

DDR 颗粒验证流程说明

文件标识: RK-CS-YF-082

发布版本: V1.5.0

日期: 2021-04-23

文件密级: 绝密 秘密 内部资料 公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴, 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。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

对各个芯片平台 DDR 颗粒兼容性和稳定性的验证流程进行说明。文档分为 Linux 3.10, Linux 4.xx(Linux 4.4 和 Linux 4.19), RV1108, RK3308 四个章节, 请根据实际测试平台情况选择对应章节进行参考。

概述

产品版本

芯片名称	内核版本
所有芯片	所有内核版本

读者对象

本文档（本指南）主要适用于以下工程师：

品质测试工程师

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	陈有敏	2017.11.21	初版
V1.1.0	陈有敏	2018.03.22	公开资料
V1.2.0	何智欢	2018.10.11	增加 RK3308 颗粒验证流程说明
V1.3.0	陈有敏	2019.06.03	修复关于 RebootTest 和 SleepTest 的描述错误
V1.3.1	何智欢	2019.11.27	修改文档格式, 同步中英文文档
V1.4.0	陈有敏	2020.09.06	修改stressapptest和memtester描述
V1.4.1	姚旭伟	2021.04.12	增加未运行 Android 系统的设备的烤机方法
V1.5.0	姚旭伟	2021.04.23	修改文档格式

目录

DDR 颗粒验证流程说明

1. Linux 3.10 DDR 颗粒验证流程说明
 - 1.1 Linux 3.10 测试固件编译
 - 1.2 Linux 3.10 测试环境搭建
 - 1.2.1 固件烧写
 - 1.2.2 自动搭建测试环境
 - 1.2.3 手动搭建测试环境
 - 1.3 Linux 3.10 确认容量是否正确
 - 1.4 Linux 3.10 定频测试
 - 1.5 Linux 3.10 变频测试
 - 1.6 Linux 3.10 reboot 拷机
2. Linux 4.xx DDR 颗粒验证流程说明
 - 2.1 Linux 4.xx 测试固件编译
 - 2.2 Linux 4.xx 测试环境搭建
 - 2.2.1 烧写固件
 - 2.2.2 自动搭建测试的环境
 - 2.2.3 手动搭建测试的环境
 - 2.2.4 通过 U 盘和串口搭建测试环境
 - 2.3 Linux 4.xx 确认颗粒容量
 - 2.4 Linux 4.xx 定频拷机
 - 2.5 Linux 4.xx 变频拷机
 - 2.6 Linux 4.xx reboot 拷机
 - 2.6.1 Android 系统下通过计算器开启 reboot 拷机
 - 2.6.2 非 Android 系统使用 rockchip_test.sh 脚本进行 reboot 拷机
 - 2.7 Linux 4.xx sleep 拷机
 - 2.7.1 Android 系统下通过计算器开启 sleep 拷机
 - 2.7.2 非 Android 系统使用 rockchip_test.sh 脚本进行 sleep 拷机
3. RV1108 DDR 颗粒验证流程说明
 - 3.1 RV1108 测试固件编译
 - 3.2 RV1108 测试环境搭建
 - 3.3 RV1108 确认容量是否正确
 - 3.4 RV1108 定频测试
 - 3.5 RV1108 变频测试
 - 3.6 RV1108 reboot 拷机
4. RK3308 DDR 颗粒验证流程说明
 - 4.1 RK3308 确定容量是否正确
 - 4.2 RK3308 拷机测试
 - 4.3 RK3308 休眠唤醒测试
 - 4.4 RK3308 reboot 拷机

NOTE

1. RV1108、RK3308 平台 DDR 颗粒验证流程与其它平台不同，RV1108 请详见本文档的"RV1108 DDR 颗粒验证流程说明"章节；RK3308 请详见本文档的"RK3308 DDR 颗粒验证流程说明"章节。其他平台，请根据 Linux Kernel 版本选择对应章节进行参考。
2. 本文中所述颗粒验证过程需要的 DDR 测试资源文件随该文档提供。

1. Linux 3.10 DDR 颗粒验证流程说明

1.1 Linux 3.10 测试固件编译

配置 kernel 代码的 menuconfig，进入 System Type，选择打开 DDR Test 和 pm_tests。

```
menuconfig
System Type  --->
  [*] /sys/pm_tests/ support
  [*] DDR Test
```

如果 menuconfig 中没有 `[] /sys/pm_tests/ support` 选项，请参考《Rockchip-Developer-Guide-DDR-CN》的"DDR 如何定频"和"如何 enable/disable kernel 中的 DDR 变频功能"章节，分别编译定频固件和变频固件。

1.2 Linux 3.10 测试环境搭建

1.2.1 固件烧写

测试开始前，需要先明确测试过程需要的如下信息：

1. 测试固件操作系统位数（eg: 32bit or 64bit）
2. 测试机器 DDR 总容量（eg: 512MB or 1GB or 2GB ...）
3. 定频测试，DDR 要跑的最高频率（eg: 456MHz or 533MHz ...）
4. 变频测试，DDR 要跑的频率范围（eg: 200MHz - 456MHz or 200MHz-533MHz ...）

1.2.2 自动搭建测试环境

进入 DDR 测试资源文件的"linux3.10_dds_test_files"目录，直接双击 push_files.bat 脚本，根据脚本提示和固件类型信息进行选择输入 1 或者 2，自动完成测试环境搭建。自动搭建无异常，可以跳过下面的"手动搭建测试环境"这一章节。

