

目 录

第 1 章 单片机基础知识.....	1
1.1 概述	1
1.1.1 微型计算机及其发展状况.....	1
1.1.2 单片机及其发展状况.....	2
1.1.3 80C51 系列单片机简介	3
1.1.4 单片机的特点及应用领域.....	4
1.2 单片机系统的组成.....	4
1.2.1 单片机系统的硬件组成.....	5
1.2.2 单片机系统软件组成.....	9
1.3 单片机中数的表示及编码.....	10
1.3.1 正数和负数在单片机中的表示方法.....	10
1.3.2 常用编码.....	12
1.4 习题	15
第 2 章 MCS-51 单片机硬件与系统扩展.....	16
2.1 MCS-51 外部引脚.....	16
2.2 P0、P1、P2、P3 口的内部结构及使用.....	18
2.2.1 P0 口的内部结构及使用特点.....	18
2.2.2 P1 口的内部结构及使用特点.....	19
2.2.3 P2 口的内部结构及使用特点.....	20
2.2.4 P3 口的内部结构及使用特点.....	20
2.2.5 P0、P1、P2、P3 作 I/O 使用实例.....	21
2.3 单片机内部硬件资源.....	24
2.3.1 片内程序存储器.....	25
2.3.2 片内部数据存储器.....	26
2.4 时钟与复位电路.....	30
2.4.1 时钟与时序.....	30
2.4.2 复位.....	32
2.5 单片机系统扩展.....	34
2.5.1 基本部分的组成.....	34

2.5.2 系统的扩展.....	35
2.6 习题	41
第3章 MCS-51 单片机指令系统.....	43
3.1 指令系统的基本概念.....	43
3.1.1 指令的基本格式.....	43
3.1.2 MCS-51 指令系统中的常用符号说明	44
3.1.3 MCS-51 指令系统的分类	45
3.1.4 寻址方式.....	45
3.2 MCS-51 指令系统.....	49
3.2.1 数据传送类指令.....	49
3.2.2 算术运算类指令.....	56
3.2.3 逻辑运算及移位指令.....	60
3.2.4 位操作类指令.....	65
3.2.5 控制转移类指令.....	66
3.3 习题	74
第4章 汇编语言与仿真系统.....	81
4.1 伪指令	81
4.2 汇编语言程序设计.....	84
4.2.1 简单程序设计.....	84
4.2.2 分支程序设计.....	87
4.2.3 循环程序设计.....	91
4.2.4 查表程序设计.....	93
4.2.5 散转程序设计.....	93
4.2.6 子程序程序设计.....	95
4.3 汇编语言程序的仿真与调试.....	96
4.4 汇编语言程序的结构.....	99
4.5 习题	102
第5章 中断系统及定时/计数器.....	105
5.1 80C51 中断系统.....	105
5.1.1 中断概述.....	105
5.1.2 中断源和中断控制器.....	107
5.1.3 中断处理流程.....	111
5.1.4 中断优先控制和中断嵌套.....	115
5.1.5 中断系统的应用.....	116

5.2	80C51 定时/计数器.....	120
5.2.1	定时方法概述.....	120
5.2.2	80C51 定时/计数器的结构和工作原理.....	121
5.2.3	定时/计数器的控制寄存器.....	122
5.2.4	定时/计数器工作方式.....	124
5.2.5	定时/计数器的应用.....	126
5.3	习题.....	134
第6章	串行接口与通信基础.....	137
6.1	串行通信基本知识.....	137
6.1.1	数据通信的基本概念.....	137
6.1.2	串行通信的方向.....	137
6.1.3	串行通信的同步方式.....	138
6.2	串行通信硬件电路.....	139
6.3	串行通信主要寄存器与设置.....	141
6.4	串行通信的应用.....	144
6.5	习题.....	152
第7章	常用接口及应用实例.....	154
7.1	显示器接口技术.....	154
7.1.1	数码管的种类.....	154
7.1.2	字型(段)编码.....	155
7.1.3	数码管接口电路.....	156
7.1.4	动态扫描程序设计.....	160
7.2	键盘接口技术.....	162
7.2.1	按键的机械特性.....	162
7.2.2	按键电路与程序设计.....	163
7.3	A/D 接口技术.....	167
7.3.1	A/D 转换器概述.....	167
7.3.2	ADC0809 内部逻辑结构.....	168
7.3.3	ADC0809 应用实例.....	169
7.3.4	串行接口 A/D 的应用.....	171
7.4	D/A 接口技术.....	173
7.4.1	D/A 转换器概述.....	173
7.4.2	DAC0832 内部逻辑结构.....	174
7.4.3	DAC0832 的工作方式.....	175
7.4.4	串行接口 D/A 的应用.....	177

