中国科学技术大学
University of Science and Technology of China
信息科学技术学院

第2章　计算机系统的基本结构与工作原理

中国科学技术大学
University of Science and Technology of China
信息科学技术学院

本章内容

2.1 计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

2.1 计算机系统的基本结构与组成

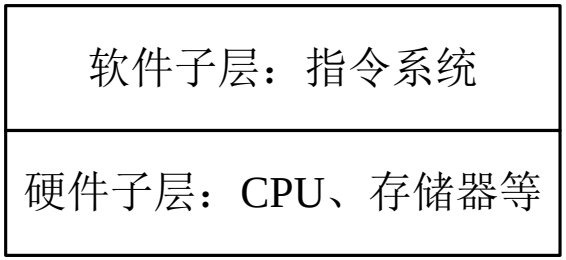
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 计算机体系结构（computer architecture）
  - 设计师眼中：计算机的基本设计思想和由此产生的逻辑结构
  - 程序员眼中：系统功能描述（指令集和编程方法)
- 计算机组成（computer organization）
  - 关注的是机器中各个功能单元的逻辑设计、物理实现和部件之间的互联组织
- 本节两个主题
  - 计算机的层次模型
  - 基于冯诺依曼架构的模型机系统结构

计算机的层次模型

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 分层目的：分析计算机各个部件之间的逻辑关系
- 在计算机的发展过程中，出现过四种层次模型
- 最初阶段：只有两层
  - 硬件层：逻辑电路
  - 软件层：指令系统



- 第二阶段：微程序设计 vs. RISC
  - 微程序设计思想 → 三层模型
    - 一条指令可以分解为多个微操作
    - 微操作可以用微指令实现
    - 多条微指令组成微程序实现指令功能
    - 微程序存储在ROM中，执行时逐条读出完成微操作

微程序

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 微程序简化了控制器硬件，并可实现复杂指令
- 设计师开始不断增加的新指令，引入新的寻址方式，以期实现更加复杂的操作，指令数开始暴涨！
  - 如：DEC PDP-11（全球第一台16位机）仅有70条指令

    下一代产品VAX 11/780，330条指令和18种寻址方式
- 更多、更复杂的指令和寻址方式导致了
  - 芯片中的器件数量增加
  - 芯片功耗不断增大
  - 但计算机性能与电路规模并不成比例

微机房

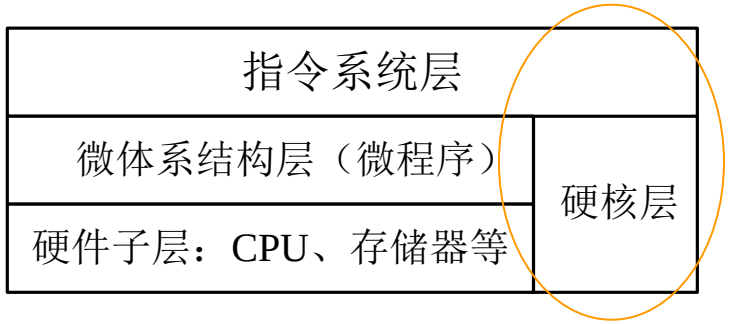
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

RISC - Reduced Instruction Set Computing

- 1975年，IBM科学家John Cocke对IBM370的性能进行了分析，发现了计算机中的“二八定律”
  - 80%的时间运行的是占总量不到20%的简单指令
  - 80%的任务是由占总量不到20%的电路完成的
- John Cocke指出，为了提高性能，应该：

  减少指令数量，一条复杂指令用多条简单指令替代

  取消微程序，指令功能由硬件电路（硬核）实现
  - 按照上述思想的计算机仍只有两层



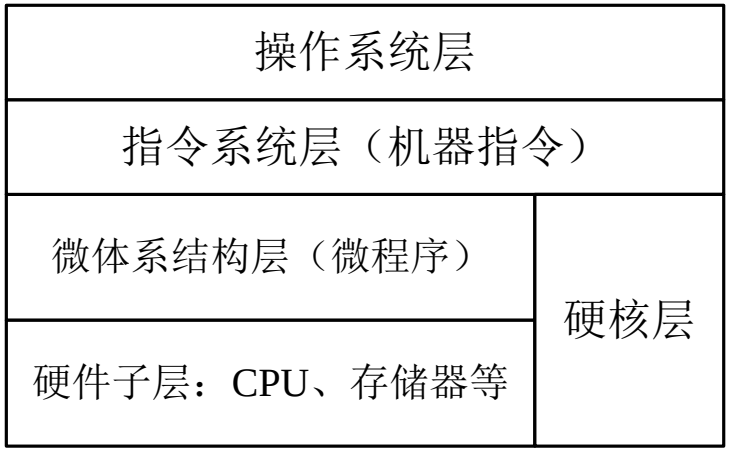
时任Berkeley大学教授的Patterson为这类计算机取名为RISC，与之相对的则被称作CISC

操作系统（Operation System）

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 早期计算机没有操作系统，必须“手工”对计算机进行管理，如任务调度、内存管理、I/O控制等
- 1964年，可运行在不同规格的IBM System/360系列大型机中的OS/360，被认为第一个真正的操作系统

  （诞生于1956年的GM-NAA I/O只能算作“批处理程序”）
- 操作系统负责管理计算机硬件资源和用户作业，提供了人机交互界面、多条用户命令和多种子程序调用接口，极大简化了计算机操作、管理的复杂性

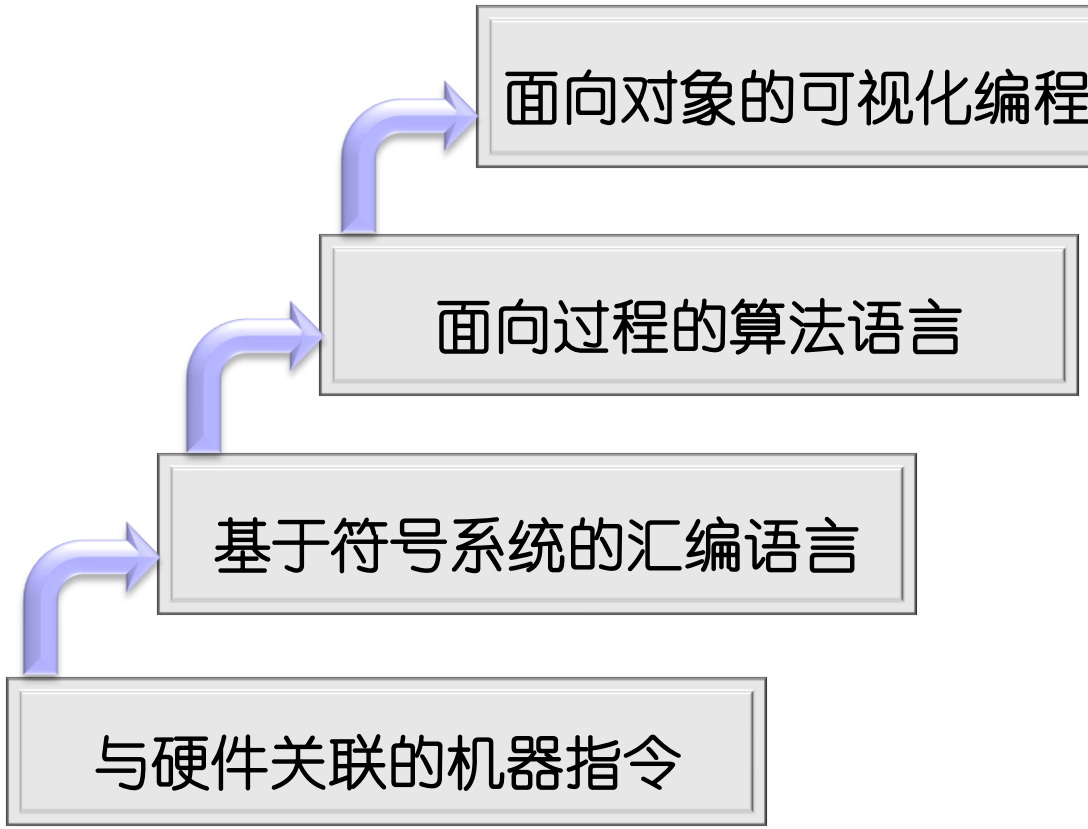


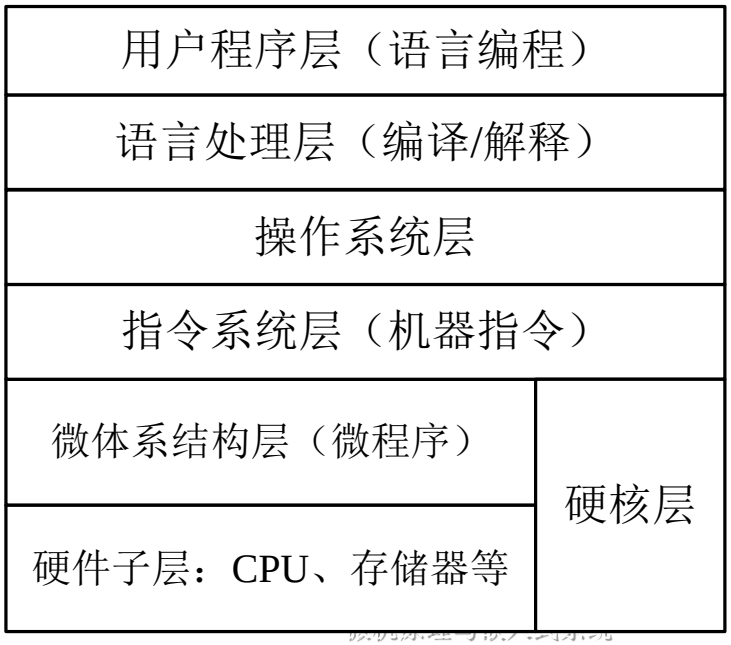
增加操作系统后的层次模型

编程语言

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 使用各种语言编写的程序，必须经过相应的编译（或解释）程序进行处理后，计算机才能识别和执行

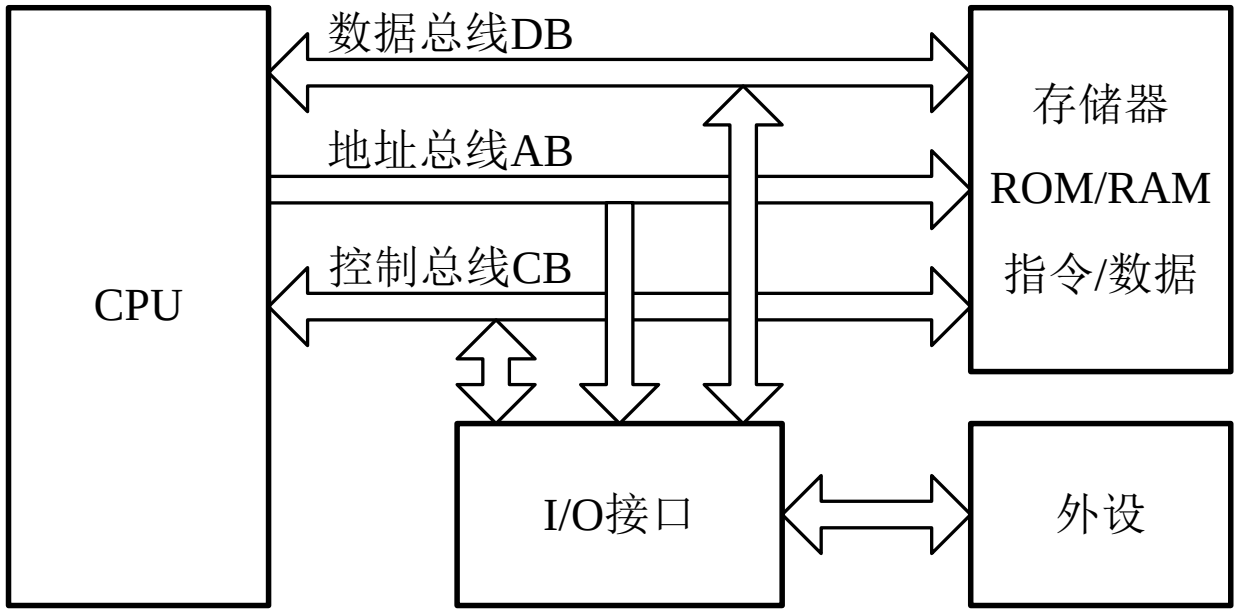




基于冯·诺依曼架构的模型机系统

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 模型机系统结构特点：
  - 以CPU为核心（现代计算机逐步转化为以存储器为核心）
  - 单总线系统（类似于快慢车道不分的混合式交通）
  - 指令和数据使用同一条总线（冯·诺依曼架构的主要缺陷）



