

ATMEL89 系列 单片机应用技术

余永权 编著

北京航空航天大学出版社

<http://www.buaapress.com.cn>

内 容 简 介

8051 系列单片机是应用较广的单片机。美国 ATMEL 公司生产的 89 系列单片机是和 8051 兼容的新型单片机,并且是目前取代传统 8051 单片机的主导机型。本书介绍 89 系列单片机的应用技术,包括 I/O 接口技术,存储器扩展技术以及 89 系列单片机在家用电器、自动控制、仪器仪表和信息通信领域的实际应用例子。这些技术方法和应用例子对单片机用户,尤其是对传统的 MCS-51 单片机用户特别有参考价值。

本书可作为单片机应用人员的参考书,也可作为 8051 系列单片机开发人员的开发应用读物。

图书在版编目(CIP)数据

ATMEL89 系列单片机应用技术/余永权编著. —北京:
北京航空航天大学出版社,2002.4
ISBN 7-81077-136-1

I. A… II. 余… III. 单片微型计算机,ATMEL89
系列 IV. TP368.1

中国版本图书馆 CIP 数据核字(2002)第 002424 号

ATMEL89 系列单片机应用技术

余永权 编著

责任编辑 刘晓明

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话(010)82317024 传真(010)82328026

<http://www.buaapress.com.cn>

E-mail:pressell@publica.bj.cninfo.net

河北省涿州市新华印刷厂印制 各地书店经销

*

开本:787×1092 1/16 印张:21.5 字数:550 千字

2002 年 4 月第 1 版 2002 年 8 月第 2 次印刷 印数:5 001~15 000 册

ISBN 7-81077-136-1/TP·077 定价:32.00 元

前 言

MCS-51 系列单片机是 Intel 公司在 20 世纪 80 年代初研制出来的,很快就在我国得到广泛的推广应用。十多年来,MCS-51 系列单片机无论在教学、工业控制、仪器仪表、信息通信,还是在交通、航运、家用电器领域,都取得了大量的应用成果。Intel 公司虽然已经把精力集中在计算机的 CPU 生产上,而渐渐放弃了微控制器的生产,但是,以 MCS-51 技术核心为主导的微控制器技术已被 ATMEL,PHILIPS 等公司所继承,并且在原有基础上又进行了新的开发,从而产生了和 MCS-51 兼容而功能更加强劲的微控制器系列。ATMEL 公司所生产的 89 系列单片机就是基于 Intel 公司的 MCS-51 系列而研制的并与 MCS-51 兼容的微控制器系列。

ATMEL 公司是美国在 20 世纪 80 年代中期成立并发展起来的半导体公司。该公司的技术优势在于 Flash 存储器技术、高质高可靠性生产技术。随着业务的发展,在 20 世纪 90 年代初,ATMEL 公司一跃成为全球最大的 EEPROM 供应商。1994 年,为了介入单片机市场,ATMEL 公司以 EEPROM 技术和 Intel 公司的 80C31 单片机核心技术进行交换,从而取得 80C31 核的使用权。ATMEL 公司把自身的先进 Flash 存储器技术和 80C31 核相结合,从而生产出了 Flash 单片机 AT89C51 系列。这是一种内部含 Flash 存储器的特殊单片机。由于它内部含有大容量的 Flash 存储器,所以,在产品开发及生产便携式商品、手提式仪器等方面有着十分广泛的应用,也是目前取代传统的 MCS-51 系列单片机的主流单片机之一。

为了使广大 MCS-51 系列单片机用户对 AT89C51 系列单片机的应用有一个广泛的了解,专门编写了这本书。

本书一共分成十章。第一章介绍 89 系列单片机概况;第二章介绍 89 系列单片机基本结构;第三章介绍 89 系列单片机的接口部件结构及功能;第四章介绍指令系统;第五章介绍 89 系列单片机的输入输出

接口技术;第六章介绍存储器应用及扩展技术;第七章介绍 89 系列单片机在家用电器领域的应用;第八章介绍 89 系列单片机在自动控制领域的应用;第九章介绍 89 系列单片机在仪器仪表领域的应用;第十章介绍 89 系列单片机在信息通信领域的应用。

本书的特点是,不但介绍了 89 系列单片机在应用中的一些技术性方法,而且更重要的是给出了大量的实际应用例子,而这些例子在各个领域应用中都有较实际的参考价值。

本书在编写过程中得到有关部门和人员的积极支持,特别是得到北京航空航天大学出版社有关领导的大力支持以及马广云博士的积极帮助;另外,本书还参考和引用了有关方面的应用例子,其来源都在书后的参考文献中给出,在此对有关作者表示诚挚的感谢。这些例子给本书增色不少。

在本书的编写过程中,编者力求做到准确无误,条理分明,易于理解。由于编者水平有限,书中可能还存在各种不妥之处,敬请读者指正。

编者于广东工业大学

2001 年 10 月

目 录

第一章 89 系列单片机概况

1.1 89 系列单片机简述	1
1.1.1 89 系列单片机的特点	1
1.1.2 89 系列单片机结构简况	1
1.1.3 89 系列单片机的型号说明	2
1.2 89 系列单片机的分档	3
1.2.1 标准型	3
1.2.2 低档型	4
1.2.3 高档型	4

第二章 89 系列单片机基本结构

2.1 结构框图	6
2.2 存储器组织	7
2.2.1 程序存储器	8
2.2.2 数据存储器	9
2.2.3 地址分配及寻址区	11
2.3 CPU 定时时序	17
2.4 中断系统结构	20
2.4.1 中断控制寄存器	20
2.4.2 中断响应过程	21
2.5 特殊功能寄存器 SFR	23
2.6 89 系列典型单片机结构	25
2.6.1 AT89C51 单片机	25
2.6.2 AT89C2051 单片机	29
2.6.3 AT89S8252 单片机	31

第三章 接口部件结构及功能

3.1 I/O 端口的结构及功能	36
3.1.1 I/O 端口的结构	37
3.1.2 写端口操作	38
3.1.3 读端口操作	39
3.1.4 读—修改—写端口操作	39
3.1.5 外部存储器的存取操作	40
3.2 串行接口的结构及功能	41
3.2.1 串行口的方式控制	42

3.2.2 串行口的工作方式	43
3.2.3 多机通信过程	49
3.2.4 波特率的产生	50
3.3 定时器的结构及功能	52
3.3.1 定时器 T0,T1	53
3.3.2 定时器 T2	57
3.4 中断功能	61
3.4.1 中断请求	61
3.4.2 中断控制和优先	62
3.4.3 中断处理过程	63
3.4.4 中断过程的注意事项	64
3.5 振荡器、复位和省电方式	65
3.5.1 振荡器	65
3.5.2 复位	67
3.5.3 省电方式	69

第四章 指令系统

4.1 指令寻址方式及标志位	71
4.1.1 指令寻址方式	71
4.1.2 指令的状态标志	72
4.2 指令系统分类说明	73
4.2.1 算术操作类指令	73
4.2.2 逻辑操作类指令	74
4.2.3 数据传送类指令	76
4.2.4 布尔变量操作类指令	79
4.2.5 控制程序转移类指令	80
4.3 指令系统表	82
4.3.1 指令系统总表	82
4.3.2 按操作码排序的指令总表	85

第五章 输入输出接口技术

5.1 串-并、并-串行转换技术	89
5.2 A/D 转换技术	91
5.2.1 外接 RC 实现 A/D 转换	91
5.2.2 一次积分 A/D 转换	95
5.2.3 逐次逼近 A/D 转换	98
5.3 显示接口技术	100
5.3.1 LED 显示接口技术	100
5.3.2 LCD 显示接口技术	103
5.4 和 PC 串行通信技术	111
5.4.1 串行通信的电路结构和原理	111

5.4.2 串行通信的软件	112
5.5 晶闸管控制接口技术	114
5.5.1 单相自动升压的晶闸管接口技术	115
5.5.2 三相桥式晶闸管接口技术	117

第六章 存储器应用及扩展技术

6.1 签名字节的读出及存储器加密	124
6.1.1 签名字节的读出	124
6.1.2 存储器加密	127
6.2 Flash 存储器的编程方法	129
6.2.1 AT89C51 Flash 存储器编程	130
6.2.2 AT89C1051 Flash 存储器编程	134
6.3 串行 EEPROM 接口方法	136
6.3.1 串行 EEPROM 及其工作原理	136
6.3.2 AT24CXX 系列 EEPROM 的接口方法	152
6.3.3 AT93C06/46 的接口方法	170
6.4 数据存储器的串行扩展技术	188
6.4.1 数据存储器的串行扩展逻辑电路	188
6.4.2 读/写操作及工作程序	192

第七章 家用电器领域的应用

7.1 AT89C2051 控制的洗衣机	195
7.1.1 洗衣机的控制功能要求	195
7.1.2 控制逻辑电路	196
7.1.3 工作控制程序	198
7.2 多功能密码锁	200
7.2.1 多功能密码锁的控制电路	200
7.2.2 密码锁的控制程序	202
7.3 用电力线通信的子母电子钟	205
7.3.1 子母电子钟系统结构	205
7.3.2 母钟的逻辑结构	206
7.3.3 子钟的逻辑结构	207
7.3.4 实时时钟芯片 MC146818	208
7.3.5 MC146818 的工作	211
7.3.6 子母电子钟的工作程序	212
7.4 电话语音报警器	214
7.4.1 电话语音报警器的功能	214
7.4.2 报警器的工作原理	215
7.4.3 控制程序框图	217
7.5 电池充电器	217
7.5.1 电池的充电速率	218

7.5.2	电池的充电特性	219
7.5.3	充电终止状态检测	221
7.5.4	安全充电的措施	222
7.5.5	充电电路结构	223
7.5.6	充电器的控制软件	226

第八章 自动控制领域的应用

8.1	双坐标步进电机控制	229
8.1.1	控制系统的结构原理	229
8.1.2	步进电机控制	231
8.1.3	步进电机控制软件框图	232
8.2	交流电源电压组合控制	234
8.2.1	交流电压组合控制基本原理	235
8.2.2	控制系统的逻辑电路	238
8.2.3	电压组合控制的软件	241
8.3	地震数据采集系统	242
8.3.1	地震数据采集系统的结构框图	243
8.3.2	数据采集系统的逻辑电路	245
8.3.3	数据采集系统的工作程序	246
8.4	刀闸自动操作控制	247
8.4.1	刀闸自动操作控制原理	248
8.4.2	控制系统逻辑电路	249
8.4.3	控制系统的工作程序框图	251
8.5	滚动式广告窗控制	252
8.5.1	滚动式广告窗控制原理	252
8.5.2	超声波的发射和接收原理	253
8.5.3	语音播放原理	256
8.5.4	步进电机控制原理	259
8.5.5	控制软件框图	261

第九章 仪器仪表领域的应用

9.1	智能煤气表	262
9.1.1	预收费智能煤气表的结构原理	263
9.1.2	电路原理	264
9.1.3	控制软件的功能	266
9.2	水位记录仪	267
9.2.1	水位记录仪的基本性能	267
9.2.2	水位记录仪控制逻辑结构	268
9.2.3	水位记录仪的软件结构	269
9.3	在线湿度检测仪	271
9.3.1	湿度传感器	271

9.3.2	湿度检测原理	271
9.3.3	仪器的控制电路结构	274
9.3.4	在线湿度检测仪的软件	278
9.3.5	仪器的一些特点	280
9.4	环保设备运行记录仪	281
9.4.1	环保监测网与数据记录	281
9.4.2	系统电路逻辑结构	282
9.4.3	记录仪工作软件	285
9.5	IC 卡电度表	286
9.5.1	IC 卡电度表的逻辑电路	287
9.5.2	IC 卡电度表的控制软件	295

第十章 信息通信领域的应用

10.1	小范围专用寻呼系统	298
10.1.1	寻呼系统的组成	298
10.1.2	主叫单机的结构原理	299
10.1.3	被叫主机的结构原理	300
10.1.4	主叫单机的发射主程序流程	301
10.1.5	被叫主机接收显示主程序流程	303
10.2	通信信息的滤波与纠错	304
10.2.1	信息的滤波	304
10.2.2	RS 码的编译码器	307
10.3	无线短波传真通信控制	312
10.3.1	传真通信过程	312
10.3.2	传真转储模块电路结构	313
10.3.3	传真转储模块控制软件	315
10.4	程控交换机呼叫信息缓冲系统	317
10.4.1	呼叫信息缓冲系统的基本功能	318
10.4.2	系统硬件结构框图	318
10.4.3	呼叫信息缓冲系统软件	319
10.5	现场总线通信适配器	321
10.5.1	适配器的功能要求	322
10.5.2	适配器的电路结构	322
10.5.3	软件简介	326

附 录

参考文献

第一章 89 系列单片机概况

ATMEL89 系列单片机(简称 89 系列单片机)是 ATMEL 公司的 8 位 Flash 单片机系列。这个系列单片机的最大特点就是在片内含有 Flash 存储器,因此,有着十分广泛的用途,特别是在便携式、省电和特殊信息保存的仪器和系统中显得更为有用。

1.1 89 系列单片机简述

89 系列单片机是以 8031 核构成的。所以,它和 8051 系列单片机是兼容的系列。这个系列对于 8051 为基础的系统来说,是十分容易进行取代和构造的。故对于熟悉 8051 的用户来说,用 ATMEL 公司的 89 系列单片机进行取代 8051 的系统设计是轻而易举的事。

1.1.1 89 系列单片机的特点

89 系列单片机对于一般用户来说,存在下列很明显的优点。

1. 内部含 Flash 存储器

由于内部含 Flash 存储器,因此在系统的开发过程中可以十分容易地进行程序的修改。这就大大缩短了系统的开发周期。同时,在系统工作过程中,能有效地保存一些数据信息,即使外界电源损坏也不影响信息的保存。

2. 和 AT80C51 插座兼容

89 系列单片机的引脚和 80C51 是一样的,所以,当用 89 系列单片机取代 80C51 时,可以直接进行代换。这时,不管采用 40 引脚还是 44 引脚的产品,只要用相同引脚的 89 系列单片机取代 80C51 的单片机即可。

3. 静态时钟方式

89 系列单片机采用静态时钟方式,所以可以节省电能。这对于降低便携式产品的功耗十分有用。

4. 错误编程亦无废品产生

一般的 OTP 产品,一旦错误编程就成了废品。而 89 系列单片机内部采用了 Flash 存储器,所以,错误编程之后仍可以重新编程,直到正确为止,故不存在废品。

5. 可反复进行系统试验

用 89 系列单片机设计的系统,可以反复进行系统试验。每次试验可以编入不同的程序,这样可以保证用户的系统设计达到最优。而且随用户的需要和发展,还可以进行修改,使系统能不断追随用户的最新要求。

1.1.2 89 系列单片机结构简况

89 系列单片机的内部结构和 80C51 相近,主要含有如下一些部件。

① 8031 CPU。

- ② 振荡电路。
- ③ 总线控制部件。
- ④ 中断控制部件。
- ⑤ 片内 Flash 存储器。
- ⑥ 片内 RAM。
- ⑦ 并行 I/O 接口。
- ⑧ 定时器。
- ⑨ 串行 I/O 接口。

在 89 系列单片机中,AT89C1051 的 Flash 存储容量最小,只有 1 KB;而 AT89C52,AT89LV52,AT89S8252 的 Flash 存储器容量最大,有 8 KB。

这个系列中,结构最简单的是 AT89C1051,内部也不含串行接口;最复杂的是 AT89S8252,它内部不但含标准的串行接口,还含一个串行外围接口 SPI、Watchdog 定时器、双数据指针、电源下降的中断恢复等功能和部件。

89 系列单片机一共有 7 种型号,分别为 AT89C51,AT89LV51,AT89C52,AT89LV52,AT89C2051,AT89C1051,AT89S8252。其中 AT89LV51 和 AT89LV52 分别是 AT89C51 和 AT89C52 的低电压产品,最低电压可以低至 2.7 V。而 AT89C1051 和 AT89C2051 则是低档型低电压产品。它们只有 20 条引脚,最低电压也为 2.7 V。

1.1.3 89 系列单片机的型号说明

89 系列单片机的型号编码由 3 个部分组成,分别是前缀、型号、后缀。它们的格式如下:
AT89C××××××××

其中:AT 是前缀;

89C××××是型号;

××××是后缀。

下面分别对这 3 个部分进行说明,并且对其中有关参数的表示和含义作出相应的解释。

1. 前 缀

前缀由字母“AT”组成,表示该器件是 ATMEL 公司的产品。

2. 型 号

型号由“89C××××”或“89LV××××”或“89S××××”等表示。

“89C××××”中,9 表示内部含 Flash 存储器;C 表示是 CMOS 产品。

“89LV××××”中,LV 表示低压产品。

“89S××××”中,S 表示含可下载 Flash 存储器。

这个部分的××××表示器件型号数,例如:51,1051,8252 等。

3. 后 缀

后缀由“××××”4 个参数组成,每个参数的表示和含义不同。在型号与后缀部分有“-”号隔开。

后缀中的第一个参数×用于表示速度。它的含义如下:

×=12,表示速度为 12 MHz;

×=16,表示速度为 16 MHz;

$\times = 20$, 表示速度为 20 MHz;

$\times = 24$, 表示速度为 24 MHz。

后缀中的第二个参数 $\times$ 用于表示封装。它的含义如下:

$\times = D$, Cerdip;

$\times = J$, 塑料 J 引线芯片载体;

$\times = L$, 无引线芯片载体;

$\times = P$, 表示塑料双列直插 DIP 封装;

$\times = S$, 表示 SOIC 封装;

$\times = Q$, 表示 PQFP 封装;

$\times = A$, 表示 TQFP 封装;

$\times = W$, 表示裸芯片。

后缀中第三个参数 $\times$ 用于表示温度范围。它的含义如下:

$\times = C$, 表示商业产品, 温度范围为 $0 \sim +70\text{ }^{\circ}\text{C}$;

$\times = I$, 表示工业产品, 温度范围为 $-40 \sim +85\text{ }^{\circ}\text{C}$;

$\times = A$, 表示汽车用产品, 温度范围为 $-40 \sim +125\text{ }^{\circ}\text{C}$;

$\times = M$, 表示军用产品, 温度范围为 $-55 \sim +150\text{ }^{\circ}\text{C}$ 。

后缀中的第四个参数 $\times$ 用于说明产品的处理情况。它的含义如下:

$\times$ 为空, 表示处理工艺是标准工艺;

$\times = /883$, 表示处理工艺采用 MIL—STD—883 标准。

例如, 有一个单片机型号为“AT89C51—12PI”, 则表示含义为: 该单片机是 ATMEL 公司的 Flash 单片机, 内部是 C51 结构, 速度为 12 MHz, 封装为 DIP, 是工业用产品, 按标准处理工艺生产。

1.2 89 系列单片机的分档

89 系列单片机可分成标准型号、低档型号和高档型号 3 类。

89 系列单片机的标准型有 AT89C51 等 4 种型号。它们的基本结构和 AT89C51 是类同的, 是 80C51 的兼容产品。89 系列单片机的低档型有 AT89C1051 等 2 种型号。它们的 CPU 核和 AT89C51 是相同的, 但是并行 I/O 较少; 高档型的有 AT89S8252, 是一种可下载的 Flash 单片机。它和 IBM 微机通信进行下载程序是十分方便的。

1.2.1 标准型

89 系列单片机中, 标准型单片机有 AT89C51, AT89LV51, AT89C52, AT89LV52 这 4 种型号。

标准型的 89 系列单片机是和 MCS—51 系列单片机兼容的。在内部含有 4 KB 或 8 KB 可重复编程的 Flash 存储器, 可进行 1 000 次擦写操作。全静态工作为 0 Hz~24 MHz; 有 3 级程序存储器锁定; 内部含 128~256 字节的 RAM; 有 32 条可编程的 I/O 线; 有 2~3 个 16 位定时器/计数器; 有 6~8 级中断; 有通用串行接口; 有低电压空闲及电源下降方式。

在这 4 种型号中,AT89C51 是一种基本型号;而 AT89LV51 是一种能在低电压范围工作的改进型,可在 2.7~6 V 电压范围工作;其他功能和 AT89C51 相同。

AT89C52 是在 AT89C51 的基础上,存储器容量、定时器和中断能力等得到改进的型号。AT89C52 的 Flash 存储器容量为 8 KB,16 位定时器/计数器有 3 个,中断有 8 级。而 AT89C51 的 Flash 存储器容量为 4 KB,16 位定时器/计数器有 2 个,中断只有 6 级。AT89LV52 是 AT89C52 的低电压型号,可在 2.7~6 V 电压范围内工作。

标准型单片机的主要性能如下:

- ① 4 KB 或 8 KB 的 Flash 存储器;
- ② 128 或 256 字节内部 RAM;
- ③ 32 条可编程 I/O 线;
- ④ 2~3 个 16 位定时器/计数器;
- ⑤ 6~8 个中断源;
- ⑥ 3 级程序存储器保密;
- ⑦ 可编程串行接口;
- ⑧ 片内时钟振荡器。

1.2.2 低档型

在 89 系列单片机中,除了并行 I/O 端口数较少之外,其他部件结构基本和 AT89C51 差不多。之所以被称为低档型,主要是因为它的引脚只有 20 条,比标准型的 40 条引脚少得多。低档型的单片机有 AT89C1051 和 AT89C2051 两种型号。

AT89C1051 的 Flash 存储器只有 1 KB;RAM 只有 64 字节;内部不含串行接口;内部的中断响应只有 3 种;保密锁定位只有 2 位。这些也是和标准型的 AT89C51 有区别的地方。

AT89C2051 的 Flash 存储器只有 2 KB;RAM 只有 128 字节;保密锁定位有 2 位。这当然和 AT89C51 不同。

正因为在上述有关部件上 AT89C1051,AT89C2051 的功能比标准型 AT89C51 要弱,所以它们就处于低档位置。

低档型单片机的主要性能如下:

- ① 1 KB 或 2 KB 的 Flash 存储器;
- ② 64 或 128 字节片内 RAM;
- ③ 15 条可编程 I/O 线;
- ④ 1~2 个 16 位定时器/计数器;
- ⑤ 3~6 个中断源;
- ⑥ 2 级存储器加密;
- ⑦ 可编程串行接口;
- ⑧ 片内振荡器。

1.2.3 高档型

在 89 系列单片机中,高档型只有一种型号 AT89S8252。它是在标准型的基础上增加了一些功能形成的。所增加的功能主要有如下几点。

① 8 KB Flash 存储器有可下载功能。下载功能是由 IBM 微机通过 AT89S8252 的串行外围接口 SPI 执行的。

② 除了 8 KB Flash 存储器之外,AT89S8252 还含有一个 2 KB 的 EEPROM,从而提高了存储容量。

- ③ 含有 9 个中断响应的能力。
- ④ 含标准型和低档型所不具有的 SPI 接口。
- ⑤ 含有 Watchdog 定时器。
- ⑥ 含有双数据指针。
- ⑦ 含有从电源下降的中断恢复。

上述增加的功能使得 AT89S8252 成为 ATMEL 公司 89 系列单片机的高档型号。

表 1-1 给出了 89 系列单片机各种型号的性能比较。从中可以看出:AT89C1051 是这个系列中功能最弱的型号。由于其内部的 Flash 存储器只有 1 KB,所以,程序容量不大,只能用于功能较弱的用途。而 AT89C52 是功能较强的型号,可以用于较复杂的用途。功能最强的要数 AT89S8252。它的特别之处在于多了一个 2 KB 的 EEPROM,故可用于较复杂的控制目标。

表 1-1 89 系列单片机概况

型 号	AT89C51	AT89C52	AT89C1051	AT89C2051	AT89S8252
Flash/KB	4	8	1	2	8
片内 RAM/字节	128	256	64	128	256
I/O/条	32	32	15	15	32
定时器/个	2	3	1	2	3
中断源/个	6	8	3	6	9
串行接口/个	1	1	1	1	1
M 加密/级	3	3	2	2	3
片内振荡器	有	有	有	有	有
EEPROM/KB	无	无	无	无	2

第二章 89 系列单片机基本结构

这一章介绍 89 系列单片机的基本结构,包括结构框图、存储器及其地址分配、中断系统的结构、CPU 定时以及专用的寄存器等。

2.1 结构框图

89 系列单片机有 AT89C 系列的标准型及低档型,还有 AT89S 系列的高档型。AT89S 系列是在 AT89C 系列的基础上增加一些特别的功能部件组成的。所以,两者在结构上基本相似,但在个别功能模块上和功能上有些区别。

图 2-1 是 AT89C 单片机的结构框图。它主要由下面几个部分组成:1 个 8 位中央处理单元(CPU)、片内 Flash 存储器、片内 RAM、4 个 8 位的双向可寻址 I/O 口、1 个全双工 UART(通用异步接收发送器)的串行接口、2 个 16 位的定时器/计数器、多个优先级的嵌套中断结构,以及一个片内振荡器和时钟电路。

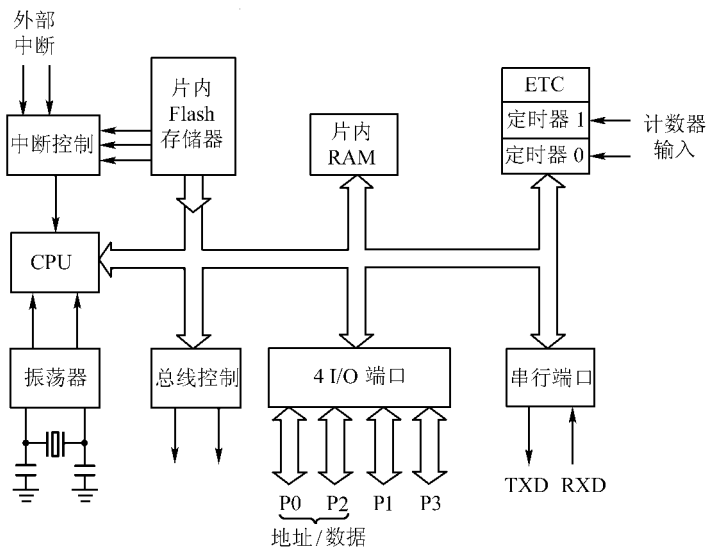


图 2-1 AT89C 单片机的结构框图

在 AT89C 单片机结构中,最显著的特点是内部含有 Flash 存储器,而在其他方面的结构,则和 Intel 公司的 8051 的结构没有太大的区别。

AT89S 单片机的结构框图如图 2-2 所示。与 AT89C 比较,它还多了片内 EEPROM、SPI 串行总线接口和 Watchdog 定时器。

为了尽可能地发挥 CMOS 电路功耗低的特点,ATMEL 公司的 Flash 单片机有两种由软件产生的低功耗方式:空闲方式(Idle Mode)和掉电方式(Power Down Mode)。

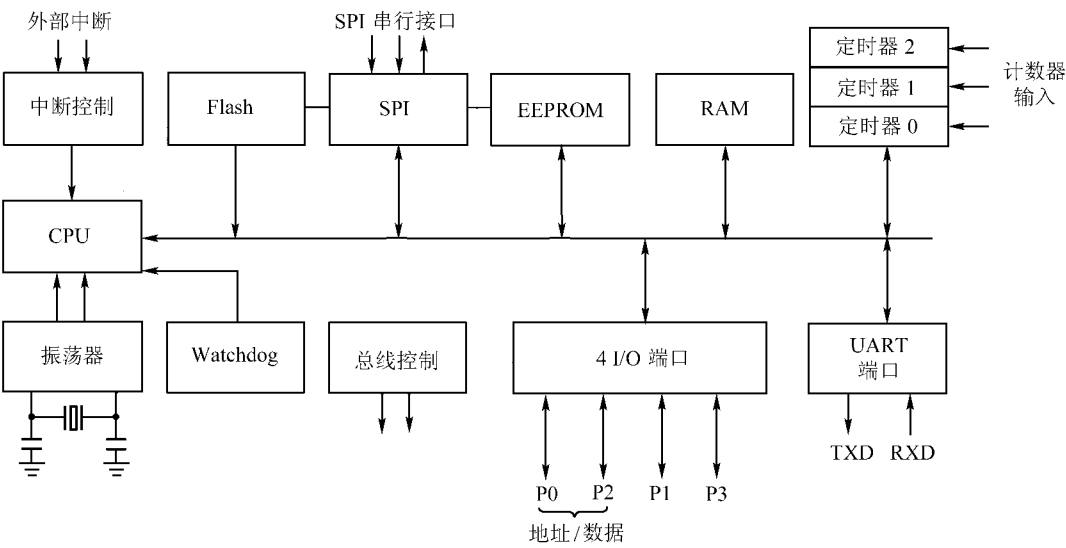


图 2-2 AT89S 单片机的结构框图

在空闲方式下,CPU 停止工作,RAM 和其他片内的部件(如振荡器、定时器/计数器、中断系统等)继续工作。此时的电流可降到大约为正常工作方式时的 15 %。

在掉电方式下,所有片内的部件都停止工作,一切功能都暂停,只有片内 RAM 的内容被保持。这种方式下的电流可降到 15 μ A 以下,最小可降到 0.6 μ A。

另外,这些部件是用静态逻辑设计的,不需要连续的时钟信号。也就是说,在等待一个内部事件时,时钟的频率可降低,甚至可停下来。

从图 2-1 和图 2-2 中可以看出:AT89S 型单片机的功能比 AT89C 型的要强,特别是 Watchdog 定时器的存在,对提高单片机的工作可靠性有很重要的作用。

2.2 存储器组织

所有的 ATMEL Flash 单片机都将程序存储器和数据存储器分为不同的存储空间。89 系列单片机的典型存储器的结构如图 2-3 所示。

程序和数据存储器分为不同的逻辑空间,使得可用 8 位地址来访问数据存储器。这样可以提高 8 位 CPU 的存储和处理速度。尽管如此,也可通过数据指针(DPTR)寄存器来产生 16 位的数据存储器地址。

程序存储器只可读不可写,用于存放编好的程序和表格常数。89 系列单片机可寻址的程序存储器总空间为 64 KB。外部程序存储器的读选通脉冲为 $\overline{\text{PSEN}}$ (程序存储允许信号)。

数据存储器在物理上和逻辑上都分为两个地址空间:一个内部和一个外部数据存储器空间。外部数据存储器的寻址空间可达 64 KB。访问外部数据存储器时,CPU 发出读和写的信号—— $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ 。

将 $\overline{\text{RD}}$ 和 $\overline{\text{PSEN}}$ 两个信号加到一个与门的输入端,然后用与门的输出作为外部程序/数据存

储器的读选通脉冲。这样就可将外部程序存储器空间和外部数据存储器空间合并在一起。

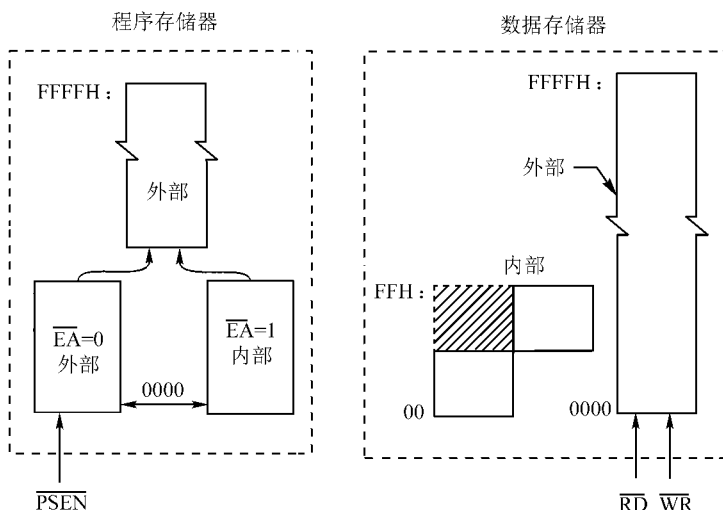


图 2-3 AT89C51/LV51 和 AT89C52/LV52 存储器结构

2.2.1 程序存储器

89 系列单片机可寻址的内部和外部程序存储器总空间为 64 KB。它没有采用程序存储器分区的方法,64 KB 的地址空间是统一的。 $\overline{EA}$ 引脚接高电平时,单片机执行内部 ROM 中的命令; $\overline{EA}$ 引脚接低电平时,单片机就从外部程序存储器中取指。

程序存储器中有几个单元专门用来存放特定的程序。这几个单元的配置情况如图 2-4 所示。

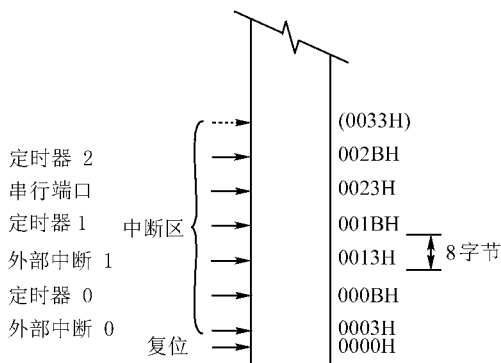


图 2-4 程序存储器的中断入口配置

由图 2-4 可知,0000H~0002H 单元用于初始化程序。单片机复位后,CPU 总是从 0000H 单元开始执行程序。另外,每个中断在程序存储器中都分配有一个固定的入口地址。中断响应后,CPU 便跳到该单元,在这里开始执行中断服务子程序。例如,外部中断 0 的入口地址被放在 0003H 单元,如果使用外部中断 0,则它的中断服务子程序必须从 0003H 单元开始。如果中断没有使用,那么它的服务单元也可作一般用途的程序存储器用。

每个中断入口地址的间隔为 8 个单元;外部中断 0 的入口地址为 0003H;定时器 0 的入口地址为 000BH;外部中断 1 的入口地址为 0013H;定时器 1 的入口地址为 001BH;以此类推。如果一个中断服务子程序足够短的话,则可全部存放在这 8 个单元中。对较长的服务子程序,则可利用一条跳转指令跳过后续的中断入口地址。

程序存储器最低端的地址可以在片内 Flash 中,或在外部存储器中。将外部存取($\overline{EA}$)引

脚接 V_{CC} 或接地,就可进行这种选择。例如,在带有 4 KB 片内 Flash 的 AT89C51 中,如果把 $\overline{EA}$ 引脚连到 V_{CC} ,当地址为 0000H~0FFFH 时,则访问内部 Flash;当地址为 1000H~FFFFH 时,则访问外部程序存储器。在 AT89C52(8 KB Flash)中,当 $\overline{EA}$ 端保持高电平时,如果地址不超过 1FFFH,则访问内部 Flash;地址超过 1FFFH(即为 2000H~FFFFH)时,将自动转向外部程序存储器。如果 $\overline{EA}$ 端接地,则只访问外部程序存储器,不管是否有内部 Flash 存储器。

外部程序存储器读选通信号 $\overline{\text{PSEN}}$ 用于读取所有的外部程序;读取内部程序时,不产生 $\overline{\text{PSEN}}$ 信号。

执行外部程序时的硬件连接方法如图 2-5 所示。

注意,在访问外部程序存储器时,16 条 I/O 线(P0 和 P2)作为总线使用。P0 端口作为地址/数据总线使用。它先输出 16 位地址的低 8 位 PCL,然后进入悬浮状态,等待程序存储器送出的指令字节。当有效地址 PCL 出现在 P0 总线上时,ALE(允许地址锁存)把这个地址锁存到地址锁存器中。同时,P2 端口输出地址的高 8 位 PCH。然后 $\overline{\text{PSEN}}$ 选通外部程序存储器,使指令送到 P0 总线上,由 CPU 取入。

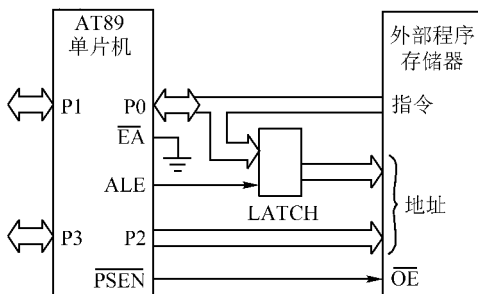


图 2-5 访问外部程序存储器的连接方法

即使所用的程序存储器的实际空间可能小于 64 KB, 程序存储器的地址总是为 16 位的。在访问外部程序存储器时, 要用到两个 8 位端口——P0 和 P2 来产生程序存储器的地址。

2.2.2 数据存储器

数据存储器在物理上和逻辑上都分为两个地址空间:一个为内部数据存储器空间;一个为外部数据存储器空间。数据存储器的配置如图 2-3 所示。

图 2-6 是访问 2 KB 外部 RAM 时的硬件连接图。在这种情况下, CPU 执行内部 Flash 中的指令($\overline{\text{EA}}$ 接 V_{CC})。P0 端口用作 RAM 的地址/数据总线, P2 端口中的 3 位也作为 RAM 的页地址。访问外部 RAM 期间, CPU 根据需要发送 $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ 信号。

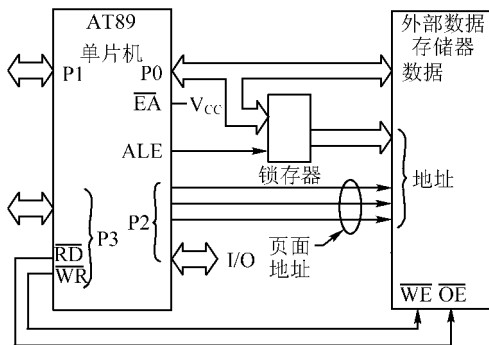


图 2-6 访问外部数据存储器的连接方法

外部数据存储器的寻址空间可达 64 KB。外部数据存储器的地址可以是 8 位或 16 位的。使用 8 位地址时,要连同另外一条或几条 I/O 线作为 RAM 的页地址,如图 2-6 所示。这时 P2 的部分引线可作为通用的 I/O 线。若采用 16 位地址,则由 P2 端口传送高 8 位地址。

内部数据存储器的地址是 8 位的,也就是说其地址空间只有 256 字节,但内部 RAM 的寻址方式实际上可提供 384 字节。高于 7FH 的直接地址访问同一个存储空

间,高于 7FH 的间接地址访问另一个存储空间。这样,在图 2-7 中,虽然高 128 字节区与专用寄存器,即特殊功能寄存器(SFR)区的地址是重合的(80H~FFH),但实际上它们是分开的。究竟访问哪一区,是通过不同的寻址方式加以区分的。访问高 128 字节区时,采用间接寻址方式;访问 SFR 区时,采用直接寻址方式;访问低 128 字节区时,两种寻址方式都可采用。

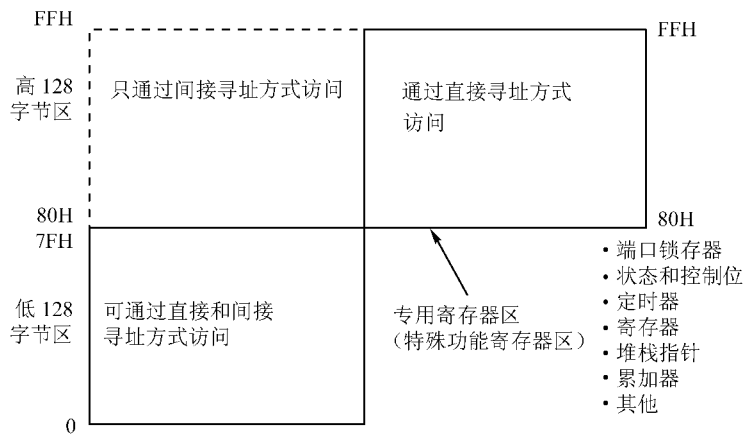


图 2-7 内部数据存储器的结构

低 128 字节区的分配情况如图 2-8 所示。最低 32 个单元(00H~1FH)是 4 个通用工作寄存器组。每个寄存器组含有 8 个 8 位寄存器,编号为 R0~R7。专用寄存器 PSW(程序状态字)中有 2 位(RS0,RS1)用来确定采用哪一工作寄存器组。这种结构能够更有效地使用指令空间,因为寄存器指令比直接寻址指令更短。

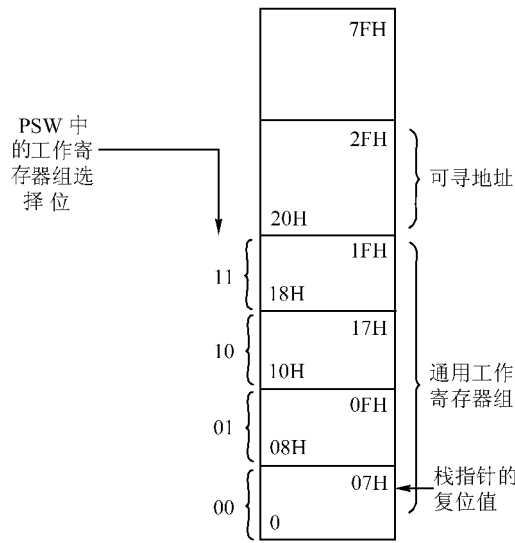


图 2-8 内部 RAM 的低 128 字节区

工作寄存器组上面的 16 个单元(20H~2FH)构成了布尔处理机的存储器空间。这 16 个单元的 128 位各自都有专门的位地址,如图 2-9 所示。它们可以被直接寻址,这些位地址是 00H~7FH。在 89 系列单片机的指令系统中,还包括了许多位操作指令,这些位操作指令可直接对这 128 位寻址。

低 128 字节区中的所有单元都既可通过直接寻址方式访问,又可通过间接寻址方式访问。而高 128 字节区则只能通过间接寻址方式来访问。仅在带有 256 字节 RAM 的单片机中才有高 128 字节区。

专用寄存器即特殊功能寄存器(SFR)区的分配情况如图 2-10 所示。

这些专用寄存器包括端口锁存器、程序状态字、定时器、累加器、栈指针、数据指针,以及其他控制寄存器等等。专用寄存器只能通过直接寻址方式来访问。通常,在所有 ATMEL 单片机的

专用寄存器(SFR)区中,寄存器的分配情况是相同的。像 AT89C51 和其他兼容机一样,在相同的地址中寄存器也是一样的。但是在 AT89C51 的升级换代产品中,还有一些附加的专用寄存器。

专用寄存器区中有一些单元是既可字节寻址又可位寻址的(见图 2-10)。凡是地址以“0”和“8”结尾(能被 8 整除)的单元都是可位寻址的,地址的范围是 80H~FFH。

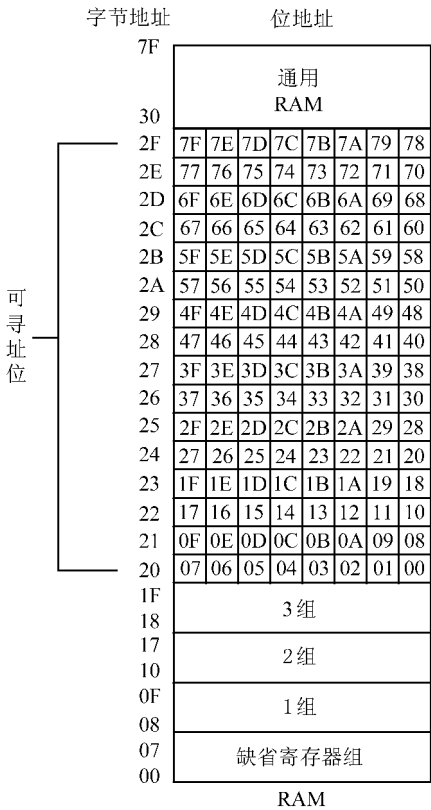


图 2-9 内部 RAM 中可寻址位的地址

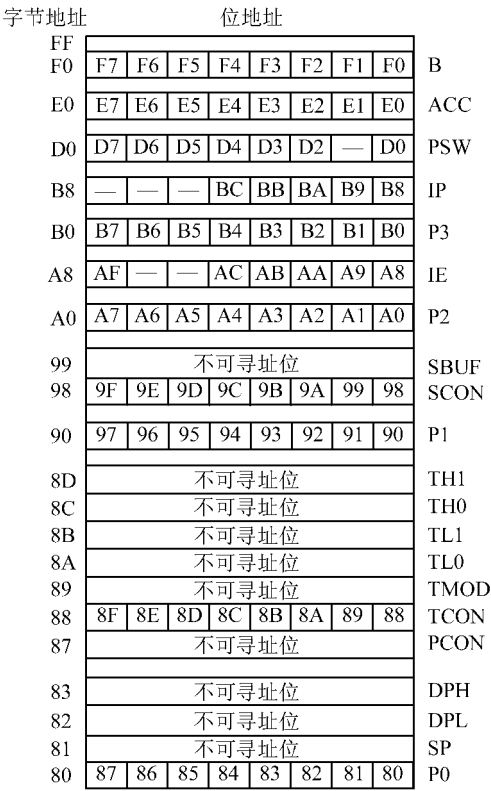


图 2-10 专用寄存器区(特殊功能寄存器区)

2.2.3 地址分配及寻址区

AT89C51, AT89C52 是最典型的 89 系列单片机。它们的存储器组织有着其自身的特点。这一节分别在程序存储器、数据存储器及其特殊区域说明这些特点。

AT89C 设有三种基本的存储器地址空间：

- 64 KB 的程序存储器地址空间(包括片内与片外)；
- 64 KB 的外部数据存储器地址空间；
- 256 字节(AT89C52 为 384 字节)的内部数据存储器空间,其中包括特殊功能寄存器。

1. 程序存储器内部及外部地址分配

AT89C 的程序存储器寻址空间为 64 KB。它可以外部寻址 64 KB;也可以内部寻址 4 KB,加上外部寻址 60 KB;或者内部寻址 8 KB,加上外部寻址 56 KB。这种地址分配方法根据型号有所区别。

AT89C51 的程序存储器的地址分配如图 2-11 所示。在图中可以看出:它的程序存储器

地址以两种形式分配：

- (1) 外部程序存储器占用全部 64 KB 地址；
- (2) 内部程序存储器占用低 4 KB 地址,外部程序存储器占用高 60 KB 地址。

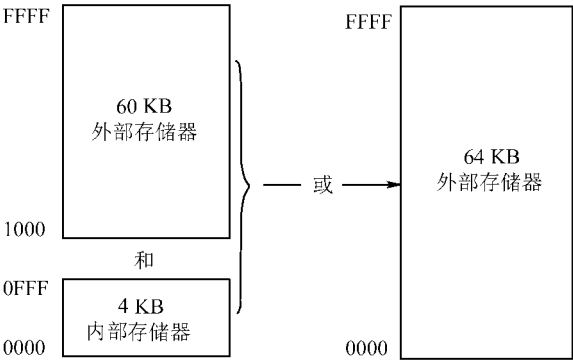


图 2 - 11 AT89C51 程序存储器

AT89C52 的程序存储器寻址空间也是 64 KB,地址分配如图 2 - 12 所示。它的程序存储器地址以两种形式分配：

- (1) 外部程序存储器占用全部 64 KB 地址；
- (2) 内部程序存储器占用低 8 KB 地址,外部程序存储器占用高 56 KB 地址。

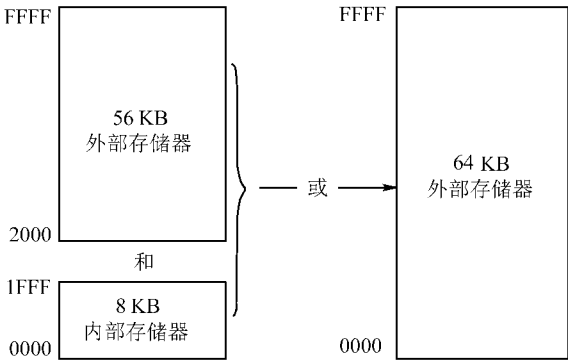


图 2 - 12 AT89C52 程序存储器

2. 数据存储器的地址分配

AT89C 可直接寻址 64 KB 外部数据存储器。指令 MOVX 为访问外部数据存储器指令。

AT89C 内部数据存储器是最灵活的地址空间,分为物理上独立的且性质不同的几个区。低 128 字节地址空间(00H~7FH)为 RAM 区。对该区访问既可以直接寻址(MOV data, addr),又可以间接寻址(MOV @Ri)。高 128 字节地址空间(80H~FFH)对于 AT89C51 为特殊功能寄存器,而对 AT89C52 则既是 RAM 区,又是特殊功能寄存器区,为二者重叠的地址空间。当这个地址空间的 RAM 区和特殊功能寄存器区共享同样的地址单元时,它们的访问是通过各自的寻址方式来加以区别的。图 2 - 13 和图 2 - 14 分别为 AT89C51 和 AT89C52 数据存储器组织。

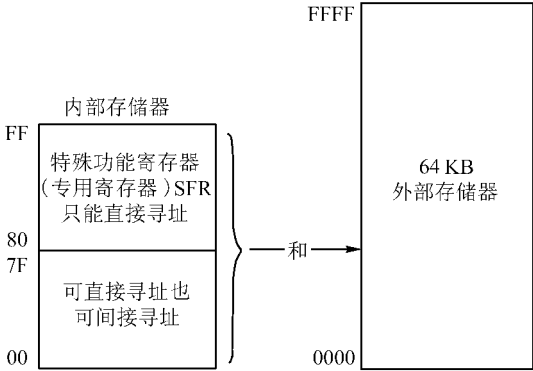


图 2 - 13 AT89C51 数据存储器组织

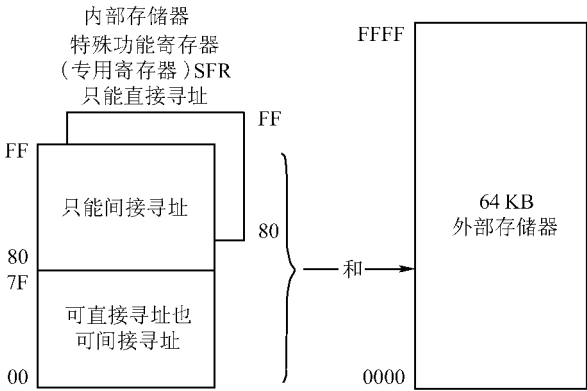


图 2 - 14 AT89C52 数据存储器组织

(1) 地址 80H~FFH 的分配

从图 2 - 13 和图 2 - 14 中可以知道,在 AT89C51 的数据存储器地址 80H~FFH 中,是用于特殊功能寄存器的;在 AT89C52 的数据存储器地址 80H~FFH 中,则可作为 RAM 存放数据,也可作特殊功能寄存器。为了区别这个地址段的功能,在 AT89C 系列单片机中,以不同寻址方式来区分。其方法如下。

① 直接寻址方式

这时把 80H~FFH 地址段当作特殊功能寄存器来访问。

典型的访问指令：

```
MOV 80H, #0AAH
```

其意义是把数据 0AAH 写到 P0 端口(P0 端口是属于特殊寄存器的一种,地址为 80H)。

② 间接寻址方式

这时把 80H~FFH 地址段当作一般数据 RAM 用。

典型的访问指令：

```
MOV R0, #80H
MOV @R0, #0BBH
```

其操作结果是把数据 0BBH 送入 RAM 的存储单元 80H 中。

不过要注意,AT89C51 不能把 80H~FFH 地址段用作数据 RAM 的存储单元,只能用作特殊功能寄存器。

(2) 00H~7FH 的地址分配

访问低 128 字节 RAM 时,既可以直接寻址,也可以间接寻址。如图 2-15 所示,该 RAM 区可以分为 3 部分。

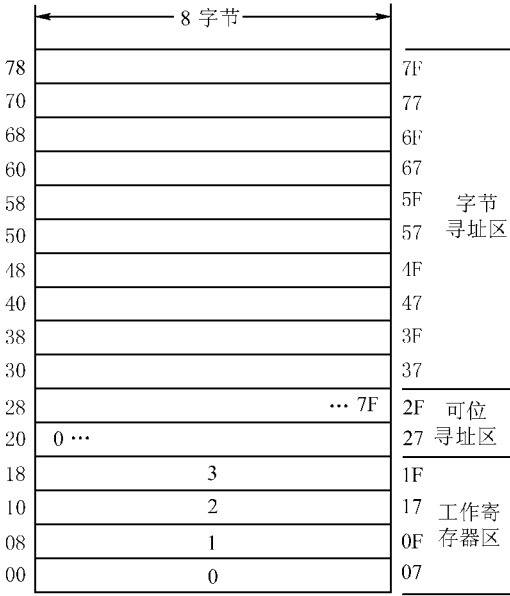


图 2-15 128 字节直接和间接寻址 RAM 区

位,每位均可直接寻址(00H~7FH)。

这些位地址有两种表示方式:第一种是位地址方式,即 00H~7FH;另一种是用字节地址(20H~2FH)加位数方式表示,两者之间用“.”号隔开。例如,位地址 00H~07H 也可表示为 20.0~20.7,位地址 08H~0FH 与 21.1~21.7 表示相同的地址区域。

这 16 字节 RAM 也可以字节寻址。

③ 字节寻址区

30H~7FH 共 80 个字节单元,为字节寻址的内部 RAM 区,可供用户作为数据存储区。但是,如果堆栈指针初始化时设置在这个区域,就要留出足够的字节单元作为栈区,以防止在数据存储时,破坏了堆栈的内容。

3. 特殊功能寄存器 SFR 的地址分配

AT89C 内的 I/O 锁存器、定时器、串行口数据缓冲器以及各种控制寄存器和状态寄存器都以特殊功能寄存器的形式出现。它们离散地分布在 80H~FFH 的地址空间范围内。

表 2-1 列出了所有特殊功能寄存器及其地址。

图 2-16 中第一列所列的特殊功能寄存器既可以字节寻址,又可以位寻址。

系统复位后,各特殊功能寄存器的状态见表 2-2。

① 工作寄存器组 0~3

低地址 RAM 区域的 00H~1FH 单元(32 字节)为工作寄存器区,共分 4 组。每组 8 个 8 位寄存器,编号为 R0~R7。在某一时刻只能选用其中的一组工作(通过程序状态字 PSW 的工作寄存器选择位来选择,系统复位后,指向工作寄存器组 0)。

堆栈指针 SP 是一个 8 位寄存器。它指示出堆栈顶部在内部 RAM 块中的位置。系统复位后,SP 初始化 07H,使得堆栈事实上由 08H 单元开始。考虑到 08H~1FH 分属于工作寄存器区 1~3,若程序设计要用到这些区,则最好把 SP 值改置为 1FH 或更大值。

② 位寻址区

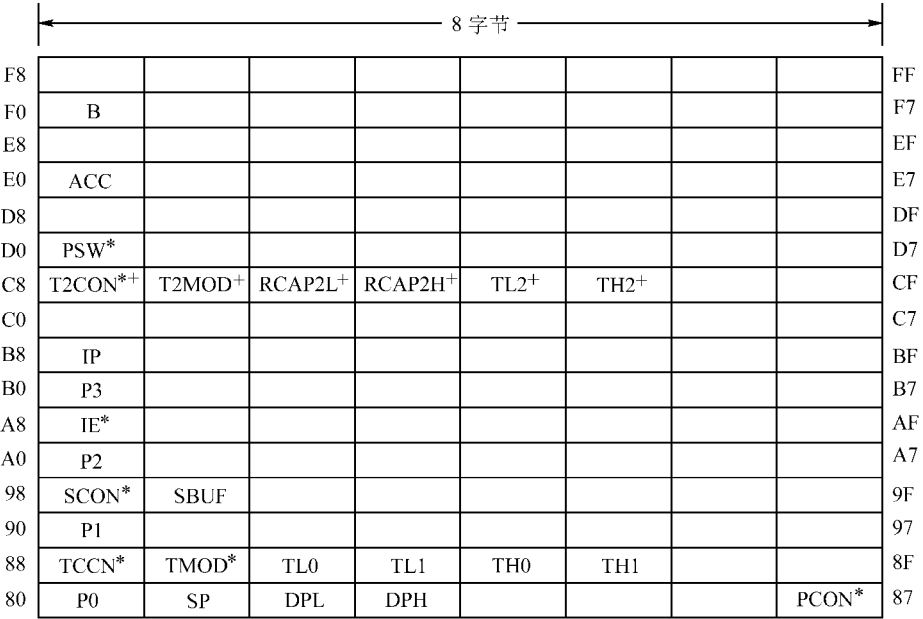
内部 RAM 区中 20H~2FH 单元(16 字节)可供位寻址。这个区域中的 128

表 2-1 特殊功能寄存器 SFR

寄存器符号	名 称	地 址
ACC *	累加器	0E0H
B *	B 寄存器	0F0H
PSW *	程序状态字	0D0H
SP	堆栈指针	81H
DPTR	数据指针(2 字节)	
DPL	低字节	82H
DPH	高字节	83H
P0 *	端口 0	80H
P1 *	端口 1	90H
P2 *	端口 2	0A0H
P3 *	端口 3	0B0H
IP *	中断优先级控制寄存器	0B8H
IE *	中断允许控制寄存器	0A8H
TMOD	定时器/计数器方式控制寄存器	89H
TCON *	定时器/计数器控制寄存器	88H
T2CON * +	定时器/计数器 2 控制寄存器	0C8H
T2MOD ⁺	定时器/计数器 2 方式控制寄存器	0C9H
TH0	定时器/计数器 0 高字节	8CH
TL0	定时器/计数器 0 低字节	8AH
TH1	定时器/计数器 1 高字节	8DH
TL1	定时器/计数器 1 低字节	8BH
TH2 ⁺	定时器/计数器 2 高字节	0CDH
TL2 ⁺	定时器/计数器 2 低字节	0CCH
RCAP2H ⁺	定时器/计数器 2 陷井寄存器高字节	0CBH
RCAP2L ⁺	定时器/计数器 2 陷井寄存器低字节	0CAH
SCON *	串行控制寄存器	98H
SBUF	串行数据缓冲器	99H
PCON	电源控制寄存器	87H

* 可位寻址；

+ 仅 AT89C52 存在。



注:可位寻址;
* 特殊功能寄存器改变方式或控制位;
+ 仅 AT89C52 存在。

图 2 - 16 特殊功能寄存器 SFR 的位置

表 2 - 2 加电或复位后特殊功能寄存器状态

寄存器	二进制值	寄存器	二进制值
ACC *	0 0 0 0 0 0 0 0	TMOD	0 0 0 0 0 0 0 0
B *	0 0 0 0 0 0 0 0	T2MOD ⁺	× × × × × × 0 0
PSW *	0 0 0 0 0 0 0 0	TCON *	0 0 0 0 0 0 0 0
SP	0 0 0 0 0 1 1 1	T2CON * +	0 0 0 0 0 0 0 0
DPTR		TH0	0 0 0 0 0 0 0 0
DPH	0 0 0 0 0 0 0 0	TL0	0 0 0 0 0 0 0 0
DPL	0 0 0 0 0 0 0 0	TH1	0 0 0 0 0 0 0 0
P0 *	1 1 1 1 1 1 1 1	TL1	0 0 0 0 0 0 0 0
P1 *	1 1 1 1 1 1 1 1	TH2 ⁺	0 0 0 0 0 0 0 0
P2 *	1 1 1 1 1 1 1 1	TL2 ⁺	0 0 0 0 0 0 0 0
P3 *	1 1 1 1 1 1 1 1	RCAP2H ⁺	0 0 0 0 0 0 0 0
IP * AT89C51	× × × 0 0 0 0 0 0	RCAP2L ⁺	0 0 0 0 0 0 0 0
AT89C52	× × 0 0 0 0 0 0 0 0	SCON *	0 0 0 0 0 0 0 0
IE * AT89C51	0 × × 0 0 0 0 0 0	SBUF	不定
AT89C52	0 × 0 0 0 0 0 0 0 0	PCON COMS	0 × × × 0 0 0 0

× 未定义;
* 可位寻址;
+ 仅 AT89C52 存在。

在表 2-1 中,特殊功能寄存器一共有 22 个。其中单字节的有 ACC,B,PSW,SP,P0,P1,P2,P3,IP,IE,TMOD,TCON,T2CON,T2MOD,SCON,SBUF,PCON 等 17 个。双字节的有 DPTR,TH0-TL0,TH1-TL1,TH2-TL2,RCAP2H-RCAP2L 等 5 个。

特殊功能寄存器都和单片机的相关部件有关,它们分别对应如下:

- ① CPU:ACC,B,PSW;
- ② 存储器:SP,DPTR;
- ③ I/O 端口:P0,P1,P2,P3;
- ④ 中断系统:IP,IE;
- ⑤ 定时器:TMOD,TCON,T2CON,T2MOD,TH0-TL0,TH1-TL1,TH2-TL2,RCAP2H-RCAP2L;
- ⑥ 串行接口:SCON,SBUF;
- ⑦ 电源:PCON。

从这些寄存器的相关部件,可以看出寄存器的大概功能及作用,它们显然是和相关部件的有关操作有密切联系的。

2.3 CPU 定时时序

89 系列单片机和 51 系列单片机一样,在内部有一个振荡器,可以用作 CPU 的时钟源。但是,89 系列单片机也允许采用外部振荡器。外部振荡器产生的信号加到振荡信号的输入端,就可以作为单片机 CPU 的时钟源。

如果采用片内的振荡电路,要在单片机的引脚 XTAL1 和 XTAL2 之间连一个石英晶体或陶瓷谐振器,并接 2 个电容到地,如图 2-17 所示。

有时也可采用外部振荡器。这时,把外部振荡器的信号直接连到 XTAL1 端,XTAL2 端悬空不用,如图 2-18 所示。

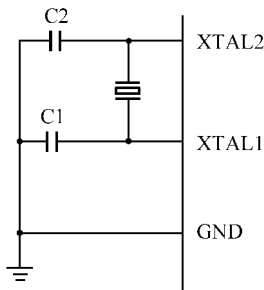


图 2-17 内部振荡器的接法

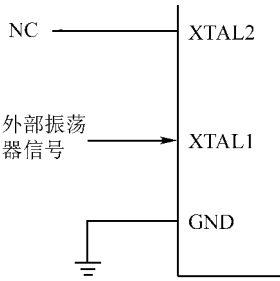


图 2-18 外部振荡器的接法

振荡器产生的信号送到 CPU,作为 CPU 的时钟信号,驱动 CPU 产生执行指令功能的机器周期。

一个机器周期由 6 个状态(S1~S6)组成,每个状态又持续 2 个振荡周期(分为 P1 和 P2)。这样,一个机器周期由 12 个振荡周期组成——S1P1(状态 1 节拍 1)~S6P2(状态 6 节拍 2)。若采用 12 MHz 的晶体振荡器,则每个机器周期恰为 1 μ s。

图 2-19 列举了几种典型指令的取指和执行时序。通常,ALE 在每个机器周期内作用两

次(即使正在执行的指令不需要);第一次在 S1P2 和 S2P1 期间;第二次在 S4P2 和 S5P1 期间。

图 2-19(a)和(b)分别表示单字节单周期和双字节单周期指令的时序。单周期指令的执行从 S1P2 开始,这时操作码被锁存到指令寄存器内。如果是双字节指令,则在同一机器周期的 S4 读第二个字节;如果是单字节指令,则在 S4 仍有读操作,但 CPU 会忽略第二次读进去的字节,且程序计数器 PC 并不加 1。不管什么情况,在 S6P2 结束时都会完成取指操作。

图 2-19(c)给出的是单字节双周期指令的时序,在 2 个机器周期内发生 4 次读操作码的操作。由于是单字节指令,故后面的 3 次读操作都是无效的。

图 2-19(d)给出的是访问外部数据存储器的指令 MOVX 的时序,这也是一条单字节双周期指令。在第一个机器周期的 S5 开始时,发送外部数据存储器的地址,随后读或写数据,读写期间在 ALE 端不输出有效信号。在第二个机器周期,即外部数据存储器已被寻址和选通后,也不产生取指操作。

无论程序存储器是内部的还是外部的,取指和执行的时序都是一样的。执行时间不取决于程序存储器是内部的还是外部的。

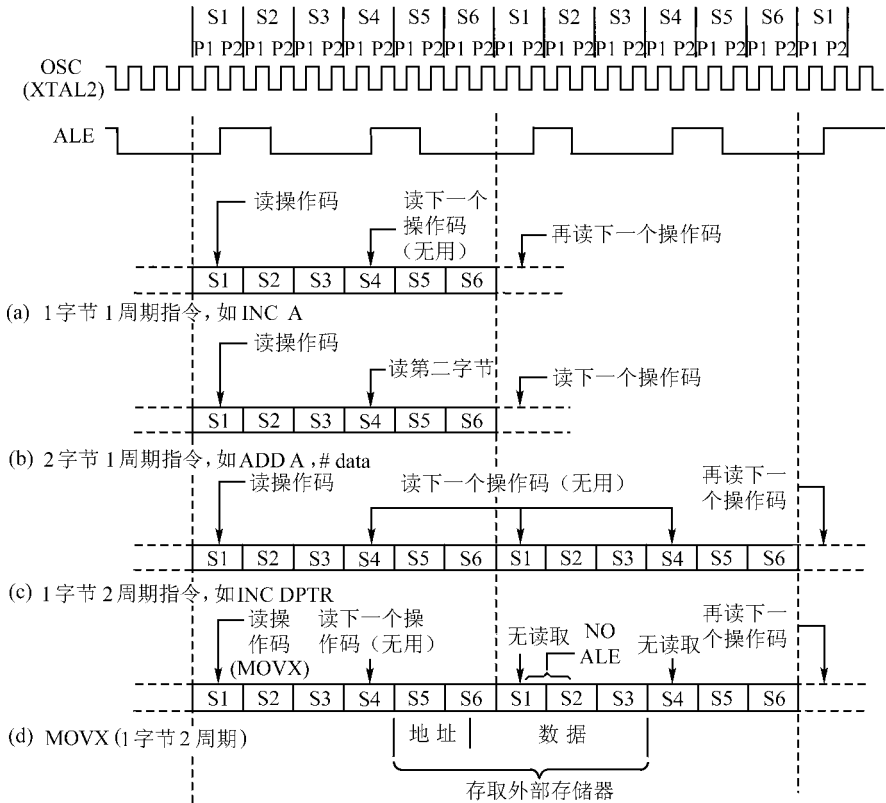


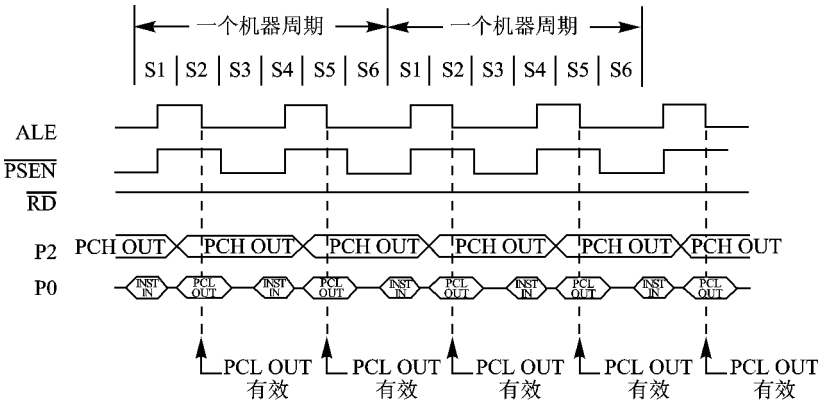
图 2-19 典型指令的取指/执行时序

图 2-20 是与访问外部程序存储器有关的时序图。其中(a)是不访问外部数据存储器,即没有执行 MOVX 类指令情况下的时序;(b)是发生访问外部数据存储器操作(即执行 MOVX 指令)时的时序。CPU 由外部程序存储器取指时,16 位地址的低 8 位 PCL 由 P0 输出;高 8 位

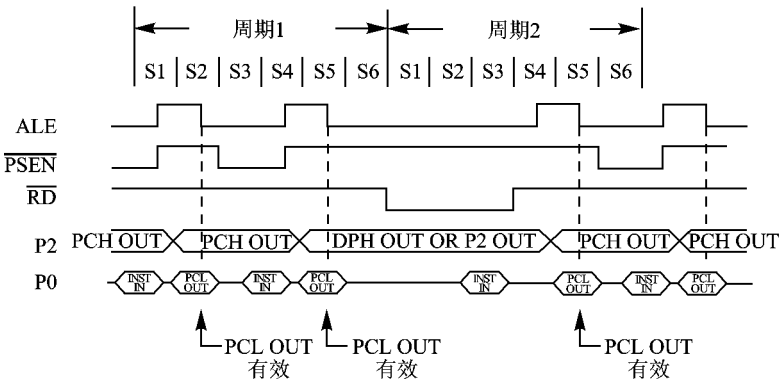
PCH 由 P2 输出;而指令由 P0 输入。

在不执行 MOVX 指令时(图 2-20(a)的情况),P2 端口专门用于输出 PCH;P0 端口作分时复用的地址/数据总线,输出 PCL,输入指令。在这种情况下,每一个机器周期中,ALE(允许地址锁存信号)两次有效。在 ALE 由高变低时,有效地址 PCL 出现在 P0 总线上,低 8 位地址锁存器把地址锁存起来,同时 $\overline{\text{PSEN}}$ 也是每个机器周期两次有效,用于选通外部程序存储器,使指令送到 P0 总线上,由 CPU 取入。当 CPU 执行内部程序存储器中的指令时, $\overline{\text{PSEN}}$ 不会出现,但 ALE 仍然是每个机器周期内两次有效,甚至在非取指操作周期中也是这样。因此它可以被用来作为时钟输出信号。要注意的是,每当访问外部数据存储器(即执行 MOVX 指令)时,将跳过 ALE 脉冲。

执行 MOVX 指令时(图 2-20(b)的情况),时序有些变化。当从外部程序存储器取入的指令是一条 MOVX 指令时,在同一周期的 S5 状态,ALE 由高变低,P0 总线上出现的将不再是有效的程序存储器低 8 位地址,而是有效的数据存储器的地址。在同一周期的 S6 状态将不再出现 $\overline{\text{PSEN}}$ 信号,下一个机器周期的第一个 ALE 有效信号也不再出现,所以在访问外部数据存储器时决不会访问外部程序存储器。当 $\overline{\text{RD}}$ 或 $\overline{\text{WR}}$ 有效时,在 P0 总线上将出现有效的输入(或输出)数据。



(a) 不访问外部数据存储器



(b) 访问外部数据存储器

图 2-20 外部程序存储器的操作时序

2.4 中断系统结构

AT89C51 可提供 5 个中断源:2 个外部中断、2 个定时器溢出中断和 1 个串行口中断。下面介绍一下 AT89C51 的中断结构。

AT89C51 的中断控制系统如图 2-21 所示。

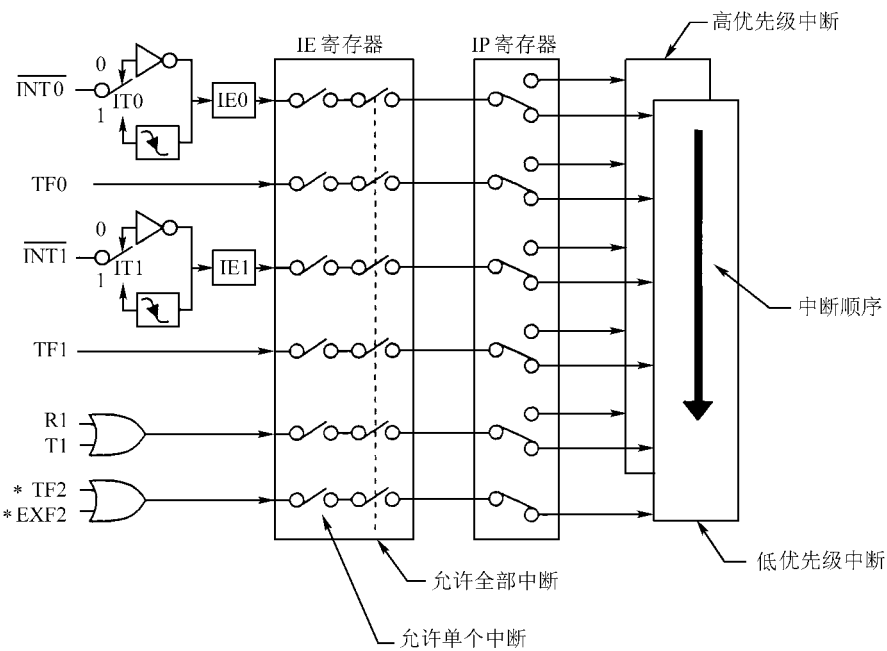


图 2-21 AT89C51 的中断控制系统

从图 2-21 中可知:中断系统主要由中断允许寄存器 IE、中断优先级寄存器 IP、优先级结构和一些逻辑门组成。IE 寄存器用于允许或禁止中断;IP 寄存器用于确定中断源的优先级;优先级结构用于执行中断源的优先排序;有关逻辑门用于输入中断请求信号。

下面分别介绍 IE、IP 寄存器的作用以及中断响应过程。

2.4.1 中断控制寄存器

中断控制寄存器有 2 个:一个是中断允许寄存器;一个是中断优先级寄存器。一个用于启动中断,一个用于确定中断源的优先级。

1. 中断允许寄存器 IE

IE(位于特殊功能寄存器区内)管理每个中断源的启动或禁止。对 IE 的某一位置 1 或清 0,可使对应的中断源允许中断或禁止中断。除了禁止某一个中断源外,IE 中还有一个总禁止位。当该位为“0”时,禁止所有的中断。AT89C51 中的 IE 寄存器各位的定义如图 2-22 所示,现说明如下。

EA(IE.7):总禁止位。若 EA=0,禁止所有的中断;若 EA=1,则每个中断源是允许还是禁止,分别由各自的允许位确定。

(MSB)				(LSB)			
EA	—	—	ES	ET1	EX1	ET0	EX0

图 2-22 中断允许寄存器 IE

- (IE. 6):保留位。
- (IE. 5):保留位。
- ES(IE. 4):串行口中断允许位。若 ES=0,禁止串行口中断。
- ET1(IE. 3):定时器 1 溢出中断允许位。若 ET1=0,禁止定时器 1 中断。
- EX1(IE. 2):外部中断 1 允许位。若 EX1=0,禁止外部中断 1。
- ET0(IE. 1):定时器 0 溢出中断允许位。若 ET0=0,禁止定时器 0 中断。
- EX0(IE. 0):外部中断 0 允许位。若 EX0=0,禁止外部中断 0。

2. 中断优先级寄存器 IP

AT89C51 的中断分为 2 个优先级。在专用寄存器(SFR)区中有一个中断优先级寄存器 IP,IP 中的一位对应于一个中断源,可决定中断源的优先级。IP 寄存器各位的定义如图 2-23 所示。

(MSB)				(LSB)			
—	—	PT2	PS	PT1	PX1	PT0	PX0

图 2-23 中断优先级寄存器 IP

- (IP. 7):保留位。
 - (IP. 6):保留位。
 - PT2(IP. 5):定时器 T2 的中断优先级控制位(仅在 AT89C52 中有)。PT2=1 为高优先级中断。
 - PS(IP. 4):串行口中断优先级设定位。若 PS=1,设定为高优先级。
 - PT1(IP. 3):定时器 1 中断优先级设定位。若 PT1=1,设定为高优先级。
 - PX1(IP. 2):外部中断 1 优先级设定位。若 PX1=1,设定为高优先级。
 - TP0(IP. 1):定时器 0 中断优先级设定位。若 PT0=1,设定为高优先级。
 - PX0(IP. 0):外部中断 0 优先级设定位。若 PX0=1,设定为高优先级。
- 低优先级中断可被高优先级中断所中断,但不能被另一个低优先级中断所中断;高优先级中断不能被任何其他中断源所中断。

当同时收到两个不同优先级的中断请求时,高优先级请求得到服务。如果同时收到几个同一优先级的中断请求,则由内部的查询顺序来决定哪一个请求得到服务,相当于在每个优先级内还同时存在另一个辅助优先结构。

2.4.2 中断响应过程

CPU 在每个机器周期顺序检查每一个中断源,在机器周期的 S6 状态采样,并按优先级处理所有被激活的中断请求。如果没有被下述条件所阻止,将在下一个机器周期的 S1 状态响应激活了的最高优先级中断请求。

- (1) CPU 正在处理相同的或更高优先级的中断；
- (2) 现在的机器周期还不是执行中指令的最后一个机器周期(即正在执行的指令完成前,任何中断请求都得不到响应)；
- (3) 正在执行的是一条中断返回(RETI)或者访问专用寄存器 IE 或 IP 的指令(即在执行这些指令后,不会马上响应中断请求,至少再执行一条其他指令后才会响应)。
- 若存在上述任一种情况,则中断查询结果就被取消。否则,在紧接着的下一个机器周期,中断查询结果将变为有效。

CPU 响应中断时,首先使相应的“优先级激活”触发器置位,以阻止同级和低级的中断。然后执行一条硬件产生的子程序调用(LCALL)指令,使控制转向相应的入口地址(这些入口地址如图 2-4 所示和表 2-3 所列),执行中断服务子程序。

表 2-3 中断向量地址

中断源	向量地址
IE0(外部中断 0)	0003H
TF0(定时器 0 溢出中断)	000BH
IE1(外部中断 1)	0013H
TF1(定时器 1 溢出中断)	001BH
RI+TI(串行口中断)	0023H
TF2+EXF2(定时器 2 中断) *	002BH

* 只有 AT89C52 存在。

硬件中断服务子程序调用时,把当时程序计数器 PC 的内容压入堆栈(但不自动保存程序状态字 PSW 和任何其他寄存器的内容),并把被响应的中断服务子程序的入口地址装入 PC 中。

中断服务子程序的最后一条指令应是 RETI(中断返回)。RETI 指令将清除“优先级激活”触发器(该触发器在响应中断时被置位),然后从堆栈中弹出顶上的 2 个字节(下一条指令地址)装入到 PC 中。CPU 从原来打断处重新执行被中断的程序。

AT89C51 只提供了两个中断优先级,但在一些应用场合需要更多的中断优先级。这种情况下,可采用软件模拟的方法为 AT89C51 单片机增加一个中断优先级,从而实现 3 级中断服务程序嵌套。

首先,在中断优先级寄存器 IP 中定义两个中断优先级——高优先级和低优先级,然后,在低优先级的中断服务程序中编写如下程序即可实现 3 级中断服务程序的嵌套。

```
PUSH    IE
MOV     IE, #MASK
CALL    LABEL
* * * * *
(执行中断服务程序)
* * * * *
POP     IE
```

RET
LABEL

在主程序中,一旦有任何低优先级的中断被响应,在中断服务程序中需在中断允许寄存器 IE 中重新置入一个屏蔽字以屏蔽当前中断,然后调用指令 CALL,通过调用 LABEL 来执行 RETI 指令。其目的是模拟一次中断返回,从而清除原来置位的低优先级状态触发器,并开始执行中断服务程序。此时,中断服务程序既可以被低优先级的中断源所中断,也可以被高优先级的中断源所中断。在以上中断服务程序结束前,还要恢复 IE 寄存器中的初始值,末尾用一条 RET 指令(注意,不是 RETI)来终止该服务程序。

上段附加的程序使低级中断服务程序的执行时间增加 $10\ \mu\text{s}$ (在 12 MHz 晶振频率条件下)。

89 系列单片机在响应中断时,需要采取以下 3 个步骤。

- (1) 将 IE 寄存器中 EA(允许所有中断)位置 1。
- (2) 将 IE 寄存器中相应的独立中断允许位置 1。
- (3) 中断服务程序从相应的向量地址(如表 2-3 所列)开始执行。

外部中断引脚 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ (P3.2 和 P3.3)必须设置为 1。它们的激活方式有两种:一种是电平激活;另一种是边沿激活。这两种方式由 TCON 寄存器中的中断方式位 IT1 或 IT0 来控制。若 $\text{IT}_x=0$,则为电平激活;若 $\text{IT}_x=1$,则为边沿激活。

2.5 特殊功能寄存器 SFR

特殊功能寄存器也称专用寄存器,是具有特殊功能的所有寄存器的集合,简称 SFR (Special Function Register)。特殊功能寄存器共含有 22 个不同寄存器。它们的地址分配在 $80\text{H}\sim\text{FFH}$ 中,即在 RAM 地址中。这些寄存器的名称和地址如图 2-16 所示和表 2-1 所列。

虽然特殊功能寄存器地址在 $80\text{H}\sim\text{FFH}$ 之中,但在 $80\text{H}\sim\text{0FFH}$ 的地址单元中,不是所有的单元都被特殊功能寄存器占用,未被占用的单元,其内容是不确定的。如对这些单元进行读操作,得到的是一些随机数,而写入则无效。所以用户编程时不应该将数据写入这些未确定的地址单元,它们是公司留待将来开发新产品时使用的。特殊功能寄存器组所包含的各特殊功能寄存器介绍如下。

1. 累加器 ACC

累加器 ACC 用于存放参加运算的操作数和运算结果,在指令系统中用 A 表示累加器。

2. B 寄存器

B 寄存器是运算器中的一个工作寄存器,用于存放乘法和除法运算中的操作数及运算结果,在其他的运算中,可作为中间结果寄存器使用。

3. 程序状态字寄存器 PSW

程序状态字寄存器 PSW 包含了各种程序状态信息。PSW 的格式及各标志的含义及功能定义如图 2-24 所示。

CY:进位标志。

AC:辅助进位标志(用于 BCD 码操作)。

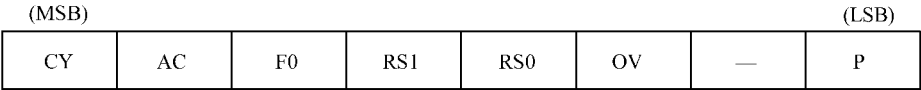


图 2-24 程序状态字寄存器 PSW

F0:0 标志(供用户使用的软件标志)。
RS1,RS0:工作寄存器组选择控制位,由软件定义工作寄存器组选择如表 2-4 所列。

表 2-4 工作寄存器组选择

RS1	RS0	寄存器组	地 址
0	0	0	00H~07H
0	1	1	08H~0FH
1	0	2	10H~17H
1	1	3	18H~1FH

OV:溢出标志。
—:由用户定义的标志位。
P:奇偶标志。此为累加器的奇偶标志位。如果 ACC 中 1 的个数为偶数,P=0;否则 P=1。此位在每个指令周期后由硬件置位或清 0。

4. 堆栈指针 SP

堆栈指针寄存器 SP 字长为 8 位。当数据被压入堆栈前,指针 SP 加 1;数据从堆栈中弹出后,指针 SP 减 1。堆栈区可设置在芯片的内部存储器的任意区域,复位后,堆栈指针 SP 的值为 07H,则堆栈区在 08H 开始的区域。

5. 数据指针 DPTR

DPTR 为 16 位的数据指针寄存器,由两个 8 位的寄存器 DPH 和 DPL 组成,可存放一个 16 位的地址值。当 CPU 访问外部数据存储器的时,就用 DPTR 作地址指针,存放外部存储器的地址。CPU 也可单独对 DPH,DPL 操作,即将 DPTR 分成两个寄存器使用。

6. I/O 端口 P0~P3

专用寄存器组中的 P0,P1,P2 和 P3 分别为各 I/O 端口的锁存器,用于 I/O 端口的读、写操作。

7. 串行口数据缓冲器 SBUF

89 系列单片机的串行通信都是经数据缓冲器 SBUF 发送和接收的。实际上在 SBUF 中有两个独立的寄存器:一个是发送缓冲器;另一个是接收缓冲器。当数据被写入 SBUF 时,就被送到发送缓冲器中准备发送;而从 SBUF 读出数据时,数据一定是从接收缓冲器中送来的。

8. 定时器寄存器

寄存器对 (TH0,TL0)、(TH1,TL1)、(TH2,TL2)分别为定时器/计数器 T0,T1,T2 的 16 位计数器寄存器,也可以单独作为一个 8 位的计数器寄存器使用。

9. 捕捉寄存器 RCAP2

16 位的捕捉寄存器由两个 8 位寄存器 RCAP2H 和 RCAP2L 组成,用于定时器 T2 的捕

捉工作方式。当 AT89C52 的外部输入端 T2EX 发生负跳变时,就将 TH2 和 TL2 的内容锁存入 RCAP2H 和 RCAP2L 中;另外,也可在 T2 的常数自动重装入方式下,这时,RCAP2 用来存放 16 位的常数值。

10. 控制寄存器

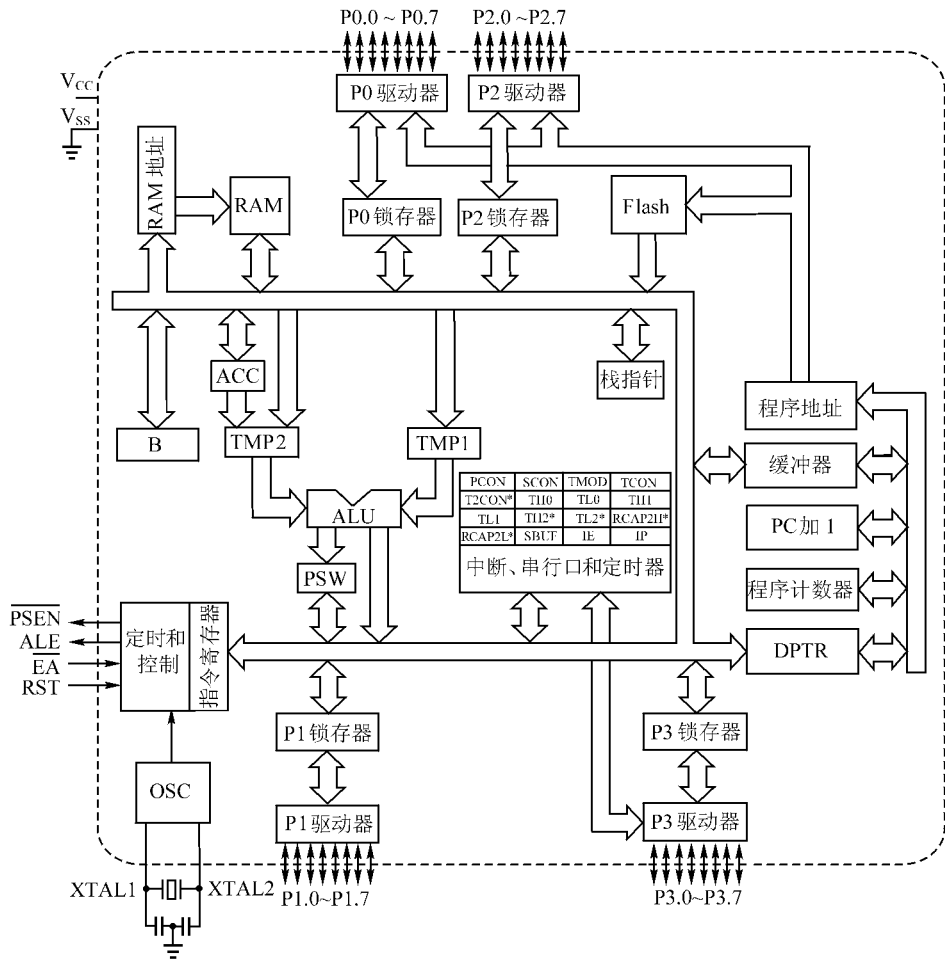
特殊功能寄存器组中还包含有多个控制寄存器 IP, IE, TMOD, TCON, T2CON, T2MOD, SCON, PCON。它们分别用于中断系统、定时器/计数器、串行通信的控制。

2.6 89 系列典型单片机结构

89 系列中,典型的单片机有 AT89C51, AT89C2051, AT89S8252 等。在这一节,以这些单片机为典型作简要的介绍,包括它们的主要性能、结构框图以及引脚功能的说明。

2.6.1 AT89C51 单片机

AT89C51 的结构框图如图 2-25 所示。



* 只在 AT89C52 中有。

图 2-25 AT89C51 结构框图

AT89C51 单片机还有一种低电压的型号,即 AT89LV51,除了电压范围有区别之外,其余特性与 AT89C51 完全一致。

AT89C51/LV51 是一种低功耗/低电压、高性能的 8 位单片机。片内带有一个 4 KB 的 Flash 可编程、可擦除只读存储器(EPROM)。它采用了 CMOS 工艺和 ATMEL 公司的高密度非易失性存储器(NURAM)技术,而且其输出引脚和指令系统都与 MSC-51 兼容。片内的 Flash 存储器允许在系统内改编程序或用常规的非易失性存储器编程器来编程。因此 AT89C51/LV51 是一种功能强、灵活性高,且价格合理的单片机,可方便地应用在各种控制领域。

1. 主要性能

- 4 KB 可改编程序 Flash 存储器(可经受 1 000 次的写入/擦除)。
- 全静态工作:0 Hz~24 MHz。
- 3 级程序存储器保密。
- 128×8 字节内部 RAM。
- 32 条可编程 I/O 线。
- 2 个 16 位定时器/计数器。
- 6 个中断源。
- 可编程串行通道。
- 片内时钟振荡器。

另外,AT89C51 是用静态逻辑来设计的,其工作频率可下降到 0 Hz,并提供两种可用软件来选择的省电方式——空闲方式(Idle Mode)和掉电方式(Power Down Mode)。在空闲方式中,CPU 停止工作,而 RAM、定时器/计数器、串行口和中断系统都继续工作。在掉电方式中,片内振荡器停止工作,由于时钟被“冻结”,使一切功能都暂停,故只保存片内 RAM 中的内容,直到下一次硬件复位为止。

2. 引脚功能说明

图 2-26 是 AT89C51/LV51 的引脚结构图,有双列直插封装(DIP)方式和方形封装方式。下面分别叙述这些引脚的功能。

(1) 主电源引脚

- ① V_{CC} :电源端。
- ② GND:接地端。

(2) 外接晶体引脚 XTAL1 和 XTAL2

① XTAL1:接外部晶体的一个引脚。在单片机内部,它是构成片内振荡器的反相放大器的输入端。当采用外部振荡器时,该引脚接收振荡器的信号,即把此信号直接接到内部时钟发生器的输入端。

② XTAL2:接外部晶体的另一个引脚。在单片机内部,它是上述振荡器的反相放大器的输出端。采用外部振荡器时,此引脚应悬浮不连接。

(3) 控制或其他电源复用引脚 $\overline{RST}$, $\overline{ALE/PROG}$, $\overline{PSEN}$ 和 $\overline{EA}/V_{PP}$

① $\overline{RST}$:复位输入端。当振荡器运行时,在该引脚上出现两个机器周期的高电平将使单片机复位。

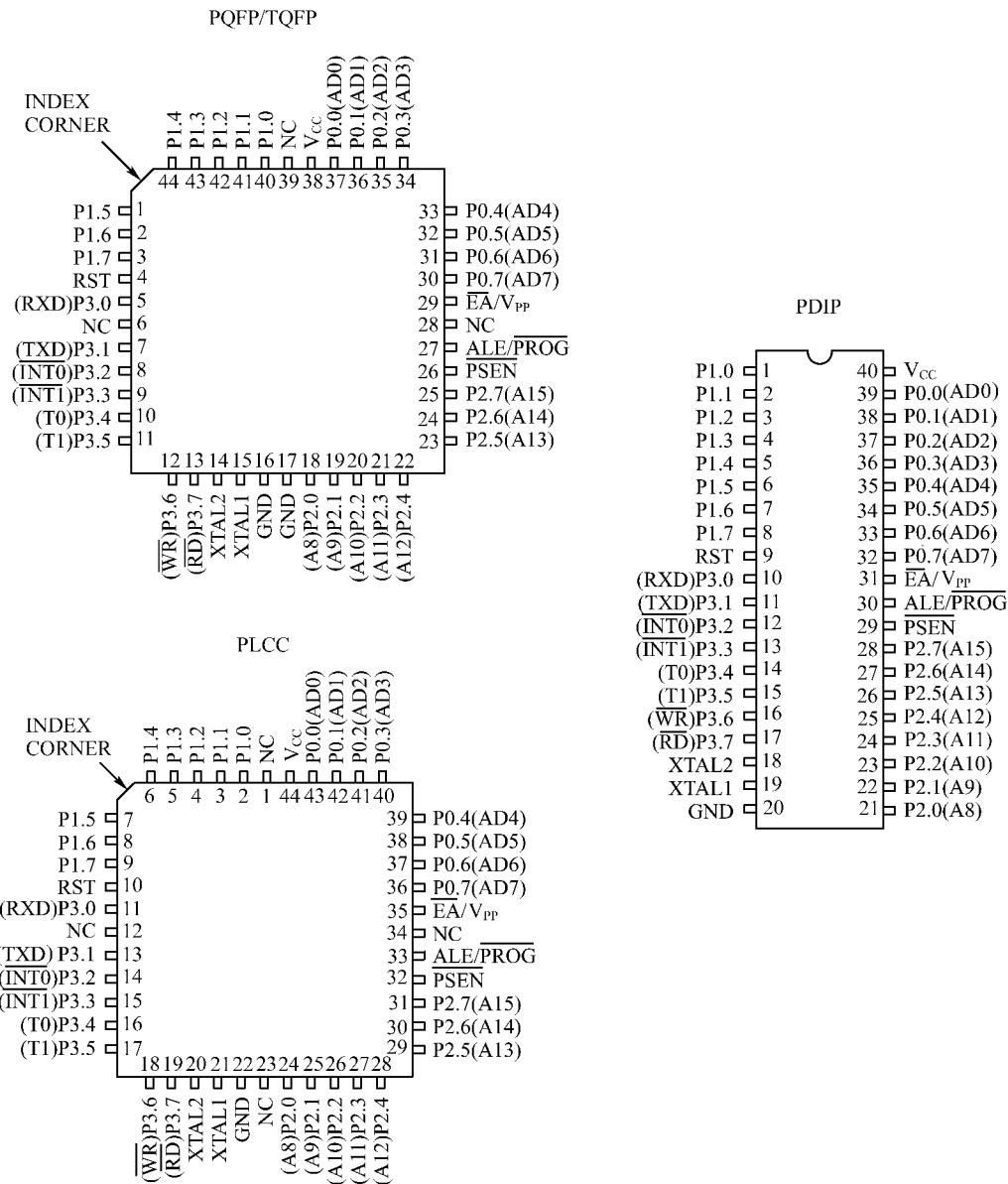


图 2-26 AT89C51/LV51 的引脚结构

② $\overline{ALE}/\overline{PROG}$:当访问外部存储器时,ALE(地址锁存允许)的输出用于锁存地址的低字节。即使不访问外部存储器,ALE 端仍以不变的频率(此频率为振荡器频率的 1/6)周期性地出现正脉冲信号。因此,它可用作对外输出的时钟,或用于定时目的。然而要注意的是:每当访问外部数据存储器时,将跳过一个 ALE 脉冲。

在对 Flash 存储器编程期间,该引脚还用于输入编程脉冲($\overline{PROG}$)。

如果需要的话,通过对专用寄存器(SFR)区中 8EH 单元的 D0 位置数,可禁止 ALE 操作。该位置数后,只有在执行一条 MOVX 或 MOVC 指令期间,ALE 才会被激活。另外,该引脚会被微弱拉高,单片机执行外部程序时,该设定禁止 ALE 位无效。

③ $\overline{\text{PSEN}}$: 程序存储允许($\overline{\text{PSEN}}$)输出是外部程序存储器的读选通信号。当 AT89C51/LV51 由外部程序存储器取指令(或常数)时,每个机器周期两次 $\overline{\text{PSEN}}$ 有效(即输出 2 个脉冲)。但在此期间内,每当访问外部数据存储器时,这两次有效的 $\overline{\text{PSEN}}$ 信号将不出现。

④ $\overline{\text{EA}}/\text{V}_{\text{PP}}$: 外部访问允许端。要使 CPU 只访问外部程序存储器(地址为 0000H~FFFFH),则 $\overline{\text{EA}}$ 端必须保持低电平(接到 GND 端)。然而要注意的是,如果保密位 LB1 被编程,复位时在内部会锁存 $\overline{\text{EA}}$ 端的状态。

当 $\overline{\text{EA}}$ 端保持高电平(接 V_{CC} 端)时,CPU 则执行内部程序存储器中的程序。

在 Flash 存储器编程期间,该引脚也用于施加 12 V 的编程允许电源 V_{PP} (如果选用 12 V 编程)。

(4) 输入/输出引脚 P0.0~P0.7, P1.0~P1.7, P2.0~P2.7 和 P3.0~P3.7

① P0 端口(P0.0~P0.7): P0 是一个 8 位漏极开路型双向 I/O 端口。作为输出口用时,每位能以吸收电流的方式驱动 8 个 TTL 输入,对端口写 1 时,又可作高阻抗输入端用。

在访问外部程序和数据存储器时,它是分时多路转换的地址(低 8 位)/数据总线,在访问期间激活了内部的上拉电阻。

在 Flash 编程时,P0 端口接收指令字节;而在校验程序时,则输出指令字节。验证时,要求外接上拉电阻。

② P1 端口(P1.0~P1.7): P1 是一个带有内部上拉电阻的 8 位双向 I/O 端口。P1 的输出缓冲器可驱动(吸收或输出电流方式)4 个 TTL 输入。对端口写 1 时,通过内部的上拉电阻把端口拉到高电位,这时可用作输入口。P1 作输入口使用时,因为有内部的上拉电阻,那些被外部信号拉低的引脚会输出一个电流(I_{IL})。

在对 Flash 编程和程序校验时,P1 接收低 8 位地址。

③ P2 端口(P2.0~P2.7): P2 是一个带有内部上拉电阻的 8 位双向 I/O 端口。P2 的输出缓冲器可驱动(吸收或输出电流方式)4 个 TTL 输入。对端口写 1 时,通过内部的上拉电阻把端口拉到高电位,这时可用作输入口。P2 作输入口使用时,因为有内部的上拉电阻,那些被外部信号拉低的引脚会输出一个电流(I_{IL})。

在访问外部程序存储器和 16 位地址的外部数据存储器(如执行 $\text{MOVX}@\text{DPTR}$ 指令)时,P2 送出高 8 位地址。在访问 8 位地址的外部数据存储器(如执行 $\text{MOVX}@\text{RI}$ 指令)时,P2 口引脚上的内容(就是专用寄存器(SFR)区中 P2 寄存器的内容),在整个访问期间不会改变。

在对 Flash 编程和程序校验期间,P2 也接收高位地址和一些控制信号。

④ P3 端口(P3.0~P3.7): P3 是一个带内部上拉电阻的 8 位双向 I/O 端口。P3 的输出缓冲器可驱动(吸收或输出电流方式)4 个 TTL 输入。对端口写 1 时,通过内部的上拉电阻把端口拉到高电位,这时可用作输入口。P3 作输入口使用时,因为有内部的上拉电阻,那些被外部信号拉低的引脚会输出一个电流(I_{IL})。

在 AT89C51 中,P3 端口还用于一些复用功能。

复用功能如表 2-5 所列。

在对 Flash 编程或程序校验时,P3 还接收一些控制信号。

表 2-5 P3 各端口引脚与复用功能表

端口引脚	复用功能
P3.0	RXD(串行输入口)
P3.1	TXD(串行输出口)
P3.2	$\overline{\text{INT0}}$ (外部中断 0)
P3.3	$\overline{\text{INT1}}$ (外部中断 1)
P3.4	T0(定时器 0 的外部输入)
P3.5	T1(定时器 1 的外部输入)
P3.6	$\overline{\text{WR}}$ (外部数据存储器写选通)
P3.7	$\overline{\text{RD}}$ (外部数据存储器读选通)

2.6.2 AT89C2051 单片机

AT89C2051 是一带有 2 KB Flash 可编程、可擦除只读存储器(EEPROM)的低压、高性能 8 位 CMOS 微型计算机,如图 2-27 所示。它采用 ATMEL 的高密非易失存储技术制造,并和工业标准 MCS-51 指令集和引脚结构兼容。通过在单块芯片上组合通用的 CPL1 和 Flash 存储器,使 AT89C2051 成为一强劲的微型计算机。它为许多嵌入式控制应用提供了高度灵活和成本低的解决办法。

AT89C2051 提供以下标准功能:2 KB Flash 存储器;128 字节 RAM;15 条 I/O 引线;2 个 16 位定时器/计数器;1 个 5 向量 2 级中断结构;1 个全双工串行口;1 个精密模拟比较器以及片内振荡器和时钟电路。此外,AT89C2051 是用可降到 0 频率的静态逻辑操作设计的,并支持两种可选的软件节电工作方式。空闲方式停止 CPU 工作,但允许 RAM、定时器/计数器、串行口和中断系统继续工作。掉电方式保存 RAM 内容,但振荡器停止工作,并禁止所有其他部件的工作直到下一个硬件复位。

1. 主要性能

- 和 MCS-51 产品兼容。
- 2 KB 可重编程 Flash 存储器。
- 耐久性:1 000 次写/擦除。
- 2.7~6 V 的操作范围。
- 全静态操作:0 Hz~24 MHz。
- 2 级加密程序存储器。
- 128×8 位内部 RAM。
- 15 条可编程 I/O 引线。
- 2 个 16 位定时器/计数器。
- 6 个中断源。
- 可编程串行 UART 通道。
- 直接 LED 驱动输出。

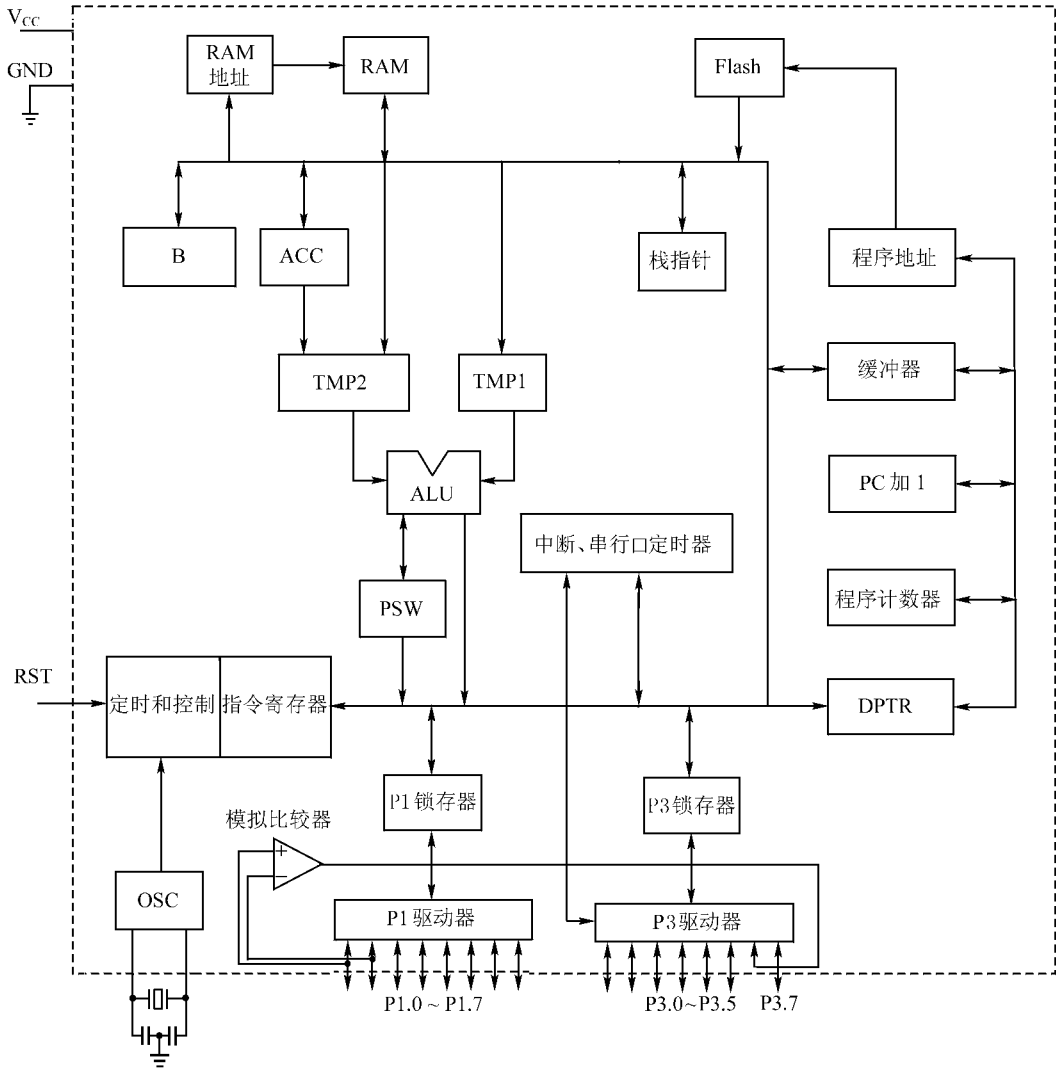


图 2-27 AT89C2051 结构框图

- 片内模拟比较器。
- 低功耗空载和掉电方式。

2. 引脚功能说明

AT89C2051 的引脚结构如图 2-28 所示。

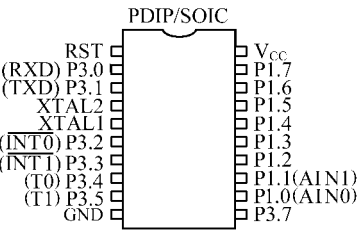


图 2-28 AT89C2051 的引脚结构

- (1) V_{CC}: 电源端。
- (2) GND: 接地端。

(3) P1 口: P1 口是一 8 位双向 I/O 口。引脚 P1.2~P1.7 提供内部上拉电阻。P1.0 和 P1.1 要求外部上拉电阻。P1.0 和 P1.1 还分别作为片内精密模拟比较器的同相输入(AIN0)和反相输入(AIN1)。P1 口输出缓冲器可吸收 20 mA 电流,并能直接驱动 LED

显示。当 P1 口引脚写入“1”时,可用作输入端。当引脚 P1.2~P1.7 用作输入端并被外部拉低时,将因内部的上拉电阻而输出电流(I_{IL})。

P1 口还在 Flash 编程和程序校验期间接收代码数据。

(4) P3 口:P3 口的 P3.0~P3.5,P3.7 是带有内部上拉电阻的 7 个双向 I/O 引脚。P3.6 用于固定输入片内比较器的输出信号,并且作为一通用 I/O 引脚而不可访问。P3 口缓冲器可吸收 20 mA 电流。当 P3 口引脚写入“1”时,它们被内部上拉电阻拉高并可用作输入端。用作输入端时,被外部拉低的 P3 口引脚将用上拉电阻而输出电流(I_{IL})。

P3 口还用于实现 AT89C2051 的各种功能,如表 2-6 所列。

表 2-6 P3 口的功能

口引脚	功 能
P3.0	RXD(串行输入端口)
P3.1	TXD(串行输出端口)
P3.2	$\overline{INT0}$ (外中断 0)
P3.3	$\overline{INT1}$ (外中断 1)
P3.4	T0(定时器 0 外部输入)
P3.5	T1(定时器 1 外部输入)

P3 口还接收一些用于 Flash 存储器编程和程序校验的控制信号。

(5) RST:复位输入。RST 一旦变成高电平,所有的 I/O 引脚就复位到“1”。当振荡器正在运行时,持续给出 RST 引脚两个机器周期的高电平便可完成复位。每一个机器周期需 12 个振荡器或时钟周期。

(6) XTAL1:作为振荡器反相放大器的输入和内部时钟发生器的输入。

(7) XTAL2:作为振荡器反相放大器的输出。

2.6.3 AT89S8252 单片机

AT89S8252 带有 8 KB 可向下装载 Flash 编程可擦除只读存储器和 2 KB EEPROM。该器件采用 ATMEL 的高密度非易失性存储器技术制造,与工业上标准的 80C51 的指令系统及引脚兼容,片内可向下装载 Flash,允许程序存储器在系统内通过 SPI 串行口改写或用通用的非易失性存储器编程器改写。通过把通用的 8 位 CPU 与可向下装载的 Flash 集成在一个芯片上,AT89S8252 便成为一个高效的微型计算机。它的应用范围广,可用于解决复杂的控制问题,且成本较低。

AT89S8252 提供了 8 KB 可向下装载 Flash;2 KB 的 EEPROM;256 字节 RAM;32 条 I/O 引线;可编程序控制定时器;双数据指针;3 个 16 位定时器/计数器;7 向量 2 级中断;1 个全双工串行口;片内振荡器和时钟电路等标准功能。此外,AT89S8252 设有静态逻辑,用于运行到零频率,并支持软件选择的两种节电运行方式。空闲方式使 CPU 停止工作,而允许 RAM、定时器/计数器、串行口和中断系统继续工作。掉电方式下,片内振荡器停止工作,由于时钟被冻结,一切功能都停止,只有片内 RAM 的内容被保存,直到中断或硬件复位才恢复正

常工作。

可向下装载 Flash,每次可以更改一个字节,并且通过 SPI 串行口可对其访问。保持复位信号有效来强制 SPI 总线进入从输入方式。如果加密位 3 不起作用,则允许对程序存储器进行写或读。

AT89S8252 的结构框图如图 2-29 所示。

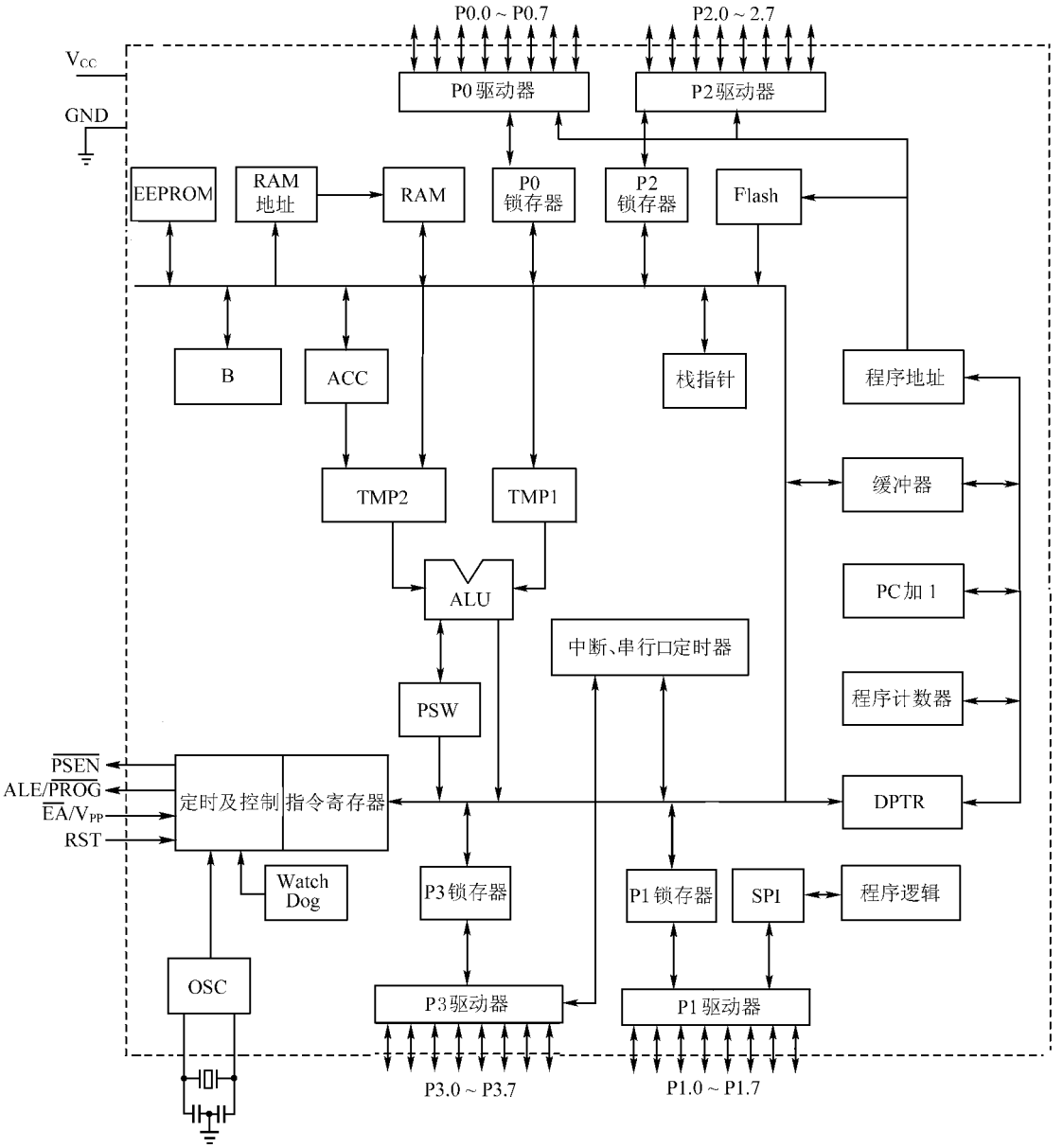


图 2-29 AT89S8252 的结构框图

1. 主要性能

- 与 MCS-51 产品兼容。
- 8 KB 片内可向下装载重编程的 Flash 存储器。SPI 串行口用于编程向下装载。Flash

存储器可写/擦 1 000 次。

- 2 KB EEPROM,可写/擦 100 000 次。
- 全静态操作:0 Hz~24 MHz。
- 3 级程序存储器加密。
- 256 字节内部 RAM。
- 32 条可编程 I/O 线。
- 3 个 16 位定时器/计数器。
- 9 个中断源。
- 可编程的 UART 串行口。
- SPI 串行口。
- 低功耗空闲方式和掉电方式。
- 通过中断终止掉电方式。
- 可编程程序控制定时器。
- 双数据指针。

2. 引脚功能说明

(1) 主电源引脚

- ① V_{CC}:电源端。
- ② GND:接地端。

(2) 输入/输出引脚

① P0 端口(P0.0~P0.7):P0 是一个 8 位漏极开路的双向 I/O 口。在作为输出口时,每条引脚可以带动 8 个 TTL 输入负载。当把“1”写入 P0 时,则它的引脚可用作高阻抗输入。当对外部程序或数据存储器进行存取时,P0 可用作多路复用的低字节地址/数据总线。在对 Flash 存储器进行编程时,P0 用于接收代码字节;在校验时,则输出代码字节;这时,需要外部上拉电阻。

② P1 端口(P1.0~P1.7):P1 是 8 位双向 I/O 口,且带内部上拉电阻。P1 的输出缓冲器可以带 4 个 TTL 负载。当把“1”写入 P1 时,其引脚的电平会由内部上拉电阻拉高,故可以用作输入引脚。当作为输入时,由于内部上拉电阻的作用,那些被输入信号拉低的引脚会输出电流 I_{IL}。另外,P1.0 和 P1.1 引脚可以配置为定时器/计数器 2 的外部计数输入(P1.0/T2)和触发输入(P1.1/T2EX),如表 2-7 所列。在 Flash 存储器编程和校验时,P1 同样接收低字节地址。

表 2-7 P1 中复用的引脚

引 脚	复用功能
P1.0	T2(定时器/计数器 2 的外部计数输入) 时钟输出
P1.1	T2EX(定时器/计数器 2 捕获触发和双向控制)
P1.5	MOSI(用于在系统编程)
P1.6	MISO(用于在系统编程)
P1.7	SCK(用于在系统编程)

③ P2 端口(P2.0~P2.7):P2 也是一个内部带上拉电阻的双向 I/O 口。它的输入输出特性和 P1 相同。所不同的是在用 16 位地址存取外部程序或数据存储器时,用于发送高字节地址。当用 8 位地址存取时,P2 发送 P2 特殊功能寄存器的内容。在 Flash 存储器编程或校验时,P2 接收高位地址及一些控制信号。

④ P3 端口(P3.0~P3.7):P3 的输入输出特性和 P1 相同。P3 另外还有两个作用:一个是在 Flash 存储器编程时可接收一些控制信号;另一个是用于 AT89S8252 特有的复用功能,如表2-8所列。

表 2-8 P3 的特有复用功能

引 脚	复用功能
P3.0	RXD(串行输入口)
P3.1	TXD(串行输出口)
P3.2	$\overline{\text{INT0}}$ (外部中断 0)
P3.3	$\overline{\text{INT1}}$ (外部中断 1)
P3.4	T0(定时器 0 外部输入)
P3.5	T1(定时器 1 外部输出)
P3.6	$\overline{\text{WR}}$ (外部数据存储器写信号)
P3.7	$\overline{\text{RD}}$ (外部数据存储器读信号)

图 2-30 是 AT89S8252 的引脚结构图,有双列直插式和方形封装方式,可用于不同的场合。

(3) 控制信号引脚

① RST:复位输入端。在振荡器运行时,RST 的输入高电平有 2 个机器周期就会对单片机复位。在 Watch Dog 溢出时,这个引脚会保持高电平达 96 个振荡器周期,在SFR AUXR(地址 8EH)寄存器中的 DISRTO 位可以用于屏蔽这种作用(AUXR 是特殊功能寄存器 SFR 中的辅助寄存器)。

② ALE/ $\overline{\text{PROG}}$:地址锁存允许信号。在存取外部存储器时,这个输出信号用于锁存低字节地址。在对 Flash 存储器编程时,这条引脚用于输入编程脉冲 $\overline{\text{PROG}}$ 。一般情况下,ALE 是振荡器频率的 6 分频信号,可用于外部定时或时钟。但是,在对外部数据存储器每次存取中,会跳过一个 ALE 脉冲。在需要时,可以把地址 8EH 中的 SFR 寄存器的 0 位置为“1”,从而屏蔽 ALE 的工作;而只有在 MOVX 或 MOVC 指令执行时 ALE 才被激活。在单片机处于外部执行方式时,对 ALE 屏蔽位置“1”并不起作用。

③ $\overline{\text{PSEN}}$:程序存储器允许信号。它用于读外部程序存储器。当 AT89S8252 在执行来自外部存储器的指令时,每个机器周期 $\overline{\text{PSEN}}$ 被激活 2 次。在对外部数据存储器的每次存取中, $\overline{\text{PSEN}}$ 的 2 次激活会被跳过。

④ $\overline{\text{EA}}/\text{V}_{\text{PP}}$:外部存取允许信号。为了确保单片机从地址为 0000H~FFFFH 的外部程序存储器中读取代码,故要把 $\overline{\text{EA}}$ 接到 GND 端,即地端。但是,如果锁定位 1 被编程,则 $\overline{\text{EA}}$ 在复位时被锁存。当执行内部程序时, $\overline{\text{EA}}$ 应接到 V_{CC} 。在对 Flash 存储器编程时,这条引脚接收

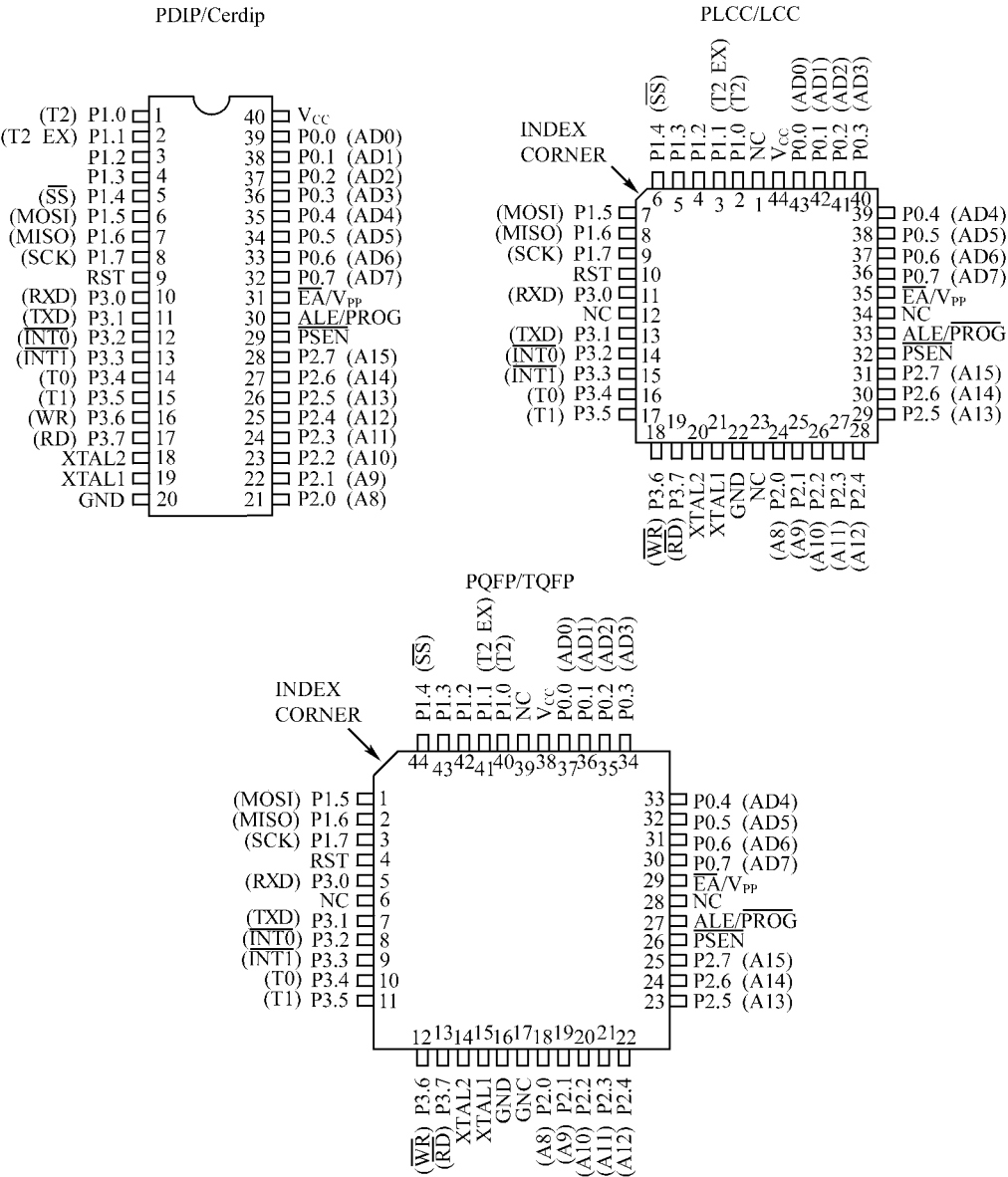


图 2 - 30 AT89S8252 的引脚结构

12 V 编程电压 V_{PP}。

(4) 振荡器引脚

- ① XTAL1: 振荡器的反相放大器输入, 内部时钟工作电路的输入。
- ② XTAL2: 振荡器的反相放大器输出。

第三章 接口部件结构及功能

接口部件是指在 89 系列单片机内部的专门部件,用于进行信号的输入输出。接口部件有 I/O 端口、定时器、串行接口、中断系统、复位电路和振荡器等。这些部件的作用分别如下。

I/O 端口:用于并行数据的输入输出。

串行接口:用于串行数据的输入输出。

定时器:用于外部计数和内部定时。

中断系统:用于接收外部中断请求和内部中断请求,产生相应中断处理。

复位电路:用于接收外部复位信号,使单片机复位。

振荡器:用于接收外部振荡信号或产生内部振荡信号。

在这一章中分别对上述接口部件的结构和功能进行介绍。

3.1 I/O 端口的结构及功能

在 AT89C51 和 AT89C52 中有 4 个双向 I/O 端口 P0~P3 口。每个端口都由锁存器、输出驱动器、输入缓冲器组成。当 CPU 控制系统与外部设备交换信息时,都是通过端口锁存器进行的。4 个 I/O 端口都可作输入输出使用,其中 P0 口和 P2 口通常用于对外部存储器的访问。在这种应用方式下,把 P0 口作为地址/数据总线口使用,分时输出外部存储器的低 8 位地址和传送 8 位数据 D0~D7。当扩展的外部存储器的地址为 16 位时,就把 P2 口用作地址总线口使用,输出高 8 位地址。另外,P2 口的引脚状态会被锁存入 P2SFR 中。

P3 口和 P1 口中的 2 个引脚(仅限于 AT89C52)都有复用功能,除了可用作普通的 I/O 口外,还可以提供其他的使用功能,如表 3-1 所列。

表 3-1 P3 口复用功能定义

端口引脚	复用功能
P1.0 *	T2(定时器 T2 外部输入)
P1.1 *	T2EX(定时器 T2 的捕捉/常数重装触发)
P3.0	RXD(串行接收器)
P3.1	TXD(串行发送端)
P3.2	$\overline{\text{INT0}}$ (外部中断 0 输入)
P3.3	$\overline{\text{INT1}}$ (外部中断 1 输入)
P3.4	T0(定时器 T0 外部输入)
P3.5	T1(定时器 T1 外部输入)
P3.6	$\overline{\text{WR}}$ (外部数据存储器写脉冲输出线)
P3.7	$\overline{\text{RD}}$ (外部数据存储器读脉冲输出线)

* P1.0 和 P1.1 的复用功能仅在 AT89C52 中有。

端口的复用功能由端口的 SFR 寄存器控制。如使用端口的复用功能,端口锁存器的对应位应为 1。

3.1.1 I/O 端口的结构

在图 3-1(a),(b),(c),(d)中,分别给出了 4 个 8 位 I/O 端口 P0,P1,P2 和 P3 的 1 位结构图。端口由锁存器、输出驱动器、输入缓冲器组成。锁存器由 D 触发器构成。当系统执行写端口操作时,CPU 通过内部总线把数据写入锁存器。而读端口却有两种方式:当执行的是读锁存器指令时,CPU 发出读锁存器信号,此时锁存器状态由触发器的 Q 端经锁存器上面的三态输入缓冲器送入内部总线;如果执行的是读端口引脚的指令,则 CPU 发出的是读引脚控制信号,直接读取端口引脚上的外部输入信息,此时引脚状态经锁存器下面的三态输入缓冲器送入内部总线。

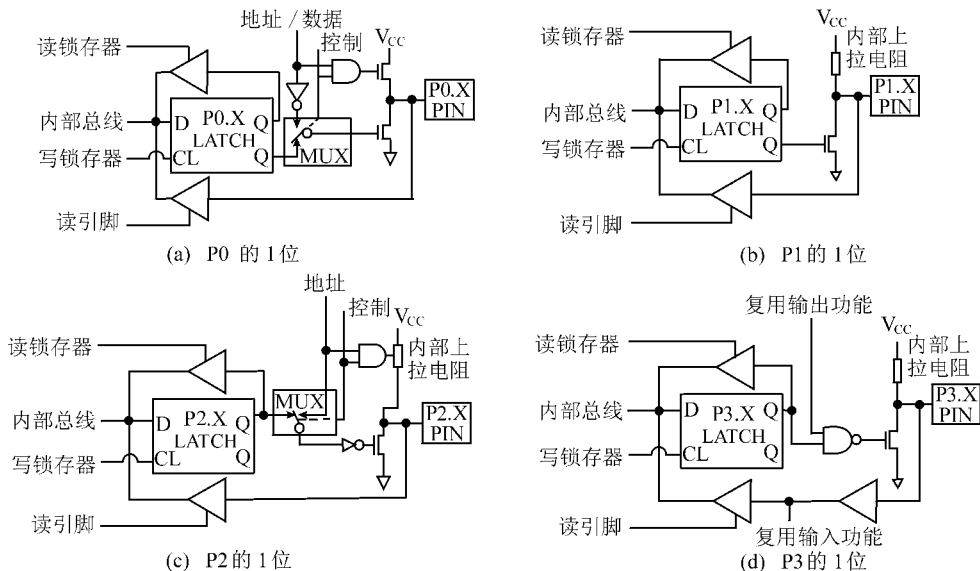


图 3-1 I/O 端口中 1 位的结构

1. P1 口

P1 口为准双向 I/O 口。图 3-1(b)是 P1 口的 1 位结构图,由端口锁存器、输出驱动器、输入缓冲器组成。在 P1 口中,它的每一位都可以分别定义作输入线或输出线使用。输出时,将“1”写入 P1 口的某一位锁存器,使输出驱动器的场效应管截止。该位的输出引脚由内部拉高电阻拉成高电平,输出为“1”。输出“0”时,将“0”写入锁存器,使输出场效应管导通,则输出引脚为低电平。当 P1 的某位作输入线时,该位的锁存器也必须保持“1”,使输出场效应管截止。这时,该位的引脚由内部提高电路拉成高电平,也可以由外部电路拉成低电平。CPU 读 P1 引脚状态时,实际上就是读出外部电路的输入信息。

