

## 第 7 章 指令系统

### 计算机硬件和软件的接口—指令系统

计算机能解题是由于机器本身存在一种语言，它既能理解人的意图，又能被机器自身识别。

**机器语言**(0 1 代码，高级语言、汇编语言编写的程序需要编译程序变成机器语言代码)是由一条条语句构成的，每一条语句又能准确表达某种语义。例如，它可以命令机器做某种操作，指出参与操作的数或其他信息在什么地方等。

计算机就是连续执行每一条机器语句而实现全自动工作的。

人们习惯把每一条机器语言的语句称为**机器指令**(机器能够直接识别并直接执行的操作命令)，而又将全部机器指令的集合称为机器的**指令系统**。因此机器的指令系统集中反映了机器的功能。

机器设计者主要研究如何确定机器的指令系统，如何用硬件电路、芯片、设备来实现机器指令系统的功能。计算机的使用者则是根据机器提供的指令系统，使用汇编语言来编制各种程序。计算机使用者根据机器指令系统所描述的机器功能，能很清楚地了解计算机内部寄存器-存储器的结构，以及计算机能直接支持的各种数据类型。

#### 7.1.1 指令的一般格式

指令由操作码和地址码两部分组成的。

**操作码**用来指明所要完成的操作，如加法、减法、传送、移位、转移等。

通常，其位数反映了机器的操作种类，也即机器允许的指令条数，如操作码 7 位，机器最多包含 128 条指令。

操作码的长度可以是**固定**的，也可以是变化的。

固定操作码长度是将操作码集中放在指令字的一个字段内。这种格式便于硬件设计，指令译码时间短，广泛应用于字长较长的、大中型计算机和超级小型计算机以及 RISC 中。

对于操作码长度不固定的指令，其操作码分散在指令字的不同字段中。这种格式可有效地压缩操作码的平均长度，在字长较短的微型计算机中被广泛采用。

操作码长度**不固定**会增加指令译码和分析的难度，使控制器的设计复杂。

#### 扩展操作码

通常采用**扩展操作码**技术，使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长度的操作码，从而在满足需要的前提下，有效地缩短指令字长。

#### 保留码点

经常出现的指令用短操作码来表示，不经常出现的低频指令用长操作码来表示。

在设计操作码不固定的指令系统时，应尽量考虑安排指令使用频度(指令在程序中出现的概率)高的指令占用短的操作码，对使用频度低的指令可占用较长的操作码，这样可以缩短经常使用指令的译码时间。当然，考虑操作码长度时也应考虑地址码的要求。

**地址码**用来指出该指令的源操作数的地址(一个或两个)、结果的地址以及下一条指令的地址。

这里的地址可以是主存的地址，也可以是寄存器的地址，甚至可以是 I/O 设备的地址。

从计算机诞生至今，指令的地址码字段经历了四地址、三地址、二地址、一地址和零地址这几个阶段。

假设指令字长等于存储字长，均为 32 位，操作码 OP 占 8 位， $A_i$  表示存储器地址，下表列出了不同地址格式指令的访存次数(包括取指令)及操作数的寻址范围。

选址格式	操作	访存次数	操作数寻址范围	备注
四地址	$(A_1)OP(A_2) \rightarrow A_3$	4	$2^6$	$A_4$ 指出下条指令地址
三地址	$(A_1)OP(A_2) \rightarrow A_3$	4	$2^8$	PC 代替 $A_4$
二地址	$(A_1)OP(A_2) \rightarrow A_1$	4	$2^{12}$	$A_1$ 代替 $A_3$
二地址	$(A_1)OP(A_2) \rightarrow A_2$	4	$2^{12}$	$A_2$ 代替 $A_3$
二地址	$(A_1)OP(A_2) \rightarrow ACC$	3	$2^{12}$	ACC 存放结果
一地址	$(ACC)OP(A_1) \rightarrow ACC$	2	$2^{24}$	ACC 存放操作数和结果

四地址

OP	$A_1$	$A_2$	$A_3$	$A_4$
----	-------	-------	-------	-------

$(A_1)OP(A_2) \rightarrow A_3$

$A_1$  为第一操作数地址；OP 为操作码字段，指明该指令要完成的操作； $A_2$  为第二操作数地址； $A_3$  为结果地址； $A_4$  为下一条指令地址。

三地址

OP	$A_1$	$A_2$	$A_3$
----	-------	-------	-------

程序中大多数指令是按顺序执行的，而程序计数器 PC 既能存放当前欲执行指令的地址，又有计数功能，能自动形成下一条指令的地址。省去  $A_4$ ，用 PC 代替。

二地址指令

OP	$A_1$	$A_2$
----	-------	-------

机器的运算结果不必都存入主存，省去  $A_3$ ，将中间结果暂存在  $A_1$  或  $A_2$  中。

一地址指令

OP	$A_1$
----	-------

ACC 既能存放参加运算的操作数，又能存放运算的中间结果，这样，完成一条一地址指令只需两次访存。

零地址指令

在指令字中无地址码，例如空操作(NOP)、停机(HLT)这类指令只有操作码。而子程序返回(RET)、中断返回(IRET)这类指令没有地址码。

可见，用一些硬件资源如 PC、ACC 存放指令中必须指明的地址码，可在不改变指令字长的前提下，扩大指令操作数的直接寻址范围。此外，用 PC、ACC 等硬件代替指令中的某些地址字段，还可以缩短指令字长，并可减少访存次数。

7.1.2 指令字长

指令字长取决于操作码的长度、操作数地址的长度和操作数地址的个数。不同机器的指令字长是不相同的。同一机器的指令字长可以是固定的，也可以是不固定的。通常指令字长取 8 的整数倍。

## 7.2 操作数类型和操作类型

### 7.2.1 操作数类型

机器中常见的操作数(指令要进行处理的数据)类型有**地址**、**数字**、**字符**、**逻辑数据**等。

**地址**(例如, 跳转指令中)实际上也可看做是一种数据, 在许多情况下要计算操作数的地址。这时, 地址可被认为是一个无符号的整数。

计算机中常见的**数字**有无符号数、有符号数、定点数、浮点数和十进制数(BCD 码, 4 位二进制数代表一位十进制数)。

在应用计算机时, 文本或者**字符串**也是一种常见的数据类型(ASCII 码等)。

计算机除了作算术运算外, 有时还需作逻辑运算, 此时  $n$  个 0 和 1 的组合不是被看做算术数字, 而是被看做**逻辑数**。逻辑数据是布尔类型的数据, 它们的每一位代表真(1)或假(0), 可参与逻辑运算。

