

UNISOC Android10.0 GSI 客户调试指南

适用产品信息	SC9863A/SC9832E/SC7731E /UMS512(T)
适用版本信息	Andorid10.0
关键字	GSI

Unisoc Confidential For hiar

声明

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

版本历史

版本	日期	备注
V1.0	2019/12/18	初稿
V1.1	2020/01/08	更新部分标题编号
V1.2	2020/02/12	更新第九章部分内容,新增 GSI data 分区格式化注意事项等,以及章节顺序调节

Unisoc Confidential For hiar

目录

1 前言	1
1.1 范围	1
1.2 缩略语	1
2 概述	2
2.1 GSI 介绍与 Android10.0 动态分区简介	2
2.2 GSI 维护方式介绍	2
2.2.1 Fastboot 工具烧录	2
2.2.2 下载工具	2
2.3 Android10.0 GSI 新变化	3
2.3.1 GSI 编译目标变化	3
2.3.2 Android10.0 GSI avb 校验方式的新变化	3
2.3.3 GSI 需要打包到 Super 镜像	3
2.3.4 GSI Rollback	3
2.4 Android10.0 XTS 测试流程	4
3 配置 fastboot 烧录环境	6
3.1 如何安装/配置如下工具	6
3.2 升级 PC 端 fastboot 工具	6
3.3 生成解锁 bootloader 所需 bin 文件生成说明	7
4 解锁 bootloader	9
5 加锁 bootLoader	11
6 Android10.0 GSI XTS 测试要求介绍	12

6.1 Android 10.0 GSI 测试要求	12
6.2 Devices upgrading to Android 10.0	12
6.3 GSI build targets for Treble Compliance tests	13
6.4 VTS 测试环境要求	13
7 下载 GSI 说明	14
8 烧录 GSI 逻辑分区	15
8.1 烧录动态分区（注意 selinux 权限问题，必须先解锁 bootloader）	15
8.2 烧录物理分区（注意 selinux 权限问题，必须先解锁）	15
9 GSI 开机验证说明	17
9.1 使用 fastboot 和 bootloader 烧录镜像验证	17
9.2 直接烧录 GSI PAC 版本验证（平台维护方式）	17
9.3 GSI 开机验证	18
9.4 GSI 自检项	19
10 GSI 分支下载以及编译说明	20
10.1 Android10.0 GSI 分支下载编译	20
10.2 Android10.0 GSI upload 策略说明	20
11 Q&A	21

1 前言

1.1 范围

此文档适用于所有 UNISOC Android 10.0 平台上 GSI 的调试与维护开发人员。

1.2 缩略语

名称	全称	定义
GSI	Generic System Image	指已针对 Android 设备调整配置的系统映像
bootloader 模式		Android10 将原 bootloader 一分为二，adb reboot bootloader 进入
fastbootd 模式		Android10 将原 bootloader 一分为二，adb reboot fastboot 进入
Host fastboot		PC 端工具，bootloader 模式和 fastbootd 模式共用

Unisoc Confidential For hiar

2 概述

2.1 GSI 介绍与 Android10.0 动态分区简介

GSI (Generic System Image)用于运行 VTS 和 CTS-on-GSI 测试。为确保运行最新版 Android 的设备能够正确实现供应商接口，需要将 Android 设备的系统镜像替换为 GSI，然后使用供应商测试套件 (VTS) 和兼容性测试套件 (CTS) 测试设备。

动态分区 (Dynamic Partitions) 是 Android 10 新增功能，是用户空间的分区系统 (userspace partitioning system to Android)。动态分区一般包括：system/vendor/product/odm 等。

fastbootd 模式是伴随动态分区功能而来的，由于动态分区是用户空间的分区，bootloader 不能访问用户空间的动态分区。因此，Android 10 将原 bootloader 一分为二，将用户空间的功能移动到 recovery 镜像称作 fastbootd 模式。

2.2 GSI 维护方式介绍

UNISOC Android10.0 GSI 维护支持两种方式：

- Fastboot 工具烧录
- 下载工具 (ResearchDownload) 下载

2.2.1 Fastboot 工具烧录

Android10.0 引入了 super 只读分区的 feature，平台默认将 system/vendor/product 一起打包到 super 分区，导致 Android10.0 需要在用户空间烧录逻辑分区，进而引入了一种新的 fastbootd 模式，另外原有的 bootloader 模式还是用于烧录物理分区。

