I.MX8MM SDCARD SECONDARY BOOT DEMO

BIYONG SUN
AUG 3, 2022

SECURE CONNECTIONS
FOR A SMARTER WORLD

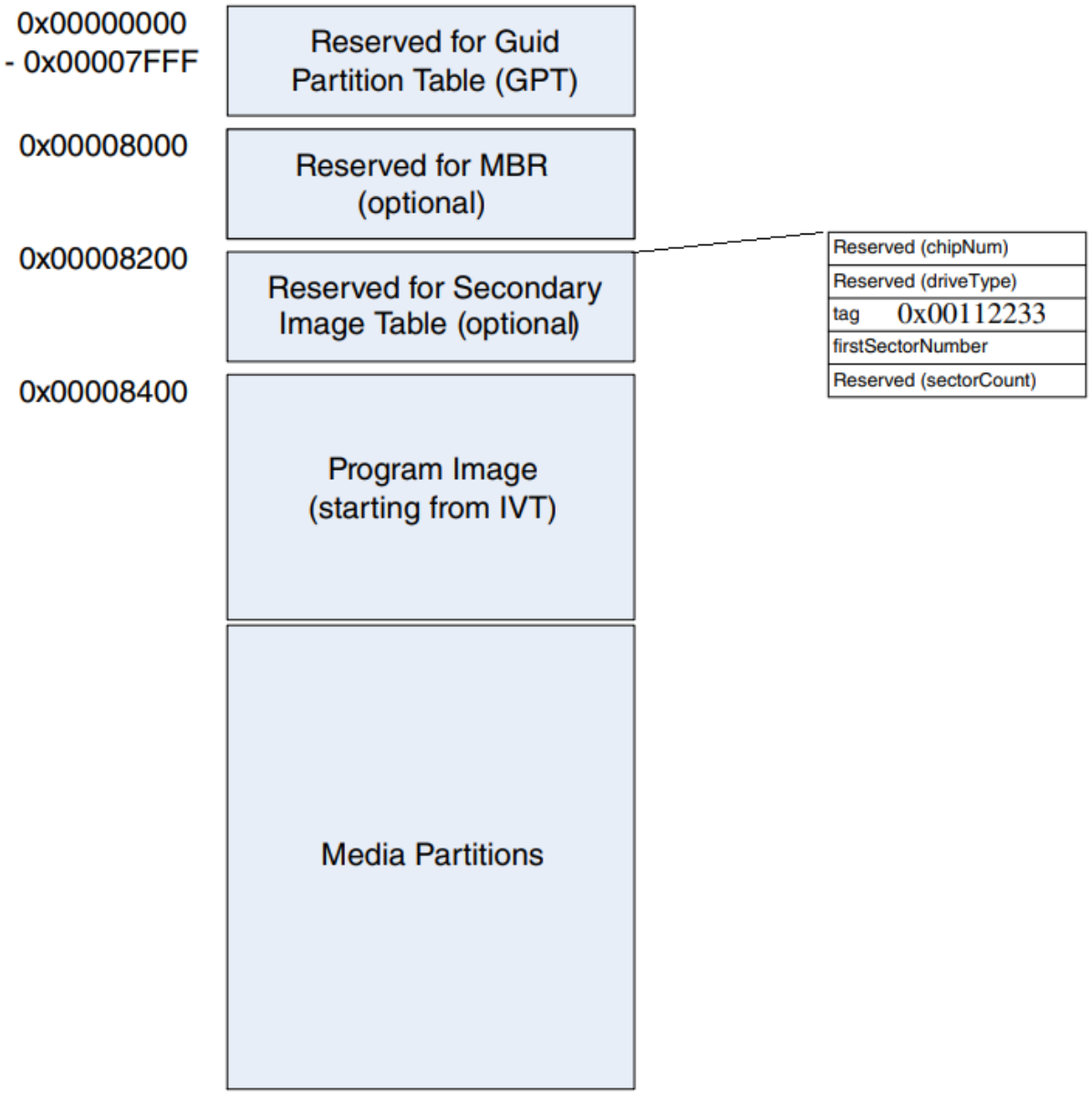
Boot Image Layout

The primary boot image is at 0x8400(33k).
The Secondary Image table is at 0x8200.

The tag indicates having secondary image.
The firstSectorNumber tells the offset.

When the primary boot fail, will try secondary boot.

The behavior is exactly the same.



Difference in BSP between i.MX6 and i.MX8MM

i.MX6 uses u-boot only as bootloader.
i.MX8MM uses spl first, then spl will load u-boot.

