Security Class: Top-Secret ( ) Secret ( ) Internal ( ) Public (√)

RK3399_Power_Consumption_Optimization

(Technical department, R & D Dept. II)

Status: [ ]Modifying [√]Released	Current Version:	V1.1
	Author:	Wang Jianhui
	Finish Date:	2019-07-25
	Auditor:	Chen Liang, Deng Xunjin, Zhang Wenping, Wei Jianxing
	Finish Date:	2019-07-26

Fuzhou Rockchip Electronics Co., Ltd

Revision History

Version no.	Author	Revision Date	Revision description	Remark
V1.0	Wang Jianhui	2018.11.21	Initial version release	
V1.1	Wang Jianhui	2019.07.25	Improve the power consumption analyzing method. Add the introduction of power consumption debug method. Add the introduction of power consumption optimization. Add sleep and resume issue analyzing. Add thermal control introduction. Add system self-analyzing checklist.	

Contents

Preface

Overview

This document mainly introduces the power consumption initial analyzing, optimizing and debugging method based on Rockchip RK3399 Android7.1 industry version, aiming to help software engineers optimize the power consumption of RK3399 platform Android7.1 industry version to balance the performance and power consumption.

Applicable object

This document is mainly suitable for below engineers:

Field application engineers

Software development engineers

1 Power consumption data of RK3399 Android7.1 industry SDK

1.1 RK3399 Android7.1 industry SDK--power consumption data based on excavator

Table 1-1 The power consumption data based on RK3399 excavator

Power consumption of RK3399 EXCAVATOR (ANDROID 7.1 REMOVE PANEL)																						
TEST SCENE		BAT		VCC3V3_SYS		VDD_CPU_B		VDD_CPU_L		VDD_GPU		VDD_LOG		VDD_0V9		VDD_CENTER		VCC_DDR		VCC_1V8		note
		Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	
Deep sleep power		8.00	5.80	3.32	12.30	0.00	0.00	0.00	0.00	0.00	0.00	0.90	15.90	0.92	1.30	0.00	0.00	1.27	4.80	1.86	1.30	①③
Desktop	Static	8.00	146.80	3.32	314.30	0.80	8.10	0.82	4.60	0.79	27.70	1.03	231.90	0.92	5.30	0.83	158.70	1.26	118.80	1.86	111.30	②④⑧
Video local	1080P	8.00	185.40	3.32	395.80	0.80	17.70	0.81	49.90	0.79	29.20	1.03	254.40	0.92	5.30	0.86	277.80	1.26	164.70	1.86	115.90	②⑤⑧
	4K	8.00	297.30	3.32	644.60	0.80	21.50	0.82	77.80	0.79	37.20	1.03	270.70	0.92	5.30	0.91	723.50	1.26	357.90	1.86	133.10	②⑥⑧
	1080P	8.00	187.80	3.32	401.30	0.80	22.50	0.82	50.90	0.79	29.80	1.03	254.40	0.92	5.30	0.86	283.50	1.26	170.90	1.86	114.90	②⑦⑧
Game local	Fish2	8.00	256.40	3.32	548.30	0.80	16.00	0.84	87.70	0.80	273.20	1.03	277.20	0.92	5.30	0.86	364.60	1.26	254.90	1.86	118.10	②⑧⑧
Video online	aiqiyi	8.00	259.60	3.32	556.40	0.80	25.80	0.82	79.20	0.80	265.30	1.03	256.90	0.92	5.40	0.86	301.20	1.26	223.40	1.86	117.10	②⑧⑧

Test Item Description: default remove screen

①: just sapphire socboard，remove L900、R901、R90016，powered at VCC_SYS，PCIE_VDD_0V9 can not be switched off when deep sleep.
②: remove excavator board U1401、R9015、FB2602、FB2604，disabled HDMI IN. powered at VCC_BATT.
③: Test time:60s. Waiting 10s,when the TP is standby,then Record data.
④: Test time:60s. Waiting 10s,when the TP is standby,then remove screen and record data.
⑤: Test time:5min. Source file:bbb_full.ffmpeg.1920x1080.mp4.libx264_10000kbps_30fps.libfaac_stereo_192kbps_48000Hz.mp4.
⑥: Test time:5min. Source file:4K_60fps_H265.mp4,Full Scrreen.
⑦: Test time:5min. Source file:1920x1080,27.0 Mbps,23.976fps,AVC,DTS AC3 AC3,1536 Kbps,变形金刚2：卷土重来sample.mkv.
⑧: Test time:5min. Fish2_v1_1_3.apk.
⑨: Test time:5min. aiqiyi_8.8.0.apk play 车手，1080p.
⑩: Used EDP panel，VCC_1V8 will increase 78.8mA .

Table 1-1 shows the power consumption data of RK3399 excavator in the room temperature environment (25℃). DDR uses lpddr3 component, and customers' current projects using lpddr3 can refer to the power consumption data in Table 1-1 to do the comparison and analysis. Table 1-1 includes the power consumption data of the scenarios such as Deep Sleep, static desk, 1080p video playing, 4K video playing, Fish2 stress test and Aiqiyi on-line video playing. The test source and apk in table 1-1 can be acquired by contacting with RK FAE e-mail fae@rock-chips.com.

1.2 RK3399 Android7.1 industry SDK—power consumption data based on EVB board

Table 1-2 The power consumption data based on RK3399 EVB board

Power consumption of RK3399 solutions																						
Power consumption of RK3399 LPDDR4 (400-800M负载变频)EVB																						
TEST SCENE		BAT		VCC3V3_SYS		VDD_CPU_B		VDD_CPU_L		VDD_GPU		VDD_LOG		VDD_0V9		VDD_CENTER		VCC_DDR		VCC_1V8		note
		Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Current	
Deep sleep power		12.0	12.0	3.3	17.7	0.0	0.0	0.0	0.0	0.0	0.0	0.9	6.0	0.9	4.2	0.0	0.0	1.1	3.3	1.8	5.8	③
Desktop	Static	12.0	114.6	3.3	317.2	0.8	5.2	0.8	4.2	0.8	23.2	0.9	201.4	0.9	38.4	0.9	297.5	1.1	144.0	1.8	75.1	④
Video local	1080P	12.0	141.6	3.3	383.9	0.8	14.8	0.8	47.7	0.8	21.9	0.9	206.9	0.9	38.1	0.9	349.7	1.1	193.9	1.8	84.7	⑤
	4K	12.0	232.4	3.3	688.7	0.8	12.8	0.8	80.2	0.8	29.5	0.9	218.1	0.9	39.0	0.9	829.8	1.1	400.6	1.8	117.8	⑥
	1080P	12.0	141.2	3.3	385.9	0.8	19.7	0.8	48.2	0.8	25.7	0.9	208.5	0.9	38.5	0.9	357.1	1.1	195.6	1.8	82.3	⑦
Game local	Fish2	12.0	161.5	3.3	422.8	0.8	15.7	0.8	73.8	0.8	225.8	0.9	213.0	0.9	38.3	0.9	350.0	1.1	232.0	1.8	85.0	⑧

