

Assembly Language for x86 Processors

7th Edition, Global Edition

Kip Irvine

Chapter 2: x86 Processor Architecture

Chapter Overview

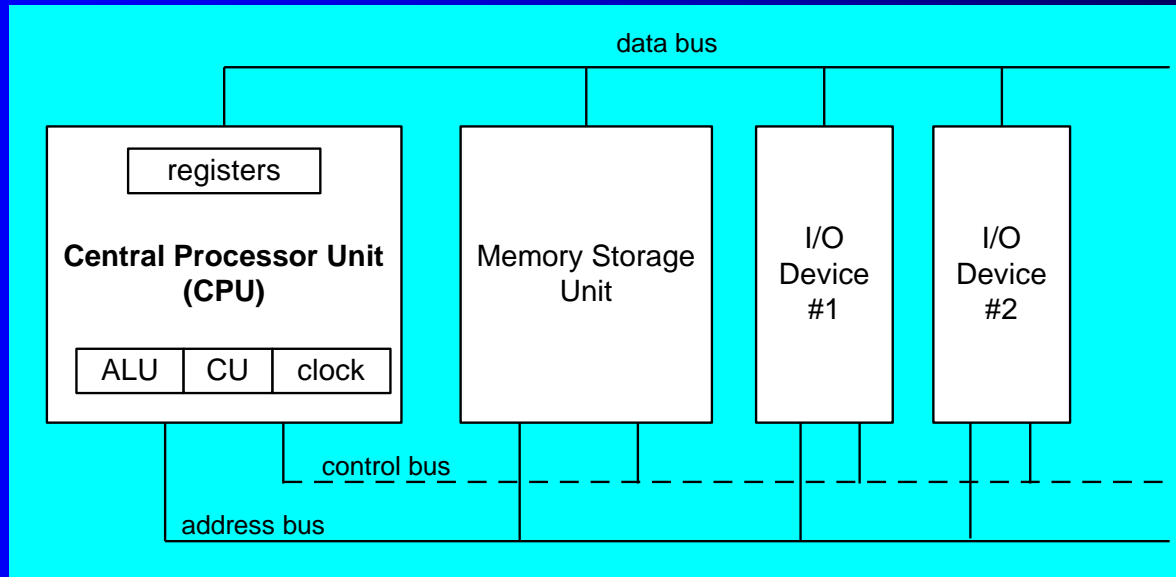
- General Concepts
- IA-32 Processor Architecture
- IA-32 Memory Management
- 64-bit Processors
- Components of an IA-32 Microcomputer
- Input-Output System

General Concepts

- Basic microcomputer design
- Instruction execution cycle
- Reading from memory
- How programs run

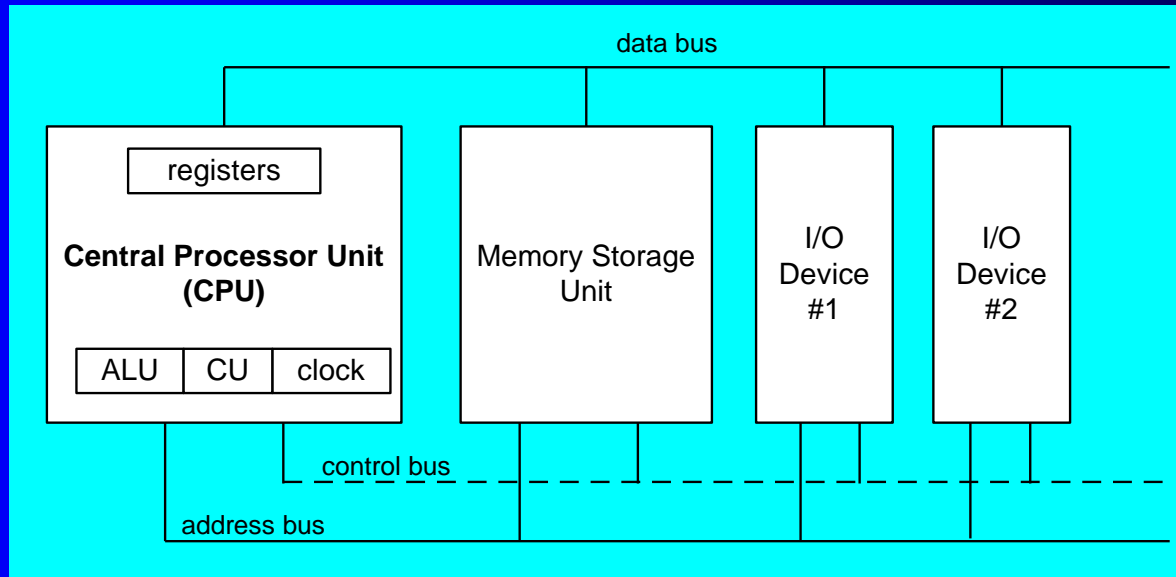
Basic Microcomputer Design (1/2)

