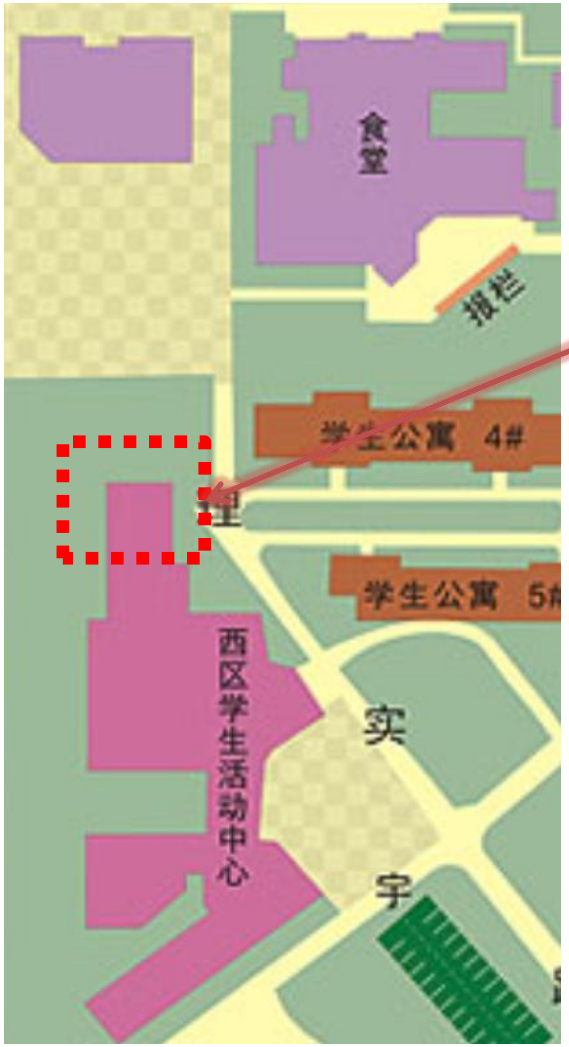
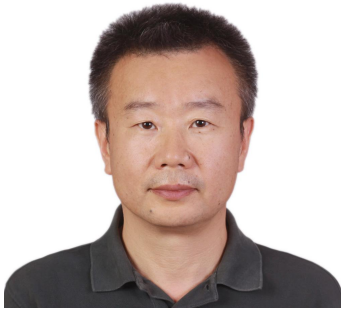
计算机体系结构

周学海
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-Spring2021/CA.html
  - http://home.ustc.edu.cn/~wyz0309/
- **QQ群号：341093440**

助教

姓名	电子邮件
吴豫章	wyz0309@mail.ustc.edu.cn
钱佳明	qj0387@mail.ustc.edu.cn
李浩然	lhr18@mail.ustc.edu.cn
庞继泽	fenggang@mail.ustc.edu.cn
高银康	gaoyinkang@mail.ustc.edu.cn

投票 最多可选1项

设置

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 课程简介**
  - 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 课程简介

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

Definition of “System”

ISO/IEC/IEEE 15288:

是人为创造的、用于在所定义的环境中提供产品或服务，以造福于用户和其他利益相关者。由相互作用的要素组合而成，以达到一个或多个目标。

The International Council on Systems Engineering (INCOSE)：

完成所定义目标的要素、子系统或组件的集成集合，这些要素包括产品、过程、人员、信息、技术、设施、服务和其他支持要素。

NASA：两种定义

- ✓ “若干功能要素组合在一起，以产生满足需求的能力。要素包括为达到目标所需的所有硬件，软件，设备，设施，人员，流程和程序。
- ✓ 完成操作功能的**最终产品**，以及为操作最终产品提供生命周期支持服务的**使能产品**。

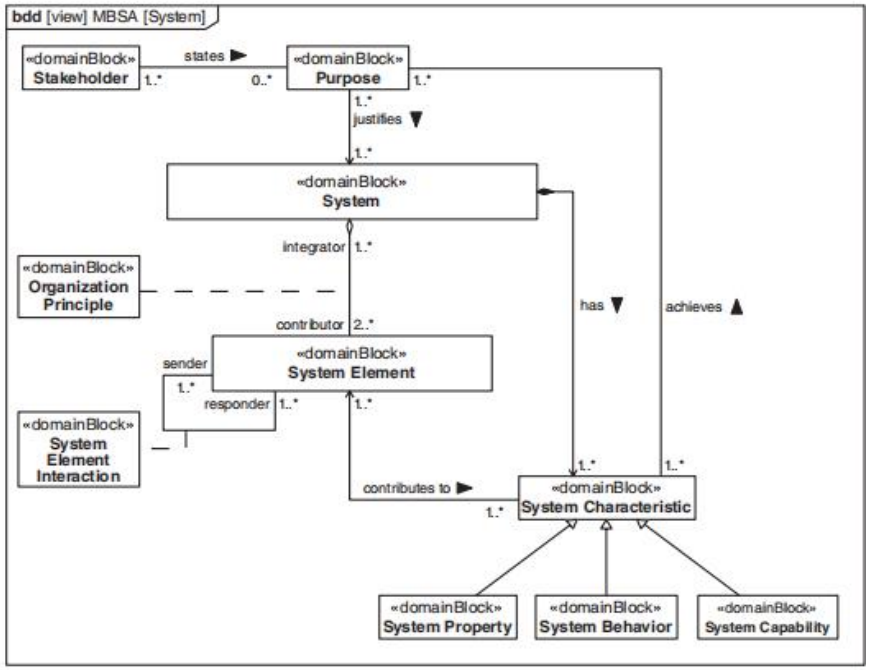


Figure 4.1. Definition of “System”.

一种简单的定义：

系统用来完成输入转变为输出的过程

In Summary: **一个系统是完成特定目标的整体，它由相互作用的部分组成。**

System Architecture

- **系统是为实现一个或多个所声明的目标而组织的相互关联的要素的组合。**
- **系统架构是系统组件的组织结构、它们之间的关系、与环境的关系、以及指导其设计和进化的原则。**
- **系统与系统架构之间的关系**
  - 每个系统都有一个系统架构，每个系统架构属于一个系统。
  - 系统架构包括关于系统组织、设计和系统演化的一些原则。

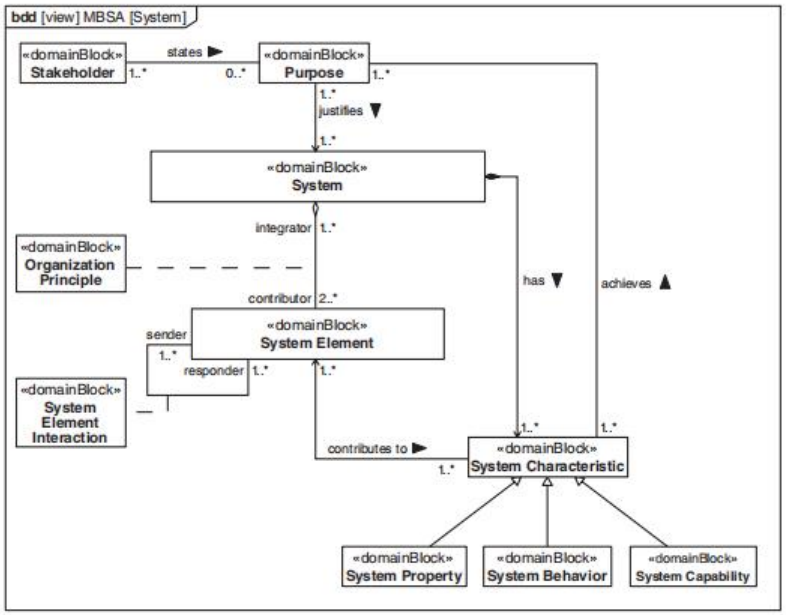


Figure 4.1. Definition of "System".

