

# I<sup>2</sup>C 总线的串行扩充技术

何立民教授

摘要: ( )

I<sup>2</sup>C 总线具有十分完善的总线协议, 可构成多主系统。在协议软件支持下, 可自动处理总线上任何可能的运行状态。本篇主要介绍 I<sup>2</sup>C 总线的虚拟技术, 在单片机应用系统中, 这种虚拟技术应用十分广泛。(本文节选自何立民教授新作《单片机高级教程》, 在此, 对我的恩师何立民先生表示衷心感谢! 一周立功 19/9/2000)

## 一、I<sup>2</sup>C 总线的串行外围扩展

### 1. 外围扩展示意

I<sup>2</sup>C 总线为同步串行数据传输总线, 用于单片机的外围扩展。其总线传输速率为 100kb/s (改进后的规范为 400kb/s), 总线驱动能力为 400pF。

图 1 为 I<sup>2</sup>C 总线外围扩展示意。图中表示出单片机应用系统常用的 I<sup>2</sup>C 总线外围通用器件、外围设备模块、接口以及其它单片机节点。

最常使用的通用外围器件有 SRAM、E<sup>2</sup>PROM、ADC/DAC、RTC、I/O 口、DTMF 等。

外围设备模块有 LED 驱动控制器构成的 LED 显示器, 各种 LCD 驱动控制器构成的段式、图形点阵、字符点阵液晶显示器等。

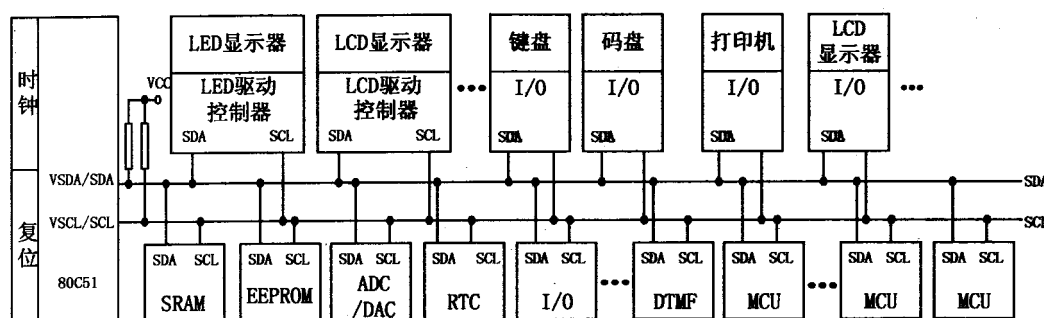


图 1 I<sup>2</sup>C 总线外围扩展示意

通过 I<sup>2</sup>C 总线通用 I/O 口器件可构成许多通用接口如键盘、码盘、打印机接口和 LCD 接口等。

I<sup>2</sup>C 总线可构成多主系统, 故 I<sup>2</sup>C 总线上可挂接一些带 I<sup>2</sup>C 总线接口的单片机, 因为带 I<sup>2</sup>C 接口的单片机中, 有支持多主功能的 SFR。除了图 1 中的通用外围器件外, Philips 公司还提供了许多视频、音像、通信系统中带 I<sup>2</sup>C 总线接口的专用外围器件, 在通信、视频、音像家电系统中 I<sup>2</sup>C 总线得到了广泛地应用。

### 2. 总线容量与驱动能力

I<sup>2</sup>C 总线的外围扩展器件都是 CMOS 器件, 总线有足够的电流驱动能力, 因此总线上扩展的节点数不是由电流负载能力决定, 而是由电容负载确定。I<sup>2</sup>C 总线上每个实际的节点器件的 I<sup>2</sup>C 总线接口都有一定的等效电容, 等效电容的存在会造成总线传输的延迟而导致数据传输出错。通常 I<sup>2</sup>C 总线负载能力为 400pF, 据此可计算出总线长度及节点数目的限制数量。

总线上的每个外围器件都有一个器件地址。总线上扩展外围器件时也要受器件地址限制。

### 3. 总线的电气结构

I<sup>2</sup>C 总线为双向同步串行总线，因此，I<sup>2</sup>C 总线接口内部为双向传输电路，如图 2 所示。总线端口输出为开漏结构，故总线上必须有上拉电阻  $R_p$ ，上拉电阻与电源电压  $V_{DD}$ 、SDA/SCL、总线串接电阻  $R_s$  有关，可参考有关数据手册选择，通常可选 5-10k  $\Omega$ 。

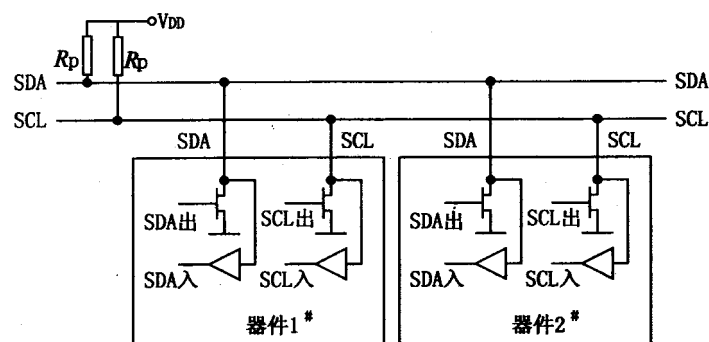


图 2 80C51 I<sup>2</sup>C 总线接口的电气结构

#### 4. 总线节点的寻址方式

挂接到总线上的所有外围器件、外设接口都是总线上的节点。在任何时刻总线上只有一个主控制器件（主节点）实现总线的控制操作，对总线上的其它节点寻址，分时实现一点的数据传送。因此，总线上每个节点都有一个固定的节点地址。

I<sup>2</sup>C 总线上的单片机都可以成为主节点，其器件地址由软件给定，存放在 I<sup>2</sup>C 总线的地址寄存器器件，称为主器件的从地址。在 I<sup>2</sup>C 总线的多主系统中，单片机作为从节点时，其从地址才有意义。

I<sup>2</sup>C 总线上所有的外围器件都有规范的器件地址。器件地址由 7 位组成，它和 1 位方向位构成了 I<sup>2</sup>C 总线器件的寻址字节 SLA。寻址字节格式如下：

D7				D0			
SLA	DA3	DA2	DA1	DA0	A2	A1	A0
							R/W

器件地址（DA3、DA2、DA1、DA0）：是 I<sup>2</sup>C 总线外围接口器件固有的地址编码，器件出厂时，就已给定。例如，I<sup>2</sup>C 总线 E<sup>2</sup>PROM AT24CXX 的器件地址为 1010，4 位 LED 驱动器 SAA1064 的器件地址为 0111。

引脚地址（A2、A1、A0）：是由 I<sup>2</sup>C 总线外围器件地址端口 A2、A1、A0 在电路中接电源或接地的不同，形成的地址数据。

数据方向（R/W）：数据方向位规定了总线上主节点对从节点的数据传送方向，R—接收，W——发送。

## 二、主方式下的 I<sup>2</sup>C 总线虚拟技术

### 1. I<sup>2</sup>C 总线虚拟技术

#### （1）多主应用的 I<sup>2</sup>C 总线

I<sup>2</sup>C 总线软、硬件协议十分巧妙，它可以用于构成多主系统。系统中有多多个 I<sup>2</sup>C 总线接口单片机时，会出现多主竞争的复杂状态。I<sup>2</sup>C 总线软、硬件协议，以及 I<sup>2</sup>C 总线单片机中的 SFR 保证了多主竞争时的协调管理。I<sup>2</sup>C 总线提供的状态处理软件包能自动处理总线上出现的 26 种状态。在使用 I<sup>2</sup>C 总线时，将这些工具软件在程序存储器中定位后，利用这些软件编制出归一化操作命令，用于 I<sup>2</sup>C 总线应用程序设计中十分简单、方便。如果没有 I<sup>2</sup>C 总线接口的单片机，要构成多主系统的虚拟 I<sup>2</sup>C 总线，就必须在虚拟 I<sup>2</sup>C 总线中解决多主竞争状况，这几乎是不可能的。因此，在多主的 I<sup>2</sup>C 总线系统中，一定要使用带 I<sup>2</sup>C 总线接口的单片机。

