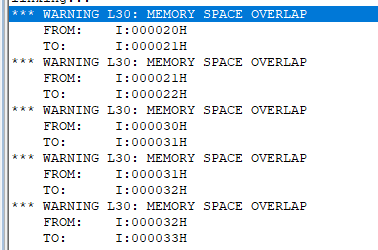
# 1 编译报 WARNING L30: MEMORY SPACE OVERLAP - (FROM)

这是因为ram和maskrom的内存空间重合了，这个警告是正常的，不会造成什么影响。



# 2 多个fw文件如何准确烧录

mptool会读flash的信息，判断其模型为TLC/SLC，然后根据模型烧入相应的fw，

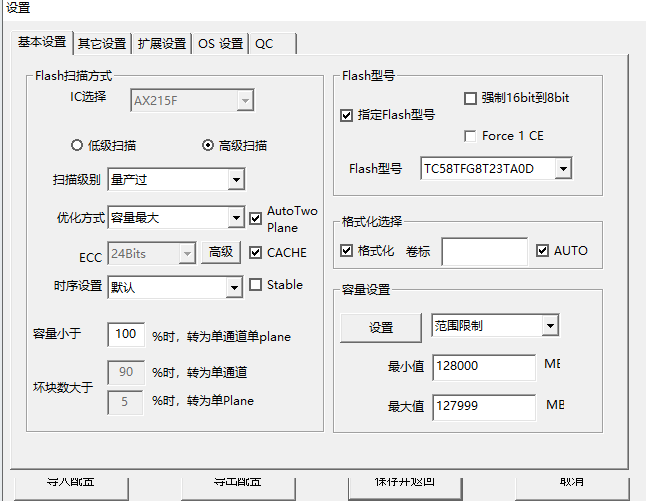
如何判断模型？

plane的状态之类，目前还不清楚

Greater fw 如何判断，如何烧录？

## 2.1 greater fw烧入

在mptool选择范围限制



# 3 文件系统丢失问题

需要查看一下文件系统的数据是否丢失，可能是由于上电恢复数据时，文件系统数据恢复错误。

使用winhex等软件查看磁盘数据，看看文件系统是否算坏

命令行使用Chck disk，磁盘状态，出现集群错误，说明是FAT表出错误了。

问题：

1 这个问题是因为量产时，格式化已经通过了，说明量产时写文件系统成功，读出来也没问题，说明文件系统没什么问题。

2 上电的时候，文件系统丢了，说明上电的时候，FTL出现问题了，需要看看映射表之类的。

解决方法：

1 看看格式化时候的读写的是文件系统的哪些地址，然后对比一下上电读写的是文件系统的哪些地址

思路：

1 通过打印看出来，在上盘的时候，设备一直在读同一个位置，6600000000000B840007，这个位置是逻辑地址0的地方，是MBR的地方。而量产时读的MBR地址页为，6600000000000A740000，明显两个地址不一样，但按理来说，得是一样的。所以需要看一下，分情况看一下：

* 1. 是量产时，写错了？
  2. 是上电时，恢复错了？

Retry1:01000B84000000008000：retry是因为数据校验错了，所以得retry。后面的地址是retry调整后的地址，也有可能越调越差的，所以这个地址既可能是好的地址，也可能是坏的地址。

F N SLC00000B84：这个是在找一个空cache，后面的地址表示这个地址是个空cache的地址

1.3

这3 byte是管理信息

SpareAreaL = XBYTE[\_SPARE\_AREA\_BUF + 0];

SpareAreaH = XBYTE[\_SPARE\_AREA\_BUF + 1];

SpareArea2 = XBYTE[\_SPARE\_AREA\_BUF + 2] & CACHE\_BLOCK\_TOTAL\_NUM\_MAX;

NF\_Read\_Start()