- The central processor unit (CPU) is where all the calculations and logic operations take place.
 - clock synchronizes CPU operations
 - control unit (CU) coordinates sequence of execution steps
 - Arithmetic logic unit (ALU) performs arithmetic and bitwise processing



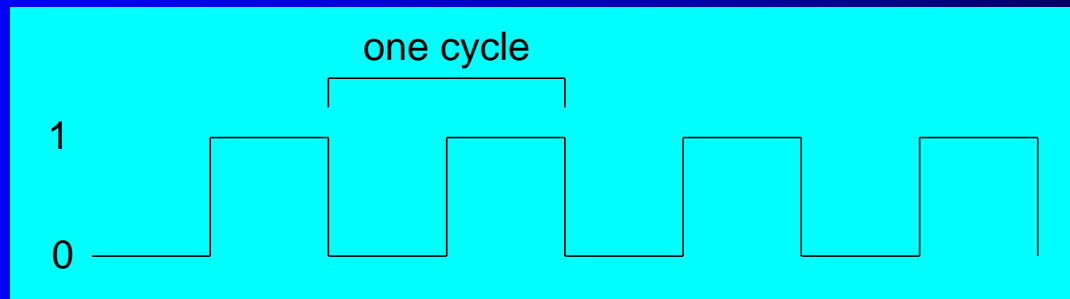
Basic Microcomputer Design (2/2)

- The memory storage unit is where instructions and data are held while a computer program is running.
- A bus is a group of parallel wires that transfer data from one part of the computer to another.
 - Data bus, address bus and control bus



Clock

- synchronizes all CPU and BUS operations
- machine (clock) cycle measures time of a single operation
 - A machine instruction requires at least one clock cycle to execute.
 - A few instructions (e.g., the multiply instruction) require in excess of 50 clocks.
- clock is used to trigger events



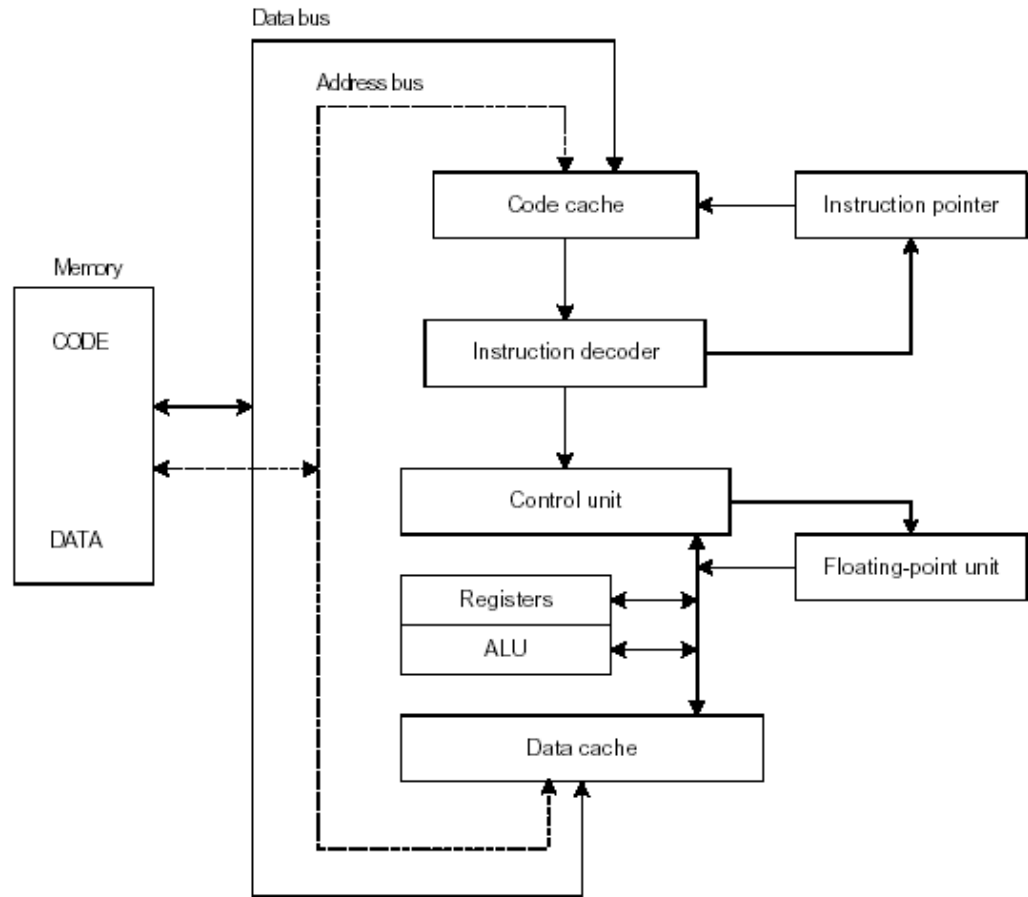
Instruction Execution Cycle (1/2)

- The execution of a single machine instruction can be divided into a sequence of individual operations.
- Three primary operations: **fetch**, **decode** and **execute**.
- Two more steps are required when the instruction uses a memory operand: **fetch operand** and **store output operand**.