## (2) 单片系统中的 I<sup>2</sup>C 总线虚拟

在单主方式的 I<sup>2</sup>C 总线系统中，总线上只有一个单片机，其余都是带 I<sup>2</sup>C 总线的外围器件。由于总线上只有一个单片机成为主节点，该单片机永远占据总线，不会出现总线竞争，主节点也不必有自己的节点地址。在这种情况下，单片机可以没有 I<sup>2</sup>C 总线接口，可以用两根 I/O 口线来虚拟 I<sup>2</sup>C 总线接口。

由于单片机应用系统绝大多数 I<sup>2</sup>C 总线都是单主系统，因此，I<sup>2</sup>C 总线的虚拟技术应用十分广泛。目前，包括许多视频、音像电器中，都采用了虚拟 I<sup>2</sup>C 总线技术。

## (3) 主方式下的 I<sup>2</sup>C 的虚拟技术

单主系统中，单片机节点不会成为从节点，故虚拟 I<sup>2</sup>C 总线只有主方式下的主发送和主接收两种操作方式。

主方式下的 I<sup>2</sup>C 总线虚拟，应按照 I<sup>2</sup>C 总线数据传送时序、主方式下的操作格式设计出主方式下的时序模拟子程序、主发送/主接收子程序，并将这些子程序归纳成虚拟 I<sup>2</sup>C 总线的软件包，在软件包的基础上给出归一化的 I<sup>2</sup>C 总线操作命令。使用者只须将虚拟 I<sup>2</sup>C 总线软件包装入程序存储器中，利用归一化操作指令来设计相关的 I<sup>2</sup>C 总线应用程序。

## 2. I<sup>2</sup>C 总线时序

### (1) I<sup>2</sup>C 总线上的数据传递时序

I<sup>2</sup>C 总线上传送数据时序如图 3 所示。总线上传送的每一帧数据均为一个字节。但启动 I<sup>2</sup>C 总线后，传送的字节数没有限制，只要求每传送一个字节后，对方回应一个应答位。在发送时，首先发送的是数据的最高位。每次传送开始有起始信号，结束时有停止信号。

在总线传送完一个字节后，可以通过对时钟线的控制，使传送暂停。例如，当某个外围器件接收 N 个字节数据后，需要一段处理时间，以便继续接收以后的字节数据，这时可在应答信号后，使 SCL 变低电平，控制总线暂停；如果主节点要求总线暂停，也可使时钟线保持低电平，控制总线暂停。

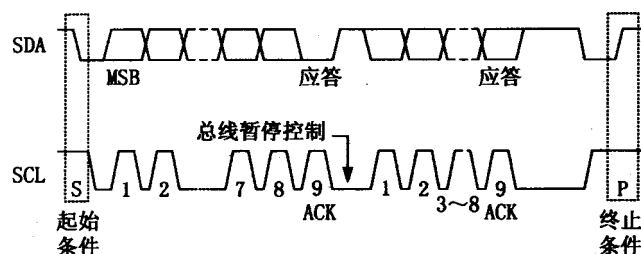


图 3 I<sup>2</sup>C 总线的数据传送时序

### (2) 总线上的时序信号

I<sup>2</sup>C 总线为同步传输总线，总线信号完全与时钟同步。I<sup>2</sup>C 总线上与数据传送有关的信号有起始信号（S）、终止信号（P）、应答信号（A）以及位传送信号等如图 4 所示。

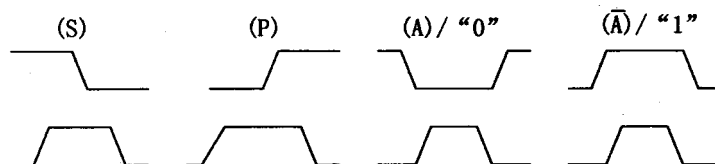


图 4 I<sup>2</sup>C 总线上的信号

①起始信号（S）。在时钟 SCL 为高电平时，数据线 SDA 出现由高电平向低电平变化时，启动 I<sup>2</sup>C 总线。

②终止信号（P）。时钟 SCL 为高电平时，数据线出现由低到高的电平变化时，将停止 I<sup>2</sup>C 总线数据传送。

③应答信号（A）。I<sup>2</sup>C 总线上第 9 个时钟脉冲对应于答位。相应数据线上低电平时为

“应答”信号 (A)，高电平时为“非应答”信号 (A)。

④数据位传送。在 I<sup>2</sup>C 总线启动后或应答信号后的第 1-8 个时钟脉冲对应于一个字节的 8 位数据传送。脉冲高电平期间，数据串行转送，低电平期间为数据准备，允许总线上数据电平变换。

### 3. 主方式下的数据操作格式

I<sup>2</sup>C 总线的数据操作格式是以条块图解形式来表达 I<sup>2</sup>C 总线的一次完整的数据传送过程。

I<sup>2</sup>C 总线上一次完整的数据传送如图 5 所示。其完整的数据操作包括起始 (S)、发送寻址字节 (SLA R/W)、应答、发送数据、应答.....直到终止 (P)。

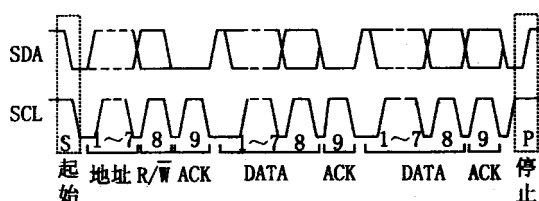
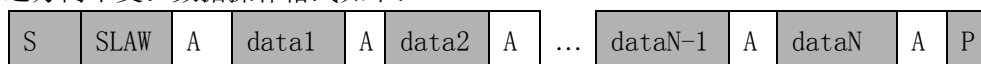




图 5 I<sup>2</sup>C 总线上一次完整的数据传送过程

对于不同方式下的操作略有不同，如果图 5 中的时序过程表示成下述操作格式，I<sup>2</sup>C 总线的数据传送过程便一目了然。

#### (1) 主发送的数据操作格式

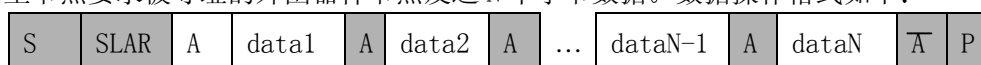
主节点向由寻址字节指定的外围器件节点发送 N 个字节数据，整个数据传送过程中数据传送方向不变。数据操作格式如下：



其中，  
 :主节点发送、从节点接收。  
 : 主节点接收、从节点发送。  
 SLAW : 寻址字节 (写)  
 data1-dataN : 写入从节点的 N 个数据。

