AN14093

利用Falcon模式和内核优化，实现i.MX 8M和i.MX 9的快速启动

第2.0版—2024年9月26日 **应用笔记**

文档信息

信息	内容
关键词	AN14093、Falcon模式、内核优化、U-Boot优化、快速启动、Falcon启动、i.MX 93、i.MX 95、i.MX 8M、Linux
摘要	本文档指导如何缩短i.MX 8M和i.MX 9系列产品的启动时间。

1 介绍

本文档指导用户如何缩短以下芯片的Linux启动时间：

- i.MX 8M系列（i.MX 8M Mini LPDDR4 EVK、i.MX 8M Nano LPDDR4 EVK和i.MX 8M Plus LPDDR4 EVK）
- i.MX 9系列（i.MX 93 LPDDR4 EVK和i.MX 95 LPDDR5EVK）

本文档的目标包括：

- 引导加载程序的优化
- Linux内核与用户空间的优化
- 对所有平台上默认启动时间与优化后启动时间的比较

1.1 软件环境

假设使用的是Ubuntu 20.04 PC。

优化过程中使用Yocto Project BSP Scarthgap版本和Linux BSP版本6.6.23_2.0.0。

硬件设置和设备：

- 恩智浦i.MX 8MM EVK LPDDR4开发套件
- 恩智浦i.MX 8MN EVK LPDDR4开发套件
- 恩智浦i.MX 8MP EVK LPDDR4开发套件
- 为11x11 mm LPDDR4内存专门设计的恩智浦i.MX 93 EVK开发套件
- 为19x19 mm LPDDR5内存专门设计的恩智浦i.MX 95 EVK开发套件
- Micro SD卡：本次实验选用了容量为32 GB的SanDisk Ultra Micro SDHC I级10速的存储卡。
- Micro-USB (i.MX 8M)或Type-C (i.MX 9)数据线用于调试端口。

2 综述

本节概述了为缩短启动时间所需进行的常规修改。

2.1 缩短引导加载时间

为了减少引导加载时间，可选择以下两种方法之一：

- **取消启动延迟**：与默认配置相比，可节省约两秒钟，同时所需的更改最小。在启动过程中，U-Boot不再等待按键输入。
- **实施Falcon模式**：与默认配置相比，可节省约四秒钟。它使得二级程序加载器（SPL，U-Boot的一部分）能够直接加载内核，跳过完整U-Boot的加载过程。

2.2 缩短Linux内核启动时间

为了减少Linux内核启动时间，可可选择以下两种方法之一：

- **减少控制台消息**：可节省约三秒钟。在内核命令行中添加`quiet`参数。
- **通过删除驱动程序和文件系统精简内核**：默认情况下，内核映像包含许多驱动程序和文件系统（例如：UBIFS），以启用板卡支持的大部分功能。您可以根据使用情况，裁剪包含的驱动程序和文件系统列表。

2.3 缩短用户空间启动时间

为了缩短用户空间启动时间，可尝试以下方法：

- **在systemd之前启动应用程序：**可节省约600毫秒。尽快启动所需进程，同时考虑其依赖关系。

2.4 测量

测量范围涵盖了从板载POR（上电复位）到INIT过程启动的整个阶段。

具体的测量设置详见《i.MX 9系列 – 启动时间测量方法》（AN14205文档）。

表1列出了测量的间隔：

表1. 测量的间隔

时间点	脉冲间的间隔	脉冲位置	启动阶段	
BootROM	nRST -> before ddr_init()	board/freescale/<board>/spl.c/board_init_ f()	**SPL**	Timeline
DDR要	before ddr_init() -> after ddr_init()	board/freescale/<board>/spl.c/board_init_ f()		
SPL初始化+加载U-Boot映像	after ddr_init() -> before image_entry()	common/spl/spl.c/jump_to_image_no_args()		
U-Boot初始化(init_sequence_f)	before image_entry() -> start init_sequence_r	common/board_r.c/board_init_r()	**U-BOOT**	
U-Boot初始化(init_sequence_r)	start init_sequence_r -> u-boot main_loop	common/main.c		
启动序列	u-boot main_loop -> before load_image	include/configs/<board>.h		
内核映像加载	before loadimage -> after loadimage	include/configs/<board>.h		
内核启动到INIT的过程	after loadimage -> /sbin/init	get the timestamp during kernel boot	**Kernel**	

