

# 单片机 原理与接口技术

朱晓辉 来婷 主编



北京理工大学出版社

BELING INSTITUTE OF TECHNOLOGY PRESS

普通高等院校工科专业单片机系列教材

# 单片机原理与接口技术

主 编 朱晓辉 来 婷

副主编 朱茜颖 张 毅

参 编 李 娜

 **北京理工大学出版社**

BEIJING INSTITUTE OF TECHNOLOGY PRESS

## 内 容 简 介

本书以 8051 单片机的 C 语言为例来学习使用 C 语言进行单片机程序开发,详细介绍了 Keil 软件的使用方法、程序的编写与调试方法及其他相关知识。

本书结合了作者多年教学经验,在单片机的 C 语言课堂教学改革基础上,融入了教学改革成果而编写,它依据学习者的认知规律来编排内容,充分体现了以人为本的指导思想。本书各章均配备了足够数量的习题,供师生使用。

本书语言通俗易懂、实例丰富,有较强的实用性和参考价值,适合高等院校电子、电气、控制及相关专业使用。

版权专有 侵权必究

---

### 图书在版编目(CIP)数据

单片机原理与接口技术 / 朱晓辉, 来婷主编. —北京: 北京理工大学出版社, 2018.8  
(2018.9 重印)

ISBN 978-7-5682-6217-0

I. ①单… II. ①朱… ②来… III. ①单片微型计算机-基础理论-高等学校-教材②单片微型计算机-接口技术-高等学校-教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2018)第 193455 号

---

---

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 /

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 11.25

字 数 / 266 千字

版 次 / 2018 年 8 月第 1 版 2018 年 9 月第 2 次印刷

定 价 / 32.00 元



责任编辑 / 陈莉华

文案编辑 / 陈莉华

责任校对 / 周瑞红

责任印制 / 李志强

---

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

# 前 言

自从 1972 年 Intel 公司推出第一片微处理器以来, 计算机技术遵循着摩尔先生提出的摩尔定律, 以每 18 个月为一个周期微处理器性能提高一倍、价格降低一半的速度快步向前发展。以微处理器为核心的微型计算机在最近几十年中发生了巨大的变化。计算机对整个社会进步的影响有目共睹, 其应用面的迅速拓宽, 对个人与社会等多方面的渗透表明, 计算机技术已不再是深居于高层次科技领域里的宠儿, 它已经深入到社会活动的一切领域之中, 闯进了平常百姓的生活里, 使人们跨入了信息时代、数字时代。单片机以其高可靠性、高性能价格比, 在工业控制系统、数据采集系统、智能化仪器仪表、办公自动化等诸多领域得到极为广泛的应用。随着单片机 C 语言编译器的出现, 那些相对缺乏硬件基础知识的相关设计人员设计单片机的大门也随之打开了。

从学习的角度看, 单片机作为一个完整的数字处理系统, 具备了构成计算机的主要单元部件, 在这个意义上称为单片微机并不过分。通过学习和应用单片机进入了计算机硬件设计之门, 可达到事半功倍的效果。本书中包含了大量 8051 单片机应用的程序代码, 编程实例丰富, 内容覆盖面广。通过这些例子的学习, 读者可以在最短的时间里准确、有效地掌握用 C 语言开发单片机系统的技术。对于单片机的学习, 希望读者在学习本书之前, 自己先熟悉一点相关的电子技术知识, 特别是数字电路基础, 这对学习中碰到的相关概念会有很大的帮助。

全书第 1 章由张毅编写、第 2~4 章由朱晓辉编写、第 5~6 章由来婷编写、第 7 章由李娜编写、第 8~9 章由朱茜颖编写。朱晓辉负责全书的策划、内容安排、文稿编写修改和审定工作。

本书在编写过程中得到了武汉工程大学邮电与信息工程学院的全力支持和帮助。本书还得到了陶智勇教授的大力支持, 陶智勇教授对本书的初稿进行了审阅, 并提出了宝贵意见。陈建虎、李炎隆等参与本书相关图表以及部分章节的核对工作, 在此对他们付出的辛勤工作一并表示衷心的感谢!

由于本书涉及的知识点较多, 尽管在编写中做了许多努力, 但由于时间仓促, 难免有不足和疏漏之处, 欢迎广大读者提出宝贵意见和建议, 以便进一步改进和提高, 使之满足实际教学的需要。

# 目 录

|                                              |    |
|----------------------------------------------|----|
| 第 1 章 单片机概述 .....                            | 1  |
| 1.1 单片机的发展史 .....                            | 1  |
| 1.1.1 单片机的应用 .....                           | 2  |
| 1.1.2 单片机的主要发展趋势 .....                       | 2  |
| 1.1.3 MCS-51 单片机和 8051、8031、89C51 等的关系 ..... | 3  |
| 1.2 8051 单片机基础知识 .....                       | 3  |
| 1.2.1 8051 单片机知识 .....                       | 3  |
| 1.2.2 单片机的存储器 .....                          | 6  |
| 1.2.3 单片机片内资源 .....                          | 12 |
| 第 2 章 C 语言与 8051 .....                       | 18 |
| 2.1 8051 的编程语言 .....                         | 18 |
| 2.1.1 C51 编译器 .....                          | 18 |
| 2.1.2 C51 程序结构 .....                         | 19 |
| 2.1.3 单片机调试应用 .....                          | 20 |
| 2.2 单片机编译软件包 Keil C51 的使用 .....              | 23 |
| 2.2.1 Keil C51 的使用方法 .....                   | 23 |
| 2.2.2 调试步骤 .....                             | 26 |
| 第 3 章 C51 数据与运算 .....                        | 30 |
| 3.1 数据与数据类型 .....                            | 30 |
| 3.1.1 常量和变量 .....                            | 31 |
| 3.1.2 C51 数据的存储类型与 8051 存储器结构 .....          | 31 |
| 3.2 8051 内部资源及其 C51 定义 .....                 | 33 |
| 3.3 运算符与表达式 .....                            | 35 |
| 3.3.1 赋值运算 .....                             | 35 |
| 3.3.2 算术运算符及算术表达式 .....                      | 35 |
| 3.3.3 算术运算的优先级与结合性 .....                     | 36 |
| 3.3.4 数据类型转换运算 .....                         | 36 |
| 3.3.5 关系运算与逻辑运算 .....                        | 36 |
| 3.3.6 位运算 .....                              | 37 |
| 3.3.7 自增减运算及复合运算 .....                       | 38 |

