Overview

In this chapter, you will learn to:
- Identify the structure and function of the expansion bus
- Explain classic system resources
- Identify the modern expansion bus slots
- Install expansion cards properly
- Troubleshoot expansion card problems

Structure and Function of the Expansion Bus

Connections

Expansion slots connect to both the Northbridge and Southbridge

System Crystal

- All integrated circuits must be regulated by a clock crystal
- Every device soldered to the motherboard is designed to run at the speed of the system crystal

Expansion Bus Crystal

- The expansion bus crystal is an extension to the external data bus
  - Used to control the part of the external data bus connected to the expansion slot
- The chipset acts as a divider between the two buses


**CPU Crystals**

- The first bus is the frontside bus that runs at the speed of the system crystal.
- The second bus is the expansion bus that runs at the speed of the expansion bus crystal.

---

**PC Bus (8-bit ISA)**

- IBM XT had 8088 processor and an 8-bit external data bus at a top speed of 4.77 MHz.
- IBM used an expansion bus that could run around 7 MHz (as fast as the system bus) called the PC bus, XT bus, or ISA bus.

---

**AT Bus (16-bit ISA)**

- The AT bus is a 16-bit bus running at a speed of 8.33 MHz and is created by adding a set of connections to the end of the PC bus.
- The PC/XT and AT buses are also known as the 8-bit ISA and 16-bit ISA respectively.

---

**System Resources**

- Expansion cards and the CPU need some way to communicate.
- **System resources** help to define how to communicate:
  - I/O addresses
  - IRQs
  - DMA channels
  - Memory addresses

---

**Who’s Who?**

- How does a device know that the CPU wants its attention?
  - To which device is the CPU talking?
  - The CPU wants to send a signal to one device.

---
I/O Mem Wire

- The expansion bus consists of the external data bus and the address bus
- Every device on the PC connects to both
- When a voltage is placed on the I/O/MEM wire, only the first 16 wires are monitored by all devices

I/O Addresses

- Different wire patterns used by the CPU to communicate with different devices inside the PC are known as I/O addresses

Device Manager

- Device Manager allows you to see what resources are being used by your devices
  - Alt-click My Computer ➔ Properties ➔ Hardware ➔ Device Manager button

Viewing Resources

- Device Manager ➔ Click "+" next to device ➔ double-click device ➔ Resources tab
  - To get to Device Manager, alt-click My Computer and choose Properties ➔ Hardware ➔ Device Manager

I/O Addresses Use Hexadecimal

- Before we can talk about I/O addresses we need to understand hexadecimal and binary math

I/O Addresses

- I/O addresses are represented as 4-digit hexadecimal values starting from 0000 and ending at FFFF
  - Remember that when the CPU turns on the I/O/MEM wire, it sends and I/O address using the first 16 address bus wires
  - These wires either have a voltage (1) or they don’t (0)
  - We represent which wires have a voltage with an address bus binary number with 16 digits
    - 0010101110101101
  - Wouldn’t it be easier to represent this long binary number with the hexadecimal number 2BAD? That’s exactly what computer techs do!
**Rules of I/O Addresses**

- All devices must have an I/O address
  - This is how the CPU talks to everything in the PC
  - The I/O address is either preset or must be assigned
- All devices use more than one I/O address
  - CPU uses different I/O addresses for different commands
  - Devices must be able to respond to the CPU with other I/O addresses
  - Hence, a range of I/O addresses is assigned
- No two devices can share the same I/O address

**I/O Address Terminology**

- When talking about I/O addresses, drop the leading zeros (1F0...not 01F0)
- Every device gets a range of I/O addresses
- The first I/O address is called the base I/O address
- Put an "h" on the end of the value to specify hex (1F0h)
- I/O addresses provide a two-way communication pathway between the peripherals and the CPU

**Interrupts**

- The CPU can initiate a conversation with any device at will
- Any device may talk to the CPU but how does a device get the CPU’s attention?
- Devices use the interruption mechanism to gain the attention of the CPU by placing a voltage on a special wire called the INT (interrupt) wire – the CPU will stop what it is doing and deal with the device

**The Original 8259 Chip**

- The 8259 chip acts as an intermediary between all the devices and the CPU’s INT wire
- It hooks to the INT wire of the CPU on one side, and has 8 other wires called the interrupt requests (IRQs) that extend out from the chip into the motherboard

