

微型计算机 系统设计

朱 家 维 著
齐 家 月

清华大学出版社

微型计算机系统设计
朱家维 齐家月 著

76
8

微型计算机系统设计

朱家维 齐家月 著

清华大学出版社

内 容 简 介

本书共分八章。首先比较了几种常用的8位微处理器,介绍了单板微型机的键盘、显示器、盒式磁带等接口设计方法、单片微型机的特点、常用的标准总线及模块系列;接着介绍了微型机系统的设计原理及方法,包括中央处理器板、存贮器系统及CRT、打印机、软磁盘接口的设计;讨论了微处理器及其系统的测试方法及设备;最后介绍了微型机系统的开发过程及开发系统的结构。

本书的系统性、先进性及实用性较强。它可以作为已具备微型机原理基础、从事微型机系统设计及应用的科技人员的参考书、一般科技人员学习微型机系统及应用的自学教材及大专院校研究生和本科生教材。

35126/13

微型计算机系统设计

朱家维 齐家月 著

☆

清华大学出版社出版

(北京·清华园)

北京通县向阳印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

☆

开本: 787×1092 1/16 印张: $19\frac{5}{8}$ 字数: 477字

1986年12月第1版 1986年12月第1次印刷

印数: 00,001—10,000

书号: 15235·236 定价: 3.25元

15235·236

前 言

自从 1971 年世界上第一个微处理器 Intel 4004 出现以来,在短短的十几年中,微处理器及微计算机经历了迅猛的发展过程。字长已从 4 位、8 位、16 位发展到 32 位。集成度从 Intel 4004 的一千多只晶体管发展到 HP-32 的 45 万。随着微处理器的飞速发展,适合不同领域、不同用途的各种不同规模的微型机系统不断推出。从单片机、单板机、模块系列、个人计算机到微型机系统,分别在微控制器、生产过程控制、数据采集与处理、商业管理与办公系统、信息处理及科学计算等方面得到了极其广泛的应用。

与此同时,由于微型机所具有的体积小、重量轻、耗电省、低价格、高可靠以及通用灵活等独特优点,亦广泛用于各种专用控制上。从卫星定向到炊具微波炉控制,从石油冶炼过程控制到血液测定,从交通自动管理到汽车火花控制等等,可以说微型机已渗透到各个领域。

微型计算机的飞速发展和广泛应用已给世界经济体系带来了极其深刻的影响,并将继续产生深远影响。我国于 1974 年开始微型计算机的研制工作,十年来,通过自力更生与引进先进技术相结合,也取得了一定的成绩,但比起先进国家来尚有很大差距。为了迅速发展我国的微型计算机事业,大力推广普及应用,则必须在科技人员中普及和提高有关微型计算机的知识。近几年对于一般微型计算机原理的基础知识已作了些普及,但对微型计算机系统及其应用则涉及尚少。为了使广大科技人员能够根据各自行业的需要,实际使用、设计和开发通用或专用的各种规模的微型计算机系统,进一步普及和提高有关微型计算机系统方面的知识是非常必要的。

鉴于 8 位机是当前应用的主流,因此本书专门阐述 8 位微型机系统及接口技术,同时为了适应对微型计算机熟悉程度不同的各种科技人员的需要,本书在内容安排上作了专门考虑。虽然本书是在微型机原理基础上的进一步提高,但为使缺乏这一基础的读者亦能看懂,同时亦为着重指出与系统有关的一些特性,第一章专门介绍“微处理器及其外围芯片”,其余各章则为直接牵涉到系统的内容。第二章介绍了使用非常广泛的单板机的硬件及软件设计。第三章介绍了在专用控制器中广为应用的单片微型机,以 8048、MC6801 及 Z8601 为例,详细讨论了典型单片机的结构。第四章介绍了微计算机系统中常用的几种并行标准总线及串行标准总线,包括 S-100 总线、MULTIBUS、IEEE-488 和 RS-232C。第五章介绍了模块系列,为那些对微处理器及其外围芯片不甚熟悉的人设计微计算机系统指出了一条简捷之路。第六章为微型计算机系统,详细介绍了功能较强的微型计算机系统及个人计算机的结构及设计,包括中央处理器板、存贮器系统、CRT、打印机及软磁盘接口设计。第七章介绍组成系统的微处理器、存贮器、插件板及微型计算机系统本身的测试,讨论了常用的测试方法、测试技术及测试设备。第八章介绍了微型机系统的开发过程、开发工具,并着重阐述了微型机开发系统的结构及其仿真器硬件。

