4 指令系统

提纲

北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

4.1 指令系统的发展与性能要求

提纲

4.1.1 指令在计算机系统中的地位

- 指令：就是要计算机执行某种操作的命令。

- 是软件和硬件分界面的一个主要标志
  - 硬件设计人员采用各种手段实现它；
  - 软件设计人员则利用它编制各种各样的系统软件和应用软件

- 也是硬件设计人员和软件设计人员之间沟通的桥梁。

4.1.2 指令系统基本概念

- 从计算机组成的层次结构来说，计算机的指令有微指令、机器指令和宏指令之分
- 微指令：是微程序级的命令，它属于硬件
- 宏指令：由若干条机器指令组成的软件指令，它属于软件
- 机器指令：介于微指令与宏指令之间，通常简称为指令，每一条指令可完成一个独立的算术运算或逻辑运算操作
- 本章所讨论的指令，是机器指令

- 一台计算机中所有机器指令的集合，称为这台计算机的指令系统
  - 指令集体系结构，简称ISA（Instruction Set Architecture）
- 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

4.1.3 发展情况

- 复杂指令系统计算机，简称CISC（Complex Instruction Set Computer）
  - 计算机的指令系统比较丰富，有专用指令来完成特定的功能，因此，处理特殊任务效率较高，例如：AES-NI指令
  - 桌面计算机流行的x86体系结构是CISC架构
  - 但是如此庞大的指令系统不但使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（Reduced Instruction Set Computer）
  - 人们又提出了便于用超大规模集成电路技术实现的精简指令系统计算机
  - 移动终端流行的ARM体系结构即是RISC架构
  - 设计者把主要精力放在那些经常使用的指令上，尽量使它们具有简单高效的特色。对不常用的功能，常通过组合指令来完成。
  - 因此，在RISC 机器上实现特殊功能时，效率可能较低。但可以利用流水技术和超标量技术加以改进和弥补

4.1.4 为什么会出现CISC到RISC的转变

- 2/8规则
  - 在任何一组东西中，最重要的只占其中的一小部分，约20%，其余80%尽管是多数，却是次要的，因此又称二八定律
- 控制器设计难度下降
  - 许多指令可用最基本的指令编程来实现。

4.1.5 对指令系统的要求

- 完备性
  - 完备性是指用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
  - 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。许多指令可用最基本的指令编程来实现。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

4.1.5 对指令系统的要求

- 规整性
  - 规整性包括指令系统的对称性、匀齐性、指令格式和数据格式的一致性
  - 对称性是指：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
  - 匀齐性是指：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是指：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
- 兼容性
  - 系列机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，在结构和性能上有差异，做到所有软件都完全兼容是不可能的，只能做到“向上兼容”，即低档机上运行的软件可以在高档机上运行

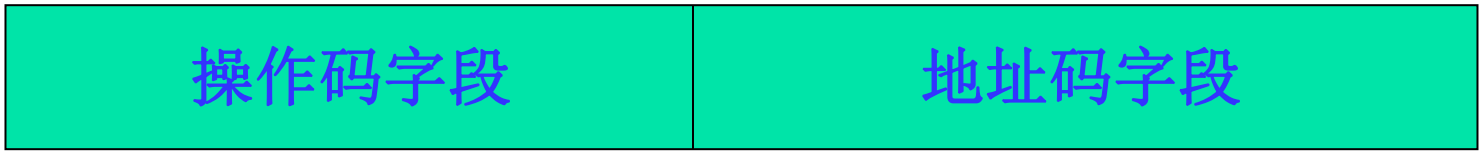
4.1.6 低级语言与高级语言关系

	比较内容	高级语言	低级语言
1	对程序员的训练要求 (1)通用算法 (2)语言规则 (3)硬件知识	 有 较少 不要	 有 较多 要
2	对机器独立的程度	独立	不独立
3	编制程序的难易程度	易	难
4	编制程序所需时间	短	较长
5	程序执行时间	较长	短
6	编译过程中对计算机资源的要求	多	少