### 7.2.2 数据在存储器中的存放方式

通常计算机中的数据存放在存储器或寄存器中, 而寄存器的位数便可反映机器字长。

一般字长可取字节的 1、2、4、8 倍, 这样便于字符处理。

不同机器的数据字长是不同的, 同一台机器也可以处理不同字长的数据。因此, 为了便于硬件实现, 通常要求多字节的数据在存储器的存储方式能满足“边界对准”的要求。存储器可按字节、半字、字、双字访问。对于不同字节的数据, 不同的机器存放的方式也不同, 有的机器以低字节地址作为字地址, 有的机器以高字节地址作为字地址。

在使用不同的机器时, 要注意数据在存储器中存放的方式, 避免应用时出错。

### 7.2.3 操作类型

不同的机器有不同的操作类型, 但几乎所有的机器都有数据传送、算术逻辑运算、移位、转移、输入输出和其他类型的操作(包括停机、空操作、开中断、关中断、置条件码等。)

**数据传送**包括寄存器与寄存器、寄存器与存储单元、存储单元与存储单元之间的传送。

如从源到目的之间的传送、对存储器读(LOAD)和写(STORE)、交换源和目的的内容、置 1、清零、进栈、出栈等。

**算术逻辑操作**可实现算术运算(加、减、乘、除、增 1、减 1、取负数即求补)和逻辑运算(与、或、非、异或)。

对于低档机而言, 一般算术运算只支持最基本的二进制加减、比较、求补等, 高档机还能支持浮点运算和十进制运算。有些机器还具有位操作功能、如位测试(测试指定位的值)、位清除(清除指定位)、位求反(对指定位求反)等。

**移位**可分为算术移位、逻辑移位和循环移位三种。

算术移位和逻辑移位分别可实现对有符号数和无符号数乘以  $2^n$ (左移)和除以  $2^n$ (右移)的运算, 并且移位操作所需时间远比乘除操作执行时间短, 因此, 移位操作经常被用来代替简单的乘法和除法操作。

在多数情况下, 计算机是按顺序执行程序每条指令的, 但有时需要改变这种顺序, 此时可采用转移类指令来完成。

**转移指令**可按其转移特征又可分为无条件转移、条件转移、跳转、过程调用与返回、陷阱等几种。

对于 I/O 单独编址的计算机而言, 通常设有**输入输出指令**, 它完成从外设中的寄存器读入一个数据到 CPU 的寄存器内, 或将数据从 CPU 的寄存器输出至某外设的寄存器中。

**其他类型的操作**包括等待指令、停机指令、空操作指令、开中断指令、关中断指令、置条件码指令等。

### 7.3 寻址方式

寻址方式是指如何确定**本条指令的操作数地址**以及**下一条将要执行指令的地址**，它与硬件结构紧密相关，而且直接影响指令格式和指令功能。

寻址方式分指令寻址和数据寻址两大类。

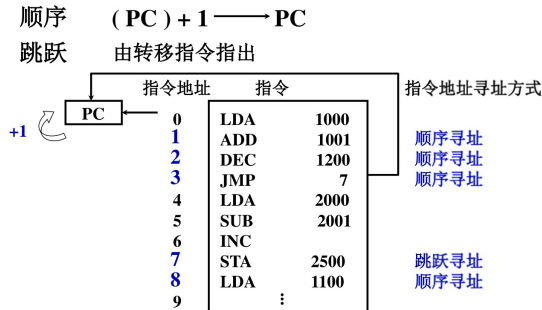
**指令寻址**分顺序寻址和跳跃寻址。

顺序寻址可通过程序计数器 PC 加 1，自动形成下一条指令的地址，跳跃寻址则通过转移类指令来实现。

右图示意了指令寻址过程。

如果程序的首地址为 0，只要先将 0 送至程序计数器 PC 中，启动机器运行后，程序便按 0,1,1,3,7,8,9,...顺序执行。其中第 1、2、3 号指令地址均由 PC 自动形成。因第三条指令为 JUM 7，故执行完第三条指令后，又顺序执行第 8 条、第 9 条指令。

#### 一、指令寻址



**数据寻址**方式种类较多，在指令字中必须设一字段来指明属于哪一种寻址方式。

指令的地址码字段通常都不代表操作数的真实地址，把它称为形式地址，记作 A。操作数的真实地址称为有效地址，记作 EA，它是由寻址方式和形式地址共同来确定的。由此可得指令的格式应如右图所示：

#### 二、数据寻址

操作码	寻址特征	形式地址 A
形式地址	指令字中的地址	
有效地址	操作数的真实地址	
约定	指令字长 = 存储字长 = 机器字长	

为了便于分析各类寻址方式，假设指令字长、存储字长、机器字长均相等。

(1) 立即寻址的形式地址 A 就是操作数本身，称作立即数(补码表示)，寻址特征 #

OP	#	A
----	---	---

指令执行阶段补访存；A 的位数限制了立即数的范围。

(2) 直接寻址

EA = A 有效地址由形式地址直接给出。

LDA		A
-----	--	---

执行阶段访问一次存储器；

A 的位数决定了该指令操作数的寻址范围；

操作数的地址不宜修改(必须修改 A)