2. P3 口

P3 口为多功能口。每一位都可以分别被定义为复用的输入功能或复用的输出功能。当 P3 口的某一位的锁存器被置 1 后,输出端可由“复用的输出功能”信号控制,作复用的输出功能的输出线使用。而实际上,如果把“复用输出功能”控制端置为 1,则 P3.X 端可实现复用的输入功能。

P3 口如果作普通的 I/O 使用,则其使用方法和原理与 P1 口相同,如图 3-1(d)所示。

3. P2 口

P2 口也是一个准双向 I/O 口。它有两种使用功能:一种是作普通的 I/O 口使用;另一种是作系统扩展的地址总线口,输出高 8 位的地址 A7~A15。P2 口的结构如图 3-1(c)所示,输出驱动器上有 2 个多路电子开关,由“控制”线控制。当开关接通锁存器的 Q 端输出时,P2 口作普通输入输出使用,功能及使用方法与 P1 口相同。当开关接通“地址”线时,用作地址总线口使用,P2 口的引脚状态由所输出的地址确定,此时,P2 锁存器的内容保持不变。

4. P0 口

P0 口为三态双向 I/O 口。它的结构与 P2 口相似,可作输入输出使用,也可作系统扩展的地址/数据总线口。P0 口内无内部上拉电阻,其输出驱动器上的上拉场效应管仅限于访问外部存储器时输出“1”地址(或数据)时使用,其余情况下,上拉场效应管截止。P0 口的输出驱动器中也有一个多路电子开关,当“控制”线控制接通锁存器时,P0 口作双向 I/O 口使用,如 P0 口的锁存器的值为“1”,则使输出驱动器中的 2 个场效应管截止,引脚浮空,此时端口可作高阻输入。锁存器的值为“0”时,下面的场效应管导通,输出为“0”。P0 口作地址/数据总线口使用时,由“控制”线控制将电子开关接通至“地址/数据”端,分时输出扩展外存的低 8 位地址 A0~A7 和数据 D0~D7。一般情况下,当 P0 口作输入输出线使用,都要外接拉高电阻。

在 AT89C51 和 AT89C52 中,I/O 端口的复位值均为“1”。

3.1.2 写端口操作

当执行指令来改变端口锁存器的值时,新的值是在最后一个指令周期的 S6P2 被写入锁存器的。在每一个时钟周期的 S1P1,锁存器的值由输出缓冲器采样,并保存至下一个机器周期的 S1P1 才输出到端口引脚。因此,当锁存器的状态发生变化时,这个变化不会立即出现在输出端,至少经过一个时钟周期后,才输出新的值。

在 P1,P2,P3 端口内,都接有内部上拉电阻。此上拉电阻分为固定部分和附加部分。当端口的状态要从“0”变为“1”时,在发生变化的那个机器周期的 S1P1 和 S1P2 接通附加的拉高电路以增加变化的速度;否则,这个状态的变化将十分缓慢。附加的拉高电路允许通过的电流比普通的上拉电阻大 100 倍,附加上拉电路如图 3-2 所示,由 3 个 P 型场效应管组成,另有 1

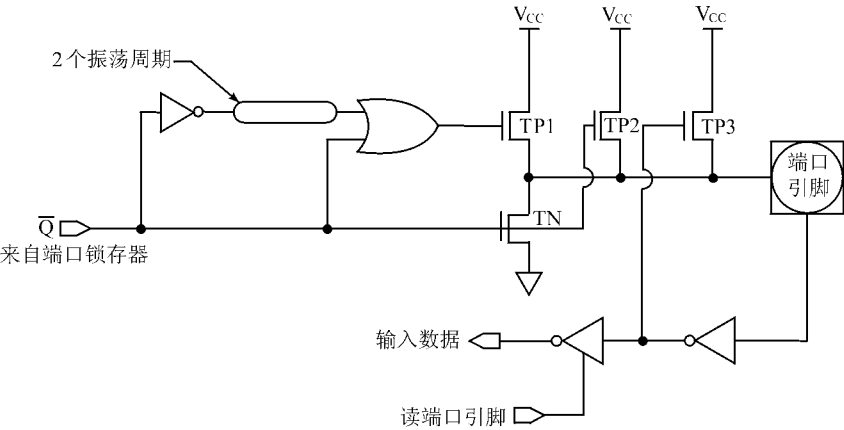


图 3-2 P1 和 P2 的附加上拉电路图

个 N 型场效应管。当把“1”信号加到 N 型场效应管时,管子导通;加入“0”信号时,管子截止。而对 P 型场效应管则相反。在图 3-2 中的 TP1 管在锁存器的状态从“0”变“1”后有 2 个振荡周期的导通时间,TP1 导通后,经过一个反相器又使 TP3 导通,这级反相器与 TP3 管构成一个锁存器并将状态保持为“1”。

当端口引脚上出现负向的干扰信号时,会使 TP3 截止,而使端口引脚进入浮空态。TP3 在 TN 截止的时候有一个微弱的上拉作用,通常它的上拉强度约为 TP3 的 1/10。由于 TP2 的上拉作用使 TP3 重新导通,所以当端口在发生干扰而不为“1”时,TP2 使它恢复为“1”。

3.1.3 读端口操作

P1~P3 口的输出缓冲器每个都能驱动 4 个 LSTTL 输入。CMOS 型的芯片引脚可由集电极开路电路和漏极开路电路驱动。但在状态从“0”变“1”时,变化的速度很慢;输入“0”时,TP3 截止,仅靠 TP2 的非常微弱的上拉来驱动状态的变化。

在外部总线方式下,P0 口的输出缓冲器能驱动 8 个 LSTTL 输入,但 P0 口用作输入输出时需外接上拉电路驱动输入。

3.1.4 读—修改—写端口操作

读端口指令实际上分为读锁存器和读端口引脚状态两种。读—修改—写指令用来读锁存器 Q 端的数值而不是读引脚状态。其操作是读入一个锁存器的值后,可能进行修改,然后重新写进锁存器中。当指令中的目的操作数是端口或端口的某位时,常使用表 3-2 所列出的指令。

表 3-2 I/O 端口常用指令

助记符	功 能	实 例
ANL	逻辑与	ANL P1,A
ORL	逻辑或	ORL P2,A
XRL	逻辑异或	XPL P3,A
JBC	测试位为 1 跳转并清 0	JBC P1.1,LABEL
CPL	位求反	CPL P3.0
INC	增 1	INC P2
DEC	减 1	DEC P2
DJNZ	减 1,结果不为 0 跳转	DJNZ P3,LABEL
MOV PX.Y C	把进位位送入 PX 口的第 Y 位	
CLR PX.Y	清 PX 口的 Y 位	
SET PX.Y	置 PX 口的 Y 位	

表 3-2 中的最后三条指令实际上就是端口的读—修改—写指令。它们可以将端口字节中的 8 位全部读入,然后修改寻址位后,将新的字节写回端口锁存器。对于读—修改—写指令,直接读锁存器而不是读端口引脚是为了避免有错误引脚上的电平信号的可能性。例如,若用一根端口引脚线去驱动一个晶体管的基极,当向此端口线写“1”时,三级管导通,并把引脚上的电平拉低,这时 CPU 如要从引脚读取数据,则会把此数据(应为“1”)错读为“0”;若从锁存器读取而不是读引脚,则读出的应该是正确的数值“1”。

3.1.5 外部存储器的存取操作

对外部存储器存取操作包含了对程序存储器(EPROM)和数据存储器(RAM)的访问。对外部程序存储器的访问是使用 $\overline{\text{PSEN}}$ 作读选通信号。对外部数据存储器的存取则使用 $\overline{\text{RD}}$ (P3.7)和 $\overline{\text{WR}}$ (P3.6)作读写选通信号。读写周期的时序如图 3-3~图 3-5 所示。当从外部程序存储器读取指令时,需要使用 16 位的地址,而对外部数据存储器的存取既可用 16 位地址(MOVX @DPTR),也可用 8 位地址(MOVX @Ri)。当使用 16 位地址时,高 8 位地址一定是从 P2 口输出,并且在整个读写周期中保持不变。但是,当 P2 口用来发送高 8 位地址时,其输出驱动电路要求有一个较强的上拉电路,特别是输出的地址位为“1”时(例如执行 MOVX @DPTR 指令时就必须),在此期间,P2 口锁存器的值并不要求为“1”,且 P2 口的锁存器的内容也不会被修改。如果外部存储器周期结束后不是紧跟着一个外部存储器周期,P2 口锁存器的内容就会重新出现在引脚上。

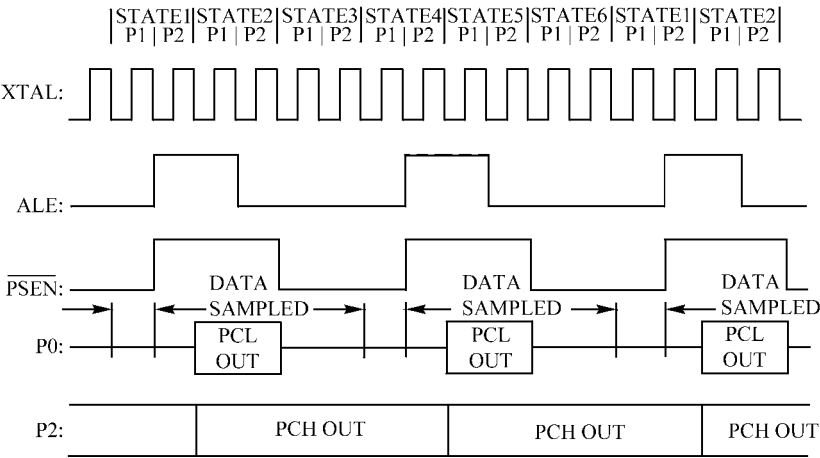


图 3-3 外部程序存储器读周期

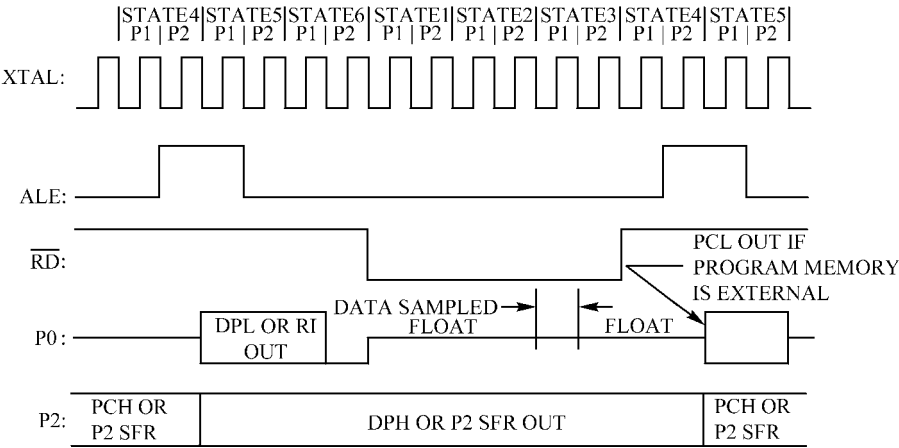


图 3-4 外部数据存储器读周期

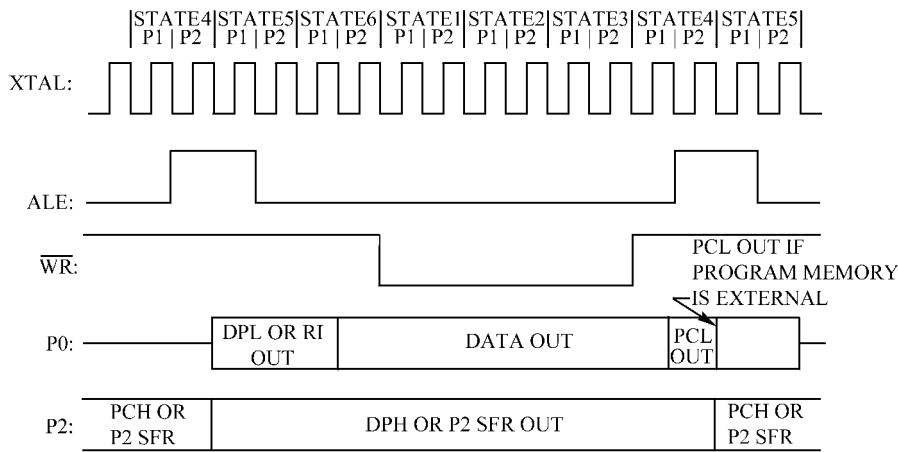


图 3-5 外部数据存储器写周期

当使用 8 位地址时(例如执行 MOVX @Ri 指令),P2 锁存器的内容(即引脚状态)维持,并在整个外部存储周期中保持不变。因此在这种应用中,有时可用 P2 口输出页地址。

在任何时候,低位地址字节与 8 位数据字节都是分时使用 P0 口的。由“地址/数据”线驱动 P0 口输出缓冲器的 2 个场效应管,因此 P0 口输出端不是漏极开路输出,也不需要上拉电路。地址锁存允许信号 ALE 用来把低 8 位地址锁入外部地址锁存器(此信号低电平有效)。在写周期,数据字节应在 WR 信号有效前送出 P0 口,并且一直维持到 WR 信号有效。在读周期,从外部存储器读入的数据是在读选通信号 RD 由低变高时被锁入 P0 口。在对外部存储器进行存取操作时,CPU 先要把 0FFH 写入 P0 口锁存器中,把 P0 口原有的数据清除掉。而用户在读取外部存储器期间不要写 P0 口,否则,输入的代码会被改变。访问外部程序存储器时,需要满足以下 2 个条件:

- (1) EA 信号有效;
- (2) 程序计数器 PC 的值要大于 0FFFH(使用 AT89C52 要大于 1FFFH)。

当 CPU 访问外部程序存储器时,P2 口只能作输出口。输出来自程序计数器 PC 的高 8 位地址,不能用作普通 I/O 口使用。此时,P2 口的输出驱动器要接一个较强的上拉电路,以输出 PC 寄存器中的“1”地址位。

3.2 串行接口的结构及功能

89 系列单片机的串行通信口可以工作于同步和异步通信方式下。当工作于异步方式时,它具有全双工的操作功能,也就是说,它可以同时进行数据的发送与接收。串行口内的接收器采用的是双缓冲结构,能够在接收到的第一个字节从接收寄存器读走之前就开始接收第二个字节(当然,如果第二个字节接收完毕,而第一个字节仍然没有被读走,那将会丢掉一个字节)。串行口的发送和接收操作都是通过特殊功能寄存器中的数据缓冲寄存器 SBUF 进行的,但在 SBUF 的内部,接收寄存器和发送寄存器在物理结构上是完全独立的。如果将数据写入 SBUF,数据会被送入发送寄存器准备发送。如果执行 SBUF 指令,则读出的数据一定来自接收缓冲器。因此,CPU 对 SBUF 的读写,实际上是分别访问 2 个不同的寄存器。这 2 个寄存器的功能决不能混淆。

3.2.1 串行口的方式控制

89 系列单片机的串行口有 4 种不同的工作方式。它们是由串行口控制寄存器 SCON 中的方式选择位定义的。在 SCON 寄存器中还包含了发送及接收的状态控制位,如发送和接收中断标志、传送的数据帧中的第 9 数据位等。控制寄存器 SCON 的格式如图 3-6 所示。

(MSB)				(LSB)			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

图 3-6 控制寄存器(SEON)的格式

SM0,SM1:串行口工作方式选择位,如表 3-3 所列。

表 3-3 串行口方式选择

SM0	SM1	方 式	功 能	波特率
0	0	0	移位寄存器	$f_{osc}/12$
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	$f_{osc}/64$ 或 $f_{osc}/32$
1	1	3	9 位 UART	可变

SM2:允许方式 2 和方式 3 的多机通信控制位,如 SM2=1,则接收到的第 9 位数据位(RB8)为 0 时,不激活 RI;对于方式 1,如 SM2=1,则只有收到有效的停止位时,才会激活 RI;对方式 0,SM2 应为 0。

REN:串行接收允许位。由软件置位以允许接收;由软件清 0 来禁止接收。

TB8:在方式 2 和方式 3 中要发送的第 9 数据位,由软件置位或清 0。

RB8:方式 2 和方式 3 中接收到的第 9 位数据。在方式 1 中,如果 SM2=0,则 RB8 是接收到的停止位;对于方式 0,该位没有被使用。

TI:发送中断标志。在方式 0 中,当发送完第 8 位后由硬件将该位置位。而在其他工作方式中,则在开始发送停止位时将 TI 置位,但必须由软件清 0。

RI:接收中断标志。在方式 0 中,当接收完第 8 位数据后由硬件将该位置位。而在其他工作方式中,在接收到停止位时,由硬件将 RI 置位(其他的规定见 SM2 控制位)。该位须由软件清 0。

串行接口的设置情况如表 3-4 所列。

表 3-4 串行口设置

方 式	SCON	SM2 变化
0	10H	单个处理器 (SM2=0)
1	50H	
2	90H	
3	D0H	
0	NA	多处理器 (SM2=1)
1	70H	
2	B0H	
3	F0H	

3.2.2 串行口的工作方式

串行口有 4 种工作方式,是由 SCON 寄存器中的 SM0,SM1 两位来决定的。下面将分别讨论各种工作方式的功能、特点和原理。

1. 方式 0

方式 0 为移位寄存器方式,也称为同步方式。它通过 RXD 端(P3.0)串行发送或接收数据,而用 TXD 端(P3.1)输出移位脉冲。数据的传送以 8 位为一帧。无论是发送或接收,都是最低有效位 LSB 居先,传送的波特率固定为振荡频率的 1/12。方式 0 的结构原理如图 3-7 所示;定时波形如图 3-8 所示。

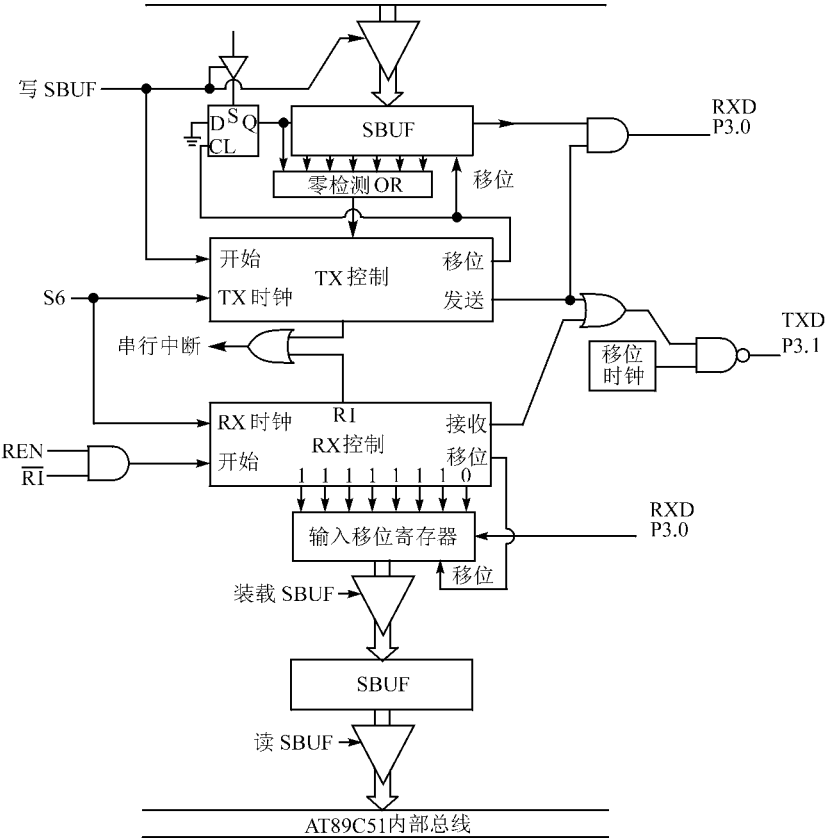


图 3-7 串行接口方式 0 的结构

(1) 方式 0 的发送

发送的初始化工作是由指令进行的。当 CPU 向发送数据缓冲器写入一个数据时,就启动串行口的发送。这时指令产生的“写 SBUF”信号在 S6P2 处将 1 写入输出移位寄存器的第 9 位,并启动 TX 控制寄存器开始发送。内部定时逻辑在“写 SBUF”和 SEND 信号被激活之间要有一个完整的机器周期。在 SEND 信号有效期间,输出移位寄存器把数据送 RXD 端输出,而移位脉冲由 TXD 端输出。从 TXD 端输出的移位脉冲在每个机器周期的 S3,S4,S5 期间为低电平;在 S6,S1,S2 期间为高电平。在每个机器周期的 S6P2 使输出移位寄存器的内容右移一位,左边移入 0。当数据的最高有效位 MSB 被移到输出移位寄存器的输出位时,原写

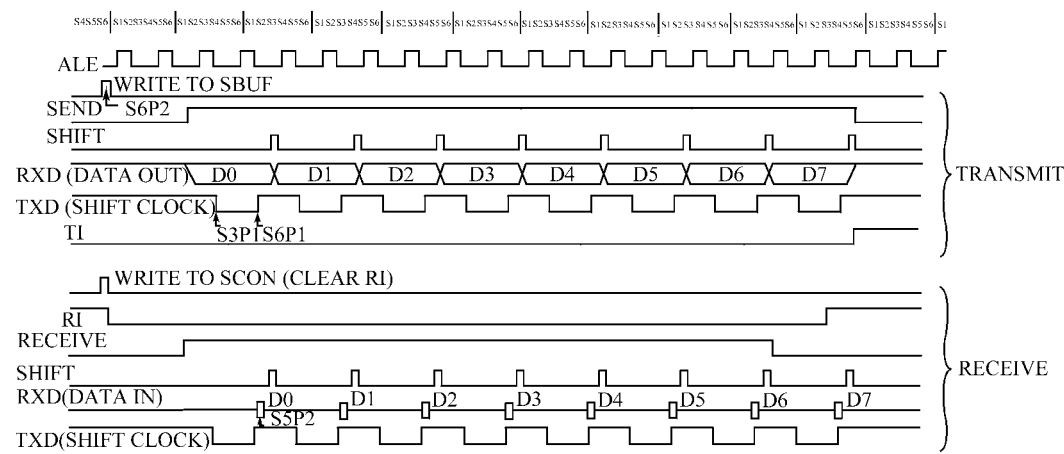


图 3-8 串行接口方式 0 的定时波形

入第 9 位的 1 则移到 MSB 的左边,而左边其余位都为 0。当 TX 控制器检测到该状态时,则控制作最后一次移位,接着清除 SEND 信号并将发送中断标志位 TI 置位。这些动作是发生在“写 SBUF”信号有效后的第 10 个机器周期的 S1P1 处。此时意味着完成一个字节的输出。

(2) 方式 0 的接收

方式 0 的接收是由设置条件标志 REN=1 和 RI=0 后开始启动的。在下一个机器周期的 S6P2 处,RX 控制器将 11111110 写入接收移位寄存器,并在下一个时钟段使 RECEIVE 控制端有效。RECEIVE 有效的信号控制将移位脉冲 RECEIVE CLOCK 从 TXD 端输出,而 SHIFT CLOCK 信号在每个机器周期的 S3P1 和 S6P1 发生跳变,并在每个机器周期的 S6P2 使接收移位寄存器左移一位,右边输入的值是在同一机器周期的 S5P2 期间从 RXD 端采样到的值。当接收到的数据逐位从右边移入接收移位寄存器时,原来置入该寄存器的值 11111110 将从左边被逐位移出。当 0 被移到左边最后一位时,它标志着 RX 控制器的下一次移位为最后一位的移位,完成后,就将接收移位寄存器的内容写入 SBUF。在“写 SCON”信号发生后的第 10 个机器周期的 S1P1 段,撤除 RECEIVE 信号,并将接收中断标志位 RI 置位,至此便完成一帧数据的接收。

2. 方式 1

方式 1 为标准异步通信方式。它通过 TXD 端发送数据,RXD 端接收数据,通信格式为每帧 10 位,其中包括 8 个数据位(先低位后高位)、一个起始位(0)、一个停止位(1)。格式如图 3-9 所示。

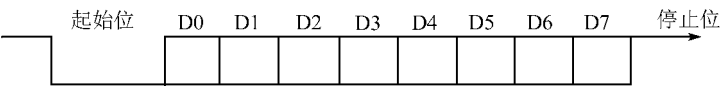


图 3-9 方式 1 数据格式

接收时,接收到的停止位要被送入 SCON 寄存器的 RB8 位。传送的波特率可变,它由 T1 的溢出率决定。在 AT89C52 中,波特率由 T1 或 T2 的溢出率决定。或由 T1 和 T2 共同决定,这时,一个用于发送,另一个用于接收。串行口方式 1 的结构原理图如图 3-10 所示,定时波形如图 3-11 所示。

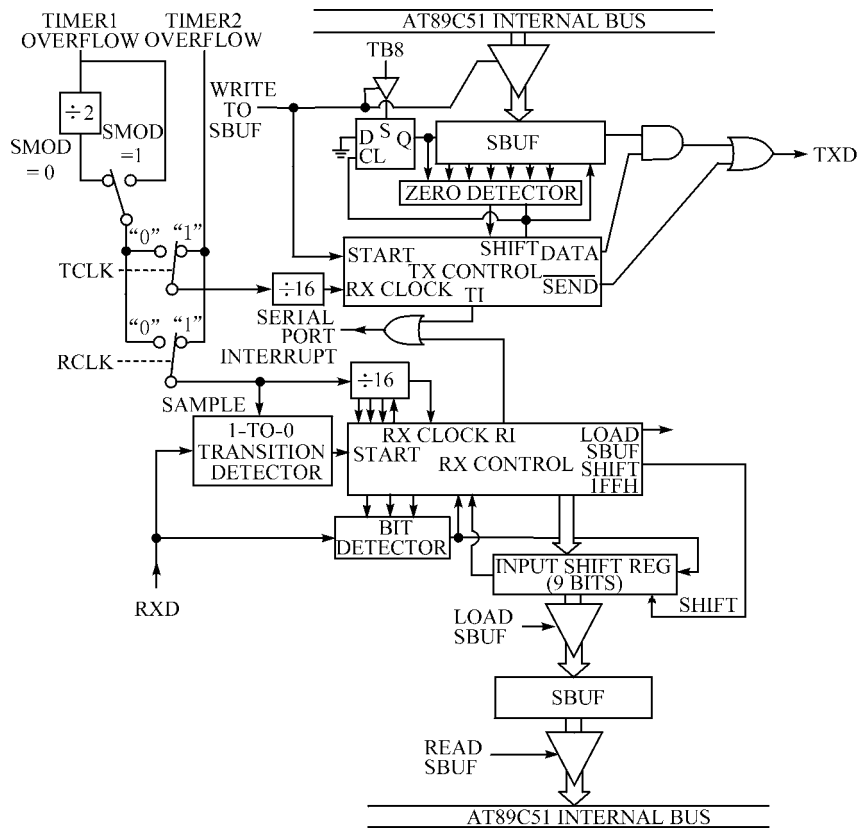


图 3-10 串行接口方式 1 的结构

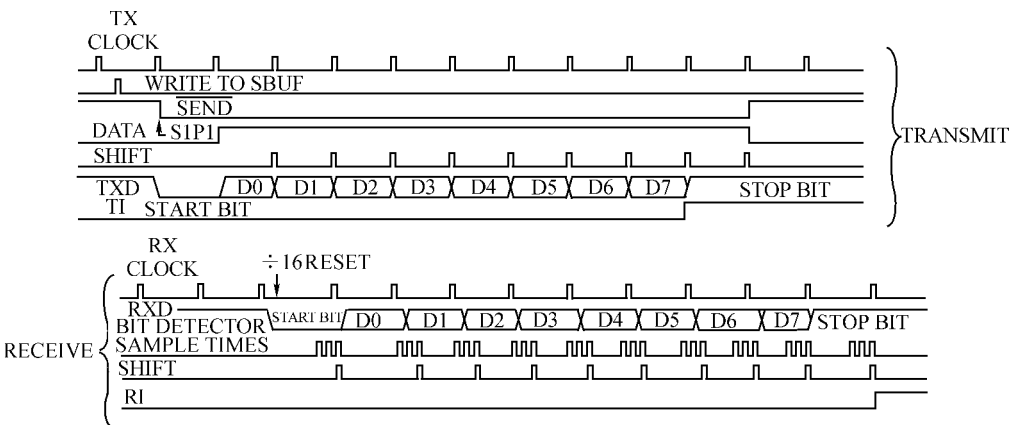


图 3-11 串行接口方式 1 的定时波形

(1) 方式 1 的发送

方式 1 的发送是由 CPU 向 SBUF 写入一个数据开始的。当发出“写 SBUF”信号时,将 1 写入发送移位寄存器的第 9 位,作 TX 控制寄存器的发送结束标志。实际的发送时间是在内部的 16 分频计数器下一次循环的机器周期的 S1P1 开始,且发送的位时间与这个 16 分频计数器同步。

当 $\overline{\text{SEND}}$ 有效时,发送开始。首先将起始位送出 TXD 端,经过一个位时间后,发出第一个移位脉冲。当发送的数据在移位寄存器中逐位右移输出时,左边移入 0。在数据字节的最高位 MSB 被移到移位寄存器的输出位置时,原来写入第 9 位的 1 就在 MSB 的左边,左边的其余位全为 0。检测电路检测到这个条件时,TX 控制器控制移位寄存器作最后一次的移位输出,在一帧数据发送完毕后,也就是在“写 SBUF”信号发出的第 10 个周期后将 $\overline{\text{SEND}}$ 信号撤除并将发送中断标志位 TI 置 1。

(2) 方式 1 的接收

方式 1 是由 SCON 寄存器中的标志位 REN 控制的。当 $\text{REN}=1$ 时,允许接收器接收。接收的操作是从检测到 RXD 端上的输入电平发生负跳变时开始的,而对 RXD 端的采样是以 16 倍于波特率的速度进行。当检测到负跳信号时,立即将内部的 16 分频计数器复位,随后将 1FFH 写入接收移位寄存器。16 分频计数器的计数脉冲是按接收的位时间来定义的。计数器的 16 个状态将传送的每一个位时间分成 16 等份,并在每个位时间的第 7、第 8、第 9 个状态时对 RXD 端进行采样。当采样到的 3 个值中至少有 2 个相同时,这个值就被接收,这样处理可以防止干扰。如果在第一个位时间接收到的值不为 0,则起始位无效,接收电路被复位,重新开始搜索输入端上的负跳变。如果接收到的起始位有效,则将其移入接收移位寄存器,并开始接收本帧的其余部分信息。当接收到的数据逐位从右边移入接收寄存器时,原有的 1 则从左边移出。起始位被移到最左边位置时(方式 1 的移位寄存器为 9 位),RX 控制器控制作最后一次的移位接收。如在最后一个移位脉冲发生后满足以下条件:

- ① $\text{RI}=0$;
- ② $\text{SM2}=1$ 或接收到的停止位为 1;

则将接收移位寄存器的内容送入 SBUF 和 SCON 的 RB8(停止位)中,并将接收中断标志位 RI 置 1。如果上述 2 个条件不满足,信息将丢失。当接收完一帧数据后,控制器继续搜索 RXD 上的负跳信号。

3. 方式 2 和方式 3

当串行口定义为方式 2 或方式 3 时,它是一个 9 位的异步通信接口,通过 RXD 端发送、TXD 端接收,通信的格式为每帧 11 位。其中有:一个起始位(0);8 个数据位(由低到高传送);一个可编程第 9 数据位;一个停止位。格式如图 3-12 所示。

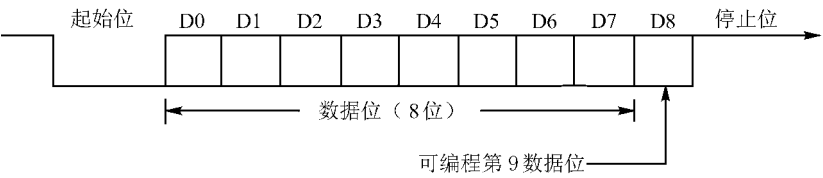


图 3-12 方式 2 和方式 3 的数据格式

在数据发送前,第 9 数据位(SCON 寄存器中的 TB8 位)可由程序设置为 1 或 0,例如,可将奇偶校验位(PSW 寄存器中的 P 位)的值移入 SCON 的 TB8 中,接收时,第 9 数据位被送入 SCON 寄存器的 RB8 位中,而将停止位忽略。方式 2 的波特率固定为振荡频率的 1/32 或 1/64,而方式 3 的波特率可变,由定时器 T1 或 T2(AT89C52)的溢出率所确定。方式 2 和方式 3 的串行口结构原理如图 3-13 和图 3-15 所示,定时波形如图 3-14 和图 3-16 所示。

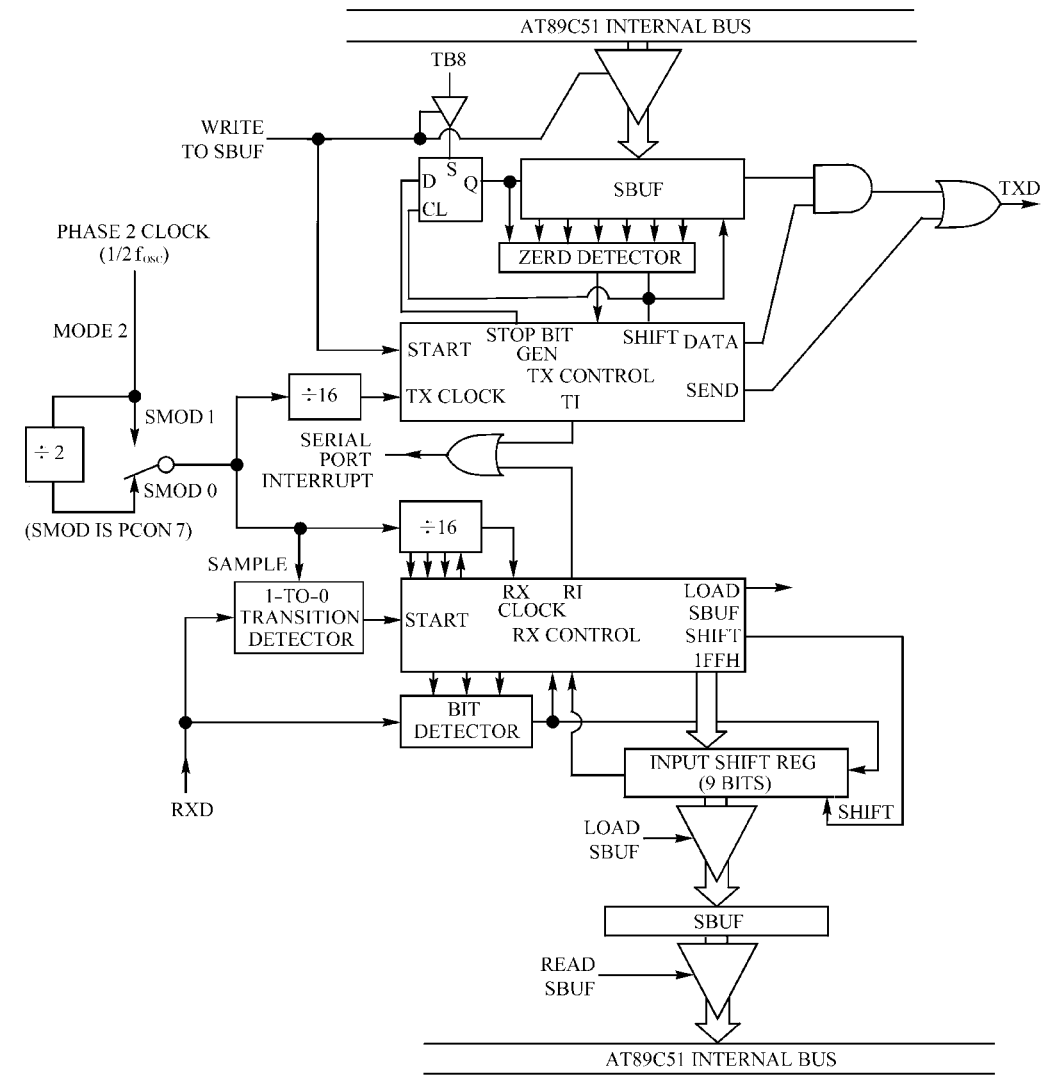


图 3-13 串行接口方式 2 的结构

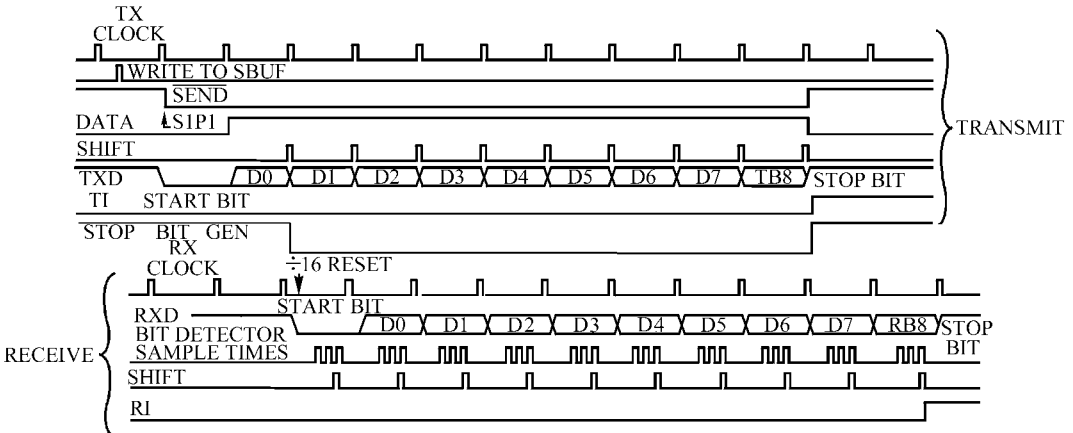


图 3-14 串行接口方式 2 的定时波形

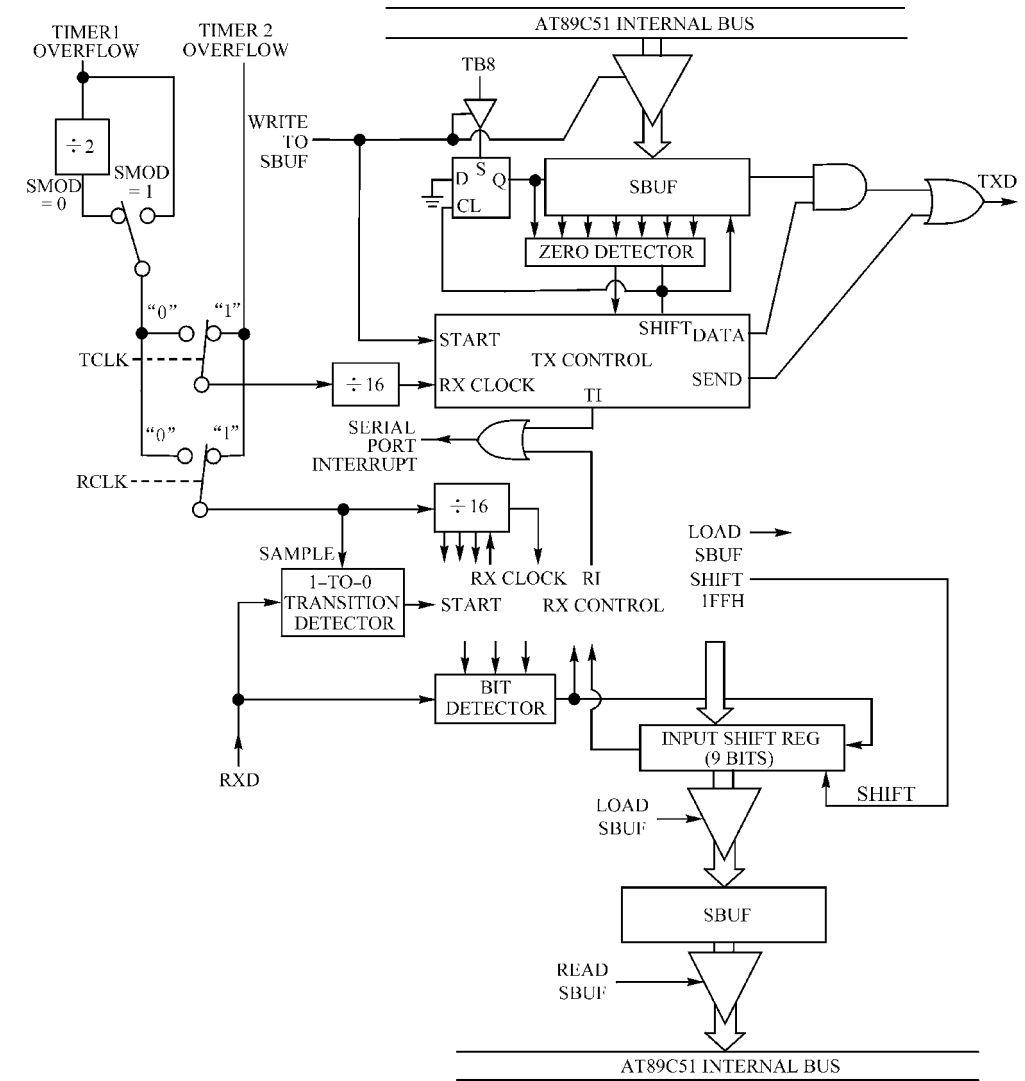


图 3 - 15 串行接口方式 3 的结构

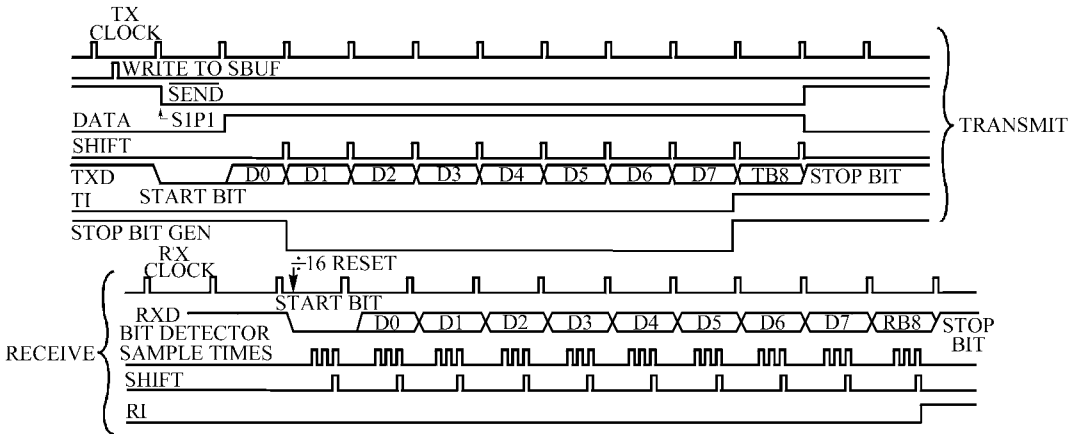


图 3 - 16 串行接口方式 3 的定时波形

(1) 方式 2 和方式 3 的发送

当 CPU 向 SBUF 写入一个数据时,就启动串行口的发送:“写 SBUF”信号控制把 TB8(第 9 数据位)取出写入发送移位寄存器的第 9 位,并可作 TX 控制器的发送标志。实际的发送是在内部 16 分频计数器下一次循环的机器周期的 S1P1 开始,且发送的位时间与 16 分频计数器同步。当 $\overline{\text{SEND}}$ 有效时,打开 TXD 端的输出控制门,经过一个位时间后 DATA 被激活,允许移位寄存器将数据输出到 TXD,再经过一个位时间后,发出第一个移位脉冲,并且将 1(停止位)写入移位寄存器的第 9 位。以后,数据从右边移出发送,左边移入 0。当 TB8 被移到移位寄存器的输出位置时,停止位就在 TB8 的左边,而其余位置全为 0。当检测电路检测到此条件时,TX 控制电路控制作最后一次的移位输出。在“写 SBUF”信号发出后的第 11 个发送时钟脉冲位置,将 $\overline{\text{SEND}}$ 信号变高电平,发送中断标志位置 1。

(2) 方式 2 和方式 3 的接收

当 REN 置 1 后,接收器允许接收,并开始以 16 倍于波特率的速度采样 TXD 端的状态。当检测到 TXD 端的输入电平有负跳变发生时,立即将内部 16 分频计数器复位,并把 1FFH 写入输入移位寄存器。接收时序与方式 1 相同。计数器的 16 个状态把每个位时间分成 16 等份,在第 7,8,9 三个状态时对 RXD 采样 3 次,在采样值中至少有 2 次相同时,接收器将此值接收。如果在第一个位时间中接收的值不为 0,则起始位无效,即将接收电路复位,继续搜索 RXD 端的负跳变;如果接收到的起始位有效,则移入移位寄存器,并开始接收本帧的其余信息位。在接收过程中,数据不断被接收并从右边逐位移入移位寄存器,而原来的 1 不断从左边移出。当起始位(0)被移到最左边的位置时(方式 2、方式 3 中的串行口移位寄存器为 9 位),RX 控制寄存器控制作最后一次的移位接收。当 RI=0 和 SM2=0 或接收到的第 9 位数据位为 1 时,将接收到的 8 位数据送 SBUF 和第 9 数据位装入 RB8,且置接收中断标志位 RI 为 1。若上述条件不满足,信息将被丢掉,RI 不置 1,接着重新检测 RXD 端的负跳变,准备接收下一帧信息。

3.2.3 多机通信过程

串行口的工作方式 2 和方式 3 特别提供了多机通信的功能。其原理是利用了这两种通信方式中的可编程第 9 数据位。由于在接收时,接收到第 9 数据位后是要将它送入 RB8 位中的,如果在接收前定义了 SM2=1,则接收机在接收到停止位后会对 RB8 位进行检测;如 RB8=1,则串行口向 CPU 发出中断请求,请求对 SBUF 中的数据进行处理;如 RB8=0,则数据不予接收;如定义了 SM2=0,则收到的数据不管 RB8 为何值都予以接收处理。

在方式 2 和方式 3 中利用串行口中断来实现多机通信的实际过程简述如下:在一个主从式的多机系统中,通信前,各从机均置 SM2=1,当主机要与某从机交换数据时,首先主机向各从机发送一个从机的标识地址(地址与数据帧的区别在于前者的第 9 位数据为 1,后者该位为 0)。当各从机接收到主机发来的地址时,都产生中断。在中断服务程序中检测接收到的地址是否与本站机号相同。如相同,则将 SM2 位清 0,准备接收主机发来的数据;如不相同,则将 SM2 置 1,主机随后发来的数据(第 9 数据位为 0)都被丢掉。

SCON 寄存器中的 SM2 标志在方式 0 中不起作用,但在方式 1 中可利用它来检测停止位的有效性。即在接收时,如 SM2=1,在收到数据的有效停止位时,才激活 RI,向 CPU 发出中断请求。

3.2.4 波特率的产生

波特率即在串行通信中每秒传送的数据位。在串行口的 4 种工作方式下,对波特率各有不同的要求。波特率的产生和电源控制寄存器 PCON 有关,参见图 3-36。

1. 串行口工作于方式 0

操作方式 0 的波特率是固定的,为振荡频率的 1/12。串行口工作于方式 0 时,不需要对定时器/计数器进行设置,只要定义 SCON 寄存器即可。

$$\text{波特率} = \frac{\text{振荡频率}}{12}$$

2. 串行口工作于方式 1

操作方式 1 的波特率是可变的。波特率可由定时器 1 或定时器 2(AT89C52)产生。

(1) 定时器/计数器 1 作波特率发生器时,工作于方式 2(自动重装载)。

$$\text{波特率} = \frac{K \times f_{\text{osc}}}{32 \times 12 \times [256 - (\text{TH1})]}$$

式中, f_{osc} 为振荡频率。当 SMOD=0 时, $K=1$; 当 SMOD=1 时, $K=2$ 。

用户熟悉波特率,但也需要了解 TH1 的重装载值,由下式可计算 TH1,即

$$\text{TH1} = 256 - \frac{K \times f_{\text{osc}}}{384 \times \text{波特率}}$$

TH1 必须为整数。按四舍五入对 TH1 取整,产生的波特率可能不理想,此时,用户必须选择另一种晶振频率(如表 3-13 所列)。

由于寄存器 PCON 不可位寻址,用户可以将要置 1 的位与寄存器 PCON 进行逻辑“或”操作(即 ORL RCON, #80H),PCON 地址为 87H。

(2) 定时器/计数器 2 作波特率发生器:定时器 2 作波特率发生器时,工作于波特率发生方式。如果定时器 2 的时钟来自 T2(P1.0)引脚,则波特率由下式给出,即

$$\text{波特率} = \frac{\text{定时器 2 溢出率}}{16}$$

如果计数时钟来自内部,则波特率由下式计算,即

$$\text{波特率} = \frac{f_{\text{osc}}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

RCAP2H, RCAP2L 为自动重装载值,由下式计算,即

$$\text{RCAP2H}, \text{RCAP2L} = 65536 - \frac{f_{\text{osc}}}{32 \times \text{波特率}}$$

3. 串行口工作于方式 3

串行口工作于方式 2 时,波特率为振荡频率的 1/32 或 1/64,取决于 PCON 寄存器中的 SMOD 位。

在这种方式下,不需要定时器,时钟来自内部时钟相位 2。

SMOD=1 时,波特率=1/32 振荡频率;

SMOD=0 时,波特率=1/64 振荡频率。

4. 串行口工作于方式 3

串行口工作于方式 3 时,波特率是可变的,其设置方法与方式 1 相同。

在表 3-5 中给出的是常用波特率以及相应的振荡频率 f_{osc} 和定时器 T1 的状态。
表 3-6 给出的是不同振荡频率时定时器 TH1 所产生的波特率。
表 3-7 给出的是波特率的误差。

表 3-5 常用的波特率值

波特率	f_{osc}/MHz	SMOD	定时器 T1		
			C/ $\overline{\text{T}}$	方 式	重装值
19.2 K	11.059	1	0	2	FDH
9.6 K	11.059	0	0	2	FDH
4.8 K	11.059	0	0	2	FAH
2.4 K	11.059	0	0	2	F4H
1.2 K	11.059	0	0	2	E8H
137.5	11.986	0	0	2	1DH
110	6	0	0	2	72H
110	12	0	0	1	FEEBH

表 3-6 不同振荡频率时的波特率

晶振频率/MHz	7.372 8	8	11.059 2	12.00	14.751 56	16.00
TH1						
E0	600	651	900	976	1 200	1 302
F0	1 200	1 302	1 800	1 953	2 400	2 604
F8	2 400	2 604	3 600	3 906	4 800	5 208
F9	2 743	2 976	8 229	4 464	5 486	5 952
FA	3 200	3 472	9 600	5 208	6 400	6 944
FF	19 200	20 833	57 600	62 500		41 666

表 3-7 波特率及其误差

波特率	晶振频率/MHz	SMOD	TH1 重装载值	实际波特率	误 差
9 600	12.000	1	-7(F9H)	8 923	7 %
2 400	12.000	0	-13(F3H)	2 404	0.16 %
1 200	12.000	0	-26(E6H)	1 202	0.16 %
19 200	11.059	1	-3(FDH)	19 200	0
9 600	11.059	0	-3(FDH)	9 600	0
2 400	11.059	0	-12(F4H)	2 400	0
1 200	11.059	0	-24(E8H)	1 200	0

注:由于四舍五入,波特率的结果会有微小误差。通常,在使用异步(起动/停止)通信时,允许误差为 5 %。使用 11.059 MHz 晶振可能获得精确的波特率。对大多数常用波特率,表 3-7 中 TH1 重装载值,使用 12.000 MHz 或 11.059 MHz 晶振。

3.3 定时器的结构及功能

AT89C51 内含 2 个 16 位的定时器/计数器 T0 和 T1,而在 AT89C52 内则有 3 个 16 位定时器/计数器 T0,T1 和 T2。它们都可以分别作定时器或计数器使用。当用于定时器方式时,定时器的输入来自内部时钟发生电路,每过一个机器周期,定时器加 1,而一个机器周期包含有 12 个振荡周期,所以,定时器的计数频率为晶振频率的 1/12。

在计数器工作方式中,计数器对外部事件计数。计数脉冲来自外部输入引脚。当外部输入引脚发生“1”和“0”的负跳变时,计数器加 1。内部硬件在每个机器周期的 S5P2 采样外部输入引脚状态。当在一个机器周期采样到高电平,接着另一个周期采样到低电平时,计数器的值将在检测到变化后的下一个周期的 S3P1 中被更新。由于对外部事件检测需要 2 个机器周期(24 个振荡周期),所以,其计数频率最高为晶振频率的 1/24。在计数器方式中,对外部输入脉冲的占空度没有特别的要求,但其高电平和低电平的持续时间至少要保持一个机器周期以上。

T0 和 T1 除了有定时器和计数器功能外,还有 4 种工作方式,分别为 13 位定时器方式、16 位定时器方式、8 位常数自动重装入方式、分定时器工作方式。而 AT89C52 除 T0 和 T1 外,T2 还有 3 种工作方式:捕捉方式、常数自动重装入方式和波特率发生器方式。

为了实现定时和计数功能,定时器中含有 3 种基本的寄存器:控制寄存器、方式寄存器和定时器/计数器。

控制寄存器是一个 8 位的寄存器,用于控制定时器的的工作状态。

方式寄存器是一个 8 位的寄存器,用于确定定时器的的工作方式。

定时器/计数器是 16 位的计数器,分为高字节和低字节两部分。一般情况下,也把定时器/计数器简称为定时器。

表 3-8 给出了和定时器有关的寄存器。

表 3-8 和定时器有关的寄存器

定时器的 SFR	用 途	地 址	位寻址
TCON	控制寄存器	88H	有
TMOD	方式寄存器	89H	无
TL0	定时器 T0 低字节	8AH	无
TL1	定时器 T1 低字节	8BH	无
TH0	定时器 T0 高字节	8CH	无
TH1	定时器 T1 高字节	8DH	无
T2CON *	定时器 T2 控制寄存器	C8H	有
T2MOD *	定时器 T2 方式寄存器	C9H	无
RCAP2L *	定时器 T2 低字节捕捉	CAH	无
RCAP2H *	定时器 T2 高字节捕捉	CBH	无
TL2 *	定时器 T2 低字节	CCH	无
TH2 *	定时器 T2 高字节	CDH	无

* 仅在 AT89C52 中有。

3.3.1 定时器 T0,T1

AT89C51 单片机有 2 个定时器/计数器,即定时器/计数器 0 和 1,简写为 T0,T1。AT89C52 单片机除了有上述 2 个定时器/计数器外,还有一个定时器/计数器 2。后者的功能比前两者强。

在这一节中,介绍定时器 T0,T1 的工作情况。

1. T0 和 T1 的工作控制

定时器 T0,T1 的工作控制是由控制寄存器 TCON 和方式寄存器 TMOD 实现的。

(1) 控制寄存器 TCON

TCON 格式如图 3-17 所示:

MSB				LSB			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

图 3-17 TCON 格式

其中,

TF1(TCON. 7):定时器 1 溢出标志。当定时器/计数器 1 溢出时,由硬件置位;当主机响应中断,转向中断服务程序时,由硬件清 0。

TR1(TCON. 6):定时器 1 运行控制位。由软件置位/复位来开启或关闭定时器/计数器 1。

TF0(TCON. 5):定时器 0 溢出标志。当定时器/计数器 0 溢出时,由硬件置位;当主机响应中断,转向中断服务程序时,由硬件清 0。

TR0(TCON. 4):定时器 0 运行控制位,由软件置位/复位来开启或关闭定时器/计数器 0。

IE1(TCON. 3):外部中断 1 跳变中断请求标志。当检测到 $\overline{INT1}$ 发生“1”→“0”跳变时,由硬件置位;当主机响应中断,转向中断服务程序时,由硬件清 0。

IT1(TCON. 2):外部中断 1 触发方式控制位。由软件置位/清 0 来选择外部中断 1 的跳变/电平触发中断请求。

IE0(TCON. 1):外部中断 0 跳变中断请求标志。当检测到 $\overline{INT0}$ 发生“1”→“0”跳变时,由硬件置位;当主机响应中断,转向中断服务程序时,由硬件清 0。

IT0(TCON. 0):外部中断 0 触发方式控制位。由软件置位/清 0 来选择外部中断 0 的跳变/电平触发中断请求。

(2) 方式寄存器 TMOD

TMOD 格式如图 3-18 所示。

MSB				LSB			
定时器 1				定时器 0			
GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0

图 3-18 TMOD 格式

其中,

GATE:门控位,当 GATE=1 时,仅当控制寄存器 TCON 的 TRX=1(X 为 0 或 1),且

INTX 端为高电平时,定时器/计数器才计数。当 $GATE=0$ 时,定时器/计数器由 TRX 控制,TRX 为“1”,允许计数;TRX 为“0”,不允许计数。

$C/\overline{T}$:定时器和计数器选择位。

$C/\overline{T}=0$ 为定时器方式(输入为系统时钟);

$C/\overline{T}=1$ 为计数器方式(计数脉中从 T_x 引脚输入)。

M1M0:工作方式选择位。含义如表 3-9 所列。

表 3-9 M1,M0 操作方式及功能

M1 M0	操作方式	功能描述
0 0	方式 0	13 位定时器
0 1	方式 1	16 位定时器/计数器
1 0	方式 2	具有自动再装入的 8 位定时器/计数器
1 1	方式 3	定时器 0 分成 2 个 8 位定时器。TL0 作为 8 位定时器/计数器,由定时器 0 控制位控制;TH0 作为 8 位定时器,由定时器 1 控制位控制。定时器 1 停止计数

2. T0 和 T1 的工作方式

定时器 T0 和 T1 都有 4 种工作方式。当工作于方式 0,1 时,T0 和 T1 的功能都是相同的,用户可以任意选用;但工作于方式 3 时,T0 和 T1 的功能就不相同。下面对各种工作方式的定时器结构及功能分别作详细的讨论。

(1) 方式 0

当 M1M0 为 00 时,定时器工作于方式 0。定时器被设置为 13 位的计数器,它由 TH1 和 TL1 的低 5 位组成,TL1 的高 3 位可忽略,如图 3-19 所示。当计数器的值为全“1”时,下次的增 1 即将计数器复 0,此时即出现溢出,定时器将溢出标志位 TF1 置为 1。图 3-19 电路是方式 0 中 T1 的结构框图。从图中可看出,工作方式由 $C/\overline{T}$ 控制,当 $C/\overline{T}=0$ 时,T1 以振荡周期的 12 分频信号作计数脉冲;当 $C/\overline{T}=1$ 时,从 T1 的外部输入端(P3.5)输入计数脉冲;当 $TR1=1,GATE=0$ 或 $\overline{INT1}=1$ 时,允许 T1 计数;当 $GATE=1$ 时,T1 由 $\overline{INT1}$ 控制,如 $TR1$ 和 $\overline{INT1}$ 同为“1”,允许 T1 计数;如 $TR1$ 为“0”或 $\overline{INT1}$ 为“0”时,开关断开,禁止 T1 计数。利用此功能,可测量脉冲的宽度。而控制位 $TR1$ 由 TCON.6 位控制。 $GATE$ 则由 TMOD.7 控制。当对运行标志 $TR1$ 置位时,不会将计数器清 0。

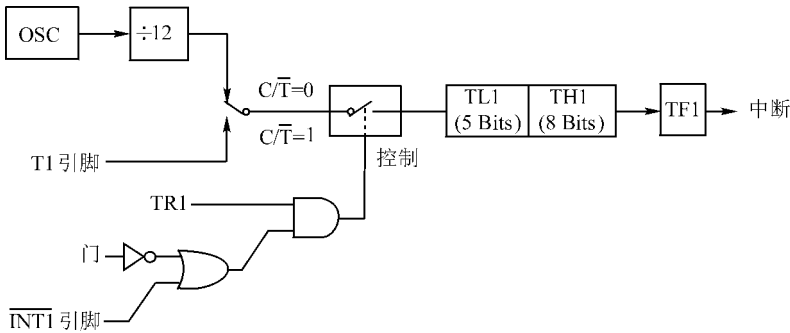


图 3-19 定时器/计数器 T1 方式 0 结构:13 位计数器

在方式 0 中,由于 T1 与 T2 的功能是完全相同的,所以在图 3-19 中,只要把 TR1,TF1,INT1 等信号换成 TR,TF0,INT0,就是 T0 的结构框图,使用时,对 TMOD 寄存器的设置应对应于 T0 的控制信号。

(2) 方式 1

方式 1 与方式 0 的差别仅是计数器的位数不同。当 T1 工作于方式 1 时,被设置为 16 位的计数器/定时器,由 TH1 和 TL1 两个 8 位寄存器组成。逻辑结构如图 3-20 所示。计数器的值从 0000H,0001H 开始到 0FFFFH,当从 0FFFFH 再增 1 时,计数器恢复为全 0,并且把溢出标记 TF1 置 1。计数器是连续计数的,而控制寄存器 TCON 中的溢出标志位 TF1 可由软件读出或写入。

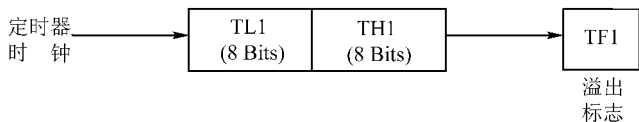


图 3-20 定时器/计数器 1 方式 1: 16 位计数器

(3) 方式 2

当 M1M0 为 10 时,定时器/计数器工作于方式 2。在此方式下,设置了一个 8 位的计数器,并具有自动恢复计数初值的功能,方式 2 的结构如图 3-21 所示。以 T1 为例,将 TL1 作计数器,将 TH1 作寄存器使用,存放计数初值。当 TL1 作增 1 计数至溢出时,除了把溢出标志位 TF1 置 1 外,同时还将 TH1 中的计数初值送入 TL1 中,使 TL1 又重新从初值开始计数。而 TH1 中的计数初值由软件编程置入,在常数重装的过程中,TH1 保持不变。在方式 2 中,T0 和 T1 的操作功能完全相同,可自由选择使用。

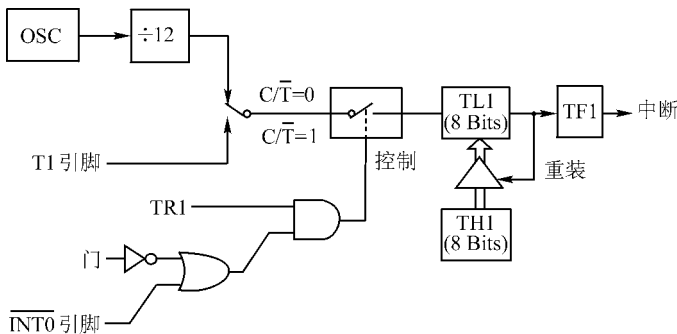


图 3-21 定时器/计数器 T1 方式 2 结构: 8 位常数自动重装

(4) 方式 3

在方式 3 中,定时器/计数器 T0 和 T1 的工作方式与前述的 2 种方式差别很大。在此方式下,T1 仅保持它原来的计数值,并停止计数,其效果相当于在电路中设置了控制位 TR1=0。所以,方式 3 只适用于 T0,此时,T0 的结构如图 3-22 所示,即将 T0 分成 2 个独立的 8 位计数器 TH0 和 TL0。其中,TL0 使用了 T0 的所有控制位:C/T,GATE,TR0,INT0 和 TF0。而 TH0 被固定设置为一个 8 位的定时器,对机器周期计数,并借用了 T1 的控制位 TR1 和 TF1。这样就由 TH0 控制了 T1 的中断标志位。

方式 3 适用于要求增加一个额外的定时器/计数器的场合。因为当 T0 工作于方式 3 时，在 AT89C51 中可出现 3 个定时器/计数器，而在 AT89C52 中可有 4 个。这时 T1 的使用可由方式控制位来控制，如定义为方式 0~方式 2，则允许计数，并且可以用来作串行口的波特率发生器以及在其他不需要中断的场合使用。

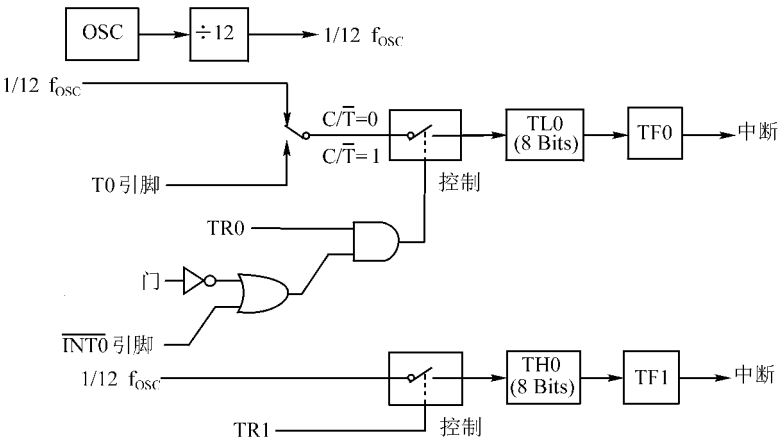


图 3-22 T0 方式 3 结构：2 个 8 位计数器

3. T0 和 T1 的工作编码

T0 和 T1 可以工作在各种不同的状态和方式下。定时器 T1 在作为定时器时有方式 0、方式 1、方式 2、方式 3 这 4 种方式；在作为计数器时也有 4 种方式。定时器 T1 分别用作定时器或计数器时，也各自有 4 种方式。

在不同的状态，即定时或计数状态下，采用不同的工作方式就有不同的编码。例如定时器 T0 作为定时器用时，有 4 种方式；而定时时钟可以是内部的或外部的，故有 8 种编码。表 3-10~表 3-13 给出了 T0 和 T1 在不同工作状态和方式时的编码。

表 3-10 T0 作为定时器时的编码

方 式	定时器 0 功能	TMOD	
		内部控制 ⁽¹⁾	外部控制 ⁽²⁾
0	13 位定时器	00H	08H
1	16 位定时器	01H	09H
2	8 位自动重载	02H	0AH
3	2 个 8 位定时器	03H	0BH

(1) 通过软件对 TR0 置位/清 0 来开启或停止定时器；
(2) 当 TR0=1 时，INT0=1，启动定时器；INT0=0，关闭定时器（硬件控制）。

表 3-11 T0 作为计数器时的编码

方 式	定时器 0 功能	TMOD	
		内部控制 ⁽¹⁾	外部控制 ⁽²⁾
0	13 位计数器	04H	0CH
1	16 位计数器	05H	0DH
2	8 位自动重装载	06H	0EH
3	1 个 8 位计数器	07H	0FH

- (1) 通过软件对 TR0 置位/清 0 来开启或停止定时器；
- (2) 当 TR0=1 时, $\overline{\text{INT0}}=1$, 启动定时器; $\overline{\text{INT0}}=0$, 关闭定时器(硬件控制)。

表 3-12 T1 作为定时器时的编码

方 式	定时器 1 功能	TMOD	
		内部控制 ⁽¹⁾	外部控制 ⁽²⁾
0	13 位定时器	00H	80H
1	16 位定时器	10H	90H
2	8 位自动重装载	20H	A0H
3	停止计数	30H	B0H

- (1) 通过软件对 TR1 置位/清 0 来开启或停止定时器；
- (2) 当 TR1=1 时, $\overline{\text{INT0}}=1$, 启动定时器; $\overline{\text{INT0}}=0$, 关闭定时器(硬件控制)。

表 3-13 T1 作为计数器时的编码

方 式	定时器 1 功能	TMOD	
		内部控制 ⁽¹⁾	外部控制 ⁽²⁾
0	13 位计数器	40H	C0H
1	16 位计数器	50H	D0H
2	8 位自动重装载	60H	E0H
3	不能用	—	—

- (1) 通过软件对 TR1 置位/清 0 来开启或停止定时器；
- (2) 当 TR1=1 时, $\overline{\text{INT1}}=1$, 启动定时器; $\overline{\text{INT1}}=0$, 关闭定时器(硬件控制)。

3.3.2 定时器 T2

在 AT89C52 单片机中,增加了一个 16 位定时器/计数器 T2。它的功能比其他 2 个定时器更强。在特殊功能寄存器组中有 5 个与 T2 有关的寄存器: TH2、TL2、定时器控制寄存器 T2CON、捕捉寄存器 RCAP2L、RCAP2H。T2 与 T0 和 T1 有类似的功能,即它也可以作定时

器或计数器使用。这取决于 T2CON 中的 C/ $\overline{T2}$ 控制位的值。另外,T2 还具有 3 种不同的工作方式,分别为:捕捉方式、自动常数重装入方式和波特率发生器方式。方式的选择由 T2CON 寄存器中的 D0,D2,D4,D5 位来定义。对应关系如表 3-14 所列。

表 3-14 定时器 T2 工作方式

RCLK+TCLK	CP/RL2	TR2	工作方式
0	0	1	16 位常数自动重装入
0	1	1	16 位捕捉方式
1	X	1	波特率发生器方式
X	X	0	停止计数

1. T2 的工作控制

T2 的工作是靠软件对 T2CON 寄存器进行设置而定义的。T2 的控制寄存器 T2CON 的格式如图 3-23 所示。

(MSB)				(LSB)			
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/ $\overline{T2}$	CP/RL2

图 3-23 T2 的控制寄存器 T2CON 格式

TF2:T2 的溢出中断标志。须由用户软件清 0,当 RCLK=1 或 TCLK=1 时,不允许将 TF2 置位。

EXF2:T2 外部中断标志。在捕捉方式和常数自动重装入方式下,当 EXEN2=1 时,在 T2EX 端发生的负跳变使 EXF2 置位。如此时 T2 中断被允许,则 EXF2=1 会使 CPU 响应中断。

RCLK:串行口接收时钟选择标志。当 RCLK=1 时,T2 工作于波特率发生器方式。此时,T2 的溢出脉冲作串行口方式 1 和方式 3 的接收脉冲,T1 提供发送的波特率;当 RCLK=0 时,T1 的溢出脉冲作接收时钟。

TCLK:串行口发送时钟选择标志。当 TCLK=1 时,T2 工作于波特率发生器,T2 的溢出脉冲作为串行口方式 1 和方式 3 的发送时钟,T1 提供发送波特率;当 TCLK=0 时,T1 的溢出脉冲作发送时钟。

EXEN2:T2 的外部允许标志。当 T2 工作于捕捉方式及常数自动重装入方式,且 EXEN2=1 时,如 T2EX 输入端上有一个负跳变信号,则会引发捕捉或常数重装入动作;而当 EXEN2=0 时,T2EX 端的电平变化对 T2 没有影响。

TR2:T2 的计数控制位。TR2=1,开始计数;TR2=0,禁止计数。

C/ $\overline{T2}$:定时器或计数器功能选择位。C/ $\overline{T2}$ =0,T2 为内部定时器(对振荡脉冲的 12 分频信号进行计数);C/ $\overline{T2}$ =1,T2 为外部事件计数器(下降沿触发)。

CP/ $\overline{RL2}$:捕捉或常数重装方式选择位。当 CP/ $\overline{RL2}$ =1 时,T2 工作于捕捉方式(即当 EXEN2=1时,T2EX 端的负跳变引发捕捉动作);当 CP/ $\overline{RL2}$ =0 时,为常数自动重装入方式(当 EXEN2=1 时,T2EX 端的负跳变引发常数重装入动作)。

2. T2 的工作方式

(1) 捕捉方式

当 $CP/\overline{RL2}=1$ 时, T2 工作于捕捉方式, 它的 16 位捕捉方式的结构原理如图 3-24 所示。T2CON 中的 EXEN2 位可控制 T2 以 2 种工作方式工作。如果 EXEN2=0, 则 T2 作定时器/计数器使用, 并由 $C/\overline{T2}$ 位决定它是作定时器或计数器。如作定时器使用, 则其计数输入为振荡脉冲的 12 分频信号; 作计数器时, 是以 T2 的外部输入引脚(P1. 0)上的输入脉冲作计数脉冲。当定时器/计数器 T2 增 1 计数至溢出后, 将 TF2 标志置 1, 并发出中断请求信号。在这种方式下, TL2 和 TH2 的内容不会送入捕捉寄存器中。

如果 EXEN2=1, T2 除实现上述定时器/计数器功能外, 还可实现捕捉功能, 即当外部输入端 T2EX(P1. 1)的输入电平发生负跳变时, 就会把 TH2 和 TL2 的内容锁入捕捉寄存器 RCAP2L 和 RCAP2H 中, 并将 T2CON 中的中断标志位 EXF2 置 1, 向 CPU 发出中断请求信号。

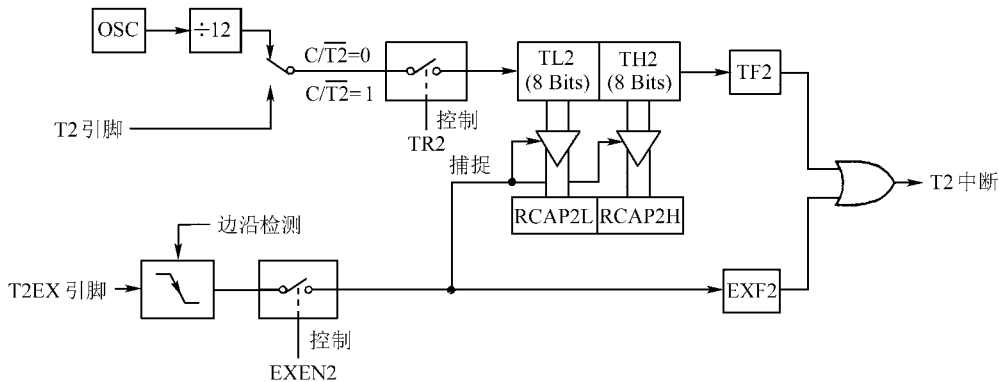


图 3-24 定时器 T2 的捕捉方式结构图

(2) 16 位常数自动重装入方式

当 $CP/\overline{RL2}=0$ 时, T2 工作于常数自动重装入方式, 它的结构原理如图 3-25 所示。

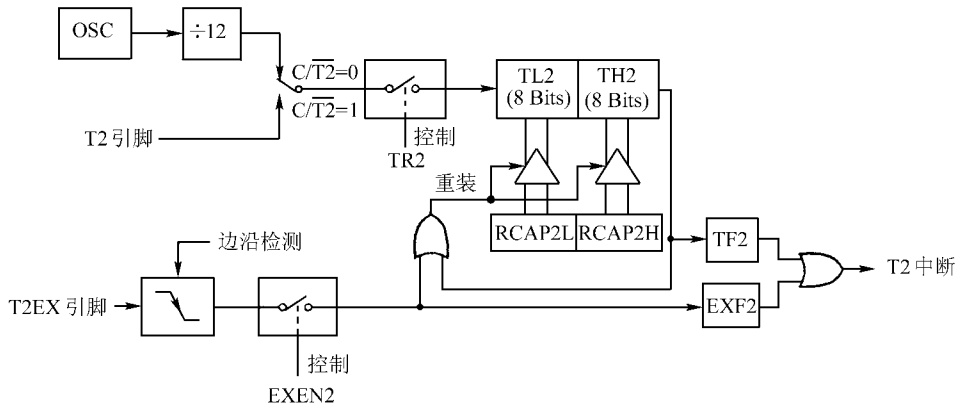


图 3-25 定时器 T2 的常数自动重装入方式结构图

如图 3-25 所示, T2 可工作于 2 种方式下, 由 T2CON 寄存器的 EXEN2 位控制选择。当 EXEN2=0 时, T2 作定时器/计数器用。当 $C/\overline{T2}=0$ 时, 作定时器用, 以振荡频率的 12 分频计数; 当 $C/\overline{T2}=1$ 时, 作计数器用, 以 T2 外部输入引脚(P1. 0)的输入脉冲作计数脉冲(下降沿触发)。当 TR2=1 时, 从初值开始增 1 计数, 计数至溢出时, 溢出信号控制打开三态门将 RCAP2L 和 TCAP2H 寄存器中存放的计数初值重新装入 TL2 和 TH2 中, 使 T2 从该值开始重新计数, 同时将溢出标志 TF2 置 1。计数器的初值在初始化时由软件编程置入。

当 EXEN2=1 时, T2 除可完成上述功能外, 还可实现以下功能。当外部输入引脚 T2EX (P1. 1)的输入电平发生负跳变时, 也可以控制将捕捉寄存器 RCAP2L 和 RCAP2H 的内容重新装入 TH2 和 TL2 中, 使 T2 重新从新值开始计数, 同时把中断标志 EXF2 置 1, 向 CPU 发出中断请求。

(3) 波特率发生器方式

T2 除可用于上述的工作方式外, 还可作串行口的波特率发生器。它的串行口波特率发生工作方式是由控制寄存器 T2CON 中的控制位 RCLK=1 或 TCLK=1 来确定的。具体的使用方法见 3.2.4 节介绍。

3. T2 的工作编码

定时器 T2 可以用作定时器, 也可以用作计数器, 而且, 无论是用作定时器或计数器都有捕捉方式和自动重装入方式。在 T2 作为定时器时, 还有波特率发生方式。

在这些方式和状态下的编码分别如表 3-15、表 3-16 所列。

表 3-15 给出的是 T2 作为定时器时的 3 种工作方式, 即自动重装入方式、捕捉方式和波特率发生方式的不同编码。

表 3-16 给出的是 T2 作为计数器时的 2 种工作方式, 即自动重装入方式、捕捉方式的不同编码。

表 3-15 T2 作为定时器时的编码

方 式	T2CON	
	内部控制 ⁽¹⁾	外部控制 ⁽²⁾
16 位自动重装载	00H	08H
16 位捕获	01H	09H
波特率发生器接收和 发送波特率相同	34H	36H
只接收	24H	26H
只发送	14H	16H

(1) 只有在定时器/计数器溢出时, 才发生捕获/重装载;
(2) 除了定时器 2 作为波特率发生器外, 定时器/计数器溢出和 T2EX(P1. 1)引脚出现信号负跳变时, 发生捕获/重装载。

表 3-16 T2 作为计数器时的编码

方 式	T2CON	
	内部控制 ⁽¹⁾	外部控制 ⁽²⁾
16 位自动重装载	02H	0AH
16 位捕获	03H	0BH

- (1) 只有在定时器/计数器溢出时,才发生捕获/重装载;
- (2) 除了定时器 2 作为波特率发生器外,定时器/计数器溢出和 T2EX(P1. 1)引脚出现信号负跳变时,发生捕获/重装载。

3.4 中断功能

所谓中断,即 CPU 在处理某件事情时,外部发生了某一事件,请求 CPU 马上处理。CPU 暂时停止当前的工作,转入处理外部事件程序,处理完后,再重回原来中断的地方,继续执行原来的工作,这一过程即为中断。如果微处理器具有一个良好的中断系统,则可提高 CPU 对外来异步事件的处理能力。在 AT89C51 内提供了 5 个中断请求源,AT89C52 有 6 个,还有 2 个中断优先级,可实现中断服务程序嵌套。在图 3-26 中列出了各中断类型及中断请求。

3.4.1 中断请求

在中断系统中,中断请求有两类:一类是外部中断请求;另一类是内部中断请求。无论是外部还是内部中断,都是由控制寄存器 TCON 的内容决定的。控制寄存器的内容和作用如图 3-14 所示。

1. 外部中断请求

$\overline{\text{INT0}}$ (P3. 2)和 $\overline{\text{INT1}}$ (P3. 3)2 个输入引脚为外部中断请求源。它们的触发方式有 2 种,分别为边沿触发和电平触发,由控制寄存器 TCON 中的低 4 位控制。控制寄存器中的有关控制位为 IE0,IT0,IE1,IT1。

外部请求 1($\overline{\text{INT1}}$)由 IE1,IT1 两位进行控制。它们的作用分别如下。

- (1) IE1:外部中断 1 请求源($\overline{\text{INT1}}$)标志。当检测到($\overline{\text{INT1}}$)上的中断请求信号时,将该标志置 1;当 CPU 响应该中断时由硬件将标志清 0。
- (2) IT1:外部中断源 1 触发方式控制位。IT1 = 0,外部中断 1 为电平触发方式,当($\overline{\text{INT1}}$)输入低电平时,置位 IE1;IT1 = 1,外部中断 1 为边沿触发方式,即($\overline{\text{INT1}}$)输入端上有一个从高变低的电平变化时将 IE1 置位。

外部请求 0($\overline{\text{INT0}}$)由 IE0,IT0 两位进行控制,其控制作用分别如下。

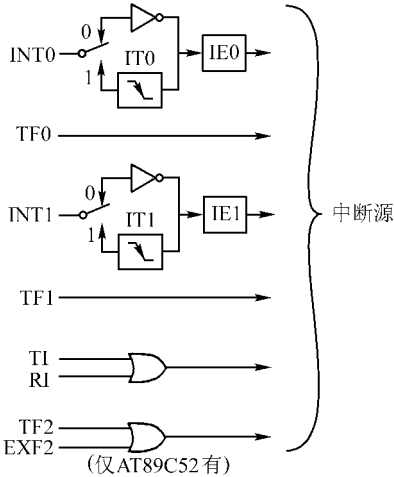


图 3-26 中断系统的中断请求

(1) IE0:外部中断 0 请求源($\overline{\text{INT0}}$)标志。功能与 IE1 类似,IE0 置 1 后,如 CPU 响应该中断请求,则由硬件将 IE0 清 0。

(2) IT0:外部中断 0 触发方式控制位。IT0=0,外部中断 0 为电平触发方式;IT0=1,为边沿触发方式。

2. 内部中断请求

(1) 定时器/计数器中断。定时器/计数器 T0 和 T1 的中断请求是在计数器循环计数的过程中产生溢出时,将溢出中断标志 TF0 和 TF1 置位而产生的。这 2 个中断标志也是在 TCON 寄存器中,当 T0 和 T1 产生定时器溢出时,如 CPU 响应该中断,则由硬件将标志清 0。

在 AT89C52 中,还增加了定时器 T2 的中断请求源,其中请求是由标志 TF2(T2CON. 7)和 EXF2(T2CON. 6)经逻辑或以后产生的。当 CPU 响应该中断请求后,必须由软件来判别是 TF2 或 EXF2 产生的中断,所以也必须由软件将该标志清 0。

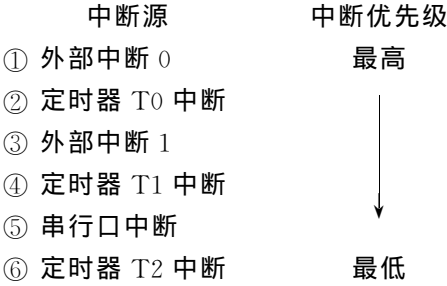
(2) 串行口中断。在串行口中包括有接收和发送 2 个中断请求源,而串行口的中断请求信号是由接收中断标志 RI 和发送中断标志 TI 经逻辑或后产生的(如图 3-26 所示)。当其中任何一个标志被置位时,都可以向 CPU 提出中断请求。事实上,CPU 在响应串行口中断时,并不清楚该中断是由 RI 或 TI 引起,所以必须在中断服务程序中判别,并由软件按 RI 或 TI 标志位清 0。

3.4.2 中断控制和优先

在 89 系列单片机中,所有的中断请求标志位都可以用软件置位和清 0,与硬件置位或清 0 的效果相同。也就是说中断请求可以由软件产生,也可以用软件取消。CPU 对每一个中断源的禁止与允许,都是通过内部的中断允许寄存器 IE 中的相应位来控制的,IE 寄存器的地址是 0A8H,如图 2-22 所示。

在 89 系列单片机的内部中断系统中设置了 2 个中断优先级,对于每一个中断源都可编程为高优先级中断或低优先级中断。它由中断优先级寄存器 IP 的相应位所定义,如对应位为“1”,即将该中断源定为高优先级。当系统复位时,IP 寄存器值为全“0”,即把所有中断请求源都定义为低优先级。一个正在被执行的低优先级的中断服务程序可以被另一个高优先级的中断源所中断,但不能被另一个同级的或低级的中断源所中断。中断优先级寄存器 IP 的地址为 0BAH,如图 2-23 所示。

当 CPU 同时接收到 2 个不同优先级的中断请求时,先响应高优先级的中断。如 CPU 同时接收到的是几个相同优先级的中断请求时,则由内部的硬件查询序列确定它们的优先服务次序。这样,在同一个优先级内,都有一个由内部查询序列确定的第二个优先结构,其排列如下:



注意:此优先级顺序只在 CPU 同时接收到几个相同优先级的中断请求时才有效。

另外,在中断寄存器 IE 及中断优先级寄存器 IP 中包含的未定义位,用户在使用时要注意不要对这些位进行写操作。

3.4.3 中断处理过程

CPU 在每一个机器周期顺序检查每一个中断源,在机器周期的 S5P2 进行采样,在下一个周期对采样到的中断请求信号按优先级处理。如果没有被下列条件所阻止,则在系统内由硬件产生一条子程序调用指令,转入被响应的中断服务程序入口。阻止中断响应的 3 个条件为:

- (1) CPU 正在处理同级或更高优先级的中断;
- (2) 当前的机器周期不是所执行指令的最后一个机器周期;
- (3) 正在执行的指令是中断返回指令 RETI 或者是对 IP,IE 寄存器的写操作指令。

以上条件中有一个存在,CPU 都会丢弃中断查询结果,阻止系统产生长调用操作。条件(2)是为了保证在转向中断服务程序之前,当前的指令被执行完毕。而条件(3)是保证如果当前执行的是 RETI 指令或者是写 IP,IE 指令,则 CPU 至少要再多执行一条指令以后,才能响应新的中断请求。

查询及产生长调用的时序如图 3-27 所示。

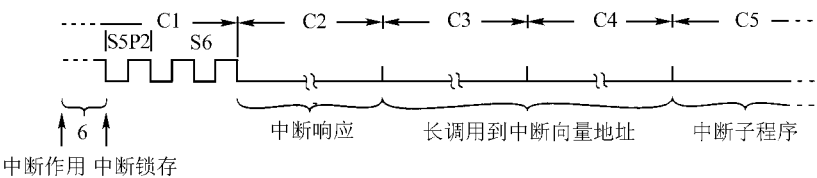


图 3-27 中断响应时序图

如果有一个高优先级的中断源在图 3-27 中机器周期 C3 的 S5P6 被激活,根据上述的规律,该中断请求会在本周周期被接受,并在机器周期 C5 和 C6 中转入中断服务程序入口。各种中断入口地址如表 3-17 所列。

表 3-17 各中断服务程序的入口地址表

中断源	标 志	入口地址
外部中断 0	IE0	0003H
定时器 T0	TF0	000BH
外部中断 1	IE1	0013H
定时器 T1	TF1	001BH
串行口中断	RI 或 TI	0023H
定时器 T2	TF2 或 EXF2	002BH
系统复位	RST	0000H

CPU 响应中断时,会执行一条由硬件产生的子程序调用指令(LCALL),控制转入相应的服务程序中,并先将中断标志清 0(串行口及定时器 T2 的中断标志要由软件清 0),把程序计

数器 PC 的内容压入堆栈(但不保存 PSW 内容),然后将该中断服务程序的入口地址送入 PC。各中断服务程序的入口地址的分配在表 3-17 中一一给出。

CPU 从入口地址开始执行相应的中断服务程序,直到最后执行一条 RETI 指令结束整个中断过程。在执行 RETI 指令时,CPU 控制将栈顶 2 个字节的内容弹出送入程序计数器 PC,然后 CPU 从原来的断点位置继续执行程序。但是,CPU 的现场保护和恢复必须由用户的中断服务程序实现。综前所述,整个中断响应过程 CPU 所执行的操作步骤如下。

- ① 完成当前指令的操作。
- ② 将 PC 内容压入堆栈。
- ③ 保存当前的中断状态。
- ④ 阻止同级的中断请求。
- ⑤ 将中断程序入口地址送 PC 寄存器。
- ⑥ 执行中断服务程序。

最后,当执行到 RITI 指令时即结束中断,从堆栈中弹出断点地址到 PC,返回原处继续执行原程序。

在中断请求过程中所产生的标志如表 3-18 所列。表中同时给出了标志所在的特殊功能寄存器及其位置。

表 3-18 中断标志位

中断源	标 志	在 SFR 寄存器中的位置
外部中断 0	IE0	TCON. 1
外部中断 1	IE1	TCON. 3
定时器 T0	TF1	TCON. 7
定时器 T1	TF0	TCON. 5
串行口中断	TI	SCON. 1
串行口中断	RI	SCON. 0
定时器 T2	TF2	T2CON. 7(AT89C52)
定时器 T2	EXF2	T2CON. 6(AT89C52)

3.4.4 中断过程的注意事项

1. 外部中断请求问题

外部中断请求源是由 TCON 寄存器的 IT1 和 IT0 位来定义它们的触发方式。当 $IT\times=0$ 时, $\overline{INT\times}$ 端的中断触发方式被置为电平触发方式。这时如果检测到 $\overline{INT\times}$ 的输入电平为低电平,则触发中断请求,即 $IE\times$ 标志置位。当 $IT\times=1$ 时,外部输入 $INT\times$ 为边沿触发方式。在这种方式下,如果在一个周期中对 $\overline{INT\times}$ 端的采样为高电平,而下一个周期的采样值为低电平,则将 $IE\times$ 置位,发出中断请求信号。由于外部中断输入引脚在每个机器周期中只被采样一次,所以输入引脚上的高电平或低电平信号应至少维持 12 个振荡周期的时间,以保证中断请求信号被检测到,并将 $IE\times$ 标志置位。当 CPU 响应中断后,将 $IE\times$ 标志清 0,而外部引脚

$\overline{\text{INTX}}$ 上的中断请求信号,必须在中断服务程序执行完毕前撤消,否则将激发另一次的中断。

2. 中断响应时间

外中断信号 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 在每个机器周期的 S5P2 被采样后经反相锁存入中断标志位 IE0 和 IE1 中。同样,定时器 T2 的标志 EXF2 和串行口中断标志 RI, TI 都是在每个机器周期的 S5P2 被置位。在下一个机器周期被查询电路查询。而定时器 T0 和 T1 的中断标志位 TF0 和 TF1 是在定时器产生溢出的周期的 S5P2 被置位,下一个机器周期被查询。定时器 T2 的中断标志 TF2 也是在溢出周期的 S5P2 被置位,但是在同一个机器周期被查询。

如果中断请求被激活且满足响应条件,CPU 将执行一条硬件子程序调用指令(LCALL),控制转入相应的中断服务程序入口,这个调用需要 2 个周期时间。这样,在外部中断请求信号产生到其中断服务程序的第一个指令被执行之间,至少需要 3 个完整的机器周期,在图 3-27 的时序中,也说明了中断响应所需要的时间。

如果中断请求被前面所列的 3 个条件之一所阻止,所需的响应时间就更长些。如果正在处理同等级或优先级更高的中断,那么等待的时间就取决于处理中的中断服务程序的过程。如果正在执行的程序指令不是在它的最后一个机器周期,则等待的时间不会超过 3 个机器周期,因为最长的指令只有 4 个机器周期(如 MUL 和 DIV 指令)。如果 CPU 正在执行的是 RETI 指令或者是访问 IP,IE 的指令,则等待时间不会多于 5 个周期。其中,1 个周期完成当前的指令,另外 4 个周期完成下一条指令(假设为最长的 MUL 或 DIV 指令)。所以,在一个单中断系统中,中断的响应时间为 3~9 个周期。

3. 利用外部中断执行单步操作

在 AT89C51 单片机的中断系统中,允许用户单步运行用户程序,可以使用户很方便地进行程序的调试。单步操作的原理是利用了中断系统的以下特点:当有中断请求时,如 CPU 正在处理同等优先级中断时,其请求不被接受,并且在执行中断返回指令 RETI 后,还必须执行多一条指令时,才可接受新的中断。利用此特性,就可以实现程序的单步运行控制。

典型的方法就是使用外部中断,在这里使用了外部中断 0,并将其设为电平触发方式。中断服务程序如下:

```
JNB      P3.2,$      ;等待 $\overline{\text{INT0}}$ 变高
JB       P3.2,$      ;等待 $\overline{\text{INT0}}$ 变低
RETI                                ;中断返回
```

现在,如果 $\overline{\text{INT0}}$ 端(P3.2)的输入为低电平,则 CPU 就会响应 $\overline{\text{INT0}}$ 中断并转入中断服务程序执行等待指令,等待 $\overline{\text{INT0}}$ 端变高电平,直到 $\overline{\text{INT0}}$ 有一个正脉冲输入(即 $\overline{\text{INT0}}$ 端从低变高再变低),然后才执行 RETI 指令返回原用户程序,执行一条指令又再次响应 $\overline{\text{INT0}}$ 中断(因此时 $\overline{\text{INT0}}$ 为低电平),执行中断服务程序,以等待 $\overline{\text{INT0}}$ 端的下一个正脉冲。这样,在 $\overline{\text{INT0}}$ 端每出现一个正脉冲,就会返回用户程序执行一条指令。如此重复,即可达到单步运行程序的目的。

3.5 振荡器、复位和省电方式

3.5.1 振荡器

89 系列单片机的内部振荡器电路如图 3-28 所示,由一个单级反相器组成。XTAL1 为

反相器的输入, XTAL2 为反相器的输出。可以利用它内部的振荡器产生时钟, 只要在 XTAL1 和 XTAL2 引脚上外接一个晶体及电容组成的并联谐振电路, 便构成一个完整的振荡信号发生器, 如图 3-30 所示, 此方式称为内部方式。另一种使用方法如图 3-29 所示, 由外部时钟源提供一个时钟信号到 XTAL1 端输入, 而 XTAL2 端浮空。在组成一个单片机应用系统时, 多数采用图 3-30 所示的方式, 这种方式的结构紧凑, 成本低廉, 可靠性高。

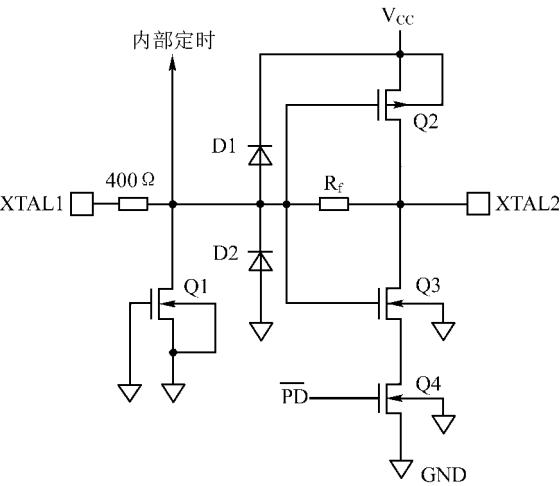


图 3-28 AT89C51 单片机内部振荡器电路

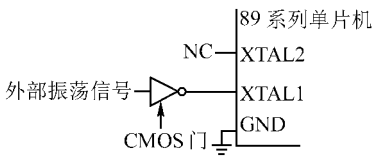


图 3-29 外部时钟接法

振荡器的等效电路如图 3-30 上部所示。在图中给出了外接元件, 即外接晶体及电容 C1、C2, 并组成并联谐振电路。在电路中, 对电容 C1 和 C2 的值要求不是很严格, 如使用高质的晶振, 则不管频率为多少, C1、C2 通常都选择 30 pF。有时, 在某些应用场合, 为了降低成本, 晶体振荡器可用陶瓷振荡器代替。如果使用陶瓷振荡器, 则电容 C1、C2 的值取 47 pF。

通常, 在单片机中对所使用的振荡晶体的参数要求如下。

ESR(等效串联电阻): 根据所需频率按图 3-31 选取。

C0(并联电容): 最大 7.0 pF。

CL(负载电容): 30 pF+3 pF。

通常, 其误差及温度变化的范围要按系统的要求来确定。

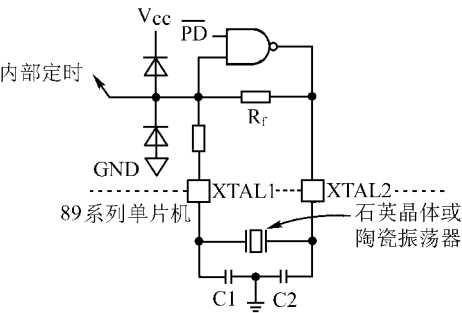


图 3-30 片内振荡器等效电路

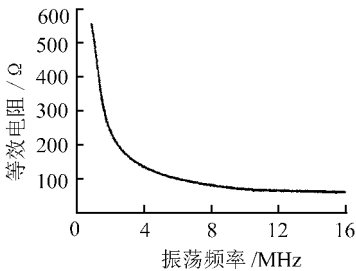


图 3-31 ESR 与频率的关系曲线

3.5.2 复 位

89 系列单片机与其他微处理器一样,在启动时都需要复位,使 CPU 及系统各部件处于确定的初始状态,并从初态开始工作。89 系列单片机的复位信号是从 RST 引脚输入到芯片内的施密特触发器中的。当系统处于正常工作状态时,且振荡器稳定后,如 RST 引脚上有一个高电平并维持 2 个机器周期(24 个振荡周期),则 CPU 就可以响应并将系统复位。复位时序如图 3-32 所示,因外部的复位信号是与内部时钟异步的,所以在每个机器周期的 S5P2 都对 RST 引脚上的状态采样。当在 RST 端采样到“1”信号且该信号维持 19 个振荡周期以后,将 ALE 和 PSEN 接成高电平,使器件复位。在 RST 端电压变低后,经过 1~2 个机器周期后退出复位状态,重新启动时钟,并恢复 ALE 和 PSEN 的状态。如果在系统复位期间将 ALE 和 PSEN 引脚拉成低电平,则会引起芯片进入不定状态。

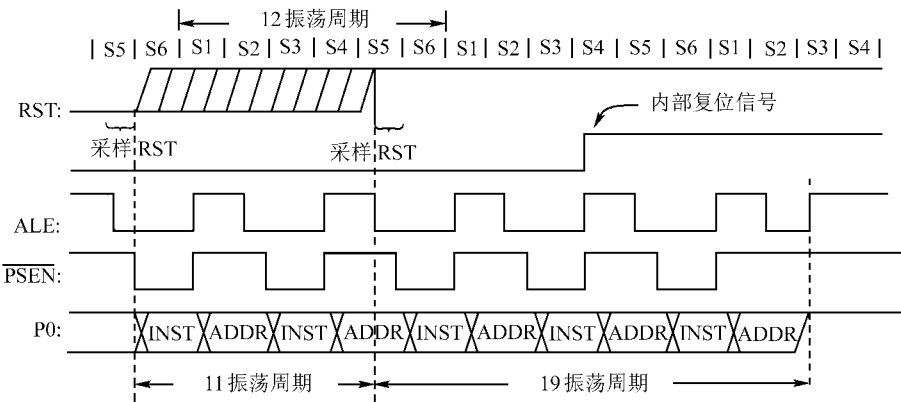


图 3-32 内部复位定时时序

1. 手动复位

手动复位需要人为在复位输入端 RST 上加入高电平。一般采用的办法是在 RST 端和正电源 V_{CC} 之间接一个按钮。当人为按下按钮时,则 V_{CC} 的 +5 V 电平就会直接加到 RST 端。由于人的动作很快也会使按钮保持接通达数十毫秒,所以,保证能满足复位的时间要求。手动复位的电路如图 3-33 所示。

2. 上电复位

AT89C51 的上电复位电路如图 3-34 所示,只要在 RST 复位输入引脚上接一电容至 V_{CC} 端,下接一个电阻到地即可。对于 CMOS 型单片机,由于在 RST 端内部有一个下拉电阻,故可将外部电阻去掉,而将外接电容减至 $1\ \mu\text{F}$ 。

上电复位的过程是在加电时,复位电路通过电容加给 RST 端一个短暂的高电平信号,此高电平信号随着 V_{CC} 对电容的充电过程而逐渐回落,即 RST 端的高电平持续时间取决于电容的充电时间。为了保证系统能够可靠地复位,RST 端的高电平信号必须维持足够长的时间。

上电时, V_{CC} 的上升时间约为 10 ms,而振荡器的起振时间取决于振荡频率,如晶振频率为 10 MHz,起振时间为 1 ms;晶振频率为 1 MHz,起振时间则为 10 ms。

在图 3-33 的复位电路中,当 V_{CC} 掉电时,必然会使 RST 端电压迅速下降到 0 V 以下,但

是,由于内部电路的限制作用,这个负电压将不会对器件产生损害。另外,在复位期间,端口引脚处于随机状态,复位后,系统将端口置为全“1”态。

如果系统在上电时得不到有效的复位,则在程序计数器 PC 中将得不到一个合适的初值,因此,CPU 可能会从一个未被定义的位置开始执行程序。

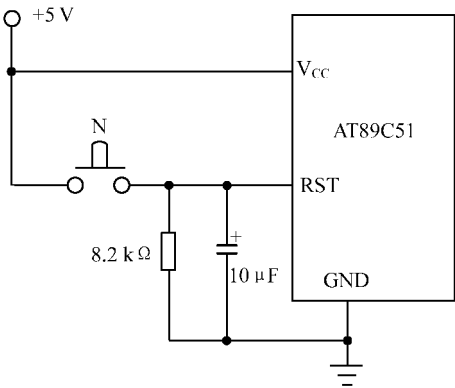


图 3-33 手动复位电路

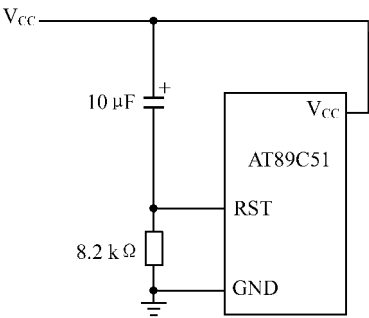


图 3-34 上电复位电路

3. 复位后寄存器的状态

当系统复位时,内部寄存器的状态如表 3-19 所列,即在 SFRS 中,除了端口锁存器、堆栈指针 SP 和串行口的 SBUF 外,其余的寄存器全部清 0,端口锁存器的复位值为 0FFH,堆栈指针值为 07H,SBUF 内为不定值。内部 RAM 的状态不受复位的影响,在系统上电时,RAM 的内容是不定的。

表 3-19 各特殊功能寄存器的复位值

专用寄存器	复位值	专用寄存器	复位值
PC	0000H	TCON	00H
ACC	00H	T2CON(AT89C52)	00H
B	00H	TH0	00H
PSW	00H	TL0	00H
SP	07H	TH1	00H
DPTR	0000H	TL1	00H
P0~P3	FFH	TH2(AT89C52)	00H
IP(AT89C51)	×××00000B	TL2(AT89C52)	00H
IP(AT89C52)	××000000B	RCAP2H(AT89C52)	00H
IE(AT89C51)	0××00000B	RCAP2L(AT89C52)	00H
IE(AT89C52)	0×000000B	SCON	00H
TMOD	00H	SBUF	不定
T2MOD(AT89C52)	××××××00B	PCON(CHMOS)	0×××0000B

3.5.3 省电方式

89 系列单片机提供了 2 种省电工作方式:空闲方式和掉电方式。目的是尽可能地降低系统的功耗。在省电工作方下, V_{CC} 是由后备电源提供,其内部控制电路如图 3-35 所示。在空闲工作方式中 ($IDL = 1$), 振荡器继续工作,时钟脉冲继续输出到中断系统、串行口以及定时器模块,但却不提供给 CPU。在掉电方式中 ($PD = 1$), 振荡器停止工作。两种工作方式都是由 SFR 中的电源控制寄存器 PCON 的控制位来定义,PCON 寄存器的控制格式如图 3-36 所示。

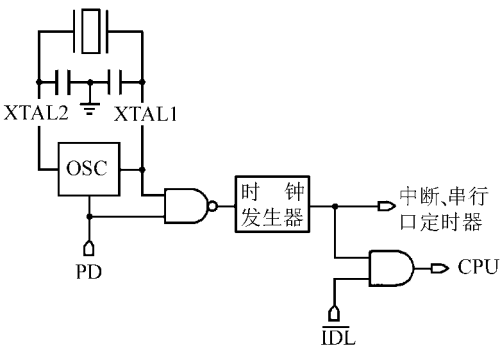


图 3-35 空闲和掉电方式控制电路

SMOD: 串行口波特率倍率控制位 (详见串行口波特率一节)。
GF1: 通用标志位。
GF0: 通用标志位。

(MSB)				(LSB)			
SMOD	—	—	—	GF1	GF0	PD	IDL

图 3-36 PCON 寄存器

PD: 掉电方式控制位。PD=1 进入掉电工作方式。
IDL: 空闲方式控制位。ILD=1, 进入空闲工作方式。
如同时将 PD 和 IDL 置 1, 则进入掉电工作方式。PCON 寄存器的复位值为 0XXX000, PCON. 4~PCON. 6 为保留位。用户不要对它们做写操作。

1. 空闲工作方式
- 当 CPU 执行完置 $IDL=1$ (PCON. 1) 的指令后, 系统进入了空闲工作方式。这时, 内部时钟不向 CPU 提供, 而只供给中断、串行口、定时器部分。CPU 的内部状态维持, 即包括堆栈指针 SP、程序计数器 PC、程序状态字 PSW、累加器 ACC, 其他所有的内容保持不变, 端口状态也保持不变。ALE 和 $\overline{PSEN}$ 保持逻辑高电平。
- 进入空闲方式后, 有两种方法可以使系统退出空闲方式。
- 一是任何的中断请求被响应都可以由硬件将 PCON. 0 (IDL) 清 0 而中止空闲工作方式。当执行完中断服务程序返回时, 从置空闲工作方式指令的下一条指令开始继续执行程序。另外, PCON 寄存器中的 GF0 和 GF1 标志可用来指示中断是否在正常情况下或在空闲方式下发生。例如, 在执行置空闲方式的指令前, 先置标志位 GF0 (或 GF1); 当空闲工作方式被中断中止时, 在中断服务程序中可检测标志位, 以判断出系统是在什么情况下发生中断, 如 GF0 (GF1) 为 1, 则是在空闲方式下进入中断。
- 另一种退出空闲方式的方法是硬件复位, 由于在空闲工作方式下振荡器仍然工作, 因此硬件复位仅需 2 个机器周期便可完成。而 RST 端的复位信号直接将 PCON. 0 (IDL) 清 0, 从而退出空闲状态, CPU 则从进入空闲方式的下一条指令开始重新执行程序。如图 3-32 所示, 在内部系统复位开始, 还可以有 2~3 个指令周期, 在这一段时间里, 系统硬件禁止访问内部

RAM 区,但允许访问端口。但是,为了防止对端口的操作出现错误,在置空闲工作方式指令的下一条指令中,不应该为写端口或写外部 RAM 的指令。

2. 掉电工作方式

当 CPU 执行一条置 PCON.1 位(PD)为 1 的指令后,系统进入掉电工作方式。在这种工作方式下,内部振荡器停止工作。由于没有振荡时钟,因此,所有的功能部件都停止工作。但内部 RAM 区和特殊功能寄存器的内容被保留,而端口的输出状态值都保存在对应的 SFR 中,ALE 和 $\overline{\text{PSEN}}$ 都为低电平。

退出掉电方式的惟一一个方法是由硬件复位,复位后将所有的特殊功能寄存器的内容初始化,但不改变内部 RAM 区的数据。

在掉电工作方式下, V_{CC} 可以降到 2 V,但在进入掉电方式之前, V_{CC} 不能降低。而在准备退出掉电方式之前, V_{CC} 必须恢复正常的工作电压值,并维持一段时间(约 10 ms),使振荡器重新启动并稳定后方可退出掉电方式。

第四章 指令系统

89 系列单片机是和 MCS-51 兼容的单片机。它的结构和指令系统使其适合控制应用。它为访问内部 RAM 提供了很多快速的寻址方式,便于对小型数据结构进行字节操作。该指令系统还支持位变量作为一个单独的数据类型,在需要布尔变量处理的控制和逻辑系统中,允许直接对位进行操作。

这一章,介绍指令的寻址方式以及指令系统中的各类指令,最后给出指令总表。

4.1 指令寻址方式及标志位

在执行指令时,指令需要一定的数据参与操作,要解决如何取数据的问题。取数据在本质上就是要给出数据所在的地址,因此,取数据就转化成寻址的问题。为了能灵活地进行数据存取,故对指令的工作设置了各种各样的寻址方法。

指令在执行的过程中,其产生的状态和结果依赖于指令的操作及相关的数。这些状态和结果有进位标志 CY、辅助进位标志 AC 和溢出标志 OV。这 3 个标志分别位于程序状态字 PSW 的 D7、D6 和 D2 位。

4.1.1 指令寻址方式

指令寻址时,在指令格式中用一些符号来描述地址或数据。这些符号和意义如下。

Rn: 当前选择的寄存器组合中的寄存器 R7~R0。

direct: 8 位内部数据的地址。它可以是一个内部数据 RAM 位置(0~127)或是一个 SFR [例如: I/O 端口、控制寄存器、状态寄存器等地址(128~255)]。

@Ri: 间接由 R1 或 R0 寄存器寻址的 8 位内部数据 RAM 位置(0~255)。

#data: 包含在指令中的 8 位常数。

#data 16: 包含在指令中的 16 位常数。

#addr 16: 16 位目的地址,在 LCALL 和 LJMP 指令中使用,可以转移到 64 KB 程序内存地址中的任何地址。

addr 11: 11 位目的地址,在 ACALL 和 AJMP 指令中使用,可以转移到 2 KB 页面的程序内存中的任何地址。

rel: 有符号的 8 位偏移量字节(二进制补码),在 SJMP 和所有的条件转移指令中使用。范围是 -128~+127,与接下来指令中的第一个字节有关。

Bit: 在内部数据 RAM 中的立即寻址位或在特殊功能寄存器中的立即寻址位。

在 89 系列单片机中,指令系统的寻址方式有如下 6 种。

1. 直接寻址

在直接寻址(Dir)方式中,操作数的有效地址处于指令操作码的后继一个字节中。采用直接寻址方式的指令是双字节指令。只有内部 RAM 的低 128 字节和专用寄存器才能采用直接寻址方式。由于内部 RAM 高 128 字节的地址与专用寄存器的地址是重叠的,故直接寻址方式就不能用于内部 RAM 的高 128 字节。

2. 间接寻址

在间接寻址(Ind)方式中,指令指定某个寄存器来存放操作数的地址。内部和外部 RAM 都可采用这种寻址方式,但不能用这种寻址方式对专用寄存器进行访问。

对 8 位地址,可采用栈指针寄存器或工作寄存器 R0 或 R1 作为间址寄存器,可对 256 字节的单元进行寻址;对 16 位地址,只能采用 16 位的数据指针寄存器(DPTR)作为间址寄存器,可对 64 KB 的存储器空间寻址。

3. 寄存器寻址

寄存器寻址(Reg)方式可用于访问选定寄存器组的 8 个工作寄存器 R0~R7。这时指令操作码的低 3 位用来表示所用的寄存器。由于这种寻址方式可省去一个地址字节(被寻址寄存器的内容就是操作数),故用这种访问寄存器的指令可缩短程序长度。指令执行时,访问选定寄存器组 8 个寄存器中的一个,而寄存器组则由 PSW 中寄存器组选择控制位(RS1,RS0)来确定。

4. 特殊功能寄存器寻址

一些指令是专门用于某些寄存器的。例如,有些指令总是对累加器进行操作,因此就不再需要给出其地址。在这些情况下,操作码本身就确定了某个寄存器。凡是涉及到累加器 A 的指令在汇编时都对 A 给一个专用的操作码。

5. 立即寻址

在立即寻址(Imm)方式中,跟在操作码后的字节就是实际的操作数。采用立即寻址方式的指令是双字节的。第一个字节是操作码,第二个字节是立即操作数。因此,操作数就是放在程序存储器内的常数。

例如指令 MOV A, #100 表示把十进制数 100 传送给累加器 A,十进制数 100 也可用十六进制数 64H 来表示。在指令的数据前加 #,表示为立即数。

6. 变址寻址

这种寻址方式用于访问程序存储器,尤其适用于查表访问。用一个 16 位寄存器(DPTR 或 PC)作为基址寄存器,存放表头的起始地址;累加器 A(变址寄存器)存放查表的项数。项地址就是基址寄存器(DPTR 或 PC)加上变址寄存器(A)的和。

变址寻址方式的另一种情况是用在条件转移指令中。在这种情况下,转移指令的目标地址就是基址寄存器加上变址寄存器的和。

4.1.2 指令的状态标志

在指令执行时,对程序状态字 PSW 中的有关状态标志位 CY,AC,OV 产生影响。在指令系统中,有关指令以及其对状态标志的影响如表 4-1 所列。

表 4-1 影响状态标志的指令

指 令	标 志			指 令	标 志		
	CY	OV	AC		CY	OV	AC
ADD	×	×	×	CLR C	0		
ADDC	×	×	×	CPL C	×		
SUBB	×	×	×	ANL C, bit	×		
MUL	0	×		ANL C, /bit	×		
DIV	0	×		ORL C, bit	×		
DA	×			ORL C, bit	×		
RRC	×			MOV C, bit	×		
RLC	×			CJNE	×		
SETB C	×						

4.2 指令系统分类说明

指令系统分成算术操作类、逻辑操作类、数据传送类、布尔变量操作类、控制程序转移类等 5 类指令。下面分别说明各类指令的含义和操作。

4.2.1 算术操作类指令

算术操作类指令如表 4-2 所列。它一共有 11 条指令,都是执行算术运算中的加、减、乘、除操作,或加 1、减 1 操作的。

表 4-2 算术操作类指令

助记符	功 能	寻址方式				执行时间/ μs
		Dir	Ind	Reg	Imm	
ADD A,<byte>	$A = A + \text{<byte>}$	×	×	×	×	1
ADDC A,<byte>	$A = A + \text{<byte>} + C$	×	×	×	×	1
SUBB A,<byte>	$A = A - \text{<byte>} - C$	×	×	×	×	1
INC A	$A = A + 1$	累加器				1
INC <byte>	$\text{<byte>} = \text{<byte>} + 1$	×	×	×		1
INC DPTR	$\text{DPTR} = \text{DPTR} + 1$	数据指针				2
DEC A	$A = A - 1$	累加器				1
DEC <byte>	$\text{<byte>} = \text{<byte>} - 1$	×	×	×		1
MUL AB	$B : A = B \times A$	ACC 和 B				4
DIV AB	$A = \text{Int}(A/B) \quad B = \text{Mod}(A/B)$	ACC 和 B				4
DA A	十进制调整	累加器				1

表 4-2 中用<byte>表示每种指令可采用的操作数寻址方式。例如指令 ADD A,<byte>可表示：

ADD A, 7FH (直接寻址)

ADD A, @R0 (间接寻址)

ADD A, R7 (寄存器寻址)

ADD A, #127 (立即寻址)

表 4-2 所给出的执行时间是在假定晶振频率为 12 MHz 的情况下得到的。在这些指令中,除 INC DPTR 指令要用 2 μ s,乘法和除法指令要用 4 μ s 以外,其他指令都只需要 1 μ s。

注意,不用累加器 A 也可对内部 RAM 中任何一个单元的内容实现加 1 或减 1 操作。

指令 INC DPTR 是对数据指针的内容执行加 1 操作,数据指针(DPTR)可产生 16 位的外部存储器地址,故可对 64 KB 的外部存储空间进行寻址。

乘法指令 MUL AB 是完成累加器 A 的内容与寄存器 B 的内容相乘,并将 16 位的结果放回 A 和 B 中,低 8 位放在 A 中,高 8 位放在 B 中。

除法指令 DIV AB 是完成累加器 A 的内容除以寄存器 B 的内容,并将 8 位的商放在 A 中,而 8 位的余数放在 B 中。

其实除法指令在数制转换和移位操作方面的用处也很大,例如在移位操作中,要实现将一个数除以 2ⁿ 的运算就必须将该数右移 n 位。如果用 DIV AB 指令来进行这个运算,则可在 4 μ s 内完成其移位操作,并将被移出的数放在寄存器 B 中。

DA A 指令是对 2 个 BCD 数相加后的结果进行调整,使它也调整为 2 位 BCD 码的数,因此,在执行对 BCD 数的 ADD 和 ADDC 运算后,紧接着要执行一条 DA A 指令,以保证运算的结果也为 BCD 数。

4.2.2 逻辑操作类指令

逻辑操作类指令一共有 16 条,如表 4-3 所列。

表 4-3 逻辑操作类指令

助记符	功 能	寻址方式				执行时间/ μ s
		Dir	Ind	Reg	Imm	
ANL A,<byte>	A=A. AND. <byte>	×	×	×	×	1
ANL <byte>,A	<byte>=<byte>. AND. A	×				1
ANL <byte>,#data	<byte>=<byte>. AND. #data	×				2
ORL A,<byte>	A=A. OR. <byte>	×	×	×	×	1
ORL <byte>,A	<byte>=<byte>. OR. A	×				1
ORL <byte>,#data	<byte>=<byte>. OR. #data	×				2
XRL A,<byte>	A=A. XOR. <byte>	×	×	×	×	1
XRL <byte>,A	<byte>=<byte>. XOR. A	×				1
XRL <byte>,#data	<byte>=<byte>. XOR. #data	×				2
CLR A	A=00H	累加器				1

续表 4-3

助记符	功 能	寻址方式				执行时间/ μs
		Dir	Ind	Reg	Imm	
CPL A	A=NOT. A	累加器				1
RL A	累加器左环移位	累加器				1
RLC A	累加器连进位标志左环移位	累加器				1
RR A	累加器右环移位	累加器				1
RRC A	累加器连进位标志右环移位	累加器				1
SWAP A	累加器高 4 位与低 4 位交换	累加器				1

在表 4-3 中也列出了可用来访问操作数<byte>的寻址方式。例如,指令 ANL A,<byte>可有下面几种形式:

```
ANL A,7FH      (直接寻址)
ANL A,@R1      (间接寻址)
ANL A,R6       (寄存器寻址)
ANL A,#53H     (立即寻址)
```

所有专用累加器的逻辑操作指令执行时间都为 1 μs (晶振频率为 12 MHz),其他指令的执行时间为 2 μs 。

注意,不用累加器也可对直接寻址的内部 RAM 中低 128 字节区的任一单元或专用寄存器的指定位进行布尔变量操作。例如,从下面这个例子中可看到,指令 XRL<byte>,#data 可提供一种快速和方便地对端口位方式进行取反控制的方法。

```
XRL P1,#0FFH
```

执行这条指令后的结果使端口 P1 原来为“0”的位变“1”,而原来为“1”的位变为“0”。

如果是在响应中断时进行这种操作,不用累加器还可节省时间,因为在服务子程序中不需将累加器的内容入栈。

循环指令(RL A,RLC A)等可使累加器的内容向左或向右移一位。循环左移是将最高有效位(MSB)移到最低有效位(LSB)位置,而循环右移则是将 LSB 移到 MSB 位置。

SWAP A 指令是将累加器 A 中的高 4 位和低 4 位交换。这种交换在 BCD 码操作中非常有用。例如,已知在累加器中有一个小于 100 的二进制数,用下面一段程序就可很快将它转换为 BCD 码。

```
MOV B,#10
DIV AB
SWAP A
ADD A,B
```

说明:二进制数转换为 BCD 码的一般算法是把二进制数除以 1 000,100,10 等 10 的各次幂,所得的商即为千、百、十位数,余数为个位数。在上面这段程序中,用 DIV 指令分离出十位数和个位数,十位数放在累加器 A 中的低 4 位,个位数放在寄存器 B 中,用 SWAP 指令把十位数换到 A 的高 4 位中,最后执行 ADD A,B 指令后,便可得到一个 BCD 码。

4.2.3 数据传送类指令

1. 内部 RAM 传送指令

在内部 RAM 中存取数据的指令实际上是在寄存器 A 和 RAM 的存储单元之间传送数据。这些传送指令和寻址指令如表 4-4 所列。这些指令传送速度较快,当振荡频率为 12 MHz 时,它们的执行时间为 1 μ s 或 2 μ s。

表 4-4 数据传送指令

助记符	功 能	寻址方式				执行时间/ μ s
		Dir	Ind	Reg	Imm	
MOV A,<src>	A=<src>	×	×	×	×	1
MOV <dest>,A	<dest>=A	×	×	×		1
MOV <dest>,<src>	<dest>=<src>	×	×	×	×	2
MOV DPTR,#data16	DPTR=16 位立即数				×	2
PUSH <src>	INC SP;MOV"@SP",<src>	×				2
POP <dest>	MOV <dest>,"@SP";DEC SP	×				2
XCH A,<byte>	ACC 和<byte>交换数据	×	×	×		1
XCHD A,@Ri	ACC 和@Ri 低 4 位交换		×			1

MOV <dest>,<src>指令能实现内部 RAM 之间、专用寄存器之间或专用寄存器与内部 RAM 之间的直接数据传送,不需要经过累加器。

注意,在所有 ATMEL 的 Flash 单片机中,堆栈是设置在片内的 RAM 中,并可向上升。PUSH 指令首先使栈指针(SP)加 1,然后再把一个字节压入到当前 SP 指定地址的堆栈单元中去。PUSH 和 POP 指令只用于把直接寻址的一个字节压入栈顶,或把栈顶的内容传送到直接寻址的一个单元中去,而堆栈的位置是不固定的,可以通过栈指针(SP)来确定栈顶的地址,即用间接寻址方式来访问堆栈。如果有高 128 字节区,堆栈也可设置在高 128 字节区中,但不能设置在专用寄存器(SFR)区。在没有高 128 字节区的单片机中,如果 SP 指向高 128 字节,那么在执行 PUSH 指令时,要压入栈顶的字节就会丢失;在执行 POP 指令时,则要从栈顶弹出的字节不能确定。

数据传送类指令中,包含一条 16 位的传送指令。在程序存储器中进行查表访问或要对整个外部数据存储器空间(64 KB)寻址时,用这条指令来设置 DPTR 的起始值。

XCH A,<byte>指令完成将累加器中的内容与所寻址单元的内容交换。XCHD A,@Ri 指令则只进行低 4 位内容的交换。

要知道 XCH 和 XCHD 指令是如何进行数据处理的,请看一个将 8 位 BCD 数向右移 2 位的例子。图 4-1 对用直接传送指令和用 XCH 指令来完成这种情况的情况作了一个比较。每条指令的旁边都给出了存放 BCD 数寄存器的内容和累加器的内容,表示执行指令后的状态。

执行这段程序后,累加器中的内容就是被移出的两个数。从这个例子可看到,用直接传送指令需要 14 个指令字节,执行时间为 9 μ s;若用 XCH 指令则只需要 9 个指令字节,执行时间为

5 μs 。对于同样的操作,用 XCH 指令所需的指令字节少,而且执行时间也几乎要快 1 倍。

如果要将 1 个 BCD 数向右移奇数位,就必然要执行 1 次将 BCD 数右移 1 位的操作。下面给出一段用 XCHD 指令实现将 1 个 BCD 数右移 1 位的程序,如图 4-2 所示。

	2A	2B	2C	2D	2E	ACC
MOV A,2EH	00	12	34	56	78	78
MOV 2EH,2DH	00	12	34	56	56	78
MOV 2DH,2CH	00	12	34	34	56	78
MOV 2CH,2BH	00	12	12	34	56	78
MOV 2BH,#0	00	00	12	34	56	78
采用直接传送指令:14 字节,9 μs						
	2A	2B	2C	2D	2E	ACC
CLR A	00	12	34	56	78	00
XCH A,2BH	00	00	34	56	78	12
XCH A,2CH	00	00	12	56	78	34
XCH A,2DH	00	00	12	34	78	56
XCH A,2EH	00	00	12	34	56	78
采用 XCH 指令:9 字节,5 μs						

图 4-1 1 个 BCD 码右移 2 位

	2A	2B	2C	2D	2E	ACC
MOV R1,#2EH	00	12	34	56	78	×
MOV R0,#2DH	00	12	34	56	56	×
loop for R1=2EH:						
LOOP:MOV A,@R1	00	12	34	56	78	78
XCHD A,@R0	00	12	34	58	78	76
SWAP A	00	12	34	58	78	67
MOV @R1,A	00	12	34	58	67	67
DEC R1	00	12	34	58	67	67
DEC R0	00	12	34	58	67	67
CJNE R1,#2AH,LOOP						
loop for R1=2DH:	00	12	38	45	67	45
loop for R1=2CH:	00	18	23	45	67	23
loop for R1=2BH:	08	01	23	45	67	01
CLR A	08	01	23	45	67	00
XCH A,2AH	00	01	23	45	67	08

图 4-2 1 个 BCD 码右移 1 位

在这个例子中,首先将指针 R1 和 R0 指向存放最低 4 位 BCD 数的两个单元(2EH, 2DH),然后从 2EH(该单元内存放要移位的 BCD 数的最后 2 位)开始执行循环处理程序。每

项循环 1 次,指针减 1,直到指针 R1 指向 2AH。此时,最右边的数已被移出,并放在 2AH 单元中,按照移位规则,1 个数右移 1 位后,最左边一位应该为 0,所以最后两条指令是完成把 2AH 单元的内容送入累加器 A,并对 2AH 单元清 0。这样就完成了将 1 个 BCD 数右移 1 位的操作,被移出的数放在累加器 A 中。

2. 外部 RAM 传送指令

在表 4-5 中列出了用于访问外部数据存储器的数据传送指令。这些指令只能采用间接寻址方式。其中,@Ri 为单字节地址,Ri 可以是选定寄存器组的 R0 或 R1;@DPTR 为双字节地址。当外部 RAM 只有几 KB 时,如果采用 16 位地址,则 P2 端口的 8 条口线全部都要用来作地址线;如果采用 8 位地址,则不需要将 P2 端口的全部口线用完,可省下一些口线留作它用。

表 4-5 外部 RAM 传送指令

地址长度/位	助记符	功 能	执行时间/ μ s
8	MOVX A,@Ri	外部 RAM 内容送累加器	2
8	MOVX @Ri,A	累加器内容送外部 RAM	2
16	MOVX A,@DPTR	外部 RAM 内容送累加器	2
16	MOVX @DPTR,A	累加器内容送外部 RAM	2

注意,在访问外部 RAM 时,目的操作数或源操作数总是放在累加器 A 中。另外,只在执行 MOVX 指令期间,才会产生外部 RAM 的读和写选通信号,平时这些信号是不会出现的。如果根本不打算用到这些信号,则这些信号的引脚还可用作外部 I/O 线。

3. 查表指令

表 4-6 中给出的两条 MOVC 指令是程序存储器内的查表指令,用来实现程序存储器到累加器的代码或常数传送,每次传送一个字节。由于这两条指令只访问程序存储器,故表格中的数据只能读出而不能改写。“传送常数”的助记符是 MOVC。

表 4-6 查表指令

助记符	功 能	执行时间/ μ s
MOVC A,@A+DPTR	程序代码送累加器(相对数据指针)	2
MOVC A,@A+PC	程序代码送累加器(相对程序计数器)	2

如果要查找放在外部程序存储器中的表格,那么读选通信号是 $\overline{\text{PSEN}}$ (外部程序存储器允许)。

用表 4-6 中的第一条指令查表时,表中的数据项可达到 256 个(编号为 0~255)。查表时,先把要查找项的项数装入累加器 A,表头的起始地址装入数据指针(DPTR)中,然后执行这条指令,将查找到的内容送到累加器中。

表 4-6 中第二条指令的执行情况与第一条指令相仿,只是表头的起始地址放在程序计数器(PC)中,通过子程序来查表。查表时,首先把要查找项的项数装入累加器中,然后调用子程序。

下面来看一个用 MOVC A,@A+PC 指令来查表的例子。

```
MOV    A,ENTRY-NUMBER
CALL   TABLE
TABLE:MOVC A,@A+PC
RET
```

在程序存储器中,表格就跟在 RET(返回)指令后面。这种表格可含有 255 个数据项(编号为 1~255),0 编号不能使用,因为在执行 MOVC 指令时,PC 中放的是 RET 指令的地址,编号为 0 的项也就是 RET 本身的操作码。

4.2.4 布尔变量操作类指令

ATMEL 公司生产的 Flash 单片机中,有一个完整的布尔处理器,即位处理机。数据存储器的内部 RAM 区中,有 128 位是可直接寻址的,如图 2-9 所示。专用寄存器区中有一些单元(这些单元的直接地址可被 8 除尽)的各位(除 IP. 7,IP. 6,IP. 5,IE. 6 和 IE. 5 以外)也是可位寻址的,如图 2-10 所示。另外所有的端口线也都是可位寻址的。这些可寻址位就构成了布尔处理机。所以在指令系统中有一个专用处理布尔变量的指令子集,包括有布尔变量的传送、逻辑运算、置位、清除、控制程序转移等指令,如表 4-7 所列。

表 4-7 布尔变量操作类指令

助记符	功 能	执行时间/ μ s
ANL C, bit	C=C. AND. bit	2
ANL C, /bit	C=C. AND. NOT. bit	2
ORL C, bit	C=C. OR. bit	2
ORL C, /bit	C=C. OR. NOT. bit	2
MOV C, bit	C=bit	1
MOV bit, C	bit=C	2
CLR C	C=0	1
CLR bit	bit=0	1
SETB C	C=1	1
SETB bit	bit=1	1
CPL C	C=, NOT. C	1
CPL bit	bit=, NOT. bit	1
JC rel	C=1 转移	2
JNC rel	C=0 转移	2
JB bit, rel	bit=1 转移	2
JNB bit, rel	bit=0 转移	2
JBC bit, rel	bit=1 转移; CLR bit	2

所有位操作都是通过直接寻址方式进行的。位地址可以是内部 RAM 20H~2FH 单元中连续的 128 位(位地址为 00H~7FH),也可以是专用寄存器区中的可寻址位,分布在 80H~FFH 范围内,但不是连续的。

下面举一个例子看看如何将一个内部标志传送到端口的某一位。

```
MOV C,FLAG
MOV P1.0,C
```

程序中,FLAG 可以是内部 RAM 中(20H~2FH 单元)的位地址或专用寄存器区中的位地址,根据标志位的内容(1 或 0)对一条 I/O 线(本例中是 P1 端口的最低位)进行置位或清除。

PSW 中的进位标志 C 用作布尔处理机中的位累加器。凡是涉及到进位标志 C 的位操作指令,汇编时都看作进位标志专用指令。进位标志也有一个直接地址,因为它在 PSW 寄存器

中,而 PSW 是可位寻址的。

布尔变量操作指令中有逻辑与(ANL)和逻辑或(ORL),但没有异或(XRL)操作。用软件来实现异或操作是比较简单的。下面举个例子,用软件来实现两位的异或操作:C=bit1 XPL bit2。

```
MOV  C,bit 1
JNB  bit 2,  OVER
CPL  C
OVER(继续)
```

首先将 bit 1 送 C。如果 bit2=0,则 C 中内容就为正确的结果,即当 bit2=0 时,bit1 XRL bit2=bit1。如果 bit2=1,则只要对 C 取反(CPL C)就可得到正确的结果。

上面这段程序中用到的 JNB 指令,是一条位测试指令。JC,JB,JBC 执行被测试位置位转移;JNC,JNB 执行被测试位清 0 转移。在上面这个例子中,被测试位是 bit2。如果 bit2=0,则跳过 CPL C 这条指令。

JBC 是一条置位转移并复位指令,用这条指令不仅可对一个标志进行测试,同时当满足条件(被测试位为 1)转移时,还会自动将其复位。

PSW 中的每一位都是可直接寻址的,所以除了 C 以外,像奇偶标志(P)和其他一般用途的标志等都可用位测试指令来检测。

这些转移指令中的目标地址,在用汇编语言编写程序时通常是用一个标号或实际地址来表示的。当对程序进行汇编时,就会计算出一个带符号(2 的补码)的偏移字节(即为转移指令后一条指令到目标地址的距离),其范围为-128~+127。负数表示向后转移,正数表示向前转移。如果执行该转移指令,则将计算出的偏移字节数加到程序计数器 PC 中去。

4.2.5 控制程序转移类指令

控制程序转移指令中有无条件转移指令和条件转移指令。

无条件(绝对)转移指令如表 4-8 所列。

表 4-8 无条件转移指令

助记符	功 能	执行时间/ μ s
JMP addr	绝对转移	2
JMP @A+DPTR	间接转移	2
CALL addr	子程序调用	2
RET	子程序返回	2
RETI	中断返回	2
NOP	空操作	1

在表 4-8 中只列出一条 JMP addr 指令,实际上按目标地址安排不同,可分为 3 条指令: SJMP, LJMP 和 AJMP。JMP 是不考虑转移地址时所用的一般助记符。

SJMP 是短转移指令,指令中的目标地址是一个相对偏移量(见前面的说明)。这是一条双字节指令,由操作码和相对偏移量字节组成。相对 SJMP 的下一条指令其转移的范围为-128~+127 个字节。

LJMP 是长转移指令。指令中的目标地址是一个 16 位数,即这条指令允许转移的目标地址

在 64 KB 程序存储器空间的任意单元。这是一条 3 字节指令,由操作码和 2 个地址字节组成。

AJMP 是一条绝对转移指令,指令中的目标地址是一个 11 位数。这是双字节指令,第 1 个字节是操作码和目标地址的高 3 位,第 2 个字节是目标地址的低 8 位。执行这条指令时,将 11 位地址送到 PC 中的低 11 位,而 PC 中的高 5 位内容保持不变。因此目标地址必须与它下面的指令存放在同一个 2 KB 存储区内。

在用汇编语言编写程序时,编程人员也同样是用一个标号或 16 位数来表示这些转移指令中的目标地址的,汇编时会自动将它转换成正确的格式(相对偏移字节数)。如果指令中所要求的偏移字节数超出跳转范围时,就会将一段“目标地址超出范围”的信息写入列表文件中。

JMP @A+DPTR 是间接长转移指令,可用它来实现程序的散转。执行该指令时,目标地址由数据指针(DPTR)和累加器 A 的内容相加形成。通常 DPTR 中存放转移表的首地址,累加器 A 中存放转移表的变址。下面我们来看一个实现 5 种分支转移的程序:

```

MOV  DPTR,#JUMP_TABLE
MOV  A, INDEX_NUMBER
RL   A
JMP  @A+DPTR

JUMP_TABLE: AJMP CASE-0
             AJMP CASE-1
             AJMP CASE-2
             AJMP CASE-3
             AJMP CASE-4

```

将转移表的首址送到 DPTR 中,把待处理命令编号(0~4)存放在累加器 A 中。由于 AJMP 是双字节指令,所以 A 中必须是偶数,指令 RL A 是将 A 中的内容乘以 2,即将编号(0~4)变为偶数。当 A 为 0 时,散转到 CASE-0;A 为 2 时,散转到 CASE-1...