|                               |    |
|-------------------------------|----|
| 第4章 C51 流程控制语句 .....          | 40 |
| 4.1 顺序结构 .....                | 40 |
| 4.2 选择结构 .....                | 42 |
| 4.2.1 if 语句的三种基本形式 .....      | 42 |
| 4.2.2 switch-case 语句 .....    | 45 |
| 4.2.3 break 语句 .....          | 46 |
| 4.3 循环结构 .....                | 47 |
| 4.4 C51 数组 .....              | 52 |
| 4.5 函数 .....                  | 53 |
| 4.6 程序设计 .....                | 55 |
| 第5章 8051 内部资源的 C 编程 .....     | 62 |
| 5.1 中断概述 .....                | 62 |
| 5.1.1 中断相关的概念 .....           | 62 |
| 5.1.2 中断源 .....               | 63 |
| 5.1.3 中断响应 .....              | 65 |
| 5.1.4 中断寄存器组切换 .....          | 67 |
| 5.1.5 中断的编程 .....             | 67 |
| 5.2 定时器/计数器 (T/C) .....       | 69 |
| 5.2.1 定时器/计数器概述 .....         | 69 |
| 5.2.2 定时器/计数器的控制 .....        | 71 |
| 5.2.3 定时器/计数器的工作方式 .....      | 71 |
| 5.2.4 定时器/计数器的初始化 .....       | 72 |
| 5.2.5 定时器/计数器的应用举例 .....      | 72 |
| 5.3 串行口 .....                 | 78 |
| 5.3.1 串口概述 .....              | 78 |
| 5.3.2 8051 单片机的串行接口结构 .....   | 79 |
| 5.3.3 串行口的控制与状态寄存器 .....      | 79 |
| 5.3.4 串行口的工作方式 .....          | 81 |
| 5.3.5 串行口初始化 .....            | 82 |
| 5.3.6 串行口应用编程实例 .....         | 83 |
| 第6章 8051 扩展资源的 C 编程 .....     | 88 |
| 6.1 可编程外围定时器 8253 .....       | 88 |
| 6.2 可编程外围并行接口 8255 .....      | 91 |
| 6.2.1 8255 芯片的内部结构与引脚 .....   | 92 |
| 6.2.2 8255 的命令字和工作方式 .....    | 93 |
| 6.2.3 8255 与 8051 的接口设计 ..... | 95 |
| 第7章 8051 数据采集的 C 编程 .....     | 98 |
| 7.1 8 位 D/A 芯片 DAC0832 .....  | 98 |
| 7.1.1 DAC0832 的结构与引脚 .....    | 98 |

|                                           |     |
|-------------------------------------------|-----|
| 7.1.2 8031 与 DAC0832 接口电路的应用 .....        | 99  |
| 7.2 8 位 A/D 芯片 ADC0809 .....              | 102 |
| 7.2.1 ADC0809 的结构和引脚 .....                | 102 |
| 7.2.2 ADC0809 与 8031 的接口 .....            | 103 |
| <b>第 8 章 8051 单片机与输入/输出外设的 C 编程</b> ..... | 107 |
| 8.1 键盘和数码显示 .....                         | 107 |
| 8.1.1 矩阵式键盘与 8051 的接口 .....               | 107 |
| 8.1.2 七段 LED 显示器与 8051 的接口 .....          | 109 |
| 8.2 字符型 LCD 显示模块 .....                    | 111 |
| 8.2.1 字符型 LCD 的结构和引脚 .....                | 112 |
| 8.2.2 显示板控制器的指令系统 .....                   | 113 |
| 8.3 点阵型 LCD 显示模块 .....                    | 116 |
| 8.3.1 HD61830 的特点和引脚 .....                | 117 |
| 8.3.2 HD61830 指令集 .....                   | 118 |
| 8.3.3 与内藏 HD61830 的液晶模块的接口和编程 .....       | 120 |
| <b>第 9 章 单片机应用实例</b> .....                | 130 |
| 9.1 并行接口和定时中断的应用 .....                    | 130 |
| 9.1.1 用 P0 口显示字符串常量 .....                 | 130 |
| 9.1.2 用 if 语句控制 P0 口 8 位 LED 的流水方向 .....  | 131 |
| 9.1.3 用定时器写的流水灯 .....                     | 132 |
| 9.1.4 用字符型数组控制 P0 口 8 位 LED 流水点亮 .....    | 133 |
| 9.1.5 用定时器 T1 中断控制两个 LED 以不同周期闪烁 .....    | 134 |
| 9.2 键盘的应用 .....                           | 135 |
| 9.2.1 用 8255 的 PA 口与 4×4 键盘相接 .....       | 135 |
| 9.2.2 带键盘设置的秒计时器 .....                    | 138 |
| 9.3 串口的应用 .....                           | 141 |
| 9.3.1 键盘输入串口显示 BCD 码 .....                | 141 |
| 9.3.2 串口从键盘输入并显示 0~F .....                | 144 |
| 9.4 脉宽调制 (PWM) 的应用 .....                  | 146 |
| 9.4.1 PWM 控制电机的方法 .....                   | 146 |
| 9.4.2 步进电机控制 .....                        | 150 |
| 9.5 电动自行车的速度测试系统 .....                    | 156 |
| 9.6 在单片机上用液晶手机实现汉字输入功能 .....              | 159 |
| <b>附录 A C51 中的关键字</b> .....               | 167 |
| <b>附录 B ANSI C 标准关键字</b> .....            | 168 |
| <b>参考文献</b> .....                         | 169 |

