

计算机组织结构

0 课程简介

刘博涵

2022年9月6日



南京大學
NANJING UNIVERSITY

分班教学



任桐炜

1班



刘博涵

2班

相同

- 起始资料：均以2021年课程资料为基础
- 教学过程：大致的讲授顺序、作业答案公布
- 考核评分：相同的考核时间、形式和内容

不同

- 讲授细节：授课的局部顺序、内容等
- 改革思路：课程改革的方向和办法（期末后讨论融合）

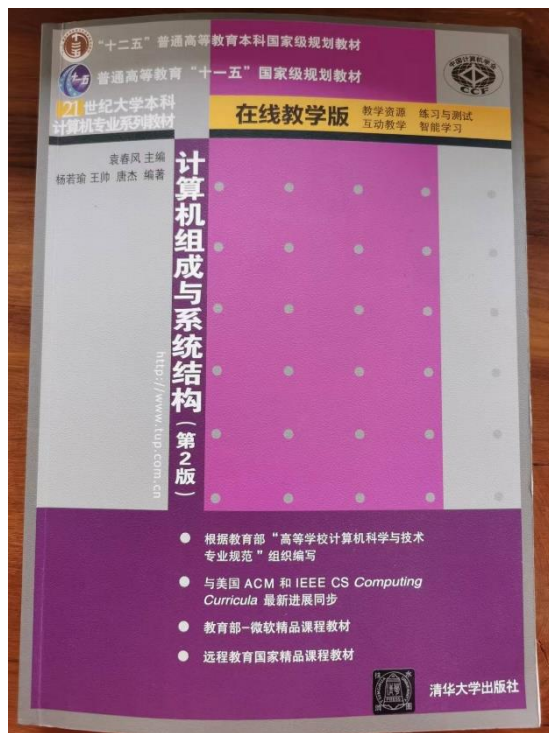


教师 & 助教

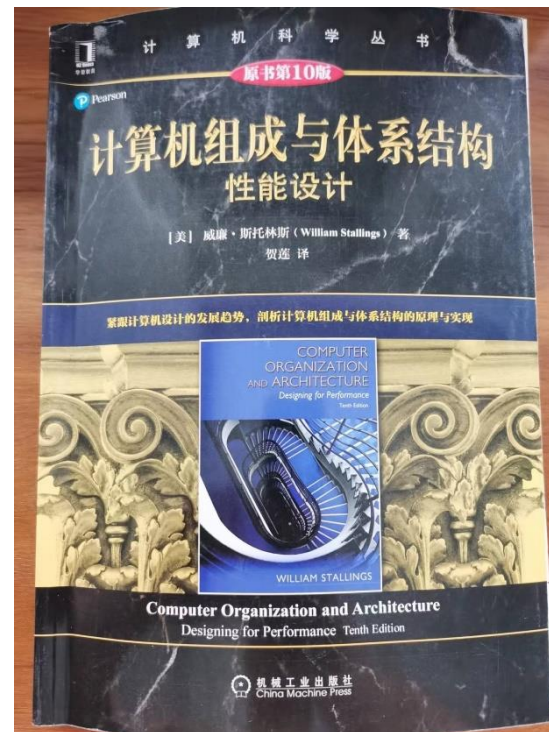
- 教师：刘博涵 助理研究员
 - 实验室：软件研发效能实验室
 - 研究兴趣：经验软件工程，软件过程仿真，可信AI等
 - 联系方式：课上交流 或 发邮件至bohanliu@nju.edu.cn
 - 邮件主题：**COA22: ******
- 助教：马维刚 硕士二年级
 - 联系方式：1975126910@qq.com



教材



袁春风 主编，杨若瑜、王帅、唐杰 编著. 计算机组成与系统结构 (第2版)

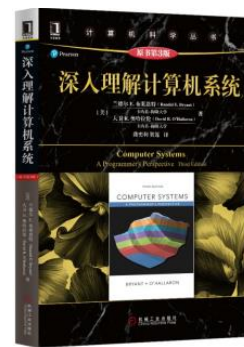


William Stallings.
计算机组成与体系结构：性能设计 (第9版)



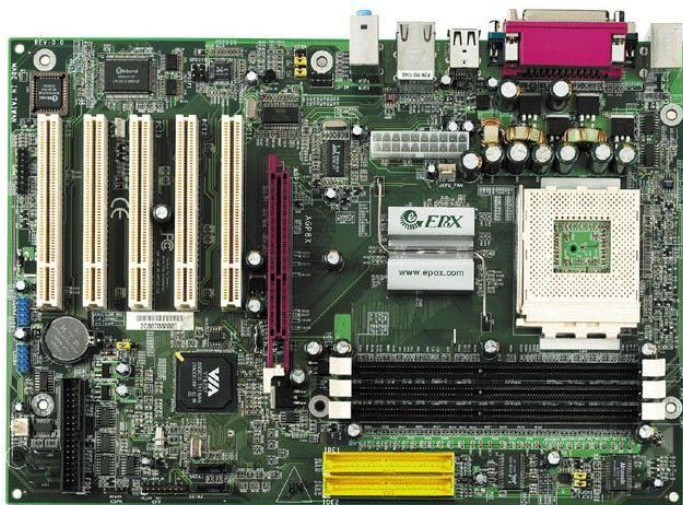
前续课程

- 计算系统基础
 - 房春荣 老师 和 何铁科 老师
- 参考书
 - 陈道蓄 主编, 王浩然、葛季栋 编著. 计算系统基础
 - 袁春风、余子濠 编著. 计算机系统基础 (第2版)
 - Randal E.Bryant 著. 深入理解计算机系统 (第3版)

