

图1 针对高斯噪声

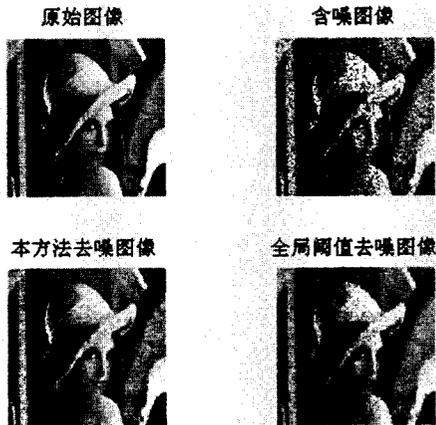


图2 针对非高斯噪声

实验结果表明,改进的自适应遗传算法寻优速度较快,且不容易出现过早收敛现象,获得的阈值更接近于最优阈值,将这种方法应用于小波阈值去噪领域,相对于传统方法,去噪效

果有了明显的改善。由于寻优效果随遗传算法各参数的改进会有所提高,因此算法性能的改良是下一步研究要解决的问题。如果对遗传算法性能进一步改善,并适当增加小波的分解层数,本文的去噪效果会更加明显。□

参考文献

[1] 沈贵明,刘劲松. 应用小波变换技术的图象噪声消除新方法[J]. 厦门大学学报(自然科学版), 2000,39(5): 612.  
 [2] 刘娟花,李福德. 基于小波变换的信号去噪研究[J]. 西安理工大学学报, 2004, 20(3): 1-2.  
 [3] Weyrich N., Warhola G T.. Wavelet shrinkage and generalized cross validation for image denoising [J]. IEEE Trans on Image Processing, 1998; 7(1): 82-90.  
 [4] 陈武凡. 小波分析及其在图像处理中的应用[M]. 北京: 科学出版社, 2003.  
 [5] 李水根, 吴纪桃. 分形与小波[M]. 北京: 科学出版社, 2002.  
 [6] Vidakovic B., Lozoya C. B.. On time-dependent wavelet denoising [J]. IEEE Trans on Signal Processing, 1998, 46(9): 2549-2554.  
 [7] Zhang X P., Desail M. D. Adaptive denoising based on SURE risk [J]. IEEE Signal Processing Letters, 1998; 5(10): 265-267.  
 [8] 董卫军, 卢燕宇. 利用小波分析进行图像去噪[J]. 微机发展, 2003, 13(7): 88-90.  
 [9] 郑德忠, 周颖, 荆楠. 基于GCV准则的小波图像去噪方法研究[J]. 仪器仪表学报, 2006, 27(5): 2269-2269.  
 [10] 王小平. 遗传算法[M]. 西安: 西安交通大学出版社, 2002.  
 [11] 陈明杰, 刘胜. 改进自适应遗传算法在函数优化中的应用研究[J]. 哈尔滨工程大学学报, 2007, 28(8): 875-879.

作者简介:李万臣(1963-),男,教授,硕士生导师,1984年毕业于哈尔滨船学院电子测量专业,研究方向为微弱信号处理和数据压缩技术,在信息通信和信号处理等方面,发表过多篇有价值的学术论文;邵嫒(1984-),女,研究生,研究方向为微弱信号处理和数据压缩技术。  
 收稿日期:2008-09-08(8435)

文章编号: 1671-1041(2009)01-0114-03

## DSP 外挂 FLASH 在线编程的研究与实现

李娟,王金海,王敏,薛青

(天津工业大学 信息与通信工程学院,天津 300160)

摘要: 本文采用 TMS320VC5509A 和 AM29LV800,以实际 DSP 系统开发为基础,给出了 DSP 与 Flash 接口的硬件连接和完整的 Flash 擦除与烧写程序,详细说明了 Flash 进行在线编程的实现步骤,完成了 DSP 系统复位上电后的自举引导。

关键词: TMS320VC5509A; Flash 存储器; Flash 编程  
 中图分类号: TN911.72 文献标识码: B

目前 DSP 已经被广泛的应用到了各种数字信号处理系统中,如果 DSP 系统要脱离仿真器运行的话,就必须将程序代码存储在非易失性存储器中。Flash 就是一种高密度、非易失性的电可擦写的存储器。它体积小、功耗低、可进行多次(一万余次以上)的电擦写,而且成本比传统的 EPROM 要低,很适合于低功耗、小尺寸和高性能的便携式系统。与硬盘相比,它还具有抗振动、抗干扰等优点<sup>[1]</sup>。于是怎样对 Flash 进行在线擦除和读写,以及怎样将 Flash 中的数据在上电时加载进 DSP 内部的 RAM,是 Flash 在 DSP 系统中得到应用要解决的基本问题。