# 第 1 章

---

## 单片机概述

一台能够工作的计算机要由这样几个部分构成：**CPU**（进行运算、控制）、**RAM**（数据存储）、**ROM**（程序存储）、输入/输出设备（例如串行口、并行输出口等）。在个人计算机上这些部分被分成若干块芯片，安装在一个被称为主板的印制电路板上。而在单片机中，这些部分，全部被做到一块集成电路芯片中了，所以就称为单片（单芯片）机，而且有一些单片机中除了上述部分外，还集成了其他部分，如 **ADC**、**DAC** 等。

单片机的价格并不高，从几元人民币到几十元人民币，体积也不大，一般用 40 脚封装，当然功能多的一些单片机，其引脚也比较多，如 68 个引脚，功能少的只有 10 多个或 20 多个引脚，有的甚至只有 8 个引脚。

为什么会这样呢？

功能有强弱。打个比方，市场上有的组合音响一套才卖几百块钱，可是有的一台功放机就要卖好几千。另外这种芯片的生产量很大，技术也很成熟，51 系列的单片机已经做了十几年，所以价格就低了。

既然如此，单片机的功能肯定不强，为什么要学它呢？

实际工作中并不是任何需要计算机的场合都要求计算机有很高的性能，一个控制电冰箱温度的计算机难道要用 PⅢ？应用的关键是看是否够用，是否有很好的性能价格比。所以 8051 单片机的生产已经几十年了，依然没有被淘汰，还在不断的发展中。

### 1.1 单片机的发展史

上述简单描述了单片机，那么单片机是如何发展起来的呢，这是因为单片机是在一块硅片上集成了各种部件的微型计算机。随着大规模集成电路技术的发展，可以将中央处理器（**CPU**）、数据存储器（**RAM**）、程序存储器（**ROM**）、定时器/计数器以及输入/输出（**I/O**）接口电路等主要计算机部件，集成在一块电路芯片上。虽然单片机只是一个芯片，但从组成和功能上，都已具有了微机系统的含义。由于单片机能独立执行内部程序，所以又称它为微型控制器（**Microcontroller**）。

单片机自从问世以来，性能在不断地提高和完善，它不仅能够满足很多应用场合的需要，而且具有集成度高、功能强、速度快、体积小、使用方便、性能可靠、价格低廉等特点。因此，在工业控制、智能仪器仪表、数据采集和处理、通信、智能接口、商业营销等领域得到广泛的应用，并且正在逐步取代现有的多片微机应用系统。单片机的潜力越来越被人们所重



视,也更扩大了单片机的应用范围,也进一步促进了单片机技术的发展。单片机的发展史大致可分为以下 3 个阶段。

**第一阶段(1976—1978):**初级单片机微处理阶段。该时期的单片机具有 8 位 CPU,并行 I/O 端口、8 位时序同步计数器,寻址范围为 4 KB,但是没有串行口。

**第二阶段(1978—1982):**高性能单片机微机处理阶段,该时期的单片机具有 I/O 串行端口,有多级中断处理系统,15 位时序同步技术器, RAM、ROM 容量加大,寻址范围可达 64 KB。

**第三阶段(1982 至今):**8 位单片机微处理改良型及 16 位单片机微处理阶段。

### 1.1.1 单片机的应用

由于单片机具有显著的优点,它已成为科技领域的有力工具,人类生活的得力助手。它的应用遍及各个领域,主要表现在以下几个方面。

#### 1. 单片机在智能仪表中的应用

单片机广泛地用于各种仪器仪表,使仪器仪表智能化,并可以提高测量的自动化程度和精度,简化仪器仪表的硬件结构,提高其性能价格比。

#### 2. 单片机在机电一体化中的应用

机电一体化是机械工业发展的方向。机电一体化产品是指集成机械技术、微电子技术、计算机技术于一体,具有智能化特征的机电产品,例如微型单片机控制的数字机床、钻床等。单片机作为产品中的控制器,能充分发挥它的体积小、可靠性高、功能强等优点,可大大提高机器的自动化、智能化程度。

#### 3. 单片机在实时控制中的应用

单片机广泛地用于各种实时控制系统中。例如,在工业测控、航空航天、尖端武器、机器人等各种实时控制系统中,都可以用单片机作为控制器。单片机的实时数据处理能力和控制功能,可使系统保持在最佳工作状态,提高系统的工作效率和产品质量。