1.2.3 手动搭建测试环境

如果自动搭建测试环境失败，可以通过手动搭建来完成。请选择"linux3.10_dds_test_files"目录里的测试文件进行安装。

1. 安装捕鱼达人APK

Note: 捕鱼达人为第三方APK，请自行获取，也可选择其他能在测试期间自动运行的APK或者循环播放视频。

```
<adb_tool> adb.exe install fishingjoy1.apk
```

2. push google stressapptest

请根据测试机器的固件为 linux 64bit 或 linux 32bit，在“static_stressapptest”目录下选择对应的 stressapptest 进行 push。

Eg:

- 如测试机器的固件为 linux 32bit，则选择对应的 stressapptest_32bit

```
<adb_tool> adb.exe push stressapptest_32bit /data/stressapptest
<adb_tool> adb.exe shell chmod 0777 /data/stressapptest
```

- 如测试机器的固件为 linux 64bit，则选择对应的 stressapptest_64bit

```
<adb_tool> adb.exe push stressapptest_64bit /data/stressapptest
<adb_tool> adb.exe shell chmod 0777 /data/stressapptest
```

3. push memtester

请根据测试机器的固件为 linux 64bit 或 linux 32bit，在“static_memtester”目录下选择对应的 memtester 进行 push。

Eg:

- 如测试机器的固件为 linux 32bit，则选择对应的 memtester_32bit

```
<adb_tool> adb.exe push memtester_32bit /data/memtester
<adb_tool> adb.exe shell chmod 0777 /data/memtester
```

- 如测试机器的固件为 linux 64bit，则选择对应的 memtester_64bit

```
<adb_tool> adb.exe push memtester_64bit /data/memtester
<adb_tool> adb.exe shell chmod 0777 /data/memtester
```

4. sync

```
<adb_tool> adb.exe shell sync
```

1.3 Linux 3.10 确认容量是否正确

通过 `<rkxxxx:/ $> cat /proc/meminfo` 查看 MemTotal 容量是否与测试机器实际容量相符。

log eg:

```
<rkxxxx:/ $> cat /proc/meminfo
MemTotal:          2038904 kB
```

512MB 约等于 533504kB

1GB 约等于 1048576kB

1.5GB 约等于 1582080kB

2GB 约等于 2097152kB

3GB 约等于 3145728kB

4GB 约等于 4194304kB

由于系统内存分配管理差异的原因，得到的容量有些偏差，属于正常。

1.4 Linux 3.10 定频测试

1. 开启捕鱼达人 APK
2. 串口控制台上输入 su 命令

```
<rkxxxx:/ $> su
```

3. DDR 定频到拷机频率

请根据测试机器所支持的 DDR 最高频率，进行设置。

Eg:

如果测试机器所支持的 DDR 最高频率为 533MHz。

```
<rkxxxx:/ #> echo set clk_ddr 533000000 > /sys/pm_tests/clk_rate
```

4. google stressapptest 拷机，拷机时间 12 小时以上

- 执行 stressapptest 程序

stressapptest 关键参数说明：

-M mbytes，指定申请测试的内存空间大小，单位为MB。一般申请总容量的八分之一进行测试，如果总容量是 1GB 则申请 128MB 进行测试，如果总容量是 2GB 则申请 256MB 进行测试。

-s seconds，指定测试运行时间，单位是秒。运行时间12小时则参数为43200。

Eg:

内存总容量 1GB 则申请 128MB 进行 stressapptest，运行时间12小时(43200秒)，执行命令如下：

```
<rkxxxx:/ #> /data/stressapptest -s 43200 -i 4 -C 4 -W --stop_on_errors -M 128
```

- 确认拷机结果

拷机结束，确认机器是否正常，捕鱼达人是否正常运行，stressapptest 结果是 PASS 还是 FAIL。stressapptest 每隔 10 秒会打印一条 log，log 显示测试剩余时间。测试完成后会打印测试结果，如果测试通过打印 "Status: PASS - please verify no corrected errors"，如果测试失败打印 "Status: FAIL - test discovered HW problems"。

6. memtester 拷机，拷机时间 memtester 12 小时以上

- 执行 memtester 程序

memtester 关键参数说明：

<mem>m，指定申请测试的内存空间大小，单位为 MB。一般申请总容量的八分之一进行测试，如果总容量是 1GB 则申请 128MB 进行测试，如果总容量是 2GB 则申请 256MB 进行测试。

Eg:

内存总容量 1GB 则申请 128MB 进行 memtester，执行命令如下：

```
<rkxxxx:/ #> /data/memtester 128m
```

- 确认拷机结果

拷机结束，确认机器是否正常，捕鱼达人是否正常运行，memtester 是否在正常运行。DDR 测试资源文件目录里的 memtester 程序进行过修改，测试过程如果有发现错误会自动停止测试，如果持续测试 12 小时以上后，memtester 仍在继续运行，说明测试过程没有发现错误。

- memtester 运行过程如果没有发现错误，会持续打印如下 log:

```
Loop 10:
  Stuck Address      : ok
  Random Value      : ok
  Compare XOR       : ok
  Compare SUB       : ok
  Compare MUL       : ok
  Compare DIV       : ok
  Compare OR        : ok
  Compare AND       : ok
  Sequential Increment: ok
  Solid Bits        : ok
  Block Sequential  : ok
  Checkerboard      : ok
  Bit Spread        : ok
  Bit Flip          : ok
  Walking Ones      : ok
  Walking Zeroes    : ok
```

