X-Loader Skip User Manual

This scheme is a Generic Multiple Partition format. It uses skip method for bad blocks handling within each partition. Bad blocks within any partition do not affect the starting location of the other partitions.

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 = "X-Loader Skip"

Spare Area = "ECC"

- This option should be selected to enable creating require ECC data in spare area when data file does not already contain spare area ECC information.

X-Loader Skip: X-Loader ECC Number of blocksArea = "x"

- "x" is the number of blocks from the beginning of the file that are of the X-Loader format. The partitions that have the X-Loader format have a different structure for the storage of the ECC values in the spare area of the device.

All other Special Features are not required for using this this BBM scheme..

Image Preparation
The image data file should have format shown in example below. The value of "FF" should be used to pad between each partition.

Partition Table Format is standard “C:\PartitionTable.mbn” format:

- Following Bit Map provides an example image file format and associated "PartitionTable.mbn" file:

```
The image file should be created using the following table.

(byte addresses)
0x00000000-0x00020000 : "X-loader(1) Image"
0x00020000-0x00040000 : "X-loader(2) Image"
0x00040000-0x00060000 : "X-loader(3) Image"
0x00060000-0x00080000 : "X-loader(4) Image"
0x00080000-0x000c0000 : "U-boot"
0x000c0000-0x000e0000 : "U-Boot Environment"
0x000e0000-0x002e0000 : "Linux Kernel"
0x002e0000-0x00ae0000 : "Ram Disk"
0x00ae0000-0x046e0000 : "System Data"
0x046e0000-0x08000000 : "Users Data"

The above partitions should be represented in a '*.mbn'
file as follows:

00000000h: 00 00 00 00 01 00 00 00 01 00 00 00 FF FF FF FF
00000010h: 01 00 00 00 02 00 00 00 01 00 00 00 FF FF FF FF
00000020h: 02 00 00 00 03 00 00 00 01 00 00 00 FF FF FF FF
00000030h: 03 00 00 00 04 00 00 00 01 00 00 00 FF FF FF FF
00000040h: 04 00 00 00 06 00 00 00 02 00 00 00 FF FF FF FF
00000050h: 06 00 00 00 07 00 00 00 01 00 00 00 FF FF FF FF
00000060h: 07 00 00 00 17 00 00 00 0A 00 00 00 FF FF FF FF
00000070h: 17 00 00 00 57 00 00 00 14 00 00 00 FF FF FF FF
00000080h: 57 00 00 00 37 02 00 00 90 01 00 00 FF FF FF FF
00000090h: 37 02 00 00 00 04 00 00 C2 01 00 00 FF FF FF FF
000000a0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000b0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000c0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000d0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000e0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Please note that the above example of an '*.mbn' file uses
arbitrary numbers for the length of each partition.
```

- To create the "PartitionTable.mbn" with fixed length of 256 bytes, refer to the following instructions:
  1. Binary file fixed length 256 bytes.
  2. Organization:16 rows x 4 columns. Each table item is 32-bits, little endian byte ordering.
  3. Each row of the table describes configuration for one partition. Up to 16 partitions can be used.
  4. 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 written to Device in the current partition.
     -- **NOTE: Actual Data Length + maximum # bad blocks allowed <= End Adr - Start Adr + 1**
     iv. Example “PartitionTable.mbn” file:

NAND Flash Block			
Start Adr	End Adr	Actual Data Length	Reserved
0x0	0x7FF	0x360	0xFFFFFFFF
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
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF

Exceptions

- None

Revision History

V1.0 Date 1/16/2013

Appendix

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