2.2.2 下载工具

UNISOC 平台维护 GSI 使用打 pac 的方式，全编译时会将 GSI 仓库的 system 跟 vendor/product 打包生成 super_gsi32/super_gsi64 的 super 镜像，制作版本时可以根据当前工程的架构选择合适的

super_gsi,生成对应的 GSI pac。然后使用下载工具烧录 pac 即可验证。

2.3 Android10.0 GSI 新变化

2.3.1 GSI 编译目标变化

Android10.0 GSI 编译变体由之前的 userdebug 类型变成了 user+google 签名的类型，这个变化目的是为了更接近最终释放的 user 版本，以及保护 GSI 安全。

2.3.2 Android10.0 GSI avb 校验方式的新变化

Android10.0 GSI avb 校验使用的 google pubkey ,平台预置了 google 发布的 pubkey ,故跟 Android9.0 相比，不再需要替换 vbmeta (10.0 之前的方式) 完成 disabled avb 功能。

1. Add the following in device/<company>/<project>/device.mk:
Installs gsi keys into ramdisk, to boot a GSI with verified boot.
\$ (call inherit-product, \$(SRC_TARGET_DIR)/product/gsi_keys.mk)
2. Append avb_keys=/avb/q-gsi.avbpubkey:key2:key3 for the /system entry in the fstab file, for example:
system /system ext4 ro,barrier=1
wait,slotselect,avb=vbmeta_system,logical,first_stage_mount,avb_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey

2.3.3 GSI 需要打包到 Super 镜像

UNISOC 平台维护 GSI 使用打 pac 的方式 ,全编译时会把 GSI 仓库的 system 跟 vendor/product 打包生成 super_gsi32/super_gsi64 的 super 镜像，制作版本时可以根据当前工程的架构选择合适的 super_gsi, 生成对应的 GSI pac。

2.3.4 GSI Rollback

Android 10.0 平台 GSI 与安全补丁匹配规则引入了新的变化,Google 升级要求是 GSI 的 SPL >= 安全补丁的 SPL。 GSI 的 SPL 不能比 vbmeta_system(设备)的 SPL 旧。

举例说明：

如果当前安全补丁的 SPL ：2020-04-05，

则可选的 GSI 的 SPL ：2020-04-05，2020-05-01，2020-05-05 等，

不可选的 GSI 的 SPL ：2020-04-01，2020-03-05 等，

故建议在合入安全补丁前，需优先升级 GSI，如果升级顺序不符，存在 GSI 版本无法开机的风险。

无法开机原因说明：

若 GSI 的 SPL 比安全补丁旧，在 first init 阶段会发生 GSI Rollback，导致 GSI 版本在 lock 状态下无法开机。

具体代码参考：

system/core/init/first_stage_mount.cpp

```
static bool IsStandaloneImageRollback(const AvbHandle& builtin_vbmeta,
                                       const AvbHandle& standalone_vbmeta,
                                       const FstabEntry& fstab_entry) {
    std::string old_spl = builtin_vbmeta.GetSecurityPatchLevel(fstab_entry);
    std::string new_spl = standalone_vbmeta.GetSecurityPatchLevel(fstab_entry);

    bool rollbacked = false;
    if (old_spl.empty() || new_spl.empty() || new_spl < old_spl) {
        rollbacked = true;
    }

    if (rollbacked) {
        LOG(ERROR) << "Image rollback detected for " << fstab_entry.mount_point
                   << ", SPL switches from '" << old_spl << "' to '" << new_spl << "'";
        if (AvbHandle::IsDeviceUnlocked()) {
            LOG(INFO) << "Allowing rollbacked standalone image when the device is unlocked";
            return false;
        }
    }

    return rollbacked;
}
```

也就是说升级安全补丁前，一定要先升级 GSI，否则 lock 状态下 GSI 版本存在无法开机的风险。

2.4 Android10.0 XTS 测试流程

1. Run CTS on original system
2. Flash boot.img with Root Ramdisk
3. Run STS
4. Restore original boot.img
5. Flash GSI and disable verified boot
6. Run CTS-on-GSI

7. Flash boot.img with Root Ramdisk

8. Run VTS

如上所述，跟 GSI 相关的 CTS-ON-GSI 测试需要搭载正常的 boot.img+system.img(GSI),VTS 测试则需要搭载 boot-debug.img+system.img(GSI)。同时 VTS 测试需要 unlock bootloader。

