



# 51 单片机实践教程

吴银琴 陈 锴 编著

## 版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

### 内 容 简 介

本书以 MCS-51 系列单片机为背景,重点突出单片机系统的开发应用,可作为在已经掌握 MCS-51 系列单片机的基本原理与结构的基础上,进行应用开发的高级实践教程。本书通过对 MCU51DB 实验箱的电路分析,阐述了单片机系统的硬件设计基本方法;介绍了如何使用 C51 高级语言编程,并在单片机开发平台软件  $\mu$ Vision2 上进行编译、调试和仿真。读者可以通过对 10 个基本实验和 10 个综合性实验的学习,真正掌握单片机系统的软硬件系统开发过程和方法。另外综合性实验中还涉及单片机系统中常用的各种外围模块的设计与使用,可帮助读者积累单片机系统的开发经验。

本书既可作为 MCS-51 单片机原理及应用的配套实验教材,也可作为希望掌握单片机系统开发技术的大学生的自学教材,还可供从事单片机应用开发的工程技术人员自学与参考。

#### 图书在版编目(CIP)数据

51 单片机实践教程/吴银琴,陈锬编著. —北京:科学出版社,2011.2

ISBN 978-7-03-030085-0

I. 5… II. ①吴… ②陈… III. 单片微型计算机—高等学校—教材  
IV. TP368.1

中国版本图书馆 CIP 数据核字(2011)第 012550 号

责任编辑:高 嵘/责任校对:王望容

责任印制:彭 超/封面设计:苏 波

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉市新华印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

\*

2011 年 2 月第 一 版 开本:787×1092 1/16

2011 年 2 月第一次印刷 印张:13 1/2

印数:1—3 000 字数:262 000

定价:24.50 元

(如有印装质量问题,我社负责调换)

# 前 言

单片微型计算机(micro controller unit, MCU, 简称单片机)作为计算机微型化发展方向的一个分支,从 20 世纪 70 年代至今,得到了快速的发展和广泛的应用,如今单片机的出货量和市场份额都超过了计算机的其他发展分支,形成了庞大的单片机家族和单片机市场。单片机的发展历程反映了计算机技术对电子和机电一体化系统的渗透与影响,这是一种深远并具有革命性意义的影响。计算机软硬件技术被引入到电子系统的设计、生产和运行的所有环节中,彻底颠覆和改变了电子系统的设计原理、方法、方案和结构,也改变了电子系统的设计、生产和使用过程。微控制器正是计算机技术向电子系统进军的主力部队,它作为电子系统的核心控制器,相比于其他计算机系统,在应用中集中表现了集成度高、体积小、抗干扰能力强、实时性好、功耗低、使用方便、性能可靠、价格低廉、开发周期短、开发费用低等优势,伴随着电子系统在现代工业自动化、信息化进程中的广泛使用而得以壮大。微控制器自身也不断发展,其运算能力、性能价格比、性能功耗比不断提高,各类系列产品日益丰富,被大量应用到工业控制、智能仪器仪表、数据采集与处理、通信系统、网络系统、家用电器、手持电子设备等各个领域。

以单片机为载体的计算机软硬件技术在电子系统中的应用如此广泛,以至于单片机技术已经成为现代电子工程师必须掌握的技能和技术手段。单片机作为核心控制器被“嵌入”到其他的系统中从而构成嵌入式系统,而这种嵌入式系统技术的特征就是将各种技术手段综合应用,形成以应用为中心、电子技术和计算机技术为基础的应用技术。完成嵌入式系统的设计不仅需要熟悉单片机的应用,还要了解被单片机“嵌入”系统的工作原理,并进行整体论证和设计。单片机系统的设计一般都是硬件、软件综合设计,这要求我们除了需要掌握模拟和数字电子技术等硬件技术外,还必须掌握数据结构、软件编程、算法设计等软件技术,若要完成嵌入式系统的整体设计,还必须掌握应用对象的数学建模、控制方法、信号处理算法等理论方法,这些都要求我们不仅需要具有扎实的理论基础,还要能够得心应手地对其进行综合应用。不过令人庆幸的是,单片机技术的入门是非常容易的。在笔者从事的单片机教学过程中几乎所有的学生都能够快速入门,能够快速地熟悉单片机的开发过程,编写出驱动单片机引脚的程序,输出几个开关信号,点亮几个发光二极管,这些都是令教授者和学习者轻松和愉悦的事情,但是要想真正成为一名优秀的电子系统工程师绝非易事。如果初学者已经掌握了单片机的基本开发方法,需要进阶,那么也许就需要考虑一下单片机中如何实现“多任务”? 如何管理多个任务之间的优先响应级别? 如何保证紧急事件的响应时间? 如何提高 CPU 的运行效率? 如何尝试一下还未用过的通信接口,如 USB、局域网? 如何尝试一下用单片机在 LCD 显示屏上显示简单的动画? 能否用单片机实现一个简单的 web server? 如何将多个单片机组成网络? 如何用单片机控制机器人完成随轨迹线移动、投球、踢球呢? 要想成为优秀的设计人员,就需要对诸如此类的问题进行深入研究、认真思考、不断实践,只有不断修炼、勇于探索、不惜汗水的人才能成为真正的“高手”,才能在竞争激励的人才市场中得到尊重和认可,才能立于不

