

**IT**

**Essentials I:  
PC Hardware  
and Software**



# **IRQ, DMA, I/O Addresses**

IT Essentials Chapter 1

# System Resources

---

- In the realm of PC configuration, the word *system resources* refers to the mechanisms used to interface, communicate, and control individual device adapters and controllers, along with the I/O ports.
- All the PC's devices, including the motherboard, expansion cards, and peripherals, must share system resources.

# System Resources

---

- As a computer technician, you need to know:
  - Which Interrupt Requests (IRQ) are used with which devices
  - Which devices are assigned to which Input/Output (I/O) addresses and
  - The common uses of Direct Memory Access (DMA) channels
- In other words, you must know the relationships of the system resources and the physical and logical devices to which they are assigned.

# Interrupt Request (IRQ)

---

- An interrupt is a signal from one part of the computer to the processor indicating that a service or special action be taken that only the CPU can perform.
- When a device needs the CPU to perform a task, transfer data from memory, issue an I/O, etc., it signals the CPU using its IRQ line.
- Each device is assigned a specific IRQ number so that the processor knows the device to which it needs to respond.

# Interrupt Request (IRQ)

---

- When you add a new device to a PC, you sometimes needed to set its IRQ number by setting a DIP switch. This specifies which interrupt line the device may use.
- IRQ conflicts used to be a common problem when adding expansion boards, but the Plug-and-Play specification has removed this headache in most cases.
- A signal informing a program that an event has occurred. When a program receives an interrupt signal, it takes a specified action (which can be to ignore the signal).

# Interrupt Request (IRQ)

---

- Interrupt requests are sent to a special system component, called an *interrupt controller*.
- The interrupt controller receives and verifies requests and passes them on to the processor.
- Two interrupt controllers have been on PCs since the 286 class computer.
  - Each controller manages eight IRQ lines with each IRQ tied directly to a particular device.
  - The two interrupt controllers are linked (cascaded), through IRQ 2.

# Interrupt Request (IRQ)

---

- Interrupt signals can cause a program to suspend itself temporarily to service the interrupt.
- Interrupt signals can come from a variety of sources. For example, every keystroke generates an interrupt signal. Interrupts can also be generated by other devices, such as a printer, to indicate that some event has occurred.
- These are called *hardware interrupts*.
- Interrupt signals initiated by programs are called *software interrupts*. A software interrupt is also called a *trap* or an *exception*.

# Interrupt Request (IRQ)

---

- PCs support 256 types of software interrupts and 15 hardware interrupts.
- Each type of software interrupt is associated with an *interrupt handler* -- a routine that takes control when the interrupt occurs. For example, when you press a key on your keyboard, this triggers a specific interrupt handler.
- The complete list of interrupts and associated interrupt handlers is stored in a table called the *interrupt vector table*, which resides in the first 1K of addressable memory.

# Interrupt Request Conflicts

---

- If two devices were assigned the same IRQ, the processor could become confused and send its response to the wrong device at possibly the wrong time.
- Assigning two devices to the same IRQ creates an *IRQ conflict*.
- An IRQ conflict can cause both devices to perform sporadically (in the best case), or not to work at all (in the worst case).
- Similar devices can share IRQs, but they can't be used at the same time.

