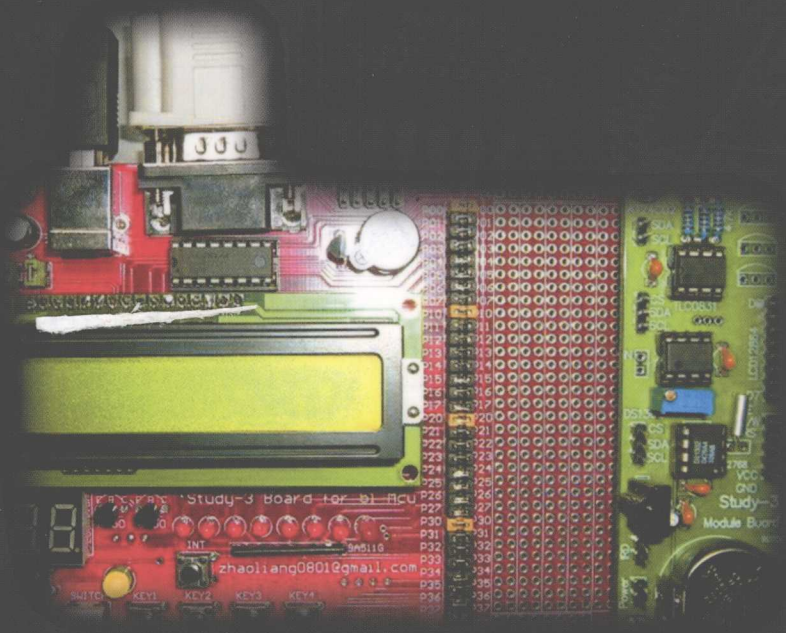
C51单片机
典型模块设计与应用

边春元 李文涛 江杰 杜平◎等编著



机械工业出版社
CHINA MACHINE PRESS

C51 单片机典型模块设计与应用

边春元　李文涛
江　杰　杜　平　等编著

机 械 工 业 出 版 社

本书以单片机的功能为模块，以外围设备、相关电路设计的实际应用为内容，向读者介绍了如何将单片机硬件、程序和外围设备的选择合理地实施到项目开发中。

本书8个章节中的单片机使用的典型实例都是经过精挑细选后确定的，基本覆盖了单片机的主要应用技术，例如单片机中断和定时器的使用、单片机的输入/输出、单片机的数据采集功能、单片机在控制系统中的应用、单片机的通信以及单片机的算法和信号处理等，并且内容取自于实际应用项目。

通过学习本书的实例，读者除可以掌握单片机的具体应用方法外，还可获得如何针对一个具体的项目需求设计解决方案，以及如何运用单片机的关键技术来满足项目需求。本书专业性和实用性较强，对于利用单片机进行实际项目开发具有非常高的参考价值。

本书适合单片机开发人员、单片机应用系统设计人员以及进行课程设计和毕业设计的大学本科生阅读和参考。

图书在版编目(CIP)数据

C51单片机典型模块设计与应用/边春元等编著. —北京：机械工业出版社，2008.4

ISBN 978-7-111-23624-5

Ⅰ.C… Ⅱ.边… Ⅲ.单片微型计算机 Ⅳ.TP368.1

中国版本图书馆CIP数据核字(2008)第029923号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑：张俊红 责任编辑：林 桢 版式设计：冉晓华

责任校对：刘志文 封面设计：王奕文 责任印制：洪汉军

北京铭成印刷有限公司印刷

2008年4月第1版第1次印刷

184mm×260mm ·21.25印张·527千字

0001—4000册

标准书号：ISBN 978-7-111-23624-5

定价：40.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

销售服务热线电话：(010)68326294

购书热线电话：(010)88379639 88379641 88379643

编辑热线电话：(010)88379764

封面无防伪标均为盗版

前　言

单片机技术的出现给现代工业测控领域带来了一次崭新的技术革命。目前，单片机仍以其高可靠性、高性价比等优点，在工业控制系统、数据采集和信号处理系统、智能化仪器仪表、智能家电等诸多领域得到广泛的应用。