败之地。

在单片机技术的学习过程中,请大家记住两大要诀,即多实践和多阅读。理论结合实践才是单片机技术的精要,如果只是能读懂书,看会书,那是毫无意义的。单片机的嵌入式应用决定了单片机技术的范围已远远超越微控制器本身,往往考虑更多的是单片机的外围电路和外部系统的设计,这需要进行大量的阅读来弥补我们知识面的狭窄,并快速地吸收他人智慧从而得到更优的设计方案。本书从最常用的 8 位单片机系列之一的 MCS-51 系列单片机起步,以期打开单片机技术这扇门。

在学习单片机的过程中,离开了实际的单片机系统,任何实践都无从谈起,要学习单片机技术的基本条件是必须拥有一套能够进行单片机开发的单片机实验系统。虽然现在有一些软件支持对单片机的仿真,可以在 PC 机上仿真学习开发单片机系统,但经验告诉我们,在真实的系统上开发获得的感受和记忆才能更深地刻蚀在我们的大脑中,成为我们最可信的智力财富。为了方便学生学习和实践 51 系列单片机,笔者专门设计了以 STC89C58RD+ 单片机为核心的 MCU51DB 单片机实验教学仪器,该仪器扩展了常用的单片机外围器件,配套使用安装在 PC 机上的单片机集成开发软件、下载程序和程序下载线,能够完成单片机系统的开发和调试。MCU51DB 单片机实验教学仪器既可作为 MCS-51 系列单片微型计算机技术课堂教学的配套实验教学仪器,也可以作为各种课程设计的实验箱平台。

编著者  
2010 年 10 月

# 目 录

<b>第 1 章 STC89C58RD+单片机</b> .....	1
1.1 主要性能 .....	1
1.2 1280 字节片内 RAM 数据存储器 .....	3
1.3 32 K 字节片内 flash 程序存储器 .....	4
1.4 IAP 技术 .....	5
1.5 双数据指针的使用 .....	8
1.6 P4 口及中断 .....	9
1.7 看门狗的应用 .....	11
1.8 复位电路 .....	13
1.9 降低电磁辐射 EMI 的设计 .....	15
<b>第 2 章 MCU51DB 实验箱硬件说明</b> .....	17
2.1 MCU51DB 实验箱概述 .....	17
2.2 实验箱硬件电路设计 .....	20
<b>第 3 章 开发环境简介</b> .....	32
3.1 Keil 软件 .....	32
3.2 Keil C51 的编译环境 $\mu$ Vision2 .....	43
3.3 创建项目 .....	52
3.4 开发工具选项 .....	55
3.5 代码分块 .....	59
3.6 上电初始化程序 STARTUP.A51 .....	60
<b>第 4 章 STC 下载软件</b> .....	69
4.1 USB 驱动程序的安装 .....	69
4.2 STC_ISP 下载软件使用 .....	72
4.3 STC_ISP 的串口调试助手 .....	75
<b>第 5 章 Keil C51 与标准 C</b> .....	77
5.1 数据类型 .....	77
5.2 C51 的关键字及扩展关键字 .....	80
5.3 存储类型 .....	82
5.4 使用 Keil C 时的注意事项 .....	85
5.5 Keil C51 的配置文件 .....	86
5.6 Keil C51 库函数原型 .....	94
5.7 C51 与汇编的接口 .....	106
5.8 C51 的代码优化 .....	110

<b>第 6 章 基本实验</b> .....	114
6.1 存储器实验 .....	115
6.2 二进制数与 BCD 的转换.....	116
6.3 P1 口实验 .....	119
6.4 8155 扩展实验 .....	121
6.5 矩阵键盘的使用 .....	124
6.6 外部中断使用 .....	128
6.7 定时器/计数器的实验.....	129
6.8 A/D 实验 .....	130
6.9 D/A 实验 .....	133
6.10 双机通信实验.....	134
<b>第 7 章 综合实验</b> .....	138
7.1 交通灯控制 .....	138
7.2 键盘及 LED 扩展专用芯片 CH451 的使用 .....	142
7.3 看门狗的使用 .....	152
7.4 电子琴的制作 .....	153
7.5 字符型 LCD1602A 使用 .....	156
7.6 点阵型 LCD 使用.....	163
7.7 单总线——温度采集系统 .....	171
7.8 SPI 总线——日历系统 .....	182
7.9 串行 EEPROM 的使用 .....	190
7.10 LED 点阵显示 .....	197
<b>参考文献</b> .....	205
<b>附录 实验箱常用接口函数</b> .....	206

因为单片机的嵌入式应用特点,为了适应不同的应用要求,众多的芯片厂家向市场推出了众多的单片机系列和丰富的产品型号,不同系列和型号的单片机具有不同的技术特点、应用领域和推广营销方式,使得单片机产品市场表现出百家齐放、百家争鸣的特征。在单片机家族的众多成员中,MCS-51 系列单片机以其悠久的历史,成熟的技术支持,大量的生产厂家和兼容型号,高性能价格比,仍然为单片机应用领域中实用最广泛的一种单片机型。同时,全球各单片机生成厂商在 MCS-51 的内核基础上,派生出了大量的 51 内核系列单片机,以增强 51 单片机的各种功能。

STC89 系列单片机就是宏晶科技生产的兼容 MCS-51 的派生产品,它们在指令系统、硬件结构和片内资源上与标准 8051 单片机完全兼容,同时增加了大量新功能。本章详细介绍了 STC89 系列中的 STC89C58RD + 单片机与传统 80C51 的不同之处、STC89C58RD + 单片机的特色之处及使用时需要特别注意的地方,这样可以使已经具备了 8051 单片机基本知识的读者更快地学习和了解 STC89C58RD + 单片机,更好地理解后面的实验内容。

