Course Overview

CIT 595

Spring 2007

INTRODUCTION

Digital Systems Organization & Design

- A **digital** system is one that uses discrete values (often electrical voltages)
  - Esp. those representable as binary information or finite symbols for input, processing, transmission, storage, or display, rather than a continuous spectrum of values (i.e. as in an analog system)
- In CIT 593 we studied that a computer input, processing, transmission, storage or display is done using binary system.
- In CIT 595 we will explore the organization and design of a computer system

Why study computer organization ?

- To understand the physical limitations of the underlying hardware on which your software runs
- Design better programs, including system software such as compilers, operating systems, and device drivers
- Optimize program behavior
- Evaluate computer system performance (time, price and space tradeoff)

Computer Architecture vs. Organization

Computer Architecture

- Logical aspects of system implementation as seen by the programmer
- Such as instruction sets, instruction formats, data types, addressing modes
- E.g. LC3 ISA studied in CIT 593
- Answers the question: *How do I design a computer?*
  - E.g. Do I want 1, 8, or 32 register? We'll see that number of registers affects program performance due to organization of memory

Computer Arch vs. Org

Computer Organization

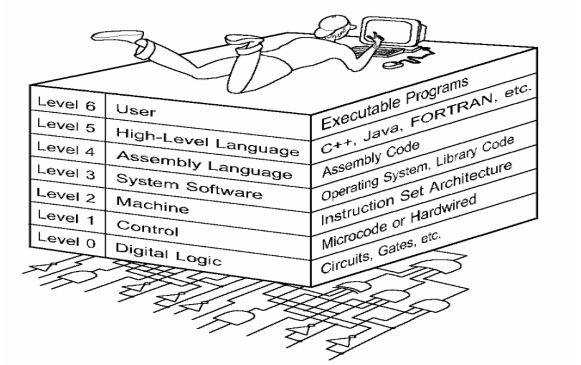
- Encompasses all physical aspects of computer systems
- Such as circuit design, control signals, memory types
- Answers the question: *How does a computer work?*
- We will study this in CIT 595 and more…

It will be evident that both architecture and organization are interrelated and interdependent

REVIEW

The Computer Level Hierarchy

Each layer is an abstraction of the level below it



The Computer Level Hierarchy

- Level 6: The User Level
  - Composed of application programs such as Word Processor, Paint etc.
  - The implementation of the application is hidden completely from the user
- Level 5: High-Level Language Level
  - The level allows users to write their own application with languages such as C, Java and many more
  - High-level languages are easier to read, write, and maintain
  - User at this level sees very little of the lower level

The Computer Level Hierarchy

- Level 4: Assembly Language Level
  - Lowest human readable form before dealing with 1s and 0s (machine language)
  - Assembler converts assembly to machine language
- Level 3: System Software Level
  - Operating System software supervises other programs
    - Controls execution of multiple programs
    - Protects system resources. E.g. Memory and I/O devices
  - Other utilities
    - Compilers, Interpreters, Linkers, Library etc.
  - The software can be written in both assembly and high-level language
    - High-level is much more portable i.e. easier to modify to work on other machines

The Computer Level Hierarchy

- Level 2: Machine Level
  - Also known as the Instruction Set Architecture (ISA) Level
  - Consists of instructions that are particular to the architecture of the machine
  - Programs written in machine language (0s and 1s) need no compilers, interpreters, or assemblers
- Level 1: Micro-architectural Level
  - Detailed organization of a processor implementation
    - How the control unit interprets machine instructions (from fetch thru execute stages)
  - There can be different implementations of a single ISA
  - In the book this level is called “Control level”

The Computer Level Hierarchy

- Level 0: Digital Logic Level
  - This level is where we view physical devices as just switches (On/Off)
  - Instead of viewing their physical behavior (i.e. in terms of voltages and currents) we use two value logic i.e. 0 (off) and 1(on)
  - We will briefly look at the physical electronic components – mainly the transistor technology

CIT 593	CIT 595
level 6 - 4, and level 2	level 3, 1, 0

First Generation Digital Computer

- On the ENIAC, all programming was done at the digital logic level
- Programming the computer involved moving plugs and wires
- A different hardware configuration was needed to solve every unique problem type

Configuring the ENIAC to solve a "simple" problem required many days labor by skilled technicians