本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

- 存储器的组织和地址
- 字的对齐-对准存放
- 小端格式和大端格式
- 存储器操作
- 存储器的分级

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

- 存储器的组织和地址
- 字的对齐-对准存放
- 小端格式和大端格式
- 存储器操作
- 存储器的分级

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

存储器的组织和地址

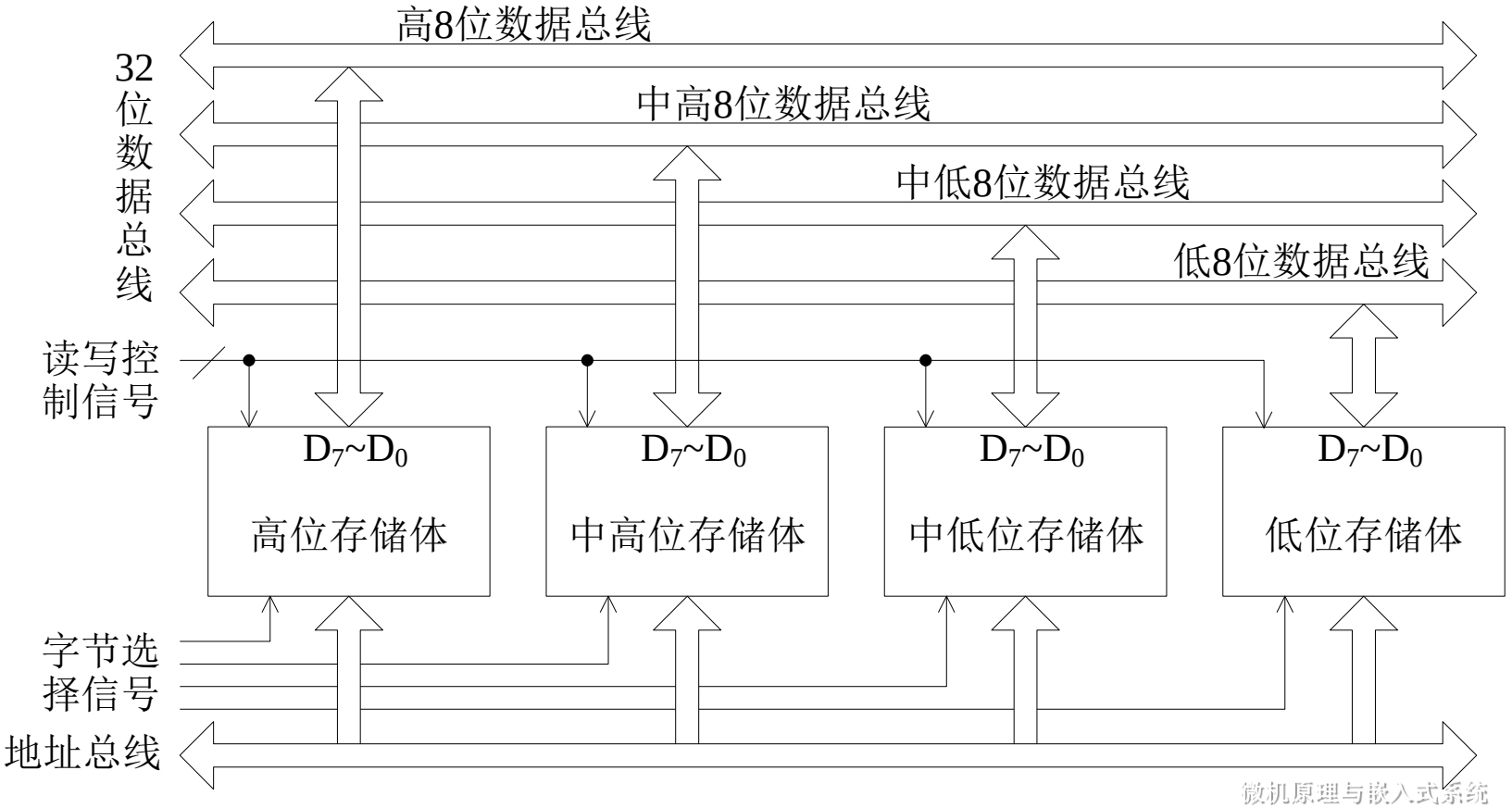
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 每个字节拥有一个独一无二的物理地址（PA，Physical Address），字节是计算机可访问的最小存储单元
- 字节寻址存储器：按照字节组织存储器，连续地址对应于连续的字节单元。例如：32位计算机中，一个字有4个字节，连续的字被分配到*n*、*n*+4、*n*+8 ... ...中
- 总线宽度为8位的计算机，CPU访问存储器时，总线一次可以传送一个字节数据
- 若总线宽度是16位、32位或64位，CPU访问存储器时，希望一次能够传送一个完整的字（2/4/8字节），或者根据需要一次传送这个字中的一部分字节，存储器应该如何组织？存储器与总线应该怎样连接？

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

计算机的存储器分体结构

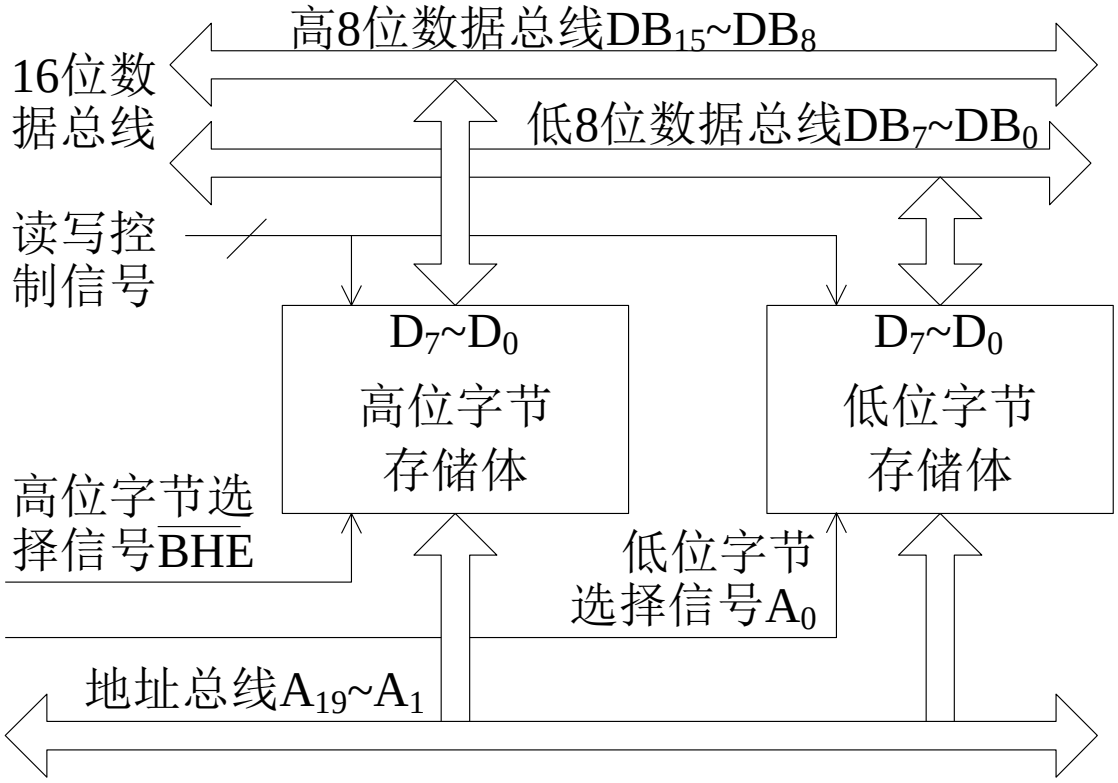
- 对于32位计算机，总容量为2^{32}的存储器分成4个存储体，每个存储体为2^{30}，分别与32位数据总线按下图方式连接；每个存储体只需30条地址线，用字节选择信号进行选择



示例：Intel 8086存储器分体结构

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- Intel 8086数据总线位宽为16位，地址总线位宽为20位
  - 总容量为1 MB的存储系统分成2个512KB的存储体
  - 高位和低位字节存储体分别连接DB的高8位和低8位



- $\overline{BHE}$ 和A_0为存储体选择信号
- $\overline{BHE}$ 有效（低电平）选中高字节存储体；A_0=0选择低字节存储体；都有效同时选中两个存储体

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

字的对齐——对准存放

- 8位计算机没有对准存放问题
- 对准存放：
  - 16位机的字起始地址应该是2的倍数，如0、2、4...
  - 32位机的字起始地址应该是4的倍数，如0、4、8 …
  - 64位机的字起始地址应该是8的倍数，如0、8、16…
- 对准存放不是必须的，但如果采用对准存放，存取一个字只需要一次总线操作即可完成
- 为此，有些计算机的指令系统中（如Intel x86）专门提供了对准存放指令

收割机作业时需要对准

小端格式和大端格式

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 假设W由B_3、B_2、B_1和B_0 组成，B_3是最高字节，B_0是最低字节，存储W需使用4个地址连续的内存单元
- 对于地址依次为m、m+1、m+2和m+3的连续4个存储单元， m单元地址最小，称为尾部（Endian）；m+3单元地址最大，称为头部

m	尾部
m+1	
m+2	
m+3	头部

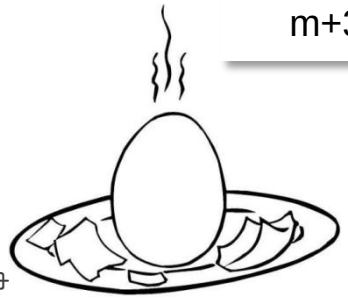
- W有两种存放格式：
  - Intel x86采用小尾或小端格式
  - Motorola采用大尾或大端格式
- 如果采用对准存放，m是整个字的地址

m	B_0
m+1	B_1
m+2	B_2
m+3	B_3

小端格式
高位高地址
低位低地址

m	B_3
m+1	B_2
m+2	B_1
m+3	B_0

大端格式
高位低地址
低位高地址



一场由剥鸡蛋引起的战争

存储器操作

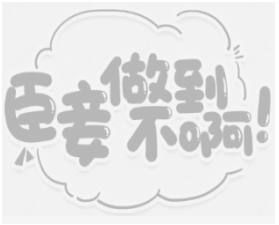
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 两个最基本操作：读出（read）和写入（write）
- 读操作：
  - 将一个指定存储单元的内容读出并传送到CPU中，读操作之后存储单元的内容保持不变
  - 过程：CPU发送地址信号和读命令，被选中单元中的数据被读出到DB上，CPU采样数据并存入内部寄存器
- 写操作
  - CPU向指定单元传送数据，并覆盖目的单元原有内容
  - 过程：CPU通过AB、DB和CB分别发送地址、数据和写命令，DB上的数据被写入到被选中单元
- 连续数据读写：只需在第一次读写时发送地址

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

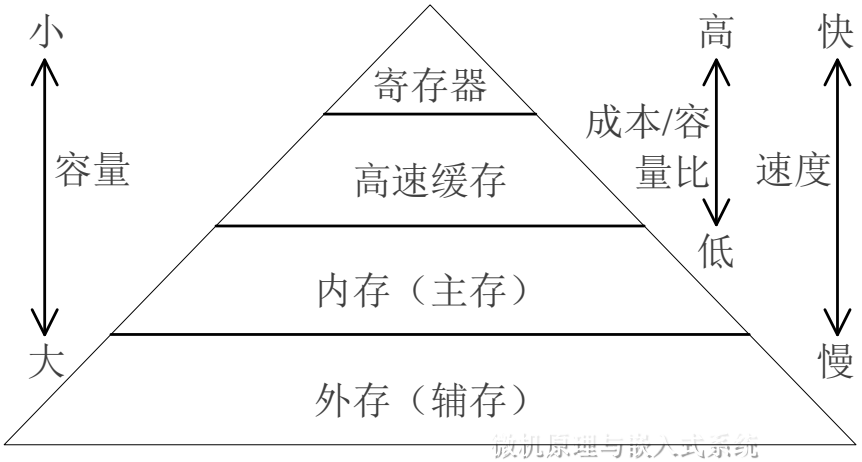
存储器的分级

- 对存储器的要求：速度快、容量大、成本低
- 要同时满足这三个要求“太难了”
- 破局方法：分级存储体系结构
  - 使用外存满足大容量、低成本和非易失的要求
  - 使用DRAM型内存，兼顾容量、速度和成本
  - 使用高速缓存，减少CPU访问内存的开销



高速缓存（Cache）

位于CPU与内存之间，SRAM型小容量快速存储器，用于存放CPU最近使用过或者可能要使用的指令和数据



本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

- 运算器
- 控制器
- 寄存器阵列
- 地址与数据缓冲器
- 数据通道

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

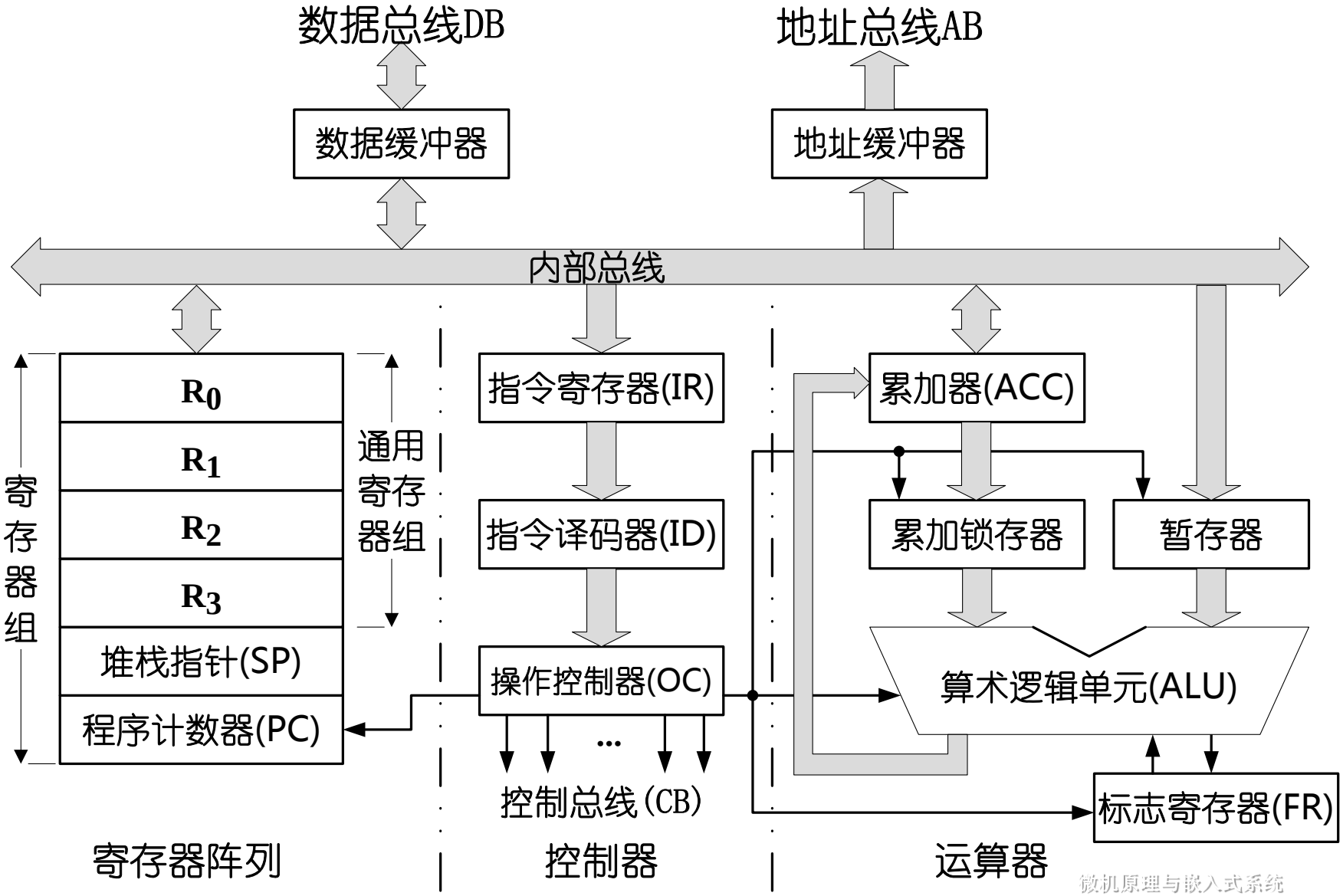
2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