本书系作者多年从事微处理器及微型计算机系统研制工作的总结,同时亦是对国内外有关微型计算机系统的大量最新资料、手册的综合、分析、总结与提炼。本书多次作为清华大学计算机系研究生“微计算机系统设计”课程的素材,并在教学过程中不断予以充实提高。本书力求不是仅给读者提供一般原理性的概念,而是结合大量系统实例,给出实用的硬件组成及

软件程序，同时指出各种方案的特点，以供读者在实际应用中参考。因此本书具有系统性、先进性和实用性，它可以作为从事微计算机系统设计及应用的科技人员的参考书，也可以作为微计算机系统及应用培训班的教材，还可以作为大专院校研究生和本科生的教材。

本书由朱家维、齐家月合著。其中第三章单片微型机、第四章 标准总线、第五章 模块系列、第六章 微型计算机系统、第八章 微型机系统的开发 由朱家维编著，第一章 微处理器及其外围芯片、第二章 单板微型机、第七章 微处理器及微型机系统测试 由齐家月编著，并由朱家维审核。

由于作者水平有限，书中难免存在缺点错误，恳切希望读者予以指正。

朱家维 齐家月

1984 年 5 月 25 日

于清华大学

目 录

前 言

第一章 微处理器及其外围芯片	1
第一节 常用的八位微处理器.....	1
一、编程模型.....	1
二、指令格式.....	2
三、寻址方式.....	2
四、指令系统.....	4
五、基本时序.....	7
六、外部引线及其功能.....	11
七、读写控制信号.....	14
八、对存储器存取时间的要求及等待态的插入.....	15
九、中断响应时间.....	16
十、总线响应时间.....	18
十一、电气特性.....	18
第二节 外围接口芯片.....	19
一、分类.....	19
二、可编程外围接口命令特征.....	22
三、接口与CPU及外界的应答工作方式.....	27
第二章 单板微型机	30
第一节 组成.....	30
一、硬件.....	30
二、I/O设备.....	30
三、扩展能力.....	31
四、软件.....	31
第二节 键盘接口.....	31
一、一般原理.....	31
二、键盘重键分析及键入程序设计.....	35
三、地址线扫描法.....	42
四、可编程序键盘接口.....	44
五、键盘编码器.....	48
六、键译码及命令转移程序.....	49
第三节 LED显示器接口.....	54
一、用7447七段译码/驱动器实现段译码.....	54
二、采用多路开关进行多位显示.....	55
三、硬件进行段译码、软件实现多位显示.....	55
四、软件实现段译码及多位显示.....	56

五、显示器“隐式更新”	59
六、可编程显示接口8279	60
七、点阵式显示器	63
第四节 音频盒式磁带接口	64
一、记录格式	64
二、频率调制法	66
三、周期调制法	78
四、双相脉冲制	80
五、幅度调制法	82
第五节 监控程序	86
一、组成	86
二、初始引导程序	87
三、单步命令	87
四、断点	89
五、寄存器保护区	92
六、寄存器的显示	93
第三章 单片微型机	96
第一节 概述	96
第二节 8048 单片微型机	97
一、结构	97
二、编程模型	97
三、寻址方式	101
四、指令系统	102
第三节 8048 的扩展	108
一、程序存贮器的扩展	108
二、数据存贮器的扩展	109
三、I/O扩展	110
第四节 MCS-48 系列	111
第五节 结构形式的变化——6801 及 Z8 单片机	112
一、MC6801 的结构	112
二、Z8601 的结构	116
第六节 中断结构	117
一、8048 的中断结构	118
二、MC6801 的中断结构	118
三、Z8601 的中断结构	119
第七节 RAM的掉电维持	119
一、Z8601 的掉电保护方式	119
二、MC6801 的掉电保护方式	120
三、8048 的掉电保护方式	120

