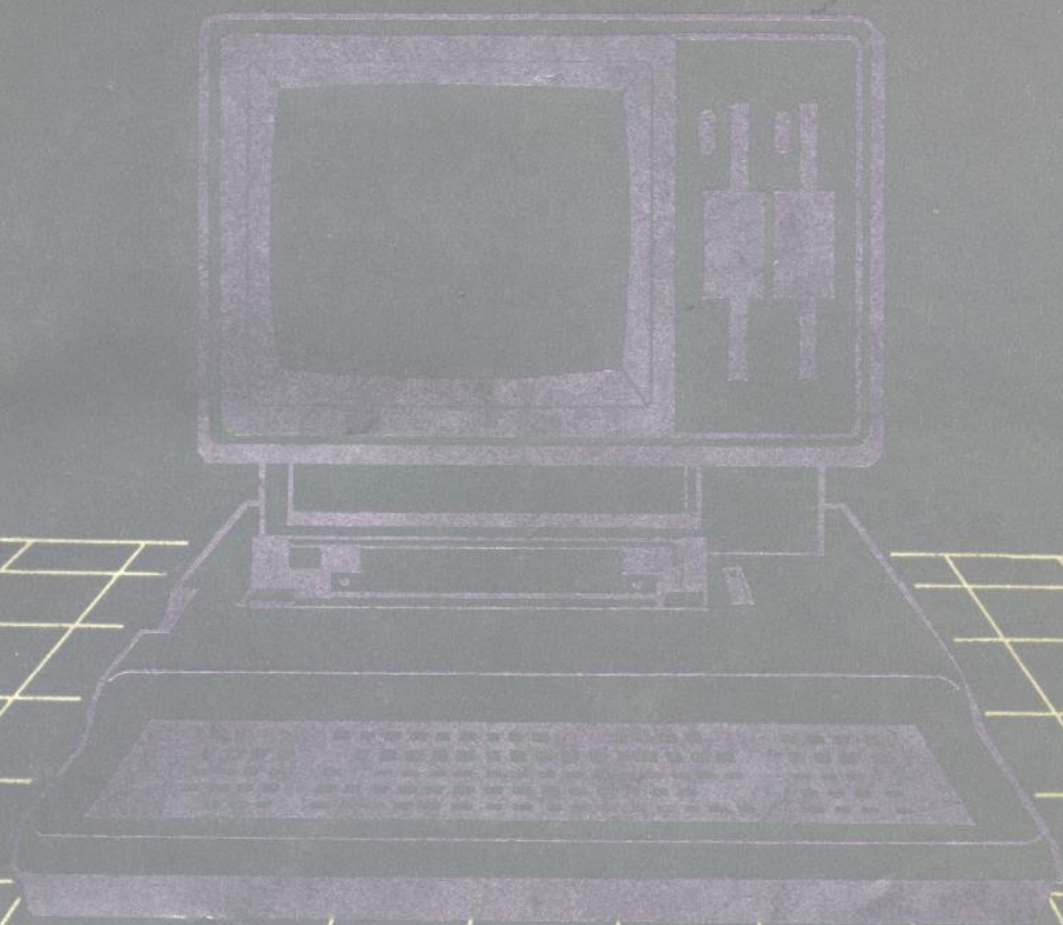




微型计算机 芯片组装与应用

李瑜祥 编著

上海科学技术出版社

6
1

微型计算机芯片组装与应用

李瑜祥 编著

上海科学技术出版社

微型计算机芯片组装与应用

李瑜祥 编著

上海科学技术出版社出版

(上海瑞金二路450号)

新华书店上海发行所发行 上海商务印刷厂印刷

开本787×1092 1/16 印张13.5 插页3 字数318,000

1986年5月第1版 1986年5月第1次印刷

印数 1-6,700

统一书号: 15119·2428 定价: 2.55 元

内 容 简 介

本书以应用最广的 Z-80、M6800 和 8080 三种机种为例, 详细介绍了微型计算机芯片组装方法与应用。

全书共分七章, 内容包括: 微型计算机基础知识; Z-80、M6800/6802、8080/8085 的工作原理与芯片组装方法; 常用接口芯片以及存储器与微型计算机的连接方法及应用; 键盘、数字显示器、盒式磁带机、行式打印机、CRT 显示器、软磁盘存储器以及 A/D 和 D/A 转换芯片等常用外围设备的接口原理及其应用程序; Z-80、M6800、8080 的指令系统以及微型计算机在数据采集和工业生产过程控制等方面的典型应用。