模型机CPU内部结构

中国科学技术大学 University of Science and Technology of China 信息科学技术学院



中国科学技术大学 University of Science and Technology of China 信息科学技术学院

运算器

- 基本组成：
  - 算术逻辑单元ALU（Arithmetic Logical Unit）
  - 累加器ACC（Accumulator）
  - 标志寄存器FR（Flag Register）
  - 暂存寄存器
- 运算器的核心：ALU：
  - 负责运算，也是数据传送的一条重要途径
  - 组成：带有先行进位功能的全加器（简称加法器）、移位寄存器以及相应的控制逻辑
  - 加法器是ALU最主要的部件，所有二进制算术运算都可通过加法和移位来实现

运算器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 累加器ACC：特殊寄存器
  - 提供需要送入ALU的操作数，存储ALU的计算结果
  - 早期的CPU（如Intel公司的8086和Motorola公司的68000）只有一个ACC，因ACC与ALU之间密不可分，身份特殊，常被划归到运算器中，不属于通用寄存器组
  - 现代CPU中有很多通用寄存器（甚至所有通用寄存器）都可以当作累加器来使用，累加器的地位不再特殊，ACC这一称呼也逐渐淡出江湖
  - ACC下方的累加锁存器：其作用是防止ALU的输出经ACC再反馈到ALU的输入端

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

运算器

- 暂存器：
  - 暂时存放需要送入ALU的操作数，但不存放计算结果
  - 暂存器是透明度，程序员不可见
- 标志寄存器：分为状态（条件码）位和控制位
  - 状态位：记录ALU运算后的状态或者特征，如
    - 结果是否为零？是否为负数？是否有溢出？是否有进位？
    - 后续指令可根据状态标志决定程序执行顺序
  - 例如：
    - ARM处理器的程序状态寄存器（PSR）中有4个状态位（条件码）
    - Intel 8086中还有辅助进位位A和奇偶校验位P（P现已不用）

Z → 结果为零
N → 结果为负
V → 出现溢出
C → 发生进位

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

运算器

- 标志寄存器：控制标志位
- 作用：是对CPU的某些行为进行控制和管理，例如：
  - Intel 8086的标志寄存器中有3个控制标志位
    - D（Direction）：串操作的地址改变方向
    - I（Interrupt）：是否允许外部中断
    - T（Trap）：单步中断

D=1 →地址减量
I=1 →允许中断
T=1 →单步中断

  - Intel 8086提供了对D和I进行单独操作的指令，如
    - STI：将I置位为1，允许外部可屏蔽中断，亦称开中断
    - CLI：将I复位为0，禁止外部可屏蔽中断，亦称关中断
  - 此外，Intel 8086还提供了对状态标志位C（Carry）的3条操作指令，操作内容分别是置位、复位和取反

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

控制器

- 整个CPU的指挥控制中心
- 功能和作用：根据指令中的操作码和时序信号，产生各种控制信号，对系统各个部件的工作过程进行控制，指挥和协调整个计算机有序地工作
- 主要构成：
  - 指令寄存器IR（Instruction Register）
  - 指令译码器ID（Instruction Decoder）
  - 操作控制器OC（Operation Controller）
  - 有种观点认为还包括程序计数器PC（Program Counter）
    （另一种观点认为PC应属于数据通道）

控制器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 指令寄存器IR
  - 临时存放从内存或者Cache中取出的下一条待执行指令，其输出作为指令译码器的输入
- 指令译码器ID
  - 计算机能且只能执行“指令”
  - 指令由操作码和地址码两部分构成
    - 操作码表示要执行什么操作
    - 地址码表明指令执行时操作对象（操作数）的存放地址
  - 指令译码器只对操作码进行译码，分析和识别指令应该执行什么样的操作

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

控制器

- 操作控制器
  - 根据指令译码器的译码结果，产生所需的各种控制信号并发送到相关部件，控制这些部件完成规定的操作
  - 操作控制器内部包括时序脉冲发生器、控制信号发生器、启停电路和复位逻辑等
- 程序计数器PC（Intel 8086中称为指令指针IP）
  - 存放下一条待执行指令在内存中的地址
  - 计算机开机时，指向引导程序的第一条指令
  - 顺序执行时，每条指令执行后自动修改，PC=PC+n，n与指令字长以及PC的单位有关
  - 遇到转移指令，转移目标地址→PC

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

控制器

- 微操作
  - 每条指令的执行过程都可以分解为一系列的微操作
  - 例如：
    - 假设指令“ADD R1，R2，R3”的功能为 R2+R3 →R1
    - 可以分解为以下8个微操作

① 取指	⑤ 加法器完成相加
② 指令译码	⑥ 结果→ R1
③ R2→加法器	⑦ 更新状态寄存器
④ R3 →加法器	⑧ PC=PC+n

  - 微操作特点：可由简单电路实现；可被多个指令复用

控制器的实现方式

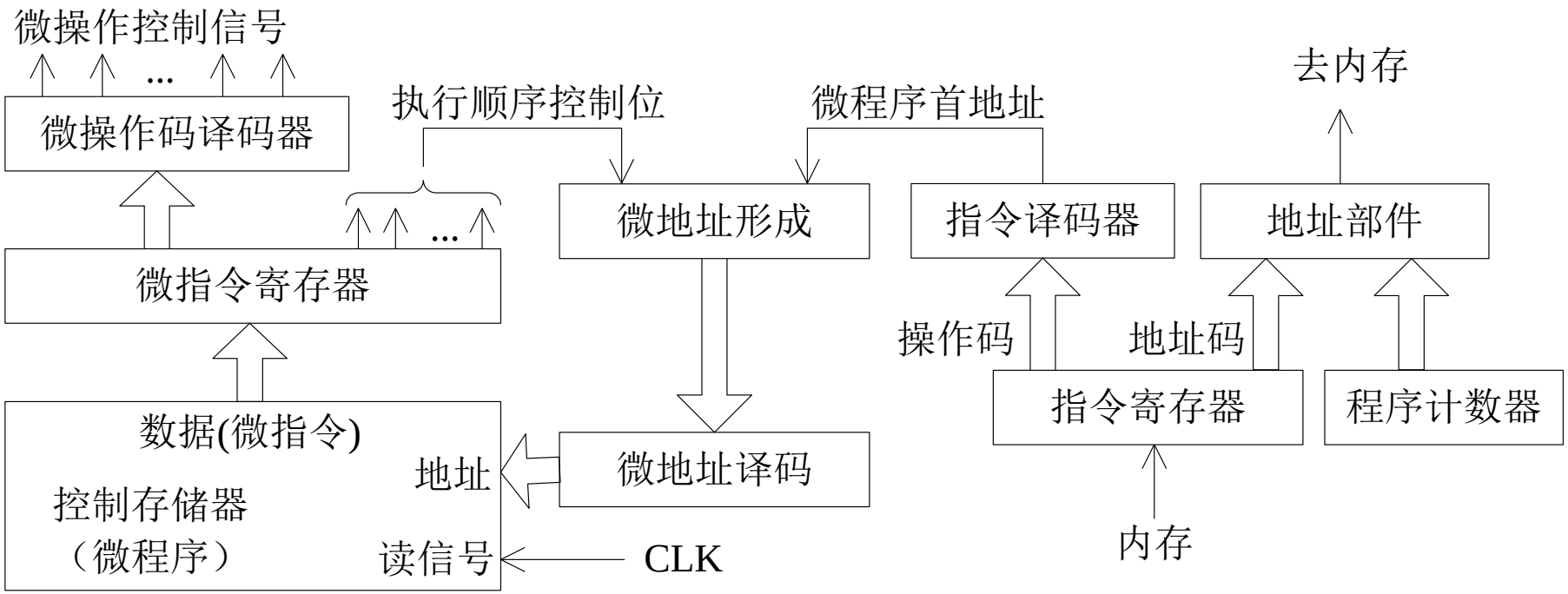
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 两种实现方式：微程序控制器和硬连线控制器
- 微程序控制器
  - 指令执行过程看作多个微操作序贯执行完成的
  - 对每个微操作进行编码，形成微操作码，微操作码可由简单电路产生微操作控制信号
  - 执行顺序控制位：指示后续微操作的执行顺序
  - 微操作码+执行顺序控制位 = 微指令
  - 指令→一段由若干微指令编排而成微程序
  - 所有指令对应的微程序都存放在控制存储器CM中
  - 指令执行时，对应的微指令从CM中逐条读出，其中微操作码经过译码产生微操作控制信号

控制器的实现方式

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 微程序控制器结构
  - 微地址：微指令在CM中的存放地址
  - CLK：作用等于读信号



控制器的实现方式

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 工作原理和过程
  - 计算机指令分为操作码和操作数地址两部分
  - 操作码由指令译码器译码，译码结果是该指令对应的微程序在CM中的首地址（微程序的入口地址）
  - 该地址经微地址译码器译码后，从CM中读出第一条微指令，其中微操作码部分送往微操作码译码器进行译码，生成相应的控制信号以实现规定的微操作。
  - 执行顺序控制位送往微地址形成电路，生成下一条微指令的微地址
  - 不断重复上述过程，直到这段微程序全部执行完毕
- 特点：硬件电路简单，可支持复杂指令，速度慢

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

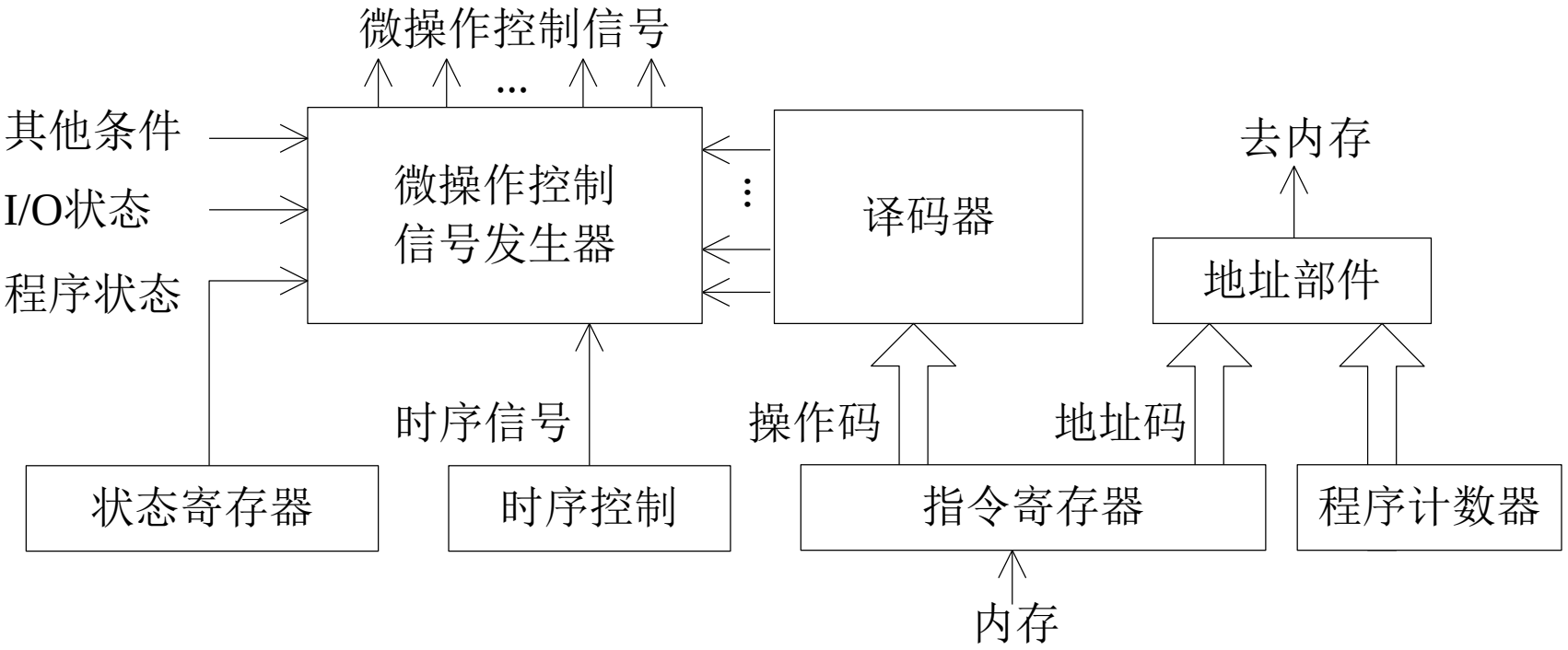
控制器的实现方式

- 硬连线控制器
  - 也称为组合逻辑控制器，最早采用的控制器设计方法
  - 把控制器看作专门产生固定时序的控制信号的逻辑电路，以使用元件少和速度快作为设计目标
  - 因指令功能的多样性和差异性，导致所实现的控制器逻辑电路复杂、规模庞大，并且一旦形成就无法变更，除非重新设计和重新布线
  - 设计步骤：
    - 输出：需要产生的微操作控制信号
    - 输入：微操作信号类型、执行条件和时序等
    - 列出逻辑表达式，经过化简，设计相应的逻辑电路。

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

控制器的实现方式

- 硬连线控制器一般结构



- 特点：速度快，电路复杂，不支持复杂指令，调试和改动困难，一度被微程序取代。近年因RISC的兴起和VLSI的进步，又重新焕发青春，再度兴起

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

寄存器阵列

- 也称为寄存器组、寄存器堆和寄存器文件
- CPU内部的若干高速存储单元，每个都有编号或名称，根据指令中的编号或者名称对其直接访问
- CPU与寄存器之间的数据交换是通过内部总线直接进行的，所以CPU与寄存器之间的数据传送速度最快
- 受指令长度限制，寄存器数量有限，只能暂时存放CPU工作时所需的少量数据和地址
- 分为专用寄存器和通用寄存器两大类
  - 专用寄存器作用固定，如PSR（FR）、IR和PC等
  - 通用寄存器：为ALU运算提供一个存储区，早期数量较少且通用性差，现在数量增加，通用性增强

模型机CPU中其他部件

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 地址和数据缓冲器
  - CPU内部总线与系统总线之间的接口，提供地址和数据传送缓冲，同时增加CPU的系统总线驱动能力
- 数据通道
  - 计算机各部件按功能划分为两大阵营：CU和EU
  - CU就是控制器，负责指令译码，生成相应的控制信号
  - EU负责指令执行，如生成地址、读取和传送数据、计算和处理数据、存储结果、更新PSR和PC
  - 在指令执行过程中，数据是在运算器、寄存器阵列和系统总线接口之间通过内部总线进行传送，所以这几个部件也被称为**数据通道**（data-path）