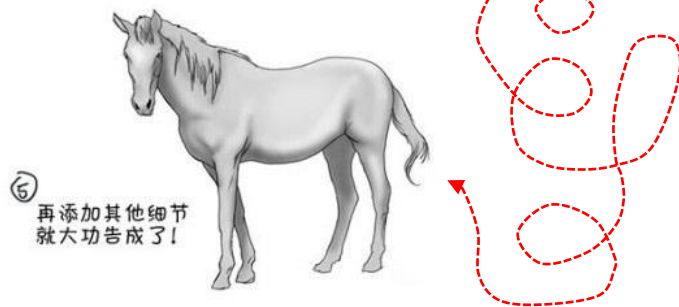
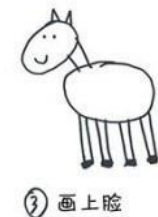
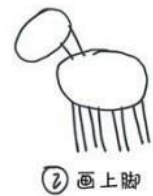
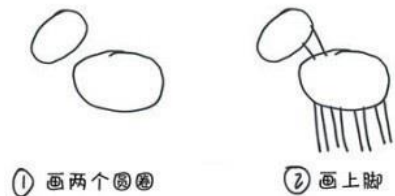


教学目标

- 更多了解计算机构成和如何工作
- 初步了解计算机性能
- 为解决编程问题提供更多思路



怎样画马



教学大纲

导论

- 第01讲：计算机系统概述
- 第02讲：计算机的顶层视图
- 第03讲：数据的机器级表示

中央处理器 (CPU)

算术

- 第04讲：整数算术
- 第05讲：浮点数算术
- 第06讲：十进制算术

- 第14讲：指令系统
- 第15讲：指令周期和指令流水线
- 第16讲：控制器

存储

- 第07讲：内部存储器
- 第08讲：高速缓冲存储器 (Cache)
- 第09讲：外部存储器
- 第10讲：数据校验码
- 第11讲：磁盘冗余阵列 (RAID)
- 第12讲：虚拟存储器

总线

- 第13讲：总线

输入 / 输出

- 第17讲：输入 / 输出



教学计划

课堂讲授
+
习题讲解
(7次课)

课堂讲授 (17次课)

日期 星期 年 月 周次			一	二	三	四	五	六	日
二 零 二 年	九 月	1 单	5	6	7	8	9	中秋	11
		2 双	12	13	14	15	16	17	18
		3 单	19	20	21	22	23	24	25
		4 双	26	27	28	29	30		
	十 月	5 单	节	4	5	6	7	8	9
		6 双	10	11	12	13	14	15	16
		7 单	17	18	19	20	21	22	23
		8 双	24	25	26	27	28	29	30
		9 单	31						
	十一 月	10 双		1	2	3	4	5	6
		11 单	7	8	9	10	11	12	13
		12 双	14	15	16	17	18	19	20
		13 单	21	22	23	24	25	26	27
	十二 月	14 双				1	2	3	4
		15 单	5	6	7	8	9	10	11
		16 双	12	13	14	15	16	17	18
			19	20	21	22	23	24	25
二 一	考试		26	27	28	29	30	31	
			2	3	4	5	6	7	8
									元旦



评分标准

- 书面练习：10%（仅限按时提交）
- 编程练习：20%（仅限按时提交）
- 上机考试（期中）：30%（考2次，取成绩高的那次）
- 书面考试（期末）：40%
- 奖励分数：10%
 - 对课程建设具有实质性贡献

所有修读该课程的同学采用统一评分标准

即便重修/免修不免考的同学也需要按时提交所有作业和参加考试



教学平台

教学平台：腾讯会议（线上教学）+ **教学立方**（书面练习和课件）+ Seecoder（编程练习和考试）

学生通过邀请码加课

"教学立方"公众号二维码



本课程的课程邀请码:**MP3GG7D6**

请告知学生:

关注以上二维码，注册/登录后通过邀请码加入本课程。



这门课难在哪里？



这门课难在哪里

设计计算机系统以获得高性能向来是非常重要的要求，但这个要求从来没有像现在这样强烈和难以满足。计算机系统的所有基本性能特征，包括处理器速度、存储器速度、存储容量和互联数据速率都在迅速提高，并且在以不同的速度提高。我们总是想设计出均衡的系统，它可以充分发挥所有元素的最佳性能和全部价值，但各种技术不平衡的发展速度使我们的目标难以实现。因此，计算机设计越来越成为一个补偿游戏，在某个方面改变结构或功能，以补偿另一个方面的性能不足。我们将在许多设计决策中看到这个让人筋疲力尽的游戏。



课堂纪律

- 不要影响授课正常进行
- 积极参与（教学过程管理要求）



鸣谢



本课件原作者
南京大学
任桐炜 教授



教材作者
慕课：计算机系统基础
南京大学
袁春风 教授



慕课：计算机组成
北京大学
陆俊林 副教授



计算机组织结构

1 计算机系统概述

刘博涵

2022年9月6日



南京大學
NANJING UNIVERSITY

教材对应章节



第1章 计算机系统概述



第1章 导论 第2章 计算机的演变和性能

计算机无处不在



什么改变了，什么没有变？

回顾: 什么是计算机?



什么是计算机？

- 计算机是指 “**通用电子数字计算机**（ general-purpose electronic digital computer ） ”
 - **通用**：不是一种专用设备
 - 所有计算机在给予足够时间和容量存储器的条件下，都可以完成同样的计算
 - 当希望完成新的计算时，不需要对计算机重新设计
 - **电子**（非机械）：采用电子元器件
 - **数字**（非模拟）：信息采用数字化的形式表示
- 计算机系统
 - **硬件**：处理器，存储器，外部设备，
 - **软件**：程序，文档，