本书通俗易懂, 内容深入浅出, 可作为工程技术人员、工矿企业微机训练班学员和大专院校非计算机专业师生的参考书和自修课本。

JS305/L3

前 言

微型计算机的出现是七十年代科学技术发展的重大成果之一。在它问世之后短短的时间内,就获得了极其迅速的发展。八十年代初单片微型计算机和高性能的16位、32位微型计算机相继问世,更进一步促进了微型计算机应用的发展。目前微型计算机已广泛应用于数据处理、工业生产过程控制、智能化仪器仪表、节约能源、企业管理、生物医学、现代通信技术、交通运输、军事、商业银行、家用电器等各个领域。实践证明,微型计算机已成为解放人类繁重的脑力劳动,促进现代科学技术发展的有力工具。

近年来,我国在微型计算机生产和应用方面也有了迅速的发展。事实证明,推广微型计算机的应用是我国实现技术改造、促进四个现代化建设的一个经济而又有效的途径。

今天,组装一台单板微型机,或把它装入仪器仪表、电气设备和自控装置内,使之构成微型机化仪器设备,已经是容易做到的事。根据目前我国微型计算机生产及应用的现状和发展趋势,本书以国内外流行的、产量大、用户多的Z-80、M6800和8080三种8位微型计算机为例,着重讲述如何用微处理器、存贮器和各种通用的、专用的输入/输出接口芯片等器件构成微型计算机系统和各种新型的A/D和D/A转换器芯片等常用外围设备与微机的接口技术以及这三种微机在数据采集和处理、工业生产过程控制等方面的典型应用实例。

本书从实用角度出发,由浅入深地阐述了微型机支持器件的工作原理和使用方法,突出微型计算机原理的主要线索,摒弃一些细支末节。既注重硬件又注重软件,采用硬件和软件交融在一起的方法讲述微型计算机组装和应用的基础知识。通过许多实例分析将使读者具有举一反三、开拓新知识的能力,为他们从事微型计算机的组装、调试和使用打下坚实的基础。

由于编者水平有限,书中难免有不妥之处,敬请读者批评指正。

编 者

目 录

前言

第一章 微型计算机基础知识	1
1-1 微型计算机概述	1
1-2 数制和码制	2
1-3 指令、程序和语言	7
1-4 微处理器的结构和工作原理	8
1-5 存储器 and 堆栈	10
1-6 微型计算机的输入/输出和中断	11
1-7 总线	12
1-8 微型计算机的软件	12
1-9 微型计算机组装的注意事项	13
第二章 8080/8085 微型计算机的组装	14
2-1 8080 微型计算机系统的组成	14
2-2 8080 CPU	14
2-3 组合 CPU	19
2-4 地址译码器的连接	20
2-5 存储器的连接	23
2-6 8212 八位输入/输出接口	26
2-7 8255 A 可编程序并行接口	29
2-8 8251 可编程序串行接口	35
2-9 8214 优先权中断控制部件	40
2-10 8080 微型计算机组装实例	43
2-11 8085 微型计算机的组装	44
第三章 Z-80 微型计算机的组装	48
3-1 Z-80 微型计算机系统的组成	48
3-2 Z-80 CPU	48
3-3 存储器的连接	51
3-4 Z-80 并行 I/O 接口(PIO)	53
3-5 Z-80 计数器/定时器(CTC)	60
3-6 Z-80 串行 I/O 接口(SIO)	64
3-7 键盘接口	73
3-8 数码显示器接口	75
3-9 盒式磁带录音机接口	78
3-10 打印机接口	79
3-11 CRT 显示器接口	80
3-12 软磁盘存储器接口	84
3-13 Z-80 微型计算机组装实例	87

