

新世纪高校机电类规划教材

单片机原理与应用 学习概要及题解

主 编	霍孟友	王爱群
副主编	裘著燕	李建美
参 编	李学勇	
主 审	路长厚	



机械工业出版社

本书以新世纪高校机电类规划教材《单片机原理与应用》为基础,对 MCS—51 系列单片机学习要点、难点进行了概括,解答了原教材中的习题,同时收集了近年来一些理工科高校考研试题以及有关单片机教材中的习题,以基本概念填空、单项选择、程序分析、综合应用设计等习题形式做了系统解答,基本涵盖了 MCS—51 系列单片机学习的基本知识与要点问题。本书共分为 10 章,内容包括计算机原理概述、单片机结构原理、指令系统和汇编语言程序设计、存储器扩展技术、中断系统、定时器/计数器原理及应用、串行接口及应用、并行接口扩展及应用、A/D 与 D/A 转换器接口技术、综合应用等部分。

本书可作为高等学校机电类专业微机原理课程学习的辅助教材,也可作为工程技术人员加深理解单片机应用技术的参考用书。

图书在版编目 (CIP) 数据

单片机原理与应用学习概要及题解/霍孟友,王爱群主编. —北京:机械工业出版社,2005.3

新世纪高校机电类规划教材

ISBN 7-111-16092-4

I. 单... II. ①霍...②王... 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 (2005) 第 007610 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:刘小慧 版式设计:张世琴 责任校对:张 媛

封面设计:姚 毅 责任印制:杨 曦

北京机工印刷厂印刷·新华书店北京发行所发行

2005 年 3 月第 1 版第 1 次印刷

787mm×1092mm¹/₁₆·8.75 印张·214 千字

定价:15.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68326294

封面无防伪标均为盗版

前 言

由于单片机小巧、控制功能强、应用灵活、价格低廉，非常适合于机、电、仪一体化产品，在工业控制、机电一体化产品、家用电器、智能仪表等诸多领域显现出了较为广阔的应用前景。为此，许多理工科高校的“微机原理及应用”课程以单片机作为学习对象，对微机原理结构、接口方法以及应用技术进行系统性的学习。

尽管各有关单片机原理教材的编者注重了教材内容循序渐进、内容突出等编写技术问题，使学生在在学习过程中尽快地进入角色、掌握学习内容，但是，同学们还是普遍反映单片机原理学习抽象、理解较为困难，尤其是习题无从下手，习题解答不全面，甚至有的学生直到课程学习结束才有系统性的认识，大大地影响了学习效果。目前，大多数学习单片机的同学希望有一本能够检验他们阶段性学习效果的辅导教材，比如有关题解的教材，通过这种教材而不是内容大体相似的微机原理教材的辅助对他们理解单片机学习要点的程度进行直接检验。

本书是以新世纪高校机电类规划教材《单片机原理与应用》为基础，主要围绕目前同学们学习单片机原理知识所遇到的问题而编写的。为了使内容更为紧凑，对原教材的章节进行了部分调整。全书内容基本分为学习要求与内容概要以及习题解答两大部分。学习要求与内容概要部分把每一章学习要求，内容要点、难点进行了概述，通过这部分内容的学习可以使学生提纲携领地掌握每一章的学习内容、要点；习题解答部分分为两个方面，一是对原教材中大部分（有一定难度）的习题进行了解答，二是增选了有关单片机学习教材的习题以及近年来一些理工科高校考研试题，以基本概念填空、单项选择、程序分析、综合应用设计等习题形式做了系统解答。在习题选择方面力求涵盖面广，特别注重应用实例的解答。通过本书的进一步学习辅导，希望学生扎实地掌握微机原理知识和工程应用的基本方法，为解决实际问题打下良好的基础。

参加本书编写的编者具有较为丰富的教学和实践经验，编写人员主要为山东大学的霍孟友、王爱群、裘著燕、李建美和李学勇，由霍孟友、王爱群任主编，霍孟友负责统稿，山东大学路长厚教授负责主审。

在本书编写过程中，我们参考并引用了国内部分优秀教材的有关资料，在此谨向他们深表谢意。

限于编者的水平，书中错误和不足之处在所难免，恳请读者批评指正。

编 者

目 录

前言

第一章 概述	1	第一节 学习要求与内容概要	69
第一节 学习要求与内容概要	1	第二节 原教材习题及解答	71
第二节 原教材习题及解答	3	第三节 增选习题及解答	74
第三节 增选习题及解答	4		
第二章 MCS—51 单片机的结构与原理	7	第七章 MCS—51 单片机串行口及应用	80
第一节 学习要求与内容概要	7	第一节 学习要求与内容概要	80
第二节 原教材习题及解答	11	第二节 原教材习题及解答	81
第三节 增选习题及解答	14	第三节 增选习题及解答	85
第三章 MCS—51 单片机指令系统与程序设计	22	第八章 MCS—51 单片机并行口扩展及应用	93
第一节 学习要求与内容概要	22	第一节 学习要求与内容概要	93
第二节 原教材习题及解答	25	第二节 原教材习题及解答	97
第三节 增选习题及解答	34	第三节 增选习题及解答	104
第四章 MCS—51 单片机存储器扩展	47	第九章 单片机与 A/D、D/A 转换器的接口	112
第一节 学习要求与内容概要	47	第一节 学习要求与内容概要	112
第二节 原教材习题及解答	52	第二节 原教材习题及解答	115
第三节 增选习题及解答	55	第三节 增选习题及解答	116
第五章 中断与中断系统	59	第十章 MCS—51 单片机应用系统设计方法与举例	119
第一节 学习要求与内容概要	59	第一节 学习要求与内容概要	119
第二节 原教材习题及解答	61	第二节 原教材习题及解答	120
第三节 增选习题及解答	63	第三节 增选习题及解答	125
第六章 定时器/计数器	69	参考文献	135

第一章 概 述

第一节 学习要求与内容概要

一、学习要求

(1) 了解计算机的基本结构以及工作原理；了解单片机的基本概念及其与一般计算机的区别；了解目前常用的单片机系列及其应用领域。

(2) 了解数制的基本知识并完成不同数制之间的转换；理解计算机中带符号数的原码、反码以及补码的表示方法及其求法；理解 BCD 码以及 ASCII 码的意义及其表示方法。

(3) 了解基本门电路的原理及其表示方法，理解组合逻辑电路和时序逻辑电路的区别；理解三态门驱动电路的基本原理，了解一些常用的三态门电路；理解 D 触发器的基本原理，了解由其组成的一些基本的控制器件。

二、一般计算机原理与结构学习内容概要

一个完整的计算机系统由计算机软件和计算机硬件两大部分组成。软件主要包括系统软件和应用软件。硬件主要包括运算器、存储器、控制器和输入/输出设备。

三、单片机基本概念学习概要

单片机是在一块芯片上集成了中央处理器（CPU）、随机存储器（RAM）、程序存储器（ROM）、中断系统、定时器/计数器以及 I/O 接口等的微型计算机。国际上通常称其为微控制器 MCU，又称为嵌入式微控制器 EMCU。

MCS-51 单片机是 8 位单片机的主流机型，分为三个基本产品：8051、8751 和 8031。其他 8 位单片机主要有 ATMEL 公司的 AT89 系列 Flash 单片机；Microchip 公司的 PIC 单片机以及 TI 公司的 MSP430 单片机等。

单片机的应用领域为日常的电器产品、工业控制、网络以及通信。

四、数制学习内容概要

1. 进位计数制

一个数的数值可以用不同的进制来表示。十进制（Decimal）用到的数码为 0、1、2、3、4、5、6、7、8、9，运算规则：逢十进一，借一当十；二进制（Binary）用到的数码为 0、1，其运算规则：逢二进一，借一当二；十六进制（Hexadecimal）用到的数码为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，运算规则：逢十六进一，借一当十六。

2. 不同数制之间的转换

十六进制数转换为二进制数，只要把每一位十六进制数用对应的四位二进制数代替就可以。对于带小数点数的转换，小数点后面的每位十六进制数同样对应转换为 4 位二进制数，最末尾的零可以省略。

二进制数转换为十六进制数，二进制数的整数部分由小数点向左，每 4 位二进制数对应一位十六进制数，最高位不足的部分左面补零；小数部分由小数点向右，每 4 位二进制数对应一位十六进制数，最后面不足部分右面补零，就转换成了十六进制数。

二进制数或十六进制数转换成十进制数，按照科学计数法把二进制数、十六进制数展开就得到了十进制数。

十进制数转换为二进制数或十六进制数时，要把整数部分和小数部分分别换算，然后再将转换结果加在一起。

(1) 整数部分转换 整数部分连续被基数 2 或 16 所除，依次记下余数，直到商为 0 为止；第一个余数是转换二进制数的最低位 K_0 ，最后一个余数是最高位 K_n 。

(2) 小数部分的转换 十进制小数连续乘以基数 2 或 16，依次记下积的整数部分，直到积为 0 为止。第一个整数是二或十六进制小数的最高位 K_{-1} ，最后一个整数是最低位 K_{-m} 。

有些十进制数不能精确地转换成十六进制数。

3. 计算机中数的表示方法

一个带符号数在计算机中可以分别用原码、反码或补码三种方法表示。

原码是符号位用 0 表示正数、用 1 表示负数，而数值位表示数的绝对值。

正数的原码与原数值大小相同，负数的原码最高位为符号位，其余为数字位。正数的反码与正数的原码相同，而负数的反码是由其数值位按位求反得到的。

正数的补码与正数的原码相同，负数的补码是由它的反码加 1 以后得到的。

4. BCD 码

如果用 4 位二进制编码来表示一位十进制数，就形成了二进制编码的十进制数，简称二-十进制数，又称 BCD 码。要将十进制数和 BCD 码进行转换，只要把 0~9 与对应的 0000B~1001B 互换就行了。

压缩 BCD 码是指用一个字节的 8 位二进制数表示两个 BCD 码；非压缩 BCD 码指用一个字节的低 4 位二进制数表示 1 个 BCD 码，而高 4 位没有意义。

5. ASCII 码

ASCII 码是美国标准信息交换代码 (American Standard Code for Information Interchange) 的缩写。ASCII 码用 7 位二进制数表示，包括英文 26 个大写字母、26 个小写字母、0~9 十个数字，还有一些专用符号如“:”、“!”、“%”及控制符号，如换行、换页、回车等共 128 个字符。

五、数字电路基础学习内容概要

数字电路是由电阻、电容、二极管、晶体管等基本电路元件进行不同组合构成的逻辑电路。按照电路的基本组成为 TTL 门电路、CMOS 电路等多种类型；按集成度又可以分为：小规模集成电路、中规模集成电路、大规模集成电路和超大规模集成电路等；按照有没有记忆能力，数字逻辑电路分为：组合逻辑电路和时序逻辑电路两大类。

1. 基本门电路

基本门电路属于组合逻辑电路。基本门电路有与门电路、或门电路、非门电路以及异或门电路等。

2. 三态门驱动电路

将输出含有“1”(H)、“0”(L)、高阻态(Z)三态的集成电路(IC)称为三态集成电路。高阻态是指电路的输出管脚(输出端)与芯片内部电路呈高电阻绝缘状态。

3. 触发器

触发器是计算机存储装置的基本单元,是时序电路中最具代表性的一种电路。常见的触发器有三种基本类型:RS 触发器、D 触发器以及 JK 触发器。触发器都具有两个状态相反的输出端 Q 和 $\bar{Q}$ 。

第二节 原教材[⊖]习题及解答

1-1 计算机的基本结构由哪几部分组成,其中各部分的作用是什么?

答:一个完整的计算机系统由计算机硬件和计算机软件两大部分组成。

硬件各个部分及其作用:运算器——用来完成计算机的运算;存储器——用于记忆原始题目、原始数据和中间结果,以及存放使机器能自动进行运算而编制的各种命令;控制器——由它根据事先给定的命令发出各种控制信息,使整个计算过程一步步地进行;输入设备——用于输入原始数据和命令;输出设备——用于输出计算结果。

软件主要包括系统软件和应用软件。系统软件是为了方便用户和充分发挥计算机效能,向用户提供的一系列软件,包括监控程序、操作系统、汇编系统、解释程序、编译程序、诊断程序以及程序库等;应用软件是专门为解决某个应用领域里的具体任务,如进行生产过程控制或数据处理问题而编制的程序。

1-2 单片机与微型计算机的区别与相似之处有哪些?

答:单片机就是在一块芯片上集成了中央处理器(CPU)、随机存储器(RAM)、程序存储器(ROM)、中断系统、定时器/计数器,以及 I/O 口等的微型计算机;单片机是计算机发展的一个分支,它具有一般微型计算机所具有的各种硬件结构,而且也必须在软件的支持下才能工作,所以无论从功能还是结构上说,单片机都是一种简单的计算机。它与一般计算机的区别就在于它的高度集成性,这也就决定了它的体积小、功耗低、功能强、性能价格比高等优点,所以它广泛地应用于控制领域。

1-3 通用型单片机与专用型单片机有何区别?专用型单片机完成的功能,通用型单片机能否完成?

答:通用型单片机,其内部资源比较丰富,性能全面,适应性强,能满足各种不同的应用要求。专用型单片机是针对一种产品或一种控制应用而专门设计的,设计时已经对系统结构、软硬件资源、可靠性和成本等进行了优化处理,在使用时不需要进行进一步的设计就可以完成控制功能。

通用型单片机一般都可以完成专用型单片机完成的功能,只不过系统结构、开发成本等都不如专用型单片机理想。

1-4 单片机又叫“嵌入式微控制器”,为什么?

答:通用计算机体积相对较大且成本较高,无法嵌入到大多数对象中去,而单片机体积

⊖ 原教材指霍孟友主编的由机械工业出版社出版的《单片机原理与应用》(ISBN 7-111-13274-2/TP·2979 (课)). 后同。

较小、成本较低,从而满足工控对象的嵌入式应用要求,现在它已经广泛应用于自动化装置、智能化仪器仪表、过程控制和家用电器等嵌入式系统,所以单片机又叫“嵌入式微控制器”。

1-5 MCS—51 系列单片机片内程序存储器有“掩膜 ROM 型”和“EPROM 型”,二者有何不同之处?

答:掩膜 ROM 型,采用一道掩膜工艺过程决定其中的信息,一旦出厂,信息就不可以改变了,所以又称为不可改写的 ROM;对于 EPROM 型,用户可以根据需要擦除其中的内容,然后再用编程器写入新的信息,所以又称为可擦除的 ROM。

第三节 增选习题及解答

一、填空题

1. 请写出与下列英文所对应的计算机部件中文名称: ALU _____; Trigger _____; Register _____; Memory _____; ROM _____; RAM _____; Counter _____; Accumulator _____。

2. 设 $(AL) = 45H$, 若是无符号数, 它代表 _____, 若是带符号数, 它代表 _____, 若是 BCD 数, 它代表 _____, 若是 ASCII 码, 它代表 _____。

3. 16 位无符号定点整数的数值表示范围是 _____; 16 位整数 (含一位符号位) 的补码表示范围是 _____; 8 位无符号整数的数值表示范围是 _____; 8 位整数 (含一位符号位) 的补码表示范围是 _____。

4. 异或门的运算规则是相异为 _____, 相同为 _____。

5. 具有两个稳定状态并能接收、保持和输送信号的电路叫 _____。

6. TTL、CMOS 与非门、与门的多余的输入端的处理方法是 _____。

7. TTL、CMOS 或非门、或门的多余的输入端的处理方法是 _____。

填空题参考答案:

1. 算术逻辑运算单元、触发器、寄存器、存储器、只读存储器、随机存储器、计数器、累加器 2. 69D、+69D、45D、‘E’ 3. $0 \sim 65535$ 、 $-32768 \sim +32767$ 、 $0 \sim 255$ 、 $-128 \sim +127$

4. 1、0 5. 触发器 6. 接高电平 7. 接地

二、选择题

1. 完整的计算机系统包括 ()。

- A. 硬件系统和软件系统 B. 主机和外部设备
C. 主机和应用程序 D. 运算器、存储器和控制器

2. 在计算机中, 表示地址通常使用 ()。

- A. 无符号数 B. 原码 C. 反码 D. 补码

3. 若某机器数为 $10000000B$, 它代表 0, 则它是 () 码形式, 它代表 -128, 则它是 () 形式。

- A. 补码 B. 原码 C. 反码 D. 原码或反码

4. 8 位二进制补码表示的整数数据范围为 ()。

A. - 128 ~ 127 B. - 127 ~ 127 C. - 128 ~ 128 D. - 127 ~ 128

5. 用 8 位二进制数表示“-0”的补码为 ()。

A. 10000000 B. 11111111 C. 00000000 D. 00000001

6. $[x]_{\text{原}} = 10111101\text{B}$, $[x]_{\text{反}} = 10111101\text{B}$, $[x]_{\text{补}} = 10111101\text{B}$ 三个数中 ()。

A. x_1 最小 B. x_2 最小 C. x_3 最小 D. $x_2 = x_1 = x_3$

7. “与非门”中的某一个输入值为“0”，那么它的输出值为 ()。

A. 为“0” B. 为“1”
C. 要取决于其他输入端的值 D. 取决于正逻辑还是负逻辑

8. 若门电路的两个输入量为“1”、“1”，输出量为“0”，不可能完成此功能的是 ()。

A. “异或”门 B. “与非”门 C. “或非”门 D. “与”门

9. 下列不正确的逻辑表达式是 ()。

A. $A + \bar{A}B = A + B$ B. $AB = \overline{A + B}$
C. $A(A + B) = A$ D. $(A + B)(A + \bar{B}) = A$

选择题参考答案：

1. A 2. A 3. B、A 4. A 5. C 6. C 7. B 8. D 9. B

三、简答题

1. 简述一下今后单片机的发展方向。

答：目前的单片机已经超过 500 多个品种，在各个方面都有了很大的发展。目前单片机正向着大容量、高性能，以及小容量、低价格，外围电路内装化的方向发展，而且应用领域也正在不断扩大。

2. 什么叫溢出？两个补码进行加法或减法运算时，什么情况下会产生溢出？举例说明。

答：两个带符号数进行运算，运算结果超出了数据的表示范围，则为溢出。两个同号数相加，当结果的符号位与参加运算的数的符号不同时，则为溢出；两个异号数相减，当结果的符号位与被减数的符号位不同时，则为溢出。

3. 三态输出电路有什么意义？

答：三态输出电路的意义在于能使电路和总线脱离，使总线具有公共通道的作用。

四、计算题

1. 将下列二进制数转换成十进制数：(1) 101101.10B (2) 111.111B

解：把二进制数位中数值是 1 的那些位的权值相加，其和就是等效的十进制数，由转换规则得：

(1) $101101.11\text{B} = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 41.75\text{D}$

(2) $111.111\text{B} = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 7.875\text{D}$

2. 已知 x 及 y ，试分别计算 $[x + y]_{\text{补}}$ 、 $[x - y]_{\text{补}}$ ，并指出是否产生溢出（设补码均用 8 位表示）。

$$(1) x = +1001110, y = +0010110$$

$$\text{解: } [x]_{\text{补}} = 01001110, [y]_{\text{补}} = 00010110$$

$$[-y]_{\text{补}} = [-0010110]_{\text{补}} = [10010110]_{\text{反}} + 1 = 11101001 + 1 = 11101010$$

$$[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} = 01001110 + 00010110 = 01100100\text{B} \text{ (无溢出, 结果正确)}$$

$$[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} = 01001110 + 11101010 = 100111000\text{B} = 00111000\text{B} \text{ (溢出, 自动}$$

丢, 结果正确)

$$(2) x = +0101101, y = -1100100$$

$$\text{解: } [x]_{\text{补}} = 00101101, [y]_{\text{补}} = [-1100100]_{\text{补}} = [11100100]_{\text{反}} + 1 = 10011011 + 1 = 10011100$$

$$[-y]_{\text{补}} = [(+1100100)]_{\text{补}} = 01100100$$

$$[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} = 00101101 + 10011100 = 11001001\text{B} \text{ (无溢出, 结果正确)}$$

$$[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} = 00101101 + 01100100 = 10010001\text{B}$$

(两个正数相加, 结果为负, 溢出, 结果错)

3. 一个 16 位二进制整数, 若采用补码表示, 由 5 个“1”和 11 个“0”组成, 则十进制表示的最小值是多少? 最大值是多少?

解: 根据补码的定义, 最小数应为 $[x]_{\text{补}} = 1000000000001111$, 它是一个负数, 十进制大小为: $x = [1000000000001111]_{\text{反}} + 1 = 1111111111110001 = -32753$, 最大数变为 $[x]_{\text{补}} = 0111111000000000$, 它是一个正数, 十进制大小为 $x = +31744$ 。

4. 完成下列 BCD 码运算 $58 + 39$, $58 - 39$ 。

解: 进行 BCD 码加法:

$$(1) \text{ 将十进制数用 BCD 码表示: } 58\text{D} = 0101\ 1000\text{BCD} \quad 39\text{D} = 0011\ 1001\ \text{BCD}_0$$

$$(2) \text{ 进行 BCD 码加法: } 0101\ 1000\ \text{BCD} + 0011\ 1001\ \text{BCD} = 1001\ 0001\ \text{BCD}_0$$

(3) 调整: 运算后低 4 位 = 0001, AF = 1, 低 4 位加 6; 运算后高 4 位 = 1001, CF = 0, 高 4 位加 0。于是: $1001\ 0001\text{BCD} + 0000\ 0110\ \text{BCD} = 1001\ 0111\ \text{BCD}$ 。所以 $58 + 39 = 97$ 。

进行 BCD 码减法:

$$(1) \text{ 将十进制数用 BCD 码表示: } 58\text{D} = 0101\ 1000\ \text{BCD} \quad 39\text{D} = 0011\ 1001\ \text{BCD}_0$$

$$(2) \text{ 进行 BCD 码减法: } 0101\ 1000\ \text{BCD} - 0011\ 1001\ \text{BCD} = 0001\ 1111\ \text{BCD}_0$$

(3) 调整: 运算后低 4 位 = 1111, AF = 1, 低 4 位减 6; 运算后高 4 位 = 0001, CF = 0, 高 4 位减 0。于是 $0001\ 1111\text{BCD} - 0000\ 0110\ \text{BCD} = 0001\ 1001\ \text{BCD}$ 。所以 $58 - 39 = 19$ 。

第二章 MCS—51 单片机的结构与原理

第一节 学习要求与内容概要

一、学习要求

- (1) 熟悉 MCS—51 系列单片机的内部结构和原理。
- (2) 重点掌握：1) MCS—51 系列单片机的组成与结构。2) 单片机引脚及其功能。3) 存储器配置及其特点，21 个特殊功能寄存器的功能。4) 振荡器和时钟电路、复位电路及其功能。
- (3) 了解内容：1) CPU 时序。2) 并行输入/输出端口 P0 ~ P3 的内部结构。3) 单片机基本工作系统组成。

二、MCS—51 系列单片机组成与结构学习内容概要

MCS—51 系列单片机主要由 1 个 8 位微处理器 (CPU)、片内数据存储器 (RAM)、片内程序存储器 (ROM 或 EPROM)、4 个 8 位并行 I/O 接口 P0 ~ P3、2 个定时器/计数器、5 个中断源的中断管理控制系统、1 个全双工串行 I/O 接口 UART 及片内振荡器与时钟产生电路等的基本部分组成。MCS—51 系列单片机芯片及制造工艺如表 2-1。

表 2-1 MCS—51 系列单片机芯片及制造工艺

ROM 型	无 ROM 型	EPROM 型	片内 ROM	片内 RAM	16 位定时器	制造工艺
8051	8031	8751	4KB	128B	2	HMOS
8051AH	8031AH	8751H	4KB	128B	2	HMOS
8052AH	8032AH	8752BH	8KB	256B	3	HMOS
80C51BH	80C31BH	87C51	4KB	128B	2	CHMOS

1. CPU

CPU 主要由运算器与控制器两部分组成，其中运算器包括：算术与逻辑运算单元 ALU、累加器 ACC、寄存器 B、状态寄存器 PSW 以及暂存器 TMP1 和 TMP2 等；而控制器包括：程序计数器 PC、指令计数器 IR、指令译码器 ID、振荡器及定时器等。

注意：用户不能直接对程序计数器 PC 和两个暂存器赋值。

2. 存储器

- (1) 程序存储器，用于存放程序和常数表格。
- (2) 数据存储器，用于存放运算数据和结果等。

3. 接口

8051 有 4 个 8 位并行口，即 P0 ~ P3，每个端口各有 8 条 I/O 线，它们都是双向端口。

三、单片机引脚及其功能学习内容概要

MCS-51 系列芯片通常有两种封装：即双列直插式封装和方形封装。本节重点掌握双列直插式封装形式，共 40 条引脚。大致分为电源（ V_{CC} 和 V_{SS} ），时钟（XTAL1 和 XTAL2），4 个输入/输出口 P0、P1、P2 和 P3（每一口各 8 条，共 32 条），控制信号线（RST，ALE， $\overline{PSEN}$ 和 $\overline{EA}$ ）。各引脚主要功能如下：

(1) XTAL1 和 XTAL2 单片机外接时钟信号的两个输入端。

(2) P0、P1、P2 和 P3 4 个并行口，均可以作为一般输入/输出口使用，其中在扩展系统中 P0 又可作为地址/数据线，P0 和 P2 构成 16 位地址总线（P0 为低 8 位地址，P2 为高 8 位地址），P3 还可作为第二功能用。

(3) RST 复位/备用电源线，主要功能是当从此引脚输入大于两个机器周期的高电平信号，就可以使 8051 处于复位（初始）状态，产生此信号的复位电路有两种：上电复位和手动按键复位。

(4) ALE/ $\overline{PROG}$ 地址锁存允许信号端。CPU 访问片外扩展存储器时，P0 口的低 8 位地址由 ALE 输出信号锁存到外部的地址寄存器。在无外部扩展存储器时，ALE 引脚以 1/6 振荡频率的固定速率输出脉冲信号，因而可用作外部时钟或定时。

(5) $\overline{PSEN}$ 程序存储器允许输出信号端。在主机访问片外程序存储器时，将程序计数器 PC 的 16 位地址输出到 P0 和 P2 口外部的地址寄存器后，该引脚输出负脉冲选通外部程序存储器。

(6) $\overline{EA}/V_{PP}$ 外部程序存储器地址允许输入端/固化编程电压输入端。 $\overline{EA}$ 引脚电平的高低决定了系统复位后 CPU 是从片内程序存储器还是从片外扩展存储器的 0000H 字节单元开始取指令执行。

四、单片机存储器配置学习内容概要

8051 存储器空间分布如图 2-1 所示。

1. 程序存储器空间

程序存储器空间用于存放程序和常数表格。

8031 内部没有 ROM 存储器，8051 和 8751 内部有 4KB ROM 存储器，地址范围为 0000H ~ 0FFFH。三者都可以外接外部 ROM，但片内和片外之和不能超过 64KB，因为程序计数器 PC 为 16 位，只能寻址 64KB 的地址空间。片内外程序存储器空间的应用由 $\overline{EA}$ 引脚来控制。对于 ROM（EPROM）的低位地址空间 0000H ~ 002AH，这些单元为预留单元，用于上电复位后引导程序地址及 5 个中断服务程序的入口地址。在实际应用系统中，主程序的存放一般是从 0030H 单元开始的。

2. 数据存储器空间

数据存储器空间用于原始数据及运算结果的存放。

8051 可以外扩最大为 64KB 的数据存储器空间，片外 RAM 与片外 I/O 设备统一编址，都利用 MOVX 指令进行存、取数操作。

8051 片内具有 128 个字节单元数据存储器 and 21 个字节的特殊功能寄存器，地址范围为 00H ~ 0FFH，分为低 128B（00H ~ 7FH）数据存储器区和高 128B（80H ~ 0FFH）的特殊功能寄存器（SFR）区，通过 MOV 指令访问。低 128B 的结构为：00H ~ 1FH 为 4 组（R0 ~ R7）32

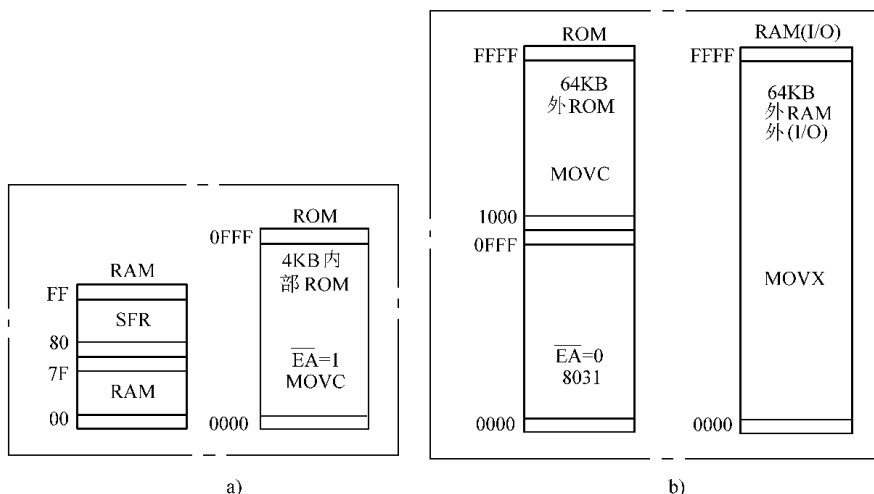


图 2-1 8051 存储器配置图

a) 片内存储器 b) 片外扩展存储器

个工作寄存器，20H ~ 2FH 为 16B 的位寻址区（128 位），30H ~ 7FH 为数据缓冲区，其中一部分区域可作为栈区来使用。

需要说明的几点：

(1) 对于工作寄存器 R0 ~ R7，其各自所对应的字节地址根据所属当前工作寄存器组而定。

(2) 位寻址区 20H ~ 2FH 的位地址为 00H ~ 7FH，可以通过不同的寻址方式和指令来区分某一地址是字节地址还是位地址，例如，对于下面两条指令：

MOV A, 20H ; 字节地址 20H 单元的内容送入累加器 A

MOV C, 20H ; 位地址 20H 单元的内容送入位累加器 C

(3) 一般数据存储区的一部分单元可以按照“后进先出”或“先进后出”规则来存放数据，即栈区，其数据的存取是通过栈指针 SP 来管理的。堆栈操作有压栈 (PUSH) 和出栈 (POP) 两种方式。栈顶指针 SP 复位状态为 07H，这样第一个进入栈区的数据地址应为 08H，而 08H 常用作工作寄存器区，因而开辟栈区时，常把 SP 的值进行修改，使栈区位于 30H ~ 7FH 中。

高 128B 为特殊功能寄存器区 (SFR)，并不是所有的 128 个单元都有定义，只有 21 个单元有定义，离散的分布在高 128B 内。

程序计数器 PC 不属于 21 个特殊功能寄存器，PC 不可寻址，即用户不能直接对其赋值。

五、振荡器与时钟电路及 CPU 时序学习内容概要

8051 内部有一个高增益反向放大器，用于构成振荡器，引脚 XTAL1 和 XTAL2 分别是此放大器的输入端和输出端，在 XTAL1 和 XTAL2 两端跨接晶体或陶瓷谐振器，就构成了稳定的自激振荡器，其发出的脉冲直接送入内部时钟电路。8051 也可使用外部振荡脉冲，外部时钟信号由 XTAL2 直接送至内部时钟电路。

1. CPU 时序的有关概念

(1) 振荡周期：由单片机片内振荡电路产生，常定义为时钟脉冲频率的倒数，是时序中最小的时间单位。

(2) 时钟周期：又称为状态周期 (S)，为振荡周期的 2 倍。

(3) 机器周期：MCS—51 系列单片机的 1 个机器周期由 6 个 S 状态周期组成，分为 S1P1 (状态 1 拍 1)，S1P2、S2P1、S2P2、S3P1、S3P2、S4P1、S4P2、S5P1、S5P2、S6P1 和 S6P2 共 12 个状态拍。

(4) 指令周期：执行一条指令所需要的时间，一般由若干个机器周期组成。MCS—51 系列单片机有单机器周期指令、双机器周期指令、四机器周期指令三种。

2. 时序

(1) 取指令/执行指令时序。单片机执行每一条指令分为取指令和执行指令两个阶段。单片机把程序计数器 PC 中的地址送到程序存储器，从中取出需要执行指令的操作码和操作数。指令执行是对指令操作码进行译码，以产生一系列控制信号完成指令的执行。

ALE 引脚上出现的信号是周期性的，每个机器周期内出现两次高电平，出现时刻为 S1P2 和 S4P2，持续时间为一个状态 S，ALE 信号每出现一次，CPU 就进行一次取指操作，根据不同指令的字节数和机器周期数，取指令操作有所差异。

(2) 访问片外 ROM/RAM 的指令时序。8051 有两类可以访问片外存储器的指令，一类是读片外 ROM 指令，另一类是访问片外 RAM 指令。这两类指令执行时所产生的时序除与 ALE 引脚有关，还与 P0、P2、 $\overline{\text{PSEN}}$ 和 $\overline{\text{RD}}$ 有关。

六、并行输入/输出端口 P0 ~ P3 学习内容概要

1. 并行输入/输出端口 P0 ~ P3 的功能

每个端口都是 8 位准双向口，共占 32 只引脚。每一条 I/O 线都能独立地用做输入或输出线。单片机系统具有片外扩展存储器时，P2 口传送高 8 位地址，P0 作为数据线与低 8 位地址线分时复用。P3 口还具有第二功能。

2. 并行输入/输出端口 P0 ~ P3 的结构

四个并行输入/输出端口 P0 ~ P3 的结构是基本相同的，但又有所差别。

相同点：每个端口都包括一个锁存器（即特殊功能寄存器 P0 ~ P3）、一个输出驱动器和输入缓冲器。

不同点：

(1) P0 既可作 I/O 端口使用，也可作地址/数据总线使用。当把它作通用口输出时，输出级是开漏电路，只有外接上拉电阻，才有高电平输出；把它作地址/数据总线时，无需外接电阻，此时不能再作 I/O 口使用。

(2) P1 ~ P3 口输出级接有内部上拉负载电阻，所以不必再外接上拉电阻。

(3) P0、P2 口的位结构比 P1 口多了一个转换控制部分。当 P0、P2 口作通用 I/O 口时，各自的多路开关 MUX 分别倒向锁存器输出 $\overline{Q}$ 和 Q 端，构成输出驱动电路。

在系统扩展片外存储器时，P0 口输出低 8 位地址，还分时作数据线使用；P2 口输出高 8 位地址。

(4) P3 口除了具有准双向 I/O 口的功能之外，各引脚还具有第二功能。在结构上，由 P3 口某一位的第二功能控制端来控制。

3. 并行输入/输出端口 P0 ~ P3 的端口负载能力

P0 口的每一位输出可驱动 8 个 TTL 负载；P1 ~ P3 口的输出级接有内部上拉负载电阻，每一位输出可驱动 4 个 TTL 负载。

4. 并行输入/输出端口 P0 ~ P3 的接口要求

P0 ~ P3 口都是准双向 I/O 口，作输入时，必须先对相应端口锁存器写“1”，使驱动管 FET 截止。

七、单片机基本工作系统组成学习内容概要

本节以应用 8051 单片机驱动步进电动机为例，了解单片机的作用和外部特性。

1. 系统基本组成

- (1) 8051 单片机。
- (2) 程序存储器 2764，8KB EPROM 型程序存储器芯片。
- (3) 74LS373，锁存 CPU P0 口输出的低 8 位地址信息。
- (4) 步进电动机驱动电路。

2. 引脚连接

- (1) 8051 的 XTAL1 和 XTAL2 连接石英晶体和两个微调电容，为系统提供振荡时钟。
- (2) RST 引脚采用由电阻、电容构成的上电复位电路。
- (3) $\overline{\text{EA}}$ 引脚接地，因 8051 的片内 ROM 没有用。
- (4) 13 位地址线由 P0 和 P2 口的部分位线构成；数据线由 P0 口构成。
- (5) ALE 与 74LS373 的锁存控制端 LE 相连，控制 P0 口低 8 位地址锁存； $\overline{\text{PSEN}}$ 与 2764 的输出允许控制端 $\overline{\text{OE}}$ 相连，以控制 2764 输出数据（指令）。

第二节 原教材习题及解答

2-1 8051 单片机芯片内部包含哪些主要逻辑功能部件？

答：8051 单片机芯片内部包含 1 个 8 位微处理器（CPU），片内数据存储器 RAM，片内程序存储器 ROM，4 个 8 位并行 I/O 口 P0 ~ P3，2 个定时器/计数器，5 个中断源的中断控制系统，1 个 UART（通用异步接收发送器）的串行 I/O 口，片内振荡器和时钟产生电路。

2-2 8051 存储器分哪几个地址空间？如何区别不同空间寻址？

答：

- (1) 片内、外统一编址的 64KB 程序存储器；
- (2) 片内 128B 数据存储器和 21 个特殊功能寄存器；
- (3) 片外 64KB 数据存储器。

硬件： $\overline{\text{EA}}$ 引脚接高电平时，CPU 从片内 0000H 单元开始取指令，接低电平时 CPU 直接访问片外 EPROM。

软件：MOV 指令访问片内数据存储器，MOVX 指令访问片外数据存储器，MOVC 指令用于读取程序存储器中的常数。

2-3 简述 8051 片内 RAM 的空间分配。

答：低 128B 数据存储器区分为：工作寄存器区（00H ~ 1FH）、位寻址区（20H ~ 2FH）

和一般数据存储区 (30H~7FH)。

高 128B 数据存储区离散地分布着 21 个特殊功能寄存器。

2-4 简述直接位寻址区的空间分配, 片内 RAM 中包含哪些可位寻址单元?

答: MCS-51 单片机片内 RAM 中有两个区域可进行位寻址:

(1) 片内 RAM 低 128B 的位寻址区, 地址为 20H~2FH 的 16 个字节单元共 128 位, 每一位都有相应的位地址, 可用位寻址方式对其进行置位、复位、内容传送、逻辑运算等操作。128 位的位地址定义为 00H~7FH。

(2) 片内 RAM 高 128B 的存储器区, 有 21 个特殊功能寄存器。其中字节地址正好能被 8 整除的字节单元中的每一位都可以按位寻址、操作。

2-5 8051 单片机的 P0~P3 口结构有何不同, 用做通用 I/O 口输入数据时应注意什么?

答: P0~P3 口功能不完全相同。

(1) 访问外扩展存储器时, P2 口送出 16 位地址的高 8 位地址, P0 口先送出 16 位地址的低 8 位地址, 再做数据的输入/输出通道。

(2) 在无片外扩展存储器的系统中, P0、P2 口不需要做地址口时, 和 P1、P3 口一样, 都可作为准双向通用 I/O 口使用。P0 的输出级无上拉电阻, 在作为通用 I/O 口时需外接上拉电阻, 且 P0 口的每一位输出可驱动 8 个 TTL 负载, 而 P0~P3 口的输出级接有上拉负载电阻, 每一位输出可驱动 4 个 TTL 负载。

在作通用 I/O 输入数据时应注意: 应先对相应的端口锁存器写 1, 以防止误读。

2-6 8051 单片机 $\overline{\text{EA}}$ 引脚有何功能? 在使用 8031 时, $\overline{\text{EA}}$ 引脚应如何处理?

答: $\overline{\text{EA}}$ 引脚为外部程序存储器地址允许输入端, 其电平的高低决定了系统复位后 CPU 是从片内程序存储器还是片外扩展存储器的 0000H 字节单元开始取指令。

(1) 当引脚 $\overline{\text{EA}}$ 接成高电平时, CPU 首先从片内 0000H 字节单元开始取指令执行程序。当指令地址寄存器 PC 中的内容超过 0FFFH 后, 就自动转向片外扩展的 EPROM 中取指令执行, 这时芯片外部的重叠地址为 0000H~0FFFH 的低 4KB EPROM 忽略不用。

(2) 当引脚 $\overline{\text{EA}}$ 接成低电平时, 复位后 CPU 直接从片外 EPROM 的 0000H 字节单元开始取指令执行, 这时芯片内部 0000H~0FFFH 的 4KB 单元被忽略不用。

在使用 8031 时, 由于其内部没有 ROM/EPROM, 所以必须使用外部扩展程序存储器, 它的 $\overline{\text{EA}}$ 引脚应该接成低电平, CPU 直接从片外扩展的 EPROM 中取指令执行。

2-7 8051 单片机有哪些信号需要芯片引脚以第二功能的方式提供?