1 打开打印，将SD2 pin复用为uart，所以数据传输只使用了1线模式？

2 关闭打印，则默认使用4线模式？然后格式化会失败

## 解决方法

情况：量产时，文件系统数据读写成功，但是设备重新上电时，文件系统丢失。

分析：量产时文件系统数据校验成功，是因为量产时的校验将扇区地址的数据读上来，它时连续地，就算此时切column，跳地址失败，也不会有问题，只需要一个个扇区读上来就可以了，然而上电的时候，设备需要从MBR，找到DBR等信息，所以就会涉及到地址跳转问题，但是恰好这个flash在主控发生切column地址的命令时，失效了，导致地址没有切过去，文件系统数据就不对了。

原因：可能是由于fw在切column时，发命令失败，没有切过去，导致page映射到同一个block(00 00 )地址，LBA0地址为文件系统的地址，所以会导致上电时，文件系统丢失情况。方法：将上电切换地址的方式修改成发page地址时，就将column地址带上，就不用切column地址了，row + cloumn，这样就不用去切地址了，但是这种方法有一个缺点，就是只使用于连续sector读写的情况，假如一个page中有坏sector的话，则会读写到坏的sector，导致读出来数据不对/数据写不进去的情况，针对这种情况，还需要进一步看看。

1 上电切column有问题

2 flash RB存在问题

解决办法：在等完RB之后，在读状态，等待cache/array ready

# 4调速度的几个思路

1 调整配置文件(SDSetting.ini)的RcTrimming，将值调大，相当于加快了主控的速率

2 调整fw的SD\_HS\_CACHE\_BUF\_USE\_CACHE\_CNT的数量，将数量调大，相当于增加了收数据的Buff数量，数据收发更快了。

3关于nand flash几个思路

3.1 2 plane

3.2 cache

如果量产工具勾选了这两个选项，并且量产成功了，说明nand flash的2 plane和cache没有问题，是好的。



# 5 低格开不了卡

可以开debug模式，查看Channels#1文件夹的BcsSectorsA plane0的数据，这是低格产生的数据，可以看出flash的状态-好/坏。

# 6 H2失败

1 先将H2工具选择H2对比，它会把读写错误得数据保存下来。

2 打开打印，数据出错时，会打印它的逻辑page

3 看看相应的page:block的error bits和program status如果

3.1 当MLC的error bit，错误很多，并且大多数block都是这样的，则认为它在MLC模式下很差，需要降bin，将它降到SLC模式下，或许会得到改善。

3.2 如果当出现状态E1的时候，可踢出坏page即可。

38:014d 0089状态E1

4:03b7:0000 状态E1

26:017b:0003 error bit和status 都没错

去除坏page

33:03b9 001f mlc program status 很多错误

18:03af:0000 相邻特征都是坏的-SLC也有很多坏的

47:0044 0024 相邻特征都是坏的

降低mlc容量，使触发跑slc模式，会降bin

37:H2 PASS

46:h2 pass

31:H2 PASS

45:H2 PAS

42:03c7 0011 对比错误，但是r f对应的err bit/proram status都没有异常 - 重新低格，所有偶数slc block整个变坏了-------->按奇数,偶数block分开抽样扫多次，强制把特性绑定到所有奇数或偶数block.不好改

39:高格信息不完整，需要低格

15:打印是坏的，乱码

21:高格信息不完整

23:低格成功，但QCTOOL失败

27:量产时，卡初始化失败

30：量产成功，H2开卡格式化失败

## 6.1 变量和page/block的对应关系

//yScrtyUnitCnt = 100;

printHexSync(BECNTTmp);

prints("r f");

printHexSync(CurLg2PhTableAddrL);

//ER02和ER01在这里表示的是物理block

printHexSync(ER02);/\*R3\_BlockAddrH\*/

printHexSync(ER01);/\*R2\_BlockAddrL\*/

printHexSync(R4\_PageAddrH);/\*R4\_PageAddrH\*/

printHexSync(R0\_PageAddrL);/\*R0\_PageAddrL\*/

printHexSync(ER03);/\*R1\_Sector\*/

prints("|");

printHexSync(ZoneLBA);

printHexSync(BlockLBAH); //BlockLBA表示逻辑block

printHexSync(BlockLBAL);

