

第一章 微型计算机基础知识

单片机技术是计算机技术的一个独特分支，在众多领域尤其是在智能仪表、检测和控制系统中单片机有着广泛的应用。读者通过对本章的学习，能够对单片机有一个初步的认识，了解单片机的发展概况、特点和应用，同时对计算机中的数和计算机语言有一个基本的认识，为后续章节的学习打下基础。

第一节 单片机的发展和特点

一、什么是单片机

所谓单片机就是在一片芯片上集成了 CPU、ROM、RAM、定时/计数器和多种 I/O 接口电路等而具有一定规模的微型计算机。

一个最基本的微型计算机通常由以下几部分组成：

- 1) 中央处理器 (CPU)，包括 ALU、控制器和寄存器组。
- 2) 存储器，包括 ROM 和 RAM。
- 3) 输入输出 (I/O) 接口，与外部输入输出设备连接。

随着超大规模集成电路技术的发展和计算机微型化的需要，把上述微型计算机的基本功能部件，全部集成在一片半导体芯片上，使得一片集成电路芯片就是一个完整的微型计算机。这种集成电路芯片被称为单片微型计算机 (Single Chip Microcomputer)，简称单片机。由于它的结构及功能均是按工业控制要求设计的，所以其确切名称应是单片微控制器 (Single Chip Microcontroller)。

二、单片机发展概况

自从 1975 年美国德州仪器公司 (Texas Instruments) 的第一个单片微型计算机 (简称单片机) TMS-1000 问世以来，迄今为止，仅有 20 多年的历史，单片机技术已成为计算机技术的一个独特分支，在众多领域尤其是在智能仪表、检测和控制系统中，单片机有着广泛的应用。

单片机作为微型计算机的一个分支，它的产生与发展和微处理器的产生与发展大体同步，主要分为三个阶段。

第一阶段 (1974~1978)：初级单片机阶段，以 Intel 公司的 MCS-48 为代表。这个系列的单片机在片内集成了 8 位 CPU、并行 I/O 口、8 位定时器/计数器、RAM 等，无串行 I/O，寻址范围不大于 4KB。

第二阶段 (1978~1983)：高性能单片机阶段，以 MCS-51 系列为代表，这个阶段的单片机均带有串行 I/O 口，具有多级中断处理系统，定时器/计数器为 16 位，片内 RAM 和 ROM 容量相对增大，且寻址范围可达 64KB。这类单片机应用领域极其广泛，由于其优良的性价比，特别适合我国的国情，故在我国得到广泛应用。

第三阶段 (1983~)：8 位单片机巩固完善及 16 位单片机推出阶段，16 位单片机除了

CPU 为 16 位以外，片内 RAM 和 ROM 的容量进一步增大，片内 RAM 增加为 232B，ROM 为 8KB，且片内带有高速输入输出部件、多通道 10 位 A/D 转换器，具有 8 级中断等。近年来，32 位单片机也已进入实用阶段。

三、MCS 系列单片机简介

国际市场上有众多类型的单片机，由于种种原因，国内广泛使用的是 Intel 公司生产的 MCS-48、MCS-51、MCS-96 等三个系列的几十种产品。由于 MCS-48 单片机逐步趋于淘汰，而 MCS-51 系列单片机基本上可以满足用户的一般要求，另外，MCS-96 系列的应用也日趋广泛，故下面主要介绍这两个系列的产品。

1. MCS-51 系列单片机

MCS-51 系列单片机是高档 8 位机。它与 MCS-48 系列相比，性能全面提高；其许多功能也超过了 8085CPU 和 Z80 CPU，成为当前工业测控类应用系统的优选单片机。

与 MCS-48 系列相比，MCS-51 单片机的性能提高主要有：片内存储器容量和外部存储器寻址范围增大；I/O 口线数增加；定时器 / 计数器的数量和位数增加；中断功能加强；设置了一个全双工串行口，具有通信能力；指令系统增加了乘除等指令，并加强了位操作功能。

MCS-51 系列单片机包括下列型号：

(1) 8051/8751/8031 这 3 种芯片常称为 8051 子系列，它们之间的区别仅在于片内程序存储器不同。8051 片内有 4KB 的 ROM，8751 片内有 4KB 的 EPROM，8031 片内无程序存储器，其他结构性能相同。其中 8031 易于开发，价格低，应用广泛。

(2) 8052/8752/8032 这是 8051/8751/8031 的改进型，常称为 8052 子系列。其片内 ROM 和 RAM 容量比 8051 各增加一倍，ROM 为 8KB，RAM 为 256B，另外增加了一个定时器 / 计数器和一个中断源。

(3) 80C51/87C51/80C31 这 3 个型号是 8051 子系列的 CHMOS 型芯片，可称为 80C51 子系列，两者功能兼容。CHMOS 型芯片的基本特点是集成度高和功耗低。

2. MCS-96 系列单片机

Intel 公司于 1983 年推出的 MCS-96 系列 16 位单片机微机是目前性能最好的单片机产品之一。它包括一个高性能的 16 位 CPU、8KB 的程序存储器、232B 的数据存储器、功能丰富的 I/O 接口、10 位 A/D 转换器等，其主要性能特点为：

1) 高性能的 16 位 CPU，主频 12MHz，采用了新颖的寄存器堆 / 逻辑部件 (RALU)。该 CPU 的最大特点是采用寄存器-寄存器结构，CPU 直接面向 256B 的寄存器空间，消除了累加器结构中存在的前瓶颈效应，大大提高了操作速度和数据吞吐能力。

2) 具有 8KB 的内部程序存储器，内部程序存储器可以加密。具有 256B 的内部寄存器和专用寄存器，其中 232B 为内部寄存器，它兼有通用寄存器和高速 RAM 的功能。

3) 4 条高速触发输入线，6 条高速脉冲输出线，同时具有 2 个 16 位定时器。另外，还有 4 个受高速输出部件控制的软件定时器，具有 9 个中断源，8 个中断优先级。

4) 有的芯片有 10 位 A/D 转换器。

5) 运算能力和运算速度大大提高，在 12MHz 频率输入下，可实现 16 位加法运算 ($1\mu\text{s}$)，16 位 \times 16 位乘法和 16 位 \div 16 位除法运算 ($6.5\mu\text{s}$)。许多指令既可用于双操作数，也可用于三操作数，平均指令执行时间为 $1\sim 2\mu\text{s}$ 。

