

单片机 入门与开发

李平 杜涛 王靖 编著



机械工业出版社
China Machine Press

单片机及嵌入式系统开发的基础是单片机的理论与开发。本书从单片机基本理论出发，以开发实例为重点，特别编写了单片机及嵌入式系统开发的成套教程，以两册出版。本教程以富士通的8位单片机F²MC-8FX系列为主，介绍单片机的理论与实践。

本册《单片机入门与开发》为原理篇。第1~3章介绍单片机的背景知识。第4章介绍单片机的体系结构。第5~9章详细介绍单片机的主要硬件结构和工作原理，同时举出了典型开发实例，便于初学者深刻领会单片机原理，并着手入门开发。第10章介绍单片机C语言编程。第11章介绍单片机开发系统。第12章给出了综合应用开发实例。

本书适合作为高等院校电子信息类专业的教材，也可供嵌入式系统开发人员参考。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

单片机入门与开发/李平,杜涛,王靖编著. —北京:机械工业出版社, 2008.8

ISBN 978-7-111-24726-5

I. 单… II. ①李… ②杜… ③王… III. 单片微型计算机-教材 IV. TP368.1

中国版本图书馆CIP数据核字(2008)第110135号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:曾珊

三河市明辉印装有限公司印刷·新华书店北京发行所发行

2008年8月第1版第1次印刷

184mm×260mm·16.5印张

标准书号:ISBN 978-7-111-24726-5

定价:30.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010) 68326294

序

我欣慰地看到，由有着多年的单片机教学经验的教授执笔，历时两千多个日日夜夜，在汲取了诸多单片机业界专家的宝贵知识及从业经验后，《单片机入门与开发》和《单片机应用开发与实践》两本教材终于问世了。

面对众多的单片机教材，富士通一贯坚持不盲从、厚积薄发的原则，精心推出 8 位单片机系列的基础教材。需要指出的是，本套教材所依据的富士通 8 位灵活微控制器提供了丰富的片上外设资源；结合高性能 CPU 内核，应用广泛。丰富的产品系列可提供各种 Flash/ROM/RAM 型号、通用基本资源和增强型专用资源以及强化功能。

新型代码兼容了 F²MC-8FX 系列并提供改进的指令集，包括增强的乘法/除法指令。片上 LIN 功能适合汽车应用，支持主控/从属运行以及 LIN 同步中断探测。

双操作闪存通过同时执行程序代码和擦/写某一数据闪存区域，支持引导装入程序以及高成本效益的片上 EEPROM 仿真。

所有 8 位 MCU 都由高品质的软件和硬件开发工具提供支持，这使富士通的产品不仅全面而且非常适合那些希望自己的产品从微控制器革命中获益的受众。

我们衷心希望这套教材成为以先进产品为代表的主流教材，成为老师和学生教学相长的良师益友。

富士通微电子亚太集团 市场总监

郑国威

2008 年 3 月

前 言

二十世纪,人类社会跨越了三个“电”的时代,即电气时代、电子时代和电脑时代。大多数人所理解的电脑,通常是指个人计算机,简称PC机。实际上还有另一类计算机,就是把智能赋予各种机械的单片机(亦称微控制器),广泛应用于如汽车电子、智能仪表、实时工控、通信设备、导航系统、家用电器等各个领域。

单片机作为微型计算机的一个重要分支,它不是完成某一个逻辑功能的芯片,而是把一个计算机系统集成到一颗芯片上。它的体积小、质量轻、价格便宜,为学习、应用和开发提供了便利条件。

单片机及嵌入式系统开发的基础是单片机的理论与开发技术。编著者从单片机基本理论出发,以开发实例为重点,特别编写了单片机及嵌入式系统开发的成套教程,以两册出版。

本套教程的编写思路是把原理教学和实践演练独立成册,原理篇定名为《单片机入门与开发》,实践篇定名为《单片机应用开发与实践》。原理篇和实践篇两册相对独立,学练分明,适合于原理课和实践课分设的教学模式。

