

# CMPE 310 Systems Design and Programming

## L2: Chapter 1 – *Introduction to the Microprocessor and Computer*

UMBC

AN HONORS UNIVERSITY IN MARYLAND

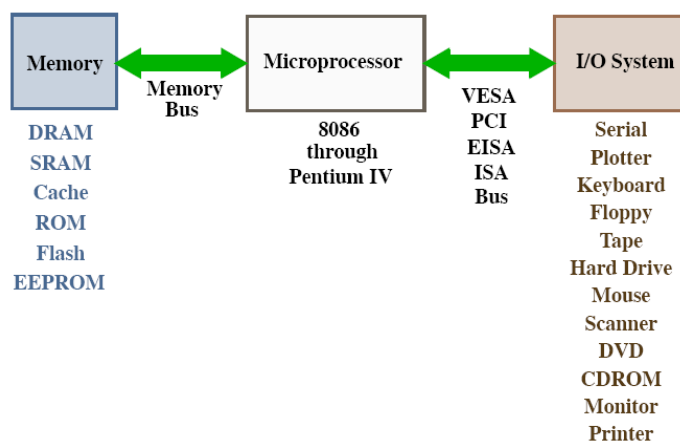
## L2 Objectives

- \* Describe the purpose of the major components of a computer system
- \* Describe the purpose of the three types of buses
- \* Describe the role of the CPU
- \* List the major components of the CPU
- \* Describe the purpose of each of the major CPU components

# Architecture

- \* “Architecture” can refer to
  - \* High-level description of hardware; could be
    - \* Overall system
    - \* Microprocessor
    - \* Subsystem within processor
  - \* Operations available to programmer
    - \* Instruction set architecture
  - \* Other applications to computing (e.g., “software architecture”) we won’t discuss
- \* Commonly used to discuss functional units and how they work together

## Microprocessor Architecture



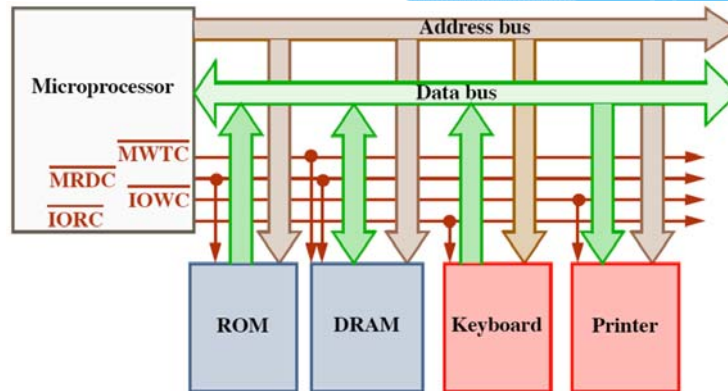
## Microprocessor Architecture

- \* The MPU communicates with memory and I/O using the system bus consisting of:
  - \* Address bus: unidirectional and carries memory and I/O addresses
  - \* Data bus: bidirectional; transfers binary data and instructions between MPU and memory and I/O
  - \* Control lines: Read and Write timing signals asserted by MPU

## Basic Bus Architecture

- \* **Address:**
  - \* If I/O, a value between 0000H and FFFFH is issued.
  - \* If memory, it depends on the architecture:
    - \* **20**-bits (8086/8088)
    - \* **24**-bits (80286/80386SX)
    - \* **25**-bits (80386SL/SLC/EX)
    - \* **32**-bits (80386DX/80486/Pentium)
    - \* **36**-bits (Pentium Pro/II/III)
- \* **Data:**
  - \* **8**-bits (8088)
  - \* **16**-bits (8086/80286/80386SX/SL/SLC/EX)
  - \* **32**-bits (80386DX/80486)
  - \* **64**-bits (Pentium/Pro/II/III)
- \* **Control:**
  - \* Most systems have at least 4 control bus connections (active low).
  - \* **MRDC**, **MWTC**, **IORC**, **IOWC**.

