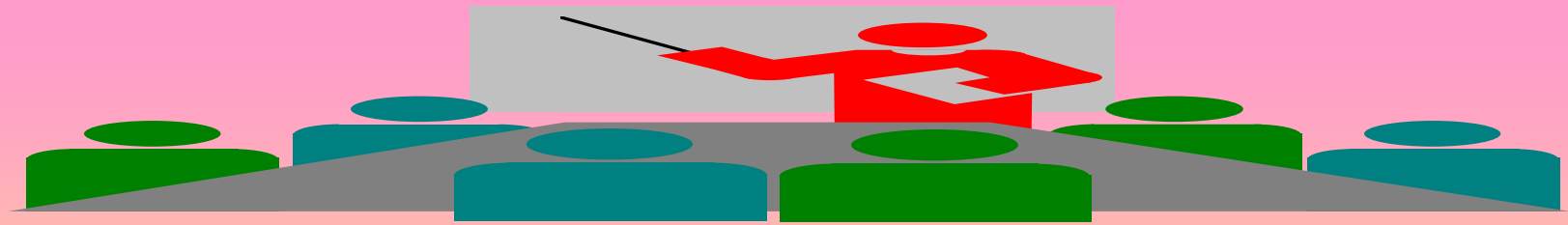
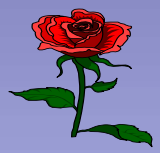


Computer Organization & Design

The Hardware/Software Interface

施青松

http://10.214.26.103 **Email:** zjsqs@zju.edu.cn

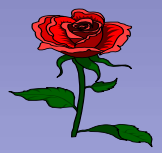


Background

It is very easy to design CPU IP Core!

It is not easy to design good CPU!

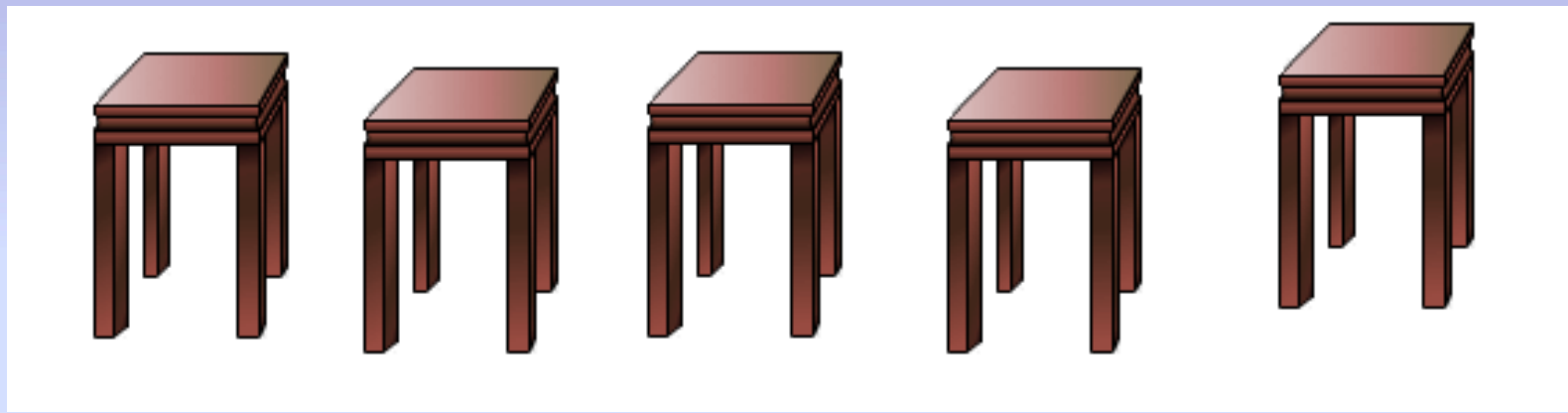
To design successfully is far more difficult than one!



课程地位

- 考研统考课程之一

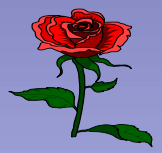
软件：汇编语言→编译→ OS →算法语言→软件工程



硬件：数字电路→组成→硬件实现→接口→体系结构

软件专业

计算机专业



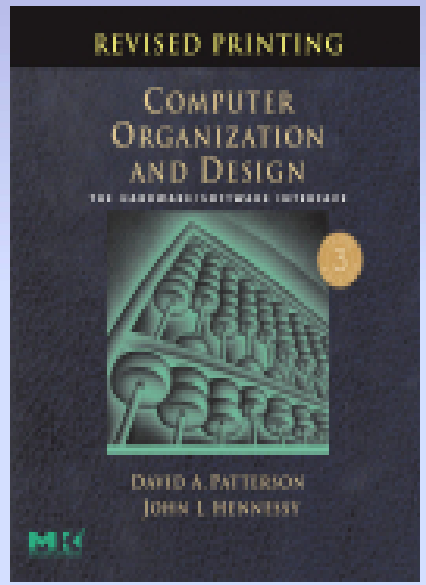
课程体系：三位一体、循序递进

立足基础、加强实践、服务专业、进入国际

- 数字逻辑课程：计算机组成相关部件的设计 基础
  - 组合电路设计、时序电路设计
- 计算机组成：设计简单RISC-CPU核 核心
  - ALU部件
  - 单周期实现、多周期实现简单的32位RISC-CPU
    - 写入FPGA，用实验板卡做测试验证。
- 计算机系统结构：设计流水线RISC-CPU核心 提高

课程教材

- Computer Organization & Design
  ——The Hardware/Software Interface



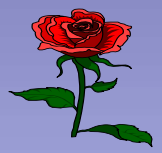
John L. Hennessy

Stanford University

David A. Patterson

California University,Bereley

实验教材



如何学好这门课？----耕耘与收获

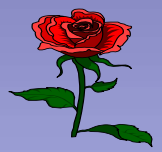
- 孟子曰：

  舜发于畎亩之中，傅说举于版筑之间，胶鬲举于鱼盐之中，管夷吾举于士，孙叔敖举于海，百里奚举于市。

- 故天将降大任于斯人也，必先苦其心志，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为，所以动心忍性，曾益其所不能。

- 人恒过，然后能改。困于心，衡于虑，而后作。征于色，发于声，而后喻。入则无法家拂士，出则无敌国外患者，国恒亡。

- 然后知生于忧患，而死于安乐也。



成功的秘诀

出生

- 舜从田野之中被任用，傅说从筑墙工作中被举用，胶鬲从贩卖鱼盐的工作中被举用，管夷吾从狱官手里释放后被举用为相，孙叔敖从海边被举用进了朝廷，百里奚从市井中被举用登上了相位。

磨难

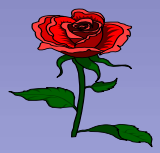
- 所以上天将要降落重大责任在这样的人身上，一定要道先使他的内心痛苦，使他的筋骨劳累，使他经受饥饿，以致肌肤消瘦，使他受贫困之苦，使他做的事颠倒错乱，总不如意，通过那些来使他的内心警觉，使他的性格坚定，增加他不具备的才能。

历程

- 人经常犯错误，然后才能改正；内心困苦，思虑阻塞，然后才能有所作为；这一切表现到脸色上，抒发到言语中，然后才被人了解。在内（国内）如果没有坚持法度的世臣和辅佐君主的贤士，在外（国际）如果没有敌对国家和外患，此国便经常导致灭亡。

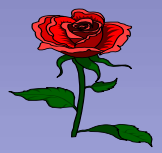
祖训

- 这就可以说明，忧愁患害可以使人生存，而安逸享乐使人萎靡死亡。



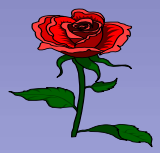
课堂教学的作用

- 教学是双方互动的，不能一边倒。大学应素质教育为主，要鼓励学生在教师指导下的自学与动手。
- 课堂教学作用是:引出知识及相关知识点,引导学生猎取知识的方向，分析知识的难点，学会分析讨论解决问题的途经，节省课余时间，提高自学的效率。
- 学会‘止于至善’，知道‘物极必反’
  - 大学之道，在明明德，在亲民，在止於至善。知止而後有定，定而後能静，静而後能安，安而後能虑，虑而後能得。物有本末，事有终始，知所先後，则近道矣。



课堂教学----实践的指导方针

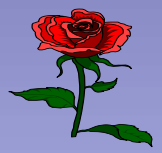
- 注重知识的系统性、连贯性，强化实践能力
  - 立足组成，三位一体，从程序员角度俯视组成结构
  - 知其来路，又知其去路；知其然，知其所以然。
- 培养自主学习能力
  - 引出组成及相关知识的自主获取和消化方法
    - 力求充分体现培养学生硬件知识的自学方法
  - 引导猎取知识的方向，给出分析问题的途经
    - 节省课余时间，提高预习、复习、自学的效率。
- 启发式、鼓励式课堂交互
  - 引出关键问题，开展提问和讨论
  - 培养讨论，争论，辩论的学习气氛
  - 核心、重要知识点学生上台
  - *课程设计presentation



