SigmaStar Display

Spinand 母片制作

REVISION HISTORY

Revision No.	Description	Date
0.1	• Initial release	07/26/2019
0.2	• Beta release	01/14/2020
	•	

TABLE OF CONTENTS

1. SPINAND 母片制作方法

先到如下链接下载母片制作工具 SpinandPackBinTool.rar：

1.1. 基本文件主要有下面五个:

1.可执行程序 SstarMakeBin

2.初始化文件 SPINAND.INI(spinand)

SPINAND.INI 中的 nandid 需要根据实际 flash 的 datasheet 来填写, image_directory 为对应需要打包的 image 路径：

```
[path]
root_directory=./
image_directory=images/
script_file=auto_update_spinand.txt
outpath=./

[nand]
nandtype=SPINAND
nandid=0xC2-0x12

[env]
env_type=NANDRAW
env_part=ENV
#env_volume=MPOOL
#env_offset=0

[cis]
ecctype=10
nni=SPINANDINFO.sni
pni=PARTINFO.pni
#ppm=PAIRPAGEMAP_v2.ppm

[fcie]
type=fcie5

[boot]
#boottype=BFN
#miu=bfn_miu.bin
```

3.烧录器的配置文件 leap.def & snchk.def(leap 烧录器用 leap.def,其他烧录器一般用 snchk.def)

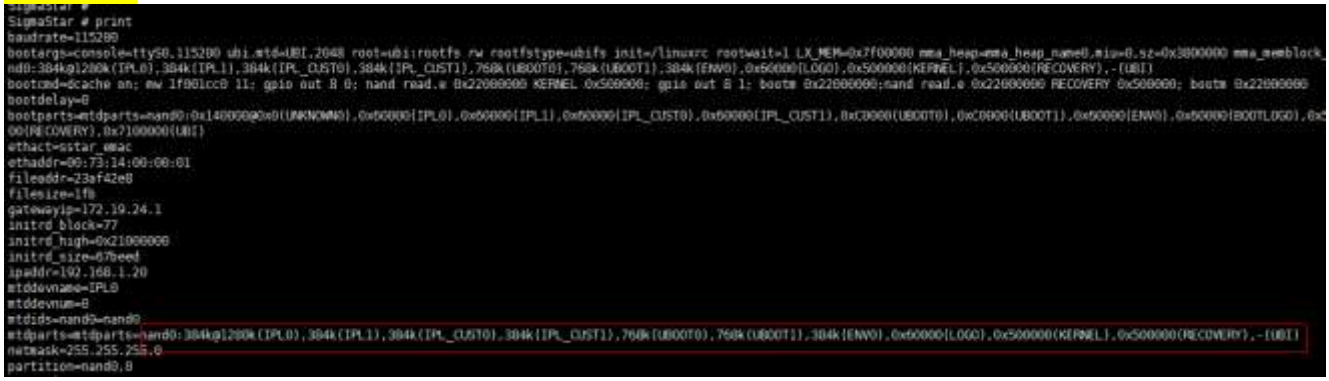
4.用来制作母片的 image 文件以及对应的烧录脚本，统一放在 images 文件夹下：

images	2020/3/18 11:05	文件夹	
auto_update_spinand.txt	2020/3/4 14:36	文本文档	1 KB
readme.txt	2020/3/4 15:20	文本文档	4 KB
SPINAND.INI	2020/3/18 11:08	配置设置	1 KB
SstarMakeBin	2020/3/4 12:43	文件	2,049 KB
leap.def	2020/3/17 20:53	DEF 文件	1 KB
snchk.def	2020/3/17 20:53	DEF 文件	1 KB