//PageLBA在这里表示的是逻辑page

printHexSync(PageLBAH);

printHexSync(PageLBAL);

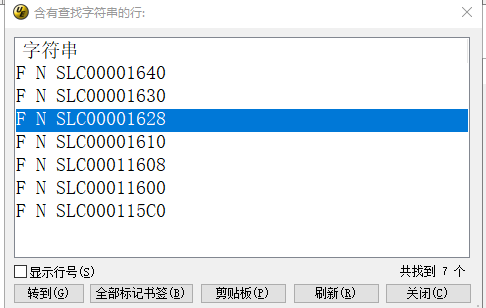
prints("\n");

F N SLC 00001640

这里0000代表的是逻辑block

1640是物理block

TLC是3个SLC对应一个TLC block，所以有可能是3个逻辑block对应一个物理block，也有可能是4个，或者多个。



## 6.2 解决方法

1 打开打印，使用QCTOOL/H2测试。测试失败，则在打印中会看见r f字符，表示retry fail。并且会打印出retry fail的物理block和逻辑

2 拿到物理block和逻辑block后，通过低格信息，找到Final\_PSBModel(step 6).txt文档，里面的信息有提供SLC还是MLC，然后在data block数据中找到对应的物理page

3 再查看mlc error bits plane0 cyc0\_Times0.txt和slc pro status plane0 cyc0\_Times0.txt文档

mlc error bits plane0 cyc0\_Times0文档 提供了ecc错误bit数信息

slc pro status plane0 cyc0\_Times0文档提供了program status的信息

4 分析这些信息，分情况讨论

1 指定的block ecc不大，但是相邻特征的block的ecc很大，则说明相邻特征block是坏的，可以尝试踢出那些block

2 如果ecc不大，但是program status错误，可以将program status错误状态踢出，会有所改善。

这两种方法都会导致容量下降。

均可使用物理block和物理page找到相应的状态信息。

将原本的低格信息，改成对应的通道，工具选择低格。

比较新的低格信息和原来的低格信息，看看两者有什么规律，有没有变坏，是奇数块坏还是偶数块坏，ecc有没有变大或者变小。

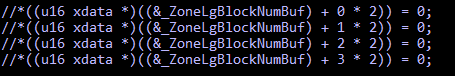
### 6.2.1 H2失败的一些思路

1 可以使用H2 TOOL来测试设备，如果测试失败，则可以在相应的通道文件夹找到ERR和OK的对比，分析在哪一个LBA错的，从而可以得到在ProInfo文件里查看是哪个plane错误。

例子：

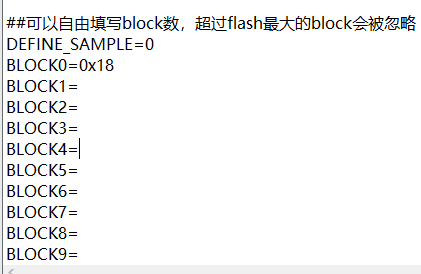
如果是plane1错误，则屏蔽plane0，直接跑plane1，进一步确认plane1的问题。

**如何屏蔽plane0：**



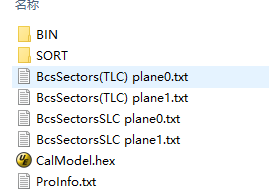
## mp扫出来是好块，fw使用的时候变坏

可以在配置文件里，将那些fw使用的坏块，填入DEFFINE\_SAMPLE的block块，然后开启低格，预扫完成后，即可停止低格。



如何判断预扫完成？

只要输出BcsSectors(TLC) planeX文件即完成了预扫，停止低格。



### 预扫目的：

看看再次低格，这个设置的block的ECC状态和初始的低格的block是否一致，如果不一致或者再次低格的block的ECC不可纠正，那就是这个block有问题。

### 预扫分析

通过r f 找到相应的block和page，看预扫的相应的block的ECC是否也是不可纠正的，或者看正扫的时候，能否和预扫的状态相互对应起来，如是，则这个block有问题，则要分析mp为什么会认为这是好块。