本文以基于 TI 公司的 TMS320VC5509A 和 AMD 公司的 AM29LV800 Flash 开发系统为背景,详细介绍了该系统的硬件设计,Flash 的擦除和烧写软件设计,实现了 DSP 的自举引导。

### 1 硬件电路的设计<sup>[2][3]</sup>

本设计采用的是 TI 公司的一款 16 位高速低功耗的 DSP 芯片 TMS320VC5509A(以下简称 C5509A)和 AMD 公司生产的 Flash 存储器 AM29LV800。C5509A 片内拥有 128 × 16k 高速 RAM,性价比很高,被广泛应用于嵌入式手持设备、通信数据采集等领域。C5509A 的数据和程序空间统一编址,通过四个片选信号,可寻址 16K 字节的外部数据/程序空间,并可通过 EMIF(外部存储器接口)实现多种存储器的无缝连接。

C5509A 的外部存储空间被分成 CE0-CE3 四个空间,分别由 EMIF 的 CE0#-CE3#引脚管理。而 CE<sub>n</sub>(n=0-3)一般与外设的片选端连接。C5509A 与 AM29LV800 就是通过 EMIF

进行无缝连接的。AM29LV800 使用 3V 单电源供电,可使内部产生高电压进行编程和擦除操作;可以对其任一扇区进行读、写或擦除操作,而不影响其它部分的数据。DSP 与 Flash 的硬件连接图如图 1 所示。

C5509A 使用 GPIO 管脚实现在系统开机或复位时配置 A[13:1],D[15:0]的功能。上电复位时,当外部电路提供高电平信号输入到 GPIO0 管脚,也就是提供逻辑“1”给这个管脚时,A[13:0],D[15:0]被设置成外部扩展总线。反之,A[13:0],D[15:0]被设置成主机接口[HPI]。C5509A 采用统一的编址方式,即存储器的地址号没有重叠。但是,存储器宽度分为两种不同的情况,当存储器按照程序存储空间使用时,地址编码采用字节寻址方式,即每 8 位存储器占用一个地址编号,此时,A0 信号有效。而按照数据存储空间使用时,地址编码采用字寻址方式,即每 16 位存储器占用一个地址编号,此时,A0 信号无效。在本设计中,GPIO0 为缺省状态,即为高电平(逻辑 1),存储器按照数据存储空间使用。

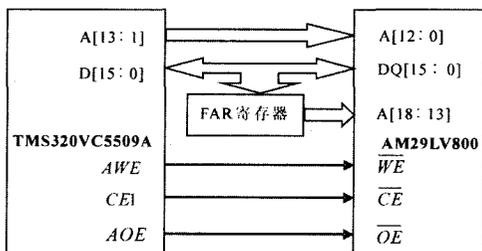


图 1 TMS320VC5509A 与 AM29LV800 硬件连接示意图

C5509A(PGE 封装)只有 14 条地址线,只能寻址 16K 字的异步存储器空间,而要对 AM29LV800 的全部 512k 字节地址需要按照分页方式访问,这个访问可以通过控制 CPLD 的 FAR 寄存器来实现。其中,Flash 的高位地址线由 C5509A 的 FAR 控制寄存器控制,FAR 可以驱动 Flash 的高位地址线处于一个固定的状态,从而实现分页的目的。上电复位时,FAR 寄存器的值被设置为 0x3F,此时,所有的高位地址线处于高电平状态,DSP 访问 Flash 的最后 16K 地址单元。此后随着复位的结束,用户程序开始工作,这样,就可以对 FAR 寄存器写值,改变 Flash 的高位地址,实现换页功能。

## 2 Flash 存储器的操作命令<sup>[4][5]</sup>

只要向 Flash 存储器的特定寄存器写入地址和数据命令,就可对 Flash 存储器编程。但要按一定的顺序操作,否则就会导致 Flash 存储器复位。AM29LV800 的写操作只能将 1 变成 0,而 0 变成 1 的操作必须通过擦除操作进行。所以每次写 Flash 之前,必须进行片擦除,使存储单元值变成 0xFFFF,才能进行编程。Flash 的擦除和编程命令都分为字操作和字节操作两种,在本设计中存储器按照数据存储空间使用,所以采用的是字寻址方式即字操作。