images 文件夹是从 project\image\output\images 下面的 images 拷贝过来的

5.配置分区和 image 路径的脚本 auto_update_spinand.txt

需要注意的是 auto_update_spinand.txt 下面 dynpart 这行的分区信息需要跟 uboot 下面的 mtdparts 保持一致：



6. 需要注意的是，ubi 文件格式需要额外的空间来存一下 ubi 的信息，所以 ubi 分区不能全部用完，需要预留如下大小的空间用来存放 ubi 信息：

常见的 1G bit，2G bit，4G bit NAND 的预留量算出来为 9MB，17MB，34MB

不然在制作母片的时候会报如下 peb 不够的 error：

```
last_partition : nand0,0
env_partition  : nand0,0
mtdparts_init 1764 enter
Creating dynamic volume appconfigs of size 5242880
[ubi_create_volume]: search for vacant volume ID
[ubi_create_volume]: not enough PEBs, total=903, avail=40, vol reserved=42
UBI error: ubi_create_volume: cannot create volume 3, error -28
[do_mstar]: command error!
[do_mstar]: command error!
[do_mstar]: command error!
[nandbin:1319]update fail, please check log
[main]: Make spinand bin failed
```

Eg：公版 spinand 128M 的 flash ,ubi 分区总的大小 903 个 block ,需要预留的大小为>= 903*2*4k + 4*4k

```
Checksum ok!!
IDX:          StartBlk:           BlkCnt:     BackupBlkCnt:        PartType:
  0:     0,(0000000000)   10,(0X00140000)    0,(0000000000)  0X0020,(CIS)
  1:    10,(0X00140000)    1,(0X00020000)    2,(0X00040000)  0X0003,(IPL0)
  2:    13,(0X001A0000)    1,(0X00020000)    2,(0X00040000)  0X0003,(IPL1)
  3:    16,(0X00200000)    1,(0X00020000)    2,(0X00040000)  0X0001,(IPL_CUST0)
  4:    19,(0X00260000)    1,(0X00020000)    2,(0X00040000)  0X0001,(IPL_CUST1)
  5:    22,(0X002C0000)    2,(0X00040000)    4,(0X00080000)  0X0006,(UBOOT0)
  6:    28,(0X00380000)    2,(0X00040000)    4,(0X00080000)  0X0006,(UBOOT1)
  7:    34,(0X00440000)    3,(0X00060000)    0,(0000000000)  0X000D,(ENV)
  8:    37,(0X004A0000)    1,(0X00020000)    0,(0000000000)  0XF000,(KEY_CUST)
  9:    38,(0X004C0000)    3,(0X00060000)    0,(0000000000)  0X0002,(LOGO)
 10:    41,(0X00520000)   40,(0X00500000)    0,(0000000000)  0X0012,(KERNEL)
 11:    81,(0X00A20000)   40,(0X00500000)    0,(0000000000)  0X0009,(RECOVERY)
 12:   121,(0X00F20000)  903,(0X070E0000)    0,(0000000000)  0X0021,(UBI)
```

所以在 project\image\configs\i2m\spinand.ubifs.p2.partition.config 中，ubi 分区总的大小不能超过 0X070E0000 - 903*2*4k + 4*4k

```
66  rootfs$(RESOUCE)    = $(OUTPUTDIR)/rootfs
67  rootfs$(FSTYPE)     = ubifs
68  rootfs$(PATSIZE)    = 0xA00000
69  rootfs$(BOOTENV)    = console=ttyS0,115200 ubi.mtd=UBI,2048 root=ubi:rootfs ro rootfstype=ubifs init=/linuxrc rootwait=1
70
71  miservice$(RESOUCE)     = $(OUTPUTDIR)/miservice/config
72  miservice$(FSTYPE)      = ubifs
73  miservice$(PATSIZE)     = 0xA00000
74  miservice$(MOUNTTG)     = /config
75  miservice$(MOUNTPT)     = ubi0:miservice
76  miservice$(OPTIONS)     = rw
77  miservice$(OTABLK)      = /dev/ubi0_1
78
79  customer$(RESOUCE)      = $(OUTPUTDIR)/customer
80  customer$(FSTYPE)       = ubifs
81  customer$(PATSIZE)      = 0x5000000
82  customer$(MOUNTTG)      = /customer
83  customer$(MOUNTPT)      = ubi0:customer
84  customer$(OPTIONS)      = rw
85  customer$(OTABLK)       = /dev/ubi0_2
86
87  appconfigs$(RESOUCE)    = $(OUTPUTDIR)/appconfigs
88  appconfigs$(FSTYPE)     = ubifs
89  appconfigs$(PATSIZE)    = 0x500000
90  appconfigs$(MOUNTTG)    = /appconfigs
91  appconfigs$(MOUNTPT)    = ubi0:appconfigs
92  appconfigs$(OPTIONS)    = rw
93  appconfigs$(OTABLK)     = /dev/ubi0_3
```