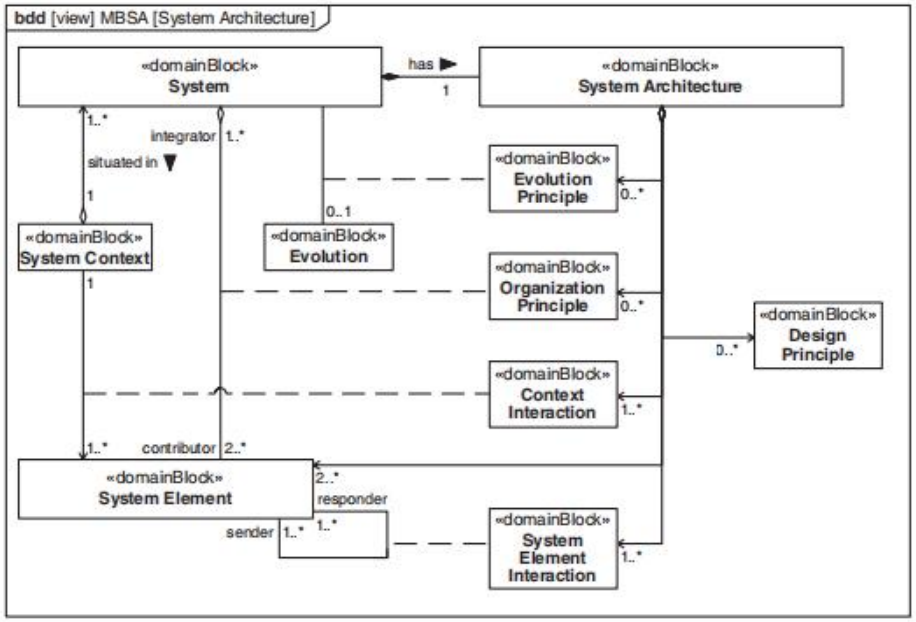
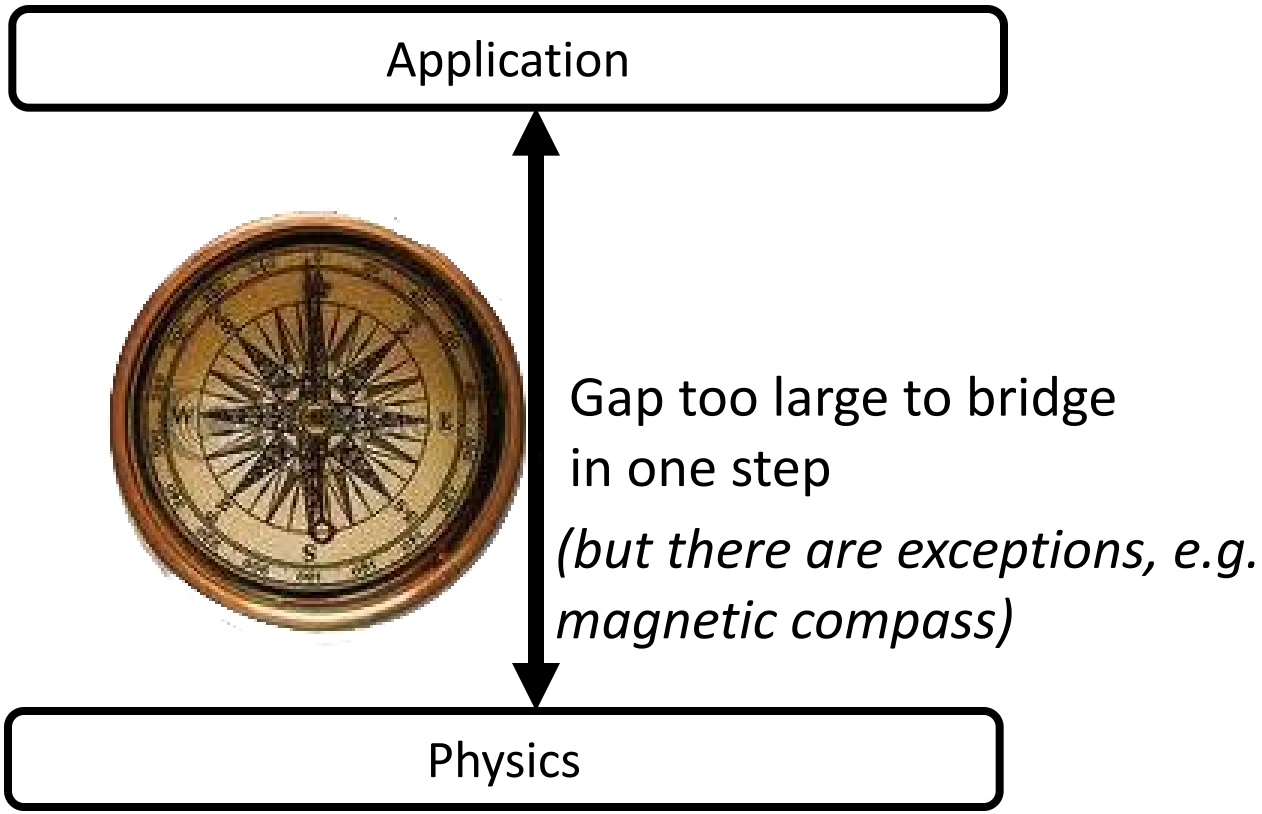


Figure 4.2. Definition of "System Architecture."

Weilkiens T , Lamm J G , Roth S , et al. Model-Based System Architecture[M]. John Wiley & Sons, Inc, 2015.

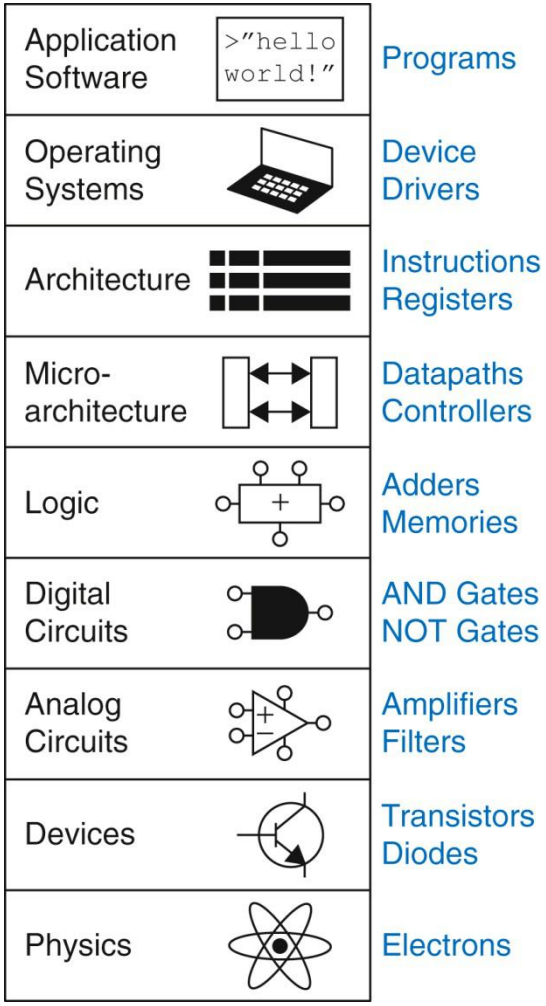
What is Computer Architecture?



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**

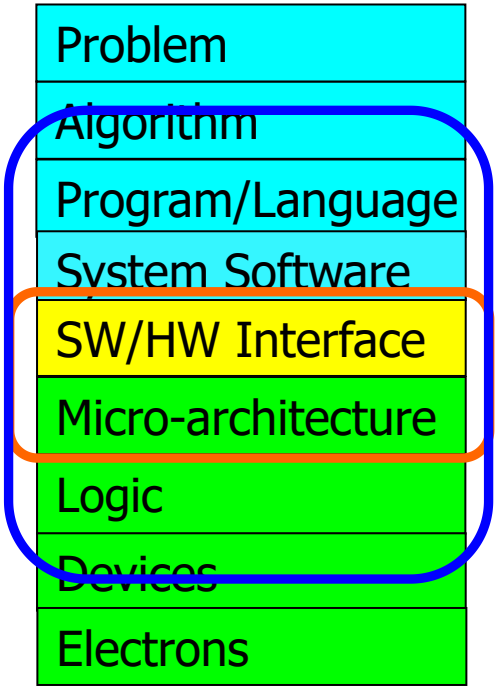
现代计算机系统的抽象层次

- Application
- Algorithm
- Programming Language
- Operating System/Virtual Machine
- Instruction Set Architecture (ISA)
- Microarchitecture
- Gates/Register-Transfer Level (RTL)
- Circuits
- Devices
- Physics