实验教学----知识的感性化

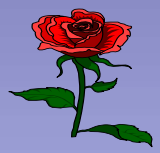
1	MIPS汇编模拟	(光盘)用软件进行汇编反汇编MIPS模拟机实现实验
2	硬件设计基础	Spartan实验板与ISE软件进行硬件设计基础实验
3	基本组件设计	MUX、寄存器组组件设计
4	ALU与ALU控制器	ALU设计实验，ALU控制器
5	R类型指令设计	单指令设计实现
6	CPU控制器	CPU控制器设计
7	单时钟数据通道	单时钟数据通道设计
8	多时钟数据通道	多时钟数据通道设计
9	微程序控制单元	微程序控制单元设计
10	微程序控制处理器	微程序控制数据通道设计
11	有限指令CPU设计	9条指令的IP核实现
12	MIPS处理器系统模拟	编写MIPS模拟执行

以实验课为准



考核

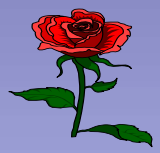
- 平时　　15%
  - 作业、阅读：光盘+一篇论文
- 期中　　15%(统一时间)
  - **5.4 A Simple Implementation Scheme**
- 期末　　70%
  - The all and the one
- 英文试卷



Content at Classroom

- **Chapter One: Computer Abstractions and Technology**
- **Chapter Two: Instructions: Language of the Computer**

2.1 Introduction
2.2 Operations of the Computer Hardware
2.3 Operands of the Computer Hardware
2.4 Representing Instructions in the Computer
2.5 Logical Operations
2.6 Instructions for Making Decisions
2.7 Supporting Procedures in Computer Hardware
2.8 Communicating with People
2.9 MIPS Addressing for 32-bit Immediates and Addresses
2.10 Starting a Program
2.11 How Compilers Optimize
2.12 How Compilers Work: An Introduction
2.13 A C Sort Example to Put It All Together
2.14 Implementing an Object Oriented Language
2.15 Arrays versus Pointers



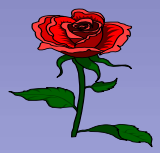
Content at Classroom-2

- **Chapter Three: Arithmetic for Computers**

  3.1 Introduction
  3.2 Signed and Unsigned Numbers
  3.3 Addition and Subtraction
  3.4 Multiplication
  3.5 Division
  3.6 Floating Point

- **Chapter Five: The Processor: Datapath and Control**

  5.1 Introduction
  5.2 Logic Design Conventions
  5.3 Building a Datapath
  5.4 A Simple Implementation Scheme
  5.5 A Multicycle Implementation
  5.7 Exceptions
  5.8 Microprogramming: Simplifying Control Design
  5.9 An Introduction to Digital Design Using a Hardware Design Language



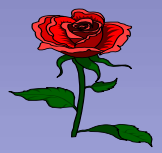
Content at Classroom-3

- **Chapter Seven: Large and Fast: Exploiting Memory Hierarchy**

  7.1 Introduction
  7.2 The Basics of Caches
  7.3 Measuring and Improving Cache Performance
  7.4 Virtual Memory
  7.5 A Common Framework for Memory Hierarchies

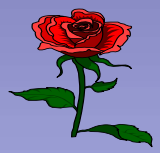
- **Chapter Eight: Storage, Networks, and Other Peripherals**

  8.1 Introduction
  8.2 Disk Storage and Dependability
  8.3 Networks
  8.4 Buses: Connecting I/O Devices to Processor and Memory
  8.5 Interfacing I/O Devices to the Memory, Processor, and Operating System
  8.6 I/O Performance Measures: Examples from Disk and File Systems
  8.7 Designing an I/O System

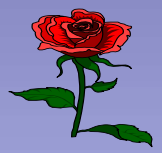


Kernel

- How does Hardware support HLL?
- Arithmetic for Computers
- **Datapath and Control**
- Exploiting Memory Hierarchy
- Storage, Networks, and Other Peripherals



考研大纲
《计算机组成》课程分析

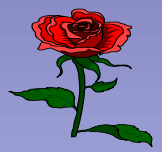


考查目标

- 计算机学科专业基础综合考试涵盖
  - 数据机构(45分)
  - 计算机组成原理(45分)
  - 操作系统(35分)
  - 计算机网络(25分)

计组是最重要两门课程之一

- 要求
  - 考生比较系统地掌握上述专业基础课程的概念、基本原理和方法
  - 能够运用所学的基本原理和基本方法分析、判断和解决有关理论问题和实际问题



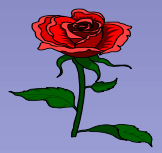
考试形式和试卷结构

- 试卷满分及考试时间
  - 满分150分，考试时间180分钟(3小时)
- 答题方式
  - 答题方式为闭卷、笔试
- 试卷内容分布
  - 数据结构 45分
  - 计算机组成原理 45分
  - 操作系统 35分
  - 计算机网络 25分
- 试卷题型结构
  - 单项选择题 80分(40小题，每小题2分)
  - 综合应用题 70分

按比例，计组有45分：
选择题13.3题目26.7分
应用题23.3分

两种分配方案
如应用题20分，则选择题12.5道
如应用题25分，则选择题10道

应用题型：简答5分一个，问答10分一个，简单设计10分一个，复杂一些的设计15分



计算机组成原理

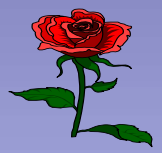
- 考查目标

1. 理解单处理器计算机系统中各部件的内部工作原理、组成结构以及相互连接方式，具有完整的计算机系统的整机概念。
2. 理解计算机系统层次化结构概念，熟悉硬件与软件之间的界面，掌握指令集体系结构的基本知识和基本实现方法。
3. 能够运用计算机组成的基本原理和基本方法，对有关计算机硬件系统中的理论和实际问题进行计算、分析，并能对一些基本部件进行简单设计。

目标1：以MIPS为主,本课程主要介绍的是RISC，补充CISC处理器(X86)后续微机原理课程主要介绍X86结构

目标2：这部分包括了汇编，本课程介绍RISC汇编，CISC汇编在微机原理课程介绍中；内容还涉及到部分计算机体系结构课程，后面有详述

目标3：这部分涉及了数字电路知识，由逻辑与计算机设计基础课程介绍



大知识点分析

- 大纲涉及七大知识点
  - 一、 计算机系统概述
  - 二、 数据的表示和运算
  - 三、 存储器层次机构
  - 四、 指令系统
  - 五、 中央处理器(CPU)
  - 六、 总线
  - 七、 输入输出(I/O)系统

本课程的大纲：

一、概述

二、MIPS汇编语言（属于RISC指令集）

三、计算机代数（含数的表示、ALU设计）

四、数据通道（含控制器）设计*

五、存储层次

六、输入输出（含一小部分总线知识点）

结论：在大知识点上，
本课程覆盖大纲

*三中的ALU设计，加上四的控制器，合在一起就是中央处理器设计

Chapter 1

Computer Abstractions and Technology

Contents of Chapter 1

- 1.1 Introduction
- 1.2 Computer Language and Software System
- 1.3 Computer Hardware System
- 1.4 Integrated Circuits
- 1.5 Real Stuff: Manufacturing Pentium Chips
- 1.6 History of Computer Development

1.1 Introduction

- Computers have led to a third revolution for civilization
- The following applications used to be "computer science fiction"
  - Automatic teller machines
  - Computers in automobiles
  - Laptop computers
  - Human genome project
  - World Wide Web

- Tomorrow's science fiction computer applications
  - Cashless society
  - Automated intelligent highways
  - Genuinely ubiquitous computing:
    No one carries computers because they are available everywhere.

- Classes of Computer Applications and Their Characteristics
  - Desktop computer
  - Servers
  - Embedded computer

- The influence of hardware on software
  - In the past
    - Memory size was very small
    - Programmers must minimize memory space to make programs fast
  - Nowadays
    - The hierarchical nature of memories
    - The parallel nature of processors
    - Programmers must understand computer organization more

- Brief introduction to this course
  - The internal organization of computers and its influence on the performance of programs
  - The hierarchy of software and hardware
    - How are programs written in high-level language translated into the language of the hardware, and how does it run?
    - What is the interface between the software and the hardware, and how does software instruct the hardware to perform?
    - What determines the performance of a program, and a programmer improve the performance?
    - What techniques can used to improve performance?

