

君正 Nand Flash 烧录器手册

Revision: 1.0
Date: June 2008



北京君正集成电路有限公司
Ingenic Semiconductor Co. Ltd

君正 Nand Flash 烧录器手册

Copyright © Ingenic Semiconductor Co. Ltd 2006. All rights reserved.

Release history

Date	Revision	Change
June 2008	1.0	First release

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co. Ltd

**E801C, Power Creative Building,
No.1, Shangdi East Road,
Haidian District, Beijing 100085, China
Tel: 86-10-58851002
Fax: 86-10-58851005
Http: //www.ingenic.cn**

内容

1 概述	1
2 烧录器硬件说明.....	3
3 烧录器软件使用说明	5
3.1 功能说明.....	5
3.2 使用准备.....	5
3.3 命令行用法	6
3.4 配置文件详解	7
3.5 将配置编译在内部	8
3.6 错误处理.....	9
3.7 边缘情况的考虑.....	9

1 概述

北京君正提供 Nand Flash 烧录器的参考设计，主要用于产品阶段的 Nand Flash 烧录。该工具包括烧录器硬件和烧录器程序两个部分，两者配合，构成完整的 Nand Flash 烧录平台。

烧录器硬件具有以下特点：

- 提供 Jz4730、Jz4740 两个硬件平台；
- 一个硬件平台带多个 Nand Flash 插座；
- 能够支持 MLC 和 SLC Nand Flash；

烧录器软件具有以下特点：

- 独立的应用程序，运行在 Linux 平台，并且不依赖于 MTD 驱动；
- 同时支持目前和未来所有的君正系列处理器；
- 以母片方式实现烧录，烧录对象与操作系统无关，与文件系统无关；
- 最大程度保证烧录的稳定性、可靠性，并提供完备的日志文件记录；
- 提供灵活的、可配置的使用接口；
- 按重写-放弃策略管理坏块，并支持多片选；
- 软件提供开放接口，方便用户修改、定制。

2 烧录器硬件说明

(请参考烧录器硬件参考设计)

3 烧录器软件使用说明

3.1 功能说明

烧录程序实际上只提供两种功能：

1. 制作母片文件：把 Nand Flash 中指定范围的数据读出到的指定文件中，读过程中提供 ECC 保护。
2. 烧录母片文件：把指定文件写入到 Nand Flash 的指定范围并进行校验，写过程中也提供 ECC 保护，对校验失败的块采取重试-放弃策略，重试三次仍失败则把该块标记为坏块。

依靠这两种功能，就可以实现与操作系统无关，与文件系统无关的 Nand Flash 烧录。

一般情况下，一个型号产品完整的烧录步骤包括以下：

1. 制作母片：将最终程序、数据等按照系统定义的范围写入一个原始的 Nand Flash 中；
2. 制作母片文件：把母片 Nand Flash 中的数据原封不动读到一个或多个指定文件中；
3. 烧录母片文件：将制作好的母片文件按照原来的范围烧录到新的 Nand Flash 中。

烧录程序可以确保读写过程的正确性，也就是说，如果烧录过程中没有报错、没有坏块，母片的数据将原封不动复制到新的 Nand Flash 中。

只要产品的系统没有变化，母片文件只需要制作一次，利用这些母片文件，同类型产品可以稳定、可靠地迅速繁殖。

3.2 使用准备

烧录软件在第一次使用之前包括可执行程序 `nandprog` 和配置文件 `nandprog.cfg`，第一次使用之后将在同目录下增加两个文件：`number.log` 和 `nandprog.log`。其中前者为临时记数文件，后者为烧录过程的日志记录，批量烧录之后可查询 `nandprog.log` 了解每一次的烧录细节。

烧录软件在真正执行烧录之前需要得到足够的配置信息，这些信息包括：

1. 处理器型号：目前只有 Jz4730 或者 Jz4740。
2. Nand Flash 接口参数：数据线宽度、地址周期、页大小等。
3. 坏块标志位置，位于块内第几个页，位于页内第几个字节。
4. ECC 的类型和存储位置。