7. 默认是从 images 目录下面去读取 SPINANDINFO.sni & PARTINFO.pni 文件，所以在制作母片前需要把 images\boot 下面的 SPINANDINFO.sni & PARTINFO.pni 拷贝到 images 下面，不然会报如下错误：

```
nand_init_chip:41 enter
[_MDrv_SPINAND_GET_INFO]: Cannot open ./images/SPINANDINFO.sni
##############################################
#        SPINAND INFO                        #
##############################################
[MDrv_SPINAND_Init]: page size: 2048
[MDrv_SPINAND_Init]: spare size: 64
[MDrv_SPINAND_Init]: page count: 64
[MDrv_SPINAND_Init]: block count: 1024
[board_spinand_init]: Cannot open pni file ./images/PARTINFO.pni
##############################################
```

1.2. 执行 SstarMakeBin 生成 nand.bin

需要注意的是在执行打包命令前，需要将 SpinandPackBinTool\images\scripts\[[set_partition.es 下面这行屏蔽掉：

```
# <- this is for comment / total file size must be less than 4KB
mtdparts del CIS
#setenv mtdparts $(mtdparts),0x60000(LOGO),0x500000(KERNEL),0x500000(RECOVERY),-(UBI)
saveenv
nand erase.part UBI
ubi part UBI
ubi create rootfs 0xA00000
 ubi create miservice 0xA00000
 ubi create customer 0x5100000
 ubi create appconfigs 0x300000

% <- this is end of file symbol

```

然后在 linux 执行

./SstarMakeBin -n SPINAND.INI

这样会生成一个 nand.bin

```
aaron.feng@sigmastar:~/Code/P2/Beta_Release/mstarbin/release/SpinandPackBinTool$ ./SstarMakeBin -n SPINAND.INI
[parse_ini_file:1161]ini file=SPINAND.INI
[iniparser_line][467]: [path]
[iniparser_line][467]: root_directory=./
[iniparser_line][467]: image_directory=images/
[iniparser_line][467]: script_file=auto_update_spinand.txt
[iniparser_line][467]: outpath=./
[iniparser_line][467]:
[iniparser_line][467]: [nand]
[iniparser_line][467]: nandtype=SPINAND
[iniparser_line][467]: nandid=0xC2-0x12
[iniparser_line][467]:
[iniparser_line][467]: [env]
[iniparser_line][467]: env_type=NANDRAW
[iniparser_line][467]: env_part=ENV
[iniparser_line][467]: #env_volume=MPOOL
[iniparser_line][467]: #env_offset=0
[iniparser_line][467]:
Nand Flash basic information:
    main=2048 bytes(Hex:0x800),  spare=64 bytes(Hex:0x40)
    main+spare=2112 bytes(Hex:0x840),  page number per block=64
    block size=0x20000(128KB, main),  block size=0x21000(main+spare)

[nand_truncate_outfile]: SPINAND type
[nand_truncate_outfile]: truncate file with length 0x1b75000
[nandbin:1355]Nand bin file done!!!

aaron.feng@sigmastar:~/Code/P2/Beta_Release/mstarbin/release/SpinandPackBinTool$
aaron.feng@sigmastar:~/Code/P2/Beta_Release/mstarbin/release/SpinandPackBinTool$ ls
auto_update_spinand.txt  images  leap.def  nand.bin  readme.txt  snchk.def  SPINAND.INI  SstarMakeBin
aaron.feng@sigmastar:~/Code/P2/Beta_Release/mstarbin/release/SpinandPackBinTool$
```

