

微型原理与接口技术

Hardware Principles and Interfacing of Modern Computer

Lecture 1: Introduction

陈启军，张 伟

Email: zhang_wi@mail.tongji.edu.cn

Dept. Of Automation, TongJi University



Contents

- 认识计算机
- 从**Moore's Law**谈起——兼谈计算机的发展历史
- 计算的基本原理——从**Turing Machine** 和 **Von. Neumman** 机谈起
- 未来趋势：迈向高可用性计算之路
- 计算机领域中一些常见的思想
- 背景补充：数的进制及其转换
- 实例分析兼课本内容回顾
- 小结

Reference

- 🌐 《Intel微处理器全系列》结构、编程与接口（第六版）
The Intel Microprocessors: 8086 / 8088...: Architecture, Programming, and Interfacing, Sixth Edition, [美]Barry B. Brey（巴里 B.布雷）著，机械工业出版社，2005年4月. 书号：ISBN 7-111-16052-5

Section Objectives

- 计算机系统工作的**最基本原理**
- 计算机技术的 **最基本发展规律**
- 掌握有关基本名词和术语 —— 我们的“行话”

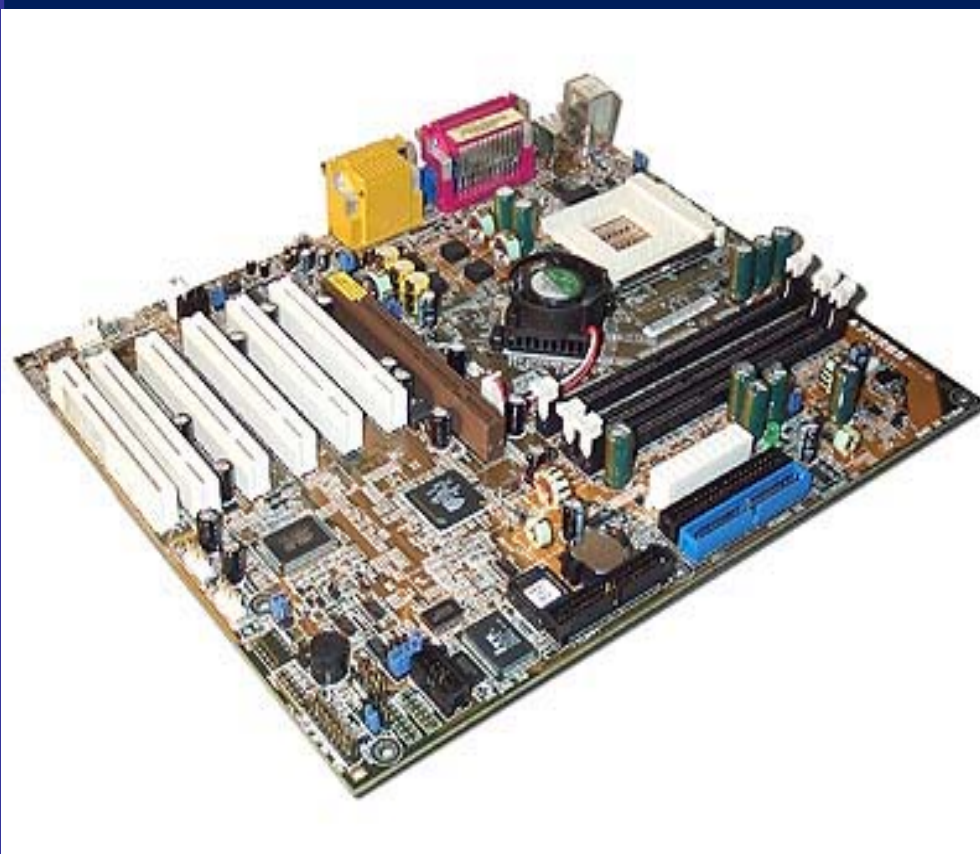
认识计算机

主板

模块化的设计，分为许多个功能块，每个功能块由一些芯片或元件来完成。大致说来，主板由以下几个部分组成：CPU插槽，内存插槽，高速缓存、局域总线和扩展总线、硬盘、软驱、串口、并口等外设接口；时钟和CMOS主板BIOS控制芯片等。

扩充插槽

可插入所需要的扩展卡
(并行/串行/鼠标/传真/MODEM/AUDIO/VIDEO)



认识计算机

历史上的庞然大物

世界上第一台现代意义的电子计算机是1946年美国宾夕法尼亚大学设计制造的“ENIAC”

