

基于 SEP0718 的 Linux Nand Flash 驱动设计

董玉刚, 谭守标, 徐超

(安徽大学电子信息工程学院, 安徽合肥 230039)

摘要:随着消费电子的发展,消费者对便携式电子产品的存储量要求不断增长。硬盘作为一种大容量的存储设备虽然能满足用户对存储量的要求,但是由于其体积功耗等缺点限制了其实际的应用,而 Nand Flash 具有体积小、功耗低、速度快及抗震等优异特性,使得采用 Nand Flash 作为便携式产品的存储介质成为可能,随着 Nand Flash 容量的不断增大,目前 Nand Flash 已广泛应用于便携式产品中。介绍了 Nand Flash 的硬件特点,分析了 Linux MTD 层(Memory Technology Device)的特性。并基于东南大学国家 ASIC 中心自主设计的嵌入式 ARM11 微处理器芯片 SEP0718,针对 Linux 嵌入式操作系统,设计和实现了 Nand Flash 驱动,就其一些功能如读、写、擦做出了详细的描述,并给出了 Nand Flash 驱动在非 DMA 和 DMA 下的实验结果。

关键词: Nand Flash; MTD; 坏块; 文件系统; 嵌入式系统

中图分类号: TP39

文献标识码: A

文章编号: 1673-629X(2011)04-0187-03

Design of Driver of Linux Nand Flash Based on SEP0718

DONG Yu-gang, TAN Shou-biao, XU Chao

(Institute of Electronics and Information Engineering, Anhui University, Hefei 230039, China)

Abstract: With the development of consumed electronic products, the consumers continuously increase the demand of storage capability of potable electronic products. As a mass storage device, though the hard disk can satisfy user's requirements, it is restricted to use in practical applications because of hard disk's large cubage and high power consumption. Nand Flash has the characteristics of small cubage, low power consumption, fast transmission speed and shakeable. As a result, Nand Flash is ready to become the main storage in potable electronic products. With increasement of the Nand Flash, Nand Flash has now been applied in most of the potable electronic products. Describe Nand Flash hardware features, and analyse the Linux MTD layer (Memory Technology Device) features. Based on embedded microprocessor chip SEP0718 researched and developed independently by Southeast University National ASIC Center, Nand Flash driven detailed is designed and achieved in the environment of Linux operating system, describe some of its function like reading, writing, wipe, and for driver of Nand Flash, gives under non-DMA and DMA results.

Key words: Nand Flash; MTD; bad block; file system; embedded system

0 引言

Nand Flash 是一种非易失性的存储介质,其结构是于 1989 年东芝公司提出的。它以快速的读写速度以及低廉的价格,及极高的存储密度,使其在嵌入式领域得到了广泛的应用。但由于 Nand Flash 具有复杂的硬件接口和巨大的存储容量,进而对软件的需求提出了更高的要求。

文中以东芯 SEP0718 微处理器^[1](东南大学自主设计的)为基础,分析了 K9F1G08U0B^[2]的内部结构,阐述了 Linux MTD 系统层次框架,并在此基础上进行了 Nand Flash 驱动。

1 Nand Flash 的硬件特点

Nand Flash 的数据是以 bit 的方式保存在 memory cell 中,一般来说,一个 cell 中只能存储一个 bit。这些 cell 以 8 个或者 16 个为单位,连成 bit line,形成所谓 byte(x8)/word(x16),这就是 Nand Device 的位宽。这些 Line 会再组成 Page。

Nand Flash 以页为单位读写数据,而以块为单位擦除数据。按照这样的组织方式可以形成所谓的三类地址:Block Address, Page Address, Column Address。

对 K9F1G08U0B^[2]来说:

$$1 \text{ page} = (2k + 64) \text{ Byte}$$

$$1 \text{ block} = (2k + 64) \text{ Byte} \times 64 \text{ pages} = (128k + 4k) \text{ Byte}$$

$$1 \text{ device} = (2k + 64) \text{ Byte} \times 64 \text{ pages} \times 1024 \text{ blocks} = (128M + 4M) \text{ Byte} = 1056 \text{ Mbits}$$

收稿日期:2010-08-31;修回日期:2010-12-13

基金项目:核高基项目(2009ZX01031-001-004)

