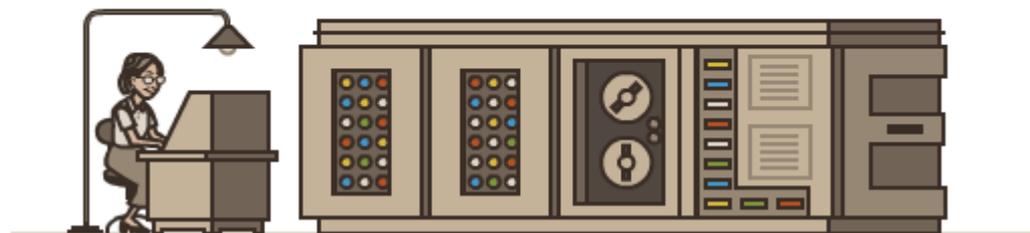


计算机组成原理

课程概要：内容，方法

李曦

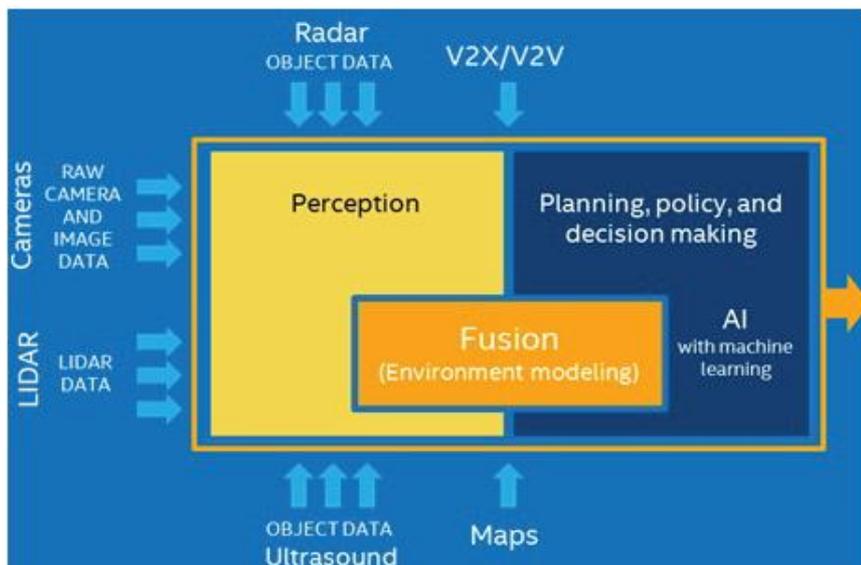
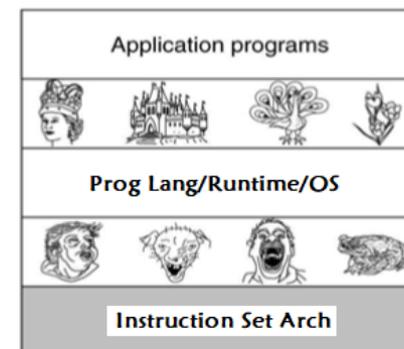
llxx@ustc.edu.cn



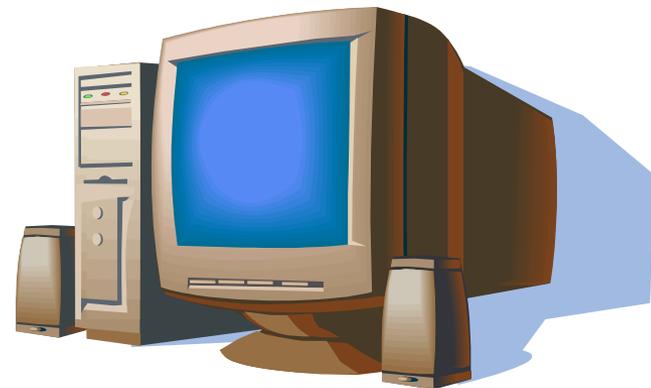
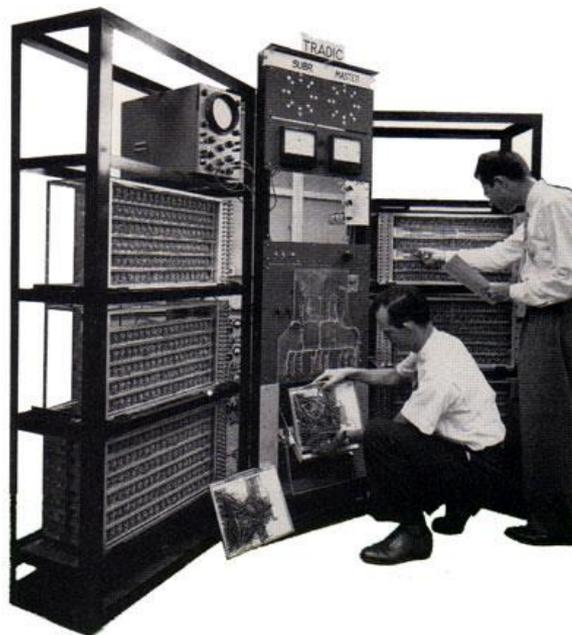
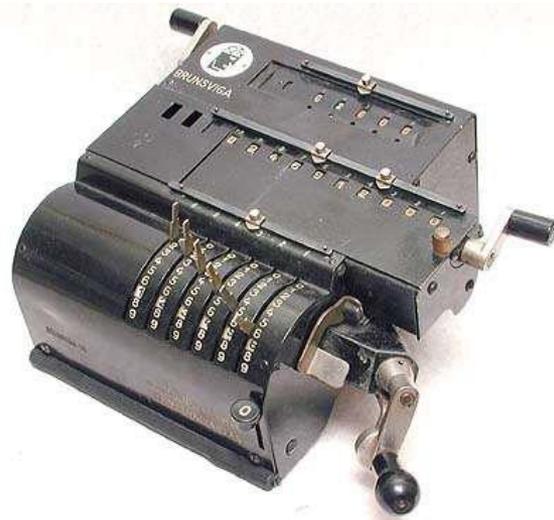
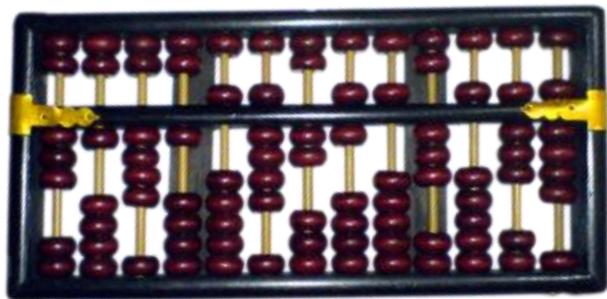
李曦： Who's Who😊



- 教师@软件学院/计算机学院
 - 博士，教授，博导
 - 研究方向：实时计算机系统结构
 - 实时系统，嵌入式智能系统
 - 科学问题：系统的时间行为可预测性与可组合性

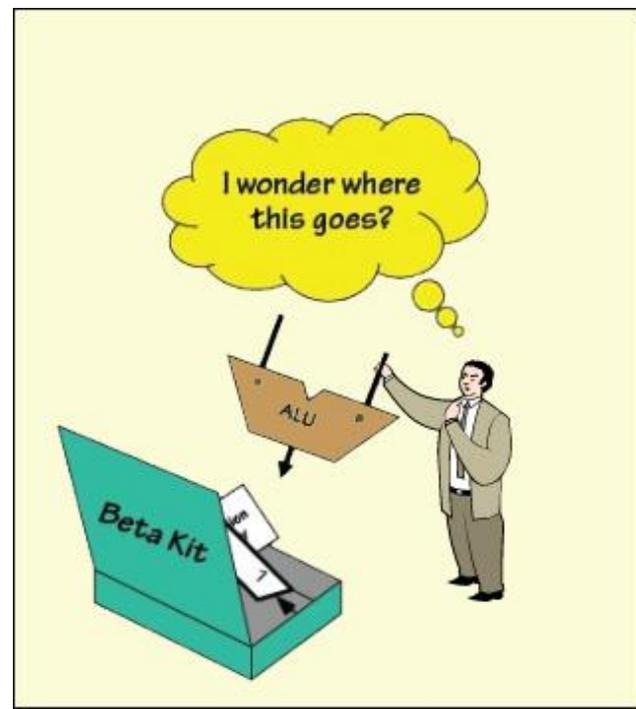


计算机组成结构，最小的？



算盘 \approx 计算器 \neq 计算机?
Tiny?

计算机在干啥？





程序执行用到哪些部件？正确性？

```
/*---sum.c---*/
```

```
int sum(int a[ ], unsigned len)
```

```
{  
    int i, sum = 0;  
    for (i = 0; i <= len-1; i++)  
        sum += a[i];  
    return sum;  
}
```

```
/*---main.c---*/
```

```
int main()
```

```
{  
    int a[1]={100};  
    int s;  
    s=sum(a,0);  
    printf("%d",s);  
}
```

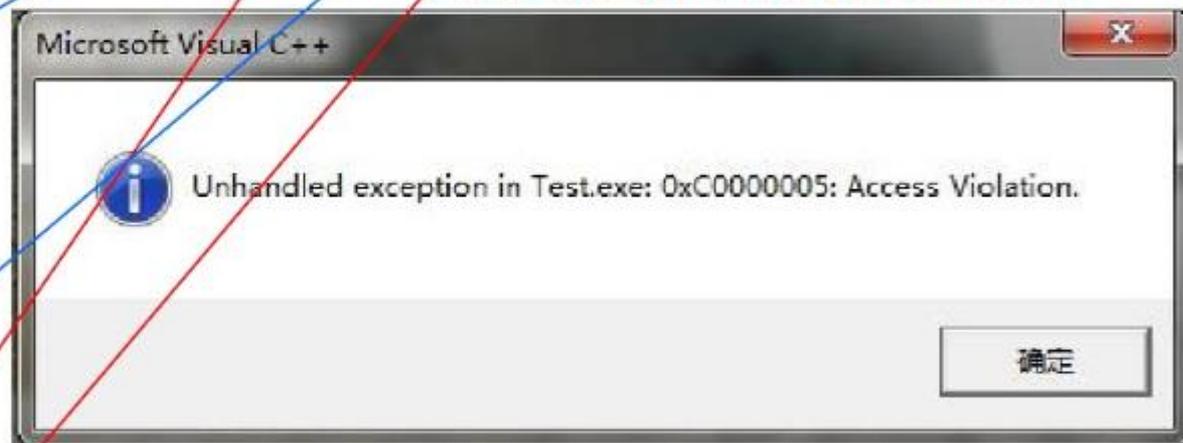
数据的表示

数据的运算

各类语句的转换与表示(指令)

各类复杂数据类型的转换表示

过程(函数)调用的转换表示



链接(linker)和加载

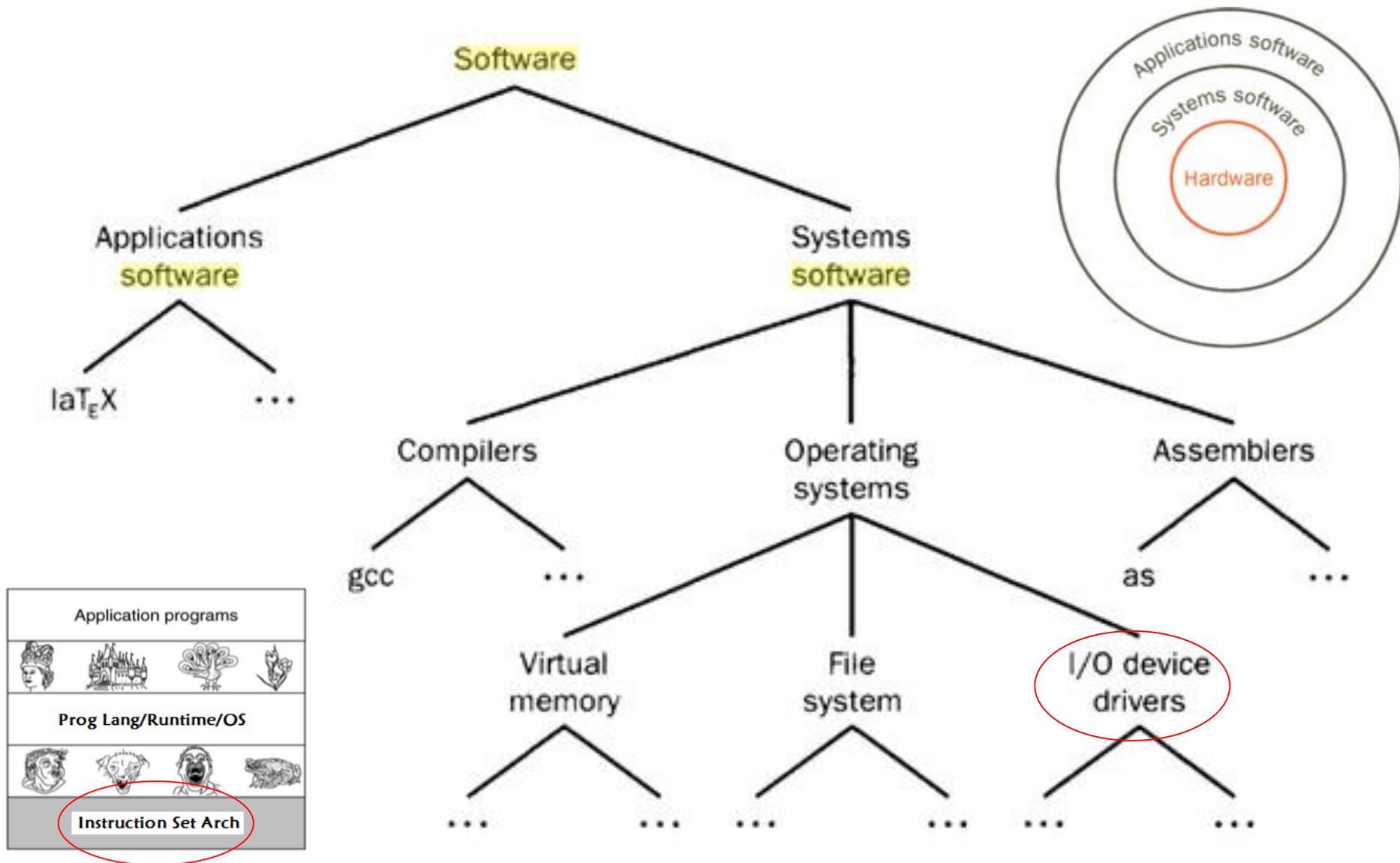
程序执行(存储器访问)

