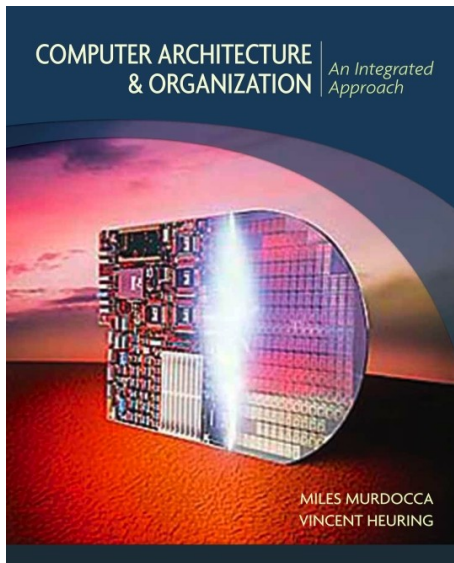


Organización de las Computadoras

Leonardo Giovanini



Gestión de periféricos

Interrupciones

Contenidos

10.1 Gestión de periféricos

- *Buses del sistema*
- *Gestión programada*
- *Gestión por acceso directo*

10.2 Gestión por interrupciones

10.3 Excepciones

10.4 Interrupciones

10.5 Implementaciones

Gestión de periféricos

Buses del sistema

Los sistemas informáticos generalmente constan de tres partes:

- i) la unidad central de procesamiento que procesa los datos;
- ii) la memoria que contiene los programas y los datos a procesar y
- iii) los dispositivos de entrada/salida que se comunican con el mundo exterior.



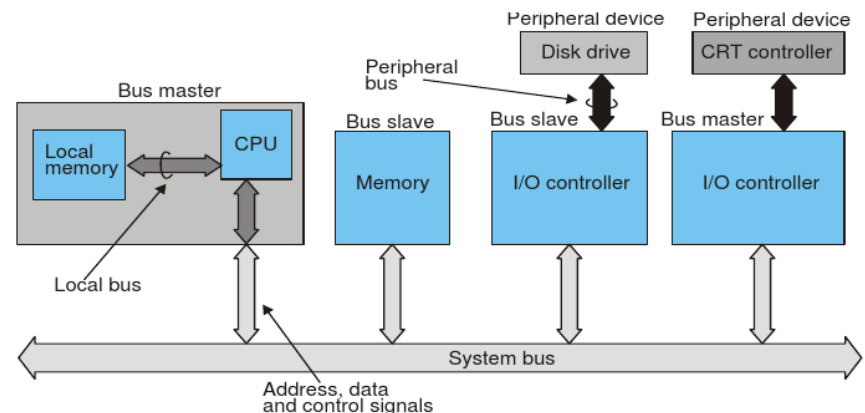
En arquitectura de computadoras, un **bus** es **un sistema de comunicación** que transfiere datos entre componentes dentro de una computadora o entre computadoras.

Dependiendo de la información transportada los buses se clasifican en:

Bus de datos - proporcionan una ruta para mover datos entre los módulos del sistema. Es bidireccional y su ancho, ancho de banda y latencia determinan el rendimiento general del sistema;

Bus de direcciones - designa el origen/destino de los datos en el bus de datos. Es unidireccional y su ancho determina la capacidad de memoria y periféricos posibles del sistema;

Bus de control - controla el acceso/uso de los datos y direcciones, y tiene un bidireccional. Transmite información de comando y sincronización entre los módulos del sistema. Las señales de comando especifican las operaciones, mientras que las de sincronización indican la validez de los datos y la información de la dirección.



Gestión de periféricos

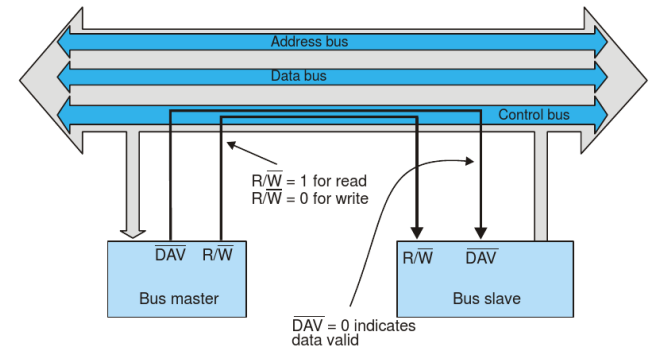
El **arbitraje de bus** refiere a la gestión de los buses del sistema durante su operación, coordinando las actividades de los dispositivos que solicitan transferencias de datos.

El controlador que tiene acceso a un bus en una instancia se conoce como bus master y su selección debe tener en cuenta las necesidades del sistema.

Hay dos enfoques para el arbitraje de buses:

Centralizado - un único árbitro gestiona el bus; y

Distribuido - todos los dispositivos pueden gestionar el bus.

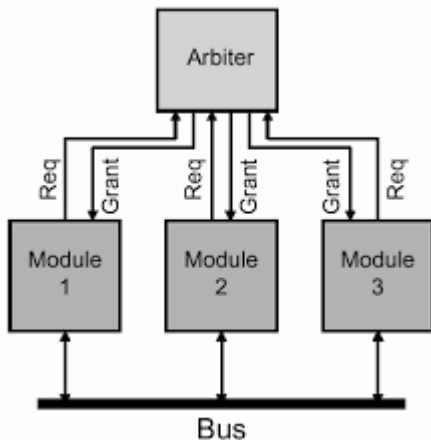


Hay tres tipos de transferencias que pueden realizarse:

- Memoria a periférico;
- Periférico a memoria; y
- Entre bloques de memoria.

Hay tres mecanismos para gestionar las transferencias de datos:

- **Gestión programada;**
- Gestión por interrupciones; y
- **Gestión por acceso directo.**

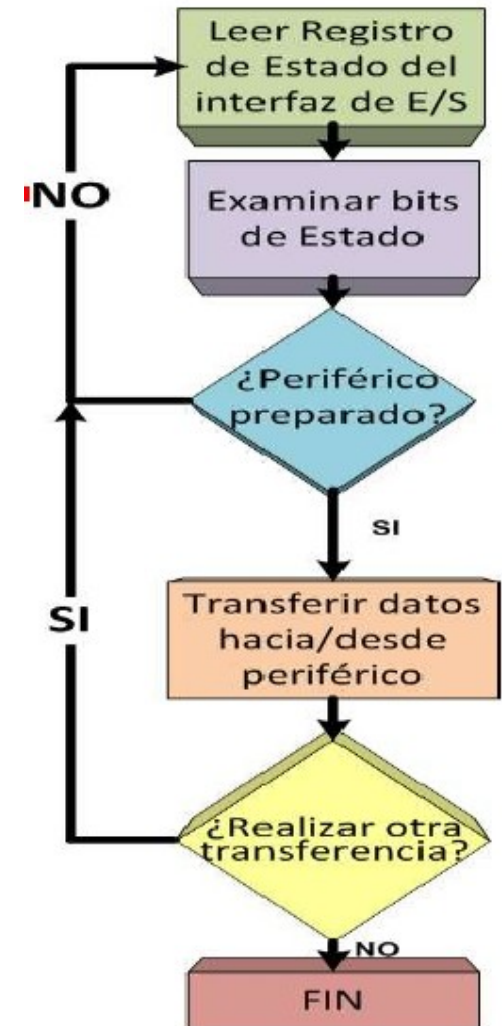
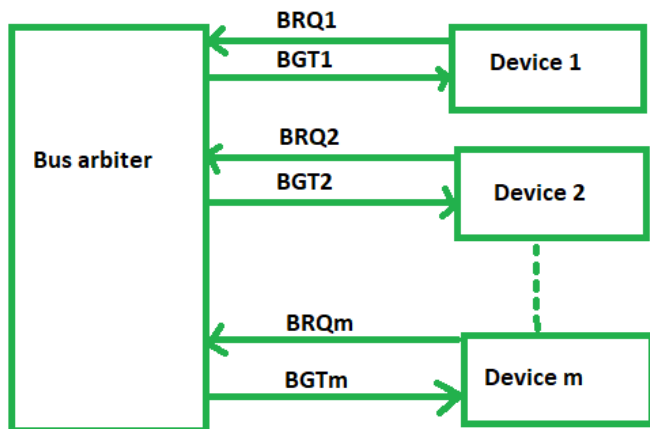


Gestión programada

Es un mecanismo por el cual las tareas necesarias para atender los periféricos se organiza en el programa. La rutina atiende secuencialmente cada uno de los periféricos hasta completar la lista de tareas.

Ventajas: se tiene control total sobre el proceso, tiene una alta repetibilidad en la ejecución, es robusto a los eventos externos sincronicos.

Inconvenientes: tiene altos tiempos de latencia, es ineficiente en el uso del tiempo, no puede atender eventos asincronicos y la carga de programacion es elevada.

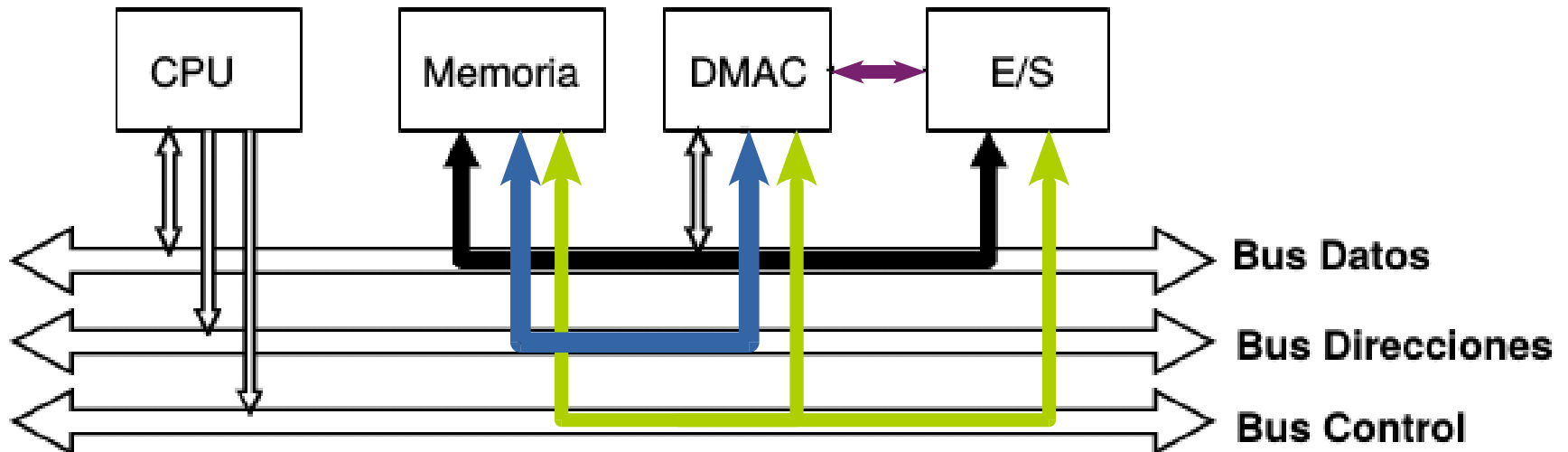


Gestión de periféricos

Gestión por acceso directo

Este mecanismo liberará al procesador durante la transferencia de los datos. El periférico avisa al controlador de DMA, el cual toma control de los buses y transfiere los datos.