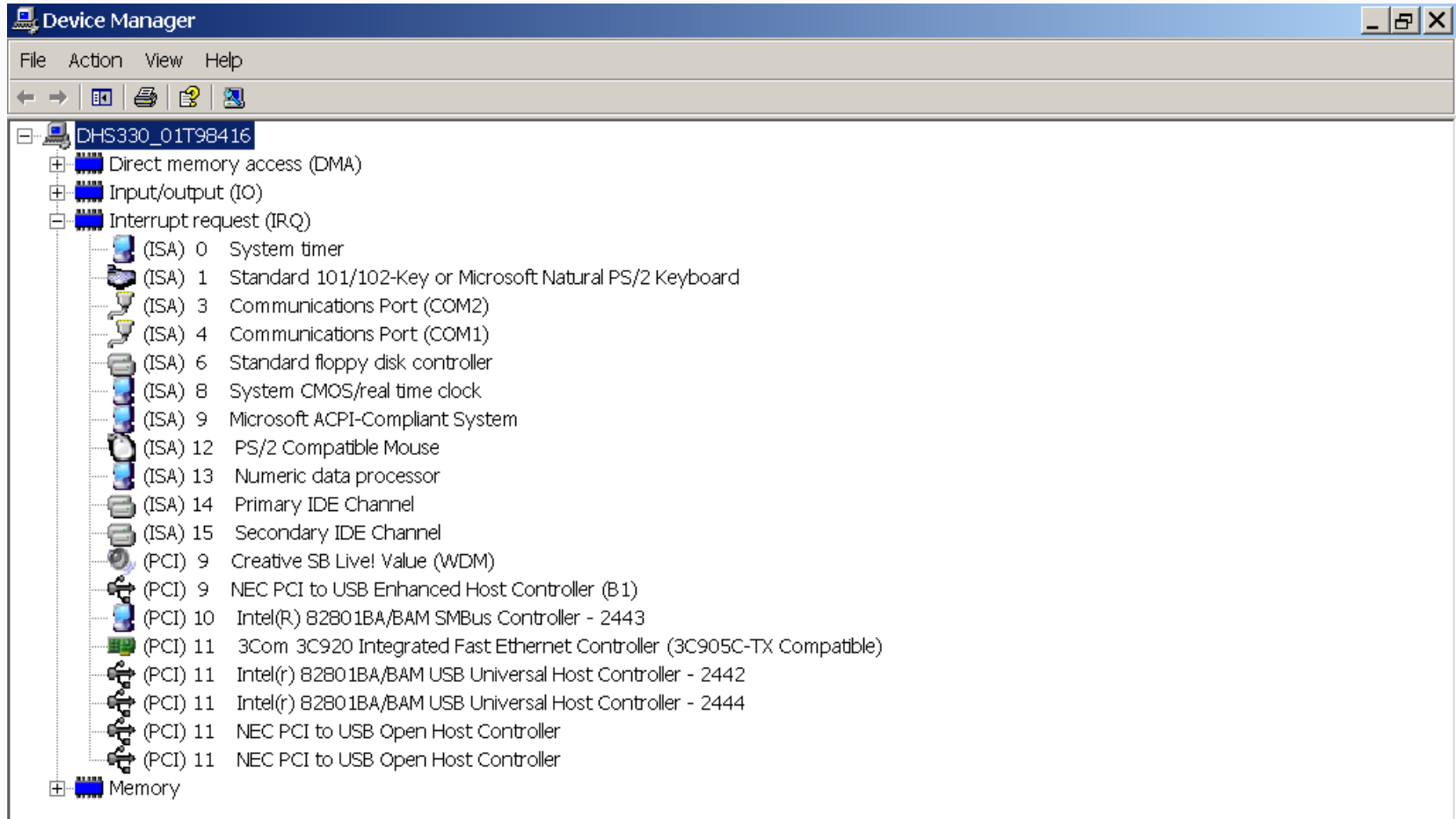
# Interrupt Request Conflicts

---

- IRQs are assigned by the system BIOS during POST and the boot process.
- Reassigning an IRQ or changing the assigned IRQ of a device is done differently depending on the adapter card and perhaps the operating system.
  - On most DOS (including Win 3x) machines, the IRQ of a device is set by either a jumper on the adapter card or through the use of proprietary installation software.
  - On Win 9x and NT PCs, the IRQ can be changed through the Device Manager.