异常和中断处理

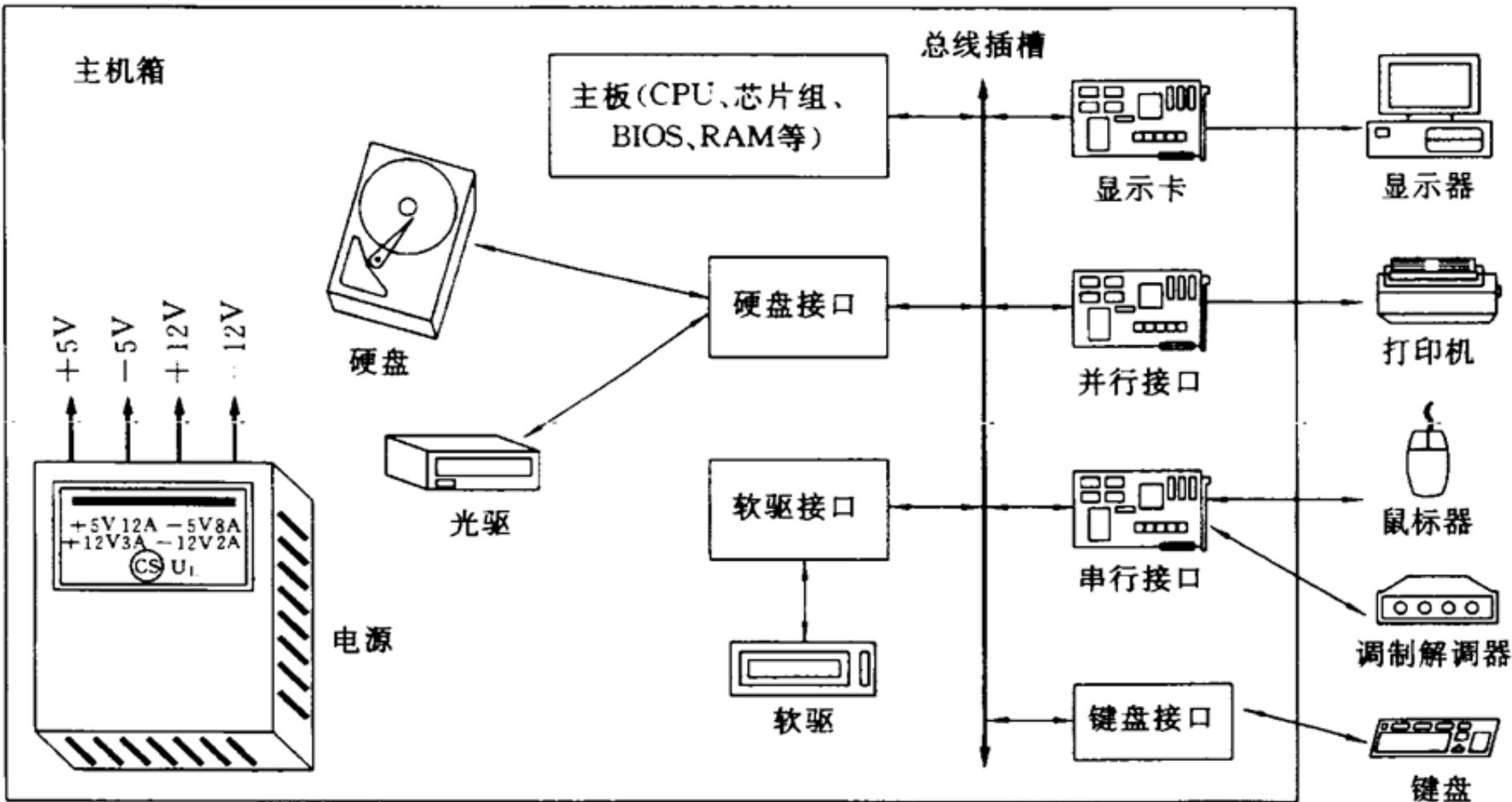
输入输出(I/O)

