

## 嵌入式系统中的存储技术（一）Flash 存储

摘要：随着嵌入式系统的发展，存储技术已经成为嵌入式技术中非常重要的一部分。不同的嵌入式系统需要根据实际应用需求和成本因素，来选择不同的存储技术及相应的存储介质。本文将列举一些常见的产品及应用，围绕 Flash Memory、Flash Disk、硬盘等存储技术进行分析和比较。

关键字：嵌入式，嵌入式 Linux，flash，存储卡，硬盘，存储介质，文件系统

### Flash

#### 1，Flash 的背景

嵌入式硬件系统一般都需要有软件的支持才能够正常工作，嵌入式系统需要为软件提供相应的存储空间。在以往的单片机系统内，一般使用 ROM(Read Only Memory)或 EPROM(Erasable Programmable Read Only Memory)存储程序。由于现有的嵌入式系统越来越复杂，原有的 ROM 由于容量、灵活性差等的限制，无法满足日益复杂的应用要求。



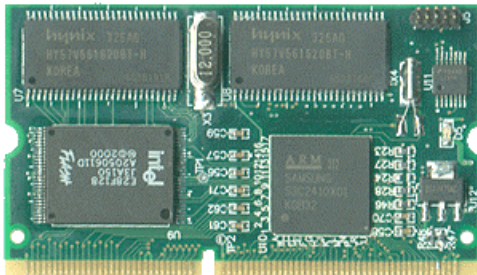
图一 Samsung NAND Flash 芯片

随着 Flash 技术的成熟和普及，一些单片机芯片（如 Freescale 的 8 位单片机 MC68HC908）已经开始采用片内 FLASH 取代过去常用的片内 ROM 或 EPROM，使单片机具有了在线编程写入或擦除的功能。

可以毫不夸张的说，Flash 已经是目前嵌入式最为主要的存储介质。早在二十多年前，Flash Memory 就已经被发明出来。在这么多年的发展过程中，Flash Memory 技术经过了多次变革和发展，但其变化的总体趋势一直都是：存储容量越来越大、数据读写速度越来越快、性能价格比越来越高。目前日益普及的手机、数码相机、数字摄像机、路由器、交换机等都已广泛利用 Flash Memory 作为永久性存储器，只不过由于它封装在设备内部，只有开发商和设计人员直接接触、了解它而已。随着优盘的普及，Flash Memory 正在被越来越多的普通用户熟悉和关注。

#### 2，Flash 的种类

Flash Memory 按内部构架和实现技术可以分为 AND、NAND、NOR 和 DiNOR 等几种，目前以 NOR flash 和 NAND flash 为主流。NOR 技术是由 Intel 公司 1988 年首先推出，而 NAND 技术是由东芝公司于 1989 年发明。NAND 技术在设计之初是为了数据存储应用，NOR 则是为了满足程序代码的高速访问，并且支持程序在芯片内部运行。目前关于两种技术的讨论很多。客观来看，二者各有优势和不足。NOR 工作电压低、随机读取快、功耗低、稳定性高；而 NAND 则写回速度快、芯片面积小，特别是容量大有很大优势。



图二 Intel NOR Flash 用于华恒科技的 HHARM2410 嵌入式核心板

大多数情况下 Flash 只是用来存储少量的代码，NOR flash 更适合用来满足这部分代码或更加复杂一些的系统存储。NOR flash 占据了容量为 1~16MB 存储芯片的大部分市场。以目前主流的嵌入式操作系统来看，对于 Embedded Linux，Vxworks，uC/OS 等，一般 2MB 容量的 Flash 即可满足系统、应用及一定数据存储的要求；而对于 Microsoft 公司提供的 WinCE 操作系统，则一般需要 32MB 以上的 flash 存储空间，以满足较为复杂的图形显示、多种字库、以及丰富的应用的支持。在这种情况下，容量更大的 NAND flash 对于控

制成本而言就是更好的选择了。

由于 NAND flash 比 NOR flash 更容易出现位交换(在某些情况下,一个比特位会发生反转或被报告反转了)和坏扇区,因此如果需要用 NAND flash 来存储操作系统、配置文件或其他敏感信息时必须通过 EDC/ECC 算法和初始化扫描、标记等措施来做一定的处理。这个问题对于用 NAND 存储多媒体信息时倒不是致命的。某种角度上来说,NAND 更适合于数据存储。目前一些移动存储设备就大量的使用到了 NAND flash,如 :Disk On Chip(DOC),Disk On Module,以及 Compact Flash、Secure Digital、PC Cards 和 MMC 存储卡等等。

### 3, Flash 的烧写和编程

flash 闪存是非易失存储器,可以对称为块的存储器单元块进行擦写和再编程。任何 flash 器件的写入操作只能在空或已擦除的单元内进行,所以大多数情况下,在进行写入操作之前必须先执行擦除。NAND 器件执行擦除操作比较简单,而 NOR 则要求在进行擦除前先要将目标块内所有的位都写为 0。

对于 Flash 进行编程,我们可以使用烧片器,但这种方式一般仅适合还未将器件焊接到 PCB 板上前使用。对于开发人员而言,在开发过程中需要不定时地对 Flash 进行擦除和烧写,以目前应用广泛的 32 位嵌入式系统为例,可以使用 BDM、JTAG 接口的调试器及在线仿真工具。