本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

- 模型机指令集
- 指令周期
- 指令执行流程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

模型机指令集

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 指令：分为微指令、机器指令和宏指令
  - 微 指 令：微程序级的命令
  - 机器指令：简称**指令**。CPU能识别和直接执行的一条二进制编码序列，包括操作码和操作数两部分
  - 宏 指 令：由若干条机器指令组成的软件指令
- 指令系统：一台计算机中所有指令的集合
  - 指令集架构ISA（Instruction Set Architecture）：程序员眼中的计算机体系结构
- 汇编指令：
  - 助记符→操作码，标号和符号→指令和操作数地址

模型机指令集

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

指令类型		操作码	操作数	说明
算术类	加法	ADD	Rd, Rs1, Rs2 Rd, Rs, Imm	(Rs1)＋(Rs2) → Rd (Rs)＋Imm → Rd
	减法	SUB	Rd, Rs1, Rs2 Rd, Rd, Imm	(Rs1)–(Rs2) → Rd (Rs)–Imm → Rd
逻辑类	位与	AND	Rd, Rs1, Rs2 Rd, Rs, Imm	(Rs1) ∧ (Rs2) → Rd (Rs) ∧ Imm → Rd
	位或	OR	Rd, Rs1, Rs2 Rd, Rs, Imm	(Rs1) ∨ (Rs2) → Rd (Rs) ∨ Imm → Rd
	位非	NOR	Rd, Rs	$\overline{(RS)}$ → Rd
传送类	存储器或者I/O读	LDR	Rd, address	[address] → Rd
	存储器或者I/O写	STR	Rs, address	Rs → [address]
	寄存器访问	MOV	Rd, Rs Rd, Imm	(Rs) → (Rd) Imm → Rd

模型机指令集

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

指令类型		操作码	操作数	说明
控制类	无条件转移	JMP	Label	Label → (PC)
	条件转移	JX/JNX	Label	条件成立则Label→(PC)
	过程调用	CALL	SR_Label	SR_Label → (PC)
	过程返回	RET	—	断点指令地址 → (PC)
其他类	停机	HLT	—	

- RISC指令风格：定长指令、Load/Store体系、三操作数
- Rs-源操作数，Rd-目的操作数，Imm-立即数
- Label-标号（用符号表示的指令地址）

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

模型机指令集

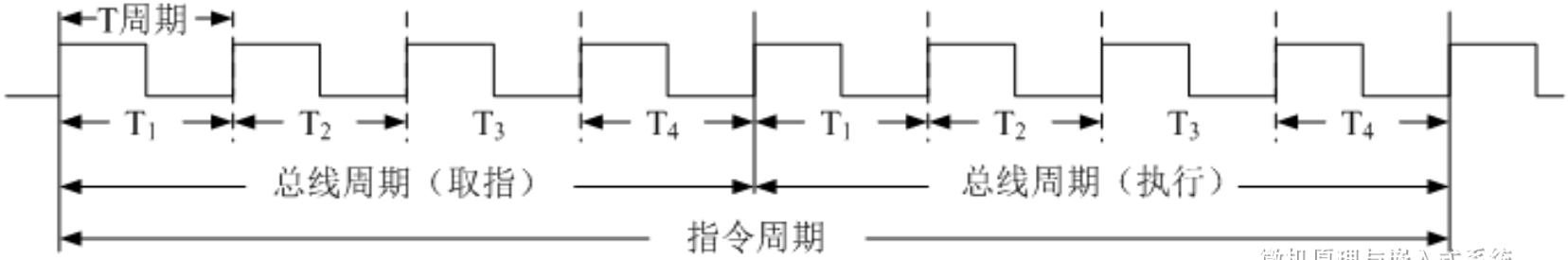
- 定长指令：每条指令长度固定。
- 特点：（假设32位机，32位指令长度）
  - 一次取指操作读取一个完整的指令
  - 受指令位数限制，对立即数的大小或者类型有要求
  - 同样原因，对内存寻址时，无法在指令中直接给出内存单元地址（没有类似于Intel 8086的直接寻址方式）
  - 怎么办？内存单元地址可用如下方式表示：
    - 某个32位寄存器中的数值—寄存器间接寻址
    - 某个32位寄存器内容+偏移量—基址加偏移量寻址

      （同样原因，偏移量大小受限，例如小于2^{12}=4096）
    - 某两个寄存器之和——基址加索引寻址

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

指令周期

- 指令周期：开始取值到完成指令操作的时间
  - 因功能和操作内容不同，许多处理器指令周期也不同
  - 有些采用流水线技术的RISC处理器，所有指令执行时间相同，被称为单周期处理器
- 一个指令周期分为若干个CPU周期（也称为机器周期）
  - 一个CPU周期等于一次取指时间，也称为总线周期
- 一个总线周期包括若干个T周期，也称为时钟周期
  - T周期或时钟周期是处理器最基本的时间单位



模型机指令执行流程

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

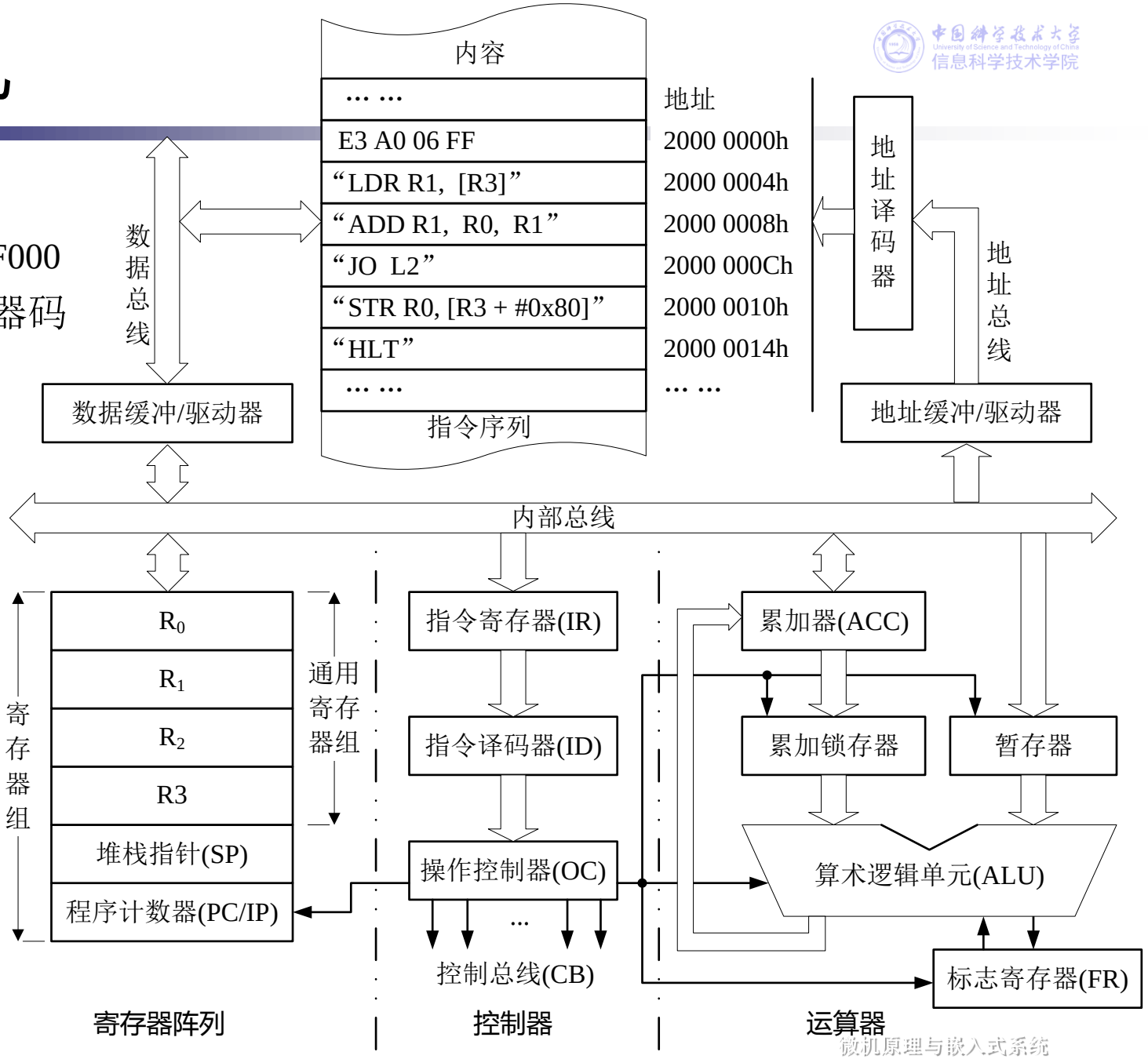
- 示例：利用模型机汇编指令编程实现如下操作
  - 将数据000FF000h（0x000FF000）与内存中某个字数据相加，字数据的地址位于R3寄存器中。如果相加结果没有溢出，则将0x000FF000存入由R3＋80h指定的内存单元，然后停机；如果溢出，直接停机
  - 模型机汇编语言源程序片段：

```
START:                                ; START是第1条指令的标号（符号地址）
    MOV    R0, #0x000FF000            ; 000FF000h → R0，“#”表示是立即数
    LDR    R1, [R3]                   ; [ R3] → R1
    ADD    R1, R0, R1                 ; R0 + R1 → R0
    JO  L2                            ; 溢出则跳转到标号为L2的指令去执行
    STR    R0, [R3 + #0x80]           ; 没有溢出则将R0的内容→ [R3 + 0x80]
L2: HLT                               ; 停机。L2是该指令的标号
```

模型机

假设第一条指令
MOV R0, #0x000FF000
对应的二进制机器码为：E3 A0 06 FF

假设2000 0000h是第一条指令的地址



模型机指令执行流程

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 用汇编语言编写的程序称为**汇编语言源程序**，简称**汇编程序**。将其转换成机器语言程序的过程称为**汇编**，能够实现汇编功能的软件称为**汇编器**或者**汇编软件**
- 汇编后的机器指令顺序存放，若指令长度为4字节，后一条指令地址等于前一条地址加4（PC的单位为字节）
- 讲义图2.12中的几个假设
  - 模型机是32位，数据总线32位，地址总线32位
  - RISC结构，定长指令设计，每条指令长度也是32位
  - 待运行程序的首地址为0x2000 0000
  - 第一条指令的机器码为“E3 A0 06 FF”

模型机指令执行流程

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 执行过程：根据程序的名字，把0x20000000装入程序计数器PC，然后顺序执行如下操作：

① PC内容0x20000000送至地址缓冲器/驱动器，地址总线的输出经地址译码器译码，寻址内存单元

② PC值自动加4（假设PC内容的单位是字节），指向下一条指令的存放地址（何时修改PC有不同的策略）

③ OC发读信号，将“E3 A0 06 FF”读出到数据总线；

④ 由于是取指操作，数据总线上的数据被装入IR

⑤ ID对操作码译码，OC产生相应的控制信号

⑥ 第一条指令源操作数是立即数（取指时能从指令编码中立即得到的数），被装入R0寄存器后指令执行完毕

模型机指令执行流程

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 后续指令执行过程不再进行动作分解
  - 第2条：Load操作，从内存取操作数到R1，操作数地址由R3提供
  - 第3条：ADD运算，R0与R1相加，结果存入R1寄存器，完成运算后更新FR中相关状态位
  - 第4条：条件转移，溢出则跳转执行标号为L2的指令，PC = PC + m（m是转移目的指令与转移指令之间的相对距离）；否则继续执行下一条指令
  - 第5条：Store操作，源操作数是R0寄存器的内容，目的操作数的地址是R3寄存器的内容再加上偏移量80h
  - 第6条：HLT停机，持续执行空操作

模型机指令执行流程

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 总结和讨论
  - 指令执行时，指令操作码送到IR，经IR译码后OC产生操作控制信号
  - 指令的地址码送到地址形成部件，生成地址信号
  - CPU中的所有数据都在数据通道中传送
  - 指令执行过程属于多级串行作业，始终有一部分部件处于空闲状态，部件的利用率不高
  - 模型机属于冯·诺依曼结构，串行作业方式的取指和存取操作数在时间上相互错开，不会出现冲突
  - 如果改用流水线方式，前后指令的取指和存取操作数可能同时发生。冯·诺依曼结构将造成总线竞争，故不太适合流水线模式（不适合不代表不可以）

本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

- CISC和RISC
- 流水线技术
- 超标量机和多发射技术
- 超线程处理器
- 多处理器计算机和多计算机系统
- 多核处理器

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

计算机体系结构的进步

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 从20世纪80年代中期开始，伴随着微电子、通信以及网络技术的发展，计算机体系结构也发生了巨大的变化，使得计算机系统的性能呈现出突飞猛进的进步，主要表现在以下几个方面：
  - RISC的兴起以及与CISC的相互借鉴与融合
  - 指令集（包括指令功能、格式和寻址方式）的优化
  - 使用高速缓存以减少数据存取时间
  - 流水线、超标量和乱序执行等并行计算技术
  - 不断推陈出新的高性能总线
  - 指令和数据分别存储的哈佛结构
  - 多机系统、多核芯片和多线程处理器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

CISC和RISC

- CISC：指令数量多、功能丰富、可实现复杂操作
  - 采用微程序控制器，“初心”是减少硬件实现难度，减少硬件规模
  - 控制器结构简单、规整。增加新指令，或者为已有指令添加新的功能较为容易 → 指令系统规模日渐庞大

指令多 功能复杂 → 微程序和微指令多 → 控制器规模增加 → 微程序执行流程长

  - 结果却有违初衷。控制器占据了芯片大部分面积，性能与晶体管数量不成正比
  - 转换思路 → RISC体系结构诞生

CISC处理器指令特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

1）指令长度不一

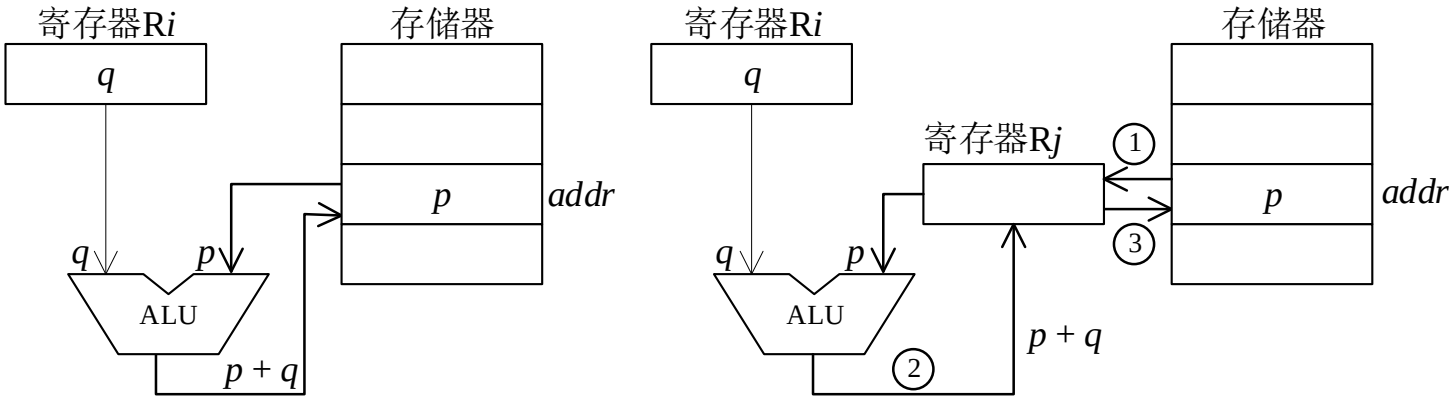
- 例如：
  - Intel 8086处理器的最短指令只有一个字节，如CLC（将进位状态标志位清零），最长指令6字节。指令长度的变化范围为1~6字节
  - Motorola 68020处理器（32位）的指令长度从半个字到8个字，变化范围为2~32字节
- 较长指令取指需要使用多个总线周期和多次总线操作
- 长短不一的指令给指令译码器设计带来挑战，增加了控制器的复杂性和电路规模，也不利于采用流水线和超标量等新技术