尽管目前有关 C51 系列单片机类的图书品种繁多，侧重点也都各不相同，但主要还是以讲解理论或者具体芯片为主，即便是有模块的应用，也是为了说明知识点，还很少看到案例丰富的“功能模块剖析”，但是事实上很多常用的典型模块已经有很多人做过，是无需从头开发的。虽然目前市面上已有类似的图书，但从全面性、典型性、新颖性等方面来说，还有所欠缺。针对这种情况，我们撰写了这部系统、实用、新颖、内容齐全、面向实际工程应用的 C51 单片机常用功能模块参考用书，以供从事工业测控类技术开发人员学习、参考、借鉴之用。

全书共分 8 章，第 1 章介绍单片机片内及扩展资源操作；第 2 章介绍数据采集模块；第 3 章介绍人机接口模块；第 4 章介绍信号发生、测量、滤波与处理模块；第 5 章介绍通信与数据传输模块；第 6 章介绍控制模块；第 7 章介绍电磁兼容与可靠性设计；第 8 章着重介绍基于 C51 单片机的数据结构与典型算法的模块。

本书中的大部分程序源代码采用 C 语言编写，主要是考虑在实际应用中为了提高开发效率和程序的可移植性。同时，一些程序使用了汇编语言编写，这主要是针对那些对实时性要求较高的应用环境。读者在使用这些典型模块的程序源代码时，只需要将程序源代码全部内容链接在应用程序之后，统一编译即可。

本书主要由边春元、李文涛、江杰和杜平共同编著完成，参与部分章节写作的还有王志强、杨东升、宿国军、梁洪力、王宇龙、张春有、刘金海、梁日军、姜海波、渠建新、陈佰林、田雪和宋晗等。另外，张学静、姜海燕、程显奎、吴鹏、刘艳霞、柳艳菊、屈久良和刘建华参与了资料整理和程序调试等工作，这里对所有工作人员的努力一并表示感谢！

由于学识所限，错误和疏漏之处在所难免，恳请读者批评指正。

作　者

目　录

第 1 章　单片机片内及扩展资源操作

8051/80C51 是整个 MCS-51 系列单片机的核心。该系列的其他型号的单片机都是在这一内核的基础上发展起来的。本章将向读者详细介绍 8051/80C51 单片机的片内资源及典型应用。

1.1　C51 单片机 I/O 口的应用

1.1.1　实现目标

本节主要介绍 C51 单片机 I/O 功能的应用，从引脚的名称上看，MCS-51 有 I/O 口线 P0～P3 共 32 根，但 P3 口是多用途的，当用作替代功能时，就不能作为一般的 I/O 口使用。在接有外部存储器时，P0 和 P2 也不能再用作 I/O 口使用。

为了简单、直观地阐明单片机基本 I/O 口的使用方法，下面以发光二极管(LED)的流水灯实验为例，期望通过实战能够帮助读者深入理解 C51 单片机 I/O 口的原理和功能。

1.1.2　设计思路

在 LED 流水灯实例中，发光二极管是一种将电能转换为光能的半导体器件。当发光二极管两端加上一定的正向电压，使之流过一定的工作电流时，会发出可见光。根据使用的材料不同，发光二极管有红、黄、绿、蓝、紫等几种。发光的亮度与正向工作电流成正比，即工作电流越大，亮度越强。一般来讲，正常工作电流 I_g 为 5～20mA，压降 V_g 为 1.5～2.0V 之间。在使用发光二极管时，限流电阻的选择尤为重要，阻值过大或过小，二极管都将不能正常发光，甚至烧毁器件。限流电阻 R_x 应满足如下条件

$$R_x = \frac{V_{CC} - V_g}{I_g}$$

式中，V_{CC}是电源电压；V_g 是发光二极管工作时的管压降值；I_g 是发光二极管的工作电流极限。

这样，在 5V 电源电压下，限流电阻 R_x 的取值范围是

$$\frac{(4.01-2.0)\text{V}}{20\text{mA}} < R_x < \frac{(4.0-1.5)\text{V}}{5\text{mA}}$$