**IRQs for the System Timer, Keyboard, and ISA Slots**

- Every device has an IRQ
  - The system timer, keyboard controller, and ISA slots are shown

**Cascaded 8259s**

- The 8259 chips were designed to run in a cascade to provide more IRQs
  - IRQs 2 and 9 are the same and referred to as IRQ2/9
### 16-Bit IRQ Map

<table>
<thead>
<tr>
<th>IRQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>System timer</td>
</tr>
<tr>
<td>1</td>
<td>Keyboard</td>
</tr>
<tr>
<td>2/79</td>
<td>Open for use</td>
</tr>
<tr>
<td>3</td>
<td>Default COM1, COM4</td>
</tr>
<tr>
<td>4</td>
<td>Default COM2, COM3</td>
</tr>
<tr>
<td>5</td>
<td>LPT2</td>
</tr>
<tr>
<td>6</td>
<td>Floppy Drive</td>
</tr>
<tr>
<td>7</td>
<td>LPT1</td>
</tr>
<tr>
<td>8</td>
<td>Real-time clock</td>
</tr>
<tr>
<td>9</td>
<td>Open for use</td>
</tr>
<tr>
<td>10</td>
<td>Open for use</td>
</tr>
<tr>
<td>11</td>
<td>Open for use</td>
</tr>
<tr>
<td>12</td>
<td>Multi-user processor</td>
</tr>
<tr>
<td>13</td>
<td>Primary hard-drive controller</td>
</tr>
<tr>
<td>14</td>
<td>Secondary hard-drive controller</td>
</tr>
</tbody>
</table>

### The Rules of IRQs

- **Every device needs an IRQ**
  - Joysticks do not use interrupts (there’s always an exception!)
- **No two devices can share an IRQ**
  - What happens if they do?
  - Could two devices exist using the same interrupt?

### COM and LPT Ports

- **Every peripheral needs an IRQ and I/O address**
- IBM created standard preset combinations of IRQs and I/O addresses
- These preset combinations for **serial devices** and **parallel devices** are called COM ports and LPT ports respectively
  - COM1 uses IRQ 4
  - COM2 uses IRQ 3
  - LPT1 uses IRQ 7
  - LPT2 uses IRQ 5

### COM and LPT Ports

- COM3 and COM4 are two common port standards that are assigned the 3E8-3EF and 2E8-2EF set of I/O addresses respectively
- COM3 uses IRQ4 and COM4 uses IRQ3
  - These are the same IRQ’s used for COM1 and COM2
  - How can two devices use the same IRQ...as long as they never talk at the same time!
- LPT (Line Printer) port settings apply to parallel connections for devices such as printers

### COM Port Assignments

<table>
<thead>
<tr>
<th>Port</th>
<th>I/O Base Address</th>
<th>IRQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM1</td>
<td>03F8</td>
<td>4</td>
</tr>
<tr>
<td>COM2</td>
<td>02F8</td>
<td>3</td>
</tr>
<tr>
<td>COM3</td>
<td>03F8</td>
<td>4</td>
</tr>
<tr>
<td>COM4</td>
<td>02F8</td>
<td>3</td>
</tr>
<tr>
<td>LPT1</td>
<td>0378</td>
<td>7</td>
</tr>
<tr>
<td>LPT2</td>
<td>0278</td>
<td>5</td>
</tr>
</tbody>
</table>

### Physical vs. I/O Ports

**I/O ports:**
- A serial port is a 9 or 15-pin male DB connector, whereas the COM port is just the I/O address and IRQ assigned to it
- A parallel port is a 25-pin female DB connector, whereas the LPT port is just the IRQ and an I/O address assigned to it
**COM Ports and LPT Ports Today**

- IBM dictated a specific I/O address and IRQ for a particular COM or LPT port
- The IRQ can be changed as long as the device can handle it, and the software that communicates with the device knows about the change
- Change COM or LPT IRQs in the CMOS setup

**Direct Memory Access (DMA)**

- Direct Memory Access (DMA) is the process of accessing memory without using the CPU
- It enables the system to run background applications without interrupting the CPU

**The 8237 Chip**

- The 8237 chip is used to control DMA functions
- It links to the CPU via a HRQ wire
- The chip has 4 wires, called DMA requests (DRQs) or DMA channels, which lead to the DRAM refresh circuitry and ISA slots
- No two devices can share DRQs

