Assembly Programming (Chapter2)

CSE3030

Prof. Youngjae Kim

Chapter 2: X86 Processor Architecture

- Assembly language is the ideal software tool for communicating directly with a machine.
  - Assembly programmers must be familiar with the processor's internal architecture and capabilities.
  - At least, they need to understand basic operations that take place inside the processor when instructions are executed.
- What will we mainly cover in this chapter?
  - We will discuss how programs are loaded and executed by the operating system.
  - We will take a look at a sample motherboard layout, which will give some insight into the hardware environment of x86 systems.
  - We will discuss how layered input/output works between application programs and operating systems.
  - All of these are the hardware foundation to begin writing assembly language programs.

Chapter Overview

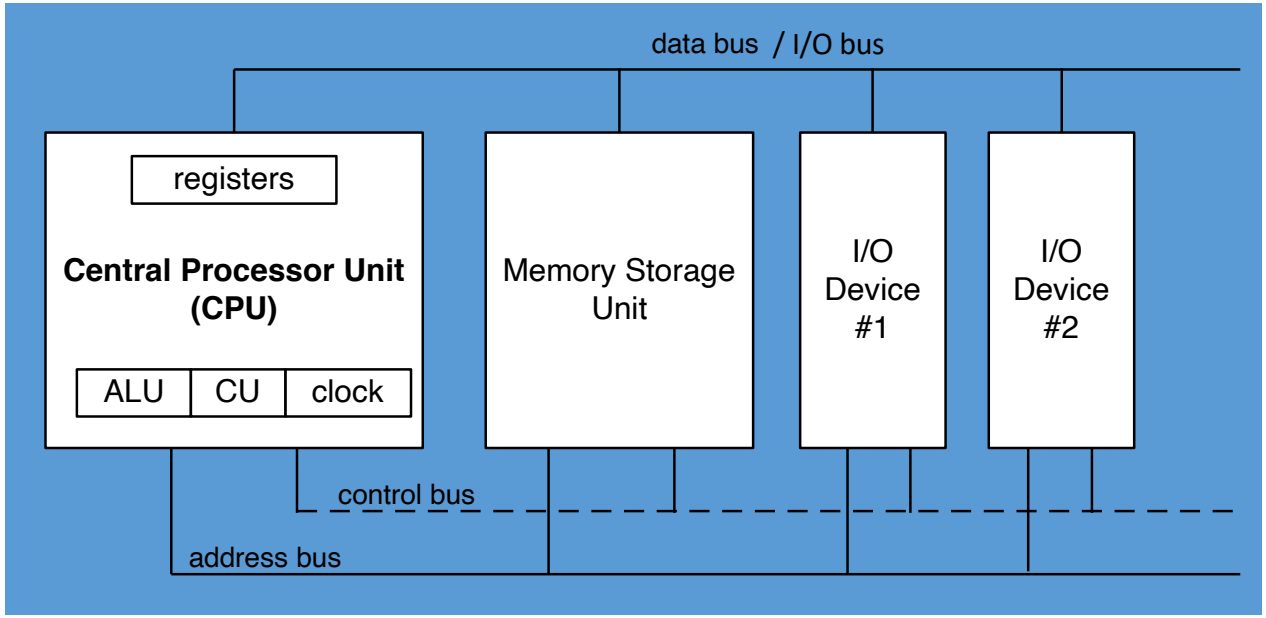
- General Concepts
- IA-32 Processor Architecture
- IA-32 Memory Management
- 64-bit Processors
- Components of an IA-32 Microcomputer
- Input-Output System

General Concepts

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

Basic Microcomputer Design

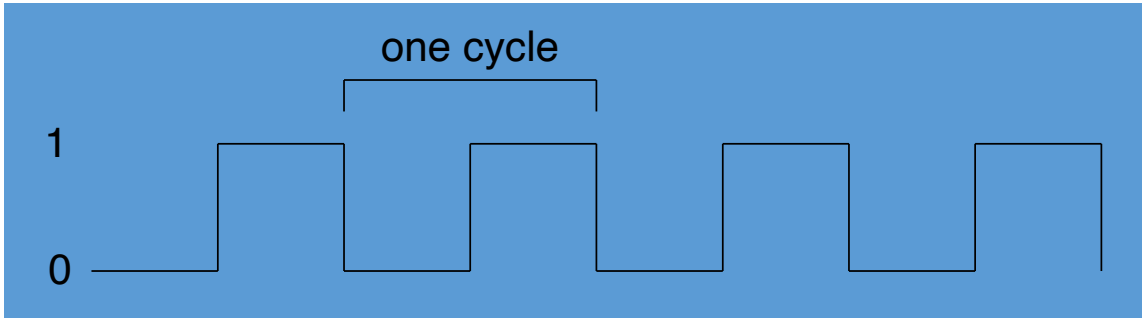
- Clock synchronizes CPU operations
- Control unit (CU) coordinates sequence of execution steps
- ALU performs arithmetic and bitwise processing



Block diagram of a microcomputer

Clock

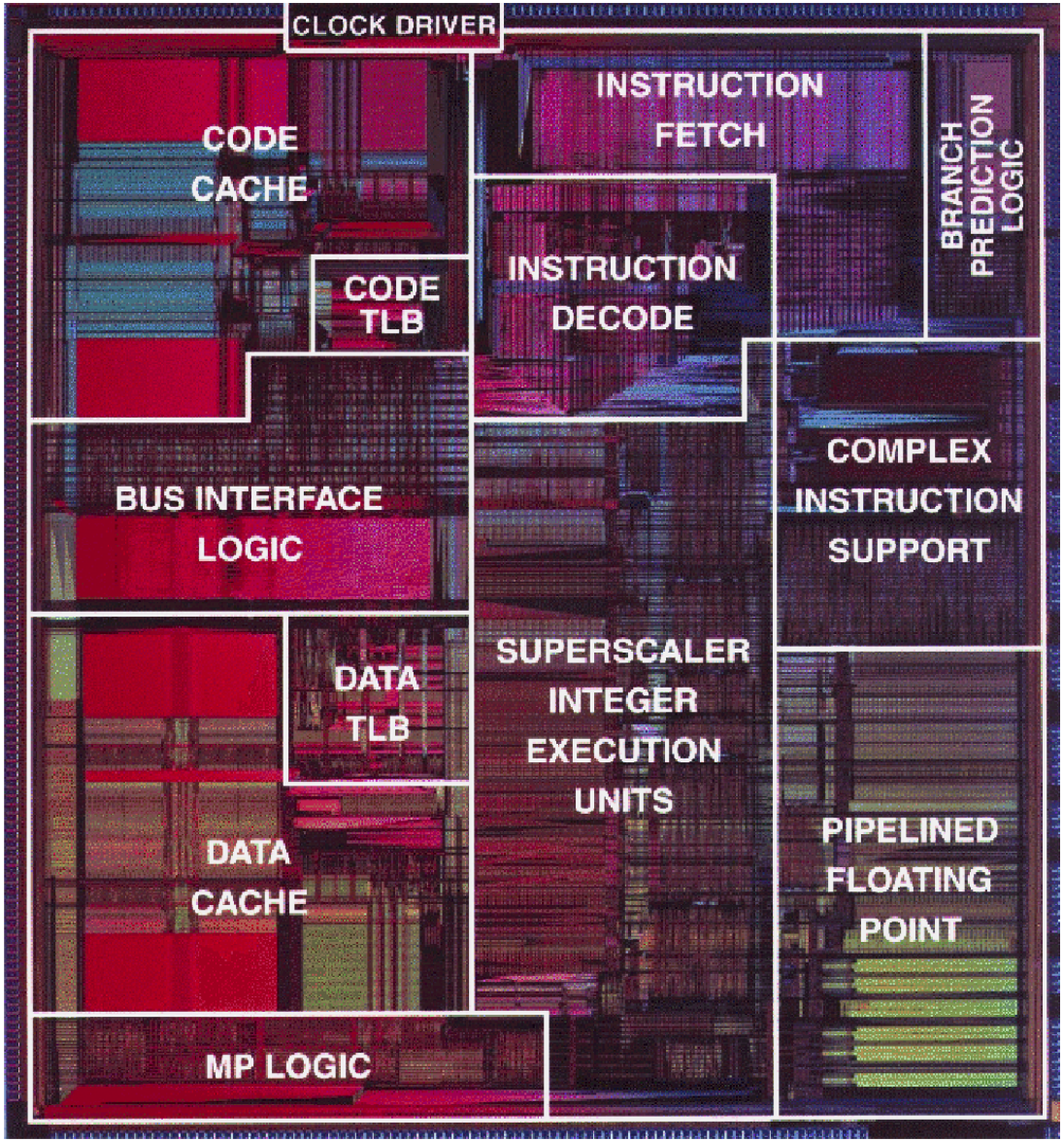
- synchronizes all CPU and BUS operations
- Machine (clock) cycle measures time of a single operation
- Clock is used to trigger events