### 预扫的方法

是抽样的扫，但是在配置文件填入的block，会强制扫，填入的block要在同一个plane，不然会扫不到。

### DEFINE\_SAMPLE

DEFINE\_SAMPLE，表示抽样/预扫 扫描，它会将指定的block数加入到扫描行列之中



### TotalScanBlock

TotalScanBlock=0 通过设置扫描的block数，mp会扫描指定的block数，尽量让block都在指定范围之内

## 6.3 care status是什么意思？

将坏的状态和坏的page踢掉，只留下一些好的page。

## 6.4 变量说明

R1\_sector表示sector的偏移量

# 7 量产工具错误

## 7.1 错误代码04&05

可能flash有问题，导致开不出容量

## 7.2 错误代码50

## 7.2 error code:34

fw卡死了，代码跑死了。

或者填参把代码覆盖了。

在低格时出现error34，说明低格成功了，只是代码跑死了。

## 7.3 error08

### 7.3.1 fw解决方法

#### 4p时遇到的问题：

1.可能由于fw在ram中的随机种子错了

2.2p时在poer up set bound和读写 get bound的时候，bound的block不对。可能要查看其bound的算法有没有问题

3Sort\_SD.ini文件的参数变量有问题

4.Zone\_SIZE是指block的数量，fw里填的Zone\_SIZE要比flash spec的更大一些

5.上电填参的ZoneLBA信息可能超界，如sansumg的大概有64bytes，但是之前fw遍历时，遍历了128bytes。

6.zone cache可能溢出，215F+4P代码的有32个zone cache

### 7.3.2 一些思路

1.先区别是SLC出问题还是MLC出问题，取消copyback\_one\_order

2.如果没有打印出F N TLC，则可能说明是SLC出问题了，可能写cache有问题。然后看看低格信息的读写数据是则么错的，有无规律

3.change\_ce函数有判断flash类型，或者program的中间会有判断flash类型

4.Prog\_A\_Page\_CacheBlock();函数可以看到pragram时配置的命令和地址

5.注意看低格信息的中的读写对比数据有什么问题，比如错误的规律等等。

例子：

在6TDK的读写数据错误中，发现每8k的数据是错误的，16k buffer，8k为LP，8k为RP，所以可能后8k数据都是错误的说明RP数据fw没有读到正确的column地址，sector地址是fw从填参前后代码来看的，但是填参后代码是mp使用sort\_sd.ini文件填入的数据，所以可能mp填错了。

6可以看看sort\_SD.ini的 \_FLASH\_FUNCTIONEXT 值是什么，它代表了是SDR还是DDR模式，0x08是SDR，0x28是DDR。如果flash跑不了DDR，但是参数又填了DDR，则会导致格式化失败。

6.1 为什么flash是跑不了DDR？2p和DDR之间有什么关系？

### 7 为什么mp使用正常，fw使用异常

9T25：

可以把一个retry fail的块，fw只使用这一个块，然后看看数据是否能写进去(可使用bushound)，假如写不进去，则说明mp找的块有问题，再和mp去商量为什么会用这个块。

retry fail是在SLC还是TLC

通过直接写TLC方式，发现几乎每个块都出现了相同的ECC，这可能是不对的，猜测是这种方式存在问题，然后通过对比两次低格信息，发现ECC变化不大，现在通过稳定优先的开卡方式，然后对比低格信息试试看

## 7.4 error11

## 7.5 error 30

## 7.6 error36&40

Err36&40是因为没有插卡，Mp没读到卡导致的

## 7.7 error03

# 8 code相关问题

## 8.1 可以打印从主机接收的数据吗，或者发送给主机的数据

## 8.2 关于代码超code size

可以注释掉一些不用的命令，比如SD的擦除，CMD22/23?上次好像是这么注释的

## 8.3 fw 打印err3/err2

代表写下去立即读出来就已经错了。

# 9 卡出现系统提示磁盘驱动器问题

## 9.1 分析

系统提示驱动器问题，一般是因为系统检测到文件系统，大概率是MBR/DBR某个字段出现了错误造成的，但这个字段对整体的文件系统没有太大影响，所以系统只提示驱动器有问题。

