

基于 MCS-51 单片机的 bootloader 设计解析

吴小安

(东南大学 成贤学院, 江苏 南京 210088)

摘要:传统的 ISP 技术有使用 JTAG 技术或者单片机内部驻留 BOOT 程序的方法, 这些方法或者成本较高, 或者无法适应大规模应用的需求。介绍了一种使用 CPLD 辅助实现单片机下载更新应用程序的方法, 应用该方法, 可以使用大容量的片外 FLASH 做为程序存储器, 在较低的成本下实现大规模应用中的程序在线更新。根据实际需要, 我们可以设计符合自己需求的 bootloader 程序。

关键词:单片机; CPLD; ISP

中图分类号: TP368.2

文献标识码: A

文章编号: 1671-5322(2011)03-0066-04

51 系列单片机是美国 Intel 公司在 1980 年推出的高性能 8 位单片机, 在我国的应用非常广泛。随着单片机技术的发展和广泛应用, 许多使用单片机的高新技术产品, 程序容量越来越大, 常常超过了单片机的 64kB 程序容量寻址空间, 而普通带 ISP(In-System Programming 在系统可编程, 指电路板上的空白器件可以编程写入最终用户代码, 而不需要从电路板上取下器件, 已经编程的器件也可以用 ISP 方式擦除或再编程)功能的单片机最大只能寻址到 64kB 的内部 FLASH, 这时就需要用到 C-51 的 BANK 技术^[1], 使用外部大容量 FLASH, 程序容量的问题可以用 BANK 技术解决了, 可是用户需要自己解决其 ISP 程序。本文详细介绍了一种利用单片机的 EA 脚的功能实现的 IAP(In Application Programming 在应用编程, 即在程序运行中编程, 就是片子提供一系列的硬件/软件上的机制, 当片子在运行程序的时候可以提供一种改变 flash 数据的方法)功能^[1]。

1 常用 ISP 技术的介绍

常用 ISP 技术有使用 JTAG 技术或者单片机内部驻留 BOOT 程序的方法。JTAG 技术比较简单, 利用市场上销售的 JTAG 工具下载程序, 就是需要花费一定的硬件成本; 单片机内部驻留

BOOT 程序的方法一般是厂家出厂时在单片机内部某个固定的地址驻留了 BOOT 程序, 在一定的条件下运行 BOOT 程序从串口下载用户程序。

1.1 JTAG 下载程序方法

这种方法在 51 单片机应用中比较少见, 一般只有特殊单片机才会有, 如意法半导体的 UPSD 单片机, 它支持 IEEE1149.1 JTAG 引脚(TCK, TMS, TDI, TDO), 使用 PSDsoft Express 可以用来下载应用程序。由于 UPSD 使用了 JTAG 技术, 并且内部集成了大容量的 FLASH 和 SRAM, 可以应用于大程序应用中。但由于 UPSD 相对于通用单片机应用成本相对还是比较高的, 在一些应用场合还是有一定的局限性。

1.2 单片机内部驻留 BOOT 程序应用方法

此方法目前应用比较广泛, 介绍一下 PHILIPS 公司带有 ISP 功能的单片机, 这种单片机在出厂时芯片中就有一个 ISP 服务程序, 其一般是位于某个 FLASH 空间中, 单片机中还包含两个特殊的 FLASH 寄存器: 引导向量和状态字节(注意它们在 FLASH 中不在 SFR 中)出厂时为 0xFC 和 0xFF(对于 P89C51RD2)。

例如 P89C51RD2 芯片系统复位时单片机检查状态字节中的内容。如果状态字为 0, 则转去 0000H 地址开始执行程序, 这是用户程序的正常

收稿日期: 2011-07-01

作者简介: 吴小安(1981-), 江苏兴化人, 助理工程师, 主要研究方向为电类相关专业基础及实验教学。

起始地址;如果状态字不为0,则将引导向量的值作为程序计数器的高8位,低8位固定为0x00。若引导向量为0xFC,则程序计数器内容为0xFC00,即程序转到0xFC00地址开始执行,而ISP服务程序进入了ISP状态,就可以使用专用下载软件串口下载应用软件了。

目前这种应用比较普遍,也比较可靠实惠,但一般单片机内部FLASH最大不超过64kB,所以在一些大型应用场合并不能满足需求。

2 基于片外大容量FLASH的IAP技术实践

2.1 硬件设计

设计中由于使用大容量的片外FLASH做为程序存储器,因此使用MCS-51单片机内部FLASH作为BOOT程序区。单片机的EA脚是程序存储器的内外部选通线,8052单片机内置有8kB的程序存储器,当EA为高电平并且程序地址小于8kB时,读取内部程序存储器指令数据,而超过8kB地址则读取外部指令数据。如EA为低电平,则不管地址大小,一律读取外部程序存储器指令。本设计中使用了CPLD,在CPLD中使用状态机实现EA脚的高低电平切换。硬件逻辑图如图1所示。