Instruction Execution Cycle (2/2)

- Fetch
- Decode
- Fetch operands
- Execute
- Store output

Figure 2-2 Simplified Pentium CPU Block Diagram.



Reading from Memory

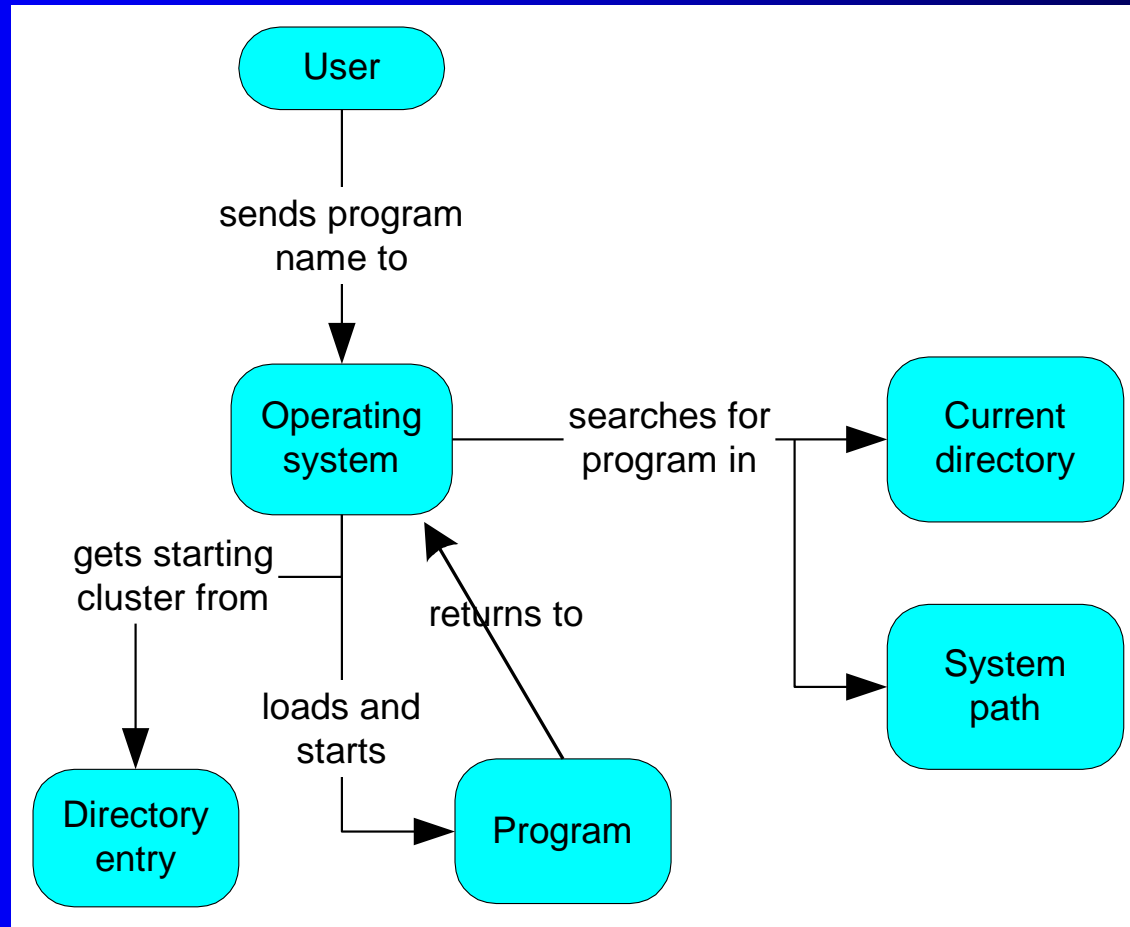
Multiple machine cycles are required when reading from memory, because it responds much more slowly than the CPU. The steps are:

1. Place the address of the value you want to read on the address bus.
2. Assert (change the value of) the processor's RD (read) pin.
3. Wait one clock cycle for the memory chips to respond.
4. Copy the data from the data bus into the destination operand

Cache Memory

- High-speed expensive static RAM both inside and outside the CPU.
 - Level-1 cache: inside the CPU
 - Level-2 cache: outside the CPU
- Cache hit: when data to be read is already in cache memory
- Cache miss: when data to be read is not in cache memory.