CISC处理器指令的特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2）非Load/Store体系

- 算术和逻辑运算指令的操作数可以是存储器数
  - 例如：存储器数+寄存器数→存储器



（a）CISC处理器一条指令即可完成　　（b）RISC处理器需要3条指令才能完成

  - CISC仅需要一条指令： ADD [addr]，Ri
  - RISC则需要三条指令：LDR、ADD和STR，比CISC至少多出2次取值和译码操作

CISC处理器指令的特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

3）MOVE操作

- 格式为： Move destination，source 的传送指令，可实现寄存器与寄存器之间，以及寄存器与存储器之间的数据（复制）传送
- 大多数CISC处理器规定，MOVE指令的源操作数和目的操作数最多只能有一个是存储单元
- 同一条总线上的两个存储单元，如果彼此之间需要传送数据，无论是RISC和CISC处理器，无论采用多条简单指令还是一条复杂指令，数据传送过程都分为两步
  - 1st，将源操作数从内存单元中读出到某个通用寄存器暂存
  - 2nd，将暂存的内容写入目的存储单元

CISC处理器指令的特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

4）两操作数

- 在现代CISC处理器中，大多数算术和逻辑运算指令只有两个操作数，其格式一般为：

  OPR(操作码)　DST(目的操作数)，SRC(源操作数)

- 例如，加法指令：

  Add B, A

对应的操作是(B)＋(A)→B。指令执行后，结果送到B原来的存储位置，替换原先的内容。这意味着上述指令中虽然只有两个操作数，但实际上DST不仅是目的操作数，也是参加运算的两个源操作数之一。

CISC处理器指令的特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

5）指令功能强大、寻址方式多样、程序简洁

- CISC处理器的指令功能强大，可以实现复杂操作，灵活多样的寻址方式有利于软件编程
- 与RISC处理器相比，完成同样任务CISC处理器所需的指令数量较少，软件显得较为简洁
- 指令数量少意味着所需的取指和译码操作次数也较少，在不考虑其他技术因素（如流水线）的情况下，CISC处理器的执行效率要高于RISC处理器

- 此外，CISC处理器采用微程序控制器，逻辑结构简单清晰，易于增加新的指令，或者对已有指令功能进行调整，所以在相当长的一段时间内“一统江湖”

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

CISC处理器的性能问题

- 主要影响因素：微程序控制器的工作流程
  - 完成一条指令需要从控制ROM中顺序读出多条微指令，需要多个在时间上序贯执行的微操作，这种在时间上的串行作业模式将影响指令的执行速度
- 两种解决思路：
  - 提高处理器的工作时钟频率，加快微操作的节奏。但是增加时钟频率受到半导体材料物理特性的限制，并且难以消除由此产生的功耗和发热问题
  - 使用流水线和超标量等技术，让多条指令在时间上并行执行。但是囿于CISC体系结构的特点，流水线和超标量的设计和实现遭遇了很多困难

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

RISC体系结构的诞生

- John Cocke的贡献：除了发现处理器中的“二八定律”以外，研究并提出了以下意见
  - 减少指令数量，简化寻址方式，只保留部分最常用的指令，降低指令译码器的设计和实现难度
  - 复杂操作通过编译技术由多条简单指令组合完成
  - 摒弃微程序和微指令，操作控制器采用硬连线逻辑
  - 算术和逻辑运算都安排在寄存器之间进行
  - 所有指令长度相同，并尽可能在相同的时钟周期内完成

指令少且简单 → **硬件简单电路少** → **增加Cache** → **研制周期缩短** → **便于并行处理**

  - 1980年， John Cocke主持研制的IBM 801获得成功

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

RISC体系结构的诞生

- Patterson & Hennessy两位的主要贡献
  - Patterson：创造了RISC这个术语，主持研制了RISC-I处理器，验证了RISC架构的优势，指导了SPARC处理器的设计
  - Hennessy：创办了MIPS公司，主持研发了MIPS处理器，（龙芯处理器采用的就是MIPS架构，龙芯指令集与MIPS兼容）
  - 两位共同主编了《Computer Architecture - A Quantitative Approach》一书，被誉为计算机体系结构领域的圣经，为人才培养做出了卓越贡献
  - 如今两位前后脚离开了学术界， Hennessy现任Alphabet公司董事长， Patterson仍在领衔研发TPU（谷歌公司用于人工智能计算的一款处理器）

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

RISC处理器及其指令的特点

- RISC处理器：
  - 摒弃微程序设计思想，采用硬连线方式实现控制器
  - 为了减少硬件实现难度，采用精简指令集
  - 减少处理器电路数，多余的芯片面积用于增加Cache容量及寄存器数量，利用层次化的存储结构优化数据传送
  - 指令简单、长度一致、执行时间相同等特点使其易于引入流水线和超标量等可大幅度提高处理器性能的并行处理技术
  - CISC处理器一条复杂指令就能实现的操作，RISC处理器需要使用多条简单指令才能完成
  - 需要优秀的程序编译器，优化由数量较多的简单指令构成的程序代码

RISC处理器及其指令的特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- RISC处理器的指令简单明了，易于理解。纯粹RISC风格的指令具有以下一些主要特点：
  - 寻址方式简单，种类较少；
  - 指令集中的指令数量较少；
  - Load/Store体系结构；
  - 每条指令长度一致，执行时间相同；
  - 面向寄存器的编程思想（早期的CISC属于面向累加器）；
  - 算术和逻辑运算指令普遍支持三操作数；
  - 只能对寄存器操作数进行算术和逻辑运算；
  - 程序代码量较大，因为执行复杂操作需要使用较多的简单指令

CISC和RISC的相互借鉴和融合

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 两种体系结构相互借鉴和融合，使得现代计算机中几乎不再有纯粹的CISC或者RISC计算机，例如
  - 为了减少执行一项操作所需的指令数量，RISC处理器也增加了一些能够快速执行的非RISC指令
  - 为降低硬连线控制器的设计难度，有些RISC处理器还是部分采用了微程序和微指令设计
  - Intel通过与RISC阵营的合作引入RISC技术，从P6开始，处理器中有一部分采用了RISC设计
  - Intel 64位IA-64架构处理器是在HP的帮助下完成的；Intel所提出的EPIC（Explicitly Parallel Instruction Computers，精确并行指令计算机）体系大部分是Alpha的遗产

流水线技术

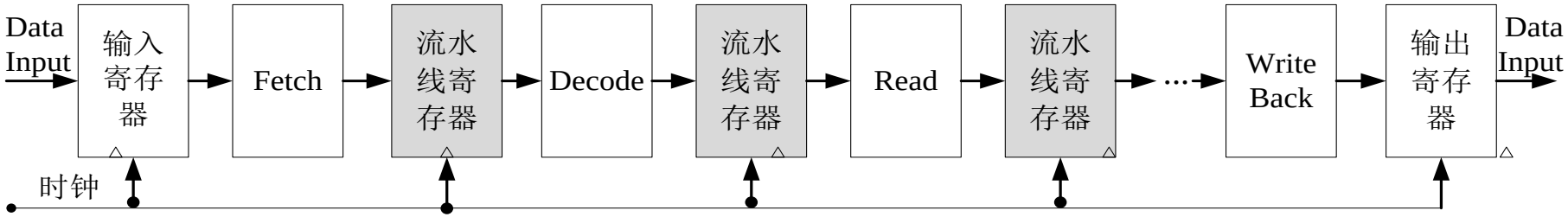
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 指令执行过程可分解为多个步骤，假设分解为：
  - Fetch（取指）、Decode（译码）、Read（取操作数）、Execute（执行）和Writeback（回写）5个步骤
- 每个步骤都有专用部件完成操作，各部件分工明确，各司其职，每个步骤按照顺序执行
- 模型机任何时候只能执行一条指令，前后指令呈多级串联作业模式，将会造成部件的“窝工”，例如
  - 取指时，译码、取操作数、执行和回写处于等待状态
  - 执行时，取指、译码、执行和回写又处于等待状态
- 如何消除或者减少“窝工”现象以提高处理器性能？

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

流水线技术

- 将功能部件按指令操作步骤顺序进行排列部署，前后部件之间增加缓冲寄存器，构成指令处理流水线
- 前后两个部件经过缓冲寄存器隔离后，可以相对独立地并行工作



- 部件之间的工作交接（数据传递）将通过缓存寄存器进行
- 这种缓存寄存器被称为流水线寄存器。

流水线技术

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 多条指令可以在流水线上以时间重叠方式序贯执行

周期	1	2	3	4	5	6	7	8	9	10
指令1	Fetch	Decode	Read	Execute	Write					
指令2						Fetch	Decode	Read	Execute	Write

（a）非流水线多级串行执行

周期	1	2	3	4	5	6	7	8	9	10
指令1	Fetch	Decode	Read	Execute	Write					
指令2		Fetch	Decode	Read	Execute	Write				
指令3			Fetch	Decode	Read	Execute	Write			
指令4				Fetch	Decode	Read	Execute	Write		
指令5					Fetch	Decode	Read	Execute	Write	
指令6						Fetch	Decode	Read	Execute	Write
指令7							Fetch	Decode	Read	Execute
指令8								Fetch	Decode	Read
指令9									Fetch	Decode
指令10										Fetch

（b）指令时间重叠流水线方式执行

流水线技术

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 由流水线时空图可看出，在10个流水线周期内
  - 串行作业模式只能完成2条指令的执行
  - 流水线方式至少可以完成5条指令的执行，并且指令6到指令10也开始了部分执行。多条指令以时间重叠方式执行，使得IPC大大提高
- 注意到增加流水线寄存器后，增加了流水线寄存器的写入时间和额外的门电路时延。因此，单条指令在流水线上执行所花费的时间要比非流水线方式更长
- 时空图有一个假设，流水线上每一级所需的处理时间相同。如果不一致，流水线周期将受制于最慢的部件，这将导致流水线的性能下降

流水线中的主要问题

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 流水线高效运行的前提：保持畅通，不发生断流
- 但指令执行过程会出现以下三种相关冲突（也称冒险）

1）资源相关，也称为结构相关

- 多条指令在同一个周期内争用同一个公用部件。例如：冯·诺依曼结构计算机的Fetch、Load和Store操作都使用公用总线接口访问同一个存储器，前一条指令的数据存取操作可能会影响后续指令的取指操作
- 解决方法：
  - ①后面一条指令等待一个节拍再启动。称作向流水线插入气泡（Bubble）或插入阻塞，这将造成流水线性能下降
  - ②采用哈佛结构。解除存取操作数与取指之间的资源相关

流水线中的主要问题

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2）数据相关

- 后一条指令执行需要使用前一条指令的结果。例如：流水线上前后执行的两条算逻指令：

```
SUB    R1, R2, R3     ; R2减R3结果写入R1
AND    R4, R1, R5     ; R1和R5逻辑与运算，结果写入R4
```

周期	1	2	3	4	5	6	7
SUB	Fetch	Decode	Read	Execute	Write		
ADD		Fetch	Decode	Read	Execute	Write	

- SUB指令在第5个周期才将结果写回R1，但是AND指令在第4个周期就要读R1进行运算，而程序的本意是“写后读”（Read After Write，RAW）
- RAW是一种最为常见的数据相关

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

流水线中的主要问题

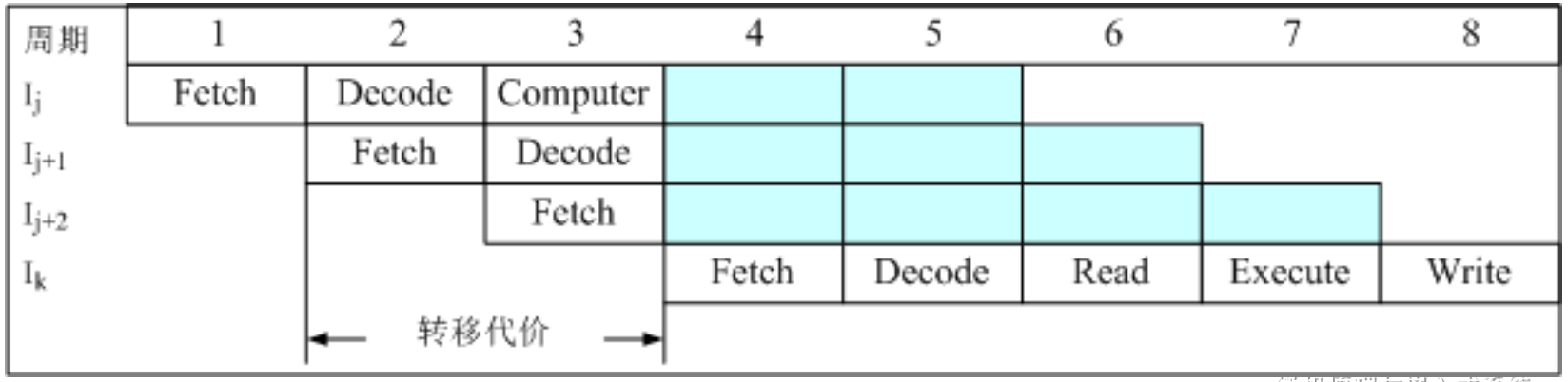
- 流水线还可能存在“写后写”（WAW）和“读后写”（WAR）两类数据相关
  - WAW：I_{j+1}试图在指令I_j写数据之前写数据，这样最终结果将由I_j决定，而程序本意是保留I_{j+1}的结果
  - WAR：I_{j+1}试图在指令I_j读一个数据之前写该数据，此时指令I_j读到的是被I_{j+1}“篡改”后的数据
  - 插入气泡可消除数据相关，但将造成流水线性能下降
  - 解决之道：
    - 定向推送，前一条指令执行结果通过专用通道直接推送给下一条，减少一个流水线周期，可减少数据相关
    - 优化编译器，对前后指令进行检查，调整执行顺序

