

## MANUAL DE USUARIO

Con este manual, se pretende proporcionar suficiente información, como para usar en su totalidad las posibilidades iniciales de la I/O card que posee, modos de programación y características para desarrollar programas propios y adaptar a la aplicación que necesite en cada caso.

Con esta tarjeta podremos hacer uso de puertos de entrada y salida, generar interrupciones que más tarde serán capturadas y tratadas en el PC y poseer unos timer para muy diversas aplicaciones, en el caso de la aplicación que se entrega con la tarjeta los timer son usados para generar las interrupciones, mediante pulsos, como si de un reloj se tratase. (0-1-0-1....)

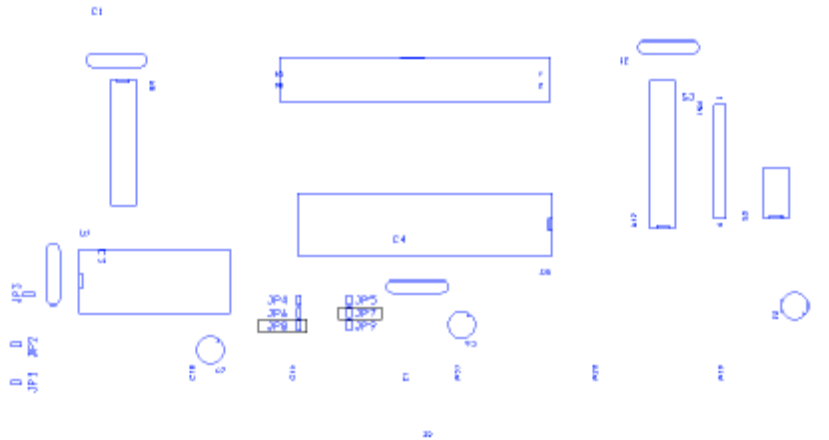
## INSTALACIÓN Y CONFIGURACIÓN

Como primer paso daremos a nuestra tarjeta la dirección base a partir de la cual nos comunicaremos con ella y los diversos dispositivos que esta posee, esto se indica en los micro interruptores incorporados en la placa de modo que:

**11xxxx1111**

Los bits que podremos variar serán los indicados con **x** y si se cambia esta dirección tendremos que tenerlo en cuenta para cambiarlo en las aplicaciones para que la tarjeta funcione, ya que de otro modo mandaríamos la información a otra dirección en la cual el CHIP-SELECT de nuestra tarjeta no reconocerá como suya y no habilitará ninguno de nuestros integrados para que realice las operaciones indicadas, para nuestros programas de chequeo usamos la dirección 320H de modo que en el lugar de las **xxxx** deberá aparecer **0010**, valor junto a los valores fijos de los 2 bit de más peso generaremos el CHIP-SELECT antes mencionado.

## LOS JUMPERS:



Los jumpers JP5, JP7 y JP9, corresponden al reloj externo y JP4, JP6 y JP8 al reloj interno, correspondencias:

**CLK0** - JP8 y JP9.

**CLK1** – JP6 y JP7

**CLK2** – JP4 y JP5

En la descripción anterior se indica a que corresponde cada uno de los jumpers con respecto al timer, para nuestro uno de la placa, según el software proporcionado han de estar conectados los indicados en el dibujo, JP8 y JP7, correspondiendo uno a reloj interno y otro a externo, coincidiendo con la salida del contador 0.

Para que usamos estos jumpers, con estos jumpers lo que conseguimos es que el timer tenga un reloj internos o externo, en nuestro caso usamos las dos opciones, el CLK0 usa reloj interno y CLK1, reloj externo, proporcionado por la salida del anterior y convenientemente conectado en el fajín de pruebas que se suministra con la tarjeta, de modo que aquí tenemos el reloj antes comentados que generará las interrupciones, cuya conexión a esta también se encuentra realizada en el indicado fajín

Una vez estos pasos estén realizados introduciremos la tarjeta en nuestro pc, en el slot isa (con el ordenador apagado), encenderemos el ordenador cargaremos el programa y veremos si está bien pinchada en el ordenador y si hemos seguido correctamente los pasos indicados anteriormente, en caso de que el software nos

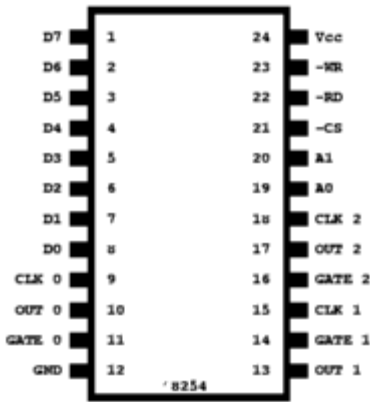
indicara que no es correcta la prueba, apagaremos el ordenador y volveremos a seguir los pasos uno a uno.

