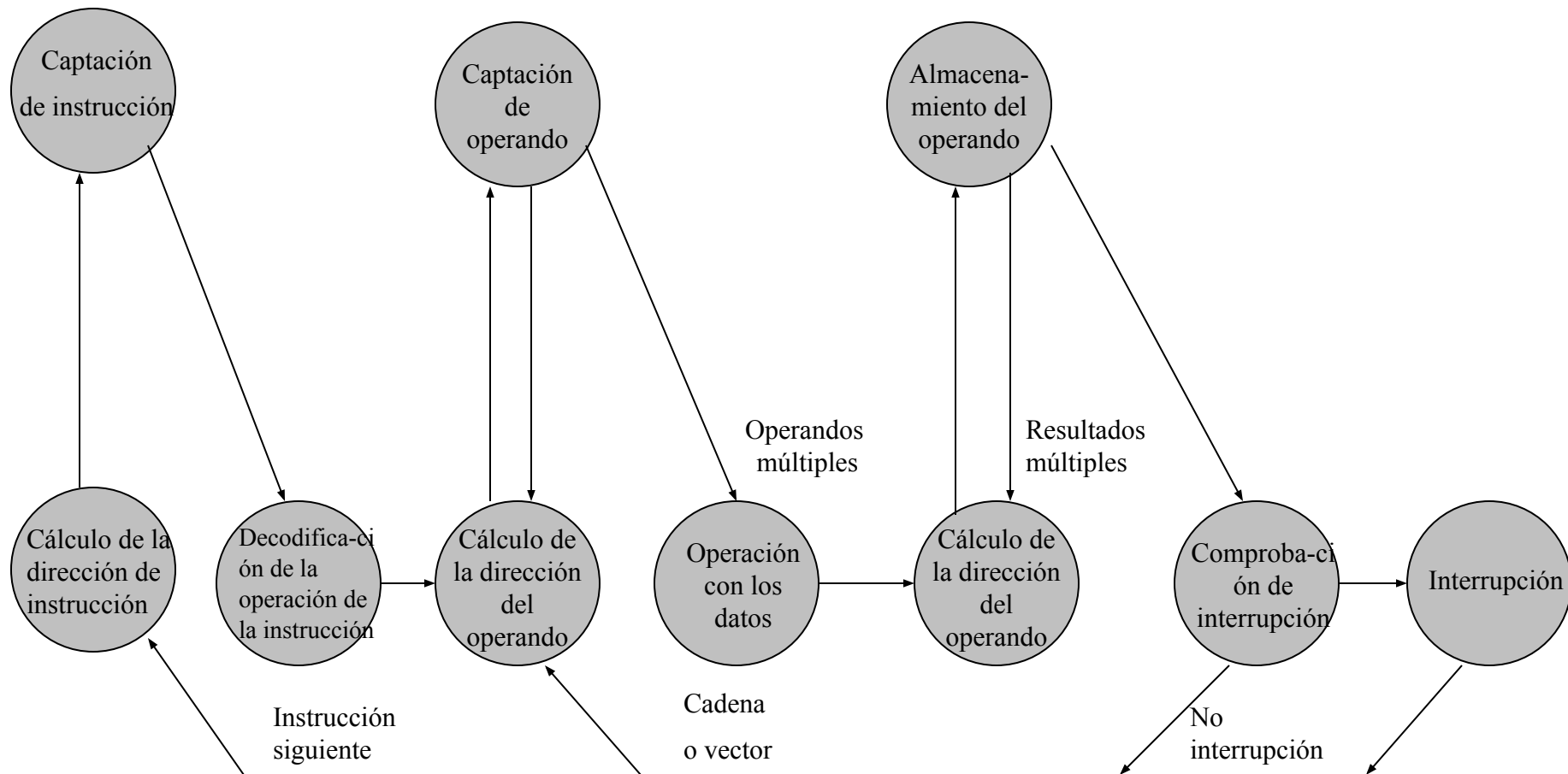
Es un llamado a una subrutina (procedimiento). Es por medio de una instrucción específica (INT N). Este llamado se denomina "interrupción" porque el mecanismo para encontrar la subrutina a ejecutar es el mismo que para las interrupciones por hardware.

Por hardware

Es un llamado a una subrutina (procedimiento). Es por medio de una señal externa. NO hay una instrucción específica en el programa.