6) 可使用面向工业控制的高级语言，如 C 语言、Fortran 语言、PL/M 语言等。

MCS-96 系列包括有多种型号产品，其中 8098 单片机以 8 位机的价格而具有 16 位机的性能，受到用户的欢迎。这一类单片机有 3 个型号：8098 片内无 ROM，8398 片内有 8KB ROM，8795 片内有 8KB EPROM。

Intel 公司生产的主要单片机系列性能介绍见表 1-1。

表 1-1 Intel 公司主要单片机系列

系列	型号	片内存储器 /B		片外存储器直接寻址范围/B		I/O 口线		中 断 源	定时器/ 计数器 (个×位)	晶振 /MHz	典型 指令 周期 /μs	封装 DIP	其他
		ROM/ EPROM	RAM	RAM	EPROM	并 行	串 行						
MCS-48 (8 位机)	8048	1K/	64	256	4K	27		2	1×8	2~8	1.9	40	
	8748	/1K	64	256	4K	27		2	1×8	2~8	1.9	40	
	8035	—	64	256	4K	27		2	1×8	2~3	1.9	40	
	8049	2K/	128	256	4K	27		2	1×8	2~11	1.36	40	
	8749	/2K	128	256	4K	27		2	1×8	2~11	1.36	40	
	8039	—	128	256	4K	27		2	1×8	2~11	1.36	40	
MCS-51 (8 位机)	8051	4K/	128	64K	64K	32	UART	5	2×16	2~12	1	40	
	8751	/4K	128	64K	64K	32	UART	5	2×16	2~12	1	40	
	8031	—	128	64K	64K	32	UART	5	2×16	2~12	1	40	
	8052AH	8K/	256	64K	64K	32	UART	5	3×16	2~12	1	40	
	8752AH	/8K	256	64K	64K	32	UART	5	3×16	2~12	1	40	
	8032AH	—	256	64K	64K	32	UART	5	3×16	2~12	1	40	
	80C51BH	4K/	128	64K	64K	32	UART	5	2×16	2~12	1	40	
	80C31BH	—	128	64K	64K	32	UART	5	2×16	2~12	1	40	CHMOS
	87C51BH	/4K	128	64K	64K	32	UART	5	2×16	2~12	1	40	
	80C252	8K/	256	64K	64K	32	UART	7	3×16	2~12	1	40	CHMOS, 有 脉宽调制输 出, 高速输 出片内固化
	87C252	/8K	256	64K	64K	32	UART	7	3×16	2~12	1	40	
	83C252	—	256	64K	64K	32	UART	7	3×16	2~12	1	40	有 BASIC 解释程序
MCS-96 (16 位机)	8094	—	232	64K	64K	32	UART	8	4×16 软件	12	1~2	48	
	8095	—	232	64K	64K	32	UART	8	4×16 软件	12	1~2	48	4×10 位 A/D
	8096	—	232	64K	64K	48	UART	8	4×16 软件	12	1~2	68	
	8097	—	232	64K	64K	48	UART	8	4×16 软件	12	1~2	68	8×10 位 A/D
	8394	8K/	232	64K	64K	32	UART	8	4×16 软件	12	1~2	48	
	8395	8K/	232	64K	64K	32	UART	8	4×16 软件	12	1~2	48	4×10 位 A/D
	8396	8K/	232	64K	64K	48	UART	8	4×16 软件	12	1~2	68	
	8397	8K/	232	64K	64K	48	UART	8	4×16 软件	12	1~2	68	8×10 位 A/D
	8095BH	—	232	64K	64K	48	UART	8	4×16 软件	12	1~2	48	8×10 位 A/D
	8396BH	8K/	232	64K	64K	48	UART	8	4×16 软件	12	1~2	68	
8797BH	/8K	232	64K	64K	48	UART	8	4×16 软件	12	1~2	48	8×10 位 A/D	
准 16 位机	8098	—	232	64K	64K	32	UART	8	4×16 软件	12	1~2	48	4×10 位 A/D

四、单片机的应用

单片机在控制应用领域中，有如下几方面的优点：

1) 体积小、成本低、运用灵活、易于产品化，它能方便地组成各种智能化的控制设备和仪器，做到机电仪一体化。

2) 面向控制，能针对性地解决从简单到复杂的各类控制任务，因此，能获得最佳的性能价格比。

3) 抗干扰能力强，适应温度范围宽，在各种恶劣的环境下都能可靠地工作，这是其他机种无法比拟的。

4) 可以方便地实现多机和分布式控制，使整个控制系统的效率和可靠性大为提高。

单片机的应用范围很广，根据使用情况大致可分为如下四大类：

1. 单片机在智能仪器仪表中的应用

单片机具有体积小、功耗低、控制功能强等优点，故可广泛应用于各类仪表（包括湿度、温度、流量、流速、电压、频率、功率、厚度、角度、长度、硬度、元素、压力测定等）中。引入单片机使得仪器仪表数字化、智能化、微型化，且功能大大提高。例如，精密数字温度计、智能电度表、智能流速仪、微机多功能 pH 测试仪等等。

2. 单片机在工业测控中的应用

用单片机可以构成各种工业测控系统、自适应控制系统、数据采集系统等。例如，MCS-51 单片机控制电镀生产线、温室人工气候控制、报警系统控制、IBM-PC/XT 和单片机组成的二级计算机控制系统等。

3. 单片机在计算机网络与通信技术中的应用

MCS 系列单片机具有通信接口，为单片机在计算机网络与通信设备中的应用提供了良好的条件。例如，MCS 系列单片机控制的串行自动呼叫应答系统、列车无线通信系统、MCS-51 单片机控制无线遥控系统等。

4. 单片机在日常生活及家电中的应用

单片机愈来愈广泛地应用于日常生活中的智能电气产品以及家电中，例如，电子称、银行计息电脑、电脑缝纫机、心率监护仪、电冰箱控制、彩色电视机、洗衣机控制等等。

要开发单片机的应用系统，不但要掌握单片机硬件和软件方面的知识，而且还要深入了解各应用系统的专业知识，只有将这两方面的知识融会贯通和有机结合，才能设计出优良的应用系统。

