

## 第四讲 中央处理器的指令系统与寻址方式

曾维达

### 摘要

不同类型的中央处理器(CPU)具有不同的指令系统，但也有其共性。本文以Apple机的CPU-R6502为例，分析了指令系统的分类与功能，读者可以此类推去分析和掌握其它型号的CPU指令系统。

### 一、概述

利用CPU的指令系统编制程序，命令计算机进行某些控制、采样、计算等等，是计算机用户经常会遇到的课题之一。因此了解和掌握CPU的指令系统，用以编制程序也是计算机用户应掌握的一种技能。

任何一种CPU均应根据使用方向来考虑其总体结构，并设计其指令系统。指令系统功能的强弱取决于CPU的内部结构，也与系统的设计安排有关。

以Apple机的CPU-R6502为例，它是一种较低档的CPU，是在MOTOROLA公司的MC6800基础上改进设计的。其内部寄存器较少且均为8位。除用2个8位寄存器构成1个程序寄存器PC外，规定堆栈指针只记低8位，其高8位地址由计算机系统设计时设定，现定为02(即第2页)。除累加器A外，只有X和Y2个变址寄存器。为弥补内部寄存器少的缺陷，采用了两种办法：增加寻址方式使程序编制更为灵活；允许内部存储器直接参加运算。这样不仅简化电路，且减少了把数从存储器传送到寄存器去的步骤，节省了指令执行时间，加快了运算速度。因此其基本指令只有56条，变化

不同的寻址方式，组合后的指令可达150条左右，且每条指令的执行时间在只有1.023MHz主频的条件下，仅需2—7μs。

迄今为止，计算机设计的原则还是把程序和数据都看成是数据流，它们被存放于计算机的内存中，程序执行的方式是根据PC的指示，串行地不断重复下述过程，直至程序结束(图1)。

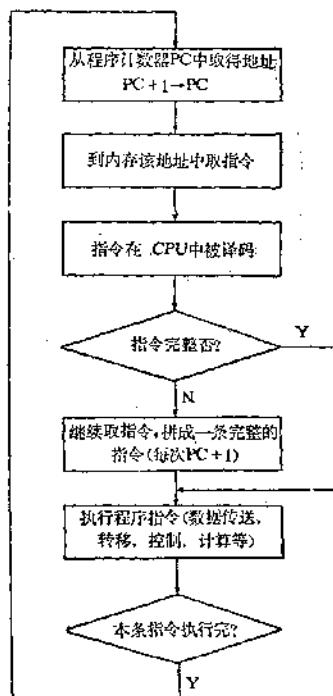


图1 程序执行的方式

由此可见，CPU的指令系统至少应具备对数据的存取、传送和移动，算术和逻辑运算，以及对程序运行的控制功能。无论是取指令还是运算都离不开数据的传送，因此指令

系统中数据传送指令是否灵活和快速，是衡量指令系统性能是否良好的重要标志之一。

数据主要存储在内存，取指令和运算则主要在CPU内部进行。数据的传送方式有下述三种：

1. 直接送—数据入CPU内某一寄存器中；

2. 寄存器间传送或交换数据；

3. 寄存器和内存之间数据传送或交换。

## 二、寻址方式

R6502有10类12种寻址方式：

### 1. 隐含寻址

CPU内某一寄存器的数据送往(Transmission)CPU内的另一寄存器。它不与内存联系，因而也无需内存寻址。指令的助记符名称已直观的表达了指令的含义。如TSX，指将CPU中SP寄存器中之数传向X变址寄存器。

### 2. 立即寻址

程序指定将某一数值( $\leq 255$ )以16进制形式直接送往某个寄存器。数值前加“#”标志。因直接指定数据及寄存器，也不必到内存中去寻址，故称立即寻址，该数称立即数，助记符以“Load”的缩写LD表示。如指令LD A # \$11，说明向累加器A送16进制数11。

### 3. 零页寻址

每个微机系统都将内存分页，这是为了便于寻址。8位微机系统的64K内存，一般以256个内存单元作为1页。其中00H—FFH单元称为0页，属系统的专用区。

如欲将0页中某内存单元的内容送入累加器A，则可写指令LD A 35H，表明将内存35H单元中的内容送往累加器A。这种0页寻址方式对将系统工作区工作单元的值送往寄存器中去是很方便的。

