CMPE 310: Systems Design and Programming

Instructor:

Chintan Patel

Text:

Barry B. Brey, 'The Intel Microprocessors' Eighth Edition, Pearson/Prentice Hall (2009).

Supplementary text:

Muhammad Ali Mazidi and Janice Gillispie Mazidi, 'The 80x86 IBM PC and Compatible Computers (Volumes I&II), Assembly Language, Design, and Interfacing', Third Edition, Prentice Hall (2000).

Frank Vahid and Tony Givargis, 'Embedded System Design', John Wiley (2002).

Raj Kamal, 'Embedded Systems', McGraw Hill (2008).

Lab Text:

Bob Neveln, 'Linux Assembly Language Programming', Prentice Hall PTR.

Web:

http://www.cs.umbc.edu/~cpatel2

Course Description

This course covers:

- Assembly language Programming (Intel x86).
- General system design concepts, devices and support chips
- Specifically covers architecture of the Intel microprocessors.
- Hardware configuration and control of:
  - Common microprocessor support chips, e.g. Interrupt controller, DMA controller.
  - Popular I/O devices, e.g. UART, parallel IO, timers.

Prerequisites:

- Experience with the C programming language.
- Some familiarity with Operating Systems, such as Windows.
- Experience with the Linux operating system.

Projects and Labs:

- Assembly Language Programming
- Hardware Project (Board Design and programming)

Systems Overview

General Purpose Computing Systems

Personal Computers, laptops, workstations, mainframes and servers

Systems for Dedicated Functions

- Usually embedded within larger electronic devices (referred to as embedded systems)
- Difficult to define exactly as they encompass a wide variety of electronic systems
- Definitions from several authors:
  - Any computing systems other than a general purpose computer
  - A systems consisting of hardware, main application software and an optional real time operating systems (RTOS)
  - Loosely defined, it is any device that includes a programmable computer but is not itself intended to be a general-purpose computer
  - Electronic systems that contain a microprocessor or microcontroller, but we do not think of them as computers - the computer is hidden or embedded in the system
  - It is a system whose principal function is not computational, but which is controlled by a computer embedded within it, ...
  - And many more....

Examples of Embedded Systems

- Consumer electronics
  Cell phones, pagers, digital cameras, camcorders, PDAs, DVD players, calculators
- Home Appliances
  Microwave ovens, answering machines, thermostats, home security systems
- Office Automation
  Fax machines, copiers, printers, scanners
- Business Equipment
  Cash registers, curbside check-in, alarm systems, card readers, ATMs, product scanners
- Automobiles
  Transmission control, cruise control, fuel injection, antilock brakes, active suspension
- Computing system peripherals and networking systems
  Networking equipment, routers, printers, network cards, monitors and displays
- Any many more
- E.g. In 1999, a typical American household had one PC and about 40-50 embedded computers, this has risen to about 300-400 by 2004. An average car can have more than 50 embedded computers.

System Characteristics

Single-functioned

Usually executes a specific program repeatedly

Exceptions are in cases when a system's program is updated with a newer program

Program can be swapped in and out of the system due to size limitations, depending on the function required at a specific time

Tightly Constrained

Tight constraints for design metrics such as cost, size, performance and power

Reactive and Real Time

Many systems must continually react to changes in system's environment and must compute certain results in real time without delay e.g. car cruise control

Contrast to desktop systems that typically focus on computations with relatively infrequent reactions to input devices (from the computer's perspective).

Delay in computations on desktop systems, while inconvenient to the computer user, typically does not result in a system failure

Design Metrics

- NRE Cost (nonrecurring engineering cost): One-time monetary cost of designing the system
- Unit Cost: The monetary cost of manufacturing each copy of the system
- Size: The physical space required by the system
- Performance: The execution time of the system
- Power: The amount of power consumed by the system, which may determine the life-time of the battery or the cooling requirements
- Flexibility: Ability to change the functionality of the system, software
- Time-to-prototype: The time needed to build a working version of the system, which may be bigger and more expensive than the final system implementation, used for system verification, debugging and refinements
- Time-to-market: Time required to develop a system to the point than it can be sold
- Maintainability: The ability to modify the system after its initial release, especially by designers who did not originally design the system
- Correctness: Confidence that we have implemented the system's functionality correctly
- Safety: The probability that the system will not cause harm