第四章 M 6800/6802 微型计算机的组装	90
4-1 M 6800 微型计算机系统的组成	90
4-2 M 6800/6802 微处理器(MPU)	90
4-3 MPU 的辅助电路	93
4-4 存储器的连接	95
4-5 MC 6821/6820 外围接口适配器(PIA)	97
4-6 MC 6850 异步通信接口适配器(ACIA)	102
4-7 M 6800/6802 微型计算机组装实例	108
第五章 8080 Z-80 和 M 6800 的指令系统	109
5-1 8080 微处理器的指令系统	109
5-2 Z-80 微处理器的指令系统	114
5-3 M 6800 微处理器的指令系统	139
第六章 模拟/数字转换芯片及其应用接口	147
6-1 模拟/数字转换与微型计算机控制系统	147
6-2 典型的 A/D 和 D/A 芯片	147
6-3 8 位 A/D 芯片与微型计算机接口	154
6-4 8 位以上的 A/D 芯片与微型计算机接口	158
6-5 多路 A/D 输入接口	162
6-6 D/A 芯片与微型计算机接口	167
6-7 带有多路开关和采样/保持器的模数转换接口	171
6-8 数据采集的定时问题	173
第七章 微型计算机应用实例	175
7-1 8080 微型计算机控制抽真空系统	175
7-2 8080 微型计算机进行小信号测量	176
7-3 M 6800 微型计算机进行信号处理	180
7-4 微型计算机控制液体混合器操作	182
7-5 Z-80 微型计算机检测生产过程的参数	184
7-6 Z-80 微型计算机进行 PID 调节	191
7-7 8080 分布式微型计算机控制系统	196
参考文献	202
附录 1 S-100 总线	202
附录 2 EIA RS-232 串行接口信号	204
附录 3 微型计算机常用的集成电路	205

第一章

微型计算机基础知识

1-1 微型计算机概述

自1971年世界上第一片微处理器 Intel 4004 和由它组装成的 MCS-4 微型计算机出现以来,微型计算机在短短几年中迅猛发展。至1980年为止,全世界累计生产了将近1200万台微机。微型计算机的发展可分为四代:

1. 第一代(1971~1973年) 以 Intel 4004 和 8008 微处理器为代表,采用 PMOS 工艺,字长4位和8位。

2. 第二代(1974~1978年) 以 Intel 8080/8085、MC 6800 和 Z80 为代表,采用 NMOS 工艺,8位字长。

3. 第三代(1978~1981年) 以 Intel 8086、MC 68000 和 Z8000 为代表,采用 HMOS 等先进的集成电路工艺,字长16位。

4. 第四代(1981年以后) 32位的高档微型计算机,其代表有 I 8800、IAPX 432(一个芯片上集成了100000个晶体管)等。

微机是在小型计算机的基础上发展起来的,它吸收了小型和大型计算机的优点和先进技术,具有体积小、可靠性高、价格低廉、功耗低、灵活性和通用性好等许多优点。

一、微处理器和微型计算机

电子数字计算机由运算器、控制器、存贮器、输入和输出设备这五个基本部分组成,如图1-1所示。其中控制器担负着控制和指挥整个机器的信息处理过程的任务,运算器具有对信息进行运算、整理、判断分析的功能,这两部分通常合称为中央处理单元或中央处理机(CPU)。

在一片或几片芯片上集成具有中央处理单元(CPU)功能的大规模集成电路叫做微处理器(μP)。

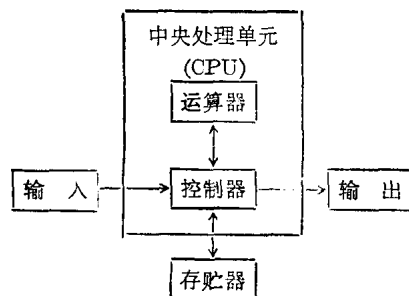


图 1-1 计算机结构框图

以微处理器为核心,加上大规模集成电路的存贮器芯片、输入/输出接口芯片和把各部件连接起来的三组总线等组成的计算机叫做微型计算机(下面简称微机(μO))。

二、微型计算机的分类

1. 按微处理器处理数据的位数分类

(1) 4位微型计算机 近年来生产的4位单片微机,多为专用而制造,价格便宜,产量大,广泛用于各种家用电器、仪器仪表、工业控制、办公设备等方面。

(2) 8位微型计算机 这是目前微机的主流,应用范围广泛,如工业控制设备、交通控制、仪器仪表、商业用设备等。

(3) 16 位微型计算机 这是高性能的微型计算机,可取代小型计算机,用于集中式数据处理。目前正向着高集成度(单片的微处理器)、高速度方向发展。

(4) 位片式微型计算机 位片机是采用双极性工艺,分别做成 2 位或 4 位的位片微处理器,能拼装成所需要的有一定字长的微型计算机,其特点是工作速度高。如: ECL 工艺的 MEOL 10800 位片式微处理器系列,时钟频率达 20 MHz,适用于高速的计算机、仪器及通信设备。

