Android 15 DDR DVFS
开发指南

版本号: 1.0
发布日期: 2024.12.24

版本历史

版本号	日期	制/修订人	内容描述
1.0	2024.12.24	AWA2099	1. 增加 A733 平台说明

目　录

1 概述

1.1 编写目的

本文主要讲解如何使用 DDR DVFS，以及常用的 DVFS 的 debug 手段。

1.2 适用范围

适用于全志 A733 Android 15 平台。

1.3 目标读者

需要对 DDR DVFS 进行开发，适配及 debug 的开发人员。

2 DDR DVFS 原理介绍

DDR DVFS，dram 动态调频框架，可以基于当前带宽负载动态调节 DDR 频率，可以有效降低 dram 的功耗。

2.1 基础框架

当前的 dram 调频是通过 CPUs 来完成的，其基本的框架如下。

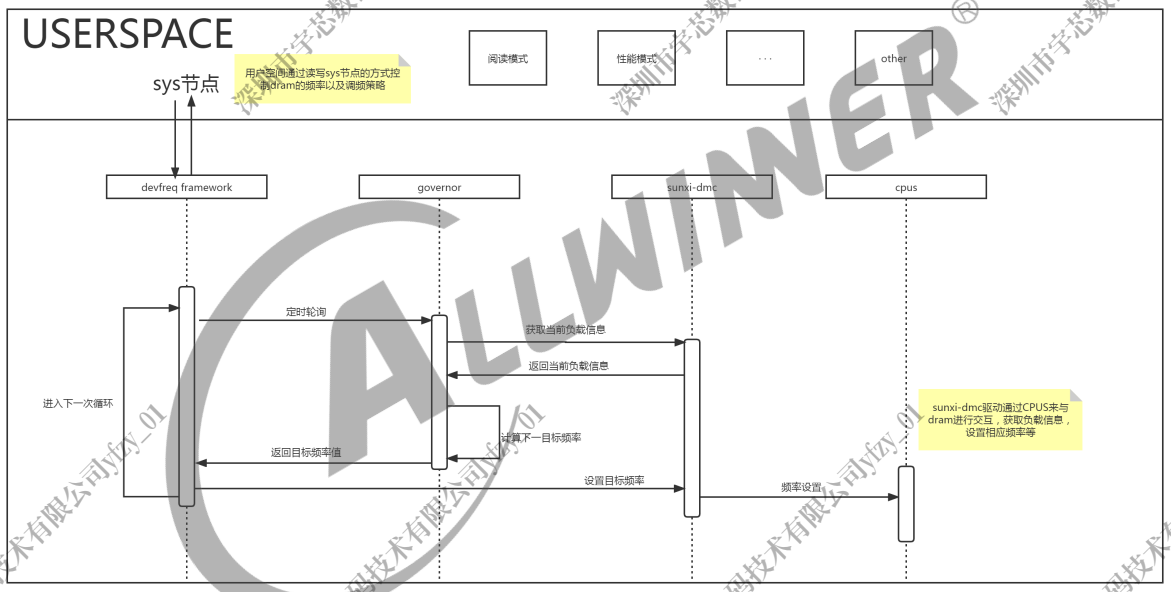

图 2-1: dram 调频框架

2.2 策略说明

当前 DVFS 调频框架有如下策略。

策略	基本效果	是否使用
Performance	性能模式-定频最高频率	使用并测试
Powersave	省电模式-在满足需求的情况下频率尽可能低	未使用未测试
Userspace	用户指定模式-根据用户需求设置频率	使用并测试
sunxi_actmon	全志定制化策略-尽可能覆盖全场景	使用并测试

策略说明：

- Performance：该模式会指定运行最高频率，不会自主调频；
- Userspace：用户指定策略，可自主设定运行设置的频率；
- sunxi_actmon：全志基于 SimpleOndemand 深度定制化开发，实时响应带宽峰值负载并根据需要进行调频