The Processing Unit

Microprocessor

- The CPU is a unit that fetches and processes a set of general-purpose instructions
- The CPU instruction set includes instructions for data transfer, ALU operations, stack operations, input and output (IO) operations and program control, sequencing and supervising operations
- A microprocessor is a single VLSI chip that has a CPU and may also have other units (e.g. caches, floating point processing arithmetic unit, pipelining and super-scaling units) that are additionally present and result in faster processing of instructions.
- A system designer need not be concerned about the design of the microprocessor, only needs to understand the architecture related to the programming of the processor's memory to carry out the required functionality i.e. implement the software
- Examples: Intel 8085, Intel x86 processors, Motorola 68HCxxx, Sun Sparc, IBM PowerPC etc.
- Time-to-market and NRE costs are lower when systems are designed with microprocessors as the designer must only write a program. Flexibility is also high.
- Unit cost may be low in small quantities compared with designing a dedicated chip
- Performance varies by application, unit cost may be high for larger volumes, size and power might be higher due to unnecessary processor hardware

The Processing Unit

Microcontroller

- A microcontroller is a single chip unit which, though having limited computational capabilities, possesses enhanced input-output capabilities and a number of on-chip functional units
- Particularly suited for use in embedded systems for real-time control applications with on-chip program memory and devices
- Common peripherals include serial communication devices, timers, counters, pulse-width modulators, analog-to-digital and digital-to-analog convertors
- Enables single-chip system implementation and hence smaller and lower-cost products
- Examples: Motorola 68HC11xx, HC12xx, HC16xx, Intel 8051, 80251, PIC 16F84, PIC18, ARM9, ARM7, Atmel AVR etc.

Single-Purpose Processor

- A digital circuit designed to execute exactly one program
- Commonly referred to as coprocessor, accelerator or peripheral
- Examples JPGE codec, Serial-to-Ethernet convertor, etc.

The Processing Unit

Digital Signal Processor

- Essential for systems that require large number of operations on digital signals, which are the digital encoding of analog signals like video and audio
- They carry out common signal processing tasks like signal filtering, tranformations or combinations
- Used widely in image processing applications, multimedia, audio, video, HDTV, DSP modem and telecommunication processing systems.
- They perform math-intensive operations, including operations like multiply and add or shift and add etc.
- Examples: TI TMS320Cxx, Analog Devices SHARC, Motorola 5600xx, etc.

Application Specific Instruction-Set Processors (ASIPs or ASSPs)

- A programmable processor optimized for a particular class of applications having common characteristics. Microcontrollers and DSPs can be considered as ASSPs.
- ASIPs are available for broad application classes (e.g. graphics processor) as well as very small application classes, some as small as a handful of programs
- Example: ASSP chip with TCP, UDP, IP, ARP and Ethernet 10/100 MAC logic.

The Processing Unit

Programmable Logic Devices (PLD)/ Field Programmable Gate Arrays (FPGA)

- Contains general purpose logic elements that can be programmed to implement desired functionality, very flexible for implementing custom logic circuits
- PLD usually are smaller and contain programmable gates like AND/OR arrays
- FPGAs provide lot more functionality and can be used to implement complex designs
- FPGAs can have on-chip microprocessors, memory, DSP, communication devices
- Examples: Xilinx Virtex, Spartan series FPGAs, Actel, Altera, Lattice, QuickLogic

Application Specific Integrated Circuits (ASICs)/ System-on-a-chip (SOCs)

- Custom designed VLSI chips that perform the required function
- Functionality can be integrated using IP (Intellectual property) cores
- General purpose processors are also available as IP cores and can be integrated on the chip
- Embedded processors are available from ARM, Intel, Texas Instruments and various other vendors
- Only feasible for high volume, relatively high cost systems as NRE costs and time-to-market can be significant

80x86 Evolution

- In this course we focus on the Intel x86 architecture, associated peripherals and assembly language programming. However, concepts covered apply to other logic families.

4004:

- 4-bit microprocessor.
- 4KB main memory.
- 45 instructions.
- PMOS technology.
- 50 KIPS

8008: (1971)

- **8-bit** version of 4004.
- **16KB** main memory.
- 48 instructions.
- NMOS technology.

8080: (1973)

- 8-bit microprocessor.
- **64KB** main memory.
- 2 microseconds clock cycle time; 500,000 instructions/sec.
- 10X faster than 8008.

80x86 Evolution

8085: (1977)

- 8-bit microprocessor - upgraded version of the 8080.
- 64KB main memory.
- **1.3 microseconds** clock cycle time; 769,230 instructions/sec.
- 246 instructions.
- Intel sold 100 million copies of this 8-bit microprocessor.

