

<<单片机技术-第2版>>

图书基本信息

书名：<<单片机技术-第2版>>

13位ISBN编号：9787040347524

10位ISBN编号：7040347520

出版时间：2012-7

出版时间：姜大源、王胜元 高等教育出版社 (2012-07出版)

作者：姜大源，王胜元 编

页数：200

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<单片机技术-第2版>>

### 内容概要

《单片机技术（第2版）》是电气智能技术应用系列用书，是教育部职业教育与成人教育司推荐教材的修订版。

《单片机技术（第2版）》为适应21世纪对电气智能技术应用型人才的需要编写，教材用丰富的实例，解析MCS-51单片机在人们实际生活中的应用，重应用、重动手能力的培养。该书主要内容有MCS-51单片机的硬件结构、指令系统、汇编语言程序设计、中断系统、系统扩展等理论知识等，并结合实例详细地讲述了单片机应用系统的设计、开发、调试流程。

《单片机技术（第2版）》可作为电子信息、电气控制应用技术培训用书、“电气智能技术应用”工程师认证培训教材以及全国职业院校电类专业教学用书，也可供相关工程人员参考。

## 书籍目录

1 单片机基础 1.1 认识单片机 1.1.1 概述 1.1.2 单片机的发展及主流产品 1.1.3 单片机的应用 1.2 计算机中的数制及其相互转换 1.2.1 二进制数和十进制数之间的转换 1.2.2 十六进制数和十进制数之间的转换 1.2.3 二进制数和十六进制数之间的转换 1.3 二进制数的运算 1.3.1 算术运算 1.3.2 逻辑运算 1.4 带符号数的表示 1.5 定点数和浮点数 1.6 BCD码和ASCII码 习题 2 单片机的硬件结构和原理 2.1 概述 2.2 MCS—51系列单片机的硬件结构 2.3 中央处理器CPU 2.4 存储器的结构 2.4.1 内部数据存储器 2.4.2 外部数据存储器 2.4.3 程序存储器 2.5 并行输入/输出(I/O) 2.5.1 P0口 2.5.2 P1口 2.5.3 P2口 2.5.4 P3口 2.5.5 P0口~P3口的带负载能力及端口要求 2.6 单片机的引脚及功能 2.7 单片机工作的基本时序 2.7.1 典型指令的取指和执行时序 2.7.2 单片机的复位电路 习题 3 MCS—51系列单片机指令系统 3.1 MCS—51系列单片机指令格式 3.2 寻址方式 3.2.1 寄存器寻址 3.2.2 直接寻址 3.2.3 寄存器间接寻址 3.2.4 立即寻址 3.2.5 位寻址 3.2.6 变址寻址 3.2.7 相对寻址 3.3 指令中符号意义说明 3.4 数据传送类指令 3.4.1 内部RAM和特殊功能寄存器(SFR)数据传送指令 3.4.2 累加器A与片外RAM之间的数据传送指令 3.4.3 程序存储器ROM向累加器A传送指令 3.4.4 堆栈操作 3.4.5 字节交换指令 3.5 算术运算类指令 3.5.1 不带进位位的加法指令 3.5.2 带进位位的加法指令 3.5.3 带借位的减法指令 3.5.4 加1指令 3.5.5 减1指令 3.5.6 乘法指令 3.5.7 除法指令 3.5.8 十进制调整指令 3.6 逻辑运算类指令 3.6.1 逻辑与运算指令 3.6.2 逻辑或运算指令 3.6.3 逻辑异或运算指令 3.6.4 累加器清零取反指令 3.6.5 移位指令 3.7 控制转移类指令 3.7.1 无条件转移指令 3.7.2 条件转移指令 3.7.3 子程序调用和返回指令 3.8 位操作指令 3.8.1 位传送指令 3.8.2 位置位和位清零指令 3.8.3 位逻辑运算指令 习题 4 汇编语言程序设计简介 4.1 伪指令 4.2 汇编语言程序设计 4.2.1 汇编语言程序设计的基本概念 4.2.2 汇编语言命令格式 4.2.3 简单程序设计 4.2.4 分支程序设计 4.2.5 循环程序设计 4.2.6 查表程序设计 4.2.7 子程序 4.2.8 汇编语言编程举例 习题 5 中断系统 5.1 中断概述 5.1.1 中断的特点 5.1.2 中断的基本概念 5.1.3 中断的过程 5.1.4 中断嵌套 5.2 MCS—51系列单片机中断系统 5.2.1 中断源 5.2.2 中断控制 5.2.3 中断响应过程 5.2.4 中断响应等待时间 5.2.5 中断请求的撤除 5.3 中断系统的应用 5.3.1 中断初始化 5.3.2 中断服务子程序 习题 6 MCS—51系列单片机定时器/计数器及串行接口 6.1 MCS—51系列单片机的定时器/计数器 6.1.1 定时器/计数器的结构及工作原理 6.1.2 定时器/计数器的工作方式 寄存器和控制寄存器 6.1.3 定时器/计数器的工作方式 6.1.4 定时器/计数器应用举例 6.2 MCS—51系列单片机的串行接口 6.2.1 串行通信概述 6.2.2 与串行口相关的特殊功能寄存器 6.2.3 串行口的工作模式 6.2.4 波特率的选择 6.2.5 单片机的多机通信 6.2.6 串行口应用实例 习题 7 单片机系统扩展与接口技术 7.1 外部总线的扩展 7.1.1 片外三总线结构 7.1.2 总线驱动能力 7.2 外部存储器的扩展 7.2.1 程序存储器的扩展 7.2.2 数据存储器的扩展 7.2.3 同时扩展程序存储器和数据存储器 7.3 输入/输出接口的扩展 7.3.1 8155的结构及功能说明 7.3.2 8155的寄存器 7.3.3 8155和89C51的接口连接方法 7.4 A/D和D/A接口功能的扩展 7.4.1 MCS—51系列单片机与8位D/A转换器接口技术 7.4.2 MCS—51系列单片机与8位A/D转换器接口技术 习题 8 单片机应用系统的设计与开发 8.1 单片机应用系统的开发过程 8.2 MedWin仿真软件的使用 8.2.1 MedWin的安装与启动 8.2.2 编辑程序 8.2.3 汇编语言源程序的汇编 8.2.4 汇编程序的调试与运行 8.3 Keil C51 8.3.1 Keil C51集成开发环境— 8.3.2 Keil C51  $\mu$  Vision2软件安装 8.3.3 Keil C51软件的使用 8.3.4 Keil C编程实例 8.4 单片机系统设计实例 8.4.1 单片机控制交通灯 8.4.2 霓虹灯 8.4.3 单片机控制的点阵动态扫描显示电路 8.4.4 电子闹钟 8.4.5 串行通信 习题 9 单片机系统的抗干扰设计 9.1 硬件抗干扰设计 9.2 软硬件结合——看门狗(Watchdog)技术 9.3 软件抗干扰设计 附录 MCS—51系列单片机指令表 参考文献

