

# 第四章：指令

## ▼ 基本概念

- 微指令：**微程序级的，属于硬件
- 宏指令：**由若干机器指令组成，属于软件
- 机器指令：**介于微指令和宏指令之间，简称为指令，每条指令可完成一个独立的算术运算或逻辑运算
- 指令系统：**一台计算机中所有机器指令的集合
- 复杂指令系统计算机（CISC）**
- 精简指令系统计算机（RISC）**

类 别	CISC	RISC
对比项目		
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

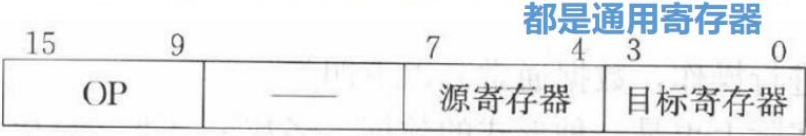
## ▼ 指令格式

- 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→ $2^n$

### 多字长指令

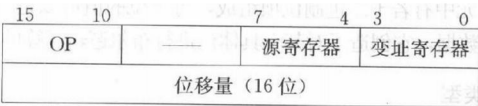
- 优点：**提供足够地址位来解决访问寻址问题
- 缺点：**必须两次或多次访存，降低CPU运算速度，占用空间

- 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

16位字长，指令格式如下，分析指令格式特点



- 一个在寄存器，一个在存储器
- 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）

## ▼ 定长/变长操作码

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。

**固定长度操作码：**便于译码，扩展性差，信息冗余极大

**可变长度操作码：**能缩短指令平均长度

**扩展操作码：**保持指令字长度不变而增加指令数量

➤ 三地址指令	操作码（4位）	A1（6位）	A2（6位）	A3（6位）
➤ 二地址指令	操作码（4位）	A1（6位）	A2（6位）	
➤ 单地址指令	操作码（4位）	A1（6位）		
➤ 零地址指令	操作码（4位）			

二地址根据操作数的物理位置分为：

SS：存储器-存储器（多次访存）

SR：存储器-寄存器（一次访存）

RR：寄存器-寄存器（无需访存）

- 一地址指令可能有两个操作数（其中一个隐含）

### 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？

- 编码特征：n位（操作码） $\rightarrow 2^n$ 
  - 5位操作码 $\rightarrow 32$ 条指令      256-m-n位
  - 6位操作码 $\rightarrow 64$ 条指令
  - 7位操作码 $\rightarrow 128$ 条指令
  - 8位操作码 $\rightarrow 256$ 条指令 .....

23. 某计算机按字节编址，指令字长固定且只有两种指令格式，其中三地址指令29条、二地址指令107条，每个地址字段为6位，则指令字长至少应该是

- A. 23位      B. 24位      C. 28位      D. 32位
- 

### 扩展操作码

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？

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 $m \times 2^n$ 种状态

- 有15条三地址指令  
共 $2^4=16$ 种状态  
留出 $16-15=1$ 种
- 有12条二地址指令  
共 $1 \times 2^4=16$ 种  
留出 $16-12=4$ 种
- 有62条一地址指令  
共 $4 \times 2^4=64$ 种  
留出 $64-62=2$ 种
- 有32条零地址指令  
共 $2 \times 2^4=32$ 种

32位

假设指令字长为16位，操作数的地址码为6位，指令有零地址、一地址、二地址3种格式。

- 1) 设操作码固定，若零地址指令有M种，一地址指令有N种，则二地址指令最多有几种？
- 2) 采用扩展操作码技术，二地址指令最多有几种？
- 3) 采用扩展操作码技术，若二地址指令有P条，零地址指令有Q条，则一地址指令最多有几种？

1)

	操作码	地址码	地址码
零	操作码	地址码	地址码
一	操作码	地址码	地址码
二	操作码	地址码	地址码

16-6-6=4      6      6

最多可表示 $2^4=16$ 种不同的操作码，对应16种不同的指令，其中M种用于零地址指令，N种用于一地址指令，故二地址指令最多有 $16-M-N$ 种。

2)

	操作码	地址码	地址码
零	操作码		
一	操作码		地址码
二	操作码	地址码	地址码

16-6-6=4      6      6

此时二地址指令最多可表示 $2^4=16$ 种不同的操作码，在采用扩展操作码技术时，至少留出1种编码用于一地址指令的扩展，故二地址指令最多 $16-1=15$ 种。

3)