# Interrupt Request (IRQ)

## Device Manager Screen Shot (IRQ)



# Assigning IRQs

IRQ	Typical Use	Description
IRQ 0	System timer	This interrupt is reserved for the internal system timer. It is never available to peripherals or other devices.
IRQ 1	Keyboard	This interrupt is reserved for the keyboard controller. Even on devices without a keyboard, this interrupt is exclusively for keyboard input.
IRQ 2	Cascade interrupt for IRQs 8-15	This interrupt cascades the second interrupt controller to the first.

# Assigning IRQs

IRQ	Typical Use	Description
IRQ 3	Second serial port (COM2)	The interrupt for the second serial port and often the default interrupt for the fourth serial port (COM4).
IRQ 4	First serial port (COM1)	This interrupt is normally used for the first serial port. On devices that do not use a PS/2 mouse, this interrupt is almost always used by the serial mouse. This is also the default interrupt for the third serial port (COM3).
IRQ 5	Parallel Port 2 (LPT2) Sound card	This interrupt is the first choice that most sound cards make when looking for an IRQ setting.

# Assigning IRQs

IRQ	Typical Use	Description
IRQ 6	This interrupt is reserved for the floppy disk controller.	This interrupt is reserved for the floppy disk controller.
IRQ 7	Parallel port 1 (LPT1)	This interrupt is normally reserved for the use of the printer. If a printer is not being used, this interrupt can be used for other devices that use parallel ports.
IRQ 8	Real-time clock	This interrupt is reserved for the system's real-time clock timer and can not be used for any other purpose.

# Assigning IRQs

IRQ	Typical Use	Description
IRQ 9	Cascades to IRQ 2	This interrupt is typically left open on devices for the use of peripherals.
IRQ 10	Open interrupt	This interrupt is typically left open on devices for the use of peripherals.
IRQ 11	Open interrupt	This interrupt is typically left open on devices for the use of peripherals.
IRQ 12	PS/2 mouse	This interrupt is reserved for the PS/2 mouse on machines that use one. If a PS/2 mouse is not used, the interrupt can be used for other peripherals, such as network card.

# Assigning IRQs

IRQ	Typical Use	Description
IRQ 13	Floating point unit/coprocessor	This interrupt is reserved for the integrated floating point unit. It is never available to peripherals or other devices as it is used exclusively for internal signaling.
IRQ 14	Primary IDE channel	This interrupt is reserved for use by the primary IDE controller. On systems that do not use IDE devices, the IRQ can be used for another purpose.
IRQ 15	Secondary IDE channel	This interrupt is reserved for use by the secondary IDE controller.

# Input/Output Addresses

---

- You can think of computer memory as an array of storage boxes, each of which is one byte in length. Each box has an address (a unique number) assigned to it. By specifying a memory address, the computer know where to put and where to find a particular byte of data.
- The computer can manipulate only data that is in main memory. Therefore, every program you execute and every file you access must be copied from a storage device into main memory and assigned a specific block of memory.

# Input/Output Addresses

---

- Every device in the PC uses *input/output addresses* (called I/O address or port addresses).
- The address in the I/O address points to the location in memory that's assigned to a specific device to use for exchanging information between itself and the rest of the PC.
- The I/O address is like a device's internal post office box number.
- The "Blue Screen of Death" happens when two programs try to access the same memory block.

# Input/Output Addresses

---

- All devices have an I/O address assigned to them along with a segment of memory to hold messages and data.
- The size of the memory segment varies with the amount of data a device needs.
  - In general it ranges from 1 to 32 bytes – 4, 8, or 16 is common.
- These areas of memory allow a device to do its work without worrying about what other devices or the processor may be doing

# Input/Output Addresses

---

- I/O example
  - When a modem receives data, it wants to pass the data along to the PC for processing – where can the data be put?
  - The modem writes the data to the I/O address of the COM port to which the modem is attached.
  - When the CPU is ready to process this data, it knows right where to look.
  - This process is called **memory-mapped I/O**.