3 引导加载程序优化

本章包含以下内容：

3.1 默认启动模式

图1描述了默认的启动顺序。在上电或重置后，这些板卡执行存储在其只读存储器（ROM）中的**BootROM**（主程序加载器）。

BootROM负责配置片上系统（SoC），它会进行基础外设的初始化，例如设置锁相环（PLL）、调整时钟配置、初始化内存（SRAM）。然后，它会找到一个启动设备，并从中加载引导加载程序映像，这个映像可能包含U-Boot SPL、Arm Trusted Firmware(ATF)、U-Boot等多个组件。

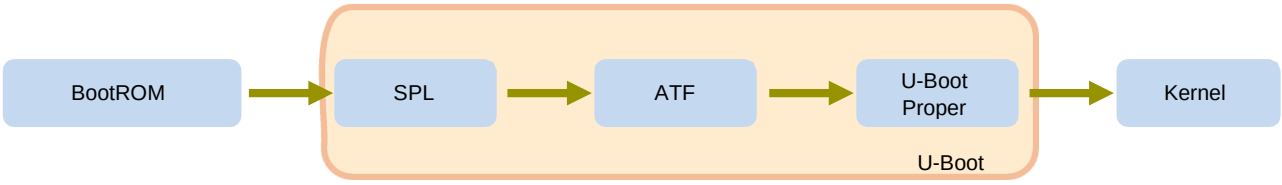

图1. 默认启动顺序

由于典型的U-Boot映像无法完全放入内部SRAM中，因此被分为两部分：**SPL**和**U-Boot本身**。

SPL是引导加载程序的第一阶段，是一个较小的预加载器，与U-Boot共享相同的源代码，但只包含一套能够适应SRAM的最小代码集。SPL被加载到SRAM中。它负责配置并初始化一些外设，尤其是动态随机存储器（DRAM）。之后，SPL会将ATF和U-Boot本身加载到DRAM。最终，SPL会跳转到ATF，而ATF随后会跳转到U-Boot本身。

ATF是i.MX8/9系列新近加入的组件，它为Armv8架构提供了可信赖的参考代码基础。它实现了各种ARM接口标准，包括电源状态协调接口（PSCI）。这个二进制文件通常包含在引导加载程序的二进制文件中，在U-Boot的早期阶段启动。如果没有ATF，内核无法设置必须在安全世界环境中执行的服务。

U-Boot本身是引导加载程序的第二阶段。它提供了一种灵活的方式来加载和启动Linux内核。同时，它还提供了一套简易工具，允许用户通过命令行界面与此板的硬件进行交互。U-Boot本身从DRAM运行，并初始化额外的硬件设备，如网络、USB和DSI/CSI等。然后，它加载并准备好扁平化设备树（FDT）。U-Boot的主要任务是加载和启动内核映像。

Linux内核从DRAM运行，并完全接管系统。从此刻起，U-Boot不再对系统有任何控制权。

3.2 Falcon模式

Falcon模式是U-Boot中的一项功能，它允许SPL直接启动Linux内核，实现快速启动。这种模式完全跳过了U-Boot的加载和初始化过程，显著缩短了引导加载程序所需的时间。