什么是“组织”与“结构”？



计算机系统抽象层

计算机体系结构即为ISA

ISA是对硬件的抽象
所有软件功能都建立在ISA之上

**微体系结构即为
计算机硬件组成**



组织与结构

组织 (Organization) : 对编程人员不可见

- 操作单元及其相互连接
- 包括：控制信号，存储技术，.....
 - 例如：实现乘法是通过硬件单元还是重复加法？

结构 (Architecture) : 对编程人员可见

- 直接影响程序逻辑执行的属性
- 包括：指令集，表示数据类型的位数，.....
 - 例如：是否有乘法指令？



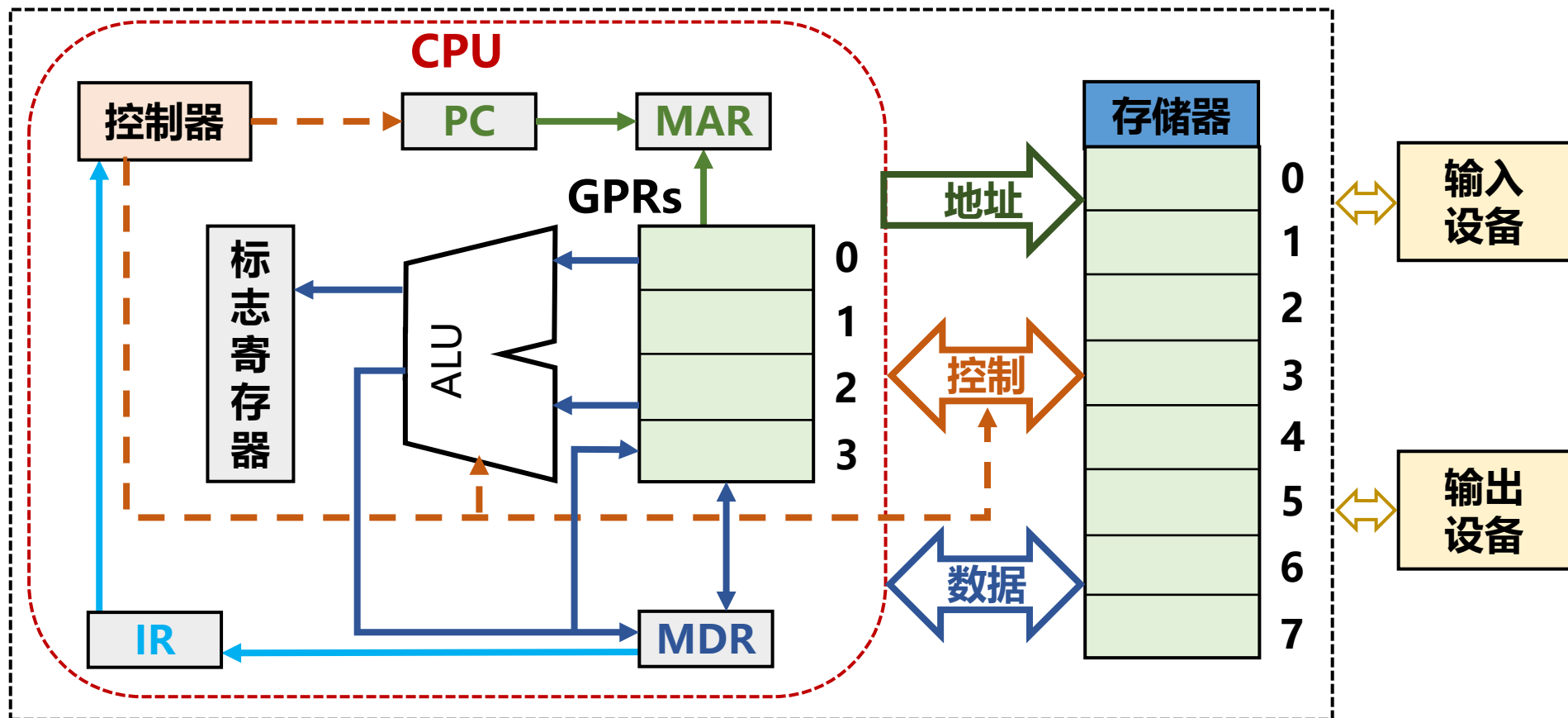
指令集体系结构 (ISA)

Instruction Set Architecture (ISA), 有时简称为**指令系统**。常与**计算机体系结构混用**。

- ISA是一种规约 (Specification) , 它规定了**如何使用硬件**
 - 可执行的指令的集合, 包括**指令格式**、**操作种类**以及每种操作对应的操作数的相应规定;
 - 指令可以接受的**操作数类型**;
 - 操作数所能存放的寄存器组的结构, 包括每个**寄存器的名称**、**编号**、**长度和用途**;
 - 操作数所能存放的**存储空间的大小和编址方式**;
 - 操作数在存储空间存放时按照**大端还是小端方式**存放;
 - 指令获取操作数的方式, 即**寻址方式**;
 - 指令执行过程的控制方式, 包括**程序计数器 (PC)**、**条件码定义**等。
- ISA在**通用计算机系统是必不可少**的一个**抽象层**



