

QorIQ IFC NAND Flash Run time code program in MP stage



Background

- Customer asked NXP to provide one procedure to burn run time code into NAND flash by external NAND programmer based on QorIQ IFC interface in Mass Production stage. Due to NAND Flash bad block and ECC issue, it's not reasonable to follow NOR flash procedure to burn whole NAND flash image new NAND flash. We tried to find a way to avoid NAND flash bad block problem and met Mass Production – a lot, fast, workable and easy to burn in NAND flash image.



Procedure 1 - Image preparing in NAND FLASH by u-boot

- Prepares one workable NAND flash run time image without bad block and must reserve ROM space in the NAND flash bottom side.
- Must provide code start address. If there are 3 image codes(like kernel, rootfs and dtb, just provides the first image start address).
- Need to provide BCH format as the project used. QorIQ support Hardware BCH 4 bit ECC, 8 bit ECC and 24 bit ECC, 40 bit ECC. Please refer to QorIQ RM.



Procedure 1 - Example

- For example, if using 256MB, suggest to reserve 30MB in the bottom. This is for bad block if external NAND Flash programmer burn in image, check bad block and got to, record the block and move code to next block.
- Start code address at 0x400000
- Supports BCH 4 bit ECC.
- Refer to T1040RM **24.5.6 Use of ECC algorithms**

And the NAND flash we used, the summary in short as below:

Large page : Page : 2KB + 64 B

Block : 128 KB (64 pages)

BB : at offsets 0 (1 byte = 0xff => ok)

ECC byte : at offsets 8 through A, 0x18~1A, 0x28~2A and 0x38~3A in oob.

IFC 4b-ECC bytes : BB: 0x0 & 0x1; 32 parity bytes at offset 0x8h

OOB:

```
ff ff ff ff ff ff ff // Bad block
aa f2 d3 3d 61 8d 90 00 // the 1st 512 byte
7e cf d0 3e b8 f8 80 00 // the 2nd 512byte
42 fd 8d 2c 44 49 40 00 // the 3rd 512 byte
36 0e f7 1f d1 f6 c0 00 // the 4th 512 byte
ff ff ff ff ff ff ff
ff ff ff ff ff ff ff
ff ff ff ff ff ff ff
```



Procedure 2 - Image generate out in Kernel

- After burn the all run time codes into NAND flash(including kernel, rootfs and dtb), in the run time code, maybe 3-5 MTD partitions for NAND.
- Built alternative image, All NAND uses one MTD partition.
- Loading alternative image through TFTP.
- Perform “nanddump --b=skipbad -o -l \$size -f “file name” /dev/partition x to get all NAND image.



Procedure 2 - Example (1)

- Built alternative image through TFTP, MTD partition as below, Whole 256MB NAND on MTD8

```
root@t1042d4rdb:/var/tmp# cat /proc/mtd
```

```
dev: size erasesize name
```

```
mtd0: 00020000 00020000 "RCW and Reserved space"
```

```
mtd1: 00080000 00020000 "Bootloaderlog"
```

```
mtd2: 00080000 00020000 "Bootloader Log Expansion"
```

```
mtd3: 00080000 00020000 "Regulatory Info"
```

```
mtd4: 00100000 00020000 "NOR Bootloader Expansion"
```

```
mtd5: 00010000 00020000 "NOR FMAN Ucode current bank"
```

```
mtd6: 00020000 00020000 "NOR UBOOT evironment current bank"
```

```
mtd7: 000c0000 00020000 "NOR UBOOT current bank"
```

```
mtd8: 10000000 00020000 "NAND 256MB space for MP"
```



Procedure 2 - Example (2)

- Dump whole 256MB NAND Flash as file in kernel, refer to below:

```
root@t1042d4rdb:/var/tmp# nanddump --b=skipbad -o -l 0x10000000 -f test /dev/mtd8
```

ECC failed: 0

ECC corrected: 0

Number of bad blocks: 4

Number of bbt blocks: 0

Block size 131072, page size 2048, OOB size 64

Dumping data starting at 0x00000000 and ending at 0x10000000...

fm1-gb0 Link is down

ECC: 1 corrected bitflip(s) at offset 0x0a593800



Procedure 2 - Example (3)

- Check the NAND Dump file test.

```
root@t1042d4rdb:/var/tmp# ls -al
```

```
total 305436
```

```
drwxrwxrwt 2 root root    200 Feb  5 07:50 .
```

```
drwxrwxrwt 4 root root    80 Jan  1  2000 ..
```

```
-rw-r--r-- 1 root root    63 Jan  1  2000 firmwrae_version
```

```
-rw-r--r-- 1 root root     2 Jan  1  2000 mac6_cnt
```

```
-rw-r--r-- 1 root root     0 Feb  5 07:49 netserver.debug_2314
```

```
-rw-r--r-- 1 root root 276283392 Feb  5 07:51 test
```

```
-rw-r--r-- 1 root root     2 Jan  1  2000 wifi_done
```

```
-rw-r--r-- 1 root root    30 Jan  1  2000 wifi_mod.txt
```



Procedure 2 - Example (4)

- Provide the whole 256MB NAND flash image you dumped to NAND flash vendor, and run time Code start address and end address, BCH 4 bit ECC.
- Here, everything is ready.





SECURE CONNECTIONS
FOR A SMARTER WORLD