时钟芯片 DS1302 的原理及其 Proteus 仿真设计

欧阳乔
(无锡商业职业技术学院 江苏 无锡 214153)

摘要:本文详细介绍了实时时钟芯片 DS1302 的结构原理以及 C51 软件设计,同时给出了基于 Proteus 用 DS1302 设计的串行 LED 显示的日历/时钟电路,说明了它的仿真方法。

关键词:DS1302;单片机;Proteus;仿真

一、引言

在控制及数据采集中经常需要进行实时记录,采用实时时钟芯片提供时钟数据是比较方便的一种方法,常用的实时时钟芯片有 DS12B887、DS1302、PCF8563 等。本文介绍的实时时钟芯片 DS1302 是 Maxim/Dallas 公司的一种具有涓细电流充电能力的芯片,除具备一般时钟芯片的主要功能外,其主要特点是采用同步串行数据传输,可为掉电保护电源提供可编程的充电功能,并且可以关闭充电功能;低功耗,在保持状态时,功率小于 1mW;采用普通 32.768kHz 晶振。

二、DS1302 结构与工作原理

DS1302 芯片内含有一个实时时钟/日历和 31 字节静态 RAM。实时时钟/日历电路提供秒、分、时、星期、日期、月、年的信息,每月的天数和闰年的天数可自动调整,时钟的运行可以采用 24 小时或带 AM/PM 指示的 12 小时格式。RAM 为用户提供备用,其访问可视作与时钟/日历统一编址。通过简单的串行接口与单片机进行连接,仅需要根线。数据可单字节形式或多达 31 字节的多字节形式传送到时钟或从其中读出。

(一)引脚功能(如图 1)

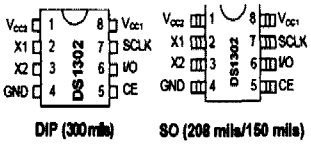

图 1 DS1302 引脚图

VCC1,VCC2:电源引脚,其中 VCC2 接主电源,VCC1 接备用电源。

GND:接地。

I/O:数据输入/输出端。

SCLK:串行同步时钟输入端。

X1,X2:32.768kHz 晶振引脚,内部有振荡器。若采用外部振荡信号时,32768Hz 的信号从 X1 脚输入,X2 脚悬空。

CE:片选端,高电平有效,在有些早期资料中该引脚标成/RST,实际功能一样。

CE 端接通控制逻辑,当其为"0"时,I/O 引脚变为高阻截状态,所有的数据传送中止,当其为"1"时,允许数据传送。CE 由"0"至"1"时,SCLK 必须为"0"。

DS1302 的内部主要由移位寄存器、指令和控制逻辑、振荡分频电路、实时时钟以及 RAM 组成。每次操作时,必须首先把 CE 置为高电平,再把提供地址和命令信息的 8 位装入移位寄存器。数据在 SCLK 的上升沿串行输入。无论是读周期还是写周期发生,也无论传送方式是单字节还是多字节,开始 8 位将指定内部何处被进行访问。在开始 8 个时钟周期把含有地址信息的命令字装入移位寄存器之后,紧随其后的时钟在读操作时输出数据,在写操作时输入数据。

7	6	5	4	3	2	1	0
1	RAM / CK	A4	A3	A2	A1	A0	RD / WR

图 3 DS1302 命令字节格式

(三)命令字节

如图 3 所示,每一数据传送由命令字节开始。最高有效位必须是"1"。如果它是零,禁止写 DS1302。位 6 为逻辑为"0"指向时钟/日历操作,"1"指向 RAM 操作。位 1 至 5 指定被访问的时钟/日历寄存器单元或 RAM 单元的地址,若将位 6 也看成地址的话,则两者相当于统一编址,时钟/日历在低地址空间,RAM 处于高地址空间。最低位指定操作方式,"1"为写,"0"为读。命令字节总是从最低位开始输入。

(四)命令地址及数据格式 各寄存器及 RAM 单元的命令地址和数据格式如图 4 所示。命令地址信息中,对相同寄存器或 RAM 单元的读写操作命令地址都只相差一个位即读写控制位的值,其他位相同。DS1302 有 9 个寄存器和 31 个 RAM 单元,其中有 7 个寄存器与日历、时钟相关,存放的数据位为 BCD 码形式,1 个写保护寄存器,1 个电源充电控制寄存器,31 个 RAM 单元可作为单片机普通的片外串行扩展 RAM 使用。

