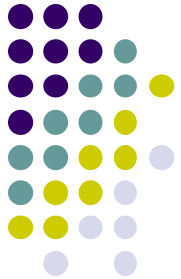


计算机组成与系统结构

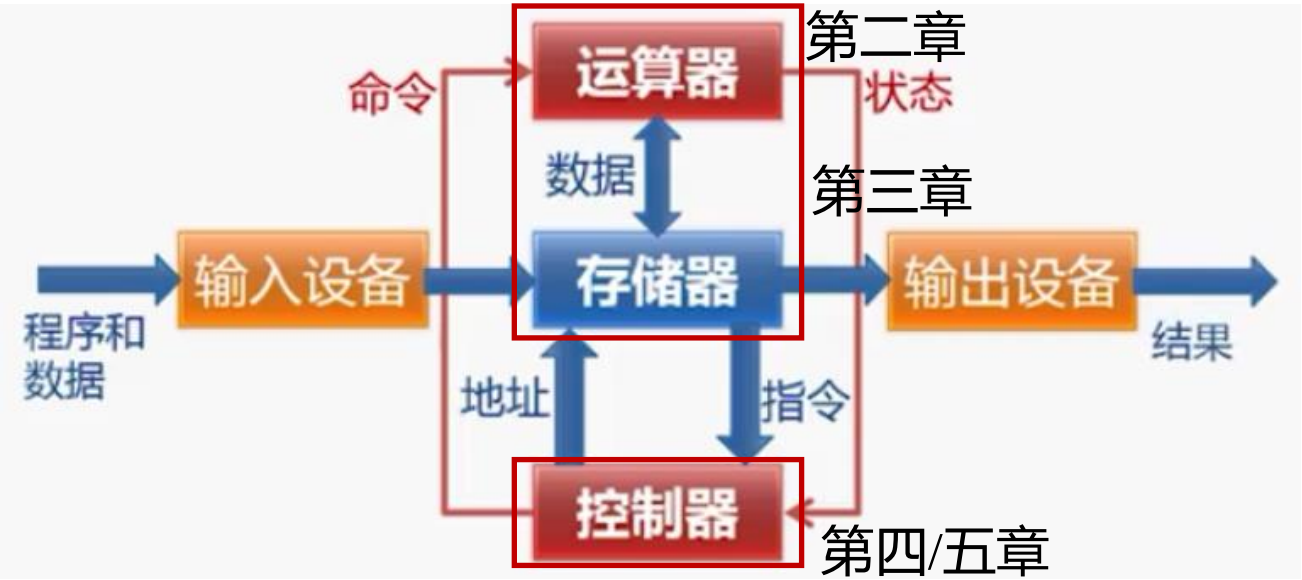
第四章 指令系统 (1)