7.5	串行 RAM 接口技术.....	180
7.6	可编程 I/O 接口技术.....	186
7.6.1	8255A 可编程并行 I/O 出接口.....	187
7.6.2	8155 可编程多功能接口.....	191
7.7	习题.....	195
第 8 章	综合设计实例.....	197
8.1	设计概述.....	197
8.2	设计举例.....	199
8.3	单片机应用系统的一般调试方法.....	232
8.3.1	硬件调试.....	232
8.3.2	软件调试.....	233
8.3.3	系统联调.....	234
8.3.4	现场调试.....	234
8.4	习题.....	234
参考文献		235

第 1 章 单片机基础知识

1.1 概 述

自从 1946 年第一台电子计算机发明以来,随着大规模集成技术的发展,计算机得到了飞速的发展和广泛的应用,特别是 1971 年微型计算机的问世,使计算机能够真正进入社会的各个方面,使我们能够大跨步地走入电脑时代。

1.1.1 微型计算机及其发展状况

1. 微型计算机

计算机的发展经历了从电子管、晶体管、集成电路到大规模集成电路共四个阶段,也就是通常所说的第一代计算机、第二代计算机、第三代计算机和第四代计算机。微型计算机是大规模集成电路技术的直接产物,因此属于第四代计算机,1971 年 11 月,Intel 公司成功地将控制功能部件和运算部件集成在一起,制成了世界上第一块中央处理芯片——微处理器 (Microprocessor) Intel 4004,微型计算机由此踏上了高速发展之路。

微处理器 (通常称为 CPU) 是一种大规模集成电路器件,它包括计算机的控制部件和运算部件,具有控制和运算功能。为了与巨型、大型、中型、小型计算机的 CPU 相区别,微型计算机的 CPU 也可以称为微处理器 MPU (Micro Processing Unit)。

在微处理器的基础上,加上用于存储程序和数据的存储器,以及与输入/输出设备相连接的输入/输出接口电路,就组成了微型计算机 MC (Microcomputer),如图 1-1 所示。

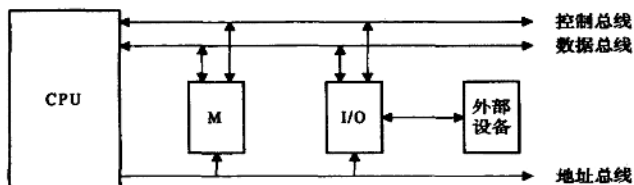


图 1-1 微型计算机的组成

而所谓的单片机,就是将中央处理器 CPU、随机存取存储器 RAM(Random Access Memory)、只读存储器 ROM(Read Only Memory)、定时计数器以及输入/输出(Input/Output)接口电路等主要计算机部件,集成在一块电路芯片上的微型计算机。

2. 微型计算机的发展状况

在其发展过程中,按照微处理器的功能,可以分为两个发展阶段。

(1) 第一阶段,初始阶段。以 Intel 推出 4004 开始,逐步推出 Intel 8008、8080、8085、Motorola 公司的 MC 6800 和 Zilog 公司的 Z 80。这些微处理器的功能和速度都逐步提高,已经具有实用价值。

(2) 第二阶段,高速发展阶段。20 世纪 70 年代中期,随着大规模集成电路技术的发展,微处理器的集成度的提高,促成微型计算机向两个方向发展:

① 向高速、高性能的高档机方向发展,也就是通常所说的 PC 机。主要以 Intel 公司的 8086、80286、386、486、586、奔腾 II~IV 为代表,着重突出高速数据的计算处理能力,数据宽度不断更新,从 8 位、16 位发展到 32 位、64 位,在图像处理、数据处理、模拟仿真、多媒体等领域中应用十分广泛。

② 向稳定可靠、体积小、价廉的单片机方向发展。主要以 Intel 公司的 MCS-48、MCS-51、MCS-96……为代表,重点突出控制能力,实行嵌入式应用,因此单片机也称为嵌入式微控制器。在智能仪表、智能家电、工业测控系统等领域得到广泛应用。

嵌入式系统和 PC 机系统形成了微型计算机的两大分支。嵌入式系统主要满足对象的测控能力,兼顾数据处理能力;而 PC 机系统主要满足海量高速数据处理,兼顾控制能力。通过网络系统,两者之间通过串行通信,各取所长,优势互补,形成了具有高速运算、实时控制的网络控制系统。

