

汇编语言程序设计 学习指导书

李兆凤 刘晓星 编

中央广播电视大学出版社

汇编语言程序设计

学习指导

李兆凤 刘晓星 编

中央广播电视大学出版社

汇编语言程序设计学习指导

李兆凤 刘晓星 编

*

中央广播电视大学出版社出版

新华书店北京发行所发行

北京印刷二厂印装

*

开本 787×1092 1/16 印张 10 千字 246

1988年10月第1版 1989年4月第1次印刷

印数 1—13000

定价 2.10 元

ISBN7-304-00336-7/TP·23

前 言

本书是中央电大出版社出版的教材《汇编语言程序设计》的配套指导书,书中对教材内容按章节进行了系统性指导。每章内容包括两大部分:第一部分为内容要点和基本要求;第二部分为部分例题与习题选解,它对程序设计的具体实例进行了分析,给出了典型的分析方法和操作技能。希望本材料对学生学习本课程有所帮助。

编 者

1988年10月

目 录

第一章 概述.....	1
第二章 Z-80CPU 的结构	3
第三章 Z-80 指令系统	6
第四章 汇编语言和汇编程序	25
第五章 基本程序设计	28
第六章 算术运算程序设计	67
第七章 非数值处理程序设计	80
第八章 输入/输出程序设计	108
第九章 中断程序设计.....	118
第十章 宏指令和条件汇编.....	131
附录一.....	132
附录二.....	133
附录三.....	144
附录四.....	153

第一章 概 述

内容要点和基本要求

本章是本门课程的基本知识部分,是一般了解的内容,但其中的概念术语必须掌握。

第一节 引 言

这一节中,主要讲述了微处理机、微型计算机系统和软件等内容,我们将必须掌握的几个概念术语叙述如下:

微处理机:将运算器和控制器制作在一块半导体芯片上的集成电路器件,称为微处理机。也称微处理器。

微型计算机:由微处理机配上存贮器、输入/输出接口电路等芯片,由总线联结构成的设备称为微型计算机。

微型计算机系统:微型计算机配上外部设备、电源和软件,则构成一个可以独立工作的完整的计算机整体,我们称之为微型计算机系统。

单板计算机:将微处理器、一定容量的存贮器和输入/输出接口电路装在一块印刷电路板上,就称为单板计算机。

单片计算机:将 CPU 与存贮器、输入/输出接口电路制作在同一块集成电路芯片上,就称为单片计算机。

软件:使计算机能够正常运行和进行运算操作的各种程序的总和,称为软件。一般来说,软件应当包括:

系统软件

程序设计语言

应用软件

等三部分。

第二节 汇编语言程序设计

这一节中主要包括汇编语言的概念、性质、作用、优缺点和高级语言的定义以及汇编语言与高级语言的比较等内容。这一节中主要应当掌握汇编语言和高级语言的定义和它们各自的优缺点。这一节中主要的概念定义如下:

汇编语言:使用助记符指令代替机器指令而形成的语言称为汇编语言。

高级语言:高级语言是一种脱离具体计算机、面向过程、更符合人们思维和更为人所理解与学习的语言。

程序,使计算机执行特定任务的一系列指令语句的集合称为程序。

汇编语言的优点有:比机器语言容易理解、记忆和便于交流;执行速度快。缺点是:比高级语言难理解;需要使用者熟练掌握机器内部结构;程序冗长、易于出错、调试困难、不通用等。

高级语言的优点有:容易理解、便于书写、不易出错、调试简单、通用性强、便于交流和移植以及具有较强的文件处理功能等。

第三节 如何学习汇编语言程序设计

这一节主要为大家在学习时提供一个捷径,换句话说就是提供一种学习方法,希望大家认真阅读、领会。

第二章 Z-80 CPU 的结构

内容要点和基本要求

本章共有两节内容,其中第一节的内容是必须重点掌握的。第二节作为一般了解。现将每节的内容要点分述如下。

第一节 Z-80 CPU 的结构组成

对于初学者来说,Z-80 CPU 的组成结构是非常复杂的,但是因为我们所学习的汇编语言是面向机器的语言,必须要求对机器本身的内部结构有比较清楚的了解,因此,弄清 CPU 的组成结构是先决条件。