- memtester 运行过程如果有发现错误，会自动停止测试并退出，退出时打印如下 log:

```
FAILURE: 0xffffffff != 0xffffbfff at offset 0x03b7d9e4.
EXIT_FAIL_OTHERTEST
```

1.5 Linux 3.10 变频测试

如果机器前面做过定频测试，要重启机器，否则后续的变频命令会无法进行。

1. 开启捕鱼达人 APK
2. 串口控制台上输入 su 命令

```
<rkxxxx:/ $> su
```

3. 后台执行 memtester

memtester 关键参数说明：

<mem>m, 指定申请测试的内存空间大小，单位为MB。一般申请总容量的八分之一进行测试，如果总容量是 1GB 则申请 128MB 进行测试，如果总容量是 2GB 则申请 256MB 进行测试。

Eg:

如果总容量是 1GB 则申请 128MB 进行 memtester，执行命令如下：


```
<rkxxxx:/ #> /data/memtester 128m > /data/memtester_log.txt &
```

4. 执行变频命令

根据测试机器支持的 DDR 频率范围进行设置。

Eg:

如果测试机器支持的 DDR 频率范围为 200MHz 到 533MHz。

```
<rkxxxx:/ #> echo 'a:200M-533M-1000000T' > proc/driver/ddr_ts
```

Note: 变频测试测试过程，由于是强制变频，机器可能会出现由于带宽不足等原因所致的屏幕闪烁等现象，属正常现象。

5. 确认拷机结果，拷机时间 12 小时以上

- 确认捕鱼达人是否正常运行，机器是否正常
- 确认变频程序运行是否正常，变频 log 是否在正常打印
- 确认 memtester 是否正常运行

在串口输入 `<rkxxxx:/ #> ps | grep memtester`，看 memtester 进程是否存在。

Eg:

```
<rkxxxx:/ #> ps | grep memtester
root      14309 1730  74332  68156          0 5e980bf564 R /data/memtester
```

1.6 Linux 3.10 reboot 拷机

开启计算器 Calculator，输入 "83991906="，点击"RebootTest"，拷机时间 12 小时以上。

2. Linux 4.xx DDR 颗粒验证流程说明

2.1 Linux 4.xx 测试固件编译

使能 DDR 变频功能，打开测试机器对应的板级 DTS 文件，找到 dfi 和 dmc 节点，配置 status = "okay"。

```
&dfi {
    status = "okay";
};

&dmc {
    status = "okay";
    .....
};
```

关于测试固件编译这里只做简单说明，详细介绍请参考《Rockchip-Developer-Guide-DDR-CN》的"如何 enable/disable kernel 中的 DDR 变频功能"章节。

2.2 Linux 4.xx 测试环境搭建

2.2.1 烧写固件

测试开始前，需要先明确测试过程需要的如下信息：

1. 测试固件操作系统位数（32bit or 64bit）
2. 测试机器 DDR 总容量（512MB or 1GB or 2GB ...）
3. 定频测试，DDR 要跑的最高频率（456MHz or 533MHz ...）

2.2.2 自动搭建测试的环境

1. 进入 DDR 测试资源文件的"linux4.xx_ddr_test_files"目录，双击 push_files.bat 脚本文件，根据脚本提示和固件类型信息进行选择输入 1 或者 2，自动完成测试环境搭建。

Note: 运行脚本后，需要通过打印的 log 检查是否每项都有被正常执行，确认是否有报错信息。

2. 自动搭建无异常，可以跳过下面的"手动搭建测试环境"这一章节。

2.2.3 手动搭建测试的环境

如果自动搭建测试环境失败，可以通过手动搭建来完成。请选择"linux4.xx_ddr_test_files"目录里的测试文件进行安装。

1. 安装捕鱼达人APK

Note: 捕鱼达人为第三方 APK，请自行获取，也可选择其他能在测试期间自动运行的APK或者循环播放视频。未运行 Android 系统的设备请跳过此步骤。

```
<adb_tool> adb.exe install fishingjoy1.apk
```

2. push google stressapptest

请根据测试机器的固件为 linux 64bit 或 linux 32bit，在“static_stressapptest”目录下选择对应的 stressapptest 进行 push。

Eg:

- 如测试机器的固件为 linux 32bit，则选择对应的 stressapptest_32bit

```
<adb_tool> adb.exe push stressapptest_32bit /data/stressapptest  
<adb_tool> adb.exe shell chmod 0777 /data/stressapptest
```

- 如测试机器的固件为 linux 64bit，则选择对应的 stressapptest_64bit

```
<adb_tool> adb.exe push stressapptest_64bit /data/stressapptest  
<adb_tool> adb.exe shell chmod 0777 /data/stressapptest
```

3. push memtester

请根据测试机器的固件为 linux 64bit 或 linux 32bit，在“static_memtester”目录下选择对应的 memtester 进行 push。

Eg:

- 如测试机器的固件为 linux 32bit，则选择对应的 memtester_32bit

```
<adb_tool> adb.exe push memtester_32bit /data/memtester  
<adb_tool> adb.exe shell chmod 0777 /data/memtester
```

- 如测试机器的固件为 linux 64bit，则选择对应的 memtester_64bit

```
<adb_tool> adb.exe push memtester_64bit /data/memtester  
<adb_tool> adb.exe shell chmod 0777 /data/memtester
```

4. push ddr_freq_scan.sh 脚本

```
<adb_tool> adb.exe push ddr_freq_scan.sh /data/ddr_freq_scan.sh  
<adb_tool> adb.exe shell chmod 0777 /data/ddr_freq_scan.sh
```

