



中国科学技术大学  
University of Science and Technology of China

# 计算机体系结构

周学海

[xhzhou@ustc.edu.cn](mailto:xhzhou@ustc.edu.cn)

0551-63606864

中国科学技术大学



# Welcome to .....

- 主讲: 周学海(xhzhou@ustc.edu.cn)
- 办公地点: 高能效智能计算实验室
- 办公电话: 0551-63606864



面向嵌入式应用

服务于智慧城市与企业级智能应用的智能服务器

可提供公有云服务的智能计算机集群



## 高能效智能计算平台

- 功耗、性能、小型化
- 时间可预测性问题
- 软硬件协同设计

- 混合异构体系结构设计
- 单节点系统的资源调度
- 智能处理板卡设计
- 软硬件协同设计

- 节点互联技术
- 存储系统设计
- 分布式系统资源调度



# Welcome to .....

- **课程主页:**

- bb.ustc.edu.cn

- <http://staff.ustc.edu.cn/~xhzhou/CA-Spring2020/CA.html>

- <http://home.ustc.edu.cn/~hyf15/>

- **QQ群号: 1029041637**



# 助教

姓名	电子邮件
黄一凡	hyf15@mail.ustc.edu.cn
王昱昊	ustcwyh@mail.ustc.edu.cn
刘奕品	ypliu88@mail.ustc.edu.cn
吴豫章	752865034@qq.com
任志才	renzcai@mail.ustc.edu.cn
唐昱尧	ty909@mail.ustc.edu.cn

本课程的先修课程为：数字逻辑、计算机组成原理。关于先修课程，请选择：

- A 已修过这两门课程
- B 已修过数字逻辑，但未修过计算机组成原理
- C 已修过计算机组成原理，但未修过数字逻辑
- D 数字逻辑和计算机组成原理均未修过

提交



# Chapter1 量化设计与分析基础

- **1.1 引言**
  - 计算机体系结构的定义
- **1.2 体系结构发展历史、现状及趋势**
  - 现代计算机系统发展趋势
- **1.3 定量分析基础**



---

# 1.1 引言

---

什么是计算机体系结构

为什么要学习计算机体系结构

本课程的基本要求





# What is Computer Architecture?

Application



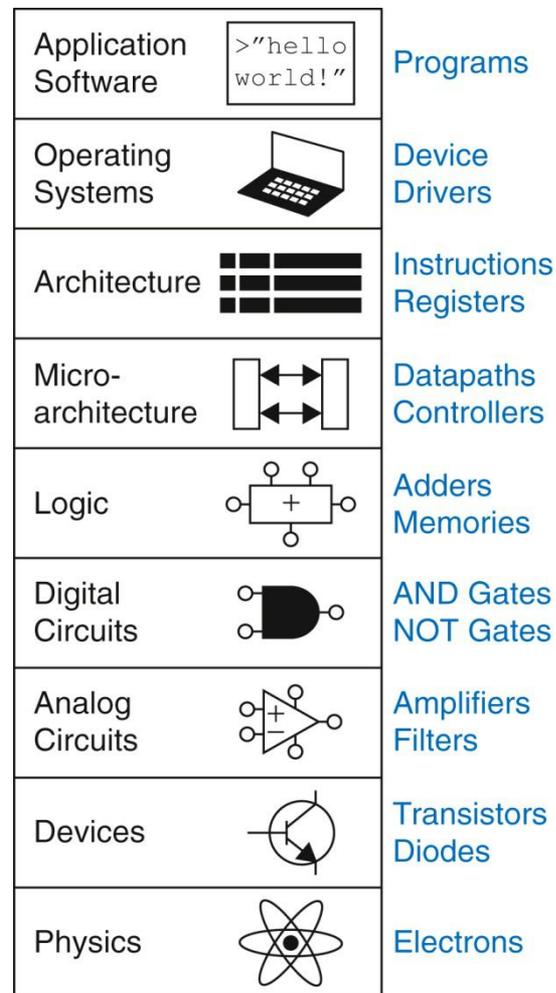
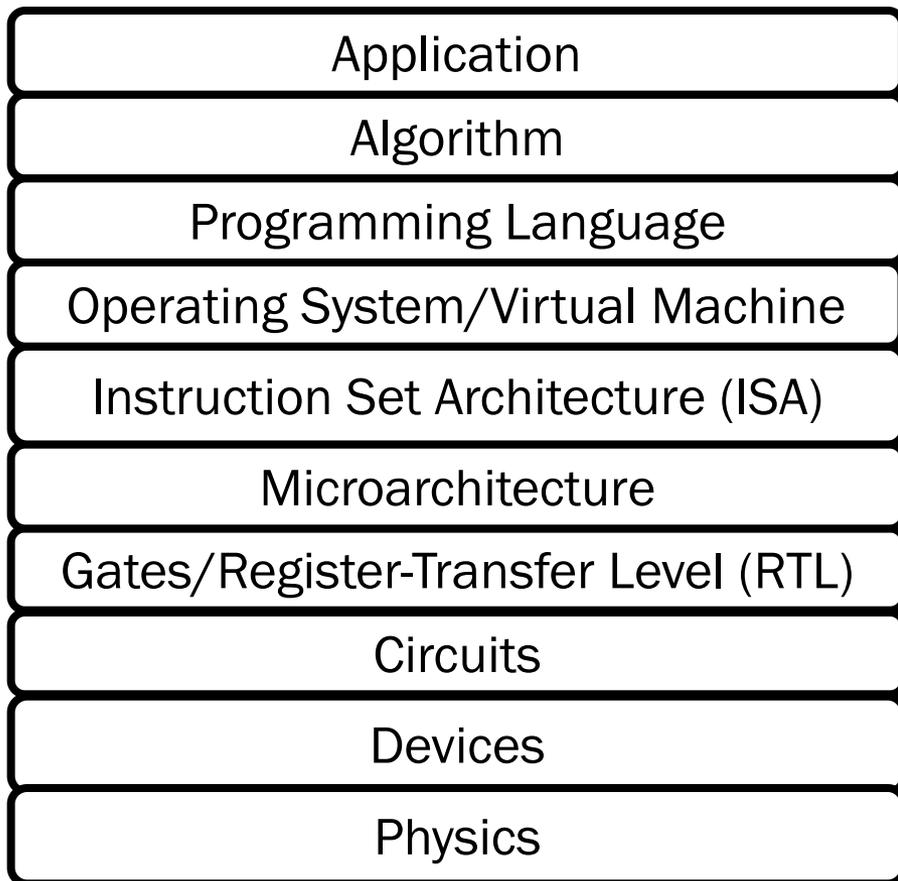
Gap too large to bridge  
in one step  
*(but there are exceptions, e.g.  
magnetic compass)*