图2展示了Falcon模式的启动顺序。

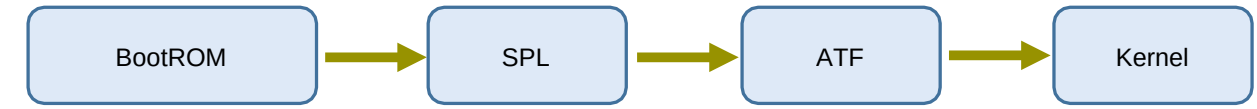

图2. Falcon模式的启动顺序

要实现这个模式，需要执行以下操作：

- 激活Falcon模式所需的一些特定配置。
- 提前准备好FDT。
- 配置ATF以跳转至内核。
- 生成FIT/容器映像，包含ATF和内核映像。

3.2.1 Falcon模式的启用

为了启用Falcon模式，请按照以下步骤操作：

1. 要创建Yocto工程，请按照《i.MX Yocto工程用户指南》（文档IMXLXYOCTOUG）中的**第3节**和**第4节**进行操作。这将在`~/imx-yocto-bsp`目录中初始化Yocto环境。

2. 下载快速启动Yocto层：

```
$ cd ~/imx-yocto-bsp/sources
$ git clone https://github.com/nxp-imx-support/meta-imx-fastboot.git
```

3. 设置构建目录：

```
$ DISTRO=fsl-imx-wayland MACHINE=<machine_name> source sources/meta-imx-
fastboot/tools/imx-setup-fastboot.sh -b <build_dir>
```

其中<machine_name>设置为：
- 对于i.MX 8M Nano，为imx8mnevk-fastboot
- 对于i.MX 8M Mini，为imx8mmevk-fastboot
- 对于i.MX 8M Plus，为imx8mpevk-fastboot
- 对于i.MX 93，为imx93evk-fastboot
- 对于i.MX 95，为imx95-19x19-lpddr5-evk-fastboot

4. 构建映像：

```
$ bitbake imx-image-fastboot
```

5. 将生成的imx-image-fastboot-<machine_name>.rootfs.wic.zst映像写入SD卡。此映像位于~/imx-yocto-bsp/<build_dir>/deploy/images/<machine_name>目录中。
6. 启动此板并在U-Boot中停止它。
7. [仅限i.MX 95]在U-Boot中运行以下命令，可启用DDR Quickboot：

```
u-boot=> qb save
```

这将保存经过训练的DDR参数，从而将DDR初始化时间缩短约4秒。

8. 准备好扁平化设备树，并使用以下命令将其写入SD卡：

```
u-boot=> run prepare_fdt
```

在Falcon模式下启动时，准备设备树是一个关键步骤。通常，U-Boot会在启动Linux时进行FDT修复。这意味着，U-Boot会在初始设备树中添加内核参数和内存节点以及其他修改。
这些参数可以在配置文件中找到：uboot-imx/configs/<board>_evk.h，名称为bootargs，其中<board>为：imx8mm、imx8mn、imx8mp、imx93或imx95。它们指定控制台参数，并告诉内核在哪里可以找到根文件系统。

9. 每次启动此主板时，它都会自动以Falcon模式启动。您可以在开机过程中按住任意键返回U-Boot模式。

3.2.2 Falcon模式的实现细节

第3.2.1节“Falcon模式的启用”中所述的步骤通过修改U-Boot、ATF和mkimage工具来启用Falcon模式。本节会介绍后台发生的情况。

1. **U-Boot补丁：**recipes-bsp/u-boot/files/0001-Enable-Fast-Boot-on-i.MX-8M-Family-and-i.MX-93.patch
   - 为每个平台（i.MX 8M和i.MX 9）创建Falcon配置文件，这些文件存放在uboot-imx/configs/目录下，命名为<board>_falcon_defconfig。这些配置文件基于默认配置文件<board>_defconfig，并添加了如下Falcon支持：
     启用的参数[设为y]：
     CONFIG_SPL_SERIAL
     CONFIG_CMD_SPL — 在U-Boot中启用spl export命令，用于生成Falcon的FDT
     CONFIG_SPL_MMC — 允许SPL使用SPL MMC API从MMC读取数据
     CONFIG_MMC_BROKEN_CD **[仅限i.MX 93]**