**Cascaded 8237 Chips**

- Provides up to 7 DMA channels
- DRQ0 and DRQ4 are linked
- DMA channels 1 thru 3 are for 8-bit transfers
- DMA channels 5 thru 7 are for 16-bit transfers
- Designed for the ISA bus

**DMA Assignments**

<table>
<thead>
<tr>
<th>DMA</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRQ0/4</td>
<td>8-bit</td>
<td>None</td>
</tr>
<tr>
<td>DRQ1</td>
<td>8-bit</td>
<td>Open for use</td>
</tr>
<tr>
<td>DRQ2</td>
<td>8-bit</td>
<td>Floppy drive controller</td>
</tr>
<tr>
<td>DRQ3</td>
<td>8-bit</td>
<td>Open for use</td>
</tr>
<tr>
<td>DRQ5</td>
<td>16-bit</td>
<td>Open for use</td>
</tr>
<tr>
<td>DRQ6</td>
<td>16-bit</td>
<td>Open for use</td>
</tr>
<tr>
<td>DRQ7</td>
<td>16-bit</td>
<td>Open for use</td>
</tr>
</tbody>
</table>

**Bus Mastering**

- Bus masters are devices that use the DMA without accessing the 8237 or CPU
- They have a circuitry that allows them to watch for other devices accessing the external data bus
  - No two devices can use the external data bus at the same time
- Bus mastering is extremely popular in hard drives
  - All IDE hard drives take advantage of bus mastering
- Sound cards and floppy drives still use the old DMA
Modern Expansion Bus

Microchannel Architecture (MCA)
- MCA had a 32-bit bus to match the 386 CPU’s external data bus with a speed of 12MHz
  - May self configure devices
  - IBM proprietary and expensive

Enhanced ISA (EISA)
- EISA (EE-sah) was a 32-bit self-configuring expansion bus that was cheaper than the MCA
  - Used a unique double-slot connector compatible with ISA devices

VESA VL-Bus
- The Video Electronics Standards Association (VESA) created the VESA local (VL) bus to solve the problems of speed and throughput
  - Speed of 33 MHz
  - Paired with an ISA slot typically
  - Controlled only the functions specific to VL-bus devices...relying on the ISA slot for basic control functions

Local Bus Concept

Local Bus Concept
**Peripheral Component Interconnect (PCI)** provides a stronger, faster, and flexible alternative to any other expansion bus

- The flexible design enables the PCI to coexist with other buses, and scale up in speed and throughput
- PCI devices are self-configuring
- PCI Special Interest Group (SIG) defined I/O addresses and IRQs for most devices...using a sharable Interrupt Channel instead of IRQs
- Fully implements DMA – allowing PCI devices to transfer data among themselves

**Bus mastering enables the PCI devices to transfer data between themselves**

- Its burst mode feature enables efficient data transfers

**Divides its chipset duties between two chips**

- Northbridge (or PCI controller) performs the classic functions and controls the PCI bus
- Southbridge (PCI to ISA bridge or just PCI bridge) acts as an intermediary between the PCI bus and the other bus

**Accelerated Graphic Port**

- One of the main reasons Intel designed the AGP was to allow the video card to have a high speed connection directly to the system ram
- The AGP was created in 1996 as a bus specifically designed for high performance graphics and video support
- AGP is a high speed connection and runs at a base frequency of 66 MHz which is double that of a standard PCI
- AGP is based on PCI, but it contains additions and enhancements and is physically, electronically, and logically independent of the PCI

**PCI-X**

- PCI-X is a huge advancement to PCI
  - Fully backward compatible
  - Uses the same connectors, so it will accept standard PCI cards
  - 32-bit and the more common 64-bit versions
  - Much faster with four speed grades
    - PCI-X 66
    - PCI-X 133
    - PCI-X 266
    - PCI-X 533
  - Parallel interface like PCI
PCI-Express

- More revolutionary than evolutionary
  - Serial interface
  - Not hardware compatible
  - Software compatible
  - Flexible, scalable, native hot swap/hot plug capable
  - Very fast at 2.0 Gbps for each lane between the controller and the device
    - May use up to 32 lanes for a maximum bandwidth of 128 Gbps