## DESCRIPCIÓN Y USO DE LOS DISTINTOS DISPOSITIVOS

### 8254- TIMER

No es más que un contador:

Programación:



D7..D0: BUS de datos bidireccional de 3 estados.

CLK 0: CLOCK 0, entrada de reloj al contador 0.

OUT 0: Salida del contador 0.

GATE0: Puerta de entrada al contador 0.

CLK 1: CLOCK 1, entrada de reloj al contador 1.

OUT 1: Salida del contador 1.

GATE1: Puerta de entrada al contador 1.

CLK 2: CLOCK 2, entrada de reloj al contador 2.

OUT 2: Salida del contador 2.

GATE2: Puerta de entrada al contador 2.

A0..A1: Líneas de dirección para seleccionar uno de los tres contadores o el registro de la palabra de control.

-CS: Habilita la comunicación con la CPU.

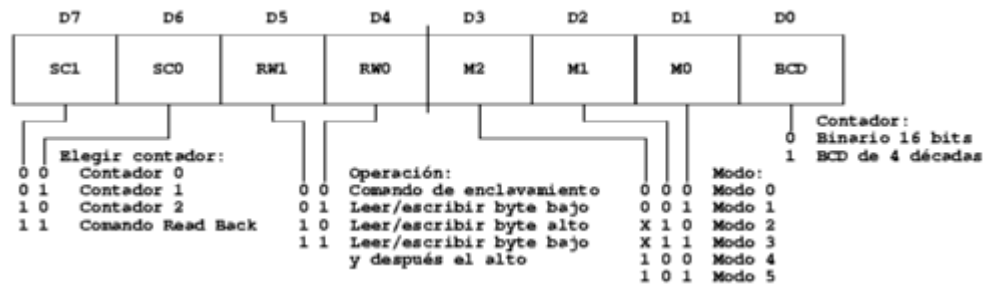
-WR: Permite al 8254 aceptar datos de la CPU.

-RD: Permite al 8254 enviar datos a la CPU.

Es el timer, como podemos observar en la descripción de los pines del integrado, tenemos una entrada de datos, en nuestro caso vienen dadas por el bus de datos del ordenador, unas de control, para seleccionar los contadores procedentes del bus de direcciones, cada uno de los contadores pueden ser programados de forma independiente, tenemos la entrada del reloj, en la placa existen unos jumpers, con los que conseguimos seleccionar para cada uno de los relojes de donde le lleguen los pulsos si directamente del bus o de nuestro conector IDE, con lo cual la señal estará generada desde el exterior.

Tenemos las salidas del contador conectadas a un buffer y de este a nuestro conector IDE que será nuestro conector de salida al y del exterior..

Para programar el 8254 hay que enviar primero una palabra de control y, después, un valor de cuenta inicial. Los contadores se seleccionan con las líneas A0 y A1; el 8254 ocupa normalmente 4 direcciones de E/S consecutivas ligadas a los contadores 0, 1, 2 y al registro de la palabra de control. Para enviar la cuenta inicial se utiliza simplemente el puerto E/S. El formato de la palabra:



Dirección base 323H

Posiciones en el IDE, pines: (después de pasar por el buffer)

14-16-18

### 8255 – PUERTO



D0..D7: Bus de datos bidireccional de 3 estados.

RESET: Esta señal borra el registro de control y todos los puertos (A, B y C) son colocados en modo entrada.

-RD: Utilizada por la CPU para leer información de estado o datos procedentes del 8255.

-WR: Utilizada por la CPU para enviar palabras de control o datos al 8255.

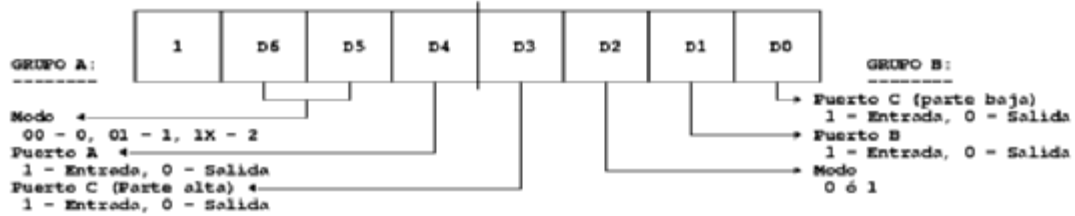
A0..A1: Líneas de dirección: permiten seleccionar uno de los tres puertos o el registro de control.

PA0..PA7: Puerto A: puerto de entrada/salida de 8 bits.

