

SpiNor 母片制作

a. 因目前ISP_TOOL不支持SPI NOR 32M的flash寻址（目前最大支持16M寻址），所以之前提供 <SPI NOR母片制作文档.pdf> 无法使用SPI_TOOL去dump母片，这里使用脚本的方式制作

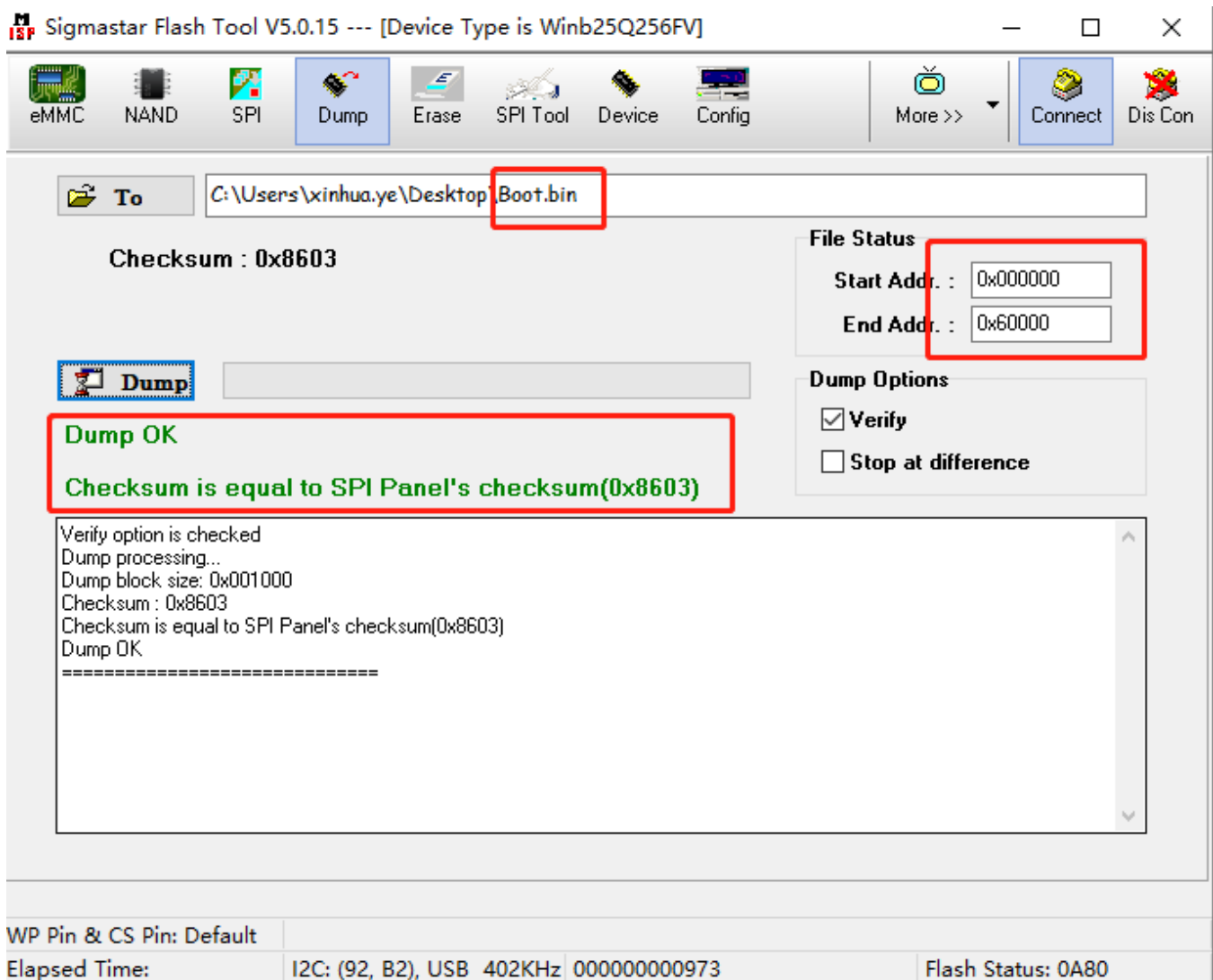
b. 因env数据是需要uboot运行之后才写到对应分区的，所以BOOT分区部分还是需要依赖ISP_TOOL去dump，具体操作如下：