吕昕晨

lvxinchen@bupt.edu.cn

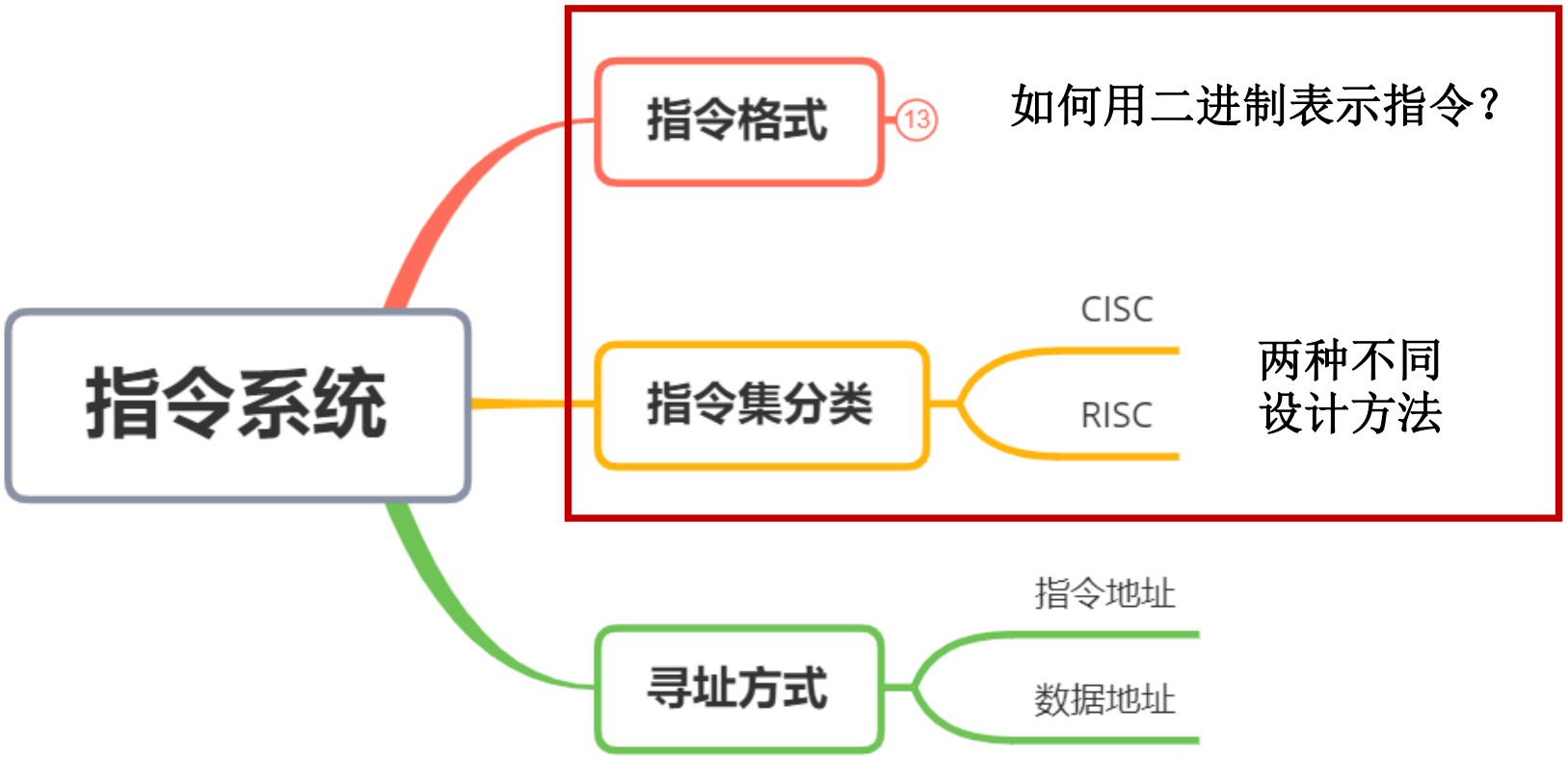
网络空间安全学院

冯诺依曼结构—控制器

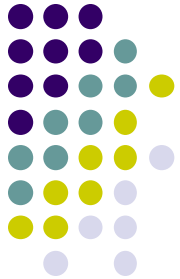


- 第四章 指令系统（机器指令）
- 第五章 中央处理机（微指令）

第四章安排——指令系统


指令系统
指令格式
13
如何用二进制表示指令？
指令集分类
CISC
RISC
两种不同
设计方法
寻址方式
指令地址
数据地址

第四章 指令系统



- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
- 典型指令集

机器指令/微指令

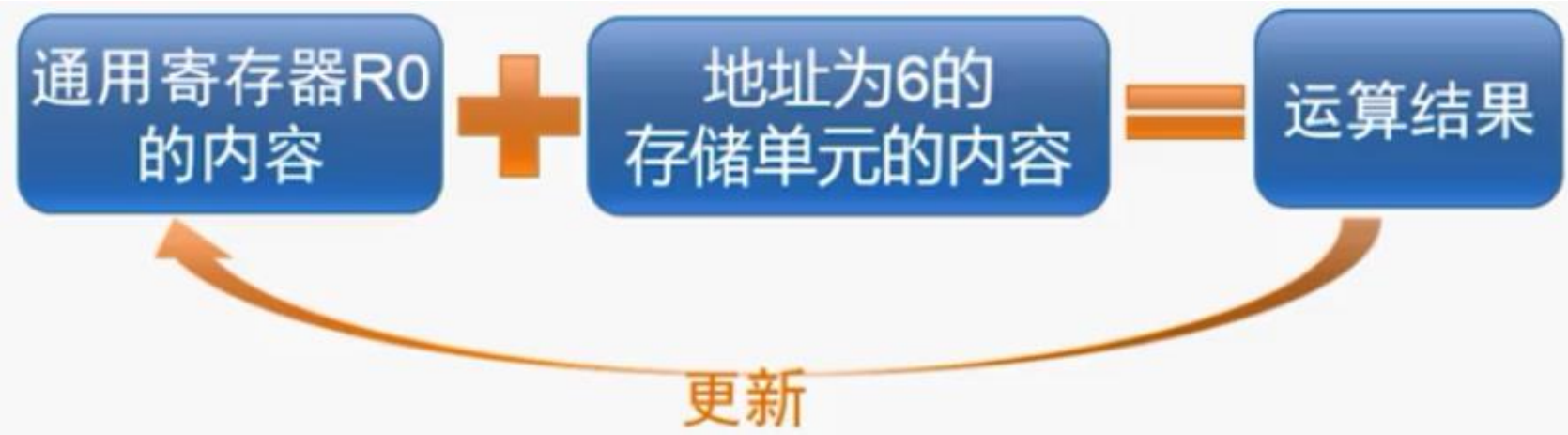
- 指令：计算机执行某种操作的命令
- 指令分类：
  - 微指令：微程序级的命令，属于硬件（第五章）
  - 机器指令：通常简称为指令（二进制序列），每一条指令可完成一个独立的算术运算或逻辑运算操作（第四章）
- 指令系统：一台计算机中所有机器指令的集合
  - 指令系统是表征一台计算机性能的重要因素
  - 它的格式与功能不仅直接影响到机器的硬件结构，而且也直接影响到系统软件，影响到机器的适用范围

指令示例

- 指令示例
  - 加法指令：ADD R0，[6]

