

计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第21次课：指令系统-上

杜国栋

信息科学与工程学院计算机科学与工程系

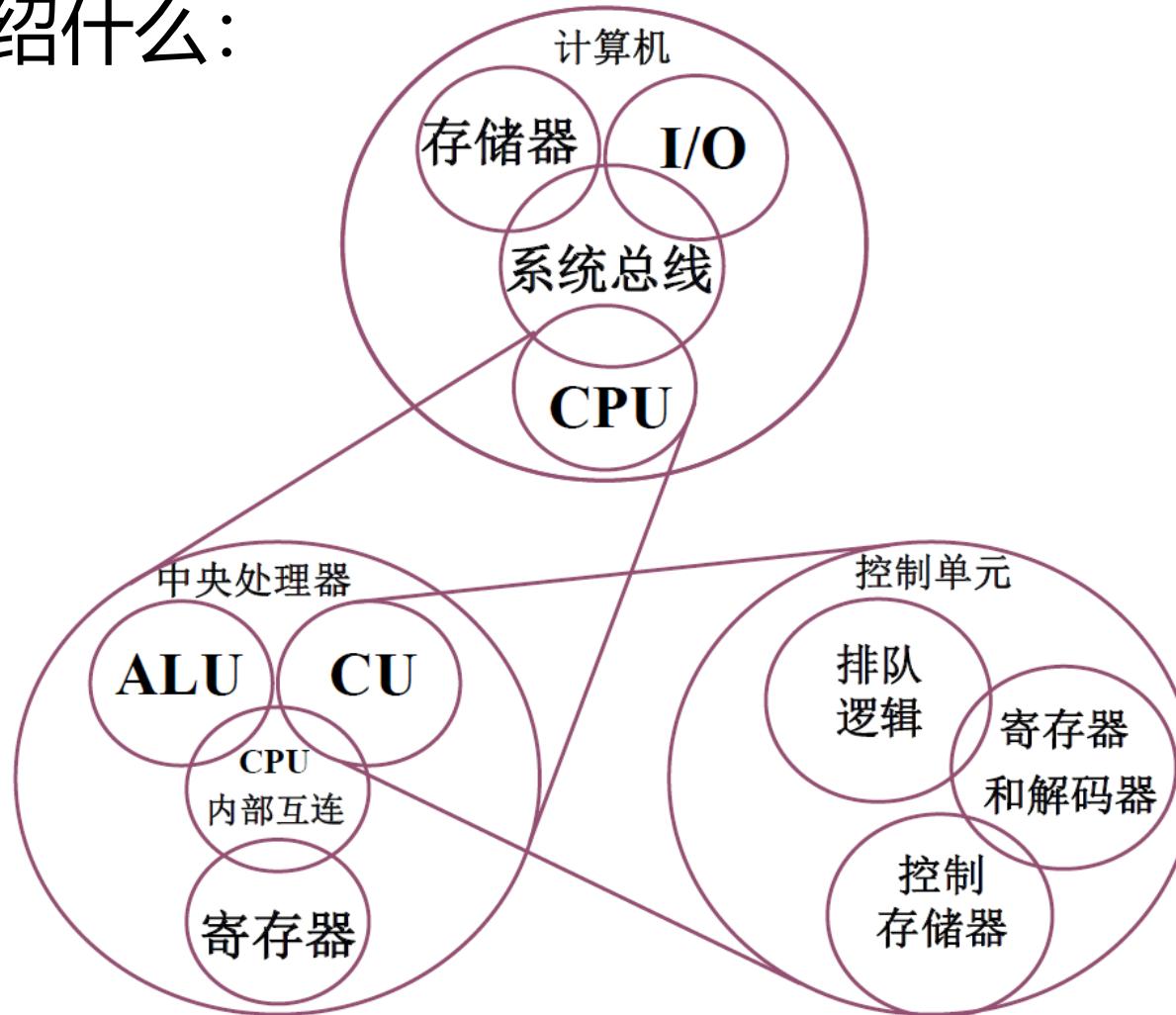
gddu@ysu.edu.cn



燕山大学
YANSHAN UNIVERSITY



本部分将介绍什么：





快手极速版

作者:老K的无聊杂谈

(1)

华为 中国企业之光?