What's Next

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

Example: Intel Pentium CPU



Operations inside the Computers

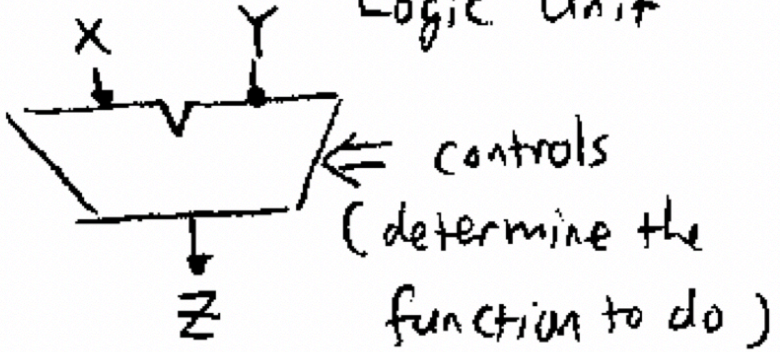
arithemetic : + (−, x, ÷, modular)

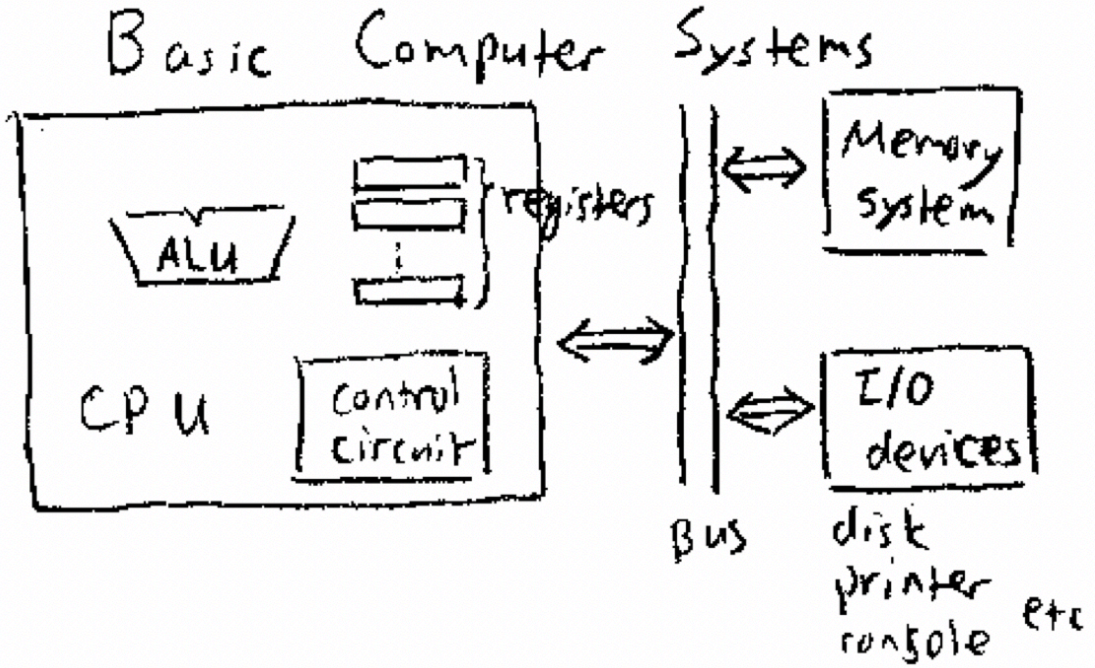
logical : NOT, AND, OR, ...

shift, rotation etc.

