



***Les IRQ font partie des figures de proue parmi les cauchemars des utilisateurs de PC qui font de la vidéo, débutants ou moins débutants. Daniel nous explique de quoi il s'agit au juste.***

## A. Introduction

Déjà, il faut savoir qu'il existe 3 sortes d'interruption :

- 1) Les interruptions processeur (celles générées par une erreur de calcul interne du genre "Divide by zero" par exemple).
- 2) Les interruptions "hardware" ou matérielle, et que nous allons détailler ci-après.
- 3) Les interruptions "software" ou logicielle (lorsqu'une application a besoin d'avoir accès à un morceau de code commun contenu et contrôlé par un autre morceau de code et situé dans une location mémoire très différente et qui peut varier, comme le BIOS par exemple).

## B. Les IRQ

Tout d'abord, IRQ veut dire "Interrupt ReQuest", tout simplement, c'est-à-dire Requête d'Interruption, et cette appellation est exclusivement réservée aux interruptions matérielles. Elles sont générées, comme leur nom l'indique, par des composants extérieurs au processeur, mais pas forcément extérieurs au PC lui-même. Le "System Timer" interne, par exemple, interrompt le processeur en utilisant l'IRQ 0 ; c'est un composant de la carte mère. A chaque fois que vous pressez une touche du clavier vous interrompez le processeur par l'intermédiaire de l'IRQ 1. En fait, tout périphérique connecté d'une manière ou d'une autre au processeur et ayant besoin de se faire entendre devra interrompre le processeur

## C. La table des interruptions

Chaque fois que le processeur est interrompu, il va chercher l'adresse du code à exécuter dans une table. Par exemple, le pilote logiciel fourni avec une carte d'acquisition vidéo et utilisant l'IRQ 10 devra aller mettre dans cette table, lors de son installation en mémoire, l'adresse du code qu'il souhaite voir exécuter lorsque son périphérique (la carte d'acquisition) désirera prendre la main.

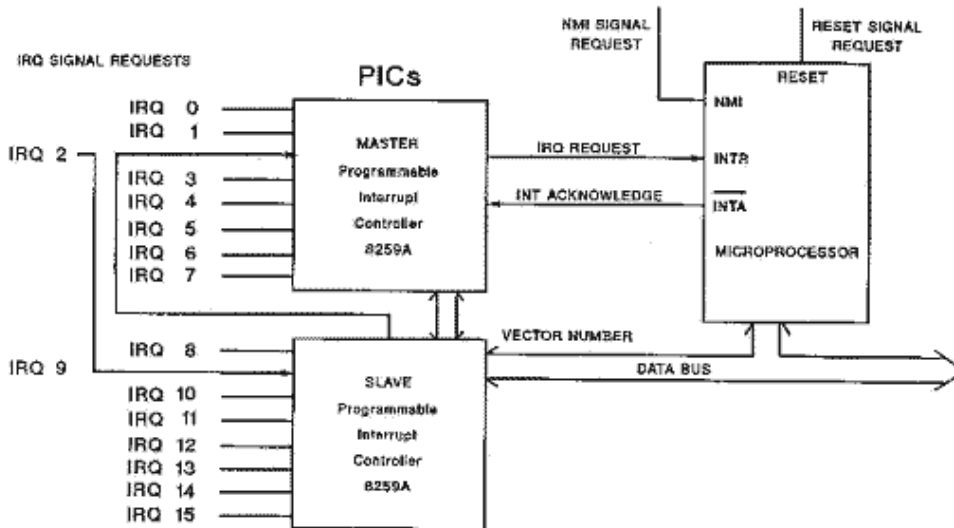
Ceci était vrai lorsque les IRQ n'étaient pas partageables. Maintenant c'est un peu plus compliqué puisqu'un morceau du système d'exploitation va gérer ce processus dynamiquement, mais nous verrons cela un peu plus tard.

## D. Le mécanisme des interruptions hardware

### a) Le PIC

Il y a quelques années nous pouvions encore physiquement trouver sur nos cartes mères 2 composants qui géraient les interruptions : les PICs ("Programmable Interrupt Controller").

Maintenant ils sont intégrés au sein de ce que l'on appelle le "Chipset" ("ensemble de composants" en français, regroupés au sein d'1 ou 2 boîtiers en tout), mais le principe reste exactement le même. Il y a 2 PICs, sachant chacun gérer 8 interruptions hardware. Le premier gère les IRQ de 0 à 7, et le second de 8 à 15. La sortie du second PIC est câblée sur l'entrée correspondant à l'IRQ 2, donc sur le premier PIC, et seule la sortie du premier PIC part réellement vers le processeur (voir figure ci-dessous).



Le plus haut niveau de priorité étant l'IRQ 0, l'ordre des priorités, en tenant compte de l'emboîtement des 2 PICs, est donc :

- ..... IRQ 0 = System Timer
- ..... IRQ 1 = Clavier
- ..... IRQ 8 = Real Time Clock (horloge temps réel)
- ..... IRQ 9 = "disponible"
- ..... IRQ 10 = "disponible"
- ..... IRQ 11 = "disponible"
- ..... IRQ 12 = Souris PS/2
- ..... IRQ 13 = Coprocesseur arithmétique
- ..... IRQ 14 = Canal IDE primaire (pour les disques durs - 2 maximum)
- ..... IRQ 15 = Canal IDE secondaire (pour les CD-ROM et graveurs - 2 maximum)
- ..... IRQ 3 = COM2 (second port série)
- ..... IRQ 4 = COM1 (premier port série)
- ..... IRQ 5 = "disponible" (souvent utilisée pour la carte son en compatibilité Sound Blaster)
- ..... IRQ 6 = Lecteur de disquettes (2 maximum)

➤..... IRQ 7 = LPT1 (le port parallèle)

Et c'est tout ! Sur un PC de base seule 4 IRQs restent disponibles, et encore, le son étant maintenant une composante obligatoire des PCs, 3 IRQs seulement sont utilisables pour tous les autres périphériques et autres accessoires.