5. sync

```
<adb_tool> adb.exe shell sync
```

2.2.4 通过 U 盘和串口搭建测试环境

如果测试机器的 ADB 无法连接，可以通过 U 盘将测试过程需要用到文件拷贝到测试板，然后通过串口搭建测试环境。

1. 准备工作

- 开机后，添加 `wake_lock` 防止机器进入二级待机 `echo 1 > /sys/power/wake_lock`，或者通过设置 `Setting->Display->Sleep->Never sleep` 让机器保持唤醒状态
- U 盘接入电脑，将 "linux4.xx_ddr_test_files"，"static_memtester" 和 "static_stressapptest" 目录拷贝到 U 盘
- 测试板串口控制台上输入 `su` 命令

```
<rk3399:/ $> su
```

- U 盘接入测试板，将 "linux4.xx_ddr_test_files"，"static_memtester" 和 "static_stressapptest" 目录拷贝到机器 `/data` 目录。U 盘正常是加载在 `/mnt/media_rw/***` 目录下 (***: 每块 U 盘的挂载节点名有不同，请用 `tab` 键补全)。

Eg:

如果U盘的挂载节点为 `/mnt/media_rw/B4FE-5315`，执行命令如下：

```
<rk3399:/ #> cp -rf /mnt/media_rw/B4FE-5315/linux4.xx_ddr_test_files /data/
<rk3399:/ #> cp -rf /mnt/media_rw/B4FE-5315/static_memtester /data/
<rk3399:/ #> cp -rf /mnt/media_rw/B4FE-5315/static_stressapptest /data/
```

2. 自动搭建测试环境

- 如测试机器的固件为 `linux 32bit`，则选择对应的 `test_files_install_32bit.sh` 脚本

```
<rk3399:/ #> chmod 0777
/data/linux4.xx_ddr_test_files/test_files_install_32bit.sh
<rk3399:/ #> /data/linux4.xx_ddr_test_files/test_files_install_32bit.sh
```

- 如测试机器的固件为 `linux 64bit`，则选择对应的 `test_files_install_64bit.sh` 脚本

```
<rk3399:/ #> chmod 0777
/data/linux4.xx_ddr_test_files/test_files_install_64bit.sh
<rk3399:/ #> /data/linux4.xx_ddr_test_files/test_files_install_64bit.sh
```

需要通过打印的 `log` 检查是否每项都有被正常执行。自动搭建无异常，可以跳过下面的"手动搭建测试环境"这一部分。

3. 手动搭建测试环境

如果自动搭建测试环境失败，可以通过手动搭建来完成。

- 如测试机器的固件为 `linux 32bit`，则拷贝对应的测试所需文件

```
<rk3399:/ #> cp /data/static_stressapptest/stressapptest_32bit
/data/stressapptest
<rk3399:/ #> cp /data/static_memtester/memtester_32bit /data/memtester
<rk3399:/ #> cp /data/linux4.xx_ddr_test_files/ddr_freq_scan.sh
/data/ddr_freq_scan.sh
```

- 如测试机器的固件为 `linux 64bit`，则拷贝对应的测试所需文件

```
<rk3399:/ #> cp /data/static_stressapptest/stressapptest_64bit
/data/stressapptest
<rk3399:/ #> cp /data/static_memtester/memtester_64bit /data/memtester
<rk3399:/ #> cp /data/linux4.xx_ddr_test_files/ddr_freq_scan.sh
/data/ddr_freq_scan.sh
```

- 更改文件权限

```
<rk3399:/ #> chmod 777 /data/memtester /data/stressapptest /data/ddr_freq_scan.sh
```

- 安装捕鱼达人 APK

Note: 捕鱼达人为第三方APK，请自行获取，也可选择其他能在测试期间自动运行的APK或者循环播放视频。未运行 Android 系统的设备请跳过此步骤。

```
<rk3399:/ #> pm install /data/fishingjoy1.apk
```

- sync

```
<rk3399:/ #> sync
```

2.3 Linux 4.xx 确认颗粒容量

通过 `<rkxxxx:/ #> cat /proc/meminfo` 查看 MemTotal 项所示容量是否与测试机器 DDR 总容量一致。

log eg:

```
rkxxxx:/ # cat /proc/meminfo
MemTotal:          2038904 kB
```

512MB 约等于 533504kB

1GB 约等于 1048576kB

1.5GB 约等于 1582080kB

2GB 约等于 2097152kB

3GB 约等于 3145728kB

4GB 约等于 4194304kB

由于系统内存分配管理差异的原因，得到的容量有些偏差，属于正常。

2.4 Linux 4.xx 定频拷机

1. 开启捕鱼达人 APK（未运行 Android 系统的设备请跳过此步骤）
2. 先输入 su 命令

```
<rkxxxx:/ $> su
```

3. 定频到拷机频率

根据测试机器 DDR 要跑的最高频率进行设置。

Eg:

- 如果是跑 928MHz

```
<rkxxxx:/ #> /data/ddr_freq_scan.sh 933000000
```

- 如果是跑 800MHz

```
<rkxxxx:/ #> /data/ddr_freq_scan.sh 800000000
```

- 如果是跑 600MHz

```
<rkxxxx:/ #> /data/ddr_freq_scan.sh 600000000
```

4. 通过 log 确认频率是否正确

log eg:

```
130|rkxxxx:/ # /data/ddr_freq_scan.sh 800000000  
already change to 800000000 done.  
change frequency to available max frequency done.
```

