

Buses del sistema

Rául Alcañiz de la Fuente
Periféricos e Interfaces
ULPGC Curso 2008/2009

Índice de contenidos

Introducción	3
1.- Principales buses en un sistema moderno	4
1.1.- Bus del procesador – FSB	5
Socket 7	5
Socket 370	6
Socket 478	7
1.2.- El Bus de memoria	7
1.3.- La necesidad de los slots de expansión	7
2.- Tipos de buses de Entrada/Salida	8
2.1.- Bus ISA	8
2.1.1.- El bus ISA de 8 Bits	9
2.1.2.- El bus ISA de 16 Bits	9
2.2.- El bus MCA	10
2.3.- El bus EISA	10
2.3.- Buses locales	10
2.3.1.- VL-Bus	10
2.3.2.- El bus PCI	11
2.3.3.- El bus PCI-Express	12
2.3.4.- Bus AGP	14
3.- Recursos del sistema	16
3.1.- Interrupciones	16
3.1.1.- Interrupciones del bus ISA de 8 bits	18
3.1.2.- Interrupciones del bus ISA de 16 bits, EISA y MCA	18
3.1.3.- Interrupciones PCI	20
3.1.4.- Advanced Programmable Interrupt Controller (APIC)	21
3.2.- Canales DMA	21
3.2.1.- Canales DMA del bus ISA de 8 bits	21
3.2.2.- Canales DMA del bus ISA de 16 bits	22
3.3.- Direcciones de los puertos de E/S	22
4.- Resolución de conflictos de recursos	26
4.1.- Resolviendo conflictos manualmente	27
4.2.- Problemas de menos: Tarjetas especiales	28
4.2.1.- Tarjetas de sonido	28
4.2.2.- Tarjetas SCSI	29
4.2.3.- Tarjetas de red	29
4.2.4.- Adaptadores COM	29
4.2.5.- USB	29
4.3.- Sistemas Plug & Play	30
4.3.1.- Componentes hardware	30
4.3.2.- Componentes de la BIOS	30
4.3.3.- Componentes del SO	30

Introducción

El corazón de cualquier placa madre son los buses, cuya función es llevar señales entre los componentes del PC.

Un bus es una vía por la que circula información dentro del PC y permite también comunicar dos o más componentes del ordenador.

Los buses dentro del PC poseen un orden jerárquico:

El bus más lento es conectado al inmediatamente más rápido que esté por encima de él. Los ordenadores actuales poseen como mínimo 3 buses, a los cuales se conectan los diferentes dispositivos del sistema.

En este tema también nos pararemos a ver los recursos del sistema del PC, así como los problemas actuales y pasados que podemos tener con ellos.

1.- Principales buses en un sistema moderno

- Bus del procesador (FSB – Front Side Bus)

El FSB es el bus más rápido del sistema ya que se usa para transferir datos entre el CPU y el bus del sistema principal, o entre la CPU y la memoria caché externa, de ahí que tenga que tener una alta velocidad.

- Bus ISA

Es un bus de 8/16 bits y 8 Mhz actualmente en desuso. Es el antecesor del PCI y eran usados para conectar al PC periféricos de baja velocidad como modems, tarjetas de sonido...

- Bus PCI

Bus estándar para conectar periféricos de alta velocidad (tarjetas SCSI, tarjetas de red, gráficas...) directamente a la placa base. A diferencia del ISA estos permiten una configuración dinámica del dispositivo. Se pueden identificar fácilmente en la placa como una ranura blanca (32 bits) y suele haber entre 4 y 6 en la placa.

Del bus PCI se han derivado otros buses como el PCI-X (más ancho de banda aunque con el inconveniente que al instalar más de un dispositivo se perdía velocidad) o el PCI-Express (Más lento en un principio que el PCI-X pero sin el inconveniente del otro).

- Bus AGP

Puerto desarrollado en el 96 como solución a los cuellos de botella que se producían en las tarjetas gráficas que se conectaban al PCI. Es un bus de 32 bits como el PCI pero con mejoras como, por ejemplo, más canales de acceso a RAM. Debido a la arquitectura del bus solo nos podemos encontrar con una de estas ranuras en la placa base y es de uso exclusivo para las tarjetas gráficas.

Algunas Placas tienen unos conectores especiales (Audio Modem Riser - AMR - o Communication and Networking Riser – CNR) dedicados para opciones de comunicación y de Red. En la placa hay también buses que no son tan apreciables como los que hemos estado nombrando, es decir, que no son tan visibles o palpables físicamente como como las ranuras del PCI o del AGP. Estamos hablando del buses como el del Hub Interface o el LPC, para por ejemplo, conectar la bios al CPU.

1.1 .- Bus del procesador (Front Side Bus – FSB)

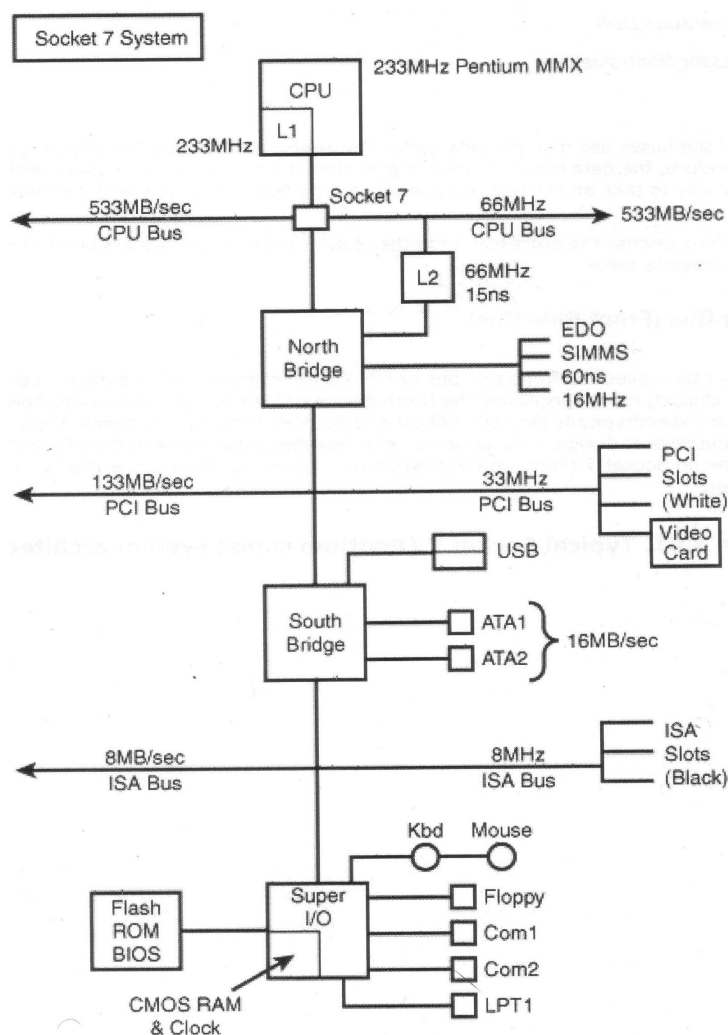
El bus del procesador es el medio de comunicación entre la CPU y el chipset de la placa (mediante el puente norte o el Memory Controller Hub). En los sistemas modernos trabaja entre 66 y 800 Mhz dependiendo del diseño del chipset y de la placa. Veamos algunas arquitecturas de buses del procesador.

Socket 7 (pentium)

Esta arquitectura tiene una caché externa a la CPU (L2), montada en la placa y conectada al bus del procesador que trabaja a la velocidad de la placa (66 Mhz). Por este motivo, mientras se conseguía aumentar más y más la velocidad de los procesadores, pues la L2 seguía trabajando a la velocidad relativamente baja de la placa.

El problema se resolvió en los procesadores P6 (pentium II, celeron, pentium pro...), en los cuales se quita la L2 de la placa y se introduce y se conecta directamente a la CPU, obteniendo así la misma velocidad que el resto de la CPU.

Como curiosidad decir que el nombre de FSB del bus del procesador viene de que el bus de la L2 se le llamó Back-Side bus, así que la industria empezó a llamar al bus de la CPU FSB.

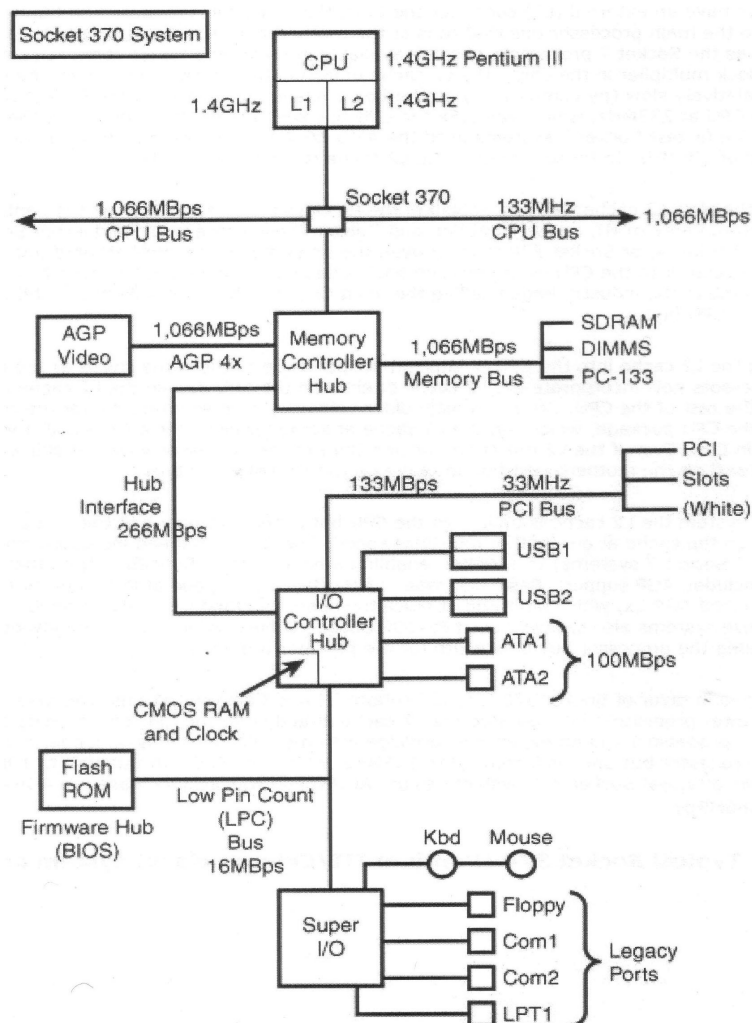


Socket 370 (pentium III/Celeron)

Con la medida tomada, nombrada arriba, y aumentando la velocidad del bus de la CPU a 133 Mhz (1066 Mbps de ancho de banda), nace el Socket 370. Como vemos en la imagen ya se incorpora la L2 dentro de la caja de la CPU y también se cambia el diseño del puente norte-sur por la llamada Hub Architecture. Esta arquitectura cambia la conexión que había entre el puente norte y sur por el 'Hub Interface', que trabaja a 266 Mbps (doble de rendimiento que antes). De esta forma dejamos libre el bus del dispositivo PCI, permitiendo a los dispositivos PCI usar todo el ancho de banda sin tener que compartirlo con el puente sur, como pasaba antes.