8086: (1978) 8088 (1979)

- **16-bit** microprocessor.
- **1MB** main memory.
- 2.5 MIPS (400 ns).
- 4- or 6-byte instruction cache.
- Other improvements included more registers and additional instructions.

80286: (1983)

- 16-bit microprocessor very similar in instruction set to the 8086.
- **16MB** main memory.
- 4.0 MIPS (250 ns/8MHz).

80x86 Evolution

80386: (1986)

- **32-bit** microprocessor.
- **4GB** main memory.
- 12-33MHz.
- Memory management unit added.
- Variations: DX, EX, SL, SLC (cache) and SX.

  80386SX: 16MB through a 16-bit data bus and 24 bit address bus.

80486: (1989)

- 32-bit microprocessor, 32-bit data bus and 32-bit address bus.
- 4GB main memory.
- 20-50MHz. Later at 66 and 100MHz
- Incorporated an 80386-like microprocessor, 80387-like floating point coprocessor and an 8K byte cache on one package.
- About half of the instructions executed in 1 clock instead of 2 on the 386.
- Variations: SX, DX2, DX4.

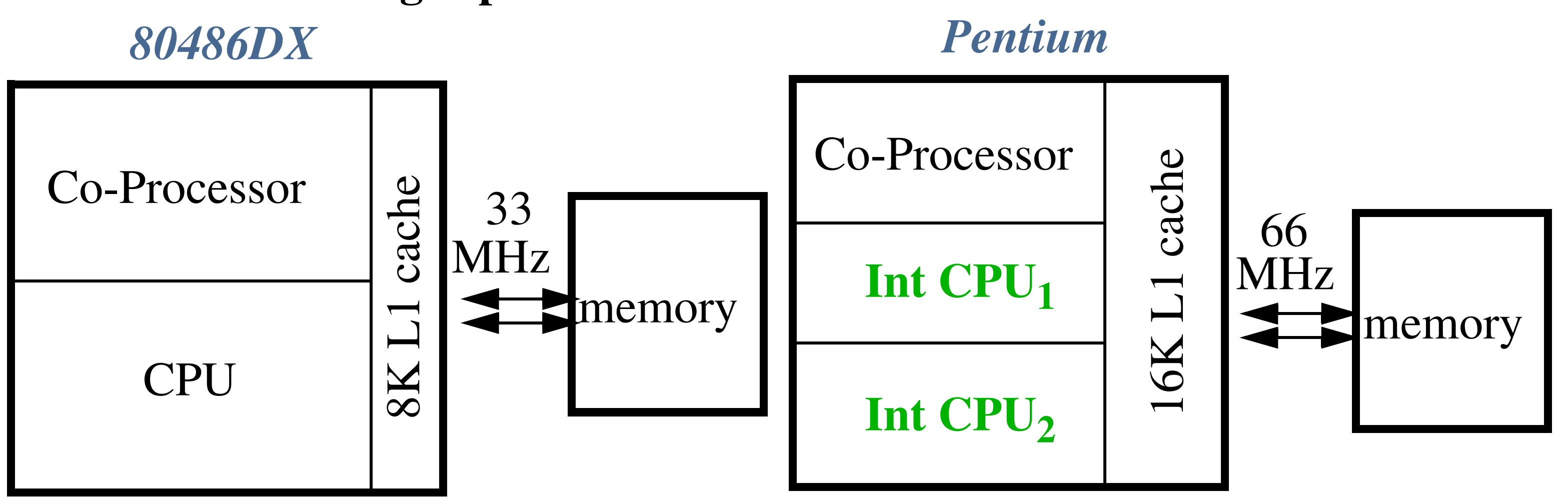
  DX2: Double clocked version:

  66MHz clock cycle time with memory transfers at 33MHz.