El procesador programa al controlador de DMA antes de cada proceso de transferencia.



Ventajas - eficiente para grandes bloques de datos y periféricos muy rápidos;

Inconvenientes - detiene al procesador.

Gestión por interrupciones

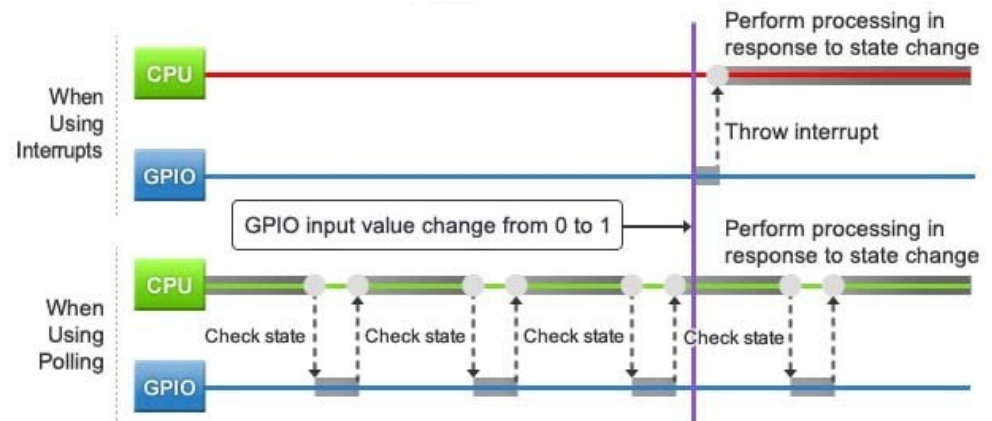
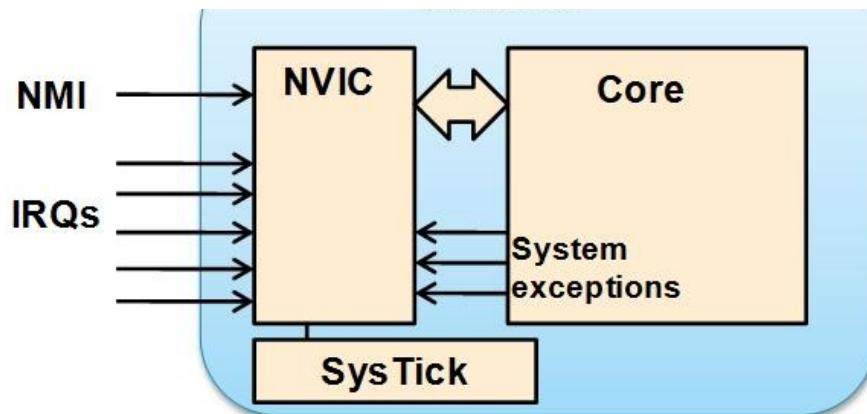
Gestión por interrupciones

Una interrupción es una señal (hardware o software) que indica al procesador que un evento de alta prioridad requiere la interrupción del código que el procesador está ejecutando.

El procesador responde suspendiendo sus actividades actuales, guardando su estado y ejecutando una rutina de servicio de la interrupción. Una vez que finalizada, el procesador reanuda las actividades normales.

Las interrupciones

- **Permiten responder a eventos sincronos y asincronos;**
- **Especificar las rutinas especifica de atención de cada evento; y**
- **Provee una forma simple de comunicación entre procesos.**



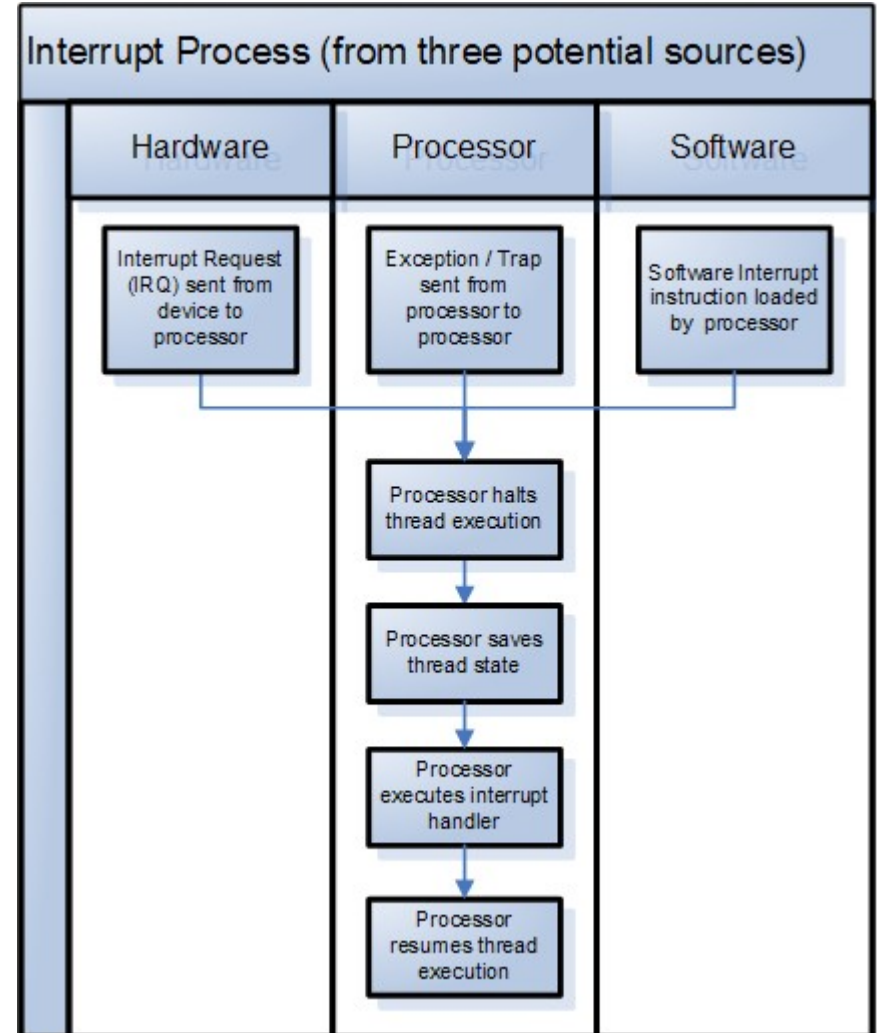
Ventajas: el procesador puede ejecutar otro código mientras espera un evento

Inconvenientes: dificulta el cálculo de los tiempos de ejecución y latencia, es complejo gestionar interrupciones anidadas.

Gestión por interrupciones

Las etapas de la ejecución de una interrupción son:

1. **Completar la ejecución** de la instrucción en curso (si se puede);
2. **Determinar la fuente** de la interrupción;
3. **Analizar** si se atiende, o no, la interrupción;
4. **Si se atiende, salvar la información** del programa en ejecución;
5. **Determinar la dirección** de la rutina de atención y **ejecutarla**; y
6. Una vez finalizada la ejecución de la rutina de interrupción **reasumir la ejecución** del programa.



Gestión por interrupciones

Las interrupciones se pueden clasificar según su **mecanismo de activación**

- **Excepciones** - se producen por **errores durante la ejecución de una instrucción o fallas en el hardware**, son **síncronas con el funcionamiento del procesador**. Pueden ser precisas (cuando se conoce su causa) o imprecisas (cuando no se conoce sus causas) y se pueden determinar analizando la traza del programa en curso. Normalmente son causadas al **realizarse operaciones no permitidas** (división por cero, desbordamiento de pila, acceso a posiciones de memoria no permitidas, tratar de ejecutar instrucción no reconocida) o falta de datos para ejecutar instrucciones (tabla de segmentos incompleta, violación de propiedades al acceder la memoria, segmento de código o dato no presente en la memoria).
- **Interrupciones por hardware (IRQ)** - son producidas por **eventos externos al programa**, son **asíncronas a la ejecución del programa** (se pueden producir en cualquier momento independientemente de lo que esté haciendo el procesador). Las causas que las producen **son externas al procesador y están relacionadas con los dispositivos de entrada / salida**.
- **Interrupciones por software (TRAP)** - son producidas por el **programa en ejecución**. Para generarlas existen distintas instrucciones en el **código de máquina** que permiten al programador producir una interrupción.

Excepciones

Excepciones

Las excepciones son eventos que **rompe la secuencia** de ejecución del programa para **atender errores que ocurren durante la ejecución de una instrucción o fallas** en el hardware de la computadora.