即 R_x 取值应为 150～700Ω 之间，一般来说建议选用 470Ω。

在单片机实验电路中，有 8 个发光二极管，它们与系统的连接原理图如图 1-4 所示。其中 8 个二极管的阳极接在一起，通过 9012 晶体管与 +5V 电源相连，而阴极分别通过限流电阻 R20～R27 连接到控制端 KD-Q0～KD-Q7。这样，控制 8 个发光二极管就需要 8 个 I/O 口。但由于单片机的 I/O 口资源是有限的，这样往往是不行的，因此常采用如图 1-1 所示的串/

并转换电路来扩充系统资源。串/并转换电路其实质是一个串入并出的移位寄存器，串行数据在同步移位脉冲 CLK 的作用下经串行数据线 SDA 把数据移位输出到 KD-Q0 ~ KD-Q7 端，这样仅需 3 根线(也可以用 2 根线)就可以分别控制 8 个发光二极管的亮灭。

串行输入数据为 0x7F 时，各输出口移位的时序如图 1-1 所示。由图 1-1 可以看出，数据 0x7F (01111111B)的最高位 0 在第一个时钟脉冲的下降沿时被移位输出到 KD-Q0 端，而其他输出端口 KD-Q1 ~ KD-Q7 的状态是不确定的，与上一状态有关。在第二个时钟脉冲的下降沿到来时，数据 0x7F (01111111B)的次高位 1 被移位输出到 KD-Q0 端口，而 KD-Q0 端口上一状态的值 0 被移位输出到了 KD-Q1 端口。这样，在 8 个时钟脉冲作用下，数据 0x7F 将被移位输出到 KD-Q7 ~ KD-Q0 端口。

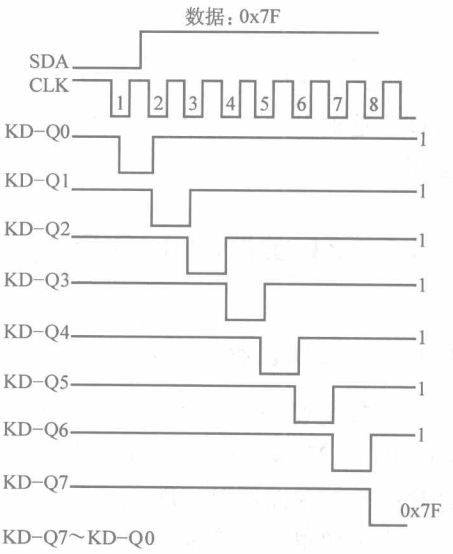


图 1-1　各输出口移位时序

串行数据线 SDA、同步时钟 CLK 和 EBITO 通过跳线分别与 P1.0、P1.1 和 P1.7 相连，可用上述方法把串行数据移位输出到并行口 KD-Q0 ~ KD-Q7 上。

如图 1-4 所示，要点亮一个发光二极管必须满足 2 个条件：发光二极管的阳极必须接电源正极，也就是 EBITO 脚为低电位，使 PNP 型晶体管(9012)可靠导通；该发光二极管的控制端 KD-Qx 为低电位。

发光二极管流水灯原理：首先向 P1.7 口送 0，使其为低电平，这样发光二极管的阳极因 V1 导通而带电；此时可向控制端发送数据 0x7F，由于此时 KD-Q7 为“0”而其他口为高电平“1”，所以仅有发光二极管 L8 发光。这样延时一段时间后发送第二个数据 0xBF，即让 KD-Q6 为低电平而保证其他位为高电平，此时 L8 灭而 L7 亮，依此类推 L8 ~ L1 将依次轮流点亮。

1.1.3　元器件选型

串/并转换电路可以使用复杂可编程逻辑器件(CPLD)器件，也可以直接采用 74HC595 或 74LS164 串/并转换芯片。74HC595 的引脚排列如图 1-2 所示。

