

昉·惊鸿-7110启动手册

防·星光 2 版本: 1.4 日期: 2024/12/11 Doc ID: JH7110-BUGCH-001

法律声明

阅读本文件前的重要法律告知。

版权注释

版权 ©广东赛昉科技有限公司, 2023。版权所有。

本文档中的说明均基于"视为正确"提供,可能包含部分错误。内容可能因产品开发而定期更新或修订。广东赛昉科技有限公司 (以下简称"赛昉科技")保留对本协议中的任何内容进行更改的权利,恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件,无论是明示的还是默示的,包括但不限于适销性、特定用途适用性和 非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任,并明确表示无需承担任何及所有连带责任,包括但 不限于间接、偶然、特殊、惩戒性或由此造成的损害。

本文件中的所有材料受版权保护,为赛昉科技所有。不得以任何方式修改、编辑或断章取义本文件中的说明,本文件或 其任何部分仅限用于内部使用或教育培训。

联系我们:

地址: 广东省佛山市顺德区大良街道云路社区昊阳路2号A区S201室

网站: <u>http://www.starfivetech.com</u>

邮箱:

- <u>sales@starfivetech.com</u>(销售)
- <u>support@starfivetech.com</u> (支持)

前言

关于本指南和技术支持信息

关于本手册

本手册主要为赛昉科技 昉·惊鸿-7110用户和合作伙伴提供启动昉·惊鸿-7110 SoC和昉·星光 2单板计算机的深度理解。

修订历史

Table 0-1 修订历史

版本	发布说明	修订
1.4	2024/12/11	更新了 <u>启动流程内存映射 (on page 13)</u> 章节。
1.3	2023/11/06	更新了启动顺序。
1.2	2023/07/14	由于赛昉科技不再建议昉·惊鸿-7110用户直接从SD卡和eMMC引导,因此删除了所有相关描述并更新了启动流程图。
1.1.3	2023/05/10	更新了 <u>防星光 2启动模式设置 (on page 14)</u> 。
1.1.2	2022/02/02	小改动。更正 <u>启动地址分配 (on page 7</u>)中偏移值的错误。
1.1.1	2022/01/19	小改动。删除了同样在其他文件中交付的重复部分。
1.1	2022/01/18	细化启动过程并重新发布。
1.0	2022/01/16	首次发布。

注释和注意事项

•

• (

本指南中可能会出现以下注释和注意事项:

- *i* Tip: 建议如何在某个主题或步骤中应用信息。
- ・

 Note: 解释某个特例或阐释一个重要的点。
 - Ⅰmportant: 指出与某个主题或步骤有关的重要信息。
 - CAUTION:
 - 表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
 - Warning:
 - 表明某个操作或步骤可能导致物理伤害或硬件损坏。

Contents

st of Tables5	
st of Figures	
律声明2	
[言3	
简介7	
1.1. 概述	
1.2. 启动资源	
1.3. 启动地址分配	
1.4. BootROM	
1.5. SPL	
1.6. OpenSBI	
1.7. U-Boot	
启动流程12	
启动流程内存映射	
昉·星光 2启动模式设置14	

List of Tables

Table 0-1 修订历史	3
Table 1-1 启动源选择的值	7
Table 1-2 16 M Flash启动地址分配	7
Table 1-3 SD/eMMC启动地址分配	8
Table 1-4 BootROM加载源	9
Table 4-1 启动模式设置	14

_{目录} List of Figures

9
-

1. 简介

1.1. 概述

本文件旨在:

- •介绍在昉·惊鸿-7110上启动Linux操作系统的所有启动阶段。
- •提供了关于如何生成镜像包及其位置的说明。
- •提供了关于如何编写不同媒体和引导位置的说明。

本手册参考的源代码基于以下环境:

- OpenSBI v1.2
- •U-Boot版本: 2021.10
- Linux内核版本: 5.15
- •硬件开发板: 昉·星光2 (1.2 A/1.3 B)

Note:

对于不同的U-Boot或Linux内核版本,以上信息可能略有不同。

文件位置

找到包含以下信息的昉·惊鸿-7110软件开发工具包(SDK)。

- 代码仓: <u>https://github.com/starfive-tech/VisionFive2</u>
- 分支: JH7110_VisionFive2_devel
- Tag: 选择最新的tag。例如, VF2_v2.11.5比VF2_v2.10.10更新。

1.2. 启动资源

电源域AON_RGPIO用于选择启动Vector和BootLoader源,并为获取BootLoader镜像提供多种方法。 防·惊鸿-7110 SoC可以从下表中列出的任一源代码启动,并由AON_RGPIO[1,0](0x1702002c)进行选择。

处理器	BootROM	启动Vector	源列表
U74	0x00_2A00_0000	0x00_1301_0000	Quad SPI NOR flash memory
		0x00_1000_0000	UARTO

1.3. 启动地址分配

下图为16 M Flash的启动地址分配。

Table 1-2 16 M Flash启动地址分配

位移	长度	描述
0x0	0x80000	SPL
0xF0000	0x10000	U-Boot环境变量

Table 1-2 16 M Flash启动地址分配 (continued)

位移	长度	描述
0x100000	0x400000	<pre>fw_payload.img(OpenSBI + U-Boot)</pre>
0x600000	0x1000000	保留

Note:

自本文档1.2版起, 启动方式不再建议从SD卡或eMMC启动。但为了维护代码结构, 修改以下SD/eMMC启动地 址为"保留"。在设计基于昉·惊鸿-7110的设备时, 请注意这一变化。

Table 1-3 SD/eMMC启动地址分配

位移	长度	描述	注释
0x0	0x200	GPT PMBR	0x4: 备份地址
0x200	0x200	GPT表头	
0x400	0x1F_FC00	保留	
0x20_0000	0x20_0000	保留	分区1
0x40_0000	0x40_0000	保留	分区2
0x80_0000	0x1240_0000	Initramfs + UEnv.txt	分区3
0x12C0_0000	磁盘结束	系统rootfs	分区4

1.4. BootROM

BootROM是一个硬编码的启动程序,在防·惊鸿-7110上以0x2A00_0000的地址偏移量写入。该程序主要用于加载和执行 辅助程序加载器(SPL)。



BootROM使开发人员能够通过将SPL读取到SRAM(0x8000000)来插入来自不同介质访问的程序,包括QSPI flash和 UART。

通过使用AON_RGPIO[1,0](0x1702002c)的位,开发人员可以确认他们的启动模式。

下表解释了BootROM如何加载源。

Table 1-4 BootROM加载源

RGPIO1	RGPIO0	启动源	注释
0x0	0x0	Quad SPI NOR flash mem- ory	从扇区0读取SPL
0x1	0x1	UART0	当系统检测到UART的启动模式时,将进入Xmodem接收模式。 然后,用户可以使用串行电缆连接在Xmodem模式下导入恢复程 序。一旦文件确认传输完成,BootROM将自动运行恢复程序。

Note:

除了在BootROM中之外,您还可以在spl_tool中更改备份地址。

1.5. SPL

SPL是一个基于U-boot的启动程序。SPL的主要用途是促进DDR初始化并加载映像文件fw_payload.img(U-Boot+OpenSBI)。SPL从地址0x100000读取fw_payload.img,然后将其加载到DDR的地址0x40000000以进行操作。

1.6. OpenSBI

9

OpenSBI的二进制文件与U-Boot编译的二进制文件打包,以生成最终的fw_payload.bin文件。OpenSBI的主要功能包括:

- 为Linux提供基本的系统调用
- 将模式从M模式切换到S模式
- 跳转到0x4020_0000(位于DDR中)以执行U-Boot

正常输出信息如下图所示。

Figure 1-2 OpenSBI输出示例

U-Boot SPL 2021.10-00008-g48be500431-dirty (Jun 28 2023 - 18:39:23 +0800) DDR: 8G version: g8ad50857. Trying to boot from SPI