流水线中的主要问题

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

3）控制相关

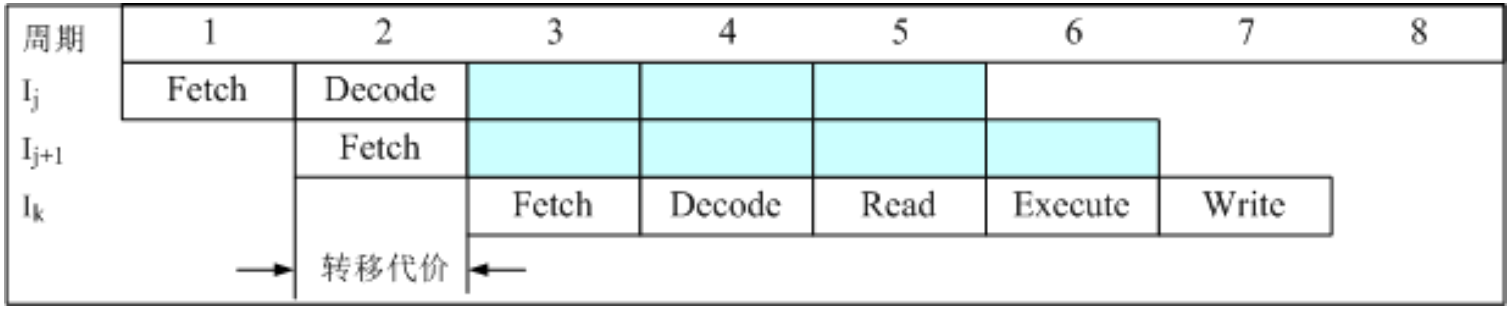
- 遇到转移指令时，后续已进入流水线的指令都应清空
- 以无条件转移（包括子程序调用）指令为例：
  - 假设指令I_j是无条件转移指令，其执行步骤为：取指、译码、计算转移地址（Computer）并更新程序计算器PC。在第4个周期读取转移目标指令I_k。在此之前流水线上的指令I_{j+1}和I_{j+2} 应清除，造成流水线断流。产生两个流水线周期延迟被称为转移代价（branch penalty）



流水线中的主要问题

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 减少转移代价的方法
  - 对于无条件转移指令，增加电路，在译码阶段提前计算转移目标地址，在第3个周期读取转移目标指令I_k，将转移代价减少到一个流水线周期



  - 类似方法同样适用于少数条件转移指令
- 但是，大多数条件转移指令是否转移取决于状态标志位，而标志位在ALU运算后才更新，转移代价较大。流水线级数却多，代价越大。该如何解决呢？

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

转移预测技术

- 转移延迟槽（branch delay slot）：转移指令I_j后面的一个**时间片**。无论是否转移，位于转移延迟槽的指令总是会被执行。一个启示：
  - 如果能对转移与否做出正确预测，则可根据预测结果选择合适的指令“装入”转移延迟槽。例如，假设预测结果是转移，位于转移延迟槽的是转移目的指令I_k，转移没有任何代价
- 问题：如何对转移指令的行为进行预测？
- 动态转移预测：根据转移指令过去的行为进行预测
  - 评估：用2bit（权值）对转移指令过去行为进行量化，‘11’总是转移，‘00’总是不转移

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

动态转移预测

- 动态“打分”：每发生一次转移，权值+1，加到11b为止；每发生一次不转移，-1，减到00b为止。
- 使用BTB（转移目标缓冲器），收集和存储了近期所有转移指令的有关信息，并按照查找表的形式进行组织
- BTB不能太大，一般为1024个表项，其内容包括：
  - 转移指令I_j的地址（查找表索引）
  - I_j转移可能性的量化结果（2bit权值）
  - 转移目标指令I_k的地址
- 每条指令在取指时，处理器根据其地址在BTB中进行快速搜索，若有记录则表明这是转移指令，并根据其“档案”进行相应处理，最后再根据这条指令的实际行为修正BTB的记录内容。

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

超标量（super-scalar）和多发射技术

- 标量处理器：只能处理标量数据，一条指令一次只能处理一个数据，属于SISD
- 向量处理器（阵列处理器），一条指令完成一个向量计算，属于 SIMD，用于科学计算和信号处理等领域
- 上述流水线只能处理单条数据，属于标量流水线
- 超标量处理器：拥有多条流水线，通过空间并行方式提高处理能力
  - 将多条指令分发到多条流水线上，同时执行
  - 分发前需配对检查，不同流水线上的指令不相干
  - 可以使用不同类型的流水线，如整数与浮点数
- 多发射技术：多个指令分发单元

超线程（Hyper-threading）处理器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 进程（process）：程序的动态执行过程
  - 多任务系统中， CPU的运行时间被划分成多个时间片，CPU在不同的时间片轮流为每个任务进行服务
  - 早期，进程被作为作业调度和资源管理的基本单位
  - 进程运行时拥有所需的全部资源；任务切换时，操作系统回收资源并重新分配，无效开销太大
- 线程（Thread）：能独立执行的代码最基本单位
  - 每个进程拥有若干个线程
  - 线程是作业调度和执行的基本单位，拥有少量的必备资源（如PC），与进程中的其他线程共享全部资源
  - 线程调度时资源不回收，无效开销小

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

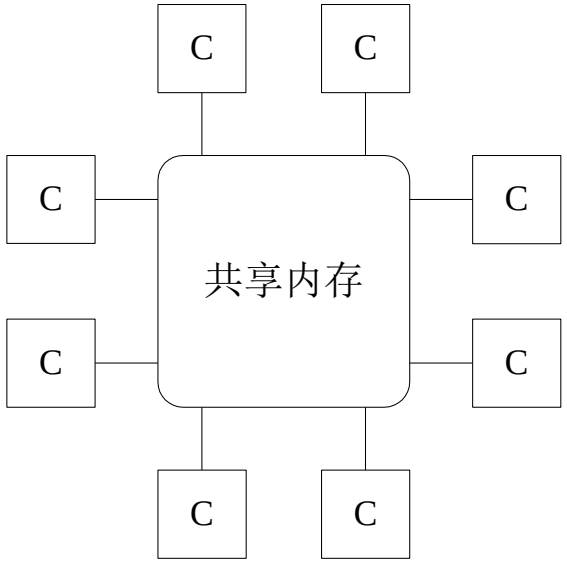
超线程处理器

- 单处理器同时只能执行一个线程，多线程只是利于操作系统的任务调度，减少无效开销，性能提高有限
- 超线程技术：为进一步减少处理器内部的硬件资源闲置，对流水线进行改造并添加少量部件，使处理器在同一时间可以执行两个线程
  - 超线程只是有了两个逻辑上的线程处理单元，每个线程并不是独自拥有所需的全部资源
  - ALU、FPU、Cach和总线接口等仍是两个线程共享
  - 超线程需要操作系统、应用软件以及主板BIOS的支持
  - 性能提升：多任务时可提升30%，单任务时处理器性能不升反降

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

多处理器计算机

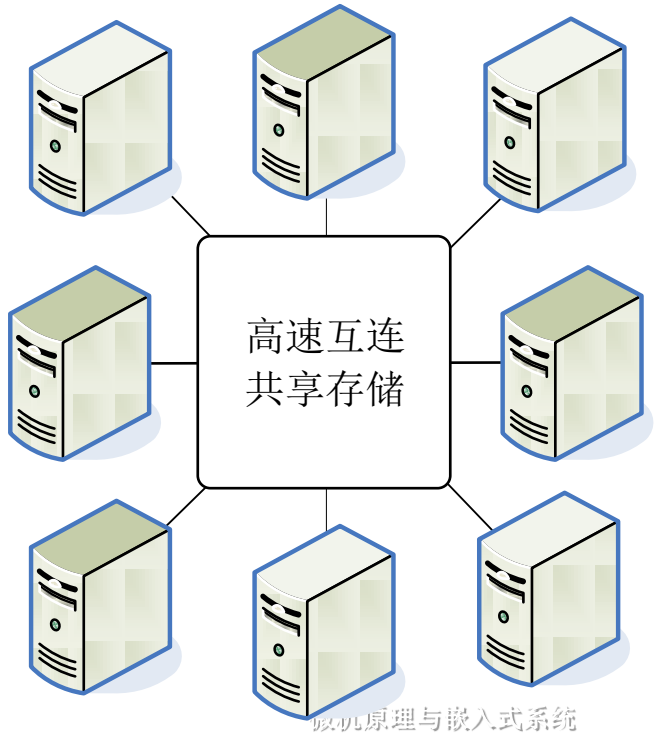
- 一个计算机有分布在不同的芯片上的多个处理器
- 多个处理器芯片通过共享内存或共享总线进行数据交换，并行工作，属于一种紧耦合多处理器系统
- 多处理器计算机分为：
  - 非对称多处理器计算机AMP
    - 处理器有主从之分，不同的处理器承担不同的任务
  - 全对称多处理器计算机SMP
    - 现代服务器的主流架构。各处理器地位相同，对称工作；所有共享系统内存、I/O通道和外部设备。



中国科学技术大学 University of Science and Technology of China 信息科学技术学院

多计算机系统

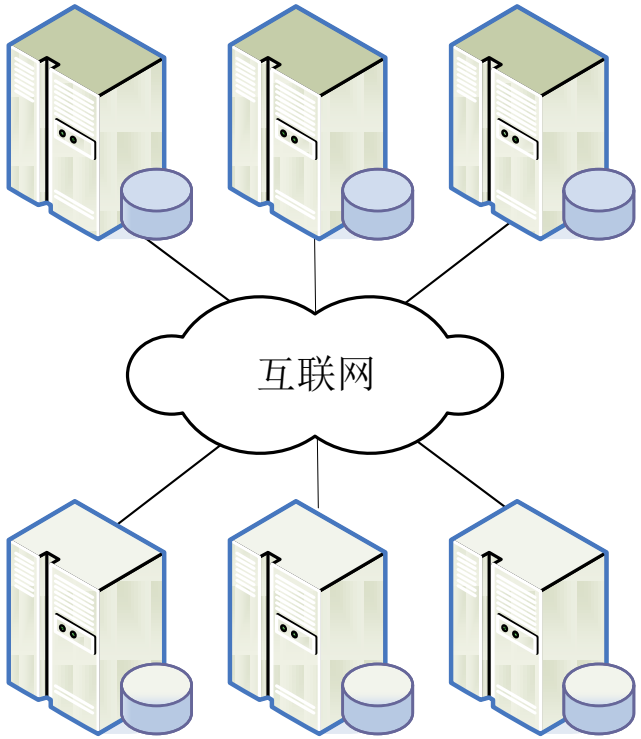
- 多台计算机通过局域网以及私有网络彼此互连
- 每台计算机受各自独立的操作系统控制，有属于自己的存储系统和I/O设备，属于一种松耦合系统
- 可通过LAN或者SAN（存储域网）共享外部存储器，组成计算机集群（Cluster）
- 除提高了计算能力以外，也减少单点故障，具备高可用性（High Availability），普遍应用于执行关键任务的信息系统（mission-critical systems）中



中国科学技术大学 University of Science and Technology of China 信息科学技术学院

分布式计算机系统

- 若干独立计算机或者集群通过网络互连而成
- 有一个全网统一的分布式操作系统，能够对用户所需的各种资源进行统一调度和管理，并且保证系统的一致性与透明性（不可见）
- 用户无需关心系统中的资源分布情况以及计算机差异
- 计算机之间没有主从之分，彼此既合作又自治，协同工作
- 分布式计算机系统是计算机应用领域发展的一个重要方向，也是云计算的主要基础



中国科学技术大学 University of Science and Technology of China 信息科学技术学院

多核处理器（Multi-core）

- 可集成的电路数越来越多，可以把多个功能完整的CPU集成在一个芯片上——单芯片多内核处理器
- 每个计算内核普遍采用超标量和超级流水线技术，拥有所需的全部计算资源，可彼此独立地执行任务
- 多个内核通过片内总线或交叉开关矩阵互连，可看作一个片上多处理器机CMP（Chip Multiprocessor）系统，对外呈现为一个统一的处理器
- 分为同构多核和异构多核两种类型
  - 同构-所有内核数相同，异构-根据需求选配不同的内核
- 芯片上的内核数量是不是多多益善？

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

多核处理器（Multi-core）

- 多核处理器性能与任务的并行性有关
- 安达尔定律（Amdahl‘s law）加速比 $S=\dfrac{1}{1\text{-}a+\dfrac{a}{n}}$

  n：节点数

  a：可并行代码比例

  - 若a=0，全是串行，S=1；　　若a=1，全是并行，S=n
  - 若20%代码为串行，a=0.8，即使n→∞，S→5
- 应在核数、功耗、代价和实际效果间寻求平衡
- 异构多核的特点：
  - “让专业的人做专业的事”，避免“大马拉小车”和“小马拉大车”
  - 范例：华为海思麒麟处理器