如何用二进制表示指令？

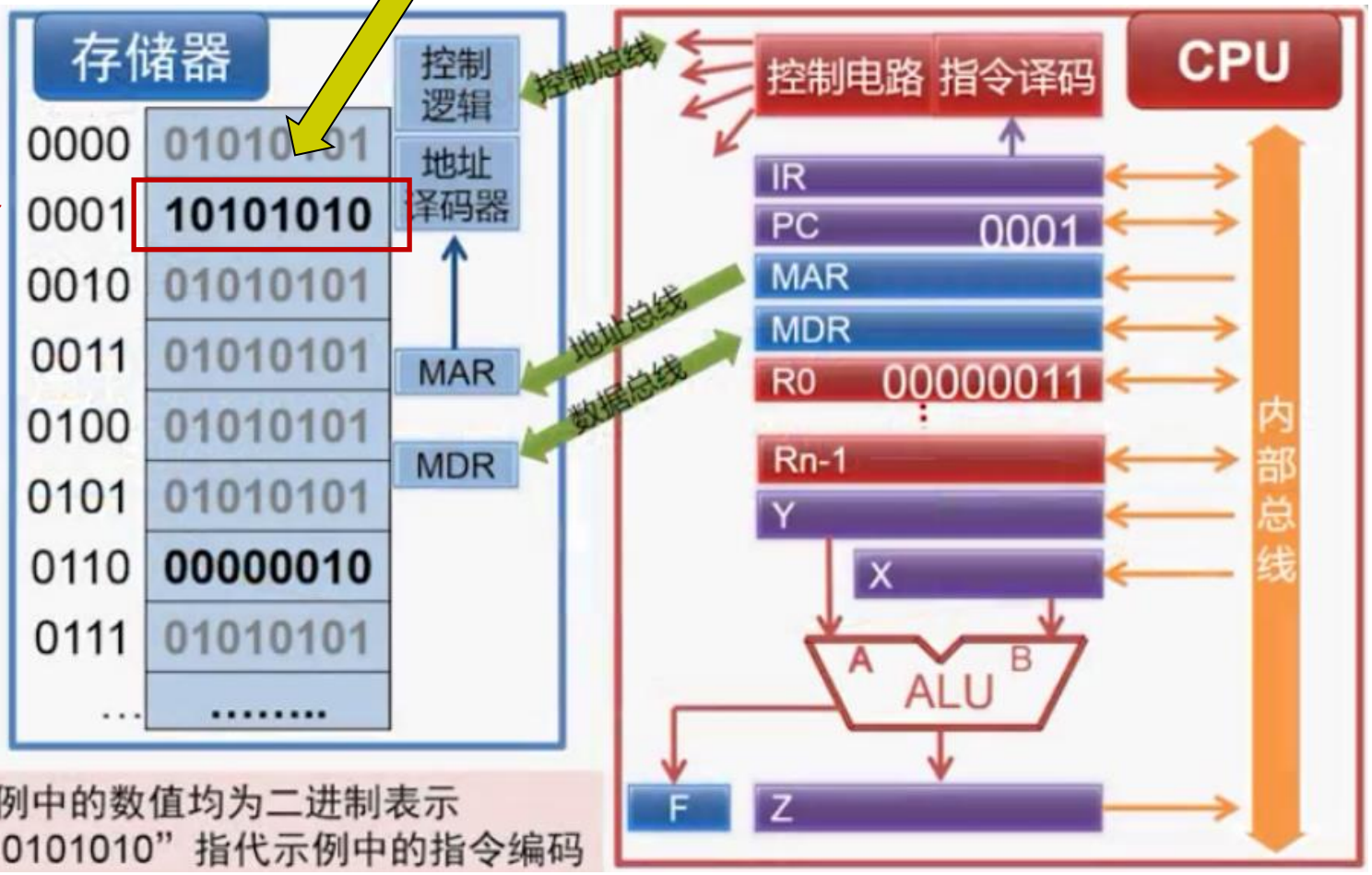
- 指令功能



指令示例

汇编语言：ADD R0，[6]

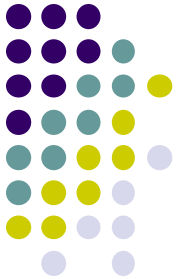
机器指令



注1：本例中的数值均为二进制表示

注2：“10101010”指代示例中的指令编码

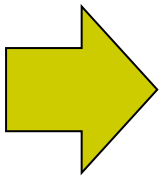
设计要求：简单指令系统



- 运算类指令
  - ADD R, M
    - 功能：将寄存器R内容与存储器M中内容相加后存入R
- 访存类指令
  - LOAD R, M
    - 功能：将存储器M中内容装入寄存器R
  - STORE M, R
    - 功能：将寄存器R的内容存入存储器M
- 转移类指令
  - JMP L
    - 功能：无条件跳转至标号L处

设计思路：编码方式

- 运算类指令
  - ADD R, M
- 访存类指令
  - LOAD R, M
  - STORE M, R
- 转移类指令
  - JMP L



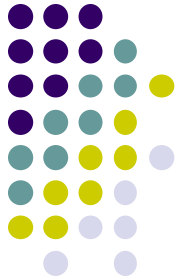
问题：指令的二进制表示

独立编码

1）指令含义（操作功能）

2）指令相关地址（操作对象）

常见指令格式

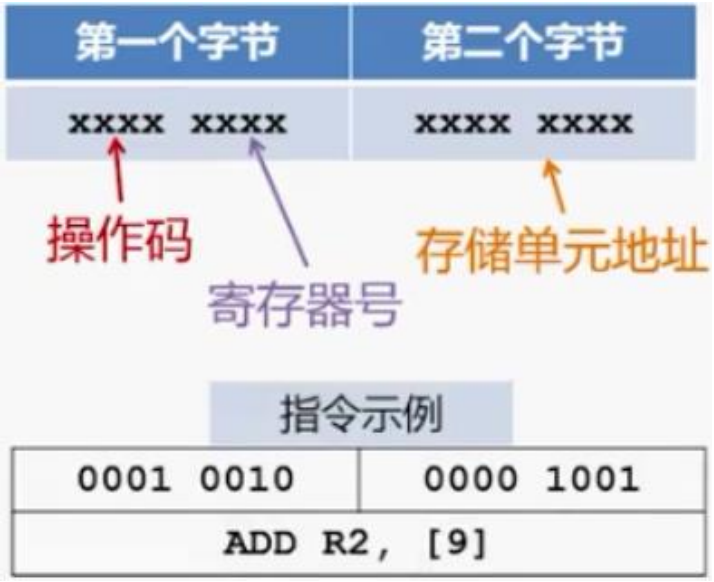



操作码（OP）
地址码(A)
一条指令的结构
用户要干什么？
停机中断
求反求补
加减乘除
…
对谁进行操作？
不需要操作对象
需要一个操作对象
需要两个操作对象
…

指令格式对应

- 指令长度
  - 等长指令，均为2字节
- 第一个字节高4位为操作码
  - LOAD：0000；ADD：0001
  - STORE：0010；JMP：0011
  - 可扩展至16条指令
- 第一个字节低4位为寄存器号
  - R0~R15，支持16个寄存器
- 第二个字节是存储单元地址
  - 存储器地址范围最大为256个字节