答:

(1) VPD (9 脚): 片内 RAM 备用电源的输入端。

(2) $\overline{\text{PROG}}$ (30 脚): 对于片内带有可编程序存储器的芯片, 作为编程写入的编程脉冲输入端, 把编写好的程序指令代码存入程序存储器中。

(3) VPP (31 脚): 对于片内有 EPROM 的芯片固化程序时, 它作为加高编程电压的输入端。

(4) RXD (10 脚): 串行口通信数据输入线。

(5) TXD (11 脚): 串行口通信数据输出线。

(6) $\overline{\text{INT0}}$ (12 脚): 外部中断 0 输入信号。

(7) $\overline{\text{INT1}}$ (13 脚): 外部中断 1 输出信号。

(8) T0 (14 脚)：定时器 0 的脉冲信号外部输入。

(9) T1 (15 脚)：定时器 1 的脉冲信号外部输入。

(10) $\overline{\text{WR}}$ (16 脚)：CPU 至片外数据存储器的“写选通控制”输出。

(11) $\overline{\text{RD}}$ (17 脚)：CPU 至片外数据存储器的“读选通控制”输出。

2-8 内 RAM 低 128B 单元划分为哪三个主要区域？各区域的主要功能是什么？

答：地址范围为 00H ~ 7FH 的 128B 单元，按功能特点划分为不同的几个区，分述如下。

(1) 工作寄存器区 地址范围为 00H ~ 1FH 的 32B 存储器单元，分为 4 个组，每组有 8 个工作寄存器字节单元，定名为 R0、R1、R2、R3、R4、R5、R6 和 R7。通过改变程序状态字特殊功能寄存器 PSW 中 RS1、RS0 两位的内容，每组工作寄存器均可选作 CPU 的当前工作寄存器组。

(2) 位寻址区 地址为 20H ~ 2FH 的 16 个字节单元中共 128 位，每一位都有相应的位地址，可用位寻址方式对其进行置位、复位、内容传送、逻辑运算等操作。

(3) 一般数据存储器区 地址范围为 30H ~ 7FH 的区域，可用于原始数据及运算结果的存储。值得注意的是，30H ~ 7FH 中一部分字节单元可以开辟出一块遵守“后进先出”或“先进后出”规则的特殊数据区——堆栈。

2-9 使单片机系统复位有哪几种方法？复位后特殊功能寄存器初始状态如何？

答：使单片机系统复位有两种方法：上电自动复位和按键手动复位。

系统复位的主要功能就是把 PC 初始化为 0000H，使单片机从 0000H 单元开始执行程序。另外，系统复位还使一些特殊功能寄存器恢复到复位状态。

PC : 0000H	ACC : 00H	PSW : 00H
SP : 07H	DPTR : 0000H	P0 ~ P3 : FFH
IP : * * 000000B	IE : 0 * 000000B	TMOD : 00H
TCON : 00H	TL0 : 00H	TH0 : 00H
TL1 : 00H	TH1 : 00H	SCON : 00H
SBUF : 不定	PCON : 0 * * * 0000B	

2-10 开机复位后，CPU 使用哪组工作寄存器作为当前工作寄存器？它们的地址是什么？

答：开机复位后，由于 (RS1 = 0, RS0 = 0)，CPU 使用第 0 组工作寄存器作为当前工作寄存器，它的地址为 00H ~ 07H。

2-11 CPU 如何确定和改变当前工作寄存器组？

答：通过改变程序状态字特殊功能寄存器 PSW 中 RS1、RS0 两位的内容，每组工作寄存器均可被选作当前工作寄存器。

RS1	RS0	当前工作寄存器
0	0	第 0 组
0	1	第 1 组
1	0	第 2 组
1	1	第 3 组

2-12 程序状态寄存器 PSW 的作用是什么？有哪些常用标志位？作用是什么？

答：程序状态寄存器 PSW 为特殊功能寄存器，它的各位包含了程序执行后的状态信息，

供程序查询或判别之用，其各位的含义如下：

CY：进位标志位；AC：半进位标志位；F0：用户标志位；RS1，RS0：工作寄存器组选择控制位；P：奇偶校验位；OV：溢出标志位。

2-13 位地址与字节单元地址 7CH 如何区别？位地址 7CH 具体在片内 RAM 的什么位置？

答：8051 采用不同的寻址方式和指令分别对位和字节进行不同的操作，如 SETB 20H 指的是 20H 位单元的内容置为 1，而 MOV 20H, #0FFH 指的是片内 RAM20H 字节单元的内容变为 FFH。

位地址 7CH 在片内 RAM 低 128B 的 2FH 字节单元的第四位。

2-14 8051 单片机时钟周期与振荡周期之间有什么关系？

答：8051 单片机的时钟周期为振荡周期的两倍。

2-15 一个机器周期的时序是怎样划分的？

答：MCS-51 系列单片机工作的一个机器周期由 6 个 S 状态周期组成，即 S1 ~ S6，每个状态又持续两个振荡周期，即 P1 和 P2 两拍，故一个机器周期包括 12 个振荡周期。一个机器周期具体分为 S1P1、S1P2、S2P1、S2P2、S3P1、S3P2、S4P1、S4P2、S5P1、S5P2、S6P1、S6P2，共 12 状态拍。若采用 12MHz 晶体振荡器，则每个机器周期恰为 $1\mu\text{s}$ 。

2-16 什么叫堆栈？堆栈指针 SP 的作用是什么？8051 单片机堆栈容量不能超过多少字节？

答：一般在 30H ~ 7FH 的地址范围内，有一部分字节单元可开辟出一块遵守“后进先出”或“先进后出”规则的特殊数据区，就叫堆栈。

SP 为堆栈指针，它始终存放堆栈顶部字节单元的地址（栈顶指针）。数据压栈时，SP 首先自动加 1，然后数据压入 SP 内容指示的字节单元中；而数据出栈时，SP 内容指示的字节单元中的数据弹出，然后 SP 自动减 1，这样 SP 始终指向堆栈顶部字节单元的地址。

8051 单片机堆栈容量最多不能超过 128B。初始化后，(SP) = 07H，如果 CPU 使用工作寄存器一组，那么堆栈和工作寄存器将发生冲突，因此常把堆栈移至 30 ~ 7FH 区域之中。

2-17 分析并说明原教材中图 2-16 系统中各信号线的作用。

答：P1 口用作一般 I/O 口，P1.0、P1.1、P1.2 通过光电隔离电路向三相步进电动机的功率放大电路顺序地发出步进脉冲信号；P1.6 和 P1.7 作为输入，检测控制步进电动机正反转的按键状态；XTAL1 和 XTAL2 接外部时钟电路，为系统提供振荡时钟；P0 口分时用作地址/数据总线，取指令时输出低 8 位地址，经地址锁存器 74LS373 锁存，其他时间 P0 口用作数据总线输出 8 位数据；高 5 位地址由 P2.0 ~ P2.4 提供；ALE 作为地址输出锁存控制端； $\overline{\text{EA}}$ 接地，CPU 直接从外扩 EPROM 中取指令执行； $\overline{\text{PSEN}}$ 为程序存储器输出允许控制端，在 CPU 读取 EPROM 中数据时，使 EPROM 输出指令。

第三节 增选习题及解答

一、填空题

1. 堆栈寄存器的符号是_____，复位后它的内容为_____。
2. 累加器 A 的内容有偶数个 1 时，特殊功能寄存器_____的_____位内容为_____。

3. 内 RAM 的 03H 字节单元的地址符号另记为_____。
4. 堆栈操作遵循的原则是_____。
5. 8051 单片机有 4 个 I/O 口，它们分别是_____，复位后对应它们的寄存器的内容为_____。
6. 为了正确地读取 8051 的 P1 口引脚的状态值，应该先向它_____。
7. 8051 的_____引脚应该至少保持_____个机器周期的_____电平，才能使 8051 复位。
8. 8051 的封装形式为_____。
9. 利用 8051 组成的工作系统，程序放在内部 ROM，它的 $\overline{\text{EA}}$ 引脚应该接_____电平。
10. 8051 单片机的 ALE 引脚在 CPU 对外存非操作期间，它输出频率为晶体振荡器频率_____的脉冲信号。
11. 8051 内部程序存储器的容量为_____字节，8031 的内部程序存储器的容量为_____字节。
12. 8031 最大可以扩展_____程序存储器，最大扩展的数据存储器的容量是_____。
13. 位寻址区在内 RAM 低 128B 单元的_____字节中，高 128B 单元中有_____个特殊功能寄存器的位可以进行位操作。
14. 当前工作寄存器的选择由特殊功能寄存器_____的_____位的内容决定。
15. 用于位处理的布尔处理器是借用的特殊功能寄存器_____的_____位完成的。
16. 一个机器周期包括_____个振荡周期，具体分为_____个状态周期。
17. 按执行时间，8051 的指令分为_____周期、_____周期和_____周期指令共三种。
18. 单片机由 CPU、存储器和_____三部分组成。
19. CPU 由_____和_____两部分组成。
20. 若不使用 MCS-51 片内程序存储器，引脚 $\overline{\text{EA}}$ 必须_____。
21. 当 MCS-51 引脚 ALE 信号有效时，表示从 P0 口稳定地送出了_____地址。
22. 当 MCS-51 的 P0 口作为输出端口时，每位能驱动_____个 TTL 负载。
23. MCS-51 有 () 个并行 I/O 口，由于是准双向口，所以在输入时必须先_____。
24. MCS-51 中凡字节地址能被_____整除的特殊功能寄存器均能位寻址。
25. MCS-51 系统中，当 $\overline{\text{PSEN}}$ 信号有效时，表示 CPU 要从_____读取信息。
26. MCS-51 有_____组工作寄存器，它们的地址范围是_____。
27. MCS-51 单片机内部 RAM 中的位寻址区位于低 128B 中的_____地址单元内。
28. 经过扩展外部程序存储器的 8031，若 PC 从程序存储器 0000H 开始执行，则 $\overline{\text{EA}}$ 引脚必须接_____电平。
29. 当 $(\text{RS1}, \text{RS0}) = (0, 1)$ 时，工作寄存器 R7 对应内部 RAM 中的_____字节单元。
30. 设计 8031 系统时，_____口不能用作一般 I/O 口。

31. 单片机复位后, ACC 为 _____, PSW 为 _____, SP 为 _____, P0 ~ P3 为 _____, PC 为 _____。
32. 8051 内部含 _____ ROM, 8751 是 EPROM 型, 内含 4KB EPROM。总的来讲, MCS—51 系列单片机的存储器配置在物理上分为 () 个独立的存储器空间, 在逻辑上分为 _____ 个存储器空间。
33. 单片机系统复位后, (PSW) = _____, 因此片内 RAM 寄存区的当前寄存器是第 _____ 组, 8 个寄存器的单元地址为 _____。
34. 在 MCS—51 中 PC 和 DPTR 都用于传送地址, 但 PC 是为访问 _____ 存储器传送地址, 而 DPTR 是为访问 _____ 存储器传送地址。
35. 在位操作中, 与字节操作中的累加器 ACC 作用相同的是 _____。
36. 若 MCS—51 单片机使用频率为 6MHz 的晶振, 则一个机器周期为 _____, 指令周期为 _____。
37. 8051 复位后, CPU 从 _____ 单元开始执行程序。SP 的内容为 _____, 如果不对 SP 重新赋值, 压入堆栈的第一个数据将位于片内 RAM 的 _____ 单元。
38. MCS—51 系列单片机最大的寻址范围为 _____。
39. 某时钟频率为 8MHz, 若某条指令为双机器周期指令, 则这条指令的执行时间为 _____。
40. MCS—51 单片机的扩展系统中, 使用锁存器的目的是 _____。

填空题参考答案:

1. SP, 07H 2. PSW, 0, 0 3. R3 4. 先进后出或后进先出 5. P0 ~ P3, FFH 6. 写“1” 7. RST, 两个, 高 8. 双列直插式 9. 高 10. 六分频 11. 4KB, 00B 12. 64KB, 64KB 13. 20H ~ 2FH, 11 14. PSW, RS1 和 RS0 15. PSW, CY 16. 12, 6 17. 单、双、四 18. 接口 19. 运算器和控制器 20. 接地 21. 低 8 位 22. 8 23. 四, 写“1” 24. 8 25. 程序存储器 26. 4 组, 00H - 1FH 27. 20H ~ 2FH 28. 低 29. 0FH 30. P0、P2 31. 00H, 00H, 07H, FFH, 0000H 32. 4KB 掩膜, 四, 三 33. 00H, 0, 00H ~ 07H 34. 程序, 数据 35. CY 36. $2\mu\text{s}$, $2\mu\text{s}$ 或 $4\mu\text{s}$ 或 $8\mu\text{s}$ 37. 0000H, 07H, 08H 38. 64KB 39. $3\mu\text{s}$ 40. 锁存 P0 口送出的低 8 位地址

二、选择题

1. 当使用 8751 且 $\overline{\text{EA}} = 1$, 程序存储器地址小于 1000H 时, 访问的是 ()。
- A. 片内 ROM B. 片外 ROM C. 片内/外 ROM D. 不定
2. 堆栈操作遵循的原则是 ()。
- A. 先进后出 B. 先进先出 C. 后进后出 D. 随机
3. 欲访问 8051 单片机的内部程序存储器, 则 $\overline{\text{EA}}$ 引脚必须为 ()。
- A. 高电平 B. 低电平 C. 高低电平 D. 与 PC 值有关
4. MCS—51 中, 一个机器周期由 () 个时钟周期组成。
- A. 4 B. 8 C. 6 D. 12
5. 若用数据指针 DPTR 作为间址寄存器访问外部数据存储器, 它可访问的整个外部数据存储器的空间为 ()。

- A. 4KB B. 8KB C. 16KB D. 64KB
6. MCS-51 系列单片机复位后其 P1 口的输出状态为 ()。
- A. 00H B. 0FFH C. 0F0H D. 不定
7. 当 MCS-51 复位时, 下面说法正确的是 ()。
- A. PC = 0000H B. SP = 00H C. SBUF = 00H D. P0 = 00H
8. 使用 8751, 且 $\overline{EA} = 1$ 时, 则可以外扩 ROM ()。
- A. 64KB B. 60KB C. 58KB D. 56KB
9. 下面部件不属于 CPU 结构中控制器部分的是 ()。
- A. PC B. IR C. PSW D. ID
10. P1 口的每一位能驱动 ()。
- A. 2 个 TTL 负载 B. 4 个 TTL 负载
C. 8 个 TTL 负载 D. 6 个 TTL 负载
11. PC 中存放的是 ()。
- A. 下一条要执行的指令的地址 B. 当前正在执行的指令
C. 当前正在执行指令的地址 D. 下一条要执行的指令
12. 8031 是 ()
- A. CPU B. 微处理器 C. 单片微型计算机 D. 控制器
13. 由 PC 内容指定 ROM 地址单元中取出的指令, 放于 ()。
- A. 运算器与控制器 B. 运算器与存储器
C. 指令寄存器和指令译码器 D. 控制器与存储器
14. 所谓 CPU 是指 ()。
- A. 运算器与控制器 B. 运算器与存储器
C. 输入输出设备 D. 控制器与存储器
15. 关于 MCS-51 的堆栈操作, 正确的说法是 ()。
- A. 先入栈, 再修改栈指针 B. 先修改栈指针, 再出栈
C. 先修改栈指针, 再入栈 D. 以上都不对
16. 指令寄存器的功能是 ()。
- A. 存放指令地址 B. 存放当前正在执行的指令
C. 存放 PC 的内容 D. 存放指令地址及操作数
17. MCS-51 外扩存储器时, 系统的数据总线是由 () 构成。
- A. P0 B. P1 C. P2 D. P3
18. MCS-51 有两种读取并行 I/O 口信息的方法: 一种是读引脚, 还有一种是 ()。
- A. 读锁存器 B. 读缓冲器 C. 读累加器 A D. 读存储器
19. MCS-51 的并行 I/O 口读 - 改 - 写操作, 是针对该口的 ()。
- A. 引脚 B. 缓冲器 C. 地址线 D. 内部锁存器
20. 假定设置堆栈指针 SP 的值为 36H, 在执行指令 PUSH 20H 后 SP 的值为 ()。
- A. 36H B. 37H C. 38H D. 21H

选择题参考答案:

1. A 2. A 3. A 4. C 5. D 6. B 7. A 8. B 9. C 10. B 11. A 12. C 13. C

14. A 15. C 16. B 17. A 18. A 19. D 20. B

三、简答题

1. 什么是微处理器？

答：微处理器是微型计算机的核心部件，又称为中央处理单元 CPU。它包括两个主要部分：运算器、控制器。

2. 8051 单片机在片内有哪些主要逻辑功能部件？每个逻辑部件的主要功能是什么？

答：(1) CPU (中央处理器)：8 位。

功能：中央处理器 CPU 由控制器与运算器构成，是单片机的控制和指挥中心。

(2) 片内 128B 数据存储器 and 21 个特殊功能寄存器。

功能：地址为 00H ~ 7FH 的低 RAM 区，用于数据暂存和数据缓冲等；地址为 80H ~ FFH 的特殊功能寄存器区，只有离散分布 21 个特殊功能寄存器 SP、DPTR、PCON、…、IE、IP、P0、P1、P2、P3，用以存放相应功能部件的控制命令、状态或数据。

(3) 程序存储器：4KB。

功能：程序存储器用于存放经调试正确的应用程序和表格之类的固定常数。

(4) 并行 I/O 口：8 位，4 个。

功能：8051 单片机的并行 I/O 口，是单片机数据信息或控制信号进、出的通道，不仅可以作为一般输入或输出口使用，而且在扩展外部存储器和 I/O 接口时，还可作为系统地址、数据总线或是控制信号线等。

(5) 串行接口：全双工，1 个。

功能：全双工串行 I/O 口，使单片机具有串行通信的能力。

(6) 定时器/计数器：16 位，2 个。

功能：在单片机的实际应用中，定时器/计数器提供精确的定时，或者对外部事件进行计数。

(7) 片内振荡器和时钟电路：1 个。

功能：8051 的这部分电路只要外接振荡元件，如晶体振荡器或外部时钟源，就能产生计算机工作所需要的时钟信号。

3. 程序计数器 PC、数据指针 DPTR、堆栈指针 SP 分别有哪些特点？

答：(1) 程序计数器 PC 的特点 程序计数器 PC 存放着下一条将从程序存储器中取出的指令的地址。程序计数器 PC 的数据长度决定了程序存储器可以直接寻址的范围。

程序计数器 PC 不可寻址，即不能对其赋值，其内容变化方式有：

1) 自动加 1，此为最基本的变化方式。

2) 执行条件或无条件转移指令时，其内容由转移指令直接修改，改变原程序执行方向。

3) 执行调用指令或响应中断时，PC 的内容变化过程如下：

PC 的当前值，即下一条将要执行的指令的地址送入堆栈，加以保护；将子程序的入口地址或者中断矢量地址送入 PC，改变原程序执行方向，执行子程序或中断服务程序；子程序或中断服务程序执行完毕，遇到返回指令 RET 或 RETI 时，将栈顶的内容送到 PC 中，程序又返回到原来的地方，继续执行主程序。

(2) 数据指针 DPTR 的特点 数据指针 DPTR 是作为片外数据存储器寻址用的地址寄存

器。

(3) 栈顶指针 SP 的特点 SP 始终存放堆栈顶部字节单元的地址。其内容变化是自动管理。进行堆栈操作有两种方式,即压栈 (PUSH) 和出栈 (POP)。压栈时, SP 首先自动加 1, 然后数据压入 SP 内容指示的字节单元中;而出栈时, SP 内容指示的字节单元中的数据弹出, 然后 SP 自动减 1, 这样 SP 始终指向堆栈顶部字节单元的地址。

(4) 程序计数器 PC 和数据指针 DPTR、栈顶指针 SP 的异同

1) 相同点:三者都是与地址有关的地址寄存器,其中 PC 与程序存储器的地址有关,而 DPTR 与数据存储器的地址有关,在对程序存储器进行变址间接寻址时, DPTR 作为基址寄存器使用, SP 则与栈区的地址有关。

PC 和 DPTR 都是 16 位的地址寄存器,其内容都是通过 P0 和 P2 口输出的, PC 的输出与 ALE 及 $\overline{\text{PSEN}}$ 有关, DPTR 与 ALE、 $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ 有关。

2) 不同点: PC 只能作为一个 16 位的寄存器, 自动加 1; SP 是一个 8 位的寄存器; DPTR 可以作为一个 16 位的寄存器, 也可作为两个 8 位的寄存器使用。

例如: MOV DPTR, # 2050H ; 可用以下两条指令来代替, 即 MOV DPH, # 20H ; MOV DPL, # 50H。

PC 是不可以访问的, 即用户不能对其赋值。而 DPTR 和 SP 均可以访问。

4. 复位的作用是什么? 画出 8031 单片机手动及自动复位电路图, 并说明原理。

答: (1) 复位是单片机的初始化操作, 单片机在启动运行时, 都需要先复位。它的作用是使 CPU 和系统中其他部件都处于一个确定的初始状态, 并从这个状态开始工作。

(2) 复位方法有两种, 即上电自动复位和手动按键复位。电路图及原理参考原教材第二章/第二节 MCS-51 单片机引脚及其功能。

5. 8051 单片机的存储器在结构上有何特点? 在物理上和逻辑上各有哪几种地址空间? 访问不同空间的指令格式有何区别?

答: (1) 8051 单片机存储器在结构上是采用哈佛型结构, 即将程序和数据分别放在程序存储器和数据存储器内。

(2) 其存储空间在物理结构上可划分为如下四个空间: 片内程序存储器、片外程序存储器、片内数据存储器、片外数据存储器。

从用户使用的角度, 即从逻辑上, 又可划分为如下三个空间:

1) 片内、外统一编址的 64KB 程序存储器 (地址范围为 0000H ~ FFFFH, 用 16 位地址表示), 访问时采用 MOVC 指令。

2) 片内 128B 数据寄存器和 21 个特殊功能寄存器 (地址范围为 00 ~ FFH, 用 8 位地址表示)。访问这部分空间采用 MOV 指令。

3) 片外 64KB 数据存储器 (地址范围为 0000H ~ FFFFH, 用 16 位地址表示)。访问时则利用 MOVX 指令。

6. 8051 单片机的通用工作寄存器组有什么特点? 如何实现工作寄存器现场保护?

答: 通用工作寄存器组的特点是: 用寄存器直接寻址, 指令的数量最多, 均为单周期指令, 执行速度快。

对于工作寄存器的现场保护, 一般在主程序中使用一组工作寄存器; 而在进入子程序或中断服务程序时, 切换到另一组工作寄存器; 在返回主程序前, 再重新切换回原来的工作寄

寄存器。

7. 8051 单片机的程序存储器中存放什么数据？有哪些操作方式？

答：程序存储器存放系统应用程序和表格常数。

程序存储器的操作方式有：

1) 程序指令的自主操作。程序由程序计数器 PC 指引进行顺序操作或转移操作。

2) 表格常数的查表操作。8051 单片机的指令系统中具有查表指令为 MOVC，通过 PC 或 DPTR 的基址 + 变址寻址方式实现查表操作。

8. 为什么说 8051 单片机具有很强的布尔（位）处理功能？可以位寻址的单元有哪些？采用布尔处理有哪些优点？

答：因为：

1) 累加器 CY：进位/借位标志位。在布尔运算中，CY 用于存放运算结果或数据源，是位处理时数据传送中的中心。

2) 位寻址区：内部数据 RAM 区的 20H ~ 2FH 的 16 个字节单元，共包含 128 位（00H ~ 7FH），是可位寻址的 RAM 区。11 个可位寻址的特殊功能寄存器的相应位。

3) 位操作指令系统：位操作指令可实现对位的置位、清 0、取反、判位转移、传送、位逻辑运算等操作。

采用布尔处理方法的优点是：利用位逻辑操作功能进行随机逻辑设计，可把逻辑表达式直接变换成软件执行，方法简便，免去了过多的数据往返传送、字节屏蔽和测试分支，大大简化了编程，节省存储空间，加快了处理速度，还可以实现复杂的组合逻辑处理功能。所有这些特别适用于某些数据采集、实时测控等应用系统。

9. MCS—51 系列单片机的程序存储器中有几个具有特殊功能的单元？各自功能是什么？

答：MCS—51 系列单片机的程序存储器中有 7 个保留特殊功能单元，其中 0000H 为复位入口地址、0003H 为外部中断 0 服务程序入口地址、000BH 为 T0 溢出中断入口、0013H 为外部中断 1 服务程序入口地址、001BH 为 T1 溢出中断入口、0023H 为串行接口中断入口、002BH 为 8052 中断服务程序入口地址。

10. 堆栈的作用是什么？在编程时，为什么有时要对堆栈指针 SP 重新赋值？如果 CPU 在操作中要使用第 0、1 两组工作寄存器，SP 的初值应为多大？

答：堆栈的主要作用是暂时存放数据和地址，通常用来保护断点和现场。

堆栈由 8 位特殊功能寄存器 SP 自动管理，它始终存放堆栈顶部字节单元的地址（栈顶指针）。进行堆栈操作有两种方式，一种叫数据压入（PUSH）堆栈，另一种叫数据弹出（POP）堆栈。数据压栈时，SP 首先自动加 1，然后数据压入 SP 内容指示的字节单元中；而数据出栈时，SP 内容指示的字节单元中的数据弹出，然后 SP 自动减 1，这样 SP 始终指向堆栈顶部字节单元。系统复位后，SP 初始化为 07H，使得堆栈事实上由 08H 开始。因为 08H ~ 1FH 单元为工作寄存器区，20H ~ 2FH 为位寻址区。在程序设计中很可能要用到这些区，所以用户在编程时要对堆栈指针 SP 重新赋值，最好把 SP 初值设为 2FH 或更大值，当然同时还要顾及允许的堆栈深度。在使用堆栈时要注意，由于堆栈的占用，会减少内部 RAM 的可利用单元，如设置不当，可能引起内部 RAM 单元冲突。

如果 CPU 在操作中要使用第 0、1 两组工作寄存器，SP 的初值应不小于 0FH。

11. MCS—51 系列单片机运行出错或程序进入死循环，如何摆脱困境？

答：当由于程序运行或操作错误使系统处于死锁状态时，需按复位键以重新启动。

12.8051 单片机的 4 个 I/O 端口的作用是什么？80C51 的片外三总线是如何分配的？

答：8051 单片机有 4 个 I/O 端口，每个端口都是 8 位双向口，共占 32 根引脚。

在无片外扩展存储器的系统中，这 4 个端口的每一位都可以作为双向通用 I/O 端口使用。需要注意的是，P0 ~ P3 口都是准双向 I/O 口，作输入时，必须先对相应端口锁存器写“1”，使驱动管 FET 截止。

在具有片外扩展存储器的系统中，P2 口作为高 8 位的地址线，P0 口分时作为低 8 位地址线和双向数据总线，P3 口的某些第二功能作为控制线。

13.8051 单片机的 I/O 口有什么特点？读 I/O 口有几种方法？

答：8051 单片机的每一条 I/O 线都能独立地用做输入或输出线。每个端口都包括一个锁存器（即特殊功能寄存器 P0 ~ P3），一个输出驱动器和输入缓冲器。输出时数据可以锁存，输入时数据可以缓冲。同时，I/O 口又是外部的输入/输出通道，必须有相应的引脚，故形成了 I/O 口的锁存器加引脚的典型结构。

读 I/O 口有两种方法，一是读锁存器，二是读引脚。

读锁存器操作：读锁存器是读入一个口锁存器的值，进行处理后，重新写回到锁存器中，这些指令称为读 - 修改 - 写指令。这些指令是一些逻辑运算指令、置位/清除指令、条件转移指令以及将 I/O 口作为目的地址的操作指令。

读引脚的操作：如果在指令中，某个 I/O 口被指定为源操作数，则该指令为读引脚的操作指令。例如：执行“MOV A, P1”时，P1 口的引脚状态传送到累加器中；而相对应的“MOV P1, A”指令，则是将累加器的内容传送到 P1 口锁存器中。

14. 什么是准双向口？使用准双向口时要注意什么？

答：P0、P1、P2、P3 口做普通 I/O 口使用时，都是准双向口结构。

准双向口的输入操作和输出操作本质不同。输入操作是读引脚状态，输出操作是对口锁存器的写入操作。由口锁存器和引脚电路可知：当由内部总线给口锁存器置 0 或 1 时，锁存器中的“0”、“1”状态立即反映到引脚上。但是在输入操作（读引脚）时，如果口锁存器状态“0”，引脚钳位在“0”状态，导致无法读出引脚的高电平输入，因而，在进行输入操作时，必须先对相应端口锁存器写“1”，使驱动管 FET 截止。

准双向口做输入口时，应先使锁存器置 1，使驱动管 FET 截止，然后再读引脚。例如，要将 P1 口状态读入到累加器 A 中，应执行以下两条指令：

```
MOV P1, #0FFH      ; 将 P1 口设置为输入方式
MOV A, P1            ; 读 P1 口引脚状态到 ACC 中
```

第三章 MCS—51 单片机指令系统与程序设计

第一节 学习要求与内容概要

一、学习要求

- (1) 掌握 51 系列单片机的 7 种寻址方式，并能实际应用，会计算相对寻址方式中目标地址或偏移量 rel。
- (2) 熟记 51 系列单片机指令系统的 111 条指令和常用的伪指令，熟悉指令的功能、寻址方式、操作对象和结果，以及指令执行后对 PSW 有关位的影响，能对给出的程序进行分析。
- (3) 熟悉汇编语言编程的步骤、方法和技巧，学会模块化的程序设计方法。

二、指令寻址方式学习内容概要

寻址方式是指如何找到存放被操作数据的位置（地址）的方法。被操作数所在位置或地址涉及到的存储器空间如表 3-1。

表 3-1 每一种寻址方式可涉及的存储器空间

寻址方式	存储器空间	指令举例
寄存器寻址	R0 ~ R7、A, B, Cy (位), DPTR (双字节)	MOV 55H, R3
直接寻址	内部数据存储器 00H ~ 7FH 特殊功能寄存器	MOV A, 55H
寄存器间接寻址	内部数据存储器 [$@R1$, $@R0$, $@SP$ (仅 PUSH 和 POP)] 内部数据存储器的低 4 位 ($@R1$, $@R0$, 仅 XCHD) 外部数据存储器 ($@R1$, $@R0$, $@DPTR$)	MOV A, $@R0$ XCHD A, $@R1$ MOVX A, $@DPTR$
立即数寻址	程序存储器 (操作码常数)	MOV A, #55H
变址间接寻址	程序存储器 ($@A + DPTR$, $@A + PC$)	MOVC A, $@A + DPTR$
相对寻址	程序存储器 256B 范围 (PC + rel)	SJMP 55H
位寻址	内部数据存储器中的 128 个位地址 (00H ~ 7FH) 特殊功能寄存器中的位地址	CLR 00H SETB EA

三、指令系统学习内容概要

1. 数据传送类指令

数据传送类指令有 29 条，是指令系统中数量最多与使用最多的指令。其中，以 A 为一操作数的传送指令、交换指令的执行时间为 1 个机器周期；不以 A 为一操作数的传送指令、访问片外 RAM 指令、基址寄存器加变址寄存器间址操作指令、进出栈指令等执行时间均为 2 个机器周期；用立即数置数的指令，凡是双字节的，其执行时间为 1 个机器周期，三字节的，执行时间为 2 个机器周期。

在这 29 条数据传送类指令中，重点和难点在于 MOV、MOVX、MOVC 指令的区分和使用。

MOV 指令只访问片内数据存储区（即片内 RAM 和 SFR 区），MOV 操作不仅可以访问片内数据存储区的各字节单元，也可以访问位单元。

MOVX 指令专门用于访问片外 64KB 的 RAM（包括片外的 I/O 外设芯片），对应的读/写指令为 MOVX A, @DPTR 和 MOVX @DPTR, A 以及 MOVX A, @Ri 和 MOVX @Ri, A。MOVX 指令的操作码会使 $\overline{RD}$ 或 $\overline{WR}$ 信号有效，导致片外的数据存储区或外设，如 8155、A/D、D/A 等芯片选通。

MOVC 指令专门用来访问片内或片外 64KB 的 ROM 空间，对于这个空间，用户如果要读取其中的特殊数据（例如表格内容），必须用 MOVC A, @A + DPTR 或 MOVC A, @A + PC 指令。MOVC 指令的操作码会选通片内 ROM 区，或使 $\overline{PSEN}$ 信号有效选通片外 ROM 区。

2. 算术运算类指令

算术运算指令包括加、减、乘、除、加 1、减 1 等 24 条指令。这类指令中除了 INC、DEC 指令外，都会对标志位产生影响。

3. 逻辑运算类指令

51 系列单片机指令系统中逻辑运算指令共 24 条，按操作个数的不同可分为单操作数指令和双操作数指令两类，包括与、或、异或、循环、累加器“清零”与“求反”指令。这些指令的操作数都是 8 位。逻辑运算指令不影响标志位。

4. 控制转移类指令（全称：控制程序转移指令）

51 系列单片机指令系统中，控制转移类指令共有 17 条，包括无条件转移指令、条件转移指令、子程序调用与返回指令和空操作指令。这类指令通过修改 PC 的内容来控制程序的执行过程，可极大地提高程序的效率，实现复杂功能。

无条件转移指令、条件转移指令中的比较转移指令和循环转移指令、子程序调用和返回指令是学习中的重点和难点，也是要考查的重点。

5. 位操作指令

MCS-51 单片机借用 PSW 的 CY 位作为位处理器（又称布尔处理器）完成位操作功能。位操作指令共 17 条，包括位传送、位逻辑运算和位控制转移指令三类。指令中的操作数都是 1 位。希望读者在编程中充分灵活利用这些指令。

6. 其他一些问题

(1) 关于溢出的讨论。溢出表示：运算结果超出了一个字长所能表示的数据范围。当执行有符号数的加法指令 ADD 或减法指令 SUBB 时，且 D_6 位出现向 D_7 位的进位或借位，记为 $C_{6Y} = 1$ ，而 D_7 位没有向 CY 位进位或借位，记为 $C_{7Y} = 0$ ；当 $C_{6Y} = 1$ ， $C_{7Y} = 0$ 或者 $C_{6Y} = 0$ ， $C_{7Y} = 1$ 时， $OV = 1$ ，所以溢出的逻辑表达是 $OV = C_{6Y} \vee C_{7Y}$ 。

例如，带符号数字长 8 位，最高位 D_7 用于表示正负号，数据位 7 位，能表示 -128 ~ +127 之间的数，若超出此范围即产生溢出。

01010100 (+84)
+ 01101001 (+105)

CY = 0 10111101 (+189) 结果为负数产生正溢出

$C_{6Y} = 1$ $C_{7Y} = 0$ $OV = C_{6Y} \vee C_{7Y} = 1 \vee 0 = 1$

无符号数的表示范围是 0 ~ 255, 所以无符号数加法运算溢出和进位是一致的。

(2) 关于 MOV P1, #0FFH 的理解。这句指令容易让人产生误解, 其含义到底理解为输出全“1”到 P1 口呢, 还是理解为从 P1 口“读”引脚(输入)的预备动作呢? 这要根据硬件进行分析, 即单片机如果所接的外设为单纯的输出设备, 则理解为送出全“1”信号; 若单片机所接的外设为单纯的输入设备, 则理解为正确“读”并行口数据之前的预备动作。

(3) 注意程序中符号运算结果的“保号”问题。当需要将单字节符号数扩展为双字节时, 也需要注意“保号”。对于用补码表示的数, 正数的符号扩展应该在其前面补 0; 而负数的符号扩展, 则应该在其前面补 1。例如 68 用 8 位表示为 44H, 用 16 位表示为 0044H; -68 用 8 位表示为 BCH, 用 16 位表示为 FFBCH。

原码数的扩展比较简单, 例如单字节扩展为双字节时, 仅将符号位移至双字节的最高位(D₁₅)即可; 无符号数是正数, 无符号数的扩展是在其前面补 0, 如无符号数 255 用 8 位表示为 FFH, 用 16 位表示为 00FFH。

(4) 51 系列单片机“读”引脚和读锁存器指令(或读-改-写)的区别。“读取引脚前—状态以便修改后再送出的”的指令都应当读取端口 D 锁存器 Q 端的信息而不是引脚信息。例如, MOV C, P1.0 是读 P1.0 的引脚; 而 CPL P1.0 则是读-改-写指令, 它先读 P1.0 的锁存器 Q 端的状态, 接着取反, 然后再次送到 P1.0 的引脚上。

当 P1 口外接输入设备时, 要想使 P1 引脚上反映真实的输入信号, 必须给 P1 口的这一位送一个“1”电平, 先让输出 MOS 管截止才行, 否则当 MOS 管导通时, P1 口引脚上将永远为低电平, 无法正确地反映外设的输入信号。

(5) 把累加器 A 写成 A 与写成 ACC 的差别。A 和 ACC 虽然都是指累加器, 但是在指令中二者有区别: 包含 ACC 的指令在汇编后机器码中必有一个字节的操作数为 ACC 的字节地址 E0H, 而含有 A 的指令在汇编后机器码中隐含在指令操作码中, 所以程序指令中的 A 不能用 ACC 代替; 反之, 特殊功能寄存器直接寻址和位名称寻址要用 ACC, 而不能用 A 替代。

四、伪指令学习内容概要

伪指令即指示性语句, 向汇编过程(把汇编语言源程序编译成机器码语言程序的专用软件)发命令, 在汇编程序编译源程序过程中完成某些处理, 如定义数据、程序定位等。伪指令不产生目标代码, 它不属于 MCS-51 单片机指令系统, 是应汇编程序要求而出现的。常用的汇编语言伪指令如表 3-2 所示。

表 3-2 汇编程序伪指令

伪指令	格 式	说 明
BIT	符号名称 BIT 地址	定义一个在位数据空间的地址
DB	[标号] DB 表达式清单	生成一个字节 (BYTE) 值的清单
END	END	表明汇编程序结束
EQU	符号名称 EQU 表达式或符号名	永久的设置符号值
ORG	ORG 表达式	设置当前段的位置计数器 (PC)

五、汇编语言源程序设计基本方法学习内容概要

任何复杂的程序都可以分解成顺序结构部分、分支结构部分、循环结构部分和子程序, 因此, 大而复杂的程序可以分解设计。

编写汇编语言源程序的目的就是使其在系统应用板上运行，与硬件密切配合，达到系统要求的目的。要使其真正运行，一般必须经过调试过程，排除编写出现的语法错误和算法错误。为此对程序编写提出了需特别注意的几个方面的要求。

1. 编写程序应与硬件系统密切配合

程序中使用的单元地址特别是扩展单元、扩展端口地址应该从硬件电路准确计算得到，并准确使用单元地址。单元地址的错误使用有可能损坏硬件系统。

2. 正确管理、使用堆栈

堆栈是保护现场、数据传递的重要工具，但若使用不当将会造成数据的混乱，甚至破坏程序的正常运行。例如，调用子程序或中断服务时，程序返回指令应该准确地恢复断点地址到 PC 中，若子程序中 PUSH 和 POP 指令没有成对使用，就会造成 PC 不能正确恢复，致使程序不能回断点处继续执行。

3. 程序定位应该准确

系统复位后 PC 的值为 0000H，即程序从程序存储器的 0000H 单元取指令执行，因此，利用汇编程序汇编的完整程序一般利用 ORG 0000H 进行定位，而对于有子程序、中断服务子程序的源程序还需要对相应的子程序进行定位，同样是利用 ORG 伪指令，如外部中断 0 服务子程序用 ORG 0003H 对子程序定位，同时主程序起始应该使用 AJMP 指令以越过中断服务子程序；对于其他子程序的定位，可以在主程序和中断服务子程序定位以后剩余的程序空间的任何地方定位。

另外，程序定位应该避免冲突，子程序等定位应处在其他程序不使用的区域，所以 ORG 语句后面的定位地址值估算应尽量准确。

第二节 原教材习题及解答^①

3-1 什么是寻址方式？MCS-51 单片机有哪几种寻址方式？对 8031 内部 RAM 的 128 ~ 255 字节地址的空间寻址要注意什么？

答：寻址方式是如何找到存放被操作数据的位置（地址）的方法；MCS-51 单片机有 7 种寻址方式：寄存器寻址、直接寻址、立即数寻址、寄存器间接寻址、变址间接寻址、相对寻址和位寻址；8031 内部 RAM 的 128 ~ 255 字节为特殊功能寄存器区，只允许使用直接寻址方式。

3-2 MCS-51 系列单片机无条件转移指令有几种？如何选用？

答：(1) 长转移指令

LJMP addr16；目标地址的选择范围为 64KB 空间任意单元，不影响标志位。

(2) 绝对转移指令

AJMP addr11；由 PC11 ~ 15 地址信息提供低 11 位地址，它和指令组成 16 位目标地址，程序无条件转向同一 2KB 存储空间中的目标地址执行，不影响标志位。

(3) 短转移指令

^① 该节对大部分（有一定难度）的习题进行了解答。因此题号是不连续的。后面章节的习题解答也会遇到类似情况。

SJMP rel；指令中 **rel** 是一个有符号数偏移量，范围 $-128 \sim +127$ ，以补码形式给出，正数表示程序向前跳，负数表示向后跳，跳转范围超过 $+127$ 或 $-128B$ ，本指令不能用。

(4) 间接转移指令

JMP @A + DPTR；把 **A** 中 8 位无符号数与 **DPTR** 中 16 位数相加，结果装入程序计数器 **PC**，控制程序转到目的地址执行程序。这是一条多分支转移指令，由 **DPTR** 决定首地址，由累加器 **A** 动态地选择转到某一支。

3-3 MCS—51 系列单片机绝对调用和长调用有何本质上的区别？如何选用？

答：绝对调用指令 **ACALL** 所调用的子程序首地址必须在 **ACALL** 指令同一个 2KB 范围内的程序存储器中，而长调用指令 **LCALL** 可以寻址 64KB 范围内任意子程序空间。

3-4 MCS—51 系列单片机条件转移指令有何特点？如何求 **rel**？

答：条件转移指令是当满足给定条件时，程序转向目标地址执行；条件不满足则顺序执行下一条指令。条件转移指令分为累加器 **A** 判零转移指令 (**JZ**、**JNZ**)、比较转移指令 (**CJNE**) 和循环转移指令 (**DJNZ**)。

rel 为相对偏移量，即补码形式的相对地址。

$\text{rel} = (\text{目标地址} - \text{PC 当前值})$ 的补码

3-5 MCS—51 系列单片机比较转移指令有何独特之处，可以在哪些量之间进行比较？

答：格式为 **CJNE (目的字节), (源字节), rel**；其独特之处在于可以实现三支转移。它可以是累加器内容与立即数或直接地址单元内容比较，可以是工作寄存器内容与立即数比较，也可以是内部 **RAM** 字节单元内容与立即数比较。指令本身在两个参加比较的数不等时转移，再根据 **CY** 位的内容又可以实现两个方向转移。

3-6 若要完成以下数据传送，如何应用 MCS—51 指令予以实现？

(1) **R1** 的内容传送到 **R0**：

```
MOV A, R1
MOV R0, A
```

(2) 外部 **RAM 20H** 单元的内容送入 **R0**：

```
MOV R1, # 20H
MOVX A, @R1
MOV R0, A
```

(3) 外部 **RAM 0020H** 单元内容送内部 **RAM 20H** 单元：

```
MOV R0, # 20H
MOVX A, @R0
MOV 20H, A
```

(4) 外部 **RAM 1000H** 单元内容送内部 **RAM 20H** 单元：

```
MOV DPTR, # 1000H
MOVX A, @DPTR
MOV 20H, A
```

(5) 外部 **ROM 2000H** 单元内容送 **R0**：

```
MOV A, # 00H
```

```
MOV DPTR, # 2000H
MOVC A, @A + DPTR
MOV R0, A
```

(6) 外部 ROM 2000H 单元内容送内部 RAM 20H 单元：

```
MOV A, # 00H
MOV DPTR, # 2000H
MOVC A, @A + DPTR
MOV 20H, A
```

(7) 外部 ROM 2000H 单元内容送外部 RAM 0020H 单元：

```
MOV A, # 00H
MOV R0, # 20H
MOV DPTR, # 2000H
MOVC A, @A + DPTR
MOVX @R0, A
```

3-8 已知 $SP = 25H$, $PC = 2345H$, 标号 LABEL 所在的地址为 $3456H$, 问执行长调用指令 LCALL LABEL 后, 堆栈指针和堆栈的内容发生什么变化? PC 值等于什么?

答: $(SP) = 27H$, $(26H) = 48H$, $(27H) = 23H$; $(PC) = 3456H$ 。

3-9 上题中的 LCALL 指令能否直接换成 ACALL 指令, 为什么? 如果使用 ACALL 指令, 则可调用的地址范围是什么?

答: 不能, 因为 LABEL 所在地址 $3456H$ 与 PC 不在同一 2 KB 范围内, 即高 5 位地址不同。若要使用 ACALL 指令, 可调用的地址范围是 $2000H \sim 27FFH$ 。

3-10 有一个 16 位二进制数, 高 8 位存于 21H 单元, 低 8 位存于 20H 单元。执行如下程序段, 试问:

(1) 程序段功能是什么?