# Input/Output Addresses

---

- I/O addresses are expressed in hexadecimal (i.e. 3F8h or 0x3F8). The lower case h indicates it's a hexadecimal address.
- When working with I/O addresses, it's not important that you can determine the size of the memory segment assigned or even if you can decipher the hex address itself.
- Just remember these addresses are in hexadecimal and ignore the *h* or *0x*.
- For A+ testing, memorize only the starting addresses of devices that are common to all PCs; such as keyboard, LPT1, and COM1 etc.

# Common I/O Address Assignments

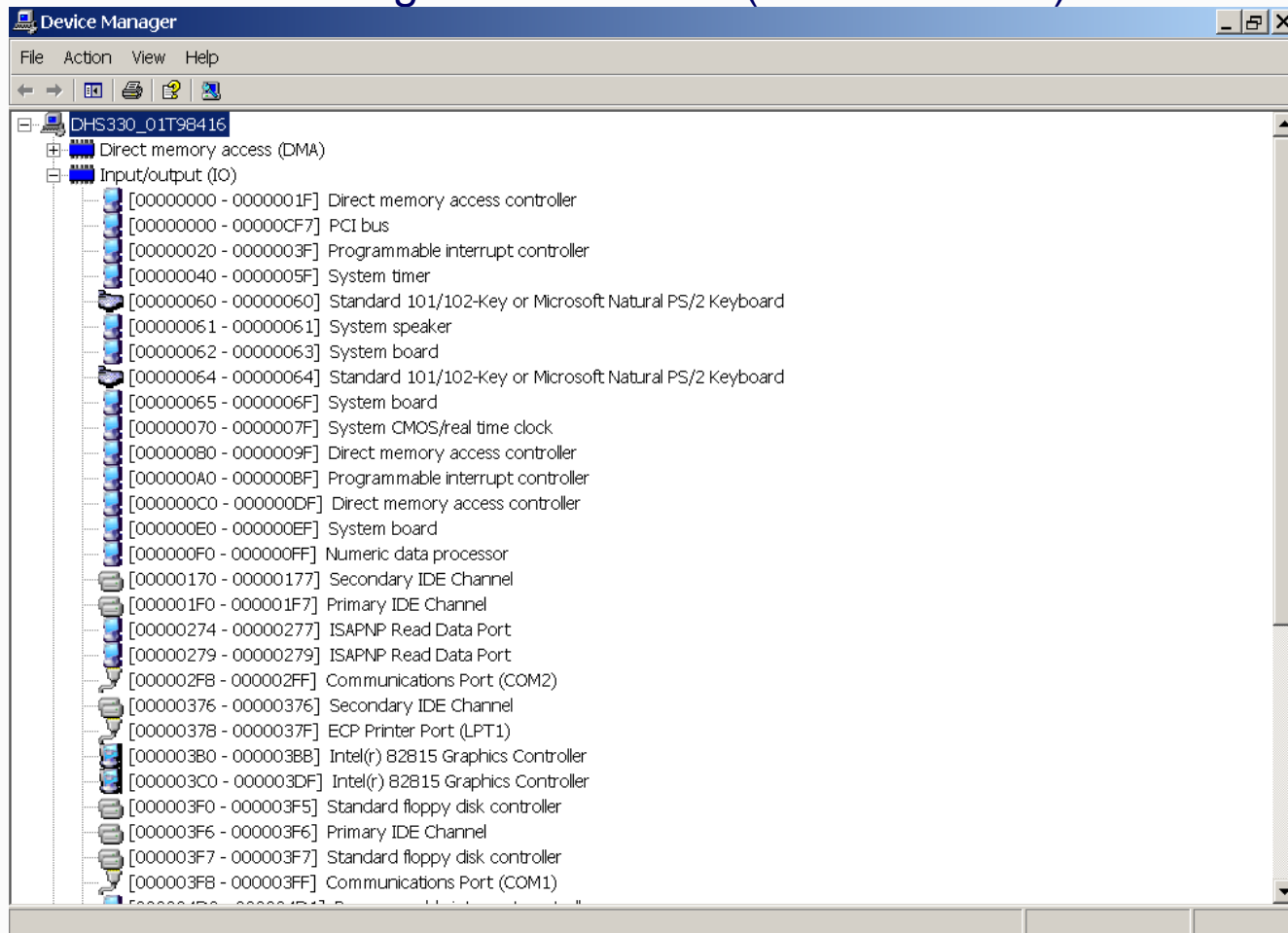
I/O Port Address (in hex)	Typical Device or Port Assignment
1F0-1F7	Primary hard disk controller
200-207	Game port joystick
220-22F	Sound card
294-297	PCI bus (data communication line)
278-27F	LPT2 or LPT3
2E8-2EF	Com4 serial port
2F8-2FF	Com2 serial port
376-376	PCI IDE controller
378-37F	LPT1 printer port