有两种方法将上述信息传递给执行程序：

1. 在源码中添加固定的配置信息，编译在可执行程序内部，使用时通过命令行参数选定指定的配置。

这种方法适用于烧录定型产品。

2. 通过配置文件，程序执行时从配置文件读取配置信息，然后执行。这种方法具有更大的通用性。

3.3 命令行用法

命令行格式：nandprog spn epn ot ofn cs [index]

spn: 操作起始页数，相对于 Nand Flash 的第一个页

epn: 操作终止页数，相对于 Nand Flash 的第一个页

ot : 操作类型，读或 (r) 者写 (w)

ofn : 目标或者源文件名

cs : 片选索引，指定烧录平台的某个芯片

index: 该项可选，如果命令行中没有指定该值，则程序从配置文件中读取相关配置信息；如果指定了该值，则程序使用编译在内部的对应配置信息。

使用实例：

1. 对于 linux 系统，将 MTD2 分区的数据读出来制作母片文件 parent.mtd2，通过配置文件传递配置信息：

```
# ./nandprog 4096 65535 r parent.mtd2 0
Deal command line: spage4096 epage65535 ops0 file:parent.mtd2 cs0
Load configuration file success!
Map EMC_BASE success :2aac5000
Map NAND_PORT success :2aac6000
Map GPIO_PORT success :2aae6000
Memory map all success!
Read block 32 finish
Read block 33 finish
.....
```

上述参数中 4096 和 65535 分别是 MTD2 分区的第一个页号和最后一个页号，r 表示读操作，0 表示对平台上第 0 个 Nand Flash。

2. 对于 linux 系统，母片文件 parent.mtd2 烧录到 MTD2 分区中，通过配置索引传递配置信息：

```
# ./nandprog 4096 65535 w parent.mtd2 0 5
Deal command line: spage4096 epage65535 ops1 file:parent.mtd2 cs0
Load configuration index success!
Map EMC_BASE success :2aac4000
Map NAND_PORT success :2aac5000
Map GPIO_PORT success :2aae5000
Memory map all success!
```

```
Write block 32 finish
Write block 33 finish
.....
```

上述参数中 **w** 表示写操作，最后的参数 **5** 表示使用编译在程序内部的第 **5** 个配置。

3.4 配置文件详解

可以通过配置文件向执行程序传递配置信息，配置文件的典型格式如下：

```
;烧录器配置文件样本
[NANDPROG]                ;配置信息开始标志，与后呼应。
CPUYPE                    JZ4740 ;选择处理器型号，目前可以为 JZ4730/JZ4740

BUSWIDTH                  8      ; Nand Flash 数据位宽
ROWCYCLES                 3      ; Nand Flash 操作周期
PAGESIZE                  2048   ; Nand Flash 页大小
PAGEPERBLOCK              128    ; Nand Flash 每块包含的页数
OOBSIZE                   64     ; Nand Flash 的 OOB 大小

BADBLOCKPOS               0      ;定义坏块标志位置，此值为页内相对于 OOB 的偏移
BADBLOCKPAGE              0      ;定义坏块标志位置，此值为块内第几个页

ECCTYPE                   RS     ;选择 ECC 类型，目前可以为 RS/HM
;下面的括号内定义 ECC 存储位置的偏移，个数由 ECC 类型确定，RS 为 36 个，HM 为 24 个
{
    28 29 30 31 32 33 34 35 36
    37 38 39 40 41 42 43 44 45
    46 47 48 49 50 51 52 53 54
    55 56 57 58 59 60 61 62 63
}
;位置之间用空格分隔，不要在'{}'之间添加其他字符
[END]                    ;配置信息结束标志，执行程序忽略开始、结束标志之外的所有信息
```

这个配置文件针对的系统是：

1. 基于 Jz4740 处理器；
2. 8 位宽 2K 页的 Nand Flash，每个块包括 128 个页；
3. 坏块标志位于每个块的第 0 页，页内 OOB 中的第 0 个字节；
4. 使用 RS 纠错码，纠错码信息存储在 OOB 中的偏移为 28 字节、29 字节……

实际上这个配置用于烧录 Linux 文件系统，基于 PAVO 开发板，使用 RS 纠错码。使用时只需要修改该文件中的对应数据即可，文件的格式尽量不要改动。

3.5 将配置编译在内部

将配置编译在执行程序的内部的方法比使用配置文件麻烦一些，需要修改源代码并重新编译。烧录软件的代码组织结构如下：

```
|-- Makefile
|-- common
|   |-- cmdline.c
|   |-- loadcfg.c
|   `-- main.c
|-- include
|   |-- configs.h
|   |-- include.h
|   |-- jz4730.h
|   |-- jz4740.h
|   `-- nand_ecc.h
|-- jz4730
|   `-- nandflash_4730.c
`-- jz4740
    `-- nandflash_4740.c
```