这个 bin 就是烧录程序，然后使用烧录器烧录即可。

1.3. 制作母片出错检查

1. 如果是执行生成 bin 时有报错，先检查一下是否所执行的命令不支持，比如 sd 的升级脚本需要改为 tftp 的脚本形式
2. 是否执行了一些 save 命令，重新保存了分区信息，导致分区信息与在 auto_update_spinand.txt 中指定的分区不一致
3. 如果是烧录到板子上跑不起来，首先根据分区信息，检查生成的 bin 文件对应偏移地址处的文件信息是否正确，比如 IPL 的分区偏移为 0x140000，则检查生成的 bin 文件地址 0x140000 位置处的文件内容是否为 IPL 文件的内容
4. 检查烧录器烧录时 oob 区是否 disable
5. 用 flash tool 工具分别 dump 出板子上 flash 开始处、IPL 分区偏移处、IPL_CUST 分区偏移处、UBOOT 分区偏移处、kernel 分区偏移处的部分文件对比是否与烧录的 bin 文件一致

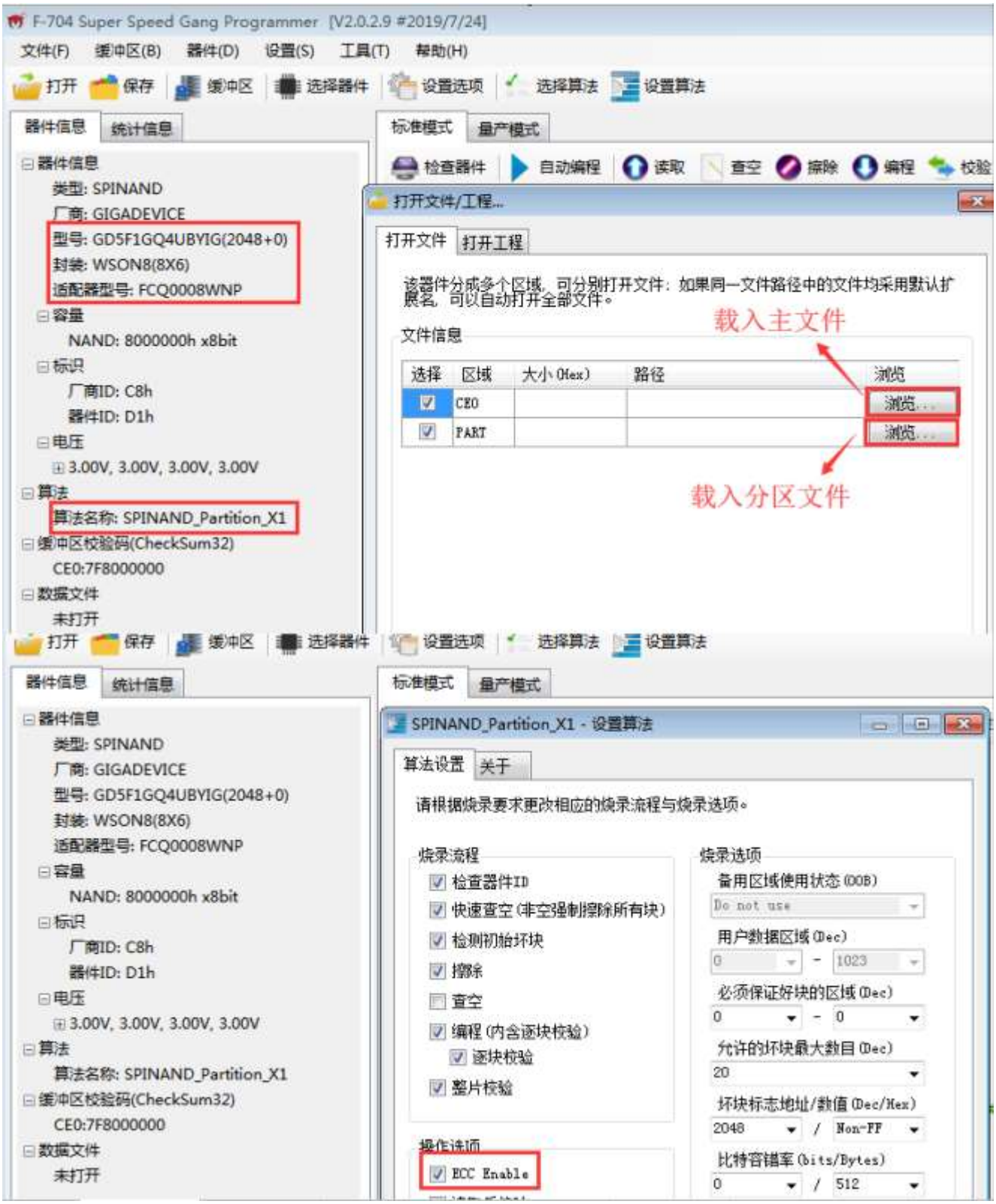
2. 已经支持的 FLASH 列表

目前已经支持的 flash 列表参考如下链接：flash_list.xlsx

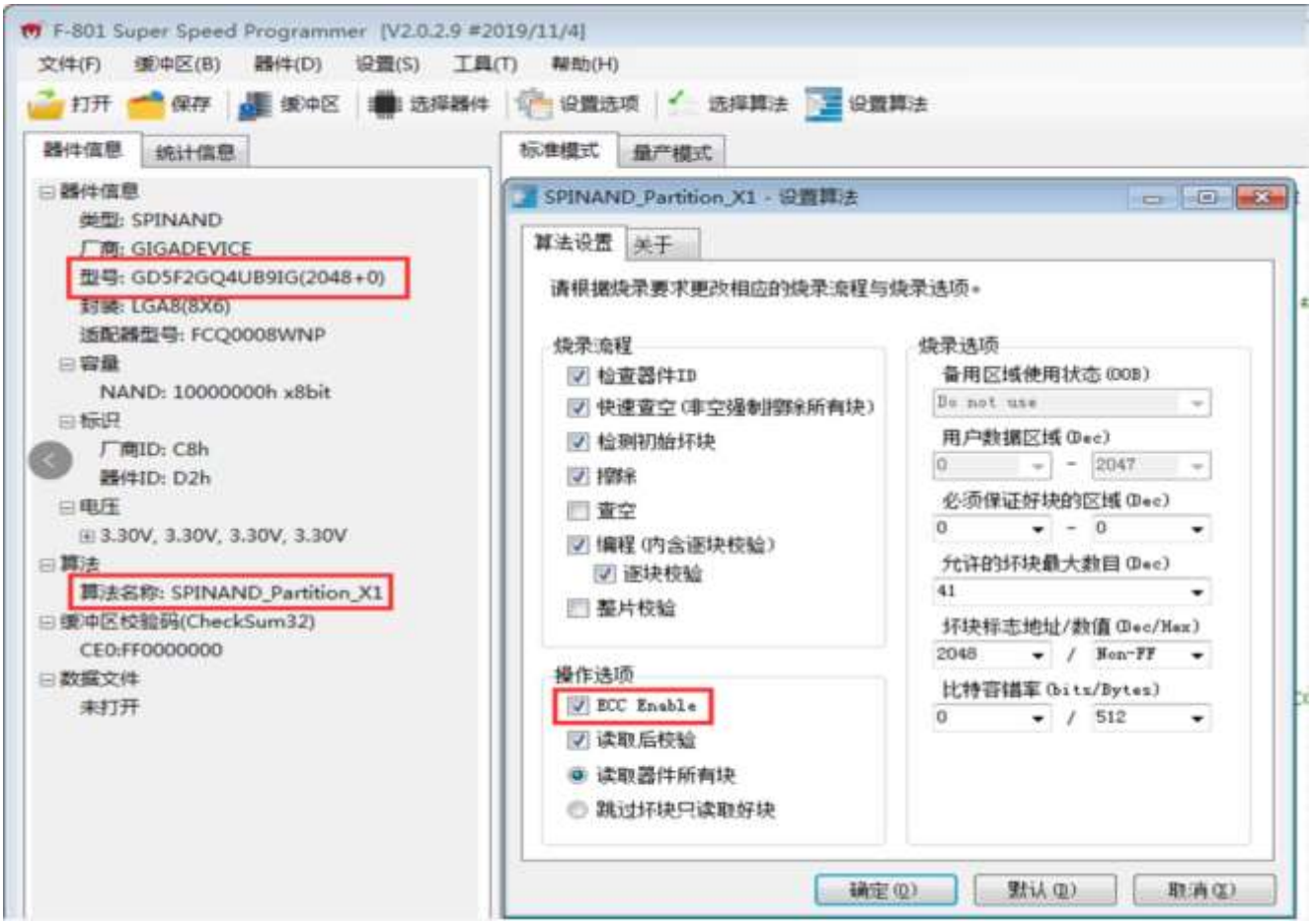
3. 烧录器参考配置

烧录器参考如下配置(不同烧录器配置可能不同)：

针对 GD5F1GQ4UBYIG 烧录器配置如下：



针对 GD5F2GQ4UB9IG 烧录器配置如下：



4. 母片制作配置

应用产品信息

CPU 品牌/型号	Sigmastar
软件系统	Linux

烧录数据文件(镜像文件)

NO	文件名	文件大小(bytes)	对应芯片地址/或块编号
1			
2			
3			
…			

镜像文件中是否包含 Spare(OOB)数据:

☐ 有 Spare(OOB)数据: 字节

✓ 无 Spare(OOB)数据

坏块管理模式(是否有分区)