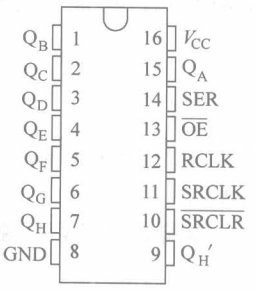


图 1-2　74HC595 的引脚排列

74HC595 的工作时序如图 1-3 所示。

1.1.4　电路设计

发光二极管的流水灯实验硬件电路原理如图 1-4 所示。

1.1.5　程序设计

程序流程如图 1-5 所示，实现流水灯功能的程序编写方法如下。

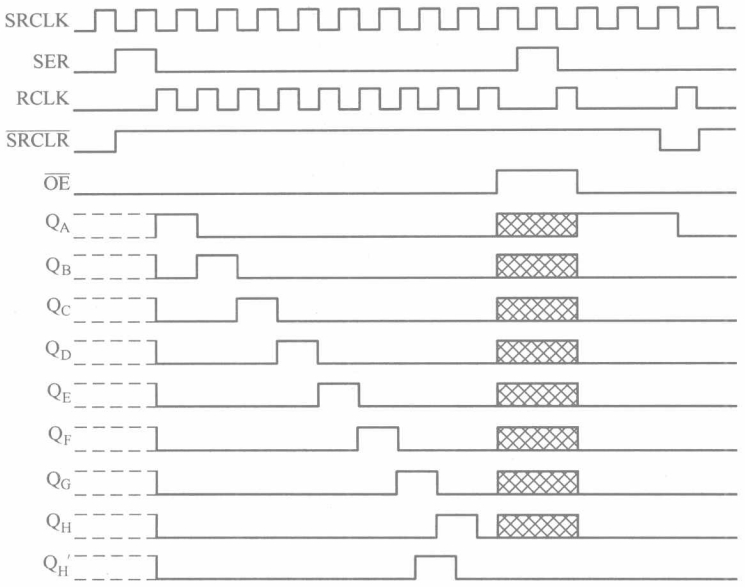


图 1-3　74HC595 的工作时序

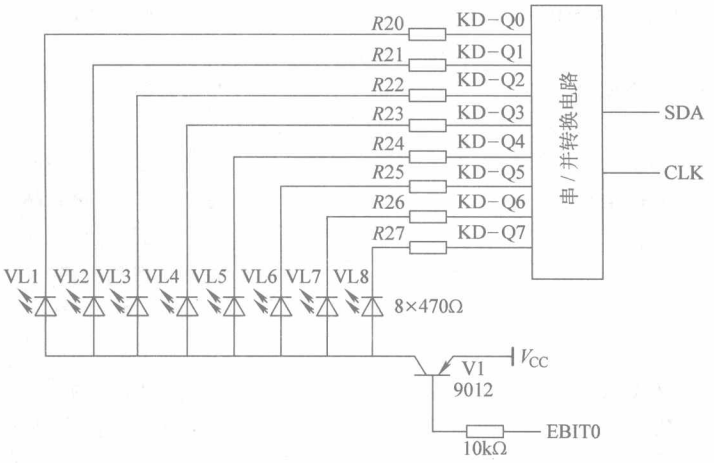


图 1-4　发光二极管的硬件原理

```
#include < reg52. h >                          //加载头文件
#include < intrins. h >
sbit SDA = P1^0;                              //定义串行数据线
sbit CLK = P1^1;                              //定义同步时钟
sbit LEDP = P1^7;                             //定义控制位
void delay( unsigned int i)                   //延时子程序
{
    unsigned int j,k;                         //定义中间变量
    for( k = 0;k < i;k ++ )                   //延时 i×1000 个时钟周期
```