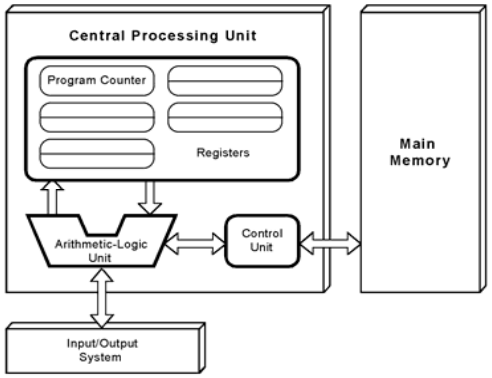
The von Neumann Model

- Inventors of the ENIAC, John Mauchley and J. Presper Eckert, conceived of a computer that could store instructions in memory
- John von Neumann, popularized it
- Stored-program computers have become known as von Neumann Architecture systems

The von Neumann Model

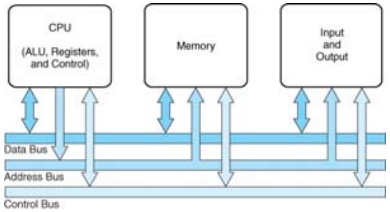
- Today's stored-program computers have the following characteristics
  - ➢ Three hardware systems:
    - A central processing unit (CPU)
      - ALU, registers, PC, control unit
    - A main memory system
    - An I/O system
  - ➢ The capacity to carry out sequential instruction processing
  - ➢ The data transfer between the CPU and memory or CPU and to I/O system done using a bus. A bus collection of wires that can transfer x-bits

The von Neumann Model



Adaptations to Neumann Model

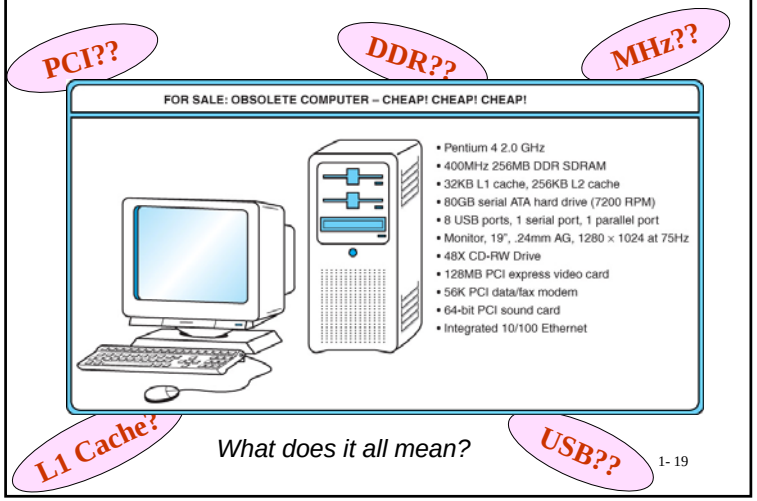
- Over the years Neumann Model has changed to support more hardware devices reason being
  - Larger program size
  - Memory speed has not caught up with processor speed
  - More interaction with outside world e.g. networking, digital media etc



The Modified von Neumann Architecture

PREVIEW

Modern Computer System Example



What does it all mean?

Before we jump into the example….

Measurement Metric

- Kilo- (K) = 1 thousand = 10^3 and 2^{10}
- Mega- (M) = 1 million = 10^6 and 2^{20}
- Giga- (G) = 1 billion = 10^9 and 2^{30}
- Tera- (T) = 1 trillion = 10^{12} and 2^{40}

- Whether a metric refers to a power of 10 or a power of 2 typically depends upon what is being measured
  - power of 2 is associated with memory size
  - power of 10 is associated with speed i.e. how fast is information transferred

Pentium 4 2.0 GHz

- Note: Hertz = cycles per second (frequency)
  - E.g. 1GHz = 10^9Hz (Giga = 10^9)
- 2.0 GHz refers to clock speed/rate of the Pentium 4
  - The **clock rate** is the fundamental rate at which a processor can execute an instruction
  - E.g. adding two numbers takes 0.5 x 10^{-9} s or 0.5ns
  - From the above example we can say that 2 billion instructions can be executed in one second
  - However we will see that the number of instruction executed in a second is only proportional to the speed (not equal)

400 MHz 256MB DDR SDRAM

- 400 MHz refers to rate at which data is transferred between CPU and Memory (i.e. bus transfer rate)
- DDR SDRAM is type of memory…more on in chapter 6
- The memory capacity is 256MB (i.e. how much information you can store)