### 1.1 主要性能

STC89C58RD + 单片机在指令系统、硬件结构和片内资源上与标准 8052 单片机完全兼容,其主要特性有:80C51 核心处理器单元;5 V 工作电压,最高时钟频率 0~80 MHz;256+1024 字节的片内 RAM 数据存储器和 32 K 字节片内 flash 程序存储器,在应用可编程(IAP)、系统可编程(ISP)可实现远程软件升级,无需编程器;8 K 字节的 EEPROM;支持 12 时钟(默认)或 6 时钟模式;双 DPTR 数据指针;增强型 UART;32+4 位 I/O 口;3 个 16 位定时器/计数器,8 个中断源,4 个优先级;可编程看门狗定时器(WDT);低 EMI 方式(ALE 禁止);兼容 TTL 和 CMOS 逻辑电平。

STC89C58RD + 单片机的 DIP-40 封装系列与 8051 的完全兼容,PLCC、PQFP 小型封装为 3.3 V 工作电压,支持 ISP/IAP,无需编程器,可灵活应用于各种控制领域。各种封装分别如图 1.1.1、图 1.1.2 和图 1.1.3 所示。

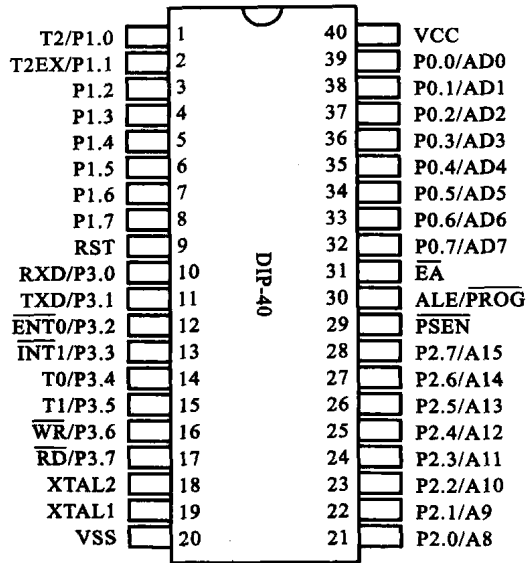


图 1.1.1 STC89C58RD+单片机的 DIP-40 封装

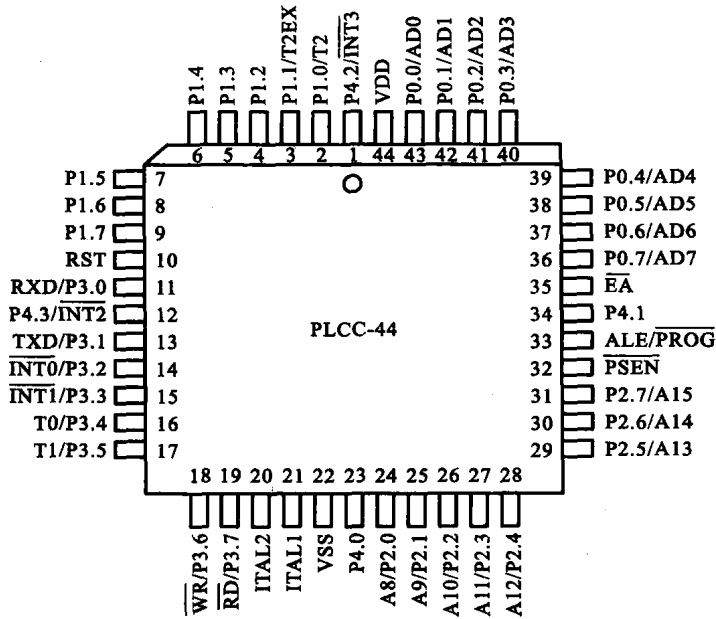


图 1.1.2 STC89C58RD+单片机的 PLCC-44 封装



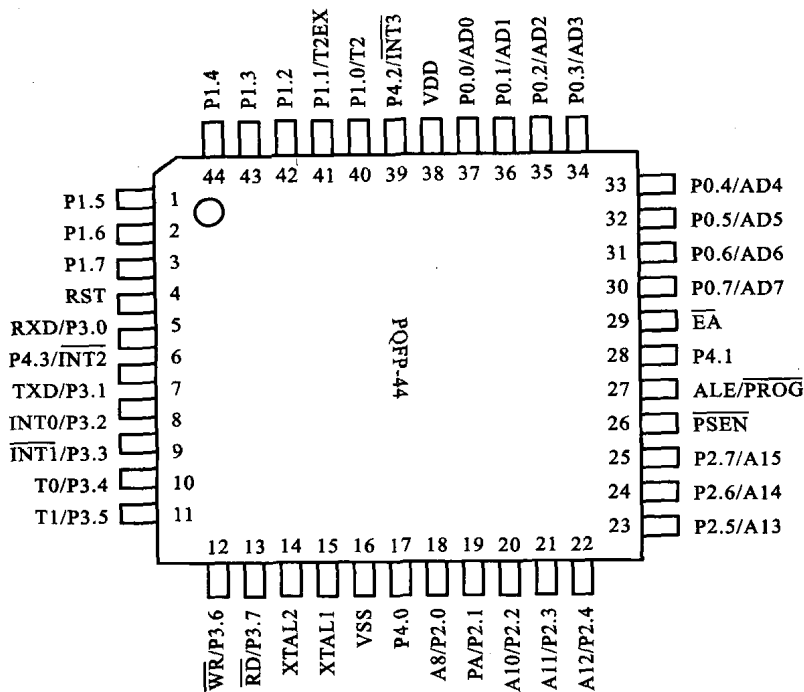


图 1.1.3 STC89C58RD+单片机的 PQFP-44 封装

## 1.2 1280 字节片内 RAM 数据存储

STC89C58RD+片内有 1280 字节的 RAM 数据存储，包括 256 字节的片内 RAM 和 1024 字节的片内扩展 RAM，其中 256 字节的片内 RAM 的地址编码为 00~FFH；1024 字节的片内扩展 RAM 占用外部数据存储的地址，地址编码为 0000H~03FFH。

英特尔生产的早期 8051 单片机的内部 RAM 只有 128(8051)或 256(8052)字节，使用特点为：内部 RAM 的低 128 字节(字节地址:00H~7FH)，寻址方式可以是直接寻址或寄存器间接寻址；内部 RAM 的高 128 字节(字节地址:80H~FFH)，寻址方式只可以是寄存器间接寻址；特殊功能寄存器 SFR(字节地址:80H~FFH)，寻址方式只可以是直接寻址。

值得注意的是，虽然内部 RAM 的高 128 字节与特殊功能寄存器 SFR 所占用的地址编码是一样的，但它们是不同的物理区间，要通过寻址方式加以区分。可见传统的 8051 系列单片机只有 128/256 字节的 RAM 可供用户使用，资源相对较少。STC 系列单片机在单片机内部增加了 RAM 以满足用户需要。STC89C58RD+单片机扩展了 1024 个字节的 RAM(0000H~03FFH)，这里涉及一个特殊功能寄存器——扩展 RAM 管理及禁止 ALE 输出寄存器 AUXR 的使用，该特殊功能寄存器只能写，不能读。