CONFIG_SPL_LOAD_FIT**[仅限i.MX 8M]**
CONFIG_SPL_FS_FAT— 允许SPL从FAT分区读取数据
CONFIG_FIT **[仅限i.MX 8M]**
CONFIG_SPL_OS_BOOT — 激活Falcon模式
CONFIG_SPL_MMC_IO_VOLTAGE和CONFIG_SPL_MMC_UHC_SUPPORT — 为SPL启用MMC高速传输功能，减少内核映像的加载时间（***对于i.MX 8MM不适用**，因为由于OCRAM大小限制，不支持SPL DM）
CONFIG_LTO（在某些平台上，此功能默认启用）— 通过添加链接时优化来减小二进制文件大小。为确保带有FAT支持的SPL映像适合，此项为必选。
禁用的参数[设为n]：
CONFIG_SPL_BOOTROM_SUPPORT
设置参数：
CONFIG_SYS_SPL_ARGS_ADDR
其中：
– 对于i.MX 8MN、i.MX 8MM和i.MX 8MP，设置为0x43000000
– 对于i.MX 93，设置为0x83000000
– 对于i.MX 95，设置为0x93000000
CONFIG_SPL_FS_LOAD_PAYLOAD_NAME：
– 对于i.MX 8M，为u-boot-atf.itb
– 对于i.MX 9，为u-boot-atf-container.img
CONFIG_SPL_FS_LOAD_KERNEL_NAME：
– 对于i.MX 8M，为kernel-atf.itb
– 对于i.MX 9，为kernel-atf-container.img
CONFIG_SPL_FS_LOAD_ARGS_NAME：
– 对于i.MX 8MM，为imx8mm-evk-falcon.dtb
– 对于i.MX 8MN，为imx8mn-evk-falcon.dtb
– 对于i.MX 8MP，为imx8mp-evk-falcon.dtb
– 对于i.MX 93，为imx93-11x11-evk-falcon.dtb
– 对于i.MX 95，为imx95-19x19-evk-falcon.dtb
CONFIG_CMD_SPL_WRITE_SIZE设置为0x20000
CONFIG_SYS_BOOTM_LEN设置为0x2300000
– 对于i.MX 8MN和i.MX 8MP，增加了当前长度以适应内核映像的大小。
CONFIG_SPL_CUSTOM_SYS_MALLOC_ADDR设置为0x42600000 **[仅限i.MX8M*]**
– 内核映像太大，将覆盖SPL malloc空间。增加了i.MX 8M系列的起始地址。

- 实现了spl_start_uboot()函数，位于uboot-imx/board/freescale/<board>/spl.c，其中<board>为imx8mm_evk、imx8mn_evk、imx8mp_evk、imx93_evk或imx95_evk。此函数负责检查SPL是应该启动内核还是U-Boot。如果在启动过程中按下了任意键，则函数返回**1**，这意味着必须启动U-Boot。否则，SPL应该启动内核。
- 实现了spl_fit_read()函数，位于arch/arm/mach-imx/imx9/scmi/soc.c文件。由于USDHC 控制器是非安全主控器，因此无法访问DDR安全区域。此函数仅对i.MX 95是必选的，它负责将容器映像从SD加载到DDR。
- 重置i.MX 8M 系列的以太网PHY。为了在操作状态中使以太网MAC能与PHY交互，在启动内核前，必须重置此PHY。PHY的重置操作在board_init_r()函数中完成，该函数位于uboot-imx/common/spl/spl.c文件中。

- 在include/configs/<board>.h文件中，新建两个U-Boot变量：prepare_fdt和mmcargs_fastboot。prepare_fdt变量生成并保存固定的FDT。prepare_fdt使用mmcargs_fastboot设置内核参数。