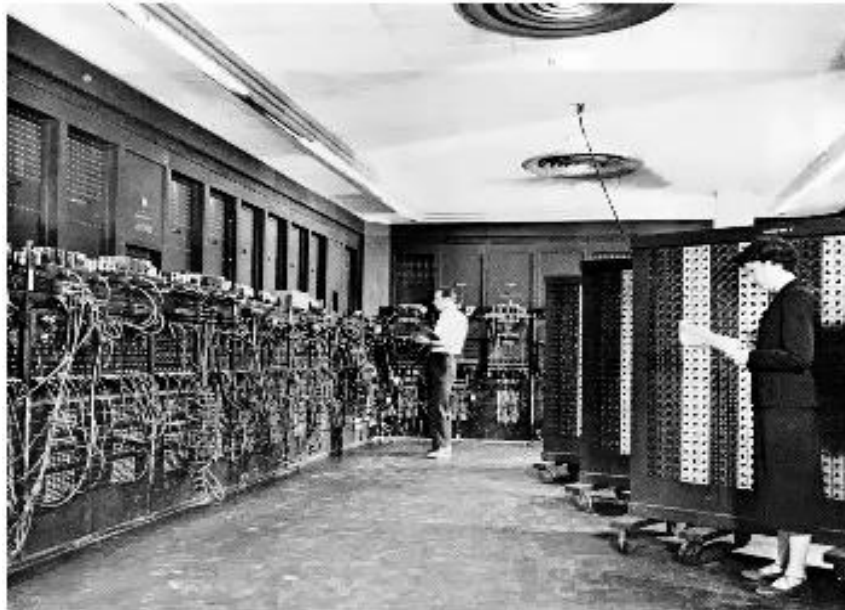
ISA与组成（微结构）之间的关系



不同ISA规定的指令集不同，如IA-32，MIPS，ARM等
计算机组成必须能够实现ISA规定的功能，如提供GPR、标志、运算电路等
同一种ISA可以有不同的计算机组成，如乘法指令可用ALU或乘法器实现

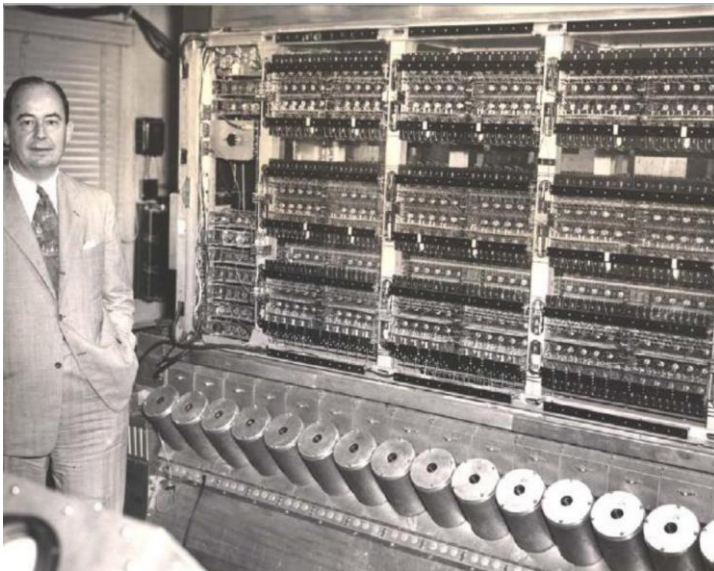
计算机简史

- 第一代：真空管（1946-1957）
 - ENIAC（1946-1955）：第一台通用计算机，十进制，手动编程
 - Electronic Numerical Integrator And Computer
 - ABC（1937）：世界上第一台电子计算机，不可编程
 - Atanasoff–Berry Computer

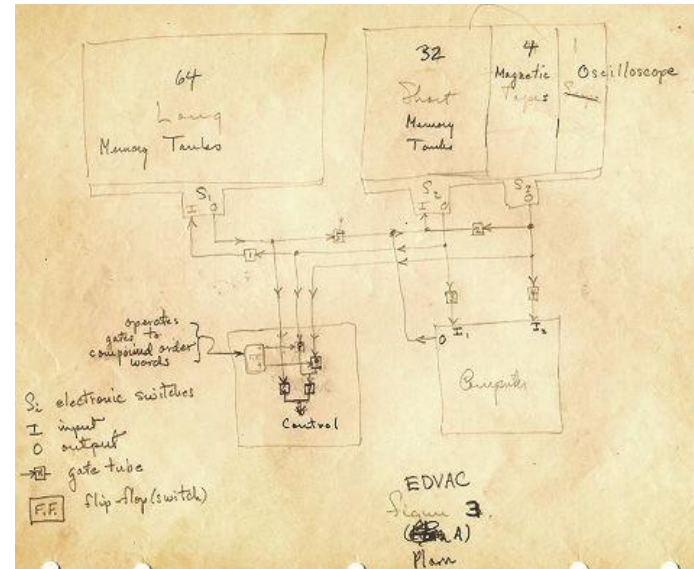


计算机简史

- 第一代：真空管（1946-1957）
 - EDVAC（1944-1951）：**冯·诺伊曼结构**
 - Electronic Discrete Variable Automatic Computer



冯·诺伊曼（von Neumann）



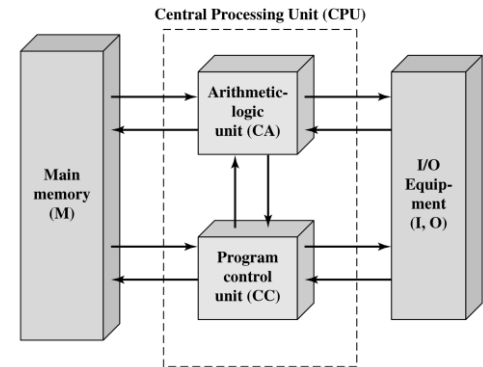
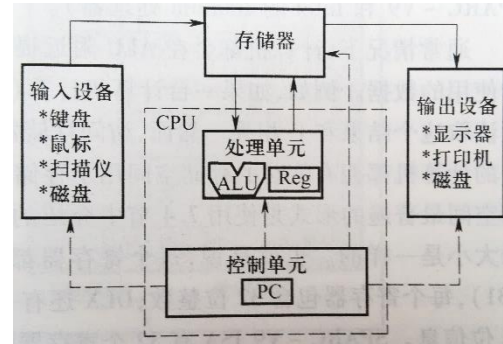
The First Draft Report on the EDVAC
von Neumann (1945)

回顾: 什么是“冯·诺伊曼结构”？



冯·诺伊曼结构

- 又称为“普林斯顿结构”
- 三个基本原则
 - 二进制
 - 存储程序
 - 5个组成部分



- 主存储器：地址和存储的内容
- 算术逻辑单元 / 处理单元：执行信息的实际处理
- 程序控制单元 / 控制单元：指挥信息的处理
- 输入设备：将信息送入计算机中
- 输出设备：将处理结果以某种形式显示在计算机外