表 1.2.1 特殊功能寄存器 AUXR

寄存器名	地址	7	6	5	4	3	2	1	0	复位值
AUXR	8EH	-	-	-	-	-	-	EXTRAM	ALEOFF	xxxx, xx00

注：“-”为无关位

51 单片机汇编语言规定片外数据存储器必须使用 MOVX 指令访问,而 STC89C58RD+片内扩展 RAM 从软件逻辑上仍划分为外部数据存储器,也必须使用 MOVX 指令访问。例如,MOVX A,@Ri 或 MOVX @Ri,A 指令分别用来读、写片内扩展 RAM 的 00H~FFH 存储空间,共 256 字节。

STC89C58RD+除了内部集成了 1024 字节的片内扩展 RAM,片外还可利用 P0、P2 总线扩展 64 K 字节外部数据存储器,但这将导致片内扩展 RAM 与片外扩展的 RAM 地址产生重叠,解决方法是通过设置 AUXR 寄存器的 EXTRAM 位来选择将地址 0000H~03FFH 分配给片内扩展 RAM 还是分配给片外扩展 RAM。

当 AUXR.1 为“0”,DPTR 的值在 0000H~03FFH 范围内时,MOVX A,@DPTR 或 MOVX @DPTR,A 分别用来读、写片内扩展 RAM 的 0000H~03FFH 存储空间,共 1024 字节,即默认为使用片内扩展 RAM。

当 AUXR.1 为“1”,DPTR 的值在 0000H~03FFH 范围内时,MOVX A,@DPTR 或 MOVX @DPTR,A 分别用来读、写片外扩展 RAM 的 0000H~03FFH 存储空间,共 1024 字节。

当 DPTR 的值大于等于 0400H 时,MOVX A,@DPTR 或 MOVX @DPTR,A 分别用来读、写片外扩展 RAM 存储空间。

此外,用户在使用 MOVX 指令时,要注意 MOVX 指令、MOV 指令及 MOVC 指令的区别:MOV 指令用于访问片内 RAM;MOVX 指令用于访问片内及片外扩展 RAM;MOVC 指令用于访问程序存储器。

### 1.3 32 K 字节片内 flash 程序存储器

STC89C58RD+单片机的片内 flash 程序存储器,被分为 block0 和 block1 两个存储块。block0 容量为 32K 字节,地址范围为 0000H~7FFFH,为用户应用程序区(AP 区);block1 容量为 8 K 字节,地址范围为 E000H~FFFFH,为 ISP 引导区空间。STC89C58RD+单片机的 flash 存储空间如表 1.3.1 所示。

表 1.3.1 STC89C58RD+ flash 存储空间

型号	用户应用程序区	ISP 引导区空间,从 ISP 区启动的 MCU 才有 ISP 功能	
STC89C58RD+	AP space 地址	OK	8 K(保留 IAP)
	0000H~7FFFH	禁止 ISP/IAP 操作	E000H~FFFFH(保留 IAP)