PB0..PB7: Puerto B: puerto de entrada/salida de 8 bits.

PC0..PC7: Puerto C: puerto de entrada/salida de 8 bits.

El 8255 es nuestra puerta de comunicación de información con el exterior tanto entrada como salida poseemos 3 puertos conectados a nuestro IDE, las entrada/salidas de datos directas al bus, entrada de control para seleccionar el puerto deseado en cada momento. La palabra de control es:



Dirección base 320H

Posiciones en el IDE, pines:

#### PUERTO A :

|     |   |     |   |
|-----|---|-----|---|
| pa0 | 8 | pa4 | 1 |
| pa1 | 6 | pa5 | 3 |
| pa2 | 4 | pa6 | 5 |
| pa3 | 2 | pa7 | 7 |

#### PUERTO B :

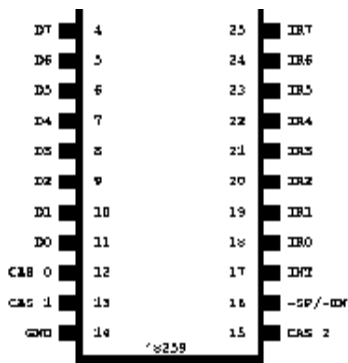
|     |    |     |    |
|-----|----|-----|----|
| pb0 | 9  | pb4 | 17 |
| pb1 | 11 | pb5 | 19 |
| pb2 | 13 | pb6 | 21 |
| pb3 | 15 | pb7 | 23 |

#### PUERTO C:

|     |    |     |    |
|-----|----|-----|----|
| pc0 | 25 | pc4 | 33 |
| pc1 | 27 | pc5 | 35 |
| pc2 | 29 | pc6 | 37 |
| pc3 | 31 | pc7 | 39 |

## 8259- CONTROLADOR DE INTERRUPCIONES

Este circuito integrado está especialmente diseñado para controlar las interrupciones en sistemas basados en el 8080/8085 y en el 8086. Puede controlar hasta 8 interrupciones vectorizadas. Además, a un 8259 se le pueden conectar en cascada un



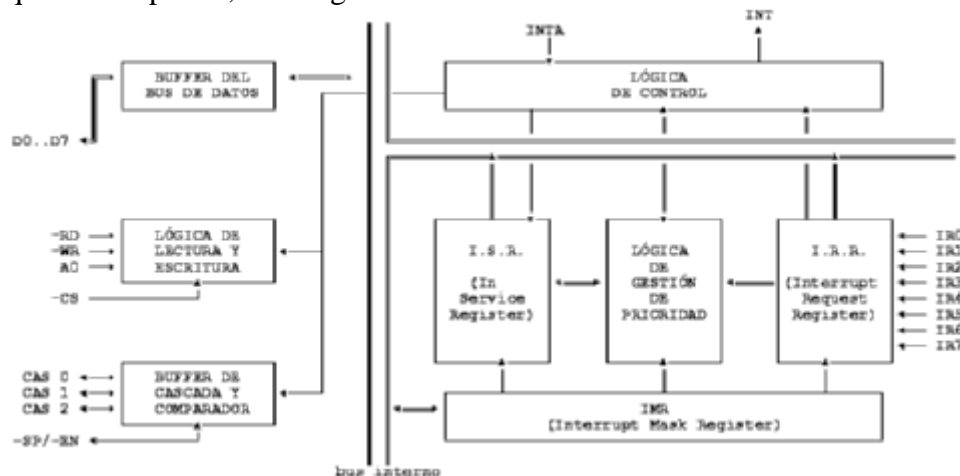
máximo de 8 chips 8259 adicionales, lo que permite gestionar sistemas con hasta 64 interrupciones, como veremos.

El significado e interpretación de las señales se muestra a la derecha:

- CS: Habilita la comunicación con la CPU.
- WR: Permite al 8259 aceptar comandos de la CPU.
- RD: Permite al 8259 dejar la información en el bus de datos.
- D7..D0: Bus de datos bidireccional, por el que se transmite la información de control/estado y el número de vector de interrupción.
- CAS0..CAS2: Líneas de cascada, actúan como salida en el 8259 maestro y como entrada en los 8259 esclavos, en un sistema con varios 8259 interconectados, constituyendo un bus local.
- SP/-EN: Pin de doble función: en el *buffered mode* del 8259 actuará como -EN, para habilitar los buffers del bus; en el modo normal indicará si el 8259 es maestro o esclavo (-SP).
- INT: Conectado a la patilla INT de la CPU para producir la interrupción cuando llegue el momento.
- IR0..IR7: Líneas asíncronas de petición de interrupción. Una petición de interrupción se ejecuta manteniendo IR en alto hasta que se recibe el reconocimiento (modo por flancos) o simplemente poniendo en alto la línea IR (modo por niveles).
- INTA: Línea de reconocimiento de interrupción, por medio de esta línea se fuerza al 8259 a depositar en el bus la información del vector de interrupción. INTA es independiente de -CS.
- A0: En conjunción con -CS, -WR y -RD es empleada para enviar las palabras de comando al 8259 y para solicitar información al mismo. Suele ir conectada a la línea A0 de la CPU.