Unisoc Confidential For hiar

3 配置 fastboot 烧录环境

Android10.0 使用 fastboot 工具烧录 GSI，需要安装特定的环境，如 adb，fastboot。

3.1 如何安装/配置如下工具

➤ Adb

下载可执行程序，设置环境变量，安装 adb 驱动（官网下载）就可以正常连接手机。

➤ Fastboot

安装 Fastboot 驱动（官网下载），（连接手机，需要进入 fastboot 模式才能安装 fastboot 驱动）。

3.2 升级 PC 端 fastboot 工具

需要更新为 Android10.0 的 PC 工具，具体可以在源码根目录下，make fastboot -j16，即可以生成 fastboot 工具。

Linux 版：

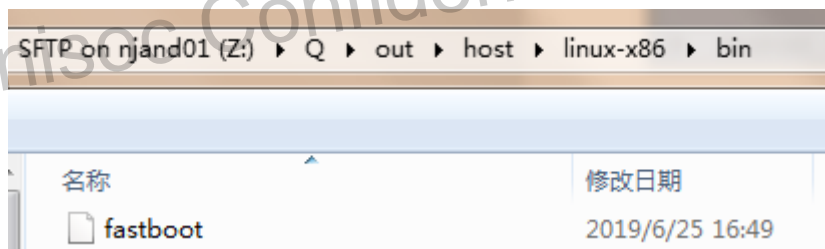


图 1 linux 版 fastboot 路径

Windows 版：

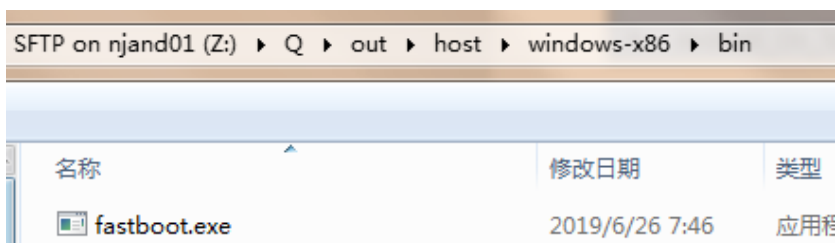
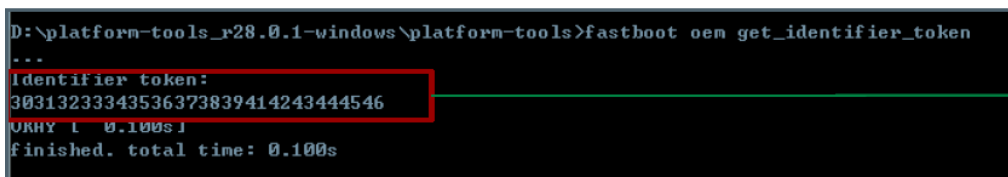


图 2 windows 版 fastboot 路径

3.3 生成解锁 bootloader 所需 bin 文件生成说明

生成解锁文件，具体步骤如下：

- adb reboot bootloader
- fastboot oem get_identifier_token
- 使用命令生成解锁文件 signature.bin



```
D:\platform-tools_r28.0.1-windows\platform-tools>fastboot oem get_identifier_token
...
Identifier token:
30313233343536373839414243444546
URHY [ 0.100s]
finished. total time: 0.100s
```

设备序列号

图 3 获取设备序列号

如以生成解锁文件 signature.bin 为例说明(需先

cd vendor/sprd/proprieties-source/packimage_scripts)，执行如下命令，即可以生成相关设备的解锁文件(在当前目录下)。

```
./signidentifier_unlockbootloader.sh 30313233343536373839414243444546
```

```
signimage/sprd/config/rsa4096_vbmeta.pem signature.bin
```

如何生成解锁文件（需在 linux 环境下生成），使用 shell 脚本生成对应设备的解锁文件，脚本所在位置：

```
vendor/sprd/proprieties-source/packimage_scripts/signidentifier_unlockbootloader.sh
```

使用命令生成解锁文件（如命名为：signature.bin）具体格式如下：

```
signidentifier_unlockbootloader.sh <设备序列号> <证书文件> <解锁文件（bin）>
```