- 运算类指令
  - ADD R, M
- 访存类指令
  - LOAD R, M
  - STORE M, R
- 转移类指令
  - JMP L



指令示例

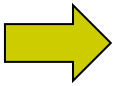
- 操作码表
  - LOAD：0000；ADD：0001
  - STORE：0010；JMP：0011
- 机器指令翻译
  - 机器语言：0000 0000 0000 0000
  - 汇编语言：LOAD R0, [0]

  - 机器语言： 0010 0011 0000 1111
  - 汇编语言： STORE R3, [15]

  - 机器语言： 0101 0000 1111 0110
  - 汇编语言：指令错误，操作码未定义

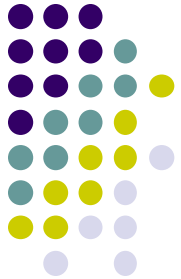
程序举例——简单与复杂指令集

- 任务说明
  - 将M1的内容与M2的内容相加后存入M3
  - 完成运算后，程序跳转到L处继续执行
  - M1-M3为存储单元地址
- 程序实现
  - LOAD R, M1
  - ADD R, M2
  - STORE R, M3

  ADD M3, M1, M2
  - JMP L
- 思考
  - 如果有汇编语言支持ADD M3, M1, M2?
  - 简单/复杂指令集优缺点?

第四章 指令系统



- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
- 典型指令集

指令体系结构

- **指令系统体系结构ISA**（Instruction Set Architecture）
- 机器语言程序员看到的计算机的属性，是与程序设计有关的计算机架构
  - 寄存器组织
  - 存储器的组织和寻址方式
  - I/O系统结构
  - 数据类型及其表示
  - 指令系统
  - 中断机制
  - 机器工作状态的定义及切换
  - 保护机制

思考题

辨析指令相关概念，说明其含义及表示范围关系

1）指令

2）指令系统

3）指令系统体系结构

指令分类：

微指令：微程序级的命令，属于硬件（第五章）

机器指令：通常简称为指令（二进制序列），每一条指令可完成一个独立的算术运算或逻辑运算操作（第四章）

指令系统：一台计算机中所有机器指令的集合

指令系统体系结构ISA：机器语言程序员看到的计算机的属性，是与程序设计有关的计算机架构

指令系统性能的要求

- **完备性**
  - 汇编语言编写各种程序时，指令系统直接提供的指令足够使用，而不必用软件来实现
  - 完备性要求指令系统丰富、功能齐全、使用方便
  - 一台计算机中最基本、必不可少的指令是不多的
  - 例如，乘除运算指令、浮点运算指令可直接用硬件来实现，也可用基本指令编写的程序来实现。采用硬件指令的目的是提高程序执行速度，便于用户编写程序。
- **有效性**
  - 有效性是指利用该指令系统所编写的程序能够高效率地运行
  - 高效率主要表现在程序占据存储空间小、执行速度快
  - 一般来说，一个功能更强、更完善的指令系统，必定有更好的有效性

指令系统性能的要求

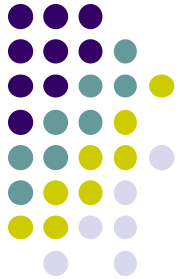
- **规整性：**对称性、匀齐性、指令格式和数据格式的一致性
  - 对称性：在指令系统中所有的寄存器和存储器单元都可同等对待，所有的指令都可使用各种寻址方式
  - 匀齐性：一种操作性质的指令可以支持各种数据类型，如算术运算指令可支持字节、字、双字整数的运算，十进制数运算和单、双精度浮点数运算等
  - 指令格式和数据格式的一致性：指令长度和数据长度有一定的关系，以方便处理和存取
  - 例如指令长度和数据长度通常是字节长度的整数倍（半字长）
- **兼容性**
  - 各机种之间具有相同的基本结构和共同的基本指令集，因而指令系统是兼容的，即各机种上基本软件可以通用
  - 但由于不同机种推出的时间不同，兼容性指低档机上运行的软件可以在高档机上运行

指令系统设计

- 指令系统要求
  - 完备性
  - 有效性
  - 规整性
  - 兼容性

- 问题1：规定指令格式，支持更多指令功能
  - 操作码、操作数、寄存器号、存储单元地址……
- 问题2：寻址方式
  - 如何给出下一条指令地址
  - 地址范围受地址码限制（8位地址码→256个字节）
  - 增加地址码，影响指令系统效率
  - 设计寻址方式，支持更大范围寻址

指令系统设计——指令格式（问题1）



- 问题1.1：常见指令格式，如何进行设计
  - 操作码、操作数、寄存器号、存储单元地址……
    - 指令明确指出操作含义操作类别/功能（加减乘除、访存、跳转）——操作码（MOV、ADD等）
    - 操作具体内容：操作数、寄存器、存储单元地址——地址码（R0、AX、DX等）
- 问题1.2：给定指令格式，分析指令功能
  - 支持操作数目、寻址范围、寄存器、整体类别
    - 给定指令格式，硬件、软件限制条件

第四章 指令系统

- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
  - 操作码
  - 地址码
  - 操作码扩展设计
  - 指令特征分析
- 典型指令集

操作码（1）

- 操作码：OP（Operation code）
  - 表示该指令应进行什么性质的操作
  - 如进行加法、减法、乘法、除法、取数、存数等
  - 不同的指令用操作码字段的不同编码来表示，每一种编码代表一种指令
- 组成操作码字段的位数一取决于计算机指令系统的规模
- 较大的指令系统就需要更多的位数
- n位（操作码）→2^n
  - 32条指令：5位操作码