2. 按微型计算机组装形式分类

(1) 单板微型计算机 这种微机把微处理器、RAM、EPROM 存储器以及并行、串行的 I/O 接口组装在一块印刷板上,并配有键盘、显示装置、盒式磁带录音机等外部设备。在 EPROM 中存放监控程序。单板机使用简便,具有较强的扩充能力和接口能力。

(2) 多板微型计算机 这种微机把多块微型计算机印刷线路板和各种输入/输出设备控制器、电源、控制面板等部件组装在一个机架内,配有 CRT 显示器、打印机、软盘或硬盘等各种外部设备,并包含有较丰富的软件系统。它的功能比较强,能担负较复杂的数据处理和控制任务。

(3) 单片微处理器 这种单片机把微处理器、1~2 K 字节的 ROM、64~128 字节的 RAM 和 I/O 接口集成在一个芯片上,如 Intel 8048、8049、8748、8749、Z-8 等产品。它们适用于智能化的仪器仪表和控制装置。

3. 按半导体集成电路制造工艺分类

(1) PMOS 工艺 这种工艺具有价格低、集成度高等优点。但速度慢,且与 TTL 电路不兼容。

(2) NMOS 工艺 速度比 PMOS 高,但低于 TTL 的速度,可与 TTL 兼容。采用 NMOS 工艺制造的微处理器有 Intel 8080、8085、M6800、Z-80 等等。

(3) CMOS 工艺 速度和价格适中,能与 TTL 兼容,功耗低,抗干扰能力和环境适应性强。其产品有 IM 6100、ROA CDP 1802D 等。CMOS 微机适用于军事、航空、卫星等领域。

(4) HMOS 工艺 它是高性能、短沟道的 NMOS 工艺。集成度高,一个芯片上可集成几万个晶体管,工作速度快、门电路延迟时间 2~3 ns, μP 时钟频率达 8 MHz,是超大规模集成电路的一种重要工艺。其产品有 Intel 8086、M68000 等。

(5) 双极型工艺 它包括肖特基 TTL、EOL (射极耦合逻辑电路)、 I^2L (集成注入逻辑电路)和 I^3L (等平面集成注入逻辑电路)等。双极型 μP 尤其是 EOL 工作速度很高。 I^2L 工艺集成度高、工艺简单、功耗低、速度低于 TTL。产品有: SBP 9900(16 位)等。 I^3L 速度较快,产品有 Fairchild 9940(16 位),时钟频率达 10 MHz。

1-2 数制和码制

微型计算机所处理的一切信息(包括命令、数值、字符)统称为数据。微机使用的数制有二进制、十六进制、八进制等,下面分别加以介绍。

一、二进制

1. 二进制数

二进制数制只能用 0 和 1 两个符号表示任何一个数,它的基数是 2,按“逢二进一”原则

进位。

2. 二进制数化为十进制数

例如二进制数 1011.111 相当于十进制数

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 11.875$$

上式中二进制数各位的“权”分别为 8、4、2、1、0.5、0.25、0.125。

3. 十进制数化为二进制数

例如把十进制 12 化为二进制数的步骤如下：

商	余数
$\frac{12}{2} = 6 \dots\dots 0$	0
$\frac{6}{2} = 3 \dots\dots 0$	0
$\frac{3}{2} = 1 \dots\dots 1$	1
$\frac{1}{2} = 0 \dots\dots 1$	1

(二进制数结果)

4. 二进制加法和减法

二进制加法规则

$$0+0=0$$

$$0+1=1$$

$$1+1=0(\text{且进位为 } 1)$$

$$1+1+1=1(\text{且进位为 } 1)$$

二进制减法规则

$$0-0=0$$

$$1-0=1$$

$$1-1=0$$

$$0-1=1(\text{且借位为 } 1)$$

例 1-1

$$\begin{array}{r} 1011 \\ + 1101 \\ \hline 11000 \end{array}$$

$$\begin{array}{r} 1101 \\ - 0110 \\ \hline 0111 \end{array}$$

5. 二进制乘法和除法

二进制乘法规则

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

二进制除法规则与十进制除法类似,从被除数的最高位开始检查,定出需要超过除数的位数。找到这个位时商记 1,并把选定的被除数值减除数,然后把被除数的下一位移到余数上。如果余数减除数不够减则商为 0,把被除数的再下一位移到余数上……直至全部被除

