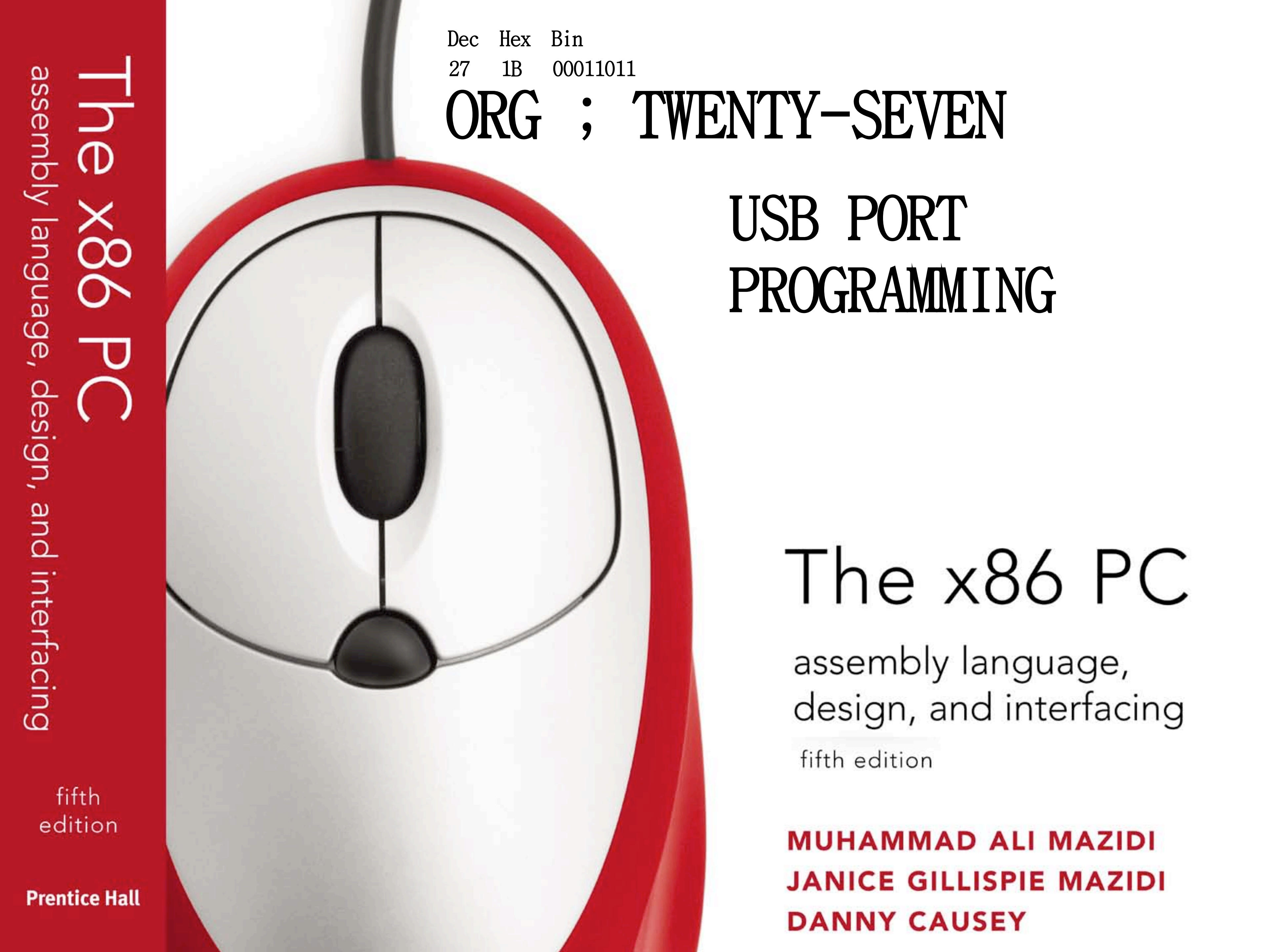
Dec Hex Bin
27 1B 00011011

ORG ; TWENTY-SEVEN

USB PORT PROGRAMMING



OBJECTIVES

this chapter enables the student to:

- Compare USB performance with PCI.
- Compare USB performance with LPT and serial COM ports.
- Understand the difference between the host and peripheral devices.
- Define the terms upstream and downstream.
- Describe the role of hubs in USB port expansion.
- Understand the difference between A-type and B-type connectors.
- Describe the USB cable signals.

OBJECTIVES (*cont*)

this chapter enables the student to:

- Describe the differences between bus and self-powered hubs in USB.
- Understand the current needs of peripheral devices and hosts.
- Program USB devices in the Windows operating system using C#.