Au cas où 2 IRQs viendraient à arriver en même temps, le contrôleur d'interruptions (PIC) ira servir la plus prioritaire en premier, et gardera la seconde sous le coude pour la servir après que la première aura fini son travail. C'est pour cela que le clavier est branché sur l'IRQ 1, de manière à ce que l'utilisateur puisse reprendre la main de façon prioritaire. Il ne faut tout de même pas oublier que tout morceau de software peut, quand il le désire et en l'occurrence avant l'exécution d'une partie critique ou "temps réel" de code, mettre un masque d'interruption, c'est-à-dire interdire au processeur de se laisser interrompre. C'est d'ailleurs dans ces moments là, et quand quelque chose se passe mal, que l'on en arrive à un PC "planté" : un morceau de code ayant interdit les interruptions est en attente d'un événement qui n'arrive pas.

On accuse souvent le partage des interruptions, processus décrit un peu plus bas, d'être la cause de lenteur dans la capture et/ou dans la restitution de la vidéo, et donc de générer des saccades. Mais en fait, qu'il y ait beaucoup plus de lignes d'IRQ ou pas, en d'autres termes que l'on ait besoin de les partager ou pas, le problème réel vient simplement du fait qu'il faut servir plus de choses que le PC n'est capable d'en absorber. Le partage lui-même n'y est généralement pas pour grand chose.

## **b) L'APIC**

Depuis peu de temps, le hardware de nos PC a évolué. Les chipsets intègrent maintenant, en plus du PIC standard, un APIC ("Advanced Programmable Interrupt Controller"). Nous reparlerons un peu plus tard des intérêts qu'apporte l'APIC en terme de gestion des interruptions. D'un pur point de vue matériel, il est présent sur toutes les machines multiprocesseurs et sur les nouvelles générations de PC compatibles Windows XP. L'APIC permet de traiter jusqu'à 256 requêtes d'interruption différentes - les 16 premières étant équivalentes aux 16 IRQ que nous venons de décrire. En réalité l'APIC est présent dans le chipset de nombreux PC modernes, au même titre que le simple PIC, par contre, il n'est pas forcément utilisable. Pour qu'il le soit, il faut que le BIOS d'un tel PC ait été écrit pour cela, c'est-à-dire qu'il ait configuré cet APIC, en plus du PIC, pour être prêt à travailler. Jusqu'à présent seuls les PC multiprocesseur étaient dans ce cas, car il s'agit alors d'une obligation pour pouvoir le faire fonctionner en multiprocesseur justement. On parle alors de BIOS "I/O APIC" ou même plus directement "IOAPIC" (I/O pour "Input" "Output" - entrée sortie). Depuis peu, Microsoft impose aux fabricants de PC de systématiser l'utilisation de l'APIC, même sur les PCs monoprocesseur. C'est une des conditions à l'obtention du logo Microsoft "Designed for Windows XP" que les constructeurs collent fièrement sur la face avant de chaque machine. Ceci sera visible, bien-sûr sous Windows XP, mais également sous Windows 2000 (voir chapitre H). Nous parlerons alors du fonctionnement de Windows 2000 en "mode PIC" ou en "mode APIC". Par contre Windows 98 ne saura pas utiliser ce nouveau contrôleur d'interruptions et se contentera, même sur ces nouvelles machines pourtant configurées pour le support de l'APIC, à travailler avec le PIC standard.

## **E. Les attributions d'IRQ**

Avant même de parler de partage d'interruption, il faut d'abord parler d'attribution. Dans l'exemple ci-dessus nous avons donc 4 IRQs disponibles. Mais en fait, seulement 4 (9, 10, 11 et 5) restent disponibles parce que les autres ont été attribuées. Cette étape a été réalisée par le BIOS tout simplement parce que le câblage physique des composants avec le PIC est fait sur la carte mère, et que le BIOS le sait ou le voit (le composant gérant le port parallèle LPT1 est physiquement connecté à l'entrée 7 du PIC par exemple, ou à travers une sorte de commutateur électronique piloté par le BIOS qui pourra alors le connecter ou non, donc lui assigner l'IRQ 7, ou la laisser libre).

Ajoutons maintenant une carte son dans notre PC. Il y a 3 cas possible :

1) Une vieille carte ISA, avec des cavaliers de configuration.

Comme nous savons que l'IRQ 5 est disponible et que beaucoup d'anciennes applications DOS s'attendent à ce que cette carte parle à travers l'IRQ 5 (compatibilité Sound Blaster), configurons la donc en IRQ 5 avec le bon choix de cavalier.

2) Une carte plus récente, mais toujours ISA, sans cavalier de configuration.

Il s'agit de la version PnP (raccourci pour "Plug-and-Play" c'est-à-dire "on branche et ça marche") de cette carte. La seule différence est que cette carte ira se configurer toute seule lors du premier démarrage du PC, et comme par hasard sur l'IRQ 5 puisqu'elle verra que cette IRQ est justement disponible. A ce niveau là, il faut savoir que les lignes d'interruptions correspondant aux IRQ disponibles sont physiquement présentes sur le bus ISA, et donc que la carte va réellement être connectée à l'IRQ 5.

3) Une carte PCI.

A propos, "PCI" sont les initiales de "Peripheral Component Interconnect" ; il s'agit d'une norme régie par un groupe d'intérêt (voir à <http://www.pcisig.com/> pour plus de détails). Revenons-en à notre carte PCI : n'y cherchez pas de cavalier, toutes les cartes PCI sont obligatoirement PnP ; mais c'est une autre sorte de PnP, beaucoup plus subtil. En effet, maintenant les fameuses lignes d'interruption ne sont plus connectées au bus PCI, contrairement aux connecteurs ISA. Elles ont été remplacées par d'autres lignes d'interruptions : les interruptions PCI. C'est finalement le BIOS ou/et l'OS qui iront au moment voulu connecter ces lignes à une entrée du PIC, donc à une vraie IRQ. En conclusion, plus de connexion permanente entre les cartes PCI et les IRQ contrairement aux cartes ISA, donc une totale flexibilité.

A ce niveau là, 2 cas de figure se présentent : soit nous allons charger un OS PnP ou non. Et oui, ça se complique !

1) Un OS (raccourci pour "Operating System", système d'exploitation en français) dit "PnP" est un OS qui, comme une carte, se débrouille tout seul pour attribuer ses ressources disponibles. Les OS PnP sont :

- ..... Windows 95 (Win95)
- ..... Windows 98 (Win98)
- ..... Windows 2000 (W2K).

2) Les OS non PnP sont

➤..... DOS

➤..... Windows NT (NT4).

