

---

# **Monahans with Internal ECC BBM User Manual**

## **General Description and Name**

Monahans with Internal ECC BBM. This scheme mainly detects bad blocks in the device and program the data to the reserved area. For example, if block 7 of a device is the first bad block, then block 7 of the image will be programmed into the last good block of the device. This BBM uses device internal ECC for all blocks.

## **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 = **“Monahans with Internal ECC”**

Spare area : Please refer to “Description of common NAND special features.pdf”. *Normally set as “ECC” for this BBM.* [Default ‘Disabled’]

MON: Number of Reserved blocks = the NUMBER of blocks reserved for the bad blocks. Normally **16**.

The following special features are optional and can be ignored if default value works.

MON: Force filling FF to reserved bad block tables = whether fill the pages of the reserved bad block table to all 0xFF, this item is only required while these pages of customer data file is not blank. [Default ‘Disabled’]

MON: Next Block of Reserved Area = The NEXT block index after the reserved area. This item is used to specify the reserved area location and normally keep its value as the block amount of the device. [Default as device block amount]

## **Special Notes**

None.

## **Revision History**

V1.0 May 25, 2011

---

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