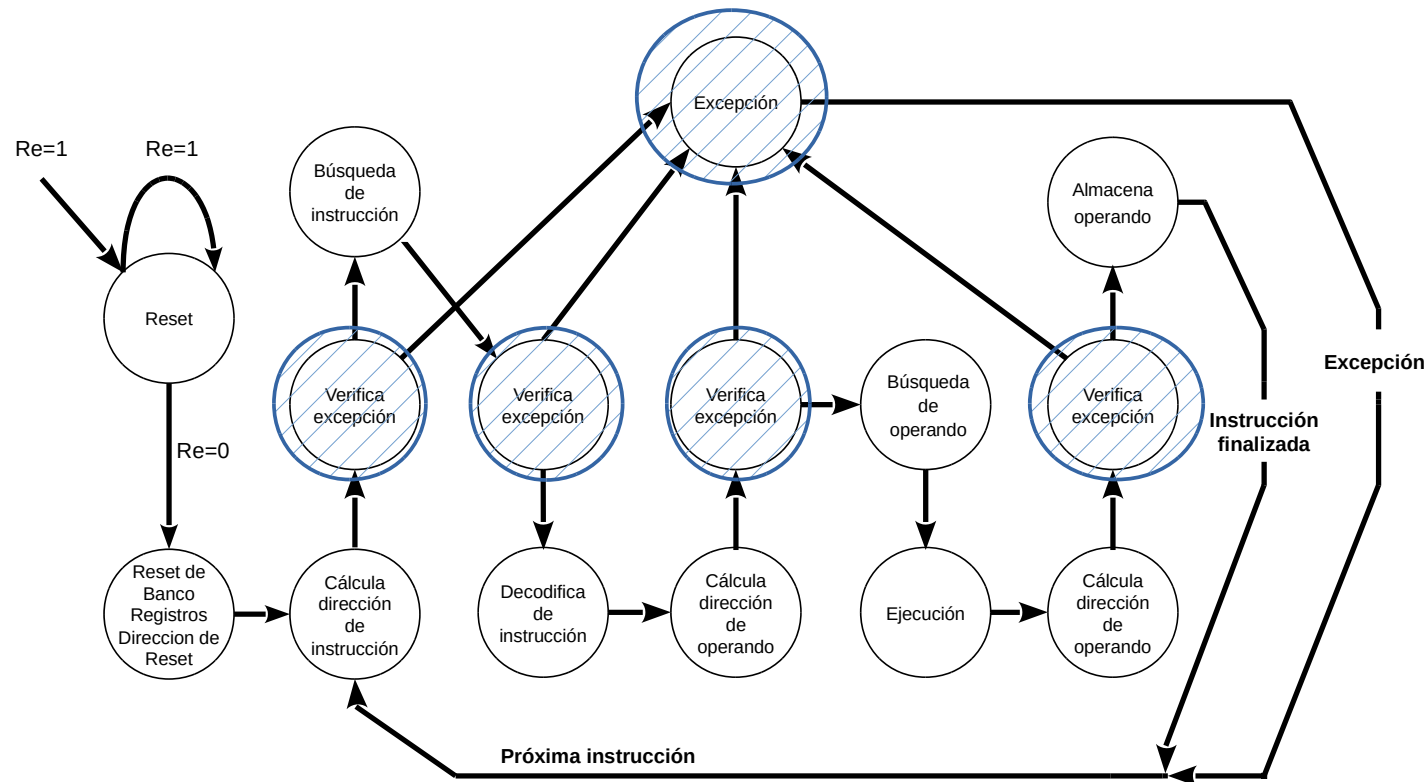
Son eventos **síncronos** con el funcionamiento del procesador **y** se clasifican en **precisas** (cuando se conoce su causa) o **imprecisas** (cuando no se conoce sus causa), en cuyo caso la causa se determina analizando la traza del programa en curso.

Las causas de excepciones son

- Realizar **tareas de depuración** del programa (breakpoint, trace, etc.);
- Realizar **operaciones no permitidas** (división por cero, códigos desconocidos);
- **Accesos ilegales a la memoria** (desbordamiento de pila, acceso a posiciones de memoria no permitidas, operaciones no permitidas, violación de privilegios); y
- **Falta de datos para ejecutar instrucciones** (tabla de segmentos incompleta, segmento no presente en la memoria).

Excepciones

Para implementar excepciones, se agrega etapas de detección de causas de excepciones a lo largo del ciclo de instrucción, la verificación de excepciones chequean al final del ciclo de ejecución si se produce alguna de las condiciones de excepción.



Si se detecta la ocurrencia de una excepción, el procesador suspende la ejecución de la instrucciones, guarda la dirección de la instrucción que se está ejecutando y carga el contador de programa con la dirección de la rutina de atención de la excepción.

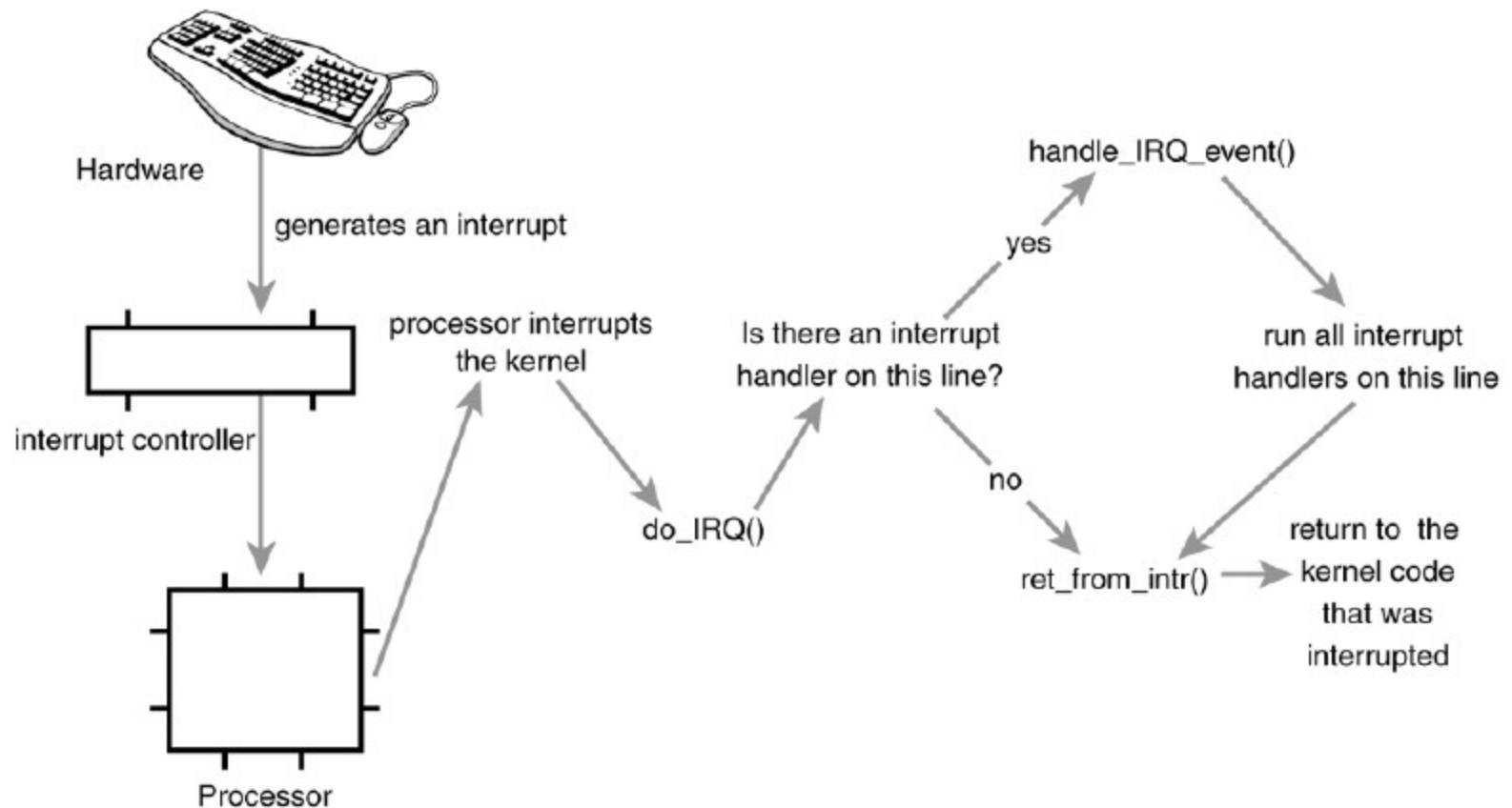
La gestión de las excepciones es similar a la gestión de las interrupciones.

Interrupciones

Interrupciones

Una interrupción es un mecanismo que **rompe la secuencia** de ejecución del programa para atender un evento **no previsto, asincronico y externo** a la ejecución del programa.

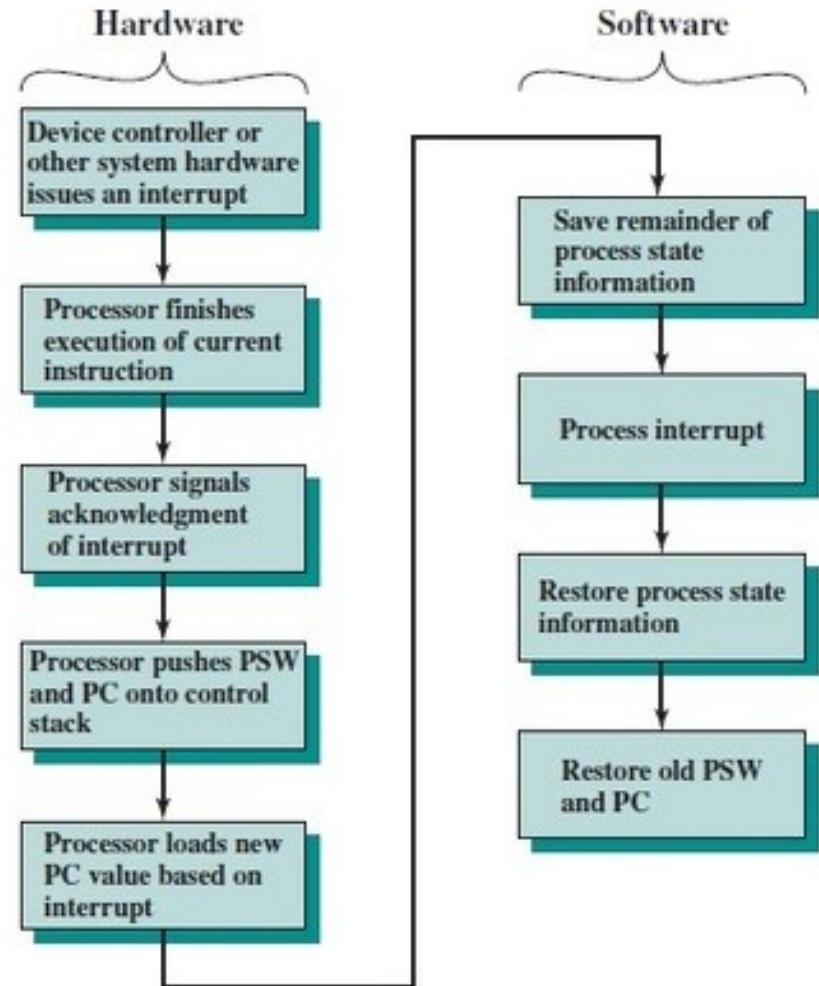
Son generadas por los dispositivos de la computadora