其中 `include` 目录中定义头文件，`common` 目录中存放命令行接口代码，`jz4730` 和 `jz4740` 中分别包含两个型号处理器的 Nand Flash 驱动。

用户添加自己的配置需要修改 `include/configs.h` 和 `include/nand_ecc.h` 头文件。首先修改 `include/nand_ecc.h` 头文件，在该文件中添加 ECC 存储位置信息，例如：

```
{
    .eccname = LINUXRS,
    .eccbytes = 36,
    .eccpos =
    {
        28, 29, 30, 31,
        32, 33, 34, 35, 36, 37, 38, 39,
        40, 41, 42, 43, 44, 45, 46, 47,
        48, 49, 50, 51, 52, 53, 54, 55,
        56, 57, 58, 59, 60, 61, 62, 63
    },
},
```

按照这个格式来定义即可，赋值的含义与配置文件一致。

然后修改 `include/configs.h` 头文件，文件中已经有许多配置，基本格式如下：

```

{
    //No 5: 该编号就是当前配置在整个配置数组中的下标, 命令行传递此值
    //The config for jz4740 linux use HW RS
    .pt = JZ4740,           //选择处理器型号
    .et = HARDRS,          //选择 ECC 类型, 为 HARDRS 或者 HARDHM
    .ep = 3,                //指定使用 ECC 存储位置的数组下标, 在 nand_ecc.h 中
    .bw = 8,                //Nand Flash 数据位宽
    .ps = 2048,            //Nand Flash 页大小
    .os = 64,              //Nand Flash 的 OOB 大小
    .ppb = 128,            //Nand Flash 每块包含的页数
    .rc = 3,                //Nand Flash 操作周期
    .bbp = 1,              //定义坏块标志位置, 此值为页内相对于 OOB 的偏移
    .bba = 0,              //定义坏块标志位置, 此值为块内第几个页

    //以下不需要修改。
    .ebase = 0x13010000,
    .dport = 0x18000000,
    .gport = 0x10010000,
    .....
},
    
```

修改妥当之后, 重新编译, 使用时在命令行传递索引值, 就可以使用新配置。这种方式不通过配置文件, 可以提高烧录速度。

3.6 错误处理

烧录过程中会出现一些错误, 其中有一些是无关紧要的, 另一些则是反映烧录的具体情况。这些错误中前者只会在命令行输出提示, 后者则会准确地记录在日志文件中。错误大全如下:

1. 命令行输入错误, 日志文件不记录;
2. 烧录准备、初始化不成功, 不记录;
3. 跳过了旧的坏块, 记录;
4. 数据校验失败, 重试, 记录;
5. 重试失败, 放弃并标记新的坏块, 记录;
6. 数据没有完全烧录到 Nand Flash 中, 即空间不足了, 记录;
7. 烧录失败, 即烧录后 Nand Flash 返回失败状态, 记录;
8. 最后记录每次烧录的最终状态, 是否成功, 当没有发生 6 的情况, 都认为烧录成功。

3.7 边缘情况的考虑

烧录过程中会发生一些很低概率, 有危险, 又难以完全避免的情况, 烧录程序最大程度避免这些问题:

1. 坏块太多：如果目标 **Nand Flash** 中坏块太多，那么母片文件的数据有可能没有完全写入，这样就导致烧录失败，这种情况无法避免。
2. 制作母片文件有错：有两种原因可能造成这种情况，一为 **Nand Flash** 中某页发生了超过 **ECC** 纠错能力的错误，无法纠正；二为错误的位置位于没有 **ECC** 保护的区域。第二种情况概率比第一种大，实际上 **Nand Flash** 中每一页都有若干字节没有系统 **ECC** 的保护，但是可能有文件系统 **ECC** 的保护。烧录程序无法处理这样的情况，但是有方法避免：
 - a. 母片 **Nand Flash** 使用 **SLC**;
 - b. 系统使用带有局部 **ECC** 保护的文件系统;
 - c. 用多个母片制作多个母片文件，进行对比，比较无误才使用。
3. 出现新坏块，并且这个坏块连坏块标记都打不上，那就放弃这个 **Nand Flash** 芯片吧。