作者简介:董玉刚(1983-),男,硕士,研究方向为嵌入式;谭守标,教授,博士,研究方向为计算机应用。

Column Address 表示数据在页中的地址,大小范围 0 ~ (2k+4) Byte,用 A[0:11]表示;

Page Address 表示页在块中的地址,大小范围 0 ~ 63,用 A[17:12]表示;

Block Address 表示块在 flash 中的位置,大小范围 0 ~ 1024,用 A[27:18]表示。

2 Linux MTD 结构

在 Linux 系统中提供了 MTD 系统^[3],用它来建立 Flash 针对 Linux 的统一、抽象的接口。MTD 将文件系统与底层的 Flash 驱动存储器进行了隔离,使开发者无须关心 Flash 作为字符设备或者块设备与 Linux 内核的接口。Linux 系统中的 Flash 设备驱动及接口可分为四层,从上到下依次是:设备节点、MTD 设备层、MTD 原始设备层和硬件驱动层,见图 1。

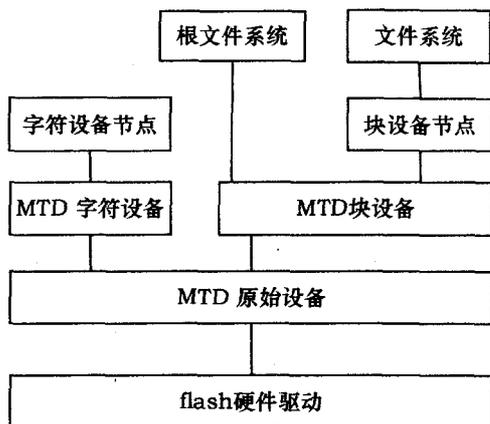


图 1 Linux MTD 系统

在引入 MTD 后,底层 Flash 驱动直接与 MTD 原始设备层交互,利用其提供的接口注册设备和分区。用于描述 MTD 原始设备的数据结构是 Mtd_info,这其中定义了大量关于 MTD 的数据和操作函数。Mtd_info 是表示 MTD 原始设备^[4]的结构体,每个分区也被认为是一个 Mtd_info,例如,如果有两个 MTD 原始设备,而每个上面有 3 个分区,在系统中就将共有 6 个 Mtd_info 结构体,这些 Mtd_info 的指针被存放在名为 Mtd_table 的数组里。

3 SEP0718 Nand Flash 驱动的实现

SEP0718 采用冯诺依曼结构的微处理,内嵌 ARM1136 内核,使用 0.18μm 标准 CMOS 的工艺设计。支持链表 DMA 传输和外部 DMA 传输^[5],支持双通道 8-bit Nand Flash 接口和 16bit Nand Flash,支持软件/硬件 ECC 校验(硬件可选 16, 24, 30bits BCH 校验),支持 4 级到 5 级地址,并支持 2kB、4kB 与 8kB page Nand Flash 启动。它由东南大学国家专用集成电路系统工程技术研究设计中心设计。

Nand Flash 控制器的系统接口见图 2。

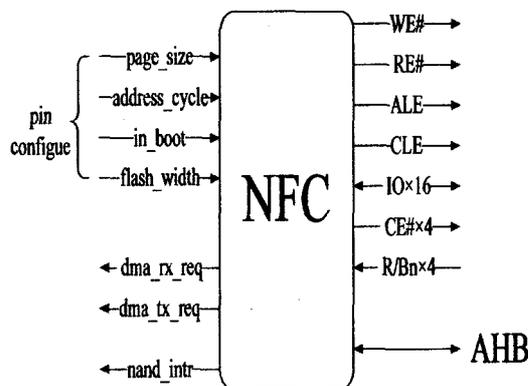


图 2 Nand Flash 控制器的系统接口

3.1 Nand Flash 读操作

向 NF_CMD 写入命令字(0x00)。向 NF_ADDR 分四次(四个地址周期)或五次(五个地址周期)写入地址周期。连续读 NF_SDATA,接收 1024+32 字节(16bit 纠错情况)或 1024+46 字节(24bit 纠错情况)或 1024+56 字节数据。注意,进入 ECC 模块的数据结构必须是