## Microprocessor Architecture



Microprocessor-Based System with Buses: Address, Data, and Control

## Basic Bus Architecture

### \* Bus Standards

- \* **ISA (Industry Standard Architecture): 8 MHz**
  - \* 8-bit (8086/8088)
  - \* 16-bit (80286-Pentium)
- \* **EISA: 8 MHz**
  - \* 32-bit (older 386 and 486 machines).
- \* **PCI (Peripheral Component Interconnect): 33 MHz**
  - \* 32-bit or 64-bit (Pentiums)
  - \* New: PCI Express and PCI-X 533 MTS
- \* **VESA (Video Electronic Standards Association): Runs at processor speed.**
  - \* 32-bit or 64-bit (Pentiums)
  - \* Only disk and video. Competes with the PCI but is not popular.

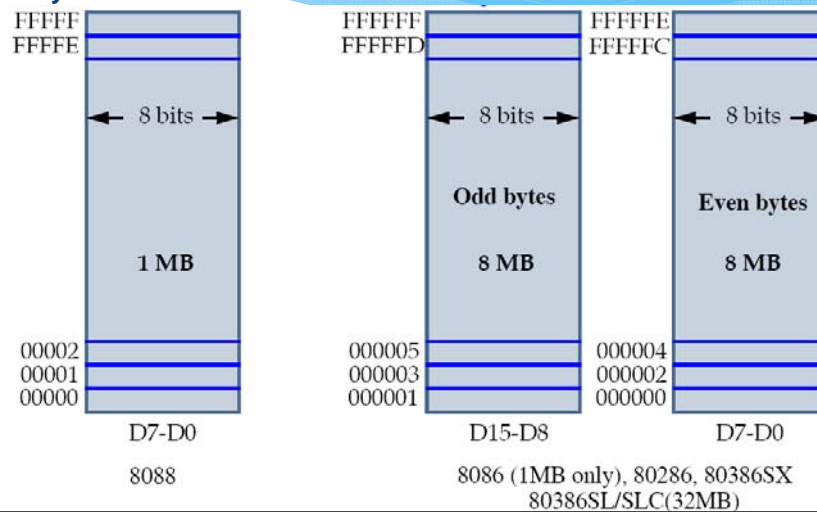
## Basic Bus Architecture

### \* Bus Standards

- \* **USB (Universal Serial Bus): 1.5 Mbps, 12 Mbps, 480 Mbps and now 5 Gbps.**
  - \* Newest systems.
  - \* Serial connection to microprocessor.
  - \* For keyboards, the mouse, modems and sound cards.
  - \* **To reduce system cost through fewer wires.**
- \* **AGP (Advanced Graphics Port): 66MHz**
  - \* Newest systems.
  - \* Fast parallel connection: Across 64-bits for 533MB/sec.
  - \* For video cards.
  - \* **Latest AGP 3.0 with peak bandwidth of 2.1GB/s.**
  - \* **To accommodate the new DVD (Digital Versatile Disk) players.**