③: Test time:60s. Waiting 10s,when the TP is standby,then Record data.
④: Test time:60s. Waiting 10s,when the TP is standby,then remove screen and record data.
⑤: Test time:5min. Source file:bbb_full.ffmpeg.1920x1080.mp4.libx264_10000kbps_30fps.libfaac_stereo_192kbps_48000Hz.mp4.
⑥: Test time:5min. Source file:4K_60fps_H265.mp4,Full Scrreen.
⑦: Test time:5min. Source file:1920x1080,27.0 Mbps,28.976fps,AVC,DTS AC3 AC3,1536 Kbps,变形金刚2：卷土重来sample.mkv.
⑧: Test time:5min. Fish2_v1_1_3.apk.

Table 1-2 shows the power consumption data RK3399 EVB board in the room temperature environment (25℃). DDR uses lpddr4 component, and customers' current projects using lpddr4 can refer to the power consumption data in Table 1-2 to do the comparison and analysis. Table 1-2 includes the power consumption data of the scenarios such as Deep Sleep, static desk, 1080p video playing, 4K video

playing and Fish2 stress test. The test source and apk can be acquired by contacting with RK FAE e-mail fae@rock-chips.com.

2 Power consumption initial analyzing step

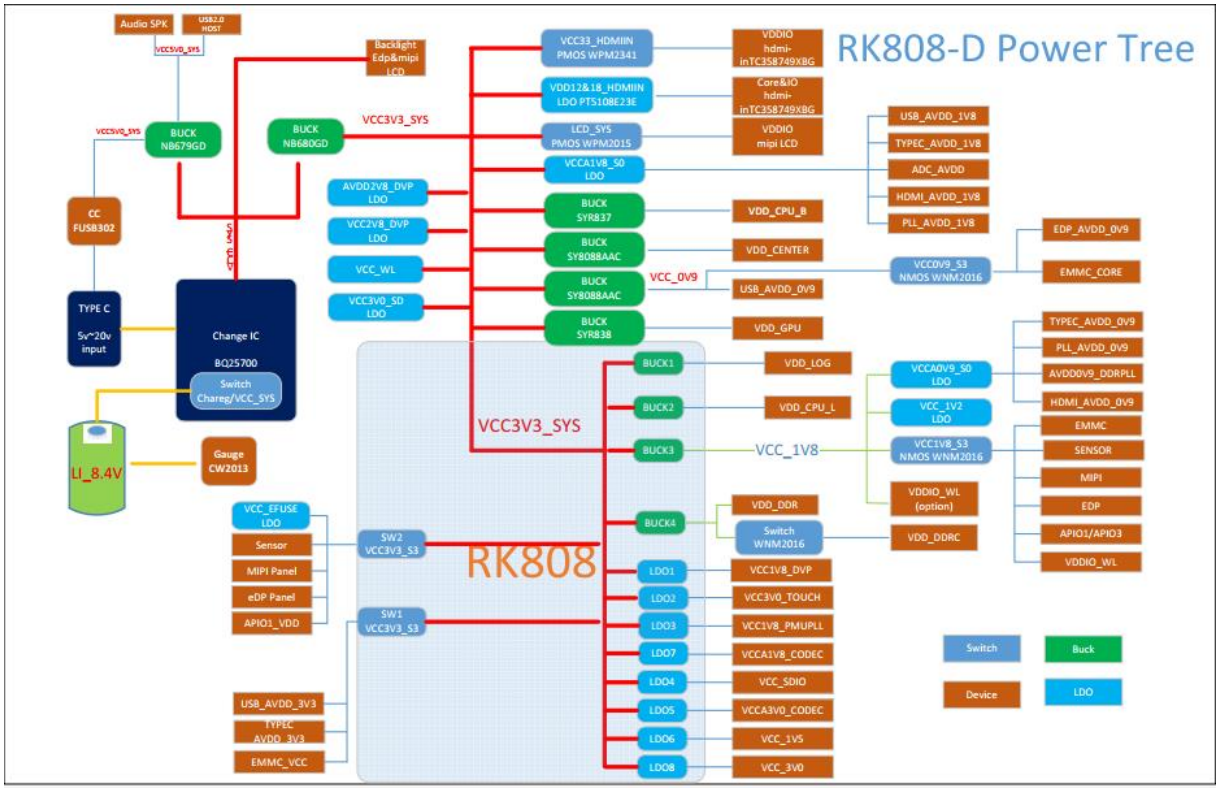
2.1 Compare total power consumption

Firstly, compare total power consumption in the same scenario, to initially check if the power consumption of customer board is normal or not. BAT column in Table 1-1 and Table 1-2 shows the total power consumption of the scenarios based on RK3399 Android7.1 industry SDK board. If there is a big difference between total power consumption of customers' board and the reference data of SDK board provided by RK, it is recommended to further break down the power consumption. Please refer to chapter 2.2.

Note: when testing the power consumption data, ensure that all the test conditions are the same except hardware difference. For example, for comparing the power consumption data of 1080p video playing scenario, use the same Android version, the same player APK, and the same video source, removing edp panel and hdmi display.