READ	WRITE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	RANGE
81h	80h	CH	10 Seconds			Seconds				00-59
83h	82h		10 Minutes			Minutes				00-59
85h	84h	12/24	0	AM/PM	Hour	Hour				1-12/0-23
87h	86h	0	0	10 Date		Date				1-31
89h	88h	0	0	0	10 Month	Month				1-12
8Bh	8Ah	0	0	0	0	0	Day			1-7
8Dh	8Ch	10 Year				Year				00-99
8Fh	8Eh	WP	0	0	0	0	0	0	0	—
91h	90h	TCS	TCS	TCS	TCS	DS	DS	RS	RS	—
C1h	C0h									00-FFh
C3h	C2h									00-FFh
C5h	C4h									00-FFh
.	.									.
.	.									.
.	.									.
FDh	FCh									00-FFh

图 4 DS1302 寄存器和 RAM 单元数据格式

秒寄存器的位 7 即 CH 位是时钟暂停位。当此位设置"1"时,时钟振荡器停止,DS1302 被置入低功耗备份方式,其电流小于 0.1uA。当把此位写成"0"时,时钟将启动。

小时寄存器的位是 12/24 小时方式选择位。当它为"1"时选择 12 小时方式,此方式下位 5 是 AM/PM 位,此位为"1"表示 PM。在 24 小时方式下,位 5 与位 4 合为小时的十位。

WP 是写保护位,所在寄存器为写保护寄存器。开始低 7 位置为"0",在读操作时总是"0"。在对时钟或 RAM 进行写操作之前,位 7 必须为"0"。若其为"1",则其它任何寄存器或 RAM 单元都被写保护。

BURST 是多字节方式操作指令,在保持 CE="1"时,该指令写入后,可进行连续多字节的读写操作。

(五)读/写时序

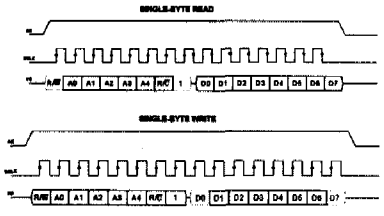

图 5 单字节读/写时序

DS1302 的读/写时序如图 5 所示。输入数据时,在 SCLK 上升沿,I/O 输入的数据位被有效接受。输出数据时,在 SCLK 下降沿时 I/O 输出数据位。要注意的是,在读 DS1302 时,当 8 位命令字写入后紧跟着的第一个下降沿芯片即开始输出数据。

三、Proteus 软件介绍

Proteus VSM 软件是来自英国 Labcenter Electronics 公司的 EDA 工具软件,和其它 EDA 工具相比,其革命性的突破是针对微处理器的应用,可以直接在基于原理图的虚拟原型上编程,并实现软件源码级的实时调试。如有显示及输出,还能看到运行后输入输出的效果。配合系统配置的虚拟仪器如示波器、逻辑分析仪等,Proteus 提供了完备的电子设计开发环境。Proteu 可从工程的角度直接看程序运行和电路工作的过程和结果。从某种意义上讲,解决了实验和工程应用脱节矛盾。

另外 Proteus 有比较丰富的元器件模型,使用者也可以自己建立新的元器件模型,支持 MCS-51 及其派生系列、ATMEL 的 AVR 系列、Microchip 公司的 PIC 系列及 Motorola 公司等多种 MCU,随着高版本的推出,现在 Proteus 对 ARM 支持功能也越来越完善。同时 Proteus 本身有 PCB 设计功能,同时又能够生成多种格式的网络表文件,供相应的专业 PCB 设计工具调用。对基于 MCS-51 及其派生系列单片机的设计系统,Proteus 可以很方便地与 Keil C51 集成开发环境连接。

本文就使用 Proteus 对 DS1302 的基本应用仿真设计作一介绍。

四、电路设计

在 Proteus 环境下建立电路很方便、直接到器件库选用器件并连接即可,操作方法类似 Multisim。电路在仿真过程中,各引脚信号的高、低电平的变化在电路上会以红、蓝两种颜色呈现出来,一定程度上使电路的检查调试提供了方便,比较直观。

