续表

| 地址格式 | 操作 | 访存次数 | 操作数寻址范围 | 备注 |
|---|---|---|---|---|
| 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 |

由此可见,用一些硬件资源如 PC、ACC 存放指令字中须指明的地址码,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 等硬件代替指令字中的某些地址字段,还可以缩短指令字长,并可减少访存次数。究竟采用什么地址格式,必须从机器性能出发综合考虑。

(3) 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同的,同一机器的指令字长可以是固定的,也可以是可变的(按字节的倍数变化)。

2. 操作数类型

机器中常见的操作数类型有地址、数字、字符、逻辑数据等。地址可被看做一个无符号整数;数字可以是有符号数、无符号数、定点数、浮点数和十进制数;字符普遍采用 ASCII 码;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。

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,存储器可按字节、半字、字、双字访问。对于不同字长的数据,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,读者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。

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)。

## 7.2.2 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。

1. 指令寻址

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。

2. 数据寻址

数据寻址方式的种类较多,为了区别各种方式,在指令字中通常设一字段,用来指明属于哪种寻址方式。这样,指令字的地址码字段并不代表操作数的真实地址,把它称为形式地址,记为A。操作数的真实地址叫做有效地址,记为 EA,它是由寻址方式和形式地址共同确定的。由此可得指令的格式如图 7.3 所示。

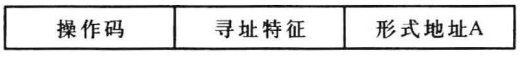



图 7.3 带有寻址特征的一地址指令格式

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

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)。图 7.4 是立即寻址示意图,图中#表示立即寻址特征。


图 7.4 立即寻址示意图

立即寻址的特点是指令在执行阶段不访存;A 的位数限制了立即数的范围。

(2) 直接寻址

直接寻址 EA=A,有效地址 EA 由形式地址 A 直接给出。图 7.5 是直接寻址示意图。

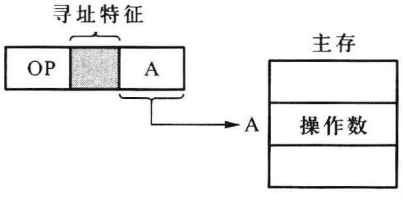



图 7.5 直接寻址示意图

直接寻址的特点是:指令在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;操作数的地址不易修改(只有修改 A 的值,才能修改操作数的地址)。

(3) 隐含寻址

隐含寻址的操作数隐含在操作码中。图 7.6 是隐含寻址示意图。

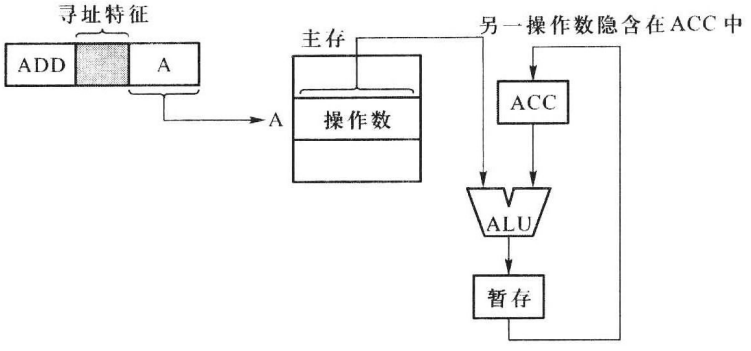



图 7.6 隐含寻址示意图

隐含寻址的特点是:因指令字中少了一个地址字段,可缩短指令字长。例如 Intel 8086 的 MUL 指令,只需指出乘数的地址,其被乘数隐含在 AX(16 位)或 AL(8 位)中。