2.2 Break down the power consumption of each path

In order to break down the power consumption, it is necessary to be familiar with Power Tree of the project. Picture 1-1 is the Power Tree of RK3399 excavator. During the project power consumption optimization, you can list Power Tree according to the hardware schematic, and then series connect the resistor with certain resistance (generally recommended to series connect 20mR resistance) in the corresponding circuit, to measure the current of the corresponding circuit. According to the power consumption table in chapter 1.1, RK3399 excavator breaks down the power consumptions as follows: VCC3V3_SYS, VDD_CPU_B, VDD_CPU_L, VDD_GPU, VDD_LOG, VDD_0V9, VDD_CENTER, VCC_DDR and VCC_1V8.



Picture 1-1 RK3399 excavator Power Tree

After breaking down the power consumption, compare the data. Firstly compare if CPU big/little core, GPU, DDR power consumption is normal or not, to confirm if CPU frequency scaling, GPU frequency scaling, and DDR frequency scaling is normal or not. If the voltage is increased, confirm if the voltage increased is 12.5mV times or not. If the frequency scaling strategy is normal, need to further compare and analyze other peripheral power consumptions.

2.3 Relative introduction of the power break down (take excavator Power Tree as example)

VCC3V3_SYS supplies power for VDD_CPU_B, VDD_CPU_L, VDD_GPU, VDD_LOG, VDD_0V9, VDD_CENTER, VCC_DDR, VCC_1V8（VCC3V3_S3）, etc. As can be seen from picture 1-1, Audio SPK, USB2.0 HOST, FUSB302, Backlight and EDP/MIPI LCD have separate power supply, so we can roughly estimate the power consumption for these paths: BTA power consumption - VCC3V3_SYS power consumption.

VDD_CPU_B supplies power for CPU big core. If this power consumption has big difference, need to check if CPU big core is always running with high frequency, and if each level voltage of CPU big core is

reasonable.

VDD_CPU_L supplies power for CPU little core. If this power consumption has big difference, need to check if CPU little core is always running with high frequency, and if each level voltage of CPU little core is reasonable.

VDD_GPU supplies power for GPU. If this power consumption has big difference, need to check if GPU is fixed with high frequency, and if each level voltage of GPU is reasonable.

VDD_LOG supplies power for MPI CSI, ISP, CLK, etc. If the power consumption data is relatively large, need to check if clk is correctly configured, and power domain is correctly configured. VDD_LOG voltage is recommended between 0.94V-1V.

VDD_0V9 supplies power for usb2.0 USB_AVDD_0V9, EDP_AVDD_0V9, EMMC_CORE, etc.

VDD_CENTER supplies power for phy and ddr control. If this power consumption has big difference, need to check if there is problem with DDR frequency, and confirm if phy and ddr controller are normal.

VDD_DDR supplies power for ddr component. If this power consumption has big difference, check if ddr frequency scaling is enabled or not, and each level voltage is set correctly or not. Probably ddr is always running with high frequency. For lpddr4 component, currently there are only two levels 416M and 856M available, so the power consumption is relatively large.

VCC_1V8 supplies power for TYPEC_AVDD_0V9, PLL_AVDD_0V9, AVDD0V9_DDRPLL, HDMI_AVDD_0V9, EMMC, SENSOR, APIO1/APIO3, etc. When this power consumption is relatively large, need to check if the peripheral power consumption is normal or not, and gpio voltage is configured correctly.

3 Running power consumption debug introduction

3.1 Real-time inquire system loading

RK3399 Android7.1 industry sdk integrates hardware_monitor.sh script after V2.1.0, which can inquire the following information: CPU current frequency, loading, temperature; GPU current frequency, loading, temperature; DDR current frequency, loading; UI refresh rate; information running time and so on. When using hardware_monitor.sh script, need to do su. Picture 1-2 shows the output information when

running hardware_monitor.sh.

Note:

1. Some customers use lpddr4, and do not enable ddr frequency scaling function. It can be used only after deleting ddr information acquisition in hardware_monitor.sh.

2. It is not able to use this script for user version image.

```
su
rk3399_a11:/ # hardware_monitor.sh
hardware_monitor.sh

Hardware Monitor for RK3399 , Version: 1.7
        F - Freq(MHz)
        L - Load(%)
        T - Temperature(C)

[Model]: rk3399-a11
[Firmware]: rk3399_a11-userdebug 7.1.2 NHG47K eng.zwp.20190723.174420 test-keys
[Kernel]: Linux version 4.4.126 (zwp@rk-intel-1) (gcc version 6.3.1 20170404 (Linaro GCC 6.3-2017.05) ) #394 SMP PREEMPT Tue Jul 23 17:41:04 CST 2019


UPTIME(s)       CPU(LF/BF/L/T)  GPU(F/L/T)      VPU/HEVC(F)     DDR(F/L)        FPS
5:05,   1008/408/01/32  200/00/32      000/000         528/19          0.0
5:05,   1200/816/18/33  200/46/32      000/000         800/19          58.0
5:05,   1200/816/27/33  200/10/32      000/000         528/24          59.9
5:05,   1416/408/27/32  200/43/32      000/000         800/39          58.2
5:05,   1416/408/32/33  200/32/32      000/000         800/25          58.3
5:05,   1416/408/26/33  200/55/32      000/000         528/30          57.9
5:05,   1200/408/25/33  200/46/33      000/000         800/27          60.0
5:06,   1416/408/21/33  200/38/32      000/000         600/31          58.0
5:06,   1200/408/18/33  200/00/32      000/000         528/22          57.9
```

Picture 1-2 hardware_monitor.sh information

Parameter description:

UPTIME: system running time

CPU L：little core CPU frequency, CPU B：big core CPU frequency, CPU L：CPU loading, CPU T：CPU temperature GPU F：GPU frequency, GPU L：GPU loading, GPU T：GPU temperature

VPU/HEVC（F）: currently not used

DDR F：DDR frequency, DDR L：DDR loading

FPS: UI refresh rate

3.2 Real-time inquire CPU loading

RK3399 Android7.1 industry sdk replaces the system native top tool after V2.1.7. The command: top -t -m 10 can check CPU loading status, the thread is running on which CPU, and how much the loading is. Picture 1-3 shows the system thread information printed by top.