二	操作码	地址码	地址码
$2^4-2$	0000	XXXXXX	XXXXXX
$=16-2$	:		
$=14$ 种	1101	XXXXXX	XXXXXX

一	操作码	地址码
$2 \times 2^6-2$	1110000000	XXXXXX
$=2 \times 64-2$	:	
$=127$ 种	1110111111	XXXXXX
	1111000000	XXXXXX
	:	
	1111111101	XXXXXX

零

操作码	
$2 \times 2^6$	111111110000000
$=128$ 种	:
	111111110111111
	111111110000000
	:
	111111111111111

二地址指令操作码4位，可有 $2^4$ 种编码，二地址指令有P条，则留出 $2^4-P$ 种编码用于一地址指令的扩展。一地址指令操作码字段可有 $(2^4-P) \times 2^6$ 种编码，设留出n种用于零地址指令的扩展，则 $n \times 2^6=Q$ ，那么 $n=Q/2^6$ ，故一地址指令最多有 $(2^4-P) \times 2^6-Q/2^6$ 种。

**变长操作码**

一处理器中共有32个寄存器，使用16位立即数，其指令系统结构中共有142条指令。在某个给定的程序中，20%的指令带有一个输入寄存器和一个输出寄存器；30%的指令带有两个输入寄存器和一个输出寄存器；25%的指令带有一个输入寄存器、一个输出寄存器和一个立即数寄存器；其余的25%指令带有一个立即数输入寄存器和一个输出寄存器。

1) 对于以上4种指令类型中的任意一种指令类型来说，共需要多少位？假定指令系统结构要求所有指令长度必须是8的整数倍。

2) 与使用定长指令集编码相比，当采用变长指令集编码时，该程序能够少占用多少存储空间？

对指令进行编码需要8比特信息       $2^7=128, 2^8=256$

对寄存器进行编址需要5比特信息       $2^5=32$

指令	操作码	地址码
第一类指令: $8+5+5=18$ 取24	第二类指令: $8+5+5+5=23$ 取24	
第三类指令: $8+5+5+16=34$ 取40	第四类指令: $8+16+5=29$ 取32	

变长指令:  $20\% \times 24 + 30\% \times 24 + 25\% \times 40 + 25\% \times 32 = 30\text{bit}$

$(40-30)/40=25\%$

▼ 寻址方式

▼ 指令寻址方式

顺序方式 (PC=PC+1)

跳跃方式 (JMP语句)

例：JMP 7

- 1. 跳转语句后跟绝对地址：PC=7
- 2. 跳转语句后采用相对寻址（给出偏移量）：PC=PC+7+指令长度

- **按字节编址**：“1”代表指令字长
- **按字编址**：字长16则一个字=2B，所有地址表示都以2B为单位。  
“1”看指令长度，单字指令为1，双字指令为2（不用管字长是多少）

某计算机系统字长为32位（按字节寻址），当前PC=1000H，将执行JMP 8指令（采用相对寻址方式，单字长指令），请问下一条语句执行地址为

字长=32/8=4字节  
1000H+8+4("1"为单字长)=100CH

▼ 操作数寻址方式

寻址方式总结（重要）

寻址方式	有效地址	访存次数	示例
隐含寻址	语句默认指定	0	STC、CLC
立即寻址	指令包含操作数	0	MOV AH, 80H
寄存器寻址	EA=Ri	0	MOV AH, 80H
直接寻址	EA=A	1	ADD R0, [6]
间接寻址	EA=(A)	2	
寄存器间接寻址	EA=(Ri)	1	MOV AX,[ESP]
相对寻址	EA=(PC)+A	1	转移指令
基址寻址	EA=(Bx)+A	1	扩大访存、段寻址
变址寻址	EA=(Rx)+A	1	循环语句
堆栈寻址	EA=(SS)+SP	1	POP、PUSH

偏移寻址

- 相对寻址：转移指令
- 基址寻址：扩大寻址空间
- 变址寻址：数组，A不变，Rx自增



处理机指令格式如下

1位		3位	
X	OP	寄存器	地址

其中，寄存器为3位（12位通用寄存器）；最高位X用来指明是否选用变址寻址（X=1，通用寄存器为变址寄存器）；主存最大容量为16384字节（按字节寻址）。

（1）要求不使用通用寄存器也能访问每个一主存地址，且支持88条指令，地址码域应分配多少位，OP码域应分配多少位，指令字共多少位

OP：7位（ $2^7=128$ ）

地址码：14位（ $2^{14}=16384$ ）

指令字长度 $14+7+3+1=25$ 位

处理机指令格式如下

1位	7位	3位	14位
X	OP	寄存器	地址

（2）X=0，指令字中明确使用通用寄存器，且通用寄存器为基址寄存器。给出硬件设计规则，使通用寄存器能访问主存每一个位置。

基址寄存器（12位，左移2位），与形式地址相加

（3）假设主存容量扩充到32768字节，硬件结构已确定不变，采用什么方法可完成寻址？

15位，采用间接寻址方式

间接寻址	$EA=(A)$
寄存器间接寻址	$EA=(Ri)$
相对寻址	$EA=(PC)+A$
基址寻址	$EA=(Bx)+A$
变址寻址	$EA=(Rx)+A$