(3) 隐含寻址

操作数地址隐含在操作码中

ADD		A
-----	--	---

另一个操作数隐含在 ACC 中。

指令字中少了一个地址字段，可缩短指令字长。

(4) 间接寻址

EA = (A) 有效地址 EA 由形式地址 A 间接提供

指令字中的形式地址不直接指出操作数的地址，而是指出操作数有效地址所在的存储单元地址。

OP		A
----	--	---

执行指令阶段 2 次访存；可扩大寻址范围；便于编制程序。

例. 某机指令格式如下图所示：



图中 I 为间址特征位(I = 0，直接寻址；I = 1，一次间址寻址)。假设存储器部分单元有以下内容：

地址号(十六进制)	00	01	02	03	04	05	06	07
内容(十六进制)	01	5E	9D	74	A4	15	04	A0

指出下列机器指令(十六进制表示)的有效地址。

(1) D7；(2) DF；(3) DE；(4) D2。

答：

(1) 1101 0 111，直接寻址，有效地址为 07H

(2) 1101 1 111，间址寻址，有效地址为 07 地址号对应的内容 A0H

(3) 1101 1 110，间址寻址，有效地址为 06 地址号对应的内容 04H

(4) 1101 0 010，直接寻址，有效地址为 02H

(5) 寄存器寻址

$EA = R_i$       有效地址 EA 即为寄存器编号



执行阶段不访存，只访问寄存器，执行速度快；寄存器个数有限，可缩短指令字长。

(6) 寄存器间接寻址

$EA = (R_i)$       有效地址 EA 在寄存器中



有效地址在寄存器中，操作数在存储器中，执行阶段访存；便于编制循环程序

(7) 基址寻址

$EA = (BR) + A$ ，其中 BR 为基址寄存器(专用)，也可用通用寄存器作为基址寄存器。

基址寻址的特点是：可扩大操作数的寻址范围(基址寄存器的位数大于形式地址 A 的位数)；有利于多道程序运行；基址寄存器的内容有操作系统或管理程序确定；在程序执行过程中，基址寄存器的内容不变(作为基地址)，形式地址可变(作为偏移量)。

值得注意的是，当采用通用寄存器作基址寄存器时，可由用户指定哪个寄存器作基址寄存器，但其内容仍由操作系统确定。

(8) 变址寻址

$EA = (IX) + A$ ，其中 IX 为变址寄存器(专用)，也可用通用寄存器作为变址寄存器。

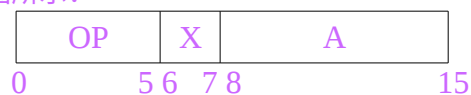
变址寻址的特点是：可扩大操作数的寻址范围(变址寄存器的位数大于形式地址 A 的位数)；变址寄存器的内容由用户给定；在程序执行过程中，变址寄存器的内容可变(作为偏移量)，形式地址 A 不变(作为基地址)。

(9) 相对寻址

$EA = (PC) + A$ ，A 是相对于当前指令地址的位移量，可正可负，用补码表示。

相对寻址的特点是：A 的位数决定操作数的寻址范围；便于程序浮动；广泛应用于转移指令。

例：某机指令格式如下图所示：



图中 X 为寻址特征位，且

当  $X = 0$  时，不变址；当  $X = 1$  时，用变址寄存器  $X_1$  进行变址； $X = 2$  时，用变址寄存器  $X_2$  进行变址； $X = 3$  时，相对变址。

设  $(PC) = 1234H$ ,  $[X_1] = 0037H$ ,  $[X_2] = 1122H$ ，确定下列指令的有效地址(指令和地址均用十六进制表示)：

(1) 4420H；(2) 2244；(3) 1322；(4) 3521。

答：

(1) 010001 00 00100000,  $X = 0$ ，不变址，结果 0020H

(2) 001000 10 01000100,  $X = 2$ ，用变址寄存器  $X_2$  进行变址，有效地址为  $0044 + 1122 = 1166H$

(3) 000100 11 00100010,  $X = 3$ ，相对变址，有效地址为程序计数器的内容(当前指令的地址)和指令字中的形式地址  $A$  相加而成， $1234 + 0022 = 1256H$

(4) 001101 01 00100001,  $X = 1$ ，用变址寄存器  $X_1$  进行变址，有效地址为  $0037 + 0021 = 0058H$

#### (10) 堆栈寻址

堆栈寻址要求计算机中设有堆栈。

堆栈既可用寄存器组(硬堆栈)来实现，也可利用主存的一部分空间作堆栈(软堆栈)，其特点是先进后出。

堆栈的站定地址由  $SP$  指出，堆栈寻址的有效地址  $EA$  隐含在堆栈指针  $SP$  中。每次进栈或出栈， $SP$  自动修改。如进栈  $(SP) - 1 \rightarrow SP$ 。

堆栈寻址的特点是：因有效地址隐含在  $SP$  中，所以指令中可以少一个地址字段；进栈出栈要修改地址指针，进栈  $(SP) - \Delta \rightarrow SP$ ，出栈  $(SP) + \Delta \rightarrow SP$ 。 $\Delta$  取值与主存编址方式有关，若按字编址， $\Delta$  取 1；若按字节编址，当字长为 16 位， $\Delta$  取 2，当字长为 32 位， $\Delta$  取 4。

掌握机器指令的寻址方式对于汇编语言编程的用户十分重要。

对于参与机器指令系统的设计人员而言，了解寻址方式对确定指令格式是必不可少的。

对于广大读者来说，只有透彻了解了机器指令的寻址方式，才能加深对机器内信息流程及整机工作概念的理解。

## 7.4 指令格式举例

指令格式不仅体现了指令系统的各种功能，而且也突出地反映了机器的硬件结构特点。设计指令格式时必须从诸多方面综合考虑，并经一段模拟运行后，最后确定。

### 7.4.1 设计指令格式应考虑的各种因素

指令系统集中反映了机器的性能，有是程序员编程的依据。用户在编程时既希望指令系统很丰富，便于用户选择，同时还要求机器执行程序时速度快、占用主存空间少，实现高效运行。此外，为了继承已有的软件，必须考虑新机器的指令系统与同一系列机器指令系统的兼容性，即高档机必须能兼容低档机的程序运行，称之为向上兼容。

指令格式集中体现了指令系统的功能，为此，在确定指令格式时，必须从以下几个方面综合考虑：

(1) 操作类型：包括指令数及操作的难易程度。

(2) 数据类型：确定哪些数据类型可以参与操作。

(3) 指令格式：包括指令字长、操作码位数、地址码位数、地址个数、寻址方式类型，以及指令字长和操作码位数是否可变等。

(4) 寻址方式：包括指令和操作数具体有哪些寻址方式。

(5) 寄存器个数：寄存器的多少直接影响指令的执行时间。

## 7.5 RISC 技术

RISC 即精简指令系统计算机(Reduced Instruction Set Computer), 与其对应的是 CISC(Complex Instruction Set Computer), 即复杂指令系统计算。

在人们进一步分析 CISC 后, 发现一个 80-20 规律, 即典型程序中 80%的语句仅使用 20%的指令(这些指令都属于简单指令)。而且当执行频度高的简单指令时, 因复杂指令的存在, 致使执行速度也无法提高。人们从 80-20 规律中得到启示: 能否用 20%的简单指令, 重新组合不常用的 80%的指令功能呢? 这便引发出 RISC(Reduced Instruction Set Computer)技术。