Seulement pour qu'un OS PnP puisse se débrouiller tout seul, encore faut-il pouvoir dire au BIOS de ne pas attribuer les ressources lui-même.

A l'inverse, dans le cas d'un OS non-PnP, il faut impérativement dire au BIOS d'attribuer les ressources lui-même sinon elles ne pourront tout simplement pas marcher une fois l'OS chargé.

Pour cela, il y a un champ dans le SETUP des BIOS s'intitulant généralement "PnP OS" où l'on répond par oui ou par non, ou "Operating System running" où l'on répond par DOS/NT ou par Windows95/.

## **F. Le partage des interruptions**

Tout composant dit "legacy", terme technique anglo-saxon faisant référence à l'ancienne norme PC AT et que l'on pourrait avantageusement traduire par "l'héritage", ne peut en aucun cas accepter le partage des interruptions. La raison est très simple : ces composants sont physiquement connectés aux entrées du PIC comme nous l'avons déjà vu, alors que pour pouvoir partager une interruption il faut bien sûr pouvoir gérer une connexion de plusieurs lignes d'interruption intermédiaires avant d'en brancher physiquement une sur le PIC. Pour cela, seule la flexibilité qu'offre le bus PCI le permet. De fait, dans le tableau des IRQs ci-dessus, seules les IRQs 9, 10 et 11 peuvent encore être considérées comme partageables. En effet, toutes les autres IRQs sont utilisées par des composants dits "legacy", et ce, en considérant que le PC est équipé d'une carte son connectée sur le bus ISA du PC.

Les BIOS des PCs modernes autorisent maintenant aussi le partage d'interruption sur les IRQs de 3 à 7, à partir du moment où aucun composant "legacy" n'est connecté dessus. Par exemple, si vous n'utilisez aucun de vos ports série, vous pouvez alors, depuis le SETUP de certains BIOS, les déclarer comme inactifs. Ceci correspondra à une réelle déconnexion physique des ports qui seront alors rendus invisibles, même si l'OS chargé derrière est PnP, et le bénéfice direct sera la libération des IRQs 3 et 4 dans cet exemple précis.

Les périphériques PCI peuvent partager les mêmes IRQs parce qu'au démarrage du PC le BIOS construit une table appelée "PCI IRQ routing table". Cette table contient les entrées de chaque périphérique PCI qui se verra attribuer un numéro d'IRQ ISA (rappelez-vous, les numéros de 0 à 15 que l'on a décrit plus haut). Ces assignations d'IRQ ISA sont aussi liées, par construction matérielle cette fois, aux lignes physiques d'interruptions PCI associées aux différents connecteurs PCI d'un PC donné. L'ensemble des 2 se combine en ce qui s'appelle un "Link Value". Un Link Value est utilisé à chaque fois qu'une communication s'établit avec un périphérique, et bien que des périphériques puissent partager une même IRQ ils auront tous un Link Value unique: il n'y a donc pas d'erreur possible !

## **G. La gestion des interruptions sous Win95 (depuis la SR2) et sous Win98**

### **a) Définition de l'IRQ "Steering"**

Ce terme d'IRQ Steering (direction d'IRQ) désigne le processus software au sein de l'OS qui assigne dynamiquement les IRQs du bus PCI vers les périphériques en ayant besoin. Comme on a l'a déjà vu, seules les IRQ non déjà assignées à un périphérique ISA (ou legacy) pourront l'être vers un périphérique PCI, car une IRQ ne peut en aucun cas être assignée à la fois à un périphérique ISA et à un périphérique PCI. Ce moteur d'assignation des IRQ PCI (PCI bus IRQ Steering) va donc attribuer les IRQs, que le BIOS l'ait ou non déjà fait. Il est donc préférable de ne pas le laisser faire au BIOS pour éviter que Windows assigne une IRQ différente de celle attribuée par le BIOS à un périphérique donné (ceci est très rare, mais peut toutefois se produire).

## ***b) Comment Win95 (depuis la SR2) et Win98 utilisent l'IRQ Steering***

Ce "PCI bus IRQ Steering" donne à Windows la possibilité de reprogrammer les interruptions PCI tout en rééquilibrant les ressources entre les périphériques PCI et ISA PnP versus les ressources non-PnP.

Par exemple, imaginons que vous ayez déjà une carte PCI dans votre PC, pour laquelle l'IRQ 10 a été attribuée. Maintenant vous ajoutez une vieille carte ISA non-PnP configurée par cavalier sur l'IRQ 10 justement. Vous vous trouvez alors dans un cas flagrant de conflit d'IRQ. Mais pas pour longtemps, parce qu'automatiquement Windows va :

- ..... invalider le périphérique PCI utilisant l'IRQ 10,
- ..... reprogrammer une IRQ libre pour être une IRQ PCI (par exemple la 11),
- ..... assigner un "IRQ Holder" (terme défini au paragraphe ci-dessous) à l'IRQ 11,
- ..... reprogrammer l'IRQ 10 pour être une IRQ ISA (legacy),
- ..... retirer l'IRQ Holder qui existait pour l'IRQ 10.

De ce fait, la carte ISA non-PnP pourra cette fois utiliser sans problème l'IRQ 10, et inversement notre carte PCI se verra assigner l'IRQ 11 qui a donc été réservée pour cet usage.

## ***c) Les IRQ Holders***

Un "IRQ Holder for PCI Steering" est un indicateur montrant qu'une IRQ a été programmée pour servir des périphériques PCI, même si aucun périphérique ne s'en sert concrètement (une sorte de réservation en quelque sorte). Bien évidemment aucun périphérique ISA ne pourra utiliser une IRQ sur laquelle se trouve un IRQ Holder.

## ***d) Comment voir quelles sont les IRQs programmées pour le PCI***

- ..... Depuis le panneau de configuration, double-cliquer sur l'icône Système.
- ..... Cliquer sur l'onglet "Gestionnaire de périphériques".
- ..... Double-cliquer sur "Ordinateur".
- ..... S'assurer que la case "Requêtes d'interruption" est bien cochée.

➤..... La liste de toutes les interruptions est alors affichée, y compris les IRQ Holders, sous la mention de "IRQ Holder for PCI Steering" pour Win95 à partir de la SR2 ainsi que pour Win98 dans le cas de machines n'ayant pas de BIOS ACPI (APM), et sous la mention de "IRQ Holder ACPI pour PCI IRQ Steering" pour Win98 dans le cas de machines ayant un BIOS ACPI.

