

量产烧写

使用指南

文档版本 23

发布日期 2018-12-05

版权所有 © 深圳市海思半导体有限公司2018。保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明

(上)、**HISILICON**、海思和其他海思商标均为深圳市海思半导体有限公司的商标。 本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: http://www.hisilicon.com/cn/

客户服务电话: 4008302118

客户服务邮箱: support@hisilicon.com

前言

概述

本文主要介绍量产烧录方案,包括如何制作量产烧录镜像、烧录方法及烧写注意事项等。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本	
Hi3798C	V2XX	
Hi3798M	V1XX/V2XX(H)/V3XX(H)	
Hi3796M	V1XX/V2XX	
Hi3716M	V31X	
Hi3716M	V33X	
Hi3716M	V4XX	

读者对象

本文档(本指南)主要适用于以下工程师:

- 技术支持工程师
- 软件开发工程师

作者信息

章节号	章节名称	作者信息
1	概述	L00227819
2	量产镜像烧写指导	L00227819
3	辅助信息烧录指导	Z00213260/G00180098/T00273561
4	OTP和高安烧写指南	G00227338
5	Secure Key Provision使用指导	H00378177
6	相关工具使用指导	Y00250933
7	量产问题定位指南	L00176142

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明	
2013-11-5	00B01	第一次版本发布。	
2014-01-02	01	2.2.1章节分成Linux SDK和Android SDK两个小节描述。	
2014-03-13	02	修改2.3.3章节。	
2014-05-23	03	删除原来的2.2硬件注意事项和Hipro-usb小节。	
2014-06-18	04	增加支持Hi379XX系列芯片。增加HiProinfo使用说明	
2014-07-15	05	新增4.1.2章节。新增第5章"量产问题定位指南"。	
2014-08-21	06	增加支持Hi3716MV310芯片的说明。	
2014-11-25	07	增加案例五中的说明内容。	
2014-12-30	08	修改3.2.1章节注意事项,添加了如何制作 ProgrammerforCA.bin文件的方法。	
2015-01-26	09	修改4.1.2.1章节的说明内容。	
2015-03-10	10	新增支持Hi3110EV500芯片。	
2015-04-24	11	新增支持Hi3798CV200、Hi3716MV420/410芯片。	
2016-03-10	12	新增支持Hi3716MV330芯片。	

修订日期	版本	修订说明	
2016-04-19	13	修改4.1.2章节。	
2016-07-20	14	增加Linux SDK+BootLoader/AppLoader方案;增加 Hi3798CV200量产问题定位指南;调整概述部分。	
2016-11-02	15	新增支持Hi3798MV200。	
2016-11-30	16	1.2.4章节中完善制作HiPro-serial镜像步骤。	
2017-03-07	17	去掉芯片平台限制描述。	
2017-07-11	18	全面更新文档大纲。新增支持Hi3798MV200	
2017-08-28	19	更新3.1章节中HIProInfo工具使用说明,新增3.3章节量产 DRM key烧录方法具体描述	
2017-12-18	20	全面刷新。	
2018-06-05	21	拆分3.2章节为3.2.1和3.2.2两个子章节。3.2.1为原3.2章节内容;另新增3.2.2章节,介绍Hi3796MV200系列芯片的HDCP Key Provision方法;修改5.2.4章节。	
2018-10-10	22	修改2.1.3.1、5.2.3、5.2.4章节。	
2018-12-05	23	新增3.5章节。修改3、5、6、7章节	

目录

前言	ii
1 概述	1
2 量产镜像烧录指导	
2.1 Linux	
2.1.1 准备工作	
2.1.1.1 工具	
2.1.1.2 镜像文件定义	
2.1.1.3 镜像生成方法	
2.1.2 SPI NOR+NAND Flash、NAND Flash 或 SPI NAND 的方案	
2.1.2.1 烧录器烧写方案	
2.1.2.2 U 盘烧写方案	
2.1.3 eMMC 的方案	
2.1.3.1 烧录器烧写方案	
2.1.3.2 U 盘烧写方案	
2.2 Android.	
2.2.1 准备工作	
2.2.1.1 工具	
2.2.1.2 镜像文件定义	
2.2.1.3 镜像生成方法	
2.2.2 SPI NOR+NAND Flash、NAND Flash 或 SPI NAND 的方案	
2.2.2.1 烧录器烧写方案	
2.2.2.2 U 盘烧写方案	
2.2.3 eMMC 的方案	16
2.2.3.1 烧录器烧写方案	16
2.2.3.2 U 盘烧写方案	16
3 辅助信息烧录指导	17
3.1 量产 MAC 和 ID 烧录方法	17
3.1.1 工具介绍	
3.1.2 使用方法	
3.1.2.1 工具配置说明	
3.1.2.2 烧写 MAC 和 ID	
3.1.3 常见问题	
· / = · · · · - · · · · · · · · · · · · · · · · · · ·	

3.2 量产 HDCP key 烧录方法	23
3.2.1 方法一	24
3.2.2 方法二	24
3.3 量产 DRM key 烧录方法	24
3.4 量产产品相关信息烧录方法	26
3.5 Android 量产 Keymaster Keybox 烧录方法	28
4 OTP 和高安烧录指导	29
4.1 非安全芯片 OTP 烧录	29
4.2 高安芯片 OTP 烧录	29
5 Secure Key Provision 使用指导	30
5.1 配置和编译	30
5.2 应用场景	30
5.2.1 HDCP Key Provision.	31
5.2.2 其他 rootkey Key Provision	31
6 相关工具使用指导	32
6.1 镜像制作工具使用指导	32
6.1.1 HiBurn 烧片器镜像介绍	32
6.1.1.1 制作 Nand 烧片器镜像	32
6.1.1.2 制作 Emmc 烧片器镜像	33
6.1.1.3 制作镜像无效数据设置	33
6.1.2 HiLoader 镜像制作介绍	34
6.1.2.1 打包 HISI 协议的升级文件	
6.1.2.2 打包 SSU 协议的升级文件	37
6.1.2.2.1 插入 SSU 协议	37
6.1.2.2.2 插入资源链接描述符	39
6.1.2.2.3 插入空数据包	40
6.1.2.3 注意事项	40
6.2 HDCPKey 切割工具使用指导	41
6.2.1 工具概述	41
6.2.2 环境概述	41
6.2.3 快速入门	41
6.2.3.1 安装	41
6.2.3.2 主界面概览	
6.2.4 界面及功能描述	
6.3 Secure_key Provision 工具使用指导	44
6.3.1 HDCP_Secure_KeyProvision 工具使用指导	
6.3.1.1 工具概述	
6.3.1.2 环境概述	
6.3.1.3 快速入门	
6.3.1.4 功能描述	
6.3.2 Rootkey Secure KeyProvision 工具使用指导	47

6.3.2.1 工具概述	47
6.3.2.2 环境概述	47
6.3.2.3 快速入门	47
6.3.2.4 功能描述	48
7 量产问题定位指南4	49
7.1 DDR training 相关打印	
7.1.1 Boot 下 ddr 时序窗口打印	49
7.1.2 DDR 报错问题打印	50

插图目录

2	图 2-1 SDK 配置界面	4
图	图 2-2 Upgrade 界面	5
S	图 2-3 Loader Type 界面	5
2	图 2-4 保存配置	6
冬	图 2-5 HiTool 工具界面	6
S	图 2-6 配置分区后的 HiTool 界面	7
2	图 2-7 eMMC 空白区填充配置	7
图	图 2-8 镜像制作完成的 HiTool 界面	8
S	图 2-9 HiTool 工具界面	8
冬	图 2-10 配置分区后的 HiTool 界面	9
冬	图 2-11 制作 NAND 烧录器镜像界面	9
图	图 2-12 设置 NAND 器件属性	10
冬	图 2-13 镜像制作完成的 HiTool 界面	10
冬	图 2-14 NAND 器件上逻辑分区布局	12
图	图 3-1 HiProinfo 工具 PC 端界面	18
冬	图 3-2 HiProinfo 工具主界面	21
2	图 3-3 HiProinfo 工具选择串口	22
2	图 3-4 HiProinfo 工具选择变化规则	22
8	图 3-5 HiProinfo 烧录成功	23
2	图 3-6 HiProinfo 工具 PC 端界面	25
图	图 3-7 HiProInfo 工具界面	27
冬	图 6-1 制作 Nand 烧片器镜像界面	32
8	图 6-2 制作烧片器镜像过程	33
2	图 6-3 制作镜像无效数据设置	34
3	图 6-4 HISI 协议打包界面	35
2	图 6-5 升级文件列表	35
2	图 6-6 升级文件参数配置	36
3	图 6-7 打包方式	36
2	图 6-8 SSU 协议打包界面	37
2	图 6-9 升级文件列表	37
2	图 6-10 参数配置	38
2	图 6-11 生成目标升级文件	38
冬	图 6-12 有线传输选择	39

图 6-13 地面传输选择	39
图 6-14 打包方式	39
图 6-15 文件参数配置	40
图 6-16 生成目标升级文件	40
图 6-17 工具主界面图	42
图 6-18 key 值的修改	42
图 6-19 HDCP_Key 配置使用图	43
图 6-20 ini 文件	45
图 6-21 ini 文件配置	46
图 6-22 生成的 bin 文件	
图 6-23 ini 文件	47
图 6-24 ini 文件	48
图 6-25 生成目标文件	48
图 7-1 写窗口	50
图 7-2 读窗口	50

表格目录

表 1-1 各平台的量产烧录方案	1
表 2-1 烧写起始地址和烧写长度	11
表 2-2 扩展 CSD 寄存器配置	13

】 概述

海思平台支持的存储器件有: SPI NOR、SPI NAND、NAND Flash和eMMC,通常包含以下存储搭配方案:

- SPI NOR + NAND Flash
- NAND Flash
- SPI NAND
- eMMC

各平台的存储搭配和可以支持的量产烧录方案如表1-1所示。

表 1-1 各平台的量产烧录方案

平台	Flash	方案
Hi3798MV100/ Hi3796MV100\ Hi3798MV310/ Hi3798MV200H/ Hi3798MV300H	NAND FlasheMMC	方案一、方案二
Hi3798CV200、 Hi3798MV200、 Hi3798MV300、 Hi3796MV200/ Hi3716MV450	 SPI NOR + NAND Flash NAND Flash SPI NAND eMMC 	方案一、方案二
Hi3716M V410/ Hi3716MV420	SPI NOR + NAND FlashNAND Flash	方案一

方案一

使用烧录器烧录,需要购买NAND或eMMC烧录器,成本高,但烧录速度快,效率高,适用于大批量生产。

方案二

使用U盘烧写,配合Android recovery或是AppLoader升级,可以实现全镜像烧写,不需要烧录器,缺点是需要每台机顶盒单独烧录,不适合大批量生产。

具体使用哪种方案,需要客户根据使用的存储器件、成本、量产规模等因素选取。

2 量产镜像烧录指导

2.1 Linux

2.1.1 准备工作

2.1.1.1 工具

HiTool中集成了HiBurn、HiLoader、HiFastplay等工具,本文档中只需要使用到HiBurn和HiLoader,推荐使用同SDK一起配套发布的HiTool,以避免出现兼容性问题。