The Transformation Hierarchy

Computer Architecture (expanded view)



Computer Architecture (narrow view)

计算机体系结构研究范畴

早期的定义：Instruction-Set Architecture

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

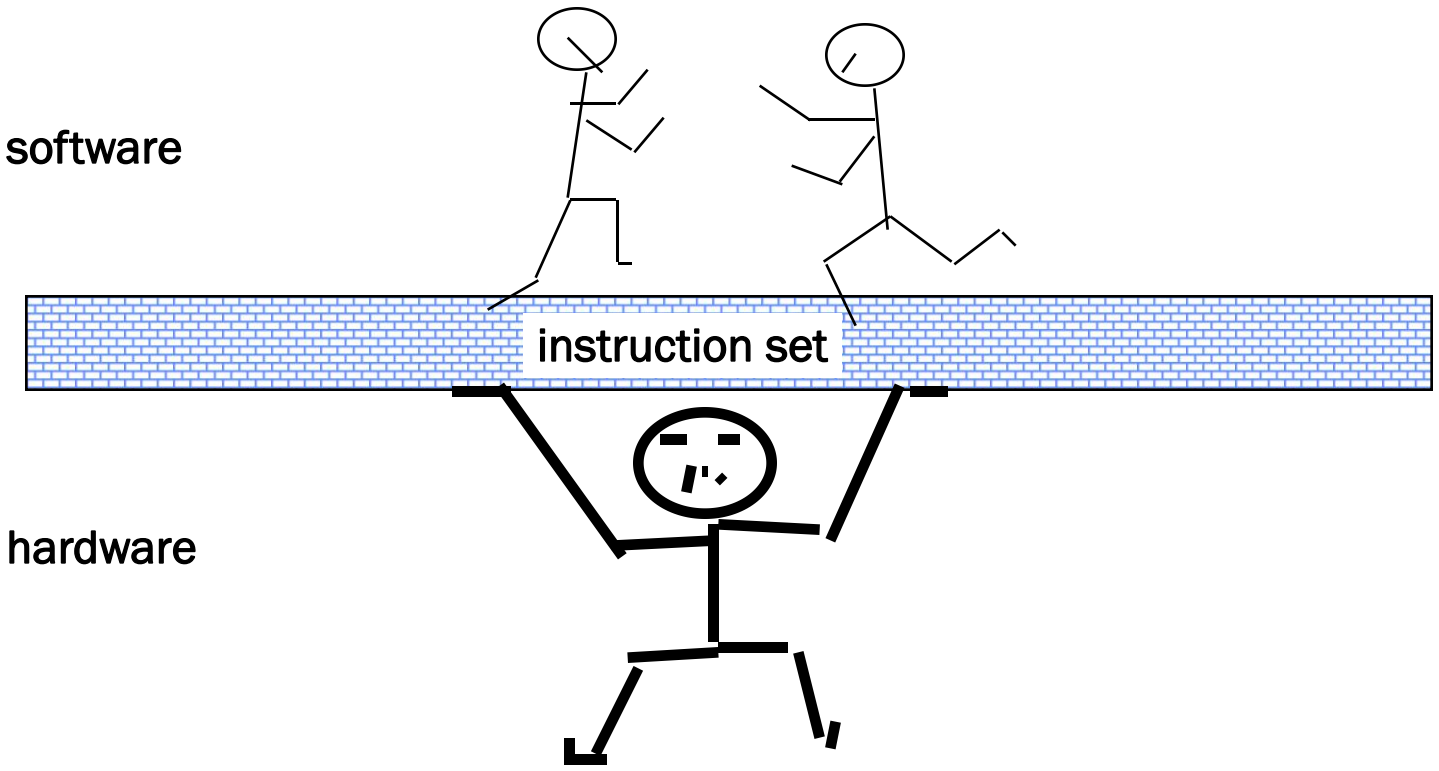
狭隘的观点（narrow view）：

包括：软件设计者与硬件设备设计者（VLSI）之间的中间层（ISA），以及微体系结构

扩展的观点（expanded view）：

包括：算法层、程序/语言层、系统软件层、ISA层、微体系结构、逻辑电路层、以及器件层

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)
  - ……

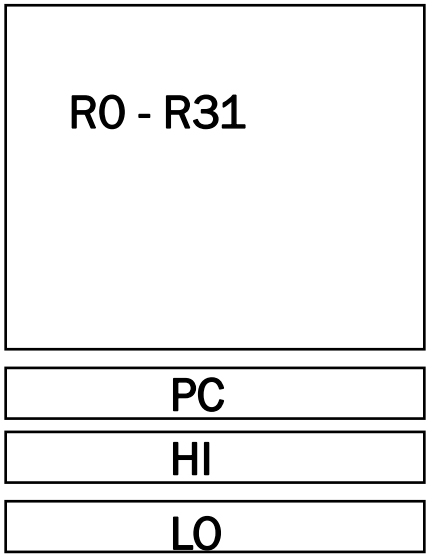
指令集结构举例

- 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, ...)
- RISC-V now

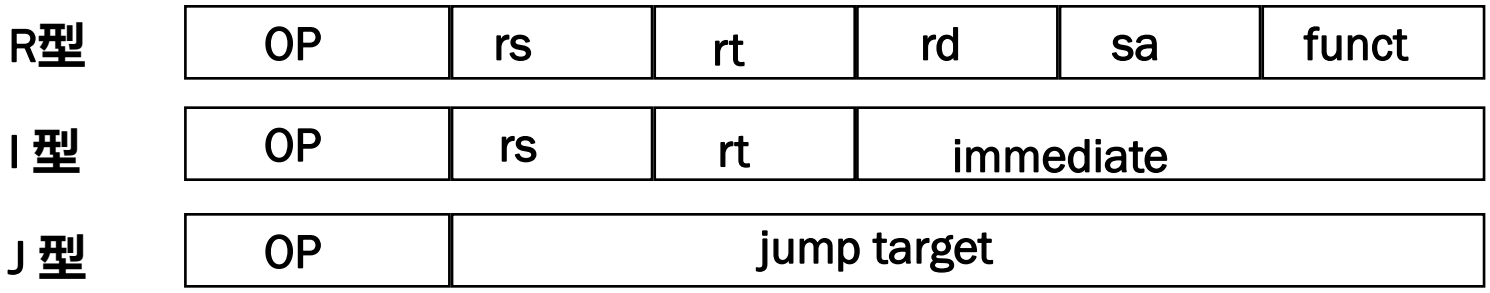
MIPS R3000 Instruction Set Architecture (Summary)

- **指令类型**
  - Load/Store
  - Computational
  - Jump and Branch
  - Floating Point
    - coprocessor
  - Memory Management
  - Special