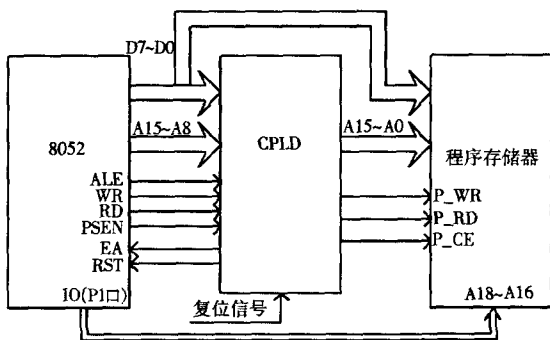


图1 硬件逻辑结构

Fig.1 Hardware logical structure

硬件设计中程序存储器的选用AMIC公司的A29040B,该存储器有512kB的存储空间,分为8个扇区,每个扇区大小为64kB,它的扇区分布如表1所示^[2]。

CPLD选用LATTICE公司的ISP4128-TQ128,该芯片使用3.3V供电,具有较低功耗;共有128个管脚,其中可使用I/O数达到92个,宏单元数有128个,可以完成较大规模的应用,并且价格适中。最重要的是该芯片的I/O口具有集

表1 存储空间扇区分布

Table 1 Storage space sector distribution

Sector	A18	A17	A16	Address Range
SA0	0	0	0	00000h - 0FFFFh
SA1	0	0	1	10000h - 1FFFFh
SA2	0	1	0	20000h - 2FFFFh
SA3	0	1	1	30000h - 3FFFFh
SA4	1	0	0	40000h - 4FFFFh
SA5	1	0	1	50000h - 5FFFFh
SA6	1	1	0	60000h - 6FFFFh
SA7	1	1	1	70000h - 7FFFFh

电极开路输出功能,可以根据外部上拉电阻的电平决定输出电平,并且I/O口输入兼容5V输入,这极大的方便了5V应用系统。

单片机则选用极为通用的8052系列的单片机STC89C52,它具有如下特点:

(1)8kB的内部FLASH,256B的内部RAM;

(2)可选择6CLOCK模式,外部时钟频率可以降低一半;

(3)ALE信号可以输出禁止;

可以看出单片机内部的8kB FLASH,256B的RAM可以在运行内部BOOT时使用,由于51单片机是基于哈佛结构,程序空间和数据空间分别位于不同的寻址空间,运行BOOT时,EA为高,内部8kB FLASH为程序空间,内部RAM为数据空间,使用CPLD将外部512kB FLASH分配在数据空间,这样就可以对它进行写操作了,CPLD切换程序空间和数据空间的代码如下3:

Ram_Zk_CE1: process (EAState, WR, RD, PSEN) —数据、程序空间的切换模块

```
begin
if( EAState = '0' ) then —程序空间
WrPrg <= '1';
RdPrg <= PSEN;
Else —数据空间
WrPrg <= WR;
RdPrg <= RD;
end if;
end process;
```

CPLD切换程序空间和数据空间如上述代码,可以看出是根据EA脚的状态信号(EAState)切换的,EA脚为高时,外部FLASH的WR脚为高,RD脚接CPU的PSEN脚,此时外部FLASH为程序空间;EA脚为低时,外部FLASH的WR脚接

CPU 的 WR 脚, RD 脚接 CPU 的 RD 脚, 此时外部 FLASH 为数据空间。EA 脚的切换在 CPLD 中使用的状态机技术来完成, 实现代码如下:

```
ResetData: process ( SCURRENT_STATE, Sig-
ZlReset, SCounter, SystemReset)
begin
  if( SystemReset = '1' ) then—外部芯片复位
  期间为 1
    SNEXT_STATE <= ST0;
  else
    case SCURRENT_STATE is
      when ST0 => EAState <= '1'; ResetState
      <= '0'; SClear <= '1'; SCounterEn <= '0';
      if( SigZlReset = '1' ) then—运行内部程序
      EA = 1 RESET = 0
        SNEXT_STATE <= ST1;—执行软复位后跳
        转到外部程序
      else
        SNEXT_STATE <= ST0;—继续执行内部程
        序
      end if;
      when ST1 => EAState <= '0'; ResetState
      <= '1'; SClear <= '0'; SCounterEn <= '1';
      if( SCounter = "1111111") then —EA = 0
      RESET = 1 形成复位脉冲的高电平
        SNEXT_STATE <= ST2;
      else
        SNEXT_STATE <= ST1;—循环等待复位操
        作完成
      end if;
      when ST2 => EAState <= '0'; ResetState
      <= '0'; SClear <= '1'; SCounterEn <= '0';
      if( SigZlReset = '0' ) then—复位完成后的状
      态 EA = 0 RESET = 0
        SNEXT_STATE <= ST3;—软跳转到内部程
        序
      else
        SNEXT_STATE <= ST2;
      end if;
      when ST3 => EAState <= '1'; ResetState
      <= '1'; SClear <= '0'; SCounterEn <= '1';
      if( SCounter = "1111111") then—形成复位
      脉冲的高电平
        SNEXT_STATE <= ST0;
```

```
else
```

```
SNEXT_STATE <= ST3;
```

```
end if;
```

```
when others => SNEXT_STATE <= ST0;
```

```
end case;
```

```
end if;
```

```
end process;
```