(2) 能否用 MOV 代替程序段中的 XCH 语句而不改变程序的逻辑功能? 写出相应程序段。

(3) 两个程序段结果是否相同? 差别在哪里?

```
CLR    C
XCH    A, 21H
RRC    A
XCH    A, 21H
XCH    A, 20H
RRC    A
XCH    A, 20H
```

答: (1) 程序段的功能是將此 16 位二进制数除 2 后取整。

(2) 能用 MOV 代替程序段中的 XCH 语句而不改变程序的逻辑功能, 如下:

```
CLR    C
MOV    A, 21H
RRC    A
MOV    21H, A
```

```

MOV    A, 20H
RRC    A
MOV    20H, A

```

(3) 两个程序段结果不相同。原程序执行后 A 的内容为程序运行前 A 中内容，即此程序没有改变 A 中内容；MOV 语句程序中 A 的内容为原 20H 单元内容右移一位后的内容。

3-11 分析以下程序段运行的结果。若将 DA A 指令取消，则结果会有什么不同？

```

CLR    C
MOV    20H, #99H
MOV    A, 20H
ADD    A, #01H
DA     A
MOV    20H, A

```

答：结果为 (20H) = 00H, CY = 1。若取消 DA A 指令，(20H) = 9AH, CY = 0。

3-12 “由于 SJMP 指令的转移范围是 256B，而 AJMP 指令转移范围是 2KB，所以在程序中 SJMP 指令都可以用 AJMP 指令来代替。”请问这种说法是否正确，为什么？

答：不正确。AJMP 转移的目标地址必须在 AJMP 后面指令的第一个字节开始的同一 2KB 范围内，不能改变高 5 位地址，因此，即使相邻很近的两个地址，只要它的高 5 位发生改变，就不能跳转。而 SJMP 可以通过几次跳转，最终到任意目标地址。

3-13 使用位操作指令实现下列逻辑操作，要求不更改单元其他位的内容。

解：(1) 使 ACC.0 置 1： SETB ACC.0

(2) 清除累加器的高四位： ANL A, #0FH

或 CLR ACC.7

 CLR ACC.6

 CLR ACC.5

 CLR ACC.4

(3) 清除 ACC.3, ACC.4, ACC.5, ACC.6：

 ANL A, #87H

或 CLR ACC.3

 CLR ACC.4

 CLR ACC.5

 CLR ACC.6

3-14 将片内 RAM 中 0FH 单元的内容传送到寄存器 B，对 0FH 单元的寻址可有三种方法：R 寻址、R 间址、Direct 寻址。请分别编出相应的程序，比较其字节数、机器周期和优缺点。

答：R 寻址： SETB RS0, 1 字节，单周期

 CLR RS1； 1 字节，单周期

 MOV 0F0H, R7； 2 字节，双周期

R 间址： MOV R0, #0FH； 2 字节，单周期

 MOV 0F0H, @R0； 2 字节，双周期

Direct 寻址： MOV A, 0FH ; 2 字节, 单周期
 MOV 0F0H, A ; 2 字节, 单周期

3-15 怎样把位 40H 的内容移至 30H 位？

答： MOV C, 40H
 MOV 30H, C

3-16 编写一段程序, 模拟如图 3-1 所示逻辑电路的逻辑功能。要求将 4 个输入与非门的功能模拟先写成一个子程序, 然后多次调用得到整个电路的功能模拟。设 X、Y、Z、W 和 F 都已定义为位地址, 若程序中还需要其他位地址, 也可另行定义。

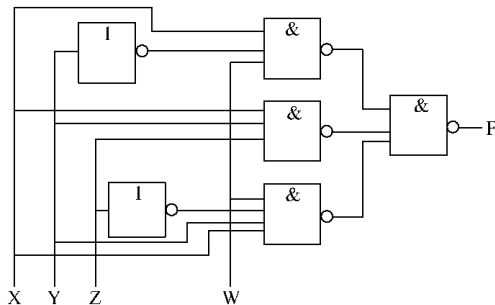


图 3-1 逻辑电路

提示：首先用伪指令定义 X、Y、Z、W、F 是单片机引脚端口位, 然后再编程, 注意操作对象是位单元, 应该使用位逻辑指令。

4-1 两个 4 位 BCD 码数相加求和。设被加数存于内部 RAM 的 40H、41H 单元, 加数存于 45H、46H 单元, 要求和数存于 50H、51H 单元 (均前者为低 2 位, 后者为高 2 位)。请编制加法程序段。若进行 BCD 码减法运算, 应如何考虑？

解：程序如下：

```
CLR    C
MOV    A, 40H
ADD    A, 45H
DA     A
MOV    50H, A
MOV    A, 41H
ADDC   A, 46H
DA     A
MOV    51H, A
CLR    A
ADDC   A, #00H
MOV    52H, A
```

编写多字节十进制 (BCD 码) 减法程序段：

设被减数低字节地址存于 R0 中, 减数低字节地址存于 R1 中, 字节数存于 R3 中。相减的结果依次存于原被减数单元。

START: SETB C

```

SUBA : CLR      A
        ADDC    A, # 99H      ; 加 99
        SUBB    A, @R1        ; 减去减数
        ADD     A, @R0        ; 作加法, 加上被加数
        DA      A              ; 十进制调整
        MOV     @R0, A        ; 存入对应的被减数单元
        INC     R0             ; 指向下一个字节单元
        INC     R1
        DJNZ    R3, SUBA      ; 若 (R3) - 1 ≠ 0, 继续作减法
        RET

```

注意：8051 中的十进制调整指令 (DA A) 只能用于二进制加法指令之后, 不能用于二进制减法结果的调整。进行十进制 (BCD 码) 减法只能采用先求减数的十进制补码, 然后把结果与被减数相加。

如果被减数为 A, 减数为 B, 十进制的位数为 n, 则

$$A - B = A + B_{bu} = A + 10^n - B = A + 9 \cdots 9 + 1 - B = 9 \cdots 9 + 1 - B + A$$

式中, $9 \cdots 9$ 的个数为 n_0 。

4-2 试编写程序, 查找在内部 RAM 20H-50H 单元中出现 00H 的次数, 并将查找的结果存入 51H 单元。

解：程序如下：

```

        MOV     R0, # 20H
        MOV     R7, # 31H
        MOV     51H, # 00H
LOOP1 : CJNE    @R0, # 00H, LOOP
        INC     51H
LOOP :   INC     R0
        DJNZ    R7, LOOP1
        HERE    SJMP HERE

```

4-3 晶振频率为 12MHz, 请用循环转移指令编制延时 20ms 的延时子程序。

解：一个机器周期为 1μs, 延时 20ms 需要机器周期数 $20000\mu s / 1\mu s = 20000$

延时程序如下：

```

        MOV     R5, # T1      ; 1 个机器周期
LOOP2 : MOV     R4, # T2      ; T1 个机器周期
LOOP1 : NOP          ; T1 × T2 个机器周期
        NOP          ; T1 × T2 个机器周期
        DJNZ    R4, LOOP1    ; 2 × T1 × T2 个机器周期
        DJNZ    R5, LOOP2    ; 2 × T1 个机器周期
        RET           ; 1 个机器周期

```

总计时 $1 + 3 \times T1 + 4 \times T1 \times T2 + 1 = 20000$

则 $T1 = 202 = 0CAH, T2 = 24 = 18H$ (答案不唯一, $T1$ 与 $T2$ 的值 ≤ 255)

4-4 已知两个 8 位无符号数 a、b 存放在 BUF 和 BUF + 1 单元，编写程序，计算 $5a + b$ ，结果可能大于 8 位，仍放回 BUF 和 BUF + 1 单元（BUF + 1 单元为高字节）。

解：根据题意，编写程序如下：

```
MOV    A, BUF
MOV    B, # 05H
MUL    AB
CLR    C
ADD    A, BUF + 1
MOV    BUF, A
MOV    A, B
ADDC   A, # 00H
MOV    BUF + 1, A
```

4-5 外部数据 RAM 从 2000H 到 2100H 有一个数据块，现要将它们传送到从 3000H 到 3100H 的区域，试编写有关程序。

方法 1：

```
MOV    P2, # 20H
MOV    R0, # 00H
MOV    R2, # 00H
LOOP:  MOVX  A, @R0                ; P2 口即高 8 位地址
        SETB  P2.4                ; 20H→30H
        MOVX  @R0, A
        CLR   P2.4                ; 30H→20H
        INC   R0
        DJNZ  R2, LOOP
        MOV   DPTR, # 2100H
        MOVX  A, @DPTR
        MOV   DPTR, # 3100H
        MOVX  @DPTR, A
```

方法 2：

```
MOV    DPTR, # 3000H
PUSH   DPL
PUSH   DPH                ; 压栈保护目标地址
MOV    DPTR, # 2000H
MOV    R2, # 01H
MOV    R3, # 01H          ; 数据长度送 R3R2
LOOP:  MOVX  A, @DPTR
        INC   DPTR
        MOV   R5, DPH
        MOV   R4, DPL      ; 源地址加 1 保护
        POP   DPH
        POP   DPL          ; 弹出目标地址
```

```

MOVX    @DPTR, A
INC     DPTR
PUSH    DPL
PUSH    DPH                ; 目标地址加 1 压栈保护
MOV     DPH, R5
MOV     DPL, R4            ; 取出源地址
CLR     C
MOV     A, R2
SUBB    A, # 01H
MOV     R2, A              ; 数据区长度低 8 位减 1
MOV     A, R3
SUBB    A, # 00H
MOV     R3, A              ; 数据区长度高 8 位减借位
ORL     A, R2              ; (R3) 与 (R2) 相或
JNZ     LOOP              ; 两者全零时, 101H 个数传送完毕。

```

HERE: SJMP HERE

4-6 求16位带符号二进制补码数的绝对值。假定补码放在内部 RAM 的 num 和 num + 1 单元, 求得的绝对值仍放在原单元中。(设 num + 1 中内容为高 8 位)

解: 根据题意, 编写子程序如下:

```

ABC:      MOV     A, num + 1
          JB      ACC.7, FUSHU      ; 最高位为 1, 负数, 转移处理
          RET                                ; 正数, 补码即绝对值

FUSHU:    MOV     R0, # num
          MOV     A, @R0
          CPL     A
          ADD     A, # 01H          ; 低 8 位取反加 1
          MOV     @R0, A
          INC     R0
          MOV     A, @R0
          CPL     A
          ADDC    A, # 00H          ; 高 8 位取反加 1, 同时加进位
          MOV     @R0, A
          RET

```

4-7 下列程序段经汇编后, 从1000H开始的各有关存储单元的内容是什么?

```

ORG      1000H
TAB1     EQU    1234H
TAB2     EQU    3000H
DB       "START"
DW       TAB1, TAB2, 70H

```

答：1000H：53H；1001H：54H；1002H：41H；1003H：52H；1004H：54H
 1005H：12H；1006H：34H；1007H：30H；1008H：00H；1009H：00H
 100AH：70H

4-8 编程将20H单元中的两个 BCD 数拆开，并变成相应的 ASCII 码存入 21H 和 22H 单元。

解：根据题意，编写程序如下：

```

MOV    R0, # 22H
MOV    @R0, # 00H
MOV    A, 20H
XCHD   A, @R0
ORL    22H, # 30H           ; 转换低 4 位
SWAP   A
ORL    A, # 30H             ; 转换高 4 位
MOV    21H, A
HERE:   SJMP   HERE

```

4-9 存放在片内RAM的 DAT 单元中的自变量 X 是一个无符号数，试编写程序求下面函数的函数值，并存放到片内 RAM 的 FUNC 单元中。

$$Y = \begin{cases} X & (X \geq 50) \\ 5X & (50 > X \geq 20) \\ 2X & (X < 20) \end{cases}$$

解：根据题意设计程序如下：

```

MOV    A, DAT
CJNE   A, # 32H, LOOP
LOOP:  JNC    DONE
CJNE   A, # 14H, LOOP1
LOOP1: JC    LOOP2
RL     A
RL     A
ADD    A, DAT
SJMP   DONE
LOOP2: RL    A
DONE:  MOV    FUNC, A
HERE:  SJMP   HERE

```

4-11 已知片内RAM以 ADDR 为起始地址的区域中存放着 24 个无符号数，试编写程序找出最小值，并存入 MIN 单元。

解：根据题意，编写程序如下：

```

MOV    R0, # ADDR           ; 数据区首地址
MOV    R7, # 18H            ; 数据区长度

```

```

MOV    A, @R0        ; 读第一个数
DEC    R7
LOOP:  INC    R0
MOV    2AH, @R0      ; 读下一个数
CJNE   A, 2AH, CHK    ; 数值比较
CHK:   JC     LOOP1    ; A 值小转移
MOV    A, @R0
LOOP1: DJNZ   R7, LOOP ; 继续
MOV    2BH, A         ; 极值送 2BH 单元
HERE:  AJMP   HERE    ; 停止

```

第三节 增选习题及解答

一、填空题

1. MCS-51 特殊功能寄存器只能采用_____寻址方式。
2. 片内 20H ~ 2FH 范围内的数据存储器, 既可以字节寻址又可以_____寻址。
3. 访问 MCS-51 程序存储器传送指令的操作码助记符应为_____。
4. 访问 MCS-51 片内 RAM 应该使用的传送指令的助记符是_____。
5. 当 MCS-51 使用间接寻址方式访问 RAM 高 128B, 会产生_____。
6. 指令是通知计算机完成某种操作的_____。
7. 汇编语言中可以使用伪指令, 它们不是真正的指令, 只是用来_____。
8. 累加器 (A) = 80H, 执行完指令 ADD A, # 83H 后, 进位位 C = _____。
9. 执行 ANL A, # 0FH 指令后, 累加器 A 的高 4 位 = _____。
10. JZ rel 的操作码的地址为 1000H, rel = 20H, 它的转移目的地址为_____。
11. JBC 00H, e 操作码的地址为 2000H, e = 70H, 它的转移目的地址为_____。
12. 累加器 (A) = 7EH, (20H) = # 04H, MCS-51 执行完 ADD A, 20H 指令后, PSW.0 为_____。
13. MOV PSW, # 10H 是将 MCS-51 的工作寄存器置为第_____组。
14. 指令 DJNZ R7, e 的指令操作码所在地址为 3000H, e = EFH, 则它的转移目的地址为_____。
15. ORL A, # 0F0H 是将 A 的高 4 位置 1, 而低 4 位_____。
16. SJMP e 的指令操作码地址为 0050H, e = 65H, 那么它的转移目的地址为_____。
17. 设 DPTR = 2000H, (A) = 80H, 则 MOVC A, @A + DPTR 的操作数的实际地址为_____。
18. 执行当前指令后, PC 内容为_____。
19. MOV C, 20H 源寻址方式为_____寻址。
20. INC 指令_____影响 CY 位。
21. 指令 LCALL 37B0H, 首地址在 2000H, 所完成的操作是_____入栈, (PC) = _____。

23. ORG 1000H

LCALL 4000H

ORG 4000H

ADD A, R2; 执行完 LCALL 后 (PC) = _____。

24. 通过堆栈操作实现子程序调用, 首先要将_____的内容入栈, 以进行断点保护。调用返回时再进行出栈操作, 把保护的断点弹回_____。

25. 一台计算机的指令系统就是它所能执行的_____集合。

26. 以助记符形式表示的计算机指令就是它的_____语言。

27. 在直接寻址方式中, 只能使用_____位二进制数作为直接地址, 因此其寻址对象只限于_____。

28. 在寄存器间接寻址方式中, 其“间接”体现在指令中寄存器的内容不是操作数, 而是操作数的_____。

29. 在变址寻址方式中, 以_____作变址寄存器, 以_____或_____作基址寄存器。

30. 假定累加器 A 的内容为 30H, 执行指令 1000H: MOV A, @A + PC 后, 把程序存储器_____单元的内容送累加器 A 中。

31. 假定 DPTR 的内容为 8100H, 累加器 A 的内容为 4DH, 执行下列指令 MOV A, @A + DPTR 后, 送入 A 的是程序存储器_____单元的内容。

32. 设 (SP) = 60H, (ACC) = 30H, (B) = 70H, 执行指令 PUSH ACC, PUSH B 后, SP 的内容为_____, 61H 单元的内容为_____, 62H 单元的内容为_____。

33. 假定 (SP) = 62H, (61H) = 30H, (62H) = 70H。执行下列指令 POP DPH, POP DPL 后, DPTR 的内容为_____, SP 的内容为_____。

34. 假定 (A) = 85H, (R0) = 20H, (20H) = 0AFH。执行指令 ADD A, @R0 后, 累加器 A 的内容为_____, CY 的内容为_____, AC 的内容为_____, OV 的内容为_____。

35. 假定 (A) = 85H, (R0) = 20H, (20H) = 0AFH。执行指令 ADD A, 20H 后, 累加器 A 的内容为_____, CY 的内容为_____, AC 的内容为_____, OV 的内容为_____。

36. 假定 (A) = 0FFH, (R3) = 0FH, (30H) = 0F0H, (R0) = 40H, (40H) = 00H。执行指令: INC A, INC R3, INC 30H, INC @R0 后, A 的内容为_____, R3 的内容为_____, 30H 的内容为_____, 40H 的内容为_____。

37. 在 MCS-51 中 PC 和 DPTR 都用于提供地址, 但 PC 是为访问_____存储器提供地址, 而 DPTR 是为访问_____存储器提供地址。

38. 在位操作中, 能起到与字节操作中累加器作用的是_____。

39. 累加器 A 中存放一个其值小于或等于 127 的 8 位无符号数, CY 清“0”后执行 RLC A 指令, 则 A 中数变为原来的_____倍。

40. 假定 (A) = 56, (R5) = 67。执行指令 ADD A, R5; DAA 后, 累加器 A 的内容为_____, CY 的内容为_____。

41. 假定 (A) = 0FH, (R7) = 19H, (30H) = 00H, (R1) = 40H, (40H) = 0FFH。执行指令

```

DEC    A
DEC    R7
DEC    30H
DEC    @R1

```

后, A 的内容为_____, R7 的内容为_____, 30H 的内容为_____, 40H 的内容为_____。

42. 假定 (A) = 50H, (B) = 0A0H。执行指令: MUL AB 后, 寄存器 B 的内容为_____, A 的内容为_____, CY 的内容为_____, OV 的内容为_____。

43. 假定 (A) = 0FBH, (B) = 12H。执行指令 DIV AB 后, 累加器 A 的内容为_____, 寄存器 B 的内容为_____, CY 的内容为_____, OV 的内容为_____。

44. 假定 (A) = 0C5H。执行指令 SWAP A 后, 累加器 A 的内容为_____。

45. 执行如下指令序列

```

MOV    C, P1.0
ANL    C, P1.1
ANL    C, /P1.2
MOV    P3.0, C

```

后, 所实现的逻辑运算式为_____。

46. 假定 addr11 = 00100000000B, 标号 qaz 的地址为 1030H。执行指令 qaz: AJMP addr11 后, 程序转移地址_____去执行。

47. 假定标号 qaz 的地址为 0100H, 标号 qwe 值为 0123H (即跳转的目标地址为 0123H)。应执行指令 qaz: SJMP qwe, 该指令的相对偏移量 (即指令的第二字节) 为_____。

48. DPTR 是 MCS-51 中惟一一个 16 位寄存器, 在程序中常用来做为 MOVC 指令的访问程序存储器的_____使用。

49. 在 R7 初值为 00H 的情况下, DJNZ R7, rel 指令将循环执行_____次。

50. 欲使 P1 口的低 4 位输出 0 而高 4 位不变, 应执行一条_____指令。欲使 P1 口的高 4 位输出 1 而低 4 位不变, 应执行一条_____指令。

51. MCS-51 的两条查表指令是_____和_____。

填空题参考答案:

1. 直接 2. 位 3. MOVC 4. MOV 5. 错误 6. 命令 7. 对汇编过程进行某种控制
8. 1 9. 0000 10. 1022H 11. 2073H 12. 0 13. 2 14. 2FF1H 15. 不变 16. 00B7H
17. 2080H 18. 下一条将要读取的指令码首地址 19. 位 20. 不 21. 2003H 22. 寄存器间接寻址
23. 4000H 24. PC, PC 25. 指令 26. 汇编 27. 8 片内 RAM 28. 地址 29. A, PC, DPTR
30. 1031H 31. 814DH 32. 62H, 30H, 70H 33. 7030H 60H 34. 34H, 1, 1, 1 35. 34H, 1, 1, 1
36. 00H, 10H, 0F1H, 01H 37. 程序, 数据 38. CY 39. 2 40. 00100011, 1
41. 0EH, 18H, 0FFH, 0FEH 42. 32H, 00H, 0, 1 43. 0DH, 11H, 0, 0 44. 5CH
45. P3.0 = (P1.0) ∧ (P1.1) ∧ $\overline{P1.2}$ 46. 1100H 47. 0123 - 0102 = 21H 48. 基址寄存器
49. 256 50. ANL P1, # F0H; ORL P1, # F0H 51. MOVC A, @A + DPTR, MOVC A, @A + PC

二、选择题

1. 在中断服务程序中, 至少应有一条 ()。

A. 传送指令 B. 转移指令 C. 加法指令 D. 中断返回指令

2. 要用传送指令访问 MCS-51 片外 RAM, 它的指令操作码助记符应是 ()。

A. MOV B. MOVX C. MOVC D. 以上都行

3. ORG 2000H

LCALL 3000H

ORG 3000H

RET

上面程序执行完 RET 指令后, (PC) = ()。

A. 2000H B. 3000H C. 2003H D. 3003H

4. JNZ e 指令的寻址方式是 ()。

A. 立即寻址 B. 寄存器寻址 C. 相对寻址 D. 位寻址

5. 执行 LCALL 4000H 指令时, MCS-51 所完成的操作是 ()。

A. 保护 PC B. 4000H → HPC

C. 保护现场 D. PC + 3 入栈, 4000H → PC

6. MOVX A, @DPTR 指令中源操作数的寻址方式是 ()。

A. 寄存器寻址 B. 寄存器间接寻址 C. 直接寻址 D. 立即寻址

7. ORG 0003H

LJMP 2000H

ORG 000BH

LJMP 3000H

当 CPU 响应外部中断 0 后, PC 的值是 ()。

A. 0003H B. 2000H C. 000BH D. 3000H

8. PUSH ACC 指令, MCS-51 完成的操作是 ()。

A. SP + 1 → SP (ACC) → (SP)

B. (ACC) → (SP) SP - 1 → SP

C. SP - 1 → SP (ACC) → (SP)

D. (ACC) → (SP) SP + 1 → SP

9. 把 P0 口高 4 位变 0, 低 4 位不变, 应使用指令 ()。

A. ORL P0, # 0FH B. ORL P0, # 0F0H

C. ANL P0, # 0F0H D. ANL P0, # 0FH

10. LCALL 存于 2000H 开始的地址单元中, 执行完相应子程序中的返回指令后, PC = ()。

A. 2000H B. 2001H C. 2002H D. 2003H

11. MCS-51 执行完 MOVA, # 08H 后, PSW 的 () 位被置位。

A. C B. F0 C. OV D. P

12. 89C51 执行 ADDA, 20H 指令时, 首先在 P0 口上出现的信息是 ()。

A. 操作码地址 B. 操作码 C. 操作数 D. 操作数地址

13. 要访问 MCS-51 的特殊功能寄存器应使用的寻址方式是 ()。
- A. 寄存器间接寻址 B. 变址寻址 (C) 直接寻址 D. 相对寻址
14. 将 MCS-51 的工作寄存器置成 3 区的指令是 ()。
- A. MOV PSW, # 13H B. MOV PSW, # 18H
C. SETB PSW.4 ; CLR PSW.3 D. SETB PSW.3 ; CLR PSW.4
15. MCS-51 的相对转移指令的最大负跳距离为 ()。
- A. 2KB B. 128B C. 127B D. 256B
16. MOV C, 00H 指令的第 2 操作数寻址方式是 ()。
- A. 位寻址 B. 直接寻址 C. 立即寻址 D. 寄存器寻址
17. ORG 0000H
 ALMP 0040H
 ORG 040H
 MOV SP, # 00H
- 当执行完左边的程序后, PC 的值是 ()。
- A. 0040H B. 0041H C. 0042H D. 0043H
18. 程序设计的方法一般有 ()。
- A. 1 种 B. 2 种 C. 3 种 D. 4 种
19. MCS-51 的并行 I/O 口信息有两种读取方法, 一种是读引脚, 还有一种是 ()。
- A. 读锁存器 B. 读数据 C. 读 A 累加器 D. 读 CPU
20. MCS-51 的并行 I/O 口读 - 改 - 写操作, 是针对该口的 ()。
- A. 引脚 B. 片选信号 C. 地址线 D. 内部锁存器
21. 以下指令中, 属于单纯读引脚的指令是 ()。
- A. MOV P1, A B. ORL P1, # 0FH
C. MOV C, P1.5 D. DJNZ P1, short-lable
22. 指令 AJMP 的跳转范围是 ()。
- A. 256B B. 1KB C. 2KB D. 64KB
23. 以下运算中对溢出标志 OV 没有影响或不受 OV 影响的运算是 ()。
- A. 逻辑运算 B. 符号数加减法运算
C. 数据大小比较 D. 除法运算
24. 在算术运算中, 与辅助进位位 AC 有关的是 ()。
- A. 二进制数 B. 八进制数 C. 十进制数 D. 十六进制数
25. 在相对寻址方式中, “相对” 两字是指跳转 ()。
- A. 地址偏移量 rel B. 当前指令的首地址
C. 当前指令的末地址 D. DPTR 值
26. 在寄存器间接寻址方式中, 指定寄存器中存放的是 ()。
- A. 操作数 B. 操作数地址
C. 转移地址 D. 地址偏移量
27. 对程序存储器的读操作, 只能使用 ()。
- A. MOV 指令 B. PUSH 指令

C. MOVX 指令 D. MOVC 指令

28. 必须进行十进制调整的十进制运算 ()。

A. 有加法和减法 B. 有乘法和除法

C. 只有加法 D. 只有减法

29. 执行返回指令时, 返回的断点是 ()。

A. 调用指令的首地址 B. 调用指令的末地址

C. 调用指令下一条指令的首地址 D. 返回指令的末地址

30. 可以为访问程序存储器提供或构成地址的有 ()。

A. 只有程序计数器 PC B. 只有 PC 和累加器 A

C. 只有 PC, A 和数据指针 DPTR D. PC、A、DPTR 和堆栈指针 SP

选择题参考答案：

1.D 2.B 3.C 4.C 5.D 6.B 7.B 8.A 9.D 10.D 11.D 12.A 13.C 14.B
15.B 16.A 17.C 18.D 19.A 20.D 21.C 22.C 23.A 24.C 25.A 26.B 27.D
28.C 29.C 30.C

三、简答题

1. 位地址 7CH 与字节地址 7CH 如何区别? 位地址 7CH 具体在片内 RAM 中什么位置?

答: 字节地址是片内 RAM 的单元地址, 而位地址是片内 RAM 单元中的某一位。7CH 字节地址为 RAMR 的 7CH 单元, 而 7CH 位地址是 RAM2FH 单元中 D4 位。

2. 程序存储器的 0543H 和 0544H 两单元中存有一条 AJMP 指令。若其代码为 E165H, 则目的地址等于什么?

答: AJMP 指令之目的地址的高 5 位来自程序计数器 PC 的高 5 位。在把这条指令的两个字节从程序存储器取出并送入指令寄存器 IR 中去之后, PC 内容加 2, 由原来的 0543H 变成了 0545H。其高 5 位为 00000B, 目的地址的低 11 位为操作码高 3 位与指令第二个字节的程序组合。指令代码第一个字节是 E1H, 其最高 3 位等于 111B; 第二个字节为 65H。因此这条 AJMP 指令的目的地址等于 0765H。

3. 读下面一段程序, 并以简单方法对它进行改写, 限用五条指令。

```
MOV    R0, #21H
MOV    A, 20H
ANL    A, #0FH
MOV    @R0, A
INC    R0
MOV    A, 20H
SWAP   A
ANL    A, #0FH
MOV    @R0, A
```

答: 这段程序的任务是把片内 RAM20H 单元的内容分解成高 4 位和低 4 位, 低 4 位存入 21H 单元, 高 4 位存入 22H 单元。可以认为这是把 20H 单元内的数据分成两个十六进制位或两个 BCD 位。这可用除法进行:

```

MOV    A, 20H        ;取数
MOV    B, #10H       ;除数为 16
DIV     AB            ;分离十六进制位
MOV    21H, B        ;存低位
MOV    22H, A        ;存高位

```

原理： $abH = a \times 16^1 + b \times 16^0 = a \times 16 + b \times 1$ ，其中设 a 为高 4 位， b 为低 4 位。

4. 用一条什么指令可以取代下列四条指令？

```

MOV     DPTR, #1234H
PUSH    DPL
PUSH    DPH
RET

```

答：这四条指令的任务是要转移到目的地址 1234H 去，所以可用一条 LJMP 1234H 指令来取代它们。

5. 为什么 SJMP 指令为 $rel = 0FEH$ 时，将实现单指令的无限循环？

答：HERE：SJMP HERE；上述指令的机器码为 80FE，即 $rel = 0FEH$ ，此时程序将在原地进行无限循环。在等待中断或程序结束时常用此种办法。

6. 片内 RAM20H ~ 2FH 中的 128 个位地址与直接地址 00H ~ 7FH 形式完全相同，如何在指令中区分出位寻址操作和直接寻址操作？

答：虽然片内 RAM20H ~ 2FH 中的 128 个位地址与直接地址 00H ~ 7FH 形式完全相同，但是在位寻址操作和直接寻址操作中，这两者的区别是很明显的。位寻址的操作只适用于位指令，而直接地址操作对这些指令是无效的。

四、程序运行分析

1. 位地址为 M、N、Y，程序如下，求程序功能表达式。

```

MOV     C, M
ANL     C, /N
MOV     Y, C
MOV     C, M
ANL     C, N
ORL     C, Y
MOV     Y, C

```

答：程序功能表达式为 $(Y) = (M) \cdot (\overline{N}) + (M) \cdot (N)$ 。

2. 程序存储器空间表格如下：

地址	2000H	2001H	2002H	2003H	...
内容	3FH	06H	5BH	4FH	...

已知：片内 RAM 的 20H 中为 01H，执行下列程序后 (30H) 为多少？

```

MOV     A, 20H
INC     A

```

```

MOV     DPTR, # 2000H
MOVC    A, @A + DPTR
CPL     A
MOV     30H, A
END:    SJMP    END

```

答：执行程序后，(30H) = A4H。

3. 阅读下列程序段并回答问题。

```

CLR     C
MOV     A, # 9AH
SUBB    A, 60H
ADD     A, 61H
DA      A
MOV     62H, A

```

(1) 请问该程序执行何种操作？

(2) 已知初值：(60H) = 23H, (61H) = 61H, 请问运行后：(62H) = _____ ?

答：(1) 操作是单字节 BCD 码运算，是将 (61H) - (60H) → 62H

(2) (62H) = 38H。

4. 解读下列程序，然后填写有关寄存器内容。

```

(1)  MOV     R1, # 48H
      MOV     48H, # 51H
      CJNE    @R1, # 51H, 00H
      JNC     NEXT1
      MOV     A, # 0FFH
      SJMP    NEXT2
NEXT1: MOV     A, # 0AAH
NEXT2: SJMP    NEXT2

```

则累加器 (A) = () ?

```

(2)  MOV     A, # 0FBH
      MOV     PSW, # 10H
      ADD     A, # 7FH

```

若 PSW = 00, 当执行完上述程序段后, PSW 各位值为多少？

答：(1)

MOV	R1, # 48H	；48H → (R1)	
MOV	48H, # 51H	；51H → (48H)	
CJNE	@R1, # 51H, 00H	；(R1) 值等于 51H, 顺序执行	
JNC	NEXT1	；没借位转, NEXT1	
MOV	A, # 0FFH	；有借位, FFH → (A)	
SJMP	NEXT2		
NEXT1:	MOV	A, # 0AAH	；0AAH → (A)
NEXT2:	SJMP	NEXT2	

累加器 (A) = 0AAH

```
(2)      MOV      A, # 0FBH
          MOV      PSW, # 10H           ; 00010000→PSW
          ADD      A, # 7FH
```

由程序知，累加器完成加法运算，即

$$\begin{array}{r} 11111011 \\ + 01111111 \\ \hline 1\ 01111010 \end{array}$$

所以，有进位 $CY = 1$ ； $C6 \oplus C7 = 1 \oplus 1 = 0$ ， $OV = 0$ ；A 中有奇数个 1， $P = 1$ ；有辅助进位位 $AC = 1$ 。

当执行完上述程序段后，PSW 各位状态为：

CY	AC	F0	RS1	RS0	OV	F1	P
1	1	0	1	0	0	0	1

5. 试说明下列指令的作用，并将其翻译成机器码，执行最后一条指令对 PSW 有何影响？A 的终值为多少？

```
MOV      A, # 20H
MOV      B, A
ADD      A, B
SUBB     A, # 10H
DIV      AB
```

答：下面列出程序中各指令相应的机器码，执行最后一条指令对 PSW 的影响及 A 的终值。

机器码	源程序	执行每条指令后的结果
74 20	MOV A, # 20H	；把立即数 20H 送入 A
F5 F0	MOV B, A	；把 20H 送入 B
25 F0	ADD A, B	；A 与 B 中值相加等于 40H，送入 A
94 10	SUBB A, # 10H	；A 中值 40H 减 10H 等于 30H，送入 A
84	DIV AB	；A 中值与 B 相除，商 01H 送入 A，余数 10H 送入 B

执行此指令后 PSW 中 $P = 1$ ， $OV = 0$ ， $CY = 0$ 。

6. 用 80C51 单片机的 P1 端口作为输出，经驱动电路接 8 只发光二极管，如图 3-2 所示。当输出位是“1”时，发光二极管点亮；输出位是“0”时，发光二极管为暗。试分析下述程序执行过程及发光二极管点亮的工作规律。

```
LP:      MOV      P1, # 81H
          LCALL    DELAY
          MOV      P1, # 42H
          LCALL    DELAY
          MOV      P1, # 24H
          LCALL    DELAY
```

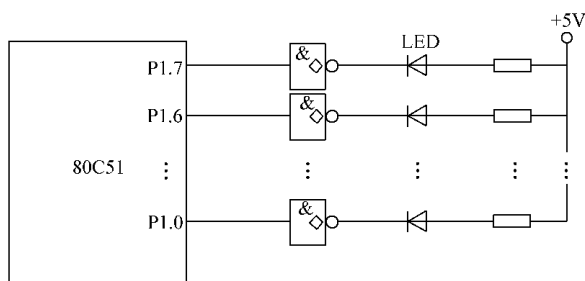


图 3-2 P1 驱动发光二极管电路

```

MOV     P1, # 18H
LCALL   DELAY
MOV     P1, # 24H
LCALL   DELAY
MOV     P1, # 42H
LCALL   DELAY
SJMP    LP

```

子程序：

```

DELAY: MOV     R2, # 0FAH
L1:     MOV     R3, # 0FAH
L2:     DJNZ    R3, L2
        DJNZ    R2, L1
        RET

```

答：上述程序执行过程及发光二极管点亮的工作规律为：首先是第 1 和第 8 个灯亮；延时一段时间后，第 2 和第 7 个灯亮；再延时一段时间后，第 3 和第 6 个灯亮；又延时一段时间后，第 4 和第 5 个灯亮。接着再延时一段时间后，重复上述过程。

7. 在第 6 题中，若系统的晶振频率为 6MHz，求子程序 DE-LAY 的延时时间。若想加长或缩短延时时间，应怎样修改？

答：设系统的晶振频率为 6MHz，子程序 DELAY 的延时时间计算如下：

指令周期数		执行时间	
DELAY: MOV	R2, # 0FAH	1	$2\mu\text{s}$
L1: MOV	R3, # 0FAH	1	$2\mu\text{s}$
L2: DJNZ	R3, L2	2	$4\mu\text{s}$
	DJNZ R2, L1	2	$4\mu\text{s}$
	RET	1	$2\mu\text{s}$

因为 $0\text{FAH} = 250$ ，所以总时间 T 计算如下：

$$T = 4\mu\text{s} + (250 \times 4\mu\text{s} + 4\mu\text{s} + 4\mu\text{s}) \times 250 + 4\mu\text{s} = 252.008\mu\text{s}$$

若想加长延时时间，可以增加循环次数。

若想缩短延时时间，可以减少循环次数。

8. 根据题 7 程序完成的功能，试编制灯亮移位程序，即 8 个发光二极管每次亮一个，循

环左移，一个一个地亮，循环不止。

答：根据题图 7 程序完成的功能，编制灯亮移位程序如下：

```

MOV      A, #1
L1:      MOV      P1, A
        RL        A
        LCALL     DELAY      ; DELAY 同第 8 题中的延时程序
        SJMP      L1
DELAY:   (略)

```

五、程序设计

1. R1 中存有一位 BCD 码，请编程将它转换成 ASCII 码，存入片外 RAM1000H 单元中。

解：程序如下：

```

RIB-AI:  MOV      A, R1
        ORL      A, #30H      ; (ADDA, #30H 也可以)
        MOV      DPTR, #1000H
        MOVX     @DPTR, A
        RET

```

2. 片内 RAM40H 开始的单元中有 10 个二进制数，编程找出其中最大值并存于 50H 单元中。

解：程序如下：

```

START:   MOV      R0, #40H      ; 数据块首地址送 R0
        MOV      R7, #09H      ; 比较次数送 R7
        MOV      A, @R0        ; 取数送 A
LOOP:    INC      R0
        CJNE     A, @R0, NEXT   ; (A) 与 (30H) 相比
NEXT:    JNC      BIE1          ; (A) ≥ (30H) 转 BIE1
        MOV      A, @R0        ; (A) < (30H)，大数送 A
BIE1:    DJNZ     R7, LOOP      ; 比较次数减 1，不为 0，继续比较
        MOV      50H, A        ; 比较结束，大数送 50H
        RET

```

3. 请用三种方法实现累加器 A 与寄存器 B 的内容交换。

答：方法 1：XCH A, B

方法 2：MOV R0, B

MOV B, A

MOV A, R0

方法 3 用四条指令实现：

PUSH ACC

PUSH B

POP ACC

POP B

4. 求片外 RAM3000H 和 3001H 单元数据的平均值, 并传送给 3002H 单元。

解: 程序如下:

```

MOVDPTR, # 3000H      ; 设置第一个数据地址指针
MOVB, @DPTR           ; 取第一个数据
MOV R0, B             ; 将第一个数据送 R0
INCDPTR              ; 设置第二个数据地址指针
MOVB, @DPTR           ; 取第二个数据
ADD A, R0             ; 两个数据相加
RRC A                ; 带进位 C 右移一位, 相当除以 2
INCDPTR              ; 设置结果单元地址指针
MOVB, @DPTR, A        ; 存平均值
RET

```

5. 分别写出实现如下功能的程序段。

(1) 将片内 RAM30H 的中间 4 位, 31H 的低 2 位, 32H 的高 2 位按序拼成一个新字节, 存入 33H 单元。

(2) 将 DPTR 中间 8 位取反, 其余位不变。

解: 程序如下:

```

(1)  MOV A, 30H
      ANL A, # 3CH      ; 30H 中间 4 位送 A
      RL A              ; 将中间 4 位移至高 4 位
      RL A
      MOV 33H, A
      ANL 31H, # 3      ; 取 31H 的低 2 位, 高 6 位为 0
      ANL 32H, # 0C0H   ; 取 32H 的高 2 位, 低 6 位为 0
      MOV A, 31H        ; 31H 的低 2 位送 A
      ORL A, 32H        ; 32H 单元高 2 位存入 A 中高 2 位, 31H 单元低
                        ; 2 位存入 A 中低 2 位, A 中间 4 位为 0
      RL A              ; 将 31H 低 2 位、32H 高 2 位移至 A 低 4 位
      RL A
      ORL 33H, A        ; 将 31H 低 2 位、32H 高 2 位拼入 33H 中

(2)  XRL DPH, # 0FH
      XRL DPL, # 0F0H

```

6. 在 ROM 中有一个 10 以内的平方值表, 根据 R0 中的数查出平方值, 若平方值超出表的范围, 则将 FFH 存入 A。

解: 程序如下:

```

ORG 0030H
MOVDPTR, # TAB      ; TAB 为表首地址
MOV A, R0

```

```

        CJNE     A, # 10, NEXT
NEXT:   JNC      NEXT1
        MOVC     A, @A + DPTR
        SJMP     NEXT2
NEXT1:  MOV      A, # 0FFH
NEXT2:  SJMP     NEXT2
TAB:    0, 1, 4, 9, 16, 25, 36, 49, 64, 81

```

7. 列举四条能使累加器 A 清 0 的指令。

解：(1) MOV A, # 00H
 (2) XRL A, ACC
 (3) ANL A, # 00H
 (4) CLR A

8. 复位后，跳过中断程序区，重新设置堆栈，并将工作寄存器切换至 3 区。

解：程序如下：

```

        ORG      0000H
        LJMP     0030H
        ORG      0030H
        MOV      SP, # 60H
        SETB     RS1
        SETB     RS0
        .....

```

9. 已知当前 PC 值为 2000H，请用两种方法将程序存储器 20F0H 中的常数送入累加器 A 中。

答：方法一：以 PC 作为基址寄存器。

```

MOV     A, # 0EDH      ; 当前单字节指令地址为 2000H，偏移量送 A
MOV     A, @A + PC     ; 双字节指令，(20F0H) → A

```

方法二：以 DPTR 做为基址寄存器。

```

MOV     DPTR, # 20F0H
MOV     A, # 00H
MOVC    A, @A + DPTR

```

10. 如果 R0 的内容为 0，将 R1 置为 0；如果 R0 的内容非 0，置 R1 为 FFH，试编写子程序。

解：程序如下：

```

        MOV     A, R0
        JZ      ZE
        MOV     R1, # 0FFH
        RET
ZE:     MOV     R1, # 0
        RET

```

第四章 MCS—51 单片机存储器扩展

第一节 学习要求与内容概要

一、学习要求

- (1) 了解存储器的分类及特点,熟悉常用的 EPROM、Flash ROM、RAM 和 E²PROM 芯片的性能及引脚功能。
- (2) 熟悉三总线的概念,掌握通过总线进行最小系统设计及外部扩展的方法。
- (3) 掌握 MCS—51 系列单片机最小应用系统组成结构;掌握常用扩展方法,包括 RAM 和 ROM 混合扩展方法。
- (4) 掌握 $\overline{WR}$ 和 $\overline{RD}$ 控制信号线的含义及读/写程序设计方法。

二、常用存储器学习内容概要

存储器主要分为随机读取存储器 (RAM) 和只读存储器 (ROM) 两大类。

1. 随机读取存储器

读取、存入数据的时间很短,断电后 RAM 中存放的信息将丢失。随机读取存储器主要包括:静态 RAM 和动态 RAM 两类,后者需要一个刷新信号端。

2. 只读存储器

断电后 ROM 中存放的信息不会丢失,适宜存放程序、常数、表格等,又称为程序存储器。具体包括:

- (1) 掩膜式 ROM, 厂家生产芯片时将程序烧结写入芯片,用户只能读出程序而不能改写。已不常用。
- (2) 可编程只读存储器 (PROM), 用户将程序写入芯片,但是程序一经写入就不能改写了。已不常用。
- (3) 可擦除再编程只读存储器 (EPROM), 利用 EPROM 编程器写入储存内容,可用紫外线照射芯片背部的玻璃窗口进行擦除。
- (4) 电可擦除可编程只读存储器 (EEPROM), 采用在线电擦除方式,擦除、写入、读出的电源都用 +5V。目前写入的时间较长,约需 10ms,读出时间几百纳秒。
- (5) Flash 存储器:一种兼有紫外线擦除 EPROM 和电擦除 EEPROM 二者优点的新型非易失大容量存储器件,写入数据的时间较短。

三、单片机扩展三总线构成学习内容概要

1. 片外三总线

MCS—51 系列单片机用于系统扩展的片外引脚构成单片机的外部三总线结构:地址总线 (AB)、数据总线 (DB) 和控制总线 (CB)。8051 单片机的三总线结构如图 4-1 所示。

- (1) 地址总线 (Address Bus 即 AB)。地址总线用于传送地址信号,以便进行存储单元和

a) 片外扩展三总线 b) P0 口数据/地址线分离实现

I/O 端口的选择。它是单向的，只能由单片机发出。地址总线的数目决定可直接访问的存储单元的数目。

(2) **数据总线 (Data Bus 即 DB)**。数据总线用于传送数据, 由 P0 口担任, 即图 4-1 中 D0 ~ D7。MCS-51 系列单片机数据总线的位数与处理数据的字长一致, 且为双向传输。

(3) **控制总线** (Control Bus 即 CB)。用于系统扩展的控制总线主要有：

1) $\overline{\text{RD}}$ (P3.7)、 $\overline{\text{WR}}$ (P3.6) 作为扩展数据存储器或 I/O 端口读/写选通输出控制信号。执行 MOVX 时, 根据读/写要求使 $\overline{\text{RD}}$ 或 $\overline{\text{WR}}$ 有效。

2) $\overline{\text{PSEN}}$ 连接外扩 EPROM 的 $\overline{\text{OE}}$ 引脚, 作为扩展程序存储器的读选通信号, 取指令时自动生成。

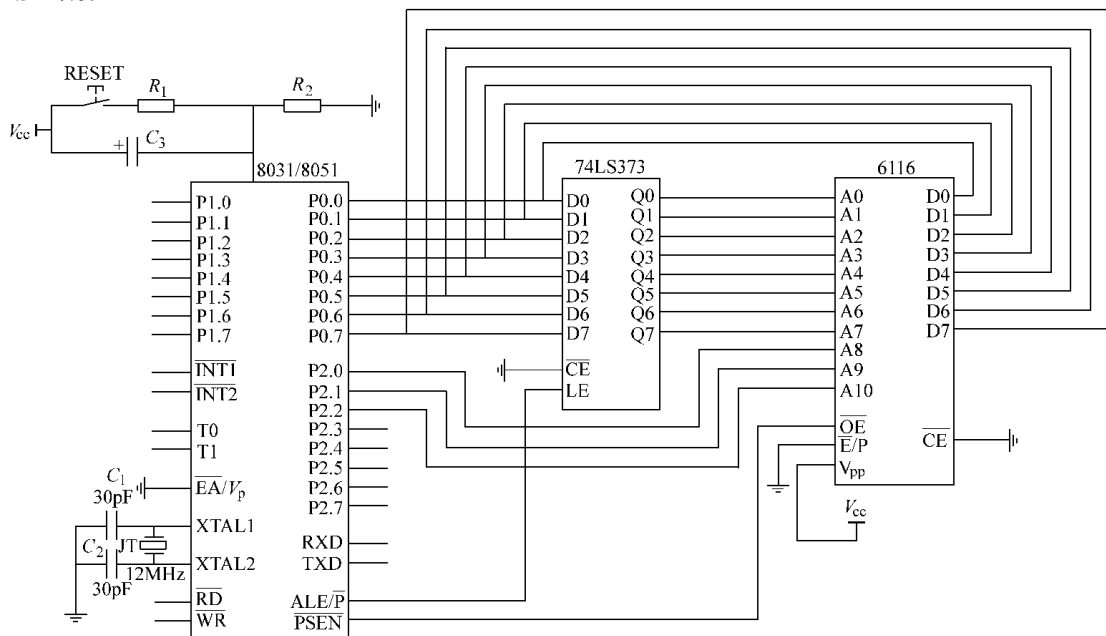


图 4-2 8031/8051 最小应用系统

3) ALE 作为地址锁存选通信号, 以实现 P0 口输出低 8 位地址的锁存, 是系统硬件生成的固定信号。

4) $\overline{EA}$ 作为内、外程序存储器选择信号, 一般根据系统扩展的情况由外电路设置。

2.8031/8051/89C51 最小应用系统

所谓最小应用系统是指维护单片机运行的最简单的配置系统。由于 51 系列不同型号单片机的内部资源不同, 组成的最小应用系统有所区别。廉价的 89C51 出现后, 最小应用系统就变得非常简单。

(1) 8031/8051 的最小应用系统。8051 内部有 4K 的掩膜 ROM, 对用户来说一般无法使用, 所以和 8031 一样, 也必须外扩 EPROM, 扩展电路如图 4-2 所示。

(2) 89C51 最小应用系统。89C51 片内有 4K 字节 Flash ROM, 构成最小应用系统时, 只要将单片机接上时钟电路所需的晶体及复位电路即可, 如图 4-3 所示。应该注意: 由于不需要外扩程序存储器, $\overline{EA}$ 接高电平; P0、P1、P2、P3 均可作 I/O 口使用。目前, 这种系统应用十分普遍。

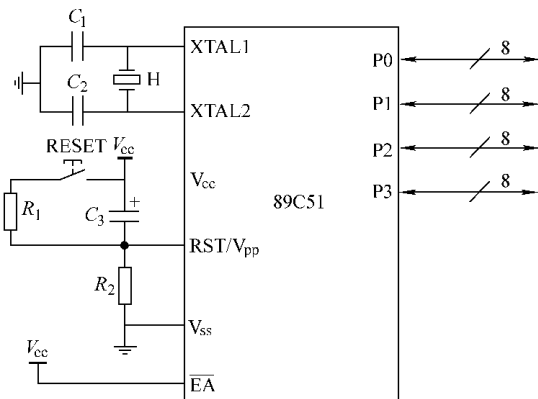


图 4-3 89C51 最小应用系统

四、单片机程序存储器扩展学习内容概要

若把图 4-2 中的 2716 更换为 2732、2764、27128 或 27256 等容量更大的 EPROM, 就可以使扩展的片外程序存储器的容量将依次达到 4KB、8KB、16KB、32KB。图 4-4 是 8751 利用 27128 扩展 16KB 程序存储器的原理图。

对于图 4-4 所示的扩展电路, 片外程序存储器 16K 字节单元的地址空间为: 0000H ~ 3FFFH。因为 $\overline{EA}$ 接 +5V, 故片外程序存储器中 4K 字节单元 0000H ~ 0FFFH 和片内程序存储器 4KB 字节单元重复, 故这一段存储器空间将作废。

用于扩展 EPROM 操作的控制线主要是 $\overline{OE}$, 它由单片机的 $\overline{PSEN}$ 连接控制。

五、单片机数据存储器扩展学习内容概要

单片机扩展 RAM 的目的在于用其存放数据, 图 4-5 8051 利用 62256 扩展 32KBRAM 的电路。扩展 RAM 的控制线有两条: 读选通信号线 $\overline{OE}$ 和写允许信号线 $\overline{WE}$, 分别与单片机的 $\overline{RD}$ 与 $\overline{WR}$ 引脚连接。

向片外 RAM 写数据的程序指令段如下:

```
MOV DPTR, # addr16; addr16 为外 RAM 要寻址字节单元 16 位地址
MOVX @DPTR, A; A 的内容存入由 DPTR 内容寻址的外 RAM 字节单元
MOVX @Ri, A; A 内容存入由 Ri (R0、R1) 内容寻址的外 RAM 单元
从外 RAM 读数据, 程序指令段如下:
```

```
MOV DPTR, # addr16; 外 RAM 要寻址字节单元地址 16 位送数据指针 DPTR
MOVX A, @DPTR; 由 DPTR 内容寻址的外 RAM 字节单元内容读入 A 中
```