Physics

**广义的定义：**计算机体系结构指计算机（硬件）系统的抽象表示，基于这些抽象表示使得我们可以**更好地**使用可用的制造技术**实现（信息处理）硬件系统，高效地设计与实现（信息处理）软件系统**



# 现代计算机系统的抽象层次



Copyright © 2016 Elsevier Ltd. All rights reserved.



# What is Computer Architecture?

- **计算机体系结构是研究如何选择（设计）功能部件和互连方法来满足计算机系统的功能、性能、价格约束的科学**
- **描述计算机系统的功能、结构组织和实现的一组规则和方法。**
- **计算机体系结构是软件设计者与硬件设备设计者（VLSI）之间的中间层，是软件与硬件的接口（Interface）**
- **计算机体系结构定义为：一组指令及机器的一系列状态**



# 计算机体系结构的定义

## 过去的定义：

程序员可见的计算系统的属性。包括：概念性的结构和功能行为。不包括：数据流和控制流的组织、逻辑设计以及物理实现。

– Amdahl, Blaauw, and Brooks, 1964

**Instruction-Set Architecture**

## 广义的定义：

计算机体系结构研究如何 **设计、组织，以及使用可用的生产技术实现信息处理（硬件）系统**，该系统可有效地执行软件应用，并满足价格、功耗和性能约束



# 计算机体系结构的定义 (续)

- **过去的观点:**

- 指令集架构 (ISA) 设计

- 即 体系结构设计需要关注并确定:

- 寄存器组织、存储模型、寻址方式、指令操作数、硬件支持的操作种类、指令编码方式
- 如何处理中断和异常?