## DESCRIPCIÓN FUNCIONAL

El diagrama funcional del 8259, con la estructura interna de las diversas partes que lo componen, es el siguiente:



Los principales registros internos del 8259 son el **IRR** (Interrupt Request Register) y el **ISR** (In Service Register). El IRR almacena todas las peticiones de interrupción pendientes; el ISR almacena todas las interrupciones que están siendo atendidas en un momento dado. La **lógica de gestión de prioridad** determina qué interrupción, de las solicitadas en el IRR, debe ser atendida primero: cuando lleguen las

señales INTA dicha interrupción será la primera procesada y su bit correspondiente se activará en el ISR. El **buffer del bus de datos** conecta el 8259 con el bus de datos de la placa principal del ordenador: su diseño en 3 estados permite desconectarlo cuando sea necesario; a través de este bus circulan las palabras de control y la información de estado. La **lógica de lectura y escritura** acepta los comandos que envía la CPU: aquí hay registros para almacenar las palabras de inicialización y operación que envía el procesador; también sirve para transferir el estado del 8259 hacia el bus de datos. El **buffer de cascada/comparador** almacena y compara las identificaciones de todos los 8259 que posea el sistema: el 8259 maestro envía la identificación del 8259 esclavo en las líneas CAS, los 8259 esclavos la leen y el implicado en la operación coloca en el bus de datos la dirección (vector) de la rutina que atenderá la interrupción en los 2 próximos (o el próximo) ciclos INTA.

### **FUNCIONAMIENTO DEL 8259**

El funcionamiento del 8259 varía ligeramente en función del sistema en que esté instalado, según sea este un 8086 o un 8080/8085. Veremos primero el caso del 8086:

- 1) Una o más líneas IR son activadas por los periféricos, lo que pone a 1 el correspondiente bit del IRR.
- 2) El 8259 evalúa la prioridad de estas interrupciones y solicita la interrupción a la CPU (línea INT) si es necesario.
- 3) Cuando la CPU reconoce la interrupción, envía la señal -INTA.  
Nada más recibida la señal -INTA de la CPU, el 8259 activa el bit correspondiente a
- 4) la interrupción de mayor prioridad (la que va a ser procesada) en el ISR y lo borra en el IRR. En este ciclo, el 8259 aún no controla el bus de datos.  
Cuando la CPU envía un segundo ciclo -INTA, el 8259 deposita en el bus de datos un
- 5) valor de 8 bits que indica el número de vector de interrupción del 8086, para que la CPU lo pueda leer.  
En el modo AEOI del 8259, el bit de la interrupción en el ISR es borrado nada más acabar el segundo pulso -INTA; en caso contrario, ese bit permanece activo hasta que
- 6) la CPU envíe el comando EOI al final de la rutina que trata la interrupción (caso más normal).

En el caso de sistemas basados en el 8080/8085, el funcionamiento es idéntico hasta el punto (3), pero a continuación sucede lo siguiente:

- Nada más recibida la señal -INTA de la CPU, el 8259 activa el bit correspondiente a
- 4) la interrupción de mayor prioridad (la que va a ser procesada) en el ISR y lo borra en el IRR. En este ciclo, el 8259 deposita en el bus de datos el valor 11001101b, correspondiente al código de operación de la instrucción CALL del 8080/85.
  - 5) Esta instrucción CALL provoca que la CPU envíe dos pulsos -INTA.  
El 8259 utiliza estos dos pulsos -INTA para depositar en el bus de datos,
  - 6) sucesivamente, la parte baja y alta de la dirección de memoria del ordenador de la rutina de servicio de la interrupción (16 bits).  
Esto completa la instrucción CALL de 3 bytes. En el modo AEOI del 8259, el bit de
  - 7) la interrupción en el ISR es borrado nada más acabar el tercer pulso -INTA; en caso contrario, ese bit permanece activo hasta que la CPU envíe el comando EOI al final de la rutina que trata la interrupción.

Si en el paso (4), con ambos tipos de microprocesador, no está presente la petición de interrupción (por ejemplo, porque ha sido excesivamente corta) el 8259 envía una interrupción de nivel 7 (si hubiera un 8259 conectado en IR7, las líneas CAS

permanecerían inactivas y la dirección de la rutina de servicio de interrupción sería suministrada por el 8259 maestro).

