

AT93C46/56/55 串行 EEPROM 及单片机程序

文章作者：湖北武汉高压研究所 张蓬鹤

文章出处：国外电子元器件

摘要：AT93C46/56/66 是 Atmel 公司生产的低功耗、低电压、电可擦除、可编程只读存储器，采用 CMOS 工艺技术制造并带有 3 线串行接口，其容量分别为 1kB/4kB，可重复写 100 万次，数据可保存 100 年以上。文中介绍了该存储器的引脚功能和指令时序，给出了 AT93C46/56/66 和单片机的接口应用电路和软件程序。

关键词：EEPROM 存储器 接口应用 程序 AT93C46/56/66

16 位单片机以其适于高速控制场合及功能多等优点已在工业控制领域中占领了一定的市场。由于 EEPROM 能在不脱离系统的情况下修改其存储单元中的内容，故在 16 位单片机中的应用愈来愈广泛。本文结合 16 位机的特点，详细介绍 AT93 系列 EEPROM 及其使用方法。

AT93C46/56/66 是 ATMEL 公司推出的低功耗、低电压电可擦除的可编程只读存储器。它采用 CMOS 技术和 Fairchild Semiconductor 公司的 MicroWire 工业标准 3 线串行接口，具有 1kB/2kB/4kB 的容量，并可通过 ORG 管脚配置成 128×8/256×8/512×8 或 64×16/128×16/256×16 等结构。该系列存储器可靠性高，能够重复写 100 万次，数据可以保存 100 年不丢失；采用 8 脚 PDIP/SOIC 封装和 14 脚 SOI 封装（SOI 封装为 JEDEC 和 EIAJ 标准），与并行的 EEPROM 相比，AT93C46/56/66 可大大节省印制板空间，且接线简单，因而在多功能的精密测试仪中具有广阔的前途。

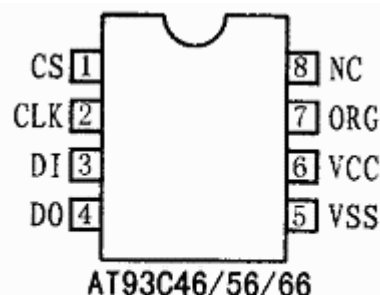


图 1 AT93C46/56/66 管脚图

1 引脚功能

AT93C46/56/66 存储器芯片的引脚排列如图 1 所示。各引脚的功能如下：

CS：片选信号。高电平有效，低电平时进入等待模式。在连续的指令之间，CS 信号必须持续至少 250 ns 的低电平，才能保证芯片正常工作。

CLK：串行时钟信号。在 CLK 的上升沿，操作码、地址和数据位进入器件或从器件输出。在发送序列时，CLK 最好不停止，以防止读/写数据的错误。

D I：串行数据输入。可在 C L K 的同步下输入起始位、操作码、地址位和数据位。

D O：串行数据输出。在 C L K 同步下读周期时，用于输出数据；而在地址擦 / 写周期或芯片擦 / 写周期时，该端用于提供忙 / 闲信息。

V S S：接地。

V C C：接 + 5 V 电源。

O R G：存储器构造配置端。该端接 V C C 或悬空时，输出为 1 6 位；接 G N D 时，输出为 8 位。

N C：空脚，不连接。

表 1 AT93C46/56/66 系统指令

指 令	起始位	操作码	地址位	数据位	备 注
			*8 *16	空	
READ	1	10	AnAo AnAo	空	从指定的单元读数
EWEN	1	00	11xxxx 11xxxx	空	允许写指令
ERASE	1	11	An~Ao An-1~Ao	空	擦除指定单元
WRITE	1	01	An~Ao An-1~Ao	D7~D0 D15~D0	写入存储单元
ERAL	1	00	10xxxx 10xxxx	空	擦除存储器所有单元
WRAL	1	00	01xxxx 01xxxx	D7~D0 D15~D0	写入存储器所有单元
EWDS	1	00	00xxxx 00xxxx	空	禁止写指令

表中，93C46 n=6;93C56n=7; 93C66n=8

2 指令及时序

A T 9 3 C 4 6 / 5 6 / 6 6 的指令如表 1 所列，各指令的具体含义如下：

(1) 擦 / 写允许指令 (E W E N)

由于在上电复位后 A T 9 3 C 4 6 / 5 6 / 6 6 首先将处于擦 / 写不允许状态。故该指令必须在所有编程模式前执行，一旦该指令执行后，只要外部没有断电就可以对芯片进行编程。

(2) 地址擦指令 (E R A S E)

该指令用于强迫指定地址中所有数据位都为“1”。一旦信息在DI端上被译码，就需使CS信号保持至少250ns的低电平，然后将CS置为高电平，这时，DO端就会指示“忙”标志。DO为“0”，表示编程正在进行；DO为“1”，表示该指定地址的寄存器单元已擦完，可以执行下一条指令。

(3) 地址写指令 (WRITE)

写指令时，先写地址，然后将16位的或8位数据写入到指定地址中。当DI端输出最后一个数据位后，在CLK时钟的下一个上升沿以前，CS必须为低，且需至少保持250ns，然后将CS置为高电平。需要说明的是：写周期时，每写一个字节需耗时4ms。

