



使用 IRQ9。实现了新老兼容。

所以说，谁都不能分配 IRQ2。一般也不要手动分配 IRQ9。如果真的特别需要分配 IRQ9，一般也要把它分配给原使用 IRQ2 的外设。比如：MPU-401 MDI 混音器等。

## IRQ 编号

从图 1 中可以看到，IRQ 是有编号的。但是一定要明白，IRQ 编号并不是 8259A 中断控制器的接脚编号，也不是与接脚连接的中断线的编号。IRQ 编号主要是供中断请求程序使用的编号。当某条中断线、某个接脚供某个编号的 IRQ 使用时，习惯上可以用该编号的 IRQ 来命名该中断线、接脚。

除了按 IRQ 编号来实现不同中断请求的操作外，IRQ 编号还有个表示优先级的功能。

## IRQ 的常规分配

理论上讲，电脑中的每个外设都应单独使用一个 IRQ。如果两个或多个外设使用同一个 IRQ，而且同时发出中断请求，冲突就产生了（就好像是先救妈还是先救老婆的问题）。CPU 就无法判断应该与哪一个外设进行“对话”，造成外设无法使用。

常见的 IRQ 冲突现象有，系统不能正确检测出新设备、有些硬件工作不正常（如声卡不发声），严重的会出现死机。

为了最大限度地避免冲突，为了避免各自分配 IRQ 引起的麻烦。大家约定俗成地按照常规约定，来默认分配 IRQ。比如：在所有的电脑中，所有的操作系统中都会自动把 IRQ0 默认分配给系统计时器使用。

许多 IRQ 一般都是按照常规约定，由操作系统、可自动检测 IRQ 的软件自动分配的。有些是不可改动的，有些是可以改动的。见图 1 和下表 1。

表 1

| IRQ 编号 | 默认分配                      | 分配原则  |
|--------|---------------------------|---|
| IRQ0   | Time (系统计时器)              | 不能更改。   |
| IRQ1   | Keyboard (键盘)             | 不能更改。   |
| IRQ2   | Redirect IRQ9 (重定向)       | 不能更改。   |
| IRQ3   | COM2、COM4 (串行插座)          | 内置调制解调器 (首选) 等串行外设。                                   |
| IRQ4   | COM1、COM3 (串行插座)          | 串行鼠标 (首选) 等串行外设。                                      |
| IRQ5   | LPT2 (并行插座)               | 并口打印机 (首选) 或声卡等外设。因为许多电脑都没有 LPT2 插座，所以 IRQ5 一般都分配给声卡。 |
| IRQ6   | FDD (软盘驱动器插座)             |   |
| IRQ7   | LPT1 (并行插座)               | 并口打印机 (首选) 或声卡等外设。                                    |
| IRQ8   | CMOS Alert (即时时钟)         | 不能更改。   |
| IRQ9   | Redirect IRQ2 (重定向)       | 不建议分配。<br>特殊情况下，可分配给 MPU-401 MDI (混音器)。               |
| IRQ10  | 未分配                       | 网卡 (首选) 等外设。  |
| IRQ11  | 未分配                       | SCSI 控制器 (首选) 或 AGP 显卡。                               |
| IRQ12  | PS/2 Mouse (鼠标插座)         | PS/2 鼠标 (首选) 等外设。                                     |
| IRQ13  | 数学协处理器，如：FPU              | 不能更改。   |
| IRQ14  | Primary IDE (主 IDE 插座)    | 主 IDE 硬盘 (首选) 等 IDE 外设。                               |
| IRQ15  | Secondary IDE (第二 IDE 插座) | 第二 IDE 硬盘 (首选) 等 IDE 外设。                              |

从表 1 中可以看出，相当一部分 IRQ 是分配给外设插座的。外设是插入相关插座，通过插座与中断控制器相连的。

因为有些插座只能插接一种外设，所以它的 IRQ 常被直接称为某个外设的 IRQ。比如：键盘插座（IRQ1）软盘驱动器插座（IRQ6）等。

而有些插座可以插接不同的外设，所以其 IRQ 可以被任何插入的外设使用。比如：在 Secondary IDE 插座（IRQ15）中可插接第二硬盘。如果没有第二硬盘，插接的是光驱等设备，则光驱就使用 IRQ15。

### IRQ 的优先级

15 个 IRQ，最起码可以对应 15 个外设。那么，如果有两个外设同时提出中断请求，谁会优先使用 CPU 呢？如果只有一个中断控制器，自然是 IRQ0 最打么，IRQ7 排老末。现在的问题是，有两个级联的中断控制器，结果是 IRQ8—IRQ15 借了 IRQ2 的光，加塞排在了 IRQ3 前面了。下面的表 2 就是常规的优先级。