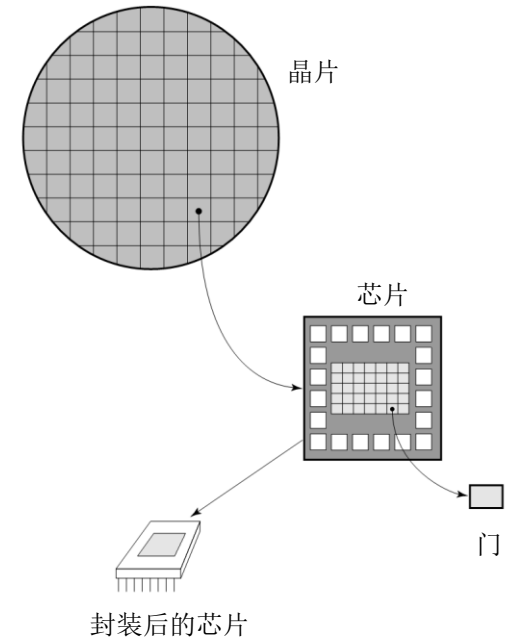
计算机简史

- 第二代：晶体管（1958-1964）
 - NCR和RCA, IBM 7000：晶体管体积更小、更便宜、发热更少，而且能以与电子管相同的方式建造计算机
 - 采用更复杂的算术逻辑单元和控制器，使用高级编程语言，并为计算机提供了系统软件

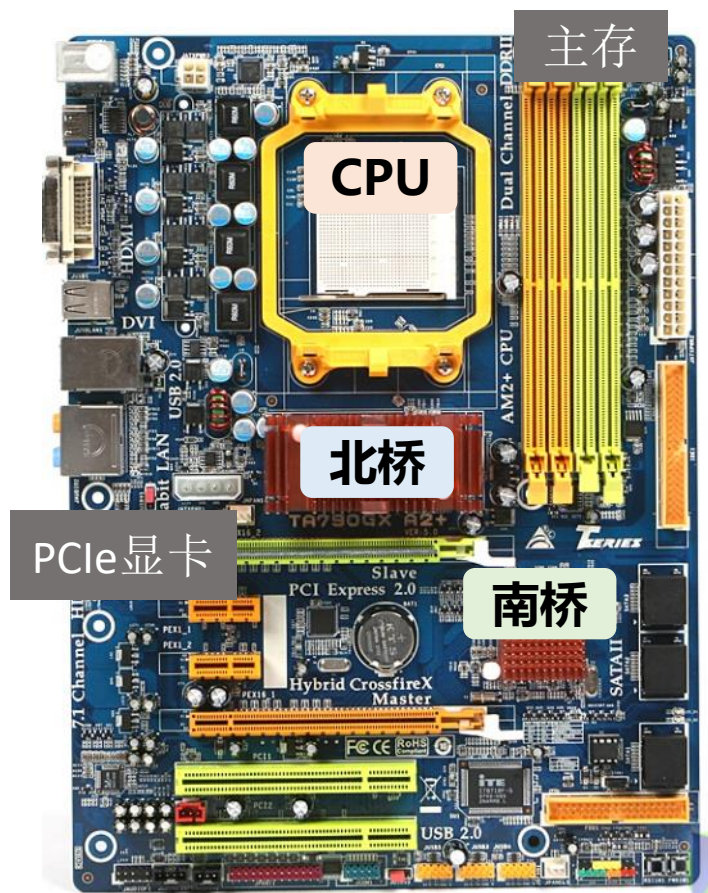


计算机简史

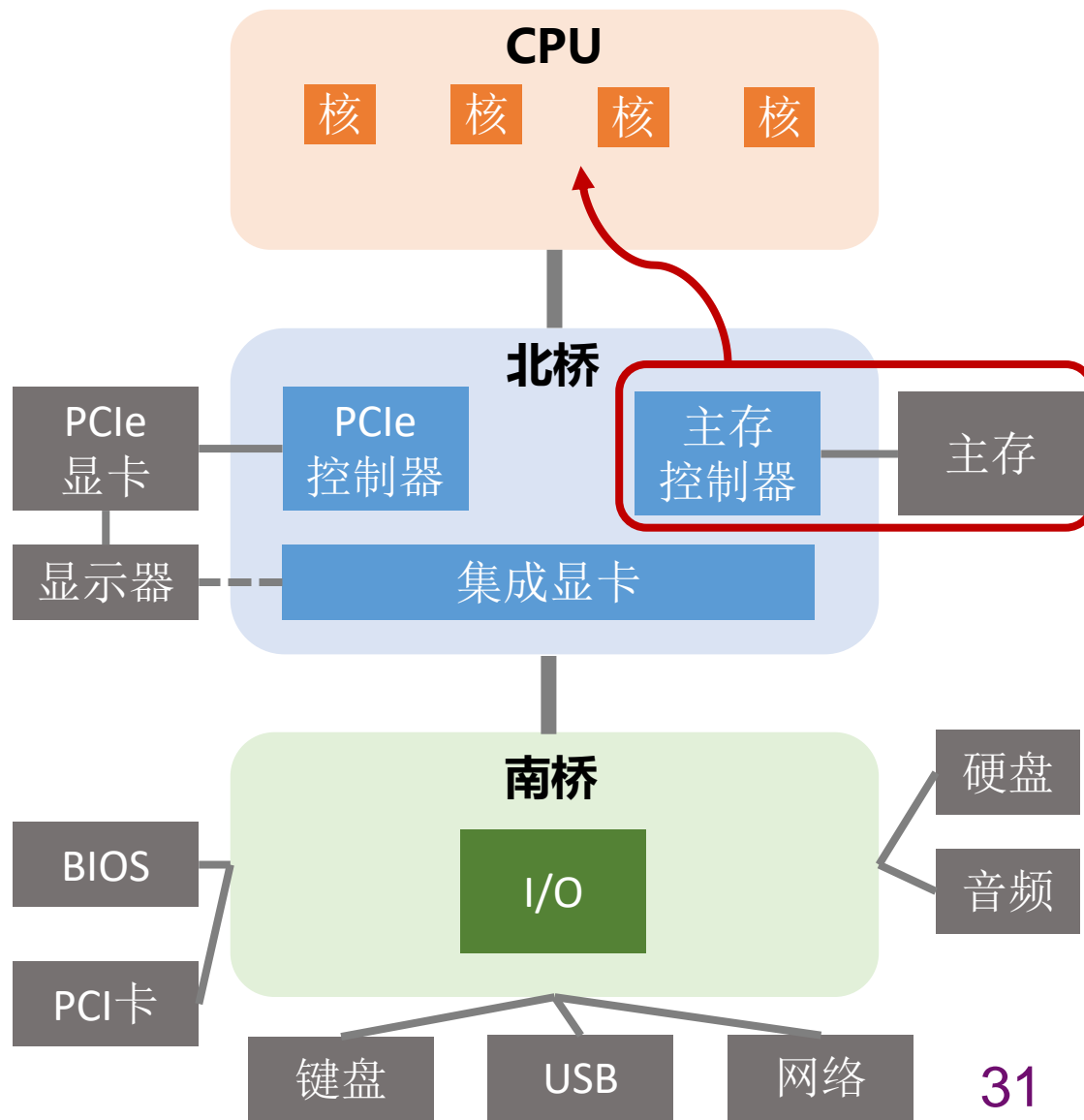
- 第三代及后续几代：集成电路（1965-现在）
 - 思想：
 - 将整个电路安装在很小的硅片上，而不是用分立元件搭成的等价电路
 - 晶体管可以通过金属化过程相互连接，以形成电路
 - 规模：
 - 小 → 大 → 超大 → 巨大 ...