数的位都下移完为止。

例 1-2

$$\begin{array}{r}
 1010 \\
 \times 0101 \\
 \hline
 1010 \\
 0000 \\
 + 1010 \\
 \hline
 110010
 \end{array}
 \qquad
 \begin{array}{r}
 1011(\text{商}) \\
 101 \overline{)110111} \\
 \underline{101} \\
 0011 \\
 \underline{0000} \\
 111 \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 0
 \end{array}$$

二、八进制

八进制数制只用 8 个数字 0、1、2、3、4、5、6、7 来表示一个数,其基数为 8,按“逢八进一”的原则进位。

例如八进制数 145.232 相当于十进制数:

$$1 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3} = 101.3007$$

二进制数往往很长,若按三位一组分成若干组即可转换成八进制数,例如:

二进制数	011	101	001	111	110
八进制数	3	5	1	7	6

三、十六进制

十六进制的基数是 16, 它用 0、1、2、3、4、5、6、7、8、9 十个数字加上 A、B、C、D、E、F 六个字母共 16 个符号来表示一个数。按“逢十六进一”的原则进位。

例如十六进制数 10AE 相当于十进制数:

$$1 \times 16^3 + 0 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 = 4270$$

将很长的二进制按四位一组分成若干组,便可转换得十六进制数。例如:

二进制数	0111	1100	1011	1001
十六进制数	7	C	B	9

微型计算机的数据用十六进制来表示时,一般规定在数的后面加 H 标记,十进制数一般不用标记。表 1-1 列出了十进制、二进制、八进制、十六进制数的对照表。

四、二十进制(BCD 码)

二十进制就是用四位二进制数来代替一位十进制数字编码,四位二进制数可以表示十六个不同状态,我们仅取其中十个状态来表示 0~9 十个十进制数,其余六个状态(1010 至 1111)不用。用二进制数代替一位十进制数编码的方法有多种,最常用的是 8421 码即二进制各位的“权”依次是 $2^3=8$ 、 $2^2=4$ 、 $2^1=2$ 、 $2^0=1$,见表 1-2。

五、数的定点制和浮点制表示形式

微型计算机中数的小数点不作为一个单独信息存放,确定小数点的位置一般有两种方法:

1. 定点制

在定点制中,小数点的位置固定不变,通常规定在数的最高位之前,符号位之后。

表 1-1 各种数制的对照表

十 进 制	二 进 制	八 进 制	十 六 进 制
0	0 0 0 0	0	0
1	0 0 0 1	1	1
2	0 0 1 0	2	2
3	0 0 1 1	3	3
4	0 1 0 0	4	4
5	0 1 0 1	5	5
6	0 1 1 0	6	6
7	0 1 1 1	7	7
8	1 0 0 0	10	8
9	1 0 0 1	11	9
10	1 0 1 0	12	A
11	1 0 1 1	13	B
12	1 1 0 0	14	C
13	1 1 0 1	15	D
14	1 1 1 0	16	E
15	1 1 1 1	17	F

表 1-2 二-十进制数码(8421 码)

十 进 制	二-十进制 (8421 码)	十 进 制	二-十进制 (8421 码)
0	0 0 0 0	10	0 0 0 1 0 0 0 0
1	0 0 0 1	11	0 0 0 1 0 0 0 1
2	0 0 1 0	12	0 0 0 1 0 0 1 0
3	0 0 1 1	13	0 0 0 1 0 0 1 1
4	0 1 0 0	14	0 0 0 1 0 1 0 0
5	0 1 0 1	15	0 0 0 1 0 1 0 1
6	0 1 1 0	16	0 0 0 1 0 1 1 0
7	0 1 1 1	17	0 0 0 1 0 1 1 1
8	1 0 0 0	18	0 0 0 1 1 0 0 0
9	1 0 0 1	19	0 0 0 1 1 0 0 1

例如八位二进制数用最高位作为符号位, 0 代表正, 1 代表负。

0 . 1 0 1 1 0 1 1

符号位 数 值 位

微机一般都采用定点制, 因此, 参加运算的数要先经过处理(例如乘上一个因子), 使其绝对值小于 1。运算后的结果要预先作出估计, 其值应小于 1, 否则微机将产生“溢出”, 迫使它停止现有运算, 转去处理溢出错误。

2. 浮点制

在浮点制中, 数的小数点位置是浮动的, 而不是固定的。每个浮点数由阶和尾数组成, 尾数是小于 1 的小数, 阶和阶符用来确定小数点的真正位置。采用二组二进制数来分别表示阶码和尾数。