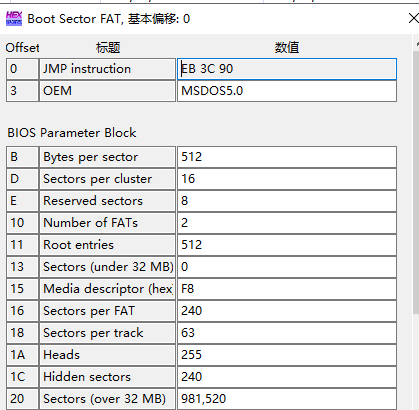
## 9.2 案例：荃天的215DA+K9F4GD8U0F

经过对比正常卡的文件系统和异常卡的文件系统，发现FAT文件系统的DBR的BPB扩展参数的reserved(保留)字段不一样。

图1 是正常的BPB扩展参数，**在0x25的位置unused的字段为0**

图2是异常的BPB扩展参数，**在0x25的位置unused的字段为1.**

**经过测试，将异常的unused的值修改为0，系统则不会显示驱动器异常。**



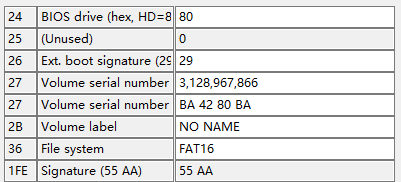
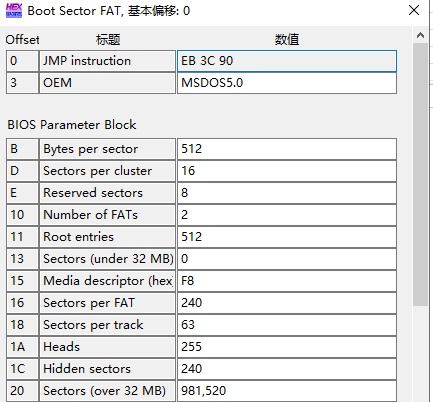