- C语言计算机模型？

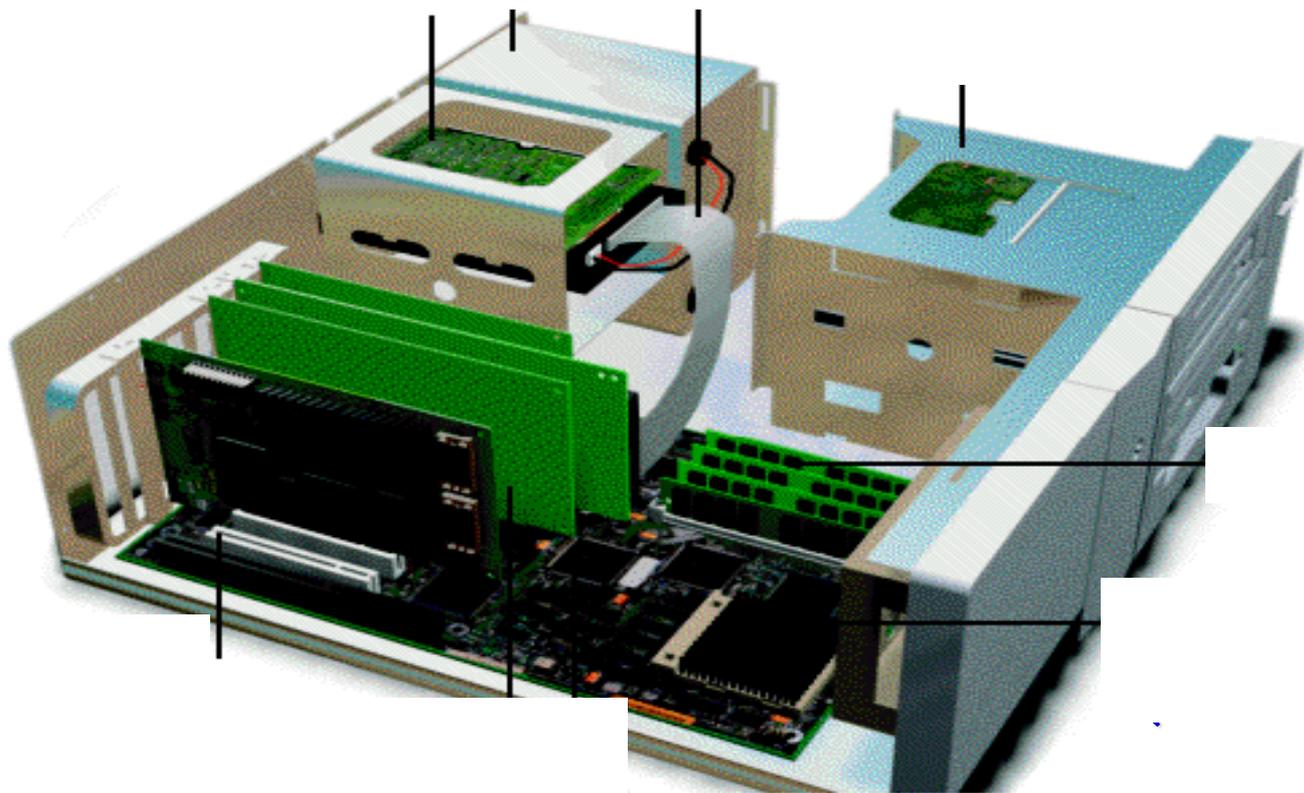
计算机组成：软件组件，与软硬件接口

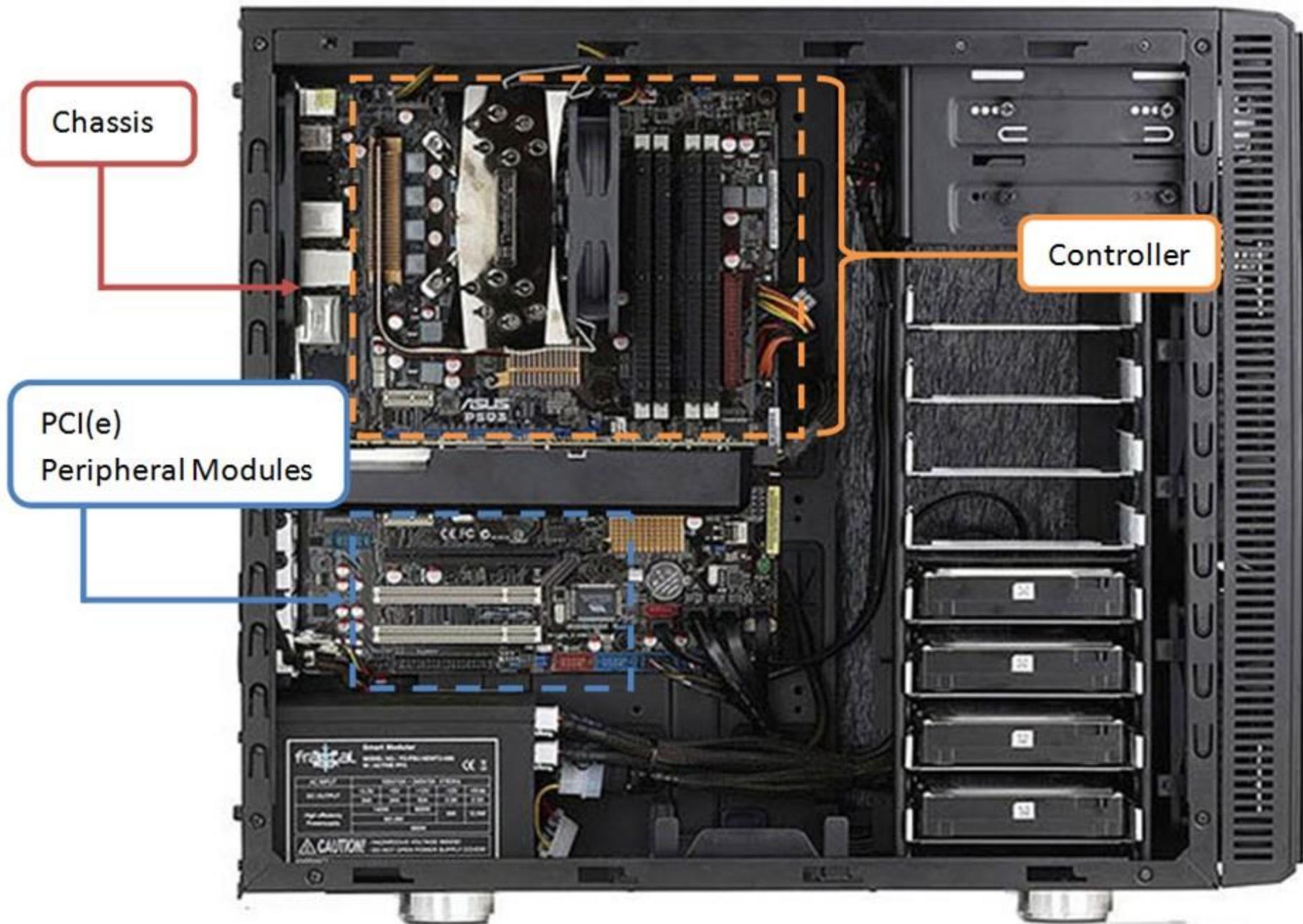


计算机组成：硬件组件，与接口

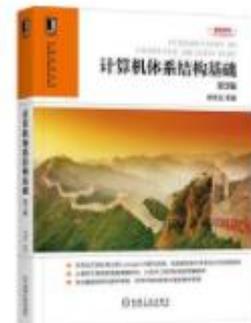
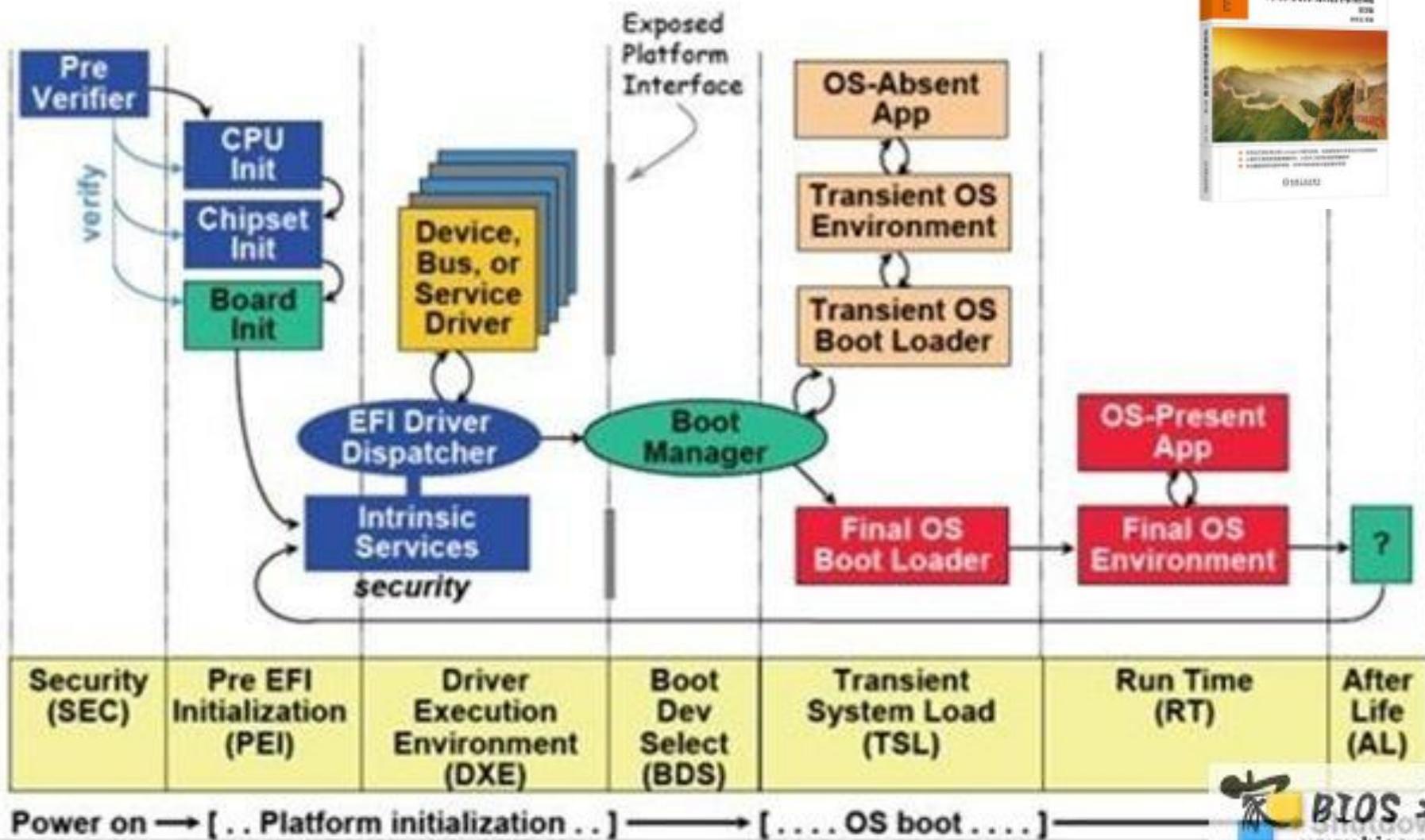


计算机组成 (CBD, 基于组件的设计)





何时开始执行你的C程序？ 胡伟武Ch7

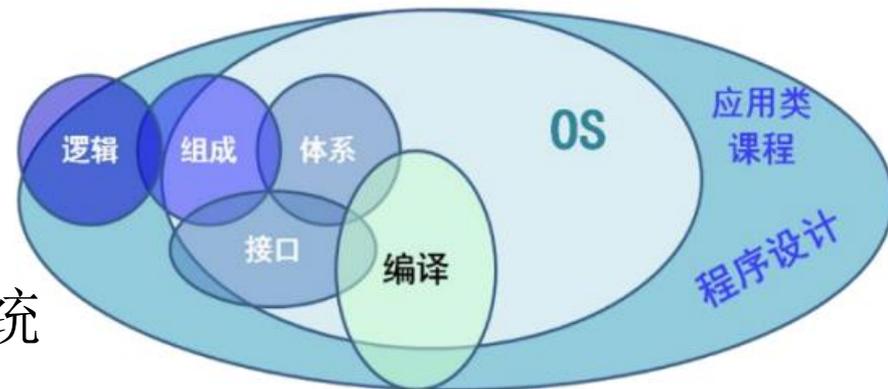
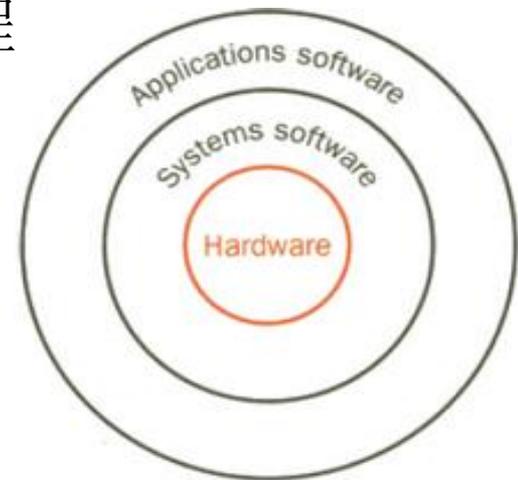


COD课程的作用： what, how, why



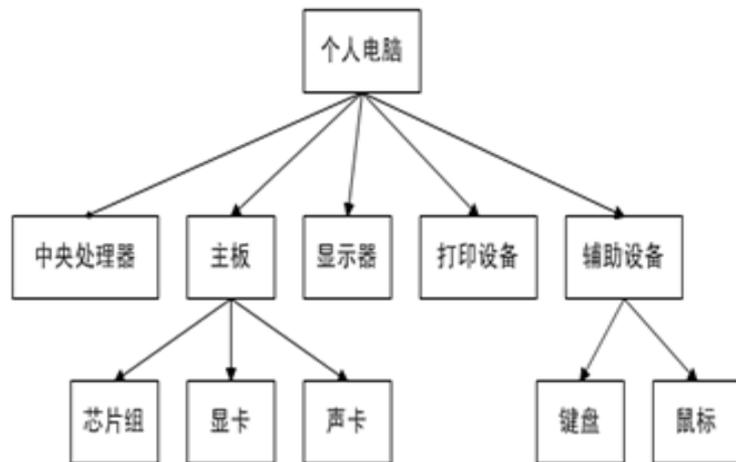
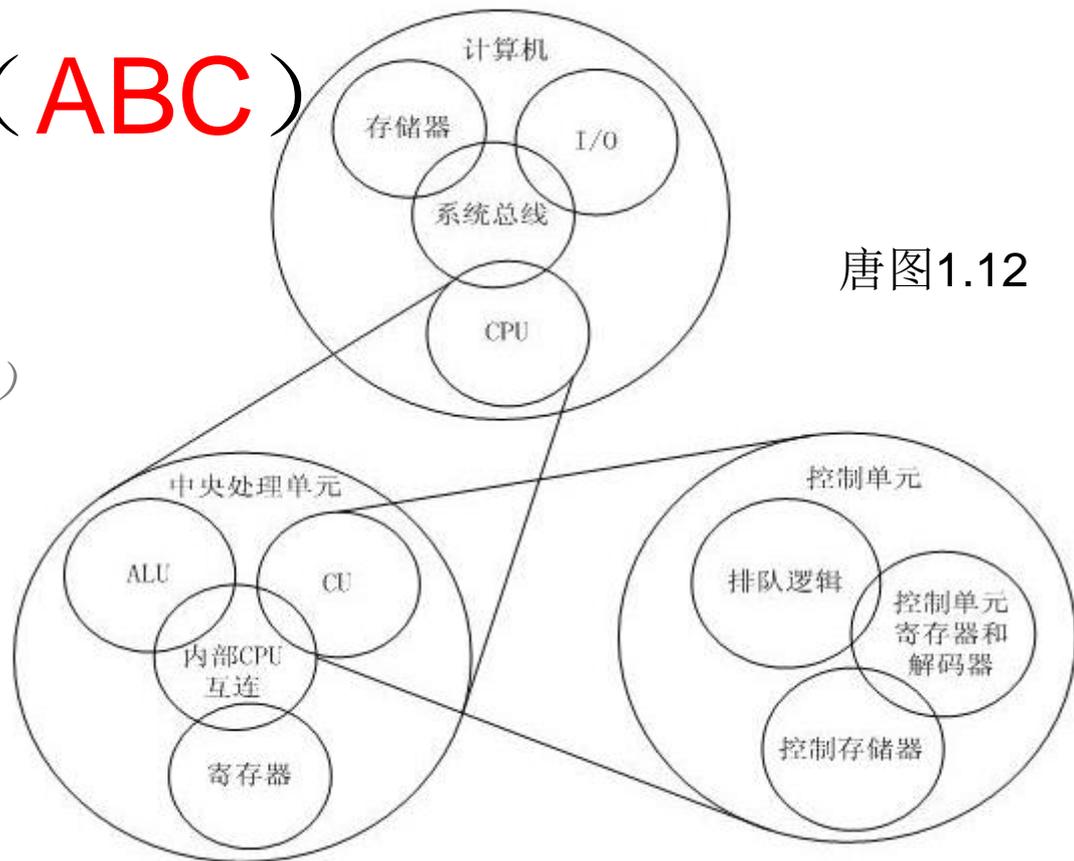
- 专业核心课：所有专业课的基础
 - 所有大学计算机系都开设COD相关课程
- 密切相关课程
 - 数字逻辑电路，程序设计
 - 组成原理
 - 重点“单核单处理器”系统
 - 延伸：多核多处理器系统
 - 强调软硬件接口
 - 体系结构：量化设计
 - 操作系统
 - 编译原理
 - 微机原理与接口/嵌入式系统
 - 并行与分布式系统

图1-3



课堂教学内容 (ABC)

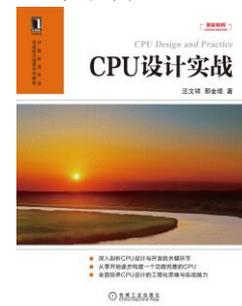
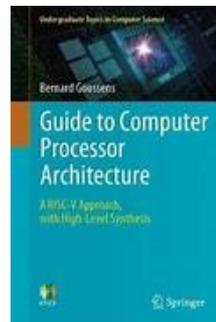
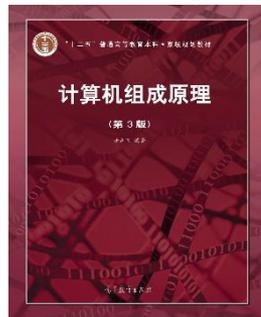
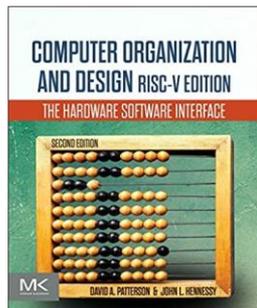
- 中央处理器 (CPU)
 - 数据表示
 - 算术逻辑运算 (ALU算法)
 - 指令系统
 - ALU、访存、分支、系统
 - 数据通路
 - 单周期、多周期、流水线
 - 控制单元
 - 异常与中断
- 外围部件
 - 系统总线
 - 总线结构、总线控制、总线性能
 - 存储器
 - 缓存、主存、辅存、虚存
 - 输入输出系统
 - I/O接口、I/O控制方式、外设





教材与参考书

- 《Computer Organization and Design: The HW-SW I/F》
 - David Patterson & John Hennessy, 5th, RV2版, 2018/2021
 - CPU和Cache强, MEM、总线、I/O弱
 - COD涉及并行和性能分析: 一些学校用作Computer architecture教材
- 《计算机组成原理》, 3rd, 2008/2020, 唐朔飞
 - 以William Stallings Ed. 4th为框架, 参考7th和COD 3rd
 - CISC, A模型CPU, 但无OS、VM、并行等内容
 - CPU和Cache弱, MEM、总线、I/O强
- “Guide to Computer Processor Architecture”, 2023
 - “A RISC-V Approach, with High-Level Synthesis”, Rernard Goossens
 - 从SOC角度, Vitis HLS, FPGA
 - 单核, SMT, 多核, AXI
- “CPU设计实战”, 2021
 - 汪文祥(龙芯首席工程师), 邢金璋
 - Verilog: 处理器核、高级缓存、总线接口控制器、DDR3/4内存控制器等