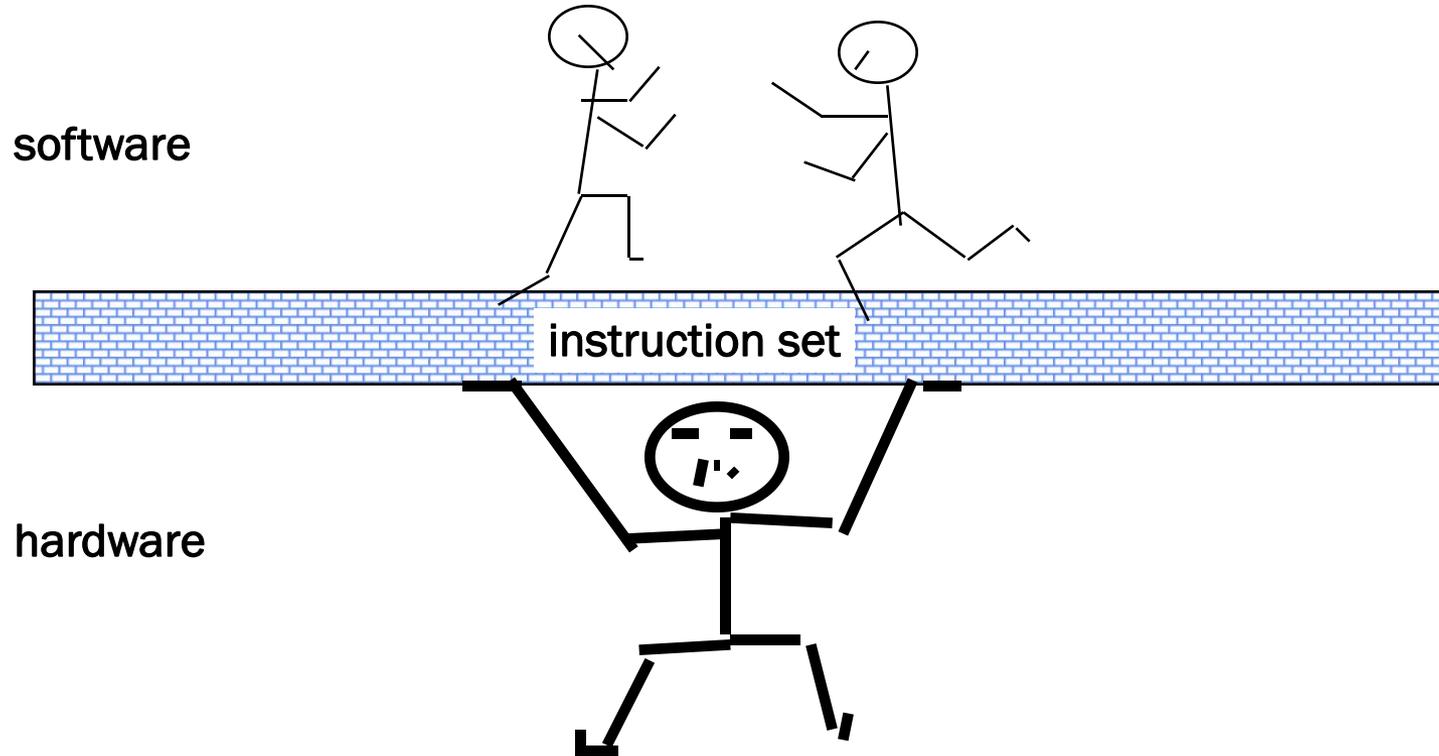
- **目前的观点:**

- 根据目标机器的特定需求, 在成本、功耗、可用性等约束下最大化机器性能

- 包括 指令集架构 (ISA), 计算机组织 (微体系结构), 硬件实现



# ISA: a Critical Interface





# ISA需说明的主要内容

- **Memory addressing**
- **Addressing modes**
- **Types and sizes of operands**
- **Operations**
- **Control flow instructions**
- **Encoding an ISA**
- **.....**
- **优秀的ISA所具有的特征**
  - 可持续用于很多代机器上(**portability**)
  - 可以适用于多个领域 (**generality**)
  - 对上层提供方便的功能 (**convenient functionality**)
  - 可以由下层有效地实现 (**efficient implementation**)



# 指令集结构举例

- **Digital Alpha(v1, v3) 1992-97**
- **HP PA-RISC (v1.1, v2.0) 1986-96**
- **Sun Sparc(v8, v9) 1987-95**
- **SGI MIPS (MIPS I, II, III, IV, V) 1986-96**
- **Intel(8086,80286,80386, 1978-96  
80486,Pentium, MMX, ...)**



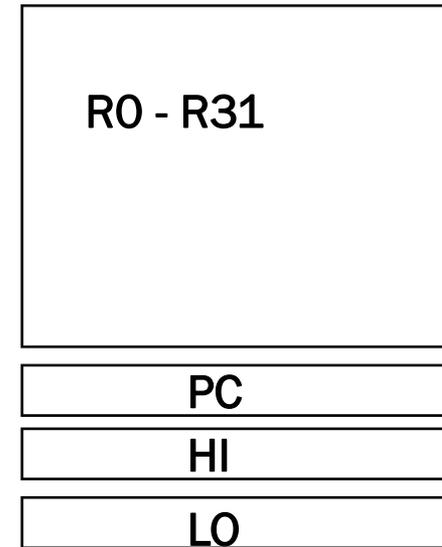
## • 指令类型

- Load/Store
- Computational
- Jump and Branch
- Floating Point
  - coprocessor
- Memory Management
- Special