六、原码、反码和补码

微型计算机在进行算术运算时, 对带符号的数规定有一定的表示规格(即码制)。

1. 原码

符号位为 0, 表示是正数; 符号位为 1, 表示是负数。正数的原码就是该数本身, 负数的原码: 数本身不变, 只是在符号位上写上“1”(最左边一位)。

例如: 0.1001101 是正数(+0.1001101)
 1.1001101 是负数(-0.1001101)

2. 反码

正数的反码就是原码。负数的反码就是: 符号位为“1”, 将其余数值位各位变反, 即将 0 换成 1, 将 1 换成 0。

例如: +0.1101001 的反码为 0.1101001
 -0.1101001 的反码为 1.0010110

0 的反码有两种表示方法:

+0 的反码为 0.0000000
-0 的反码为 1.1111111

3. 补码

一个数 X 以 N 为模的补码的求法是:

$$[X]_{\#} = N + X$$

微型机常用一个数 X 以 2 为模的补码, 其求法为:

当 X 为正数, 其补码等于 X 本身;

当 X 为负数, 其反码的末位加 1, 即得补码(微型机硬件很容易实现这一运算)。

例如: X = +0.1011011 $[X]_{\#} = 0.1011011$
 X = -0.1011011 $[X]_{\#} = 1.0100100 + 0.0000001 = 1.0100101$

反码和补码用来简化减法运算。

表 1-3 7 位 ASCII 码

高位 $b_6b_5b_4$ 低位 $b_3b_2b_1b_0$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	NUL	DLE	SP	0	@	P	,	p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	"	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	\$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
0 1 1 1	BEL	ETB	'	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	j	z
1 0 1 1	VT	ESC	+	;	K	[	k	{
1 1 0 0	FF	FS	,	<	L	\	l	
1 1 0 1	CR	GS	-	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	↑	n	~
1 1 1 1	SI	US	/	?	O	↓	o	DEL

例如,可用补码的加法来代替两个数的减法:

$$X = +0.1101001 \quad Y = -0.0100010$$

求 $X + Y$ 采用补码运算较为简便:

$$[X]_{\text{补}} = 0.1101001$$

$$[Y]_{\text{补}} = 1.1011110$$

$$[X]_{\text{补}} + [Y]_{\text{补}} = 0.1000111 \quad (\text{仍取八位})$$

七、字符代码

微型计算机除了处理数字信息外,还要处理大量非数字信息。例如 26 个英文字母和各种控制符号,这些字母和符号统称字符。

目前微型计算机大多采用 ASCII 码(美国信息交换标准代码)。ASCII 码用 6 位、7 位或 8 位二进制数码来表示英文字母、数字、控制符号和专用字符等。全 ASCII 码有 7 个数位(7 位可构成 128 种排列),可以表示一百多个不同的字母和符号(表 1-3)。7 位码上附加上 1 位奇偶校验位,便成为 8 位码。

1-3 指令、程序和语言

一、指令

微型计算机的指令就是指示微机执行特定的基本操作的命令,微机所能执行的基本操作的总和叫机器的指令系统。不同类型的微型机具有不同的指令系统。微机内部的电路只能直接接受二进制数码,因此指令实际上是二进制位的组合,它必须以二进制数据的形式存入微处理器中。

每条指令应该包括以下这些信息:

- (1) 进行什么操作 指令中指明所执行的特定操作的部分称为操作码。
- (2) 操作数据来源、操作结果送到什么地方去 指令中指明操作数据来源和目的地的部分称为地址码。
- (3) 下一条指令放在什么地方 本指令执行完毕后应指明下一条指令的地址是紧接着上一条指令的位置,还是转移到另一个特定的地址去。

微机一般以 8 位二进制数为基本单位——字节。微处理器一条指令包含的字节数可以是单字节、双字节、三字节或四字节。微型机的指令可以用助记符或二进制数码(多数用 16 进制数表示,这样较简短)来表示。助记符就是帮助记忆的符号,例如 ADD 表示加法指令。