教材主要内容比较

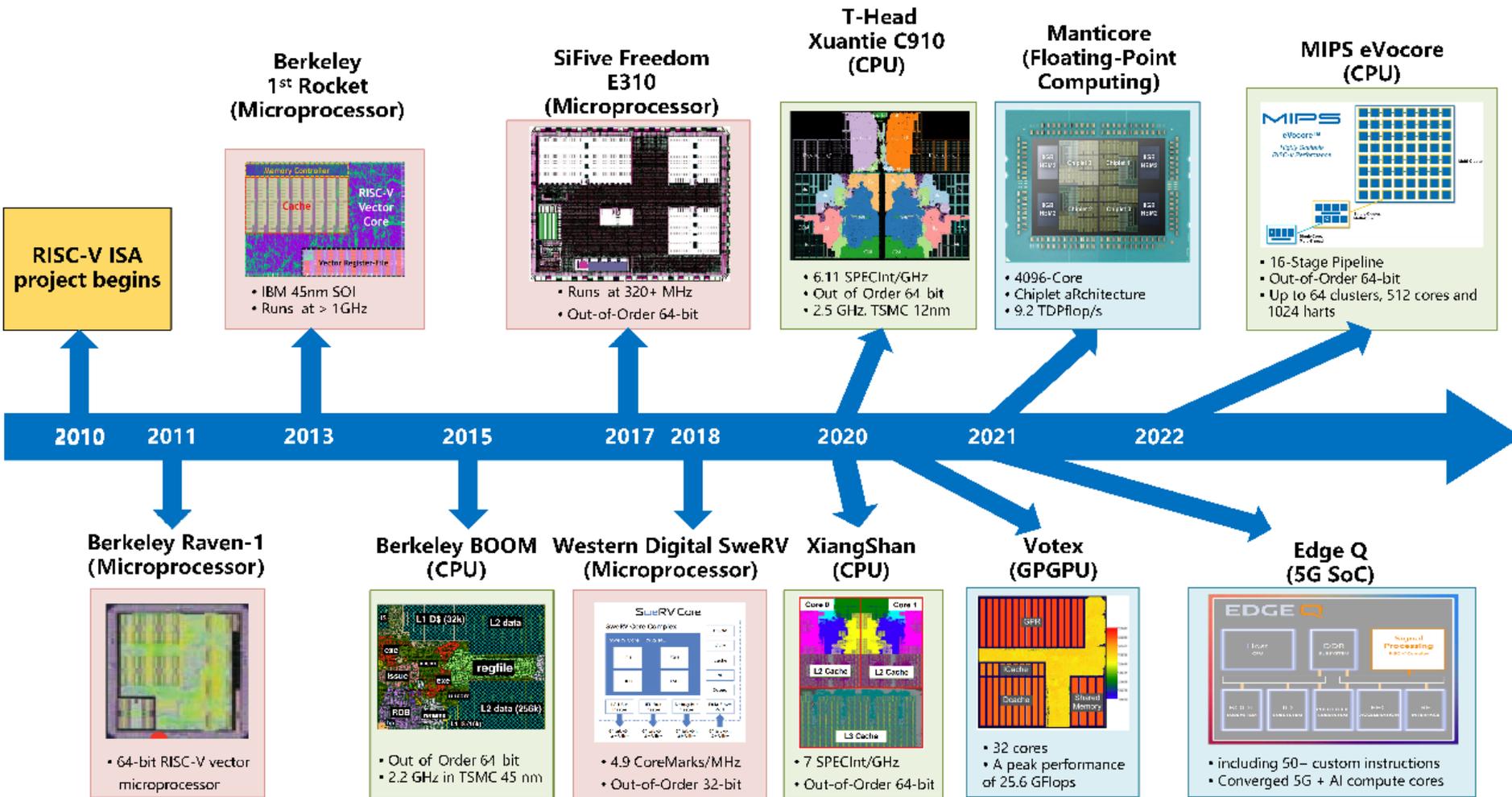
COD5: RISC-V, 2ed

1. 计算机概念与工艺
 2. 指令集与汇编语言
 3. ALU算法
 4. RISC处理器
 - 单周期、多周期、流水线
 - 异常: CP0
 - 微程序: COD5 (C.9)
 5. 存储器层次结构
 - SRAM、DRAM、FLASH、DISK
 - Cache、Cache控制器
 - 可靠性问题 (海明码)
 - 虚存
 6. 并行处理器
 - 硬件多线程
 - 多核
- COD5: “I/O不单独成章” !
- 总线?

- 唐朔飞: 以CISC为主
 - 第一部分 概论
 - 冯诺依曼机, 性能指标
 - 第二部分 计算机硬件系统
 - 总线, 内存, Cache, 外存, I/O
 - 第三部分 CPU
 - 算逻运算、ISA, CPU结构, 中断系统
 - 介绍了RISC、流水线概念
 - 第四部分 CU
 - A模型处理器, 微操作
 - 多周期: 硬连线实现, 微程序实现
- 强调系统完整性: 总线/内存/I/O
 - 唐本无: 虚存, 并行
- 考研: 唐朔飞or白中英

我们: COD5 (CPU、Cache、存储器、异常、MMU) + 唐 (总线、存储器、I/O、中断、DMA)

Typical RISC-V processors



Course Schedule

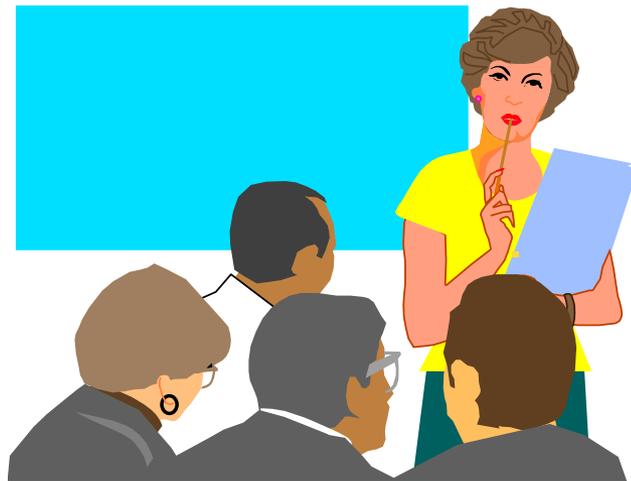


Lec	Topic	Read BEFORE class		Keypoint
		RV2	唐	
1	概论	CHP1	第一、二章	
	逻辑设计基础	4.2、附录 A		自学
2	指令系统	CHP2, 5.14		冒泡排序
	汇编程序设计	2.12、2.13,		COD4 附录 B
	RISC 处理器概述	附录 D		
3	RISC 数据通路 (单周期、多周期)	4.4, 4.5		
	控制器 (组合, 顺序, 微程序)	附录 C	微程序 10.2	
	RISC 数据通路 (流水线)	4.6、4.7、4.14		
	流水线冒险	4.8、4.9、4.14		
	多发射、HMT	4.11、6.4		
	异常与中断	4.10	8.4	
4	内存: SRAM, DRAM, ROM, SDRAM	5.2、A.9	第四章	
	Cache 原理	5.3、5.4	4.3	
	Cache Controllers	5.9、5.12		
	外存	5.2, 5.11	第四章	
	虚存	5.7		
	存储器的可靠性 (海明码)	5.5、5.11、A.9	第四章	
5	总线	6.9	第三章	
6	I/O	1.4、6.9	第五章	TinyComputer
7	多核架构: 原子指令, CCP	2.11、5.10、6.5		
8	A 模型 CPU 设计 (多周期, 微程序)		第八/九/十	
9	算术运算	CHP3	第六章	
10	专题: 并行, 非冯机, 低功耗	CHP1、CHP6		

课程成绩考核：发现**抄袭**取消成绩



- 课堂教学：70%
 - 期末考试：50%（闭卷）
 - 作业：30%，**不能补交**！
 - 课堂练习：10%（出勤5%，正确5%）
 - 课内实验：5%
 - 冒泡汇编，Tiny系统设计
 - 实验报告：参照**小论文**格式
 - 调研：2~3篇，5%
 - 调研报告：参照**小论文**格式
 - 资料来源：参考书、Internet
- 实验课：30%

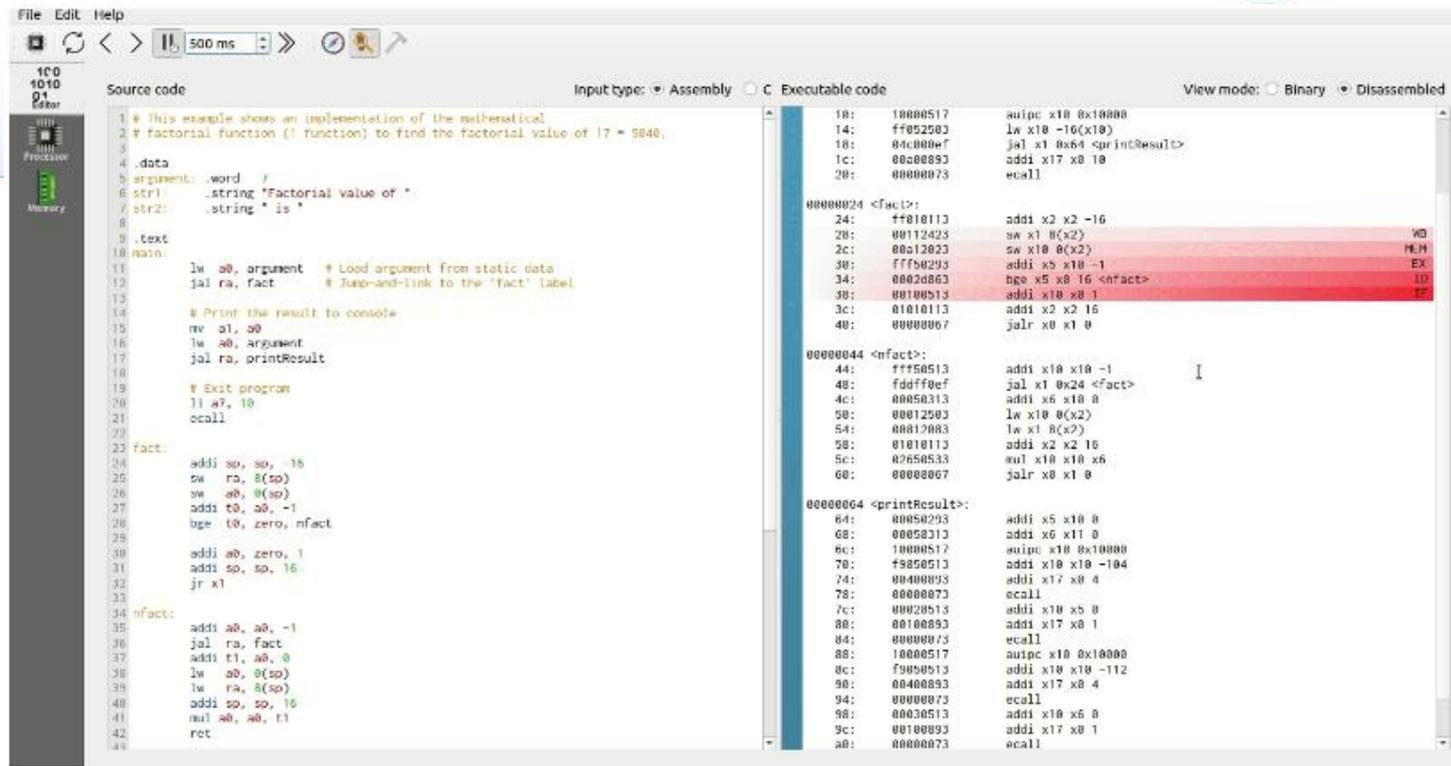


课内实验

```
-d
0B06:0100 36 B9 98 BF 76 97 B8 00-29 CD 21 80 3C 00 74 06 6...v...>.!<.t.
0B06:0110 BA 92 8A E9 53 FF A1 BC-98 80 FC 3A 34 00 F5 0A .....S.....:4...
0B06:0120 8A 0E D8 99 0C 20 2C 60-A2 76 97 A0 BB 98 8A 26 .....`..v....&
0B06:0130 1E 99 25 02 02 0A C0 74-13 3A C4 75 0F 80 3E D9 ..%.....t.:u..>.
0B06:0140 99 00 74 08 FE 06 17 99-32 C0 EB 06 34 02 22 C4 ..t....2...4..."
0B06:0150 D0 E8 0A 06 D9 99 A2 C7-96 D0 E0 D0 E0 A2 D2 99 .....
0B06:0160 80 3E D4 99 00 75 24 A2-D8 99 0A C9 75 1D 0A C0 ->...u$.u...
0B06:0170 74 19 8B 0E D5 96 E3 13-B0 1A 06 33 FF 8E 06 B4 t.....3....
```

