

## **INTERRUPCIONES**

### **CONCEPTO DE INTERRUPCIONES**

Una interrupción es una operación del hardware que indica a la CPU que detenga la ejecución de lo que estaba haciendo, conserva la situación del proceso y envía a la CPU a una dirección determinada en la que tiene un programa de control de interrupción o de servicio de interrupciones.

### **CAUSAS DE INTERRUPCIONES**

- \* Error de máquina detectado
- \* Error de programa
- \* Interrupción externa (el operador pide una determinada función)
- \* Interrupción de entrada/salida (el dispositivo de E/S pide atención o hace una indicación de que ha iniciado una operación)
- \* Llamada al supervisor por el programa

### **NECESIDAD DE UN SISTEMA DE INTERRUPCIONES**

Surge de la sincronización entre CPU y los dispositivos. Dado que los componentes mecánicos no cumplen con la exactitud de tiempos de ejecución que se da en la CPU, y puede averiguar:

- Si el dispositivo está preparado
- Si una operación E/S se está ejecutando
- Si una operación ha terminado

Cuando se trata de periféricos resulta eficiente la modalidad de interrupción para la atención de estos => atiende al periférico cuando lo solicite y puede utilizar el resto de tiempo disponible en ejecutar otro proceso (p.e. la atención de otro periférico)

### **EJEMPLO:**

La interrupción desde un periférico se inicia, cuando tiene información para entregar, a través del envío de una señal a la CPU para solicitar atención conocida como IRQ (Interrupt Request)

Si la interrupción está enmascarada la CPU hace caso omiso de la solicitud. Si la interrupción no está enmascarada el computador responde al periférico, que recibirá la información, por medio de una señal IACK (Interrupt Acknowledge).

Luego el periférico o manejador de interrupciones envía una dirección de memoria a la CPU. Esta dirección de memoria contiene a su vez la dirección de la rutina que efectivamente administra la recepción de la información.

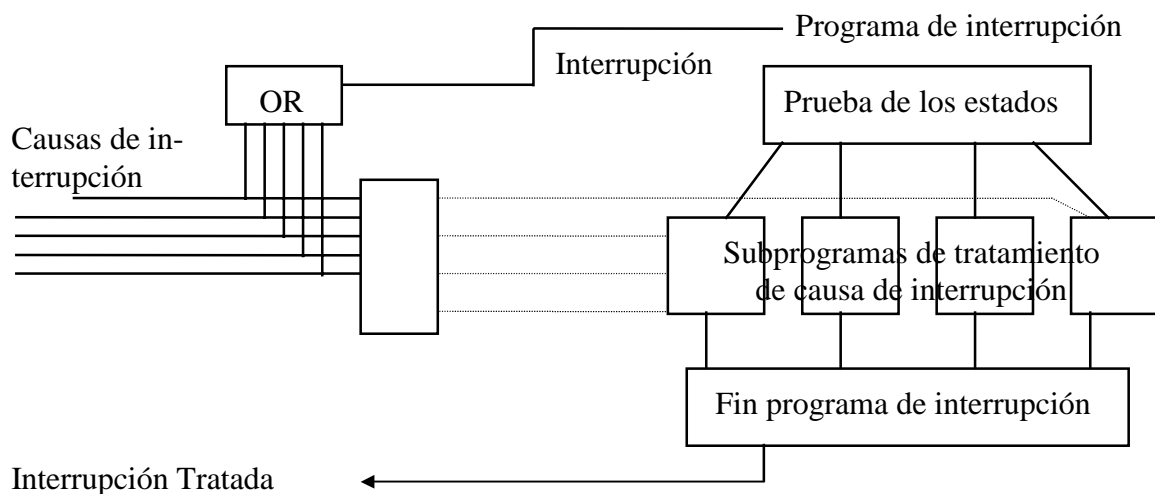
La CPU hace un salto a la posición de memoria indicada en la celda correspondiente a la de la interrupción y la rutina contenida a partir de allí toma el control del sistema. El "PROGRAM COUNTER" (contiene la dirección de memoria de la próxima instrucción a ejecutar) y el registro de status son empujados al stack (pila).

Una vez finalizada la ejecución de la interrupción, devuelve el control a la CPU que puede continuar con el proceso interrumpido o iniciar otro nuevo.