27.1: USB PORTS: AN OVERVIEW

- USB, the *universal serial bus* is, next to PCI, one of the most important additions to the PC system.
- While ISA & PCI buses provide a high rate of data transfer between the CPU and the outside world, they have some major limitations:
  - Both ISA and PCI are *inside* the PC & their expansion slots take too much physical space on the motherboard.
  - ISA and PCI buses require too much power.
    - For every ISA expansion slot, an extra 25 watts.
    - For every PCI slot, an extra 10 watts.
  - The most important disadvantage of serial & parallel ports is the limit of 4 of each in a given motherboard.

27.1: USB PORTS: AN OVERVIEW
major features of USB

- Some of the most important features of the USB:
  - A single USB port can accommodate up to 127 devices such as a mouse, scanner, printer, modem, and so on.
    - Daisy chained together with the help of external hubs.
    - The devices are recognized automatically by the PC.
    - No opening of the PC case to connect additional devices.
  - Data transfer rates for USB 1.1 are 1.5 & 12 megabits per second (Mb/s), to a USB 2.0 maximum limit of 480 Mbs.
  - USB is hot-pluggable—new devices can be connected to the USB port without first turning off the PC.
    - Not the case with ISA and PCI.

27.1: USB PORTS: AN OVERVIEW
major features of USB

- More important USB features:
  - USB does not have to burden the system's power supply.
    - Each new device requires no more than a maximum of 500 mW from the PC power supply.
  - USB hubs outside the PC can have their own power supply.
    - USB is also equipped with power managing capability.
- Starting with Windows 98, USB device management is part of the operating system.
  - The same is true for Windows NT/2000, XP, and Vista.

27.1: USB PORTS: AN OVERVIEW
bus comparison

- A comparison of ISA, PCI, COM, LPT, and USB.
  - In the calculation of data transfer rate (bus bandwidth) for ISA & PCI buses, a 2-clock read/write cycle is assumed.
  - 2-microsecond timing for the parallel port is assumed.

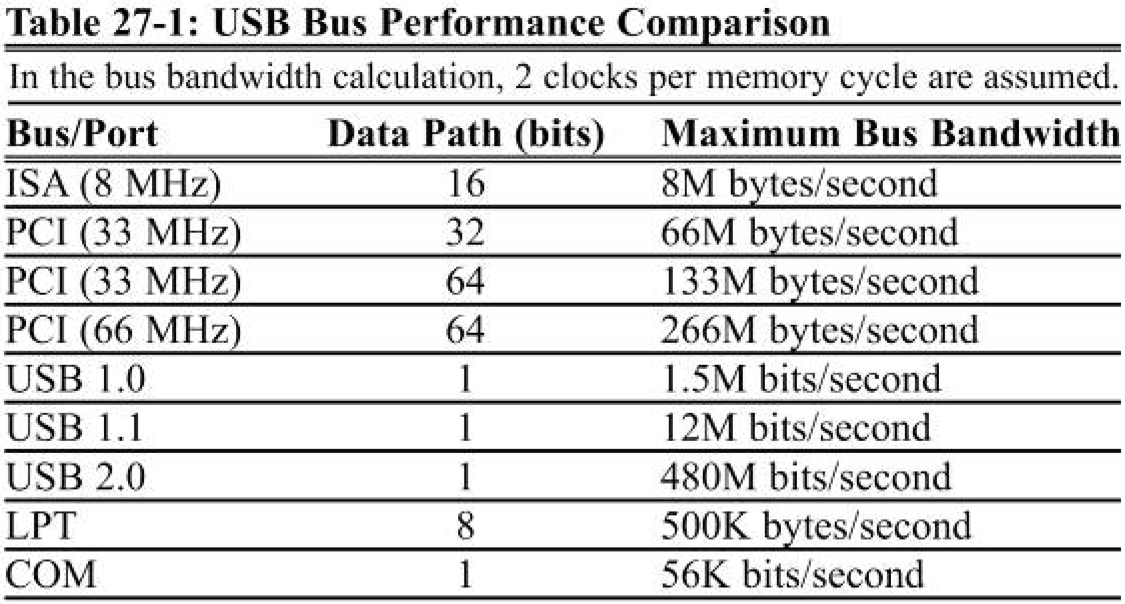


Table 27-1: USB Bus Performance Comparison

In the bus bandwidth calculation, 2 clocks per memory cycle are assumed.

Bus/Port	Data Path (bits)	Maximum Bus Bandwidth
ISA (8 MHz)	16	8M bytes/second
PCI (33 MHz)	32	66M bytes/second
PCI (33 MHz)	64	133M bytes/second
PCI (66 MHz)	64	266M bytes/second
USB 1.0	1	1.5M bits/second
USB 1.1	1	12M bits/second
USB 2.0	1	480M bits/second
LPT	8	500K bytes/second
COM	1	56K bits/second

Microsoft & Intel have worked to eliminate ISA bus, LPT, & COM ports from PC motherboards.

