Hardware Interrupts (IRQ) and Conflicts

The Concept of Interrupts

Because the processor cannot simultaneously process several pieces of information (it processes one piece of information at a time), a program being run can, thanks to an interrupt request, be momentarily suspended while an interrupt takes place. The interrupted program can then continue running. There are 256 different interrupt addresses.

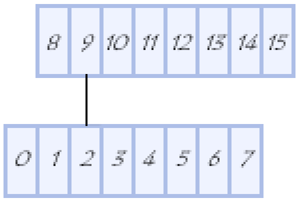
An interrupt becomes a hardware interrupt when it is requested by one of the computer's hardware components. There are many peripherals in a computer. These peripherals generally need to use the system resources if only to communicate with the system itself.

When a peripheral wants to access a resource, it sends an interrupt request to the processor in order to get its attention. The peripherals have an interrupt number that is called an **IRQ** (*Interruption ReQuest*. It is as if each peripheral pulls a "string" that is attached to a bell in order to tell the computer that it wants the computer to pay attention to it.

This "string" is in fact a physical line that links each expansion slot as well as each I/O interface to the motherboard. For an 8-bit ISA slot, for example, there are 8 IRQ lines that link the 8-bit ISA slots to the motherboard (IRQ0 to IRQ7). These IRQs are controlled by an "**interrupt controller**" that is in charge of allowing the

IRQ with the greatest priority "to speak".

When 16-bit slots were introduced, IRQs 8 to 15 were added, as was a second interrupt controller. The two groups of interrupts are linked by IRQ 2 which is connected (or "**cascaded**") to IRQ 9. In a way, this cascade "inserts" IRQs 8 to 15 between IRQs 1 and 3:



Given that priority goes from lowest to highest IRQ, and IRQs 8 to 15 are inserted between IRQs 1 and 3, the order of priority is as follows:

0 > 1 > 8 > 9 > 10 > 11 > 12 > 13 > 14 > 15 > 3 > 4 > 5 > 6 > 7

DMA

The peripherals regularly need to "borrow memory" from the system in order to use it as a **buffer** zone, i.e. a temporary storage area that allows I/O data to be quickly saved.

Thus, a direct memory access channel, called a **DMA** (*Direct Memory Access*was defined as a solution to this.

The DMA channel indicates an access to one of the computer's random access memory (RAM) slots, located by a "RAM Start Address" and an "end address". This method allows a peripheral to borrow special channels that give it direct access to the memory, without the intervention of the microprocessor, in order to unload these tasks.

A PC has 8 DMA channels. The first four DMA channels have an 8-bit bandwidth while DMAs 4 to 7 have a 16-bit bandwidth.

The DMA channels are generally assigned as follows:

- DMA0 - free
- DMA1 - (sound card)/ free
- DMA2 - floppy disk controller

- DMA3 – parallel port (printer port)
- DMA4 – direct memory access (DMA) controller

(connected to DMA0)

- DMA1 – (sound card)/ free
- DMA6 – (SCSI)/ free
- DMA7 – available

Base Addresses

Sometimes peripherals need to exchange information with the system, which is why memory addresses were assigned to them for the sending and receiving of data. These addresses are called "**base addresses**" (the following terms are also sometimes used: "*input/output ports*", "*I/O ports*", "*I/O addresses*", "*I/O port addresses*", or "*base ports*").

It is by using this base address that the peripheral can communicate with theoperating system. Therefore, there is only one unique base address for each peripheral.

Here is a list of some common base addresses:

- 060h – keyboard
- 170h/376h – secondary IDE controller
- 1F0h/3F6h – primary IDE controller
- 220h – sound card
- 300h – network card
- 330h – SCSI adapter card
- 3F2h – disk drive controller
- 3F8h – COM1
- 2F8h – COM2
- 3E8h – COM3
- 2E8h – COM4
- 378h – LPT1

- 278h - LPT2

However, all of these elements are user-transparent, i.e. users do not have to worry about them.

Hardware Conflicts

An interrupt is a line that links the peripheral to the processor. An interrupt is a hardware interrupt when it is requested by one of the PC's hardware components. For example, this is the case when a key is touched and the keyboard wants to get the processor's attention for this event. However, all 256 interrupts cannot be requested as hardware interrupts and different peripherals always make very specific interrupts.

Thus, when expansion boards are installed, you must make sure during configuration that the same interrupt is not used for two different peripherals. If this were to happen, a "**hardware conflict**" would occur and neither peripheral would function.

Indeed, if two peripherals use the same interrupt, the system will not know how to distinguish between them. A hardware conflict does not only occur when two peripherals have the same hardware. A conflict can also occur when two peripherals have the same I/O address or use the same DMA channels.

IRQ Configuration

The IRQ of an expansion board can be modified in order to assign it an IRQ number that is not being used

by another peripheral.

- On older peripherals, this IRQ number is attached to jumpers that are on the board.
- On recent boards (that have a BIOS Plug & Play), resource (IRQ, DMA, I/O addresses) parametering is automatic. It can also be carried out by the OS or with the help of utilities provided with the expansion board. The *plug & play*mode must sometimes be deactivated in order to be able to modify the parameters manually.

It is still not easy to find available resources for all peripherals. Here then is a non-exhaustive list of resources that are generally used, which therefore cannot be assigned manually:

IRQ	Peripheral
0	Internal Clock
1	keyboard
2	programmable interrupt controller Cascade to IRQs 8 to 15
3	COM2/COM4 communications port
4	COM1/COM3 communications port
5	**free**
6	floppy disk controller
7	LPT1 printer port
8	CMOS (Real-time clock)
9	**free**
10	**free**
11	**free**
12	PS2 mouse port/**free**
13	numeric data processor (math coprocessor)
14	primary hard drive controller (IDE)
15	secondary hard drive controller (IDE)

⚠	The COM1 and COM4 ports as well as the COM2 and COM3 ports use the same interrupts. This may seem illogical in that the same interrupt cannot be used by two peripherals. In reality, it is possible to use the COM1 port as well as the

	COM4 port (as well as the COM2 port and the COM3 port) so long as the are not active at the same time. Otherwise, the computer might freeze or function abnormally.

Resolving Hardware Conflicts

If you have a hardware problem, first try to isolate the problem in order to determine which peripheral is causing the problem. This means that you must attempt to eliminate as many variables as possible until you discover which element is responsible:

- by opening the computer casing and removing one by one the elements that might have caused the conflict
- by deactivating the software in the OS in order to deactivate the peripherals

Source: http://en.kioskea.net/contents/394-hardware-interrupts-irq-and-conflicts