### 4. 绝对寻址

指明64K内存地址中的某一单元，令将其内容送往某寄存器时，将低8位地址写在

前面，高8位地址写在后面，以16进制方式书写。如LD A 5136，表示将3651H单元的内容送往A。

这种绝对寻址和上面的0页寻址都属直接寻址，所不同的是0页其高8位地址恒为0故不必写出，且前者为3字节指令，0页寻址则为2字节指令。

### 5. 相对寻址

指令中指出转移地址的偏移量，它是相对于当前的地址而言的，故称相对寻址。相对寻址的地址偏移量，规定只能在向前128个单元或向后128个单元的范围之内，即在前后共1页的范围内可用，规定凡偏移量为80H—FFH，被认为是要往后转移；偏移量为01H—7FH，被认为是要向前转移。这种寻址方式多用于条件转移指令。

### 6. 间接寻址

间接寻址是从指明的地址单元中找到应转往的低8位地址，再从该地址单元的下一单元找到应转往的高8位地址，两者拼成的地址才是程序应转去的地址。

间接寻址的方法虽较复杂，但指令较灵活。如同一条无条件转移指令JMP (OA)(OB)，要求程序转往由OBOA、OBOB单元的内容拼成的地址中。设内存的内容为：

OBOA 00H

OBOB 10H

则程序将转往1000H单元去。如将OBOA和OBOB的内容改变了，则执行同一条程序，程序转向也大不相同。

### 7. 零页(X或Y)变址

所谓0页变址是指将X或Y变址寄存器的内容与0页中某地址单元的地址相加得一个新的地址，取该地址的内容供传送。设X变址寄存器的内容为15H，有指令为LD A 80，X，则先计算 $80H + 15H = 95H$ ，取95H单元的内容送A。

变址方式比直接寻址要慢。这是因为必须先执行一次加法后才能获得地址。如改变X的内容，同一条指令就可以转至不同内存

地址取数送A，这样程序可较灵活。

#### 8. 绝对(X和Y)变址

本寻址方式的地址为绝对地址，如指令 LD A 00 10, X，说明需将X的内容加在1000H上求得新地址，将该地址内容送A。设X内容为30H，则应将1030H的内容送A。

这种寻址方式适用于X和Y变址寄存器。

#### 9. X变址间接寻址

指令LD A (A3, X)要求将X变址寄存器的内容与0页地址A3相加，设X内容为12H，即得B5H。然后从B5H和B6H两单元中找出低地址与高地址，再取此拼合地址的内容送A。即

$$X = 12H \quad \text{与 } A3 \text{ 相加得 } B5H$$

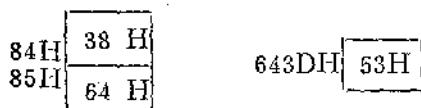


上述指令执行的结果，A内容为40H。

Y变址寄存器不能用此法寻址。

#### 10. 间址、Y变址寻址

此方式指令的书写与上近似但不相同。如LD A(84), Y设Y为05H, ( )中为0页中某地址，但其求地址之法是先从84H与85H中找出新地址，再与Y内容相加，即先间接寻址后再变址，获得新地址后取值送A。设



指令执行后A为53H。

上述10类12种寻址方式中，不少涉及0页地址，看来0页似只有00H—FFH共256个单元，范围很小，实际上经过相对寻址、间址、变址等寻址方式，可以对64K内存中的任一地址单元寻址、取放数。这正好反映了多样化的寻址方式可以弥补其功能的不足。

要掌握与熟悉任一种CPU，了解该指

令系统的寻址方式是必需的。一般说来，同字长的CPU，其内部结构越复杂、功能越强，其寻址方式反而较简单；内部结构简单的，其寻址方式则应多样化以增加编程的灵活性。对不同字长的CPU来说，字长越长，其直接寻址的范围越宽。为避免每次都在大范围的内存空间中去寻址，需将内存分段再分页，寻址时先确定段地址再找具体地址。如16位的CPU-8086，其寻址方式就是考虑段地址的。