Registers



3 种指令格式: all 32 bits wide

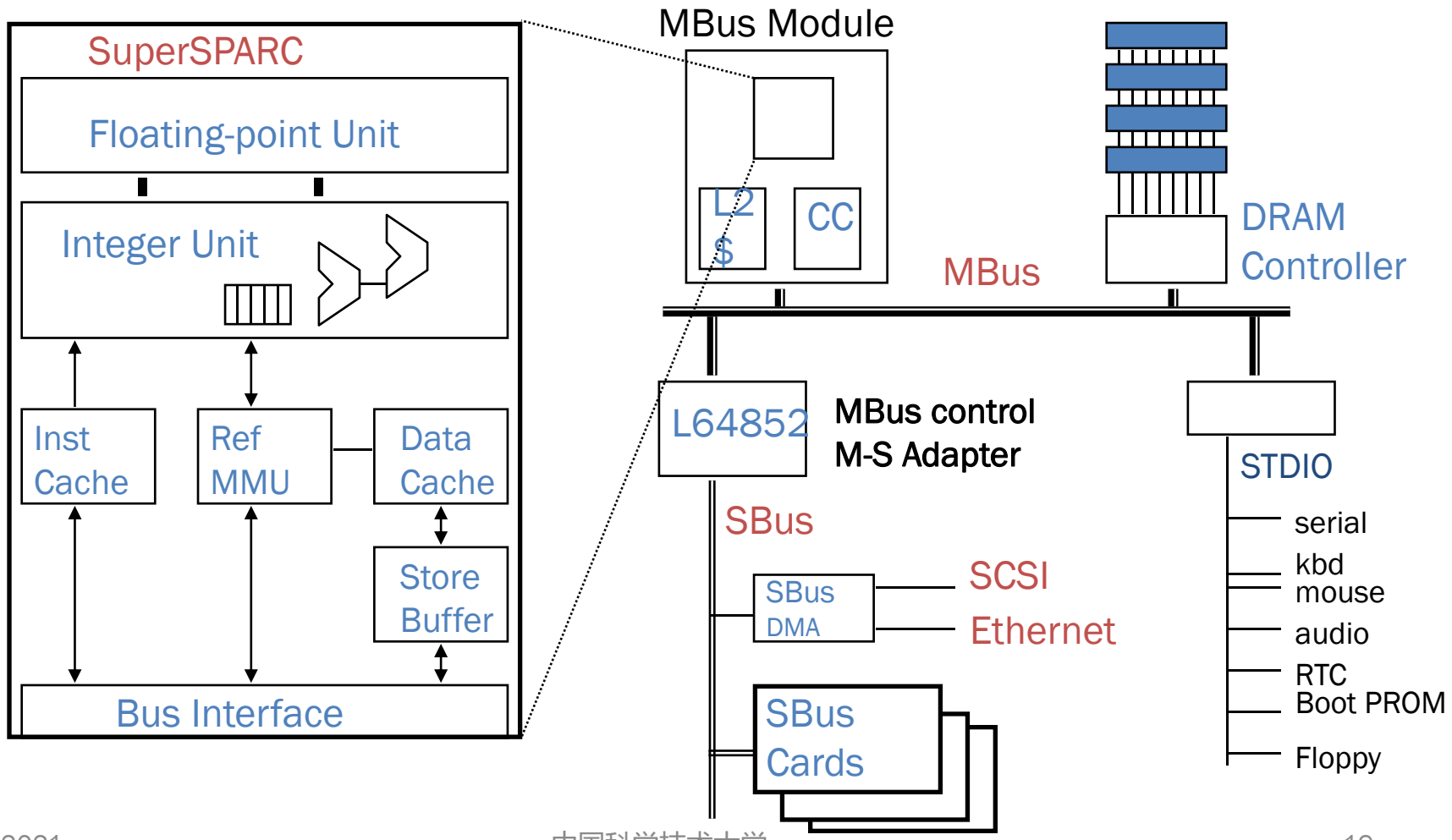


计算机组成与实现

- **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

计算机体系结构研究范畴

早期的定义：Instruction-Set Architecture

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

狭隘的观点（narrow view）：

包括：软件设计者与硬件设备设计者（VLSI）之间的中间层（ISA），以及微体系结构

扩展的观点（expanded view）：

包括：算法层、程序/语言层、系统软件层、ISA层、微体系结构、逻辑电路层、以及器件层

1.1 引言

什么是计算机体系结构

为什么要学习、研究计算机体系结构

本课程的基本要求

Computer Architecture

- **计算机体系结构：**
  - 设计、选择和连接硬件组件以及设计硬件/软件接口以创建一个满足功能、性能、能耗、成本和其他具体目标的计算系统的科学和艺术。
- **目的：实现一组设计目标**
  - E.g., 在工作负载X, Y, Z上的最高性能
  - E.g., 最长的电池寿命，可装进口袋，成本< X
  - E.g., 在所有已知工作负载中以最佳性能/成本比获得最佳平均性能…
  - 设计一台超级计算机与设计一部智能手机设计目标是不同的，但是，许多基本原则是相似的

Different Platforms, Different Goals

Source: http://www.sia-online.org (semiconductor industry association)

Different Platforms, Different Goals

Source: https://iq.intel.com/5-awesome-uses-for-drone-technology/

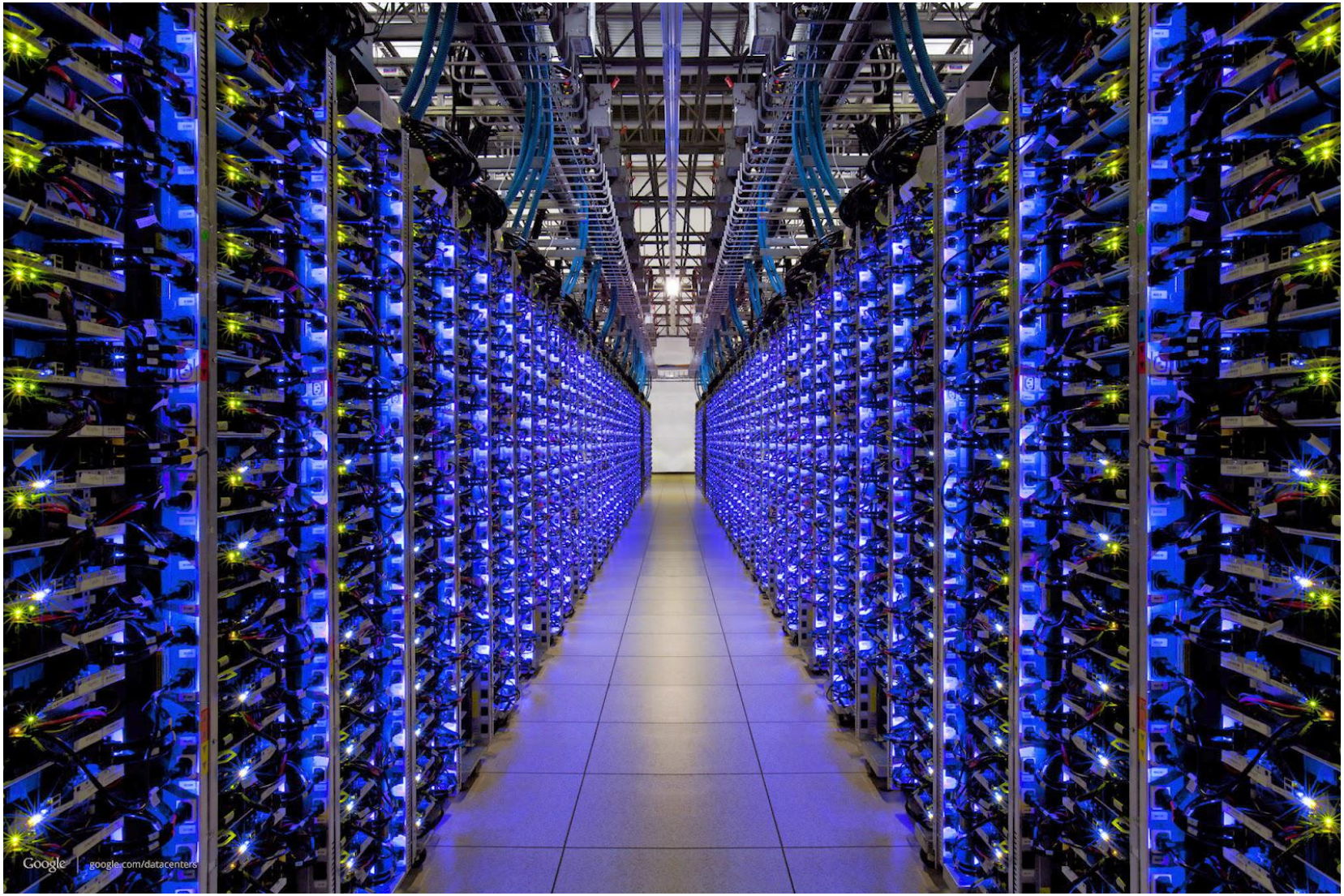
Different Platforms, Different Goals

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

Different Platforms, Different Goals

Source: http://sm.pcmag.com/pcmag_uk/photo/g/google-self-driving-car-the-guts/google-self-driving-car-the-guts_dwx8.jpg

Different Platforms, Different Goals



Source: http://datacentervoice.com/wp-content/uploads/2015/10/data-center.jpg

Different Platforms, Different Goals

Source: https://fossbytes.com/wp-content/uploads/2015/06/Supercomputer-TIANHE2-china.jpg