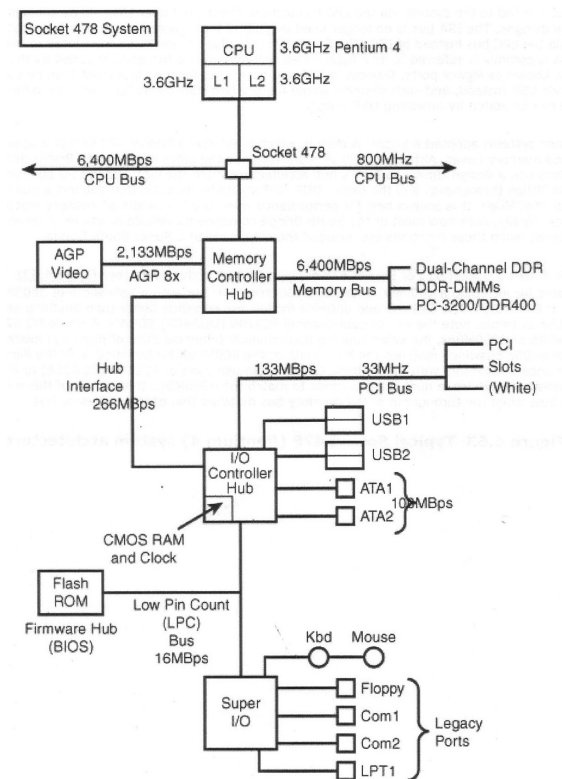
También vemos como el chip 'Flash Rom Bios' está conectado al sistema por el LPC en lugar de por el chip Super I/O como el antiguo diseño Socket 7. El bus ISA también queda obsoleto en esta arquitectura, ya que en su lugar se utiliza para conectar el Super I/O al 'I/O Controller Hub' el LPC.

Como detalle podríamos decir que el chip Super I/O lo podríamos suprimir también del sistemas: Los llamados Legacy-free systems (el nombre se debe a que a los puertos conectados al chip se les llama legacy ports). En lugar del Super I/O se podrían conectar al sistema los legacy ports mediante vía USB, cargándonos así el chip.



Socket 478 (primeros Pentium 4)

Es un arquitectura idéntica a la anterior salvo en las velocidades. Se ha conseguido aumentar la velocidad del bus de la CPU (Pudiendo llegar ahora a 800 Mhz con un ancho de banda de 6400 Mbps) y se introducen nuevos tipos de RAM (Ram DDR, PC-3200...).



1.2.- El bus de memoria

El bus de memoria se usa para transferir información entre la CPU y la RAM. Normalmente está conectado al puente Norte o al Memory Controller Hub. La velocidad depende del tipo de memoria del chipset. El caso más óptimo sería que el bus de memoria trabajase a la misma velocidad que el FSB, lo cual ya se contempla en los sistemas actuales. En estas características no hace falta tener una caché en la placa, de ahí que cuando se quitó la L2 de la placa de las antiguas arquitecturas, ningún fabricante pensó en poner una nueva caché L3.

1.3.- La necesidad de los slots de expansión

Los buses de entrada/salida (o slots de expansión) permiten conectar a la CPU con distintos periféricos. Estos slots nos proporcionan la posibilidad de añadir al PC dispositivos para ampliar sus capacidades, dispositivos como por ejemplo las tarjetas de sonido, tarjetas de video...

Como nota destacar que en los actuales PCs muchos dispositivos básicos como pueden ser tarjetas de sonido, interfaces IDE, puertos USB, controladores de ratón y teclado... vienen integrados en la placa base, aunque se siguen usando los slots para añadir periféricos más potentes.

2.- Tipos de buses de entrada/salida

Como nombramos antes, los buses de E/S son slots de expansión necesarios en el PC y han estado presentes desde el primer PC. Unos dispositivos de E/S rápidos son imprescindibles para un mejor funcionamiento del PC. La necesidad de tener buses de E/S para mejorar el rendimiento del ordenador envuelve tres áreas principales:

- CPUs más rápidas
- Incremento de la demanda del software
- Mayores requerimientos multimedia

Una de las principales razones por la que se tardaba en mejorar las estructuras de los buses de E/S era por la compatibilidad. No servía de nada mejorar estos sistemas si el nuevo no iba a ser compatible con los anteriores y tenías que tirar todo lo que tenías a la basura. Uno de los sucesos que se puede marcar en el mundo de los ordenadores es la estandarización de los Pcs: Si un nuevo bus del sistema de alto rendimiento era introducido en el sistema, debía ser compatible con lo que ya había en él, consiguiendo así que las tarjetas viejas de E/S no se quedasen obsoletas.

En esta parte veremos diferentes tipos de buses de E/S que ha habido en la historia. Por ahora decir que las principales diferencias entre ellos consisten en la cantidad de datos que pueden transferir a la vez y la velocidad a la que pueden hacerlo.

2.1.- Bus ISA (Industry Standard Architecture)

Fue introducido en el 81 como un bus de 8 bits, el cual se amplió en el 84 a 16 bits. El ISA fue la principal arquitectura usada en la mayoría de los Pcs hasta finales de los 90 y fue diseñado para conectar tarjetas de ampliación a la placa base. El motivo principal por el que el ISA aguantase tanto en el mercado era por el simple hecho que aportaba lo necesario para mantenerse: fiabilidad, asequibilidad y compatibilidad.

NOTA.

El bus ISA actualmente ha desaparecido de los sistemas de escritorio de hoy en día (windows, linux, ...) y son pocas las compañías que los fabrican o los venden. Para lo que más se usa ahora mismo es para el diseño de ordenadores industriales, pero se espera que pronto desaparezca también.

Como hemos nombrado, han existido dos versiones claramente diferenciadas del bus ISA: La de 8 y la de 16 bits (basadas en la cantidad de datos que podían transferir a la vez). En un principio el ISA de 8 bits trabajaba a 4,77 Mhz, mientras que el de 16 bits lo hacía a 8 Mhz, aunque después, por cuestiones de compatibilidad en ambas arquitecturas se estandarizó la velocidad máxima a 8.33 Mhz.

2.1.1- El bus ISA de 8 bits

Arquitectura que fue usada en los primeros ordenadores IBM originales y se mantuvo durante muchos años en los sistemas. Sin embargo el bus ISA de 8 bits ya no se incluye en los ordenadores actuales, aunque aún los podemos ver si abrimos un 286 o un 386.

Físicamente el bus ISA de 8 bits es un conector para una tarjeta (un slot en la placa base) con 62 patillas de contacto.

Electrónicamente nos encontramos con 8 líneas de datos y 20 de direcciones, pudiendo manejar así hasta 1 Mb de memoria.

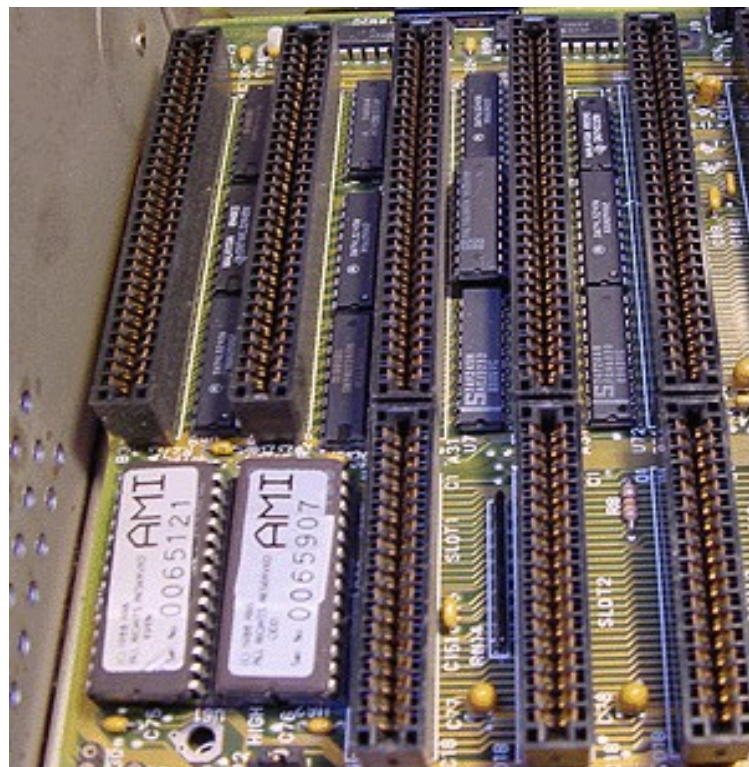
2.1.2- El bus isa de 16 bits

IBM mete un bombazo en el mundo de los ordenadores, en el 84, al introducir el PC-AT con procesadores 286. Estos procesadores tenían un bus de datos de 16 bits, o sea que la comunicación entre el procesador y la placa o la memoria era de 16 bits en vez de 8 como antes. Así que podremos transferir la misma cantidad de datos en la mitad de tiempo.

Lo que más podemos destacar es que esta nueva arquitectura del ISA de 16 bits es compatible con su antecesora, por lo que las tarjetas de 8 bits las podremos conectar a estos nuevos slots de 16 bits, aunque evidentemente, irán más lentas que una de 16 bits.