3 种指令格式: all 32 bits wide

<b>R型</b>	OP	rs	rt	rd	sa	funct
<b>I型</b>	OP	rs	rt	immediate		
<b>J型</b>	OP	jump target				

Registers





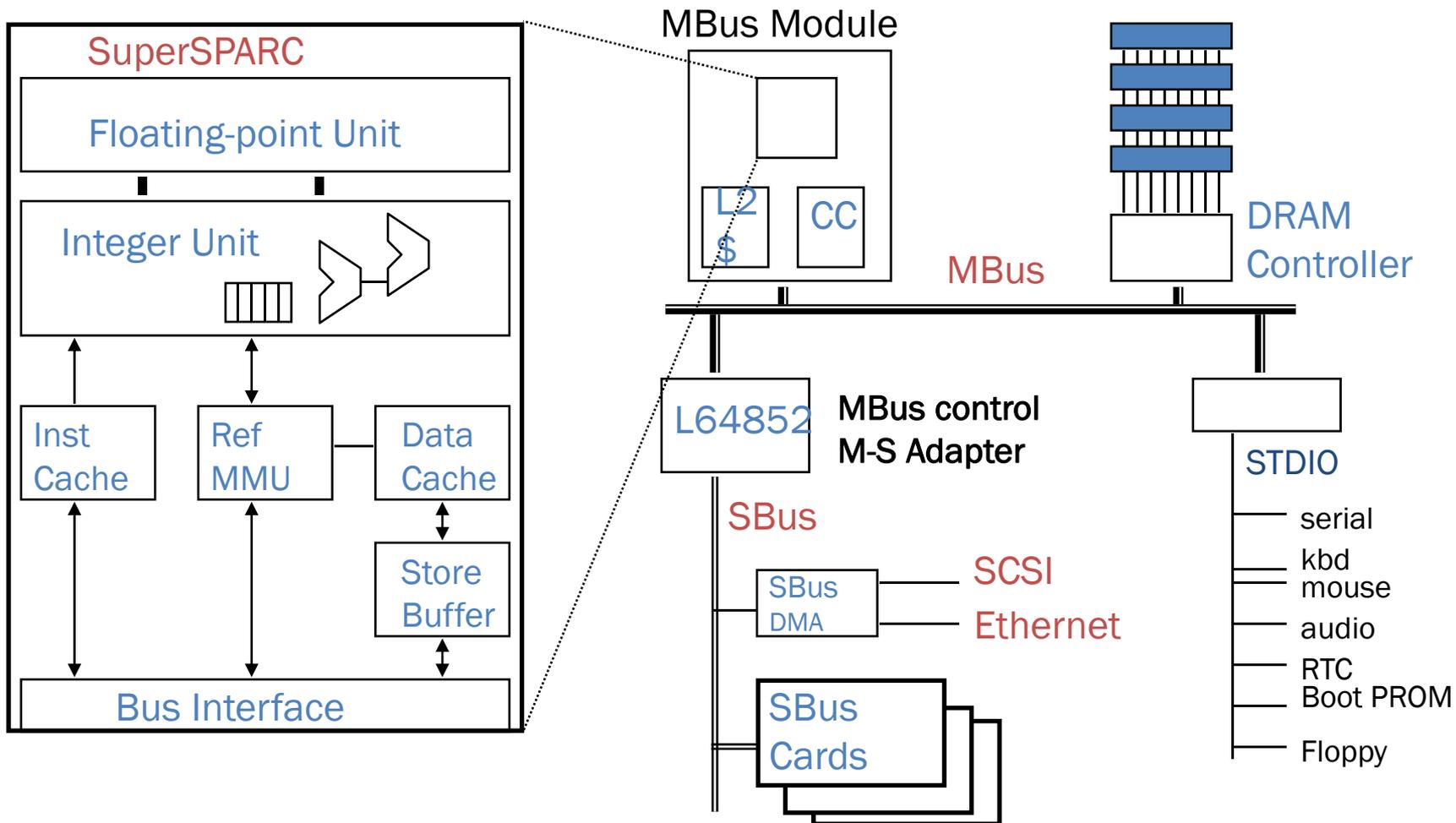
# 计算机组成与实现

- **计算机组成 (Computer Organization or Microarchitecture): ISA的逻辑实现**
  - 物理机器级中的数据流和控制流的组成以及逻辑设计等
- **计算机实现 (Computer Implementation): 计算机组成的物理实现**
  - CPU, MEMORY等的物理结构, 器件的集成度、速度, 模块、插件、底板的划分与连接、信号传输、电源、冷却及整机装配技术等
- **例如**
  - 确定指令系统中是否有乘法指令 (Architecture)
  - 确定用加法器实现乘法 还是用专门的乘法器实现 (Organization)
  - 器件的选定及所用的微组装技术 (Implementation)