- HiBurn用于制作烧录器镜像。
- HiLoader用于制作升级镜像。

2.1.1.2 镜像文件定义

涉及到需要准备的镜像定义如下:

● 正常镜像

指产品按正常使用SDK时编译出的镜像,包括boot镜像、bootargs镜像、kernel镜像和文件系统镜像等,通常是基于它来制作烧录器使用或升级时使用的镜像。

● apploader镜像

指通过U盘烧写时使用的loader镜像,由它完成对升级镜像的校验和烧写,不使用U盘烧写方案则不涉及。由apploader配置文件编译生成,镜像名称为apploader.bin。

● 支持apploader的boot镜像

指通过U盘烧写时使用的boot镜像,不使用U盘烧写方案则不涉及。由正常镜像的配置打开apploader功能后编译生成的boot镜像,镜像名称为: fastboot.bin。

● 升级镜像

指通过U盘烧写或升级时使用的镜像,由HiLoader工具根据正常镜像制作产生,镜像名称为usb_update.bin。

● 烧录器镜像

指通过烧录器烧写时使用的镜像,由HiBurn工具根据正常镜像制作产生,NAND 烧录器镜像和eMMC烧录器镜像制作参数略有点不同,详见2.1.1.3 章节。镜像名称在制作时指定。

2.1.1.3 镜像生成方法

- 正常镜像生成方法:见《Linux开发环境用户指南》文档的编译和运行SDK章节。
- apploader镜像的生成方法:

步骤1 根据芯片及单板类型选择对应的apploader配置文件。

以Hi3796MV200为例,选择hi3796mv2dmo hi3796mv200 apploader cfg.mak

步骤2 执行make build生成apploader镜像。

镜像地址: out/hi3796mv200/hi3796mv200dmo apploader/image/ xx image/ apploader.bin

----结束

● 支持apploader的boot镜像生成方法见下:

步骤1 根据芯片及单板类型选择对应配置文件配置文件。

以Hi3796MV200为例,选择hi3796mv2dmo_hi3796mv200_cfg.mak, 执行:

cp configs/hi3796mv200/hi3796mv2dmo_hi3796mv200_cfg.mak cfg.mak

将当前配置改为hi3796mv2dmo的配置,建议在复制前先备份好原来的cfg.mak文件,以便生成镜像文件后,可以恢复原来的配置。

步骤2 执行make menuconfig进入SDK配置界面。

图 2-1 SDK 配置界面

步骤3 进入Upgrade,打开Upgrade Support。

图 2-2 Upgrade 界面

步骤4 进入Upgrade Support,Loader Type选择Apploader。

图 2-3 Loader Type 界面

步骤5 退出menuconfig并保存配置。

图 2-4 保存配置



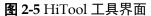
步骤6 执行make build完整编译,或者使用make hiboot只编译boot镜像。

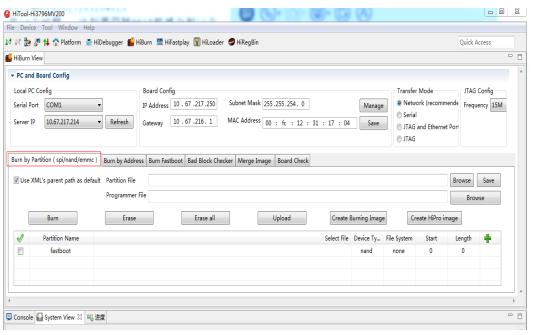
镜像地址: out/hi3796mv200/hi3796mv200dmo/image/xx image/ fastboot.bin

----结束

- 升级镜像usb_update.bin的制作方法: 使用HiLoader工具将正常镜像打包成USB升级文件,具体方法见《HiLoader 工具 使用指南》2.1章节。
- eMMC烧录器镜像的制作方法:

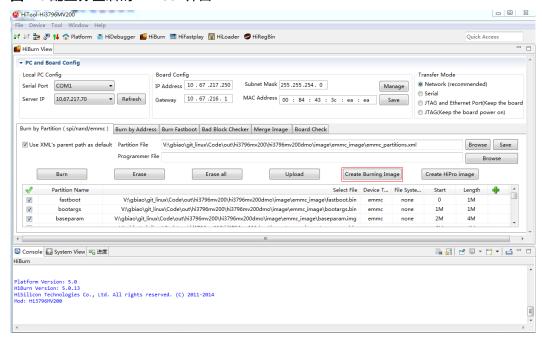
步骤1 打开HiTool工具,选择HiBurn,然后选择烧写eMMC栏。





步骤2 根据正常镜像配置各个分区,或者导入xml格式的分区表。

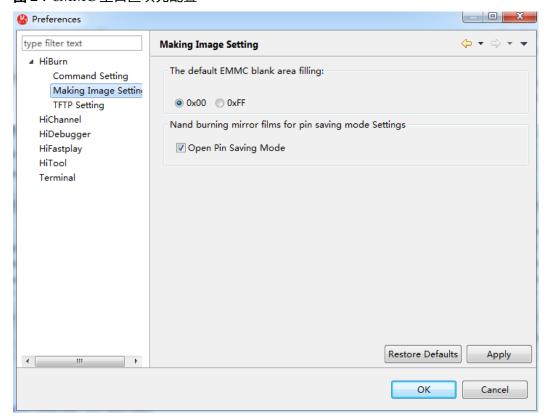
图 2-6 配置分区后的 HiTool 界面



步骤3 选择菜单栏->窗口->首选项,点击HiBurn,进行设置。

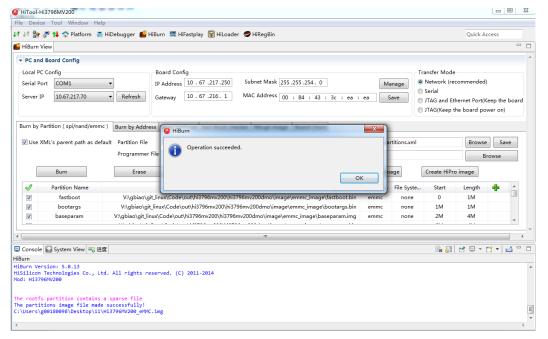
由于不同的器件对于空白区域设置不同,有些要求填充0x00,有些要求填充0xFF,用户最好与flash厂家或烧片器厂家确认后进行设置,保证烧片器时的性能。

图 2-7 eMMC 空白区填充配置



步骤4 点击创建烧片器镜像按钮(见<mark>图2-6</mark>中的红框部分),并保存镜像文件,例如生成的镜像名称为Hi3796MV200_eMMC.img。

图 2-8 镜像制作完成的 HiTool 界面



----结束

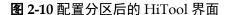
● NAND烧录器镜像的制作方法:

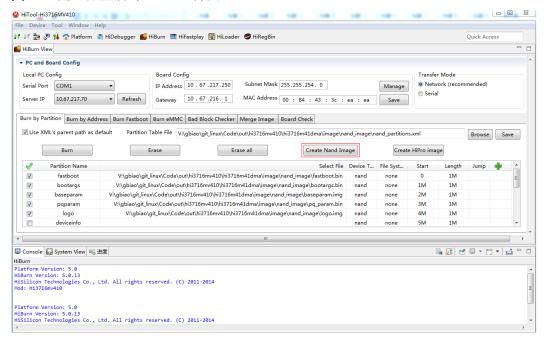
步骤1 打开HiTool工具,选择HiBurn,然后选择到烧写NAND栏。

图 2-9 HiTool 工具界面



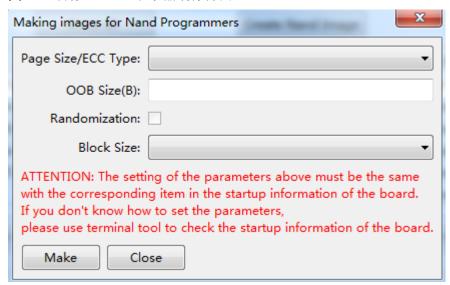
步骤2 根据正常镜像配置各个分区,或者导入xml格式分区表。





步骤3 点击创建烧录器镜像按钮(见<mark>图2-10</mark>中的红框部分),弹出制作NAND烧录器镜像界面。

图 2-11 制作 NAND 烧录器镜像界面

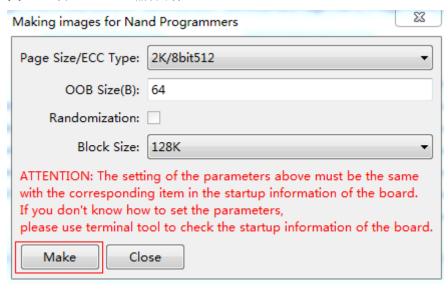


步骤4 根据Boot中打印的NAND器件信息,配置好页大小和ECC类型,OOB区的大小,如果器件支持Randomization,需要勾选,并配置好块大小,然后点击制作按钮。

本文档中都以某公司的NAND Flash器件为例,特性如下:

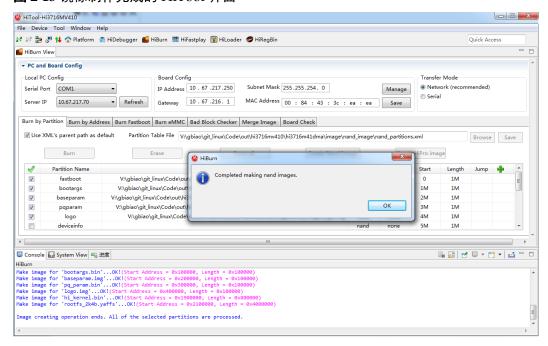
- 容量: 512Mbyte
- 块大小: 128Kbyte
- 页大小: 2KByte
- SpareArea区大小: 64Byte (以下称OOB区)
- ECC类型: 8bit

图 2-12 设置 NAND 器件属性



步骤5 选择好保存路径后,HiTool工具会自动遍历每个NAND分区文件生成对应的镜像,每个镜像均以.burn作为扩展名。

图 2-13 镜像制作完成的 HiTool 界面



----结束

2.1.2 SPI NOR+NAND Flash、NAND Flash 或 SPI NAND 的方案

2.1.2.1 烧录器烧写方案

对于SPI NOR的镜像,直接使用原始编译生成的文件即可烧录,如果烧写时遇到问题,需要联系SPI NOR烧录器厂家支持。

对于SPI NAND和NAND的镜像,需要使用HiTool工具进行转换,生成烧录器使用的镜像文件。本章节将重点描述如何制作Nand烧录器镜像。

使用烧录器烧写NAND, 主要包括以下步骤:

步骤1 通过正常镜像制作烧录器镜像文件,详见2.1.1.3 镜像生成方法 章节。

步骤2 计算每个烧录文件的烧写起始地址,获得烧录文件的烧写长度,具体的计算方法见后文。

步骤3 使用烧写器按照每个烧录文件的烧写起始地址和烧写长度,对NAND Flash中的区域进行单独烧写。具体烧录器的使用方法,需要与烧录器厂家联系。

----结束

用户进行分区时是以"页大小"为单位,而烧录器是以"页大小+OOB区"为单位,所以从用户分区到烧录器地址需要进行换算。烧录器看到的是物理地址,所以:

烧写起始地址 =
$$\frac{逻辑分区起始地址}{芯片页大小 + 芯片00B大小)}$$

例如,对于本文给出的Nand Flash芯片特性,有:

烧写起始地址 =
$$\frac{$$
逻辑分区起始地址 $\times (2K + 64)$

表 2-1 烧写起始地址和烧写长度

分区	原始文件	逻辑分区起始 地址	逻辑 分区 大小	转换后的烧 录器文件	烧写起始地 址	烧写长度
1	Boot.bin	0x0000_0000	1M	Boot.burn	0x0000_000 0	0x0002_8 380
2	Params.bin	0x0010_0000	1M	Params.burn	0x0010_800 0	0x0004_2 000
3	Kernel.img	0x0020_0000	5M	Kernel.burn	0x0021_000 0	0x0026_3 040
4	rootfs.yaffs2	0x0070_0000	50M	Yaffs2.burn	0x0073_800 0	0x006e_A 240

分区	原始文件	逻辑分区起始 地址	逻辑 分区 大小	转换后的烧 录器文件	烧写起始地 址	烧写长度
5	Ubi.img	0x0390_0000	100M	Ubi.burn	0x03AC_80 00	0x0124_E 000

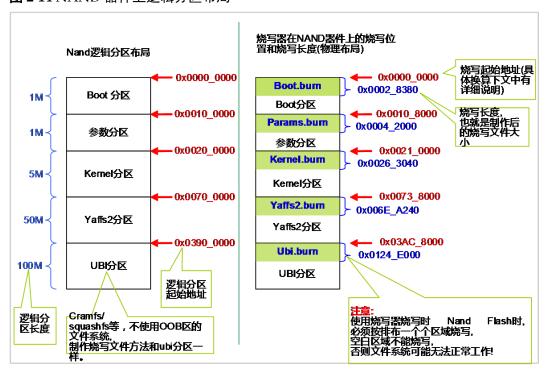
注意

烧录器烧写NAND Flash时,必须按排布一个个区域烧写,空白区域不烧写,否则文件系统可能无法正常工作。

烧录器必须能控制对NAND Flash中的某个区域进行单独烧写。

NAND器件上烧写时的逻辑分区布局见下图:

图 2-14 NAND 器件上逻辑分区布局



□□说明

- 以上分区只是一个示例,用户应该按照自己的实际情况来计算分区起始位置和分区大小。
- 分区之间不能重叠。
- 分区起始地址,长度应该是NAND Flash块大小整数倍。
- 分区大小要大于需要烧录的数据大小,并为坏块保留余量。
- 所有分区的大小之和要小于NAND Flash器件容量。
- Cramfs/Squashfs等不用OOB区的文件系统,制作烧写文件的方法和Kernel一样。
- 以上示例的物理布局中,未标示坏块。用户应该选择有跳坏块功能的烧录器,烧录器遇到坏块,自动跳到下一个块烧写。

2.1.2.2 U 盘烧写方案

Hi3716MV410、Hi3716MV420芯片不适用于本章节。

U盘烧写的准备工作如下:

- 准备U盘,格式化为FAT32文件系统,注意,簇大小不能超过8K。
- 准备好4个镜像文件: 支持apploader的boot(fastboot.bin),正常镜像的bootargs(bootargs.bin),apploader镜像(apploader.bin)和升级镜像(usb update.bin)。

□说明

- 不支持USB读卡器、SD卡转USB、USB硬盘、扩展分区U盘。
- U盘中有多个分区情况下,请确保只有一个分区包含usb update.bin升级包。
- U盘型号可使用金士顿品牌,推荐使用海思测试过比较稳定的型号: Kingston DataTraveler 4G USB2.0。

U盘烧写方案操作流程如下:

步骤1 将镜像文件拷贝到U盘根目录,并插入到单板USB2.0口。

步骤2 按下USB升级按键后,再给单板上电。

如果是空板,则不用按下USB升级按键。

步骤3 约2~5秒后指示灯闪烁,进入USB升级流程。

步骤4 约几分钟后,指示灯常亮,升级完成。

----结束

2.1.3 eMMC 的方案

2.1.3.1 烧录器烧写方案

∭说明

如果采用烧录器,请联系厂家确认是否与当前使用的芯片平台兼容。 不同烧录器烧录方法不一样,请先咨询烧录器厂家。

使用烧录器烧录,主要包括两部分准备工作:

- 制作烧录器镜像
- 配置eMMC的扩展CSD寄存器

制作烧录器镜像请参考**2.1.1.3 镜像生成方法**章节,此部分主要说明在使用烧录器烧写前需要配置eMMC哪些寄存器和配置为什么值。

eMMC 器件包含有 BOOT1、BOOT2、RPMB和 USER DATA 分区,同时支持 n_RST 管脚和下电复位。海思芯片方案所有镜像数据都烧写到 USER DATA 分区,BOOT 也从 USER DATA 区启动。烧录器必须按**表2-2**配置寄存器的值,否则单板无法启动。

表 2-2 扩展 CSD 寄存器配置

寄存器编号	寄存器值	说明
-------	------	----

	Hi3798MV100 Hi3796MV100 Hi3716MV410 Hi3716MV420 Hi3798CV200 Hi3798MV200 Hi3798MV300	Hi3796MV200 Hi3716MV450 Hi3798MV310 Hi3798MV200H Hi3798MV300H	
179	0x38	0x38	此寄存器用于配置 boot 分区,默认从 USER DATA 区开 始配置。
177	0x1	0x2	此寄存器用于配置 eMMC 在boot模式 下的总线宽度, 0x1表示4bits,0x2 表示8bits。
162	0x1	0x1	此寄存器用于配置 eMMC 器件的 n_RST 管脚是否有效。默认使用 n_RST 管脚,该寄存器必须配置成 0x1。

□□说明

- 必须在烧录之前完成eMMC扩展寄存器的配置。
- 部分烧录器可能不支持设置扩展CSD寄存器的功能,需烧录器厂家支持。
- 具体的设置由于各厂家的eMMC烧录器不同而存在差异,请参考烧录器手册来配置。

2.1.3.2 U 盘烧写方案

eMMC的U盘烧写方案与SPI、NAND上的步骤相同,请参考2.1.2.2 U盘烧写方案章节。

2.2 Android

2.2.1 准备工作

2.2.1.1 工具

HiTool中集成了HiBurn、HiLoader、HiFastplay等工具。推荐使用同SDK一起配套发布的HiTool,以避免出现兼容性问题。

● HiBurn用于制作烧录器镜像。

2.2.1.2 镜像文件定义

涉及到需要准备的镜像定义如下:

● 正常镜像

指产品按正常使用SDK时编译出的镜像,包括boot镜像、bootargs镜像、kernel镜像、文件系统镜像和升级包等,通常是基于它来制作烧录器使用或升级时使用的镜像。

● 烧录器镜像

指通过烧录器烧写时使用的镜像,由HiBurn工具根据正常镜像制作产生,NAND 烧录器镜像和eMMC烧录器镜像制作参数略有点不同,详见 2.1.1.3 镜像生成方法章节。镜像名称在制作时自定义。

2.2.1.3 镜像生成方法

- 正常镜像生成方法:见《Android 解决方案 开发指南》文档的开发编译章节。
- 烧录器镜像的制作方法与Linux相同,由正常镜像通过HiTool工具生成,见**2.1.1.3 镜像生成方法**章节

2.2.2 SPI NOR+NAND Flash、NAND Flash 或 SPI NAND 的方案

2.2.2.1 烧录器烧写方案

烧录器烧写方案与Linux上烧录器烧写方案相同,见2.1.2.1 烧录器烧写方案章节。

2.2.2.2 U 盘烧写方案

本小节仅适用于Hi398MV100、Hi3798CV200、Hi3798MV200、Hi3796MV200、Hi3716MV450、Hi3798MV300和Hi3798MV310芯片。准备工作如下:

- 准备U盘,格式化为FAT32文件系统,注意,簇大小不能超过8KB。
- 准备好4个镜像文件:正常镜像的boot(fastboot.bin),正常镜像的bootargs(bootargs.bin),正常镜像的recovery(recovery.img)和正常镜像的升级包(update.zip)。

□ 说明

- 不支持USB读卡器、SD卡转USB、USB硬盘、扩展分区U盘。
- U盘中有多个分区情况下,请确保只有一个分区包含usb update.bin升级包。
- U盘型号可使用金士顿品牌,推荐使用功能如下测试过比较稳定的型号: Kingston DataTraveler 4G USB2.0。

U盘烧写方案操作流程如下:

步骤1 将镜像文件拷贝到U盘根目录,并插入到单板USB2.0口。

步骤2 按下USB升级按键后,再给单板上的。

如果是空板,可以不用按下按键。

步骤3 约2~5秒后指示灯闪烁,进入USB升级流程。

步骤4 约几分钟后,指示灯常亮,升级完成。

----结束

2.2.3 eMMC 的方案

2.2.3.1 烧录器烧写方案

烧录器烧写方案与Linux上烧录器烧写方案相同,见2.1.2.1 烧录器烧写方案章节。

2.2.3.2 U 盘烧写方案

eMMC的U盘烧写方案与SPI、NAND上的步骤相同,请参考2.2.3.2 U盘烧写方案章节。

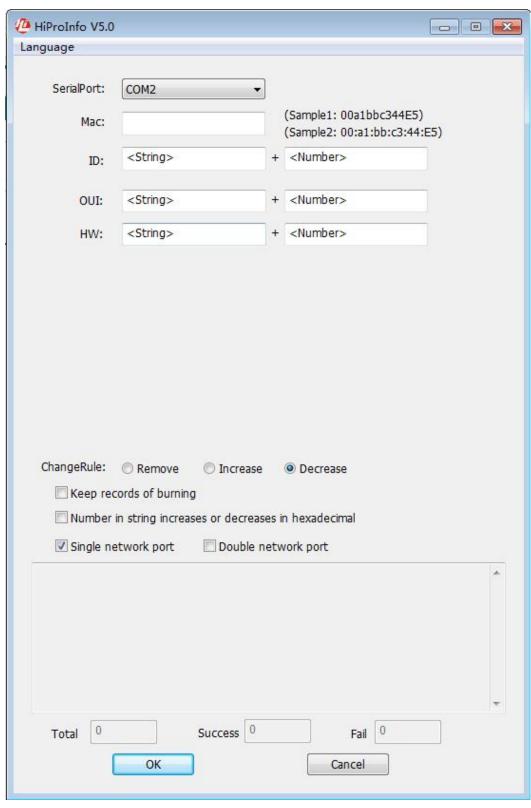
3 辅助信息烧录指导

3.1 量产 MAC 和 ID 烧录方法

3.1.1 工具介绍

HiProinfo工具是一个量产MAC、ID、HW、OUI、DRMKey等信息的烧录工具,支持通过串口烧录。

图 3-1 HiProinfo 工具 PC 端界面



HiProinfo工具要求单板为非裸板,工具通过串口下载分组数据,然后等待fastboot运行,通过与fastboot交互完成单板MAC、ID烧录。

3.1.2 使用方法

3.1.2.1 工具配置说明

HiProinfo工具会根据用户配置的不同打开或关闭部分功能。在工具所在目录下存在一个名为BurnConfig.ini的配置文件,用户在打开工具之前,必须按照自己的需求修改配置文件的各个配置选项,如想关闭某项配置只需在对应配置前使用";"将其注释即可。

