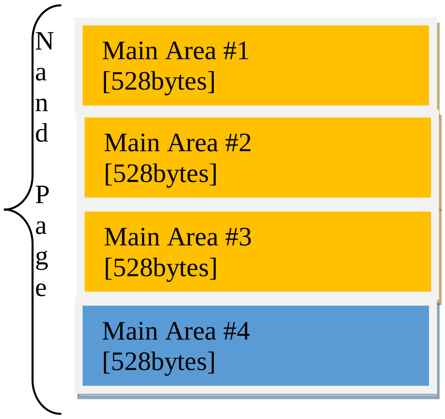
Qualcomm IPQ40XX BCH4 User Manual

General Description and Name

This scheme Implements the skip block method within particular partition, with special page structure and BCH4 ECC calculation.

The special page structure is as follows:



Main Area #1-#3 structure.

Bytes [0-463] Data
Byte [464] – Bad block Marker
Bytes [465-516] Data
Bytes [517-523] – ECC(7bytes)
Bytes [524-527] – user spare[all 0x00]

Main Area #4 structure.

Bytes [0-463] Data
Byte [464] – Bad block Marker
Bytes [465-500] Data
Bytes [501-516] – 0xff
Bytes [517-523] – ECC (7bytes)
Bytes [524-527] – user spare[all 0x00]

Page: 2048 + 64bytes(OOB/spare).
In this case the 64 bytes are spreaded
Throughout the main areas of the page.

Bad block will be considered
by flash factory according to DataSheet
As byte 2048 of the first page of the block
(i.e. byte 464 of the 4th main area).
But note that the controller
& it's driver uses and places the marker at
each main area of the page, so u should
program it as the controller does.

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 = "Qualcomm IPQ40XX BCH4"

Spare area = "ECC"
[Default 'Disabled']

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

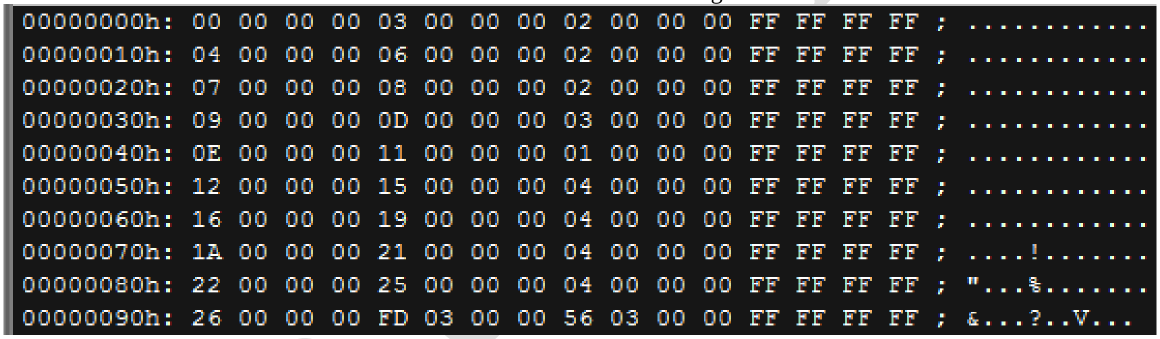
Special Notes

- 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**

Example PartitionTable.mbn:

Start Adr End Adr Actual Data Length Attribute



```
00000000h: 00 00 00 00 03 00 00 00 02 00 00 00 FF FF FF FF ; ..............
00000010h: 04 00 00 00 06 00 00 00 02 00 00 00 FF FF FF FF ; ..............
00000020h: 07 00 00 00 08 00 00 00 02 00 00 00 FF FF FF FF ; ..............
00000030h: 09 00 00 00 0D 00 00 00 03 00 00 00 FF FF FF FF ; ..............
00000040h: 0E 00 00 00 11 00 00 00 01 00 00 00 FF FF FF FF ; ..............
00000050h: 12 00 00 00 15 00 00 00 04 00 00 00 FF FF FF FF ; ..............
00000060h: 16 00 00 00 19 00 00 00 04 00 00 00 FF FF FF FF ; ..............
00000070h: 1A 00 00 00 21 00 00 00 04 00 00 00 FF FF FF FF ; ....!.........
00000080h: 22 00 00 00 25 00 00 00 04 00 00 00 FF FF FF FF ; "...%.........
00000090h: 26 00 00 00 FD 03 00 00 56 03 00 00 FF FF FF FF ; &...?..V...
```

Revision History

V1.0 Date: 2017-07-24
Create this spec.

Appendix

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