第八节	8748 EPROM 编程、检验及擦除	121
一、	编程/检验	121
二、	8748 擦除特性	122
第九节	8048/8748 ROM/EPROM 的读出	122
第十节	单片微型机的开发	122
一、	采用开发系统	123
二、	专用简易开发工具——PROMPT 48	123
第四章	标准总线	125
第一节	概述	125
第二节	S-100 总线	129
第三节	MULTIBUS	129
一、	总线信号定义及其模块关系	129
二、	异步应答方式	132
三、	禁止操作	133
四、	总线仲裁	134
第四节	IEEE-488	138
一、	结构特点	138
二、	三线式信号交换	140
三、	总线数据的传送过程	140
四、	中断请求及其鉴别	143
五、	IEEE-488 与微处理器的接口电路	145
第五节	RS-232C 串行接口总线	146
一、	串行信息格式	146
二、	电气特性	146
三、	总线规定	147
第五章	模块系列	148
第一节	概述	148
一、	以元件级组成专用系统	148
二、	以模块级组成专用系统	148
第二节	产品系列	149
第三节	模块举例 iSBC 80/30	149
一、	结构	149
二、	跳线器选择	157
三、	多总线控制裁决逻辑	158
四、	双口控制器	161
第四节	iSBC 单板微型机的扩充结构	163
一、	iSBX 总线及 Multimodule(多模件扩展板)系列	163
二、	插座扩充结构	165
第五节	模块应用举例	166

第六章 微型计算机系统	169
第一节 概述	169
第二节 中央处理器板的结构	170
一、时钟电路	170
二、等待态插入电路	171
三、监控程序地址的设置与退出	174
四、标准总线信号的产生	179
五、复位电路	182
六、驱动器	183
第三节 存贮器系统	186
一、动态 RAM 的引脚结构及读写时序	186
二、时序的产生方法	187
三、存贮器容量的扩大	189
四、DMA 传送	194
五、动态存贮器的刷新	196
第四节 CRT 接口	198
一、独立的 CRT 终端	198
二、主机控制的 CRT	201
三、CRTC 控制的 CRT 显示器	205
四、CRT 产生特殊图形	218
第五节 打印机接口	221
一、打印机的基本工作原理	222
二、打印机的有关特性	223
三、主机与打印机接口电路	224
四、微处理器控制的打印机	228
五、汉字打印	230
第六节 软磁盘接口	233
一、概述	233
二、软盘控制器	235
三、CPU 与软磁盘控制器的数据交换方式	244
第七章 微处理器及微型机系统测试	251
第一节 存贮器测试	251
一、直流测试	251
二、交流测试	251
三、功能测试	251
第二节 微处理器测试	255
一、输入测试图案的产生	255
二、输出预期响应产生法	256
三、测试方法	256

四、测试技术·····	257
第三节 微处理器插件板模块测试·····	261
一、测试方法·····	261
二、测试方案·····	262
第四节 微型机系统测试·····	263
一、系统调试·····	263
二、系统诊断·····	264
第五节 微处理器及其系统的测试设备·····	264
一、微处理器及 LSI 自动测试系统 ·····	264
二、逻辑分析仪·····	265
三、微型机开发系统·····	266
第八章 微型机系统的开发 ·····	267
第一节 开发过程·····	267
第二节 开发工具·····	268
一、单板机同级开发系统·····	269
二、简易开发系统·····	269
三、微型机开发系统的分类·····	272
第三节 微型机开发系统的功能与结构·····	273
一、从功能要求看其结构特点·····	273
二、典型开发系统的配备及组成·····	274
三、MDS 与 ICE 的信息通路 ·····	276
第四节 仿真器硬件·····	278
一、跟踪板·····	278
二、处理器板·····	283
三、电缆电路板·····	288
四、ICE 工作过程 ·····	288
附录 ·····	292
附录 1 8080/8085 指令系统 ·····	292
附录 2 Z-80 指令系统 ·····	294
附录 3 6800 指令系统 ·····	297
附录 4 6502 指令系统 ·····	298

第一章 微处理器及其外围芯片

微型计算机系统即是以微处理器为中心,配以存储器、外围芯片及输入/输出设备所组成的系统。因此,在设计时,必须根据需求选择适当的微处理器及其外围芯片。自1971年第一台微处理器 Intel 4004 问世以来,微处理器得到了迅猛的发展。不仅从4位、8位、16位一直发展到了 Intel 的 iAPX432 32位微处理器,而且在各个级别、特别是8位和16位,也出现了众多的机种。这就为系统设计者提供了广阔的选择余地,同时也要求设计者对各个机种、至少对典型的几种主要微处理器的性能有所了解。本书主要讨论八位微型计算机系统,因而这里对几种常用的八位微处理器的特点及性能作一介绍。