K

前面我们讲过智能系统的诞生

剩余 04:35

快手极速版

作者:老K的无聊杂谈

(2)

鸿蒙系统开源 其他厂商却无法使用?

K

鸿蒙系统

剩余 04:16

快手极速版

作者:老K的无聊杂谈

(3)

鸿蒙系统助俄罗斯 摆脱通信困局?

K

在明确了开源概念后

剩余 03:36





设计CPU的一般过程：

指令系统→ 数据通路→ 控制器→ CPU定型

◎ 指令：*instruction*，计算机执行某类操作的信息的集合，是CPU工作的主要依据。

◎ 指令集：*instruction set*，处理器能执行的全体指令的集合（CISC、RISC）

- √ 决定了计算机的硬件功能
- √ 计算机中软硬件的分界面



1、CISC(Complex Instruction Set Computing) 复杂指令集计算

早期计算机部件昂贵、速度慢，为了扩展硬件功能，不得不将更多更复杂指令加入到指令系统，以提高计算机的处理能力 → 复杂指令集

2、RISC(Reduced Instruction Set Computing) 精简指令集计算

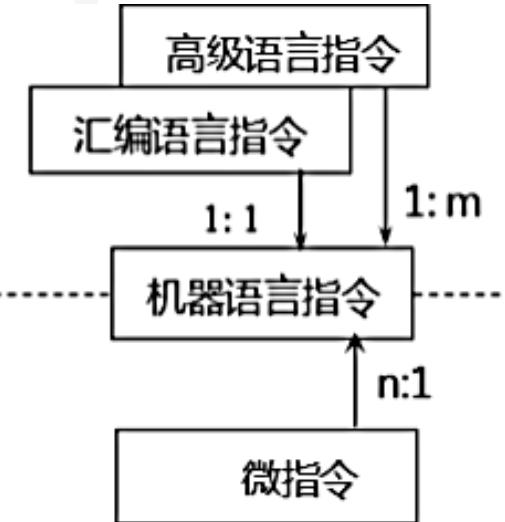
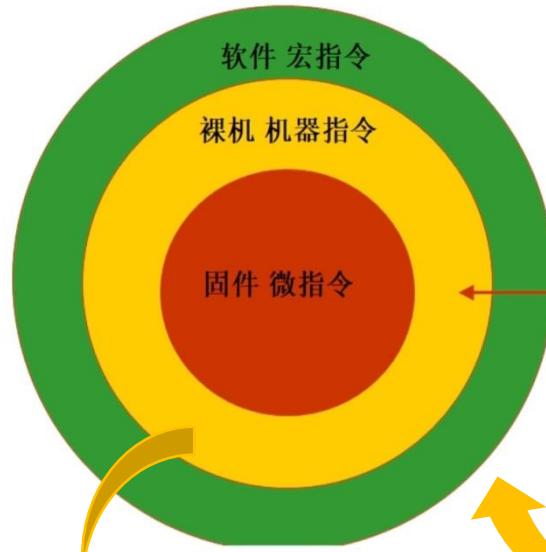
随着半导体技术进步，80年代开始逐渐直接通过硬件方式，而不是扩充指令来实现复杂功能，指令规模逐渐缩小、指令进一步简化 → 精简指令集

简单指令(约占20%的)约占80%的使用频率，复杂指令(约占80%)只占大约20%的使用频率。

下列几项中，不符合RISC指令系统的特点是（ ）。

- A 指令长度固定，指令种类少
- B 指令功能尽可能强
- C 增加寄存器的数目，以尽量减少访存次数
- D 选取使用频率最高的一些简单指令，以及很有用但不复杂的指令