- Brief introduction to Chapter 1
  - Basic ideas and definitions
  - Major components of software and hardware
  - Introduction to integrated circuits
  - Technology that fuels the computer revolution

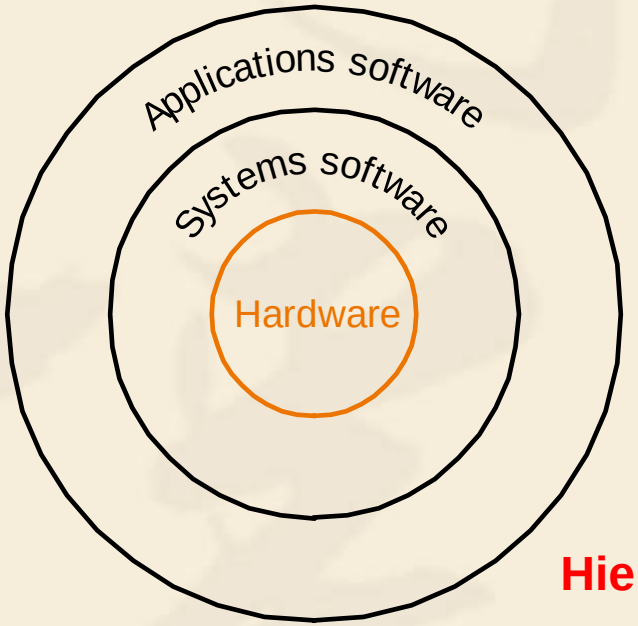
Where is the performance bottleneck?

Hardware or software component	How this component affects performance	Where is this topic covered?
Algorithm	Determines both the number of source-level statements and the number of I/O operations executed	Other books!
Programming language,compiler, and architecture	Determines the number of machine instructions for each source-level statements	Chapter 2 and 3
Processor and memory system	Determines how fast instructions can be executed	Chapter 5,6 and 7
I/O system(hardware and operating system)	Determines how fast I/O operations may be executed	Chapter 8

1.2 Below Your Program

From a High-Level Language to the Language of Hardware

A simplified view of hardware and software as hierarchical layers



Hierarchical layers

Problem:

should we really place compilers in the systems software level ?

Some terms

- Machine language
  - Computers only understands electrical signals
  - Easiest signals: *on* and *off*
  - Binary numbers express machine instructions ex. 1000110010100000 means to add two numbers
  - Very tedious to write
- Assembly language
  - Symbolic notations ex. add A, B
  - The assembler translates them into machine instruction
  - Programmers have to think like the machine

- High-level programming language
  - Notations more closer to the natural language
    ex. A + B
  - The compiler translates them into assembly language statements
  - Subroutine library ---- reusing programs
  - Advantages over assembly language
    - Programmers can think in a more natural language
    - Improved programming productivity
    - Programs can be independent of hardware

- Categorize software by its use
  - Systems software ---- aimed at programmers
  - Applications software ---- aimed at users

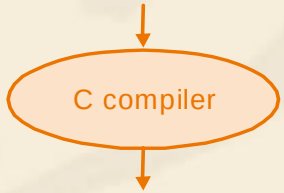
- Operating System
  - Handing basic input and output operations
  - Allocating storage and memory
  - Providing for sharing the computer among multiple applications using it simultaneously
- Compiler
  - Translation of a program written in HLL

From a High-Level Language to the Language of Hardware

- The process of compiling and assembling

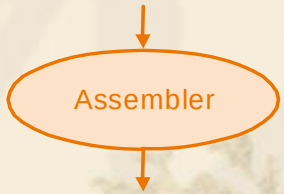
High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```



Assembly language program (for MIPS)

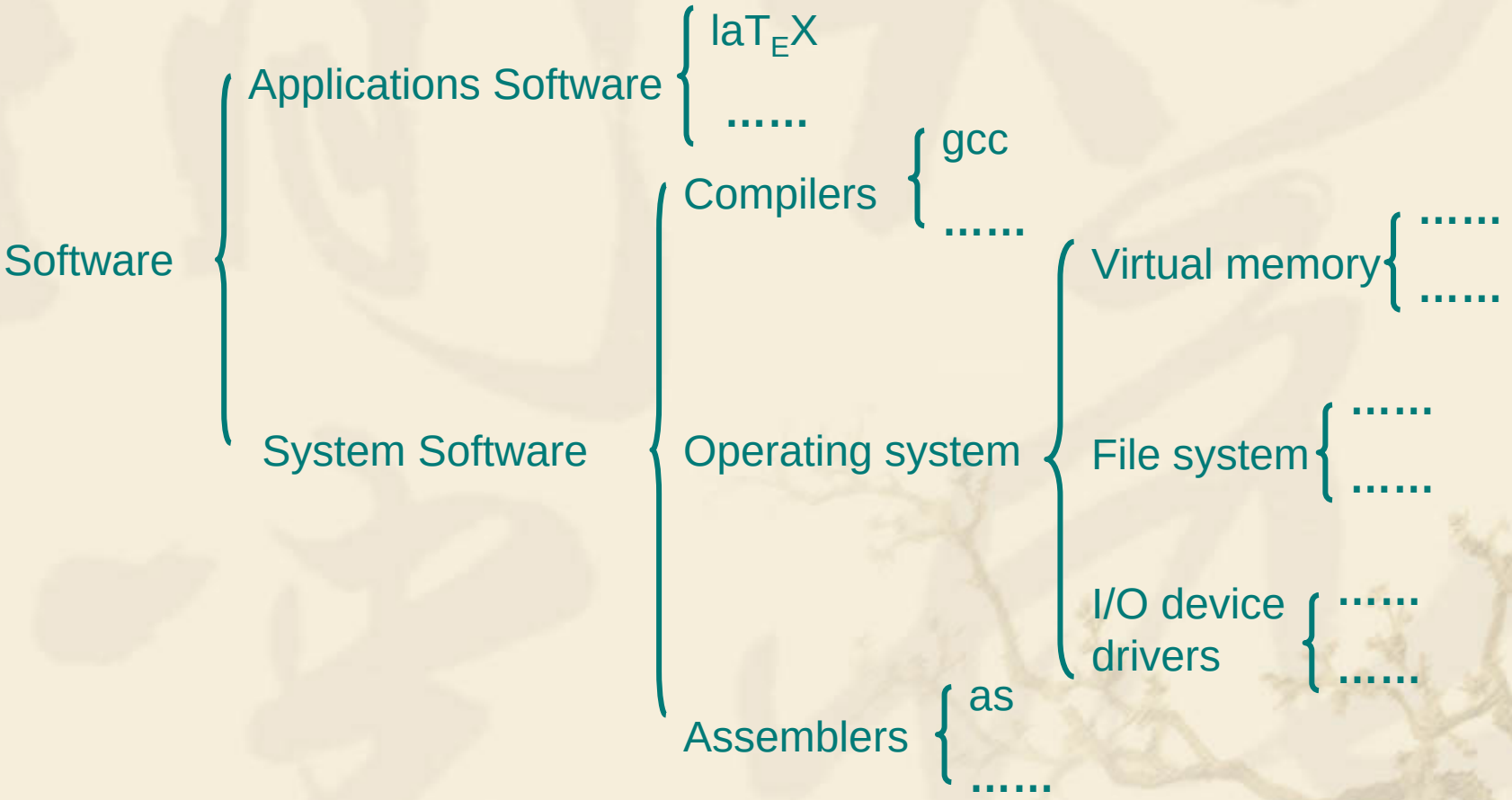
```
swap:
     muli $2, $5,4
     add  $2, $4,$2
     lw   $15, 0($2)
     lw   $16, 4($2)
     sw   $16, 0($2)
     sw   $15, 4($2)
     jr   $31
```



Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

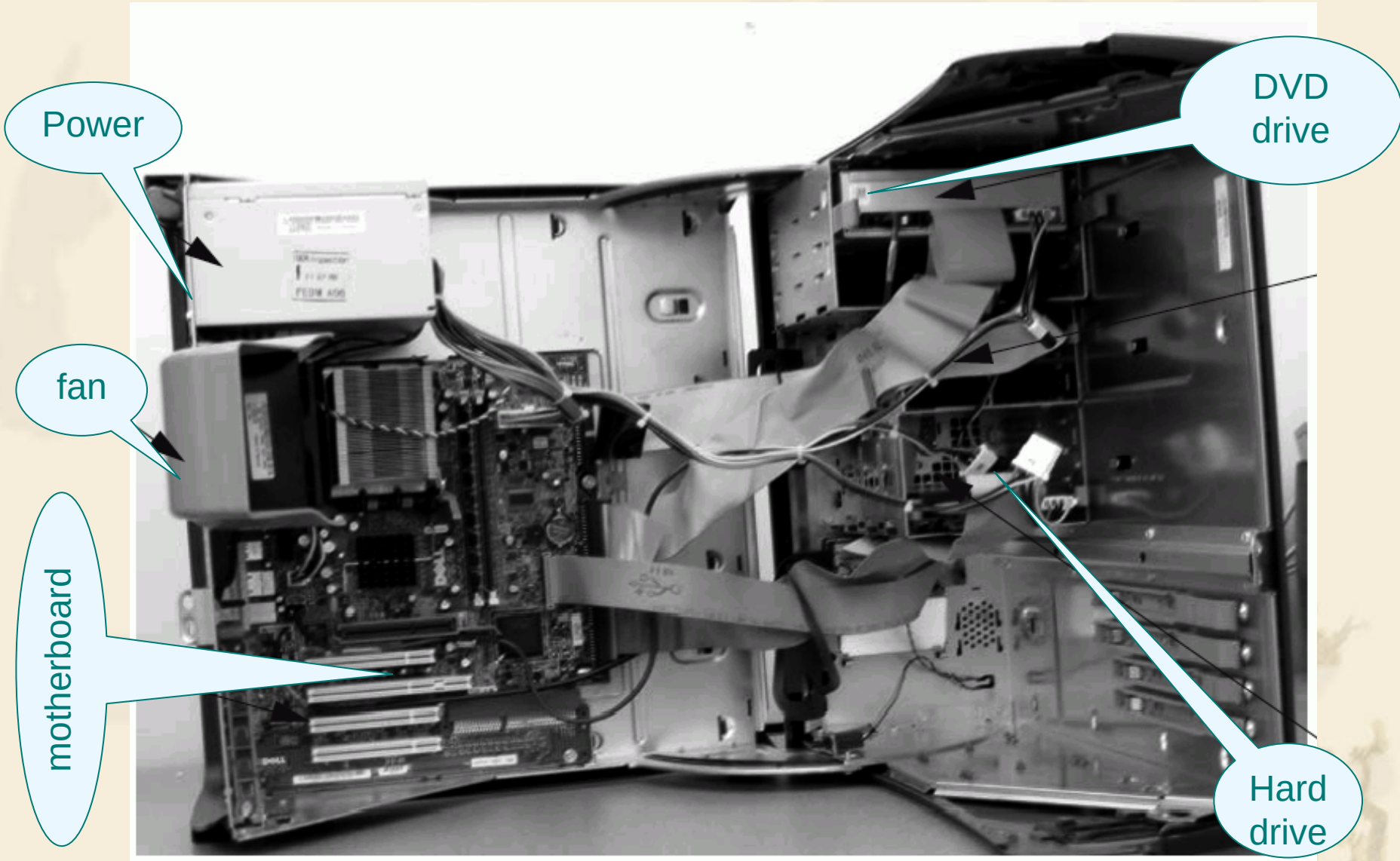
An example of the decomposability of computer systems



1.3 Under the Covers

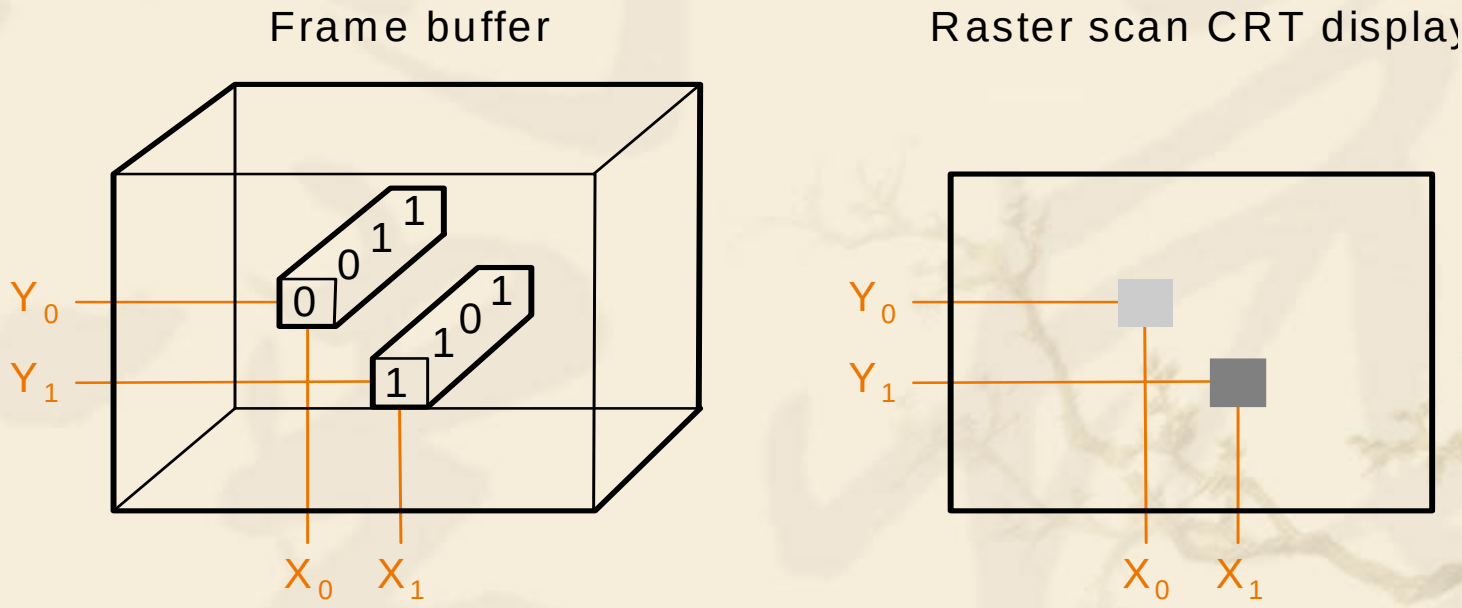
Computer Hardware System

- Mouse
  - The mechanical version
    - Moving the mouse rolls the large ball inside
    - The ball makes contact with an x-wheel and a y-wheel
    - Decide the distance and direction the mouse moves according to the rotation of wheels
  - The photoelectric version
    - Better orientation and better precision


Power
fan
motherboard
DVD drive
Hard drive

- Display
  - CRT (raster cathode ray tube) display
    - Scan an image one line at a time, 30 to 75 times / s
    - Pixels and the bit map, 512×340 to 1560×1280
    - The more bits per pixel, the more colors to be displayed
  - LCD (liquid crystal display)
    - Thin and low-power
    - The LCD pixel is not the source of light
    - Rod-shaped molecules in a liquid that form a twisting helix that bends light entering the display

- Hardware support for graphics ---- raster refresh buffer (**frame buffer**) to store bit map
- Goal of bit map ---- to faithfully represent what is on the screen