图1



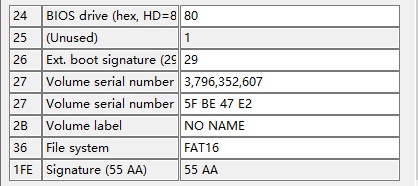


图2

## 9.3 解决方法

1 以后遇到类似问题，优先考虑文件系统的某些信息错误，检查文件系统某些字段。

## 9.4 一些问题

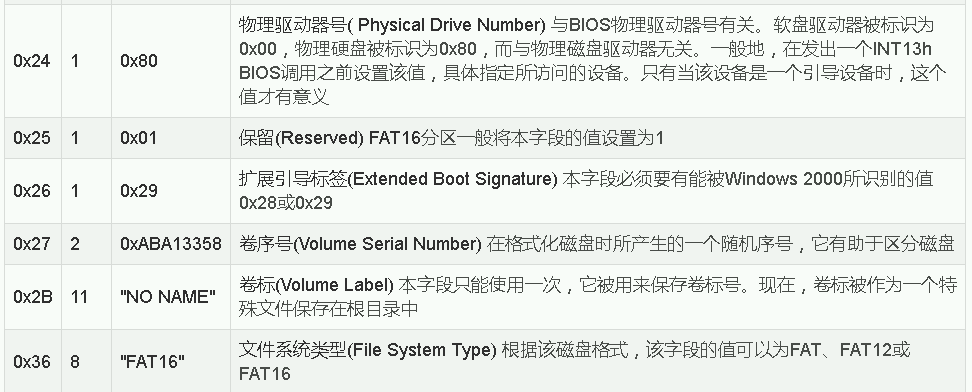
### 9.4.1 关于文件系统字段为何会异常

1 考虑是mptool在格式化写入文件系统信息时，误判某些信息导致的？但是不太确定是不是这个原因，但好像只有这个才能解释得通。

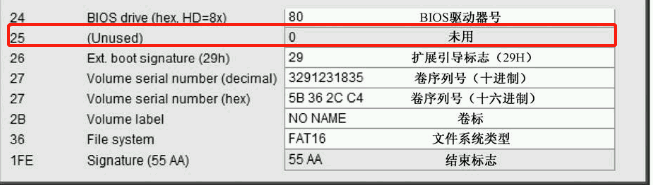
### 9.4.2 FAT文件系统的unused字段是干嘛的

只找到这些信息，但似乎还是不清楚，有矛盾的地方









10

使用读卡器将数据拷进flash/sd卡中，然后放入产品中，开机，然后取下来，再校验数据是否出错。

# 10 如何把fw放入任意文件夹烧入

# 11 低格信息

ProInfo.txt是扫描出来的bad block，fw不会使用这些坏块的。

# 12 为什么要填参

## 12.1 关于proinfo.txt的bad block

bad blk(含cache):这个表中包含cache block& bad block，如果想要看好的block，要fw打印出来看。

# 13 打印乱码怎么办

## 13.1 调整波特率

# 14 cache block和data block bound

Cache block是软件填参到Bin文件中去的，然后fw去bound

Data block是怎么样bound？

Find\_A\_Blank\_Block(void);函数中还有一次set bound，是data block set bound

# 15 打印F N TLC代表什么

代表3个SLC写完了，正在把3个SLC写入TLC

# 16 为什么SD卡在切到高速模式时，需要将时钟采样边沿从下降沿切为上升沿

在低速模式时，时钟采样边沿在下降沿采样数据，采样时间为10ns，在上升沿输出数据。

在高度模式时，时钟采样边沿切换为上升沿，在下一个上升沿输出数据，所以采样时间为

上升沿+下降沿的时间，依然为10ns.猜测这样改动是因为可以准确采样到数据。

10ns

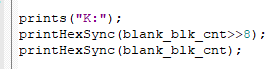
10ns

5ns

5ns

# 17 找不到block的原因

fw找的空块数要大于等于填参时候的空块数。



**\_FLASH\_ZONE\_LGBLOCK\_NUMBUF 表示的是flash有多少个空快数。**

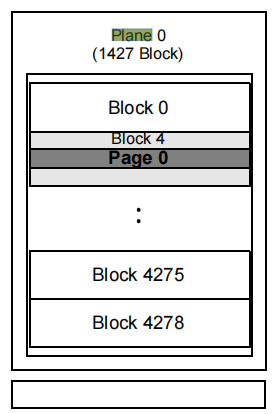
**00000310 \_FLASH\_ZONE\_LGBLOCK\_NUMBUF 0310 003C**

**打印坏块和好快的数量**

**1个plane有多少个block？看flash的结构**

**在上电时找到的好的block都是空的block，因为量产时会把它擦除。**

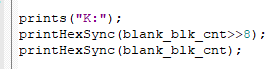
**每一个plane有1427 (0x593)block？**



1. 要么就是坏块表算错了
2. 要么就是填参填多了坏块表

分析原因：

通过打印可以知道fw找到的空(好)块数是多少



接着去ProInfo.txt去看BBT表中的坏块数量，去算一下好块数量是多少。是否和bin文件的填参地址( **\_FLASH\_ZONE\_LGBLOCK\_NUMBUF**)的值一样或者是否相差不大，如果相差很大，则是mp填参或者算坏块表算错了，需要mp改一下。

## 17.1 关于坏块表BBT

# 19 215F 4plane方案

使用p0和p1绑定在一起，p2和p3绑定在一起的方案。因为215F的缓存只有16kB，所以只能两个plane组成着来用，1个plane使用8k。

# 20 如何确定是否扫描了两次

看低格信息中的slc\_err\_bit\_plane\_cycle0和slc\_err\_bit\_plane\_cycle1，有两cycle，则说明扫描了两次