● IsCA: 是否高安单板。普通单板需配置IsCA=0,高安即CA单板需配置IsCA=1

注意

目前只有部分芯片平台支持CA,具体请咨询FAE。

烧录CA单板时工具目录下必须存在ProgrammerforCA.bin文件。Android SDK版本中,ProgrammerforCA.bin就是安全版本生成的fastboot.bin,制作方法请参考《海思智能机顶盒安全方案 使用指南》第2章节,以Hi3798MV100单板为例,生成目录为out/target/product/Hi3798MV100/Security_L2/MAINTAIN/fastboot.bin,将其改名为ProgrammerforCA.bin放到工具目录下即可;Linux SDK版本中请使用make advca_programmer指令生成advca_programmer.bin,将其改名为ProgrammerforCA.bin放到工具目录下即可。

- ProgrammerFileName: 引导烧写的programmer文件名 对于Hi3798CV200/Hi3798MV200/Hi3796MV200芯片,当烧写的是miniboot时,或 者是不支持烧写指令的高安芯片时,需要配置此参数,默认文件名为 advca_programmer.bin,需要将此文件拷贝到工具工作目录下。该文件可以从SDK 编译出来的镜像包中获取。
- BoardType: 单板类型

当前支持的BoardType类型如下:

- Hi3716MV300单板配置BoardType=1;
- Hi3716MV310单板配置BoardType=3;
- Hi3798MV100/Hi3796MV100 单板需配置BoardType=4;
- Hi3798CV200/Hi3798MV200/Hi3796MV200/Hi3798MV300单板需配置 BoardType=5;
- Hi3716MV330单板需配置BoardType=6。
- MacBurnFlashType: Flash类型
 - NAND需配置MacBurnFlashType=1;
 - SPI需配置MacBurnFlashType=2;
 - eMMC需配置MacBurnFlashType=3;
 - 自动获取flash类型,配置MacBurnFlashType=0。
- MacBurnAddress: 烧写地址
 - NAND和SPI默认配置MacBurnAddress=0x3f0000;
 - eMMC默认配置MacBurnAddress=0xc00000;
 - 高安即CA默认配置MacBurnAddress=0x80000。

注意

MacBurnAddress配置错误会导致烧写失败或者烧错位置;地址配置必须块对齐,否则会导致烧写失败。默认配置仅供参考,具体数值请以实际的deviceinfo分区地址为准,deviceinfo分区的起始地址可以从烧写用的xml分区表中获取到。部分高安芯片不支持自动获取Flash类型和烧写地址,故建议用户手动配置这两个参数。

● CfgName: 配置自定义烧写信息。

例如配置Loader升级里面的OUI、HW相关序列号等烧写,每一条记录之间用半角竖线"|"隔开,每一条记录可以支持2种格式:

- 只有自定义信息: 此时工具会在界面上给出自定义配置框,且ID与MAC一样支持递增与递减。 最多支持3条这样的记录。
- 自定义输入固定字符串: 以<自定义名称:字符串>的形式配置,例如CfgName=OUI:123456789|HW, 工具会直接烧写配置的字符串123456789不再在界面上提供输如框。
- ExtraInfo: 配置额外烧写信息。

例如:可以进行共享分区文件烧写,或者其他版本信息等烧写,每一条记录之间 用半角竖线"|"隔开,每一条记录可以支持3种格式:

- 只配置偏移地址:

此时,工具会在界面上给出前缀字符与ID的配置框,且ID与MAC一样支持递增与递减。最多支持4条这样的记录。

偏移地址与烧写字符串:

以<地址:字符串的形式>给出。工具会直接将字符串写入对应的偏移地址,例如可以直接写入软件的版本号: ExtraInfo=0x1000: versionA|0x2000|0x3000。 支持的条目数没有限制。

- 将文件烧写到偏移地址:需以以下格式来配置文件路径: ExtraInfo=0x1000:<file:绝对路径>,例如ExtraInfo=0x1000:<file:D: \LoaderDB.bin>,文件不宜过大。

注意

ExtraInfo中的地址配置也必须块对齐,否则会导致烧写失败。

- AppLanguage: 默认语言配置。
 - "1"表示中文;
 - "2"表示英文。
- IDRule: ID输入内容限制
 - "0"表示无限制;
 - "1"表示只能输入数字:
 - "2"表示只能输入字母;
 - "3"表示只能输入字母或数字;
 - "4"表示必须有且只有字母和数字。

- IDLength: ID输入长度限制
 - "0"表示无限制;
 - 其他数字表示ID长度。
- DRMAddress: DRM key的flash临时缓冲地址
 - 如果要烧写WideWine drm key,则需配置此地址,默认为0x4000;
 - 如果不需要烧写,则无需配置。
- IsShowID: 是否显示ID栏 如果不需要显示ID栏,将此参数设置为0。

3.1.2.2 烧写 MAC 和 ID

烧录MAC、ID的步骤如下:

步骤1 用户按照需求修改工具所在目录中的BurnConfig.ini配置文件。 配置文件必须和工具放在同一目录。

步骤2 将PC机的串口与单板串口连接。

步骤3 在PC机上运行SDK的tools\windows\HiProinfo\HiProinfo_XX目录下的HiProinfo.exe程序。

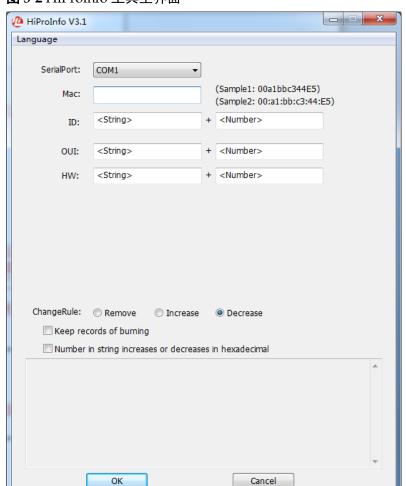


图 3-2 HiProinfo 工具主界面

步骤4 选择串口,选择已连接到单板的串口,确保该串口没有被其他工具占用,否则会报错。

图 3-3 HiProinfo 工具选择串口



步骤5 输入要烧写的MAC地址和ID以及其他信息。

第一次输入之后MAC会自动保存此配置,之后均会自动加载,输入格式请参见工具界面的例子。

步骤6 选择变化规则。

烧写成功时,会遵照此规则对上面填写的MAC地址及序列号进行自动更新,以免去手动修改的麻烦。此选项采用单选框形式。当选择Remove变化规则时,可以通过扫描枪扫描当前单板的Mac和ID值,光标位置请放在mac编辑框,扫描时工具会自动跳转。

图 3-4 HiProinfo 工具选择变化规则



步骤7 选择是否保存烧写记录,点击确定按钮,单板上电开始烧写。

注意

HiProinfo工具支持按回车键切换输入框,在最后一个输入框内按回车将直接开始烧写,因此在使用回车快速输入的方式时,请确认界面上其他配置(如串口,变化规则等)是否已经正确选择。串口、变化规则和保存烧写记录这3项会在退出时被保存,下次使用工具时将沿用上一次的选择。

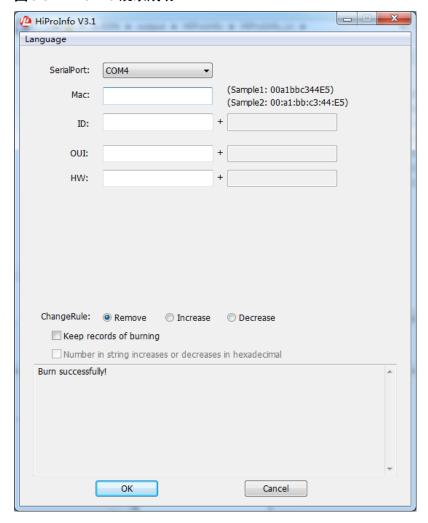


图 3-5 HiProinfo 烧录成功

----结束

3.1.3 常见问题

烧录失败,通常有以下几种原因:

- 串口线接触不好。
- 串口被占用。
- 单板类型配置错误。
- Flash类型配置错误。
- 单板为裸板。

请根据工具报错时的提示进行排查。

3.2 量产 HDCP key 烧录方法

我们提供两种HDCP Key烧录方法供客户选择:

- 方法一: 主要针对当前没有安全CPU的芯片。例如: Hi3798MV100系列/ Hi3716MV310系列/Hi3798MV200系列。
- 方法二:主要基于有安全CPU的芯片。例如: Hi3796MV200/Hi3716MV450/ Hi3716MV430系列芯片,此方案也称为"HDCP Key Provision方案"。

注意

HDCP1.4协议,OEM需手动烧写相关密钥到芯片OTP区域。

3.2.1 方法一

SDK包中有Code/sample/hdcpkey(编译生成路径Code/out/\$(CHIP_TYPE)/\$(CFG_VERSION)/obj/sample/hdcpkey)的测试用例,包含两种烧写方式:

- 原始HDCP key的烧写。
- 已经加密后的HDCP key的烧写。 关于如何产生海思加密后的HDCP key,请参考**6.2 HDCPKey切割工具使用指导**。

其拆分工具位于: SDK/tools/windows/HdcpTools之内。

3.2.2 方法二

此方法又称HDCP Secure Key Provision。对比方法一,此方法在安全执行环境中解密和 烧写HDCP Rootkey,安全性更高。

HDCP Secure Key Provision方法使用步骤如下:

步骤1 使用HDCP_Secure_KeyProvision工具加生成加密的HDCP RootKey和HDCP Key准备烧写。

工具的使用请参考6.3.1 HDCP Secure KeyProvision工具使用指导。

步骤2 参考Keyprovision sample烧写HDCP Rootkey,并得到加密的HDCP Key文件 EncryptedKey 332bytes.bin。

具体请参考5.2.1 HDCP Key Provision。

步骤3 将步骤2中输出的EncryptedKey_332bytes.bin烧写到Flash。

□□说明

使用HDCP_Secure_KeyProvision工具加密HDCP Key支持所有Key一包或一个Key一包的方式(ini 文件中PackagingType1_4参数控制), Keyprovision sample是以一个Key一包为例, 如果使用所有Key一包的方式,需要自行调整代码实现。