同样,在表 4-8 中有一条 CALL addr 指令,实际上根据子程序入口地址位置的不同,也可分为两条:ACALL 和 LCALL。CALL 是不考虑子程序入口地址位置时所用的助记符。

ACALL 是短调用指令,采用 11 位地址。所调用的子程序的入口地址必须与 ACALL 后面指令的第一个字节在同一个 2 KB 的程序存储器区中。

LCALL 是长调用指令,采用 16 位地址。该指令可以调用 64 KB 范围内程序存储器中的任何一个子程序。

用汇编语言编写程序时,同样是用一个标号或 16 位数来表示子程序的入口地址,汇编时会自动将它转换成正确的格式。

RET 是子程序返回指令。它使 CPU 从子程序返回到执行 CALL 的下一条指令。RET 指令用在由 ACALL 或 LCALL 调用的子程序末尾。

RETI 是中断返回指令,用来从一个中断服务子程序中返回。RET 和 RETI 的区别在于 RETI 指令除了执行 RET 指令的功能以外,还清除内部相应的中断状态寄存器(该触发器由 CPU 响应中断时置位,指示 CPU 当前是否在处理高级或低级中断)。因此中断服务程序必须以 RETI 为结束指令。CPU 执行 RETI 指令后至少再执行一条指令,才能响应新的中断请求。

条件转移指令如表 4-9 所列。

所有条件转移指令的目标地址都是用相对偏移量的格式表示的。因此转移范围相对于该

指令的下一条指令为-128~+127。然后,与其他转移指令一样,用户在用汇编语言编写程序时,只用一个标号或 16 位数来表示转移目标地址。

表 4-9 条件转移指令

助记符	功 能	寻址方式				执行时间/ μ s
		Dir	Ind	Reg	Imm	
JZ rel	A=0 转移	累加器 A				2
JNZ rel	A≠0 转移	累加器 A				2
DJNZ <byte>, rel	减 1 不为零转移	×		×		2
CJNE A,<byte>, rel	A≠<byte>转移	×			×	2
CJNE <byte>, #data, rel	<byte>≠#data 转移		×	×		2

JZ 和 JNZ 指令用来检测累加器中的内容是 0 或非 0。
DJNZ(减 1 不为 0 转移)指令用于程序的循环控制。如果一个程序要循环执行 N 次,可将 N 送到一个计数单元,在循环程序的出口用一条 DJNZ 指令来判断和控制循环的继续或终止。下面看看 N=10 的情况:

```
MOV      COUNTER, #10
LOOP: (开始
      *
      * }      循环体
      *
      * )
      结束)
      DJNZ      COUNTER,LOOP
      (继续)
```

CJNE(比较不相等转移)指令也可用来作循环控制,参见图 4-2 所给出的例子。该指令的功能是比较两个操作数的大小。如果它们的值不相等则转移。在图 4-2 这个例子中,两个操作数分别是 R1 中的内容和立即数 2AH。R1 中的初始值为 2EH,每执行一次循环程序,R1 的内容减 1,直到 R1 的内容为 2AH 时终止循环。
这条指令还可用作“大于”或“小于”比较。对两个无符号操作进行比较,根据比较的结果设置进位标志 C,并确定是否按第 3 个操作数转移。如果第 1 操作数小于第 2 操作数,则置位进位标志 C;如果第 1 操作数大于或等于第 2 操作数,则清 0 进位标志 C,不影响任何一个操作数的内容。

4.3 指令系统表

4.3.1 指令系统总表

指令系统总表如表 4-10 所列。

表 4-10 指令总表

	0	1	2	3	4	5	6	7
0	NOP	JBC bit, rel [3B, 2C]	JB bit, rel [3B, 2C]	JNB bit, rel [3B, 2C]	JC rel [2B, 2C]	JNC rel [2B, 2C]	JZ rel [2B, 2C]	JNZ rel [2B, 2C]
1	AJMP (P0) [2B, 2C]	ACALL (P0) [2B, 2C]	AJMP (P1) [2B, 2C]	ACALL (P1) [2B, 2C]	AJMP (P2) [2B, 2C]	ACALL (P2) [2B, 2C]	AJMP (P3) [2B, 2C]	ACALL (P3) [2B, 2C]
2	LJMP addr16 [3B, 2C]	LCALL addr16 [3B, 2C]	RET [2C]	RETI [2C]	ORL dir, A [2B]	ANL dir, A [2B]	XRL dir, a [2B]	ORL C, bit [2B, 2C]
3	RR A	RRC A	RL A	RLC A	ORL dir, #data [3B, 2C]	ANL dir, #data [3B, 2C]	XRL dir, #data [3B, 2C]	JMP @A+DPTR [2C]
4	INC A	DEC A	ADD A, #data [2B]	ADDC A, #data [2B]	ORL A, #data [2B]	ANL A, #data [2B]	XRL A, #data [2B]	MOV A, #data [2B]
5	INC dir [2B]	DEC dir [2B]	ADD A, dir [2B]	ADDC A, dir [2B]	ORL A, dir [2B]	ANL A, dir [2B]	XRL A, dir [2B]	MOV dir, #data [3B, 2C]
6	INC @R0	DEC @R0	ADD A, @R0	ADDC A, @R0	ORL A, @R0	ANL A, @R0	XRL A, @R0	MOV @R0, @data [2B]
7	INC @R1	DEC @R1	ADD A, @R1	ADDC A, @R1	ORL A, @R1	ANL A, @R1	XRL A, @R1	MOV @R1, #data [2B]
8	INC R0	DEC R0	ADD A, R0	ADDC A, R0	ORL A, R0	ANL A, R0	XRL A, R0	MOV R0, #data [2B]
9	INC R1	DEC R1	ADD A, R1	ADDC A, R1	ORL A, R1	ANL A, R1	XRL A, R1	MOV R1, #data [2B]
A	INC R2	DEC R2	ADD A, R2	ADDC A, R2	ORL A, R2	ANL A, R2	XRL A, R2	MOV R2, #data [2B]
B	INC R3	DEC R3	ADD A, R3	ADDC A, R3	ORL A, R3	ANL A, R3	XRL A, R3	MOV R3, #data [2B]
C	INC R4	DEC R4	ADD A, R4	ADDC A, R4	ORL A, R4	ANL A, R4	XRL A, R4	MOV R4, #data [2B]
D	INC R5	DEC R5	ADD A, R5	ADDC A, R5	ORL A, R5	ANL A, R5	XRL A, R5	MOV R5, #data [2B]
E	INC R6	DEC R6	ADD A, R6	ADDC A, R6	ORL A, R6	ANL A, R6	XRL A, R6	MOV R6, #data [2B]
F	INC R7	DEC R7	ADD A, R7	ADDC A, R7	ORL A, R7	ANL A, R7	XRL A, R7	MOV R7, #data [2B]

续表 4-10

	8	9	A	B	C	D	E	F
0	SJMP REL [2B, 2C]	MOV DPTR, # data 16 [3B, 2C]	ORL C, /bit [2B, 2C]	ANL C, /bit [2B, 2C]	PUSH dir [2B, 2C]	POP dir [2B, 2C]	MOVX A, @DPTR [2C]	MOVX @DPTR, A [2C]
1	AJMP (P4) [2B, 2C]	ACALL (P4) [2B, 2C]	AJMP (P5) [2B, 2C]	ACALL (P5) [2B, 2C]	AJMP (P6) [2B, 2C]	ACALL (P6) [2B, 2C]	AJMP (P7) [2B, 2C]	ACALL (P7) [2B, 2C]
2	ANL C, bit [2B, 2C]	MOV bit, C [2B, 2C]	MOV C, bit [2B]	CPL bit [2B]	CLR bit [2B]	SETB bit [2B]	MOVX A, @R0 [2C]	MOVX @R0, A [2C]
3	MOVC A, @A+PC [2C]	MOVC A, @A+DPTR [2C]	INC DPTR [2C]	CPL C	CLR C	SETB C	MOVX A, @RI [2C]	MOVX @RI, A [2C]
4	DIV AB [2B, 4C]	SUBB A, #data [2B]	MUL AB [4C]	CJNEA, #data, rel [3B, 2C]	SWAP A	DA A	CLR A	CPL A
5	MOV dir, dir [3B, 2C]	SUBB A, dir [2B]		CJNE A, dir, rel [3B, 2C]	XCH A, dir [2B]	DJNZ dir, rel [3B, 2C]	MOV A, dir [2B]	MOV dir, A [2B]
6	MOV dir, @R0 [2B, 2C]	SUBB A, @R0	MOV @R0, dir [2B, 2C]	CJNE @R0, #data, rel [3B, 2C]	XCH A, @R0	XCHD A, @R0	MOV A, @R0	MOV @R0, A
7	MOV dir, @R1 [2B, 2C]	SUBB A, @R1	MOV @R1, dir [2B, 2C]	CJNE @R1, #data, rel [3B, 2C]	XCH A, @R1	XCHD A, @R1	MOV A, @R1	MOV @R1, A
8	MOV dir, R0 [2B, 2C]	SUBB A, R0	MOV R0, dir [2B, 2C]	CJNE R0, #data, rel [3B, 2C]	XCH A, R0	DJNZ R0, rel [2B, 2C]	MOV A, R0	MOV R0, A
9	MOV dir, R1 [2B, 2C]	SUBB A, R1	MOV R1, dir [2B, 2C]	CJNE R1, #data, rel [3B, 2C]	XCH A, R1	DJNZ R1, rel [2B, 2C]	MOV A, R1	MOV R1, A
A	MOV dir, R2 [2B, 2C]	SUBB A, R2	MOV R2, dir [2B, 2C]	CJNE R2, #data, rel [3B, 2C]	XCH A, R2	DJNZ R2, rel [2B, 2C]	MOV A, R2	MOV R2, A
B	MOV dir, R3 [2B, 2C]	SUBB A, R3	MOV R3, dir [2B, 2C]	CJNE R3, #data, rel [3B, 2C]	XCH A, R3	DJNZ R3, rel [2B, 2C]	MOV A, R3	MOV R3, A
C	MOV dir, R4 [2B, 2C]	SUBB A, R4	MOV R4, dir [2B, 2C]	CJNE R4, #data, rel [3B, 2C]	XCH A, R4	DJNZ R4, rel [2B, 2C]	MOV A, R4	MOV R4, A
D	MOV dir, R5 [2B, 2C]	SUBB A, R5	MOV R5, dir [2B, 2C]	CJNE R5, #data, rel [3B, 2C]	XCH A, R5	DJNZ R5, rel [2B, 2C]	MOV A, R5	MOV R5, A
E	MOV dir, R6 [2B, 2C]	SUBB A, R6	MOV R6, dir [2B, 2C]	CJNE R6, #data, rel [3B, 2C]	XCH A, R6	DJNZ R6, rel [2B, 2C]	MOV A, R6	MOV R6, A
F	MOV dir, R7 [2B, 2C]	SUBB A, R7	MOV R7, dir [2B, 2C]	CJNE R7, #data, rel [3B, 2C]	XCH A, R7	DJNZ R7, rel [2B, 2C]	MOV A, R7	MOF R7, A

在表 4-10 中,有关括号内的数字含义如下:

[2B]:2 字节;[3B]:3 字节;[2C]:2 周期;[4C]:4 周期。无括号及数字标志的指令都是 1

字节、1 周期指令。

4.3.2 按操作码排序的指令总表

表 4-11 给出的是按操作码大小,从 00H~FFH 进行排序的指令总表。

表 4-11 按操作码排序的指令总表

十六进制码	字节数	助记符	操作对象	十六进制码	字节数	助记符	操作对象
00	1	NOP		22	1	RET	
01	2	AJMP	code addr	23	1	RL	A
02	3	LJMP	code addr	24	2	ADD	A, # data
03	1	RR	A	25	2	ADD	A, data addr
04	1	INC	A	26	1	ADD	A, @R0
05	2	INC	data addr	27	1	ADD	A, @R1
06	1	INC	@R0	28	1	ADD	A, R0
07	1	INC	@R1	29	1	ADD	A, R1
08	1	INC	R0	2A	1	ADD	A, R2
09	1	INC	R1	2B	1	ADD	A, R3
0A	1	INC	R2	2C	1	ADD	A, R4
0B	1	INC	R3	2D	1	ADD	A, R5
0C	1	INC	R4	2E	1	ADD	A, R6
0D	1	INC	R5	2F	1	ADD	A, R7
0E	1	INC	R6	30	3	JNB	bit addr, code addr
0F	1	INC	R7	31	2	ACALL	code addr
10	3	JBC	bit addr, code addr	32	1	RETI	
11	2	ACALL	code addr	33	1	RLC	A
12	3	LCALL	code addr	34	2	ADDC	A, # data
13	1	RRC	A	35	2	ADDC	A, data addr
14	1	DEC	A	36	1	ADDC	A, @R0
15	2	DEC	data addr	37	1	ADDC	A, @R1
16	1	DEC	@R0	38	1	ADDC	A, R0
17	1	DEC	@R1	39	1	ADDC	A, R1
18	1	DEC	R0	3A	1	ADDC	A, R2
19	1	DEC	R1	3B	1	ADDC	A, R3
1A	1	DEC	R2	3C	1	ADDC	A, R4
1B	1	DEC	R3	3D	1	ADDC	A, R5
1C	1	DEC	R4	3E	1	ADDC	A, R6
1D	1	DEC	R5	3F	1	ADDC	A, R7
1E	1	DEC	R6	40	2	JC	code addr
1F	1	DEC	R7	41	2	AJMP	code addr
20	3	JB	bit addr, code addr	42	2	ORL	data addr, A
21	2	AJMP	code addr	43	3	ORL	data addr, # data

续表 4-11

十六进制码	字节数	助记符	操作对象	十六进制码	字节数	助记符	操作对象
44	2	ORL	A, # data	6B	1	XRL	A, R3
45	2	ORL	A, data addr	6C	1	XRL	A, R4
46	1	ORL	A, @R0	6D	1	XRL	A, R5
47	1	ORL	A, @R1	6E	1	XRL	A, R6
48	1	ORL	A, R0	6F	1	XRL	A, R7
49	1	ORL	A, R1	70	2	JNZ	code addr
4A	1	ORL	A, R2	71	2	ACALL	code addr
4B	1	ORL	A, R3	72	2	ORL	C bit addr
4C	1	ORL	A, R4	73	1	JMP	@A+DPTR
4D	1	ORL	A, R5	74	2	MOV	A, # data
4E	1	ORL	A, R6	75	3	MOV	data addr, # data
4F	1	ORL	A, R7	76	2	MOV	@R0, # data
50	2	JNC	code addr	77	2	MOV	@R1, # data
51	2	ACALL	code addr	78	2	MOV	R0, # data
52	2	ANL	data addr, A	79	2	MOV	R1, # data
53	3	ANL	data addr, # data	7A	2	MOV	R2, # data
54	2	ANL	A, # data	7B	2	MOV	R3, # data
55	2	ANL	A, data addr	7C	2	MOV	R4, # data
56	1	ANL	A, @R0	7D	2	MOV	R5, # data
57	1	ANL	A, @R1	7E	2	MOV	R6, # data
58	1	ANL	A, R0	7F	2	MOV	R7, # data
59	1	ANL	A, R1	80	2	SJMP	code addr
5A	1	ANL	A, R2	81	2	AJMP	code addr
5B	1	ANL	A, R3	82	2	ANL	C bit addr
5C	1	ANL	A, R4	83	1	MOVC	A, @A+PC
5D	1	ANL	A, R5	84	1	DIV	AB
5E	1	ANL	A, R6	85	3	MOV	data addr, data addr
5F	1	ANL	A, R7	86	2	MOV	data addr, @R0
60	2	JZ	code addr	87	2	MOV	data addr, @R1
61	2	AJMP	code addr	88	2	MOV	data addr, R0
62	2	XRL	data addr, A	89	2	MOV	data addr, R1
63	3	XRL	data addr, # data	8A	2	MOV	data addr, R2
64	2	XRL	A, # data	8B	2	MOV	data addr, R3
65	2	XRL	A, data addr	8C	2	MOV	data addr, R4
66	1	XRL	A, @R0	8D	2	MOV	data addr, R5
67	1	XRL	A, @R1	8E	2	MOV	data addr, R6
68	1	XRL	A, R0	8F	2	MOV	data addr, R7
69	1	XRL	A, R1	90	3	MOV	DPTR, # data
6A	1	XRL	A, R2	91	2	ACALL	code addr

续表 4-11

十六进制码	字节数	助记符	操作对象	十六进制码	字节数	助记符	操作对象
92	2	MOV	bit addr, C	B9	3	CNJE	R1, #data, code addr
93	1	MOVC	A, @A+DPTR	BA	3	CNJE	R2, #data, code addr
94	2	SUBB	A, #data	BB	3	CNJE	R3, #data, code addr
95	2	SUBB	A, data addr	BC	3	CNJE	R4, #data, code addr
96	1	SUBB	A, @R0	BD	3	CNJE	R5, #data, code addr
97	1	SUBB	A, @R1	BE	3	CNJE	R6, #data, code addr
98	1	SUBB	A, R0	BF	3	CNJE	R7, #data, code addr
99	1	SUBB	A, R1	C0	2	PUSH	data addr
9A	1	SUBB	A, R2	C1	2	AJMP	code addr
9B	1	SUBB	A, R3	C2	2	CLR	bit addr
9C	1	SUBB	A, R4	C3	1	CLR	C
9D	1	SUBB	A, R5	C4	1	SWAP	A
9E	1	SUBB	A, R6	C5	2	XCH	A, data addr
9F	1	SUBB	A, R7	C6	1	XCH	A, @R0
A0	2	ORL	C, bit addr	C7	1	XCH	A, @R1
A1	2	AJMP	code addr	C8	1	XCH	A, R0
A2	2	MOV	C bit addr	C9	1	XCH	A, R1
A3	1	INC	DPTR	CA	1	XCH	A, R2
A4	1	MUL	AB	CB	1	XCH	A, R3
A5		reserved		CC	1	XCH	A, R4
A6	2	MOV	@R0, data addr	CD	1	XCH	A, R5
A7	2	MOV	@R1, data addr	CE	1	XCH	A, R6
A8	2	MOV	R0, data addr	CF	1	XCH	A, R7
A9	2	MOV	R1, data addr	D0	2	POP	data addr
AA	2	MOV	R2, data addr	D1	2	ACALL	code addr
AB	2	MOV	R3, data addr	D2	2	SETB	bit addr
AC	2	MOV	R4, data addr	D3	1	SETB	C
AD	2	MOV	R5, data addr	D4	1	DA	A
AE	2	MOV	R6, data addr	D5	3	DJNZ	data addr, code addr
AF	2	MOV	R7, data addr	D6	1	XCHD	A, @R0
B0	2	ANL	C, bit addr	D7	1	XCHD	A, @R1
B1	2	ACALL	code addr	D8	2	DJNZ	R0, code addr
B2	2	CPL	bit addr	D9	2	DJNZ	R1, code addr
B3	1	CPL	C	DA	2	DJNZ	R2, code addr
B4	3	CJNE	A, #data code addr	DB	2	DJNZ	R3, code addr
B5	3	CJNE	A, data addr, code addr	DC	2	DJNZ	R4, code addr
B6	3	CJNE	@R0, #data, code addr	DD	2	DJNZ	R5, code addr
B7	3	CJNE	@R1, #data, code addr	DE	2	DJNZ	R6, code addr
B8	3	CJNE	R0, #data, code addr	DF	2	DJNZ	R7, code addr

续表 4-11

十六进制码	字节数	助记符	操作对象	十六进制码	字节数	助记符	操作对象
E0	1	MOVX	A, @DPTR	F0	1	MOVX	@DPTR, A
E1	2	AJMP	code addr	F1	2	ACALL	code addr
E2	1	MOVX	A, @R0	F2	1	MOVX	@R0, A
E3	1	MOVX	A, @R1	F3	1	MOVX	@R1, A
E4	1	CLR	A	F4	1	CPL	A
E5	2	MOV	A, data addr	F5	2	MOV	data addr, A
E6	1	MOV	A, @R0	F6	1	MOV	@R0, A
E7	1	MOV	A, @R1	F7	1	MOV	@R1, A
E8	1	MOV	A, R0	F8	1	MOV	R0, A
E9	1	MOV	A, R1	F9	1	MOV	R1, A
EA	1	MOV	A, R2	FA	1	MOV	R2, A
EB	1	MOV	A, R3	FB	1	MOV	R3, A
EC	1	MOV	A, R4	FC	1	MOV	R4, A
ED	1	MOV	A, R5	FD	1	MOV	R5, A
EE	1	MOV	A, R6	FE	1	MOV	R6, A
EF	1	MOV	A, R7	FF	1	MOV	R7, A

第五章 输入输出接口技术

89 系列单片机在应用中需要和外部进行信息交换。这些信息交换方式有并行输入输出、串行输入输出,还有串-并、并-串行信息转换及 A/D,D/A 转换等技术;除此之外,还有显示驱动技术、功率器件驱动技术等。在这一章中,介绍 89 系列单片机的一些接口技术。这些都是关于 A/D、D/A、串-并行转换、LED 驱动、功率器件驱动的技术。至于串行输入输出和并行输入输出已在 P0~P3 端口和串行接口的功能中加以介绍了,在这一章不再重复。

5.1 串-并、并-串行转换技术

89 系列单片机在片内含有并行接口和串行接口。在一些特殊的应用场合,例如在一些 LCD 显示屏显示图形的场合,输出的数据量较大;在显示和采样分离并距离较远的地方,又要求数据传送线较少。这样,就要求采用并-串和串-并行转换接口,以满足实际应用的需要。现在很多 LCD 显示屏都已经含有驱动电路,要求输入的信号是 8 位数据,以便于对 LCD 的存储器地址进行寻址和传送数据。因此,采用一个串-并行转换接口,使 89 系列单片机可以通过串行口输出信息,并由串-并行接口电路将其变换到 LCD 的接口电路中的 8 位输入端,则可以使单片机的所有并行接口执行其他有效的工作,使单片机的效率更充分地发挥。

一种典型的串-并、并-串行接口方法如图 5-1 所示。

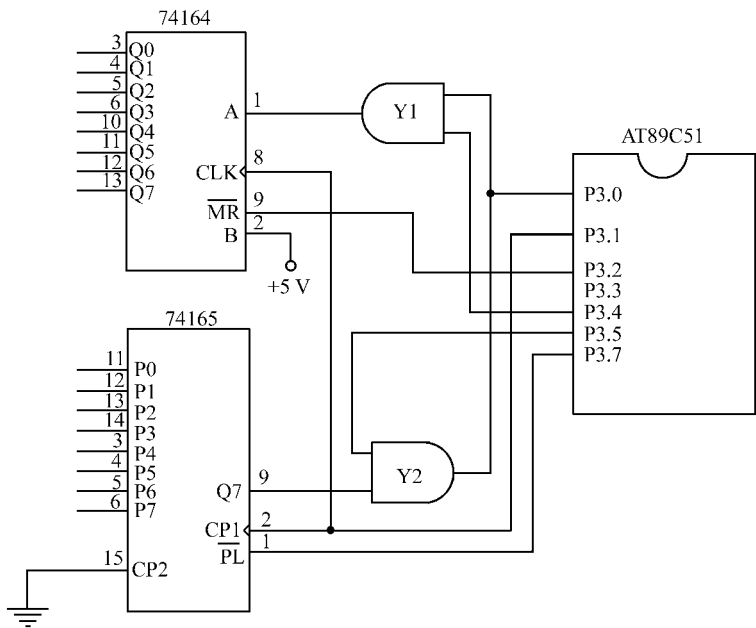


图 5-1 串-并、并-串行转换接口

在图 5-1 中所给出的串-并、并-串行转换接口中,单片机 AT89C51 工作时,其内部串行接口工作在方式 0。这时,它是通过 P3.0 端(RXD 端)执行数据的串行发送或接收的,而 P3.1 端(TXD)端则用于输出移位脉冲。在发送或接收时,都是最低有效位 LSB 居先,最高有效位 MSB 居后。传送的频率为振荡频率的 1/12。这种方式 0 的串行工作方式也称移位寄存器方式,或同步方式。

1. 串-并行转换接口原理

串-并行转换是指从单片机 AT89C51 串行输出,通过转换之后成为并行信号输出的过程。这时,AT89C51 的 P3.0 端即 RXD 执行的是数据发送,所以 8 位串行数据的低位先送出,最后送出数据的最高位。由于在 P3.1 端即 TXD 端发出的是同步脉冲,所以可以用它作为外部串-并行寄存器的数据时钟。

在图 5-1 中,74164 和与门 Y1 组成了串-并行转换接口。74164 的 $\overline{MR}$ 端是清 0 端,当 $\overline{MR}=0$ 时,对 74164 清 0;当 $\overline{MR}=1$ 时,则可对 74164 送入串行 8 位数据。

当 AT89C51 工作在串行方式 0 时,首先在 P3.2、P3.4 输出高电平,令 $\overline{MR}=1$,则 74164 处于接收串行数据状态。接着从 P3.0 端送出串行数据,其顺序为先低位,后高位;同时,P3.1 端发出同步脉冲到 74164 的时钟输出端 CLK。在输出 8 个脉冲之后,在串行口中的 8 位数据就会输出并寄存在 74164 中。这时,串-并行转换过程完成。

2. 并-串行转换接口原理

并-串行转换是指从外部并行输入的 8 位数据,通过转换之后成为串行信号,输入到 AT89C51 的串行口中去的过程。这时,AT89C51 的 P3.0 端口即 RXD 执行的是数据接收,接收时是先接收串行数据的最高有效位 MSB,最后接收最低有效位 LSB。同时,在 P3.1 端发出的是同步脉冲,它用作 74165 的串行移位时钟。

在图 5-1 中,74165 和与门 Y2 组成了并-串行转换接口。74165 的 $\overline{PL}$ 是并行装入控制信号。当 $\overline{PL}=0$ 时,74165 装入外部输入的 8 位数据;当 $\overline{PL}=1$ 时,只要从 CP1 端有正脉冲来,则 74165 执行右移串行输出,从而从高位开始执行串行 8 位数据传送。在 CP1 端每来一个正脉冲,则 74165 右移一位。当 CP1 端送出了 8 个正脉冲之后,则 74165 中的 8 位数据串行输出完毕。

当 AT89C51 工作于串行方式 0 时,首先在 P3.5 输出高电平,令 Y2 门打开,允许来自 74165 的 Q7 端串行信号输入。同时,P3.7 应输出高电平,使 74165 处于串行输出状态。当 P3.1 每产生一个脉冲输出时,则送到 74165 的 CP1 时钟输入端,使 74165 右移一位。在 P3.1 输出了 8 个正脉冲之后,74165 的 8 位数据全部串行移出并送到 AT89C51 的串行口内。这时,并-串行转换结束。

3. 串-并、并-串转换程序

串-并、并-串行转换程序如下所示:

```
MOV      SCON, #00H           ;方式 0
CLR      TI                    ;清发送中断
...
SEND:    SETB    P3.4           ;允许串行输出
          SETB    P3.2           ;允许 74164 接收
          MOV     SBUF, A         ;准备发送 A
WAITE:   JNB     TI, WAITE       ;等待发送结束
```

	CLR	TI	
	CLR	P3.4	;停止串行输出
	...		
RECE1:	SETB	P3.5	;允许串行输入
	SETB	P3.7	
	CLI	P3.7	;74165 寄存数据
	SETB	P3.7	
	CLR	RI	;清接收中断
	SETB	REN	;允许接收
WAITR:	JNB	RI, WAITR	;等待接收完毕
	CLR	REN	;停止接收
	CLR	RI	
	MOV	A, SBUF	;送结果到 A
	CLR	P3.5	;停止输入

5.2 A/D 转换技术

在 89 系列单片机中,其内部不含 A/D 转换器,而在一些应用场合中,需要对模拟信号进行检测。这样,就要求利用 89 系列单片机的一些特点,结合外部器件组成可以实现 A/D 转换功能的结构,以适应有关用途。

5.2.1 外接 RC 实现 A/D 转换

AT89C2051 内含模拟比较器电路,P1.0,P1.1 分别作为比较器的同相输入端和反相输入端,P3.6 为比较器的输出端,无外引脚。当用 P1.0,P1.1 作输入输出时应上拉电阻,因为内部无上拉电阻;当用该比较器作电源电压检测时,不要把电源直接接到口线上,这样容易烧坏该引脚,可以经过分压后再输入。

利用这个比较器可制作简单的、低精度的 A/D 转换电路,原理如图 5-2 所示。在 P3.7 的输出端接有一个 RC 回路, $R=267\text{ k}\Omega$, $C=2\text{ nF}$,RC 回路的 RC 相接点直接和 P1.0 相连,即电容 C 的电压 V_C 将会输入到 P1.0 端,这个端也就是 AT89C2051 的内部比较器正输入端,而被转换的外部模拟电压 V_X 则从 P1.1 输入,也就是从内部比较器的负输入端输入。

在图 5-2 中所示的电路中,P3.7 输出端还接一个 $5.1\text{ k}\Omega$ 电阻到正电源 V_{CC} 。这个电阻是 P3.7 的外部上拉电阻,用于保证 P3.7 在输出为逻辑“1”时有足够的高电平,也就是接近于 V_{CC} 的高电平。

实现 A/D 的原理如下。

一开始时,P3.7 输出低电平“0”,电容 C 的电压 $V_C=0$,有被测信号 V_X 从 P1.1 输入。由于存在 $V_X>0$,故 AT89C2051 内部的比较器两个输入端有 $P1.0=0$, $P1.1>0$,则比较器输出信号“0”到 P3.6。要注意 P3.6 只在 AT89C2051 的内部,它并不引出到外面。接着,P3.7 输出高电平“1”,则通过 R 对电容 C 充电,电容 C 的电压 V_C 随着充电而升高。当到达 $V_C=V_X$ 时,则比较器输出变为高电平“1”。只要测出开始对电容 C 充电时到比较器输出高电平“1”时的时间 t,这个时间 t 就可以转换成外部被测电压 V_X 的电压值,从而完成电压的 A/D 转换过程。

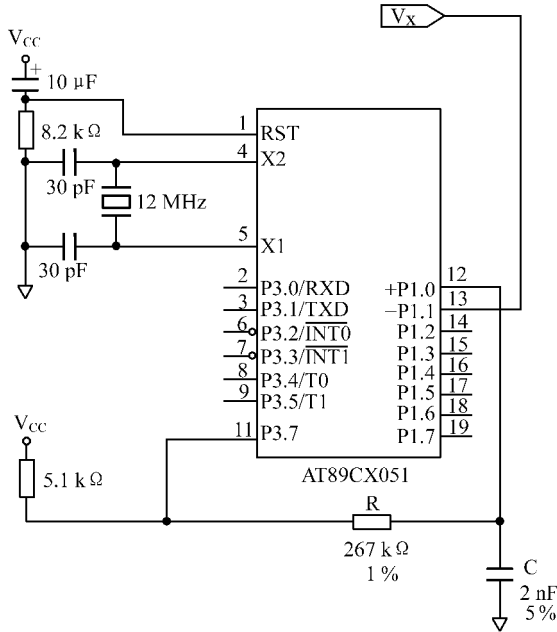


图 5-2 A/D 转换电路原理

图 5-3 给出了典型的电容电荷的充放电周期波形。在 $V_c = V_{cc}/2$ 的电压线附近, 曲线的充电时间、放电时间等同。下面的等式和讨论, 除特别说明外都是应用于充电周期的。

由指数等式给出了电容上的电压的时间函数

$$V_c = V_{cc}(1 - e^{-t/RC}) \tag{5-1}$$

这里 V_c 是电容在时间 t 时的电压, V_{cc} 是电源电压, RC 是电阻和电容值的乘积。注意电压的单位为 V , 时间的单位为 s , 电阻的单位为 Ω 以及电容的单位为 F 。乘积 RC 叫作网络的时间常数, 它的值将影响波形的形状。当开始充电或放电时波形最陡峭, 然后随时间增加而变平坦。

从式(5-1)可知: 电容 C 上充电电压 V_c 是用指数表示的, 所以用 RC 回路实现 A/D 存在两个问题。

第一个问题是在单片机中如何利用程序求解式(5-1)所给出的指数方程。

为了解决这个问题, 可以使用查表的方法, 这样微控制器就无需实时计算指数方程了。以单位内部时钟为间隔进行抽样, 预先将指数方程式对应的时间和电压进行编程和格式化并制成一个表, 用于软件之中。

第二个问题是 RC 模数转换方法中 RC 变量的值可能会引起很大的误差。图 5-4 给出了电容的充电曲线在电阻和电容的乘积改变时的变化情况。

电容充放电周期的对称性可以减少 RC 变量对精确度的影响。可以利用充电周期测出的电压比 $V_{cc}/2$ 小和放电周期测量电压比 $V_{cc}/2$ 大的特性, 在 $V_{cc}/2$ 将正误差与负误差抵消掉。

在元件值被指定前, 必须先决定比较器输出的抽样时间的间隔。抽样间隔应该尽可能地短, 因为抽样间隔越短则转换器精确度就越高, 转换时间就越短。抽样间隔受执行必要代码的

时间所限制,这个时间由微控制器的时钟频率决定。在一些应用中,微控制器工作在 12 MHz 的时钟下,取样间隔 $5\ \mu\text{s}$ 。

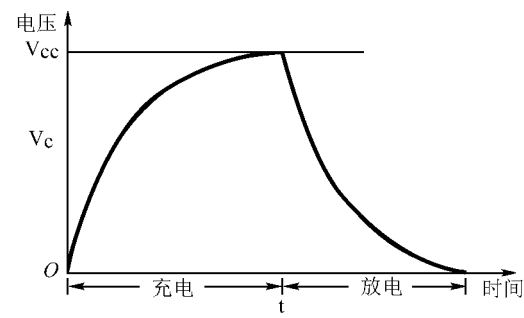


图 5-3 RC 回路的充电放电时间

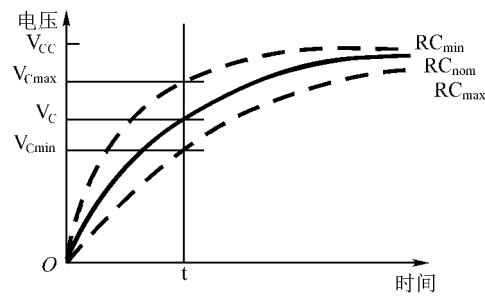


图 5-4 RC 参数变化和电容电压 V_c 的关系

时间常数(RC)会影响电容充放电曲线的形状。必须选择适当的常数值,以便波形最陡峭的部分在期望的分辨率下求解。充电周期最陡峭的部分在充电开始部分,放电周期最陡峭的部分则在放电开始部分。由于波形的对称,可以用相同的时间常数测量充放电部分。图 5-5 给出了电容充电初期电压和抽样时间之间的关系。在图中, ΔV 是转换器期望的电压解; $\Delta\tau$ 是先前决定的抽样间隔时间。曲线上标明的 V_c 代表这个电容上的电压,曲线在这个尺度内成线性。在这个图中,曲线的斜率是理想斜率,以便抽样能在靠近电压间隔的中心处发生。第一个抽样点处于起始点与 $\Delta\tau/2$ 的对应电压之间。

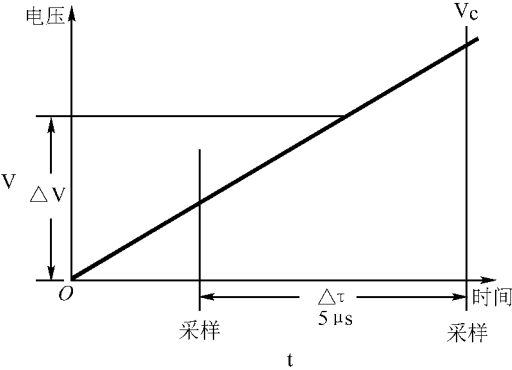


图 5-5 原点处抽样时间与电压的关系

为了获得时间常数的最小值,这个常数将由第一个抽样点的斜率产生,为得到 RC 的解,方程式可化为

$$RC = -t / \ln(1 - V_c / V_{cc}) \tag{5-2}$$

然后,设 ΔV 为期望的最小值(0.05 V),设抽样间隔为先前决定的值(5 ms),并计算第一个抽样点($V_c = 1/2\Delta V, t = 1/2\Delta\tau$)的 RC 值。

RC 的实际值应大于等于时间计算的最小常数值。要求电阻值误差小于 1%,电容值误差小于 5%。

$$[R_{nom} - (1 - 0.01)][C_{nom} - (1 - 0.05)] = 4.99 \times 10^{-4}$$

R 的取值是有约束的。再参考图 5-2,注意有一个 5.1 kΩ 的上拉电阻连接在微控制器的引脚 11 上。这个电阻对引脚 11 起上拉作用,但是改变了网络的时间常数,给充放电周期的充电部分带来致命影响,导致充放电周期波形不对称,给模数转换带来错误。为了最大限度地减少这种误差,要求 R 的值要远大于上拉电阻。在这个电压表应用中,R 值为 267 kΩ,已远大于 5.1 kΩ 的上拉电阻。

时间常数 RC 决定电容的充放电周期的长短。

将充/放电曲线渐近线的电压减去/或加上预定分辨率的一半代入式(5-3),即可求出充/放电周期的近似值。充电部分的渐近线为 V_{CC} 。由于波形的对称,所得的充电周期的时间即为放电周期的时间。为求周期的时间,式(5-1)可化为

$$t = -RC \times \ln(1 - V_C/V_{CC}) \quad (5-3)$$

设分辨率为 0.05 V,则期望的电容电压为

$$V_C = V_{CC} - (1/2)(0.05) = V_{CC} - 0.025$$

代入式(5-3)得 V_C 充电到渐近线 V_{CC} 时所用的时间

$$T_{\max} = -R_{\max} \times C_{\max} \times \ln[1 - (V_{CC} - 0.025)/V_{CC}] = 3 \text{ ms}$$

其中

$$R_{\max} = R_{\text{nom}}(1 + 0.01) = 1.01 \times 267 \text{ k}\Omega$$

$$C_{\max} = C_{\text{nom}}(1 + 0.05) = 1.05 \times 2 \text{ nF}$$

最小的采样数由电容电压达到 $V_{CC}/2$ 时的时间除以采样间隔得到。计算时,时间常数取最大值,因为电阻值和电容值大时,电容电压升高的速度慢些。

将 $V_C = V_{CC}/2$, RC 代入式(5-3),得

$$T_{\max} = 393 \text{ }\mu\text{s}$$

半个周期的最小采样数为 N

$$N = T_{\max}/\Delta\tau = (393 \times 10^{-6})/(5 \times 10^{-6}) = 79$$

其中: $\Delta\tau$ 为采样间隔, $\Delta\tau = 5 \text{ }\mu\text{s}$ 。

为了在单片机中避免执行解指数方程的运算,建立采样数和对应电压的表格,是一种有效而简单的方法。

实际上,是用一个时钟对充电或放电时间进行计数。这个计数值即是上面所述的采样数,而电容的电压 V_C 是和外部被转换的电压 V_X 相等的。因此,可以预先计算采样数 N 以及对应的 V_C ,建立 N 和 V_C 的对应表。

为了充分提高精确度,测量 $0 \sim V_{CC}/2$ 的电压时用充电周期,测量 $V_{CC}/2 \sim V_{CC}$ 的电压时用放电周期。结果,表的入口数将是前面计算的采样数字的 2 倍。

对应表中包含有采样次数和对应的计算电压。对每个半周期,表的第 N 个入口对应的电压为 $T = (N-1)\Delta\tau$ 时的电容电压, $\Delta\tau$ 是采样间隔。

在充电的半周期,对应电压的计算用式(5-1)

$$V_C = V_{CC}(1 - e^{-t/RC})$$

在放电的半周期,则应用下式

$$V_C = V_{CC} \times e^{-t/RC}$$

对应表的大小,取决于采样的次数。由于采样次数由采样间隔和转换分辨率决定,所以要求的分辨率越高则表就会越大。在应用中,如果要求的分辨率为 0.05 V,则最后得出的对应表要 158 个入口。

在充电的半周期中,设 $N = (0 \sim 78)$,则 $t = N\Delta\tau$

设采样间隔 $\Delta\tau = 5 \text{ }\mu\text{s}$, $R = 267 \text{ k}\Omega$, $C = 2 \text{ nF}$, $V = 5 \text{ V}$,并代入式(5-1),得

$$V = 5 \times (1 - e^{-0.0093633N}) \quad (5-4)$$

同理可得,在放电的半周期中

$$V = 5 \times e^{-0.0093633N} \quad (5-5)$$

最后可以得到充电半周期中所有的采样点的对应电压值：

N=0	V=0.000
N=1	V=0.047
⋮	⋮
N=74	V=2.499
⋮	⋮
N=78	V=2.591

以及放电半周期中采样点和对应电压值：

N=0	V=5.000
N=1	V=4.953
⋮	⋮
N=74	V=2.501
N=75	V=2.477
⋮	⋮
N=78	V=2.409

如上所示,两个半周期的采样数都大于达到 2.5 V 所需的采样数。注意 N=0 和 N=1 的计算电压之差小于分辨率 0.05 V,而且相应的计算电压之差随 N 的增加而减少,所以可以达到要求的分辨率。在电路中时间和电压的映射关系是非线性的。

上面的对应表中的电压值并不是最后用于软件查表用的电压值。在应用中,这些计算电压(四舍五入,保留一位小数)以 BCD 格式存储于表中。例如:2.523 V 在表中表示为 25H,用于显示 2.5 V。

考虑要求的精确度为±0.1 V,然而 RC 的精确度有时小于±0.1 V。因为即使使用精确的元件,元件值的变量引起的误差也可能要高达±0.104 V,这一点将在下面详细说明。

为了计算 $V_C=2.5\text{ V}$ 的最大误差情况,首先将标准的 R,C 值代入式(5-3),得

$$t=-R_{\text{nom}}C_{\text{nom}}\times\ln(1-V_C/V_{\text{CC}})=-R_{\text{nom}}C_{\text{nom}}\times\ln(0.5)$$

将 t 代入方程式(5-1),可求得

$$V_{\text{Cmin}}=V_{\text{CC}}[1-e^{-t/(R_{\text{max(max)}}C)}]=2.399\text{ V}$$

同理,可求得

$$V_{\text{Cmax}}=V_{\text{CC}}[1-e^{-t/(R_{\text{min(min)}}C)}]=2.607\text{ V}$$

结果可得在 2.5 V 处的最大的误差为±(2.607-2.399)/2,即±0.104 V。元件值越精确,误差越小,同时电容的特性也会影响转换精度和线性。

5.2.2 一次积分 A/D 转换

外接 RC 进行 A/D 转换,实质上是双积分型 A/D 原理。采用恰当的方法,可以使一次积分取代双积分进行 A/D 转换。由于一次积分运算简单,所用时间较短,所以,这是一种较好的方法。在这一节中,首先介绍双积分的原理,然后介绍从双积分变成一次积分并实现一次积分 A/D 的过程。

1. 双积分 A/D 的基本原理

双积分 A/D 原理可以用图 5-6 所示的电路来说明。

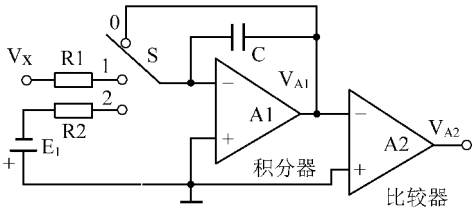


图 5-6 双积分 A/D 原理

从图中看出电路由运算放大器 A1、A2、电容 C、电阻 R1、R2 以及电源 E₁ 组成。其中,A1 在电路中用作积分器,A2 用作比较器。它们的工作过程如下:

(1) 电容 C 充电

开关 K 处于位置 0 时,电容 C 被短接,积分器输出 V_{A10} = 0 V,比较器输出 V_{A2} = “0”;开关 K 处于位置 1 时,待测量电压 V_x

经电阻 R1 对电容 C 正向充电,V_{A1} 下降,V_{A2} = “1”,经时间 T₁,V_{A1} 降至 -E₀,电容 C 积分的电荷为 Q₁

$$Q_1 = E_0 C = \int_0^{T_1} \frac{V_x}{R_1} dt = \frac{V_x}{R_1} T_1 \tag{5-6}$$

(2) 电容 C 放电

开关 K 处于位置 2,E₁ 经 R2 对电容 C 反向充电,经时间 T₂,V_{A1} 复原至 0 V,V_{A2} = “0”,电容 C 释放的电荷为 Q₂

$$Q_2 = E_0 C = \int_0^{T_2} \frac{E_1}{R_2} dt = \frac{E_1}{R_2} T_2 \tag{5-7}$$

电容 C 从 V_{A1} = 0 V,充电到 -E₀;然后又放电恢复到使 V_{A1} = 0 V。显然,其充电电荷 Q₁ 和放电电荷 Q₂ 相等。即有

$$Q_1 = Q_2$$

从而有

$$\frac{V_x}{R_1} T_1 = \frac{E_1}{R_2} T_2$$

处理之后有 V_x

$$V_x = \frac{R_1 T_2}{R_2 T_1} E_1 \tag{5-8}$$

由于在图 5-6 中,E₁,R1,R2 是已知的恒定值,而 T₁,T₂ 都可以测量出来,故 V_x 是可求的,这是一般双积分 A/D 的基本思想。在实际中,可以把比较器 A2 的输出 V_{A2} 接到单片机的有关输入端。首先测量开关 K 在接通位置 1 时到转向接通位置 2 时所经历的时间,这段时间实际是位置 1 停留的时间,也就是电容 C 的充电时间 T₁。接着测量开关 K 在接通位置 2 时到 V_{A1} 恢复为 0 V 的时间,这段时间实质是电容放电的时间 T₂。

从双积分 A/D 原理的工作过程可知:在 A/D 过程中之所以要考虑电容充电时的积分,也要考虑电容放电时的积分,是因为在充电时的积分中,所充电的电荷 Q₁ 是未知的,即电容的电压 V_C = -E₀ 是未知的,故式(5-6)的方程中有

$$Q_1 = E_0 C \tag{5-9}$$

或

$$Q_1 = V_x T_1 / R_1 \tag{5-10}$$

也就是说:在式(5-9)中存在 Q₁ 和 E₀ 两个未知量;式(5-10)中存在 Q₁ 和 V_x 两个未知量;所以方程不可解,或有无数解,即所求的输入电压 V_x 无法解出。

为了求出 V_x,所以要增加电容 C 放电时的积分过程。把充电和放电时的积分过程式

(5-6)、式(5-7)结合,就可以求出式(5-8),根据式(5-8)则被测电压 V_X 可以求出。

双积分 A/D 存在一个明显的缺点:需要充放电两个积分过程,故时间较长,影响了 A/D 转换速度。这个缺点还产生了由此导致的 2 个问题:一是时间较长,也会产生时间的误差;二是时间较长,也会存在 V_X 的变化。这样,就影响了 A/D 转换的精度。

2. 一次积分 A/D 原理

在式(5-6)中,如果 E_0 是一个已知的量,则有

$$E_0 C = V_X T_1 / R_1 \quad (5-11)$$

则被测的输入电压 V_X 可以求出

$$V_X = \frac{R_1 C E_0}{T_1} \quad (5-12)$$

很明显,只要有办法把 E_0 确定下来,用一次积分是可以实现 A/D 转换的。这样,既可以提高 A/D 转换的速度,也可以提高精度。

一次积分 A/D 转换原理如图 5-7 所示。它和图 5-6 的区别在于如下几点。

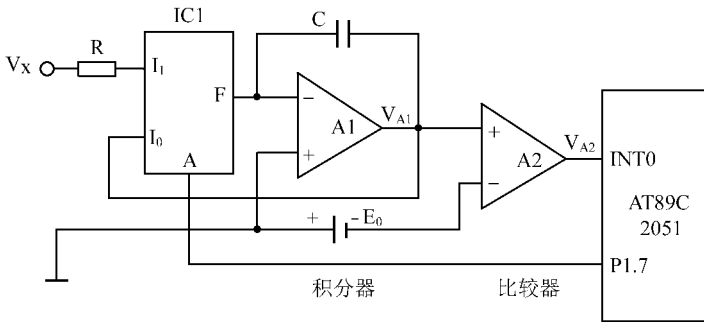


图 5-7 一次积分 A/D 原理

(1) 用模拟开关 IC1 取代采样开关 K。模拟开关的 I_1 接被测电压 V_X , I_0 输入端则接到运算放大器 A1 的输出端 V_{A1} , 输出端 F 和 A1 的负输入端相连。

(2) 运算放大器 A2 接成同相放大方式。

(3) 运算放大器 A1 的正输入端和 A2 的负输入端之间加上一个电源 E_0 , 并且 E_0 的正端接 A1 的正输入端, E_0 的负端接 A2 的负输入端。

当模拟开关 IC1 的地址端 $A=0$ 时,则 IC1 的 I_0 和 F 相连,电容 C 两边短路,则积分器 A1 的输出 $V_{A1}=0$ V。这时,比较器 A2 的正输入端为 0 V,负输入端为 $-E_0$,则有输出 $V_{A2}=“1”$,即高电平。

当模拟开关 IC1 的地址端 $A=“1”$ 时,则 IC1 的 I_1 和 F 相连,被测电压 V_X 通过电阻 R 对电容 C 充电。随着 C 的充电过程, V_{A1} 的电压下降,假定经过时间 T 之后, $V_{A1}=-E_0$,则这时比较器 A2 因两个输入端同时为 $-E_0$ 而输出 $V_{A2}=“0”$,这时电容充电为 Q。显然有

$$Q = E_0 C = \int_0^T \frac{V_X}{R} dt = \frac{V_X}{R} T \quad (5-13)$$

从而有

$$V_X = \frac{RC}{T} E_0 \quad (5-14)$$

由于在式(5-14)中, R, C 和 E_0 是恒定值,而时间 T 可由单片机测出,故被测电压 V_X 可求。

在这个过程中,只需一次对电容 C 进行充电积分,就可以实现 A/D,故称一次积分 A/D 转换。

在图 5-7 中用 AT89C2051 单片机来完成电路的控制、测量和计算工作。这是一种内带 2 KB Flash 存储器,性价比很高的 51 兼容型单片机。将其 INT0 设置为门控方式,定时计数器 T₀ 设置为定时方式 1,P1.7 连模拟开关 IC1 的 A,使之自动完成电路的控制、时间 T 的测量以及 V_x 的计算工作,其时序如图 5-8 所示。

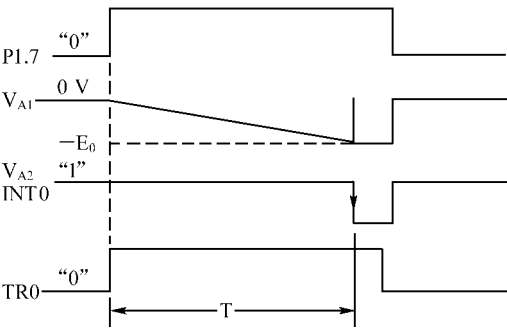


图 5-8 一次积分 A/D 时序图

(5-14)可以写成

P1.7=“0”时,C 被短路,V_{A1}=0 V,V_{A2}=“1”,INT0=“1”,做好准备。

P1.7=“1”,TR₀=“1”,启动定时器 T₀ 开始计时,同时开始正向积分。当 V_{A1}=−E₀ 时,比较器 A2 翻转,V_{A2}=“0”,INT0=“0”。在单片机 INT0 端的下降沿,T₀ 自动停止计时。A/D 转换结束。

在 A/D 结束之后,只要取出计时值 T,就可根据式(5-14)求出被测电压 V_x。如果把 RC 看成积分电路的时间常数 τ=RC,则式

$$V_x = \frac{\tau}{T} E_0 \tag{5-15}$$

特别是令:E₀^{*}=τE₀,则有

$$V_x = \frac{E_0^*}{T} \tag{5-16}$$

式(5-16)在单片机内部可以很简单地只执行除法。为了进一步简化,可以把式(5-16)进行预先处理,形成表格存入存储器中,则 89 系列单片机执行 A/D 转换时,只是一个简单的查表过程。

5.2.3 逐次逼近 A/D 转换

这个转换方法虽增加了元件数,但却提高了解的精度和缩短了转换时间。

逐次逼近(SA)模数变换器由数模转换器(DAC)、比较器和逐次逼近寄存器(SAR)等几部分组成。其主要原理为:将一待转换的模拟输入信号与推测信号相比较,根据推测信号大于还是小于输入信号来决定增大或减小该推测信号,以便向模拟信号逼近。推测信号由 DAC 的输出获得,当推测信号与模拟信号相等时,向 DAC 输入的数字就是对应模拟输入量的数值。

由 SAR 的最高位开始推测,从 SAR 的最高位起依次置 1,每置一位都要进行测试。若模拟输入信号小于推测信号,则将该位清 0;反之则保持为 1。就这样依次比较每一位,直到最末位为止。此时 SAR 内的数值即为 A/D 转换的结果。

这个应用如图 5-9 所示,具有完整模拟比较器的 AT89CX051 微控制器用软件完成 SAR 的功能,减少了元件的数量。DAC 选用型号为 MC1408-8(8 位、电流输出)的廉价 DAC,也可用 MC1408-7(7 位)或 MC1408-6(6 位)。MC1408 系列在满刻度电流输出为 1.992 mA、室温 25℃时可保证精度在±1/2 LSB 内。MC1408-8 的相对精确度低于 0.19%,保障 8 位

的单调性和线性。DAC 输出电流的建立时间为 300 ns。

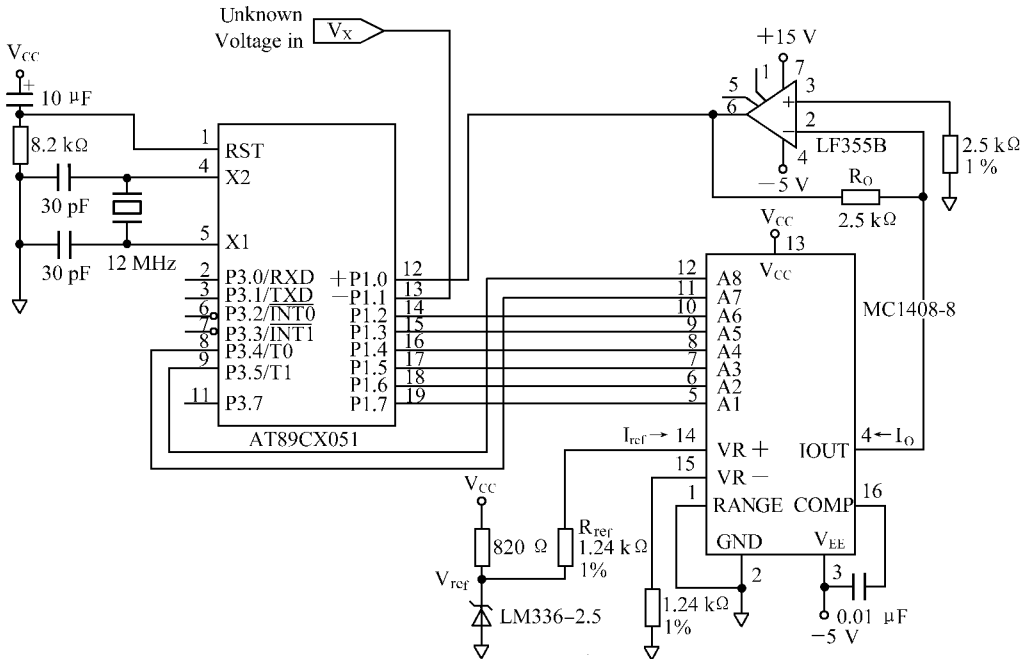


图 5-9 逐次逼近 A/D 转换器

DAC 含有二进制加权、电流导入选择,可用二进制代码选择不同的电流输入。输入电流经一个串联电阻从 LM336-2.5(电流精度电压参考)输入。输出的电流由电压电流转换器(LF355B 运放)转换为电压。选择 LF355B 运放作为 I/V 转换器是因为它的补偿电压低且有高的输出转换速率。I/V 转换器的输出作为推测电压输入到 AT89CX051 比较的正端。当推测电压超过被测的未知电压时,比较器输出为高,这可由软件检测到并用于确定 SAR 的值。

LM336-2.5 参考提供额定的 2.490 V 输出(V_{ref})。实际电压可能是 2.390~2.590 V。详细的资料请参考 LM336-2.5 数据资料。连接在 DAC 引脚 14 的参考电阻(R_{ref})的额定值是 1 240 Ω ,产生的参考电流为 2.008 mA($2.490 \text{ V}(I_{ref})/1\,240 \Omega$)。DAC 的 8 位数据端输入 00H~FFH 则可得 0($I_{ref} \times 0/256$)~2.000($I_{ref} \times 255/256$) mA 的电流。注意 DAC 输出电流的符号与基准电流的符号是相反的。输出电压由 DAC 的输出电流 I_O 和 I/V 转换器增益电阻 R_O 决定。额定的满量程电压输出为 $2.000 \text{ mA} \times 2\,500 \Omega = 5.000 \text{ V}$ 。

这个电路没有为补偿或增益提供调整。由于 LF355 运放是低补偿电压规格所以无需补偿电压调整。如果补偿电压必须调整,则增加补偿调整电路可在 F355B 数据资料中找到。增益的调整可以通过改变 I/V 转换器的增益电阻 R_O 达到。

运放的同相输入接的电阻的阻值应该和增益电阻相同,以便平衡输入的偏差电流。接在 DAC 的引脚 15 的 1 240 Ω 电阻和接在运放引脚 3 的 2 500 Ω 电阻可以除去,其对电路的性能无太大影响。

MC1408-8 DAC 需要 +5.0 V 的电源供应及 -5~15 V 的电源供应,±5.0 V 电源选用小功率的电源。LF355B 运放的双极电压为 ±5~±15 V。为了和 DAC 兼容,可在负极选 5 V,当然如果需要可换成 15 V。

A/D 转换的速度由下列因素决定：DAC 的稳定时间和转换率；比较器的转换率和响应时间；运放的稳定时间；执行逐次逼近算法的时间。这个 DAC 输出的稳定时间和比较器响应时间比起运放的延迟及执行 SA 算法的时间是微不足道的，所以可能忽视。运放的最大输入电压是 5 V，需要 1 μ s 的转换时间和 4 μ s 的稳定时间（参见 LF355B 数据资料）。在这个软件中包含了这个延迟。使用 12 MHz 的系统时钟则指令周期为 1 μ s，所以在 300 μ s 内便可完成 8 位的 A/D 转换。在转换过程未知的输入电压必须保持不变。

逐次逼近 A/D 转换器的明显缺点是需要双极电源和大量的微控制器 I/O 口来控制 DAC。LF355B 运放的正极 +15 V 可用 +5 V 代替，而功能不变。对于 7 位或 6 位的 DAC，微控制器只需消耗 7 位或 6 位的 I/O 口。当然也可以用串行输入的 DAC（较贵）代替并行输入的 DAC，以便减少 I/O 口的消耗。

5.3 显示接口技术

89 系列单片机的应用中，通常要进行信息显示，而显示器件大多是采用 LED 或 LCD。对 LED 和 LCD 器件来说，它们的接口方法是不同的。因为 LED 一般所需的电流较大，而显示器件很多是 7 段数字显示器，所以，在显示时要求给出和显示数字对应的显示码。对于 LCD 器件来说，现在已经配置了专门的驱动电器，在进行信息显示时，往往把信息写入驱动电路的存储器中。由于存储器和 LCD 显示屏是一种映射关系，故只要写入其存储器的信息适当，则可以在 LCD 中显示出对应的正确信息。

5.3.1 LED 显示接口技术

在 89 系列单片机中，AT89C1051 和 AT89C2051 只有 P1 和 P3 两个 I/O 端口；而 AT89C51，AT89C52，AT89LV51 和 AT89LV52 则有 P0，P1，P2 和 P3 共 4 个 I/O 端口。同时，这些单片机的 I/O 引脚所能承受的低电平灌入电流不同，如表 5-1 所列。

表 5-1 I/O 引脚的负载能力

型 号	AT89C1051 AT89C2052	AT89C51 AT89LV51 AT89C52 AT89LV52
每条引脚最大电流/mA	20	10
所有引脚最大总电流/mA	80 含 P1，P3 端口	71 其中：26(P0 端口) 15(P1 端口) 15(P2 端口) 15(P3 端口)

从表 5-1 中可看出，作为单条引脚，可以允许 10~20 mA 电流流入；但对于所有的 I/O 引脚，总的灌入电流，即低电平流入电流不得超过 71~80 mA。对于扫描式的 7 段 LED 显示器，显然 89 系列可以直接驱动其工作。这时，显示电路的接口结构如图 5-10 所示。在某一时刻，只有一个 7 段显示器通电工作，在最大电流时，应是 7 段都处于亮的显示状态。这时，P1.1~P1.7 都会流过的电流为 I_{OL} 。

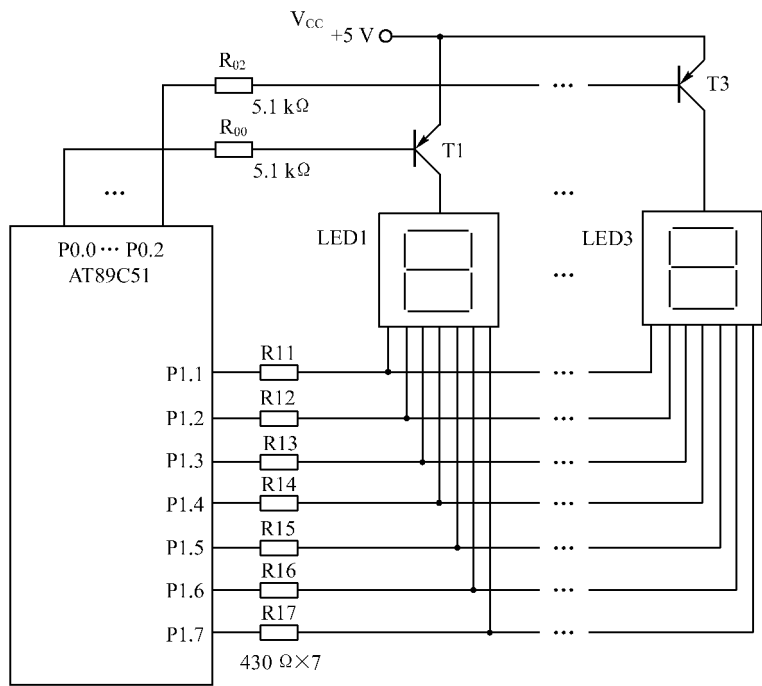


图 5-10 扫描式显示的 7 段显示结构

$$I_{OL} = \frac{V_{CC} - V_T - V_{LED}}{R}$$

其中： V_{CC} 是 LED 显示器的电源， $V_{CC}=5\text{ V}$ ；
 V_T 是晶体管 T1,T2 的压降，一般 $V_T=0.3\text{ V}$ ；
 V_{LED} 是 LED 的压降， $V_{LED}=1.3\text{ V}$ ；
 R 是 LED 阴极串联电阻 R11~R17， $R=430\text{ }\Omega$ 。

因此，流过 P1.1~P1.7 的电流 I_{OL} 为

$$I_{OL} = \frac{5 - 0.3 - 1.3}{430\text{ }\Omega} = 7.9\text{ mA}$$

由于在 P1.1~P1.7 引脚中流入的电流为 7.9 mA，小于每条引脚允许的最大电流 10 mA，故是安全的。另外，P1.1~P1.7 引脚的总电流为 $7.9 \times 7\text{ mA}$ ，即 55.3 mA，也小于允许的总电流 71 mA，所以，总的来说，不会超过芯片的功率。因此，图 5-10 所示的 LED 显示结构是一种有效的电路。

在占用 I/O 引脚较多时，就希望节省在显示时所用的 I/O 引脚，以供其他目的应用。因此，图 5-10 所示的结构就会显得占用太多的 I/O 引脚。为了减少显示时所占用的引脚，有时就在单片机之外加上一片专门的显示驱动电路，而这个显示驱动电路和单片机之间采用串行通信，这样就可以节省较多的 I/O 引脚。

外加专用显示驱动芯片的接口方法如图 5-11 所示，其中 MC14499 是具有串行输入功能的 4 位 LED 显示器驱动能力的芯片。单片机 AT89C51 通过 P1 端口的 P1.1~P1.3 和 MC14499 进行串行数据传送，MC14499 则直接去驱动 4 个 7 段 LED 数字显示器显示。

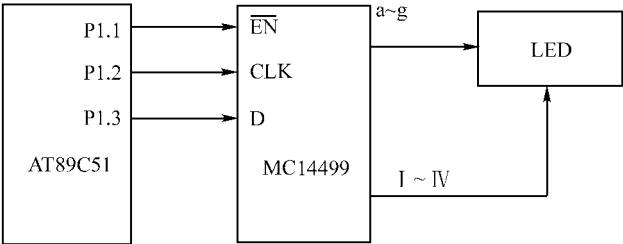


图 5 - 11 外加 LED 显示芯片

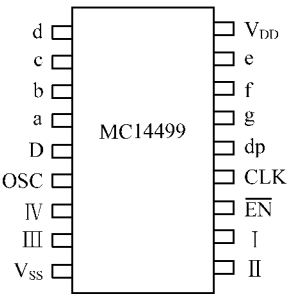


图 5 - 12 MC14499 引脚

MC14499 是 MOTORLA 公司推出的产品,具有管理 4 位 LED 显示器的能力,其显示方式采用动态扫描,功耗较低。由于 MC14499 是一种串行输入接口芯片,所以与单片机的接口很简单,既可以利用单片机的串行口进行控制,又可以采用普通并行 I/O 口进行控制。

MC14499 的引脚如图 5 - 12 所示。其引脚的功能说明如下:

- V_{DD} :正电源端;
- V_{SS} :电源地;
- a~g:7 段显示输出;

- dp:小数点显示输出;
- I ~ IV:4 个位选通端;
- OSC:晶振输入端,外接电容;
- D:串行数据输入端;
- CLK:时钟输入端,为串行输入数据提供同步时钟信号;

$\overline{EN}$:使能端,为低电平时,允许 MC14499 接收串行输入数据;为高电平时,片内移位寄存器将数据送入锁存器中锁存。

MC14499 芯片内部由 20 位移位寄存器、锁存器、多路输出器、译码器和振荡器等组成。外部输入的数据进入移位寄存器,在使能端置位时,再将数据送至锁存器。多路输出从锁存器取出数据送到译码驱动器,经译码后转换成 7 段码送到段和小数点驱动器输出。另外,片内振荡器产生的信号经过 4 分频、译码后,提供 4 个位控制信号,再经驱动器送至 4 条位控制线,以实现动态扫描。

MC14499 的串行输入数据时序图如图 5 - 13 所示。由图中可以看出,当使能端为低电平时,在时钟输入的下降沿串行数据被送入 MC14499 中;当使能端为高电平时,对 MC14499 的输入数据无效。

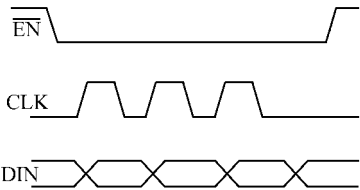


图 5 - 13 MC14499 的串行输入数据时序图

MC14499 每次能接收 20 位串行输入数据,在这接收的 20 位数据中,前 16 位表示 4 个 4 位 BCD 码,最后 4 位表示小数点选择位,其具体的 20 位数据输入格式如表 5 - 2 所列。

表 5-2 20 位数据输入格式

20 19 18 17	16 15 14 13	12 11 10 9	8 7 6 5	4 3 2 1
LSB MSB	LSB MSB	LSB MSB	LSB MSB	Ⅳ Ⅲ Ⅱ Ⅰ
Ⅳ 段 BCD 码	Ⅲ 段 BCD 码	Ⅱ 段 BCD 码	Ⅰ 段 BCD 码	小数点选择

显示控制程序十分简单,主要完成下列有关工作。

- ① 选通 MC14499,准备显示。
- ② 把需显示的 4 个 BCD 码送入 MC14499。
- ③ 把小数点选择码送入 MC14499。
- ④ 禁止、屏蔽 MC14499。

在电路连接中,AT89C51 的 P1.1 引脚和 MC14499 的能动端(enable)相连,P1.2 引脚产生 MC14499 所需的串行输入时钟信号并和 MC14499 的时钟输入端 CLK 相连,P.13 引脚用于输出显示的串行数据并且和 MC14499 串行数据输入端 D 相连。

小数点选择位、段码Ⅰ、段码Ⅱ、段码Ⅲ和段码Ⅳ以 BCD 码的形式存放在数据存储器中,这些数据存储区的首地址存放在寄存器 R0 之中。

当 MC14499 接收完 20 位数据时,必须使其能动端为高电平,以屏蔽其工作,防止干扰信号的串入。否则,MC14499 在干扰信号串入之后会显示错误的结果。而在 MC14499 开始接收显示数据时,则应使其能动端为低电平,直到数据接收完毕。

显示程序如下所示:

```
CLR      P1.1      ;选通芯片
MOV      R2, #2     ;准备送入前 16 位数据
LOOP1: MOV      R1, #8   ;准备送入前 16 位数据
MOV      A, @R0     ;数据送入累加器
LCALL    WRT0       ;调用数据写入子程序
INC      R0         ;存储区地址加 1
DJNZ     R2, LOOP1
MOV      A, @R0     ;后 4 位数据送入累加器
MOV      R1, #4     ;准备送入后 4 位数据
LCALL    WRT0       ;调用数据写入子程序
SETB     P1.1      ;禁止对芯片操作
END      ;程序结束
WRT0: SETB     P1.2   ;时钟高电平
RLC      A          ;数据送入进位标志位
MOV      P1.3, C    ;数据送入 MC14499
CLR      P1.2      ;时钟低电平
DJZN     R1, WRT0
RET      ;子程序结束
```

5.3.2 LCD 显示接口技术

LCD 显示器一般需要专门的驱动芯片进行显示控制,所以,LCD 显示器通常把 LCD 显示

屏和控制模块结合在一起组成一个整体。由于这种显示器包含了 LCD 显示屏和控制模块,故有时也称为 LCD 显示器模块。在实际应用中,LCD 显示器模块是应用极广泛的组件,在这一节中,介绍 AT89C51 和 LCD 显示器模块的接口技术。

1. 图形点阵式 LCD 显示器模块

图形点阵式 LCD 显示器模块有各种各样的型号,功能大小也不尽相同。适合于小型仪器的模块中较为典型的是 EDM12864B。

EDM12864B 液晶显示器模块是全屏幕图形点阵式液晶显示器组件,由控制器、驱动器和全点阵液晶显示器组成。可完成图形显示,也可以显示汉字(4×8 个 16×16 点阵汉字),与 CPU 接口是 8 位数据线和几条地址线,所以,和单片机 AT89C51 相连接十分方便、简单。EDM12864B 采用 3 种电源线,它们分别是 V_{CC} 、GND 和 V_{EE} 。在工作时, $V_{CC}=5\text{ V}$, $GND=0\text{ V}$, $V_{EE}=0\sim-12\text{ V}$ 。 V_{EE} 是用于调节 LCD 显示亮度的偏置电压。当 V_{EE} 的电压调节到恰当的值时,可取得最理想的亮度。

EDM12864B 内部含有 8 个主要部件,即 LCD 显示屏、指令寄存器 IR、数据寄存器 DR、忙标志 BF、显示控制触发器 DFF、XY 地址计数器、显示数据存储寄存器 DDRAM、Z 地址计数器。整个模块有 20 条引脚,其中 18 条引脚用于电源以及信息数据传送。模块还具有 7 种工作指令,专门用于单片机对 LCD 显示器模块执行控制。

EDM12864B 型 LCD 显示器模块的外形结构如图 5-14 所示。

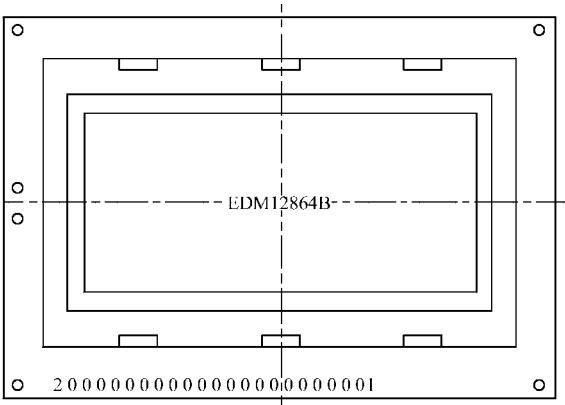


图 5-14 EDM12864B 型 LCD 显示器模块外形结构

该模块中间是一个 66.52×33.24 mm 的显示屏,有 128×64 点的显示阵列。每点的尺寸为 0.48×0.48 mm,点间距离为 0.04 mm。显示器模块的功耗为 5.5~10.5 mW。它有 20 个引脚点,排列在显示屏的下端,编号从右到左进行编排。引脚点的编号和符号如图 5-15 所示。

PIN	1	2	3	4	5	6	7~14	15	16	17	18	19	20
SYM	V_{SS}	V_{DD}	V_{EE}	D/I	R/W	E	DB0~DB7	CS1	CS2	RST	V_{EE}	—	—

图 5-15 EDM12864B 模块的引脚编号含义

2. 显示器模块的部件

EDM12864B 型 LCD 显示器模块包括 8 个主要部件,这些部件分别说明如下。

(1) 指令寄存器(IR)

IR 用来寄存指令码,当 $D/I=0$ 时,在 E 信号下降沿的作用下,指令写入 IR。

(2) 数据寄存器(DR)

DR 是用来寄存数据的。当 $DR=1$ 时,在 E 信号的作用下,图形显示数据写入 DR,或由 DR 读到 $DB7\sim DB0$ 数据总线。DR 和 DDRAM 之间的数据传输是组件内部自动执行的。

(3) 忙标志(BF)

BF 标志组件内部的工作情况。 $BF=1$ 表示组件在进行内部操作,此时组件不接受外部指令和数据; $BF=0$ 时,组件为准备状态,随时可接受外部指令和数据。

(4) 显示控制触发器(DFE)

此触发器用于控制组件屏幕显示的开和关。 $DFE=1$ 为开显示,DDRAM 的内容就显示在屏幕上; $DFE=0$ 为关显示。

(5) XY 地址计数器

XY 地址计数器是一个 9 位计数器。高 3 位是 X 地址计数器,低 6 位是 Y 地址计数器。XY 地址计数器实际上是作为 DDRAM 的地址指针,X 地址计数器为 DDRAM 的页指针,Y 地址计数器为 DDRAM 的 Y 地址指针。

(6) 显示数据 RAM(DDRAM)

DDRAM 是存储图形显示数据的。数据为 1 表示显示选择;数据为 0 表示显示非选择。DDRAM 与地址和显示位置的关系见 DDRAM 地址表。

(7) Z 地址计数器

Z 地址计数器是一个 6 位计数器。此计数器具备循环计数功能,是用于显示行扫描同步的。当一行扫描完成,此地址计数器自动加 1,指向下一行扫描数据。RST 复位后 Z 地址计数器为 0。

Z 地址计数器可以用指令 DISPLAY START LINE 预置。因此,显示屏幕的起始行就由此指令控制,即 DDRAM 的数据从哪一行开始显示在屏幕的第一行。此组件的 DDRAM 共 64 行,屏幕可以循环显示 64 行。

(8) LCD 显示屏

LCD 显示屏用于显示 DDRAM 中所存储的图形数据。当用于显示图像时,可利用全屏 128×64 的格式显示;当用于显示英文字母时,可用于显示 8 行英文,每行 20 个字母;当用于显示汉字时,可显示 4 行汉字,每行 8 个汉字。即可显示 160 个英文字母或 32 个汉字。

3. 显示器模块的外部接口功能

显示器模块的外部接口是通过它的外接引脚,由外部对其进行读写操作来共同形成的。外接引脚提供了接口的通道,而外部单片机对模块的读写操作则执行了单片机和模块之间的信息交流,也就是对显示内容的交流。

(1) 外接引脚编号、符号及其含义

外接引脚编号为 1~20。它所对应的符号及含义如表 5-3 所列。

表 5-3 外接引脚及其含义

引脚号	引脚名称	电 平	功能描述
1	V _{SS}	—	电源地:0 V
2	V _{DD}	—	电源电压: +5 V
3, 18	V _{EE}	—	液晶显示器驱动电压:0~-12 V
4	D/I	H/L	D/I=“H”,表示 DB7~DB0 为显示数据; D/I=“L”,表示 DB7~DB0 为指令数据
5	R/W	H/L	R/W=“H”,E=“H”,数据写到 DB7~DB0; R/W=“L”,E=“H→L”,数据写到 DB7~DB0
6	E	H, H→L	使能信号:R/W=“L”,E 信号下降沿锁存 DB7~DB0, R/W=“H”,E=“H”,DDRAM 数据读到 DB7~DB0
7~14	DB0~DB7	数据总线	
15~16	CS1,CS2	H/L	高电平有效,CS1=1,CS2=0,选择左半屏,相反则选右半屏
17	RST	H/L	低电平时复位
19~20	空脚		

(2) 读/写时序

利用单片机可以实现对 LCD 显示器模块的读写操作,也就是把显示信息取出或写入到模块中的显示数据存储器 DDRAM 中。LCD 显示器模块内部 DDRAM 的读写有一定的时序要求。在这里分别介绍其读时序和写时序。

① EDM12864B 型 LCD 显示器模块的读时序

LCD 显示器模块的读时序由信号 E,R/W,CS,D/I 和 DB0~DB7 组成。E 是能动信号,R/W 是读写信号,D/I 是显示/指令标志信号,CS 是选择信号,DB0~DB7 是数据信号。

EDM12864B 的读时序如图 5-16 所示。从图中可以看出:当 R/W=1 时,执行读操作,

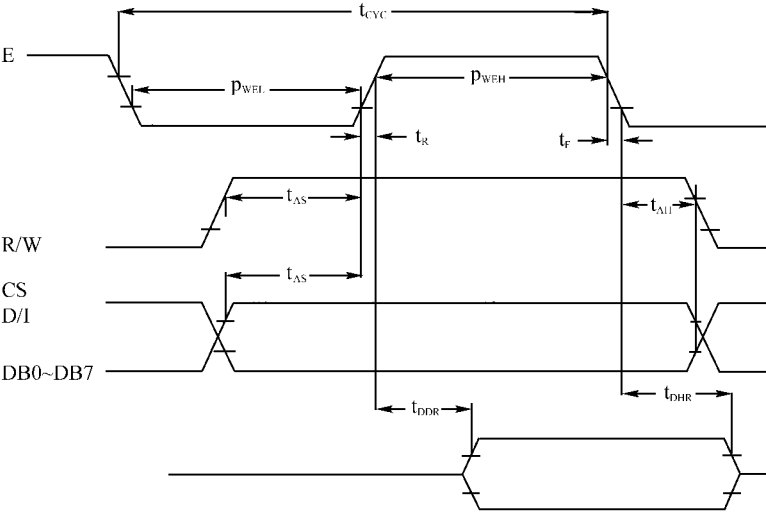


图 5-16 EDM12864B 的读时序

在 E 信号的上升沿开始执行读出,把 EDM12864B 内的 DDRAM 内容读出到数据总线 DB0~DB7 上。这时,如果 D/I=1,则读出的是显示数据;如果 D/I=0,则读出的是指令数据。CS2=1,CS1=0,读右半屏的内容;CS2=0,CS1=1,读左半屏的内容。

时序图中的有关参数的含义如表 5-4 所列。

表 5-4 读写时序图的参数含义

名 称	符 号	最小值	典型值	最大值	单 位
E 周期时间	t_{CYC}	1 000	—	—	ns
E 高电平宽度	P_{WEH}	450	—	—	ns
E 低电平宽度	P_{WEL}	450	—	—	ns
E 上升时间	t_R	—	—	25	ns
E 下降时间	t_F	—	—	25	ns
地址建立时间	t_{AS}	140	—	—	ns
地址保持时间	t_{AH}	10	—	—	ns
数据建立时间	t_{DSW}	200	—	—	ns
数据延迟时间	t_{DDR}	—	—	320	ns
写数据保持时间	t_{DHW}	10	—	—	ns
读数据保持时间	t_{DHR}	20	—	—	ns

② EDM12864B 型 LCD 显示器模块的写时序

写时序和读时序的区别在于 R/W 信号为低电平,即 R/W=0。这时,必须先把写入的数据由外部单片机送入数据总线 DB0~DB7,然后在能动信号 E 的下降沿时,把数据从 DB0~DB7 写入到模块的显示数据存储器 DDRAM 中。同样,在 D/I=1 时,写入的是数据;D/I=0 时,写入的是指令。在 CS2=1,CS1=0 时,把数据写入右半屏;在 CS2=0,CS1=1 时,把数据写入左半屏。

EDM12864B 的写时序如图 5-17 所示。

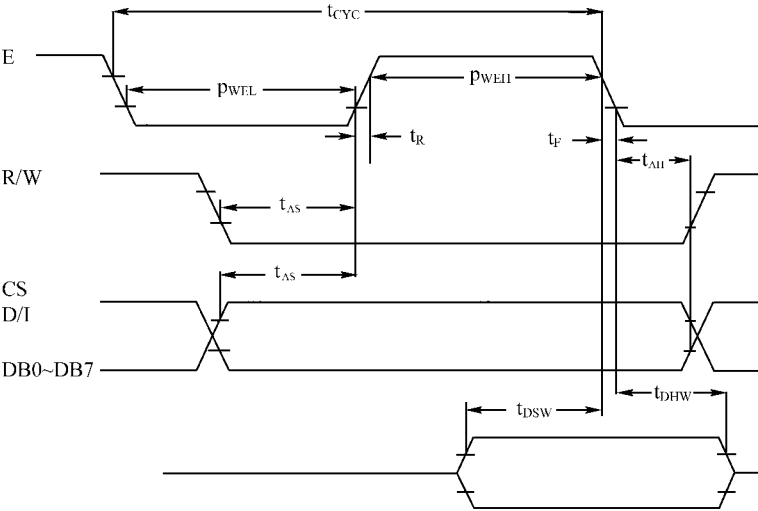


图 5-17 EDM12864B 的写时序

4. AT89C51 和 LCD 显示器模块的接口

AT89C51 和 LCD 显示器模块的接口包括硬件接口和软件接口两个部分。硬件接口是指 AT89C51 和 LCD 显示器模块的信息逻辑接线方法。软件接口是指 AT89C51 对 LCD 显示器模块的显示控制方法及程序。下面分别介绍这两种接口方法。

(1) AT89C51 和 LCD 显示器模块的硬件接口

AT89C51 和 EDM12864B 的硬件接口连线如图 5-18 所示。在图中看出,LCD 显示器模块的能动信号 E 是由 AT89C51 的读写信号 $\overline{RD}$, $\overline{WR}$ 和地址信号 ADDR 组成的。其中 ADDR 只是 AT89C51 输出的一个高位地址,或是 I/O 端口的一条输出引脚;当 ADDR=0 时,允许读写信号 $\overline{RD}$, $\overline{WR}$ 产生能动信号 E。

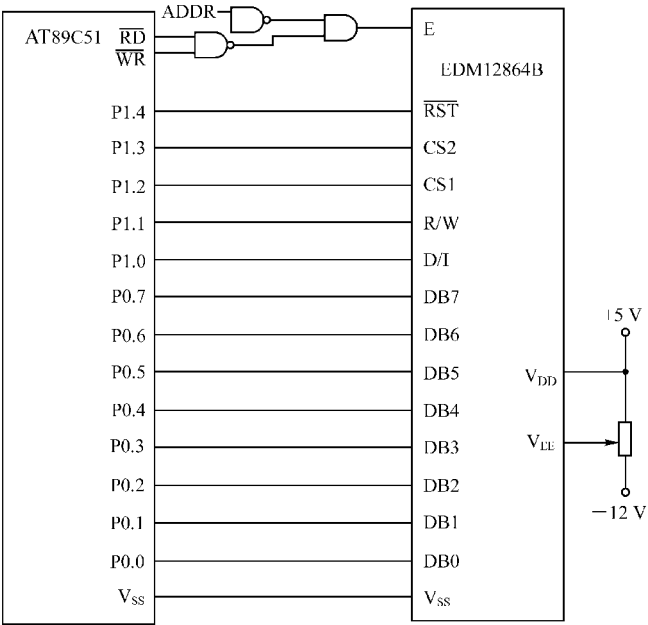


图 5-18 AT89C51 和 LCD 显示器模块接口

AT89C51 的 P1 端口的 P1.0~P1.4 用于产生控制信号,用于控制 LCD 显示模块的工作。其中 P1.0 用于产生数据指令标志信号 D/I;P1.1 用于产生读写信号 R/W;P1.2,P1.3 用于产生选择信号 CS1,CS2;P1.4 则用于产生复位信号 RST。

AT89C51 的 P0 端口用于传送数据或指令,故 P0.7~P0.0 和 LCD 显示器模块的 DB7~DB0 相连。

为使显示屏能具有满意的亮度,在+5 V 和-12 V 之间接一个电位器,用于调节 V_{EE} 的电压。 V_{EE} 的电压一般取 0~-12 V。

(2) AT89C51 和 LCD 显示器模块的软件接口

在 EDM12864B 型 LCD 显示器模块应用中,有 7 种指令格式。这些指令用于对模块执行专门的控制功能。这些指令及其功能分别说明如下。

① 显示开关控制

D=1:开显示(DISPLAY ON)。

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	D

D=0:关显示(DISPLAY OFF)。此时的 DDRAM 内容不变。只要 D=0 变成 D=1,原来的显示就会显示在屏幕上。

② 设置显示起始行

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	A5	A4	A3	A2	A1	A0

在 Z 地址计数器一节已经描述了显示起始行是由 Z 地址计数器控制的。A5~A0 这 6 位地址自动送入 Z 地址计数器,起始行的地址可以是 0~63 的任意一行。

举例:

选择 A5~A0 是 62,则起始行与 DDRAM 行的对应关系如下:

DDRAM 行: 62 63 0 1 2 3 … 60 61

屏幕显示行: 1 2 3 4 5 6 … 63 64

③ 设置页地址

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	A2	A1	A0

设置页地址指令为 SET PAGE“X ADDRESS”。所谓页地址就是 DDRAM 的行地址。8 行为一页,组件共 64 行即 8 页。A2~A0 表示 0~7 页。

读写数据对页地址没有影响。页地址由本指令或 RST 信号改变。复位后页地址为 0。页地址与 DDRAM 的对应关系如表 5-5 所列。

表 5-5 DDRAM 地址表

CS1=1,CS2=0								CS1=0,CS2=1							
Y=	0	1	2	3	...	062	63	0	1	2	3	...	62	63	行 号
X=0	DB0						DB0	DB0						DB0	0
	↓						↓	↓						↓	↓
	DB7						DB7	DB7						DB7	7
↓	DB0						DB0	DB0						DB0	8
	↓						↓	↓						↓	↓
	DB7						DB7	DB7						DB7	55
X=7	DB0						DB0	DB0						DB0	56
	↓						↓	↓						↓	↓
	DB7						DB7	DB7						DB7	63

④ 设置 Y 地址

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A5	A4	A3	A2	A1	A0

设置 Y 地址指令为 SET Y ADDRESS。此指令的作用是将 A5~A0 送入 Y 地址计数器

作为 DDRAM 的 Y 地址指针。在对 DDRAM 进行读写操作后,Y 地址指针自动加 1,指向下一个 DDRAM 单元。

⑤ 读状态

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	BF	0	ON/OFF	RST	0	0	0	0

读状态指令为 STATUS READ。当 R/W=1,D/I=0 时,在 E 信号为“H”的作用下,状态分别输出到数据总线(DB7~DB0)的相应位。

BF:见 BF 标志位一节。

ON/OFF:表示 DFF 触发器的状态,见 DFF 触发器一节。

RST:RST=1 表示内部正在初始化,此时组件不接收任何指令和数据。

⑥ 写显示数据

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	D7	D6	D5	D4	D3	D2	D1	D0

写显示数据指令为 WRITE DISPLAY DATA。D7~D0 为显示数据。此指令把 D7~D0 写入相应的 DDRAM 单元。Y 地址指针自动加 1。

⑦ 读显示数据

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

读显示数据的指令为 READ DISPLAY DATA。此指令把 DDRAM 的内容 D7~D0 读到数据总线 DB7~DB0。Y 地址指针自动加 1。

对图 5-18 所示的电路,说明有关指令的使用方法,如下面给出的程序段所示。在下面程序中有关符号含义为:R0 为间接寻址寄存器,ID 为指令码,DATA 为显示数据。

① 显示开/关控制(DISPLAY ON/OFF)程序段。

```
CLR          P1
SETB         RST
...
SETB         CS1
LCALL        BF
CLR          R/W
CLR          D/I
MOV          A, ID
MOVBX        @R0, A           ;左 64 列
...
CLR          CS1
SETB         CS2
LCALL        BF
CLR          R/W
CLR          D/I
```

```
MOV      A, ID
MOVX     @R0, A                ;右 64 列
```

② 读状态程序段。

```
BF: SETB  R/W
CLR       D/I                ;状态读到 A
MOVX     A, @R0
JB        ACC. 4, BF
JB        ACC. 7, BF
RET
```

③ 写显示 RAM 数据(WRITE DISPLAY DATA)程序段,假如 X 地址、Y 地址已设置完成。

```
...
SETB     CS1
LCALL    BF
CLR       R/W
SETB     D/I
MOV      A, #DATA
MOVX     @R0, A                ;数据写入相应 DDRAM 单元
```

5.4 和 PC 串行通信技术

在很多信息处理的场合,往往用单片机执行信息的检测任务,然后把检测结果传送到 PC 机中去,最后由 PC 机进行信息的处理过程。这样,就需要在单片机和 PC 机之间建立一种通信结构。这种通信结构可以在单片机和 PC 机之间进行有效而正确的信息传送。在这一节中,介绍 89 系列单片机和 PC 机之间的串行通信方法和技术。

5.4.1 串行通信的电路结构和原理

串行通信的电路结构如图 5-19 所示。

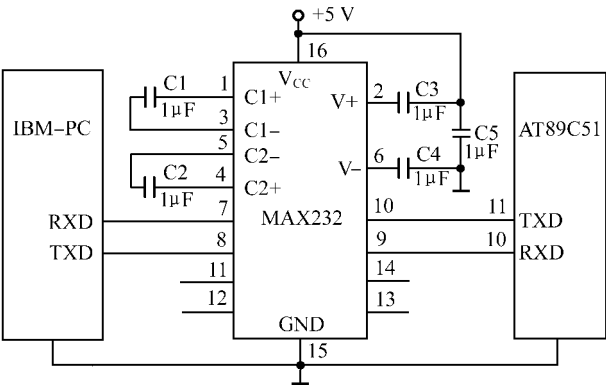


图 5-19 串行通信电路结构

AT89C51 单片机通过串行口直接接收 PC 机传送来的串行数据,然后把接收的数据存入数据存储器。同样,AT89C51 通过串行口直接把数据传送给 PC 机。首先,AT89C51 从数据存储器取出数据,接着送到串行口,由串行口发送给 PC 机。

由于 PC 机的串行口都是 RS-232C 标准的接口,所以,其输入输出在电平上和采用 TTL 电平的 AT89C51 在接口时会产生电平不同的问题。为了解决这个问题,在 PC 机和 AT89C51 单片机的串行通信电路中加入了 MAXIM 公司的 MAX32 芯片。这种芯片可以实现 TTL 电平和 RS-232C 接口电平之间的转换,也就是可以把 5 V 电平表示“1”,0 V 电平表示“0”的逻辑,转换成-3~15 V 电平表示“1”,+3~15 V 表示“0”的逻辑。这样,在 PC 机和 AT89C51 单片机进行串行通信时,就可以顺利执行。

串行接口有同步和异步两种基本的通信方式。AT89C51 和 PC 机进行串行通信时采用异步通信方式。在这种异步通信方式中,数据的发送和接收按照规定的格式进行,如图 5-20 所示。

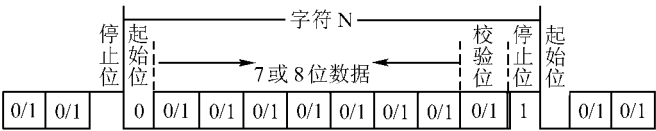


图 5-20 串行通信的异步方式

数据发送和接收均将起始位和停止位作为开始和结束的标志。在异步通信中,起始位占用一位(低电平),用来表示字符开始。其后为 7 或 8 位的数据编码,第 8 位通常作为奇偶校验位。最后为停止(高电平),用来表示字符传送结束。上述字符格式通常作为一个串行帧,如无奇偶校验位,即为常见的 N. 8. 1 帧格式。

串行通信中,每秒传送的数据位称为波特率。如数据传送的波特率为 1 200 波特,采用 N. 8. 1 帧格式(10 位),则每秒传送字节为 120 个,而字节中每一位传送时间即为波特率的倒数: $t=1/1\,200=0.833\text{ ms}$ 。同样,如数据传送的波特率为 9 600 波特,则字节中每一位传送时间为 $t=1/9\,600=0.104\text{ ms}$ 。

在实际工作中,可以根据 PC 机和 AT89C51 通信时所约定的波特率,进行有效的通信。一般情况下,AT89C51 采用方式 1 的串行工作方式,这样和 PC 机可以顺利地进行异步通信。除了电路结构之外,要实现 PC 机和 AT89C51 之间的异步通信,还需要有恰当的通信软件。

5.4.2 串行通信的软件

在 PC 机和 AT89C51 进行通信时,需要两部分软件,一部分是 PC 机串行通信软件,另一部分是 AT89C51 串行通信软件,这两种软件以不同的方式和软件编制。PC 机的通信软件采用 Visual Basic 语言编程,而 AT89C51 采用汇编语言进行编程。下面分别对这两种通信软件进行介绍。

1. PC 机串行通信软件

PC 机采用 Visual Basic(简称 VB)编程。目前,VB 已成为 Windows 系统开发的主要语言,以其高效、简单易学及功能强大的特点越来越为广大程序设计人员及用户所青睐。VB 支持面向对象的程序设计,具有结构化的事件驱动编程模式并可以使用无限扩增的控件,而且可以十分简便地作出良好的人机界面。在标准串口通信方面,VB 提供了具有强大功能的通信

控件 MSCOMM, 文件名为 MSCOMM.VBX。该控件可设置串行通信的数据发送和接收, 对串口状态及串口通信的信息格式和协议进行设置。这是一个标准的 10 位串口通信, 包括 8 位标准数据和数据的起始位和停止位。在发送或接收数据过程中触发 OnComm 事件, 通过编程访问 CommEvent 属性, 了解通信事件的类型, 分别进行各自的处理。每个通信控件对应一个串口, 可以设计多个通信控件来访问多个通信口, VB 通信控件设计软件如下:

```

Global comm-time As Integer           ;定义一些全局变量
Global data (1000) As Integer
Global Receive(1000) As Integer
Private Sub Form - Load( )'           ;串口初始化
Comm1. CommPort=1                     ;设置串口 1
Comm1. Settings="9600,N,8,1"         ;波特率:9 600 奇校验,8 位数据,1 位停止位
Comm1. InputLen=0                     ;读取接收缓冲区的所有字符
Comm1. InBufferSize=256               ;设置接收缓冲区为 256 字节
Comm1. OutBufferSize=256              ;设置发送缓冲区为 256 字节
Comm1. PortOpen=True                  ;打开串口 1
Comm1. SThreshold=1                   ;发送缓冲区空触发发送事件
Comm1. RThreshold=1                   ;每一个字符到接收缓冲区都触发接收事件
End Sub
Public Sub send-data( )                ;发送 n 个数据
For i=0 To n
Comm1. Output=Chr(data(i))
Next i
End Sub
Public Sub Comm1-OnComm( )             ;接收数据触发 OnComm( )事件
Select Case Comm1. CommEven           ;在 CommEvent 中接收数据
Case comEvReceive
Receive(comm-time)=Asc(Comm1. Input) ;将接收数据放在 Receive(comm - time)数组中
Comm-time=comm-time+1
End Select
End Sub

```

2. AT89C51 串行通信软件

单片机 AT89C51 采用 MCS51 系列的汇编语言编程。它有一个标准的串行口, 有 4 种工作方式, 其中方式 1 是标准的 10 位异步通信方式, 10 位数据和 PC 机的标准串口相对应, 由串行口控制寄存器 SCON 设置状态, 其字节地址为 98H。

本系统中 AT89C51 串口控制器 SCON 设置为 50H, SM0, SM1 为“0”和“1”, 即为串行工作方式 1; REN 为“1”, 即允许串口接收。另外, 还应使 AT89C51 的中断允许寄存器 IE 开放或禁止所有中断位 EA 为“1”, 开放或禁止串行通道中断位 ES 为“1”, 即允许串行口中断。

执行串行通信时 AT89C51 的程序如下:

```

MOV      SP, #07
INIT:    MOV      TH1, #FD             ;设置波特率

```

```
MOV TL1, #00
MOV TMOD, #20 ;定时器 2 为工作方式 2
MOV TCON, #40 ;启动定时器
SETB ES ;允许串口中断
MOV SCON, #50 ;置串行口工作方式 1
STRAT: CLR EA ;清中断
MOV R0, #DATA1 ;存放接收数据首地址 #DATA1
MOV R1, #DATA2 ;存放发送数据首地址 #DATA2
MOV R2, #N ;数据个数
SETB EA ;开中断
WAIT: DJNZ R2, WAIT ;等待发送接收
CLR EA ;发送、接收完毕,清中断
END ;结束
...
INTS: PUSH ACC ;CPU 状态入栈
PUSH PSW
JB TI, SEND ;发送完毕,有发送中断,转
WAITI: JNB RI, WAITI ;未接收完毕,等待
MOV A, SBUF ;把串行缓冲器接收的结果送 A
MOV @R0, A ;存入有关地址中
CLR 98H ;清接收中断标志 RI
INC R0 ;接收数据存放地址加 1
AJMP OUTIN
SEND: MOV A, @R1 ;把发送数据送 A
MOV SUBF, A ;从 A 送到串行口
INC R1 ;发送数据存放地址加 1
CLR 99H ;清发送中断标志 TI
OUTIN: POP PSW
POP ACC
DEC R2 ;数据个数减 1
RETI
```

在上面程序中,INTS 是串行口中断入口,也就是地址 0023H。串行口中断处理程序所做的工作就是把 R1 内容所指明的地址的内容发送出去,把接收的数据存入 R0 内容所指明的地址中。接收中断标志 RI 的位地址是 98H,而发送中断标志 TI 的地址是 99H。故清 RI 标志是对地址 98H 执行清 0,清 TI 是对地址 99H 执行清 0。在中断处理返回之前对寄存器 R2 减 1,用于监视传送的数据个数。在程序中要特别注意的是:串行口中的发送 SBUF 和接收 SBUF 是两个独立的电路,它们的工作互不影响。

5.5 晶闸管控制接口技术

在工业应用中,经常要用到大功率器件驱动大功率的负载。在各种大功率器件中,最常用

	MOV	TL0,R4	;移相角为 α
	MVO	TH0,R5	
	CLR	TR1	;关 T1
	CLR	TF1	
	MOV	R6,#80H	;送脉宽数据,脉宽 128 μ s
	MOV	R7,#0FFH	
	MOV	TL1,R6	
	MOV	TH1,R7	
M000:	JB	P3.0,M000	;等待过零
MAIN:	JNB	P3.0,MAIN	;过零脉冲上升沿开始计时
	SETB	TR0	;开 T0
M001:	JNB	TF0,MOO1	;T0 计时,等待溢出,形成移相角 α
	CLR	TR0	;关 T0
	CLR	TF0	
	CLR	P1.0	;P1.0 输出“0”,触发晶闸管
	CLR	TR1	;关 T1
	SETB	TR1	;开 T1
M002:	JNB	TF1 M002	;T1 计数,等待溢出,形成脉宽
	SETB	P1.0	;停止触发
	CLR	TR1	;关 T1
	CLR	TF1	
	MOV	TL1,R6	;为下半周期的脉宽作准备
	MOV	TH1,R7	
	CLR	C	;清进位,为修改移相角 α 作准备
	MOV	A,R4	;对 T0 的预置值加 1
	ADD	A,#01H	
	MOV	R4,A	
	MOV	A,R5	
	ADDC	A,#00H	
	MOV	R5,A	
	MOV	TL0,R4	
	MOV	TH01,R5	
	JNB	R4,#0FFH,MAIN	;未达 220 V,继续升压
	JNB	R5,#0FFH,MAIN	
M003:	CLR	P1.0	;到 220 V,保持不变
	AJMP	M003	

5.5.2 三相桥式晶闸管接口技术

在工业应用中,三相电压控制是常用的控制技术,无论是一般的电压控制还是变频控制,都需要对三相桥式的晶闸管电路进行触发控制。在三相控制中,在桥式结构中所需的晶闸管通常有 6 个。由于三相电压相互之间相差 120°的相位,所以,控制的过程相对于单相而言就较为复杂。这主要考虑如下几种原因。

- (1) 每相之间的控制电路必须互相隔离,以防止相间产生短路。
- (2) 每个晶闸管的控制不同,这包括移相的同步起点不同。
- (3) 在控制桥中的 6 个晶闸管必须按照三相的特点进行触发,每相之间相差 120° 。

这样,三相桥式晶闸管的接口就较之单相复杂。接口电路应该包括以下一些电路:三相检测电路、过零检测电路、晶闸管隔离触发电路、相位同步控制电路等。

三相全控桥式晶闸管的接口电路如图 5-23 所示。三相全控桥式电路如图 5-24 所示。

在图 5-23 中看出,三相全控桥式晶闸管接口电路主要由过零及同步检测电路、单片机 AT89C51、锁存器 74LS374 和隔离式晶闸管触发电路组成。这些电路用于进行相位检测、过零检测、隔离触发、相位同步控制等功能。

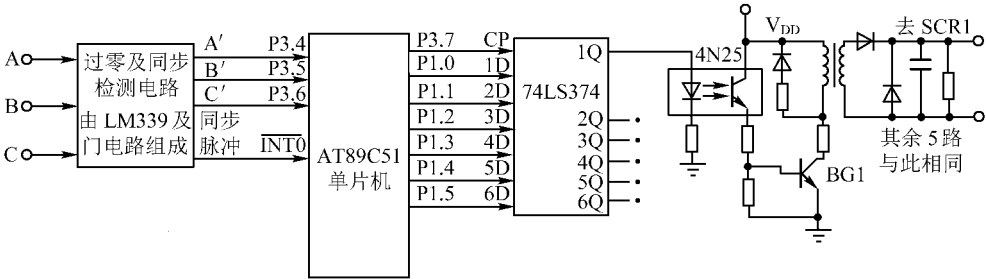


图 5-23 三相全控桥式晶闸管接口电路

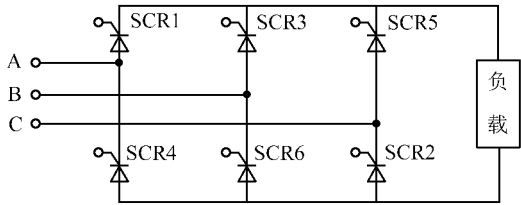


图 5-24 三相全控桥式电路

1. 过零及同步检测电路

过零及同步检测电路用于对三相电源进行过零点检测,以便对各相的移相进行控制。在移相控制中,是以过零点为起始点进行延时触发的,所以,过零点是一个十分重要的基准点。另外,为了对三相电源进行区分,过零及同步检测电路也把三相电源转换成方波,以供单片机进行相序的识别。

过零及同步检测电路所产生的有关信号波形如图 5-25 所示。

在图 5-25 中, A' 、 B' 、 C' 分别是 A、B、C 相的交流电压转换成的方波,这些方波和原来的交流电压相位相同,其上升沿和下降沿与交流电压的过零点对应。同步脉冲用于指示过零点,同步脉冲的下降沿是三相电压的一个过零点。

三相电压所产生的方波 A' 、 B' 、 C' 由半波整流和放大电路形成,这个电路的结构如图 5-26 所示。三相变压器的初级接成三角形,而次级接成星形,故有中性点 O。通过半波整流的单相电压经由比较器 $P_i(i=1,2,3)$ 放大后产生和各相对应的方波,分别由 A' 、 B' 、 C' 输出到单片机 AT89C51 的 P3.4、P3.5、P3.6 的输入端。由于方波的上升沿和下降沿是交流电压的

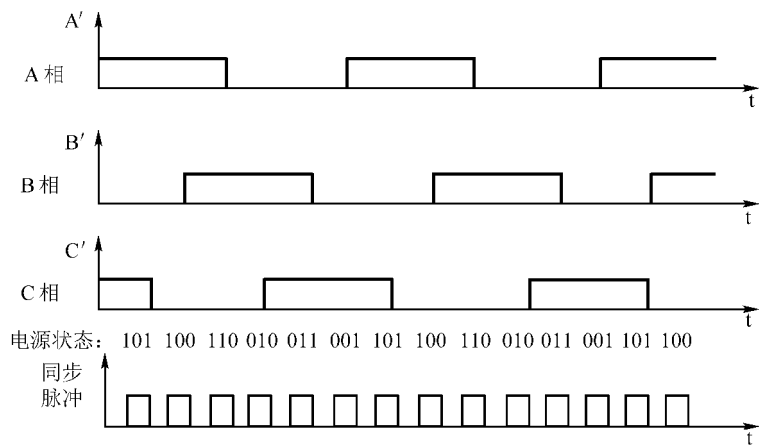


图 5-25 过零及同步检测电路的输出波形

过零点,所以,同步脉冲可以利用方波产生。产生同步脉冲的电路如图 5-27 所示。图中给出的是 A 相同步脉冲的产生电路,其余 2 相相同。当方波从 A' 点输入后,通过 CR 组成的微分电路产生向上的尖峰脉冲和向下的尖峰脉冲,这两个尖峰脉冲分别与方波的上升沿和下降沿对应。上升尖峰脉冲由 D4 二极管整流后送到比较器 P4 的正输入端,从而使 P4 输出一个正脉冲;下降尖峰脉冲通过 D5 二极管加到 P5 的负输入端,则 P5 也产生一个正脉冲输出。这两个正脉冲通过反相后加到与门 Y,之后则在 I 输出。A 相同步脉冲和相关信号的关系如图 5-28 所示。

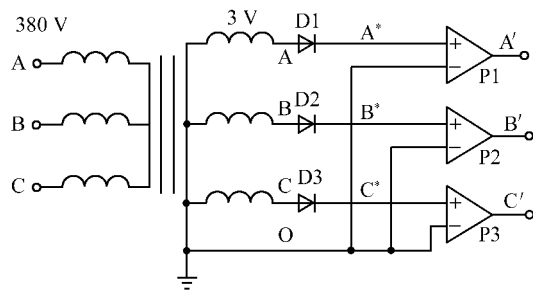


图 5-26 半波整流和放大电路

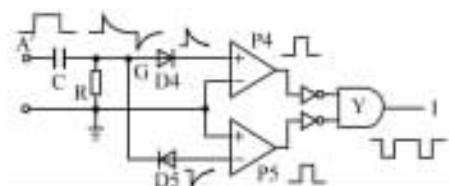


图 5-27 同步脉冲发生电路

把三相产生的同步脉冲进行处理,也就是把各相产生的同步脉冲用一个三端输入的与门执行逻辑与操作,就可以得出最后的同步脉冲信号。该同步脉冲信号就可以送入 AT89C51 的中断输入端 $\overline{\text{INTO}}$,进行过零中断请求。

2. 单片机 AT89C51

在图 5-28 中可以看出,一相电压在一个电压周期内会产生 2 个同步脉冲,很明显三相电压在一个电压周期内会产生 6 个同步脉冲,每个同步脉冲之间相隔 60° 。

AT89C51 单片机的功能就是通过程序产生相应的移相控制信号。移相是由定时器 T0 定时产生的延时时间实现的。

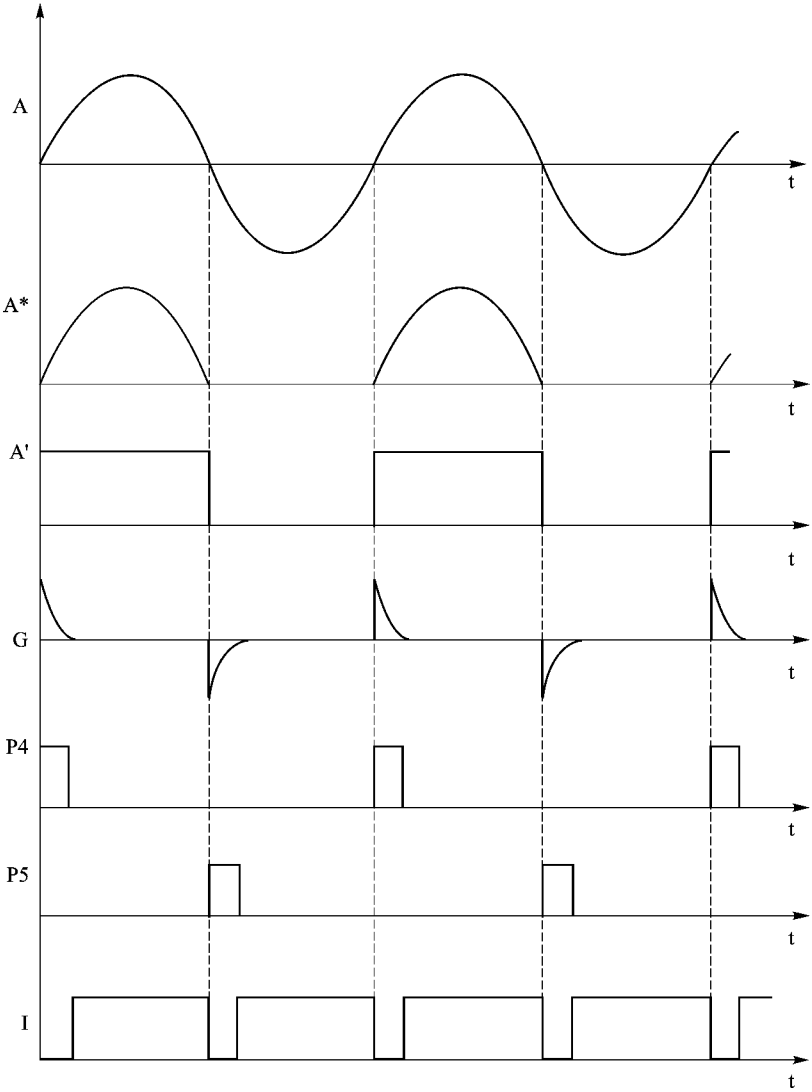


图 5 - 28 同步脉冲的产生

当同步脉冲输入到 AT89C51 的中断端 $\overline{\text{INT0}}$ 时,单片机 AT89C51 产生中断;在中断响应时,把低电平送出到 P3.7,从而 74LS374 不能接收任何数据;同时,T0 开始执行定时计数。当 T0 计数产生溢出时,AT89C51 响应 T0 溢出中断,这时,单片机在 P3.7 输出高电平,把单片机已送出到 P1.0~P1.5 的数据锁存入 74LS374。

定时器 T0 的定时时间所对应的相角最大不能超过 60° ,原因在于每个同步脉冲之间只相隔 60° 。而从 P1.0~P1.5 送出的数据是用于触发晶闸管的,对于三相全控桥式电路在任何时候都必须有 2 个晶闸管导通,才能形成导通回路,因此采用双脉冲触发方式,即在触发时同时触发 2 个晶闸管。在 P1.0~P1.5 输出的触发数据如表 5 - 6 所列,从中可知任何时刻都有 2 个晶闸管被触发,因为输出数据为“1”则触发对应的晶闸管。

表 5-6 触发数据

电源状态			P1 口输出						触发字 (P1.7=P1.6=1)	SCR 导通号
P3.4	P3.5	P3.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0		
1	0	1	1	0	0	0	0	1	E1H	1,6
1	0	0	0	0	0	0	1	1	C3H	1,2
1	1	0	0	0	0	1	1	0	C6H	3,2
0	1	0	0	0	1	1	0	0	CCH	3,4
0	1	1	0	1	1	0	0	0	D8H	5,4
0	0	1	1	1	0	0	0	0	F0H	5,6

在三相整流电路中,一般把相电压的交点称为自然换相点。自然换相点在各相由负到正的过零点再移相 30°的位置。通常的三相桥式整流电路中,对晶闸管的移相角控制都是以自然换相点为起始点考虑的。移相角也称控制角。因此,对于通常的三相桥式整流电路,对晶闸管的移相角有两种限制:一种是电阻性负载,移相角为 0~120°;另一种是电感性负载,移相角为 0~90°。

图 5-23 所示的电路中,是以电压过零点同步移相的,为了以自然换相点为起点进行移相,而过零点和自然换相点之间相差 30°,故可以考虑电阻性负载的移相从 30°开始,一直到 150°,即移相范围也为 120°。对于电感性负载的移相从 30°开始,一直到 120°,即移相范围也为 90°。显然,这样就会和自然换相点为起点同步的结果一致了。在处理中,以 α 表示移相角,则移相范围对于电阻性负载为 β 。

$$\beta = \alpha + 30^\circ; \quad \alpha = 0 \sim 120^\circ$$

对于电感性负载的移相范围为 β 。

$$\beta = \alpha + 30^\circ; \quad \alpha = 0 \sim 90^\circ$$

在实际工作时,把 β 作为定时器 T0 的延时值,则利用过零点为移相的同步点,照样可以得到与自然换相点为移相同步点一样的结果。

由于每个同步脉冲只相隔 60°,所以,定时器 T0 的定时要小于 60°。为了实现恰当的移相,首先在以过零点为同步脉冲的条件下,考虑 T0 定时值 β 的范围值的有关情况。假定 β 的范围可达 0~180°,则显然要分为下列几种区间考虑:0~60°,60~120°,120~180°。

(1) $\beta=0\sim60^\circ$ 。

由于 $\beta=\alpha+30^\circ$,则有 $\alpha=\beta-30^\circ$ 。

在 $\beta<30^\circ$ 时, α 小于 0,故没有意义。

$\beta=30\sim60^\circ$ 时, $\alpha=0\sim30^\circ$

(2) $\beta=60\sim120^\circ$ 。

由于 $\beta=\alpha+30^\circ$,则有 $\alpha=\beta-30^\circ$,从而有 $\alpha=30\sim90^\circ$ 。

(3) $\beta=120\sim180^\circ$ 。

同理有 $\alpha=90\sim150^\circ$ 。

这样,移相角 α 分成 3 个区间:0~30°,30~90°,90~150°,并且和 3 个相邻的同步脉冲对

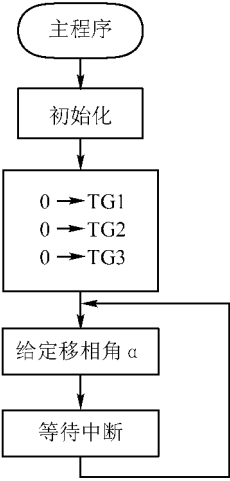


图 5-29 主程序框图

应。在实际中,当负载为电阻性时, $\alpha=0\sim120^\circ$,则所分的区间为: $0\sim30^\circ$, $30\sim90^\circ$, $90\sim120^\circ$;当负载为电感性时, $\alpha=0\sim90^\circ$,则所分的区间为: $0\sim30^\circ$, $30\sim90^\circ$ 。

实现有效的移相,是由 AT89C51 的软件完成的。这些软件包括主程序、外部过零中断处理程序、T0 中断处理程序。它们的框图分别如图 5-29、图 5-30、图 5-31 所示。在主程序中, TG1, TG2, TG3 分别是 3 个触发字 D 的存储单元。TG1 用于存放即时触发字, TG2 用于存放延时一个同步周期的触发字, TG3 用于存放延时 2 个同步周期的触发字。外部过零脉冲中断处理程序框图如图 5-30 所示。它分别按 α 为 $0\sim30^\circ$, $30\sim90^\circ$, $90\sim120^\circ$ 三个区间进行不同的处理。当 α 在 $0\sim30^\circ$ 时,把 $\alpha+30^\circ$ 作为延时数据送入 T0,这相当于以相电压交点为同步脉冲进行移相;同时,取 P3.4~P3.6 的状态求对应触发字 D 作为即时触

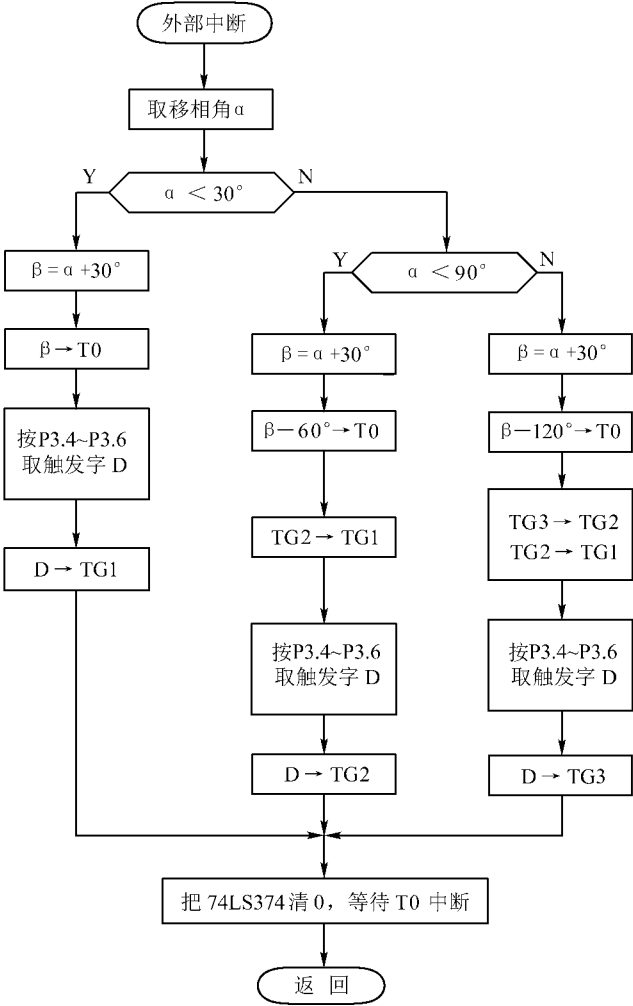
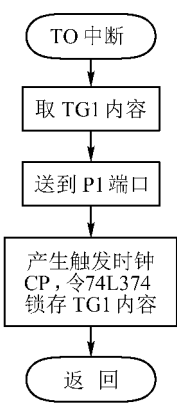


