

Nand Flash 文件系统解决方案

一. Nand Flash 结构及特性

NAND Flash 的结构如图 1 所示：

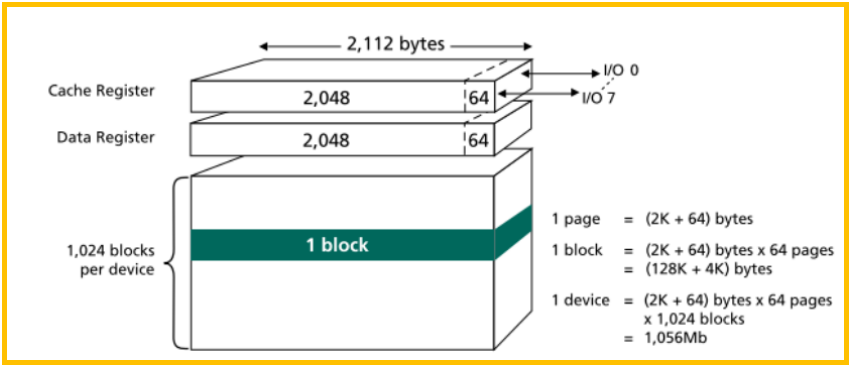


图 1 NAND Flash 结构

NAND Flash 由块（block）组成，每块又由若干页（page）组成，每页由数据区和冗余区（spare area）组成。页是数据写入的基础单元，块是擦除的基础单元。

在对 NAND Flash 进行写操作（页编程）时，只能把相应的位由“1”写为“0”，只有对块进行擦除操作时，才能把该块内所有位由“0”写为“1”。因此，在写入数据时，如果该页内已存有数据，必须先擦除该块。

SLC（Single-Level Cell）小容量的 NAND Flash 页一般为（512+16）Bytes（数据区 512Bytes，冗余区 16Bytes），大容量的 NAND Flash 页为（2048+64）Bytes。

为降低 Flash 的制造成本，半导体制造商先后推出了 MLC（Multi-Level Cell）和 TLC（Trinary-Level Cell）技术，即在一个 Flash 存储单元内存储更多的位信息，其原理如图 2 所示。与 SLC 相比，MLC 和 TLC 的存储密度更高，成本也相对降低，但管理难度更大：其传输速度、耗电量、擦写次数等方面的性能都要比前者差。

因此在工业应用场合，存储空间要求不是特别大的情况下，SLC 仍然得到广泛的应用。

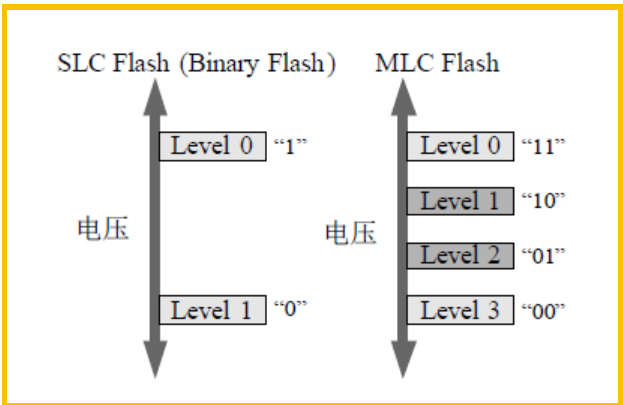


图 2 SLC 和 TCL 区别

使用 NAND Flash 需要解决以下问题：

1. 块擦除次数有限

NAND Flash 的块擦除次数是有限的。SLC 一般不超过 10 万次，而 MLC 擦除次数仅有 1 万次左右，具体数据以所使用的 NAND Flash 的产品手册为准。如果频繁擦除某块最终导致超过擦除次数限制，该块内的数据将变得不可靠，进而影响整个存储器的使用，所以需要采用损耗均衡算法使各个块近似均衡使用。

2. 异位更新（Non-In-Place Update）

由于 NAND Flash 先擦后写的物理特性，如果将文件存储在固定的块内，会面临掉电数据丢失及占用较大 RAM 等问题。所以，NAND Flash 一般采用异位更新，即将要更新的数据读入 RAM 中修改后写入其他空闲块，在适当的时候擦除修改前数据所在块。因此，需要地址映射、垃圾回收等技术来管理存储空间。

3. 坏块管理

坏块是指包含一位或多位错误的块，由于 NAND Flash 的制造工艺不能保证存储单元在其生命周期中保持可靠，因此，在 NAND 的生产及使用过程中会产生坏块。如果在坏块内编程或将坏块擦除，会发生错误，所以需要一定的机制管理坏块，使坏块不参与数据存储。