#### (2) 主接收的数据操作格式

主节点要求被寻址的外围器件节点发送 N 个字节数据。数据操作格式如下：



其中，SLAR：寻址字节（读）。

在主接收中第一个应答是从节点接收到寻址字节 SLAR 后发回的应答位，其余的应答位都是由主控器在接收到数据后向从节点发出的应答位。

#### (3) 操作格式的应用特性

①操作格式是 I<sup>2</sup>C 总线的重要应用界面。对于具体外围器件的运行原理不同，其操作内容也不同。实际应用中要根据具体器件的运行原理列出具体的操作格式。

②无论哪种工作方式，都是由主控器来启动总线，发送寻址字节和终止运行。

③I<sup>2</sup>C 总线数据操作格式表示了 I<sup>2</sup>C 总线的一次完整的数据传送过程，它在 I<sup>2</sup>C 总线通用软件包的支持下自动完成。应用程序设计时只要按照归一化的操作指令准备好入口条件即可。

④在 I<sup>2</sup>C 总线接口的外围器件中，器件内部有多个地址空间时，其读写操作都有地址自动加 1 功能，简化了 I<sup>2</sup>C 总线的外部寻址。

### 三、主方式下的虚拟 I<sup>2</sup>C 总线软件包

为了在使用虚拟 I<sup>2</sup>C 总线时，有一个通用的傻瓜化界面，何立民教授设计了一个主方式下的虚拟 I<sup>2</sup>C 总线通用软件包 VIIC。并在 VICC 基础上给出了归一化操作命令。使用虚拟 I<sup>2</sup>C 总线时，装入 VIIC 软件包后，在应用程序设计中，只须使用三条归一化操作命令。

#### 1. 总线时序及数据操作模拟

##### (1) 典型信号的时序要求

I<sup>2</sup>C 总线数据传送时，有起始位 (S)、终止位 (P)、发送“0”及应答位 (A)、发送“1”及非应答位 ( $\bar{A}$ ) 等信号。按照典型 I<sup>2</sup>C 总线传送速率要求，这些信号时序如图 6 所示。

##### (2) I<sup>2</sup>C 总线典型信号的模拟子程序

对于 I<sup>2</sup>C 总线的典型信号，可以用指令操作来模拟其时序过程。设虚拟 I<sup>2</sup>C 总线的虚拟端口为 VSDA 数据线和 VSCL 时钟线。80C51 单片机的系统时钟为 6MHz，相应的单周期指令速度为 2 $\mu$ s。起始 (STA)、终止 (STOP)、发送应答位 (MACK) 发送非应答位 (MNACK) 的模拟子程序如下：

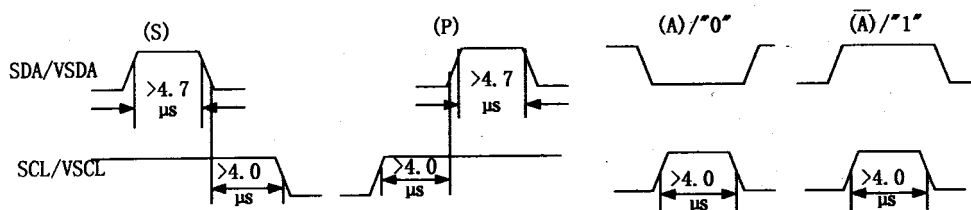


图 6 I<sup>2</sup>C 总线数据传送典型信号时序

启动 STA

```

STA: SETB    VSDA
      SETB    VSCL
      NOP
      NOP
      CLR     VSDA
      NOP
      NOP
      CLR     VSCL
      RET

```

停止 STOP

```

STOP: CLR    VSDA
      SETB    VSCL
      NOP
      NOP
      SETB    VSDA
      NOP
      NOP
      CLR     VSDA
      CLR     VSCL
      RET

```

发送应答位 MACK

```

MACK: CLR    VSDA

```

```

        SETB      VSCL
        NOP
        NOP
        CLR       VSCL
        SETB      VSDA
        RET
发送非应答位 MNACK
        MNACK: SETB      VSDA
        SETB      VSCL
        NOP
        NOP
        CLR       VSCL
        CLR       VSDA
        RET

```

在使用上述子程序时，如果单片机的主时钟不是 6MHz 时，应调整 NOP 指令个数，以满足时序要求。

### (3) I<sup>2</sup>C 总线模拟的通用子程序

从 I<sup>2</sup>C 总线的数据操作中可以看出，除了基本的启动 (STA)、终止 (STOP)、发送应答位 (MACK)、发送非应答位 (MNACK) 外，还应有应答位检查 (CACK) 和归一化的读写子程序。如发送一个字节 (WRBYT)、接收一个字节 (RBYT)、发送 N 个字节 (WRNBYT) 和接收 N 个字节的子程序 (RDNBYT)。

#### 应答位检查子程序 CACK

在应答位检查子程序 (CACK) 中，设置了标志位，CACK 中用 F0 作标志位，当检查到正常应答位后，F0=0，否则 F0=1。

```

CACK: SETB      VSDA      ; 置 VSDA 为输入方式
      SETB      VSCL      ; 使 VSDA 上数据有效
      CLR       F0        ; 预设 F0=0
      MOV       C, VSDA    ; 输入 VSDA 引脚状态
      JNC       CEND       ; 检查 VSDA 状态，正常应答转 CEND，且 F0=0
      SETB      F0        ; 无正常应答，F0=1
CEND: CLR       VSCL      ; 子程序结束，使 VSCL=0
      RET

```

#### 发送一个字节数据 (WRBYT) 子程序

该子程序是向虚拟 I<sup>2</sup>C 总线的数据线 VSDA 上发送一个字节数据的操作。调用本子程序前要发送的数据送入 A 中。占用资源：R0，C。

```

WRBYT: MOV      R0, #08H   ; 8 位数据长度送 R0 中
      WLP: RLC      A       ; 发送数据左移，使发送位入 C
      JC        WR1        ; 判断发送“1”还是“0”，发送“1”转 WR1
      AJMP      WR0        ; 发送“0”转 WR0
WLP1:  DJNZ      R0, WLP    ; 8 位是否发送完，未完转 WLP
      RET           ; 8 位发送完结束
      WR1: SETB      VSDA   ; 发送“1”程序段
      SETB      VSCL
      NOP

```

```

NOP
CLR      VSCL
CLR      VSDA
AJMP     WLP1
WRO:     CLR      VSDA          ; 发送“0”程序段
SETB     VSCL
NOP
NOP
CLR      VSCL
AJMP     WLP1

```

#### 从 VSDA 上接收一个字节数据 (RDBYT) 子程序

该子程序用来从 VSDA 是读取一个字节数据，执行本程序后，从 VSDA 上读取的一个字节存放在 R2 或 A 中。资源占用：R0, R2, C。

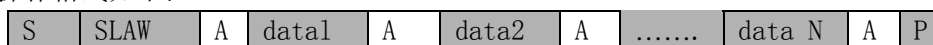
```

RDBYT: MOV     R0, #08H          ; 8 位数据长度入 R0
RLP:  SETB     VSDA              ; 置 VSDA 为输入方式
      SETB     VSCL              ; 使 VSDA 上数据有效
      MOV      C, VSDA           ; 读入 VSDA 引脚状态
      MOV      A, R2              ; 读入“0”程序段，由 C 拼装入 R2 中
      RLC      A
      MOV      R2, A
      CLR      VSCL              ; 使 VSCL=0 可继续接收数据位
      DJNZ     R0, RLP           ; 8 位读完否？未读完转 RLP
      RET