5. google stressapptest 拷机, 拷机时间 12 小时以上

- 执行 stressapptest 程序

stressapptest 关键参数说明:

-M mbytes, 指定申请测试的内存空间大小, 单位为MB。一般申请总容量的八分之一进行测试, 如果总容量是 1GB 则申请 128MB 进行测试, 如果总容量是 2GB 则申请 256MB 进行测试。

-s seconds, 指定测试运行时间, 单位是秒。运行时间12小时则参数为43200。

Eg:

内存总容量 1GB 则申请 128MB 进行 stressapptest, 运行时间12小时(43200秒), 执行命令如下:

```
<rkxxxx:/ #> /data/stressapptest -s 43200 -i 4 -C 4 -W --stop_on_errors -M 128
```

- 确认拷机结果

拷机结束, 确认机器是否正常, 捕鱼达人是否正常运行, stressapptest 结果是 PASS 还是 FAIL。

stressapptest 每隔 10 秒会打印一条 log, log 显示测试剩余时间。测试完成后会打印测试结果, 如果测试通过打印 Status: PASS - please verify no corrected errors, 如果测试失败打印 Status: FAIL - test discovered HW problems。

6. memtester 拷机, 拷机时间 12 小时以上

- 执行 memtester 程序

memtester 关键参数说明:

<mem>m, 指定申请测试的内存空间大小, 单位为MB。一般申请总容量的八分之一进行测试, 如果总容量是 1GB 则申请 128MB 进行测试, 如果总容量是 2GB 则申请 256MB 进行测试。

Eg:

内存总容量 1GB 则申请 128MB 进行 memtester，执行命令如下：

```
<rkxxxx:/ #> /data/memtester 128m
```

- 确认拷机结果

拷机结束，确认机器是否正常，捕鱼达人是否正常运行，memtester 是否在正常运行。DDR 测试资源文件目录里的 memtester 程序进行过修改，测试过程如果有发现错误会自动停止测试，如果持续测试 12 小时以上后，memtester 仍在继续运行，说明测试过程没有发现错误。

- memtester 运行过程如果没有发现错误，会持续打印如下 log:

```
Loop 10:
  Stuck Address      : ok
  Random Value       : ok
  Compare XOR        : ok
  Compare SUB        : ok
  Compare MUL        : ok
  Compare DIV        : ok
  Compare OR         : ok
  Compare AND        : ok
  Sequential Increment: ok
  Solid Bits         : ok
  Block Sequential   : ok
  Checkerboard       : ok
  Bit Spread         : ok
  Bit Flip           : ok
  Walking Ones       : ok
  Walking Zeroes    : ok
```

- memtester 运行过程如果有发现错误，会自动停止测试并退出，退出时打印如下 log:

```
FAILURE: 0xffffffff != 0xffffbfff at offset 0x03b7d9e4.
EXIT_FAIL_OTHERTEST
```

2.5 Linux 4.xx 变频拷机

1. 开启捕鱼达人 APK（未运行 Android 系统的设备请跳过此步骤）
2. 先输入 su 命令

```
<rkxxxx:/ $> su
```

3. 后台执行 memtester

memtester 关键参数说明：

<mem>m, 指定申请测试的内存空间大小，单位为MB。一般申请总容量的八分之一进行测试，如果总容量是 1GB 则申请 128MB 进行测试，如果总容量是 2GB 则申请 256MB 进行测试。

Eg:

如果总容量是 1GB 则申请 128MB 进行 memtester，执行命令如下：

```
<rkxxxx:/ #> /data/memtester 128m > /data/memtester_log.txt &
```

4. 执行测试脚本

```
<rkxxxx:/ #> /data/ddr_freq_scan.sh
```

Note: 变频测试测试过程，由于是强制变频，机器可能会出现由于带宽不足等原因所致的屏幕闪烁等现象，属正常现象。

5. 确认拷机结果，拷机时间 12 小时以上

- 确认捕鱼达人是否正常运行，机器是否正常
- 确认 memtester 是否正常运行

在串口输入 `<rkxxxx:/ #> ps | grep memtester`，看 memtester 进程是否存在。

Eg:

```
<rkxxxx:/ #> ps | grep memtester
root      14309 1730  74332  68156          0 5e980bf564 R /data/memtester
```

- 确认变频脚本运行是否正常运行，变频 log 是否在正常打印

log eg:

```
DDR freq will change to 600000000 8786
already change to 600000000 done
DDR freq will change to 800000000 8787
already change to 800000000 done
DDR freq will change to 200000000 8788
already change to 200000000 done
```

2.6 Linux 4.xx reboot 拷机

2.6.1 Android 系统下通过计算器开启 reboot 拷机

==为防止做 reboot 过程机器进入休眠，影响测试，请通过设置 setting->security->set screen lock->None，让机器一开机跳过锁屏界面，直接进入主界面。同时通过设置 Setting->Display->Sleep->Never sleep 让机器保持唤醒状态。==