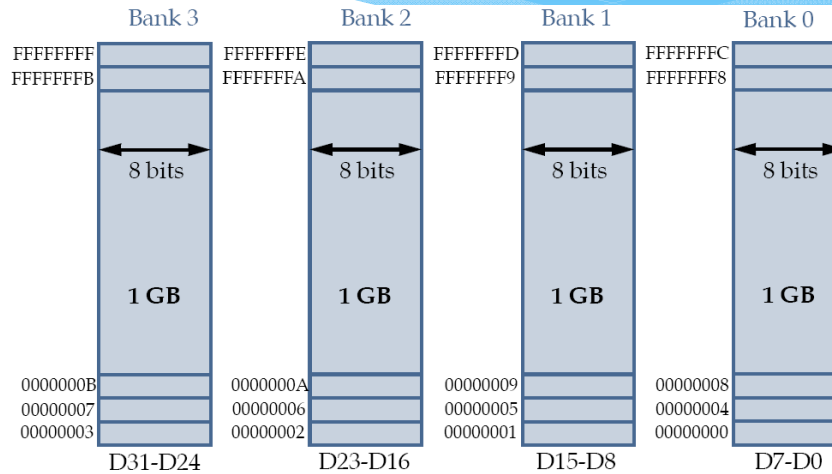
## Basic Memory Architecture

### \* Bank Layout



# Basic Memory Architecture

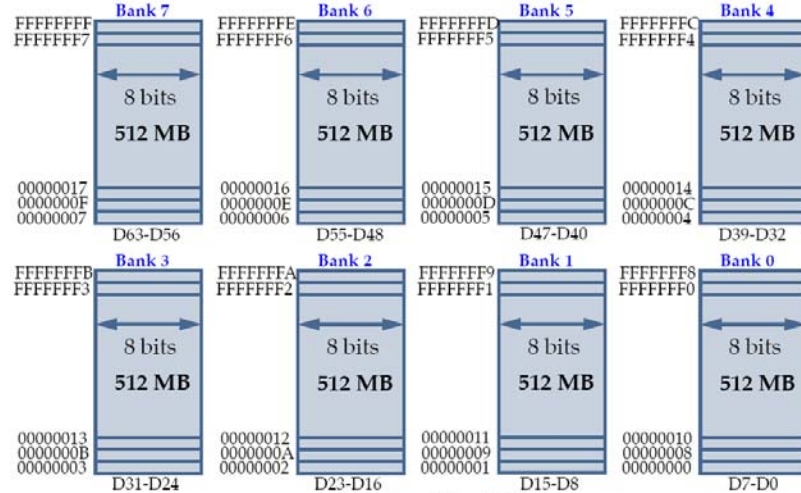
## \* Bank Layout



80386DX, 80486

# Basic Memory Architecture

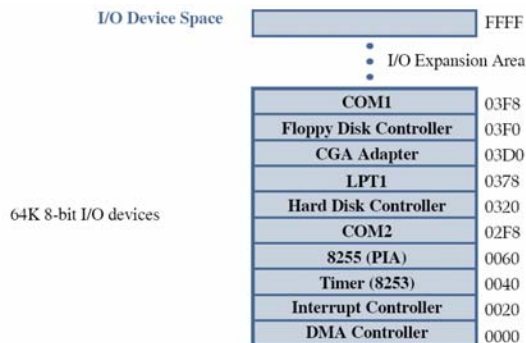
## \* Bank Layout



Pentium/Pro/II/III

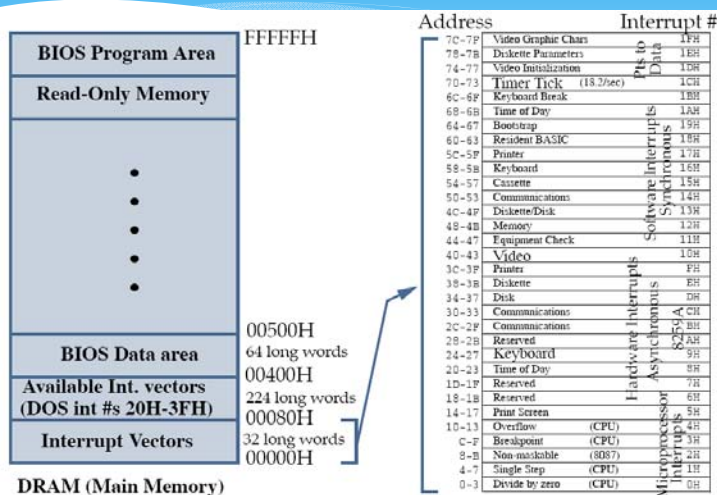
## I/O Space

- \* It is important to notice that these I/O addresses are NOT memory-mapped addresses on the 80x86 machines