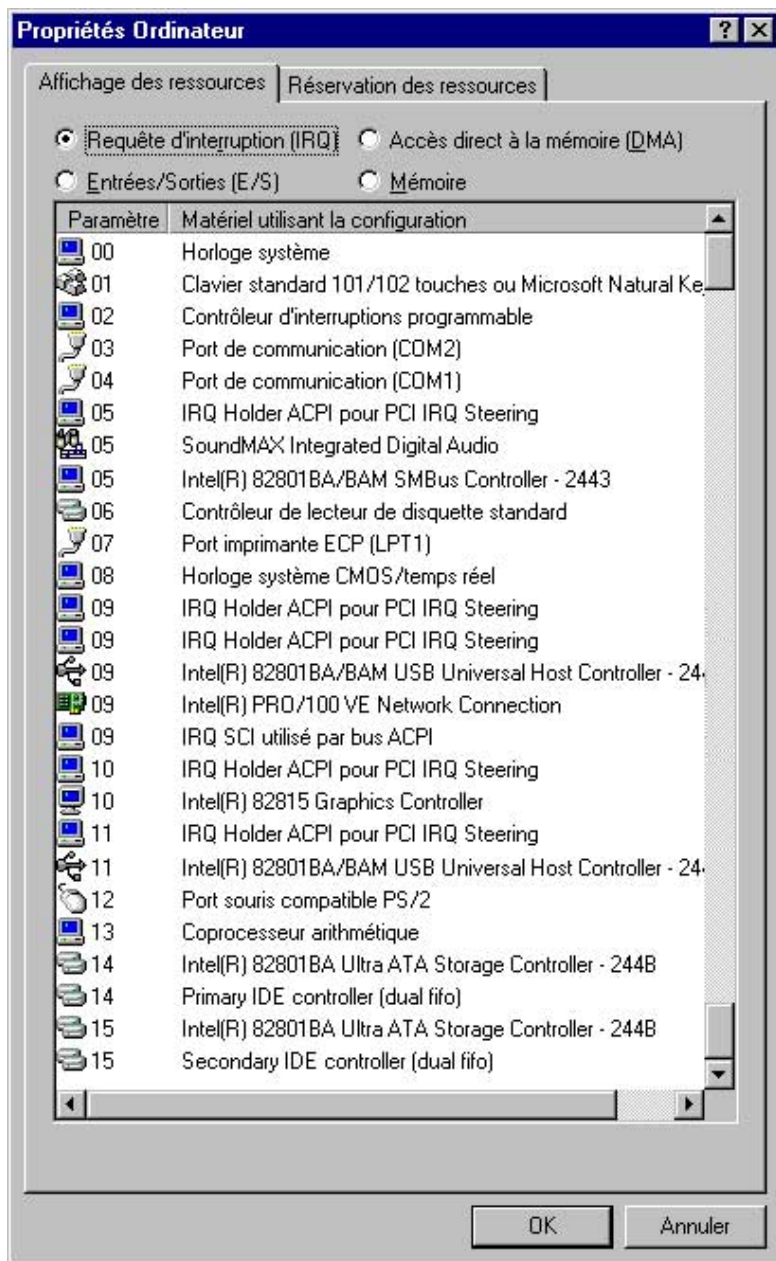
"ACPI" sont les initiales de "Advanced Configuration and Power Interface". Il s'agit d'une évolution de l'ancienne spécification "APM" qui sont les initiales de "Advanced Power Management". Il s'agit en fait de la gestion de la mise en route et de la mise en veille progressive jusqu'à l'arrêt du PC. L'énorme avantage de l'ACPI comparé à l'APM est sa capacité à gérer la mise en route, l'endormissement et l'arrêt des périphériques connectés au PC. Pour cela, il faut bien, d'une manière ou d'une autre, que la couche ACPI dans Windows gère les ressources des périphériques, en particulier les IRQs. Pourquoi ? Et bien par exemple, la norme ACPI permet à un PC éteint ou en veille profonde de se réveiller lorsque le téléphone sonne pour activer la fonction répondeur ou fax, et ce, grâce à l'IRQ assignée au modem. Voilà pourquoi la gestion des IRQs est différente entre une ancienne machine seulement APM et un PC moderne ACPI.

### ***e) Comment voir et changer l'état de l'IRQ Steering***

- ..... Depuis le panneau de configuration, double-cliquer sur l'icône Système.
- ..... Cliquer sur l'onglet "Gestionnaire de périphériques".
- ..... S'assurer que la case "Afficher les périphériques par type" est bien cochée.
- ..... Double cliquer sur "Périphériques Système".
- ..... Double cliquer sur "Bus PCI"
- ..... Cliquer sur l'onglet "Steering IRQ"

De là, vous pouvez invalider purement et simplement le fonctionnement de l'IRQ Steering, ou, plus légèrement, changer la manière dont Windows va établir ses priorités pour assigner les IRQs. Et c'est justement à partir de là qu'il faut commencer pour tenter de résoudre les problèmes de conflit d'IRQ que vous pourriez avoir sur votre PC, mais seulement si Windows n'arrive pas à le faire correctement de lui-même. C'est par exemple - cas extrême il s'entend - en invalidant l'IRQ Steering que vous pourrez enfin réussir à ce que les configurations d'IRQ que vous avez entrées vous-même dans le SETUP de votre BIOS soient vraiment celles utilisées par Windows ! ATTENTION: en fonction du BIOS de votre PC, même après avoir invalidé l'IRQ Steering, il se peut que les "IRQ Holder for PCI Steering" apparaissent encore dans le statut des IRQ. Il faut alors aussi invalider cette fonction depuis le SETUP de votre BIOS, ou plus exactement lui dire de ne pas autoriser cette fonction.

### ***f) Exemple de configuration d'IRQ sous Windows 98***



Il s'agit ici de la table des IRQ d'un PC récent dont le BIOS est APIC. Mais hélas, comme nous sommes sous Windows 98 seul le PIC standard est utilisé et donc seules 16 IRQ sont disponibles.

