SRC : system reset controller .chapter 42 0x400F8000u

GPR iomux GPR memory map

0x20E\_4000 chapter 26.5

## Detect if it is infinite boot

Pu\_irom\_boot\_zas.c

void pu\_irom\_boot(void)

#define HAPI\_SRC\_SBMR\_INF\_LOOP()\

((hapi\_src\_reg\_volatile.src\_sbmr1 & HAPI\_BT\_INF\_LOOP) >> HAPI\_BT\_INF\_LOOP\_SHIFT)

位与0x80000000后右移31位，检测infinite loop是否拨上。

SRC\_SBMR1寄存器地址：0x20D\_8004

#define HAPI\_SRC\_SBMR2\_DIR\_BT\_DIS()\

((hapi\_src\_reg\_volatile.src\_sbmr2 & SMBR2\_DIR\_BT\_DIS) >> SMBR2\_DIR\_BT\_DIS\_SHIFT)

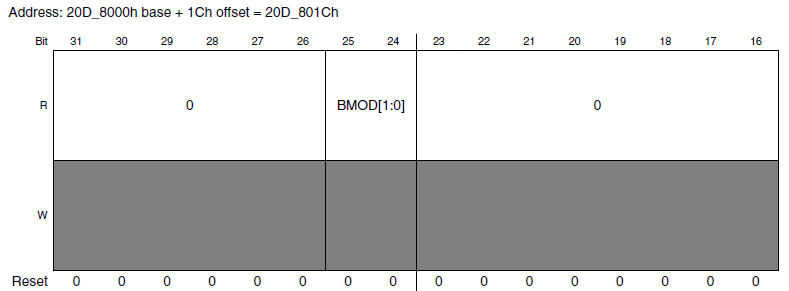
位与0x3，右移三位，检测DIR\_BD\_DIS

## Get boot mode

mcu\_boot = HAPI\_SRC\_get\_bootmode();

#define HAPI\_SRC\_get\_bootmode() ((hapi\_src\_reg\_volatile.src\_sbmr2 & SBMR2\_BMOD\_MASK) >> SBMR2\_BMOD\_SHIFT)

SBMR2 Bit[24：25]决定启动模式



## Check security status

hab\_rvt.report\_status(&hab\_config, NULL);

读取fuse，查看HAB是否使能

## BT\_FUSE\_SEL

If boot\_mode="00" (Development) MCU\_BOOTMODE\_INTERNAL\_FUSE

0=Boot mode configuration is taken from GPIOs.

1=Boot mode configuration is taken from fuses.

If boot\_mode="10" (Production) MCU\_BOOTMODE\_INTERNAL

0 - Boot using Serila Loader (USB)

1- Boot mode configuration is taken from fuses.

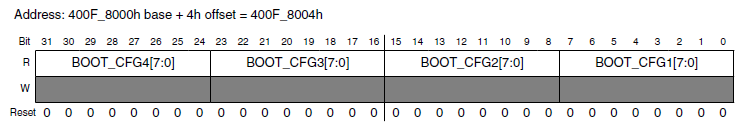
## DIR\_BT\_DIS

0 - Direct boot from external memory is allowed

1 - Direct boot from external memory is not allowed

## Boot device selection

Fuse map reset后会加载到SBMR1中



pu\_irom\_setup\_boot\_selection 🡪 get\_flash\_device

此处会根据SBMR2中的BOOTCFG1也就是0x450，选择boot device，并进行初始化

## Redundant boot

Redundant boot由一个寄存器来控制

get\_persist\_redundant\_boot\_value

srtc\_lpgr\_val = SRC->GPR[9];

redundant\_boot = (srtc\_lpgr\_val & PERSIST\_REDUNDANT\_BOOT\_MASK) >> PERSIST\_REDUNDANT\_BOOT\_SHIFT;

bit[27:26]决定redundant boot的机会，一共3次机会

如果第一份image boot失败，寄存器会+1，然后reset板子后，查看第二份image。第二份image boot失败，寄存器再+1，查看第三份image，以此类推