Different Platforms, Different Goals

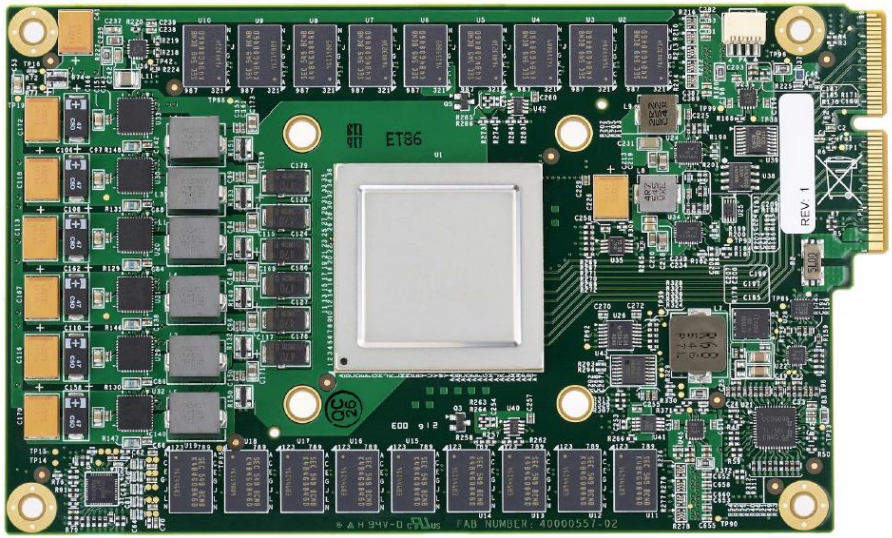


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

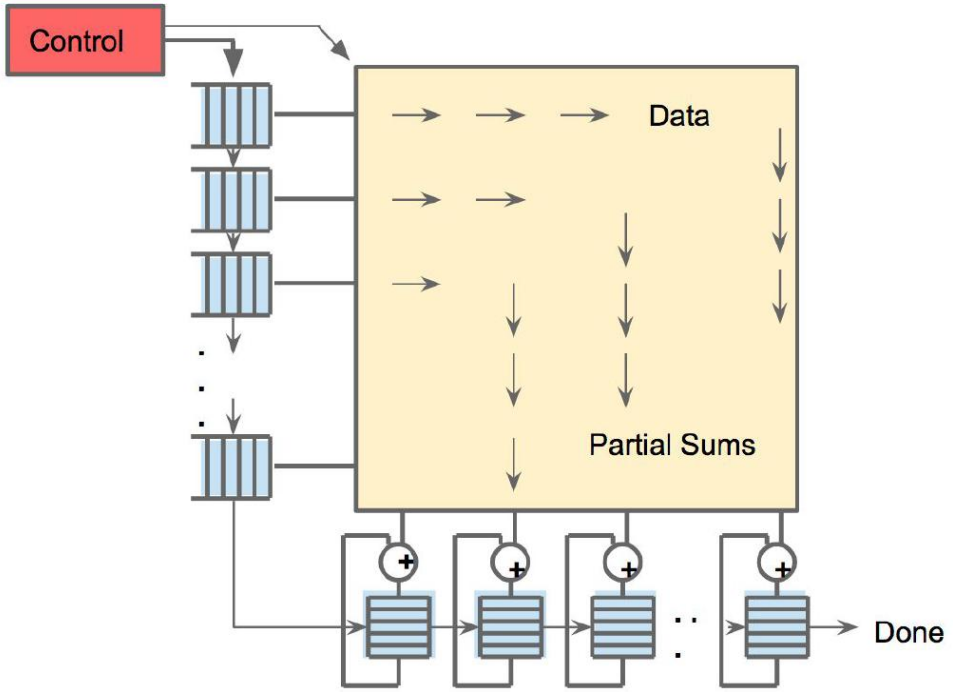
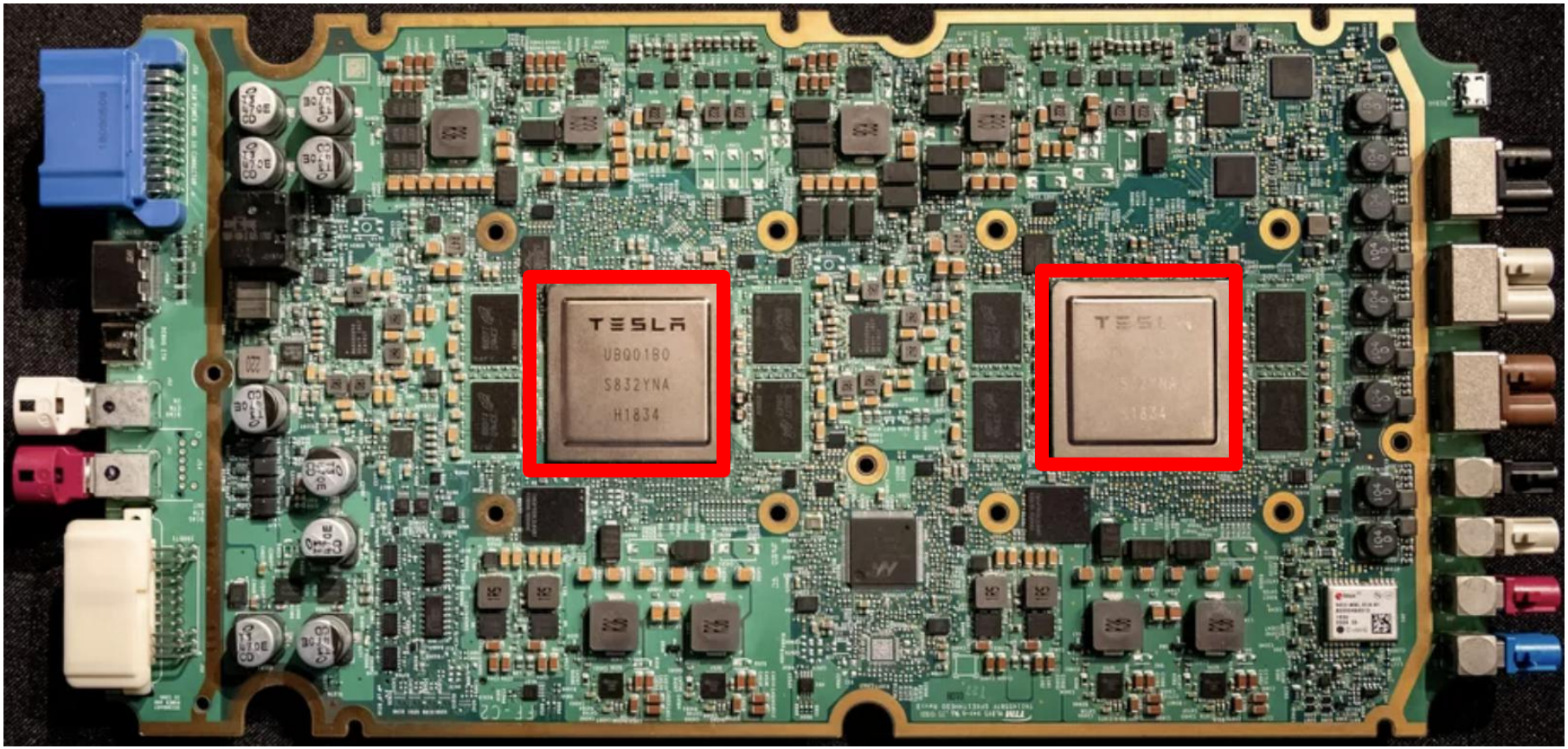


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

Different Platforms, Different Goals

- ML accelerator: 260 mm^2, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- *Two redundant chips for better safety.*



https://youtu.be/Ucp0TTmvqOE?t=4236

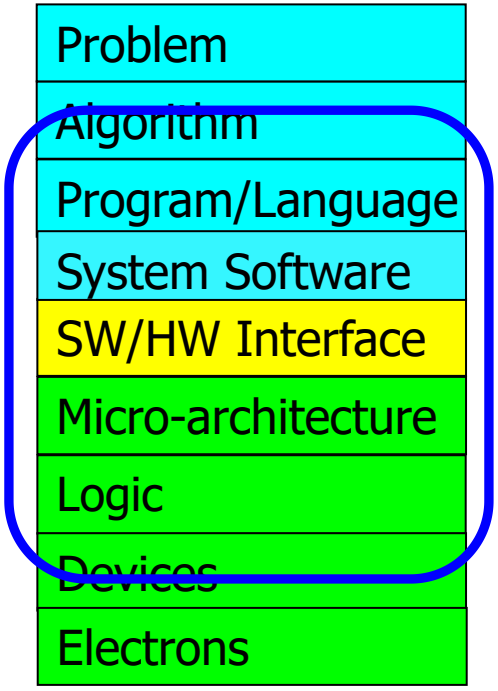
计算机体系结构设计者的任务

- **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、能效、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