Parameter: -t means to print the thread loading information, -m 10 means to print top 10 threads according to CPU loading.

```
  PID   TID PR CPU% S     VSS     RSS PCY UID      Thread          Proc
 1607 14639  0  10% S 1833472K 163616K  fg u0_a45   CameraThread    com.alipay.zoloz.smile
 1607 14599  5   8% R 1833472K 163616K  fg u0_a45   pool-10-thread- com.alipay.zoloz.smile
 1607 14641  2   2% S 1833472K 163616K  fg u0_a45   Thread-134      com.alipay.zoloz.smile
 1607 14633  1   1% S 1833472K 163616K  fg u0_a45   CameraThread    com.alipay.zoloz.smile
 1607  1623  3   0% R 1835368K 165540K  fg u0_a45   HeapTaskDaemon  com.alipay.zoloz.smile
 1557  1653  3   0% S 1796780K 169904K  fg u0_a43   RenderThread    zoloz.phone.android.alipay.com.dragonfly
 1607  1668  1   0% S 1835368K 164592K  fg u0_a45   Thread-2        com.alipay.zoloz.smile
 1557  1557  3   0% S 1796780K 169904K  fg u0_a43   y.com.dragonfly zoloz.phone.android.alipay.com.dragonfly
  229   229  3   0% S 301132K  56780K  fg system   surfaceflinger  /system/bin/surfaceflinger
15008 15008  4   0% R    796K    400K  fg root     top             top
 2329  2329  2   0% S   7716K   2596K  fg root     hardware_monito /system/bin/sh
 1692  1736  0   0% S 1712524K  98944K  bg u0_a45   Thread-2        com.alipay.zoloz.smile:push
  229   257  3   0% S 301132K  56780K  fg system   Binder:229_1    /system/bin/surfaceflinger
  464 14577  1   0% S 2218096K 187188K  fg system   Binder:464_E    system_server
 1557  1666  3   0% S 1796780K 169904K  fg u0_a43   mali-cmar-backe zoloz.phone.android.alipay.com.dragonfly
```

Picture 1-3 top system information

Print information description:

PID：process ID，TID：thread ID

PR: which CPU current thread is running on

CPU%: CPU loading of current thread, when CPU loading of some thread is close to 17% (6 CPU cores, 100/6=17%), it means current thread is already full running on this CPU core.

Thread: thread name, Proc: process name

Note:

Check the system loading of some application through top tool. If some thread of apk occupies large CPU loading, and apk has few threads, and CPU loading of other threads is low, the task of this thread can be divided into several threads for processing, which can reduce CPU loading and achieve the purpose of reducing the power consumption.

3.3 CLK check

Check each sub system clk status, can see clock frequency, enable_count, prepare_count, parent-child clock relationship and so on. The commonly used clk commands are as below (take saradc as example):

The command to inquire CLK Tree:

cat /d/clk/clk_summary

Set the frequency (unit Hz):

echo 24000000 > /sys/kernel/debug/clk_saradc/clk_rate

Acquire the frequency:

cat /sys/kernel/debug/clk_saradc/clk_rate

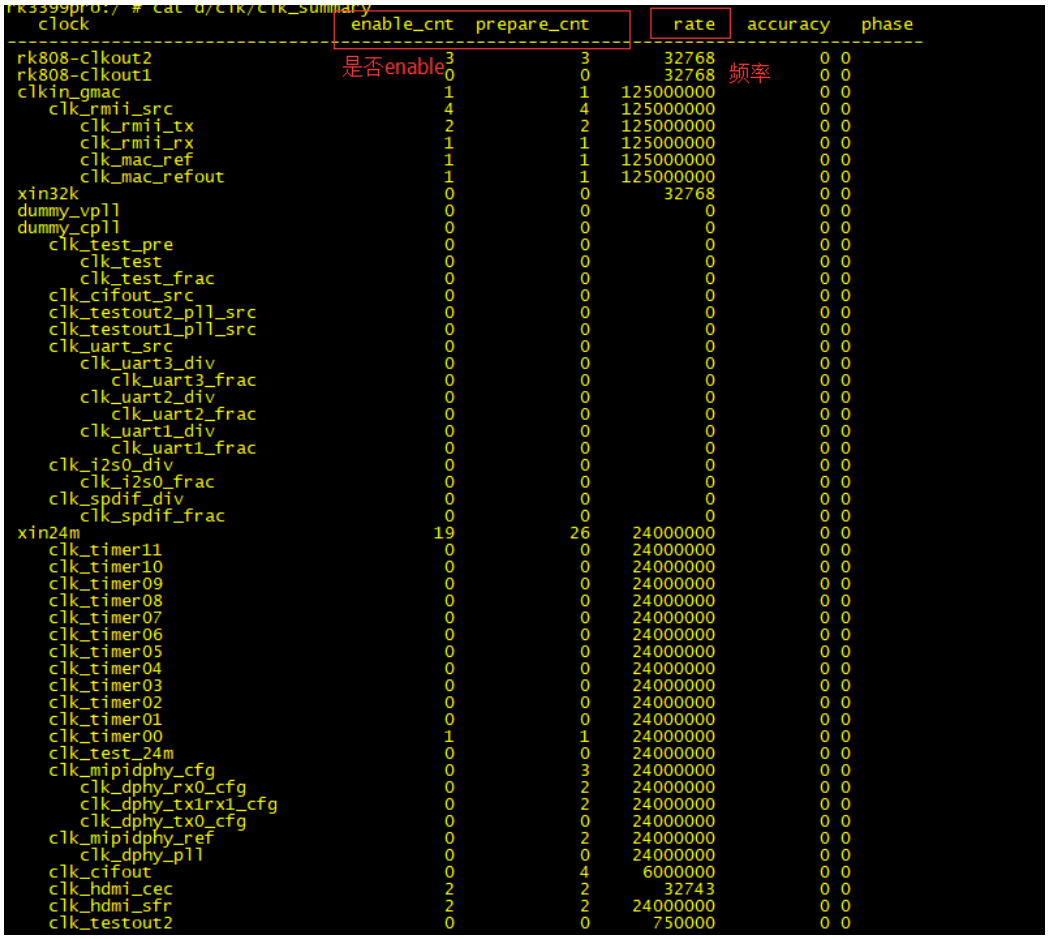
Enable clk:

echo 1 > /sys/kernel/debug/clk_saradc/clk_enable_count

Disable clk:

echo 0 > /sys/kernel/debug/clk_saradc/clk_enable_count

Note: enable_count will increase 1 after the driver applying to enable the clock (clk_enable). But for some clocks constantly enabled by default, enable_count may be 1 or 0, and you don't need to care about them for checking power consumption. Mainly check if the CLK enable_cnt of unused device is 0 or not. If it is not 0, you can manually use CLK disabling command to disable for debugging.