第二节 计算机中数的表示方法

一、进位计数制及其相互转换

1. 二进制数

或许是双手为十指的缘故，人们习惯于用十进制，逢十进一。事实上，人们还采用了其他一些进制，如六十进制（ $1\text{min}=60\text{s}$ ），十二进制（1 年等于 12 个月， $1\text{ft}=12\text{in}$ ）等。人们生活中也有用二进制的，如鞋、筷子、手套等，都是逢二进一。可见用什么进制完全取决于人们的需要。计算机中的程序和数据等信息采用二进制数表示，是由于二进制在电路中最容易实现。二进制中只有“0”和“1”两个基本数字，在电路中具有两种稳定状态以代表“0”

和“1”的物理量是很多的，如电压的高和低、电灯的亮和灭、电容器的充电和放电、脉冲的有和无、晶体管的导通和截止等等。而要在电路中实现十个稳定的状态是比较困难的。此外，由于二进制的运算规则非常简单，因此，计算机运算部件的结构也可以大大简化。

二进制数的进位规则是逢二进一。二进制数的加法运算规则为： $0+0=0$ $0+1=1$ $1+0=1$ $1+1=10$ ；乘法规则为： $0\times 0=0$ $0\times 1=0$ $1\times 0=0$ $1\times 1=1$ 。可见，加法和乘法规则各有 4 条；而采用十进制，则各有规则 55 条。

计算机只能识别 0 和 1，因此，数在计算机内部是以二进制的形式表示和存储的。

2. 十六进制数

十六进制数是指逢十六进位的计数制，应有十六个不同数码，除 0~9 十个数码外，通常还借用英文字母 A、B、C、D、E、F 共六个数码。

表 1-2 是常见的十进制数与二进制数和十六进制数的对照表。

表 1-2 十、二、十六进制数的对应关系

十进制	二进制	十六进制	十进制	二进制	十六进制
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

3. 二进制、十进制、十六进制数的相互转换

(1) 二进制与十六进制数的互换 一个 8 位二进制数，可以写成 2 位十六进制数，所以这两种进位制数进行转换时，可以把 4 位二进制数划为 1 组，然后对每 1 组进行相应的变换，例如二进制数 01100110 转换为十六进制数可写成

$$\begin{array}{cc} 0110 & 0110 \\ 6 & 6 \end{array}$$

把十六进制数 43 转换为二进制数可写成

$$\begin{array}{cc} 4 & 3 \\ 0100 & 0011 \end{array}$$

为了不使二进制、十进制或十六进制数相混淆，规定在二进制数的后面加上符号 B，如 01100110B；在十六进制数后面加上符号 H，如 66H、43H 等。如果数的后面没有符号，一般就指该数为十进制数。如果被转换的数不是整数，而是带有小数，则分组时应以小数点为基准，整数从小数点开始，从右向左每四位划为一组，如果最后一组不足 4 位，可以用零补齐，以数 1101100.11011 为例，可在数的前后补零，即

$$\begin{array}{cccc} 0110 & 1100 & 1101 & 1000 \\ 6 & C & D & 8 \end{array}$$

故：1101100.11011B=6C.D8H

(2) 二进制与十进制数的互换 对于二进制整数，各位数的位权可以用底数为 2 的 $n-1$

次幂来确定, n 表示该数的位数, 即第 1 位的位权为 $2^0=1$, 第 2 位的位权为 $2^1=2$ 若已知一个二进制数为 $10101010B$, 对应的二进制值应为

$$10101010B = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 170$$

对于二进制小数, 其小数点以后各位的位权, 可以用底数为 2 的负 n 次幂来确定, n 同样表示位数。即从小数点向右算起, 第一位的位权为 $2^{-1}=0.5$, 第二位的位权为 $2^{-2}=0.25\dots$, 例如求 $11001100.00110011B$ 的十进制值则

$$\begin{aligned} 11001100.00110011B &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &\quad + 1 \times 2^{-7} + 1 \times 2^{-8} \\ &= 204.19921875 \end{aligned}$$

反过来, 要将十进制整数转换为二进制数, 可以采用逐次除以 2 余数反序排列的方法。所谓反序排列, 指第 1 次除以 2 的余数排在最低位, 人们习惯上称为“除二取余, 自下而上”。以十进制数 25 为例, 逐次除以 2 列式如下

$$\begin{array}{r} 25 \div 2 = 12 \quad \text{余 } 1 \uparrow \\ 12 \div 2 = 6 \quad \text{余 } 0 \\ 6 \div 2 = 3 \quad \text{余 } 0 \\ 3 \div 2 = 1 \quad \text{余 } 1 \\ 1 \div 2 = 0 \quad \text{余 } 1 \end{array}$$

由于 8 位微型计算机习惯将二进制数写成 8 位, 可得

$$25 = 00011001B$$

如果二进制数不超过 8 位, 即十进制数不超过 255, 也可以不必列式, 而直接用口算转换。

要把十进制小数转换为二进制数, 可以采用小数部分逐次乘以 2, 其乘积的整数个位 (即所谓溢出位) 按正序排列的方法, 人们习惯上称为“乘二取整, 自上而下”。以 33.6875 为例, 其整数部分

$$33 = 00100001B$$

其小数部分

$$\begin{array}{r} 0.6875 \times 2 = 1.375 \quad \text{溢出为 } 1 \mid \\ 0.375 \times 2 = 0.75 \quad \text{溢出为 } 0 \mid \\ 0.75 \times 2 = 1.5 \quad \text{溢出为 } 1 \mid \\ 0.5 \times 2 = 1 \quad \text{溢出为 } 1 \downarrow \end{array}$$

可得出 $33.6875 = 00100001.10110000B$

(3) 十进制数与十六进制数的互换 我们已经掌握了十进制数与二进制数的互换以及二进制数与十六进制数的互换。因此, 要把十进制数转换为十六进制数, 可以先转换成二进制数, 再改写成十六进制数。反之, 十六进制数也可以先改写成二进制数, 再换成十进制数。