1.1.2 单片机及其发展状况

1. 单片机名称的来源

根据单片机的物理形态和内容,早期的单片机称为 Single-Chip Microcomputer。从仙童公司的第一款单片机 F-8 开始,甚至到 MCS-51 时期,这些单片机确实准确地体现了 Single-Chip Microcomputer 的形态和内容。但是,随着单片机发展到 MCS-96、新一代 80C51、M68HC05 等系列单片机时,这些单片机面向对象,突出控制功能,在片内集成了许多外围电路及外设接口,如 A/D、PWM、WDT 等,突破了传统意义的单片机结构,发展成 Microcontroller 的体系结构,因此,目前国外已经逐步统一称单片机为 MCU(Micro Controller Unit)。由于它实行嵌入式应用,因此也称为嵌入式微控制器。

2. 单片机发展状况

从最初的单片机发展到如今的新一代单片机,大致经历了三个年代。如以 Intel 8 位单片机为例,这三个年代大致可分为:

(1) 第一代。以 1976 年推出的 MCS-48 系列为代表,主要特征是将 CPU 和计算机外围电路集成到了一个芯片上,属于低档型产品。这个阶段也是单片机的起步和探索阶段。

(2) 第二代。以 MCS-51 的 8051 和 8052 为代表,主要特征是为单片机配置了完善的外部并行总线 (AB/DB/CB) 和具有多机识别功能的串行通讯接口 (UART)。这个阶段是单片机的完善阶段。

(3) 第三代。以 80C51 为代表,包括了 Intel 公司发展的 MCS-51 系列的新一代产品,如 8XC152、80C51FA/FB 等,还包括 Philips、Siemens、ADM、ATMEL、华邦等公司以 80C51 为核心推出的各具特色的单片机,主要特征是向外部接口电路扩展,突出 Micro-controller 的内容,使用 A/D、PWM、WDT、PCA (可编程计数器阵列) 等功能部件。这一阶段属于单片机的巩固和发展阶段。

新一代单片机为外部提供了更加完善的总线结构,为系统的发展和配置打下了良好的基础。同时也向高速、大寻址范围、强运算能力的专用型单片机发展。

目前,单片机在国内得到了广泛的应用,以 80C51 系列的单片机为主,其中 ATMEL 公司的配置 Flash ROM 的 AT89 系列最为流行。同时,在最近几年,Microchip 公司的具有 RISC (精简指令集计算机) 体系结构的 PIC 系列单片机发展也很迅速,大有与 80C51 系列单片机一决雌雄的态势。

1.1.3 80C51 系列单片机简介

尽管各类单片机很多,但无论从世界范围还是从国内范围来看,使用最为广泛的还是 MCS-51 型单片机,因此本书也将以 MCS-51 型系列为主,来介绍单片机的原理及应用。

MCS-51 型系列单片机共有十几种芯片,表 1-1 列举了比较典型的几种芯片的型号及主要技术性能指标。

表 1-1 MCS-51 系列单片机分类表

子 系 列	片内 ROM 形式			ROM 容量		RAM 容量		计 数 器	并 行 口	串 行 口	中 断 源	制 造 工 艺
	无 ROM	掩膜 ROM	EPROM	片内	片外	片内	片外					
51 子 系列	8031	8051	8751	4KB	60KB	128B	64KB	2×16	4×8	1	5	HMOS
	80C31	80C51	87C51	4KB	60KB	128B	64KB					CHMOS
52 子 系列	8032	8052	8752	8KB	56KB	256B	64KB	3×16	4×8	1	7	HMOS
	83C252	80C252	87C252	8KB	56KB	256B	64KB					CHMOS

80C51 型单片机属于 Intel 公司的 MCS-51 系列单片机, MCS-51 系列单片机采用两种半导体工艺生产。一种是 HMOS 工艺, 即高密度短沟道 MOS 工艺。另外一种 CHMOS 工艺, 即互补金属氧化物的 HMOS 工艺。表 1-1 的芯片型号中带有“C”的都为 CHMOS 工艺, 其余的为一般的 HMOS 工艺。

CHMOS 是 CMOS 和 HMOS 的结合, 除具有 HMOS 高速度、高密度的特点外, 还具有 CMOS 低功耗的优点。比如 8051 型的功耗为 630mW, 而 80C51 的功耗只有 120mW。

目前, Intel 公司将 80C51 型单片机的内核使用权以专利互换或者出售的形式转让给其他的著名 IC 制造商, 如 Philips、ATMEL、AMD、Dallas、Siemens、LG、华邦等。这些公司在 80C51 内核基础上, 扩展了针对不同需求的外围电路, 如 A/D、PWM、WDT, 引入使用方便并且价格便宜的 Flash ROM 等, 使 80C51 的功能更加齐全、针对性更强, 巩固并发展了 Intel 公司单片机的地位, 成为当今世界 8 位单片机的主流。

1.1.4 单片机的特点及应用领域

单片机的主要特点如下:

- (1) 很高的性价比。目前许多单片机的价格只要几元人民币。
- (2) 集成度高, 体积小, 可靠性好。内部采用总线结构, 减少各芯片间的连线。
- (3) 控制能力强。单片机的指令丰富, 能满足各种工业控制的要求。
- (4) 低功耗、低电压, 一般在 3V~6V 范围内工作, 低电压供电的单机电源下限可达 1V~2V, 1V 以下供电的单片机也已诞生, 便于生产便携式设备。
- (5) 易扩展。可根据需要进行并行或者串行扩展, 形成网络控制系统。

由于单片机的特点比较突出, 因此, 在各个应用领域都可以见到它的身影, 主要的应用领域有:

- (1) 工业自动化控制。这是最早采用单片机控制的领域之一, 如各种测控系统、PLC 等。
- (2) 智能化家用电器。用单片机控制来替代传统的电子线路控制是当前家用电器的发展趋势, 如空调、洗衣机、电视机等。
- (3) 智能化仪表。采用单片机的智能化仪表, 加强了数据处理能力和网络数据传送能力, 提高了仪表的档次, 如各种探测仪表、自动抄表系统等。
- (4) 办公自动化设备。目前的办公设备中多数都嵌入了单片机系统, 如打印机、复印机、扫描仪等。
- (5) 军用航空等尖端领域的应用更加突出。

1.2 单片机系统的组成

当单片机内部的计算机外围功能单元不能满足对象控制要求的时候, 通过系统扩展,

在外部并行总线上扩展相应的计算机外围功能单元所构成的系统，称为单片机系统。这个定义强调的是单片机系统的硬件组成，而一个完整的单片机系统应该包括硬件系统和软件系统两大部分，如图 1-2 所示。

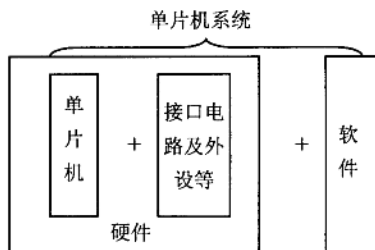


图 1-2 单片机系统结构框图

1.2.1 单片机系统的硬件组成

单片机系统的硬件由单片机芯片和外部接口电路及设备组成。而单片机芯片则包括了中央处理器（CPU）、存储器（ROM/RAM）、I/O 接口及其他功能单元（定时计数器、中断系统、串行接口）。它们通过 AB（地址总线）、DB（数据总线）、CB（控制总线）相互连接，如图 1-3 所示。

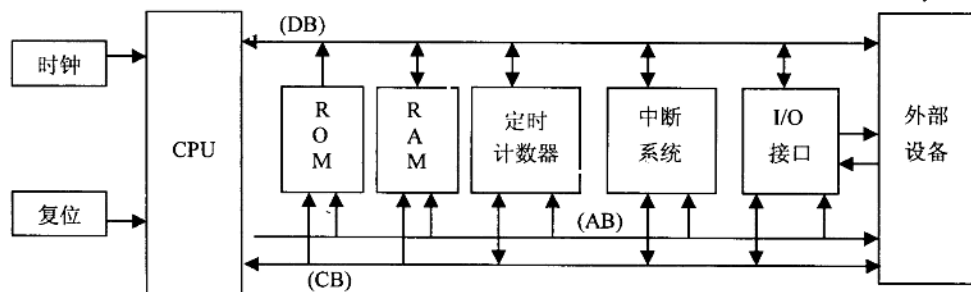


图 1-3 单片机系统的硬件结构图

1. 微处理器（CPU）

CPU 主要由两部分组成：运算器和控制器。

（1）运算器

运算器主要完成算术运算和逻辑运算并进行逻辑测试，如零值测试和两个值的比较，通常算术操作产生一个运算结果，而一个逻辑操作产生一个判决。运算器主要由以下几部

分组成。

① 累加器 A: 用来存放参与算术或者逻辑运算的一个操作数和运算结果。

② 暂存器 TMP: 用来暂时存放参与算术或者逻辑运算的另一个操作数。该操作数主要来自其他数据寄存器或者内存单元中。

③ 算术逻辑单元 ALU: 主要完成把传送到微处理器的数据进行算术和逻辑运算。ALU 具有两个主要的输入来源, 一个累加器, 一个来自数据寄存器, 它能够完成这两个数的相加和相减, 也能够完成某些逻辑运算。

④ 标志寄存器 F: 用来存放 ALU 运算结果的标志位, 如进位标志、溢出标志等。

例如: 将两个数 12H 和 31H 相加, 在相加之前, 操作数 12H 存放在累加器 A 中, 31H 存放在数据寄存器中, 执行两个数相加的控制线发出“加”操作信号, ALU 就把两个数相加, 所加结果存入累加器 A 中, 覆盖原来累加器 A 中的内容, 执行完后, 累加器 A 中的内容就是 43H。