4.2 指令格式

指令格式

- 影响计算机指令格式的因素
  - 机器的字长
  - 存储器的容量
  - 指令的功能
- 指令能反映以下信息
  - 做什么操作
  - 如果需要操作数，从哪里取
  - 结果送哪里
  - 下一条指令从哪里取
- 所以指令格式包括两个方面：



提纲

4.2.1 操作码

- 指令的操作码OP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
- 设计计算机时，对指令系统的每一条指令都要规定一个操作码
- 组成操作码字段的位数一般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来表示每条特定的指令。
  - 固定长度操作码：便于译码，扩展性差
  - 可变长度操作码：能缩短指令平均长度
- 操作码的的位数决定了所能表示的操作数量，n位操作码最多表示2^n种操作

4.2.1 可变长度的操作码(可扩展的操作码)

- 例如：假设某机器的指令长度为16位，包括4位基本操作码和三个4位地址码段。①表示三地址指令：因有4位操作码则能表示16条；②表示二地址指令：因有8位操作码则可表示256条；③表示一地址指令：因有12位操作码则可表示4096条
- 如果需要三地址、二地址、一地址指令各15条、零地址指令16条，则一样能够采用可变格式操作码实现。例如可以这样规定：
  - 15条三地址指令的操作码为：0000 ~ 1110 （操作码4位可表示16条指令，由于只有15条，所以还剩余一种状态1111，可以做二地址指令的标记）
  - 15条二地址指令的操作码为：前4位1111， 即 1111 0000 ~ 1111 1110
  - 15条一地址指令的操作码为：前8位均为1，即 11111111 0000 ~11111111 1110
  - 16条零地址指令的操作码为：前12位均为1，即 111111111110000~111111111111111
- 这是操作码不固定的指令格式，四位是基本的操作码，还可以扩充，但是指令的字数不变，就是说把不用的地址码部分可以做操作码用

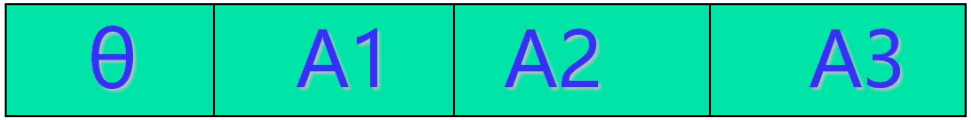
4.2.2 地址码

- 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

操作码（4位）	A1（6位）	A2（6位）	A3（6位）
操作码（4位）	A1（6位）	A2（6位）	
操作码（4位）	A1（6位）		
操作码（4位）			

4.2.2 地址码

- 三地址指令
- 指令格式如下：



- 操作码θ　第一操作数A1　第二操作数A2　　结果A3
- 功能描述：
  - (A1)θ(A2)→A3
  - (PC) +1→PC
- 这种格式指令长度仍比较长，所以只在字长较长的大、中型机中使用，而小型、微型机中很少使用。

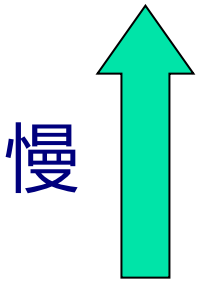
4.2.2 地址码

- 二地址指令
- 指令格式如下：

- 操作码θ　第一操作数A1　第二操作数A2
- 功能描述：
  - (A1)θ(A2)→A1
  - (PC)+1→PC
- 二地址指令在计算机中得到了广泛的应用，但是在使用时有一点必须注意：指令执行之后，A1中原存的内容已经被新的运算结果替换了。

4.2.2 地址码

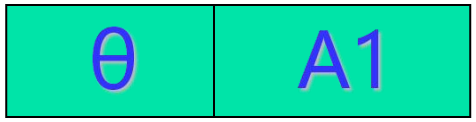
- 二地址指令根据操作数的物理位置分为：



  - SS　存储器-存储器类型
  - RS　寄存器-存储器类型
  - RR　寄存器-寄存器类型

4.2.2 地址码

- 一地址指令
- 指令格式如下：

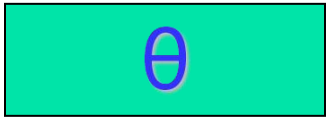


- 操作码θ　第一操作数A1
- 功能描述：
  - (A1)θ(A1) →A1
  - (PC)+1→PC
- 单操作数运算指令，如“+1”、“-1”、“求反”
- 指令中给出一个源操作数的地址