 提交



什么是机器指令？

能够被机器直接理解并执行的语言-机器语言

机器语言的语句-机器指令

什么是指令系统？

机器能够理解的全部机器指令的集合称为该机器的指令系统

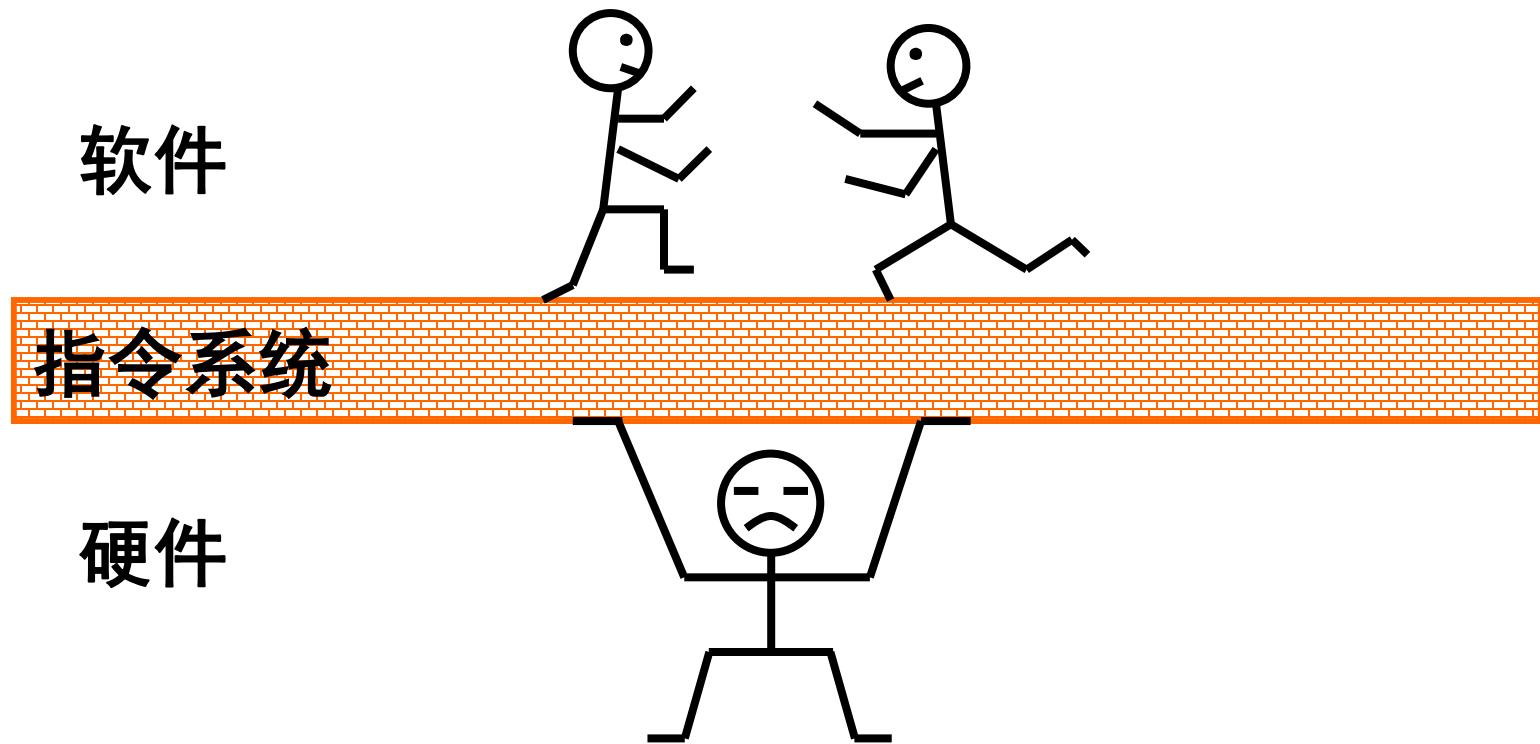
指令系统-软/硬件界面 反映机器的功能 不同CPU有所不同

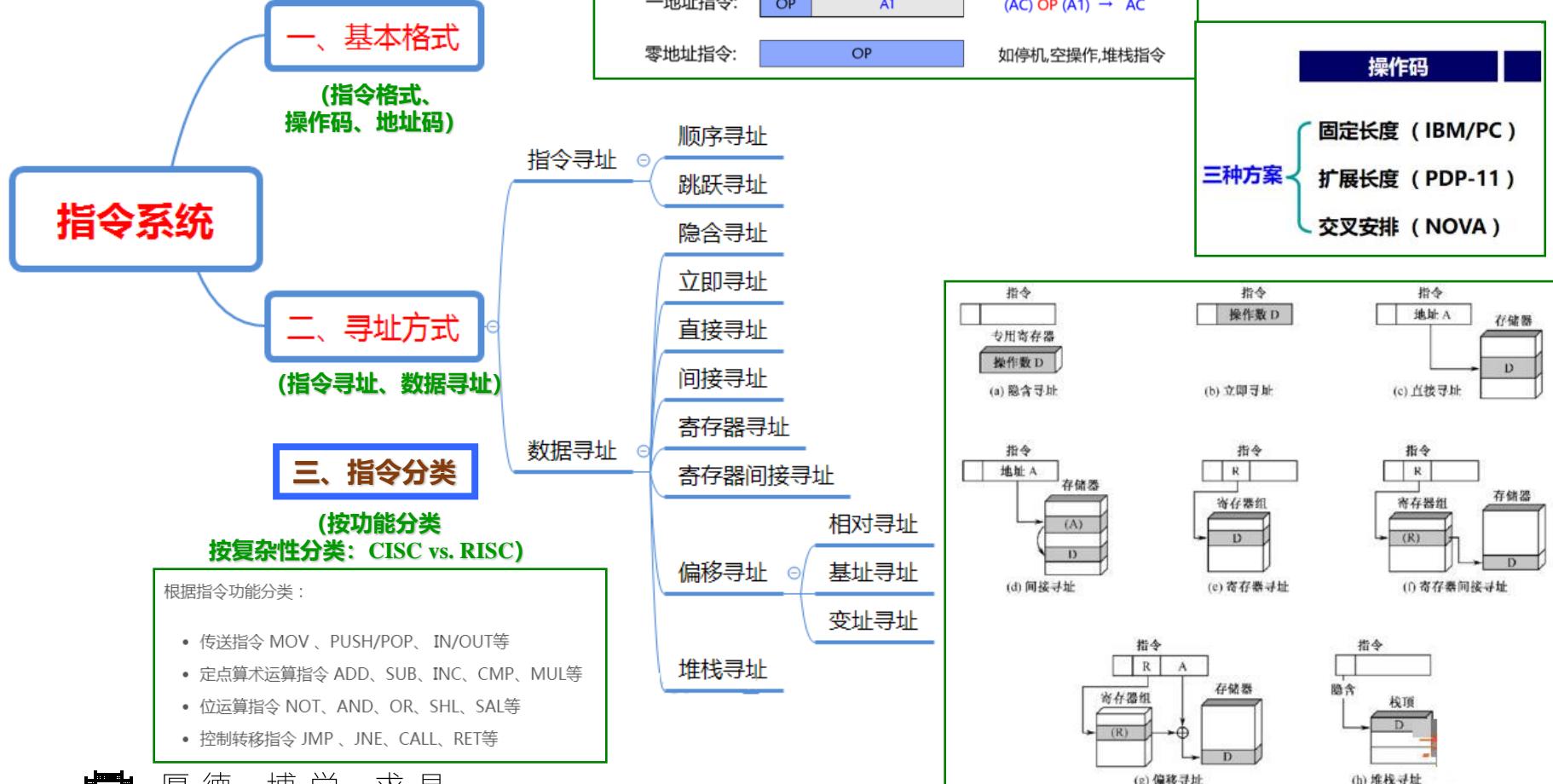
厚德·博学·求是

芯片架构	运营者	特点	代表厂商
X86	Intel公司	功能强大、通用性、兼容性、与实用性	Intel、AMD
ARM	Acorn公司	低功耗、低成本	苹果、谷歌、IBM、华为
RISC-V	RISC-V基金会	完全开源、架构简单、易于移植、模块化设计、完整的工具链	三星、英伟达、西部数据
MIPS	MIPS公司	简洁、优化、具有高度扩展性、包含大量的寄存器、指令数和字符、可视的管道延时隙	龙芯中科