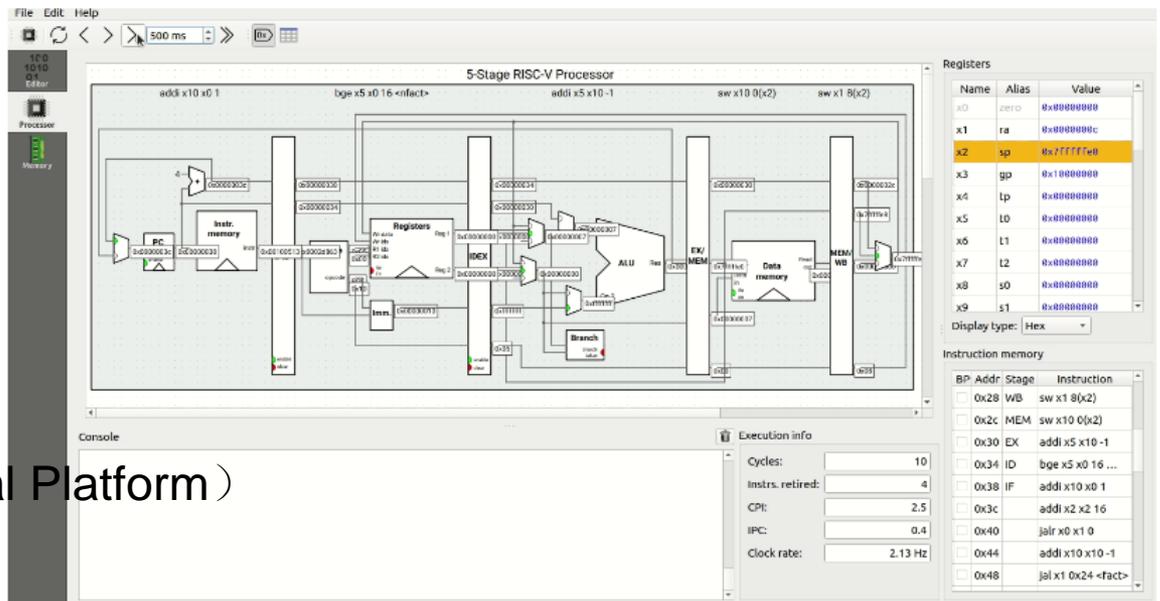
- **【必选】** 汇编语言程序设计：任意10个数冒泡排序
 - 基于RISC-V仿真器Ripes/Spike
 - 成果要求：
 - 报告设计过程、结果（汇编代码、内存数据段映像）、出现的典型问题及解决过程；
 - 要求结果体现**个人ID**：10个数中含**姓名拼音**字母
- **【任选】** 基于Xilinx ISE，设计一个TinyComputer
 - 系统硬件组成：包含CPU、on_chip_ram和JTAG UART三个模块；
 - 编写并运行一个简单C程序“hello_world_small”；
 - 成果要求：
 - 报告设计过程、结果（原理图、仿真结果、源代码）、出现的典型问题及解决过程；
 - 要求结果体现**个人ID**：“hello_world_small—**llxx**”

Ripes



RV32/64仿真器和汇编器
支持五级ppl
支持Cache?
支持MMIO

Spike (linux)
RARS, emulsiv
GEM5, RISC-V VP (Virtual Platform)
SMPCACHE, Cacti





调研/课内实验：小论文格式

- 题目（宋体，小二，加粗）
- 姓名、学号（宋体，小四）
- 摘要（宋体，小四）
- 引言（宋体，小四）
- 正文（宋体，小四）
 - 问题，过程，结果
- 结论（宋体，小四）
- 参考文献（宋体，小五）
（不少于3篇，中英文皆可）



实验课：1学分

- Xilinx ISE：逻辑仿真

- Digilent Nexys3平台

- Spartan-6 FPGA

- 48Mb片外存储器：含2个非易失相变存储器

- 100 ~ 200MHz 振荡器（差分）

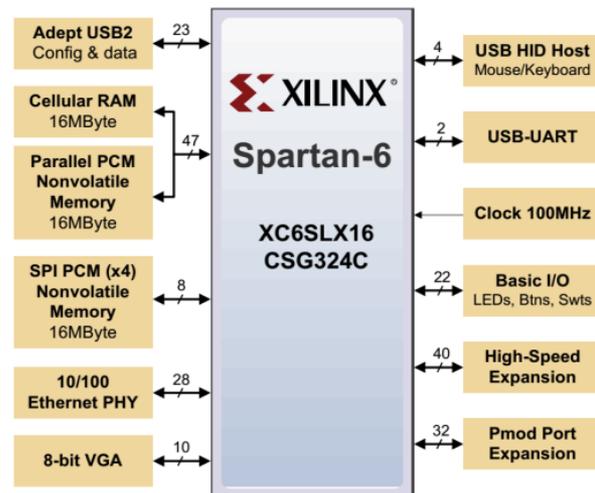


- RV32 ISA子集？

- 约40条指令，VerilogHDL

- 单周期/多周期实现（必做）

- 流水线实现（选作）



实验内容与计划：第5-15周【？】



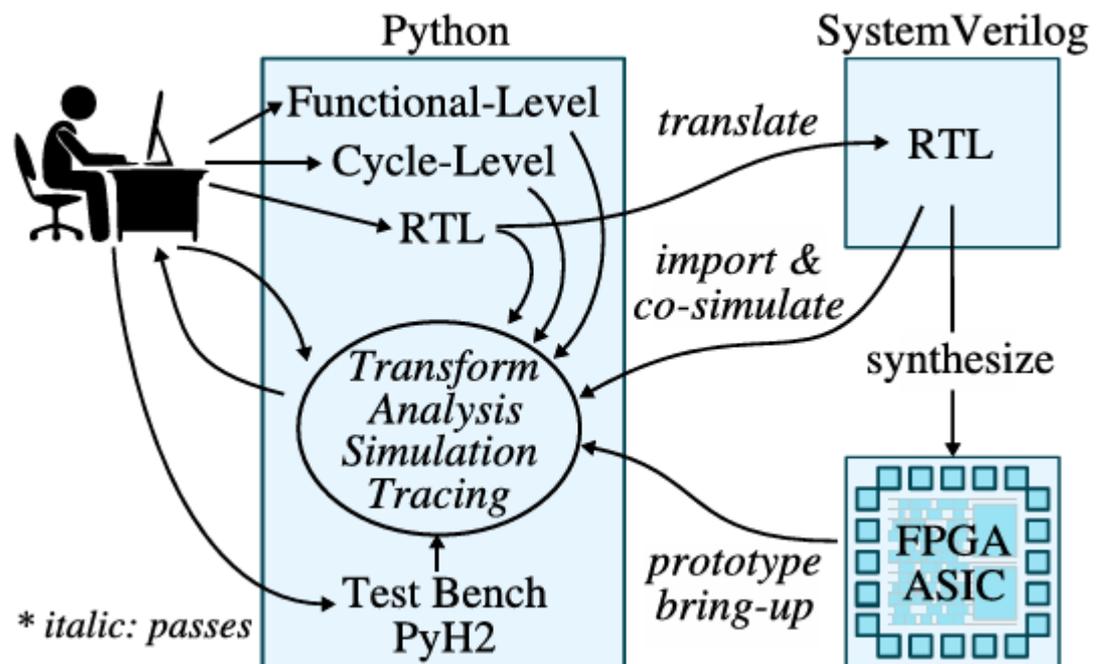
- 运算与排序（1周）
- 寄存器堆与队列（1周）
- 单周期CPU（1周）
- 多周期CPU（2周）
- 中断与异常（1周）
- 总线与I/O接口（1周）
- 综合设计（3周）



PyMTL3 Overview, 康奈尔大学2014



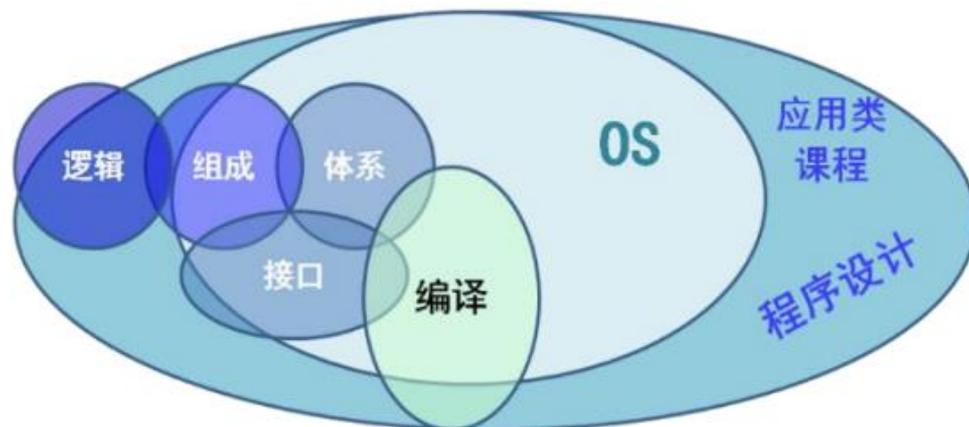
- A Python Framework
 - for Open-Source Hardware Modeling, Generation, Simulation, and Verification
- 代替SystemC



大纲



- 课程简介
- 教材选择
- 教学计划
- 成绩考核
- 学习方法
 - *Learn to do*
 - *Learn to be*
 - *Learn to be with others*
 - *Learn to learn*



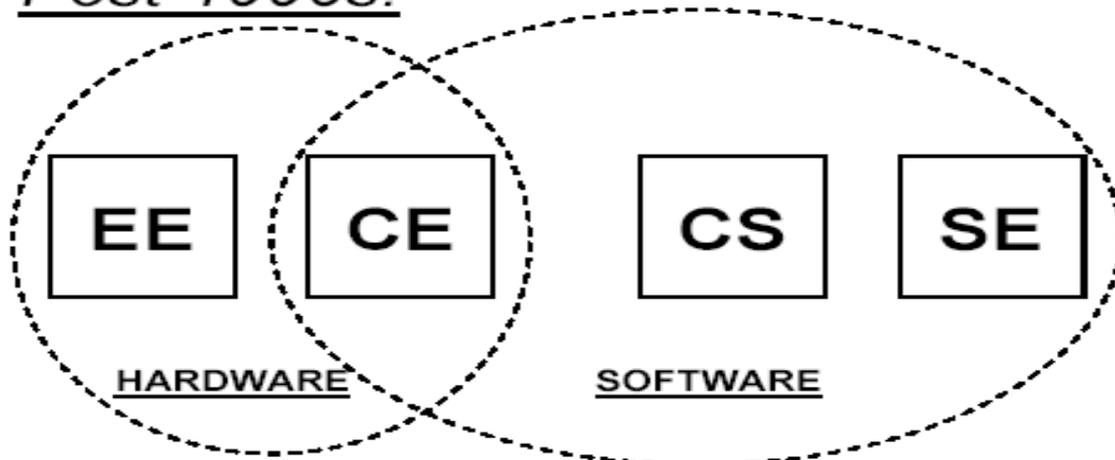
课程知识范畴：科学、技术、工程？



Pre-1990s:



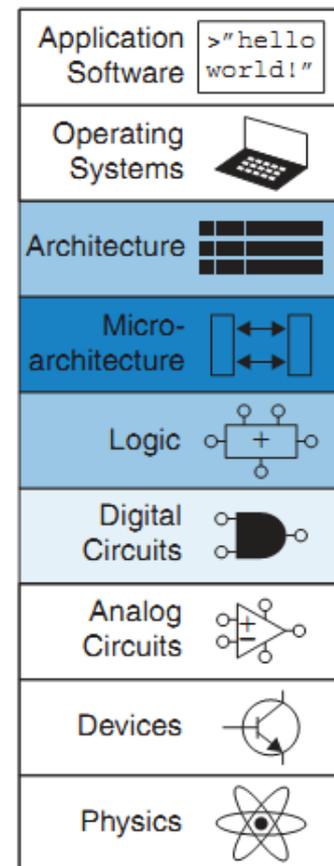
Post-1990s:



课程标准： CS2013/2023, CE2016



- *IEEE/ACM Computer Science Curriculum 2013*
 - ARKA含8个KU(5个必修, 3个选修)
 - 数字逻辑与系统
 - 数据表示
 - 汇编级机器组成
 - 存储系统组成与结构
 - 接口与通信
 - 功能组成
 - 多处理及结构
 - 性能优化
 - CE2016增加
 - “标准与工具”
 - “分布式系统结构, 并行与分布式算法”
- 两个层次
 - system architecture
 - the organization of the computer **at the cache and bus level**
 - organization/micro-architecture
 - the internal implementation of a computer **at the register and functional unit level**



计算机组成 vs. 计算机体系结构



- 计算机体系结构(Computer architecture)