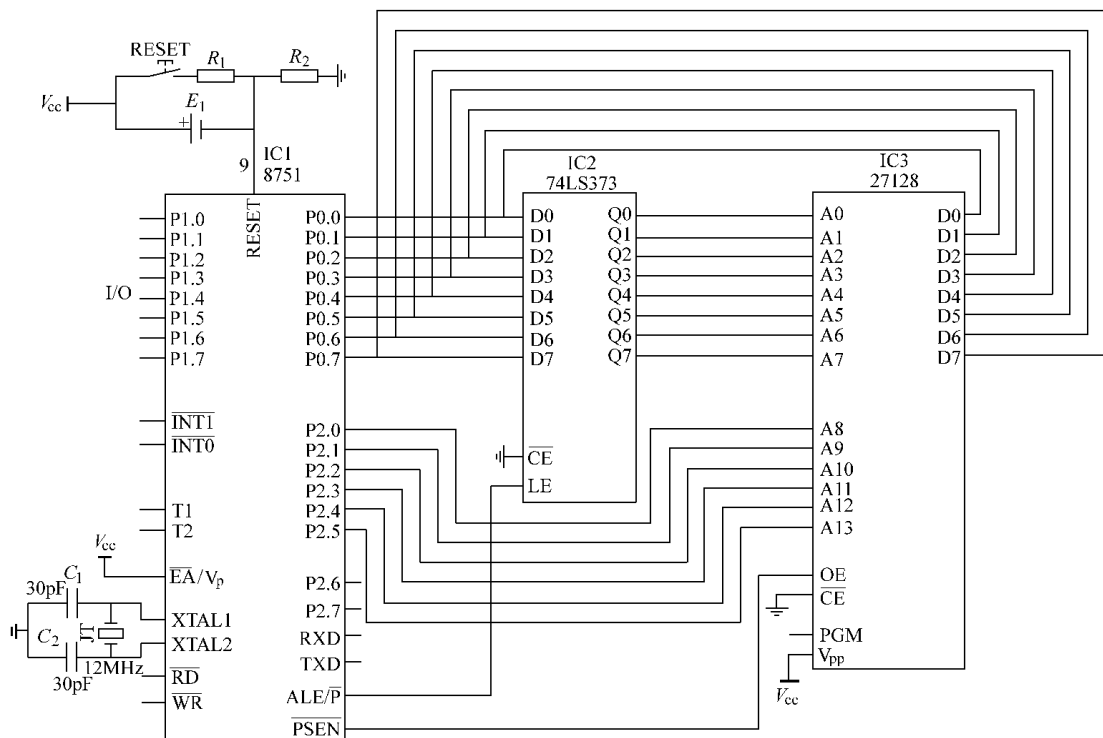


图 4-4 应用一片 27128EPROM 的扩展电路

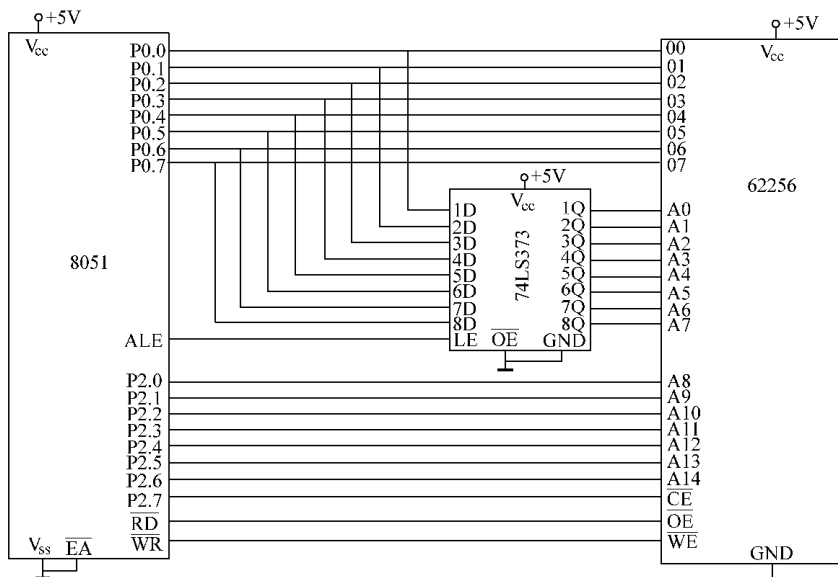


图 4-5 8051 扩展 32KBRAM

MOVX A, @Ri ; 由 Ri 内容寻址的外 RAM 字节单元内容读入 A 中

六、选址译码方法与混合扩展学习内容概要

1. 选址译码方法

(1) 线选择法寻址：将 P2 口的某一根地址线与存储器（或 I/O 口）的片选信号直接相连。线选法简单，无需外加任何其他的译码电路，适用于不太复杂的系统。

图 4-6 是利用两片 6264 和两片 2764 分别扩展 16K 字节数据存储器 and 16K 字节程序存储器的系统连接图，利用 P2.5 作为 IC1 和 IC3 的片选控制信号线，P2.5 的非逻辑作为 IC2 的片选控制信号线，P2.6 作为 IC4 的片选控制信号线。

每个芯片地址空间的计算，应该是在本芯片被选中而其他同类别芯片未被选中的情况下计算得到的。所谓同类别芯片，指的是所有扩展程序存储器芯片属于程序存储器类，而所有扩展数据存储器芯片和 I/O 端口属于数据存储器类。

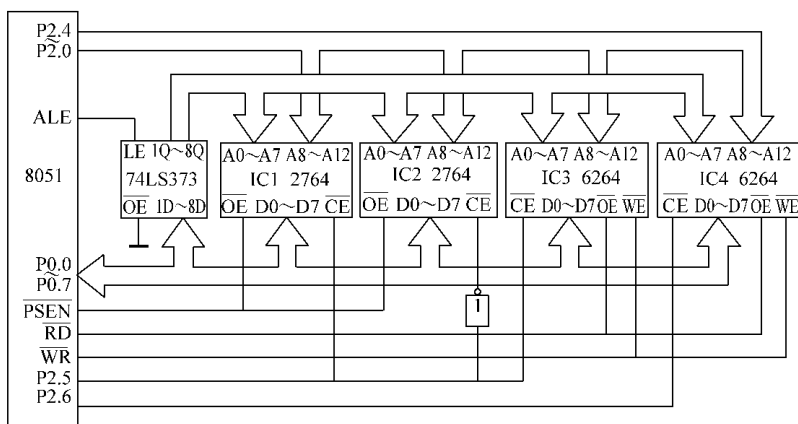


图 4-6 线选法扩展 16K 字节数据存储器 and 16K 字节程序存储器

值得指出的是，同一个单片机系统中数据存储器芯片和程序存储器芯片可以具有相同的地址空间而不会造成混乱，这主要是由 CPU 取指令或者存、取数据时分别给出不同的控制信号 $\overline{\text{PSEN}}$ 和 $\overline{\text{RD}}$ 或 $\overline{\text{WR}}$ 决定的。尽管送出的地址相同， $\overline{\text{PSEN}}$ 有效时，CPU 操作的对象是程序存储器芯片，而 $\overline{\text{RD}}$ 或 $\overline{\text{WR}}$ 有效时，CPU 操作的对象则是数据存储器芯片。

(2) 地址译码法寻址。当线选法所需的地址选择线多于可用的地址线时，一般采用地址译码法。译码法就是利用译码器，如 74LS138，对单片机的某些高位地址线进行译码，其译码器的输出作为存储器（或 I/O 口）的片选信号。这种方法存储空间连续，能有效地利用存储空间，适用于多存储器、多 I/O 口的扩展。

地址译码法选址必须采用地址译码器。图 4-7 是采用 74LS138 译码器扩展存储器的一个实例。

2. 程序存储空间和数据存储空间的混合

图 4-8 所示为 8051 以混合方式外扩 1 片 8KBE² PROM2864 的系统连接图。将 8051 的 $\overline{\text{RD}}$ 信号和 $\overline{\text{PSEN}}$ 信号相“与”，其输出控制 2864 的读允许片选端 $\overline{\text{OE}}$ ，使程序空间和数据空间混合。在执行 MOVX 指令时，产生 $\overline{\text{RD}}$ 或 $\overline{\text{WR}}$ 信号，从 2864 读取或写入数据，而执行 2864 中的程序时， $\overline{\text{PSEN}}$ 选通 2864 的 $\overline{\text{OE}}$ 端，从中取出指令代码到 CPU 分析执行。

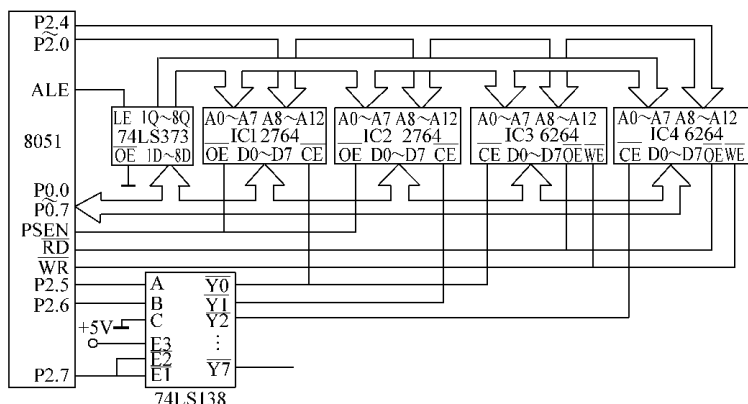


图 4-7 译码器扩展存储器

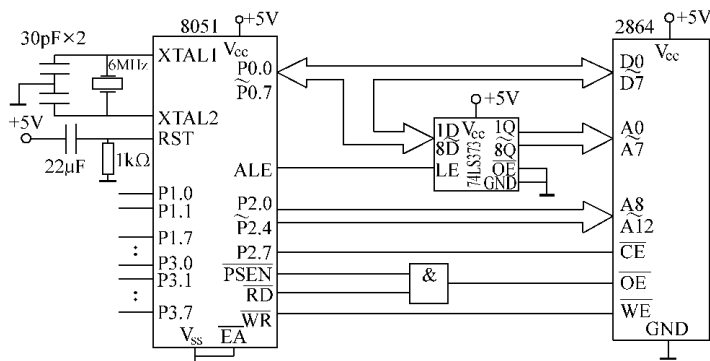


图 4-8 8051 混合扩展系统连接图

第二节 原教材习题及解答

5-1 什么是 ROM 和 RAM？简述它们的具体分类与特点？常用的芯片各有哪些？芯片功能引脚各有哪些？

答：ROM 即只读存储器。其分类是：1) 掩模 ROM，厂家生产芯片时烧结写入存储内容，再不能更改；2) PROM (Programmable ROM)，芯片出厂后可以进行一次特殊方式写入内容的 ROM；3) 可擦除 ROM，存储内容可擦除，然后由编程器重新写入内容的 ROM。ROM 的特点是：在线只能读，不能写，断电后内容保存。

常用的 EPROM 芯片有：2764 (8K × 8 位)，27128 (16K × 8 位)，27256 (32K × 8 位) 等型号；常用的并行 E²PROM 芯片有：2816 (2K × 8 位)，2864 (8K × 8 位) 等型号。

RAM 即随机读写存储器。其分类是：

RAM 分为 TTL 电路制成的双极型 RAM 和 MOS 电路制成的单极型 RAM，而 MOS 型 RAM 又分为静态 RAM 和动态 RAM。双极型 RAM 读写速度高、功耗不大、集成度低，价格较贵；

单极型 RAM 集成度高、功耗低、价格便宜,故一般微机系统使用 MOS 型 RAM。RAM 的特点是在线可读可写,断电后内容消失。目前常用的静态 RAM 芯片有 6116 ($2K \times 8$ 位), 6264 ($8K \times 8$ 位) 和 62256 ($32K \times 8$ 位) 等型号。芯片引脚功能详见课本。

5-2 简述单片机系统中的三大总线。系统扩展时单片机提供的三大总线信号有哪些? 怎样生成数据和地址线?

答: 单片机系统中使用的三大总线为: 地址总线、数据总线和控制总线。

系统扩展时单片机提供的三大总线信号有:

(1) 地址总线: P0 低 8 位地址线, P2 高 8 位地址线。

(2) 数据总线: P0 口 8 位数据双向总线。

(3) 控制总线: MCS—51 单片机的控制总线包括: $\overline{ALE}$ 、 $\overline{PSEN}$ 、 $\overline{EA}$ 、 $\overline{RD}$ 、 $\overline{WR}$ 。

P0 口分时复用为低 8 位地址线和 8 位数据总线, 当 CPU 访问外 RAM 或外部程序存储器时, 由 P0 口首先送出低 8 位地址, 再做 8 位数据线使用。

5-3 简述 8051 单片机 CPU 访问片外扩展程序存储器的过程。

答: 外部扩展的程序存储器没有地址锁存功能, 扩展程序存储器时必须扩展低 8 位地址锁存器, 如 74LS373 等。CPU 访问片外扩展程序存储器时, 程序指针 PC 的内容就是即将访问外部程序存储器的 16 位单元地址, 它的高 8 位 (PCH) 由 P2 口送出, 而低 8 位 (PCL) 由 P0 口送出, 同时 $\overline{ALE}$ 引脚输出地址锁存允许信号, 把 P0 口输出的低 8 位地址锁存在地址锁存器 (如 74LS373) 中, 这样 P2 口和地址锁存器输出的内容形成访问外部程序存储器的 16 位地址, 进而寻址程序存储器单元, CPU 使 $\overline{PSEN}$ 信号有效, 把被寻址单元中的内容 (指令码) 由数据线输出, 经 P0 口输入至 CPU 执行。

5-4 简述 8051 单片机 CPU 访问片外扩展数据存储器的过程。

答: 外部扩展的数据存储器没有地址锁存功能, 扩展数据存储器时必须扩展低 8 位地址锁存器, 如 74LS373 等。当 CPU 使用 $MOVX A, @DPTR$ (或者 $MOVX @DPTR, A$) 指令访问片外扩展数据存储时, 数据指针 DPTR 的内容就是即将访问外部数据存储器的 16 位单元地址, 它的高 8 位 (DPH) 由 P2 口送出, 而低 8 位 (DPL) 由 P0 口送出, 同时 $\overline{ALE}$ 引脚输出地址锁存允许信号, 把 P0 口输出的低 8 位地址锁存在地址锁存器 (如 74LS373) 中, 这样 P2 口和地址锁存器输出的内容形成访问外部数据存储器的 16 位地址, 进而寻址数据存储器单元, 这时 CPU 使 $\overline{RD}$ (或者 $\overline{WR}$) 信号有效, 把被寻址单元中的内容 (数据) 由数据线输出, 经 P0 口输入至 CPU 的 A 中。

CPU 使用 $MOVX A, @R0$ (或者 $MOVX @R0, A$) 指令访问片外扩展数据存储器的过程和上面类似, 请自行分析。

5-5 画出 8031 单片机采用 2 片 2764EPROM 芯片扩展 16KB 程序存储器的电路图。

解: 扩展电路如图 4-9 所示。

5-6 画出 8031 单片机扩展 1 片 2764 和 1 片 6264 的电路图, 并指出各个芯片的地址范围。

解: 一种扩展电路如图 4-10 所示。按照此硬件电路图, 地址范围是:

2764: 0000H ~ 1FFFH; 6264: 0000H ~ 1FFFH。

5-7 设有一个 8051 单片机系统片外扩展了 RAM, 请编写将片内 RAM 中从 30H 单元开始的 10 个字节单元内容依次移到片外 RAM 从 0100H 单元开始的 10 个字节单元中的程序。

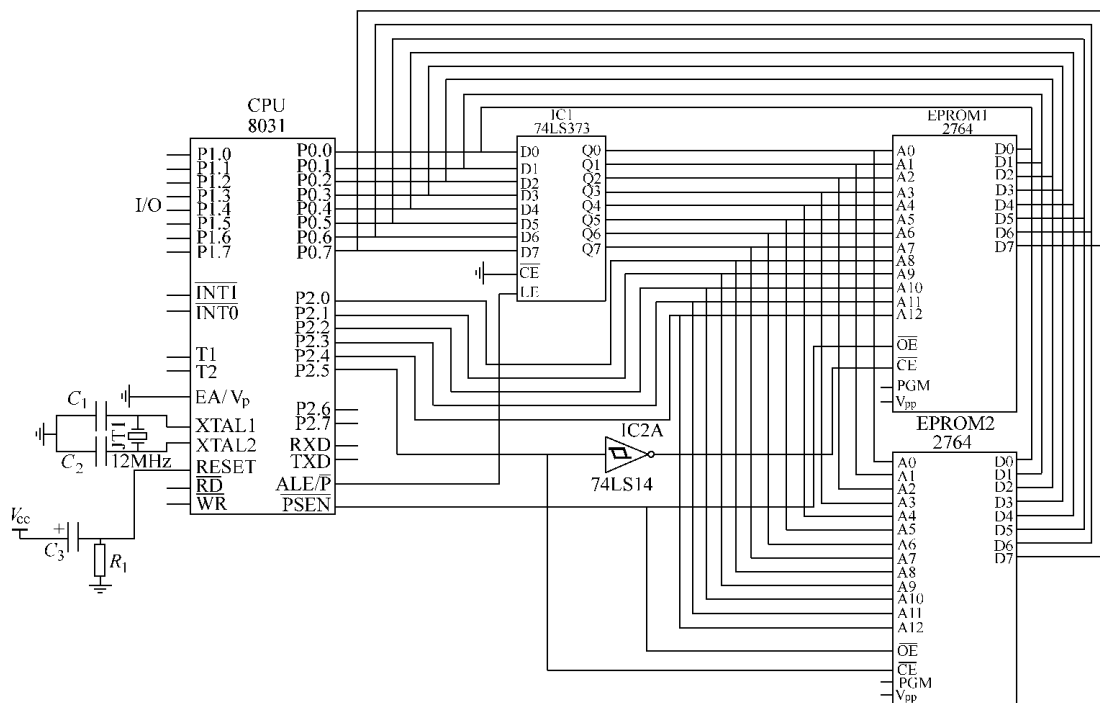


图 4-9 2 片 2764EPROM 扩展 16KB 程序存储器电路图

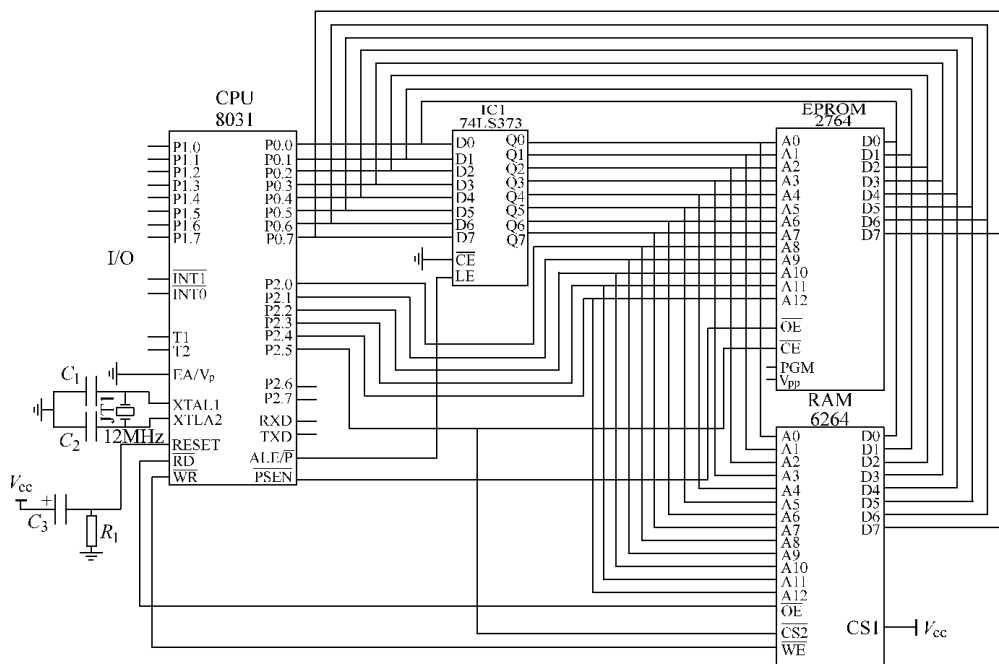


图 4-10 1 片 2764 和 1 片 6264 扩展存储器电路图

解：程序如下：

```

MOV R0,    # 30H
MOV DPTR,  # 0100H
MOV R7,    # 0AH
LOOP:      MOV A,      @R0
           MOVX  @DPTR, A
           INC   R0
           INC   DPTR
           DJNZ  R7,    LOOP
HERE:      SJMP   HERE

```

5-8 图 4-11 给出的是 8031 单片机以总线形式扩展 4 片 8KB ~ 8 位存储器芯片连线示意图, 请计算每片存储器芯片的地址范围。

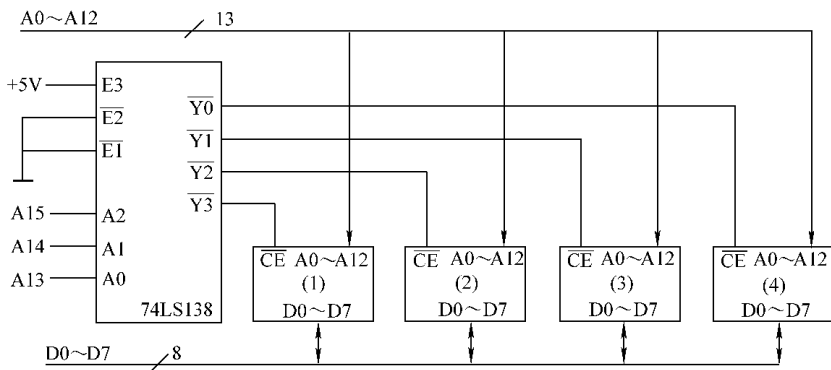


图 4-11 8031 扩展 4 片存储器芯片连接示意图

答：根据译码器 74LS138 的真值表，当 4 个芯片分别被选中时，计算出 4 个芯片的地址为：

$\overline{Y0}$ 选中的芯片：0000H ~ 1FFFH； $\overline{Y1}$ 选中的芯片：2000H ~ 3FFFH

$\overline{Y2}$ 选中的芯片：4000H ~ 5FFFH； $\overline{Y3}$ 选中的芯片：6000H ~ 7FFFH

第三节 增选习题及解答

一、填空题

1. 若不使用 MCS—51 片内存储器，引脚 $\overline{EA}$ 必须接_____。
2. 当 MCS—51 引脚 ALE 信号有效时，表示从 P0 口稳定地送出了_____地址。
3. MCS—51 的 P0 口作为输出端口时，每位能驱动_____个 SL 型 TTL 负载。
4. MCS—51 系统中，当 $\overline{PSEN}$ 信号有效时，表示 CPU 要从_____存储器读取信息。
5. 当使用 8751，且 $\overline{EA}=1$ ，程序存储器地址小于_____时，访问的是片内 ROM。
6. MCS—51 可扩展片外 RAM _____ KB，但当外扩 I/O 口后，其外部 RAM 寻址空间将_____。
7. 计算机的系统总线有地址总线、控制总线和_____总线。

8. MCS-51 在外扩 ROM、RAM 或 I/O 时, 它的地址总线是_____。

9. 80C51 含_____ KB 掩膜 ROM。

10. 80C51 在物理上有_____个独立的存储器空间。

11. 8751 是 EPROM 型, 内含_____ KB EPROM; 89C51 是 Flash ROM 型, 内含_____ KB Flash ROM。

12. 12 根地址线可寻址_____ KB 存储单元。

13. 8031 其 EA 引脚必须接_____。可作通用 I/O 的至少有 P_____口的 8 条 I/O 线, 最多还可加上 P_____口的 8 条 I/O 线。P_____口作地址/数据总线, 传送地址码的_____ 8 位; P_____口作地址总线, 传送地址码的_____ 8 位。MOVX 指令用来对_____ RAM 进行读写操作。

填空题参考答案:

1. 接地 2. 低 8 位 3. 8 4. 程序 5. 1000H 6. 64 变小 7. 数据总线 8. P0、P2
9. 4 10. 4 11. 4, 4 12. 4 13. 地; 1, 3, 0, 低; 2, 高; 外部

二、选择题

1. 6264 芯片是 ()。

A. E²PROM B. RAM C. Flash ROM D. EPROM

2. 产生 $\overline{WR}$ 信号的指令是 ()

A. MOVX A, @DPTR B. MOVC A, @A + PC
C. MOVX A, @A + DPTR D. MOVX @DPTR, A

3. 若某存储器芯片地址线为 12 根, 那么它的存储容量为 ()。

A. 1KB B. 2KB C. 4KB D. 8KB

4. 使用 8751, 且 $\overline{EA} = 1$, 则可以外扩 ROM ()。

A. 64KB B. 60KB C. 58KB D. 56KB

5. 当 8031 外扩程序存储器 8KB 时, 需使用 EPROM2716 ()。

A. 2 片 B. 3 片 C. 4 片 D. 5 片

6. 某种存储器芯片是 8KB × 4/片, 那么它的地址线根数是 ()。

A. 11 种 B. 12 根 C. 13 根 D. 14 根

7. 执行 MOVX A, @DPTR 指令时, MCS-51 产生的控制信号是 ()。

A. $\overline{PSEN}$ B. ALE C. $\overline{RD}$ D. $\overline{WR}$

8. 执行 MOVX @DPTR, A 指令时, MCS-51 产生 () 控制信号。

A. $\overline{PSEN}$ B. $\overline{WR}$ C. ALE D. $\overline{RD}$

9. 74LS138 芯片是 ()。

A. 驱动器 B. 译码器 C. 锁存器 D. 编码器

10. 当 ALE 信号有效时, 表示 ()。

A. 从 ROM 中读取数据 B. 从 P0 口可靠地送出地址低 8 位
C. 从 P0 口送出数据 D. 从 RAM 中读取数据

选择题参考答案:

1. B 2. D 3. C 4. B 5. C 6. C 7. C 8. B 9. B 10. B

三、简答题

1. 若 80C51 的片内 ROM 内容已不符合要求, 那么片内硬件如何继续使用?

答 把 80C51 的 $\overline{\text{EA}}$ 引脚接地, 片外扩接 EPROM 芯片, 就等于宣布片内 ROM 作废, 完全执行片外 EPROM 中的程序。这样, 片内硬件资源不受影响, 可继续使用。

2. 在 8051 扩展系统中, 程序存储器和数据存储器共用 16 位地址线和 8 位数据线, 为什么两个存储空间不会发生冲突?

答: 程序存储器和数据存储器虽然共用 16 位地址线和 8 位数据线, 但由于数据存储器的读和写由 $\overline{\text{RD}}$ (P3.7) 和 $\overline{\text{WR}}$ (P3.6) 信号控制, 而程序存储器由读选通信号 $\overline{\text{PSEN}}$ 控制, 因此, 两者虽然共处同一地址空间, 但由于控制信号不同, 所以不会发生总线冲突。

3. 为什么当 P2 口作为扩展存储器的高 8 位地址后, 不再适宜作通用 I/O 口了?

答: P2 口用做扩展存储器的高 8 位地址总线后, 即使没有全部占用, 空余的几根也不宜作通用 I/O 线, 否则会给软件编写及其使用带来不必要的麻烦。

四、存储器扩展电路设计和扩展地址空间计算

1. 使用 89C51 外扩 8KB RAM。请画出系统电路原理图, 写出地址分布。

解: 89C51 单片机外扩 8KB RAM 的接口电路如图 4-12 所示。如把 P2 口没有用到的高位地址假设为 1 状态, 则 6264 的地址范围为 C000H ~ DFFFH。

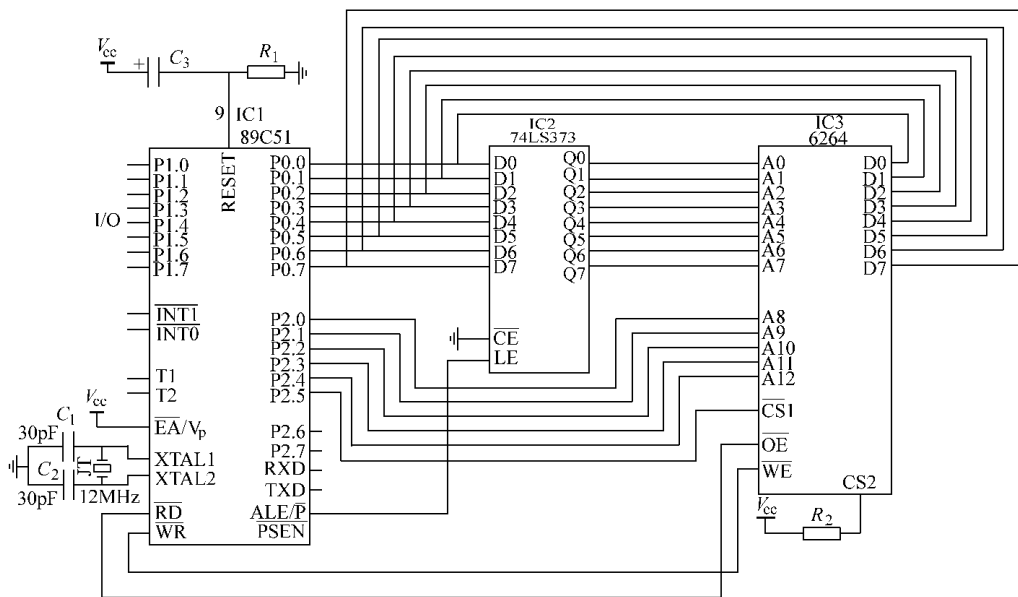


图 4-12 89C51 单片机外扩 8KB RAM 的接口电路

2. 利用 89C51 扩展 2KB RAM, 画出相应的连接电路图, 并说明地址分布。

解: 连接电路如图 4-13 所示。当没有使用的 P2 口地址线认作状态 0 时, 得到一组地址为 0000H ~ 07FFFH。

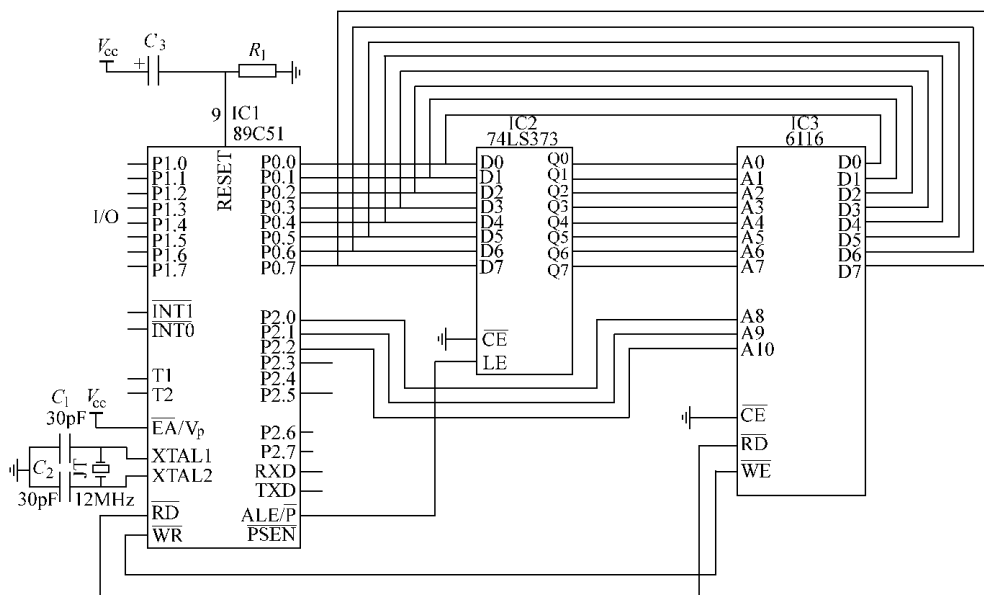


图 4-13 89C51 外扩 2KBRAM6116 的连接电路图

第五章 中断与中断系统

第一节 学习要求与内容概要

一、学习要求

- (1) 了解中断、中断源、中断请求、中断查询、中断优先级、中断嵌套、中断屏蔽、中断响应和中断处理过程等概念；
- (2) 掌握 MCS—51 系列单片机的中断系统的硬件结构；
- (3) 掌握 CPU 对中断响应的步骤，包括：保护断点、中断矢量的使用、中断允许或禁止的实现方法，以及主程序和中断服务程序的连接方法等；
- (4) 掌握中断服务程序的一般格式。

二、中断与中断系统学习内容概要

(1) 中断：当 CPU 正在执行某程序时，某一随机事件请求 CPU 迅速去处理，CPU 暂时中止现行的程序，转去处理所发生的事件（程序）。处理完该事件以后，再回到原来被中止的地方，继续原来的工作。

(2) 中断源：产生中断请求的事件。

(3) 中断请求或中断申请：中断源向 CPU 提出的处理请求。通常 CPU 为每个中断源设置一个中断请求触发器。当中断源向 CPU 发出中断请求时，把该触发器置“1”，且一直保持这一状态，直到 CPU 响应该中断请求后才能而且必须清除这一信号，以防止重复中断。

(4) 中断源识别：指当 CPU 要响应某个中断源的中断请求时，如何找到该中断源的中断服务程序入口地址，执行相应的中断处理程序，进行中断处理的方法。

(5) 中断优先级：指 CPU 响应各中断请求的一种先后次序。

(6) 多重中断或中断嵌套：当 CPU 正在执行中断服务程序时，又有新的中断源发出申请，若新来的中断优先级别更高，则 CPU 应该中断现行的中断服务程序，响应优先级别高的中断，在高级中断处理完后，再返回原先的中断服务程序。

(7) 中断屏蔽：每个中断源都对应一个中断屏蔽触发器来控制该中断源的申请信号能否送到 CPU，当该触发器的状态为“0”时，CPU 不响应该中断源的中断请求，这称为中断屏蔽（或中断禁止），反之为中断开放。

(8) 中断响应：CPU 暂时中止自身的事务（主程序），转去处理事件（中断服务）。中断响应的条件是：中断开放且当前指令执行完毕。

(9) 中断返回：中断返回指令把保存在堆栈中的断点地址送入 PC，返回到主程序被中断的地方继续执行原来的主程序。

一个完整的中断过程应包括中断请求、中断判优、中断响应、中断处理和中断返回。实现这种功能的部件称为中断系统。

三、MCS—51 系列单片机中断系统结构及中断控制学习内容概要

MCS—51 系列单片机中断系统的结构如图 5-1 所示。

MCS—51 系列单片机设立了 5 个中断源：外部中断 0 即 $\overline{\text{INT0}}$ ，外部中断 1 即 $\overline{\text{INT1}}$ ，定时器/计数器 T0 溢出中断、定时器/计数器 T1 溢出中断和串行口中断。提供 2 个中断优先级，中断处理程序可实现二级中断嵌套。

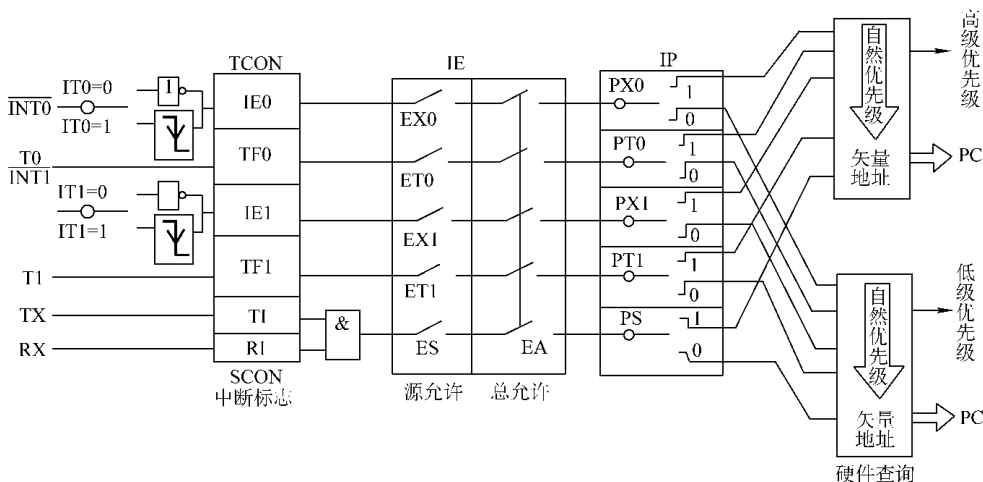


图 5-1 MCS—51 系列单片机中断系统的结构

MCS—51 中断系统有以下 4 个特殊功能寄存器：

(1) 定时器控制寄存器 TCON (用 6 位)。TCON 为定时器/计数器 T0 和 T1 的控制寄存器，同时也锁存 T0 和 T1 及外部中断 0 和 1 的中断标志等。TCON 的字节地址为 88H，可以位寻址。

在外部中断的电平触发方式中，必须注意两点：

- 1) 中断源必须持续请求，一直到 CPU 响应其中断为止；
- 2) 在中断服务程序返回之前，必须撤销中断请求信号，否则机器将以为又发生另一次中断请求。

(2) 串行口控制寄存器 SCON (用 2 位)，最低 2 位锁存串行口接收中断和发送中断标志 RI 和 TI。SCON 字节地址为 98H，可以位寻址。

值得注意的是：在计算机响应串行口的中断请求之后，不能由硬件而只能由软件将它的中断请求标志位清零。这是它与其他 4 种中断的不同之处。

(3) 中断允许寄存器 IE 控制中断源的开放或屏蔽。IE 的字节地址是 A8H，可以位寻址。

中断允许寄存器 IE 对中断的开放和关闭实现两级控制。所谓两级控制，就是有一个总的开关中断控制位 EA (IE.7) 和各中断源自己的中断允许控制位。当 EA = 0 时，屏蔽所有的中断申请，即任何中断申请都不被接受；当 EA = 1 时，CPU 开放中断，但各个中断源是否允许开放还要由 IE 的低 5 位的各对应控制位的状态进行中断允许控制。

(4) 中断优先级寄存器 IP 字节地址为 B8H，可按位寻址。若 CPU 正在执行高优先级的中断服务程序，则高优先级生效触发器的状态为 1，所有后来的中断都被禁止；若 CPU 正在执行低优先级的中断服务程序，所有同级的中断都被禁止，但开放高优先级的中断。

用户应根据需要对 IP 中的相应位进行置 1 或清 0 设置各中断源的优先级。置 1，相应的

中断源为高级中断；反之，清 0，相应的中断源为低级中断。当同时接收到几个同一优先级的中断请求时，响应优先级顺序排列如表 5-1 所示。

表 5-1 各中断源的矢量地址及优先级顺序

中断源	中断入口地址	同一级的中断优先级
外部中断 0	0003H	最高 ↓ 最低
外部中断 1	000BH	
定时器 T0 中断	0013H	
定时器 T1 中断	001BH	
串行口中断	0023H	

四、中断初始化程序框架

下面是中断初始化程序框架。在编写初始化程序时，应根据题目的实际要求，选择所需的中断，填写相应的中断控制字。

```
ORG    0000H
AJMP   MAIN
ORG    0003H           ; 外中断INT0中断入口地址
AJMP   W1
ORG    000BH           ; 定时器/计数器 T0 中断入口地址
AJMP   W2
ORG    0013H           ; 外中断INT1中断入口地址
AJMP   W3
ORG    001BH           ; 定时器/计数器 T1 中断入口地址
AJMP   W4
ORG    001BH           ; 串行口中断入口地址
AJMP   W5
ORG    0050H           ; 主程序
MAIN:  MOV    IP, # data1      ; 根据题目要求，设置中断优先级
       MOV    IE, # data2     ; 根据题目要求，选择开放中断源
HERE:  SJMP   HERE            ; 等待中断
```

第二节 原教材习题及解答

6-1 什么是中断？什么是中断源？

答：当 CPU 正在运行时，外部发生的某一随机事件（如一个电平的变化，一个脉冲沿的发生或定时器计数溢出等）请求 CPU 迅速去处理，于是，CPU 暂时中止现行的程序，转去处理所发生的事件。处理完该事件以后，再回到原来被中止的地方，继续工作。这种在程序执行过程中由于外界的原因而被中间打断的情况称为中断。产生中断的请求源称为中断源。

6-2 MCS—51 提供了哪几种中断源？在中断管理上有什么特点？各中断源中断优先权的高低是如何排列确定的？

答：MCS—51 单片机设立了 5 个中断源，其中 2 个为外部中断源，外部中断 0 即 $\overline{\text{INT0}}$ ，外部中断 1 即 $\overline{\text{INT1}}$ ，3 个为片内中断源（内部中断源），分别为定时器/计数器 T0 溢出中断、定时器/计数器 T1 溢出中断和串行口中断。由特殊功能寄存器 TCON 和 SCON 的相应位表示了各中断源的状态。提供 2 个中断优先级，中断处理程序可实现二级中断嵌套。各中断源可以由中断优先级控制寄存器 IP 程控为高优先级或低优先级中断。中断源的排列顺序由中断优先级控制寄存器和内部查询电路共同决定，并且各中断源可以由中断允许控制寄存器 IE 定义为开放或屏蔽状态。

6-3 MCS—51 响应中断的条件是什么？CPU 响应中断时，不同的中断源，其中断入口地址各是多少？

答：MCS—51 响应中断的基本条件有：

- (1) 由中断源发出中断请求。
- (2) 中断总允许位 $\text{EA} = 1$ ，即 CPU 开中断。
- (3) 发出中断请求的中断源所对应的中断允许位为 1，即该中断没有被屏蔽。

若上述条件满足，并且不存在任意一种下列情况，则 CPU 会响应中断：

- (1) CPU 正在执行同级或更高级中断服务程序。
- (2) 当前的指令周期还未结束。这样可保证指令的完整，防止中断返回后出现错误指令。

(3) 现行指令为 RETI 或者是访问 IE 或 IP 的指令，该指令以及紧接着的另一条指令还未执行完。为了保证 CPU 能正确返回，CPU 在执行 RETI 或访问 IE、IP 的指令后，至少需要再执行一条指令，才会响应新的中断请求。

若存在上述任何一种情况，则 CPU 将丢弃中断查询结果，此次中断申请无效，不会再被响应。若没有上述情况，CPU 将在紧接着的下一个机器周期内响应中断。

各中断源及与之对应的矢量地址略。

6-4 MCS—51 的外部中断有哪两种触发方式？它们对触发脉冲或电平有什么要求？应如何选择和设定？

答：MCS—51 的外部中断有两种触发方式：边沿触发方式和电平触发方式。对于边沿触发方式，CPU 在每一个机器周期 S5P2 期间采样外部中断 1（或 0）请求引脚的输入电平。如果在相继的两个机器周期采样过程中，一个机器周期采样到外部中断 1（或 0）请求信号为高电平，接着的下一个机器周期采样到外部中断 1（或 0）请求信号为低电平，出现了下跳沿，则使 IE1（或 IE0）置 1。直到 CPU 响应该中断时，才由硬件使 IE1（或 IE0）清 0。对于电平触发方式，CPU 在每一个机器周期 S5P2 期间采样外部中断 1（或 0）请求引脚的输入电平。若外部中断 1 请求信号为低电平，则使 IE1（或 IE0）置 1，若外部中断 1（或 0）请求信号为高电平，则使 IE1（或 IE0）清 0。可由 TCON 中对应的 IT0（或 IT1）位程控为低电平或下降沿有效。IT0（或 IT1）= 1 时，外部中断 0（或 1）程控为边沿触发方式。IT0（或 IT1）= 0 时，外部中断 0（或 1）程控为电平触发方式。

6-5 MCS—51 单片机的中断系统中有几个优先级？如何设定？

答：MCS—51 单片机的中断系统中有两个中断优先级。每一个中断请求源均可编程为高

优先级中断或低优先级中断，可以实现二级中断嵌套。MCS—51 的中断系统中设置了一个中断优先级寄存器 IP，可按位寻址。

IP 中的低 5 位为各中断源优先级的控制位，可用软件来设定。若某几个控制位为 1，则相应的中断源就规定为高级中断；反之，若某几个控制位为 0，则相应的中断源就规定为低级中断。

6-6 某 MCS—51 系统用于定时测试压力、温度，定时用 T0 来实现，压力超限和温度超限的报警信号分别由INT0和INT1输入，中断优先权排列顺序依次为压力超限→温度超限→定时检测，试确定专用寄存器 IE 和 IP 的内容，并编写初始化程序。

答：8051 对中断源的开放或屏蔽是由中断允许寄存器 IE 控制的。本题目需要开放 T0、INT0和INT1，并且不要忘记开放中断允许总控制位，因此 IE 为：

IE	1			0	0	1	1	1
----	---	--	--	---	---	---	---	---

即 IE = 87H。

按照内部查询的顺序，中断优先权的排列顺序依次为： $\overline{\text{INT0}} \rightarrow \text{T0} \rightarrow \overline{\text{INT1}}$ ，也就是说，定时检测的优先权高于温度超限，这与题目要求不符，因此，应该将 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 设定为高优先级，将 T0 设定为低优先级，所以 IP 为：

IP				0	0	1	0	1
----	--	--	--	---	---	---	---	---

即 IP = 05H。

初始化程序如下：

```
ORG    0000H
AJMP   MAIN
ORG    0003H           ; 外中断 $\overline{\text{INT0}}$ 中断入口地址
AJMP   W1
ORG    000BH           ; 定时器/计数器 T0 中断入口地址
AJMP   W2
ORG    0013H           ; 外中断 $\overline{\text{INT1}}$ 中断入口地址
AJMP   W3
ORG    0050H           ; 主程序
MAIN:  MOV    IP, # 05H      ; 设置中断优先级
        MOV    IE, # 87H    ; 选择开放中断源
HERE:  SJMP   HERE          ; 等待中断
```

第三节 增选习题及解答

一、填空题

1. 当使用慢速外设时，最佳的传输方式是_____。
2. MCS—51 单片机有_____个中断源，有_____个中断优先级，优先级由软件填写特殊功能寄存器_____加以选择。

3. MCS—51 单片机中, T0 中断服务程序入口地址为_____。
4. 外部中断请求标志位是_____和_____。
5. MCS—51 单片机晶振频率为 12MHz, 响应单重中断的最短时间是_____, 最长时间是_____。
6. MCS—51 单片机外部中断有_____种中断信号触发方式, 若选用外部中断 0 为边沿触发方式, 则需用指令_____。
7. MCS—51 单片机中断系统选用外部中断 1 和定时器/计数器 T1 中断为高优先级, 则中断优先级控制字 (IP) = _____。
8. 对采用电平触发方式的外部中断, 必须在中断返回前_____。
9. _____指令以及任何访问_____和_____寄存器的指令执行完后, CPU 不能马上响应中断。