操作码字段	地址码字段

操作码——指令数目

- 编码特征：n位（操作码）→2^n
  - 5位操作码→32条指令
  - 6位操作码→64条指令
  - 7位操作码→128条指令
  - 8位操作码→256条指令 ……
  - 问题——编码存在浪费：130条指令，8位操作码
- 解决思路
  - 允许浪费，指令长度不变——等长方法
  - 操作码变长，充分利用——变长方法（Huffman）

操作码（2）

- 等长操作码（一般情况下采用等长操作码）
  - 指令规整，译码简单
  - 例如IBM 370机
    - 该机字长32位，共有183条指令
    - 固定长度编码的主要缺点是：信息的冗余极大
- 不等长操作码/地址码（难点）
  - 充分利用指令长度
  - 适合于单片机等，指令字较短系统

单选题 1分

设置

某计算机指令系统共有70条指令，操作码长度至少为

A 5位

B 6位

C 7位

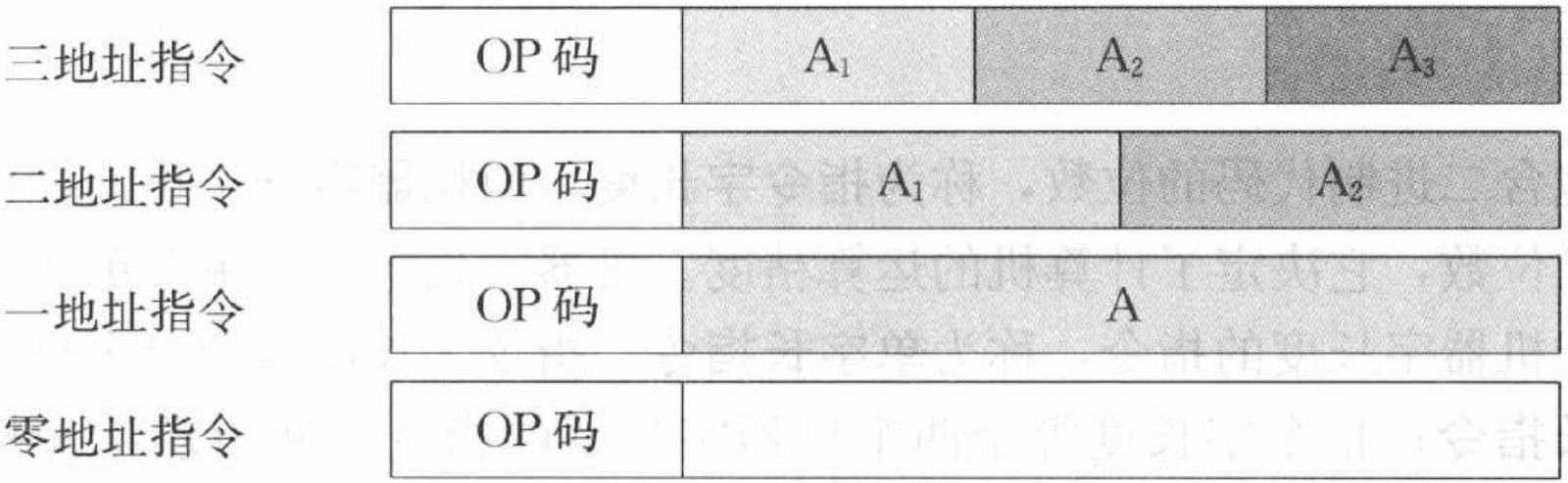
D 8位

第四章 指令系统

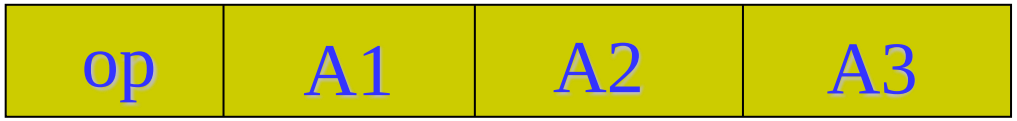
- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
  - 操作码
  - 地址码
  - 操作码扩展设计
  - 指令特征分析
- 典型指令集

指令分类

- 根据一条指令中有几个操作数地址，可将该指令称为几操作数指令或几地址指令
  - 三地址指令
  - 二地址指令
  - 单地址指令
  - 零地址指令



三地址指令



- 指令组成
  - 操作码op
  - 第一操作数A1/第二操作数A2 、结果A3
- 指令功能
  - (A1) op (A2)→A3
  - (PC) +1→PC
- 特点
  - 指令长度仍比较长，通常不用于微型机
  - A1/A2/A3通常为寄存器，加快指令执行速度

ADD R0,R1,R2
R0=R1+R2

二地址指令（1）

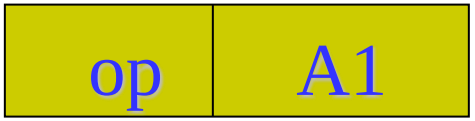
- 指令组成
  - 操作码op
  - 第一操作数A1、第二操作数A2
- 指令功能
  - (A1) op (A2)→A1
  - (PC)+1→PC
- 特点
  - 二地址指令在计算机中得到了广泛的应用，可支持访存

ADD R0, [6]

二地址指令（2）

- 根据操作数的物理位置分为：
  - 存储器-存储器（SS）
    - 参与操作的数均存放在内存里，需要多次访问内存
    - 速度慢
  - 寄存器-存储器类型（RS）
    - 参与操作的数存放在内存与寄存器内，一般需要一次访存
    - 速度较慢
  - 寄存器-寄存器类型（RR）
    - 参与操作的数均存放在寄存器中，不需要访问内存
    - 速度快

一地址指令



- 指令组成
  - 操作码op
  - 操作数A
- 指令功能
  - (AC) op (A) →AC / C op (A) →A ……
  - (PC)+1→PC
  - AC表示累加寄存器AC中的数、C代表常数
- 特点
  - 单操作数运算指令，+1、-1、取反
  - 指令中给出一个源操作数的地址，另一操作数隐含