Para poder conseguir un slot en el que pudiéramos conectar una tarjeta u otra, lo que se hizo fue que a la arquitectura del ISA de 8 bits se le añadió un conector adicional y opcional. Físicamente este conector constaba de 36 patillas de contactos, que si las sumamos a las 62 del ISA original, tendremos un conector de 98 contactos.

Slot de conexión ISA de 8 y 16 bits



2.2.- El bus MCA (Micro Channel Architecture)

Con la llegada de los procesadores con bus de datos de 32 bits, el bus ISA de 16 bits se empezaba a quedar pequeño. IBM aprovecha el peso que tiene en el mercado para lanzar un bus con derecho de propietario: quien quisiera incluir este bus en sus equipos debía pagar un canon a IBM por la utilización de éste. Estamos hablando del MCA, que en sus comienzos aparece con versiones de 16 y 32 bits. Es muy superior al ISA en todos los aspectos, e irónicamente esta fue su tumba. En 1987 el mundo de los Pcs no estaba preparado para tan desarrollada tecnología. Un sistema Plug & Play y los aspectos tecnológicos del bus lo hacían bastante costoso, y si sumamos el tributo que había que pagar a IBM para utilizar el MCA nos salíamos ya de los presupuestos. Así que los fabricantes buscaron una salida: El bus EISA.

2.3.- El bus EISA (Extended Industry Standard Architecture)

Como nombramos antes, este bus se diseñó para evitar tener que utilizar el caro MCA. En el 88 Compaq lidera el desarrollo de este bus de 32 bits compatible con los anteriores y de libre utilización. Físicamente para mantener la compatibilidad lo que se hace es añadir al slot de conexión del ISA anterior 90 patillas sin incrementar el tamaño que ya tenía.

2.4.- Buses locales

El bus ISA, MCA y EISA tienen una cosa en común: son todos relativamente lentos. A continuación vamos a explicar el concepto de bus local usado en los tipos de buses que veremos a partir de ahora.

En los primeros Pcs, los dispositivos de E/S se acoplaban directamente sobre el bus del sistema, que era el único bus del PC. El problema de esto era que de esta manera se forzaba a la CPU a transferir datos a la misma velocidad que los dispositivos de E/S. En aquella época, al trabajar las CPUs a velocidades bastantes, no había mayor problema.

Hoy día, forzar a la CPU a trabajar en sincronía con la E/S es impensable. Lo que se hizo para quitar este problema fue crear el concepto de bus local: Se separaron los dispositivos de E/S del bus del sistema. Se crean buses dedicados a la E/S (buses locales) y se conectan estos buses al bus del sistema consiguiendo así no limitar la velocidad de la CPU.

2.4.1.- Bus local VL

Aunque con el bus EISA se consiguió una transferencia de 32 bits, existía el problema de la diferencia de la velocidad del reloj del bus y el del micro, creándose así cuellos de botella en esa parte (sobre todo en aplicaciones gráficas de video).

Nec lidera un grupo industrial llamado VESA, en el que crea, en el 92, el VL-Bus; un gran paso para eliminar estos cuellos. Funcionaba a 32 bits, aunque podía realizar operaciones de 16 bits. Para mantener la compatibilidad con las tarjetas ISA y EISA, lo que se hace es meter un tercer conector de borde alineado con dos del EISA. Al contrario que el EISA, MCA y PCI, el VL no busca sustituir al ISA, sino complementarlo.

Solamente los slots destinados para tarjetas que usan el bus VL están conectados a la CPU mediante un bus VL, de tal manera que las otras ranuras permanecerán sin ser molestadas y las tarjetas ISA pueden hacer su servicio sin inconvenientes.

La especificación VL-Bus no establece límites en la velocidad del reloj, sin embargo, contra más cantidad de conectores haya, supondrá una mayor capacitancia, lo que hará que disminuya la velocidad a la vez que aumenta la frecuencia. Por tanto se recomendaba a los diseñadores que para sistemas que operasen a más de 33 Mhz no se conectasen más de 3 dispositivos al bus local; a 40 Mhz solo se recomendaban 2 dispositivos; y para sistemas de 50 Mhz solo 1. Es una gran desventaja para el bus VL, así que no es de extrañar que cuando llegó el PCI se lo llevó por delante.

2.4.2.- El bus PCI (Peripheral Component Interconnect)

A principios del 92 Intel lidera la creación de otro grupo de industrias parecido al VESA, el PCI Special Interest Group, creado por la necesidad de vencer las necesidades del bus ISA y EISA y no tener las complicaciones del VL.

En junio del 92 se lanzan las primeras especificaciones del bus PCI y desde entonces ha experimentado muchos cambios, como podemos ver en la siguiente tabla:

PCI Specifications		
PCI Specification	Released	Major Change
PCI 1.0	June 1992	Original 32/64-bit specification
PCI 2.0	April 1993	Defined connectors and expansion boards
PCI 2.1	June 1995	66MHz operation, transaction ordering, latency changes
PCI 2.2	Jan. 1999	Power management, mechanical clarifications
PCI-X 1.0	Sept. 1999	133MHz operation, addendum to 2.2
Mini-PCI	Nov. 1999	Small form factor boards, addendum to 2.2
PCI 2.3	March 2002	3.3V signaling, low-profile add-in cards
PCI-X 2.0	July 2002	266MHz and 533MHz operation, supports subdivision of 64-bit data bus into 32-bit or 16-bit segments for use by multiple devices, 3.3V/1.5V signaling
PCI-Express 1.0	July 2002	2.5GBps per lane per direction, using 0.8V signaling, resulting in 250MBps per lane; designed to eventually replace PCI 2.x in PC systems

El PCI, un bus paralelo, volvió a diseñar el bus tradicional de los Pcs, insertando otro bus entre la CPU y el bus de E/S que ya teníamos. El PCI es independiente de la CPU, por lo que podremos instalarlo en sistemas con cualquier procesador (Intel, Alpha, Macintosh...). Al tener este independentismo con la CPU, va a permitir instalar CPUs más rápidas en el PC sin preocuparse de si las tarjetas que tenemos podrían o no soportar la frecuencia del reloj de la CPU (uno de los principales problemas del VL).

Al comienzo de todo el PCI era un bus de 32 bits y 33 Mhz, con un ancho de banda de 133 Mbps ($33,3 \times 4 = 133$ Mbps.). La siguiente tabla muestra la evolución del bus PCI en cuanto a velocidad.

PCI Bus Types				
PCI Bus Type	Bus Width (Bits)	Bus Speed (MHz)	Data Cycles per Clock	Bandwidth (MBps)
PCI	32	33	1	133
PCI 66MHz	32	66	1	266
PCI 64-bit	64	33	1	266
PCI 66MHz/64-bit	64	66	1	533
PCI-X 64	64 ^[*]	66	1	533

PCI-X 133	64 ^[*]	133	1	1066
PCI-X 266	64 ^[*]	133	2	2132
PCI-X 533	64 ^[*]	133	4	4266
PCI-Express ^[**]	1	2500	0.8	250
PCI-Express ^[**]	16	2500	0.8	4000
PCI-Express ^[**]	32	2500	0.8	8000

Como mejora añadida tenemos que el bus PCI puede funcionar simultáneamente con el bus del procesador, no lo suplanta. Así que esto brindó la posibilidad de tener diálogos concurrentes con el bus local: El micro podrá estar trayendo datos memoria principal a la vez que el PCI está transfiriendo información entre otras partes del sistema.

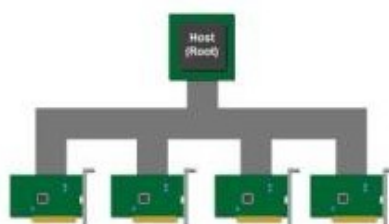
Físicamente nos encontramos con 3 especificaciones distintas del PCI, cada una diseñada para un sistema concreto con distintos requerimientos de energía, cada una de ellas con versiones de 32 y 64 bits. Tenemos las de 5 V para Pcs de sobremesa, la de 3,3 V para portátiles y la universal para la mayoría de los sistemas, que combina las especificaciones de 5 y de 3.3 V para tarjetas duales.

Otra gran mejora del bus PCI es el hecho de que el bus es PnP (Plug and Play - enchufar y listo). Por tanto, las tarjetas PCI no tenían jumpers o interruptores integrados para configurarlas. Al conectar una tarjeta, el sistema PnP la identifica y prepara automáticamente para ser usada (al contrario que las ISA que necesitaban de programas externos para configurarlas). Esto suponía complicar la arquitectura de la placa, más complejidad, pero era algo necesario. Ya empezaba a haber una amplia gama de periféricos y se necesitaban establecer diferentes clases separadas por criterios de velocidad.

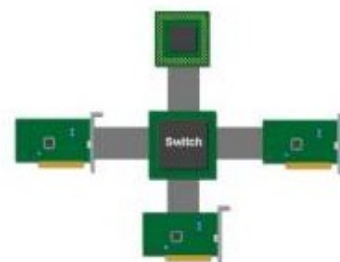
2.4.3.- El PCI-Express

Durante el 2001 se investiga sobre un nuevo bus de alta velocidad, un bus 3-GIO (third generation I/O). A mediados del 2002 se aprueba un nuevo bus: El PCI-Express. Bus considerado de 3ª generación porque con él se pretende remplazar a los buses ISA (1ª generación) y PCI (2ª generación) en los Pcs.

En el mundo del PCI-Express el recurso compartido deja de ser el bus y pasa a ser el switch: cada dispositivo en el sistema tiene un acceso directo y exclusivo al switch como podemos ver en el dibujo.



