Computer Architecture and Organization

Course et2-010

Henk Corporaal
Delft University of Technology
Spring 2000

Email: h.corporaal@et.tudelft.nl
URL : http://cardit.et.tudelft.nl/~heco/courses/et1-101

Introduction

Purpose of course:

- learn how computers operate
- learn how to construct and build a computer system
  - emphasis on processor design and implementation
- learn how to program a computer
  - assembly level
  - machine level
- learn design principles
- operating system introduction

- processor design as part of system design

Material

- Book: Computer Organization & Design
  David A. Patterson and John L. Hennessy
  Morgan Kaufmann, 2nd Ed. 1998

- Handouts
  - Introduction to Operating Systems
  - Slides
  - Exercises
  - Miscellaneous material

- Lab. material (SPIM assembler + simulator)

Look at the web site
http://cardit.et.tudelft.nl/~heco/courses/et2-010

Topics of Discussion

- Computer system overview
- Measuring and comparing performance
- ISA: instruction set architecture
- Computer arithmetic
- How to implement a processor
- How to implement a fast processor
- Memory hierarchy and caching
- Interfacing
- OS

Prerequisites: Computer Systems, first year (digital technique and introduction programming)

Preliminary Schedule

Chapter 1	1
Chapter 2	2
Chapter 3	4
Chapter 4	3
Chapter 5	4
Chapter 6	3
Chapter 7	4
Chapter 8/9	1
Other processors	2
OS	4
Total	28 Contact hours
Lab. assignment	4 + 4 hours

Chapter 1

Computer system overview

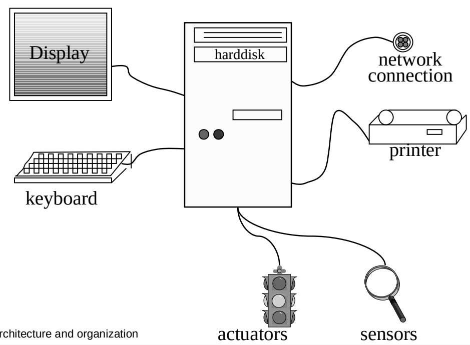
Topics

- Computer revolution
- What is a computer system?
- Processor
- Memory
- Chips and semiconductor
- Communication
- Programming a computer (basics)

Computer revolution

- Computers everywhere
  - PCs, portables, and workstations O(100M)/year
  - Embedded systems O(1000M)/year
  - Playstations O(?)
- World Wide Web
- Huge calculations
  - Human genome project
  - Large simulations
- Information retrieval
  - Huge databases
  - Online libraries
  - Web pages
- Computer performance growth

What is a computer system?



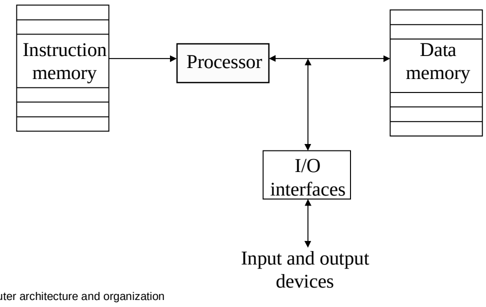
What is a computer system?

Components:

- input (mouse, keyboard)
- output (display, printer)
- memory
  - internal: DRAM, SRAM
  - external: hard disk drives, CD, floppy drive
- network

- see figs 1.4 - 1.12

Structure of a computer system?



Focus

- Our primary focus: the processor (CPU = central processing unit)
  - datapath and
  - control

- Implemented using millions of transistors
- Impossible to understand by looking at each transistor
- We need… abstractions …. at many levels

Abstraction

- Abstraction:
  - Give a complex item/object/structure a name
    - hide detail
  - The name represents this item/object/structure and can be used inside other items/objects/structures
  - This gives a *hierarchical description*
- Examples:
  - Binary abstraction
  - Circuit abstraction
  - Machine instruction abstraction
  - Program abstraction
  - Data abstraction

Abstraction layers

Level	
High level language	Software
Intermediate language level	Compiler
Operating system level	Architecture / assembly
Instruction set architecture (ISA)	
Register transfer level (RTL)	Implementation / digital design
Gate level	Realization
VLSI level	