Axiom

为达到最高的能效和性能:

我们必须以扩展的眼光看待计算机架构



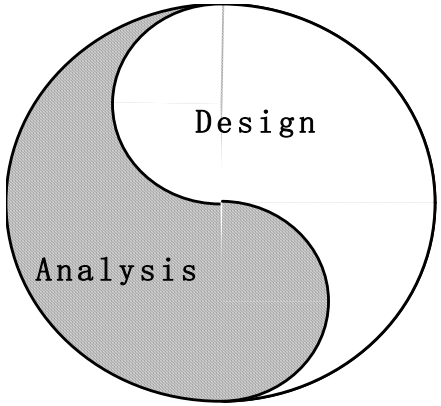
算法层到器件层的跨层次协同设计

在设计目标范围内尽可能地定制化

体系结构进步带来的益处

- **实现更好的系统：使计算机更快、更便宜、更小、更可靠……**
  - By exploiting advances and changes in underlying technology/circuits
- **算力的提高使新的应用成为可能**
  - Life-like 3D visualization 20 years ago? Virtual reality?
  - Self-driving cars?
  - Personalized genomics? Personalized medicine?
- **能够更好地解决实际问题**
  - Software innovation is built on trends and changes in computer architecture
    - > 50% performance improvement per year has enabled this innovation

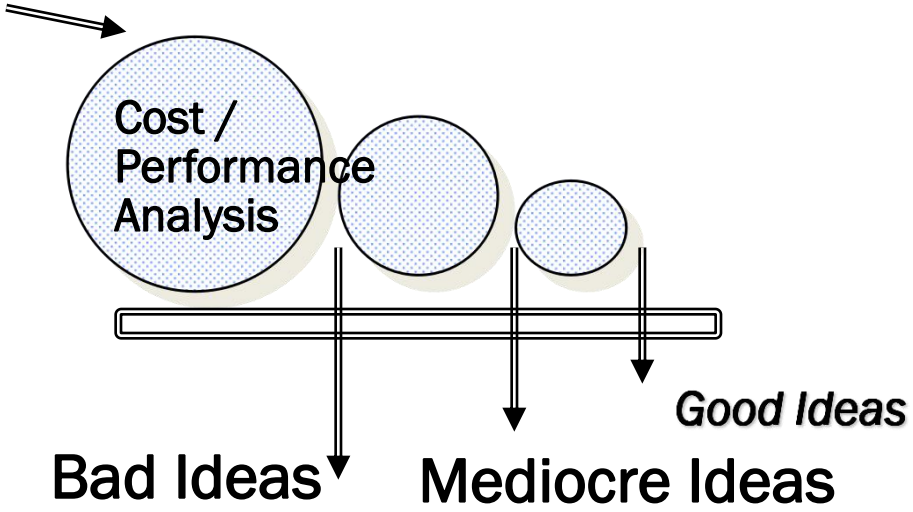
计算机体系结构设计过程

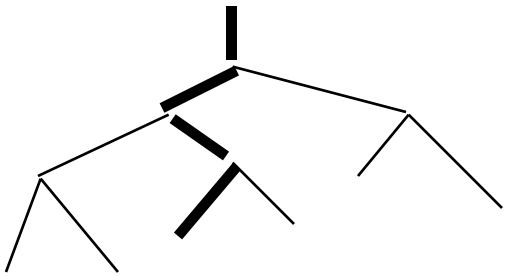


体系结构设计是循环渐进的过程:

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

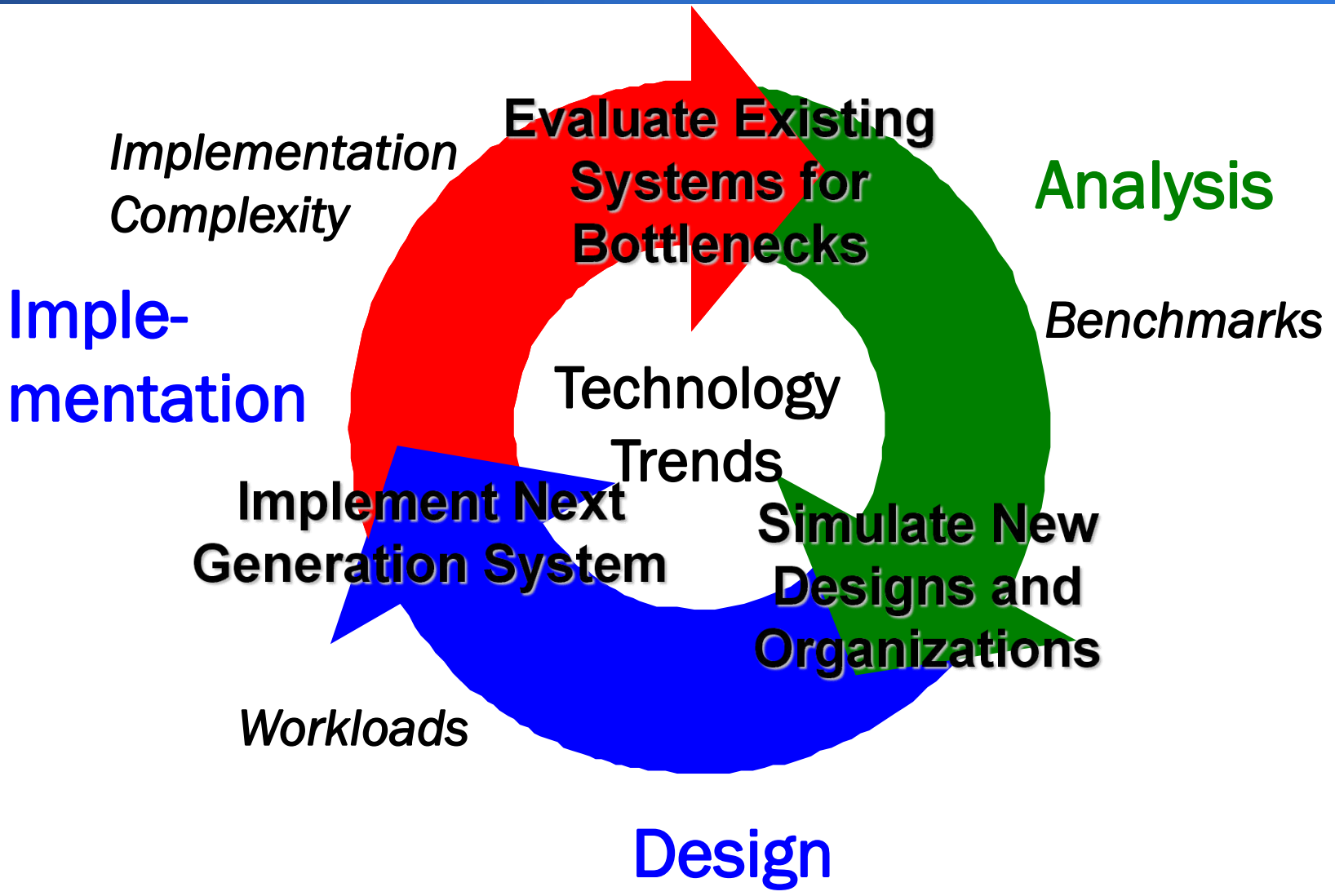
Creativity





Good measurement tools are required to accurately evaluate the selection.