RISC 的主要特点:

- (1) 选取使用频度高的一些简单指令, 复杂指令的功能由简单指令的组合来实现。
- (2) 指令长度固定, 指令格式种类少, 寻址方式种类少。
- (3) 只有 LOAD/STORE 指令访存, 其余指令的操作都在寄存器之间进行。
- (4) CPU 中有多个通用寄存器。
- (5) 控制器采用组合逻辑控制。
- (6) 采用流水技术, 大部分指令在 1 个时钟周期完成。
- (7) 采用优化的编译程序。

CISC 的主要特点:

- (1) 指令系统复杂庞大, 各种指令使用频度相差很大。
- (2) 指令字长不固定, 指令格式多, 寻址方式多。
- (3) 可以访存的指令不受限制。
- (4) CPU 中设有专用寄存器。
- (5) 各种指令执行时间相差很大, 大多数指令需多个时钟周期才能完成。
- (6) 控制器大多数采用微程序控制。
- (7) 难以用优化编译生成高效的目标代码程序。

RISC 和 CISC 的比较:

- (1) RISC 更能充分利用 VLSI 芯片的面积。
- (2) RISC 更能提高运算速度。
- (3) RISC 便于设计, 可降低成本, 提高可靠性。
- (4) RISC 有利于编译程序代码优化。
- (5) RISC 不易实现指令系统兼容。

选择题

1. 指令系统采用不同的寻址方式的目的是**缩短指令字长, 扩大寻址空间, 提高编程灵活性**。
2. 零地址运算指令在指令格式中不给出操作数地址, 它的操作数来自于**栈顶和次栈顶**。
3. 一地址指令中, 为完成两个数的算术运算, 除地址译码指明的一个操作数外, 另一个数常采用**隐含寻址方式**。
4. 二地址指令中, 操作数的物理位置可安排在**两个主存单元、两个寄存器、一个主存单元和一个寄存器**。
5. 操作数在寄存器中的寻址方式称为**寄存器直接寻址**。
6. 寄存器间接寻址中, 操作数在**主存单元**中。
7. 变址寻址方式中, 操作数的有效地址是**变址寄存器内容加上形式地址**。
8. 基址寻址方式中, 操作数的有效地址是**基址寄存器内容加上形式地址(位移量)**。
9. 采用基址寻址可扩大寻址范围, 且**基址寄存器内容**由操作系统确定, 在程序执行过程中可变。
10. 采用变址寻址可扩大寻址范围, 且**变址寄存器内容**由用户确定, 在程序执行过程中可变。
11. 变址寻址和基址寻址的有效地址形成方式类似, 但是在程序执行过程中, **基址寄存器的内容不可变, 变址寄存器的内容可变**。

12. 堆栈寻址方式中, 设 A 为累加器, SP 为堆栈指示器,  $M_{SP}$  为 SP 指示的栈顶单元, 如果进栈操作的动作顺序是  $(A) \rightarrow MSP, (SP)-1 \rightarrow SP$ , 那么出栈操作的动作顺序应为  $(SP)+1 \rightarrow SP, (M_{SP}) \rightarrow A$ 。
14. 设变址寄存器内容为 X, 形式地址为 D, 某机具有先变址再间址的寻址方式, 则这种寻址方式的有效地址为  $EA = ((X) + D)$ 。
15. 设变址寄存器内容为 X, 形式地址为 D, 某机具有先间址再变址的寻址方式, 则这种寻址方式的有效地址为  $EA = (X) + (D)$ 。
16. IBM PC 中采用了段寻址方式, 在寻访一个主存具体单元时, 由一个基地址加上某寄存器提供的 16 位偏移量来形成 20 位物理地址。这个基地址是由 CPU 中四个 16 位段寄存器之一自动左移 4 位来提供。
17. 程序控制类指令的功能是**改变程序执行的顺序**。
18. 运算型指令的寻址和转移型指令的寻址不同点在于**前者取操作数, 后者决定程序转移地址**。
19. 指令的寻址方式有顺序和跳跃两种, 采用跳跃寻址可以实现**程序的条件转移和无条件转移**。
20. 扩展操作码是一种**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。
23. 设相对寻址的转移指令占两个字节, 第一字节是操作码, 第二字节是相对位移量(可正可负), 则该转移的地址范围是 **256**。
24. 直接、间接、立即三种寻址方式指令的执行速度, 由快至慢的排序是**立即、直接、间接**。
25. 一条指令中包含的信息有**操作码、地址码**。
26. 为了缩短指令中地址码的位数, 应采用**寄存器寻址**。
27. 若数据在存储器中采用低字节地址为字地址的存放方式, 则十六进制数 12345678H 按字节地址由小到大依次存为 **78563412**。
28. 在指令格式设计中, 采用扩展操作码的目的是**增加指令数量**。
29. 设机器字长为 16 位, 存储器按字编址, 对于单字节指令而言, 读取该指令后, PC 值能自动增加 **1**。
30. 设机器字长为 16 位, 存储器按字节编址, CPU 读取一条单字长指令后, PC 值自动加 **2**。
32. 指令系统中采用不同寻址方式的主要目的是**缩短指令字长, 扩大寻址空间, 提高编程的灵活性**。
33. 指令操作所需的数据不可能来自**控制存储器**。
34. 转移指令的主要操作是**改变程序计数器 PC 的值**。
35. 子程序调用指令完整的功能是**改变程序计数器的值和堆栈指针 SP 的值**。
36. 子程序返回指令完整的功能是**从堆栈中恢复程序计数器的值**。
37. 通常一地址格式的算术运算指令, 另一个操作数隐含在**累加器**中。
38. 为了充分利用存储空间, 指令的长度通常可取字节的整数倍; 单字长指令可加快取指令的速度。
39. 在二地址指令中, 运算结果通常存放在其中一个地址码提供的地址中。
40. 在一地址格式的指令中, 可能有一个操作数, 也可能有两个操作数。
41. 下列三种类型的指令, SS 型执行时间最长。
42. 操作数地址存放在寄存器的寻址方式是**寄存器间接寻址**。
43. **相对寻址**对于实现程序浮动提供了较好的支持。
44. **变址寻址**便于处理数组问题。
45. **寄存器间址**有利于编制循环程序。
46. 在下列寻址方式中, **变址寻址**方式需要先计算, 再访问主存。
47. 下列叙述中, **指令执行采用流水方式、指令长度固定、只有 LOAD/STORE 指令访问存储器、配置多个通用寄存器**能反映 RISC 的特征。
48. 下列叙述中, **丰富的寻址方式、大多数指令需要多个时钟周期才能执行完成、各种指令都可以访存**能反映 CISC 的特征。