How a Program Runs



IA-32 Processor Architecture

- Modes of operation
- Basic execution environment
- Floating-point unit
- Intel Microprocessor history

Modes of Operation

- Protected mode
 - native mode (Windows, Linux)
 - Real-address mode
 - native MS-DOS
 - System management mode
 - power management, system security, diagnostics
- Virtual-8086 mode
 - hybrid of Protected
 - each program has its own 8086 computer

Basic Execution Environment

- Addressable memory
- General-purpose registers
- Index and base registers
- Specialized register uses
- Status flags
- Floating-point, MMX, XMM registers

Addressable Memory

- Protected mode
 - 4 GB
 - 32-bit address
- Real-address and Virtual-8086 modes
 - 1 MB space
 - 20-bit address

General-Purpose Registers

- Registers are high-speed storage locations directly inside the CPU
- Optimized for speed.
 - e.g., loop processing

32-bit General-Purpose Registers

EAX
EBX
ECX
EDX

EBP
ESP
ESI
EDI

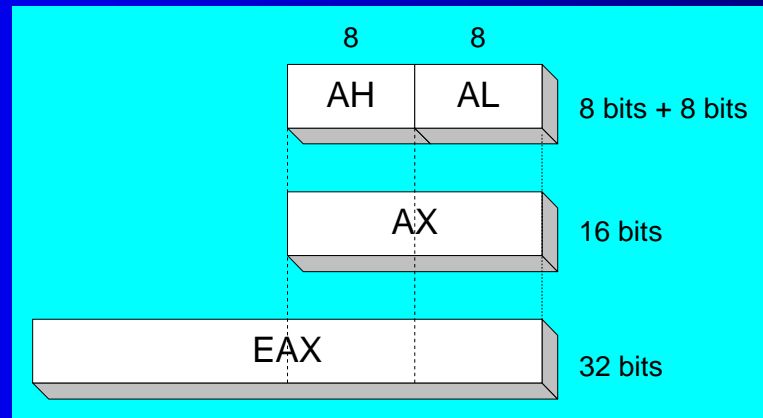
16-bit Segment Registers

EFLAGS
EIP

CS	ES
SS	FS
DS	GS

Accessing Parts of Registers

- Used for arithmetic and data movement
- Use 8-bit name, 16-bit name, or 32-bit name
- Applies to EAX, EBX, ECX, and EDX



32-bit	16-bit	8-bit (high)	8-bit (low)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL

Index and Base Registers

- Some registers have only a 16-bit name for their lower half:

32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP

Some Specialized Register Uses (1 of 2)

- General-Purpose
 - EAX – accumulator (used by multiplication and division instructions)
 - ECX – loop counter
 - ESP – stack pointer addresses data on the stack
 - ESI, EDI – source/destination index registers (for high-speed memory transfer)
 - EBP – extended frame pointer (used by high-level languages to reference function parameters and local variables)
- Segment (as base locations for pre-assigned memory areas)
 - CS – code segment
 - DS – data segment
 - SS – stack segment
 - ES, FS, GS - additional segments

Some Specialized Register Uses (2 of 2)

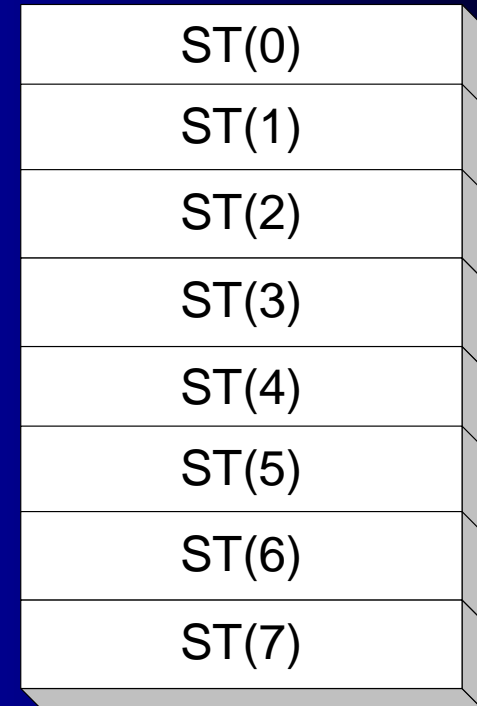
- EIP – instruction pointer
 - Containing the address of the next instruction to be executed
- EFLAGS
 - status and control flags
 - Control the operation of the CPU or reflect the outcome of some CPU operation each flag is a single binary
 - each flag is a single binary bit