填空题参考答案:

1. 中断 2. 5, 2, IP 3. 000BH 4. IE0, IE1 5. 3 μ s, 8 μ s 6. 2, SETB IT0 7. 0CH 8. 撤销中断请求信号 9. RETI, IE, IP

二、选择题

1. CPU 响应 () 中断时, 不会由 CPU 自动清除中断请求标志。
 - (A) 外部中断 0
 - (B) 外部中断 1
 - (C) 串行口中断
 - (D) 定时器/计数器 T0 溢出中断
 - (E) 定时器/计数器 T1 溢出中断
2. MCS—51 中, CPU 正在处理定时器/计数器 T1 中断, 若有同一优先级的外部中断 $\overline{\text{INT0}}$ 又提出中断请求, 则 CPU ()。
 - (A) 响应外部中断 $\overline{\text{INT0}}$
 - (B) 继续进行原来的中断处理
 - (C) 发生错误
 - (D) 不确定
3. 中断服务程序的最后一条指令必须是 ()。
 - (A) END
 - (B) RET
 - (C) RETI
 - (D) AJMP
4. 串行口中断的入口地址是 ()。
 - (A) 0003H
 - (B) 000BH
 - (C) 0013H
 - (D) 001BH
 - (E) 0023H
5. 在中断服务程序中, 至少应有一条 ()
 - (A) 传送指令
 - (B) 转移指令
 - (C) 加法指令
 - (D) 中断返回指令
6. MCS—51 单片机响应中断时, 下列哪种操作不会自动发生 ()。
 - (A) 保护现场
 - (B) 保护 PC
 - (C) 找到中断入口
 - (D) 转入中断入口
7. MCS—51 单片机的中断允许触发器内容为 83H, CPU 将响应的中断请求是 ()。
 - (A) $\overline{\text{INT0}}$, $\overline{\text{INT1}}$
 - (B) T0, T1
 - (C) T1, 串行接口
 - (D) $\overline{\text{INT0}}$, T0
8. 若 MCS—51 单片机的中断源都编程为同级, 当它们同时申请中断时, CPU 首先响应 ()。
 - (A) $\overline{\text{INT1}}$
 - (B) $\overline{\text{INT0}}$
 - (C) T1
 - (D) T0
9. 执行指令: MOV IE, # 03H 后, MCS—51 单片机将响应的中断有 ()。
 - (A) 1 个
 - (B) 2 个
 - (C) 3 个
 - (D) 0 个
10. 各中断源发出的中断请求信号, 都会标记在 MCS—51 单片机系统中的 ()。

(A) IE (B) TMOD (C) TCON/SCON (D) IP

选择题参考答案：

1. C 2. B 3. C 4. E 5. D 6. A 7. D 8. B 9. D 10. C

三、简答题

1. MCS—51 单片机提供几种中断？按中断优先级写出各中断源。

答：MCS—51 单片机提供了 5 种中断，按照中断优先级别由高到低为： $\overline{\text{INT0}}$ 、T0、 $\overline{\text{INT1}}$ 、T1 和串行口中断。

2. 中断处理和 CPU 调用子程序处理有什么异同点？

答：中断处理和 CPU 调用子程序处理都要中断原来执行的主程序，转去执行相应的处理程序。中断处理程序从相应的中断矢量地址开始执行，最后一条指令应是返回指令“RETI”。执行中断返回指令 RETI 的结果是清除“优先级生效”触发器，以使其他同级或低级的中断能被响应。然后，从堆栈中弹出两个字节到 PC，使程序回到原来被中断处继续执行。CPU 调用子程序处理时，会无条件转移到子程序处执行。子程序的最后一条指令是“RET”。RET 指令也能返回到原来被中断的地方，但不能恢复中断控制系统。因此，中断返回一定要用 RETI 指令而不能以 RET 指令代替。

3. 什么是中断矢量？若某个中断源的中断服务程序的字节数超出了 8 个单元，应如何处理？

答：中断矢量是指每一中断源均有一矢量地址，用来存放中断服务程序入口地址，它具有一定的指向性。若中断服务程序字节数超出了 8 个单元，可将其放于其他位置，并在中断入口地址处放入跳转指令，指出中断服务程序的起始地址。

4. 保护断点和保护现场各解决什么问题？

答：保存断点：将 PC 中下一条指令的地址入栈保护起来，以便中断返回时，能继续从断点处执行原来的主程序。

保护现场：中断服务程序一般要使用某些寄存器和存储器，如果主程序也用到这些资源，则在中断服务程序开始就应用指令进行现场保护，并在中断返回前恢复现场，以保证中断返回后主程序能正常运行。一般使用入栈指令来保护现场。

5. 简述 8051 单片机的中断响应过程。

答：8051 单片机的中断响应包括 5 个步骤：

(1) 中断请求：首先要发出中断请求，由 CPU 在每个机器周期顺序采样每个中断源。

(2) 中断判优：解决请求中断的事件优先级的顺序问题。CPU 在下一个机器周期按优先级顺序查询中断标志，如查询到某个中断标志为 1，将在接下来的机器周期按优先级进行中断处理。

(3) 中断响应：中断正在进行的处理任务，转向中断请求相对应的处理程序。8051 首先置位相应的中断优先级生效触发器，然后由硬件把当前的 PC 值压入堆栈，以保护断点，再将相应的中断服务的入口地址送入 PC，于是 CPU 接着从中断服务程序的入口地址开始执行。

(4) 中断处理：就是执行中断服务程序，完成中断源提出的处理要求。

(5) 中断返回：由中断服务程序返回到被中断的程序。

6. 8051 单片机有 5 个中断源，可以设置两个优先级。试分析以下几种中断优先级的安排是否可能？若可能，如何设定？若不可能，说明理由。优先级顺序为由高到低。

(1) 定时器 T0，定时器 T1，外部中断 $\overline{\text{INT0}}$ ，外部中断 $\overline{\text{INT1}}$ ，串行口中断。

(2) 串行口中断，外部中断 $\overline{\text{INT0}}$ ，定时器 T0，外部中断 $\overline{\text{INT1}}$ ，定时器 T1。

(3) 外部中断 $\overline{\text{INT0}}$ ，定时器 T1，外部中断 $\overline{\text{INT1}}$ ，定时器 T0，串行口中断。

(4) 外部中断 $\overline{\text{INT0}}$ ，外部中断 $\overline{\text{INT1}}$ ，串行口中断，定时器 T1，定时器 T0。

(5) 串行口中断，定时器 T0，外部中断 $\overline{\text{INT0}}$ ，外部中断 $\overline{\text{INT1}}$ ，定时器 T1。

(6) 外部中断 $\overline{\text{INT0}}$ ，外部中断 $\overline{\text{INT1}}$ ，定时器 T0，串行口中断，定时器 T1。

(7) 外部中断 $\overline{\text{INT0}}$ ，定时器 T1，定时器 T0，外部中断 $\overline{\text{INT1}}$ ，串行口中断。

答：8051 单片机内部查询的优先级顺序为：外部中断 $\overline{\text{INT0}}$ ，定时器 T0，外部中断 $\overline{\text{INT1}}$ ，定时器 T1，串行口中断

(1) 可以。将 T0、T1 设定为高优先级。

用指令：MOV IP, # 0AH

(2) 可以。将串行口中断设定为高优先级。

用指令：MOV IP, # 10H

(3) 不可以。由于定时器 T1 和外部中断 $\overline{\text{INT1}}$ 的内部查询优先级低于定时器 T0，需要将它们设定为高优先级，但因为只能设置一级高优先级，而定时器 T1 在同级中的优先级低于外部中断 $\overline{\text{INT1}}$ ，所以无法实现。

(4) 不可以。若将外部中断 $\overline{\text{INT0}}$ ，外部中断 $\overline{\text{INT1}}$ ，串行口中断设定为高优先级，则在低优先级中定时器 T1 比定时器 T0 要低，无法达到要求。

(5) 不可以。必须将串行口中断设定为高优先级，但在低优先级中，定时器 T0 比外部中断 $\overline{\text{INT0}}$ 要低，所以无法实现。

(6) 不可以。必须将外部中断 $\overline{\text{INT0}}$ 、外部中断 $\overline{\text{INT1}}$ 、定时器 T0、串行口中断设定为高优先级，但外部中断 $\overline{\text{INT1}}$ 比定时器 T0 优先级低，所以无法实现。

(7) 可以。将外部中断 $\overline{\text{INT0}}$ 、定时器 T1 设定为高优先级。

用指令：MOV IP, # 09H

7. 请问在什么情况下可响应新的中断请求？

答：符合以下 6 个条件可以响应新的中断请求：

(1) 有中断源发出中断请求。

(2) 中断总允许位 EA = 1，即 CPU 开放中断。

(3) 申请中断的中断源的中断允许位为 1，即中断没有被屏蔽。

(4) 无同级或更高级中断正在被服务。

(5) 当前的指令周期已结束。

(6) 若现行指令为 RETI 或访问 IE 或 IP 指令时，该指令以及紧接着的下一条指令已执行完。

8. 若要求中断源 T0、T1、 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 允许中断，禁止串行口中断，将 T0、T1 设定为高优先级， $\overline{\text{INT0}}$ 、 $\overline{\text{INT1}}$ 设定为低优先级，请确定 IE、IP 控制字，并编写初始化程序。

答：根据要求，控制字 IE、IP 分别为：

IE	1			0	1	1	1	1
----	---	--	--	---	---	---	---	---

即 IE = 8FH。

IP	0			0	1	0	1	0
----	---	--	--	---	---	---	---	---

即 IP = 0AH。

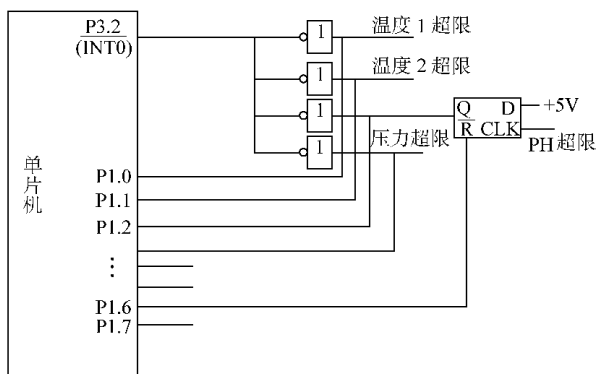
初始化程序如下：

```
ORG    0000H
AJMP   MAIN
ORG    0003H      ; 外中断INT0中断入口地址
AJMP   W1
ORG    000BH      ; 外中断 T0 中断入口地址
AJMP   W2
ORG    0013H      ; 外中断INT1中断入口地址
AJMP   W3
ORG    001BH      ; 外中断 T1 中断入口地址
AJMP   W4
ORG    0050H      ; 主程序
MAIN:  MOV    IE, # 8FH      ; 开中断
      MOV    IP, # 0AH      ; 设置中断优先级
      .....
HERE:  AJMP   HERE          ; 等待
W1:    .....
      RETI
W2:    .....
      RETI
W3:    .....
      RETI
W4:    .....
      RETI
```

9. 某工业监控系统，具有温度、压力、PH 值等多路监控功能。在 PH 值 < 7 时，将向 CPU 申请中断，CPU 响应中断后令 P3.0 引脚输出高电平，经驱动，使加碱管道电磁阀接通 1s，以调整 PH 值。详见电路图 5-2。请设计控制软件。

答：PH 值 < 7 时，由外部中断 0 申请中断，但是由外部中断 0 提出中断请求的中断源有很多，究竟是哪一项监控参数超限报警，需要检查 P1 口信息后判定，并转向相应的中断服务程序入口地址，作出相应处理。设定 PH 值 < 7 时 P1.2 将置 1。

图 5-2 中右侧是 PH 超限中断请求撤除电路。其实只是添用了一个 D 触发器，超限信号自 CLK 端送入，复位端用于撤除此次的中断请求，对于用电平触发的外部中断，一般应采取撤除中断请求的措施，防止一次中断请求却重复多次响应。

图 5-2 多个中断源共用 $\overline{\text{INT0}}$ 申请中断

编写程序如下：

```

ORG    0000H
AJMP   MAIN
ORG    0003H           ; 外部中断 0 中断入口地址
AJMP   INT0
ORG    0030H
MAIN:  MOV    IE, # 81H       ; 允许外部中断 0 中断
      CLR    IT0             ; 外部中断 0 设定为电平触发方式
      .....
HERE:  SJMP   HERE           ; 等待中断
ORG    0050H           ; 外部中断 0 服务程序
INT0:  JB     P1.0, TE1       ; 是否温度 1 超限
      JB     P1.1, TE2       ; 是否温度 2 超限
      JB     P1.2, PH        ; 是否 PH 值超限
      JB     P1.3, PL        ; 是否压力超限
      .....
ORG    0080H           ; PH 值超标处理程序
PH:    PUSH   PSW            ; 保存现场
      PUSH   A
      SETB   P3.0            ; 接通加碱管道电磁阀
      ACALL  DELAY1s         ; 调延时 1s 子程序
      CLR    P3.0            ; 1s 时间到关闭加碱管道电磁阀
      ANL    P1, # BFH       ; 产生一个 P1.6 的负脉冲, 用来撤除
      ORL    P1, # 40H       ; PH < 7 的中断请求
      POP    A               ; 恢复现场
      POP    PSW
      RETI                    ; 中断返回

```


第六章 定时器/计数器

第一节 学习要求与内容概要

一、学习要求

- (1) 定时器/计数器的工作原理及其控制字的含义；
- (2) 4 种工作方式的功能及其初值的计算；
- (3) 定时器/计数器的初始化编程方法。

二、定时器/计数器的工作原理与组成学习内容概要

1. 定时器/计数器的工作原理

图 6-1 为定时器/计数器原理图。它的核心是一个加 1 计数器，每输入一个脉冲，计数值加 1。加 1 计数器的输入脉冲有两个来源，一个是外部脉冲源 TX 引脚，另一个是系统的时钟振荡器。当计数到计数器各位全为 1（即计数器满）时，再输入一个脉冲就置位计数器的溢出中断标志。如果定时器/计数器工作于定时状态，则表示定时时间到；若工作于计数状态，则表示计数回零。

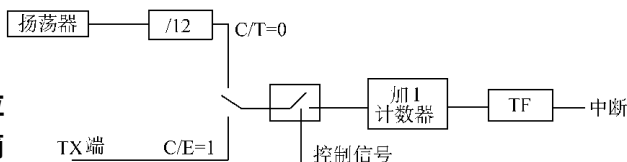


图 6-1 定时器/计数器的工作原理

2. 定时器/计数器的组成

MCS-51 单片机的内部有两个 16 位加 1 可编程定时器/计数器 T0 和 T1（简称定时器 T0、T1）。定时器 T0 由 8 位特殊功能寄存器 TH0 和 TL0 组成，定时器 T1 由 8 位特殊功能寄存器 TH1 和 TL1 组成。

(1) 定时器的控制寄存器 TCON 定时器的控制寄存器 TCON 具有中断控制和定时控制两种功能。TCON 的字节地址是 88H，其中，低 4 位字段与外部中断有关，高 4 位字段控制定时功能。

(2) 定时器的方式寄存器 TMOD 用于控制定时器/计数器的两种功能及四种工作方式。TMOD 的字节地址为 89H，它不可被位寻址，其内容只能用字节传送指令来设置。其中，高 4 位字段用于定时器 T1 的工作方式控制，低 4 位字段用于定时器 T0 的工作方式控制。

在这个控制字中尤其要注意 GATE 门控制位的运用：

当 GATE = 1 时，T0（或 T1）的工作受 $\overline{\text{INT0}}$ （ $\overline{\text{INT1}}$ ）引脚电平控制，通常称为外部控制。此时，只有 TR0（或 TR1）为 1 时，且 $\overline{\text{INT0}}$ （或 $\overline{\text{INT1}}$ ）输入为高电平时，定时器 T0（或 T1）才允许开始工作。多用于测量 $\overline{\text{INT0}}$ （ $\overline{\text{INT1}}$ ）引脚上正脉冲的宽度。

当 GATE = 0 时，只要使 TR0（或 TR1）置 1，就可使定时器/计数器开始工作，而不必受 $\overline{\text{INT0}}$ （或 $\overline{\text{INT1}}$ ）的影响，因此通常称为内部控制。通过定时和计数相结合，可以测量外部信号的脉冲宽度。

定时器的工作方式由 M1、M0 两位的状态确定，其对应关系如表 6-1 所示。

表 6-1 工作方式、功能及初值计算

M1	M0	工作方式	功能说明	定时初值计算
0	0	方式 0	13 位定时器/计数器	$(2^{13} - X) \div f = N$ TH = X 高 8 位, TL = X 低 5 位
0	1	方式 1	16 位定时器/计数器	$(2^{16} - X) \div f = N$ TH = X 高 8 位, TL = X 低 8 位
1	0	方式 2	自动重装初值常数的 8 位定时器/计数器	$(2^8 - X) \div f = N$ TH = TL = 8 位 X
1	1	方式 3	仅适用于 T0, 分为两个 8 位定时器/计数器	$(2^8 - X_1) \times T = N_1$ TH0 = 8 位 X_1 $(2^8 - X_2) \times T = N_2$ TL0 = 8 位 X_2

注：f—计数器输入脉冲信号频率；N—定时时间；X—定时初值。

三、定时器/计数器的工作方式学习内容概要

(1) 方式 0 定时器是 13 位计数器结构，它由 TH 的高 8 位和 TL 的低 5 位构成，TL 高 3 位不用。TL 的低 5 位计满溢出时，向 TH 进位；13 位计数器计满溢出时，向定时器溢出中断标志位 TF0 (TF1) 进位，使 TF0 (TF1) 置位。

(2) 方式 1 在方式 1 中，16 位计数器由两个 8 位寄存器 TH 和 TL 组成。其中 TL 计满溢出时，向 TH 进位；16 位计数器计满溢出时，向定时溢出中断标志位进位，使 TF0 (TF1) 置 1。

(3) 方式 2 具有自动重新装入计数初值的功能。定时器 T0 (T1) 是一个可自动重新装入计数初值的 8 位定时器，TL0 (TL1) 作为 8 位计数器，TH0 (TH1) 为常数寄存器。这种工作方式省去了用户程序中重新装载初值的部分，能获得高精度的定时时间。方式 2 常用于串行口波特率发生器。

(4) 方式 3 它只适用于定时器/计数器 T0，且是为了增加一个附加的 8 位定时器而提供的。如果定时器 T1 置为方式 3，其效果与 TR1 = 0 相同，即关闭定时器 T1。

如果定时器 T0 置为方式 3，则 TH0 和 TL0 是两个独立的 8 位计数器。其中，TL0 既可以作为定时器，也可以作为计数器，它使用 T0 的状态控制位 $\overline{C/T}$ 、GATE、TR0、TF0 和外部信号输入端 $\overline{INT0}$ 。而 TH0 只能作为一个 8 位定时器使用，它的运行控制位和溢出中断标志位借用定时器 T1 的 TR1 和 TF1。因此，TH0 计满溢出时，将 TF1 置位，并且 TH0 定时的启动和关闭受 TR1 状态的控制。

由于此时 TH0 借用了 T1 的溢出标志位 TF1，因此，T1 虽可工作于方式 0、1、2，但只能用做串行口的波特率发生器，或不需要中断控制的场合。

四、定时器/计数器的应用举例

定时器/计数器的应用编程主要有几点：

- (1) 选择工作定时器/计数器及其工作方式。
- (2) 确定控制字。
- (3) 计算初值。
- (4) 编制初始化程序。

定时器/计数器初始化程序设计的基本步骤如下：

(1) 确定工作方式，对寄存器 TMOD 赋值

常用命令语句为：MOV TMOD, # * * H；

(2) 预置定时或计数的初值 X。

常用命令语句为：MOV TL* , # * * H；

MOV TH* , # * * H；

(3) 根据需要开放定时器/计数器的中断

常用命令语句为：MOV IE, # * * H；

(4) 启动定时器/计数器工作

常用命令语句为：SETB TR* ；

(5) 停止定时器/计数器工作

常用命令语句为：CLR TR* ；

第二节 原教材习题及解答

7-1 8031 的 T0、T1 用作定时器时，其定时时间与哪些因素有关？

答：与系统振荡频率、工作方式、定时初值有关。

7-2 MCS- 51 单片机的晶振频率为 12MHz，使用单片机的内部定时方式产生频率为 10kHz 的方波信号。

答：(1) 首先选择工作方式。由于是方波，所以高、低电平各占一个周期的一半时间，方波频率为 10kHz。周期是 $1/(10 \times 10^3) \text{ s} = 100\mu\text{s}$ ，则高、低电平的定时时间都是 $100\mu\text{s}/2 = 50\mu\text{s}$ ，可以选择定时器 T0 以工作方式 2 来产生连续方波。

(2) 计算初值。机器周期 $T_m = 12 \times 1/(12 \times 10^6) \text{ s} = 1\mu\text{s}$ ，则定时初值 $X = \text{最大计数值} - (\text{定时时间} t/\text{机器周期} T_m) = 256 - 50/1 = 206 = 0CEH$ 。

(3) 确定控制字。

TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
	0	0	0	0	0	0	1	0

即 TMOD = 02H。

(4) 选择 P1.0 输出方波信号，编写如下程序。

```
ORG      0000H
RESET:  AJMP    MAIN          ;复位后转主程序
ORG      000BH
AJMP    IT0P                  ;转 T0 中断服务程序
ORG      0100H
MAIN:   MOV     SP, # 60H
        MOV     TMOD, # 02H    ;设置 T0 的方式 2
        MOV     TH0, # 0CEH    ;设置计数初值
        MOV     TL0, # 0CEH
```

SETB ET0
SETB EA
SETB TR0
LOOP： SJMP LOOP
ORG 0200H
IT0P： CPL P1.0
RETI

允许 T0 中断
CPU 开放中断
启动定时器
等待中断
输出方波信号

7-3 设 MCS—51 单片机的晶振频率为 6MHz。若要求定时值分别为 0.1ms、1ms、10ms，定时器 T0 工作在方式 0、方式 1 和方式 2 时，其定时器初值各应是多少？

答：定时初值如表 6-2。

表 6-2 定时器初值

时间/ms 方式	0.1		1		10	
	TH0	TL0	TH0	TL0	TH0	TL0
方式 0	0FEH	0EH	0F0H	0CH	63H	18H
方式 1	0FFH	0CEH	0FEH	0CH	ECH	78H
方式 2	0CEH	0CEH	无法一次实现定时， 可用 0.1ms 循环 10 次		无法一次实现定时， 可用 0.1ms 循环 100 次	

7-4 设 MCS—51 单片机的晶振频率为 6MHz，使用定时器 T1 的定时方式 1，在 P1.0 输出周期为 20us、占空比为 60% 的矩形脉冲，以查询方式编写程序。

答：(1) 题目中指定了定时器 T1 的定时方式 1。

(2) 计算初值。机器周期 $T_m = 12 \times 1 / (6 \times 10^6) \text{ s} = 2\mu\text{s}$ ，周期为 20us、占空比为 60% 的矩形脉冲的高电平持续时间为 $20 \times 60\% = 12\mu\text{s}$ ，低电平持续时间为 $20 \times (1 - 60\%) = 8\mu\text{s}$ ，则定时初值 $X_{\text{高}} = \text{最大计数值} - (\text{定时时间} t / \text{机器周期 } T_m) = 2^{13} - (12/2) = 8186$ ，所以 $\text{TH1}_{\text{高}} = 11111000 = 0\text{F8H}$ ， $\text{TL1}_{\text{高}} = 11111010 = 0\text{FAH}$ ，同理可得 $\text{TH1}_{\text{低}} = 11111000 = 0\text{F8H}$ ， $\text{TL1}_{\text{低}} = 1111100 = 0\text{FCH}$ 。

(3) 确定控制字。

TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
	0	0	0	1	0	0	0	0

即 TMOD = 10H。

(4) 编写程序。

ORG 0100H
MAIN： MOV SP， # 60H
MOV TMOD， # 10H
SETB TR1
LOOPH： MOV TH1， # 0F8H
MOV TL1， # 0FAH
SETB P1.0
LOOP： JBC TF1， LOOP1

设置 T1 工作于方式 1 定时
启动定时器
设置高电平时初值
输出高电平
查询定时溢出位 TF1

```

    AJMP    LOOP
LOOP1: JB    P1.0, LOOPL      判断高、低电平
    AJMP    LOOPH
LOOPL: MOV    TH1, # 0F8H    设置低电平定时初值
    MOV    TL1, # 0FCH
    CLR    P1.0             输出低电平
    AJMP    LOOP

```

7-5 设 MCS—51 单片机的晶振频率为 12MHz, T0 用于 20ms 定时, T1 用于 100 次计数, T0 和 T1 均要求重复工作。(1) 外部计数脉冲应从何引脚输入? (2) 试根据上述要求编写程序。

答: (1) 外部计数脉冲应从 T1 (P3.5) 引脚输入。

(2) 程序设计。

1) 题目中指定了使用 T0 做为定时器, 由于定时时间较长, 可以选择它的工作方式 1, 使用 T1 做为计数器, 选择 T1 的工作方式 2。

2) 计算初值。机器周期 $T_m = 12 \times 1 / (12 \times 10^6) \text{ s} = 1 \mu\text{s}$, 则 20ms 定时的定时初值 $X = \text{最大计数值} - (\text{定时时间} t / \text{机器周期 } T_m) = 2^{16} - (20 \times 10^3 / 1) = 45536$, 所以 $\text{TH0} = 0\text{B1H}$, $\text{TL0} = 0\text{E0H}$; 100 次计数的计数初值 $X = \text{最大计数值} - \text{计数个数} = 256 - 100 = 156$, 所以 $\text{TH1} = \text{TL1} = 9\text{CH}$ 。

3) 确定控制字。

TMOD	GATE	$\overline{\text{CT}}$	M1	M0	GATE	$\overline{\text{CT}}$	M1	M0
	0	1	1	0	0	0	0	1

即 $\text{TMOD} = 61\text{H}$ 。

4) 编写程序。

```

    ORG    0000H
RESET: AJMP    MAIN          ;复位后转主程序
    ORG    000BH
    AJMP    IT0P            ;转 T0 中断服务程序
    ORG    001BH
    AJMP    IT1P            ;转 T1 中断服务程序
    ORG    0100H
MAIN:  MOV    SP, # 60H
    MOV    TMOD, # 61H      设置定时器 T0 方式 1, 计数器 T1 方式 2
    MOV    TH0, # 0B1H      设置定时初值
    MOV    TL0, # 0E0H
    MOV    TH0, # 09CH      设置计数初值
    MOV    TL0, # 09CH
    SETB    ET0             允许 T0 中断
    SETB    ET1             允许 T1 中断

```

SETB	EA	∫CPU 开放中断
SETB	TR0	启动定时器
SETB	TR1	启动计数器
LOOP:	SJMP LOOP	等待中断
.....		
IT0P:	T0 定时中断服务程序	
.....		
IT1P:	T1 计数中断服务程序	

第三节 增选习题及解答

一、填空题

1. MCS-51 单片机有____个 16 位可编程定时器/计数器, 有____种工作方式。
2. MCS-51 单片机的最大计数值为____, 此时工作于方式____。
3. 若定时器/计数器工作于方式 0 时, 为____位的定时器/计数器, TL 的低____位计满溢出时, 向高____位的 TH 进位。
4. 当定时器/计数器 T0 工作在方式 3 时, 要占用定时器/计数器 T1 的____和____两个控制位。
5. 当把定时器/计数器 T0 定义为一个可自动重新装入初值的 8 位定时器/计数器时, ____作为 8 位计数器, ____为常数寄存器。
6. 使用定时器 T1 设置串行通信的波特率时, 应把定时器 T1 设定为工作方式 2, 即____。
7. 若 8031 的晶振频率为 12MHz, 则其两个定时器/计数器对重复频率高于____MHz 的外部事件是不能正确计数的。
8. 在定时器/计数器 T0 工作方式 3 下, TH0 溢出时, ____标志将被硬件置 1 去请求中断。
9. T0 和 T1 两引脚也可作为外部中断输入引脚, 这时 TMOD 寄存器中的 C/T 位应当为____。若 M0、M1 两位置成 10B, 则计数初值应当是 TH = TL = ____。
10. 欲使 T1 停止工作, 应执行一条指令____, 在工作方式 3 下, 欲使 TH0 开始工作, 应执行一条指令____。

填空题参考答案:

1. 2, 4 2. 65536, 1 3. 13, 5, 8 4. TR1, TF1 5. TL0, TH0 6. 自动重装载 7. 0.5
8. TF1 9. 1, 0FFH (或 255) 10. CLR TR1, SETB TR1

二、选择题

1. 下面哪一种工作方式仅适用于定时器 T0 ()。
A. 方式 0 B. 方式 1 C. 方式 2 D. 方式 3
2. 若系统晶振频率是 12MHz, 利用定时器/计数器 1 定时 1ms, 在方式 0 下的定时初值为 ()。
A. TH1 = 0E0H, TL1 = 18H B. TH1 = 18H, TL1 = E0H
C. TH1 = 1CH, TL1 = 18H D. TH1 = 18H, TL1 = 1CH

3. 若 MCS-51 单片机的晶振频率为 24MHz, 则其内部的定时器/计数器利用计数器对外部输入脉冲的最高计数频率是 ()。

A. 1MHz B. 6MHz C. 12MHz D. 24MHz

4. 设定定时器/计数器 T1 的工作于外部控制方式, 则只有在 () 情况下, T1 才开始工作。

A. TR1 = 0, INT1 = 0 B. TR1 = 1, INT1 = 0

C. TR1 = 0, INT1 = 1 D. TR1 = 1, INT1 = 1

5. 设定定时器/计数器 T0 工作于方式 3, 则 TH0 作为一个独立的 8 位定时器, 它的运行由控制位 ____ 决定。

A. GATE B. INT0 C. TR0 D. TR1

选择题参考答案:

1. D 2. A 3. A 4. D 5. D

三、简答题

1. 8051 定时器做定时和计数时, 其计数脉冲分别由谁提供?

答: 做定时器时, 计数脉冲由 8051 片内振荡器输出经 12 分频后的脉冲提供, 做计数器时, 计数脉冲由外部信号通过引脚 P3.4 和 P3.5 提供。

2. 8051 定时器的门控信号 GATE 设置为 1 时定时器如何启动?

答: 只有外部中断输入引脚 $\overline{\text{INT0}}$ (或 $\overline{\text{INT1}}$), 引脚为高电平且由软件使 TR0 (或 TR1) 置 1 时, 才能启动定时器工作。

3. 当定时器 T0 用作方式 3 时, 由于 TR1 位已被 T0 占用, 如何控制定时器 T1 的运行?

答: 定时器 T0 用作方式 3 时, 由于 TR1 位已被 T0 占用, 此时通过选定工作方式及写入计数初值后, T1 就开始工作, 主要用于产生串行通信的波特率。

4. 请设计读定时器/计数器的子程序。

答: 在定时器/计数器工作过程中, 经常需要读出定时器/计数器的数值。假设读取定时器/计数器 T0 的数值, 并要求将读得的 TH0 和 TL0 的数值分别送入 R0、R1 中保存, 则设计子程序如下:

READ: MOV	A, TH0	读 TH0 内容
MOV	R1, TL0	读 TL0 内容, 送入 R1
CJNE	A, TH0, READ	再读 TH0 内容, 与上次读值相比较, 如不相等, 转去重读
MOV	R0, TH0	如果相等, 送入 R0
RET		

说明: 该程序对 TH0 读取了两次, 也就是采用了先读 TH0, 后读 TL0, 再读 TH0 的方法, 这样做的目的是为了防止错读, 因为不能在同一时刻同时读取 TH0 和 TL0, 而且在读取的过程中定时器/计数器的工作没有停止, 计数值是一直在变化的, 如果在读取的过程中刚好有 TL0 溢出向 TH0 进位, 则容易错读, 重读 TH0 就可防止错误。

5. 已知 8051 单片机的系统振荡频率为 6MHz, 利用定时器使 P1.0 每隔 500us 输出一个 50us 的正脉冲。

答: (1) 选择定时器及其工作方式。因为要重复输出脉冲, 因此选择定时器 T0 工作于

方式 2。

(2) 计算初值。机器周期 $T_m = 12 \times 1 / (6 \times 10^6) \text{ s} = 2\mu\text{s}$ ，则 $500\mu\text{s}$ 定时的定时初值 $X = \text{最大计数值} - (\text{定时时间 } t / \text{机器周期 } T_m) = 2^8 - (500/2) = 6$ ，所以 $\text{TH0} = \text{TL0} = 06\text{H}$ ，而 $50\mu\text{s}$ 定时的定时初值 $X = \text{最大计数值} - (\text{定时时间 } t / \text{机器周期 } T_m) = 2^8 - (50/2) = 231$ ，所以 $\text{TH0} = \text{TL0} = 0\text{E7H}$ 。

(3) 确定控制字。

TMOD	GATE	$\overline{\text{C/T}}$	M1	M0	GATE	$\overline{\text{C/T}}$	M1	M0
	0	0	0	0	0	0	1	0

即 $\text{TMOD} = 02\text{H}$ 。

(4) 编写程序段。

	ORG	0100H	
	MOV	TMOD, # 02H	设置 T0 工作方式 2
NEXT:	MOV	TH0, # 06H	设置 500us 定时初值
	MOV	TL0, # 06H	
	CLR	P1.0	P1.0 清零,低电平
	SETB	TR0	启动定时器 T0 开始工作
TM1:	JBC	TF0, LOOP	查询定时时间到否?
	SJMP	TM1	等待
LOOP:	SETB	P1.0	P1.0 输出高电平
	MOV	TH0, # 0E7H	设置 50us 定时初值
	MOV	TL0, # 0E7H	
TM2:	JBC	TF0, NEXT	查询定时时间到否?
	SJMP	TM2	等待

6. 8051 晶振频率为 12MHz ，试用中断方法，设计秒、分定时器，分别由 P1.0 ， P1.1 发出脉冲信号。

答：(1) 选择定时器及其工作方式。因为晶振频率为 12MHz 时，方式 1 最长，定时时间是 $65536\mu\text{s}$ ，所以以定时时间 50ms 作单位，由程序循环 20 次实现 1s 定时，再以此为单位，由程序循环 60 次实现 1min 定时。

(2) 计算初值。机器周期 $T_m = 12 \times 1 / (12 \times 10^6) \text{ s} = 1\mu\text{s}$ ，则 50ms 定时的定时初值 $X = \text{最大计数值} - (\text{定时时间 } t / \text{机器周期 } T_m) = 2^{16} - (50000/1) = 15536 = 3\text{CB0H}$ ，所以 $\text{TH0} = 3\text{CH}$ ， $\text{TL0} = 0\text{B0H}$ 。

(3) 确定控制字。

TMOD	GATE	$\overline{\text{C/T}}$	M1	M0	GATE	$\overline{\text{C/T}}$	M1	M0
	0	0	0	0	0	0	0	1

即 $\text{TMOD} = 01\text{H}$ 。

IE	EA			ES	ETI	EXI	ET0	EX0
	1	0	0	0	0	0	1	0

即 $IE = 82H$ 。

(4) 编写程序。

	ORG	0000H	
	LJMP	MAIN	跳转到主程序
	ORG	000BH	定时器 T0 中断程序入口
	AJMP	AA	跳转到中断服务程序
	ORG	0100H	主程序入口地址
MAIN :	MOV	TMOD, # 01H	设置定时器 T0 工作方式 1
	MOV	R0, # 14H	定时器 50ms 的 20 次循环 = 1s
	MOV	R1, # 3CH	定时 1s 的 60 次循环 = 1min
	CLR	P1.0	
	CLR	P1.1	
	MOV	TH0, # 3CH	定时器 T0 赋初值
	MOV	TL0, # 0B0H	
	MOV	IE, # 82H	CPU 开中断, T0 开中断
	SETB	TR0	启动 T0
HERE :	SJMP	HERE	循环等待定时时间到
	ORG	0200H	中断服务程序入口地址
AA :	MOV	TL0, # 0B0H	重赋初值
	MOV	TH0, # 3CH	
	DJNZ	R0, LOOP	不到 1s 继续循环
	MOV	R0, # 14H	重赋 50ms 的循环次数
	SETB	P1.0	发秒脉冲
	NOP		
	CLR	P1.0	
	DJNZ	R1, LOOP	不到 1min 继续循环
	MOV	R1, # 3CH	重赋 1s 的循环次数
	SETB	P1.1	发分脉冲
	NOP		
	CLR	P1.1	
LOOP :	RET		中断返回

7. 若 8051 晶振频率为 6MHz, 用定时器/计数器 T1 对外部事件进行计数, 每计数 5000 个脉冲后, 定时器/计数器 T1 转为定时方式, 定时 5ms 后, 又转为计数方式, 如此不断循环, 请编写程序。

答: (1) 设定 T1 分别工作于定时和计数工作方式 1。

(2) 计算初值。机器周期 $T_m = 12 \times 1 / (6 \times 10^6) s = 2\mu s$, 则 50ms 定时的定时初值 $X = \text{最大计数值} - (\text{定时时间} t / \text{机器周期 } T_m) = 2^{16} - (5000/2) = 63036 = 0F63CH$, 所以定时初值 $TH0 = 0F6H$, $TL0 = 3CH$, 而计数 5000 次的计数初值 $X = \text{最大计数值} - \text{计数值} = 2^{16} - 5000 = 15536 = 3CB0H$, 所以计数初值 $TH0 = 3CH$, $TL0 = 0B0H$ 。

(3) 确定控制字。

TMOD	GATE	$C/\bar{T}$	M1	M0	GATE	$C/\bar{T}$	M1	M0
	0	0	0	1	0	0	0	0

即定时方式控制字 TMOD = 10H。

TMOD	GATE	$C/\bar{T}$	M1	M0	GATE	$C/\bar{T}$	M1	M0
	0	1	0	1	0	0	0	0

即计数方式控制字 TMOD = 50H。

(4) 编写程序段。

```

ORG      0100H
L1:      MOV     TMOD, # 50H      设置 T1 为计数方式且工作于方式 1
          MOV     TH1, # 3CH      设置计数初值
          MOV     TL1, # 0B0H
          SETB    TR1             启动 T1 计数器
LOOP1:   JBC     TF1, LOOP2       查询计数有无溢出？有溢出转去定时
          SJMP    LOOP1           无溢出，继续查询
LOOP2:   CLR     TR1             关闭 T1
          MOV     TMOD, # 10H     设置 T1 为定时方式且工作于方式 1
          MOV     TH0, # 0F6H     设置初值
          MOV     TL0, # 3CH
          SETB    TR1             启动 T1 定时
LOOP3:   JBC     TF1, L1          查询定时时间到，转去计数
          SJMP    LOOP3           定时时间未到，继续查询

```

8. 设 MCS - 51 单片机系统的晶振频率为 12MHz，编程测试外部脉冲输入引脚 $\overline{\text{INT0}}$ (P3.2) 上输入的正脉冲宽度 (设正脉冲宽度小于 65ms)，并将测量数值以 μs 为单位存放于 R1, R2 中。

答：当 GATE = 1 时，T0 的工作不仅由 TR0 控制，也要受 $\overline{\text{INT0}}$ 引脚电平控制，只有当 TR0 为 1，且 $\overline{\text{INT0}}$ 输入为高电平时，定时器 T0 才启动。利用这个功能，可以测试外部脉冲输入引脚 $\overline{\text{INT0}}$ 上输入的正脉冲宽度。

(1) 选择定时器及其工作方式：选择定时器 T0 工作于方式 1 定时。

(2) 计算初值。机器周期 $T_m = 12 \times 1 / (12 \times 10^6) \text{ s} = 1\mu\text{s}$ ，将 TL0 和 TH0 的初值都定义为 00H，则它们的读数就是所要测试的以 μs 为单位的定时时间。

(3) 确定控制字。

TMOD	GATE	$C/\bar{T}$	M1	M0	GATE	$C/\bar{T}$	M1	M0
	0	0	0	0	1	0	0	1

即 TMOD = 09H。

(4) 编写程序段。

```

ORG      0100H
L1:      MOV     TMOD, # 09H      设置 T0 工作于方式 1 定时且受外部控制

```

MOV TH0, # 00H
MOV TL0, # 00H
HR1 : JB P3.2, HR1
SETB TR0
HR2 : JNB P3.2, HR2
HR3 : JB P3.2, HR3
CLR TR0
MOV R1, TL0
MOV R2, TH0

设置计数初值

等待INT0引脚为低电平
;TR0 = 1, 但 T0 启动由INT0引脚变高决定
等待INT0引脚高电平到来, T0 开始工作
计数进行中等待INT0引脚再次变低
当INT0引脚为低电平时, 停止定时器 T0
输出定时时间的低 8 位
输出定时时间的高 8 位

9. 有一生产流水线，流水线上通过的工件经光电转换电路产生计数脉冲，脉冲整形后送入 T0 端，要求每生产 100 个工件，向 P1.0 发出一包装命令正脉冲，包装成一箱，请编写控制程序。

答：(1) 选择定时器及其工作方式。选择定时器 T0 工作于方式 2 计数。
(2) 计算初值。计数初值 $X = \text{最大计数值} - \text{计数值} = 2^8 - 100 = 9CH$ ，所以 TH0 = TL0 = 9CH。

(3) 确定控制字。

TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
	0	0	0	0	0	1	1	0

即 TMOD = 06H。

(4) 采用中断方式，编写程序。

ORG 0000H
AJMP MAIN
ORG 000BH
AJMP COUNT
ORG 0100H
MAIN : MOV TMOD, # 06H
MOV TH1, # 9CH
MOV TL1, # 9CH
CLR P1.0
SETB ET0
SETB EA
SETB TR0
HERE : SJMP HERE
ORG 0200H
COUNT : SETB P1.0
NOP
CLR P1.0
RETI

;T0 中断入口
转向中断处理程序
主程序
设置 T0 工作于方式 2 计数
设置计数初值

清除包装信号

启动 T1 计数器
计数等待

发出包装命令脉冲

第七章 MCS—51 单片机串行口及应用

第一节 学习要求与内容概要

一、学习要求

(1) 熟悉内容：数据通信中的一些概念，如并行通信/串行通信、异步串行通信/同步串行通信、单工/半双工/全双工通信方式及波特率等概念；MCS—51 单片机串行口的基本结构；双机通信/多机通信的基本原理、硬件系统及软件设计。

(2) 重点掌握内容：MCS—51 单片机串行口工作过程；与串行口有关的特殊功能寄存器的各位的含义及功能；串行口四种工作方式下的数据格式、波特率计算及实际应用。

二、通信技术基础学习内容概要

本节主要是了解数据通信的一些基本概念。

(1) 数据通信的两种主要方式是并行通信与串行通信。并行通信是在传送数据过程中每个字节的各位同时进行传送的通信方式，而串行通信是指每个字节的各位（通常为 8 位）按顺序一位位地分别进行传送的通信方式。并行通信传输速度快，需要传输线多，适宜于短距离通信；串行通信传输速度慢，需要的传输线少。

(2) 串行通信的两种数据传送格式是异步串行通信和同步串行通信。异步串行通信中，数据是一帧一帧传送的，每帧数据包括一位起始位、一个字节的的数据、一位校验位和一位停止位。每帧数据之间可以插入若干个高电平的空闲位。异步串行通信要求发送数据和接收数据双方约定相同的数据格式和速率；用启、停位来协同发送与接受过程；接收和发送端采用独立的移位脉冲；每帧数据只有一个字节数据；传输速度较慢。同步串行通信是一种连续的数据传送方式。每次传送一帧数据，每帧数据由同步字符和若干个数据及校验字符组成。同步串行通信发送和接受双方由同一个同步脉冲控制；需传送同步脉冲信号；每帧数据有多个字节数据；同步串行通信速度较快。

(3) 串行数据的传送方式。按数据的传送方向可分为单工（数据向一个方向传送）、半双工（允许数据分时进行双向传送）、全双工（允许双方同时进行数据的接收和发送）通信方式。

(4) 波特率定义为每秒传送二进制数的位数，单位为 bit/s。

三、MCS—51 单片机串行口学习内容概要

1. MCS—51 单片机串行口的结构和功能

MCS—51 单片机串行口是一个全双工通信接口。

(1) 外部特性

串行口的数据发送和接收是通过两个引脚 RXD（P3.0，串行通信数据输入端）和 TXD（P3.1，串行通信数据输出端）与外界进行串行通信。

② 内部主要结构

1) 串行口缓冲器 SBUF 它的地址为 99H。在物理上, SBUF 对应串行口接收缓冲器和串行口发送缓冲器两个独立的单元, 可同时发送和接收数据。

2) 发送控制器 它和发送缓冲器 SBUF 控制数据的发送。

3) 接收控制器和输入移位寄存器 由接收缓冲器 SBUF 控制数据的接收。

4) 串行口控制寄存器 SCON 用于选择串行口的工作方式和指示串行口的工作状态。

③ 串行口的工作过程

1) 发送过程 在 CPU 执行写 SBUF (发送) 指令 “MOV SBUF, A” 时, CPU 将数据装入 SBUF, TB8 装入发送移位寄存器的第 9 位, 启动发送过程, 由 TXD 向外发送一帧数据, 发送完毕, 将串行口发送中断标志位 TI 位置 1, 向 CPU 申请中断。

2) 接收过程 在串行口接收中断标志位 RI = 0, 且允许接收位 REN = 1 时, 当接收控制器检测到 RXD 端的负跳变时, 便启动接受过程。CPU 以波特率的速率接收一帧数据, 并移入接收移位寄存器; 然后装入 SBUF (接收), 并对 RI 置 1, 申请中断。执行指令 “MOV A, SBUF”, 读串行口接收缓冲器, 将串行接收的数据读入 A 中。

2. 与串行口有关的三个特殊功能寄存器

三个特殊功能寄存器分别是串行口缓冲器 SBUF, 串行口控制寄存器 SCON 和波特率控制寄存器 PCON。其中, SCON 用于选择串行口的工作方式和指示串行口的工作状态; PCON 主要用于波特率选择和断电控制。

3. 串行口的工作方式

(1) 工作方式 0 移位寄存器输入输出工作方式, 此时的波特率为固定值, 它是晶体振荡器的 $1/12$, 即 $f_{osc}/12$ 。

(2) 工作方式 1 10 位异步串行通信工作方式, 一帧数据包括一个起始位、8 位数据位 (低位在先)、一个停止位。工作方式 1 的波特率由定时器 T1 的溢出率决定, 即

$$\text{方式 1 波特率} = \frac{2^{SMOD}}{32} \times T_1 \text{ 的溢出率}$$

(3) 工作方式 2 11 位异步串行通信方式包括一位启动位、8 位数据位 (低位在先)、一位可编程的第 8 位数据 (TB8)、一个停止位。方式 2 的波特率与晶体振荡器的频率 f_{osc} 及 SMOD 位内容有关, 关系式为

$$\text{方式 2 波特率} = \frac{2^{SMOD}}{64} \times f_{osc}$$

(4) 工作方式 3 11 位可变波特率异步串行通信方式包括一位启动位、8 位数据位 (低位在先)、一位可编程的第 8 位数据 (TB8)、一个停止位。方式 3 波特率设置与工作方式 1 相同。

第二节 原教材习题及解答

8-1 计算机的两种通信方式是什么? 各有什么特点?