Il m'a fallu un peu tricher, en passant par MS-Paint, pour réussir à afficher toutes les IRQ en une seule page (cela se voit par les 2 ascenseurs)...

Cet exemple est intéressant car la configuration est assez chargée. Chose curieuse, l'IRQ 2 apparaît, ce qui est assez rare, et plutôt inutile en fait. Comme nous l'avons vu un peu plus haut, l'IRQ 2 est utilisée pour cascader le second PIC sur le premier. Elle est donc listée ici comme réservée pour le contrôleur d'interruptions programmable, c'est-à-dire le PIC. Ouf, tout va bien !

L'IRQ 5 semble partagée. Elle est utilisée par le composant audio qui est intégré dans le chipset. C'est son rôle principal. Mais elle est également utilisée par le SMBus ("System Management Bus"). C'est ce qui permet au PC de reprogrammer certaines chose en interne (du genre mémoire spéciale ou sont stocké les numéros de série, par exemple). Cette programmation ne se fait généralement pas depuis Windows, ou alors depuis certains programmes très particuliers seuls habilités à le faire, et utilisés très exceptionnellement. On ne parlera donc pas vraiment de partage dans ce cas particulier.

Et enfin nous avons un vrai partage de l'IRQ 9 entre le premier contrôleur USB (ce PC a 2 contrôleurs USB de 2 ports chacun, le second étant sur l'IRQ 11) et le LAN ("Local Area Network" - carte réseau intégrée ici dans le chipset). On y voit donc 2 "IRQ Holder ACPI pour PCI IRQ Steering", un pour chacun. L'IRQ SCI ("System Configuration Interrupt"), et aussi connecté à l'IRQ 9, sert ici à la gestion interne de l'ACPI.

Je vous encourage maintenant à poursuivre votre lecture, puis à revenir voir cet exemple après avoir lu la section sur Windows 2000 en mode APIC. En effet, j'ai utilisé exactement le même PC, sur lequel j'ai simplement réinstallé W2K. Vous verrez ainsi clairement la différence !

## **H. La gestion du partage des interruptions sous Windows 2000**

### ***a) Introduction***

Windows 2000 (W2K) a été développé avec l'objectif, parmi beaucoup d'autres bien sûr, de justement permettre le partage des IRQs pour les périphériques PCI. Tout périphérique PCI, même s'il se voit attribuer des ressources par le BIOS sera ré-analysé par W2K qui n'hésitera pas à les changer si nécessaire. C'est donc un comportement normal sous W2K que d'avoir ce partage, et ce, tout particulièrement pour les PC ACPI, ACPI qui sous Windows 2000 ne fonctionnera correctement qu'à cette condition.

### ***b) La configuration des IRQs sous W2K sur des PCs ACPI***

Sous Windows 2000 vous trouverez généralement l'IRQ 9 assignée à une grande partie des périphériques PCI, et même probablement à tous ! C'est normal, c'est comme cela que cela doit être, et on ne peut pas changer les configurations d'IRQ sous W2K.

Dans le détail des IRQs du Gestionnaire de Périphériques vous verrez apparaître la liste des IRQs ISA (de 0 à 15) avec éventuellement quelques trous pour celles non-utilisées, et en particulier en face de l'IRQ ISA 9 la mention "Microsoft ACPI-Compliant System", puis en dessous la liste des IRQ PCI, toutes portant le même numéro (9). En fait seule l'IRQ 9 est utilisable par le bus PCI pour l'IRQ Steering sous W2K. Pour la première fois dans le mode des PCs un OS gère différemment les IRQs ISA (donc non-partageables) et les IRQs PCI.

Cette caractéristique permet en toute sécurité d'ajouter d'autres périphériques sans aucun risque de conflit d'IRQs. Grâce aux Link Values définis par le BIOS et qui permet de savoir avec qui communiquer, une seule IRQ servira l'ensemble des périphériques. Ce processus permet de simplifier la gestion interne et est indispensable à la gestion du Power Management (ACPI).

### ***c) La re-programmation des IRQs sous W2K sur des PCs ACPI***

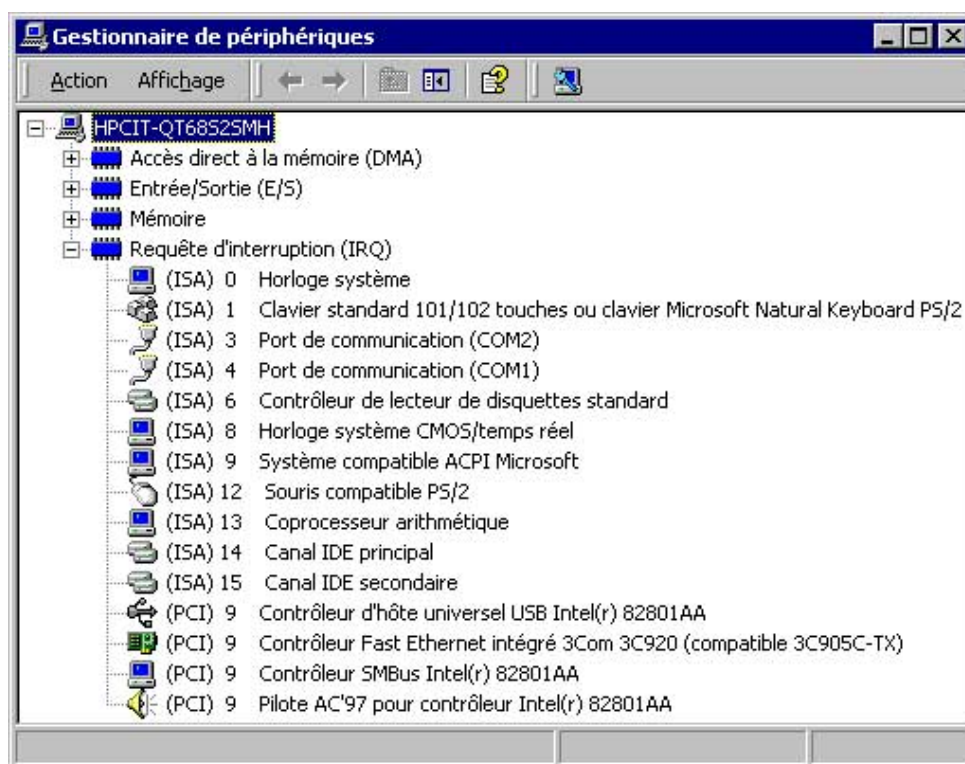
Windows 2000, au contraire de Win98, n'a pas la possibilité de reprogrammer dynamiquement les interruptions PCI pour rééquilibrer les ressources entre les périphériques PCI et ISA. Lorsque les ressources PCI sont définies, elles ne peuvent être changées. En d'autres termes, vous ne pouvez en aucun cas ajouter une vieille carte ISA configurée sur l'IRQ 9 dans un système où Windows 2000 a été installé. En revanche, vous

pouvez tout à fait ajouter une carte PCI, qui ira de fait partager avec les autres cette même IRQ 9.