以某一个具体的 board 而言，在如下位置指定了证书等相关的 Config 配置：

```
device/sprd/pike2/common/security_feature.mk
```

```
#FOR Verified Boot
#1.0|2.0
PRODUCT_VBOOT := V2
BOARD_AVB_ENABLE := true
CONFIG_PATH:=vendor/sprd/proprieties-source/package_scripts/signimage/sprd/config
```

如果修改了 CONFIG_PATH，需要定制证书文件，请使用对应目录下的 rsa4096_vbmeta.pem 证书文件。

```
chunlei.liu@NJand01:~/F/vendor/sprd/proprieties-source/package_scripts/signimage/sprd/config$ ls
aeskey          rsa2048_0_pub.pem  rsa4096_boot_pub.bin  rsa4096_recovery.pem  rsa4096_vbmeta_pub.bin
aeskey_128      rsa2048_1.pem      rsa4096_modem.pem     rsa4096_recovery_pub.bin  rsa4096_vendor.pem
dynamic_ta_privatekey.pem  rsa2048_1_pub.pem  rsa4096_modem_pub.bin  rsa4096_system.pem     rsa4096_vendor_pub.bin
genkey.sh       rsa2048_devkey_pub.pem  rsa4096_product.pem  rsa4096_system_pub.bin  version.cfg
rsa2048_0.pem   rsa4096_boot.pem      rsa4096_product_pub.bin  rsa4096_vbmeta.pem
```



图 4 vbmeta 证书配置

Unisoc Confidential For hiar

4 解锁 bootloader

Android10.0 平台默认支持 Super 分区 ,所以必须先解锁设备 ,才可以烧录逻辑分区 ,才可以烧录 GSI 镜像。

解锁设备需要如下文件以及工具：

- fastboot 工具；
- signature.bin (参考 3.3)；

烧录动态分区必须先解锁 bootloader。

解锁步骤说明：

1. 烧录对应的 user/user-debug 版本，开机以备后续解锁。(user 版本需要打开—开发者选项—usbdebug 模式)，如果开机 logo 界面显示已解锁，则无需解锁，直接烧录 GSI (如何确认，请参考具体步骤第 5 步)。

2. adb reboot bootloader

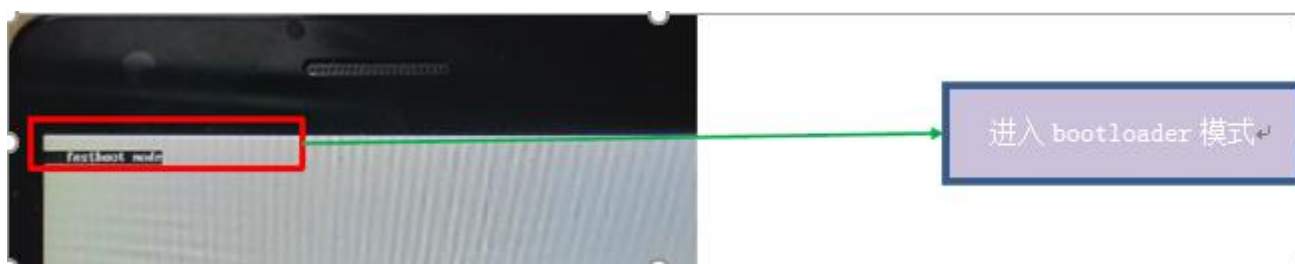


图 5 进入 bootloader 模式

3. fastboot oem get_identifier_token

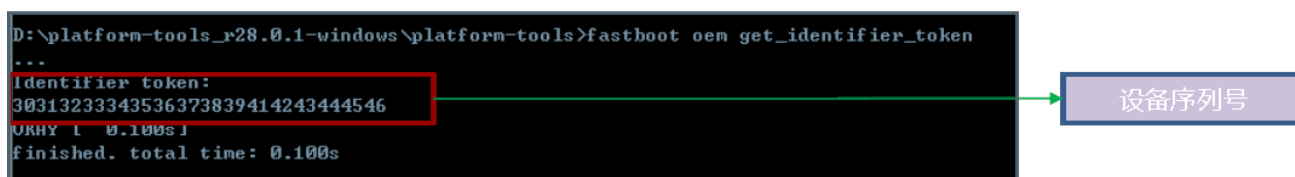


图 6 获取设备序列号

4. fastboot flashing unlock_bootloader signature.bin (参考 2.3 节生成)