指令系统在计算机中的地位：







课程目标

- 掌握指令操作码的扩展技术；
- 熟悉指令格式的分类；
- 了解指令格式和指令长度与字长的关系。





指令系统

➤ 一个完善的指令系统应满足如下四方面的要求：

➤ **完备性**

指令够用，指令系统丰富，功能齐全，不必用软件实现；

➤ **有效性**

高效率运行，占用存储空间小，执行速度快；

➤ **规存性**

对称性，匀齐性，指令格式与数据格式的一致性；

➤ **兼容性**

向上兼容，低档机上的软件可以在高档机上运行。





指令格式

- 一条指令包含的信息：
 - 操作码
 - 操作数的地址
 - 操作结果的存储地址
 - 下一条指令的地址

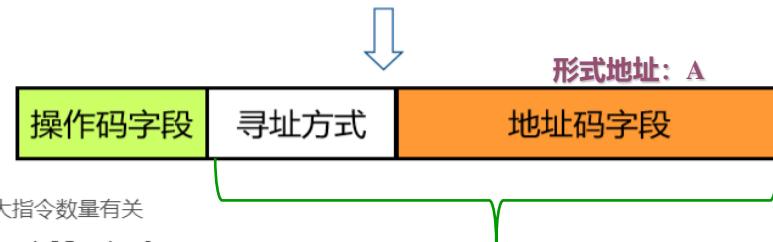




指令格式

■ 用二进制代码表示指令的结构形式

- ◆ 指令要求计算机完成什么功能?
(表征指令的操作特性与功能) → 设置操作码
- ◆ 指令要求计算机处理什么数据?
(指定参与操作的操作数地址) → 设置数据源/目
- ◆ 计算机怎样得到要处理的数据?
(寻找当前参与运算的操作数) → 设置寻址方式



- 操作码字段的位数与支持的最大指令数量有关
 - 对于定长操作码而言, $\text{LengthOP} = \lceil \log_2 n \rceil$
 - 支持变长操作码时, 操作码向不用的地址码字段扩展
- 寻址方式字段的位数与支持的寻址方式种类有关
- 地址码字段的作用及影响与其位数和寻址方式有关





指令格式-地址码

✓ 零地址结构指令

θ

功能：①用于处理机的特殊控制（如HLT, NOP）。
②针对隐含约定的寄存器，如返回指令：

RST



隐含操作：(SP)→PC; SP⁺ → SP;

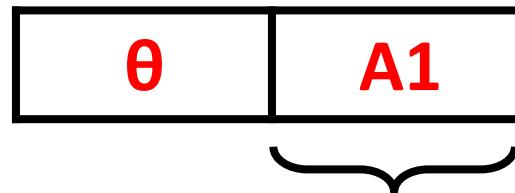
把堆栈栈顶单元保存的返回地址打入PC。





指令格式-地址码

✓ 一地址结构指令



双操作数: $A1 \cup [PC]_H \rightarrow PC$ 如: J addr

单操作数:

$\theta(A1) \rightarrow A1$

$(PC)^+ \rightarrow PC$ 如: INC R0

隐含约定



自动修改PC的内容，使PC指向下一条指令



厚德·博学·求是



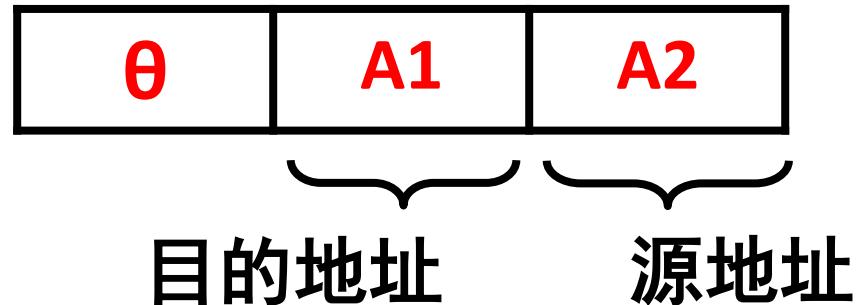
指令格式-地址码

✓ 二地址结构指令

功能：

(A1) θ (A2) → A1

$(PC)^+ \rightarrow PC$



如： ADD R1, R0

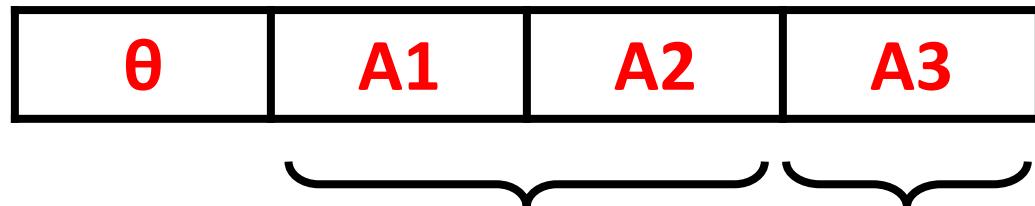
自动修改PC的内容，使PC指向下一条指令





指令格式-地址码

✓ 三地址结构指令



功能：

(A1) θ (A2) \rightarrow A3

$(PC)^+ \rightarrow PC$

如：ADD rd, rs, rt



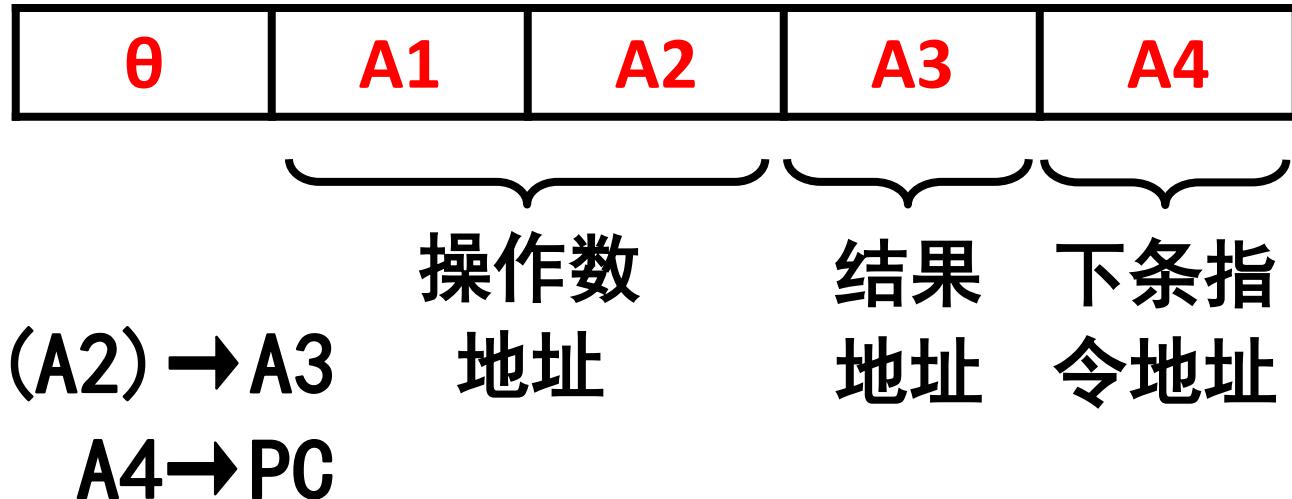
自动修改PC的内容，使PC指向下一条指令





指令格式-地址码

✓ 四地址结构指令



功能：(A1) θ (A2) → A3

操作数地址 **结果地址** **下条指令地址**

A4 → PC

一般用PC寄存器指示下条指令的地址。

四地址结构指令在RISC中很少会使用。



指令格式-操作码

- 操作码字段的位数取决于指令系统的规模；

- 操作码的类型：

- 固定长度的操作码

操作码字段为4位，
则指令系统中的指
令数目为 $2^4=16$ 条。

- ◆ 特征：所有指令长度均相同。

- ◆ 优点：控制简单，速度快，适用于指令条数不多的

- 可变长度的操作码

- ◆ 特征：频繁使用的指令用位数较少的操作码；

- 不常使用的指令可利用操作码扩展技术进行扩展；

- ◆ 优点：充分利用软硬件资源，适用于大规模的指令系统。





指令格式-操作码的扩展技术

设某指令长**16位**, 包括**4位基本操作码字段**和**3个4位地址码**

字段。

OP	A1	A2	A3
----	----	----	----

① 若全是三地址指令, 则最多能有多少条指令?

② 若三地址指令需**15条** —

两地址指令需**15条** — 应如何安排?