# Example Organization

- TI SuperSPARC™ TMS390Z50 in Sun SPARCstation20





# 指令集架构 vs. 微体系结构

- **Architecture / Instruction Set Architecture (ISA)**
  - Class of ISA: register-memory or register-register architectures
  - Programmer visible state (Register and Memory)
  - Addressing Modes: how memory addresses are computed
  - Data types and sizes for integer and floating-point operands
  - Instructions, encoding, and operation
  - Exception and Interrupt semantics
- **Microarchitecture / Organization**
  - Tradeoffs on how to implement the ISA for speed, energy, cost
  - Pipeline width and depth, cache size, peak power, bus width, execution order, etc

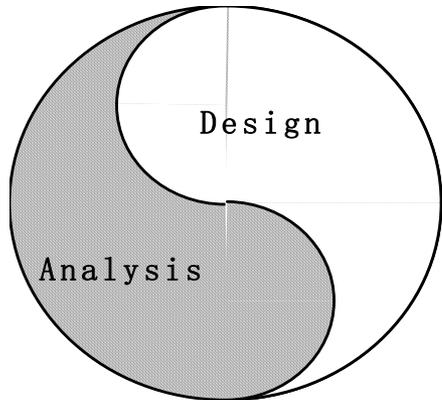


# 计算机体系结构设计者的任务

- **设计和实现不同档次的计算机系统**
  - Understand software demands
  - Understand technology trends
  - Understand architecture trends
  - Understand economics of computer systems
- **最大化性能、可编程性等指标**
  - 在一定的技术和成本的限制下
- **体系结构现状:**
  - 现代微处理器大多为多核处理器
  - 单芯片中通常集成多个处理器核心
  - 每个处理器核心支持多线程执行



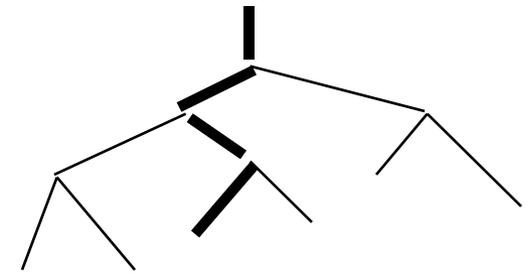
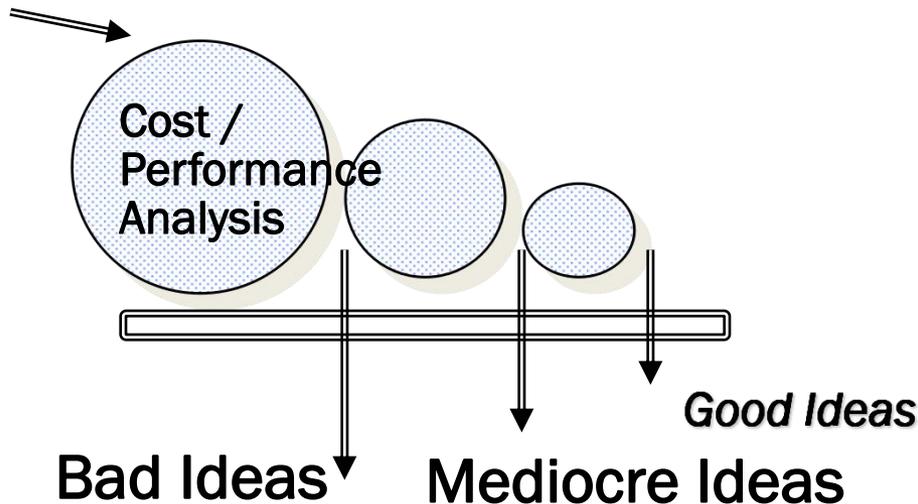
# 计算机体系结构设计过程



**体系结构设计是循环渐进的过程:**

- Search the possible design space
- Make selections
- Evaluate the selections made