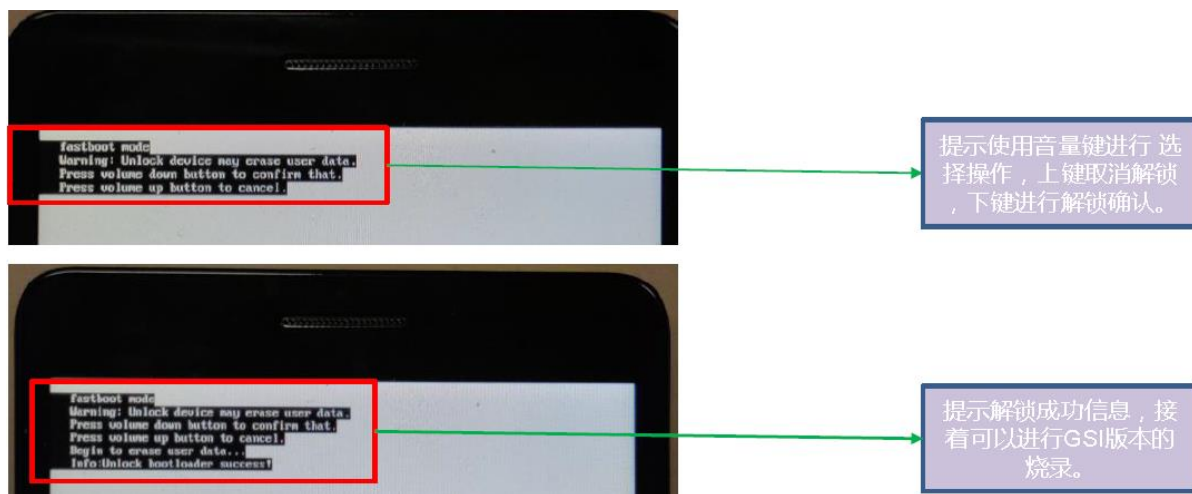


图 7 解锁 bootloader 模式

5. 重启设备确认 bootloader 是否解锁成功



图 8 确认解锁是否成功

总结：设备解锁需要执行如下步骤：

- adb reboot bootloader
- fastboot oem get_identifier_token
- fastboot flashing unlock_bootloader signature.bin

5 加锁 bootLoader

Android10.0 上某些 XTS 测试环境要求在 lock 状态下测试，故需加锁，执行如下命令即可：

- fastboot flashing lock

如下图所示：

```
C:\Users\chunlei.liu>fastboot flashing lock
OKAY [ 0.094s ]
Finished. Total time: 0.094s
```

图 9 lock bootloader 模式

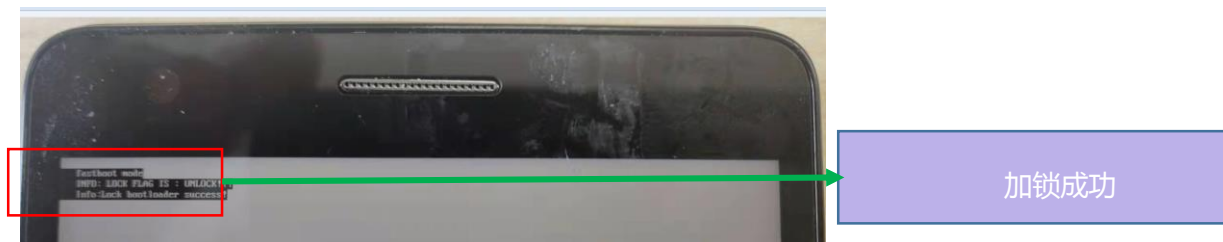


图 10 lock bootloader 模式

Unisoc Confidential For hiar

6 Android10.0 GSI XTS 测试要求介绍

6.1 Android 10.0 GSI 测试要求

XTS 测试 google 有特定的环境要求，如 CTS /CTS-On_GSI 需要在 bootloader locked 状态测试，

VTS/STS 需要在 bootloader unlock 状态下测试，具体如下表所示。

Test Suite	Test with	Build	Dynamic System Update	Debug ramdisk	Device state	Notes
CTS	OEM's system	user			Locked	
CTS-on-GSI	GSI	user	Y		Locked	
STS	OEM's system	user		Y	Unlocked	with root
VTS	GSI	user	Y	Y	unlocked	with root