Interrupciones por hardware

Diagrama de estados de un ciclo de instrucción con interrupciones



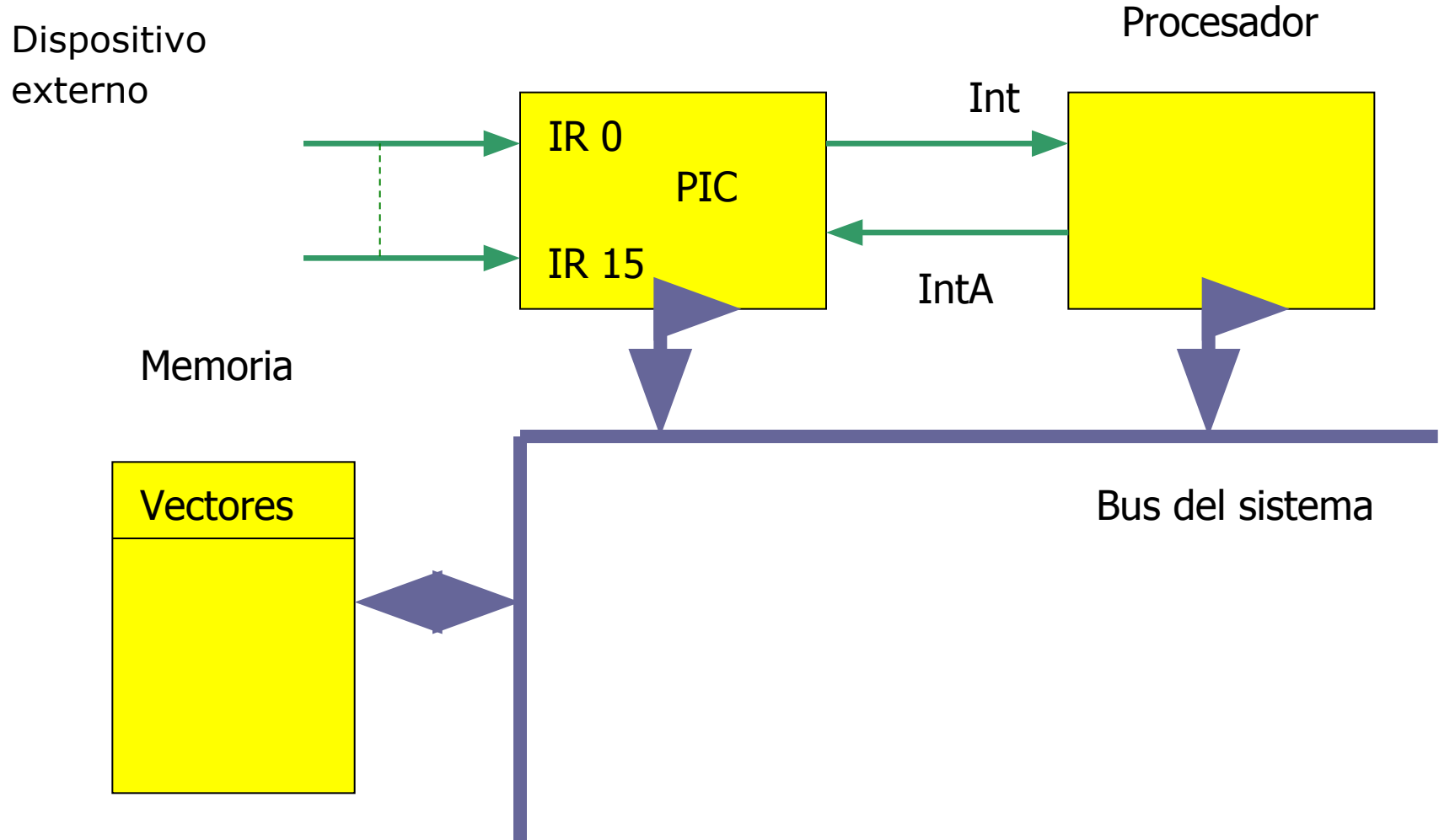
Ciclo de interrupción

- Añadido al ciclo de instrucción.
- El procesador comprueba si se ha generado alguna interrupción.
 - indicada por la presencia de una señal de interrupción.
- Si no hay señales de interrupción, capta la siguiente instrucción.
- Si hay alguna interrupción pendiente:
 - Se suspende la ejecución del programa en curso
 - Guarda su contexto (próxima instrucción a ejecutar y otros)
 - Carga el PC con la dirección de comienzo de una rutina de gestión de interrupción
 - Finalizada la rutina de gestión el procesador retoma la ejecución del programa del usuario en el punto de interrupción.