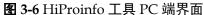
----结束

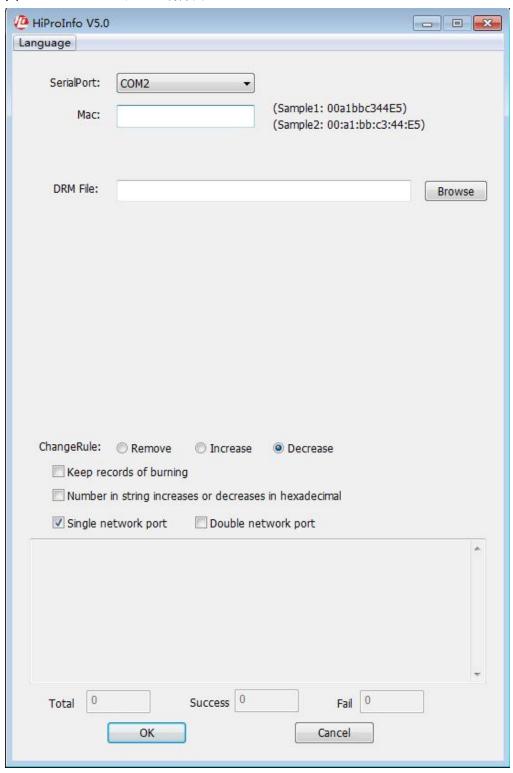
3.3 量产 DRM key 烧录方法

烧录DRM key的步骤如下:

步骤1 使用HiWidevine工具将Google发布出来的Widevine Keybox打包文件解析成量产烧写需要的bin文件,模式请选择Merge,具体操作请参考《HiWidevineTool 工具 使用指南.pdf》;

步骤2 参考3.1.2.1 工具配置说明章节,配置HiProInfo工具的BurnConfig.ini文件,打开HiProinfo工具,如图3-6所示;





步骤3 将单板与PC通过串口连接,在工具上面选择正确的串口,,并且保证未被其他程序占用;

步骤4 点击browse按钮,选择步骤1中生成的bin文件;

步骤5 选择是否保存烧写记录,点击确定按钮,单板上电开始烧写。

----结束

□□说明

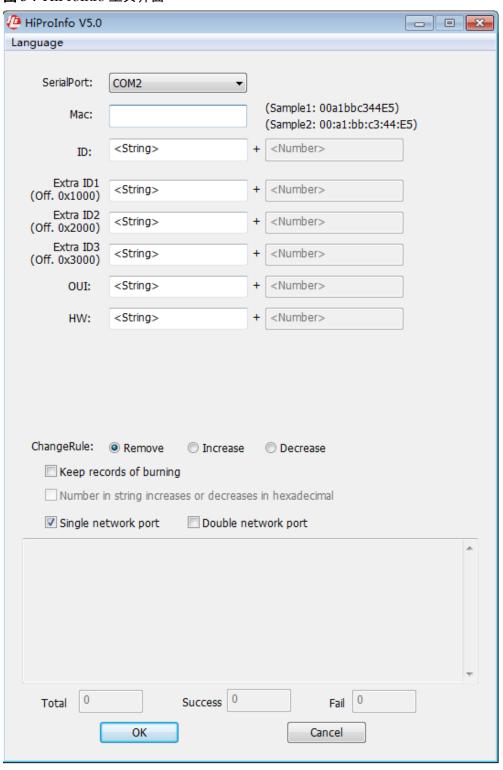
drmkey的数据是按照key+keybox的顺序烧写在板端。

3.4 量产产品相关信息烧录方法

量产产品相关信息操作步骤如下:

步骤1 参考3.1.2.1 工具配置说明,配置HiProInfo工具的BurnConfig.ini文件,其中跟产品信息相关的参数为CfgName和ExtraInfo。打开HiProInfo工具,如图3-7所示;

图 3-7 HiProInfo 工具界面



步骤2 参考步骤4进行后续烧写。

----结束

3.5 Android 量产 Keymaster Keybox 烧录方法

烧录keymaster keybox的步骤如下:

- 步骤1 向google申请keymaster keybox。将申请到的keymaster keybox文件(如: KeyBox_Fom_google.output)解压后放置到配置库device\hisilicon\bigfish\hardware \keymaster\sample目录下。
- **步骤2** 在device\hisilicon\bigfish\hardware\keymaster\sample目录下执行./build_native.sh,编译 ParseXMLKeybox.cpp生成keybox解析工具。
- **步骤3** 执行./ParseXMLKeybox KeyBox_Fom_google.output解析所有keybox,每个keybox生成一个独立文件。
- **步骤4** 选择一个keybox文件(如: keybox_00001.bin),将其更名为keymaster_keybox.bin,执行adb push推送到单板sdcard目录下。
- **步骤5** AndroidP版本,启动HiKeymasterProvision应用。(目的:做文件的拷贝。将/sdcard目录下面的keybox文件拷贝到/data/vendor/keymaster目录。AndroidO版本跳过此步骤)
- 步骤6 adb shell进入shell之后:
 - AndroidO版本: 执行sample_keymaster_provision /vendor/lib /sdcard/keymaster keybox.bin
 - AndroidP版本: 执行sample_keymaster_provision /vendor/lib/ /data/vendor/keymaster/keymaster keybox.bin

正确执行会有"provision successfully"的打印。

----结束

4 OTP 和高安烧录指导

4.1 非安全芯片 OTP 烧录

海思芯片有非安全芯片和海思公共安全芯片(原"H"Mark芯片),机顶盒厂商可以根据市场需要,自行烧写芯片,锁定为非安全芯片或海思公共安全芯片。

- 将芯片烧写并锁定为非安全芯片: HI_S32 HI_UNF_OTP_BurnToNormalChipset(HI_VOID)
- 将芯片烧写并锁定为海思公共安全芯片: HI_S32 HI_UNF_OTP_BurnToSecureChipset(HI_VOID) 获取芯片烧写锁定状态: HI_S32 HI_UNF_OTP_GetIDWordLockFlag(HI_BOOL *pbLockFlag)

Boot下和大系统下都提供了这三个接口,厂商可以根据自己生产方便,灵活选择在 Boot下烧写或在厂测软件中烧写。更多信息见《高安CA开发指南》第3.3.9海思非安全 芯片章节和第3.3.10海思公共安全芯片章节。

4.2 高安芯片 OTP 烧录

不同CAS厂商OTP烧录存在很大差异,具体烧录方案请参见对应CAS组件包中的文档《XXX 开发指南》,其中XXX表示对应CAS的名称。

5 Secure Key Provision 使用指导

5.1 配置和编译

步骤1 执行make menuconfig命令进入配置界面,选择如下描述配置项

Linux(REE) System --->
Features --->
[*] Keyprovision Support

步骤2 如果没有编译过整个SDK镜像,则执行make build - j指令,并烧录镜像到flash中;如果系统在步骤1前已经被编译过,则忽略该步骤,跳到下一步。

步骤3 对于Hi3796mv200/Hi3716mv450芯片,重新编译、烧写boot镜像;对于Hi3716mv430芯片,重新编译hi_pmoc.ko或文件系统并替换相应镜像。注意如果打开了"Build MSP in Kernel"选项,hi_pmoc.ko会编译进内核中,此时需要替换的是内核。

步骤4 在Code\sample\keyprovision下执行make命令,获取海思提供的参考用例 sample_keyprovision(编译生成路径

Code\out\ \$(CHIP TYPR)\\$(CFG VERSION)\obj\sample\keyprovision)

----结束

注意

Secure Key Provision 依赖于特定的SMCU镜像, 当选择支持Key Provision方案时, SMCU镜像仅仅用于量产过程中烧录相关的rootkey, 无法支持高安待机等功能, 所以量产软件需要打开Keyprovison Support选项, 最终发布的软件则需要关闭此选项。对于Hi3796mv200/Hi3716mv450芯片, SMCU镜像编译到boot镜像里; 而Hi3716mv430芯片, SMCU镜像编译到hi_pmoc.ko里, 最终镜像需要关闭"Keyprovision Support", 并重新编译回烧boot或者hi pmoc.ko (或内核/文件系统)。

5.2 应用场景

目前Hi3796mv200和Hi3716mv450芯片支持HDCP rootkey的烧写。Hi3716mv430芯片支持HDCP rootkey 和sw rootkey的烧写。

在Code\sample\keyprovision\keyprovision.c里封装了HI_KEYPROVISION_Burn接口,该接口实现将rootkey解密后烧写到OTP中。客户可将keyprovision.c整个文件适配到自己

的量产程序中,HI_KEYPROVISION_Burn的使用可参考Code\sample\keyprovision\sample keyprovision.c。

下面以sample为例,介绍Secure Key Provision功能的应用。

5.2.1 HDCP Key Provision

步骤1 参考**6.3.1 HDCP_Secure_KeyProvision 工具使用指导**章节,生成供烧写的数据,假设为keyprovision.bin。

步骤2 执行./sample keyprovision keyprovision.bin。

该步实现将HDCP rootkey烧写到了OTP,并在执行sample的当前路径生成了EncryptedKey_332bytes.bin文件,该文件用于**3.2.2 方法**二章节。

----结束

5.2.2 其他 rootkey Key Provision

步骤1 参考**6.3.2** Rootkey_Secure_KeyProvision 工具使用指导章节,生成供烧写的数据,假设为keyprovision.bin

步骤2 执行./sample keyprovision keyprovision.bin。

该步实现将rootkey烧写到了OTP中。该应用场景当前支持在Hi3716mv430芯片上烧写sw rootkey。

----结束

注意

- 1、由于OTP不可重复烧写,对于某一 rootkey 的烧写,仅可调用 sample_keyprovision 一次。第二次调用会报错""burn rootkey to otp error."。
- 2、rootkey烧写后必须重启单板,rootkey才会生效。

6相关工具使用指导

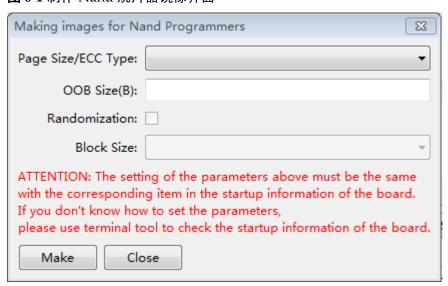
6.1 镜像制作工具使用指导

6.1.1 HiBurn 烧片器镜像介绍

6.1.1.1 制作 Nand 烧片器镜像

HiBurn提供了制作Nand烧片器镜像的功能。配置好分区列表后,点击制作Nand烧片器镜像按钮 ,会弹出Nand烧片器镜像制作界面。如<mark>图6-1</mark>所示。

图 6-1 制作 Nand 烧片器镜像界面



选择对话框中的各项数据以后(其中Randomization功能对8K及以上Page Size的器件开放),点击"制作"按钮,即可生成Nand烧片器的镜像。

注意

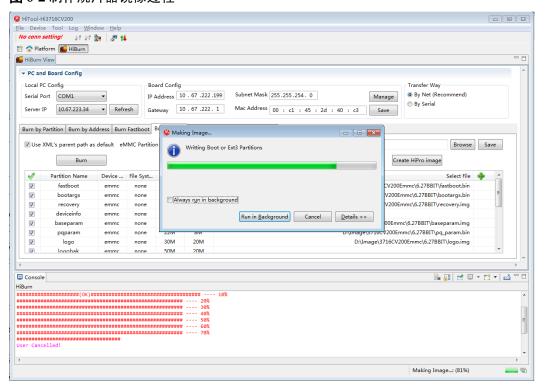
- 填入或选择的各项参数,必须与单板启动信息(可以使用超级终端类软件捕捉单板的启动信息并查看)中对应项的数值一致。
- 若用户不勾选某个分区,或者不为勾选的分区指定烧写文件,则无法制作该分区的 镜像文件。
- 若为非yaffs分区制作镜像,则分区表中的文件系统一项不能指定为yaffs; 为yaffs分区制作镜像时,文件系统必须指定为yaffs。否则,会导致做出的镜像不正确。