2. **ATF补丁**：recipes-bsp/imx-atf/files/0001-Add-support-to-jump-to-Kernel-directly-from-ATF.patch
   - 增加了对直接跳转到内核的支持。由于ATF不支持在恩智浦平台上直接跳转到内核，因此必须在bl31_early_platform_setup2()函数中将FDT地址作为参数传递，该函数位于imx-atf/plat/imx/imx8m/<board>/<board>_bl31_setup.c路径下，适用于i.MX 8M系列，以及位于imx-atf/plat/imx/<board>/<board>_bl31_setup.c路径下，适用于i.MX 9系列。
3. **mkimage补丁**：recipes-bsp/imx-mkimage/files/0001-Add-scripts-for-Fast-Boot-implementation-for-i.MX8M-.patch
   - 对于i.MX 8M，为U-Boot FIT映像源(u-boot.its)的uboot-1节点添加了“os”属性。当启用Falcon模式时，如果spl_start_uboot()返回**1**)，则加载U-Boot时需要这一属性。否则，U-Boot无法启动。
   - 在soc.mak文件中创建两个新目标，以生成：
     – 包含ATF和内核的映像。
     – uImage，它是FDT准备过程中spl export命令使用的。

3.2.3 内存映射

本节介绍了Falcon模式期间的内存映射。

Boot ROM加载SPL，SPL在片上RAM（OCRAM-内部处理器内存）中运行。SPL负责初始化动态RAM（DDR），将ATF加载到OCRAM，然后将内核设备树和内核映像加载到DDR。SPL在DDR中预留了一块内存空间用于动态内存分配（malloc），这部分内存在SPL运行期间是不允许被覆盖的。

图3展示了以i.MX93为例的内存映射。

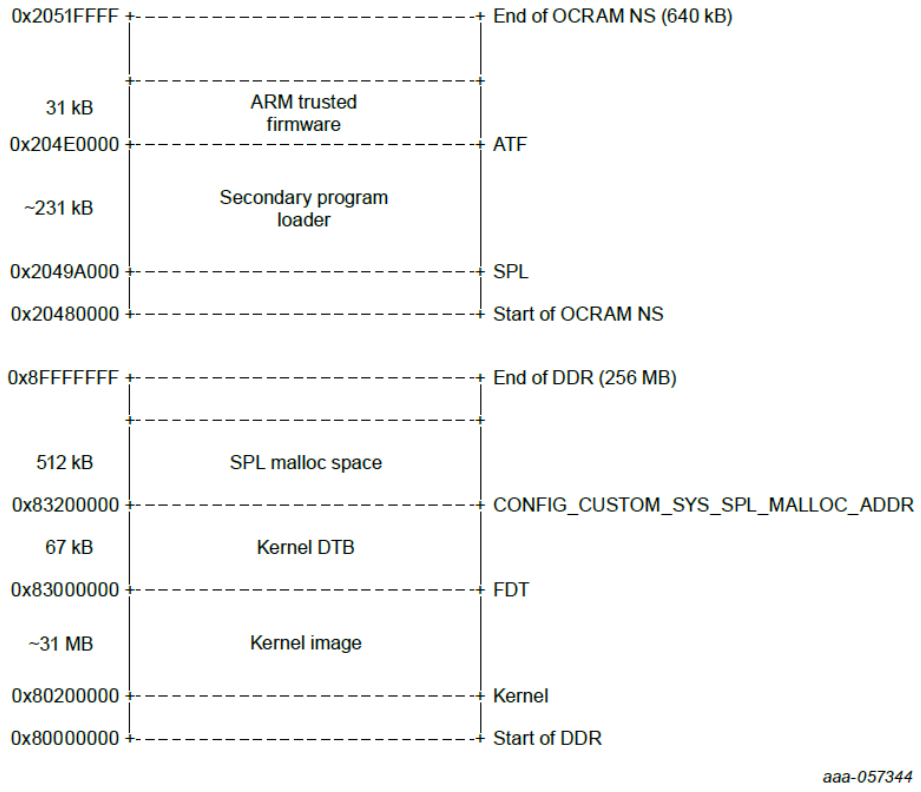

图3. 在Falcon模式期间的内存映射

对于其他平台，只有地址不同，如表2所列。

表2. 内存地址

平台	SPL	ATF	内核映像	内核DTB
i.MX 95	0x4aa00000	0x8A200000	0x90200000	0x93000000
i.MX 8M Mini	0x007e1000	0x00920000	0x40200000	0x43000000
i.MX 8M Nano	0x00912000	0x00960000	0x40200000	0x43000000
i.MX 8M Plus	0x00920000	0x00970000	0x40200000	0x43000000

