

智能终端中正确高效地使用 Flash 的研究

赵 雷¹,程 瑶²

(1. 中国电子科技集团公司第10研究所,四川 成都 610000;
2. 重庆理工大学 电子信息与自动化学院,重庆 400050)

摘 要:随着当今宽带无线接入技术和移动终端技术的飞速发展,移动互联网得到了迅猛发展。Flash 闪存作为一种非易失性存储器在智能移动终端中得到了非常广泛的应用。文中以 S3C2410A 处理器和 RC28F256P30 Flash 为平台,讨论了 Flash 在移动互联网应用中的常见问题,对 Flash 的互斥访问、操作结果状态判断以及地址对齐问题进行了实验分析。通过分析和实验验证说明,对 Flash 的读、写及擦除等操作需要互斥进行,各种操作不能出现交叉,否则将无法得到预期的结果;在写入、擦除等操作后需要对操作结果进行全面的状态判断,以免出现错误的结果;在对 Flash 访问时需要注意地址对齐问题,否则可能会出现数据的错位。通过遵守上面的原则,可以避免在 Flash 使用中的错误。

关键词:移动互联网;Flash;互斥访问;状态判断;对齐访问

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2016)05-0128-04

doi:10.3969/j.issn.1673-629X.2016.05.27

Research on Correct and Efficient Application of Flash in Intelligent Terminal

ZHAO Lei¹,CHENG Yao²

(1. The Tenth Research Institute of China Electronic Technology Group Corporation,
Chengdu 610000,China;

2. School of Electronic Information and Automation,Chongqing University of Technology,
Chongqing 400050,China)

Abstract: Along with the flying development of broadband wireless access and mobile terminal, mobile internet has developed very rapidly. As a kind of non-volatile memory, Flash has been used widely in smart mobile terminals. The common problems for Flash in mobile internet application have been discussed in this paper based on Samsung S3C2410A processor and Intel RC28F256P30 Flash. It is proved through analysis and experiment that various operations to Flash should be mutually exclusive, they should not be interleaved, or you may not get the expected result. A full status check should be taken after word program, sector erase to avoid wrong result indication. And one should pay attention to the address alignment when using Flash, if the address is not aligned, the data wrote to Flash may get shifted unexpectedly. It can avoid the errors in using Flash according to above principles.

Key words: mobile Internet; Flash; exclusive access; status check; aligned access

0 引 言

移动互联网是指通过智能移动终端,采用移动无线通信方式获取网络服务。随着宽带无线接入技术和移动终端技术的飞速发展,移动互联网得到了迅猛发展,它已经渗透到了人们生活和工作的方方面面,改变了人类的生活方式。4G 时代的开启必将为移动互联网的发展注入更多的活力,移动互联网产业必将迎来

前所未有的飞跃^[1-4]。

Flash 闪存是一种广泛应用于智能移动终端中的非易失性存储器。它结合了 ROM 和 RAM 的优点,在可以断电保存数据的同时具有较高的访问速度。其存储特性相当于硬盘,这项特性正是闪存成为各类移动终端设备存储介质的基础^[5-10]。正确和规范地使用 Flash 是非常有必要的,如果在 Flash 使用中出错的话会导致系统出错,轻则功能失常,重则系统崩溃,无法

收稿日期:2015-06-30

修回日期:2015-10-13

网络出版时间:2016-05-05

基金项目:国家自然科学基金资助项目(51205434,51275551);重庆市教委科学技术研究资助项目(KJ1500935)

作者简介:赵 雷(1980-),男,硕士,高级工程师,研究方向为嵌入式系统设计。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160505.0817.042.html>

使用。

文中采用基于 S3C2410A 和 RC28F256P30 的实验平台对 Flash 应用中的互斥访问、操作状态判断以及对齐访问进行了实验分析^[11-14],提出了在 Flash 应用中需要注意的各种问题及可能导致的后果。