### 三、指令系统

微机的CPU指令都是微程序形式的，其指令的长度一般由1—3（或4）个字节组成。为了便于识别记忆，这些指令又以助记符的形式表示。助记符往往取相应含义的英文字头来表示。每条指令有一部份用以说明所执行的操作要求，称操作码；另一部份说明该种操作所用数据的来源及结果送往何处等，统称为操作数。它可以是地址、寄存器名，也可以是数据。即指令的形式可分为下列几种之一：

操作码

操作码 操作数

操作码 操作数 操作数

无论何种形式的指令其第一个字节都是操作码，因此每次取指令译码时，总可以根据指令的第一个字节确定是何种操作，是否还需继续取操作数以及确定取几个操作数。

为了便于记忆指令，习惯上都把指令按操作目的分类。如R6502的56条指令就可以按操作目的分为三大类：

数据的存取、传送和移动

算术和逻辑运算

程序运行的控制

每大类中又可细分为：

1. 数据的存取、传送和移动指令（20条）

（1）取数 从内存单元或某个寄存器中取数，送往累加器A或变址寄存器X、Y

中去。以LD表示向某寄存器“加载”。指令有三条：LD A、LD X、LD Y。

(2) 存数 把变址寄存器或累加器的内容送往某内存单元中去。以存贮的字头ST表示。指令有三条：STA、STX、STY。

(3) 传送 在CPU内部的寄存器和累加器中数据可相互传送。规定以传送的字头T表示，送出数据的称源寄存器写在前面，接收数据的寄存器为目的寄存器写在后面。指令有六条：TAX、TXA、TAY、TYA、TSX、TXS。累加器A可以和X或Y传送数据，而堆栈指针寄存器SP只能和X交换数据。

(4) 堆栈操作 程序运行遇有分支时，应将现场保存于堆栈，待分支处理完毕后，再把压入堆栈中的内容弹出，称为堆栈操作。R6502规定累加器A和标志寄存器PSW可作堆栈操作，即压入(PUSH)和弹出(PLAYBACK)。这四条指令是：PHA、PHP、PLA、PLP。

(5) 移位 分算术移位——移出之位以0填充，和逻辑移位——无论是寄存器还是内存单元均以8位的数值循环填充。执行移位时，标志寄存器PSW中的进位位C也参与移位，以便根据移位来的值，决定程序下一步应如何分支。R6502的移位指令有四条：ROL、ROR、ASL、LSR(图2)。

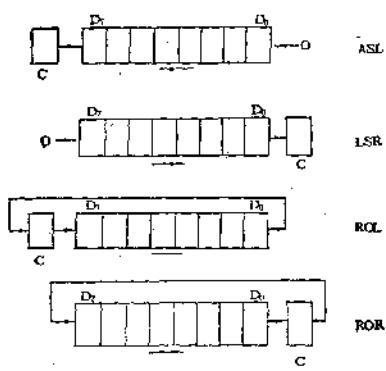


图2 移位指令示意图

## 2. 算术与逻辑运算指令(19条)

(1) 累加器的加减运算 加减运算会出现进位与借位，因此将存贮器内容与累加器内容相加(ADD)减(SUB)时应连同进位位C一起进行。二条指令为：ADC、SBC。

(2) 存贮器或X、Y寄存器内容的增1或减1 用增量与减量的字头表示，共有六条指令：INC、INX、INY、DEC、DEX、DEY。

每执行一次该指令，该寄存器的内容增1或减1。

(3) 存贮器内容与A、X、Y内容的比较 比较时是以A、X、Y的内容与存贮器的内容作一次减法比较，两者实际内容并不改变，比较结果产生的正、负、进位、借位或零将影响标志寄存器中的N、Z、C的值。三条指令为：CMP、CPX、CPY。

(4) 存贮器和累加器A的内容进行逻辑运算 运算时采用逐位相与、相或、相异或的方法。三条指令为：AND、ORA、EOR。

(5) 标志寄存器的置位和消除 能置位和清除的位有进位位C、十进位D、溢出位V。五条指令为：SEC、CLC、SED、CLD、CLV。

## 3. 程序运行控制指令(17条)

(1) 无条件转移指令JMP和转向子程序的返址指令JSR。

(2) 条件转移指令 根据标志寄存器中的C、Z、N、V是清(C)还是置(S)来决定程序是否转移。连同强迫中断BRK共有九条指令：

BCC (C = 0)	BCS (C = 1)
BNE (Z = 0)	BEQ (Z = 1)
BLP (N = 0)	BMI (N = 1)
BVC (V = 0)	BVS (V = 1)