3.2.4 Falcon模式期间的函数调用

以下是SPL Falcon模式期间调用的重要函数，以供参考。

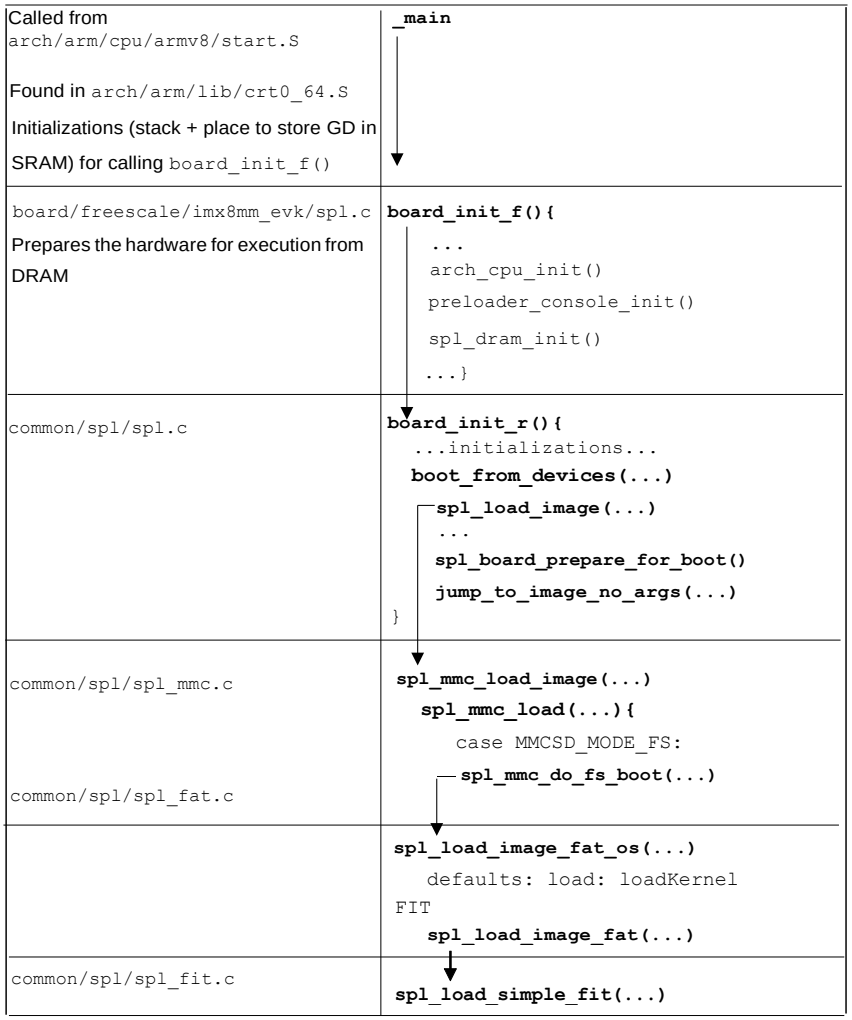

图4. SPL Falcon模式期间调用的函数

4 内核空间优化

本章节包括以下信息。

- 第4.1节 “添加quiet”
- 第4.2节 “删除不必要的驱动程序和文件系统”

4.1 添加quiet

为了将内核启动时间缩短大约一半，建议在内核bootargs中使用quiet参数。这样做可以在Linux启动过程中抑制调试消息的输出。在第3.2.1节 “Falcon模式的启用” 中构建的Yocto映像中，已经使用mmcargs_fastboot变量添加了此参数。