#### 填空题

1. 指令中的地址码字段(形式地址)有不同的含义, 它是通过**寻址方式**体现的, 因为通过某种方式的变换, 可以得出有效地址。常见的指令地址格式有**零地址、一地址、二地址和三地址**。



2. 在非立即寻址的一地址格式指令中，其中一个操作数通过指令的地址字段安排在**寄存器或存储器**中。
3. 在二地址格式指令中，操作数的物理位置有三种形式，它们是**寄存器-寄存器型、寄存器-存储器型和存储器-存储器型**。
4. 对于一条隐含寻址的算术运算指令，其指令字中不明确给出**操作数的地址**，其中一个操作数通常隐含在**累加器**中。
5. 立即寻址的指令其指令的地址字段指出的不是**操作数的地址**，而是**操作数本身**。
6. 寄存器直接寻址操作数在**寄存器**中，寄存器间接寻址操作数在**存储器**中，所以执行指令的速度前者比后者快。
7. 设形式地址为 X，则在直接寻址方式中，操作数的有效地址为 **X**，在间接寻址方式中，操作数的有效地址为 **(X)**；在相对寻址中，操作数的有效地址为 **(PC)+X(X 可正可负)**。
8. 变址寻址和基址寻址的区别是：基址寻址中的基址寄存器提供**基准量**，指令的地址码字段提供**位移量**。而变址寻址中的变址寄存器提供**修改量**，指令的地址码字段提供**基准量**。
9. 把两种寻址方式相结合就形成了复合寻址方式，常见的复合寻址方式可把**变址和间址**相结合，它可分为**先变址再间址和先间址再变址**两种。
10. 指令寻址的基本方式有两种，一种是**顺序寻址方式**，其指令地址由**程序计数器**给出，另一种是**跳跃寻址方式**，其指令地址由**指令本身**给出。
11. 条件转移、无条件转移、子程序调用命令、中断返回指令都属**程序控制(或跳转)**类指令，这类指令字的地址码字段指出的地址不是**操作数的地址**，而是**下一条指令的地址**。
12. 堆栈寻址需在 CPU 内设一个专用的寄存器，称为**堆栈指示器**，其内容是**栈顶的地址**。
13. 不同机器的指令系统各不相同，一个较完整的指令系统应包括**数据传送、算术逻辑运算、程序控制、输入输出、其他**等类指令。
14. 常见的数据传送类指令的功能可实现**寄存器和寄存器之间或寄存器与存储器之间**的数据传送。
15. 设指令字长等于存储字长，均为 24 位，若某指令系统可完成 108 中操作，操作码长度固定，且具有直接、间接(一次间址)、变址、基址、相对、立即等寻址方式，则在保证最大范围内直接寻址的前提下，指令字中操作码占 7 位，寻址特征占 3 位，可直接寻址的范围是  $2^{14}$ ，一次间址的范围是  $2^{24}$ 。
17. 某机采用三地址格式指令，共能完成 50 种操作，若机器可在 1K 范围内寻址，则指令字长应取 **36** 位，其中操作码占 **6** 位，地址码占 **30** 位。
18. 某机指令字长 24 位，功能完成 130 种操作，采用单地址格式可直接寻址的范围是  $2^{16}$ ，采用二地址格式指令，可直接寻址的范围是  $2^8$ 。
19. 某机共有 156 条指令，采用一地址格式，则指令字需取 **24** 位才能直接寻址 64K 个存储单元。完成一条这样格式的加法指令，需访问**两次**存储器。
20. 设指令字长等于存储字长均为 16 位，若某指令系统共能完成 58 种操作，且具有立即、间接、直接、变址四种寻址方式(编址寄存器为 32 位)，则该指令系统可直接寻址的范围是  $2^8$ ，一次间址的寻址范围是  $2^{16}$ ，变址寻址的范围是  $2^{32}$ ，立即数(有符号数)的范围是  $-2^7 \sim 2^7-1$ 。
21. 设 D 为指令字中的形式地址， $D = FCH$ ， $(D) = 40712$ ，如果采用直接寻址方式，有效地址是 **FCH**，参与操作的操作数是 **40172**。如果采用一次间接寻址，其间接地址是 **FCH**，有效地址是 **40172**，参与操作的操作数是 **(40172)**。
22. 某机器指令字长 16 位，每个操作数的地址码长 6 位，设操作码长度固定，指令分为零地址、一地址和二地址三种格式。若零地址指令有 P 种，一地址指令有 Q 种，则二地址指令最多有 **16-P-Q** 种。若按变长度操作码考虑，则二地址指令最多允许有 **15** 种。
23. 某机指令字长 32 位，共有 64 种操作，若 CPU 内有 32 位的通用寄存器，采用寄存器-存储器型指令，能直接寻址的最大主存空间是  $2^{22}$ ，如果采用通用寄存器作为基址寄存器，则寄存器-存储器型指令能寻址的最大主存空间是  $2^{32}$ 。
24. RISC 的英文全名是 **Reduced Instruction Set Computer**，它的中文含义是**精简指令系统计算机**；CISC 的英文全名是 **Complex Instruction Set Computer**，它的中文含义是**复杂指令系统计算机**。