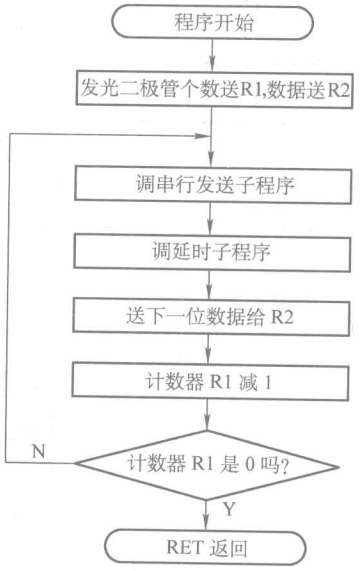


图 1-5　流水灯程序流程

```
        for(j=0;j<1000;j++);                    //延时 1000 个时钟周期
}
void send(unsigned char a)                      //串行发送子程序
{
    unsigned char i;                            //串行数据位定义变量
    for(i=0;i<8;i++)                            //发送 1B 数据循环体
        {
        if(_crol_(a,i)&0x80)                    //判断此位是 1 吗?
            SDA=1;                              //是,则置 P1.0 为 1
        else
            SDA=0;                              //否则置 P1.0 为 0
            CLK=0;                                  //发送一同步时钟信号
            CLK=1;
}
}
main()                                          //主程序
{
    unsigned char m;                            //定义循环变量
    unsigned char DLED;                         //定义变量
    LEDP=0;                                     //置 P1.7 为低电平
    while(1)                                    //无穷循环
    {
        DLED=0x7f;                              //送初始值
        for(m=0;m<8;m++)                        //流水循环体
```

```
            {
              send(DLED);                    //调用串行发送子程序
              delay(0x10);                   //调用延时子程序
              DLED = _cror_(DLED,1);         //更新变量
            }
        }
    }
```

1.1.6 典型应用

本例提供了单片机和外部驱动发光二极管的门电路的接口电路，该模块可用于实现工业自动控制系统的监控和报警等功能。

1.1.7 经验分享

由于P0、P1、P2和P3口用于普通的I/O口时，都是准双向口结构，因此它们的输入、输出操作不同，即输入操作是读引脚状态，而输出操作是对口锁存器的写入操作。这样当内部总线给口锁存器置0或置1时，锁存器中的“0”或“1”状态会立即反应到引脚上；而在输入操作(读引脚)时，如果锁存器的状态为“0”，该引脚会被钳位在低电平，导致无法读出该引脚的高电平输入。在准双向口作为输入口时，应先将锁存器置1，即先向该I/O口写1，使该I/O口工作于输入方式，然后再读引脚。

P0口为开漏输出结构，因此作为普通I/O口使用时必须外加上拉电阻，建议选用阻值为10kΩ的电阻。所有I/O口的驱动能力是有限的，因此在某些场合需要扩展一些驱动电路。

1.2 C51单片机的FPGA并行配置

1.2.1 实现目标

常用的外部控制器一般采用微控制器或CPU这样的智能主机，在采用微控制器配置时，如采用MCS-51单片机，一般只使用一位宽的串行数据通道而不是字节宽的并行数据通道。该方法的优点是不必采用Altera公司的专用存储器，使用廉价的通用程序存储器就可以实现，这对需要大容量器件配置文件时，在降低成本上是非常有利的。该方法的另一个突出优点是，可实现单系统多方案的配置。因为数据的配置过程是通过外部智能控制器进行的，所以可以根据需要，在通用存储器中对单一系统存放多种功能的配置文件，再由外部控制器根据具体情况自动选择对芯片配置何种功能。这一优点是其他任何一种配置方法所不具备的。本节将介绍利用AT89C52微处理器对FLEX 10K系列现场可编程门阵列(FPGA)器件进行被动并行异步(PPA)配置的方法。

1.2.2 设计思路

Altera公司生产的具有在线可配置性ICR(接口控制寄存器)功能的FPGA器件有

FLEX6000、FLEX 10K、APEX 和 ACEX 等系列。它们的配置方式可分为 PS(被动串行)、PPS(被动并行同步)、PPA(被动并行异步)、PSA(被动串行异步)和 JTAG(联合测试行动小组)等5种方式。这5种方式都适用于单片机配置。PS 方式因电路简单，对配置时钟的要求相对较低而被广泛应用。