INC BH

零地址指令

- 指令组成
  - 操作码op
- 特点
  - “停机”、“空操作”、“清除”等控制类指令

HLT、CLC

指令地址分类总结

三地址指令

OP	A_1	A_2	A_3（结果）

指令含义：$(A_1)OP(A_2)\rightarrow A_3$

二地址指令

OP	A_1（目的操作数）	A_2（源操作数）

指令含义：$(A_1)OP(A_2)\rightarrow A_1$

一地址指令

OP	A_1

指令含义：1. $OP(A_1)\rightarrow A_1$，如加1、减1、取反、求补等

2. $(ACC)OP(A_1)\rightarrow ACC$，隐含约定的目的地址为ACC

零地址指令

OP

指令含义：1. 不需要操作数，如空操作、停机、关中断等指令

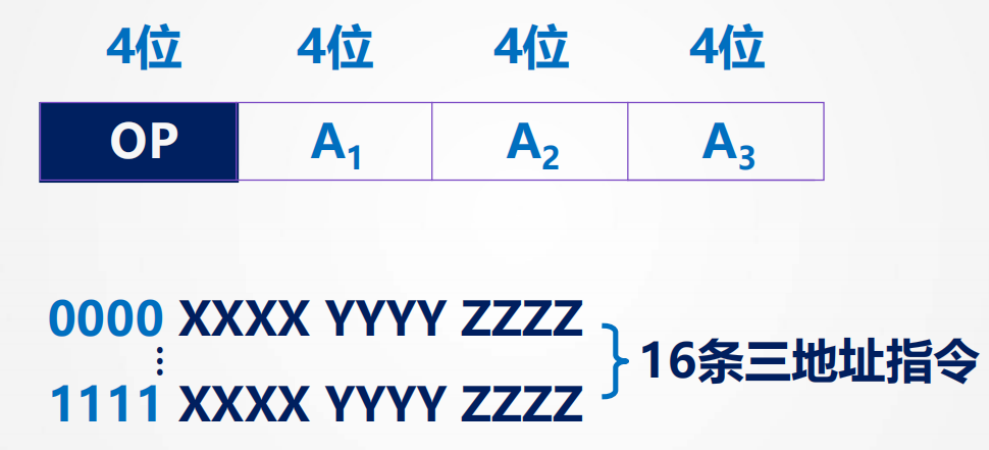
2. 堆栈计算机，两个操作数隐含存放在栈顶和次栈顶，计算结果压回栈顶

第四章 指令系统

- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
  - 操作码
  - 地址码
  - 操作码扩展设计（变长OP）
  - 指令特征分析
- 典型指令集

操作码扩展示例（变长OP）

定长情况



- 问题：如何设计变长OP（不改变指令长度，地址码4位）
  - 需求1：15条三地址+15条二地址+15条一地址+16条零地址
  - 需求2：15条三地址+14条二地址+31条一地址+16条零地址

操作码扩展设计原则

- 不允许短码是长码的前缀，即短操作码不能与长操作码的前面部分的代码相同

- 扩展：15条三地址，留下1个**码点1111**用于指令扩展
- 类比：哈夫曼编码（对使用频率高的指令分配较短的操作码，尽可能提升指令译码效率）

操作码扩展设计—需求1

- 需求1：15条三地址+15条二地址+15条一地址+16条零地址

- 扩展码点：1111 1111

操作码扩展设计—需求1

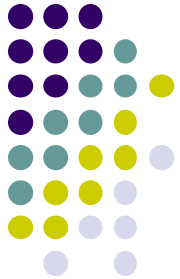
- 需求1：15条三地址+15条二地址+15条一地址+16条零地址

- 扩展码点：1111 1111 1111

操作码扩展设计—需求1

- 需求1：15条三地址+15条二地址+15条一地址+16条零地址

操作码扩展设计—需求2



- 需求2：15条三地址+14条二地址+31条一地址+16条零地址

操作码扩展设计—需求2

- 需求2：15条三地址+14条二地址+31条一地址+16条零地址

定长操作码

某指令长度为20位，具有双操作数、单操作数、无操作数三类指令，操作数地址6位、操作码长度8位，已设计出m条双操作数指令，n条无操作数指令，请问单操作数指令最多为?

A 128

B 64

C 256-m-n

D 128-m

- 编码特征：n位（操作码）→2^n
  - 5位操作码→32条指令
  - 6位操作码→64条指令
  - 7位操作码→128条指令
  - 8位操作码→256条指令 ……

变长操作码

某指令系统的指令字长固定为16位，单个操作数地址4位。已知该系统支持15条三地址指令、12条二地址指令、62条一地址指令，请问该指令系统最多能支持多少条零地址指令?

A 16

B 32

C 64

D 128

求解原则：设地址长度为n，上一层留出m种状态，下一层可扩展出m×2^n种状态

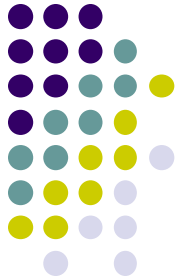
a) 有15条三地址指令
共2^4=16种状态
留出16-15=1种

b) 有12条二地址指令
共1 ×2^4=16种
留出16-12=4种

c) 有62条一地址指令
共4 ×2^4=64种
留出64-62=2种

d) 有32条零地址指令
共2 ×2^4=32种

第四章 指令系统



- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
  - 操作码
  - 地址码
  - 操作码扩展设计
  - 指令特征分析
- 典型指令集