Status Flags

- Carry
 - unsigned arithmetic out of range
- Overflow
 - signed arithmetic out of range
- Sign
 - result is negative
- Zero
 - result is zero
- Auxiliary Carry
 - carry from bit 3 to bit 4
- Parity
 - sum of 1 bits is an even number

Floating-Point, MMX, XMM Registers

- Eight 80-bit floating-point data registers
 - ST(0), ST(1), . . . , ST(7)
 - arranged in a stack
 - used for all floating-point arithmetic
- Eight 64-bit MMX registers
- Eight 128-bit XMM registers for single-instruction multiple-data (SIMD) operations



ST(0)
ST(1)
ST(2)
ST(3)
ST(4)
ST(5)
ST(6)
ST(7)

IA-32 Memory Management

- Real-address mode
- Calculating linear addresses
- Protected mode
- Multi-segment model
- Paging

Protected Mode (1 of 2)

- 4 GB addressable RAM
 - (00000000 to FFFFFFFFh)
- Each program assigned a memory partition which is protected from other programs
- Designed for multitasking
- Supported by Linux & MS-Windows

64-Bit Processors

- 64-Bit Operation Modes
 - Compatibility mode – can run existing 16-bit and 32-bit applications (Windows supports only 32-bit apps in this mode)
 - 64-bit mode – Windows 64 uses this
- Basic Execution Environment
 - addresses can be 64 bits (48 bits, in practice)
 - 16 64-bit general purpose registers
 - 64-bit instruction pointer named RIP

64-Bit General Purpose Registers

- 32-bit general purpose registers:
 - EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP, R8D, R9D, R10D, R11D, R12D, R13D, R14D, R15D
- 64-bit general purpose registers:
 - RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8, R9, R10, R11, R12, R13, R14, R15