冯·诺伊曼结构的实现与演变

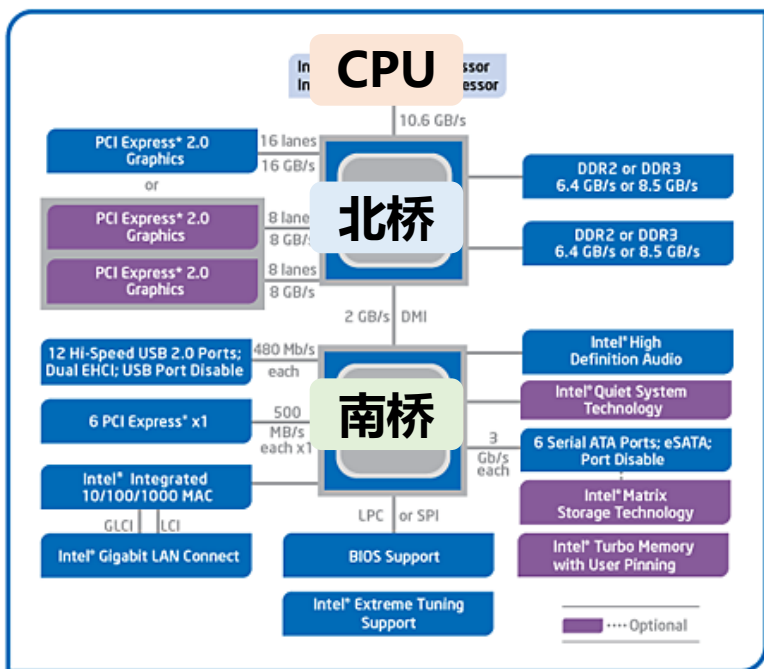


2009年停产



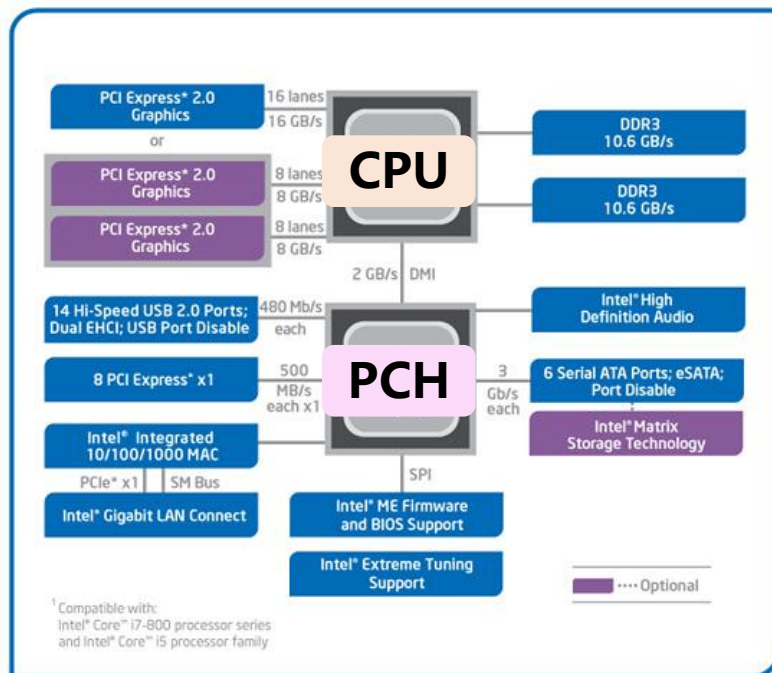
冯·诺伊曼结构的实现与演变

2008年底进入酷睿时代，南北桥变成PCH



Intel® P45 Express Chipset Block Diagram

40系列



¹ Compatible with:
Intel® Core™ i7-800 processor series
and Intel® Core™ i5 processor family