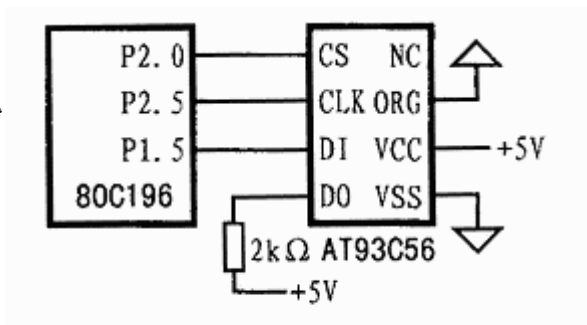


图2 AT93C56同80C196的硬件连接

(4) 地址读指令 (READ)

读指令用于从指定的单元中把数据从高位到低位输出至DO端，但逻辑“0”位先于数据位输出。读指令在CLK的上升沿触发，且需经过一段时间方可稳定。为防止出错，建议在读指令结束后，再输出2~3个CLK脉冲。

(5) 芯片擦指令 (ERASE)

该指令可将整个存储器阵列置为1，其它功能与地址擦指令相同。

(6) 芯片写指令 WRAL

该指令可将命令中指定的数据写入整个存储器阵列，其它功能与地址写指令相同。该指令周期所花费时间的最大值为30ms。

(7) 擦/写禁止指令 (EWDS)

使用该指令可对写入的数据进行保护，操作步骤与擦/写允许指令相同。

3 AT93C56的应用

3.1 AT93C56与80C196KB的连接

串行EEPROM芯片AT93C56同80C196单片机接口的硬件电路连接方法如图2所示。图中，由于MCS-96系列16位单片机的P1口为准双向口，因此，为了防止数据错位，在向其P1口写数据时，必须先将P1口置“1”。

3.2 软件编程

该读 / 写程序采用 P L / M 语言编写, 由于 P L / M 语言介于高级语言与汇编语言之间, 故其对数据、地址位的处理比较繁琐。其中 A D D R 为指定的地址单元, D A T A 为写入的数据, 在写数据、地址时均从高位开始。本程序已在仿真器上调试通过。且已被应用于电量测试仪中 (如接触电阻测试仪)。

```
DECLARE SETCS LITERALLY 'CALL BITSET . IO-PORT2, 0 ' ;  
  
DECLARE CLRCS LITERALLY 'CALL BITCLR . IOPORT2, 0 ' ;  
  
DECLARE SETCLK LITERALLY 'CALL BITSET . IOPORT2, 5 ' ;  
  
DECLARE CLRCLK LITERALLY 'CALL BITCLR . IOPORT2, 5 ' ;  
  
DECLARE SETDI LITERALLY 'CALL BITSET . IO-PORT1, 5 ' ;  
  
DECLARE CLRDI LITERALLY 'CALL BITCLR . IOPORT1, 5 ' ;  
  
EWEN: PROCEDURE PUBLIC;  
  
DECLARE I BYTE;  
  
CLRCS; CLRDI; CLRCLK;  
  
SETCS; SETDI; SETCLK;  
  
DO I=1 TO 2;  
  
CLRCLK; CLRDI; SETCLK;  
  
END;  
  
DO I=1 TO 2;  
  
CLRCLK; SETDI; SETCLK;  
  
END;
```

```
DO I = 0 TO 6;

CLRCLK; SETDI; SETCLK;

END;

CLRCLK; CLRCS; SETCS;

SETCLK; CLRCLK;

END EWEN;

READ: PROCEDURE (ADDR) PUBLIC;

DECLARE (ADDR, I, COUNT) BYTE;

CLRCS; CLRDI;

CLRCLK; SETCS;

DO I = 1 TO 2;

CLRCLK; SETDI; SETCLK;

END;

DO I = 1 TO 2;

CLRCLK; CLRDI; SETCLK;

DO COUNT = 1 TO 7;

BITOUT (COUNT) = SHR (BITOUT (COUNT), 1);

END;

CLRCS; CLRDI; CLRCLK;

CALL TIME (100);
```

```
SETCS; CLRC;

END READ;

WRITE: PROCEDURE (ADDR, DATA) PUBLIC;

DECLARE (ADDR, DATA) BYTE;

CLRCS; CLRDI; CLRCLK;

SETCS; SETDI; SETCLK;

CLRDI; CLRCLK; SETCLK;

SETDI; CLRCLK; SETCLK;

END;

DO I = -1 TO 8;

X=ROL (ADDR, 1);

ADDR=X; CLRCLK;

IF ((X AND 01H) = 01H) THEN SETDI;

ELSE CLRDI;

SETCLK;

END;

DO I = 1 TO 7;

CLRCLK; SETCLK;

END;

DO I = 1 TO 8
```

```
CALL BITSET (. IOPORT1, 7);

COUNT=COUNT-1;

CLRCLK; SETCLK;

BITOUT (COUNT) = IOPORT1;

END;

CLRCLK; SETCLK;

J=0;

AA: DO I= 1 TO 8;

J=J+1;

X=ROL (ADDR, 1;

ADDR=X; CLRCLK;

IF ((X AND 01H) = 01H) THEN SETDI

ELSE CLRDI;

SETCLK;

END;

IF J<>16 THEN DO;

ADDR =DATA;

GOTO AA;

END;

CLRCS; SETCS;
```

CALL TIME (1 0 0) ;

END;