(一)89C51 与 DS1302 的电路连接 DS1302 与单片机的接口很简单,电路基本固定,数据输出输出脚、同步脉冲输入脚、片选脚分别接单片机模拟串口脚即可。由于 Proteus 软件中对 DS1302 器件的标注仍采用旧的标法,所以此处 CE 脚标注成/RST,程序中对此脚也以/RST 进行标记,特此说明。

(二)显示驱动电路

此处显示驱动采用 3 片串行移位寄存器 74HC595 级联进行动态扫描显示驱动,接口简单。P2.0 为数据位,P2.1 数据同步时钟输出,P2.2 锁存信号输出,将已经移位进入三片 74HC595 的 24 位数据从输出端输出并锁存。U1 为段码输出,U2 和 U3 为位码输出,由此可看出本电路在显示数据输出时,位码的高 8 位先输出,再输出位码低 8 位,最后输出段码的 8 位。

(三)日历/时钟显示电路与仿真结果

显示电路采用两块 8LED 显示器,接线少,电路硬件清晰。上面一块为年、月、日显示,下面一块为时、分、秒显示

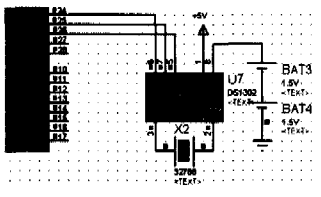

图 6 DS1302 与 89C51 连接电路

五、程序设计

C51 程序可以有两种加载到芯片的方法,一种是编译完成并生成 HEX 文件后,然后再在 Proteus 软件下进入单片机芯片模式编辑窗口,将 HEX 文件添加即可。还有一种方法是将 Proteus 和 Keil 软件联机调试,此处不再细述。顺便提一句,若用汇编语言的话,也可以在 Proteus 菜单中用添加源文件方式