指令字长度

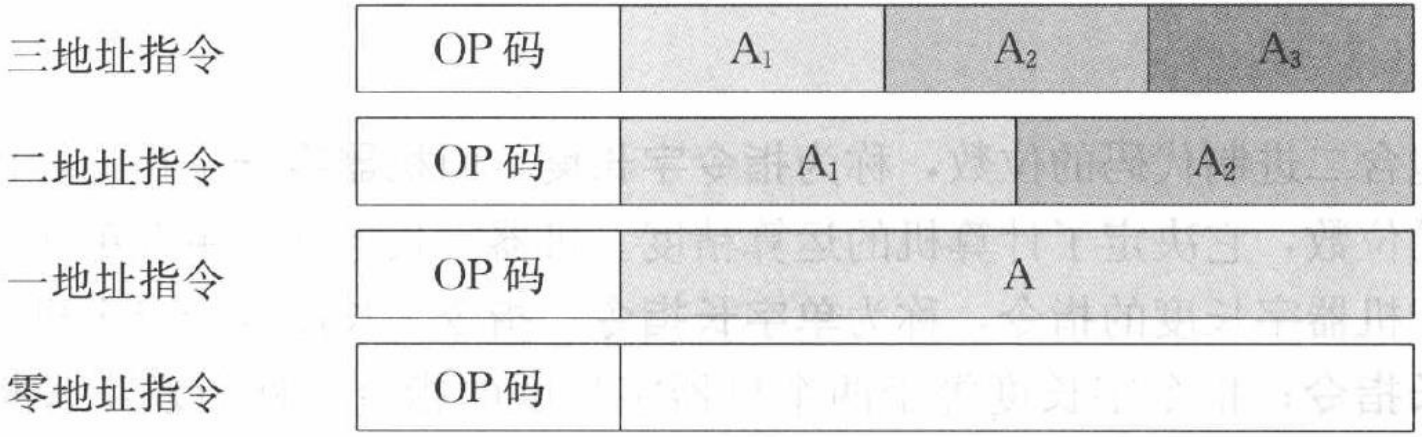
- 基本概念
  - 指令字长度（一个指令字包含二进制代码的位数）
  - 机器字长：计算机能直接处理的二进制数据的位数
- 指令字长度
  - 半字长指令、单字长指令、双字长指令
- 指令字长可变
- 例如，IBM370系列（32位）
  - 16位（半字长）
  - 32位（单字长）
  - 48位（一个半字长）

指令字长度

- 多字长指令
  - 优点：提供足够的地址位来解决访问内存任何单元的寻址问题
  - 缺点：必须两次或多次访问内存以取出一整条指令，降低了CPU的运算速度，又占用了更多的存储空间
- 等长指令系统
  - RISC：各种指令字长度是相等的，指令字结构简单，且指令字长度是不变的
- 变长指令系统
  - CISC：各种指令字长度随指令功能而异，结构灵活，能充分利用指令长度，但指令的控制较复杂

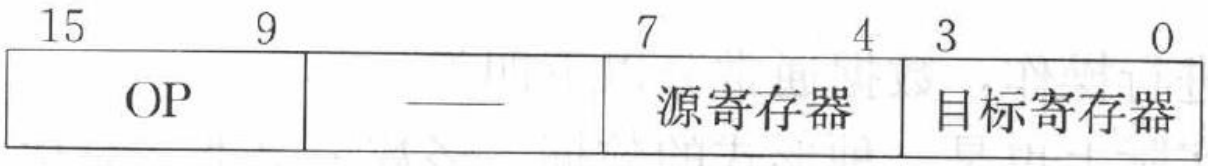
指令分析总结（重要）

- 给定指令格式分析指令特点
  - 指令字长度：半字长、单字长、双字长
  - 指令类型：零地址、一地址、二地址（RR、RS、SS）、三地址
  - 支持指令数目：OP码位数→2^n
  - 寄存器数目：寄存器位数→2^n
  - 寻址方式（后续内容）