在程序中书写十六进制数时, 若打头的数为 $A \sim F$, 则应在 $A \sim F$ 之前再加一个 0, 以表示这是一个数而不是其他符号。

例 1 十进制数 33.6875 转换为十六进制数。

$$33.6875 = 00100001.10110000B = 21.BH$$

二、带符号的二进制数

1. 带符号的二进制数与无符号的二进制数

在数学运算中，表示一个数的正负，可以在数的前面冠以正号或负号。但计算机只能辨认 0 或 1 两个数码，不能辨认其他符号，因此，在字长为 8 位的二进制数中，将最高位规定为符号位，最高位为 0，表示该数为正；最高位为 1，表示该数为负。设有两个带符号数： $X = +91$ ， $Y = -91$ ，它们在 8 位字长的机器中可表示为

$$X = 01011011 \quad Y = 11011011$$

为了区分原来的数与它在机器中的表示形式，我们将一个数连同它的符号在机器中以数码化形式表示的数称为机器数，而把原来按一般书写形式表示的数称为机器数的真值或真数。

如机器数 11011011 的真值为 -91 或记为 -1011011B。

应该注意，同样一个二进制数，它既可以是无符号数，也可以是带符号数。例如，二进制数 11001100B，既可以是无符号数 204，也可以是带符号数 -76。到底它是 204 还是 -76，取决于事先约定，仅从数的本身无法判别它属于什么数。

下面要介绍的相对转移指令中的偏移量，约定必须使用带符号数，因此，出现在偏移量中的二进制数，总是一个带符号数，在填写偏移量时，应使用带符号数，而程序中的地址值，约定为无符号数，对应使用无符号数。

总之，一个数是带符号的还是无符号的二进制数，从数的本身是无法区别的，只能根据它出现的场合，以及该场合约定使用什么数才能区分它们。

2. 带符号数的表示方法

由于带符号数涉及到符号位的处理，所以运算比较复杂，为了简化带符号数的运算，可采用不同的编码方法表示带符号的数，即原码、反码、补码表示法。

(1) 原码 表示一个带符号的二进制数，其最高位为符号位，正数符号位为 0，负数符号位为 1，其余位表示数值。例如

$$X = +1010101B \quad [X]_{\text{原码}} = 01010101B$$

$$X = -1010101B \quad [X]_{\text{原码}} = 11010101B$$

式中， X 为真值； $[X]_{\text{原码}}$ 表示真值 X 的原码，对于数 0，其原码可以表示为

$$[+0]_{\text{原码}} = 00000000B \quad [-0]_{\text{原码}} = 10000000B$$

(2) 反码 表示一个带符号的二进制数，其最高位为符号位，正数的符号位为 0，负数的符号位为 1，正数的反码与原码相同；负数的反码表示是在其原码的基础上保持符号位不变，其余各位逐位取反而得。例如

$$X = +1010101B \quad [X]_{\text{反码}} = 01010101B$$

$$X = -1010101B \quad [X]_{\text{反码}} = 10101010B$$

对于数 0 的反码，也有两种形式： $[+0]_{\text{反码}} = 00000000$ $[-0]_{\text{反码}} = 11111111$

(3) 补码表示一个带符号的二进制数，其最高位为符号位，正数的符号位为 0，其余各位表示数值，正数的补码与反码、原码完全相同；对于负数，符号位为 1，则其余各位逐位取反后再加 1，简称为取反加 1。例如：

$$X = +1010101B \quad [X]_{\text{补码}} = 01010101B$$

$$X = -1010101B \quad [X]_{\text{补码}} = 10101011B$$

对于数 0 的补码, $[+0]_{\text{补码}} = 00000000$ $[-0]_{\text{补码}} = 00000000$

表 1-3 是 8 位带符号二进数的原码、反码、补码对照表, 注意表中的 10000000 不是 (-0) 的补码, 而是 (-128) 的补码。

表 1-3 二进制数原码、反码、补码对照表

十进制数	二进制数	原码	反码	补码
+0	+0000000	0000000	0000000	0000000
+1	+0000001	0000001	0000001	0000001
+2	+0000010	0000010	0000010	0000010
⋮	⋮	⋮	⋮	⋮
+126	+1111110	01111110	01111110	01111110
+127	+1111111	01111111	01111111	01111111
-0	-0000000	1000000	1111111	0000000
-1	-0000001	1000001	1111110	1111111
-2	-0000010	1000010	1111101	1111110
⋮	⋮	⋮	⋮	⋮
-126	-1111110	11111110	1000001	1000010
-127	-1111111	11111111	1000000	1000001
-128	-1000000	无法表示	无法表示	1000000

三、二进制数的运算

1. 算术运算

二进制数和十进制数的算术运算规则基本相同, 所不同的仅仅是前者逢 2 进 1, 后者逢 10 进 1, 借位时, 从高位借 1 到低位, 前者当 2, 后者当 10。

无符号二进制数的算术运算有加法、减法、乘法和除法, 运算规则与十进制数的算术运算规则相同。

加法规则 $0+0=0$ $0+1=1$ $1+0=1$ $1+1=10$

乘法规则 $0\times 0=0$ $0\times 1=0$ $1\times 0=0$ $1\times 1=1$

带符号的二进制数, 则情况比较复杂, 因为带符号二进制数有三种表示方法, 其中反码和原码用得较少, 下面主要介绍补码加减运算。

补码运算不存在符号与数值分开的问题, 加减运算时连同符号位一起参加运算, 其运算结果是否正确, 由特殊标志判定。

设有 X、Y 两个数, 用补码表示如下

$X=10011111\text{B}$ (-97 的补码)

$Y=00001000\text{B}$ ($+8$ 的补码)

若求 $X+Y$ 之和, 可不用考虑两数的符号, 直接相加, 得出的和为 10100111B (-89 的补码), 可见直接相加结果必定是正确的。

若求 $X-Y$ 之差, 也可以直接相减, 即

$X=10011111\text{B}$ (-97 的补码)

$-Y=00001000\text{B}$ ($+8$ 的补码)

$X-Y=10010111\text{B}$ (-105 的补码)

若求 $Y-X$ 之差，同样也用减法，即

$$Y = 00001000B \quad (+8 \text{ 的补码})$$

$$-X = 10011111B \quad (-97 \text{ 的补码})$$

$$Y - X = \boxed{1}01101001B \quad (+105 \text{ 的补码})$$

丢失 ↙

也就是说做减法时，不论两数符号如何，其相减结果不论是数值还是符号都将是正确的。

在上述 $Y-X$ 算式中，最高位发生进位只是因为字长为 8 位的计算机中，若运算结果并未超出补码的记数容量（ $-128 \sim +127$ ），这时的进位被视为自然丢弃。计算机在运算中，这种自然丢弃并不影响结果的正确。