6.1.1.2 制作 Emmc 烧片器镜像

制作烧片器镜像功能可以将当前分区列表中选择的文件制作为烧片器镜像文件。配置

好分区列表后,点击制作烧片器镜像按钮 Create Programmer Image , 在弹出的保存对话框中设置好文件路径,制作烧片器镜像就开始了,如图6-2所示。

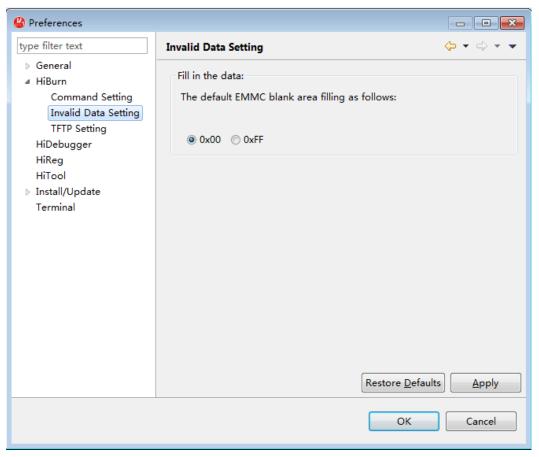
图 6-2 制作烧片器镜像过程



6.1.1.3 制作镜像无效数据设置

制作EMMC烧片器镜像中的无效数据的填充值也可通过首选项进行设置,点击菜单栏中"窗口"->"首选项"进入首选项对话框,进入"HiBurn"下"制作镜像无效数据设置"页面,如图6-3所示。

图 6-3 制作镜像无效数据设置

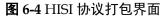


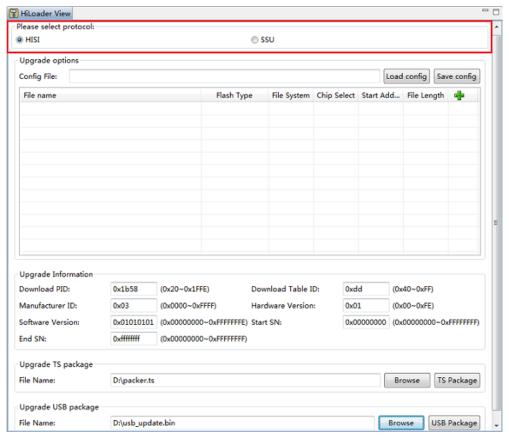
6.1.2 HiLoader 镜像制作介绍

6.1.2.1 打包 HISI 协议的升级文件

打包海思协议的升级文件步骤如下:

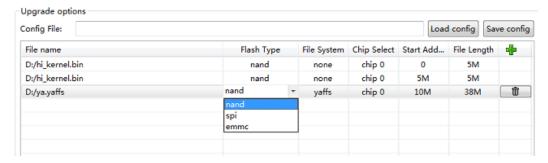
步骤1 打开HiLoader工具,在主界面上选择HISI协议,可看到HISI协议界面,如图6-4所示。





步骤2 在升级文件选项组里,单击 → 按钮,增加一行分区。可以在这一行选择器件类型、器件片选(支持4种器件片选)以及是否需要文件系统以及文件系统的类型,还可以修改分区的起始地址、分区长度以及分区的镜像文件。也可通过加载配置文件进行批量添加。另外,也可通过单击 — 按钮,删除一行分区。如图6-5所示。

图 6-5 升级文件列表



- 文件名:通过浏览添加分区对应的镜像文件。
- 器件类型:分区所在器件的类型。包含的器件有: nand (对应NAND flash), spi (对应SPI flash), emmc (对应eMMC flash)。

- 文件系统: 分区的文件系统类型。NAND/SPI flash支持的文件系统类型有: none, yaffs, ubi; eMMC flash支持的文件系统类型有: none, ext3/4; 其中none用于boot分区和kernel分区的打包。
- 器件片选:同一种芯片下第几块芯片。
- 开始地址:分区在器件上的起始位置。
- 分区长度:分区在器件上占用的长度。

步骤3 重复步骤 2添加其他的升级文件。

步骤4 在升级文件参数配置区域,对文件参数进行配置,如图6-6所示。

图 6-6 升级文件参数配置

pgrade information						
Download the PID:	0x1b58	(0x20~0x1FFE)	Download the table ID:	0xdd	(0x40~0xFF)	
Manufacturer ID:	0x0003	(0x0000~0xFFFF)	Hardware version number:	0x00000001	(0x00000000~0xFFFFFFF)	
Software version number:	0x01010101	(0x00000000~0xFFFFFFF)	Start sequence number:	0x00000000	(0x00000000~0xFFFFFFF)	
End sequence number:	0xfffffff	(0x00000000~0xFFFFFFF)				

- 下载PID: 升级流PID, 16位宽度(0x20~0x1FFE)。
- 下载Table ID: 指示下载序列的table ID, 8位宽度(0x40~0xFF)。
- 厂商ID号:表示厂家代号,16位宽度(0x0000~0xFFFF)。
- 硬件版本号:表示下载软件适用的硬件版本号,32位宽度(0x00000000~0xFFFFFFF)。
- 软件版本号:表示下载软件的版本号,32位宽度(0x00000000~0xFFFFFFF)。
- 起始序列号:表示需要更新软件的机顶盒的起始序列号,32位宽度 (0x0000000~0xFFFFFFF)。
- 结束序列号:表示需要更新软件的机顶盒的终止序列号,32位宽度(0x0000000~0xFFFFFFF)。

步骤5 在目标升级文件生成区域,单击 **Browse** 按钮可以选择生成的目标升级文件的名字 及路径、然后点击对应的打包按钮进行打包。

海思协议支持两种打包方式:

- TS流打包:或者点击 TS Package 打包生成HISI OTA协议的目标升级文件。
- USB打包:点击 USB Package 打包生成HISI FILE协议的目标升级文件。

⚠注意

USB升级只能使用Hisi协议,Loader中使用的 USB升级协议强制使用了Hisi协议。如图6-7所示。

图 6-7 打包方式



----结束

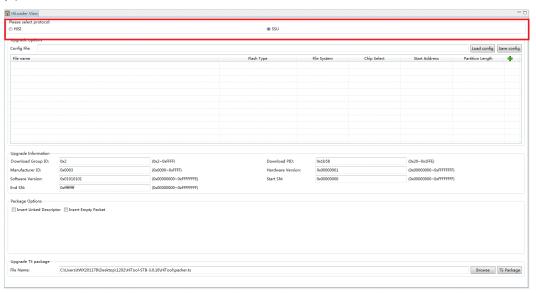
6.1.2.2 打包 SSU 协议的升级文件

6.1.2.2.1 插入 SSU 协议

插入SSU协议的升级文件步骤如下:

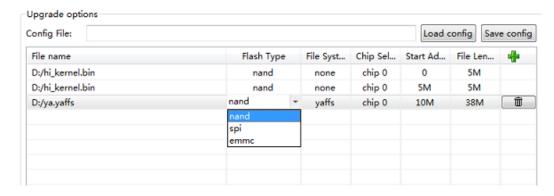
步骤1 打开HiLoader工具,在主界面上选择SSU协议,可看到SSU协议打包的界面,如图6-8 所示。

图 6-8 SSU 协议打包界面



步骤2 在升级文件选项组里,单击 ╈钮,增加一行分区。可以在这一行选择器件类型、器件片选(支持1种器件片选)以及是否需要文件系统以及文件系统的类型,还可以修改分区的起始地址、分区长度以及分区对应的镜像文件。也可通过加载配置文件进行批量添加。另外,也可通过单击 按钮,删除一行分区。如图6-9所示。

图 6-9 升级文件列表



- 文件名:通过浏览添加分区对应的镜像文件。
- 器件类型:分区所在器件的类型。包含的器件有:
 - NAND (对应NAND Flash)
 - SPI (对应SPI Flash)
 - eMMC (对应eMMC Flash)
- 文件系统:分区的文件系统类型。
 - NAND/SPI Flash支持的文件系统类型有: none, yaffs, ubi;
 - eMMC flash支持的文件系统类型有: none, ext3/4; 其中none用于boot分区和 kernel分区的打包。
- 器件片选:同一种芯片下第几块芯片。
- 开始地址:分区在器件上的起始位置。
- 分区长度:分区在器件上占用的长度。

步骤3 重复步骤 2添加其他的升级文件。

步骤4 在升级文件参数配置区域,对文件参数进行配置,如图6-10所示。

图 6-10 参数配置

Upgrade information					
Download the group ID:	0x2	(0x2~0xFFFF)	Download the PID:	0x1b58	(0x20~0x1FFE)
Manufacturer ID:	0x0003	(0x0000~0xFFFF)	Hardware version number:	0x00000001	(0x00000000~0xFFFFFFF)
Software version number:	0x01010101	(0x00000000~0xFFFFFFF)	Start sequence number:	0x00000000	(0x00000000~0xFFFFFFF)
End sequence number:	0xfffffff	(0x00000000~0xFFFFFFFF)			

- 下载Group ID, 16位宽度(0x2~0xFFFF)。
- 下载PID: 指示升级流PID, 16位宽度(0x20~0x1FFE)。
- 厂商ID号:表示厂家代号,16位宽度(0x0000~0xFFFF)。
- 硬件版本号:表示下载软件适用的硬件版本号,32位宽度(0x00000000~0xFFFFFFF)。
- 软件版本号:表示下载软件的版本号,32位宽度(0x00000000~0xFFFFFFF)。
- 起始序列号:表示需要更新软件的机顶盒的起始序列号,32位宽度 (0x0000000~0xFFFFFFF)。
- 结束序列号:表示需要更新软件的机顶盒的终止序列号,32位宽度(0x0000000~0xFFFFFFF)。

步骤5 在目标升级文件生成区域,单击 Browse 按钮可以选择生成的目标升级文件的名字及路径、然后点击对应的打包按钮进行打包。SSU协议包含一种打包方式: TS流打包,点击 TS Package 打包生成SSU协议的目标升级文件。如图6-11所示。

图 6-11 生成目标升级文件



----结束

6.1.2.2.2 插入资源链接描述符

插入资源链接描述符步骤如下:

步骤1 执行步骤1. 打开HiLoader工具,在主界面上选择SSU协议,可看到SSU协议打...到步骤4。

步骤2 选择"插入链接描述符"。

步骤3 选择一种传输方式。

若选择有线传输,在打包选择参数配置区域,对以下文件参数进行配置,如<mark>图6-12</mark>所示。

- 升级方式:未定义,强制升级,非强制升级
- 升级流符号率: 1000~99999KS/s。
- 调制方式:未定义,16QAM,32 QAM,64 QAM,128 QAM,256 QAM
- 频率: 1~9999MHz, 可输入小数, 小数点前4位和后4位。