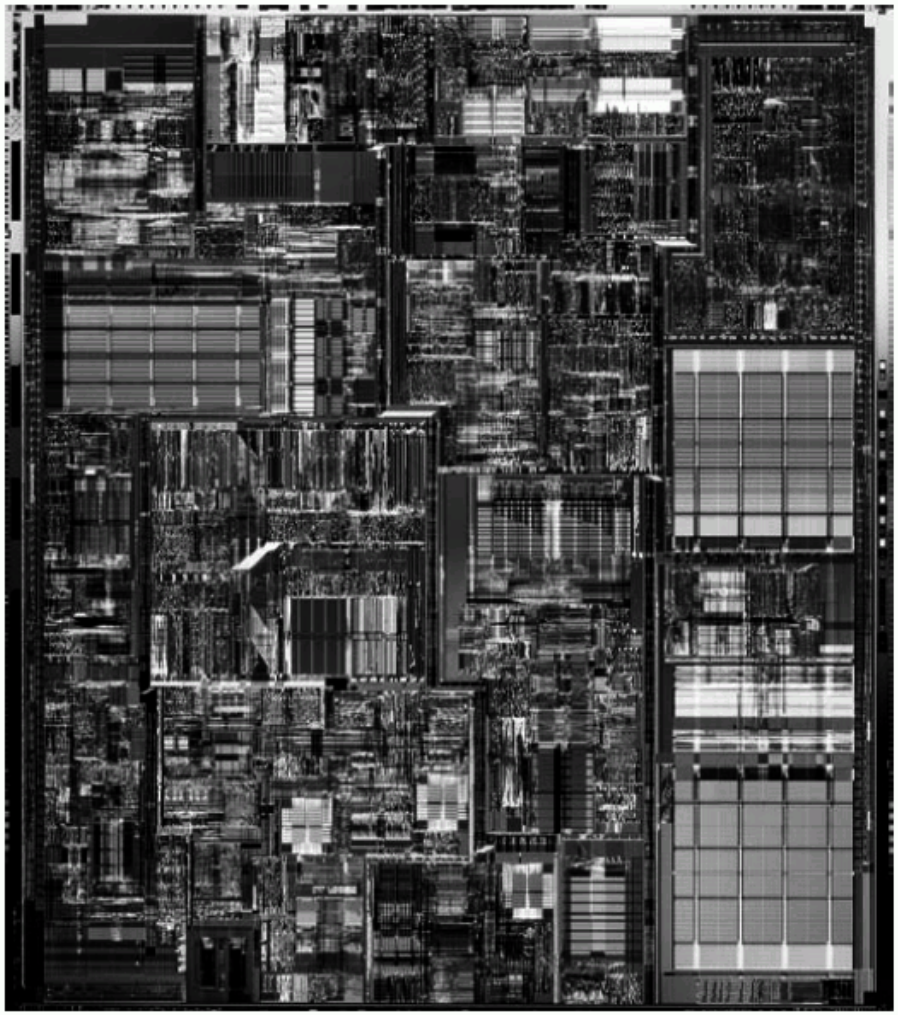


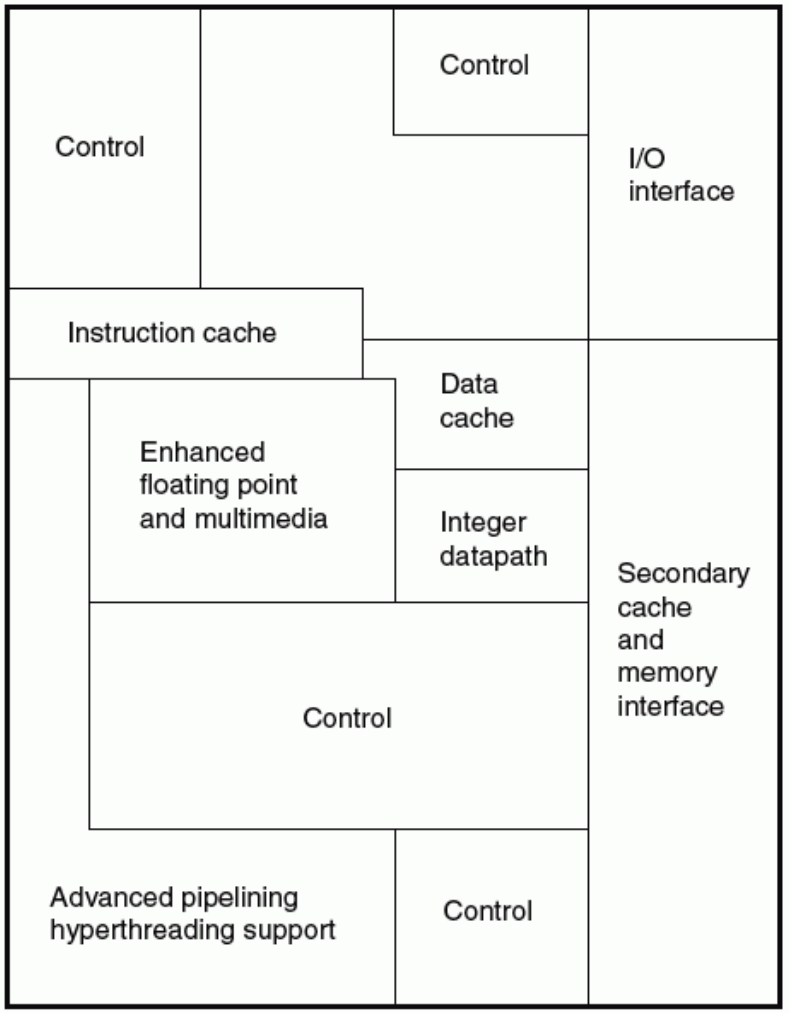
- Motherboard and the hardware on it
  - Motherboard
    - Thin, green, plastic, covered with dozens of small rectangles which contain **integrated circuits (chips)**
    - Three pieces: the piece connecting to the I/O devices, memory, and processor
  - Memory
    - Place to keep running programs and data needed
    - Each memory board contains some integrated circuits
    - DRAM and cache
  - Central Processor unit ----CPU
    - Add numbers, tests numbers, signals I/O devices to activate, and so on
    - CPU (central processor unit)

- Datapath
  - The component of processor that performs arithmetic operations
- Control
  - The component of processor that commands the datapath, memory, and I/O device according to the instructions of the program

