

Multiple Partition Style Specification

General Description and Name

This BBM is for the system that has the number of partitions less or equals to 256. With the configuration of attribute data in partition table file, each partition allows to have different block handling style – None Style and Skip Style.

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).

Relevant User Options

The following special features on the special features tab apply to this scheme. The default values should work in normal cases but you are free to select any value if needed.

Bad Block Handling Type = "Multiple Partition Style"

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

Spare area : Please set as enable if data file includes the spare area data. *Please normally set as "Enable" for this BBM* [Default not selected]

Check BB Marker In DataFile : Please set as disabled if needs to override the semi-vendor specified BB mark, otherwise set as enabled. *please normally set as "Enable" for this BBM.*[Default 'Enabled']

Format of PartitionTable.mbn:

- a. Binary file fixed length 4096 bytes.
 - b. Organization: 256 rows x 4 columns. Each table column is 32-bits, little endian byte ordering.
 - c. Each row of the table describes configuration for one partition. Up to 256 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.
 - bit0~31, 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.
 - (currently only these 2 styles supported.)
- if bit0~31 is 0xFFFFFFFF, then defaultly skip style(0x01) will be used.
- 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:

<http://ftp.dataio.com/FCNotes/Footnote/PartitionTable.zip>

Partition Table used by Data I/O

NAND Flash Block			
Start Adr	End Adr	Actual Data Length	Attribute
0x0	0x7FF	0x360	0x00000000
0x800	0xFFF	0x30	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