4. 差错校验码（Error Correction Code, ECC）

由于采用串联的架构，NAND 的晶体管之间容易造成影响，使逻辑“0”变成逻辑“1”，这种现象称为位翻转，需要使用差错校验码来进行纠错。常用的 ECC 有：Hamming 码、BCH 码、Reed-Solomon 码等。SLC NAND Flash 需要能检测 2 位，纠正 1 位错误的 ECC，而 MLC NAND Flash 通常需要检测并纠正 4 位的 ECC 来保证数据的安全可靠。

STM32 的 FSMC 硬件内置 ECC 功能。

二. ST 解决方案

ST 提供适用于 SLC 的 NFTL（NAND Flash Translation Layer）和 FAT 类文件系统来解决 NAND Flash 存储的问题。

解决方案如图 3 所示

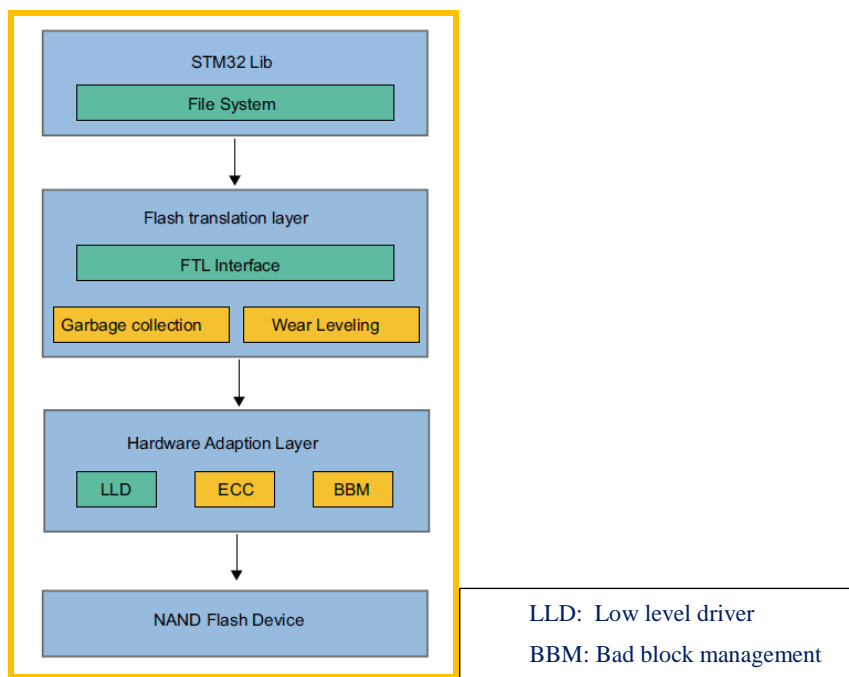


图 3 NAND Flash 文件系统解决方案

1. NFTL:

FTL 层的主要功能是针对 NAND Flash 的物理特性，封装底层硬件相关的操作和复杂的管理控制功能，向上提供设备读写接口，使得文件系统访问 NAND Flash 类似于访问磁盘；向下提供底层硬件接口信息，以方便用户更换 Flash。FTL 的主要功能包括地址映射、损耗均衡控制、垃圾回收、掉电保护、坏块管理及 ECC 校验等。

2. 文件系统

嵌入式应用中一般采用 FAT 文件系统，方便该系统能和别的系统之间直接进行文件互访，共享数据，存储格式的通用性是这一层的重点。它通过 FTL 提供的设备读写接口，封装文件管理操作，向上层提供文件操作接口。

由于嵌入式环境及 Flash 的物理特性，还会对 Flash 文件系统带来如下要求：

掉电安全:

嵌入式系统的运行环境比较恶劣，要求较高的可靠性。这要求无论程序崩溃或系统掉电，都不能影响数据的一致性和完整性。因为 Flash 存储器以块为擦除单元，块中原有数据的保存就比较麻烦，因为数据写入、垃圾回收等操作对系统异常终止都非常敏感，极易造成数据丢失和数据垃圾。

资源消耗低:

嵌入式系统中的存储空间有限，因此对资源消耗有较高的要求。

图 4 和图 5 是 ST 解决方案中文件系统和 NFTL 层资源消耗信息（IAR 编译环境），由此可见，ST 的解决方案在资源消耗方面是非常出色的。

		FatSL	FatFs
Footprint	RAM	<1K bytes	4.2~6.5K bytes
	Flash	about 4K bytes	2.2~2.8K bytes
Fail-Safe Design		Y	N

图 4 文件系统资源消耗

Code size	Flash	RAM
With Optimization (High size)	6.5KB	5KB
Without Optimization	11.7KB	5KB

图 5 NTFL 资源消耗

关于 NTFL 层的软件和文档，如有需要，请与 ST 的微控制器市场部联系。