内部寄存器

Z-80CPU 的内部寄存器由 18 个 8 位寄存器和 4 个 16 位寄存器组成,一般将它们分为两类:

1. 通用寄存器组

主寄存器组		备用寄存器组	
A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

其中 A 也称为累加器, F 称为标志寄存器。这些寄存器都是 8 位的。也可以两两组成一个 16 位的寄存器对,共可以组成 AF、BC、DE、HL、A'F'、B'C'、D'E' 和 H'L' 8 个寄存器对。主寄存器组与备用寄存器组之间的信息交换可通过交换指令来完成。

2. 特殊用途寄存器组

特殊用途寄存器组也称专用寄存器组。故名思义,它们中间的每一个寄存器都有其特定的用途。

中断向量 寄存器	I	存储器刷 新寄存器	R
变址寄存器		IX	
变址寄存器		IY	
堆栈指针		SP	
程序计数器		PC	

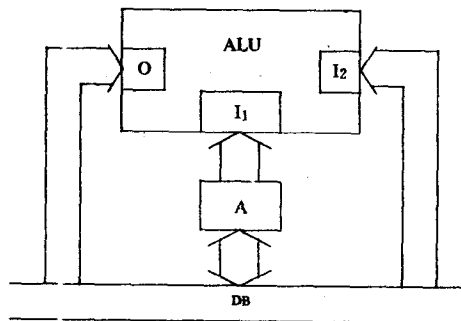
其中除 PC 外,均可以用指令置数。

算术、逻辑运算单元 ALU

CPU 中对于数的操作大都依靠 ALU 来完成。它是 CPU 的重要组成部分之一。Z-80CPU 的 ALU 具有以下功能:

加法运算、减法运算、逻辑与(AND)、逻辑或(OR)、逻辑非(NOT)、异或(EXOR)、比较、左右移位和循环移位、增 1 和减 1、置位、复位、位校验和十进制调整等。

ALU 的操作示意图如下:



图中 I_1 和 I_2 表示参加操作的数的入口, O 表示操作结果的出口。

指令寄存器和指令译码器

指令寄存器 IR 和指令译码器 ID 是 CPU 中两个比较重要的寄存器,所有的指令都必须先经过这两个寄存器,然后才能产生各种微操作命令以完成各种操作。对于指令寄存器和指令译码器,必须掌握它们各自的功能、缩写、区别和相互关系。

定时及系统控制信号电路

所谓 Z-80CPU 的定时,就是说 Z-80CPU 动作必须按照规定的时间来实行。Z-80 时钟最高工作频率为 2.5MHz。对于系统控制信号电路部分内容可作一般了解。

Z-80CPU 芯片

Z-80CPU 芯片为 40 脚双列直插式集成电路。Z-80CPU 信号共分为三大类:

地址信号($A_0 \sim A_{15}$)

数据信号($D_0 \sim D_7$)

控制信号(其中包括系统控制信号 $\overline{M}_1$ 、RD、WR、MREQ、RPSH、IORQ; CPU 控制信号 HALT、WAIT、INT、NMI、RESET; 总线控制信号 $\overline{BUSRQ}$ 、 $\overline{BUSAK}$ 。)

所有这些信号均是低电平有效。

第二节 Z-80 的时序

这一节中只需重点掌握以下几个概念：

1. 时序
2. 指令周期
3. 机器周期
4. T 周期(也称 T 状态)
5. Z-80CPU 的基本典型时序有7种。

第三章 Z-80 指令系统

内容要点和基本要求

本章主要讲授了 Z-80 的十种寻址方式和 Z-80 的指令系统,以及标志寄存器 F 的作用。是需要重点掌握的内容。大家在刚开始学习寻址方式和指令系统时,不可能立即掌握,需要通过多次甚至是几十次的重复记忆才会掌握。因此,必须多看书,多作练习。学到哪里就记到哪里,暂时用不到的可以先不记,等需要时返回来。比如零页寻址,待学习中断概念时再记不迟。