4.2.2 地址码

- 零地址指令

- 指令格式如下：



- 操作码θ
- “停机”、“空操作”、“清除”等控制类指令。

4.2.3 指令长度

- 1、指令字
- 机器字长：指计算机能直接处理的二进制数据的位数;
- 指令字长：指令字长度等于机器字长度的指令，称为单字长指令；指令字长度等于半个机器字长度的指令，称为半字长指令；指令字长度等于两个机器字长度的指令，称为双字长指令

- 2、多字长指令
- 使用多字长指令的目的，在于提供足够的地址位来解决访问内存任何单元的寻址问题
- 但是主要缺点是必须两次或多次访问内存以取出整条指令，这就降低了CPU的运算速度，同时又占用了更多的存储空间

4.2.3 指令长度

- 3、指令字结构

- 等长指令字结构：在一个指令系统中，如果各种指令字长度是相等的，称为等长指令字结构
  - 优点：指令字结构简单：取指快、译码简单。

- 变长指令字结构：如果在一个指令系统中，各种指令字长度随指令功能而异，就称为变长指令字结构
  - 优点：指令字结构灵活，能充分利用指令长度；
  - 缺点：指令的控制较复杂。

4.2.4 指令助记符

- 为了便于书写和阅读程序，每条指令通常用3个或4个英文缩写字母来表示。这种缩写码叫做指令助记符
- 数据传递
  - mov，load，store
- 算术逻辑逻运算
  - add，sub，and，not，or，xor，dec，inc，cmp
- *移位操作
  - shl，shr，srl，srr
- *转移控制
  - jmp，bnz，beq，call，ret，int，iret
- I/O指令
  - in，out
- 系统指令
  - Halt，nop，wait，sti，cli

4.2.5 指令格式举例

- 8位微型计算机的指令格式
  - 如8088，字长8位，指令结构可变
  - 包括单字长指令、双字长指令和三字长指令
  - 操作码长度固定
- PDP/11系列机（美商迪吉多电脑）的指令格式
  - 字长16位
  - 单字长指令
  - 操作码字段不固定

4.2.5 指令格式举例

- Pentium机指令格式
  - 指令长度可变，最短1个字节，最长12个字节，典型的CISC指令系统
  - 由可选前缀（0~4）、操作码（1~2）、一个由mod-R/M字节和一个SIB（Scale Index Base）比例变址字节组成的地址指定器、一个可选的位移量（0~4）和一个可选的立即数字段（0~4）构成

指令前缀	段取代	操作数长度取代	地址长度取代

操作码	Mod	Reg或操作码	R/M	S	I	B	位移量	立即数

4.2.5 指令格式举例

- 分析指令格式的特点（假设字长16位）

- ① 单字长二地址指令；② 操作码OP可指定=128条指令；③ RR型指令，两个操作数均在寄存器中，源和目标都是通用寄存器(可分别指定16个寄存器之一)；④ 这种指令格式常用于算术逻辑类指令。

- 分析指令格式的特点（假设字长16位）

- ① 双字长二地址指令；② 操作码OP可指定=64条指令；③ RS型指令，两个操作数一个在寄存器中(16个寄存器之一)，另一个在存储器中；④ 有效地址通过变址求得：E=(变址寄存器)± D，变址寄存器可有16个。

4.3 操作数类型

指令格式

- 操作数类型
  - 地址数据：地址实际上也是一种形式的数据
  - 数值数据：计算机中普遍使用的三种类型的数值数据：(1)定点数；（2）浮点数；（3）BCD（Binary-Coded Decimal）码（用4位二进制数来表示1位十进制数中的0~9这10个数码，可以表示比较长的十进制数并基此做运算）
  - 字符数据：文本数据或字符串，目前广泛使用ASCII码
  - 逻辑数据：一个单元中有几位二进制bit项组成，每个bit的值可以是1或0。当数据以这种方式看待时，称为逻辑性数据