That makes currently need to build two different u-boot for secondary boot image.
Because the primary boot image and secondary boot image start on different sector.

For i.MX6, the ROM loads the entire u-boot, it has no such issue.

Difference in BSP between i.MX6 and i.MX8MM(Cont.)

If you use the i.MX8MM primary image u-boot as the secondary image u-boot.
You will find the following error:

mmc_load_image_raw_sector: mmc block read error
SPL: failed to boot from all boot devices
ERROR ### Please RESET the board

In the test, we uses all zero as a fake primary image.
Without code change, the spl still try to read from the primary image sector offset.

i.MX8MM Primary Image and Secondary Image

#define CONFIG_SYS_MMCSD_RAW_MODE_U_BOOT_SECTOR (0x300 + CONFIG_SECONDARY_BOOT_SECTOR_OFFSET)

By default, for primary u-boot, the CONFIG_SECONDARY_BOOT_SECTOR_OFFSET is undefined.
You need to define the CONFIG_SECONDARY_BOOT_SECTOR_OFFSET for the secondary u-boot.

All those are hard coding.

So, under current BSP, if you want to use secondary boot, use need to build primary image first.
Then you have the primary image size and define the CONFIG_SECONDARY_BOOT_SECTOR_OFFSET
Build secondary image and combine them together.

My approach

SRC_GRP10[30] to know it is primary boot or secondary boot.
0x0090E6B4 to know the secondary image firstSectorNumber

Table 6-8. Persistent bits

Bit name	Bit location	Description
PERSIST_SECONDARY_BOOT	SRC_GPR10[30]	This bit identifies which image must be used—primary and secondary. Used only for the boot modes that support redundant boot.
PERSIST_BLOCK_REWRITE	SRC_GPR10[29]	This bit is used as a warning. It identifies that there are errors in the NAND blocks that hold the application image.

Reserved (chipNum)
Reserved (driveType)
tag 0x00112233
firstSectorNumber
Reserved (sectorCount)

```
0090e6a0: 00000000 00000000 00000000 00000000
0090e6b0: 00112233 00001b18 00000000 00000000
0090e6c0: 00000000 00000000 00000000 00000000
```

Patch

0001-secondary_boot-spl-auto_calc-boot_sector-lf-5.10.72-2.2.0.patch

```
--- u-boot/board/freescale/imx8mm_evk/spl.c.orig		2022-03-09 15:08:43.000000000 +0800
+++ u-boot/board/freescale/imx8mm_evk/spl.c		2022-08-02 14:23:36.068014364 +0800
@@ -326,3 +326,30 @@ void board_init_f(ulong dummy)

		board_init_r(NULL, 0);
 }
+
+#define FIRST_SECTOR_NUM	(*(volatile unsigned int  *)0x0090E6B4)
+#define SRC_GPR10_PERSIST_SECONDARY_BOOT	(0x40000000)
+unsigned long spl_mmc_get_uboot_raw_sector(struct mmc *mmc)
+{
+
+
+		u32 boot_dev = spl_boot_device();
+
+		struct src *src_regs = (struct src *)SRC_BASE_ADDR;
+		u32 src_gpr10_val;
+		u32 first_sector_num = FIRST_SECTOR_NUM;
+
+		src_gpr10_val = readl(&src_regs->gpr10);
+
+		if(src_gpr10_val & SRC_GPR10_PERSIST_SECONDARY_BOOT){
+
+			switch (boot_dev) {
+				case BOOT_DEVICE_MMC1:
+				case BOOT_DEVICE_MMC2:
+					return
CONFIG_SYS_MMCSD_RAW_MODE_U_BOOT_SECTOR + first_sector_num;
+			}
+		}
+
+		return CONFIG_SYS_MMCSD_RAW_MODE_U_BOOT_SECTOR;
+}
```

imx_sd_secondary_boot_creator.sh

./imx_sd_secondary_boot_creator.sh <soc_name> <primary image> <secondary image>
soc_name: imx6(the entire imx6 family), imx8mq, imx8mm

Use the following command to try only secondary boot image(fake primary damaged)
./imx_sd_secondary_boot_creator.sh zero.bin flash.bin

You will find following image in out directory:
secondary_boot-zero.bin-flash.bin.imx --- Image layout from 0x8200
secondary_boot_dummy_pt-zero.bin-flash.bin.sdcard --- sdcard mirror, no partition table

SECURE CONNECTIONS
FOR A SMARTER WORLD