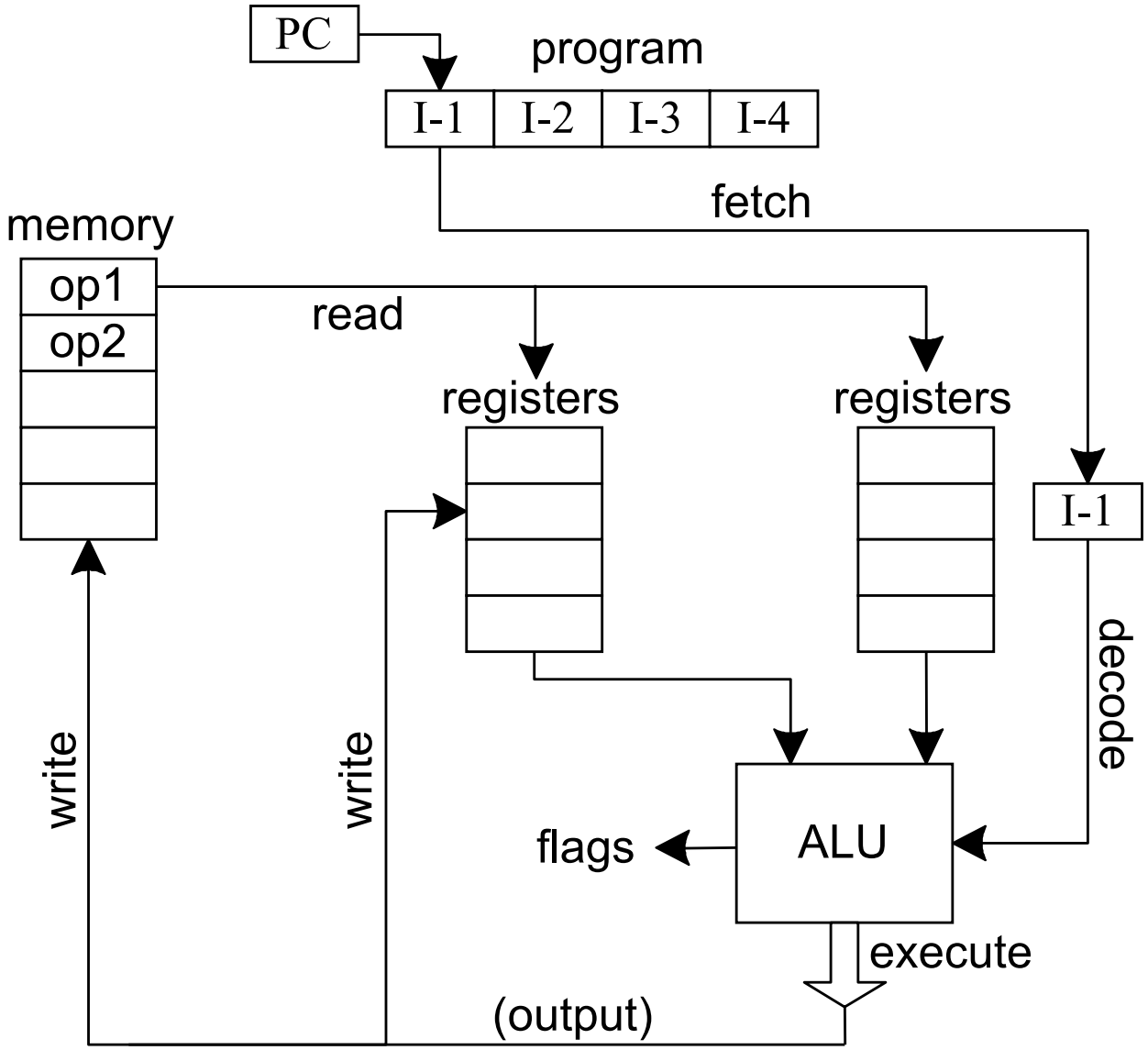
mostly data movements

ALU : Arithemetic Logic Unit





Instruction Execution Cycle

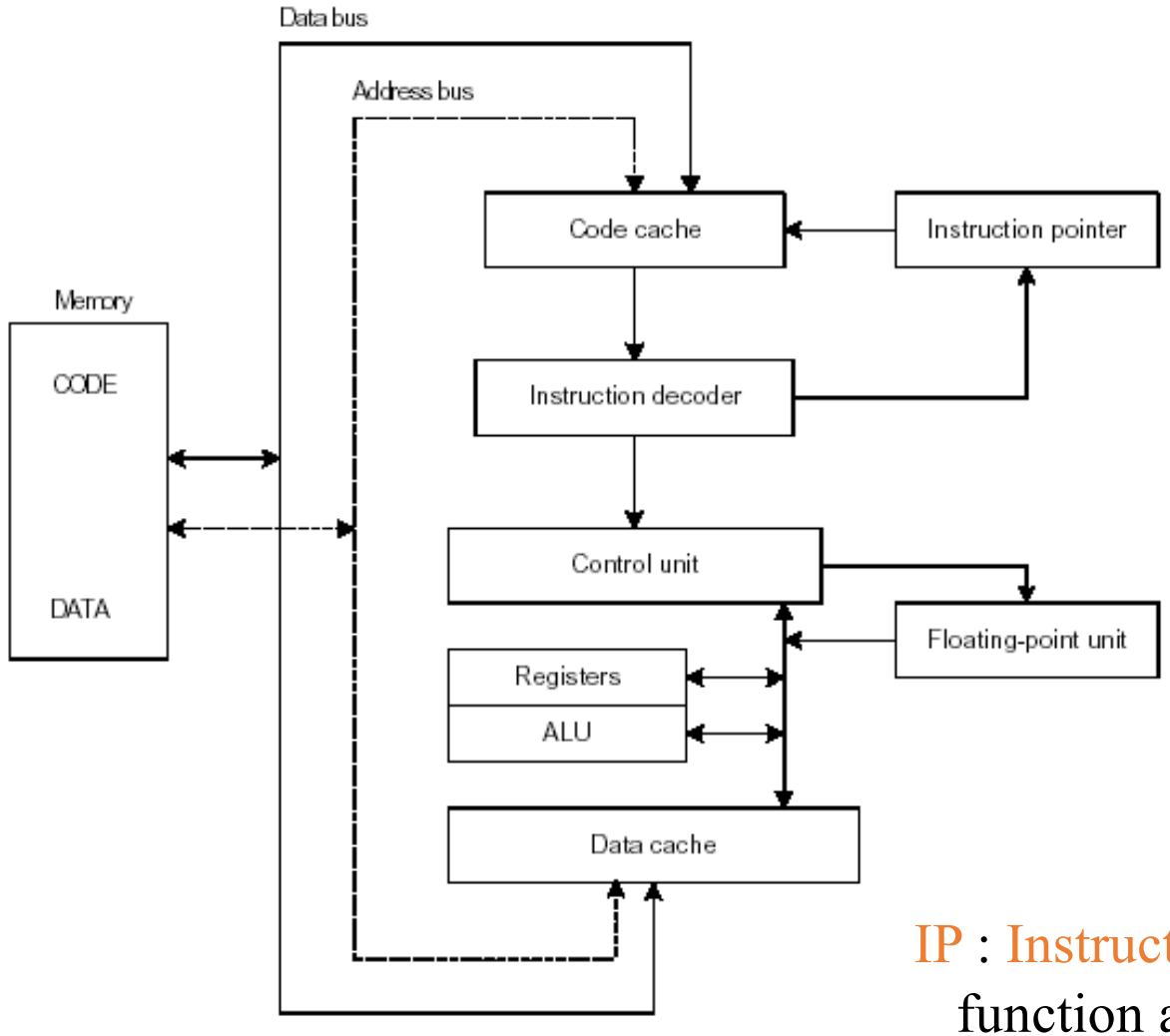


- Fetch
- Decode
- Fetch operands
- Execute
- Store output

PC: Program Counter
ALU: Arithmetic Logic Unit

Simplified CPU Block Diagram

Figure 2–2 Simplified Pentium CPU Block Diagram.



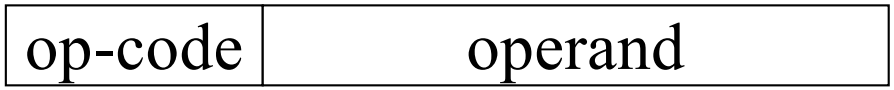
- Fetch
- Decode
- Fetch operands
- Execute
- Store output

IP : Instruction Pointer. The same function as the Program Counter

ALU: Arithmetic Logic Unit

Machine Instructions

- Statements what the CPU do.
- Generic Format



  - Op-code : operation code.
    Identifies the function to do by the CPU.
  - Operand : data or the location of the data in memory.

- An example : 1011 0000 0000 0101
  (1011 0000 : op-code, 0000 0101 : operand)
  - Move an 8 bit number (= 5) to the AL register.
  - "mov al, 5"

- A program : a sequence of machine instructions to do a specific job.

Symbolization of Machine Instructions

- Assembly Language Instructions
  - A machine instruction

$$\underbrace{1011\ 0000}_{\text{op-code}}\ \underbrace{0000\ 0101}_{\text{operand}}$$

  - An assembly language instruction

```
mov     al, 5
```

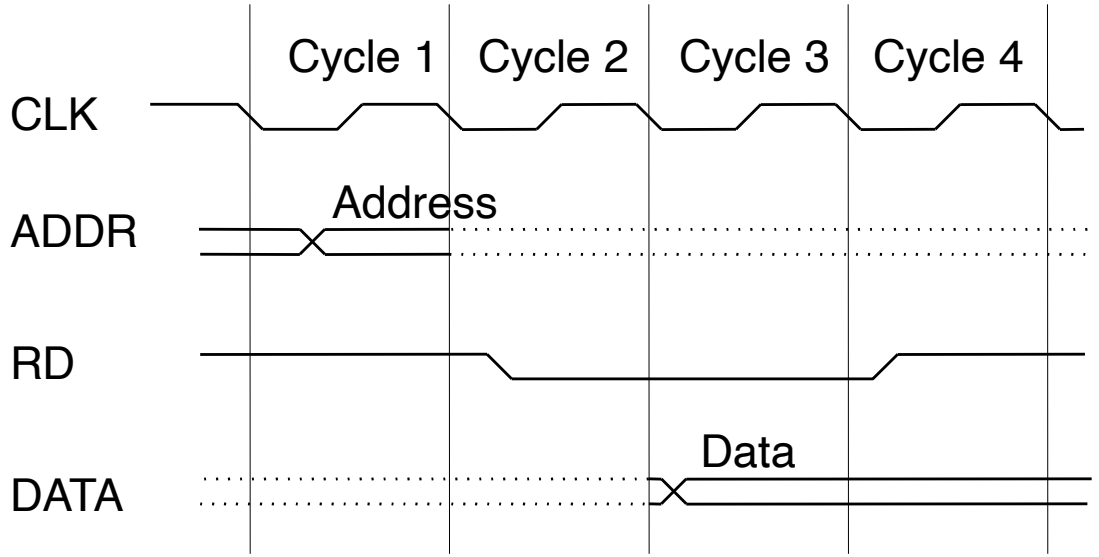
    - Symbolization of the machine instruction.
      - Mnemonic : mov
      - Operands : al, 5

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 (changing 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