## PROGRAMACIÓN DEL 8259

El 8259 acepta dos tipos de comandos generados por la CPU: los **ICW** (Initialization Command Word) que inicializan el 8259, y los **OCW** (Operation Command Word) que permiten programar la modalidad de funcionamiento. Antes de que los 8259 de un sistema comiencen a trabajar deben recibir una secuencia de ICW que los inicialice. Los ICW y OCW constan de secuencias de 2 a 4 comandos consecutivos que el 8259 espera recibir secuencialmente, unos tras otros, a través del bus de datos, según sea necesario (el propio 8259 se encarga de contarlos midiendo los pulsos de la línea -WR). Los OCW pueden ser enviados en cualquier momento, una vez realizada la inicialización.

La comunicación con el 8259 emplea las líneas -WR y -RW, así como A0. El hecho de que exista una sola línea de direcciones implica que el 8259 sólo ocupa dos direcciones de puerto de E/S en el espacio de entrada y salida del ordenador.

### **ICWS (Initialization Command Words).**

**ICW1:** Cuando un comando es enviado con A0=0 y D4=1, el 8259 lo interpreta como la primera palabra de la inicialización (ICW1) e inicia dicha secuencia de inicialización, lo que implica lo siguiente:

- Se resetea el circuito sensible a los niveles, lo que quiere decir que hasta nueva orden las líneas IR serán sensibles por flancos de transición bajo-alto.
- Se limpia el IMR.
- A la línea IR7 se le asigna un nivel de prioridad 7.
- Se desactiva el *Special Mask Mode*. Se queda listo para devolver IRR en la próxima lectura OCW3.
- Si IC4 (bit D0) es 0, todas las funciones seleccionadas en ICW4 serán puestas a 0 (*non buffered mode*, no AEOI, sistema 8080/85) e ICW4 no será necesaria.

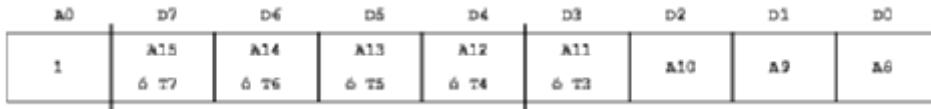


Notas: Si SNGL es 1 significa que el 8259 es único en el sistema y no será enviada ICW3. Si IC4 es 0, tampoco será enviada ICW4. En el 8080/85, las diversas interrupciones generan CALL's a 8 direcciones adyacentes separadas 4



u 8 bytes (según indique ADI): para componer la dirección, el 8259 inserta A0..A4 (o A0..A5) convenientemente, según la interrupción que se trate. En el 8086, A7..A5 y ADI son ignoradas.

**ICW2:** Se envía con A0=1, para diferenciarlo de ICW0 (hacer OUT a la siguiente dirección de puerto).



Notas: En el 8080/85, A15..A8 completan la dirección de la rutina de servicio; en el 8086, T7..T3 determinan los cinco bits más significativos del número de vector de interrupción a invocar (los 3 bajos los suministra el 8259 según la interrupción que se trate).

**ICW3:** Se envía sólo en el caso de que haya más de un 8259 en el sistema (bit SNGL de ICW1 a cero), en caso contrario en su lugar se enviaría ICW4 (si procede).

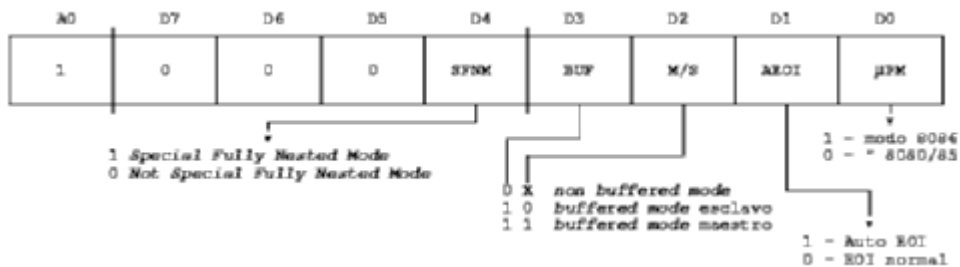
Formato de ICW3 a enviar a un 8259 maestro:



Formato de ICW3 a enviar a un 8259 esclavo para que memorice de qué línea IR del maestro cuelga:



**ICW4:** Se envía sólo si IC4=1 en ICW1, con objeto de colocar el 8259 en un modo de operación distinto del establecido por defecto (que equivale a poner a cero todos los bits de ICW4).