80x86 Evolution

Pentium: (1993)

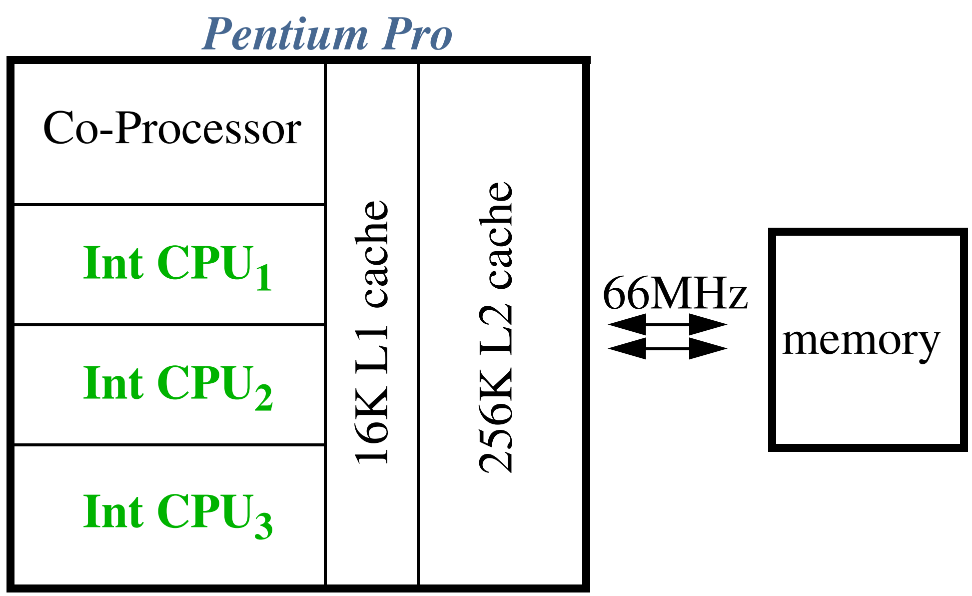
- 32-bit microprocessor, 64-bit data bus and 32-bit address bus.
- 4GB main memory.
- 60, 66, 90MHz.
  - 1-and-1/2 100MHz version.
  - Double clocked 120 and 133MHz versions.
  - Fastest version is the 233MHz (3-and-1/2 clocked version).
- **16KB L1 cache (split instruction/data: 8KB each).**
- Memory transfers at **66MHz** (instead of 33MHz).
- **Dual integer processors.**



80x86 Evolution

Pentium Pro: (1995)

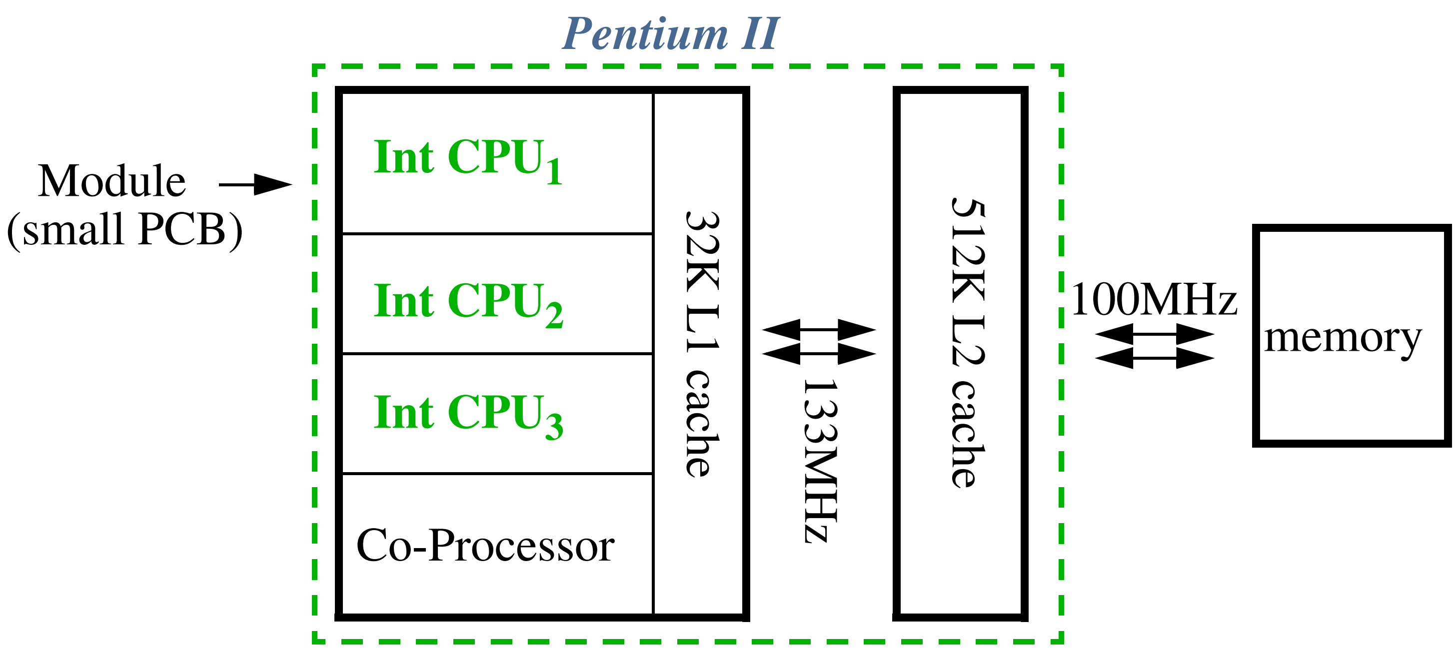
- 32-bit microprocessor, 64-bit data bus and 36-bit address bus.
- **64GB** main memory.
- Starts at 150MHz.
- 16KB L1 cache (split instruction/data: 8KB each).
- **256KB L2 cache.**
- Memory transfers at 66MHz.
- **3 integer processors.**