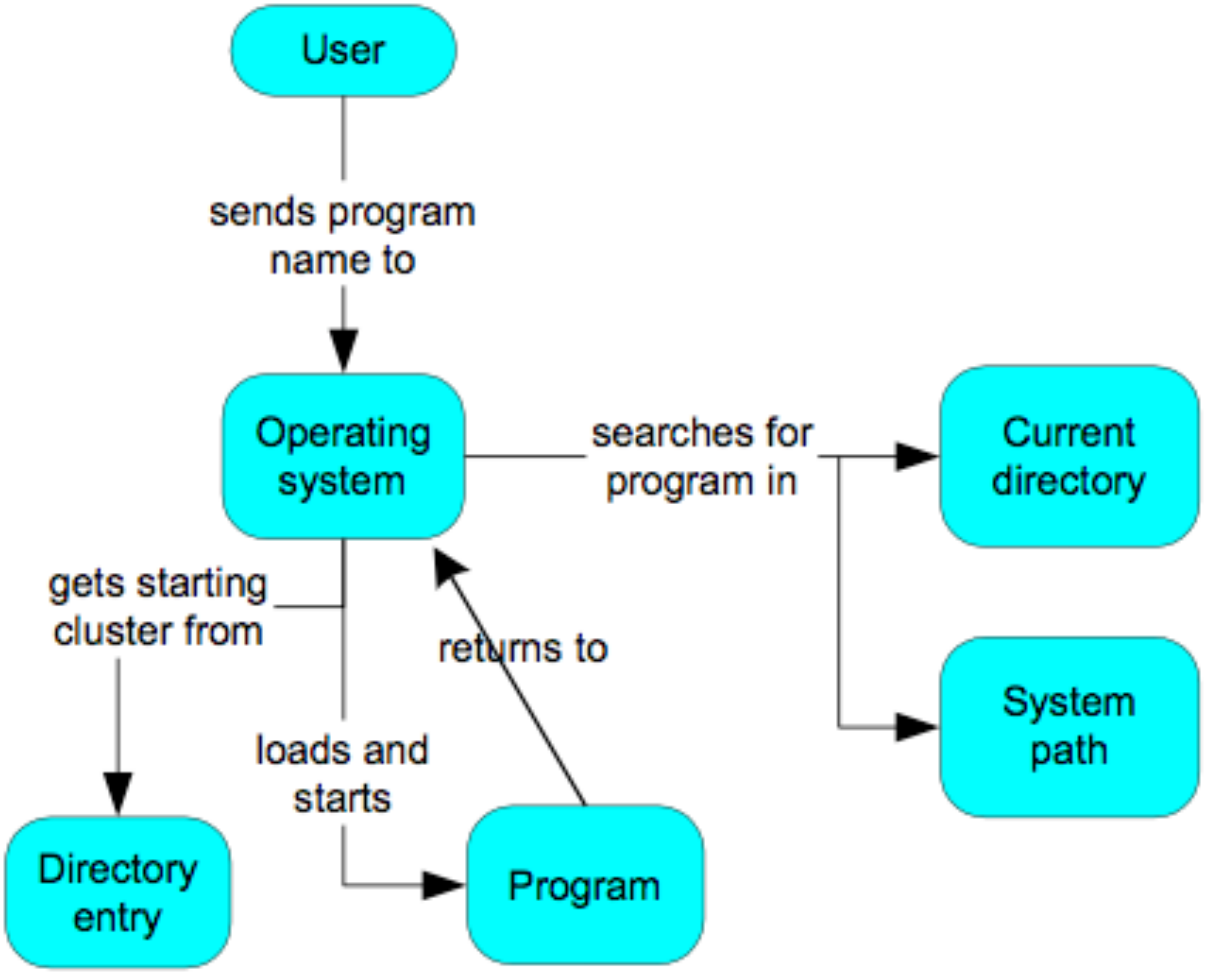
Cache Memory

- 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.

Multitasking

- OS can run multiple programs at the same time.
- Multiple threads of execution within the same program
- Scheduler utility assigns a given amount of CPU time to each running program.
- Rapid switching of tasks
  - gives illusion that all programs are running at once
  - the processor must support task switching.

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
  - A special case of Protected mode
  - each program has its own 8086 computer

Modes of Operation

- Real-Address Mode
  - Real-address mode implements the programming environment of an early Intel processor with a few extra features, such as the ability to switch into other modes. This mode is useful if a program requires direct access to system memory and hardware devices.

- System Management Mode (SMM)
  - SMM provides an OS with a mechanism for implementing functions such as power management and system security. These functions are useful implemented by computer manufacturers who customize the processor for a particular system setup.

Modes of Operation

- Protected Mode
  - Protected mode is the native state of the processor, in which all instructions and features are avaialble. Programs are given separate memory areas named segments, and the processor provents programs from referencing memory outide their assigned segments.

- Virtual-8086 Mode
  - While in protected mode, the processor can directly execute real-address mode software such as MS-DOS programs in a safe environment. In other words, if a program crashes or attempts to write data into the system memory area, it will not affect other programs running at the same time. A mordern OS can execute multiple separate virtual-8086 sessions at the same time.

Basic Execution Environment

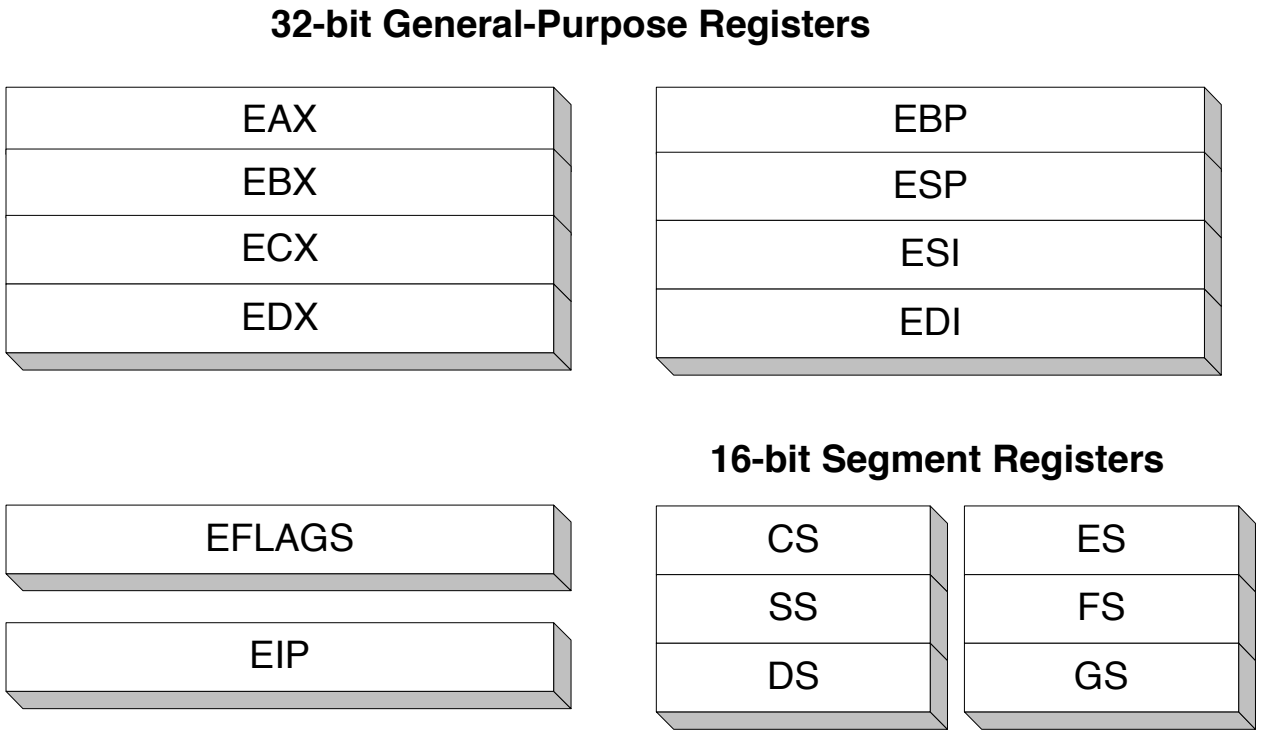
- Addressable memory
- General-purpose registers
- Index and base registers
- Specialized register uses
- Status flags
- Floating-point, MMX, XMM registers

Addressable Memory

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

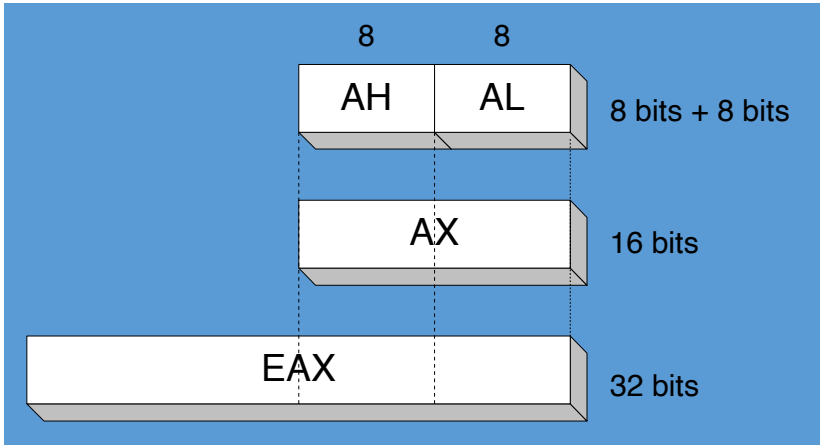
General-Purpose Registers

- Named storage locations inside the CPU, optimized for speed.