图 6-12 有线传输选择



若传输方式选择地面传输,在打包选择参数配置区域,对文件参数进行配置,如**图 6-13**所示。

- 带宽: 6MHz, 7MHz, 8MHz
- 中心频率: 32位宽度, 10Hz(0x00000001)~42,949,672,950Hz(0xFFFFFFFFFF), 输入的值必须是10的倍数。
- 星座: QPSK, 16-QAM, 64-QAM

图 6-13 地面传输选择



步骤4 最后在目标升级文件生成区域,单击 **Browse** 按钮可以选择生成的目标升级文件的 名字及路径。

然后点击对应的打包按钮进行打包。SSU协议包含一种打包方式: TS流打包,点击 TS Package 打包生成SSU协议的目标升级文件。如图6-14所示。

图 6-14 打包方式



----结束

6.1.2.2.3 插入空数据包

插入空数据包的步骤如下:

步骤1 执行步骤1. 打开HiLoader工具,在主界面上选择SSU协议,可看到SSU协议打...到步骤4。

步骤2 在打包选择参数配置区域,对文件参数进行配置,如图6-15所示。

图 6-15 文件参数配置



- 升级流符号率,28位宽度(1000~99999) KS/s。
- 有效数据符号率,28位宽度(1000~99999) KS/s。

步骤3 最后在目标升级文件生成区域,单击 **Browse** 按钮可以选择生成的目标升级文件的 名字及路径、然后点击对应的打包按钮进行打包。

SSU协议包含一种打包方式: TS流打包,点击 TS Package 打包生成SSU协议的目标升级文件。如图6-16所示。

图 6-16 生成目标升级文件



注意

用户网络丢包率高的环境下,插入空数据包减少有效数据丢失。

----结束

6.1.2.3 注意事项

HiLoader工具使用时需要注意的事项有:

- 添加分区时,输入的分区长度注意要带单位k或者m(不区分大小写)。
- 添加分区时,输入的分区长度要应该大于实际选择文件的长度。
- 添加分区时,注意分区与分区占用的地址不能存在重叠。
- 使用HiLoader工具出现问题时,注意把当时的现象截图,反馈问题时一并反馈, 将有助于问题的定位及解决。
- TS流打包时间过长的原因可能是分区较多或者分区文件过大导致打包时花费的时间过长,属正常现象,请耐心等待即可。

- 当分区的文件超过一定大小时(HISI协议480MB,SSU协议240MB),工具会自动将该分区拆分成小分区文件。这主要是由于协议相关字段长度的限制造成的。
- 对于eMMC器件的ext4文件系统,工具支持非稀疏和稀疏两种格式的文件,工具自动根据文件头进行判断。
- 关于HISI协议的器件片选,NAND器件可以从boot的打印看出器件片选,例如 Nand(Hardware):

- Block: 128KB

- Page: 2KB

- OOB: 64B

- ECC: 4bit

- Chip: 128MB*1"

其中*1表示其片选0。其他器件请咨询单板的硬件开发人员。

6.2 HDCPKey 切割工具使用指导

6.2.1 工具概述

HDCP_Key是海思提供的用来拆分从Digital Content Protection LLC组织购买的HDCP1.4 KEY的工具,运行环境为Windows平台,存放在SDK的tools\windows\HdcpTools目录下面。对于Hi3796MV200和Hi3716MV430,需要参考6.3 Secure_key Provision工具使用指导章节描述的Secure Key Provision工具。

关于HDCP1.4 KEY的介绍可参考《HMS 开发指南》中的HDCP章节。

注意

HDCP1.4 KEY必须从Digital Content Protection LLC组织购买。

6.2.2 环境概述

HDCP key工具使用的环境准备如下:

步骤1 把HDCP key工具拷贝到Windows操作系统本地硬盘。

步骤2 双击运行即可。

----结束

6.2.3 快速入门

6.2.3.1 安装

本工具无需安装,双击运行即可。

6.2.3.2 主界面概览

本工具主界面显示如图6-17所示。

图 6-17 工具主界面图



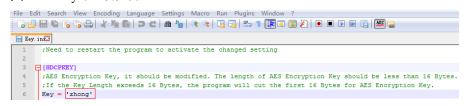
6.2.4 界面及功能描述

HDCP_Key 通过加密密匙进行加密,HDCP1.4 KEY的拆分以及加密拆分功能在此界面完成。拆分步骤如下:

步骤1 配置key.ini 文件。

建议用户配置自己的加密密匙key的值,注意务必和板端使用的解密密匙一致,否则板端解密会失败,导致key无法烧写成功,修改参考图6-18。

图 6-18 key 值的修改



步骤2 选择HDCP 1.4。如图6-19所示。

● 在下拉列表中选择HDCP 1.4

- 在Source File栏中选择HDCP1.4原始KEY包的bin文件
- Count栏会自动显示HDCP1.4 KEY的总数。

步骤3 切割HDCP1.4原始KEY包,设置Client ID。

Client ID用于标识客户信息,由32个英文字母或数字组成,由客户自行定义,建议设置为有意义的文字。如图6-19所示。

例如客户可以设置为: ClientID0000XXXX0123456789ABCDEF。这个在配置文件 key.ini 中配置。

图 6-19 HDCP_Key 配置使用图

```
;Need to restart the program to activate the changed setting

[HDCPKEY]
;AES Encryption Key. The length of AES Encryption Key should not be greater than 16 By

Key = 'zhong'

;Default Language Setting. 0 match Chinese, 1 match English.

DefaultLanguage = 0

;The first few letters of the generated keys. Use Letters and numbers if possible, oth

NameHead = HISI

;Client ID. The length of Client ID should not be greater than 16 Bytes. If Client ID

ClientID = ClientID0000XXXX

;if encrypt 2.2 Key. 0 match No, 1 match Yes, Other values or no value match No.

IfEncrypt2_2 = 0

;SOC vendor Reserved. Each 2 character match 1 Byte data. Total 7 Bytes.
```

步骤4 在Destination栏选择密钥存放文件夹,选择切割密钥的模式、切割的起始序号和生产的密钥数目。

模式下拉框提供如下三种方式:

- ALL Sets: 生产所有密钥。
- Multiple Sets: 生产1~n内连续的分KEY (n为HDCP1.4原始KEY包中KEY的数目)。 两个文本框需要分别填写希望生产的多个KEY的起始序号以及希望生产KEY个 数
- One Set: 生产指定序号的密钥。

----结束

□说明

在切割HDCP 1.4原始KEY包时,切分出来HDCP1.4 KEY文件大小为384个字节,各字段意义如下:

- 第0~7(8)字节: 海思标记位。HISI_xxx;
- 第8~15(8)字节: 工具版本号。V0000001;
- 第16~47(32)字节: Client ID, 用于标识客户信息, 由客户自行定义;
- 第48~367(320)字节: 加密后的320bytes HDCP1.4 KEY(16字节对齐);
- 第368~384(16)字节: 后续使用, 暂定为0。

6.3 Secure_key Provision 工具使用指导

注意

本章节仅适用于Hi3716M V430、Hi3716M V450、Hi3796M V200。

6.3.1 HDCP_Secure_KeyProvision 工具使用指导

6.3.1.1 工具概述

HDCP_Secure_Key_Provision是海思提供的用来拆分从Digital Content Protection LLC组织购买的HDCP1.4 KEY的工具,运行环境为Linux平台,存放在SDK的tools/linux/utils/advca/securekeyprovision下面。

关于HDCP1.4 KEY的介绍可参考《HMS 开发指南》中的HDCP章节。

注意

HDCP1.4 KEY必须从Digital Content Protection LLC组织购买。

6.3.1.2 环境概述

HDCP Secure KeyProvision工具使用的环境准备如下:

步骤1 把HDCP_Secure_KeyProvision工具拷贝到Linux操作系统中。

步骤2 解压后使用即可。

----结束

6.3.1.3 快速入门

步骤1 进入到工具64bit/bin目录下,在命令行执行./HDCP_Secure_KeyProvision,命令行会有如下打印:

Usage: ./HDCP_Secure_KeyProvision -v
Usage: ./HDCP_Secure_KeyProvision [ini path]
Optional: -S sourceFilePath -d destDir(multiple) -D destPath(one package) -c 0/1(0 for Hi3796MV200, 1 for Hi3716MV430)

步骤2 如图1所示,需要对ini文件进行配置。

图 6-20 ini 文件

```
key. ini 🔀
      #Need to restart the program to activate the changed setting
 3
    □ [HDCPKEY]
 4
 5
      HDCP1_4SrcFilePath =
 6
 7
 8
      HDCPDestFilePath = ./output.bin
9
10
      #The dest directory
      HDCPDestDirPath = ./output/
11
12
13
      HDCP1 4InitialSN = 1
14
15
      #how many key will be Generated
16
      KevNumber = 32
17
18
      #AES Encryption Key. The length of AES Encryption Key should not be greater than 16 Bytes. If Ke
      over 16 Bytes, The program will cut the first 16 Bytes for AES Encryption Key.;
19
      Key = 0x12345678123456781234567896385274
20
21
      #1.4 packaging type. 0 means all keys in 1 package, 1 means each key has 1 package. Other values
      value match all keys in 1 package.
22
      PackagingType1 4 = 1
23
24
      #The Manufacture information, it should be a string with no
25
      #more than 15 characters, if not, the 15 characters ahead will be used
26
      ManufactureInfo=
27
```

∭说明

- HDCP1_4SrcFilePath即为hdcp1.4key所在路径,也可以通过命令行参数-S直接指定;
- HDCPDestFilePath即为指定的输出文件路径(如果打包模式为0,即多个key合并为一个bin文件,则需要在ini文件中指定该路径,或者在命令行用-D指定);
- HDCPDestDirPath即为输出文件所在目录(如果打包模式为1,即将HDCPKEY切分为多个 Key,则需要在ini文件中指定该路径,或者在命令行用-d指定);
- HDCP1 4InitialSN即为指定的切割时key的起始序列号,第一个Key对应序列号1;
- KeyNumber对应要切割的Key的个数;
- Key对应加密用的OTPkey, 带上0x, 一共34个字符, 对应16个字节;
- PackagingType1_4对应打包方式,可以将切割出的Key拆成独立的包文件,或者是合并为一个包:
- ManufactureInfo对应厂商名称,不能超过15个字符。

----结束

6.3.1.4 功能描述

工具对Key进行拆分,对于单个HDCPKEY,会生成特定格式的包。

以Hi3716MV430为例,如果要拆分成多个Key,需要按照如下步骤进行操作:

步骤1 配置好ini文件的HDCP1_4InitialSN,KeyNumber,Key, PackageType1_4(1),ManufactureInfo,如**图1**所示