第一节 常用的八位微处理器

目前国内外最常用的八位微处理器包括 8080A、8085A、Z-80、6800 及 6502,它们在下述各点上不尽相同。

一、编程模型

所谓编程模型,即程序员可与之打交道的寄存器及程序状态字。为此应明确各个寄存器的功能——哪些是通用寄存器,哪些是专用寄存器,各专用寄存器的用途。还应清楚其程序状态字的设置——共设置哪些标志、各标志所在位置以及各个标志的作用和置位条件。例如图 1.1 中 Z-80 编程模型中,包括两组通用寄存器 BC、DE、HL 及 B'C'、D'E'、H'L'; 累加器 A、标志寄存器 F 及其相应的辅助寄存器 A'F'; 堆栈指针 SP; 程序计数器 PC; 变址寄存器 IX、IY; 中断向量页面寄存器 I 和动态存储器刷新地址寄存器 R。其中通用寄存器既可作为八位寄存器使用,又可组成 16 位寄存器成对使用。图中 F 寄存器即程序状态字,共设有 7 个标志,其中 P/V 合占一位,所以共占六位。它们的置位条件分别为:

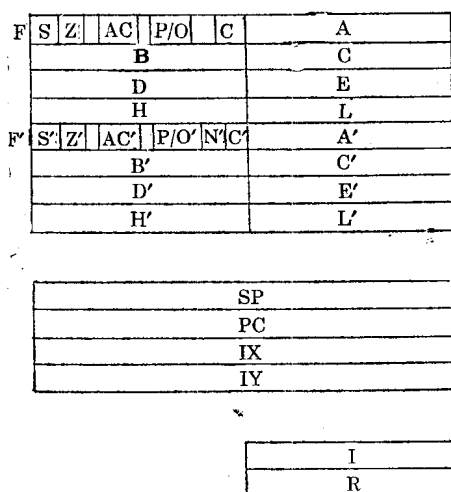


图 1.1 Z-80 编程模型

P/V: 当进行算术运算有溢出时,或者进行逻辑运算其结果中 1 的个数为偶数时置位。
N: 当进行减法运算时置位。设置该位的作用是用来控制十进制减法所需的修正。

变址寄存器 IX、IY; 中断向量页面寄存器 I 和动态存储器刷新地址寄存器 R。其中通用寄存器既可作为八位寄存器使用,又可组成 16 位寄存器成对使用。图中 F 寄存器即程序状态字,共设有 7 个标志,其中 P/V 合占一位,所以共占六位。它们的置位条件分别为:

S: 当运算结果为负、即 $IB_7=1$ 时置位。

Z: 当运算结果为零、即各位均为零时置位。

H: 当 $C_3=1$ 、即产生半进位时置位,在十进制运算时有用。

C: 在加法操作时最高位产生进位时置位, 在减法操作时产生借位时置位。

二、指令格式

微处理器基本指令的长度一般从一字节到四字节, 而第一字节总是包含操作码, 相继的字节(如果有的话)则含有寻址信息。图 1.2 中示出了几种 8 位微处理器的指令格式, 其中 Z-80 由于在 8080 的基础上增加了很多扩充指令, 所以有些新指令采用了双操作码格式——即第一、二字节均为操作码。



图 1.2 8 位微处理器指令格式

三、寻址方式

寻址的目的是为指令提供操作数的有效地址。在数据操作指令中, 操作数的有效地址是参加运算的数的地址。而在转移指令中, 则是要转移到的指令的地址。现将常用 8 位微处理器的寻址方式作一说明(参看图 1.3)。

1. 寄存器寻址

所有微处理器都具有某种形式的寄存器寻址方式, 否则寄存器就不能作任何事情。在寄存器寻址方式中, 操作数包含在处理器的一个寄存器中。由于微处理器的寄存器数目有限, 所以在操作码字节中用一个寄存器数即可对其寻址。

2. 直接寻址

直接寻址又可称作绝对寻址或扩展寻址, 它是一种表示存储器全部地址的最简单的方法, 可以方便地指定地址空间中的任何单元。由于直接寻址方式简单而又灵活, 因而所有微处理器均可具有这种寻址方式。由于一般 8 位微型机可直接寻址 64K 字节, 因而指令中要包含二个字节的有效地址。