## TIPOS DE INTERRUPCIONES

- (1) Interrupción por errores o por daños en la máquina => fallo en la alimentación eléctrica; error de paridad en memoria.
- (2) Interrupción por causa del programa (o desvío) => instrucción o dirección incorrecta; operaciones imposibles; desbordamiento de capacidad; división por cero, etc; intentos de ejecución de instrucciones o de escritura en memoria no permitidos por el estado de la máquina.
- (3) Interrupción por entrada/salida => generada por el canal para avisar fin de una operación E/S o de una falla ocurrida en el transcurso de E/S.
- (4) Interrupción externa => utilizada para avisar a la máquina acerca de cualquier modificación interesante del medio exterior, especialmente en control de procesos industriales.
- (5) Interrupción de recuento => para contar impulsos procedentes de un reloj.

Determinados procesadores presentan una sola posibilidad de interrupción, en cuyo caso se pasarán todas las causas de interrupción a través de un OR lógico y el programa de interrupción deberá comenzar por comprobar un conjunto de indicadores, para detectar cual puede ser la causa de la misma.



## EJEMPLO: INTERRUPCIONES DEL SISTEMA EN IBM-PC 8086.

Existen dos entradas de interrupciones

NMI (no enmascarables)

INTR (enmascarables)

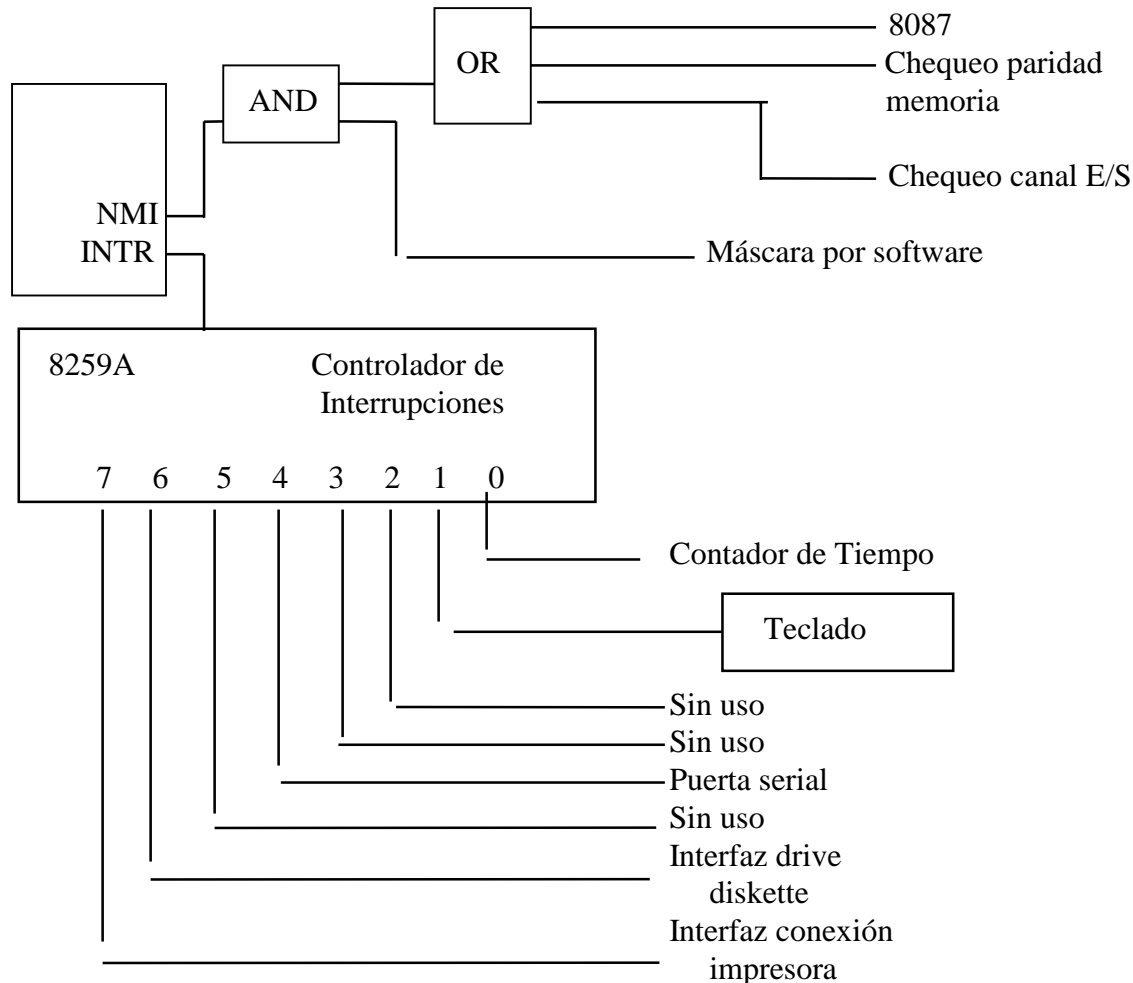
Esta clasificación viene dada por su ejecución.