Picture 1-4 CLK Tree

Configure clk in dts. Take vopb configuration as example:

vopb: vop@ff900000 {

clocks = <&cru ACLK_VOP0>, <&cru DCLK_VOP0>, <&cru HCLK_VOP0>, <&cru DCLK_VOP0_DIV>;

clock-names = "aclk_vop", "dclk_vop", "hclk_vop", "dclk_source";

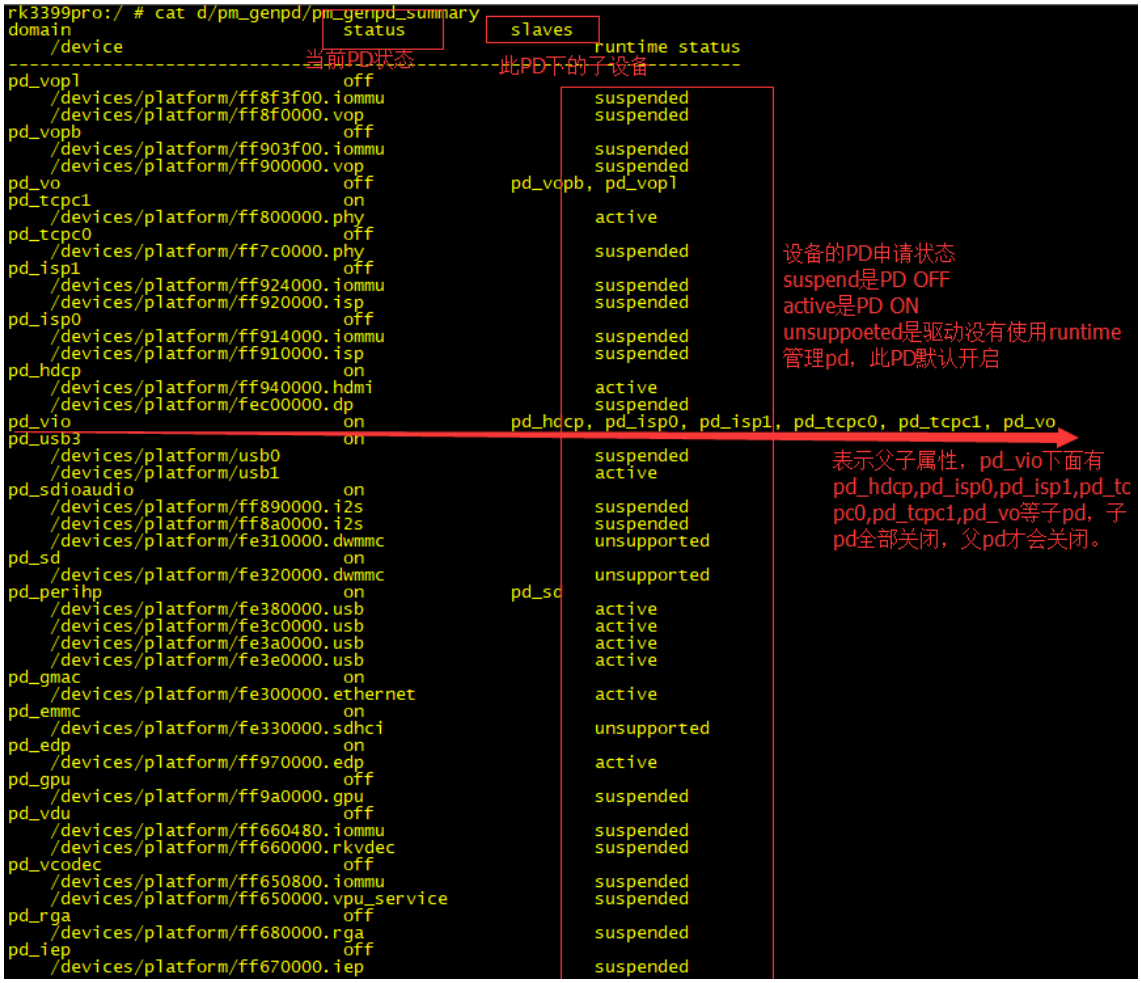
};

3.4 DP check

Confirm current status of each PD. The status of unused module should suspend, and then this PD will be closed after all the devices under this pd are suspended. For more details please refer to picture 1-5.

The command to inquire PD summary:

cat /d/pm_genpd/pm_genpd_summary



Picture 1-5 PD summary information

dts configuration (take vopb as example):

vopb: vop@ff900000 {

power-domains = <&power RK3399_PD_VOPB>;

};

Note: if PD is not invoked in dts node, it is considered that this PD is not used by devices and it will

be closed by framework after power on. If pd invocation is added in above dts node, but vop driver doesn't have runtime operation, this pd is constantly enabled (because the status is unsupported as this device doesn't support runtime). If pd invocation is added in dts node, and there is runtime operation in the driver, the status of this pd depends on the driver application.

Summary of reducing running power consumption: inquire pd summary and clk tree, for those modules not used, need to close PD and CLK to avoid the current leakage of internal mos tube.