用户应用程序区(AP 区)是指用户编写的程序存放的存储空间,ISP 引导区是指芯片出厂时就已经固化在单片机内部的一段程序所存放的存储空间。STC89C58RD+单片机可以通过 UART 串口实现程序下载,这就是因为芯片在出厂时已经在单片机内部固化了 ISP 引导码,程序首次上电时会先从 ISP 区开始执行代码。实际使用中,也就是用户在下载程序时,先点击下载软件界面上的下载,然后再开启单片机电源,当单片机检测到上位机有下载程序的需要时,便启用 ISP 下载功能给单片机下载程序;若经过短暂的时间没有检测到上位机有下载程序的需求,单片机便会从用户应用程序区(AP 区)开始执行代码。无论是程序下载还是正常启动,ISP 引导码最后均将控制权交给 block0 地址空间,进而执行 block0 中的用户程序,即用户程序总是放在 block0 的 0000H 开始的地址单元,除非用户自行修改了 ISP 引导码(目前对 STC89C58RD+型号的单片机是禁止用户自行

修改 ISP 引导码的,即 ISP 引导码是厂家固化的,用户是不可以编程的)。

STC89C58RD+ 单片机提供了特殊功能寄存器——超级 flash 配置存储器 SFCE 的 2 个位 SC0、SC1,控制上电复位/外部复位后程序是从 block0 还是 block1 启动,SFCE 寄存器的地址为 0B1H。对于 STC89C58RD+而言,SC0、SC1 是同时起作用的,即当 SC1 为“0”且 SC0 为“0”时,程序从 block0 启动;当 SC1 为“1”且 SC0 为“1”时,程序从 block1 启动,此时 block1 物理上是在后 8 K(E000H~FFFFH),逻辑上是映射到前 8 K(0000H~1FFFH),block0 前 8 K 对于 PC 指针不可见。ISP 实现的基本结构如图 1.3.2 所示。

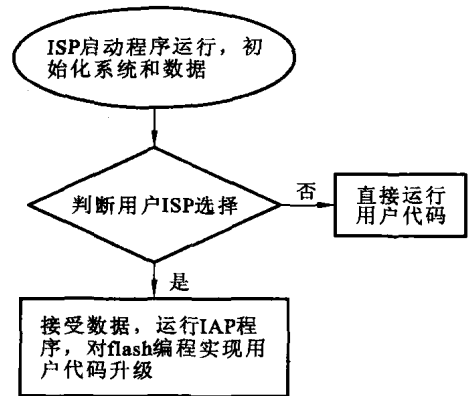


图 1.3.1 ISP 的基本结构流程图

## 1.4 IAP 技术

IAP(in application programming,在应用编程)是用于 MCU 嵌入在终端用户产品的烧录或重新烧录应用代码的技术,换句话说,也就是程序自己可以往程序存储器里写数据或修改程序。实际上单片机的 ISP 功能就是通过 IAP 技术来实现的。IAP 的基本操作与 ISP 相同,IAP 区别于 ISP 是因为它的执行是独立于 8051 核控制器。具有 IAP 功能的 8051 处理器有两个独立的执行控制器,即 8051CPU 核控制器和 IAP flash memory 控制器,它们同步工作,可以实现在两个 flash memory 区块(block0 和 block1)之间相互操作,当 8051 核用户代码在程序区块执行时,用 IAP 一个 flash block 可以编程另一个 flash block。正是因为有了 IAP,从而可以使单片机将数据写入到 flash 程序存储器中,掉电不丢失。当然写入数据的区域与程序运行的存储区要分开操作,以使程序运行不会遭到破坏。

用户要使用 IAP 功能,必须先了解与 IAP 功能设置相关的特殊功能寄存器,IAP 命令通过以下 6 个特殊功能寄存器可被 8051 核心控制器来初始化和控制:SFST(superflash 状态,超级 flash 状态寄存器)、SFCE(superflash 配置,超级 flash 配置寄存器)、SFCEM(superflash 命令,超级 flash 命令寄存器)、SFEDT(superflash 数据,超级 flash 数据寄存器)、SFAL(superflash 低地址位,超级 flash 地址低字节寄存器)、SFAH(superflash 高低值位,超级 flash 地址高字节寄存器),下面将逐个描述这些特殊功能寄存器。

SFST:字节地址为 0B6H,初始化值为 xxx0 0000B。该寄存器有两位运用到 IAP 命令,即 bit3(busy)和 bit2(flash\_busy),这两位用来表示 flash 操作在进行,可以通过 8051 核控制器读取来判断一个 flash 的操作什么时候完成。当 flash 的操作在进行时两位都为“1”,在 burst-program 命令中当每一个字节被编写后 busy 位为“0”,当整个命令被完成后 flash\_busy 为“0”。

SFCE:字节地址为 0B1H,初始化值为 0000 00xxB。该寄存器也有两位运用到 IAP 命令,即 bit7(VIS)和 bit6(IAPEN)。VIS 为使能或者禁用代码从内部 flash 的 block1 执行位,当 VIS 为“0”时,block1 中的地址(F000H 到 FFFFH)对于 PC 来说是不可用的;当

VIS为“1”时,这些地址对PC来说是可见的,并且 block1 中的代码是可执行的。IAPEN为IAP的使能位,它对于IAP的flash控制器来说就像一个开关,当IAPEN为“0”时,所有的IAP命令都被flash控制器忽略;当IAP为“1”时,flash控制器将会执行SFCM寄存器发出的命令。

SFCM:字节地址为0B2H,初始化为0000 0000B。该寄存器有8位都用于IAP命令,其中bit7(FIE)flash中断使能位,表示IAP flash控制器在命令执行完成以后是否应该发出一个中断。当FIE为“0”时,表示不会发出中断,并且在IAP命令完成以后8051核控制器将必须在SFST寄存器中的busy和flash\_busy中做出判断;当FIE为“1”时,表示当IAP命令完成以后IAP flash控制器将会在INT1上产生一个中断,也就是此时对用户而言是禁用外部INT1中断的。SFCM的bit6~bit0的使用,可以查看下文的IAP命令,如表1.4.1所示。