Accessing Parts of Registers

- 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
  - ECX – loop counter
  - ESP – stack pointer
  - ESI, EDI – index registers
  - EBP – extended frame pointer (stack)
- Segment
  - CS – code segment
  - DS – data segment
  - SS – stack segment
  - ES, FS, GS - additional segments

Some Specialized Register Uses (2 of 2)

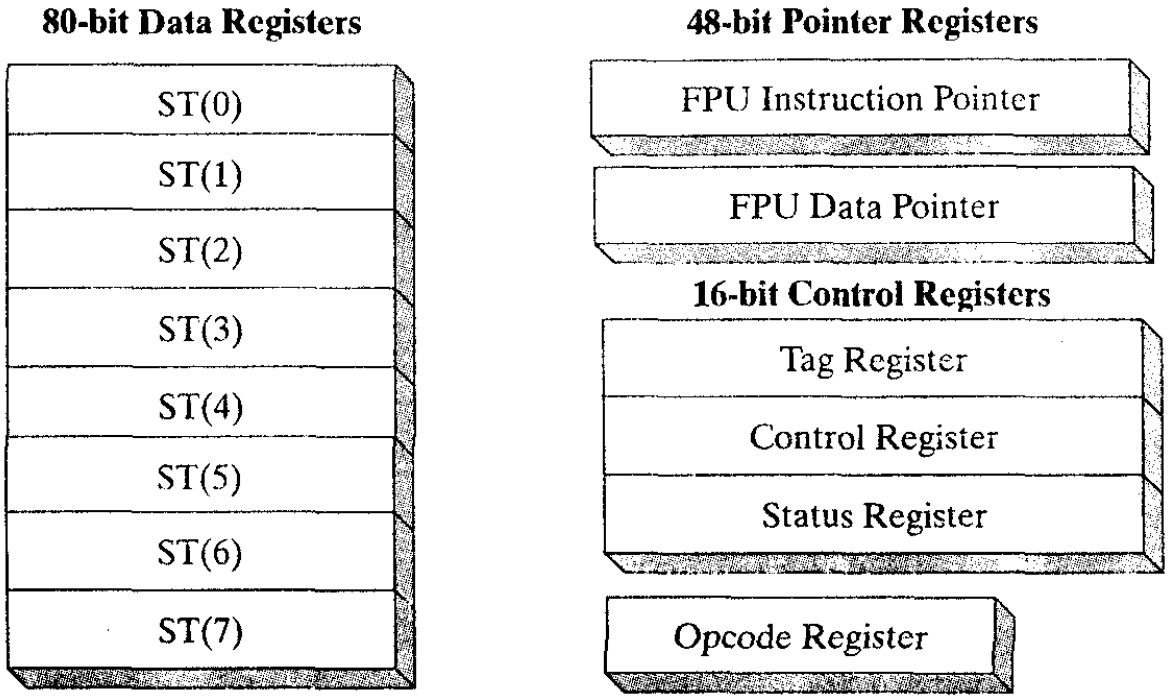
- EIP – instruction pointer
- EFLAGS
  - status and control flags
  - each flag is a single binary bit

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



- Eight 64-bit MMX registers (used for multimedia processing)
- Eight 128-bit XMM registers for single-instruction multiple-data (SIMD) operations

Intel Microprocessor History

- Early Intel Microprocessors
  - Intel 8080
    - 64K addressable RAM, 8-bit registers
    - CP/M operating system, S-100 BUS architecture
    - 8-inch floppy disks!
  - Intel 8086/8088
    - IBM-PC Used 8088
    - 1 MB addressable RAM, 16-bit registers
    - 16-bit data bus (8-bit for 8088)
    - separate floating-point unit (8087)

Intel Microprocessor History (Cont')

- The IBM-AT
  - Intel 80286
    - 16 MB addressable RAM. Protected memory.
    - Several times faster than 8086
    - Introduced IDE bus architecture. 80287 FPU.
- Intel IA-32 Family
  - Intel386
    - 4 GB, 32-bit registers, paging(virtual memory)
  - Intel486
    - Instruction pipelining
  - Pentium
    - Superscalar, 32-bit address bus, 64-bit internal data path.

Intel Microprocessor History (Cont')

- Intel P6 Family
  - Pentium Pro
    - Advanced optimization techniques in microcode
  - Pentium II
    - MMX (multimedia) instruction set
  - Pentium III
    - SIMD (streaming extensions) instructions
  - Pentium 4
    - NetBurst micro-architecture, tuned for multimedia

Intel Microprocessor History (Cont')

- 64-bit Processors
  - Intel64
    - 64-bit linear address space
    - Intel: Pentium Extreme, Xeon, Celeron D, Pentium D, Core 2, and Core i7
  - IA-32e Mode
    - Compatibility mode for legacy 16- and 32-bit applications
    - 64-bit Mode uses 64-bit addresses and operands

Intel Microprocessor History (Cont')

- Intel Technologies
  - HyperThreading technology
    - two tasks execute on a single processor at the same time
  - Dual Core processing
    - multiple processor cores in the same IC package
    - each processor has its own resources and communication path with the bus
  - Currently
    - Pentium & Celeron – dual core
    - Core 2 Duo - 2 processor cores
    - Core 2 Quad - 4 processor cores
    - Core i7 – 4 processor cores

CISC and RISC

- CISC – complex instruction set
  - Large instruction set
  - High-level operations
  - Requires microcode interpreter
  - Examples: Intel 80x86 family
- RISC – reduced instruction set
  - Simple, atomic instructions
  - Small instruction set
  - Directly executed by hardware
  - Examples:
    - ARM (Advanced RISC Machines)
    - DEC Alpha (now Compaq)

What's Next

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

IA-32 Memory Management

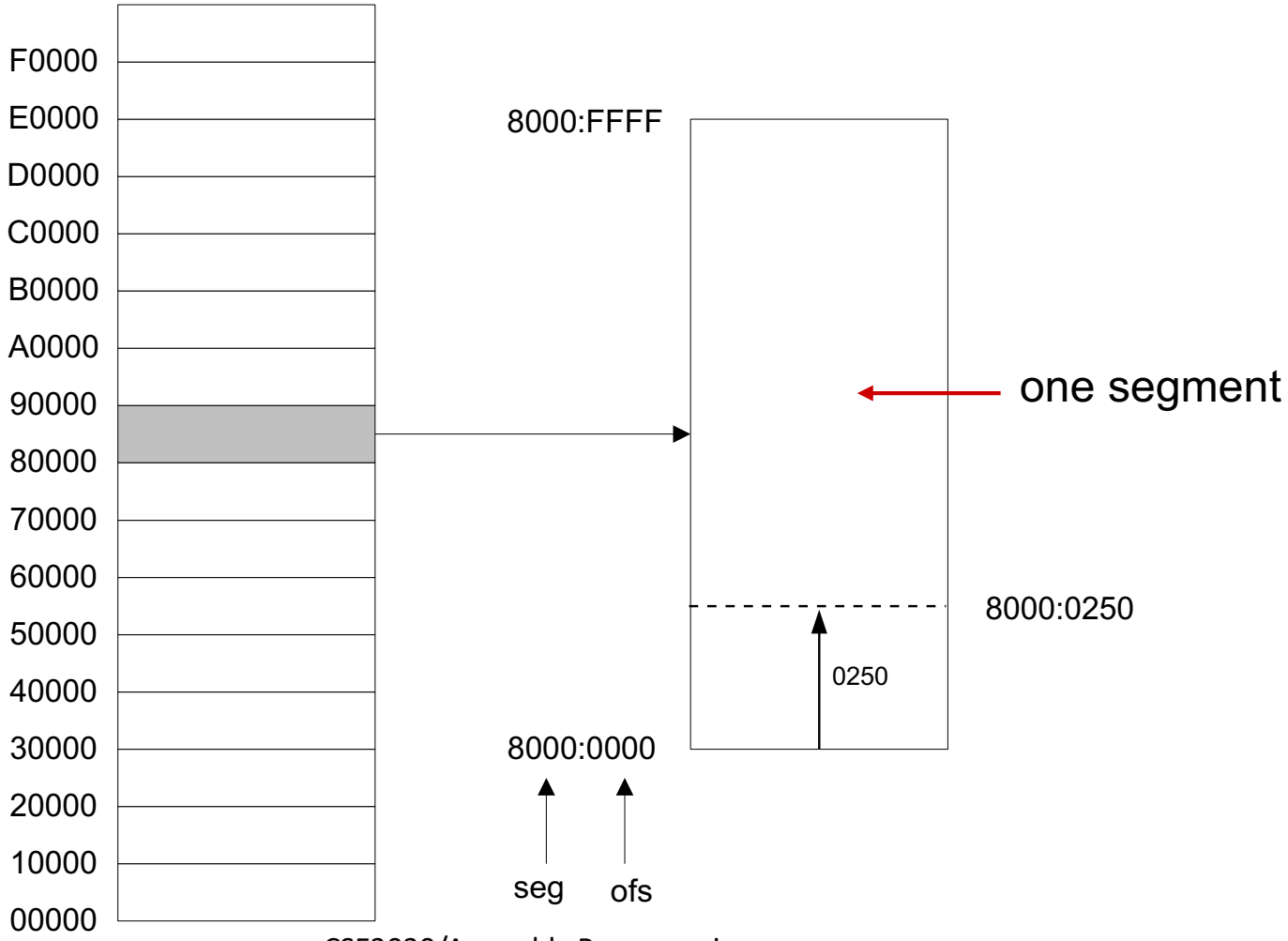
- Real-address mode
- Calculating linear addresses
- Protected mode
- Multi-segment model
- Paging