```

#### 向被控器发送 N 个字节数据 (WRNBYT) 子程序

在 I<sup>2</sup>C 总线数据传送中，主节点常常需要连续地向外围器件发送多个字节数据，本子程序是用来向 VSDA 线上发送 N 个字节数据的操作。该子程序的编写必须按照 I<sup>2</sup>C 总线规定的读、写操作格式进行。如主控器向 I<sup>2</sup>C 总线上某个外围器件连续发送 N 个数据字节时，其数据操作格式如下：



其中，SLAW：外围器件寻址字节（写）

按照上述操作格式所编写的发送 N 个字节的通用子程序 (WRNBYT) 清单如下：

```

WRNBYT: MOV     R3, NUMBYT
      LCALL     STA              ; 启动 I2C 总线
      MOV      A, SLA           ; 发送 SLAW 字节
      LCALL     WRBYT
      LCALL     CACK             ; 检查应答位
      JB       F0, WRNBYT       ; 非应答位则重发
      MOV      R1, #MTD
WRDA:  MOV      A, @R1
      LCALL     WRBYT
      LCALL     CACK
      JB       F0, WRNBYT
      INC      R1
      DJNZ     R3, WRDA

```

LCALL STOP  
RET

在使用本子程序时, 占用资源为 R1、R3, 但须调用 STA、WRBYT、CACK 子程序, 而且使用了一些符号单元, 在使用这些符号单元时, 应在内部 RAM 中分配好这些地址。这些符号单元有:

MTD: 主节点发送数据缓冲区首址;

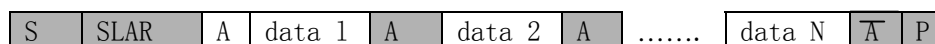
SLA: 外围器件寻址字节存放单元;

NUMBYT: 发送数据字节数存放单元。

在调用本子程序之前必须将要发送的 N 个字节数据依次存放在以 MTD 为首地址的发送数据缓冲区中。调用本子程序后, N 个字节数据依次传送到外围器件内部相应的地址单元中。

### 从外围器件读取 N 个字节数据 (RDNBYT) 子程序

在 I<sup>2</sup>C 总线系统中, 主控器按主接收方式从外围器件中读出 N 个字节数据的操作格式如下:



其中,

A: 非应答位, 主节点在接收完 N 个字节后, 必须发送一个非应答位;

SLAR: 外围器件寻址字节 (读)。

按照上述操作格式所编写的通用 N 个字节接收子程序 (RDNBYT) 清单如下:

```

RDNBYT: MOV      R3, NUMBYT
          LCALL   STA                ; 发送启动位
          MOV     A, SLA              ; 发送寻址字节 (读)
          LCALL   WRBYT
          LCALL   CACK                ; 检查应答位
          JB      F0, RDNBYT          ; 非正常应答时重新开始
RDN: MOV    R1, #MRD                ; 接收数据缓冲区首址 MRD 入 R1
RDN1: LCALL  RDBYT                  ; 读入一个字节到接收数据缓冲区中
          MOV     @R1, A
          DJNZ   R3, ACK              ; N 节读完否?未完转 ACK
          LCALL  MNACK                ; N 个字节读完发送非应答位 A
          LCALL  STOP                 ; 发送停止信号
          RET
ACK:  LCALL  MACK                    ; 发送应答位
          INC    R1                    ; 指向下一个接收数据缓冲单元
          SJMP   RDN1                 ; 转读入下一个字节数据
  
```

在使用 RDNBYT 子程序时, 占用资源 R1、R3, 但须调 STA、STOP、WRBYT、RDBYT、CACK、MACK、MNACK 等子程序, 须满足这些子程序的调用要求。RDNBYT 子程序中使用了一些符号单元, 除了在 WRNDYT 子程序中使用过的 SLA、MTD、NUMBYT 外还有以下几个:

SLA: 器件寻址 (读) 存放单元;

MRD: 主节点中数据接收缓冲区首址。

在调用 RDNBYT 子程序后, 从节点中所指定首地址 (SUBADR) 中的 N 个字节数据将被读入主节点片内以 MRD 为首址的数据缓冲器中。

## 2. 主方式虚拟 I<sup>2</sup>C 总线通用软件包 VIIC



## (1) 通用软件包 VIIC 的组成

主方式下虚拟 I<sup>2</sup>C 总线的通用软件包是虚拟 I<sup>2</sup>C 总线扩展时的一个应用程序设计平台，使用时，可将该软件包嵌入到程序存储器的任何空间。

虚拟 I<sup>2</sup>C 总线的通用软件包由信号模拟子程序 STA、STOP、MACK、MNACK 和通用子程序 CACK、WRBYT、RDBYT、WRNBYT、RDNBYT 共九个子程序组成。

通用软件包的程序清单如下：

```

①STA:   SETB      VSDA                      ; 启动 I2C 总线
        SETB      VSCL
        NOP
        NOP
        CLR       VSDA
        NOP
        NOP
        CLR       VSCL
        RET

② STOP: CLR       VSDA                      ; 停止 I2C 总线数据传送
        SETB      VSCL
        NOP
        NOP
        SETB      VSDA
        NOP
        NOP
        CLR       VSDA
        CLR       VSCL
        RET

③ MACK: CLR       VSDA                      ; 发送应答位
        SETB      VSCL
        NOP
        NOP
        CLR       VSCL
        SETB      VSDA
        RET

④MNACK: SETB      VSDA                      ; 发送非应答位
        SETB      VSCL
        NOP
        NOP
        CLR       VSCL
        CLR       VSDA
        RET

⑤CACK:  SETB      VSDA                      ; 应答位检查
        SETB      VSCL
        CLR       F0
        MOV       C, VSDA

```

```

        JNC      CEND
        SETB     F0
CEND:   CLR      VSCL
        RET
⑥ WRBYT: MOV     R0, #08H           ; 向 VSDA 线上发送一个数据字节
        WLP: RLC      A
        JC       WR1
        AJMP     WR0
WLP1:   DJNZ     R0, WLP
        RET
WR1:    SETB     VSDA
        SETB     VSCL
        NOP
        NOP
        CLR      VSCL
        CLR      VSDA
        AJMP     WLP1
WRO:    CLR      VSDA
        SETB     VSCL
        NOP
        NOP
        CLR      VSCL
        AJMP     WLP1
⑦ RDBYT: MOV     R0, #08H           ; 从 VSDA 线上读取一个数据字节
        RLJP: SETB     VSDA
        SETB     VSCL
        MOV      C, VSDA
        MOV      A, R2
        RLC      A
        MOV      R2, A
        CLR      VSCL
        DJNZ     R0, RLP
        RET
⑧ WRNBYT: MOV     R3, NUMBYT        ; 虚拟 I2C 总线发送 N 个字节数据
        LCALL    STA
        MOV      A, SLA
        LCALL    WRBYT
        LCALL    CACK
        JB       F0, WRNBYT
        MOV      R1, #MTD
WRDA:   MOV      A, @R1
        LCALL    WRBYT
        LCALL    CACK
        JB       F0, WRNBYT

```

```

        INC            R1
        DJNZ          R3, WRDA
        LCALL         STOP
        RET
⑨ RDNBYT: MOV        R3, NUMBYT          ; 模拟 I2C 总线接收 n 个字节数据
        LCALL         STA
        MOV           A, SLA
        LCALL         WRBYT
        LCALL         CACK
        JB            F0, RDNBT
RDN: MOV            R1, #MRD
RDN1: LCALL         RDBYT
        MOV           @R1, A
        DJNZ          R3, ACK
        LCALL         MNACK
        CLALL         STOP
        RET
ACK:  LCALL         MACK
        INC           R1
        SJMP          RDN1