(2) 控制器

控制器主要是协调和控制整个计算机系统的操作, 主要由以下几部分组成。

① 程序计数器 PC: 为了能够保证程序能够连续地运行下去, CPU 必须具备某种手段来确定一条指令的地址, 程序计数器就是执行这项工作。当执行程序时, CPU 将自动修改 PC 的内容, 使之总是指向下一条指令的存放地址。

② 指令寄存器 IR: 保存当前正在执行的一条指令。当执行一条指令时, 先把它从程序存储器中取出, 然后再传送到指令寄存器。

③ 指令译码器 ID: 要执行给定指令的操作, 必须给操作码译码, 以便确定所要求的操作, 指令译码器就是负责这项工作。操作码一经译码后就向操作控制器发出具体操作的特定信号。

④ 时序产生器: 用来产生脉冲序列和各种节拍脉冲, 每个节拍脉冲对应于一种操作, 就像是体育老师的口令、交响乐团的指挥一样。

⑤ 操作控制器: 根据指令译码器的信号, 产生相应的操作控制信号, 以便启动规定的动作, 比如一次内存读写操作, 一次算术逻辑操作, 一次输入/输出操作, 指挥并控制 CPU、内存和输入/输出设备之间的数据流向。

相对控制器而言, 运算器的动作是接受控制器的命令而动作的, 即运算器所进行的所有操作都是由控制器发出的控制信号来指挥的。

2. 总线

总线 (BUS) 是计算机各部件之间传送信息的公共通道。微机中有内部总线和外部总线两类。内部总线是 CPU 内部之间的连线, 外部总线是指 CPU 与其他部件之间的连线。外部总线有三种: 数据总线 DB (Data Bus)、地址总线 AB (Address Bus) 和控制总线 CB (Control Bus)。

(1) 地址总线 (AB): 地址总线宽度根据寻址范围来确定的, 当寻址范围是 $4KB=2^{12}B$ 时, 需要的地址线宽度为 12 位, 由于 80C51 单片机的寻址范围最多可达 64KB, 因此地址总线宽度为 16 位, 由 P0 口经地址锁存器提供低 8 位地址 (A0~A7); P2 口直接提供高 8 位地址 (A8~A15)。因地址信号是由 CPU 发出的, 故地址总线是单方向的。

(2) 数据总线 (DB): 由于 80C51 单片机为 8 位机, 故数据总线宽度为 8 位, 用于传送数据和指令, 由 P0 口提供。

(3) 控制总线 (CB): 控制总线随时掌控各种部件的状态, 并根据需要向有关部件发出控制命令。

3. 存储器

存储器的主要功能就是用来存放程序代码和数据。

(1) 存储器的分类

① 按照存储器的存取功能分, 可分为:

- 随机存取存储器 (Random Access Memory, RAM)。它可以随机写入和读出, 读写速度快, 但是断电后, 存储的数据就要丢失。主要用来存放各种处理数据。
- 只读存储器 (Read Only Memory, ROM)。它在一般情况下只能读不能写, 当然, 在满足一定条件下也可以完成写入操作, 否则, 读的数据又从何而来呢? 它的另外一个特点就是掉电不丢失, 能长期保存数据。从写入方式来看, ROM 也有很多分类, 主要有: 掩膜 ROM、EPROM、EEPROM、FLASHROM 和 OTPROM (一次性编程 ROM)。

② 按照存储器结构分, 可分为:

- 普林斯顿结构。一般微机只有一个地址空间, ROM 和 RAM 可以随意安排在这一地址范围内不同的空间, 即 ROM 和 RAM 的地址在同一个队列里分配不同的地址空间。CPU 访问存储器时, 一个地址对应唯一的存储空间, 可以是 ROM, 也可以是 RAM, 并用同一种指令访问。
- 哈佛结构。如 80C51 的存储器结构分程序存储器空间和数据存储器空间, 总共有 4 个物理存储空间, 即片内程序存储器空间、片外程序存储器空间、片内数据存储器空间、片外数据存储器空间, 并用不同的指令访问程序存储器和数据存储器, 这种程序存储器和数据存储器分开的结构就称为哈佛结构。

(2) 存储器的操作

存储器的操作分为读操作和写操作。

例如, 将数据存储器 30H 中的内容 03H 读出到累加器 A 中。其步骤如下:

- ① CPU 将要读存储空间地址码 30H 送到地址总线上, 选通地址为 30H 的存储单元。
- ② CPU 的控制器发出“读”信号, 建立存储器到 CPU 的数据流向。
- ③ 存储器将地址为 30H 的存储单元中的内容 03H 释放到数据总线上。