Processor

- Sold (1995)
  - 80x86 50 Million
  - MIPS 5.5 Million
  - PowerPC 3.3 Million
  - Sparc 700 K
  - HP PA-RISC 300 K
  - DEC Alpha 200 K
- Performance: see fig 1.20 (pg 30)
  - grows with about 50% per year
  - doubling each 1.6 years
  - hundred fold increase last 10 years

Processor

- How does a processor operate ?
- Performance growth, how?
  - Realization improvements
    - VLSI developments
  - Implementation improvements
    - Pipelining
    - Better circuits
  - Architecture improvements
    - Exploit parallelism
    - Prediction

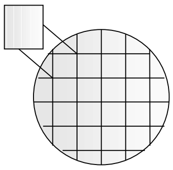
Memory

- Size
  - units: bit, byte, kbit, kbyte, Mbit, Mbyte, Gbit, Gbyte
  - growth rate: see fig 1.14
    - DRAM growth rule: 4 times each 3 years
- Types:
  - Volatile: SRAM, DRAM
  - Non-volatile:
    - Internal: ROM (and its many variants)
    - External: hard disk, floppy, CDROM, magnetic tape
- Performance metrics:
  - Latency: access time in seconds (or nano seconds)
  - Throughput: bytes/second
- Price / Perfomance or Price / Mbyte

Chips and semiconductor

- VLSI improvements
  - Density
    - Processor O(10 Million) transistors/chip
    - Memory O(250 Million) transistors/chip (DRAM)
  - Decreasing feature size: now 0.25 micron, next 0.18 micron
  - Speed of a transistor ~ 1/gate-length
- However: future problems
  - wiring delays
  - quantum effects (like charge tunneling)
- Note: chip area not much increasing
  - maximal about 1.5 cm x 1.5 cm
  - yield !!

Cost of an integrated circuit



$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer * Yield}}$$

$$\text{Dies per wafer} = \frac{\text{Wafer area}}{\text{Die area}}$$

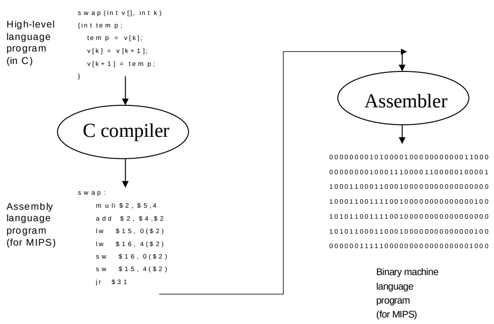
$$\text{Yield} = \frac{1}{(1+(\text{Defects per area x Die area}/2))^2}$$

Result: Cost of die = f (Die area $^?$)

Communication

- Computers can act stand alone, or in a network
- Why network connected computers?
  - Communication
    - information exchange
  - Resource sharing
    - printers, modems, scanners, large storage capacities
- Interconnection is standardized through protocol agreements
  - IP, TCP, Ethernet, Modem, Bluetooth standards, etc.

Programming a computer



Instruction Set Architecture: ISA

- A very important abstraction
  - interface between hardware and low-level software
  - standardizes instructions, machine language bit patterns, etc.
  - advantage: *different implementations of the same architecture*
  - disadvantage: *sometimes prevents using new innovations*

  ***True or False**: Binary compatibility is extraordinarily important?*

- Modern instruction set architectures:
  - 80x86/Pentium/K7, PowerPC, DEC Alpha, MIPS, SPARC, HP-PA, and what about JVM?

Where we are headed

- Performance issues (Chapter 2)
  - *vocabulary and motivation*
- A specific instruction set architecture (Chapter 3)
- Arithmetic and how to build an ALU (Chapter 4)
- Constructing a processor to execute our instructions (Chapter 5)
- Pipelining to improve performance (Chapter 6)
- Memory: caches and virtual memory (Chapter 7)
- I/O (Chapter 8)
- Preliminaries:
  - Logic design (appendix B)

  Key to a good grade: reading the book!

Exercises

- Try the following from chapter one:
- 1.1 - 1.44 about definitions and terminology
- 1.46, 1.47
- 1.48 (read ‘in depth’ section pg. 48 first), 1.50