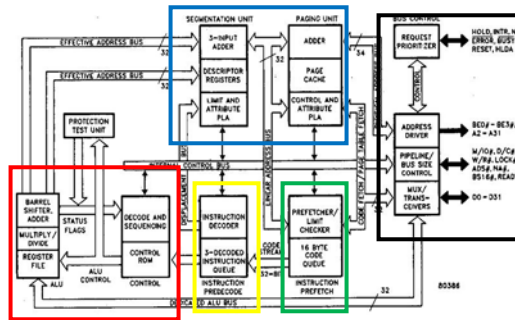
- \* Special instructions (IN/OUT) are used to communicate to the I/O devices

## Interrupt Vectors (DOS PC)



# Internal Architecture of the Intel Microprocessor

## \* High level hardware view



\* Parallel processing → high performance

\* Six functional units:

- \* Bus units
- \* Execution unit
- \* Segment unit
- \* Page unit
- \* Prefetch unit
- \* Decode unit

## Bus Interface Unit

- \* Responsible for
  - \* Fetching instruction
    - \* Operation to be executed
  - \* Reading and writing of data for memory
  - \* Input/output of data for input/output peripherals
- \* Information transfers over the bus
  - \* De-multiplexed bus
  - \* x86
    - \* 16-bit data bus
    - \* Real-mode: 20-bit address, 1M-byte physical address space
  - \* x386DX
    - \* 32-bit data bus
    - \* Real-mode: 20-bit address, 1M-byte physical address space
    - \* Protected-mode: 32-bit address bus, 4G-byte physical address space



## Execution Unit

- \* Responsible for executing instructions
- \* Element of the EU
  - \* Arithmetic/logic unit (ALU)
    - \* Performs the operation identified by the instruction: ADD, SUB, AND...
  - \* Flags register
    - \* Holds status and control information
  - \* General-purpose registers
    - \* Holds address or data information
  - \* Control ROM
    - \* Contains microcode sequences that define operations performed by machine instructions
  - \* Special multiply, and shift hardware
    - \* Accelerate multiply, divide, and rotate operations

## Operations of the Execution Unit

- \* Reads instructions from the instruction queue
- \* Accesses data
  - \* General purpose registers if necessary
  - \* Generates memory address of data storage locations in memory if necessary
  - \* Passes memory addresses to the segmentation and paging units and requests the bus unit to perform read or write bus cycles to access data operands in memory
- \* Performs the operation defined by the instruction on the selected data
- \* Tests the state of flags if necessary
  - \* Updates flag state based on instruction result

## Segmentation and Paging Unit

- \* Off-load memory-management and protection services from the bus unit
- \* Segmentation unit
  - \* Implements real-mode and protected-mode segmentation model
  - \* Contains general registers, segment registers, and instruction pointer
  - \* Holds address and data operand information
- \* Segmentation unit address generation logic
  - \* Real-mode address generation
    - \* CS:IP → code
    - \* DS:SI → data
  - \* Protected-mode address translation
    - \* Translates logical address to linear address
  - \* Protection checking

## Segmentation and Paging Unit

- \* Paging unit
  - \* Implements protected-mode paging model
  - \* Contains translation look-aside buffer
    - \* Acts as a cache for recently used page directory entries and page table entries
  - \* Translates linear address output of segmentation unit to a physical page address
  - \* Not used in real mode

## Prefetch Unit

- \* Instructions stored in FIFO queue
  - \* Holds code until ready for decoding
- \* Whenever the queue is not full, **prefetch** the next sequential instructions
  - \* Time to fetch many of the instructions in a microcomputer program is “hidden”.

## Decode Unit

- \* Offloads the responsibility of **instruction decoding** from the execution unit
  - \* Decodes instructions into the **microcode instruction** format used by the execution unit
- \* Contains another instruction queue that holds 3 fully decoded instructions
  - \* Decoded instructions are held until requested by the execution unit

## Next time

- \* Address Space
- \* Data organization

