
BBM ZIMG User Manual

General Description and Name

This scheme Implements the skip block method within particular partition. There are 3 identical Block header at the certain position.

The last 4 physical blocks should reserve for the BB table (BBT). One is primary, the other is the mirror.

This BBM also overwrite the MFG BB mark position. Pay attention to keep the BBT.

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 = "BBM ZIMG"

Spare area = "Disable"
Or "Enable" depend on your data file.

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

Check BB Marker In DataFile = "Disabled"

BB: mark position = "6"
To this BBM, the MFG BB mark is at the 6th offset of the OOB.

Bad block detection = "BBM then BB marker"

ZIMG: SwVersion = depend on your file. this value can be get from File header or the File tail.[default 0x161]

ZIMG: TotalBlocks = depend on your file. usually this is the block number of the KFS area. [default 0xCA]

ZIMG: BlockHeadStartPartition = "2"
Set the Block header start partition from the "PartitionTable File", e.g. for this case it is the 2nd row of the partition. Then this value should be 2.

User Checksum Implications

If this scheme, the TaskLink checksum will be different from the device checksum because of the bad block table that contains possible bad blocks list.

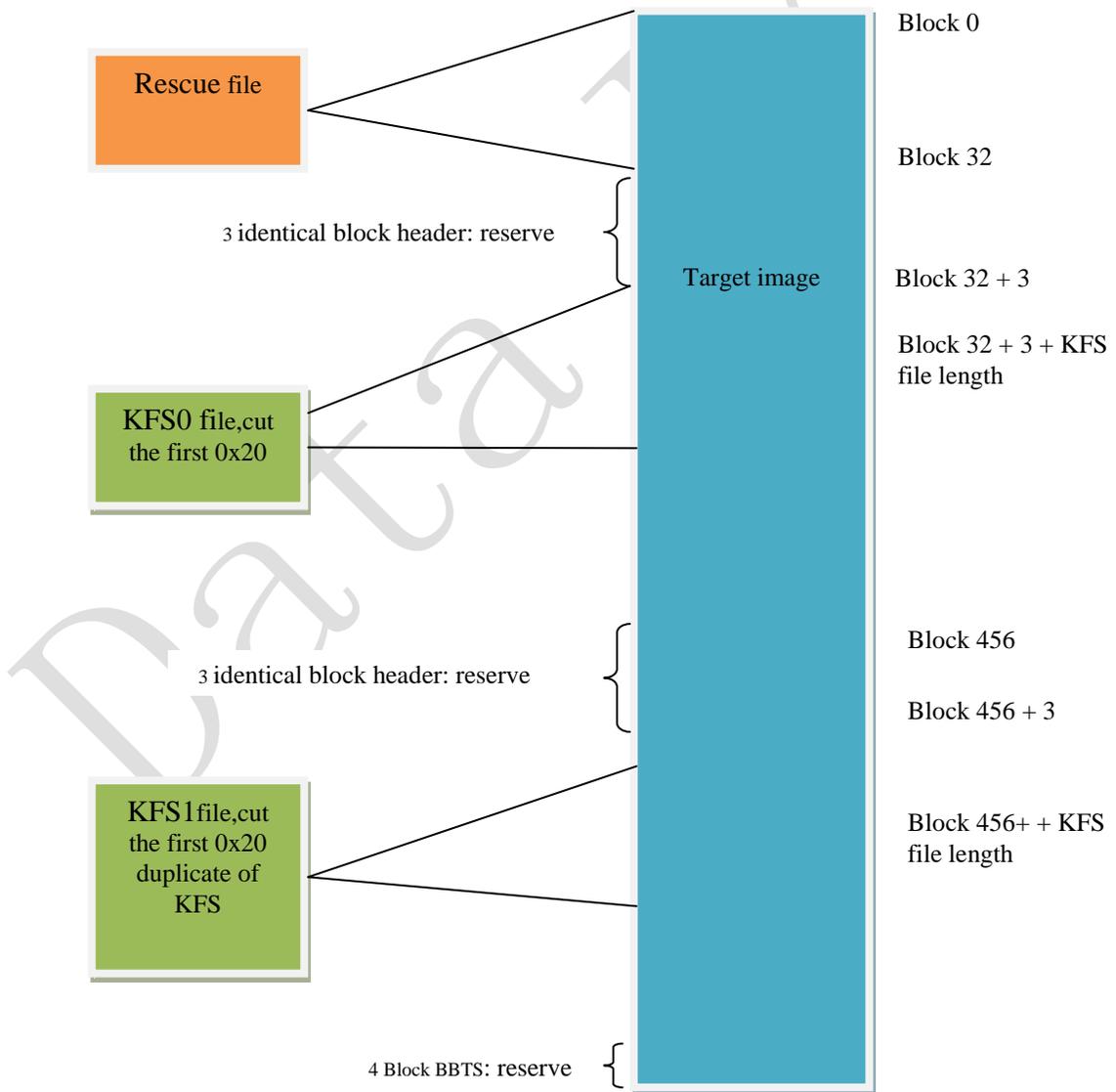
Image Preparation

This scheme support at least two file merged together.

One is the “rescue” file, other is the “KFS” file.

Due the KFS contain 0x20 Bytes of File header, which need to be cut before merge.

Also, keep the KFS position = KFS start position + 3 blocks. Because the 3 identical block header need to be reserved. The block header will be generating dynamic once programming.



Partition file Preparation

The last four blocks of the device should be reserved to store Bad block tables (BBTs). It also means your partitions shouldn't contain the four blocks for any other usage.

Special Notes

- Pay attention the BB mark position, make sure it is 0xFF.
- 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. **Attribute**: ignore

Please note to keep: Actual Data Length <= End Adr - Start Adr + 1 for this BBM

v. Example PartitionTable.mbn

NAND Flash Block			
Start Adr	End Adr	Actual Data Length	Attribute
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

file:

Revision History

- V1.0 Date: 2013-01-30
Create this spec.
- V1.1 Date: 2013-03-22
Change S.F.
- V2.0 Date: 2013-04-24
Change Picture for memory map

Appendix

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

Data I/O