**Interrupciones no enmascarables** son aquellas que deberán ejecutarse obligatoriamente sin tener que esperar, están ligados a tareas en el sistema que no se pueden retrasar como son las rutinas de bajada del sistema por corte de energía, chequeo de la integridad de memoria, etc.

**Interrupciones enmascarables** se ejecutan a solicitud del periférico correspondiente siempre que en el registro de status (FLAGS REGISTER) del procesador las interrupciones estén habilitadas. En el caso de no estarlo la solicitud será ignorada y la CPU continuará con lo que estaba realizando.

En nuestro ejemplo las fuentes de las interrupciones no enmascarables son 3, y pueden provenir del coprocesador matemático INTEL 8087, falla en la paridad de memoria RAM o falla en algún canal de E/S. Aunque en la práctica se los puede enmascarar por Software => usa una compuerta AND para habilitar la solicitud.

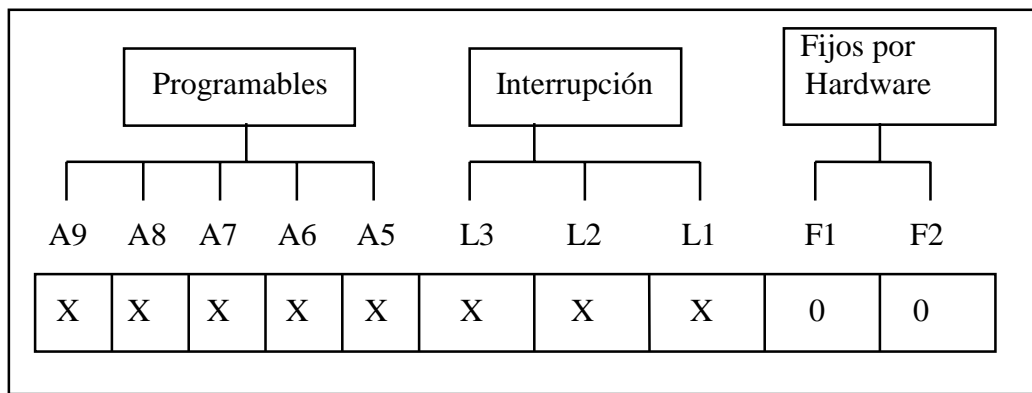
#### MICRO PROCESADOR 8086



Las interrupciones enmascarables son manejadas por el controlador de interrupciones 8259A. Es el encargado de priorizar las solicitudes mediante lógica propia. De esta forma ejecutará primero las de alta prioridad independientemente del orden de llegada, se llega al caso de que una interrupción puede ser interrumpida a su vez por otro de mayor prioridad.

El 8259A es programable y puede administrar hasta 8 interrupciones. Cuando esta pastilla recibe la señal "IACK" del procesador procede a entregar una dirección de 10 BITS de largo que corresponde a la posición en la tabla de interrupciones donde se encuentra la dirección de la rutina que atenderá al periférico.