3. 立即寻址

在立即寻址中,指令包含操作数,即一个立即操作数从紧接在指令操作码后面的存贮单元中取出。虽然立即寻址不是一种真正必须的寻址方式(因为常数亦可存贮在任意存贮单元,并象其它数据一样采用直接寻址方式进行存取)。但是,由于它取消了对常数的非顺序寻址,同时又缩短了程序的长度,加快了程序的执行,所以被微处理器广为采用。

4. 寄存器间接寻址

在寄存器间接寻址方式中,一个指定的寄存器或寄存器对中包含有操作数的16位有效地址。该寻址方式允许在程序执行时算出存贮器的地址,而不是在程序汇编时确定地址,这对在数组、堆栈、队列、连接表等数据结构中存取数据提供了方便。

5. 零页寻址

在这种寻址方式中,8080及Z80等微处理器在操作码中隐含地指定零页(0000H—00FFH)中某些规定的单元作为子程序入口地址,这种寻址方式主要应用于中断矢量的形成,如申请中断的外设提供的RSTn指令。而在6800、6502微处理器中,用操作码外的一字节的有效地址来寻址0000H—00FFH中的任一单元。

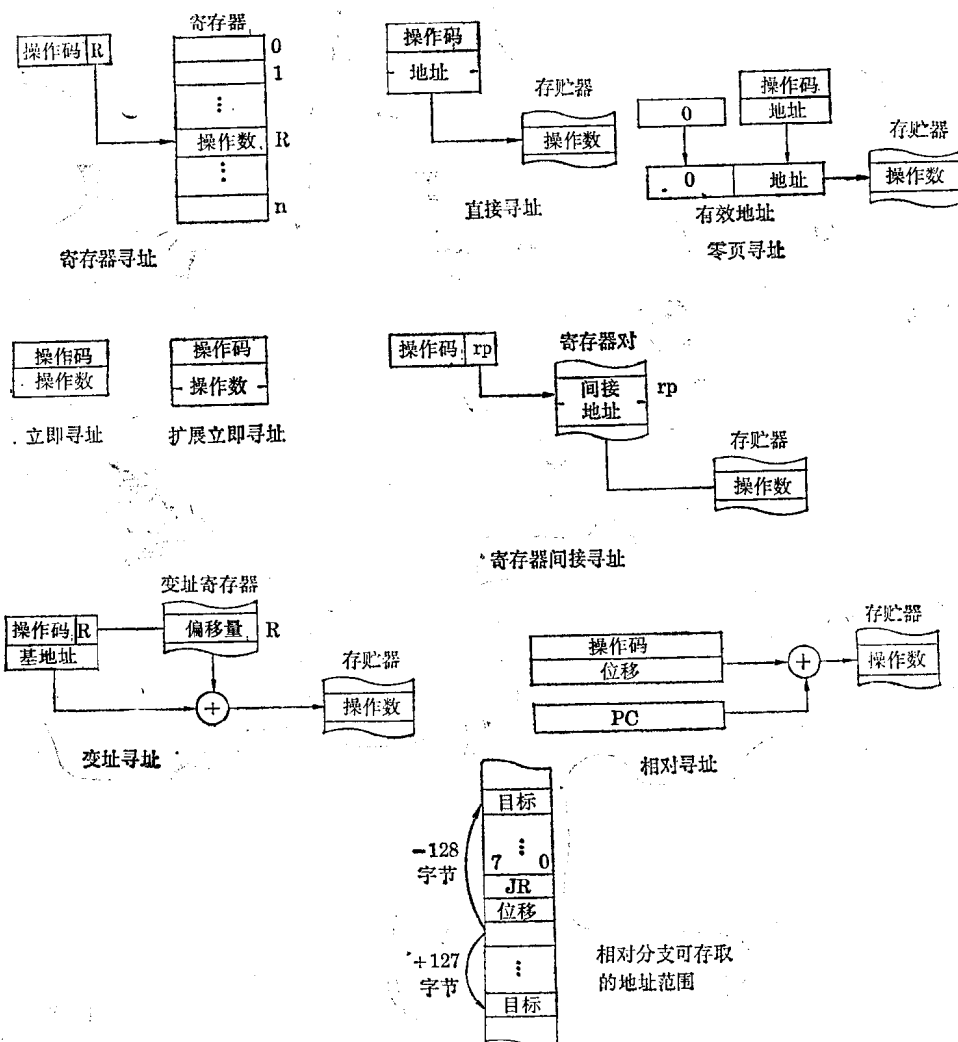


图 1.3 8 位微处理器的寻址方式

6. 变址寻址

变址寻址是把指令中给定的基地址与指定变址寄存器的偏移量相加而形成一个有效地址。该寻址方式对于存取数组和表格是非常适合的。如存取数组时,指令中的基地址对应于数组的基地址,而变址寄存器中的数值对应于数组元素的下标。在 Z80 和 6800 中设有这种强有力的寻址方式、而 8080 和 8085 则没有,这不能说不不是它们的一个缺陷。