Interrupciones

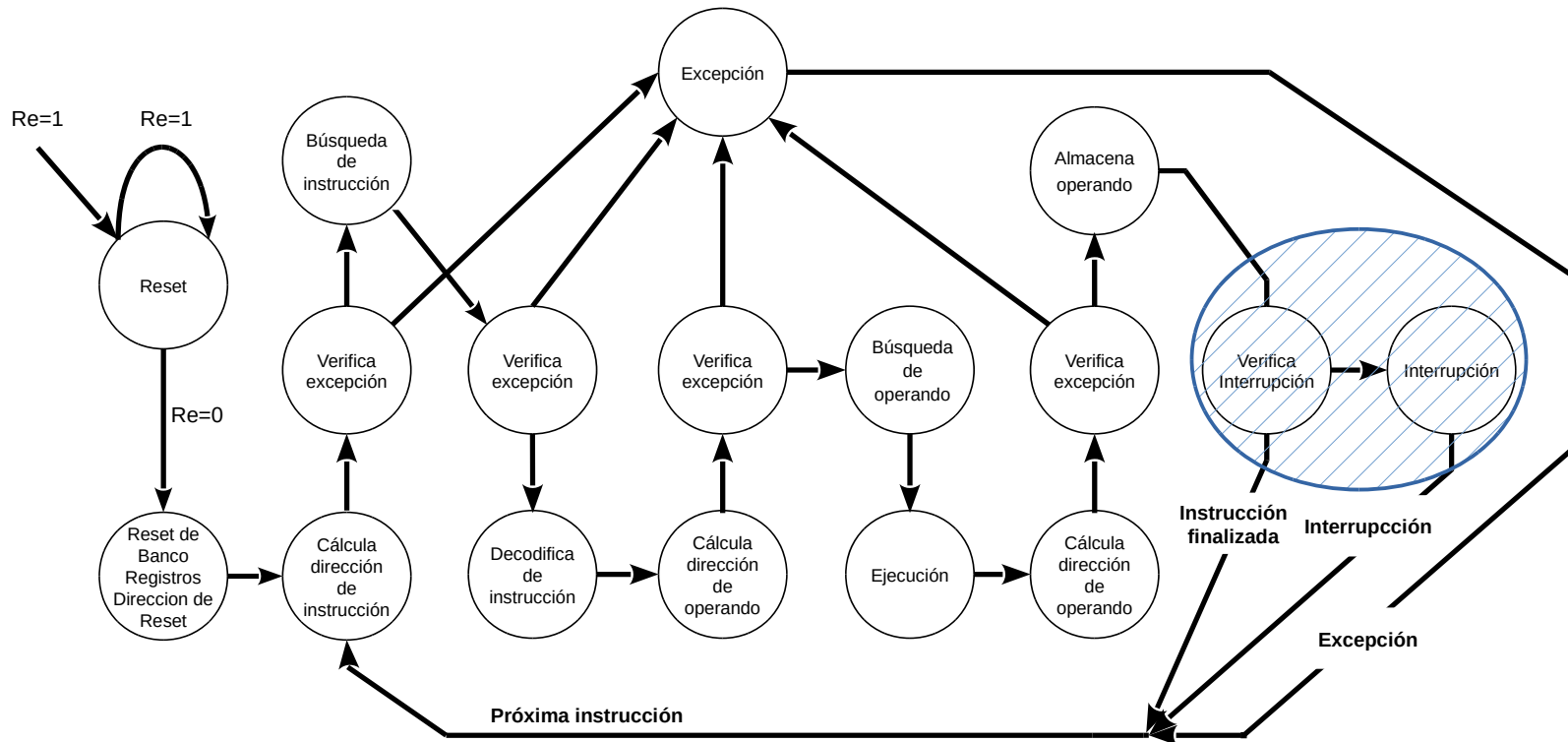
Las etapas de la ejecución de una interrupción son:

1. **Completar la ejecución** de la instrucción en curso;
2. **Determinar la fuente** de la interrupción;
3. **Analizar** si se atiende, o no, la interrupción;
4. Si se atiende, **salvar la información** del programa en ejecución;
5. **Determinar la dirección** de la rutina de atención y **ejecutarla**; y
6. Una vez finalizada la ejecución de la rutina de interrupción **reasumir la ejecución** del programa.



Interrupciones

Para implementar interrupciones, se agrega la etapa de detección de interrupciones al final del ciclo de instrucción.



Si se detecta la ocurrencia de una interrupción, el procesador suspende la ejecución de la próxima instrucción, guarda la dirección de la próxima instrucción a ejecutar y carga el contador de programa con la dirección de la rutina de atención de la interrupción.

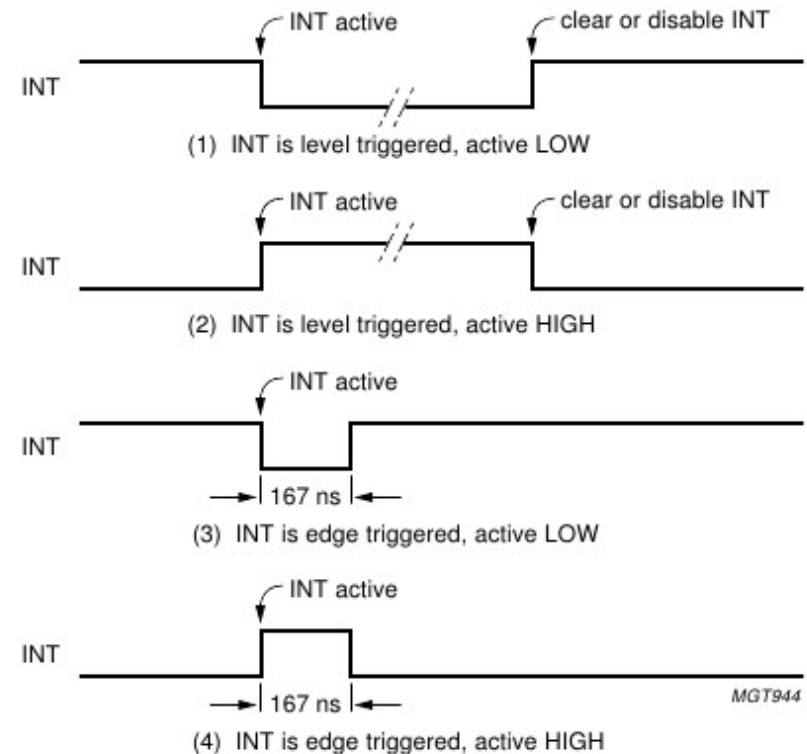
Interrupciones

Clasificación

Las interrupciones por hardware pueden clasificarse según su **mecanismo de activación** en:

- **Disparadas por nivel:** es una interrupción que se activa cambiando y manteniendo el nivel de la línea de interrupción hasta que sea atendida.
- **Dispara por flancos:** es una interrupción que se activa por cambios en los niveles de la línea de interrupción, ya sea a través de un flanco de ascenso o descenso, y volviendo la línea a su estado inactivo.
- **Disparadas por mensajes:** es una interrupción activada por cambios en la condición de la presencia de un mensaje específico.

Típicamente, múltiples mensajes de solicitud pendientes son unificados, como las interrupciones activadas por flancos

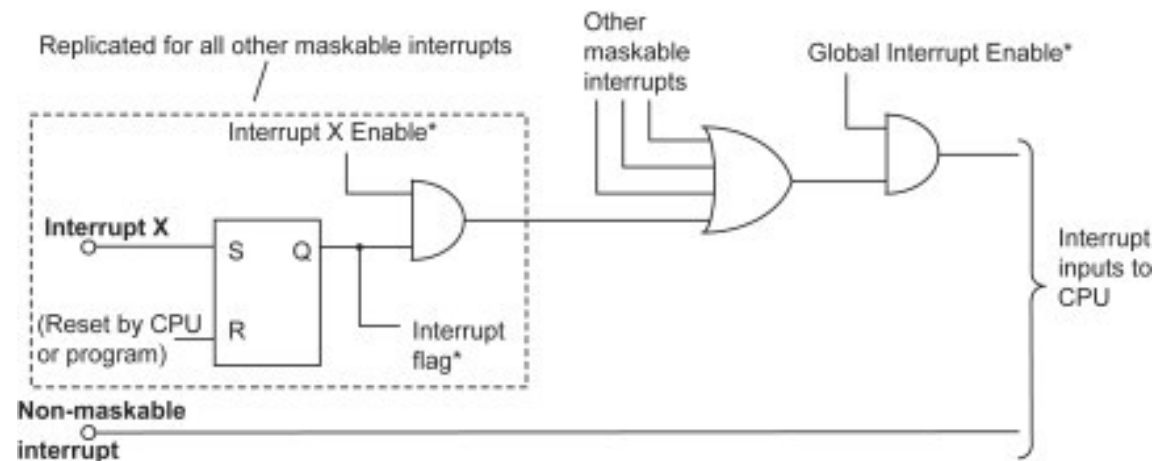


Interrupciones

Clasificación

Las interrupciones pueden clasificarse según su **mecanismo de gestión** en:

- **Interrupciones enmascarable (IRQ):** es una interrupción de hardware que puede ser ignorada o bloqueada con señales de enmascaramiento almacenadas en registros especiales.
- **Interrupciones no enmascarable (NMI):** es una interrupción de hardware que no tiene asociado un mecanismo de enmascaramiento. Las NMIs son utilizadas por las tareas de mas alta prioridad como temporizadores y watchdog timers.
- **Interrupciones inter-procesadores (IPI):** es un tipo especial de interrupción que es generada por procesador para interrumpir a otro procesador en sistemas multiprocesadores.



Interrupciones

Determinación de la fuente

Hay distintas formas de **identificar la fuente** de una interrupción:

- **Consulta:** el procesador comprueba de manera sistemática todos los dispositivos de manera que busca cuál fue el que solicitó la interrupción. El orden de consulta define la prioridad de las interrupciones.

Para implementarla, la computadora incorpora registros que guardan la información de las peticiones de interrupción (*Interrupt Flag*).

Ventajas: Usa poco hardware, es muy flexible y establece una prioridad en los dispositivos (el orden de sondeo); ya que la consulta se implementa en software;

Desventajas: Es lento porque tiene que comprobar todos los dispositivos.

PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

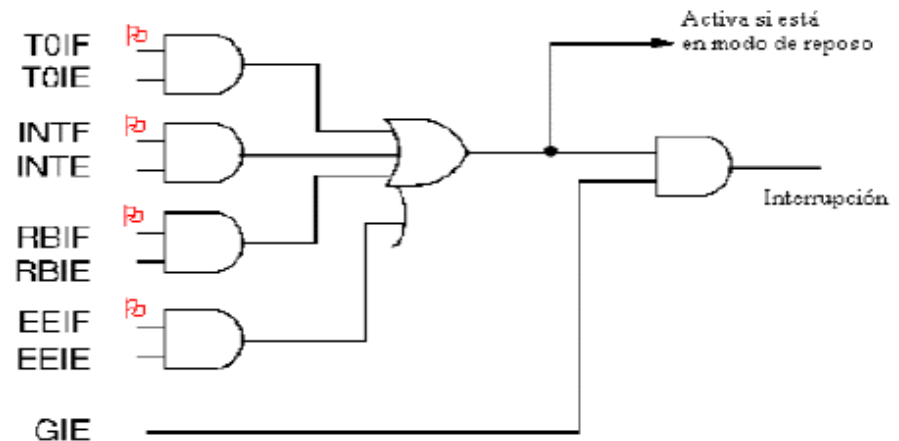
R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7				bit 0			

PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF
bit 7				bit 0			

PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

U-0	U-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF
bit 7				bit 0			

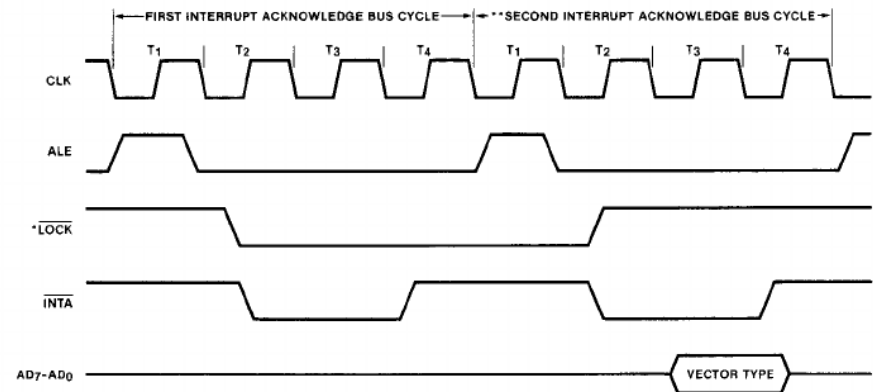
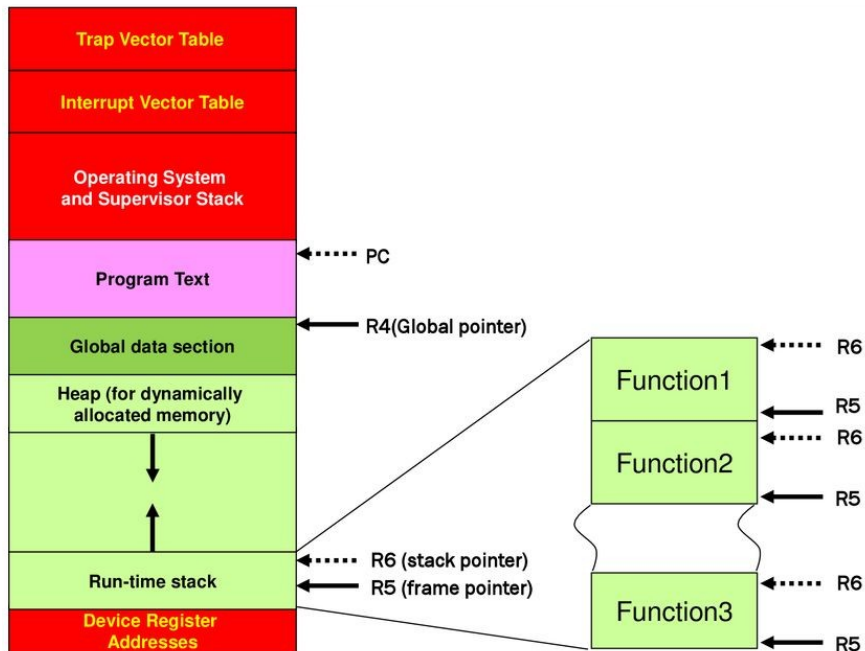


Interrupciones

Determinación de la fuente

- **Vectorizadas:** cada interrupción esta asociada con un elemento de una tabla de vectores de interrupción a través de un vector utilizado para sincronizar el procesador con el periférico.

El periférico le indica al procesador la ubicación de la dirección efectiva en la tabla de vectores a través del vector de interrupción (Vector Type) y el procesador envía indica la aceptación de la interrupción (INTA).



La tabla de vectores es un área de memoria que contiene todos los vectores de interrupción de un procesador. La ubicación de la tabla y el orden de los vectores dentro de ella, son una característica del procesador y no se pueden modificar.

Cada vector tiene un tamaño fijo que permite, almacenar una instrucción de salto al comienzo de la rutina de servicio de esa interrupción.

Ventajas: Procesa múltiples interrupciones utilizando poco hardware;

Desventajas: Es lenta y utiliza muchos accesos a los buses.

Interrupciones

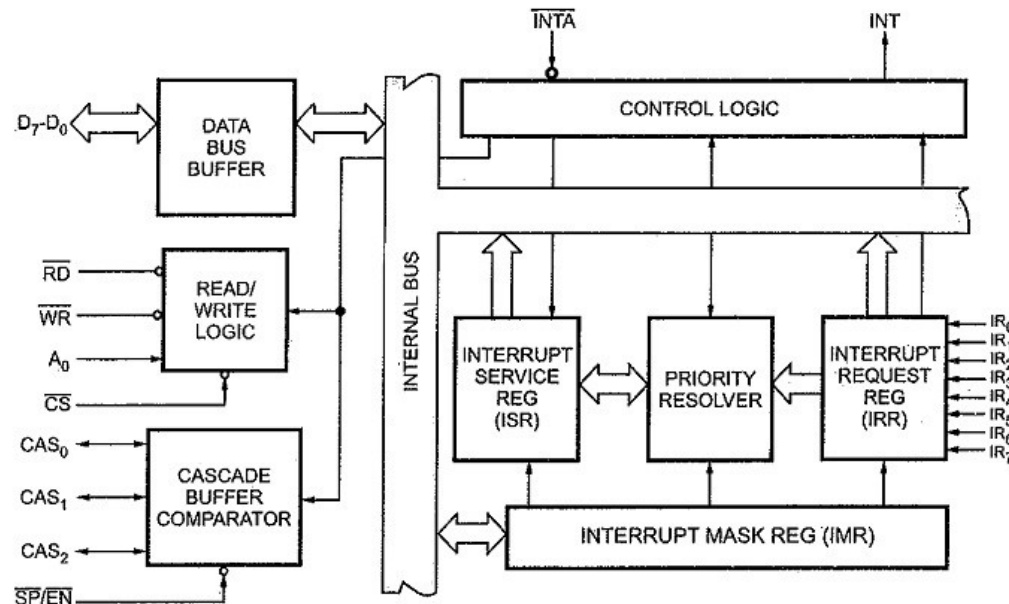
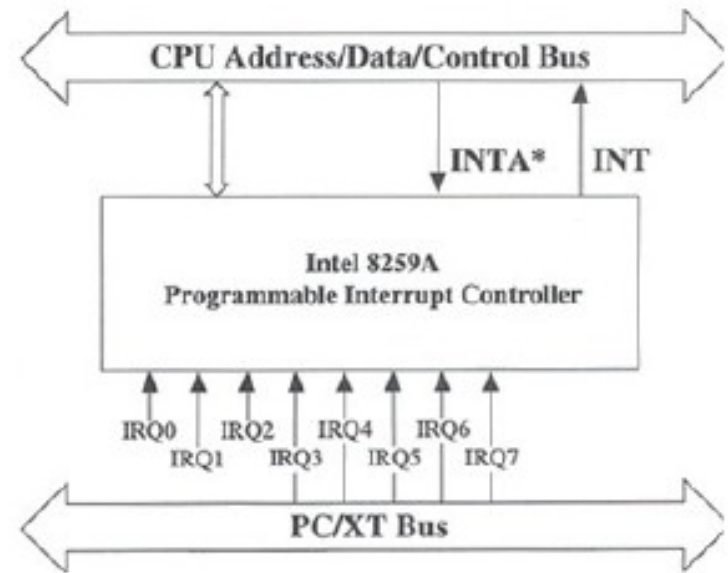
Determinación de la fuente

- **Hardware:** se utiliza un circuito dedicado a gestionar las interrupciones cuyas señales de petición se controlan de forma independiente para cada periférico.

Dispone de registros específicos para definir las prioridades y el enmascaramiento individual de cada línea de petición.

Ventajas: Es muy eficiente, veloz y flexible;

Desventajas: Es caro.

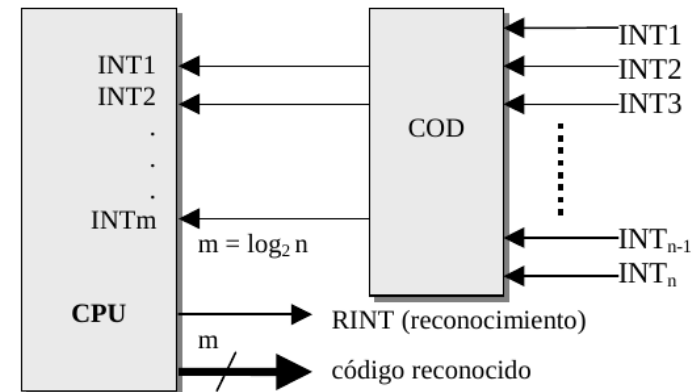
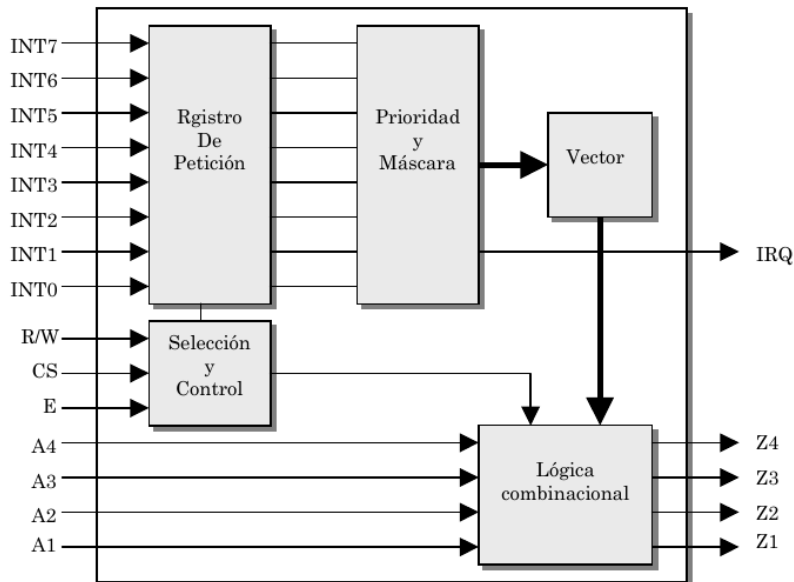


Interrupciones

Gestion de prioridad

El sistema necesita un mecanismo para priorizar las interrupciones y tratar primero las más urgentes. Para ello, existen varias alternativas:

- **Interrupciones simultáneas:** hay dos alternativas para gestionar varias interrupciones activas:
 - Un dispositivo específico que combina varias fuentes de interrupción en una o más líneas, al tiempo que permite asignar los niveles de prioridad las salidas de interrupción.