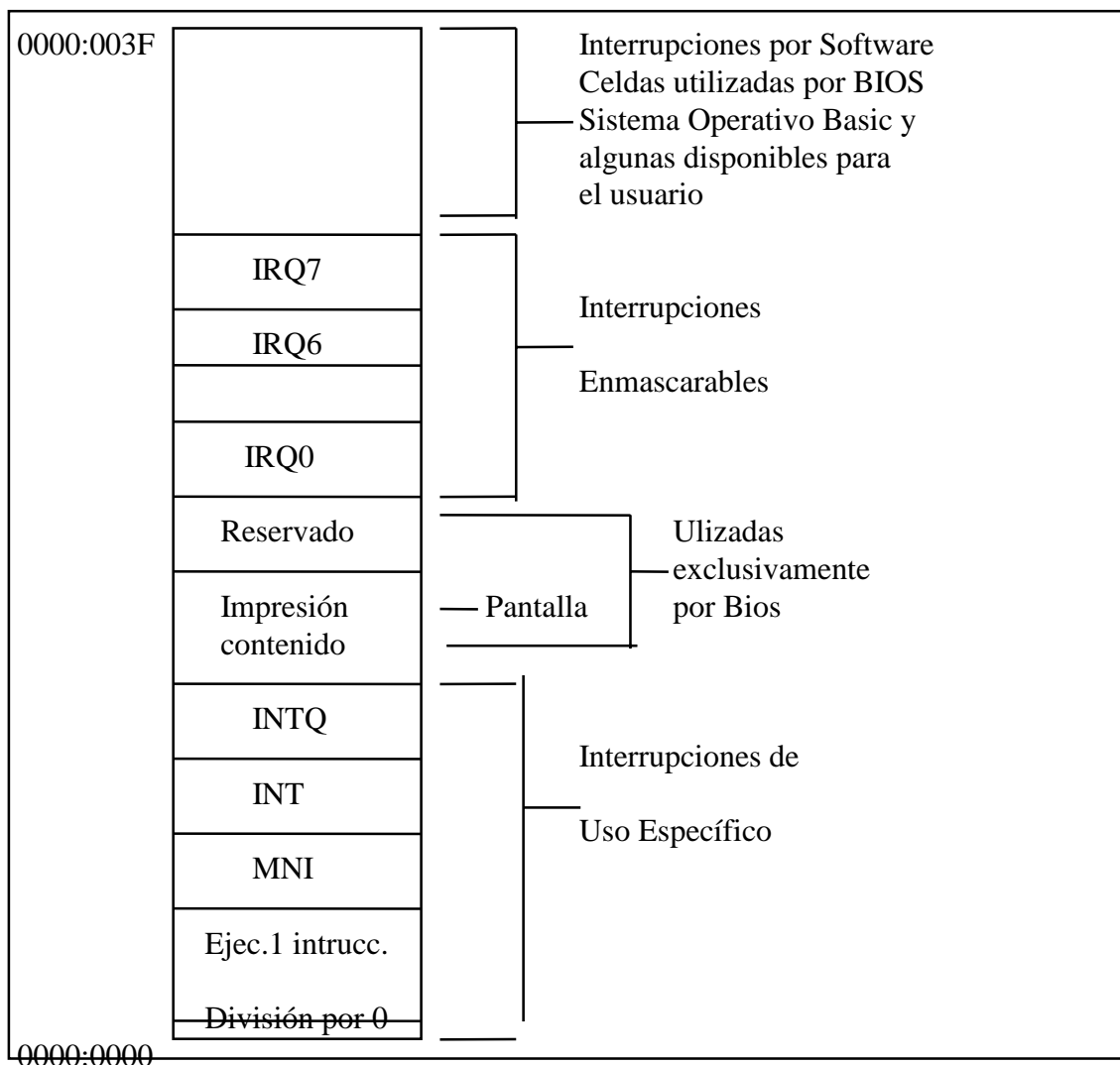
La dirección entregada a la CPU tiene el siguiente formato.