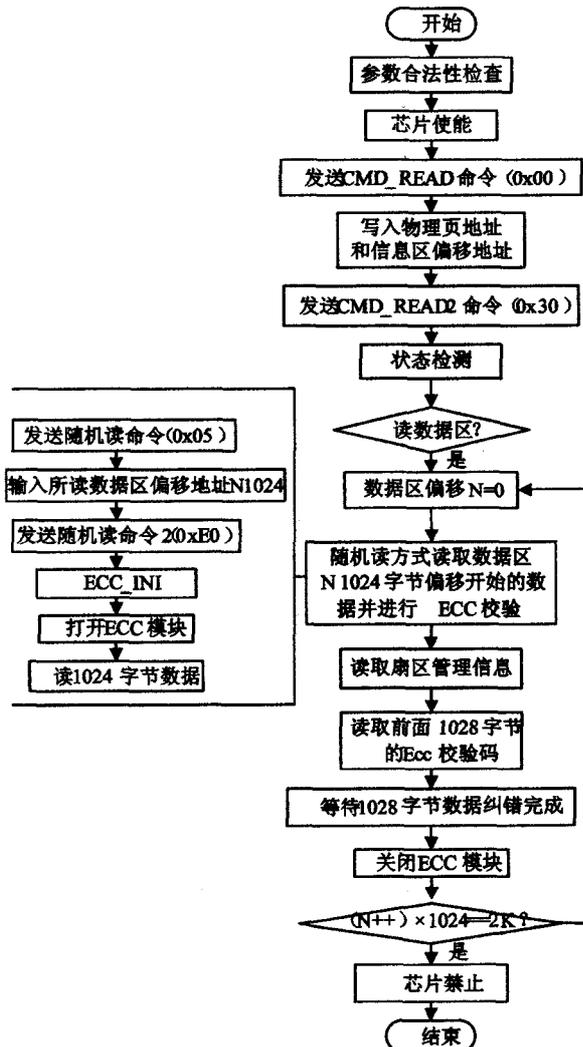


图 3 带 ECC 的读流程图

1024+3.5+ECC_PAREITO+ECC_PAREIT1 ...，这是主要原则。CFG[20]为1,关闭 ECC 模块。读 NF_STATUS[4],看 ECC 有没有 Fail。在 ECC 没有 Fail 的前提下,读 NF_STATUS[10:6],知道在 1024+3.5+ECC_PARITYx 这么多数据里面发生了几位错位。读 ERROR_ADDRx^[6],知道出错位的具体地址。

带 ECC 的读流程图见图 3。

3.2 Nand Flash 写操作

向 NF_CMD 写入命令字(0x80),向 NF_ADDR 分四次(四个地址周期)或五次(五个地址周期)写入地址周期。向 NF_SDATA 连续发送 1028 字节数据,注意,第 1028 字节的高 4 位不起作用。CFG[20]为1,关闭 ECC 模块。向 NF_SDATA 写入 ECC_PARITY 数据,注意有效位数。(16bit flash 的情况与此类似,总的原则就是要把第 1028 字节数据的高 4bit 用 ECC_PARITY0 替换)。向 NF_CMD 写入命令字(0x10),确认写入 Flash。等待芯片的 RnB 置高,即判断 NF_STATUS[0]。向 NF_CMD^[5]写入命令字(0x70),读芯片状态。读 NF_SDATA,取得芯片的状态数据,按芯片手册判断操作有没有成功。带 ECC 的写流程图见图 4。