Real-Address mode

- 1 MB RAM maximum addressable
- Application programs can access any area of memory
- Single tasking
- Supported by MS-DOS operating system, DOS mode of Win95 and 98.

Segmented Memory

- Segmented memory addressing: absolute (linear) address is a combination of a 16-bit segment value added to a 16-bit offset.



Calculating Linear Addresses

- Given a segment address, multiply it by 16 (add a hexadecimal zero), and add it to the offset
- Example 1: Convert 08F1:0100 to a linear address.
- Example 2: Convert 028F:0030 to a linear address.
- Example 3: What segment addresses correspond to the linear address 28F30h?

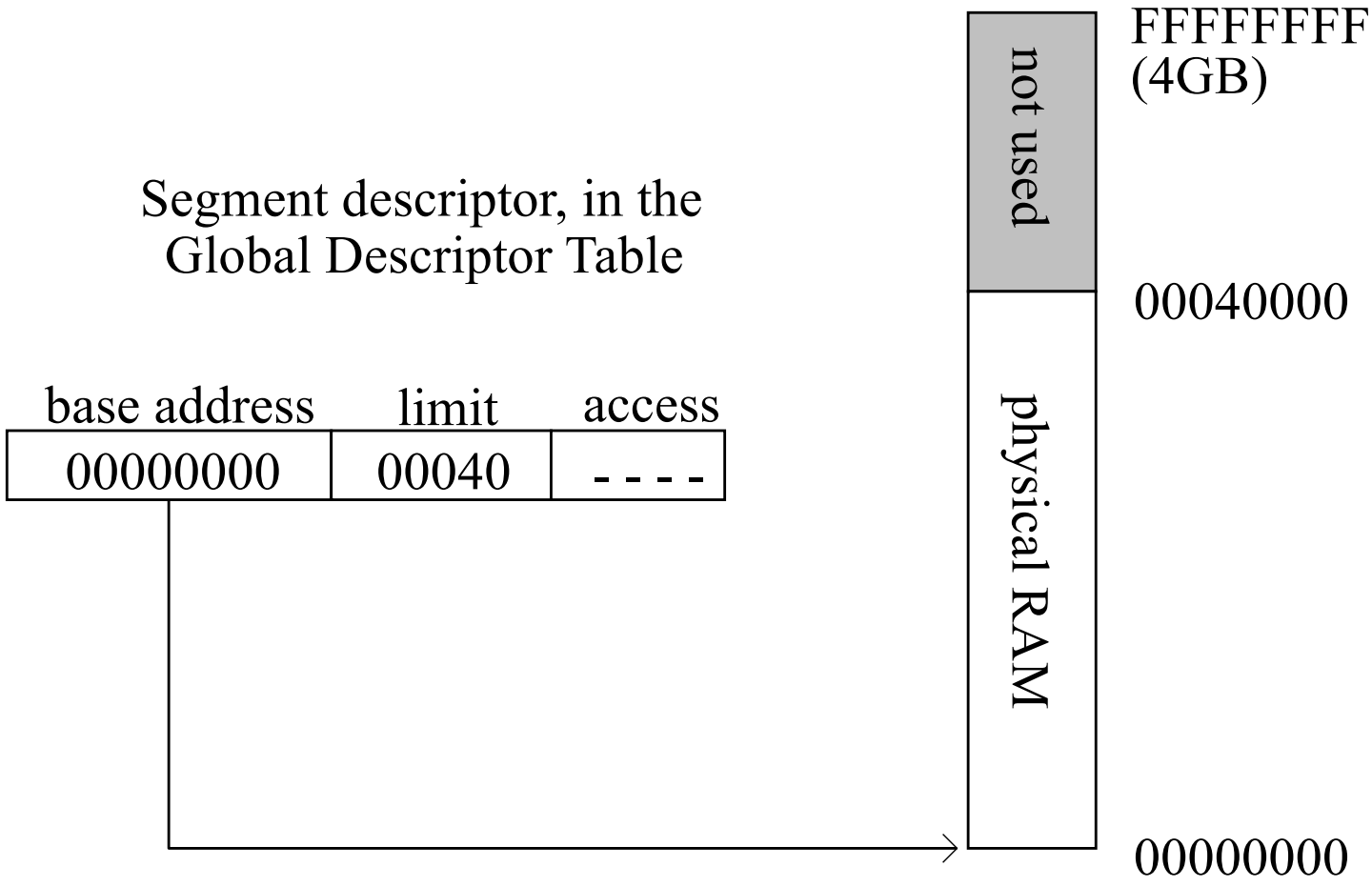
```
28F0:0030, 28F3:0000, 28B0:0430, . . .
```

Protected Mode

- 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
- Segment descriptor tables
- Program structure
  - code, data, and stack areas
  - CS, DS, SS segment descriptors
  - global descriptor table (GDT)
- MASM Programs use the Microsoft flat memory model