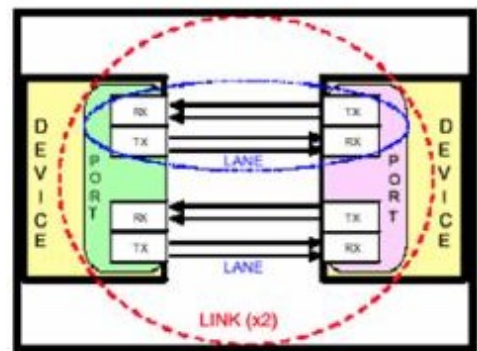
PCI



PCI Express

Las primeras generaciones del bus PCI estaban diseñadas como un arquitectura de transferencia de datos paralela, en la que los múltiples bits eran mandados simultáneamente por muchas rutas en paralelo. Cuanto más bits se podía mandar al mismo tiempo, más rápido era el rendimiento del bus. El problema era que los bits en un bus paralelo llegan a distintos instantes de tiempo y han de ser sincronizados; problema que se va agrandando a medida que se aumenta la velocidad del bus o se tiene que variar la longitud de los cables del bus, ya que tienen que medir exactamente lo mismo. Un bus en serie es mucho más simple: Manda 1 único bit a la vez y por un único cable a ticks de velocidad más altos de los que un sistema paralelo permitiría. Combinando múltiples rutas de datos en serie, el rendimiento de velocidad puede sobrepasar desmesuradamente al de los buses paralelos.

El PCI express es un bus serial diseñado para mantener la compatibilidad con los drivers y controladores del PCI paralelo. También optimiza el diseño de las placas base, pues su tecnología serie precisa tan solo de un cable frente a los 32 necesarios para el PCI paralelo. Una conexión serie de PCI-Express consta de una conexión dual, utilizando 2 pares de señales diferencialmente dirigidas de baja tensión (un par de recepción y otro de envío - 4 cables) llamados "lane". Cada uno de estos caminos ofrece una velocidad de hasta 250 Mbps y hay diseños de 1, 2, 4, 8, 16 o 32 "lanes". Por ejemplo, una configuración con 32 lanes puede enviar 8 bits en cada dirección simultáneamente, ofreciendo hasta 2000 Mbps de ancho de banda (cada canal) y necesitando solo 40 pines. Ni punto de comparación con el PCI paralelo que tiene solo 133 Mbps de ancho de banda y necesita de más de 100 pines para llevar las señales.



Físicamente las ranuras PCI-E aparecen al lado de las PCI en la placa base y podemos ver su forma en la imagen de al lado.

La primera frecuencia a la que se usó el PCI-E fue a 2.5 Ghz y las especificaciones permitirán ampliarla a más de 10 Ghz. Combinando esta frecuencia al uso de 32 lanes, PCI-E podrá soportar anchos de banda de 32 Gbps.

PCI-E fue diseñado para partirla y remplazar a la mayoría de los buses del sistema. No solo se diseñó para reemplazar al PCI, sino que se hizo con la idea de poder usarse para reemplazar la Hub architecture de Intel (u otras interfaces de alta velocidad similares a esta), conexiones entre chipsets de la placa... También remplazará a interfaces de video como el AGP, que veremos a continuación, y actuará como bus base para otras interfaces como el serial ATA, USB 2.0, Gigabit Ethernet...

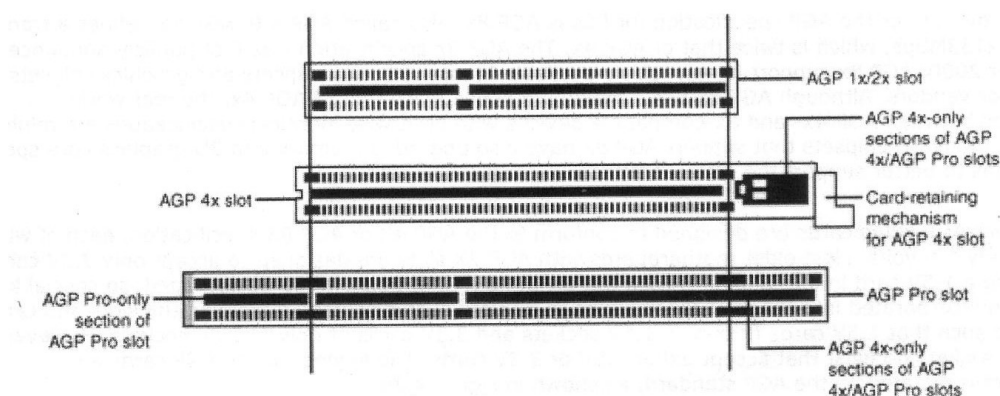
Sin embargo esto no se esperaba que sucediese de la noche a la mañana. Se seguirán incluyendo el PCI, AGP y otros buses durante algunos años más.

Con el tiempo PCI-E aparecerá más, a la vez que el PCI y el AGP lo harán menos.

2.4.4.- Bus AGP: Accelerated Graphics Port

Intel crea el AGP como un nuevo bus específicamente diseñado gráficos de alto rendimiento y un apoyo de vídeo. AGP está basado en el PCI, pero con bastantes mejoras añadidas y física, eléctrica y lógicamente independiente el PCI. El conector AGP es muy similar al PCI, sin embargo, aunque estemos hablando del “bus AGP”, realmente es un puerto, ya que solamente está diseñado para una tarjeta de video y sólo puede conectarse un dispositivo a la vez (a diferencia de un bus que pueden conectarse varios). Intel saca el AGP en julio del 96, definido a una frecuencia de reloj de 66 Mhz y funcionando a un voltaje de 3.3 V y será sustituidos en las computadores por el PCI-Express en el 2002.

El último fue el 8x, funcionando a 1,5 V y un ancho de banda de 2 Gbps. A la hora de conectar las tarjetas AGP en los slots, había que tener en cuenta el voltaje de cada uno. Si conectábamos una tarjeta de 1,5 V en una de 3, tanto la tarjeta como la placa podrían resultar dañadas. Para proteger esto se incorporaron clavijas especiales en los conectores de la placa del AGP. Las tarjetas de 1,5 solo podrían conectarse en slots de 1,5 y lo mismo para las de 3 V. Hubo también un slots universal que aceptaba tarjetas tanto de un voltaje como de otro.



PRECAUCIÓN

Si vas a usar tarjetas AGP, hay placas que requieren que uses tarjetas de 4x/8x de 1,5 V. Antes de nada es recomendable que mires las especificaciones de la placa.

El puerto AGP trabaja a una frecuencia de 66 Mhz (el doble que el PCI convencional). En el AGP original 1x se hace una sola transferencia de datos en cada ciclo, con un ancho de 32 bits, obteniendo un ancho de banda de 266 Mbps. También en la especificación original del AGP se define el modo 2x, el cual puede realizar dos transferencias en cada ciclo, consiguiendo así un ancho de banda de 533 Mbps. Y así sucesivamente para las versiones 4x y 8x.

AGP Modes Showing Clock Speeds and Bandwidth

AGP Bus Type	Bus Width (Bits)	Bus Speed (MHz)	Data Cycles per Clock	Bandwidth (MBps)
AGP	32	66	1	266
AGP 2x	32	66	2	533
AGP 4x	32	66	4	1066
AGP 8x	32	66	8	2133

Una ventaja importante por la que Intel diseñó el AGP fue para permitir a la tarjeta de video tener una alta velocidad de conexión directamente con la RAM, lo que hacía de una solución rápida y potente el integrar el video a un bajo coste. El AGP permite cargar texturas en la RAM principal, es decir, ya no se limita a la capacidad de memoria que tenga la tarjeta, posibilitando así la mejora de texturas en juegos y aplicaciones 3D al almacenar estas en la RAM y usarlas cuando se necesiten.

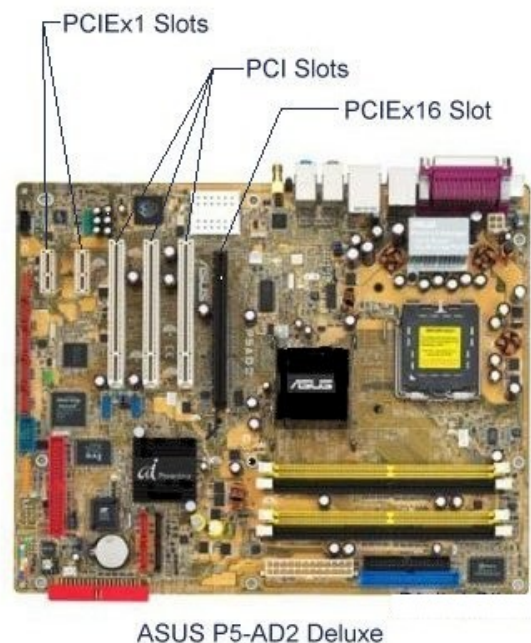
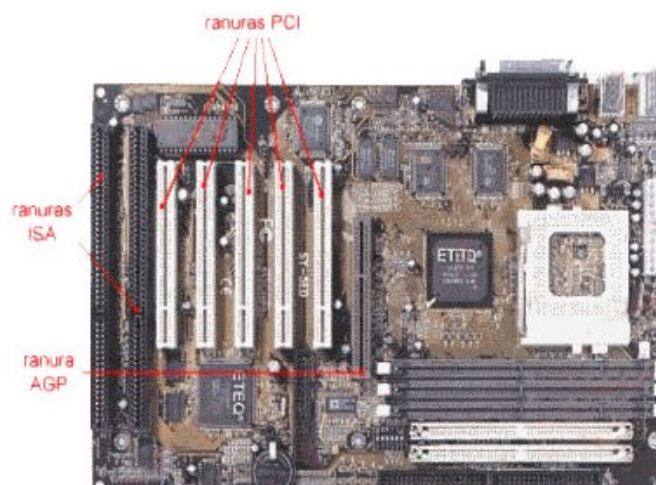
De todas maneras, aunque esto esté genial, si comparamos el AGP más logrado (8x a 2133 Mbps) es 16 veces más rápido que el PCI (133 Mbps) pero el doble de lento que el PCI-E (16x a 4000 Mbps). Por tanto, poco a poco, se fue sustituyendo el AGP por el PCI-E hasta el momento que a llegado a ni aparecer en las placas.

Distintas ranuras para tarjetas de los buses



2 Placas base con algunos de los conectores vistos.

La primera bastante más antigua que la segunda.



3.- Recursos del sistema

Los recursos del sistema son los canales de comunicación, las direcciones y otras señales hardware que los dispositivos usan para comunicarse por el bus. A bajo nivel estos recursos típicos son:

- Acceso a memoria.
- Canales IRQ (Interrupt Request – Interrupciones)
- Canales DMA (Direct Memory Acces – Accesos directos a memoria)
- Puertos de E/S

En esta parte nos centraremos en ver los típicos problemas que se pueden tener con cada uno de estos recursos.