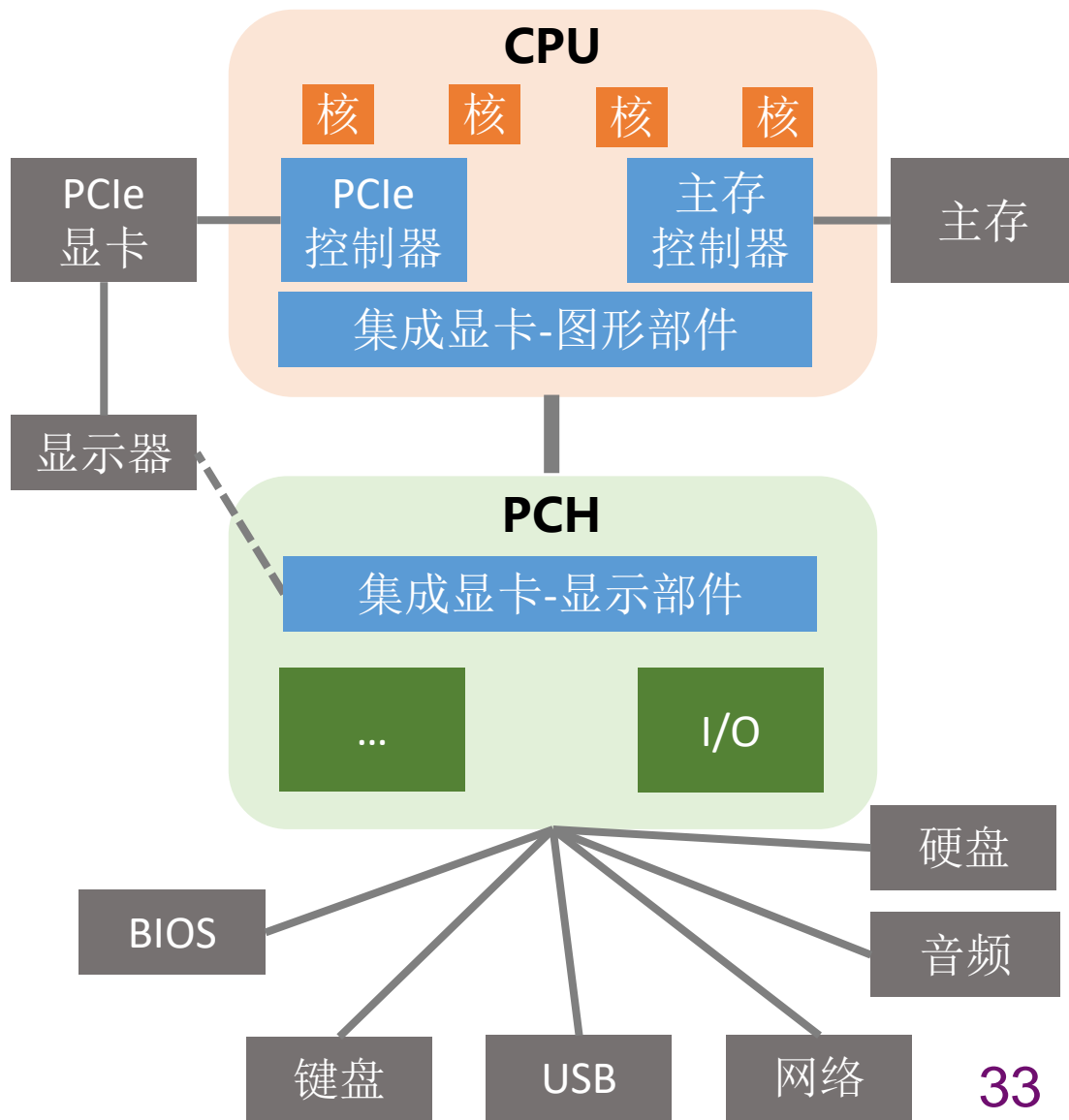
50系列



冯·诺伊曼结构的实现与演变



2022年上市



回顾: 什么是“摩尔定律”?



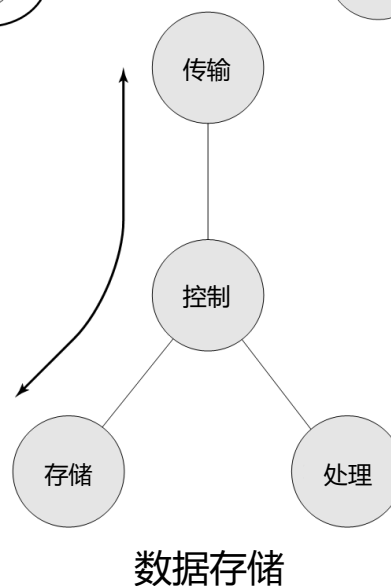
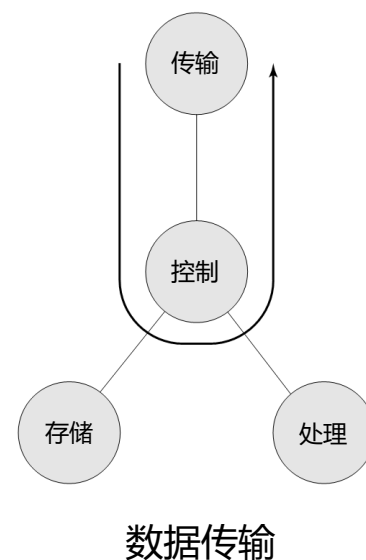
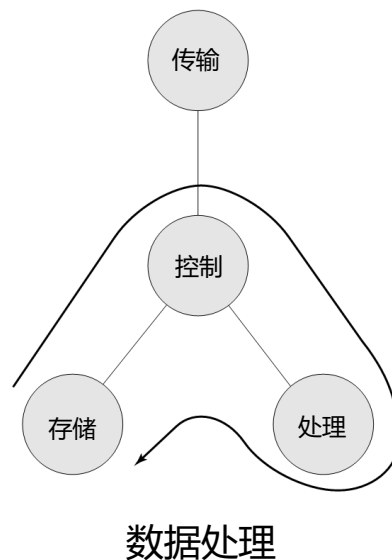
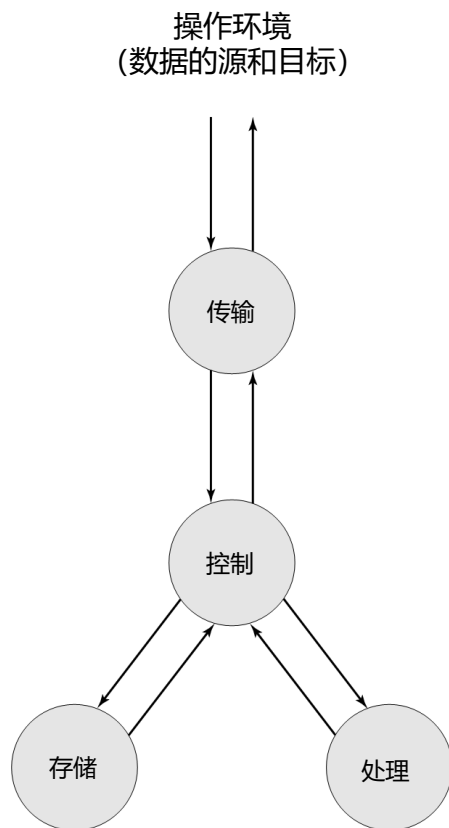
摩尔定律