80x86 Evolution

Pentium II: (1997)

- 32-bit microprocessor, 64-bit data bus and 36-bit address bus.
- 64GB main memory.
- Starts at 266MHz.
- **32KB** split instruction/data L1 caches (16KB each).
- **Module integrated 512KB L2 cache (133MHz).**
- Memory transfers at 66MHz to **100MHz** (1998).

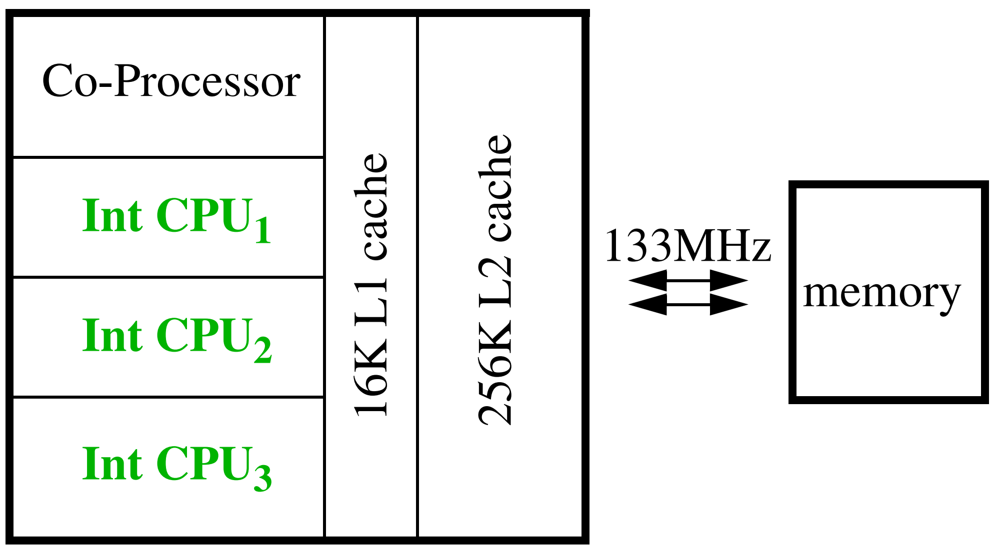


80x86 Evolution

Pentium III: (1999)

- 32-bit microprocessor, 64-bit data bus and 36-bit address bus.
- 64GB main memory.
- 800MHz and above.
- 32KB split instruction/data L1 caches (16KB each).
- **On-chip 256KB L2 cache (at-speed).**
- Memory transfers 100MHz to **133MHz.**
- **Dual Independent Bus (simultaneous L2 and system memory access).**

Pentium III



80x86 Evolution

Pentium IV: (2002)

- 1.4 to 1.9GHz and the latest at 3.20 GHz and 3.46GHz (Hyper-Threading)!
- 1MB/512KB/256KB L2 cache.
- 800 MHz (about 6.4GB/s)/533 MHz (4.3 GB/s)/ 400MHz (3.2 GB/s) system bus.
- 1066 MHz front side bus just available.
- Specialized for streaming video, game and DVD applications (144 new SIMD 128-bit instructions).
- 0.13um, more than 55 million transistors.
- Newer ones are in 90nm transistors, >125 million possible

Pentium D, Core 2 Duo, Core Duo, Core 2 Extreme Edition: (2005-2008)

- Dual and Quad processing cores
- Upto 4MB L2 cache and 1066 MHz FSB
- 65 nm and 45 nm transistors (and lots of them!!!)

Refer to the following URL for more details:

http://www.intel.com/design/