## 章节摘录

版权页：插图：MOVX A.@RO 即把RO指定的外部RAM单元的内容送至累加器A。

此外，对于堆栈操作指令（PUSH和POP）也应算作是寄存器间接寻址，即以堆栈指针（SP）作间址寄存器的间接寻址方式。

3.2.4 立即寻址 所谓立即寻址，就是操作数在指令中直接给出。

通常把出现在指令中的操作数称为立即数，因此就把这种寻址方式称为立即寻址。

为了与直接寻址指令中的直接地址相区别，在立即数前面加“#”标志。

例如 MOV A.#3AH 其中3AH就是立即数，指令功能是把数据3AH送至累加器A。

除8位立即数外，MCS—51指令系统中还有一条16位立即寻址指令，即 MOV DPTR, #data16 其功能是把16位立即数送至数据指针DPTR。

3.2.5 位寻址 MCS—51有位处理功能，可以对数据位进行操作，因此就有相应的位寻址方式。

例如指令 MOV C, 20H 其功能是把位地址为20H的位状态送至进位标志位C。

位寻址方式的寻址范围包括：内部RAM中的位寻址区。

单元地址为20H～2FH，共16个单元128位，位地址是00H—7FH。

对这128个位的寻址用直接位地址表示，例如，MOV C, 2BH指令的功能是把位寻址区的2BH位状态送至进位标志位。

专用寄存器的可寻址位。

可供位寻址的专用寄存器共有11个，实有寻址位83位，对这些寻址位在指令中有以下4种表示方法：a. 直接使用位地址。

这些位地址在表2.4中已列出。

例如，PSW寄存器位5地址为D5H。

b. 位名称表示方法。

专用寄存器中的一些寻址位是有符号名称的，例如，PSW寄存器位5是FO标志位，则可使用FO表示该位。

c. 单元地址加位的表示方法。

例如，DOH单元（即PSW寄存器）位5，表示为DOH.5。

d. 专用寄存器符号加位的表示方法。

例如，PSW寄存器的位5表示为PSW.5。

一个寻址位有多种表示方法，初看起来有些复杂，但实际上这将为程序设计带来方便。

3.2.6 变址寻址 MCS—51的变址寻址是以DPTR或PC作基址寄存器，以累加器A作变址寄存器，并以两者内容相加形成的16位地址作为操作数地址。

例如指令 MOVX A.@A+DPTR 其功能是把DPTR和A的内容相加，两者之和作为数据的存储单元地址，再把该地址单元的内容送A。

<<单片机技术-第2版>>

编辑推荐

《中等职业教育国家规划教材配套教学用书:单片机技术(电气技术应用、电气运行与控制专业)》可作为电子信息、电气技术应用、机电技术应用专业教材以及全国职业院校电类专业教学用书,也可供相关工程人员参考。

<<单片机技术-第2版>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>