Many of today’s desktop PCs have PCI & USB ports only.

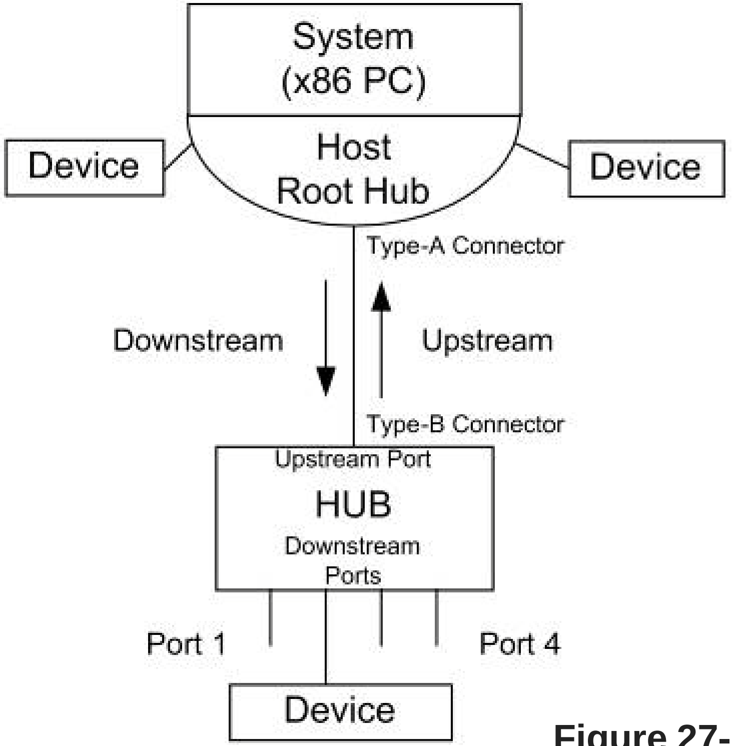
27.2: USB PORT EXPANSION/POWER MANAGEMENT

host, peripherals, and hubs

- In the USB, the *host* is the master & the peripheral I/O device is the slave.
  - In any given system there is only one host (master), which controls many peripheral I/O devices (slaves).
  - Data going toward the host is called *upstream* and data flowing from the host toward peripherals is referred to as *downstream*.
- Many x86 PC motherboards come with one USB host to be connected to peripheral I/O devices.
  - Since the board has only one host & needs to connect to many peripheral I/O devices, a *hub* is needed.

27.2: USB PORT EXPANSION/POWER MANAGEMENT
host, peripherals, and hubs

- In boards with two or more USB ports, the hub is incorporated into the host—on the x86 board it is commonly referred to as the *root hub*.



The root hub on an x86 motherboard has from two to six ports.

Each port can be connected to a peripheral I/O device or another hub.

To attach more peripheral I/O devices, an external hub is used to expand the number of USB ports.

Figure 27-1 Host and Root Hub in x86 PC

27.2: USB PORT EXPANSION/POWER MANAGEMENT
host, peripherals, and hubs

- A USB hub has one upstream port & several downstream ports, with a special USB cable to connect external hubs together.
  - The cable has **A-type** and **B-type** connectors.

The **A-type** connector is connected to a USB port.

The **B-type** is connected to the upstream hub port.

This is to prevent mixing upstream & downstream connections.