#### 4. 单片机在分布式多机系统中的应用

在比较复杂的系统中,常采用分布式多机系统。多机系统一般由若干台功能各异的单片机组成,各自完成特定的任务,它们通过串行通信相互联系、协调工作。单片机在这种系统中往往作为一个终端机,安装在系统的某些节点上,对现场信息进行实时的测量和控制。单片机的高可靠性和强抗干扰能力,使它可以置于恶劣环境的前端工作。

#### 5. 单片机在人类生活中的应用

自从单片机诞生以后,它就步入了人类生活,如洗衣机、电冰箱、电子玩具、收录机等家用电器配上单片机后,提高了智能化程度,增加了功能,备受人们喜爱。单片机将使人类生活更加方便、舒适、丰富多彩。

### 1.1.2 单片机的主要发展趋势

目前,单片机正朝着高性能和多品种方向发展,其发展趋势将是进一步向着 CMOS 化、低功耗、小体积、大容量、高性能、低价格和外围电路内装化等几个方面发展。

#### 1. CMOS 化

近年,由于 CHMOS 技术的进步,大大地促进了单片机的 CMOS 化。CMOS 芯片除了低功耗特性之外,还具有功耗的可控性,使单片机可以工作在功耗精细管理状态。这也是今后

以 80C51 取代 8051 为标准 MCU 芯片的原因。因为单片机芯片多数是采用 CMOS（金属栅氧化物）半导体工艺生产。CMOS 电路的特点是低功耗、高密度、低速度、低价格。采用双极型半导体工艺的 TTL 电路速度快，但功耗和芯片面积较大。随着技术和工艺水平的提高，又出现了 HMOS（高密度、高速度 MOS）、CHMOS 工艺以及 CHMOS 和 HMOS 工艺的结合。目前生产的 CHMOS 电路已达到 LSTTL 的速度，传输延迟时间小于 2 ns，它的综合优势已大于 TTL 电路。因而，在单片机领域，CMOS 电路正在逐渐取代 TTL 电路。

## 2. 低功耗化

单片机的功耗已从 mA 级，降至 1  $\mu$ A 以下；使用电压在 3~5 V 范围内，完全适应电池工作。低功耗化的效应不仅是功耗低，而且带来了产品的高可靠性、高抗干扰能力以及产品的便携化。

## 3. 低电压化

几乎所有的单片机都有 WAIT、STOP 等省电运行方式。允许使用的电压范围越来越宽，一般在 3~5 V 范围内工作。低电压供电的单片机电源下限已可达 1~2 V。目前 0.8 V 供电的单片机已经问世。

## 4. 低噪声与高可靠性

为提高单片机的抗电磁干扰能力，使产品能适应恶劣的工作环境，满足电磁兼容性方面更高标准的要求，各单片机厂家在单片机内部电路中都采用了新的技术措施。若用数字电路完成，所设计的电路相当复杂，大概需要十几片数字集成块，其功能也主要依赖于数字电路的各功能模块的组合来实现，焊接的过程比较复杂，成本也非常高。若用单片机来设计制作完成，由于其功能的实现主要通过软件编程来完成，那么就降低了硬件电路的复杂性，而且其成本也有所降低。

### 1.1.3 MCS-51 单片机和 8051、8031、89C51 等的关系

我们平常老是讲 8051，又有什么 8031，现在又有 89C51，它们之间究竟是什么关系呢？

MCS-51 是指由美国 Intel 公司生产的一系列单片机的总称，这一系列单片机包括了好些品种，如 8031、8051、8751、8032、8052、8752 等，其中 8051 是最早最典型的产品，该系列其他单片机都是在 8051 的基础上进行功能的增、减、改变而来的，所以人们习惯于用 8051 来称呼 MCS-51 系列单片机，而 8031 是前些年在我国最流行的单片机，所以很多场合会看到 8031 的名称。Intel 公司将 MCS-51 的核心技术授权给了很多其他公司，所以有很多公司在做以 8051 为核心的单片机，当然，功能或多或少有些改变，以满足不同的需求，其中 89C51 就是这几年来在我国非常流行的单片机，它是由美国 ATMEL 公司开发生产的。

## 1.2 8051 单片机基础知识

### 1.2.1 8051 单片机知识

#### 一、8051 单片机的特点

8051 单片机是把 CPU、ROM、RAM、I/O 口、定时器/计数器、中断等功能全集成在一块芯片上。8051 的 CPU 为 8 位；其片内有振荡器及时钟电路，有 32 根 I/O 线（即 P0 口、

P1 口、P2 口、P3 口), 对于外部存储器寻址范围, ROM、RAM 各 64 KB, 有 2 个 16 位的定时器/计数器, 5 个中断源, 2 个中断优先级; 全双工串行口; 布尔处理器。

二、MCS-51 系列单片机的性能

MCS-51 系列单片机性能如表 1.1 所示。

表 1.1 MCS-51 系列单片机的性能

| ROM 形式 |          |          | 片内 ROM/B | 片内 RAM/B | 寻址范围/B | I/O 口 |     |     |
|--------|----------|----------|----------|----------|--------|-------|-----|-----|
| 片内 ROM | 片内 EPROM | 外接 EPROM |          |          |        | 计数器   | 并行口 | 串行口 |
| 8051   | 8751     | 8031     | 4 K      | 128      | 2×64 K | 2×16  | 4×8 | 1   |
| 80C51  | 87C51    | 80C31    | 4 K      | 128      | 2×64 K | 2×16  | 4×8 | 1   |
| 8052   | 8752     | 8032     | 8 K      | 256      | 2×64 K | 2×16  | 4×8 | 1   |
| 80C52  | 87C52    | 80C32    | 8 K      | 256      | 2×64 K | 2×16  | 4×8 | 1   |