计算机工程方法学


Evaluate Existing Systems for Bottlenecks
Analysis
Benchmarks
Simulate New Designs and Organizations
Design
Workloads
Implement Next Generation System
Implementation
Imple-
mentation
Technology Trends
Implementation Complexity

为什么要学计算机体系结构

深入理解计算机体系结构:

- **更好地理解计算机的工作原理**
- **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

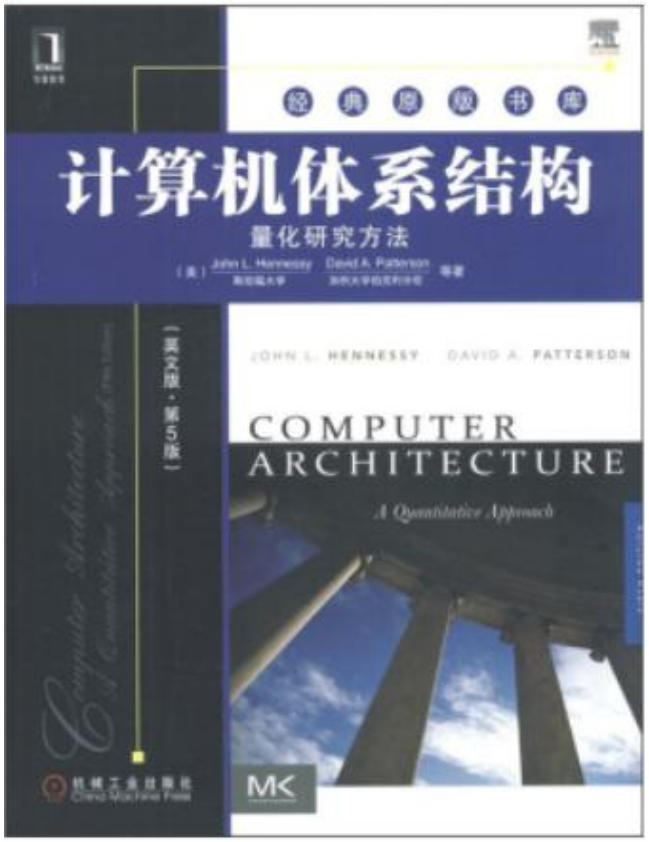
为什么要学习计算机体系结构

本课程的基本要求

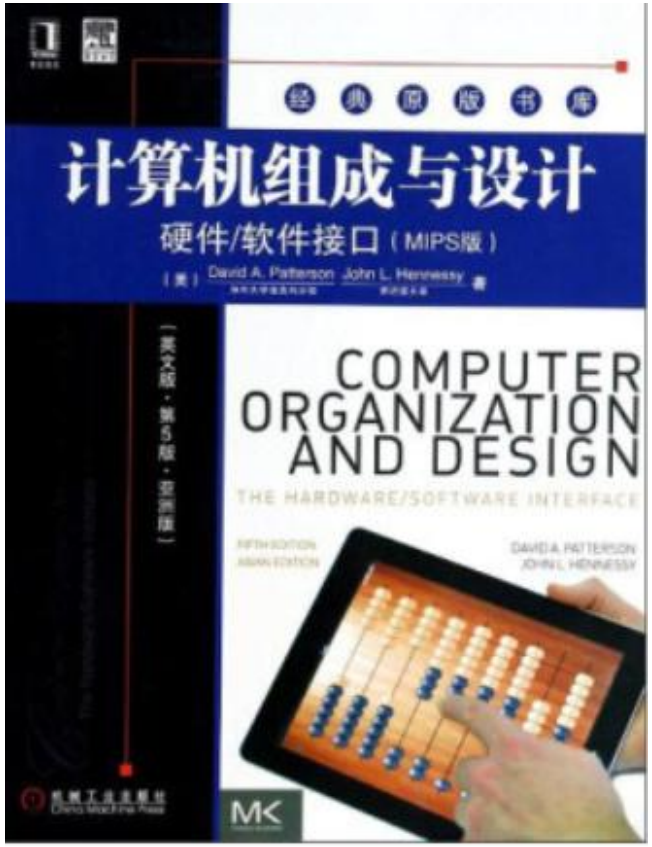
课程目标

- **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

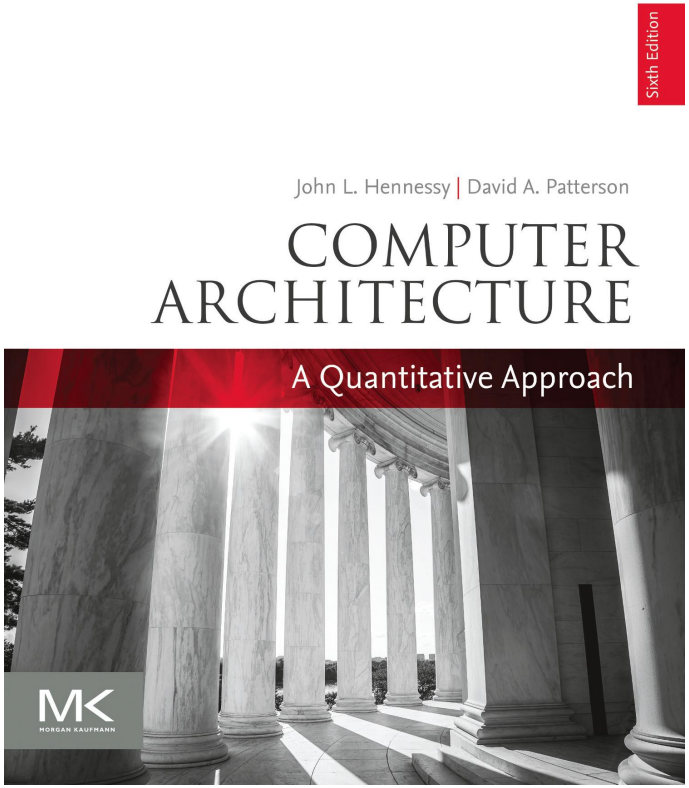


John L. Hennessy, David A. Patternson；Computer Architecture: A Quantitative Approach. Fifth Edition. 机械工业出版社，2012

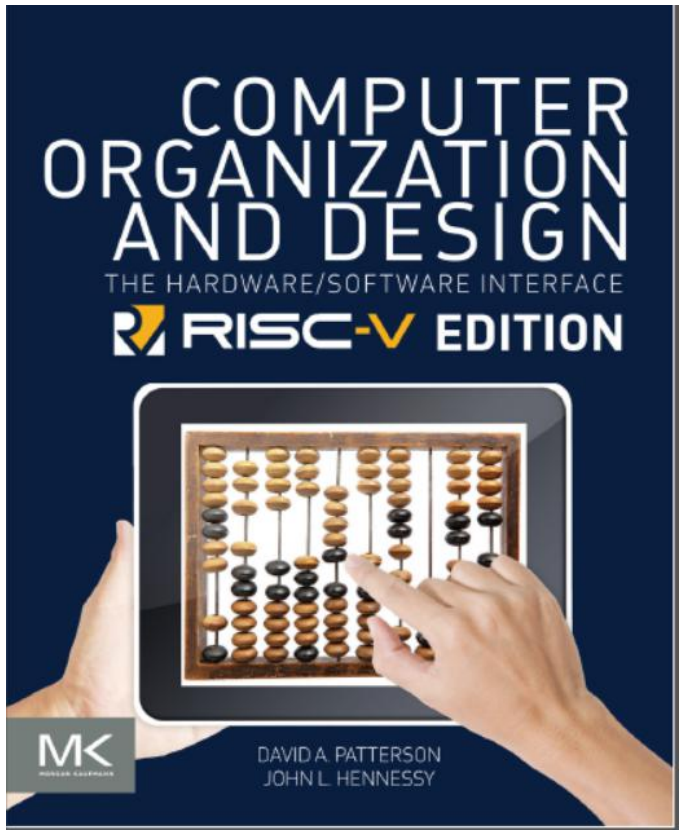


David A. Patternson, John L. Hennessy, Computer Organization & Design : The Hardware/Software Interface, Third Edition. San Francisco: Morgan Kaufmann Publishers, Inc. 2005