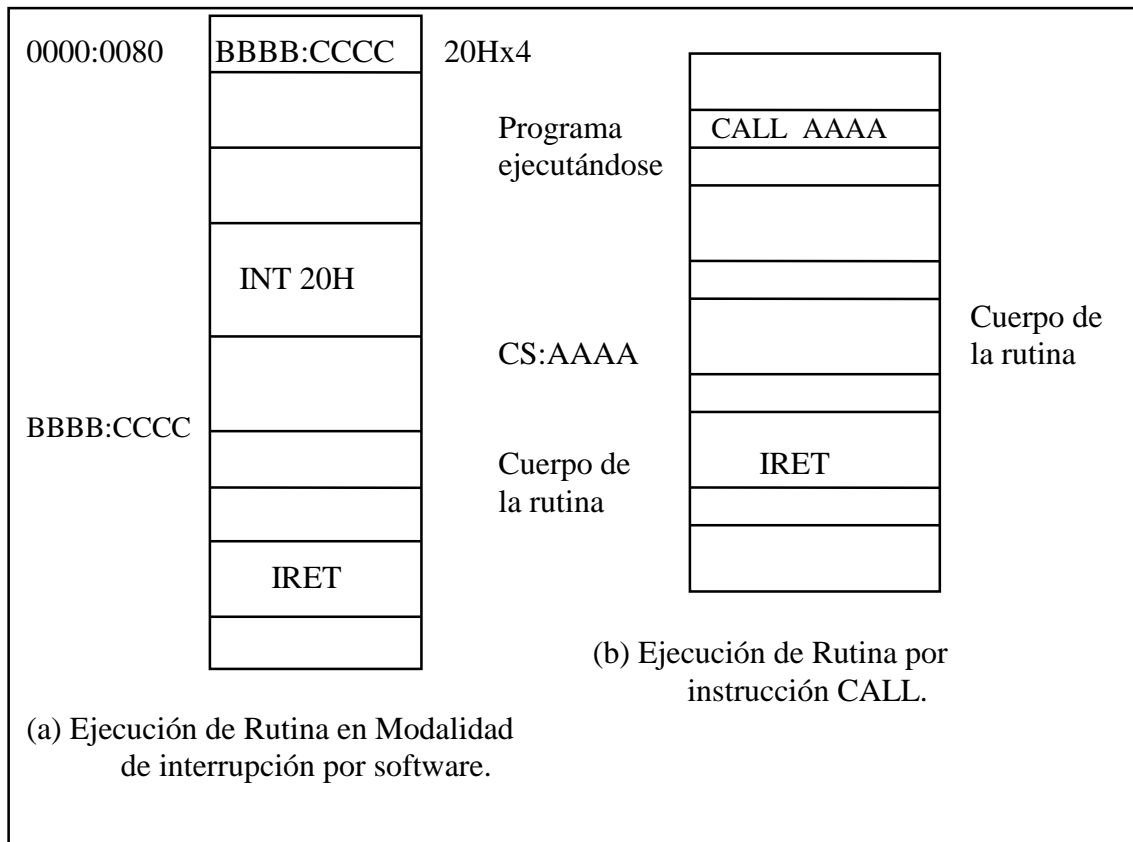


en la tabla correspondiente. El segmento de la dirección por definición es cero. De tal manera que si la CPU se encuentra con la instrucción INT 20H esta va a la dirección 0000:0080 (20x4=804) y desde allí obtiene la dirección de la rutina a ejecutar.

Por otra parte la instrucción "CALL" se caracteriza porque el program counter salta en forma directa a la rutina mencionada.

Es decir si encuentra la instrucción CALL xxxx (xxxx = número hexadecimal de 4 cifras) la próxima instrucción a ejecutar estará contenida en la dirección CS:xxxx (CS: REGISTRO DE SEGMENTO DE CÓDIGO) como CALL es un salto directo, no es posible ejecutar cambios en la posición de memoria de la rutina de llamada, pues la CPU siempre intentará ejecutar desde la dirección original, razón por la cual los sistemas operativos utilizarán poco la modalidad CALL.

