NewSTO MP With BBT BBM User Manual

General Description and Name

For this BBM, it is not allowed to program NEW and Programmed devices together. Otherwise it will fail in detecting bad blocks. It is fine just to program NEW devices in all sockets or just program Programmed devices in all sockets.

This BBM uses partitions. For each partition, it depends on ATTRIBUTE for programming method and bad block processing.

The last two good blocks are BBT blocks.

It just supports 8 bit width devices in this version.

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 = "**NewSTO MP With BBT**"

Spare area : ***Set as "Update ECC field" for this BBM***.
Please refer to "Description of common NAND special features.pdf". [Default 'Disabled']

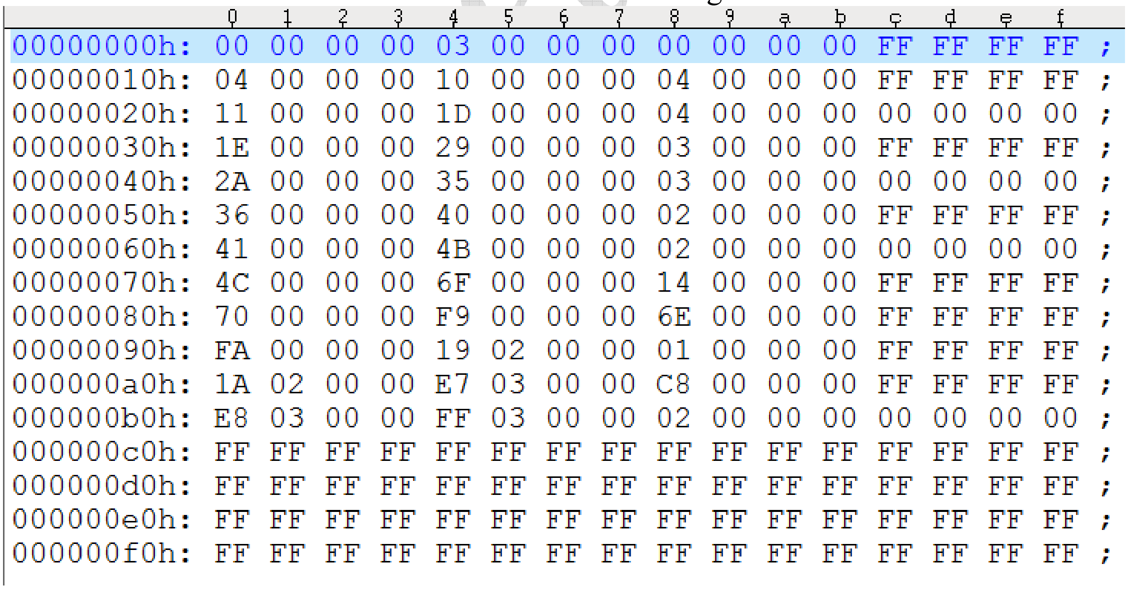
bad block detection:
Normally set as "BBM then BB marker" for this BBM.
This item is useful when customer needs to move the bad block marker location or want to override default bad block markers but use bad block table to record the information. Then we can only use the BBM specified way to detect the bad blocks while reprogramming.
BBM then BB marker - Firstly work as "*BBM specified*", if it failed to find the bad block information, then work as "*semi vendor BB marker*" to find the bad blocks.
semi vendor BB marker - The normal way, found the bad bock according to semi-vendor specified bad block marker. Actually the algorithm will work as earlier source code if this option selected.
BBM specified - This option will give the detection work to BBM. So, it can use any bad block marker location even in main area or bad block table to detect the bad blocks.
[Default '*BBM then BB marker*'.]

Do not program NEW and Programmed device together. Otherwise it will fail in detecting bad blocks.

Check BB Marker In DataFile:
Whether check the bad block mark position of the customer data file or not. Every NAND chip has its own semi-vendor specified bad block mark located in spare area, if this item enabled, programmer system will check the contents of this marker location and throw error if not all 0xFF.
[Default 'Enabled' if not listed or specified.]

PartitionTable File = The path of the partition table file on your PC.
For each partition, it includes "Start Block", "End Block", "Length" and "Attribute".
Start Block(4 Byte): Start block in the partition.
End Block(4 Byte): End block in the partition.
Length(4 Byte):   How many blocks to be programmed in the partition.
Attribute(4 Byte):
0xFFFFFFFF: Program from Start Block to End Block; Use the next good block when meeting bad blocks. For example, when block 10 is bad, program the data to block 11 if block 11 is good.
0x00000000:   Program from End Block to Start Block. Use the next good block in reverse order when meeting bad blocks. For example, when block 90 is bad, program the data to block 89 if block 89 is good.
Put un-used partitions (Filling "FF") at the end.

Start Block   End Block   Length   Attribute



```
             0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 00 00 00 00 03 00 00 00 00 00 00 00 FF FF FF FF ;
00000010h: 04 00 00 00 10 00 00 00 04 00 00 00 FF FF FF FF ;
00000020h: 11 00 00 00 1D 00 00 00 04 00 00 00 00 00 00 00 ;
00000030h: 1E 00 00 00 29 00 00 00 03 00 00 00 FF FF FF FF ;
00000040h: 2A 00 00 00 35 00 00 00 03 00 00 00 00 00 00 00 ;
00000050h: 36 00 00 00 40 00 00 00 02 00 00 00 FF FF FF FF ;
00000060h: 41 00 00 00 4B 00 00 00 02 00 00 00 00 00 00 00 ;
00000070h: 4C 00 00 00 6F 00 00 00 14 00 00 00 FF FF FF FF ;
00000080h: 70 00 00 00 F9 00 00 00 6E 00 00 00 FF FF FF FF ;
00000090h: FA 00 00 00 19 02 00 00 01 00 00 00 FF FF FF FF ;
000000a0h: 1A 02 00 00 E7 03 00 00 C8 00 00 00 FF FF FF FF ;
000000b0h: E8 03 00 00 FF 03 00 00 02 00 00 00 00 00 00 00 ;
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 ;
```

Version Number: Set version number for BBT blocks.
[Normally not used, default: 1.]

Required good block area: Start block:
Required good block area: Number of blocks:
These two items specify a range of blocks which must all be good. The device will be failed if any bad blocks found within this area.
[Always required, default ‘0’ and ‘0’.]

Special Notes

Revision History

V1.1 July 14, 2015
Add two “special features”

V1.0 June 26, 2015
Create this spec.

Appendix

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