Computer Block Diagram

- ROM (BIOS)
- RAM
- Local Bus
- CPU
  - L1
  - L2
- Graphics Adapter
- Adapters:
  - Serial
  - Parallel
  - USB
  - SCSI
  - Firewire
- PCI or AGP Local Bus
- EISA or PCI Bus
- Disk array
- Mouse
- Keyboard
- Removable storage
- Modem
- Laser printer
- Scanner
<table>
<thead>
<tr>
<th>Generation</th>
<th>Processor</th>
<th>Manufacturer</th>
<th>Data Lines</th>
<th>Address Space</th>
<th>Cache Size</th>
<th>Coprocessor</th>
<th>Clock Doubler</th>
</tr>
</thead>
<tbody>
<tr>
<td>8088</td>
<td>8088</td>
<td>Various</td>
<td>8</td>
<td>20</td>
<td>1Mb</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8086</td>
<td>8086</td>
<td>Various</td>
<td>16</td>
<td>20</td>
<td>1Mb</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80286</td>
<td>80286</td>
<td>Various</td>
<td>16</td>
<td>24</td>
<td>16Mb</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80386</td>
<td>80386SX</td>
<td>Intel, AMD</td>
<td>16</td>
<td>24</td>
<td>16Mb</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>80386DX</td>
<td>Intel, AMD</td>
<td>32</td>
<td>32</td>
<td>4Gb</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>80386SLC</td>
<td>IBM</td>
<td>16</td>
<td>24</td>
<td>16Mb</td>
<td>8Kb</td>
<td>-</td>
</tr>
<tr>
<td>80486</td>
<td>80486SX</td>
<td>Intel</td>
<td>32</td>
<td>32</td>
<td>4Gb</td>
<td>8Kb</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>80486DX</td>
<td>Intel</td>
<td>32</td>
<td>32</td>
<td>4Gb</td>
<td>8Kb</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>80486SX2</td>
<td>Intel</td>
<td>32</td>
<td>32</td>
<td>4Gb</td>
<td>8Kb</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>80486DX2</td>
<td>Intel</td>
<td>32</td>
<td>32</td>
<td>4Gb</td>
<td>8Kb</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>80486SLC2</td>
<td>IBM</td>
<td>16</td>
<td>24</td>
<td>16Mb</td>
<td>16Kb</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>80486SLC</td>
<td>Cyrix</td>
<td>16</td>
<td>24</td>
<td>16Mb</td>
<td>1Kb</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>80486DLC</td>
<td>Cyrix</td>
<td>32</td>
<td>32</td>
<td>4Gb</td>
<td>1Kb</td>
<td>-</td>
</tr>
</tbody>
</table>

The range of PC processors currently available
<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Transistors</th>
<th>Microns</th>
<th>Clock speed</th>
<th>Data width</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>8080</td>
<td>1974</td>
<td>6,000</td>
<td>6</td>
<td>2 MHz</td>
<td>8 bits</td>
<td>0.64</td>
</tr>
<tr>
<td>8088</td>
<td>1979</td>
<td>29,000</td>
<td>3</td>
<td>5 MHz</td>
<td>16 bits, 8-bit bus</td>
<td>0.33</td>
</tr>
<tr>
<td>80286</td>
<td>1982</td>
<td>134,000</td>
<td>1.5</td>
<td>6 MHz</td>
<td>16 bits</td>
<td>1</td>
</tr>
<tr>
<td>80386</td>
<td>1985</td>
<td>275,000</td>
<td>1.5</td>
<td>16 MHz</td>
<td>32 bits</td>
<td>5</td>
</tr>
<tr>
<td>80486</td>
<td>1989</td>
<td>1,200,000</td>
<td>1</td>
<td>25 MHz</td>
<td>32 bits</td>
<td>20</td>
</tr>
<tr>
<td>Pentium</td>
<td>1993</td>
<td>3,100,000</td>
<td>0.8</td>
<td>60 MHz</td>
<td>32 bits, 64-bit bus</td>
<td>100</td>
</tr>
<tr>
<td>Pentium II</td>
<td>1997</td>
<td>7,500,000</td>
<td>0.35</td>
<td>233 MHz</td>
<td>32 bits, 64-bit bus</td>
<td>~300</td>
</tr>
<tr>
<td>Pentium III</td>
<td>1999</td>
<td>9,500,000</td>
<td>0.25</td>
<td>450 MHz</td>
<td>32 bits, 64-bit bus</td>
<td>~510</td>
</tr>
<tr>
<td>Pentium 4</td>
<td>2000</td>
<td>42,000,000</td>
<td>0.18</td>
<td>1.5 GHz</td>
<td>32 bits, 64-bit bus</td>
<td>~1,700</td>
</tr>
</tbody>
</table>
Instruction Hierarchy