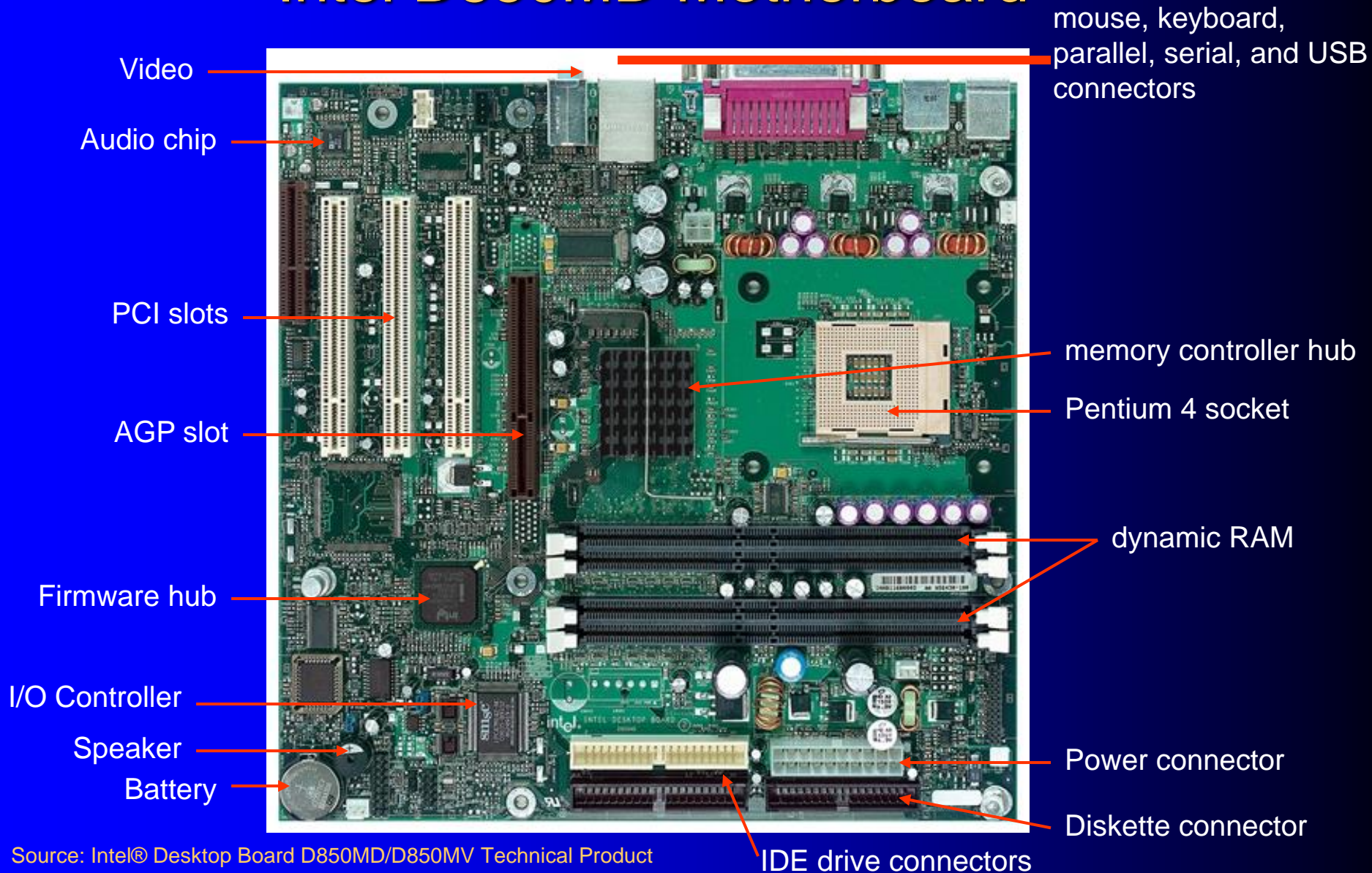
Components of an IA-32 Microcomputer

- Motherboard
- Video output
- Memory
- Input-output ports

Motherboard

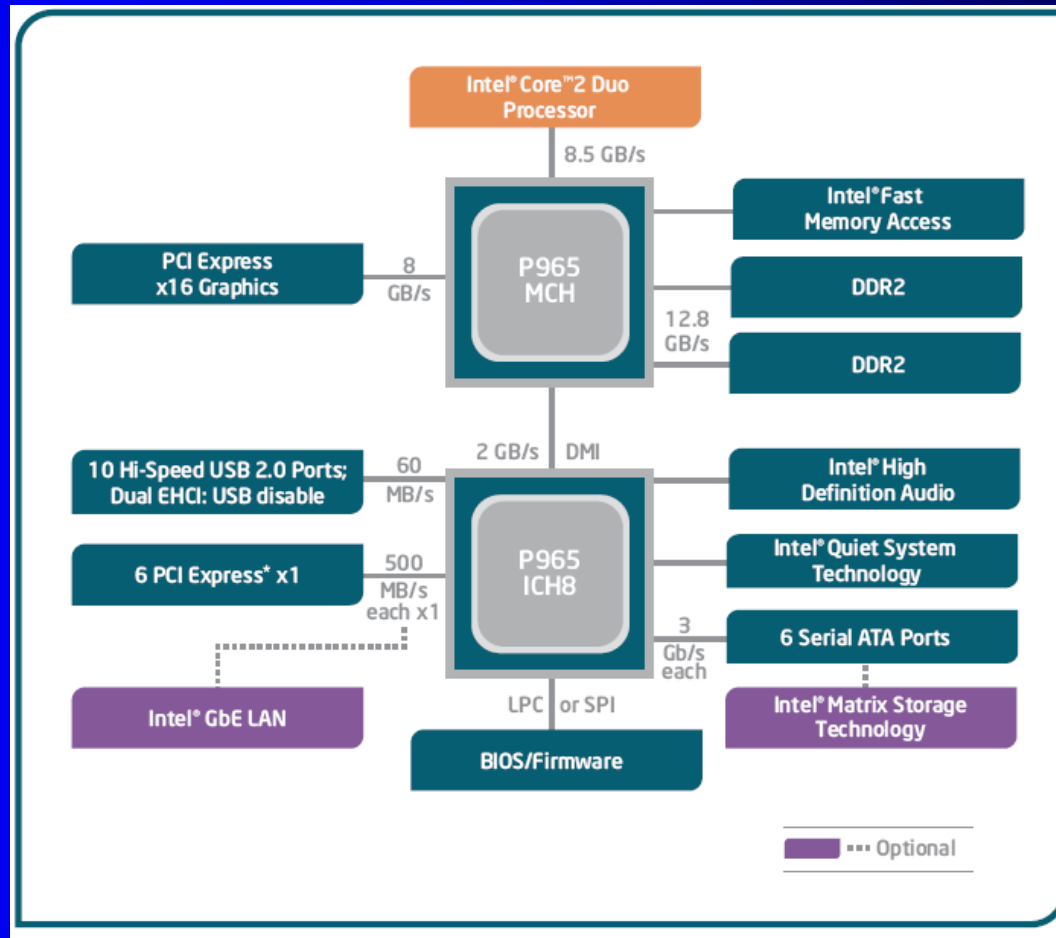
- CPU socket
- External cache memory slots
- Main memory slots
- BIOS chips
- Sound synthesizer chip (optional)
- Video controller chip (optional)
- IDE, parallel, serial, USB, video, keyboard, joystick, network, and mouse connectors
- PCI bus connectors (expansion cards)