但要注意，如果得数超过 8 位补码所允许的表示范围（即超出一 $128 \sim +127$ ），则其进位称之为溢出。溢出与自然丢弃是两种不同的概念。判别属于哪一种，则要看第 7 位与第 8 位的进位情况，如果第 7 位和第 8 位同时产生进位，即所谓双进位，则这种进位属于允许的自然丢弃；如果只有第 7 位或者只有第 8 位产生进位，即只有单进位，则这种进位属于溢出。判断是否溢出，也用符号位来判别，即两个符号相同的数进行加法运算，其结果符号与参加运算的数不同，则产生溢出；两个不同符号的数进行减法运算，其结果符号与被减数不同，则产生溢出。溢出表示其数值超出计算机字长所能表示的范围，运算结果必然是错误的，因而也是不允许的。人工判断溢出是很容易的，计算机判断溢出常用溢出标志位。

例 2 求 X 、 Y 之和。

$$X = +1100110B$$

$$Y = +1000101B$$

$$[X]_{\text{补码}} = 01100110B$$

$$+ [Y]_{\text{补码}} = 01000101B$$

$$[X+Y]_{\text{补码}} = 10101011B$$

在例 2 中，只有第 7 位产生进位，第 8 位没有进位，单进位，属于溢出，同时两个正数相加，结果却为负数，这显然是错误的，也说明产生了溢出。

例 3 求 X 、 Y 之和。

$$X = -0111010B$$

$$Y = -0110111B$$

$$[X]_{\text{补码}} = 11000110B$$

$$+ [Y]_{\text{补码}} = 11001001B$$

$$[X+Y]_{\text{补码}} = \boxed{1}10001111B$$

丢失 ↙

在例 3 中，第 7、8 位都产生进位，双进位，不属于溢出，同时两个负数相加，结果仍为负数也说明无溢出。

还应注意，溢出主要是指带符号二进制数进行加减运算时，可能产生的一种结果。对于无符号数，第 8 位不是符号，只有第 8 位的进位才称为进位，而不采用溢出这个概念。

对于无符号数，还应注意一点，当两个无符号数相减时，不允许用小的数去减大的数，因为小减大，它的差一定是负数。无符号数的前提是没有符号，显然也不允许有负数，如果这

样做，减的结果也必然是错误的。

2. 逻辑运算

二进制数的逻辑运算有“与”、“或”、“异或”、“非”等。

“与”运算规则 $0 \wedge 0 = 0$ $0 \wedge 1 = 0$ $1 \wedge 0 = 0$ $1 \wedge 1 = 1$

“或”运算规则 $0 \vee 0 = 0$ $0 \vee 1 = 1$ $1 \vee 0 = 1$ $1 \vee 1 = 1$

“异或”运算规则 $0 \oplus 0 = 0$ $0 \oplus 1 = 1$ $1 \oplus 0 = 1$ $1 \oplus 1 = 0$

“非”运算规则 $1 = 0$ $0 = 1$

四、计算机中的代码

一个二进制数可以表示一个无符号数，也可以表示一个带符号数，而且可以根据事先约定代表一个文字、一个符号或者代表一个特定的内容。当它代表文字或符号时称为代码。例如 00100100B，作为数，它表示的是无符号数 24H 或是带符号数 + 24H。如果事先约定，它又能代表符号 \$。所谓约定，可以按标准约定，也可以自行约定，下面介绍两种按标准约定的常见文字符号代码。

1. BCD 码

BCD 码是一种以二进制形式表示十进制数的编码，又称为二-十进制码，它貌似二进制，实际是十进制数。

BCD 码以 4 位为一组，常用的 8421BCD 码是选用 0000B~1001B 十种状态，代表 0~9 共 10 个数，舍弃其余的 6 种状态。当 BCD 码与十进制数进行互换时，可以按 4 位一组，逐组进行互换。BCD 码表见表 1-4。

表 1-4 BCD 码表

十进制	BCD 码	二进制	十进制	BCD 码	二进制
0	0000	0000	8	1000	1000
1	0001	0001	9	1001	1001
2	0010	0010	10	无意义	1010
3	0011	0011	11	无意义	1011
4	0100	0100	12	无意义	1100
5	0101	0101	13	无意义	1101
6	0110	0110	14	无意义	1110
7	0111	0111	15	无意义	1111

例 4 将 85.62 转换为 BCD 码。

8 5 . 6 2
1000 0101. 0110 0010

例 5 将 BCD 码 10010111.01000011 转换为十进制数。

1001 0111. 0100 0011
9 7 . 4 3

要将一个二进制数转换为 BCD 码，通常先将它转换为十进制数，然后再转换为 BCD 码。同样将 BCD 转换为二进制数，也是先转换成十进制数，再转换为二进制数。

例 6 将二进制数 11010101B 转换为 BCD 码。

$11010101B = 213 = (001000010011)_{BCD}$

2. ASCII 编码

各种计算机程序语言的源程序都是以字符（字母、数字和各种符号）的形式构成的，但它们在计算机中也必须用二进制的代码来表示。用来表示字符的一组二进制代码称为二进制编码。ASCII 码就是一种常用的二进制编码。

ASCII 码是“美国标准信息交换代码”的英文缩写，它用七位二进制数码的不同组合来表示一个个字符，共可表示 128 种字符。如字符“A”的 ASCII 码为“1000001”（十进制数为 65）；字符“0”的 ASCII 码为“0110000”（十进制数为 48）；字符“+”的 ASCII 码为“0101011”（十进制数为 43）等等，详见书末附录。

第三节 计算机语言

“语言”是人们彼此交流思想、表达意愿的重要工具。人与计算机要交换信息，也是靠“语言”来实现的。要使计算机按照人的意图去工作，就必须使计算机了解人的意图，接受人向它发出的命令和提供的信息。计算机就是通过程序设计语言来理解人的意图，并按照程序设计语言所描述的解题步骤——程序而工作的。