For example: when standby, vop0 module is not used. Then the corresponding clk and pd of this module should be closed. Clk (enable_count of both dclk_vop0 and aclk_vop0 are 0) is as below:

```
rk3399pro:/ # cat d/clk/clk_summary | grep vop
       dclk_vop1_div                    0        2     6000000          0 0
          dclk_vop1                     0        1     6000000          0 0
          dclk_vop1_frac                0        0      300000          0 0
          aclk_vop1_pre                 2        3   400000000          0 0
             aclk_vop1_noc              1        1   400000000          0 0
             aclk_vop1                  0        4   400000000          0 0
             hclk_vop1_pre              1        2   100000000          0 0
                hclk_vop1_noc           1        1   100000000          0 0
                hclk_vop1               0        4   100000000          0 0
          clk_vop1_pwm                  0        0   100000000          0 0
          clk_vop0_pwm                  0        0   100000000          0 0
          aclk_vop0_pre                 2        3   400000000          0 0
             aclk_vop0_noc              1        1   400000000          0 0
             aclk_vop0                  0        4   400000000          0 0
             hclk_vop0_pre              1        2   100000000          0 0
                hclk_vop0_noc           1        1   100000000          0 0
                hclk_vop0               0        4   100000000          0 0
          dclk_vop0_div                 0        2   200000000          0 0
             dclk_vop0                  0        1   200000000          0 0
             dclk_vop0_frac             0        0    10000000          0 0
rk3399pro:/ #
```

Picture 1-6 vop0 clk information

pd (the status of pd_vop0 is off)

```
172.16.12.236 (1) | 10.10.10.110 | 172.16.12.236 (1) (1) | Serial-COM8 | 172.16.12.236 (1) (2) | 172.16.12.236 (1) (3) | 10.10
domain                          status          slaves
    /device                                             runtime status
------------------------------------------------------------------------
pd_vopl                         off
    /devices/platform/ff8f3f00.iommu                    suspended
    /devices/platform/ff8f0000.vop                      suspended
pd_vopb                         off
    /devices/platform/ff903f00.iommu                    suspended
    /devices/platform/ff900000.vop                      suspended
pd_vo                           off             pd_vopb, pd_vopl
pd_tcpc1                        on
    /devices/platform/ff800000.phy                      active
pd_tcpc0                        off
    /devices/platform/ff7c0000.phy                      suspended
```

Picture 1-7 vop0 PD information

4 Standby power consumption debug introduction

4.1 Suspend and resume issue analyzing

Cannot enter suspend mode:

1. Use the following command to check if there is wake_lock:

cat /sys/power/wake_lock (only can see Android lock, but cannot see the lock set in kernel)

cat /sys/kernel/debug/wakeup_sources (can see all the locks)

```
1|rk3399_all:/ #
1|rk3399_all:/ # echo xxxx > /sys/power/wake_lock
rk3399_all:/ # cat /sys/kernel/debug/wakeup_sources
name                                    active_count    event_count     wakeup_count    expire_count    active_since
revent_suspend_time
xxxx                                    1               1               0               0               12877
usb@fe800000                            0               0               0               0               0
wlan_wd_wake                            3130458         3130458         0               0               0
wlan_wake                               2625896         2625896         0               0               0
5               0
PowerManagerService.WakeLocks           12              12              0               0               312437400
0646            0
PowerManagerService.Broadcasts          3               3               0               0               0
event1                                  0               0               0               0               0
event0                                  104             104             0               0               0
event2                                  4               4               0               0               0
eventpoll                               108             108             0               0               0
KeyEvents                               10480           10480           0               0               0
[timerfd]                               0               0               0               0               0
[timerfd]                               0               0               0               0               0
[timerfd]                               12457           12457           0               0               0
[timerfd]                               95              95              0               0               0
```

图 1-8 wakeup_sources 信息

Picture 1-8 wakeup_sources information

2. If there is interrupt occurring during suspend, check which module causes the interrupt according to log.

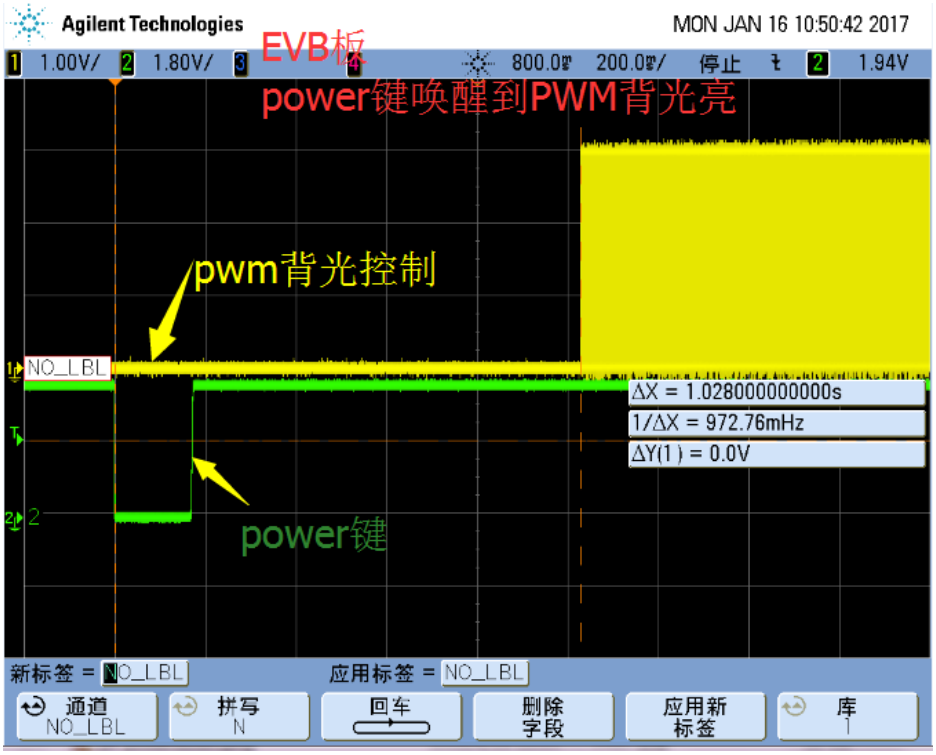
3. Check if the alarm interrupt occurring in 2s is set or not.

```
static int alarmtimer_suspend(struct device *dev)
{
    struct rtc_time tm;
    ktime_t min, now;
    unsigned long flags;
    struct rtc_device *rtc;
    int i;
    int ret;

    spin_lock_irqsave(&freezer_delta_lock, flags);
    min = freezer_delta;
    freezer_delta = ktime_set(0, 0);
    spin_unlock_irqrestore(&freezer_delta_lock, flags);

    rtc = alarmtimer_get_rtcdev();
    /* If we have no rtcdev, just return */
    if (!rtc)
        return 0;

    /* Find the soonest timer to expire*/
    for (i = 0; i < ALARM_NUMTYPE; i++) {
        struct alarm_base *base = &alarm_bases[i];
        struct timerqueue_node *next;
        ktime_t delta;

        spin_lock_irqsave(&base->lock, flags);
        next = timerqueue_getnext(&base->timerqueue);
        spin_unlock_irqrestore(&base->lock, flags);
        if (!next)
            continue;
        delta = ktime_sub(next->expires, base->gettime());
        if (!min.tv64 || (delta.tv64 < min.tv64))
            min = delta;
    }
    if (min.tv64 == 0)
        return 0;

    if (ktime_to_ns(min) < 2 * NSEC_PER_SEC) {
        __pm_wakeup_event(ws, 2 * MSEC_PER_SEC);
        return -EBUSY;
    }

    /* Setup an rtc timer to fire that far in the future */
    rtc_timer_cancel(rtc, &rtctimer);
    rtc_read_time(rtc, &tm);
    now = rtc_tm_to_ktime(tm);
```