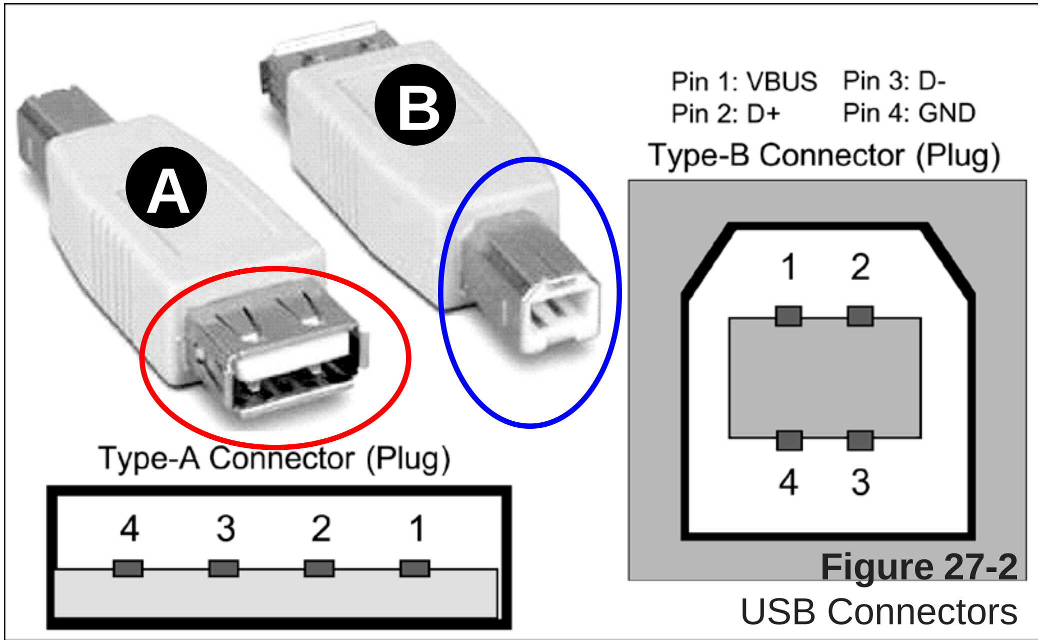


Figure 27-2
USB Connectors

27.2: USB PORT EXPANSION/POWER MANAGEMENT
host, peripherals, and hubs

- The majority of the USB cables in everyday use are USB *extension* cables, with **A-type** connectors on both ends.

Used for connecting a peripheral I/O devices such as Flash memory sticks to the USB port of the x86 PC.

Such a cable has a male (plug) **A-type** connector on one end & a female (receptacle) **A-type** connector on the other.

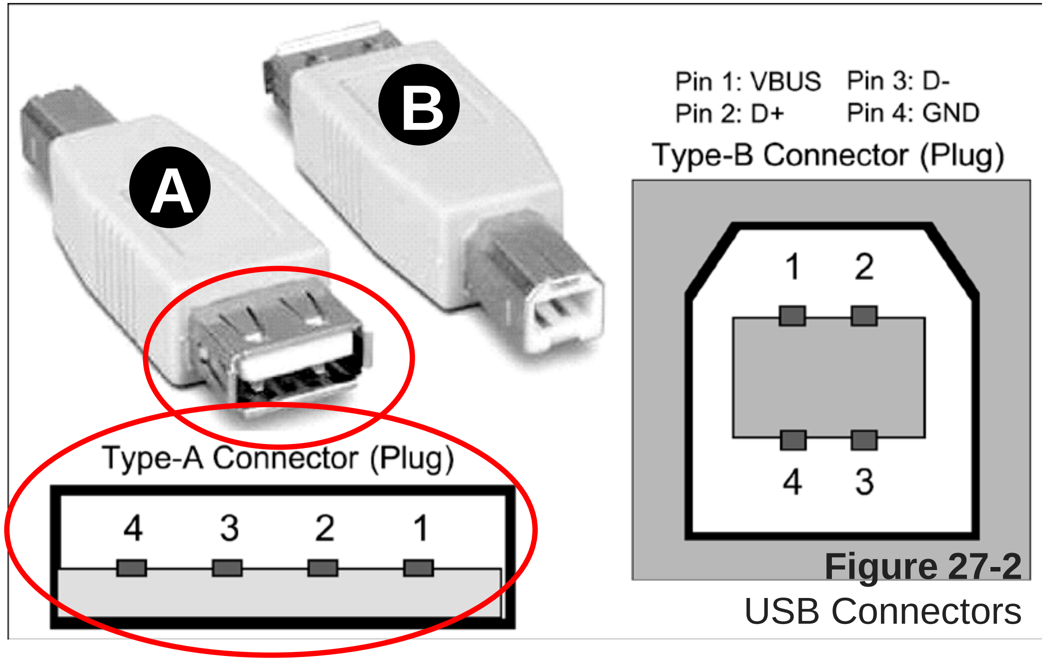


Figure 27-2
USB Connectors

27.2: USB PORT EXPANSION/POWER MANAGEMENT
expanding USB ports with external hubs

- There are two types of external hubs:
  - **Bus-powered** - power source comes from the root hub of the host PC via a USB cable.
  - **Self-powered** - the hub has its own power source.
    - It is also referred to as as an *externally* powered hub.

27.2: USB PORT EXPANSION/POWER MANAGEMENT

bus-powered hubs

- A majority of peripheral devices in everyday use, are bus-powered, such as Flash memory sticks.
  - Which means that the USB cable provides power.
- Maximum current usage by downstream peripheral devices is 100 mA.
  - A root hub with 6 USB ports uses no more than 600 mA current from the PC power supply.

27.2: USB PORT EXPANSION/POWER MANAGEMENT
bus-powered hubs

- Connecting the bus-powered external hubs increases the number of USB ports.
  - The host can provide a maximum of 500 mA of current (5V power source) per A-connector.
  - As each downstream peripheral port must be given 100 mA, the number of downstream peripheral ports in the bus-powered hub is limited to 4.
    - 4 x 100 mA + 100 mA = 500 mA

27.2: USB PORT EXPANSION/POWER MANAGEMENT self-powered hubs

- The self-powered external hub takes power from a source other than the motherboard.
  - Generally from the wall outlet.