Intel D850MD Motherboard



Source: Intel® Desktop Board D850MD/D850MV Technical Product Specification

Intel 965 Express Chipset



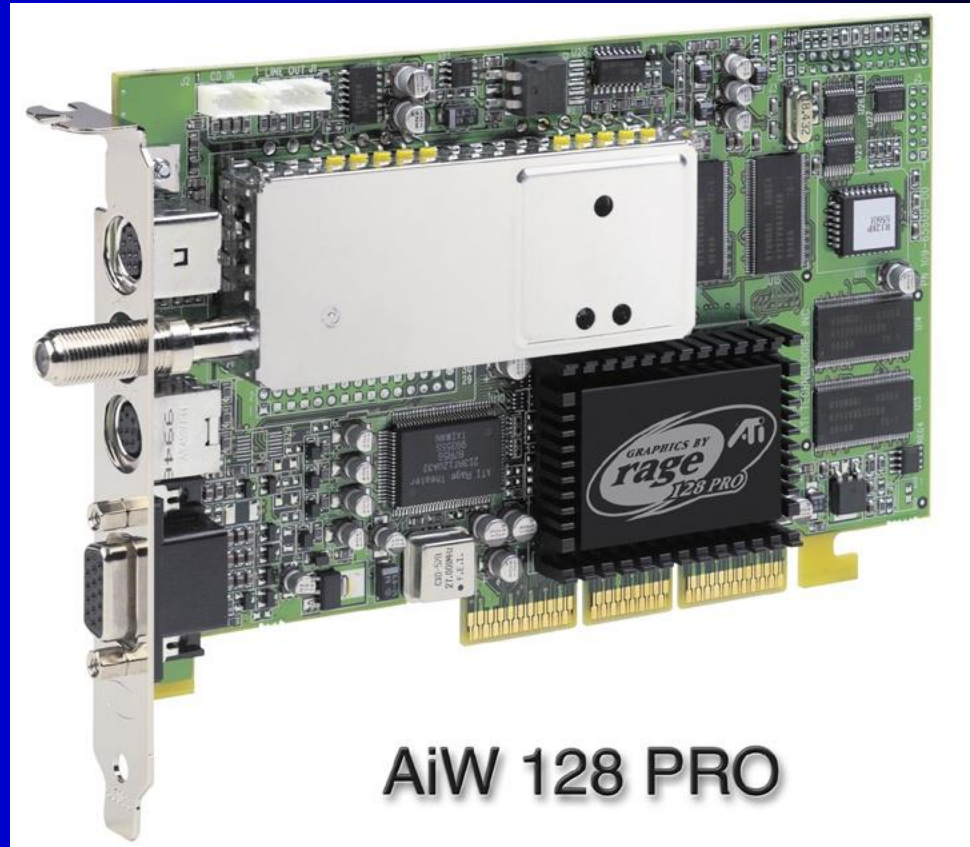
Video Output

- Video controller
 - on motherboard, or on expansion card
 - AGP ([accelerated graphics port technology](#))*
- Video memory (VRAM)
- Video CRT Display
 - uses raster scanning
 - horizontal retrace
 - vertical retrace
- Direct digital LCD monitors
 - no raster scanning required

* This link may change over time.

Sample Video Controller (ATI Corp.)

- 128-bit 3D graphics performance powered by RAGE™ 128 PRO
- 3D graphics performance
- Intelligent TV-Tuner with Digital VCR
- TV-ON-DEMAND™
- Interactive Program Guide
- Still image and MPEG-2 motion video capture
- Video editing
- Hardware DVD video playback
- Video output to TV or VCR



Memory

- ROM
 - read-only memory
- EPROM
 - erasable programmable read-only memory
- Dynamic RAM (DRAM)
 - inexpensive; must be refreshed constantly
- Static RAM (SRAM)
 - expensive; used for cache memory; no refresh required
- Video RAM (VRAM)
 - dual ported; optimized for constant video refresh
- CMOS RAM
 - complimentary metal-oxide semiconductor
 - system setup information
- See: [Intel platform memory](#) (Intel technology brief: link address may change)

Input-Output Ports

- USB (universal serial bus)
 - intelligent high-speed connection to devices
 - up to 12 megabits/second
 - USB hub connects multiple devices
 - *enumeration*: computer queries devices
 - supports *hot* connections
- Parallel
 - short cable, high speed
 - common for printers
 - bidirectional, parallel data transfer
 - Intel 8255 controller chip

Input-Output Ports (cont)

- Serial
 - RS-232 serial port
 - one bit at a time
 - uses long cables and modems
 - 16550 UART (universal asynchronous receiver transmitter)
 - programmable in assembly language