指令格式例题（1）

- 例1 16位字长，指令格式如下，分析指令格式特点

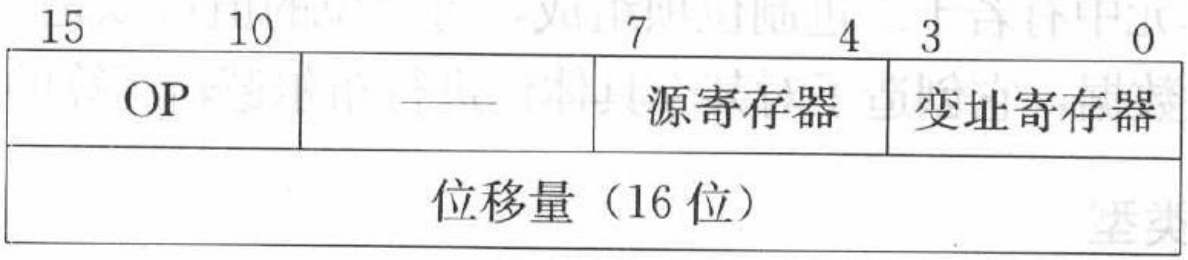


- 指令长度：单字长（16位）二地址指令
- 操作码支持：操作码字段（7位）可支持128条指令
- 指令类型：RR型指令
- 寄存器：寄存器数目：16个

主观题 10分

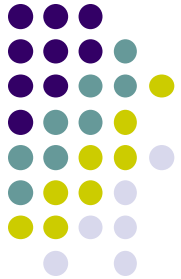
设置

16位字长，指令格式如下，分析指令格式特点



- 双字长二地址指令
- 操作码字段（6位）：可支持64条指令
- RS型指令
- 寄存器数目：16个/存储器（变址寄存器+位移量）

第四章 指令系统



- 指令与简单指令系统
- 指令系统设计要求
- 指令格式分类
- 典型指令集

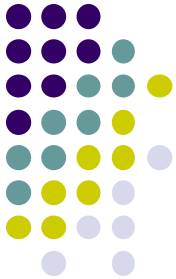
指令系统分类

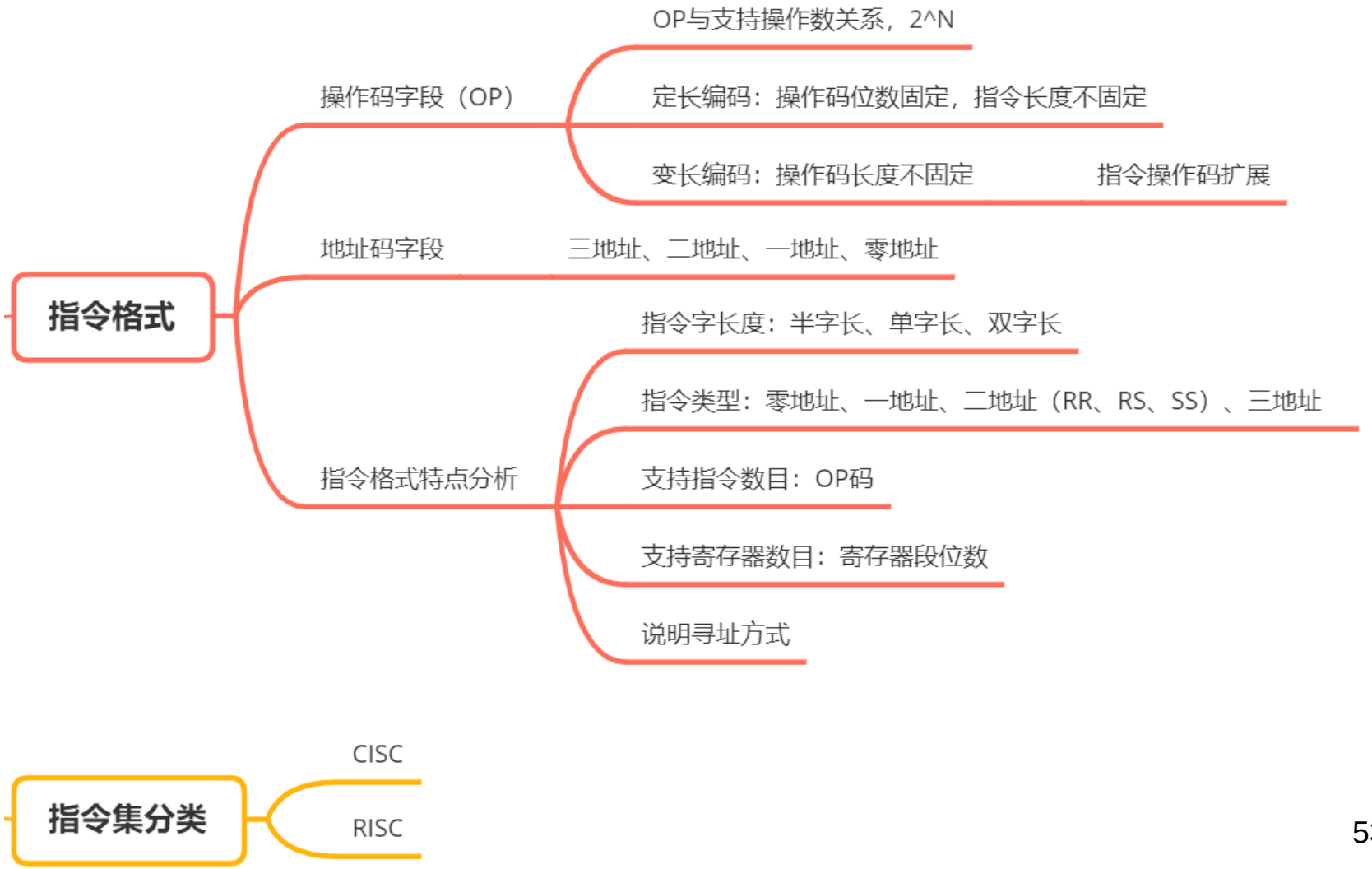
- **复杂指令系统计算机**：简称CISC（典型：x86）
  - 庞大的指令系统，指令变长
  - 译码较复杂，指令复杂度差别大
  - 计算机的研制周期变长，难以保证正确性，不易调试维护，而且由于采用了大量使用频率很低的复杂指令而造成硬件资源浪费
- **精简指令系统计算机**：简称RISC（典型：ARM/MIPS）
  - 2-8定律：20%的指令占程序80%的比例
  - 便于VLSI技术实现的精简指令系统计算机
  - 指令定长，复杂度相近，便于流水线技术实现

CISC-RISC对比

类别 / 对比项目	CISC	RISC
指令系统	复杂，庞大	简单，精简
指令数目	一般大于200条	一般小于100条
指令字长	不固定	定长
可访存指令	不加限制	只有Load/Store指令
各种指令执行时间	相差较大	绝大多数在一个周期内完成
各种指令使用频度	相差很大	都比较常用
通用寄存器数量	较少	多
目标代码	难以用优化编译生成高效的目标代码程序	采用优化的编译程序，生成代码较为高效
控制方式	绝大多数为微程序控制	绝大多数为组合逻辑控制
指令流水线	可以通过一定方式实现	必须实现

总结




指令格式
操作码字段（OP）
OP与支持操作数关系，2^N
定长编码：操作码位数固定，指令长度不固定
变长编码：操作码长度不固定
指令操作码扩展
地址码字段
三地址、二地址、一地址、零地址
指令格式特点分析
指令字长度：半字长、单字长、双字长
指令类型：零地址、一地址、二地址（RR、RS、SS）、三地址
支持指令数目：OP码
支持寄存器数目：寄存器段位数
说明寻址方式

指令集分类
CISC
RISC