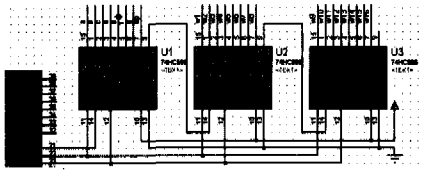

图 7 16LED 串行输出动态显动电路示驱

将汇编程序加入,软件即会自动进行编译、生成代码并添加到单片机芯片中去运行。以上 DS1302 电路的程序如下。

```
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
uint bdata OutByte;                    //74HC595 定义待输出字节变量
uchar DateTime[16];
uchar code Segment[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x40};//段码表
sbit Bit_Out=OutByte^7;
//74HC595 定义输出字节的最高位,即输出位
sbit Bout=P2^0;
//74HC595 位输出引脚
sbit Sclk=P2^1;
//74HC595 位同步脉冲输出引脚
sbit SLclk=P2^2;
//74HC595 锁存脉冲输出引脚
sbit DS1302_CLK=P2^4;                  //实时时钟时钟线引脚
sbit DS1302_IO=P2^3;                   //实时时钟数据线引脚
sbit DS1302_RST=P2^5;                  //实时时钟复位线引脚
sbit ACC0=ACC^0;
sbit ACC7=ACC^7;
#define DS1302_SECOND 0x80
#define DS1302_MINUTE 0x82
#define DS1302_HOUR 0x84
#define DS1302_WEEK 0x8A
#define DS1302_DAY 0x86
#define DS1302_MONTH 0x88
#define DS1302_YEAR 0x8C
void DS1302InputByte(unsigned char d)  //实时时钟写入一字节(内部函数)
{ unsigned char i;
  ACC = d;
  for(i=8; i>0; i--)
  { DS1302_IO = ACC0;                  //数据右移输出
    DS1302_CLK = 1;  DS1302_CLK = 0;
    ACC = ACC >> 1;
  }
}
unsigned char DS1302OutputByte(void)   //实时时钟读取一字节(内部函数)
{ unsigned char i;
  for(i=8; i>0; i--)
  { ACC = ACC >>1;                     //数据右移输入
    ACC7 = DS1302_IO;
    DS1302_CLK = 1;  DS1302_CLK = 0;
  }
  return(ACC);
}
void Write1302(unsigned char ucAddr, unsigned char ucDa)
//ucAddr: DS1302 地址, ucData: 要写的数据,RST 必须在 CLK 为 0 的情况下被置 1
{ DS1302_RST = 0;  DS1302_CLK = 0;  DS1302_RST = 1;
  DS1302InputByte(ucAddr);             // 地址,命令
  DS1302InputByte(ucDa);               // 写 1Byte 数据
  DS1302_CLK = 1;  DS1302_RST = 0;
}
unsigned char Read1302(unsigned char ucAddr) //读取 DS1302 某地址的数据
{ unsigned char ucData;
  DS1302_RST = 0;  DS1302_CLK = 0;  DS1302_RST = 1;
  DS1302InputByte(ucAddr|0x01);        // 地址,命令
  ucData = DS1302OutputByte();         // 读 1Byte 数据
  DS1302_CLK = 1;  DS1302_RST = 0;
  return(ucData);
}
void DS1302_GetTime()
{ unsigned char ReadValue;
  ReadValue = Read1302(DS1302_YEAR);
  DateTime[0] = (ReadValue&0x70)>>4; DateTime[1] = ReadValue&0x0F;
  DateTime[2] = 10;                                          //显示"-"
  ReadValue = Read1302(DS1302_MONTH);
  DateTime[3] = (ReadValue&0x70)>>4; DateTime[4] = ReadValue&0x0F;
  DateTime[5] = 10;                                          //显示"-"
  ReadValue = Read1302(DS1302_DAY);
  DateTime[6] = (ReadValue&0x70)>>4; DateTime[7] = ReadValue&0x0F;
  ReadValue = Read1302(DS1302_HOUR);
  DateTime[8] = (ReadValue&0x70)>>4; DateTime[9] = ReadValue&0x0F;
  DateTime[10] = 10;                                         //显示"-"
  ReadValue = Read1302(DS1302_MINUTE);
  DateTime[11] = (ReadValue&0x70)>>4; DateTime[12] = ReadValue&0x0F;
  DateTime[13] = 10;                                         //显示"-"
  ReadValue = Read1302(DS1302_SECOND);
  DateTime[14] = (ReadValue&0x70)>>4; DateTime[15] = ReadValue&0x0F;
}
void Initial_DS1302(void)              //秒寄存器的最高位为时钟暂停位
{ unsigned char Second=Read1302(DS1302_SECOND);
  if(Second&0x80)                      //此位为"1"时,时钟振荡器停止,为"0"启动
    DS1302_SetTime(DS1302_SECOND,0);
}
void OneLed_Out(uchar i,uint Location) //点亮一个 7 段 LED,位码取决于 Location
{ uchar j;
  OutByte=~Location;                   //先输出位码
  for(j=0;j<16;j++)
  {   Bout=Bit_Out;
      Sclk=1;  Sclk=0;                 //位同步脉冲输出
      OutByte=OutByte<<1;
  }
  OutByte=Segment[i];                  /* 再输出段码 */
  OutByte=OutByte<<8;
  for(j=0;j<8;j++)
  {   Bout=Bit_Out;
      Sclk=1;  Sclk=0;                 //位同步脉冲输出
      OutByte=OutByte<<1;
  }
  SLclk=0;  SLclk=1;  SLclk=0;         //一个锁存脉冲输出
}
void outdisplay(uchar disptimes)
{ uchar i;      uint Location=1;       //定义位码,初值为"0000000000000001"
  P1=0;
  for(;disptimes! =0;disptimes--)
  {   for(i=0;i<16;i++)                //16 个数码管显示一轮
      {   OneLed_Out(DateTime[i],Location);
          Location=Location<<1;
      }
      Location=1;
  }
}
void main()
{ uchar disptimes=100;
  Initial_DS1302();                    //初始化启动时钟
  while(1)
  {   DS1302_GetTime();
      outdisplay(disptimes);
  }
}
```

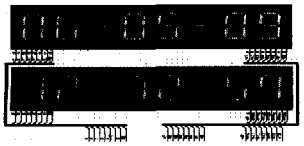

图 8 日历/时钟显示仿真结果

六、结束

DS1302 软硬件设计简单,时间记录准确,可以在各种测控系统中记录数据同时记录下数据出现的时间。由于其功耗较低,能长时间连续工作。到 Proteus6.7SP3 版本为止,器件库中只有 DS1302 一种串行实时时钟,DS1302 的电路软硬件设计在该软件下先行仿真通过后即可直接用于工程设计应用。

参考文献

[1]www.maxim-ic.com.cn, DS1302 数据资料.

[2]马忠梅等,单片机的 C 语言应用程序设计(第 3 版):北京航空航天大学出版社 2003.