Device Interfaces

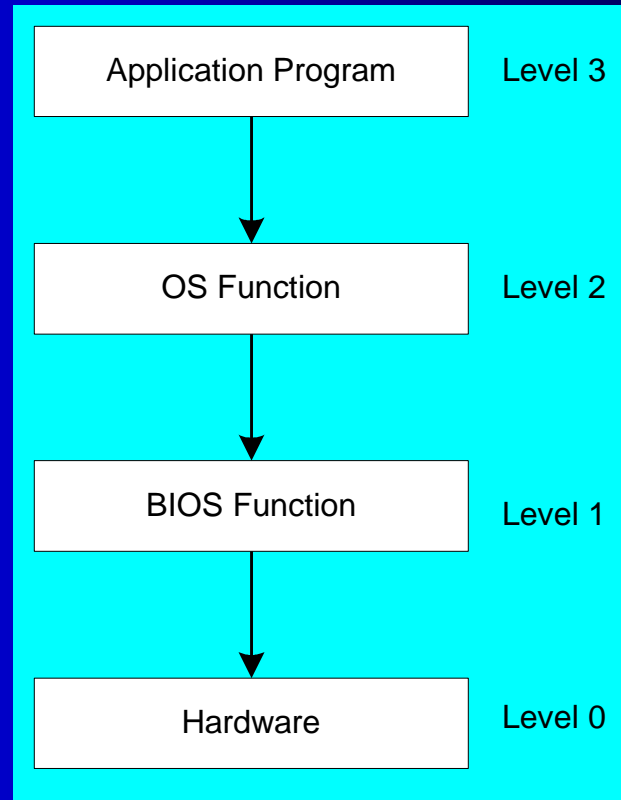
- ATA host adapters
 - intelligent drive electronics (hard drive, CDROM)
- SATA (Serial ATA)
 - inexpensive, fast, bidirectional
- FireWire
 - high speed (800 MB/sec), many devices at once
- Bluetooth
 - small amounts of data, short distances, low power usage
- Wi-Fi (wireless Ethernet)
 - IEEE 802.11 standard, faster than Bluetooth

Levels of Input-Output

- Level 3: High-level language function
 - examples: C++, Java
 - portable, convenient, not always the fastest
- Level 2: Operating system
 - Application Programming Interface (API)
 - extended capabilities, lots of details to master
- Level 1: BIOS
 - drivers that communicate directly with devices
 - OS security may prevent application-level code from working at this level

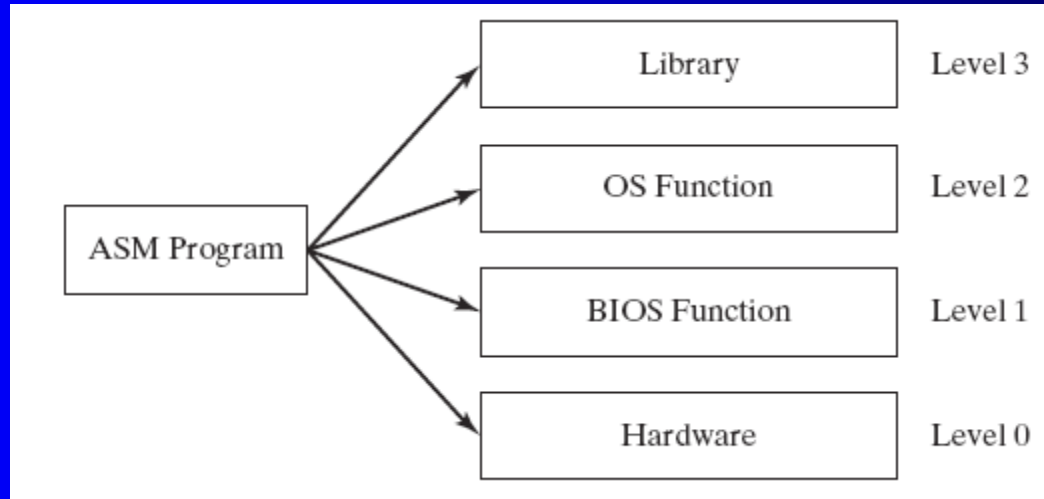
Displaying a String of Characters

When a HLL program displays a string of characters, the following steps take place:



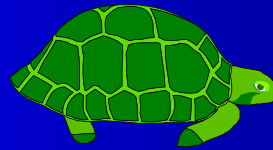
Programming levels

Assembly language programs can perform input-output at each of the following levels:



Summary

- Central Processing Unit (CPU)
- Arithmetic Logic Unit (ALU)
- Instruction execution cycle
- Multitasking
- Floating Point Unit (FPU)
- Complex Instruction Set
- Real mode and Protected mode
- Motherboard components
- Memory types
- Input/Output and access levels



42 69 6E 61 72 79

What does this say?