Notas: El *Special Fully Nested Mode*, el *buffered mode* y la modalidad AEOI serán explicadas más tarde. Nótese que con el 8086 es obligatorio enviar ICW4 para seleccionar esta CPU.

### OCWS (Operation Command Words).

Una vez inicializado, el 8259 está listo para procesar las interrupciones que se produzcan. Sin embargo, durante su funcionamiento normal está capacitado para recibir comandos de control por parte de la CPU.

#### OCW1:

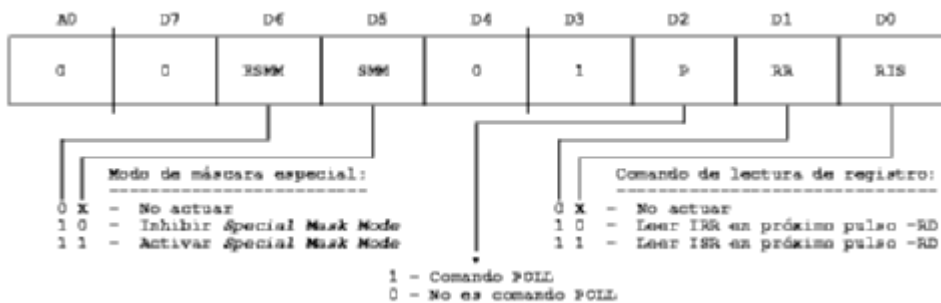


Este comando activa y borra bits en el IMR (Interrupt Mask Register). Los bits M0..M7 de OCW1 se corresponden con sus correspondientes bits del IMR. Un bit a 1 significa interrupción enmascarada (inhibida) y a 0, interrupción habilitada.

#### OCW2:



#### OCW3:



### TRABAJANDO CON EL 8259

En las ICW y, sobre todo, en las OCW, se han introducido un aluvión de elementos nuevos que serán explicados a continuación.

**Fully Nested Mode.** Por defecto, el 8259 opera en esta modalidad (modo de anidamiento completo), a menos que se le programe de otra manera. En este modo las interrupciones quedan ordenadas, por prioridades, de 0 (máxima) a 7 (mínima). Cuando

se produce un reconocimiento de interrupción por parte de la CPU, el 8259 evalúa cuál es la interrupción pendiente de mayor prioridad, coloca su número de vector en el bus y activa su bit correspondiente en el ISR. Este bit permanece activo hasta que el 8259 recibe el comando EOI (situación más normal); sin embargo, en el modo AEOI, ese bit se bajaría inmediatamente después del último -INTA. Mientras el bit del ISR esté activo, todas las interrupciones de igual o menor prioridad que lleguen permanecen inhibidas; sin embargo, las de mayor prioridad podrán interrumpir. En el caso del 8086, cuando comienza el tratamiento de la interrupción, un bit del registro de estado de la CPU mantiene inhibidas todas las interrupciones: lo normal es que el programa de control comience con STI para permitir que el 8086 envíe nuevas señales INTA al 8259, así el 8259 podrá enviar las interrupciones de mayor prioridad que le lleguen. Tras la secuencia de inicialización, las interrupciones quedan ordenadas de mayor (IR0) a menor prioridad (IR7), aunque este orden puede modificarse en la modalidad de prioridad rotatoria o con el comando de asignación de prioridad. Nótese que cuando se utiliza el modo AEOI o el *Special Mask Mode* no se respeta el modo *Fully Nested Mode* (debido a que una interrupción de menor prioridad podría interrumpir a una rutina que gestiona otra de mayor prioridad).

**Special Fully Nested Mode.** Se emplea en sistemas que tienen varios 8259 conectados. Sólo el 8259 maestro es programado en este modo, lo que implica las siguientes diferencias respecto al *Fully Nested Mode* normal:

Cuando se atiende una interrupción de un 8259 esclavo, si viene otra de mayor prioridad de ese mismo 8259 esclavo, se provoca una interrupción al maestro (normalmente, el 8259 esclavo estaría enmascarado mientras se procesa una de sus interrupciones).

Cuando acaba la rutina de servicio de interrupción, hay que enviar un EOI no específico al 8259 esclavo; además hay que leer a continuación su ISR y comprobar si es cero: en ese caso, hay que enviar además otro EOI al 8259 maestro (si no es cero significa que aún hay interrupciones en proceso en el 8259 esclavo).