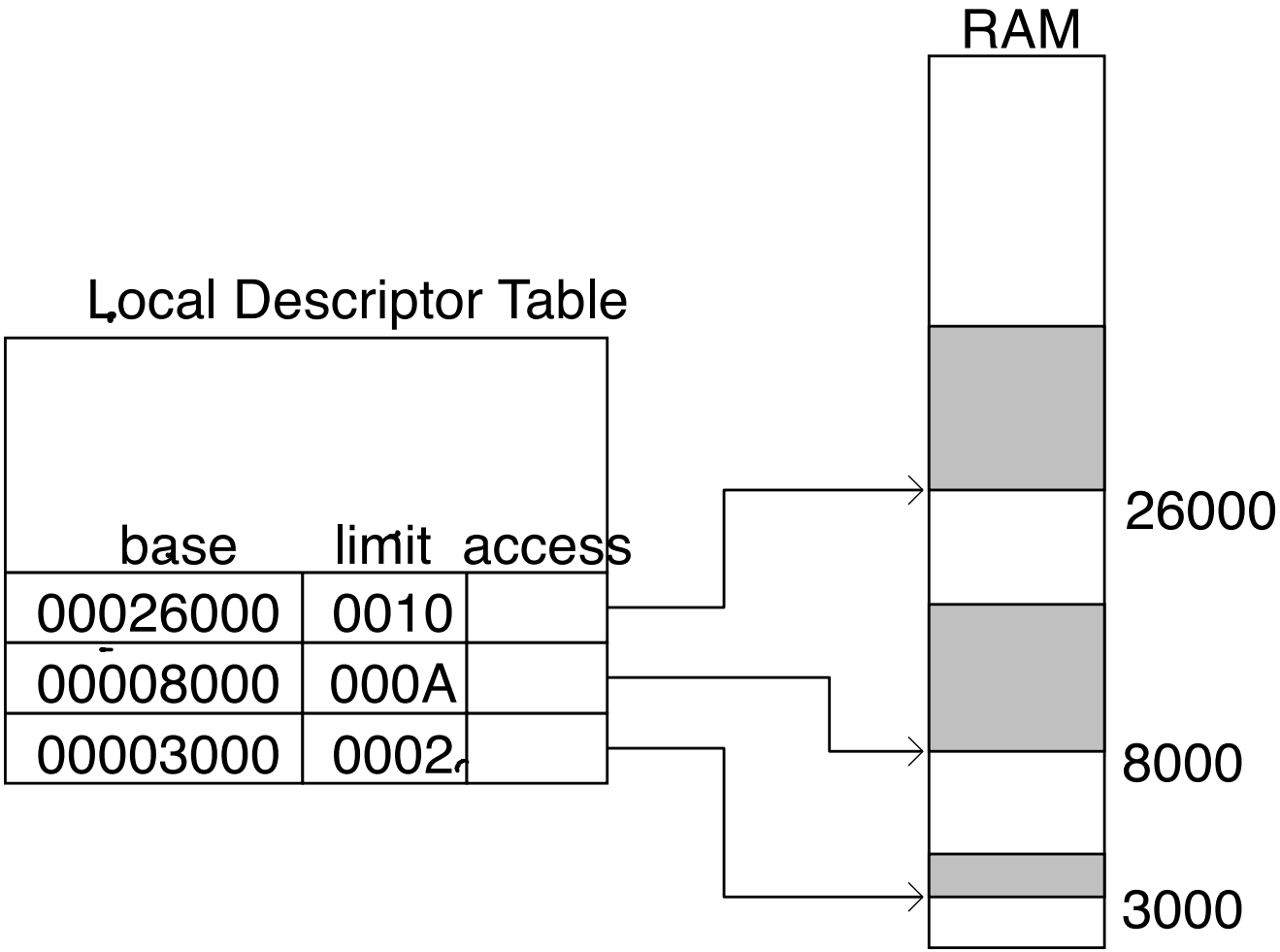
Flat Segment Model

- Single global descriptor table (GDT).
- All segments mapped to entire 32-bit address space



Multi-Segment Model

- Each program has a local descriptor table (LDT)
  - holds descriptor for each segment used by the program



Paging

- Supported directly by the CPU
- Divides each segment into 4096-byte blocks called pages.
- Sum of all programs can be larger than physical memory.
- Part of running program is in memory, part is on disk.
- Virtual memory manager (VMM) – OS utility that manages the loading and unloading of pages.
- Page fault – issued by CPU when a page must be loaded from disk.

What's Next

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

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

What's Next

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

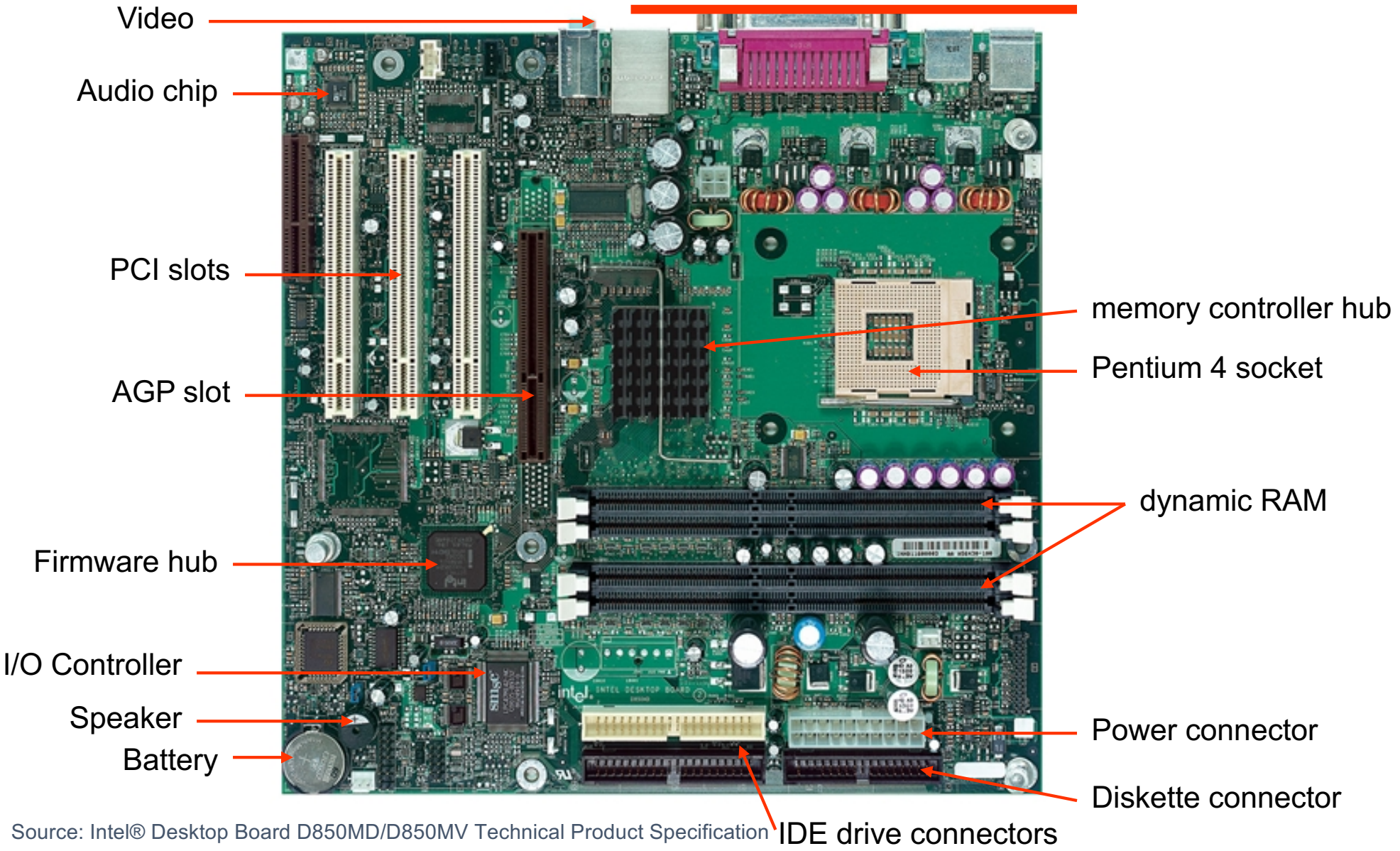
Components of an IA-32 Microcomputer

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

Motherboard

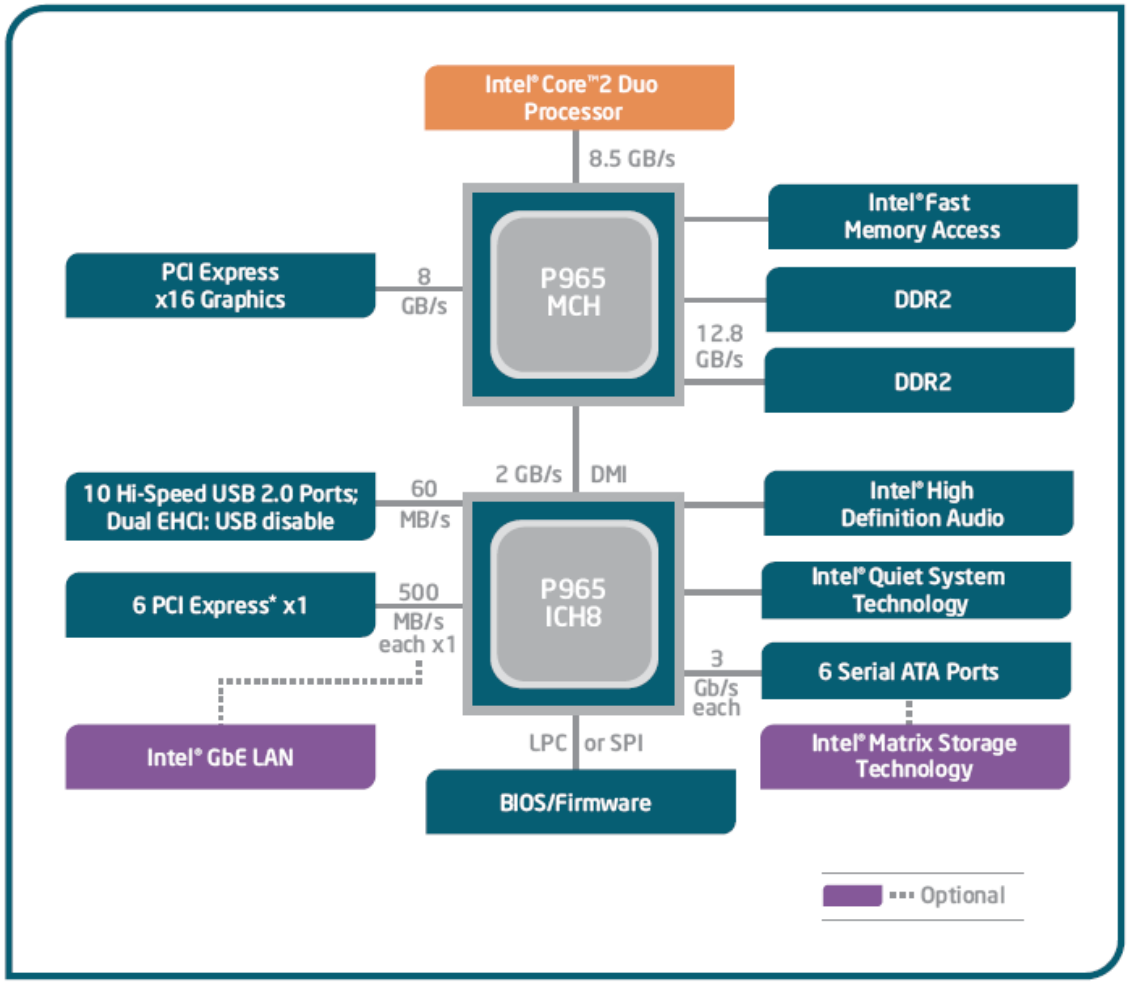
- CPU socket
- External cache memory slots
- Main memory slots
- BIOS chips
- Sound synthesizer chip (optional)
- Video controller chip (optional)
- IDE, parallel, serial, USB, video, keyboard, joystick, network, and mouse connectors
- PCI bus connectors (expansion cards)