相对而言，采用 PPA 配置模式的方案却很少见到。但由于 PPA 配置模式为并行配置，其配置速度快，且配置时钟由 FPGA 器件内部产生，而 PS 等配置模式需要外加配置时钟，故其更有利于在线实现。本实例的配置模式方案便是采用 PPA 配置模式实现的。

Altera 公司的 MAX + plus Ⅱ开发工具可以生成多种配置或编译文件，用于不同配置方法的配置系统。对于不同的目标器件，配置数据的大小不同，配置文件的大小一般由 . tbf 文件(即二进制文件)决定。本实例中，FLEX 10K 的配置文件 . rbf 的大小为 15KB。该文件包括所有的配置数据，1B 的 . rbf 文件有 8 位配置数据。

由于 Altera 公司提供的软件工具不自动生成 . rbf 文件，故文件需按照下面的步骤生成：

① 在 MAX + plus Ⅱ编译状态下，选择文件菜单中的变换 SRAM 目标文件命令；

② 在变换 SRAM 目标文件对话框中，指定要转换的文件，并且选择输出文件格式为 . rbf，之后予以确定。

在对 FPGA 器件进行配置时，单片机将 8 位的配置数据放在 FPGA 器件的数据端，并且给 nWS 一个负脉冲，在 nWS 的上升沿，FPGA 器件将该字节配置数据锁存；然后 FPGA 器件驱动 RDYnBSY 为低电平，表示它正在处理该字节信息，配置过程可以通过 nCS 和 CS 引脚暂停。

当 RDYnBSY 为低电平时，FLEX 10K 器件利用其内部振荡器，其频率一般为 10MHz，在其内部将每 1B 的配置数据串行化。当 FLEX 10K 器件准备接收下一个配置数据时，就使 RDYnBSY 变高电平。单片机检测该高电平信号后，送出下 1B 的数据。这一过程一直持续到全部数据配置完成。在配置过程中，系统需要进行实时监测，一旦出现错误，nSTATUS 将被拉至低电平，系统必须能识别出这个信号，并重新启动配置过程。

对 SRAM 加载配置数据常常采用下载电缆的配置方式，这种方法简单易行，只需直接将配置数据通过下载电缆由计算机下载至芯片，可以很方便地修改系统功能，因此被广泛应用在实际系统开发设计阶段。

但对于已经设计完的应用系统，如果每次掉电后都要通过计算机下载配置会带来很多不便。因此，如果在对系统重新上电时，系统本身能自动加载可编程逻辑器件的编程文件，从而对可编程逻辑器件进行配置，这样就省去了通过手工由下载电缆对器件进行配置的过程。该方法的前提是必须在应用系统上加存储器保存器件的编程文件，以供系统自动加载时使用。实现的方法常用的有两种，以 Altera 公司的 FLEX 10K 系列器件为例。

一种是采用主动配置的方法，即在系统对 SRAM 加载配置数据时，由可编程逻辑器件自身控制整个配置过程。FLEX 10K 系列常用的主动配置是 AS(主动串行)方式，即将编程文件存放在 Altera 公司的 EPC 系列专用存储器中，在芯片上电时由 FLEX 10K 系列器件控制整个配置过程，实现将编程文件串行地送到 FLEX 10K 器件的 DATA0 脚进行配置，且在配置结束后自动进行器件的初始化过程，且进入用户状态。该方法简单方便，不需要其他的外围控制器，由 FLEX 10K 器件自身引导整个配置过程，但是必须采用专用存储器放置配置数据，而对专用存储器的编程也需要专用的编程硬件，相对来说成本较高。

另一种是采用被动配置(PS 或 PPA、PPS)的方式。编程文件可以放在通用程序存储器中，如 EPROM、E^2PROM 或 Flash 存储器中，在 FLEX 10K 器件上电后，由芯片外部控制器自动地从通用存储器中读出编程文件并送到 FLEX 10K 器件进行配置，数据传送方式可以为串行，也可以为并行。串行传送时，即 PS 方式，配置数据送至 FLEX 10K 器件的 DATA0 脚；并行传送时，即 PPA 或 PPS 方式，配置数据送至 FLEX 10K 器件的 DATA0 ~ DATA7 脚。