7. 相对寻址

相对寻址是把指令中的一个固定的位移与程序计数器 PC 的现行值相加而得到有效地址。这种寻址方式常常用于分支(或转移)指令中。经过研究发现,大多数分支指令的目标地址在指令本身附近的 127 个字节之内。因此通常用一个字节表示位移即够。相对寻址有助于产生位置无关码,这是因为在相对寻址方式中,位移是指令地址与有效地址的差值。因此该指令与操作数可作为一个整体被移到存贮器的不同部分,而不必改变该指令中的位移。每当该指令执行时,有效地址作为现行 PC 值的函数而重新计算,所以用这种寻址方式的指令是与动态位置无关的。

8. 位寻址

所谓位寻址,就是指令能对任一寄存器或任一存贮单元的某一位进行检测、置位或复位。当微处理器用于控制系统时,这种寻址方式十分方便。

9. 隐含寻址

指令中隐含指定某一寄存器(通常为累加器 A)的内容为一个操作数,这种寻址方式称为隐含寻址或固有寻址。如 8080、Z80 的算术和逻辑操作指令,就是隐含着 A 为另一个操作数。

现将常用 8 位微处理器各自的寻址方式列于表 1.1 中。

表1.1 8080、8085、Z-80及6800的寻址方式

	8080	8085	Z-80	6800
寄存器	有	有	有	只有累加器A、B
直接	有	有	有(称作扩展)	有(称作扩展)
立即	有	有	有(还有立即扩展)	有
寄存器间接	有	有	有	无
零页	有	有	有	有(称作直接)
变址	无	无	有	有
相对	无	无	有	有
位	无	无	有	无
隐含	有	有	有	有(称作固有寻址)

注: 6502 寻址方式更多, 这里不作说明。

四、指令系统

虽然各种微处理器的结构决定了它们的指令不尽相同,但它们所执行的操作仍有很多类似之处。本节将对典型的 8 位微型机的各种指令进行讨论。

1. 条件位

条件位(或称条件码,或称标志)是一组状态位,每个状态位按各条指令的结果由处理器

自动置位。

条件分支指令测试一个或多个条件位的值,如果它们有规定值的话,则转移到一个新的地址。

条件位是微处理器状态的组成部分,在编程模型中,它们常常和其它状态信息一起送入一个寄存器中,该寄存器有的称为处理器状态字(PSW),有的称为标志(F),并设有专门的 PUSH PSW 和 POP PSW(或 Z-80 中的 PUSH AF 和 POP AF)指令可将其推入或弹出堆栈。

2. 数据传送指令

微处理器中最常用的指令是数据传送操作,它们把数据从寄存器(累加器)或存储器中的一个单元传送到另一个寄存器或存储器单元。在支持不同数据长度(如字节长和字长)的处理器中,对每种长度都有各自的数据传送指令。数据传送通常不影响条件位。

Load(LD) 和 Store(ST) 是两个最重要的数据传送指令。习惯上,LD 表示把数据从存储器传送到寄存器,而 ST 则表示从寄存器传送到存储器(如 8080、8085 及 6800 均如此)。但在 Z-80 中,LD 同时用于两个方向的传送,汇编程序由操作数的次序推断出要传送的方向。而 8080、8085 把 MOV 用于两个方向的传送。

在常用的 8 位微处理器中,通常都设有寄存器之间、立即数送寄存器(或寄存器对)、立即数送存储器单元(用寄存器间址或变址寻址)、寄存器与存储器之间的传送指令。并且设置了用于堆栈操作的 PUSH 和 POP 指令,在指令执行时堆栈指针自动减 1 或加 1(6800 的汇编符号为 PSH 和 PUL)。