一、程序设计语言

计算机程序设计语言通常包括机器语言、汇编语言和高级语言三大类。

1. 机器语言

由于计算机在当前的技术条件下只能识别“0”或“1”两种信号，也只能处理这种由“0”和“1”表示的二进制数码。因此，需要用“0”和“1”信号进行不同的排列组合，通过分析和译码电路产生各种控制信号，指挥计算机执行不同的操作。例如，在某种计算机中，表示加法和减法的机器指令代码分别为“10000111”和“10010111”。这种由“0”和“1”组成的命令称为机器指令。

所谓机器语言就是机器指令的集合。采用机器语言编写的程序，计算机可以直接执行。但是，机器指令辨认、记忆、书写都很困难，而且编出的程序不直观，易出错，程序设计的效率很低。因此，机器语言只是在早期的计算机上由少数专业人员所使用。

此外，不同类型的计算机逻辑结构不同，用甲种机器语言编写的程序不能直接在乙种机器上运行。所以，不同类型的机器语言互不通用。

2. 汇编语言

人们在实践中发现，可以用一些简单而又易于记忆、识别的符号来代表机器指令代码。这种用于设计程序的指令符号及其使用规则，就形成了另一种计算机语言——符号语言，通常称为汇编语言。例如上述用机器语言表示的加法指令和减法指令，采用汇编语言可分别写成“ADDA”和“SUBA”。可见，汇编语言是用一些助记符号（如英文缩写）来代替相应的机器指令。

汇编语言与机器语言相比，易写易读易记，为编写程序提供了方便，无疑是一大进步。但是，符号指令与机器指令是一一对应的关系，计算机并不能识别汇编语言，只有借助于汇编程序翻译成机器语言目标程序后才能执行。因此，一种汇编语言仍只适用于一种类型的机器。

机器语言和汇编语言都依赖于某一类型的计算机，是为某一类型的计算机服务的，因而称为面向机器的语言，它们都属于低级语言。

3. 高级语言

随着计算机的不断发展和普及，使用计算机的人越来越多。为了解决低级语言难学难用的问题，人们设计出了与计算机逻辑结构无关的计算机语言。这种摆脱了或基本摆脱了机器特点的语言，相对于低级语言来说，就称为高级语言。高级语言接近于人们习惯使用的自然语言和数学语言。因此，不具备计算机专业知识的人，也能够较快地掌握高级语言，并应用于各种科学计算、数据处理和事务管理等领域。由于高级语言不涉及计算机内部的逻辑结构，所以采用高级语言编制的程序可以适用于不同类型的计算机。

从 20 世纪 50 年代至今，世界上已出现过几百种计算机高级语言。目前国内外比较流行的有：适用于科学计算的 FORTRAN 语言；适用于计算机语言入门的 BASIC 语言；适用于系统程序设计的 C 语言；适用于结构化程序设计的 PASCAL 语言以及适用于商业和数据处理的 COBOL 语言等等。而且，计算机的高级语言仍在不断发展，可以预言，更加完善、功能更强、使用更方便的语言还将不断出现。

二、目的程序、源程序

在计算机上求解问题，必须先编好程序。但是程序必须由计算机能够接受的“语言”描述，计算机能够接受的最基本的“语言”是机器语言。用机器语言编写的程序，计算机能够直接“理解”。所以，机器语言程序就称为“目的程序”或“目标程序”。用汇编语言或高级语言编写的程序称为“源程序”。计算机不能直接执行源程序，必须先把源程序“翻译”成目的程序，计算机才能真正“理解”。这个翻译工作，由预先存放在计算机中的“翻译程序”来完成。

三、处理程序——翻译程序

源程序只有被“翻译程序”翻译成目的程序之后，才能被计算机执行。一台计算机上配备了某种语言的“翻译程序”，就意味着它“懂”得某种语言，配备的“翻译程序”越多，机器能接受的语言也就越多。

翻译程序通常有汇编程序、编译程序和解释程序三种类型。

(1) 汇编程序 汇编程序是把用汇编语言编制的源程序“翻译”成机器语言的程序。一般又把这种“翻译”过程叫做“汇编”。不同机器具有不同的汇编语言和相应的汇编程序。

(2) 编译程序 编译程序是把用高级语言编制的源程序“翻译”成机器语言的程序。编译程序的工作过程是，先将源程序全部翻译成目标程序，然后再执行。例如 FORTRAN、COBOL、PASCAL、C 等高级语言编写的源程序，计算机通常采用编译方式执行。不同的高级语言具有不同的编译程序。

(3) 解释程序 解释程序也是一种把源程序翻译成目标程序的程序。解释程序的工作过程是，逐句翻译源程序，并且译成一句马上就执行一句，即边翻译边执行。

编译程序和解释程序各有特色。编译程序是把源程序整个地翻译成目的程序后，才具体执行该目的程序，因而要占用大量的内存空间，但执行效率较高。而解释程序则是对源程序边解释边执行，所以它比编译方式要多花费机器时间，但是可以少占用机器的内存空间。高级语言中的 BASIC 语言一般是采用解释方式工作的（也有采用编译方式工作的 BASIC 语言，称为编译型 BASIC 语言）。

上述各种“翻译程序”通常由计算机制造厂家提供，用户不需自行编制。

本章小结

在一块集成电路芯片上集成有中央处理器、存储器、输入输出接口等基本功能部件的微型计算机称为单片微型计算机 (Single Chip Microcomputer)，简称单片机。