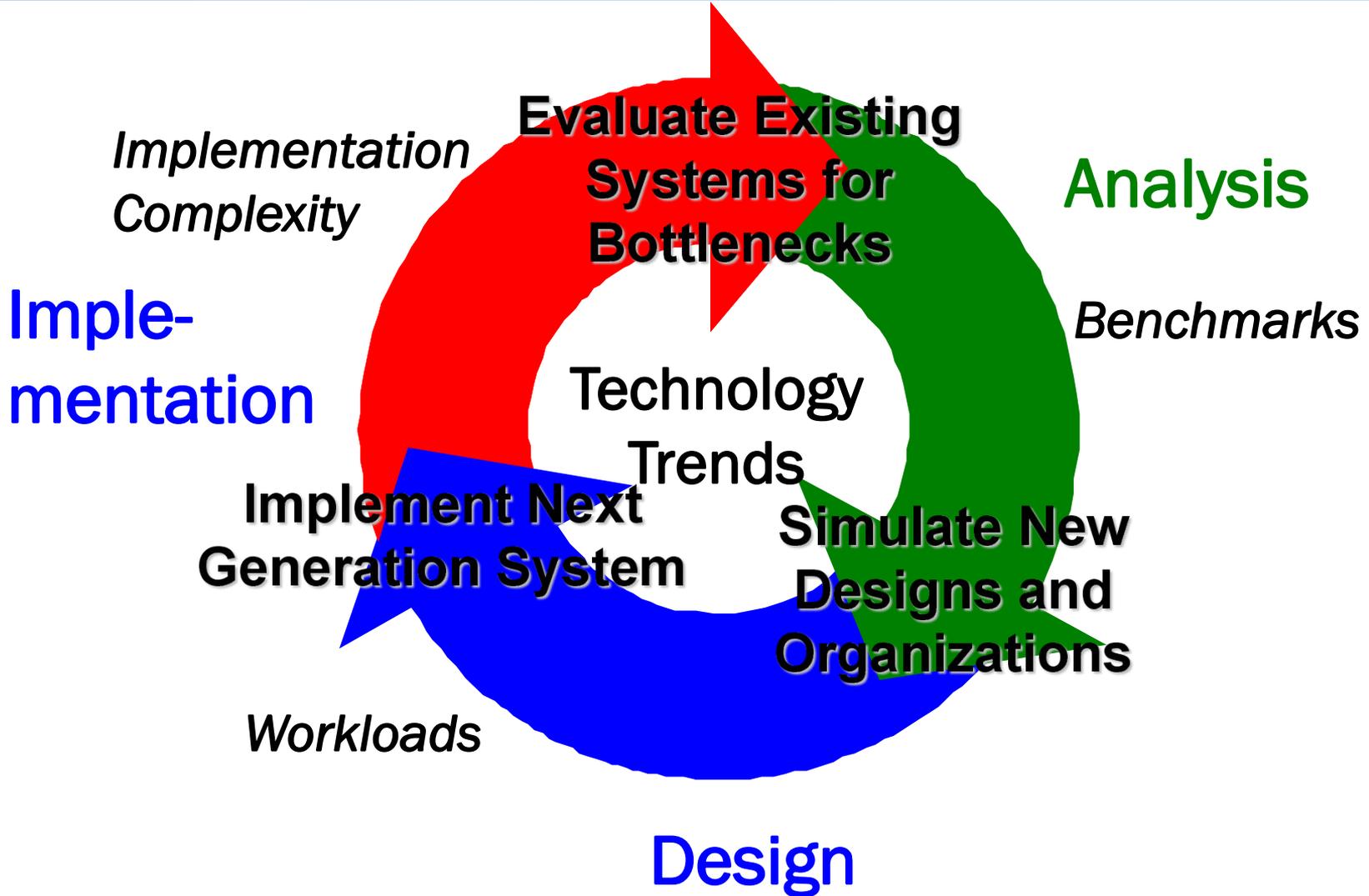
**Creativity**



**Good measurement tools are required to accurately evaluate the selection.**



# 计算机工程方法学





---

# 1.1 引言

---

什么是计算机体系结构

为什么要学习计算机体系结构

本课程的基本要求





# 为什么要学计算机体系结构

## 深入理解计算机体系结构:

- **开展体系结构研究与设计的基础**

- 体系结构领域仍然存在许多挑战性问题
  - 例如: CPU与memory之间性能差异
  - 功耗和能耗问题
  - 体系结构与应用特征的适配性问题等

– .....