- 某机双操作数指令占用3字节格式，OP为操作码，S/D表示Reg中存放的是源还是目的操作，M为寻址模式：0为直接寻址；1为基址寻址；2为相对寻址；3为立即寻址。A为偏移量，有符号数，补码表示。通用寄存器有8个，其中R7为基址寄存器。

OP (7)		S/D (1)
M (2)	Reg (3)	R (3)
A (8)		

- 问：（1）该指令格式可定义多少种不同的操作？  
OP位数给出， $2^7=128$ 种
- （2）立即寻址操作数的范围是多大？  
立即寻址操作数为A，8位补码范围：-128~127
- 问：（3）若基址寄存器为14位，采用基址寻址方式的寻址范围为多少？ 基址寻址 $EA=(R7)+A$ ， $2^{14}=16KB$
- （4）写出4种寻址方式计算有效地址的表达式  
直接 $EA=A$ ，基址 $EA=(R7)+A$ ，相对 $EA=(PC)+A$ ，立即A操作数

7. 某计算机字长为32位，主存容量为64KB，采用单字长单地址指令，共有40条指令。试采用直接、立即、变址、相对四种寻址方式设计指令格式。

31	25	23	0
OP	X	D	
$X=00$ 直接寻址 有效地址 $E=D$			
$X=01$ 立即寻址 立即数D			
$X=10$ 变址寻址 $E=(R_X)+D$ ( $R_X$ 为变址寄存器32位)			
$X=11$ 相对寻址 $E=(PC)+D$ ( $PC$ 为程序计数器32位)			

某机字长为16位，存储器按字编址，访问内存指令格式如下：

15	11	10	8	7	0
OP			M	A	

其中，OP为操作码；M为寻址特征；A为形式地址。设PC和Rx分别为程序计数器和变址寄存器，字长为16位，问：

- 该指令能定义多少种指令？ OP字段长为5位，所以指令能定义 $2^5=32$ 种指令
- 表4-2中各种寻址方式的寻址范围为多少？

寻址方式	有效地址EA的计算公式	寻址范围	
直接寻址	$EA=A$	$2^8=256$	A有8位
间接寻址	$EA=(A)$	$2^{16}=64K$	看字长，为16位 (A地址位置的值为16位)
变址寻址	$EA=(R_x)+A$	$2^{16}=64K$	看字长，为16位 ( $R_x$ 地址位置的值为16位)
相对寻址	$EA=(PC)+A$	$2^8=256$ (PC附近256)	PC是固定的，因此看A有8位

例：某16位机器所使用的指令格式和寻址方式如下所示，该机有两个20位基址寄存器，四个16位变址寄存器，十六个16位通用寄存器，指令汇编格式中的S（源），D（目标）都是通用寄存器，M是主存中的一个单元。三种指令的操作码分别是MOV（OP）=（A）H，STA（OP）=（1B）H，LDA（OP）=（3C）H。MOV是传送指令，STA为写数指令，LDA为读数指令。



- （1）分析三种指令的指令格式与寻址方式特点。
  - （1）MOV:单字长，二地址指令属于RR指令  
STA:双字长，二地址指令，RS指令 S为基址寄存器或变址寄存器  
LDA:双字长，二地址指令，RS指令 S为20位地址。
- （2）CPU完成哪一种操作所花时间最短？哪一种操作所花时间最长？第二种指令的执行时间有时会等于第三种指令的执行时间吗？
  - （2）MOV单字长取出只需要一次访存，第一个不访问存储器要快。第二种还需要计算有效地址并对存储器进行访问。第二种指令所花费的时间不等于第三种，第三种无需进行有效地址的计算。
- （3）下列情况下每个十六进制指令字分别代表什么操作？其中如果有编码不正确，如何改正才能成为合法指令？
  - ①（F0F1）H（3CD2）H      ②（2856）H  
③（6FD6）H                      ④（1C2）H
  - （3）根据条件。MOV(OP)=(A)h=001010  
STA(OP)=(1B)h=011011  
LDA(OP)=(3C)h=111100  
(F1F1)h(3CD2)h=(1111 0000 1111 0001 0011 1100 1101 0010)b，正确，表明把主存(13cd2)h地址单元的内存取至15号寄存器  
(2856) H= (0010 1000 0101 0110) b，单字长指令表明把寄存器6的内容传送到寄存器5号  
(6FD6) H=0110 1111 1101 0110b操作码不对 应修改为 (28D6) h  
(1C2) H=00000001 1100 0010b操作码不对 应修改为 (28C2) h