### 2.1 擦除命令

Flash 的擦除命令有片擦除和扇区擦除两种,都需要 6 个总线周期。用户可以通过查询标志数据线 DQ6 来确定擦除或者是编程是否完毕。AM29LV800 的擦除操作命令如表 1 所示。

表 1 Flash 擦除命令

周期	1(解锁)	2(解锁)	3(建立)	4(解锁)	5(解锁)	6(片擦除)	6(扇区擦除)
地址	0x555	0x2AA	0x555	0x555	0x2AA	0x555	扇区地址
数据	0x AA	0x55	0x80	0x AA	0x55	0x10	0x30

由表 1 可知,在第一个总线周期,向 0x555 地址的存储单元写入数据 0x AA;在第二个总线周期,向 0x2AA 地址的存储

单元写入数据 0x55;以此类推,完成六个总线周期的操作。操作完成后,Flash 存储器被完全擦除,内部数据恢复为初始状态,全为 0xFFFF。根据 Flash 的擦除命令,在本系统中,编写的 C 语言代码以完芯片的擦除如下:

```
void FLASH_write_byte (unsigned offset, unsigned int data)
{ * ((volatile unsigned int *) (FLASH_BASE_ADD + offset)) = data;
}

void erase_Flash ( )
{ FLASH_write_byte(0x555,0xAA);
.....
FLASH_write_byte(0x555,0x10);
}
```

在 C5509A 系统中,FLASH\_BASE\_ADD 在头文件中被定义为:

```
#define FLASH_BASE_ADD 0x200000
```

### 2.2 编程命令

该命令向 Flash 的指定地址写入数据,需要四个总线周期。AM29LV800 的编程操作命令如表 2 所示。

表 2 Flash 编程命令

周期	1(解锁)	2(解锁)	3(建立)	4(编程)
地址	0x555	0x2AA	0x555	编程地址
数据	0xAA	0x55	0xA0	编程数据

根据 Flash 的编程命令,在本系统中,编写 C 语言代码以完芯片的编程。

单字编程程序如下:

```
void Flash_page_prog(unsigned offset, unsigned int data)
{ FLASH_write_byte(0x555,0xAA);
.....
FLASH_write_byte(offset,data);
}
```

## 3 编程原理及具体实现过程

在 CCS 集成开发环境下,通过仿真器对 Flash 烧写的步骤如下。

第一步:编写用户程序,通过 CCS 编译、链接后生成可执行文件 dsp.out;

第二步:编写 hex55 命令文件 hex5509.cmd,将用户目标文件 dsp.out 转换为十六进制格式文件 dsp.hex。

在本例中 Hex55 命令文件如下:

```
dsp.out
-i
-map led.map
-memwidth 16
-romwidth 16
-bootorg PARALLEL
-o dsp.hex
-boot
```

第三步:经过 HEX55.exe 转化后的 hex 格式文件还是不能直接写入 Flash,所以还应将其转换为数组头文件,可以用 c 语言编写代码实现,这样就能通过写数组的方式将数据写入指定的存储单元,这也提高了数据烧写的速度。

要实现整片 Flash 的擦除和烧写,代码主程序如下:

```
void main( )
{ size = sizeof (dspscode)/sizeof (dspscode[0]); //求出要写入 Flash 的数组中数据的个数
erase_Flash(); //擦除 Flash
```

□经验交流□

```

Delay(100); //延时
Flash_page_prog (dspcode, size); //将数组数据写入
存储地址
}

```

由于 DSP 频率较高,时钟周期变小,导致 Flash 内部计数器计时的最大值小于内部算法所需要的时间,写操作就会出现异常。要解决这个问题,就要在每写入一个数据后进行一下延时。并且,当 Flash 整片擦除完成后,也需要延时后,才能对它进行写数据操作<sup>[6]</sup>。

### 4 实验结果及分析

当对 C5509A 应用系统重新上电复位后,在脱离仿真器的情况下, DSP 自举引导成功。可以观察到 Flash 中数据的变化情况,如图 2 和图 3 所示。

```

200000: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
20000F: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
20001E: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
20002D: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
20003C: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
20004B: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

```

图 2 Flash 擦除后存储单元的数据都为 FFFF

```

200000: 0000 05FA 0000 0000 0000 0598 0000 0100
20000F: 0CA4 00A5 0651 406C 0006 42C4 0876 0020
20001E: A961 997D FFEF 99C9 619A AB10 A961 991B
20002D: C902 0419 F3AB 10A9 6199 7DBF FF99 C961
20003C: A961 997D FFF3 99C9 619A AB10 A961 99EC
20004B: C961 9AAB 10A9 6199 EC0A 90C9 619A AB10