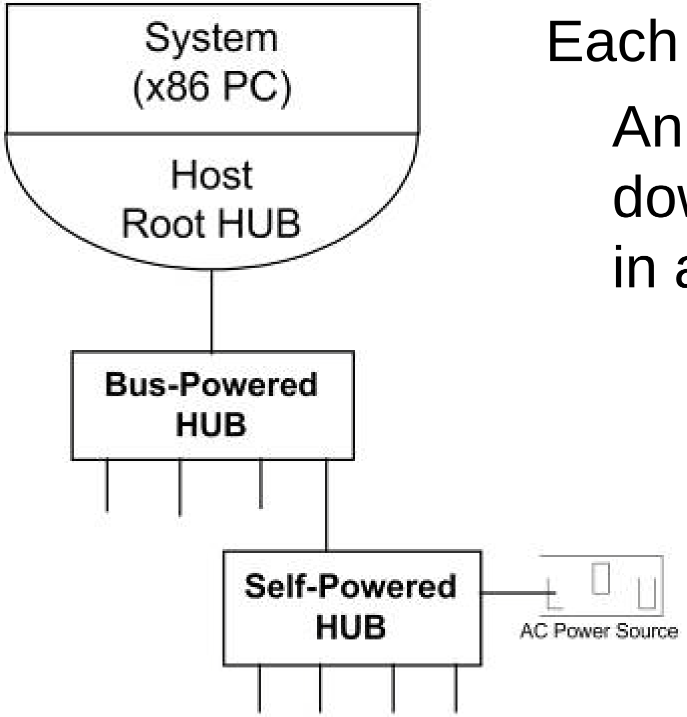


Figure 27-3 Bus-Powered and Self-Powered Hubs

Each peripheral port is provided 500 mA.

An external, self-powered hub with four downstream ports uses at least 2000 mA in addition its own internal power needs.

Printers with USB ports often have self-powered hubs, drawing from the printer power supply.

Devices with I/O functions *and* hub are often called compound devices.

27.2: USB PORT EXPANSION/POWER MANAGEMENT daisy chaining the hubs

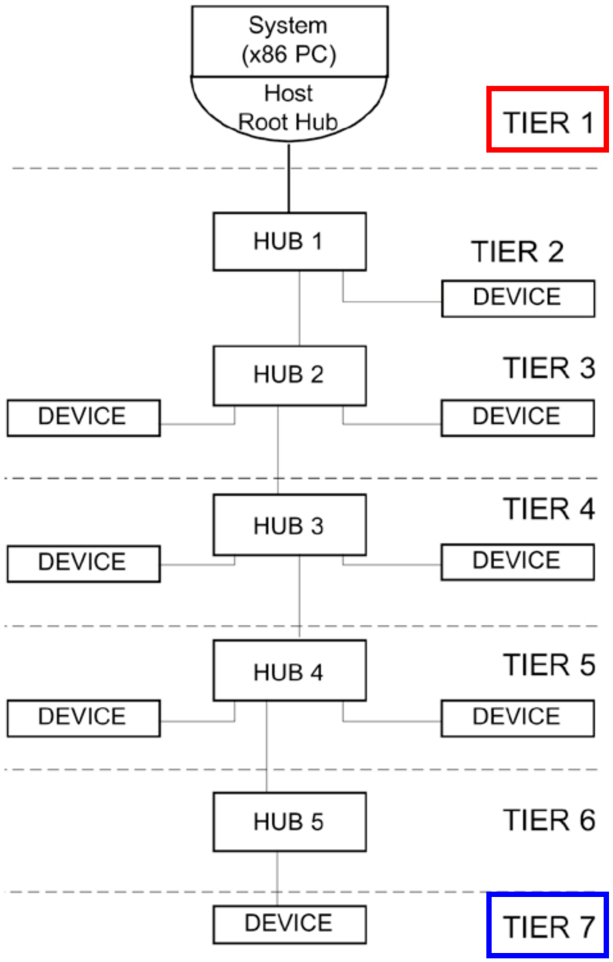


Figure 27-4 Daisy-Chaining Hubs

- The USB bus allows up to five levels of external hubs to be connected to the root hub.
  - Due to power considerations, many of these hubs must be self-powered.
  - USB terminology uses the word ***tier*** instead of *level*.
  - It states the total number of tiers (levels) cannot be more than **seven**.