下面,我们将本章各节的内容要点分述如下。

第一节 Z-80 的寻址方式

一般说来,输入计算机的数据将存于 CPU 内部寄存器、外部存储器或 I/O 接口三处当中之一处。CPU 要对这些数据(操作数)进行处理,就必须首先知道这些数据都各自在什么地方,其地址如何,寻址方式就是指在指令中数据地址的形成方式。寻址方式的多少也是衡量一台计算机性能的重要因素之一。寻址方式越多,则计算机功能就越强,使用也就越灵活。

Z-80CPU 共有十种寻址方式,都必须熟练掌握。下面我们将简明扼要地总结出 Z-80 的寻址方式及其优缺点,以帮助加深理解,为程序设计打好基础。

1. 立即寻址

这种寻址方式最为简单,操作数就包含在指令当中,并且直接参加指令所规定的操作。在存贮形式上,操作数直接跟在操作码之后,如图3-1所示。

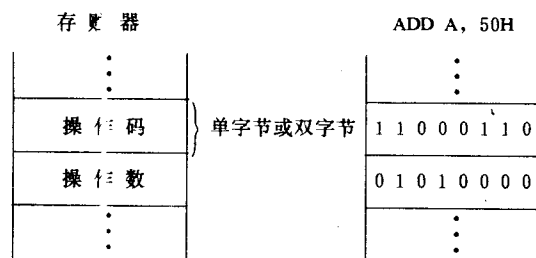


图3-1

立即寻址方式经常用于给寄存器或计数器置初值,或在计算机中处理一些常数。立即寻址方式在取指令的同时也取出数据,故指令执行速度较快。但由于数据是固定的,使用起来就不够灵活。

2. 立即扩展寻址

这种方式是立即寻址方式的扩充形式。处理的是16位的操作数。其存贮格式如图3-2所示。

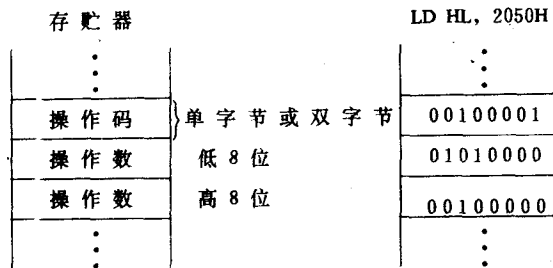


图3-2

3. 寄存器寻址

在这种寻址方式中,是对存于通用寄存器中的数据进行处理。例如指令

ADD A,H

是把寄存器 H 中的内容(操作数)与累加器 A 中的内容相加,结果送回 A 中。寄存器寻址主要用在需要多次重复使用寄存器中某个数据的程序中,这样可以节省很多访问存贮器的时间,而且指令字节数少,执行速度快,比立即寻址灵活了许多。

4. 直接寻址

直接寻址方式是指在指令中包含有操作数的存贮器地址,由于地址值是16位的,固而必须用两个字节来表示。例如指令:

LD A,(3FFFH)

的存贮格式如图3-3所示。

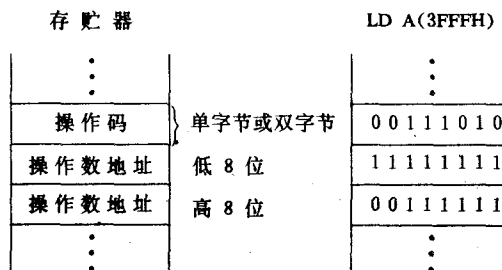


图3-3

直接寻址方式只能给出一个固定的地址,修改起来不太方便,所以在处理数组数据时缺乏灵活性,但在取出指令后只需访问一次存贮器,即可取出操作数,所以其执行速度还是比较快的。