4.2 删除不必要的驱动程序和文件系统

根据您的具体需求，您可以通过删除不必要的驱动程序和文件系统来简化内核。利用**bootgraph**，您可能希望在启动时首先分析内核时序。

要创建bootgraph，请按以下步骤操作：

1. 在内核bootargs中添加initcall_debug。
   a. 要进入U-Boot，启动时按住任意键。
   b. 添加initcall_debug，编辑mmcargs_fastboot参数。

```
u-boot=> edit mmcargs_fastboot
edit: setenv bootargs ${jh_clk} console=${console} root=${mmcroot} quiet
 initcall_debug
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
```

2. 重新生成fastboot设备树，以使用新的bootargs：

```
u-boot=> run prepare_fdt
u-boot=> reset
```

3. 启动板卡并获取内核日志。

```
root@imx8mn-lpddr4-evk-fastboot:~# dmesg > boot.log
```

*boot.log*文件记录类似以下日志的信息，可以通过这些数据来分析内核启动过程中每个函数的耗时。

```
[2.583922] initcall deferred_probe_initcall+0x0/0xb8 returned 0 after 895357
[2.583955] calling  genpd_power_off_unused+0x0/0x98 @ 1
[2.583977] initcall genpd_power_off_unused+0x0/0x98 returned 0 after 12 usec
[2.583984] calling  genpd_debug_init+0x0/0x90 @ 1
[2.584312] initcall genpd_debug_init+0x0/0x90 returned 0 after 321 usecs
[2.584333] calling  ubi_init+0x0/0x23c @ 1
[2.584627] initcall ubi_init+0x0/0x23c returned 0 after 286 usecs
```

4. 将生成的boot.log文件复制到主机PC上。回到主机PC，使用以下命令创建图表。

```
$ cd <build_dir>/tmp/work/imx8mnevk_fastboot-poky-linux/linux-imx/<git_rev>/
git/scripts
$ ./bootgraph.pl boot.log > boot.svg
```

您可以得到类似图5的结果，并分析内核启动时间的使用情况：

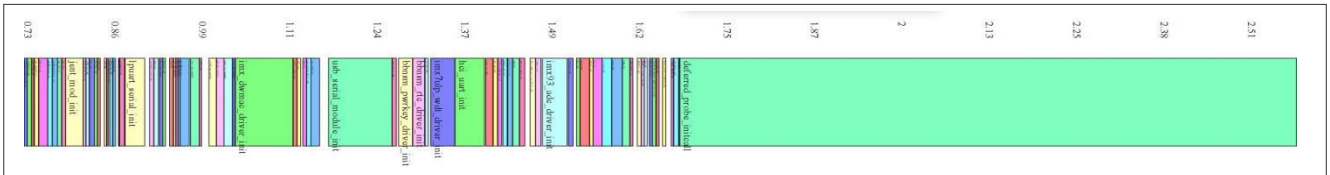

图5. 内核启动时间

5. 如果需要禁用某个驱动或功能，请更改内核配置。例如，禁用内核的调试功能和UBI文件系统。

```
$ bitbake -c menuconfig linux-imx
```

   a. 运行以下命令进入内核menuconfig：

   在menuconfig中，禁用CONFIG_UBIFS_FS和CONFIG_DEBUG_KERNEL，并保存配置。生成的.config文件包含以下几行：

```
# CONFIG_UBIFS_FS is not set
# CONFIG_DEBUG_KERNEL is not set
```

   b. 构建新的内核映像：

```
$ bitbake -f -c compile linux-imx
```

c. 生成包含新内核和ATF的映像，并使用以下命令将其复制到SD卡：

```
$ bitbake -f -c compile imx-boot
$ sudo mount /dev/sd<x>1 /mnt
$ cp <build_dir>/tmp/work/<machine_name>-poky-linux/imx-boot/<git_rev>/
git/<board>/kernel-atf<-container>.itb<.img> /mnt
$ umount /mnt
```