图 5-30 外部过零脉冲中断处理程序框图

发控制用。当 α 在 $30\sim 90^\circ$ 时,把 $\alpha + 30^\circ - 60^\circ$ 作为延时移相数据送入 T_0 ,但是,把当前 $P_{3.4}\sim P_{3.6}$ 状态求得的触发字保存下来,在下一个过零脉冲来时作为触发控制用。当 α 在 $90\sim 120^\circ$ 时,移相数据按 $\alpha + 30^\circ - 120^\circ$ 求出并送入 T_0 ,所求出的触发字则保存到第 3 个过零脉冲来时才作为触发控制用。

T_0 中断处理是在 T_0 溢出时开始执行的。它把触发数据从 $TG1$ 取出送到 $P1$ 端口,并产生 CP 脉冲,使 74LS374 锁存 $TG1$ 的内容,从而触发晶闸管工作。



3. 晶闸管的触发电路

晶闸管的触发电路由锁存器 74LS374、光电耦合电路以及脉冲变压器耦合电路组成,如图 5-23 所示。

图 5-31 T_0 中断处理程序框图

8 位锁存器 74LS374 是一片由 8 个 D 触发器组成的锁存器。它具有 3 态输出,可作总线的接口器件。接受数据的时钟在上升沿起作用。74LS374 的真值表及内部逻辑结构分别如表 5-7 所列和图 5-32 所示。在图中, CP 是时钟触发输入端, $\overline{OE}$ 是输出控制端。74LS374 的封装引脚如图 5-33 所示。

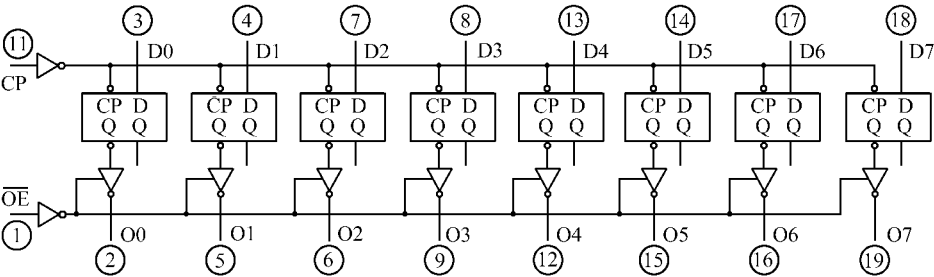


图 5-32 74LS374 的逻辑结构

表 5-7 74LS374 真值表

输入 D_n	CP	$\overline{OE}$	Q_n
H		L	H
L		L	L
X	X	H	Z

注:H—高电平;L—低电平;
X—任意;Z—高阻抗。

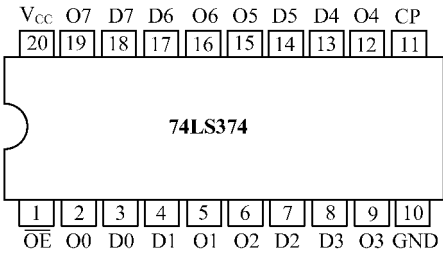


图 5-33 74LS374 的封装引脚

当 74LS374 中锁存了“1”的触发器的输出为高电平时,则对应的光电耦合器的二极管就会导通发光,使其对应的三极管导通,从而使晶体管 BG1 导通,则脉冲变压器会输出一个正脉冲去触发晶闸管。而在 74LS374 中锁存了“0”的触发器输出为低电平,其对应的光电耦合器的二极管不导通,则其对应的光敏三极管截止,从而令 BG1 截止,脉冲变压器不能输出脉冲,所以不能触发对应的晶闸管。

第六章 存储器应用及扩展技术

89 系列单片机的内部存储器采用 Flash 技术制造。这就使用户可以十分方便地对其进行重新编程;在各种开发应用中,无论修改程序或记录数据都极其有用。在一些控制过程中,有时需要根据当时的状况修改控制程序或控制算法,有时需要修改控制参数;在一些应用场合,需要纪录过程数据或环境数据,并且在失电的情况下保存。Flash 存储器就向用户提供了灵活的程序、数据修改机会,还提供了失电保存程序和数据的可靠性。

为了保护用户的权益,一般希望能够对用户的程序加以保密。保密有两重重要意义,一个意义是防止程序被修改而破坏原有系统的正常工作;另一个意义是防止程序被读出模仿,而令有关知识产权被泄露或被盗。

对于开发应用,则希望有时在外部增加有关的存储单元,用于存放更多的工作信息。

本章介绍存储器保密方法、编程方法和扩展方法。

6.1 签名字节的读出及存储器加密

89 系列单片机在内部存有特定的所谓“签名字节”,用于说明单片机的生产厂家、型号和编程电压。把签名字节读出来,可以使用户对所用的单片机有一个特别的了解。另外,为了对程序存储器内容保密,还要用户对程序存储器执行保密的处理过程。这一节中,先介绍签名字节的读出方法,然后介绍存储器保密的处理方法。

6.1.1 签名字节的读出

签名字节(Signature Bytes)是生产厂家在生产 89 系列单片机时写入到存储器的信息。它用于表明生产厂家、编程电压和型号。

一般情况下,签名字节的有关信息也会以特定的形式印在单片机封装外壳的顶面。如果由于运输或其他原因,所印的信息被破坏或擦掉,则应通过对签名字节的读出来识别单片机。

1. AT89C51/52 和 AT89LV51/52 的签名字节

签名字节有 3 个字节,分别存储在 Flash 存储器的地址 030H,031H,032H 中。它们的含义分别如下。

030H:1EH,表示生产厂家为 ATMEL。

031H:51H,表示单片机为 AT89C51。

52H,表示单片机为 AT89C52。

61H,表示单片机为 AT89LV51。

62H,表示单片机为 AT89LV52。

032H:05H,表示编程电压为 5 V。

FFH,表示编程电压为 12 V。

在单片机出厂时,会在 89 系列单片机封装外壳的顶面印上和签名字节对应的有关标志。例如单片机 AT89C52 采用 12 V 编程,则其顶面所印的标志如下:

AT89C52

XXXX

YYWW

其签名字节在 Flash 存储器中的地址内容为:

030H:1EH

031H:52H

032H:FFH

当单片机 AT89C52 采用 5 V 编程时,则其顶面所印的标志如下:

AT89C52

XXXX-5

YYWW

其签名字节在 Flash 存储器中的地址内容为:

030H:1EH

031H:52H

032H:05H

对于 AT89LV51 和 AT89LV52 单片机,它们的签名字节对应的封装顶面印刷标志和 AT89C51 的标志方法类同。

2. AT89C1051 和 AT89C2051 的签名字节

AT89C1051 和 AT89C2051 的签名字节在 Flash 存储器中的地址为 000H,001H。它们的含义分别如下。

000H:1EH,表示生产厂家为 ATMEL。

001H:11H,表示单片机为 AT89C1051。

21H,表示单片机为 AT89C2051。

002H:FFH,表示编程电压为 12 V。

从上面的地址可以看出,AT89C51/52 和 AT89LV51/52 的签名字节地址处于 Flash 存储器的 030H~032H;而 AT89C1051/1052 的签名字节地址处于 Flash 存储器的 000H~002H。它们的地址不同,单片机的识别码也不同。

3. AT89C51/52 和 AT89LV51/52 签名字节的读出

签名字节的读出对 AT89C51/52 和 AT89LV51/52 这 4 种型号的单片机来说,过程是一样的。这个读出过程有 7 个步骤。

- (1) 在 XTAL1、XTAL2 端连接石英晶体振荡器。
- (2) 在 V_{CC} 加上 +5 V 电源,GND 接地。
- (3) 把高电平接到 RST,ALE/ $\overline{PROG}$, $\overline{EA}$ / V_{PP} 端。
- (4) 把低电平接到 $\overline{PSEN}$,P2.6,P3.6,P3.7 端。
- (5) 把签名字节地址的低 8 位送到 P1 端口,高 4 位送到 P2.3~P2.0 端口。
- (6) 在 P2.7 加上一个负脉冲,同时在 P0 端口读取所读出的签名字节内容。

(7) 修改 P2.3~P2.0 和 P1 端口所输入的地址,返回第(6)点执行,直到所有签名字节读出为止。

签名字节读出时的电路逻辑结构及有关电平及波形分别如图 6-1 和图 6-2 所示。

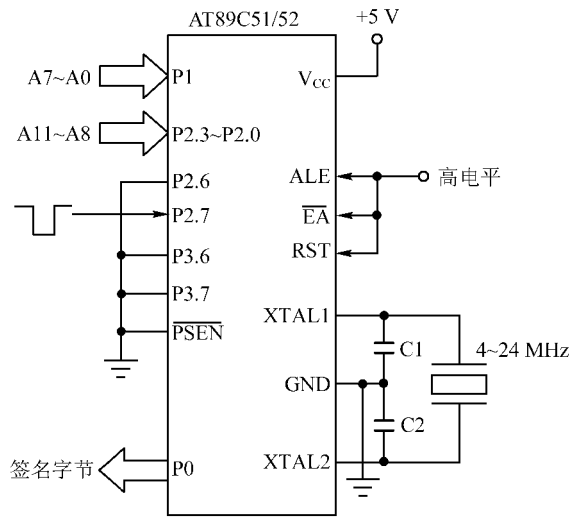


图 6-1 AT89C51/52 和 AT89LV51/52
签名字节读出的逻辑电路

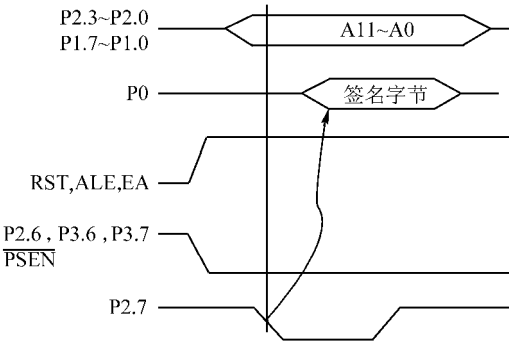


图 6-2 AT89C51/52 和 AT89LV51/52
签名字节读出的时序图

还须指出的是:签名字节能够被读出的先决条件是程序存储器尚未被封锁,即存储器的封锁位 LB1, LB2 未被编程。也就是说 LB1, LB2 都为“1”。这是因为 Flash 存储器在擦除之后所有单元内容为 FFH。另外, LB1, LB2 不能读。

4. AT89C1051 和 AT89C2051 签名字节的读出

AT89C1051 和 AT89C2051 签名字节的读出过程和控制与 AT89C51 的类同。当然,只有在程序存储器未被封锁的条件下才能读出签名字节。因为当程序存储器被封锁时,它的任何内容都是无法读出的。

AT89C1051/2051 签名字节的读出过程如下。

- (1) 在 V_{CC} 加 +5 V 电源, GND 接地。
- (2) 在 RST, P3.2 端加上高电平。
- (3) 在 P3.3, P3.5, P3.7 端加上低电平。
- (4) 在 P3.4 端加上一个负脉冲。
- (5) 在 P1 端口读取所读出的签名字节。
- (6) 在 XTAL1 端加上一个正脉冲,使地址加 1。
- (7) 返回(4)执行直到签名字节读完为止。

AT89C1051/2051 读出签名字节的电路逻辑结构及有关时序分别如图 6-3、图 6-4 所示。

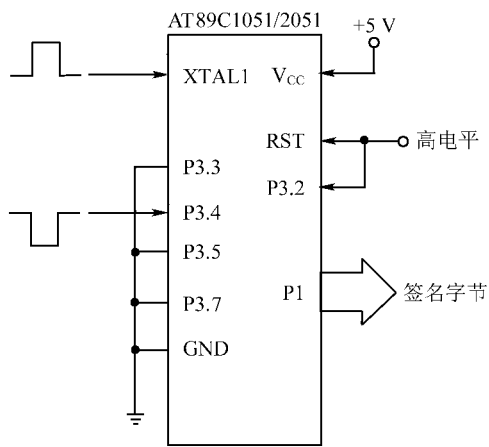


图 6-3 AT89C1051/2051 签名字节读出的逻辑电路

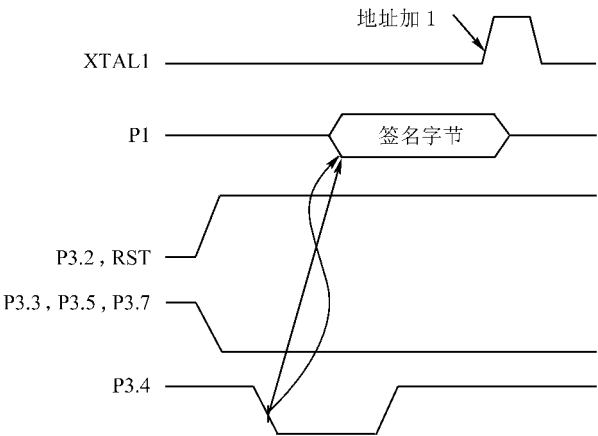


图 6-4 AT89C1051/2051 签名字节读出的时序图

6.1.2 存储器加密

目前,单片机的应用越来越广泛,在许多应用场合下,需要对程序存储器的内容加密,以防止程序被非法剽窃。而 89 系列单片机则具有此功能,即对程序存储器封锁,阻止非法读出受保护的软件。

对存储器中封锁位的处理方法可参阅有关的数据手册。在表 6-1 中,列出了程序封锁位及其功能。存储器中的封锁位只能和 Flash 存储器一起擦除,擦除后,即恢复了器件的全部功能。

在表 6-1 中,程序存储器的加密是通过对封锁位 LB(Lock Bit)编程实现的。对存储器执行封锁功能的封锁位 LB 有 3 位,分别称 LB1, LB2, LB3。由于封锁位被编程之后,Flash 存储器的内容不能用校验方法进行读出,所以实现了对程序内容的保密防盗,从而实现对存储器的加密工作。

表 6-1 程序封锁位及其功能

程序封锁位				功 能
方 式	LB1	LB2	LB3	
1	U	U	U	不设程序加密
2	P	U	U	禁止从外部程序存储器中执行 MOVC 指令读取内部程序存储器,EA 被采样并在复位时锁存,禁止对 Flash 存储器进一步编程
3	P	P	U	与方式 2 相同,同时禁止校验内部存储器
4	P	P	P	与方式 3 相同,同时还禁止外部的执行

注:P=被编程;U=未编程;其他组合未被定义。

当加密位 1 被编程时,EA 脚上的电平被采样并在复位时锁存,如果芯片在封锁位被编程

后一直未有复位操作,则锁存器的值为随机数,直到复位起作用。被锁存的 $\overline{EA}$ 值必须与当前 $\overline{EA}$ 引脚上的逻辑状态一致,以便保持单片机正确的工作功能。

89 系列单片机各种型号中的封锁位配置不同,有的有 3 位,有的只有 2 位。配置情况如表 6-2 所列。

表 6-2 程序封锁位配置表

型 号	封锁位	型 号	封锁位
AT89C51	LB1, LB2, LB3	AT89C2051	LB1, LB2
AT89C52	LB1, LB2, LB3	AT89C1051	LB1, LB2

1. AT89C51/52 和 AT89LV51/52 的存储器加密

对 AT89C51/52 和 AT89LV51/52 的存储器加密,需要对 3 位封锁位 LB1, LB2, LB3 进行编程。编程是一位一位执行的。由于 LB1, LB2, LB3 的功能有一定的关系,如表 6-1 所列,所以,编程可以选择 3 种形式,即编程 1 位、2 位或 3 位。编程时应按照 LB1→LB2→LB3 的顺序执行。每位编程时的控制信号不同。

(1) 对 LB1 编程

- ① 对 RST, P2.6, P2.7, P3.6, P3.7 接高电平。
- ② 对 $\overline{PSEN}$ 接低电平。
- ③ 对 $\overline{EA}/V_{PP}$ 加上编程电压,即 12 V 或 5 V。该电压由签名字节或封装顶面标志给出。
- ④ 在 ALE/ $\overline{PROG}$ 加上宽度大于 1 μs 、小于 110 μs 的编程负脉冲。

(2) 对 LB2 编程

- ① 对 RST, P2.6, P2.7 加上高电平。
- ② 对 $\overline{PSEN}$, P3.6, P3.7 接低电平。
- ③ 对 $\overline{EA}/V_{PP}$ 加上 12 V 或 5 V 编程电压。

该电压由签名字节或封装标志指明。

- ④ 在 ALE/ $\overline{PROG}$ 加上宽度大于 1 μs 、小于 110 μs 的编程负脉冲。

(3) 对 LB3 编程

- ① 对 RST, P2.6, P3.6 接高电平。
- ② 对 $\overline{PSEN}$, P2.7, P3.7 接低电平。
- ③ 对 $\overline{EA}/V_{PP}$ 加上编程电压,即 12 V 或 5 V。该电压由签名字节或封装标志指出。
- ④ 在 ALE/ $\overline{PROG}$ 加上宽度大于 1 μs 、小于 110 μs 的编程负脉冲。

AT89C51/52, AT89LV51/52 的程序存储器加密控制逻辑电路如图 6-5 所示。LB1~LB2 的编程有关信号及电平如表 6-3 所列。从表中可以看出封锁位 LB 的每位编程所用的信号不同。

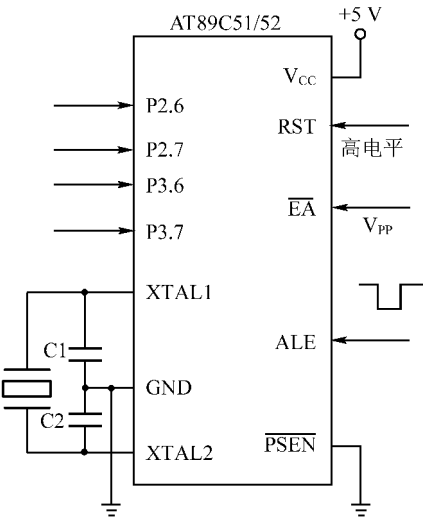


图 6-5 封锁位 LB 的编程逻辑

表 6-3 LB2~LB3 编程信号电平

封锁位	RST	$\overline{\text{PSEN}}$	$\text{ALE}/\overline{\text{PROG}}$	$\overline{\text{EA}}/\text{V}_{\text{PP}}$	P2. 6	P2. 7	P3. 6	P3. 7
LB1	H	L		H/12 V	H	H	H	H
LB2	H	L		H/12 V	H	H	L	L
LB3	H	L		H/12 V	H	L	H	L

2. AT89C1051/2051 的存储器加密

AT89C1051/2051 的程序存储器加密只需对 2 位封锁位 LB1, LB2 编程。编程按 LB1→LB2 顺序进行。编程形式可以对 LB1 一位,也可以对 LB1, LB2 两位进行。编程了 1 位或 2 位之后的保密功能如表 6-1 所示。

AT89C1051/2051 的封锁位编程和 AT89C51 所用逻辑电路结构及信号不尽相同。下面对 LB1 和 LB2 的编程分别加以说明。

(1) 对 LB1 编程

- ① 对 P3. 3, P3. 4, P3. 5, P3. 7 加上高电平。
- ② 对 RST 加上 12 V 编程电压。
- ③ 在 P3. 2/ $\overline{\text{PROG}}$ 端加上大 1 μs 、小于 110 μs 的编程负脉冲。

(2) 对 LB2 编程

- ① 对 P3. 3, P3. 4 接高电平。
- ② 对 P3. 5, P3. 7 接低电平。
- ③ 在 RST 加上 12 V 编程电压。
- ④ 在 P3. 2/ $\overline{\text{PROG}}$ 端加上大于 1 μs 、小于 110 μs 的编程负脉冲。

AT89C1051/2051 程序存储器加密的逻辑电路如图 6-6 所示。很明显,它比 AT89C51 加密的逻辑结构简单。LB1 和 LB2 编程时所需的 AT89C1051/2051 的各部分信号如表 6-4 所列。

表 6-4 封锁位编程的信号电平

封锁位	RST	P3. 2/ $\overline{\text{PROG}}$	P3. 3	P3. 4	P3. 5	P3. 7
LB1	12 V		H	H	H	H
LB2	12 V		H	H	L	L

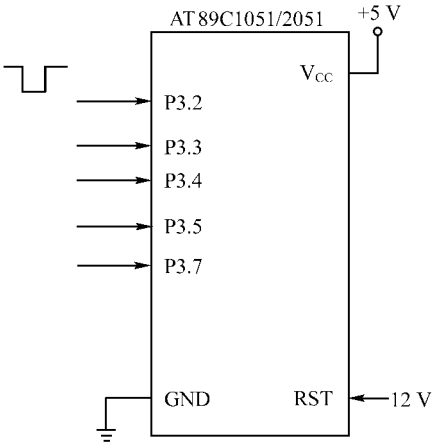


图 6-6 AT89C1051/2051 的封锁位编程逻辑

6.2 Flash 存储器的编程方法

Flash 存储器允许用户进行擦除和编程,是单片机应用开发中十分方便的软件修改器件。89 系列单片机内部的程序存储器是可以进行多次擦除和编程的 Flash 存储器。出厂时,89 系列单片机内部的 Flash 存储器已处于擦除状态,除了签名字节之外的存储单元的内容为

FFH。在编程之后如果需要再修改程序,则要对 Flash 执行擦除工作;在擦除之后,所有存储单元包括签名字节全部被擦除为 FFH。

AT89C51,AT89LV51 的 Flash 存储器容量为 4 KB,AT89C52,AT89LV52 的 Flash 存储器容量为 8 KB。它们执行的编程过程是一样的,只是送到单片机的地址量和存储器容量相对应,故 51 型单片机的地址量为 4 KB,52 型单片机的地址量为 8 KB。

AT89C1051 的 Flash 存储器容量为 1 KB,AT89C2051 的 Flash 存储器容量为 2 KB。它们的编程过程相同,只是地址量有区别。

另外,编程电压视单片机的标志有 5 V,12 V 两种。

6.2.1 AT89C51 Flash 存储器编程

AT89C51 单片机内部有一个 4 KB 的 Flash EPROM。这个 Flash 存储阵列通常是处于已擦除状态(即存储单元的内容为 FFH),随时可对它进行编程。编程接口可接收高电压(12 V)或低电压(V_{cc})的允许编程信号。低电压编程方式可很方便地对 AT89C51/LV51 内的用户程序进行编程,而高电压编程方式则可与通用的 EPROM 编程器兼容。

AT89C51 单片机中,有些允许用高电压编程方式编程,有些允许用低电压编程方式编程,各自芯片顶面的型号和片内签名字节的内容不同,则其编程电压有别,详见表 6-5。

AT89C51 的程序存储器阵列是采用字节写入方式编程的,即每次写入一个字节。要对片内的 Flash 程序存储器写入任何一个非空字节,都必须用片擦除方式将整个存储器的内容清除。

表 6-5 芯片型号和片内签名字节的内容

	$V_{pp}=12\text{ V}$	$V_{pp}=5\text{ V}$
型号及标志	AT89C51/LV51 ××××/×××× YYWW/YYWW	AT89C51/LV51 ××××-5/××××-5 YYWW/YYWW
签名字节	(030H)=1EH (031H)=51H/61H (032H)=FFH	(030H)=1EH (031H)=51H/61H (032H)=05H

1. 对 Flash 存储器编程

编程前,必须按照表 6-6 和图 6-7 建立好地址、数据和相应的控制信号。编程单元的地址加在 P1 端口和 P2 端口的 P2.0~P2.3(11 位地址为 0000H~0FFH),数据从 P0 端口输入。引脚 P2.6,P2.7 和 P3.6,P3.7 的电平选择见表 6-6。 $\overline{PSEN}$ 应保持低电平,而 RST 应保持高电平。 $\overline{EA}/V_{pp}$ 是编程电源的输入端,按要求加入编程电压。 $\overline{ALE}/\overline{PROG}$ 端输入编程脉冲(应为负脉冲信号)。编程时,采用 4~20 MHz 的振荡器。对 AT89C51/LV51 编程的步骤如下。

- (1) 在地址线上输入要编程单元的地址。
- (2) 在数据线上输入要写入的数据字节。
- (3) 激活相应的控制信号。

- (4) 在采用高电压编程方式时,将 $\overline{EA}/V_{PP}$ 端的电压加到 12 V。
- (5) 每对 Flash 存储器写入一个字节或每写入一个程序封锁位,加一个 $ALE/\overline{PROG}$ 编程负脉冲。
- 改变编程单元的地址和要写入的数据,重复步骤(1)~(5),直到全部文件编程完毕。
- 每个字节写入周期是自动定时的,通常不大于 1.5 ms。

表 6-6 Flash 编程方式

方 式	RST	$\overline{PSEN}$	$ALE/\overline{PROG}$	$\overline{EA}/V_{PP}$	P2. 6	P2. 7	P3. 6	P3. 7
写代码数据	H	L		H/12 V ⁽¹⁾	L	H	H	H
读代码数据	H	L	H	H	L	L	H	H
写封锁位	LB1	H	L		H/12 V	H	H	H
	LB2	H	L		H/12 V	H	L	L
	LB3	H	L		H/12 V	H	L	L
片擦除	H	L	 ⁽²⁾	H/12 V	H	L	L	L
读签名字节	H	L	H	H	L	L	L	L

- (1) 根据签名字节(地址为 032H)的内容选择合适的编程电压(V_{PP} =12 V 或 V_{PP} =5 V)；
- (2) 片擦除操作时要求 $\overline{PROG}$ 的脉冲宽度为 10 ms。

2. 数据查询方式

AT89C51 单片机用数据查询方式来检测一个写周期是否结束。在一个写周期期间,如果想读出最后写入的那个字,则读出数据的最高位(P0. 7)是原来写入字节最高位的反码。写周期一旦完成后,有效的数据就会出现在所有输出端上,这时可开始下一个写周期。一个写周期开始后,可在任何时间开始进行数据查询。

3. 准备就绪/忙信号

字节编程的过程也可通过 RDY/ $\overline{BSY}$ 输出信号来监视。在编程期间,当 ALE 变为高电平后,P3. 4(RDY/ $\overline{BSY}$)端的电平被拉低,表示忙(正在编程)状态。编程完毕后,P3. 4 的电平变高,表示准备就绪状态。

4. 程序的校验

如果封锁位 LB1 和 LB2 没有被编程,那么就可以对 AT89C51 内部已编好的程序进行校验。这时采用图 6-8 所示的引脚接法。程序存储器的地址仍由 P1 端口和 P2 端口的 P2. 0~P2. 3 输入,数据由 P0 端口输出。P2. 6,P2. 7 和 P3. 6,P3. 7 的电平如表 6-6 所列。 $\overline{PSEN}$ 保

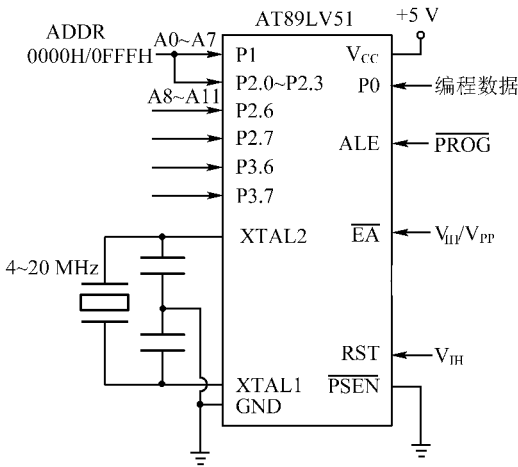


图 6-7 Flash 编程

持低电平,而 ALE,EA和 RST 保持高电平。校验时,在 P0 端口上要求外接约 10 kΩ 左右的上拉电阻。

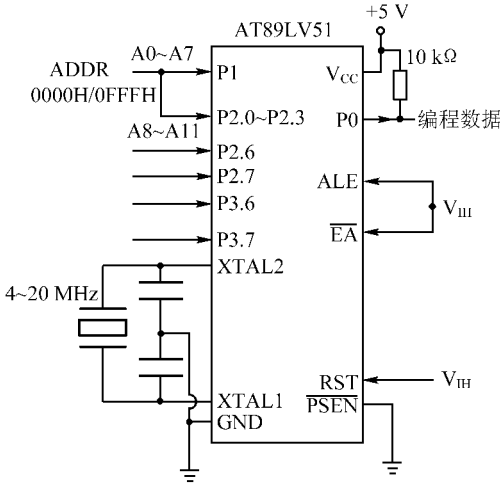


图 6-8 程序的校验

校验有关波形技术指标如表 6-7 所列。

AT89C52/LV52 单片机 Flash 存储器编程方法和 AT89C51/LV51 类同,只是在编程时,由于 AT89C52/LV52 的 Flash 存储器的容量有 8 KB,比 AT89C52/LV52 的容量大一倍,所以,向 AT89C52/LV52 的 P1 端口和 P2.0~P2.3 端口发送的地址也大一倍,为 0000H~1FFFH,其他控制信号和编程的逻辑电路是相同的。

程序封锁位不能直接校验。封锁位的校验可通过观察它们的功能是否被实现来进行。

Flash 存储器编程和程序检验时的时序图如图 6-9(高电压编程)和图 6-10(低电压编程)所示。

5. 擦除操作

用一组适当的控制信号(如表 6-6 所列),并保持 ALE/PROG 脉冲宽度(低电平的时间)约 10 ms,整个 Flash 存储器的内容就会被擦除。擦除后,Flash 存储器全为“1”状态。注意,在对程序存储器进行重新编程之前必须执行片擦除操作。

AT89C51 内部 Flash 存储器的编程和

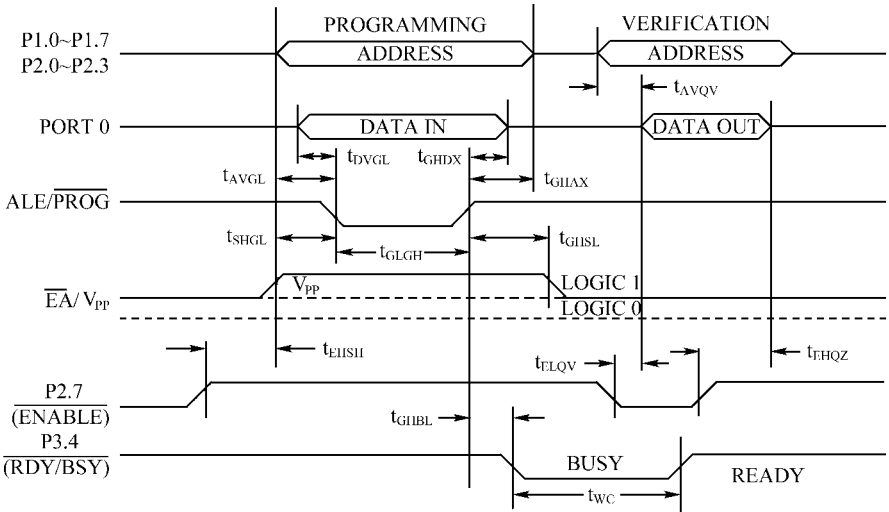


图 6-9 Flash 编程和校验的波形(高电压编程方式)

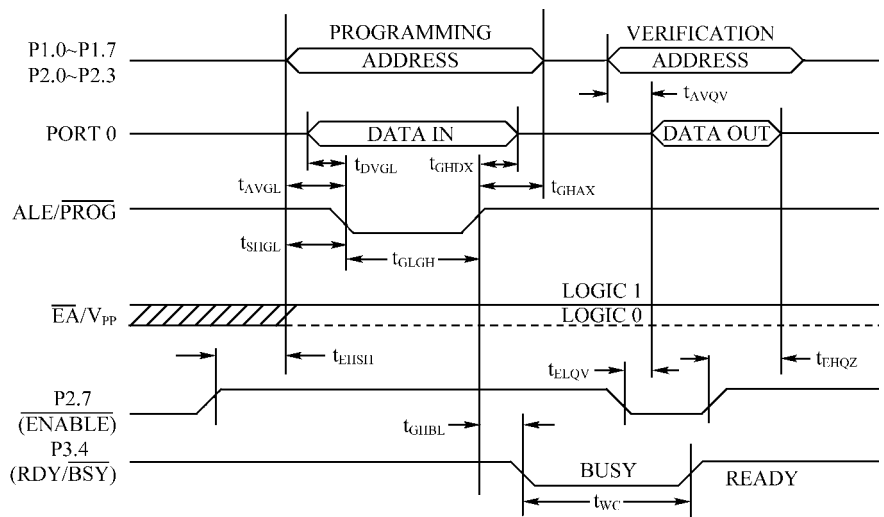


图 6-10 Flash 编程和校验的波形 (低电压编程方式)

表 6-7 Flash 编程和校验的技术指标

TA=21~27℃, VCC=5.0±10%

符 号	参 数	Min	Max	单 位
V _{PP} ⁽¹⁾	编程电压	11.5	12.5	V
I _{PP} ⁽¹⁾	编程电流		1.0(25) ⁽²⁾	mA(μA) ⁽²⁾
1/t _{CLCL}	振荡器频率	4	24(12) ⁽²⁾	MHz
t _{AVGL}	建立地址到 $\overline{\text{PROG}}$ 变低	48 t _{CLCL}		
t _{GHAX}	$\overline{\text{PROG}}$ 变低后地址保持不变	48 t _{CLCL}		
t _{DVGL}	建立数据到 $\overline{\text{PROG}}$ 变低	48 t _{CLCL}		
t _{GHDX}	$\overline{\text{PROG}}$ 变低后数据保持不变	48 t _{CLCL}		
t _{EHSH}	$\overline{\text{ENABLE}}$ 变高到加 V _{PP}	48 t _{CLCL}		
t _{SHGL}	加 V _{PP} 到 $\overline{\text{PROG}}$ 变低	10		μs
t _{GHSL} ⁽¹⁾	$\overline{\text{PROG}}$ 后保持 V _{PP}	10		μs
t _{GLGH}	$\overline{\text{PROG}}$ 宽度	1	110	μs
t _{AVQV}	地址到数据有效		48 t _{CLCL}	
t _{ELQV}	$\overline{\text{ENABLE}}$ 低到数据有效		48 t _{CLCL}	
t _{EHQV}	$\overline{\text{ENABLE}}$ 后数据悬浮	0	48 t _{CLCL}	
t _{GHBL}	$\overline{\text{PROG}}$ 变高到 $\overline{\text{BUSY}}$ 变低		1.0	μs
t _{WC}	字节写入周期时间		2.0	ms

(1) 只用于 12 V 编程方式；

(2) 只对 AT89LV51 而言。

6.2.2 AT89C1051 Flash 存储器编程

AT89C1051 是在擦除状态(即单元内容为 FFH)下用 1 KB 的片内 Flash 存储器进行封装的,并且存储器是可编程的。代码存储器一次编程一个字节。一旦存储器被编程,如果重新编程任一非空字节,则整个存储器需进行电擦除。AT89C1051 内 Flash 存储器的编程和校验结构如图 6-11、图 6-12 所示,AT89C1051 编程和校验时序如图 6-13 所示。

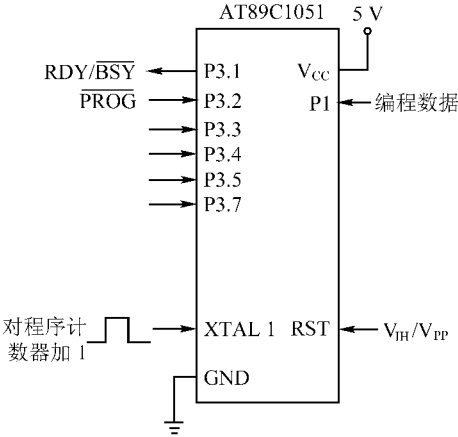


图 6-11 Flash 存储器的编程结构

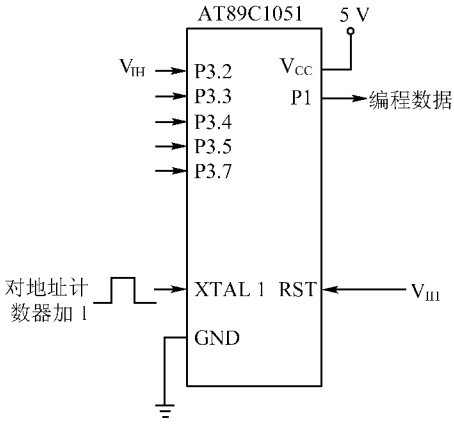


图 6-12 Flash 存储器的校验结构

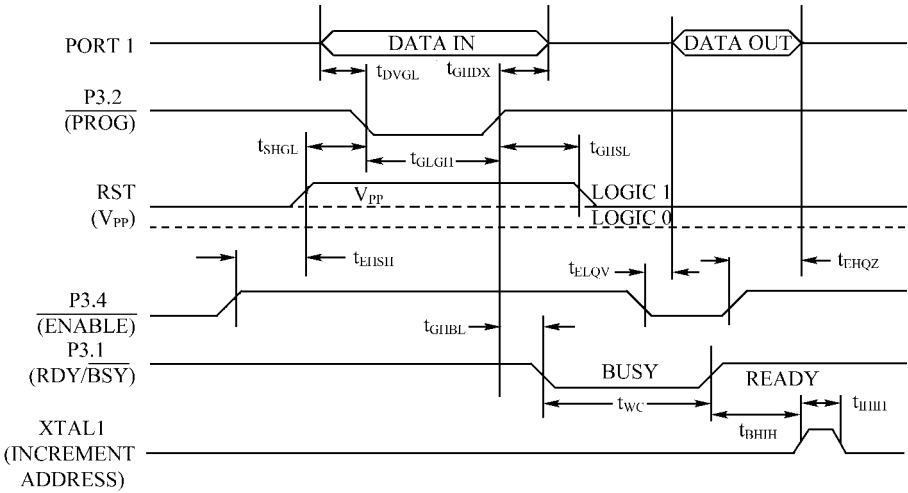


图 6-13 AT89C1051 编程和校验时序

(1) 内部地址计数器:AT89C1051 包含一内部 Flash 存储器地址计数器,它总是在 RST 上升沿复位为 000H,并在 XTAL1 引脚有一正跳变脉冲时进行计数。

(2) 编程算法:要编程 AT89C1051,则采用下列步骤。

① 上电次序:在 V_{CC} 和 GND 引脚之间加上电源。设置 RST 和 XTAL1 为 GND。所有其他引脚浮空,至少等待 10 ms。

② 设置 RST 引脚为高“H”,设置 P3.2 引脚为高“H”。

③ 对引脚 P3.3, P3.4, P3.5, P3.7 加上逻辑电平高“H”或低“L”的正确组合,来选择 Flash 存储器编程方式表 6-8 所示的一种编程操作。

编程和校验存储器由下列④~⑩点实现。

④ 在 000H 单元对 P1.0~P1.7 加入数据代码字节。

⑤ 升高 RST 到 12 V 来激活编程。

⑥ 使 P3.2 跳变一次来编程 Flash 存储器中的一字节。写字节周期是自身定时的,一般需用 1.2 ms。

⑦ 校验已被编程的数据,使 RST 从 12 V 降到逻辑电平“H”,并设置 P3.3~P3.7 引脚到正确的电平。可在 P1 口引脚读输出数据。

⑧ 对下一地址单元编程,在 XTAL1 端加正脉冲一次对内部地址计数器加 1。在 P1 口引脚加入新数据。

⑨ 重复步骤⑤~⑧,对整个 1 KB 存储器改变数据并对地址计数器加 1 或者一直到目标文件的结束。

⑩ 断电次序:设置 XTAL1 为“L”,设置 RST 为“L”,浮空所有其他 I/O 引脚,关闭 V_{CC} 电源。

(3) 数据查询:AT89C1051 具有指示写周期结束的数据查询功能。在写周期期间,如果读最后写主字节将令 P1.7 出现写入数据的反码。一旦写周期完成,则全部输出端的真实数据有效,同时下一个周期开始。数据查询可在写周期开始后的任一时刻开始。

(4) RDY/ $\overline{BSY}$:字节编程的进度还可由“RDY/ $\overline{BSY}$ ”输出信号监控。编程期间,P3.1 引脚在 P3.2 变高后被拉低来指示处于“忙”,即“ $\overline{BSY}$ ”。P3.1 在编程结束时被再次拉高来指示处于“就绪”,即“RDY”。

(5) 程序校验:如果封锁位 LB1 和 LB2 没进行编程,则代码数据可通过数据线读出。

① 使 RST 从“L”变到“H”,复位内部地址计数器为 000H。

② 为读代码数据加入正确的控制信号并在 P1 口引脚读输出数据。

③ 在 XTAL1 引脚加上一个正脉冲来对内部地址计数器加 1。

④ 在 P1 口引脚读下一个代码数据字节。

⑤ 重复步骤③和④直到读完整个存储器。

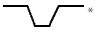
封锁位不可直接进行校验。封锁位的校验可通过对存储器的校验和写入状态来得到。

(6) 片擦除:利用控制信号的正确组合并保持 P3.2 引脚 10 ms 的低电平就可擦除整个 Flash 存储器(1 KB)和两个封锁位。Flash 存储器在片擦除操作中写入全“1”,并必须在任何非空存储器字节被再编程之前执行。

AT89C1051 的编程方式及有关控制信号的电平波形如表 6-8 所列。从表中看出:其编程所需的控制信号比 AT89C51 的要少。

AT89C2051 的 Flash 存储器编程过程和 AT89C1051 相同,所差的是 AT89C2051 有 2 KB 存储器容量,所以在 XTAL1 端所加的正脉冲数量要比 AT89C1051 多一倍。

表 6-8 AT89C1051 Flash 编程方式

方 式		RST	P3. 2/ $\overline{\text{PROG}}$	P3. 3	P3. 4	P3. 5	P3. 7
写代码数据		12 V		L	H	H	H
读代码数据		H	H	L	L	H	H
写封锁位	LB1	12 V		H	H	H	H
	LB2	12 V		H	H	L	L
片擦除		12 V		H	L	L	L
读签名字节		H	H	L	L	L	L

注：
1 内部 EEPROM 地址计数器在 RST 的上升沿复位到 000H 并由 XTAL1 引脚正脉冲执行计数。
2 编程期间 P3. 1 被拉低来指示 RDY/ $\overline{\text{BSY}}$ 。
* 片擦除需要 10 ms 的 PROG 脉冲。

6.3 串行 EEPROM 接口方法

在新一代单片机中,无论总线型还是非总线型单片机,为了简化系统结构,提高系统的可靠性,都推出了芯片间的串行数据传输技术,设置了芯片间的串行传输接口或串行总线。串行总线扩展接线灵活,极易形成用户的模块化结构,同时将大大简化其系统结构。串行器件不仅占用很少的资源 and I/O 线,而且本积大大缩小,同时还具有工作电压宽、抗干扰能力强、功耗低、数据不易丢失和支持在线编程等特点。目前,各式各样的串行接口器件层出不穷,如:串行 EEPROM、串行 ADC/DAC、串行时钟芯片、串行数字电位器、串行微处理器监控芯片、串行温度传感器等等。

串行 EEPROM 是在各种串行器件应用中使用较频繁的器件,和并行 EEPROM 相比,串行 EEPROM 的数据传送速度较低,但是其体积较小,容量小,所含的引脚也较少。所以,它特别适合于需要存放非挥发数据、要求速度不高、引脚少的单片机的应用。在这一节,介绍串行 EEPROM 芯片,以及它们和 89 系列单片机的接口技术。

6.3.1 串行 EEPROM 及其工作原理

串行 EEPROM 中,较为典型的有 ATMEL 公司的 AT24CXX 系列以及该公司生产的 AT93CXX 系列,较为著名的半导体厂家,包括 Microchip、国家半导体公司等,都有 AT93CXX 系列 EEPROM 产品。

1. AT24CXX 系列 EEPROM

AT24CXX 系列的串行电可改写及可编程只读存储器 EEPROM 有 10 种型号,其中典型的型号有 AT24C01A/02/04/08/16 等 5 种,它们的存储容量分别是 1 024/2 048/4 096/8 192/16 384 位,也就是 128/256/512/1 024/2 048 字节。这个系列一般用于低电压、低功耗的工业和商业用途,并且可以组成优化的系统。这个系列还有多种电压级别,包括 5 V(4.5~5.5 V),2.7 V(2.7~5.5 V),2.5 V(2.5~5.5 V),1.8 V(1.8~5.5 V)等 4 种电压级别。它们的封装有 8 引脚 PDIP 方式、8 引脚和 16 引脚 SOIC 方式。信息存取采用 2 线串行接口。

(1) 结构原理及引脚

AT24CXX 系列的 5 种典型型号 AT24C01A/02/04/08/16 的结构原理类同,只是存储单元的数量不同,即存储量不同。它们的结构框图如图 6-14 所示,封装的方式和引脚情况如图 6-15 所示。

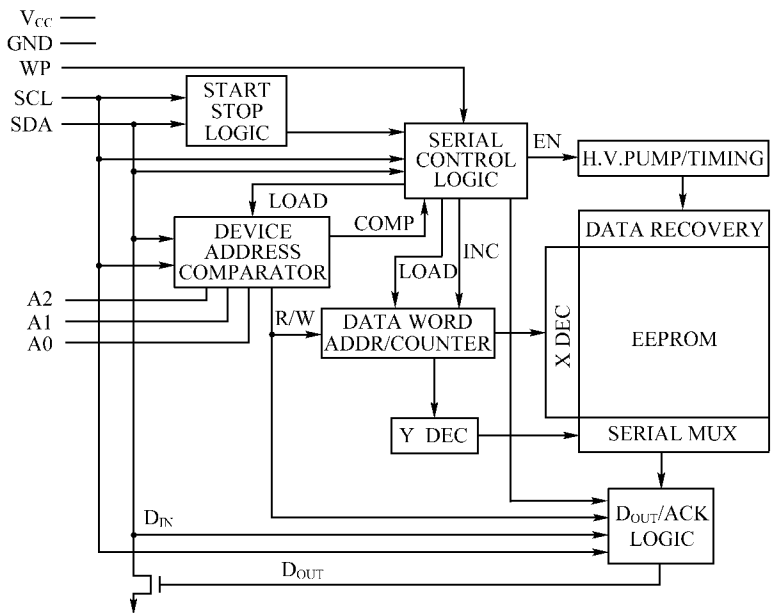


图 6-14 AT24C01A/02/04/08/16 结构框图

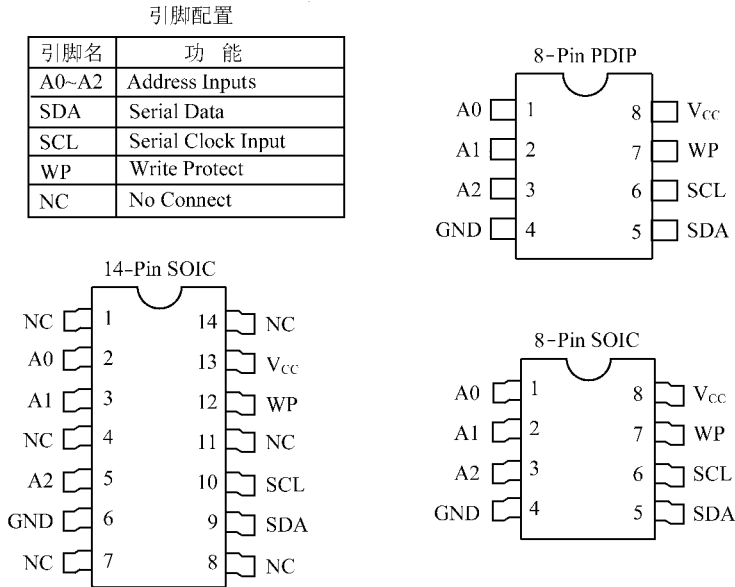


图 6-15 AT24C01A/02/04/08/16 的封装及引脚

从图 6-14 中看出:AT24C01A 等的结构由开停逻辑、串行控制逻辑、地址比较器、地址计数器、数据输出回答逻辑、泵和定时电路、存储阵列等组成。在这些逻辑的共同作用下,EEPROM 执行外部信号所确定的读写操作。

从图 6-15 中看出:AT24C01A 等型号有地址线 A0~A2,串行数据引脚 SDA,串行时钟输入引脚 SCL,写保护引脚 WP 等引脚。很明显,其引脚较少,对组成的应用系统可以减少布线,提高可靠性。

各引脚的功能和意义如下。

① V_{cc} 引脚,电源+5 V。

② GND 引脚,地线。

③ SCL 引脚,串行时钟输入端。在时钟的正跳沿即上升沿时把数据写入 EEPROM;在时钟的负跳沿即下降沿时把数据从 EEPROM 中读出来。

④ SDA 引脚,串行数据 I/O 端,用于输入和输出串行数据。这个引脚是漏极开路的端口,故可以组成“线或”结构。

⑤ A0,A1,A2 引脚,是芯片地址引脚。在型号不同时意义有些不同,但都要接固定电平。

对于 AT24C01A 和 AT24C02,A0,A1,A2 这 3 位地址可以用于芯片寻址。当用 8 片 AT24C01A 组成 1 KB 的存储器时,则第 1 片地址为“000”,故 A0,A1,A2 全部接地;第 2 片地址为“001”,故 A2 接高电平 5 V,A0,A1 接地电平;……第 8 片地址为“111”,故 A0,A1,A2 全部接高电平 5 V。在用 AT24C02 组成 2 KB 存储器时,也同样处理。

对于 AT24C04,地址 A0 引脚不接任何电平,这时只用 A1,A2 两位地址,故最多只能用 4 片 AT24C04 来构成 2 KB 的存储器。

对于 AT24C08,地址 A0,A1 引脚不接任何电平,这时只用 A2 这 1 位地址。故最多只能用 2 片 AT24C08 来构成 2 KB 的存储器。

对于 AT24C16,地址 A0,A1,A2 引脚全部不接任何电平。这时,只能用 1 片 AT24C16 来构成 2 KB 的存储器。

⑥ WP 引脚,写保护端。这个端提供了硬件数据保护。当把 WP 接地时,允许芯片执行一般读写操作;当把 WP 接到 V_{cc} 时,则对芯片实施写保护。WP 端的功能如表 6-9 所列。

⑦ NC 引脚,无用引脚,不接任何电平。

表 6-9 WP 端的功能和意义

WP 状态	AT24C01A	AT24C02	AT24C04	AT24C08	AT24C16
接 V _{cc}	1 KB 全部保护	2 KB 全部保护	4 KB 全部保护	不保护	8 KB 的高地址 4 KB 保护
接 GND	一般读/写操作				

(2) 存储器的组织及运行

① 存储器的组织:对于不同的型号,存储器的组织不一样,其关键原因在于存储器容量存在差异。对于 AT24CXX 系列的 EEPROM,其典型型号的存储器组织如下。

AT24C01A:内部含有 128 个字节,故需要 7 位地址对其内部字节进行寻址。

AT24C02:内部含有 256 个字节,故需要 8 位地址对其内部字节进行读写。

AT24C04:内部分成 2 页,每页有 256 个字节,故需要 9 位地址对其内部字节进行寻址。
AT24C08:内部分成 4 页,每页有 256 个字节,故需要 10 位地址对其内部字节进行寻址。
AT24C16:内部分成 8 页,每页含 256 个字节,故需要 11 位地址对其内部字节进行寻址。
各种型号 EEPROM 页面大小、容量、总线可接片数及可用地址的情况如表 6-10 所列。

表 6-10 各种型号 EEPROM 参数

型 号	容量/KB	页面数	总线可接片数	可用地址脚
AT24C01A	1	1	8	A0,A1,A2
AT24C02	2	1	8	A0,A1,A2
AT24C04	4	2	4	A1,A2
AT24C08	8	4	2	A2
AT24C16	16	8	1	无

② 运行方式:对于时钟及数据传送,串行数据 I/O 端 SDA 一般需要用外部上拉电阻将其电平拉高。加到 SDA 的数据只有在串行时钟 SCL 对于低电平的时间周期内可以改变。当串行时钟 SCL 处于高电平时,SDA 的数据变化用于指明起始或停止状态。在 SCL 为高电平期间,如果 SDA 从低电平上升到高电平,则表示起始状态;如果 SDA 从高电平下降到低电平,则表示停止状态,如图 6-16、图 6-17 所示。

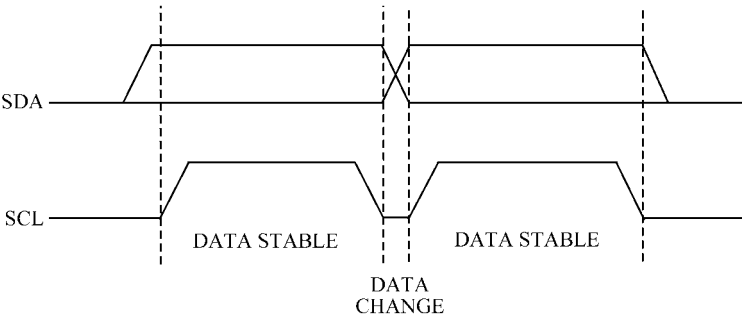


图 6-16 数据变化及稳定的有效时间

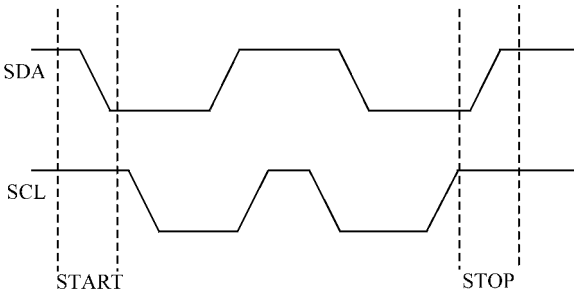


图 6-17 起始及停止状态定义

起始状态:当 SCL 为高电平时,SDA 由高电平变到低电平则处于起始状态。起始状态应处于任何其他命令之前。

停止状态:当 SCL 处于高电平时,SDA 从低电平变成高电平则处于停止状态。在执行完读序列信号之后,停止命令将把 EEPROM 置于低功耗的备用方式(Standby Mode)。

应答信号:应答信号是由接收数据的器件发出的。当 EEPROM 接收完一个写入数据之后,会在 SDA 上发一个“0”应答信号 ACK。反之,当单片机接收完来自 EEPROM 的数据后,单片机也应向 SDA 发 ACK 信号。ACK 信号在第 9 个时钟周期时出现,如图 6-18 所示。

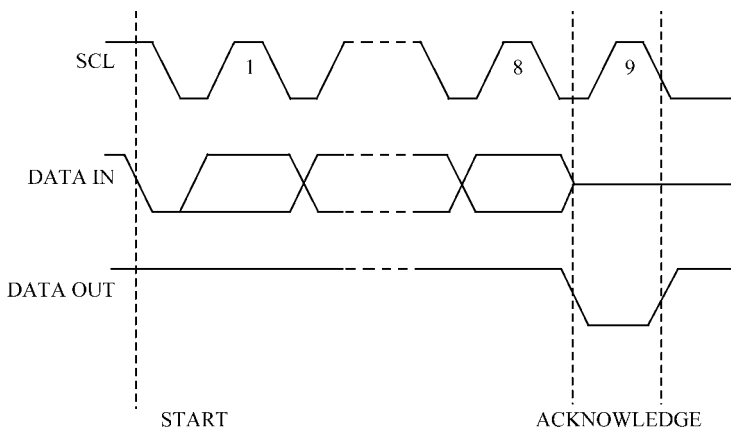


图 6-18 应答信号

备用方式(Standby Mode):AT24C01A/02/04/08/16 都具有备用方式,以保证在没有读写操作时芯片处于低功耗状态。在下面两种情况中,EEPROM 都会进入备用方式:第一,芯片通电的时候;第二,在接到停止位和完成了任何内部操作之后。

(3) 读/写操作过程

AT24C01A 等 5 种典型的 EEPROM 在进入起始状态之后,需要一个 8 位的“器件地址字”去启动存储器进行读或写操作。在写操作中,它们有“字节写”、“页面写”两种不同的写入方法。在读操作中,有“现行地址读”、“随机读”和“顺序读”3 种各具特点的读出方法。下面分别介绍器件寻址、写操作和读操作。

① **器件寻址:**所谓器件寻址(Device Addressing)就是用一个 8 位的器件地址字(Device Address Word)去选择存储器芯片。在逻辑电路中的 AT24CXX 系列的 5 种芯片中,即 AT24C01A/02/04/08/16 中,如果和器件地址字相比较结果一致,则读芯片被选中。下面对器件寻址的过程和意义加以说明。

用于存储器 EEPROM 芯片寻址的器件地址字如图 6-19 所示。它有 4 种形式,分别对应于 1 K/2 K、4 K、8 K 和 16 K 位的 EEPROM 芯片。

从图 6-19 中看出:器件地址字含有 3 个部分。第一部分是高 4 位,它们称为 EEPROM AT24C01A/02/04/08/16 的标识。第二部分称为硬布线地址,它们是标识后的 3 位。第三部分是最低位,它是读/写操作选择位。

第一部分:器件标识,器件地址字的最高 4 位。这 4 位的内容恒为“1010”,用于标识 EEPROM 器件 AT24C01A/02/04/08/16。

第二部分：硬布线地址，是与器件地址字的最高 4 位相接的低 3 位。硬布线地址的 3 位有 2 种符号： $A_i(i=0\sim2)$ 、 $P_j(j=0\sim2)$ ，其中 A_i 表示外部硬布线地址位， P_j 表示内部页面地址位。

对于 AT24C10A/02 这两种 1 K/2 K 位的 EEPROM 芯片，硬布线地址为“A2, A1, A0”。在应用时，“A2, A1, A0”的内容必须和 EEPROM 芯片的 A2, A1, A0 的硬布线情况，即逻辑连接情况相比较，如果一样，则芯片被选中；否则，不选中。

对于 AT24C04 这种 4 K 位的芯片，硬布线地址为“A2, A1, P0”。在应用时，“A2, A1”的内容必须和芯片的 A2, A1 端硬布线情况一致，芯片才能被选中。P0 是芯片内部页面地址，所以对于 AT24C04，它的 A0 不用接任何逻辑布线。

对于 AT24C08 这样含 8 K 位的芯片，硬布线地址为“A2, P1, P0”。在应用时，“A2”的内容必须和芯片 A2 端的硬布线一致，芯片才能被选中。“P1, P0”是芯片内部的页面地址，所以，AT24C08 的 A1, A0 端不接任何逻辑布线。

对于 AT24C16 这个内部含 16 K 位的芯片，不用硬布线地址，故这 3 位用于内部页面寻址。

顺便指出：“器件地址”字是用于选择芯片的，而“字地址”字则是用于对 EEPROM 中的字节寻址的。“字地址”是一个 8 位的地址。当对不同芯片内的字节寻址时，其真正地址是由硬布线地址中的页面地址和字地址结合而成的，其页面地址为真正的高位，字地址为低位。不同芯片的真正字节地址如下：

- AT24C01A/02 :真正地址=字地址
- AT24C04 :真正地址= $P0\times2^8$ +字地址
- AT24C08 :真正地址= $(P1,P0)\times2^8$ +字地址
- AT24C16 :真正地址= $(P2,P1,P0)\times2^8$ +字地址

第三部分：读/写选择位，器件地址字的最低位，并用 R/W 表示。当 R/W=1 时，执行读操作；当 R/W=0 时，执行写操作。

当 EEPROM 芯片被选中时，则输出“0”；如果 EEPROM 芯片没有被选中，则它回到备用方式。被选中的芯片，其以后的输入、输出情况视写入和读出的内容而定。

② 写操作：AT24C01A/02/04/08/16 这 5 种 EEPROM 芯片的写操作有 2 种：一种是字节写，另一种是页面写。

字节写

这种写方式只执行 1 个字节的写入。字节写的过程如图 6-20 所示，其写入过程分外部写和内部写两部分，分别说明如下。

在起始状态中，首先写入 8 位的器件地址，则 EEPROM 芯片会产生一个“0”信号 ACK 输出作为应答；接着，写入 8 位的字地址，在接收了字地址之后，EEPROM 芯片又产生一个“0”应答信号 ACK；随后，写入 8 位数据，在接收了数据之后，芯片又产生一个“0”信号 ACK 作为应答。到此为止，完成了一个字节写过程，故应在 SDA 端产生一个停止状态，这是外部写过程。

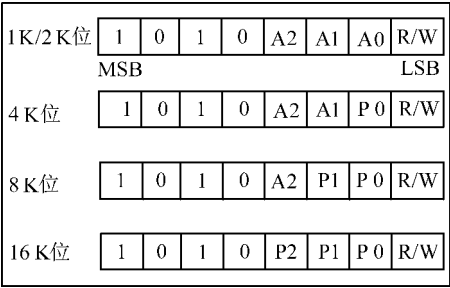


图 6-19 器件地址

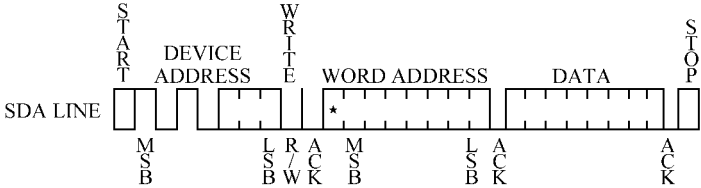


图 6-20 字节写

在这个过程中,控制 EEPROM 的单片机应在 EEPROM 的 SCL,SDA 端送入恰当的信号。当然在一个字节写过程结束时,单片机应以停止状态结束写过程。在这时,EEPROM 进入内部定时的写周期,以便把接收的数据写入到存储单元中。在 EEPROM 的内部写周期中,其所有输入被屏蔽,同时不响应外部信号直到写周期完成。这是内部写过程。内部写过程大约需要 10 ms 时间。内部写过程处于停止状态与下一次起始状态之间,如图 6-21 所示,其中 t_{WR} 是内部写的周期时间, $t_{WR}=10\text{ ms}$ 。

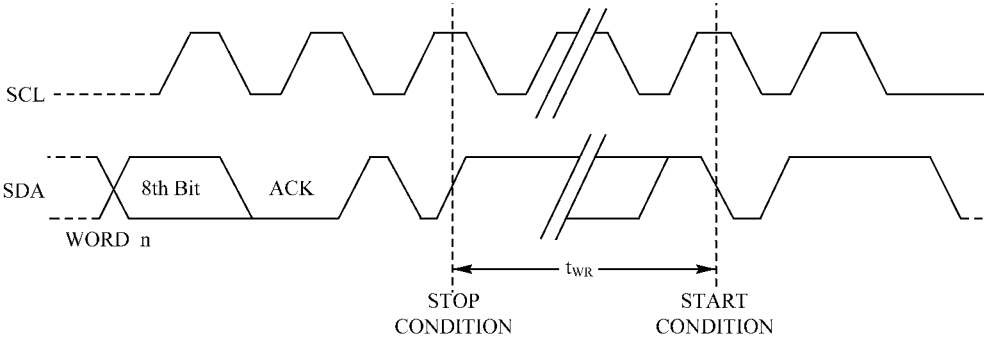


图 6-21 芯片内部写周期时间 t_{WR}

页面写

这种写入方式执行含若干字节的 1 个页面的写入。对于 AT24C01A/02,它们的 1 个页面含 8 个字节;对于 AT24C04/08/16,它们的 1 个页面含 16 个字节。使用时要注意其页面的定义不同。

页面写入的过程如图 6-22 所示。

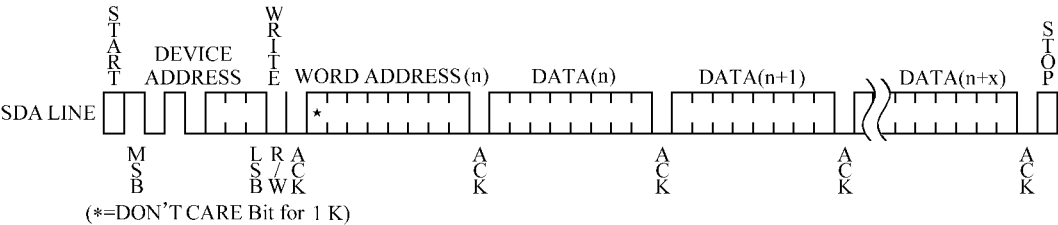


图 6-22 页面写

页面写的开头部分和字节写一样。在起始状态,首先写入 8 位器件地址;待 EEPROM 应答了“0”信号 ACK 之后,写入 8 位字地址;又待芯片应答了“0”信号 ACK 之后,写入 8 位数据。

随后页面写的过程则和字节写有区别。

当芯片接收了第一个 8 位数据并产生应答信号 ACK 之后,单片机可以连续向 EEPROM 芯片发送共 1 页面的数据。对于 AT24C01A/02,可发送共 1 个页面的 8 个字节(连第一个 8 位数据在内)。对于 AT24C04/08/16,则共可发送 1 个页面共 16 个字节(连第一个 8 位数据在内)。当然,每发一个字节都要等待芯片的应答信号 ACK。

之所以可以连续向芯片发送 1 个页面数据,是因为字地址的低 3~4 位在 EEPROM 芯片内部可实现加 1,字地址的高位不变,用于保持页面的行地址。对于 AT24C01A/02,字地址的低 3 位可实现内部加 1;对于 AT24C04/08/16,字地址的低 4 位可实现内部加 1。当传送给芯片的数据超过一个页面时,字地址会产生循环覆盖,先前所写的数据会被新的所覆盖。

页面写和字节写两者一样,都分为外部写和内部写过程。

应答查询:应答查询是单片机对 EEPROM 工作状态的一种检测。单片机查询到 EEPROM 有应答“0”信号 ACK 输出,则说明其内部定时写的周期结束,可以写入新的内容。单片机是通过发送起始状态及器件地址进行应答查询的。由于器件地址可以选择芯片,则检测芯片送出到 SDA 的状态就可以知道其是否有应答了。

③ 读操作:读操作的启动是和写操作类同的。它一样需要图 6-19 所示的器件地址字。和写操作不同的就是 R/W 信号为“1”时执行读操作。

读操作有 3 种方式,即现行地址读、随机读和顺序读。下面分别说明它们的工作过程。

现行地址读

在上次读或写操作完成之后,芯片内部字地址计数器会加 1,产生现行地址。只要没有再执行读或写操作,这个现行地址就会在 EEPROM 芯片保持接电的期间一直保存。一旦器件地址选中 EEPROM 芯片,并且有 R/W=1,则在芯片的应答信号 ACK 之后把读出的现行地址的数据送出。现行地址的数据输出时,就由单片机一位一位接收,接收后单片机不用向 EEPROM 发应答信号 ACK“0”电平,但应保证发出停止状态的信号以结束现行地址读操作。现行地址读操作如图 6-23 所示。

现行地址读会产生地址循环覆盖现象,但和写操作的循环覆盖不同。在写操作中,地址的循环覆盖是现行页面的最后一个字节写入之后,再行写入则覆盖同一页面的第一个字节。而在现行地址读操作中,地址的循环覆盖是在最后页面的最后一个字节读出之后,再行读出才覆盖第一个页面的第一个字节。

随机读

随机读和现行地址读的最大区别在于随机读会执行一个伪写入过程以把字地址装入 EEPROM 芯片中,然后执行读出,如图 6-24 所示。

显然,随机读有 2 个步骤。

第一,执行伪写入——把字地址送入 EEPROM,以选择需读的字节。

第二,执行读出——根据字地址读出对应内容。

当 EEPROM 芯片接收了器件地址及字地址时,在芯片产生应答信号 ACK 之后,单片机

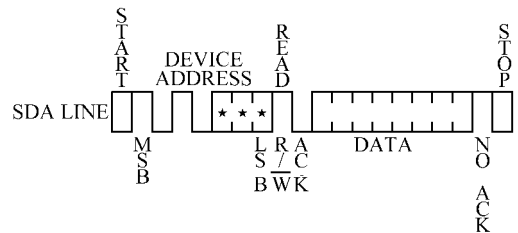


图 6-23 现行地址读

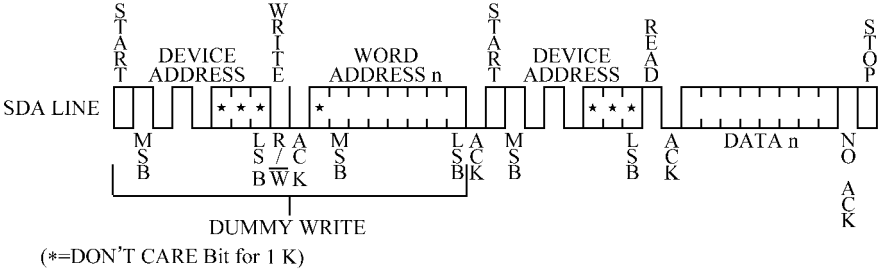


图 6-24 随机读

必须再产生一个起始状态,执行现行地址读,这时单片机再发出器件地址并且令 $R/W=1$,则 EEPROM 应答器件地址并行输出被读数据。在数据读出时由单片机执行一位一位接收,接收完毕后,单片机不用发“0”应答信号 ACK,但必须产生停止状态以结束随机读过程。

应该注意:在随机读的第二个步骤是执行现行地址读的,由于第一个步骤时芯片接收了字地址,故现行地址就是所送入的字地址。

顺序读

顺序读可以用现行地址读或随机读进行启动。它和现行地址读、随机读的最大区别在于:顺序读在读出一批数据之后才由单片机产生停止状态结束读操作;而现行地址读和随机读在读出一个数据之后就由单片机产生停止状态结束读操作。顺序读如图 6-25 所示。

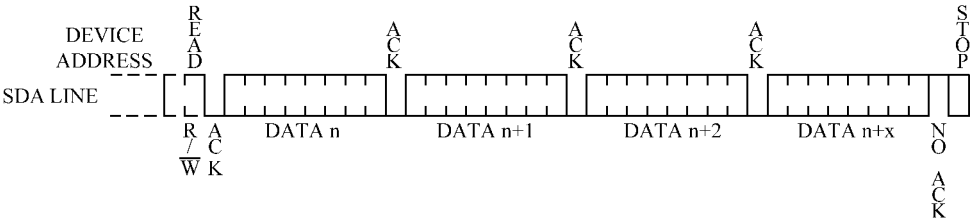


图 6-25 顺序读

执行顺序读时,首先执行现行读或随机读的有关过程,在读出第一个数据之后,单片机输出“0”应答信号 ACK。在芯片接收应答信号 ACK 后,就会对字地址进行计数加 1,随后串行输出对应的字节。当字地址计数达到存储器地址的极限时,则字地址会产生覆盖,顺序读将继续执行。只有在单片机不再产生“0”应答信号 ACK,而在接收数据之后马上产生停止状态,才会结束顺序读操作。

在对 AT24CXX 系列执行读写的 2 线串行总线工作中,其有关信号是由单片机的程序和 EEPROM 产生的。有两点特别要记住:串行时钟必须由单片机程序产生,而应答信号 ACK 则是由接收数据的器件产生,也就是写地址或数据时由 EEPROM 产生 ACK,而读数据时由单片机产生。

(4) AT24CXX 系列应用注意事项

① AT24CXX 系列型号:AT24CXX 系列 EEPROM 有 13 种型号。它们的容量不同,执行页面写时的页面定义不同,进行读写时的地址位数也不同,器件地址不同。有关主要指标如表 6-11 所列。在应用中要加以区别和注意。

表 6 - 11 AT24CXX 系列的型号及指标

型 号	容量/位	页面写时 所含字节	总线可接 芯片数	器件地址	字地址位数
AT24C01	1 K	4	1	None	
AT25C01A	1 K	8	8	A0,A1,A2	
AT24C02	2 K	8	8	A0,A1,A2	
AT24C04	4 K	16	4	A1,A2	
AT24C08	8 K	16	2	A2	
AT24C16	16 K	16	1	None	
AT24C164	16 K	16	8	A0,A1,A2	
AT24C32	32 K	32	8	A0,A1,A2	
AT24C64	64 K	32	8	A0,A1,A2	
AT24C128	128 K	64	4	A0,A1	
AT24C256	256 K	64	4	A0,A1	
AT24C512	512 K	128	4	A0,A1	
AT24C1024	1 M	256	2	A1	

② 读/写操作的种类:AT24CXX 系列的读/写操作有 5 种,如表 6 - 12 所列。

表 6 - 12 AT24CXX 系列的读/写操作

编 号	操 作	功能说明
1	字节写	写入一个字节
2	页面写	写入一个页面。不同的型号,在页面写时,一个页面所含的字节数不同,如表 6 - 11 所列
3	现行地址读	根据 AT24CXX 字地址计数器的内容,把该地址的内容读出来
4	随机读	把字地址写入 AT24CXX 的字地址计数器,再把该地址的内容读出来
5	顺序读	以现行地址读或随机读进行一批地址顺序连续的数据读出

2. AT93C06/46 串行 EEPROM

AT93C06 和 AT93C46 分别是 256 位,1 024 位的 EEPROM。它们的内部存储方式为 16×16位、64×16 位,也就是说以 16 位为一个字来存储。所以,AT93C06 内部有 16 个 16 位的数据字;AT93C46 内部有 64 个 16 位的数据字。

AT93C06/46 采用 CMOS 浮栅工艺制造,故有速度快、功耗低的特点。它们以 8 引脚的 DIP 或 SOIC 封装。由于采用串行电路接口,所以,指令、地址以及写入数据在数据输入端 DI (Data In)输入;读出的数据以及芯片状态在数据输出端 DO(Data Out)输出。所有的数据输

入输出都是以串行时钟 SK 同步,并且在 SK 的上升沿时起作用的。AT93C06 和 AT93C46 的串行接口可以很简单地和单片机相连。

AT93C06/46 有 7 种操作指令,它们分别是:读指令、擦/写启动指令、擦/写屏蔽指令、写指令、擦除指令、全写指令、全擦除指令。

AT93C06/46 在执行写指令或全写指令之前,不需要执行擦除。其擦除指令、全擦除指令只是为了保持和 NMOS 工艺的 9346 的读及编程的兼容性。为了简化执行过程,所有的编程周期都是自定时的。在 DO 端输出的“忙”状态指明芯片在执行编程。在出厂时芯片已被擦除,即所有存储单元为“1”态。

AT93C56,AT93C66 分别是 2 K 位和 4 K 位的 EEPROM。它们的引脚和 AT93C06/46 兼容。不过 AT93C56 和 AT93C66 所需的存取地址要比 AT93C06/46 多 2 位,即用 8 位地址。

(1) 封装及引脚意义

AT93C06/46 的外形封装和引脚如图 6-26 所示。在 8 条引脚中,有 2 条是无用引脚 NC,另外的 6 条引脚分别是 V_{CC} ,GND,CS,SK,DI,DO。

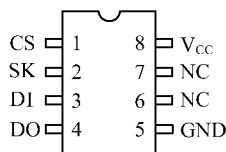


图 6-26 AT93C06/46 外形引脚

AT93C06/46 的引脚意义分别介绍如下。

① V_{CC} :电源引脚,+5 V。

② GND:地线。

③ CS:片选信号。当 CS=1 时,选中芯片;当 CS=0 时,不选中芯片且强迫芯片进入备用方式。一旦编程周期启动,则不管 CS 的输入信号是否变为 0,编程周期都会继续执行直到完结。在编程周期结束之前,如果 CS 的输入信号变为 0,则编程周期结束之后芯片马上进入备用方式。

在两条相邻指令执行中,CS 必须最少产生 500 ns 的低电平,以指明前条指令的结束。在 CS 处于低电平时,内部控制逻辑处于复位状态。

在读指令中,当读出数据完全输出之后,CS 会产生不少于 500 ns 的低电平。

在擦/写启动指令中,当指令的起始位、操作码及地址部分送出后,CS 会产生一个低电平脉冲。

在擦/写屏蔽指令中,当指令的起始位、操作码及地址部分送出后,CS 产生一个低电平脉冲。

在写指令中,当指令内容及写入数据发出后,CS 产生一个低电平脉冲。

在全写指令中,和写指令一样,当指令内容及写入数据发出之后,CS 产生一个低电平脉冲。

在擦除指令中,当指令的起始位、操作码及地址部分送出后,CS 产生一个低电平脉冲。

在全擦除指令中,和擦除指令一样,当指令内容即起始位、操作码及地址部分送出之后,CS 产生一个低电平脉冲。

CS 的低电平脉冲宽度不同厂家有所区别。一般为 100~500 ns 以上。

④ SK:串行时钟。串行时钟 SK 用于在单片机和 AT93C06/46 之间的数据通信同步。

操作码、地址和写入数据在 SK 的上升沿同步下输入到 AT93C06/46 内部。读出数据同样在 SK 的上升沿同步下从 AT93C06/46 输出。

串行时钟 SK 可以在任何位置停在当时的电平状态,也可以在任何时候继续发送。这一特点给单片机在准备操作码、地址和写入数据时以很大的自由度。

当选择信号 CS=0 时,串行时钟 SK 状态可以不加考虑;当 CS=1,但起始条件尚未被检测时,任何数量的时钟周期都不会改变 AT93C06/46 的当时状态;在起始条件检测之后,必须提供一定数量的时钟周期,以把所需的操作码、地址和写入数据同步送入 AT93C06/46 内部,然后执行相关的指令。在执行指令时,串行时钟 SK 和数据输入端 DI 的状态无关紧要。芯片等待的是新的起始状态。

⑤ DI:数据输入端。数据输入端 DI 用于在串行时钟 SK 的同步下输入起始位、操作码、地址和写入数据。

⑥ DO:数据输出端。数据输出端 DO 用于在读方式中,在串行时钟 SK 的同步下输出读出的数据。

DO 端同样在擦除或写入周期中提供就绪/忙信息。

在擦除指令时,如果串行时钟已把地址 A0 接收入芯片,则其下降沿表明已完成了指令的最后一位的接收,同时,擦除操作开始。在 CS 产生一个低电平之后,接着从低变高,在 DO 端输出就绪/忙信号。在执行写指令时,情况相同。

CS 信号从高电平下降到低电平,然后又上升到高电平,这一过程才会使 DO 端输出就绪/忙信号。如果在擦除或写入的整个时间中,CS 保持低电平或高电平,则就绪/忙信号不会在 DO 端出现。

如果在擦除或写入周期之后需要检测就绪/忙信号,则应在 DO 端接上拉电阻,以便检测处于高电平的就绪信号。

DO 和 DI 可以连接在一起组成 DI/DO 线,并且和 CS,SK 信号一同组成 3 线的接口。

在读指令被检测之后,从 DO 输出的数据的首部有一位伪“0”信号,这一点是要小心注意到,切不可把这位伪“0”信号作为数据的高位。

如果在擦除或写周期,AT93C06/46 输出就绪/忙状态,被选中的芯片不能使用 DI/DO 方式的 3 线接口。

(2) 指令操作说明

在 AT93C06/46 的工作中,有 7 种不同的指令操作。这 7 种指令及其含义如表 6-13 所列。

表 6-13 AT93C06/46 的指令及含义

指 令	起始位	操作码	地 址							输入数据	输 出	时钟周期
READ	1	1 0	A5	A4	A3	A2	A1	A0	—		D15~D0	25
WRITE	1	0 1	A5	A4	A3	A2	A1	A0	D15~D0		(RDY/ $\overline{\text{BSY}}$)	25
ERASE	1	1 1	A5	A4	A3	A2	A1	A0	—		(RDY/ $\overline{\text{BSY}}$)	9
EWEN	1	0 0	1	1	X	X	X	X	—		High-Z	9
EWDS	1	0 0	0	0	X	X	X	X	—		High-Z	9
ERAL	1	0 0	1	0	X	X	X	X	—		(RDY/ $\overline{\text{BSY}}$)	9
WRAL	1	0 0	0	1	X	X	X	X	D15~D0		(RDY/ $\overline{\text{BSY}}$)	25

AT93C06/46 在执行指令之前,需要一个起始状态。当检测到起始位时,表示进入了起始状态,一条指令接着被接收。

如果 CS 和 DI 处于高电平,则在串行时钟的上升沿时检测到起始位。实际上,CS=1,表示选中了芯片;DI=1,表示是起始位;串行时钟一来,其上升沿把起始位送入 AT93C06/46 中。此时,起始状态被检测到。

在检测到起始状态之前,CS,CLK 和 DI 可以以任何组合方式变化,当然,除了起始状态的组合。这时,任何指令操作都不会有效。

一旦 CS 变成高电平,芯片不会再处于备用方式。

紧接起始状态之后,如果指令所需的操作码、地址及写入数据已在 SK 同步下送入 EEPROM 中,则指令被马上执行。

在指令执行之后,SK 和 DI 的电平状态无关紧要,直到新的起始状态被检测到为止。

AT93C06/46 的工作指令分别说明如下。

① 读指令 READ

读指令在 DO 端输出串行 16 位数据。在数据之前会有一个伪“0”信号,16 位数据按先高位后低位方式输出。读指令的时序如图 6-27 所示。

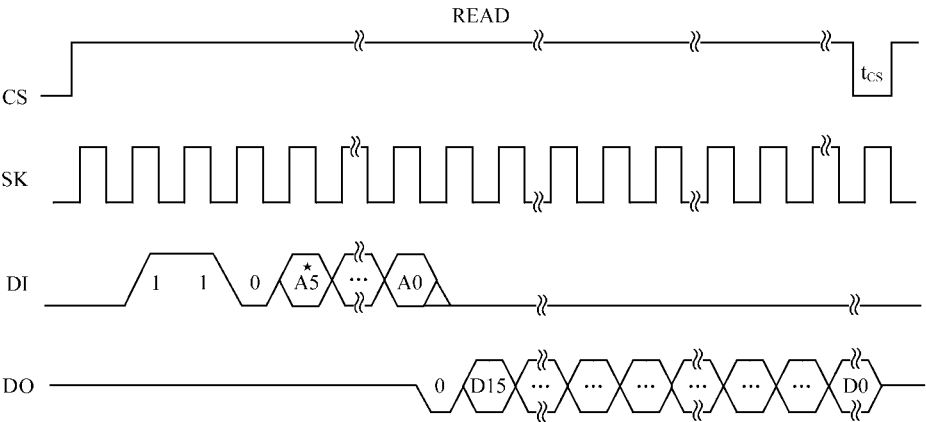


图 6-27 读指令时序

当读指令的操作码和地址被 EEPROM 芯片接收之后,则称读指令被接收。接着指令及地址被译码,AT93C06/46 把数据从所选中的存储单元送入 16 位移位寄存器中,然后串行从 DO 端输出。在读出的 16 位数据输出之前,会先在 DO 中输出 1 位伪“0”信号。所以,在 DO 输出的信号是 17 位,其格式为:0 D15 D14...D1 D0。

在数据完整输出之后,片选择信号 CS 应产生一个宽度大于 500 ns 的负脉冲。

② 擦/写启动指令 EWEN

当 EEPROM 接通电源时,会使芯片进入擦/写屏蔽状态。因此,所有的编程指令必须在执行了擦/写启动指令之后才能执行。一旦执行了擦/写启动指令 EWEN,则可一直对芯片执行各种编程操作。只有在执行擦/写屏蔽指令 EWDS 之后,才会不允许编程。当然,断电也会停止芯片的编程功能。

擦/写启动指令的时序如图 6-28 所示。

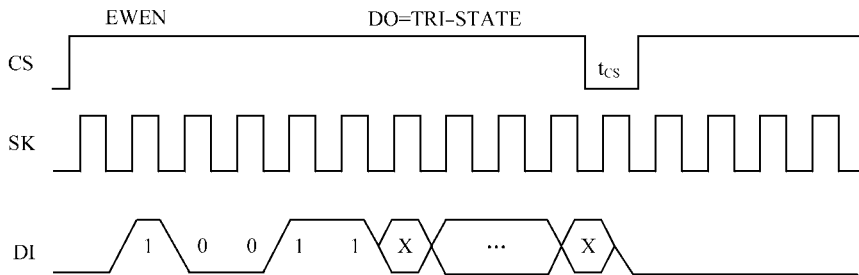


图 6-28 擦/写启动指令时序

在图中,没有 DO 的波形图,原因是在擦/写启动指令执行时,DO 的输出为高阻抗的三态输出。

③ 擦除指令 ERASE

擦除指令对地址所选中的存储单元进行擦除,令其内容为全“1”。当地址的最后一位 A0 装入芯片之后,片选信号 CS 就从高电平变低电平,这个下降沿表示芯片内部的自定时编程周期开始。擦除指令 ERASE 的时序如图 6-29 所示。

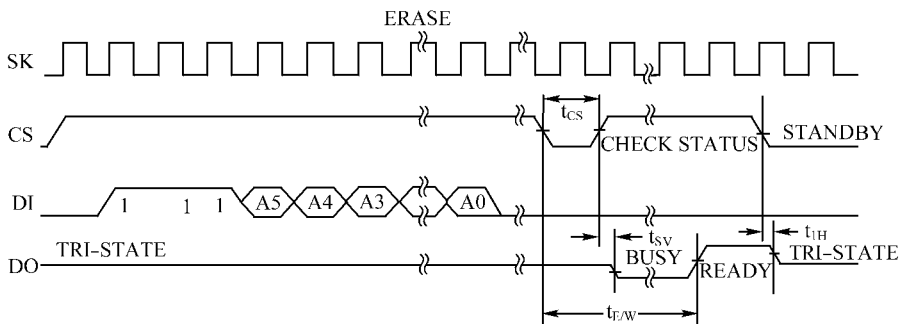


图 6-29 擦除指令时序

当 CS 信号的低电平保持了约 500 ns 之后,就会从低电平又变为高电平,DO 端就会输出芯片的就绪/忙状态,即 READY/BUSY 信号。当 DO=0 时,即就绪/忙信号为低电平,表示芯片处于编程进程之中,即是“忙”;当 DO=1,即就绪/忙信号为高电平,表示指令地址所寻址的存储单元已被擦除,芯片已就绪等待其他指令操作。

④ 全擦指令 ERAL

全擦指令同时把 EEPROM 的所有存储单元擦除为全“1”。除了操作码不同之外,全擦指令的工作周期和擦除指令是相同的。当然,全擦指令的地址部分是 6 位特殊码,而不是地址。全擦指令的时序如图 6-30 所示。

在全擦指令中,在 CS 产生了一定宽度的负脉冲之后,又恢复为高电平,DO 输出芯片的就绪/忙状态。

⑤ 写指令 WRITE

当把指令及 16 位数据传入芯片之后,写指令开始执行。写指令的时序如图 6-31 所示。当在 DI 端的 16 位数据的最后 1 位在 SK 同步送入芯片之后,CS 信号在 SK 的下一个上升沿来到之前必须降为低电平。CS 的下降沿表示芯片内部的自定时编程周期开始。

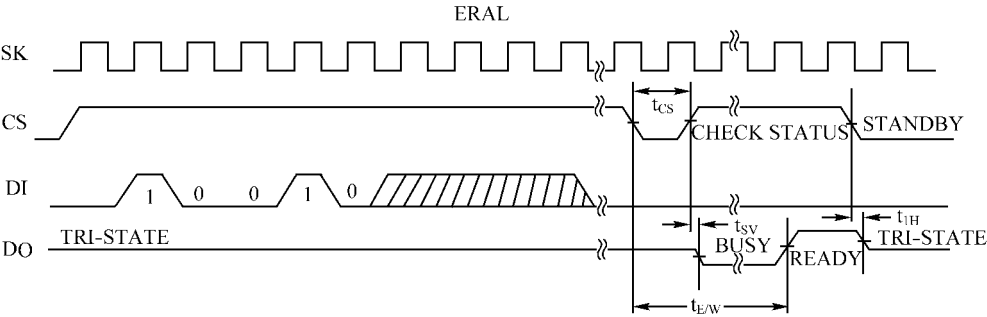


图 6-30 全擦指令时序

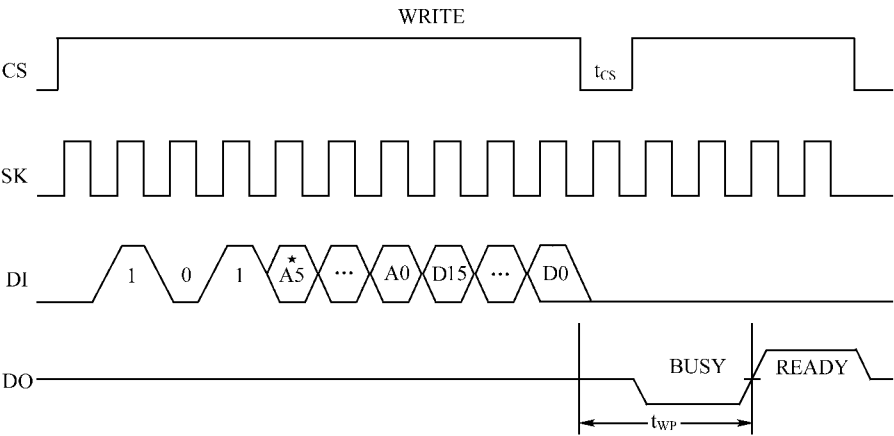


图 6-31 写指令时序

当 CS 信号保持低电平一定的宽度之后,又从低电平升到高电平。这时 DO 就输出芯片的内部状态。DO=0,表示芯片的编程仍然在进程之中;DO=1,则表示指令地址所寻址的存储单元已完成了数据的写入,芯片已就绪等待另外的指令操作。

⑥ 全写指令 WRAL

全写指令把数据同时写入到所有的存储单元中。和写指令相同,DO 端的状态表明芯片内部的工作状态。CS 在保持一定宽度的低电平之后,又变成高电平,DO 状态说明了芯片的就绪/忙状态。DO=0,芯片忙于编程;DO=1,芯片编程完毕,就绪等待其他操作指令。全写指令的时序如图 6-32 所示。

⑦ 擦/写屏蔽指令

为了保护 AT93C06/46 所写入的数据,在编程操作之后应执行擦/写屏蔽指令,以禁止任何不希望的对芯片的误写入操作。只要把擦/写屏蔽指令的内容,包括起始位、操作码、地址接收入芯片,就会马上禁止任何后续的写入。擦/写屏蔽指令的时序如图 6-33 所示。

读指令的执行与擦/写启动指令 EWEN 及擦/写屏蔽指令 EWDS 无关,所以不受这两条指令约束。只有写、擦除等有关指令才受这两条指令影响。

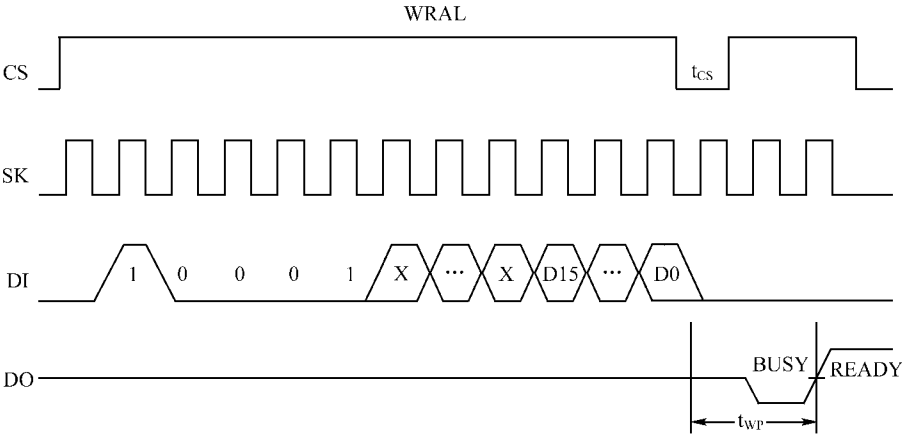


图 6-32 全写指令时序

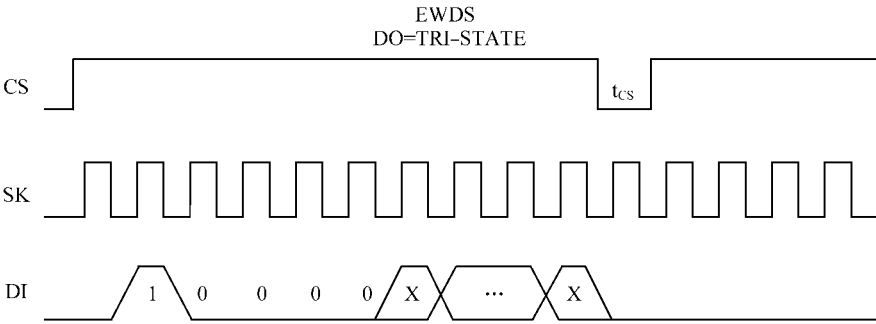


图 6-33 擦/写屏蔽指令时序

(3) 其他型号的 AT93CXX EEPROM

AT93CXX 系列的 EEPROM 除了 AT93C06/46 之外，还有 AT93C56，AT93C57，AT93C66，AT93C46A 等。

这些型号的读写操作都是相同的，即都有读、擦/写启动、擦/写屏蔽、擦除、写、全写、全擦等 7 条指令。

① AT93C46/56/57/66 的引脚 ORG

ORG 引脚是封装引脚中的 6 号引脚，在图 6-26 中，6 号引脚是 NC，即无用引脚；而在不少封装中它是 ORG 端。ORG 称为内部组织引脚，用于定义内部数据输入输出的位数。

ORG=1 时，定义存储器内部数据为 16 位；

ORG=0 时，定义存储器内部数据为 8 位。

② AT93C46/56/57/66 的指令

AT93C46/56/57/66 的容量分别是 1 K，2 K，4 K 位容量的 EEPROM。它们的指令操作码一样，只是地址位数不同。

指令格式如表 6-13 所列。

AT93C46：ORG=1 时，数据为 16 位，地址为 A5~A0 共 6 位；ORG=0 时，数据为 8 位，地址为 A1~A0 共 7 位。

AT93C57;ORG=1 时,数据为 16 位,地址为 A6~A0 共 7 位;ORG=0 时,数据为 8 位,地址为 A7~A0 共 8 位。

AT93C56/66;ORG=1,数据为 16 位,地址为 A7~A0 共 8 位;ORG=0 时,数据为 8 位,地址为 A8~A0 共 9 位。

③ AT93C46A

AT93C46A 的封装引脚 6 是无用引脚,相当于 ORG=1 时的 AT93C46 的功能。

6.3.2 AT24CXX 系列 EEPROM 的接口方法

AT24CXX 系列 EEPROM 是采用 2 线接口的存储器芯片。因此,89 系列单片机和 AT24CXX 系列 EEPROM 之间进行数据传送只需用两条连线就可以了。这两条串行通信接口线分别称串行时钟线 SCL 和串行数据线 SDA,也称 2 线串行总线。

由于 AT24CXX 系列 EEPROM 的内部容量按照型号而有所不同,所以,EEPROM 的地址硬布线也有所不同,即它们的地址引脚数量虽然有 A0~A2 共 3 条,但有效地址引脚却不同。同时,在 2 线串行总线上可以连接的片数也有区别。表 6-14 给出了各种不同型号的 EEPROM 以及它们的有关指标。这些指标包括容量、可用地址引脚、在 2 线串行总线上可以连接的片数、时钟频率、时钟低电平脉宽、时钟高电平脉宽。

表 6-14 AT24CXX 系列技术参数

型 号	容 量 /KB	有效地 址引脚	总线可 接片数	时钟/kHz	时钟低 电平脉宽/ μ s	时钟高 电平脉宽/ μ s
AT24C01A	1	A0,A1,A2	8	400 (电源 5 V)	最小 1.2 (电源 5 V)	最小 0.6 (电源 5 V)
AT24C02	2	A0,A1,A2	8	100 (电源小于 5 V)	最小 4.7 (电源小于 5 V)	最小 4.0 (电源小于 5 V)
AT24C04	4	A1,A2	4			
AT24C08	8	A2	2			
AT24C16	16	无	1			

在表 6-14 中可以看出:不同的型号在 2 线串行总线上可以连接的片数不同。对 AT24C01A 和 AT24C02,在 2 线串行总线上可以连接 8 片;而对于 AT24C16 来说,则只能连接 1 片。在实际逻辑电路中,SCL 线和 SDA 线有时要产生高电平。由于在 EEPROM 内部,它们是漏极开路结构的,所以,需要外接上拉电阻。

一种典型的 2 线串行总线连接 89 系列单片机和 AT24C02 的接口结构如图 6-34 所示。在图中,AT89C2051 通过 P1 端口的 P1.2,P1.3 和 EEPROM AT24C02 的 SCL,SDA 端相连,从而形成了 2 线串行总线。为了使 SCL,SDA 能产生可靠的高电平,在 2 线串行总线上都接了上拉电阻。接在 2 线串行总线上的 AT24C02 可达 8 片,而每一片的硬布线地址不同。第 1 片 AT24C02 的硬布线地址为“000”,故 A2,A1,A0 都接地;第 2 片的硬布线地址为“001”,故 A2,A1 接地,A0 则接 5 V……第 8 片的硬布线地址为“111”,故 A2,A1,A0 全接 5 V 电源。

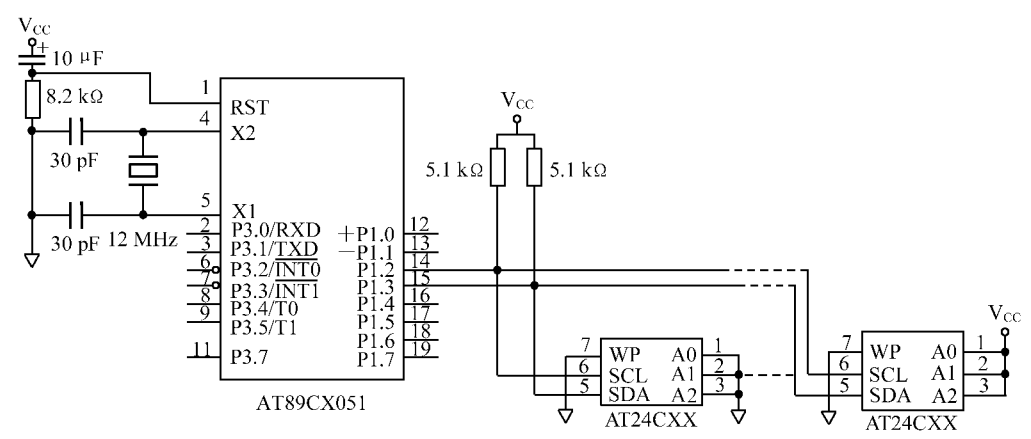


图 6-34 AT89C2051 和 AT24C02 的接口

为了在 AT89C2051 和 AT24C02 之间能进行正常的数据写入和读出,要求 AT89C2051 中的工作程序能够以恰当的时序产生串行时钟 SCL 和串行数据 SDA。而串行时钟 SCL 和串行数据 SDA 的参数必须满足 AT24C02 的性能要求,这样才能保证单片机 AT89C2051 对 AT24C02 的正确写入和读出。对于 AT24CXX 系列 EEPROM 来说,其串行时钟 SCL 及串行数据的有关参数分别如图 6-35 所示和表 6-15 所列。

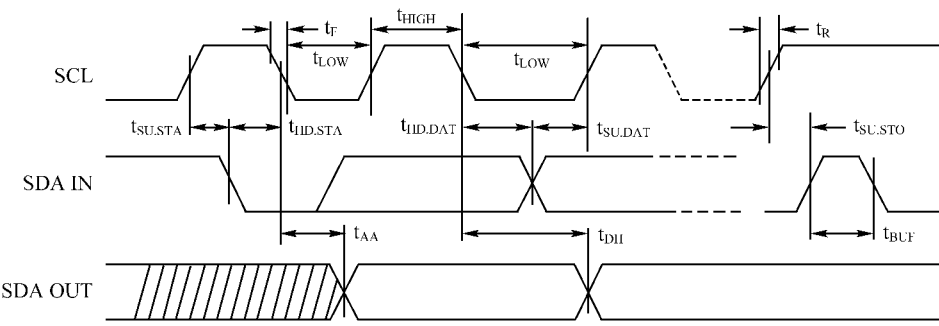


图 6-35 AT24CXX 系列串行总线时序

在图 6-35 中,给出了建立起始状态的时间 $t_{SU, STA}$,时钟脉冲低电平脉宽 t_{LOW} ,时钟高电平脉宽 t_{HIGH} ,起始状态保持时间 $t_{HD, STA}$,数据输入保持时间 $t_{HD, DAT}$,数据输入建立时间 $t_{SU, DAT}$,时钟变低到数据输出有效时间 t_{AA} ,数据输出保持时间 t_{DI} ,停止建立时间 $t_{SU, STO}$,时钟上升沿时间 t_R ,时钟下降沿时间 t_F ,总线自由时间 t_{BUF} 等。这些都是 SCL 和 SDA 的技术指标参数,在实际工作中,对这些参数都有严格的要求。这些技术指标的参数要求如表 6-15 所列。

双向数据传输协议是采用一个通用的双线总线。AT24CXX 系列允许一定数量的兼容器件共用这条双线总线。这种总线由一条串行时钟线(SCL)和一条串行数据线(SDA)组成。时钟是由总线主控器单片机产生的,且数据在数据线上串行传送时最高有效位在前,最低有效位在后,同步于时钟,这个通信协议支持 8 位的双向数据传送。

在这个应用中,单片机 AT89C2051 作为一个总线主控器,启动所有传送数据和生成用于同步数据流的时钟。串行存储器 AT89C2051 是从属部件连在总线上,由主控器控制数据的接收和发送。

表 6-15 AT24CXX 系列 EEPROM 技术参数

符 号	参 数	2.7 - , 2.5 - , 1.8 - volt		5.0 - volt		Units
		Min	Max	Min	Max	
f _{SCL}	时钟频率, SCL		100		400	kHz
t _{LOW}	时钟脉冲低电平宽度	4.7		1.2		μs
t _{HIGH}	时钟脉冲高电平宽度	4.0		0.6		μs
t _I	抑制噪声时间		100		50	ns
t _{AA}	时钟变低到数据输出有效	0.1	4.5	0.1	0.9	μs
t _{BUF}	数据传送前总线空闲时间	4.7		1.2		μs
t _{HD, STA}	起始保持时间	4.0		0.6		μs
t _{SU, STA}	起始建立时间	4.7		0.6		μs
t _{HD, DAT}	数据输入保持时间	0		0		μs
t _{SU, DAT}	数据输入建立时间	200		100		ns
t _R	输入上升时间		1.0		0.3	μs
t _F	输入下降时间		300		300	ns
t _{SU, STO}	停止建立时间	4.7		0.6		μs
t _{DH}	数据输出保持时间	100		50		ns
t _{WR}	写周期		10		10	ms

总线控制器通过在总线上产生启动信号来启动数据传送,随后则是送一个含有预定器件地址的字节。这个器件地址由 4 个固定部分和 3 个可编程部分组成,如图 6-19 所示。3 个可编程部分用作器件识别地址,而 4 个固定部分则作为存储器标识。当总线有多个类似器件供主控器选择时,可编程部分必须配对使用。

AT24CXX 串行 EEPROM 的器件地址固定部分为“1010”,可编程地址即硬布线地址为 A0,A1,A2。

器件地址的最低位用于说明读或写的操作。在最低位发送后,主控器将释放这条数据线并生成第 9 个时钟。如果从属部件识别出器件地址,它将在数据线上生成一个应答信息来响应第 9 个时钟。当从属器件忙时,将不对地址产生确认,比如用 AT24CXX 正在进行写操作时即如此。收到从属器件的地址应答信号以后,主控器继续传送数据。若定义为写操作,则主控器会在收到从属器件应答信号后发送剩余的数据;若定义为读操作,则主控器将会从从属器件发送数据时放弃数据线和时钟,在收到每一个字节后,主控器单片机将在总线上生成一个应答信号。在收到最后一位字节后应答信号将被省略。主控器通过在总线上产生一个停止信号来终止所有操作,也可以在任何时候通过这个停止信号来中断数据的传送。

结合图 6-34,给出一个程序集,它可以用于 AT89C2051 对 AT24CXX 系列 EEPROM 执行读写操作。由于这个程序集所用的数据缓冲器及堆栈单元较多,所以,只能用于 RAM 容量较大的 AT89C2051。如果把这个程序集用于 AT89C1051,则要对程序进行修改,使一些数据

缓冲器及堆栈单元可重复使用。不过要注意,最小缓冲器数量是与执行页面写时的页面大小有关的,而 AT24CXX 系列中不同的型号在执行页面写时,页面所含的字节数不同,最少含 8 个字节,最多含 256 个字节。

这个程序集提供了 AT24CXX 系列所需的 5 种读写操作,在程序中分别标识如下。

现行地址读:read-current;随机读:read-random;顺序读:read-block;字节写:write-byte;页面写:write-block。

这个程序集支持在一条 2 线总线中对多片 AT24CXX 系列 EEPROM 进行读写。每一片的器件由硬布线地址引脚 A0,A1,A2 组成。

在程序中还给出了对 AT24C64 进行读写操作并进行调试及展示的命令:字节填写 byte-fill,校验字节填写 verify-byte-fill,页面填写 page-fill 和校验页面填写 verify-page-fill。只要对程序中下列参数进行重新定义和填值,就可以用于对 AT24CXX 系列的其他型号进行读写调试及展示。这些参数分别是:EEPROM 所含的字节数 SIZE、写入时每页所含的字节数 PSIZE、填写值 FILL、器件地址中的可编程地址(即 A0,A1,A2)部分 PADDR。

当所用的单片机 AT89C2051 工作在 12 MHz 的时钟时,程序满足所有 AT24CXX 系列型号的时序要求。当采用更高速度的时钟时,则程序需要做一定的修改。

程序清单如下:

```
00A0      39      FADDR      EQU   0a0h      ; fixed address for AT24CXX
                                         EEPROM
0000      40      PADDR      EQU   0          ; programmable address (0...7)
2000      41      SIZE       EQU   2000h      ; bytes per AT24C64
0020      42      PSIZE      EQU   32         ; bytes per page for AT24C64
0055      43      FILL       EQU   55h        ; example fill value
44
45          ;Register definitions
46
REG      47      index      EQU   r0          ; buffer pointer
REG      48      kount      EQU   r1          ; byte count register
REG      49      zdata      EQU   r1          ; data register
REG      50      addr_lo    EQU   r2          ; 2-byte address register
REG      51      addr_hi    EQU   r3
52
53          ; Microcontroller connections to AT24Cxx serial bus lines
54
0092      55      SCL        BIT    p1. 2      ; serial clock
0093      56      SDA        BIT    p1. 3      ; serial data
57
58
----      59                      DSEG AT 20H
60
0040      61                      ORG    40H
```

0040	62	buffer;	DS	PSIZE	; storage for read/write data
	63				
0060	64		ORG	60H	; stack origin
0060	65	stack;	DS	20H	; stack depth
	66				
	67				
----	68		CSEG		
	69				
0000	70		ORG	0000H	; power on/reset vector
0000 020080	71		jmp	on_reset	
	72				
0003	73		ORG	0003H	; external interrupt 0 vector
0003 32	74		reti		; undefined
	75				
000B	76		ORG	000BH	; timer 0 overflow vector
000B 32	77		reti		; undefined
	78				
0013	79		ORG	0013H	; external interrupt 1 vector
0013 32	80		reti		; undefined
	81				
001B	82		ORG	001BH	; timer 1 overflow vector
001B 32	83		reti		; undefined
	84				
0023	85		ORG	0023H	; serial I/O interrupt vector
0023 32	86		reti		; undefined
	87				
0080	88		ORG	0080H	; begin code space
	89		USING	0	; register bank zero
	90	on_reset;			
0080 75815F	91		mov	sp, # (stack-1)	; initialize stack pointer
	92				
	93				; Initialize AT24Cxx serial bus lines
	94				
0083 D293	95		setb	SDA	; high
0085 D292	96		setb	SCL	; high
	97				
	98				
0087 12009D	99		call	byte_fill	
008A 400F	100		jc	fault	
	101				
008C 1200BF	102		call	verify_byte_fill	
008F 400A	103		jc	fault	
	104				

0091 1200F2	105	call	page_fill	
0094 4005	106	jc	fault	
	107			
0096 120128	108	call	verify_page_fill	
0099 4000	109	jc	fault	
	110			
	111			fault:
009B 80FE	112	jmp	\$	
	113			
	114			
	115			byte_fill:
	116			
	117			; Fill every byte in an AT24Cxx with the same value
	118			; Writes one address at a time (page mode is not used)
	119			; Returns CY set to indicate write timeout
	120			; Destroys A, B, DPTR, XDATA, ADDR_HI:ADDR_LO
	121			
009D 7955	122	mov	zdata, # FILL	; set up fill data
009F 900000	123	mov	dptr, # 0	; initialize address pointer
	124		x51:	
00A2 AA82	125	mov	addr_lo, dpl	; set up address
00A4 AB83	126	mov	addr_hi, dph	
	127			
00A6 75F078	128	mov	b, # 120	; retry counter
	129		x52:	
00A9 7400	130	mov	a, # PADDR	; programmable address
00AB 1201CD	131	call	write_byte	; try to write
00AE 5007	132	jnc	x53	; jump if write OK
	133			
00B0 D5F0F6	134	djnz	b, x52	; try again
00B3 D3	135	setb	c	; set timeout error flag
00B4 0200BE	136	jmp	x54	; exit
	137		x53:	
00B7 A3	138	inc	dptr	; advance address pointer
	139	mov	a, dpl	; check low byte
	140	cjne	a, # (LOW SIZE), x51	; jump if not last
00B8 E583	141	mov	a, dph	; check high byte
00BA B420E5	142	cjne	a, # (HIGH SIZE), x51	; jump if not last
00BD C3	143	clr	c	; clear error flag
	144		x54:	
00BE 22	145	ret		
	146			
	147			

```

148         verify_byte_fill:
149
150             ; Verify that all bytes in an AT24Cxx match a fill value
151             ; Reads and verifies one byte at a time(page mode is not used)
152             ; Performs a Random Read function to initialize the internal
153             ; address counter and checks the contents of the first address
154             ; Then performs multiple Current Address Read functions to step
155             ; through the remaining addressess
156             ; Returns CY set to indicate read timeout or compare fail
157             ; Destroys A, B, DPTR
158
00BF 900000 159             mov     dpty, #0             ; initialize address pointer/
                                           counter
00C2 AA82   160             mov     addr_lo,dpl         ; set up address
00C4 AB83   161             mov     addr_hi,dph
162
00C6 75F078 163             mov     b, #120             ; retry counter
164 x81:
00C9 7400   165             mov     a, #PADDR             ; programmable address
00CB 12020D 166             call    read_random         ; try to read
00CE 5006   167             jnc     x82                 ; jump if read OK
168
00D0 D5F0F6 169             djnz    b,x81                 ; try again
00D3 0200F0 170             jmp     x86                 ; set error flag and exit
171 x82:
00D6 B45517 172             cjne    a, #FILL,x86         ; jump if compare error
00D9 0200E6 173             jmp     x85                 ; do remaining addresses
174 x83:
00DC 7400   175             mov     a, #PADDR
00DE 1201F3 176             call    read_current
00E1 400E   177             jc      x87                 ; jump if read fails
178
00E3 B4550A 179             cjne    a, #FILL,x86         ; jump if compare error
180 x85:
00E6 A3     181             inc     dptr                 ; advance address pointer
00E7 E583   182             mov     a,dph                 ; check high byte
00E9 B420F0 183             cjne    a, #(HIGH_SIZE),x83    ; jump if not last
00EC C3     184             clr     c                     ; clear error flag
00ED 0200F1 185             jmp     x87                 ; exit
186 x86:
00F0 D3     187             setb    c                     ; set error flag
188 x87:
00F1 22     189             ret

```

```

190
191
192  page_fill:
193
194          ; Fill every byte in an AT24Cxx with the same value
195          ; Writes one page at time
196          ; Returns CY set to indicate write timeout
197          ; Destroys A, B, DPTR, KOUNT, INDEX, ADDR_HI;ADDR_LO
198
199          ; First fill buffer
200
00F2 75F020 201          mov    b, # PSIZE      ; bytes per page
00F5 7840   202          mov    index, # buffer  ; point to buffer
203  x61:
00F7 7655   204          mov    @index, # FILL   ; put fill value in buffer
00F9 08     205          inc    index            ; advance pointer
00FA D5F0FA 206          djnz   b, x61           ; next byte
207
208          ; Copy buffer to device, one page at a time
209
00FD 900000 210          mov    dptr, # 0        ; initialize address pointer
211  x62:
0100 AA82   212          mov    addr_lo, dpl      ; set up address
0102 AB83   213          mov    addr_hi, dph
0104 7920   214          mov    kount, # PSIZE   ; bytes per page
215
0106 75F078 216          mov    b, # 120        ; retry counter
217  x63:
0109 7400   218          mov    a, # PADDR       ; programmable address
010B 12015F 219          call   write_block      ; try to write
010E 5007   220          jnc    x64              ; jump if write OK
221
0110 D5F0F6 222          djnz   b, x63           ; try again
0113 D3     223          setb   c                ; set timeout error flag
0114 020127 224          jmp    x66              ; exit
225  x64:
226          ; Add page size to address pointer
227
0117 E582   228          mov    a, dpl           ; get low byte
0119 2420   229          add    a, # PSIZE       ; add page size
011B F582   230          mov    dpl, a           ; save low byte
011D 5002   231          jnc    x65              ; jump if high byte not affected
011F 0583   232          inc    dph              ; increment high byte

```

	233	x65:		
	234		cjne	a, # (LOW SIZE), x62 ; jump if low byte not last
0121 E583	235		mov	a, dph ; check high byte
0123 B420DA	236		cjne	a, # (HIGH SIZE), x62 ; jump if not last
0126 C3	237		clr	c ; clear error flag
	238	x66:		
0127 22	239		ret	
	240			
	241			
	242	verify_page_fill:		
	243			
	244			; Verify that all bytes in an AT24Cxx match a fill value
	245			; Reads and verifies one page at a time
	246			; Returns CY set to indicate read timeout or compare fail
	247			; Destroys A, B, DPTR, KOUNT, INDEX, ADDR_HI; ADDR_LO
	248			
	249			; Copy device page to buffer
	250			
0128 900000	251		mov	dptr, # 0 ; initialize address pointer
	252	x71:		
012B AA82	253		mov	addr_lo, dpl ; set up address
012D AB83	254		mov	addr_hi, dph
012F 7920	255		mov	kount, # PSIZE ; bytes per page
	256			
0131 75F078	257		mov	b, # 120 ; retry counter
	258	x72:		
0134 7400	259		mov	a, # PADDR ; programmable address
0136 12018A	260		call	read_block ; try to read
0139 5007	261		jnc	x74 ; jump if read OK
	262			
013B D5F0F6	263		djnz	b, x72 ; try again
	264	x73:		
013E D3	265		setb	c ; set error flag
013F 02015E	266		jmp	x77 ; exit
	267	x74:		
	268			; Verify buffer contents
	269			
0142 75F020	270		mov	b, # PSIZE ; bytes per page
0145 7840	271		mov	index, # buffer ; point to buffer
	272	x75:		
0147 B655F4	273		cjne	@index, # FILL, x73 ; jump if compare fails
014A 08	274		inc	index ; advance pointer
014B D5F0F9	275		djnz	b, x75 ; next byte

```

276
277                ; Add page size to address pointer
278
014E E582 279                mov    a,dpl                ; get low byte
0150 2420 280                add    a,#PSIZE            ; add page size
0152 F582 281                mov    dpl,a              ; save low byte
0154 5002 282                jnc    x76                ; jump if high byte not affected
0156 0583 283                inc    dph                ; increment high byte
284    x76:
285                cjne    a,#(LOW SIZE),x71 ; jump if low byte not last
0158 E583 286                mov    a,dph                ; check high byte
015A B420CE 287                cjne    a,#(HIGH SIZE),x71 ; jump if not last
015D C3 288                clr     c                    ; clear error flag
289    x77:
015E 22 290                ret
291
292
293    write_block:
294
295                ; Write from one byte to one page of data to an AT24Cxx
296                ; Called with programmable address in A, address of first byte
297                ; in register pair ADDR_HI; ADDR_LO, data in BUFFER, byte
                ; count
298                ; in register KOUNT
299                ; Does not wait for write cycle to complete
300                ; Returns CY set to indicate that the bus is not available
301                ; or that the addressed device failed to acknowledge
302                ; Destroys A, KOUNT, INDEX
303
015F 120239 304                call    start
0162 4025 305                jc     x38                    ; abort if bus not available
306
0164 23 307                rl     a                        ; programmable address to bits
                ; 3:1
0165 44A0 308                orl     a,#FADDR            ; add fixed address
0167 C2E0 309                clr     acc.0                ; specify write operation
0169 120261 310                call    shout                ; send device address
016C 4018 311                jc     x37                    ; abort if no acknowledge
312
016E EB 313                mov     a,addr_hi                ; send high byte of address
016F 120261 314                call    shout
0172 4012 315                jc     x37                    ; abort if no acknowledge
316

```

0174 EA	317	mov	a,addr_lo	; send low byte of address
0175 120261	318	call	shout	
0178 400C	319	jc	x37	; abort if no acknowledge
	320			
017A 7840	321	mov	index,# buffer	; point to buffer
	322		x36:	
017C E6	323	mov	a,@index	; get data
017D 120261	324	call	shout	; send data
018D 4004	325	jc	x37	; abort if no acknowledge
	326			
0182 08	327	inc	index	; advance buffer pointer
0183 D9F7	328	djnz	kount,x36	; next byte
0185 C3	329	clr	c	; clear error flag
	330		x37:	
0186 120253	331	call	stop	
	332		x38:	
0189 22	333	ret		
	334			
	335			
	336		read_block:	
	337			
	338			; Read from one byte to one page of data from an AT24Cxx
	339			; Performs a Random Read which is extended into a Sequential Read
	340			; when more than one byte is read. Called with programmable address
	341			; in A, address of first byte in register pair ADDR_HI;ADDR_LO,
	342			; byte count in register KOUNT
	343			; Returns data in BUFFER. Returns CY set to indicate that the bus is
	344			; not available or that the addressed device failed to acknowledge
	345			; Destroys A, KOUNT, INDEX
	346			
	347			; Send dummy write command to address first byte
	348			
018A 120239	349	call	start	
018D 403D	350	jc	x35	; abort if bus not available
	351			
018F 23	352	rl	a	; programmable address to bits
				3;1
0190 44A0	353	orl	a,#FADDR	; add fixed address
0192 F8	354	mov	index,a	; save copy of device address
0193 C2E0	355	clr	acc.0	; specify write operation
0195 120261	356	call	shout	; send device address
0198 402F	357	jc	x34	; abort if no acknowledge
	358			

019A EB	359	mov	a,addr_hi	; send high byte of address
019B 120261	360	call	shout	
019E 4029	361	jc	x34	; abort if no acknowledge
	362			
01A0 EA	363	mov	a,addr_lo	; send low byte of address
01A1 120261	364	call	shout	
01A4 4023	365	jc	x34	; abort if no acknowledge
	366			
	367			; Send read command and receive data
	368			
01A6 120239	369	call	start	; second start for read
01A9 401E	370	jc	x34	; abort if bus not available
	371			
01AB E8	372	mov	a,index	; get device address
01AC D2E0	373	setb	acc, 0	; specify read operation
01AE 120261	374	call	shout	; send device address
01B1 4016	375	jc	x34	; abort if no acknowledge
	376			
01B3 7840	377	mov	index, # buffer	; point to buffer
	378	x31:		
01B5 120286	379	call	shin	; receive data byte
01B8 F6	380	mov	@index,a	; save data
	381			
01B9 B90106	382	cjne	kount, # 1,x32	; jump if not last byte
01BC 1202AC	383	call	NAK	; do not acknowledge last byte
01BF 0201C8	384	jmp	x33	; done
	385	x32:		
01C2 12029F	386	call	ACK	; acknowledge byte
01C5 08	387	inc	index	; advance buffer pointer
01C6 D9ED	388	djnz	kount,x31	; next byte
	389	x33:		
01C8 C3	390	clr	c	; clear error flag
	391	x34:		
01C9 120253	392	call	stop	
	393	x35:		
01CC 22	394	ret		
	395			
	396			
	397	write_byte:		
	398			
	399			; AT24Cxx Byte Write function
	400			; Called with programmable address in A, byte address in
	401			; register pair ADDR_HI;ADDR_LO,data in register XDATA

	402				; Does not wait for write cycle to complete
	403				; Returns CY set to indicate that the bus is not available
	404				; or that the addressed device failed to acknowledge
	405				; Destroys A
	406				
01CD 120239	407		call	start	
01D0 4020	408		jc	x49	; abort if bus not available
	409				
01D2 23	410		rl	a	; programmable address to bits 3:1
01D3 44A0	411		orl	a, #FADDR	; add fixed address
01D5 C2E0	412		clr	acc. 0	; specify write operation
01D7 120261	413		call	shout	; send device address
01DA 4013	414		jc	x48	; abort if no acknowledge
	415				
01DC EB	416		mov	a, addr_hi	; send high byte of address
01DD 120261	417		call	shout	
01E0 400D	418		jc	x48	; abort if no acknowledge
	419				
01E2 EA	420		mov	a, addr_lo	; send low byte of address
01E3 120261	421		call	shout	
01E6 4007	422		jc	x48	; abort if no acknowledge
	423				
01E8 E9	424		mov	a, zdata	; get data
01E9 120261	425		call	shout	; send data
01EC 4001	426		jc	x48	; abort if no acknowledge
	427				
01EE C3	428		clr	c	; clear error flag
	429	x48;			
01EF 120253	430		call	stop	
	431	x49;			
01F2 22	432		ret		
	433				
	434				
	435	read_current;			
	436				
	437				; AT24Cxx Current Address Read function
	438				; Called with programmable address in A. Returns data in A
	439				; Returns CY set to indicate that the bus is not available
	440				; or that the addressed device failed to acknowledge
	441				
01F3 120239	442		call	start	
01F6 4014	443		jc	x45	; abort if bus not available

	444			
01F8 23	445	rl	a	; programmable address to bits 3;1
01F9 44A0	446	orl	a, #FADDR	; add fixed address
01FB D2E0	447	setb	acc. 0	; specify read operation
01FD 120261	448	call	shout	; send device address
0200 4007	449	jc	x44	; abort if no acknowledge
	450			
0202 120286	451	call	shin	; receive data byte
0205 1202AC	452	call	NAK	; do not acknowledge byte
0208 C3	453	clr	c	; clear error flag
	454			x44:
0209 120253	455	call	stop	
	456			x45:
020C 22	457	ret		
	458			
	459			
	460			read_random:
	461			
	462			; AT24Cxx Random Read function
	463			; Called with programmable address in A, byte address in
	464			; register pair ADDR_HI; ADDR_L0. Returns data in A
	465			; Returns CY set to indicate that the bus is not available
	466			; or that the addressed device failed to acknowledge
	467			
020D C0F0	468	push	b	
020F F5F0	469	mov	b, a	; save copy of programmable address
	470			
	471			; Send dummy write command to set internal address
	472			
0211 120239	473	call	start	
0214 4020	474	jc	x47	; abort if bus not available
	475			
0216 23	476	rl	a	; programmable address to bits 3;1
0217 44A0	477	orl	a, #FADDR	; add fixed address
0219 C2E0	478	clr	acc. 0	; specify write operation
021B 120261	479	call	shout	; send device address
021E 4013	480	jc	x46	; abort if no acknowledge
	481			
0220 EB	482	mov	a, addr_hi	; send high byte of address
0221 120261	483	call	shout	

0224 400D	484	jc	x46	; abort if no acknowledge
	485			
0226 EA	486	mov	a,addr_lo	; send low byte of address
0227 120261	487	call	shout	
022A 4007	488	jc	x46	; abort if no acknowledge
	489			
	490			; Call Current Address Read function
	491			
022C E5F0	492	mov	a,b	; get programmable address
022E 31F3	493	call	read_current	
0230 020236	494	jmp	x47	;exit
	495	x46:		
0233 120253	496	call	stop	
	497	x47:		
0236 D0F0	498	pop	b	
0238 22	499	ret		
	500			
	501			
	502	start:		
	503			
	504			; Send START, defined as high - to - low SDA with SCL high
	505			; Return with SCL,SDA low
	506			; Returns CY set if bus is not available
	507			
0239 D293	508	setb	SDA	
023B D292	509	setb	SCL	
	510			
	511			; Verify bus available
	512			
023D 309311	513	jnb	SDA,x40	; jump if not high
0240 30920E	514	jnb	SCL,x40	; jump if not high
	515			
0243 00	516	nop		; enforce setup delay and cycle delay
0244 C293	517	clr	SDA	
0246 00	518	nop		; enfoce hold delay
0247 00	519	nop		
0248 00	520	nop		
0249 00	521	nop		
024A 00	522	nop		
024B C292	523	clr	SCL	
	524			
024D C3	525	clr	c	; clear error flag

024E 020252	526	jmp	x41	
	527	x40:		
0251 D3	528	setb	c	; set error flag
	529	x41:		
0252 22	530	ret		
	531			
	532			
	533	stop:		
	534			
	535			; Send STOP, defined as low-to-high SDA with SCL high
	536			; SCL expected low on entry. Return with SCL, SDA high.
	537			
0253 C293	538	clr	SDA	
0255 00	539	nop		; enforce SCL low and data setup
0256 00	540	nop		
0257 D292	541	setb	SCL	
0259 00	542	nop		; enforce setup delay
025A 00	543	nop		
025B 00	544	nop		
025C 00	545	nop		
025D 00	546	nop		
025E D293	547	setb	SDA	
0260 22	548	ret		
	549			
	550			
	551	shout:		
	552			
	553			; Shift out a byte to the AT24Cxx, most significant bit first
	554			; SCL, SDA expected low on entry. Return with SCL low
	555			; Called with data to send in A
	556			; Returns CY set to indicate failure by slave to acknowledge
	557			; Destroys A
	558			
0261 C0F0	559	push	b	
0263 75F008	560	mov	b, #8	; bit counter
	561	x42:		
0266 33	562	rlc	a	; move bit into CY
0267 9293	563	mov	SDA,c	; output bit
0269 00	564	nop		; enforce SCL low and data setup
026A D292	565	setb	SCL	; raise clock
026C 00	566	nop		; enforce SCL high