图 11 xts 版本测试要求

6.2 Devices upgrading to Android 10.0

由于 Android 10.0 使用的是 user 版本的 GSI，但是升级到 Android 10.0 的设备需要搭载的 system 的类型也是不同的，需要注意，具体如下表所示。

Test Suite	Test with	Build	Notes
CTS	OEM's system	user	
STS	OEM's system	userdebug	OEM to flash OEM userdebug system
VTS	GSI	userdebug	
CTS-on-GSI	GSI	userdebug	

图 12 升级设备的测试要求

6.3 GSI build targets for Treble Compliance tests

Device type	GSI to use
Devices launching with Q	aosp_\$arch-user
Devices that launched with P and is now upgrading to Q	aosp_\$arch-userdebug
Devices that launched with O/O-MR1 and is now upgrading to Q	aosp_\$arch_ab-userdebug

图 13 Android10.0 GSI 编译类型

Android 10.0 上 GSI 必须使用 Google 正式发布的 user (signed) 版本用于 XTS 测试。

6.4 VTS 测试环境要求

VTS 测试需要 root 权限，Android10.0 上 root 权限的实现发生了变化，GSI 不再是 userdebug 的编译类型，而是 user 编译类型，root 功能只能依靠 root ramdisk 实现，也就是使用 boot-debug 镜像，同时需要解锁 bootloader，但是 cts-on-gsi 测试需要 bootloader lock 状态下进行，使用正常的 boot.img。

获取 boot-debug.img：

源码根目录 make bootimage_debug 或者全编译会生成对应镜像。

SecondStage Init 时会做 unlock 检查：

```
// See if need to load debug props to allow adb root, when the device is unlocked.
const char* force_debuggable_env = getenv("INIT_FORCE_DEBUGGABLE");
if (force_debuggable_env && AvbHandle::IsDeviceUnlocked()) {
    load_debug_prop = "true"s == force_debuggable_env;
}
```

7 下载 GSI 说明

获取 Generic System Image (google 会定期通知更新, 请选择正确的版本, 具体如下图说明, 请参考)。

网址 : https://drive.google.com/drive/u/0/folders/1_vTgPFGHVv2BTNPmKCzivvSuwKsGEY0Z

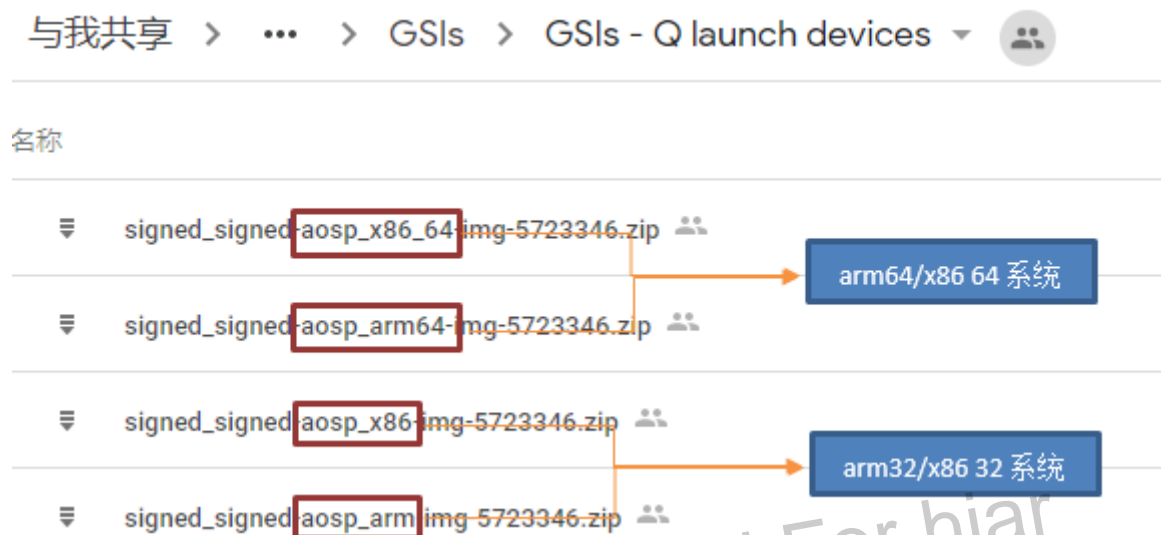


图 14 Android10.0 GSI 编译类型下载说明

8 烧录 GSI 逻辑分区

烧录和管理动态分区的前提是设备解锁，在 fastbootd 模式通过命令 fastboot getvar unlocked 可检查是否解锁（PC 端需更新 fastboot 工具）：

```
C:\Users\chunlei.liu>fastboot getvar unlocked
unlocked: yes
Finished. Total time: 0.008s
```