答: 计算机进行数据通信的两种主要方式是: 并行通信与串行通信, 两者的定义及特点参考教材《串行口》一章第一节通信技术基础。

8-3 波特率具体含义是什么？为什么说串行通信的双方波特率必须相同？

答：(1) 波特率定义为每秒传送二进制数的位数，单位为 bit/s。

(2) 如果串行通信的双方波特率不相同，发送和接收数据时，双方的速率不同会使数据接收出错，通信无法正常进行。

8-5 若设 8031 串行口为工作方式 1，允许串行口接收，双机通信，试确定串行口控制寄存器的内容。

答：根据题意，串行口控制寄存器 SCON 各位的设定如下：

0	1	0	1	0	0	0	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

指令为：MOV SCON, #50H ; 置串行口工作于方式 1，允许接收

8-6 若设 8031 串行口为工作方式 1，波特率为 1200bit/s， $f_{osc} = 11.0592\text{MHz}$ ， T_1 为自动装载工作方式，试确定 T_1 的定时常数 (SMOD = 0)。

解：根据 8031 串行口工作于工作方式 1 时的波特率表达式

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times T_1 \text{ 的溢出率}$$

T_1 的溢出率为定时时间的倒数，当其工作于方式 2 时， T_1 的溢出率 = $\frac{f_{osc}}{12(256 - X)}$ ，其中 X 为定时初值。

$$\begin{aligned} \text{所以} \quad \text{波特率} &= \frac{2^{\text{SMOD}}}{32} \times \frac{f_{osc}}{12 \times (256 - X)} \\ X &= 256 - \frac{f_{osc} \times 2^{\text{SMOD}}}{32 \times 12 \times \text{波特率}} = 256 - \frac{f_{osc} (\text{SMOD} + 1)}{384 \times \text{波特率}} \end{aligned}$$

代入已知参数值，得到 $X = \text{E8H}$ 。

8-7 试叙述利用 SM2 控制位进行多机通信的过程。

答：对于主从式多机通信，只有一台主机，但从机可以有多台。MCS—51 系列单片机进行多机通信时必须在方式 2 或方式 3 下工作，作主机的单片机的 SM2 应设定为 0，作从机的 SM2 设定为 1。主机发送并由从机接收的信息有两类：一类是地址，用于指示需要和主机通信的从机地址，由串行数据第 9 位为“1”标志；另一类是数据，由串行数据第 9 位为“0”标志。由于所有的从机的 SM2 为 1，故每个从机总能在 $\text{RI} = 0$ 时收到主机发来的地址 ($\text{RB8} = 1$)。每台从机把接收的从机地址和它的本机地址进行比较，只有比较相等的从机在程序中使 $\text{SM2} = 0$ ，以便接收随之而来的数据或命令 ($\text{RB8} = 0$)。

8-8 分别用查询方式和中断方式编写一段程序，从串行口发送 16 个字节数据。设数据缓冲区的首地址为 40H。

解：设串行口工作于方式 1，系统时钟为 11.0592MHz， T_1 选用工作方式 2 作为波特率发生器，波特率为 2400bit/s，且 $\text{SMOD} = 0$ ， T_1 的计数初值 X 为

$$X = 256 - \frac{f_{osc} \times (\text{SMOD} + 1)}{384 \times \text{波特率}} = 244 = \text{F4H}$$

(1) 查询方式

ORG 0000H

```

        AJMP    MAIN
        ORG     0030H
MAIN:    MOV     TMOD, # 20H    ; 定时器 T1 工作于方式 2
        MOV     TH1, # 0F4H    ; 置定时器 T1 计数初值
        MOV     TL1, # 0F4H
        SETB    TR1            ; 启动 T1
        MOV     SCON, # 40H    ; 串行口工作方式 1, 不允许接收
        MOV     R0, # 40H      ; R0 指向发送缓冲区首址
        MOV     R7, # 10H      ; R7 为发送数据块长度
LOOP:    MOV     A, @R0
        MOV     SBUF, A        ; 发送数据
HERE:    JNB     TI, HERE      ; 一帧未完, 继续查询
        CLR     TI            ; 一帧发完清 TI
        INC     R0
        DJNZ    R7, LOOP      ; 数据块未完转 LOOP 继续发送
        SJMP    $            ; 发送完, 结束

```

(2) 中断方式

```

        ORG     0000H
        AJMP    MAIN
        ORG     0023H
        AJMP    ITOP
        ORG     0030H
MAIN:    MOV     TMOD, # 20H    ; 定时器 T1 工作于方式 2
        MOV     TH1, # 0F4H    ; 置定时器 T1 计数初值
        MOV     TL1, # 0F4H
        SETB    TR1            ; 启动 T1
        MOV     SCON, # 40H    ; 串行口工作方式 1, 不允许接收
        MOV     R0, # 40H      ; R0 指向发送缓冲区首址
        MOV     R7, # 10H      ; R7 为发送数据块长度
        SETB    EA            ; 开中断
        SETB    ES            ; 允许串行口中断
        MOV     A, @R0
        MOV     SBUF, A        ; 发送
        CLR     F0            ; 清发送结束标志位 F0
WAIT:    JB      F0, HERE      ; F0 = 0, 继续等待; F0 = 1, 发送结束
        AJMP    WAIT
HERE:    AJMP    $
ITOP:    CLR     TI            ; 清 TI
        INC     R0            ; 缓冲区地址加 1

```



```

MOV    A, @R0
MOV    SBUF, A          ; 发送
DJNZ   R7, HE           ; 数据块未发送完继续
SETB   F0               ; 发送完, 对 F0 置 1
CLR    ES               ; 关中断
HE:    RETI              ; 中断返回

```

8-9 编写程序：在教材图 8-6 串行口显示电路中由左至右显示 1~6 六个字符。

解：显示数据时，将 P1.2 置 1；扫描键盘时，将 P1.2 置 0，避免干扰数码管的显示。设显示数据 1~6 依次存放在 31H~36H 中。字形码“0”~“F”列在表 TAB 中。

程序如下：

```

START: SETB    P1.2          ; 允许显示数据输出
      MOV      R3, # 06H     ; 置显示位数
      MOV      R0, # 36H     ; 置显示数据区的末地址
LOOP:  MOV      A, @R0        ; 取显示数据
      MOV      DPTR, # TAB   ; 置表首地址
      MOVC     A, @ A + DPTR ; 查表
      MOV      SBUF, A       ; 输出显示代码
HERE:  JNB      TI, HERE     ; 等待发送结束
      CLR      TI            ; 发送结束, 清 TI
      DEC      R0            ; 修改显示数据区地址
      DJNZ     R3, LOOP      ; 数据未发送完, 继续发送
      CLR      P1.2         ; 发送完, 关闭显示
      RET
TAB:   DB       3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H,
      DB       7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H

```

8-10 试编写一段程序将 8031 的串行口初始化为工作方式 1，波特率为 2400bit/s，允许串行口中断。设特殊功能寄存器中无关的位为 0， $f_{\text{osc}} = 11.0592\text{MHz}$ 。

解：设 T1 选用工作方式 2 作为波特率发生器，且 SMOD=0，T1 的计数初值 X 为 F4H。初始化程序如下：

```

      ORG      0000H
      AJMP     MAIN
      ORG      0023H
      AJMP     IT0P
      ORG      0030H
MAIN:  MOV      TMOD, # 20H   ; 定时器 T1 工作于方式 2
      MOV      TH1, # 0F4H   ; 置定时器 T1 计数初值
      MOV      TL1, # 0F4H
      SETB     TR1           ; 启动 T1
      MOV      SCON, # 40H   ; 串行口工作方式 1, 不允许接收

```

SETB	EA	；开总中断
SETB	ES	；允许串行口中断
.....		
ORG	0100H	
ITOP：		；中断服务程序

第三节 增选习题及解答

一、填空题

1. 串行口中断的中断标志位是由_____产生，由_____清零。
2. 进行串行通信，通信双方需具有相同的_____和_____。
3. 8031 利用串行口工作方式 1 进行通信时，设波特率为 1200bit/s，晶振频率为 11.0592MHz，T1 选方式 2，SMOD = 0，则 T1 的重新装入值为_____。
4. 要串行口工作为 10 位异步串行通信方式，工作方式应选为_____。
5. 用串行口扩展并行口时，串行口工作方式应选为_____。
6. 按照数据传送方向，串行通信分为_____、_____、_____三种方式。
7. 计算机通信主要有两种方式：_____和_____。
8. 异步串行数据通信的帧格式由_____、_____、_____和_____组成。
9. 在串行通信中，按照传送的帧的格式，分为_____和_____。
10. 串行口启动发送过程是通过执行指令_____实现，而接收过程是在接收控制器检测到 RXD 端的_____时启动的。

填空题参考答案：

1. 硬件、软件
2. 数据格式、速率
3. E8H
4. 方式 1
5. 方式 0
6. 单工、半双工、全双工
7. 并行通信、串行通信
8. 起始位、数据位、奇偶校验位、停止位
9. 异步串行通信、同步串行通信
10. “MOV SBUF, A”、负跳变

二、选择题

1. 用 MCS—51 串行口扩展并行 I/O 口，串行口工作方式应选择_____。
A. 方式 0 B. 方式 1 C. 方式 2 D. 方式 3
2. 控制串行口工作方式的寄存器是_____。
A. TCON B. PCON C. SCON D. TMOD
3. MCS—51 单片机的串行口是_____工作的。
A. 单工 B. 半双工 C. 全双工 D. 半双工/全双工
4. MCS—51 单片机的串行口工作于方式 0 时，须满足条件_____才能进行数据接收。
A. REN = 0 B. RI = 0 C. REN = 1 且 RI = 0 D. RXD = 0
5. 8031 利用串行口通信时，设波特率为 2400bit/s，晶振频率为 11.0592MHz，T1 选方式 2，SMOD = 0，则 T1 的重新装入值为_____。
A. FFH B. FDH C. F4H D. 1DH

选择题参考答案：

1.A 2.C 3.C 4.C 5.C

三、简答题

1. 用于实现串行通信的数据传送的三种方式分别是什么？并画出示意图进行说明。

答：串行通信装置的设置不同，通信线路的连接不同，它们通信能力是不同的，按数据的传送方向可分为单工，半双工，全双工通信方式。各种方式的具体含义和示意图参考原教材串行口一章的第一节通信技术基础。

2. 波特率、字符传送速率的含义分别是什么，有什么联系？

答：(1) 波特率定义为每秒传送二进制数的位数，单位为 bit/s。

(2) 字符传送速率定义为每秒传送多少个字符，单位是字符/s，一般用来描述数据传送速率。

两者之间的关系为：波特率 = 字符传送速率 × 每个字符包括的位数（包括起始位、停止位和校验位）。

例如，有一串行通信系统，传送二进制数据的速度是每秒传送 200 个字符，每个字符附加了起始、停止和校验各一位，那么字符传送速率为 200 字符/s；波特率为 $200 \times (8 + 1 + 1 + 1)$ bit/s = 200×11 bit/s = 2200 bit/s。

四、通信波特率计算

1. 若异步通信接口按方式 2 传送，已知其每分钟传送 4800 个字符，其波特率是多少？

解：已知每分钟传送 4800 个字符，方式 2 每个字符 11 位，则

波特率 = (11 bit/字符) × (4800 字符/60s) = 880 bit/s

2. 为什么定时器 T1 用作串行口波特率发生器时，常选用工作方式 2？若已知系统晶振频率为 11.0592 MHz，串行口工作方式 1，波特率为 4800 bit/s，设定 SMOD = 0，计算 T1 的初值并写出初始化程序。

答：(1) 因为定时器 T1 工作方式 2 时，自动重装初值，编程时无需重装初值，使用方便。若选用工作方式 0 或工作方式 1，当定时器 T1 计满溢出时，需在中断服务程序中重装初值。

(2) 已知系统时钟频率 f_{osc} 和通信用的波特率，可得出定时器 T1 工作于方式 2 时的初值 X 的计算公式。

串行口工作于方式 1 的波特率表达式为

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times T_1 \text{ 的溢出率}$$

T1 的溢出率为定时时间的倒数，当其工作于方式 2 时，

$$T_1 \text{ 的溢出率} = \frac{f_{osc}}{12 \times (256 - X)}$$

其中 X 为定时初值。

所以

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times \frac{f_{osc}}{12 \times (256 - X)}$$

$$X = 256 - \frac{f_{\text{osc}} \times 2^{\text{SMOD}}}{32 \times 12 \times \text{波特率}} = 256 - \frac{f_{\text{osc}} (\text{SMOD} + 1)}{384 \times \text{波特率}}$$

代入已知参数值, 得到 $X = \text{FAH}$ 。

(3) 初始化程序:

```
MOV    TMOD, # 20H    ; 定时器 T1 工作于方式 2
MOV    TH1, # 0FAH    ; 赋定时器计数初值
MOV    TL1, # 0FAH    ; 赋重装值
```

3. 若定时器 T1 设置成工作方式 2 作波特率发生器, 已知 $f_{\text{osc}} = 12\text{MHz}$ 。求可能产生的最高和最低的波特率。

解: (1) 当定时器 T1 定时时间为最小值, 此时初值为 255, 并且 $\text{SMOD} = 1$ 时产生最高波特率, 即波特率 $= \frac{2^{\text{SMOD}}}{32} \times \frac{f_{\text{osc}}}{12 \times (256 - X)} = \frac{2}{32} \times \frac{12 \times 10^6}{12 \times (256 - 255)} = 62500$

(2) 当定时器 T1 定时时间为最大值, 此时初值为 0, 且 $\text{SMOD} = 0$ 时, 产生最低波特率, 即 波特率 $= \frac{2^{\text{SMOD}}}{32} \times \frac{f_{\text{osc}}}{12 \times (256 - X)} = \frac{1}{32} \times \frac{12 \times 10^6}{12 \times (256 - 0)} = 122$

五、各种工作方式应用编程

1. 设 8051 串行口按工作方式 1 进行串行数据通信。假定波特率为 2400bit/s, 以中断方式传送数据。请编写全双工通信程序。

解: 设系统时钟频率 $f_{\text{osc}} = 11.0592\text{MHz}$ 。当波特率为 2400bit/s 时, 可取 $\text{SMOD} = 0$, T1 的计数初值为 F4H。

程序如下:

```
ORG    0000H
AJMP   MAIN
ORG    0023H
AJMP   IT0P
ORG    0060H
MAIN:  MOV    SP, # 60H    ; 置堆栈指针
        MOV    TMOD, # 20H    ; 置 T1 工作于方式 2
        MOV    TL1, # 0F4H    ; 赋 T1 计数初值
        MOV    TH1, # 0F4H    ; 赋 T1 重装值
        SETB   TR1          ; 启动 T1
        MOV    SCON, # 50H    ; 置串行口工作方式 1, 允许接收
        MOV    PCON, # 00H    ; 设 SMOD = 0
        MOV    R0, # 20H    ; 置发送数据区首址
        MOV    R1, # 40H    ; 置接收数据区首址
        MOV    R7, # 10H    ; 置发送字节长度
        MOV    R6, # 10H    ; 置接收字节长度
        SETB   EA          ; CPU 允许中断
```

	SETB	ES	; 允许串行口中断
	MOV	A, @R0	; 取第一个数据发送
	MOV	SBUF, A	; 发送数据
HERE :	SJMP	HERE	; 等待中断
	ORG	0100H	; 中断服务程序
IT0P :	JNB	RI, SEND	; 不是接收中断, 转发送
	CLR	RI	; 是接收中断, 清除接收中断标志
	MOV	A, SBUF	; 接收数据
	MOV	@R1, A	; 将接收数据送入接收数据区
	DJNZ	R6, NEXT	; 数据块未接收完, 继续
	SJMP	\$	; 数据块接收完, 结束
NEXT :	INC	R1	; 修改数据区指针
	RETI		; 中断返回
SEND :	CLR	TI	; 是发送中断, 清除发送中断标志
	DJNZ	R7, NEXT1	; 数据块未发送完, 继续
	SJMP	\$	; 数据块发送完, 结束
NEXT1 :	MOV	A, @R0	; 取发送数据
	MOV	SBUF, A	; 发送数据
	INC	R0	; 修改数据区指针
	RETI		; 中断返回

2. 设 8051 串行口按工作方式 3 进行串行数据通信。假定波特率为 4800bit/s, 采用奇校验方式。请编写通信程序, 用中断方式实现。

解 设系统时钟频率 $f_{osc} = 11.0590\text{MHz}$, 波特率为 4800bit/s 时, 可取 SMOD = 0, T1 的计数初值为 FAH。

程序如下:

	ORG	0000H	
	AJMP	MAIN	
	ORG	0023H	
	AJMP	IT0P	
	ORG	0060H	
MAIN :	MOV	SP, # 60H	; 置堆栈指针
	MOV	TMOD, # 20H	; 置 T1 工作方式 2
	MOV	TL1, # 0FAH	; 赋 T1 计数初值
	MOV	TH1, # 0FAH	; 赋 T1 重装值
	SETB	TR1	; 启动 T1
	MOV	SCON, # 0D0H	; 串行口于方式 3, 允许接收
	MOV	PCON, # 00H	; 设 SMOD = 0
	MOV	R0, # 20H	; 置发送数据区首址
	MOV	R1, # 40H	; 置接收数据区首址

	SETB	EA	；CPU 允许中断
	SETB	ES	；允许串行口中断
	MOV	A, @R0	；取第一个数据发送
	MOV	C, PSW.0	；将奇偶标志送 C
	CPL	C	；形成奇校验
	MOV	TB8, C	；奇校验标志送 TB8
	MOV	SBUF, A	；发送数据
HERE：	SJMP	HERE	；等待中断
	ORG	0020H	；中断服务程序
ITOP：	JNB	TI, REC	；不是发送中断，转接收中断
	CLR	TI	；是发送中断，清除发送中断标志
	INC	R0	；修改数据区指针
	MOV	A, @R0	；取数据发送
	MOV	C, PSW.0	；将奇偶标志送 C
	CPL	C	；形成奇校验
	MOV	TB8, C	；奇校验标志送 TB8
	MOV	SBUF, A	；发送数据
	RETI		
REC：	CLR	RI	；是接收中断，清接收中断标志
	MOV	A, SBUF	；接收数据
	MOV	C, PSW.0	；将奇偶标志送 C
	CPL	C	；形成奇校验
	JC	L1	；判断接收端的奇偶值，C = 1 转 L1
	JNB	RB8, RIGHT	；C = 0, RB8 = 0 转 RIGHT
		SJMP ERROR	；C = 0, RB8 = 1 转出错处理
L1：	JB	RB8, RIGHT	；C = 1, RB8 = 1 转 RIGHT
	SJMP	ERROR	；C = 1, RB8 = 0 转出错处理
RIGHT：	MOV	@R1, A	；将接收数据送入接收数据区
	INC	R1	；修改数据区指针
	RETI		；中断返回
ERROR：	SETB	F0	；因 $C \neq RB8$ ，校验为错，置位出错标志 F0
	SJMP	\$	；一旦发现有错立即停止执行程序

3. 在 8051 片内的 RAM 30H ~ 4FH 单元中有 32 个字节的数据，若采用方式 1 进行串行通信，波特率为 1200bit/s， $f_{osc} = 6\text{MHz}$ ，用查询和中断两种方式编写发送/接收程序对。

解：设 T1 选用工作方式 2 作为波特率发生器，且 SMOD = 0，T1 的计数初值 X 为 F3H。

(1) 查询方式程序

1) 发送程序

```

ORG      0000H
AJMP     MAIN

```

```

                ORG            0030H
MAIN:           MOV            TMOD, # 20H    ; 定时器 T1 工作于方式 2
                MOV            TH1, # 0F3H    ; 置定时器 T1 计数初值
                MOV            TL1, # 0F3H
                SETB           TR1            ; 启动 T1
                MOV            SCON, # 40H    ; 串行口工作方式 1, 不允许接收
                MOV            R0, # 30H      ; R0 指向发送缓冲区首址
                MOV            R7, # 20H      ; R7 为发送数据块长度
LOOP:           MOV            A, @R0
                MOV            SBUF, A        ; 发送数据
HERE:           JNB           TI, HERE        ; 一帧未发完, 继续查询
                CLR            TI            ; 一帧发完清 TI
                INC            R0
                DJNZ           R7, LOOP        ; 数据块未发完转 LOOP 继续发送
                SJMP           $              ; 发送完, 结束

```

2) 接收程序

```

                ORG            0000H
                AJMP           MAIN
                ORG            0030H
MAIN:           MOV            TMOD, # 20H    ; 定时器 T1 工作方式 2
                MOV            TH1, # 0F3H
                MOV            TL1, # 0F3H    ; 设置 T1 计数初值
                SETB           TR1            ; 启动 T1
                MOV            SCON, # 50H    ; 设定串行方式 1 并允许接收
                MOV            R0, # 30H      ; R0 为接收缓冲区首址
                MOV            R7, # 20H      ; R7 为接收数据块长度
HERE:           JNB           RI, HERE        ; 一帧收完? 未收完等待
                CLR            RI            ; 收完清 RI
                MOV            A, SBUF
                MOV            @R0, A        ; 将数据读入接收数据缓冲区
                INC            R0            ; 修改地址指针
                DJNZ           R7, HERE
                SJMP           $

```

(2) 中断方式程序

1) 中断发送程序

```

                ORG            0000H
                AJMP           MAIN
                ORG            0023H
                AJMP           ITOP

```

2) 中断接收程序

	ORG	0000H	
	AJMP	MAIN	
	ORG	0023H	
	AJMP	IT0P	
	ORG	0030H	
MAIN:	MOV	TMOD, # 20H	; 定时器 T1 工作方式 2
	MOV	TH1, # 0F3H	
	MOV	TL1, # 0F3H	; 设置 T1 计数初值
	SETB	TR1	; 启动 T1
	MOV	SCON, # 50H	; 设定串行方式 1 并允许接收
	MOV	R0, # 30H	; R0 为接收缓冲区首址
	MOV	R7, # 20H	; R7 为接收数据块长度
	SETB	EA	; 开中断
	SETB	ES	; 允许串行口中断

	CLR	F0	；清接收结束标志位 F0
WAIT：	JB	F0，HERE	；F0 = 0，继续等待；F0 = 1，接收结束
	AJMP	WAIT	
HERE：	AJMP	\$	
ITOP：	CLR	RI	；清 RI
	MOV	A，SBUF	
	MOV	@R0，A	；读入接收数据
	INC	R0	；缓冲区地址加 1
	DJNZ	R7，HE	；数据块未接收完继续
	SETB	F0	；接收完毕，对 F0 置 1
	CLR	ES	；关中断
HE：	RETI		；中断返回

第八章 MCS—51 单片机并行口扩展及应用

第一节 学习要求与内容概要

一、学习要求

- (1) 了解单片机并行口基本概念与功能，掌握单片机扩展并行口的基本总线要求。
- (2) 熟悉 8155 芯片结构、功能、控制字，掌握单片机接口方法与程序设计方法。
- (3) 熟悉 8255 芯片结构、功能、控制字，掌握单片机接口方法与程序设计方法。

二、单片机并行口扩展内容概要

接口是计算机连接外部设备的端口电路，是 CPU 连接外设的桥梁。一个完整的接口电路应该包括：数据端口、控制端口和状态端口三种形式。接口电路完成的功能有：地址译码和 I/O 设备选择功能、输入与输出功能、信息电平转换功能、联络功能和可编程功能。

MCS—51 单片机本身有四个以并行方式传输数据的并行 I/O 端口：P0、P1、P2 和 P3，而并行口电路端口地址采用和外部扩展 RAM 统一编制的形式进行编排，一个端口相当于一个外扩 RAM 字节单元。这样，与读、写外 RAM 指令一样，实现接口单元的读、写操作，其指令为：

读端口：MOVX A, @DPTR ; DPTR 的内容为 16 位端口地址
MOVX A, @Ri ; Ri (R0 或 R1) 的内容为低 8 位端口地址
写端口：MOVX @DPTR, A ; DPTR 的内容为 16 位端口地址
MOVX @Ri, A ; Ri (R0 或 R1) 的内容为低 8 位端口地址

MCS—51 单片机扩展并行口和扩展外部 RAM 一样需要三大总线，即数据总线、地址总线和控制总线。数据总线由 P0 口提供；地址总线由 P0 口和 P2 口构成；控制总线主要是 $\overline{WR}$ 、 $\overline{RD}$ 和 ALE。

挂接在同一数据总线上的多个外设的并行口都应该有一个有别于其他端口操作的端口地址，但是，一个输入端口和一个输出端口可以有相同的端口地址。

三、8255A 并行口扩展学习内容概要

8255A 是可编程并行口芯片，扩展了 A、B 和 C 共三个 8 位 I/O 口，内部还有一个存放 I/O 口工作方式控制字及 C 口按位操作控制字的端口，因此，一个 8255A 芯片实际占用四个端口地址。

A 口：数据输出锁存/缓冲、数据输入锁存；B 口：数据输出锁存/缓冲、数据输入缓冲；C 口：数据输出锁存/缓冲、数据输入缓冲。三个端口分成两个工作组，A 口和 C 口上半部为 A 组，B 口和 C 口下半部为 B 组。

8255A 的数据线为 D0 ~ D7，控制线为 $\overline{RD}$ 、 $\overline{WR}$ 、RESET，地址线为 $\overline{CS}$ 和 A1、A0。在 $\overline{CS}$ 为低电平有效时，由 A1、A0 选择读或写操作端口。

对 8255A 编程的控制字有两个，一个是端口 A、B、C 工作方式选择控制字，另一个是 C 口按位操作控制字，两个控制字均存入控制字寄存器（端口 C）中，它们由最高位 D7 的内容予以区别。图 8-1 是工作方式控制字格式示意图。

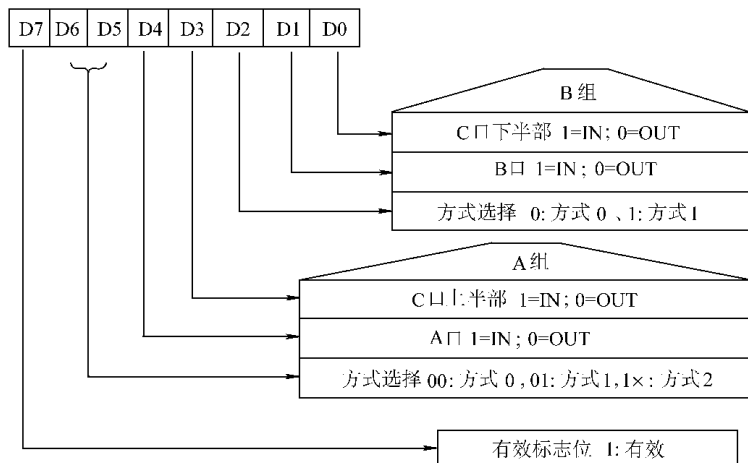


图 8-1 8255 方式控制字寄存器的格式

端口 A 具有三种工作方式：方式 0（基本 I/O 方式）、方式 1（选通工作方式）和方式 2（双向选通传输方式）。B 口具有两种工作方式：方式 0 和方式 1。如果 A 口或者 B 口工作在选通工作方式时，C 口中的一些口线必须作为选通口的控制信号线或状态信号线，而 C 口剩余的口线可以作为一般 I/O 口使用。A 口、B 口和 C 口可以同时工作在方式 0。A 口和 B 口不论工作在基本 I/O 方式还是选通工作方式都作为数据 I/O 口使用。

工作在方式 0 时，三个端口互相独立地设定为数据输入口或者数据输出口，且 C 口的上半口和下半口可以分开设置为输入口或输出口。C 口中的口线作为基本 I/O 口时，可以利用 C 口按位操作控制字对单个口线进行置位或者复位操作。

图 8-2 是 C 口按位操作控制字格式示意图。

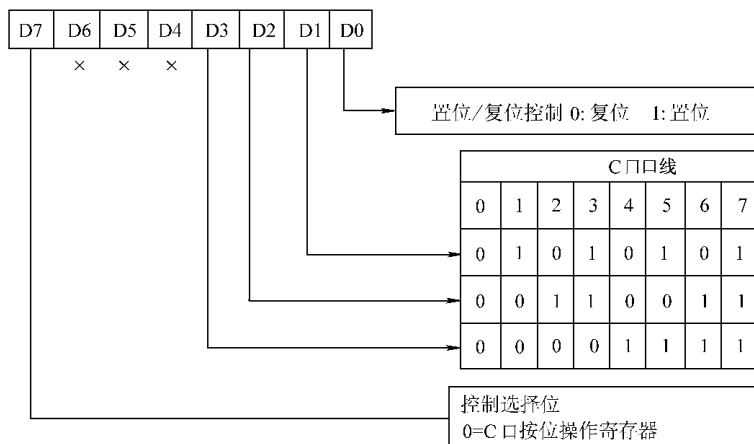


图 8-2 C 口按位操作控制寄存器

A 口或 B 口工作在方式 1 时，根据它们作为数据输入口或者数据输出口的不同，C 口分

别有不同口线被用来作为选通工作方式口的控制线或状态线。

A 口或 B 口工作在方式 1 且作数据输入口时, C 口提供的协作口线有:

协作 A 口的口线: $\overline{STB}$ (PC4)、IBF (PC5)、INTR (PC3)、INTE (PC4)。

协作 B 口的口线: $\overline{STB}$ (PC2)、IBF (PC1)、INTR (PC0)、INTE (PC2)。

A 口或 B 口工作在方式 1 且作数据输出口时, C 口提供的协作口线有:

协作 A 口的口线: $\overline{OBF}$ (PC7)、 $\overline{ACK}$ (PC6)、INTR (PC3)、INTE (PC6)。

协作 B 口的口线: $\overline{OBF}$ (PC1)、 $\overline{ACK}$ (PC2)、INTR (PC0)、INTE (PC2)。

只有 A 口能够以方式 2 工作, 这时 B 口只能工作在方式 0 状态。C 口与工作于方式 2 的 A 口协作: INTR (PC3)、 $\overline{OBF}$ (PC7)、 $\overline{ACK}$ (PC6)、IBF (PC5)、INTE1 (PC6)、INTE2 (PC4)。

需要强调的是: $\overline{STB}$ 是由外设提供的控制信号; IBF 和 $\overline{OBF}$ 是由 8255A 发出的状态信号; $\overline{ACK}$ 是由外设提供的响应信号; INTR 是由 8255A 输出中断申请信号; INTE 是由 CPU 发来的控制信号。

关于 8255A 的使用。首先应该根据接线图正确地计算出四个端口的地址, 然后根据应用需要对 8255A 进行初始化 (编写工作方式控制字)。对于基本 I/O 工作方式, 程序可以随时读取端口数据或者向端口写入数据, 但对于选通工作方式, 需要查询端口的状态, 再对端口进行相应的读、写处理。

初始化 8255A 的指令程序为:

MOV DPTR, # PORT-CONTROL ; 置控制字端口地址 PORT-CONTROL

MOV A, # data ; 控制字, 如 90H: 基本 I/O, A 口输入, B 口、C 口输出

MOVX @DPTR, A

接着可以对端口进行读、写操作, 如将 PC7 置位的程序为:

MOV DPTR, # PORT-CONTROL ; 置控制字端口地址 PORT-CONTROL

MOV A, # 0FH ; 只将 PC7 置位

MOVX @DPTR, A

读取 A 端口内容的程序为:

MOV DPTR, # PORT-A ; 置端口 A 地址 PORT-A

MOVX A, @DPTR

四、8155 并行口扩展学习内容概要

8155 是可编程并行接口芯片。它内部具有 256 个 RAM 字节单元、二个 8 位 I/O 端口 A 和 B、一个 6 位 I/O 端口 C 以及一个 14 位计数器, 还有一个命令/状态字寄存器。由于 14 位计数器占用两个端口地址单元, 故 8155 内部除了 256 个 RAM 字节单元外还有 6 个端口单元。

8155 芯片与 MCS-51 单片机连接的三大总线是: $AD_0 \sim AD_7$ 、 $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{CE}$ 、 $IO/\overline{M}$ 、RESET 和 ALE, 其中 $AD_0 \sim AD_7$ 为 8 位地址、数据复用线, 且 8155 内部有地址锁存器, $IO/\overline{M}$ 为 8155 内部端口选择或 RAM 选择控制线。当 $IO/\overline{M} = 1$ 时, CPU 选择 8155 端口; 当 $IO/\overline{M} = 0$ 时, CPU 选择内部 RAM。

命令/状态字寄存器分为 I/O 口工作方式控制字寄存器和存放 A 口和 B 口的工作状态的

标志寄存器两部分，两者共用命令/状态字口地址。前者只能写入而不能读出，其格式如图 8-3 所示。后者只能读出而不能写入，其格式如图 8-4 所示。

D7	D6	D5	D4	D3	D2	D1	D0
×	TIMER	INTEB	BFB	INTRB	INTEA	BFA	INTRA

INTRA-A口中断请求 ;BFA-A口缓冲器满 ;INTEA-A口中断允许

INTRB-B口中断请求 ;BFB-B口缓冲器满 ;INTEB-B口中断允许 ;TIMER- 定时器中断请求

图 8-3 8155 状态标志寄存器格式

和 8255A 一样，8155 的 A 口、B 口和 C 口均可工作在基本 I/O 方式（ALT1 和 ALT2）；而 A 口和 B 口还可以工作在选通方式（ALT3 和 ALT4），这时 C 口的某些口线固定地用作选通口的控制线或状态线。

PC0：INTRA，A 口中断申请信号线；

PC1：BFA，A 口缓冲器满信号线；

PC2： $\overline{STBA}$ ，A 口选通信号线；

PC3：INTRB，B 口中断申请信号线；

PC4：BFB，B 口缓冲器满信号线；

PC5： $\overline{STBB}$ ，B 口选通信号线。

8155 计数器是一个减法计数器，通过设置不同的初值以及设定信号输出方式，可以得到四种形式的输出脉冲波形。

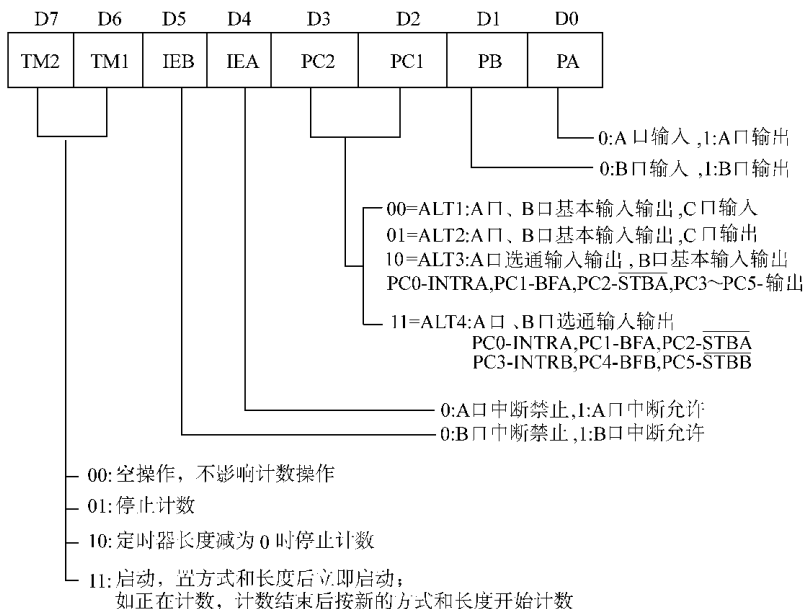


图 8-4 8155 控制命令寄存器格式

第二节 原教材习题及解答

9-1 单片机系统为什么要进行 I/O 接口扩展？I/O 接口应具有哪些基本功能？

答：MCS-51 系列单片机内部有四个 8 位 I/O 接口，但是外部扩展了 RAM、ROM 以后，可供外部设备输入、输出信息使用的只有 P1 口了。当连接外部设备两 I/O 接口不够时，需要进行 I/O 接口线扩展。

I/O 接口应具有的功能：地址译码和 I/O 设备选择功能、输入与输出功能、信息电平转换功能、联络功能和可编程功能。

9-2 8255A 有哪几种工作方式？分别是什么？端口 A、B、C 的特点有哪些？

答：8255A 有三种工作方式。它们分别是：方式 0（基本输入/输出方式）、方式 1（选通输入/输出方式）和方式 2（双向数据传送方式，仅适用于 A 口）。

端口 A、B、C 的特点：

A 口：具有一个 8 位数据输入锁存器和一个 8 位数据输出锁存/缓冲器，可工作于方式 0、方式 1 或方式 2。它总是作为数据输入/输出端口。

B 口：具有一个 8 位数据输入缓冲器和一个 8 位数据输出锁存/缓冲器，可工作于方式 0 或方式 1。它总是作为数据输入/输出端口。

C 口：具有一个 8 位数据输入缓冲器（不锁存）和一个 8 位数据输出锁存/缓冲器，C 口可作为基本的数据输入/输出端口，有时也作为控制信号和状态信号的输入/输出端口。

9-3 为什么说 8255A 各端口的输入/输出状态可构成 16 种组合？

答：在方式 0 时，C 口可以分成上半口和 C 口下半口单独设定为输入口或输出口，这样 A 口、B 口和 C 口实际上可作为 4 个口使用，故 4 个端口的输入/输出状态可构成 $2^4 = 16$ 种组合。

9-4 8255 的 A 口和 B 口在工作方式 1（选通工作方式）输入/输出状态时，C 口各引脚的功能是什么？

答：(1) A 组工作在方式 1 并作输入时，部分 C 口的口线为它协作服务，它们是：

PC4： $\overline{STBA}$

PC5：IBFA

PC3：INTRA

B 组工作在方式 1 并作输入时，部分 C 口的口线为它协作服务，它们是：

PC2： $\overline{STBB}$

PC1：IBFB

PC0：INTRB

其中

$\overline{STB}$ 选通输入控制信号，低电平有效，用来将外设送来的输入数据送入输入锁存器。

IBF 输入缓冲信号满信号，高电平有效，由 8255 输出，有效表示数据已输入至输入锁存器。

INTR 中断请求信号，高电平有效，由 8255 输出，用于向 CPU 申请中断。

INTE 中断允许信号。INTEA 由 PC4 控制置位/复位，INTEB 由 PC2 控制置位/复位。

② A 组工作在方式 1 并作输出时, 部分 C 口的口线为它协作服务, 它们是:

PC7: $\overline{\text{OBFA}}$

PC6: $\overline{\text{ACKA}}$

PC3: INTRA

B 组工作在方式 1 并作输出时, 部分 C 口的口线为它协作服务, 它们是:

PC1: $\overline{\text{OBFb}}$

PC2: $\overline{\text{ACKB}}$

PC0: INTRB

其中

$\overline{\text{OBF}}$ 输入缓冲器满信号, 低电平有效, 由 8255 送给外设, 有效表示 CPU 已经向指定的端口输出了数据。

$\overline{\text{ACK}}$ 外设响应信号, 低电平有效, 由外设送给 8255, 有效表示 CPU 通过 8255 输出的数据已经送到外设。

INTR 中断请求信号, 高电平有效。

INTE 中断允许信号。 INTEA 由 PC6 控制置位/复位, INTEB 由 PC2 控制置位/复位。

9-5 设将 8255A 置为基本输入/输出方式, A 口、C 口输出, B 口输入, 试确定 8255A 的工作方式控制字寄存器的内容。

答: 控制字内容为: 82H。

9-6 若将 8255A 的 PC7 置为高电平, 向按位操作控制字写入的内容是什么?

答: 按位操作控制字内容为: 0FH。

9-7 原教材图 9-8 所示为 8031 与 8255A 的接口电路, 试编写一段程序将 8255A 初始化为基本输入输出方式, A 口、B 口输出, C 口输入, 并从 C 口读入一个数据, 从 A 口、B 口输出一个数据。

解: 8255A 各 I/O 口和控制字寄存器的地址为:

A 口: 7FFCH; B 口: 7FFDH; C 口: 7FEH; 控制字寄存器: 7FFFH

程序:

```
MOV    DPTR, # 7FFFH      ;置控制字的地址
MOV    A, # 89H           ;置控制字的内容
MOVX   @DPTR, A
... ..

MOV    DPTR, # 7FEH       ;置 C 口的地址
MOVX   A, @DPTR           ;将 C 口数据读入累加器 A
... ..

MOV    DPTR, # 7FFCH      ;置 A 口的地址
MOV    A, # DATA1
MOVX   @DPTR, A           ;从 A 口读出数据 DATA1
... ..

MOV    DPTR, # 7FFDH      ;置 B 口的地址
MOV    A, # DATA2
```

MOVX @DPTR, A ; 从 B 口读出数据 DATA2

9-8 8031 与 8255A 接口电路如原教材图 9-8, 试编写程序将 8255A 的 PC7 清 0。

解: 8255A 的各 I/O 口和控制字寄存器的地址为:

A 口: 7FFCH; B 口: 7FFDH; C 口: 7FFEh; 控制字寄存器: 7FFFH

由题意知, C 口应置为输出口, 故程序如下:

MOV DPTR, # 7FFFH ; 置控制字的地址

MOV A, # 0EH ; 置控制字的内容, PC7 清 0

MOVX @DPTR, A

9-9 8155 内部有哪些功能部件? 如何寻址这些部件?

答: 8155 内部有两个 8 位并行 I/O 端口 (A 口和 B 口), 一个 6 位并行输入/输出端口 (C 口), 一个 256 字节的静态 RAM 以及一个 14 位“减 1”计数器。

在片选信号 $\overline{CE}$ 为 0 (有效) 的情况下, 当控制信号 $IO/\overline{M}=0$ 时, CPU 寻址 8155 内部数据存储单元, 而当 $IO/\overline{M}=1$ 时 CPU 寻址 8155 I/O 口、命令/状态寄存器以及计数器单元。

9-10 8155 的 C 口为控制信号时, 各引脚的功能是什么?

答: PC0 INTRA, A 口中断申请;

PC1 BFA, A 口缓冲器满;

PC2 $\overline{STBA}$, A 口选通;

PC3 INTRB, B 口中断申请;

PC4 BFB, B 口缓冲器满;

PC5 $\overline{STBB}$, B 口选通。

9-11 设将 8155 置为基本输入/输出方式, A 口、B 口输出, C 口输入, 试确定 8155 的工作方式控制字寄存器的内容。

答: 工作方式控制字的内容为 C3H。

9-12 在单片机系统中, 非编码键盘有哪几种形式? 各有什么特点?