SFDT:字节地址为0B5H,初始化为0000 0000B。该寄存器有8位被用来在8051核控制器和IAP flash控制器中来回的传送数据。在IAP程序操作中,该寄存器值将会被flash控制器编写到SFAH:SFAL中表明的地址;在IAP验证操作中,操作完成以后该寄存器会包括数据定位和地址SFAH:SFAL中;在IAP擦除操作中,该寄存器的值被检验来验证这些查处命令是有效的。

SFAL:字节地址为0B3H,初始化为0000 0000B。该寄存器用来存放IAP flash控制器进入内部flash的16位地址中的低8位地址。

SFAH:字节地址为0B4H,初始化为0000 0000B。该寄存器用来存放IAP flash控制器进入内部flash的16位地址中的高8位地址。

IAP功能实现的步骤为:第一步,开启IAP功能;第二步,把地址传的高字节传给SFAH,低字节传给SFAL;第三步,把待执行的命令传给SFCM;第四步,等到命令执行完后关闭IAP功能。具体控制流程图如图1.4.1所示。

STC89C58RD+的IAP操作的常用命令如表1.4.1所示。

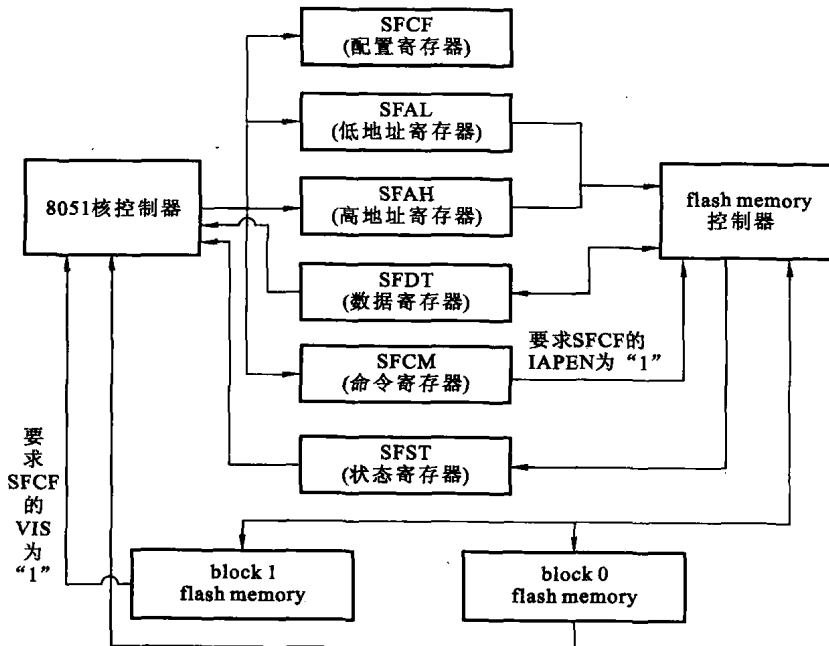


图 1.4.1 IAP 操作的控制流程图

表 1.4.1 STC89C58RD+ 的 IAP 的主要命令

操作	SFAH[7:0]	SFAL[7:0]	SFDDT[7:0]	SFCM[6:0]
chip 擦除	X	X	55H	01H
block 擦除	AH	X	55H	0DH
sector 擦除	AH	AL	X	0BH
byte 编程	AH	AL	DI	0EH
burst 编程	AH	AL	DI	06H
byte 校验	AH	AL	DO	0CH

(1) chip(片)擦除。命令格式:

```

ORL SFCF, #040H          ;启动 IAP
MOV SFDT, #55H
MOV SFCM, #01H          ;写入命令字
Wait:MOV A, SFST        ;查询 SFST.2 位,等待 IAP 操作完成
ANL A, #04H
CJNE A, #00H, Wait
...
    
```

(2) block(块)擦除。命令格式:

```

ORL SFCF, #040H          ;启动 IAP
MOV SFAH, #0F0H/#00H    ;擦除 block0/block1
MOV SFDT, #55H
MOV SFCM, #0DH          ;写入命令字
Wait:MOV A, SFST        ;查询 SFST.2 位,等待 IAP 操作完成
ANL A, #04H
CJNE A, #00H, Wait
...
    
```

(3) sector(扇)区擦除。命令格式:

```

ORL SFCF, #040H          ;启动 IAP
MOV SFAH, 区段地址高字节
MOV SFAL, 区段地址低字节
MOV SFCM, #0BH          ;写入命令字
Wait:MOV A, SFST        ;查询 SFST.2 位,等待 IAP 操作完成
ANL A, #04H
CJNE A, #00H, Wait
...
    
```

(4) byte(字节)编程。命令格式:

```

ORL SFCF, #040H          ;启动 IAP
MOV SFAH, 要编程地址高字节
MOV SFAL, 要编程地址低字节
MOV SFDT, 要编程的数据
    
```

```

MOV SFCM, # 0EH          ;写入命令字
Wait:MOV A, SFST         ;查询 SFST.2 位,等待 IAP 操作完成
ANL A, # 04H
CJNE A, # 00H, Wait
.....