Las IRQs es el recurso que nos pueden causar más problemas, ya que tienen mucha demanda: todas las tarjetas usan canales IRQ. Las DMA son menos usadas por las tarjetas así que nos darán menos problemas. Los puertos de E/S también son usados por todos los dispositivos hardware del bus; sin embargo disponemos de 64 Kb de ellos, o sea que hay de sobra para su uso.

Para cualquier recurso individual que se usa, hay que estar seguros de que solamente una tarjeta o función va a ser uso de ese recurso, ya que, en la mayoría de los casos, no debe o no puede ser compartido.

Estos recursos son requeridos y usados por la mayoría de los componen del sistema para comunicarse con el sistema y cumplir sus objetivos. Cada adaptador necesita sus recursos específicos. Por ejemplo, un puerto de comunicaciones en serie, necesita de un canal IRQ y un rango de puertos de direcciones de E/S. Sin embargo, una tarjeta de sonido necesita, a parte de lo nombrado, un canal DMA.

A medida que los ordenadores aumentan su complejidad, es lógico que la posibilidad de que haya conflictos de recursos aumente también. Los sistemas actuales, al tener tantos dispositivos adicionales, realmente pueden tener fallos a la hora de configurarse y asignar los recursos. La mayoría de los adaptadores de tarjetas permiten modificar los recursos asignados mediante un software PnP (que viene con la tarjeta), o incluso con el administrador de dispositivos de Windows. Por tanto, si hiciera falta, podríamos cambiar la configuración inicial de la tarjeta y adaptarla a nuestras necesidades.

3.1.- Interrupciones

Los canales de petición de interrupciones (o interrupciones hardware) son usados por varios dispositivos para mandar una señal a la placa de que tienen una necesidad que debe ser cubierta (como por ejemplo cuando un alumno levanta la mano para preguntar).

Estos canales de interrupción están implementados por cables en la placa y en los slots de conexiones. Cuando una interrupción en particular es invocada, una rutina especial se apodera del sistema, la cual, primero salva todos los registros que en tenga la CPU en una pila y dirige al sistema a la tabla de vectores de interrupciones. Esta tabla de vectores contiene una lista de direcciones de memoria que corresponden a los canales de interrupciones. Dependiendo de cual ha sido la interrupción, la tarea correspondiente a ese canal será ejecutada.

Después de que la rutina particular del software termine con la función que necesite la

tarjeta, el controlador software de interrupciones devuelve los registros guardados a la CPU y el sistema continúa con lo que estaba haciendo antes de la interrupción.

En cuanto a la prioridad de las interrupciones, esta está generalizada dependiendo de su numeración (excepto algún caso particular). Las interrupciones de menor numeración son las de más prioridad. Las interrupciones de gran prioridad tienen preferencia frente a las de menor prioridad, interrumpiendo a estas si fuera necesario. Como resultado, tenemos un sistema con varias interrupciones aconteciendo concurrentemente, anidadas unas con otras.

Internal Stack Overflow

Un antiguo error que nos podíamos encontrar en MS-DOS era el siguiente: Si sobrecargamos el sistema, haciendo que se quede la pila sin recursos (por ejemplo muchas interrupciones que se generan muy rápido), ocurrirá un error interno de desbordamiento de la pila y el sistema parará "*Internal Stack Overflow – system halted*"). Si esto nos ocurriese, podíamos compensarlo modificando en el archivo "*CONFIG.SYS*" los parámetros STACKS para incrementar los recursos disponibles de la pila. Sin embargo en Windows no se verá este error.

El bus ISA usa un detector de interrupciones denominadas Edge-tiggered (borde sensible), en el que una interrupción es detectada por un cambio de voltaje en un cable en particular del conector. Debido a esto no se pueden compartir interrupciones, ya que la placa no puede reconocer que slot es el que está la tarjeta que ha mandado la señal. Los resultados son confundidos si más de una tarjeta es puesta para usar una misma interrupción. Cada interrupción, por tanto, es diseñada para un dispositivo hardware concreto, y que la mayoría de las veces no puede ser compartida.

Originalmente IBM desarrolló la forma de compartir las interrupciones en el bus ISA, pero pocos dispositivos siguieron las reglas necesarias para hacer esto posible. El PCI permitirá de forma intrínseca compartir las interrupciones, de hecho, todas las tarjetas y ranuras PCI manejan interrupciones internamente. El verdadero problema es que nos encontramos técnicamente con dos sets de interrupciones hardware: las PCI y las ISA. Para que las tarjetas PCI puedan trabajar en el PC, primero hay que adaptarlas a las interrupciones ISA, que están configuradas para no compartir. En la mayoría de los casos sí que se puede asignar una interrupción no conflictiva para cada tarjeta.

El conflicto de asignar IRQs de ISA para interrupciones PCI causaba muchos problemas de configuración para los usuarios, incluso después de la aparición de desarrollado sistema operativo Windows y su tecnología PnP.

La solución para el problema fue llamada PCI IRQ Steering. PCI IRQ Steering permite a los sistemas operativos con PnP (como Windows) asignar dinámicamente, a las tarjetas PCI, estándares de interrupciones y así poder usar varias tarjetas la misma interrupción.

Las interrupciones hardware se pueden deshabilitar cuando la CPU esté siendo usada para realizar operaciones críticas. Es cosa de la BIOS y de los programas de manejo de interrupciones hacer que el sistema haga correctamente y eficazmente estas cosas.

Debido a que las interrupciones no pueden ser compartidas en los buses ISA, a menudo se corre el peligro que al añadir una nueva tarjeta al sistema nos quedemos sin interrupciones. Si dos tarjetas usan el mismo IRQ para mandar una señal al sistema, entrarían en conflicto y uno u otro no

funcionarían correctamente. En los siguientes apartados vamos a ver, tanto las IRQs que algunos dispositivos estándares usan, como las que quedan libres en el sistema.

3.1.1.- Interrupciones del bus ISA de 8 bits

En los Pcs y XT (sistemas basados en CPU de 8 bits), con buses ISA de 8 bits, nos encontramos con 8 diferentes interrupciones hardware externas. Los usos típicos los vamos a ver en esta tabla:

8-Bit ISA Bus Default Interrupt Assignments

IRQ	Function	Bus Slot
0	System Timer	No
1	Keyboard Controller	No
2	Available	Yes (8-bit)
3	Serial Port 2 (COM2:)	Yes (8-bit)
4	Serial Port 1 (COM1:)	Yes (8-bit)
5	Hard Disk Controller	Yes (8-bit)
6	Floppy Disk Controller	Yes (8-bit)
7	Parallel Port 1 (LPT1:)	Yes (8-bit)

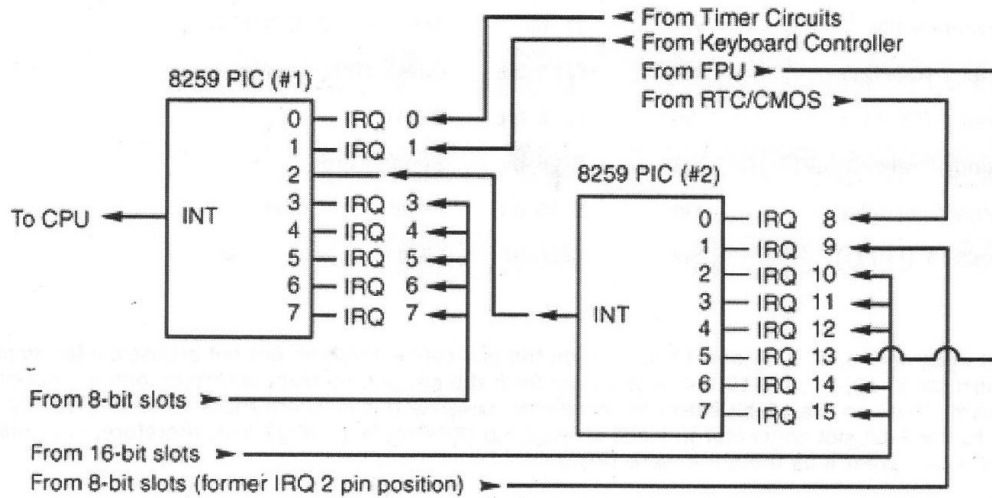
Instalar en estos sistemas varios dispositivos que necesiten el servicio del sistema IRQ es algo frustrante, ya que la única forma que teníamos de resolver el problema de la escasez de interrupciones era quitar el dispositivo que menos usáramos o que no fuéramos a usar para usar el otro.

3.1.2.- Interrupciones del bus ISA de 16 bits, EISA y el MCA

Con la llegada de los procesadores 286 se incrementa el número de interrupciones hardware que el bus podía soportar a 16, usando para ello dos controladores de interrupciones Intel 8259 y canalizando las interrupciones generadas por el segundo controlador a través no usado IRQ2 del primer controlador. Este arreglo nos provee de 16 IRQs efectivos y hace al IRQ2 inaccesible.

Enviando todas las interrupciones desde el segundo controlador IRQ a través del IRQ2 del primero, todas estas interrupciones tendrán un nivel de preferencia mayor que el IRQ3, es decir, la IRQ15 tendrá mayor preferencia que la IRQ3 debido a que está conectada al primer controlador mediante el IRQ2. Veámoslo más claro en la imagen:

Interrupt controller cascade wiring.



En la imagen vemos como el controlador 8259 #2 está conectado al controlador 8259 #1 por el IRQ2. Para evitar que las placas hicieran uso del controlador IRQ2 y evitar conflictos innecesario, lo que se hizo fue que se diseñó un camino a una nueva interrupción (IRQ9) para ocupar el hueco abierto por el IRQ2. Por tanto, si instalamos una tarjeta que use el IRQ2, en verdad estaría usando el IRQ9.

La siguiente tabla muestra los usos típicos de las interrupciones en el bus ISA de 16 bits y el PCI/AGP de 32 bits. EISA y MCA usan un mapa IRQ similar.