OpenSBI v1.2

Platform Name Platform Features Platform HART Count Platform IPI Device Platform Timer Device Platform Console Device Platform HSM Device Platform PMU Device Platform Reboot Device Platform Shutdown Device Platform Suspend Device Firmware Base Firmware Size Firmware RW Offset Runtime SBI Version	<pre>: StarFive VisionFive V2 : medeleg : 5 : aclint-mswi : aclint-mtimer @ 4000000Hz : uart8250 : : : pm-reset : pm-reset : : 0x400000000 : 392 KB : 0x40000 : 1.0</pre>
Domain0 Name Domain0 Boot HART Domain0 HARTs Domain0 Region00 Domain0 Region01 Domain0 Region02 Domain0 Region03 Domain0 Next Address Domain0 Next Arg1 Domain0 Next Mode Domain0 SysReset Domain0 SysSuspend	<pre>: root : 1 : 0*,1*,2*,3*,4* : 0x000000002000000-0x0000000000000000000</pre>
Boot HART ID Boot HART Domain Boot HART Priv Version Boot HART Base ISA Boot HART ISA Extensions Boot HART PMP Count Boot HART PMP Granularity Boot HART PMP Address Bits Boot HART MHPM Count Boot HART MIDELEG	<pre>: 1 : root : v1.11 : rv64imafdcbx : none : 8 : 4096 : 34 : 2 : 0x000000000000222</pre>

1.7. U-Boot

U-Boot以0x4020_0000运行,并在S模式下工作。它包含基本的文件系统和常用的外设驱动程序(如 GMAC、UART、QSPI、SDIO等)。U-Boot可以通过ETH(网络)、UART、QSPI、SDIO或NVMe(SSD)加载内核镜 像。

Figure 1-3 U-Boot界面



Note:

访问<u>RVspace</u>,以确保您能获取赛昉科技最新的文件和安装包。按Enter键确认操作或执行下一个命令。

2. 启动流程

本章介绍了防·惊鸿-7110 SoC的一般启动过程,包括用于启动路径等的镜像。

您可以从赛昉科技U-Boot的U-Boot TPL/SPL启动,这是镜像源代码。

下面的级联菜单显示了典型的昉·惊鸿-7110启动流程: BootROM > SPL + Open SBI + U-Boot > Kernel + File System > Boot Complete。





启动设备

昉·惊鸿-7110支持以下启动设备。

• QSPI Flash (在SPL + OpenSBI + U-Boot层) + SD Card/eMMC/NVMe (在Kernel + File System 层和后续层)

> Note:

系统将按顺序检测是否可以从以下设备顺序启动: **SD > eMMC > NVMe**。例如,如果在SD上找到引导程 序, eMMC将被忽略。

3. 启动流程内存映射

下图显示了昉·星光 2上昉·惊鸿-7110的启动流程内存映射。

Figure 3-1 启动流程内存映射



4. 昉·星光 2启动模式设置

昉·星光 2提供专门的pin,帮助用户在上电前配置启动模式。以下是可选的启动模式及其详细信息。

Table 4-1 启动模式设置

index	启动模式	RGPIO_1	RGPIO_0
1	1-bit QSPI Nor Flash	0 (L)	0 (L)
2	SDIO3.0	0 (L)	1 (H)
3	eMMC	1 (H)	0 (L)
4	UART	1 (H)	1 (H)

Note:

赛昉科技建议您使用1-bit QSPI Nor Flash模式启动,因为使用eMMC或SDIO3.0启动模式可能会发生小概率启动失败的情况。如果从eMMC或SDIO3.0启动失败,您可以尝试重启昉·星光 2。

下图显示了启动模式专用pin的位置及其定义。

Figure 4-1 启动模式设置位置



Figure 4-2 启动模式设置



QSPI RGPIO_1: 0 (L) RGPIO_0: 0 (L)



SDIO RGPIO_1: 0 (L) RGPIO_0: 1 (H)



eMMC RGPIO_1: 1 (H) RGPIO_0: 0 (L)



UART RGPIO_1: 1 (H) RGPIO_0: 1 (H)

Note: H for high level; L for low level.