单地址指令需**15条** —

零地址指令需**16条** —





指令格式-操作码的扩展技术

设某指令长16位，包括4位基本操作码字段和3个4位地址码字段。

OP	A1	A2	A3
----	----	----	----

① 若全是三地址指令，则最多能有多少条指令？

○ 操作码为4位的，则指令条数为 $2^4=16$ 。

② 若三地址指令需15条 —

两地址指令需15条 | 应如何安排？

单地址指令需15条 |

零地址指令需16条 —

○ 可使用操作码扩展技术，缩短固定操作码长度；

1. 不允许短码是长码的前缀
2. 各指令的操作码一定不能重复

4位操作码

OP	A ₁	A ₂	A ₃
0000	A ₁	A ₂	A ₃
0001	A ₁	A ₂	A ₃
⋮	⋮	⋮	⋮
1110	A ₁	A ₂	A ₃

最多15条三地址指令

8位操作码

1111	0000	A ₂	A ₃
1111	0001	A ₂	A ₃
⋮	⋮	⋮	⋮
1111	1110	A ₂	A ₃

最多15条二地址指令

12位操作码

1111	1111	0000	A ₃
1111	1111	0001	A ₃
⋮	⋮	⋮	⋮
1111	1111	1110	A ₃

最多15条一地址指令

16位操作码

1111	1111	1111	0000
1111	1111	1111	0001
⋮	⋮	⋮	⋮
1111	1111	1111	1111

16条零地址指令





指令格式-操作码的扩展技术

4 位操作码

OP	A ₁	A ₂	A ₃
0000	A ₁	A ₂	A ₃
0001	A ₁	A ₂	A ₃
⋮	⋮	⋮	⋮
1110	A ₁	A ₂	A ₃

8 位操作码

1111	0000	A ₂	A ₃
1111	0001	A ₂	A ₃
⋮	⋮	⋮	⋮
1111	1110	A ₂	A ₃

12 位操作码

1111	1111	0000	A ₃
1111	1111	0001	A ₃
⋮	⋮	⋮	⋮
1111	1111	1110	A ₃

16 位操作码

1111	1111	1111	0000
1111	1111	1111	0001
⋮	⋮	⋮	⋮
1111	1111	1111	1111

三地址指令操作码
每减少一种最多可多构成
 2^4 种二地址指令

二地址指令操作码
每减少一种最多可多
构成 2^4 种一地址指令





基于霍夫曼编码原理设计变长操作码

表 5.1 霍夫曼编码的操作码

指 令	使用频度	A 方案		B 方案	
		操作码	长度	操作码	长度
I_1	0.45	0	1 位	0	1 位
I_2	0.30	10	2 位	10	2 位
I_3	0.15	110	3 位	110	3 位
I_4	0.05	1110	4 位	11100	5 位
I_5	0.03	11110	5 位	11101	5 位
I_6	0.01	111110	6 位	11110	5 位
I_7	0.01	111111	6 位	11111	5 位





指令字长

- 指令字长决定于
 - 操作码的长度
 - 操作数地址的长度
 - 操作数地址的个数
- 指令字长 固定 指令字长=存储字长
- 指令字长 可变 指令节的倍数变化



扩展操作码是（ ）。

- A 操作码字段以外的辅助操作字段的代码
- B 指令格式中不同字段设置的操作码
- C 一种指令优化技术，即让操作码的长度随地址数的减少而增加，不同地址数的指令可以具有不同的操作码长度。
- D 指扩大操作码字段的位数。

提交

在堆栈数据结构及堆栈存储结构的计算机中，所需的操作数默认在堆栈内，因此，入栈和出栈操作常用（ ）。

- A 零地址指令格式
- B 一地址指令格式
- C 二地址指令格式
- D 三地址指令格式

 提交



假设某计算机的指令长度为16位，每个操作数的地址码长6位，指令分为无操作数、单操作数和双操作数3类。若双操作数指令有 m 条，无操作数指令 n 条，问单操作数指令最多可能有多少条？





假设某计算机的指令长度为16位，每个操作数的地址码长6位，指令分为无操作数、单操作数和双操作数3类。若双操作数指令有m条，无操作数指令n条，问单操作数指令最多可能有多少条？

解：对于双操作数，操作码长度为 $16-2 \times 6=4$ 位；

已知双操作数已有m种，所以留有 $(2^4-m) = 16-m$ 个编码给单操作数。

对于单操作数，操作码长度为 $16-6=10$ 位，可扩展位为6位；

已知无操作数已有n种，设单操作数有Y种，那么有：

$$n = [(2^4-m) \times 2^6 - Y] \times 2^6$$

$$Y = (2^4-m) \times 2^6 - n / (2^6)$$





总结

- 当用一些硬件资源代替指令字中的地址码字段后
 - 可扩大指令的寻址范围
 - 可缩短指令字长
 - 可减少访存次数
- 当指令的地址字段为寄存器时
 - 三地址 OP R1, R2, R3
 - 二地址 OP R1, R2
 - 一地址 OP R1
 - 可缩短指令字长
 - 指令执行阶段不访存





有问题欢迎随时跟我讨论

办公地点：西校区信息馆504

邮 箱：gddu@ysu.edu.cn