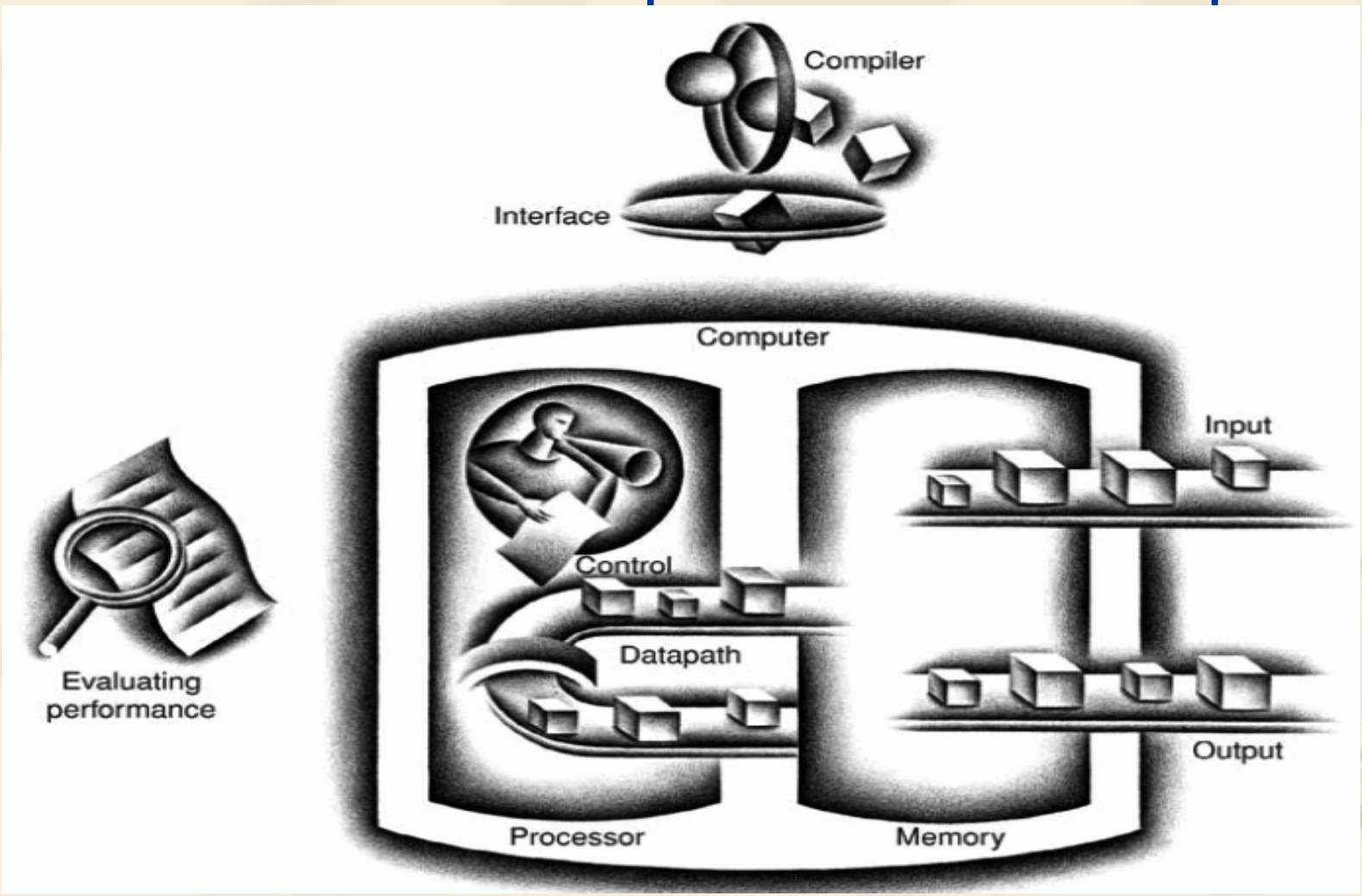
Motherboard

Inside the processor chip

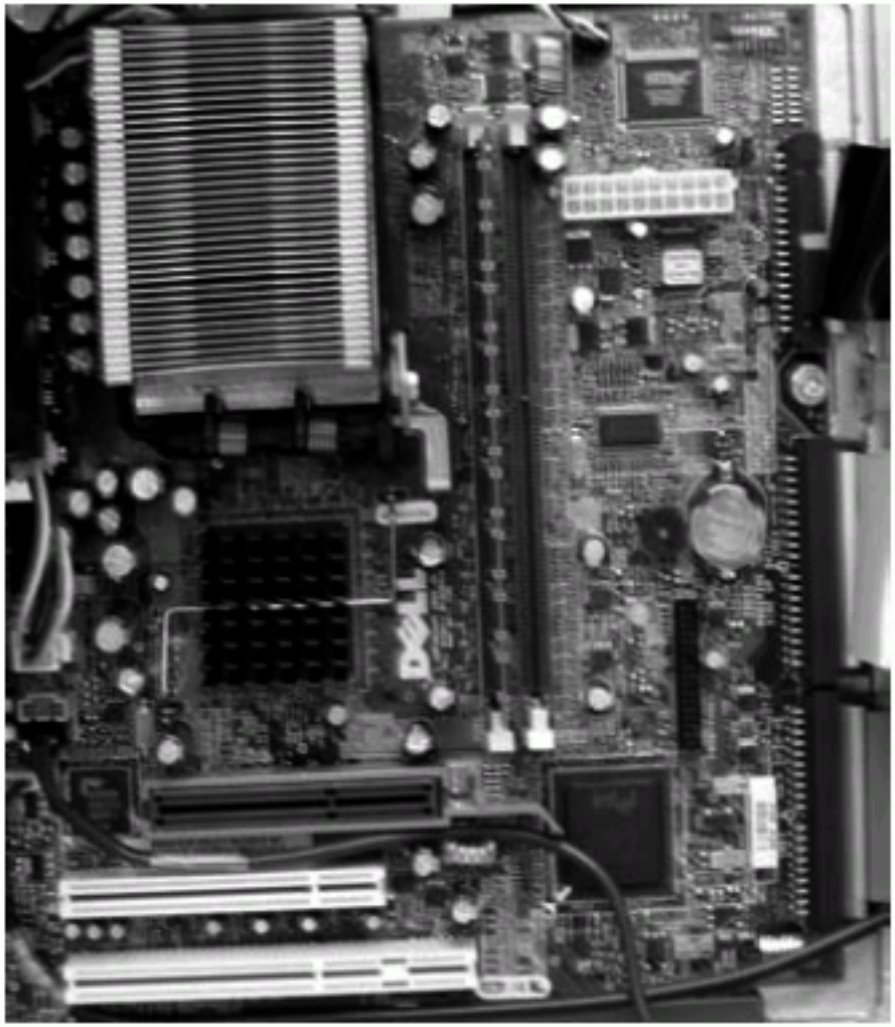


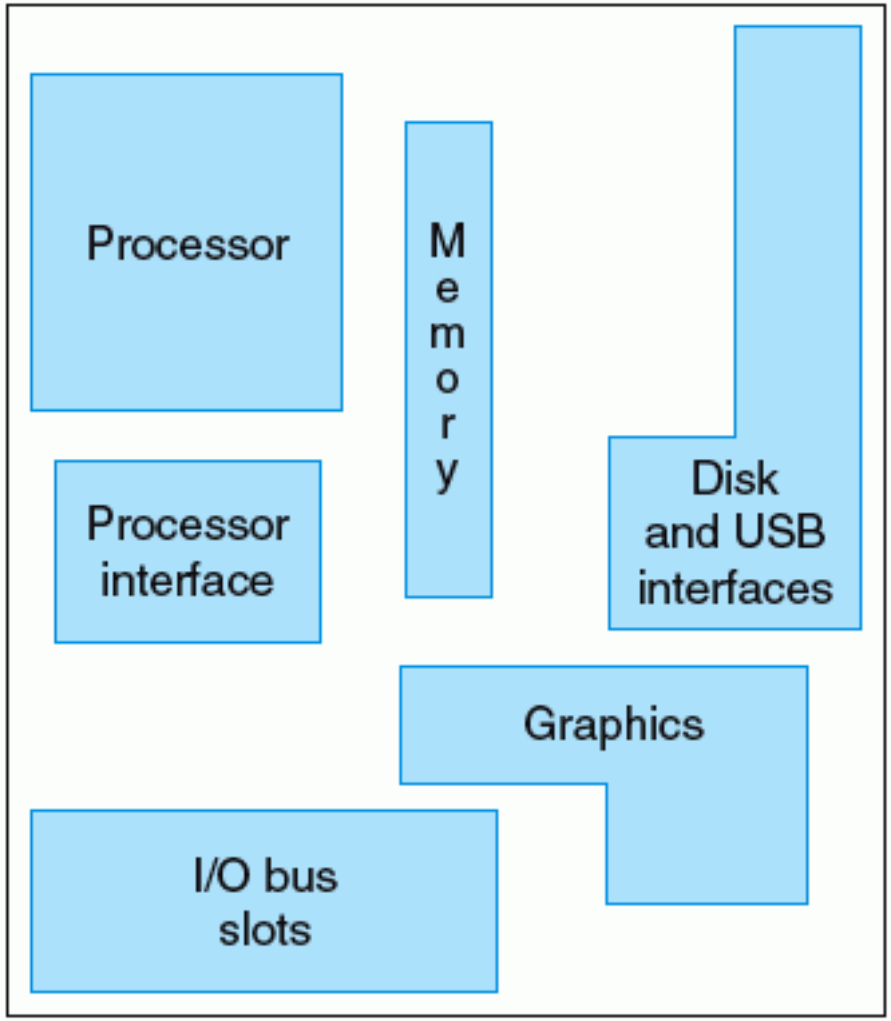


The five classic components of a computer



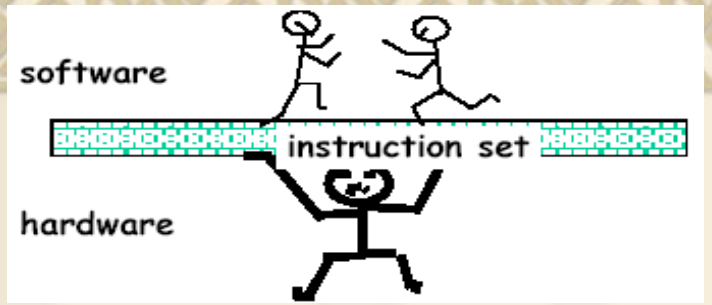
Close-up of PC motherboard



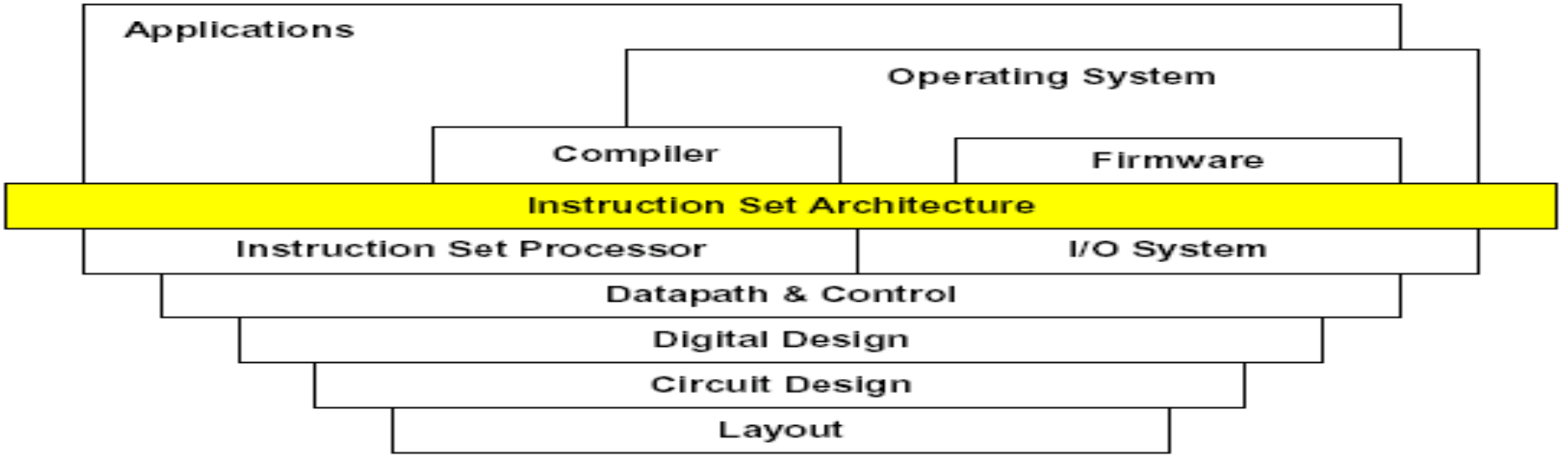


Important concept:

Virtual machine



- Abstractions
  - Lower-level details are hidden to higher levels
  - **Instruction set architecture** ---- the interface between hardware and lowest-level software
  - Many implementations of varying cost and performance can run identical software



- A safe place for data ---- secondary memory
  - Main memory is volatile
  - Secondary memory is nonvolatile
  - Magnetic disk
    - Rotating platter coated with a magnetic material
    - Floppy disk
      - Flexible mylar substance
      - 1.44~100MB
      - Removable
    - Hard disk
      - Metal
      - Mostly not removable
      - Rotate on a spindle at 3600 to 7200 r.p.m.
      - Read/write head and movable arm
      - Slower than DRAM, but cheaper for a given storage unit

- CD (optical compact disk)
- Magnetic tape

- Communicating with Other Computer
  -------Computer network
  - Communication----Information is exchanged
  - Resource sharing
  - Nonlocal access
  - LAN (local area network): Ethernet network
  - WAN (wide area network) :World Wide Web