图 15 查看 bootloader 状态

8.1 烧录动态分区（注意 selinux 权限问题，必须先解锁 bootloader）

命令格式：fastboot flash <partition name> <filename>

烧录 system 镜像，如下图所示：

fastboot 模式下烧录 system（GSI）：

烧录 System 镜像：

```
adb reboot fastboot
```

```
fastboot flash system system.img
```

```
C:\Users\chunlei.liu>Fastboot flash system C:\Users\chunlei.liu\Downloads\signed_signed-aosp_arm64-img-5649316\system.img
Invalid sparse file format at header magic
Resizing 'system' OKAY [ 0.034s]
Sending sparse 'system' 1/3 (523608 KB) OKAY [ 24.448s]
Writing 'system' OKAY [ 13.458s]
Sending sparse 'system' 2/3 (523632 KB) OKAY [ 27.135s]
Writing 'system' OKAY [ 12.974s]
Sending sparse 'system' 3/3 (130296 KB) OKAY [ 6.239s]
Writing 'system' OKAY [ 5.680s]
Finished. Total time: 102.395s
```

图 16 Android10.0 烧录 system 逻辑分区

8.2 烧录物理分区（注意 selinux 权限问题，必须先解锁）

➤ Bootloader 模式下烧录 bootimage：

烧录 boot 镜像：

```
adb reboot bootloader
```

```
fastboot flash boot boot.img
```

```
C:\Users\chunlei.liu>fastboot flash boot D:\8.1\boot_debug\boot-debug.img
< waiting for any device >
Sending 'boot' (35840 KB)                OKAY [ 1.605s]
Writing 'boot'                          OKAY [ 1.143s]
Finished. Total time: 3.062s
```

图 17 Android10.0 烧录 boot 物理分区

Unisoc Confidential For hiar

9 GSI 开机验证说明

9.1 使用 fastboot 和 bootloader 烧录镜像验证

验证步骤：

- 设备先解锁，然后在 fastboot 模式下烧录下 GSI (system.img)。
- 设备加锁，测试 cts-on-gsi
- 设备解锁，然后在 bootloader 模式下烧录 boot_debug.img，此环节无需上锁，测试 VTS。

注意：

烧录 GSI 版本,如果需要解锁设备，重启时需要进入 recovery 模式，格式化 data 分区，否则会出现 data 分区挂载失败，导致开机异常。

格式化 data 分区有两种方式：

- 采用下载工具烧录 userdata 镜像。
- Recovery 模式下手动格式化。

进入 recovery 模式：

长按电源键+音量下键，大约 20S 左右。

9.2 直接烧录 GSI PAC 版本验证（平台维护方式）

目前，google 要求 cts-on-gsi 以及 vts 在不同的版本上测试，对应的平台上会编译出两种 PAC，需要注意这个变化。

- cts-on-gsi (boot+gsi) 不可以 root
- vts(boot-debug+gsi) 可以 root

注意：

烧录 GSI 版本,如果需要解锁设备，重启时需要进入 recovery 模式，格式化 data 分区，否则会出现 data 分区挂载失败，导致开机异常。

格式化 data 分区的方法，进入 recovery 模式进行格式化，长按电源键+音量下键进入 recovery 模式。

9.3 GSI 开机验证

- 开机
- adb shell
- getprop | grep "security"

ro.build.version.security_patch 的值是否跟烧录的 GSI 发布版本的 SPL 值一致。

```
getprop | grep "security"
[ro.build.version.security_patch]: [2018-10-05]
```

图 18 Android10.0 SPL 信息

Shared with me > ... > GSI > aosp_ar

Name ↓

system-aosp_arm64_a-2019-01-05-5140556.zip	...
system-aosp_arm64_a-2019-01-01-5140556.zip	...
system-aosp_arm64_a-2018-12-05-5140556.zip	...
system-aosp_arm64_a-2018-12-01-5140556.zip	...
system-aosp_arm64_a-2018-11-05-5140556.zip	...
system-aosp_arm64_a-2018-11-01-5140556.zip	...
system-aosp_arm64_a-2018-10-05-5140556.zip	...