27.2: USB PORT EXPANSION/POWER MANAGEMENT
daisy chaining the hubs

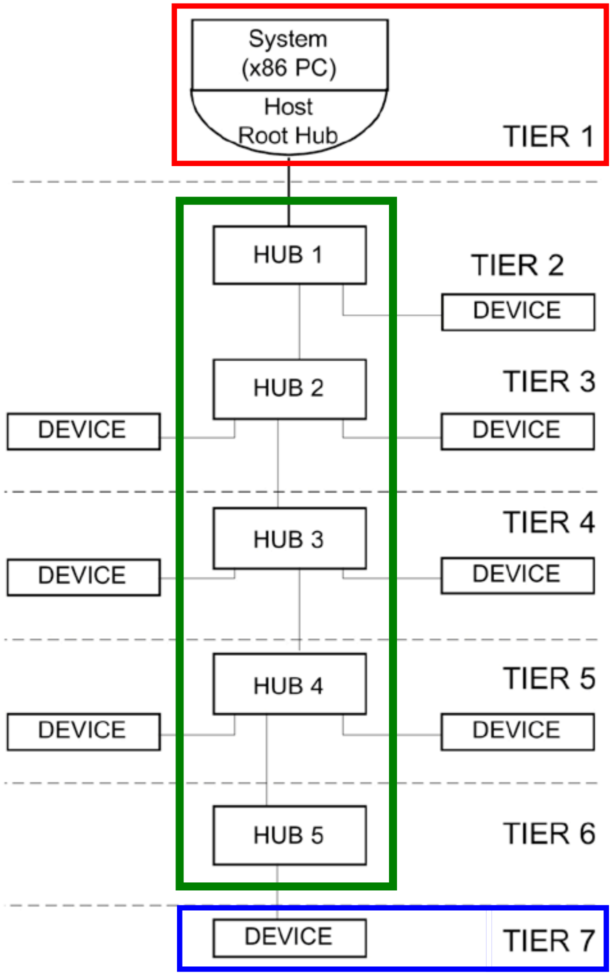


Figure 27-4 Daisy-Chaining Hubs

– The number of levels from the farthest peripheral device to the host (root hub) must be no more than **seven**.

– The host (root hub) is counted as the **first tier** (level).

– The number of external hubs that can be cascaded together is limited to **five**.

– This makes the peripheral connected to the last hub tier number **seven**.

27.2: USB PORT EXPANSION/POWER MANAGEMENT
USB cable signals

- The USB cable has four wires inside it:
  - V_{cc}, **Gnd** for the power source.
  - **D+**, and **D–** for the data path.
    - Data path between host & peripheral is half-duplex

At any given time, data…

is going *from* the **device** *to* the host…

or is coming *from* the **host** *to* the device…

but never at the *same* time.

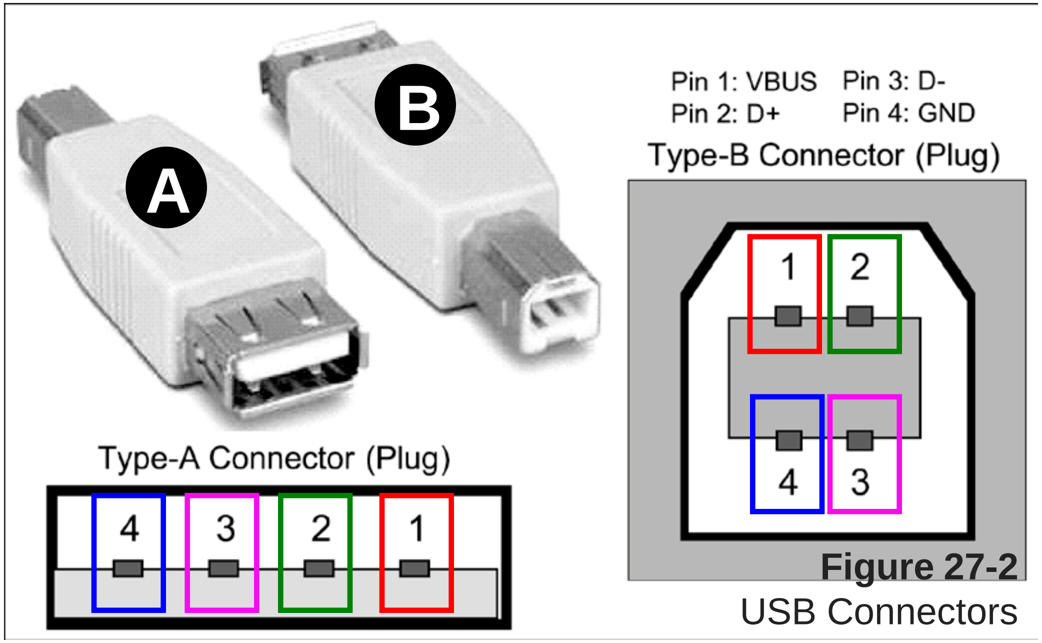


Figure 27-2
USB Connectors

USB cable signals

- The USB cable does not have any wire for clock to synchronize data transfer—it an asynchronous bus.
  - The port uses the D+ and D– pins to implement the asynchronous method of data transfer.
    - It uses NRZI (none return to zero inverted) encoding.
  - Maximum cable length for the USB is 5 meters (15 feet).
    - Less than 5 meters for high-speed data transfers

27.2: USB PORT EXPANSION/POWER MANAGEMENT
USB enumeration

- USB devices are hot-pluggable.
  - The moment it is connected to the USB port of the x86 PC, a peripheral device is recognized by the host.
- There is no need to turn off the PC and then turn it back on in order for the device to be recognized.
  - The host senses the voltage change on the data line.
  - The host enquires about this USB device and after receiving a satisfactory answer, the peripheral is assigned an address.