- 占地上百平方米
- 重量几千吨
- 功耗几十千瓦



认识计算机

历史上的尝试

- 500 B.C. abacus
- 19th, Babbage's dream of Analytical Engine(分析机)
 - 提出了用punched card进行输入的设计
 - Later in 1889, Hollerith developed the punched card for storing data
- 1946, ENIAC(Electronics numerical integrator and calculator, 电子数值积分计算器), is the first general purpose, programmable electronic computer system.
 - 1948 Bell labs invents transistors
 - 1958 Jacky Kilby of TI invents Integrated Circuits (IC)

认识计算机

- In the early 1950s, UNIVC introduces Assembly language
 - Later, IBM invents the first high level language FORTRAN. And since then, COBOL, BASIC, ADA, PASCAL, Smalltalk, C, C++, Java, C#.....
- 1971, Intel produces the first micro-processor 4004

代	发表年份	字长(bits)	型号	线宽(μm)	晶体管数(万个)	时钟频率(MHz)	速度(MIPS)
一	1971	4	4004	50	0.2	<1	0.05
	1972	8	8008		0.3		
二	1974	8	8080	20	0.5	2-4	0.5
三	1978	16	8086/8088	2-3	2.9	4.77-10	<1
	1982		80286		13.4	8-16	1-2
四	1985	32	80386	1-2	27.5	16-33	6-12
	1989		80486		120	25-66	20-40
五	1993	32	Pentium	0.6-0.8	330	60-200	100-200
六	1995	32	P/Pro	0.6	550	133-200	>300
	1996		P/MMX	0.6	450	166-233	
	1997		PII	0.35	750	233-450	
	1999		PIII	.25-.13	850	450-1200	
	2001		P4	.18-.13	1000	1300-2400	
七	2002 ?	64	Itanium	0.13	CPU:2.5K Cache:30K	800(20条指令/时钟周期)	>3000

从摩尔定律谈起

——兼谈计算机的发展历史

● “懂一点历史是有好处的！”

● 摩尔定律

- 集成电路芯片的线宽每年减小10%，芯片的规模每18个月翻一番，芯片的性能也随之提高一倍（或者说3年增长到4倍）
- 美国化学家，Gordon. Moore, 1965
- 摩尔定律是一个**精确的规律**而不是模糊的估算！

例如：MOS芯片存储器容量

1971, 1Kb, 1974, 4Kb, 1978, 16Kb, 1981, 64Kb, 1984, 1Mb, 1987 16Mb, 1990 64Mb, 1993 256Mb

从摩尔定律谈起

Moore's Law的意义

- 它是社会发展动力中技术驱动力的体现
- 它是对IT产业未来的准确预测，因此是指导公司和科研机构确定研发计划和市场策略的重要依据！
- 例：Intel是如何决定它的研发投入、研发进度和市场策略的？

从摩尔定律谈起

思考1: Moore's Law是如何得来的?

- Moore对1950-1965年IC生产企业进行调查并分析有关数据得到
- 如何分析的呢?
 - 回归分析技术(Regression)
- 进一步延伸: 对IT产业的其它指标, 例如IT产业的从业人数和产值, 购买相同计算能力和存储设备的投资, 是否也有类似的发展规律?

从摩尔定律谈起

思考2: Moore's Law对我们自己和社会的影响

- Wow, 可以买更好更快的机器, 爽啊! :)
- 每一次的技术变革都:
 - 为新兴公司的成长提供了机会!
 - 为同济大学的毕业生提供了就业可能!
 - 为貌似强大、因循守旧的传统公司敲响了丧钟!

例: Cray, Amdahl, IBM, DEC

启示: 在变革中寻求机遇!

- Moore's law告诉了我们实际的发展规律, 但是并没有说明提高的计算能力被用于何处!
 - => 赋予了我们思考的空间

从摩尔定律谈起

思考3: Moore's Law会不会失效? 何时会失效?

- 技术的发展看似越来越困难, 但是每年都会有新的技术诞生! 导致 Moore's law仍然有效!
- Moore's Law失效之后的世界是怎样的? 对IT产业的发展乃至我们的就业会有怎样的影响?
- Gordon Moore的再次预言: 30年!

安迪-格罗夫(Andy Grove):
Intel 三驾”马车”之一
Only the paranoid survive!