5. 寄存器间接寻址

这种寻址方式与寄存器寻址方式相似。所不同的是寄存器中存放的不是操作数,而是操作数所在的内存单元地址,并且使用的是寄存器对。例如指令:

ADD A,(HL)

是以寄存器对 HL 中的内容作为操作数的地址,取出操作数与累加器 A 里的内容相加,结果送回 A 中。这种寻址方式的执行速度较快(一字节),而且便于在通用寄存器中修改地址,以适应大批量数据传送的需要。

6. 变址寻址

INC (IX+90)

7. 相对寻址

JR Z,e

Diagram illustrating the instruction format for the 68000 microprocessor, showing the relationship between the instruction fields and the PC (Program Counter) value.

The instruction is divided into three main sections:

- Operation Code (操作码):** 4 bits (bits 31-28).
- Displacement (偏移量):** 16 bits (bits 31-16).
- PC-Relative Displacement:** 12 bits (bits 31-20).

The Displacement field is further divided into two 8-bit sub-fields:

- Upper 8 bits:** Labeled "2 8".
- Lower 8 bits:** Labeled "e - 2".

The PC-Relative Displacement field is further divided into two 6-bit sub-fields:

- Upper 6 bits:** Labeled "Z=0 则 * * * * + 2".
- Lower 6 bits:** Labeled "Z=1 则 * * * * + e".

The PC-Relative Displacement field is also labeled "PC".

The PC value is shown as a 4-bit field: * * * *

图 3-4

BIT 1,C

1 1 0 0 1 0 1 1
0 1 0 0 1 0 0 1

该三位的组合代表寄存器C

该三位的组合代表被操作的是第1位

• 8 •

的寄存器,例如指令

SUB C

中就隐含了存放被减数的累加器 A。

10. 零页寻址

在零页寻址方式中,就是利用专门的单字节调用指令,使程序转到存储器0页面上固定的8个位置之一。这种指令只用单字节就可以说明被调用子程序的16位地址,这样就节省了存储器空间并且提高了效率。例如指令:

RST 38H

的指令字为 FFH,本条指令执行后 0038H $\Rightarrow$ PC,使程序转到 0038H 继续执行。

我们应该看到,很多指令的操作数不止一个(比如加法指令,传送指令等),在这些指令中多使用两种寻址方式的组合。例如,在传送指令中可以用立即寻址说明源地址,同时又用寄存器间接寻址或变址寻址说明目的地址。指令:

LD (HL),32H

的操作是将立即数 32H 送至寄存器对 HL 所指定的内存地址单元中去。对源地址而言是立即寻址,而对目的地址而言却是寄存器间接寻址。因此,对寻址方式的判断不能仅看源地址或目的地址,而是要全面考虑。

第二节 Z-80 的指令系统

某种 CPU 所具有的全部指令的集合就称为指令系统。由于各种微型所使用的 CPU 不同,因此其指令系统也就不一样。但是它们的设计思想和分类却是相似的。我们所学习的是 Z-80 指令系统。

Z-80 指令系统共有 158 条指令,是 8 位 CPU 中最多的。在这些指令当中,有些是经常使用的,如数据传送和算术、逻辑运算指令等。按照指令作用的不同,可以大致分为以下八种类型:

1. 数据传送和交换指令(33条);
2. 数据块传送和搜索指令(8条);
3. 算术和逻辑运算指令(52条);
4. 循环和移位指令(23条);
5. 位操作指令(9条);
6. 转移、调用和返回指令(14条);
7. 输入/输出指令(12条);
8. CPU 基本控制指令(7条)。

上述八类指令中,绝大部分被用于处理存放在 CPU 内部寄存器、存储器或输入/输出接口中的数据,且每种指令都有其相应的寻址方式。为了帮助大家尽快地掌握 Z-80 的指令系统,我们先将这八种类型指令的功能和特点归纳如下:

1. 数据传送指令分为 8 位数传送和 16 位数传送两种。它们的作用都是在 CPU 的内部寄存器之间或是在内部寄存器与存储器之间传送数据。指令中都指明了被传送数据的源地址单元和目的地址单元。指令执行后,原地址单元的内容不变,而目的地址单元内容却与源地址单元的内容相同了。

例如:已知 A=64F, B=32H

(2000H)=20H

执行指令:

LD A,B; A $\leftarrow$ B(寄存器之间)

LD (200H),A;(200H)←A(寄存器与存贮器之间)

执行后:A=32H,B=32H

(200H)=32H

交换指令用于互换寄存器中的数据。

在数据传送和交换指令中,除指令

LD A,I 和 LD A,R

对标志位有影响外,其它指令对标志位均无影响。

2. 数据块传送指令能够把寄存器中的一个数据块搬到存贮器的其它任意位置。在处理大批数据时使用最多。数据块传送指令对 S、Z 和 C 标志无影响。

数据块搜索指令可以在存贮器的任一数据块中找出一个所需的8位信息。无论是找到了所需信息或是到达了数据块末尾,指令执行即自动结束。数据搜索指令对 C 标志无影响。

3. 算术和逻辑运算指令包括8位算术和逻辑运算指令、16位算术运算指令和通用算术指令三类。8位算术逻辑运算指令是对存放在累加器 A 中以及其他寄存器或是存贮器中的数据进行运算,结果均存于 A 中。8位算术逻辑运算指令均对各标志位有影响。

16位算术运算指令是在 CPU 的16位寄存器或两个8位寄存器对之间进行16位的加、减法运算。16位算术运算指令中的加和减指令对标志位无影响。

4. 循环和移位指令可以使任何寄存器或任何存贮单元实现带进位位 C 或不带进位位 C 的左移或右移。循环和移位指令对标志位有一定影响。

5. 位操作指令是 Z-80指令系统中所特有的指令,它可以对任一寄存器或任意存贮单元中的任意一位,进行置“1”、置“0”或“检测”。位操作指令在控制应用方面使用较多。位操作指令除检测指令外,对标志位无影响。

6. 转移、调用和返回指令可以使用户程序具备“判断”功能。可以使程序转移到任意位置上。这类指令对标志位无影响。

7. 输入/输出指令用来完成寄存器或存贮单元与外部输入/输出设备之间的数据传送。

8. CPU 基本控制指令用来实现不同的方式选择和控制。其中包括暂停、空操作、开放中断、关闭中断和建立不同的中断响应方式等。

指令格式

指令格式就是用一个字节(或若干字节)表示指令某些特征的形式。它与机器的种类、寻址方式以及指定的范围有关。一般说来,一条指令应由两部分构成:操作码和操作数。

Z-80的指令最少1个字节,最多4个字节。我们将其分别归纳如下:

1. 一字节指令

一字节指令有两种,一种是3位全是操作码,如图3-5所示;另一种则是8位中既包括操作码,又包括存放操作数的寄存器代码。如图3-6所示。

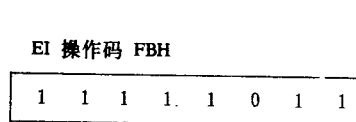


图3-5

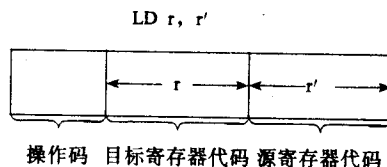


图3-6

2. 二字节指令

二字节指令有三种。一种是两个字节都是操作码,如图3-7所示;另一种是一字节操作码,

一字节立即数,如图3-8所示;还有一字节操作码,一字节相对偏移量,如图3-9所示。

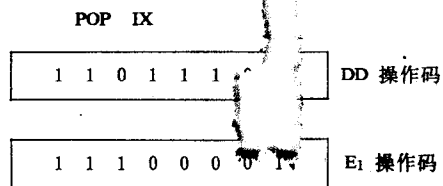


图3-7

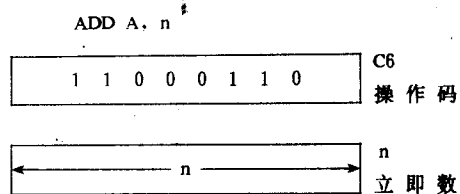


图3-8

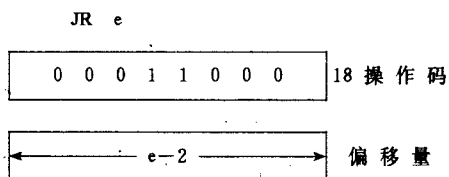


图3-9

3. 三字节指令

三字节指令分为以下三种:

一字节操作码,两字节立即数。如图3-10。

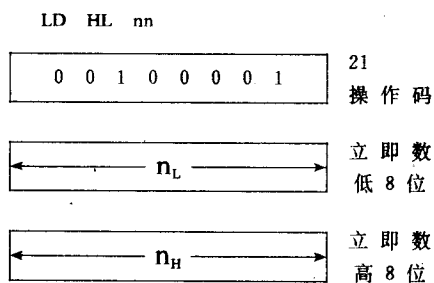


图3-10

一字节操作码,两字节地址码。如图3-11。

两字节操作码,一字节地址偏移量。如图3-12。

4. 四字节指令

四字节指令共有三种,分述如下:

两字节操作码,两字节立即数。如图3-13。

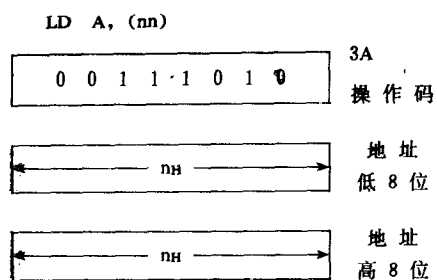


图3-11

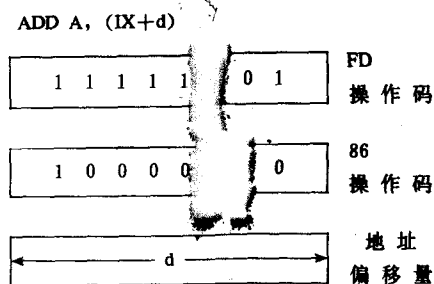


图3-12

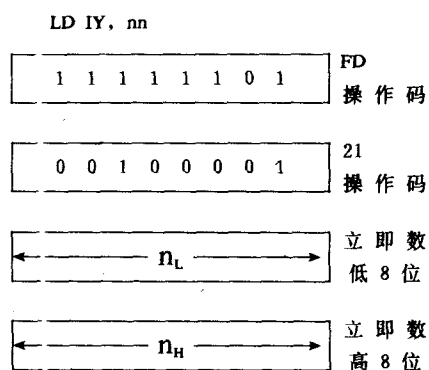


图3-13

两字节操作码,两字节地址码。如图3-14。

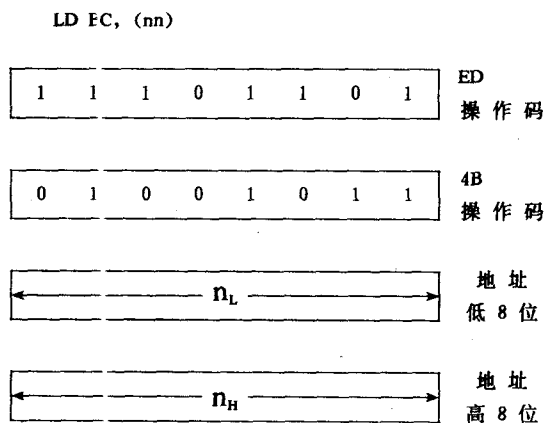


图3-14

两字节操作码,一字节地址偏移量,一字节补充操作码。如图3-15。