在 8080、8085 及 Z-80 中还设有寄存器对的内容互换指令(EX)、Z-80 还有交换两组寄存器内容的指令(EXX)。在 Z-80 中,还有数据块传送指令(LDI、LDD、LDIR 和 LDDR),可以把整个数据块从存储器的一部分传送到另一部分。

3. 算术指令。

在任何微处理器中,其基本的算术操作是二进制加法,格式为

ADD dst, src

其中 dst 为累加器 A(在 6800 中亦可为 B),src 为立即数、寄存器或存储器单元。此外,还设有相应的带进位加法(ADC)指令。

大多数微处理器也设有减法(SUB)及带借位减法指令(SBC),机器内部采用补码运算。

加法和减法指令将根据进位及借位的产生对进位位 C 进行置 1 或置 0。

此外,微处理器还设有比较(CP 或 CPM)指令,用于累加器和存储器单元之间的比较。6800 还可在两个累加器之间进行比较。8080、8085 亦可将累加器与立即数进行比较。Z-80 的比较指令最为丰富,除了上述的以外,还可将累加器与任一寄存器进行比较。比较指令使两个操作数相减,根据结果设置条件位,但不保留差值(不破坏原累加器内容)。

再者,各种微处理器均设有便于十进制运算的二—十进制调整指令 DAA,以将 A 的内容调整为相应的 BCD 码。

4. 某些单操作数指令

由于加 1 或减 1 是程序流程中一种常用的操作,所以大多数处理器具有相应的指令 INC、DEC(或 8080、8085 中的 INR、DCR、INX、DCX)。但操作对象略有差别,如对任一

寄存器或寄存器对增量或减量, 或对存贮单元增量或减量。

值得注意的是, 在很多微处理器中, 此类指令不影响条件位中的 C 位, 因此在使用 C 条件转移指令时需加以注意。

另外常有的单操作数指令是累加器或存贮器单元的内容求反、求补以及条件位的置位、复位及求反指令(如 CMA、STC、CMC、CLC 等)。

5. 逻辑和移位指令

逻辑操作即分别处理一个数据字的每一位, 前面讨论过的求反操作, 即可以认为是一个逻辑操作, 因为它对一个字节的各位逐位求反, 实际上该操作应称为 NOT(非)。各个微处理器中设置的对两个操作数进行的逻辑操作包括与(AND)、或(OR)和异或(XOR)。其中一个操作数是累加器 A, 另一个操作数是寄存器立即数或存贮器单元。

逻辑操作通常用于抽取和组合数据字节中的字段。AND 指令可把不用的位置零, OR 指令可用来组合所要的字段。

各种微处理器的逻辑操作对条件位的影响不尽相同, 但大多数均按结果将 S 和 Z 置位。对其余条件位的影响则依机器而定, 如 Z-80 的 AND 指令将 H 置 1, 将 N、C 置 0, 而对 P/V 按结果的奇偶性进行设置。6800 则将 V 复位, 而对 H、I、C 无影响。

位寻址指令也可用于把许多布尔标志组合在一个数据字中, 如 Z-80 中的位测试指令(BITS)可对任一寄存器或存贮器单元中的某一位进行测试; 6800 中的 BITA、BITB 可分别对累加器 A 或 B 中的某一位进行测试。Z-80 还提供了更为有效的位置位(SET)和位复位(RES)指令, 可对寄存器或存贮器单元的任一位置 1 或置 0。

所有微处理器都有数量不等的循环和移位指令, 但比较起来 8080、8085 的最少, 它们只具有循环左移(带进位与不带进位)和对应的循环右移指令, 且循环操作仅能对累加器进行

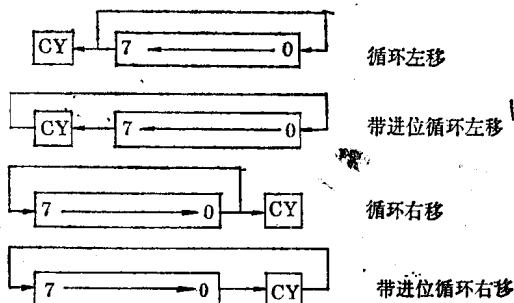


图 1.4 循环移位指令

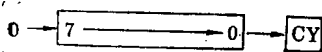


图 1.5 逻辑移位指令

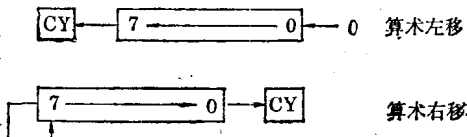


图 1.6 算术移位指令

操作。如图 1.4 所示 简单的循环左移和循环右移把 A 的每一位移动一个位置, 空出的位置用另一端排出的位填充。带进位的循环, 使用 C 条件位, 空出来的位置则用 C 装入, 而将从另一端排出的位装入 C。