本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

- Intel 8086处理器
- Intel Pentium处理器

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

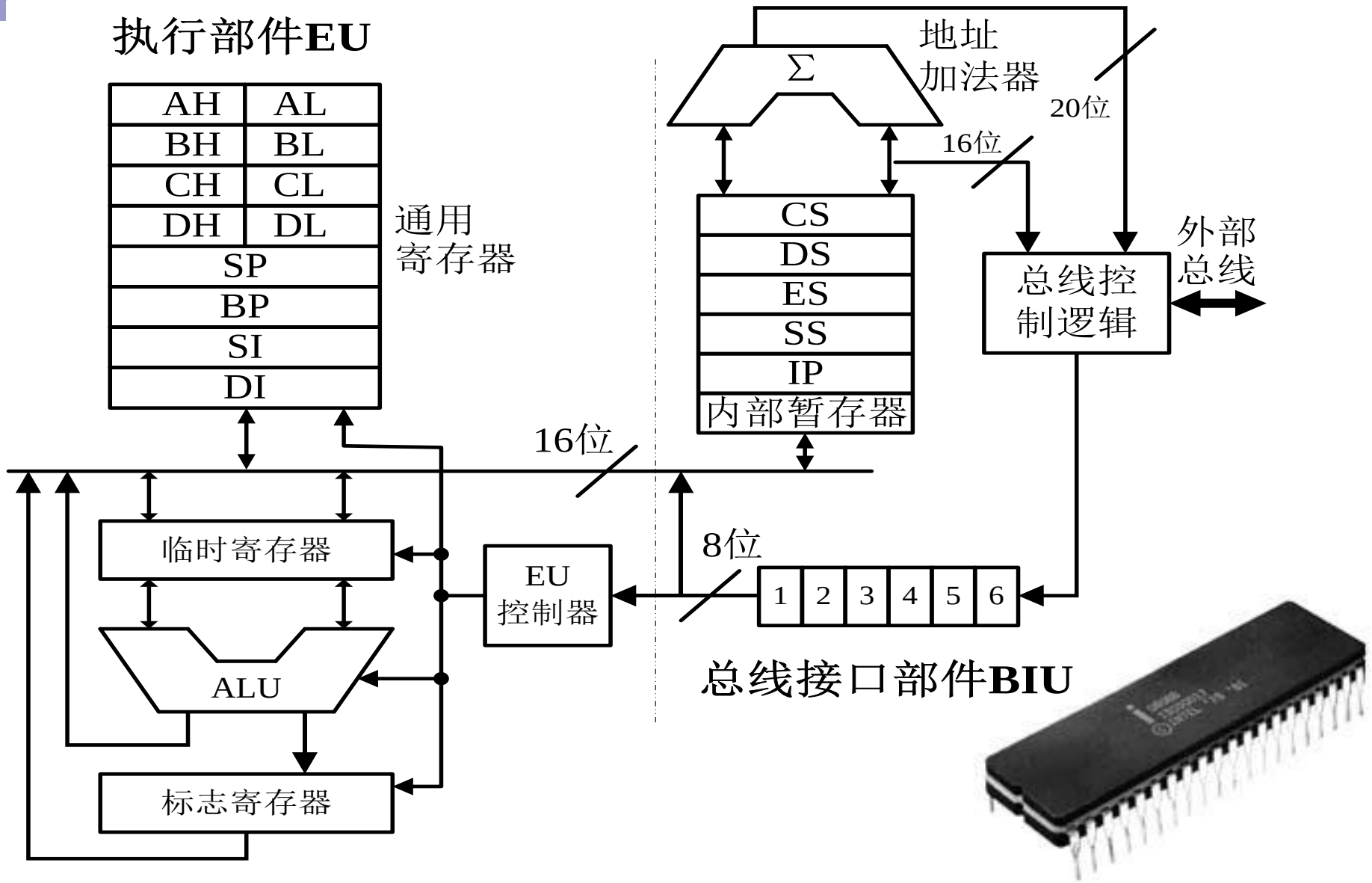
Intel 8086处理器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 全球第一款16位通用微处理器芯片，诞生于1978年
- 以8位 Intel 8080/85为基础，为了与其兼容，采用了类似的寄存器组，接口芯片多属于“遗产”
- 基本情况：
  - 内部寄存器为16位，其中有4个可以分拆成8位使用
  - 16位数据总线、20位地址总线
  - 内存分段管理，段寄存器左移4位+偏移量=20位地址
  - I/O端口独立编址，端口总数为64K个
  - 时钟频率仅有5MHz，最高不超过10MHz
  - 分成指令执行单元EU和总线接口单元BIU两部分

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

Intel 8086处理器



Intel Pentium处理器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 背景：
  - 8086之后有80286、80386和80486，先后用于基于IBM AT或者ISA架构的PC机中
  - 从80386开始，PC机进入了32位。为了与之前16位系统兼容，采用了一些较为复杂的技术
  - 同期AMD和Cyrix等公司也有类似处理器，采用与386、486相同的名称，这类处理器统称为x86处理器
- 1993年3月22日，Intel向外正式销售Pentium芯片
  - 310万个晶体管，70%电路用于兼容以往产品
  - 由于商标问题，没有按照习惯称为80586
  - 第一次有了中文产品名称：奔腾

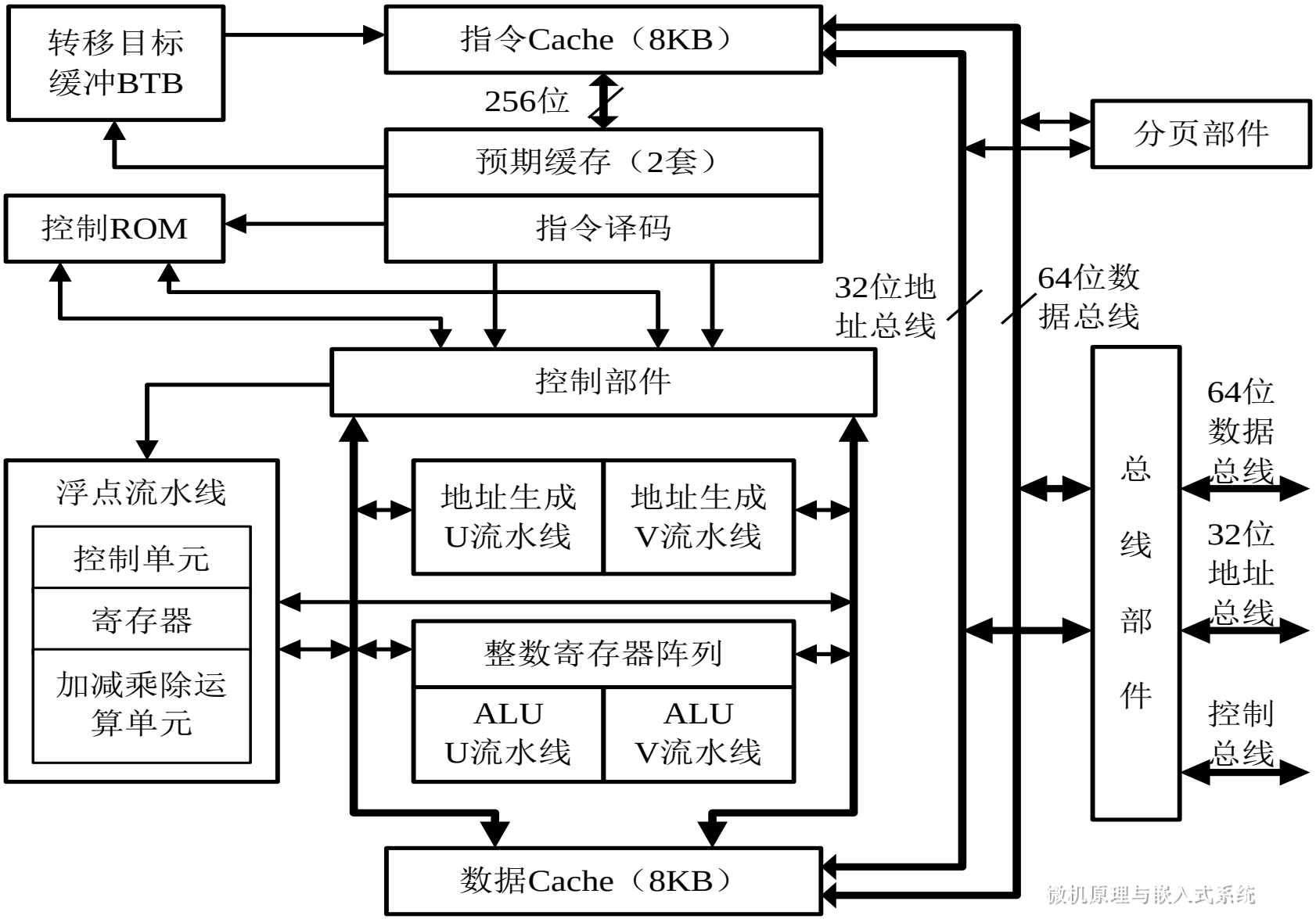
中国科学技术大学 University of Science and Technology of China 信息科学技术学院

Intel Pentium处理器

- 微处理器史上的里程碑产品，主要特点有：
  - 与以往80x86系列微处理器兼容
  - 32位地址总线，64位数据总线
  - 采用CISC结构实现超标量结构，两条并行的5级整数指令流水线（U和V），一条8级浮点运算流水线
  - 独立的指令Cache和数据Cache，数据和代码分离
  - 汲取RISC的优点，简单指令改用硬连线控制器实现
  - 基于BTB（转移目标地址缓冲器）的预测转移技术
  - 191条指令，支持9种寻址方式
  - 支持64位外部数据总线突发传输方式
  - 支持SMM模式，增强的错误检测和报告功能

Intel Pentium处理器

中国科学技术大学 University of Science and Technology of China 信息科学技术学院



本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

- ARM体系结构、ARM处理器和ARM内核
- ARM处理器的特点
- 典型ARM内核基本结构

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM体系结构

- ARM公司迄今已经推出8个大的体系结构版本，从ARMv1~ARMv8。其中，ARMv1~ARMv7属于32位架构，ARMv8则是新一代的64位架构
- 在每个大版本中，根据实现技术、功能和性能等要素，又划分若干子版本。例如，ARMv7版本被分成了：
  - ARMv7-A，对应的产品系列是ARM Cortex-A
  - ARMv7-R，对应的产品系列是ARM Cortex-R
  - ARMv7-M，对应的产品系列是ARM Cortex-M
  - 体系结构版本号与产品名称的对应关系还没有理顺

“实体清单事件”之后，ARM表示将继续与华为海思在ARMv8以及下一代的v9展开合作

中国科学技术大学
University of Science and Technology of China
信息科学技术学院

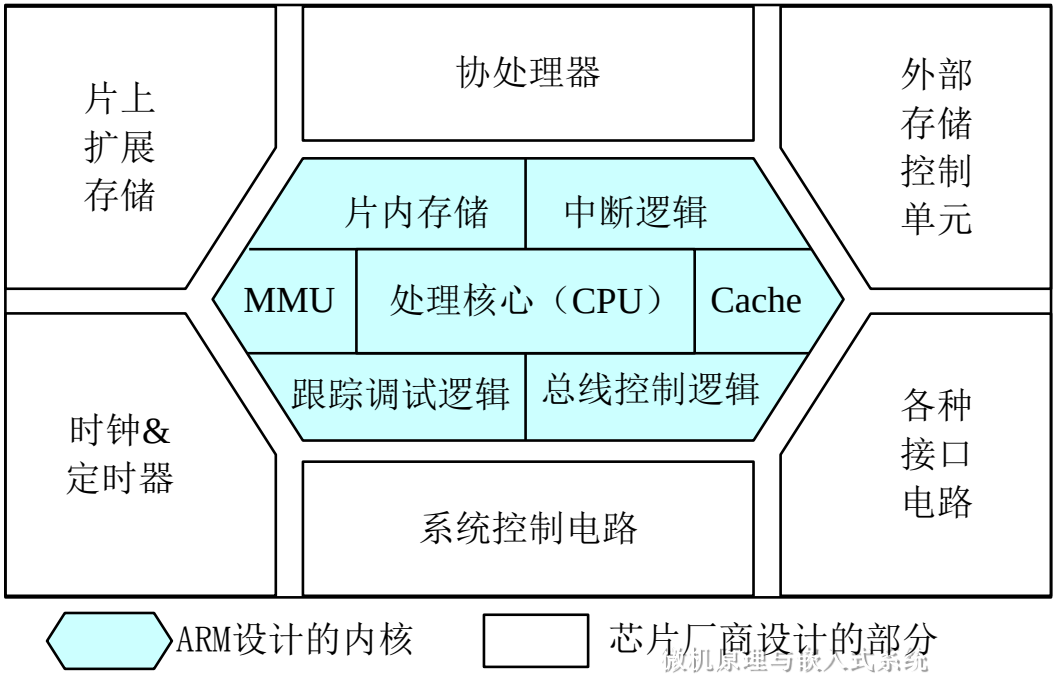
ARM处理器和ARM内核

- 严格地说，ARM处理器是ARM公司设计的处理器
  - 基于每个体系结构版本，ARM公司根据目标市场的应用需求，设计了多款嵌入式处理器，这些处理器内部配置的硬件部件、可实现的功能和适用场景等各不相同
  - 少数早期设计的处理器中，内部仅有最基本的数据处理核心，习惯上称之为**内核**（Core，可以看作CPU）
  - 如今ARM处理器普遍带有开发调试所需的嵌入式ICE或ETM等；有些还可能带有高速缓存、片上ROM和RAM、MMU或MPU、中断逻辑和硬件加速器等，在功能和性能方面各有差异
  - 基于同一体系结构版本，应用软件层面可相互兼容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM处理器和ARM内核

- 芯片制造商获得授权后，根据实际需求和产品定位，在某款ARM设计的处理器基础上，再增加诸如实时时钟、ADC、DAC、存储器、协处理器、DSP，以及各种接口单元部件，形成多种各具特色的嵌入式处理器芯片，实际上应该属于**SOC**，但是往往**也被称为ARM处理器**（芯片）
- 其中由ARM公司设计的处理器也被称为**ARM内核**
- 鉴于上述原因，本门课程对ARM设计的处理器或者ARM内核不再严格区分



ARM处理器的特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- ARM处理器的成功三要素
  - 代码密度高（代码占用内存少）
  - 功耗低
  - 性价比
- 不同版本ARM内核都具有RISC架构的共同特征：
  - 每条指令长度固定
  - 指令集中的指令数量较少
  - Load / Store体系结构
  - 只能对寄存器操作数进行算术和逻辑运算
  - 采用硬件布线逻辑，大部分指令在一个周期内完成执行。

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM处理器的特点

- 但是，ARM并没有完全照搬纯粹的RISC设计，而是以“疗效”为导向，不拘一格，对传统RISC架构有所扬弃，并吸取了一些CISC架构的特点，例如：
  - 保留少数功能强大的复杂指令，如多寄存器传送指令
  - 提供自增、自减指令和基于PC的相对寻址方式；
  - 用于转移指令和条件执行的条件码（N-负、Z-零、C-进位、V-溢出）；
  - 少数指令可以在多个周期内完成。
- ARM还有一些其他特点，如支持不同的指令集、指令的条件执行、以及移位操作的实现方式等
  - 随着ARM体系结构的发展，上述特点也可能会发生变化
  - ARM指令具体内容将在本门课程第5&6章介绍