026D 00	567	nop		
026E 00	568	nop		
026F 00	569	nop		
0270 C292	570	clr	SCL	; drop clock
0272 D5F0F1	571	djnz	b, x42	; next bit
	572			
0275 D293	573	setb	SDA	; release SDA for ACK
0277 00	574	nop		; enforce SCL low and tAA
0278 00	575	nop		
0279 D292	576	setb	SCL	; raise ACK clock
027B 00	577	nop		; enforce SCL high
027C 00	578	nop		
027D 00	579	nop		
027E 00	580	nop		
027F A293	581	mov	c, SDA	; get ACK bit
0281 C292	582	clr	SCL	; drop ACK clock
	583			
0283 D0F0	584	pop	b	
0285 22	585	ret		
	586			
	587			
	588	shin;		
	589			
	590			; Shift in a byte from the AT24Cxx, most significant bit first
	591			; SCL expected low on entry. Return with SCL low
	592			; Returns received data byte in A
	593			
0286 D293	594	setb	SDA	; make SDA an input
	595			
0288 C0F0	596	push	b	
028A 75F008	597	mov	b, # 8	; bit count
	598	x43;		
028D 00	599	nop		; enforce SCL low and data setup
028E 00	600	nop		
028F 00	601	nop		
0290 D292	602	setb	SCL	; raise clock
0292 00	603	nop		; enforce SCL high
0293 00	604	nop		
0294 A293	605	mov	c, SDA	; input bit
0296 33	606	rlc	a	; move bit into byte
0297 C292	607	clr	SCL	; drop clock
0299 D5F0F01	608	djnz	b, x43	; next bit

	609				
029C D0F0	610	pop	b		
029E 22	611	ret			
	612				
	613				
	614	ACK:			
	615				
	616			; Clock out an acknowledge bit (low)	
	617			; SCL expected low on entry. Return with SCL, SDA low	
	618				
029F C293	619	clr	SDA		; ACK bit
02A1 00	620	nop			; enforce SCL low and setup
02A2 00	621	nop			
02A3 D292	622	setb	SCL		; raise clock
02A5 00	623	nop			; enforce SCL high
02A6 00	624	nop			
02A7 00	625	nop			
02A8 00	626	nop			
02A9 C292	627	clr	SCL		; drop clock
02AB 22	628	ret			
	629				
	630				
	631	NAK:			
	632				
	633			; Clock out a negative acknowledge bit (high)	
	634			; SCL expected low on entry. Return with SCL low, SDA high	
	635				
02AC D293	636	setb	SDA		; NAK bit
02AE 00	637	nop			; enforce SCL low and data setup
02AF 00	638	nop			
02B0 D292	639	setb	SCL		; raise clock
02B2 00	640	nop			; enforce SCL high
02B3 00	641	nop			
02B4 00	642	nop			
02B5 00	643	nop			
02B6 C292	644	clr	SCL		; drop clock
02B8 22	645	ret			
	646				
	647				
	648	END			

6.3.3 AT93C06/46 的接口方法

AT93C06/46 的内部结构如图 6-36 所示。它由存储器阵列、地址译码、数据寄存器、方式译码逻辑、时钟发生器、输出缓冲器等组成。AT93C06 的存储器阵列为 16×16 位，AT93C46 的存储器阵列为 64×16 位。输入数据由数据寄存器串行移位输入，输出数据由输出缓冲器串行移位输出。对于一些公司生产的 AT93C06/46，其封装的引脚 6 用于对内部存储器阵列进行组织，故此脚标志为 ORG。当 $ORG = 1$ 时，存储器阵列组织为 64×16 位；当 $ORG = 0$ 时，存储器阵列组织为 128×8 位。

在 89 系列单片机和 AT93C06/46 之间进行数据的传送必须符合 AT93C06/46 的总线接口要求。对于 AT93C06/46 来说，它们采用的是 3 线串行总线。3 线串行总线由 SK, DI, DO 这 3 条信号线组成，其中 SK 是串行时钟线，DI 是数据输入线，DO 是数据输出线。当然，所谓数据输入或输出是对 EEPROM 芯片而言，而不是对单片机而言的。

AT93C06/46 的 3 线串行总线接口时序如图 6-37 所示。在图中给出了时钟 SK、数据输入线 DI、数据输出线 DO 以及片选信号 CS 的波形以及技术参数。

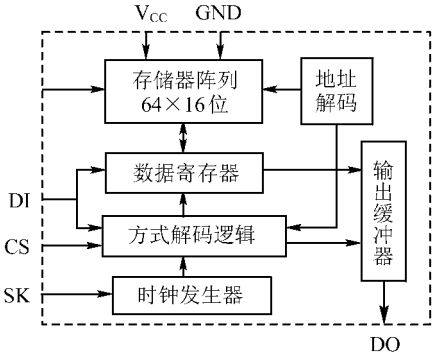


图 6-36 AT93C46 的内部结构框图

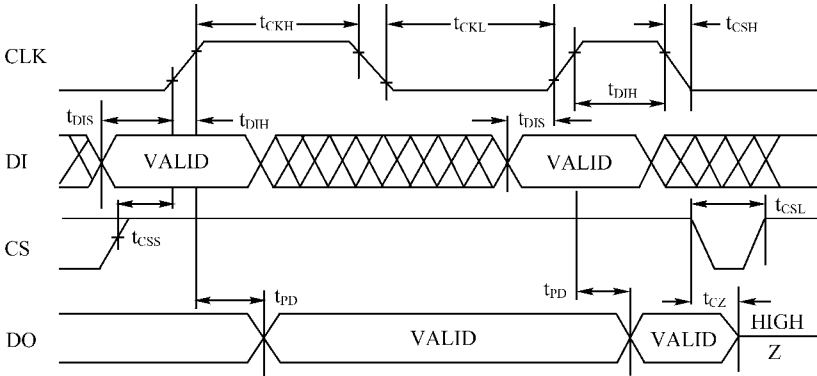


图 6-37 AT93C06/46 的数据时序

数据时序中有关技术参数有：数据输入建立时间 t_{DIS} ，片选建立时间 t_{CSS} ，数据输入保持时间 t_{DIH} ，时钟高电平时间 t_{CKH} ，时钟低电平时间 t_{CKL} ，数据输出延时时间 t_{PD} ，片选保持时间 t_{CSH} ，片选低电平时间 t_{CSL} ，数据输出屏蔽时间 t_{CZ} 。为了保证 EEPROM 读写过程的正确无误，对上述时间都有严格要求和限定。表 6-16 给出了相关技术参数的范围值。

89 系列单片机和 AT93C06/46 的接口可以采用 3 线连接或 4 线连接的方式。在 3 线连接中，把 DI 和 DO 以线或的方式接在一起，组成一条数据线 DI/DO，另外还有片选线 CS、串行时钟线 SK。一种典型的 3 线连接如图 6-38 所示。在图中，AT93C06/46 的引脚 ORG 接地，故它可以以 8 位数据的方式进行读写。

表 6-16 AT93C06/46 的技术参数

参 数	符 号	最 小	最 大	单 位	条 件
时钟频率	f_{CLK}		1	MHz	
时钟高电平时间	t_{CKH}	500		ns	
时钟低电平时间	t_{CKL}	500		ns	
片选建立时间	t_{CSS}	50		ns	
片选保持时间	t_{CSH}	0		ns	
片选低电平时间	t_{CSL}	100		ns	
数据输入建立时间	t_{DIS}	100		ns	
数据输入保持时间	t_{DIH}	100		ns	
数据输出延时时间	t_{PD}		400	ns	$CL=100\text{ pF}$
数据输出消失时间(来自 CS=low)	t_{CZ}	0	100	ns	$CL=100\text{ pF}$
数据输出消失时间(来自上一时钟)	t_{DDZ}	0	400	ns	$CL=100\text{ pF}$
状态有效时间	t_{SV}		100	ns	$CL=100\text{ pF}$
程序周期(自动擦和写)	t_{WC}		2 15	ms ms	for ERAL and WRAL
擦周期	t_{EC}		1	ms	
持续	—	100 000		E/W Cycles	

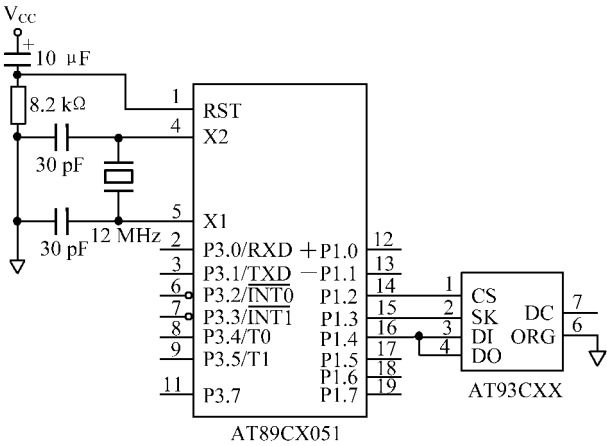


图 6-38 3 线连接的接口方法

89 系列单片机和 AT93C06/46 的接口方法也可以用 4 线连接方式。在 4 线连接中,数据输入线 DI 和数据输出线 DO 是各自独立的,再加上时钟线 SK、片选线 CS,则单片机 AT89C2051 须用 P1 端口中的 P1.2~P1.5 这 4 根 I/O 线与之对应相连,4 线相连的接口方法如图 6-39 所示。当然,在 4 线相连的接口中,也可以用单片机 I/O 端口的其他引脚和

AT93C06/46 的 DI,DO,SK,CS 端连接。在图 6-39 中,AT93C06/46 的 ORG 引脚接电源 V_{CC} ,所以,它以16 位数据方式进行读写。

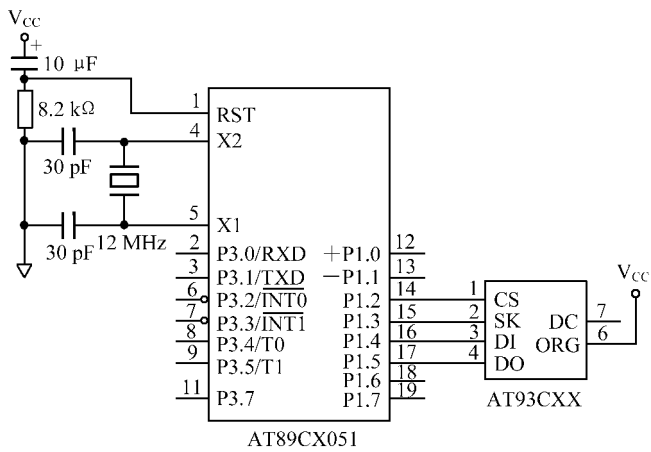


图 6-39 4 线连接的接口方法

在实际应用中,要实现对 AT93C06/46 的数据读写,最关键的有两点:一点是建立恰当的串行数据传送的硬件逻辑接口;另一点是用单片机的程序模拟 AT93C06/46 的数据传送时序。在一般的应用中,通常都采用 3 线串行总线结构。这种 3 线串行总线由串行时钟 SK、数据输入线 DI、数据输出线 DO 等 3 条信号线组成,并且按 AT93C06/46 工作的通信协议工作。所以,在单片机和 AT93C06/46 之间的连接采用 4 线连接的接口方法。因为除通信用 3 线串行总线之外,单片机和 AT93C06/46 之间还有一条连线,是用于单片机对 EEPROM 进行片选控制的,即片选 CS 线,从而组成了 4 线连接方法。

一个典型的 4 线连接方法和实际应用例子如图 6-40 所示。

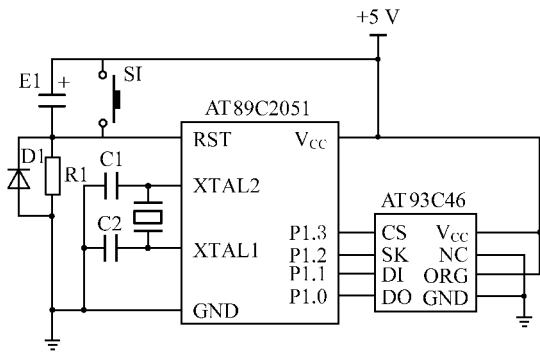


图 6-40 4 线连接的应用例子

在图 6-40 中,ORG 接高电平,故选择 16 位字的方式工作;而把 ORG 端接低电平时,会选择 8 位字的方式工作。在选择 8 位字的工作方式时,其指令格式中的地址部分要在高位多加 1 位 A6,这一点和 16 位字的工作方式有所区别。当然,在 8 位字的工作方式中,数据的串行写入和读出也是 8 位的,这一点也是必须加以注意的。考虑了 ORG 端之后的 AT93C06/46 的指令编码情况如表 6-17 所列。

表 6-17 考虑了 ORG 端的 AT93C06/46 指令

指 令	代 码 SB OP	地 址		数 据	
		x8	x16	x8	x16
READ 读	1 10	A6~A0	A5~A0		
EWEN 擦写使能	1 00	11xxxxx	11xxxx		
ERASE 擦除	1 11	A6~A0	A5~A0		
WRITE 写	1 01	A6~A0	A5~A0	D7~D0	D15~D0
ERAL 全擦	1 00	10xxxxx	10xxxx		
WRAL 全写	1 00	01xxxxx	01xxxx	D7~D0	D15~D0
EWDS 擦写禁止	1 00	00xxxxx	00xxxx		

因为 ORG 端接电源 V_{cc} 的 +5 V, 所以, AT93C46 就以 16 位数据方式工作, AT89C2051 中的工作程序就应模拟 AT93C46, 执行 16 位数据工作方式时的时序和传送协议。对 AT93C46 进行写入编程的软件流程如图 6-41 所示。

在图 6-41 所示的写入程序框图中, 采用了 R0, R1, R2 共 3 个暂时寄存器。这 3 个寄存器的定义和内容如下。

- (1) R0: 写入的数据量寄存器。它存放的是写入字节数。
- (2) R1: AT93C46 的写指令寄存器。它存放写指令的操作码“101”, 以及地址码 A5~A0。其中地址码是 AT93C46 内部存储阵列的地址。
- (3) R2: 被写数据的地址指针。这是 AT89C2051 内部待写入到 AT93C46 中去的数据的地址指针。

图 6-41 所示的程序流程框图所完成的功能, 就是把 R2 所指示的 AT89C2051 内部存储器地址中的数据, 写入到 R1 中地址部分所指明的 AT93C46 的存储单元中去。

由汇编语言编制的写入程序如下所示。它由写入主程序和写入子程序组成。写入子程序的标号为 SNDBYT。它用于执行一个字节的写入。主程序的功能主要是把数据从 AT89C2051 中取出, 并调用写入子程序 SNDBYT, 以完成写入操作。主程序还负责对数据量的控制。

写入程序的汇编语言源程序列出如下:

```
MOV      R0, #data
WRPROM: SETB  P1.3          ; 芯片选择
          SETB  P1.1
```

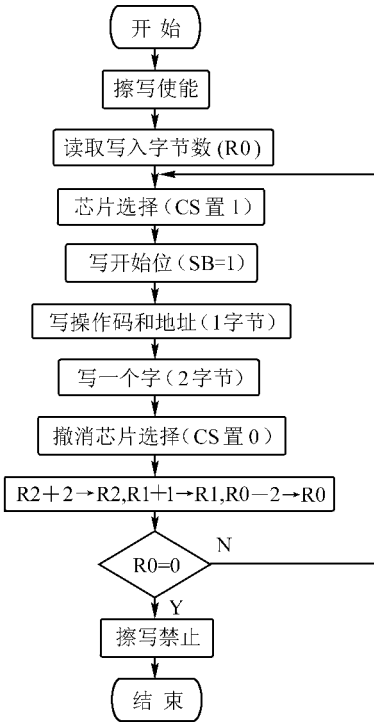


图 6-41 写程序框图

```

        CLR      P1.2          ; 写开始位
        SETB     P1.2
        MOV      A,R1
        MOV      R3,A
        LCALL    SNDBYT       ; 写操作码+起始地址
        MOV      A,@R2
        MOV      R3,A
        LCALL    SNDBYT       ; 写一个字
        INC      R2
        MOV      A,@R2
        MOV      R3,A
        LCALL    SNDBYT
        CLR      P1.3          ; 撤消芯片选择
        INC      R2
        INC      R1
        DEC      R0
        DJNZ     R0,WRPROM
        END
SNDBYT:  MOV      B,#8
        MOV      A,R3
S_LOOP:  RLC      A
        MOV      P1.1,C        ; 发送位
        CLR      P1.2          ; SK
        SETB     P1.2
        DJNZ     B,S_LOOP
        RET
```

在这个程序中,没有擦/写启动和擦/写屏蔽这两种操作。这两种指令操作分别应处于写入程序执行写入之前,以及数据写完之后。这两种指令操作的程序编制和上面给出的程序相似。因此,程序是十分简单的。这个程序和逻辑结构可以十分方便地用于各种没有串行接口总线的单片机系统中,并且实现了良好的相关数据的保存。特别在一些存在断电情况的工作过程中,存放状态数据十分有用。这种接口可以广泛用于家用电器、仪器仪表或各种生产过程的监视或控制之中。

参考图 6-39,下面的程序集给出了 AT89CX051 对 AT93CXX 系列 EEPROM 的读写操作控制。这个程序集有通用性,可以支持对 AT93CXX 系列的 8 位或 16 位读写。当单片机 AT89CX051 采用 12 MHz 时钟时,程序可以满足 AT93CXX 系列的时序要求。当时钟高于这个频率时,程序要加以修改。

程序可执行 AT93CXX 系列的 7 种指令,包括读、写、擦除、擦/写启动、擦/写屏蔽、全擦、全写这些指令。每条指令都有自己的入口地址。这些指令的调用过程如下。

1. 装入地址

当地址有 8 位或少于 8 位时,把地址装入寄存器对 ADDR-LO 中;当地址多于 8 位时,把地址装入寄存器对 ADDR-HI,ADDR-LO 中。

2. 装入数据

当数据是单字节时,把数据装入寄存器对 DATA - LO 中;当数据是双字节时,把数据装入寄存器对 DATA - HI,DATA - LO 中。

3. 调用指令

如果指令返回数据,当 SIZE=0 时,返回数据存放在 DATA - LO 中;当 SIZE=1 时,返回数据存放在 DATA - HI,DATA - LO 中。

有些指令,比如写、全写、擦除、全擦会返回进位标志,标志为“1”表示出错;标志为“0”表示成功。

当数据的位数不同时,地址的位数也有区别,AT93C46,AT93C56,AT93C57,AT93C66 的数据及地址的位数对应如表 6 - 18 所列。

表 6 - 18 AT93CXX 的数据及地址位数

型 号	AT93C46		AT93C56		AT93C57		AT93C66	
数据位数	8	16	8	16	8	16	8	16
地址位数	7	6	9	8	8	7	9	8

在下面的程序集中,括号 SIZE 用于表示数据是 8 位还是 16 位。这取决于 AT93CXX 的 ORG 引脚的状态。当 ORG=1 时,表示数据为 16 位;当 ORG=0 时,表示数据为 8 位。在 PSW 中的通用位 F0 用于 SIZE 标志,不过也可以用任何可寻址的位取代它。

AT93CXX 的信号线 CS,SK,DI,DO 和单片机的 I/O 引脚连接由程序的 73~76 行决定。在 3 线连接中,DI 和 DO 连到同一条 I/O 引脚;在 4 线连接中,则 DI 和 DO 分别连到不同的 I/O 引脚。但不需要改变程序其他代码,只要 75~76 行的内容即可。

在图 6 - 39 中,AT89C2051 的 P1.4~P1.7 分别和 AT93CXX 的 CS,SK,DI,DO 端相连。而在这个程序中从程序行 73~76 中看出:是把 AT89C2051 的 P1.2~P1.5 分别定义给 CS,SK,DI,DO 的,这一点要加以注意。如果要令其一致,可修改程序行 73~76 的内容。

程序的清单如下:

```
LOC OBJ      LINE      SOURCE
                    51
                    52          ; Number of valid address bits for the device in use (reference table
                                above)
                    53
0007          54          ; NADDR EQU    7          ; e. g. AT93C46 strapped for 8-bit data
                    55          ; NADDR EQU    6          ; e. g. AT93C46 strapped for 16-bit data
                    56
                    57          ; Data size
                    58
00D5          59          size      EQU    f0          ; 0=8-bit data, 1=16-bit data
                    60
                    61          ; Register definitions
                    62
```

REG	63	data_lo	EQU	r0	; register for byte data
REG	64	data_hi	EQU	r1	; extension register for word data
REG	65	addr_lo	EQU	r2	; register for short address
REG	66	addr_hi	EQU	r3	; extension register for long address
	67				
	68				; Microcontroller connections to AT93Cxx control pins
	69				; For a three-wire interface, define DI and DO to be the same pin
	70				; For a four-wire interface, define DI and DO to be different pins
	71				; No other code changes are required
	72				
0092	73	CS	BIT	p1.2	; AT93Cxx chip select
0093	74	SK	BIT	p1.3	; serial data clock
0094	75	DI	BIT	p1.4	; serial data input
0095	76	DO	BIT	p1.5	; serial data output
	77				
	78				
	79			DSEG AT 20H	
	80				
0020	81		ORG	20H	; stack origin
0020	82	stack:	DS	20H	; stack depth
	83				
	84				
	85			CSEG	
	86				
0000	87		ORG	0000H	; power on/reset vector
0000 020080	88		jmp	on_reset	
	89				
0003	90		ORG	0003H	; external interrupt 0 vector
0003 32	91		reti		; undefined
	92				
000B	93		ORG	000BH	; timer 0 overflow vector
000B 32	94		reti		; undefined
	95				
0013	96		ORG	0013H	; external interrupt 1 vector
0013 32	97		reti		; undefined
	98				
001B	99		ORG	001BH	; timer 1 overflow vector
001B 32	100		reti		; undefined
	101				
0023	102		ORG	0023H	; serial I/O interrupt vector
0023 32	103		reti		; undefined
	104				
0080	105		ORG	0080H	; begin code space

	106	USING 0	; register bank zero
	107	on_reset:	
0080 75811F	108	mov sp, # (stack-1)	; initialize stack pointer
	109		
	110		; Initialize AT93Cxx control lines
	111		
0083 C292	112	clr CS	; low
0085 C293	113	clr SK	; low
0087 D294	114	setb DI	; high
0089 D295	115	setb DO	; high
	116		
	117		; Initialize SIZE flag
	118		
008B C2D5	119	clr size	; specify 8-bit data
	120	setb size	; specify 16-bit data
	121		
	122		
	123	; Sample function calls	
	124		
	125		
	126		; Read (byte if SIZE/ORG=0, word if SIZE/ORG=1)
	127		
008D 7A00	128	mov addr_lo, # 0	; address
008F 1200BC	129	call read	
	130		
	131		; Write byte (SIZE/ORG=0)
	132		
0092 7A7F	133	mov addr_lo, # 7fh	; address
0094 7855	134	mov data_lo, # 55h	; data
0096 1200D7	135	call write	; check carry after return
	136		
	137		; Write word (SIZE/ORG=1)
	138		
0099 7A3F	139	mov addr_lo, # 3fh	; address
009B 7855	140	mov data_lo, # 55h	; data
009D 79AA	141	mov data_hi, # 0aah	; data
009F 1200D7	142	call write	; check carry after return
	143		
	144		; Fill with byte (SIZE/ORG=0)
	145		
00A2 7855	146	mov data_lo, # 55h	; data
00A4 120105	147	call wral	; check carry after return
	148		

	149			; Fill with word (SIZE/ORG=1)
	150			
00A7 7855	151	mov	data_lo, # 55h	; data
00A9 79AA	152	mov	data_hi, # 0aah	; data
00AB 120105	153	call	wral	; check carry after return
	154			
	155			; Erase address (byte if SIZE/ORG=0, word if SIZE/ORG=1)
	156			
00AE 7A00	157	mov	addr_lo, # 0	; address
00B0 120129	158	call	erase	; check carry after return
	159			
	160			; Erase device
	161			
00B3 120144	162	call	eral	; check carry after return
	163			
	164			; Enable write
	165			
00B6 120155	166	call	ewen	
	167			
	168			; Disable write
	169			
00B9 120163	170	call	ewds	
	171			
	172			
	173	read;		
	174			; Read data from addressed byte or word
	175			; Called with address in register ADDR_HI; ADDR_LO
	176			; Returns data in register DATA_LO if SIZE flag is low,
	177			; in register pair DATA_HI; DATA_LO if SIZE flag is high
	178			; Destroys B, DPTR
	179			
00BC D292	180	setb	CS	; raise CS
	181			
	182			; Shift out start bit (1) and op code (10)
	183			
00BE 758206	184	mov	dpl, # 110b	
00C1 75F003	185	mov	b, # 3	
00C4 1201B7	186	call	outdata	
	187			
	188			; Shift out address, MSB first
	189			
00C7 8A82	190	mov	dpl, addr_lo	; low byte
00C9 8B83	191	mov	dph, addr_hi	; high byte (may be undefined)

00CB 75F007	192	mov	b, # NADDR	; bit count
00CE 1201B7	193	call	outdata	
	194			
	195			; Read output data
	196			
00D1 120194	197	call	indata	; get return data
	198			
00D4 C292	199	clr	CS	; drop CS
00D6 22	200	ret		
	201			
	202			
	203	write:		
	204			; Write data into the addressed byte or word
	205			; ERASE is not required before WRITE
	206			; Called with address in registers ADDR_HI; ADDR_LO,
	207			; data in registers DATA_HI; DATA_LO
	208			; SIZE flag low indicates valid data in DATA_LO only;
	209			; high indicates valid data in both DATA_HI and DATA_LO
	210			; Returns CY set to indicate error. Destroys B, DPTR
	211			
00D7 D292	212	setb	CS	; raise CS
	213			
	214			; Shift out start bit (1) and op code (01)
	215			
00D9 758205	216	mov	dpl, #101b	
00DC 75F003	217	mov	b, #3	
00DF 1201B7	218	call	outdata	
	219			
	220			; Shift out address, MSB first
	221			
00E2 8A82	222	mov	dpl, addr_lo	; low byte
00E4 8B83	223	mov	dph, addr_hi	; high byte (may be undefined)
00E6 75F007	224	mov	b, # NADDR	; bit count
00E9 1201B7	225	call	outdata	
	226			
	227			; Shift out data, MSB first
	228			; Number of bytes varies according to SIZE flag
	229			
00EC 8882	230	mov	dpl, data_lo	
00EE 20D506	231	jb	size, ee61	
00F1 75F008	232	mov	b, #8	
00F4 0200FC	233	jmp	ee62	
	234	ee61:		

00F7 8983	235	mov	dph,data_hi
00F9 75F010	236	mov	b,#16
	237	ee62;	
00FC 1201B7	238	call	outdata
	239		
00FF C292	240	clr	CS ; drop CS
	241		
	242		; Poll device status
	243		
0101 120171	244	call	status ; returns CY
0104 22	245	ret	
	246		
	247		
	248	wral;	
	249		; Write data into all addresses (fill)
	250		; Valid only when Vcc is 5.0 Volts +/- 10 percent
	251		; Called with data in registers DATA_HI;DATA_LO
	252		; SIZE flag low indicates valid data in DATA_LO only
	253		; high indicates valid data in both DATA_HI and DATA_LO
	254		; Returns CY set to indicate error. Destroys B, DPTR
	255		
0105 D292	256	setb	CS ; raise CS
	257		
	258		; Shift out start bit (1), op code (00) and 01
	259		; followed by a device-specific number of undefined bits
	260		
0107 900220	261	mov	dptr,#(10001b SHL (NADDR-2))
010A 75FF0A	262	mov	b,#(NADDR+3)
010D 1201B7	263	call	outdata
	264		
	265		; Shift out data, MSB first
	266		; Number of bytes varies according to SIZE flag
	267		
0110 8882	268	mov	dpl,data_lo
0112 20D506	269	jb	size, ee71
0115 75F008	270	mov	b,#8
0118 020120	271	jmp	ee72
	272	ee71;	
011B 8983	273	mov	dph,data_hi
011D 75F010	274	mov	b,#16
	275	ee72;	
0120 1201B7	276	call	outdata
	277		

0123 C292	278	clr	CS	; drop CS
	279			
	280			; Poll device status
	281			
0125 120171	282	call	status	; returns CY
0128 22	283	ret		
	284			
	285			
	286	erase:		
	287			; Erase addressed location
	288			; An erased byte reads as all ones (FF hex)
	289			; Called with address in registers ADDR_HI:ADDR_LO
	290			; Returns CY set to indicate error. Destroys B, DPTR
	291			
0129 D292	292	setb	CS	; raise CS
	293			
	294			; Shift out start bit (1) and op code (11)
	295			
012B 900007	296	mov	dptr, # 111b	
012E 75F003	297	mov	b, # 3	
0131 1201B7	298	call	outdata	
	299			
	300			; Shift out address, MSB first
	301			
0134 8A82	302	mov	dpl,addr_lo	; low byte
0136 8B83	303	mov	dph,addr_hi	; high byte (may be undefined)
0138 75F007	304	mov	b, # NADDR	; bit count
013B 1201B7	305	call	outdata	
	306			
013E C292	307	clr	CS	; drop CS
	308			
	309			; Poll device status.
	310			
0140 120171	311	call	status	; returns CY
0143 22	312	ret		
	313			
	314			
	315	eral:		
	316			; Erase all addresses
	317			; All data will read as all ones (FF hex)
	318			; Valid only when Vcc is 5.0 Volts +/- 10 percent
	319			; Returns CY set to indicate error. Destroys B, DPTR
	320			

0144 D292	321	setb	CS	; raise CS
	322			
	323			; Shift out start bit (1), op code (00) and 10
	324			
0146 900240	325	mov	dptr, # (10010b SHL (NADDR-2))	
0149 75F00A	326	mov	b, # (NADDR+3)	
014C 1201B7	327	call	outdata	
	328			
014F C292	329	clr	CS	; drop CS
	330			
	331			; Poll device status
	332			
0151 120171	333	call	status	; returns CY
0154 22	334	ret		
	335			
	336			
	337	ewen:		
	338			; Enable erase/write
	339			; Erase/write remains enabled until disabled with EWDS
	340			; Returns nothing. Destroys B, DPTR
	341			
0155 D292	342	setb	CS	; raise CS
	343			
	344			; Shift out start bit (1), op code (00) and 11
	345			
0157 900260	346	mov	dptr, # (10011b SHL (NADDR-2))	
015A 75F00A	347	mov	b, # (NADDR+3)	
015D 1201B7	348	call	outdata	
	349			
0160 C292	350	clr	CS	; drop CS
0162 22	351	ret		
	352			
	353			
	354	ewds:		
	355			; Disable erase/write
	356			; Returns nothing. Destroys B, DPTR
	357			
0163 D292	358	setb	CS	; raise CS
	359			
	360			; Shift out start bit (1), op code (00) and 00
	361			
0165 900200	362	mov	dptr, # (10000b SHL (NADDR-2))	
0168 75F00A	363	mov	b, # (NADDR+3)	

016B 1201B7	364	call	outdata	
	365			
016E C292	366	clr	CS	; drop CS
0170 22	367	ret		
	368			
	369			
	370	status;		
	371			; Read device status
	372			; The device output is sampled every 50 microseconds until timeout
	373			; or ready. Allows a total of approximately 10 milliseconds for
	374			; the device to signal ready before timing out
	375			; The DO pin is allowed to float before status polling begins
	376			; Returns CY cleared to indicate ready, set to indicate timeout
	377			; All registers preserved (except flags)
	378			
0171 C0F0	379	push	b	
	380			
0173 D295	381	setb	D0	; float pin
0175 D292	382	setb	CS	; select device
	383			
0177 75F0DC	384	mov	b, # 220	; 220 * 50 μ s=11 ms
	385			
	386			; Delay 50 microseconds
	387	ee40;		
017A C0F0	388	push	b	; 2 μ s
017C 75F016	389	mov	b, # 22	; 2 μ s
017F D5F0FD	390	djnz	b, \$	; 2 μ s * 22
0182 D0F0	391	pop	b	; 2 μ s
	392			
0184 209507	393	jb	DO, ee41	; jump if DO is high (READY)
	394			
0187 D5F0F0	395	djnz	b, ee40	; next pass
018A D3	396	setb	c	; flag error
018B 02018F	397	jmp	ee42	; return
	398	ee41;		
018F C3	399	clr	c	; clear error flag
	400	ee42;		
018F C292	401	clr	CS	; deselect device
	402			
0191 D0F0	403	pop	b	
0193 22	404	ret		
	405			
	406			

	407	indata;	
	408	;	Fetch data from the AT93Cxx
	409	;	The AT93Cxx family presents data most significant bit first
	410	;	The bit stream is assembled into bytes by function SHIN
	411	;	The CS pin is controlled by the caller
	412	;	The DO pin is allowed to float before data is read
	413	;	If the SIZE flag is cleared, one byte is returned in register
	414	;	DATA_LO. If the SIZE flag is set, two bytes are returned in
	415	;	register pair DATA_HI:DATA_LO
	416	;	Destroys A
	417		
0194 D295	418	setb DO	; float pin
	419		
	420	;	Read and assemble first data byte
	421		
0196 1201A2	422	call shin	
0199 30D504	423	jnb size, ee21	; jump if data is only one byte wide
	424		
019C F9	425	mov data_hi, a	; save high byte
	426		
	427	;	Read and assemble second data byte
	428		
019D 1201A2	429	call shin	
	430	ee21;	
01A0 F8	431	mov data_lo, a	; save low byte
01A1 22	432	ret	
	433		
	434		
	435	shin;	
	436	;	Shift in eight bits from the AT93Cxx, most significant bit first
	437	;	and assemble them into a byte. On return, the SK pin is left low
	438	;	Returns assembled byte in A
	439		
01A2 C0F0	440	push b	
	441		
01A4 C293	442	clr SK	; drop clock
01A6 75F008	443	mov b, # 8	; init loop/delay min one μ s
	444	ee30;	
01A9 D293	445	setb SK	; raise clock
01AB 00	446	nop	; delay min one μ s
01AC A295	447	mov c, DO	; read serial data output
01AE 33	448	rlc a	; shift in bit/delay min one μ s
01AF C293	449	clr SK	; drop clock

```

01B1 D5F0F5    450          djnz      b,ee30      ; next bit/delay min one  $\mu$ s
                451
01B4 D0F0      452          pop       b
01B6 22         453          ret
                454
                455
                456          outdata;
                457          ; Message data and move it out to the AT93Cxx
                458          ; This function accepts from one to sixteen bits of data and
                459          ; formats it to be shifted out to the AT93Cxx by function SHOUT
                460          ; The CS pin is controlled by the caller
                461          ; On entry, expects from one to sixteen bits of data in DPTR,
                462          ; right-justified, and a count of the number of valid bits in B
                463          ; Returns nothing. Destroys A, DPTR
                464
01B7 C0F0      465          push      b
                466
01B9 E5F0      467          mov       a,b          ; get bit count
01BB C3         468          clr        c          ; compare count to eight
01BC 9408       469          subb      a,#8
01BE 4025       470          jc         ee6          ; jump if count less than eight
01C0 601E       471          jz         ee5          ; jump if count equals eight
                472
                473          ; Bit count greater than eight
                474          ; Use result of first subtraction in compares
                475
01C2 F5F0      476          mov       b,a          ; save adjusted bit count
01C4 C3         477          clr        c          ; compare adjusted count to eight
01C5 9408       478          subb      a,#8
01C7 4007       479          jc         ee2          ; jump if count less than eight
01C9 7027       480          jnz      ee9          ; jump if count greater than eight
                481
                482          ; Bit count equals eight
                483
01CB E583      484          mov       a, dph
01CD 0201DA     485          jmp        ee4
                486
                487          ; Bit count less than eight
                488          ; Use count to position data
                489          ee2:
01D0 C0F0      490          push      b          ; save count
01D2 E583      491          mov       a, dph      ; get data
                492          ee3:

```

01D4 03	493	rr	a	; shift data
01D5 D5F0FC	494	djnz	b, ee3	
01D8 D0F0	495	pop	b	; restore count
	496	ee4:		
01DA 1201F7	497	call	shout	
	498			
	499			; Did high byte. Do low byte, count equals eight
	500			
01DD 75F008	501	mov	b, #8	
	502			
	503			; Bit count equals eight
	504	ee5:		
01E0 E582	505	mov	a, dpl	
01E2 0201EF	506	jmp	ee8	
	507			
	508			; Bit count less than eight
	509			; Use count to position data
	510	ee6:		
01E5 C0F0	511	push	b	; save count
01E7 E582	512	mov	a, dpl	; get data
	513	ee7:		
01E9 03	514	rr	a	; shift data
01EA D5F0FC	515	djnz	b, ee7	
01ED D0F0	516	pop	b	; restore count
	517	ee8:		
01EF 1201F7	518	call	shout	
	519	ee9:		
01F2 D295	520	setb	DO	; leave pin floating
	521			
01F4 D0F0	522	pop	b	
01F6 22	523	ret		
	524			
	525			
	526	shout:		
	527			; Shift out a byte to the AT93Cxx, most significant bit first
	528			; On return, the SK pin is left low
	529			; Returns nothing. Destroys A, B
	530			
	531	ee50:		
01F7 C293	532	clr	SK	; drop clock
01F9 33	533	rlc	a	; move bit into CY
01FA 9294	534	mov	DI, c	; output bit
01FC 00	535	nop		; delay min 400 ns

01FD D293	536	setb	SK	; raise clock
01FF D5F0F5	537	djnz	b,ee50	; next bit/delay min one μ s
	538			
0202 C293	539	clr	SK	; drop clock
0204 22	540	ret		
	541			
	542			
	543	END		

SYMBOL TABLE LISTING

NAME	TYPE	VALUE	ATTRIBUTES
ADDR_HI	REG	R3	
ADDR_LO	REG	R2	
B	D	ADDR 00F0H	A
CONTROL_AT93CXX	----	-----	
CS	B	ADDR 0090H	2 A
DATA_HI	REG	R1	
DATA_LO	REG	R0	
DI	B	ADDR 0090H	4 A
DO	B	ADDR 0090H	5 A
DPH	D	ADDR 0083H	A
DPL	D	ADDR 0082H	A
EE2	C	ADDR 01D0H	A
EE21	C	ADDR 01A0H	A
EE3	C	ADDR 01D4H	A
EE30	C	ADDR 01A9H	A
EE4	C	ADDR 01DAH	A
EE40	C	ADDR 017AH	A
EE41	C	ADDR 018EH	A
EE42	C	ADDR 018FH	A
EE5	C	ADDR 01E0H	A
EE50	C	ADDR 01F7H	A
EE6	C	ADDR 01E5H	A
EE61	C	ADDR 00F7H	A
EE62	C	ADDR 00FCH	A
EE7	C	ADDR 01E9H	A
EE71	C	ADDR 011BH	A
EE72	C	ADDR 0120H	A
EE8	C	ADDR 01EFH	A
EE9	C	ADDR 01F2H	A
ERAL	C	ADDR 0144H	A
ERASE	C	ADDR 0129H	A

EWDS	C	ADDR	0163H	A
EWEN	C	ADDR	0155H	A
F0	B	ADDR	00D0H.5	A
INDATA	C	ADDR	0194H	A
NADDR		NUMB	0007H	A
ON_RESET	C	ADDR	0080H	A
OUTDATA	C	ADDR	01B7H	A
P1	D	ADDR	0090H	A
READ	C	ADDR	00BCH	A
SHIN	C	ADDR	01A2H	A
SHOUT	C	ADDR	01F7H	A
SIZE	B	ADDR	00D0H.5	A
SK	B	ADDR	0090H.3	A
SP	D	ADDR	0081H	A
STACK	D	ADDR	0020H	A
STATUS	C	ADDR	0171H	A
WRAL	C	ADDR	0105H	A
WRITE	C	ADDR	00D7H	A

6.4 数据存储器的串行扩展技术

单片机受到各领域的用户广泛应用。用其构成的简单系统方案是单片 CPU 加外围应用电路,一般不再扩充程序和数据存储器。此类方案设计成的工业仪表和工业控制器,其总线内置,抗干扰性能大为提高,但其 CPU 内部 RAM 空间小,AT89C51 为 128 字节,AT89C52 为 256 字节,这就限制了此方案的应用范围,只能应用于处理数据量比较小的地方。对于现在比较流行的高级编程语言 C,PLM 等就更显得数据区太小,特别在设计无纸记录仪时,要用大空间的数据存储器,一般为 1 MB,在此情况下,要进行数据区扩展。按传统的并行方法就得占用 CPU 的 P0 和 P2 以及部分 P3 的宝贵端口 I/O 线资源,而状态显示、键盘操作、A/D 转换等所需要的 I/O 口往往还要再进行扩充,其结果是电路结构复杂,总线外置,抗干扰性能差。解决这个问题可采用串行扩展数据存储的方法。现有的串行 EEPROM 如 93,24 系列等,其特点是占用 CPU I/O 口线比较少,如 3 线或 4 线,但其读写次数有限,且擦写时间长,一般写一字节 4 ms 左右,存储空间比较小,一般为几 K 位。它对于存储仪表重要参数很适用,但用作 CPU 数据区,频繁进行读写,显然不适合。为此,可用常规的 TTL 电路设计一个通用的串行扩展 RAM 的电路,以解决外部大容量的数据存储问题。

6.4.1 数据存储器的串行扩展逻辑电路

电路如图 6-42 所示,U1,U2,U3,U4 为 4 片 74LS164,采用级连的方法,将 CPU 输出串行的数据变成并行数据,U1,U2,U3 输出端为 SRAM 提供 20 线地址和部分控制信号线。U3 的 Q5,Q6,Q7 分别接 U7(RAM)的 WR,RD 和 U5 的 OE 端,此 3 位是读写操作的控制码。U4 输出为 CPU 写入 RAM 的数据,经 U5 锁存后送给 RAM 的数据线。在读数据的过程中,

U3 的 Q7 为高电平,即 U5 的 OE 为高电平,使 U5 的输出为高阻状态,从而使 U4 和 U7 隔离,避免发生竞争。DATA 为 CPU 串行输出的数据端;CLK 为输出和输入的时钟;CE 是 U7 的 E 端,初始状态为高电平,它控制数据读出和写入 RAM;LOAD 为 U6 的 PL 端,用它控制从 RAM 读出的数据并行装入 U6,CPU 再串行读入。

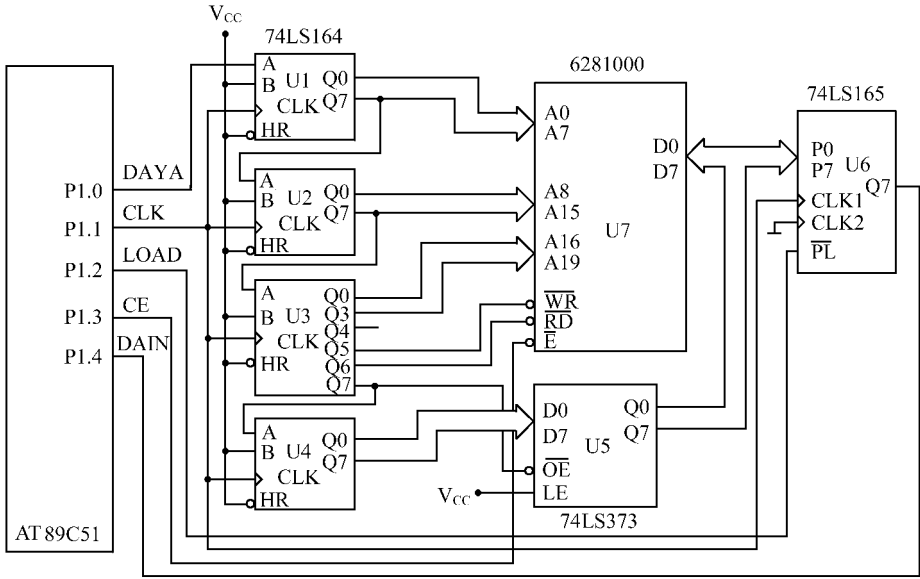


图 6-42 数据存储器串行扩展电路

下面对有关电路进行介绍。

1. 串-并行移位寄存器芯片 74LS164

在图 6-42 中,移位寄存器 74LS164 用于数据、控制信号和地址的串行输入及并行输出。由于有 4 片 74LS164,故组成 32 位信息。

74LS164 是高速 8 位串行输入、并行输出移位寄存器。它的内部逻辑结构如图 6-43 所示,封装引脚如图 6-44 所示。

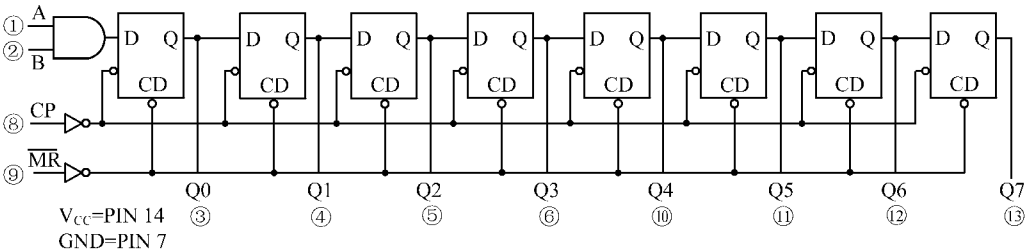


图 6-43 74LS164 的逻辑结构

74LS164 有两个输入端 A,B。这两个输入端的逻辑关系是“与”操作。通常,一端用作数据串行输入,另一端则用于控制电平的输入。当控制电平为“1”时,允许数据串行输入;当控制电平为“0”时,封锁数据的串行输入。在允许数据输入时,是由时钟信号 CP 的上升沿同步的。有时,为了方便则把 A,B 两个输入端一个接高电平或电源 V_{cc} ,另一个用于串行数据输入;有时,也可以把两个输入端 A,B 连接在一起,并用于数据串行输入。

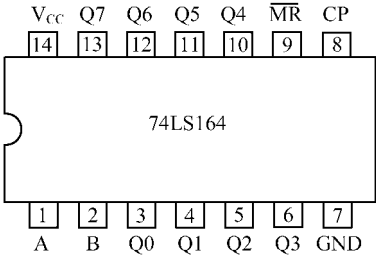


图 6-44 74LS164 引脚

74LS164 有 8 个输出端,并且称为 $Q_0 \sim Q_7$ 。它们是串行输入的 8 位数据的并行输出状态。在逻辑结构上,74LS164 内部由 8 个 D 寄存器进行串行连接组成。 $Q_0 \sim Q_7$ 则是 8 个 D 寄存器的输出。输入端 A,B 所串行的数据首先从最左边的 D 寄存器的 D 端输入,在时钟 CP 上升沿来时接收到最左边的 D 寄存器中,并可在 Q_0 输出。最左边 D 寄存器的输出端 Q_0 接第二个 D 寄存器的 D 输入端。在时钟 CP 上升沿来时,最左边的 D 寄存器接收输入的串行数据 1

位;而第二个 D 寄存器则接收最左边 D 寄存器的原来状态……在 8 个时钟之后,74LS164 就寄存了一个 8 位数据。

74LS164 还有一个主复位控制端 $\overline{MR}$ (Master Reset)。当 $\overline{MR}=0$,即低电平时,强行把所有的 8 个 D 寄存器清 0。在复位后, $Q_0 \sim Q_7$ 输出的状态全为“0”。

74LS164 的工作方式真值表如表 6-19 所列。

表 6-19 74LS164 工作方式

工作方式	输 入			输 出	
	$\overline{MR}$	A	B	Q_0	$Q_1 \sim Q_7$
复位	L	X	X	L	L-L
移位	H	l	l	L	$q_0 \sim q_6$
	H	l	h	L	$q_0 \sim q_6$
	H	h	l	L	$q_0 \sim q_6$
	H	h	h	H	$q_0 \sim q_6$

注: L(l)=LOW Voltage Levels;H(h)=HIGH Voltage Levels;X=Don't Care。

2. 8 位锁存器 74LS373

74LS373 是具有 3 态输出的 8 位锁存器,主要用于计算机的总线系统。在数据/地址复用 时,74LS373 可以用于存放地址或数据。

74LS373 的内部逻辑结构如图 6-45 所示。在图中看出,74LS373 执行 8 位并行输入和 8 位并行输出,其输入端为 $D_0 \sim D_7$,输出端为 $O_0 \sim O_7$ 。另外,还有控制信号锁存启动 LE,输出 启动 $\overline{OE}$ 。74LS373 的封装引脚如图 6-46 所示。

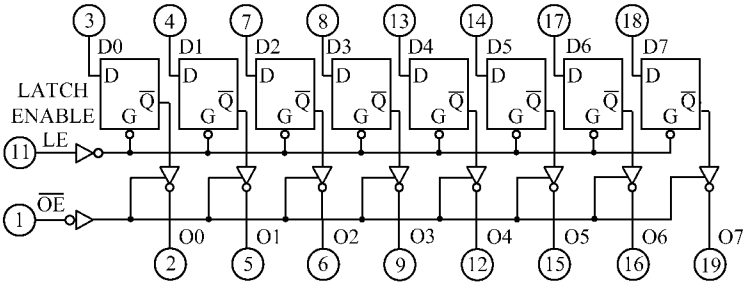


图 6-45 74LS373 逻辑结构

当锁存启动信号 $\overline{LE}$ 为高电平时,输入数据透明传送到输出端,即有 $O_n=D_n, n=0\sim7$ 。当锁存启动信号 $\overline{LE}$ 为低电平时,则不透明。不过,在锁存启动信号从高电平变低电平的下降沿时锁存 D 端的输入数据。要实现数据的透明传送,还和输出启动信号 $\overline{OE}$ 有关,当 $\overline{OE}=0, \overline{LE}=1$ 时,则实现透明传送。当 $\overline{OE}=1$ 时,不管 $\overline{LE}$ 状态如何,输出 O_n 为高阻抗。

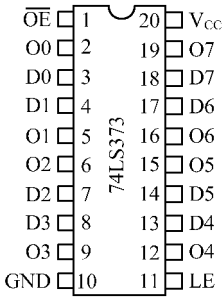


图 6-46 74LS373 引脚

3. 并-串行 8 位移位寄存器 74LS165

74LS165 由 8 个 R-S 触发器组成。它可以进行并行输入或串行输入,并且以串行方式输出。串行输出有两个端,一个输出原码,一个输出反码。

74LS165 的逻辑结构如图 6-47 所示,封装引脚如图 6-48 所示。

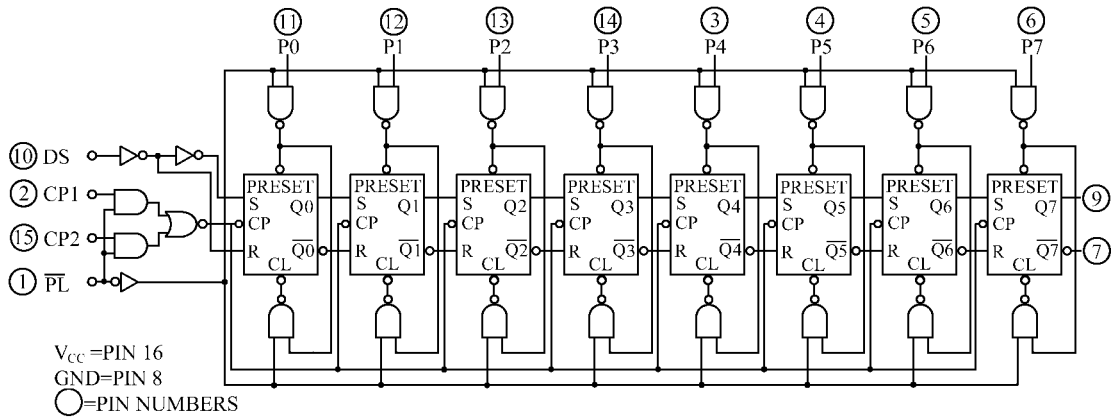


图 6-47 74LS165 的逻辑结构

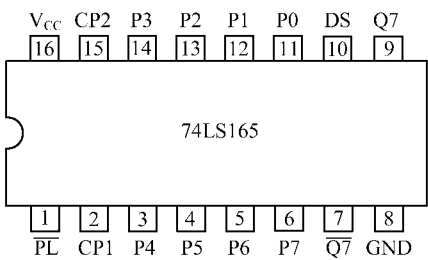


图 6-48 74LS165 的引脚

74LS165 的输入端有两种,一种是并行输入端 $P_0\sim P_7$,另一种是串行输入端 DS。

74LS165 的输出端有 $Q_7, \overline{Q7}$ 两个,它输出的是串行输出数据的原码 Q_7 、反码 $\overline{Q7}$ 。

74LS165 有两种控制信号,一种信号是时钟信号 CP_1, CP_2 ,另外一种是并行装入信号 $\overline{PL}$ 。时钟信号 CP_1, CP_2 的上升沿把串行输入端 DS 的数据串行移入 74LS165。并行装入信号 $\overline{PL}$ 在低电平时把并行输入端 $P_0\sim P_7$ 的数据并行装入 74LS165 中。

在把并行装入的数据实现串行输出时,只要发时钟信号 CP_1, CP_2 即可,当然,这时要求 $\overline{PL}$ 为高电平。

74LS165 工作真值表如表 6-20 所列。

表 6-20 74LS165 工作真值表

PL	CP		内 容								功 能
	1	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
L	X	X	P0	P1	P2	P3	P4	P5	P6	P7	并行输入
H	L		DS	Q0	Q1	Q2	Q3	Q4	Q5	Q6	右移
H	H		Q0	Q1	Q2	Q3	Q4	Q5	Q5	Q7	不变
H		L	DS	Q0	Q1	Q2	Q3	Q4	Q5	Q6	右移
H		H	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	不变

注：H=HIGH Voltage Level;L=LOW Voltage Level;X=Immaterial。

4. 存储器 6281000

6281000 是 1 MB 的随机存储器,作为 89 系列单片机的外部数据存储器。对 6281000 的读写接口采用串行方式,故接口上用 89 系列单片机的资源很少,只有 5 条 I/O 引脚。6281000 的地址为 20 位,为 A0~A19;数据为 8 位,数据总线为 D0~D7;写入信号为 $\overline{WR}$;读出信号为 $\overline{RD}$;片选信号为 $\overline{E}$ 。

6.4.2 读/写操作及工作程序

图 6-42 中所示的数据存储器扩展电路中,是依靠单片机 AT89C51 的程序控制实施对随机存储器 6281000 的读出和写入的。下面先对读写过程进行介绍,再给出其工作程序。

1. 数据存储器的数据写入

对数据存储器 6281000 进行串行写入时,要严格按逻辑电路约定串行送入接口电路的信息格式。送入接口电路的信息格式如下:

“8 位数据,010X,20 位地址”

即先把要写入的 8 位数据串行移入用 4 片 74LS164 连接的 32 位串行寄存器中,接着把 4 位控制码“010X”移入 32 位串行寄存器,最后再把 20 位的存储单元地址移入 32 位串行寄存器。这时,这 32 位信息在接口电路中的分布如下:

第 4 片 74LS164,即 U4 中存放了需写入的 8 位数据。

第 3 片 74LS164,即 U3 中的高 4 位 Q7~Q4 存放了控制码“010X”,低 4 位 Q3~Q0 存放了数据存储器的高 4 位地址 A19~A16。

第 2 片及第 1 片 74LS164,即 U2,U1 存放了数据存储器的低 16 位地址 A15~A0。

由于控制码为“010X”,并且分别从 U3 的 Q7~Q4 输出,即有 Q7=0,Q6=1,Q5=0。从图 6-42 的连线可知:U5 即 74LS373 的 $\overline{OE}=0$,故 74LS373 处于透明状态,U4 即 74LS164 所存放的 8 位数据通过 U5 直接送到数据存储器 6281000 的数据总线 D0~D7 上。同时 $\overline{RD}=1$,不能执行读出操作;而有 $\overline{WR}=0$,则 6281000 可执行写入操作。这时,只要 AT89C51 在 P1.3 输出低电平,令 6281000 的 $\overline{E}=0$,则 6281000 进入写操作过程。经过一段时间,AT89C51 在 P1.3 输出高电平,令 6281000 的 $\overline{E}=1$,则停止工作,写入过程完毕。

2. 数据存储器的数据读出

在数据读出时,由于不需对存储器送入数据,所以这时串行送入接口电路的信息格式如下。

“101X,20 位地址”

即送入串行接口电路中的是 24 位的信息,它包括控制码“101X”和地址码 20 位两个部分。这时,首先把“101X”送入 74LS164 中,接着把 20 位地址移位送入 74LS164 中。结果,U3 的高 4 位存放了“101X”;而 U3 的低 4 位存放了地址 A19~A16;U2,U1 则存放了地址 A15~A0。

由于控制码为“101X”,从图 6-42 中可知 U5 的 $\overline{OE}=1$,所以 74LS373 输出高阻抗,不影响 6281000 的数据总线 D0~D7。 $\overline{RD}=0$,6281000 可执行读出操作; $\overline{WR}=1$,6281000 不能执行写入操作。这时,只要 AT89C51 的 P1.3 送出低电平信号,则 6281000 的 $\overline{E}=0$,进入读出过程。同时,AT89C51 的 P1.2 也输出低电平,那么,U6 即 74LS165 就会并行接收 6281000 读出的 8 位并行数据 D0~D7。接着把 P1.2,P1.3 的输出电平改为高电平,则数据读出完毕。

随后,AT89C51 在 P1.1 输出移位时钟,并送到 74LS165 的时钟 CLKI 输入端,则 74LS165 串行在 Q7 输出数据;与此同时,AT89C51 在 P1.4 接收串行输入数据。

3. 读/写工作程序

下面给出对数据存储器进行串行读写的程序。

用 R0 间接寻址串行移入 74LS164 的 RAM 的数据或地址,数据存放在 R0 指向的最低单元内,接着单元的高 4 位存放写入或读出数据 4 位控制码,写入时为“010X”,读出时为“101X”;低 4 位为 RAM 的高 4 位地址,后两个单元为 RAM 的 16 地址,读出数据存放在 A 中。

写入 RAM 一字节数据程序为:

```
WRBYTE:  MOV    R3,#4                ; R3 计数移入 164 的位数为 4×8 位
          CLR    P1.1                ; P1.1 为 164 的移位脉冲,置初值
LOOP:    MOV    R2,#8                ; 计数 8 位
          MOV    A,@R0                ; 取移入 164 的 8 位数据
          ACALL  WR                    ; 移入 164 的 8 位数据
          INC    R0                    ; 指向下一字节
          DJNZ   R3,LOOP                ; 移入 164 共 32 位数据为止
          CLR    P1.3                ; 产生一个写入 RAM 脉冲
          SETB   P1.3
          RET
WR:       RLC    A                    ; A 的高位送进位 C          1 μs
          MOV    P1.0,C                ; C 送 164 的串行输入端      2 μs
          SETB   P1.1                ; 产生一个 164 的移位脉冲    1 μs
          CLR    P1.1                ;                               L1 μs
          DJNZ   R2,WR                ; 移完 8 位为止          2 μs
          RET
```

读 RAM 中的一字节数据程序为:

```
RDBYTE:  MOV    R3,#3                ; 计数移入 164 的数据为 3×8 位
LOOP1:   MOV    R2,#8                ; 计数 8 位
          MOV    A,@R0                ; 取移入 164 的数据
          ACALL  WR                    ; 调用移入 164 的 8 位数据的子程序
```

	DJNZ	R3,LOOP1	; 移入 164 共 24 位为止
	CLR	P1.3	; 选通 RAM 使其数据送到数据线上
	CLR	P1.2	; 将数据并行装入 165
	SETB	P1.2	; 装入信号无效
	SETB	P1.3	; RAM 片选无效
	MOV	R2,#7	; 计数 165 移出位数
LOOP2;	MOV	C,P1.4	; 读 165 的 Q7
	RLC	A	
LOOP4;	SETB	P1.1	; 产生一个 165 的移位脉冲
	CLR	P1.1	
	MOV	C,P1.4	; 读移出的 165 数据
	RLC	A	
	DJNZ	R2,LOOP4	; 读完 8 位数据
	RET		

第七章 家用电器领域的应用

单片机又称为微控制器,或称为嵌入式微控制器。顾名思义,单片机作为嵌入式的控制器是理所当然的。最适合于嵌入式控制的,而且面广量大的无疑是家用电器,所以,家用电器是单片机应用最多的领域之一。

在家用电器中,单片机是控制核心。它是家用电器实现智能化的心脏和大脑。由于家用电器体积小,故要求其控制器体积更小以便能嵌入其结构之中。而家用电器品种多,功能差异较大,所以又要求其控制器有灵活的控制功能。单片机以微小的体积和编程的灵活性而产生多种控制功能,完全能够满足家用电器的需要。所以,单片机在家用电器领域的广泛应用是必然的和合理的。

这一章,介绍 AT89CXX 系列在家用电器领域的应用。应用例子包括洗衣机、密码锁、电子钟、充电器和电话报警器等,代表了家用电器的不同类型。

7.1 AT89C2051 控制的洗衣机

AT89C2051 单片机是 ATMEL 公司 8 位单片机系列产品之一,是一种 20 引脚双列直插式芯片。它内含 2 KB 可反复烧录及擦除内存和 128 字节的 RAM,有 15 条可编程控制的 I/O 线,5 个中断触发源,指令与 MCS-51 系列完全兼容。基于上述特点,在需要 I/O 线不多的控制场合,选用它作为核心控制芯片,可使电路极大地简化,而且程序的编写及固化也相当方便、灵活。选用它设计制作全自动洗衣机控制电路,该电路的组成相对简单,工作原理清晰,易于理解。

由于 AT89C2051 的引脚少,内存容量不大,所以,所控制的洗衣机的功能有一定的限制。但是,由于洗衣机的基本功能是实现对衣物的洗涤,所以,关键在于进行洗衣程序的控制。从这一角度出发,对洗衣机的功能进行分析,确定其最优化的功能。这些功能有以下几种。

(1) 洗衣工作状态功能:强、弱洗涤。

(2) 洗衣程序功能:含 4 种独立程序,即标准洗衣程序、经济洗衣程序、单次洗衣程序、排水程序。

(3) 特殊功能:故障诊断、安全保护、防振、暂停、间歇工作功能,声光显示功能。

7.1.1 洗衣机的控制功能要求

洗衣机要完成洗衣工作,除了对一般洗衣过程的人工工作及效能进行模拟之外,还要根据洗衣机的机械电子性质进行有关控制和检测。

对于一台套桶式单缸波轮全自动洗衣机而言,首先要求能完成洗衣功能;同时还要根据用户的不同要求设置几种不同的洗衣程序;还要考虑水流的情况决定洗涤的强弱情况;另外,还要对洗衣过程出现的故障进行诊断;保证高速运转时脱水的安全性等。所以,对全自动洗衣机,一般要求具有如下基本功能。

(1) 强、弱洗涤功能。要求强洗时正、反转驱动时间各为 4 s,间歇时间为 1 s;弱洗时正、

反转驱动时间各为 3 s, 间歇时间为 2 s。

(2) 4 种洗衣工作程序, 即标准程序、经济程序、单独程序和排水程序。标准程序是进水→洗涤→漂洗→排水→脱水, 如此循环 3 次, 每循环一次洗涤或漂洗环节时间比上一循环同一环节时间减少 2 min, 具体是: 第一循环为洗涤, 时间为 6 min, 第二、第三次循环为漂洗, 时间分别为 4 min 和 2 min。排水时间采用动态时间法确定, 脱水时间为 2 min。经济程序与标准程序一样, 只是循环次数为二次。单独程序是进水→洗涤(6 min)→结束(留水不排不脱)。排水程序是排水→脱水→结束, 时间确定与上述程序相应环节相同。

(3) 进、排水系统故障自动诊断功能。洗衣机在进水或排水过程中, 若在一定的时间范围内进水或排水未能达到预定的水位, 就说明进、排水系统有故障, 此故障由控制系统测知并通过警告程序发出警告信号, 提醒操作者进行人工排除。

(4) 脱水期间安全保护和防振动功能。洗衣机脱水期间, 若打开机盖时, 洗衣机就会自动停止脱水操作。脱水期间, 如果出现衣物缠绕引起脱水桶重心偏移而不平衡, 洗衣机也会自动停止脱水, 以免振动过大, 待人工处理后恢复工作。

(5) 间歇驱动方式。脱水期间采取间歇驱动方式, 以便节能。本系统要求驱动 5 s, 间歇 2 s, 间歇期间靠惯性力使脱水桶保持高速旋转。

(6) 暂停功能。不管洗衣机工作在什么状态, 当按下暂停键时, 洗衣机须停止工作, 待启动键按下后洗衣机又能按原来所选择的工作方式继续工作。

(7) 声光显示功能。洗衣机各种工作方式的选择和各种工作状态均有声、光提示或显示。

7.1.2 控制逻辑电路

全自动洗衣机的控制逻辑电路如图 7-1 所示。它由单片机 AT89C2051 为核心加上有关集成电路及元器件组成。从图中看出, 这个全自动洗衣机控制逻辑电路相当简单。

全自动洗衣机的工作部件有 3 个, 这就是电机、进水阀和排水阀。电机是洗衣机的动力源, 它的转动带动洗衣桶和波轮的转动, 从而实现对衣物的洗涤。进水阀用于控制洗衣机的进水量。排水阀用于控制排水。电机在脱水时还高速旋转带动衣物脱水。

电机的状态有 3 种, 即正转、反转及停止状态。电机一般工作在这 3 种状态的不断转换之中, 从而实现洗涤。但在脱水时, 只工作在正转高速状态。

进水阀和排水阀则只有开、关这两种状态。

从图 7-1 的控制电路中可以看出: AT89C2051 的 P1 端口中的 P1.0~P1.3 共 4 条 I/O 线通过 4 块 SP1110 新型固态继电器分别直接驱动洗衣机的这些工作部件。SP1110 是一种交流固态继电器, 内有发光二极管及光触发双向可控硅, 10~50 mA 输入电流即可使双向可控硅完全导通, 输出端通态电流为 3 A(平均值), 浪涌电流 15 A(不重复)。之所以选用这个器件, 是因为它一方面可使电路进一步简化, 另一方面还可使强、弱两类电完全隔离, 保证主板的安全。

74S05 为六反相器, 用其作为中间缓冲器, 其中的 4 个反相器可分别驱动 4 个 SP1110 继电器, 剩余两反相器用于驱动 LED5 和 LED6。

74LS139 为双 2-4 线译码器, 选用它可解决 CPU I/O 线数量的不足。从控制要求可知, 洗衣机有 4 种洗衣工作程序, 因此须有 4 种不同的显示来加以区别。74LS139 双 2-4 线译码器仅占用 CPU 的 P3.0 和 P3.1 两口线即可提供 4 种不同显示的驱动, 其逻辑关系是: P3.0, P3.1 为“11”时 LED1 亮, 指示标准程序; 为“10”时 LED2 亮, 指示经济程序; 为“01”时 LED3

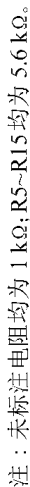


图 7-1 全自动洗衣机控制电路

亮,指示单独程序;为“00”时 LED4 亮,指示排水程序。

洗衣机的暂停功能和安全保护及防振动功能采用中断处理方式。这两个中断分别对应于 CPU 的外部中断“0”和外部中断“1”。中断请求信号通过 TC4013BP 双 D 触发器的两个 $\overline{Q}$ 分别加到 CPU 的 P3.2 和 P3.3 口线,由触发器锁存直到 CPU 响应中断为止。开盖(安全保护)或不平衡(防振动)中断信号通过由 BG1,BG2 组成的反相器送至 TC4013BP 的⑪脚 CP 端,经触发器的第⑫脚($\overline{Q}$ 端)加到 P3.3。本系统对开盖和不平衡中断采取相同的处理方法,因此,共用外部中断“1”。

为了充分利用 CPU 的 I/O 口线,P3.4 和 P3.5 采用分时复用技术,每线具有两个功能。在洗衣机未进入工作状态或洗衣机处于暂停状态期间,P3.4 为输入线,用于监测启动键的状态,当启动键按下时,洗衣机即进入工作状态或从暂停状态恢复到原来的工作状态;在洗衣机暂停中断响应期间,P3.4 为输出线,用于撤消暂停中断请求。在洗衣机进水或排水期间,P3.5 被用作输入线,用于监测水位开关状态,为 CPU 提供洗衣机的水位信息;在洗衣机高速脱水期间,当发生开盖或不平衡中断时,P3.5 为输出线,用于撤消中断请求信号。CPU 的 P3.7 线用于驱动蜂鸣器发出各种告警信号。④、⑤脚外接 6 MHz 的晶振。①脚通过 10 μ F 电容接到 +5 V 电源,可实现上电自动复位。K7 为强制复位键。洗衣机的强、弱洗可通过 K1 键进行循环选择。K1 还具有第二功能,即当洗衣机发生故障转入报警程序后,按下 K1 键可使洗衣机退出报警状态回到初始待命状态。洗衣工作程序可通过 K2 键循环选择。洗衣机的工作状态可通过 LED7~LED9 进行显示。脱水期间系统在响应开盖或不平衡中止后,CPU 采取软件查询的方式通过 P1.6 线对盖开关进行监测以确定洗衣机是否继续进行脱水操作。

7.1.3 工作控制程序

由 AT89C2051 控制的全自动洗衣机的工作程序框图如图 7-2 所示。从程序框图可以看出,这个控制程序也较为简单,所以,占用的存储器容量不大。

从程序框图中可以看出程序的基本流程,系统上电复位后,首先进行初始化,默认标准洗衣工作程序和强洗方式,然后扫描 K1,K2 和启动键 K4,这时洗衣机处于待命状态。通过 K1,K2 可分别修改强/弱洗方式和洗衣工作程序。扫描过程中当发现启动键 K4 按下时,洗衣机即从待命状态进入工作状态。

洗衣机进入工作程序后,系统首先根据 RAM 中 57H 单元的特征字判断洗衣机的洗衣工作程序,若特征字为 00H,则为排水程序(01H 为单独程序,02H 为经济程序,03H 为标准程序),这时程序直接跳转至排水操作程序段,执行单独的排水操作,否则进入进水操作程序。进水操作将 P1.0 置位驱动进水阀开启。进水期间系统不断检测水位开关 K5 的状态,当检测到 K5 闭合时,说明进水已达到预定的水位。若在规定的 4 min 极限时间内未检测到 K5 闭合,说明进水系统发生故障,此时洗衣机退出洗衣工作状态,程序跳转到 FW 为标号地址的故障处理程序段进行报警,其处理方法是:将 P1.0~P1.3 位全部置“0”,中止洗衣机的各种操作,然后蜂鸣器以响 1 s、停 2 s 的规律不断地发出报警信号,直到人工干预即按下 K1 键后为止(按下 K1 后,程序跳转回主程起始地址,洗衣机又回到待命的初始状态)。

在正常情况下,进水期间检测到 K5 闭合时,说明进水已达预定水位,这时洗衣机进入下一程序即洗涤工作。因为电机在洗涤或漂洗工作时有正、反转和间歇三种状态,所以用 P1.2,P1.3 两线才能实现对电机这三种状态的控制。其逻辑关系是:P1.3,P1.2 为“00”时电机间歇,

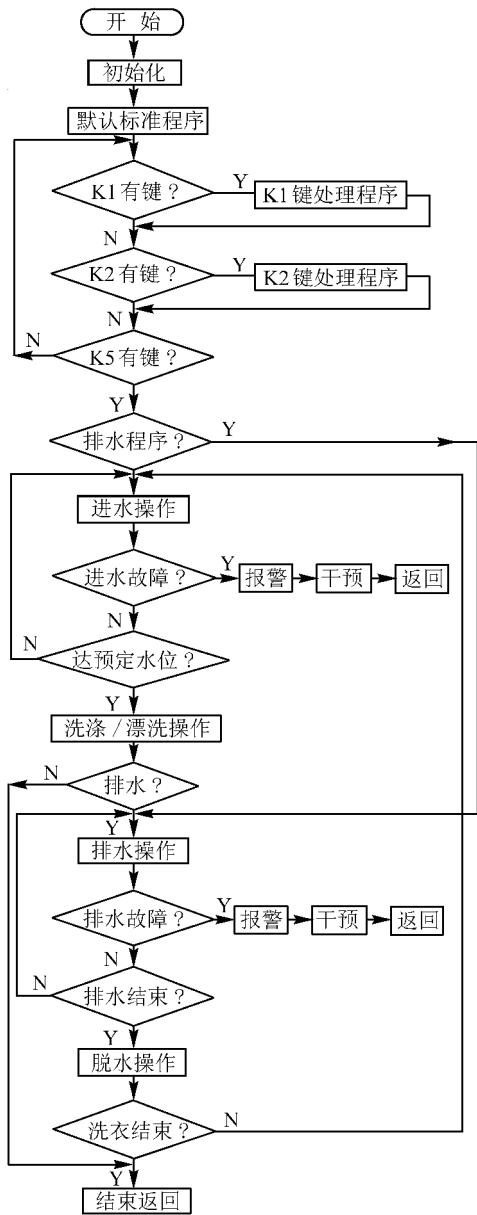


图 7-2 程序框图

为“01”时正转,为“10”时反转。洗涤时间为 6 min。洗涤结束后,系统通过一条判断指令,判断是否排水。由控制要求可知,若不排水则为单独程序,这时程序直接跳到结束报警程序段,报警三声后跳回主程序,洗衣机进入初始待命状态,单独洗衣工作程序结束,否则,进入排水进程。

排水时间采用动态时间法确定,其原理是:根据常用的空气压力水位开关的特性(即在进水中当水位达到预定水位时水位开关就接通;在排水中当桶内水位下降 11 cm 后,水位开关才断开),在排水中若从开始到开关断开所需时间为 D,则整个排水所用时间为 $2D + 50\text{ s}$ (经验值)。若在规定的 1 min 极限时间内,系统检测不到水位开关 K5 断开,说明排水系统有故障,程序跳转至故障处理程序段发出告警信号,其处理方法与进水系统故障相同。

排水结束后,洗衣机接着执行脱水操作,P1.1 维持置位状态,保持排水阀开启,P1.2 按 5 s 置位、2 s 清 0 的规律连续驱动电机高速旋转 2 min,然后脱水结束。脱水结束后系统通过一条判断指令判断整个洗衣工作是否结束。其原理是:洗衣机在每次洗涤或漂洗工作环节结束之后,洗衣工作程序标志 57H 单元减 1 一次,在脱水工作环节结束后,系统即对该单元进行检测,当检测到 57H 单元为 0 值时,说明整个洗衣工作结束,洗衣机报警三声后,即返回初始待命状态。若洗衣工作尚未结束,洗衣机再次执行进水操作,进入下一循环。

以上是洗衣机工作的大概流程,工作过程中所需的各种计时,均由定时器“0”定时中断服务程序提供。定时器“0”设置为定时方式 1,每 100 ms 产生一次中断,因此,TL0,TH0 装入的时间常数分别为 0B0H,3CH。定时器“0”中断服务子程序入口地址放在 0BH 单元中,洗衣机进入工作状态后定时器即被启动。每中断一次,70H 单元累加 1 一次,累加到 0AH 时,为 1 s,因此 71H 单元累加 1 一次,该单元累加至 3CH 时,为 1 min,这时 72H 单元累加 1 一次。系统根据这几个内存单元中的数据就可以确定洗衣机工作各个进程的时间。在暂停中断以及开盖或不平衡中断响应期间,定时中断被禁止,计时各单元内容不变。

洗衣机的暂停中断在洗衣机进入工作状态后,中断请求即被设置为允许,中断请求有效;而开盖或不平衡中断只有在洗衣进入高速脱水的工作进程时,中断请求才被设置为允许,请求信号有效。这两个中断服务程序比较简单。

所有套桶式电脑全自动洗衣机的工作原理基本是一样的。然而,不同的厂家不同型号这类洗衣机其电气控制系统电路的组成和原理则有所区别。这里所介绍的电路有别于以往的这类电路,是 AT89C2051 单片机应用的一个实例。

7.2 多功能密码锁

出于安全、方便等方面的需要,许多智能锁(如指纹辨识、IC 卡辨识)已相继问世,但这类产品的特点是针对特定指纹或有效卡,只能适用于保密要求高且仅供个别人使用的箱、柜、房间等。而且卡片式 IC 卡还有易丢失等特点,加上其成本一般较高,一定程度上限制了这类产品的普及和推广。

这一节介绍一种由 AT89C2051 编程实现的控制电路,具有按键有效指示、输入错误指示、解码有效指示、控制开锁电平、定时中断、控制报警、密码修改等功能,可在意外泄密的情况下及时修改密码。8 位十进制密码共一亿组不重复,保密性强,灵活性高,适用范围广,特别适合家庭、宾馆等场所。

7.2.1 多功能密码锁的控制电路

多功能密码锁的控制电路结构如图 7-3 所示。从图中看出,多功能密码锁控制电路十分简单,由 AT89C2051 和键盘电路组成。

AT89C2051 是与 MCS-51 系列兼容的一种高性能单片机。它具有 2 KB 片内 ROM、256 字节片内 RAM、8 个辅助寄存器。在不附加任何外围电路的情况下,能实现大部分比较复杂的逻辑控制功能,对存储容量要求不高的实际应用,AT89C2051 确实是一种不可多得的高效能单片机。

芯片引脚排列如图 7-4 所示。与 51 系列单片机类似,AT89C2051 有两个 P 口:P1 口和

P3 口。P1 口作为一般的 I/O 口,其中 P1.1 和 P1.0 可作为模拟信号输入端口。P3 口是双功能 I/O 口,除一般 I/O 功能外,P3.0 和 P3.1 兼有串行发送和接收的功能,P3.2 为外部中断输入,P3.3,P3.4 分别为定时器 0,1 的计数输入。XALT1,XALT2 为外部晶振端口。V_{CC} 与 GND 分别接电源和地。

很明显,AT89C2051 这种功能和引脚是十分适合于用作多功能密码锁的核心控制部件的。因为密码锁有一个 3×4 阵列的键盘,AT89C2051 必须能对输入的键码进行输入和处理。AT89C2051 有 15 条 I/O 引脚,完全可满足用于组成键盘的 I/O 扫描矩阵,在 3×4 阵列的键盘中,只需用 7 条 I/O 引脚就可以了。余下的 I/O 引脚可以用于其他功能。

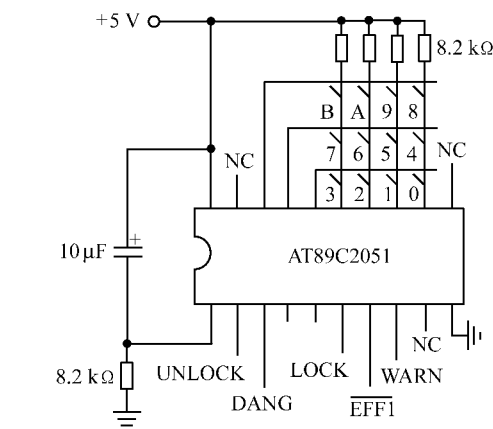


图 7-3 多功能密码锁控制电路

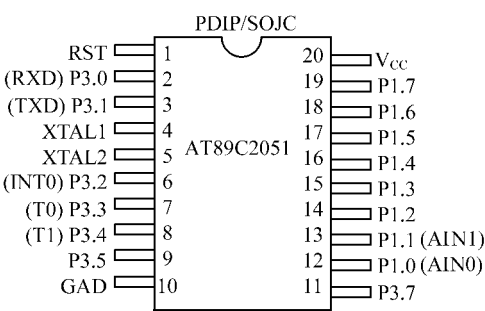


图 7-4 AT89C2051 的封装引脚

AT89C2051 的 I/O 引脚应用如下。
键盘扫描所用引脚 P1.0~P1.6,其中 P1.4~P1.6 这 3 条引脚用于输出扫描信号,P1.0~P1.3 这 4 条引脚用于检测按键的状态。P1.0~P1.6 的状态组成的键码以及对应按键如表 7-1 所列。

表 7-1 密码锁的键码

P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	键 盘
0	1	1	1	1	1	0	8
0	1	1	1	1	0	1	9
0	1	1	1	0	1	1	A
0	1	1	0	1	1	1	B
1	0	1	1	1	1	0	4
1	0	1	1	1	0	1	5
1	0	1	1	0	1	1	6
1	0	1	0	1	1	1	7
1	1	0	1	1	1	0	0
1	1	0	1	1	0	1	1
1	1	0	1	0	1	1	2
1	1	0	0	1	1	1	3

在程序执行中,是先将 P1.6 送出“0”,再判别 P1.3~P1.0 的输入。所以,实质是在 P1.6~P1.4 送出“011”,接收 P1.3~P1.0 的输入状态,这时判别 8,9,A,B 这 4 个键。接着,在 P1.6~P1.4 送出“101”,接收 P1.3~P1.0 的输入,这时判别 4,5,6,7 这 4 个键。再而,在 P1.6~P1.4 送出“110”,接收 P1.3~P1.0 的输入,这时判别 0,1,2,3 这 4 个键的状态。

开锁电平 UNLOCK 输出脚 P3.0,当输入的键正确时则在 P3.0 输出开锁电平 UNLOCK。
报警电平 DANG 输出脚 P3.1。在 6 次误码输入的条件下,进行报警。
LOCK 控制引脚 P3.2,90 s 之后停止输入。
有效按键提示 EFFI 引脚 P3.3。
错误输入警告 WARN 引脚 P3.4。

另外,还有引脚 5,4 接石英振荡器的 XTAL2,XTAL1 端,组成石英振荡器电路,在图 7-3 中没有画出。引脚 1 用于复位。

7.2.2 密码锁的控制程序

密码锁的控制程序由延时子程序、修改密码子程序、扫描输入子程序、时钟中断子程序及主程序组成。程序框图如图 7-5 所示。它对输入的 8 位十进制密码进行判别,如果输入密码正确,则可以直接开锁或修改密码之后开锁。如果不正确,并且 6 次以上输入不正确,则报警并封锁输入口。

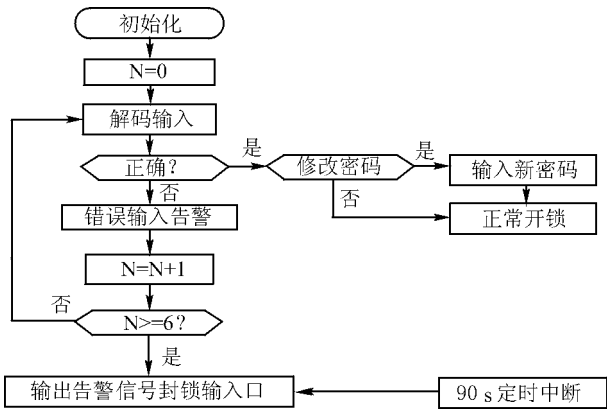


图 7-5 程序框图

1. 主程序

```
ORG      0H
SJMP     START
ORG      0BH
SJMP     INT
START                                ; 初始化
MOV      SP, #70H
SETB     P3.0
SETB     P3.1
SETB     P3.2
```

	SETB	P3.3	
	MOV	TMOD, #01H	
	MOV	TL0, #00H	
	MOV	TH0, #00H	
	SETB	TR0	
	SETB	ET0	
	SETB	EA	
	MOV	R2, #0FFH	
	MOV	3AH, #03H	
	MOV	R3, #06H	
			; 设置初始密码
PASSWORD:	MOV	R4, #08H	
	MOV	R0, #40H	
	MOV	A, #07H	
	MOV	@R0, A	
	INC	R0	
	INC	A	
	DJNZ	R4, PASSWORD	
DETPW:	MOV	R4, #08H	; 输入密码
	MOV	R0, #30H	
AGAIN:	ACALL	INPUT	
	MOV	A, R7	
	CJ	R7, #0BH, IN	
	ACALL	CHPSW	; 修改密码子程序
	JMP	GETPW	
IN:	MOV	@R0, A	
	CLR	P3.3	; 按键有效显示
	ACALL	DELAY	
	SETB	P3.3	
COMP:	MOV	R4, #08H	; 比较密码
	MOV	R0, #30H	
AGAI:	MOV	50H, @R0	
	ADD	R0, #010H	
	MOV	A, @R0	
	SUB	R0, #010H	
	CJNE	A, 50H, ONCEMORE	
	INC	R0	
	DJNZ	R4, AGAI	
	CLR	P3.0	
HERE:	AJMP	HERE	
ONCEMORE:	CLR	P3.4	; 错误输入警告
	ACALL	DELAY	

```

        SETB    P3.4
        DJNZ    R3, GETPW
        CLR     P3.1                ; 6 次错误输入报警
THERE:  AJMP    THERE
```

2. 延时子程序

```

DELAY:  MOV     R5, #30H
DELAY1: MOV     A, #0FFH
LOOP:   DEC     A
        JNZ     LOOP
        DJNZ    R5, DELAY1
        RET
```

3. 修改密码子程序

```

CHPSW:  MOV     R0, #40H
ANOTHER: ACALL  INPUT
        MOV     A, R7
        MOV     @R0, A
        INC     R0
        CLR     P3.3                ; 按键有效提示
        ACALL  DELAY
        SETB    P3.3
        DJNZ    R4, ANOTHER
        RET
```

4. 扫描键盘输入子程序

```

INPUT:  MOV     R6, #02H
        MOV     R1, #40H

LOP:    ACALL  DELAY
        MOV     P1, #0FFH            ; 扫描码
        CLR     P1.6
        MOV     A, P1
        ANL     A, #0FH                ; 检验有否按键
        CJNE    A, #0FH, INKEY
        MOV     P1, #0FFH
        CLR     P1.5
        MOV     A, P1
        ANL     A, #0FH
        CJNE    A, #0FH, INKEY
        MOV     P1, #0FFH
        CLR     P1.4
        MOV     A, P1
```

```

ANL      A, #0FH
CJNE     A, #0FH, INKEY
SJMP     LOP

INKEY:   MOV     @R1, P1
INC      R1
DJNZ     R6, LOP
DEC      R1
MOV      A, @R1
DEC      R1
CJNE     A, 40H, INPUT
RET
```

5. 时钟中断子程序

```

INT:     PUSH    ACC
MOV      TMOD, #01H
MOV      TL0, #00H
MOV      TH0, #00H
SETB     TR0
SETB     ET0
SETB     EA
DJNZ     R2, CONT
CONTINUE
DJNZ     3AH, CONT
CLR      P3.2                ; 时钟中断控制

STOP:    AJMP    STOP
CONT:    POP      ACC
RET1
```

7.3 用电力线通信的子母电子钟

在港口、飞机场、火车站、汽车客运站、学校、机关、大的公司、工厂等地方,通常需要统一的时间显示,以求工作、学习能同步按时间依某一规则进行,或者有利于旅客掌握准确的时间,防止误车误船或误失航班。为了进行统一时间显示,可采用子母电子钟系统。一般的子母钟系统是由母钟产生统一的时间,然后通过信号线发送给处于各个不同地点的子钟,并进行显示。这样,不同的地点就可以显示统一的时间。

由于一般子母电子钟系统的母钟和子钟之间需要专门的信号连线,故整个系统就要考虑布线的成本,在各个地点之间的距离较远时,这种布线的成本就会相当可观。同时也会增加工程施工难度及整个系统的工程费用。因此,采用现有的电力线作为母钟向子钟传送时间信号的媒介,是一种有效而节省成本的方法。这一节介绍用电力线通信的子母电子钟系统。

7.3.1 子母电子钟系统结构

子母电子钟系统如图 7-6 所示。它由一个母钟和 n 个子钟组成。母钟用于产生标准时

间,标准时间信号通过电力线向子钟传送,子钟接收到母钟的时间信号之后,则把时间显示出来。

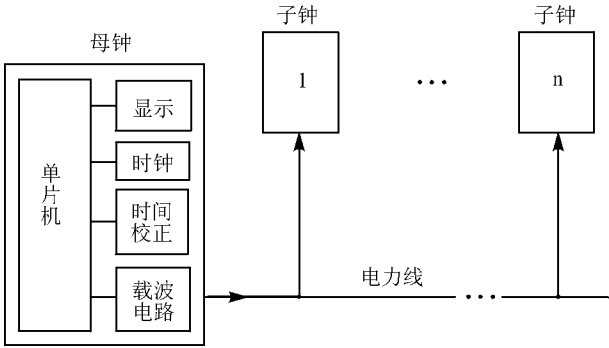


图 7-6 子母电子钟系统结构

母钟包括显示器、时钟芯片电路、时间校正电路、载波电路以及单片机等。显示器用于显示时间;时钟芯片用于产生标准时间,可包括年、月、日、时、分、秒等;时间校正电路用于接收电台时间信号,以校正母钟的时间;载波电路由载波发生器及调制信号即时间信号组合而成,同时通过耦合电路交连到电力线上;单片机用于控制时间信号的产生、时间的校正及显示,同时控制载波的时间信号的传送。

子钟的结构和母钟是类同的,最大的区别在于它没有载波发送电路,只有载波接收电路。由于电力线可以对母钟及子钟供电,同时又用于进行时间信号的传送,所以母钟和子钟不存在附加的专用时间信号传送线,这样,它们只要用电源的插头连接到电力线的插座上,就可以同时实现供电及时间信号传送的目的。这样的工作方式,使得母钟和子钟都可以做成一个独立的完整时钟仪器,从而系统的使用就十分方便,特别是子钟的布局可以十分随意,只要有电力线插座的地方都可以安装子钟。系统的安装成为一件十分简单而方便的事情。

在图 7-6 所示的子母电子钟系统中,无论母钟还是子钟,其核心部件单片机都是采用 ATMEL 公司的 AT89C51。

母钟和子钟之间的逻辑结构采用总线结构。对于同一相电力线上的母钟和子钟,直接和电力线相连即可。对于处于不同相线的母钟和子钟,应在相线之间连接容量为 $1\mu\text{F}/630\text{V}$ 的电容器。

母钟和子钟之间的通信方式采用异步串行通信方式,因此时间信号是以异步串行方式从母钟单片机发出的,而子钟的单片机也以异步串行方式接收。时间信号传送速率为 $2\,400\text{ bps}$,采用幅移键控 ASK 方式调制,载波频率为 $100\sim150\text{ kHz}$ 。

7.3.2 母钟的逻辑结构

母钟的逻辑电路结构如图 7-7 所示。

母钟的核心是 AT89C51,即 AT89C51 单片机是用于执行全面控制的,时钟芯片采用 MC146818,并用不间断电源 E_c 供电,晶振频率 $4.193\,4\text{ MHz}$,校准后日误差小于 0.1 s 。显示电路用 LED 数字显示器显示时钟电路的时间,按键电路用于对时钟电路进行调整。为了提高时间准确性,利用广播电台的报时信号对时钟电路进行自动校正。收音机接收某广播电台的广播,音频信号中含有广播电台在整点时发出的“嘟”声报时信号。利用锁相环电路 LM567,

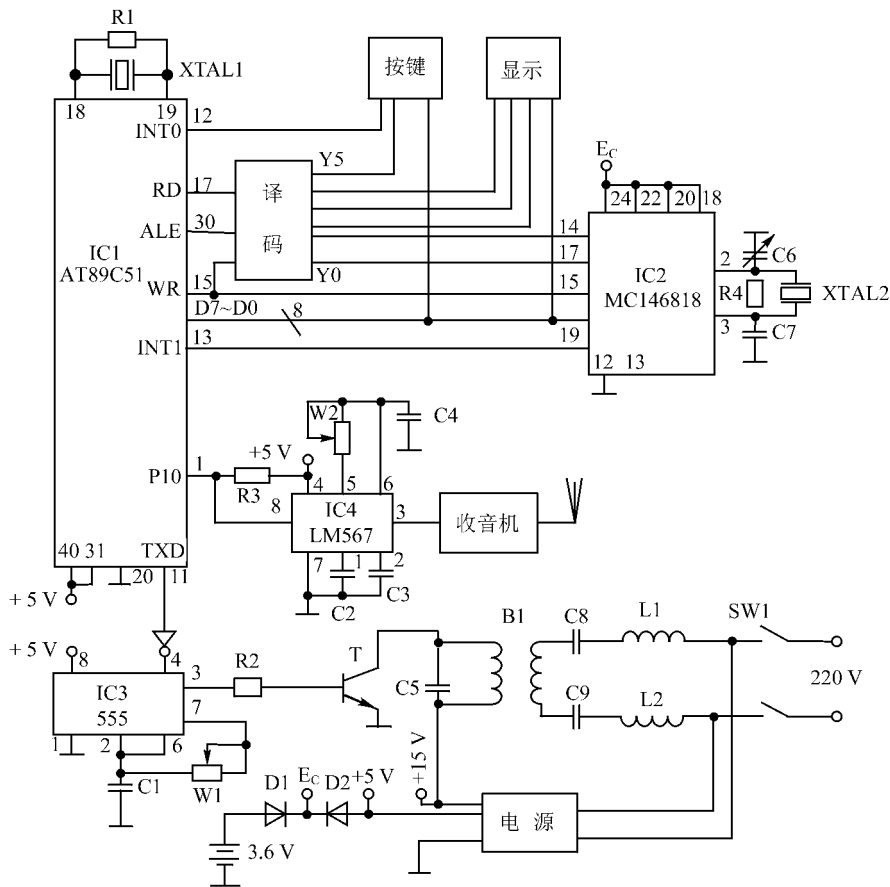


图 7-7 母钟的逻辑电路

调节 W2, 鉴别出“嘟”声报时信号, 送入 CPU 的 P10 端。定时电路的中断申请信号连接 INT1, 利用定时电路的更新周期结束中断, 每秒产生一次 INT1 中断。异步串行口发出的串行数据反相成反码之后控制 LM555 的振荡, 产生幅移键控 ASK 信号。ASK 信号经三极管选频放大后, 通过输出变压器 B1 和电容耦合后传送到电力网上。用 W1 调节 ASK 信号的载波频率, 可以使三极管输出的 ASK 信号幅值最大。

7.3.3 子钟的逻辑结构

子钟的逻辑电路和母钟的基本上相似。只是由于它用于接收、显示时间, 故有些不同。子钟也采用集成实时时钟芯片 MC146818 定时和 AT89C51 控制, 但显示用 LED 数字显示器较大, 其他电路相对简单。考虑成本和使用方便, 按键仅设加、减两个, 用于对时钟电路进行调整。串行口由发送数据改为接收数据。电力线传送来的 ASK 信号经输入变压器 B2 耦合和选频后, 再经 LM356 放大, 调节 W3, 用 LM567 锁相环检出此信号, 送入单片机的 RXD 端, 单片机利用串行口接收的时间信号校正时钟电路, 使其与母钟的时间一致。ASK 信号的接收、放大和鉴别电路如图 7-8 所示, 其他部分电路由于和母钟类同, 故省略。

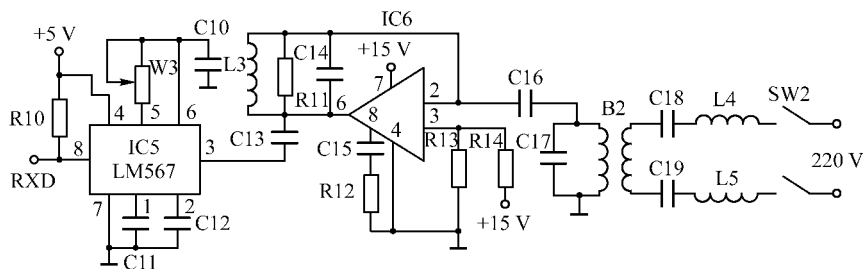


图 7-8 子钟的接收、鉴别电路

7.3.4 实时时钟芯片 MC146818

在母钟和子钟中都用到时钟芯片 MC146818,它是母钟和子钟的时间源。单片机需要对这个芯片进行恰当的控制,才能产生正确的时间。下面对 MC146818 的结构及原理进行介绍。

1. MC146818 的主要性能

MC146818 是 CMOS 实时时钟/日历芯片。该芯片可产生秒、分、时、星期、日、月及年等 7 个时标,程序能通过读写获得和修改这些时标,并能编程任意设定产生间隔为 $30.517 \mu\text{s} \sim 24 \text{ h}$ 的中断申请。该片可实现 100 年日历,可广泛用于需要实时时钟/日历的测量仪器和为各种计算机系统提供精确的时钟/日历,其主要特点是:

- (1) 带有 14 个寄存器和 50 字节供用户使用的静态 RAM;
- (2) 可选择三种时钟频率: 4.194 304 MHz, 1.048 576 MHz, 32.768 kHz;
- (3) 时标可选择二进制或 BCD 码表示;
- (4) 可编程方波输出;
- (5) 三种独立的可编程中断: 每天的报警中断,从 1 次/s $\sim$ 1 次/24 h;周期中断,周期从 $30.5 \mu\text{s} \sim 500 \text{ ms}$;更新周期结束中断。

2. MC146818 的引脚功能

MC146818 是 24 引脚双列直插式集成芯片,引脚配置如图 7-9 所示。各引脚名称及功用分别说明如下。

V_{DD} : 电源正极, +3 $\sim$ +6 V, 典型值 +5 V。

V_{SS} : 接地端, 接电池负极。

OSC1, OSC2: 时钟输入端。

CKOUT: 时钟输出端, 该脚的输出频率可等于输入时钟频率或经 4 分频后的时钟频率, 供有关计算机系统使用。

CKFS: 时钟输出控制端。该脚接高电位时, CKOUT 端输出频率等于该芯片输入的时钟频率; 该脚接低电位时, CKOUT 端输出频率等于该芯片输入的时钟频率的 1/4。

SQW: 可编程方波输出端。

AD0 $\sim$ AD7: 双向地址/数据总线。

AS: 地址选通线。

DS: 数据有效或数据读选通。

- R/ $\overline{W}$:读写启动端。
- $\overline{CE}$:片选端。
- $\overline{IRQ}$:中断请求端。
- $\overline{RESET}$:复位端。
- PS:电源电位检测端。备用电池电压低于临界最小值时,在内部寄存器内作上标志,以供用户查询。
- NC:空引脚。

MC146818 内部有 10 个时标寄存器,4 个状态寄存器和 50 字节供用户使用的静态 RAM。地址分配如图 7-10 所示。

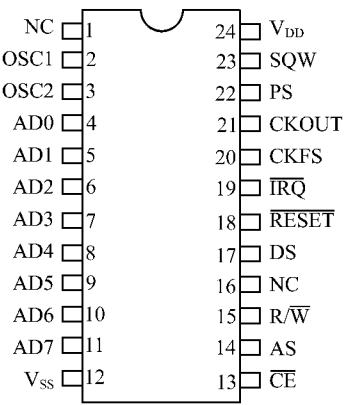


图 7-9 MC146818 引脚

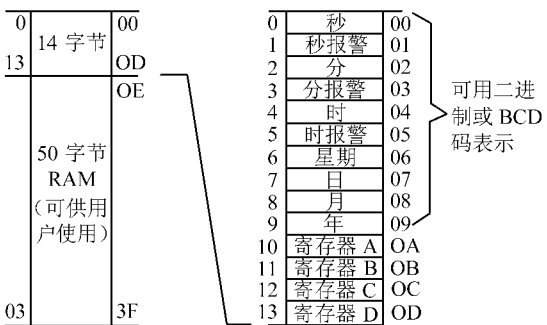


图 7-10 时标及状态寄存器

单片机可以通过读 MC146818 的内部时标寄存器得到时间和日历,也可通过写操作初始化芯片的内容。10 个时标寄存器可以选择二进制码或 BCD 码表示。50 字节静态 RAM 可供用户使用,在主机掉电时可用来保存一些重要的数据。

3. 状态寄存器 A,B,C,D

MC146818 的 4 个状态寄存器用来控制和指明芯片的工作状态,程序可随时读写这 4 个寄存器,下面简要介绍各寄存器的作用。

(1) 寄存器 A

寄存器 A 的控制字的格式如下:

b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

寄存器 A 各位不受复位的影响,除 UIP 位外,其他各位均可读写。

UIP 位:更新周期标志位。该位为“1”时,表示芯片正处于或即将开始更新周期,此时程序不准读写时标寄存器;该位为“0”时,表示至少在 244 μ s 后才开始更新周期,此时程序可读芯片内时标寄存器。该位是只读位。

DV2,DV1,DV0:除法器组态位。当除法器解除复位状态时,另一个更新周期将在半秒后开始。故在程序初始化时可用这三位精确地使芯片在设定的时间开始工作,其作用见表 7-2。

表 7-2

时钟频率	寄存器 A			工作方式	除法器是否复位
	DV2	DV1	DV0		
4.194 304 MHz	0	0	0	是	否
1.048 576 MHz	0	0	1	是	否
32.768 kHz	0	1	0	是	否
任何频率	1	1	0	否	是
任何频率	1	1	1	否	是

注：其他组合用于测试芯片。

RS3,RS2,RS1,RS0:周期中断和可编程方波输出速率选择位。各种不同的组合可以产生不同的输出,它们的编码及相应输出如表 7-3 所列。程序可以通过设置寄存器 B 的 SQWF 和 PIE 位控制是否允许周期中断和方波输出。

表 7-3 RS3,RS2,RS1,RS0 的编码及相应输出

寄存器 A 选择位				4.194 304 或 1.048 576 MHz 时基		32.768 kHz 时基	
RS3	RS2	RS1	RS0	中断周期	SQW 输出频率	中断周期	SQW 输出频率
0	0	0	0	—	—	—	—
0	0	0	1	30.517 μ s	32.768 kHz	3.906 25 ms	256 Hz
0	0	1	0	61.035 μ s	16.384 kHz	7.812 5 ms	128 Hz
0	0	1	1	122.070 μ s	8.192 kHz	122.070 μ s	8.192 kHz
0	1	0	0	244.141 μ s	4.096 kHz	244.141 μ s	4.096 kHz
0	1	0	1	488.281 μ s	2.048 kHz	455.281 μ s	2.048 kHz
0	1	1	0	976.562 μ s	1.024 kHz	976.562 μ s	1.024 kHz
0	1	1	1	1.953 125 ms	512 Hz	1.953 125 ms	512 Hz
1	0	0	0	3.906 25 ms	256 Hz	3.906 25 ms	256 Hz
1	0	0	1	7.812 5 ms	128 Hz	7.812 5 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

(2) 寄存器 B

寄存器 B 的控制字的格式如下：

b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

寄存器 B 可读可写,用于控制芯片的工作状态。

SET 位:当该位为“0”时,芯片处于正常工作状态,每秒产生一个更新周期来更新时标寄存器;该位为“1”时,芯片停止工作,程序在此期间可以通过写操作来初始化芯片的时标寄存器。

PIE,AIE,UIE 位:分别为周期中断、报警中断、更新周期结束中断允许位。各位为“1”时,允许芯片产生相应的中断。

SQWE 位:方波输出允许位。为“1”时输出方波;为“0”时输出低电平。

DM 位:时标寄存器用 BCD 码表示或用二进制表示选择位。为“1”时,用二进制码;为“0”时用 BCD 码。

24/12 位:24/12 小时模式设置位。为“1”时,用 24 小时模式;为“0”时,用 12 小时模式。

DSE 位:夏令时服务位。为“1”时,用夏时制;为“0”时不用。

(3) 寄存器 C

寄存器 C 的控制字的格式如下:

b7	b6	b5	b4	b3	b2	b1	b0
IRQF	PF	AF	UF	0	0	0	0

该寄存器标志芯片的状态,程序读该寄存器后,该寄存器将自动清 0。

IRQF 位:中断申请标志位。

该位逻辑表达式为: $IRQF=PF\times PIE+AF\times AIE+UF\times UIE$ 。无论何时,当 IRQF 位变“1”时, $\overline{IRQ}$ 引脚将变低电平,发出中断请求。

PF,AF,UF 位:这三位分别为周期中断、报警中断、更新周期结束中断标志位。只要满足各中断的条件,相应的中断标志位将置“1”。

b3~b0 位:保留位。读出值始终为“0”。

(4) 寄存器 D

寄存器 D 的控制字的格式如下:

b7	b6	b5	b4	b3	b2	b1	b0
VRT	0	0	0	0	0	0	0

寄存器 D 为只读寄存器。

VRT 位:芯片内部 RAM 与寄存器内容有效标志位。该位为“1”时,指出芯片内部的 RAM 和寄存器内容有效。读该寄存器后,该位将自动置“1”。

b6~b0 位:保留位。读出的数值始终为“0”。

7.3.5 MC146818 的工作

初始化 MC146818 时,应先将 MC146818 状态寄存器 B 中的 SET 位置“1”,禁止芯片内部的更新周期,然后初始化 10 个时标寄存器和状态寄存器 A 的内容,此后再通过读状态寄存器 C 清寄存器 C 中的中断标志位,通过读寄存器 D 而置寄存器 D 中的 VRT 位,因为读状态寄存器 D 后 VRT 位会自动置“1”,最后将状态寄存器 B 中的 SET 位置“0”,让芯片开始工作。

MC146818 处于正常工作状态时,每秒钟产生一个更新周期,更新周期的基本功能是秒时

标寄存器内容加 1,同时检查是否有溢出。如有溢出则相应进位,并自动辨认月、年的结束。其另一功能是检查 3 个报警时标寄存器的内容是否与相应时标寄存器的内容相符,如相符则 AF 位置“1”。如报警时标寄存器的内容为 COH~FFH,为“不关心”状态,即任何时刻都作相符合的结论。

在更新周期内,芯片内部时标寄存器数据处于更新阶段,故在该周期内,控制 MC146818 的单片机不能读芯片时标寄存器的内容,否则将得到不确定数据。芯片处于更新周期的标志是寄存器 A 中的 UIP 位为“1”。因此这时可以用以下方法获得满意的数据。

第一种方法是:利用更新周期结束发出的中断。

MC146818 专设一项功能。它可以编程允许在每次更新周期结束后产生中断请求,提醒单片机将有 998 ms 左右的时间去获取有效的数据,在中断之后的 998 ms 时间内,程序可先将时标数据读到芯片内部的静态 RAM 中。因为芯片内部的静态 RAM 和状态寄存器是可随时读写的。在离开中断服务子程序前应清除寄存器 C 中的 IRQF 位。

第二种方法是:利用寄存器 A 中的 UIP 位来指示芯片是否处于更新周期。在 UIP 位从低变高 244 μ s 后,芯片将开始更新周期,所以如检测到 UIP 位为低电平时,则用户至少有 244 μ s 的时间去获取有效信息。如检测到 UIP 位为“1”,也可暂缓读数据,等到 UIP 位变“0”后再读。

7.3.6 子母电子钟的工作程序

母钟和子钟的程序设计有许多相同之处,主程序用于完成初始化及对定时器、串行口、INT0、INT1 的工作方式设置等,而主要工作由相应的中断服务子程序完成。定时器 0 设置在工作方式 1,定时间隔设为 0.1 s,用于程序定时。定时器 1 设置在工作方式 2,用于串行口波特率发生器。串行口设置在工作方式 3,采用偶校验,波特率设为 2 400 波特。INT0 和 INT1 均设为低电平有效。在母钟程序中,首先显示 88:88:88 揭示程序开始,等待 10 s,在此时间内可以用按键调整定时电路的时间。若 10 s 内没有按键,则自动进入正常工作状态。在程序开始时开 INT0 和定时器 0 中断。当有按键时产生 INT0 中断,中断服务子程序读取按键编码,数字键按时、分、秒顺序存入输入缓冲区,控制键则根据输入缓冲区时、分、秒的内容设置定时电路的时间等,最后进入正常工作状态。

在正常工作时,关 INT0 中断,按键不再起作用,然后打开 INT1 中断。在 INT1 中断服务子程序中,CPU 读取定时电路的时、分、秒,在 LED 上显示,并写入异步串行口发送寄存器。发送的数据按时、分、秒顺序,最后以 AAH 结尾。为了避免混乱,各中断不允许嵌套,在中断服务子程序中关中断,返回主程序之前再开中断。INT1 中断优先级设为最高。定时器 0 用于程序定时,自始至终都开放中断,在程序运行开始的前 10 s 和以后的正常工作状态执行不同程序。为此,设一正常工作状态标志,在程序开始运行的前 10 s 设为 00H,程序正常工作时设为 FFH。图 7-11 给出定时器 0 的中断服务子程序流程图。图中正常工作状态不断检测收音机报时信号。当有“嘟”声报时信号时,P10 为低电平,若宽度达 0.5 s、间隔在 1.8~2.2 s 间即认为是正确的“嘟”声报时信号。当接收到连续 5 个“嘟”声信号时,即对定时电路校时。因此,时距“嘀”声最后一响还差 2 s,校时应为整点差 2 s。校时时,首先读取定时电路的时、分、秒,将分、秒设置成 59 min 58 s,然后再根据读取的分钟数设置小时。若读取的分钟数小于 30,则将小时减 1,否则小时数不变。这样保证了定时电路与广播电台的时间一致,保证了整个时间系统的准确性。

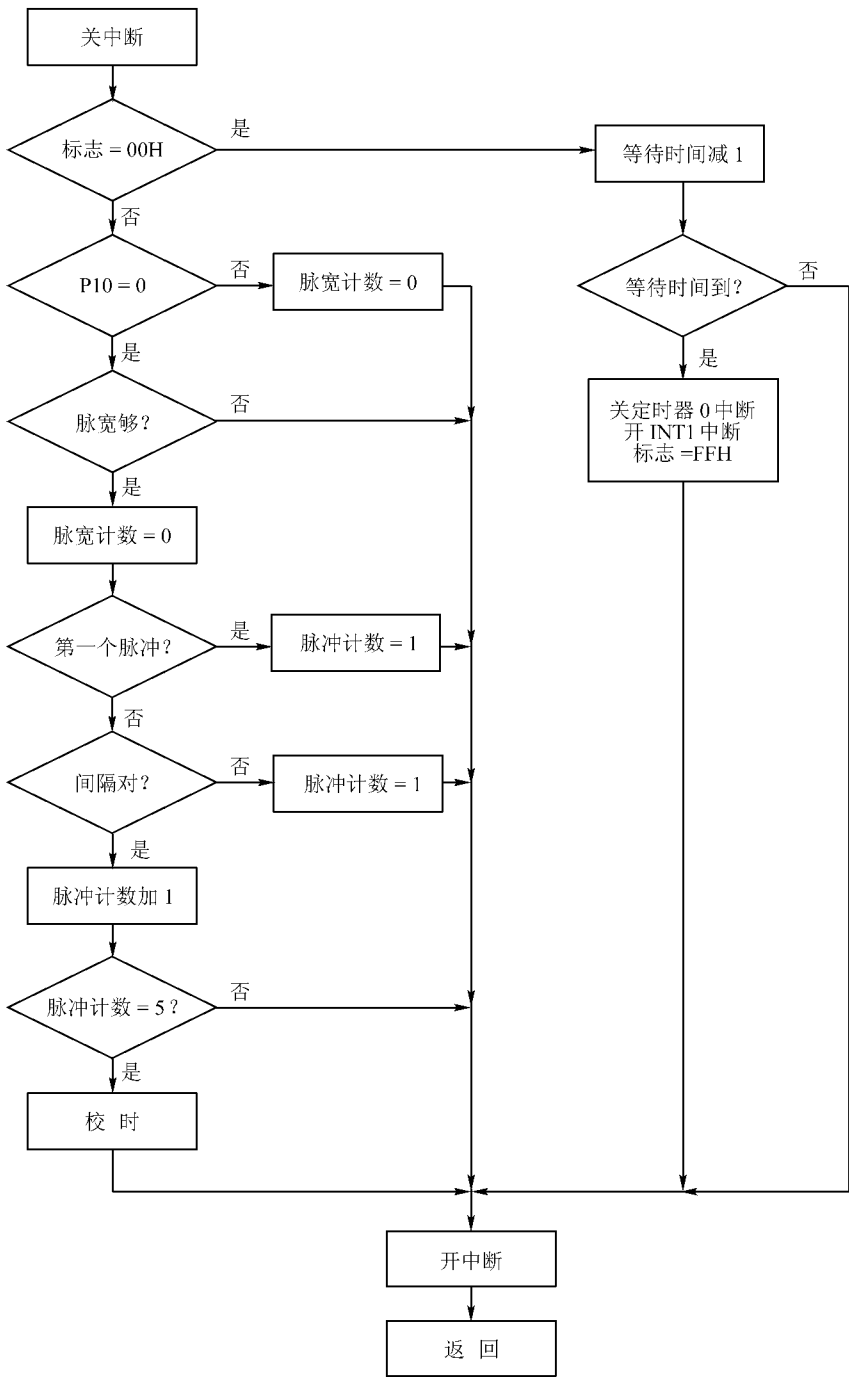


图 7-11 定时器 0 中断服务子程序框图

子钟在程序运行开始的前 10 s,开定时器 0 中断,读取定时电路的时、分、秒时间,存入显示缓冲区,并在 LED 上显示。在定时器 0 中断服务子程序中对 P1 口的 D1,D0 位检测,用加、减两键可以对 LED 显示的时、分、秒时间进行调整。调整位以闪烁方式显示,每秒钟显示和熄

灭各一次。当单按下加键时,闪烁位数据加 1;当单按下减键时,闪烁位数据减 1;当同时按下加、减两键时,闪烁位按时、分、秒顺序移动。为此,在内存变量中设一指针,用于指向闪烁位在显示缓冲区的位置。初值指向显示缓冲区前一字节,当加、减两键同时按下时,指针加 1。当指针移出显示缓冲区时,单片机即按显示缓冲区的时间设置定时电路,最后进入正常运行状态。若 10 s 内没有按键,则也自动进入正常工作状态。在正常工作时开 INT1 和串行口中断,关定时器 0 中断,加、减两键不再起作用。在 INT1 中断服务子程序中,单片机读取定时电路的时、分、秒时间,然后在 LED 上显示。在串行口接收中断服务子程序中,单片机读取接收寄存器的数据,判断偶校验正确后存入输入缓冲区,接收数据记录数加 1;若偶校验错误则舍弃,接收数据记录数不变。当接收到时、分、秒发送结束标志 AAH 时,若这时已完整接收了 3 字节的时、分、秒时间,则按照接收的时、分、秒时间校正定时电路;若没有接收到完整的 3 字节时、分、秒时间,则不作校正。最后将接收数据记录数清 0。同母钟一样,各中断不允许嵌套,INT1 中断优先级设为最高。因各部分程序较简单,在此省略程序流程图。

7.4 电话语音报警器

无论是一般的学校、机关、医院、商业机构、工厂还是居民的住宅,家居安全是人们最重视的一个因素,特别是现在建设的住宅小区,安全防范是一项很重要的建设指标。智能建筑和智能小区的推出,要求对人们的安全考虑得更全面、更详细、更深入。因此,各种安全防范的设备、系统不断地推出,也不断地更新,一方面希望能满足目前安全防范的迫切需要,另一方面追求更完善的安全环境和保障。

电话语音报警器是一种很有效的安全防范报警设施,它能及时把发生的险情向有关部门或人员报告,以引起注意并及时采取措施,排除险情,保证人们生命和财产安全。

家居险情有两大类型,一类是外侵型,像外来的抢劫、偷窃案;另一类是内致型,像火灾、煤气泄漏、缺氧、人员发病、小孩无知触及电器,小孩在高层楼房爬窗外伸等。

电话语音报警器可以实现对火警、匪警、盗警和急救这 4 种警情的报警。当出现上述 4 种警情时,报警系统就通过电话及电话线路自动地向有关部门进行拨号,在接通时进行语音报警。而且,这种报警是向多个报警点进行报警,从而保证了报警的有效性。另外,一旦电话的户外线路被切断或者出现非法并机的情况时,电话机内部的扬声器可发出报警语音,向附近的人员示警。

7.4.1 电话语音报警器的功能

电话语音报警器的主要功能有如下几点。

- (1) 可存储 10 个电话号码,电话号码最长可达 15 位,遇到警情时可自动拨打 6 个号码。
- (2) 每个号码在每次警情中可执行 9 次试拨,每轮拨号每个号码拨 3 次,共执行 3 轮拨号。
- (3) 可手动录入 6 段语音,每段语音时间最长为 16 s。
- (4) 可实现自动无间断放音操作。
- (5) 可进行来电自动应答操作。
- (6) 可进行最近 2 次来电语音自动录入,每次最长 16 s,并可手动放音。
- (7) 有 4 路由用户定义的报警信号检测输入。
- (8) 有电话户外线路异常情况检测。

(9) 室内电话机可在无警情时正常使用,随意外拨通话;有警情报警时室内电话被阻断,以保证报警器正常可靠运行。

电话语音报警器的功能模块框图如图 7-12 所示。从图中可知,它由电源、可信音检测电路、语音网络、拨号网络、户外线异常检测、语音系统、MCU 控制和 4 路报警输入组成。

电源用于向报警器提供工作所需的直流电压。

拨号网络用于执行拨号过程,以选择对方通话的电话机。拨号网络产生的是双音频 DTMF 信号。

话音网络对拨号网络产生的双音频信号进行放大,并传到电话线路上传送出去。

可信音检测用于对电话线路上的信号进行检测,以区别可信音的类别,并传送给单片机。

语音系统用于录音及放音。当外部有电话输入时,可执行录入;而进行报警时,可执行语音的放音。

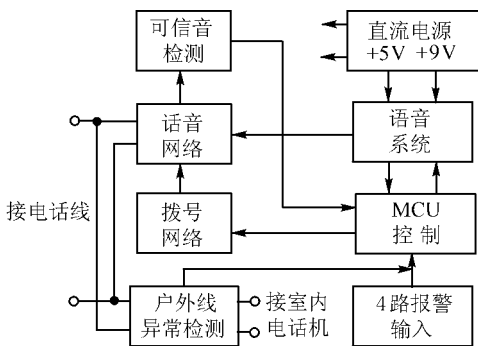


图 7-12 电话语音报警器框图

4 路报警输入可输入火警、匪警、盗警和急救这 4 种报警的警情。

MCU 控制是报警器的核心,MCU 是单片机的一种简称。MCU 是 Micro Computer Unit 或 Micro Controller Unit 的简写,故单片机可以称微型计算机器件或微控制器器件。单片机可以执行接收报警输入,产生自动拨号,户外线路异常采样、可信音检测处理、放录音控制等工作。整个语音报警器是在单片机的程序控制下完成所有工作的。单片机采用 AT89C51。

7.4.2 报警器的工作原理

电话语音报警器的简化电路原理图如图 7-13 所示。

1. 电话户外线异常检测

R4,D2(LED)组成进户线极性检测电路。当进户线极性反接时,LED 亮。户外线异常检测电路由 IC5,R1~R3,D1 等组成。本级电源由 +9 V 干电池提供,光电耦合电路 IC2 将程控交换机电源与系统主电源隔离。

程控交换机的馈电电压通常为 48 V。正常挂机状态下,或门 IC5 的脚 1 电压为逻辑高电平,光耦截止,其集电极输出为高电平;当本系统(JK1)或室内电话机(TEL)摘机时,进户线端口电压降至 10 V 左右,此时 IC5 的脚 1 电压降为逻辑低电平,而摘机回路电流流经 R1 在 IC5 的脚 2 形成逻辑高电平,光耦仍然截止。

当户外线异常,如断线、户外盗打等时,将导致 IC5 的脚 1 电压为逻辑低电平,又因系统内无摘机回路而使脚 2 也呈逻辑低电平,光耦导通,其集电极输出为低电平,并通过 P35 端口向 AT89C51 送出户外线异常采样信号。

2. 拨号存储与自动拨号

拨号电路由拨号键盘 KEY、模拟拨号开关阵列以及拨号 IC 等组成,并在单片机 AT89C51 的模拟控制或手动操作下工作。当警情发生时,单片机 AT89C51 根据程序的设定对模拟开关阵列进行有序操作,以模拟键盘的存储器拨号过程,并根据电话线路上的可信音类别作出相应的控制反应。当线路忙时重复挂机-摘机操作;当被呼叫方占线以致 3 次试拨无效

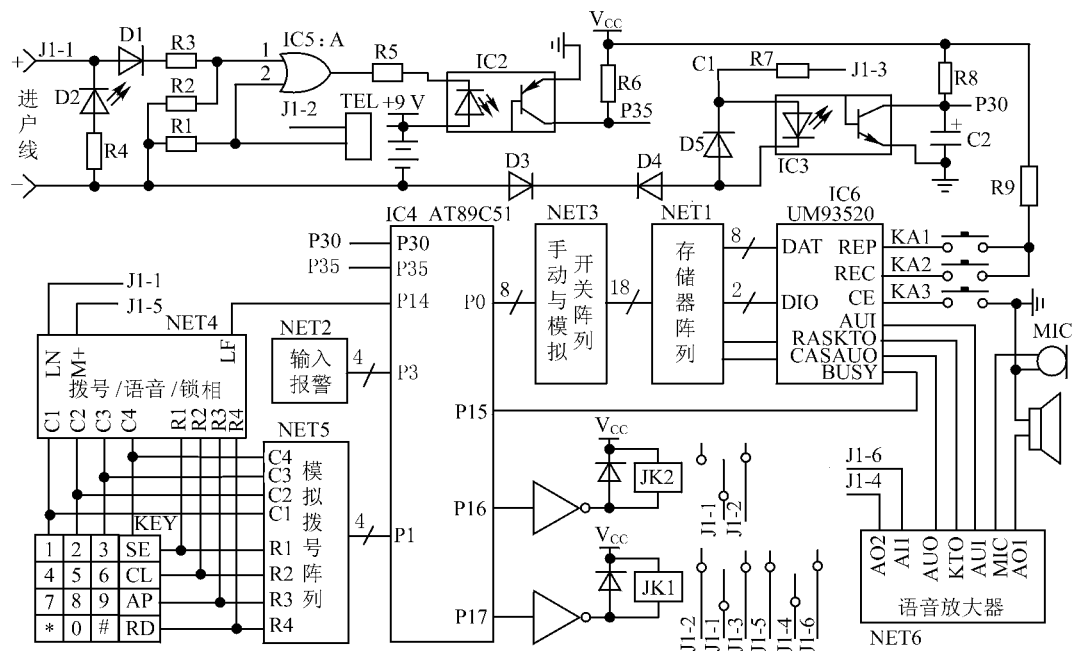


图 7-13 简化电路原理图

或振铃 8 次无人摘机时,自动挂机后顺拨下一待呼号码;当被呼方摘机时,系统在 0.5 s 内送出完整的报警语音并在对方挂机时及时退入拨号循环拨打下一电话号码。已拨通的号码在后面的拨号循环中剔除。摘挂机由 JK1 模拟完成。拨出的双音频 (DTMF) 经话音网络放大后被送到电话线路 (LN) 上,拨出号码可预先通过系统键盘存入拨号 IC 中。其号码存储保持电流由系统电源供给。

3. 线路振铃、可信音检测与自动应答

挂机时, J1-1 与 J1-3 常闭。当线路铃流出现时,由 IC3, C1, R7, D3~D5 等组成的铃流脉冲检测电路通过 P30 端口向单片机发送振铃次数脉冲。当振铃数达到程序预置值时,系统模拟摘机, J1-1 与 J1-2、J1-4 与 J1-5 闭合,话路切向语音网络 (LN)。语音网络的送话输入 (M+) 来自基本语音系统的语音放大输出端 AO2。锁相电路在话路上检出 450 Hz 可信音的调制脉冲后送 MCU 测试并处理。当振铃致系统模拟摘机时,单片机先将主人留言语音送到语音网络,若无忙音信号,说明对方尚未挂机,则在放音结束后启动自动录入操作并以“嘟”声通知对方留言。16 s 录入结束或全过程中忙音检测有效时关闭放 (录) 音并模拟挂机。

4. 语音录入与自动放音

语音录放系统由主电路 IC6 即录制与重放电路、数据存储阵列、模拟及手动开关阵列、语音放大电路等组成。由 MIC 输入的音频信号经语音放大器放大后送入 IC6 的 AUI 口按 16 kbit/s 的速率被压缩成语音数据存入 DRAM 中,同时 DRAM 刷新操作由 IC6 操作完成。放音时,由 DRAM 中取出的语音数据经 IC6 被还原成音频信号从 AU0 口输出。录入过程可通过对手动开关阵列的操作或由单片机 AT89C51 通过模拟开关阵列自动操作完成。对 IC6 的片选端 CE 的有效控制及对电话线路可信音的准确鉴别,可使自动报警语音在被呼叫方摘机时及时并完整放送,放送时最长可重复放音 3 遍。

5. MCU 系统控制

系统控制由 AT89C51 及 I/O 扩展电路组成。单片机 AT89C51 的 P0 口用于对语音自动录入的模拟操作。P10~P13 通过模拟拨号阵列完成对自动存储拨号的模拟操作。P14 为电话线路中 450 Hz 可信音的调制波检测(由锁相器输出)输入。P15 为语音电路工作状态的检测输入。P16 为室内电话机的通断控制输出,用于自动存储拨号时切断室内电话机,以保证系统可靠工作。P17 为模拟摘挂机控制输出。P31~P34 用于 4 路用户可定义的警情采样输入, P30,P35 分别为电话线路振铃检测输入及户外线异常采用输入。

7.4.3 控制程序框图

电话语音报警的控制程序框图分别如图 7-14、图 7-15 所示。其中,图 7-14 是系统复位以及应答操作处理的程序段框图,图 7-15 是报警处的程序框图,它包括火警、匪警、盗警和急救的处理。

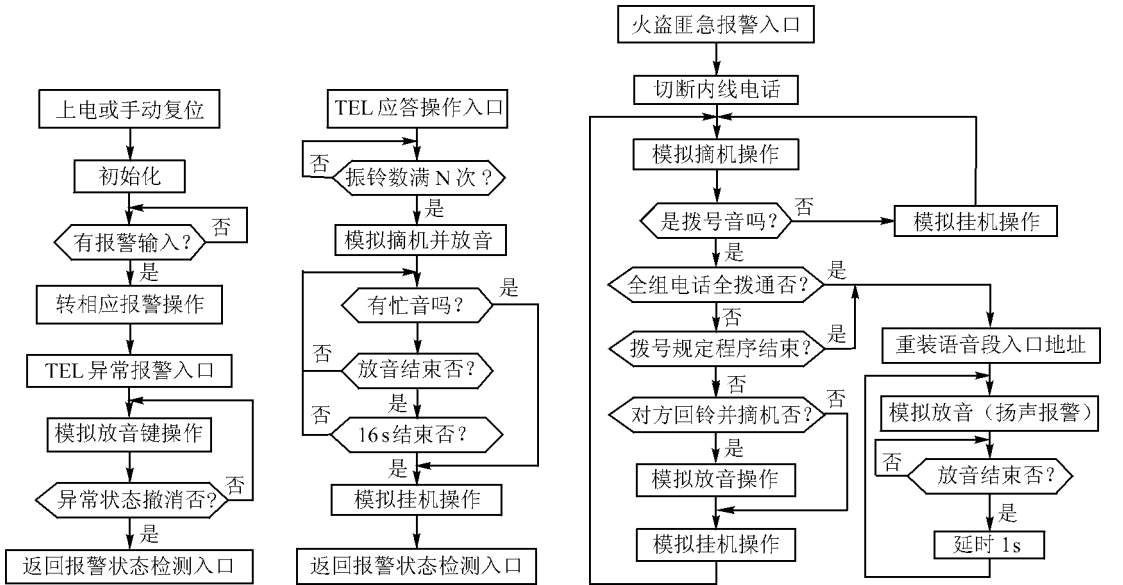


图 7-14 系统复位及应答处理框图

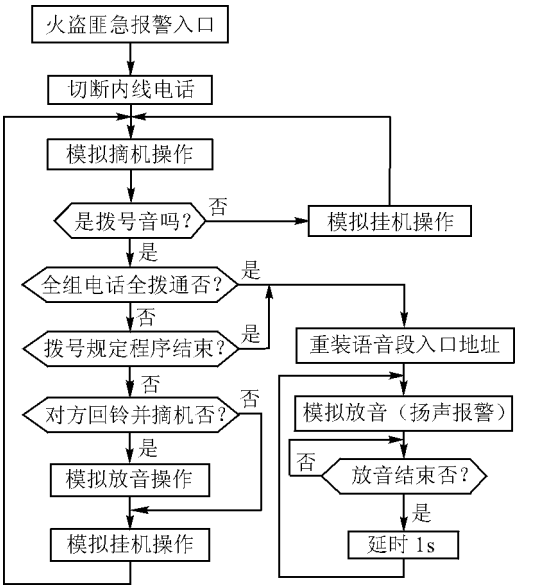


图 7-15 报警处理程序框图

系统运行时单片机 AT89C51 不断采样 P3 端口,当某 1 位端口状态有效时,系统转入相应处理子程序。遇有 4 类警情发生时,单片机 AT89C51 根据系统设定循环拨打存储号码。号码一经拨通即有人接听,便从循环中剔除。每个号码可连续试拨 3 次,全组号码可试拨 3 个循环。当全组号码均被拨通或 3 个循环结束,系统退出拨号并自动将报警语音切入外接扬声器以进行报警,扬声器可置于室外以进行室外扬声报警。一旦进入该循环,只有手动复位即按 RST 键方可解除。