16/32-Bit ISA/PCI/AGP Default Interrupt Assignments

IRQ	Standard Function	Bus Slot	Card Type	Recommended Use
0	System Timer	No		
1	Keyboard Controller	No		
2	2nd IRQ Controller Cascade	No		
8	Real-Time Clock	No		
9	Avail. (as IRQ2 or IRQ9)	Yes	8/16-bit	Network Card
10	Available	Yes	16-bit USB	
11	Available	Yes	16-bit	SCSI Host Adapter
12	Mouse Port/Available	Yes	16-bit	Mouse Port
13	Math Coprocessor	No		
14	Primary IDE	Yes	16-bit	Primary IDE (hard disks)
15	Secondary IDE	Yes	16-bit	2nd IDE (CD-ROM/Tape)
3	Serial 2 (COM2:)	Yes	8/16-bit	COM2:/Internal Modem
4	Serial 1 (COM1:)	Yes	8/16-bit	COM1:
5	Sound/Parallel 2 (LPT2:)	Yes	8/16-bit	Sound Card
6	Floppy Controller	Yes	8/16-bit	Floppy Controller
7	Parallel 1 (LPT1:)	Yes	8/16-bit	LPT1:

0, 1, 2, 8 y 13 no son conectores bus y no son accesibles desde tarjetas. Las interrupciones 10, 11, 12, 14 y 15 pertenecen al segundo controlador de interrupciones y son accesibles solo desde tarjetas de 16 bits. Las 3, 4, 5, 6 y 7 son interrupciones para tarjetas de 8 o 16 bits. La 9 es la sustitutiva del 2, como vimos antes, aunque solo accesible por tarjetas de 8 bits.

3.1.3.- Interrupciones PCI

El bus PCI está provisto de interrupciones hardware que pueden ser usadas por los dispositivos PCI para señalar al bus que necesitan ser atendidos. Las cuatro interrupciones PCI son INTA#, INTB#, INTC# e INTD#. Estas interrupciones son llamadas “level-sensitive”, nombre que viene de que es posible que diversos dispositivos del PCI, manden varios voltajes en la misma línea física de la interrupción, permitiendo que el procesador determine que dispositivo generó la interrupción.

En la especificación del PCI, una de las reglas es que cada dispositivo PCI puede tener acceso a una de las cuatro interrupciones (INT#x) y ordinariamente INT#A es usado por la ranura 1, INT#B por el slot 2...

Para que el bus PCI funcione en el PC, las interrupciones PCI deben ser puenteadas con las ISA. Debido a que las interrupciones ISA no pueden ser compartidas, en la mayoría de los casos, las tarjetas PCI, usando el INT#A, deben ser puenteadas a una interrupción ISA diferente no compartible. Por ejemplo, si tenemos un sistema con 4 slots PCI y 4 tarjetas instaladas, cada una usando la interrupción PCI INT#A, estas tarjetas deben de ser puenteadas a diferentes IRQ ISA disponibles, como el IRQ 9, 10, 11 o el 5.

Encontrar un IRQ único para cada dispositivo en los buses ISA y PCI siempre ha sido un problema: no son bastante para satisfacer todo. Configurar 2 dispositivos ISA para el mismo IRQ no es posible, pero compartir IRQs entre dispositivos PCI si que lo es. Los sistemas actuales con Windows (a partir del W95B) soportan una función conocida como PCI IRQ steering, que veremos a continuación. Los sistemas anteriores a este no lo poseían.

Generalmente la bios asigna un único IRQ a los dispositivos PCI. Si el sistema tiene PCI IRQ steering y está activado, es Windows quien asigna IRQs a los dispositivos PCI. La bios al arrancar, asigna normalmente los IRQs a los PCI, la cosa es que si hubiera algún conflicto o si fuera necesario, Windows tiene la capacidad de cambiar estas asignaciones para eliminar el problema. En el caso de que no hubiera suficientes IRQs disponibles, el IRQ Steering permite a Windows asignar un IRQ a varios dispositivos PCI, permitiendo así a todos los dispositivos trabajar simultáneamente. Sin el steering lo que se haría sería desactivar dispositivos.

A pesar de que las interrupciones PCI (INT#X) pueden y deben ser compartidas, cada tarjeta o dispositivo que podría compartir una interrupción PCI tiene que ser, como dijimos antes, dirigido o mapeado a un único IRQ ISA, el cual, normalmente, no puede ser compartido. Se podría tener varios dispositivos mapeados al mismo IRQ isa solo si:

- No hay dispositivos ISA usando ese IRQ
- La Bios y el SO soportan el PCI IRQ Steering
- Éste está activado

Sin un soporte PCI IRQ Steering, las capacidades de compartir de las interrupciones PCI tienen poca ventaja, ya que todas las asignaciones PCI-To-ISA deben ser únicas. Sin PCI IRQ Steering puedes quedarte sin interrupciones ISA disponibles, una importante razón por la que se debía cambiar de SO a W98.

Hay que tener en cuenta que no es tan complicado quedarse sin IRQs ISA, ya que no solamente las tarjetas PCI usan el bus. Podemos tener dispositivos internos usando el bus aún cuando todas las ranuras PCI están vacías. Por ejemplo, nos podemos encontrar con controladores IDE o con los USB pululando por el bus PCI. Normalmente los controladores IDE están mapeados al IRQ 14 y 15 del ISA, y los USB a alguno de los IRQ 9, 10, 11 y 5.

3.1.4.- Advanced programmable interrupt controller

Como sustituto del par de controladores de interrupciones 8259, Intel desarrolló el APIC a mediados de los 90. La mayoría de los procesadores desde el Pentium original tienen el APIC integrado.

El APIC provee soporte para múltiples procesadores, aunque también puede usarse en ordenadores con un solo procesador. El mayor beneficio del APIC para un procesador es que proporciona IRQs virtuales. La mayoría de las implementaciones de los APIC soportan hasta 24 IRQs virtuales. Hasta el W2000 se tienden a colocar los IRQs PCI en la gama tradicional de ISA de 15; sin embargo el XP ya hace pleno uso del APIC cuando éste está instalado en el equipo. Con el XP APIC limita las IRQ que se comparten para permitir a los dispositivos funcionar mejor y con menos conflictos. Por ejemplo, en un XP con APIC, las IRQs asignadas son:

- PCI IRQ 16: Audio/Video AGP (Shared)
- PCI IRQ 17: Controlador USB 1.1
- PCI IRQ 18: Controlador USB 1.1
- PCI IRQ 19: Tarjeta de red 10/100 – Controlador USB 2.0 (Shared)
- PCI IRQ 21: Controlador USB 1.1 – Controlador USB 2.2

Los tradicionales IRQs ISA son usados solo para dispositivos ISA, librándonos de conflictos entre ISA y PCI.

3.2.- Canales DMA

Los canales de acceso directo a memoria (DMA) son usados por dispositivos de comunicación que pueden enviar y recibir información a altas velocidades. Por ejemplo, un puerto en serie o paralelo no usa el canal DMA, pero una tarjeta de sonido ISA o un adaptador SCSI sí que podría. Los canales DMA pueden ser compartidos si los dispositivos no los necesitan simultáneamente. Por ejemplo, podemos tener un adaptador de red y un adaptador de BackUp compartiendo el canal DMA1, pero no podemos hacer BackUps mientras la red está funcionando. Si quisiéramos hacer ambas cosas deberíamos estar seguros que cada adaptador usa un único canal DMA.

Hay varios tipos de DMA en los PCs modernos. Los canales DMA que veremos en esta parte son los del bus ISA. Otros buses como el ATA/IDE usados por discos duros, tienen diferentes usos del DMA.

3.2.1.- Canales DMA para el bus ISA de 8 bits

En el ISA de 8 bits sin 4 los canales DMA que proporcionan alta velocidad de transferencia entre los dispositivos de E/S. 3 de los canales están disponibles para slots de expansión.

8-Bit ISA Default DMA-Channel Assignments		
DMA	Standard Function	Bus Slot
0	Dynamic RAM Refresh	No
1	Available	Yes (8-bit)
2	Floppy disk controller	Yes (8-bit)
3	Hard disk controller	Yes (8-bit)

3.2.2.- Canales DMA para el bus ISA de 16 bits

Desde la introducción de las CPUs 286, el bus ISA ha tenido 8 canales DMA, de los cuales, 7 disponibles para slots de expansión. Al igual que como pasó con las IRQ, los canales DMA añadidos fueron creados en un controlador DMA secundario conectado al primario en cascada. En este caso, el canal 4 del DMA fue el elegido como conector. Los canales del 0-4 estaban disponibles para las transferencias de 8 bits y los 5-7 para transferencias de 16 bits.

DMA	Standard Function	Bus Slot	Card Type	Transfer	Recommended Use
0	Available	Yes	16-bit	8-bit	Sound
1	Available	Yes	8/16-bit	8-bit	Sound
2	Floppy Disk Controller	Yes	8/16-bit	8-bit	Floppy Controller
3	Available	Yes	8/16-bit	8-bit	LPT1: in ECP Mode
4	1st DMA Controller Cascade	No		16-bit	
5	Available	Yes	16-bit	16-bit	Sound
6	Available	Yes	16-bit	16-bit	Available
7	Available	Yes	16-bit	16-bit	Available

El único canal DMA estándar usado por todos los sistemas es el DMA2, universalmente usado por el controlador de disquetes. Los canales 1 y 5 son usados principalmente para tarjetas ISA de sonido y por las nuevas PCI. El DMA 3 es usado cuando un puerto paralelo es configurado para trabajar en modo EPC o EPP/EPC. Algunos sistemas no estándares, como Packard Bell, usan por defecto el DMA1 en lugar del 3 para el puerto paralelo. Aunque un jumper block en la placa base de cualquiera de estos sistemas puede hacer que sea el DMA3 el estándar como puerto paralelo y evitar conflictos con tarjetas de sonido que usen el 1.

Nota.

Aunque el DMA 0 aparezca como un conector de 16 bits y solo pueda ser usado por tarjetas de 16 bits, solo puede transferir 8 bits. Debido a esto es importante que no veamos el DMA0 como una opción para conectar tarjetas de 16 bits.

