Skip Multiple Partition MI User Manual

General Description and Name

Skip Multiple Partition MI. This scheme Implements the skip block method for bad block handling but allows the user to create up to 16 partitions in the device, using a file with a .mbn extension. The format for the .mbn file and each partition is specified below.

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 = "Skip Multiple Partition MI"

Spare Area = "Disabled". This means the programmer expects the spare area data not included in the image.

Internal ECC state during programming = "Enabled". This BBM will use internal HW ECC.

PartitionTable File = The path of the partition table file on your PC.

Required good block area: Start block = "x". Please fill MI partition start block.

Required good block area: Number of blocks = "x" The total number of MI partition blocks.

Special Notes

.mbn file format

1. 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. **Spare**: reserved for future use.
    v. Example PartitionTable.mbn file:

```
00000000h: 00 00 00 00 F7 00 00 00 50 00 00 00 FF FF FF FF ;
00000010h: F8 00 00 00 FE 07 00 00 B1 01 00 00 FF FF FF FF ;
00000020h: FF 07 00 00 FF 07 00 00 01 00 00 00 FF FF FF FF ;
00000030h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000040h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000050h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000060h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000070h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000080h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000090h: FF FF FF FF FF FF FF FF FF FF FF FF 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 ;
000000f0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
```