(4) 间接寻址

间接寻址 EA=(A),有效地址 EA 由形式地址 A 间接提供。图 7.7 为间接寻址示意图。

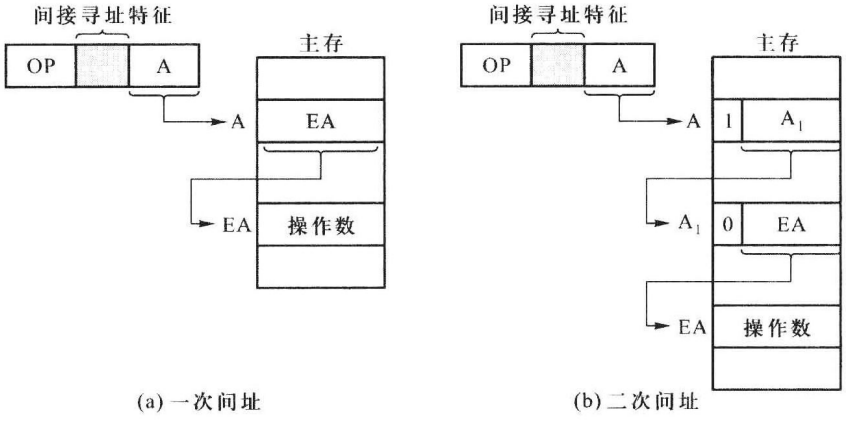



图 7.7 间接寻址示意图

间接寻址的特点是:指令在执行阶段要多次访存(一次间址需两次访存,多次间址需根据存储字的最高位确定几次访存);可扩大寻址范围(有效地址 EA 的位数大于形式地址 A 的位数);便于编制程序。

(5) 寄存器寻址

寄存器寻址 EA=$R_i$,有效地址 EA 即为寄存器编号。图 7.8 是寄存器寻址示意图。

寄存器寻址的特点是:指令在执行阶段不访存,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

寄存器间接寻址 EA=($R_i$),有效地址 EA 在寄存器中。图 7.9 是寄存器间接寻址示意图。

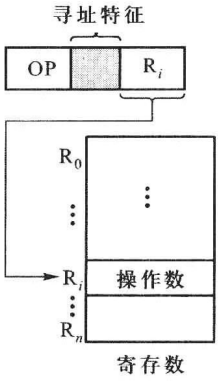



图 7.8 寄存器寻址示意图

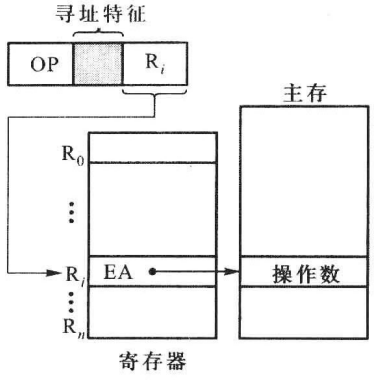
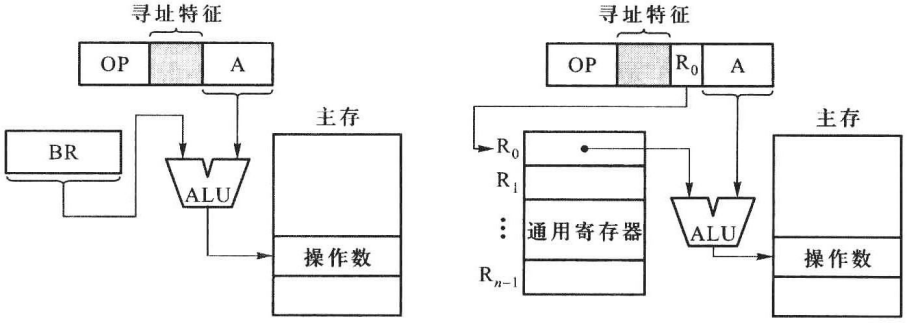



图 7.9 寄存器间接寻址示意图

寄存器间接寻址的特点是:指令的执行阶段需要访存(因有效地址在寄存器中,操作数在存储器中);便于编制循环程序。

(7) 基址寻址

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。图 7.10 是采用专用寄存器 BR 和通用寄存器作基址寄存器的两种基址寻址示意图。




(a) 专用基址寄存器 BR　　(b) 通用寄存器作基址寄存器

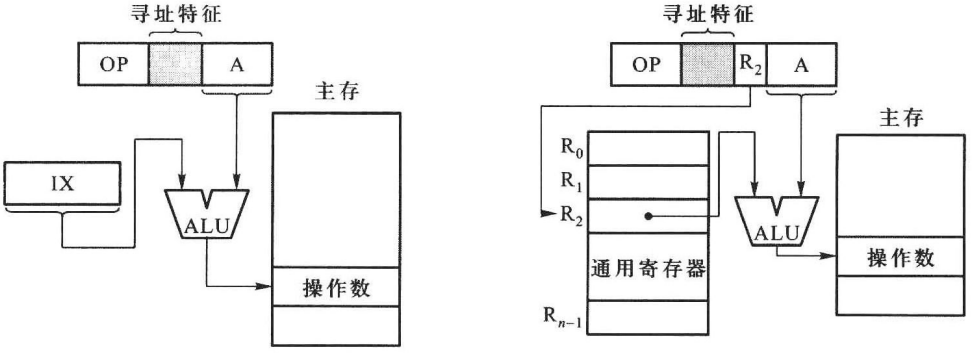
图 7.10 基址寻址示意图

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;基址寄存器的内容由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 A 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