- “程序员可见的机器属性”，ISA

- **G. M. Amdahl, 1964**

- 程序员：特指机器语言程序员

- 属性：包含结构特征和功能特征

- 对物理机器的抽象

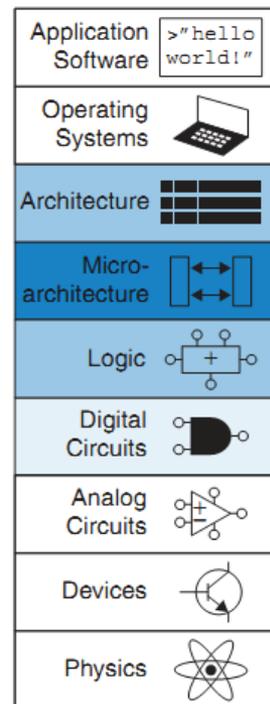
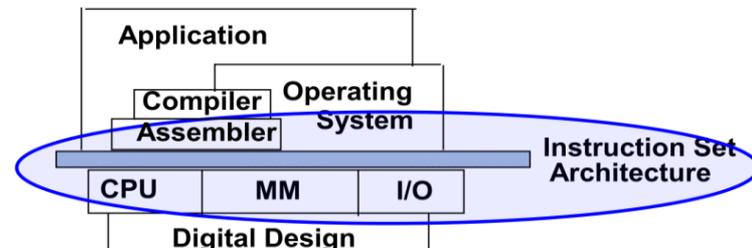
- 计算机组成(org & impl, 也称 μ -architecture)

- 计算机体系结构的逻辑实现，包括

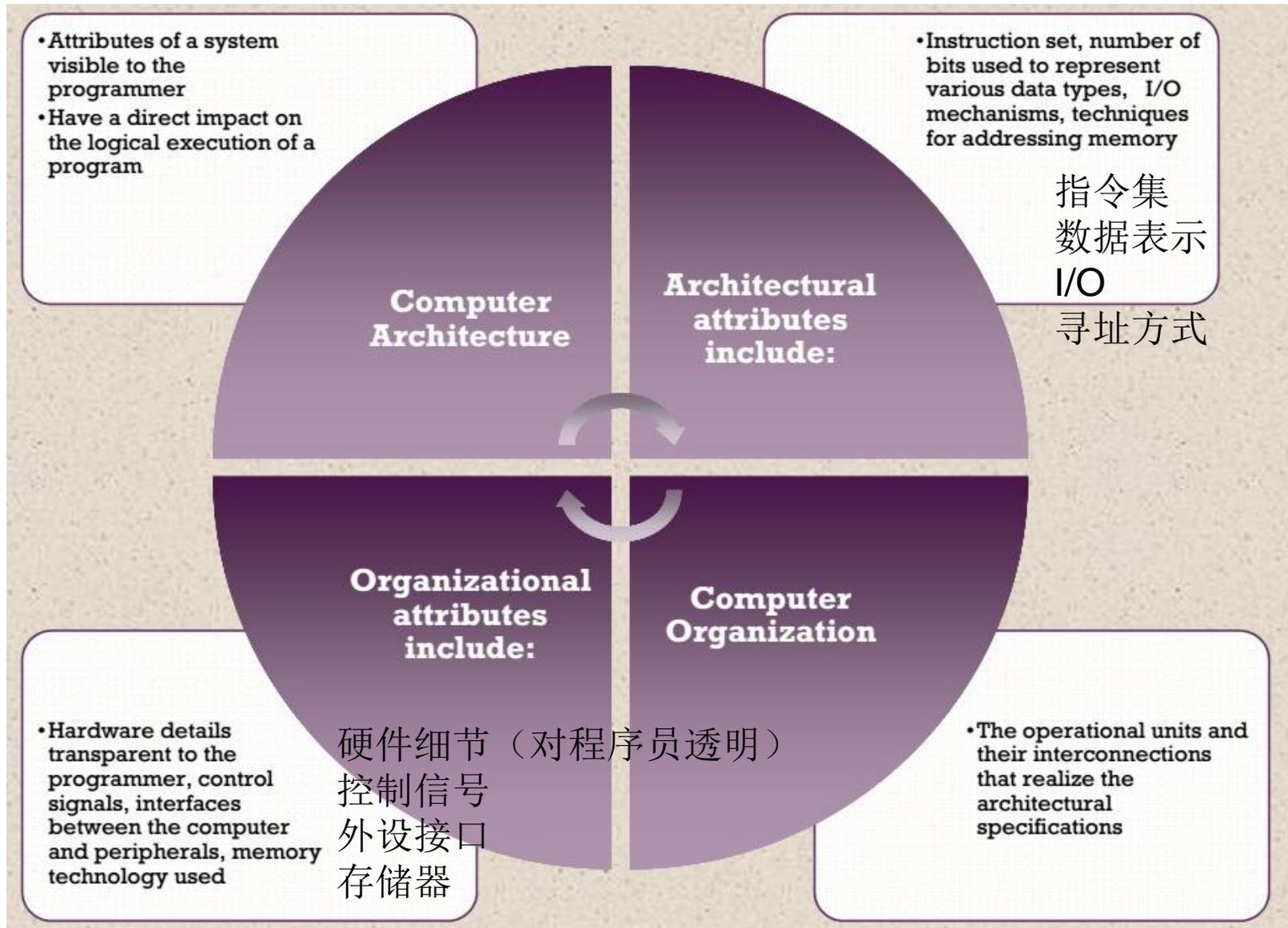
- 实际机器的数据流和控制流的组成和逻辑设计等。

- 物理机器内部各事件的排序方式与控制方式

- 各部件的功能，以及各部件的联系。

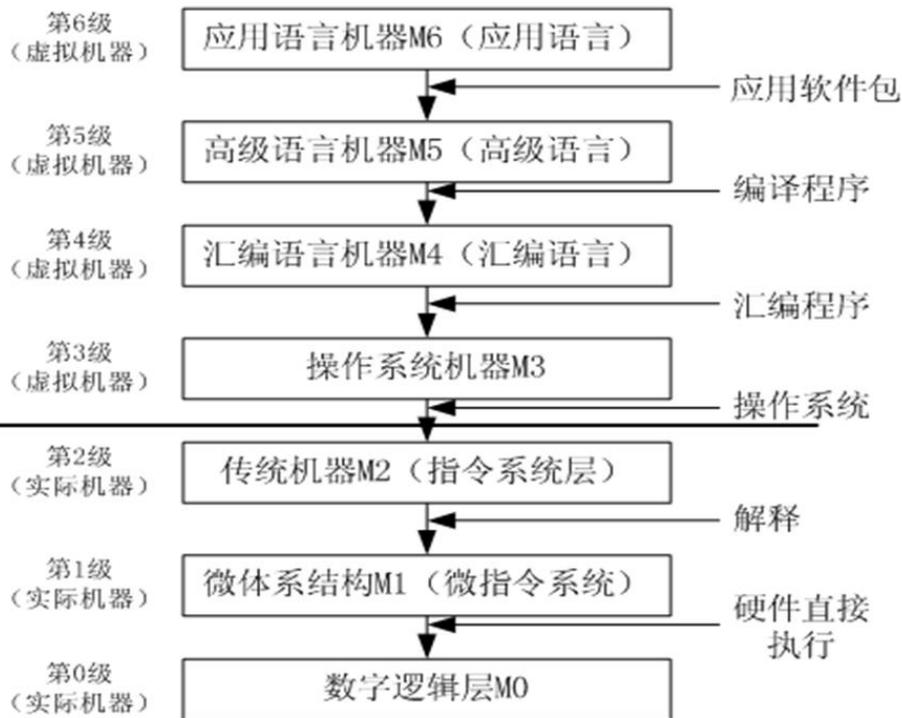


Computer Architecture & Organization

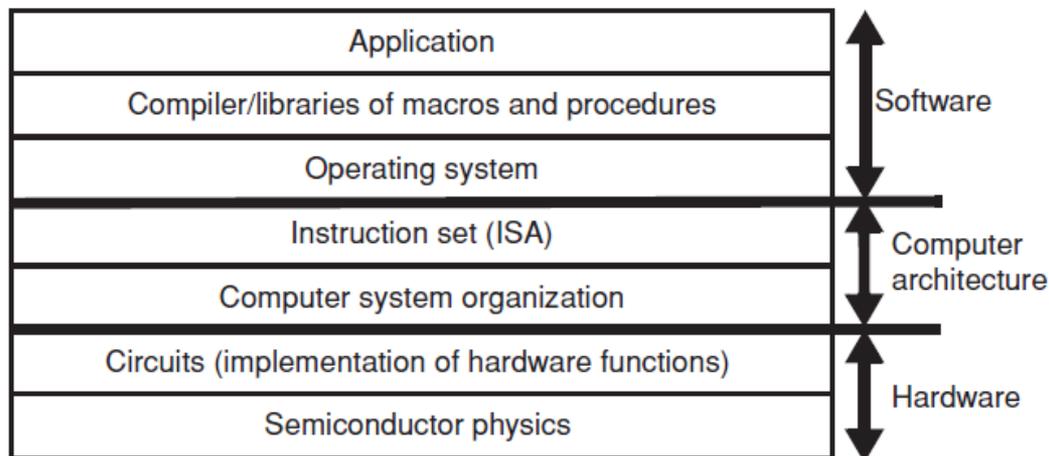
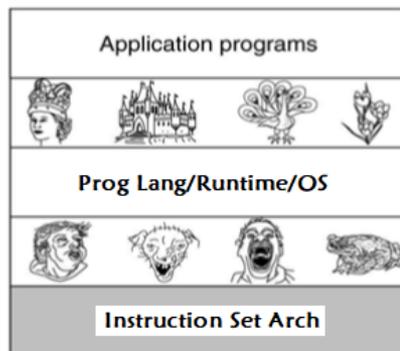




计算机系统的抽象层次：C语言计算机？



计算机系统抽象层的转换



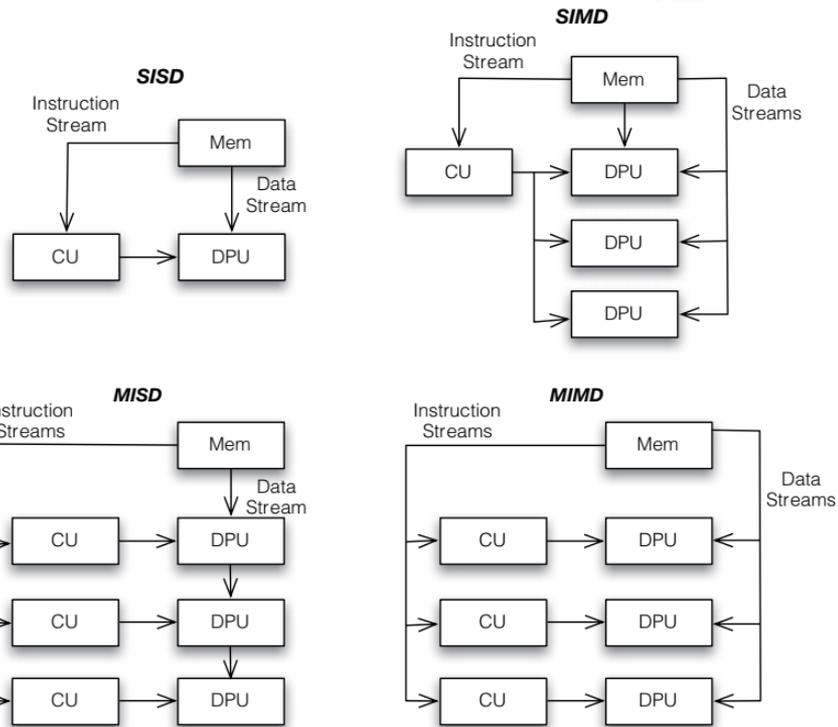
体系结构分类

- 按指令集特征

- 复杂指令集 (CISC)
- 精简指令集 (RISC)
 - 超标量 (Superscalar)
 - 超流水线 (Superpipeline)
 - 超长指令字 (VLIW)

- 按并行执行特征: Flynn分类法

- 单指令流单数据流 (SISD): Von Neumann Machine
- 单指令流多数据流 (SIMD): 向量机, GPU
- 多指令流单数据流 (MISD): 容错
- 多指令流多数据流 (MIMD): 多核, 多处理器