```

图 3 将数据烧进 Flash 后存储单元的数据情况

通过系统并行引导模式来烧写外部的 Flash 存储器,不仅

可以提高程序的调试效率,缩短系统的开发周期,同时还能够增加系统的灵活性和程序的可移植性。比采用 CCS 自带的烧录工具 Flashburn 要快很多,并且在烧写过程中只需在 CCS 环境下运行程序代码即可,减少了一些人为操作步骤,为进一步通过以太网实现 DSP 的远程调试和远程升级打下了基础。□

### 参考文献

- [1] 王旭智,张怀柱. TMS320VC5416 在系统并行引导的研究与实现[J]. 电子器件,2007,30(1)
- [2] Texas Instruments . TMS320VC5509 Fixed - Point Digital Signal Processor Data- book. 2003.
- [3] AMD . Am29LV800B Databook. 2000.
- [4] 吕晓明,段发阶. TMS320VC55x 系列 DSP 的 Flash 引导方法研究[J]. 国外电子元器件,2007,(6)
- [5] 叶海炳. 通过 TMS320VC5410 烧写 FLASH 实现并行引导装载的研究[J]. 集成电路应用,2004,(5)
- [6] 伍美俊,吴迈陵. TMS320VC5509 应用 Flash 存储数据的接口电路设计与编程[J]. 电子器件,2007,30(2)

作者简介:李娟(1983 - ),女,硕士研究生,专业为信号与信息处理,主要研究方向为 DSP 系统、信息检测和智能控制;王金海,男,教授,天津工业大学信息与通信工程学院副院长,硕士研究生导师,主要研究方向为信息检测与智能处理、嵌入式系统与应用、现代传感器及智能仪器。

收稿日期:2008 - 07 - 22(8382)

文章编号:1671 - 1041(2009)01 - 0116 - 02

## 基于 DCT 数字水印算法的 Matlab 实现

任晓扬, 韩 勇

(四川大学 电子信息学院, 成都 610064)

摘要: 数字水印技术是近年来图像保护技术研究的一个热点,通过对数字水印技术的空间域算法和变换域算法的分析,提出了一种基于 DCT 变换的图像数字水印算法 MATLAB 实现,实验结果显示出了 MATLAB 在数字水印技术处理中的方便性和高效性。

关键词: 数字水印; 离散余弦变换; matlab  
中图分类号: TP391 文献标识码: B

数字水印技术是近几年发展起来的一种信息安全技术,它是在数字信息(如图像、声音、视频信息等)中通过一定的算法加入不可见的标记,能有效保护信息的安全和著作者的所有权,近年来已成为信息信号处理及其应用领域众多学者非常关心的热点课题。

### 1 数字水印技术的复杂性

数字水印技术作为一个横跨计算机科学、生理学、密码学、数字通信等多门学科,并与 Internet 的发展密切相关的交叉科学。涉及到通信理论、编码理论、噪声理论、视听觉感知理论、扩频技术、信号处理技术、数字图像处理技术、多媒体技术、模式识别技术、算法设计等理论,数字水印的多学科性导致数字水印技术研究的难度和复杂性。所以,针对数字水印技术本身的跨学科特点,找出一种合适的编程工具,往往可以起到事半功倍的效果。

### 2 数字水印的一般原理

水印信号嵌入过程如图 1 所示,对原始信号(语音一维信

号、图像二维信号)进行变换域变换,并加入水印信息,然后进行相应变换域的反变换,得到含有水印信息的信号。



图 1 数字水印嵌入过程

图 2 所示检测抽取水印的过程是将原始信号与待检测信号同时进行变换域变换,比较两者的区别,进行嵌入水印的逆运算,得出水印信息。如果是可读的水印,那么就就此结束,如果

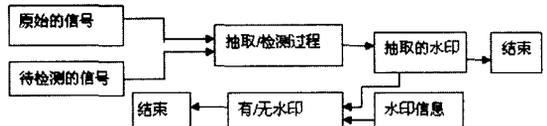


图 2 数字水印检测/抽取过程

是不可读水印,如高斯噪声,就将得出的水印与已知水印作比较,由相关性判断待检测信号含不含水印,故水印的检测有两个结束点。

### 3 数字水印技术的变换域算法

常见的典型的算法有空间域算法和变换域算法,除特别指明,这些算法主要针对图象数据。

#### 3.1 空间域算法

早期的数字水印算法从本质上来说都是从属于时空域,是直接信号空间上叠加水印信息。使用特定的密钥通过