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。图 7.11 是采用专用寄存器 IX 和通用寄存器作变址寄存器的两种变址寻址示意图。




(a) 专用变址寄存器 IX　　(b) 通用寄存器作变址寄存器

图 7.11　变址寻址示意图

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）；便于处理数组问题。

（9）相对寻址

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。图 7.12为相对寻址示意图。

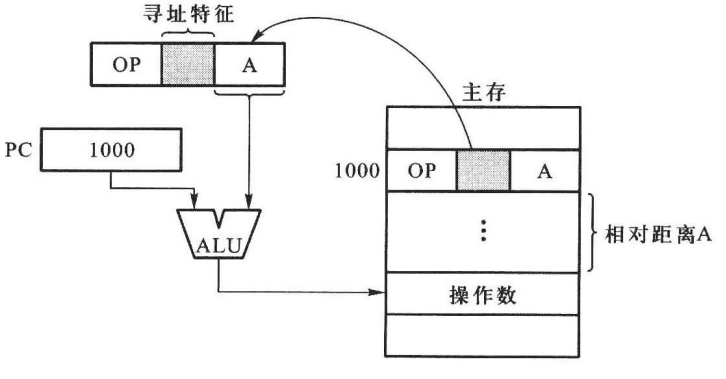



图 7.12　相对寻址示意图

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。

(10) 堆栈寻址

多个寄存器可构成硬堆栈,指定的存储空间可构成软堆栈,其特点是先进后出。堆栈的栈顶地址由 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→SP,出栈 (SP)+1→SP。图 7.13(a)~(d)为堆栈寻址示意图。

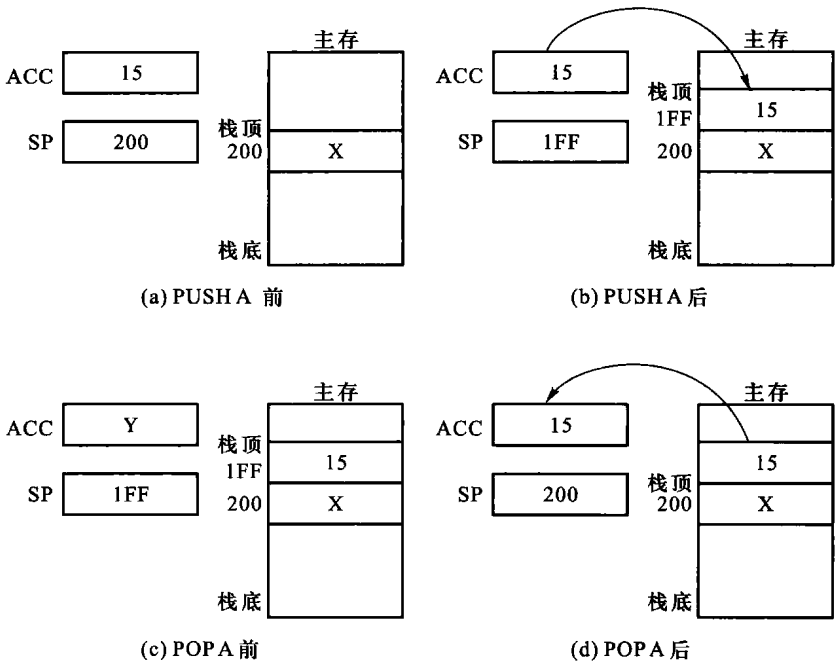



图 7.13 堆栈寻址示意图

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)-Δ→SP,出栈 (SP)+Δ→SP。Δ 取值与主存编址方式有关,若按字编址,Δ 取 1;若按字节编址,当字长为 16 位时,Δ 取 2,当字长为 32 位时,Δ 取 4。

在上述 10 种寻址方式的基础上,可进一步扩展成其他的各种寻址方式,如先间址再变址、先变址再间址以及既变址又基址等。掌握机器指令的寻址方式对于用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。