表 2

| 优先级 | IRQ 编号 | 默认分配                     |
|-----|--------|--------------------------|
| 1   | IRQ0   | Time（系统计时器）              |
| 2   | IRQ1   | Keyboard（键盘）             |
| 3   | IRQ8   | CMOS Alert（即时时钟）         |
| 4   | IRQ9   | Redirect IRQ2（重定向）       |
| 5   | IRQ10  | 未分配                      |
| 6   | IRQ11  | IRQ2                     |
| 7   | IRQ12  | PS/2 Mouse（鼠标插座）         |
| 8   | IRQ13  | 数学协处理器，如：FPU             |
| 9   | IRQ14  | Primary IDE（主 IDE 插座）    |
| 10  | IRQ15  | Secondary IDE（第二 IDE 插座） |
| 11  | IRQ3   | COM2、COM4（串行插座）          |
| 12  | IRQ4   | COM1、COM3（串行插座）          |
| 13  | IRQ5   | LPT2（并行插座）               |
| 14  | IRQ6   | FDD（软盘驱动器插座）             |
| 15  | IRQ7   | LPT1（并行插座）               |

如果一个外设可以选择不同的 IRQ，可以选择优先级靠前的 IRQ。

比如：IRQ14、15 都可以供 IDE 设备使用。在不使用 IDE 硬盘的情况下，可以供光驱等 IDE 设备使用。当然使用主 IDE 插座（IRQ14）的外设肯定比第二 IDE 插座上的外设优先使用 CPU。

### IRQ 的共享

由于有限的 IRQ 有时很难满足外设的需要，所以现实中经常会存在两个或多个外围设备使用同一个 IRQ 的共享现象。

这种共享的主要分配原则就是，共享的外设不可能同时发出中断请求。

每个主板都有自己的共享设置。比如下表 3。

表 3

| 插座       | 首选外设        | 共享 IRQ | 说明  |
|----------|-------------|--------|---|
| PIC 插槽 1 | 显卡          | IRQ11  | PIC 显卡和 AGP 显卡一般不可能同时使用。                                  |
| AGP 显卡插槽 | AGP 显卡      | IRQ11  |   |
| PIC 插槽 3 | 声卡（非集成）     | IRQ5   | 插入式（非集成）声卡与板载（集成）声卡一般不可能同时使用。                             |
| PIC 插槽 5 | 声卡（非集成）     | IRQ5   |   |
| 板载（集成）   | 声卡          | IRQ5   |   |
| LPT2 插座  | 打印机，或声卡等。   | IRQ5   | 如果使用其它外设，该外设与声卡同时提出中断请求的几率应该很小。如打印机等。                     |
| PIC 插槽 4 | 网卡、内置 Modem | IRQ10  | 如果使用其它外设，该外设与网卡同时提出中断请求的几率应该很小。                           |
| PIC 插槽 6 | 网卡、内置 Modem | IRQ10  |   |
| PIC 插槽 2 | 电视卡         |        | 独立使用  |
| COM2 插座  | 外置 Modem    | IRQ3   | 内置 Modem 和外置 Modem 一般不可能同时使用。如果使用其它外设，两外设同时提出中断请求的几率应该很小。 |
| COM4 插座  | 内置 Modem    | IRQ3   |   |
| COM1 插座  | 串行鼠标        | IRQ4   | 如果使用其它外设，两外设同时提出中断请求的几率应该很小。                              |
| COM3 插座  |             | IRQ4   |   |



## 避免 IRQ 冲突

方法 1、插座选择

主要用来解决中断冲突造成的死机和较难排除的 IRQ 冲突。

在主板 BIOS 默认的 IRQ 资源分配下，通过调换外设插入的插座来避免冲突。

方法 2、软件调整

在主板 BIOS 中，设置某外设（插座）的 IRQ。

## 从 PCI 到 APIC

APIC（Advanced Programmable Interrupt Controller）翻成咱国语叫“高级可编程中断控制器”。

其实，PIC 在很早以前就被淘汰了，取而代之的是 APIC。由于 APIC 可以兼容 PIC，所以在很多单 CPU 电脑上，我们看到的实际是 APIC 的兼容 PIC 模式。

APIC 主要应用于多 CPU 电脑，是为了解决 IRQ 太少和处理器间中断而产生的。

一个 APIC 最多可以提供 24 个 IRQ。所以，当发现电脑有 IRQ15 以上时，说明电脑使用了 APIC。当您发现电脑只有 IRQ16 以下时，电脑并不一定没使用过 APIC，可能处于 APIC 的兼容 PIC 模式。