④ CPU 将数据总线上的数据 03H 读入到累加器 A 中。

注意：读操作不影响原来单元内容，即 30H 中的内容还是 03H，类似计算机操作中的“复制”。

例如，将数据 10H 送到地址为 40H 的内部数据存储单元中。其步骤如下：

① CPU 将要写入的存储空间地址码 40H 送到地址总线上，选通地址为 40H 的存储单元。

② 将数据 10H 送到数据总线上。

③ CPU 控制器发出“写”信号，建立 CPU 到存储器的数据流向。

④ 存储器将数据 10H 送入到地址为 40H 的存储单元中。

注意：写操作要改变原来单元的内容，类似计算机操作中的“粘贴”，如果 40H 单元中原来的内容是 20H，经过这次写操作后，40H 单元中的内容就改为 10H。

(3) 堆栈

堆栈就是在单片机 RAM 中，专门划出一个区域用来临时存放一些重要数据码或者地址码。对于堆栈，应主要掌握以下几点。

① 栈底地址：用来确定堆栈的深度，一般可在程序初始化部分通过对堆栈指针 SP 的赋值来确定，如 MOV SP,#60H，就是将堆栈的栈底设定为 60H（深度为 32B）。

② 堆栈指针：用来指出当前栈顶的存储单元的地址。

③ 堆栈原则：堆栈操作遵循“先进后出”的原则。

如图 1-4 是一堆栈的结构图。通过这个图可以了解以下几点：

① 这个堆栈的栈底为 50H，在程序初始化过程中可以通过指令 MOV SP,#50H 来指定，同时确定堆栈深度为 51H~7FH，在具体使用中的堆栈深度根据实际需要保存的数据个数来确定。

② 当前堆栈指针 SP 的值为 54H，即当前栈顶为 54H。

③ 这 4 个数据的压栈顺序为 10H、14H、58H、64H，根据“先进后出”原则，这 4 个数据的出栈顺序为 64H、58H、14H、10H。

从上面几点看，堆栈与手枪弹夹的结构和工作原理十分相似。

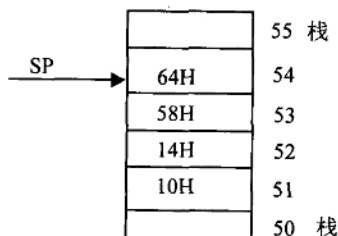


图 1-4 堆栈结构示意图

4. 输入/输出设备及其接口电路

单片机系统的输入/输出设备也称作 I/O 设备, 比如键盘、鼠标、显示器、微型打印机等。输入/输出 (I/O) 接口由大规模集成电路组成的 I/O 器件构成, 用来连接主机和相应的 I/O 设备 (如: 键盘、鼠标、显示器、打印机等), 使得这些设备和主机之间传送的数据、信息在形式上和速度上都能匹配。不同的 I/O 设备必须配置与其相适应的 I/O 接口。通常, 这些接口电路集成在一块芯片上。如 8255A、8155、8253、DAC0832、ADC0809 等扩展芯片。

1.2.2 单片机系统软件组成

只配备硬件设备的单片机系统, 并不能完成我们所需要实现的功能, 硬件是实现功能的载体, 而软件是硬件的灵魂, 目前用来设计软件的语言分为三类。

1. 机器语言

机器语言的主要特点是:

- (1) 由一组二进制码组成, 直接能被计算机识别并执行。
- (2) 不同的微处理器, 采用不同的机器语言。
- (3) 可读性差, 直观性差, 容易出错, 目前基本不采用。

2. 汇编语言

汇编语言的主要特点是:

- (1) 用助记符代替机器语言中的操作码。
- (2) 汇编语言翻译成机器语言的方法有两种: 一种是手工查表汇编, 一种是机器汇编。由于机器汇编方便, 目前基本上都是采用机器汇编。
- (3) 不同的微处理器采用不同的汇编语言。
- (4) 较直观, 可读性好, 占用内存少, 速度快。

3. 高级语言

高级语言的主要特点是:

- (1) 采用类似人类自然语言的程序设计语言。
- (2) 通过专门的编译程序翻译成机器语言。
- (3) 通用性强, 移植性好, 不随微处理器的不同而不同。
- (4) 目前较流行的有 C51、PL/M 和 BASIC 语言。

三种语言各有特点, 本书介绍的是汇编语言, 虽然不同类型单片机的汇编语言有所不同, 但还是有很多相似之处, 可以举一反三。同时, 在掌握汇编语言的基础上, 再去掌握高级语言的编程, 能达到事半功倍的效果。

1.3 单片机中数的表示及编码

在日常生活中,我们经常使用的是十进制数,而在单片机中,二进制机器编码是它的基本语言,考虑到二进制在书写和阅读方面的缺点,在单片机软件编写过程中,常引入十六进制数来表示。