3 A733 调频策略

3.1 当前调频逻辑

当前调频的基本策略框架如下。

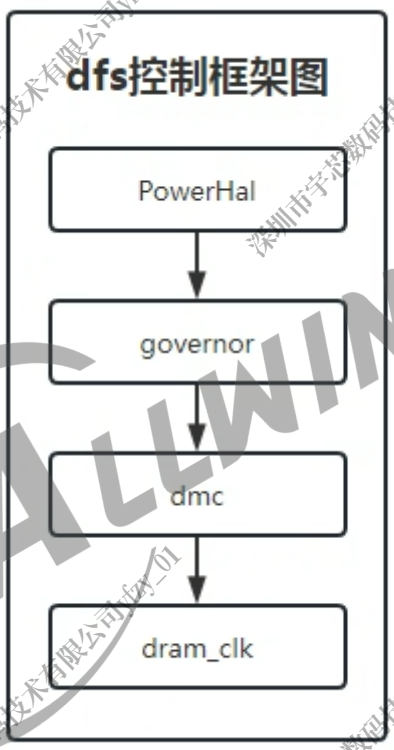

图 3-1: 调频策略

在 powerhal 中，通过场景管理，对 dfs 的 governor 进行控制，以满足性能场景的性能要求，如开机启动，应用启动等。

3.2 DVFS 使用

使用 DVFS，请按照如下步骤进行适配或者确认是否开启。

3.2.1 CPUs 支持 DVFS

主要文件包括：

```
device/config/chips/a733/bin/bl31.bin
brandy/dramlib/sun60iw2p1/arisc_liboem/libar100s.a
brandy/dramlib/sun60iw2p1/spl_libdram/libdram
```

以上文件请务必支持到最新版本。

3.2.2 libdram 支持 DVFS

DRAM 调频支持 dvfs，要确保获取的 dram 驱动版本支持 dfs，具体可以见串口 UBOOT 阶段打印。

```
[602]HELLO! SBOOT is starting!
...
[668]DRAM BOOT DRIVE INFO: V0.581_optim_mode1
...
```

说明

若获取的 libdram 无法支持 dfs，可联系 FAE 获取最新版支持。

3.2.3 dram para 参数支持 dvfs

若当前的 dram_para 参数支持 dvfs 调频，则在 boot0 打印可看到如下打印：

```
[602]HELLO! SBOOT is starting!
...
[843]DRAM BOOT DRIVE INFO: V0.581_optim_mode1
[849]DRAM_VCC set to 560 mv
[852]DRAM CLK =2400 MHZ
[854]DRAM Type =9 (8:LPDDR4,9:LPDDR5)
[991]Training result  is  = 7
[994]DRAM Pstate 1 training, frequency is 1200 Mhz
[1168]Training result  is  = 7
[1170]DRAM Pstate 2 training, frequency is 800 Mhz
[1509]Training result  is  = 7
[1512]DRAM Pstate 3 training, frequency is 400 Mhz
[1589]Training result  is  = 7
[1592]DRAM Pstate 0 training, frequency is 2400 Mhz
[1600]Actual DRAM SIZE =8192 M
[1603]DRAM SIZE =8192 MBytes, para1 = a10a, para2 = 20001001, dram_tpr13 = 65
[1613]DRAM simple test OK.
...
```

上述 log 则表示可以调频的 4 个频点，即 dvfs 调频只能调到这 4 个频点。

说明

使用 dfs 需要配置 dram para，支持 dfs 的 dram para 参数请联系 AW 获取支持，这里仅提供如何修改 dfs 频点的方法。

3.2.4 Linux 内核 DVFS 驱动加载

内核有独立的用于 dfs 调频的驱动。

3.2.4.1 驱动代码介绍

当前采用的内核使用 bsp 独立仓库，调频驱动的代码位于 BSP 仓库中。

```
devfreq/
├── event
│   ├── Kconfig
│   ├── Makefile
│   ├── sunxi-ddrpmu.c
│   └── sunxi-nsipmu.c
├── Kconfig
├── Makefile
├── sunxi-dmc.c
└── sun55iw3-devfreq.c
```

3.2.4.2 驱动支持

1. 驱动编译

支持 dvfs 功能，请检查 menuconfig 中正确开启了如下配置并配置为 m。

```
CONFIG_AW_DMC_DEVFREQ
CONFIG_AW_DDR_CCU
```

上述 config 配置对应如下 2 个驱动，开启后，请检查是否有编译出 ko 模块。

```
ccu-ddr.ko
sun55iw3-devfreq.ko
```

说明

驱动通常默认编译为 ko。

2. 驱动加载

当前 module 模块通常是在 android 启动过程的的 boot 阶段进行加载的，因此需要将上述驱动添加到即可。

补丁参考如下。

```
#enable devfs
on boot && property:persist.vendor.power.devfs.enabled=1
    insmod /vendor/lib/modules/ccu-ddr.ko
    insmod /vendor/lib/modules/sun55iw3-devfreq.ko
```

可看到，使用 dfs 还需要将 persist.vendor.power.devfs.enabled 属性配置为 1 才可加载驱动。

⚠ 注意

驱动有加载顺序要求，请按照上述顺序加载驱动即可。

3.2.5 Android 使能 DVFS 功能

驱动配置完成后，默认功能也是未使能的，只是有相关的功能而已，使用 DVFS 还需要打开 Android 层的配置。

当前 DVFS 功能仅支持灭屏场景，默认未使能，使能 DVFS 功能只需要在方案中配置相关属性即可。

```
diff  -- git  a/ad86310vc.mk b/ad86310vc.mk
index c0643c1..7b96ff0 100755
---  a/ad86310vc.mk
+++ b/ad86310vc.mk
@@ -86,6 +86,9 @@ PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
 PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
     persist. teclast. enable=1

+PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
+    persist. vendor.power.devfs.enabled = 1
+
 #add teclast prop
 PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
     ro. sk. homepage=http://www.teclast.com \
```