1. 编译对应nor版本image，通过tftp烧录到板子上

2. 板子启动到boot控制台，通过输入`mxp t.list` 确认boot分区的大小（boot分区包含IPL.bin IPL_CUST.bin MXP_SF.bin u-boot.xz.img.bin 外加env的4K分区）

```
[mxp_record]: 5
  name: BOOT
  type: 0x03
  format: 0x00
  backup:
  start: 0x00000000
  size: 0x00060000
  status: 0x00
```

3. 查询好了之后使用ISP_TOOL dump下boot分区的数据，这里建议名字Boot.bin（否则需要改make_nor_bin.sh中的ORIGIN_FILE）



4. 配置make_nor_bin.sh 脚本：需要关注的有

ORIGIN_FILE=\$IMAGE_DIR/Boot.bin //这个文件名字就是步骤3种用isp_tool dump出来的Boot.bin

TARGET_FILE=\$IMAGE_DIR/SpiFlash.bin //这个是最终制作生成的母片烧录bin 这里的文件大小跟实际的flash大小一致

按照自己的分区顺序, 使用func_merge_bin_file 函数将对应的分区的bin 跟文件size 填写好, 如下图 (按照自己实际的分区顺序跟分区size填好即可)

```
70
71 func_generate_dummy_file
72 func_merge_bin_file boot.bin 0x60000;
73 func_merge_bin_file kernel 0x200000;
74 #KEY_CUST:0x10000
75 #func pad dummy size $TARGET_FILE 0x10000;
76 func_merge_bin_file logo 0x20000;
77 func_merge_bin_file rootfs.sqfs 0x1C0000;
78 func_merge_bin_file miservice.sqfs 0x370000;
79 func_merge_bin_file customer.sqfs 0x770000;
80 func_merge_bin_file appconfigs.jffs2 0xD0000;
81
82 rm $PAD_DUMMY_BIN
```

***注意这里, 如果使用zk_gui的加密lib, 默认会有一个KEY_CUST 分区, 该分区也需要追加空数据, 这里将注释掉的打开即可 (目前size是0x10000)**

5. 将生成的image目录下的对应分区文件跟脚本文件放到相同目录, 执行脚本即会生成TARGET_FILE=\$IMAGE_DIR/SpiFlash.bin, 该文件可以直接做母片烧录文件

s (\\szbc6402) (Y:) > SpinorPackBinTool

名称	修改日期	类型	大小
appconfigs.jffs2	2020/12/11 15:52	JFFS2 文件	832 KB
Boot.bin	2020/12/11 17:28	VLC media file (....	384 KB
customer.sqfs	2020/12/11 15:52	SQFS 文件	108 KB
kernel	2020/12/11 15:52	文件	1,945 KB
logo	2020/12/11 15:52	文件	62 KB
make_nor_bin.sh	2020/12/11 18:19	SH 文件	3 KB
make_nor_bin.sh.bak	2020/12/11 18:05	BAK 文件	3 KB
make_uboot_bin_621_nor.sh.bak	2020/12/11 17:47	BAK 文件	3 KB
miservice.sqfs	2020/12/11 15:52	SQFS 文件	3,312 KB
rootfs.sqfs	2020/12/11 15:52	SQFS 文件	1,704 KB
SpiFlash.bin	2020/12/11 17:49	VLC media file (....	16,384 KB

#图四示例对应的分区表

```
SigmaStar # mpx t.list
Total MXP record count:13
[mpx_record]: 0
name: IPL
size: 0x00010000

[mpx_record]: 1
name: IPL_CUST
size: 0x00010000

[mpx_record]: 2
name: MXPT
size: 0x00010000

[mpx_record]: 3
name: UBOOT
size: 0x0002F000

[mpx_record]: 4
name: UBOOT_ENV
size: 0x00001000
```

```
[mxp_record]: 5  
name: BOOT  
size: 0x00060000
```

//虚线上方的分区地址其实都包含再BOOT分区里，也就是ISP dump数据的时候，只要dump这个BOOT分区的size就好，省去计算

```
[mxp_record]: 6  
name: KERNEL  
size: 0x00200000
```

```
[mxp_record]: 7  
name: KEY_CUST  
size: 0x00010000
```

//该分区，生成的image种没有对应的bin，所以如果有该分区，需要关注第4步骤的 * 注意部分

```
[mxp_record]: 8  
name: LOGO  
size: 0x00020000
```

```
[mxp_record]: 9  
name: rootfs  
size: 0x001C0000
```

```
[mxp_record]: 10  
name: miservice  
size: 0x00370000
```

```
[mxp_record]: 11  
name: customer  
size: 0x00770000
```

```
[mxp_record]: 12  
name: appconfigs  
size: 0x000D0000
```