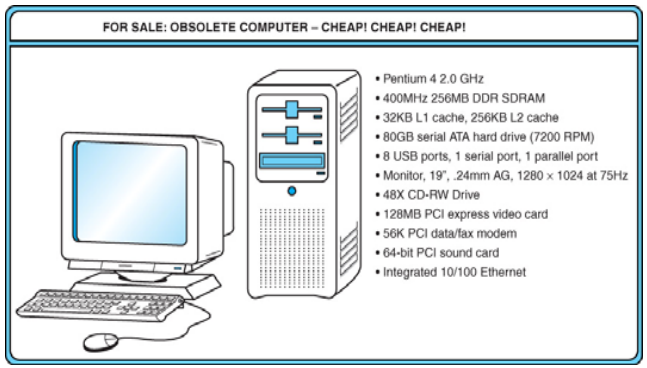
32KB L1 and 256KB L2 Cache

- Over the years there has been significant speed up in processor speed but the same has not been the case with memory
- To provide even faster to data to/from memory, most modern systems provide small fast memory in between CPU and main memory called Cache. Cache is faster than main memory in terms of data access
- More on this in chapter 3 and 6

80 GB serial ATA hard drive

- Hard Drive is another storage device that stores information (instructions/data)
- Its storage capacity is way higher than memory that the CPU interacts (256 MB vs. 80 GB)
- In chapter 7, we will see that they are inherently slower in terms of data access and hence when application program is first started we need to load it from hard drive to memory
- ATA stands for *advanced technology attachment*, which describes how the hard disk interfaces with (or connects to) inside a computer system

Example



8 USB, 1 serial and 1 Parallel Ports

- Ports (similar to a bus) allow movement of data between CPU and external devices (I/O)
- Serial ports send data as a series of pulses along a data line (i.e. 1 bit at a time)
- Parallel ports send data as a single pulse along at least eight data lines (old printers used parallel ports to transfer data to printer)
- USB, Universal Serial Bus, is also a serial interface like serial port but better
  - Hot Plugging (add and remove devices while the computer is running)

Monitor 19" .24mm AG, 1280 x 1024 at 75 Hz

- 1280 x 1024 are viewing area dimensions
- The number of times per second that the image on a monitor is repainted (is its *refresh rate i.e. 75 Hz)*
  - Low refresh rates cause screen to have jiggle or wave appearance and can strain the eyes
- Dot pitch rating of a monitor tells you just how sharp the displayed image will be
  - Measured in millimeters (mm), and a smaller number means a sharper image

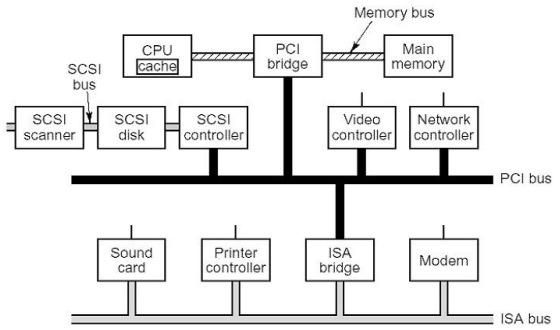
48x CD-RW

- The Compact Disk drive supports rewritable (RW) CDs that can be written to many times.
- 48x describes its speed/rate at which the drive can read that from the CD
- In chapter 7 we will learn how data is stored in and read from a CD

Peripheral Component Interface PCI

- A **peripheral** is a type of computer hardware device that is added to a computer in order to expand its abilities.
- PCI, *peripheral component interface*, is dedicated I/O bus for peripherals such as video, audio, fax/modems & Ethernet
- The PCI specification covers the physical size of the bus (including wire spacing), electrical characteristics, bus timing, and protocols. The specification was formed by PCI Special Interest Group
- In general we will study how bus operations occur

More realistic view of the system



One more thing …

- Anything missing from the example???
  - An Operating System (O.S)
- System is really incomplete without O.S., why?
  - Controls execution of multiple programs
  - Protects system resources. E.g. Memory and I/O devices
- What to look forward to?
  - Operating System (O.S) design
  - O.S. services
  - Concept of processes, and threads
  - Programming Tools

Reason for the Example?

- To introduce the components of the computer system that will be studied
  - Learn their organization and design as well as their interaction with other components

Project

- You will also get a chance to do some research on particular component (hardware or software) of the computer system (may be existing or emerging technology)

Other Special Topics

- Virtual Machines
  - Look at Java Virtual Machine (JVM)
- Alternative Architectures (Parallel and Multiprocessor)
  - Superscalar/Vector/VLIW processors, Chip multiprocessors
- Network organization and architecture (optional)
  - Very brief insight
  - An entire course will do justice to the topic!!

Next Time

Lecture 01/10

- Historical Development & some background on transistors

Lecture 01/12 (make up)

- Digital Logic

See Reading assigned on website