本教程在写法上讲究循序渐进、深入浅出、通俗易懂、趣味性强,力求将枯燥乏味的学习过程变得轻松自如,引导读者享受学习单片机的乐趣,在学习和实践中更多地体会成功。编著者在本教程编排上,尽量使读者在学习单片机的过程中,以尽可能少的时间和精力来获取尽可能全面的单片机理论知识和实践开发技术。

本套教程期望达到从入门到精通、从理论到技能的渐进式学习效果,避免传统方式的知识灌输。在传授单片机知识的同时,更注重传授开发方法和应用技巧,侧重强调将所学知识转化为实际工作能力,力图将有志于跻身单片机领域的读者培养成既懂单片机知识,又能掌握单片机及嵌入式系统产品开发技术的内行。

本教程以富士通的8位单片机F²MC-8FX系列为主,介绍单片机的理论与实践。F²MC-8FX系列单片机是富士通为满足市场需求,最新推出的具备快速处理和低功耗特性的8位单片机,主要应用于消费类电子、汽车电子及工业领域。F²MC-8FX系列8位单片机采用了0.35 μm低漏电工艺技术,掩膜产品可以在1.8 V和1 μA的低耗电工作模式(监视模式)下运行;同时,采用了流水线总线架构以提供双倍执行速度,最快处理速度则达到了F²MC-8L产品系列的5倍左右。本产品系列在设计中确保了所有的指令(除了和除法有关的指令)都和传统的F²MC-8L系列兼容,开发都在Softune中进行,因此,从F²MC-8L系列向F²MC-8FX系列移植程序非常容易。

本册《单片机入门与开发》为原理篇。第1~3章介绍单片机的背景知识。第4章介绍单片机的体系结构。第5~9章详细介绍单片机的主要硬件结构和工作原理,同时举出了典型开发实例,便于初学者深刻领会单片机原理,并着手入门开发。第10章介绍单片机C语言编程。第11章介绍单片机开发系统。第12章给出了综合应用开发实例。

在本教程的编著过程中,得到了富士通电子元器件亚太集团资深总监陈锦新先生、高级

经理王钰先生、高级经理曹森女士、应用工程师陈培俊先生和张佳佳等的大力支持，除了提供最新资料和实验物品外，还将他们在长期实践中积累的经验体会和开发技巧无私地奉献出来供广大读者分享。电子科技大学张国俊教授、罗玉香副教授为本教程审稿，并提出了诸多宝贵意见。本教程的最终成稿，是丛书编撰小组集体辛勤劳动的结晶。为本书编著工作付出心血的老师和研究生还有罗和平、胡滨、阮爱武、翟亚红、李俊宏、王劲、谢晓东、廖永波、宁辉、伍希、汤岐、曹大勇、李辉、杨秀栋、辜科等，在此一并深表诚挚的谢意！

由于需要撰写的内容不仅量大而且新颖，加之编著者的水平有限，书中不妥之处在所难免，敬请广大读者不吝赐教。

编著者

2008年3月

目 录

序 前言

第1章 微型计算机基础	1	第3章 富士通单片机概述	19
1.1 数制	1	3.1 富士通单片机简介	19
1.1.1 二进制	1	3.2 富士通8位单片机概述	19
1.1.2 二进制数与十进制数之间的转换	1	3.2.1 F ² MC-8L系列概述	19
1.1.3 十六进制	2	3.2.2 F ² MC-8FX系列概述	20
1.1.4 二进制数与十六进制数之间的转换	2	3.3 富士通16位单片机概述	20
1.1.5 二进制数与十六进制数的运算	3	3.3.1 F ² MC-16LX系列概述	20
1.2 码制	4	3.3.2 下一代16FX单片机概述	21
1.2.1 英文字符的表示方法——ASCII码	4	3.4 富士通32位单片机	21
1.2.2 BCD码	5	第4章 单片机体系结构	23
1.2.3 数据在计算机中的表示	6	4.1 单片机总体结构	23
1.3 微型计算机的组成结构	6	4.1.1 单片机的基本组成	23
1.3.1 微机的组成	6	4.1.2 单片机内部结构	24
1.3.2 微机系统的组成	7	4.2 CPU	25
1.3.3 微型计算机的总线结构	7	4.2.1 流水线的内部总线	25
1.3.4 微处理器的内部结构与基本功能	8	4.2.2 加速的时钟控制器	26
1.4 微机系统的工作过程	8	4.2.3 增强的中断	26
第2章 单片机基础	10	4.3 存储空间	26
2.1 单片机及其发展概况	10	4.3.1 存储器的映射	27
2.1.1 单片机及其特点	11	4.3.2 内存空间与寻址方式的对应关系	28
2.1.2 单片机技术现状及将来发展趋势	12	4.4 寻址方式	28
2.2 单片机硬件和软件系统	12	4.5 寄存器	31
2.2.1 单片机的硬件基础	12	4.5.1 CPU专用寄存器	31
2.2.2 单片机的软件基础	13	4.5.2 通用寄存器	33
		4.6 复位及复位电路	33
		4.6.1 复位操作	33
		4.6.2 复位源寄存器	34
		4.7 指令集	35

