

指令：指示计算机执行某些操作的命令

微指令：微程序级的命令，属于硬件

宏指令：由若干条机器指令组成的软件指令，属于软件

机器指令：介于微指令与宏指令之间，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

一个完整的指令系统应满足：完备性、有效性、规整性、兼容性。

完备性：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，而不必用软件来实现。完备性要求指令系统丰富、功能齐全、使用方便。

有效性：利用该指令系统编写的程序能够高效率的运行。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，一个功能更完善的指令系统，必定有更好地有效性。

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。

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。

指令格式：操作码字段 + 地址码字段

根据有几个操作数地址可分为：四、三、二、一、零地址指令格式

从存放操作数的物理位置来划分，指令格式分为：存储器-存储器（**SS**）型指令、寄存器 - 寄存器（**RR**）型指令、寄存器-存储器（**RS**）型指令。

分析指令、设计指令：见例题、作业、习题集

指令寻址方式：

顺序寻址方式：

指令在内存存放方式是按地址顺序安排的，执行程序时，通常是一条指令接一条指令顺序进行。这种按指令地址顺序执行的过程成为指令的顺序寻址方式。

跳跃寻址方式：

当程序要改变执行的顺序时，指令的寻址就改用跳跃寻址方式。所谓跳跃，又称跳转，是指下一条指令的地址由本条指令修改 **PC** 后给出。采用跳跃寻址方式，可以实现程序专业或者构成循环程序，从而能缩短程序长度，获奖某些程序作为公共程序引用。

操作数寻址方式：

指令字：**OPCODE** + 寻址方式特征 **MOD** + 形式地址 **A**

形式地址 **A**：指令字结构中给定的地址量。

寻址方式特征位：

通常由间址位(**I**)和变址位(**X**)组成。

若指令无间址和变址要求，则形式地址就是操作数的有效地址；

若指令中指明要进行变址或间址变换，则形式地址就不是操作数的有效地址，而必须按指定方式进行变换，才能形成有效地址。

1.隐含寻址：在指令中不明显给出而是隐含着操作数的地址。

2.立即寻址：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。

$EA = (PC)$

3.寄存器寻址：操作数所在的寄存器编号存放在指令的地址字段 **A** 中。

$EA = Ri$

4.直接寻址：操作数所在的存储器单元的地址存放在指令的地址字段 **A** 中

$EA = A$

1)寄存器直接寻址：指令地址码字段直接给出所需操作数在通用寄存器中地址编号

2)存储器直接寻址：同直接寻址

5.间接寻址：指令地址字段中的形式地址 **A** 不是操作数的真正地址，而是操作数地址的指示器，**A** 单元的内容是操作数的有效地址，即 **A** 为操作数地址的地址。

$EA = (A)$

1)寄存器间接寻址：先访问寄存器，从寄存器中读出操作数地址，在访问存储器取得操作数。

2)存储器间接寻址：访问两次存储器得到数据。第一次先从存储器读出操作数地址，第二次根据的操作数地址再取出真正的操作数。

6.相对寻址方式：根据一个基准地址及其相对量来寻找操作数的地址。

$EA = (PC) + Disp$

PC 相对寻址是指将 **PC** 的内容（即当前执行指令的地址）与地址码部分给出的位移量通过加法器相加，所得之和作为操作数的有效地址。

7.基址寻址方式：计算机设置一个寄存器，专门用来存放基准地址，该寄存器就是基址寄存器。先将指令地址码给出的地址 **A** 与基址寄存器 **RB** 的内容通过加法器相加，所得的和作为有效地址，再从存储器中读出所需的操作数。

$EA = (RB) + Disp$

8.变址寻址方式与基址寄存器区别：基准寄存器提供地址基准量而指令提供位移量，变址寄存器提供修改量而指令提供基准量。

$EA = (Ri) + Disp$