二、程序

微型计算机的程序就是使它执行特定任务的一系列指令。

事实上,程序不仅包括指令,它还包含执行规定任务所需的数据和存储器地址。

微型计算机处理信息的基本单位称做字,每个字所包含的二进制位数叫字长。本书介绍的三种微型计算机都是 8 位字长。

三、语言

人们同微型计算机打交道,要使它按预先设想的计算步骤运算,或按规定的数学模型控制生产过程,首先必须用机器能识别的语言编制程序,把人们预定的方案表达出来,这就是

程序设计。微机的指令系统就是它能识别的基本语言。

应该记住,一个好的微机使用人员必须是一个很好的程序设计人员。

微型计算机程序设计所用的语言有以下三种:

1. 机器语言

它以二进制数码来表示,是微型计算机能直接认识的语言。为便于书写和阅读,常用十六进制数或八进制数来表示。用机器语言编写的程序,其中的指令、数据、地址看起来都相同,不易区分和修改。

2. 汇编语言

编写程序时,如果把指令、数据和地址都用助记符来表示,这种语言称为汇编语言,或称为符号语言。汇编语言比机器语言容易看懂,容易编写,指令中的操作码和操作数也容易区分。用汇编语言编写出的程序叫汇编语言程序。

汇编语言程序必须由预先放入微型计算机的特定程序把它翻译成机器语言,微机才能执行,这种特定程序叫汇编程序,这种方法叫“机器汇编”。汇编程序可以从工厂买到。

目前,在很多场合下常采用“人工汇编”方法,即借助指令表,把助记符号表示的程序写成机器语言代码。在程序不太长的情况下,人工汇编显得比较简单方便。

汇编语言程序与微型计算机的结构和指令系统有密切联系,对于不同类型的微型机,即使是同一个问题,编写成的汇编语言程序也不能相互通用。因此,用汇编语言进行程序设计时,设计人员必须对微型计算机的硬件结构和指令系统有较详细的了解。

3. 高级语言

这是一种“面向过程”而不是“面向计算机”的语言,它用与我们通常解题步骤相接近的方式来描述任务、编写程序。这种程序对于不同类型的计算机基本上都可通用。

高级语言有: BASIC、FORTRAN、ALGOL、COBOL、PASCAL、PL/I、APL 等许多种类。它们有各自的特点和使用范围。

每种高级语言都有一套较复杂的“语法”规则,高级语言程序要由“编译程序”把它译成机器语言,微型机才能执行。高级语言占用的内存单元很多,需要较多的软件和硬件支援,且对于微型机的接口技术并不有效。因此,微型计算机宜采用汇编语言来执行控制任务。

针对微机内存容量较小的特点,许多厂商致力于缩小高级语言编译程序所占用的存储空间,采用了Tiny BASIC(是从 BASIC 语言系统中抽取的子集)和 PL/M (从 PL/I 语言中抽取的适合硬件结构的部分)等语言。

1-4 微处理器的结构和工作原理

一、微处理器的内部结构

不同类型的微处理器的内部结构虽然各不相同,但它们都包括以下各部分:

- (1) 算术逻辑运算部件;
- (2) 指令寄存器和控制器;
- (3) 寄存器组。

图 1-2 是标准微处理器的内部结构,它主要由以下几部分组成:

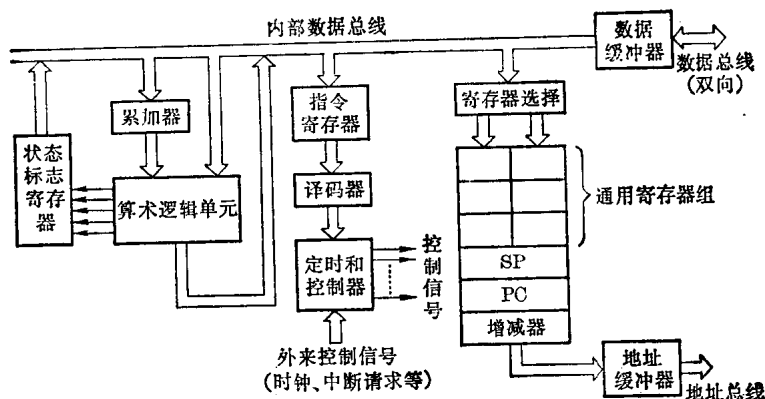


图 1-2 微处理器内部结构

1. 程序计数器(PC)

它的内容(16 位)指出了现行指令在存储器中的地址。

2. 指令寄存器(IR)和译码控制器

微处理器指令的第一字节(8 位)都包含着操作码,它被送入指令寄存器,再由指令译码器进行译码,然后经定时和控制部分结合时钟等外部送来的信息产生控制信号,用来指挥微处理器内部和外部各部分的协调工作。