25. RISC 指令系统选取使用频度较高的一些**简单指令**，复杂指令的功能由**简单指令**的组合来实现。其指令长度**固定**，指令格式种类少，寻址方式种类少，只有取数/存数指令访问存储器，其余指令的操作都在寄存器之间进行，且采用流水线技术，大部分指令在一个**时钟周期**时间内完成。
26. 操作数由指令直接给出的寻址方式是**立即寻址**。
27. 只有操作码没有地址码的指令称为**零地址格式指令**。
28. 在指令的执行阶段需要两次访问存储器的指令通常采用**存储器间接寻址**。
29. 需要通过计算才能获得有效地址的寻址方式常见的有**变址寻址**、**基址寻址**和**相对寻址**。
30. 在一地址的运算指令中，通常第一操作数在**累加器**中，第二操作数由**指令地址码**给出，运算结果在**累加器**中。
31. 操作数的地址直接在指令中给出的寻址方式是**直接寻址**。
32. 操作数的地址在寄存器中的寻址方式是**寄存器间接寻址**。
33. 操作数的地址在主存储器中的寻址方式是**存储器间接寻址**。
34. 操作数的地址隐含在指令的操作码中，这种寻址方式**隐含寻址**。
35. 在寄存器寻址中，指令的地址码给出**寄存编号**，而操作数在**寄存器**中。
36. 在寄存器间址寻址中，指令中给出的是**操作数地址**所在的寄存器编号。
37. 程序控制类指令包括各种转移指令，用户常用的有**无条件转移指令**、**条件转移指令**和**子程序调用指令**。
38. 基址寻址方式的操作数地址由**基址寄存器的内容**与**指令地址码字段**给出的地址(或形式地址)求和产生。
39. 相对寻址方式中操作数地址由**当前 PC 值**与**指令地址码字段**给出的**位移量**(或形式地址)求和产生。
40. **变址寻址**和**基址寻址**的有效地址形成方式极为类似，但它们的应用场合不同，前者主要用于处理数组程序，后者**支持多道程序的应用**。

### 问答题

1. 指令字中有哪些字段？各有何作用？如何确定这些字段的位数？

答：指令字中有三种字段：操作码字段、寻址特征字段和地址码字段。

操作码字段指出机器完成某种操作，其位数取决于指令系统的操作系统。

寻址特征字段指出该指令以何种方式寻找操作数的有效地址，其位数取决于寻址方式的种类。

地址码字段和寻址特征字段共同指出操作数或指令的有效地址，其位数与寻址范围有关。

2. 在寄存器-寄存器型，寄存器-存储器型和存储器-存储器型三类指令中，哪类指令的执行时间最长？哪类指令的执行时间最短？为什么？

答：这三类指令中寄存器-寄存器型指令执行速度最快，存储器-存储器执行速度最慢。

因为前者两个操作数都在寄存器中，后者两个操作数都在存储器中，而访问一次存储器所需的时间比访问一次寄存器的时间长得多。

3. 比较变址寻址和基址寻址的异同点。

答：

基址寻址	变址寻址
(1) 有效地址等于形式地址加上基址寄存器的内容	(1) 有效地址等于形式地址加上变址寄存器的内容
(2) 可扩大寻址范围	(2) 可扩大寻址范围
(3) 基址寄存器的内容由操作系统给定，且在程序的执行过程中不可变。	(3) 变址寄存器的内容由用户给定，且在程序的执行过程中可变。
(4) 支持多道程序技术的应用	(4) 用于处理数组程序

4. 设某机器共能完成 78 种操作，若指令字长为 16 位，试问单地址地址格式的指令其地址码可取几位？若想使指令的寻址范围扩大到  $2^{16}$ ，可采用什么办法？举出三种不同的例子加以说明。

答：根据 78 中操作，可求出操作码的位数为 7 位，则单地址格式的指令地址码占  $16-7=9$  位。

欲使指令的寻址范围扩大到  $2^{16}$ ，可采用以下三种寻址方法：

(1) 若指令字长等于存储字长均为 16 位，则采用间接寻址可使寻址范围扩大到  $2^{16}$ ，因为间址时(设非多次间址)从存储单元中取出的有效地址为 16 位。

(2) 采用变址寻址，并设变址寄存器 XR 为 16 位，则有效地址  $EA = (XR) + A$ (形式地址)，即可使寻址范围扩大到  $2^{16}$ 。

(3) 采用基址寻址，并设基址寄存器 BR 为 16 位，则有效地址  $EA = (BR) + A$ ，即可使寻址范围扩大到  $2^{16}$ 。

5. 某机器字长 32 位，CPU 内有 32 个 32 位的通用寄存器，设计一种能容纳 64 种操作的指令系统，设指令字长等于机器字长。

(1) 如果主存可直接或间接寻址，采用寄存器-存储器型指令，能直接寻址的最大存储空间是多少？画出指令格式。

(2) 如果采用通用寄存器作为基址寄存器，则上述寄存器-存储器型指令的指令格式有何特点？画出指令格式并指出这类指令可访问最大的存储空间。

答：(1) 根据题意指令格式如下：

OP	I	R	A
----	---	---	---

其中，OP 占 6 位，为操作码，可容纳 64 种操作；I 占 1 位，为直接/间接寻址方式(I=1 为间接寻址，I=0 为直接寻址)；R 占 5 位，为 32 个通用寄存器编号；A 占 20 位，为形式地址。

这种指令格式能直接寻址的存储空间为  $2^{20}$ 。

(2) 根据题意，保留(1)格式的 OP，I，R 字段，增加 B 字段，用以指出哪个寄存器为基址寄存器。此时，基址寻址的特征隐含在 OP 中。其指令格式如下：

OP	I	R	B	A
----	---	---	---	---

其中，OP 占 6 位，为操作码，对应 64 种操作；I 占 1 位，为直接/间接寻址方式；R 占 5 位，为 32 个通用寄存器编号；B 占 5 位，为基址寄存器编号；A 占 15 位，为形式地址。

因为通用寄存器为 32 位，用它作基址寄存器后，有效地址等于基址寄存器内容加上形式地址，可得 32 位的有效地址，故寻址范围可达到  $2^{32}$ 。

6. 若机器采用三地址格式访存指令，试问完成一条加法指令共需访问几次存储器？若该机功能完成 54 种操作，操作数可在 1K 地址范围内直接寻找，试画出该机器的指令格式。

答：

根据题意，指令字长为 36 位，其格式为

OP	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
----	----------------	----------------	----------------

其中 OP 为 6 位操作码，可完成 54 种操作；A<sub>1</sub> 占 10 位，第一操作数地址，寻址范围为 1K，A<sub>2</sub> 占 10 位，第二操作数地址，寻址范围 1K；A<sub>3</sub> 占 10 位，存放结果的地址，寻址范围 1K。

完成一次加法指令共需访问 4 次存储器：第一次取指令，第二次取第一操作数，第三次取第二操作数，第四次存放结果。

7. 某机指令格式如下图所示：

OP	I	A
0	3 4 5	7

图中 I 为间址特征位(I = 0，直接寻址；I = 1，一次间址寻址)。假设存储器部分单元有一下内容：

地址号(十六进制)	00	01	02	03	04	05	06	07
-----------	----	----	----	----	----	----	----	----