1 Flash 应用常见问题实验分析

1.1 Flash 互斥访问

在使用 Flash 的过程中,需要时刻注意各种 Flash 访问的互斥性,即对 Flash 的读、写和擦除操作不能出现交叉,比如说在 Flash 的某个扇区还没有擦除完成时不能进行 Flash 的写和读操作,否则会导致访问出错。由 Flash 芯片手册可知,当对 Flash 芯片写入擦除建立和擦除确认命令开始擦除处理后,芯片只能响应读取状态和擦除挂起命令,其他所有的命令都将被忽略。由此可以推断,当在 Flash 的擦除过程中又发起对 Flash 的读访问,读出的数据将会是 Flash 的状态而非存储在 Flash 上的数据,这显然不是用户所希望得到的结果。而如果在 Flash 擦除过程中进行写入操作,由于写入操作命令会被芯片忽略,数据将无法写入 Flash,同样也会造成出错。为了验证上述的分析,编写测试代码如下:

```
void testMutexAccess(void)
{
    UINT16 rdData1, rdData2, rdData3;
    UINT16 status=0;
    /* 正常擦除 */
    unlockSector(0x4100000);
    * (UINT16 * )0x4100000=0x0020;
    * (UINT16 * )0x4100000=0x00d0;
    while( ( status & 0x0080) !=0x0080)
    { status = * (UINT16 * )0x4100000; }
    * (UINT16 * )0x4100000=0x00ff;
    /* 正常写入后读出 */
    * (UINT16 * )0x4100000=0x0040;
    * (UINT16 * )0x4100000=0x1234;
    status=0;
    while( ( status & 0x0080) !=0x0080)
    { status = * (UINT16 * )0x4100000; }
    * (UINT16 * )0x4100000=0x00ff;
    rdData1 = * (UINT16 * )0x4100000;
    /* 擦除过程中读出 */
    unlockSector(0x41a0000);
    * (UINT16 * )0x41a0000=0x0020;
    * (UINT16 * )0x41a0000=0x00d0;
    * (UINT16 * )0x41a0000=0x00ff;
    rdData2 = * (UINT16 * )0x4100000;
    status=0;
```

```
while( ( status & 0x0080) !=0x0080)
{ status = * (UINT16 * )0x4100000; }
* (UINT16 * )0x4100000=0x00ff;
/* 擦除过程中写入再读出 */
unlockSector(0x4180000);
* (UINT16 * )0x4180000=0x0020;
* (UINT16 * )0x4180000=0x00d0;
* (UINT16 * )0x4100002=0x0040;
* (UINT16 * )0x4100002=0x5678;
status=0;
while( ( status & 0x0080) !=0x0080)
{ status = * (UINT16 * )0x4100002; }
* (UINT16 * )0x4100000=0x00ff;
rdData3 = * (UINT16 * )0x4100002;
logMsg( " rdData1 = %x rdData2 = %x rdData3 = %x. \n", rd-
Data1, rdData2, rdData3, 0, 0, 0);
```

测试程序首先正常擦除地址 0x4100000 处的扇区,然后在 0x4100000 地址处写入数据 0x1234,运行结果证明数据被正确写入。然后程序开始擦除 0x41a0000 地址处的扇区,并在擦除过程中读取上一步写入 0x4100000 处的数据,结果证明这种操作会出错,读出的不是刚才写入的数据而是 Flash 的状态(0x0)。最后擦除地址 0x4180000 处的扇区,并在擦除过程中往 0x4100002 地址写入数据 0x5678,结果证明数据无法写入,地址 0x4100002 处还是初始值 0xffff。

上面讨论的是操作执行过程中出现交叉操作的问题,还有另外一种交叉是 Flash 操作命令序列间直接出现交叉,其现象根据命令的不同而有差别。当向 Flash 写入擦除建立(0x0020)命令后,不是跟着写入擦除确认(0x00d0)命令而是写入编程建立命令(0x0040),这时会导致 Flash 状态出错而无法正确完成操作,如图 1 所示。