汤孟岩《并行多核体系结构基础》图1-6

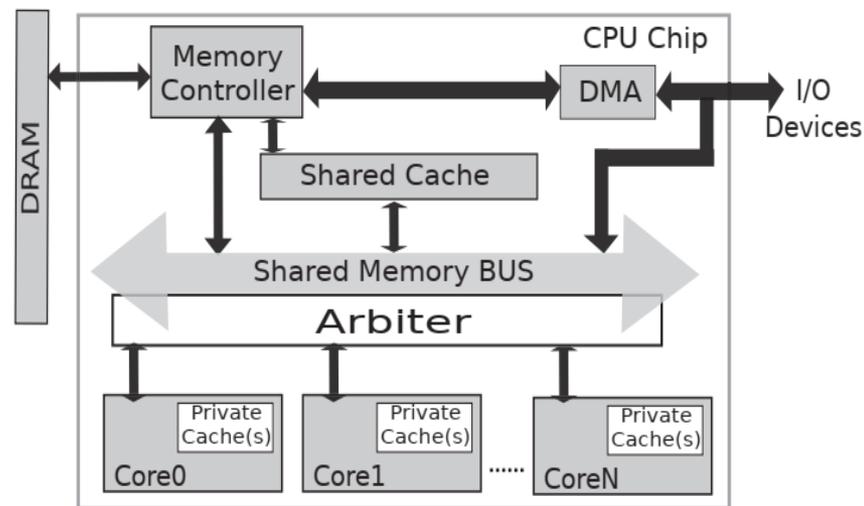
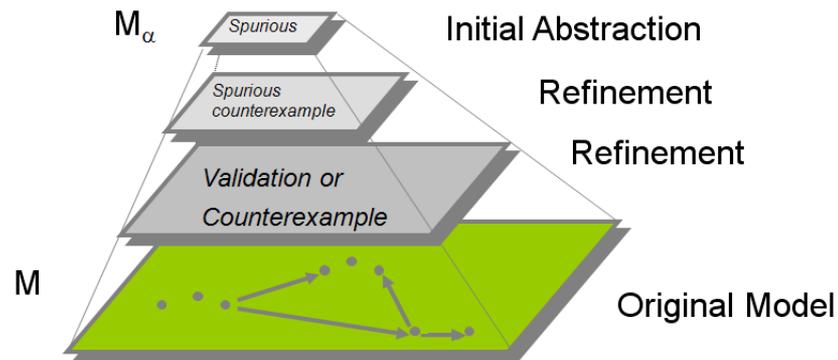
\$6.3图6-2

		Data Streams	
		Single	Multiple
Instruction Streams	Single	SISD: Intel Pentium 4	SIMD: SSE instructions of x86
	Multiple	MISD: <u>No examples today</u>	MIMD: Intel Core i7

体系结构的“7个伟大思想”，\$1.2

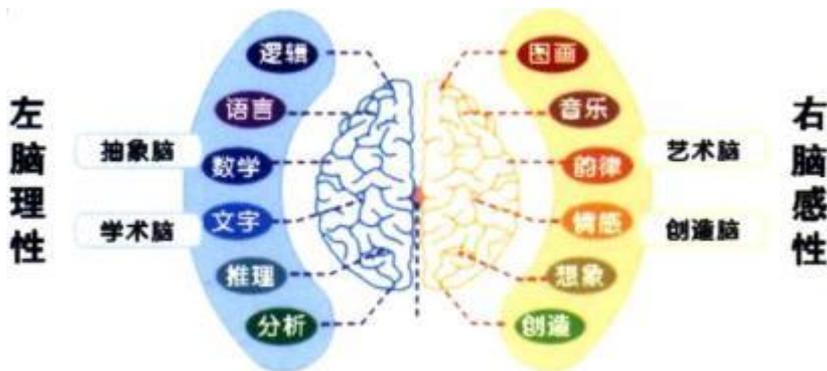


- 使用**抽象**简化设计
 - 如：软硬件接口ISA
- 加速**经常性**事件
 - the common case
- 通过**并行**提高性能（粗粒度）
 - 执行时长、吞吐率
- 通过**流水线**提高性能（细粒度）
 - 吞吐率
- 通过**预测（投机）**提高性能
 - 执行时长
- 存储系统**层次化**
 - 执行时长
- 通过**冗余**提高性能
 - 可靠性



计算思维(Computational Thinking)概念

- 周以真 (Jeannette M. Wing) @CS.CMU, 2006
 - “计算思维是运用**计算机科学**的**基础概念**去**求解**问题、**设计**系统和**理解**人类行为的涵盖了计算机科学之广度的一系列思维活动”
 - “计算思维与数学思维和艺术思维不同”
- **计算**: 机器组织? MoC? 程序的语义? 编程模型?
 - Tom Stuart, 《Understanding Computation》
- **语言**影响**思维**: “我使用的Algol语言塑造了我的计算思维”
 - Jon Bentley, 算法十大师之一, 《编程珠玑 (续)》



COD思维（“系统思维”？）



• COD思维三要素【Ilxx】

– 复杂性：“separation of concerns”

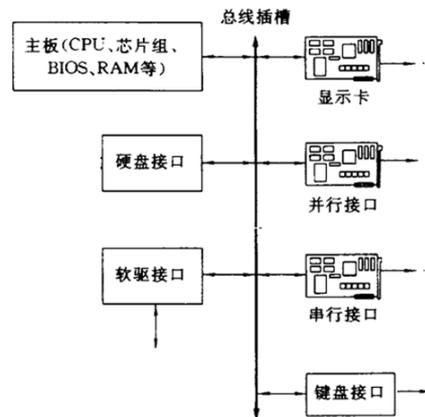
- 层次化：Abstraction/Refinement
- 模块化：Component Based Design
 - The **block** diagram way of **thinking**

– **ABC范式**：架构，行为，约束

- COD语言：语法、语义、语用？
 - 可视化：框图、原理图，CFG/DFG、时序图
 - 形式化：LTS、Automata、FSM，时态逻辑

• “计算机在干什么？”

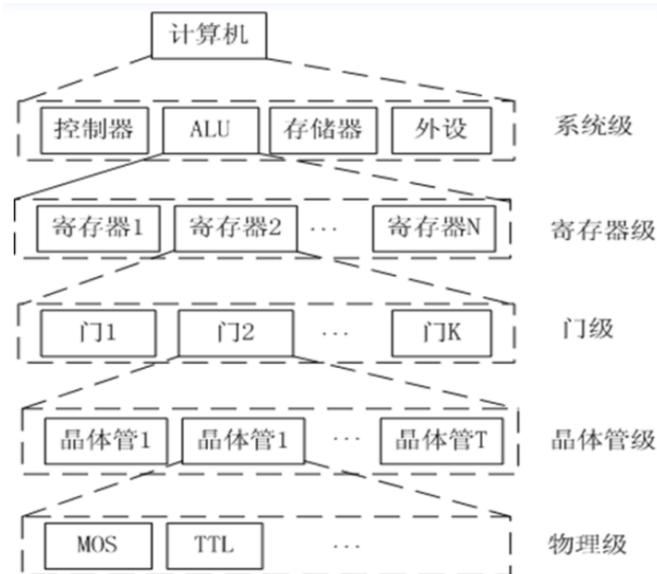
– 由哪些东东组成，计算的过程？



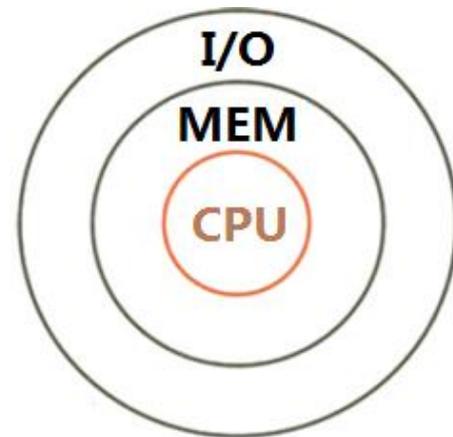
学习思路



- Yale Patt@ece.utexas
 - Top-down: 适于“设计”——精化
 - 从高级语言开始入手
 - 前提是对底层有很好的理解
 - Bottom-up: 适于“学习”
 - 《计算机系统概论》
 - 从最底层的MOS晶体管开关器件开始，逻辑门、锁存器、逻辑结构；
 - 之后，有限状态机控制、顺序电路的实现、冯·诺伊曼结构、简单的计算机LC-3；
 - 最后，LC-3的机器和汇编语言、C高级语言、递归等，最后和基本数据结构。
- 唐（William Stallings）：自外而内
 - 总线、存储、I/O、ALU、ISA、CPU
- COD: 自内而外
 - ALU、ISA、CPU、存储、I/O



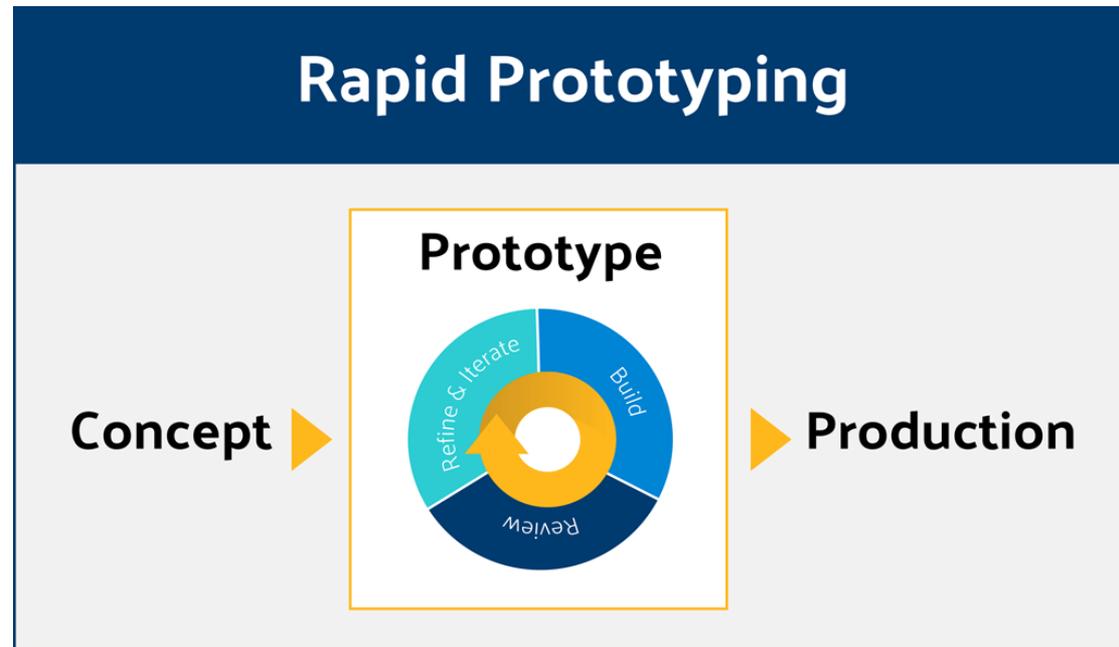
计算机设计者看到的计算机硬件系统



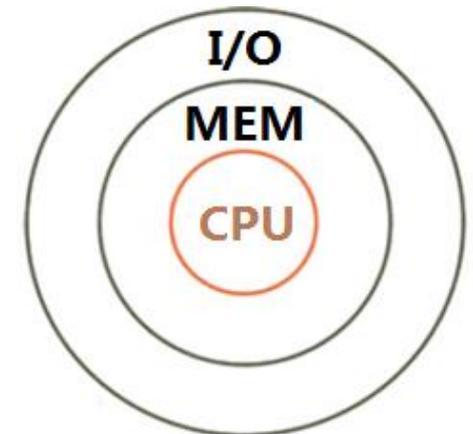
本课方法：最小计算机快速原型法



- 快速原型法
 - Build
 - Review
 - Refine & Iterate
- 工程化方法



- 最小的计算机原型=?
 - ISA, CPU, SRAM, INTR;
 - DRAM, Cache, Disk, VM;
 - Bus, I/O;
 - ALU





逻辑电路设计惯例 (convention)

- 设计表示 (ABC) : 语法, 语义

- A: Block diagram

- B: 逻辑行为, 时序行为

- State Machine: FSM

- Timing Diagram

- C: Timing Diagram参数

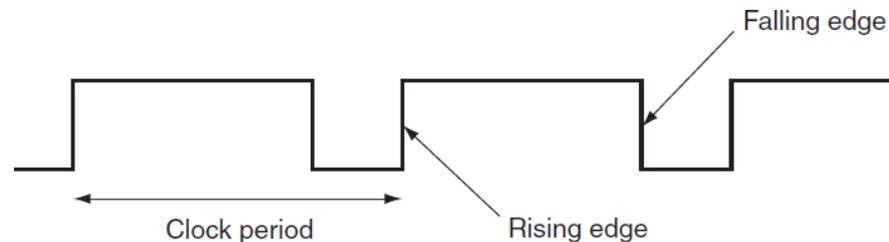
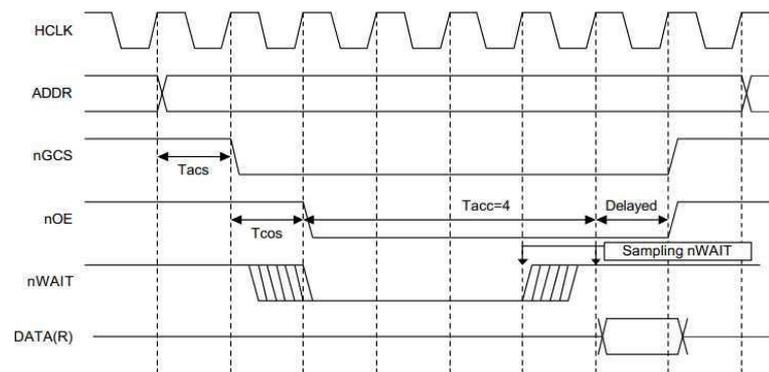
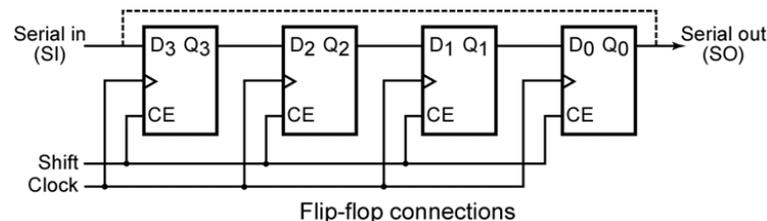
- Clocking methodology

- edge-triggered

- level-triggered

- 时钟频率?