**Modos de EOI.** El EOI (End Of Interrupt) sirve para bajar el bit del ISR que representa la interrupción que está siendo procesada. El EOI puede producirse automáticamente (AEOI) al final de la última señal INTA que envía la CPU al 8259 para una interrupción dada (tercer ciclo INTA en el 8080/85 y segundo en el 8086); sin embargo, la mayoría de los sistemas requieren una gestión de prioridades en las interrupciones, lo que significa que es más conveniente que EOI lo envíe el propio procesador al 8259, a través de OCW2, cuando acabe la rutina de gestión de interrupción, para evitar que mientras se gestiona esa interrupción se produzcan otras de igual o menor prioridad. En un sistema con varios 8259, el EOI debe ser enviado no sólo al 8259 esclavo implicado sino también al maestro. Hay dos modalidades de EOI: la específica y la no-específica. En el EOI no específico, el 8259 limpia el bit más significativo que esté activo en el ISR, que se supone que es el correspondiente a la última interrupción producida (la de mayor prioridad y que está siendo procesada). Esto es suficiente para un sistema donde se respeta el *Fully Nested Mode*. En el caso en que no fuera así, el 8259 es incapaz de determinar cuál fue el último nivel de interrupción procesado, por lo que la rutina que gestiona la interrupción debe enviar un EOI específico al 8259 indicándole qué bit hay que borrar en el ISR.

**Rotación de prioridades.** Hay sistemas en que varios periféricos tienen el mismo nivel de prioridad, en los que no interesa mantener un orden de prioridades en las líneas IR. En condiciones normales, nada más atender una interrupción de un periférico, podría venir otra que también se atendería, mientras los demás periféricos se cruzarían de brazos. La solución consiste en asignar el menor nivel de prioridad a la

interrupción recién atendida para permitir que las demás pendientes se procesen también. Para ello se envía un EOI que rote las prioridades: si, por ejemplo, se había procesado una IR3, IR3 pasará al menor nivel de prioridad e IR4 al mayor, quedando las prioridades ordenadas (de mayor a menor): IR4, IR5, IR6, IR7, IR0, IR1, IR2, IR3. Existe también una rotación específica de prioridades, a través de OCW2, que puede realizarse en un comando EOI o independientemente del mismo (comando para asignar prioridad).

**Special Mask Mode.** Hay ocasiones en las que mientras se ejecuta una rutina de servicio de interrupción es necesario permitir que se produzcan ciertas interrupciones de menor prioridad en algunos momentos, o prohibirlo en otros, sin ser quizá interesante enviar el EOI antes de tiempo. Esto implica alterar la estructura normal de prioridades. La manera de realizar esto es activando el *Special Mask Mode* a través de OCW3 durante la rutina de servicio de interrupción (es más que conveniente inhibirlo de nuevo al final). Una vez activado este modo, el IMR indica qué interrupciones están permitidas (bit a 0) y cuáles inhibidas (bit a 1). Por ello, suele ser conveniente activar el bit del IMR correspondiente a la IR en servicio (para evitar que se produzca de nuevo cuando aún no ha sido procesada). Al final hay que enviar un EOI específico, ya que este modo de trabajo altera el *Fully Nested Mode* habitual.

**Comando POLL.** En esta modalidad poco habitual, habilitada a través de OCW3, no se emplea la salida INT del 8259 o bien el microprocesador trabaja con las interrupciones inhibidas. El servicio a los periféricos es realizado por software utilizando el comando POLL. Una vez enviado el comando POLL, el 8259 interpreta la próxima lectura que se realice como un reconocimiento de interrupción, actualizando el ISR y consultando el nivel de prioridad. Durante esa lectura, la CPU obtiene en el bus de datos la palabra POLL que indica (en el bit 7) si hay alguna interrupción pendiente y, en ese caso, cuál es la de mayor prioridad (bits 0-2).

**Lectura de información del 8259.** El IMR puede ser leído a través de OCW0; para leer el contenido del IRR y el ISR hay que emplear OCW3. Para estos dos últimos registros hay que enviar una OCW3 que elija el IRR o el ISR; a continuación se puede leer el bus de datos (A0=0) sin necesidad de enviar más OCW3 (el 8259 es capaz de recordar si tiene que leer el IRR o el ISR). Esto último no es así, evidentemente, en el caso de utilizar el comando POLL (tras enviarlo, la próxima lectura se interpreta como un INTA). Tras inicializarse, el 8259 queda preparado por defecto para devolver IRR a la primera lectura.

**Buffered Mode.** Al emplear el 8259 en grandes sistemas, donde se requieren buffers en los buses de datos, si se va a emplear el modo cascada existe el problema de la habilitación de los buffers. Cuando se programa el modo buffer, la patilla -SP/-EN del 8259 actúa automáticamente como señal de habilitación de los buffers cada vez que se deposita algo en el bus de datos. Si se programa de esta manera el 8259 (bit BUF de ICW4) será preciso distinguir por software si se trata de un 8259 maestro o esclavo (bit M/S de ICW4).