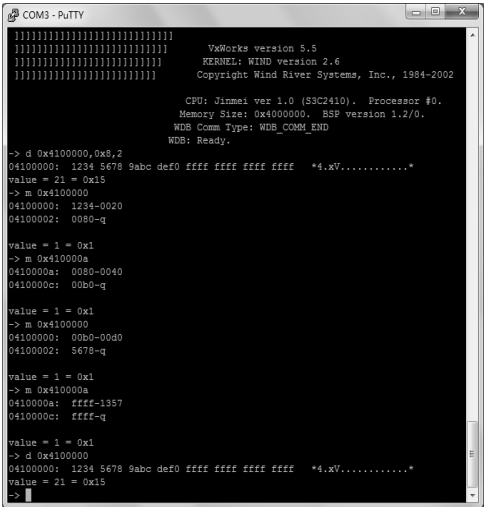


图 1 Flash 命令序列交叉访问试验

而如果首先向 Flash 写入编程建立命令(0x0040)再跟着写入擦除建立命令(0x0020)的话,擦除建立命

令会被当作数据写入 Flash 相应地址,而后续的擦除操作则无法完成,本来待写入的数据也无法正确写入。

1.2 Flash 操作状态判断

在 RC28F256P30 芯片内部有一个状态寄存器,用

于指示扇区擦除、数据写入等操作的结果,包括操作是否完成,是否成功,是否处于写保护状态以及命令序列是否正确,等等。状态寄存器的各个比特位的定义如图 2 所示。

Device Write Status	Erase Suspend Status	Erase Status	Program Status	V _{pg} Status	Program Suspend Status	Block-Locked Status	BEFP Status
DWS	ESS	ES	PS	VPPS	PSS	BLS	BWS
7	6	5	4	3	2	1	0
Bit	Name		Description				
7	Device Write Status (DWS)		0=Device is busy; program or erase cycle in progress; SR[0] valid. 1=Device is ready; SR[6:1] are valid.				
6	Erase Suspend Status (ESS)		0=Erase suspend not in effect. 1=Erase suspend in effect.				
5	Erase Status (ES)		0=Erase successful. 1=Erase fail or program sequence error when set with SR[4,7].				
4	Program Status (PS)		0=Program successful. 1=Program fail or program sequence error when set with SR[5,7].				
3	V _{pg} Status (VPPS)		0=Vpp within acceptable limits during program or erase operation. 1=Vpp<Vpplk during program or erase operation.				
2	Program Suspend Status (PSS)		0=Program suspend not in effect. 1=Program suspend in effect.				
1	Block-Locked Status (BLS)		0=Block not locked during program or erase. 1=Block locked during program or erase; operation aborted.				
0	BEFP Status (BWS)		DWS BWS 0 0 =WSM is busy and buffer is available for loading. 0 1 =WSM is busy and buffer is not available for loading. 1 0 =WSM is not busy and buffer is available for loading. 1 1 =Reserved for future use(RFU).				

图 2 状态寄存器描述

在对 Flash 芯片进行了擦除以及写入等操作后,必须对操作的结果进行判断确认避免出现各种错误。下面对操作中如果不进行状态判断可能出现的问题进行实验分析。

RC28F256P30 上电时各个扇区默认处于软件写保护 (Locked) 状态,在对扇区进行擦除操作前需要首先对待擦除扇区进行解锁操作,否则的话将无法正确完成对扇区的擦除。首先对基地址 0x4100000 处的扇区进行正常的擦除操作并观察状态,然后对 0x4180000 扇区不解锁而直接进行擦除操作,发现这样虽然状态也会指示擦除完成,但是擦除状态比特位 (Bit 5) 会提示擦除失败,而且扇区锁定状态位 (Bit 1) 会提示扇区处于锁定状态。结果如图 3 所示。