3. 通用寄存器组

它用来暂时存放操作数、中间结果或地址。通用寄存器数目多就能减少 CPU 访问存储器所花的时间。微处理器一般有八个通用寄存器。

4. 累加器(A 或 AC)

它参加算术和逻辑运算操作,并用来存放结果,是应用最频繁的寄存器。

5. 标志寄存器

当数据进行算术或逻辑运算时,可能会发生进位、溢出、全零、符号和奇偶性等状态的变化,标志寄存器用来存放这些状态的标志。

6. 算术逻辑运算单元(ALU)

它是单元字节的运算电路,一般具有加减运算,逻辑运算、移位、求补、二-十进制调整等功能。

7. 堆栈指示器(SP)

堆栈是存储器 RAM 中指定的一部分(其功能后面将详述)。堆栈指示器用来指示堆栈顶部的地址。

二、微处理器的工作原理

微处理器是微型计算机的核心,它既处理数据又控制着整个微机系统。它通过数据总线(双向)、地址总线和控制总线与存储器、输入/输出接口联系。

要使微型机完成一项计算或控制任务,操作人员必须首先通过键盘、磁带存储器等输入设备把程序和操作数据送入存储器。微处理器从规定的起始地址开始执行程序。它执行一条指令的过程是这样的:先从存储器把指令取出,然后对指令进行译码,也即对指令中的操作码和地址码进行译码,指明该指令执行什么类型的操作和操作数据的地址。接着,根据这一地址从微处理器内部的寄存器、存储器、I/O 端口取出操作数,并按指定的操作类型,对

操作数进行操作,实现处理或控制功能,例如进行各种算术和逻辑运算,对存储器进行访问与 I/O 接口进行通信等。

微处理器在执行指令时,各种动作不但按顺序,而且按精确的定时进行。为了连续地执行指令序列,它需要有时钟脉冲。这一时钟脉冲有的从 CPU 外部供给,也有的由 CPU 内部供给。微处理器的工作是周期性的,时钟脉冲的周期(称时钟周期)是处理操作的最基本单位。取出和执行一条指令的时间称为指令周期,每个指令周期包括几个机器周期(8080 有 1~5 个机器周期),而每个机器周期又包括若干个时钟周期(8080 有 3~5 个时钟周期),每个机器周期执行一种基本的操作,例如取指令、存储器读、存储器写等等。

在比较微处理器性能时,一般使用最小指令执行时间这一指标,它是指在最高时钟频率下工作时,该微处理器所有指令中最短的执行时间。

1-5 存贮器和堆栈

一、存贮器概述

存储器是微型计算机用来存储程序指令和数据的部件。

存储器有半导体存储器和磁芯存储器两种。目前绝大多数微型计算机都采用半导体存储器,因为它具有集成度高、价格低廉、体积小、功耗低,与微处理器配用方便等许多优点。

存储器一般以它包含的二进制信息位数目或者字节数目来表示它的容量。8 位微机以一个字节(8 位)表示一个存储单元,例如一片存储器的容量为 8192 位,也可以说它有 1024 个存储单元。

每个存储单元在存储器中的位置用一个“地址”(十六位二进制数)来表示。8 位微型机的地址总线通常包含 16 条(位)地址线,它们用来指定存储单元的地址,最多可指定 $2^{16} = 65536$ 个不同的存储单元。

存储器的工作速度是一个重要指标,常用存取时间来表示它,一般为 2 ns 到 2 ms。

半导体存储器一般可分为 ROM (只读存储器)和 RAM (随机存取存储器)两类。下面分别加以介绍。

1. ROM (只读存储器)

它是一种一旦编入程序后就只能进行读出操作的非易失性存储器,断电以后信息仍能保持,常用于存放固定不变的程序和数据。

ROM 按其结构和写入程序的方式可分为以下三种:

- (1) 固定掩模式 ROM。
- (2) PROM (可编程序 ROM)。
- (3) EPROM (可擦写可编程序 ROM)。

2. RAM (随机存取存储器)

它是一种既能从中读出信息,又能按指令要求写入信息的存储器。微型计算机使用的 RAM 大多数是易失性的,断电后它就会丢失所存储的信息。

RAM 按结构和性能可分为静态和动态两种类型。动态 RAM 具有集成度高(容量大)、价格便宜、存取速度较快等优点,但需附加一个信息刷新电路。