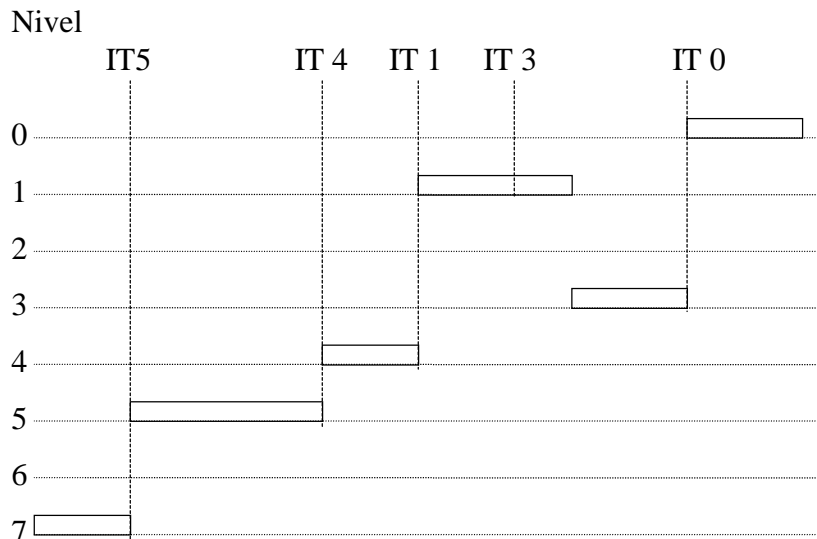




### Efecto de la jerarquización de un sistema de interrupciones

**Jerarquizados** => cada interrupción tiene asignada una prioridad para su ejecución. En este tipo de sistema una rutina de interrupción puede ser interrumpido por una demanda de interrupción clasificada con un nivel de prioridad superior.

El programa interrumpido pasa a un estado de espera. Ejemplo de un sistema de interrupción de 8 niveles. El nivel "0" es el de más alta prioridad y el "7" de más baja. Vemos como los programas se interrumpen por uno de mayor jerarquía.



**Esquema de un sistema jerarquizado de interrupciones prioritarias.** Cada nivel de interrupción tiene asociados cuatro biestables que definen los diferentes estados posibles del nivel.

**ESTADO DESACTIVADO:** El nivel no acepta ninguna demanda de interrupción.

**ESTADO ACTIVADO:** El nivel acepta y memoriza una demanda de interrupción. Un nivel de interrupción puede ser activado o desactivado por programa.

**ESTADO DE ESPERA:** El nivel pasa al estado de espera si recibe una señal de demanda de interrupción.

**ESTADO DE ESPERA INHIBIDO:** el nivel ha resultado inhibido por programa, de tal manera que la interrupción ha podido ser memorizada pero no tomada en cuenta por el ordenador => enmascarada.

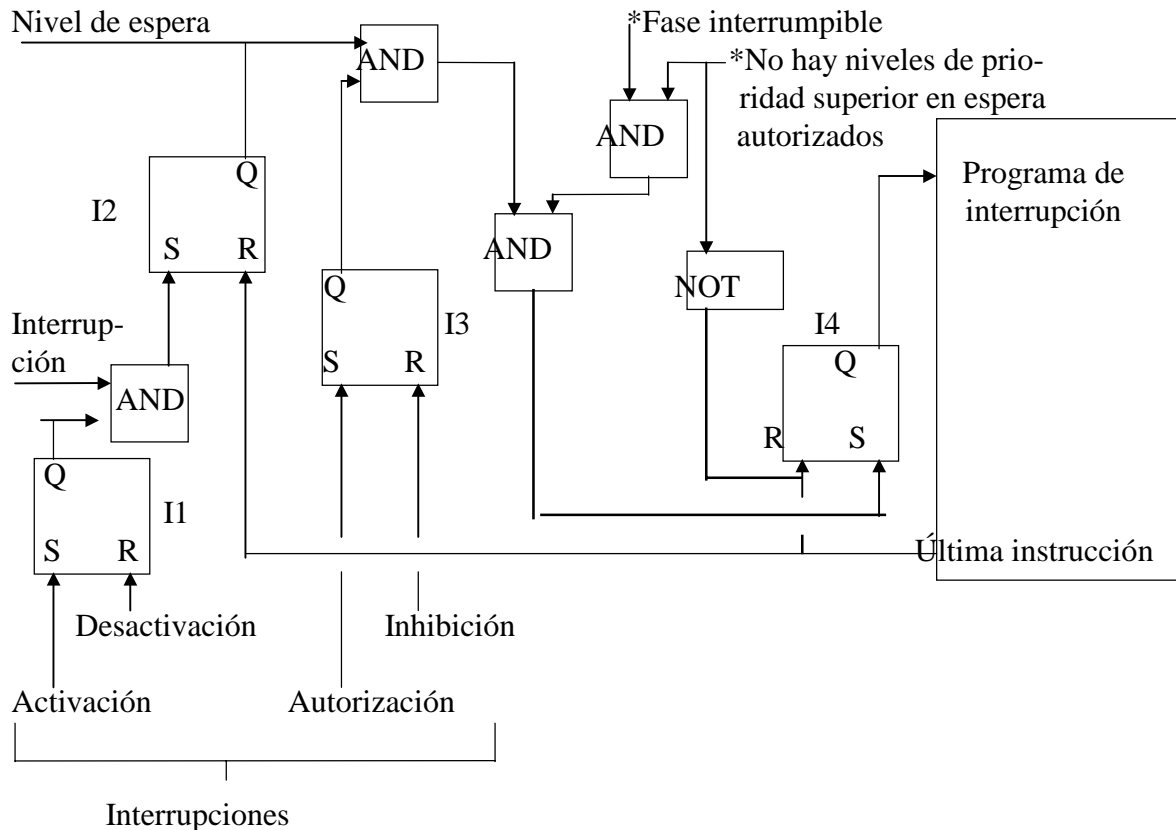
**ESTADO DE ESPERA AUTORIZADO:** La interrupción puede ser tomada en cuenta por el ordenador, si se han satisfecho las siguientes condiciones:

- (1) No existe ningún nivel de prioridad superior en el estado autorizado de espera;
- (2) la unidad central se encuentra en una fase interrumpible (generalmente al final de instrucción).

El nivel entonces pasa al estado activo.

**ESTADO ACTIVO:** Implica que la unidad central tome en cuenta la interrupción y se mantiene durante toda la ejecución del programa de interrupción.

Este esquema asocia a cada señal de interrupción dos señales: una impulsional desde el exterior (pide interrupción) y otra de larga duración que él deja activada hasta tanto no se haya tratado la interrupción es decir hasta que el nivel sea capaz de aceptar una nueva demanda de interrupción.



## EFFECTO DE UNA INTERRUPCIÓN SOBRE LAS INSTRUCCIONES

Existen instrucciones interrumpibles y no interrumpibles. La ejecución del servicio de interrupción espera al final de las segundas.

Una interrupción E/S externa, o error de máquina no grave, no afecta la instrucción, si el error está en la propia instrucción puede ocurrir que la instrucción sea:

SUPRIMIDA (no se puede realizar)

ANULADA (se inicia pero se trunca)

COMPLETADA

TERMINADA ANORMALMENTE

La gestión de errores del usuario deberá considerar el resultado que ocasiona la instrucción errónea.

## INTERRUPCIONES DE ERROR DE PROGRAMA - EMULACIÓN

Cuando se tiene un programa adquirido (código de máquina) que tiene instrucciones que no figuran en nuestro ordenador. Sería necesario producir una rutina del supervisor que las emulase (realice la misma operación). Son interrupciones dentro del programa.

## INTERRUPCIONES PROGRAMADAS (Control del SPD)

Instrucción SVC sirve para introducir interrupciones programadas. Esta permite solicitar al supervisor determinado servicio. Entonces aquellas operaciones que no puedan ser realizadas por el programa deben ser realizadas por el supervisor que toma el control gracias a una SVC del programa. SVC no es una interrupción E/S (estas son producidas por los canales de E/S).

Formato SVC n

(donde n es el operando que sirve de identificador para saber a donde bifurca hacia la rutina de tratamiento que corresponda)

Cuando el supervisor termina sus operaciones devuelve la ejecución al programa a continuación del último elemento de la lista o va a una dirección especificada como argumento.

## SERVICIOS DEL SUPERVISOR:

SVC 0 ejecutar un programa de canal (EXCP)

SVC 1 carga una fase en memoria y la ejecuta (FETCH)

SVC 4 carga una fase en memoria (LOAD)

SVC 6 cancelar tarea (CANCEL)

SVC.7 espera una interrupción (WAIT)

SVC.10 iniciar cronometrador de intervalos (SET TIME)

SVC.14 cancelar programa (EOJ)

SVC.18 crear rutina para interrupciones de reloj (STX IT operando IT)

SVC.19 efectuar regreso al programa después de una interrupción de reloj (EXIT operando IT)

SVC.34 obtener hora del día (GET TIME)