三、中央处理器

中央处理器 (CPU) 由运算器和控制逻辑构成, 其中包括若干 SFR (特殊功能寄存器)。

以 ALU (算术逻辑单元) 为中心的运算器: ALU 能对数据进行加、减、乘、除等算术运算; “与” “或” “异或” 等逻辑运算以及位操作运算。

状态寄存器的位定义如下:

|    |    |    |     |     |    |    |    |
|----|----|----|-----|-----|----|----|----|
| D7 | D6 | D5 | D4  | D3  | D2 | D1 | D0 |
| CY | AC | F0 | RS1 | RS0 | OV |    | P  |

CY—进位标志。有进位/借位时 CY=1, 否则 CY=0。

AC—半进位标志。当 D3 位向 D4 位产生进位/借位时 AC=1, 否则 AC=0, 常用于十进制调整运算中。

F0—用户可设定的标志位, 可置位/复位, 也可供测试使用。

RS1、RS0—四个通用寄存器组的选择位, 该两位的 4 种组合状态用来选择 0~3 寄存器组。其组合如表 1.2 所示。

表 1.2 RS1、RS0 与工作寄存器组的关系

| RS1 | RS0 | 工作寄存器组       |
|-----|-----|--------------|
| 0   | 0   | 0 组 (00~07H) |
| 0   | 1   | 1 组 (08~0FH) |
| 1   | 0   | 2 组 (10~17H) |
| 1   | 1   | 3 组 (18~1FH) |

OV—溢出标志。当带符号数运算结果超出 -128~+127 范围时 OV=1, 否则 OV=0。当无符号数乘法结果超过 255 时, 或当无符号数除法的除数为 0 时 OV=1, 否则 OV=0。

P—奇偶校验标志。每条指令执行完, 若 A 中 1 的个数为奇数时 P=1, 否则 P=0, 即偶校验方式。

#### 四、控制器、时钟电路和基本时序周期

控制逻辑主要包括定时和控制逻辑、指令寄存器、译码器以及地址指针 DPTR 和程序计数器 PC 等。

##### 1. 8051 的时钟

时钟是时序的基础，8051 片内由一个反相放大器构成振荡器，可以由它产生时钟。时钟可以由两种方式产生：内部方式和外部方式，如图 1.1 所示。

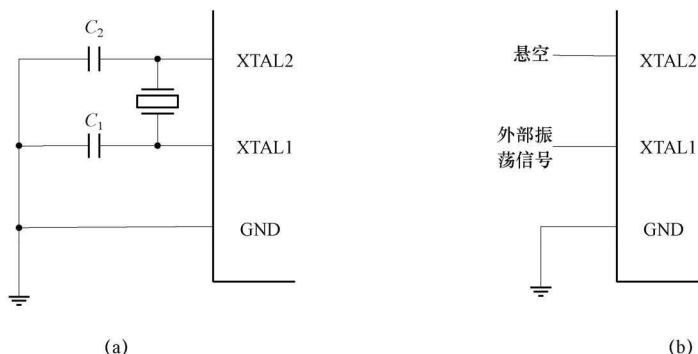


图 1.1 时钟的内部方式和外部方式

(a) 内部方式；(b) 外部方式

##### 2. 8051 的基本时序周期

振荡周期：指振荡源的周期，若为内部产生方式时，为石英晶体的振荡周期。

时钟周期：（称 S 周期）为振荡周期的 2 倍，时钟周期 = 振荡周期  $P_1$  + 振荡周期  $P_2$ 。

机器周期：一个机器周期含 6 个时钟周期。

指令周期：完成一条指令占用的全部时间。8051 的指令周期含 1~4 个机器周期，其中多数为单周期指令，还有 2 周期指令和 4 周期指令。

##### 3. 指令部件

程序计数器 PC：8051 的 PC 是 16 位的计数器，其内容为下一条待执行指令的地址，可寻址范围为 64 KB。

指令寄存器 IR：IR 用来存放当前正在执行的指令。

指令译码器 ID：ID 对 IR 中的指令操作码进行分析解释，产生相应的控制信号。

数据指针 DPTR：DPTR 是 16 位地址寄存器，既可以用于寻址外部数据存储器，也可以寻址外部程序存储器中的表格数据。DPTR 也可以寻址 64 KB 地址空间。

##### 五、复位电路

单片机复位电路如图 1.2 所示。

在振荡器运行时，有两个机器周期（24 个振荡周期）以上的高电平出现在复位引脚 RESET 时，将使单片机复位，只要这个脚保持高电平，51 芯片便循环复位。复位后 P0~P3 口均置“1”，引脚表现为高电平，程序计数器和特殊功