- 常用逻辑单元

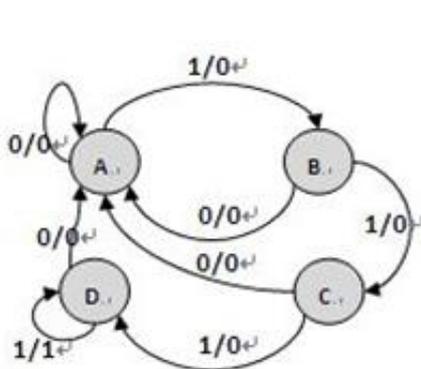
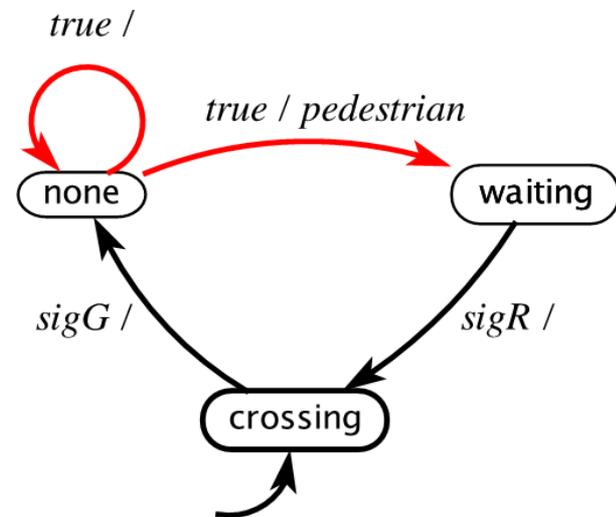


COD5\$4.2, A.7, A.8, A.10, A.11

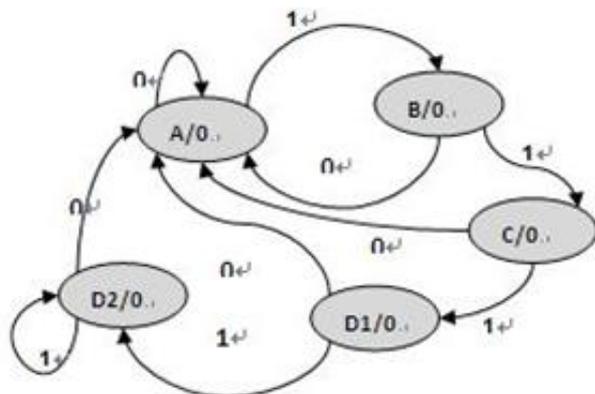


DFA (确定有限自动机) : State Machine

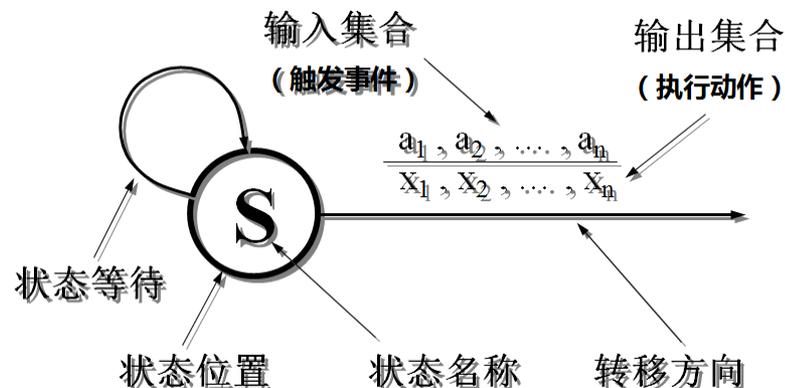
- 强调 “Sequential Machines”
 - 多steps (多时钟周期)
- FSM: 6元组, 带输出
 - Mealy型 (George Mealy, 1955)
 - $\langle S, I, O, f: S \times I \rightarrow S, h: S \times I \rightarrow O, q_0 \rangle$
 - Moore型 (Edward Moore, 1956)
 - $\langle S, I, O, f: S \times I \rightarrow S, h: S \rightarrow O, q_0 \rangle$
 - 完全因果: “one-step delay”



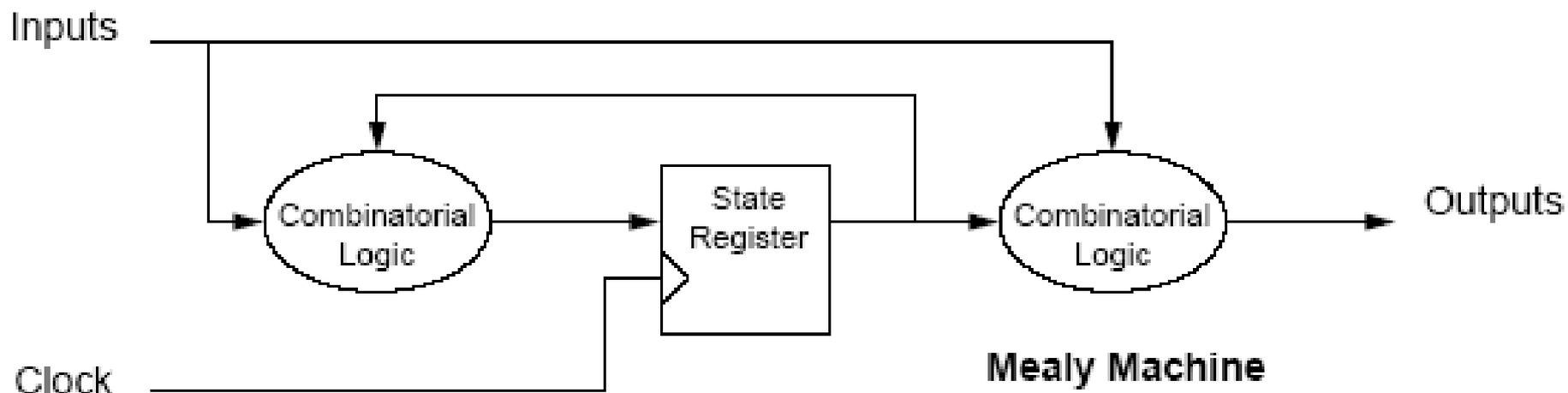
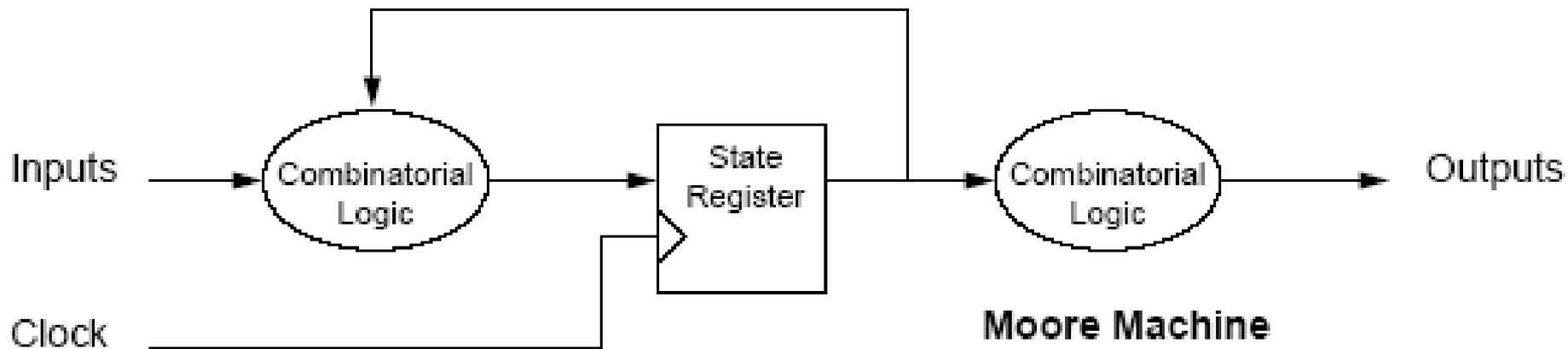
Mealy 型状态



Moore 型状态



FSM的实现结构：同步状态机实现



刻苦 刻苦 刻苦： MIT



- 在这里的第一学期上了一门叫《计算机系统设计》得本科课程，其**辛苦**程度真实一言难尽。
- 十五个星期内交了十次作业，作了六次课程设计。有的设计还分几个部分，分开交设计报告。所以设计报告大概也交了有十次左右。
- 最恐怖的是有一次，十天内要交六份作业或设计报告，而且当时正值其他几门课正在期中考试。**抱怨是没有用的**，老师说：“我很抱歉。但这门课很重要，请大家**不停的工作**”。
- 从一般的逻辑时序电路开始设计（数电都已忘得差不多了），核心是自行设计“麻雀虽小五脏俱全”得ALU，单指令周期CPU（**single cycle CPU**），多指令CPU（**Multi-cycle CPU**），以直到最后实现流水线（**pipe line**）32位MIPS CPU和Cache。
- 一门课下来，所有与计算机**CPU**有关的知识全部**融会贯通**。硬件设计水平也有了**很大提高**（就是太累）。



相关课程

- 美国UC Berkeley大学 “Machine Structure”2012年课程网站：
<http://inst.eecs.berkeley.edu/~cs61c/sp12/>
- 美国UC Berkeley大学 “Components and Design Techniques for Digital System”2012年课程网站：
<http://inst.eecs.berkeley.edu/~cs150/sp12/>
- 美国UC Berkeley大学 “Computer Architecture and Engineering”2012课程网站：
<http://inst.eecs.berkeley.edu/~cs152/sp12/>
- 美国Stanford大学 “Computer Organization and Systems” 2012年课程网站：
<https://courseware.stanford.edu/pg/courses/281000/cs107-spring-2012>
- 美国Stanford大学 “Digital Systems II”课程网站：
<http://www.stanford.edu/class/ee108b/>
- 美国Stanford大学 “Digital Systems II”课程网站：
<http://www.stanford.edu/class/cs110/>
- 美国Carnegie Mellon 大学 “Introduction to Computer Architecture”课程网站：
<http://www.cs.cmu.edu/~213/>
- 美国Carnegie Mellon 大学 “Introduction to Computer Architecture”课程网站：
<http://www.ece.cmu.edu/~ece447/>
- 美国Univ. Illinois at Urbana-Champaign “Computer Architecture II”课程网站：
<http://www.cs.uiuc.edu/class/sp11/cs232/>
- 美国麻省理工学院(MIT)“Computation Structures”课程网站：
<http://6004.csail.mit.edu>

小结

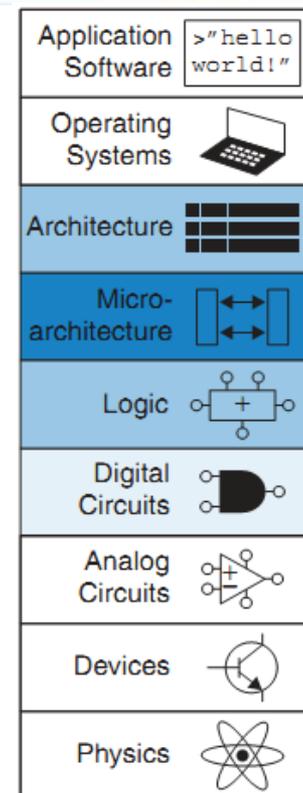


- 核心概念

- 接口、系统思维、快速原型
- 计算机组成 vs. 计算机体系结构
 - 计算机系统的层次化结构

- 挑战

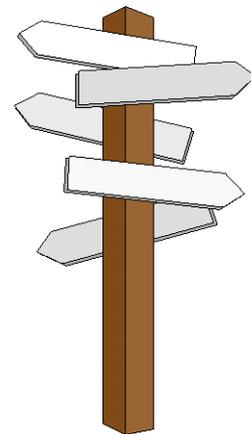
- 数电
- 概念（名词）不一致
 - 缺严谨定义：自然语言而非形式化，或示例说明
 - 习惯差异：EE/CS，SW/HW，中译
- 内容范围广：全系统，多层次



调研报告一



- 选1，下周5交，参考“小论文”格式
 - 综述当前计算机热点问题
 - 对比最新**龙芯/Intel/ARM**处理器的性能指标？
 - 分析计算机的开机过程？
 - C语言背后的计算机组成结构？
 - `getchar()`函数的实现过程？
 - PC系统活动与性能分析
 - 跑一个**Benchmark**，给出结果？
 - 每次执行同一个程序，结果都一样？
 - 主频与计算性能的关系？
 - 分析一款**RV**指令集模拟器和系统级仿真器





Thank You