Cuando el dispositivo tiene múltiples salidas de interrupción para generar, las genera en orden de prioridad relativa. Los modos comunes de un controlador de interrupciones incluyen prioridades duras, rotativas y en cascada.

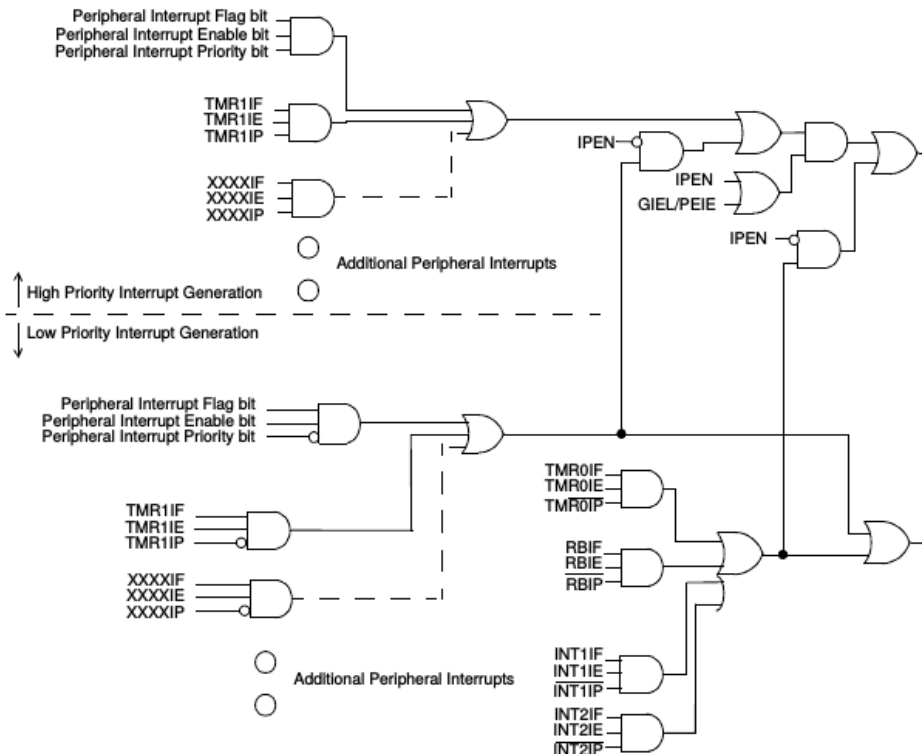
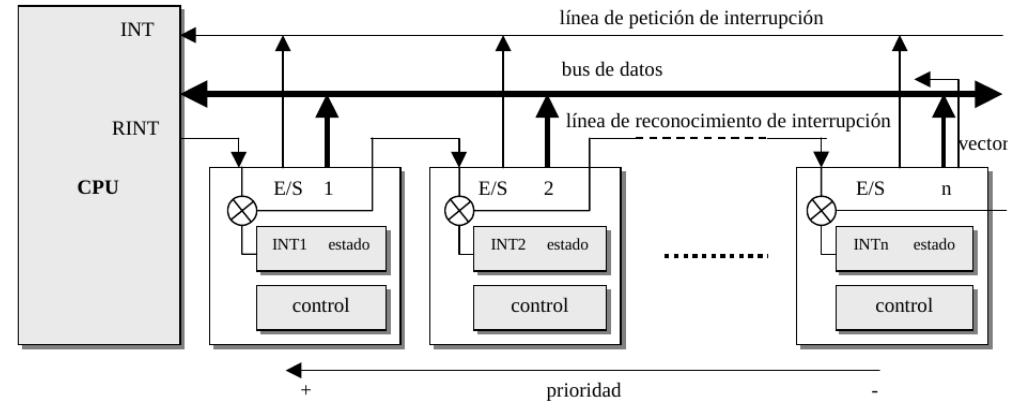
A menudo permiten la conexión en cascada de sus salidas a entradas entre sí para expandir sus capacidades.

Interrupciones

Gestion de prioridad

- Un **método distribuido** que se implementa en la práctica a través de dos técnicas;

i) **Consulta** – es un metodo de gestion en el cual una única rutina debe determinar la fuente de la interrupción al verificar todas las posibles fuentes de interrupción;



ii) **Conexión en cadena (Daisy Chain)** – es un metodo de gestion en el que los dispositivos están conectados en forma serial. La solicitud de línea de interrupción es común a todos los dispositivos.

La prioridad del dispositivo esta dada por su posicion relativa en la cadena. Mientras mas cerca del procesador, mayor la prioridad.

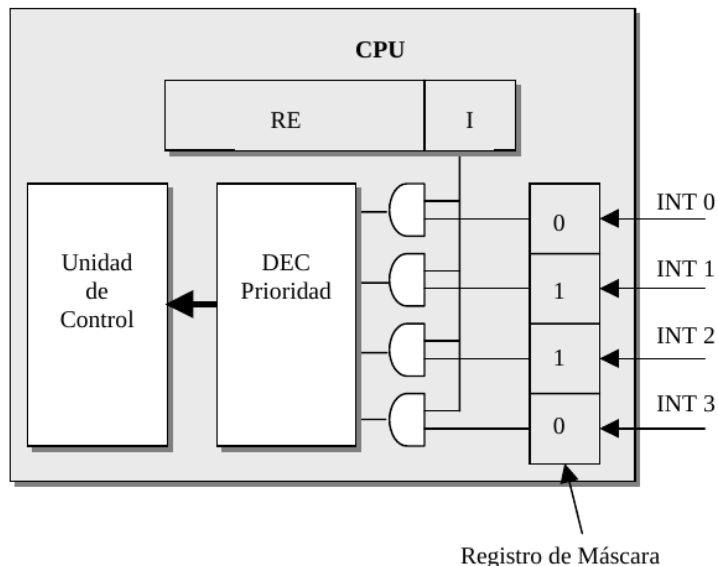
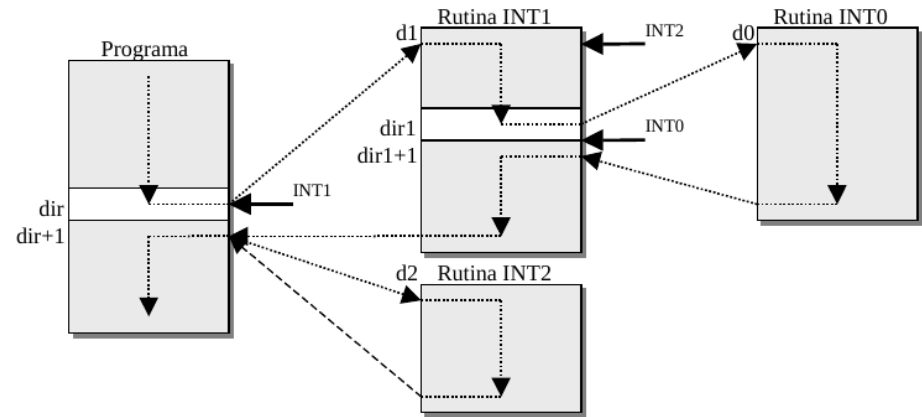
Si algún dispositivo solicita la interrupcion con un estado de nivel bajo, la línea de interrupción pasa al estado de nivel bajo y habilita la entrada de interrupción en el procesador.

Interrupciones

Gestion de prioridad

Interrupciones anidadas - en un sistema de interrupción anidado se permite una interrupción en cualquier momento y en cualquier lugar, incluso cuando la rutina de atención de una interrupción se está ejecutando.

Solo la rutina de atención de la interrupción (ISR) de mayor prioridad será ejecutada inmediatamente. Luego, el segundo ISR de prioridad más alta se ejecutará después de que se complete el más alto.



Inhibición de interrupciones - las interrupciones pueden ser desactivadas por el programador en momentos en que el procesador está ejecutando rutinas críticas que no se puede interrumpir. Si llega una interrupción, el procesador la ignora o atiende dependiendo del estado del bit asociado del registro de máscara.

Una **interrupción no enmascarable** (NMI) no puede deshabilitarse y cuando llega, el procesador la atiende. Se utiliza en caso de emergencias o fallas.

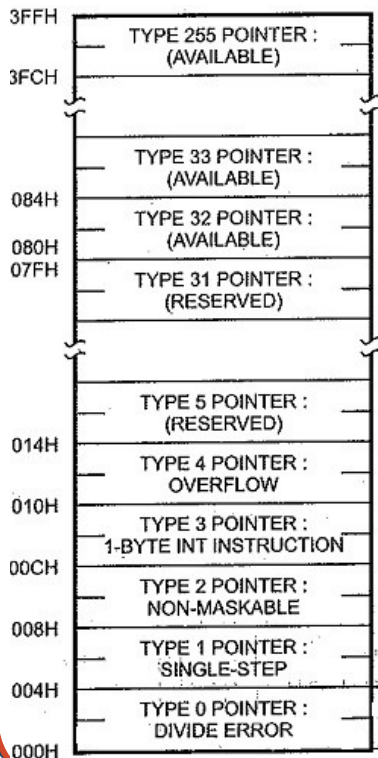
Interrupciones

Dirección de ejecución de la rutina

Hay dos alternativas para determinar la dirección de la rutina de servicio:

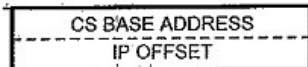
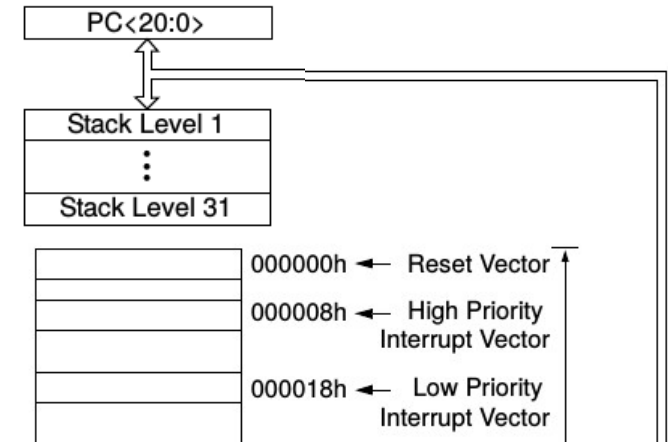
- **Direcciones fijas:** se hallan cableadas en el procesador y por tanto nunca pueden ser cambiadas.

Esto implica que las rutinas de servicio de la interrupciones siempre estarán en una posición de memoria fija.