4.7.1 特殊指令	35	第8章 通信接口	115
4.7.2 位操作指令	36	8.1 通信基础	115
4.7.3 F ² MC-8FX 指令表	36	8.1.1 异步通信	115
第5章 I/O 端口	42	8.1.2 同步通信	116
5.1 Port 0	43	8.2 UART/SIO	117
5.2 Port 1	45	8.3 LIN-UART	122
5.3 Port 2	47	8.4 I ² C	127
5.4 Port 3	49	8.5 通信接口程序示例	133
5.5 Port 4	51	第9章 A/D 转换器	136
5.6 Port 5	53	9.1 A/D 转换器的种类与特点	136
5.7 Port 6	55	9.2 A/D 转换器的工作原理	137
5.8 Port 7	58	9.3 8FX 系列单片机 A/D 转换器	139
5.9 Port 8	59	9.3.1 A/D 转换器特性	140
5.10 Port E	61	9.3.2 A/D 转换器结构	140
5.11 Port G	63	9.3.3 A/D 转换器引脚	141
第6章 中断系统	66	9.3.4 A/D 转换器的寄存器	141
6.1 中断的概念	66	9.3.5 A/D 转换器中断	144
6.2 中断的功能	66	9.3.6 A/D 转换器的操作	144
6.3 中断系统结构及中断控制	67	9.3.7 A/D 转换器使用的注意 事项	145
6.3.1 中断源	67	9.4 A/D 转换器程序示例	145
6.3.2 中断控制	68	第10章 单片机 C 语言编程	149
6.4 中断处理过程	69	10.1 概述	149
6.4.1 中断响应	69	10.1.1 单片机 C 语言的程序结 构和特点	149
6.4.2 中断处理	70	10.1.2 单片机 C 语言与标准 C 语言的区别	151
6.4.3 中断返回	71	10.1.3 数据存储器的分配	152
6.5 多重中断	71	10.1.4 单片机 C 语言与汇编 语言的混合编程	153
6.6 中断程序示例	72	10.2 算法	153
第7章 定时器/计数器	75	10.2.1 算法概述	153
7.1 时基定时器	75	10.2.2 算法的表示	154
7.2 看门狗定时器	77	10.3 C 语言的基本语法	156
7.3 预分频计数器	79	10.3.1 数据类型	156
7.4 监视计数器	82	10.3.2 标示符	159
7.5 8/16 位复合定时器	85	10.3.3 常量	160
7.6 8/16 位可编程脉冲发生器	94	10.3.4 变量	161
7.7 16 位 PPG 定时器	99		
7.8 16 位重载定时器	106		