7.5 电池充电器

随着小型放录机、随身听、手机等便携式电子产品的迅猛发展,对可重复充电的高性能电池的需求越来越大。为了提高电池的使用寿命,需要有效而合适的充电器。

在目前,广泛使用的可重复充电的电池有镍镉电池(NiCd),镍金属氧化物电池(NiMH)和锂离子电池(Li-ion)等。这 3 种电池的有关性能如表 7-4 所列。

表 7-4 电池的一些性能

性 能	NiCd 电池	NiMH 电池	Li-ion 电池
充放电次数	1 000	800	1 000
月自放电百分比	15	20	6
内 阻	低	中	高
放电速率/c	>10	<3	<2

对于驱动电机的用途,可用镍镉电池,因为其内阻较低,故可提供较大的驱动或启动电流。而对于手机和计算器等长时间小电流的产品,则可以用内阻较大的镍金属氧化物电池或锂离子电池。为了充分发挥电池的功能,必须对电池的充电过程进行合理控制。

7.5.1 电池的充电速率

在电池充电过程中,有一个极其重要的概念及指标,这就是充电速率。

电池的充电电流 I 与其容量 Q 之比称为充电速率: V_{ch}

$$V_{ch} = \frac{I}{Q} \tag{7-1}$$

其中: I 为充电电流,单位为安培 A;

Q 为电池容量,单位为安时 Ah。

充电速率 V_{ch} 单位用 C 表示,从量纲的角度看,由于电流单位为 A,容量的单位为 Ah,故 C 的量纲为 1/h 或 h^{-1} ,即小时的倒数。

对于一个容量为 10 Ah 的电池,如果以 1 A 电流对其充电,则充电速率 V_{ch} 为 0.1 C。因为

$$V_{ch} = \frac{I}{Q} = \frac{1\text{ A}}{10\text{ Ah}} = 0.1\text{ C h}^{-1} = 0.1\text{ C}$$

对于一个容量为 1 Ah 的电池,如果以 1 A 电流对其充电,则充电速率 V_{ch} 为

$$V_{ch} = \frac{1\text{ A}}{1\text{ Ah}} = 1\text{ C}$$

从上面对两种容量不同的电池,以相同的充电电流 1 A 充电时所产生的充电速率 V_{ch} 不同中可知:10 Ah 的电池充电速率为 0.1 C,1 Ah 的电池的充电速率为 1 C。而 C 是时间小时的倒数,则可知 0.1 C 时,需要充电 10 h;1 C 时需要充电 1 h。

一般而言,充电速率大于 1 C 时称为快速充电方式。而充电速率为 0.1~0.05 C 则称为标准充电方式。

在实际应用中,过度充电会对电池造成永久性损坏,所以,对充电的速率需要作一定的限定,以保证不会因过度充电而损坏电池。由于镍金属氢化物电池对充电过度较为敏感,所以一般要把对其的充电速率控制在 1.33 C 以下。不过镍镉电池对过度充电的敏感性不是很强,所以,对其充电速率可允许高于 4 C。对于锂离子电池,其对过度充电也较为敏感,故对锂离子电池的充电速率也像镍金属氢化物的充电速率一样需要控制在 1.33 C 以下。

在控制电池的充电速率的同时,还需要及时结束充电,这样,才能做到真正保护电池。

7.5.2 电池的充电特性

不同类型的电池是由不同的材料和结构制成的,因此它们有着不同的性能,特别是有不同的充电特性。这些充电特性向充电设备提供了充电的依据和控制的目标。下面分别对镍镉电池、镍金属氢化物电池及锂离子电池的充电特性加以介绍。

1. 镍镉电池的充电特性

镍镉电池是采用恒流充电的。它可以无限期地用标准的 0.1 C 速率充电,但不会产生过大的温升。

在快速充电时,镍镉电池在充电时为了防止过度充电,既需要主要的充电终止方法,又需要辅助的充电终止方法,以确保安全。而其中一种终止方法必须和温度有关。

镍镉电池以 2 C 充电速率进行充电时,其电压和温度曲线如图 7-16 所示。

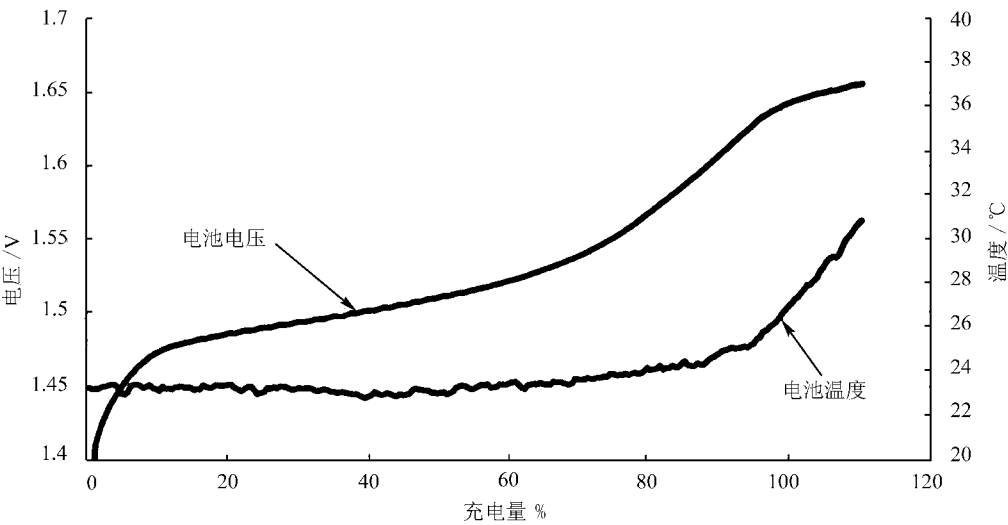


图 7-16 NiCd 电池 2 C 充电特性

从图 7-16 中看出:镍镉电池以充电速率 2 C 充电时,其电压特性分成 4 个不同的阶段。在充电量小于 10% 时,电压上升速度较快;在充电量处于 10%~70% 阶段,电压的上升速度较慢,并且基本保持一个稳定的上升率;在充电量处于 70%~100% 时,电压的上升速度变高,并且也大致保持一个恒定的上升率;在到达充电量为 100% 之后,电压的上升速度又稍微变低,但也保持近一个恒定值。

这时的温度特性分两个阶段,当充电量少于 100% 时,温度保持在 24 °C 左右;当充电量到达 100% 时,电池的温度就以较快的速度上升。

图 7-17 给出镍镉电池以充电速率 1 C 充电时的电压和温度特性。可见其特性和以 2 C 充电时的特性变化相近似。

2. 镍金属氢化物电池的充电特性

镍金属氢化物电池和镍镉电池一样,也是采用恒流充电的。

由于 NiMH 电池对过度充电敏感,过充电会损坏电池,减少可充放电的次数,所以,对

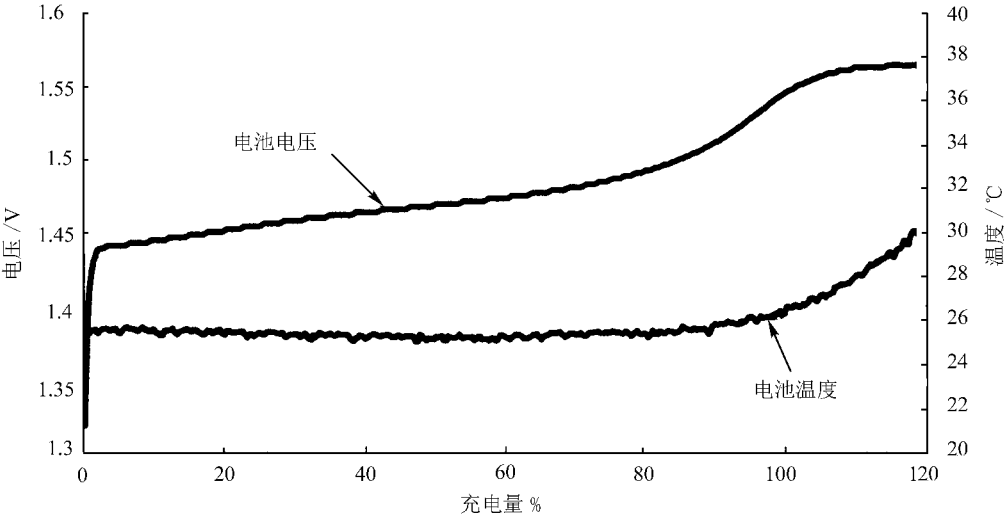


图 7-17 NiCd 电池 1 C 充电特性

NiMH 电池一般以标准的 0.1 C 充电速率充电。而接近充电量达 100% 时,还要把充电速率降到 0.025 C,以防止过充电,又可以抵消自放电的损失,维持电池的电量。

在快速充电时,NiMH 电池最好以 1 C 左右的速率充电。

NiMH 电池以 1.33 C 充电速率进行充电时的电压温度特性如图 7-18 所示。

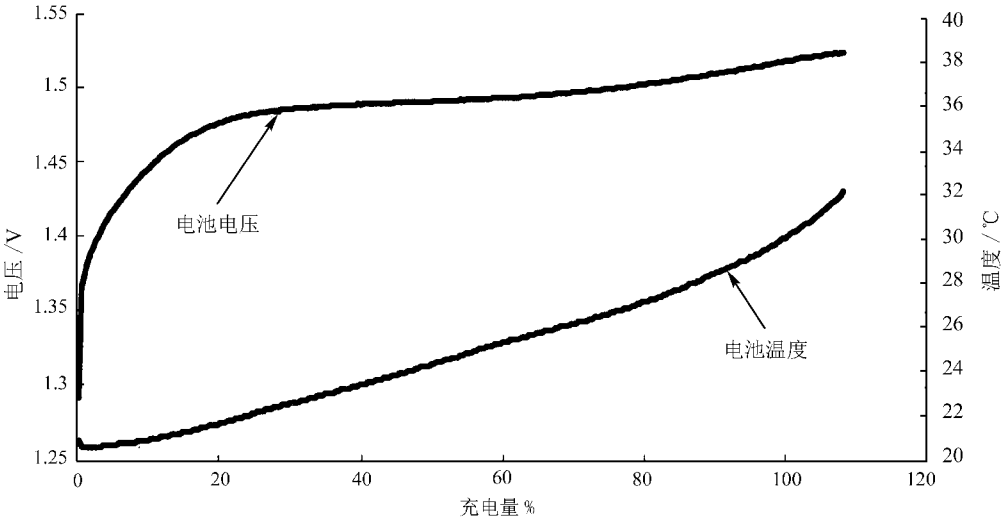


图 7-18 NiMH 电池 1.33 C 充电特性

3. 锂离子电池的充电特性

锂离子电池和镍镉电池、镍金属氢化物电池不同,它是采用恒压限流方式充电的。

锂离子电池比较容易受到过度充电、深放电和短路这 3 种情况的损害。所以,在使用中要特别注意。但是,由于锂离子电池有容量大、自放电速度低等优点,所以较为受欢迎。

锂离子电池的充电特性如图 7-19 所示。它是用充电速率 1.25 C 对锂离子电池充电时的电压温度特性。

在图 7-19 中可以看出：锂离子电池在充电的初期电压会快速上升，当充电容量达到 50% 以上时，其电压基本上恒定在最高电压上不变。而温度在初期会有一个下降的过程，随着充电量的增加，温度又会回复上升；温度的谷点大约在充电量为 50% 左右。当充电量达到 100% 后，温度又微有下降。

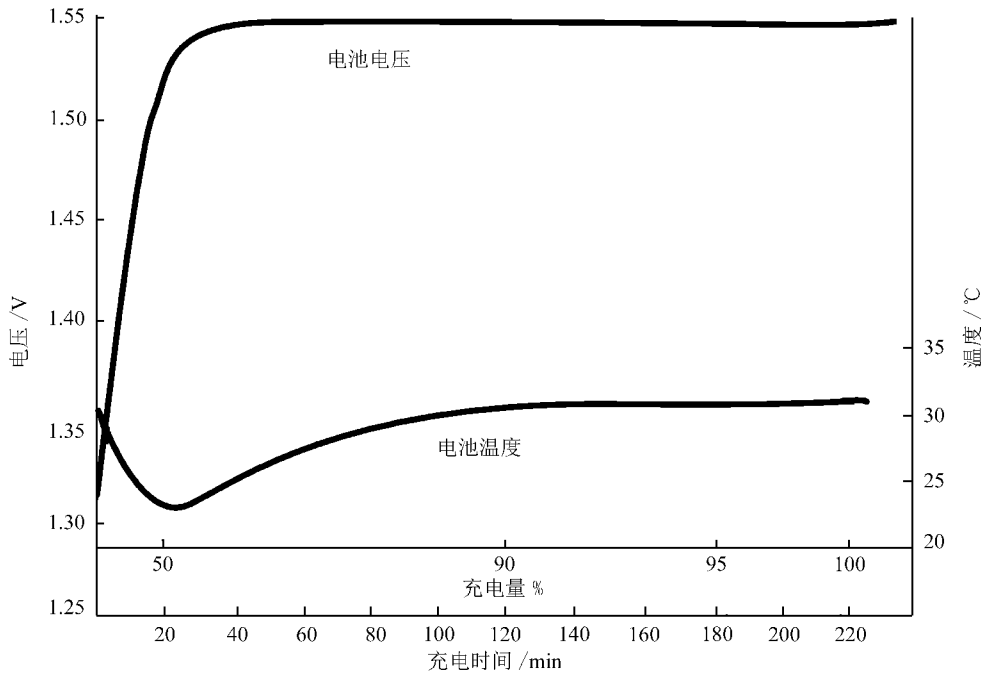


图 7-19 Li-ion 电池 1.25 C 充电特性

7.5.3 充电终止状态检测

为了防止过度充电，必须对充电的终止状态进行检测，以期在电池充电达到容量的 100% 时能马上停止充电过程，保持电池不被过度充电所损害。

电池充电达到自身容量的 100% 时会产生各种物理现象和特定的状态。对这些状态的检测可以判别电池是否处于充电终止时刻。

充电终止状态的检测有如下若干种方法。

1. -ΔV 法

有一些类型的电池，在恒流快速充电过程中，端电压会不断升高，但到达容量的 100% 时，会产生特有的端电压下降现象，一般而言，每节电池下降 -5 mV 左右。利用这一特殊现象来检测电池充电终止状态的方法称为 -ΔV 法。

NiCd 电池充电过程就会产生这种特殊现象，故 -ΔV 法可用于 NiCd 电池的充电终止状态检测。

2. 零 dV/dt 法

有一些类型的电池,在充电达到容量的 100% 时,不会产生端电压下降的现象,但是,其端电压在继续充电时不会继续上升,也就是有零电压上升率现象,即 $dV/dt=0$ 。利用零电压上升率现象来检测电池充电终止状态的方法称为零 dV/dt 法。

NiMH 电池充电过程就会产生零 dV/dt 现象,同时,当 NiCd 电池的 C 率较低时,其满容量端电压下降也很微小而不易检测。这些充电过程的终止状态可用零 dV/dt 法检测。

3. ΔT 法

有一些类型的电池,在充电的过程中电池外壳的温度会不断地缓缓上升,而且上升的速率有越来越快的趋势。当充电接近容量的 100% 时,外壳的温度和环境温度之差会达到一定的差值,通常达 10°C 左右。利用电池外壳温度和环境温度的温差来检测电池充电终止状态的方法称 ΔT 法。

4. dT/dt 法

电池在充电的过程中,其外壳温度会随着充电过程慢慢上升。到达 100% 的充电量时,电池的外壳的温度会迅速上升,这时外壳的温度变化率较大。利用异常温度变化率来检测电池充电终止状态的方法称为 dT/dt 法。

5. T 法

电池在充电时,外壳温度不断缓慢上升,但未到达满额充电时,其温度不高;到达了容量的 100% 左右时,温度达到一定的数值。利用电池外壳的温度值来检测电池充电终止状态的方法称为 T 法。

6. V 法

电池在充电时,其端电压会随充电过程不断上升,当充电量达到容量的 100% 左右时,其端电压会比额定电压高一定数值。这时的电压称为满容量电压。利用满容量电压来检测电池充电终止状态的方法称 V 法。

7. $-\Delta T$ 法

有一类电池,在充电过程中温度会产生变化,在充电超过 50% 时,温度会缓慢上升,但到达容量的 100% 时,电池外壳温度反而会稍微有所下降。利用充电后期外壳温度会下降的特性来检测电池充电终止状态的方法称为 $-\Delta T$ 法。

这种 $-\Delta T$ 法可用于检测 Li-ion 电池的充电终止状态。

在实际应用中,为了保险起见,一般都采用多种方法进行充电终止检测,通常用两种方法;而简单的充电器也可能只用一种方法。

7.5.4 安全充电的措施

过度充电会使可重复充电的电池内部产生不可逆转的化学过程,从而对电池造成不可恢复的损害,这些损害有:

- 电池容量退化、减少;
- 电池寿命缩短;
- 电池过热爆炸。

为了保证电池的安全可靠,应该考虑相关的防护方法,这些方法有如下多种。

1. 采用多种充电终止检测方法

采用 2 种或 2 种以上的充电终止检测方法,以保证可靠停止充电。

2. 采取极限保险措施

电池最高电压限制:在充电过程中,如果电池端电压超过额定电压一个限定的值,则马上停止充电。

电池过热保护:在充电中,电池外壳温度达到一定温度极限,如 $40\text{ }^{\circ}\text{C}$ 时,则马上停止充电。

充电超时保护:在充电过程中,充电的时间超过常规充电时间一定限度,如 1.5 倍时,则马上停止充电。

3. 禁止低温启动

电池在温度过低时其性能变化很大,故在电池外壳温度低于 $10\text{ }^{\circ}\text{C}$ 时,禁止充电。

4. 充电终止预报

对充电终止点进行预报,可以使充电过程及早采取低充电速率逼近 100% 容量。在快速充电过程中,端电压的上升速率会随后期充电状态而变高,接近充电终止点时达最大值,即有 $dV/dt=\max$,此后又会逐渐变低;到达充电终止点时则近似为 0,甚至转为负值,也就是电池的端电压不再上升甚至反而下降。故采用电压上升率最大判别,可以提早预报终止点。

5. 充电初期禁止 $-\Delta V$ 检测

有些电池在开始充电的初期数分钟之内,其端电压会出现短暂的电压下降,为了防止误判充电达到终止点,故这时应禁止 $-\Delta V$ 检测。

6. 自放电补偿

可重复充电电池一般都会有自放电现象,即在电池充电之后,其电容量会因电池内部放电而减少。NiCd 电池和 NiMH 电池在常温时每日大约损耗 1%;每当温度升高 $10\text{ }^{\circ}\text{C}$ 时,则自放电损耗增加 1 倍。为此,在快速充电到达充电终止点时,并不切断充电回路,而是自动转为所谓“点滴充电”状态,也就是把 C 率减少到 $C/50\sim C/10$ 之间继续进行充电。这样,可以补偿电池的自放电损耗,另外,还可随电池充电速率减少而导致温度回落,使到存电量增加,从而保持电池足量充电状态。

在点滴充电时, NiCd 电池可采用 $C/30\sim C/10$ 的充电速率,而 NiMH 电池采用 $C/50$ 的充电速率较为安全。

7. 开、短路保护

对电池的接触端子进行检测,以判别电池是否处于开路或短路,并且自动切断充电回路。

7.5.5 充电电路结构

利用廉价单片机 AT89C1051 可制作简单的智能型充电器,它使用安全可靠,能防止电池极化,还可以对一般镍镉或镍氢电池进行脉冲调制式快速充电;也可以对那些用一般普通充电器无法充电的受损电池进行良好的充电;对电极严重极化但内部电解液含量正常的电池也能充电,起到修复电池、延长电池寿命的作用。在充电的同时能对电池电压进行开路检测,当达到预设的终点电压时,可自动停止充电,无需人工看管充电过程,可确保电池安全充电并避免电池的损坏。

一般而言,1 节镍镉电池的标称电压为 1.2 V ,放电终止电压为 1 V ,约为总容量的 15%,

而充电的终止电压随环境温度在 1.4~1.55 V 之间变化,一般环境温度在 20℃ 时其终止电压为 1.5 V,平均充电速率为 C/10~C/15。在保证电池升温不超过 65℃ 时,允许用较大的电流充电以缩短充电时间。但是单一大电流充电或者充满电后仍以 C/10 充电率充电,均会造成内部电解液中水分强烈电解蒸发,生成氢气或氧气。内部压力的增大,迫使氧气扩散至负极与镉反应生成氧化镉,造成电池有效极板面积减少,致使电池容量下降。如果长期按这种方式充电,即使是全新的电池也会在短时期内造成容量减小甚至失效。因此,良好的电池充电器应具有安全可靠的自动控制装置,能防止过充电造成电池的永久性损坏。这里介绍的充电器采用 5 充 1 停的大电流脉冲充电方式,能够有效地克服电池极化现象,达到快充和延长电池使用寿命的目的。其最大的优点是,由于采用软件控制,所以可以随意改变充电调制比率,可任意设定充电的时间,这样和电压检测配合,对电池可起到双重保护的作用。

充电电路的结构如图 7-20 所示。它由单片机 AT89C1051、显示电路、电池电压检测电路、电压调节电路等组成。

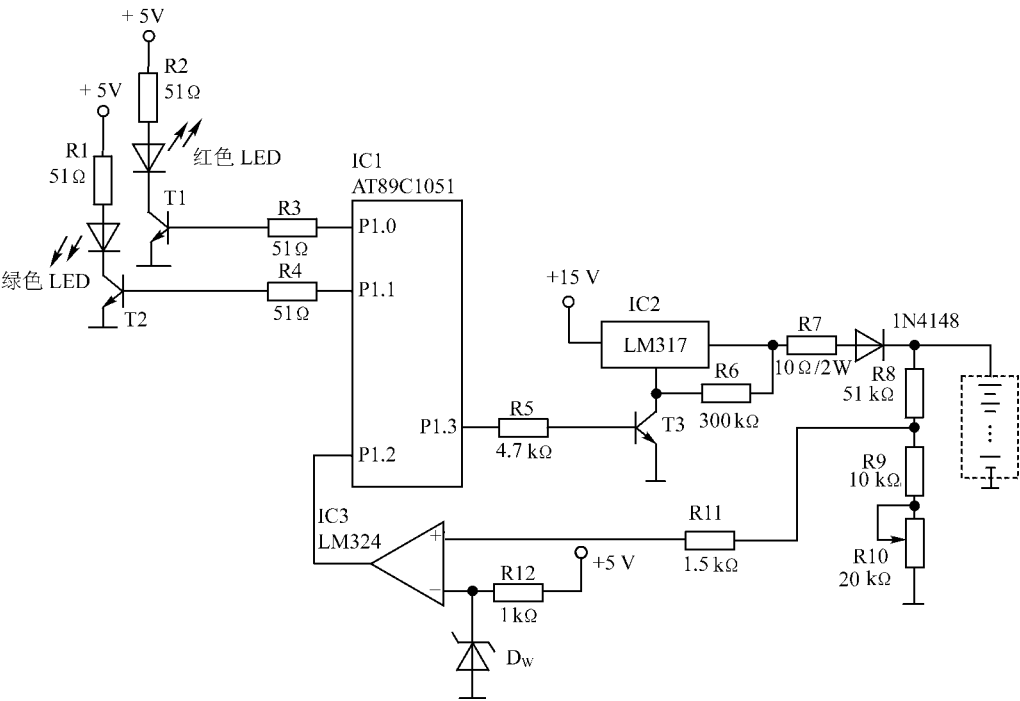


图 7-20 电池充电电路结构

下面对充电电路的各个部分分别进行介绍。

1. 电池电压检测电路

电池电压检测电路由采样电路和比较电路组成。

采样电路由 R8,R9,R10 组成,在这 3 个电阻形成的分压电路中,取电池电压在 R9,R10 上的分压作为采样信号,送到比较器 IC3 的正输入端。

比较电路由 IC3,R11,R12 以及 Dw 等组成。R12 和稳压管 Dw 构成基准比较电压,并送入 IC3 的负输入端。IC3 是比较器 LM324,它对正负输入端的信号进行比较。当正输入端信号小于基准比较电压时,IC3 输出“0”电平;当正输入端信号大于基准比较信号时,IC3 输出“1”

电平。利用这个电路可以检测电池充电终止电压。

2. 显示电路

R1,R2,R3,R4 以及与之相连的发光二极管、三极管组成 2 个显示器,用于显示相应状态。单片机 AT89C1051 的端口输出高电平时,对应的三极管导通,发光二极管发光。

3. 电压调节电路

该电路由 R5,R6,R7,IC2 以及相关的二极管、三极管组成。

IC2 是电压调节集成电路 LM317。其输出电压在 1.2~37 V 之间可调,输出电流高达 500 mA,内部有过载保护电路,有短路电流限流电路,外部封装采用 3 脚的三极管形式封装。

LM317 的标准应用电路如图 7-21 所示。其 1 是 LM317 的调节引脚,3 是输入引脚,2 是输出引脚。在 LM317 的输出引脚 2 与调节引脚 1 之间保持 1.25 V 的参考电压 V_{ref} ,并且引脚 2 为正端。

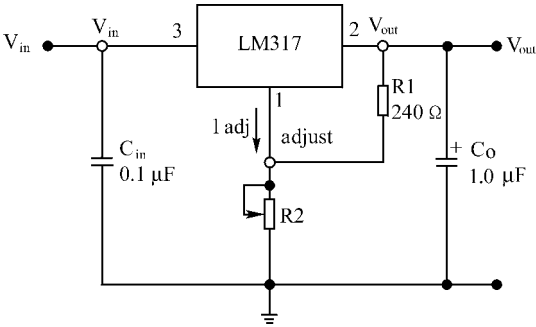


图 7-21 LM317 的标准应用

很明显,流过 R1 的电流 I_{R1} 为

$$I_{R1} = \frac{V_{ref}}{R1} \tag{7-2}$$

这个电流 I_{R1} 和引脚 1 流出的电流 I_{adj} 流过 R2,则有

$$V_{R2} = (I_{R1} + I_{adj})R2 \tag{7-3}$$

从而有输出电压 V_{out}

$$V_{out} = V_{ref} + V_{R2} = V_{ref} + (I_{R1} + I_{adj})R2$$

则

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1} \right) + I_{adj} \times R2 \tag{7-4}$$

从式(7-4)可知, $I_{adj} \times R2$ 是偏差项,否则输出电压 V_{out} 可以通过调节 R1,R2 的值来改变,因为 V_{ref} 是恒定值 1.25 V。

为了减少误差,LM317 的 I_{adj} 小于 100 μA ,并且它是一个恒定值。

从式(7-4)可以知道,如果忽略 I_{adj} ,则有

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1} \right) \tag{7-5}$$

在式(7-5)中,假定 R1 是一个恒值,那么,调节 R2 的值就可以调节输出电压 V_{out} 的值。但是,这个调节的先决条件是 $V_{in} > V_{out}$ 。当 R2 大到一定程度,按式(7-5)计算产生 $V_{out} > V_{in}$ 时,则调节无效,输出 $V_{out} = V_{in}$ 。而当 R2 小到“0”时,根据式(7-5),则有 $V_{out} = V_{ref}$ 。

4. 单片机 AT89C1051

单片机 AT89C1051 用 4 个端口引脚控制充电器工作。P1.2 用于输入电压检测信号；P1.0, P1.1 用于控制显示电路；P1.3 用于控制 LM317 的工作情况。当 P1.3=1 时, LM317 的引脚 1 接地, 这时它输出的电压 $V_{\text{out}} = V_{\text{ref}}$, 故不能对电池充电；当 P1.3=0 时, LM317 的引脚 1 所接的三极管截止, LM317 输出的电压为 V_{out} 。

$$V_{\text{out}} = V_{\text{ref}} \left(1 + \frac{R_{T3}}{R6} \right) \quad (7-6)$$

其中: R_{T3} 是三极管 T3 截止时的电阻。

$R6$ 为 300Ω , 如图 7-20 所示。

故

$$V_{\text{out}} = 1.25 \text{ V} \left(1 + \frac{R_{T3}}{300 \Omega} \right)$$

由于 R_{T3} 的阻值较大, 所以, V_{out} 输出的电压可以对电池充电。

单片机 AT89C1051 是通过 P1.3 端口输出高电平或低电平来控制 LM317 的工作的。只要调节 P1.3 输出高低电平的占空比, 就可以较好地控制充电速率, 使电池以较好的方式充电。由于 V_{in} 的电压为 15 V, 故 LM317 的输出 V_{out} 可以对多节电池充电。在改变电池的节数时, 只要在程序中改变高低电平输出的占空比即可。

7.5.6 充电器的控制软件

下面给出的充电器控制软件可以对 6 节电池进行充电。充电时采用脉冲方式, 对电池充电 5 ms, 停止充电 1 ms。对电池电压的检测周期为 1 min。在检测电池电压时, 暂停对电池的充电。同时, 红色和绿色发光二极管亮, 表示处于电池电压检测状态。

程序执行中显示状态有如下 4 种。

- (1) 程序开始执行: 红、绿色发光二极管同时闪烁 3 次。
- (2) 正在充电: 红色发光二极管亮。
- (3) 正在检测电池电压: 红、绿发光二极管同时亮。
- (4) 充满电: 绿色发光二极管亮。

程序执行的主要功能如下。

在程序首部, 红、绿发光二极管执行 3 次闪烁, 表示开始准备充电。接着对电池电压进行检测。检测时是对 P1.2 的输入电平进行判别, 如果 P1.2 为“0”, 说明未充满电, 则进入充电过程; 如果 P1.2 为“1”, 说明已充满电, 则程序在检测程序段循环, 不断检测电池电压, 一旦电池电压有回落, 则又进入充电过程。在充电时, 在 P1.3 端口输出 5 ms 的“0”电平, 使 LM317 的引脚 1 相连的三极管截止, LM317 输出电压 V_{out} 对电池充电, 接着输出 1 ms 的“1”电平, 使三极管导通, LM317 输出很低的电压 $V_{\text{out}} = V_{\text{ref}}$, 故无法对电池充电。这样, 使电池的充电在 5:1 的占空比中进行。在主程序执行中, 每 1 min 才去对电池电压进行 1 次检测。

充电器的功能较为简单, 它实际上是采用 V 法一种方法进行充电终止判别的, 而满容量电压值由稳压管 D_w 设定。

程序的清单如下:

1. 主程序

```
RED EQU P1.0 ; 正在充电
GREEN EQU P1.1 ; 充电结束
TEST EQU P1.2 ; 电池检测
ON_OFF EQU P1.3 ; LM317 调整端通断控制

ORG 0000H
LJMP MAIN
ORG 0100H
MAIN: MOV P1, #00H
FLASH3: MOV P1, #00H ; 灭
        ACALL T1S
        CPL RED
        CPL GREEN ; 亮
        ACALL T1S
        CPL RED
        CPL GREEN ; 灭
        ACALL T1S
        CPL RED
        CPL GREEN ; 亮
        ACALL T1S
        CPL RED
        CPL GREEN ; 灭
        ACALL T1S
        CPL RED
        CPL GREEN ; 亮
        ACALL T1S
        MOV PL, #00H ; 灭
        SETB ON - OFF ; 关闭 LM317 输出
        ACALL T1S
        ACALL T1S
        ACALL T1S
        ACALL TST - BAT ; 调用电池检测子程序
GO1: MOV R6, #100
GO2: MOV R5, #100
; 100 * 100 * (1+5) = 60 000 ms = 1 min
START: CLR ON - OFF ; 接通 LM317 输出
        ACALL T5MS
        SETB ON - OFF
        ACALL T1MS
        DJNZ R5, START
        DJNZ R6, GO2
        ACALL TST - BAT ; 调用电池检测子程序
```

SJMP GO1

2. 电池检测子程序

```
TST - BAT: SETB ON - OFF ; LM317 接地, 关闭输出
            SETB RED
            SETB GREEN
            ACALL T1S
            JNB TEST, BAT - LOW
            ACALL T1S
            JNB TEST, BAT - LOW
            ACALL T1S
            JNB TEST, BAT - LOW
BAT - HI:   CLR RED
            SETB GREEN
            JB TEST, TST - BAT
; 电池充满, 等待. 电压下降后返回主程序继续充电
BAT - LOW: CLR GREEN ; 继续充电
            SETB RED
            RET
```

3. 1 ms 延时子程序

```
T1MS:      MOV R2, # 01
OK3:        MOV R1, # 250
OK4:        DJNZ R1, OK4
            DJNZ R2, OK3
            RET
```

4. 5 ms 延时子程序

```
T5MS:      MOV R2, # 05
OK1:        MOV R1, # 250
OK2:        DJNZ R1, OK2
            DJNZ R2, OK1
            RET
```

5. 1 s 延时子程序

```
T1S:        MOV R3, # 200
WE:         ACALL T5MS
            DJNZ R3, WE
            RET
```

第八章 自动控制领域的应用

自动控制的应用十分广泛,无论在军事、工业、农业、交通、运输、纺织、航空、航天、机器人等领域,还是在通信、计算机外围设备上,都需要对某些对象执行自动控制,以期一个系统能自动进行工作,同时,又能把工作的质量控制在人们所希望的精度或要求之内。

自动控制已经有几十年的历史,它已经从最初的经典控制理论发展到今天的现代控制、模糊控制、神经网络控制、粗糙控制、先进控制、可拓控制。而这些控制都有着其相应的理论基础和方法学。在这一章中,介绍 89C 系列单片机在自动控制领域的应用,内容着重于单片机的应用技术,而不是这些控制方法的理论问题。故在这一章介绍的例子中,所涉及的都是单片机控制的逻辑结构,以及其程序编制的思想和结构。了解了逻辑结构之后,用软件方法就可以实现各种不同的控制过程及控制方法。

8.1 双坐标步进电机控制

双坐标步进电机控制,就是在 x 轴方向控制一台步进电机,在 y 轴方向控制一台步进电机。这两台步进电机同时驱动同一个对象,使对象在一个平面上以任意曲线运动。有时,为了方便起见,也称为二维步进电机控制。双坐标或二维步进电机控制在线切割机,摄象监视器,灯光控制等场合都有大量的应用。

在这一节中,介绍用工业控制计算机和单片机控制的二维步进电机工作系统。这个系统是由工业控制计算机发出控制命令,通过与单片机通信,按命令单片机产生控制步进电机运转的脉冲信号。按照发出不同的控制命令,使二维步进电机分别作正转、反转、快转、慢转和停止等动作,还可自成系统自动运行。系统由于采用 89C 系列单片机,电路简单可靠,结构紧凑。对于不同型号的步进电机的控制,不需改变硬件电路,通过修改软件,即能实现多种控制,灵活方便,通用性强,成本低,因此具有广泛的应用前景。

8.1.1 控制系统的结构原理

二维步进电机微机控制系统的工作原理框图如图 8-1 所示。

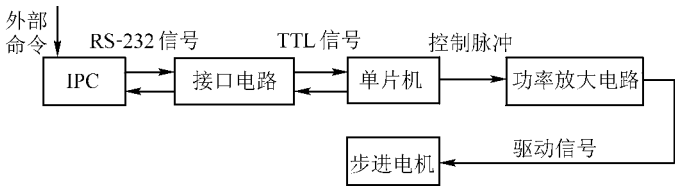


图 8-1 二维步进电机控制系统原理框图

工业控制计算机是二维步进电机的控制系统的主机,负责从键盘接收外部命令,由串行口输出后,再经接口电路发送到单片机,然后接收单片机回送的命令数据并进行比较,如相同,表

明单片机正确接收了命令；如不同，在屏幕上显示出错信息。工业控制计算机在控制结束后，可做其他事务处理工作。

接口电路实现工业控制计算机串行口信号与单片机 TTL 信号之间的转换，以实现工业控制计算机与单片机正常通信。

单片机负责从工业控制计算机上接收命令，并将其转换成控制脉冲信号，从并行口发出到步进电机驱动电路。在脱离工业控制计算机的控制时，还要保证系统能按一定程序自动运行。

功率放大电路是将单片机并行口输出的控制脉冲信号进行电压和电流放大，驱动步进电机，使步进电机随着不同的脉冲信号作正转、反转、快转、慢转等等。步进电机是执行动作设备，当脉冲按一定顺序输入步进电机的各个相时，步进电机就能实现不同的运动状态，从而可带动固定在其上的其他设备如摄像头等做相应运动。

二维步进电机控制系统电路原理图如图 8-2 所示。在图中，工业控制计算机用英文缩写 IPC 表示。系统由工业控制计算机 IPC、通信接口、AT89C51 和步进电机驱动功率放大电路组成。

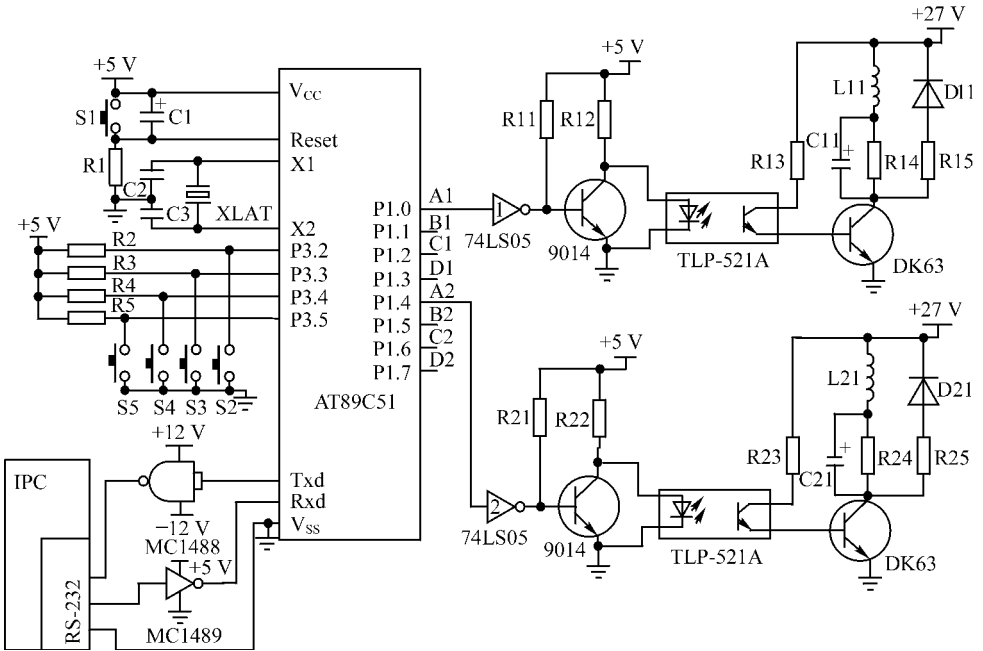


图 8-2 二维步进电机控制系统电路原理图

工业控制计算机 IPC 通过串行口与单片机通信，由于后者输入输出电平为 TTL 电平，而工业控制计算机配置的是 RS-232 标准串行接口，因此必须进行电平转换。电平转换由 MC1488 和 MC1489 执行，由 MC1488 将 TTL 电平转换为 RS-232 标准电平，工作电压为 +12 V/-12 V。MC1489 则把 RS-232 标准电平转换为 TTL 电平，工作电压为 +5 V。

单片机采用 ATMEL 的 AT89C51 单片机，片内含 4 KB 的 ROM，由 P1 口输出控制脉冲信号，P1.0~P1.3 为一组，P1.4~P1.7 为一组，分两路各控制一台步进电机。P3 口中 P3.0 为串行输入，P3.1 为串行输出，P3.2~P3.5 设置为行程保护开关，作二维步进电机正反方向最大行程保护。晶振频率选用 12 MHz。AT89C51 以中断方式执行工业控制计算机命令。

功率放大电路中采用 74LS05 将单片机 P1 口脉冲信号进行放大,经 9014 控制光电耦合器,隔离后,由功率管 DK63 驱动步进电机的各相绕组,图 8-2 中 L××即为步进电机的各相线圈。

在系统中步进电机采用 4 相步进电机,并以双 4 拍运行,其中对应 P1.0~P1.3 的为横向步进电机的 4 个相(A1~D1),对应 P1.4~P1.7 的为纵向步进电机的 4 个相(A2~D2)。工作电压为 27 V,驱动代码如表 8-1 所列,倒序输出,电机则反向运行。步进电机的运行速度由 P1 口输出的控制脉冲的频率决定。

表 8-1 4 相 4 拍驱动代码表

相	拍				
	1	2	3	4	1
A	1	0	0	1	1
B	1	1	0	0	1
C	0	1	1	0	0
D	0	0	1	1	0

8.1.2 步进电机控制

步进电机的工作状态是步进式转动。一般步进电机的工作都是由单极性直流电源供电的。只要对步进电机的各相绕组按合适的时序方式进行通电,就能步进转动。

步进电机的步进过程可以用图 8-3 来说明。在图中所示的是一个 4 相反应式步进电机,在定子中,各相的一对磁极只有一个齿,也就是磁极本身,所以,4 对相磁极则有 8 个齿。在转子中有 6 个齿,分别称 0,1,2,3,4,5 齿。直流电源通过开关 S_A, S_B, S_C, S_D 分别对步进电机的 A,B,C,D 相绕组轮流通电。

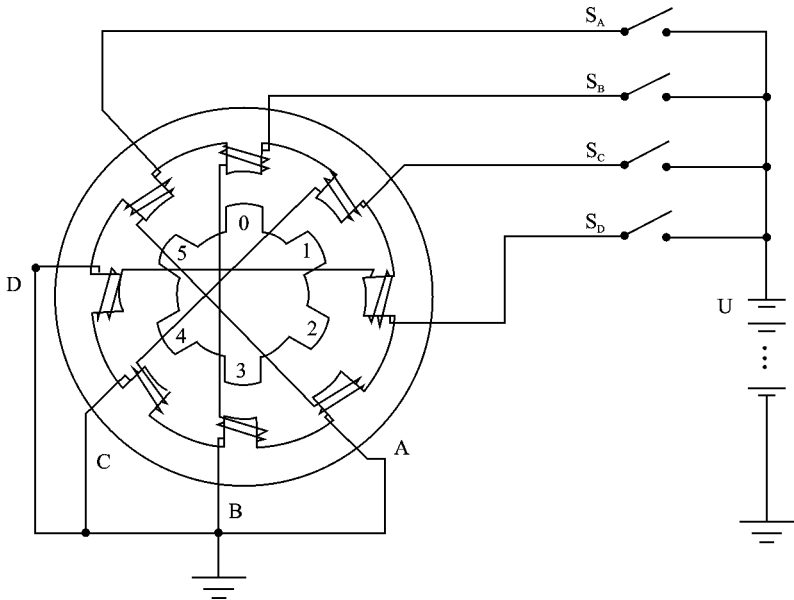


图 8-3 步进电机步进示意

开始时,开关 S_B 接通电流 U ,则 B 相磁极和转子 0,3 号齿对齐,同时,转子的 1,4 号齿就和 C,D 相绕组磁极产生错齿,2,5 号齿就和 D,A 相绕组磁极产生错齿。

当开关 S_C 接通电源, S_B 断开时,由于 C 相绕组的磁力线和 1,4 号齿之间磁力线的作用,使转子转动,令 1,4 号齿和 C 相绕组的磁极对齐。这样,则使 0,3 号齿和 A,B 相绕组产生错齿,2,5 号齿和 A,D 相绕组产生错齿。

当 S_D 开关接通电源 U , S_B 开关断开电源时,结果和上面类同,这时,只是 2,5 号齿和 D 相绕组磁极对齐,其余错齿。

很明显,只要按 A—B—C—D—A 相顺序轮流通电,则转子会沿 A,B,C,D 方向转动一个齿距位置。在图 8-3 中,由于转子齿数为 6,故齿距角为 60° 。转动了一个齿距,也就是转动了 60° 。

对于一个步进电机,如果其转子的齿数为 N ,它的齿距角为 θ_z ,有

$$\theta_z = 2\pi/N \tag{8-1}$$

当步进电机运行 R 拍可以使转子转动一个齿距位置,而步进电机每一拍实行了一次步进,则可以得到步进电机的步距角 θ_s ,并且有:

$$\theta_s = 2\pi/NR = 360^\circ/NR \tag{8-2}$$

其中: N 为转子齿数, R 为步进电机工作拍数。

图 8-3 所示的 4 相反应式步进电机,由于采用 4 拍工作方式,故步距角 θ_s 为

$$\theta_s = 360^\circ/(6 \times 4) = 15^\circ$$

当转子有 30 齿时,采用 4 拍工作方式,则步进电机的步距角为

$$\theta_s = 360^\circ/(30 \times 4) = 3^\circ$$

步进电机为工作可靠,通常采用双 4 拍的工作方式,即通电顺序为 AB—BC—CD—DA,如表 8-1 所列。另外,通电频率不能太高,否则会产生失步。

步进电机采用双 4 拍工作方式时,电源通电的时序和波形如图 8-4 所示。在图中,实线是电压波形,虚线是电流波形,最上边的是时钟波形。从图中看出,在任一时刻,总是有两相的绕组通电,有两相绕组断电,而且,通电顺序是按 AB—BC—CD—DA—AB 这种组合方式循环的。对于 4 相步进电机来说,其控制的方式有单 4 拍,双 4 拍,还有 8 拍方式。8 拍方式是把单 4 拍和双 4 拍相结合的结果。要实现 8 拍控制对步进电机有一定的要求。对于要求控制精度不是很高的地方,一般采用双 4 拍就可以了。而 8 拍控制可以减少步距角,从而提高控制精度。

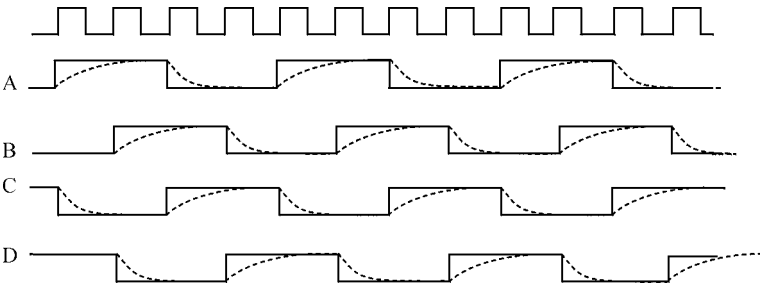


图 8-4 双 4 拍工作时序波形

8.1.3 步进电机控制软件框图

系统启动后,单片机执行主程序,自动控制二维步进电机按预定方式运动。当需要改变运动状态时,由工业控制计算机从键盘接收命令,再传送给单片机,单片机中断主程序执行,接收工业控制计算机的命令,实时控制步进电机按命令的要求运动,命令执行结束后,重新执行主程序。

为保证一定的传送距离和准确性,工业控制计算机与 AT89C51 通信时约定:波特率为 1 200 波特,信息格式为 8 个数据位和 1 个停止位,传送方式是工业控制计算机采用查询方式收发数据,单片机则采用中断方式接收信息。

AT89C51 单片机串行口工作于方式 1,允许接收,SCON 值为 50H,定时器 T1 工作于方式 2,TMOD 值为 20H,时间常数 TH=TL 为 0CBH。

系统的程序框图如图 8-5 所示。

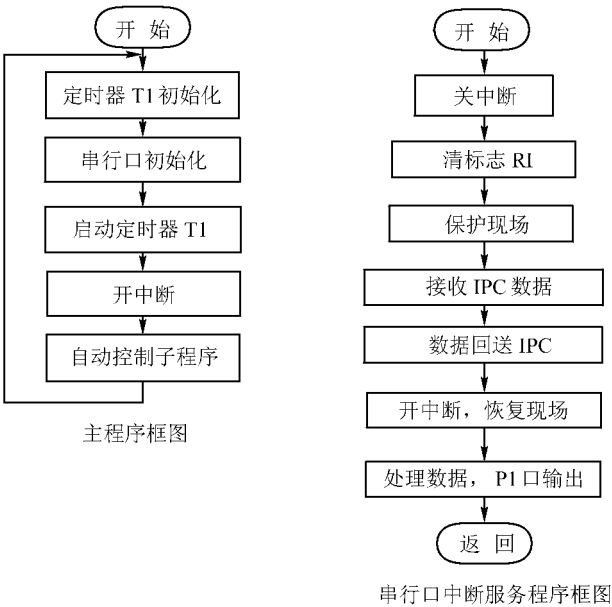


图 8-5 AT89C51 的工作程序框图

为了提高工业现场抗干扰能力,单片机的主程序中,串行口及定时器 T1 的初始化操作重复进行。

图 8-5 给出了单片机 AT89C51 的工作程序框图。它包括两个程序,一个是主程序,另一个是串行口中断服务程序。

主程序用于执行初始化工作,其中包括定时器 T1 的初始化和串行口的初始化。在初始化之后,串行口工作于方式 1,定时器 T1 工作于方式 2。接着主程序启动定时器 T1 开始工作,并且开中断,等待串行口中断一请求,则马上响应。

串行口中断服务程序在响应中断之后执行,首先保存单片机的有关现场数据,接着接收工业控制计算机送来的命令,并把命令回送,以便工业控制计算机对单片机接收情况进行校验。接着把命令所指明的动作数据从 P1 端口输出,控制步进电机执行对应命令。

图 8-6 给出工业控制计算机的程序框图,这个程序用于工业控制计算机和单片机 AT89C51 进行通信,即发送命令。这个程序的功能如下:接收键盘输入的工作命令;向单片机 AT89C51 发送命令信号;接收单片机回送的命令信息;校验单片机接收命令的正确性;显示单片机未能正确接收命令的状态等。

在实际应用中,可把双坐标步进电机控制用于各种二维方向控制。较为典型的应用例子就是采用 AT89C51 为控制核心的二维步进电机带动摄像头运转的控制系统,这个系统使用 28BF001 型步进电机,驱动电压 27 V,相电流 0.8 A,步距角为 3°,减速比为 1:100。对该系统工作设定步进电机以快慢两种速度运行,分别是 12°/s 和 3°/s,则 P1 口输出驱动代码的频率分别应为 400 Hz 和 100 Hz,通过调用延时程序实现。系统自动运行时,是以慢速运作的。

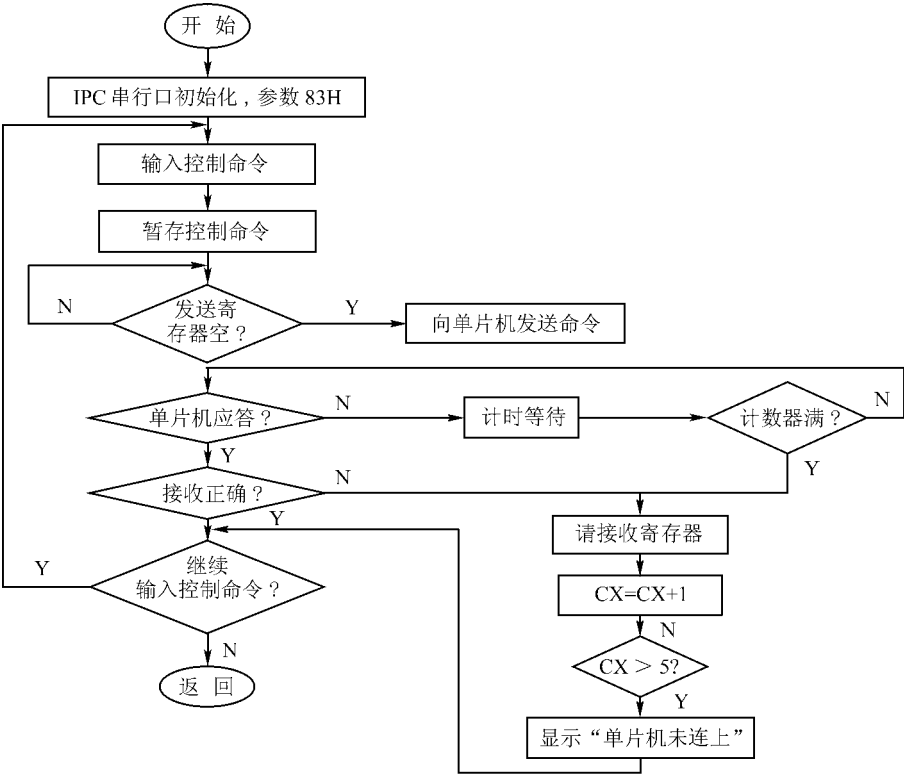


图 8-6 工业控制计算机程序框图

由于要考虑电线缠绕问题,故步进电机只能在 350°左右的范围内来回运转,而不能沿一个方向无限制地运行,因此设置了行程保护开关。行程保护开关闭合后系统转为自动返回运行。

8.2 交流电源电压组合控制

在工业或实验室中通常要用到各种电压值的交流电源,以满足一些特殊的工作过程或实验的需要。在实际中,很多交流电源的电压往往是用两种方法调节的。一种是用自耦变压器滑动式手动调压器进行变压器调节;另一种是用晶闸管对交流电源进行处理调压。

采用自耦变压器调压时,由于依赖手动控制滑动触点位置以达到输出不同电压的目的,所以,其调压过程响应速度慢,调压精度低,准确电压选择往往要慢慢来回滑动触点才能完成,同时机械结构有时会接触不好,可靠性较差。

用晶闸管对交流电源处理调压有两种方法,第一种方法是晶闸管移相的方法,这种方法的缺点在于输出的电压是不连续的非正弦波,有较大的高次谐波影响电网。第二种方法是采用双向晶闸管截波的方法,这种方法采用过零触发,在若干个周波为一个周期的时间内,使一些周波不导通,一些周波导通。当晶闸管的导通周波数和截止周波数为某一种比例时,则输出相应的电压。这种方法有两个缺点,其一是输出电压断续,其二是输出电压较难做到任意值。它的优点在于不会产生高次谐波。

无论是自耦变压器还是传统的晶闸管处理电压的方法,都难以使人感到满意。在这一节介绍一种新的交流电压调节方法。它采用过零触发方法,选择不同的输出绕组进行组合,产生

所需的输出正弦电压。这种方法的不足之处是要采用变压器,并且要求次级有多组绕组,同时,还要较多的双向晶闸管。但是,它有两个十分明显的优点:第一,输出电压是正弦波,不会对电网产生干扰;第二,输出电压可以以 0.5 V 的精度进行调节。

8.2.1 交流电压组合控制基本原理

交流电压组合控制是和传统数字控制不一样的控制方式。它利用变压器次级各种绕组的不同组合产生所需的输出电压。最基本且最典型的组合控制是按二进制数值进行组合控制。

按二进制数值进行组合控制可简称 2-W 控制。它的基本原理如下。

对于一个字长为 l 的二进制串,它可以组成有 2^l 种组合数字,即组合数字为 N

$$N = \sum_{i=1}^l 2^i \times D_i \quad (8-3)$$

其中 D_i 为 0 或 1, $i=1 \sim l$ 。

从式(8-3)可知:只要设置 l 个数值即 $2^i, i=1 \sim l$,利用 D_i 取值为 0,1 这种开关特性,只要对不同的数值执行开或关,就可以组合各种组合数字。

这种组合数字的处理可以用于交流电压的控制。其实现过程分两部分。

(1) 在交流变压器的次级分成多个绕组,每个绕组的电压值等于二进制的数值。

(2) 每个绕组的电压值可以和其他绕组的电压迭加,也可以忽略,并由开关控制。

对于单相 220 V 电源进行电压组合控制时,其变压器的次级绕组分别为 1 V, 2 V, 4 V, 8 V, 16 V, 32 V, 64 V, 128 V 共 8 个。组合控制的变压器及开关情况如图 8-7 所示。

在图 8-7 中,输出电压 U_o 从 128 V 绕组和 1 V 绕组连出。每个绕组有 2 个开关,分别接绕组的两条引出端,2 个开关的状态永远处于相反状态,即 1 个接通,则另一个开路,反亦反之。每个绕组的 2 个开关不接绕组的触点则连在一起,并且和上一绕组的下端引线相接。很明显,在图 8-7 中,有 8 组开关 $S_i, S'_i, i=1 \sim 8$,并且, S_i 和 S'_i 的状态永远相反。当 $S'_i, i=1 \sim 8$ 全部接通时, U_o 输出电压为 0;当 $S_i, i=1 \sim 8$ 全部接通时, U_o 输出电压为 255 V。

很明显,只要用 8 位二进制数去对 $S_i, S'_i, i=1 \sim 8$ 进行控制,就可以得到所需的输出电压。8 位二进制数的高位控制 S_8, S'_8 ,按顺序一直下去,最低位控制 S_1, S'_1 ,则用于控制 8 位二进制数据的值,就是电压的输出值。

电压组合控制可用多种的绕组结构进行组合。图 8-7 所示的绕组和组合控制是一种格式化形式,在实际中也可以用各种非格式化形式。一种非格式化形式的结构如图 8-8 所示。在这种结构中,变压器 T 的初级输入电压 U_1 为 220 V 交流电网电压,次级有 4 个具有 3 等分抽头的绕组和一个单绕组,每个绕组的等分电压分别为 0.5 V, 2 V, 8 V, 32 V, 单绕组电压为 128 V。G1~G18 是 18 只双向晶闸管,作为双向开关元件。若控制电路每次只使每个绕组中的一个晶闸管导通,其他晶闸管都关断,即可得到一种输出电压。假设 G1, G5, G9, G13, G17 导通,其他晶闸管都关断,则次级输出电压 U_o 为 0;若 G2, G5, G9, G13, G17 导通,其他晶闸管关断,则 $U_o = 0.5 \text{ V} \cdots$ 按这种方式控制晶闸管的通断,18 只晶闸管有 512 种组合状态,每个状态都可输出 0.5 V 的整倍数电压,即输出电压变化范围为 0~255.5 V。

为了确保调压器输出完整的正弦波,避免晶闸管在通断转换时对电网产生冲击,晶闸管控制电路采用电压过零触发方式。在硬件上,用从电网取出的过零脉冲作为晶闸管的同步触发信号;在软件上,用此作中断源,单片机在中断服务程序中输出控制码来切换晶闸管的通断。

从表 8-2 看出:显然,单片机输出的控制码与调压器的输出电压呈线性关系,理论上只需把给定输出电压乘以 2 作为控制量即可。但是,如果电网电压波动以及负载变化导致输出电压和给定电压不相等时,就需要通过电压反馈控制来实现自动调节,保证输出电压等于给定电压值。

输出电压为了克服电网波动而采用的控制方法为比例积分调节,即 PI 控制。

一般的 PID 控制器的形式如下

$$U(t) = K_p \left[e(t) + \frac{1}{t_i} \int_0^t e(t) dt + t_d \frac{de(t)}{dt} \right] \quad (8-4)$$

其中: $e(t)$ 是给定值和输出值的偏差;

$U(t)$ 是控制器输出的控制量;

K_p 是比例系数;

t_i 是积分时间常数;

t_d 是微分时间常数。

在单片机执行控制时,采样周期为 T ,采样程序为 $k, k=0, 1, \dots, k$ 。并且采样周期 T 比电网电压波形小得多,则可用矩形法计算积分,用向后差分取代微分,则有

$$\int_0^t e(t) dt = \sum_{i=0}^k e_i T \quad (8-5)$$

$$\frac{de(t)}{dt} = \frac{e(k) - e(k-1)}{T} \quad (8-6)$$

把式(8-5), (8-6)代入式(8-4),则可以得到 PID 控制器差分表达式

$$U(k) = K_p \left[e(k) + \frac{1}{t_i} \sum_{i=0}^k e_i T + t_d \frac{e(k) - e(k-1)}{T} \right] \quad (8-7)$$

其中: $U(k)$ 是在采样时刻 k 时的输出控制量;

$e(k)$ 是在采样时刻 k 时的偏差;

$e(k-1)$ 是在采样时刻 $(k-1)$ 时的偏差。

同理可以得出在 $(k-1)$ 时刻的差分方程:

$$U(k-1) = K_p \left[e(k-1) + \frac{1}{t_i} \sum_{i=0}^{k-1} e_i T + t_d \frac{e(k-1) - e(k-2)}{T} \right] \quad (8-8)$$

式(8-7)减去式(8-8)则有

$$U(k) = U(k-1) + K_p \left\{ e(k) - e(k-1) + \frac{T}{t_i} e(k) + \frac{t_d}{T} [e(k) - 2e(k-1) + e(k-2)] \right\} \quad (8-9)$$

把式(8-9)写成增量形式,令

$$\Delta U(k) = U(k) - U(k-1)$$

则有

$$\Delta U(k) = K_p [e(k) - e(k-1)] + \frac{K_p T}{t_i} e(k) + \frac{K_p t_d}{T} [e(k) - 2e(k-1) + e(k-2)] \quad (8-10)$$

令

$$K_I = \frac{K_p T}{t_i} \quad (8-11)$$

$$K_D = \frac{K_P t_D}{T}$$

(8-12)

则式(8-10)可写成

$$\Delta U(k) = K_P[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)]$$

(8-13)

其中： K_P 称比例系数；
 K_I 称积分系数；
 K_D 称微分系数。

在式(8-13)的右边中， $K_P[e(k) - e(k-1)]$ 为比例项， $K_I e(k)$ 为积分项， $K_D[e(k) - 2e(k-1) + e(k-2)]$ 为微分项。

在图 8-8 所示的非格式化组合控制中，为了克服电网波动等带来的输出电压 U_O 的变化，故采用 PI 控制对输出电压进行调节，这时，PI 控制算法如下

$$\Delta U_{PI}(k) = K_P[e(k) - e(k-1)] + K_I e(k)$$

(8-14)

其中： K_P 为比例系数；
 K_I 为积分系数；
 $e(k)$ 为 k 时刻的电压偏差， $e(k) = U_I - U_O(k)$ ；
 $e(k-1)$ 为 $k-1$ 时刻的偏差， $e(k-1) = U_I - U_O(k-1)$ 。

很明显，组合控制和 PI 控制相结合，可以保证输出电压的稳定性和准确性，特别是由于组合控制能保证输出较理想的正弦波，这一点是其他控制所不容易实现的。

8.2.2 控制系统的逻辑电路

图 8-8 所示的非格式化组合控制的控制系统逻辑电路如图 8-9 所示。控制系统的核心是单片机 AT89C51，系统包括过零检测电路、调压控制电路、交流输出电压检测电路、显示及给定电路等 4 个不同功能的逻辑电路。

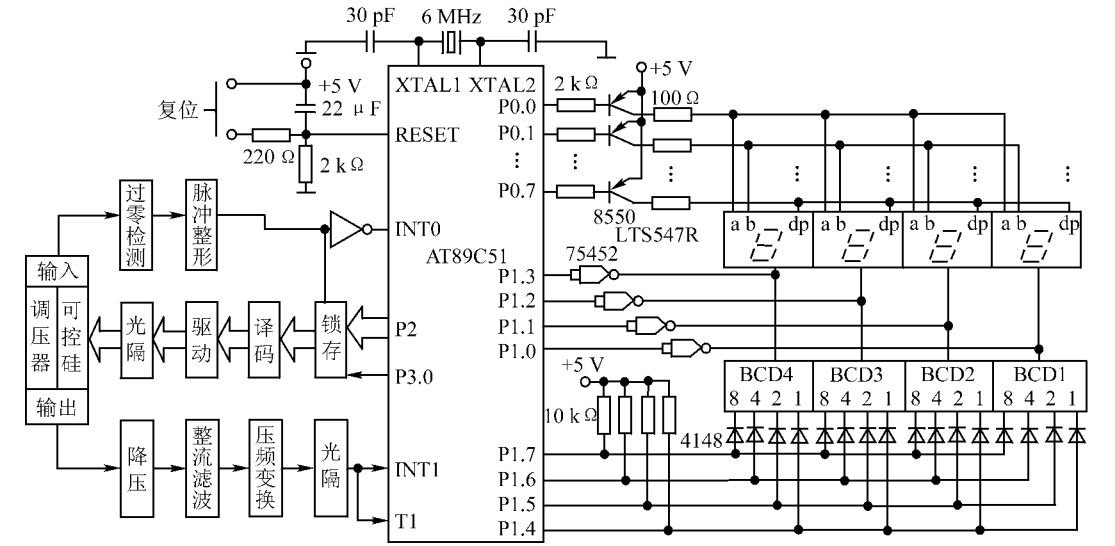


图 8-9 交流电压组合控制逻辑电路

1. 过零检测电路

过零检测电路用于对电源电压的过零点进行检测,以便对晶闸管进行同步触发控制。

交流电压过零检测电路如图 8-10 所示。它的结构非常简单,由 2 个光电耦合器 GD1、GD2,2 个反相器以及 2 个电阻组成。

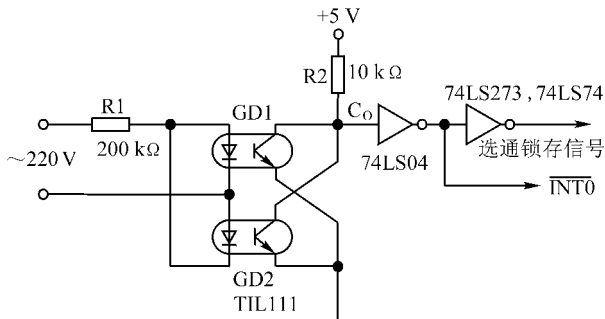


图 8-10 过零检测电路

输入交流电压为来自变压器初级的 220 V 电源信号。通过一个 200 kΩ 的电阻限制之后接到 2 个光电耦合器的输入端,故通过光电耦合器输入端的电流最大为 15 mA,从而保证了光电耦合器的可靠工作。2 个光电耦合器以反向并联的方式相连,以分别检测输入交流电压的正半波和负半波。

工作时,变压器初级的 220 V 交流电压经电阻 R1 限流后直接加到 2 个反向并联的光电耦合器 GD1、GD2 的输入端。在交流电源的正、负半周,GD1 和 GD2 分别导通,C₀ 输出低电平,在交流电源正弦波过零的瞬间,GD1 和 GD2 均不导通,C₀ 输出高电平。该脉冲信号经非门整形后作为单片机的中断请求信号和晶闸管的过零同步信号。

在交流电压的过零点,在 $\overline{\text{INT0}}$ 输出端产生负脉冲送到单片机的外部中断输入端 $\overline{\text{INT0}}$,请求外部中断。同时,过零信号通过两个反相器之后产生正脉冲,送到晶闸管调压控制电路的 74LS273 和 74LS74 中去作为时钟信号,使 74LS273 和 74LS74 锁存来自单片机的信号,以便去触发相应的晶闸管。

2. 晶闸管调压控制电路

晶闸管调压控制电路是用于触发晶闸管导通的触发电路,在给定电压值的条件下,晶闸管的触发导通情况如表 8-2 所列。调压控制电路的逻辑结构如图 8-11 所示。

从图 8-11 可知:晶闸管调压控制电路由锁存器 74LS273、74LS74、译码器 74LS139、反相器 74LS04、驱动电路 74LS07 以及光电耦合器 MOC3041 等组成。其中 74LS273、74LS74 用于锁存单片机送来的触发控制码,控制码为 0~511,并以二进制数的形式从单片机的 P3.0、P2.7~P2.0 输出。

如图 8-11 所示,AT89C51 的 P2 口和 P3.0 经 74LS273 和 74LS74 锁存并过零同步后,用 2 片 74LS139 和反相器译出 18 根控制信号,通过 3 片 74LS07 驱动后,由高压过零型双向晶闸管光电隔离器 MOC3041 控制 18 只双向晶闸管实现对变压器次级绕组的组合控制。由于每组译码器控制一组次级绕组,任何时候每组绕组仅有一个晶闸管导通,故提高了系统的可靠性。另外,用光电隔离器驱动晶闸管,既省去了脉冲变压器,又使单片机和强电隔离。

在图 8-11 中,单片机的 P2.0、P2.1 用于控制 0.5 V 的次级绕组,P2.2、P2.3 用于控制

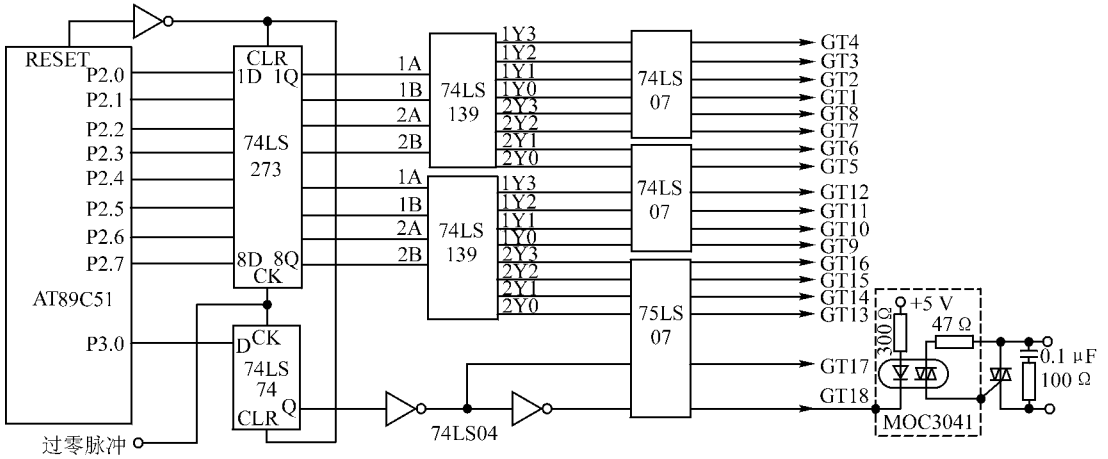


图 8-11 调压控制电路

2 V 的次级绕组,P2.4,P2.5 用于控制 8 V 的次级绕组,P2.6,P2.7 用于控制 32 V 的次级绕组,P3.0 用于控制 128 V 次级绕组。它们在编码译码后分别和 G1~G4,G5~G8,G9~G12,G13~G16,G17~G18 对应。由于在 P2 端口中,P2.0~P2.7 分别和 D0~D7 对应,故对次级绕组的控制,实际和表 8-2 中的 D0~D7 相应,同时,控制码最高位 D8 则和 P3.0 对应。

当 74LS07 的对应输出为“1”时,则光电耦合器 MOC3041 中的发光二极管不能导通,故对应晶闸管截止;当 74LS07 的对应输出为“0”时,光电耦合器中的发光二极管导通,流过的电流约为 12 mA,则其发光,从而触发晶闸管导通。

3. 输出电压检测电路

输出电压检测电路用于检测调压控制的输出电压大小,以确定其是否和给定的电压值一致。随后决定所用的控制信号,以校正输出电压的偏差。

输出电压检测电路如图 8-12 所示。它由运算放大器和相关的电阻、电容网络、V/f 变换电路、光电耦合器以及相关的电阻电容组成。

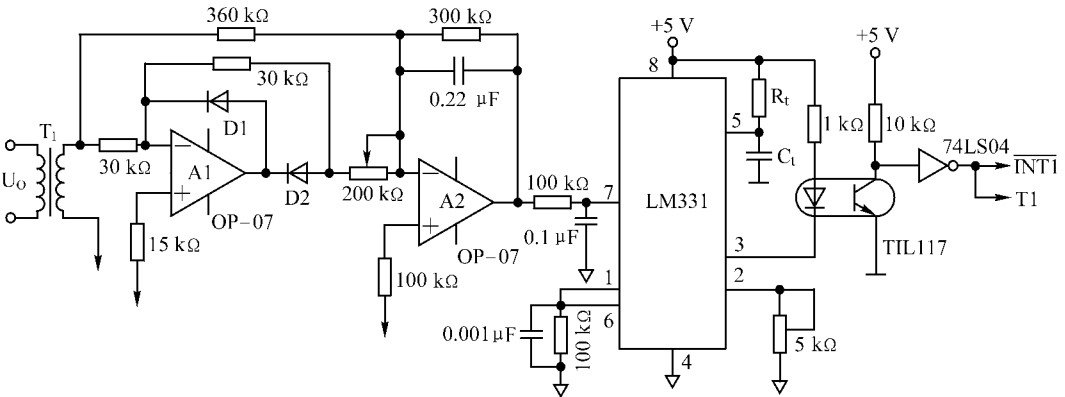


图 8-12 输出电压检测电路

如图 8-12 所示,调压器的交流输出电压经变压器 T₁ 降压后,用两个运放 A1,A2 进行整流滤波,使之变为与交流电压成比例变化的直流电压信号。为了实现强电隔离,没有采用常规

的 A/D 转换器,而是采用压/频变换电路 LM331 把电压变化转化为频率变化,经光隔后送入单片机,通过软件配合计算出相应的交流输出电压值。

输出电压检测电路产生的频率信号送到单片机的 $\overline{\text{INT1}}$ 输入端,通过外部中断 1 进行内部处理,另外,频率信号也送到定时器 1 的输入端 T1,由定时器 T1 进行处理。输入的频率信号通过外部中断 1 和定时器 T1 的共同作用,由软件处理最后确定输出电压的电压值。

4. 显示电路和给定输入电路

给定输入电路和显示电路的结构如图 8-9 所示。

给定输入电路由 4 个 BCD 码拨码盘和中功率放大器 75452 组成。拨码盘分别用 BCD4~BCD1 标志。当拨码盘拨到某一个数字位置时,如果对应的 75452 输出为“0”,则在 P1.7~P1.4 就会产生对应的 BCD 码。由于 P1.3~P1.0 的输出是采用扫描方式的,在某一时刻,只有一个 75452 的输出为“0”,故在检测 P1.7~P1.4 的 BCD 码时,就可以判别出是哪一位的 BCD 码。

显示电路由 4 个 7 段 LED 显示器、8 个晶体管、4 个 75452 反相器以及有关电阻组成。需要显示数字的显示码从 P0.7~P0.0 输出,分别对应于 7 段 LED 的各段及小数点。在 P1.3~P1.0 中输出的是扫描方式的“1”信号。故 4 个 75452 反相器轮流输出“0”电平。当 P0.7~P0.0 输出一个数字的显示码时,输出“0”电平的 75452 对应的显示器执行显示任务,其余显示器不显示任何内容。只要 P0 端口送出的显示码和 P1.3~P1.0 的“1”信号扫描信号同步,就可以在 4 个显示器上分别显示出所需的数字。

8.2.3 电压组合控制的软件

系统软件采用模块化法设计,用 MCS-51 汇编语言编写,包括主程序、中断服务程序以及 A/D 转换、数控调压、显示、给定输入、运算等子程序。其中主程序和 T0 中断服务程序框图如图 8-13 所示,其他有关子程序框图省略。

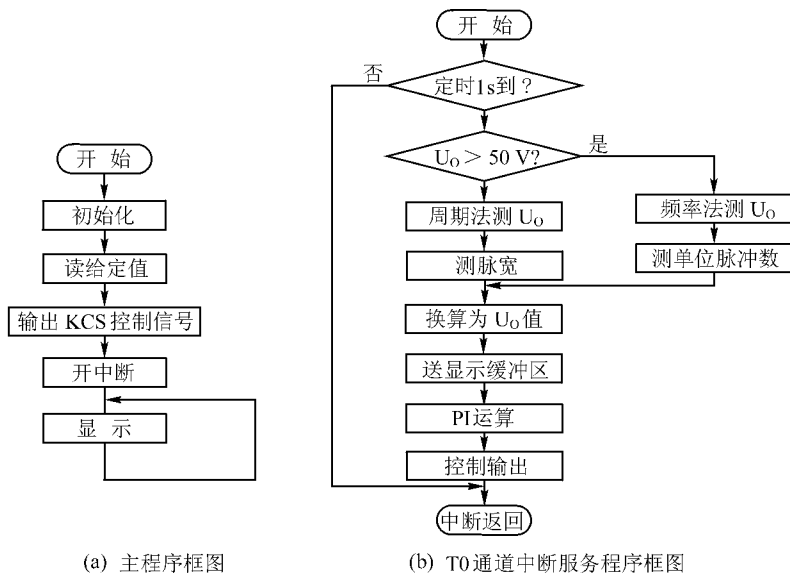


图 8-13 控制软件主程序、T0 中断服务程序框图

从图 8-13 看出: 主程序完成 T0、 $\overline{\text{INT0}}$ 、参数初始化、读给定输入值、输出 KCS 控制信号

以及显示输出电压;T0 定时 0.1 s,共定时 1 s,完成 U_o 的测量、显示准备、算法运算和控制输出。

$\overline{\text{INT0}}$ 中断服务程序完成过零控制输出,T1 中断和 $\overline{\text{INT1}}$ 配合完成 U_o 的测量。当 $U_o > 50 \text{ V}$ 时,用频率法测量 T0 定时 0.2 s 内的脉冲数,T1 设置为模式 1 计数器方式;当 $U_o \leq 50 \text{ V}$ 时,用周期法测脉宽,T1 设置为模式 1 定时方式, $\overline{\text{INT1}}$ 启动定时。

A/D 转换程序主要用于对输出电压 U_o 的检测结果进行处理,以求出输出电压 U_o 的值,以把其和给定电压进行比较,再根据偏差进行 PI 控制,从而实现输出电压的校正。

给定输入子程序用于对 BCD 拨码盘进行扫描和取给定值。工作时,主要是根据 P1.3~P1.0 进行扫描输出的信号,同步取拨码盘的输入值,输入的高 3 位拨码盘值是整数,低 1 位拨码盘值是小数。

显示子程序用于显示输出的电压值。它和给定输入子程序一样,工作时是根据 P1.3~P1.0 的扫描输出信号,同步输出显示码,那么对应的数字就会在相应显示器显示出来。高 3 位是整数电压值,低 1 位是小数电压值。

数控调压子程序用于把给定值转换成控制信号。由于给定值输入时是 BCD 码,所以需要把其转换成二进制码,然后送出到 P0 端口去控制晶闸管导通。4 位 BCD 码可以用下列公式进行处理,即

$$B = [[\text{BCD4}(2+8) + \text{BCD3}](2+8) + \text{BCD2}](2+8) + \text{BCD1}$$

其中: B 是转换之后的二进制数;

$\text{BCDi}, i=1\sim 4$, 是各位 BCD 码。

从上面可以看出,在程序执行中,只需执行左移 1 位、左移 3 位和加运算,就能实现 BCD 码和二进制数的转换。

8.3 地震数据采集系统

数据采集系统,从严格的定义上来说,应该是用计算机控制的多路数据自动检测或巡回检测,并且能够对数据实行存储、处理、分析计算以及从检测的数据中提取可用的信息,供显示、记录、打印或描绘的系统。

数据采集系统可以是一个独立的系统。例如一些智能仪表或自动化测试系统中的数据采集系统就是独立的系统。

数据采集系统也可以是一个系统的附属子系统。这时也称为数据采集子系统。例如,在过程控制系统中过程变量的检测通道。

数据采集系统一般由数据输入通道、数据存储与管理、数据处理、数据输出及显示这 4 个部分组成。输入通道要实现对被测对象的检测、采样和信号转换工作。数据存储与管理要用存储器把采集的数据存储起来,建立相应的数据库,并进行管理和调用。数据处理就是从采集到的原始数据中,删除有关干扰噪声、无关信息和非必要的信息,提取出反映被测对象特征的重要信息。另外,就是对数据进行统计分类,以便于检索;或者,把数据恢复成原来的物理量形式,以可输出的形态在输出设备上输出,如打印、显示、绘图等。数据输出及显示就是把数据以适当的格式进行输出和显示。

不同的数据采集系统有不同的要求和结构。但都是根据实际要求而制定的。数据采集系统在生产产品的参数自动检测,控制工程特性的自动检测,生产信息的自动检测,机器人的视觉、

听觉等多维信息检测,机器各种性能指标的检测,汽车整车性能的检测中都有很广泛的应用。

本节介绍地震观测中的地震数据采集系统。

8.3.1 地震数据采集系统的结构框图

在数字地震观测中,高精度、宽频带、大动态范围是数字地震观测的基本要求,也是数字地震观测的发展趋势,这就要求数字地震观测系统具有较高的分辨率。如果要求动态范围大于120 dB,就要求模拟数字转换的位数必须大于20位。目前,大于20位的A/D芯片生产厂家很多,例如ADI公司的AD7710,AD7716;CRYSTAL公司的CS5503,CS5321等。各种芯片的性能各异,用法也不同。这一节介绍用AT89C2051单片机作为控制器,与AD7716构成的廉价的地震数据采集系统。

地震数据采集系统的系统结构框图如图8-14所示。它由用于检测地震信号的传感器,检测信号放大器,22位A/D转换器AD7716,低噪声参考基准电压源AD780,单片机AT89C2051,RS-232接口芯片MAX232等组成。

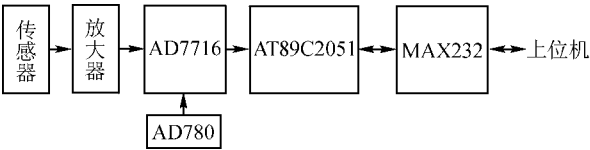


图 8-14 地震数据采集系统框图

传感器用于检测地震信号。由于地震信号的强弱与地震源离检测点的距离有一定的关系,所以要求传感器有一定的灵敏度。

检测信号放大器用于把传感器检测出来的地震信号加以放大,以达到一定的幅值利于进行A/D转换。放大器放大了之后的信号送到22位A/D转换器AD7716的模拟信号输入端AIN_i,*i*=1~4。放大器输出信号为-2.5~+2.5 V。

A/D转换器AD7716用于进行高精度A/D转换。它的外形引脚如图8-15所示。

AD7716是美国ADI公司的产品,具有4个独立通道,带宽高达584 Hz。该芯片分辨率为22位,在输入带宽为36.5 Hz时有效动态范围为111 dB;在输入带宽为584 Hz时有效动态范围为99 dB。AD7716芯片采用 $\Sigma\Delta$ 技术完成A/D转换,采用 $\Sigma\Delta$ 的A/D转换器采用过采样,然后进入数字滤波器,可降低模拟信号的带宽,满足采样定理的要求。AD7716采用5 V供电,最高转换频率为2 232 Hz,模拟输入信号为双极性,其范围为 ± 2.5 V,A/D转换结果以串行补码形式输出。AD7716为高精度A/D转换率,非常适合于信号频率不太高,但是要求高精度的信号采集

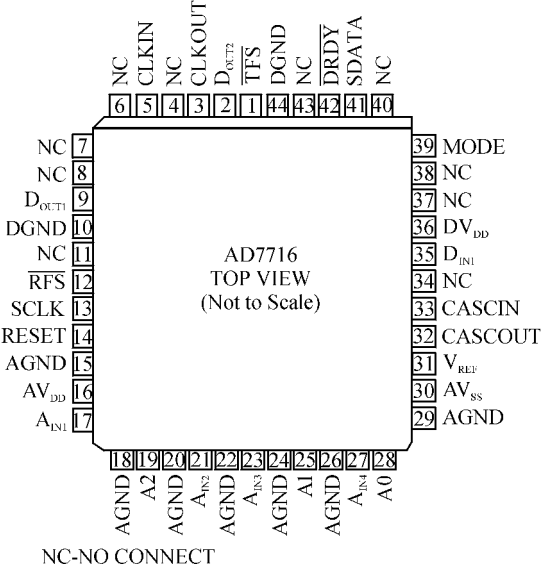


图 8-15 AD7716 外形引脚

场合,例如地震信号数据采集、生物医学信号采集等。

AD7716A/D 转换器具有以下主要特征。

- 分辨率: 22 位。
- 线性度误差: 0.006 %。
- A/D 采样率: 570 kHz。
- 最高 A/D 输出更新率: 2 232 Hz。
- A/D 转换延时: 10.8 ms(编程截止频率 36.5 Hz)。
- 模拟信号范围: $\pm 2.5\text{ V}$ 。
- 通道串扰: -85 dB 。
- 电源: $AV_{DD}=DV_{DD}=5\pm 5\%\text{ V}$ 。
 $AV_{SS}=-5\pm 5\%\text{ V}$ 。
- 功耗: 50 mW。
- 温度范围: $40\sim 85\text{ }^{\circ}\text{C}$ 。

AD7716 的封装有 44 脚 PLCC 和 44 脚 PQFP 两种,外部引脚如图 8-15 所示。其中主要引脚功能如下。

- RESET: 复位线,高电平有效。
- CLKIN: 时钟输入线。
- CLKOUT: 时钟输出线。
- MODE: 7716 工作模式选择线,高电平为主模式,低电平为从模式。
- RFS: 读 A/D 转换结果同步线,低电平有效。
- SDATA: 串行数据线。
- SCLK: 串行时钟线。
- DRDY: 数据输出有效,低电平有效。
- TFS: 芯片编程同步线。
- D_{IN1}: 数字信号输入,包含在输出结果中。
- V_{REF}: 参考电压, +2.5 V。
- A_{IN1} ~ A_{IN4}: 模拟输入端, $\pm 2.5\text{ V}$ 。

AD780 是低噪声精密电源,用作 AD7716 的 A/D 转换参考电压。AD780 的输出电压可以调节,只要调节接在 AD780 的电压输出端 V_{OUT}所连接到地的电位器的状态,并把电位器抽头的输出信号送回 AD780 的 TRIM 端即可。在地震数据采集系统中,由于输入到 AD7716 的模拟信号范围为 $-2.5\sim +2.5\text{ V}$,故 AD780 输出的参考基准电压应为 2.5 V。

AT89C2051 是地震数据采集系统的核心。数据的采集过程是在 AT89C2051 的控制下进行的。同时,AT89C2051 对 AD7716 转换结果的读取是串行进行的。数据以 AD7716 的 SDATA 端串行输出,然后,串行输入到 AT89C2051 的 P1.2 端,由 AT89C2051 接收。AT89C2051 还通过串行口 P3.0, P3.1 以及 RS-232 接口芯片 MAX232 和上位机通信。20 位 A/D 转换器 AD7716 的所有工作过程是在 AT89C2051 的控制下实现 A/D 转换的。

MAX232 是 RS-232 接口芯片,主要工作是在标准的 TTL 逻辑电平信号和 RS-232 电平信号之间实现逻辑信号的变换。TTL 逻辑电平工作在 $0\sim 5\text{ V}$,而 RS-232 逻辑电平工作在 $-15\sim +15\text{ V}$ 。

8.3.2 数据采集系统的逻辑电路

数据采集系统主要的逻辑电路有 A/D 转换电路和单片机控制电路,分别如图 8-16、图 8-17 所示。

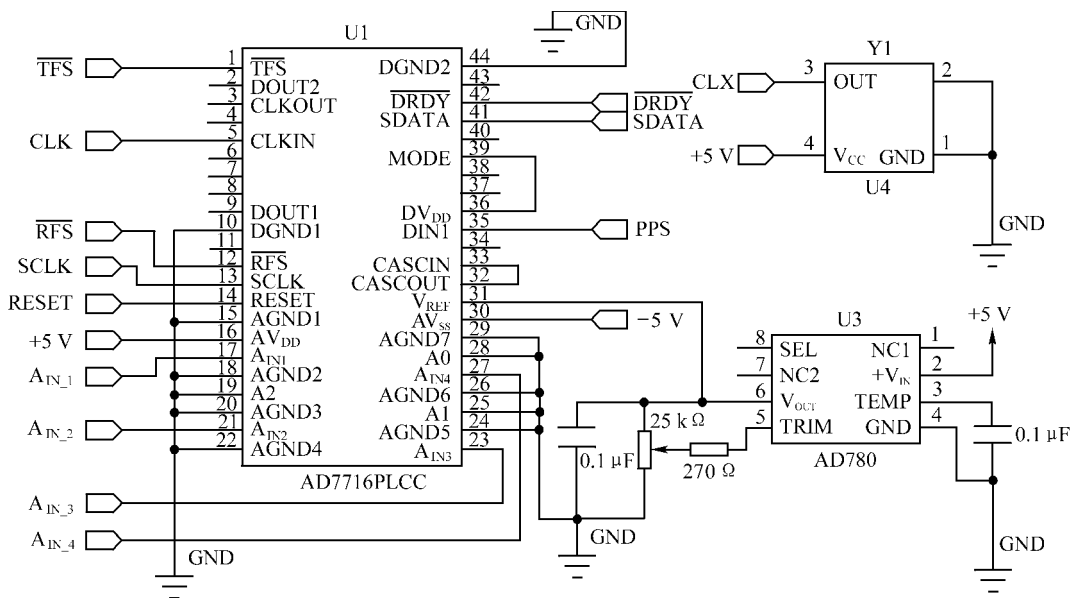


图 8-16 A/D 转换电路

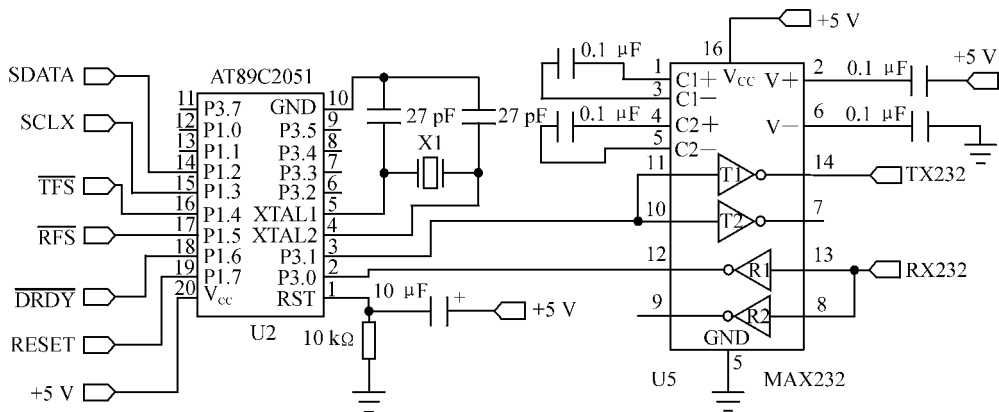


图 8-17 单片机控制电路

图 8-16 和图 8-17 为 22 位地震数据采集系统部分电路原理图。图 8-16 中 AD7716 芯片的 MODE 置低电平,工作在从模式。AD7716 的 CLKIN 接 8 MHz 外部晶体振荡器 Y1 的输出,为 AD7716 提供可靠的时基信号。参考电压源采用 AD780,为 AD7716 的低噪声基准源。输入信号 $A_{IN1} \sim A_{IN4}$ 接检测信号放大器的输出,并要求放大器输出电压范围限制在 $-2.5 \sim +2.5$ V 之间。AT89C2051 通过 P1 口的 P1.2、P1.3、P1.4、P1.5、P1.6、P1.7 与 AD7716 接口,分别接到 AD7716 的 SCLX、TFS、RFS、DRDY、RESET 等。电路中没有用 P1 口的 P1.0 和 P1.1,是因为 AT89C2051 单片机 P1.0 和 P1.1 兼作比较器输入,芯

片内部无上拉电阻,带负载能力比其他脚弱。系统采用串口 P3.0(RXD),P3.1(TXD)把信号输到上位机,串行信号通过 MAX232 芯片作电平转换后,成为 RS-232 口,通过该口与上位机进行通信,接收上位机的指令,并将 A/D 转换结果传输给上位机。在 AD7716 转换完成后,AD7716 的 DRDY 脚变低,并要求单片机提供 SCLK 时钟信号。单片机通过对 P1.4(TFS 脚)、P1.2(SDATA 脚)和 P1.3(SCLK 脚)编程输出对 AD7716 寄存器进行设置。通过对 RFS,SDATA,SCLK,DRDY 的编程从而读取 AD7716 的输出结果。

从逻辑电路可以看出:单片机工作时是用其内部振荡电路加外接晶体振荡器产生时钟信号的,并且在这个时钟信号的同步下执行程序及系统的控制工作。而 22 位 A/D 转换器 AD7716 的 A/D 转换过程是依赖外部振荡电路 Y1 提供的时钟信号同步的。至于对哪一路输入的模拟信号进行转换,何时读取 A/D 转换结果,则是在 AT89C2051 的控制下执行的。

8.3.3 数据采集系统的工作程序

数据采集系统主程序主要包括系统初始化子程序,AD7716 设置子程序,读取 AD7716A/D 转换结果子程序,串口中断通信子程序等。在程序设计中为提高效率,在读取 A/D 转换结果时采用查寻的方法,串口数据输出时采用中断的方法,使读取数据和串口输出同步进行,其中系统初始化子程序完成单片机特殊功能寄存器的初始化,并按照串口通信波特率设置定时器中 T1 的参数等。AD7716 设置子程序完成对 AD7716 参数的设置。串口中断子程序一方面完成接收并解释上位机的命令,包括 AD7716 截止频率的设定参数,通道数等;另一方面完成 A/D 转换结果的输出。下面给出具有代表性的子程序,也就是读取 AD7716 转换结果子程序,其余省略。

```
READ:MOV      B, #016      ; A/D 输出字节个数共 16 个
      MOV      R0, #30H     ; 存放数据地址指针
      SETB     DRDY
      MOV      C,DRDY      ; 读取 DRDY 状态
      JNC      OK          ; 为低电平, A/D 转换完成
      AJMP     READ        ; 为高电平, 等待
OK:   CLR      RFS         ; 将 RFS 变低
LOOP: SETB     SCLK        ; 输出 SCLK 上升沿
      MOV      C,SDATA     ; 读取 SDATA 数据
      MOV      02FH,C
      CLR      SCLK        ; 输出 SCLK 下降沿
      SETB     SCLK
      MOV      C,SDATA
      MOV      02EH,C
      CLR      SCLK
      SETB     SCLK
      MOV      C,SDATA
      MOV      02DH,C
      CLR      SCLK
      SETB     SCLK
      MOV      C,SDATA
```

```
MOV      02CH,C
CLR      SCLK
SETB     SCLK
MOV      C,SDATA
MOV      02BH,C
CLR      SCLK
SETB     SCLK
MOV      C,SDATA
MOV      02AH,C
CLR      SCLK
SETB     SCLK
MOV      C,SDATA
MOV      029H,C
CLR      SCLK
SETB     SCLK
MOV      C,SDATA
MOV      028H,C
CLR      SCLK
MOV      @R0,025H
INC      R0
DJNZ     B,LOOP
SETB     RFS
RET
```

8.4 刀闸自动操作控制

在中小型变电所中,工作人员要视供电的状态,对供电的开关机构——刀闸执行操作。刀闸通常有母刀闸、线刀闸和油断路器等 3 种。

在供电系统中,每条线路都会带大小不同的负载。供电线路的开关在带负载时断路或接通,都会产生电弧,这样,就容易造成火灾或烧伤等事故。为了防止电弧的产生,一般都有严格的操作规程。这些规程的规定简单说明如下。

- (1) 供电线路退出运行时的操作
断开油断路器→断开线刀闸→断开母刀闸
- (2) 供电线路投入运行时的操作
合上母刀闸→合上线刀闸→合上油断路器

从上面的规程可以看出:按顺序操作的目的在于一方面防止母刀闸在严重带负载时断开而产生电弧,危及安全;另一方面,防止母刀闸在带负载的情况下合闸而产生电弧,导致事故。

国内大多数中小型终端变电所,由于受资金和工作人员素质等条件的制约,其刀闸操作主要还是依靠人工完成。为了防止人工操作中带负荷操作刀闸,电力系统制订了刀闸操作规程,要求工作人员严格按规程执行。由于操作比较繁琐,工作人员误操作刀闸造成事故时有发生,严重时将烧毁设备,造成大面积停电,给国民经济造成巨大的损失,有时甚至危及到操作人员

的生命安全。为了减少此类事故的发生,一些有条件的中小型变电所装配了机械或电磁互锁,虽然减少了误操作,但同时增加了执行时间,使电力系统带故障运行时间加长,停电造成的危害和损失加大。用单片机构成的刀闸自动操作系统可以解决上述问题,以单片机 AT89C51 为核心的一种具有语音提示功能的 4 个主要用户的刀闸自动操作控制系统就是一个典型的系统。

8.4.1 刀闸自动操作控制原理

刀闸自动操作控制就是要依据供电线路的状态,在需要把供电线路的供电切断时,按先后顺序自动对油断路器、线刀闸和母刀闸执行断路操作;在需要对供电线路合闸供电时,按先后的排序对母刀闸、线刀闸和油断路器执行自动的合闸操作。

具有 4 个主要用户的小型终端变电所刀闸自动操作控制器工作原理如图 8-18 所示。当变电所监控系统检测到某用户线路例如 1 号用户有故障时,产生一个高电平的撤出运行信号给单片机,单片机查询到此信号后,立即向 1 号用户刀闸操作机构发出高电平信号,让其执行撤出运行操作。依次断开油断路器、线刀闸和母刀闸,同时单片机应做 3 件事。

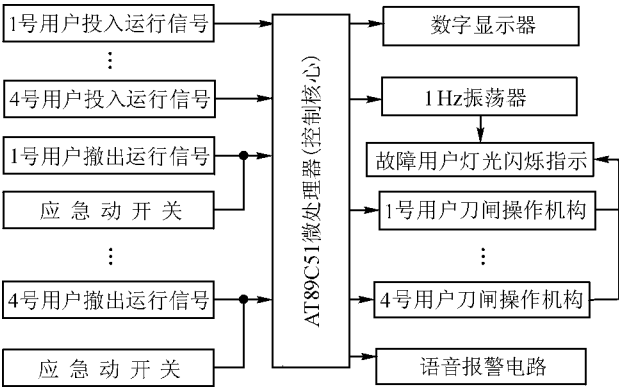


图 8-18 刀闸自动操作控制系统框图

1. 进行用户故障显示

这时,用数字显示器显示产生故障的用户号。当 1 号用户产生故障时,则显示器显示“1”,其余类同。

2. 进行用户故障语音报警

对产生故障的用户用语音报警 3 次。例如当故障是由 1 号用户出现的,则语音报警用语音发出“1 号用户故障”3 次,其余类同。

3. 进行光信号闪烁报警

为了引起操作人员的注意,还在产生故障时,启动 1 Hz 振荡器,产生 1 s 闪烁 1 次的光报警信号。

在故障处理时,当刀闸机构依次完成撤出运行的刀闸操作后,1 号用户的指示灯将开始闪烁,表示故障正在处理之中。当变电所监控系统出现了问题,没有发出撤去运行信号时,可采用应急手动开关产生高电平信号给微处理器,以保证整个系统的可靠工作。当用户线路故障已排除,该线路要投入运行时,按一下 1 号用户投入运行开关按钮,产生一个投入运行高电平信号给单片机。当单片机查询到此信号后,首先发出信号使 1 号用户刀闸操作机构执行投入运行的刀闸操作,依次闭合上母刀闸、线刀闸和油断路器,然后撤消灯光报警,同时单片机使语

音报警电路连续报出“1 号线路投入运行”语言 3 次,整个操作结束。

刀闸操作机构,将来自单片机的信号经光电隔离器、功率放大、驱动中间继电器,最后使各个执行单元动作,按规定次序执行刀闸的操作。

8.4.2 控制系统逻辑电路

图 8-18 所示的刀闸自动操作控制系统,适用于仅有 4 个主要用户的小型终端实际变电所。因用户数量少,I/O 端口线不必扩展,所以单片机可用内部含 ROM 和 RAM、价格低廉、能电擦写的 Flash 型 AT89C51 系列单片机作为其控制核心,这样可使系统更加简单可靠。所以,刀闸自动操作控制系统采用 AT89C51 为其控制核心部件,所有的系统逻辑电路均在 AT89C51 的控制下协调工作。在控制系统中,AT89C51 的控制工作有两大部分,其中第一部分检测用户运行状态,包括用户投入运行的信号和用户退出运行的信号,另外还有应急开关状态;第二部分输出控制,包括显示控制、光闪烁报警控制、语音报警控制、用户刀闸操作机构控制等。

1. 显示控制电路

显示控制电路和一般 7 段 LED 数字显示电路相同。对显示电路输入不同的显示码,则可以显示不同的数字。在刀闸自动操作控制系统中,只需显示 1~4 共 4 种数字,它们分别对应 1~4 号用户。

2. 灯光闪烁报警电路

灯光闪烁报警电路由两部分组成,一部分是闪烁控制信号振荡电路,另一部分是灯光选择电路。

灯光闪烁报警电路结构如图 8-19 所示。

灯光选择电路由指示灯和继电器触点组成。其中 J13-2 是 1 号用户退出运行继电器触点,J23-2 是 2 号用户退出运行继电器触点,J33-2 是 3 号用户退出运行继电器触点,J43-2 是 4 号用户退出运行继电器触点。指示灯 L1~L4 分别对应于 1~4 号用户。当 k 号($k=1\sim4$)用户有故障时,则 k 号用户线路退出运行,这时继电器触头 Jk3-2 就会闭合,使 k 号指示灯接通电源 +12 V,做好闪烁显示报警的准备。

闪烁控制信号振荡电路由 555 振荡电路和晶体管开关 BG10 组成。当晶体管 BG10 的基极所输入的信号为高电平时,BG10 导通,从而使其集电极所连接的指示灯发光;当基极输入的信号为低电平时,BG10 截止,其集电极所连接的指示灯不能通电,故不会发光。555 振荡电路由 LM555 集成电路和相连的电阻、电容组成。它是一个振荡频率为 1 Hz 的可控式振荡器,控制端为引脚 4。当 LM555 的引脚 4 为高电平时,振荡器产生振荡;当引脚 4 为低电平时,则振荡器停振。在刀闸自动操作控制系统中,LM555 的引脚 4 由单片机 AT89C51 输出端口的电平所控制。

当 AT89C51 送到 LM555 的引脚 4 的信号是高电平时,555 振荡器马上以 1 Hz 的频率振荡,振荡信号从 LM555 的引脚 3 输出,并加到晶体管 BG10 的基极,则晶体管按 1 Hz 的频率进行开或关。这时,只要有一个继电器触点 Jk3-2($k=1\sim4$)闭合,则对应的指示灯就会以

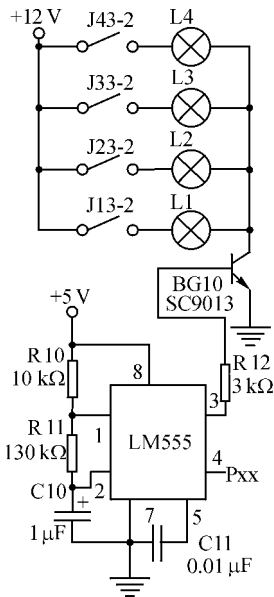


图 8-19 光闪烁电路

便。基本存储单元入口地址范围是 00H~9FH。为了使该报警电路具有通用性,特将录制语音分成 16 段。因每个基本存储单元存储语言时间为 0.125 s,而在 16 段语音中,每段语音占用了 10 个存储单元,故每段语音的录放时间为 1.25 s。16 段语音及其对应存储区域入口地址如表 8-3 所列。

表 8-3 语音存储单元分段表

内 容	0	1	2	3	4	5	6	7	8	9	十	号	用 户	故 障	投 入	运 行
人口地址	00H	09H	14H	1EH	28H	32H	3CH	46H	50H	5AH	64H	6EH	78H	82H	8CH	96H

按表 8-3 对应入口地址,用单片机控制或手动控制对 16 段语音进行录制。录制时每段语音录制时间不应超过 1.25 s。通过单片机控制检查每段语音录制效果,也可通过闭合 S1, S2 开关,按 AN1 键去检查,每按一次 AN1 键将输出一段语音信息。如果语音录制效果不理想,可反复录制,直到满意为止。S3, S4 开关分别用于手动录放音控制。

4. 用户刀闸操作机构控制电路

用户刀闸操作机构是由单片机控制的,以使这些机构按规定的顺序断开或接通相应的闸开关。在控制电路中,为了把单片机系统的逻辑电路和电力线的高压分隔开来,故在控制电路中采用光电隔离的光电耦合器 MCT275,并采用中功率晶体管 S9013 驱动中间继电器,再由继电器驱动刀闸操作机构。这样,有两个优点:第一,把 5 V 的逻辑电路和电力线路隔离;第二,光电耦合器可以实现抗干扰,保护单片机系统,防止来自电力线路的干扰信号影响单片机系统的正常工作。

8.4.3 控制系统的工作程序框图

刀闸自动操作控制系统可以对 4 个用户线路进行故障处理和控制。由于用户较少,所以,采用查询方式进行控制就完全可以满足系统实时性的要求。系统的主程序框图如图 8-21 所示。从图中可以看出:程序是按用户号进行顺序查询的。从 1 号用户到 4 号用户,查询的时间极短,对于任何用户而言,实时性是完全能满足的。

对于 4 个用户线路来说,它们的控制过程及动作是一样的,只是控制的对象不同。因此,每个用户的工作子程序是类同的。了解了其中一个用户的工作子程序,就相当于了解了其他用户的工作子程序。下面就以 1 号用户的工作子程序说明用户刀闸自动操作控制的过程,1 号用户工作子程序框图如图 8-22 所示。

从图 8-22 中可以看出:当 1 号用户线路产生故障时,单片机就转入 1 号用户工作子程序进行处理。这时,首先断开油断路器,显示故障的用户号,接着进行语音报警。然后,断开线刀闸,延时 4 s 之后,断开母刀闸,紧接着进行灯光闪烁报警。在完成上述故障响应处理之后,程序进入进行检修情况判别。一旦检修线路的工作完成,线路工作正常,则控制母刀闸闭合;延时 4 s 之后,控制线刀闸闭合;再延时 4 s,控制油断路器闭合。最后用语音报告 1 号用户线路投入运行,并返回主程序。

显然,其余 2~4 号用户的工作子程序的工作过程和内容是相同的,只是具体对象如油断路器、线刀闸、母刀闸不同,报警指示灯、报警的语音也有区别而已。因为,无论是 1 号用户,还是 2~4 号用户,它们都必须按线路投入运行和退出运行的规程执行相关的操作。

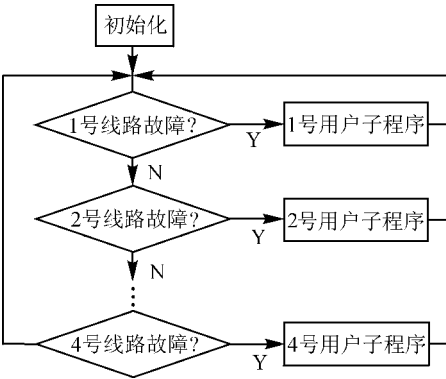


图 8-21 主程序框图

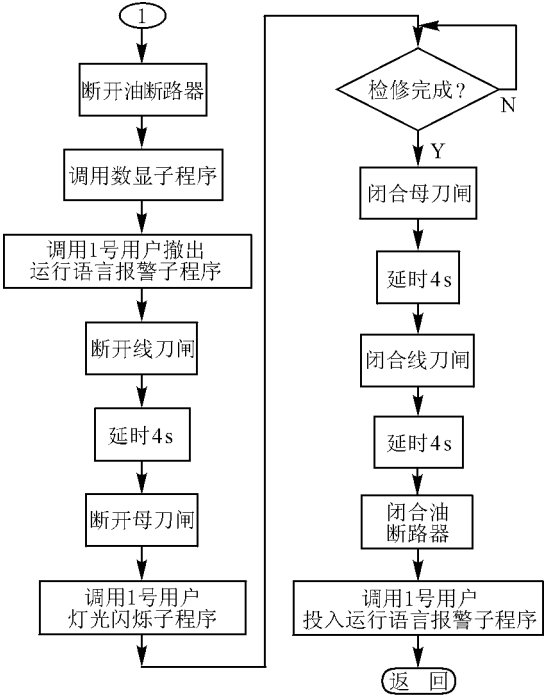


图 8-22 1号用户工作子程序框图

8.5 滚动式广告窗控制

广告是市场发展的一种信息传播形式。各种产品信息的发布,很重要的一个途径就是通过广告。因此,广告已成为一种很重要的经营手段和方法。随着商品市场的发展,广告从静态走向动态,从电视台、广播电台播放的广告到街头的霓虹灯、广告画、广告牌、广告窗等五花八门,广告也进入了自动控制的时代。滚动式广告窗就是由单片机控制的一种静态广告画面转换的自动控制系统。滚动式广告窗可以把数幅广告画面以滚动方式进行转换,使得广告窗在某一段时间内显示的是一种特定的广告,而在另一段时间内显示的是另一种特定的广告,数种广告循环滚动显示。因此,一种广告工具就可以实现多种广告的效果。这种广告工具有点类似“动态”的特点,不但容易引起人们的注意,而且也不会使人们产生乏味的感觉。在这一节中,介绍用 AT89C2051 控制的滚动式广告窗的控制系统。

8.5.1 滚动式广告窗控制原理

滚动式广告窗控制系统由单片机 AT89C2051、步进电机控制电路、语音电路、超声波发射电路、超声波接收电路组成。系统的结构框图如图 8-23 所示。

1. 步进电机控制电路

步进电机控制电路用于驱动步进电机的步进转动,而步进电机带动广告转动,从而实现了广告画面的转换功能。

由于步进电机的转动是按步距进行的,而其步距又很小,所以,控制广告画面在窗口内定

位显示时,有很高的精度,这样才能保证广告窗滚动显示不同广告画面的质量。

2. 语音电路

语音电路用于录放与对应广告画面相关的语音信息,可在广告窗显示对应的广告画面时进行语音播放。这样,会加深人们对广告的注意和了解,同时,加强广告的效果。语音和广告画面在本质上是互补的、相辅相成的。画面是目视的广告,语音是耳闻的广告,有语音的广告画面有双管齐下的广告力度。

3. 超声波发射电路

超声波发射电路是近距离检测电路的一部分。它的作用是发射用于近距离检测的源信号。由于超声波有较好的定向性和反射性,超声波发射电路由广告窗向正面方向发射,当广告窗正面有人停留时,就会反射超声波送回广告窗。

4. 超声波接收电路

超声波接收电路是近距离检测电路的另一部分。其作用是用于接收广告窗前停留的人、物所反射回来的超声波。超声波接收电路是和超声波发射电路一同使用的,它们共同组成了近距离检测电路,用于检测广告窗前停留的人、物。超声波发射电路在广告窗的正面向外发射超声波,只要有人、物在广告窗前停留,就会反射超声波,被超声波接收电路接收了之后就送入单片机,从而可以判别出广告窗前是否有人停留。

5. 单片机 AT89C2051

AT89C2051 是控制系统的核心。它的工作包括控制超声波信号的发射和超声波反射信号的接收;另外根据超声波反射的情况决定语音的播放;再有就是控制步进电机带动广告窗按一定时间周期滚动输出不同的广告画面。

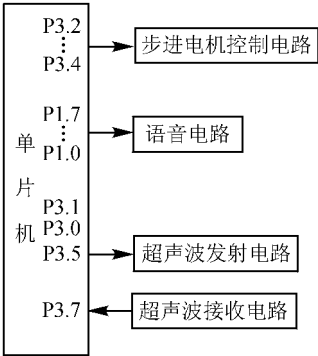


图 8-23 滚动式广告窗控制系统框图

8.5.2 超声波的发射和接收原理

近距离探测电路由发射电路和接收电路两部分组成,如图 8-24 所示。所采用的传感器是一对经过优化的压电陶瓷片,其中一个具有 40 kHz 发射功能,另一个则具有 40 kHz 接收功能。根据雷达探测原理,进入该探测电路探测范围的人或物体均可将压电陶瓷片 HD1 发射的 40 kHz 超声波反射到接收端的压电陶瓷片 HD2 上。由单片机 P3.5 口控制的 NE555 输出信号驱动功率管 BG1,再经过升压变压器 T1 升压后驱动压电陶瓷片 HD1 向广告窗前发射出 40 kHz 超声波信号。接收电路主要由 LM324 运放组成,首先两级单电源供电的同相放大器将接收电路中的压电陶瓷片所接收到的微弱信号进行放大,再由下一级所构成的比较器对其输出的幅度进行比较,超过设定门限的则被认为是有用信号,再将其送入单片机 P3.7 端口进行进一步判定。当接收到一定数量的 40 kHz 回波脉冲信号后,即视为在广告窗前有对该广告感兴趣的人,则单片机马上控制 ISD1210 语音电路播放出与广告对应的语音信息。直到人走后,才停止播放语音信息并恢复原来的滚动。

1. 超声波发射原理

超声波发射电路由 555 可控振荡电路和超声波输出电路组成。

555 可控振荡电路的核心是 555 时基集成电路。

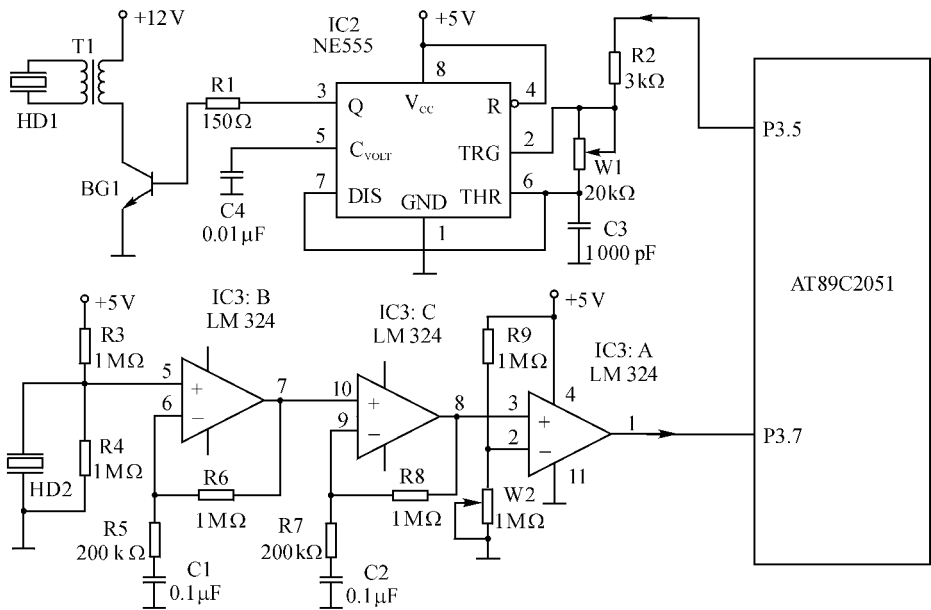


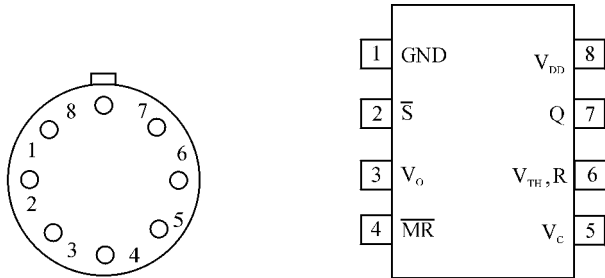
图 8-24 超声波发射和接收电路

555 时基集成电路之所以用“555”来标志其型号,是因为芯片内的基准电压由 3 个误差极小的 5 k Ω 电阻组成,有很高的分压精度。

555 时基电路之所以得到这样广泛的应用,在于它具有如下几个特点。

- (1) 555 在电路结构上是由模拟电路和数字电路组合而成。它将模拟功能与逻辑功能兼容为一体,能够产生精确的时间延迟和振荡。它拓宽了模拟集成电路的应用范围。
- (2) 该电路采用单电源。双极型 555 的电压范围为 4.5~15 V;而 CMOS 型的电源适应范围更宽,为 2~18 V。这样,它就可以和模拟运算放大器、TTL 或 CMOS 数字电路共用一个电源。
- (3) 555 可独立构成一个定时电路,且定时精度高。
- (4) 555 的最大输出电流达 200 mA,带负载能力强。可直接驱动小电机、喇叭、继电器等负载。

555 时基集成电路的封装如图 8-25 所示,一般采用双列直插式封装,也有少数采用 TO-99 型封装。



(a) 8 脚 TO-99 型

(b) 8 脚双列直插型

1—地;2—触发;3—输出;4—复位;5—控制电压;6—阈值电压;7—放电;8—电源+V_{DD}

图 8-25 555 时基电路的封装

555 时基集成电路的等效电路框图如图 8-26 所示。

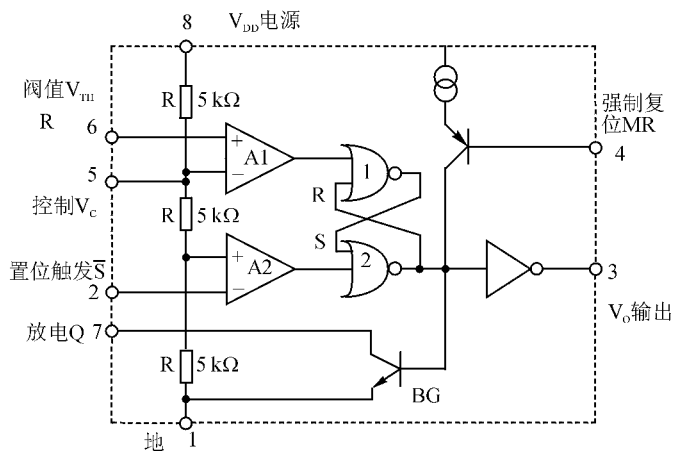


图 8-26 555 等效功能框图

从图 8-26 可见,3 个 $5\text{ k}\Omega$ 电阻组成的分压器,使内部的 2 个比较器构成一个电平触发器,上触发电平为 $2/3\text{ V}_{\text{DD}}$,下触发电平为 $1/3\text{ V}_{\text{DD}}$ 。在 5 脚控制端外接一个参考电源 V_{C} ,可以改变上、下触发电平值。比较器 A1 的输出同或非门 1 的输入端相接,比较器 A2 的输出端接到或非门 2 的输入端。由于由两个或非门组成的 RS 触发器必须用正极性信号触发,因此,加到比较器 A1 同相端 6 脚的触发信号,只有当电位高于反相端 5 脚的电位时,R-S 触发器才翻转;而加到比较器 A2 反相端 2 脚的触发信号,只有当电位低于 A2 同相端的电位 $1/3\text{ V}_{\text{DD}}$ 时,R-S 触发器才翻转。

在 555 电路中,引脚 2 的触发信号比引脚 6 的阈值触发信号要优先,就是说,只要引脚 2 有低电平 $\leq 1/3\text{ V}_{\text{DD}}$ 输入,则不论引脚 6 状态如何,都会把片内 RS 触发器置为“1”,这时或非门 2 输出为“0”,故 V_{O} 输出“1”。

555 电路的功能真值表如表 8-4 所列。

表 8-4 555 电路的功能真值表

引 脚	2($\bar{\text{S}}$)	6(R)	4($\overline{\text{MR}}$)	7(Q)	3(V_{O})
状 态	$\leq 1/3\text{ V}_{\text{DD}}$	—	$> 1.4\text{ V}$	截止	1
	$> 1/3\text{ V}_{\text{DD}}$	$\geq 2/3\text{ V}_{\text{DD}}$	$> 1.4\text{ V}$	导通	0
	$> 1/3\text{ V}_{\text{DD}}$	$< 2/3\text{ V}_{\text{DD}}$	$> 1.4\text{ V}$	保持	保持
	—	—	$< 0.3\text{ V}$	导通	0

根据 555 电路的真值表,在图 8-24 超声波发射电路中,只单片机输入输出端口 P3.5 输出为低电平,即 $\text{P3.5}=0$,555 的引脚 2 输入的电平低于 $1/3\text{ V}_{\text{DD}}$,则 555 电路只处于 $V_{\text{O}}=1$ 状态,即其引脚 3 输出“1”。当 $\text{P3.5}=1$ 时,则有 555 电路的引脚 2 输入电平高于 $1/3\text{ V}_{\text{DD}}$,而同时 P3.5 输出的高电平经 R2 和 W1 对 C3 充电,当 C3 充电达到 $2/3\text{ V}_{\text{DD}}$ 时,则 555 的引脚 6 会使其内部 RS 触发器翻转,从而使引脚 3 输出“0”。同时,由于引脚 7 导通接地,则 C3 的电荷通过内部晶体管 BG1 放电。当 C3 的电荷放到一定程度,C3 的电压下降到使 R2、W1 分压比

小于 $1/3 V_{DD}$, 输入到引脚 2 中, 则 555 的引脚 3 又输出“1”, 同时, RS 触发器输出的“0”信号使 BG 截止, C3 停止放电。P3.5 送来的高电平又经 R2, W1 对 C3 充电……以后过程重复进行。显然, 这是一个振荡过程。振荡频率由 R2, W1 和 C3 的值确定。

由此看出:

第一, 单片机 P3.5=0 时, 555 电路输出静态电平, $V_O=1$ 。

第二, 单片机 P3.5=1 时, 555 电路输出振荡信号, 频率由 R2, W1, C3 的参数决定。只要恰当选择这些参数, 可以使 555 电路输出 40 kHz 的超声波信号。

555 电路产生超声波信号通过功放电路的晶体管 BG1 放大之后, 经由变压器 T1 耦合给压电陶瓷片 HD1 转换成声波发射出去。

2. 超声波接收原理

超声波接收原理十分简单, 压电陶瓷片 HD2 接收反射的超声波之后, 会产生一定的微弱电压, 这个电压由 2 级同相的放大器进行放大, 经过 2 级放大之后产生足够大的信号, 再和第 3 级的给定信号进行比较; 当超过给定的比较阈值, 则认为反射的超声波是真信号, 否则以为是伪信号。第 3 级比较器产生的 40 kHz 信号送到单片机的 P3.7 端。单片机对 P3.7 端输入的信号进行计数, 当其值足够大时, 可以认为有人停留在广告窗前, 否则, 只是表示有人路过广告窗前。

8.5.3 语音播放原理

当单片机检测到有超声波反射时, 如果反射的时间有一定长度, 则认为有人停在广告窗前面, 则单片机控制语音芯片, 播放和广告画面对应的语言信息。

语音芯片采用 ISD1200/1400 语音处理集成电路。这种语音芯片内部有基准时钟源, 以单一电源 +5 V 供电。只需少量的外围元件就可以构成一个录放音电路。在录音时只需要一个按键控制, 放音时也只需要一个按键控制, 所以控制极其方便。另外, 采用单片机的电平也可完成录放音按键的控制功能。在芯片内存储的语音信息可保存 100 年, 芯片的录放次数可达 10 万次。

1. ISD1200/1400 的内部逻辑结构

ISD1200/1400 系列语音集成电路有 4 种不同的型号, 如表 8-5 所列。

表 8-5 ISD1200/1400 系列语音芯片

器件型号	录放音时间/s	最大输入采样频率/kHz	通频带上限/kHz	分辨率/ms
ISD1210	10	6.4	2.7	125
ISD1212	12	5.3	2.3	150
ISD1416	16	8.0	3.4	100
ISD1420	20	6.4	2.7	125

ISD1200/1400 系列语音集成电路的内部结构由内部时钟电路、自动增益控制电路、前置放大电路、滤波器、差动功率放大电路、电源电路、存储器 EEPROM、地址译码电路、存储控制电路等组成, 结构框图如图 8-27 所示。