内容(十六进制)	01	5E	9D	74	A4	15	04	A0
----------	----	----	----	----	----	----	----	----

指出下列机器指令(十六进制表示)的有效地址。

(1) D7; (2) DF; (3) DE; (4) D2。

答:

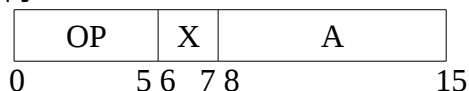
(1) 1101 0 111, 直接寻址, 有效地址为 07H

(2) 1101 1 111, 间址寻址, 有效地址为 07 地址号对应的内容 A0H

(3) 1101 1 110, 间址寻址, 有效地址为 06 地址号对应的内容 04H

(4) 1101 0 010, 直接寻址, 有效地址为 02H

8. 某机指令格式如下图所示:



图中 X 为寻址特征位, 且

当  $X = 0$  时, 不变址; 当  $X = 1$  时, 用变址寄存器  $X_1$  进行变址;  $X = 2$  时, 用变址寄存器  $X_2$  进行变址;  $X = 3$  时, 相对变址。

设  $(PC) = 1234H$ ,  $[X_1] = 0037H$ ,  $[X_2] = 1122H$ , 确定下列指令的有效地址(指令和地址均用十六进制表示):

(1) 4420H; (2) 2244; (3) 1322; (4) 3521。

答:

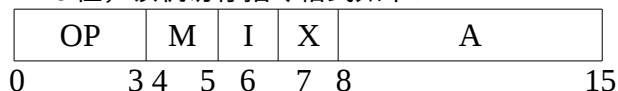
(1) 010001 00 00100000,  $X = 0$ , 不变址, 结果 0020H

(2) 001000 10 01000100,  $X = 2$ , 用变址寄存器  $X_2$  进行变址, 有效地址为  $0044 + 1122 = 1166H$

(3) 000100 11 00100010,  $X = 3$ , 相对变址, 有效地址为程序计数器的内容(当前指令的地址)和指令字中的形式地址 A 相加而成,  $1234 + 0022 = 1256H$

(4) 001101 01 00100001,  $X = 1$ , 用变址寄存器  $X_1$  进行变址, 有效地址为  $0037 + 0021 = 0058H$

9. 某机存储器容量为  $64K \times 16$  位, 该机访存指令格式如下:



其中 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;

I 为间址特征( $I = 1$  间址); X 为变址特征( $X = 1$  变址)。

设 PC 为程序计数器,  $R_X$  为变址寄存器,  $R_B$  为基址寄存器, 试问:

(1) 该指令能定义多少中操作?

(2) 立即寻址操作数的范围?

(3) 在非间址情况下, 除立即寻址外, 写出每种寻址方式计算有效地址的表达式。

(4) 设基址寄存器为 14 位, 在非变址直接基址寻址时, 指令的寻址范围是多少?

(5) 间接寻址时, 寻址范围是多少? 若允许多重间址, 寻址范围又是多少?

答:

(1) 该指令能定义 16 种操作。(操作码  $OP = 4$ )

(2) 立即寻址操作数的范围是 -128~+127(补码)

(3) 直接寻址  $EA = A$ ; 基址寻址  $EA = (R_B) + A$

变址寻址  $EA = (R_X) + A$ ; 相对寻址  $EA = (PC) + A$

(4) 非变址直接基址寻址时,  $EA = (R_B) + A$ ,  $R_B$  为 14 位, 可寻址的范围为  $2^{14}$ 。

(5) 间接寻址时, 不考虑多次间址, 寻址范围为 64K, 因为从存储器中读出的 16 位数为有效地址。如果考虑多重间址, 需用最高 1 位作多次间址标志(1 为多次间址), 此时寻址范围为 32K。

10. 一种一地址指令的格式如下所示：

OP	I	X	A
----	---	---	---

其中 I 为间址特征，X 为寻址模式，A 为形式地址。设 R 为通用寄存器，也可作为变址寄存器。在表 7.2 中填入适当的寻址方式名称。

寻址方式名称	I	X	有效地址 EA
直接寻址	0	00	$EA = A$
相对寻址	0	01	$EA = (PC) + A$
变址寻址	0	10	$EA = (R) + A$
寄存器直接寻址	0	11	$EA = R$
间接寻址	1	00	$EA = (A)$
先相对后间址	1	01	$EA = ((PC) + A)$
先变址再间址	1	10	$EA = ((R) + A)$
寄存器间接寻址	1	11	$EA = (R)$

11. 某机使用的指令格式和寻址方式如图 7.17 所示，该机有 16 个 16 位的通用寄存器，并可选定任一个通用寄存器作为变址寄存器。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。

0	6	7	8	11	12	15
OP	—	目标	源			

OP	变址	变址寄存器	源
形式地址			

OP	—	目标	
20 位地址			

试问：

- (1) CPU 完成哪一种操作花的时间最短，为什么？
- (2) CPU 完成哪一种操作花的时间最长，为什么？
- (3) 第 2 种指令的执行时间有时会等于第三种指令的执行时间吗？为什么？
- (4) 哪一种指令操作数的寻址范围最大？为什么？

答：

- (1) CPU 完成第一种指令花的时间最短，因为是 RR 型指令，执行指令时不访问存储器。
- (2) CPU 完成第二种指令所花的时间最长，因为是 RS 型指令，执行指令时需访问存储器，且要通过变址运算求得有效地址，故所需时间长。
- (3) 不可能，因为第三种指令虽需访问存储器，但不必进行地址变换运算。
- (4) 由于第三种指令的源操作数地址为 20 位的主存地址，因此它的寻址范围最大，为  $2^{20}$ 。(第二种指令通用寄存器位数和形式地址均为 16 位，第一种指令操作数在寄存器中，寻址范围为 16，最小)