10.3.5	运算符与表达式	161	10.11.2	七段LED数码管应用 程序	192
10.4	数据类型转换	164	10.11.3	串行EEPROM程序	196
10.5	C语言语句	165	10.11.4	液晶显示程序	205
10.6	选择结构程序设计	167	第11章	单片机开发系统	208
10.6.1	if语句	167	11.1	开发工具	208
10.6.2	if语句的嵌套	168	11.2	开发环境	209
10.6.3	switch语句	169	11.2.1	界面浏览及其功能	209
10.7	循环结构程序设计	169	11.2.2	建立工程实例	216
10.7.1	while循环	170	11.3	仿真器	228
10.7.2	do-while循环	170	11.3.1	仿真器简介	228
10.7.3	for循环	171	11.3.2	后台监视适配器	229
10.7.4	循环嵌套	171	11.3.3	MCU板	229
10.7.5	break语句、continue 语句和goto语句	172	11.3.4	转接板	230
10.8	C语言的函数	173	11.3.5	8FX开发套件的设置	230
10.8.1	模块化程序设计方法	173	11.4	程序烧写	234
10.8.2	函数的调用	174	11.4.1	硬件链接	234
10.8.3	函数间的参数传递	176	11.4.2	烧写软件	234
10.8.4	函数的嵌套与递归调用	177	第12章	单片机综合应用开发实例	237
10.8.5	中断服务函数	178	12.1	步进电机控制系统	237
10.8.6	变量的作用域和存储 方式	178	12.2	语音播放系统	238
10.9	指针	181	附录A	F ² MC-8FX系列单片机I/O 电路类型	245
10.9.1	指针的概念	181	附录B	F ² MC-8FX系列单片机 I/O映射表	247
10.9.2	指针变量的定义与 初始化	182	附录C	F ² MC-8FX系列单片机中 断源列表	252
10.9.3	指针的运算及引用	182	附录D	F ² MC-8FX系列单片机 指令表	253
10.9.4	指针与函数	183			
10.10	编译预处理	185			
10.10.1	宏定义	186			
10.10.2	文件包含	187			
10.10.3	条件编译	188			
10.11	C语言程序设计示例	189			
10.11.1	键盘接口设计	189			

第 1 章 微型计算机基础

1.1 数制

多位数码中每一位的构成方法以及从低位到高位进位的规则称为数制。计算机应用中常用数制有三种：二进制、十进制和十六进制，其中只有二进制是计算机能直接处理的。但是二进制表达过于繁琐，所以引入十六进制数。而十进制是人们最习惯和熟悉的数制。

1.1.1 二进制

在二进制数中，每一位只有 0 或者 1 两个可能的数码，基数为 2，低位与相邻高位之间的关系是逢二进一，所以称为二进制。二进制数常以结尾的 B 示之。任意一个二进制数按权可展开为：

$$\begin{aligned}\text{二进制数} &= \sum_{i=-m}^{n-1} k_i 2^i \\ &= k_{n-1} \times 2^{n-1} + k_{n-2} \times 2^{n-2} + \cdots + k_0 \times 2^0 + k_{-1} \times 2^{-1} \\ &\quad + k_{-2} \times 2^{-2} + \cdots + k_{-m} \times 2^{-m}\end{aligned}\quad (1-1)$$

其中， k_i 表示二进制数的第 i 位，权为 2^i ， k_i 从 0、1 两个数字中选用； m 、 n 为正整数， n 为小数点左边的位数， m 为小数点右边的位数。

例如：

$$\begin{aligned}10101.11\text{B} &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 21.75\text{D}\end{aligned}$$

1.1.2 二进制数与十进制数之间的转换

1. 二—十转换

把二进制数转换为等值的十进制数称为二—十转换。转化时只需将二进制数按式 (1-1) 展开，然后把各项数值按十进制数相加，就可以得到相应等值的十进制数了。例如：

$$\begin{aligned}(1101.01)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= (13.25)_{10}\end{aligned}$$

2. 十—二转换

十进制数转换为二进制数要分整数和小数两部分进行。整数部分的转换采用“除 2 取余”法：除 2 取余，商为 0 止，余数倒置。例如，将 $(123)_{10}$ 化为二进制数可如下进行：

$$\begin{array}{ll}
2 \overline{)123} \cdots \cdots \cdots \text{余数} = 1 = k_0 \\
2 \overline{)61} \cdots \cdots \cdots \text{余数} = 1 = k_1 \\
2 \overline{)30} \cdots \cdots \cdots \text{余数} = 0 = k_2 \\
2 \overline{)15} \cdots \cdots \cdots \text{余数} = 1 = k_3 \\
2 \overline{)7} \cdots \cdots \cdots \text{余数} = 1 = k_4 \\
2 \overline{)3} \cdots \cdots \cdots \text{余数} = 1 = k_5 \\
2 \overline{)1} \cdots \cdots \cdots \text{余数} = 1 = k_6 \\
0
\end{array}$$