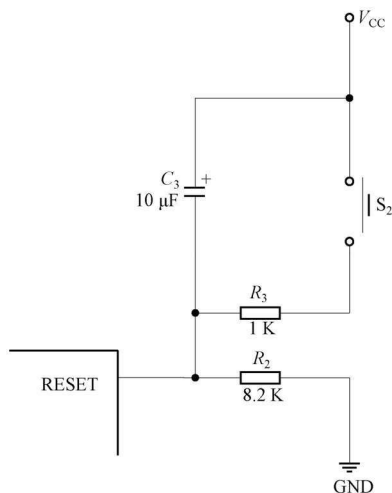


图 1.2 单片机复位电路

能寄存器 SFR 全部清零。当复位脚由高电平变为低电平时，芯片在 ROM 的 00H 处开始运行程序。复位操作不会对内部 RAM 有所影响。

## 1.2.2 单片机的存储器

8051 单片机的存储器的特点：采用哈佛结构，程序存储器与数据存储器分开，两者各有一个相互独立的 64 KB (0x0000~0xFFFF) 的寻址空间（准确地说，内部数据存储器与外部数据存储器不是一回事）。

其存储器结构如图 1.3 所示。

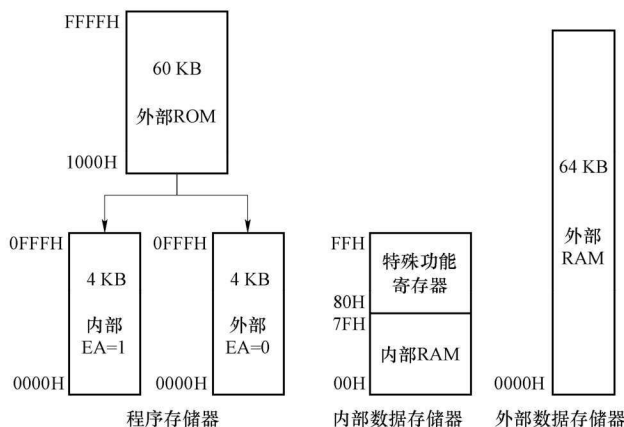


图 1.3 单片机的存储器结构

### 一、程序存储器

一个微处理器能够聪明地执行某种任务，除了它们强大的硬件外，还需要它们运行的软件，其实微处理器并不聪明，它们只是完全按照设计人员预先编写的程序而执行之。设计人员编写的程序就存放在微处理器的程序存储器中，俗称只读程序存储器 (ROM)。程序相当于给微处理器处理问题的一系列命令。其实程序和数据一样，都是由机器码组成的代码串，只是将程序代码存放于程序存储器中。

8051 单片机具有 64 KB 程序存储器寻址空间，它用于存放用户程序、数据和表格等信息。对于内部无 ROM 的 8031 单片机，它的程序存储器必须外接，空间地址为 64 KB，此时单片机  $\overline{\text{EA}}/\text{VPP}$  引脚必须接地，强制 CPU 从外部程序存储器读取程序。对于内部有 ROM 的 8051 等单片机，正常运行时，则需  $\overline{\text{EA}}/\text{VPP}$  引脚接高电平，使 CPU 先从内部的程序存储中读取程序，当 PC 值超过内部 ROM 的容量时，才会转向外部的程序存储器读取程序。

8051 片内有 4 KB 的程序存储单元，其地址为 0000H~0FFFH，单片机启动复位后，程序计数器的内容为 0000H，所以系统将从 0000H 单元开始执行程序。但在程序存储器中有些特殊的单元，在使用中应加以注意。

其中一组特殊单元是 0000H~0002H，系统复位后，PC 为 0000H，单片机从 0000H 单元开始执行程序，如果程序不是从 0000H 单元开始，则应在这三个单元中存放一条无条件转移指令，让 CPU 直接去执行用户指定的程序。

另一组特殊单元是 0003H~002AH，这 40 个单元各有用途，它们被均匀地分为 5 段，其定义如下：

0003H~000AH: 外部中断 0 中断地址区。

000BH~0012H: 定时器/计数器 0 中断地址区。

0013H~001AH: 外部中断 1 中断地址区。

001BH~0022H: 定时器/计数器 1 中断地址区。

0023H~002AH: 串行中断地址区。

可见以上的 40 个单元是专门用于存放中断处理程序的地址单元, 中断响应后, 按中断的类型, 自动转到各自的中断区去执行程序。因此以上地址单元不能用于存放程序的其他内容, 只能存放中断服务程序。但是通常情况下, 每段只有 8 个地址单元, 是不能存下完整的中断服务程序的, 因而一般也在中断响应的地址区安放一条无条件转移指令, 指向程序存储器的其他真正存放中断服务程序的空间去执行, 这样中断响应后, CPU 读到这条转移指令, 便转向其他地方去继续执行中断服务程序。

## 二、数据存储器

数据存储器也称为随机存取数据存储器。8051 单片机的数据存储器在物理上和逻辑上都分为两个地址空间, 一个是内部数据存储器, 另一个是外部数据存储器。8051 单片机内部 RAM 有 128 B 或 256 B 的用户数据存储器 (不同的型号也有区别), 它们是用于存放执行的中间结果和过程数据的。8051 单片机的数据存储器均可读写, 部分单元还可以位寻址。

### 1. 8051 内部数据存储器

8051 内部 RAM 共有 256 个单元, 这 256 个单元共分为两部分。其一是地址从 00H~7FH 单元 (共 128 B) 为用户数据 RAM (内部 RAM)。从 80H~FFH 地址单元 (也是 128 B) 为特殊功能寄存器 (SFR) 单元。从表 1.3 中可清楚地看出它们的结构分布。