可见仅仅对操作执行状态位 (Bit 7) 进行判断是不够的,必须还要判断其他的状态位才能确认操作是否正确完成,否则可能会导致错误。

对 Flash 的各种操作命令序列必须连续且正确地输入,比如说扇区擦除操作,必须先在扇区地址范围内写入擦除建立命令 0x0020,然后跟着写入擦除确认命令 0x00d0。如果后面写入的命令不是 0x00d0,则 Flash 状态寄存器会报错,其 BIT4 和 BIT5 会同时置位,如图 4 所示。

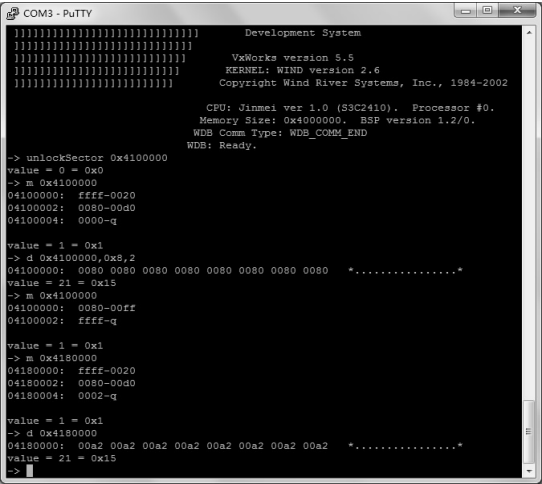


图 3 Flash 不解锁直接擦除状态判断

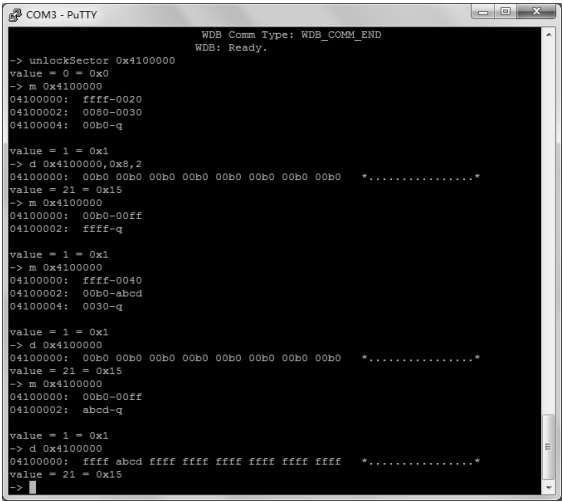


图 4 Flash 错误命令序列输入实验

对于 Flash 状态判断的另一个需要注意的地方就是,当出现错误状态后将要进行其他新的操作前要使用状态清除命令 (clear status) 清除之前的错误状态,否则该错误状态将会一直存在而影响后续操作的结果

判断,导致后续本来正确的操作被判断为出错。为了验证这个问题进行如下实验。首先故意输入错误的命令序列(0x0020,0x0030)导致状态寄存器提示出错,对Flash进行复位后进行Flash的写入,实验显示Flash的写入成功完成,但是读出状态仍然是开始错误命令序列的0x00b0错误状态,结果如图5所示。