# Common I/O Address Assignments

I/O Port Address (in hex)	Typical Device or Port Assignment
→ 3E8-3EF	COM3 serial port
3F0-3F7	Floppy disk controller
3F6-3F6	PCI primary IDE controller
→ 3F8-3EF	COM1 serial port
E000-E01F	USB host controller
E800-E87F	Fast Ethernet Adapter
F000-F00F	IDE controllers

# Common I/O Address Assignments

## Device Manager Screen Shot (I/O Addresses)



# Accessing Memory Directly with DMA

---

- A *direct memory access* (DMA) channel allows a device to bypass the processor to directly access memory.
- Those devices with a DMA channel assignment gain the advantage of faster data transfers that do not have to pass through the CPU.
- DMA use is common in some disk drives, tape backup drives, and sound cards.
- One drawback to using DMA is that while the DMA device is working faster; the CPU may be put on hold, slowing everything else until the DMA data transfer is complete.

# Accessing Memory Directly with DMA

---

- Channels 0 through 3 are generally available in all PCs.
- Channels 4 through 7 are available only in 286 and later systems.
- Each group of four channels has its own DMA controller chip that services its requests.
- DMA channel assignments are made through the BIOS setup utility:
  - with the dedicated configuration software of a device, or
  - with DIP switches or jumpers on the device adapter card

# DMA Channel Assignments

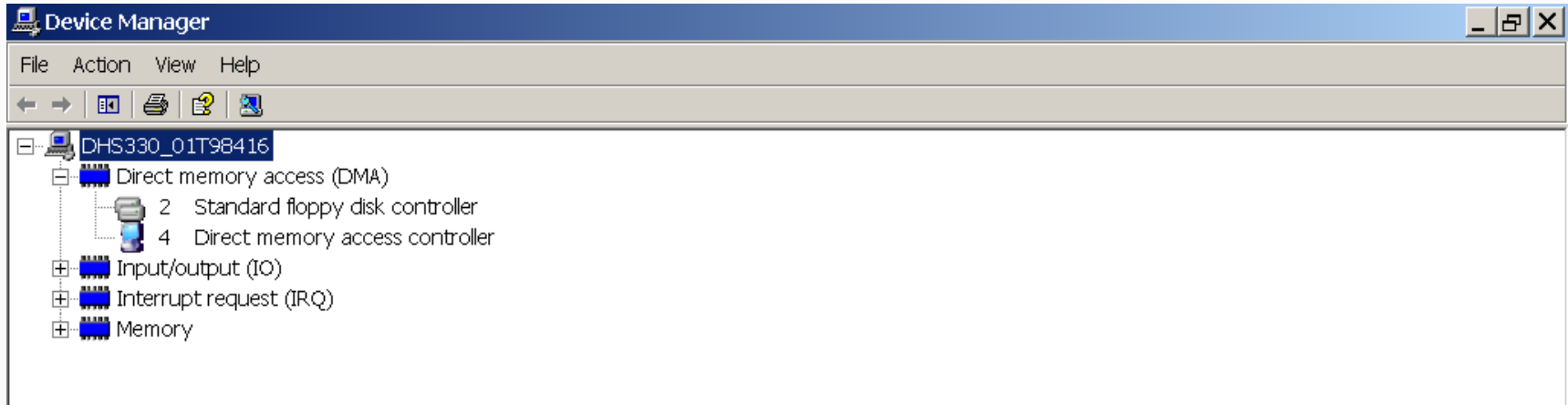
DMA	Default Device	Can be used for:
0	Memory Refresh	
1	Sound Card (low DMA setting)	Network cards, SCSI adapters, parallel printing ports and voice modems
2	Floppy Drive Controller	
3	Available	Network cards, SCSI adapters, parallel printing ports, voice modems, and sound cards