```

该通用软件包可以设置在程序存储器的任何空间。

该通用软件占用的资源有 R0、R1、R2、R3、F0、C。

通用软件包中的符号单元有：

MTD：发送缓冲区首址；

MRD：接收缓冲区首址；

SLA：寻址字节（SLA W/R）存放单元；

NUMBYT：传送字节数 N 存放单元。

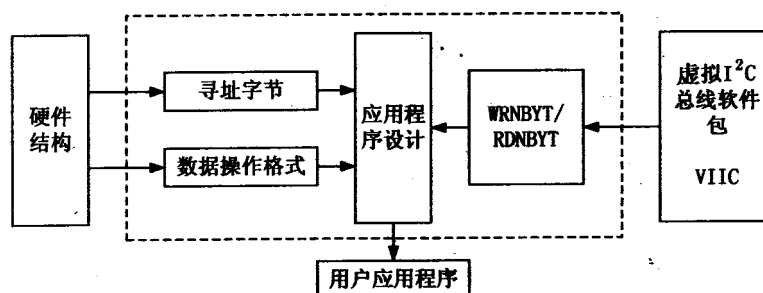
使用通用软件包时应事先分配好这些符号地址。

## （2）归一化的通用读写子程序 WRNBYT/RDNBYT

虚拟 I<sup>2</sup>C 总线软件包中，与应用程序设计直接相关的是通用读写子程序 WRNBYT/RDNBYT，它是虚拟 I<sup>2</sup>C 总线的归一化子程序，可普遍应用在虚拟 I<sup>2</sup>C 总线主方式下 I<sup>2</sup>C 总线外围扩展的应用程序设计中。

## 3. 应用程序设计界面

在通用软件包的平台支持下，I<sup>2</sup>C 总线应用程序设计异常简便，其应用程序设计界面如图 7 所示。可以看出，与应用程序设计直接相关的有：寻址字节、数据操作格式和归一化读写子程序。

图7 虚拟 I<sup>2</sup>C 总线应用系统设计界面

(1) 寻址字节。寻址字节有寻址字节(写)SLAW 和寻址字节(读)SLAR。寻址字节表明了 I<sup>2</sup>C 总线上的操作对象及数据传送方向。

(2) 数据操作格式。由具体节点器件给出的,能完全表达器件工作过程中,数据操作流程的数据流。

(3) 归一化读写子程序。归一化读写子程序是由通用软件包提供的入口界面软件。主方式下虚拟 I<sup>2</sup>C 总线的通用软件包为 VIIC,其归一化读写子程序 RDNBYT、WRNBYT。

#### 4. 虚拟 I<sup>2</sup>C 总线应用指南

(1) 适用范围。虚拟 I<sup>2</sup>C 总线只适用于主方式下 I<sup>2</sup>C 总线的外围扩展,即适于单主系统。

(2) 通用软件包(VIIC)装载。在虚拟 I<sup>2</sup>C 总线应用程序设计,应首先将通用软件包 VIIC 嵌入到程序存储器的任一空间,作为 I<sup>2</sup>C 总线应用程序设计的软件支持平台。

(3) 软件包中标记符号赋值。VIIC 中有许多标记符号,要根据系统的资源分配,在汇编程序设计中通过赋值语句赋值。这些标记符号有:

VSDA	虚拟 I <sup>2</sup> C 总线数据线;
VSCL	虚拟 I <sup>2</sup> C 总线时钟线;
SLA	寻址字节存放单元;
SLAW	寻址字节写;
SLAR	寻址字节读;
NUMBYT	传送字节数存放单元;
MTD	发送数据缓冲区;
MRD	接收字节缓冲区。

(4) 使用归一化操作命令。在程序中嵌入了通用软件包(VIIC),并进行标号赋值后,在应用程序设计时,只须灵活应用下述归一化操作命令即可:

```
MOV SLA, #SLAW/#SLAR    ; 指向那个节点,并认定发送(SLAW)还是接收(SLAR)
MOV NUMBYT, #N           ; 传送几个字节
LCALL WRNBYT/RDNBYT      ; 调用归一化发送或接收子程序 WRNBYT/RDNBYT
```

#### 四、虚拟 I<sup>2</sup>C 总线应用实例

本节中列举了单片机应用系统中常用的 I<sup>2</sup>C 总线外围器件 E<sup>2</sup>PROM、I/O 口、ADC/DAC、LED 驱动/控制器,以及由 I/O 构成的键盘、码盘接口应用实例。

##### (一) 虚拟 I<sup>2</sup>C 总线实验演示系统

图 8 是本节要介绍的一些 I<sup>2</sup>C 总线外围器件演示电路系统。系统中有独立的 I<sup>2</sup>C 总线外围器件 E<sup>2</sup>PROM 和 ADC/DAC;有 LED 驱动器 SAA1064 构成的 8 位 LED 显示器;还有由 I<sup>2</sup>C

总线通用 I/O 口 PCF8574 构成的 4 键/4LED 电路和 4 位 BCD 码码盘接口电路。

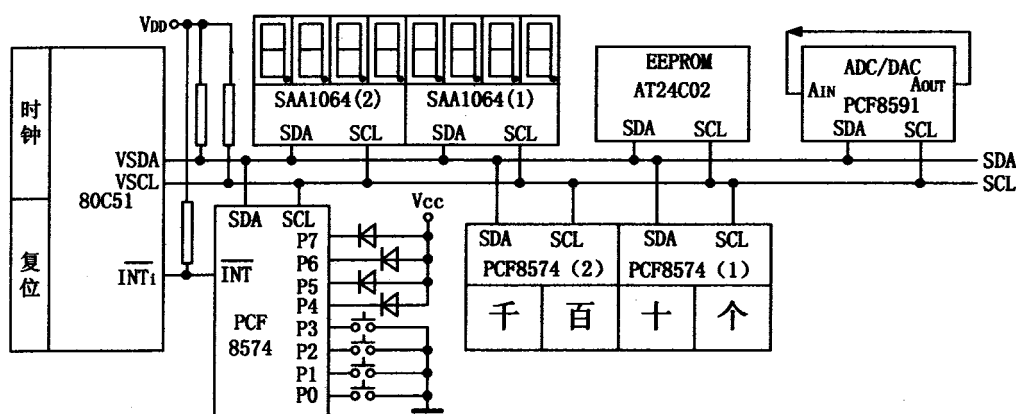


图 8 虚拟 I²C 总线实验演示系统

系统中所有外围器件节点都有器件地址和引脚地址构成的节点寻址字节。一个系统中的所有节点地址不允许出现重复。

在图 8 中的所有应用实例中，我们都使用虚拟 I²C 总线包 VIIC，并假设 VIIC 已安放在程序存储器中。这样，在应用程序实例中，我们只使用以下三条归一化操作指令，即

```
MOV    SLA, #SLAR/W
MOV    NUMBYT,
LCALL  RDNBYT/WRNBYT
```

使用 VIIC 及归一化操作指令，假设我们对 VIIC 及归一化操作指令中的符号标记已进行了以下伪指令定义，即

```
VSDA    EQU    P1.7
VSCL    EQU    P1.6
SLA      EQU    50H
NUMBYT   EQU    51H
MTD      EQU    30H
MRD      EQU    40H
SLAW     EQU    外围器件寻址字节(写)
SLAR     EQU    外围器件寻址字节(读)
```