对nand 来说，支持的redundant boot对应的image存储的位置在FCB中可以设置

semcNandFCB->firmwareTable[0].startPage = FIRMWARE\_COPY0\_START\_PAGE\_INDEX;

semcNandFCB->firmwareTable[0].pagesInFirmware = 10;//pages that the image used

bl\_semc\_nand\_boot\_init（）这个函数会根据

redundant\_boot = (srtc\_lpgr\_val & PERSIST\_REDUNDANT\_BOOT\_MASK) >> PERSIST\_REDUNDANT\_BOOT\_SHIFT;

寄存器中的值，选择FCB中相应的image用来进行boot

## download image

### download initial image：

功能： 初始化device，如果是nand的话，进行数据的读取操作

返回值： 返回true or false

如果 download initial image 返回 true，则对image进行HAB，成功则返回image entry，然后boot

如果download initial image返回false，GoSecondaryBoot（）则设置persist redundant boot+ 1 ，并reset。

Nand flash 的redundant有三次机会。也就是说原始image不算，可以存储三个备份image

这里还有个函数，用于secondary boot。set\_persist\_secondary\_boot\_value()，这个函数将寄存器设置为1，支持secondary boot

# mScale ROM流程：

BootMode ： SRC->SBMR2 直接从GPIO中加载

Boot device ：SRC->SBMR1 BT\_FUSE\_SEL决定是从fuse加载到SBMR1，还是从GPIO加载

If boot\_mode="00" (Development)

0=Boot mode configuration is taken from GPIOs.

1=Boot mode configuration is taken from fuses.

If boot\_mode="10" (Production)

0 - Boot using Serila Loader (USB)

1- Boot mode configuration is taken from fuses.

## Infinite loop

DIR\_BT\_DIS=0 && infinite loop=1

进入infinite loop

## Serial download

Boot mode = serial download 或者boot from fuse && BT\_FUSE\_SEL = 0（从GPIO加载fuse值）

则进入serial download

## Test mode

Boot mode = test mode && （DIR\_BT\_DIS = 0）

进入死循环

## Primary boot、 recovery boot、manufacture boot、redundant boot

Boot mode = internal boot（boot from GPIO） || boot from fuse || DIR\_BT\_DIS =1

条件满足后从相应的device boot，

SD 跟EMMC 支持secondary boot，所谓secondary boot，就是讲boot table指向另一个地方。

Primary boot 失败后，会进入recovery boot

如果板子支持EEPROM\_SUPPORT，则存在recovery boot

Recovery boot失败后，会进入manufacture boot

Manufacture boot就是从SD插槽boot

Redundant boot

mScale 的nand只有一次redundant boot的机会。。

存放两个boot image，若第一份boot成功则忽略第二份，如果失败，boot 第二份image

# Uboot使用方法

## SD \ EMMC 操作

mmc list : 列出当前SD跟EMMC

mmc dev 0x0 0x0 :选中设备0, 分区0

mmc erase 0x0 0x02000 : 擦除选中设备分区的0 block开始的0x2000个block，一个block = 0x200

mmc read 0x00800000 0x2 0x400 : 读取0x2 block开始的数据到0x00800000中

loadx 点击传送-》X mode将image通过UART传送到DDR中

md 0x4008000 10 ： 显示0x82000000 开始的10个word长度的数据

mmc write 0x40080000 0x2 0x300 :将0x82000000处长度为0x300 block的数据写入选中设备的0x2 block处

## Fuse 操作

fuse read 0 1 ：读取bank 0 word1的数据

fuse prog 0 1 00000001 ： 向bank 0 word 1 处写入0x00000001

bank 以及word可以通过fuse map查看

# secondary boot

SD跟EMMC支持secondary boot

typedef struct

{

UINT32 chipNum; /\* Chip Select, ROM does not use it \*/

UINT32 driveType; /\* Always system drive, ROM does not use it \*/

UINT32 tag; /\* Tag, must be 0x584D2E69 (i.MX) \*/

UINT32 firstSectorNumber; /\* Block address of secondary image, block size is 512B \*/