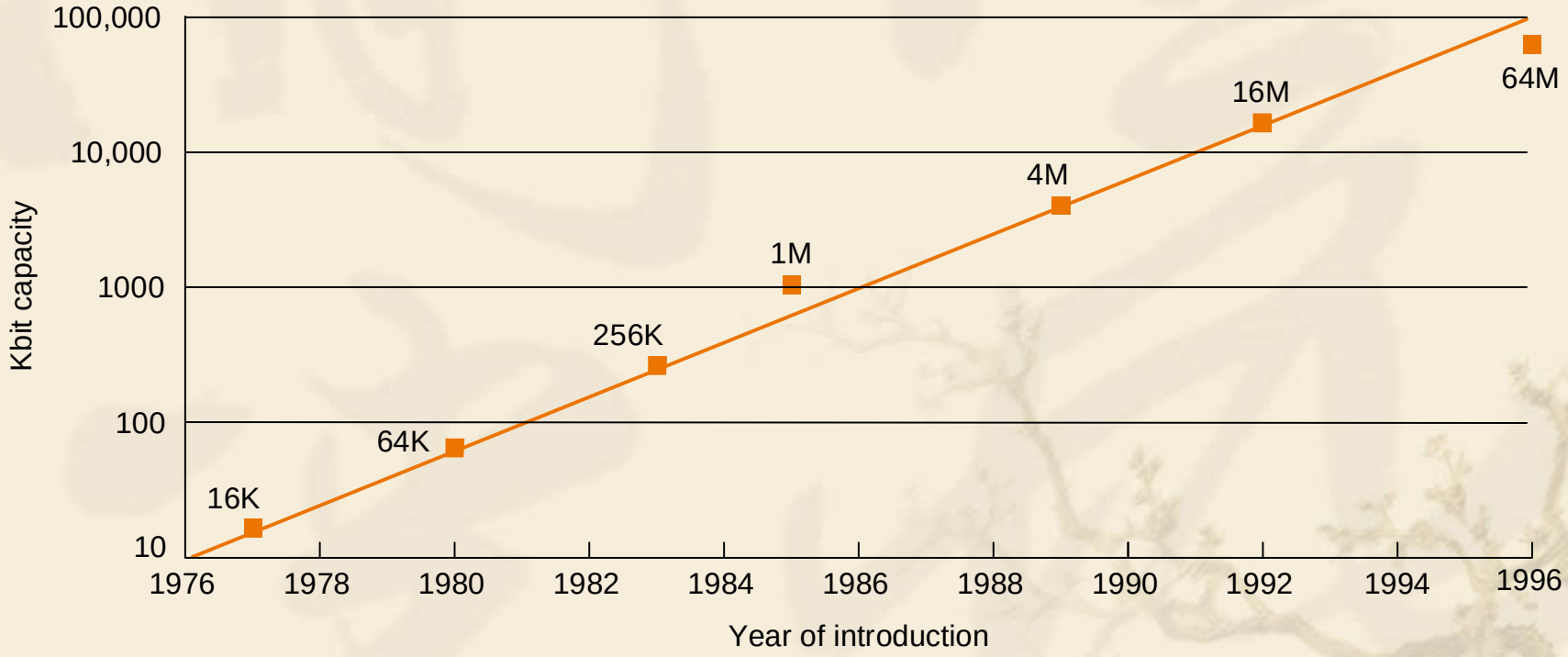
1.4 Real Stuff: Manufacturing Pentium 4 Chips

Semicoductor Integrated Circuits

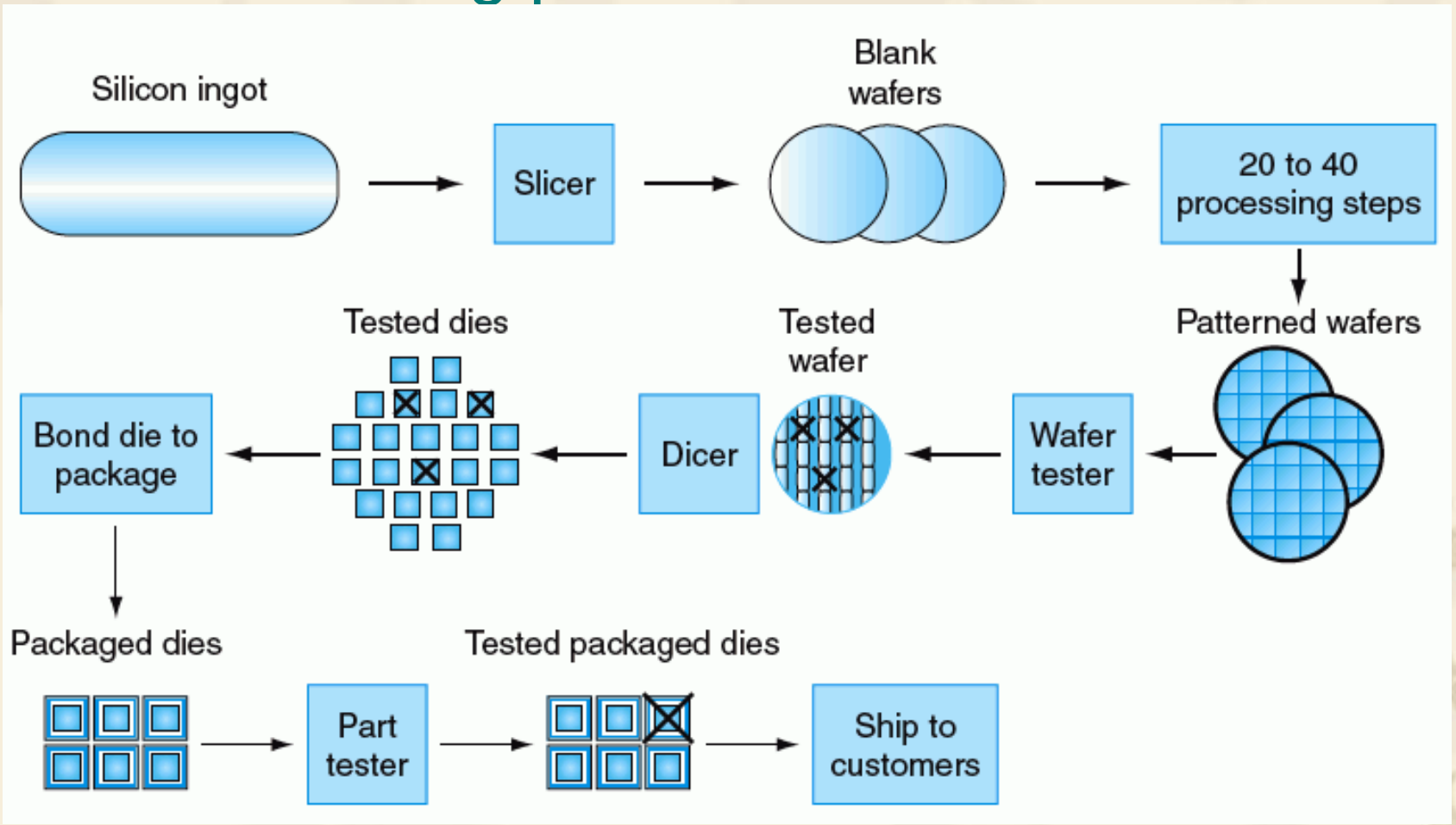
- Relative performance / unit cost of technologies used in computers

Year	Technology used in computers	Relative performance / unit cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated Circuit	900
1995	Very large-scale integrated Circuit	2,400,000