从摩尔定律谈起

● “物有本末，事有终始。知所先后则近道矣”（《大学》）

- 世界上的万事万物都有自己的发展过程及因果关系。知道了这个过程和关系，也就触摸到了事物的发展规律。

● 让历史告诉未来！

- 学习历史的用意在于发现其中的规律，并据此对未来尽心预测，从而指导我们的活动。

从摩尔定律谈起

 **Discussion: What will the world be in the future according to Moore's Law ?**

从摩尔定律谈起

思考4：根据Moore's Law预测未来

有多小？	Smart Dust Project, MEMS
有多快？	光速互连器件/光交换器件，量子计算，GHz器件，无限并行与扩展
有多强？	海量数据要瞬间能够传输和处理完毕，例如Internet 2计划，人体三维影像重建数据处理，地球大气数据处理，海洋数据处理，地质勘测数据处理
有多方便（智能程度）？	“计算无处不在”，与生物体结合，机器学习，仿生物机器人
有多便宜？	一次性使用，随手可丢弃（与环保的矛盾）

计算的基本原理

——从Turing Machine谈起

图灵机(Turing Machine)

- 图灵机由三部分组成：一个可以无限延伸的纸带，一个读写头和一个控制器
- 图灵机是一种思想模型
 - 类比 状态机思想
- 图灵测试
- ACM Turing Award

计算的基本原理

冯·诺伊曼机(Von. Neumann)

- 两个突出的创新点：程序存储 和 程序控制
- 五大部件(Parts):
- 存储器(MEM), 运算器(ALU), 控制器(CU), 输入设备和输出设备
 - $CPU = ALU + CU$

计算的基本原理

VN结构中被忽略的一个重要部件——总线(Bus)

- 总线的作用：部件互连
- 总线设计的思想：
 - 接口与实现：总线接口以协议的形式体现
 - 分时共享
- 总线按功能分类
 - 地址总线，数据总线和控制总线
 - 改进的Harvard结构：数据总线和指令总线

计算的基本原理

指令的执行流程

- 以一条指令为例，体会五大部件以及总线的活动流程
- 前提：CPU已知指令的地址
- 常见的几个阶段：
 - **取指**：CPU通过地址总线AB将地址送给MEM，并通过CB发送读指令给MEM，然后等待MEM将指令放到数据总线DB上；
 - **取操作数**：CPU根据指令中指出的寻址方式计算操作数地址，仿照上述操作从MEM读出操作数；
 - **指令执行**：主要是在CPU的ALU部件中执行；
 - **保存操作数**：为取操作数流程的反向操作
 - **计算下一条指令地址**：

计算的基本原理

思考1：指令是什么样子的？指令是什么？

- 指令：计算机可以识别的命令
- 从高层到低层可以有：
 - 第四代高级语言指令：高级语言指令
SELECT * FROM PRODUCTS (SQL)
 - 第三代高级语言指令：
Counter++; (C language)
 - 汇编语言指令(二进制指助记码)：MOV AX, 08H
 - 机器语言指令(二进制指令)：
10110000,00001000,10110011,00000101
- 计算机裸机仅能识别二进制的机器语言指令。从上一层指令到下一层指令必须经过“翻译”的过程才能真正被计算机执行

计算的基本原理

思考2：指令序列（即程序）是如何连续执行的？

– CPU中的寄存器PC和IR

- 程序计数器PC (Program Counter)中记录了下一条即将被执行的指令的地址
- 指令寄存器IR (Instruction Register)中记录了当前正在被执行的指令

思考3：计算机裸机加电时，CPU如何知道第一条指令在哪里？

思考4：计算机执行机器指令的速度有多快？

– 概念：频率，时钟和周期，指令周期的概念

迈向高可用性计算

● 概念：高可用性计算(High availability)

- 高性能(high performance computing)
- 普适(ubiquitous computing): 嵌入 + 移动 + 低功耗
- 高可靠性 (略)
- 高安全性 (略)

● 通往高性能计算之路——突破现有约束！

- 器件(Component)技术的改进，例：更高的IC集成度，更高的频率
- 部件(Parts)的改进，例：更高吞吐量的内存和总线
- 系统结构的改进
- 并行和分布式技术
- 能量约束

迈向高可用性计算

思考1：常见的部件改进技术

- 针对特定应用增加相应的指令，例如DSP芯片中的FFT指令，Multimedia芯片中的多媒体处理指令
- 软件功能硬件化：例如在高端MCU中增加专门的乘法器代替软件乘法程序代码，用硬件看门狗代替软件看门狗
- 在系统的不同层次之间增加Cache，以缩小不同层次之间的性能差异
- 追求各个部件之间的性能均衡