表 1.3 内部 RAM 存储器结构

| 内部 RAM 地址 |                | 功 能                                   |
|-----------|----------------|---------------------------------------|
| 00H       | 0 区            | 4 组通用寄存器 R0~R7 也可作 RAM 使用, R0、R1 可位寻址 |
| 08H       | 1 区            |                                       |
| 10H       | 2 区            |                                       |
| 1FH       | 3 区            |                                       |
| 20H       | 位寻址区 00H~7FH   | 全部可位寻址, 共 16 个字节、128 位                |
| 2FH       |                |                                       |
| 30H       | 数据缓冲区、堆栈区、工作单元 | 只能字节寻址                                |
| 7FH       |                |                                       |
| 80H       | 特殊功能寄存器区 (SFR) | 可字节寻址, 也可位寻址                          |
| FFH       |                |                                       |

00H~1FH 共 32 个单元被均匀地分为四块, 每块包含八个 8 位寄存器, 均以 R0~R7 来命名, 常称这些寄存器为通用寄存器。这四块中的寄存器都称为 R0~R7, 那么在程序中怎么区分和使用它们呢? 聪明的 Intel 工程师们又安排了一个寄存器——程序状态字寄存器 (PSW) 来管理它们, CPU 只要定义这个寄存器 PSW 的第 3 和第 4 位 (RS0 和 RS1), 即可选中这四组通用寄存器。对应的编码关系如表 1.4 所示。

表 1.4 通用寄存器区

| RS1 | RS0 | R0  | R1  | R2  | R3  | R4  | R5  | R6  | R7  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 0   | 00H | 01H | 02H | 03H | 04H | 05H | 06H | 07H |
| 0   | 1   | 08H | 09H | 0AH | 0BH | 0CH | 0DH | 0EH | 0FH |
| 1   | 0   | 10H | 11H | 12H | 13H | 14H | 15H | 16H | 17H |
| 1   | 1   | 18H | 19H | 1AH | 1BH | 1CH | 1DH | 1EH | 1FH |

内部 RAM 的 20H~2FH 单元为位寻址区，既可作为一般单元用字节寻址，也可对它们的位进行寻址。位寻址区共有 16 个字节，128 个位，位地址为 00H~7FH。位地址分配如表 1.5 所示，CPU 能直接寻址这些位，执行如置“1”、清“0”、求“反”、转移、传送和逻辑等操作。我们常称 8051 单片机具有布尔处理功能，布尔处理的存储空间指的就是这些位寻址区，见表 1.5。

表 1.5 RAM 位寻址区地址表

| 单元地址 | MSB 位地址 LSB |     |     |     |     |     |     |     |
|------|-------------|-----|-----|-----|-----|-----|-----|-----|
| 2FH  | 7FH         | 7EH | 7DH | 7CH | 7BH | 7AH | 79H | 78H |
| 2EH  | 77H         | 76H | 75H | 74H | 73H | 72H | 71H | 70H |
| 2DH  | 6FH         | 6EH | 6DH | 6CH | 6BH | 6AH | 69H | 68H |
| 2CH  | 67H         | 66H | 65H | 64H | 63H | 62H | 61H | 60H |
| 2BH  | 5FH         | 5EH | 5DH | 5CH | 5BH | 5AH | 59H | 58H |
| 2AH  | 57H         | 56H | 55H | 54H | 53H | 52H | 51H | 50H |
| 29H  | 4FH         | 4EH | 4DH | 4CH | 4BH | 4AH | 49H | 48H |
| 28H  | 47H         | 46H | 45H | 44H | 43H | 42H | 41H | 40H |
| 27H  | 3FH         | 3EH | 3DH | 3CH | 3BH | 3AH | 39H | 38H |
| 26H  | 37H         | 36H | 35H | 34H | 33H | 32H | 31H | 30H |
| 25H  | 2FH         | 2EH | 2DH | 2CH | 2BH | 2AH | 29H | 28H |
| 24H  | 27H         | 26H | 25H | 24H | 23H | 22H | 21H | 20H |
| 23H  | 1FH         | 1EH | 1DH | 1CH | 1BH | 1AH | 19H | 18H |
| 22H  | 17H         | 16H | 15H | 14H | 13H | 12H | 11H | 10H |
| 21H  | 0FH         | 0EH | 0DH | 0CH | 0BH | 0AH | 09H | 08H |
| 20H  | 07H         | 06H | 05H | 04H | 03H | 02H | 01H | 00H |

## 2. 特殊功能寄存器

特殊功能寄存器（SFR）也称为专用寄存器，特殊功能寄存器反映了 8051 单片机的运行状态。很多功能也通过特殊功能寄存器来定义和控制程序的执行。

8051 单片机有 21 个特殊功能寄存器，它们被离散地分布在内部 RAM 的 80H~FFH 地址中，这些寄存器的功能已做了专门的规定，用户不能修改其结构。表 1.6 是特殊功能寄存器分布一览表。下面对其主要的寄存器做一些简单的介绍。