答: 非编码键盘可以分为两种结构形式: 独立式按键和行列式按键。

独立式按键是各按键相互独立地接通一条输入数据线, 电路结构和处理程序简单; 缺点是当按键数较多时, 要占用较多的 I/O 口线。

行列式按键将 I/O 口线分作行线与列线两部分。行线与列线交叉处通过按键来连通, 这种结构只需要 N 条行线和 M 条列线即可组成具有 $N \times M$ 个按键的键盘, 适应于键数较多的场合。

9-13 8031 与键盘、数码管显示器接口电路如原教材图 9-25 所示, 试编写程序, 将键入的数字 (0~F) 显示在 6 位数码管显示器上 (左进右出, 逐位右移)。

解: 新键入数字在数码管最左位显示, 已键入数字则在数码管上顺序右移。

主程序:

MAIN: MOV R0, # 79H ; 键入键码缓冲区 79H~7E 置初值 00H

MOV R7, # 06H ; 79H 内容显示在最右边 LED 上

LOOP1: MOV @R0, # 00H

INC R0

DJNZ R7, LOOP1


```

MOV    DPTR, # 7F00H    ; 8155 命令/状态寄存器地址
MOV    A, # 03H         ; 基本 I/O 方式, A、B 出, C 入,
MOVX   @DPTR, A
LK1:   ACALL KEYCLOSE    ; 调用: 有无按键判别子程序
      JZ     LK1         ; 无键闭合, 继续等待
      ACALL TIM          ; 调用延时子程序, 消除键盘抖动
      ACALL TIM
      ACALL KEYCLOSE    ; 调用: 有无按键判别子程序
      JZ     LK1         ; 无键闭合, (A) = 0, 返回 LK1
LK:    ACALL KEYCODE     ; 调用按键识别子程序, 结果存 A
      MOV    R0, # 7EH   ; 新键入的键号存入缓冲区
      MOV    R7, # 06H   ; 原有数据右移
LOOP2: XCH    A, @R0
      DEC    R0
      DJNZ   R7, LOOP2
      ACALL  DIR          ; 调用显示子程序
      AJMP   LK1

```

显示子程序:

```

DIR:   MOV    R0, # 79H   ; 置显示缓冲区首地址
      MOV    R3, # 01H   ; 置位选码初值
      MOV    A, R3
LOOP3: MOV    DPTR, # 7F01H ; A 口地址送 DPTR
      MOVX   @DPTR, A     ; 输出位选码, 一个 LED 显示
      INC    DPTR         ; DPTR 指向 B 口
      MOV    A, @R0       ; 取被显示的数据
      ADD    A, # 14      ; 形成查表的偏移地址
      MOVC   A, @A + PC   ; 求出该键值显示的字型码
      MOVX   @DPTR, A     ; 输出显示代码
      ACALL  TIM          ; 延时
      INC    R0           ; 指向下一个显示缓冲区单元
      MOV    A, R3
      JB     ACC. 5, LOOP4 ; 6 位显示完毕则转 LOOP4
      RL     A            ; 显示位左移
      MOV    R3, A
      AJMP   LOOP3
LOOP4: RET
      DB     3FH, 06H, 5BH, 4FH, 66H, 6DH ; 共阴极 “0 ~ F” 字型
      DB     7DH, 07H, 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H, 00H

```

按键识别子程序:

```

KEYCODE: MOV    R2, # 0FEH    ; A 口输出第一次扫描代码
          MOV    R4, # 00H    ; 列号送 R4
L:        MOV    DPTR, # 7F01H ; A 口输出列扫描值, 逐列扫描
          MOV    A, R2
          MOVX   @DPTR, A      ; 输出扫描信号
          INC    DPTR
          INC    DPTR          ; C 口读入行状态值
          MOVX   A, @DPTR      ; 读 C 口状态
          JB     ACC. 0, L1     ; PC0 = 1, 第 0 行无键按下转移
          MOV    A, # 00H      ; 0 行 0 列键号送 A
          AJMP   CODE
L1:       JB     ACC. 1, L2     ; PC1 = 1, 第一行无键按下转移
          MOV    A, # 04H      ; 1 行 0 列键号送 A
          AJMP   CODE
L2:       JB     ACC. 2, L3     ; PC2 = 1, 第二行无键按下转移
          MOV    A, # 08H      ; 2 行 0 列键号送 A
          AJMP   CODE
L3:       JB     ACC. 3, NEXT   ; PC3 = 1, 第三行无键按下转移
          MOV    A, # 0CH      ; 3 行 0 列键号送 A
CODE:     ADD    A, R4          ; 形成键码
NS:       RET
NEXT:     INC    R4             ; 扫描列号加 1
          MOV    A, R2          ; 扫描值送 A
          JNB    ACC. 3, NS     ; 4 列扫描完毕, 返回
          RL     A              ; 形成下次扫描代码
          MOV    R2, A
          AJMP   L

```

判有无按键子程序:

```

KEYCLOSE: MOV    DPTR, # 7F01H ; A 口地址
          MOV    A, # 00H
          MOVX   @DPTR, A      ; 输出扫描代码 00H
          INC    DPTR
          INC    DPTR
          MOVX   A, @DPTR      ; 读入 C 口状态
          CPL    A
          ANL    A, # 0FH      ; 屏蔽高 4 位
          RET

```

软件延时 6ms 子程序:

```

TIM:      MOV    R7, # 0CH

```

```

DLT1:    MOV     R6, # 0FFH
DLT2:    DJNZ    R6, DLT2
          DJNZ    R7, DLT1
          RET

```

9-14 利用原教材图 9-28 所示 8031 与 TP uP_40A 打印机接口电路, 编写打印字符串 “THIS IS TEST !” 的子程序。

解: 打印机相对于 CPU 速度较慢。CPU 一般以应答形式向打印机发送数据。TP uP_40A 微型打印机除了数据线 DB0 ~ DB7 以外, 设置了三个主要应答信号线:

$\overline{\text{STB}}$: 数据选通信号。CPU 发出的控制信号, 其上升沿使数据线上的数据送入打印机的内部锁存器。

$\overline{\text{ACK}}$: 打印机给 CPU 的应答信号。该信号低电平有效, 有效时表明打印机已取走数据线上的数据。

BUSY: 打印机输出的忙状态信号, 高电平时表明打印机正在打印数据, 低电平时“闲”, 可输入数据。

8155 设置为基本 I/O 方式, A 口输出数据, C 口输出且 PC0 用于产生 $\overline{\text{STB}}$ 信号, 打印机的 “BUSY” 接中断引脚 (P3.2)。当 “BUSY” 由高电平变为低电平即打印机由忙变闲时 PC0 产生上升沿跳变, 使 “ $\overline{\text{STB}}$ ” 有效, 把数据打入打印机。

方法一 查询法实现的子程序:

```

PRINT_CH:  MOV     A, # 82H           ; 方式 0, A、C 口出, B 口入
            MOV     DPTR, # 7FFFH     ; 控制命令寄存器地址
            MOVX    @DPTR, A
            MOV     R7, # 13          ; “THIS IS TEST !” 共 13 个字符
            MOV     DPTR, # CHA_TAB   ; “THIS IS TEST !” 数据表首地址
PTRL_LOOP: MOV     A, # 00H
            MOVC    A, @A + DPTR       ; 取表中每个字符的 ASCII 码送 A
            ACALL   CHA_OUT            ; 调用字符输出子程序
            INC     DPTR
            DJNZ    R7, PRL_LOOP
            MOV     A, # 0DH           ; 输出 ‘CR’ 命令, 回车符
            ACALL   CHA_OUT
            RET

```

输出字符子程序:

```

CHA_OUT:   PUSH    DPL                ; 保存查表地址
            PUSH    DPH
            MOV     DPTR, # 7FFCH     ; A 口地址
            MOVX    @DPTR, A          ; 一个打印字符送 A 口
            MOV     DPTR, # 7FFEh     ; C 口地址
            MOV     A, # 00H          ; PC0 置 0

```

```

MOVX    @DPTR, A
NOP
NOP
NOP
NOP
MOV     A, # 01H           ; PC0 置 1, 产生 $\overline{STB}$ 上升沿跳变
MOVX    @DPTR, A
WAIT_LOW: JB      P3. 2, WAIT_LOW ; 若“BUSY”为“忙”, 等待
POP     DPH                ; 恢复查表地址
POP     DPL
RET

```

CHA_TAB: DB 'THIS IS TEST!' ; 打印字符的 ASCII 数据表

方法二 中断法实现的子程序:

```

ORG     0003H
AJMP    PRT
ORG     2000H
PRINT_CHA: SETB   EA           ; 开所有中断
          SETB EX0           ; 允许 $\overline{INT0}$ 中断
          SETB IT0          ; 令 $\overline{INT0}$ 为边沿出发中断
MOV     A, # 82H             ; 方式 0, A、C 口出, B 口入
MOV     DPTR, # 7FFFH        ; 控制命令寄存器地址
MOVX    @DPTR, A
MOV     DPTR, # CHA_TAB      ; “THIS IS TEST!” 数据表首地址
MOV     A, # 00H
MOVC    A, @A + DPTR         ; 表中第一个字符的 ASCII 码送 A
INC     DPTR                 ; 形成下一个要取表字符的地址
MOV     R5, DPL              ; 保存下一个要取表字符的地址
MOV     R4, DPH
MOV     DPTR, # 7FFCH        ; A 口地址
MOVX    @DPTR, A             ; 第一个字符输出到 A 口
MOV     DPTR, # 7FFEh        ; C 口地址
MOV     A, # 00H             ; PC0 置 0
MOVX    @DPTR, A
NOP
NOP
NOP
NOP
MOV     A, # 01H

```

```

MOVX    @DPTR, A          ; PC0 置 1, 产生 $\overline{STB}$ 上升沿跳变
MOV     R7, # 12          ; “THIS IS TEST !” 还剩 12 个字符
CLR     F0                ; F0 全部打印完成以后置位
HERE:   JNB     F0, HERE   ; F0 等待中断, 直到全部数据打印完毕
RET

```

中断服务子程序：

```

PRT:    MOV     DPL, R5      ; 下一个要取表字符的地址送 DPTR
        MOV     DPH, R4
        MOV     A, # 00H
        MOVC    A, @A + DPTR ; 取表中每个字符的 ASCII 码, 送 A
        INC     DPTR        ; 形成下一个要取表字符的地址
        MOV     R5, DPL     ; 保存下一个要取表字符的地址
        MOV     R4, DPH
        MOV     DPTR, # 7FFCH ; A 口地址
        MOVX    @DPTR, A    ; 第一个字符输出到 A 口
        MOV     DPTR, # 7FEH ; C 口地址
        MOV     A, # 00H    ; PC0 置 0
        MOVX    @DPTR, A
        NOP                      ; 使低电平有一定的宽度
        NOP
        NOP
        NOP
        MOV     A, # 01H      ; PC0 置 1, 产生 $\overline{STB}$ 上升沿有效信号
        MOVX    @DPTR, A
        DJNZ    R7, DONE     ; 若未打完则继续
        SETB    F0          ; 全部数据打印完时, 置位 F0
        CLR     EX0         ; 打印完, 关中断
DONE:    RETI
CHA_TAB: DB      'THIS IS TEST !'
        END

```

第三节 增选习题及解答

一、填空题

1. 8155 内部计数器有____种脉冲输出方式, 具体输出方式由____决定。

2. 写 8255A 控制字的时候, 需将 A1、A0 置为____。

3. 8255A 有两个控制字:____和____。

4. 设将 8155 置为基本输入/输出方式: A 口、B 口输出, C 口输入, 启动计数, A 口、B 口禁止中断, 则应向 8155 工作方式控制字寄存器写入____。

5. 非编码键盘可分为两种结构形式：____和____。

填空题参考答案：

1. 四种、计数器高字节中 M1、M2 两位（最高两位）的状态 2. 1, 1 3. 工作方式控制字、C 口按位操作控制字 4. C3H 5. 独立式按键、行列式按键

二、选择题

1. 使用 8255A 可以扩展出____根 I/O 口线？

A. 16 根 B. 24 根 C. 22 根 D. 32 根

2. 8255A 端口 A、B 均工作在方式 0 的输入方式时，端口 C 可作为____。

A. 两个 4 位 I/O 端口或一个 8 位 I/O 端口 B. 状态端口
C. 部分引脚作端口 A、端口 B 的联络信号 D. 全部作联络信号

3. 8255A 端口 B 设置为方式 1 且为输出，其 $\overline{\text{ACKB}}$ 收到一个负脉冲说明____。

A. CPU 已将一个数据写到端口 B B. 请求 CPU 送下一个数据
C. 外设已将数据由端口 B 取走 D. INTRB 中断请求被 CPU 响应

4. 若要求 8155 计数器输出连续方波，则须将计数器中的 M2、M1 置为____。

A. 0, 0 B. 0, 1 C. 1, 0 D. 1, 1

5. 8155 的控制命令寄存器____。

A. 只能写入，不能读出 B. 只能读出，不能写入
C. 既能读出，也能写入 D. 不能读出，也不能写入

选择题参考答案：

1. B 2. A 3. C 4. B 5. A

三、简答题

1. 8255A 由哪几部分组成？有哪几种工作方式？

答：8255A 有以下四部分组成：数据端口 A、B、C；A 组控制和 B 组控制；读/写控制逻辑；数据总线缓冲器。

8255A 有三种工作方式：1) 工作方式 0：基本输入输出工作方式；2) 工作方式 1：选通式输入输出工作方式；3) 工作方式 2：双向传送方式，仅适用于端口 A。

2. 8255A 的 B 口可以设定为方式 2 吗？为什么？

答：8255A 的 B 口不可以设定为方式 2。只有 A 口可以方式 2 工作，为双向口，其输入或输出都有独立的状态信息，占用 C 口的 5 根联络线（PC3，……，PC7）。这时，C 口不能为 B 口提供足够的联络线，故 B 口不能工作在方式 2 和方式 1，只能工作在方式 0。

3. 8155 有哪几种工作方式？怎样进行选择？

答：8155 有三种工作方式，即通用 I/O 口、CPU 片外 256 字节 RAM 存取以及定时器/计数器方式。

IO/ $\overline{\text{M}}$ 引脚置为高电平时：

1) 作为通用 I/O 口。I/O 工作方式选择是通过对 8155 内部命令寄存器送命令字来实现的。

2) 作为定时器/计数器使用。分别选择定时器/计数器的高、低 8 位寄存器地址，通过对

高 8 位的高 2 位的设置可定义 4 种输出波形。高 8 位中的低 6 位和低 8 位寄存器共 14 位，用于存放 14 位计数器初值。

$\text{IO}/\overline{\text{M}}$ 引脚置为低电平时：作为 CPU 片外 256B 的 RAM 使用，其寻址范围由片选线 $\overline{\text{CE}}$ （高位地址译码）和 AD0 ~ AD7 决定，应与应用系统中其他数据存储单元统一编址。

四、8255A 接口芯片应用

1. 用单片机完成某生产过程控制。该过程有六个工序，循环进行，每个工序时间相等，均为 10s。单片机通过 8255A 的 A 口进行控制。A 口中的一位控制一个工序的启停。设 8255A 的 A 口地址为 90H，B 口地址为 91H，C 口地址为 92H，控制口地址为 93H。试编写有关程序。设时钟为 6MHz。

解：设利用 8255A 中 A 口 PA0 ~ PA5 的 6 位完成六个工序控制。规定某一位为 1 时启动一道工序，为 0 时停止该工序。8255A 初始化只需规定 A 口为输出方式即可，控制字为 80H。利用软件延时实现时间控制，程序如下：

```

MOV      A, # 80H          ; 8255A 控制字
MOV      R0, # 93H         ; 控制字寄存器地址
MOVX     @R0, A             ; 输出控制字
MOV      R0, # 90H         ; A 口地址
LOOP:    MOV      A, # 01H   ; 从第一道工序起
LOOP1:   MOVX     @R0, A     ; 输出控制信号
        ACALL    DELAY      ; 延时 10s
        RL       A          ; 下一道工序
        JNB      ACC. 6, LOOP1 ; 6 道工序未完，继续
        SJMP     LOOP       ; 6 道工序完成，重新开始
DELAY:   MOV      22H, # 50  ; 延时子程序，10s
L1:      MOV      21H, # 100
L2:      MOV      20H, # 250
L3:      NOP
        NOP
        DJNZ     20H, L3
        DJNZ     21H, L2
        DJNZ     22H, L1
        RET

```

2. 对如图 8-5 所示的接口电路，要求编写有关程序段，使当波段开关接至位置 1 到 7 时数码管显示相应的数字 1 到 7，当开关接至位置 8 时，则退出程序。

解：由图知端口地址：A 口：7FFCH；B 口：7FFDH；C 口：7FFE H；控制口：7FFFH。8255A 设定为基本 I/O 工作方式。A 口为输出口，输出数字：1、2、……、7 的共阴极显示字型码，B 口为输入口，读入开关的状态，当开关拨至 1、2、……、8 中的某个位置时，该引脚即变为低电平。

程序如下：

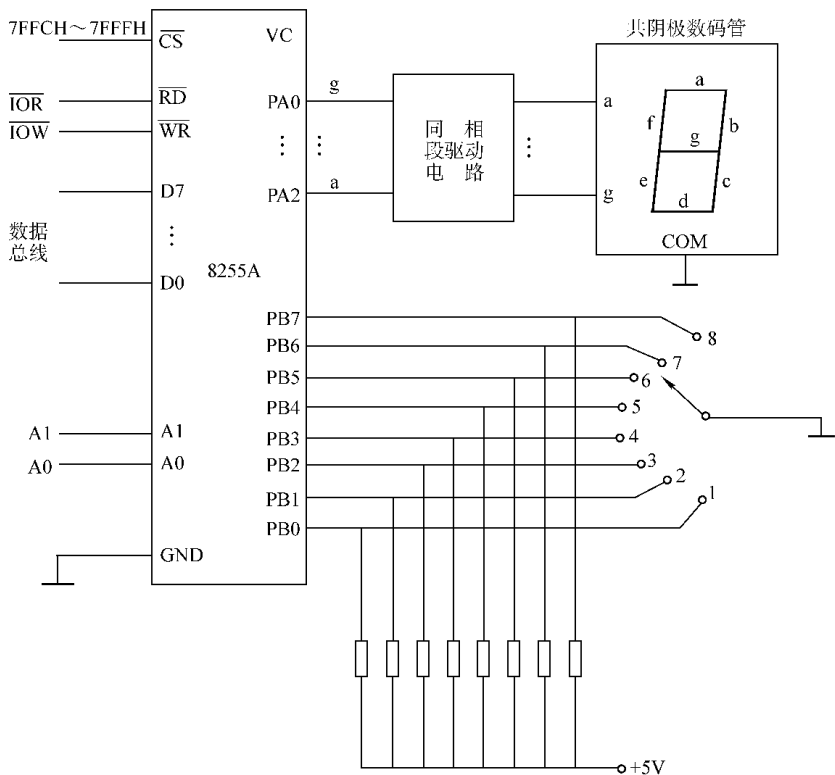


图 8-5 8255A 连接开关、LED 数码管应用电路

```
MAIN:  MOV     DPTR, # 7FFFH    ;控制字寄存器地址
        MOV     A, # 82H       ;控制字 82H
        MOVX    @DPTR, A       ;写入控制状态字, 初始化 8255A
START:  MOV     DPTR, # 7FFDH   ;B 口地址
        MOVX    A, @DPTR       ;由 B 口读开关状态进入 A
        CPL     A               ;将 A 中内容取反, 只有一位为高电平
        PUSH    A               ;保存 A 中内容
        XRL     A, # 80H       ;判断开关 8 是否闭合
        JZ      NEXT           ;A 内容为 80H 即键 8 闭合, 退出程序
        POP     A               ;非键 8 闭合时, 求显示字形码
        MOV     R0, # 00H      ;R0 用于记忆开关在哪一位
LOOP:   RRC     A               ;右移一位
        INC     R0              ;R0 加 1
        JNC     LOOP           ;进位标志为 0, 继续循环
        MOV     DPTR, TAB      ;表的首地址送 DPTR
        MOV     A, R0           ;要显示的数字值送 A
        DEC     A               ;查表时偏移量减 1
        MOVC    A, @A + DPTR    ;求出显示数值的字型码
```



```

MOV      DPTR, # 7FFCH
MOVX     @DPTR, A      ; 从 A 口输出显示字型码
SJMP     START         ; 返回
NEXT:    AJMP     NEXT  ; 程序暂停
TAB:     DB 06H,5BH,4FH,66H,6DH,7DH,07H ; 1,2,3,4,5,6,7 共阴字型码

```

五、8155 接口芯片应用

1. 从 8155 的 A 口输入数据，并进行判断。若 A 口读入数据不为 0，则将该数据存入 8155 的 RAM 中（从起始单元开始，数据总数不超过 256 个），同时从 B 口输出，并将 PC0 置 1；若 A 口读入数据为 0，则停止输入数据，同时将 PC0 清 0。试编写程序。电路如图 8-6 所示。

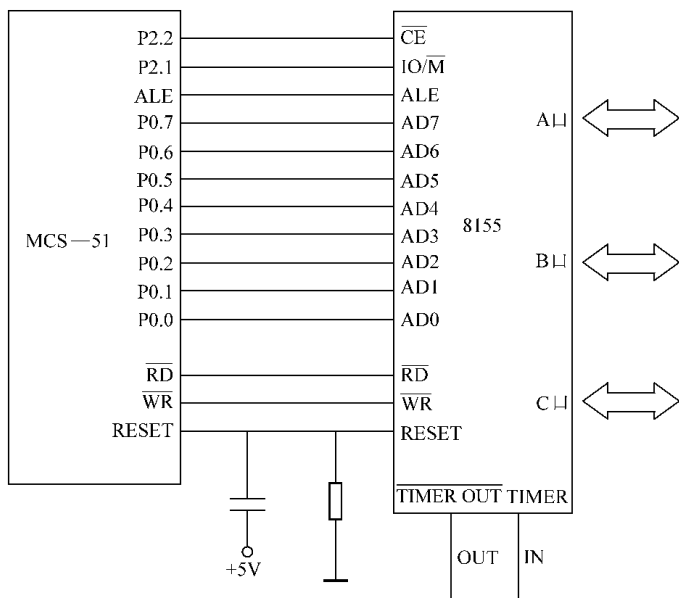


图 8-6 8155 接口电路

解：由图可知 $P2.2 = 0$ ，且 $P2.1 = 1$ 时，单片机可寻址端口，因此，控制寄存器的地址为：0200H，而 A、B、C 口的地址分别为：0201H、0202H、0203H。8155 的 RAM 的地址为：0000 ~ 00FFH。

程序如下：

```

MOV      DPTR, # 0200H  ; 指向命令口
MOV      A, # 06H       ; 命令字，基本 I/O, A 入, B、C 出
MOVX     @DPTR, A
MOV      R0, # 00H      ; 指向 8155 RAM 区首址
MOV      R1, # 00H      ; 数据总数为 256 个
LP1:     MOV      DPTR, # 0201H ; 指向 A 口
MOVX     A, @DPTR      ; 从 A 口读入数据

```

```

JZ      LP3          ; 为 0 则转 LP3
MOVX    @R0, A       ; 存入 RAM 单元
INC     R0           ; 指向下一单元
INC     DPTR         ; 指向 B 口
MOVX    @DPTR, A     ; 从 B 口输出
INC     DPTR         ; 指向 C 口
SETB    ACC. 0       ; 使 PC0 = 1
MOVX    @DPTR, A     ; 由 C 口输出
DJNZ    R1, LP1      ; 未完则反复
SJMP    LP2
LP3:     MOV     DPTR, # 0203H ; 指向 C 口
        MOVX    @DPTR, A     ; 回送, 即 PC0 = 0
LP2:     SJMP    $           ; 暂停

```

2. 单片机通过 8155 的 A 口以输出方式对顺序控制过程的 6 道工序进行控制。设每道工序运行时间为 10s。利用 C 口的某一位接收负脉冲形式的系统故障报警信号。当出现报警信号时, 单片机中断, 停止送出顺序控制信号, 同时利用一位 B 口线输出报警控制信号, 使警铃或指示灯发出指示。试画出连接电路图并编写实现程序。设 8155 命令/状态寄存器的地址为 00H, 端口 A、B、C 的地址分别为 01H, 02H 和 03H。

解: 根据题意, 接口连接电路如图 8-7 所示。题中强制性地将引脚 $\text{IO}/\overline{\text{M}}$ 接至高电平, 片选信号接地, 满足了题意中的端口地址要求, 这时片中 RAM 便不能使用。

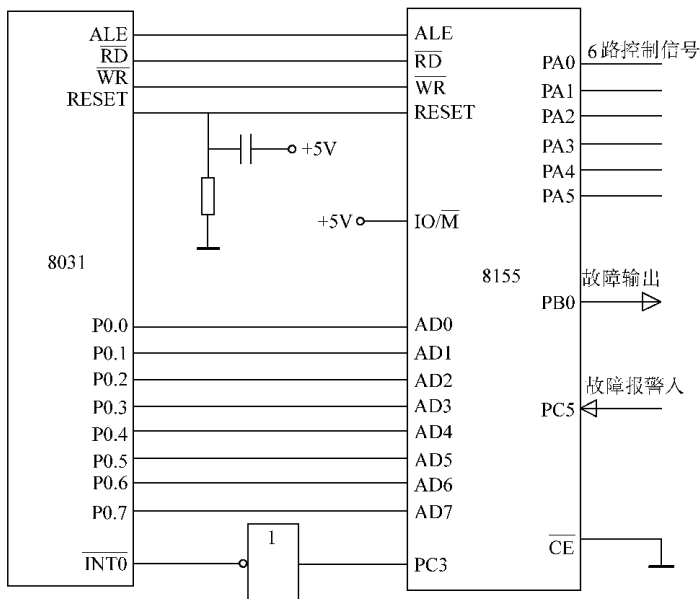


图 8-7 8155 接口电路

由题意要求, 8155 设定在选通工作方式 ALT4, A 口和 B 口均作输出口, C 口同时为 A 口及 B 口提供联络信号。这里 C 口主要为 B 口提供联络信号: PC3 ($\overline{\text{INTRB}}$) 和 PC5 ($\overline{\text{STBB}}$) 两路联络信号, 而 A 口不需要联络信号, 且设定 A 口禁止中断而 B 口允许中断。

编程思路为: 正常工作时, A 口周期性地输出顺序控制信号, B 口 (PB0) 输出数据 0 关闭报警灯或铃。当系统发生故障时, 向 PC5 提供负脉冲形式的报警信号, 该信号实际上是联络信号 $\overline{\text{STBB}}$, 它由低电平变为高电平时触发 $\overline{\text{INTRB}}$ (PC3), 使其由低电平变为高电平, 再通过反相器由高电平变为低电平, 向单片机申请中断。中断服务程序中禁止 A 口工作输出同时启动 B 口的报警灯或铃。

8155 的控制字为: D1D0 = 11 (A 口、B 口都为输出方式), D3D2 = 11 (C 口为 ALT4 方式), D5D4 = 01 (B 口允许中断, A 口不允许中断), D7D6 = 00, 即控制字为 1FH。

控制程序如下:

```

                ORG      0000H
                AJMP     MAIN
                ORG      0003H
                AJMP     ROUT
MAIN:          SETB     EA                ; 开所有中断
                SETB     EX0              ; 允许  $\overline{\text{INT0}}$  中断
                SETB     ITO              ; 令  $\overline{\text{INT0}}$  为边沿触发方式
                MOV      R0, 00H          ; 命令口地址
                MOV      A, # 1FH         ; 8155 控制字
                MOVX     @R0, A           ; 输出控制字
                MOV      R0, # 02H        ; B 口地址
                CLR      A                ; 关闭报警灯或铃
                MOVX     @R0, A           ; 报警输出清 0
                DEC      R0                ; A 口地址
LOOP:          MOV      A, # 01H          ; 第一路控制信号
LOOP1:         MOVX     @R0, A           ; 输出
                ACALL    DELAY            ; 延时 10s
                RL        A                ; 下一路控制信号
                JNB      ACC. 6, LOOP1
                SJMP     LOOP
HERE:          AJMP     HERE              ; 报警处理以后, 工作控制暂停延时子程序;
DELAY:         MOV      22H, # 50
L1:            MOV      21H, # 100
L2:            MOV      20H, # 250
L3:            NOP
                NOP
                DJNZ     20H, L3
                DJNZ     21H, L2

```

DJNZ 22H, L1

RET

中断服务程序：

```
ROUT:  MOV    R0, # 02H      ; 8155 的 B 口地址
        MOV    A, # 01H
        MOVX   @R0, A        ; 输出报警信号
        MOV    A, # 00H      ; 工作控制输出信号全为 0, 控制停止
        MOV    R0, # 00H     ; 8155 的 A 口地址
        MOVX   @R0, A        ; 输出
        MOV    DPTR, # HERE  ; 中断返回时不返回到原断点处
        MOV    A, DPL        ; 强制将程序返回到主程序 HERE 处
        PUSH   ACC           ; 将 HERE 地址压栈
        MOV    A, DPH
        PUSH   ACC
        RETI
```

第九章 单片机与 A/D、D/A 转换器的接口

第一节 学习要求与内容概要

一、学习要求

- (1) 了解 A/D 转换器的分类与性能指标，熟悉 ADC0809 芯片的结构、功能，掌握单片机接口方法与程序设计方法。
- (2) 了解 D/A 转换器的分类与性能指标，熟悉 DAC0832 芯片的结构、功能，掌握单片机接口方法与程序设计方法。

二、A/D 转换器及接口学习内容概要

1. A/D 转换器分类及性能指标

A/D 转换器是将模拟量信号转换为数字量信号的接口电路，按工作原理分为：逐位比较型、并行比较型、双积分式、电压频率（V/F）型及计数比较型等。

A/D 转换器的主要性能技术指标有：分辨率、转换时间、转换精度等。

2. ADC0809 转换器及其接口

ADC0809 是逐次逼近型 8 位 A/D 转换芯片。图 9-1 为 ADC0809 内部逻辑结构及引脚图。

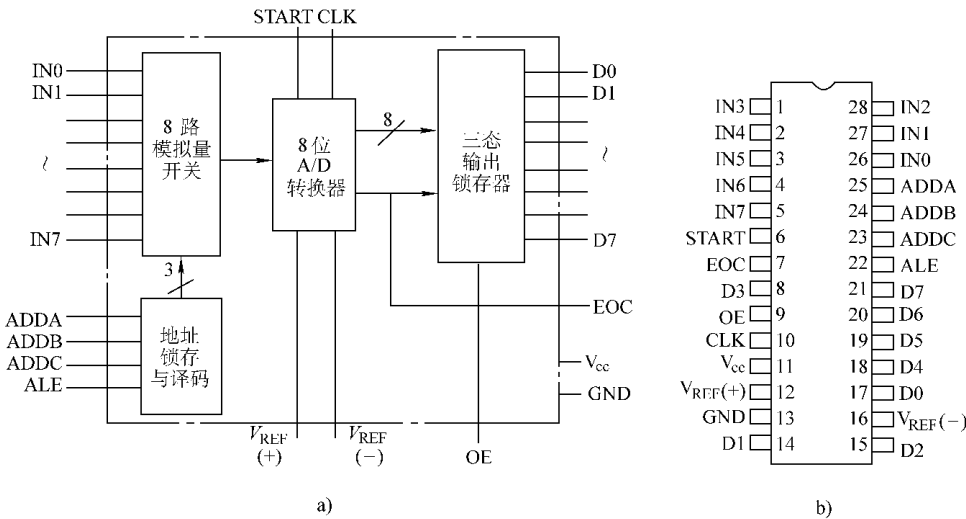


图 9-1 ADC0809 转换器结构及引脚

a) ADC0809 内部逻辑结构 b) ADC0809 引脚

ADC0809 芯片采用多路模拟开关，允许 8 路模拟量分时输入，共享一个 A/D 转换器完成转换。模拟输入通道选择地址及转换结果均有锁存译码器。主要引脚及功能如下：

IN0 ~ IN7：8 个模拟通道输入端。

D0 ~ D7：8 位数字量转换结果输出端。

ADDA、ADDB、ADDC：模拟通道选择地址。

ALE：地址锁存信号输入端。

START：启动转换信号输入端，加上正脉冲后 A/D 开始转换。

EOC：转换结束输出信号。转换开始后 EOC 信号变低；转换结束时 EOC 信号返回高电平。

OE：输出允许控制端，高电平时打开三态输出锁存器，输出转换结果。

CLK：芯片工作时钟信号。

$V_{REF(+)}$ 和 $V_{REF(-)}$ ：芯片工作参考电压输入端。

图 9-2 为 ADC0809 与 8051 中断方式连接的电路图。

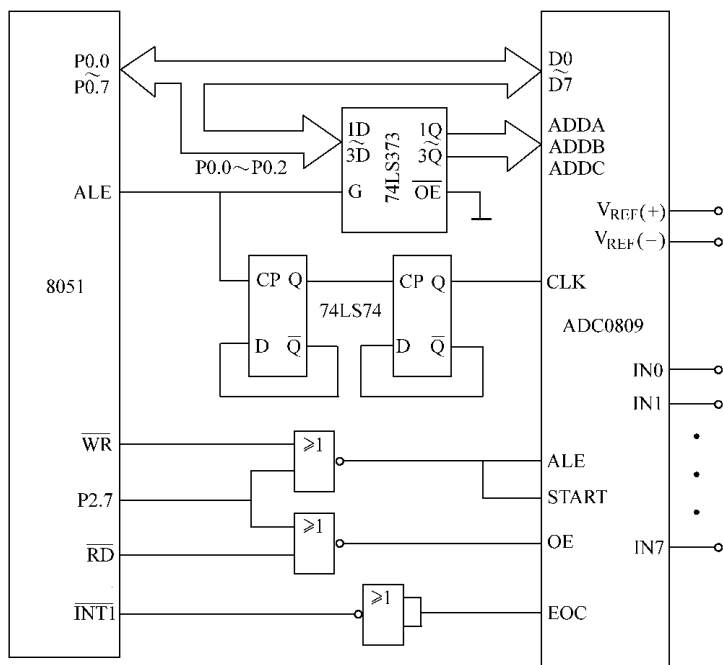


图 9-2 ADC0809 与 8051 连接电路图

对于 ADC 的使用，首先启动转换，然后检测有关引脚的状态读取转换结果。对于图 9-2 所示的电路，启动 ADC0809 转换的命令是：

```
MOV DPTR, # PORT          ; PORT 要转换的某一路模拟通道的口地址
MOVX @DPTR, A              ; 执行该指令时，P2.7 与  $\overline{WR}$  合成，在 START 端得到正脉冲转换启动信号
```

读取转换结果的指令为：

```
MOV DPTR, # PORT          ; PORT 要转换的某一路模拟信号的口地址
MOVX A, @DPTR              ; 执行该指令时 P2.7 与  $\overline{RD}$  合成，使 OE 高电平
```

三、D/A 转换器及接口学习内容概要

1. D / A 转换器分类及性能指标

D/A 转换器是将数字量信号转换为模拟量信号的接口电路，按照与 CPU 的接口方式分为并行和串行两大类，按生产工艺分为双极型和 MOS 型等。

D/A 转换器主要性能技术指标有：分辨率、建立时间、转换精度等。

2. DAC0832 转换器及其接口

DAC0832 是 8 位单片 D/A 转换器，主要由两个 8 位寄存器和一个 8 位 D/A 转换器组成。其引脚图和逻辑框图如图 9-3 所示。

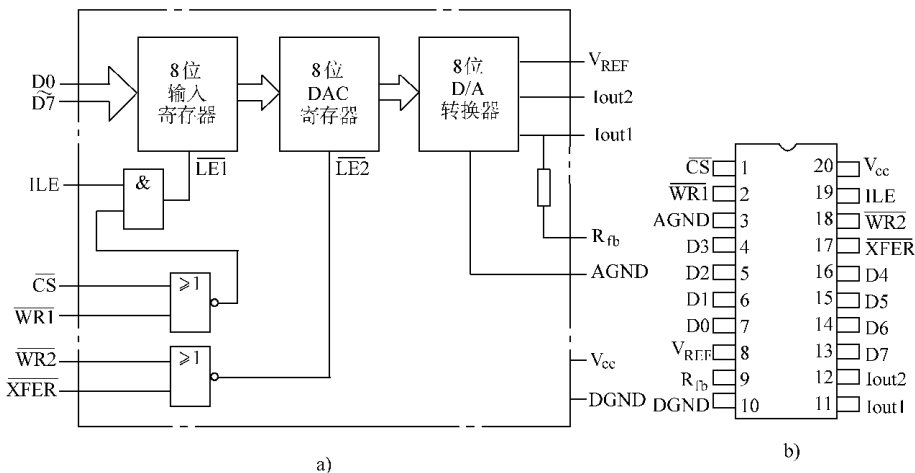


图 9-3 DAC0832 转换器结构与引脚

a) DAC0832 内部逻辑结构 b) DAC0832 引脚

DAC0832 主要引脚及功能：

D0 ~ D7：转换数字量数据输入线；

ILE：数据锁存允许信号输入线，高电平有效；

$\overline{\text{CS}}$ ：输入寄存器芯片选择信号输入线，低电平有效；

$\overline{\text{WR1}}$ ：输入寄存器“写”选通信号输入线，低电平有效；

$\overline{\text{XFER}}$ ：数据转移控制信号输入线，低电平有效；

$\overline{\text{WR2}}$ ：DAC 寄存器“写”选通信号输入线；

V_{REF} ：基准电压输入线，可以超出 $\pm 10\text{V}$ ；

R_{FB} ：芯片内反馈电阻信号线；

Iout1 和 Iout2：转换结果电流输出线，Iout1 和 Iout2 的和为常数，而 Iout1 与转换数字量成线性；

V_{CC} ：工作电源正端，范围 $+5 \sim +15\text{V}$ ，最佳为 $+15\text{V}$ ；

DGND：芯片数字电路地；

AGND：芯片模拟电路地。

DAC0832 转换器与 8051 的接口，可以采用片内两个寄存器均不起锁存作用的直通连接方式，也可以采用片内两个寄存器中只有一个起锁存作用的单缓冲连接方式，或者采用片内两个寄存器均起锁存作用的双缓冲连接方式。

直接把要转换的数据送至 DAC 端口，DAC 就进行转换，进而在输出端得到模拟量信号，指令为：

```

MOV    DPTR, # PORT    ; 设置 D/A 口地址
MOV    A, # data        ; 转换数字量
MOVX   @DPTR, A        ; 输出端得到对应于 A 内容的模拟量

```

第二节 原教材习题及解答

10-1 A/D 转换器的作用是什么？常用器件分类有哪些？主要性能技术指标有哪些？

答：A/D 转换器将采集到的模拟量信号转化为计算机能够处理的数字量信号。

目前常用的 A/D 转换器按工作原理分为：逐位比较型、并行比较型、双积分式、电压频率（V/F）型及计数比较型等。

A/D 转换器的主要性能技术指标有：分辨率、转换时间和转换精度等。

10-2 D/A 转换器的作用是什么？主要性能技术指标有哪些？

答：D/A 转换器的作用是将数字量信号转化为模拟量信号。

D/A 转换器主要性能技术指标有：分辨率、建立时间和转换精度等。

10-5 利用原教材图 10-2 所示的 ADC0809 与单片机的接口电路，请编写由通道 2 采集一组数据的处理程序，要求每组采样 4 个数据，每个数据间隔 1ms，并存于 8051 内部 RAM 70H ~ 73H 字节单元中。

解：这里采用中断方式启动转换、读取结果。程序如下：

```

ORG      0000H
AJMP     MAIN                ; 复位后转 MAIN 执行
ORG      0013H              ; 外部中断 1 服务程序入口地址
AJMP     ADCIN2             ; CPU 响应中断时转 ADCIN2 处执行
MAIN:    MOV     R0, # 70H   ; 数据暂存区首址
         MOV     R2, # 04H   ; 每组 4 个数据
         SETB    IT1        ; 中断边沿触发方式
         SETB    EA         ; 开放总中断
         SETB    EX1        ; 开放外部中断 1
         MOV     DPTR, # 7FFAH ; 指向 ADC0809 模拟信道 2
LOOP:    MOVX    @DPTR, A    ; 启动 A/D 第一次转换
HERE:    SJMP     HERE      ; 等待中断
ADCIN2:  MOVX    A, @DPTR    ; 中断服务子程序，读取转换结果
         MOV     @R0, A      ; 转换后数据存入内存
         INC     R0          ; 更新暂存单元
         ACALL   DELAY       ; 间隔 1ms
         DJNZ    R2, COMMON  ; 4 个数据没有采样完，继续
         CLR     EA
         RETI
COMMON:  MOV     DPTR, # 7FFAH ; 指向 ADC0809 模拟信道 2 地址
         MOVX    @DPTR, A    ; 再次启动 A/D 转换

```



```

                RETI                                ; 中断返回
DELAY:  MOV     R7, # 02H                          ; 延时 1ms 子程序
D1:     MOV     R6, # 0F9H
D2:     DJNZ    R6, D2
        DJNZ    R7, D1
        RET

```

10-6 利用原教材图 10-6 所示的 DAC0832 与单片机的接口电路, 请编写由 DAC0832 产生特殊矩形波程序, 要求矩形波周期为 100ms, 占空比为 1:4, 波形高电平电压为 3.5V, 低电平为 0.8V (设 $V_{REF} = -5V$)。

解: 设矩形波高电平电压 3.5V 对应的数字量为 X, 低电平电压 0.8V 对应的数字量为 Y, 则

$$\frac{3.5}{5} = \frac{X}{255}, \frac{0.8}{5} = \frac{Y}{255}, \text{进而得 } X = 0B3H, Y = 29H$$

根据题意编写程序如下:

```

                ORG     0000H
                AJMP    MAIN
MAIN:  MOV     DPTR, # 7FFFH    ; 设置 D/A 口地址
LOOP:  MOV     A, # 0B3H       ; 将高电平 3.5V 对应的数字量送给 A
        MOVX   @DPTR, A       ; D/A 输出相应模拟量
        ACALL  DLY20          ; 延时 20ms
        MOV    A, # 29H       ; 将低电平 0.8V 对应的数字量送给 A
        MOVX   @DPTR, A       ; D/A 输出相应模拟量
        ACALL  DLY80          ; 延时 80ms
        AJMP   LOOP           ; 循环
DLY20:  MOV     R7, # 28H      ; 延时 20ms 子程序
D1:     MOV     R6, # 0F9H
D2:     DJNZ    R6, D2
        DJNZ    R7, D1
        RET
DLY80:  MOV     R7, # 0A0H     ; 延时 80ms 子程序
D1:     MOV     R6, # 0F9H
D2:     DJNZ    R6, D2
        DJNZ    R7, D1
        RET

```

第三节 增选习题及解答

一、填空题

1. A/D 转换器将____转化为____, D/A 转换器将____转化为____。

2. ADC0809 是____位____型 A/D 转换器芯片, 内部有____路模拟开关。
3. A/D 转换器的三个重要指标分别是____、____和____。
4. 一个 10V 的 8 位 D/A 转换器的分辨率约为____。
5. DAC0832 转换器有三种接口工作方式, 分别是____、____和____。

填空题参考答案:

1. 模拟量、数字量, 数字量、模拟量
2. 8, 逐次逼近, 8
3. 分辨率、转换时间、转换精度
4. $10/256 = 39\text{mV}$
5. 直通式工作方式、单缓冲工作方式、双缓冲工作方式

二、选择题

1. 并行比较型 A/D 转换器, 其转换速度较____, 精度较____。
A. 快 高 B. 慢 高 C. 快 低 D. 慢 低
2. ADC0809 可用____引脚经中断逻辑向 CPU 申请中断。
A. OE B. START C. EOC D. ALE
3. 下面选项中哪一项不是 D/A 转换器的技术指标____。
A. 分辨率 B. 转换时间 C. 转换精度 D. 建立时间
4. ADC0809 启动 A/D 转换的方式是____。
A. 高电平 B. 低电平 C. 正脉冲 D. 负脉冲

选择题参考答案:

1. C
2. C
3. B
4. C

三、简答题

设某 DAC 完成二进制 14 位转换, 满量程模拟输出电压为 10V, 试问它的分辨率和转换精度各为多少?