迈向高可用性计算

思考2：常见的系统结构改进技术

- 三大思想：资源共享，时间复用和空间复用
- 指令流水技术
- 超标量技术
- 超长指令字
- CISC和RISC的历史之争和今天的走向融合

迈向高可用性计算

- 补充说明：高可用性计算发展的两个主要方向：
 - 高性能计算(high performance computing): 20世纪的发展主题，今天仍然有很大市场和空间。许多过去的高性能计算研究中诞生的成果今天已经被或正在被用于各种低端计算设备，这种规律在以后仍然有效。
 - 普适计算(ubiquitous computing): 移动计算+嵌入式计算(mobile computing and embedded computing): 今天的主题

问题和思考：移动计算/嵌入式计算中的一个热点问题
在满足功能性要求(functionality)和可接受性能(reasonable performance)的前提下，如何尽可能降低设备或者系统的功耗，延长使用时间？

计算机领域中一些常见的思想

思想1：接口与实现

(Interface & Implementation)

- 不同实现可以拥有相同接口，从而方便了计算机部件的更换、升级
- 硬件模块的接口体现为电气信号的时序和交互
- 软件模块的接口体现为函数声明和函数交互时序
- 对接口的规定，常常体现为协议的形式，特别是在总线技术中

计算机领域中一些常见的思想

思想2：层次化思想

– 例1：虚拟机器思想

- 应用语言机器M5
- 高级语言机器M4
- 汇编语言机器M3
- 裸机M1

– 例2：网络的分层体系

– 例3：存储系统的分层体系

- 在不同的层次之间，可以通过cache弥补性能差异而接口保持不变
- 压缩层次有助于提高性能，例如OSI 7层和TCP/IP 4层

计算机领域中一些常见的思想

思想2：层次化思想

- 为了方便，增加层次，从对机器友好到对人友好
- 例：MSRA在智能人机交互方面的工作，Lee Kaifu的agent目标
- 为了性能，减少层次：IC Design, DSP, SOC
- 在具有相同接口、不同性能的层次之间可以插入cache缓冲
 - Cache命中率和性能的问题

计算机领域中一些常见的思想

思想3：互连思想

- 三种基本的互连手段：总线(Bus)，交换(Switch)和无线(Wireless)
 - 总线：共享传输媒介，需要对媒介访问进行控制；一般都比较简单，成本低
 - 交换：是对总线思想的改进；高性能，高成本
 - 网络技术中交换技术的广泛应用
 - 未来可能的全光交换在计算机中的应用
 - 无线：本质上是共享无线媒介，仍然需要解决信道征用问题；非常灵活
- 其它的分类标准：同步异步，串行并行

计算机领域中一些常见的思想

思想4：智能化

- 让每一个部件都自主的决定自己的运行

- 例：简单IO => DMA技术 => SCSI 协议 => IOP

计算机领域中一些常见的思想

思想5：性能瓶颈与性能平衡

- 追求多个部件之间的平衡，这样可提高整体性能，提高资源利用率；
- 消灭少数部件所带来的瓶颈

计算机领域中一些常见的思想

思想6：资源共享与分时复用

- 资源：例 CPU的计算能力
- 目的：提高资源利用率

计算机领域中一些常见的思想

思想7：并行化，分布化，网络化

- 并行化即空间复用，借助多个部件同时执行任务以提高性能；例如：多CPU机器
- 并行化的更高层次是分布化，如IBM AS400的集群系统，最大支持16台机器
- 小规模分布化之上是网络化，例如GRID 计算

计算机领域中一些常见的思想

思想8：虚拟

- 虚拟的本质在于保持接口不变
- 例如：虚拟光驱(Daemon)，虚拟内存(Virtual Memory)，虚拟计算机(VMware)，虚拟机(Java VM, C# VM)，网络虚拟硬盘

实例分析兼课本内容回顾

- **概念：机器周期和指令周期 (P8)**
 - 思考：这两个周期具有怎样的关系？
- **概念：指令执行的周期/频率 和 内存访问频率**
 - 实际中：倍频技术 (double-clock)
- **性能衡量指标 (P11)**
 - MIPS和MFLOPS
 - iCOMP2 rating index (Intel) (P11)
 - Benchmark Package
- **Memory map: 描述Memory是如何被使用的 (P18, 20)**
- **TPA和XMS (transient program area & extended memory system), EMS(expanded memory system) (P15, 21), Upper Memory (P22)**
- **Memory Space & I/O Space => 内存地址空间和I/O地址空间的关系 (P22)**

Lecture 1 总结

- 关键知识点
- Moore's Law
- Von. Neumann结构
- 指令的概念
- 计算机指令的动态执行流程