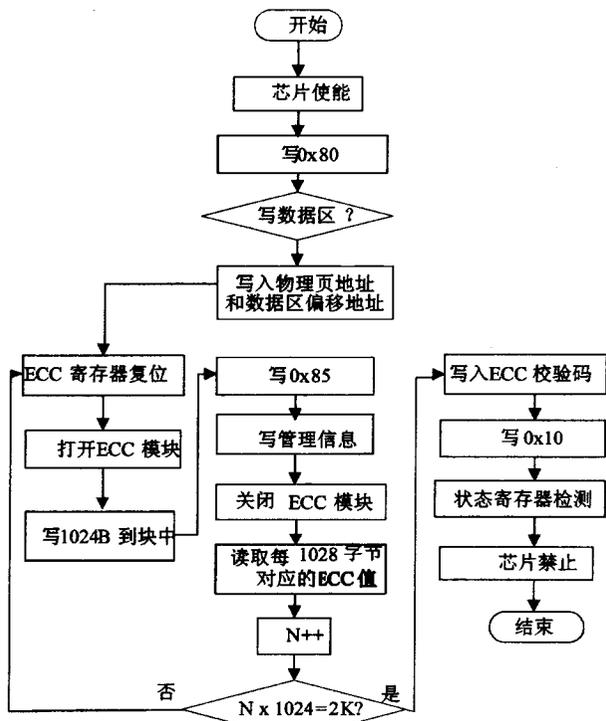


图 4 带 ECC 的写流程图

3.3 Nand Flash 擦操作

Nand Flash 是进行块擦除的。向 NF_CMD 写入命令字(0x60)。向 NF_ADDR 分三次(有三个 row address 周期的情况)或两次(有两个 row address 周期的情况)写入行地址。向 NF_CMD 写入命令字(0xd0)。等待芯片的 RnB 置高,限判断 NF_STATUS[0]。向 NF

_CMD 写入命令字(0x70),读芯片状态。读 NF_SDATA,取得芯片的状态数据,按芯片手册判断操作有没有成功^[7]。

3.4 Nand Flash 的驱动的的总体结构

综合前面所说的两点,Nand Flash 的底层驱动和框架驱动结合起来的的关系图如图 5 所示。

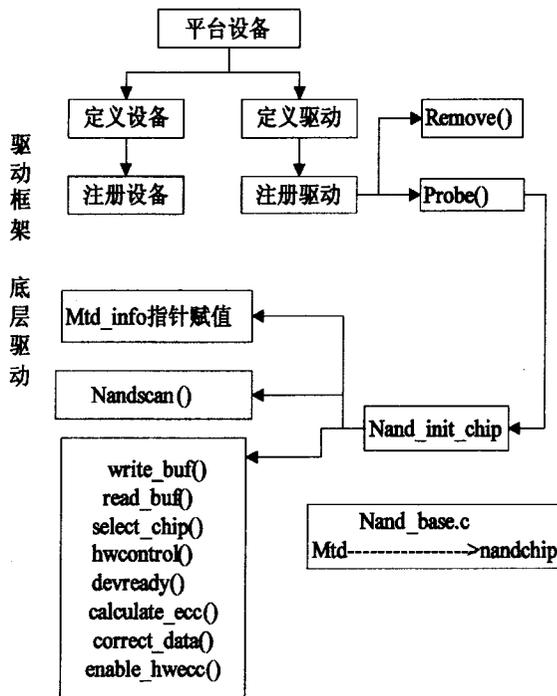


图 5 Nand Flash 的驱动的的总体结构

驱动的主体部分是从 Probe()^[8]开始的,Probe()中定义了 Nand_init_chip()函数,使得上层的框架驱动与底层驱动紧密地链接起来。

但是,要是作为嵌入式开发,光有驱动还是不够的,还需要将驱动当成模块,烧写进芯片中去,自然就少不了 Init()和 Exit()函数,所以在驱动函数的最后一一定要加上这两个函数,以及 Module_init(), Module_exit()向内核注册模块^[9]以及退出模块的函数。

4 测试与分析

系统用的是 yaffs 文件系统^[8],通过 Linux Kernel 中设计的 CONFIG—MTD—NAND—SEP0718—DMA^[10]宏开关,分别编译非 DMA 和 DMA 方式^[11]下的 zlmage^[12],各自加载测试。其中数据都是在相等的条件下,重新启动系统保持稳定后,分别对读、写、擦进行测试。以 20M 文件为例,测试结果见表 1。

由对比测试可以看出,读性能提高了约 1.8 倍,写性能提高约 1.3 倍,擦除速度接近原来的 2 倍。可靠性不易测试,但从理论上可知可靠性提高了。

不同的用例数据,将此数据与GPS接口输出的信号帧进行对比,经多次对不同数据进行比较,输入数据和输出信号一致;性能测试,利用主机软件设置UBS-To-GPS的帧发送频率,从1Hz调整到20Hz,都能正常输出;最后与被测设备联试,将主机输入数据与被测设备接收到的数据进行比较,数据也一致。目前模拟源已多次对被测设备完成了调试和测试,工作正常。

5 结束语

采用磁隔离技术的GPS信号模拟源满足相应的技术要求,性能稳定,使用方便。GPS信号输出波形好,使用磁隔离功率比光耦的小,其功率仅为光耦的1/10,若用的多将会省更多功率,另外安全可靠,在使用中从未对被测设备和本机造成损坏,工作正常。采用USB供电,不需单独的供电电源。模块体积小,携带方便。通过主机软件的灵活设置,可对被测设备进行功能、性能测试,正常和异常测试以及强度测试在内的全面测试。

参考文献:

- [1] 侯俊钦,吴小培,杨一军. 微伏信号放大系统设计[J]. 计算机技术与发展,2007,17(10):198-201.
[2] 张波,丁金华,孙秋花,等. 基于微控制器的隔离变送器

的设计[J]. 大连工业大学学报,2010,29(3):223-226.