La raison de cette limitation est liée à la gestion des cartes multiprocesseurs, de l'IOAPIC (caractéristiques des BIOS multiprocesseurs) et des bus PCI multi-branche (donc globalement des cartes mères beaucoup plus complexes) qu'offre W2K au contraire de Win98. Dans ces circonstances complexes la re-programmation dynamique des IRQs serait beaucoup trop risquée et cette caractéristique n'a donc pas été implémentée au sein de W2K.

Si vous êtes dans ce cas précis (PC relativement récent, donc compatible ACPI, et W2K), ne perdez pas votre temps avec les IRQs, n'essayez pas de changer les configurations de votre BIOS pour forcer telle ou telle carte PCI sur telle ou telle IRQ : W2K ne s'en occupera pas, il les laissera toutes sur l'IRQ 9, point-barre ! Mais d'autre part, ne prenez pas peur du fait que toutes vos cartes PCI partagent cette IRQ 9. Cela marchera sans problème, c'est fait pour. Et si vous avez des problèmes de performance, de saccades pendant vos captures vidéo, ne les mettez pas sur le compte du partage de l'IRQ 9, c'est ailleurs qu'il faut chercher la cause de cette mauvaise performance.

#### ***d) Exemple de configuration d'IRQ sous W2K en mode PIC***



Je n'ai pas pu utiliser ici le même PC que dans l'exemple précédant puisqu'il s'agissait d'une machine équipée d'un BIOS APIC et que cet exemple traite de W2K en mode PIC. Je parlerai donc de la simplification de l'affichage dans l'exemple suivant (section g de ce même chapitre), histoire de comparer les systèmes d'exploitations entre eux sur un même matériel.

Que voyons-nous donc d'intéressant ici ? Déjà la séparation entre les IRQ ISA et les IRQ PCI. Ensuite nous voyons l'IRQ ISA 9, qui est réservée pour l'ACPI, et donc, nous voyons de fait l'IRQ PCI 9 (c'est-à-dire la même physiquement que l'IRQ ISA 9 précédemment réservée) qui se retrouve partagée entre tous les autres périphériques PCI de ce PC (l'USB, le

LAN, l'audio et également le SMBus avec les mêmes remarques que dans l'exemple précédant à ce sujet).

### ***e) La configuration et la re-programmation des IRQ sous W2K sur des PCs non-ACPI***

Dans le cas où votre PC serait assez ancien, avec un BIOS et un hardware ne supportant pas la norme ACPI, Windows 2000 utilisera lors de son installation une HAL différente. La HAL (Hardware Abstraction Layer) est la couche de l'OS qui dialogue directement avec les composants électroniques sur la carte mère et directement avec le BIOS du PC. Normalement le programme d'installation de W2K installe la HAL ACPI, mais dans ce cas il chargera une HAL "Standard PC" puisque le BIOS lui aura répondu qu'il n'est pas ACPI.

Ceci aura 2 effets principaux :

- 1) aucune fonction de gestion de l'alimentation (Power Management) ne sera disponible,
- 2) la configuration manuelle des IRQ par le BIOS sera partiellement disponible.

Ne rêvez pas ! Ce n'est pas une solution à vos problèmes, rappelez-vous : il n'y a pas de problème d'IRQ !

D'une manière générale ce type de configuration se comportera normalement de la même manière si vous ne faites aucun réglage particulier dans le SETUP de votre BIOS et si vous laissez bien le BIOS configuré en "OS PnP". Vous aurez donc l'IRQ 9 pour tous vos périphériques PCI, et tout marchera parfaitement.

Mais si vraiment vous y tenez, alors, depuis le BIOS, forcez telle ou telle carte sur l'IRQ que vous voulez lui assigner, configurez votre BIOS en "PnP Operating System = no", et bootez W2K. Puis regardez l'effet que ces modifications auront apportées à vos attributions de ressources au sein de W2K. Ne rêvez pas non plus, vous n'arriverez jamais à faire vraiment ce que vous voulez, en tous cas jamais du premier coup.

Mais encore une fois, ce n'est pas par ce biais, sauf cas très exceptionnel, que vous pourrez résoudre vos problèmes, et en tous cas, si l'idée vous venait de vouloir remplacer la HAL ACPI, normalement chargée sur un système ACPI, par la HAL Standard PC pour gagner en flexibilité sur les attributions d'IRQ, ne dites surtout pas que l'idée vient de moi et ne comptez surtout pas sur mon support : les résultats sont imprévisibles ! La seule solution à peu près sûre pour y arriver sans risque de tout perdre est de réinstaller W2K depuis le début en pressant F5 pendant la phase d'inspection du hardware (au tout début de l'installation), et en choisissant dans la liste proposée la HAL "Standard PC" au lieu de la HAL "Advanced Configuration and Power Interface (ACPI) PC".