- **更好地设计与实现操作系统、编译器**

- 需要重新评估当前的假设和权衡
  - 例如: 当面临网络性能持续提升、并行系统、异构系统日益普遍
- 现代计算机需要更好的优化编译器和更好的编程语言

- **更好地设计与实现应用程序**

- 可更好地理解算法、数据结构和编程语言选择对性能的影响



# 1.1 引言

什么是计算机体系结构

为什么要学习计算机体系结构

本课程的基本要求



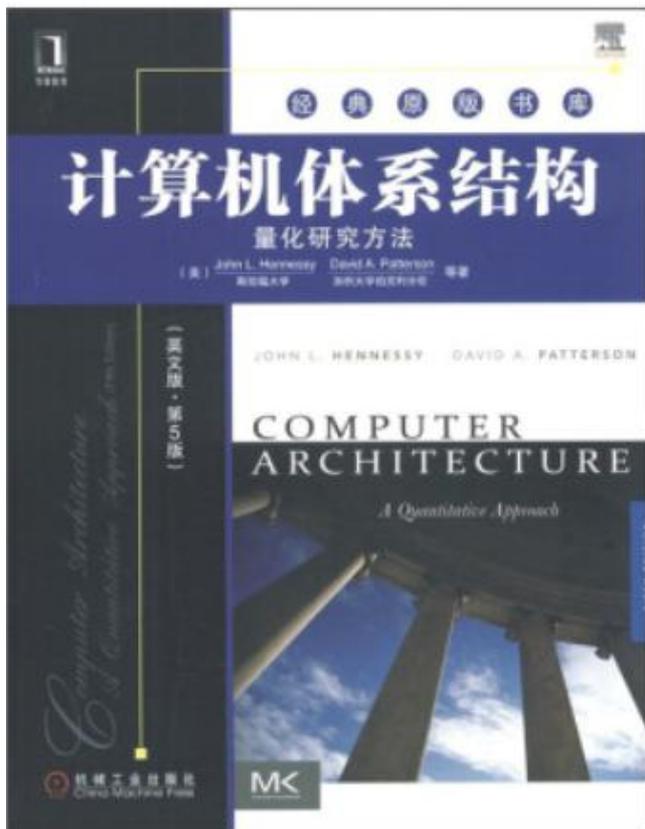


# 课程目标

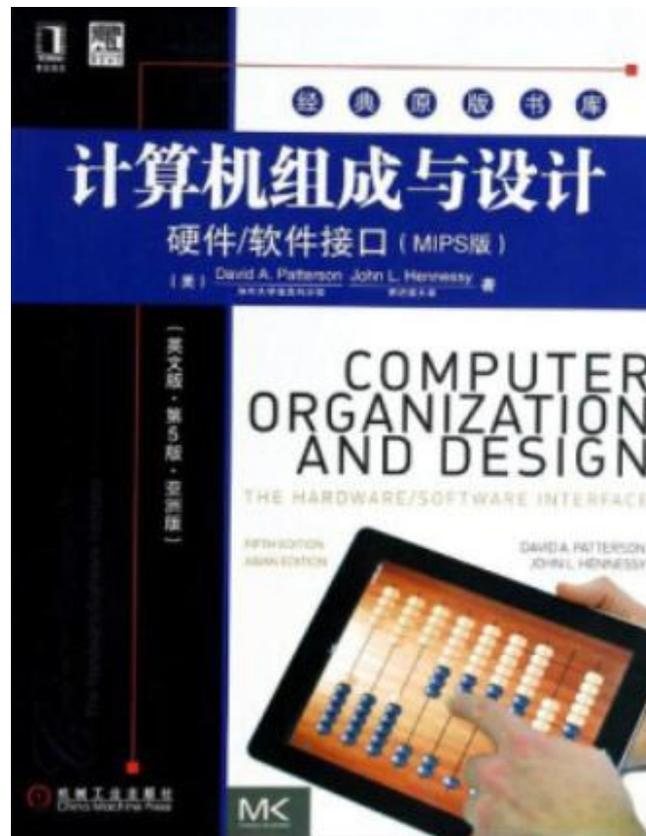
- **掌握**系统定量分析的基本方法和技术
- **深入理解**提高CPU性能的基本方法
- **深入理解**存储系统的基本原理和优化方法
- **理解**数据级并行、指令级并行、线程级并行的基本原理和方法
- **初步理解**面向特定领域的处理器设计



# 教材与主要参考书



**John L. Hennessy, David A. Patterson; Computer Architecture: A Quantitative Approach. Fifth Edition. 机械工业出版社, 2012**



**David A. Patterson, John L. Hennessy, Computer Organization & Design : The Hardware/Software Interface, Third Edition. San Francisco: Morgan Kaufmann Publishers, Inc. 2005**