- [3] 兰颖,李刚. iCoupler (R)数字隔离技术及其应用[J]. 电子技术应用,2006,32(11):89-93.
[4] Krakauer D. Data and Power Isolation in One Package Provides Total Isolation Solution[J]. Electronic Design, 2006, 54(13):33-33.
[5] Morrison D. Single Component Isolates Data And Power [J]. Power Electronics Technology, 2006,32(3):54-54.
[6] Conner M. Chip provides both signal isolation and 500 mW of regulated, isolated power[J]. Electrical Design News, 2008,53(12):15-15.
[7] 田野,桂欣,李一兵. 基于ADuM540x隔离通信电路的设计与实现[J]. 哈尔滨商业大学学报(自然科学版), 2009,25(5):574-577.
[8] 张念淮,江浩. USB总线接口开发指南[M]. 北京:国防工业出版社,2001.
[9] 许永和. USB外围设备设计及应用[M]. 北京:中国电力出版社,2002.
[10] 李海波,丛培田. C8051F340单片机在轴承故障诊断中的应用[J]. 沈阳理工大学学报,2008,27(4):13-15.
[11] 李腾飞. C8051F340的USBXpress开发包USB通信设计[J]. 单片机与嵌入式系统应用,2010(6):74-75.
[12] 尹杰君,姚旺生,喻济兵. 基于C8051F340数据采集系统的设计[J]. 计算机与数字工程,2010,38(10):159-162.

(上接第189页)

表1 以20M文件为例的测试结果(ms)

已有容量	10%		70%		90%	
	非DMA	DMA方式	非DMA	DMA方式	非DMA	DMA方式
读	6115	4724	6125	4691	6126	4852
写	9857	5618	11093	5945	11203	6013
擦除	833	452	817	471	793	466

5 结束语

文中是以SEP0718处理器为硬件平台,研究了Nand Flash驱动的执行效率。详细地讲述了Nand Flash驱动读、写、擦的执行过程,并在DMA与非DMA下比较了Nand Flash驱动的读、写、擦速度。根据实验表明,文中实现的Nand Flash驱动提供的速度在一般的嵌入式应用中已经足够。

参考文献:

- [1] SEP0718 Datasheet[M]. 南京:东南大学国家专用集成电路系统工程技术研究中心,2009.
[2] Flash Memory K9FIG08U0B Datasheet[M]. [s.l.]:Samsung Corp,2006.
[3] 宋宝华. Linux设备驱动开发详解[M]. 北京:人民邮电出

版社,2008.

- [4] 肖鹏,刘新宁,杜洪华. 基于AMBA总线的NANDFLASH控制器软硬件划分设计[J]. 现代电子技术, 2008, 31(2):2-3.
[5] 肖建. 一种在片上系统中实现Nand Flash控制器的方法[J]. 南京邮电学报,2005,27(5):2-4.
[6] 沈棋,时龙兴. 大容量NAND FLASH驱动及YAFFS文件系统WinCE下的实现[D]. 南京:东南大学,2007.
[7] 史斌,丁志刚,张伟宏. 基于PXA3xx处理器的NAND闪存DMA方案[J]. 计算机应用, 2009,29(8):2136-2142.
[8] 胡一飞,徐中伟,谢世环. NAND Flash上均匀损耗与掉电恢复在线测试[J]. 单片机与嵌入式系统应用,2007(3):5-8.
[9] 鲍国彬. 嵌入式工业控制系统的Flash文件存储系统[J]. 福建电脑,2006(2):130-131.
[10] Rubini A, Corbet J. Linux Device Dirver[M]. USA: O'Reilly & Associates Inc., 2001.
[11] Improving Performance Using Two-Plane Commands Introduction Technical Note[M]. [s.l.]:Micron Corp,2007.
[12] Fisher R. Optimizing NAND Flash Performance[R]. [s.l.]: [s.n.],2008.