图 6-21 ini 文件配置

#Need to restart the program to activate the changed setting □ [HDCPKEY] HDCP1 4SrcFilePath = HDCPDestFilePath = HDCPDestDirPath = HDCP1 4InitialSN = 1 #how many key will be Generated KeyNumber = 7#AES Encryption Key. The length of AES Encryption Key should not be greate Length over 16 Bytes, The program will cut the first 16 Bytes for AES Encr Key = 0x74523896785634127856341278563412#1.4 packaging type. 0 means all keys in 1 package, 1 means each key has 1 or no value match all keys in 1 package. PackagingType1 4 = 1#The Manufacture information, it should be a string with no #more than 15 characters, if not, the 15 characters ahead will be used ManufactureInfo=

步骤2 将ini文件与工具拷贝到同目录

步骤3 在命令行执行./HDCP_Secure_Key_Provision key.ini -S cut1.4key.bin -d ./output_Dir/ -c 1,在./output_Dir/cut1.4key/0目录下(其中output_Dir即为命令行指定的路径,cut1.4key 即为指定的Key文件名称,0),就会生成如图2所示的文件

图 6-22 生成的 bin 文件

@ 0000001.bin	2018/12/22 10:05	BIN 文件	1 KB
@ 0000002.bin	2018/12/22 10:05	BIN 文件	1 KB
@ 0000003.bin	2018/12/22 10:05	BIN 文件	1 KB
6 0000004.bin	2018/12/22 10:05	BIN 文件	1 KB
6 0000005.bin	2018/12/22 10:05	BIN 文件	1 KB
6 0000006.bin	2018/12/22 10:05	BIN 文件	1 KB
6 0000007.bin	2018/12/22 10:05	BIN 文件	1 KB

----结束

□说明

- 当ini文件和命令行参数都指定了同一个参数,如,都指定了源文件所在路径,此时以命令行参数为准。
- 在切割HDCP 1.4原始KEY包时,切分出来单个HDCP1.4 KEY文件大小为812个字节,各字段意义如下:
 - 第0~31(32)字节为头区
 - 第32~127 (96) 字节为Key区
 - 第128~479 (352) 字节为Data区
 - 第480~811 (332) 字节为加密的HDCPKey

6.3.2 Rootkey_Secure_KeyProvision 工具使用指导

6.3.2.1 工具概述

该工具用于加密sw rootkey, stb ta root key等。

6.3.2.2 环境概述

Rootkey Secure KeyProvision工具使用的环境准备如下:

步骤1 把RootKey Secure KeyProvision工具拷贝到Linux操作系统中。

步骤2 解压后使用即可。

----结束

6.3.2.3 快速入门

步骤1 进入到工具bin64文件夹下,将ini文件夹下的key.ini文件拷贝到该目录,在命令行上执行./Rootkey Secure KeyProvision,会有如下打印:

```
Usage: ./Rootkey_Secure_KeyProvision -v
Usage: ./Rootkey_Secure_KeyProvision iniFile -D destFile(one package) -c
0/1(0 means Hi3796MV200, 1 means Hi3716MV430)
(For example: ./Rootkey_Secure_KeyProvision ../ini/key.ini -D ./
output.bin -c 1)
```

步骤2 配置ini文件,如图6-23所示。

图 6-23 ini 文件

```
#The Manufacture information, it should be a string with no
#more than 15 characters, if not, the 15 characters ahead will be used
ManufactureInfo=

#The RootKeyName can be set sw_rootkey, stb_ta_rootkey..., all of the let
RootKeyName = sw_rootkey

#The RootKey should be with 32 characters, that is 16 bytes
RootKey = 0x0123456789abcdef0123456789abcdef
```

□□说明

- ManufactureInfo对应厂商信息,不能超过15个字符。
- RootKeyName对应Rootkey的名称,此处必须为小写字符。
- RootKey即为rootKey, 算上0x, 为34个字符, 对应16个字节。

----结束

6.3.2.4 功能描述

步骤1 配置ini文件,如图6-24所示。

图 6-24 ini 文件

#The Manufacture information, it should be a string with no
#more than 15 characters, if not, the 15 characters ahead will be used
ManufactureInfo=hisilicon

#The RootKeyName can be set sw_rootkey, stb_ta_rootkey..., all of the lett
RootKeyName = sw_rootkey

#The RootKey should be with 32 characters, that is 16 bytes
RootKey = 0x0123456789abcdef0123456789abcdef

步骤2 在命令行输入./Rootkey_Secure_KeyProvision key.ini -c 1 -D out.bin,按回车

步骤3 在工具同目录下,会生成out.bin文件,如图6-25所示。

图 6-25 生成目标文件

名称	修改日期	类型	大小
	2018/12/5 11:33	Configuration Se	1 KB
🕼 out.bin	2018/12/22 15:24	BIN 文件	1 KB
Rootkey_Secure_KeyProvision	2018/12/5 11:33	文件	1,626 KB

----结束

了 量产问题定位指南

7.1 DDR training 相关打印

DDR training是海思平台内嵌的DDR时序自适应算法,主要有三方面的功能:

- 1. 单板上电DDR最佳时序窗口自适应;
- 2. Boot下ddr时序窗口打印;
- 3. DDR报错问题打印。

附件简要介绍第2、3两种用法。

7.1.1 Boot 下 ddr 时序窗口打印

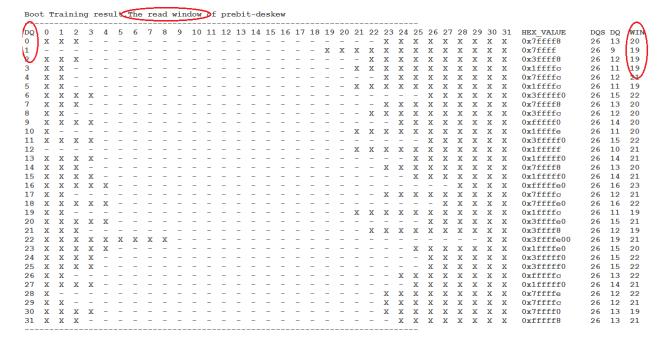
在fastboot命令行下,输入"ddr training"命令,则会出现如图7-1打印,主要包含以下几点信息:

- "The write window"和 "The read window":表示DDR写时序窗口和DDR读的时序窗口•
- DQ列:从0~31表示DQ0~DQ31,表示DDR的32bit,和主芯片出线管脚名称对应, 主芯片DDR的出线管脚名称区分两层板和四层板,请参考实际发布包原理图 Symbol;
- WIN列:表示对应DQ信号的总时序窗口,越大越好,要求窗口大于等于14级;
- "-":表示DDR时序窗口OK; "X":表示DDR时序窗口Fail; 每个DQ对应的"-"越多,则表示这个DQ的时序窗口越好,但如果整体往一边偏 的很多,则需要检查下boot版本。

图 7-1 写窗口

Boot Training result. The write window of prebit-deskew 8 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 HEX VALUE DQphDQ X X X 0xfffffe0 16 17 23 24 24 23 23 0xffffff0 18 16 X X X 18 18 16 15 0xfffffe0 0x7fffff0 14 16 15 23 23 24 0x3fffff8 18 18 18 18 0x7fffff8 0x1fffffe0 0x1fffffe0 17 17 24 24 14 17 15 18 18 10 0x3fffff8 23 24 24 24 24 22 22 22 x 18 12 0x7fffff8 18 18 20 15 13 14 X X 0xffffff00 0x7fffff8 -Х – X 18 17 17 х 17 12 15 --X 0xfffffc0 16 17 0x3fffff 11 18 19 0x7ffffe 12 12 22 23 23 21 22 0xfffffe 17 17 17 12 10 14 20 x x 0xfffffe 21 22 x x 0x1fffff8 х 23 24 X x x x 0xfffff 0x3fffff0 17 15 20 22 10 15 15 16 15 23 22 23 15 15 15 25 0x7fffff0 26 27 х 0x7ffffff0 28 0x3fffff8 15 15 14 16 23 22 29 0x7ffffe0 -Х 30 Х 0x1fffff8 15 14 22 Х

图 7-2 读窗口



7.1.2 DDR 报错问题打印

在使用hitools 工具烧写或者启动fastboot的过程中,DDR training算法会对DDR系统进行自检,出现异常则会有相应的打印:

Hi3798MV100/Hi3796MV100:

- 软件write leveling错误打印: SWL + 0000000X, X表示对应的byte;
- 软件dqs gating错误打印: SGA + 0000000X, X 表示 对应的byte;
- 硬件dqs gating错误打印: HGA + 00000000, 无法确认具体哪个DQS;
- 硬件read dataeye错误打印: HRD + 00000000, 无法确认具体哪个DQ;
- 软件read dataeye错误打印: SRD + 0000000X, X表示 第一个错误的DQ;
- 软件write dataeye错误打印: SWD + 0000000X, X表示第一个错误的DQ;

Hi3798CV200:

- 软件write leveling错误打印: DDRTR WL Err: PhyXXXXXXXByteXXXXXXXX
- 硬件gate错误打印: DDRTR HWG Err: PhyXXXXXXXX
- 软件gate错误打印: DDRTR Gate Err: PhyXXXXXXXXByteXXXXXXXX
- 软件DDRT错误打印: DDRTR DDRT Err: PhyXXXXXXXX
- 硬件读dataeye错误打印: DDRTR HWRD Err: PhyXXXXXXXX
- 软件mpr错误打印: DDRTR Dataeye Err: PhyXXXXXXXByteXXXXXXXXDQXXXXXXXX
- 软件dataeye错误打印: DDRTR Dataeye Err: PhyXXXXXXXByteXXXXXXXDQXXXXXXXX
- 软件Lowpower command addresss错误打印: DDRTR LPCA Err: PhyXXXXXXXX
- 精简的DDR training 错误打印:

- 字母E为Error缩写,表示错误码,后面的8位数字为十六进制:
 - i. 0x00000001 Write Leveling error
 - ii. 0x00000002 Hardware Gating error
 - iii. 0x00000004 Software Gating error
 - iv. 0x00000008 DDRT test time out
 - v. 0x00000010 Hardware read dataeye error
 - vi. 0x00000020 MPR error
 - vii. 0x00000040 Dataeye error
 - viii. 0x00000080 LPDDR CA error
- 字母P表示DDR PHY基址,后面的8位数字为十六进制。
- 字母B表示Byte 序号,后面的8位数字为十六进制。
- 字母D表示DQ 序号,后面的8位数字为十六进制。

出现以上错误打印信息,基本是因为DDR与主芯片之间DQS/DQ 等信号的通断或者信号质量有问题,导致单板无法正常启动。

海思方案为了保证系统正常启动后,能稳定的运行,在DDR自检时,对系统正常启动的DDR时序窗口进行了限制。从实验数据显示,DDR时序窗口大于等于12级,则基本能保证DDR系统稳定运行,因此,DDR training算法,在系统上电DDR自检时,如果发现部分DQ时序窗口小于12级,则会打印: DDR training failed, result: 00000004, 防止量产时漏筛。

可以通过在fastboot表格中配置寄存器: 0xf80000d0的bit31=1来disable掉功能,重新编译fastboot烧写后,系统能正常启动,从而通过在fastboot命令行输入: ddr training 命令,查看详细的DDR时序窗口,协助定位DQ时序偏小的原因。