其中 SLAW 和 SLAR 视外围器件的器件地址和引脚地址连接情况而定，将在各部分应用中述及。

## （二）E²PROM 的读写应用

### 1 硬件电路设计

#### （1）器件选择

带 I²C 总线接口的 E²PROM 有许多型号系列，其中 AT24CXX 系列使用十分普遍，有 AT2401/02/04/08/16 等，其容量分别为 128\*8/256\*8/512\*8/1024\*8/2048\*8。图 9（b）是一个 AT24C02 的 E²PROM 扩展电路，图 9（a）是其封装引脚示意。

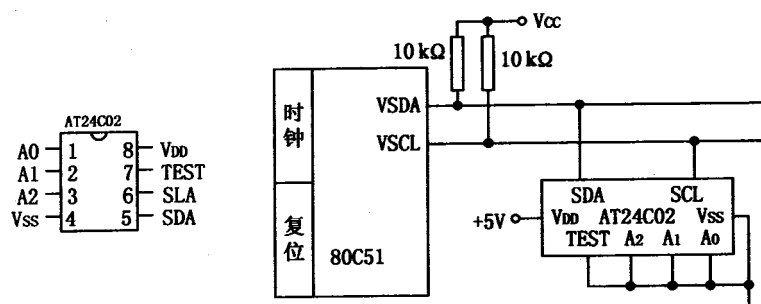


图 9 AT24C02 及其外围扩展电路

## (2) 电路设计

AT24C02 的 TEST 脚为测试端，系统中可接地处理。A2、A1、A0 可任接，因引 I<sup>2</sup>C 总线上可连接多达 8 片，总容量为  $8 \times 256 \times 8 = 2048 \times 8$ 。由于片内子地址采用 8 位地址指针寻址，超过  $256 \times 8$  时，要占用引脚地址，如果使用 AT24C04，则 A0 作为子地址软件寻址位；使用 AT24C08、AT24C16 时，A1、A2 也分别作为子地址的软件寻址位，这时相应的外部 A0、A1、A2 无效。

## (3) 节点地址

AT24C02XX 的器件地址是 1010，A2、A1、A0 为引脚地址。按照图 9 (b) 中的连接方式引脚地址为 000，因此 AT24C02 在系统中的寻址字节 SLAW=A0H，SLAR=A1H。

## 2. AT24C02 的结构与应用原理

AT24C02 为  $256 \times 8$  E<sup>2</sup>PROM，图 10 为其内部结构示意图。由输入缓冲器和 E<sup>2</sup>PROM 阵列组成。由于 E<sup>2</sup>PROM 的半导体工艺特性，写入时间为 5–10ms，如查从外部直接写入 E<sup>2</sup>PROM，每写一个字节都要等候 5–10ms，成批数据写入时等候时间很长。在设置 SRAM 性质的输入缓冲器时，对 E<sup>2</sup>PROM 的写入变成对 SRAM 缓冲器的装载，装载完后启动一个自动写入逻辑将缓冲器中的全部数据一次写入 E<sup>2</sup>PROM 阵列中。对缓冲器的输入称为页写，缓冲器的空量称为页写字节数。AT24C02 的页写字节数为 8，占用最低 3 位地址，只要从最低 3 位零地址开始写入，不超过页写字节数时，对 E<sup>2</sup>PROM 器件的写入操作与对 SRAM 的操作相同。若超过页写字节数时，应等候 5–10ms 后再启动一次写操作。

由于缓冲区容量较小，只占据最低 3 位，且不具备溢出进位功能，在从非零地址写入 8 个字节数，或从零地址写入字节数超过 8 个字节时，会形成地址翻卷，导致写入出错。

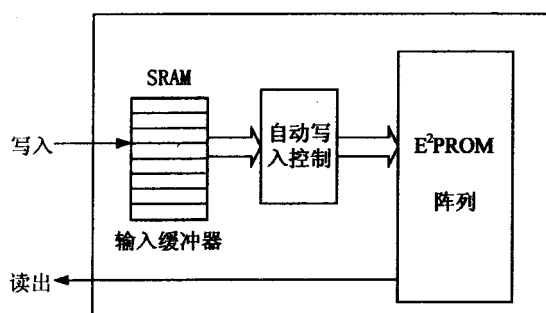
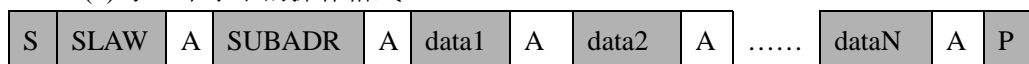


图 10 AT24C02 E<sup>2</sup>PROM 结构示意图

### (1) 写 N 个字节的操作格式



## 3. AT24CXX 的数据操作格式

在 I<sup>2</sup>C 总线中，对 AT24C02 内部存储单元读写时，除了要寻址该器件的节点地址外，还须指定存储读写的子地址 (SUBADR)。

按照 AT24C02 的器件手册，读、写 N 个字节的数据操作格式如下：

## (2) 读 N 个字节操作格式

S	SLAW	A	SUBADR	A	S	SLAR	A	data1	A	data2	A	...	dataN	$\overline{A}$	P
---	------	---	--------	---	---	------	---	-------	---	-------	---	-----	-------	----------------	---

在读操作中，除了发送寻址字节外，还要发送子地址 SUBADR。因此，在读 N 个字节操作前，要进行一个字节（SUBADR）的写操作，然后重新启动读操作。

## (三) LED 显示应用

## 1. 硬件电路设计

## (1) 器件选择

在 I<sup>2</sup>C 总线接口的通用外围电路中，有可实现 LED 驱动控制的 SAA1064 接口芯片。该器件可静态驱动 2 位 LED、动态驱动 4 位 LED，只有一个地址引脚 ADR，但可选择 4 种电平状态。故一个 I<sup>2</sup>C 总线上最多只能挂接 4 个 SAA1064，最多可扩展 16 位 LED 显示。

SAA1064 中有动态驱动控制电路，不须外部动态驱动管理，故在外部仍呈现出静态 LED 驱动特性。

## (2) 电路设计

图 11 (b) 是由两片 SAA1064 构成的 8 位 LED 显示电路。两片 SAA1064 都是标准的动态驱动控制接法，不同的 SAA1064 只在地址引脚 ADR 上连接不同，以区别不同的节点地址。

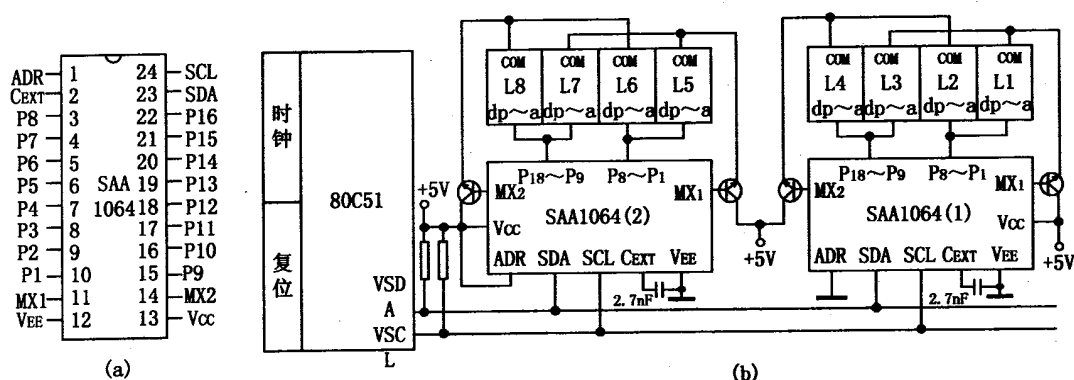


图 11 SAA1064 及其 8 位 LED 显示电路

## (3) 节点地址

SAA1064 的器件地址是 0111。地址引脚只有一个即 ADR，引脚地址 A2、A1、A0 采取 ADR 模拟电平的比较编址。ADR 引脚电平为 0、3/8V<sub>DD</sub>、V<sub>DD</sub> 时，相应引脚地址 A2、A1、A0 为 000、001、010、011。