所以, $(123)_{10} = (1111011)_2$ 。

小数部分的转换采用“乘2取整”法: 乘2取整, 直到小数部分为0或满足精度要求, 整数正置。例如将 $(0.625)_{10}$ 化为二进制数可如下进行:

$$\begin{array}{ll}
0.625 \times 2 = 1.25 \cdots \cdots \text{整数} = 1 = k_{-1} \\
0.25 \times 2 = 0.50 \cdots \cdots \text{整数} = 0 = k_{-2} \\
0.50 \times 2 = 1.00 \cdots \cdots \text{整数} = 1 = k_{-3}
\end{array}$$

所以, $(0.625)_{10} = (0.101)_2$ 。

综上两例, 得

$$(123.625)_{10} = (1111011.101)_2。$$

1.1.3 十六进制

在十六进制数中, 每一位有0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F十六个可能的数码, 基数为16, 低位与相邻高位之间的关系是逢十六进一, 所以称为十六进制。十六进制数常以结尾的H示之。任意一个十六进制数按权可展开为:

$$\begin{aligned}
\text{十六进制数} &= \sum_{i=-m}^{n-1} k_i 16^i \\
&= k_{n-1} \times 16^{n-1} + k_{n-2} \times 16^{n-2} + \cdots + k_0 \times 16^0 + k_{-1} \times 16^{-1} \\
&\quad + k_{-2} \times 16^{-2} + \cdots + k_{-m} \times 16^{-m} \qquad (1-2)
\end{aligned}$$

其中, k_i 表示十六进制数的第 i 位, 权为 16^i , k_i 从0~9、A、B、C、D、E、F十六个数码中选择; m 、 n 为正整数, n 为小数点左边的位数, m 为小数点右边的位数。例如:

$$\begin{aligned}
(C5D.A6)_{16} &= 11 \times 16^2 + 5 \times 16^1 + 12 \times 16^0 + 10 \times 16^{-1} + 6 \times 16^{-2} \\
&= (3165.6484375)_{10}
\end{aligned}$$

1.1.4 二进制数与十六进制数之间的转换

把二进制数转换为等值的十六进制数称为二—十六转换。由于4位二进制数刚好对应1位十六进制数, 所以二进制转换为十六进制数的方法是: 将要转换的二进制数以小数点为界向左和向右分开, 不足4位补零, 然后将4位二进制数表示为1位十六进制数。例如, 将二进制数 10100110、01111110 转换成十六进制数可以如下进行:

1010,0110.0111,1110

A 6 7 E

所以, $(10100110.01111110)_2 = (A6.7E)_{16}$ 。

同理,十六进制转换为二进制数时,将每1位十六进制转换为对应的4位二进制数即可。

1.1.5 二进制数与十六进制数的运算

在计算机中,二进制数与十六进制数的运算可分为算术运算和逻辑运算。算术运算包括二进制数和十六进制数的加、减、乘、除,与十进制数类似。逻辑运算包括二进制数和十六进制运算,与、或、非、异或。本节主要介绍二进制数的运算。

1. 算术运算

二进制数的算术运算与大家所熟悉的十进制数运算十分相似,唯一的区别就在于运算规则为逢2进1。

$$\begin{array}{r} \text{加法运算} \\ 10110101 \\ + 10001111 \\ \hline \text{进位 } 1 \quad 111111 \\ 01000100 \end{array}$$

$$\begin{array}{r} \text{乘法运算} \\ \begin{array}{r} 1001 \\ \times 0101 \\ \hline 1001 \\ 0000 \\ 1001 \\ 0000 \\ \hline 0101101 \end{array} \end{array}$$

$$\begin{array}{r} \text{减法运算} \\ 01000100 \\ - 10100101 \\ \hline \text{借位 } 1 \quad 111111 \\ 10011111 \end{array}$$

$$\begin{array}{r} \text{除法运算} \\ \begin{array}{r} 110 \\ 110 \overline{) 100110} \\ \underline{110} \\ 111 \\ \underline{110} \\ 10 \end{array} \end{array}$$

2. 逻辑运算

针对十六进制和二进制数的逻辑运算主要有4种:与(AND)、或(OR)、异或(XOR),以及取反(NOT)。

逻辑与、或以及异或运算作用于两个单个位的操作数,运算结果如下:

与:

$$0 \text{ AND } 0 = 0$$

$$0 \text{ AND } 1 = 0$$

$$1 \text{ AND } 0 = 0$$

$$1 \text{ AND } 1 = 1$$

或:

$$0 \text{ OR } 0 = 0$$

$$0 \text{ OR } 1 = 1$$

$$1 \text{ OR } 0 = 1$$

$$1 \text{ OR } 1 = 1$$

异或:

$$0 \text{ XOR } 0 = 0$$

$$0 \text{ XOR } 1 = 1$$

$$1 \text{ XOR } 0 = 1$$

$$1 \text{ XOR } 1 = 0$$

表 1-1, 表 1-2 和表 1-3 是与、或和异或运算的真值表。真值表类似于小学学过的乘法表, 表中最左边一列中的数值对应于运算的左操作数, 最上边一行的数值对应于右操作数, 而在行列交叉处(对应于特定的一对输入)的数值则是结果。

表 1-1 与运算的真值表

AND	0	1
0	0	0
1	0	1

表 1-2 或运算的真值表

OR	0	1
0	0	1
1	1	1

表 1-3 异或运算的真值表

XOR	0	1
0	0	1
1	1	0

通俗地讲, 逻辑与运算可以描述为: “如果第一操作数是 1 而且第二操作数是 1, 结果是 1, 否则结果就是 0”。换一种说法, 可以描述为: “如果任一操作数是 0 或者两个操作数都是 0, 结果就是 0。”逻辑与操作在需要置 0 时比较有用, 只要任一操作数为 0, 不管另一个操作数的值是什么, 结果永远是 0。如果某一个操作数为 1, 那么结果就等于另一个操作数的值。

同样, 逻辑或运算可以描述为: “如果第一操作数或者第二操作数(或者都)是 1, 那么结果就是 1, 否则结果就是 0”。逻辑或又被称为同或(inclusive-OR)运算。如果逻辑或的某一个操作数为 1, 那么结果永远为 1, 如果某一个操作数为 0, 那么结果永远等于另一个操作数的值。

逻辑异或运算同样可以描述为: “如果第一或者第二操作数中的任意一个, 但不是都为 1, 那么结果为 1, 否则结果就是 0”。如果任一个操作数为 1, 那么结果永远是另一个操作数的取反。

逻辑非运算是一元运算(即它只接收一个操作数)。逻辑非的真值表如表 1-4 所示。逻辑非只是简单地将操作数的值取反。

表 1-4 非运算的真值表

NOT	0	1
	1	0

1.2 码制

不同的数码不仅可以表示数量的大小, 而且还能用来表示不同的事物。当用来表示事物时, 这些数码已没有表示数量大小的含义, 只是表示不同事物的代号而已, 这些数码称为代码。码制就是编制代码过程中遵循的一定的规则。

1.2.1 英文字符的表示方法——ASCII 码

ASCII 码是一种字符编码, 是美国信息交换标准代码的简称。它由 7 位二进制数码构成, 共有 128 个字符, 如表 1-5 所示。

ASCII 码主要用于微机与外设通信。当微机与 ASCII 码制的键盘、打印机及 CRT 等连用

时，均以 ASCII 码形式数据传输。例如，当敲打 ASCII 码制的键盘中的某一键，键盘中的单片机便将所敲的键码转换成 ASCII 码传入微机进行相应处理。

表 1-5 ASCII 码字符表(7 位码)

LSD $b_3b_2b_1b_0$		MSD $b_6b_5b_4$							
		0	1	2	3	4	5	6	7
		000	001	010	011	110	101	110	111
0	0000	NUL	DLE	SP	0	@	P	,	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	,	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	
C	1100	FF	FS	,	<	L	\	L	
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	US	/	?	O	←	o	DEL