2. ISD1200/1400 的封装引脚及含义

ISD1200/1400 系列语音芯片在表 8-5 中列出 4 种, 每种型号的最后 2 位数字表示语音

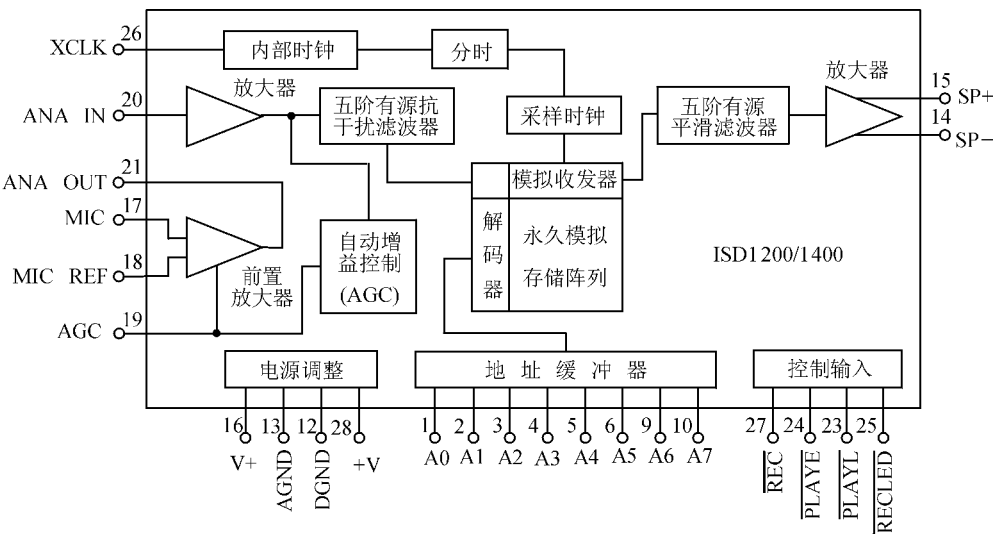


图 8-27 ISD1200/1400 芯片结构框图

录放时间的长度。从表中可知,录放音时间最短的为 10 s,最长为 20 s。

ISD1200/1400 系列语音芯片的封装引脚如图 8-28 所示。它是有 28 条引脚的双列直插式芯片。各条引脚的功能含义说明如下。

A0~A7: 引脚 1~6,9,10,地址输入端或控制命令输入端。A7,A6 同时为高电平时,A4~A0 为控制命令;否则,A7~A0 为地址。

DGND: 引脚 12,数字信号地线。

AGND: 引脚 13,模拟信号地线。

SP-,SP+: 扬声器连接端,输出音频信号。

V+: 模拟信号电源,+5 V。

MIC: 引脚 17,话筒输入端。

MIC REF: 引脚 18,话筒参考输入端。

AGC: 引脚 19,自动增益控制端。

ANA IN: 引脚 20,模拟信号输入端。

ANA OUT: 引脚 21,模拟信号输出端。

PLAYL: 引脚 23,放音控制电平触发端。当该端为低电平时,芯片进入放音周期;当该端为高电平时,停止放音。

PLAYE: 引脚 24,放音控制脉冲触发端。该端输入由高电平向低电平跳变的下降沿时,芯片进入放音周期。

RECLE: 引脚 25,录音显示端。该端接发光二极管,在录音时作录音指示灯。

XCLK: 引脚 26,时钟控制端。

REC: 引脚 27,录音控制端。该端为低电平时,芯片进入录音状态,录音期间该端必须保持低电平。REC信号的优先级高于PLAYL和PLAYE两种放音信号。

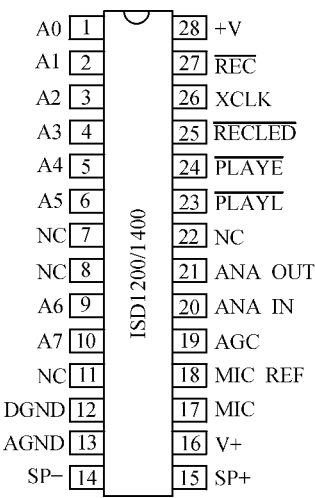


图 8-28 封装引脚

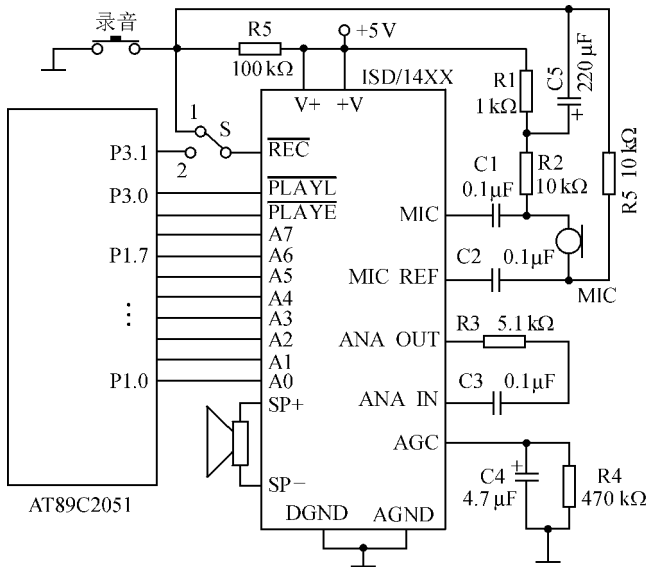


图 8-30 随机地址录放音电路

当 A7,A6 都取值为高电平时,A4,A3,A1,A0 各位用作芯片操作方式控制。

A0 用作信息检索控制。例如在放音时,如果不清楚每段语音的实际地址,可用 A0 控制检索过程,A0 每输入一个负脉冲,地址计数器就可跳过一段,指向下一段的起址。

A1 用作删除信息段结束标志的控制位,例如在录音期间可用 A1 删除每段信息中间的结束标志的插入,使全部录制内容只保留最后一个结束标志符,以便将来可以连续重放。

A3 可用来进行片内信息循环重放的控制,既可以循环重放一段,又可以循环重放全部。

A4 用于连续寻址控制,一般情况下芯片放完一段遇上结束标志就要停止,计数器会自动复位。用 A4=0 就可制止这种复位过程,实现片内信息的连续播放。A1 若与 A0 联用,控制效果更好。

当 A7,A6 中有一个为低电平时,A7~A0 用作 EEPROM 的地址编码。编址范围为 00H~9FH,共 160 个地址码,对应着 160 段语音存储单元。若芯片的录放时间为 t_m ,则每段录放时间 $t_s = t_m/160$,称为分辨率。根据它可以估算出从录放开始算起的每段语音起始时间。

录放地址信息都是在有效录放控制命令下降沿时写入片内地址锁存器的,因此可以有选择地进行语音的存储录放。因此,在录音时,在不同地址内,录入不同广告画面对应的语音;在显示一幅广告画面时,在单片机控制下同步放出对应语音。

8.5.4 步进电机控制原理

在滚动式广告窗中,是由步进电机带动广告画面滚动显示的。在系统中,单片机发控制信号给步进电机驱动电路,再由步进电机驱动电路来驱动步进电机。步进电机控制原理如图 8-31 所示。

单片机的 P3.2 用于发送步进脉冲信号,故和步进电机的步进端相接;P3.3 用于控制步进电机的转动方向,所以和驱动电路的方向端相连;P3.4 用于控制步进电机脱机,故和脱机端相连。另外,单片机的电源和驱动电路的 4 端相连。实际上,步进电机驱动电路 SH-20806C 的

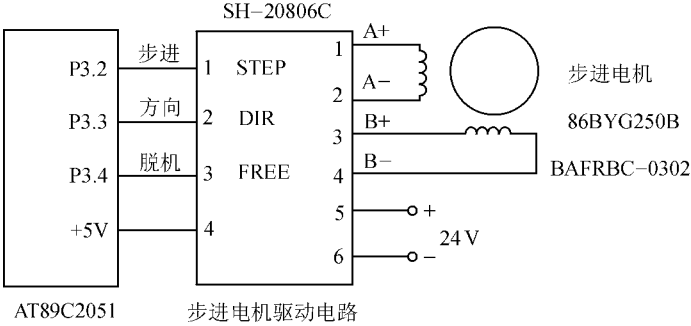


图 8-31 步进电机控制电路

输入端 1,2,3 都是和光电耦合器的输入发光二极管的负端相连的,从而有 3 路光电隔离的控制信号控制步进电机。而输入端 4 则和 3 路光电耦合器的输入发光二极管的正极相连,为发光二极管提供 +5 V 电源。

步进电机驱动电路 SH-20806C 是一个内部可以自动产生对步进电机控制信号的电路。它只要求外部向其提供 3 种控制信号: 步进脉冲信号、方向控制信号和脱机控制信号。

1. 步进脉冲信号 STEP

步进脉冲信号最高频率为 50 kHz。这个信号为低电平时,使驱动电路内部光电耦合器的发光二极管导通。驱动电路在步进脉冲信号的下降沿时响应工作。脉冲低电平的有效宽度不能小于 10 μ s。

2. 方向控制信号 DIR

这个信号用于控制步进电机的旋转方向。当这个信号为高电平时为一种转动方向,为低电平时为反方向。为了保证可靠控制方向,要求方向控制信号比步进脉冲信号至少提早 20 μ s 建立。步进电机的初始运行方向和电机接线有关,把电机的两相互换,则可改变电机的初始运行方向。

3. 脱机控制信号 FREE

这个信号用于控制步进电机电源的开关。当它为低电平时,驱动电路切断步进电机绕组电流,使电机处于自由状态,此时步进脉冲信号不能被响应。自由状态可显著降低驱动电路和步进电机的功耗和温升。当不用此功能时,这个信号可以悬空。

步进脉冲信号、方向信号以及脱机信号的时序波形图如图 8-32 所示。

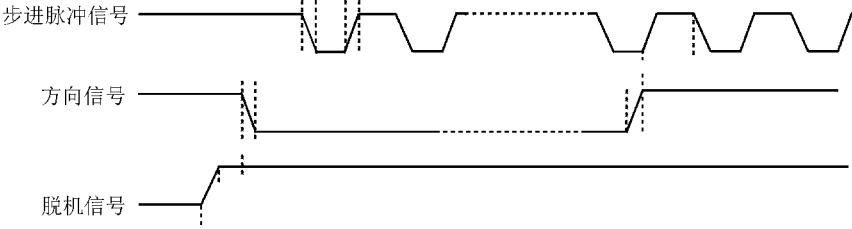


图 8-32 步进、方向、脱机信号波形时序

步进电机采用型号为 86BYG250B-BAFRBC-0302 的两相混合式步进电机。其相数为 2,步距角为 1.8°,静态相电流 3 A,相电阻 1.2 Ω ,相电感为 7.2 mH,转动增量为 3 000 gcm^2 。

8.5.5 控制软件框图

滚动式广告窗的控制软件包括 3 种主要功能：步进电机控制功能、超声波控制及接收功能、语音播放功能。另外还有广告画面停留时间的定时功能、初始化功能等。

滚动式广告窗控制软件框图如图 8-33 所示。

程序开始执行时,先对 AT89C2051 的有关部件包括定时器、I/O 端口进行初始化。接着,开始执行步进电机控制,使广告窗滚动显示一幅新的广告画,接着步进电机停止工作,广告画静态显示。程序开始定时,以确定广告画静态停留的时间。在定时过程中,单片机控制超声波发射电路发射超声波,接着程序判别定时时间是否结束。如果没有结束,则去判别是否接收到超声波的反射信号,实际上是检测 P3.7 的输入信号。当接收到超声波信号并有一定时间时,则播放对应的语音信号,也就是说一幅广告画应该和一段语音对应。如果没有收到超声波信号,则不播放语音。此后返回去对定时时间进行判别,如果没有结束定时,则重复上述相关过程;如果定时结束,则单片机控制发射超声波电路停止工作,然后返回程序开始部分,重新执行步进电机控制,滚动翻出一幅新的广告画,并静态显示……,此过程不断执行下去。所以,可以循环显示若干幅在广告窗内部安装好的广告画。

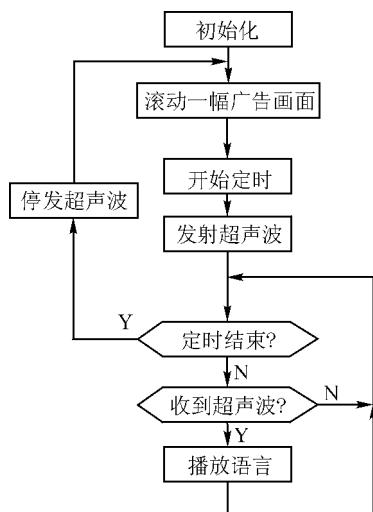


图 8-33 控制软件框图

第九章 仪器仪表领域的应用

单片机有体积小,内部还含有多种 I/O 接口的特点,因此很适合用于各种体积小的嵌入式系统中。仪器和仪表有体积小、功能各异、用途多、形式多样的特点,特别适合用单片机进行控制。

在仪器仪表中,单片机可以进行各种智能化处理。这些功能包括数据的检测、采集;数据特征的提取;对象的控制,例如一般的 PID 控制、模糊控制、自适应控制等;特种功能的输出,包括语音控制输出、显示控制输出、PWM 信号的输出;自身功能的自校正,如温度补偿、零点自校正、非线性的追踪模拟等。单片机控制的智能化仪表基本上有如下几个方面。

- ① 对象的自适应控制。
- ② 自身误差的自校正。
- ③ 系统故障的自修复。

要完全实现上述智能化功能还需要技术的不断进步和更深入的科学研究。目前,用单片机控制的仪器仪表已经部分地实现了某些智能化功能。在这一节中,介绍 89 系列单片机在仪器仪表中的有关应用。

9.1 智能煤气表

现在城市管道煤气已基本普及,随着管道煤气用户的增加,作为煤气计费依据的煤气表具数量与日俱增,有的城市达几十万甚至上百万用户。对于如此庞大的煤气表用户,抄表、收费工作将是一项十分繁琐的工作。由于用户各种客观情况的存在,例如生活水平、居住环境、生活习惯等不一样,给抄表、收费工作带来了很大困难。另外由于现阶段煤气用气方式是用户先用气后交费,这要求煤气生产部门必须先垫资金。针对这种现状,为了解决收费抄表等问题,需要进行智能煤气表的开发研制。过去的开发基本上可以分为两类:一类是参考 IC 卡电能表的形式,采用 IC 卡,甚至采用电磁阀来控制用户煤气支管,这种形式实际使用不现实。其主要原因是表具的供电问题。煤气表和电能表不同,煤气属易燃易爆气体,不能在煤气表具上接市电电源,一般须用电池,而电池有使用期限问题,如果未及时更换电池或电池损坏掉电则会造成气量纠纷,即经济纠纷,更别提控制煤气管了。另一类是户外抄表煤气表形式,这种表具无需电源,只在常规表具数码盘上加了 BCD 码输出电路,将煤气表头上每一位数码接至串行输出电路即可。抄表时只需将手持抄表器插入户外插头就可以将用户表头数据读出,读出表头数据时由手持抄表器提供串行输出电源,抄完数后表具恢复机械计数方式。这种表具解决了抄表问题,但不能实现预收费,另外由于用户煤气表一般安装在较潮湿的灶台下,长期使用后表头 BCD 码编码金属片和数码盘接触不良,易造成读数不准确。为了解决煤气抄表、收费管理等问题,必须开发一种新的智能煤气表,从根本上解决存在的问题。采用 AT89C2051 为核心构造的预收费智能煤气表就是解决上述存在问题的一种有效方法之一。

9.1.1 预收费智能煤气表的结构原理

所谓预收费智能煤气表就是需要先交钱后用气的煤气表。为方便用户,煤气表只是提醒用户交钱,并不会因为用气超过预付款而停止供应煤气,这样有利于有特殊情况的用户可及时用煤气,同时,又能督促用户及时付款。当用户付款时,管理机构会把用户存在的欠款及预收款一齐收下。用户也就可以在预交款的情况下用煤气。故预收费智能煤气表是一种检测和监督仪表。它的功能主要有 3 个。

- (1) 检测煤气表的读数,并显示给用户。
- (2) 进行用气情况存储及用气报警。
- (3) 和煤气管理计算机系统进行信息交换,实现报表、抄表功能。

预收费智能煤气表的结构框图如图 9-1 所示。

读数传感器把煤气表的实际读数从煤气表中取出,再送到单片机 AT89C2051 中去。这是一个最关键的环节。因为智能煤气表要解决的最重要的问题就是抄表、计费。把煤气表中的用气量读取出来,就是实现了抄表工作。

单片机在读取了煤气表的气量之后,就把所读的煤气量存入电路的电可改写存储器 EEPROM 中。由于 EEPROM 是非易失性存储器,在取消了电源之后仍能把所存储的信息保存下来,故它可以长期存放用户的用气量。EEPROM 存储了用户所用的煤气量,就可以供煤气管理系统的微型计算机通过系统的串行总线进行访问。

LCD 显示器用于显示用户所用的煤气总量以及用户预付款余款所对应的煤气量。很明显,LCD 显示器是用户用于监视用气情况的。一般的机械煤气表给出的煤气量,是历史所用的煤气总量,它无法依据用户的预付款、预付款前的历史用气总量、预付款后所用的气量来计算余款所对应的剩余气量。但是在智能煤气表中单片机可以完成这一功能,并在 LCD 上显示出来。

声音报警电路是一种安全防范的预报电路。在用户的煤气用量接近预付款所对应的用量时,则产生报警,从而提醒用户应该及时再交新的预付款了。这种报警有效地提醒用户,防止透支用气。这样,既可使用户以最合理的价格用气,又保证了煤气公司的资金及时回收。按照一定的规则,凡透支用气,则透支部分煤气价格会不同,另外迟付款要补交滞纳金。所以,预付款报警是必须和合理的。

智能煤气表实际上只是煤气计算机管理系统的一个检测终端。所以,每一个智能煤气表都需要通过串行接口和计算机管理系统的主机相连。因此,需要用串行接口转换电路把单片机 AT89C2051 的串行信号转换成标准的 RS-485 信号,以供主机的串行通信接口接收和发送。在主机和各个智能煤气表终端通信的结构中,是采用标准的 RS-485 串行总线协议的。主机通过串行总线和单片机 AT89C2051 通信,从而读取用户所用煤气的总量,同时,主机还可根据用户预付款的情况、煤气价格,处理之后送回单片机,从而单片机可以通过 LCD 显示余款的气量。

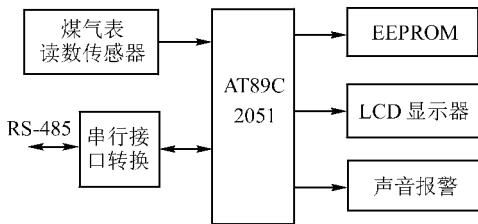


图 9-1 预收费智能煤气表电路框图

是可编程带看门狗监控的串行 EEPROM,存储容量为 512 字节,可提供最少为 10 万次擦写。芯片 8 引脚封装,和 AT89C2051 接口电路如图 9-1 所示。

EEPROM 的读写过程都是类同的。X25045 的串行读写控制和第六章中 6.3 节所介绍的串行 EEPROM 接口方法类似。具体工作过程可参考相关的 X25045 数据手册。

3. LCD 显示电路

由于单片机 AT89C2051 没有 LCD 的驱动接口,故不能直接驱动 LCD 显示屏。为了能够采用 LCD 显示器进行用户用气量显示,以便节省电源,故采用 8 位数字显示的 LCD 显示模块。LCD 显示模块内部已含有显示驱动电路,并且外部接口简单,只有两条接口线,其中一条是数据输入线 DI,另一条是串行移位时钟线 CLK。只要在数据线上恰当送入有关数据的串行信号,则在串行时钟 CLK 的同步之下会送入到 LCD 模块内,再由模块内部的控制电路处理之后实现正确显示。在智能煤气表电路中,单片机 AT89C2051 通过 P1.6 和 P1.7 与 LCD 模块相连,其中 P1.6 用作数据线;P1.7 用作串行时钟线。为了减少电路的功耗,用晶体管 N1 作为开关对 LCD 模块的电源 V_{DD} 进行控制。当不显示气量信息时,AT89C2051 的 P3.7 输出低电平,则晶体管 N1 截止,LCD 模块不供电;当显示气量信息时,单片机 P3.7 输出高电平,则晶体管 N1 导通,从而对 LCD 模块供电,显示送入的气量数字值。

4. 按键接口电路

在智能煤气表中,按键有 2 个:一个是复位按键,一个是查询按键。

复位按键的两端分别接电源 V_{CC} 和单片机 AT89C2051 的复位端 RST。当按键按下时,则高电平 V_{CC} 加到 RST 端,强行使单片机进入复位状态。智能煤气表执行程序的初始化,并进行自检,显示自检的结果信息。

查询按键的两端分别接地和单片机的 T0 输入端。T0 端处接了一个 $20\text{ k}\Omega$ 的上拉电阻。当按键未按下时,T0 输入高电平;按键按下时,则 T0 输入低电平。AT89C2051 内部定时器 T0 执行计数,并溢出产生中断请求;单片机执行 T0 中断处理,查询用户的用气总量及剩余气量,并显示给用户。

5. 远程通信串行接口电路

煤气计算机管理系统是一个信息采集管理系统。它由主机、串行通信线路和智能煤气表组成。主机是一台微型计算机系统,串行通信采用 RS-485 标准协议。智能煤气表是系统的检测终端,检测过程采用广播方式。由于智能煤气表采用 8 位单片机控制,故智能煤气表的表地址也采用 8 位编码,因此,在系统中最多可以接 256 个智能煤气表。

由于单片机的串行口的输出性能不允许进行长距离带负载通信,所以在单片机的串行口和主机的串行通信线路之间要进行串行接口转换,以把单片机的串行接口转换成 RS-485 串行接口,以便远程通信。

串行接口转换电路采用 MAX3082。它是有 8 条引脚的集成电路,内部有一个接收器和发送器。接收器有一个输入端,一个输出端,同时还有一个有效控制端;发送器也一样。故包括电源 V_{CC} 和地引脚共有 8 条引脚。

MAX3082 是 MAXIM 公司的 3080 系列 RS-485 接口芯片。它既符合 RS-485 规范,同时又提供故障保护输出的集成电路。一般情况下,当 RS-485 总线存在短路、开路或简单空闲时,差分总线电压变成 0.0 V ,但 RS-485 接收器的电压阈值规定为 $\pm 200\text{ mV}$ 。由于 0.0 V 处于上下阈值之间,故在故障情况下接收器的输出状态是不确定的。为了解决故障时逻辑状

态问题,MAXIM 公司把阈值通过芯片内整定为 -50 mV 和 -200 mV 。由于上限阈值为 -50 mV ,而故障时总线电压为 0.0 V ,从而保证使接收器内产生逻辑“1”。

另外,为了使电磁干扰以及终端未匹配的连线的反射最小,MAX3082 的转换速率有严格的规定,它最适合于在速率为 115 kbps 以及 115 kbps 以下应用。MAX3082 的工作电压为 $+4.5\sim+5.5\text{ V}$ 。静态工作电流为 $600\text{ }\mu\text{A}$ 。

单片机和主机通过 MAX3082 采用半双工工作方式。单片机平时处于接收状态,等待主机的查询信号。当接收到地址串行口中断信号时,则单片机进行分析。首先判别第 9 位是否为“1”,如果是,则说明是地址串行口中断信号;接着判别是否为本煤气表地址,如果接收到的地址是本煤气表地址,则通过串行口及 MAX3082 把有关信息如所用煤气总量等回传给主机,在传送完毕之后,单片机的串行口又处于接收状态,等待主机下一次的地址串行口中断信号。

6. 电源及供电问题

煤气计算机管理系统的系统框图如图 9-4 所示。它由主机和多达 256 个智能煤气表组成,在主机和智能煤气表之间采用 RS-485 串行总线相连并执行信息通信。

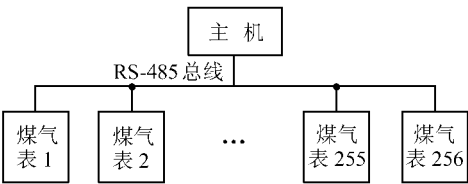


图 9-4 主机和智能煤气表的连接框图

如图 9-2 所示。采用集中供电主要是考虑到每个智能煤气表只要一和 RS-485 相连即可实现供电,而无需单独设置直流电源或交流稳压电源。这样有安装方便、简单、无需进行单独电源维护等特点。

在主机附设的电源中,还采用可充电电池和交流电源整流稳压同时工作的方式。当有工频电源供电时,则由工频电源供电,同时对电池充电;当工频电源停电时,则由电池供电,保证智能煤气表的正常运行。

9.1.3 控制软件的功能

软件主要分为 4 个模块:外部中断计数模块、查询按键显示模块、通信模块、报警模块。为了降低功耗,单片机大部分时间运行在待机状态,当产生外部计数中断、通信中断、按键中断、复位时,单片机退出待机状态,并执行相应程序。

1. 外部中断计数模块

INT0,INT1 中断设为下降沿触发中断。和 INT0,INT1 中断相对应,软件设置两个中断标志:Flag0,Flag1。当表头数码盘转动至干簧管 P1 时,产生 INT0 中断,设置 Flag0=1,Flag1=0;数码盘转动至干簧管 P2 时,产生 INT1 中断,这时清除 Flag0 和 Flag1,读数计数器加 1,如此周而复始执行,累加计数器可以跟踪表头数码指示数据。

2. 查询按键显示模块

按查询键后,产生计数器 T0 溢出中断,进入中断处理后显示用气量和剩余气量。显示器为 8 位 LCD,用气总量和剩余气量采用切换显示方式,各显示 2 次后关闭。

3. 通信模块

智能煤气表通过 RS-485 串行远程通信口和主机系统相连接,当上位主机所传送的地址选中一个煤气表时,该表就将总用气量及相关信息回传给主机,由主机进行信息管理。对于主机来说,得到了煤气表用气总量就可以了。用户购气时,主机将底数即用气总量加上购气量即为用户可用气总量的下限。

4. 报警模块

智能煤气表用声音报警,主要作用是提醒用户剩余气量已抵下限,应去购气并提交预付款。其另一个作用是智能煤气表工作异常时,如受外界强电磁场、人为损坏线路等影响时,上位主机选中该煤气表声音报警事件回传给上位主机。

9.2 水位记录仪

在水文工程、地质勘探、环境监测、水文气象、矿山防治水、工农业供排水、水情预报和地震监测等部门工作的水文工作者的一项重要任务就是掌握地表水、地下水和降水的动态变化,研究其变化规律。但目前测试水位的手段仍然落后。一部分地区还采用原始的手工测量方式,它大大增加了测报人员的劳动强度,同时由于水动态需长期监测和记录,每日处理数据量大,给测报者分析数据带来很大的麻烦和困难。另外也有一部分地区采用浮子式水位记录仪。这种仪器以模拟方式将水位变化数据记录在滚筒式记录纸上,在实际读取数据时,则需要较多的人工推算,其劳动强度大,精度低。而且上述两种方法得到的数据都不能直接用计算机进行处理。在实际应用中,急需一种带多种功能,能对水位变化自动监视、自动测量、对数据自动采集、用智能 IC 卡自动储存水位数据的水位记录仪,且这种记录仪能使用计算机直接处理数据。

这一节介绍的水位记录仪,是一种以 AT89C51 单片机为核心控制器的智能化仪器。它能够自动监视水位变化,自动跟踪并记录每一个水位变化值及水位变化值的时间,用智能 IC 卡储存水位数据。它无需人工看守,用户只需定期前往将存储数据的 IC 卡取回,并插入微型计算机中绘制出水位动态变化曲线即可,特别适用于野外水位长时间监测使用。该仪器充分地考虑了单片机内部硬件资源的利用,并且注意了元器件的节约和优化系统设计,使该仪器成为一种稳定可靠、智能化程序高、成本低、体积小、便于携带、适应性好的产品。

9.2.1 水位记录仪的基本性能

根据水位记录仪应用的特殊环境以及工作性质,对水位记录仪的基本性能有一定的要求,才能满足对应环境及条件下进行水位检测和记录的需要。对水位记录仪所采用的有关水位检测方法、记录方法、以及一些功能特点的简要介绍如下。

水位记录仪采用重锤自动跟踪水位的方法,即通过电子开关控制电机电源,带动可逆电机,使两根测量探针自动跟踪水面。

该仪器能够系统地自动测量每个水位变化值,并由 AT89C51 单片机采集水位值和测量水位值的时间,并存储于机内 IC 卡中。

根据具体情况需要,一个月或半个月或一个星期定期前往安装了水位记录仪的观察点,在水位记录仪中换上新的智能 IC 存储卡。同时,将已存储水位资料的智能 IC 存储卡取回,插入微型计算机中进行数据处理。

水位记录仪的性能指标有如下几点：

- (1) 灵敏度: 水位每变化(上升或下降)0.5~1 cm 时, 则进行响应;
- (2) 测量的误差小于 1 cm;
- (3) 监测水位变化幅度为 ± 50 m;
- (4) 功耗低, 工作电流不大于 100 mA;
- (5) 仪器总体结构采用圆柱型外壳, 并由铝合金铸成, 底盘与支承方螺纹铸成一体, 这既便于安装, 又能增强整体刚性。整个内部分为两层, 上层为电池盒, 便于更换, 下层容装所有机体。

9.2.2 水位记录仪控制逻辑结构

控制逻辑系统由单片机系统、自动跟踪、数据采集、显示和智能 IC 卡数据存储等部分构成。在仪器中充分地考虑了单片机内部硬件资源的利用, 并且注意了元器件的节约和优化系统设计, 使该仪器构成了一个完成用户功能要求的最小系统。其框图如图 9-5 所示。

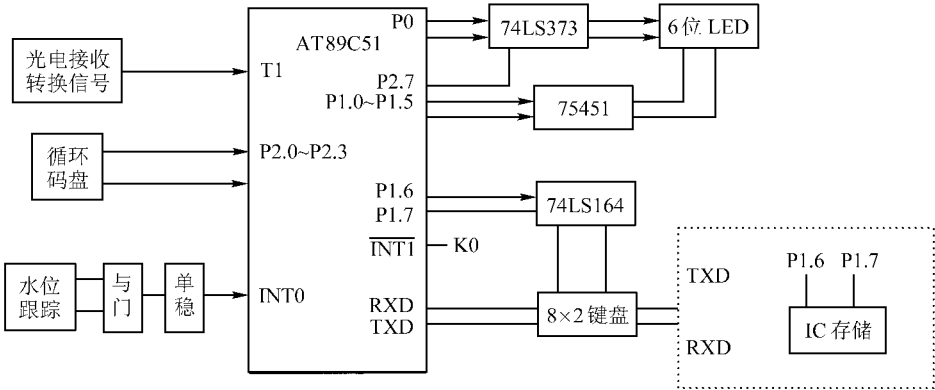


图 9-5 控制逻辑结构

水位记录仪主要是在野外使用, 耗电量小为首要准则和要求。因此选用单片机 AT89C51 为其核心控制器。程序利用 AT89C51 片内 EEPROM 存放, 数据存储用 EEPROM 做成的 IC 卡。因此在关机后或意外情况下可有效地实现数据保护。由于野外工作时一般不需要显示, 因此采用通常使用的 LED 7 段显示器, 并加上控制开关, 需要校对数据时才闭合开关显示。这样就达到了既节省电能, 又不影响显示功能的目的。下面介绍一些主要硬件电路功能。

1. 水位跟踪

水位跟踪由一个独立的检测头和电子开关组成的升降控制电路完成。检测头由两根金属探针和一个电阻 R 组成。当水位发生变化使检测头探针浸入水面之下时, 继电器控制电源正向加至电机, 使电机正转带动探针上升, 跟踪水位; 当水位发生变化下降并令检测头探针全部离开水面时, 继电器控制电源反向加至电机, 使电机反转带动探针下降, 跟踪水位, 直到探针接触水面且水位停止变化为止。此时继电器不吸合, 故电机不转动。同时继电器的一组触点信号经与门和单稳态电路处理, 将此信号送入 AT89C51 单片机 INT0 端口, 向 CPU 申请中断, 采集数据。

2. 数据采集和量程

当电机转动, 带动探针跟踪水位的同时, 电机输出的转距经蜗轮蜗杆减速 ($N=1/150$,

M=2)后驱动循环码盘工作,记录水位数据个位即厘米位,在循环码盘的双侧安有一对光电接收装置,完成厘米位向高位进位的工作。接收的光电脉冲送入 AT89C51 的计数器 T1,向 CPU 申请中断进行计数。同时此转矩再经二级齿轮减速后带动卷线筒转动,从而实现探针的升降和水位变化的显示,达到跟踪水位的目的。由于卷线筒中心是一个 $M10\times 0.3\text{ mm}$ 的内螺孔,它的转轴是一长为 130 mm 的 $M10\times 0.3\text{ mm}$ 螺杆,所以当卷线筒转动时它还沿轴线方向平动,由此完成测量传感线在卷线筒上自动排线的任务,并保证了测量的精度为 1 cm 。卷线筒长度 $L=54\text{ mm}$,直径 $D=94.74\text{ mm}$,传感线直径 $d=0.25\text{ mm}$,按公式 $H=\pi DL/d$ 计算,可得水位记录仪的测量范围 $H=50\text{ m}$ 。

3. 水位数据存储

为方便用户随时提取数据,故将数据存入 EEPROM 智能 IC 卡中。单片机通过 RXD, TXD 串行口输出数据到 IC 卡中。P1.6 和 P1.7 为 IC 卡判断输入信号。ATMEL 公司的 AT24C 系列为串行 EEPROM,工作电流为 3 mA ,静态电流为 $30\sim 110\text{ }\mu\text{A}$ 。它具有可靠的写保护措施,擦写次数为 10 万次,数据保护时间 100 年,故智能 IC 卡采用 AT24C 系列 EEPROM。

4. 键盘电路

由 P1.6 和 P1.7 以及串行口控制一个 8×2 的键盘。键盘的作用是将有关参数例如日期、时钟、水位初值等输入单片机内。一般情况下,当把各参数送入机内后,键盘就脱机不用,只是在更改参数时才重新接入。因此该键盘与 EEPROM 存储器合用 TXD 和 RXD 串行口,并由 INT1 上的按钮开关 S0 选择。

9.2.3 水位记录仪的软件结构

水位记录仪软件编制采用了模块化程序设计思想,整个系统分为主程序模块、键处理和参数输入模块、时钟日历子程序、数据处理子程序、数据采集子程序等。这样做具有程序功能强、成功率高、调试方便等优点。下面介绍这些主要程序功能。

1. 主程序

主控模块是整个软件的核心,主要完成系统的初始化,以及一些功能的判别、中断的响应等。其流程如图 9-6 所示。当初始状态时,将键盘接入。按下开关 K0,程序初始化后判断有 $K0=1$,则程序进入键盘处理和参数输入、显示模块。这时应通过键盘输入参数和日期、时间,然后进入等待中断状态。

如果程序判别的结果有 $K0=0$,即开关 K0 并没有闭合,则按照 Watchdog 工作的情况进行复处理,处理完毕仍然等待中断请求。

2. 键处理和参数输入、显示模块

该模块主要是水位记录仪在初始状态时,需要输入时钟日期参数和水位基准参数值时用。键盘输入采用扫描方式。

首先程序判别开关 K0 的状态,如果 $K0=1$,说明开关 K0 闭合,则这时串行口和键盘连接,并和 IC 卡电路脱离。利用串行口的输出信息以及 P1.6, P1.7 的状态,可以判别 8×2 键盘中的按下按键。

在显示时,显示数字的显示码在 P0 端口输出,而位选择

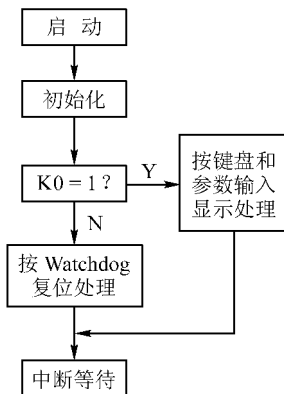


图 9-6 主程序框图

信号从 P1.0~P1.5 输出。显示数字时也是采用扫描方式。所以,在 P1.0~P1.5 输出的选择信号在某一时刻只有一位为选择电平,其余为非选择电平,并且,选择电平是按 P1.0→P1.5 方向一位传向一位,从而可以扫描 6 位的数字显示器。

3. 时钟及日历程序

在记录水位的同时,需要记下与水位值对应的时间。故利用单片机 AT89C51 的定时器 T0 通过软件计数实现内部时钟。为提高计时精度,设置定时器 T0 中断周期为 1 ms,并通过累计中断次数和设置不同进制,获得秒、分、时、日、月、年形成日历和时钟。对时钟而言,采用一个字节存储 2 位 BCD 码。在这 6 种不同进制中,每月天数各不相同,二月份平年和润年天数也不同,因此“日”的时制是一个变数,在其进位时,必须由月和年的当前值来确定。

4. 数据处理中断服务程序

该子程序是当光电信号从 T1 口产生中断请求时的服务程序。根据量程为 50 m,并精确到厘米级的要求,采用一个字节储存 2 位 BCD 码,共需 2 个字节。低位前半字节为厘米位,故从循环码直接读数,分米位在每中断一次时计数加 1,分米位满 10 则向前进位改写相应字节,实现对半位加 1 操作,以此类推。程序流程如图 9-7 所示。

5. 数据采集中断服务程序

当水位跟踪信号从 INT0 产生中断请求时,进入该程序工作。主要任务是采集当时的日期、时间、水位数据,然后将这些数据输入智能 IC 卡存储。每次中断产生时,所采集的数据都在上一次地址的基础上把地址加 1 后存储。程序流程如图 9-8 所示。

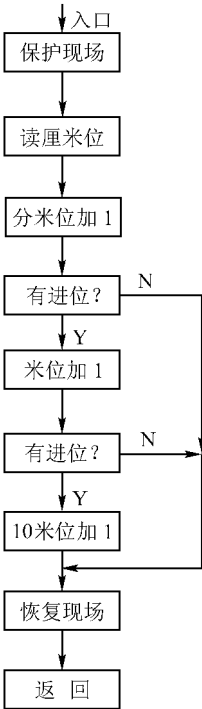


图 9-7 数据处理中断程序框图

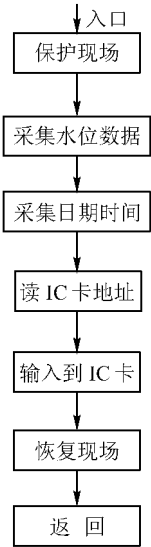


图 9-8 数据采集中断服务程序框图

在水位记录仪的控制软件中,除了上述主要的程序功能之外,还有其他一些特别的功能,这些功能包括水位误差修正、程序盲飞的处理等。

在水位记录仪的物理结构中,由绕线筒长度、传感线直径及排线问题引起的误差,局部时虽然很微小,但累计起来不容忽视。为了克服误差的累积,通过软件进行误差修正,从而在最终结果保证了误差小于 1 cm。

除此之外,在程序中,还加有 Watchdog 软件。当程序被干扰而产生盲飞从而超过规定时间时,系统转入错误处理程序,执行软件复位。在每个模块中都设置有若干个软件陷阱,当程序由于意外原因落入陷阱时,机器会自动执行软件复位。软件陷阱一般安排在正常条件下不会被执行的位置。因而不管程序受到什么干扰而产生不正常的跳转,都会由软件复位使程序回复到正常流程,这样就达到了连续无误工作的目的。

9.3 在线湿度检测仪

湿度测量在自动化系统及其他许多高新技术领域中占有日益重要的地位。目前国内外各种湿度仪大多数都是便携式的,如电容式的、陶瓷式的镜面湿度仪、电解式湿度仪等等。而在线湿度仪的种类不是很多,这里介绍一种采用单片机技术研制出的氧化铝膜状湿度传感式在线湿度仪。

在线湿度检测仪具有在线自动检测露点湿度,对检测数据进行实时采集,对历史数据实行分析处理,采用 LCD 显示屏显示数据,以及与上位主机进行通信等功能。并且,在逻辑电路上和软件设计上采取了各种抗干扰的措施,所以,在线湿度检测仪能够在恶劣的环境或有严重干扰的工业现场长时间稳定地运行,具有可靠性高、精度高以及检测的重复性好等明显优点。

9.3.1 湿度传感器

湿度传感器采用氧化铝膜状湿度传感器,这种传感器和电容式、陶瓷式等湿度传感器不同。氧化铝膜状湿度传感器主要由保护层、屏蔽层、活性电容、电极层和基底等构成。传感器工作时,活性电容绝缘层的水气与周围气体平衡,屏蔽层主要使传感器与外界隔绝,减少外界非湿度因素对传感器的影响,保护层用以保证传感器免受污染。氧化铝膜状湿度传感器屏蔽层和保护层的渗透性良好,环境湿度的变化可以直接通过活性电容的变化反映出来。它是目前环境湿度及过程湿度检测的主流传感器,其主要优点为:

- (1) 响应快速,能准确快速地反映对象湿度变化特性;
- (2) 湿度检测范围宽($0.5 \sim 50\,000$ ppmv),对环境温度要求宽($-80 \sim +20$ °C);
- (3) 时效度化小;
- (4) 能在 $0 \sim 100\%$ RH 范围内进行全程检测。

9.3.2 湿度检测原理

潮湿和干燥是日常人们对空气中水份含量的一种概括的描述。而在物理学中,对空气中水份含量是用湿度这个物理量来量度的。

用物理学上的定义来说,湿度是表示空气中水蒸气的含量多少的物理量。

湿度的表示方法很多,有混合比、比湿度、体积比、摩尔分数和饱和度等。常用湿度表示方

法是绝对湿度、相对湿度和露点温度 3 种。

1. 绝对湿度

单位体积空气中所含的水蒸气的量,称绝对湿度。绝对湿度也就是指空气中水蒸气的密度,或水蒸气浓度。

绝对湿度用下式来定义,即

$$\rho = \frac{m}{V} \quad (9-1)$$

其中: V 是空气的体积;

m 是空气体积 V 中水蒸气的密度;

ρ 是绝对湿度;单位为 g/m^3 或 mg/m^3 ,即克/立方米或毫克/立方米。

通常把空气分成干空气和水蒸气两部分,因此,空气的压强 p_a 则由干空气分压和水蒸气分压组成,并且有

$$p_a = p_d + p_w \quad (9-2)$$

其中: p_d 是干空气分压;

p_w 是水蒸气分压。

空气的绝对湿度在本质上也是水气湿度,很明显和水蒸气分压有关。故绝对湿度可以用水蒸气分压来表示

$$\text{绝对湿度} = \frac{p_w \cdot m_w}{RT} \quad (9-3)$$

其中: p_w 是水蒸气分压;

m_w 是水蒸气摩尔质量;

R 是理想气体常数;

T 是空气的绝对温度。

用水蒸气分压表示绝对湿度时,单位采用毫米汞柱 mmHg。

2. 相对湿度

在实际生活中,有不少现象并不受绝对湿度影响。例如,食品的干燥、人体的舒适感、水蒸发的速度、衣物的晒干等,这些和绝对湿度的直接关系不大。但是,这些现象则和空气的水蒸气分压与同一温度下水的饱和水蒸气压之间的差值有关。

在同一个地方同一种绝对湿度,也就是同一个水蒸气密度的情况下,白天特别是中午人们会觉得干爽,但在清晨或傍晚人们就会觉得有点潮凉。其原因在于中午温度高,这时的水蒸气饱和气压较高,空气中水蒸气分压和饱和气压的差值较大;而在清晨或傍晚温度低,这时水蒸气饱和气压较低,空气中水蒸气分压和饱和气压的差值较小。为此,人们提出了一个更广泛应用的概念——相对湿度。

相对湿度是指在一定温度下,空气中所含有水蒸气分压 p_w 和同一温度的水的饱和水蒸气压 p_s 之比的百分数。

$$\text{相对湿度} = \frac{p_w}{p_s} \quad (9-4)$$

其中: p_w 是水蒸气分压;

p_s 是饱和水蒸气压。

相对湿度是一个无量纲量,一般用%RH表示。RH是英文相对湿度 Relative Humidity 两上单词的第一个字母。

在不同温度中,水的饱和水蒸气压是不同的。各种温度以及对应的饱和水蒸气压如表 9-1 所列。

在表 9-1 中, $t\text{ }^{\circ}\text{C}$ 是表示摄氏制的温度, p_s 是饱和水蒸气压强。 p_s 的单位用毫米汞柱 mmHg 表示。1 mmHg 等于 136 mm 水柱,也等于 133.32 Pa。

在知道某特定温度中空气的水蒸气分压 p_w 时,从表 9-1 可查出对应温度下的饱和水蒸气的压强 p_s 。接着用式(9-4)就可以十分容易求出相对湿度。空气中水蒸气分压是用露点来求出的。

表 9-1 各种温度中饱和水蒸气压

$t/^{\circ}\text{C}$	p_s/mmHg	$t/^{\circ}\text{C}$	p_s/mmHg	$t/^{\circ}\text{C}$	p_s/mmHg	$t/^{\circ}\text{C}$	p_s/mmHg	$t/^{\circ}\text{C}$	p_s/mmHg	$t/^{\circ}\text{C}$	p_s/mmHg
-20	0.77	-9	2.13	2	5.29	13	11.23	24	22.38	35	42.18
-19	0.85	-8	2.32	3	5.69	14	11.99	25	23.78	36	44.56
-18	0.94	-7	2.53	4	6.10	15	12.79	26	25.21	37	47.07
-17	1.03	-6	2.76	5	6.54	16	13.63	27	26.74	38	49.69
-16	1.13	-5	3.01	6	7.01	17	14.53	28	28.35	39	52.44
-15	1.24	-4	3.28	7	7.51	18	15.48	29	30.04	40	55.32
-14	1.36	-3	3.57	8	8.05	19	16.48	30	31.82	50	92.50
-13	1.49	-2	3.88	9	8.61	20	17.54	31	33.70	60	149.4
-12	1.63	-1	4.22	10	9.21	21	18.65	32	35.66	70	233.7
-11	1.78	0	4.58	11	9.84	22	19.83	33	37.73	80	355.1
-10	1.95	1	4.98	12	10.52	23	21.07	34	39.90	100	760.0

3. 露 点

从表 9-1 中可以看出,温度越低,饱和水蒸气压就越低。如果给定一定的空气水蒸气分压 p_w ,随着温度下降,则 p_w 和饱和水蒸气压 p_s 的差值就越小。当温度下降到一个恰当的值时,空气水蒸气分压 p_w 和这时温度中的饱和水蒸气压 p_s 相等。这时,空气中的水蒸气也就会因进入饱和而凝结成露珠。

使空气水蒸气结成露珠的温度称为露点温度,简称露点。

由于在露点时,空气中水蒸气分压 p_w 和这时温度中的饱和水蒸气压 p_s 相同,很明显,测出空气中水蒸气的露点,查表 9-1 就可以知道空气中水蒸气分压 p_w 。

因此,也可以说,使空气中水蒸气饱和的温度称露点;或者,使空气中水蒸气相对湿度达到 100 % 的温度称露点。

利用露点,可以很容易从表 9-1 查出空气中水蒸气的绝对湿度。

在空气中的水分处于微量水平时,往往采用体积比来表示微量水分。体积比是一个特定的物理量,它的定义说明如下。

空气中水蒸气体积与干气体积之比称体积比。体积比的单位为百万分之一,用 1×10^{-6}

表示。

体积比、空气水蒸气分压 p_w 和空气压强 p_a 之间的关系可以用下式表示,即

$$\text{体积比} = \frac{p_w}{p_d} = \frac{p_w}{p_a - p_w} \tag{9-5}$$

利用式(9-5),结合表 9-1,可以进行体积比、绝对湿度、相对湿度和露点之间的相互转换。

在实际工作中,经常用露点表示空气中的含水量。对湿度的检测在实际处理中是检测露点。氧化铝湿度传感器在外界湿度发生变化时,其输出阻抗也发生变化。采用氧化铝湿度传感器构成的湿度检测探头,由相关电路和氧化铝湿度传感器共同组成,电路把氧化铝湿度传感器的输出阻抗转换成对应的输出电压。这样,氧化铝湿度传感器探头就把湿度变换成对应的电压值,也就是传感器探头把水蒸气压变换成对应的电压值。由于水蒸气压 p_w 和露点 t 之间是函数关系,那么,在得到水蒸气压 p_w 时,可以通过一系列的函数处理求出露点 t 。水蒸气压 p_w 和露点 t 的关系也可以从表 9-1 中求出,而且可以看出它们之间是指数关系,即随着露点 t 的升高,水蒸气压 p_w 会按指数曲线升高。

在氧化铝湿度传感器探头中,其输出的电压值是代表着水蒸气压 p_w 的,因此,很容易知道,该电压和露点也会存在指数关系。

在检测过程中,先把氧化铝湿度传感器探头的输出电压送入到电压频率转换电路,把电压值转换成对应的频率信号,然后,由单片机对频率信号进行检测,同时进行一系列相关的处理,从而可以求出体积比 P 和露点 t 。然后,把这些结果送入到 LCD 显示模块中显示出来。

9.3.3 仪器的控制电路结构

控制系统主要由单片机 AT89C52、湿度检测和传送电路、存储器、串行通信接口电路、键盘输入电路、LCD 显示模块、外部定时监视器 Watchdog 电路组成。

系统构成如图 9-9 所示。信号处理主要采用 OP07 放大器,它具有超低漂移、高精度、高共模抑制比和低失调等优点,特别适宜于微弱信号的精确放大。由于数字信号的传输对环境要求较高,很容易被外来的电磁干扰所“淹没”,而频率信号只是一种时间量,处理频率信号只与时间量有关,因此选用电压频率变换器 VFC32 作 A/D 转换。VFC32 具有抗干扰能力强、线性度高、信号几乎无衰减等优点。单片机测控系统主要由掉电信息不丢失的数据存储器 SRAM MK48Z02/12、串行通信接口 MAX232 以及分频器、LCD 显示器、薄膜键盘片和 AT89C52 单片机等构成。AT89C52 内部自带 8 KB Flash 可编程、可电擦除的只读 EPROM,主控程序存放其中,另外扩展一片 27C256 存放系统字库及表格等。此外,系统还设置了 WDT“看门狗”电路,它由 MAX813L 电压监控/监视定时器来实现,同时还设置了软件陷阱等,故提高了系统抗干扰和故障自诊断的能力。

在控制电路中,由于对湿度的 A/D 转换采用了电压频率变换器,它对整个仪器的质量有很大的影响,故 VFC32 是一个十分关键的电路。

VFC32 是一片电压频率和频率电压转换器,也简称 V/F 与 F/V 转换器。它可以把电压转换成对应信号频率,也可以把频率转换成电压。这种芯片在进行 V/F 转换时,可以对正电压或负电压进行转换,也可以对正输入电流或负输入电流进行转换;而构成这种转换结构只需要少量外部元件。在进行 F/V 转换时,一样也只需要少量外部元件。

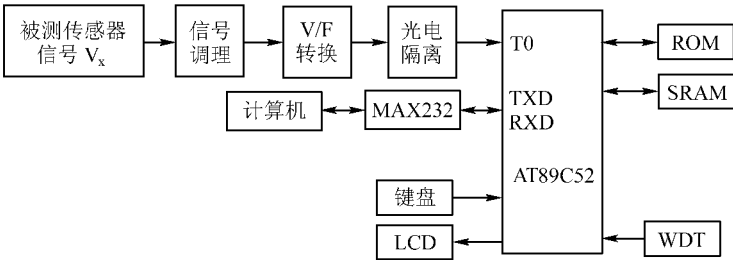


图 9-9 湿度检测仪控制逻辑框图

在 V/F 转换时,由于信号是通过集电极开路的晶体管输出的,故输出频率信号的电压最高可达 30 V,这时只要在频率信号输出端接一个上拉电阻到 30 V 电源即可。由于开路的集电极电流最大为 8 mA,故上拉电阻 R_p 的值应符合 $R_p \geq V_f/8 \text{ mA}$ 的要求,其中 V_f 是频率输出端上拉电阻 R_p 所接的电源电压。在输出频率信号需要和 CMOS 器件相连时, $V_f=15 \text{ V}$;在和 TTL 器件相连时, $V_f=5 \text{ V}$ 。

VFC32 的主要性能指标有如下几点。

- (1) 高度线性。
在频率为 10 kHz 时,最大偏差为 $\pm 0.01 \%$ 。
在频率为 100 kHz 时,最大偏差为 $\pm 0.05 \%$ 。
在频率为 500 kHz 时,最大偏差为 $\pm 0.2 \%$ 。
- (2) 输出信号可以和 CMOS, TTL 信号兼容。
- (3) 可进行 V/F, F/V 转换。
- (4) 可进行电压或电流输入。

VFC32 的内部结构及引脚如图 9-10 所示。从图中看出它主要由 5 个部件组成。这些部件分别是输入运算放大器 A、比较器 C、单稳触发器 M、恒流源开关 K 和集电极开路晶体管 B。

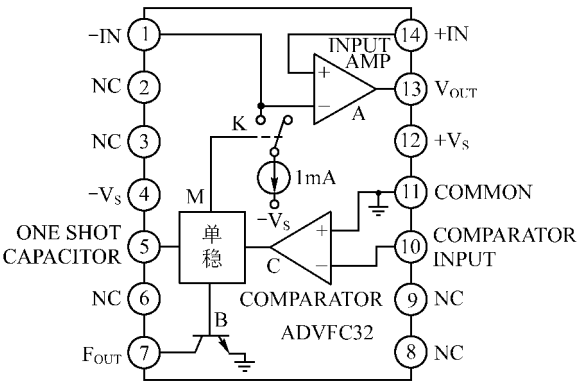


图 9-10 VFC32 结构框图

VFC32 的引脚含义如下。
引脚 1: -IN, 输入运放的负输入端。

- 引脚 4: $-V_s$, 芯片负电源。
- 引脚 5: 单稳外接电容器端。
- 引脚 7: 晶体管的开路集电极输出端。
- 引脚 10: 比较器的信号输入端。
- 引脚 11: 比较器的信号输入公共端。
- 引脚 12: $+V_s$, 芯片的正电源。
- 引脚 13: V_{OUT} , 输入运算放大器的输出端。
- 引脚 14: $+IN$, 输入运放的正输入端。
- 其余引脚 2, 3, 6, 8, 9 等均为无用引脚。

正电压输入的单极性 V/F 转换过程如图 9-11 所示, 其转换过程基于输入信号幅度与内部 1 mA 电流源比较。当输入电压 V_{IN} 加到引脚 1 时, 则通过 R_3, R_1 产生电流对 C_2 充电, 这时, 1 mA 电流源和输入运算放大器的负输入端不相连。随着 C_2 的充电过程, 输入运算放大器 A 的输出下降, 当其输出电压达到 0 V 时, 则比较器 C 触发单稳触发器 M, 单稳触发器输出令晶体管 B 截止, 则在引脚 7 输出高电平; 在此同时, 单稳触发器的电容 C_1 开始反向充电, 其充电过程和内部的参数有关, 当反向充电达 -3.4 V 时, 则单稳触发器翻转, 恢复原来的稳态, 如图 9-12 所示, 在引脚 7 输出低电平。这段时间是单稳触发器的延时时间, 并记为 t_{os} , 它由下式计算出来, 即

$$t_{os} \approx (C_1 + 44 \text{ pF}) \times 6.7 \text{ k}\Omega$$

(9-6)

用式(9-6)计算出来的时间 t_{os} 一般采用 ns 为单位, 原因是一般输出频率都大于 10 kHz。

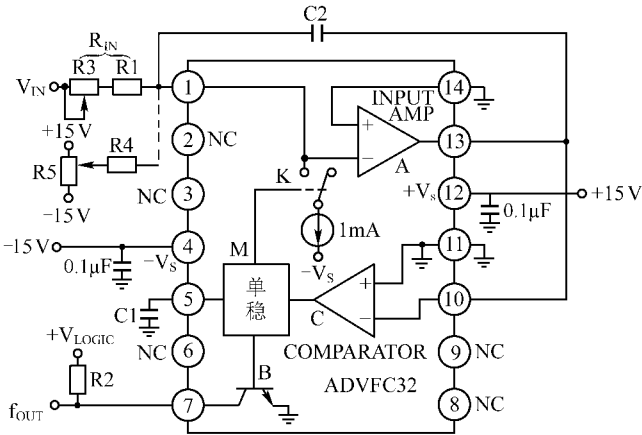


图 9-11 正电压 V/F 电路

在单稳触发器 M 被比较器 C 触发后, 单稳触发器 M 同时控制恒流开关 K, 使 1 mA 恒流源和输入运算放大器的负输入端相连。如果用 I_{IN} 表示 V_{IN} 所产生的输入电流, 则当 1 mA 接入引脚 1 时, 产生的电流为 $(1 \text{ mA} - I_{IN})$, 这个电流使积分电容 C_2 进行放电。在单稳触发器 M 的延时时间为 t_{os} , 则 C_2 所放的电荷为

$$(1 \text{ mA} - I_{IN}) \times t_{os}$$

(9-7)

V/F 转换电压波形如图 9-12 所示。

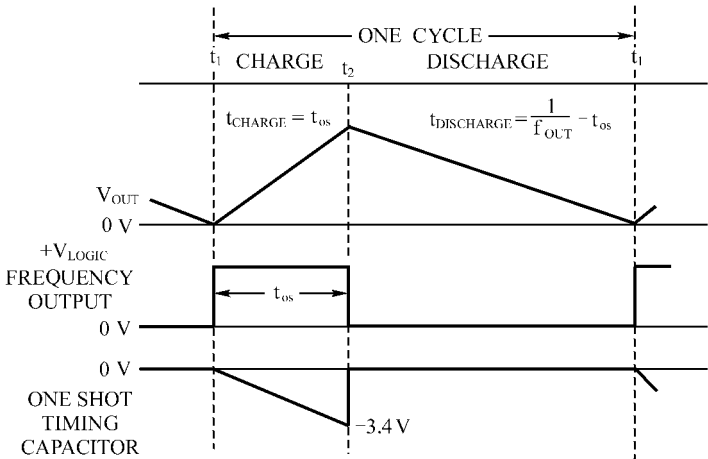


图 9-12 V/F 转换电压波形

考虑到积分电容 C2 这时应把原来 V_{IN} 所充的电荷全部释放掉,而 VFC32 产生的输出信号为 f_{OUT} ,则信号周期为 $1/f_{OUT}$ 。很明显: t_{os} 是输出信号为高电平时间, $[(1/f_{OUT})-t_{os}]$ 则是输出信号为低电平时间。容易理解, V_{IN} 对电容 C2 充电的电荷为: $I_{IN} \times (1/f_{OUT} - t_{os})$,这样,则有

$$(1\text{ mA} - I_{IN}) \times t_{os} = I_{IN} \times \left(\frac{1}{f_{OUT}} - t_{os} \right) \tag{9-8}$$

对式(9-8)进行整理,则有

$$f_{OUT} = \frac{I_{IN}}{1\text{ mA} \times t_{os}} \tag{9-9}$$

在输入 V_{IN} 的回路中有 R1,R3 存在,其中 R3 是可调电阻,当 R3 调节稳定后为一定的阻值,这时 R1,R3 串联为电阻 R_{IN} ,则 V_{IN} 产生的输入电流 I_{IN} 为

$$I_{IN} = V_{IN}/R_{IN} \tag{9-10}$$

把式(9-6)和(9-10)代入式(9-9),有

$$f_{OUT} = \frac{V_{IN}/R_{IN}}{1\text{ mA} \times (C1 + 44\text{ pF}) \times 6.7\text{ k}\Omega} \tag{9-11}$$

式(9-11)是描述 VFC32 工作的数字方程。

对于所需的输入电压 V_{IN} 和输出频率 f_{OUT} ,为了优化考虑,可按下列方程选择有关参数:

$$C1 = \frac{3.7 \times 10^7\text{ pF/s}}{f_{OUT}} - 44\text{ pF} \tag{9-12}$$

$$C2 = \frac{10^{-4}\text{ F/s}}{f_{OUT}} \tag{9-13}$$

$$R_{IN} = \frac{V_{IN}}{0.25\text{ mA}} \tag{9-14}$$

$$R2 \geq \frac{+V_{LOGIC}}{8\text{ mA}} \tag{9-15}$$

在上面公式中,F 表示物理量法拉,s 表示物理量秒。

对于 R_{IN} 和 C1,要求它们有很低的温度系数,以防止因温度变化而影响 V/F 的转换精度。其余元件的温度系数并不需过于严格要求。一般建议采用的输出频率、输入电压、以及 C1,

R_{IN} , $C2$ 的值如表 9-2 所列。

R_{IN} 由 $R1$ 和 $R3$ 组成, 其中 $R3$ 是可变电阻。在系统考虑时, $R1 = R_{IN} \times 90\%$, $R3 = R_{IN} \times 20\%$ 。这种 $R3$ 的变动可以允许有 $\pm 10\%$ 的增益调节, 从而补偿 VFC32 的全标度偏差以及 $C1$ 偏差。

为了进一步提高精度, 可采用 $R4, R5$ 组成的偏置电路。 $R5$ 取值为 $10 \sim 100\text{ k}\Omega$ 之间, $R4$ 取值均为 $10\text{ M}\Omega$ 。

V 还可用于负电压、双极性电压转换以及 F/V 转换, 只需外部元件稍作改动即可。

表 9-2 V/F 转换时有关参数

V_{IN}/V	f_{OUT}/kHz	$C1/\text{pF}$	$R_{IN}/\text{k}\Omega$	$C2$
1	10	3 650	4. 0	$0.01\text{ }\mu\text{F}$
10	10	3 650	40	$0.01\text{ }\mu\text{F}$
1	100	330	4. 0	$1\text{ }000\text{ pF}$
10	100	330	40	$1\text{ }000\text{ pF}$

9.3.4 在线湿度检测仪的软件

氧化铝探头对气体中水的蒸气压响应值以电参量 U (单位为 V) 表示, 而露点又是水的蒸气压的函数, 通过对 U 和露点 t (单位为 $^{\circ}\text{C}$) 作图, 发现两者呈指数关系: $U = Ae^{Bt} + C$ 。 A, B, C 为特定氧化铝探头常数。对仪器标定时, 常用 $Y = aX + b$ 分段直线拟合的方法来逼近非线性

曲线。随着探头的使用, U 与露点的曲线关系发生变化, 因而 a, b 值也发生改变。图 9-13 给出了探头随使用时间而发生变化的情况。其中, 曲线 (1) 为用标准水分发生器标定的曲线, 曲线 (2) 为探头原设置的响应曲线。因此, 必须对仪器进行校正。根据标定的数据来重新确定 a, b 值, 从而得到新的正确拟合曲线。将该曲线分段直线拟合值存入 SRAM 数据存储器中, 单片机进行测试时, 通过对电参量进行计算、查表、寻找正确的拟合直线段等一系列处理即可得到被测对象的露点值。在实际的程序设计中, 由于体积比 P 与露点 t 之间关系已知, 可按一定的形式

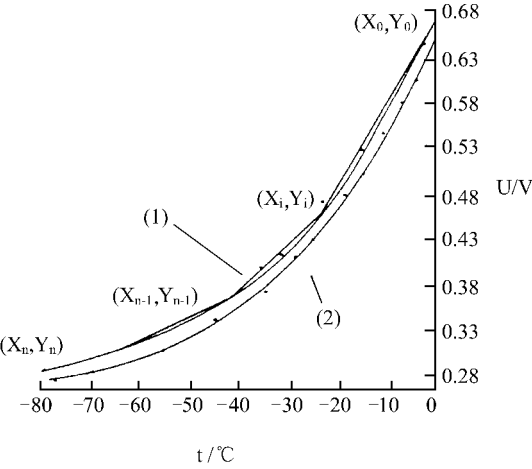


图 9-13 探头的响应曲线

排列存入程序存储器 EEPROM 中, 通过查表求解出 t 。

关系式为:

$$\frac{P - P_{n-1}}{P_n - P_{n-1}} = \frac{t - t_{n-1}}{t_n - t_{n-1}}$$

得到:

$$t = \frac{t_n - t_{n-1}}{P_n - P_{n-1}} P + t_{n-1} - \frac{t_n - t_{n-1}}{P_n - P_{n-1}} P_{n-1} \tag{9-16}$$

式中, P 根据传感器测量信号计算可得, 再由 P 确定 P_n, P_{n-1} , 对应的 t_n, t_{n-1} 通过查表可得到, 这样由式(9-16)即可求出露点值 t 。主程序流程图如图 9-14 所示。

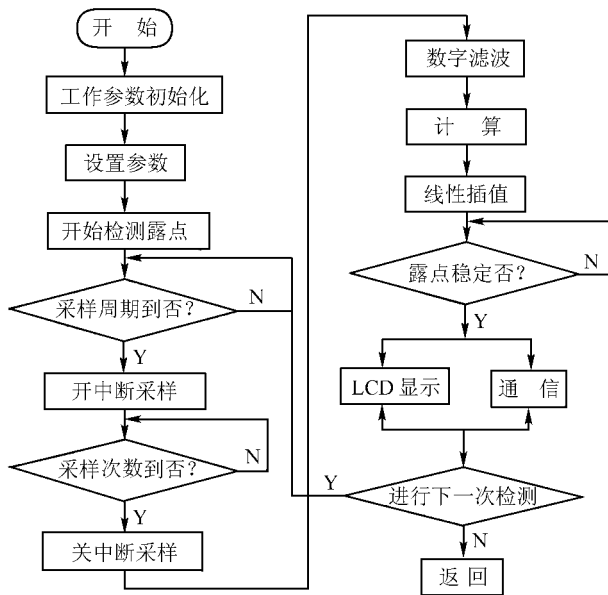


图 9-14 主程序流程图

从图 9-14 中看出: 主程序分初始化和工作过程两大部分。初始化部分主要是完成单片机内部的资源分配处理和工作过程的初始参数设置这两种功能。在工作过程部分, 其任务就是要将湿度检测之后进行相关的处理, 从而得出正确湿度值, 并以体积比和露点的方式显示。在这部分中, 进行中断采样, 并为了保证采样信息的正确性而采用多次采样的方法, 以供数字滤波用, 从而求出正确的结果, 排除干扰信号。同时, 在滤波之后调用线性插值子程序, 通过线性插值求出露点值 t 。在求出露点 t 之后, 把结果送到 LCD 显示模块显示。同时, 也把结果通过串行通信口送到管理级的上位主计算机中去。在完成上述过程之后, 则返回程序的首部开始下一次的检测过程。

中断处理子程序的流程如图 9-15 所示。它是对湿度检测信号进行输入和处理的程序。为了保证所采样的湿度的正确性, 采用多次重复采样的方法。在采样了一批数据之后, 再用数字滤波技术对采样信号中的有关干扰信号进行排除。在数字滤波中可用多种滤波方法。为了采样成批的数据, 故利用这个中断子程序进行多次采样。直到达到给定的采样次数之后, 由主程序关闭中断, 这个子程序才不执行。在这个程序中, 主要是对湿度检测电路送来的频率信号进行计数, 计数是由定时器 T_0 来实现的。通过定时器 T_0 计数而求出湿度检测电路的频率信号的频率值, 并且将其存入存储器中。

线性插值计算子程序如图 9-16 所示。这个程序的任务就是利用所采样得到的体积比 P , 并按照式(9-16)进行处理, 从而可以求出露点 t 。把求出的露点 t 存入存储器, 以供主程序显示以及和上位主机通信时用。

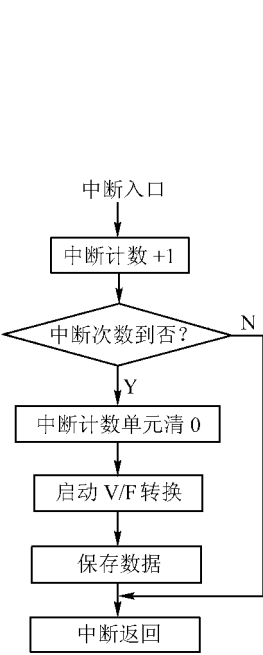


图 9-15 中断子程序流程图

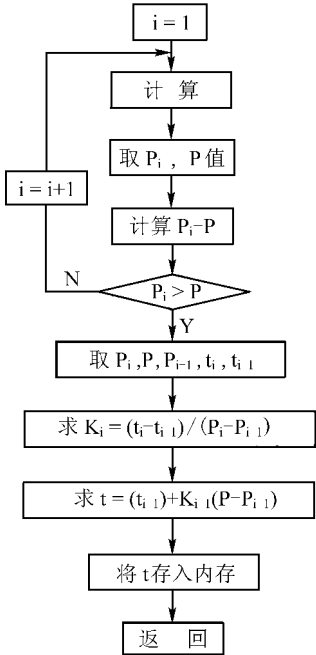


图 9-16 线性插值计算流程图

9.3.5 仪器的一些特点

在线湿度检测仪由于采用氧化铝膜状传感器、LCD 显示模块、外部 Watchdog 等电路,所以它具有较好的特性。首先,采用氧化铝膜状传感器有检测可靠、响应快的特点;采用 LCD 显示模块有利于简化电路设计,且编程简化;利用外部 Watchdog 可提高整机可靠性。另外,还应注意如下几点。

- (1) V/F 采样间隔设定:利用软件延时进行采样,满足 Shannon 采样定理,可以定时采样和储存数据。
- (2) 与上位 PC 主机通信:PC 机能以中断方式控制在线湿度仪的检测,而在线湿度仪可将实时测得的各种数据以中断方式传输至 PC 机,具有历史数据分析、趋势图绘制、储存数据打印输出等功能。
- (3) 检测:氧化铝探头将检测到的信号经过处理、滤波、V/F 转换、分频、整形后,将得到的数据送至单片机系统,单片机经过处理后将露点温度、绝对湿度通过 LCD 屏显示出来。
- (4) 电源:AC 220±10 V;50 Hz。
- (5) 范围:露点为-80~+20 ℃;温度为-10~+40 ℃。
- (6) 测量精度:-60~+20 ℃ 以内测量精度为±0.5 ℃;-80~-60 ℃ 以内测量精度为±1 ℃;在整个测量范围内,重复性测量精度优于±0.5 ℃。
- (7) 通信接口:RS-232 串行口。
- (8) 显示:内置 T6963C 控制器型图形液晶显示器,显示屏大小为 240×128 点。

在线湿度检测仪响应快速,湿度检测范围宽,时效度化小,具有可靠性高、精度高、重复性好等优点,能在环境恶劣的工业现场长时间稳定运行。所程序与测量电路及单片机系统紧

密配合,可以方便地对氧化铝膜状湿度传感器进行线性化处理,同时也适于其他非标准传感器,并可以使其线性化效果达到最优。

9.4 环保设备运行记录仪

环保设备运行记录仪是用于对污染处理设备的运行状况和污染排放情况、指标进行监控的仪器,这种仪器可以记录污染处理设备的运行状况,使环保部门可以有依据地进行污染情况的调研,及时采取有效措施,排除污染的来源或清除污染,从而加强对有污染物质排放的企业或部门的监管力度,保护人类的生态环境,提高人们的生活水准。

环保设备运行记录仪既然是一种仪器,它一方面必须能独立工作,完成其所具有的各种检测和记录任务;另一方面,它应能以一定的方式联网,从而能和上位管理主机联系通信,向其提供所检测的数据,以便进行分析处理。

下面介绍环保设备运行记录仪的主要功能、系统结构以及工作软件。这里介绍的环保设备运行记录仪是以 AT89C52 为控制器组成的。由于记录仪的功能较多,要求单片机的程序存储器有较多的存储单元以及较强的接口功能。

9.4.1 环保监测网与数据记录

环保设备运行监测系统由环保设备运行记录仪、现场检测仪表和传感器、读码器和管理计算机组成。所有的运行记录仪所检测 and 记录的数据都通过通信线送到管理计算机中去。记录仪安装于环保设备运行现场,例如化工厂、电镀厂等有排污过程的单位,全天候监测并记录环保设备的运行状况和主要污染物排放指标;读码器用于控制记录仪的运行及对记录仪进行数据读取;管理计算机用于对原始数据进行统计、报表、存储及查询。记录仪既可单机运行,也可组成区域性环保监测网,如图 9-17 所示。

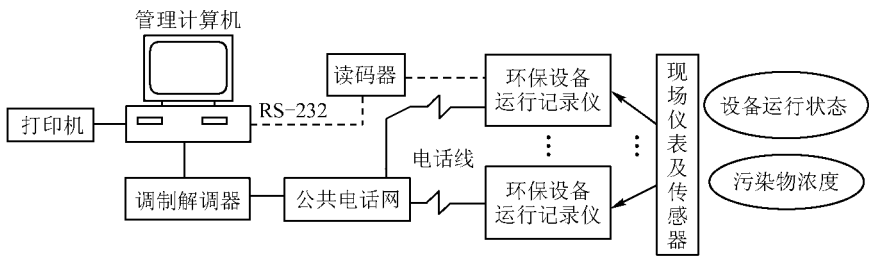


图 9-17 环保监测网结构框图

记录仪是整个系统的关键设备,其主要功能如下。

- (1) 记录仪共有 8 个数据采集通道,可接受来自现场仪表或传感器输出的 4~20 mA(或 0~10 mA)的标准电流信号。每一通道可根据监视对象自由设定为开关量通道,用于实时记录环保设备开机/关机运行情况;也可以设定为累积量通道,用于对排放的污水流量进行累积;或者设定为即时量通道,用于记录排放污染物浓度。
- (2) 记录仪能产生本身的运行记录,例如记录仪上电/掉电时间。
- (3) 可通过读码器或管理计算机设定被监测环保设备的工作电流范围,对被监测环保设备的工作电流发出超限报警。

- (4) 记录仪具有与管理计算机和读码器通信的标准 RS-232 接口。
- (5) 记录仪内置通用调制解调器 MODEM 及 MODEM 控制器,可与管理计算机远程通信。

9.4.2 系统电路逻辑结构

根据功能要求和记录仪的工作特点,记录仪最关键的工作性能有运行可靠性、记录数据准确性及数据存储容量足够大三个方面。记录仪采用 ATMEL 公司的 AT89C52 单片机,片内集成了 8 KB FLASH 程序存储器,从而不需外扩程序存储器,同时也提高了系统的运行稳定性。图 9-18 给出了记录仪的结构框图。

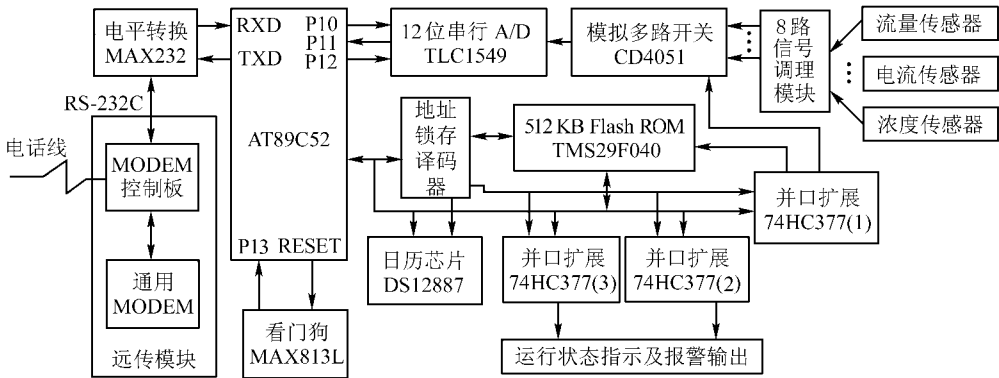


图 9-18 环保设备运行记录仪结构框图

1. 数据采用集通道

数据采集通道由模拟多路开关和串行 A/D 转换器组成。

环保设备在现场处理的结果包括输出的废水、废气等物质,由传感器进行检测,产生相应的电压信号或电流信号,通过 8 通道的信号调理模块,进行信号放大,变成 0~5 V 的直流模拟信号,再由模拟多路开关进行选择,在某一时刻选择一种检测结果传送给串行 A/D 转换器,再将其转换成数字信号,然后送入单片机 AT89C52 中处理。无论传感器所检测的是排污的流量、浓度、酸度或其他指标和物理量,都被转换成标准电压信号,再送入信号调理模块。如果传感器输出的是电流信号,则接上负载电阻就可以转换成电压信号。

数据采集通道的模拟多路开关采用 CD4051,这是结构为 8×1 的开关,即是 8 选 1 的开关。这种开关的性能如下:

- 电源范围:3~18 V;
- 模拟输入输出信号:3~18 V;
- 传送特性:线性;
- 开关特性:导通时低阻抗,断路时低漏电流;
- 开关电流:最大±25 mA;
- 控制引脚输入电流:最大±10 mA;
- 功耗:最大 500 mW。

CD4051 的封装引脚如图 9-19 所示。模拟多路开关 CD4051 还有多种替代型号,包括 MC14051B 等。它们的结构和性能是一样的,封装引脚也一样,可以相互置换。从图 9-19 中

可知,CD4051 有开关部分和控制部分两部分。开关部分有开关 I/O 端 X0,X1,X2,⋯,X7 共 8 个端;另外有 1 个公共 I/O 端——X 端。控制部分有选择控制端 A,B,C;另外还有 1 个禁止端。再加上电源引脚,则 CD4051 有 16 条引脚。在电源引脚中,除了电源电压引脚为 V_{DD} 之外,还有垫片电源 V_{SS} 和地电源 V_{EE} 。一定要注意在应用中电源必须符合要求: $V_{EE} \leq V_{SS}$ 。通常在使用时把 V_{EE} 和 V_{SS} 连接在一起接地。

CD4051 的内部逻辑结构如图 9-20 所示。在图中可以看出逻辑结构由电平转换电路、8 选 1 译码电路和开关电路组成。其中,Inh 是禁止信号,当 $Inh=1$ 时,则 CD4051 的所有开关断路,即芯片不工作;当 $Inh=0$ 时,CD4051 才正常工作。它的开关情况由控制信号 A,B,C 决定。

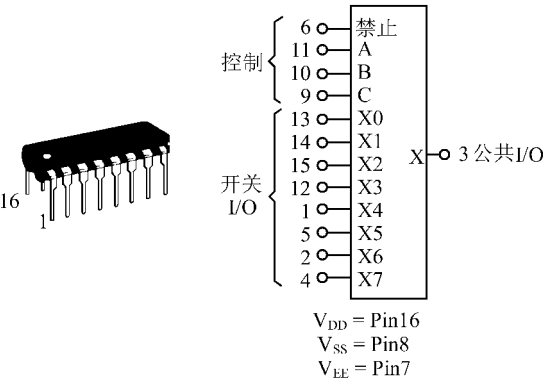


图 9-19 CD4051 的引脚

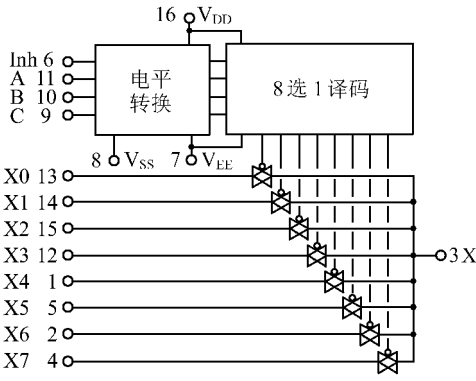


图 9-20 CD4051 逻辑结构

CD4051 的真值表如表 9-3 所列。从真值表中可以看出控制信号 A,B,C,以及开关 I/O 端和公共 I/O 端的关系。当 $CBA=000$ 时,X0 端和 X 端接通; $CBA=001$ 时,X1 和 X 相连接通;⋯; $CBA=111$,X7 和 X 相通。在表 9-2 中,符号“*”表示不管其状态如何,即可为任何状态。在 $Inh=1$ 时,不管 CBA 的状态如何,开关端和公共端都不能接通,所以这时,CD4051 处于全断开状态,CBA 的控制信号不起作用。

表 9-3 CD4051 真值表

Inh 电平	C	B	A	开关端	公共端
1	*	*	*		
0	0	0	0	X0	X
0	0	0	1	X1	X
0	0	1	0	X2	X
0	0	1	1	X3	X
0	1	0	0	X4	X
0	1	0	1	X5	X
0	1	1	0	X6	X
0	1	1	1	X7	X

* 表示可为任何状态。

数据采集通道 A/D 转换器选用了美国 TI 公司的高性能 10 位串行 A/D 转换器 TLC1549。该 A/D 转换器为 CMOS 工艺;采样频率可达 40 kHz;具有较宽的工作电压范围,

为 3~6 V,典型值为 5 V;功耗低;与单片机接口简单,占用资源少,只需用三线接口。

2. 实时时钟

为了能准确记录环保设备的开机/关机时间,记录仪采用了广泛应用于各类工控仪表中的 DALLAS 日历芯片 DS12887。DS12887 内置电池和晶体振荡器,可直接和单片机数据总线连接,它的运行稳定性好、精度高,免维护时间可达 10 年以上,从而满足了记录仪对时间的要求。由于 DS12887 内部有可掉电记忆的 114 字节 RAM,也为记录仪的掉电系统维护提供了极大的方便。

3. 大容量 Flash 存储器

记录仪要求具有掉电记忆的大容量数据存储器,通过比较测试,选用 TI 公司的 Flash 存储器 TMS29F040,它是容量为 512 KB 的 8 位 EEPROM。记录仪除具有扩展 512 KB 的 Flash 存储器外,还有多个 I/O 接口芯片,显然所需地址空间已超出 AT89C52 单片机常规 64 KB 片外 RAM 空间。64 KB 的空间是由 16 根地址线进行寻址的,其中由 P0 口提供低 8 位地址,P2 口提供高 8 位地址。要想扩大外部存储空间,只有增加地址线,如可将 P1 口 I/O 线当作地址线。环保设备记录仪的逻辑电路系统中,扩展 512 KB 的 Flash 存储器需增加 3 根地址线,而 P1 口线仅有 P1.7 可用。为此采用了图 9-21 所示的线选方法,实现了超大容量存储器扩展。

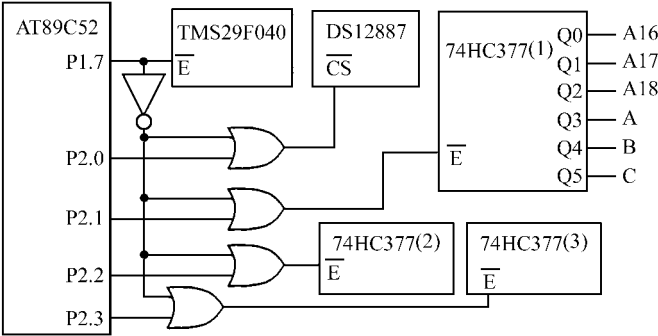


图 9-21 大容量存储器扩展

图 9-21 中的日历芯片为 DS12887。3 片并行输出口 74HC377 的地址与 Flash 存储器 TMS29F040 的地址重叠,但当置 P1.7 为“1”时,Flash 存储器 TMS29F040 的片选信号无效,故此时单片机可对上述 I/O 接口芯片执行控制。并行输出口 74HC377(1)的输出 Q0,Q1,Q2 作为 Flash 存储器 TMS29F040 的高 3 位地址 A16,A17,A18,输出 Q3,Q4,Q5 作为采样通道多路模拟开关的通道地址 A,B,C。当单片机对 Flash 存储器 TMS29F040 操作时,先置 P1.7 为“1”,通过并行输出口 74HC377(1)的 Q0,Q1,Q2 输出 Flash 存储器 TMS29F040 的高 3 位地址 A16,A17,A18,然后清 P1.7 为“0”,选中 Flash 存储器 TMS29F040,即可按常规方式进行编程。程序如下:

```
SETB    P1.7
MOV     A, #add           ; # add 为 Flash 存储器 TMS29F040 的高 3 位地址
MOV     DPTR, #0 FDFDH   ; FDFDH 为并行输出口 74HC377(1)的地址
MOVX    @DPTR,A
CLR     P1.7
```

4. 通信接口

由于记录仪用于检测和记录设备的运行状态数据,所以具有“黑匣子”功能的特点。为了

简化人机界面,仅保留必要的通道状态 LED 显示,记录仪各种初始化及记录数据抄取均通过串口通信完成。为了和管理计算机实现正常通信,采用 MAXIM 公司的 MAX232 单+5 V 电源 RS-232 接口芯片,它能够提供与 PC 通信的标准 RS-232 电平。环保设备运行记录仪的通信分近程和远程两种方式,近程通信方式直接通过标准 RS-232 口进行,而远程通信方式则通过内置远程模块即 MODEM 及其控制板来实现。

9.4.3 记录仪工作软件

环保设备运行记录仪需要对所处理的污染物质进行检测、记录和对设备自身运行状态进行检测、记录。所以,其工作软件由采样滤波模块、数据记录模块、通信管理模块、数据管理模块等主要功能模块组成。下面对各个功能模块进行简要的介绍。

1. 采样滤波模块

这个模块根据设定的采样周期,将 8 个通道的模拟信号经过 A/D 转换器转换成数字信号后输入到数据缓冲区内,并由平均值滤波算法滤除信号中的工频干扰,以提高记录数据的准确性。

在执行平均值滤波时,其算法为

$$y_n = \frac{1}{N} \sum_{i=1}^N x_i$$

其中: y_n 为第 n 次滤波器输出;

x_i 为第 i 次采样值;

N 为采样次数。

2. 数据记录模块

记录仪的 8 个通道均可自由设定为开关量、累积量或即时量通道;程序首先判断各通道的记录类型,然后根据通道类型对采样数据做相应处理以产生对应的运行记录。图 9-22 给出了该模块的流程图。

数据记录模块在判别了通道类型之后,如果识别了通道传送的是开关量,那么进而判别采样值是否高于下限;当得知采样值高于下限,说明环保设备处于开机状态,并记录开机时间;接着判别采样值是否高于上限。如果高于上限,则设置超限报警标志;如果低于下限,则清除超限报警标志。当判别出采样值低于下限时,说明环保设备处于关机状态;接着判别设备是否设置了开机标志,如果已设置了开机标志,则说明原来设备处于开机状态,故这时应清除开机标志,并记录保存关机时间。

在判别出通道类型为累积量时,则把采样量的原有的累加值相加,然后存储入缓冲区;接着判别是否到达记录周期,如果到达记录周期则存储当时记录周期的开始时间及累积值;随后,把下一记录周期的开始时间缓冲区、累积值缓冲区清 0。

在判别出通道类型为瞬时值时,马上判别记录周期是否到达。如果是,则把采样值及当时的时间存入存储器中保存。

为了防止记录仪掉电时丢失数据,记录数据缓冲区开设在 DS12887 非挥发型 RAM 单元中,上电时程序对记录数据缓冲区状态进行分析并处理,以保证数据的完整。

3. 通信管理模块

管理计算机对记录仪的所有操作都是通过 RS-232 串口通信来实现的。通信管理模块完成操作命令的接收、纠错、密码及序列号识别、命令解释及散转执行、数据发送等功能。

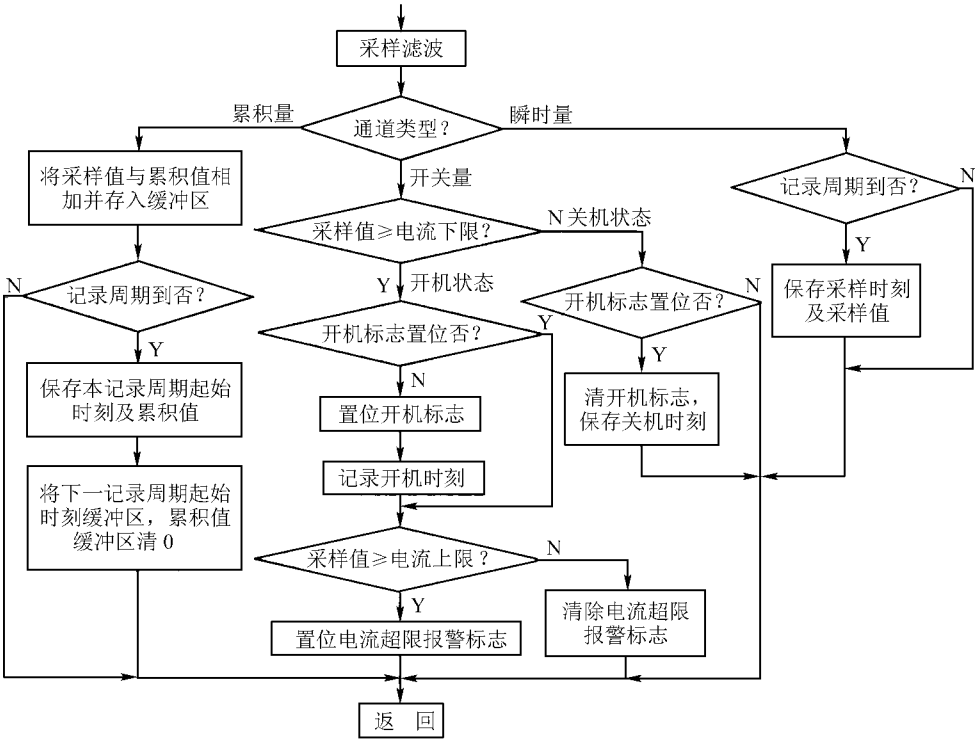


图 9-22 数据记录模块流程图

4. 数据库管理模块

该模块用于运行数据的查询、存储管理、数据删除等功能。读码器或管理计算机通过 RS-232 串行口以记录通道号和记录时间范围,有关关键字读取保存在记录仪 Flash 存储器中,运行数据管理模块根据上述关键字进行检索,将符合条件的记录送往发送缓冲区,由通信管理模块通过串行口发送。

记录仪产生的运行记录交由数据库管理模块编程并写入 Flash 存储器。编程前先判断存储器的剩余存储容量大小。若剩余空间的字节数不能写下一条完整记录时,管理模块将整理存储区以获取空余的存储空间,以便存储有关记录数据。读码器或管理计算机可查询记录仪剩余存储容量百分比。

由于记录仪在有污染和工业干扰的情况下运行,所以工作环境较差,容易受到电磁干扰。为了提高记录仪运行的可靠性,除了在软件上采用指令冗余和软件陷阱等措施外,硬件上还采用了“看门狗”MAX813L 进行定时运行监视,在程序盲飞到一个不可知的死循环中,而冗余指令和软件陷阱均无能为力而造成“死机”时,MAX813L 会在设定的时间内产生溢出,强制系统复位。

9.5 IC 卡电度表

随着现代化进程的发展,城市、农村居民用电量越来越大,用电家庭越来越广泛,这样,供电和收费问题就成了电力供电系统的一项极其繁重的任务。为了解决这一问题,首先就得从抄表着手进行改进,克服过去手工抄表所导致的枯燥无味、重复吃力、走家串户,令人烦累的弊

病。人们对供电的电度表进行了各种改进,研制了无线读表、通过接口连接的手提便携计算机读表等装置。但是,这些装置都存在各种问题,例如无线读表存在干扰,读数有时不准;手提便携计算机读表需人工接插通信接口,接口有时接触不良,读表过程和人工抄表类同,只是不用看表的读数和手工抄录而已,劳动强度仍然很大。一些地方采用了计算机网络自动读表抄表,大大地简化了抄表过程,无需人工到现场进行劳动操作,这是发展的方向,也是实现自动抄表和供电收费的最有效的方法。但是,这种先进的方式并非任何地方都有条件实现。在那些尚未用计算机网络进行供电监控和收费的地方,需要用一种符合现实而又能减少劳动强度的有效收费方法。IC 卡电度表就是这样一种可以监视用电情况,简化抄表手续,收费方便的仪器。

IC 卡电度表在用电监视和收费方面所具有的功能和优点有如下几点。

- (1) 用户只需到专门的供电部门购买 IC 卡,插入电度表,即可实现计费供电。
- (2) 可适用于不同电价的计费。
- (3) 能显示用电量。
- (4) 在购电费为零时能控制电路断电,停止供电。

在这一节中,介绍 IC 卡电度表的逻辑控制电路及其工作原理。这种电度表采用复位中断计数的方法,从而十分有效地克服了各种外界不良干扰的问题。即使在外界存在微波炉、冲击钻、电磁灶、木工机床、电吹风等会发出各种脉冲或谐波干扰的家用电器或工具条件下都能正确无误地进行电度数的计量,所以,有很高的可靠性和适应性,能满足电力系统供电计费的需要。

9.5.1 IC 卡电度表的逻辑电路

IC 卡电度表的控制逻辑电路如图 9-23 所示。它由 AT89C2051 和相关的电路组合而成,电路结构十分简单。

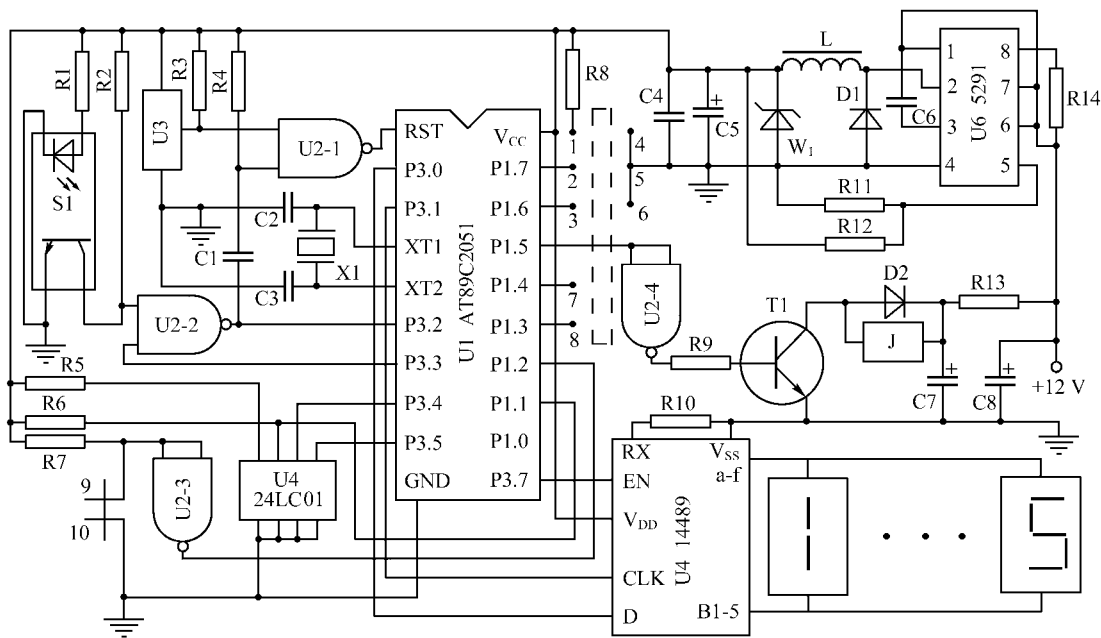


图 9-23 IC 卡电度表控制逻辑电路原理

除了单片机 AT89C2051 之外,还有上电复位电路 HT7044A,光电耦合器 S1,外部 EEPROM24LC01,电源芯片 5291,多功能 LED 显示驱动器 14489,继电器和驱动晶体管 T1 等。

1. 电度计数输入电路

电度计数输入电路由光电耦合器 S1 和与非型施密特电路 U2-2 组成。IC 卡电度表不损坏原来机械电度表的结构,而只是由反射式光电耦合器 S1 对机械电度表的电度计数转盘上的不反光标志进行检测,检测结果再由 U2-2 的与非型施密特电路进行整形,然后送入到单片机 AT89C2051 内部。电度计数输入电路还受到单片机的控制,只有在 AT89C2051 的 P3.3 端口输出为高电平时,才允许将电度计数输入电路的信号送到单片机的输入端口 P3.2 中去。

光电耦合器 S1 的初级发光二极管长期通电发光,当电度表的转盘的不反光标志转到发光二极管所在位置时,则不产生反光到其光敏三极管上,所以其光敏三极管会截止,则与非型施密特电路 U2-2 就会在 P3.3=1 时,输出低电平到单片机 P3.2 端口。当电度表的转盘不反光标志偏离发光二极管所在位置,则有光线反射给其光敏三极管,所以光敏三极管导通,则 U2-2 就会输出高电平到单片机 AT89C2051 的 P3.2 端口。

在单片机 AT89C2051 的端口 P3.3=0 时,封锁来自光电耦合器的信号,并强行置 U2-2 输出高电平,则单片机的端口 P3.2 输入的是高电平。

电度表计数输入电路在本质上有两路,一路将 U2-2 的输出送到 U2-1 一个输入端。当 P3.3=1 时,允许单片机接收电度表计数信号。如果这时光电耦合器 S1 检测到不反光标志时,则其输出高电平,从而使 U2-2 输出低电平。这个低电平会通过 C1 到 U2-1 的输入端产生一个负跳变。这个负跳变过 U2-1 后产生一个正跳变送入 AT89C2051 的复位端 RST,从而使单片机复位。单片机每产生一次复位,则电表用电量计数累加一次,并存入外部 EEPROM24LC01 中。

另外一路在 U2-2 输出低电平时直接送到单片机的端口 P3.2 中去。这个低电平在不反光标志和光电耦合器相遇的期间一直会保持不变,故单片机在复位后进行用电量计数时,还对端口 P3.2 进行查询,当查询到 P3.2 为低电平时,证明的确是检测到不反光标志,从而确认复位计数脉冲有效。

这样进行处理,使得电表的电度计数有极高的可靠性,保证了电表信息的高度可信度。

2. 通电复位电路

通电复位电路由集成电路 U3 和电阻 R3、与非门 U2-1 组成。

U3 是集成电路 MAX707/708,它是通电、断电监视电路,在通电和断电时可产生复位信号输出。

MAX707 对电源电压的监视门槛为 4.65 V,MAX708 对电源电压的监视门槛为 4.40 V。它们的结构和引脚如图 9-24 所示。在图中,标出的监视门槛电压为 4.65 V,是 MAX707 的门槛,而 MAX708 门槛电压没有在图中标出。

MAX707/708 内部由上电比较器,复位发生器,反相器以及失电比较器组成。它们的引脚及功能分别说明如下。

引脚 1: $\overline{\text{MR}}$,手动复位输入。当 $\overline{\text{MR}}$ 输入信号低于 0.8 V 时,产生复位脉冲信号输出。当 $\overline{\text{MR}}$ 输入低电平时,会有 250 μA 的内部拉出电流,该拉出电流可以驱动连接在 $\overline{\text{MR}}$ 端的 TTL 或 CMOS 逻辑门,也可以由开关短路到地。一般在 $\overline{\text{MR}}$ 输入的手动复位信号是由开关或逻辑门产生,这时,手动开关应接到地,或逻辑门应输出低电平。所以,MAX707/708 的内部拉出

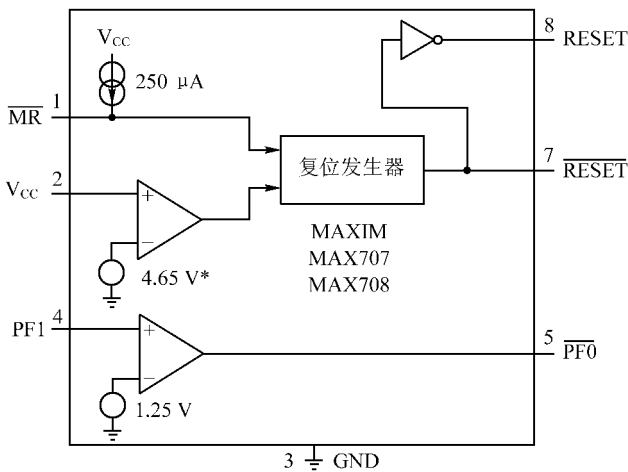


图 9-24 MAX707/708 的结构引脚

电流会作为外部逻辑门的灌入电流,或开关短路到地的电流。

引脚 2: V_{CC} , +5 V 电源。

引脚 3: GND, 信号地。

引脚 4: PFI, 电源电压下降监视输入端。当 PFI 端输入低于 1.25 V 时, 就会使 $\overline{PFO}$ 端输出低电平。如果 PFI 端不用时, 把其接到 GND 或 V_{CC} 端。

引脚 5: $\overline{PFO}$, 电源电压下降监视输出端。当 PFI 端输入低于 1.25 V 时, 就会使 $\overline{PFO}$ 端输出低电平, 同时接收灌入电流, 其他状态 $\overline{PFO}$ 输出高电平。

引脚 6: 空脚, 无用。

引脚 7: $\overline{RESET}$, 低电平复位输出脉冲端, 脉冲宽度为 200 ms。如果电源 V_{CC} 低于复位门槛 4.65 V 时, 则保持输出低电平而不是脉冲。要注意, MAX707 的复位门槛为 4.65 V, 但 MAX708 的复位门槛为 4.40 V, 两者有所不同。在接通电源 V_{CC} 时, 由于 V_{CC} 从 0→5 V, 故会产生 200 ms 的复位脉冲输出。在 $\overline{MR}$ 有低电平脉冲输入时也会产生 200 ms 复位脉冲输出。

引脚 8: RESET, 高电平复位输出脉冲端。这个信号是 $\overline{RESET}$ 的反相信号, 由 $\overline{RESET}$ 通过一个内部的反相器产生。

复位电路的复位功能和过程有下列几种。

第一种: 通电复位。

在接上电源 V_{CC} 使 MAX707/708 通电时, 电源 V_{CC} 从 0→5 V, 这时有一个过渡过程。在过渡过程中的一部分时间中, 存在 V_{CC} 小于 4.65 V 的情况, 则上电比较器就会输出低电平送到复位发生器中, 从而产生 200 ms 的复位脉冲输出。

第二种: 手动复位。

在 $\overline{MR}$ 端接一个按键, 按键另一端接地, 则按键按下时, 会产生一个低电平脉冲送到复位发生器中去, 使其产生 200 ms 复位脉冲输出。

第三种: 电源下降复位。

把电源和电源下降输入端 PFI 相连, 当电源下降, 并且有 PFI 电平小于 1.25 V 时, 就在 $\overline{PFO}$ 端输出低电平。如果把 $\overline{PFO}$ 端和 $\overline{MR}$ 相连, 则使 $\overline{PFO}$ 输出的低电平加到 $\overline{MR}$ 端而送入到复

位发生器中,因而使复位发生器产生 200 ms 复位脉冲信号输出。

在 IC 卡电度表中,只用到通电复位功能,所以 MAX707/708 的 V_{CC} 端接到电源上,而 GND 端接到系统的地线上,其复位脉冲 $\overline{RESET}$ 端则接到与非门 U2-1 的一个输入端。因此,当通电时,MAX707/708 所产生的 $\overline{RESET}$ 端输出的低电平复位脉冲,由 U2-1 与非门反相之后成为高电平复位脉冲送到单片机的复位端 RESET,单片机由此产生复位操作。

3. 外部 EEPROM

在 IC 卡电度表中,通常采用外部 EEPROM 存储有关参数,包括用户购电量等。外部 EEPROM 是型号为 24LC01 的存储器,它的容量为 1 K 位,在内部以 128×8 方式存储。存取采用串行方式,串行接口用 2 线方式,其中 SCL 为串行时钟,SDA 为串行数据线。SDA 用于双向数据串行传送,这条引脚是漏极开路输出的,故需要外部上拉电阻。24LC01 的读写过程和 24C01A 类似。

4. 稳压电源

稳压电源由开关电源芯片 5291 和电感 L、电容 C4、C5 等滤波环节组成。开关电源有效率高、功耗低、工作电压范围宽等优点。

5. 用户供电控制电路

用户供电控制电路用于控制用户用电以及停止供电。在用户购电量尚未用完时,该电路控制供电开关向用户供电;当用户购电量已用完毕时,则该电路控制供电开关断开,停止向用户供电。

用户供电控制电路由与非门 U2-4、电阻 R9、晶体管 T1、继电器 J、阻流电阻 R13、续流二极管 D2 组成。

当用户的购电量用完时,单片机从 P1.5 输出低电平,由 U2-4 反相后成高电平,经电阻 R9 加到晶体管 T1 的基极,使 T1 导通,则继电器 J 线包有电流流过,继电器吸合,使继电器的常闭触头断开,常开触头接通,从而使供电开关开路,停止向用户供电。

当用户的购电量尚未用完时,单片机在 P1.5 输出高电平,使晶体管截止,继电器 J 线包无电流流过,继电器不吸合,常闭触头接通,常开触头断开,供电开关闭合向用户供电。

6. 显示电路

显示电路由多功能 LED 显示驱动器 MC14489 和多个 LED 构成。MC14489 是 7 段 LED 直接驱动芯片,它可以驱动 5 个 7 段 LED。MC14489 可以用串行接口和单片机进行通信。MC14489 对 LED 的驱动是以共阴极方式进行的,故显示器件应采用共阴极 7 段 LED。

MC14489 的封装引脚如图 9-25 所示,其中有数字接口、显示接口和电源三类。各引脚的含义说明如下。

数字接口类引脚有 4 条分别如下。

DATA IN: 引脚 12, 串行数据输入端。数据输入时先输入最高有效位 MSB, 每一位都是在时钟 CLOCK 的上升沿时移入 MC14489。在芯片内部有 2 个十分重要的寄存器,一个是配置寄存器,另一个是显示寄存器。

配置寄存器有 8 位。当串行连续输入 8 位二进制位时,这个 8 位数会存入配置寄存器内。

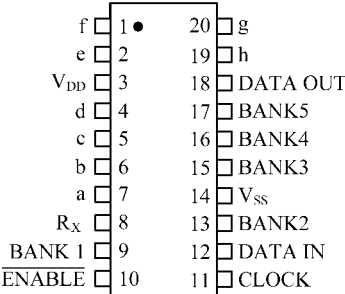


图 9-25 MC14489 封装引脚

显示寄存器有 24 位。当连续串行输入 24 位二进制位时,则会存入显示寄存器。

数据存入配置寄存器或显示寄存器只是由输入串行数据的位数决定,而无需地址或其他信息。

CLOCK:引脚 11,串行时钟输入端。在串行时钟信号的低变高的上升沿时把串行输入数据移入 MC14489 的内部移位寄存器。在串行时钟信号的高变低的下降沿时,则把内部移位寄存器的数据从数据输出端 DATA OUT 移位输出。在 MC14489 内部含有一个 24 位的移位寄存器,它专门用于寄存输入的数据。当输入的串行时钟有 8 个周期时,则输入 8 位串行数据,并从移位寄存器再送入到配置寄存器;当输入的串行时钟有 24 个周期时,则输入 24 位串行数据,并从移位寄存器再送入显示寄存器。

ENABLE:引脚 10,允许 MC14489 在串行总线中使用的控制信号。这个信号用于 MC14489 和其他外围芯片在串行总线中相连接时,对 MC14489 进行选择应用。ENABLE 为低电平时,选中 MC14489 芯片占用串行总线。

DATA OUT:引脚 18,串行数据输出端。当串行时钟从高变低的下降沿时,MC14489 内部的移位寄存器内容从 DATA OUT 移位输出。

显示接口类引脚有 14 条,分别如下。

R_X :引脚 8,外部电流调节电阻连接端。电阻应在 R_X 和地端 V_{SS} 之间连接。外部电阻用于确定 LED 驱动段号 a~h 的峰值电流。在 R_X 端所连接到地的电阻不能小于 700 Ω 。所连接的电阻及电流的关系曲线如图 9-26 所示。

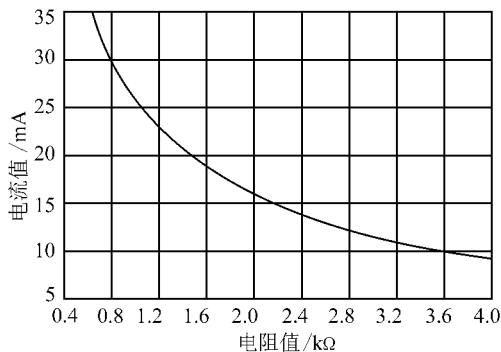


图 9-26 R_X 与地间相连电阻及电流关系曲线

a~h:引脚 1,2,4~17,19,20,LED 显示器的 a~h 段电流驱动端。LED 显示器应是其阴极显示器,其中 h 是显示器的小数点。当 h 不用时,h 端应开路。

当 a~h 不用于驱动 7 段 LED 显示器,而用于单独的发光二极管时,则其中的 e~g 不用,这时 a,b,c,d 分别可单独驱动 5 个发光二极管,h 也可以单独驱动 5 个发光二极管。

BANK1~5:引脚 9,13,15,16,17,显示器选择开关。这 5 条输出引脚是低阻抗对地开关,电流控制能力高达 320 mA。这些引脚应直接连接到共阴极的 7 段 LED 的阴极或发光二极管的阴极。显示以 1 kHz 速率扫描进行。

电源类引脚有 2 条,它们分别有:

V_{DD} :引脚 3,正电源,一般为 3~6 V,典型值为 5 V。为了有效驱动 7 段 LED 显示器, V_{DD} 的电压应取 4.5~6 V。

V_{SS} : 引脚 14, 地端。

MC14489 的典型应用如图 9-27 所示。图中表示的是单片机对 MC14489 的控制以及 MC14489 对 7 段 LED 显示器的驱动情况。其中,MCU/MPU 是单片机或微处理器,它通过串行接口和 MC14489 的 DATA IN,CLOCK 以及 $\overline{ENABLE}$ 连接,同时,也可以接收 MC14489 的数据输出端 DATA OUT 的信号数据。从图中看出:MC14489 可以驱动 5 个 7 段的 LED 显示器。在 MC14489 内部,配置寄存器和显示寄存器的有关功能说明如下。

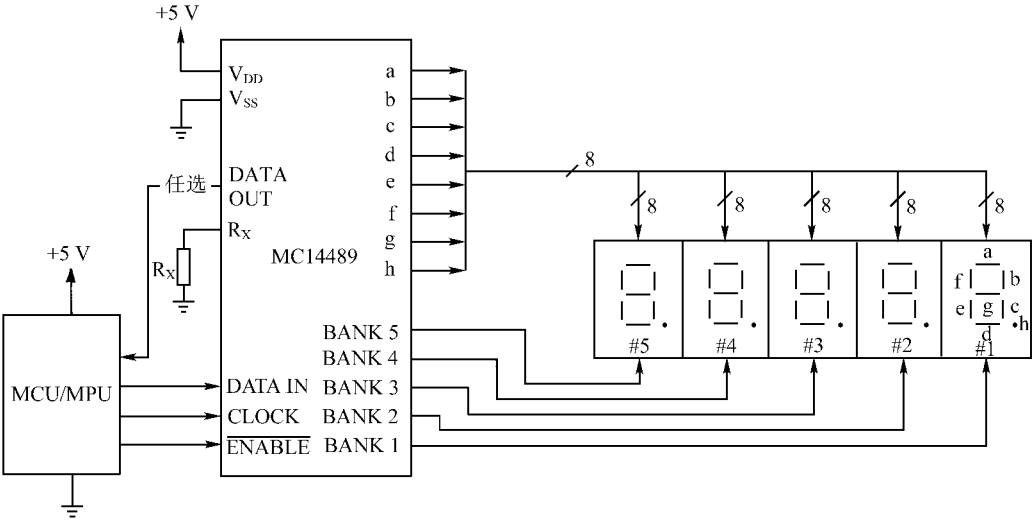


图 9-27 MC14489 的典型应用

配置寄存器的作用和功能用于确定显示的方式。显示方式有低电源方式和一般方式,在一般方式中又分成十六进制译码显示、特殊译码显示和不译码显示 3 种情况。配置寄存器的数据输入及各位意义如图 9-28 所示。

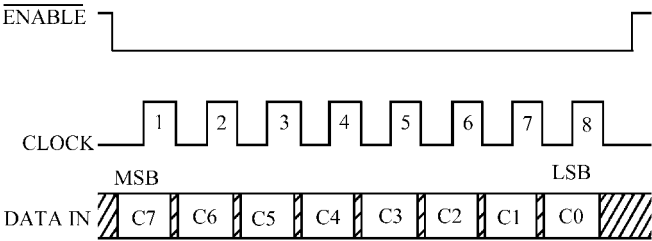


图 9-28 配置寄存器及其各位输入

配置寄存器各位分别用 C7~C0 表示,其中 C7 为最高有效位,当数据串行输入 MC14489 时,最高有效位 MSB 先输入。在数据输入时, $\overline{ENABLE}$ 先变成低电平,串行数据在时钟的作用下,先为最高有效位 MSB 输入,最后为最低有效位 LSB 输入,送入移位寄存器中。在 8 位数据移入移位寄存器之后, $\overline{ENABLE}$ 信号变为高电平。由于输入的是 8 位数据,故这个数据被存入配置寄存器中。配置寄存器的各位意义如下。

C0:C0=0,MC14489 以低电源工作方式工作,通电复位时会强制令 C0=0;C0=1 时,MC14489 以一般方式工作。在低电源工作方式时 MC14489 处于静态状态,并使显示器熄灭。

在一般工作方式中,还有十六进制译码,特殊译码和不译码 3 种情况,这些情况由 C1~C7 决定。

C1:用于控制 BANK1 端对应的显示数据。C1 = 0 时, BANK1 对应的显示数据进行十六进制译码显示;C1 = 1 时,取决于 C6 的状态。

C2:用于控制 BANK2 端对应的显示数据。C2 = 0 时, BANK2 对应的数据以十六进制译码显示;C2 = 1 时,取决于 C6 的状态。

C3:用于控制 BANK3 端对应的显示数据。C3 = 0 时, BANK3 对应的数据以十六进制译码显示;C3 = 1 时,取决于 C6 的状态。

C4:用于控制 BANK4 端对应的显示数据。C4 = 0 时,以十六进制译码显示;C4 = 1 时,取决于 C7 的状态。

C5:用于控制 BANK5 端对应的显示数据。C5 = 0 时,以十六进制译码显示;C5 = 1 时,取决于 C7 的状态。

C6:在 C1~C3 为“1”时,决定 BANK1~BANK3 对应的数据的译码方式。C6 = 0 时,不进行译码显示;C6 = 1 时,实行特殊译码显示。

C7:在 C4,C5 为“1”时,决定 BANK4,BANK5 的对应数据的译码显示方式。C7 = 0 时,不进行译码显示;C7 = 1 时,实行特殊译码显示。

显示寄存器用于存放显示数据和显示控制码。显示寄存器有 24 位,分别用 D23~D0 表示,其中,D23 是最高有效位 MSB。显示寄存器的 24 位数据输入和配置寄存器 8 位数据输入类同,在 24 位数据输入移位寄存器之后,则 $\overline{\text{ENABLE}}$ 又变为高电平,24 位数据从移位寄存器送入显示寄存器中。显示寄存器数据的输入以及各位的含义如图 9-29 所示。

在显示寄存器中,含 3 个字节的数据,并且分成 6 组,每组数据 4 位。低 5 组是显示数据,最高 1 组是显示控制码。各组数据及其含义如下。

D0~D3:对应 BANK1 端的显示数据组 1,其中 D3 是最高有效位。

D4~D7:对应 BANK2 端的显示数据组 2,其中 D7 是最高有效位。

D8~D11:对应 BANK3 端的显示数据组 3,其中 D11 是最高有效位。

D12~D15:对应 BANK4 端的显示数据组 4,其中 D15 是最高有效位。

D16~D19:对应 BANK5 端的显示数据组 5,其中 D19 是最高有效位。

D20~D23:这 4 位是显示控制码。其中 D23 是亮灭控制位,D23 = 0,熄灭 LED;D23 = 1,

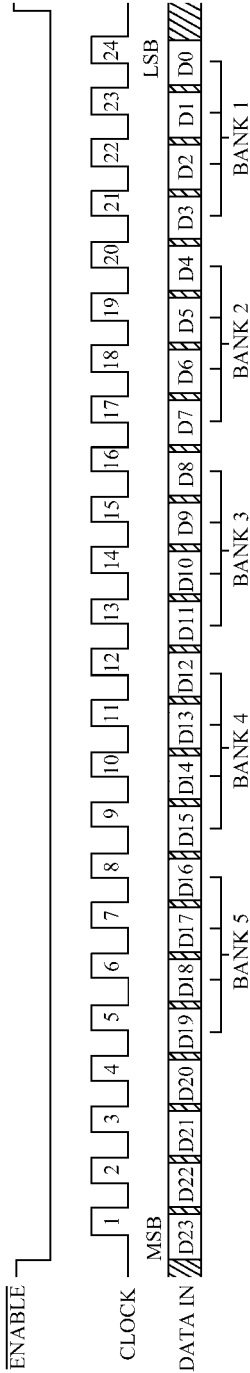


图 9-29 显示寄存器的数据输入及意义

开亮 LED。D22~D20 用于控制各组显示数据的显示,其控制作用如表 9-4 所列。

表 9-4 D22,D21,D20 的含义

D22	D21	D20	含 义
0	0	0	所有 1 输出无效
0	0	1	BANK1 中的数据为 1 输出有效
0	1	0	BANK2 中的数据为 1 输出有效
0	1	1	BANK3 中的数据为 1 输出有效
1	0	0	BANK4 中的数据为 1 输出有效
1	0	1	BANK5 中的数据为 1 输出有效
1	1	0	BANK1 和 BANK2 中的数据为 1 输出有效
1	1	1	BANK1~5 的数据为 1 输出有效

在配置寄存器和显示寄存器的两者共同作用下,MC14489 所执行的 3 种显示情况,即十六进制译码显示,特殊显示和不译码显示的有关代码及显示情况如表 9-5 所列。

表 9-5 MC14489 的 3 种显示情况

显示数据组的值		7 段显示符号		不译码 (C1~C7)			
十六进制	二进制 MSB LSB	十六进制译码 (C1~C5)	特殊译码 (C1~C7)	d	c	b	a
\$0	L L L L	0					
\$1	L L L H	1	c				on
\$2	L L H L	2	H			on	
\$3	L L H H	3	h			on	on
\$4	L H L L	4	J		on		
\$5	L H L H	5	L		on		on
\$6	L H H L	6	n		on	on	
\$7	L H H H	7	o		on	on	on
\$8	H L L L	8	P	on			
\$9	H L L H	9	r	on			on
\$A	H L H L	R	U	on		on	
\$B	H L H H	b	u	on		on	on
\$C	H H L L	C	y	on	on		
\$D	H H L H	d	-	on	on		on
\$E	H H H L	E	=	on	on	on	
\$F	H H H H	F	o	on	on	on	on

在不译码显示时,e,f,g 无用并输出为“0”。h 的输出和译码不受影响,保持原态。不译码显示主要用于对显示灯的独立控制;对带符号的半数字显示;信号符的控制,例如 AM,PM,KV,Hg,UHF,mm 等符号的显示。

7. IC 卡输入电路

IC 卡输入电路由单片机 AT89C2051 的 P1.3, P1.4, P1.6, P1.7 和电阻 R8 组成。IC 卡的接口有 8 条引脚,其中 4,5,6 这 3 条引脚被短路;引脚 1 通过 R8 接电源+5 V;引脚 2 接单片机的 P1.7;引脚 3 接 P1.6;引脚 7 接 P1.4;引脚 8 接 P1.3。IC 卡是 ATMEL 公司的 AT88SC102 加密卡。这种 IC 卡存储容量为 1 K 位,擦写时间为 1 ms,它是带逻辑加密位的 EEPROM,使用温度为-25~+70 ℃,保存信息时间可达 100 年,擦写次数可达 10 万次。在写入时,最大电流为 4 mA;读出时,最大电流为 3 mA。IC 卡内记录用户档案以及购电量、总用电量等有关信息。

9.5.2 IC 卡电度表的控制软件

在 IC 卡电度表使用时,要先把 IC 卡进行初始化。并把和用户对应相关的信息存入到 IC 卡中,然后,把 IC 卡 and 对应电度表配套使用。初用时,将有关信息送入电度表内 EEPROM 中存储,以后用卡进行校核实现一表一卡。购电时,将卡插入售电微机,根据卡上信息,自动调出用户档案。购电后,把 IC 卡插入电度表,由表执行读卡操作。用户购电量输入电度表内 EEPROM 存储后,自动消除卡上购电量信息,并提示用户拔出 IC 卡。为了生产使用方便,系统中配有专用校验卡,主要有脉冲校验卡。插入脉冲校验卡后,LED 数码管显示转盘转数,根据 LED 数码管显示值和目测转盘色标转数,可以校验光电传感器及系统工作是否正常。系统中还设有 0.3 kW·h 运行实验卡及清 0 卡等,用于检测断电系统是否正常。与校验卡配合,表内设有相应软件,用于生产过程检验。在电价变化后,用于更改电价卡,修改表内存储电量。此卡插入后可将表内原存储电量,按新旧价比率折算为相应的存储电量。

IC 卡电度表的控制软件框图如图 9-30 所示。它主要由 5 个部分组成,即 IC 卡信息输入程序段、电价修改程序段、校验程序段、上电程序段、电量计量程序段。

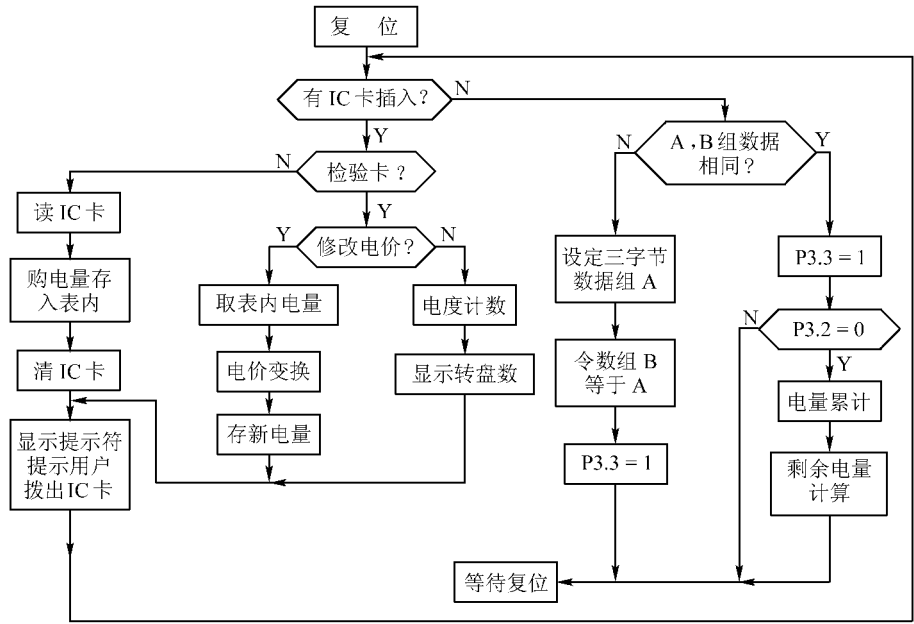


图 9-30 电度表控制软件框图

1. IC 卡信息输入程序段

控制软件对 IC 卡进行校验,如果检测到有 IC 卡存在时,则读 IC 卡的信息。IC 卡中含有和电度表一一对应的用户信息,如果对应信息不对,则说明该 IC 卡不是和电度表配对的,则其内部的信息电度不接受。如果 IC 卡和电度表配对,则 IC 卡的内部信息被电度表所接受。IC 卡中最重要的信息是购电量,单片机把其中的购电量读出之后,存入电度表的 EEPROM 存储器 24LC01 中。为了保险起见,对存入的内容执行校验,无误后,则由程序自动清除 IC 卡中的购电量,同时在 LED 数字显示器中显示相关的提示符,提示用户可以取走 IC 卡。在用户未取走 IC 卡之前,IC 卡一切内容不再被读取。同时,若用户误把 IC 卡插入电度表中,如果程序检查到有 IC 卡插入,程序执行的工作是把电度表内 EEPROM 的电量和 IC 卡中的购电量相加后再存入 EEPROM 中,故不会影响到购电量存入的正确性。

2. 电度表计量正确性校验程序段

校验程序段用于对数字检测的结果和电度表转盘的转数进行对比校正。由于电度表是以转盘的转数来计算用户的用电量的,而 IC 卡电度表也是以转盘的转数为采样对象而进行用电量计算的,这两者必须是一致的。为了校验两者的一致性,采用专门的校验卡。当检测到插入电度表的是校验卡时,则转入校验程序段,单片机对电度表的转盘的转数实行采样,并把实际转数在显示器上显示出来。这样,校验人员就可以把目测的转盘转数和显示转数进行比较,两者一样,则说明通过校验;否则,要对电度表的控制电路进行维修,从而保证 IC 卡电度表工作的正确性及可靠性。

3. 电价修改程序段

在电度表内的 EEPROM 中,存储了用户的有关信息以及用电的有关信息。这些用电信息包括已用电量、剩余电量以及电价。当电价修改之后,可以用更改电价卡插入电度表中,进行剩余电量的修正。在更改电价卡中含有新电价及旧电价,在单片机检测到插入的是更改电价卡之后,则接收该卡的电价信息,同时转入电价修改程序。在该程序中,把新旧电价进行处理,从而求出对应于旧电价的剩余电量在采用新电价之后所折算出的新的剩余电量。并把这个新的剩余电量送入电度表内的 EEPROM 中存放。处理完毕,提示取出更改电价卡。

4. 上电程序段

上电程序段是在电度表的控制电路在接通电源时所进行的有关工作的执行程序。这一段程序主要是进行有关初始化工作,其中最关键的是设置三字节为两组的识别码。这两组识别码分为 A 组、B 组,而且两组识别码一样。这段程序和电量计量程序段都是在电度表不插 IC 卡时的程序的主要流程。两者的区别在于一个是上电时执行的,一个是在上电之后执行的。因为,上电时产生复位,而电度表转盘转动时也会产生复位,电量计量程序段就是利用复位进行计数的。所以,用两个程序段的分支利用对 A,B 两组数据进行判别。一般在通电上电时刻,数据存储器内的内容是随机的,A,B 两组的数据几乎不可能相等。所以,当有 A,B 两组数据不等时,说明是上电复位,这时进入上电程序段,并把 A,B 组数据设置为一样,标志已初始化。

5. 电量计量程序段

当 A,B 组数据相等时,进入该程序段。在这个程序段,主要功能是实现对用电量的计量。用电量的计量是通过复位计数进行用电量累计的。每当转盘的不反光标志和光电耦合器相遇时,产生一个负跳变信号到单片机 AT89C2051 的复位端 RST,使单片机产生复位并由程序的

首部开始执行。当程序对数据组 A,B 实施比较,说明这次复位不是上电复位时,为了确认这次复位的真实性,单片机再次令 P3.3 输出高电平,允许光电耦合器的输出信号通过 U2-2 送到 P3.2 端口。由于不反光标志在电度表的转盘上,转动时和光电耦合器有一定的时间遇合,而复位过程很快,故复位后进入电量计量程序段时不反光标志仍然还会对光电耦合器起作用。所以,这时 U2-2 送到 P3.2 端口的电平为低电平。单片机在检测到这个低电平时则进一步确认这次复位是转盘转动复位,故进行用电量计数。在用电量计数完毕后,等待下次复位。

第十章 信息通信领域的应用

信息通信是一个很广泛的领域,在这个领域中涉及到广播电视、计算机网络、电话、传真、对讲机、寻呼机、手机等。单片机在这个领域也有很广泛的应用。在电话通信系统中,程控交换机内部就大量应用单片机,很多电话机、寻呼机、手机、对讲机内部都是由单片机控制的。而且,单片机在信息通信领域有应用越来越多的趋势,应用面也越来越广。在这一章中,介绍 89 系列单片机在通信领域的几个应用例子。

10.1 小范围专用寻呼系统

在各种工作环境中,通常需要专用的寻呼系统,以实现在某种特殊环境下工作人员之间的必要通信,确保工作的正常执行和可靠性。这种专用的寻呼系统一般是在小范围之内使用的,只要满足在有效范围内人员之间的必要信息通信即可,故功率很小,而且传递的信息较少,相关的信息也有特定的意义。这种小范围的寻呼系统有结构简单、使用方便、造价低廉等一系列优点,适合于工业、农业中需要小范围寻呼时使用,也可以作为保安系统用于各种公司及部门。

小范围寻呼因其设计灵活而有很多种,但按其呼叫形式来分,则可分为一呼多和多呼一两种。在这一节中介绍一种能在噪声污染严重的各种环境下实现无线联络的寻呼系统。该系统中各单机均由 AT89C2051 完成基本控制,用程序进行密码识别和数据译码。不用另外附加的译码芯片,便能很方便地实现多位数据显示,各种音频信号也可以通过软件方法在片内生成。该系统的主叫单机和被叫主机的硬件结构十分简单,如果来实现更多功能,无需改动线路,只要修改一下程序即可。下面以多呼一这种类型为例,介绍其原理及主程序。

10.1.1 寻呼系统的组成

多呼一寻呼系统的结构框图如图 10-1 所示。它由主叫单机群和被叫主机组成。主叫单机群由 00~99 共 100 个单机构成,每个单机的结构和功能完全相同。当某一单机想呼叫主机时,按下 S1 键,则该单机即向空中发射无线信号,此信号就会被主机接收。单机发射出的信号

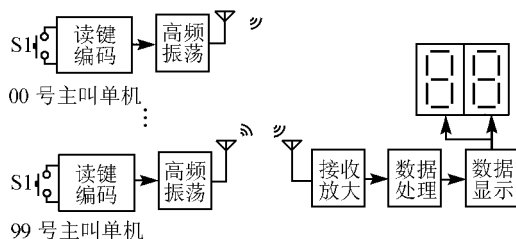


图 10-1 多呼一寻呼系统结构框图

高频载波信号后,通过解调后得到对应的数据信息,单片机把其与内存特征码进行多次对照,如果发现数据信息和系统特征码不一致,便放弃此次接收到的数据,从而把外系统的干扰剔除;如