在图 11(b) 中，两片 SAA1064 有两个节点地址。为连接简单，通常 ADR 一个接地，一个接 V<sub>DD</sub>，按图中接法，SAA1064 (1) 的 SLAW/SLAR 为 70H/71H；SAA1064 (2) 的 SLAW/SLAR 为 76H/77H。

## 2. SAA1064 的结构与应用原理

## (1) 封装与引脚功能

图 11 (a) 是 SAA1064 DIP24 封装引脚示意。引脚功能分述如下：

ADR 地址引脚端，4 个模拟电平状态对应 4 个引脚地址。

C<sub>EXT</sub> 时钟振荡器外接电容，典型值为 2.7nF。

P1-P16 段驱动输出端口，分别为两个 8 位口，P1、P9 为最低位，P8、P16 为最高位。口锁存器置 1 时，端口状态为低电平。

MX1、MX2 动态显示方式下，公共段驱动输出端。外部应用驱动晶体管。

SDA、SCL I<sup>2</sup>C 总线数据端、时钟端。

$V_{DD}$ 、 $V_{EE}$  电源 4.5-15V, 单片机系统中可用 5V。

## (2) LED 驱动控制原理

SAA1064 有 2\*8 个输出端口, 可静态控制 2 位 LED 显示, 但在 MX1、MX2 的动态控制管理下可实现 4 位 LED 驱动控制。图 11 (b) 即是 SAA1064 动态驱动的应用方式。

LED 的段驱动端口为 P1-P16。给口锁存器置 1 时, 端口为低电平状态, 相应的 LED 段点亮, LED 为共阳极器件, 但由于段驱动为送“1”点亮, 故 LED 的段码为共阴极。

## (3) 片内寄存单元

在 SAA1064 中有 5 个寄存器单元, 分别为 1 个控制寄存器和 4 个显示寄存器, 如表 1 所示。

表 1 SAA1064 片内寄存器单元及装载内容

地址单元	00H	01H	02H	03H	04H
装载内容	控制命令 COM	显示段码 1 data 1	显示段码 2 data 2	显示段码 3 data 3	显示段码 4 data 4

## (4) 控制命令格式

SAA1064 具有较强的控制功能, 能实现亮度控制, 显示器测试, 静、动态及位的亮、暗显示控制, 这些控制位都设置在控制命令 (COM) 中。控制命令格式为:

	D7							D0
00H	——	C6	C5	C4	C3	C2	C1	C0

各控制位功能如下:

C0 静、动态显示选择, C0=1 动态显示。

C1 显示位 1、3 暗亮选择, C1=1 选择亮。

C2 显示位 2、4 暗亮选择, C2=1 选择亮。

C3 测试位, C3=1 时, 所有段点亮。

C4、C5、C6 驱动电流控制位, C4、C5、C6 为“1”时, 驱动电流分别为 3mA、6mA、12mA。皆为 1 时输出驱动电流最大, 达 21mA。

## 3. 数据操作格式

SAA1064 的显示驱动控制只须对其进行写操作。SAA1064 的写数据操作格式如下:

S	SLAW	A	SUBADR	A	COM	A	data1	A	data2	A	data3	A	data4	A	P
---	------	---	--------	---	-----	---	-------	---	-------	---	-------	---	-------	---	---

只须对 SAA1064 连续写入 SUBADR、COM、data1-data4 数据。其中 SUBADR=00H; COM 按控制要求选择; data1-data4 为共阴极段码; SLAW 为寻址字节 (写)。

## (四) I/O 口扩展应用

### 1. 硬件电路设计

#### (1) 器件选择

在 I<sup>2</sup>C 总线接口的通用外围电路中, 有 I/O 口扩展器件 PCF8574。这是一个带中断输出的 8 位准双向口, 可广泛用于 I/O 口接口的外围扩展。PCF8574 的中断输出引脚在 I/O 端口输入状态改变时, 有中断信号输出。因此, 用 PCF8574 扩展的外围器件及外设接口, 可工作在中断方式。在对 PCF8574 进行一次读写操作后, 便自动清除中断请求。I/O 端口的驱动能力  $I_{OL}=25mA$ ,  $I_{OH}<300\mu A$

#### (2) 电路设计

所有用 I/O 口构成的外围电路和外设接口, 都可以用 PCF8574 来完成, 例如, 各种类型的独立式键盘与行列式键盘、码盘输入接口、继电器驱动控制、LED 驱动控制等。图 12 (b) 给出了 PCF8574 与 80C51 I<sup>2</sup>C 总线的接口电路。PCF8574 的中断输出 (INT) 也是开



漏输出，要加上拉电阻，I/O 口内部有上拉电阻。

### (3) 节点地址

为了使 I<sup>2</sup>C 总线中可以挂接更多 I<sup>2</sup>C 总线的 I/O 口器件，I<sup>2</sup>C 总线接口的 I/O 口器件有两个器件地址，即 0100 和 0111，分属于 PCF8574 和 PCF8574A。在本节 I<sup>2</sup>C 总线演示系统中使用的 SAA1064 的器件地址为 0111，为避免重复而选用 PCF8574。

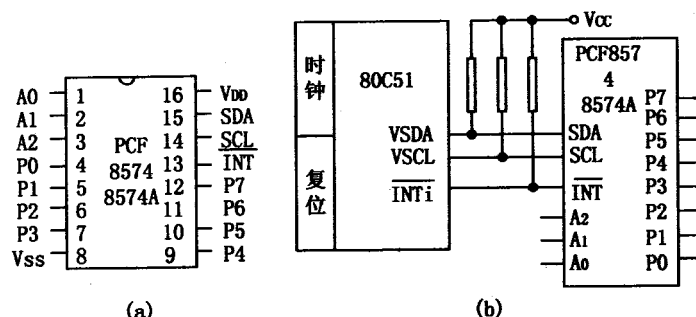


图 12 PCF8574 及其与 80C51 I<sup>2</sup>C 总线的接口方式

## 2. PCF8574 结构与应用原理

### (1) 封装及引脚功能

图 12 (a) 为 PCF8574 DIP 16 的封装引脚示意。引脚功能分述如下：

A2-A0	地址引脚；
P0-P7	8 位准双向口，驱动能力 $I_{OL}=25\text{mA}$ ， $I_{OH} < 300\mu\text{A}$ ；
SDA、SCL	I <sup>2</sup> C 总线接口；
INT	中断请求输出，低电平有效；
V <sub>DD</sub> 、V <sub>SS</sub>	电源端 2.5-6V，典型值+5V；

### (2) 应用原理

当 I<sup>2</sup>C 总线主节点对 PCF8574 进行一个字节的写操作进，即实现了 I/O 口的数据输出。I<sup>2</sup>C 总线发送到 PCF8574 中的串行数据，在应答位过后出现在 I/O 端口上。I<sup>2</sup>C 总线不断送数，I/O 口上的数据不断更迭。

当主节点对 PCF8574 读操作时，即可实现 I/O 端口数据的输入，读入的数据存放在接收缓冲区 MRD 中。

当 PCF8574 输入端电平状态改变时，中断请求输出端 INT 出现低电平，中断输出有效。从 I/O 输入端状态变化到中断输出 INT 变低电平的迟后时间约 4  $\mu\text{s}$ 。在对其读写操作后，中断请求复位（呈高电平）。