5 用户空间优化

本章包含以下信息。

5.1 在systemd启动前运行应用程序

如果需要，可以在systemd启动之前运行一个程序。

- 创建脚本/home/root/newinit.sh，通过该脚本可以在systemd启动前运行您的程序。以下是一个简单的示例，展示如何做。请将echo行替换为您想要运行的应用程序。

```
#!/bin/sh

echo “Early start” > /dev/kmsg

exec /lib/systemd/systemd
```

- 将此脚本设置为可执行状态：

```
$ chmod +x newinit.sh
```

- 将/sbin/init链接指向您的newinit.sh脚本。

```
$ ln -sf /home/root/newinit.sh /sbin/init
```

注意：

如需恢复至初始配置，请使用以下命令：

```
$ ln -sf /lib/systemd/systemd /sbin/init
```

- 重启板卡并查看内核日志。在dmesg中，搜索“**Early start**”字符串，可以看到newinit.sh脚本已在init进程之前执行。

6 结果

表3. 初始启动时间测量

	SPL			U-Boot				KERNEL	
板卡	BOOT ROM	DDR初始化	SPL初始化+加载U-Boot映像	U-Boot初始化(init_sequence_f)	U-Boot初始化(init_sequence_r)	启动序列	内核映像加载	ATF+内核启动到INIT的过程	总时间
	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)
i.MX 8MN	161	241	162	363	790	2894	333	3506	8450
i.MX 8MP	162	301	175	373	1726	4181	345	3627	10890

表3. 初始启动时间测量（续）

	SPL			U-Boot				KERNEL	
板卡	BOOT ROM	DDR初始化	SPL初始化+加载U-Boot映像	U-Boot初始化(init_sequence_f)	U-Boot初始化(init_sequence_r)	启动序列	内核映像加载	ATF+内核启动到INIT的过程	总时间
	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)
i.MX 8MM	142	265	117	412	812	2970	396	5002	10116
i.MX 93	369	111	117	628	1172	3271	412	3090	9170
i.MX 95	350	5052	216	482	652	4142	373	6250	17527

表4. 优化后启动时间测量数据

	SPL				KERNEL	
板卡	BOOT ROM	DDR初始化	SPL初始化	内核映像加载[1]	ATF + 内核启动到INIT的过程[2]	总时间
	(ms)	(ms)	(ms)	(ms)	(ms)	(ms)
i.MX 8MN	203	240	86	376	1185	2090
i.MX 8MP	187	301	97	382	1237	2204
i.MX 8MM[3]	139	265	63	1336	2956[4]	4759
i.MX 93	374	111	89	366	1391	2330
i.MX 95	350	20[5]	184	410	3307	4271

[1] CONFIG_DEBUG_KERNEL禁用，从而减小了内核映像的大小，这也相应减少了内核映像的加载时间。
[2] 使用quiet参数抑制内核日志消息。
[3] i.MX 8MM的内核映像加载时间较长，因为SPL不支持MMC UHS。
[4] i.MX 8M Mini EVK并未像i.MX 8M Plus那样内置连接至PCIe端口的Wi-Fi模块。因此，在PCIe PHY初始化过程中，系统会等待活动链接，这一过程会占用额外时间。如果在PCIe接口上连接了Wi-Fi模块，内核的启动时间可以缩短至1215 ms，总体启动时间则减少至3018 ms。
[5] DDR Quickboot已启用。

7 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

8 修订历史

表5汇总了本文的修订情况。

表5. 修订历史

文档ID	发布日期	说明
AN14093 v.2.0	2024年9月26日	• 重新构建文档 • 添加了i.MX95 EVK评估板 • 基于LF6.6.23版本，更新了使用Yocto的实现
AN14093 v.1.2	2024年2月26日	源代码中的版权日期
AN14093 v.1.1	2024年1月24日	添加了对i.MX 93 A1的支持
AN14093 v.1.0	2023年10月9日	首次公开发布

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com.cn/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AN14093

利用Falcon模式和内核优化，实现i.MX 8M和i.MX 9的快速启动

目录