3.3.- Direcciones de los puertos de E/S

Los puertos de E/S del ordenador permiten la comunicación entre los dispositivos y el software del sistema. Son equivalentes a dos vías de canales de radio. Si quieres hablar con un puerto serial, necesitas saber que puerto de E/S (canal de radio) está escuchando. Del mismo modo, si quieres recibir datos desde un puerto serial, necesitaras escuchar el mismo canal en el que el dispositivo está transmitiendo.

A diferencia de los canales DMA e IRQs, el sistema tiene miles de puertos de E/S, exactamente 65.535 (0000h – FFFFh), y cada dispositivo suele usar varios puertos. El problema en este caso es el preocuparse a la hora de configurar dos dispositivos que usen el mismo puerto. Sin embargo, actualmente todos los sistemas incluyen el PnP, el que resuelve el conflicto y selecciona puertos alternativos para los dispositivos en conflicto.

Una confusión típica es que, como los puertos fueron diseñados con numeración hexadecimal, se confundan con direcciones de memoria. Los puertos de E/S son PUERTOS no memoria. La diferencia entre esto es que cuando tú mandas un dato por la dirección de memoria 1000h, quedará almacenado en la SIMM o DIMM, mientras que si lo mandas por el puerto de E/S 1000h, estás mandando fuera del bus el dato por ese canal, esperando que algún “oyente” lo escuche. Si no hay nadie escuchando ese canal, el dato alcanzará el final del bus y será desechado.

Los drivers de los dispositivos son los que principalmente interactúan con los dispositivos en los distintos puertos. El driver debe conocer los puertos con los que necesita trabajar un dispositivo. Esto no es problema ya que el driver y el dispositivo se fabrica en la misma compañía.

A continuación, a modo de curiosidad, se detallan unas tablas con las direcciones de los puertos de E/S que usan distintos dispositivos, aunque para saber con mayor exactitud las direcciones, es conveniente que se lean los manuales de los dispositivos.

Bus-Based Device Port Addresses

Address (hex)	Size	Description
01300133	4 bytes	Adaptec SCSI adapter (alternate)
01340137	4 bytes	Adaptec SCSI adapter (alternate)
0168016F	8 bytes	Fourth IDE interface
01700177	8 bytes	Secondary IDE interface
01E801EF	8 bytes	Third IDE interface
01F001F7	8 bytes	Primary IDE/AT (16-bit) hard disk controller
02000207	8 bytes	Gameport or joystick adapter
02100217	8 bytes	IBM XT expansion chassis
02200233	20 bytes	Creative Labs Sound Blaster 16 audio (default)
02300233	4 bytes	Adaptec SCSI adapter (alternate)
02340237	4 bytes	Adaptec SCSI adapter (alternate)
0238023B	4 bytes	MS bus mouse (alternate)
023C023F	4 bytes	MS bus mouse (default)
0240024F	16 bytes	SMC Ethernet adapter (default)
02400253	20 bytes	Creative Labs Sound Blaster 16 audio (alternate)
0258025F	8 bytes	Intel above board
0260026F	16 bytes	SMC Ethernet adapter (alternate)
02600273	20 bytes	Creative Labs Sound Blaster 16 audio (alternate)

02700273	4 bytes	Plug and Play I/O read ports
0278027F	8 bytes	Parallel port 2 (LPT2)
0280028F	16 bytes	SMC Ethernet adapter (alternate)
02800293	20 bytes	Creative Labs Sound Blaster 16 audio (alternate)
02A002AF	16 bytes	SMC Ethernet adapter (alternate)
02C002CF	16 bytes	SMC Ethernet adapter (alternate)
02E002EF	16 bytes	SMC Ethernet adapter (alternate)
02E802EF	8 bytes	Serial port 4 (COM4)
02EC02EF	4 bytes	Video, 8514, or ATI standard ports
02F802FF	8 bytes	Serial port 2 (COM2)
03000301	2 bytes	MPU-401 MIDI port (secondary)
0300030F	16 bytes	SMC Ethernet adapter (alternate)
03200323	4 bytes	XT (8-bit) hard disk controller
0320032F	16 bytes	SMC Ethernet adapter (alternate)
03300331	2 bytes	MPU-401 MIDI port (default)
03300333	4 bytes	Adaptec SCSI adapter (default)
03340337	4 bytes	Adaptec SCSI adapter (alternate)
0340034F	16 bytes	SMC Ethernet adapter (alternate)
0360036F	16 bytes	SMC Ethernet adapter (alternate)
0366	1 byte	Fourth IDE command port
0367, bits 6:0	7 bits	Fourth IDE status port
03700375	6 bytes	Secondary floppy controller
0376	1 byte	Secondary IDE command port
0377, bit 7	1 bit	Secondary floppy controller disk change
0377, bits 6:0	7 bits	Secondary IDE status port
0378037F	8 bytes	Parallel Port 1 (LPT1)
0380038F	16 bytes	SMC Ethernet adapter (alternate)
0388038B	4 bytes	Audio - FM synthesizer
03B003BB	12 bytes	Video, Mono/EGA/VGA standard ports
03BC03BF	4 bytes	Parallel port 1 (LPT1) in some systems
03BC03BF	4 bytes	Parallel port 3 (LPT3)

03C003CF	16 bytes	Video, EGA/VGA standard ports
03D003DF	16 bytes	Video, CGA/EGA/VGA standard ports
03E6	1 byte	Third IDE command port
03E7, bits 6:0	7 bits	Third IDE status port
03E803EF	8 bytes	Serial port 3 (COM3)
03F003F5	6 bytes	Primary floppy controller
03F6	1 byte	Primary IDE command port
03F7, bit 7	1 bit	Primary floppy controller disk change
03F7, bits 6:0	7 bits	Primary IDE status port
03F803FF	8 bytes	Serial port 1 (COM1)
04D004D1	2 bytes	Edge/level triggered PCI interrupt controller
05300537	8 bytes	Windows sound system (default)
0604060B	8 bytes	Windows sound system (alternate)
0678067F	8 bytes	LPT2 in ECP mode
0778077F	8 bytes	LPT1 in ECP mode
0A200A23	4 bytes	IBM Token-Ring adapter (default)
0A240A27	4 bytes	IBM Token-Ring adapter (alternate)
0CF80CFB	4 bytes	PCI configuration address registers
0CF9	1 byte	Turbo and reset control register
0CFC0CFF	4 bytes	PCI configuration data registers
FF00FF07	8 bytes	IDE bus master registers
FF80FF9F	32 bytes	Universal serial bus
FFA0FFA7	8 bytes	Primary bus master IDE registers
FFA8FFAF	8 bytes	Secondary bus master IDE registers

4.- Resolución de conflictos de recursos

Los recursos en el sistema, como hemos visto, son limitados y desafortunadamente, la demanda de estos recursos parece no serlo. Como continuamente estamos añadiendo más y más tarjetas y periféricos, el riesgo de conflictos entre recursos aumenta. Actualmente y como norma general, al ser los sistemas PnP, los conflictos se solucionan automáticamente, aunque no siempre pasa esto.

¿Cómo saber cuando tenemos algún conflicto de recursos? Vamos a ver algunas formas de detectarlo:

- El dispositivo transfiere datos erróneamente
- El sistema falla frecuentemente
- La tarjeta de sonido no emite bien los sonidos
- El ratón no va
- Se ve la pantalla mal sin razón aparente
- La impresora imprime cosas raras
- El ordenador se inicia en modo a prueba de fallos o como la última configuración correcta conocida.

En Windows también podemos detectar un conflicto en el administrador de dispositivos mediante una luz amarilla o roja.

A continuación aprenderemos algunos pasos para solucionar conflictos de recursos o detectarlos cuando se den.

¡CUIDADO!

Debemos de tener cuidado con el diagnóstico de conflictos de recursos: Tal vez lo que tenemos es un virus y no un problema de recursos. Muchos virus están diseñados para mostrarse como interferencias o problemas periódicos, pudiéndose confundir como un conflicto de recursos. Si no estás seguro que te pasa, lo mejor es pasarle un escaneo del antivirus al PC para por si acaso.

La primera medida para resolver conflictos es prevenirlos. Especialmente si estamos construyendo un nuevo sistema debemos seguir ciertos pasos que nos ayudarán a evitar algunos quebraderos de cabeza. Uno de ellos es escapar del uso de viejos dispositivos ISA. Por definición los ISA no pueden compartir IRQs y esto es uno de los recursos que más se demandan. Es una mejor opción usar tarjetas PCI o AGP, que pueden compartir recursos IRQs con el IRQ Steering.

Consejo:

Los puertos serial, paralelo y el PS2 del ratón, encontrados todavía en sistemas recientes, son todos dispositivos ISA, que no pueden compartir IRQs. Si no usas estos puertos, puedes usar estos IRQs para otros dispositivos si:

Desactivas los puertos no usados en la BIOS

Configuras la BIOS para usar el IRQ formalmente con otro dispositivo con configuración PnP (normalmente esto es automático).

Por otro lado también te puede ayudar el instalar las tarjetas en un orden determinado y no todas a la vez. Modificando esta secuencia, a menudo ayuda a resolver conflictos, ya que muchas tarjetas usan sólo uno o dos IRQs específicos para cada marca o modelo de tarjeta. Instalando las tarjetas en un orden determinado, el PnP puede trabajar más fácilmente para los conflictos de IRQ que puedan causar las configuraciones por defecto de las tarjetas.

La primera vez que se inicie un nuevo sistema que hemos montado o se ha añadido alguna mejora, lo primero que se debe hacer es chequear el setup de la BIOS. Si el SO tiene soporte PnP (W98, Me, 2000, XP...), debes asegurarte que esté activado. Si no lo tiene (W 95, Windows NT), asegúrate que está desactivado de la BIOS.

En un arranque inicial se recomienda una configuración mínima, con solo una tarjeta gráfica, memoria y dispositivos de almacenamiento. Esto hace que sea difícil encontrarnos con conflictos en la configuración inicial. Si la placa viene con un CD de drivers específicos para el chipset u otros puertos integrados en la placa, este es el momento para instalarlos.

Después de que se haya configurado lo básico del sistema es hora de empezar a añadir uno a uno los demás dispositivos en un orden determinado. Así que apagamos el PC, insertamos los dispositivos, encendemos y procedemos a instalar los drivers necesarios y configurar el dispositivo. Seguramente tendremos que reiniciar de nuevo el PC cuando hemos completado la configuración.