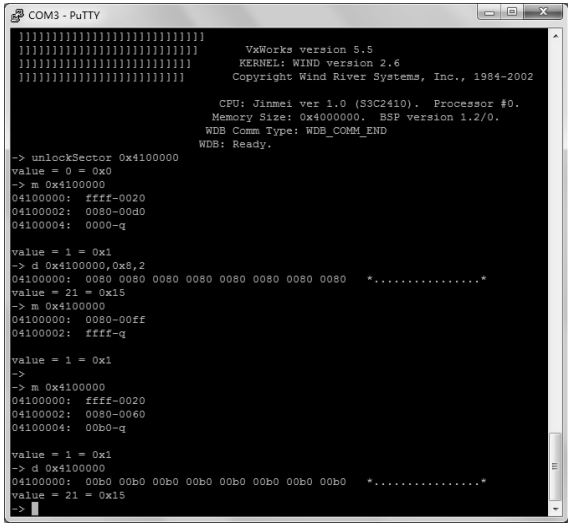


图5 Flash 错误状态清除实验

1.3 Flash 对齐访问

所谓对齐访问是指处理器在访问存储器时目标地址是所访问数据类型字节数的整数倍。文中设计选用的处理器S3C2410A作为ARM处理器要求对齐访问,即存取字数据时要求四字节对齐,目标地址的最低两位(A0、A1)会被忽略,而存取半字时要求两字节对齐,目标地址的最低一位(A0)会被忽略。对于Flash的操作,也要考虑对齐问题。文中设计的Flash是16位的,要求双字节对齐,如果使用了非对齐的地址访问,则可能得到与预期不一致的结果。比如说向Flash中0x4100001地址写入数据,因为地址是奇数,没有双字节对齐,则数据在写入Flash时会发生错位,因为最低位(A0)地址是会被忽略的,所以其实最终是向0x4100000和0x4100001两个字节地址写入了数据,而非预期的0x4100001和0x4100002。编写的写入测试程序如下:

```
STATUS programWord( UINT32 flashAddress,UINT16 data)
{
STATUS status=0;
* (volatile UINT16 *) (flashAddress)= 0x0040;
* (volatile UINT16 *) (flashAddress)= data;
while( ( status & 0x0080) !=0x0080)
{
status = * (UINT16 *) (flashAddress);
}
```

```
* (UINT16 *) (flashAddress)= 0x00ff;
}
```

2 结束语

移动互联网在人们的日常工作和生活中得到了广泛的应用,改变了人们的生活方式。Flash几乎存在于每个智能移动终端中,其正确使用显得尤为重要。为了能够在应用中正确高效地使用Flash,避免因为Flash使用不当而造成的各种损失,文中通过实验研究说明了在Flash使用中必须要注意的事项;对Flash的各种操作必须互斥进行,不能交叉,否则可能导致出错;在各种操作后必须进行全面的状态判断,不能只是简单地判断操作是否完成,以免得到错误的结果;在对Flash访问时要注意地址对齐,在进行非对齐的访问时会发生数据的错位。

参考文献:

[1] 罗军舟,吴文甲,杨明.移动互联网:终端、网络与服务[J].计算机学报,2011,34(11):2029-2051.

[2] 马化腾.移动互联网未来空间[J].中国报业,2014(1):68-69.

[3] 郭贺铨.移动互联网催生新型应用,引发产业变革[N].中国信息化周报,2014-01-20(5).

[4] 彭兰,颜云霞.移动互联网,一种思维与市场的变革[N].新华日报,2014-04-16(7).

[5] 邹逢兴.计算机硬件技术基础[M].北京:高等教育出版社,2000:182-196.

[6] 杜春雷.ARM体系结构与编程[M].北京:清华大学出版社,2004:165-229.

[7] 郑文静,李明强,舒继武.Flash存储技术[J].计算机研究与发展,2010,47(4):716-726.

[8] 潘立阳,朱钧.Flash存储器技术与发展[J].微电子学,2002,32(1):1-6.

[9] 战辉.基于FLASH的固态存储器[J].上海航天,2004,21(6):36-41.

[10] 唐彩彬,杨晓刚,张正璠,等.一种片上可测试Flash存储器的设计[J].微电子学,2015,45(2):237-240.

[11] SAMSUNG. S3C2410a user's manual[EB/OL]. 2004. <http://www.samsung.com>.

[12] INTEL. RC28F256P30 datasheet[EB/OL]. 2005. <http://www.intel.com>.

[13] Cheng Yao. Promote the Ethernet performance based on inline assemble[J]. Advanced Materials Research, 2014, 926-930: 2690-2691.

[14] Cheng Yao, Zhao Lei. Efficient Ethernet performance optimization based on DMA[J]. Lecture Notes in Electrical Engineering, 2014, 4: 325-331.