开启计算器 Calculator，输入 "83991906="，点击"RebootTest"，拷机时间 12 小时以上。

2.6.2 非 Android 系统使用 rockchip_test.sh 脚本进行 reboot 拷机

1. 输入 su 命令

```
[root@RV1126_RV1109:/]# su
```

2. 运行 /rockchip_test/rockchip_test.sh，选择 auto reboot test 对应的数字


```
[root@RV1126_RV1109:/]# /rockchip_test/rockchip_test.sh
```

```
[root@RV1126_RV1109:/]# /rockchip_test/rockchip_test.sh
*****
***                                     ***
***          *****                  ***
***      *ROCKCHIPS TEST TOOLS*      ***
***          *                        *   ***
***          *****                  ***
***                                     ***
*****
*****
ddr test :                1 (memtester & stressapptest)
cpufreq test:            2 (cpufreq stresstest)
flash stress test:      3
bluetooth test:         4 (bluetooth on&off test)
audio test:             5
recovery test:          6 (default wipe all)
suspend_resume test:    7 (suspend & resume)
wifi test:              8
ethernet test:          9
auto reboot test:      10
ddr freq scaling test  11
npu stress test        12
camera test            13 (use rkisp_demo)
video test             14 (use gstreamer-wayland and app_demo)
gpu test               15 (use glmark2)
chromium test          16 (chromium with video hardware acceleration)
*****
please input your test moudle:
10
```

3. 拷机 12 小时以上，确认设备是否正常。可以通过以下命令关闭 reboot test

```
[root@RV1126_RV1109:/]# echo off > /data/cfg/rockchip_test/reboot_cnt
```

2.7 Linux 4.xx sleep 拷机

RK3399 使用 LPDDR4 时，必须进行这项测试；其他 DDR 类型及其他平台是 optional。

2.7.1 Android 系统下通过计算器开启 sleep 拷机

拔掉连接 ADB 的 USB 线，开启计算器 Calculator,输入 "83991906=",点击"SleepTest", 拷机时间 12 小时以上。

2.7.2 非 Android 系统使用 rockchip_test.sh 脚本进行 sleep 拷机

1. 输入 su 命令

```
[root@RV1126_RV1109:/]# su
```

2. 运行 /rockchip_test/rockchip_test.sh，选择 suspend_resume test 对应的数字，再选择 auto suspend (resume by rtc) 对应的数字

```
[root@RV1126_RV1109:/]# /rockchip_test/rockchip_test.sh
```

```
[root@RV1126_RV1109:/]# /rockchip_test/rockchip_test.sh
*****
***                                     ***
***          *****                   ***
***      *ROCKCHIPS TEST TOOLS*       ***
***          *             *           ***
***          *****                   ***
***                                     ***
*****
*****
ddr test :                1 (memtester & stressapptest)
cpufreq test:            2 (cpufreq stresstest)
flash stress test:      3
bluetooth test:         4 (bluetooth on&off test)
audio test:             5
recovery test:          6 (default wipe all)
suspend_resume test:    7 (suspend & resume)
wifi test:              8
ethernet test:          9
auto reboot test:      10
ddr freq scaling test  11
npu stress test        12
camera test            13 (use rkisp_demo)
video test             14 (use gstreamer-wayland and app_demo)
gpu test               15 (use glmark2)
chromium test          16 (chromium with video hardware acceleration)
*****
please input your test moudle:
7 ←
*****
auto suspend:           1
suspend (resume by key): 2
auto suspend (resume by rtc): 3
*****
3 ←
```

3. 拷机 12 小时以上，确认设备是否正常

3. RV1108 DDR 颗粒验证流程说明

3.1 RV1108 测试固件编译

打开 DDR Test 和 pm_tests 配置。配置 kernel 代码的 menuconfig，进入 System Type，选择打开 DDR Test 和 pm_tests。

```
menuconfig
System Type --->
[*] /sys/pm_tests/ support
[*] DDR Test
```

如果 menuconfig 中没有 `[] /sys/pm_tests/ support` 选项，请参考《Rockchip-Developer-Guide-DDR-CN》的"DDR 如何定频"和"如何 enable/disable kernel 中的 DDR 变频功能"章节，分别编译定频固件和变频固件。

3.2 RV1108 测试环境搭建

将"rv1108_dds_test_files"目录下的 stressapptest 和 memtester_32bit 考到 SD 卡根目录，把卡插到测试板上。

3.3 RV1108 确认容量是否正确

通过 `<rv1108:/ #> cat /proc/meminfo` 查看 MemTotal 项所示容量是否与测试机器 DDR 总容量相符。

log eg:

```
<rv1108:/ #> cat /proc/meminfo
MemTotal:      133376 kB
```

64MB 约等于 66688kB

128MB 约等于 133376kB

256MB 约等于 266752kB

512MB 约等于 533504kB

由于系统内存分配管理差异的原因，得到的容量有些偏差，属于正常。

3.4 RV1108 定频测试

1. DDR 定频到拷机频率

请根据测试机器要跑的 DDR 最高频率，进行设置。

Eg:

如果测试机器要跑的 DDR 最高频率为 800MHz。

```
<rv1108:/ #> echo set clk_dds 800000000 > /sys/pm_tests/clk_rate
```

2. google stressapptest 拷机，拷机时间 12 小时以上

如果总容量是 128MB 则申请 16MB 进行 stressapptest，一般是总容量的八分之一

```
<rv1108:/ #> /mnt/sdcard/stressapptest -s 500 -i 1 -C 1 -W --stop_on_errors -M 16
```

3. 确认拷机结果

拷机结束，确认机器是否正常，stressapptest 结果是 PASS 还是 FAIL。stressapptest 每隔 10 秒会打印一条 log，log 显示测试剩余时间。测试完成后会打印测试结果，如果测试通过打印 Status: PASS，如果测试失败打印 Status: FAIL。

4. memtester 拷机，拷机时间 12 小时以上

如果总容量是 128MB 则申请 16MB 进行 memtester，一般是总容量的八分之一

```
<rv1108:/ #> /mnt/sdcard/memtester_32bit 16m
```