```

(5) byte 字节校验。命令格式:

```

ORL SFCF, # 0C0H        ;启动 IAP
MOV SFAH, 要校验数据的地址高字节
MOV SFAH, 要校验数据的地址低字节
MOV SFCM, # 0CH        ;写入命令字
NOP
MOV A, SFDT            ;A 保存规定单元地址读出的数据
.....

```

用户在使用 IAP 时需要注意:启动 IAP 要尽量使用 ORL 指令,不要使用 MOV 指令,否则极易造成系统软件复位、程序存储器 REMAP 等;IAP 编程的数据读出,要使用字节校验指令,不要使用 MOVC 指令,MOVC 指令在 IAP 中对有地址重叠的地方不能读出正确的数值;IAP 编程不要忘记打开 IAP 功能,特别是数据读出时也需要打开 IAP 功能。

## 1.5 双数据指针的使用

在 8051 体系中,数据指针 DPTR 作为一个特殊的 16 位寄存器,用于寻址 64KB 的外部数据存储空间(XDATA)或程序存储空间(CODE),使用时通常设计成一个 16 位指针,指向一个常数表。而现在有许多 8051 派生型器件都支持双数据指针,同时有两个 16 位指针可以被使用,以提高数据指针使用的性能。STC89C89RD+ 的 DPTR 就被增强为 DPTR0 和 DPTR1 两个 16 位寄存器,但是这两个 16 位寄存器仍然使用与原来相同的寄存器标号及地址,系统增加了一个双数据指针特殊功能寄存器 AUXR1 来管理。

表 1.5.1 双数据指针特殊功能寄存器 AUXR1

寄存器名	地址	7	6	5	4	3	2	1	0	复位值
AUXR1	A2H	—	—	—	—	GF2	—	—	DPS	xxxx,0xx0

表中,GF2 为通用功能用户定义位;DPS 为 DPTR 寄存器选择位,“0”表示选择 DPTR0,“1”表示选择 DPTR1;AUXR1 特殊功能寄存器是只支持字节操作,不支持位操作的,加之 DPS 位于该寄存器的最低位,所以在实际使用中,可通过一个简单的 INC AUXR1 指令,DPS 位便会反转,由“0”变成“1”或由“1”变成“0”,即可以实现两个数据指针的切换。例如,

```

AUXR1 DATA 0A2H      ;定义 AUXR1 特殊功能寄存器地址
MOV AUXR1, # 00H      ;DPS 为"0",选择使用 DPTR0
MOV DPTR, # 100H      ;赋值 DPTR0 为 100H
MOV A, # 55H

```

```

MOVX @DPTR,A      ;将扩展 RAM 的 100H 单元赋值为 55H
INC AUXR1          ;DPS 为 "1", 选择使用 DPTR1
MOV DPTR,#100H     ;赋值 DPTR1 为 100H
MOVX A,@DPTR      ;读取扩展 RAM 的 100H 单元值
INC AUXR1          ;DPS 为 "0", 选择使用 DPTR0
MOV DPTR,#200H     ;赋值 DPTR0 为 200H
MOVX @DPTR,A      ;将扩展 RAM 的 100H 单元赋值为累加器 A 中的值

```

在 Keil  $\mu$ Vision2 环境下,软件仿真 STC 系列器件中有双 DPTR 的型号时,仿真器中会有 AUXR1、DPTR0、DPTR1 三个寄存器。如果不使能双 DPTR 特性,仿真时 DPTR 的值是混乱的。以 STC89C58RD+ 为例,双 DPTR 特性的仿真设定的具体步骤为:在“File/Device Database”菜单中选择“STC89C58RD+”型号,在“Options”框中“CPU=”一项后加上“MODP2”,然后单击“Update”更新器件库,即可打开双数据指针调试,此时再启动 Debug,就会有 AUXR1、DPTR[0]、DPTR[1]这 3 个寄存器。若加载上述程序段时,会清楚地看到双数据指针的操作和 AUXR1 的变化。

## 1.6 P4 口及中断

STC89C58RD+ 单片机的中断与传统的 8052 完全兼容,由于 LQFP-44 体积小,扩展了寄存器 P4 的 P4.0~P4.3 口,STC89C58RD+ 单片机的 36 个 I/O 口如表 1.6.1 所示。

表 1.6.1 STC89C58RD+ 单片机的 I/O 口

寄存器名	地址	7	6	5	4	3	2	1	0	复位值
P0	80H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	1111,1111
P1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	1111,1111
P2	A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	1111,1111
P3	B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	1111,1111
P4	E8H	—	—	—	—	P4.3	P4.2	P4.1	P4.0	xxxx,1111

STC89C58RD+ 单片机有 8 个中断源,除了传统的 8052 的 6 个中断源,即外部中断 0、定时/计数器中断 0、外部中断 1、定时/计数器中断 1、串口中断和定时/计数器中断 2,还扩展了外部中断 2(P4.2 引脚)和外部中断 3(P4.3 引脚)2 个中断源。这 8 个中断源分为 4 个优先级。涉及的中断特殊功能寄存器有 4 个,即中断允许寄存器 IE、中断优先级低字节 IP、中断优先级高字节 IPH 及辅助中断控制寄存器 XICON,如表 1.6.2 所示。

表 1.6.2 中断的特殊功能寄存器

寄存器名	地址	7	6	5	4	3	2	1	0	复位值
IE	A8H	EA	—	ET2	ES	ET1	EX1	ET0	EX0	0000,0000
IP	B8H	—	—	PT2	PS	PT1	PX1	PT0	PX0	xx00,0000
IPH	B7H	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
XICON	C0H	PX3	EX3	IE3	IT3	PX2	EX2	IE2	IT2	0000,0000