答: 分辨率是指 D/A 转换器能分辨的最小输出模拟增量, 它决定于输入数字量的二进制位数。满量程为 10V 的 14 位 DAC 芯片的分辨率为 $10 / 2^{14} = 610\mu\text{V}$ 。

转换精度是指满量程时 DAC 的实际模拟输出值和理论值的接近程度。通常情况下, DAC 的转换精度为分辨率之半, 即 10V 的 14 位芯片的转换精度为 $305\mu\text{V}$ 。

四、应用

1. 利用原教材图 10-7 的 DAC0832 与 8051 的接口电路, 能够产生多种波形。试编写程序产生三角波。

解: 这里通过改变程序中的两个数值 DATA1 和 DATA2 来改变三角波的波形。但要求 D/A 转换器的建立时间足够小才能利用下面的编程思路。

```
ORG      0000H
MOV      DPTR, # 7FFFH      ; 设置 D/A 口地址
MOV      A, # DATA1        ; 置初值
LOOP1:   MOVX   @DPTR, A      ; 输入, D/A 转换
INC      A                  ; 输出增大
CJNE     A, # DATA2, LOOP1   ; 未达到最大值则继续增加数字量
```

```

LOOP2:  MOVX    @DPTR, A           ; 输入, D/A 转换
        DEC     A                 ; 输出减小
        CJNE    A, # DATA1, LOOP2 ; 未达到初值则继续减小数字量
        AJMP    LOOP1            ; 减到最低, 循环
        END

```

2. 利用原教材图 10-2 所示的接口电路, 请编写程序采集通道 IN0 ~ IN7 上的模拟电压, 每 20ms 采集一个通道, 并存入单片机内部 RAM 20H 开始的数据区中。

解: 编程如下:

```

        ORG     0000H
        AJMP    MAIN
        ORG     0013H           ; 外部中断服务程序入口地址
        AJMP    INT_1

MAIN:    MOV     R0, # 20H       ; 保存数据区首址
        SETB    EA              ; 开总中断
        SETB    IT1             ; 中断边沿触发方式
        SETB    EX1             ; 开外部中断 1
        MOV     R7, # 08H       ; 8 路计数区初值
        MOV     DPTR, # 7FF8H   ; 指向 ADC0809 的 IN0 通道
        MOVX    @DPTR, A        ; 启动 A/D 转换
HERE:    SJMP    HERE           ; 等待中断
INT_1:   MOVX    A, @DPTR        ; 读取转换结果
        MOV     @R0, A
        INC     R0              ; 指向下一存储单元
        INC     DPTR            ; 模拟通道号增 1
        DJNZ    R7, GOON        ; 未完, 继续
        RETI

GOON:    ACALL   DELAY           ; 调用 20ms 延时程序
        MOVX    @DPTR, A        ; 启动下一路 A/D 转换
        RETI                    ; 中断返回

DELAY:   MOV     R5, # 5AH
LOOP2:   MOV     R4, # 1BH
LOOP1:   NOP
        NOP
        DJNZ    R4, LOOP1
        DJNZ    R5, LOOP2
        RET
        END

```

第十章 MCS—51 单片机应用 系统设计方法与举例

第一节 学习要求与内容概要

一、学习要求

- (1) 了解 MCS—51 单片机应用系统的设计方法。
- (2) 熟悉利用主要接口芯片进行 MCS—51 单片机综合应用扩展的电路连接方法。
- (3) 掌握 MCS—51 单片机应用系统编制完整控制程序的基本方法。

二、MCS—51 单片机综合应用学习内容概要

1. MCS—51 单片机应用系统设计方法

单片机应用系统设计的步骤主要体现为以下几个方面：首先根据系统所要求完成的功能确定系统的总体设计方案，接着进行相应的硬件系统设计与软件程序编制，然后在硬件电路和软件程序的相互配合下完成系统的仿真与调试，通过调试发现系统硬件与软件存在的问题，解决问题直至完全调试成功。

单片机应用系统硬件系统设计基本上包括单片机系统、信号检测模块、信号控制模块、人机对话功能模块及电源模块，有的还包括通信功能模块等，其中，单片机系统包括单片机基本系统部分和扩展部分。

软件系统设计的过程为：首先分析问题，然后确定解决问题的具体算法，再根据算法流程编写程序，最后结合硬件联合调试程序，直至调试成功。

系统调试是单片机应用系统设计完成的主要任务。它包括前期的硬件部分调试和其后的应用程序调试。只有排除了硬件错误的系统才可以为应用程序调试成功创造条件。排除应用系统硬件错误除了单纯的一些硬件检测方法以外，还要借助一定的软件才能发现一些隐蔽的硬件错误。比如，一些逻辑器件的好坏可以通过给输入引脚施加一定条件的激励，然后根据测量、判断输出结果发现硬件问题，但是，一些复杂芯片，如可编程器件、随机存储器等存在的问题则需要指令程序写入数据再读出数据予以分析对比才可以发现问题。对于复杂软件程序的调试，可以借助仿真器完成。仿真器在 PC 机的支持下实现单片机芯片的功能，通过对程序运行结果的对话式分析并修改程序，直至得到正确的应用程序。正确的程序固化到程序存储器芯片或者单片机芯片内部程序存储器中，在没有错误的电路板上，插上所有芯片即完成了单片机应用系统设计。

2. MCS—51 单片机利用主要接口芯片进行综合应用扩展

MCS—51 单片机只有配置上外部设备才能构成具有实际意义的单片机应用系统。单片机连接外部设备是通过接口电路完成的。本教材中介绍了可编程 I/O 接口电路芯片，如 8155、8255A 等，A/D 转换电路接口芯片，如 ADC0809，D/A 转换电路接口芯片，如 DAC0832，以及存储器扩展芯片等，根据应用系统的要求，选择性地使用接口芯片可以构建一个单片机综

合性应用系统。

对于利用这些芯片进行系统接口扩展，应该注意以下两点问题：

(1) 充分了解接口芯片的结构和功能，掌握各种引脚的功能和特点，这样才能保证引脚使用的正确性。接口芯片和单片机之间的连接主要是通过三大总线完成的。8 位数据总线直接和 P0 口连接即可，但是如果连接的芯片数目多于 8 个，P0 口的输出就必须添加双向驱动器，如 74LS245 进行总线驱动；对于地址总线的连接，如果需要 P0 口输出的低 8 位地址线，有的芯片内部含有地址锁存器，这种芯片就不需要单片机外接低 8 位地址锁存器对输出的低 8 位地址进行锁存；对于接口电路需要的工作控制信号以及状态输出信号，一定了解它们的有效电平状态，有的是高电平有效，有的是低电平有效，还有的是边沿触发（上升沿或下降沿），根据它们的特点设计合适的连接电路。

(2) 合理地编排综合扩展各芯片的地址。由于使用芯片较多或者扩展存储器的容量较大，这时往往借助地址译码器，如 74LS138 完成对扩展连接各芯片的地址进行统一安排。这时应特别注意，一是对于片内没有程序存储器的单片机芯片，如 8031 等它们的外部扩展的程序存储器的地址一定是从 0000H 单元值开始；二是进行 I/O 接口扩展，接口芯片的端口地址和外扩 RAM 同一编址，一定避免发生操作冲突，即端口一定不能占用 RAM 单元的地址，但是数据输出端口和另一个数据输入端口可以具有相同的端口地址。

3. MCS—51 单片机应用系统控制程序完整编制的各种基本方法

合理运用程序编制的各种基本方法，使程序明了、高效，符合结构化程序设计的规范。比如，定时实现可以采用软件定时也可以采用硬件定时，如果应用系统程序复杂、CPU 管理任务较多，应该选用硬件定时；对于一些事件包括设备状态的检测，可以采用主程序循环查询法也可以采用中断捕捉法。中断捕捉法具有快捷、优先的显著特点。

为了便于程序的阅读和调试，MCS—51 单片机应用系统程序编制应该采用模块化设计，就是把完成一定功能部分的程序设计成为一个子程序模块，如 A/D 转换模块、D/A 转换模块、8255A 开关量处理模块等；另外，各程序的定位一定合适，比如，不同的中断服务程序具有不同的中断服务程序入口，且要作开中断、关中断、设置中断触发方式等有关中断的命令设置。

为了适应较为复杂的任务管理，程序编制尽量使用事件状态标志位、标志字节。通过伪指令对各种标志进行定义，使程序易于编写和阅读。

第二节 原教材习题及解答

12-3 试设计一个十字路口交通灯控制系统，要求如下：

(1) 观察一个装有交通灯的十字路口的车辆运行规律，根据东西和南北方向红、绿、黄灯变化的规律，画出运行规律流程图。

(2) 设计由 8031 单片机控制交通灯的硬件电路图；设计系统运行程序。

解：假设十字路口东西方向和南北方向的车流量相当，可以使两个方向的通行时间一样长，即东西方向通行时：东西方向绿灯亮 25s，然后黄灯亮 5s，南北方向红灯亮 30s；之后南北方向通行时：南北方向绿灯亮 25s，黄灯亮 5s，同时东西方向红灯亮 30s，依此循环。以东西方向绿灯先亮为起始，运行规律如图 10-1 所示。

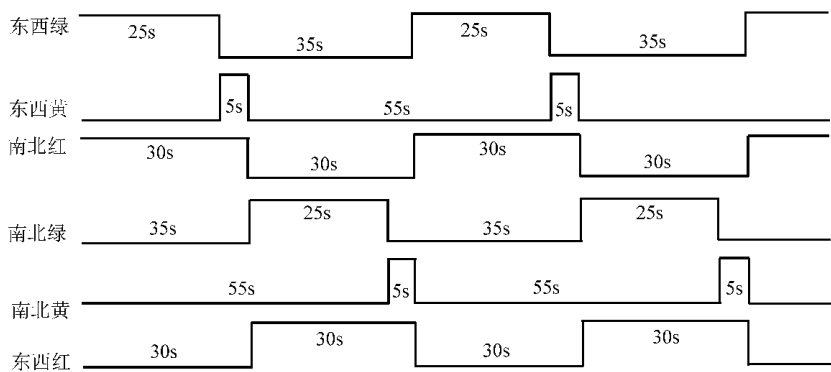


图 10-1 交通灯运行时序图

为简便起见，此处用发光二极管来代替交通灯，硬件连接如图 10-2 所示（省略了程序存储器扩展部分）。按下 S1 键工作开始，按下 S2 键则停止工作，P1 口某一位为高电平时，对应控制的交通灯亮。控制输出代码及时间如表 10-1 所示。

程序如下：

START

BIT 00H

； S1 键按下置位， S2 键松开时清 0

ORG

0000H

AJMP

MAIN

ORG

0003H

AJMP

INTRP

； 外部中断 0 入口，交通灯开始工作

ORG

0013H

AJMP

EXIT

； 外部中断 1 入口，交通灯停止工作

MAIN：

ANL

P1， # 00H

； P1 口输出 0，灯均不亮

SETB

EX0

； 允许外部中断 0 中断

SETB

IT0

； 外部中断 0 为边沿触发方式

SETB

EX1

； 允许外部中断 1 中断

SETB

IT1

； 外部中断 1 为边沿触发方式

SETB

EA

； CPU 开中断

表 10-1 交通灯控制代码及时间

P1.7	P1.6	P1.5 东西绿	P1.4 东西黄	P1.3 南北红	P1.2 南北绿	P1.1 南北黄	P1.0 东西红	输出 代码	工作 时间/s
0	0	1	0	1	0	0	0	28H	25
0	0	0	1	1	0	0	0	18H	5
0		0	0	0	1	0	1	05H	25
0		0	0	0	0	1	1	03H	5

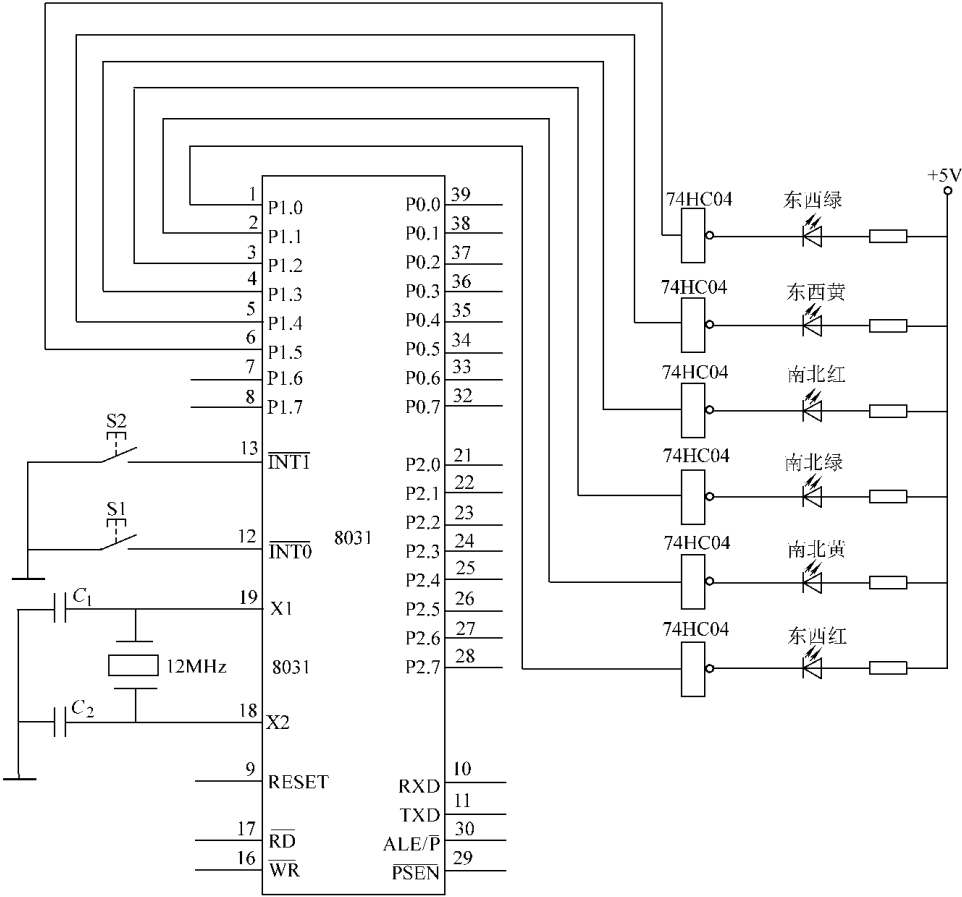


图 10-2 8031 控制信号灯接口电路

```
CLR      START      ; 清 START 标志位
LOOP :   JNB      START, $      ; SYART= “0” 时，等待
        MOV      90H, # 28H      ; 东西绿灯、南北红灯亮 25s
        MOV      R0, # 05H
L1 :     ACALL    DELAY5          ; 5s 延时子程序
        DJNZ     R0, L1
        MOV      90H, # 18H      ; 东西黄灯、南北红灯亮 5s
        ACALL    DELAY5
        MOV      90H, # 05H      ; 东西红灯、南北绿灯亮 25s
        MOV      R0, # 05H
L2 :     ACALL    DELAY5
        DJNZ     R0, L2
        MOV      90H, # 03H      ; 东西红灯、南北黄灯亮 5s
        ACALL    DELAY5
        AJMP     LOOP
DELAY5 : MOV      R1, # 50        ; 延时 5s
```

```

LOOP1: MOV    R2, # 200
LOOP2: MOV    R3, # 248
        DJNZ   R3, $
        DJNZ   R2, LOOP2
        DJNZ   R1, LOOP1
        RET
INTRP: SETB   START                ; S1 键按下时, 置位 START
        RETI
EXIT:  ANL    P1, # 00H            ; S2 键按下时, 所有灯灭
        CLR    START              ; START 清 0
        POP    ACC
        POP    ACC
        MOV    DPTR, # LOOP        ; LOOP 地址压栈
        PUSH   DPL                ; 中断强制返回至 LOOP 处
        PUSH   DPH
        RETI
        END

```

12-4 试设计步进电动机起、停、调速控制系统, 要求如下:

- (1) 分别用开关 S1、S2 控制步进电动机的起动和停止, S3、S4 控制调速。
- (2) 用 8031 单片机设计出单三拍电动机的接口电路。
- (3) 编写控制程序。

解: 8031 单片机与单三拍电动机接口电路如图 10-3 所示:

8031 单片机利用 P1.0、P1.1、P1.2 控制步进电动机具体连接电路参见原教材图 12-6。这里, S1: 起动, S2: 停止, S3: 加速, S4: 减速。

利用定时器中断控制步进电动机的运行速度。步进电动机正常运行时定时器初值为: DA_nor₀。每按一次加速键 S3, 定时器初值递加 ΔDA ; 每按一次减速键 S4, 定时器初值递减 ΔDA 。

程序如下:

```

                ORG    0000H
                AJMP   MAIN
                ORG    001BH
                AJMP   TO_int
MAIN:          MOV    SP, # 60H        ; 设置堆栈指针
                MOV    TMOD, # 10H    ; T1 方式 1
                MOV    TL1, # DA_norl ; 正常运行定时器初值低位
                MOV    TH1, # DA_norh ; 正常运行定时器初值高位
                SETB   EA              ; 开中断
                SETB   ET1            ; 允许定时器 0 中断

```

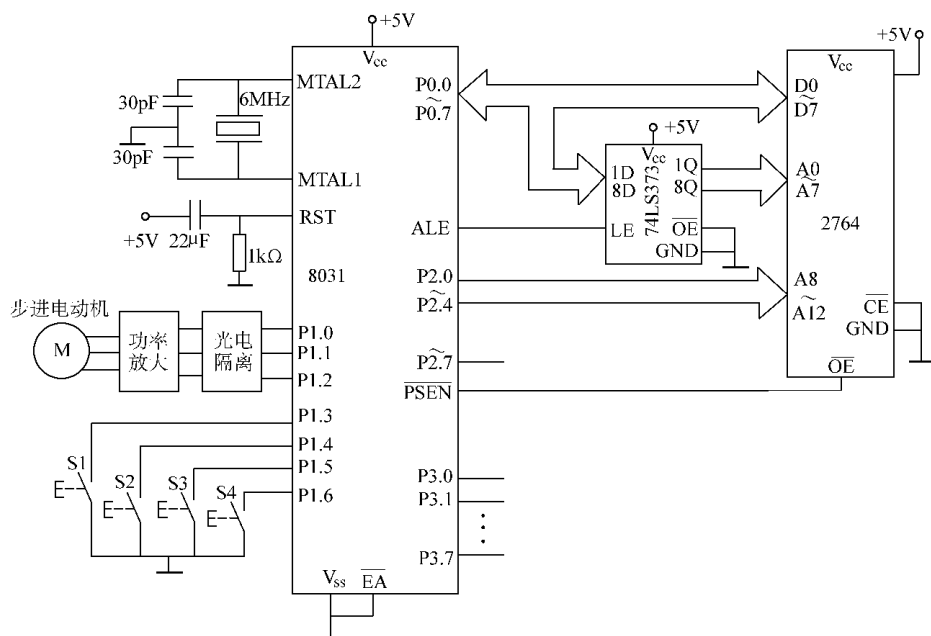



图 10-3 89C51 控制步进电动机接口电路

ANL P1, # 0F8H ; P1.0 = P1.1 = P1.2 = 0, 步进电动机停止

MOV R0, # 00H ; 查表偏移量放 R0

WAIT_ START: JB P1.3, \$; S1 键没按下, 等待步进电动机启动

MOV R7, # DA_ norh ; R7、R6 存放正常运行定时器初值

MOV R6, # DA_ norl

SETB TR0 ; 启动定时器, 步进电动机运行

LOOP: JB P1.4, CON_ 1 ; S3 键没按下, 查 S4 键

MOV A, R6 ; S3 按下, 定时器初值增加 ΔDA

ADD A, # ΔDA

MOV R6, A

MOV A, R7

ADDC A, # 0

MOV R7, A

CON_ 1: JB P1.5, WAIT_ STOP

MOV A, R6 ; S4 按下, 定时器初值减小 ΔDA

CLR C

SUBB A, # ΔDA

MOV R6, A

MOV A, R7

SUBB A, # 0

MOV R7, A

WAIT_ STOP: JB P1.6, LOOP ; S2 键没有按下, 继续监测 S3、S4

```

        ANL    P1, # 0F8H           ; S2 键按下, 停机
        CLR    TR0
        SJMP   WAIT_ START         ; 停机后, 等待继续
T0_int : MOV    TL1, R6
        MOV    TH1, R7
        MOV    DPTR, # TAB         ; 正转控制字表首地址
        MOV    A, R0
        MOVC   A, @A + DPTR        ; 查表取控制字代码
        MOV    P1, A               ; 送出控制代码
        INC    R0                  ; 查表单元上升
        CJNE   R0, # 03H, RET_I
        MOV    R0, # 00H
RET_I : RETI                        ; 返回
TAB :   DB     01H, 02H, 04H
        END

```

第三节 增选习题及解答

1. 图 10-4 是利用单片机通过 BCD 码译码/驱动芯片 7447 (U₂、U₃) 驱动两个共阳极 LED 数码管实现 99s 马表电路图。图中由 T₁ 引脚连接的按键: 控制/停止/归零显示时间值, 即: 开始时两位数码管显示“00”, 第一次按键后就开始计时; 第二次按键后计时停止; 第三次按键后计时归零。试编写功能实现程序。

解: 要实现定时可以采用软件定时也可以采用硬件定时, 这里利用定时器定时/中断处理进行编程。程序如下:

```

        ORG    0000H
        AJMP   MAIN
        ORG    000BH           ; 定时器 T0 中断服务程序
        MOV    TH0, # 0D8H     ; 10ms 定时初值重新赋值
        MOV    TL0, # 0F0H
        DJNZ   R1, RETRN       ; 100 次定时中断才达到 1s 定时
        MOV    R1, # 100       ; 重赋初值, 中断次数
        MOV    A, R0           ; 计时秒值单元
        ADD    A, # 01H        ; 计时累加 1s
        DA     A               ; 作十进位调整
        MOV    R0, A           ; 存计时值
        MOV    P2, A           ; 输出显示当前定时值
RETRN : RETI
MAIN :   MOV    P2, # 00H       ; 开始时 P2 输出 00H, 马表显示“00”
        MOV    TMOD, # 01H     ; T0 为 16 位定时方式

```

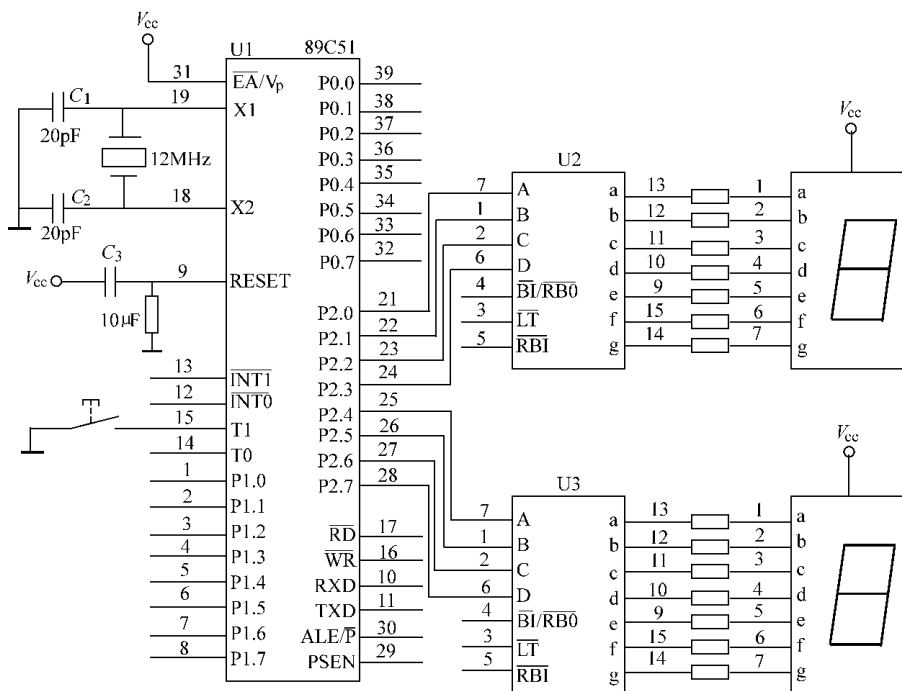


图 10-4 99s 马表电路图

MOV	TH0 , # 0D8H	; 10ms 定时初值
MOV	TL0 , # 0F0H	
SETB	EA	; 开中断总允许
SETB	ET0	; 允许 T0 溢出中断
MOV	R1, # 100	; T0 定时中断 100 次即为 1s
MOV	R0, # 00	; 定时时间保存单元, 初值为 0
JB	P3.5, \$	; T1 = 1, 没有按键, 继续等待
SETB	TR0	; 第一次按键, 定时启动
ACALL	DELAY	; 调消除抖动延时程序
JNB	P3.5, \$	; T1 = 0, 按键没有释放继续等待
JB	P3.5, \$	; T1 = 1, 没有第二次按键, 继续等待
CLR	TR0	; 第二次按键后, 定时停止
ACALL	DELAY	; 调消除抖动延时程序
JNB	P3.5, \$	; T1 = 0, 按键没有释放继续等待
JB	P3.5, \$	; T1 = 1, 没有按键, 继续等待
MOV	P2, # 00	; 第三次按键后, 显示清零
ACALL	DELAY	; 调消除抖动延时程序
JNB	P3.5, \$	; T1 = 0, 按键没有释放, 继续等待
AJMP	MAIN	; 重新开始
MOV	R6, # 60	; 30ms

```

D1:      MOV    R7, # 248
          DJNZ   R7, $
          DJNZ   R6, D1
          RET
          END

```

2. 图 10-5 示意的 MCS-51 系统扩展了一片 RAM6264 和一片 DAC0832, 分析图中 RAM 地址范围以及 D/A 转换器通道地址。选用 6264 中的 8 个连续单元作为 D/A 转换的数字量存储区, 每隔 0.1s 从 RAM 区顺序取出一个数字量送入 DAC0832 输出, 8 个值作为一个周期, 试设计完成连续输出该周期信号的子程序。

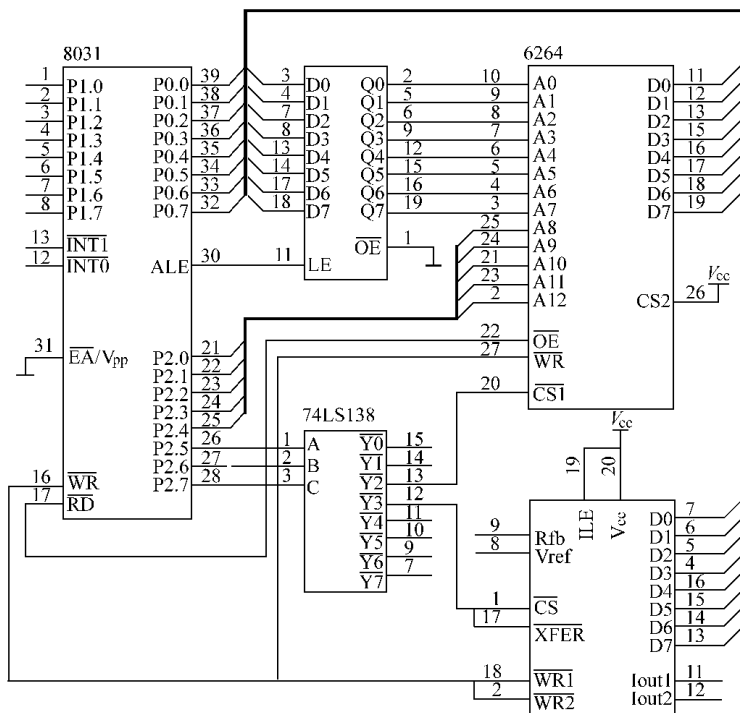


图 10-5 MCS-51 应用系统电路图

解: RAM 芯片 6264 被选中时, 74LS138 的 $\overline{Y_2} = 0$, 故 6264 的地址范围为: 4000H ~ 5FFFH ($P2.7 = 0$, $P2.6 = 1$, $P2.5 = 0$); 而 DAC0832 被选中时, 74LS138 的 $\overline{Y_3} = 0$, 故 DAC0832 的口地址为: 6000H ($P2.7 = 0$, $P2.6 = 1$, $P2.5 = 1$)。

根据题意编制程序如下:

```

DAC0832: MOV    R7, # 08H           ; 输出 8 个数据
          MOV    DPTR, # 4000H       ; 6264 存 8 个数据的首地址
LOOP:    MOVX   A, @DPTR             ; 从片外 RAM 存储区读取数据
          INC    DPTR
          PUSH   DPL                  ; 保护地址
          PUSH   DPH
          MOV    DPTR, # 6000H       ; DA 转换器地址送 DPTR

```

```

MOVX    @DPTR , A           ; 送数据至 DA 转换器
ACALL   TIM                 ; 调用延时子程序, 等待转换结束
POP     DPH                 ; 恢复存储区地址
POP     DPL
DJNZ    R7 , LOOP          ; 若 8 个值没有输出完毕, 则继续
RET
TIM:    MOV     R3 , # 18H
AGAIN:  NOP
NOP
DJNZ    R3 , AGAIN
RET

```

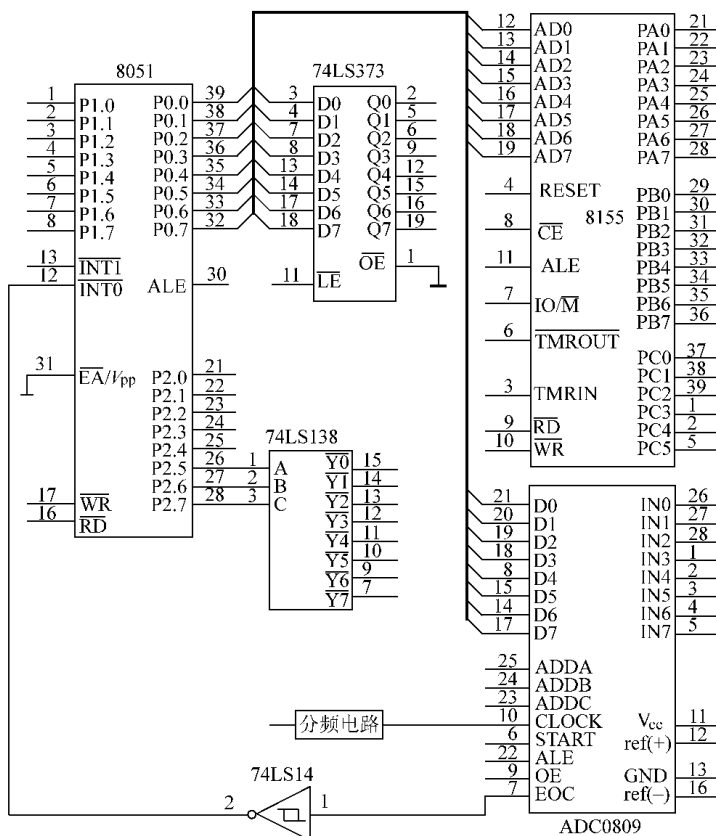


图 10-6 MCS-51 应用接口电路

3. 图 10-6 所示的 MCS-51 系统扩展了一片 ADC0809 和一片 8155, 但电路未完成, 现要求:

(1) 完成图中 ADC0809 和 8155 的连接, 并分析出 ADC0809 的 IN0 ~ IN7 通道地址, 以及 8155 的命令/状态字寄存器、PA 口、PB 口和 PC 口的地址。

(2) 用 ADC0809 的通道 IN1 进行连续 AD 转换, 采样时间间隔自定, 采样值存入 8051 内部 RAM 的 50H 单元。若采样值大于 80H, 置 8155 的 PA0 输出高电平。请写出该程序。

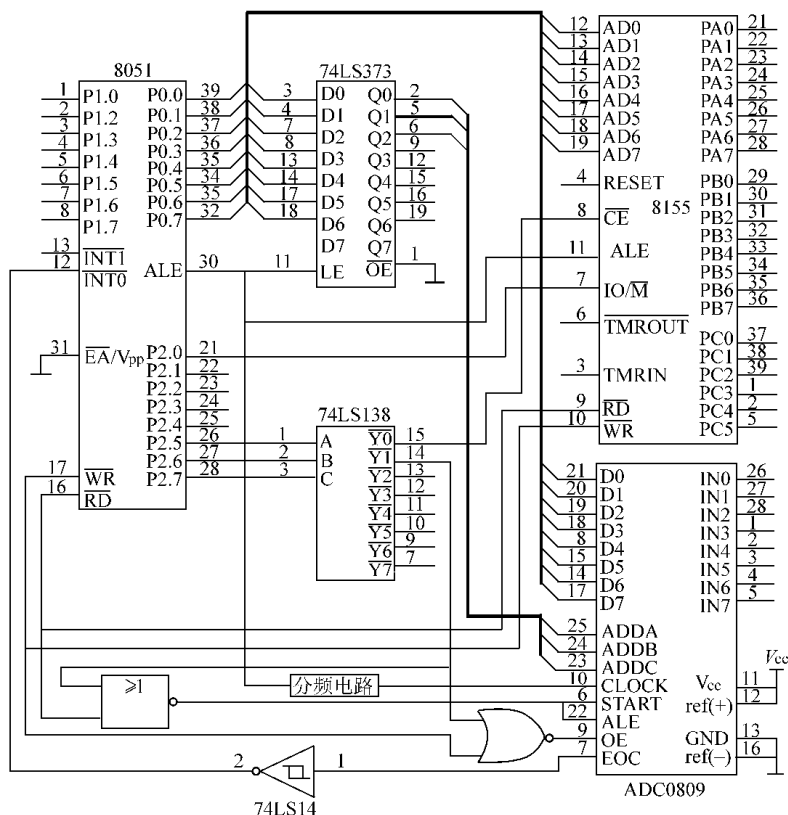


图 10-7 单片机应用接口电路

解：(1) 线路连接如图 10-7 所示。由此图可得：ADC0809 被选中时，74LS138 的 $\overline{Y1} = 0$ (P2.7 = 0、P2.6 = 0、P2.5 = 1)，故 ADC0809 各通道的一组地址为：

IN0：2000H (P2.7 = 0、P2.6 = 0、P2.5 = 1，其他为 0)

IN1：2001H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.0 = 1，其他为 0)

IN2：2002H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.1 = 1，其他为 0)

IN3：2003H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.1 = 1、P0.0 = 1，其他为 0)

IN4：2004H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.2 = 1，其他为 0)

IN5：2005H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.2 = 1、P0.0 = 1，其他为 0)

IN6：2006H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.2 = 1、P0.1 = 1，其他为 0)

IN7：2007H (P2.7 = 0、P2.6 = 0、P2.5 = 1、P0.2 = 1、P0.1 = 1、P0.0 = 1，其他为 0)

8155 被选中时，74LS138 的 $\overline{Y0} = 0$ (P2.7 = 0、P2.6 = 0、P2.5 = 0)，故 8155 各通道地址为：

命令寄存器：1FF8H (P2.7、P2.6、P2.5、P0.2、P0.1、P0.0 为 0，其他为 1)

PA 口地址：1FF9H (P2.7、P2.6、P2.5、P0.2、P0.1 为 0，其他为 1)

PB 口地址：1FFAH (P2.7、P2.6、P2.5、P0.2、P0.0 为 0，其他为 1)

PC 口地址：1FFBH (P2.7、P2.6、P2.5、P0.2 为 0，其他为 1)

(2) 利用外部中断检测转换结束标志信号。转换处理程序如下：

主程序：

```

MAIN:  MOV    DPTR, # 1FF8H      ; 8155 命令字端口
        MOV    A, # 0DH          ; 命令控制字, 方式 ALT4, A 口双向
        MOVX   @DPTR, A
        SETB   EA                ; 总中断允许
        SETB   EX0               ; 允许INT0中断
        SETB   IT0               ; INT0边沿触发
        MOV    R0, # 50H         ; 置内 RAM 数据区首地址
        MOV    DPTR, # 2000H     ; 通道 IN1 地址送 DPTR
        MOV    A, # 0H
        MOVX   @DPTR, A         ; 启动 A/D 转换
HERE:  AJMP    HERE              ; 等待中断

```

中断服务子程序:

```

        MOV    DPTR, # 2000H     ; 通道 IN1 地址
        MOVX   A, @DPTR          ; 输入数字量送 A
        MOV    @R0, A            ; 存入内 RAM
        INC    R0
        CJNE   A, # 80H, CHK      ; 数值比较
CHK:    JC     ST                 ; 若输入的数小于 80H, 则转移
        MOV    DPTR, # 1FF9H     ; 8155 的 A 口地址送 DPTR
        MOVX   A, @DPTR
        ORL    A, # 01H          ; PA0 置高
        MOVX   @DPTR, A
ST:     RETI

```

4. 图 10-8 所示的 MCS-51 系统扩展了一片 EPROM 芯片 2764、一片 RAM 芯片 6264 和一片并行口芯片 8155。设定 2764 的地址范围为 0000H ~ 1FFFH, 6264 的地址范围为 0000H ~ 1FFFH, 8155 内 RAM 地址为 2000H ~ 20FFH, 8155 I/O 口地址为 21XXH。要求:

(1) 补充完成图中的系统硬件连接。

(2) 设定 8155 PA 口为基本输出方式, PB 口为基本输入方式。每 0.1s 读取 PB 口的状态值放入 RAM 6264 的起始单元, 并将该数据取反后由 PA 口输出, 依此循环。设计出相应的软件程序。系统时钟频率为 6MHz。

解: (1) 硬件连接补充连线如图 10-9 所示。

(2) 根据题意, 8155 的控制字为: C1H, 即 A 口出, B 口入, 定时器启动。

命令状态口地址: 2100H; A 口地址: 2101H; B 口地址: 2102H; C 口地址: 2103H

定时器低 8 位地址: 2104H; 定时器高 8 位地址: 2105H。

程序实现方法一 (软件延时法):

```

        MOV    DPTR, # 2100H     ; 命令口地址
        MOV    A, # 0C1H         ; 命令字
        MOVX   @DPTR, A
LOOP:   MOV    DPTR, # 2102H     ; B 口地址

```

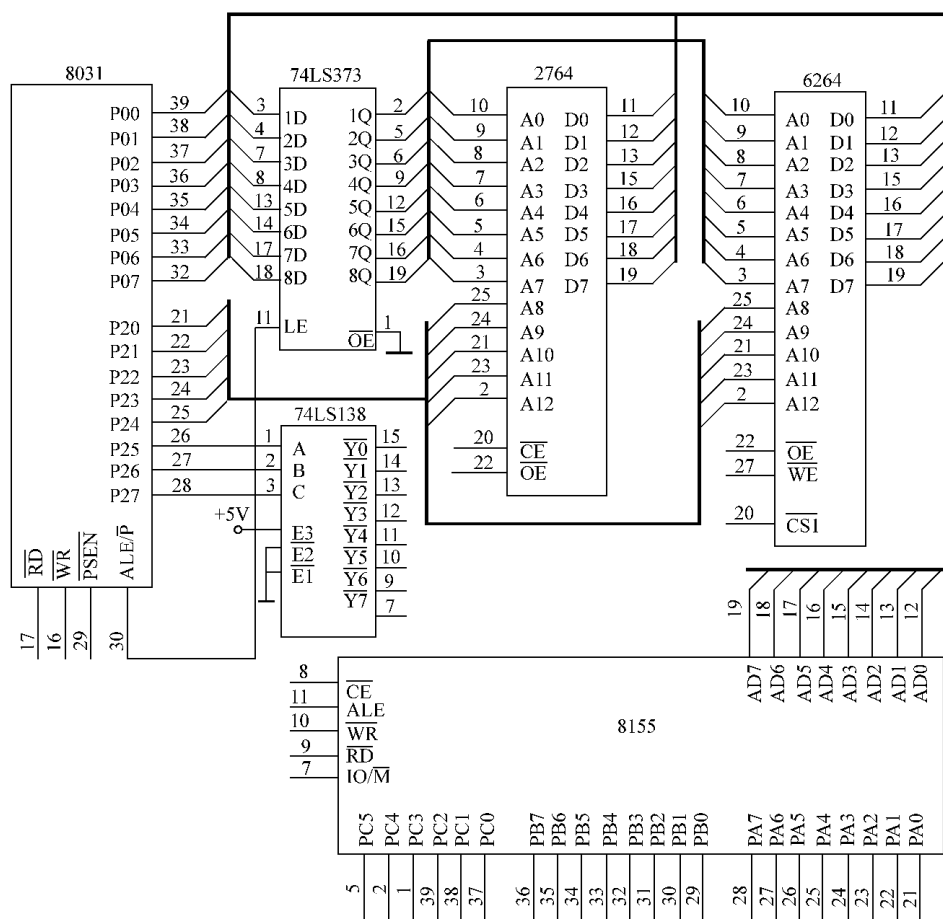


图 10-8 单片机系统扩展电路

MOVX	A, @DPTR	；读入 B 口状态值
MOV	DPTR, # 0000H	；6264 RAM 起始单元地址
MOVX	@DPTR, A	
CPL	A	；将读入的数取反
MOV	DPTR, # 2101H	；A 口地址
MOVX	@DPTR, A	；由 A 口输出
ACALL	DELAY	；延时
SJMP	LOOP	

DELAY:	MOV	R1, # 69H	；延时 0.1s 子程序
L1:	MOV	R0, # 180H	
L2:	NOP		
	NOP		
	NOP		
	DJNZ	R0, L2	

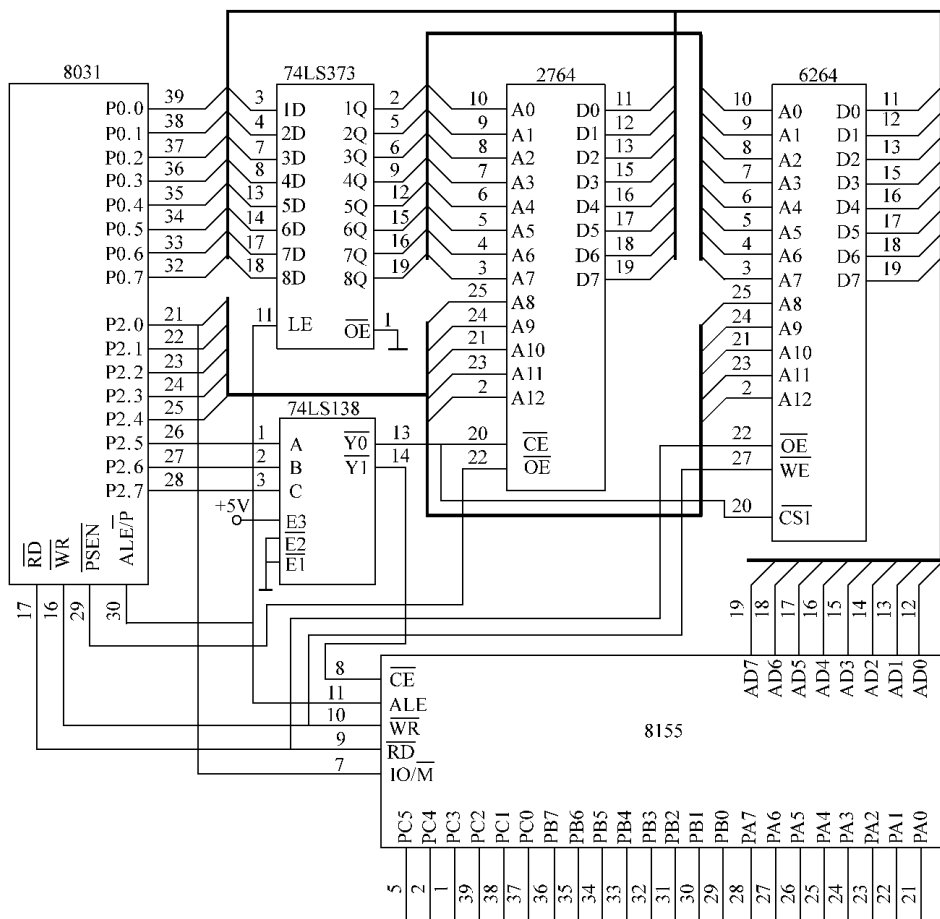


图 10-9 单片机系统扩展电路图

DJNZ R1, L1

RET

程序实现方法二（硬件定时法）：设 T1 工作为方式 1，定时为 0.1s，则初值为 15536，即 3CB0H。故 T1 的高 8 位初值为：3CH，T1 的低 8 位初值为：B0H。

利用查询方式检测定时结束标志，故程序如下：

```

MAIN: MOV     DPTR, # 2100H           ; 初始化 8155
      MOV     A, # 0C1H               ; 命令字
      MOVX    @DPTR, A
      MOV     TMOD, # 10H             ; T1 方式 1，定时
      SETB    TR1                     ; 启动 T1
LOOP:  MOV     TL1, # 0B0H             ; 装入 T1 计数初值
      MOV     TH1, # 3CH
WAIT:  JBC     TF1, NEXT               ; 定时时间到，清 T1 标志位，转移
      SJMP    WAIT                     ; 等待溢出
NEXT:  MOV     DPTR, # 2102H           ; B 口地址
  
```

```

MOVX  A, @DPTR          ; 读入 B 口状态值
MOV   DPTR, # 0000H      ; 6264 RAM 起始单元地址
MOVX  @DPTR, A
CPL   A                  ; 将读入的数取反
MOV   DPTR, # 2101H      ; A 口地址
MOVX  @DPTR, A           ; 由 A 口输出
SJMP  LOOP
END

```

5. AT89 单片机是美国 ATMEL 公司在 INTEL 公司单片机 8031 核的基础上生产的 8 位单片机, 内部集成了 Flash 存储器, 故又称为 Flash 单片机。图 10-10 是利用 AT89C2051 进行水塔水位控制的电路图, 继电器带电时驱动电动机补水。请自行分析工作原理并编制运行控制程序。

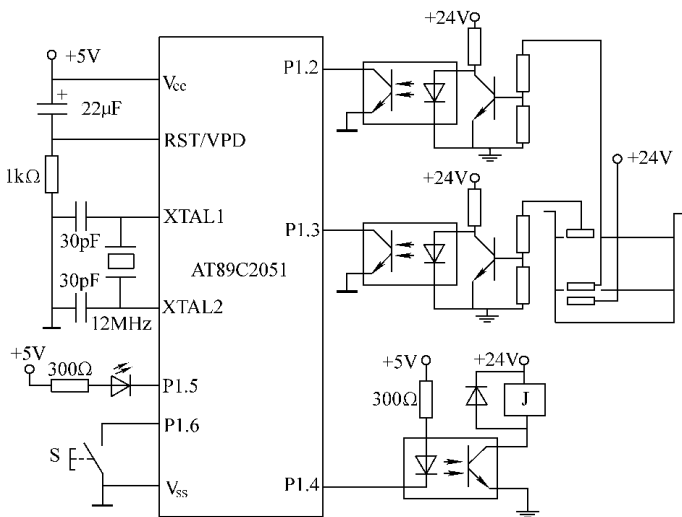


图 10-10 水塔水位控制电路

解: 利用 P1.2 和 P1.3 连接的导体检测塔内水位, 通过 P1.4 光电隔离驱动的继电器控制异步电动机带动水泵向塔内补水。当出现系统故障时, 利用 P1.5 控制发光二极管 LED 进行灯光报警。由图可以得到表 10-2 所示的工作关系控制表。

主程序如下:

```

ORG    0000H
LOOP:  MOV   A, P1          ; 读 P1 口内容到 A
        JB   ACC.6, GO      ; 水位超下限时, 转 TWO
        SETB P1.4          ; 按键没有按下时, 停机
        AJMP LOOP          ; 转 LOOP, 继续查询 S 状态
GO:     JNB  ACC.2, ONE     ; 水位为下限时, 转 ONE
        JB   ACC.3, TWO     ; 水位超下限时, 转 TWO
BACK:   ACALL D5S          ; 调用 5s 延时子程序
        AJMP LOOP          ; 转 LOOP 继续循环

```

表 10-2 检测输入与水位状态及水泵电动机工作状态对应关系表

P1.2	P1.3	水位状态	电动机状态
0	0	低于下限	运转
1	0	处于上、下限制之间	维持
1	1	升至上限	停转
0	1	混乱 (检测单元故障)	停机

```
ORG    0100H
```

```
D5S :      MOV      R3, # 19H
LOOP3 :    MOV      R1, # 85H
LOOP1 :    MOV      R2, # 0FAH
LOOP2 :    DJNZ     R2, LOOP2
           DJNZ     R1, LOOP1
           DJNZ     R3, LOOP3
           RET
           END
```

参 考 文 献

1. 霍孟友等著. 单片机原理与应用. 北京: 机械工业出版社, 2004
2. 马家辰, 孙玉德编. MCS—51 单片机原理及接口技术 (修订版). 哈尔滨: 哈尔滨工业大学出版社, 1998
3. 徐淑华, 程退安, 姚万生著. 单片微型机原理及应用. 哈尔滨: 哈尔滨工业大学出版社, 1997
4. 李朝青著. 单片机原理及接口技术. 北京: 北京航空航天大学出版社, 1999
5. 何利民著. MCS—51 单片机及应用系统设计. 北京: 北京航空航天大学出版社, 1998
6. 张洪洞, 蓝清华著. 单片机应用技术教程. 北京: 清华大学出版社, 1997
7. 朝乾斌, 李光斌著. 单片微型计算机原理与应用. 武汉: 华中理工大学出版社, 1998
8. 潘新民等著. 单片微型计算机使用系统设计. 北京: 人民邮电出版社, 1992