Pour plus de détail sur ce sujet si vous êtes anglophone :

➤..... [How to Specify a Specific or Third-Party HAL During Windows 2000 Setup](#)

Pour les plus fous :

➤..... [How to Troubleshoot Windows 2000 Hardware Abstraction Layer Issues](#)

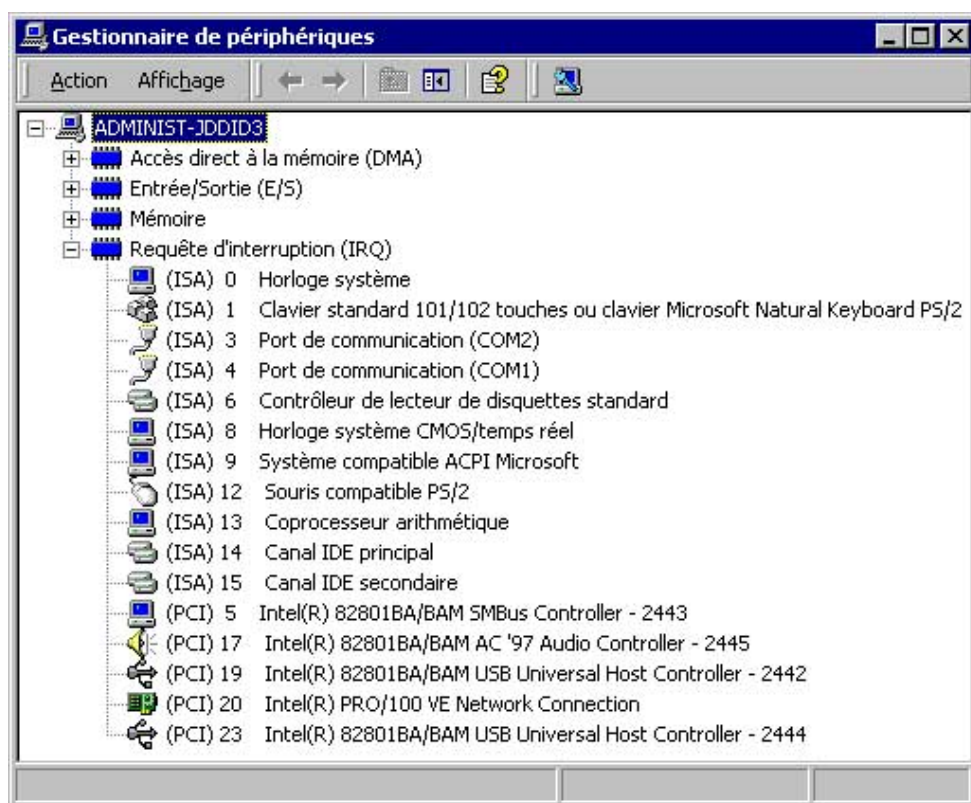
## ***f) Les IRQ sous W2K sur des PCs ACPI en mode APIC***

Comme nous en avons parlé dans le chapitre D, les PC multiprocesseurs et depuis peu certains PC monoprocesseur (et très bientôt tous les PC) ont été configurés par leur BIOS pour pouvoir utiliser l'APIC. Cet APIC est capable de gérer 256 interruptions plutôt que seulement 16. Si W2K a été installé sur une telle machine il utilisera donc l'APIC plutôt que le PIC. Le résultat sera immédiat et flagrant : plutôt que de partager les 16 IRQ existantes entre les différents périphériques, W2K va simplement assigner par ordre croissant les 256 numéros disponibles. Les 16 premiers (de 0 à 15) seront généralement réservés aux périphériques ISA, puis les chiffres au-delà de 15 pour les périphériques PCI.

Toutefois, cela ne va pas fondamentalement changer les choses... Rappelez-vous : les Link Values définis par le BIOS en mode PIC permettaient de retrouver le pilote à utiliser lors de l'arrivée d'une interruption sur une ligne partagée. Le mode APIC permet donc de légèrement simplifier ce processus en attribuant une ligne unique par ressource. Le gain de temps est en réalité infime, "epsilonesque" même...

Par contre, la tranquillité d'esprit qu'apporte cette nouvelle caractéristique des PCs d'aujourd'hui devrait clore définitivement ce très controversé débat sur le partage des interruptions.

## ***g) Exemple de configuration d'IRQ sous W2K en mode APIC***



Regarder attentivement cet exemple, et maintenant retourner rapidement faire un tour au chapitre G section f. Il s'agit exactement du même PC, qui a donc exactement les mêmes ressources et les mêmes besoins en terme d'IRQ ! Seulement voilà, nous sommes maintenant sous W2K qui lui, au contraire de Win98 dans le cas précédant, sait enfin gérer l'APIC de cette machine. Nous disposons donc maintenant de 256 IRQ au lieu des seuls 16 d'avant. Les différences sont frappantes :

- .....Déjà l'inutile IRQ 2 disparaît. Elle disparaît réellement car elle reste réservée pour le PIC qui, bien que non-utilisé dans ce cas, est toujours physiquement présent dans le chipset.
- .....L'IRQ 5 utilisée par le SMBus est listée plus loin, avec les IRQ PCI. Cela clarifie le fait qu'il s'agisse d'un composant PCI et non d'un composant ISA.
- .....L'IRQ ISA 9 est toujours utilisée par W2K pour la gestion de l'ACPI, pas de changement puisqu'il s'agit d'un composant de type ISA cette fois (notez la plus grande simplicité et clarté des libellés).
- .....L'audio intégré au chipset se retrouve maintenant en IRQ17,
- .....Le premier contrôleur USB en 19,
- .....Le LAN en 20,
- ..... Et enfin le second contrôleur USB en 23.

Etonnant, non ?

## I. Questions / Réponses entre Claude et Dan

**Claude :** Si le processeur est en train de travailler à une acquisition ou à une exportation et qu'une interruption de niveau plus élevée arrive, cela ne peut-il perturber le flux de données, à moins que le logiciel de capture ne masque les interruptions ?

**Dan :** Il faudrait pouvoir analyser le code de chaque driver et de chaque application de capture pour pouvoir répondre précisément... A vrai dire le masquage des interruptions ne peut que très rarement être utilisé pour de nombreuses raisons liées à l'OS et plus particulièrement aux fonctions multitâches de l'OS (très vaste débat particulièrement complexe). C'est seulement dans certains bouts de code très critiques que certains drivers utilisent le masquage des interruptions et pendant une période très très courte. Typiquement, on peut toujours interrompre une capture en pressant la touche ESC. Néanmoins cela ne veut pas dire pour autant que les interruptions n'aient pas été masquées à un moment critique, mais seulement suffisamment peu longtemps pour que l'utilisateur ne s'en rende pas compte. Prenons un exemple concret : imaginons qu'une mémoire tampon ait été alloué pour la capture. Elle va donc être remplie par le driver ou par un morceau de l'application de capture qui gère la communication avec le port IEEE1394. Un autre morceau de cette application de capture va vider ce tampon en remplissant le fichier AVI sur le disque dur. Puisqu'on ne peut arrêter ou ralentir le déroulement de la bande du caméscope, ce tampon risque de déborder s'il n'est pas vidé au moins aussi vite qu'il est rempli. Alors, lorsqu'il devient bien plein, disons rempli à 80% par exemple, le driver doit agir pour éviter le trop plein générateur de "lost frame". Pour cela il va alors masquer les interruptions, (et sûrement faire d'autres choses aussi, du genre "changer sa priorité" par exemple pour que toutes les ressources système disponibles soient focalisées sur le vidage de ce tampon). Quelques fractions de secondes plus tard, lorsque ce buffer sera de nouveau suffisamment vide, disons à 20% par exemple, les interruptions seront de nouveau libérées... Mais attention ceci n'est qu'une hypothèse pour aider à la compréhension du mécanisme.