Installing Expansion Cards

Steps to Installing Expansion Cards

1. Knowledge
2. Physical installation
3. Assigning resources to the card
4. Device drivers
5. Verify

Step 1: Knowledge

1. Learn about the device by reading the documentation
   - Do you have device drivers for your operating system (with Windows 98 and later you usually do)
   - Does the device work with your operating system?
   - Check the Hardware Compatibility List for Windows 2000 or XP
     - Devices on this list have been certified by Microsoft to work with Windows
     - The Windows installation CD contains the HCL, or you may get the latest version on Microsoft’s web site at www.microsoft.com

Step 2: Physical Installation

2. Install the card
   - Hold the card on its edges - don’t touch connectors or the components on the card
   - Insert at the proper angle
   - Use the connection screw which helps to ground the card
   - Use proper ESD procedures

Step 3: Assigning Resources

3. Every expansion card must have system resources assigned to it
   - Every PCI motherboard and card can use Plug and Play
     - Some of the later ISA cards were Plug and Play-capable
     - But sometimes Plug and Play doesn’t work properly, so every good tech needs to know how to troubleshoot it
**PnP (Plug and Play)**

- PnP consists of a series of standards designed to enable self-configuration of devices
- Devices that do not support PnP are called legacy devices

**PnP Requirements**

1. **PnP BIOS**
   - All Pentium and later computers have a PnP BIOS
2. **PnP Operating System**
   - Windows 95 or later
3. **PnP device**
   - No one makes non-PnP devices anymore

**PnP BIOS**

```
<table>
<thead>
<tr>
<th>System BIOS</th>
<th>Legacy BIOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>386</td>
<td>386</td>
</tr>
<tr>
<td>486</td>
<td>486</td>
</tr>
<tr>
<td>Pentium</td>
<td>Pentium</td>
</tr>
<tr>
<td>Pentium II</td>
<td>Pentium II</td>
</tr>
</tbody>
</table>
```

**Legacy Devices in the CMOS**

- System resources refers to I/O addresses, IRQs, and DMA used

**Legacy Devices have Jumpers**

- How PnP Works
  - Initially PnP devices remain quiet while the BIOS determines resources required by legacy devices

```
PnP BIOS instructs PnP 1 and PnP 2 to be quiet.
PnP 1 and PnP 2 go quiet.
```
How PnP Works

• A legacy device list is created to reserve those system resources

Next the PnP BIOS checks with the PnP devices to see which system resources are options for each device

The PnP BIOS then assigns system resources based on that information

1. PnP BIOS assigns resources.

Step 4: Device Drivers

4. All devices require BIOS, which for expansion cards is almost always a device driver
   - Devices will come with device drivers on the installation CD
   - It is recommended that you get the latest drivers from the manufacturer’s web site
   - To update the device drivers you may need to uninstall the current driver first
   - Windows XP has a feature to rollback your system to the previous driver just in case the new driver you installed does not work

Updating the Driver

• Alt-click on the device in Device Manager and choose Update Driver...
  - Choose Uninstall to remove the current driver

Driver Rollback

• Alt-click the device in Device Manager and choose Properties, then the Driver tab
**Step 5: Verify**

5. Check the device properties in Device Manager to verify it is working properly.

**Troubleshooting Expansion Cards**

**Device Manager**

- Check for the device in Device Manager
  - Alt-click My Computer > Properties > Hardware tab > Device Manager
  - If the device does not show up in Device Manager
    - Run the Add/Remove Hardware Wizard in Control Panel
    - If it still doesn't show up, the device is damaged or it is a legacy device whose system resources are not configured properly

**Device Manager Symbols**

- **Black ! on a yellow circle**
  - Device is missing, Windows doesn't recognize it, or a device driver problem – device may still work
- **Red X**
  - Disabled device – system resource conflict or damaged device – will not work
- **Blue I on a white background**
  - System resources were configured manually
- **Green ?**
  - Windows does not have the correct driver but has successfully installed a compatible driver

**Rules for Device Installation**

- **Installing legacy devices:**
  1. Run the Device Manager to determine the available resources for the system
  2. Configure the device to use the resources
     - You may need to set jumpers, flip switches, or run a special setup program
  3. Run the Add/Remove Hardware Wizard to inform Windows of the legacy device

**Device Manager Sorted by IRQ**
Changing the Resources

- The Resources tab of the device Properties sheet show which resources Windows has assigned.
- You may use the automatic settings, or uncheck the box and click Change Settings to manually specify settings to use.

Legacy Setup Software Program