
Mixed style Multiple Partition User Manual

General Description and Name

This BBM is for the system that has several partitions which might have different bad block handling style even store BB table.

Please note the *padding blocks* (which don't need to be programmed) within each partition or not belong to any partitions should be stuffed with all 0xFF in customer's data file. Only the partition table specified blocks of each partition will be programmed (which means all other blocks will not be programmed).

For the BB table partition, programmer will only update the signature, version, ECC and bitmap field within the 528byte-sized-pages which needs to store BB table, all other fields and pages will be kept as the data file.

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 = "Mixed style Multiple Partition"

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

PartitionTable File : Point to a .mbn file which describes the partition information.

Check BB Marker In DataFile : Please refer to "Description of common NAND special features.pdf". ***please normally set as "disabled" for this BBM.***[Default 'Enabled']

bad block detection : Please refer to "Description of common NAND special features.pdf". ***please normally set as "BBM then BB marker" for reprogramming under BBM.***[Default 'semi vendor BB mark']

Special Notes

Format of PartitionTable.mbn:

- a. Binary file fixed length 256 bytes.
- b. Organization:16 rows x 4 columns. Each table item is 32-bits, little endian byte ordering.
- c. Each row of the table describes configuration for one partition. Up to 16 partitions can be used.
- d. Partition configuration:
 - i. **Start Adr**: address of start of partition in flash blocks. The programmer will set the file read pointer and the programmer write pointer to Start Adr. If Start Adr=0xFFFFFFFF, skip to the next partition.
 - ii. **End Adr**: last valid block in the current partition. The last data block programmed must be equal to or less than End Adr, otherwise the programmer will reject the flash device.
 - iii. **Actual Data Length**: number of blocks of data to read from the input file and write to the flash in the current partition
 - iv. **Attribute**: specify the attributes for current partition.
 - bit0~7, specify the bad block handling style.
 - 0x00 for none and abandon the data if bad block met;
 - 0x01 for skip and program the data to next good block if bad block met.
 - 0x02 this partition stores BB table, and data will be programmed to next good block if bad block met.
 - 0x03 this partition stores BB table, and data for mirror table will be abandon if bad block met[data for primary table will work as 0x02 option.].
 - (currently only these 4 styles supported.)

Bit16~31, specify how many bad blocks allowed in current partition.

if bit0~7 is 0xFF, then defaultly skip style(0x01) will be used; if bit16~31 is 0xFFFF, then defaultly maximum bad blocks will be allowed. *Please note to close the BlankCheck operation if "none" style existed!*

Please note to keep: Actual Data Length + max bad blocks allowed <= End Adr - Start

Adr + 1

- v. Example PartitionTable.mbn file:



H:\ExchangeFiles\
PartitionTable_mixed.

NAND Flash Block			
Start Adr	End Adr	Actual Data Length	Attribute
0x0	0x7FF	0x360	0x90000
0x800	0xFFF	0x30	0xFFFF0001
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF

Revision History

V1.0 June 11, 2009
Create this spec.

Appendix

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

Data I/O