UINT32 sectorCount; /\* Not used by ROM \*/

} SECONDARY\_IMG\_TBL\_T;

结构体里面的数据是小端模式。

SD\EMMC 一个block的长度为0x200

Tag ： 0x00112233

FirstSectorNumber ：

EMMC的image start address = 0x8000 IVT offset = 0x400

Table address = 0x200

设定secondary start address为0x9000，则firstSectorNumber为：

(0x9000-0x8000) /0x200 = 8

将制作好的table烧写到0x8200， image烧写到0x9000.reset板子后persist register变为1，secondary image就boot成功了

mScale的SPL image的数据从IVT开始，所以image需要烧写到0x9400处

mmc dev 0x0 0x0

mmc erase 0x0 0x2000

//烧写image

mmc write 0x40080000 0x4a 0x2000

烧写table

mmc write 0x40080000 0x41 0x1

# SDP host

sdphost.exe -u 0x1fc9,0x012b -- error-status

sdphost.exe -u 0x1fc9,0x012b -- write-file 0x800400 ..\demo\SPL

sdphost.exe -u 0x1fc9,0x012b -- jump-address 0x800400

# EMMC fast boot

Fast boot, 8 bit ddr mode

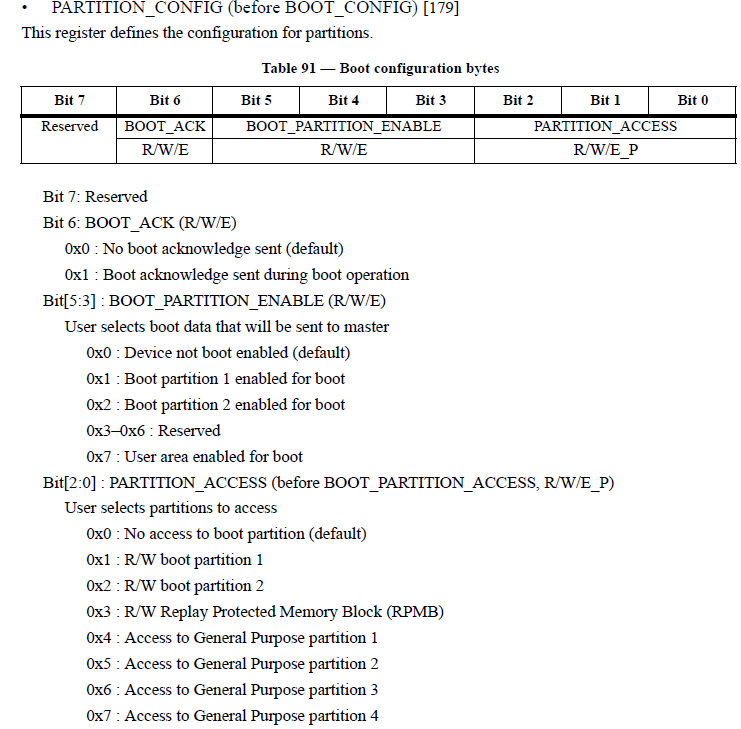
Fuse prog 1 3 100000e0

Validation板 EMMC 对应dev 0 ， usdhc 0

Boot\_ack对应fuse 0x4A0 bit[0] bank 2 word 2

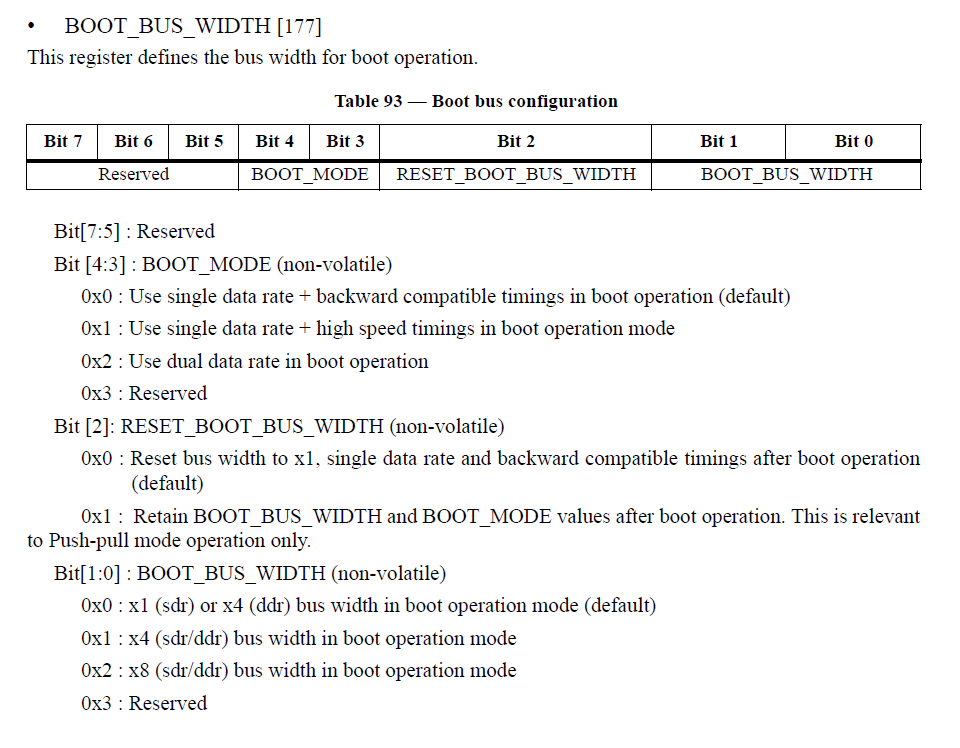
mmc partconf dev boot\_ack boot\_partition partition\_access

mmc partconf 0 1 7 0



mmc bootbus dev boot\_bus\_width reset\_boot\_bus\_width boot\_mode

mmc bootbus 0 2 0 2



# pu\_irom\_log\_buffer

# serial download

C:\Users\b57252\Desktop\Doc\NewFeatures\ROM\M580D\sdphost>sdphost.exe -u 0x1fc9

,0x012b -- write-file 0x800000 C:\Users\b57252\Desktop\Doc\NewFeatures\ROM\M580D

\demo\spl\_signed.bin

C:\Users\b57252\Desktop\Doc\NewFeatures\ROM\M580D\sdphost>sdphost.exe -u 0x1fc9

,0x012b -- jump-address 0x800400

# Test Cases

## EMMC boot

### regular boot, 1bit, normal speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002000

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x8400/0x200=0x42)

mmc write 0x40080000 0x42 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

### regular boot, 8bit, normal speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002020

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x8400/0x200=0x42)

mmc write 0x40080000 0x42 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

### regular boot, 8bit ddr, normal speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002060

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x8400/0x200=0x42)

mmc write 0x40080000 0x42 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

### regular boot, 1 bit, high speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002004

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x8400/0x200=0x42)

mmc write 0x40080000 0x42 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

### regular boot, 8 bit, high speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002024

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x8400/0x200=0x42)

mmc write 0x40080000 0x42 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

### regular boot, 8 bit ddr, high speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002064

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x8400/0x200=0x42)

mmc write 0x40080000 0x42 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

### fast boot, 8 bit, normal speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 100020a0

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x400/0x200=0x2)

mmc write 0x40080000 0x2 0x200

6. configure EMMC to meet the fuse setting

mmc partconf 0 0 7 0

mmc bootbus 0 2 0 0

7. set boot mode from fuse and re-power the board, demo boot successfully.