Intel D850MD Motherboard



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

Intel 965 Express Chipset

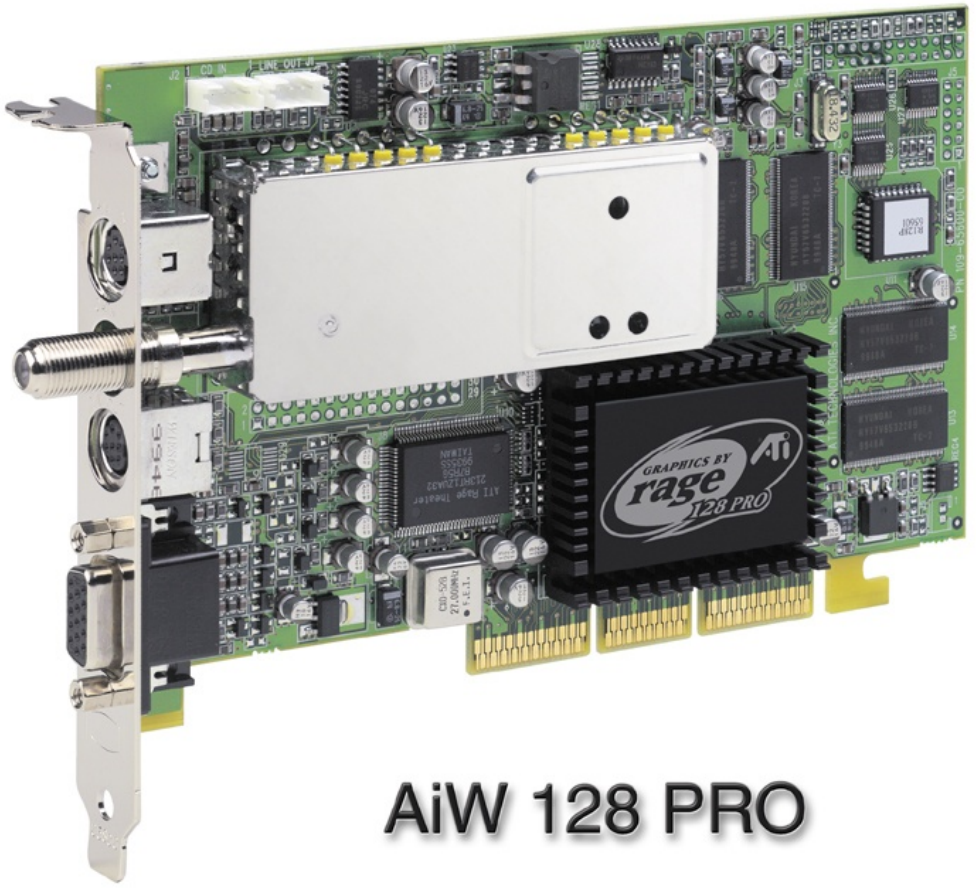


Video Output

- 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

Sample Video Controller (ATI Corp.)

- 128-bit 3D graphics performance powered by RAGE™ 128 PRO
- 3D graphics performance
- Intelligent TV-Tuner with Digital
- TV-ON-DEMAND™
- Interactive Program Guide
- Still image and MPEG-2 motion
- 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

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

What's Next

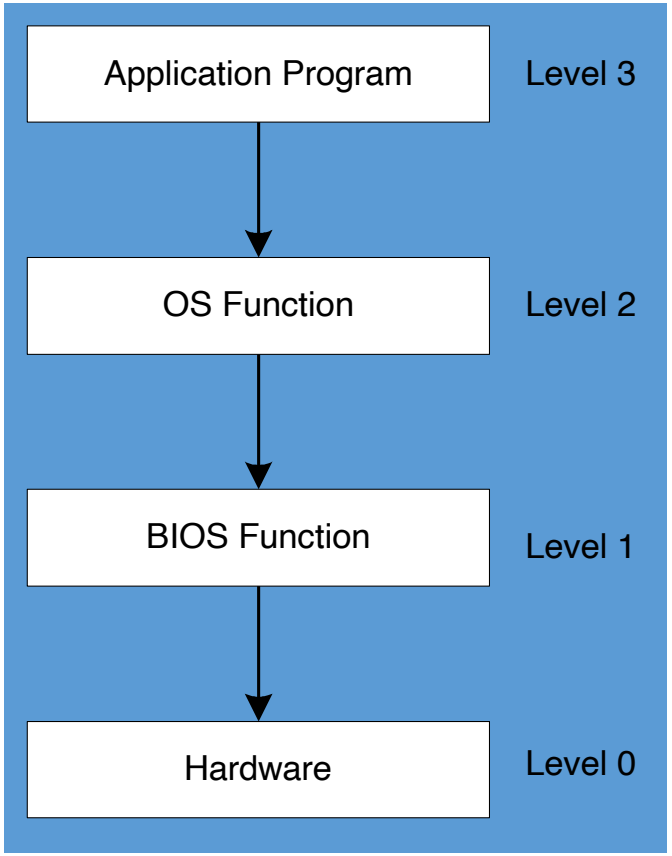
- General Concepts
- IA-32 Processor Architecture
- IA-32 Memory Management
- Components of an IA-32 Microcomputer
- **Input-Output System**

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

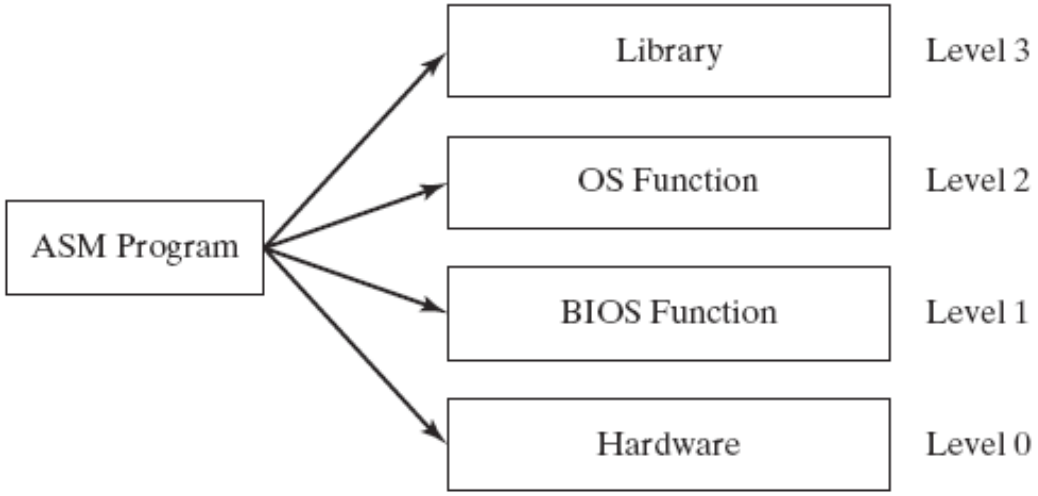
Displaying a String of Characters

- 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