Gated Instructions

\textit{add/or/and}

Microcode

ALU

Instructions

Software Programs

RAM/External

Firmware

ROM/Internal
Basic Procedure for Program Interpretation & Execution

1. Start
2. Fetch next instruction
3. Decode instruction (Determine the operation and operand designators)
4. Fetch designated operands
5. Branch to designated operation
6. Execute primitive operation 1
7. Execute primitive operation 2
8. ... (Continue for k operations)
9. Execute primitive operation k
10. Execute halt operation
11. Stop
Role of an Operating System

- User interface
  - GUI
- Application interface
- Memory Management
- Time Management
  - Multitasking, virtual machine handler
- File System Manager
- Programming interface
  - API’s, ActiveX
- Hardware Interface
  - Device Drivers, VxDs
- Multimedia Framework
  - Video for Windows, DirectX
Space and Time Management

- Intel Hardware Processing Modes
  - Real mode
  - Protected mode
  - Virtual 8086 mode

- Operating System Multiprocessing Modes:
  - Time sharing
  - Cooperative multitasking
  - Preemptive multitasking
Figure 18-1: This diagram shows a PC with 16MB of RAM (640K of conventional memory plus 15744K of extended memory = 16384K plus 384K of UMBs, for a total of 16768K). EMS memory is taken from the extended memory by using EMM386.exe or other third-party memory managers, or is given to DOS programs running in Windows DOS sessions by Windows 95 itself.
**Comparative OS Architectures:**

**WINDOWS 3.1**

Windows 3.1 embodies trade-offs between performance and protection that hark back to the days of the 286. While it provides good performance for Win16 applications, DOS applications, real-mode device drivers, and virtual device drivers (VxDs) it offers almost no protection against badly behaved applications.

The segmented memory space of a Win16 application is visible to and addressable by all other Win16 applications. Operating-system kernel DLLs and VxDs are also visible to all applications. This design, where a single pool of memory is shared among applications and the OS, provides fast performance as API calls to system DLLs do not entail ring transitions or context switches. The clear disadvantage is that any application can scribble on memory belonging to other applications or operating-system components, potentially bringing down the whole system.

Win16 applications are cooperatively multitasked. Under this system, the application, rather than the operating system, is responsible for giving up control to the next application. A crashed or misbehaving program can keep all others from receiving system resources.

**WINDOWS 95**

Windows 95 strikes a balance among performance, compatibility, and robustness. It offers fast execution of Win32, Win16, and DOS applications, and it can use real-mode device drivers. While it offers better crash protection than Windows 3.1, it remains vulnerable on several fronts.

Addresses between 2GB and 4GB are mapped into the address space of each Win32 application and are shared by all processes.

Addresses between 3GB and 4GB contain the Ring 0 components, including virtual device drivers (VxDs), the virtual machine management subsystem, and the file management subsystem. This memory area is visible to and writable by Win32 apps.

Addresses between 2GB and 3GB contain all Ring 3 Windows system DLLs, such as Kernel, GDI, and User, as well as any Win16 applications.

All 16-bit Windows applications run here in a shared address space, where they are cooperatively multitasked. This virtual 16-bit Windows machine is subject to the same vulnerabilities as Windows 3.1.

Between 4MB and 2GB, each Win32 app sees a private flat address space; it cannot see any other Win32 processes. Errors occurring here can usually be trapped without bringing down the operating system.

Addresses from 0 bytes to 4MB are shared by all processes. This is required for compatibility with real-mode device drivers, TSRs, and some 16-bit Windows applications, but makes it possible for any process to corrupt components residing in those addresses. While the first 64K is not addressable by Win32 applications, it can be addressed and potentially corrupted by 16-bit applications.
### Windows Memory

#### Windows 9x
- **0xFFFFFFF**: 1GB region for system code, shared by all Win32 processes (don't use)
- **0xC0000000**: 1GB region for memory-mapped files, shared Win32 DLLs, 16-bit applications; shared by all Win32 processes
- **0xB0000000**: 2GB region for applications, private to each Win32 process
- **0x00000000**: 4MB region for DOS and 16-bit Windows (don't use)
- **0x000000FF**: 4K region for DOS and 16-bit Windows (inaccessible)

#### Windows NT
- **0xFFFFFFF**: 2GB region for system code, shared by all Win32 processes (inaccessible)
- **0x80000000**: 64K region (don't use)
- **0x7FFFFF**: 2GB region for applications, private to each Win32 process
- **0x7FF0000**: 64K region (don't use)