表 1.6 特殊功能寄存器分布

| 标识符号 | 地址      | 寄存器名称         |
|------|---------|---------------|
| ACC  | E0H     | 累加器           |
| B    | F0H     | B 寄存器         |
| PSW  | D0H     | 程序状态字寄存器      |
| DPTR | 82H、83H | 数据指针（16 位）    |
| SP   | 81H     | 堆栈指针          |
| IE   | A8H     | 中断允许控制寄存器     |
| IP   | B8H     | 中断优先控制寄存器     |
| P0   | 80H     | I/O 口 0 寄存器   |
| P1   | 90H     | I/O 口 1 寄存器   |
| P2   | A0H     | I/O 口 2 寄存器   |
| P3   | B0H     | I/O 口 3 寄存器   |
| PCON | 87H     | 电源控制及波特率选择寄存器 |
| SCON | 98H     | 串行口控制寄存器      |
| SBUF | 99H     | 串行数据缓冲器       |
| TCON | 88H     | 定时控制寄存器       |
| TMOD | 89H     | 定时器方式选择寄存器    |
| TH0  | 8CH     | 定时器 0 的高 8 位  |
| TL0  | 8AH     | 定时器 0 的低 8 位  |
| TH1  | 8DH     | 定时器 1 的高 8 位  |
| TL1  | 8BH     | 定时器 1 的低 8 位  |

### 1) 程序计数器 (PC)

程序计数器在物理上是独立的，它不属于特殊内部数据存储器块中。PC 是一个 16 位的计数器，用于存放一条要执行的指令地址，寻址范围为 64 KB，PC 有自动加 1 功能，即完成了一条指令的执行后，其内容自动加 1。PC 本身并没有地址，因而不可寻址，用户无法对它进行读写，但是可以通过转移、调用、返回等指令改变其内容，以控制程序按要求去执行。

### 2) 累加器 (ACC)

累加器 A 是一个最常用的专用寄存器，大部分单操作指令的一个操作数取自累加器，很多双操作数指令中的一个操作数也取自累加器。加、减、乘、除运算的指令，运算结果都存放于累加器 A 和寄存器 B 中。大部分的数据操作都会通过累加器 A 进行，它形象于一个交通要道，在程序比较复杂的运算中，累加器成了制约软件效率的“瓶颈”，它的功能较多，地位也十分重要。以至于后来发展的单片机，有的集成了多累加器结构，或者使用寄存器阵列来代替累加器，即赋予更多寄存器以累加器的功能，目的是解决累加器的“交通堵塞”问题。提高单片机的软件效率。

### 3) 寄存器 B

在乘法指令中，乘法指令中的两个操作数分别取自累加器 A 和寄存器 B，其结果存放



于累加器 A 和寄存器 B 中。除法指令中,被除数取自累加器 A,除数取自寄存器 B,结果商存放于累加器 A,余数存放于寄存器 B 中。

#### 4) 程序状态字 (PSW)

程序状态字是一个 8 位寄存器,用于存放程序运行的状态信息,这个寄存器的一些位可由软件设置,有些位则由硬件运行时自动设置。

#### 5) 数据指针 (DPTR)

数据指针为 16 位寄存器,编程时,既可以按 16 位寄存器来使用,也可以按两个 8 位寄存器来使用,即高位字节寄存器 DPH 和低位字节寄存器 DPL。

DPTR 主要用来保存 16 位地址,当对 64 KB 外部数据存储器寻址时,可作为间址寄存器使用,此时,使用如下两条指令:

```
MOVX A,@DPTR
MOVX @DPTR,A
```

在访问程序存储器时,DPTR 可用来作基址寄存器,采用基址+变址寻址方式访问程序存储器,这条指令常用于读取程序存储器内的表格数据。

```
MOVC A,@A + DPTR
```

#### 6) 堆栈指针 (SP)

堆栈是一种数据结构,它是一个 8 位寄存器,它指示堆栈顶部在内部 RAM 中的位置。系统复位后,SP 的初始值为 07H,使得堆栈实际上是从 08H 开始的。但从 RAM 的结构分布中可知,08H~1FH 隶属 1~3 工作寄存器区,若编程时需要用到这些数据单元,必须对堆栈指针 SP 进行初始化,原则上设在任何一个区域均可,但一般设在 1FH~30H 范围较为适宜。

数据被写入堆栈称为入栈 (PUSH,有些文献也称作插入运算或压入),从堆栈中取出数据称为出栈 (POP,也称为删除运算或弹出),堆栈的最主要特征是“后进先出”规则,也即最先入栈的数据放在堆栈的最底部,而最后入栈的数据放在栈的顶部,因此,最后入栈的数据出栈时则是最先的。这和往一个箱里存放书本一样,如果需将最先放入箱底部的书取出,必须先取走最上层的书籍,非常相似这个道理。

那么堆栈有何用途呢?堆栈的设立是为了中断操作和子程序的调用而用于保存数据,即常说的断点保护和现场保护。微处理器无论是在转入子程序还是中断服务程序的执行,执行完后,都要回到主程序中来,在转入子程序和中断服务程序前,必须先将现场的数据保存起来,否则返回时,CPU 并不知道原来的程序执行到哪一步,原来的中间结果如何。所以在转入执行其他子程序前,先将需要保存的数据压入堆栈中保存,以备返回时,再复原当时的数据,供主程序继续执行。

转入中断服务程序或子程序时,需要保存的数据可能有若干个,都需要一一地保留。如果微处理器进行多重子程序或中断服务程序嵌套,那么需保存的数据就更多,这要求堆栈还需要有相当的容量。否则会造成堆栈溢出,丢失应备份的数据。轻则使运算和执行结果错误,重则使整个程序紊乱。

8051 单片机的堆栈是在 RAM 中开辟的,即堆栈要占据一定的 RAM 存储单元。同时 8051 单片机的堆栈可以由用户设置,SP 的初始值不同,堆栈的位置则可以设置为不同,不同的设计人员,使用的堆栈区则不同;不同的应用要求,堆栈要求的容量也有所不同。堆栈的操作