☐ 跳过方式: Skip IB

✓ 分区方式: Multiple Partitions (with skip IB)

*** 需要提供分区表文件

是否使用 On-chip ECC(仅带有片内 ECC 功能的 Flash)

☐ 不使用 On-chip ECC

最大允许错误位参数: (bits) / (bytes)

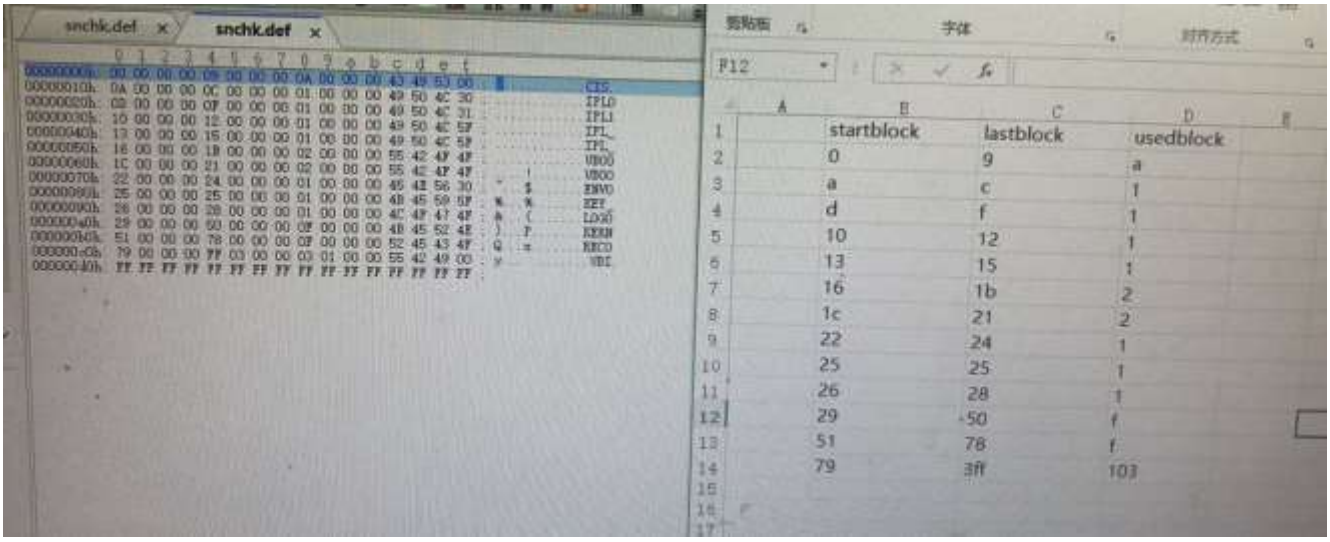
*** 镜像文件中必须包含 Spare(OOB)数据

✓ 使用 On-chip ECC

坏块标记

标记位置(地址): 800h 坏块标记数据: 00h

分区信息配置参考制作母片动态生成的 snchk.def 文件，使用二进制打开这个文件格式如下：



每一行是一个分区，每四个字节是一个数据，1~4 个字节表示的是 startblock，5~8 表示 lastblock,9~12 表示 usedblock，需要注意的是 def 格式是大端存储的