### 3. 数据操作格式

I<sup>2</sup>C 总线对 PCF8574 I/O 口的输出操作是一个字节的写入操作，对 PCF8574 I/O 口的输入操作是一个字节的读出操作。

ADC、DAC 应用 PCF8574/8574A 输出的数据操作格式如下：

S	SLAW	A	P0data	A	P
---	------	---	--------	---	---

主器件发送数据 P0data, PCF8574/8574A 送回应答位后，数据便出现在 I/O 端口上。PCF8574/8574A 的输入数据操作格式如下：

S	SLAR	A	PIdata	A	P
---	------	---	--------	---	---

主器件发送了寻址字节 SLAR 后, PCF8574 在第一位应答位的 SCL 上升沿处将口状态数据 PIdata 捕获到口锁存器中, 随后, 主器件交 PCF8574 口锁存器中的数据读入接收缓冲器中。

## (五) ACD、DAC 应用

### 1. 硬件电路设计

#### (1) 器件选择

目前已有一些厂家推出了带 I<sup>2</sup>C 总线接口的 ADC 和 DAC。PCF8591 是最早推出的带 I<sup>2</sup>C 接口的 ADC 的 DAC。

PCF8591 中有 4 路 8 位 ADC 和 1 路 8 位 DAC。图 13 (a) 为 PCF8591 DIP16 封装引脚示

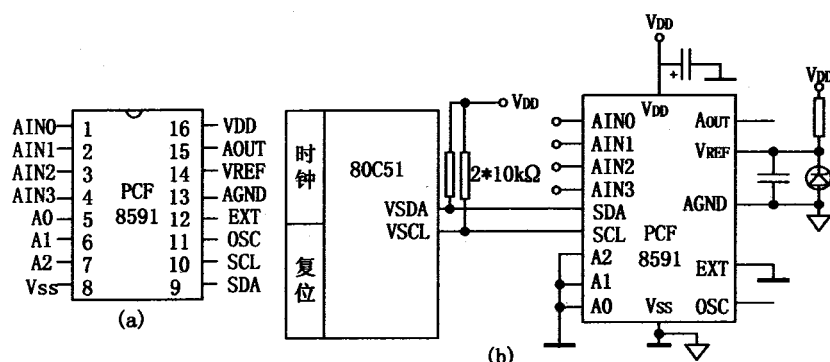


图 13 PCF8591 与 80C51 的虚拟 I<sup>2</sup>C 总线接口的方式

意, 其引脚功能如下:

AIN0-AIN3	模拟信号输入端, 可选择 4 种输入方式;
A0-A2	引脚地址输入端;
V <sub>DD</sub> 、V <sub>SS</sub>	电源输入端, 2.5-6V;
SDA、SCL	I <sup>2</sup> C 总线的数据端、时钟端;
OSC	外部时钟输入端、内部时钟输出端;
EXT	内、外部时钟选择线, 使用内部时钟时, EXT 接地;
AGND	模拟信号地;
V <sub>REF</sub>	基准电源输入端;
A <sub>OUT</sub>	DAC 的模拟量输出端;

#### (2) 电路设计

图 13(b) 为 PCF8591 的接口电路。图中 PCF851 使用内部时钟电路, EXT 接地, OSC 可输出内部时钟。基准电压取自 V<sub>DD</sub>, 采取稳压措施。模拟信号输入可采用单端或差分方式。

#### (3) 节点地址

PCF8591 的器件地址为 1001, 图中 A2、A1、A0 接地, 故其寻址 (写) SLAW=90H, 寻址字节 (读) SLAR=91H。对于 PCF8591 写操作实现 D/A 转换、读操作时实现为 A/D 转换, 但无论 D/A 转换还原 A/D 转换都要先写入控制命令。

## 2. 结构与应用原理

PCF8591 内部有一个控制寄存器, 用来存放控制命令, 控制命令如图 14 所示。

D1、D0 为模拟量输入通道选择, 可任意选择当前模拟输入通信号。

D2 为自动增量选择位。D2 置位、A/D 通道信号将自动循环递增, 每次 A/D 转换结束都

自动选择下一通道。若选择不存在的通道号（如 2 路差分输入时只有通道 0、1 有效，通道 2、3 不存在）将导致选择最高的有效通道号。

D3、D7 必须设置为 0。

D5、D4 为模拟量输入方式选择，分别为单端输入、3 路差分、单端与差分混合、两路差分输入 4 种方式。

D6 为模拟量输出允许，D6=1 时激活模拟量输出。

PCF8591 上电复位后，控制寄存器的状态为 00H，D/A 转换器和振荡器被禁止而处于节电方式，模拟量输出切换到高阻态。

### 3. 数据操作格式

#### (1) DAC 数据操作格式

对 PCF8591 进行写操作，实现 D/A 转换，其数据操作格式如下：

S	SLAW	A	COM	A	data	A
---	------	---	-----	---	------	---

首先写入控制命令 COM，设 D6=1，允许模拟量输出。然后写入 D/A 转换数据 data，在随后应答位结束时，相应的模拟电压出现在 PCF8591 的 A<sub>OUT</sub> 端口上。在写入 data 之前 A<sub>OUT</sub> 为高阻态或先前 DAC 寄存器中数据所对应的模拟电压。

#### (2) ADC 数据操作格式

对 PCF8591 进行读操作，实现 A/D 转换，其数据操作格式如下：

S	SLAW	A	COM	A	S	SLAR	A	data 0	A	data 1	A
---	------	---	-----	---	---	------	---	--------	---	--------	---

在读操作以前，应先写入控制命令 COM，在控制命令中，选择通道、模拟量输入方式，通道是否自动转换等。在重新启动的读操作中，PCF8591 在接收 SLAR、发回应答位后启动模拟电压采集、A/D 转换，其 A/D 转换结果为一个读出的数据 data 1，而此时读出的 data 0 是先前 A/D 转换的数据。

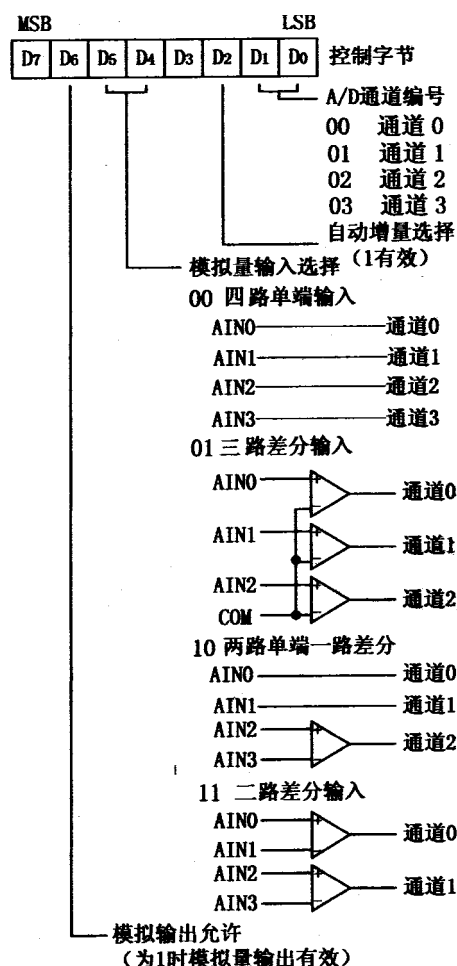


图 14 PCF8591 控制命令格式