▼ 典型指令

数据传送指令：

取数指令、存数指令、传送指令、成组传送指令、字节交换指令、清累加器指令、堆栈操作指令

这类指令主要用来实现主存和寄存器之间，或寄存器和寄存器之间的数据传送。

算术运算指令：

二进制定点加、减、乘、除指令，浮点加、减、乘、除指令，求反、求补指令，算术移位指令（带符号移位），比较指令，十进制加、减运算指令



这类指令主要用于定点或浮点的算术运算。

### 逻辑运算指令：

逻辑加、逻辑乘、按位加、逻辑移位（不带符号移位）等指令，主要用于无符号数的位操作、代码的转换、判断及运算

### 程序控制指令（转移指令）：

条件转移指令、无条件转移指令、转子程序指令、返回主程序指令、中断返回指令

转移指令的转移地址一般采用直接寻址和相对寻址方式来确定。

### 输入输出指令：

输入输出指令主要用来启动外围设备，检查测试外围设备的工作状态，并实现外部设备和CPU之间，或外围设备与外围设备之间的信息传送。

### 字符串处理指令：

包括字符串传送、字符串转换（把一种编码的字符串转换成另一种编码的字符串）、字符串替换（把某一字符串用另一字符串替换）

非数值处理指令

### 特权指令：

这类指令只用于操作系统或其他系统软件，一般不直接提供给用户使用。

### 其他指令：

状态寄存器置位、复位指令、测试指令、暂停指令，空操作指令，

以及其他一些系统控制用的特殊指令。

## ▼ 选择题

为了缩短指令中某个地址段的位数，有效的方法是采取（**D**）。

A. 立即寻址      B. 变址寻址      C. 基址寻址      D. 寄存器寻址

简化地址结构的基本方法是尽量采用（**B**）。

A. 寄存器寻址      B. 隐地址      C. 直接寻址      D. 间接寻址

设指令中的地址码为A，变址寄存器为X，程序计数器为PC，则变址间址寻址方式的操作数有效地址EA是（**B**）。

A.  $((PC)+A)$       B.  $((X)+A)$       C.  $(X)+(A)$       D.  $(X)+A$

先变址再将B作为形式地址进行间址

变址： $(X)+A \rightarrow B$       间址： $(B) = ((X)+A)$

堆栈寻址方式中，设A为累加器，SP为堆栈指示器，Msp为SP指示的栈顶单元。如果进栈操作的动作是：

(A)→Msp, (SP)-1→SP, 那么出栈操作的动作应为 (B)。

- A. (Msp)→A, (SP)+1→SP      B. (SP)+1→SP, (Msp)→A  
C. (SP)-1→SP, (Msp)→A      D. (Msp)→A, (SP)-1→SP

相对寻址方式中，指令所提供的相对地址实质上是一种 ( )。

- A. 立即数  
B. 内存地址  
C. 以本条指令在内存中首地址为基准位置的偏移量  
D. 以下条指令在内存中首地址为基准位置的偏移量

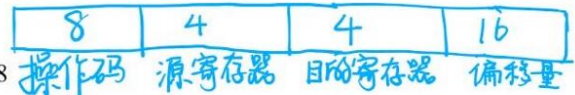
26. 某计算机有 <sup>24</sup>16 个通用寄存器，采用 32 位定长指令字，操作码字段（含寻址方式位）为 8 位，Store 指令的源操作数和目的操作数分别采用寄存器直接寻址和基址寻址方式。若基址寄存器可使用任一通用寄存器，且偏移量用补码表示，则 Store 指令中偏移量的取值范围是 16。

A. - 32768~+32767

B. - 32767~+32768

C. -65536~+65535

D. - 65535~+65536



28. 下列指令系统的特点中，有利于实现指令流水线的是 (D)

- I. 指令格式规整且长度一致 → 取指时间一致、简化指令译码逻辑  
II. 指令和数据按边界对齐存放 → 加速访存  
III. 只有 Load/Store 指令能对操作数进行存储访问 → 流水段划分一致

A. 仅 I、II

B. 仅 II、III

C. 仅 I、III

D. I、II、III