Growth of capacity per DRAM chip over time

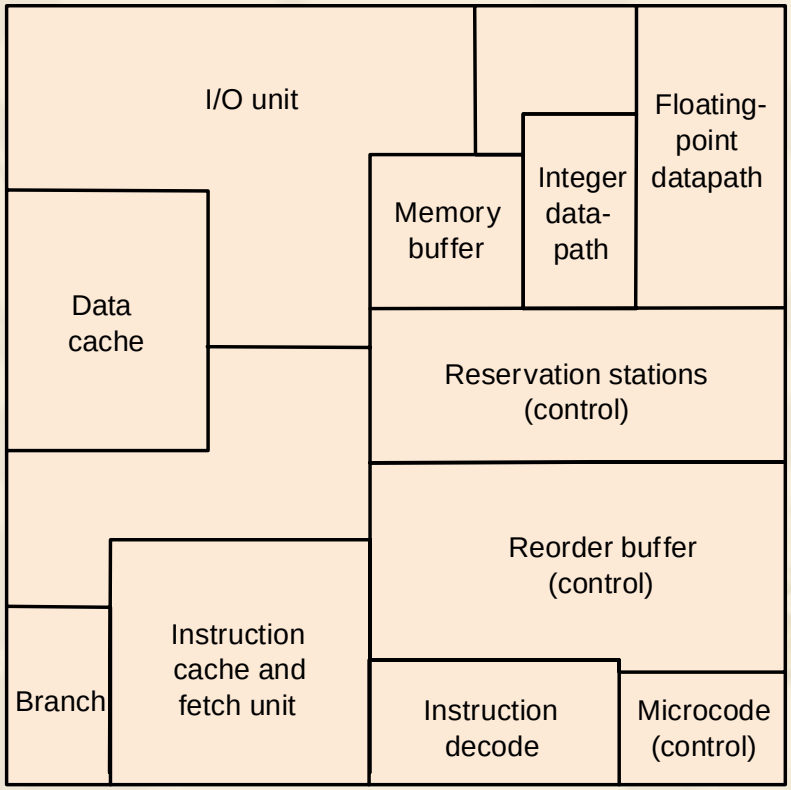


- The semiconductor silicon and the chip manufacturing process



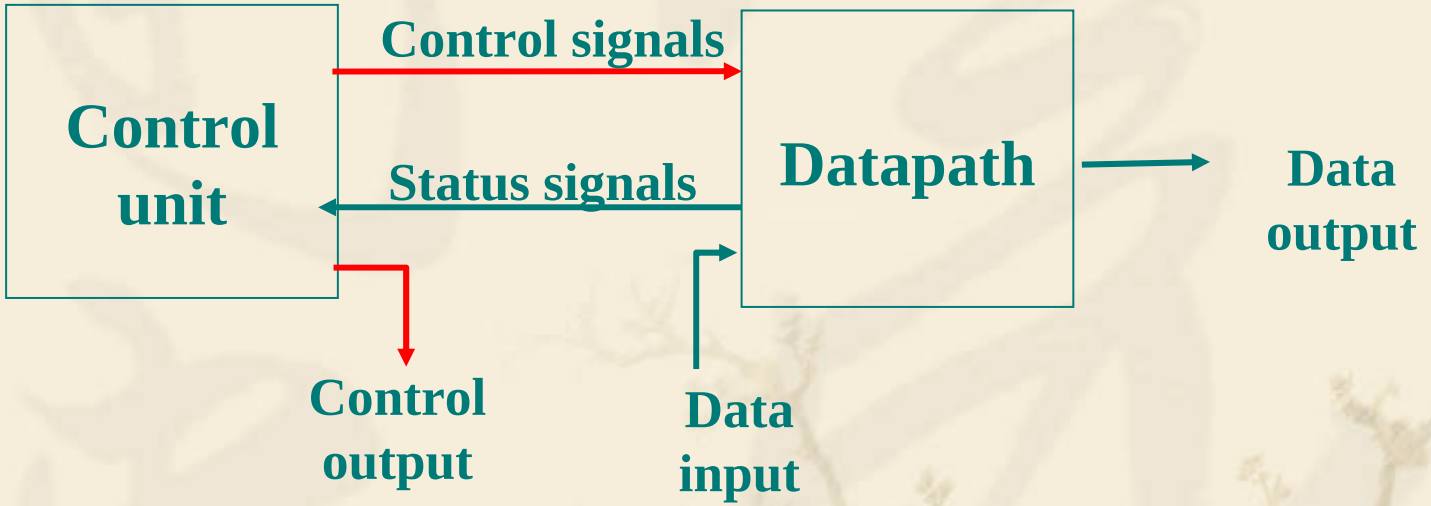
Manufacturing Pentium 4 Chips

- Major blocks of a Pentium Pro die

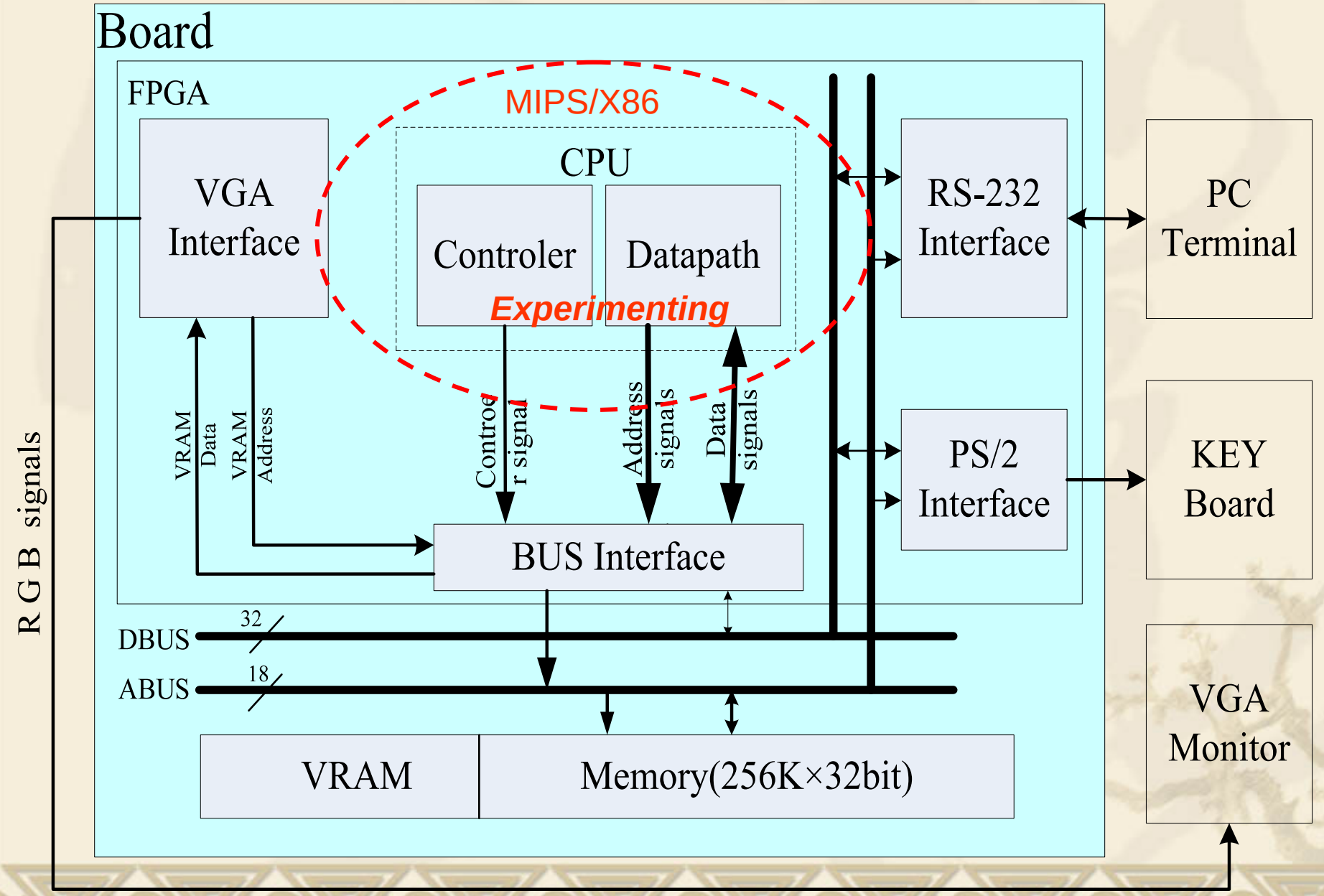


Digital circuits vs Computer organization

- Digital circuit
  - General circuits that controls logical event with logical gates (Hardware)



- Computer organization
  - Special circuits that processes logical action with instructions (Software)


Board
FPGA
MIPS/X86
CPU
VGA Interface
Controler
Datapath
Experimenting
RS-232 Interface
PC Terminal
PS/2 Interface
KEY Board
VGA Monitor
VRAM Data
VRAM Address
Control signal
Address signals
Data signals
BUS Interface
32
DBUS
18
ABUS
VRAM
Memory(256K×32bit)
R G B signals

1.5 History of Computer Development

- The first electronic computers
  - ENIAC (Electronic Numerical Integrator and Calculator)
    - J. Presper Eckert and John Mauchly
    - Publicly known in 1946
    - 30 tons, 80 feet long, 8.5 feet high, several feet wide
    - 18,000 vacuum tubes
  - EDVAC (Electronic Discrete Variable Automatic Computer)
    - John von Neumann's memo about stored-program computer
    - von Neumann Computer

- EDSAC (Electronic Delay Storage Automatic Calculator)
  - Operational in 1949
  - First full-scale, operational, stored-program computer in the world
- Other computers(omitted)
- Harvard architecture:
  Program memory and data memory are independent.

- Commercial Developments
  - Eckert-Mauchly Computer Corporation
    - Formed in 1947
    - $1 million for each of the 48 computers
  - IBM computers
    - First one, the IBM 701, shipped in 1952
    - Investing $5 billion for System/360 in 1964
  - Digital Equipment Corporation (DEC)
    - The first commercial minicomputer PDP-8 in 1965
    - Low-cost design, under $20,000
  - CDC 6600
    - The first supercomputer, built in 1963

- Cray Research, Inc.
  - Cray-1 in 1976
  - The fastest, the most expensive, the best performance/cost for scientific programs.
- Personal computer
  - Apple II
    - In 1977
    - Low cost, high volume, high reliability
  - IBM Personal Computer
    - Announced in 1981
    - Best-selling computer of any kind
    - Microprocessors of Intel and operating systems of Microsoft became popular

- Computer Generations
  - First generation
    - 1950-1959, vacuum tubes, commercial electronic computer
  - Second generation
    - 1960-1968, transistors, cheaper computers
  - Third generation
    - 1969-1977, integrated circuit, minicomputer
  - Fourth generation
    - 1978-1997, LSI and VLSI, PCs and workstations
  - Fifth generation
    - 1998-?, micromation and hugeness