5. 确认拷机结果

拷机结束，确认机器是否正常，memtester 是否正常运行。"DDR 测试资源文件"目录里的 memtester 程序进行过修改，测试过程如果有发现错误会自动停止测试，如果持续测试 12H+后，memtester 仍在继续运行，说明测试过程没有发现错误。

- memtester 运行过程如果没有发现错误，会持续打印如下 log:

```
Loop 10:
  Stuck Address      : ok
  Random Value       : ok
  Compare XOR        : ok
  Compare SUB        : ok
  Compare MUL        : ok
  Compare DIV        : ok
  Compare OR         : ok
  Compare AND        : ok
  Sequential Increment: ok
  Solid Bits         : ok
  Block Sequential   : ok
  Checkerboard       : ok
  Bit Spread         : ok
  Bit Flip           : ok
  Walking Ones       : ok
  Walking Zeroes     : ok
```

- memtester 运行过程如果有发现错误，会自动停止测试并退出，退出时打印如下 log:

```
FAILURE: 0xffffffff != 0xffffbfff at offset 0x03b7d9e4.  
EXIT_FAIL_OTHERTEST
```

3.5 RV1108 变频测试

如果机器前面做过定频测试，要重启机器，否则后续的变频命令会无法进行。

1. 后台执行 memtester

如果总容量是 128MB 则申请 16MB 进行 memtester，一般是总容量的十六分之一

```
<rv1108:/ #> /mnt/sdcard/memtester_32bit 16m > /data/memtester_log.txt &
```

2. 执行变频命令

变频测试频率范围为 400MHz 到测试机器要跑的 DDR 最高频率之间进行。

Eg:

如果测试机器 DDR 要跑的最高频率为 800MHz:

```
<rv1108:/ #> echo 'a:400M-800M-1000000T' > proc/driver/ddr_ts
```

3. 确认拷机结果，拷机时间 12 小时以上

- 确认机器是否正常
- 确认变频程序运行是否正常，变频 log 是否在正常打印
- 确认 memtester 是否正常运行

在串口输入 `<rkxxxx:/ #> ps | grep memtester`，看 memtester 进程是否存在。

Eg:

```
<rkxxxx:/ #> ps | grep memtester  
root      14309 1730  74332 68156          0 5e980bf564 R /data/memtester
```

3.6 RV1108 reboot 拷机

可用 1108 自带 reboot 功能，menu -> debug -> reboot test。

4. RK3308 DDR 颗粒验证流程说明

4.1 RK3308 确定容量是否正确

通过 `<rk3308:/ #> cat /proc/meminfo` 查看 MemTotal 项所示容量是否与测试机器 DDR 总容量相符。

log eg:

```
<rk3308:/ #> cat /proc/meminfo
MemTotal:      246832 kB
MemFree:       201800 kB
```

64MB 约等于 66688kB

128MB 约等于 133376kB

256MB 约等于 266752kB

512MB 约等于 533504kB

由于系统内存分配管理差异的原因，得到的容量有些偏差，属于正常。

4.2 RK3308 拷机测试

由于 3308 当前不支持变频功能，ddr 频率由 loader 初始化期间设置的频率，到后面是不会修改 ddr 频率的。DDR 颗粒测试请使用最高频率的设置，即 DDR3 请使用 800MHz 的 loader，DDR2 和 LPDDR2 请使用 533MHz 的 loader。

1. memtester 拷机，拷机时间 12 小时以上

首先要确认测试文件是否存在：

```
<rk3308:/ #> ls usr/bin/memtester
usr/bin/memtester
```

- 如果有测试文件，memtester 测试命令如下：（如果总容量是 128MB 则申请 16MB 进行 memtester，一般是总容量的八分之一。）

```
<rk3308:/ #> memtester 16m > /data/memtester_log.txt &
```

- 如果没有测试文件，则请将随此文件一同发布的 memtester_32bit.32bit（或 memtester_64bit.64bit）文件通过 adb push 到 data 分区：（memtester_32bit.32bit 适用于 32 位系统，memtester_64bit.64bit 适用于 64 位系统）

32 位系统命令：

```
adb push \*文件路径*\memtester_32bit.32bit data/memtester
```

64 位系统命令:

```
adb push \*文件路径*\memtester_64bit.64bit data/memtester
```

将 memtester_32bit.32bit 文件修改权限为可执行:

```
<rk3308:/ #> chmod 777 /data/memtester
```

memtester 测试命令如下 (如果总容量是 128MB 则申请 16MB 进行 memtester, 一般是总容量的八分之一):

```
<rk3308:/ #> /data/memtester 16m > /data/memtester_log.txt &
```

2. 确认拷机结果

- 拷机结束, 确认机器是否正常。
- 确认 memtester 打印是否正常: (注意, memtester 出错=不会=停止测试, 需要查看所有打印是否正确)

正确打印:

```
Loop 1:
  Stuck Address      : ok
  Random Value       : ok
  Compare XOR        : ok
  Compare SUB        : ok
  Compare MUL        : ok
  Compare DIV        : ok
  Compare OR         : ok
  Compare AND        : ok
  Sequential Increment: ok
  Solid Bits         : ok
  Block Sequential   : ok
  Checkerboard       : ok
  Bit Spread         : ok
  Bit Flip           : ok
  Walking Ones       : ok
  Walking Zeroes    : ok
  8-bit Writes      : ok
  16-bit Writes     : ok
```