注：表中二进制代码按顺序 $b_6b_5b_4b_3b_2b_1b_0$ 排列。

1.2.2 BCD 码

BCD 码是一种二进制形式的十进制码，也称二—十进制码。它用 4 位二进制数表示一位十进制数，最常用的是 8421BCD 码，如表 1-6 所示。

表 1-6 8421BCD 码

十进制数	BCD 码	二进制数
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001

8421BCD 码用 0000H ~ 1001H 代表十进制数 0 ~ 9, 运算法则是逢 10 进 1。8421BCD 码每位的权分别是“8”、“4”、“2”、“1”, 故得此名。

例如, 1997 的 BCD 码为 0001 1001 1001 0111。

1.2.3 数据在计算机中的表示

计算机处理的是二进制数。二进制数又分为带符号数和不带符号数两种。

1. 带符号数

以 8 位二进制数为例, 带符号的 8 位二进制数用最高位 D7 表示数的正或负, “0”代表“+”, “1”代表“-”, D7 称为符号位, D6 ~ D0 为数值位。

上述的 8 位带符号二进制数又有三种不同的表达形式, 即原码、反码和补码。在计算机中, 所有带符号数都是以补码形式存放的。

(1) 原码

一个二进制数, 用最高位表示数的符号, 其后各位表示数值本身, 这种表示方法称为原码。原码的表示范围是 -127 ~ +124, 例如,

$X = +1011010B$ $[X]_{\text{原}} = 01011010B$; $X = -1011010B$ $[X]_{\text{原}} = 11011010B$

(2) 反码

正数的反码与原码相同。符号位一定为“0”, 其余位为数值位。

负数的反码符号位为“1”, 数值位将其原码的数值位逐位求反。

反码的表示范围是 -127 ~ +127, 例如,

$X = -1011010B$ $[X]_{\text{原}} = 01011010B$ $[X]_{\text{反}} = 10100101B$

(3) 补码

正数的补码与原码相同。

负数的补码符号位为 1, 数值位将其原码的数值位逐位求反后加 1, 即负数的反码加 1。

补码的表示范围是 -128 ~ +127, 例如,

$X = -1011010B$ $[X]_{\text{补}} = 10100110B$

2. 无符号数

无符号数的 8 位二进制数没有符号位, 从 D7 ~ D0 皆为数值位, 所以无符号二进制数的表示范围是 0 ~ +255。

1.3 微型计算机的组成结构

1.3.1 微机的组成

微型计算机由微处理器 CPU、一定容量的内部存储器(包括 ROM、RAM)、输入/输出接口电路组成。各功能部件之间通过总线有机地连接在一起, 其中微处理器是整个微型计算机的核心部件。

内部存储器, 按照读写方式的不同, 分为 ROM 和 RAM 两种类型。

输入/输出接口电路是外围设备与微型计算机之间的连接电路,在两者之间进行信息交换的过程中,起暂存、缓冲、类型变换及时序匹配的作用。

总线是 CPU 与其他各功能部件之间进行信息传输的通道,按所传送信息的不同类型,总线可以分为数据总线 DB、地址总线 AB 和控制总线 CB 三种类型。

1.3.2 微机系统的组成

微型计算机再加上外设、电源、软件等构成微机系统。微机系统常用的外围设备有显示器、打印机、键盘等;系统软件一般包括操作系统、编译、编辑、汇编软件等。

1.3.3 微型计算机的总线结构

微型计算机是由具有不同功能的一组功能部件组成的,系统中各功能部件的类型和它们之间的相互连接关系称为微型计算机的结构。

微型计算机大多采用总线结构,因为在微型计算机系统中,无论是各部件之间的信息传送,还是处理器内部信息的传送,都是通过总线进行的。

1. 什么是总线

所谓总线,是连接多个功能部件或多个装置的一组公共信号线。根据总线在系统中的不同位置,可以分为内部总线和外部总线。内部总线是 CPU 内部各功能部件和寄存器之间的连线;外部总线是连接系统的总线,即连接 CPU、存储器和 I/O 接口的总线,又称为系统总线。

微型计算机采用了总线结构后,系统中各功能部件之间的相互关系变为各个部件面向总线的单一关系。一个部件只要符合总线标准,就可以连接到采用这种总线标准的系统中,使系统的功能可以很方便地得以发展,微型机中目前主要采用的外部总线标准有:PC 总线,ISA 总线,VESA 总线等。