只需要使能 persist.vendor.power.devfs.enabled 即可。

说明

使用 DVFS 必现先确保 PowerHal 是使能的。

3.3 DVFS 调试

完成 DVFS 的使用后，可看到驱动基本加载，启动过程有相关驱动加载的打印。

```
[    14.169214][  T238] dram_clk:1200
[    14.172963][  T238] dram_div:0x1f090503
```

启动完成后，通过打开 debug 调试节点，也可看到相关调频的打印，其中，drate 表示为频率。

打开 debug 节点命令。

```
echo 1 > /sys/module/sun55iw3_devfreq/parameters/dbg_level
```

打开后，调频的打印如下

```
[110871.413488][  T528] rate:369600000, rw_data:0x3d2c
[110871.419180][  T528] drate:369M load:15 rw:466M total:2956M
```

但适配完成，可能由于使用场景，ddr 频率与公版不同，还会存在一些体验上的问题，如花屏，卡顿，响应慢等情况，这时可进行 dvfs 的调试。

3.3.1 修改 dvfs 频点

当前的 dfs 参数最多支持 4 个 ddr 频点调频，其中，和调频相关的参数主要是 dram_tpr2 参数。

以下列参数为例，描述如何设置频点：

```
dram_clk        = 924
...
dram_tpr2       = 0xf7090603
...
```

可以看到，当前 ddr 主频为 924M，dfs 参数为 0xf7090603，4 个频点的参数分别是 17、09、06、03，其中表示的频点分别为 154M，369M，528M，924M。

1. 频点解析

1. 0xf7，为 11110111，只需低 5 位，高 3 位为其他参数，故实际为 0x17。
2. 0x09，06，03：只取低 7 位，最高位为其他参数。

2. 频点计算

频点修改及频点计算公司如下。

```
频点=主频*4÷(tpr参数+1)
```

那么 924M = 924x4÷(3+1)，528M = 924x4÷(6+1)。

举例如果要修改 369M 频点为高一级频点，那么 924x4÷(8+1) = 410M，因此对于的 dram_tpr2 参数就是修改为：

```
dram_clk        = 792
...
dram_tpr2       = 0xf7080603
...
```

其余频点的设置参考即可。

⚠注意

1. **注意最低频率不要设置的太低，否则容易出现黑屏后音乐卡顿等问题；**
2. **修改 ddr 参数，最好在 FAE 指导下进行。**

3.3.2 dts 参数调试

dts 的配置主要有：

- 调频参数，需要根据实际调整。

3.3.2.1 调频参数的修改

调频参数主要涉及 2 个参数：upthreshold 和 downdifferential，当前 dts 的配置如下：

```
sunxi_dmcfreq: dmcfreq@3120000 {
        compatible = "allwinner,sun55iw3-dmc", "syscon";
        reg = <0x0 0x03120000 0x0 0x11000>,
              <0x0 0x02020000 0x0 0x4000>;
        interrupts = <GIC_SPI 121 IRQ_TYPE_LEVEL_HIGH>;
        clocks = <&ddr_clk>, <&ccu CLK_NSI>;
        clock-names = "dram", "bus";
        operating-points-v2 = <&dram_opp_table>;
        upthreshold = <35>;
        downdifferential = <20>;
        normalvoltage = <900000>;
        boostvoltage = <950000>;
};
```

其中，upthreshold 和 downdifferential 主要用于决策何时升频，何时降频，以当前 upthreshold = <35> 和 downdifferential = <20> 为例。

- 升频：当前带宽负载高于理论带宽的 35% 时提升到下一频点，即 upthreshold 配置的是升频参数。
- 降频：当前带宽负载低于理论带宽的 35-20=15% 且低于次一频点的 (35+20)/2=27.5% 时，降频到低一频点，即降频参数为 upthreshold - downdifferential 。

当前的参数设置的是比较保守的，主要原因如下。

- ddr 是实时设备，而当前调频策略存在滞后性，参数要保守，留有余量。
- 当前的参数未考虑到带宽有效系数，统计带宽时使用的是实际带宽，计算使用的是理论最大带宽。

3.3.3 Android 属性关闭 dvfs

启动后，若需要对 dfs 功能进行开关以进行 debug，可以通过设置属性 persist.vendor.power.devfs.enabled 为 0 或者 1 后重启，来关闭 dfs 功能。