出错打印:

```
Loop 92:
  Stuck Address      : ok
  Random Value       : FAILURE: 0x37fe0f4190f6b999 != 0x37fe0f4196f6b999 at
offset 0x0027a958.
FAILURE: 0x2823d0d6f62a4b01 != 0x2823d0d6f02a4b01 at offset 0x0027a958.
  Compare XOR        : FAILURE: 0x4c4f418e764340e8 != 0x4c4f418e704340e8 at
offset 0x0027a958.
  Compare SUB        : FAILURE: 0x2856fb8ee22bd230 != 0xfe5ee503ee2bd230 at
offset 0x0027a958.
  Compare MUL        : FAILURE: 0x00000000 != 0x00000001 at offset 0x0027a958.
```

```
Compare DIV      : FAILURE: 0xefb2eceaffbf9604 != 0xefb2eceaffbf9605 at
offset 0x0027a958.
Compare OR       : FAILURE: 0xcbb2a0e8cfbb8200 != 0xcbb2a0e8cfbb8201 at
offset 0x0027a958.
Compare AND      : Sequential Increment: ok
Solid Bits       : ok
Block Sequential : ok
Checkerboard     : ok
Bit Spread       : ok
Bit Flip         : ok
Walking Ones    : ok
Walking Zeroes  : ok
8-bit Writes    : ok
16-bit Writes   : ok
```

3. google stressapptest 拷机，拷机时间 12 小时以上

首先要确认测试文件是否存在：

```
<rk3308:/ #> ls usr/bin/stressapptest
usr/bin/stressapptest
```

- 如果有测试文件，stressapptest 测试命令如下：（如果总容量是 256MB 则申请 32MB 进行 stressapptest，一般是总容量的八分之一。时间设置由-s 后面的参数控制，单位是秒。如下为拷机 24 小时的命令）

```
<rk3308:/ #> stressapptest -s 86400 -i 4 -C 4 -W --stop_on_errors -M 32
```

- 如果没有测试文件，则请将随此文件一同发布的 stressapptest_32bit（或 stressapptest_64bit）文件通过 adb push 到 data 分区（stressapptest_32bit 适用于 32 位系统，stressapptest_64bit 适用于 64 位系统）：

32 位系统命令：

```
adb push \*文件路径*\stressapptest_32bit data/stressapptest
```

64 位系统命令：

```
adb push \*文件路径*\stressapptest_64bit data/stressapptest
```

将 stressapptest_32bit 文件修改权限为可执行：

```
<rk3308:/ #> chmod 777 /data/stressapptest
```

stressapptest 测试命令如下（如果总容量是 256MB 则申请 32MB 进行 stressapptest，一般是总容量的八分之一。时间设置由-s 后面的参数控制，单位是秒。如下为拷机 24 小时的命令）：

```
<rk3308:/ #> /data/stressapptest -s 86400 -i 4 -C 4 -W --stop_on_errors -M 32
```

4. 确认拷机结果

- 拷机结束，确认机器是否正常。

- stressapptest 结果是 PASS 还是 FAIL。stressapptest 每隔 10 秒会打印一条 log，log 显示测试剩余时间。测试完成后会打印测试结果，如果测试通过打印 Status: PASS，如果测试失败打印 Status: FAIL。

4.3 RK3308 休眠唤醒测试

休眠唤醒需要 kernel 使能自动唤醒功能。打开 rk3308.dtsi 文件，找到休眠唤醒节点 rockchip_suspend,位或上 RKPM_TIMEOUT_WAKEUP_EN，使能自动唤醒：

```
rockchip_suspend: rockchip-suspend {
    ...
    rockchip,wakeup-config = <
        (0
        | RKPM_GPIO0_WAKEUP_EN
        | RKPM_TIMEOUT_WAKEUP_EN
        )
    >;
};
```

编译好固件后，建议使用 3308 测试脚本的休眠唤醒测试。首先要确认测试文件是否存在：

```
<rk3308:/ #> ls rockchip_test/rockchip_test.sh
rockchip_test/rockchip_test.sh
```

- 若有测试文件，则休眠自动唤醒测试命令如下：

```
<rk3308:/ #> /rockchip_test/rockchip_test.sh
...
please input your test mouldle: //串口先输入8<enter>,再输入1<enter>
8
1
```

- 若没有测试文件，可以在串口命令行直接输入命令进行休眠唤醒测试，命令如下：

```
<rk3308:/ #> while true; do echo mem > /sys/power/state; sleep 5; done
```

拷机时间 12h+，确认机器是否正常。

4.4 RK3308 reboot 拷机

建议使用 3308 测试脚本的 reboot 测试命令。首先要确认测试文件是否存在：

```
<rk3308:/ #> ls rockchip_test/rockchip_test.sh
rockchip_test/rockchip_test.sh
```

- 若有测试文件，reboot 测试命令如下：

```
<rk3308:/ #> /rockchip_test/rockchip_test.sh
...
please input your test moudle: //串口输入13<enter>
13
```

- 若无测试文件，则请将随此文件一同发布的 auto_reboot_test.sh 文件通过 adb push 到 data 分区：

```
adb push \*文件路径*\auto_reboot_test.sh data/.
```

将 auto_reboot_test.sh 文件修改权限为可执行：

```
<rk3308:/ #> chmod 777 /data/auto_reboot_test.sh
```

reboot 测试命令如下：

```
<rk3308:/ #> /data/auto_reboot_test.sh
```

拷机时间 12h+，确认机器是否正常。