- **摩尔定律** (Gordon Moore, 1965)
 - 当**价格不变**时, 单芯片上所能包含的晶体管数量每年翻一番 (1965-1969) / 1970年起减慢为每18个月翻一番
 - 影响
 - 更小的尺寸带来更多**灵活性**和**可能性**
 - 由于单个芯片的成本几乎不变, 计算机逻辑电路和存储电路的**成本显著下降**
 - 减小了对**电能消耗和冷却**的要求
 - 集成电路上的内部连接比焊接更可靠, **芯片间的连接更少**



计算机发展：变与不变

- 基本功能



计算机发展：变 与 不变

- 运算速度

发展阶段	大致时间	技术	典型速度（每秒的操作次数）
1	1946–1957	真空管（电子管）	40,000
2	1958–1964	晶体管	200,000
3	1965–1971	小规模和中规模 集成电路	1,000,000
4	1972–1977	大规模集成电路	10,000,000
5	1978–1991	超大规模集成电路	100,000,000
6	1991–	巨大规模集成电路	1,000,000,000

2022年，苹果的M1 Ultra芯片做到了**1140亿**晶体管，是52年前的**1亿倍**



计算机性能

- 计算机的关键参数之一
 - 性能，成本，尺寸，安全性，可靠性，能耗，
- 性能评价标准
 - CPU：速度
 - 存储器：速度，容量
 - I/O：速度，容量
 -

计算机设计的主要目标是：提高CPU性能



CPU性能

- 系统时钟

- **时钟频率** / 时钟速度（单位：Hz）：计算机在单位时间内（例如1秒钟）执行最基本操作的次数
- **时钟周期** / 周期时间（单位：s）：执行每次最基本操作的时间
 - 时钟滴答（有时也称为“时钟周期”）：CPU 中用于**同步**执行最基本操作的单个电子脉冲
 - 因此，周期时间即为两个电子脉冲之间的时间
 - 时钟周期是时钟频率的倒数



CPU性能

- 指令执行

- 处理器由时钟驱动，时钟具有固定的频率 f ，或等价于固定的时钟周期 t
- 如果用 CPI_i 来表示指定类型 i 所需要的周期数，用 I_i 表示在某一给定程序中所执行的 i 类指令的条数
- 则我们可以计算整个CPI如下：

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_c}, \quad I_c = \sum_{i=1}^n I_i$$

执行一个给定程序的处理时间表示为：

$$T = I_c \times CPI \times t$$

$$T = I_c \times [p + [m \times k]] \times t$$

存储器周期时间和处理器周期时间之比

译码和执行指令 存储器访问次数

在处理器和存储器之间传输数据



CPU性能

- 每秒百万条指令 (MIPS) :

$$MIPS = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

- 每秒百万条浮点操作 (MFLOPS) :

$$MFLOPS = \frac{N_{floating-point\ op}}{T \times 10^6}$$



CPU性能

- 基准程序
 - 使用一系列基准程序来测量系统的性能
 - 平均结果：
 - 算数平均值: $R_A = \frac{1}{m} \sum_{i=1}^m R_i$
 - 调和平均值: $R_H = \frac{m}{\sum_{i=1}^m \frac{1}{R_i}}$



性能设计的基本原则

大概率事件优先原则

- 对于大概率事件（或最常见的事件），赋予它优先的处理权和资源使用权

$$\text{系统加速比} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$

Amdahl定律

- 加快某部件执行速度所获得的系统性能加速比，受限于该部件在系统中所占的重要性比例
- 性能增加的递减规则：**如果仅仅对计算机中的一部分做性能改进，改进越多，系统获得的效果越小。

$$\text{系统总加速比} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}} = \frac{1}{(1 - \text{局部占比}) + \frac{\text{局部占比}}{\text{提升的性能}}}$$



性能设计的基本原则

只提高CPU性能够吗？

怎么提高计算机性能？



总结

- 概念
 - 计算机，组织，结构
- 计算机发展历史
 - 真空管 → 晶体管 → 集成电路
 - 冯·诺伊曼结构，摩尔定律，
- 计算机发展
 - 基本功能，运算速度
- 计算机性能
 - CPU性能评价：时钟频率，CPI，MIPS，MFLOPS，基准程序
 - 性能设计的基本原则：大概率事件优先原则 + Amdahl定律



谢谢

bohanliu@nju.edu.cn