12. 设机器字长、指令字长和存储字长均为 16 位，指令系统功能完成 50 种操作，采用相对寻址、间址、直接寻址。试问：

(1) 指令格式如何确定？各种寻址方式的有效地址如何形成？

(2) 能否增加其他寻址方式？说明理由。

答：

(1) 依据题意，指令格式为

6	2	8
OP	X	A

其中 OP 为操作码，6 位，可完成 50 种操作。

X 为寻址模式，2 位

X = 00，直接寻址，有效地址 EA = A；

X = 01，相对寻址，有效地址 EA = (PC) + A；

X = 10，间接寻址，有效地址 EA = (A)。

(2) 由于上述指令格式中寻址模式 X = 11 尚未使用，故可增加一种寻址方式，如立即寻址，此时 A 为操作数。

13. 设用八进制数表示下列单元地址及内容：

地址	6	11	15	17	23	2023
内容	100015	000035	000017	000023	000011	001000

寄存器 R<sub>3</sub> 中放 000015，程序计数器 PC 中放 002000(均为八进制)，试求下表中的有效地址 EA 和指令执行后 R<sub>1</sub> 或 PC 的内容(均用八进制表示)。

指令助记符	有效地址 EA	R <sub>1</sub> 或 PC 内容
① LDA 1,6	EA = 6	R <sub>1</sub> = 100015
② LDA 1, -7,3	EA = 6 (-7 + 13 = 6, R <sub>3</sub> 作为变址寄存器)	R <sub>1</sub> = 100015
③ LDA 1, @6	EA = 17 (100015, 1 表示多次间址, 15 对应 17)	R <sub>1</sub> = 000023
④ LDA 1,6,3	EA = 23 (6 + 13 = 23)	R <sub>1</sub> = 000011
⑤ LDA 1,@ 15	EA = 17 (间接寻址)	R <sub>1</sub> = 000023
⑥ LDA 1,@ 23	EA = 11	R <sub>1</sub> = 000035
⑦ LDA 1,23	EA = 23	R <sub>1</sub> = 000011
⑧ LDA 1,@ 6,3	EA = 11 (17 + 15 = 23, 再间接寻址 11)	R <sub>1</sub> = 000035
⑨ JMP * -7	EA = 1771 (002000 - 7 = 001771)	PC = 001771
⑩ JMP @ * +23	EA = 1000 (002000 + 23 = 002023, 再间接寻址)	PC = 001000

load accumulator 将数据读入累加器 ACC

说明：

(1) LDA 表示取数指令，后面的 1 表示 R<sub>1</sub>，逗号后的第一个数为形式地址(或位移量)，用八进制表示，@表示间接地址，\*表示相对寻址，第二个逗号后的 3 表示用 R<sub>3</sub> 作为变址寄存器。JMP 为无条件转移指令。

(2) 表中⑧⑩为复合寻址方式，前者为先变址再间址，后者为先相对寻址再间址。

(3) 间接访问某一存储单元时，存储字的最高位用于区分是否多次间址，低 15 位表示有效地址。如取出的数据最高位为 0，则为一次间址，如取出的数据最高位为 1，则有多次间址寻址功能。

15. 某机主存容量为  $64K \times 16$  位，并且指令字长、机器字长和存储字长相等，采用单字长一地址指令，共有 60 条。试设计四种寻址方式的指令格式，并说明每一种寻址方式的寻址范围及有效地址计算方法。

答：

根据题意指令格式如下所示：

6	2	8
OP	X	A

其中 OP 为操作码，6 位，可完成 60 种操作。

X 为寻址模式，2 位，允许有 4 种寻址方式，设计如下：

X = 00 时，直接寻址， $EA = A$ ，(256)；

X = 01 时，间接寻址， $EA = (A)$ ，(64K)

X = 10 时，变址寻址， $EA = (R_X) + A$ ，(64K)

X = 01 时，间接寻址， $EA = (R_B) + A$ ，(64K)

$R_X$  为变址寄存器(16 位)， $R_B$  为基址寄存器(16 位)，A 为形式地址。

16. 已知一台 16 位的计算机配有 16 个通用寄存器，设计一种方案，用指定的通用寄存器组中的某些寄存器来实现对 1M 地址空间的存储器寻址，参加这种寻址的通用寄存器该采用什么办法区别出来？

答：欲对 1M 地址空间寻址，必须形成 20 位的有效地址，可以指定某些通用寄存器和形式地址拼接而成。

如将 16 位通用寄存器的内容左移 4 位(低位补 0)，然后加上形式地址；或用 4 位形式地址作为有效地址的高 4 位，用 16 位通用寄存器作为有效地址的低 16 位。这两种方法都需有 1 个 20 位的 MAR。

参与这种寻址方式的通用寄存器可用赋予地址编号来加以区分。16 个通用寄存器用 4 位地址给  $R_0 \sim R_{15}$  命名，由设计者选定哪几种寄存器参与这种方式的寻址。

17. 试比较间接寻址和变址寻址。

答：间接寻址和变址寻址都可以扩大寻址范围，但它们形成有效地址的方式不同：间址需通过访存(若是多次间址还需多次访存)得到有效地址；而变址需要通过地址变换(将变址寄存器内容加上形式地址)得到有效地址，故通常间址指令执行时间比变址指令长。此外，两种指令的应用场合不同，变址寻址特别适合于处理数组问题。

18. RISC 指令系统具有哪些主要特点？

答：RISC 指令系统通过简化指令，使计算机的结构更加简单合理，并通过减少指令执行周期数的途径，达到提高机器速度的目的。其特点归纳如下：

(1) 选取使用频度较高的一些简单指令，复杂指令的功能由执行频度高的简单指令组合来实现。

(2) 指令长度固定，指令格式和寻址方式种类少。

(3) CPU 中通用寄存器数量少，大多数指令操作都在寄存器之间进行，只有取数(LOAD)和存数(STORE)指令访问存储器。

26. 设机器字长为 12 位，若主存容量为  $64K \times 12$  位，为使一条 12 位字长的转移指令能够转移到主存中的任一单元，应选用何种寻址方式？说明理由。

答：为使一条 12 位字长的转移指令能转至  $64K \times 12$  位的主存任一单元，可采用基址寻址。由于机器字长为 12 位，故可将寄存器内容左移 4 位，低位补 0，形成 16 位基址地址，然后和形式地址相加，所得地址既可访问 64K 主存的任一单元。

27. 设某机存储字长、指令字长和机器字长三者相等。若主机容量为  $256K \times 16$  位，欲使一条转移指令能够转移到主存的任一位置，可选用何种寻址方式，为什么？

答：采用扩充寻址可使 16 长的转移指令转至 256K 主存的任一单元。用 16 位字长的寄存器作为扩充地址寄存器，其内容作为高 16 位地址，再与形式地址拼接后所得的有效地址便可访问 256K 的存储空间，因为 256K 字的存储器对应 18 位地址码，而形式地址码的位数必大于 2 位。