参考操作。

```
su
setprop persist.vendor.power.devfs.enabled 0
reboot
```

说明

属性设置为 0 后，驱动将不会加载。

4 调试节点

4.1 常用节点及路径

加载了调频驱动后，会生成如下节点。

A733 平台

```
/sys/class/devfreq/a020000.dmcfreq
```

说明

sunxi-dmcfreq 名字可能会存在变更，但通常都会以 dmcfreq 结尾

其中有如下节点，及主要节点说明。

```
available_frequencies        //调频的频率，这里是dts里配置的频率表
available_governors          //可使用的策略
cur_freq                     //当前频率
device -> ../../../    sunxi-dmcfreq
governor                     //当前策略
max_freq                     //最高频率
min_freq                     //最低频率
name
polling_interval             //轮询监测时间间隔，单位为ms，默认为100ms，通常不修改
power
subsystem -> ../../../../../    class/devfreq
target_freq
trans_stat
uevent
waiting_for_supplier
```

4.2 userspace 设置频率

当策略设置为 userspace 时，用户可以手动设置频率，且会生成一个新的用于设置频率的节点。

A733 平台

```
/sys/class/devfreq/a020000.dmcfreq/userspace/set_freq
```

以 A733 平台为例，举例一次使用 userspace 设置频率的流程。

```
console:/sys/class/devfreq/a020000.dmcfreq # # echo userspace > governor  //切换策略为userspace
console:/sys/class/devfreq/a020000.dmcfreq # cat available_frequencies  //查看支持的频率
400000000 800000000 1200000000 2400000000
```

```
console:/sys/class/devfreq/a020000.dmcfreq # cat cur_freq                    //查看当前频率
400000000

console:/sys/class/devfreq/a020000.dmcfreq # echo 1200000000 > userspace/set_freq  //设置频率为1200M
ad86310vc:/sys/class/devfreq/sunxi-dmcfreq #
```

5 FAQ

5.1 如何查看当前调频是否有生效

1. 当调频生效时，首先要确保/sys/class/devfreq/sunxi-dmcfreq 下有相应的节点；
2. 通过 userspace 时手动设置频率可以正常设置，无异常打印；
3. 调频成功打印如下，后面的表示为频率。

```
[2022-07-19 12:14:48]  [73456.041906][T17912] prate:1584000000, drate:633600000
[2022-07-19 12:14:48]  [73456.168549][T17912] prate:1584000000, drate:792000000
[2022-07-19 12:14:48]  [73456.727253][T17912] prate:1584000000, drate:352000000
[2022-07-19 12:14:49]  [73457.759446][T17912] prate:1584000000, drate:633600000
[2022-07-19 12:14:50]  [73458.635403][T17912] prate:1584000000, drate:352000000
[2022-07-19 12:14:51]  [73459.296636][ T5097] prate:1584000000, drate:792000000
[2022-07-19 12:14:51]  [73459.459322][ T5097] prate:1584000000, drate:633600000
[2022-07-19 12:14:51]  [73459.611422][ T5097] prate:1584000000, drate:352000000
[2022-07-19 12:14:51]  [73459.732015][ T5097] prate:1584000000, drate:633600000
[2022-07-19 12:14:52]  [73459.975335][ T5097] prate:1584000000, drate:352000000
[2022-07-19 12:14:52]  [73463.907520][ T5097] prate:1584000000, drate:633600000
[2022-07-19 12:14:56]  [73464.035260][ T5097] prate:1584000000, drate:792000000
[2022-07-19 12:14:56]  [73464.783262][ T7905] prate:1584000000, drate:352000000
[2022-07-19 12:14:57]  [73464.907466][ T7905] prate:1584000000, drate:633600000
```

说明

部分平台需要先放开调频打印，以 733 为例，可参考 DVFS 调试章节打开节点后查看。

5.2 如何一键关闭 dfs 功能

如果已经配置了相关的驱动，dts，dram_para，但是发现想要关闭 dfs 功能，或临时关闭该功能用于调试，又不想去移除驱动等，那么只需要如下操作即可关闭 dfs 功能。

5.2.1 临时调试关闭

临时关闭只需要设置如下属性并重启即可。

```
persist. vendor.power.devfs.enabled
```

相应命令，获取 root 然后设置属性，确认设置完成后重启即可。

```
console:/ $ su
console:/ # setprop persist.vendor.power.devfs.enabled 0
console:/ # getprop persist.vendor.power.devfs.enabled
0
console:/ # reboot
```

5.2.2 固件默认关闭

默认关闭只需要在方案 mk 里去掉相应的属性设置，或者将属性设置为 0 均可。

```
diff  -- git  a/ad86310vc.mk b/ad86310vc.mk
index c0643c1..7b96ff0 100755
---  a/ad86310vc.mk
+++ b/ad86310vc.mk
@@ -86,6 +86,9 @@ PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
 PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
     persist. teclast. enable=1

-PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \
-     persist. vendor.power.devfs.enabled = 1
```

著作权声明

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明





（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的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所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。