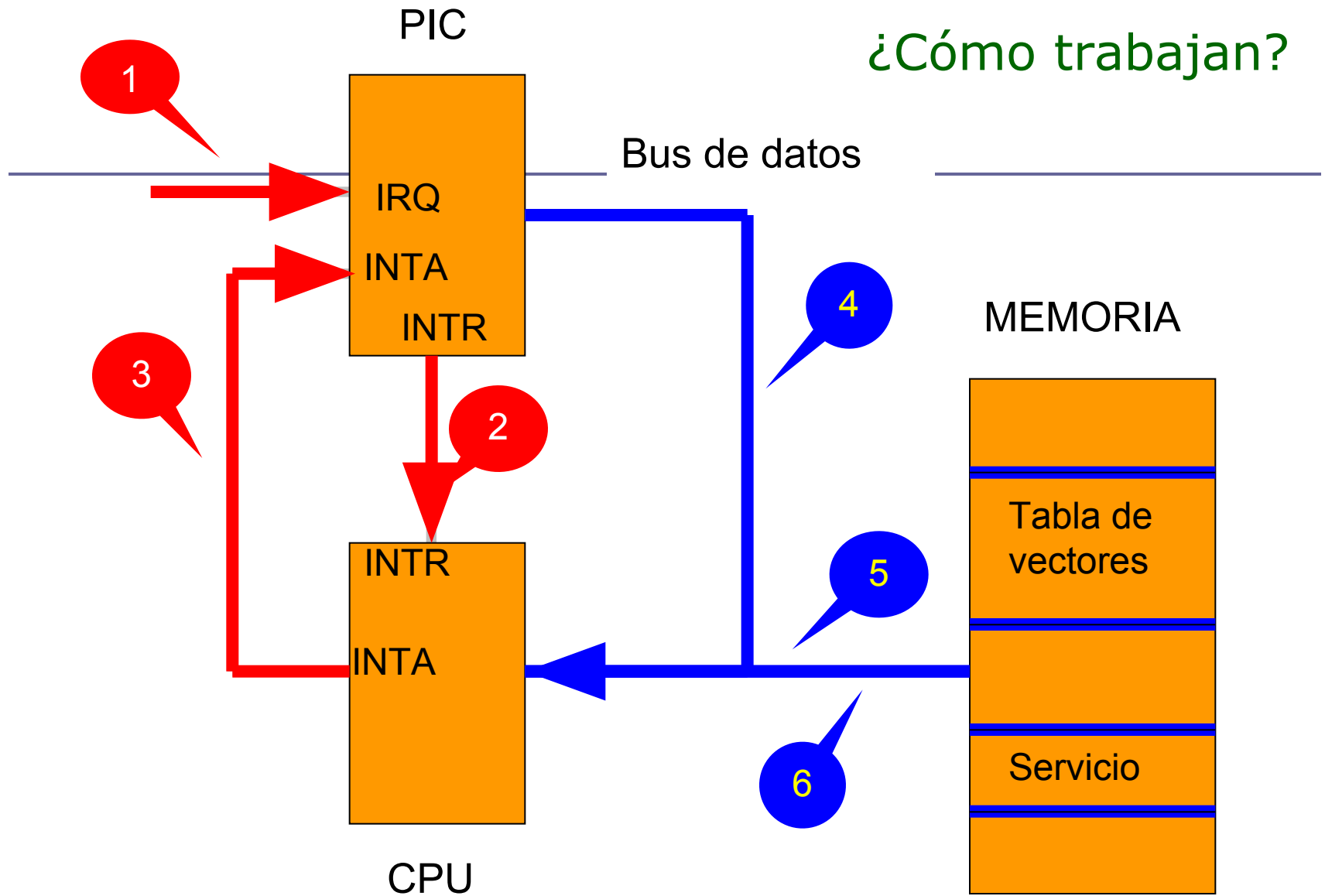
Escenario de trabajo

- El procesador tiene una única entrada de interrupciones.
- ¿ Cómo solucionamos tener varias entradas de interrupciones ?
- Existe un dispositivo : controlador de interrupciones (PIC)
- Interrupciones priorizadas

Conexionado



¿Cómo trabajan?



1. El PIC recibe un pedido de interrupción de un dispositivo externo y prioriza éste con otros pedidos que pueden llegar ó estar pendientes.
2. Un pedido de interrupción es enviado al procesador, por la línea INTR.
3. El procesador contesta por la línea INTA y solicita un puntero de 8 bits para la tabla.
4. El procesador recibe el puntero y lo usa para acceder a la tabla de vectores (direcciones) donde se encuentra la dirección del servicio.
5. Accede a la tabla y obtiene la dirección buscada.
6. Salva en la pila la dirección actual, los flags y salta a la dirección del servicio.

