Skip DBBT MPF BCH16 iMX6 D3 User Manual

This is a Multiple Partition format scheme. Any bad blocks within the first two partitions (FCB & DBBT) are dropped (up to "iMX FW Starting Block #"). The rest of partitions use "Skip bad block" method within each partition. This BBM is similar as Skip DBBT MPF iMX6 D3, difference is DBBT calculate 26 Byte ECC.

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 DBBT MPF BCH16 iMX6 D3 "

Spare Area = "Enabled "

PartitionTable File = "C: \PartitionTable.mbn "

Error bits allowed in one page: How many error bits allowed within one paged while preprogramming, this depends on the ECC method. [Normally required, default is 0].

iMX FCB Copies = "8 "

iMX DBBT Create? = "Enabled "
When this option is enabled, Discovered Bad Block Tables (DBBT) is added.

iMX DBBT Copies (if enabled) = "8 "
Eight copies of DBBT table (including CRC and ECC) will be generated and programmed into the NAND device.

iMX FW Starting Block # = "10 ", hex value, means start block #16.
This FW starting block # has to be greater or equal to FCB & DBBT values.

iMX DBBT last partition included in DBBT = "4 "
DBBT only contains boot area.

iMX ETFS Partition - Erased Block Start # = "FFFFFFFF ", The start block which contains erased signature only, for the erased block, only the first page will be programmed. Customer could set this value, otherwise the software will scan the data file and find the start block.

Image Preparation:

Customer data file contains FCB, DBBT, and other parts. The image file contains ECC. DBBT and its ECC will be update during programming. DBBT parts use BCH16 method, 26 bytes ECC.

Partition Table Format:

- A binary file with fixed length of 256 bytes.
- Organization: 16 rows x 4 columns. Each table item is 32-bits, little endian byte ordering.
- Each row of the table describes configuration for one partition. Up to 16 partitions can be used.
- Partition configuration:
  - i. **Start Addr**: address of start of partition in flash blocks. The programmer will set the file read pointer and the programmer write pointer to Start Addr. If Start Addr is 0xFFFFFFFF, skip to the next partition.
  - ii. **End Addr**: last valid block in the current partition. The last data block programmed must be equal to or less than End Addr, 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. **Note**: For optimal option, the following example should be used and the 3rd to the last partitions should be adjusted as needed.

Your NAND device have partitions, we need the partition information(start address, length). Please supply:

1. We have 8 blocks for 8 FCBs (start block 0, 8 blocks). Cannot grow.
2. We have 8 blocks for 8 DBBTs (start block 8, 8 blocks). Cannot grow.
3. One (or maybe more) blocks for 1st copy of bootloader (start block 16, nominal 1 block). Can grow to maximum of 24 blocks.
4. One (or maybe more) blocks for 2nd copy of bootloader (start block 40, nominal 1 block). Can grow to maximum of 24 blocks.
5. First copy of BSP block (start block 64, nominal about 72 blocks). Can grow to maximum of 100 blocks.
6. Second copy of BSP block (start block 164, nominal about 72 blocks). Can grow to maximum of 100 blocks.
7. First copy of reprog. block (start block 264, nominal about 72 blocks). Can grow to maximum of 100 blocks.
8. Second copy of reprog. block (start block 364, nominal about 72 blocks). Can grow to maximum of 100 blocks.
9. ETFS partition – starts at block 512, takes all the space to the end. If bad blocks are skipped, it is safe not to write the blocks at the end.

PartitionTable.mbn

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
00000000h:	00	00	00	00	07	00	00	00	08	00	00	00	FF	FF	FF	FF	;
00000010h:	08	00	00	00	0F	00	00	00	08	00	00	00	FF	FF	FF	FF	;
00000020h:	10	00	00	00	27	00	00	00	01	00	00	00	FF	FF	FF	FF	;
00000030h:	28	00	00	00	3F	00	00	00	01	00	00	00	FF	FF	FF	FF	;
00000040h:	40	00	00	00	A3	00	00	00	48	00	00	00	FF	FF	FF	FF	;
00000050h:	A4	00	00	00	07	01	00	00	48	00	00	00	FF	FF	FF	FF	;
00000060h:	08	01	00	00	6B	01	00	00	48	00	00	00	FF	FF	FF	FF	;
00000070h:	6C	01	00	00	FF	01	00	00	48	00	00	00	FF	FF	FF	FF	;
00000080h:	00	02	00	00	FF	0F	00	00	00	0E	00	00	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	;

Revision History

V1.0 12/16/2020
Initial release

Appendix

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