——指令复位 20 位计数器

```
SCounter: process ( TimeClk, SCounterEn,
SClear, SCounter, SystemReset)
```

```
begin
```

```
if( SClear = '1' or SystemReset = '1' ) then
```

```
SCounter <= "0000000";
```

```
elsif ( TimeClk'event and TimeClk = '1' ) then
```

```
if( SCounterEn = '1' ) then
```

```
SCounter <= SCounter + 1;
```

```
end if;
```

```
end if;
```

```
end process;
```

如上代码可知开机后 EA = 1, 系统的控制权在 89C52 BIOS 程序, 5 秒钟内判断是否有某 3 个键同时按下, 若有则系统的控制权仍在 89C52 BIOS 程序, 并进入下载状态, 选择串口下载。下载完成后向 CPLD 发一个指令, 使 SigZlReset 信号为 1, 启动 CPLD 的状态机继续进入下一个状态: 完成 EA = 0 并复位的过程, 之后将控制权交给运行 A29040 程序的 89C52, 开始正常运行。由此可见只要不想下载应用程序, 直接向 CPLD 发一个指令, 使 SigZlReset 信号为 1 就直接运行应用程序。

2.2 软件设计

在这个应用设计中有了 CPLD 后, 软件设计的功能相对来说简单多了, 软件流程如图 2 所示。

其实代码实现也比较简单, 因为是运行在内部 FLASH 中, 所以代码要求紧凑^[2], 功能单一, 核心代码如下 4:

```
ucRet = 1;
```

```
uiOverTime = 0xffff;
```

```
do
```

```
{
```

```
if( KeyHit() )
```

```
break;
```

```
} while( —uiOverTime ); //一定时间内检测
是否有键按下
```

```
uci = KeyGet();
```

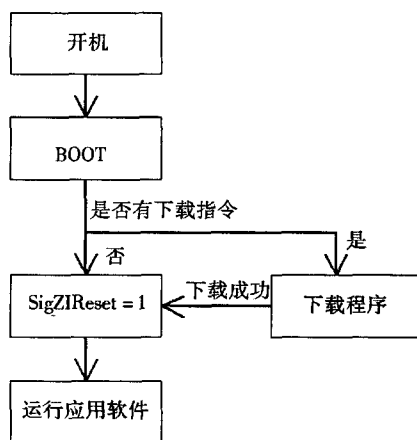


图2 软件流程

Fig.2 Software proces

```

if( uci == 0x3f) //有键按下,检测是否约定
按键
ucRet = 0;
if( ucRet) //不是约定按键,转到应用程序运
行
{
Beep(5,5,2);

```

```

XBYTE[0X7FFE] = 0xFE; //向 CPLD 发送
指令,使 SigZlReset 信号为 1

```

```

else //是约定按键,进入下载流程

```

```

Beep(50,5,1);

```

这里用指定按键代表是否有下载指令,开机后由于 CPLD 状态即置 EA 脚为高,所以从单片机内 0 地址开始运行,这个地址上是用户自己编制 BOOT 程序,也就是上面的程序。程序一开始检测是否有规定的按键按下,并规定超时时间,如果在规定的时间内按键没有按下,则发送指令给 CPLD 启动状态机进入下一个状态,使 EA 为低,复位系统,系统即开始运行单片机应用程序;如果检测到规定的按键按下,则系统开始下载应用程序到外部 FLASH。

3 小结

本文讨论的基于 CPLD 的大型应用平台的 ISP 技术具有经济实惠,实现简单,应用可靠,在现实产品中得到了很好的应用,是一个不错的设计。

参考文献:

- [1] 李全利,迟荣强. 单片机原理及接口技术[M]. 北京:高等教育出版社,2004.
- [2] 毛玉良. 微机系统原理及应用[M]. 南京:东南大学出版社,2006.
- [3] 戴先金. 51 单片机及其 C 语言程序开发实例[M]. 北京:清华大学出版社,2008.

Designing of the Bootloader Based on MCS-51

WU Xiao-an

(Chengxian College SouthEast University, Nanjing Jiangsu 210088, China)

Abstract: The traditional ISP technology uses JTAG technology or the methods of MCU internal resident BOOT program. The costs of these methods are high, or they can't adapt to the demand of large-scale application. This paper introduces an approach to realize the downloading and updating application program by the assistance of CPLD. By using this method, you can use high-capacity external FLASH as a program storage, and realize the updating of program on line in large-scale application with relatively lower cost. According to actual needs, we can design the bootloader program complying with our demands.

Keywords: MCU; CPLD; ISP

(责任编辑:沈建新)