2S内就唤醒就不让睡。

Wake up time is too long:

Take RK3399 as example, it takes 972ms for backlight pwm to output the wave after pressing the power key, as shown in picture 1-9.



Picture 1-9 The wave of wake up process

If the wake up time is too long, you can locate through the following method:

1. The command format:

echo N > /sys/module/printk/parameters/console_suspend

echo 1 > /sys/power/pm_print_times

Print out the suspend time of each device.

```
[   70.672395] calling  rfkill2+ @ 277, parent: mmc2:0001:2, cb: rfkill_suspend
[   70.673053] call rfkill2+ returned 0 after 1 usecs
[   70.673511] calling  rfkill1+ @ 277, parent: phy0, cb: rfkill_suspend
[   70.674100] call rfkill1+ returned 0 after 0 usecs
[   70.674576] calling  phy0+ @ 180, parent: mmc2:0001:2, cb: wiphy_suspend
[   70.674604] call phy0+ returned 0 after 6 usecs
[   70.675607] calling  mmc2:0001:3+ @ 277, parent: mmc2:0001, cb: pm_generic_suspend
[   70.675628] call mmc2:0001:3+ returned 0 after 0 usecs
[   70.675646] calling  mmc2:0001:2+ @ 277, parent: mmc2:0001, cb: pm_generic_suspend
[   70.675658] bcmsdh_sdmmc_suspend Enter func->num=2
[   70.675669] bcmsdh_sdmmc_suspend Exit
[   70.675679] call mmc2:0001:2+ returned 0 after 20 usecs
[   70.675695] calling  mmc2:0001:1+ @ 277, parent: mmc2:0001, cb: pm_generic_suspend
[   70.675706] bcmsdh_sdmmc_suspend Enter func->num=1
[   70.675716] call mmc2:0001:1+ returned 0 after 8 usecs
[   70.675736] calling  mmc2:0001+ @ 277, parent: mmc2, cb: mmc_bus_suspend
[   70.675752] call mmc2:0001+ returned 0 after 4 usecs
[   70.675775] calling  input1+ @ 277, parent: rockchip-key, cb: input_dev_suspend
[   70.675790] call input1+ returned 0 after 2 usecs
[   70.675821] calling  es8316-sound+ @ 277, parent: platform, cb: platform_pm_suspend
[   70.680107] call es8316-sound+ returned 0 after 4171 usecs
[   70.680130] calling  dmc+ @ 277, parent: platform, cb: platform_pm_suspend
[   70.680154] call dmc+ returned 0 after 11 usecs
[   70.680173] calling  fe310000.dwmmc+ @ 277, parent: platform, cb: pm_generic_suspend
[   70.680187] call fe310000.dwmmc+ returned 0 after 1 usecs
[   70.680297] calling  usb4+ @ 180, parent: fe3e0000.usb, cb: usb_dev_suspend
[   70.736885] call usb4+ returned 0 after 55234 usecs
[   70.736970] calling  fe3e0000.usb+ @ 277, parent: platform, cb: pm_generic_suspend
[   70.736999] call fe3e0000.usb+ returned 0 after 14 usecs
[   70.737102] calling  usb3+ @ 180, parent: fe3a0000.usb, cb: usb_dev_suspend
[   70.793279] call usb3+ returned 0 after 54834 usecs
[   70.793409] calling  fe3a0000.usb+ @ 277, parent: platform, cb: pm_generic_suspend
[   70.793438] call fe3a0000.usb+ returned 0 after 14 usecs
[   70.793606] calling  usb2+ @ 180, parent: fe3c0000.usb, cb: usb_dev_suspend
[   70.806331] call usb2+ returned 0 after 12308 usecs
```

2. Open DPM_WATCHDOG_TIMEROUT

```
Symbol: DPM_WATCHDOG_TIMEOUT [=5]
Type  : integer
Range : [1 120]
Prompt: Watchdog timeout in seconds
  Location:
    -> Power management options
      -> Device power management core functionality (PM [=y])
        -> Power Management Debug Support (PM_DEBUG [=y])
(2)       -> Device suspend/resume watchdog (DPM_WATCHDOG [=y])
  Defined at kernel/power/Kconfig:211
  Depends on: DPM_WATCHDOG [=y]
```