如何进行十进制、二进制、十六进制之间的相互转换,是基本的数字处理能力,这里不再详述。对此有疑问的初学者,可以参考有关的数字电路基础的数据,本书主要介绍正负数在计算机中的表示方法和有关编码知识。

1.3.1 正数和负数在单片机中的表示方法

在日常数字表示中,有符号数的正负性可以用“+”、“-”来表示,但是在计算机数字表示中,有符号数的正负根据其最高位是“0”还是“1”来区分。如在8位微机中,D7位表示这个数的符号,是“1”,表示负数,是“0”,表示正数,其余位表示数值位的大小,如图1-5所示。

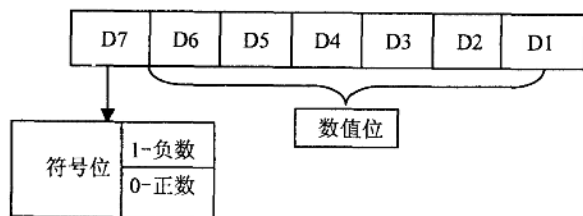


图 1-5 8位有符号数的表示方法

【例 1-1】指出 N1 和 N2 的值。

N1:	1	0	0	0	0	1	1	0
N2:	0	0	0	0	0	1	1	1

解:

由于 N1 的 D7 位为 1, 是负数, 后面的数值位的值为 6, 则 $N1 = -6$;

由于 N2 的 D7 位为 0, 是正数, 后面的数值位的值为 7, 则 $N2 = +7$ 。

那么, 在计算机中, -6 和 $+7$ 的存储形式真的如此吗? 我们来看一下下面的例子。

【例 1-2】求 $S = N1 + N2 = -6 + 7$ 。

解: 根据例 1-1 的结果, $-6 = 1000\ 0110B$, $+7 = 0000\ 0111B$, 则运算过程如下:

$$\begin{array}{r}
 10000110\text{B} \quad \text{N1} \\
 +00000111\text{B} \quad \text{N2} \\
 \hline
 10001101\text{B} \quad \text{S}=-13
 \end{array}$$

S=-13, 很明显, 结果错误, 那么问题出在哪里呢? 主要是因为符号数在计算机中的表示方法出现错误。在计算机中, 有符号数的表示方法有 3 种: 原码、反码和补码。我们上面两个例子采用的都是原码, 但是, 计算机中, 负数是以补码形式存储并参与运算的。下面我们来具体了解正负数的原码、反码和补码。

1. 正数的原码、反码、补码

正数的表示最简单, 它的原码、反码和补码都一样, 即

$$[\text{X}]_{\text{原}} = [\text{X}]_{\text{反}} = [\text{X}]_{\text{补}} = \text{X}$$

【例 1-3】X=+9, 求其原码、反码、补码。

解: +9=00001001B, 则

$$[\text{+9}]_{\text{原}} = [\text{+9}]_{\text{反}} = [\text{+9}]_{\text{补}} = 00001001\text{B}$$

2. 负数的原码、反码和补码

负数的原码: 符号位为 1, 其余位为数值位。

【例 1-4】X=-9, 求 X 的原码。

解: 符号位为 1, 数值位为 9=0001001

所以, $[-9]_{\text{原}} = 10001001\text{B}$ 。

负数的反码: 原码的符号位不变, 其余位取反。

【例 1-5】X=-9, 求 X 的反码。

解: $[-9]_{\text{原}} = 10001001\text{B}$, 根据符号位不变, 其余位取反的方法可得,

$$[-9]_{\text{反}} = 11110110\text{B}$$

负数的补码: 反码加 1, 即 $[\text{X}]_{\text{补}} = [\text{X}]_{\text{反}} + 1$ 。

【例 1-6】X=-9, 求 X 的补码。

解: $[-9]_{\text{原}} = 10001001\text{B}$, $[-9]_{\text{反}} = 11110110\text{B}$, 根据补码等于反码加 1, 则

$$[-9]_{\text{补}} = [-9]_{\text{反}} + 1 = 11110110\text{B} + 1 = 11110111\text{B}$$