1.2.3 电路设计

AT89C52 对 FLEX 10K 并行配置的硬件电路示意图如图 1-6 所示。经 MAX + plus Ⅱ编译生成的配置文件(. sof)通过格式转换成为 rbf 格式的文件，被存储在图中所示的存储器中。当使用 PPA 配置方式时，需要将 MSEL1 和 MSEL0 锁定为高电平。为了不使 DCLK 出现不确定信号，必须将其经过 1kΩ 电阻上拉到 V_{CC}。在采用 PPA 配置方式时，nCS 和 CS 两个片选信号只需用一个。因此，如果采用其中一个作为片选信号时，另一个必须将其直接置为有效位。如果选用 CS 作片选信号来控制配置时，nCS 必须接地。如果选用 nCS 作片选信号来控制配置时，CS 必须接高电平。本实例中采用后者。nRS 为读选通输入信号，低输入时引导 FLEX 10K 器件将 RDYnBSY 信号置于 DATA7 脚。当 nRS 不用时，必须将其置为高电平。nCE 为 FLEX 10K 器件的使能输入，nCE 为低电平时使能配置过程，而且为单片配置时，nCE 必须始终为低电平。由于本实例为单片配置，故将 nCE 直接接地。然后将 FLEX 10K 的 nCONFIG、CONF _ DONE、nSTATUS、RDYnBSY 分别接到 AT89C52 的 P1.7、P1.4、P1.5、P1.3 引脚上。DATA[7..0]接到 AT89C52 的 P0.7 ~ P0.0。nWS 为写选通输入，由低电平到高电平的跳变时，锁存 DATA[7..0]脚上的字节数据。要注意的是，nSTATUS 和 CONF _ DONE 脚由于是双向漏极开路端口，所以在做输出使用时，应该经过 1kΩ 的电阻上拉