Consejo de instalación de tarjetas adicionales.

1. Tarjeta de sonido
2. Módem
3. Tarjeta de red
4. Dispositivos auxiliares de video (Decodificadores MPEG, aceleradoras 3D...)
5. Adaptadores SCSI
6. Cualquier otra

4.1.- Resolviendo conflictos manualmente

En el pasado, la única forma de resolver conflictos manualmente era quitar la tapa de la carcasa del PC y empezar a cambiar los switches y los jumpers de la tarjeta hasta conseguir lo deseado. Afortunadamente hoy día esto es mucho más sencillo con el PnP ya que toda la configuración se hace mediante el administrador de dispositivos del SO (aunque las primeras tarjetas con PnP todavía tenían jumpers o switches para configurarlas).

Siempre que vayamos a modificar alguna cosa es conveniente anotar la configuración previa que estaba antes de guardar los cambios. Así, si fuera necesario, podríamos recuperar la configuración inicial.

Ten a mano siempre los manuales de todas las tarjetas, podrías necesitarlos, y sobre todo, si las tarjetas se configuran manualmente o si tienes que cambiarla a modo PnP.

Si vas a probar configuraciones se recomienda que tengas en mente estas tres preguntas:

- ¿Cuándo apareció el conflicto por primera vez?

Si, por ejemplo, apareció después de instalar una tarjeta nueva, blanco y en botella.

- ¿Hay dos dispositivos similares que no tiran?

Por ejemplo, si el módem, con puertos serial, o el ratón que usa puerto COM, no caminan, las posibilidades que los dispositivos están en conflicto con otros son bastante altas.

- ¿Ha tenido otra gente el mismo problema? ¿Cómo lo resolvieron?

Los foros, los grupos de Internet, página del fabricante... son buenos sitios para recopilar información

4.2.- Problemas de menos: Tarjetas especiales

Los dispositivos que quisiéramos instalar en el sistema requieren de líneas IRQ o canales DMA, lo que quiere decir que hay un mundo de conflictos esperando a que vengan los nuevos dispositivos. Puedes evitar muchos problemas si antes de instalar la tarjeta lees el manual, particularmente si usas tarjetas ISA. El manual detalla los IRQ o los DMA que la tarjeta requiere.

La siguiente sección describe alguno de los conflictos con los que podríamos encontrarnos al instalar algunas de las tarjetas más usadas como tarjetas de sonido, SCSI y adaptadores de red.

4.2.1.- Tarjetas de sonido

Las tarjetas de sonido son, probablemente, el recurso individual que más acapara en el sistema. Generalmente usan como mínimo 1 IRQ, 2 DMA y múltiples rangos de direcciones de puertos de E/S. El uso de tantos recursos es debido a que en realidad las tarjetas de sonido son varias piezas de hardware sobre una misma tabla. La siguiente tabla muestra los recursos usados por una tarjeta de sonido.

Default Resources Used by Creative Labs SB512

Device	IRQ	I/O Port Address	16-Bit DMA Channel	8-Bit DMA Channel
Sound Blaster 16 emulation	5	0220022F 03300331 ^[1] 0388038B ^[2]	5	1
Creative Multimedia Interface	N/A	D400D407	n/a	n/a
Creative Gameport Joystick	N/A	02000207	n/a	n/a
Creative SB512	9 ^[3]	C860C87F	n/a	n/a

La mayoría de las tarjetas de sonido incluyen modelos basados en PCI emulando a la Sound Blaster 16

Las tarjetas de sonido requieren, como ya hemos dicho, de un gran número de recursos. Si leemos la documentación de la tarjeta y determinamos que canales necesita, comparamos lo que necesitan con los IRQs y DMAs que están ya usándose, y cambiamos la configuración de los otros dispositivos para evitar conflictos, la instalación será rápida y limpia. Desafortunadamente muchos fabricantes no suministran información detallada de las tarjetas, así que no queda otra que instalarla y usar el administrador de dispositivos para grabar que configuración va mejor antes de instalar

otras tarjetas.

Para evitar movidas, lo mejor que podemos hacer es instalar antes que nada (excepto tarjetas de video) la tarjeta de sonido y cambiar la configuración de los otros dispositivos que vayamos instalando sucesivamente si diesen algún conflicto.

4.2.2.- Tarjetas SCSI

Los adaptadores de tarjeta SCSI usan más recursos que cualquier otro dispositivo que tengamos que enchufar en la placa (excepto la tarjeta de sonido). A veces las SCSI usan recursos que les hace entrar en conflicto con las tarjetas de sonido o de red. Una típica SCSI ISA, con una BIOS integrada, requiere de una línea IRQ, un canal DMA y un rango de direcciones de puertos de E/S, además de ROM y RAM. Afortunadamente, el típico adaptador SCSI es también fácil de reconfigurar, y cambiando alguna de sus configuraciones no debería afectar al rendimiento. Las SCSI basadas en PCI requieren lo mismo excepto el canal DMA.

Antes de instalar una SCSI, lee el manual de la tarjeta y asegurate, como siempre, que las IRQ, canales DMA, puertos de E/S y la memoria que necesita están disponibles. Si no es así deberás reconfigurar la SCSI o los dispositivos implicados en el conflicto.

4.2.3.- Tarjetas de red

Las redes cada vez son más populares gracias a la facilidad de configuración de redes de trabajo/casa y el uso de tarjetas de red, como cable o módems DSL, para conectarse por banda ancha a Internet. Una tarjeta de red común no necesita tantos recursos como una SCSI o las tarjetas de sonido, pero requiere como mínimo un rango de puertos de E/S y una interrupción. Estate atento de que no hay nada usando lo que necesita.

4.2.4.- Adaptadores COM

Un adaptador de puerto serial tiene como mínimo dos puertos montados. Estos puertos COM requieren de una interrupción y un rango de puertos de E/S cada uno. Con los puertos de E/S no hay mayor problema, ya que los rango de hasta 4 puertos COM están bien definidos. El problema está con las interrupciones.

Las antiguas instalaciones de más de 2 puertos serial tenían el problema que las adicionales compartían las mismas interrupciones de las 2 primeras. Esto, evidentemente, es malo y lo único que hacía era causar problemas a los software de Windows. Para estas viejas placas asegúrate que cada serial tiene un único rango de puertos de E/S y una sola interrupción.

Hoy día hay tarjetas adaptadoras de multipuertos con los que se puede llegar hasta 12 COM sin ningún tipo de conflicto. De todas formas no es necesario tener tantos, ya que tenemos el USB o la ethernet 10/100 que pueden ser usados para realizar tareas de puertos serial.

4.2.5.- USB (Universal Serial Bus)

Los puertos USB corresponden con el USB 1.1 o el USB 2.0. Un problema potencial de estos puertos es que los USB toman interrupciones del sistema (a veces más de una) y muchos ordenadores no tenían ninguna libre o estaban al límite. Si el sistema tiene PCI IRQ Steering, no va a tener mayor problema ya que el IRQ usado por el controlador USB puede ser compartido por otros dispositivos PCI. Si no tienes disponibles interrupciones y no poses el Steering, deberías mirar que dispositivo puedes quitar (COM, LPT...) para conseguir interrupciones para otros dispositivos.

La gran ventaja del USB es que los buses USB usan solo un IRQ sin importar cuantos puertos USB están instalados en el PC en un controlador USB individual. Algunos sistemas con múltiples controladores USB usan IRQs adicionales, pero los controladores USB pueden compartir IRQs entre ellos y con otros dispositivos PCI como dijimos antes.

Si no usas los USB puede desactivarlo en el setup CMOS de la placa, de modo que queden libres las IRQs que no estén usando.

4.3.- Sistemas Plug & Play

PnP representa la mayor revolución en tecnología de interfaces. PnP llega en un principio al mercado en el 95 y la mayoría de las placas base y las tarjetas ya lo incorporan a partir del 96. Antes del PnP, los usuarios de los Pcs debían manipular jumpers, switches y un infierno de cosas para instalar una tarjeta, teniendo como resultado una pila de conflictos de recursos y tarjetas que no se conseguía que funcionasen.

PnP no fue totalmente un concepto nuevo. Las llaves que abrieron esta puerta fueron las características del MCA y del EISA de hacía 10 años, pero que las limitaciones que tenían impidieron que llegasen a buen puerto. Los primeros sistemas PCI también incorporaban una forma de configuración PnP pero, al no tener el manager previsor de conflictos entre tarjetas PCI e ISA, hubieron muchos problemas de configuración para los usuarios.

Ahora mismo el PnP es de uso indiscutible: la eliminación de la preocupación de los usuarios por que no haya problemas de conflictos no tiene precio.

Para que el PnP funcione correctamente debe de estar integrado en el hardware que vayamos a usar, la BIOS y el SO.

4.3.1.- Componentes hardware

Al hablar de componentes hardware nos referimos a los sistemas informáticos y a los adaptadores de tarjeta. Los adaptadores PnP de tarjeta están comunicados con la BIOS y el SO para convenir cuales recursos del sistema son necesarios. La BIOS y el SO resuelven los conflictos (mientras puedan) e informan al adaptador que recursos puede usar. Entonces el adaptador modifica su configuración para usar los recursos especificados.

4.3.2.- Componentes de la BIOS

La necesidad que la BIOS fuera PnP significó que todas las BIOS de los Pcs de antes del 96 tuvieran que ser actualizadas o conseguir nuevas máquinas con BIOS PnP. Para que la BIOS fuera compatible debía proporcionar 13 llamadas de funciones del sistema adicionales, usadas por los componentes PnP del SO. La BIOS PnP la desarrollaron conjuntamente Intel, Compaq y Phoenix Technologies.

4.3.3.- Componentes del SO

El sistema operativo tienen la obligación de informar al usuario de posibles conflictos que no pueden ser resueltos por la BIOS. Dependiendo del SO, el usuario podrá solucionar el problema configurándolo manualmente mediante el administrador de dispositivos o abriendo el PC y cambiando switches de la tarjeta. Cuando se inicie de nuevo el sistema, revisará si hay conflictos nuevos y si han solucionado los que había.