### fast boot, 8 bit ddr, normal speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 100020e0

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x400/0x200=0x2)

mmc write 0x40080000 0x2 0x200

6. configure EMMC to meet the fuse setting

mmc partconf 0 0 7 0

mmc bootbus 0 2 0 2

7. set boot mode from fuse and re-power the board, demo boot successfully.

### fast boot, 8 bit ddr, high speed

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 100020e4

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x400/0x200=0x2)

mmc write 0x40080000 0x2 0x200

6. configure EMMC to meet the fuse setting

mmc partconf 0 0 7 0

mmc bootbus 0 2 0 2

7. set boot mode from fuse and re-power the board, demo boot successfully.

### fast boot, 8 bit ddr, high speed, acknowledge open

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 100020e4

fuse prog –y 2 2 1

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x400 (demo is start from IVT so we must write the demo to 0x8400 , block = 0x400/0x200=0x2)

mmc write 0x40080000 0x2 0x200

6. configure EMMC to meet the fuse setting

mmc partconf 0 1 7 0

mmc bootbus 0 2 0 2

7. set boot mode from fuse and re-power the board, demo boot successfully.

### Secondary boot

1. using Uboot to configure the boot fuse

fuse prog –y 1 3 10002000

2. choose EMMC, erase the first 0x200 block(1 block = 0x200 bytes.)

mmc dev 0x0 0x0

mmc erase 0x0 0x200

3. open x mode for transmitting the demo

loadx

4. using UART to transmit secondary boot demo to DDR in x mode.

click transmit -> transmit file (using x mode) -> choose demo ->click transmit

5. write demo from DDR to EMMC 0x8400 (demo is start from 0x8200, block = 0x8200/0x200=0x41)

mmc write 0x40080000 0x41 0x200

6. set boot mode from fuse and re-power the board, demo boot successfully.

## HAB(SD boot)

1. using Uboot to configure the boot fuse from SD and set HAB in close mode

fuse prog –y 1 3 10001400

fuse prog –y 1 1 20000

2. burn SRK and OTPMK

Burn the SRK fuse

fuse prog -y 6 0 A863EBC2

fuse prog -y 6 1 AFA98ABA

fuse prog -y 6 2 3B528055

fuse prog -y 6 3 CB816E31

fuse prog -y 7 0 21EF6EF0

fuse prog -y 7 1 BF0E8D24

fuse prog -y 7 2 1E17181E

fuse prog -y 7 3 415F3075

Burn OTPMK

fuse prog -y 4 0 ED53D1C3

fuse prog -y 4 1 98A92EFD

fuse prog -y 4 2 50EF2622

fuse prog -y 4 3 A0B9D347

fuse prog -y 5 0 8E13C7B9

fuse prog -y 5 1 84027CA8

fuse prog -y 5 2 2C8CD201

fuse prog -y 5 3 A29A0B2C

3. using linux to burn signed demo to SD

dd if=spl\_signed.bin of=/dev/sdb bs=32768 seek=1

4. set boot mode from fuse and re-power the board, demo boot successfully.

## SD boot

## Manufacture boot

1. set boot mode as serial download, insert the SD card with demo which can boot successfully

2. re-power the board, demo boot successfully

## Serial download

1. set boot mode as serial download

2. insert the USB cable to board and open a cmd window

3. using SDP host to write demo to 0x800400

sdphost.exe  -u 0x1fc9,0x012b -- write-file 0x800400 SPL

4. execute the demo, it will work successfully

Sdphost.exe –u 0x1fc9,0x012b – jump-address 0x800400

## Serial download (HAB mode)

1. set boot mode as serial download and prepare an board which can pass HAB

2. insert the USB cable to board and open a cmd window

3. using SDP host to write demo to 0x800000

sdphost.exe  -u 0x1fc9,0x012b -- write-file 0x800000 spl\_signed.bin

4. execute the demo, it will work successfully

Sdphost.exe -u 0x1fc9,0x012b – jump-address 0x800400