#### 12.4.3. - EL 8259 DENTRO DEL ORDENADOR.

Los PC/XT vienen equipados con un 8259 conectado a la dirección base E/S 20h; este controlador de interrupciones es accedido, por tanto, por los puertos 20h (A0=0) y 21h (A0=1). En los AT y máquinas superiores, adicionalmente, existe un segundo 8259 conectado en cascada a la línea IR2 del primero. Este segundo controlador es accedido a través de los puertos 0A0h y 0A1h. La BIOS del ordenador, al arrancar la máquina, coloca la base de interrupciones del primer controlador en 8, lo que significa que las respectivas IR0..IR7 están ligadas a los vectores de interrupción 8..15; el segundo 8259 de los AT genera las interrupciones comprendidas entre 70h y 77h. La

asignación de líneas IR para los diversos periféricos del ordenador es la siguiente (por orden de prioridad):

|        |                                  |                             |
|--------|----------------------------------|-----------------------------|
| IRQ 0  | Temporizador                     | (INT 08h)                   |
| IRQ 1  | Teclado                          | (INT 09h)                   |
| IRQ 2  | En los PC/XT: canal E/S          | (INT 0Ah)                   |
| IRQ 8  | Reloj de tiempo real             | (INT 70h) -+                |
| IRQ 9  | Simulación de IRQ2               | (INT 71h)                   |
| IRQ 10 | Reservado                        | (INT 72h)                   |
| IRQ 11 | Reservado                        | (INT 73h)                   |
| IRQ 12 | Reservado                        | (INT 74h)  > Sólo AT y PS/2 |
| IRQ 13 | Coprocesador aritmético          | (INT 75h)                   |
| IRQ 14 | Controlador de disco duro        | (INT 76h)                   |
| IRQ 15 | Reservado                        | (INT 77h) -+                |
| IRQ 3  | COM2                             | (INT 0Bh)                   |
| IRQ 4  | COM1                             | (INT 0Ch)                   |
| IRQ 5  | Disco duro PC/XT (LPT2 en el AT) | (INT 0Dh)                   |
| IRQ 6  | Controlador de disquetes         | (INT 0Eh)                   |
| IRQ 7  | LPT1                             | (INT 0Fh)                   |

En los AT, la línea IR2 del 8259 maestro es empleada para colgar de ella el segundo 8259 esclavo. Como la línea IR2 está en el slot de expansión de 8 bits, por razones de compatibilidad los AT tienen conectado en su lugar la IR9 que simula la IR2 original. Cuando se produce una IR9 debido a un periférico de XT que pretendía generar una IR2, el AT ejecuta una rutina de servicio en INT 71h que salta simplemente a la INT 0Ah (tras enviar un EOI al 8259 esclavo).

La colocación de IRQ0-IRQ7 en el rango INT 8-INT 15 fue bastante torpe por parte de IBM, al saltarse la especificación de Intel que reserva las primeras 32 interrupciones para el procesador. En modo protegido, algunas de esas excepciones es estrictamente necesario controlarlas. Por ello, los sistemas operativos que trabajan en modo extendido y ciertos extensores del DOS (como las versiones 3.x de WINDOWS) se ven obligados a *mover* de sitio estas interrupciones. En concreto, WINDOWS 3.x las coloca en INT 50h-INT 57h (por software, las máquinas virtuales 8086 emulan las correspondientes INT 8-INT 15). Además, en el modo protegido del 286/386 (o el virtual-86 del 386) la tradicional tabla de vectores de interrupción es sustituida por otra de descriptores, aunque el funcionamiento global es similar.

La interrupción no enmascarable del 80x86 no está controlada por el 8259: es generada por la circuitería que controla la memoria si se detecta un error de paridad. La interrupción no enmascarable puede ser enmascarada en los ordenadores compatibles gracias a la circuitería de apoyo al procesador, aunque no es frecuente; en los AT el bit 7 del puerto 70h controla su habilitación (si es cero, la NMI está habilitada) sin embargo también se podría inhibir el control de paridad directamente (activando los bits 2 y 3 de la dirección E/S 61h, respetando el resto de los bits de ese puerto por medio de una lectura previa). En los PC/XT, es el puerto 0A0h el que controla la habilitación de la NMI, también con el bit 7 (con la diferencia de que debe estar a cero para inhibirla).

CON ESTA INFORMACIÓN Y LOS PROGRAMAS PROPORCIONADOS EN EL DISQUETE JUNTO A LA TARJETA PODREMOS USAR LA TOTALIDAD DE LAS OPCIONES QUE LA TARJETA NOS OFRECE, AHORA TAN SOLO DEBEMOS EJECUTAR LOS DIFERENTES PROGRAMAS Y PROBAR.