Google官网下载GSI压缩包时，对应的日期就是SPL

图 19 Android10.0 GSI 下载 SPL

- getprop | grep "build id"

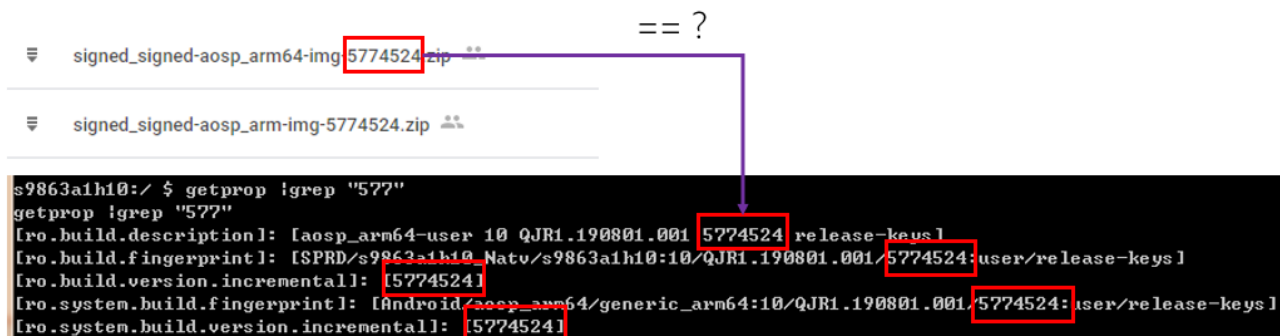


图 20 Android10.0 build number 说明

9.4 GSI 自检项

升级 GSI 需要自检一下项目：

1. SPL(security patch level) 确认 GSI 的 SPL 不能比即将升级安全补丁的 SPL 旧 ,否则 GSI rollback , 导致 lock 状态下无法开机。
2. Fastbootd 模式是否可以 reboot to system
3. Bootloader lock 状态测试 CTS-on-GSI 版本是否可以正常开机
4. Bootloader unlock 状态测试 VTS 版本是否可以正常开机
5. Bootloader unlock 状态测试 VTS 版本是否可以正常 adb root

10 GSI 分支下载以及编译说明

10.1 Android10.0 GSI 分支下载编译

GSI 分支下载办法：

```
$ repo init -u https://android.googlesource.com/platform/manifest -b android10-gsi
```

```
$ repo sync -cq
```

```
$ source build/envsetup.sh
```

```
$ lunch aosp_arm64-user
```

```
$ make -j24
```

10.2 Android10.0 GSI upload 策略说明

GSI 相关修改需要首先 upstream 到 master 分支，再由 master 分支 cherry pick 到 android10-gsi 分支。

Unisoc Confidential For hiar

11 Q&A

1) Q：若设备未解锁，是否能烧录和管理动态分区？

A：不能，会提示 “Command not available on locked devices” 。

2) Q：如何进入 fastbootd 模式？

A：进入方式有多种，可以在正常开机后执行 adb reboot fastboot 进入，也可以先进入 recovery 模式再通过菜单选择 “Enter fastboot” 进入，还可以从 bootloader 模式执行 fastboot reboot fastboot 进入。

3) Q：用什么工具烧录动态分区？

A：手机连接 PC 后，通过 PC 工具（host fastboot 可执行文件）来烧录动态分区。

注意 bootloader 模式和 fastbootd 模式共用一个 PC 工具（host fastboot），android 原生的 fastboot 输出路径在 out/host/linux-x86/bin/fastboot。

4) Q: DSU 时如何获取 raw 格式的 GSI 镜像的大小？

A: 在 linux 上执行 du -b system.img，google 目前发布的都是 raw 格式无需转换。

5) Q: DSU 需如何配置安装到 SDcard？

A: 只需插入 SDcard，系统优先安装到 SDcard，否则安装到默认的数据分区。

6) Q: 烧录 GSI 是否需要 disable avb？

A: Q 上平台默认预置了 google GSI 的 pubkey，故可以不 disable avb。