21 假如slc找错block了

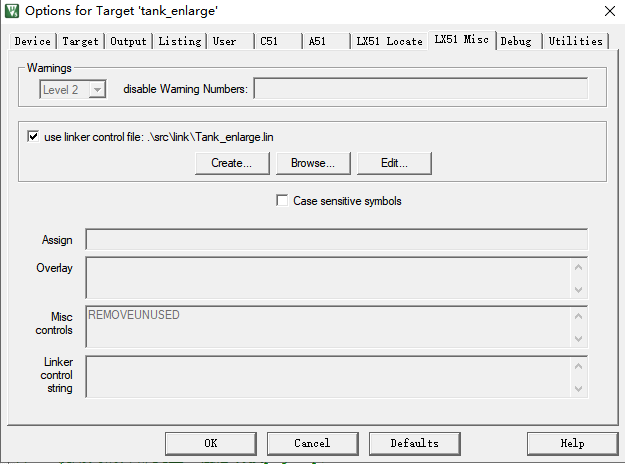
通过F N SLC可以看到slc的逻辑block和物理block的对应关系，如果错误，则去void SLC\_Find\_Reserved\_Blocks(void);函数去打印BlockAddrTmpH,

BlockAddrTmpL的值来查看fw遍历的SLC block，如果不正确，则说明可能代码填参可能错误，则可以打开填参后的代码，找到9c10那个位置，看看mp是否填对了参数。

fw可能找错位置了。

# 21 超过code size解决方法

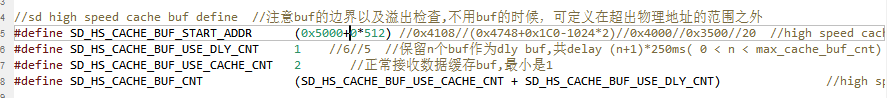
LX51 Misc



代码空间的映射关系在.map文件中

# 22 升级的相关问题

1.buffer的大小需要减小，升级之后相关的buffer需要减小的。



SDsetting.ini文件中有关升级的有以下三个值：

EnLargerVersion=0x13

EnlargerVersionReadDelayCnt=0x00

EnlargerVersionWriteDelayCnt=0x00

## 22.1 升级的测试

1.拷贝比实际容量多的数据，不会掉盘--包括过H2、拷贝文件等。

2.看H2/拷贝文件的正确的数据，是在实际容量之内吗

# 23 keil 报错

## 23.1 \*\*\* ERROR L120: CONTENT BELONGS TO ERRONEOUS SEGMENT

这个报错好像是因为keil空间小，代码+变量等的空间超过了keil的ram空间

215F为什么要放那么变量在data空间，是因为，DPTR指针和xdata区有关联，使用xdata区的变量，如果DPTR设置自增的话，会导致xdata变量取值不准确。

## 23.2 keil的link文件

# 24 变量解释

## 24.1 物理block：

R3\_BlockAddrH

R2\_BlockAddrL

物理page：

R4\_PageAddrH

R0\_PageAddrL

密钥种子：

DataCoDecKeyIndex

管理空间:

SpareAreaL

SpareAreaH

SpareArea2

(u16)\_ZoneLgBlockNumBuf---表示的是plane的number，占2个字节

## 24.2 ZoneLBA和Zone

1.Zone在215F中代表的是plane，一个zone中，page模型都是一样的。如果是2 plane，2个plane为1个zone，如果是4p，则4个plane为一个zone。