- **Direcciones variables:** no se halla cableada en el procesador de manera que el dispositivo debe dar información acerca de la localización de la dirección de comienzo de la rutina de servicio asociada a dicho periférico. Las alternativas más utilizadas son:

- **Direccionamiento absoluto:** el dispositivo o la interfaz se encarga de conocer y de enviar la dirección de la rutina al procesador para que éste la localice y ejecute.
- **Direccionamiento indirecto:** se mantiene una tabla de vectores de interrupción (direcciones de comienzo de las distintas rutinas) y a cada interrupción se le asocia un índice por el cual se accederá a la tabla y se recuperará la dirección de comienzo.



Implementaciones

Implementacion

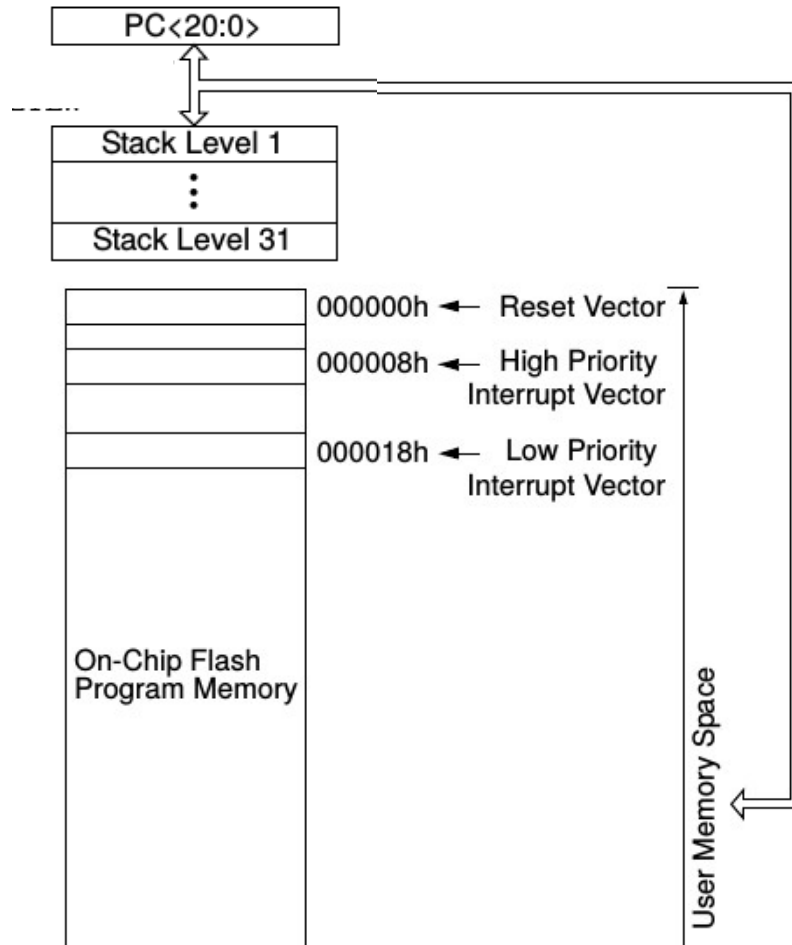
RISC-V

Implementacion

RISC-V

Implementación Microchip

Organización de la memoria



INTCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7				bit 0			

PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7				bit 0			

PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF
bit 7				bit 0			

PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

U-0	U-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF
bit 7				bit 0			

PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7				bit 0			

PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE
bit 7				bit 0			

IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

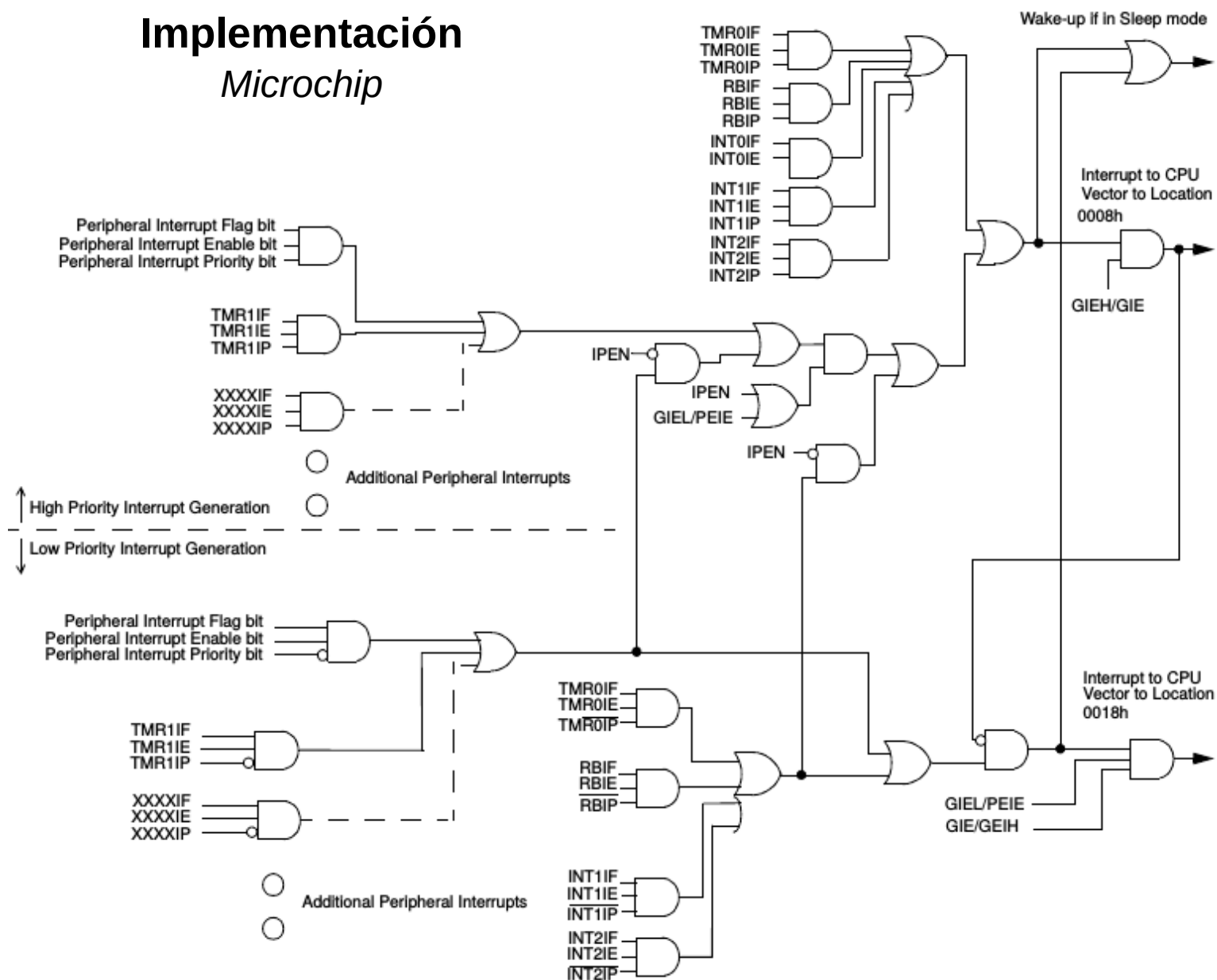
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP ⁽¹⁾	ADIP	RC1IP	TX1IP	SSPIP	CCP1IP	TMR2IP	TMR1IP
bit 7				bit 0			

IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

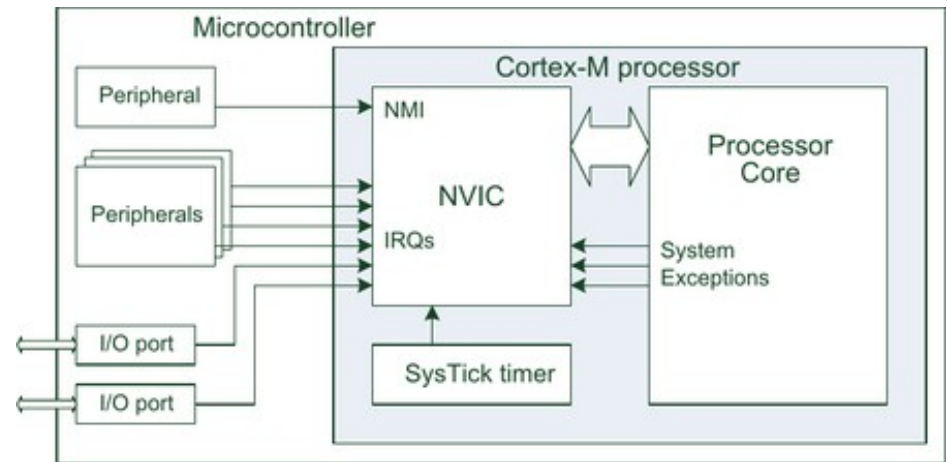
U-0	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP
bit 7				bit 0			

Recursos de gestión

Implementación Microchip



Implementación ARM

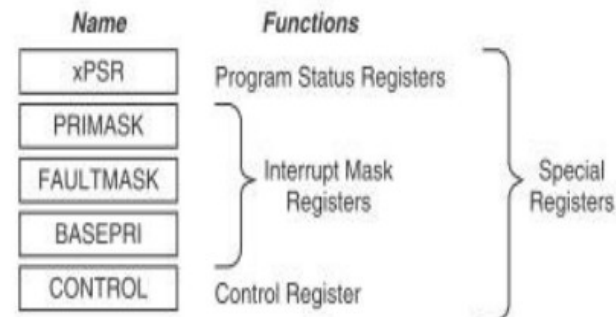
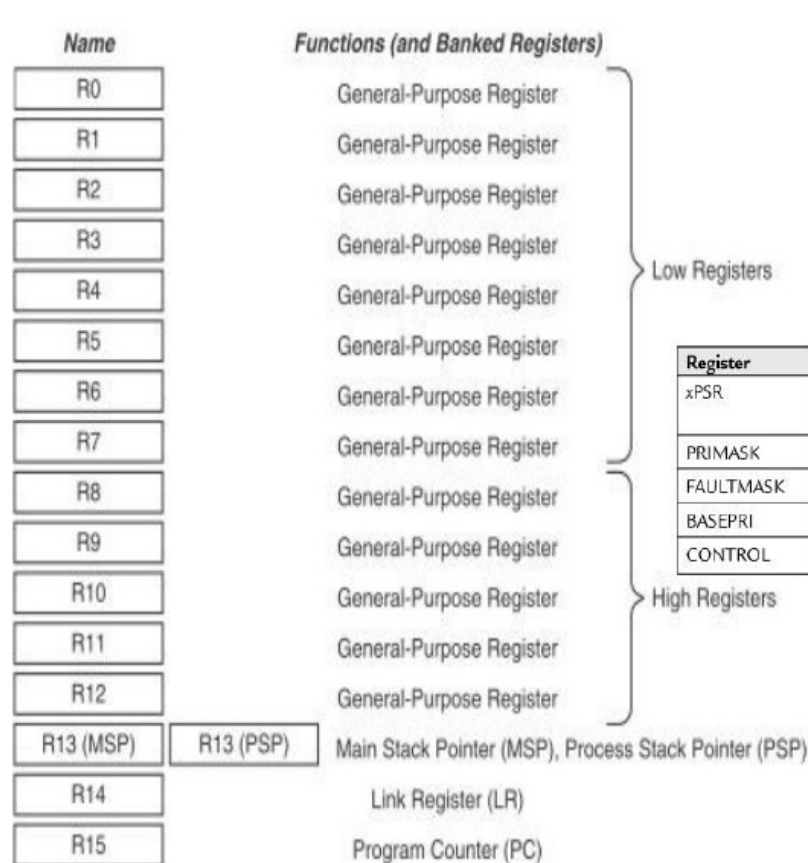