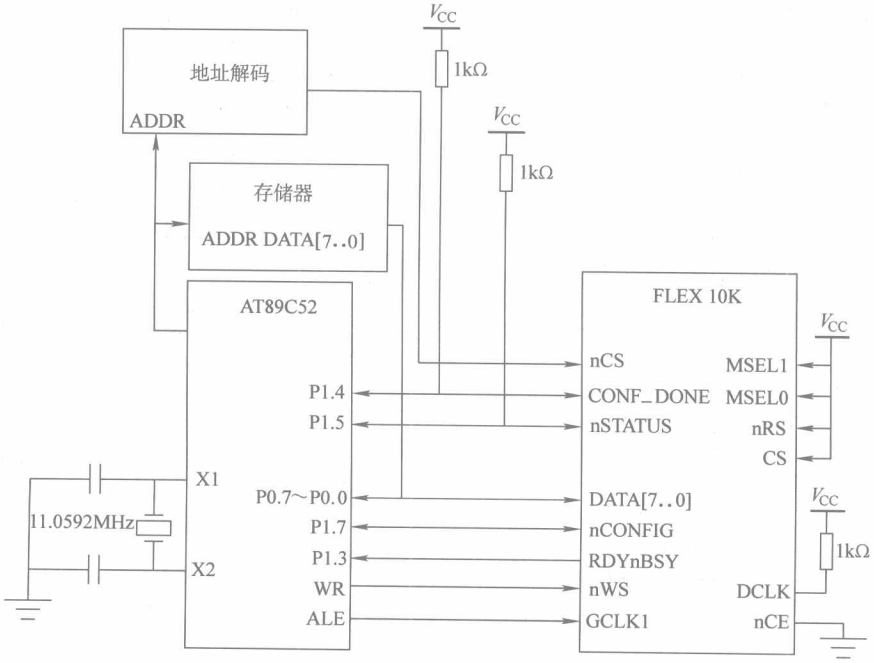


图 1-6 AT89C52 对 FLEX 10K 并行配置的硬件电路示意图

到 V_{CC}。

1.2.4 程序设计

单片机配置 FPGA 的程序流程如图 1-7 所示。

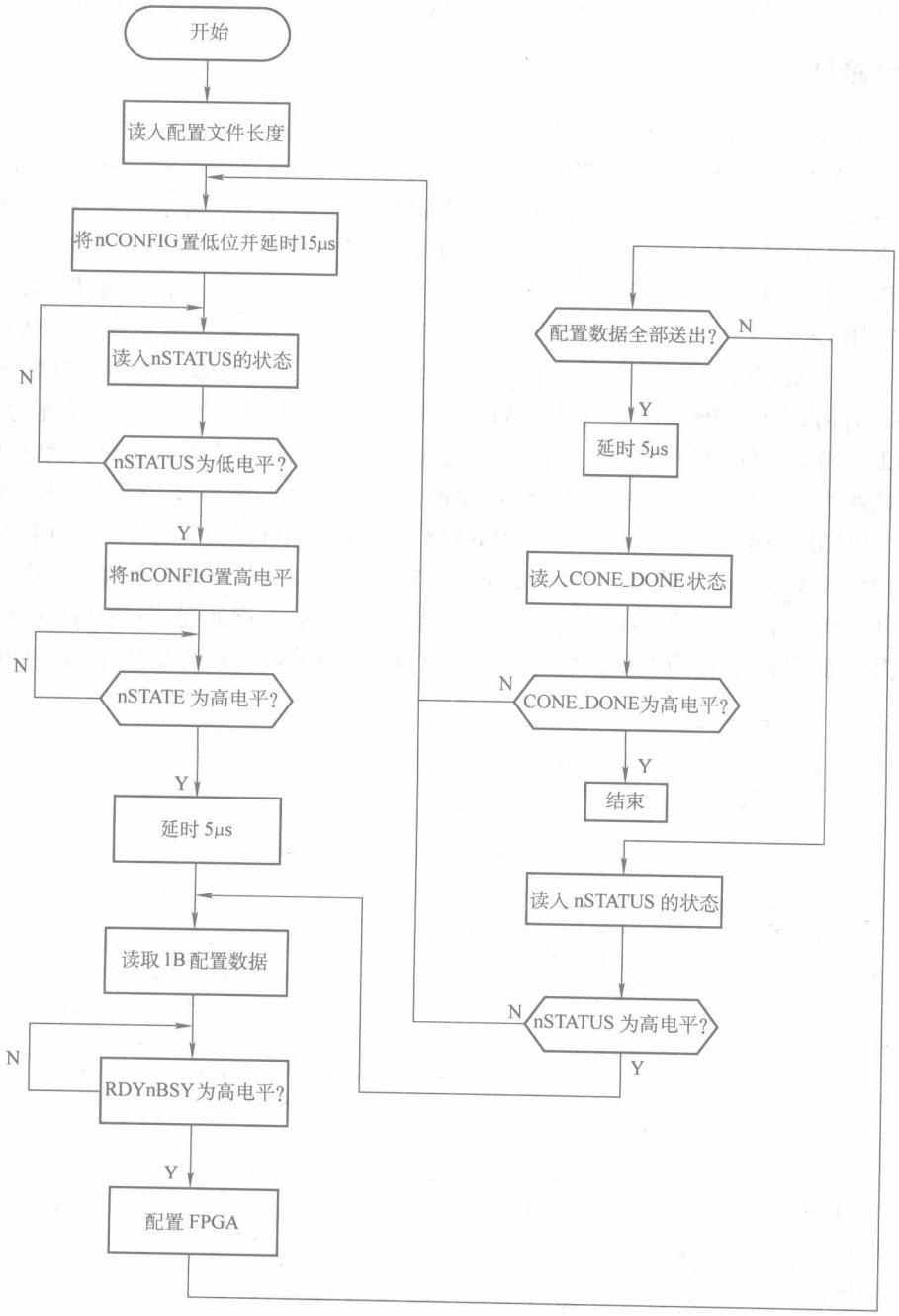


图 1-7　单片机配置 FPGA 的程序流程

单片机并行配置 FPGA 的汇编程序如下：