#### BDM 调试卡 :



图三 华恒科技提供的 BDM 调试卡及软件光盘

BDM, Background Debug Mode, 是嵌入式处理器为开发者提供了一种简捷、高效而极其廉价、低成本的底层软硬件调试方法,主要有 Coldfire、PowerPC 等嵌入式处理器支持。可以支持以下主要功能:

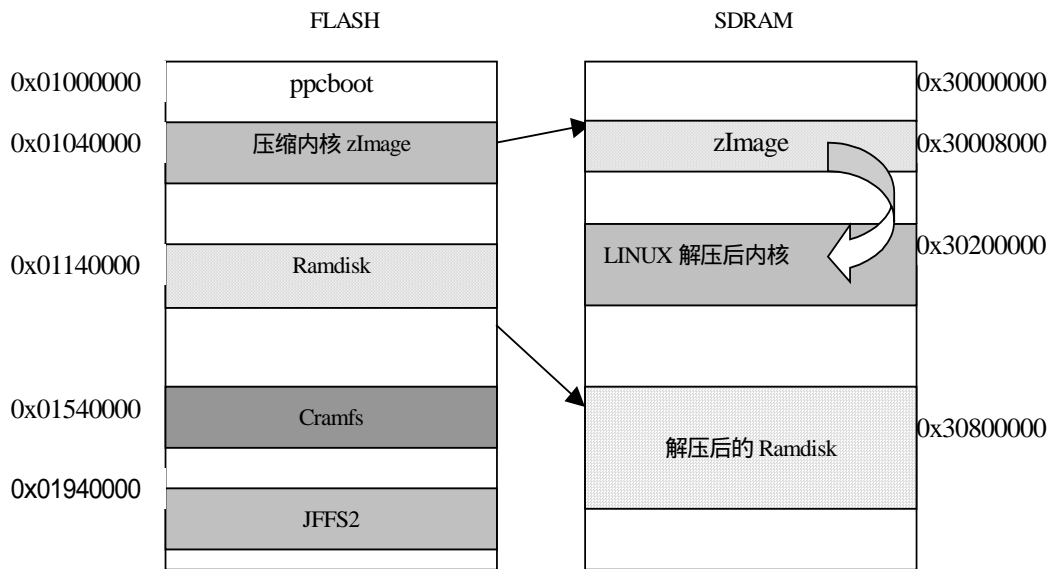
- 检测硬件裸板的 RAM;
- 检测硬件裸板的 FLASH;
- 检测硬件裸板的基本外设,如串口等只要设置几个寄存器就可以工作的简单外设;
- 下载代码到硬件板卡的 RAM 中,并使之运行;
- 烧写代码到硬件板卡的 FLASH 中,并使之运行;
- 调试 RAM 或

#### 4, Flash 存储内容

在未编译前, 原始的代码、库文件等占用的空间较大。视操作系统的不同及扩展代码量, 可以从几个 MB (uC/OS II) 到上百 MB (Embedded Linux), 甚至上 GB (WinCE) 的空间。嵌入式系统内的 Flash 芯片一般仅存储编译好的二进制 (binary) 可执行文件, 也有部分是经过压缩后的映像文件 (image)。内容主要包括: 启动代码 (Boot loader) / 板级支持包 (BSP) 、系统内核 (Kernel) 、文件系统 (File System) 等。

典型内存布局

```
/* Top Of Memory */  
Bootloader  
Parameter Area  
Kernel  
Filesystem  
/* End Of Memory */
```



HHARM2410 Flash 中存储空间地址分配

图五 典型

#### Flash 内存布局

Boot loader : 由于 BDM/JTAG 接口的编程、烧写速度比较慢, 因此一般的做法是先烧写一小段程序 (几十 K), 这部分程序可以作为一个小系统实现对系统硬件的初始化, 并使一些接口驱动起来以进行通讯和传输。一般可以通过串口或以以太网口下载、烧写更大的操作系统内核及文件系统。而当产品实际应用时, Boot loader 则起到了将相关 Kernel、Fire System 等复制、解压到 SDRAM 中并引导启动运行的作用。

BSP : Board Support Package, 主要包括了一些对系统硬件的驱动等底层软件。

Kernel : 主要针对嵌入式 Linux 而言, 包括了一些操作系统的底层软件及驱动等。

File System : 管理操作系统所需的各类文件。存在于 Flash 中的文件系统主要有, FAT、Ramdisk、Romfs、Jffs、Yaffs、Cramfs、TFFS 等等。

虽然 Flash 支持程序直接运行, 但在实际的使用中, 一般都会通过 Bootloader 引导, 将 Flash 中的内容复制、解压到内存 (RAM) 中运行。嵌入式系统目前用的比较多的是 SDRAM, 由于 RAM 不作为本文介绍的重点, 在这里不作详细介绍。

## 总结

作为系统内置的存储芯片，Flash 主要作为“系统存储空间”用来存储关键的操作系统、软件、程序等；而对于“用户存储空间”，可以选择在系统存储以外的其它 Flash 扇区开辟，或额外增加独立的 Flash 芯片作为存储空间，当然按照成本、容量等方面的考虑 NOR 和 NAND flash 都是可以考虑的。

当然嵌入式系统内置的存储空间毕竟有限，并且出于稳定性、灵活性考虑，各种基于大容量、低成本的 NAND Flash 的存储卡，也将在嵌入式系统中也发挥着越来越重要的作用。