教材与主要参考书

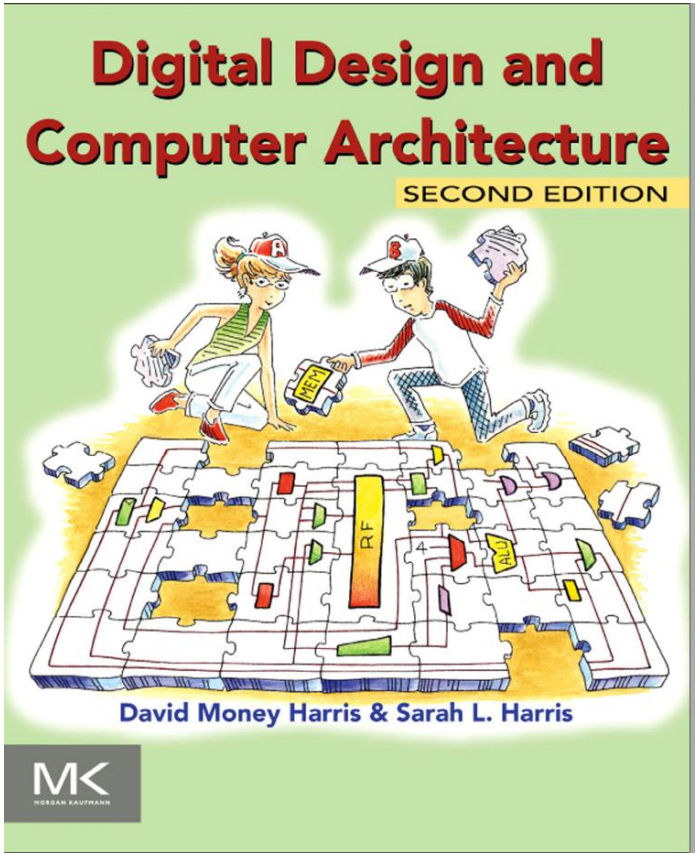


John L. Hennessy, David A. Patternson; Computer Architecture: A Quantitative Approach; sixth Edition.



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

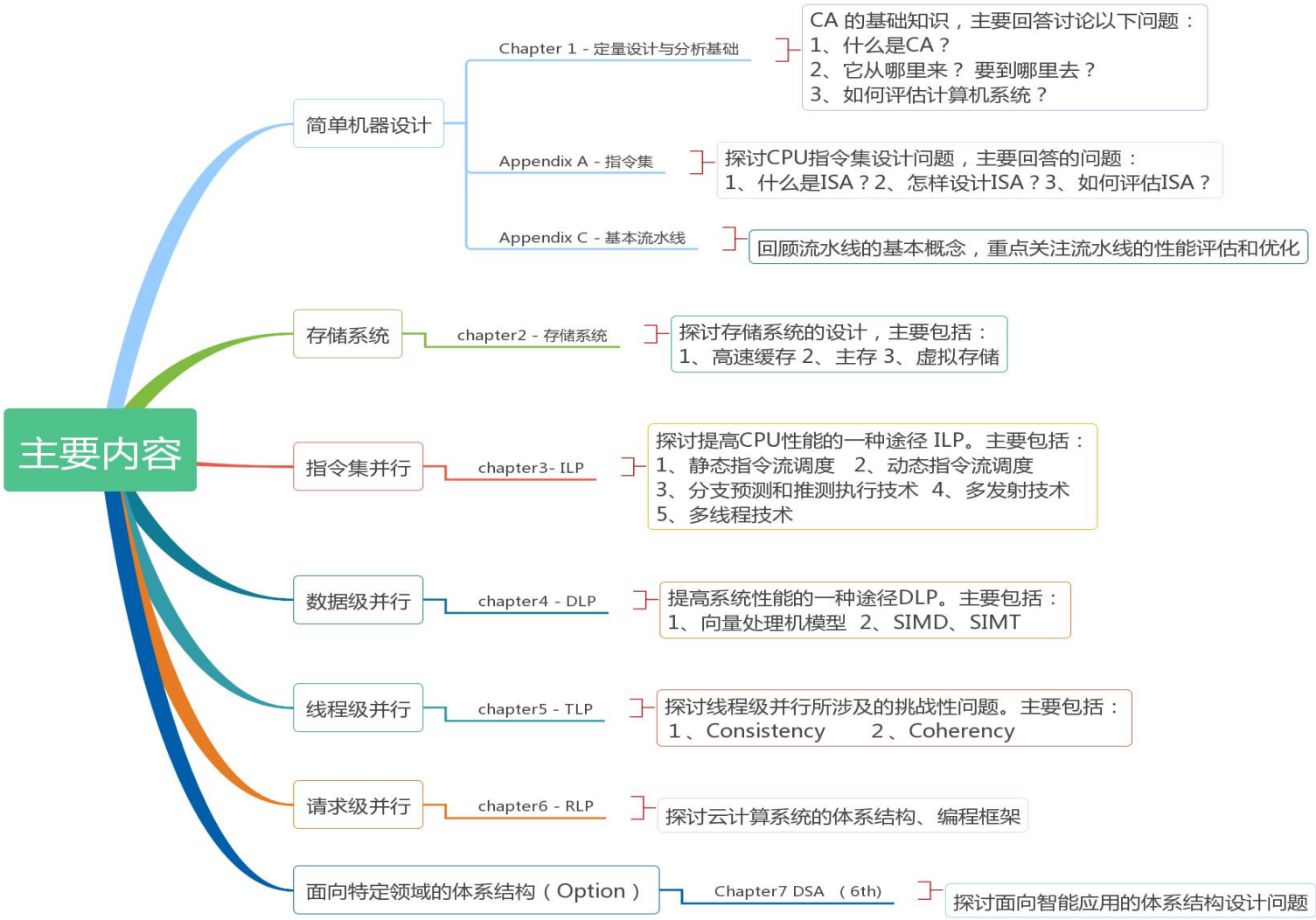
教材与主要参考书



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


主要内容
简单机器设计
Chapter 1 - 定量设计与分析基础
CA 的基础知识，主要回答讨论以下问题：
1、什么是CA？
2、它从哪里来？ 要到哪里去？
3、如何评估计算机系统？
Appendix A - 指令集
探讨CPU指令集设计问题，主要回答的问题：
1、什么是ISA？2、怎样设计ISA？3、如何评估ISA？
Appendix C - 基本流水线
回顾流水线的基本概念，重点关注流水线的性能评估和优化
存储系统
chapter2 - 存储系统
探讨存储系统的设计，主要包括：
1、高速缓存 2、主存 3、虚拟存储
指令集并行
chapter3- ILP
探讨提高CPU性能的一种途径 ILP。主要包括：
1、静态指令流调度 2、动态指令流调度
3、分支预测和推测执行技术 4、多发射技术
5、多线程技术
数据级并行
chapter4 - DLP
提高系统性能的一种途径DLP。主要包括：
1、向量处理机模型 2、SIMD、SIMT
线程级并行
chapter5 - TLP
探讨线程级并行所涉及的挑战性问题。主要包括：
1 、Consistency 2 、Coherency
请求级并行
chapter6 - RLP
探讨云计算系统的体系结构、编程框架
面向特定领域的体系结构（Option）
Chapter7 DSA （6th）
探讨面向智能应用的体系结构设计问题

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(011135.02)**
  - 授课总学时60学时，实验30学时
  - 3C203: 1(3,4)，5(3,4)
- **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）**

体系结构的定义

- **计算机体系结构**
  - 设计、选择和连接硬件组件以及设计硬件/软件接口，以创建一个满足功能、性能、能耗、成本和其他具体目标的计算系统的科学和艺术。
  - 体现为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。

- **目标约束**
  - 实现一组设计目标。性能、能效等
  - 不同的平台有不同的设计目标

计算机体系结构研究范畴

早期的定义：Instruction-Set Architecture

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

狭隘的观点（narrow view）：

包括：软件设计者与硬件设备设计者（VLSI）之间的中间层（ISA），以及微体系结构

扩展的观点（expanded view）：

包括：算法层、程序/语言层、系统软件层、ISA层、微体系结构、逻辑电路层、以及器件层

主观题 10分

设置

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

作答

Acknowledgements

- **These slides contain material developed and copyright by:**
  - John Kubiatowicz (UCB)
  - Krste Asanovic (UCB)
  - John Hennessy (Standford)and David Patterson (UCB)
  - Chenxi Zhang (Tongji)
  - Muhamed Mudawar (KFUPM)
  - Onur Mutlu（ETH Zürich）
- **UCB material derived from course CS152、CS252、CS61C**
- **KFUPM material derived from course COE501、COE502**