由表 1.6.2 可见,这 4 个中断特殊功能寄存器中的有些位是与传统 8052 完全兼容的,在这里将不再做赘述,下面主要对新增的一些位及功能做相应的描述。

PX3H、PX3:外部中断 3 的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PX2H、PX2:外部中断 2 的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PX1H、PX1:外部中断 1 的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PX0H、PX0:外部中断 0 的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PT2H、PT2:定时器 2 中断的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PT1H、PT1:定时器 1 中断的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PT0H、PT0:定时器 0 中断的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

PSH、PS:串口中断的优先级设置位。“00”表示优先级为最低级别 0;“01”表示优先级为 1;“10”表示优先级为 2;“11”表示优先级为最高级别 3。

EX3:外部中断 3 允许位。“1”表示允许外部中断 3;“0”表示禁止外部中断 3。

EX2:外部中断 2 允许位。“1”表示允许外部中断 2;“0”表示禁止外部中断 2。

IT3:外部中断 3 的外部触发方式位。“1”表示下降沿触发有效;“0”表示低电平触发有效。

IT2:外部中断 2 的外部触发方式位。“1”表示下降沿触发有效;“0”表示低电平触发有效。

IE3:外部中断 3 的中断标志位。“1”表示有中断请求产生;“0”表示暂无中断请求产生。

IE2:外部中断 2 的中断标志位。“1”表示有中断请求产生;为“0”表示暂无中断请求产生。

STC89C58RD+单片机的 8 个中断的中断请求标志位及中断向量的入口地址如表 1.6.3 所示,同一级别的中断优先级顺序为:外部中断 0、定时/计数器 0 中断、外部中断 1、定时/计数器 1 中断、串口中断、定时/计数器 2 中断、外部中断 2、外部中断 3。

表 1.6.3 中断向量表入口地址

中断源	中断请求标志位	中断向量地址
外部中断 0	IE0	0003H
定时/计数器 0 中断	TF0	000BH
外部中断 1	IE1	0013H
定时/计数器 1 中断	TF1	001BH
串口中断	RI+TI	0023H
定时/计数器 2 中断	TF2+EXF2	002BH
外部中断 2	IE2	0033H
外部中断 3	IE3	003BH



## 1.7 看门狗的应用

看门狗(watchdog)概念及其应用出现在由单片机构成的微型计算机系统中,由于单片机的运行会受到各种干扰,造成程序的跑飞,也可能陷入死循环,程序的正常运行被打断,使得单片机控制的系统无法继续工作,这样会使整个系统陷入停滞状态,发生不可预料的后果,所以出于对单片机运行状态进行实时监测的考虑,便产生了一种专门用于监测单片机程序运行状态的电路,俗称看门狗。

看门狗电路有时又称为看门狗定时器(watchdog timer),顾名思义就是一个定时器电路,通常看门狗电路是用一个专门的看门狗芯片连接单片机来实现的。看门狗电路一般有两个引脚:一个用来作输入,用作喂狗(kicking the dog or service the dog);另一个用来作输出,输出到单片机的复位端,当程序跑飞时,看门狗定时超时,就会给出一个复位信号到单片机,使单片机复位。

看门狗的工作过程是这样的:看门狗芯片的输入和单片机的一个 I/O 引脚相连,该 I/O 引脚通过单片机的程序控制,可定时地向看门狗芯片的输入引脚上送入一个信号,这段程序是分散地放在单片机其他控制语句中间的,而且只要看门狗接受到了特定信号,内部定时器就会重新计时,一旦单片机由于干扰造成程序跑飞后而陷入某一程序段进入死循环状态时,给看门狗输入引脚送特定信号的程序便将不被执行,如果到达内部定时器定时时长时,看门狗电路输入仍然得不到单片机送来的信号,便会在它和单片机复位引脚相连的输出引脚上送出一个复位信号,使单片机发生复位,这样单片机将从程序存储器的起始位置重新开始执行程序,从而便实现了单片机的自动复位。

传统的 8051 单片机是没看门狗功能的,若用户需要使用看门狗就需要外接看门狗电路,这样就增加了电路设计的复杂性。STC89C58RD+单片机内部自带有看门狗,通过对相应特殊功能寄存器的设置就可实现看门狗的应用,用户使用起来就非常方便,STC89C58RD+单片机内部有一个专门的看门狗定时器控制寄存器,不支持位寻址。此外值得注意的是,STC 单片机的看门狗一旦启动,就不能停止。

表 1.7.1 看门狗定时器控制寄存器

寄存器名	地址	7	6	5	4	3	2	1	0	复位值
WDT_CONTR	E1H	—	—	EN_WDT	CKR_WDT	IDLE_WDT	PS2	PS1	PS0	xxxx,xxxx

表 1.7.1 中,EN\_WDT 为看门狗允许位,“1”表示启动看门狗;CLR\_WDT 为看门狗清零位,“1”表示看门狗定时器将重新计数,硬件自动清“0”此位;IDLE\_WDT 为看门狗“IDLE”模式位,“1”表示看门狗定时器在单片机的“空闲模式”计数,“0”表示看门狗定时器在单片机的“空闲模式”时不计数;PS2、PS1、PS0 为看门狗定时器预分频值位,不同值对应预分频数如表 1.7.2 所示。