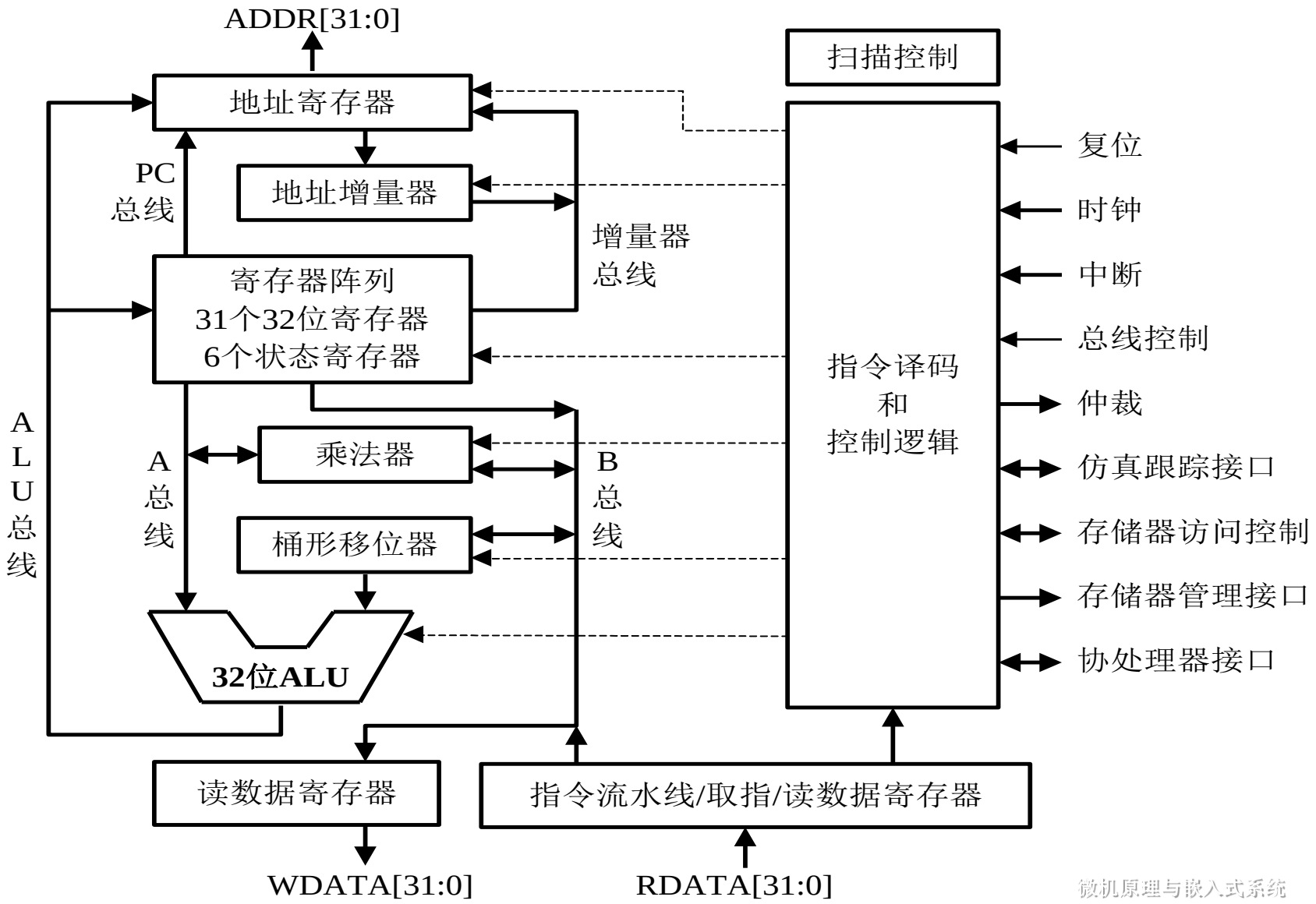
典型ARM内核的基本结构和特点

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

1）ARM7TDMI，ARM7系列基本型产品，1995年发布

- 属于ARMv4T版本，其产品后缀各位含义是：
  - T表示支持16位的Thumb指令（ARM指令集的子集）
    （Thumb指令集与ARM指令集的关系类似于拇指与臂膀）
  - D表示支持片上Debug
  - M表示内嵌硬件乘法器
  - I表示内嵌ICE逻辑
- ARM7TDMI-S是ARM7TDMI的可综合（synthesizable）版本，可以看做是一种软核
- 虽是家族中的低端产品，基于以上2款内核的嵌入式处理器，在问世后20年间内累计销量超过300亿片

ARM7TDMI内部结构



中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM7TDMI简介

- ARM7家族其他产品的基础，**内核中的内核**
- 包括一个32位ALU、一个32位桶形移位寄存器和一个32位×8位乘法器
- 共有37个程序可访问的32位物理寄存器，包括31个通用寄存器和6个状态寄存器
  - 但这些资源不是同时可见，在不同工作状态以及不同工作模式下只能看到其中一部分
  - 在任何状态和模式下，最多只能看到其中的18个
- “写”和“读”数据总线分开，片外只需配置单向总线驱动器，没有双向驱动器的方向转换时延

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM7TDMI的特点

- 获得如此广泛的应用，得益于以下优点：
  - 有ARM和Thumb两种状态，可软件切换，分别支持全功能的32位ARM指令集和简洁的16位Thumb指令集
  - 一条3级（取指、译码和执行）流水线，性能可达0.9 MIPS/MHz；支持8bit、16bit和32bit数据操作
  - 快速中断响应能力
  - 写数据和读数据总线分开，片外无需双向驱动器
  - 冯诺依曼结构，系统简洁，门电路数量较少
  - 高性能、低成本，超低功耗，尤其适合对功耗有苛刻要求的场合，如依赖电池供电的各种手持式电子设备
- 基于ARM7TDMI内核的SOC不计其数… …

ARM9系列产品简介

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- ARM9产品家族可分为ARM9和ARM9E两个系列
  - ARM9系列是基于ARMv4T版本的普通型产品，包括ARM9TDMI、ARM920T、ARM922T和ARM940T
  - ARM9E系列则是基于ARMv5TE版本的增强型产品，具有DSP和Java扩展功能，包括ARM926EJ和ARM946E
- ARM9系列是ARMv4T版本的高端产品，与ARM7相比，在体系结构上主要有以下几方面改进：
  - 指令流水线升级为5级（取指、译码、执行、存储器访问和回写），每级电路更简单，执行速度更快
  - 采用哈佛结构，减少发生资源冲突的概率
  - Thumb指令采用硬件译码，速度高于软件译码的ARM7

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM9TDMI

- ARM9TDMI是ARM9产品家族中的基本型产品
- 同系列的其他处理器（如ARM920T、ARM922T和ARM940T）都是以ARM9TDMI为核心，扩展和集成了其他一些功能部件所构成的。这些功能部件有：
  - 指令Cache和数据Cache、AMBA总线（Advanced Microcontroller Bus Architecture）接口、嵌入式跟踪宏单元ETM、MMU或者MPU等
- ARM9TDMI属于ARMv4T版本中的高端内核，是在ARM7TDMI基础上做的改进，其结构与后者类似
- ARM9TDMI-S是ARM9TDMI的可综合版本

ARM920T处理器简介

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

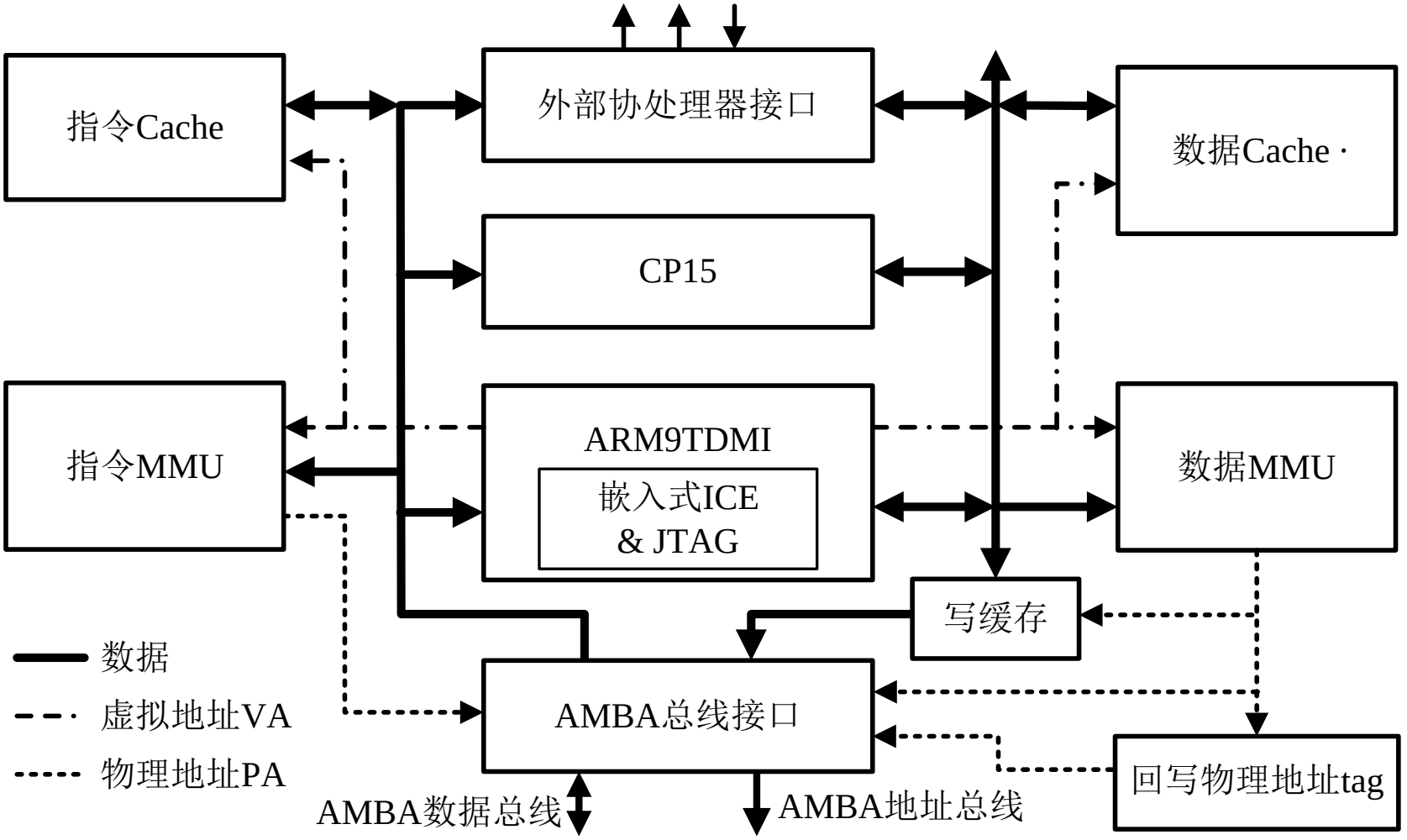
- 属于ARM9系列，以ARM9TDMI为核心，另外配置了
  - 各16KB的数据和指令cache
  - 数据和指令MMU
  - 写缓存（16字的数据， 4个地址）
  - CP15（co-processor）
  - 外部协处理器接口
  - 嵌入式跟踪宏单元ETM

AMBA总线

- 支持VxWorks，WindowsCE和Linux等嵌入式OS
- 典型目标SOC芯片：三星公司S3C2410、S3C2450…
  - 高性价比和低功耗，应用范围广泛

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM920T处理器结构



中国科学技术大学 University of Science and Technology of China 信息科学技术学院

ARM920T处理器结构

- 两类地址信号：物理地址PA和虚拟地址VA
  - Physical Address：每个存储单元所拥有的真实地址
  - Virtual Address：编程时所使用的地址，也称为逻辑地址
  - 由MMU负责PA和VA的映射和转换
- CP15：系统控制协处理器，用于管理和控制Cache、MMU、时钟类型和大小端设定等系统级操作
- 回写物理地址TAG：地址标记寄存器，存放cache中需要回写（更新）数据字段在主存中的地址信息
- 数据总线上的写缓存：提高数据回写操作的速度
- JTAG：符合JTAG规范的测试接口

JTAG：由Joint Test Action Group定义的一套标准测试协议

本章内容

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

2.1计算机系统的基本结构与组成

2.2 模型机存储器子系统

2.3 模型机CPU子系统

2.4 模型机指令集和指令执行过程

2.5 计算机体系结构的改进

2.6 Intel x86典型微处理器简介

2.7 ARM嵌入式处理器简介

2.8 计算机性能评测

计算机性能评测

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 评价一台计算机的性能需用多个维度进行综合评价
- 定性描述指标
  - 机器字长
  - 存储容量，主要是Cache大小和内存容量
  - 总线带宽和数据吞吐速率，主要取决于采用总线标准
  - 能耗与环保
  - RASIS 特性：
    - Reliability，可靠性，MTTF与MBTF
    - Availability，可用性，系统正常运行时间的百分比
    - Serviceability，可维护性
    - Integrity，集成性，all in one
    - Security，安全性

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

计算机性能评测

- 定量描述指标：速度，如各种“跑分”
- 早期指标：MIPS，对CISC和RISC不能客观评价
- 现在指标：基准测试（Benchmark Test）结果，如：
  - Whetstone和Dhrystone：前者包括浮点，后者只有整数，使用FORTRAN编写，代码量太小，与编译器有关
  - CoreMark：2009年由EEMBC（Embedded Microprocessor Benchmark Consortium）提出，用C语言编写，包含嵌入式系统常见的4种计算（矩阵、查找和排序、状态机和CRC），已成为嵌入式内核性能评测的事实标准
  - SPEC（Standard Performance Evaluation Corporation）测试，通用计算机使用最多的基准测试

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

计算机性能评测

- 常用的SPEC测试有：
  - SPECjbb，用于评测JAVA应用服务器的性能
  - SPECint，用于评测整数计算及编译器优化能力
  - SPECfp，用于评测浮点数计算及编译器优化能力
  - SPEC CPU，用于评测单核或多核处理器在进行整数及浮点数计算时的性能，包括多个种类和多个测试项目
- 其他测试基准
  - TPC（Transaction Processing Performance Council），主要针对计算机在数据库应用及事务处理方面的性能。常见的有
    - TPC-C：反映OLTP（联机事务处理）性能
    - TPC-H：反映OLAP（联机事务分析）性能
  - TPC是大型信息系统核心主机选型的重要依据

计算机性能评测

中国科学技术大学 University of Science and Technology of China 信息科学技术学院

- 其他基准测试
  - Linpack（Linear system package，线型系统软件包），一种高性能计算机系统浮点性能测试基准。通过在计算机（集群）系统中运行Linpack测试程序，可以得到能够反映高性能计算机浮点计算性能的测试结果Flops。在高性能计算领域，Linpack指标受到普遍重视
  - SAP基准测试。SAP研发的ERP在大型企业中得到广泛应用。SAP基准测试组织是由SAP及其合作伙伴组成。围绕企业经营活动中的各种业务需求，SAP基准测试组织制定和发布了多个测试功能模块，其测试结果对于ERP系统的硬件选型和配置具有一定指导意义

中国科学技术大学
University of Science and Technology of China
信息科学技术学院

习题

① 2.6

② 2.10

③ 2.13

④ 2.14

⑤ 2.15

⑥ 2.16

⑦ 2.18

⑧ 2.20

⑨ 2.24

⑩ 2.25