2.215DA是512 block为一个zone。

# 25 ECC

SLC/MLC开12/15 bits，TLC开20/24 bits。

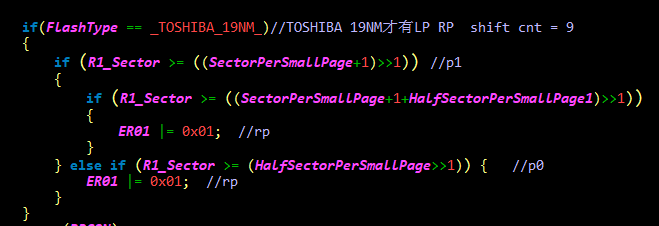
mp开ecc的最好/大值是flash ecc的最大值的三分之二

# 26 flash

## 26.1 6TDK

分为LP/RP

fw使用ER01是在LP/RP，ER01是0，表示是LP，为1代表RP



# 27 逻辑地址空间和物理块空间怎么对应起来

# 28 retry table

是当flash ecc很大时，可以尝试使用retry table的数据来把ECC减小。

Retry table包括前缀 命令 retry数据

retry数据可能有7组，每组7个数据

# 29 homs medule

# 30 三星4p

1 Code中KK:打印的是3 bytes 物理地址，表示哪个block的哪个page，从打印来看，打印了1570 block，但是从ProInfo文件来看，没有1570这个块，所以可能上电set bound，或者get bound的时候，有错误，所以查看power up set bound信息

2 通过加打印在power up set bound函数，发现code没有走到那里，可能上电填参有问题导致条件不满足

# 31 BCM

# 32 GC

Copy\_back会触发GC，然后block会记录每个page的有效扇区数，有效扇区数越少，垃圾越多，怎么哪个扇区是有效，哪个扇区是无效的？通过每写一个page，就会记录LBA->phy的mapping信息，通过读这个信息，来判断phy是否对应相应的LBA，如果对应，则是有效数据。

215F是block映射，它没有gc，只有cache需不需要整理，如何整理的问题

1.215F一般什么时候会做copy back

2.在prepare\_write\_lba时，为什么先要chk\_need\_copyback

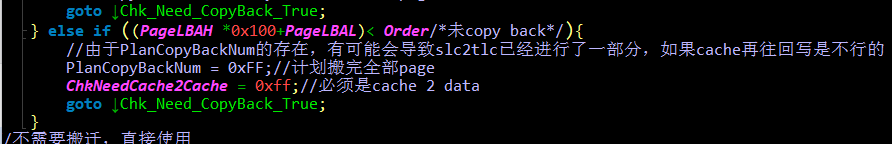
3.copy\_back\_0，copy\_back\_1，copy\_back\_2这几个有什么区别

1 需要整理了就copy，因为host写的顺序是随机的；

2 要先判断是否需要搬迁，释放空闲的swap来

swap相当于是一个交换区，因为host写的数据都是随机的，而215f是block管理算法，所以就需要处理

3 分别是copy n个page、copy一个page的前面的sector、copy一个page后面的sector



因為 cache 是 ring buffer，同時slc2tlc 不是 atomic 做完 可能會分段，所以不能夠回頭寫 必須設為初始

命中率低是什么意思？

就是有一些block存放的是冷数据，不是经常访问的，则需要把这种block做gc，因为这些数据不经常访问，占用一个block太浪费了。

gc的两种方式：

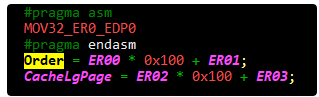
Merge:是block写满了，则找一个空block做gc

Swap:是将一些

挑选不经常访问的数据，然后将不访问的组从swap中踢出，因为swap buf存的是经常访问的热数据

如何合并

就是看有幾組 buff 空間



# 33 fw里为什么要做随机

因为适应flash的特性，flash对有规律的数据会有影响，所以要把数据做随机再存入flash，读出来再反随机化，返回给主机

# 34 空块表

1 215F上电时，会从填参地址拿到空块，然后建表

2 在写入资料时，会从空块表，找出空块，使用空块存入资料

3

1. lg2ph表 都是呼叫 Write\_WritingBlockBuf() 更新

a. powerup 時 建表

b. program 前 觸發 GC 時

c. program 結束後 更新表內容

2. blank table 更新時間,

a. powerup 時 建表

b. 呼叫 Find\_A\_Blank\_Block() 找空block.

c. 呼叫 Mark\_A\_Blank\_Block() 釋放空表