Interrupt #	Interrupt	Memory Location	Priority Level
0	Stack Pointer Initial Value	0x00000000	
1	Reset	0x00000004	-3 Highest
2	NMI	0x00000008	-2
3	Hard Fault	0x0000000C	-1
4	Memory Management Fault	0x00000010	Programmable
5	Bus Fault	0x00000014	Programmable
6	Usage Fault (undefined instructions, divide by zero, unaligned memory access,)	0x00000018	Programmable
7	Reserved	0x0000001C	Programmable
8	Reserved	0x00000020	Programmable
9	Reserved	0x00000024	Programmable
10	Reserved	0x00000028	Programmable
11	SVCall	0x0000002C	Programmable
12	Debug Monitor	0x00000030	Programmable
13	Reserved	0x00000034	Programmable
14	PendSV	0x00000038	Programmable
15	SysTick	0x0000003C	Programmable
16	IRQ for peripherals	0x00000040	Programmable
17	IRQ for peripherals	0x00000044	Programmable
...
255	IRQ for peripherals	0x000003FC	Programmable

Controlador de Interrupciones

- Los registros de control, prioridad y vectores están almacenados en memoria;
- Contiene registros de control para las excepciones, la MMU, el temporizador SYSTICK y la lógica de depuración;
- Soporta hasta 240 entradas externas de interrupción (IRQ); y
- Una entrada no enmascarable (NMI).

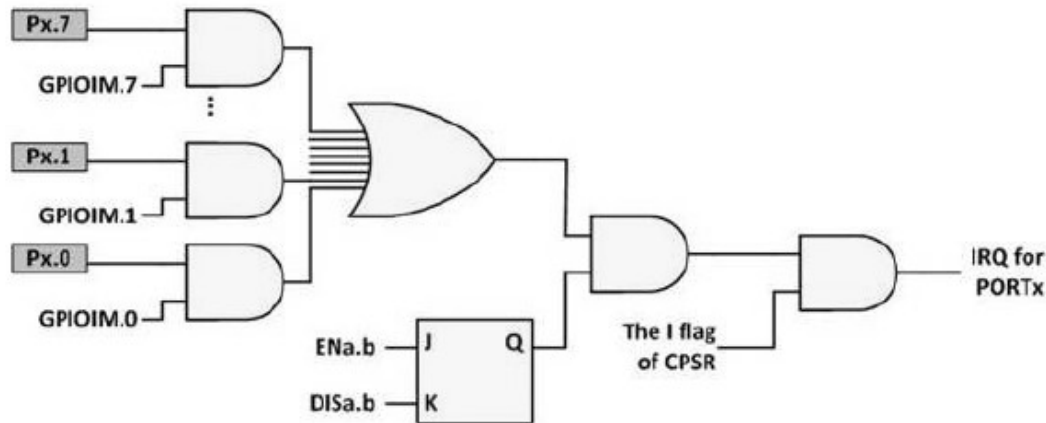
Implementación - ARM



Register	Function
xPSR	Provide ALU flags (zero flag, carry flag), execution status, and current executing interrupt number
PRIMASK	Disable all interrupts except the nonmaskable interrupt (NMI) and HardFault
FAULTMASK	Disable all interrupts except the NMI
BASEPRI	Disable all interrupts of specific priority level or lower priority level
CONTROL	Define privileged status and stack pointer selection

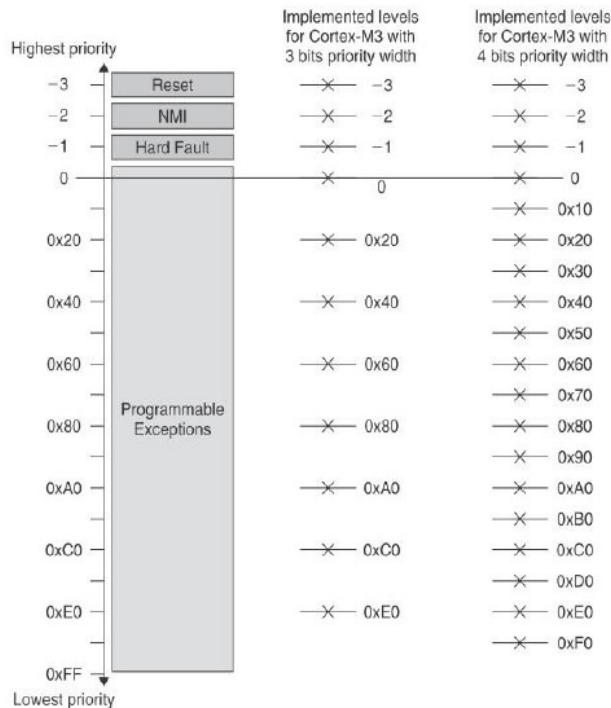
Address	Name	Type	Reset	Description
0xE000E004	ICTR	RO	-	Interrupt Controller Type Register, ICTR
0xE000E100 - 0xE000E11C	NVIC_ISER0 - NVIC_ISER7	RW	0x00000000	Interrupt Set-Enable Registers
0xE000E180 - 0xE000E19C	NVIC_ICER0 - NVIC_ICER7	RW	0x00000000	Interrupt Clear-Enable Registers
0xE000E200 - 0xE000E21C	NVIC_ISPR0 - NVIC_ISPR7	RW	0x00000000	Interrupt Set-Pending Registers
0xE000E280 - 0xE000E29C	NVIC_ICPR0 - NVIC_ICPR7	RW	0x00000000	Interrupt Clear-Pending Registers
0xE000E300 - 0xE000E31C	NVIC_IABR0 - NVIC_IABR7	RO	0x00000000	Interrupt Active Bit Register
0xE000E400 - 0xE000E41F	NVIC_IPR0 - NVIC_IPR59	RW	0x00000000	Interrupt Priority Register

Implementación - ARM



Las interrupciones están deshabilitadas después del Reset, para habilitarlas hay que:

1. Habilitar la interrupciones específicas de cada módulo;
2. Habilitar las interrupciones en el controlador NVIC; y
3. Habilitar las interrupciones globalmente.



Available Priority Levels with 3-Bit or 4-Bit Priority Width

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Implemented				Not implemented, read as zero			

Bit 7,6,5,4	Bit 3,2,1,0	Value
0000	0000	0x00
0010	0000	0x20
0100	0000	0x40
0110	0000	0x60
1000	0000	0x80
1010	0000	0xA0
1100	0000	0xC0
1110	0000	0xE0

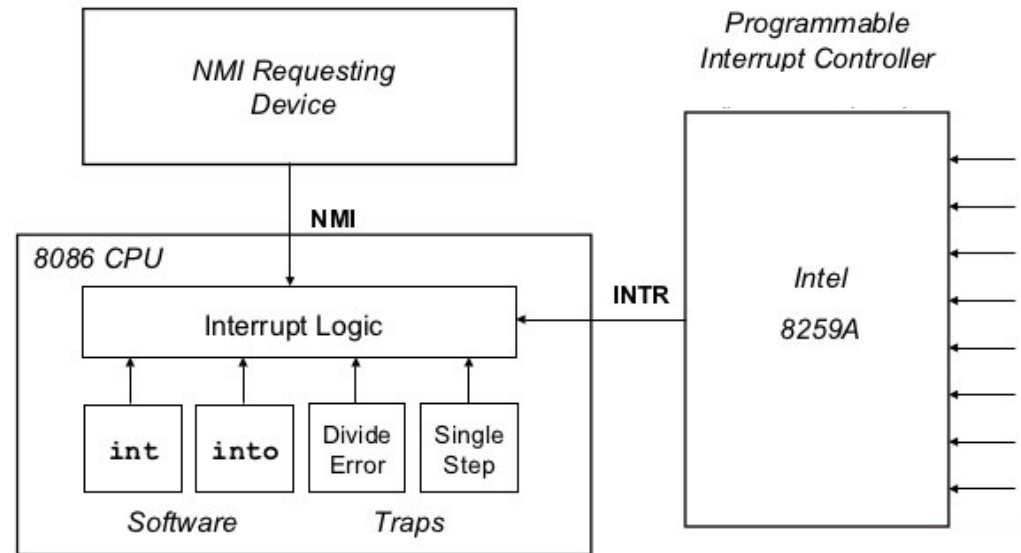
A Priority Level Register with 3-bit Implemented

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Implemented				Not implemented, read as 0			

A Priority Level Register with 4-bit Implemented

Implementación INTEL

080H	32-255 User defined	
	14-31 Reserved	
040H	Coprocessor error	16
03CH	Unassigned	15
038H	Page fault	14
034H	General protection	13
030H	Stack seg overrun	12
02CH	Segment not present	11
028H	Invalid task state seg	10
024H	Coproc seg overrun	9
020H	Double fault	8
01CH	Coprocessor not avail	7
018H	Undefined Opcode	6
014H	Bound	5
010H	Overflow (INTO)	4
00CH	1-byte breakpoint	3
008H	NMI pin	2
004H	Single-step	1
000H	Divide error	0



Controlador de Interrupciones

- Los registros de control y prioridades están disponibles en el área de entrada/salida;
- La tabla de vectores se encuentra en memoria;
- Contiene vectores para la MMU y la lógica de depuración;
- Soporta hasta 250 entradas externas de interrupción (IRQ);
- Una entrada no enmascarable (NMI); y
- Una entrada por software (INT).

Implementación INTEL

Las interrupciones están deshabilitadas después del Reset, para habilitarlas hay que:

1. Configurar la interrupciones específicas de cada módulo;
2. Habilitar las interrupciones en el controlador; y
3. Habilitar las interrupciones globalmente.