3. 零的原码、反码和补码

零的原码: 由于零分为+0和-0, 则其原码有两种:

$$[\text{+0}]_{\text{原}} = 00000000\text{B}$$

$$[\text{-0}]_{\text{原}} = 10000000\text{B}$$

零的反码也有两种:

$$[\text{+0}]_{\text{反}} = 00000000\text{B}$$

$$[-0]_{\text{反}} = 11111111\text{B}$$

零点补码：由于+0的补码等于原码，-0的补码等于其反码加1，所以，不管是+0还是-0，它的补码只有一个：

$$[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000\text{B}$$

综上所述，归纳如下。

对于正数： $[X]_{\text{原}} = [X]_{\text{反}} = [X]_{\text{补}} = X$

对于负数： $[X]_{\text{反}} = [X]_{\text{原}}$ 数值位取反，符号位不变

$$[X]_{\text{补}} = [X]_{\text{反}} + 1$$

学会了补码的运算，我们回过来看例 1-2，采用补码运算，

$$[-6]_{\text{补}} = 11111010\text{B}$$

$$[+7]_{\text{补}} = 00000111\text{B}$$

则 $S = -6 + 7$ 的运算如下：

$$\begin{array}{r} 11111010\text{B} \quad N1 \\ +00000111\text{B} \quad N2 \\ \hline 00000001\text{B} \quad S=1 \end{array}$$

结果正确，至此，大家应该明白在计算机中，有符号数的运算采用的是数的补码形式。

1.3.2 常用编码

1. BCD 码

采用二进制码对每一个十进制数进行编码，称为 BCD 码，用 $[\dots\dots]_{\text{BCD}}$ 表示，例如：

$51 = [0101\ 0001]_{\text{BCD}}$ ，在这里， $[0101\ 0001]_{\text{BCD}}$ 不能认为是二进制码 01010001B，因为 01010001B 的十进制为 81，而 $[0101\ 0001]_{\text{BCD}}$ 的十进制为 51，显然存在区别。具体的十进制与 BCD 码之间的关系如表 1-2 所示。

表 1-2 8421BCD 编码表

十 进 制 数	8421BCD 码	十 进 制 数	8421BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

2. BCD 码的加法运算

BCD 码的加法运算与二进制码的加法运算规则一样，但有时候会出错，例如：

$$\begin{array}{r}
 \text{【0101 1001】}_{\text{BCD}} 59 \\
 + \text{【0010 1000】}_{\text{BCD}} 28 \\
 \hline
 \text{【1000 0001】}_{\text{BCD}} 81
 \end{array}$$

显然, 运算结果是错误的, 因为在运算过程中, 低四位向高四位有进位, 所以要进行修正, 具体的修正条件和方法如下:

- (1) 低四位向高四位有进位, 低四位加 6 修正。
- (2) 高四位向更高位有进位, 高四位加 6 修正。
- (3) 低四位出现非法 BCD 码, 低四位加 6 修正。
- (4) 高四位出现非法 BCD 码, 高四位加 6 修正。
- (5) 同一个四位出现两种修正条件, 则只需修正一次即可。

【例 1-7】已知 $X = \text{【0110 0001】}_{\text{BCD}}$, $Y = \text{【0010 0110】}_{\text{BCD}}$, 求 $S = X + Y$ 。

解:

$$\begin{array}{r}
 \text{【0110 0001】}_{\text{BCD}} X (61) \\
 + \text{【0010 0110】}_{\text{BCD}} Y (26) \\
 \hline
 \text{【1000 0111】}_{\text{BCD}} S (87)
 \end{array}$$

不满足要修正的条件, 无需修正, 结果正确。

【例 1-8】已知 $X = \text{【0100 1000】}_{\text{BCD}}$, $Y = \text{【0101 1001】}_{\text{BCD}}$, 求 $S = X + Y$ 。

解:

$$\begin{array}{r}
 \text{【0100 1000】}_{\text{BCD}} X (48) \\
 + \text{【0101 1001】}_{\text{BCD}} Y (59) \\
 \hline
 \text{【1010 0001】}_{\text{BCD}} \quad ; \text{满足 (1)、(4) 修正条件, 进行加 6 修正} \\
 + \quad \text{0110 0110} \\
 \hline
 \text{【0001 0000 0111】}_{\text{BCD}} S (107)
 \end{array}$$

结果: $S = \text{【00010000 0111】}_{\text{BCD}}$, 由于高四位加 6 后, 向更高位有进位 1, 其实也就是向百位有进位, 所以最后的结果应为 107。

3. BCD 码减法

BCD 码在进行减法运算时, 也会出现需要修正的现象, 其修正条件和方法如下:

- (1) 低四位向高四位有借位, 低四位减 6 修正。
- (2) 高四位向更高位有借位, 高四位减 6 修正。
- (3) 低四位出现非法 BCD 码, 低四位减 6 修正。
- (4) 高四位出现非法 BCD 码, 高四位减 6 修正。
- (5) 若同一个四位出现两种修正条件, 则只需修正一次即可。

【例 1-9】已知 $X = \text{【0010 0010】}_{\text{BCD}}$, $Y = \text{【0001 0001】}_{\text{BCD}}$, 求 $S = X - Y$ 。