**Claude :** Avec tout ce que tu as écrit ma recommandation tout azimuth serait, hors Windows 2000, de mettre la carte d'acquisition sur l'IRQ 9 et pour l'obtenir créer un profil matériel

retirant le maximum de choses de manière à ce que l'OS ne puisse que vouloir utiliser cette IRQ de niveau le plus prioritaire pour les slots PCI.

Dan : Je ne suis pas sûr que le jeu en vaille la chandelle... Déjà, le coup de l'IRQ la plus prioritaire n'a d'importance que lorsque 2 IRQs arrivent en même temps ou lorsque la seconde arrive alors que la première n'a pas encore fini d'être traitée. Si une IRQ 10 arrive en plein traitement de la 9, elle attendra que la 9 ait fini, alors que si c'est l'IRQ 1 qui arrive, le processus de traitement de la 9 sera effectivement interrompu. La manipulation que tu suggères n'a d'intérêt que si tu rencontres réellement des problèmes de capture ou d'exportation vidéo, mais autrement, un bon conseil, si ton PC ne marche pas trop mal, en d'autres termes si tes captures et tes exportations vidéo se déroulent correctement, alors, ne touche à rien ! Il est bien évident qu'il faut mettre toutes les chances de son côté pour réussir une capture et une exportation, en ne faisant rien faire d'autre au PC en même temps. S'il est assez facile de penser à ne pas jouer à Doom en même temps, il peut être plus difficile de désactiver la fonction "Répondeur téléphonique" de son modem juste avant de lancer une capture. Dans ce cas, effectivement Claude, ta solution de créer plusieurs profils utilisateur est très intéressante, car elle permet assez simplement d'invalider tel ou tel périphérique d'un profil, en éliminant par exemple le modem pour une utilisation purement "Vidéo" de son PC.

Claude : Et un point dont tu ne parles pas c'est la manière dont la position physique d'un slot influe sur l'attribution de l'IRQ, par exemple où placer une carte modem PCI pour que son IRQ soit d'un rang moins prioritaire.

Dan : Cela dépend du HW du PC concerné et de la capacité de son BIOS à bien gérer ce HW. Normalement, un "super PC" devrait avoir la possibilité de connecter n'importe quelle ligne d'interruption PCI sur n'importe quelle IRQ ISA. Hélas, ce n'est pas toujours le cas, et de fait certains connecteurs peuvent se retrouver limités. Cela n'a plus beaucoup d'importance avec W2K maintenant, il suffit de pouvoir toutes les connecter sur la 9, et ceci est d'ailleurs imposé par Microsoft pour avoir le droit au logo W2K. Donc la réponse à cette question est dans la documentation technique de chaque PC...

Claude : Et comment se passe l'attribution des IRQs pour les slots AGP ?

Dan : De la même manière et en même temps. Le slot AGP (je dis bien "le", car il est unique dans un PC) est en fait un second bus PCI, juste plus rapide, et dédié pour une seule carte, la carte graphique. Cette carte graphique n'utilise une interruption que dans un mode très particulier (et très rare) ou elle doit synchroniser l'application qui la pilote, en fait pour que cette application sache quand le balayage arrive en fin de ligne et en bas de l'image. Ce n'est quasiment plus jamais utilisé, sauf pour de très anciennes applications. Oublie ! Jamais d'interruption de la part de la carte graphique. D'ailleurs, l'interruption utilisée par la carte graphique est quasiment la seule que Win98 prend le risque de partager avec une autre (n'oublions pas que W2K est le seul aujourd'hui à vraiment bien gérer le partage des interruptions).

Claude : Si j'ai bien compris, l'OS est capable de gérer les demandes d'interruptions de matériels reliés à la même IRQ ; la rapidité à laquelle il décide, fait que c'est totalement transparent pour les délais d'exécution, non ?

Dan : Déjà, les interruptions ne sont reliées à la même IRQ que lorsque l'OS décide de les y relier, et ce presque exclusivement dans le cas de W2K comme nous l'avons vu. Pour ce qui est du délai d'exécution, il s'agit en effet que de quelques lignes de code assembleur à

exécuter, donc un délai normalement inférieur au millionième de seconde sur une machine actuelle !

Claude : Par exemple si une carte SCSI partage la même IRQ que la carte d'acquisition IEEE1394, les demandes peut-être répétées de la carte pour gérer les disques durs ne peuvent-elles créer des soucis de rapidité / continuité d'un travail de rendu / transfert de fichier DV effectué en simultané ?

Dan : Il y a autre chose dont nous n'avons pas parlé, c'est le DMA Bus Mastering !!! Certaines cartes comme les cartes SCSI ou les cartes IEEE1394 sont généralement des cartes dites "Bus Master", c'est-à-dire qu'elles sont capables de prendre le contrôle du bus PCI, et de fait elles ont un accès direct à la mémoire et aux autres cartes ! Typiquement, une carte Studio DV est capable d'envoyer son flux vidéo directement à la carte SCSI ou au port IDE (s'il est configuré en DMA) sans même passer par le processeur. Si une IRQ arrive à ce moment là, le processeur n'aura rien à interrompre réellement puisqu'il ne faisait qu'attendre que quelqu'un daigne vouloir l'occuper. Aucun risque dans ce cas de gêner la communication entre le 1394 et le SCSI qui continuera... sauf si le processeur a besoin à son tour du bus PCI ou, pire encore, des disques SCSI. Mais le DMA c'est un autre débat, bien plus complexe encore.

**Daniel Millet (juin 2001- révisé juillet 2001)**