# 教材与主要参考书

Sixth Edition

John L. Hennessy | David A. Patterson  
**COMPUTER  
ARCHITECTURE**



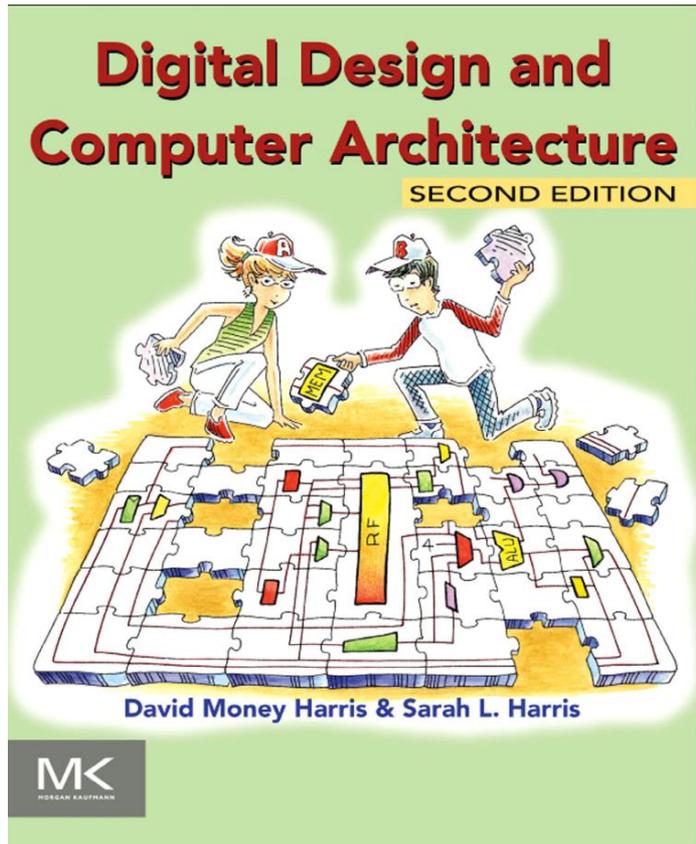
John L. Hennessy, David A. Patterson;  
Computer Architecture: A Quantitative  
Approach; sixth Edition.



David A. Patterson, John L. Hennessy ;  
Computer Organization and Design- The  
Hardware/Software Interface; RISC-V Edition.



# 教材与主要参考书



David Money Harris & Sarah L. Harris;  
Digital Design and Computer Architecture;  
Second Edition; Morgan Kaufmann  
Publishers, Inc. 2013

Sarah L. Harris & David Money Harris ;  
Digital Design and Computer Architecture;  
ARM Edition; Morgan Kaufmann  
Publishers, Inc. 2016



# 教材包含的主要内容





# 本课程的主要内容

- **简单机器设计 (Chapter 1, Appendix A, Appendix C)**
  - ISAs, Iron Law, simple pipelines
- **存储系统(Chapter 2, Appendix B)**
  - DRAM, caches, virtual memory systems
- **指令级并行(Chapter 3)**
  - score-boarding, out-of-order issue
  - SuperScalar, VLIW machines, multithreaded machines
- **数据级并行(Chapter 4)**
  - vector machines, SIMD, SIMT (GPU)
- **线程级并行(Chapter 5)**
  - memory models, cache coherence, synchronization
- **面向特定领域的处理器体系结构 (DSA)**
  - IPU、DSP、GPU



# 评分规则

- **授课(011135.01)**

- 授课总学时60学时, 实验30学时
- 3C301: 1(3,4), 5(6,7)

- **评分**

- 平时作业      10%
- 实验            40%
- 随堂测验      15%
- 期终考试      35%



# 关于作弊

- **作业**
- **实验**
- **考试 (测验)**



# 小结

- **过去的观点:**

- Instruction Set Architecture (ISA) 设计
- 即 体系结构设计需要关注并确定:
  - 寄存器组织、如何访问存储器、寻址方式、指令操作数、硬件支持的操作种类、指令编码方式、中断和异常的语义等

- **目前的观点:**

- 根据目标机器的特定需求, 在成本、功耗、可用性等约束下最大化机器性能
- 包括 ISA, 计算机组织 (微体系结构), 硬件实现



- 1、简要描述你对计算机体系结构这一概念的理解。
- 2、请根据你的理解阐述计算机体系结构和计算机组成原理这两门课程主要的差异在哪里？

正常使用主观题需2.0以上版本雨课堂



# Acknowledgements

- **These slides contain material developed and copyright by:**
  - John Kubiawicz (UCB)
  - Krste Asanovic (UCB)
  - John Hennessy (Stanford) and David Patterson (UCB)
  - Chenxi Zhang (Tongji)
  - Muhamed Mudawar (KFUPM)
- **UCB material derived from course CS152, CS252, CS61C**
- **KFUPM material derived from course COE501, COE502**