27.2: USB PORT EXPANSION/POWER MANAGEMENT
USB enumeration

- A USB address value has seven bits.
  - – 0 to 127 (0–7F in hex, or 0000000–1111111 in binary).
    - The host assigns addresses between 1 and 127, since address 0 is the default address of any device before it is recognized by the host.
- Recognizing a device and assigning it an address is called *enumeration* in the USB literature.
  - – As long as this peripheral is connected, and the host is powered, this unique address value is not changed.
    - If the peripheral is disconnected from the host, the host will take back the address and keep it in the address pool.

27.3: USB PORT PROGRAMMING

- In programming the peripherals connected to the x86 PC USB host, the following points must be remembered:
  - A single USB host can connect with up to 127 peripheral devices.
  - Although the x86 can have multiple USB hosts, it is assumed there is only *one* host in the x86 PC.
  - In the Windows OS, opening a connection to a peripheral device is like opening a file.
  - The API provided with the driver handles where the data will go to or come from.

27.3: USB PORT PROGRAMMING communicating via USB-Bluetooth

- As an example of programming a USB port using C#, is a USB-to-Bluetooth device to communicate between two PCs.
  - Each PC will have a USB-Bluetooth device connected to its USB port.

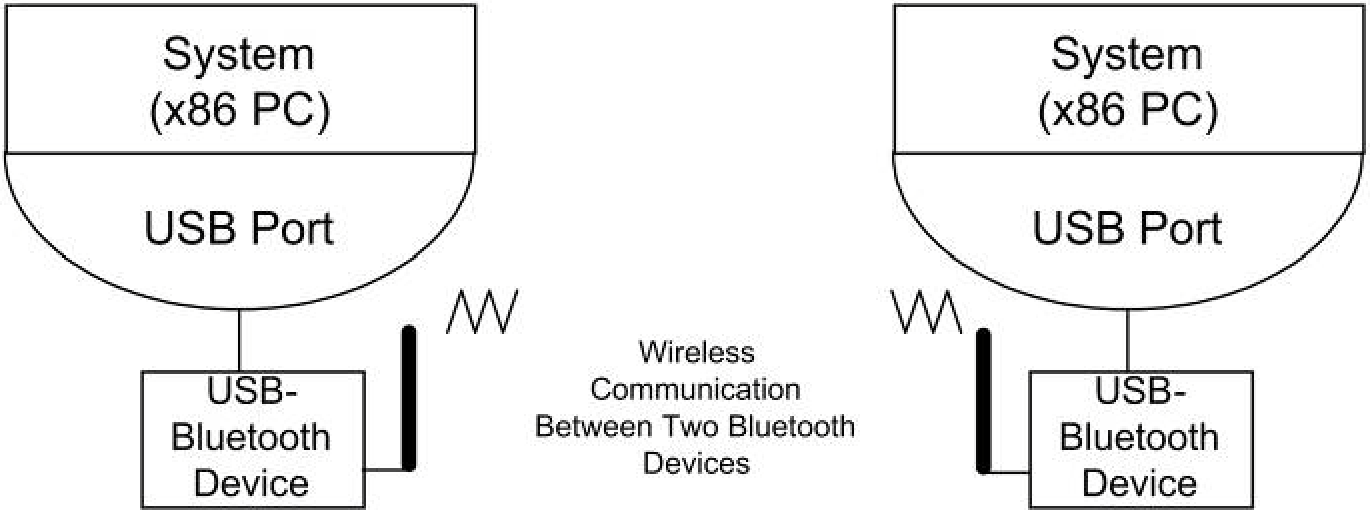


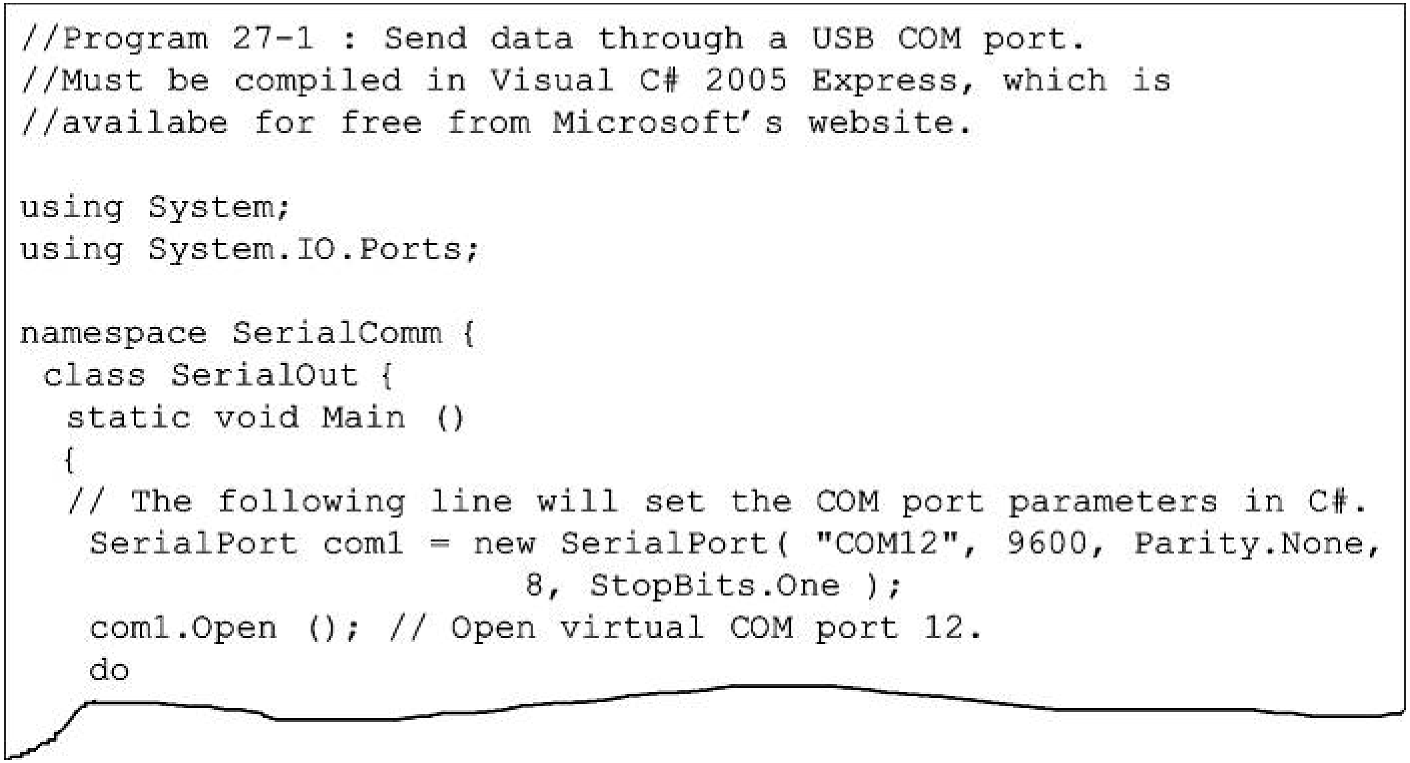
Figure 27-5 PC Communications via USB-Bluetooth Devices

27.3: USB PORT PROGRAMMING
communicating via USB-Bluetooth

- The following steps show how to use the Windows HyperTerminal to establish communication:
  - 1. Use the Bluetooth utility provided with the Bluetooth device to open a virtual COM port on the PC.
  - 2. Configure HyperTerminal to communicate on the virtual COM port assigned by the Bluetooth utility.
  - 3. Perform the same tasks on the second PC.
  - 4. Send messages between the two PCs to verify communications.

27.3: USB PORT PROGRAMMING
communicating via USB-Bluetooth

In Program 27-1, the message “Hello World” is written to the USB-Bluetooth device.



```
//Program 27-1 : Send data through a USB COM port.
//Must be compiled in Visual C# 2005 Express, which is
//availabe for free from Microsoft’ s website.

using System;
using System.IO.Ports;

namespace SerialComm {
 class SerialOut {
  static void Main ()
  {
  // The following line will set the COM port parameters in C#.
   SerialPort com1 = new SerialPort( "COM12", 9600, Parity.None,
                                    8, StopBits.One );
   com1.Open (); // Open virtual COM port 12.
   do
```

See the entire program listing on page 695 of your textbook.

27.3: USB PORT PROGRAMMING communicating via USB-Bluetooth

- Program 27-2, also on page 695 receives the message from the USB-Bluetooth device.
  - By running the above two programs, the message is transferred via the USB port to Bluetooth, and from Bluetooth wirelessly to another PC.

27.3: USB PORT PROGRAMMING communicating via USB

- *Is there any way to communicate between two PCs via a null USB cable?*
  - Absolutely not, since two USB hosts *cannot* talk to each other directly.
    - The USB **host** can *only* talk to a USB **device**.

Dec	Hex	Bin
27	1B	00011011

ENDS ; TWENTY-SEVEN