2. 三总线结构

按所传送信息的不同类型,总线可以分为数据总线、地址总线和控制总线三种类型,通常称微型计算机采用三总线结构。

(1) 地址总线(Address Bus)

地址总线是微型计算机用来传送地址信息的信号线。地址总线的位数决定了 CPU 可以直接寻址的内存空间的大小。因为地址总是从 CPU 发出的,所以地址总线是单向三态总线。“单向”指信息只能沿一个方向传送,“三态”指除了输出高、低电平状态外,还可以处于高阻抗状态(浮空状态)。

(2) 数据总线(Data Bus)

数据总线是 CPU 用来传送数据信息的信号线(双向、三态)。数据总线是双向三态总线,即数据既可以从 CPU 送到其他部件,也可以从其他部件传送给 CPU,数据总线的位数和处理器的位数相对应。

(3) 控制总线(Control Bus)

控制总线是用来传送控制信号的一组总线。这组信号线比较复杂,由它来实现 CPU 对外部功能部件(包括存储器和 I/O 接口)的控制及接收外部传送给 CPU 的状态信号,不同的微处理器采用不同的控制信号。控制总线的信号线,有的为单向,有的为双向或三态,有的为非三态,这取决于具体的信号线。

1.3.4 微处理器的内部结构与基本功能

微处理器的内部结构受大规模集成电路制造工艺的严格约束,芯片的面积不能过大,芯片引脚的数量也不能过多。因此,通用微处理器的内部结构及其同外部电路的连接方式都有比较严格的规定。微处理器外部一般采用上述三总线结构;内部则采用单总线结构,即内部所有单元电路都挂在内部总线上,分时享用。

一个典型的8位微处理器的结构,主要包括以下几个重要部分:累加器,算术逻辑运算单元(ALU),状态标志寄存器,寄存器阵列,指令寄存器,指令译码器和定时及各种控制信号的产生电路。

1. 累加器和算术逻辑运算部件

累加器和算术逻辑运算部件(ALU)主要用来完成数据的算术和逻辑运算。ALU有2个输入端和2个输出端,其中一端接至累加器,接收由累加器送来的一个操作数;另一端通过数据总线接到寄存器阵列,以接收第二个操作数。参加运算的操作数在ALU中进行规定的操作运算,运算结束后,一方面将结果送至累加器,同时将操作结果的特征状态送标志寄存器。累加器是一个特殊的寄存器,它的字长和微处理器的字长相同,累加器具有输入/输出和移位功能,微处理器采用累加器结构可以简化某些逻辑运算。由于所有运算的数据都要通过累加器,故累加器在微处理器中占有很重要的位置。

2. 寄存器阵列

- 1) 通用寄存器组:可由用户灵活支配,用来寄存参与运算的数据或地址信息。
- 2) 地址寄存器:专门用来存放地址信息的寄存器。
- 3) 指令指针IP:它的作用是指明下一条指令在存储器中的地址。每取一个指令字节,IP自动加1,如果程序需要转移或分支,只要把转移地址放入IP即可。
- 4) 变址寄存器SI, DI:变址寄存器的作用是用来存放要修改的地址,也可以用来暂存数据。
- 5) 堆栈指示器SP:用来指示RAM中堆栈栈顶的地址。SP寄存器的内容随着堆栈操作的进行,自动发生变化。

3. 指令寄存器,指令译码器和定时及各种控制信号的产生电路

- 1) 指令寄存器用来存放当前正在执行的指令代码;
- 2) 指令译码器用来对指令代码进行分析、译码,根据指令译码的结果,输出相应的控制信号;
- 3) 时序逻辑电路产生出执行此条命令所需的全部控制信号,如各种操作电位、不同节拍的信号、时序脉冲等。

4. 内部总线和总线缓冲器

内部总线把CPU内各寄存器和ALU连接起来,以实现各单元之间的信息传送。内部总线分为内部数据总线和地址总线,它们分别通过数据缓冲器和地址缓冲器与芯片外的系统总线相连。缓冲器用来暂时存放信息(数据或地址),它具有驱动放大能力。

1.4 微机系统的工作过程

1. 计算机工作过程的实质

计算机之所以能在没有人直接干预的情况下,自动地完成各种信息处理任务,是因为人