包括载波及调制在载波之上的两组数据信息。所有的单机发射的载频率是相同的,都能被被叫主机接收。两组数据信息中的一组代表系统特征,另一组则代表单机身份特征。100 个单机都属同一系统,所以代表系统特征的那组数据信息是完全一致的。代表各个单机的身份特征的那组数据信息表示的是各个单机独特的身份,故是各不相同的。被叫主机收到

果一致,说明此信息来自本系统,就将收到的单机身份特征码进行译码处理后送数码管显示出来,并发出声音或振动提示。这样,任何一部主叫单机都可以主动地与被叫主机进行呼叫联络。

10.1.2 主叫单机的结构原理

主叫单机的电原理图如图 10-2 中所示。其中,AT89C2051,C9,R3 和振荡器、滤波环节等组成了单机最小系统。AT89C2051 是美国 ATMEL 公司推出的一种带有 2 KB Flash 可编程可擦除只读存储器(EPROM)的低压高性能 CMOS 8 位微控制器,它有 P3 口和 P1 口共 15 条标准输入输出 I/O 口线,功能强大,完全可以满足主叫单机的工作需要。在多呼一寻呼系统的主叫单机电路中,由于只有高频振荡、按键判断和指示灯这 3 个控制对象,所以,仅用其中 3 条 I/O 口线,其他大部分 I/O 线均闲置悬空,不可接电源正极。在电路中将按键 S1 接在 P1.2 和地之间。电路工作时,先向 P1.2 送高电平,然后再读取 P1.2 的电平并且进行判别。如果 S1 未按下,由于 P1.2 内部电阻的上拉作用,P1.2 电平为高;如果 S1 被按下,P1.2 对地短路,电平为低。单片机 AT89C2051 通过读取 P1.2 的电平的高低,即可判断出 S1 是否按下,从而决定是否发射信息。晶体管 VT2 的基极通过 R2 与 P3.1 端口即 TXD 相连接。当 P3.1 呈低电平时,VT2 截止,其发射极与集电极之间呈高阻态,这样,由晶体管 VT1 和电感 L2、电容 C5 等组成的高频振荡电路就因得不到电源而停止工作;当 P3.1 呈高电平时,VT2 导通,其集电极与发射极间几乎呈短路状态,由 VT1 等组成的高频振荡电路由于接通了电源而起振,电磁波通过天线 Y 辐射出去。显然 AT89C2051 通过程序不断改变 P3.1 电平的高低,就能控制振荡电路的振荡与停止,从而将要发送的信息调制于高频载波之上。发光二极管 VD1 用作电路工作状态指示灯,其正极通过电阻 R4 与电源正极相连,其负极与 P1.3 相连。当 P1.3 为高电平时,VD1 截止不发光;当 P1.3 为低电平时,则相当于将 VD1 的负极与地连接。这时,正电源 V_{cc} 通过 R4,VD1 构成回路流通到地,使 VD1 得到大约 4 mA 的导通电流,于是发光二极管 VD1 发光。在程序的控制下,每隔一定的周期使 P1.3 的电平高低变化一次,发光二极管 VD1 就会以一定的频率闪烁发光,起到指示电路工作状态的作用。

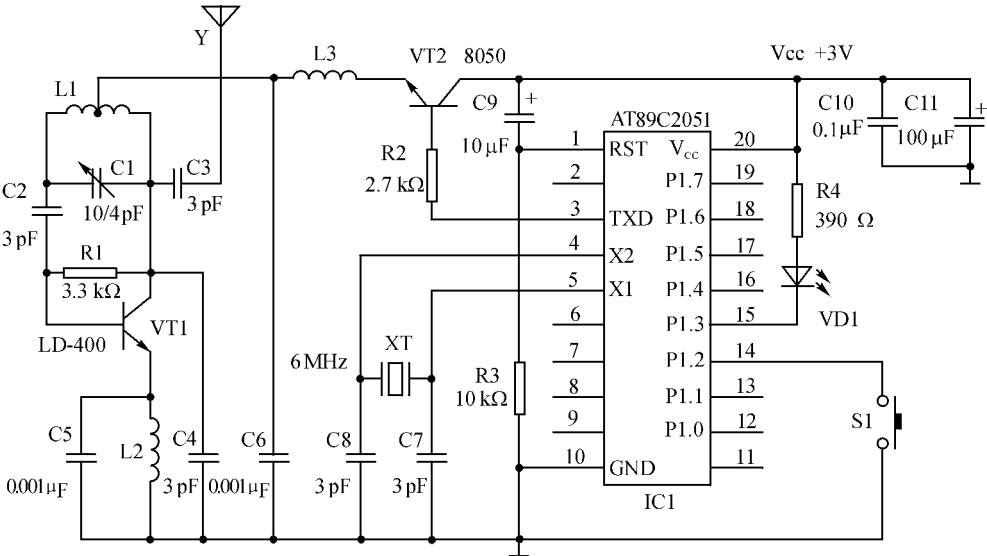


图 10-2 主叫单机电路原理

电容 C9 和电阻 R3 组成了通电复位电路,在接通电源瞬间使单片机复位,从程序的首部开始执行。电容 C10 和 C11 是电源的滤波电容。

10.1.3 被叫主机的结构原理

被叫主机接收显示电原理图如图 10-3 所示。在图中,石英振荡器 XT、电容 C11 和 C12 用来实现 6 MHz 选频振荡,为系统提供准确的时钟源。R9 和 C13 组成了复位电路为单片机提供复位信号,使系统在接通电源时开始运行。由于被叫主机有信息接收、数据显示、声音提示、发光指示、振动提示和按键判断等输入输出控制任务,同时,还必须考虑尽量减小体积,充分利用 AT89C2051 的硬件资源,以便组成一个优化的系统。因此,单片机 AT89C2051 的 15 条口线全部被利用。由于 P3.6 为片内比较器输出端,片外无此引脚,所以 P3 口剩下的 7 条 I/O 线分别与发光二极管数码管 LED1 和 LED2 的字段 a,b,c,d,e,f,g 相连接,用于数码显示。而小数“dp”空闲不用。LED1 和 LED2 是共阴极高亮度发光二极管显示器。由图 10-3 可知,当向 P3 口发送数据 FFH(1111 1111)时,P3 口各 I/O 线电平为高,可以对 LED1 或 LED2 的所有字段提供电流,LED1 或 LED2 就可以显示出“8”字;如果向 P3 口送数据 CFH(1100 1111)时,可对 LED1 或 LED2 相应的 a,b,c,d,g 字段提供电流,LED1 或 LED2 可显示“3”字。各数字相应显示代码在程序中以表格方式给出,只要查表即可得到,不一一叙述。在电路结构中利用 P1 口的两根口线 P1.2 和 P1.3 分别与 LED1 和 LED2 的阴极相连接作为数码显示的位控制。当 P1.2 为低电平时,LED2 可以被选中显示;当 P1.3 为低电平时,LED1 可以被选中显示。这样,控制 P1.2 或 P1.3 电平高低的状态,并将要显示的数字代码从 P3 口送出,就可以分别显示个位和十位数字了。P1.4 与 S1 相接,如果 S1 未被按下,程序将循环进

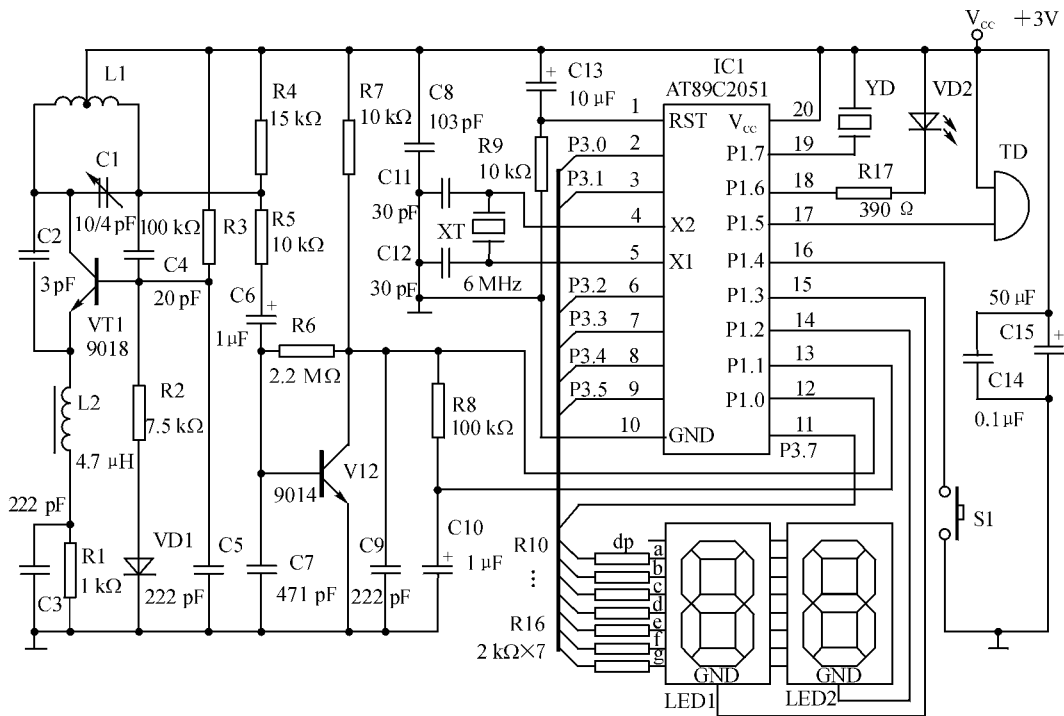


图 10-3 被叫主机电路原理

行警声、振动提示和扫描显示;当 S1 被按下后,程序返回初始化。蜂鸣器 YD、发光管 VD2 和振动器 TD 分别与 P1.7、P1.6 和 P1.5 相连接,通过向这些端口线送出频率不同的脉冲信号,即可实现声音、闪光和振动提示。单片机 AT89C2051 片内比较器的输出端与 P3.6 相连接,但没有外引脚。而比较器的同相输入端和反相输入端分别与 P1 口的 P1.0 和 P1.1 相连接。由于晶体管 VT1、VT2 组成了接收解调电路,由 VT1 以及相关的元件共同构成接收调谐电路,并进行检波解调,则 VT2 对解调后的信号进行放大,并在电阻 R8 上输出相应的信号到 P1.0 和 P1.1,这样,将要整形的脉冲信号加至输入端,P3.6 就可输出被整形后的标准方波,并由单片机处理。

10.1.4 主叫单机的发射主程序流程

发射主程序流程图如图 10-4 所示。结合图 10-2 电路,系统上电后,进行初始化。将系统特征码存于 RAM 的存储单元 50H~51H,而本机身份特征码于 52H 之中,然后再向 P1 口送数据 FFH(1111 1111),向 P3 口送数据 FDH(1111 1101)。此时,P3 口的 P3.1 即引脚 3 TXD 呈低电平,从而会使 VT2 截止,由 VT1 等组成的发射电路没有电源故不工作;P1 口的 P1.3 即引脚 15 输出高电位,所以发光二极管 VD1 截止不发光。然后,系统进入判键和等待状态指示程序。进入此程序后,单片机周期性地使 P1.3 输出的电平按 0→1→0→1→…变化,使 VD1 间歇发光,并不断地读 P1.2 即引脚 14 的数据以检测 P1.2 的电平,当单片机想呼叫主机,按下 S1 键时,则 P1.2 呈低电平。如果单片机检测到 P1.2 呈低电平,经 20 ms 延时消除抖动后仍不变成高电平,则确认发射键 S1 按下,然后调用数据发射子程序,将 50H~52H 的数据通过 P3.1 即 TXD 端送至 VT2 的基极,推动其导通与截止,将信息按约定的编码标准和一定的波特率调制 VT1 发射出去。

在图 10-2 中,晶体管 VT1 及电容 C1、C2 等组成改进型电容三点式振荡电路,频率为 280 MHz。为提高频率稳定性,电感 L1 可印制在线路板上,框状尺寸为 1.2 cm×2.5 cm,线宽为 2 mm。电感线圈 L3 的作用是防止高频信号通过电源等构成回路造成损耗,可采用空芯线圈或集成电感,电感量在 20~47 μ H 之间均可。VT2 的开关状态决定高频振荡电路的发射与截止,也就是实现对高频载波的调制。从上面分析可知,调制信号是通过 P3.1 即 TXD 端输出的,此信号为方波。可以自行制定一种编码标准,利用方波宽度表达数据信息,从而实现数据的发射和接收。在图 10-5 中,给出了所制定的编码协议。在图中脉冲 A 的脉宽 $T_A=200\ \mu$ s,该脉冲代表“0”;脉冲 B 的脉宽 $T_B=400\ \mu$ s,该脉冲代表“1”;脉冲 C 的脉宽 $T_C=600\ \mu$ s,该脉冲代表起始符。在传送数据时,就是传送这 3 种脉种的组合。这些方波靠软件控制在单片机内发生。进入数据发射程序后,首先向寄存器 R2 送立即数 #18H,实际上是送数据 24,即发送数据位数,即 3 个字节共 24 位;再对 P3.1 端口送高电平信号“1”,使 P3.1 呈高电平,经 600 μ s 延时后再对 P3.1 端口清 0,使 P3.1 呈低电平,完成起始符发射。接着检测被发送数据的首位,根据编码约定,采用上述方法改变 P3.1 电平的高低并

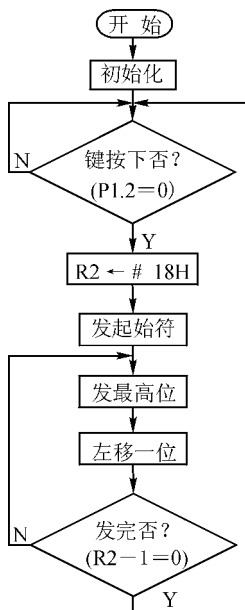


图 10-4 发射程序框图

进行相应延时,发送“0”字符或“1”字符,并将被发送数字串左移一位,然后再将首位发送出去。每发完一位数据,都要执行 R2 内容减 1 的操作,并判断 R2 内容是否为“0”。若不为“0”,说明 3 个字节的数据尚未发完,返回去继续发送;若为“0”,说明已发完 3 个字节,完成了一个数据帧的发送并返回到程序首部去判别 S1 键状态。这样,通过 P3.1 对 VT2 的推动,要发送的信息经调制后就驮载在高频电磁波之上被传送出去。在图 10-6 中,A 为 P3.1 输出值为 #56H 的一字节数据的方波序列,B 为调制后的高频载波。

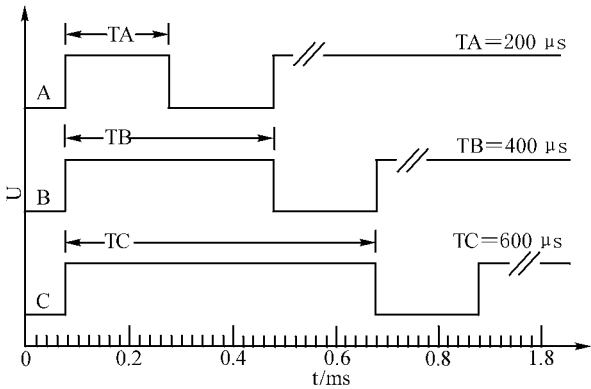


图 10-5 编码标准

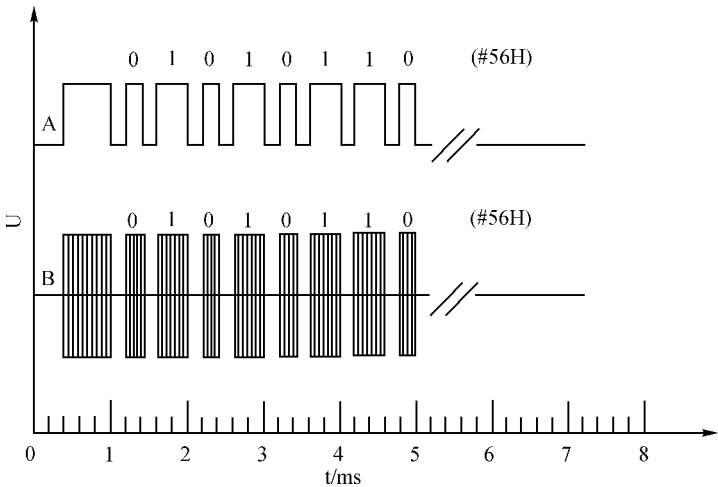


图 10-6 调制与载波

在图 10-6 中,发射的调制信号是 56H,故其二进制信号为“0101 0110”,所以,在调制时,先送出 600 μ s 高电平的起始符,接着是 200 μ s 高电平的“0”符,经过 200 μ s 低电平延时之后是 400 μ s 高电平的“1”符……,直到 8 位数字全部发送完毕为止。当这些调制信号从单片机的 P3.1 端口,即 TXD 端发出时,对晶体管 VT2 进行开关控制,从而控制由 VT1 晶体管等所产生的 280 MHz 振荡信号的发射。在图 10-6 中,被调制的振荡信号如图形 B 所示。

在实际上,发射的是 3 字节的数据,而不是图 10-6 所示的单字节数据。无论发射的是多少字节的数据,都需要起始符。所以,在寻呼系统中,3 字节数据的发射是遵循如下协议标准的。

(1) 发射脉宽为 $600\ \mu\text{s}$ 的起始符,即在 P3.1 端送出 $600\ \mu\text{s}$ 的高电平,然后送出 $200\ \mu\text{s}$ 的低电平。

(2) 发射 2 字节的系统特征码,也就是在 P3.1 端送出和特征码各位相应的电平脉冲。在发送“1”时,先送 $400\ \mu\text{s}$ 高电平,后送 $200\ \mu\text{s}$ 低电平;在发送“0”时,先送 $200\ \mu\text{s}$ 高电平,后送 $200\ \mu\text{s}$ 低电平。

(3) 发射 1 字节的本机身份特征码,发送方法和系统特征码发送方法相同。

10.1.5 被叫主机接收显示主程序流程

接收显示主程序流程如图 10-7 所示。在图 10-3 所示的被叫主机电路原理图中,系统上电后单片机先初始化,也就是装入显示数字代码表,然后向 P3 口送 $\#40\text{H}$,向 P1 口送 $\#\text{FFH}$,将系统特征识别码存于数据存储器 RAM 的存储单元 3AH 和 3BH 之中,并将 $40\text{H}\sim 43\text{H}$ 单元清 0。初始化的结果使发光二极管数码管熄灭,关闭发声压电陶瓷片 HD,使振动器 YD 停振和发光二极管 VD1 不发光,然后进入接收状态。在接收状态,循环调用接收译码子程序并检查 43H 中的接收结果提示信息。

在图 10-3 中接收解调电路主要由晶体管 VT1 和 VT2 以及相关元件等组成。VT1 将高频信号进行接收和检波,VT2 将检出的调制信号进行放大,放大后的信号通过 P1.0 和 P1.1 端口输入到 AT89C2051 的片内比较器进行整形,并在内部从 P3.6 输出。这样,接收程序只要通过对 P3.6 输出的方波按约定的标准进行宽度鉴别,就能实现对主叫单片机发射过来的数据进行正确接收。接收机执行 LCALL 指令进入接收子程序,首先检测 P3.6 输出的电平,如发现 P3.6 输出电平跳高,就进行脉冲宽度鉴别,这时持续读取 P3.6 的输出电平。如果此高电平脉冲宽度远大于或小于 $600\ \mu\text{s}$,即认为是干扰信号,立即返回主程序;如果 P3.6 输出的高电平脉冲

宽度接近 $600\ \mu\text{s}$,再检测 P3.6 跳变为低电平后的持续时间,如果此信号在远小于 $200\ \mu\text{s}$ 时间内跳变为高电平,则此信号亦为干扰信号,系统不予理睬,将其剔除;如果 P3.6 跳变为低电平后在接近 $200\ \mu\text{s}$ 的时间内跳回高电平,则判定为起始符有效。接着以相同的方式来判断随后接收的各位数据。此系统约定 3 个字节的数据为一个传送帧。所以,正确接收到起始符之后再正确接收到 24 位数据,程序才进行数据处理。收到的前两个字节为系统特征码,并将其存于 IC1 片内 RAM 的 40H 和 41H 存储单元之中,收到的第 3 个字节为主叫机身份代码,则存于 42H 之中。然后,取 40H 和 41H 的内容与 3AH 和 3BH 的内容相对照,若不一致,则说明主叫机并非属于本系统,故返回主程序;若一致,说明接收下来的信息来自本系统,就将 42H 存储单元中的数据取出,将其高 4 位和低 4 位经查表译码后分别存于地址为 47H 和 48H 的显示缓冲存储单元中,然后再向存储单元 43H 送立即数 $\#\text{E7H}$,以此作为已正确接到本系统有效数据的提示信息。主程序查看到这个提示信息后,先向 P1.7 送出警声信号,向 P1.5 送出振动信号,然后将 47H 中的内容送 P3 口以供显示,同时执行“CLR P1.3”指令,使 P1.3 输出低电平,这样,主叫单片机身份码的十位数字就在 LED1 上显示出来;然后执行“SETB P1.3”指

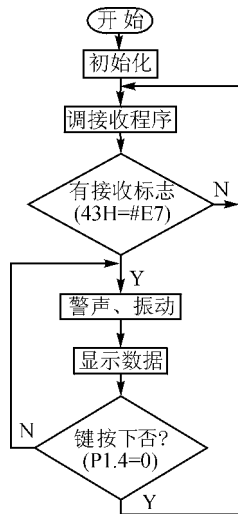


图 10-7 接收显示主程序框图

令熄灭 LED1, 再将地址 48H 中的内容送 P3 口, 执行“CLR P1.2”指令选中 LED2, 将主叫单片机身份码的个位数字显示出来。被叫接收机接收显示范围为 00~99 号, 故共可配套接收 100 个本系统的主叫单片机发来的信息。

这个多呼一系统性能稳定可靠, 实用性和再开发性强, 能够在各种强噪声的环境中实现信息传递。比如纺织车间, 每个工作带一个主叫单机, 车间负责人带一个接收机, 联络起来就十分方便。这种系统也可以用于旅游团体的联络, 学生野外活动联络, 建筑工地的工人联络等场合。

10.2 通信信息的滤波与纠错

在通信系统中, 信息在传递过程中会受到来自系统外部或内部噪声信号的干扰。受干扰的信息在传送中或接收中都会产生错误的结果。这是人们不愿看到的情况。干扰的存在破坏了信息的正确传送, 对通信系统的可靠性造成了极大的威胁。为了克服噪声信号对正确信息的干扰, 在通信系统中采用了大量的抗干扰措施, 其中最常用的方法就是采用滤波器和纠错码。

10.2.1 信息的滤波

滤波的意义就是把信号频谱中的某些成分衰减掉, 或者说是过滤掉。

任何改变信号频谱的网络都称为滤波器。滤波在无线电工程领域和信息处理的连续系统理论中有着极重要的地位。

滤波器一般可以分成两大类型, 即经典滤波器和现代滤波器。

经典滤波器的滤波作用是基于有用成分和无用成分各自占有不同的频带这种假设的基础上的。在经典滤波器把无用成分的频带滤去之后, 得到的只是有用成分的频带, 这样就取得了期望的滤波结果。经典滤波器当然存在一个不足, 就是在有用成分和无用成分的频谱重选时, 其滤波效果就无效。但是, 当噪声和有用信号分布于不同频谱时, 经典滤波器是十分有用的。

现代滤波器研究的是对含有噪声的时间序列数据进行特征估计或信号估计。在现代滤波器的理论中, 信号和噪声都被看作随机信号, 并利用它们的统计特征求取其最佳的估计算法。现代滤波器的估计算法可以用有关硬件结合软件实现。

在通信中, 通常信息的频带是有明确的宽度的, 而往往干扰噪声和信息的频带不同, 故经典滤波器仍有很大的用场。在各种通信、测量系统及仪器中, 常用的模拟滤波器是典型的经典滤波器, 而这些滤波器往往是 RC 组成的滤波网络。众所周知, RC 滤波器很难达到理想的滤波效果, 原因在于 RC 滤波器的截止频率不能线性调节。这里介绍的由 AT89C51 单片机和 MAX292 构成的截止频率可任意设置有源模拟滤波模块, 这种滤波模块的最大优点在于截止频率可由单片机内部的程序进行控制, 设置简单, 滤波效果好, 在系统中使用方便。

单片机控制的滤波模块的关键元件是美国 MAXIM 公司生产的 MAX292 芯片。MAX292 是 CMOS 8 阶(8 个极点)的开关电容式的有源贝塞尔型低通滤波器, 它的特点是可实现线性延迟的相频响应, 并能够保持阶跃输入信号的基本形状, 同时无过冲现象, 减轻了对较高频率分量的衰减, 比较快地取得稳定的响应。其在一 3 dB 处对应的频率即截止频率 f_c 。既可由外部时钟信号控制, 也可以利用片内时钟源加一外接电容所产生的时钟信号进行控制, 截止频率 f_c 的变化范围是 0.1 Hz~25 kHz。其引脚及内部结构框图如图 10-8 所示。

MAX292 主要由一个 8 阶开关电容滤波器、一个时钟产生电路和一个独立的运算放大器

在图 10-9 所示的滤波模块结构中,按键和相关的门电路 74LS00 组成了输入设定电路;液晶显示模块 LCM 作为截止频率显示器;MAX292 是可控的低通滤波芯片;AT89C51 用于整个系统模块的控制。

按键输入电路中有 4 个按键,分别用于工作控制、频率输入。频率输入由位选、位值及确认这 3 个按键实现。位选键用于选择频率数据的位;位值键用于对所选中的位设置数值;确认键用于对所输入的键值即截止频率进行确认。这 3 个按键分别和 AT89C51 的 P17,P15,P14 端口相连接。工作控制由开关键控制,并且和 AT89C51 的 P16 端口相连。当开关键、位选键、位值键和确认键这 4 个键都处于断开状态时,则对应端口的输入为高电平,而中断输入口 $\overline{\text{INTO}}$ 也为高电平,这时没有中断产生。当有一个按键按下时,则该按键两个触点接通,从而在 $\overline{\text{INTO}}$ 口产生一个低电平脉冲,同时在对端口也产相同的低电平脉冲。从 $\overline{\text{INTO}}$ 输入的低电平产生中断请求,单片机响应了之后进入中断处理程序,在中断处理程序中,会对 P14~P17 端口的输入电平进行判别,从而分辨出是哪一个按键按下。

液晶显示模块 LCM 用于显示截止频率 f_0 , 显示范围从 0.1 Hz~25 kHz。显示信号是通过单片机 AT89C51 的 P0 端口送到液晶显示模块 LCM 中去的。

MAX292 的工作状态由单片机 AT89C51 所产生的时钟信号控制。单片机产生的用于控制 MAX292 的脉冲时钟信号从 P10 端口输出,并直接连接到 MAX292 的时钟输入端 CLK 上。只要单片机送入到 MAX292 的时钟信号频率确定,就可以确定 MAX292 滤波的截止频率,故单片机对 MAX292 的控制异常简单。单片机产生的控制 MAX292 的时钟信号频率是根据用户输入并在 LCM 上显示的频率数据,由程序产生的。

AT89C51 是滤波模块的控制核心。在这个滤波模块中,只用到 AT89C51 的部分资源,所以,AT89C51 的功能还有很大的开发余地。在控制中,真正对 MAX292 起作用的只有 P10 端口一条 I/O 引脚。因此,只要把系统作一定的修改,还可以增加其他功能。也可以采用简单的方法进行按键的信号输入,例如,只要采用查询的方法而不是中断的方法进行按键输入,就可以节省集成电路 74LS00。这样,不但减少系统的成本,也减少了系统的体积。

滤波模块的软件主要由键判断、用户输入值计算、根据用户输入值设定 P10 口输出、系统确认运行几部分组成,其软件程序框图如图 10-10 所示。

滤波模块的控制程序较为简单,它首先判别是否有按键按下,如果没有按键按下,则返回等待按键按下;如果有按键按下,则响应中断,把 P14~P17 的输入信号取到单片机内,并进行判别。判别有两种过程:第一种是按键的判别,第二种是数值的位选和位值设定。判别按键是以键码为标志的,当在 P14~P17 中输入的键码为“0111”时,则说明按下的键是位选键;键码为“1011”时,按下的是开关键;键码为“1101”时,是位值键,键码为“1110”时,是确认键。在开关键按下时,若尚未开始滤波,则程序开始执行滤波工作;而在已滤波状态,按开关键,则停止滤波工作。在位选键按下时,则显示数字被移位显示。在位值键按下时,数值值自动增加。确认键按下时,则刚才输入的数值有效;在数值有效时,确认键再按下,则数值未被确认,故还可以修改数据。

在确认键对输入的数据进行确认了之后,则说明所输入的频率数据有效,这个频率即是滤

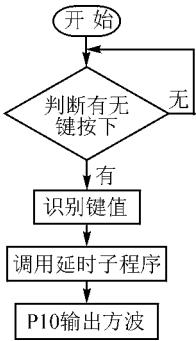


图 10-10 控制程序框图

波的截止频率。这时,单片机会根据这个截止频率的值,由程序产生用于控制 MAX292 的时钟频率,并且从 P10 端口送出到 MAX292 的时钟输入端 CLK 中去。在 P10 端口输出方波有多种方法,最容易的方法是利用输入的截止频率值求出时钟频率值,接着计算出时钟周期,再把时钟半周期值送入定时器,由定时器进行计数产生中断,每中断一次则改变 P10 端口的值一次,即对 P10 端口的值求反一次。这样,就可以在 P10 端口产生和截止频率对应的时钟频率输出,从而保证 MAX292 以截止频率进行滤波。

10.2.2 RS 码的编译码器

纠错码是提高通信可靠性的一门学科。在 1948 年,Shannon 在“A mathematical theory of communication”中提出了信道编码定理,从而奠定了这门学科的基础,并且在近 50 年来已取得了丰硕的成果。在实际应用中,利用纠错码的差错控制技术对通信系统提高可靠性水平来说,已经是一种极重要的技术方法。纠错编译码器在现代通信系统中已经成为极重要的组成部分。在纠错码中,最常用的有线性分组码、BCH 码、RS 码、Coppa 码等。BCH 码是纠正多个随机错误的线性分组码,而 RS 码是多进制的最大距离可分码。BCH 码是一种循环码,它的纠错能力强,构造简便,编译码简单,译码方法速度快,是至今所发现的性能优良的线性纠错码的一个类别。RS 码则是有很强纠错功能的一类多进制 BCH 码,它不仅在一般通信系统中广泛使用,而且在 CD 和 DVD 等光盘存储系统中也得到大量的应用。现代通信中,由于环境的复杂性,通信信号所受到的较多是随机性突发干扰。很明显,采用纠错码,例如 RS 码对干扰进行排除是一种十分有效的途径和可行办法。

在通信系统中,为了保证信息的正确性,需要对外界环境的各种干扰进行排除。而在各种抗干扰措施中,采用纠错码对传送的数据进行纠错是一种常用的方法之一。各种纠错码都有其一套编码和解码的算法,并且其复杂程度不同。把各种纠错码的编译码器用集成电路实现,可以嵌入到各种通信系统及通信设备之中,这样不但减少了通信系统可靠性开发的困难,而且减少了可靠性开发的重复性,同时,还可以把编译码器做成一个模块,在系统中作元件使用,从而使系统的开发变得方便并且更为容易。

单片机以其低廉的成本和高度的灵活性,在工业控制、通信乃至家用电器等各个领域取得了极为广泛的应用。在工业控制中,经常涉及到数据的远程采集,如电力负荷监控系统,需要将远方受控站点的采集数据送回主站点进行处理,或是主控站需将控制指令发送至远方受控站点,这对数据的准确性提出了较高的要求。然而在实际的数据传输过程中,由于信道的干扰,数据往往会出现错码,特别是在借助无线通信进行数据传输的情况下,突发干扰会造成大片错码。采用前向纠错码(FEC),可以极大地提高系统的可靠性。在常用的纠错码中,Reed - Solomon 码具有很好的抗突发干扰的能力,特别是将其与交织技术相结合,能很好地对抗因突发干扰而造成的成片错误。但是由于 RS 码是一种定义在 $GF(2^m)$ 上的多进制 BCH 码,其编码算法和译码算法都涉及到 $GF(2^m)$ 域上的运算,实现起来较为复杂,因而在实用中一般采用 ASIC 芯片做成专用的 RS 编译码器。如果能够将 RS 编译码算法在单片机中实现,就可以将 RS 的编译码程序作为功能模块嵌入到需要对数据进行纠错的其他工业控制软件中,而不需要专门配备一个 RS 编译码器。这无论从成本上还是从应用的灵活性上而言,都是非常具有吸引力的。

1. Reed - Solomon 编译码算法

RS(Reed - Solomon)码是在有限域 $GF(2^m)$ 上构造的线性分组码。它可以纠正多重的突

以直接采用代数译码方案。基本原理如下。

假设接收多项式

$$\gamma(x) = \sum_{i=0}^{15} \gamma_i x^i = v(x) + e(x) \quad (10-3)$$

其中错误多项式 $e(x)$ 表示为

$$e(x) = \sum_{i=0}^{15} e_i x^i, e_i \in GF(2^5) \quad (10-4)$$

伴随子定义为

$$S_i = \gamma(\alpha^i) = v(\alpha^i) + e(\alpha^i) = e(\alpha^i) = \sum_{j=0}^{15} e_j (\alpha^i)^j \quad (10-5)$$

其中 $i=14, 15, 16, 17$ 。伴随子包含了关于错误的出现位置及错误值的信息。

对于 RS(16,12) 码, 根据其错码的多少可以分为以下 3 种不同的情况。

(1) 包含两个错码: 设在第 i_1 个码元位置错误值为 Y_1 , 在第 i_2 个码元位置错误值为 Y_2 , 则校正子可构成如下的方程组:

$$\begin{aligned} S_{14} &= e(\alpha^{14}) = \sum_{i=0}^{15} e_i (\alpha^{14})^i = Y_1 X_1^{14} + Y_2 X_2^{14} \\ S_{15} &= e(\alpha^{15}) = \sum_{i=0}^{15} e_i (\alpha^{15})^i = Y_1 X_1^{15} + Y_2 X_2^{15} \\ S_{16} &= e(\alpha^{16}) = \sum_{i=0}^{15} e_i (\alpha^{16})^i = Y_1 X_1^{16} + Y_2 X_2^{16} \\ S_{17} &= e(\alpha^{17}) = \sum_{i=0}^{15} e_i (\alpha^{17})^i = Y_1 X_1^{17} + Y_2 X_2^{17} \end{aligned} \quad (10-6)$$

其中 $X_1 = \alpha^{i_1}, X_2 = \alpha^{i_2}$ 。

(2) 有一个错码。可令上式中的 $Y_2=0$,

求得

$$\begin{aligned} X_1 &= S_{15}/S_{14} \\ Y_1 &= S_{14}/X_1^{14} \end{aligned} \quad (10-7)$$

同时可求得有一个错码的条件为

$$S_{15}^2 = S_{14} S_{16} \quad (10-8)$$

(3) 错码数超过两个。可根据式(10-6)求得此时需满足的条件为

$$S_{14} S_{17} + S_{15} S_{16} = 0 \quad (10-9)$$

可以看到, 在译码算法中, 求解由式(10-6)所构成的方程组并不是一件容易的事情, 因为它是一个非线性方程组, 错误的位置 X_1 和 X_2 是一个 2 阶多项式的根, 所以可引入错位多项式

$$\sigma(x) = x^2 + \sigma_1 x + \sigma_0 \quad (10-10)$$

它的两个系数与校正子的关系为

$$\sigma_1 = \frac{S_{14} S_{17} + S_{15} S_{16}}{S_{15}^2 + S_{14} S_{16}}, \quad \sigma_0 = \frac{S_{16}^2 + S_{15} S_{17}}{S_{15}^2 + S_{14} S_{16}} \quad (10-11)$$

它的两个根之间的关系为

$$X_2 = \sigma_1 + X_1 \quad (10-12)$$

有两种方法来求得第一个根 X_1 ：

第一种方法是钱氏搜索法,即将 $x=\alpha^i(i=0,1,\cdots,15)$ 代入,如果 $\sigma(\alpha^i)=0$,则 α^i 为其一个根 X_1 ,而另一个根 X_2 则可由式(10-12)计算出来。

第二种方法是采用查表法,它的思路是事先做好一张表,表的内容是多项式的一个根,而表的指针地址则由式(10-11)所计算出来的系数表达式。因为对 $\sigma(x)$ 的两个系数总共有 2^{10} 种组合,所以表的大小总加起来有 $1\text{ K}\times 5$ 位。

计算出错误的位置 X_1 和 X_2 后,就可由方程组求得错误值 Y_1 和 Y_2

$$Y_1 = \frac{S_{15} + S_{14}X_2}{\sigma_1 X_1^{14}}, \quad Y_2 = \frac{S_{14} + Y_1 X_1^{14}}{X_2^{14}}$$

(10-13)

2. 单片机型的 RS 编译码器

由单片机组成的 RS 编译码器可以以一个完整的部件形式嵌入到通信系统中,从而可获得体积结构、逻辑结构都令人满意的系统。

以单片机为核心的 RS 编译码器如图 10-12 所示。它主要由下列有关部分组成。

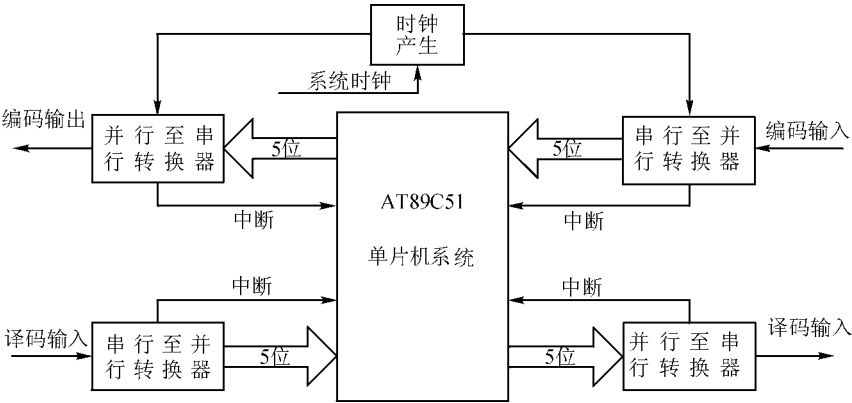


图 10-12 基于单片机的 RS 编译码器

(1) 单片机系统。这是 RS 编译码器的核心,负责编码信号输入后的编码工作,把信号编码完毕之后再输出到外部去;同时,它也负责需译码信号的译码工作,译码之后把结果送出到外部去。无论编码还是译码,都是由单片机的内部软件实现的。单片机软件按 RS 算法对信号进行编译码处理。

(2) 并/串行转换器。单片机对信号进行了编码或译码之后,产生 5 位字长的信号码。并/串行转换器的任务就是把单片机输出的 5 位字长并行信号码转换成串行信号码,以串行方式发送到外部去。

(3) 串/并行转换器。来自外部的信号是 5 位串行信号码。这种信号码有 2 种类型:一种是未编码信号,故需用单片机对其进行编码;另一种是已编码的信号,则需要译码。无论是需编码信号还是需译码信号,都要把它们从串行方式转换成并行方式,才能送入单片机处理。串/并行转换器就是完成把 5 位串行码转换成 5 位并行码这个任务的部件。

(4) 时钟发生器。时钟发生器是由系统时钟同步的。它产生的时钟信号同时送到串/并行以及并/串行转换器,以实现串行和并行信号转换时的同步。

从图 10-11 所示的 RS 编译码器可以看出:这个编译码器十分简单,结构简便,可以做成

一个结构完整、功能专用、体积小的专用模块。这样,嵌入到通信系统中则十分方便和有效,因为它几乎可以作为一个零件在系统中使用。

3. RS 编译码器的软件实现

在单片机内,RS 算法是通过有关软件实现的。

首先介绍编码过程。编码的过程如图 10-11 所示。利用单片机实现只需完成 12 个节拍的移位及反馈运算,以求得 $x^4 m(x)$ 除以 $g(x)$ 的余式。12 个节拍后,移位寄存器中的 4 个符号即为监督位,将 4 个监督位附加在 12 位信息码后,即可得到 RS(16,12) 码字 $(m_{15}, m_{14}, m_{13}, \dots, m_5, \gamma_3, \gamma_2, \gamma_1, \gamma_0)$ 。

特别要指出的是,所有的运算均是在 $GF(2^5)$ 中进行的。所以无论是移位寄存单元还是反馈系数,都是 5 位的符号单元,在单片机中均需占用一个 8 位的存储单元,它的加法和乘法运算同样需遵循 $GF(2^5)$ 域的运算准则。

译码程序的流程图如图 10-13 所示。

首先根据接收到的符号序列计算出伴随子

$$S_i = (\dots((v_{15}\alpha^i + v_{14})\alpha^i + v_{13})\alpha^i + \dots + v_1)\alpha^i + v_0)$$

其中 $i=1,2,3,4$ 。

然后分阶段进行合理性判别。首先判别是否有错码。如果无错码,则无需进行进一步的处理,可以结束;如果有错码,则应判别所接收的符号序列错码的个数。如果错码大于 2 个,则超过了这个编译码器的处理能力,故也退出处理;如果错码不多于 2 个,则对错码数进行确定,并按错码 1 个和 2 个的情况分别求取错码位置,然后求取错码的值。在得到错码位置和错码值后,对错码进行纠正,然后结束处理过程。

找到伴随子后,就可根据式(10-8)和式(10-9)计算出错码的个数。如果没有错码或是错码的个数大于 RS(16,12) 码的纠错能力 2 个码元时,则程序退出处理,否则进行纠错处理。如果发生了一个错误,则可根据式(10-7)计算出错误的位置及其错误值。对于两个错误的情况,则首先需根据式(10-10)构造错位多项式 $\sigma(x)$,然后利用钱氏搜索法求得它的根。根据求得的根获得误码发生的位置,然后代入方程组(10-13)中求解出误码值。有了误码位置和误码值,就可对接收到的码字进行纠错。这里采用钱氏搜索法的原因是这种算法与查表法相比,所需的存储单元要少得多。这样,只需利用 AT89C51 容量为 256 字节的片内数据 RAM 和容量为 4 KB 的片内程序 EEPROM 就可单片实现 RS 编译码器,而不需要再外加数据和程序存储器。

系统的原理框图如图 10-12 所示。该系统设计成一个独立的 RS 编译码系统,其中串/

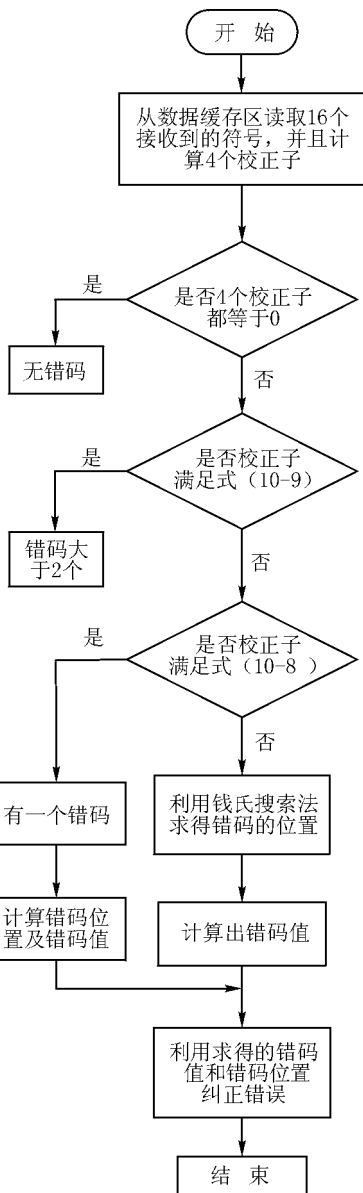


图 10-13 RS(16,12)译码算法流程图

并转换器是为了将比特流转化为 $GF(2^5)$ 域内的符号,因为 $RS(16,12)$ 是定义在 $GF(2^5)$ 上的,所以每个符号由 5 位组成,如下式表示

$$m_i = (b_{i5}, b_{i4}, b_{i3}, b_{i2}, b_{i1}) \quad \text{其中 } i=0,1,2,\dots,11, b_{ij} \text{ 为二进制码元。}$$

同样,并/串转换器的作用是将译码后的符号恢复成比特流。

10.3 无线短波传真通信控制

随着电子技术的进步和发展,通信理论不断地更新,通信的工具和方法越来越多,也越来越先进。传真通信在通信领域已经有近百年的历史,由于这种通信方式操作简单,快捷方便,内容直观,易于辨认,故仍然被人们广泛应用。特别在近十几年来,随着 G3 类传真标准的发布,传真通信得到了迅速的发展和更广泛的应用。目前,传真通信是人们在文本、图像传送中所使用的极其重要的工具。它在商务、会议、办公等方面有着十分重要的优势,并且受到人们普遍的欢迎。

G3 类传真真是 CCITT(ITU 前身)制定的基于数字通信技术(HDLC)的传真标准,适用于公用电话交换网(PSTN)上进行传真通信。由于采用了图像压缩技术,不论是传输时间还是传真质量较 G2, G1 类均得到了很大的提高,目前几乎所有的传真机及计算机传真通信均采用 G3 类传真标准。但是,在一些特殊应用场合,如无线短波通信中,由于信道的高误码率、低速率及不能全双工链路实时应答等多种因素,使得传真信号不能直接上信道传输。为了克服这些问题,专门采用了一个“传真转储模块”,用于解决短波信道的传真通信问题。其基本思路是:在本地由“传真转储模块”与普通传真机相连,按传真规程协议接收下传真文件,存储在 RAM 中,再经 HF modem 及短波发信机按短波数字通信方式发送;在远端由短波收音机、HF modem 及“传真转储模块”接收并存储在 RAM 中,然后再按传真协议转发给传真机输出。

“传真转储模块”是解决无线短波传真通信的关键。这个模块是以 AT89C55 单片机为核心组成的,在 AT89C55 的控制下,“传真转储模块”完成传真的收发控制,实现不同信道的传真通信。

10.3.1 传真通信过程

传真即为多页静止图像的传输。T. 30 协议规定了有关在 PSTN 上进行传真通信的细节,其收、发两端通过低速率(一般为 300 bps)的 HDLC 帧交换信息,协调每一步操作,然后在预定的速率上传送正式传真信息(报文)。一个完整的传真过程分成 5 个阶段,如图 10-14 所示。这 5 个阶段分别称为阶段 A, B, C, D, E。其中,阶段 A, E 分别是建立呼叫和释放呼叫,而阶段 B, C, D 则是工作阶段,即执行传真过程。这 5 个阶段分别说明如下。

(1) 阶段 A, 呼叫建立:通过人工或自动(AT 指令程控)的方法,摘机拨号,建立从发方到收方的电话电路并接至传真机,完成呼叫建立。

(2) 阶段 B, 识别和协商:由于传真机的性能存在差异,包括编码方式、通信速率、页面规格(分辨率、长、宽度)等,因此,在进行正式报文传输前,收、发传真机要先进行性能协商,统一成双方都能接受的指标。此外,线路质量到底适合多高的速率传送报文也需检测验证,因而本阶段还进行线路质量的检测,称之为“G3 训练”,即先以预定的速率发送测试码,收方接收并验证可以按此速率传真,否则需下降速率再训练,直至认可。

- (3) 阶段 C,报文传输:在完成报文前协商、训练后,即进入本阶段,开始传真报文的收发,以及报文中的同步、差错控制、线路检测等。
- (4) 阶段 D,页后过程:当一页传真完成后,进入此阶段,通过低速率 HDLC 帧联络确认当前页传真是否成功,下一步是接着传下一页还是结束本次传真(进入下一步)。
- (5) 阶段 E,呼叫释放:通过人工或自动(AT 指令)方式挂机。

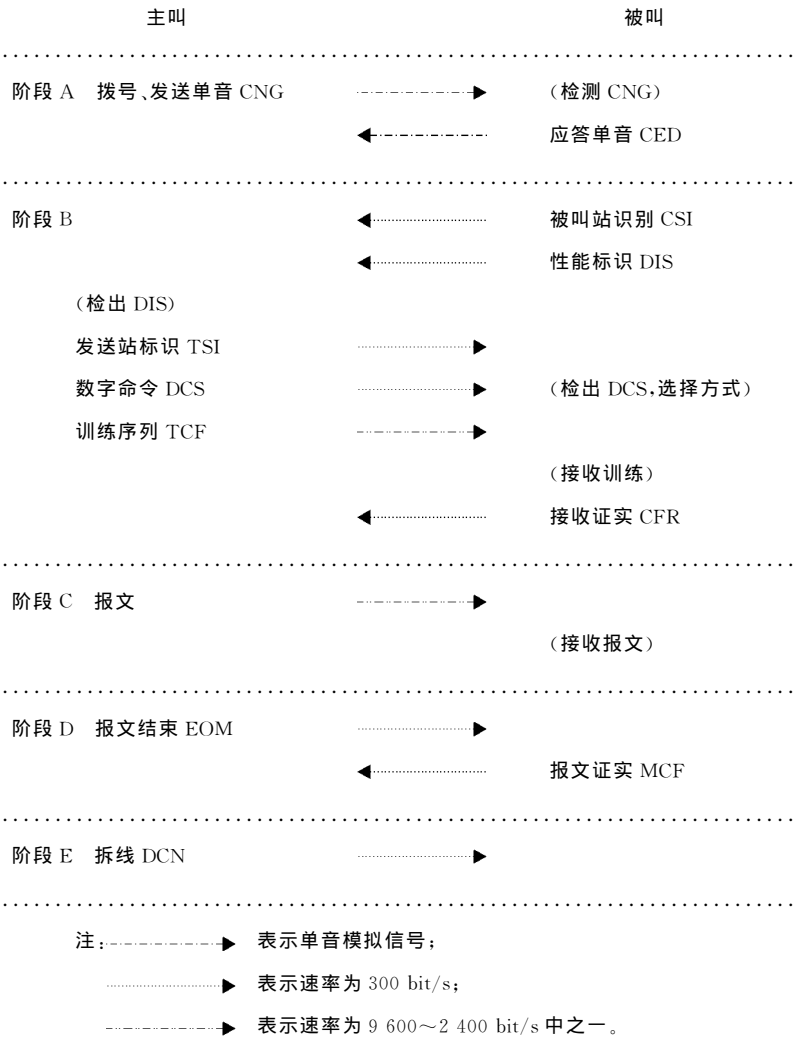


图 10-14 传真通信过程

10.3.2 传真转储模块电路结构

传真转储模块是无线短波收/发信机和传真机之间的一个中间部件。它的作用有两个:一个是把本地传真机的发送信号接收入模块的存储器中,在 AT89C55 控制下发送到 HF modem,再由短波发信机发送出去;另一个作用是由短波收信机接收远方发来的传真信号,通过模块处理后送往传真机输出。

传真转储模块包括 AT89C55 单片机、随机存储器 628512、Fax/Data Modem 芯片

4. RS-232C 串行接口

通过 P14~P17 组成 RTS/CTS 等握手信号,AT89C55 的 P30,P31 为 RXD,TXD,实现基本 RS-232C 串行接口与 HF modem 相连,完成至短波电台的数据通信接口。

5. 74LS373 锁存器

74LS373 用于在存取 628512 RAM 时存放低位地址,因为 P00~P07 是地址/数据复用 I/O 端口。在对 RAM 存取时,先发送低 8 位地址,后发送数据;而在控制 Fax/Data Modem 芯片 RC224ATF 时,则发送数据。

10.3.3 传真转储模块控制软件

传真转储模块控制软件要完成的主要工作有 3 个,它们分别如下。

第一,通过对传真接口的控制,实现和传真机的连接,完成传真文件的收/发任务。

第二,控制 RS-232 串行口格式的数据流,实现和 HF modem 的数据通信。

第三,实现对 512 KB 的随机存储器 628512 的存储单元进行存取。

下面分别介绍这 3 种主要的软件功能。

1. 对 628512 RAM 的存取

由于受 MCS51 系列单片机寻址空间的限制,AT89C55 最大只能同时访问 64 KB 的数据空间(16 位地址 A0~A15)。而 Fax Modem 等其他外围芯片也要占据地址空间,为了软件编程简单同时使硬件设计最简化,存储器只占用 8051 的 0000H~7FFFH 数据地址空间(32 KB,使用地址 A0~A14)。用 P10~P13 与存储器的 A15~A18 相连,作为页面地址,将 628512 的 512 KB 划分成 16 个页面,采用译码方式进行页面切换,每次只访问其中的某一 32 KB 存储空间。在跨越不同页面访问时,需先设置页面地址,再进行访问。下段程序是将 RAM 中 07FF0H~1000FH 的 32 字节单元内容读入片上 RAM 的例子。

```

; R0=片上 RAM 区首址
MOV    R1,#11110000B    ; R1 的位 0~3 为 RAM 的 A15~A18 页面地址
MOV    DPTR,#7FF0H      ; RAM 首址
MOV    R7,#32           ; 读字节计数器初始化
RLL:   MOV    P1,R1       ; 设置页面地址
        MOVX  A,@DPTR    ; 读一字节
        MOV    @R0,A      ; 存入片上 RAM
        INC    R0         ; 写地址加 1
        INC    DPTR       ; 读单元地址加 1
        MOV    A,DPL
        ORL    A,DPH
        JNZ    SKIP       ; 当(DPTR)+1=0 时,地址转入下一页面
        INC    R1         ; 修改页面地址
SKIP:   DJNZ   R7,RLL     ; 循环计数
...
```

2. RS-232 串行信号的接口

对于外置式调制解调器,大都采用 RS-232 串行接口与数据源设备通信。标准 RS-232

具有多个辅助电路信号进行通信间的握手协调。随着技术的发展,不论是 Modem 还是数据源(通常是计算机控制)都采用内部 buffer 来缓冲通信收、发,因此可以采用经济的 3 线(收、发、地)连接方式。但在这个模块中,由于 HF 信道的特殊性(HF Modem 采用如交织、纠错重发等技术手段来保证信道的低误码率,使得信道的实际通信速率随时在变化),因此,与 HF Modem 的 232 接口必须具备完备的握手信号以避免数据在 HF Modem 中拥塞而丢失。在这个模块中,采用 AT89C55 的 P14~P17 为其握手信号 I/O 脚,并采用 MC14488/14499 将 TTL 电平转换成 EIA($\pm 15\text{ V}$)电平。

RS-232 的握手信号时序如图 10-16 所示。单片机 AT89C55 的 P14~P17 引脚输出的信号必须满足握手信号的时序要求。这个时序的产生是由 AT89C55 的软件控制的。

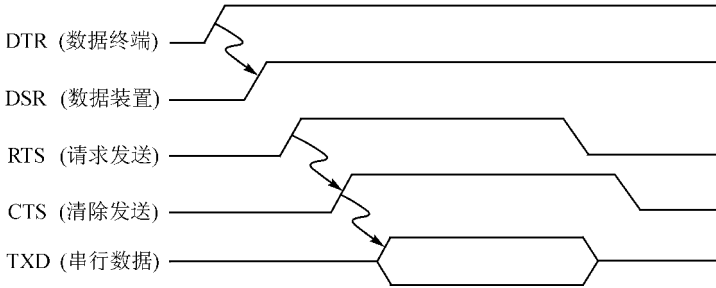


图 10-16 RS-232 握手时序

3. 对传真机收/发的控制

G3 类传真实际由两个标准构成,其一是描述通信协议的 T. 30 建议,其二是描述传真图像编码方案的 T. 4 建议。G3 传真的编码方案主要有 MH 码(改进的霍夫曼码)和 MR 码(改进的像素相对地址指定码)两种,它们是通过传真机上的编码器将光电扫描信号进行编码或将收到的传真文件经译码器解码输出来实现的。这部分工作由传真机完成,因此“传真转储模块”主要实现传真通信协议 T. 30,即可完成与传真机从沟通联络到传真文件的收、发等整个传真过程。

(1) “1 类传真”服务。众所周知,Data Modem 是通过标准 AT 指令来完成其操作的控制,如摘机拨号、应答、速率设置等,而 Fax Modem 则是通过专门的扩展 AT 指令来实现相应的控制。这些扩展 AT 指令按照 EIA 的标准,分成“class 1”,“class 2”,“class 2.0”三类,以“class 1”为最基础(不仅要准备传真图像 T. 4,而且要处理完整的传真会话 T. 30),其应用也较广泛。

RC224ATF 芯片提供“class 1”AT 指令,亦称之为“1 类传真”服务。它对 T. 30 一无所知,仅提供一些简单而低级的原始指令。控制程序通过使用这些指令,对 HDLC 帧进行装配、分解及分析有关应答信息,来获知运行状态,控制传真信号的发送。

(2) “T. 30 协议”的实现。要实现 T. 30 协议,实际上是用单片机 AT89C55 按所需的要求来控制 Fax/Data Modem 芯片 RC224ATF。由于这个芯片是一种在通信中通用的 Modem 芯片,在硬件上它提供面向寄存器的编程接口,实际上即是一个仿真 INS8250 的 8 位并行 CPU 控制接口,能很方便地访问其中的数据收/发及线路状态寄存器。

作为控制软件,最底层是对 RC224ATF 寄存器组的访问操作,以完成向 Fax 芯片写入或读出一个字节的信息,称之为物理层控制。此时软件操作和判断的是收/发及线路状态寄存器的内容。下例为采用汇编指令的代码段。

；读一个字节数据子程序

```

READ:    MOV    DPTR,#8005H    ; 线路状态寄存器
LR:      MOVX   A,@DPTR        ; 读状态字
          JNB    ACC.0,LR       ; 若 B,收到数据
          ANL    A,#00011110B   ; 屏蔽无用标志位
          JZ     RR             ; 若 Z,收未出错;当前接收字节出错,处理
RR:      MOV    DPTR,#8000H    ; 接收数据寄存器
          MOVX   A,@DPTR        ; 读一个字节数据
          RET

```

；写一个字节数据子程序

```

WRITE:   MOV    B,A            ; 保存待发字节
          MOV    DPTR,#8005H    ; 线路状态寄存器
WR:      MOVX   A,@DPTR        ; 读状态字
          JNB    ACC.5,WR       ; 若 B,发送寄存器空
          MOV    A,B
RR:      MOV    DPTR,#8000H    ; 发送数据寄存器
          MOVX   @DPTR,A        ; 写一个字节数据
          RET

```

在实现物理层控制的基础上,再完成的是数据链路层的控制,在此即是在完成对 AT 指令流和 HDLC 帧收/发的基础上,进一步实现传真会话协议 T. 30 控制。

通常,数据 Modem 芯片在一上电或复位后,与计算机的接口自动进入 AT 指令状态,此时向 Modem 发送的信息均被认为是 AT 指令而加以解释执行。在完成拨号、连接(connect)后,Modem 自动切换至数据状态,此时向 Modem 发送的信息均被认为是数据而上线路传送到对方,直到计算机向 Modem 发送一个换码序列(连续的“+++”串),才使 Modem 退回至 AT 命令状态。对 Fax Modem 来讲,基本模式也是这样,上电初始在 AT 命令态,此时应用 Class-1 扩展 AT 命令集对其进行操作,实现传真的呼叫建立、性能协商、线路验证等。

10.4 程控交换机呼叫信息缓冲系统

程控交换机是电话通信系统中应用极广的通信交换设备,在一般的电话通信过程中,程控交换机生成的电话呼叫信息一般通过其标准的串行接口输出,电话计费系统软件可以用计算机直接从交换机接收呼叫信息。但呼叫信息的输出是随机的,取决于交换机支持的网上分机电话的使用状态。为防止信息丢失,计算机系统必须保持全天 24 h 连续工作。程控交换机话单自动存储器的设计与应用,一定程度上提高了计费工作的灵活性,降低了系统的运行成本。但早期系统或是不具备多任务工作特性,使用不便;或是采用静态 RAM 作缓存,导致数据存储可靠性不高,系统功耗较大;而且系统功能设计较低且不规范,灵活性较差。

以 AT89C52 单片机为核心,采用 Intel Strata 系列 Flash 存储器作缓存,设计开发的新一代程控交换机呼叫信息缓冲系统,用于串行数据的接收、缓存和读出,具有多任务工作特性,而且实时性好,读出速度快,数据存储可靠性高,使用灵活方便。

这一节介绍这种呼叫信息缓冲系统的有关性能、结构及其软件功能。

表 10-1 中所列的资源分配方案说明了在呼叫信息缓冲系统中,AT89C52 采用哪一种内部部件对系统的相关电路或部件进行控制。

2. Flash 存储器阵列

采用 Intel 公司新近推出的 Strata 系列 Flash 存储器 28F320J5 和 28F640J5 构成缓冲存储器阵列,硬件支持 4 MB,8 MB,12 MB 或 16 MB 缓存空间配置,采用无跳线设计。

3. Modem 接口电路

设置扩展 UART 接口芯片 TL16C550,采用 MAXIM 公司的 MAX3225 完成 CMOS 与 RS-232C 电平转换,用于与计算机或 Modem 接口。此设计方案的特点是:串口收发各有 16 字节的硬件 FIFO,速度高达 460 kbps,接口抗静电能力为 15 kV。

4. 串行接口

利用 AT89C52 的口线构成软件串行接口。采用电流环 I/O 驱动电路和相应的电平转换电路与交换机串口连接,支持通信距离可达 1 km。

5. 电池控制电路

后备电池管理及 DC-DC 变换电路。串行输出 A/D 变换器 TLC0838C 主要用于外部电源及电池电压及容量的监测,而并行输出口 74HC374 芯片则用于电池充放电控制及 MAX3225、电流环电路的开关控制。

6. 参数保存及监视电路

单片串行 EEPROM X25045 用于系统参数的配置与保存、低电源电压监视、可编程硬件看门狗电路。

7. 片外存储器

Flash 存储器 AT29C256 用于微控制器片外执行系统软件。用户还可以利用 AT89C52 内部程序存储器中的重载模块和 AT29C256,实施系统软件的在线升级。

10.4.3 呼叫信息缓冲系统软件

这个软件主要由主程序、信息缓冲存储以及软串行口等 3 大部分组成。下面分别对这几部分进行简要介绍。

1. 主程序流程

图 10-18 为包括初始化和主循环的系统软件主体流程。其中,初始化程序完成系统主缓存容量及其读写断点的检测,包括实现对 28F320J5 与 28F640J5 混合配置及无跳线硬件设计的支持;从串行 EEPROM 中读取系统配置参数;设置各种报警门限值;初始化软串口和 TL16C550 硬串口波特率等;单片机和 TL16C550 接口硬件的初始化编程。主循环中,后备电池管理主要包括电池工作模式及其工作时间;电池可维持系统工作时间;及电池容量信息的管理、电池充放电控制等。LED 及蜂鸣器控制提供诸如空闲缓存容量低于门限、直接或 Modem

表 10-1 AT89C52 硬件资源分配方案

资 源	功能分配
P0,P2	扩展外部存储器和接口电路
P2.5~P2.7	被译码提供外扩芯片的片选
P1.0~P1.4	X25045,TLC0838C 的 I/O 及时钟
P1.5~P1.6	Buffer, Power LED 控制
P1.7,P3.5	软串口 CTS,RTS
P3.4	软串口 TXD
P3.3(INT1)	TL16C550 中断信号
P3.2(INT0)	软串口 RXD,Data LED 控制
P3.0,P3.1	蜂鸣器、Strata 缓存控制
P3.6,P3.7	外扩存储器、接口的读写信号
定时器 T0	定时产生蜂鸣器控制波形
定时器 T1	电池管理,LED 控制定时
定时器 T2	用于软串口工作定时中断

连接、电池供电、软串口连接故障等状态的指示或声光报警。T1 定时中断服务为其提供基准定时,而 T0 则用于产生约 2.4 kHz 的“蜂鸣器”驱动波形。

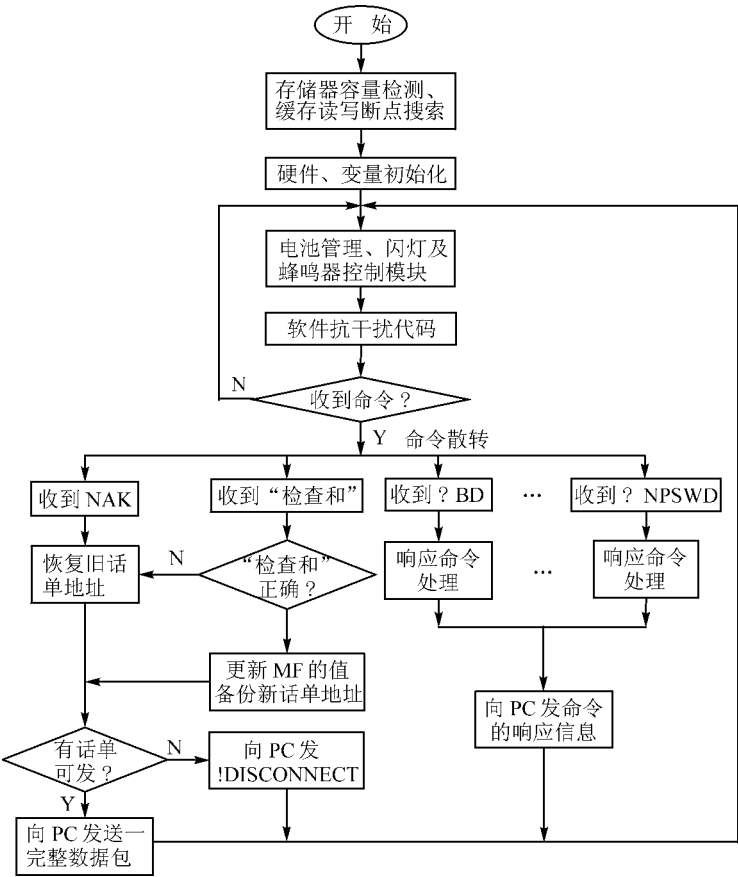


图 10-18 主程序流程

表 10-2 系统支持的主要命令及含义

命 令	含 义
NAK	请求系统发信息包
ACK+检查和	响应接收的信息包
? BA	起动波特率自动识别功能
? BATT	查询电池状态信息
? BC	查询波特率调整次数
? BD	查询软串口当前波特率
? BF	固定或切换软串口波特率
? INIMEM	清系统的主缓存空间
? LOGOFF	系统进入保护模式
? MD	查询系统缓存的全部话单
? MF	查询系统主存空闲容量(KB)
? MONITOR	设置/取消系统监视接收功能
? MT	查询系统主缓存总容量(KB)
? NPSWD	修改系统口令
? RELOAD	请求重载系统软件

表 10-2 为系统支持的命令集(共 37 个命令)中最基本的一部分。其中“NAK”和“ACK+检查和”用于呼叫信息读出协议。系统收到“NAK”或错误“检查和”时,发送“NHK+话单”信息包(一条话单为一次电话呼叫产生的以 CR 或 CR/LF 结束的呼叫信息串);收到正确“检查和”时,发送“ACK+话单”信息包;无话单可读时发送响应信息“!DISCONNECT”。此外,协议规定一般命令以问号开头,响应信息以感叹号开头。

2. 信息缓冲存储

交换机呼叫信息的接收及缓冲存储,是嵌入在各功能模块中完成的,如图10-19

所示。针对 Strata Flash 存储器的编程特点,在 AT89C52 内部 RAM 区开辟 2 个大小为 16 字节的编程缓冲区。软串口接收到的串行数据经动态压缩之后依次循环填入缓冲区,一旦填满一个缓冲区,即通知系统实施缓冲编程。各功能模块在执行过程中及时主动检测相应的通知标志,确保呼叫信息的实时缓存。

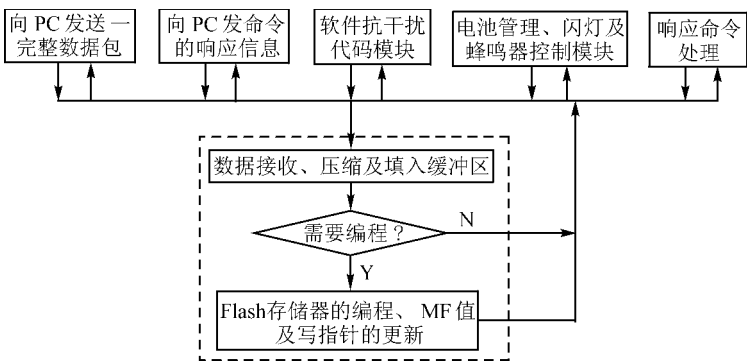


图 10-19 呼叫信息嵌入接收与存储

3. 软件串行接口

以 AT89C52 的 INT0 引脚为 RXD,结合 INT0 边沿中断和 T2 定时中断服务程序的设计,以软件方式控制串行呼叫信息的采样接收。同时通过监测记录串行帧起始位及 LSB 的特征,设计相应的技术准则,实现对交换机串口输出波特率的自动识别与跟踪。

以 AT89C52 为核心,选用性价比高的主存储器和外围接口芯片,配合强大灵活的软件设计,开发程控交换机呼叫信息缓冲系统,实现了低成本、大容量、高实时和多任务特性。此外,系统还具有以下重要特点:

- (1) 软串口具有波特率自动识别能力;
- (2) 信息读出速度可达 460 kbps;
- (3) 后备电池管理,声光报警;
- (4) 系统软件在线重载;
- (5) 命令集设计规范,功能完备。

10.5 现场总线通信适配器

计算机控制正处于从单机控制进入网络控制这种发展潮流之中。从控制系统的角度上考虑,网络控制有利于取得全局综合最优效果,有利于充分利用资源,有利于进行调度管理,从而取得高质、低耗、省时等一系列效果。

单机控制在一些领域仍然有很大的市场,但是,网络控制已渐渐深入到各个领域之中,特别在工业控制领域,单机控制已无法满足生产过程及设备工作的要求。因为在这些领域中,生产过程或设备只是系统的一个部分或基本控制单元,它们需要外部的相关的输入控制信息或数据信息,同时也需要和其他设备进行信息交换、联机工作或者实现互操作。为了达到这些目标,采用网络控制技术是一种必然的发展趋向。用网络控制技术将众多设备联接成网络进行控制,对提高系统的控制综合水准、资源共享、安全可靠性以及任务调度都有极大的好处。

现场总线以及各种信息总线是网络控制中的重要技术标准,是网络控制中信息流控制的规则,在网络控制中有着极其重要的意义。

实际生产的巨大需求促进了各种总线的发展,目前,国内外的现场通信网络较为流行的总线有:RS-232,RS-422/485,HART,Profibus, Dupline, CAN, Lonworks, FF 以及 CEBus。我国的实际情况是:RS-232,RS-422,RS-485 应用最为普及;CAN 网络的发展速度最快;而其他优秀的网络如 FF,Lonworks 在国内普及正在开始。现在国内外大部分的网络终端控制设备带有 RS-232,RS-422,RS-485 或者 CAN 接口,为了便于利用这些控制设备进行系统集成,需要现场总线适配器进行信息规则的转换。

这一节介绍可以在 RS-232,RS-422/485 和 CAN 总线之间实现信息接口的现场总线适配器。

10.5.1 适配器的功能要求

现场总线通信适配器包括 3 个通信网络接口,分别是 RS-232,RS-422/485 和 CAN 接口,能够完成以下 5 种功能。

1. RS-232 到 RS-422/485 之间的通信适配

带有 RS-232 接口的主机如 PC 操作站、过程站等,或从机如现场控制单元、智能网络节点等,可以通过现场总线适配器的 RS-422/485 接口,连入由其构成的局部控制网络,实现 RS-422/485 网络数据和命令的传输。

2. RS-232 到 CAN 网之间的通信适配

带有 RS-232 接口的主机如 PC 操作站、过程站等,或从机如现场控制单元、智能网络节点等可以通过现场总线适配器的 CAN 接口连入 CAN 网,实现基于 CAN 网的网络数据和命令的传输。

3. RS-422/485 到 CAN 之间的通信适配

现场总线通信适配器能够将一个系统中的 RS-422/485 网络和 CAN 网实现相互连接。这样,当源地址在 RS-422/485 网络,而目标地址在 CAN 网时,或者当源地址在 CAN 网络,而目标地址在 RS-422/485 网时,通信适配器可以完成两个网络之间的数据和命令的传输。由此解决了系统集成过程中不同网络终端设备的互连问题,使系统设计具有更宽的选择范围,达到更先进的水平。

4. 相同或不同网络之间的距离、节点数和分支扩展

由于 RS-422/485 和 CAN 网均为总线型网络,它们的通信距离、网络节点数目和网络分支扩展都有一定程序的限制,采用现场总线适配器后,可以任意扩展和连接,这在组建较为复杂的总线型网络时十分有用。

5. 通过连接外置 MODEM,实现网络数据和命令的更远距离传输

当更远距离的监控或管理中心需要遥测或遥控本地网络即 RS-422/485 或 CAN 组网的设备时,现场总线网络通信适配器可以采用连接外置 MODEM 方式,通过电话线或专线,完成本地 RS-422/485 网络或 CAN 网与远方监控或管理中心的数据和命令的传输。

10.5.2 适配器的电路结构

现场总线通信适配器由 5 部分组成,它们分别是适配器的最小系统、RS-232 接口电路、

RS-422/485 接口电路、CAN 接口和电源部件。其中电源是适配器的供电部件,适配器的最小系统是适配器的核心,其余 3 部分是接口部件,用于在 RS-232,RS-422/485 和 CAN 网络之间进行通道相连。

1. 适配器电源部件

电源采用体积小、重量轻的 AC-DC 模块,交流输入电压范围 165~265 V,输出两组电源,分别为 V_{cc} ,GND(+5 V/250 mA)和 +5 V,GND2(+5 V/150 mA)。这两个电源分成最小系统供电电源和接口供电电源两路,一方面可以满足系统工作的需要;另一方面,可以减少接口和最小系统在工作时因负载变动而带来的影响。

2. 适配器的最小系统

现场总线通信适配器的最小系统包括 MCU、监控复位电路、RAM 和非易失性串行 EEPROM 这几部分。

MCU 采用 ATMEL 公司的 AT89C51。它是适配器的信息中央处理单元,在软件的支持下,执行程序命令及数据的接收、处理并发送出相应的网络命令和数据,发挥总线适配器在现场局部网络中的作用。

复位电路采用 MAXIM 公司的 MAX824。它具有高、低复位电平输出、上电复位、欠电压复位、“看门狗”等多种功能,是 μP 监控复位电路的理想选择。

RAM62256(256 KB)用于通信过程中各网络的接收和发送缓冲区、网络节点的接收和发送缓冲区及 CPU 数据处理缓冲。

EEPROM AT93C46(512 KB)用于存放网络配置,包括网络种类、各网络节点数目及地址信息表等。

总线适配器最小系统电路原理如图 10-20 所示。

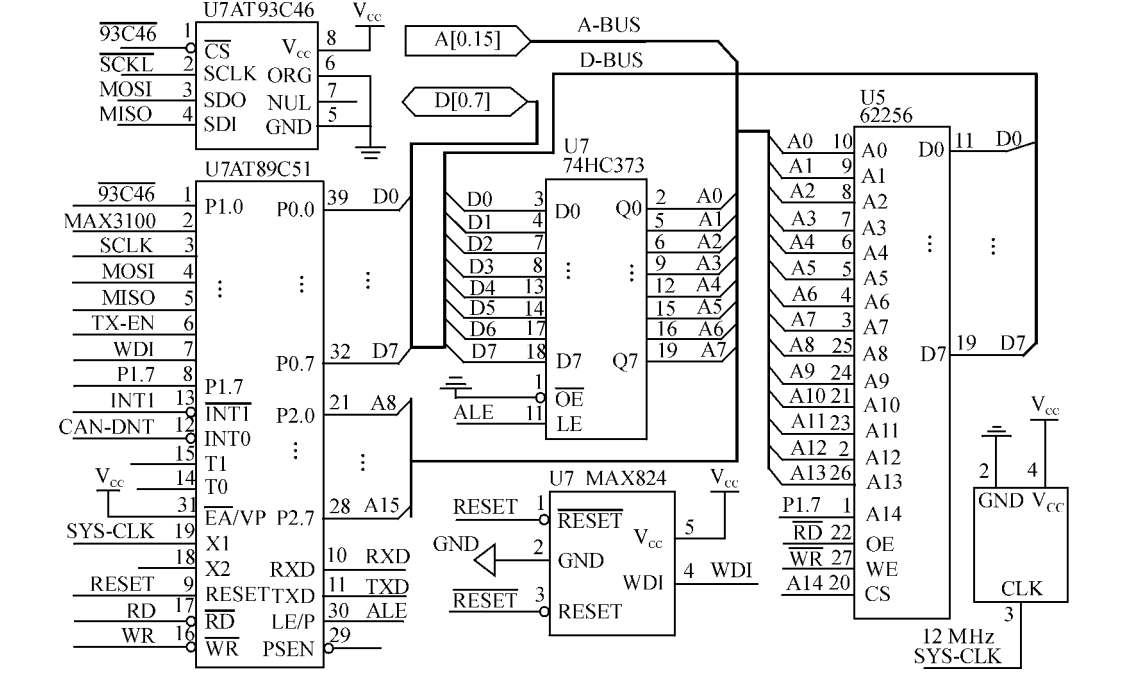


图 10-20 总线适配器最小系统电路原理图

AT89C51 通过并行口 P1.0,P1.2~P1.4 和串行 EEPROM AT93C46 连接,其串行读写由程序产生的信号执行。

AT89C51 通过 P0 端口和 P2 端口对随机存储器 RAM 62256 进行读写,其中 P0 端口是地址数据复用端口。AT89C51 对 62256 进行读写时,先通过 P0,P2 端口发送地址信号,其低位地址的 8 位则由 74HC373 锁存器锁存,接着 P0 端口就用于传送数据。

适配器的最小系统也就是一个基本计算机系统。由于 AT89C51 内部已含有控制器及运算器,外部再加上存储器 62256 以及 AT93C46,则组成了最起码的计算机系统,也称为最小系统。

3. RS-232 接口电路

由于现场总线适配器包括 RS-232 和 RS-422/485 两个独立的接口,势必需要两个异步串行口(UART),而 CPUAT89C51 只带有一个 UART,显然在适配器中需要外扩一个 UART。鉴于体积、功耗和性能价格比上的考虑,采用 MAXIM 公司的 SPI 接口的 UART 控制器 MAX3100 正好满足要求。其主要性能为:SPI/Microwire 兼容的 μP 接口,8 字节接收 FIFO,9 位地址识别中断,IrDA SIR 时序兼容,RTS 输出和 CTS 输入,最高波特率 230 kbps,工作电压+2.7~+5.5 V,工作电流仅 0.5 mA,DIP 14 或 QSOP 16 封装。由于 MAX3100 优良的性能,很适合在适配器中使用。

在电路结构上,RS-232 接口电路如图 10-21 所示。它包括 UART 控制器 MAX3100 和 RS-232/TTL 电平转换集成电路 MAX202。MAX202 是采用单一+5 V 电源供电的电平转换电路,比传统的采用 $\pm 12\text{ V}$ 或 $\pm 15\text{ V}$ 供电的 MC14488 和 MC14489 两片集成电路进行 RS-232/TTL 电平转换有很大的优越性。这样不但降低了功耗,减少了成本,而且还减少了体积,从而提高了系统的可靠性。

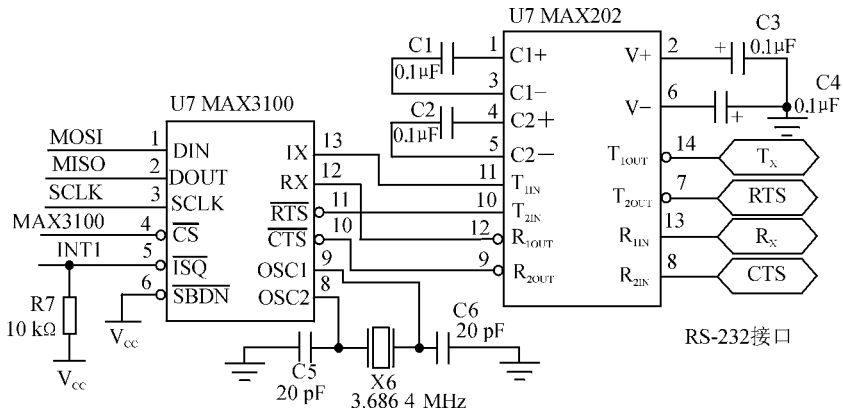


图 10-21 RS-232 接口电路

AT89C51 是通过 P1 端口的 P1.1~P1.4 和 MAX3100 进行信号传送的。由于在电路结构上,串行 EEPROMAT93C46 采用 AT89C51 的 P1.0,P1.2~P1.4 和 AT89C51 进行数据传送和控制,这样 P1.2~P1.4 所传送的信号 SCLK,MOSI,MISO 是公用通道。为了区别信号的传送对象,故 AT89C51 用 P1.0 选择 AT93C46,用 P1.1 选择 MAX3100。这样,就可以把两者的工作区分开来,从而有条不紊地执行各自的工作。

在 AT93C46 和 RS-232 接口电路的工作过程中,由于采用的是 AT89C51 的 P1 端口,它不是一个标准的串行端口,故相关的信号必须由软件进行模拟。因此,AT93C46 和 RS-232

的串行通信信号是由程序控制的数据序列形成的。这一点,和 RS-422/485 的接口电路有很大的区别。当然,由软件产生的串行通信信号在速度上会受到一定的限制。不过在一般的 RS-232 通信中,其通信速度并不高,通常在 9 600 bps 的水平左右,甚至更低。所以,这种电路结构是可以满足实际需要的。

4. RS-422/485 接口电路

RS-422/485 接口电路如图 10-22 所示。它由 MAX1482, TL76N137, TLP521 等组成。这是一个典型的 RS-422/485 通信接口电路。当所有的 RS-422/485 选择开关断开时,则可用于执行全双工的 RS-422 工作;当所有的 RS-422/485 选择开关闭合时,则可用于半双工的 RS-485 工作。

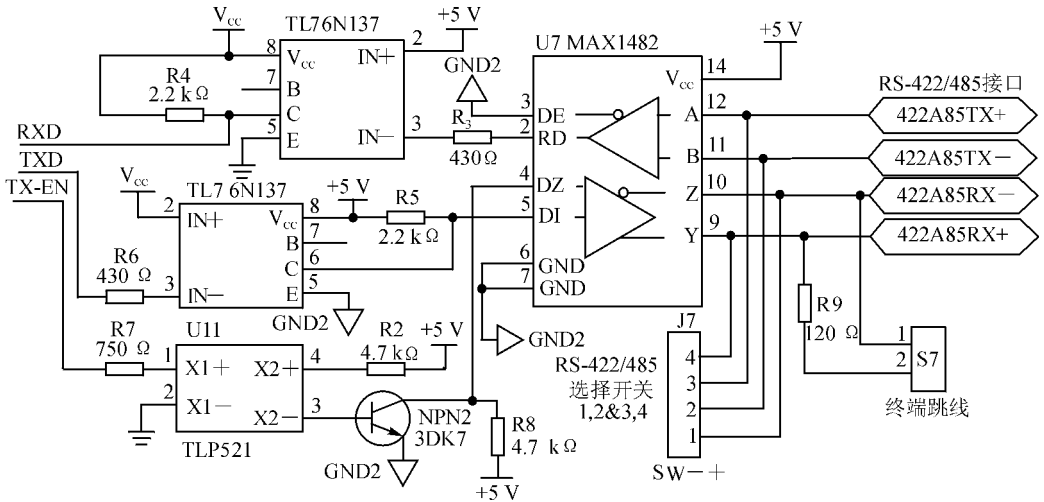


图 10-22 RS-422/485 接口电路

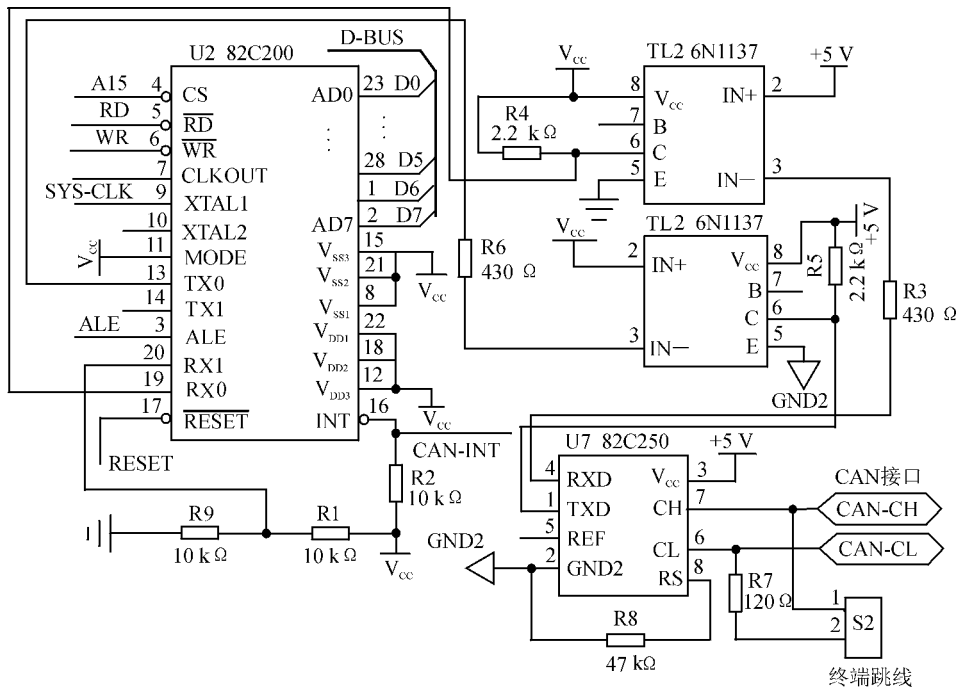
RS-422/485 接口电路是通过 AT89C51 内部的 UART 控制器进行通信的,并且在 AT89C51 的 P1 端口中,P1.5 端口用于控制发送的启动端 TX-EN。RS-422/485 接口电路中的 RXD, TXD 分别和 AT89C51 的 RXD, TXD 连接。

MAX1482 集成电路用于实现 TTL/RS-422 的电平转换。它具有多达 256 个网络节点的收发能力。它的特点就是可取代传统的需用 MC3486 和 MC3487 两片集成电路才能完成的工作。除此之外,由于它的功耗低,体积小,提高了系统的可靠性。

5. CAN 接口电路

CAN 总线之所以在国内乃至全球范围内一直保持高速发展,完全取决于其卓越的性能。最早 CAN 总线主要应用在汽车领域,随着集成工艺的发展和成本的降低,被越来越多的控制领域内的软硬件工程师所接受,广泛地应用于各行各业和控制局域网中。CAN 总线主要性能指标为:多主站仲裁结构(分地址优先级,非破坏方式仲裁),支持主从或广播方式,不加任何扩展最多 110 个节点,最高通信速率 1 Msp,最远通信距离 5 km。通常 CAN 控制器内部设有接收和发送缓冲区,通信以帧为单位,最多 8 个字节的数据,硬件自动进行 16 位 CRC 校验,而且具有极强的总线和通信错误的管理能力。

图 10-23 所示的 CAN 接口电路中采用了 PHILIPS 公司生产的 CAN 总线控制器和 CAN 接口芯片,它们分别为 82C200(引脚兼容的升级版本为 SJC1000)和 82C250。



10.5.3 软件简介

软件设计采用模块化结构,依据各个功能部分进行模块划分,大致划分为:总线适配器内部参数初始化;主循环程序;RS-232,RS-422/485 和 CAN 接口初始化;各总线网络、各网络节点接收和发送缓冲区初始化;RS-232,RS-422/485 和 CAN 接口对应的中断服务程序;通信信息数据包的命令解释和处理程序;RS-232,RS-422/485 和 CAN 接口对应于通信信息包的装载和卸载程序;RS-232,RS-422/485 和 CAN 接口对应的出错处理程序。

为便于网络命令和数据处理,现场总线适配器采用统一的数据包格式。它们是:源逻辑地址(2 字节)、目标逻辑地址(2 字节)、命令(1 字节)、数据长度(1 字节)、数据、数据累加校验和(2 字节)。其中源地址和目标地址定义基本相同,各位定义如下:

- B15:代表主从方式或广播方式;
B14:代表主机或从机;
B13,B12:代表第几个 RS-232 网络区间(1~3,0 代表不是该网络地址,以下定义相同);
B11,B10:代表第几个 RS-422/485 网络区间;
B9,B8:代表第几个 CAN 网络区间;
B7~B0:对应网络区间下的物理地址(同一个网络区间下最多允许 256 个网络节点)。

参考文献

- 1 余永权. Flash 单片机原理及应用. 北京:电子工业出版社,1997
- 2 ATMEL Corporation, Microcontroller Data Book. Oct 1995
- 3 ATMEL Corporation, Nonvolatile Memory Data Book. May 1996
- 4 何立民. 单片机应用技术选编(5). 北京:北京航空航天大学出版社,1997
- 5 何立民. 单片机应用技术选编(6). 北京:北京航空航天大学出版社,1998
- 6 王长龙等. 串行输入显示驱动芯片 MC14499 在数字血压计中的应用. 浙江大学学报,1998(12):697~700
- 7 黄采伦,夏新军. 一种基于单片微机的可控硅智能控制模块. 自动化与仪器仪表,1999(6)
- 8 张克臣. 新型晶闸管触发模块的应用. 电子世界,1999(8)
- 9 沈飙,夏海燕. 运用 Visual Basic 实现 PC 与 89C51 单片机之间的串行通信. 计算机自动测量与控制,1999(2)
- 10 杜开初. 用单片机实现一次积分型 A/D 和电容的测量. 浙江大学学报,1998(12)
- 11 大连东方电脑股份有限公司,大连东方显示器材公司. EDM12864B 图形点阵式液晶显示器模块原理与应用手册.
- 12 刘兴瑜,文生平. 串行接口器件的单片机直接编程设计. 广东自动化与信息工程,2001(1):14~16
- 13 克强. 用 AT89C2051 单片机制作洗衣机控制电路. 电子世界,2001(3):35,39~40
- 14 徐欣,樊昀. 一种由单片机实现的多功能密码锁. 计算机应用研究,1998(3)
- 15 孟祥增,刘瑞梅. 一种用电力网连接的子母钟. 仪表技术,1998(3)
- 16 王玫. MCU 电话语音报警器设计. 电子技术,1998(3)
- 17 王传林. 介绍一款智能型充电器的制作. 浙江大学学报,1998(12):788~791
- 18 严云洋,胡光. 数控二维步进电机运行系统. 浙江大学学报,1998(12):957~961
- 19 曹龙汉. 一种单片机数控交流调压器. 自动化与仪器仪表,1999(5)
- 20 李海亮. 基于 AT89C2051 单片机的 22 位地震数据采集系统. 浙江大学学报,1998(12):202~207
- 21 张运楚,齐保良,邹军. 一种环保设备运行记录仪的研制. 电子技术应用,2001(4):39~41
- 22 彭宣戈. 用智能卡存储水位数据的水位记录仪. 1996 年全国单片机学术交流会论文集,1996. 247~250
- 23 伍运霞,苏又平,刘昌振. 氧化铝膜状传感式在线湿度仪的研制. 电子技术应用,2001(4):35~36
- 24 陈国旺,王军,叶经方. 可实现预收费的智能煤气表研制. 浙江大学学报,1998(12):132~135
- 25 高振东,邱丹,王东. IC 卡电度表复位中断计数方法. 浙江大学学报,1998(12):675~677
- 26 滕进世. 用 89C2051 的小范围寻呼系统低电压数字寻呼. 电子世界,2001(2):38~40
- 27 彭伟军,宋文涛,罗汉文. 基于单片机的 Reed - Solomon 编解码器的算法与实现. 浙江大学学报,1998(12):452~457
- 28 瞿雷. 单片机实现传真通信控制. 浙江大学学报,1998(12):306~310
- 29 易茂祥,丁志中,戴礼荣. 程控交换机呼叫信息缓冲系统的设计. 电子技术应用,2001(4):47~49
- 30 刘武光. 现场总线适配器的软硬件设计. 电子技术应用,1999(8)

附 录

ATMEL8051 系列单片机选型指南

ATMEL8051 系列单片机选型指南如附录表 1、附录表 2 所列。

附录表 1 ATMEL8051 系列单片机选型指南一

	Program Memory /KB	EEPROM / 字节	RAM / 字节	I/O Pins	UART	Timers	Other Features	V _{cc} /V	(5V/3V) Max speed /MHz	Packages
ISP FLASH										
AT89S51	4		128	32	1	2	WDT	4.0-5.5	33	PDIP40,PLCC44,TQFP44
AT89LS51	4		128	32	1	2	WDT	2.7-5.5	16	PDIP40,PLCC44,TQFP44
AT89S52	8		256	32	1	3	WDT	4.0-5.5	33	PDIP40,PLCC44,TQFP44
AT89LS52	8		256	32	1	3	WDT	2.7-5.5	33	PDIP40,PLCC44,TQFP44
AT89S8252	8	2K	256	32	1	3	WDT	3.0-6.0	12	PDIP40,PLCC44,TQFP44
AT89S53	12		256	32	1	3	WDT	3.0-6.0	24	PDIP40,PLCC44,TQFP44
T89C5115	16	2K	512	20	1	3+PCA	WDT,A/D	4.5-5.5	40	SOIC28,PLCC28,VQFP32,SOIC24*
T89C5115	16	2K	512	20	1	3+PCA	WDT,A/D	2.7-3.6	24	SOIC28,PLCC28,VQFP32,SOIC24*
T89C51RB2	16		1 280	32	1	3+PCA	WDT,SPI,BRG,KBD	4.5-5.5	40	PDIP40,PLCC44,VQFP44
T89C51RB2	16		1 280	32	1	3+PCA	WDT,SPI,BRG,KBD	2.7-3.3	30	PDIP40,PLCC44,VQFP44
T89C51RC2	32		1 280	32	1	3+PCA	WDT,SPI,BRG,KBD	4.5-5.5	40	PDIP40,PLCC44,VQFP44
T89C51RC2	32		1 280	32	1	3+PCA	WDT,SPI,BRG,KBD	2.7-3.3	30	PDIP40,PLCC44,VQFP44
T89C51IC2	32		1 280	34	1	3+PCA	WDT,SPI,I2C,RTC,BRG,KBD	4.5-5.5	40	PLCC44,VQFP44
T89C51IC2	32		1 280	34	1	3+PCA	WDT,SPI,I2C,RTC,BRG,KBD	2.7-3.3	30	PLCC44,VQFP44
T89C51AC2	32	2K	1 280	34	1	3+PCA	WDT,A/D	4.5-5.5	40	PLCC44,VQFP44,CA-BGA64*
T89C51AC2	32	2K	1 280	34	1	3+PCA	WDT,A/D	2.7-3.6	20	PLCC44,VQFP44,CA-BGA64*
T89C51RD2	64	2K	1 280	32/48	1	3+PCA	WDT	4.5-5.5	40	PDIP40,PLCC44,VQFP44,PLCC68,VQFP64
T89C51RD2	64	2K	1 280	32/48	1	3+PCA	WDT	2.7-3.6	25	PDIP40,PLCC44,VQFP44,PLCC68,VQFP64
FLASH										
AT89C1051U	1		64	14	1	2		2.7-6.0	24/12	PDIP20,SOIC 20
AT89C2051	2		128	14	1	2		2.7-6.0	24/12	PDIP20,SOIC 20
AT89C4051	4		128	14	1	2		2.7-6.0	24/12	PDIP20,SOIC 20
AT89C51	4		128	32	1	2		4.0-6.0	33	PDIP40,PLCC44,TQFP44
AT89LV51	4		128	32	1	2		2.7-6.0	16	PDIP40,PLCC44,TQFP44
AT89C52	8		256	32	1	3		4.0-6.0	33	PDIP40,PLCC44,TQFP44
AT89LV52	8		256	32	1	3		2.7-6.0	16	PDIP40,PLCC44,TQFP44
AT89C55WD	20		256	32	1	3	WDT	3.0-5.5	33	PDIP40,PLCC44,TQFP44
AT89LV55WD	20		256	32	1	3	WDT	2.7-5.5	12	PDIP40,PLCC44,TQFP44
AT89LV55	20		256	32	1	3		2.7-5.5	12	PDIP40,PLCC44,TQFP44
AT89C51RC	32		512	32	1	3	WDT	3.0-5.5	33	PDIP40,PLCC44,TQFP44
AT89LV51RC	32		512	32	1	3	WDT	2.7-5.5	12	PDIP40,PLCC44,TQFP44
OTP										
T87C5111	4		256	21	1	2+PCA	WDT,A/D,RC osc.,Prog.I/O,SPI	4.5-5.5	66	SSOP24,SOIC24,DIL24
T87C5111	4		256	21	1	2+PCA	WDT,A/D,RC osc.,Prog.I/O,SPI	2.7-5.5	40	SSOP24,SOIC24,DIL24
T87C5112	8		256	40	1	2+PCA	WDT,A/D,RC osc.,Prog.I/O,SPI	4.5-5.5	66	LQFP48,PLCC52
T87C5112	8		256	40	1	2+PCA	WDT,A/D,RC osc.,Prog.I/O,SPI	2.7-5.5	40	LQFP48,PLCC52
TS87C52X2	8		256	32	1	3		4.5-5.5	60	PDIP40,PLC44,VQF44
TS87C52X2	8		256	32	1	3		2.7-5.5	40	PDIP40,PLC44,VQF44
AT87C5103	12		512	19		2+PCA	SPI	2.7-5.5	16	SSOP16,SOIC24
TS87C54X2	16		256	32	1	3	WDT	4.5-5.5	60	PDIP40,PLC44,VQFP44
TS87C54X2	16		256	32	1	3	WDT	2.7-5.5	40	PDIP40,PLC44,VQFP44
TS87C51U2	16		256	32	2	3	WDT,BRG	4.5-5.5	60	PDIP40,PLC44,VQFP44
TS87C51U2	16		256	32	2	3	WDT,BRG	2.7-5.5	40	PDIP40,PLC44,VQFP44
T87C5101	16		512	22	1	3		4.5-5.5	66	SOIC24,SOIC28,SSOP24,PDIL24
T87C5101	16		512	22	1	3		2.7-5.5	40	SOIC24,SOIC28,SSOP24,PDIL24
TS87C51RB2	16		512	32	1	3+PCA	WDT	4.5-5.5	60	PDIP40,PLCC44,VQFP44

续附表 1

	Program Memory / KB	EEPROM / 字节	RAM / 字节	I/O Pins	UART	Timers	Other Features	V _{cc} / V	(5V/3V) Max speed / MHz	Packages
TS87C51RB2	16		512	32	1	3+PCA	WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44
AT87F55WD	20		256	32	1	3	WDT	3.0-5.5	33	PDIP40, PLCC44, VQFP44
OTP										
TS87C58X2	32		256	32	1	3	WDT	4.5-5.5	60	PDIP40, PLCC44, VQFP44
TS87C58X2	32		256	32	1	3	WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44
AT87F51RC	32		512	32	1	3	WDT	3.0-5.5	33	PDIP40, PLCC44, VQFP44
TS87C51RC2	32		512	32	1	3+PCA	WDT	4.5-5.5	60	PDIP40, PLCC44, VQFP44
TS87C51RC2	32		512	32	1	3+PCA	WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44
TSC87251G2D	32	1 024	32	1	3+EWC		C251 core, SPI, I2C	4.5-5.5	24	PDIP40, PLCC44, VQFP44
TSC87251G2D	32	1 024	32	1	3+EWC		C251 core, SPI, I2C	2.7-5.5	16	PDIP40, PLCC44, VQFP44
TS87C51RD2	64	1 024	32/48	1	3+PCA		WDT	4.5-5.5	60	PDIP40, PLCC44, VQFP44, PLCC68, VQFP64
TS87C51RD2	64	1 024	32/48	1	3+PCA		WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44, PLCC68, VQFP64
ROM										
T83C5111	4		256	21	1	2+PCA	WDT, A/D	4.5-5.5	66	SSOP24, SOIC24, DIL24
T83C5111	4		256	21	1	2+PCA	WDT, A/D	2.7-5.5	40	SSOP24, SOIC24, DIL24
T83C5112	8		256	40	1	2+PCA	WDT, A/D	4.5-5.5	66	LQFP48, PLCC52
T83C5112	8		256	40	1	2+PCA	WDT, A/D	2.7-5.5	40	LQFP48, PLCC52
TS80C52X2	8		256	32	1	3		4.5-5.5	60	PDIL40, PLC44, VQFP44
TS80C52X2	8		256	32	1	3		2.7-5.5	40	PDIL40, PLC44, VQFP44
T83C5102	8		512	22	1	3		4.5-5.5	66	SOIC24, SOIC28, SSOP24, PDIL24
T83C5102	8		512	22	1	3		2.7-5.5	40	SOIC24, SOIC28, SSOP24, PDIL24
AT83C5103	12		512	19		2+PCA	SPI	2.7-5.5	16	SSOP16, SOIC24
TS80C54X2	16		256	32	1	3	WDT	4.5-5.5	60	PDIL40, PLC44, VQFP44
TS80C54X2	16		256	32	1	3	WDT	2.7-5.5	40	PDIL40, PLC44, VQFP44
TS83C51U2	16		256	32	2	3	WDT, BRG	4.5-5.5	60	PDIL40, PLC44, VQFP44
TS80C51U2	16		256	32	2	3	WDT, BRG	2.7-5.5	40	PDIL40, PLC44, VQFP44
T83C5101	16		512	22	1	3		4.5-5.5	66	SOIC24, SOIC28, SSOP24, PDIL24
T83C5101	16		512	22	1	3		2.7-5.5	40	SOIC24, SOIC28, SSOP24, PDIL24
TS83C51RB2	16		512	32	1	3+PCA	WDT	4.5-5.5	60	PDIP40, PLCC44, VQFP44
TS83C51RB2	16		512	32	1	3+PCA	WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44
TS83C51RB2	16	1 280	32	1	3+PCA		WDT, SPI, BRG, KBD	4.5-5.5	40	PDIP40, PLCC44, VQFP44
TS83C51RB2	16	1 280	32	1	3+PCA		WDT, SPI, BRG, KBD	2.7-3.6	30	PDIP40, PLCC44, VQFP44
T83C51IB2	16	1 280	34	1	3+PCA		WDT, SPI, I2C, RTC, BRG, KBD	4.5-5.5	40	PLCC44, VQFP44
T83C51IB2	16	1 280	34	1	3+PCA		WDT, SPI, I2C, RTC, BRG, KBD	2.7-3.6	30	PLCC44, VQFP44
TSC83251G1D	16	1 024	32	1	3+EWC		C251 core, SPI, I2C	4.5-5.5	24	PDIL40, PLC44, VQFP44
TSC83251G1D	16	1 024	32	1	3+EWC		C251 core, SPI, I2C	2.7-5.5	12	PDIL40, PLC44, VQFP44
TS80C58X2	32		256	32	1	3	WDT	4.5-5.5	60	PDIL40, PLC44, VQFP44
TS80C58X2	32		256	32	1	3	WDT	2.7-5.5	40	PDIL40, PLC44, VQFP44
TS83C51RC2	32		512	32	1	3+PCA	WDT	4.5-5.5	60	PDIP40, PLCC44, VQFP44
TS83C51RC2	32		512	32	1	3+PCA	WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44
T83C51RC2	32		1 280	32	1	3+PCA	WDT, SPI, BRG, KBD	4.5-5.5	40	PDIP40, PLCC44, VQFP44
T83C51RC2	32		1 280	32	1	3+PCA	WDT, SPI, BRG, KBD	2.7-3.6	30	PDIP40, PLCC44, VQFP44
T83C51IC2	32		1 280	34	1	3+PCA	WDT, SPI, I2C, RTC, BRG, KBD	4.5-5.5	40	PLCC44, VQFP44
T83C51IC2	32		1 280	34	1	3+PCA	WDT, SPI, I2C, RTC, BRG, KBD	2.7-3.6	30	PLCC44, VQFP44
TSC83251G2D	32	1 024	32	1	3+EWC		C251 core, SPI, I2C	4.5-5.5	24	PDIL40, PLC44, VQFP44
TSC83251G2D	32	1 024	32	1	3+EWC		C251 core, SPI, I2C	2.7-5.5	16	PDIL40, PLC44, VQFP44
TS83C51RD2	64	1 024	32/48	1	3+PCA		WDT	4.5-5.5	60	PDIP40, PLCC44, VQFP44, PLCC68, VQFP64
TS83C51RD2	64	1 024	32/48	1	3+PCA		WDT	2.7-5.5	40	PDIP40, PLCC44, VQFP44, PLCC68, VQFP64
ROMless										
TS80C31X2			128	32	1	2		4.5-5.5	60	PDIL40, PLCC44, VQFP44
TS80C31X2			128	32	1	2		2.7-5.5	40	PDIL40, PLCC44, VQFP44
TS80C32X2			256	32	1	3		4.5-5.5	60	PDIL40, PLCC44, VQFP44
TS80C32X2			256	32	1	3		2.7-5.5	40	PDIL40, PLCC44, VQFP44
TS80C51U2			256	32	2	3	WDT, BRG	4.5-5.5	60	PDIL40, PLCC44, VQFP44
TS80C51U2			256	32	2	3	WDT, BRG	2.7-5.5	40	PDIL40, PLCC44, VQFP44
T80C5112			256	40	1	2+PCA	WDT, SPI	4.5-5.5	66	LQFP48, PLCC52
T80C5112			256	40	1	2+PCA	WDT, SPI	2.7-5.5	40	LQFP48, PLCC52
TS80C51RA2			512	32	1	3+PCA	WDT	4.5-5.5	60	PDIL40, PLCC44, VQFP44
TS80C51RA2			512	32	1	3+PCA	WDT	2.7-5.5	40	PDIL40, PLCC44, VQFP44
TS80C51RD2			1 024	32/48	1	3+PCA	WDT	4.5-5.5	60	PDIL40, PLC44, VQFP44, PLCC68, VQFP64
TS80C51RD2			1 024	32/48	1	3+PCA	WDT	2.7-5.5	40	PDIL40, PLC44, VQFP44, PLCC68, VQFP64
TSC80251G2D			1 024	32	1	3+EWC	C251 core, SPI, I2C, WDT	4.5-5.5	24	PDIL40, PLC44, VQFP44
TSC80251G2D			1 024	32	1	3+EWC	C251 core, SPI, I2C, WDT	2.7-5.5	16	PDIL40, PLCC44, VQFP44
T80C51ID2			1 280	32/34	1	3+PCA	WDT, SPI, BRG, KBD	4.5-5.5	40	PDIL40, PLCC44, VQFP44
T80C51ID2			1 280	32/34	1	3+PCA	WDT, SPI, BRG, KBD	4.5-5.5	40	PDIL40, PLCC44, VQFP44

附录表 2 ATME8051 系列单片机选型指南二

	Program Memory /KB					EEPROM /KB	RAM / 字节	I/O Pins	UART	TIMERS	Other Features	V _{cc} / V	(5V/3V) Max speed /MHz	Packages
	FLASH	OTP	CRAM	ROM	Bootloader									
MP3 Decoder														
AT89C51SND1A	64				4		2 304	44	1	2	MP3 Decoder, 12C, MMC, ADC, USB, SPI, I2S	2. 7-3. 6	20	TQFP80, DIE
AT83SND1Axxx				64			2 304	44	1	2	MP3 Decoder, 12C, MMC, ADC, USB, SPI, I2S	2. 7-3. 6	20	TQFP80, DIE
Data Acquisition														
AT89S4D12	4					128	256	5			SPI	3. 0-3. 6	12	PLCC28, SO28, PLCC32
Card Reader														
T83C5121				16			512	14/30	1	2	ISO7816, DC/DC, WTD	2. 85-5. 5	16	SSOP24, PLCC52, LQFP48 *
T85C5121			16		16		512	14/30	1	2	ISO7816, DC/DC, WTD	2. 85-5. 5	16	SSOP24, PLCC52, LQFP48 *
T89C5121	16				16		512	14/30	1	2	ISO7816, DC/DC, WTD	2. 85-5. 5	16	SSOP24, LQFP48 *
CAN Multiplexing														
T89C51CC02UA	16				2	2	512	20	1	3+PCA	A/D, CAN 4 chan. , WTD UART Bootloader	2. 7-5. 5	40/24	SOIC24 * , SOIC28, PLCC28, VQFP32
T89C51CC02CA	16				2	2	512	20	1	3+PCA	A/D, CAN 4 chan. , WTD CAN Bootloader	2. 7-5. 5	40/24	SOIC24 * , SOIC28, PLCC28, VQFP32
T89C51CC01UA	32				2	2	1 280	34	1	3+PCA	A/D, CAN 15 chan. , WTD UART Bootloader	2. 7-5. 5	40/20	VQFP44, PLCC44, CA-BGA64 *
T89C51CC01CA	32				2	2	1 280	34	1	3+PCA	A/D, CAN 15 chan. , WTD CAN Bootloader	2. 7-5. 5	40/20	VQFP44, PLCC44, CA-BGA64 *

* Check for availability.

ATMEL 公司代表处及代理商名录

美国爱特梅尔股份有限公司

地址：香港九龙尖沙咀东部麼地道 77 号华懋广场 1219 室

电话：852-27219778 传真：852-27221369

网址：www.atmel.com

爱特梅尔北京代表处

地址：北京市海淀区学院南路 70 号 710 室（邮编：100081）

电话：010-62186224, 62186225, 62180478 传真：010-62186227

爱特梅尔上海代表处

地址：上海市长宁区番禺路 586 号东方商务大楼 4 楼 A 座（邮编：200052）

电话：021-62809234, 62808604, 62808605 传真：021-62807592

北天星国际有限公司

地址：香港九龙湾宏开道 16 号德福大厦 1801 室

电话：852-23455777 传真：852-23452912

网址：www.po-star.com E-mail: postar@netvigator.com

北京分公司

地址：北京市海淀区海淀南路 15 号海南饭店写字楼 405 号

电话：010-62655730/1/2 传真：010-62655733

E-mail: keweiz@263.net

深圳分公司

地址：广东省深圳市福田区华强北路 2006 号华联发大厦 907 室

电话：0755-3998862 内 传真：0755-3998555

E-mail: L33@po-star.com

福州分公司

地址：福建省福州市台江区广达 438 号双丰大厦 9H

电话：0591-3265210 传真：0591-3280950

E-mail: btx888@public.fz.fj.cn

上海分公司

地址：上海市复兴中路 1 号沪能大厦 1208 室

电话：021-63900603 传真：021-63900865

E-mail: shbtx@ONLINE.SH.COM

武汉分公司

电话：027-87322883 传真：027-87322883

E-mail: whbtx@postar.com

深圳市威立姆电子有限公司

地址：广东省深圳市福田区泰然工贸园苍松大厦 1806 室（邮编：518084）

电话：0755-3849169 传真：0755-3849170

网址：www.william.com.cn E-mail: sales@william.com.cn

北京威立姆公司

地址：北京市海淀区学院南路 70 号 718 室（邮编：100081）

电话：010-62175225/6 传真：010-62175224

南京威立姆科技有限公司

地址：南京市瑞金路 48 号瑞金大厦 1701 室（邮编：210007）

电话：025-4485918, 4611078, 4612118, 4619698, 4619898 传真：025-4599598

E-mail: williamn@public1.pptt.js.cn

成都爱梅尔电子有限责任公司

地址：四川省成都市一环路南二段天歌科技大厦 817 室（邮编：610041）

电话及传真：028-5441471, 5445330, 5445404

E-mail: robin.zhang@elong.com

广州市天河双龙电子有限公司

地址：广州市天河路 561 号龙苑大厦 A3 座新赛格 331 室（邮编：510630）

电话：020-85510191, 87578852, 87505012 传真：620 分机

网址：www.sl.com.cn

广州市天河双龙电子有限公司北京分公司

地址：北京市海淀区知春路 132 号中发大厦 616 室（邮编：100086）

电话：010-82623550, 82623551, 62653785 传真：010-62653785

广州市天河双龙电子有限公司上海分公司

地址：上海市北京东路 668 号赛格市场 1A06（邮编：200001）

电话：021-53081501 传真：021-53081502

百特(集团)电子有限公司

电话：852-27939783 传真：852-27939851

网址：www.baite-flash.com E-mail: hkbaite@netvigator.com

深圳公司

地址：广东省深圳市福田区福虹路世贸广场 A 座 1603 室（邮编：518033）

电话：0755-3679100 传真：0755-3679072

E-mail: Baite168@public.szptt.net.cn

北京百特明瑞电子有限公司

地址：北京市海淀区海淀南路 30 号航天长城大厦 1311 室（邮编：100080）

电话：010-68748487, 68748497 传真：010-68748408

E-mail: Bj-better@china.com

上海公司

地址：上海市淮海西路 55 号申通信息广场 221 座

电话：021-52989261/2/3/4 传真：021-52988260

E-mail: Shflash@online.sh.cn

北京欣润达电子有限公司

地址：北京市海淀区中关村中路 123 号都市网景 B2302 室

电话：010-62121813, 62121392, 62510201~5 传真：010-62121392

北京欣润达电子有限公司深圳办事处

地址：深圳市福田区福明路 40 号都市 E 站 12A10 室

电话：0755-3995160/1/2 传真：0755-3995110

北京航空航天大学出版社单片机与嵌入式系统图书总汇

书 名	作 者	定价
嵌入式系统原理及应用开发技术	桑 楠	23.0
ARM 嵌入式处理器结构与应用基础(含光盘)	马忠梅	36.0
DSP 技术丛书		
DSP 基本理论与应用技术	李哲英	即将出版
DSP 基础与应用系统设计	王念旭	62.0
TMS320C54x DSP 结构、原理及应用	戴明桢	25.0
TMS320C54x DSP 应用程序设计与开发	刘益成	即将出版
TMS320C54x DSP 应用系统设计	郑 红	即将出版
TMS320LF240x DSP 结构、原理及应用(含光盘)	刘和平	36.0
TMS320F206 DSP 结构、原理及应用	李 刚	即将出版
单片机教材系列		
单片机实验与实践教程(二)	夏继强	17.0
单片机初级教程——单片机基础	张迎新	24.0
单片机中级教程——单片机原理及应用	张俊敏	23.0
单片机高级教程——单片机应用与设计	何立民	28.5
单片机实用教程(高校计算机教学系列教材)	李 勋	17.5
MCS-51 系列单片机应用系统设计	何立民	35.0
单片机微型计算机大学读本	李 勋	18.0
单片机基础	李广弟	20.0
单片机基础(修订版)	李广弟	29.0
单片机原理及接口技术(简明修订版)	李朝青	26.0
单片机的 C 语言应用程序设计(修订版)	马忠梅	24.5
MCS-51/96 系列单片机原理及应用(修订版)	孙涵芳	35.0
单片机应用技术选编系列		
单片机应用技术选编(1)~(8)	何立民	
(1)~(8)的价格分别为:46.0、32.0、52.0、48.0、49.0、52.0、55.0、59.5		
单片机外围器件实用手册系列		
单片机外围器件实用手册——电源器件分册	关德新	57.0
单片机外围器件实用手册——存储器分册	窦振中	55.0
单片机外围器件实用手册——数据传输接口器件分册	邬宽明	54.0
单片机外围器件实用手册——输入通道器件分册	纪宗南	48.0
单片机应用技术丛书		
单片机应用系统的功率接口技术	余永权	10.8
低功耗单片机微机系统设计	何为民	9.5
I ² C 总线应用系统设计	何立民	15.5
单片机的 C 语言应用程序设计	马忠梅	23.5
单片机程序设计基础	周航慈	23.5
单片机应用系统抗干扰技术	王幸之	38.5
PIC 单片机系列		
PIC 系列单片机原理和程序设计	窦振中	29.0
PIC 系列单片机应用设计与实例	窦振中	45.0
PIC 系列单片机的开发应用技术	武 锋	23.0
PIC 系列单片机接口技术及应用系统设计	王有绪	36.0
PIC16F87x 数据手册——28/40 脚 8 位 FLASH 单片机	刘和平	22.0
PIC16F87x 单片机实用软件与接口技术——汇编语言及其应用(含光盘)	刘和平	39.0
PIC16F87x 单片机实用软件与接口技术——C 语言及其应用(含光盘)	刘和平	32.0
PIC 单片机实用教程——基础篇	李学海	29.5
PIC 单片机实用教程——提高篇	李学海	即将出版
Intel 公司 MCS-51/96/196 单片机系列		
MCS-51/196 单片机浮点程序和实用程序	张克彦	21.0
MCS-51 系列单片机接口电路与应用程序实例	沈德金	13.0
单片机微型计算机及其应用	徐爱卿	4.0
单片机接口技术实验指导	沈德金	9.5
MCS-51/96 系列单片机原理及应用	孙涵芳	16.0
MCS-51 系列单片机实用接口技术	李 华	40.0

书 名	作 者	定价
MCS-51/96 系列单片机原理及应用(修订版)	孙涵芳	35.0
Intel 16 位单片机	孙涵芳	53.0
MCS-51 系列单片机应用系统设计	何立民	35.0
MOTOROLA 单片机系列		
Motorola 单片机及专用集成芯片应用系统设计	陈粤初	29.0
MC68332 单片机结构与应用	张 宁	28.0
Motorola M68HC05 系列单片机及其应用基础	刚寒冰	25.0
Motorola J-K 系列廉价单片机原理及应用	余永权	19.5
Motorola 单片机应用技术	张 辉	15.0
Motorola 单片机原理及应用系统开发技术	谢宋和	26.0
PHILIPS 单片机系列		
80C51 系列微处理器系统原理、功能集成与应用	邬宽明	17.5
80C51 系列小型微控制器原理、应用和开发	邬宽明	19.0
80C51XA 十六位微控制器系统设计、器件和应用	邬宽明	27.5
PHILIPS 51LPC 系列单片机原理及应用设计	周航慈	19.0
EPSON 单片机系列		
EPSON 单片机原理与应用	宋建国	33.0
EPSON 8 位单片机原理与应用	薛宗祥	20.0
EPSON 图形液晶显示控制器系列 SED135X 和 SED1374	郭 强	20.0
EPSON 单片系列液晶显示器及其应用	李维谔	20.0
ATMEL 单片机系列		
AVR 单片机原理及应用	宋建国	28.0
AVR 高速嵌入式单片机原理与应用	耿德根	40.0
AVR 单片机应用技术	李 勋	23.0
AVR 单片机应用设计	丁化成	22.0
ATMEL 89 系列单片机应用技术	余永权	32.0
其它公司单片机系列		
与 8051 兼容的高性能、高速单片机—C8051Fxxx	李 刚	34.0
C8051Fxxx 高速 SOC 单片机原理及应用	潘琢金	27.0
EM78447B 单片机应用研究与制作(含光盘)	黄 勇	23.0
ADuC8xx 系列单片机原理与应用技术	李 刚	18.0
亿恒(西门子)C166 系列 16 位单片机原理与开发(含光盘)	程 军	40.0
MPS430 系列 FLASH 型超低功耗 16 位单片机	胡大可	30.0
MSP430 系列超低功耗 16 位单片机原理与应用	胡大可	29.5
三菱 3850 系列单片机原理和应用	陆延丰	25.0
三菱 M16C/62 单片机原理与应用	陆延丰	39.0
东芝单片机原理及其在家用电器中的应用	谌清平	29.5
日立 H8/3048 系列单片机应用技术	李 勋	29.5
青少年单片机系列丛书		
青少年机器人初级教程——大空穿接机 LOGO、QBASIC 语言编程	卢燕林	10.0
少儿单片机工程师	盛焕鸣	7.0
DP801 单片机系统入门	车金相	7.0
当代美猴王——少年单片机应用与开发	胥筱汀	17.0
单片机模糊技术应用选编系列		
模糊技术与神经网络技术选编(1)~(5)	刘增良	
(1)~(5)的价格分别为:50.0、56.0、50.0、50.0、50.0		
计算机模糊控制原理及应用	戎月莉	7.5
模糊信息优化处理技术及其应用	黄崇福	16.0
模糊专家系统原理与设计	刘有才	18.0
单片机模糊逻辑开发软件	余永权	16.8
模糊逻辑控制技术及应用	窦振中	23.0
模糊逻辑与神经网络——理论研究与探索	刘增良	33.0
单片机模糊逻辑与模糊控制	余永权	40.0
神经模糊系统及其应用	王士同	28.0
模糊控制技术与模糊家用电器	余永权	23.0
单片机综合类		
电磁兼容性原理设计及预测技术	蔡仁刚	23.0
传感器接口与检测仪器电路(重印)	吕俊芳	12.0
单片机微型计算机应用和开发系统	徐爱卿	12.0
单片机应用程序设计技术	周航慈	7.3

书 名	作 者	定价
单片机系统微机化开发常用软件(第一辑)	盛焕鸣	6.0
单片机应用系统设计与实践	陈粤初	17.0
单片机总线扩展技术	尤一鸣	10.5
美国国家半导体公司 HPC46003/46083 16 位单片机原理及应用	陈泽文	12.0
DEC Alpha 21064 微处理器结构与应用	张 宁	19.2
8098 单片机原理及实用接口技术	李杏春	38.0
CAN 总线原理和应用系统设计	邬宽明	25.0
嵌入式单片机 8XC251 用户指南	王 勇	22.0
单片机现场可编程外围芯片 PSD 的原理及应用	孙涵芳	39.5
单片机人机接口实例集	公茂法	17.0
智能化测量控制系统仪表原理与设计	徐爱钧	32.0
单片机测控系统设计大全	王福瑞	38.0
从 C 到嵌入式 C 编程语言——入门、实用、深入	梁合庆	35.0

书 名	作 者	定价
Intel 80C196 单片机应用实践与 C 语言开发	程 军	32.0
PC 机及单片机数据通信技术	李朝青	28.0
《单片机与嵌入式系统应用》杂志 2001 年(1~6)合订本		42.0
《单片机与嵌入式系统应用》杂志 2001 年(7~12)合订本		42.0
单片机 C 高级语言程序设计及其应用	袁 涛	18.0
2001 嵌入式及单片机国际学术交流会议论文集集(中文)	沈绪榜 何立民	150
从神经芯片到控制网络	凌志浩	21.0
计算机语音技术(修订版)	朱民雄	37.0
电动机的单片机控制	王晓明	24.0
单片机 & DSP 外围——数字 IC 技术手册(上册)	李朝青	即将出版
单片机 & DSP 外围——数字 IC 技术手册(下册)	李朝青	即将出版

其它计算机硬件、网络技术、电子工程和 EDA 等

书 名	作 者	定价
计算机外部设备及网络设备	宁 章	35.0
计算机外设与接口简明教程	袁新燕	19.0
最新计算机硬件推介及选购指南	陆 水	24.0
微机原理及接口技术 (自学文凭考试指定教材)	邵鸿余	30.0
微型计算机硬件接口实用编程	孙文杰	35.0
电子计算机外部设备	张 宁	21.0
计算机词典——硬件探秘	第三波	23.0
计算机控制系统——理论、设计与实现	高金源	25.0
计算机控制系统基础——计算机教程	陈炳和	30.0
计算机组成原理与系统结构	蒋本珊	26.0
微型计算机技术	田 辉	24.0
网络技术教程	郭艳玲	28.0
计算机网络原理与应用	肖永生	18.0
计算机网络技术与网络集成应用	马 莉	26.0
计算机网络实用教程	郭秋萍	24.0
计算机网络概念、原理、技术及应用(网络大全)	朱稼兴	49.0
Windows NT 中 Intranet 组建和管理技术	李明柱	22.0
网络及 Internet Explore 5.0 实用指南	朱宏伟	32.0

书 名	作 者	定价
计算机系统及网络维护教程	龚 兵	18.5
光纤通信及网络技术	徐宝强	19.0
千兆位、千兆位计算机网络技术	余镇危	16.0
电路基础与模拟电子技术	李树雄	25.0
新型电子电路应用实例精选	苏文平	25.0
电子工艺及电子工程设计	汤元信	20.0
新传感器技术及应用	刘广玉	14.0
大规模可编程逻辑器件与数字系统设计	杨 晖	13.5
复杂数字电路与系统的 VerilogHDL 设计	夏宇闻	12.0
数字通信系统的 SystemView 仿真与分析	青 松	25.0
LabVIEW——易学易用的图形化编程语言	李 刚	26.0
Linux 系统集成	邱仲藩	24.0
Linux 基础教程	黄 河	36.0
现代数字移动通信原理及实用技术	何希才	26.0
信号处理技术基础	周浩敏	22.0
可编程控制器	郑瑜平	10.0
MATLAB 教程——基于 6. X 版本	张志涌	34.0
MATLAB 基础与应用简明教程	张 平	18.0
精通 MATLAB(5.3)(附光盘)	张志涌	57.0
掌握和精通 MATLAB	张志涌	28.0

可直接向出版社邮购以上图书。通信及汇款地址：北京航空航天大学出版社邮购组(邮编 100083)
电话：010—82317024, 传真：010—82328026。另加 2 元挂号费。(表中反白者为新书)

《单片机与嵌入式系统应用》月刊

- 北京航空航天大学出版社承办
- 何立民教授主编
- 中央级科技期刊
- 单片机与嵌入式系统专业期刊
- 特色：专业期刊、专家办刊、着眼世界、面向国内、应用为主、读者第一。
- 已开辟有 8 个栏目：业界论坛、专题论述、技术纵横、新器件新技术、应用天地、经验交流、学习园地、编读往来
- 杂志社网址：<http://www.microcontroller.com.cn>, <http://www.dpi.com.cn>

诚挚欢迎业界人士向本刊投稿, 欢迎广大读者订阅本刊。

联系方式

通信或汇款地址：北京市海淀区学院路 37 号《单片机与嵌入式系统应用》杂志社
 邮编：100083 电话：(010)82317029, 82313656 传真：010-82317043
 E-mail: mcu@publica.bj.cninfo.net (投稿亦请用此 E-mail 地址)
 广告业务联系人：王金萍 魏洪亮 开户行：北京市商业银行学院路支行
 户名：《单片机与嵌入式系统应用》杂志社 账号：010903391001201110299-36

订阅方式

国内统一刊号：CN 11-4530/V 国际标准刊号：ISSN 1009-623X
 每期定价：8 元，全年定价：96 元，每月 1 日出版。
 第一种订阅方式：通过邮局订阅, 邮发代号：2-765。
 第二种订阅方式：直接向《单片机与嵌入式系统应用》杂志社订阅, 另加 15% 邮资。