二进制数转换为十进制数 —— 加权展开。

十进制整数转换为二进制数 —— 除二取余，自下而上。

十进制小数转换为二进制数 —— 乘二取整，自上而下。

二进制数转换为十六进制数 —— 从小数点开始，四位化一位。

十六进制数转换为二进制数 —— 一位化四位。

二进制数转换为八进制数 —— 从小数点开始，三位化一位。

八进制数转换为二进制数 —— 一位化三位。

十六进制数转换为十进制数 —— 先转换为二进制数，再转换为十进制数。

十进制数转换为十六进制数 —— 先转换为二进制数，再转换为十六进制数。

BCD 码转换为二进制数或十六进制数 —— 先转换为十进制数，再转换为二进制数或十六进制数。

原码、反码、补码均是机器数，均是带符号的数在机器中的不同表示形式。

若 $X > 0$ ，则 $[X]_{\text{原码}} = [X]_{\text{反码}} = [X]_{\text{补码}}$

若 $X < 0$ ，则 $[X]_{\text{原码}}$ 符号位不变，数值位取反 $[X]_{\text{反码}}$

符号位不变，数值
位取反，末位加 1

末位加 1

$[X]_{\text{补码}}$ 连同符号位一起取反，末位加 1 $[-X]_{\text{补码}}$

习 题 一

一、填空题

1. 十进制 121 转换为 8 位二进制数，应为
2. 十进制数 112.375 转换成十六进制数是
3. 十进制数 123.45 转换成八进制数是
4. 十进制 254.28 转换为 BCD 码，应为
5. 与八进制小数 0.1 等值的十六进制小数为
6. $11101111101.1B =$ $H =$ D 。
7. $ABCD.EFH =$ B 。
8. 数字符号 0 的 ASCII 码十进制表示为 48，数字符号 9 的 ASCII 码十进制表示为
9. 一个字节的十六进制数最大相当于十进制数的 _____，两个字节的十六进制数最大值相当于十进制数的 _____。
10. 将 BCD 码 0011011101010010 转换为十进制数，应为
11. $(-51)_{10}$ 的 8 位二进制补码表示为

12. 在八位二进制的机器中，补码表示带符号整数的范围是从 _____ (小)到 _____ (大)。这两个数在机器字中的补码表示为 _____ B 和 _____ B。数字 0 的补码表示为 _____ B。

二、简答题

1. 什么是单片机？单片机与一般微型计算机相比，具有哪些特点？
2. 8 位单片机和 16 位单片机的本质区别是什么？
3. MCS-51 系列中 8031, 8051, 8751 有什么区别？
4. 单片机主要应用于哪些领域？
5. 在 MCS-51 系列单片机中，程序存储器和数据存储器最大可扩展为多少字节？
6. 在各种系列的单片机中，片内 ROM 的配置有几种形式？用户应根据什么原则选用？
7. 字符（字母、数字和各种符号）在计算机中如何表示？
8. 计算机中为什么要采用二进制？
9. 什么叫机器语言、汇编语言、高级语言？它们各有什么特点？
10. 简述计算机硬件和软件的关系。

三、计算题

1. 将下列十进制数转换为十六进制和二进制数。
 - (1) 44.375 (2) 171.96875 (3) 0.6875 (4) 127.65625
2. 将下列十六进制数转换为二进制和十进制数。
 - (1) 07E2H (2) 1FFFH (3) 3200H (4) 2F7BH2
3. 下列机器数为带符号数的补码，试写出它们对应的真值。
 - (1) E2H (2) 1FH (3) 00111100B (4) 11000011B
4. 已知 X、Y，试分别计算 $[X+Y]_{\text{补码}}$ 和 $[X-Y]_{\text{补码}}$ ，并指出是否产生溢出。
 - (1) $X=+1001011$ $Y=+0101010$ (2) $X=+1001100$ $Y=-1000101$
 - (3) $X=-0101110$ $Y=+0111110$ (4) $X=-1010100$ $Y=-1110101$

第二章 MCS-51 系列单片机的结构

MCS-51 系列单片机包括许多类型，常用的有 8051 子系列、8052 子系列、80C51 子系列等，它们的内部结构基本相同。从这一章开始将主要以 8051 子系列为背景，具体介绍 MCS-51 系列单片机的硬件、软件和接口技术。本章是从用户应用的角度出发，分析 MCS-51 单片机的内部结构和原理，目的是为了理解和掌握 MCS-51 的外特性。

第一节 MCS-51 系列单片机的内部结构

一、MCS-51 系列单片机的结构框图

前面已经提到单片机是在一块芯片集成了 CPU、RAM、ROM、定时器 / 计数器、I/O 接口以及串行通信接口等基本功能部件的一个完整的微型计算机。MCS-51 单片机内部包括以下部件：