(3) 中断屏蔽位I置位和清除指令：SEI、CLI。

(4) 中断返回和子程序返回指令：

## RTI、RTS。

(5) 位测试指令 BIT 存贮器的内容与累加器内容逐位相与，两者内容都不改变，但存贮器中该地址最高两位相与的结果分别顺序存入标志位 N 和 V 中，供程序判别。

(6) 空操作指令 NOP CPU 不作其它操作，仅程序计数器 PC 内容加 1。

为使读者对 R6502 的基本指令有个更清晰的概念，图 3 列出了它们的关系。对于衍

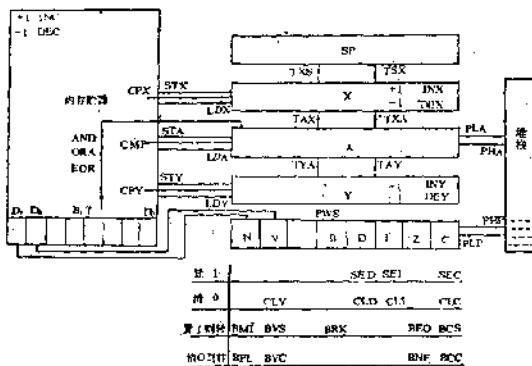


图 3 R6502 部份基本指令示意图

生的 150 条指令本文从略。

掌握了一种 CPU 指令后，如遇其它类型的 CPU，只要能弄清 CPU 的结构，了解其寻址方式，再经实际编程的锻炼，熟悉另一种 CPU 的指令也是不难的。目前微机的 CPU 已形成了几种优选系列，如 8 位的 CPU——8080/8085、Z80、MC6800 等。尽管 CPU 型号不下数百种，指令的助记符也不尽相同，但都与这几种优选系列相似或兼容。读者如掌握了上述几种主系列的指令系统，分析与掌握其它的 CPU 指令系统便不会有太多的困难。

## 我国海洋气象导航 科研成果通过鉴定

大连海洋气象导航试验协作组，研制了一套系统的导航技术方法，其中包括，强风区预报、风浪转换计算、船舶性能曲线的确定、数据库的建立、最佳航线决策、跟踪导航和航线综合分析等。1985 年 3 月 22 日，由国家气象局、上海海运学院、海军司令部航保部、农牧渔业部、辽宁省气象局、大连水产学院、海军水面舰艇学院、大连海运学院和广州远洋运输公司等单位的专家组成鉴定小组，对该科研成果进行了鉴定，认为其“理论可靠、计算方法可靠”，对于航行安全和航行的经济效益“提供了具有实用价值的决策技术”，“填补了我国海洋气象导航技术领域的空白”。

海洋气象导航亦称确定最佳航线。它在保障船舶安全、缩短航时、节省燃料、减少船损和货损等方面有重要作用。由于过去国内没有开展气象导航业务，各远洋船舶所采用的气象导航，一直是支付外汇向外国导航公司申请的。

1981年11月起，由大连市气象台、海军水面舰艇学院、大连水产学院、大连远洋运输公司和大连海运学院等单位科技人员协作，开始了气象导航的试验研究工作。1983年9—11月，在初步完成图上作业和模拟对比试验的基础上，对大连远洋运输公司“镜泊湖”轮往返大连至美国旧金山和“阳澄湖”轮自旧金山至日本的三个航次进行了气象导航实船对比试验，初步摸索了在我国现有条件下开展海洋气象导航的途径。

1984年10—12月又对天津远洋运输公司的三万吨级“莺歌海”轮往返天津至温哥华（到 $170^{\circ}\text{E}$ 后改驶太子港）和大连远洋运输公司的一万五千吨级“纳木湖”轮往返大连至美国西雅图进行了第二次气象导航实船对比试验。试验证明，其结果与全球营业范围最大的美国航路公司的导航结果接近。

（王奉安）