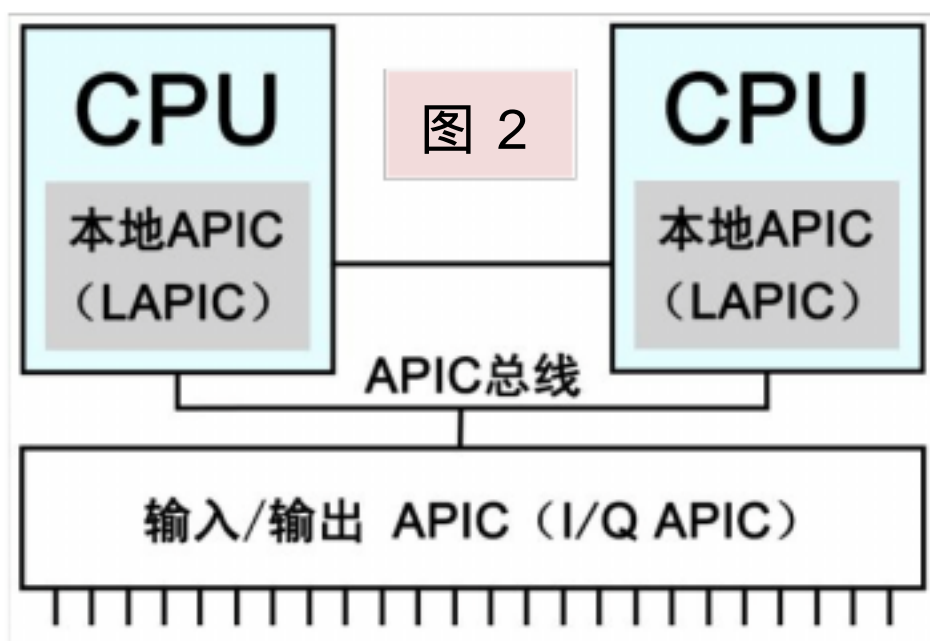
当然，通过使用多个 I/O APIC（最多 8 个），可以使 IRQ 达到上百个。

## APIC 的组成

APIC 主要由两部分组成，一个叫 LAPIC（Local APIC，本地高级中断控制器），一个叫 I/O APIC（输入/输出高级中断控制器）。早期还有 APIS 串行总线控制器，后来被放到系统总线中了。

LAPIC 位于 CPU 中。I/O APIC 位于主板南桥芯片中。

在一个典型的多 CPU 电脑中，通常有一个 I/O APIC 和多个 LAPIC，它们相互配合，形成一个中断的分发网络。典型的情况如右图 2。



LAPIC 能够智能的决定是否接受总线上传递过来的中断信息，还可以处理本地端中断。CPU 通过彼此的 LAPIC 发送中断，在某种程度上彼此进行控制。

I/O APIC，主要是接收外设的中断信号，并翻译成消息，选择发给接收中断请求的 CPU 的 LAPIC。

## APIC 的进步

与被淘汰的 PIC 相比，APIC 主要有几大进步：

进步 1、增大了 IRQ 数量。

进步 2、分隔使用 IRQ。

只有 PCI 设备才能使用 IRQ16~IRQ23，ISA 设备仍然使用 IRQ0~IRQ15。PCI 外设不需要去与 ISA 外设抢 IRQ 了。

进步 3、独特的优先级控制。

中断优先级不是由连接在 I/O APIC 哪个管脚上来决定的。I/O APIC 会使用内部的 PRT（可编程重定向表）把外设的中断请求变为中断请求信息，发送给一个或多个 CPU 的 LAPIC。每个 I/O APIC 都对应一个重定向表项（RTE），每一项都有可以被单独编程以指明中断向量和优先级、目标处理器以及选择处理器的方式。

## APIC 的启用

要使用 APIC 功能，必须选用 Windows 2000、XP=NT 4.0 等操作系统（不适用 Windows 98 SE 或 Windows ME）。当然，必须先启用（Enable）主板 BIOS 中的 APIC 选项，然后重新安装操作系统，APIC 模式才能真正发挥作用（新外设才能被正确识别）。

在安装完操作系统后，如果更改 APIC 选项，会导致操作系统无法正确启动或运行。

在主板的 BIOS 的电源管理类设置中，可以看到一个名为 APIC Function（高级可编程中断控制器功能）的选项，可以将该项设为开启（Enable）或关闭（Disable）。

在单 CPU 电脑中，当设为关闭（ Disable ）时，使用通常的 PCI（ ACPI ）模式。在 Windows 操作系统的设备管理器的 Computer（ 电脑 ）项中，会显示为 Advanced Configuration and Power Interface（ ACPI ） PC。当设为开启（ Enable ）时，则会显示为 ACPI Uniprocessor（ 单处理器 ） PC。

### 结束语

相信看到这儿，中断请求这玩意儿可以明白个大概了。不过要手动设置，还需要了解一些复杂点的东西。不是本文要深入的内容了。

就