Z-80 和 6800 则还设有逻辑移位和算术移位指令, 并且可对寄存器(6800 对累加器 A 或 B)和存贮器单元的内容进行操作。

逻辑移位是指用零装入空位的移位操作。排出的位通常保持在 C 中, 而 C 的原有值则丢失, 如图 1.5 所示。

算术移位把操作数作为带符号的 2 的补码数来处理。其操作按照如下方式进行: 左移一位等效于乘 2, 而右移一位等效于除以 2, 如图 1.6 所示。

一般移位和循环指令对 S 和 Z 条件位的影响很明显, 它们将依结果而被置位或复位。而对 P/V 位, 则在算术移位时根据是否溢

出进行置位或复位,在逻辑移位或循环时则随机器不同而有不同的设置方式。

6. 程序控制指令

计算机之所以能自动地执行复杂的程序流程,一主要原因就是设置了程序控制指令,因为它可根据条件进行操作或实现重复,因此这是一类十分重要的指令。其中最简单的是无条件转移(JMP),在8080、8085中JMP采用直接寻址,但在另外几种典型的8位微处理器中则均允许其采用相对寻址方式来确定有效地址。JMP的基本操作是将有效地址装入程序计数器(PC),从而把控制转到存在该地址的指令。

转移到子程序(CALL或BSR)指令提供一种调用子程序的手段,且在转移到子程序前在堆栈中保存其返回地址。与此相对应的是返回指令(RET),其作用是当子程序执行结束时,用以将控制返回到调用程序。

条件转移指令测试所规定的条件,只在条件成立时才转移。由于典型的8位微处理器均设有条件码,因而该类指令将测试一个或多个条件位的值。如8080、8085的JCN则可根据C(C=1)、NC(C=0)、Z(Z=1)、NZ(Z=0)、P(S=0)、M(S=1)、PD(P/V=0)、PE(P/V=1)分别进行测试和转移。

通常把利用程序计数器相对寻址来确定目标的转移叫作分支指令。典型的分支指令包含一个靠近现行指令的目标地址的短位移。这就缩短了分支指令的长度,并且和位置无关。由于这种指令很有用,因而大多数微处理器均提供有短位移(一字节)的条件分支指令。其位移值的范围为-128—+127的带符号的二进制补码。

几种常用8位微处理器的转移和分支指令列于表1.2中。

7. 输入/输出指令

微处理器输入/输出指令的有无,取决于微处理器输入/输出的结构。对于具有存贮器统一编址的I/O的机器来说,其与I/O接口的通讯与存取存贮器一样,因而不必另设指令(如6800)。而对于具有隔离I/O的机器,则设有输入(IN)和输出(OUT)指令(如8080、8085和Z-80),可以存取256个输入端口和输出端口。I/O端口一次可以存取一个字节。

IN指令从所指定的端口把一个字节传送到累加器;OUT指令把累加器内容传送到端口。可以在I/O指令中由一个8位数来指定端口地址,或由某寄存器的值来指定端口地址。

五、基本时序

微处理器完整地执行一条指令所需要的时间称作指令周期。每个指令周期由1—6个机器周期(M)组成,而每个机器周期又由3—6个状态组成。这种准同步的方法是目前多数微处理器所采用的基本时序,如8080、8085和Z-80。

一个指令周期所需要的机器周期数目一般地取决于该条指令的执行过程中与存贮器或I/O打交道的次数。由于取指周期对于任何指令都是需要的,因此所有指令都至少需要一个机器周期(即 M_1)。而后面是否还需要别的机器周期以及需要多少个,则依该指令还要对存贮器(或I/O)进行读/写的次数。这样将访问一次存贮器(或I/O)所需要的时间规定为机器周期的长度有利于时序电路的设计。

CPU对存贮器进行读/写,需要依次地送出地址,送读/写控制信号及送出数据(或读回数据)。为了协调这些操作,每个机器周期又由若干个状态(即时钟周期)组成。对于一般的存贮器(或I/O)读/写周期,从送出地址到读出数据需三个状态。而对于取指周期,由于从