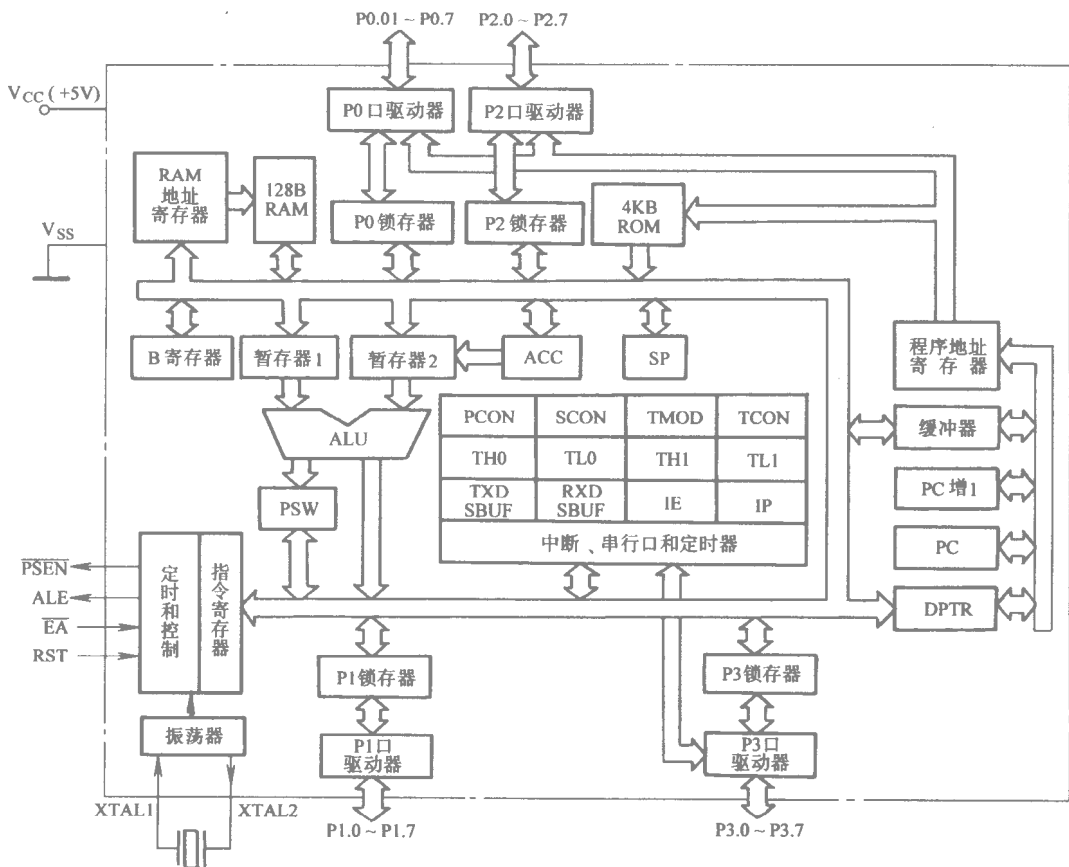


图 2-1 8051 内部结构框图

- 1) 一个 8 位 CPU。
- 2) 一个片内时钟振荡器，最高时钟频率为 12MHz。
- 3) 4KB 程序存储器 ROM/EPROM，8031 片内无 ROM。
- 4) 128B 数据存储器 RAM。
- 5) 可寻址外部程序存储器和数据存储器空间 64KB。
- 6) 21 个特殊功能寄存器 SFR。
- 7) 4 个 8 位并行 I/O 口，共 32 根 I/O 线。
- 8) 1 个全双工串行口。
- 9) 2 个 16 位定时器 / 计数器。
- 10) 具有 5 个中断源，2 个优先级的中断结构。
- 11) 具有位寻址功能，适用于位（布尔）处理。

MCS-51 系列单片机典型芯片是 8051，其内部结构框图如图 2-1 所示。与 8051 结构基本相同的产品还有 8751 和 8031。8751 是一个用 EPROM 代替 ROM 的 8051；8031 是一个无 ROM 的 8051，它从外部 ROM 取所有指令。今后，除特别说明外，用 8051 这个名称来代表 8051、8751 和 8031。

二、CPU 结构

若去掉图 2-1 中的存储器电路和 I/O 部件，剩下的便是 CPU。它可以分为运算器和控制器两部分。运算器功能部件包括算术逻辑运算部件 ALU、累加器 ACC、寄存器 B、暂存寄存器 TR、程序计数器 PC、程序状态字寄存器 PSW、堆栈指针 SP、数据指针寄存器 DPTR 以及位（布尔）处理器等。控制器功能部件包括指令寄存器、指令译码器、定时控制逻辑电路等。

1. 算术逻辑运算部件 ALU

算术逻辑运算部件 ALU 在定时控制逻辑电路发出的内部控制信号的控制下，可以进行如下算术 / 逻辑操作：

- 1) 带进位和不带进位的加法。
- 2) 带借位减法。
- 3) 8 位无符号数乘法和除法。
- 4) 逻辑与、或、异或操作。
- 5) 加 1、减 1 操作。
- 6) 按位求反操作。
- 7) 循环左、右移位操作。
- 8) 半字节交换。
- 9) 二 / 十进制调整。
- 10) 比较和条件转移的判断等操作。

2. 累加器 ACC

累加器 ACC 是 8 位寄存器，它通过暂存器和 ALU 相连，它是 CPU 中工作最忙、最常用的专用寄存器，许多指令的操作数取自于 ACC，许多运算结果也存放在 ACC 中。在指令系统中，累加器 ACC 的助记符常记作 A。

3. 程序状态字寄存器 PSW

程序状态字寄存器也是一个 8 位寄存器，相当于标志寄存器，用于存放指令执行结果的一些特征，供程序查询和判别之用。其格式如下

	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
PSW	Cy	AC	F0	RS1	RS0	OV	F1	P

其中每一位的具体含义如下：

Cy：进位标志。在进行加（或减）法运算时，如果执行结果最高位 D₇ 有进（或借）位，Cy 置 1；否则 Cy 清 0。在进行位操作时，Cy 又是位操作累加器，指令助记符用 C 表示。

AC：辅助进位。当进行加（或减）法运算时，如果低半字节 D₃ 向高半字节进（或借）位时，AC 置 1；否则 AC 清 0。

F0：用户标志。由用户根据需要对 F0 置位或复位，作为软件标志。

RS1 和 **RS0**：工作寄存器组选择控制。由用户程序改变 RS1 和 RS0 组合中的内容，以选择片内 RAM 中的 4 组工作寄存器之一作为当前的工作寄存器。工作寄存器组的选择见表 2-1。

单片机在复位后，RS1 和 RS0 都为 0。CPU 自然选择工作寄存器组 0 作为当前工作寄存器。根据需要，用户可以利用传送指令或位操作指令来改变 RS1、RS0 的内容，选择其他工作寄存器组，这种设置对程序中保护现场提供了方便。

表 2-1 当前工作寄存器组的选择

RS1 (PSW.4)	RS0 (PSW.3)	当前使用的工作寄存器组 R0~R7
0	0	工作寄存器组 0 (00H~07H)
0	1	工作寄存器组 1 (08H~0FH)
1	0	工作寄存器组 2 (10H~17H)
1	1	工作寄存器组 3 (18H~1FH)

OV：溢出标志。用于补码运算时，当运算结果超出一 128~+127 范围，产生溢出，OV 置 1；否则无溢出，OV 清 0。

P：奇偶标志。该标志位始终跟踪累加器 ACC 中 1 的数目的奇偶性。如果 ACC 中 1 的个数为奇数，则 P 置 1；否则若 A 中 1 的个数为偶数或 A=00H（没有 1），则 P 清 0。无论执行什么指令，只要 A 中 1 的个数改变，P 就随之而变。以后在指令系统中，凡是累加器 ACC 的内容对 P 标志位的影响都不再赘述。

F1：这一位的含义，用户可以利用这一位来建立用户标志，如同 F0 那样，但要用位地址 D1H 或符号 PSW.1 来表示这一位。在汇编语言中，用户可以事先给这一位定义一个名称（如 F1），以后就可以很方便地直接使用所定义的名称来表示 PSW.1。

三、位（布尔）处理器

特别值得一提的是 MCS-51 单片机内部的位处理器。MCS-51 片内的 CPU 还是一个性能优异的位处理器，也就是说 MCS-51 实际上又是一个完整的一位微型计算机。它有自己的 CPU、位寄存器、位累加器（即进标志 Cy）、I/O 接口和指令系统。它们组成了一个完整的、独立的而且功能很强的位处理单片机，这是 MCS-51 系列单片机的突出优点之一。MCS-51 单片机对于位变量操作（布尔处理）有置位、清 0、取反、测试转移、传送、逻辑与和逻辑或运算等等。