# DMA Channel Assignments

DMA	Default Device	Can be used for:
4	Cascades for DMA 0-3	
5	Sound Card (high DMA setting)	Network cards, SCSI adaptors
6	Available	Network cards, sound cards (high DMA setting)
7	Available	Network cards, sound cards (high DMA setting)

# DMA Channel Assignments

## Device Manager Screen Shot (DMA Channels)



# Naming the Logical Devices

---

- Many devices are assigned both a physical address and a logical name.
- Logical names are assigned to:
  - Serial ports (COM1 to COM4)
  - Parallel ports (LPT1 and LPT2)
- Logical names eliminate the need for software to use what could be the moving target of I/O addresses.
- Logical names are assigned during POST by the system BIOS.
- The BIOS searches the I/O addresses for devices in a preset order and assigns them a logical name in numerical order each time the system boots.

# Logical Device Name Assignments

Port	I/O Address	Default IRQ
COM1		
COM2		
COM3		
COM4		
LPT1		
LPT2		

You can remember the I/O address easier if you notice the pattern. Listing all the COM ports in order then LPT ports in order.

# Logical Device Name Assignments

Port	I/O Address	Default IRQ
COM1	3	
COM2	2	
COM3	3	
COM4	2	
LPT1	3	
LPT2	2	

You can remember the I/O address easier if you notice the pattern. Listing all the COM ports in order then LPT ports in order. **Starting from top to bottom the pattern for the first digit of the address is 3 2 3 2 3 2.**

# Logical Device Name Assignments

Port	I/O Address	Default IRQ
COM1	3F	
COM2	2F	
COM3	3E	
COM4	2E	
LPT1	37	
LPT2	27	

You can remember the I/O address easier if you notice the pattern. Listing all the COM ports in order then LPT ports in order. Starting from top to bottom the pattern for the first digit of the address is 3 2 3 2 3 2. **Starting from top to bottom the pattern for the second digit of the address is F F E E 7 7.**

# Logical Device Name Assignments

Port	I/O Address	Default IRQ
COM1	3F8	
COM2	2F8	
COM3	3E8	
COM4	2E8	
LPT1	378	
LPT2	278	

You can remember the I/O address easier if you notice the pattern. Listing all the COM ports in order then LPT ports in order. Starting from top to bottom the pattern for the first digit of the address is 3 2 3 2 3 2. Starting from top to bottom the pattern for the second digit of the address is F F E E 7 7. **The third digit in all of the addresses is 8.**

# Logical Device Name Assignments

Port	I/O Address	Default IRQ
COM1	3F8-3FFh	4
COM2	2F8-2FFh	3
COM3	3E8-3EFh	4
COM4	2E8-2EFh	3
LPT1	378-37Fh	7
LPT2	278-27Fh	5

The beginning address is all you really need to remember for A+. You might also see a pattern in the default IRQ assignment for each port. If you can **remember 4 3 4 3 for the COM ports** you will only have to know that **IRQ 7 is LPT1** and **IRQ 5 is LPT2**