GBBM2.3 User Manual

General Description and Name

This scheme Implements the Samsung GBBM2.3 bad block handling type. It uses an extra table for LPIA information. (Please refer to GBBM 2.3 spec for some abbreviations). This BBM is only for OneNand 2Kbyte or 4Kbyte page device.

Relevant User Options

The following special features on the special features tab apply to this scheme. The default values might work in some cases but please make sure to set the right value according to your system.
Please note only the below special feature items are related to this scheme and ignore any others. If any of below items doesn't exist, please check whether the right version has been installed or contact Data I/O for support by submitting Device Support Request through this address:
http://www.dataio.com/support/dsr.asp

Bad Block Handling Type = "GBBM2.3"

Spare area : Please refer to "Description of common NAND special features.pdf". ***Normally set as " Enabled" or "Disabled" for this BBM***.[Default 'Disabled']

Unlocked area: Start block = "0x?" It depends on the customer in hex.

RBA area: Number of blocks = "?" It depends on the customer in dec. Samsung recommends this value = Max number of bad blocks + 6 extra block. Take 1024 blocks as an example, it should be 26.

RBA area: Start blocks = "?" It depends on the customer in dec. It equals number of the device blocks – RBA area : Number of blocks.

UBA area: Start blocks = "?" It depends on the customer in dec.

Partition Table Format start size = "BLOCKS" OR "BYTES". It depends on what type of table is used.

Partition Table File = "?". The path of the PIA table file.

CPU Endian = "?". It depends on the CPU Endian. Little Endian or Big Endian. So far only little Endian is supported.
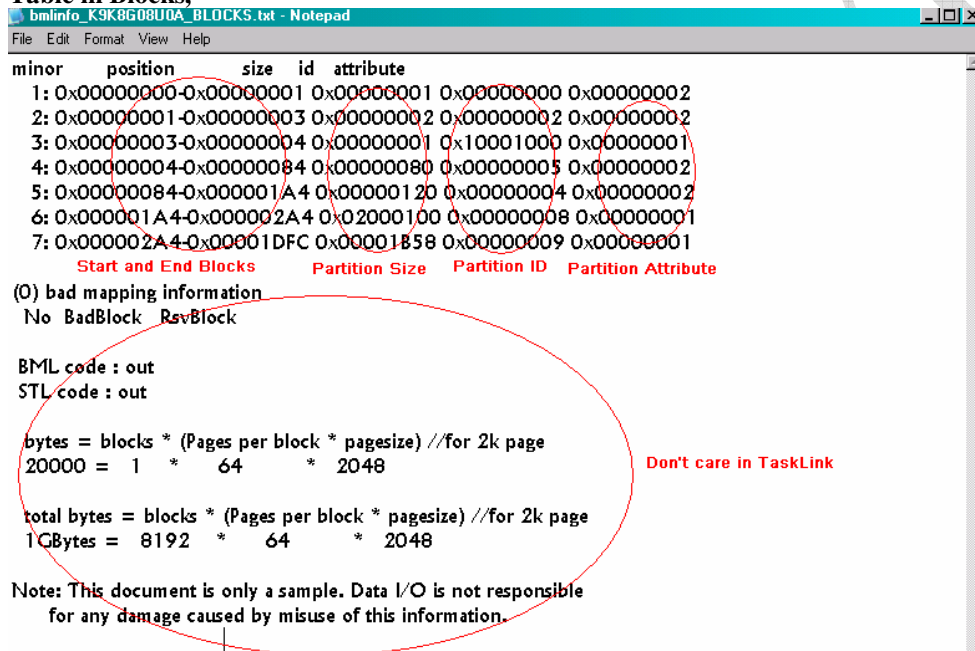
Special Notes

Bad block table called BMS in GBBM2.3.

If this scheme, or any bad blocking scheme is used, there are implications to the checksum of the image. As with any device, TLWin produces a checksum over the entire image rather than just the user data file.

If the chosen spare area option is "enabled", then the checksum will not be affected. It will be a checksum of the user data plus FFs to the end of the device image.

Structure of Table

Table in Blocks,

```
bmlinfo_K9K8G08U0A_BLOCKS.txt - Notepad                                    _ □ ×
File  Edit  Format  View  Help

minor      position        size   id   attribute
  1: 0x00000000-0x00000001 0x00000001 0x00000000 0x00000002
  2: 0x00000001-0x00000003 0x00000002 0x00000002 0x00000002
  3: 0x00000003-0x00000004 0x00000001 0x10001000 0x00000001
  4: 0x00000004-0x00000084 0x00000080 0x00000005 0x00000002
  5: 0x00000084-0x000001A4 0x00000120 0x00000004 0x00000002
  6: 0x000001A4-0x000002A4 0x02000100 0x00000008 0x00000001
  7: 0x000002A4-0x00001DFC 0x00001B58 0x00000009 0x00000001
          Start and End Blocks    Partition Size   Partition ID   Partition Attribute
(O) bad mapping information
  No  BadBlock  RsvBlock

BML code : out
STL code : out

bytes = blocks * (Pages per block * pagesize) //for 2k page
20000 =   1    *     64      *   2048                     Don't care in TaskLink

total bytes = blocks * (Pages per block * pagesize) //for 2k page
1GBytes =  8192   *    64       *   2048

Note: This document is only a sample. Data I/O is not responsible
      for any damage caused by misuse of this information.
```

Table in Bytes,

```
bmlinfo_K9K8G08U0A_BYTES.txt - Notepad
File  Edit  Format  View  Help
minor      position         size     id             attribute
   1: 0x00000000-0x00020000 0x00020000 0x00000000 0x00000002
   2: 0x00020000-0x00060000 0x00040000 0x00000002 0x00000002
   3: 0x00060000-0x00080000 0x00020000 0x10001000 0x00000001
   4: 0x00080000-0x01080000 0x01000000 0x00000005 0x00000002
   5: 0x01080000-0x03480000 0x02400000 0x00000004 0x00000002
   6: 0x03480000-0x05480000 0x02000000 0x00000008 0x00000001
   7: 0x05480000-0x3BF80000 0x36B00000 0x00000009 0x00000001

(0) bad mapping information
 No  BadBlock  RsvBlock

BML code : out
STL code : out

 bytes = blocks * (Pages per block * pagesize) //for 2k page
 20000 =   1   *    64       *   2048

 total bytes = blocks * (Pages per block * pagesize) //for 2k page
 1GBytes =  8192  *   64       *   2048

Note: This document is only a sample. Data I/O is not responsible
     for any damage caused by misuse of this information.
```

The difference between the 2 types of tables only lays in the position and size information.

Revision History

V1.0   Feb. 10, 2010
        Create this spec.

Appendix

You can get the file "Description of common NAND special features.pdf" from
http://ftp.dataio.com/FCNotes/BBM/