The time can be configured, but only with second accuracy.

```
        [ ]     Extra PM attributes in sysfs for low-level debugging/testing
        [ ]     Test suspend/resume and wakealarm during bootup
        [*]     Device suspend/resume watchdog
        (60)      watchdog timeout in seconds (NEW)
        [ ] Enable workqueue power-efficient mode by default
```

```
--- a/drivers/mfd/rk808.c
+++ b/drivers/mfd/rk808.c
@@ -842,12 +842,12 @@ err_irq:
        regmap_del_irq_chip(client->irq, rk808->irq_data);
        return ret;
 }
-
+#include <linux/delay.h>
 static int rk808_suspend(struct device *dev)
 {
        int i, ret;
        struct rk808 *rk808 = i2c_get_clientdata(rk808_i2c_client);
-
+       mdelay(8000);
        for (i = 0; i < suspend_reg_num; i++) {
                ret = regmap_update_bits(rk808->regmap,
                                         suspend_reg[i].addr,
(END)
```

```
[   31.897864] rk808 0-001c: **** DPM device timeout ****
[   31.898102] [<c010f608>] (unwind_backtrace) from [<c010b800>] (show_stack+0x10/0x14)
[   31.898248] [<c010b800>] (show_stack) from [<c0504c9c>] (dpm_watchdog_handler+0x20/0x4c)
[   31.898388] [<c0504c9c>] (dpm_watchdog_handler) from [<c018cb90>] (call_timer_fn+0xa0/0x20c)
[   31.898514] [<c018cb90>] (call_timer_fn) from [<c018cf3c>] (run_timer_softirq+0x240/0x2cc)
[   31.898636] [<c018cf3c>] (run_timer_softirq) from [<c012a45c>] (__do_softirq+0x138/0x354)
[   31.898750] [<c012a45c>] (__do_softirq) from [<c012a900>] (irq_exit+0x88/0xf8)
[   31.898870] [<c012a900>] (irq_exit) from [<c017b884>] (__handle_domain_irq+0x8c/0xb0)
[   31.898986] [<c017b884>] (__handle_domain_irq) from [<c010142c>] (gic_handle_irq+0x44/0x74)
[   31.899086] [<c010142c>] (gic_handle_irq) from [<c010c314>] (__irq_svc+0x54/0x90)
[   31.899153] Exception stack(0xdc90bbf0 to 0xdc90bc38)
[   31.899235] bbe0:                                     dc90bc50 00000000 fd640800 c1298c80
[   31.899341] bc00: dc90bc50 3835d96f 00005dbf c05239d4 ddb158d0 ddb34e20 c0ee31e3 c1258f7c
[   31.899434] bc20: c05239d4 dc90bc40 c03c86c0 c010edf4 a00f0113 ffffffff
[   31.899543] [<c010c314>] (__irq_svc) from [<c010edf4>] (arch_timer_read_counter_long+0x8/0x18)
[   31.899674] [<c010edf4>] (arch_timer_read_counter_long) from [<c03c86c0>] (read_current_timer+0x20/0x38)
[   31.899794] [<c03c86c0>] (read_current_timer) from [<c03c8708>] (__timer_delay+0x30/0x6c)
[   31.899918] [<c03c8708>] (__timer_delay) from [<c0523a04>] (rk808_suspend+0x30/0xac)
[   31.900043] [<c0523a04>] (rk808_suspend) from [<c0504d80>] (dpm_run_callback+0xb8/0x23c)
[   31.900154] [<c0504d80>] (dpm_run_callback) from [<c0505b5c>] (__device_suspend+0x254/0x340)
[   31.900263] [<c0505b5c>] (__device_suspend) from [<c0507694>] (dpm_suspend+0x12c/0x38c)
[   31.900373] [<c0507694>] (dpm_suspend) from [<c0176df0>] (suspend_devices_and_enter+0x78/0x350)
[   31.900484] [<c0176df0>] (suspend_devices_and_enter) from [<c0177734>] (pm_suspend+0x66c/0x78c)
[   31.900585] [<c0177734>] (pm_suspend) from [<c017591c>] (state_store+0x40/0x68)
[   31.900691] [<c017591c>] (state_store) from [<c0297b18>] (kernfs_fop_write+0x148/0x1ac)
[   31.900804] [<c0297b18>] (kernfs_fop_write) from [<c0234558>] (__vfs_write+0x2c/0xf4)
[   31.900907] [<c0234558>] (__vfs_write) from [<c0234d50>] (vfs_write+0xac/0x17c)
[   31.901002] [<c0234d50>] (vfs_write) from [<c0235588>] (SyS_write+0x4c/0xa4)
[   31.901104] [<c0235588>] (SyS_write) from [<c0107180>] (ret_fast_syscall+0x0/0x3c)
[   31.901322] Kernel panic - not syncing: rk808 0-001c: unrecoverable failure
[   31.901322]
[   32.134735] CPU: 0 PID: 547 Comm: system_server Not tainted 4.4.77 #623
[   32.141381] Hardware name: Rockchip (Device Tree)
[   32.146180] [<c010f608>] (unwind_backtrace) from [<c010b800>] (show_stack+0x10/0x14)
[   32.153998] [<c010b800>] (show_stack) from [<c03ca7d4>] (dump_stack+0x7c/0x9c)
[   32.161311] [<c03ca7d4>] (dump_stack) from [<c01ec0d4>] (panic+0x94/0x208)
[   32.168265] [<c01ec0d4>] (panic) from [<c0504cc0>] (dpm_watchdog_handler+0x44/0x4c)
[   32.175999] [<c0504cc0>] (dpm_watchdog_handler) from [<c018cb90>] (call_timer_fn+0xa0/0x20c)
[   32.184515] [<c018cb90>] (call_timer_fn) from [<c018cf3c>] (run_timer_softirq+0x240/0x2cc)
```

Suspend and resume analyzing skill:

1. Check if the system enters suspend or not: the system is in suspend mode if pmic_sleep is high level.
2. Check if each voltage is normal or not in suspend. Measure each voltage when pmic_sleep pin is high.
3. If it is high probability issue, you can set **rockchip,sleep-debug-en** as 1, then it will print more log for suspend and resume, which is convenient for issue analyzing.
4. Check if cutting off some power supply will cause the system unstable during suspend. The analyzing method: remove the configuration of RKPM_SLP_AP_PWROFF in rockchip,sleep-mode-config.

4.2 Standby power consumption analyzing

Check the status of the device:

Check if there is any device not suspending or frequently suspending and resuming.

Check power supply:

For many peripherals in deep sleep mode, the power supply can be closed when they are not used.

Close method:

（1）Initiatively close by the device

Such as the power of LCD backlight or 3.3V IO power of LCD, when the screen is turned off, it will use regulator_disable or pull up/pull down the PIN to close the output in VOP driver.

（2）When using PMIC, pull up the SLEEP pin of deep sleep to close

In PMIC node, configure regulator-off-in-suspend; property will close the output after the sleep pin of deep sleep is pulled up.

Shown as below:

```
vdd_cpu_l: DCDC_REG2 {
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <750000>;
        regulator-max-microvolt = <1350000>;
        regulator-ramp-delay = <6001>;
        regulator-name = "vdd_cpu_l";
        regulator-state-mem {
            regulator-off-in-suspend;
```

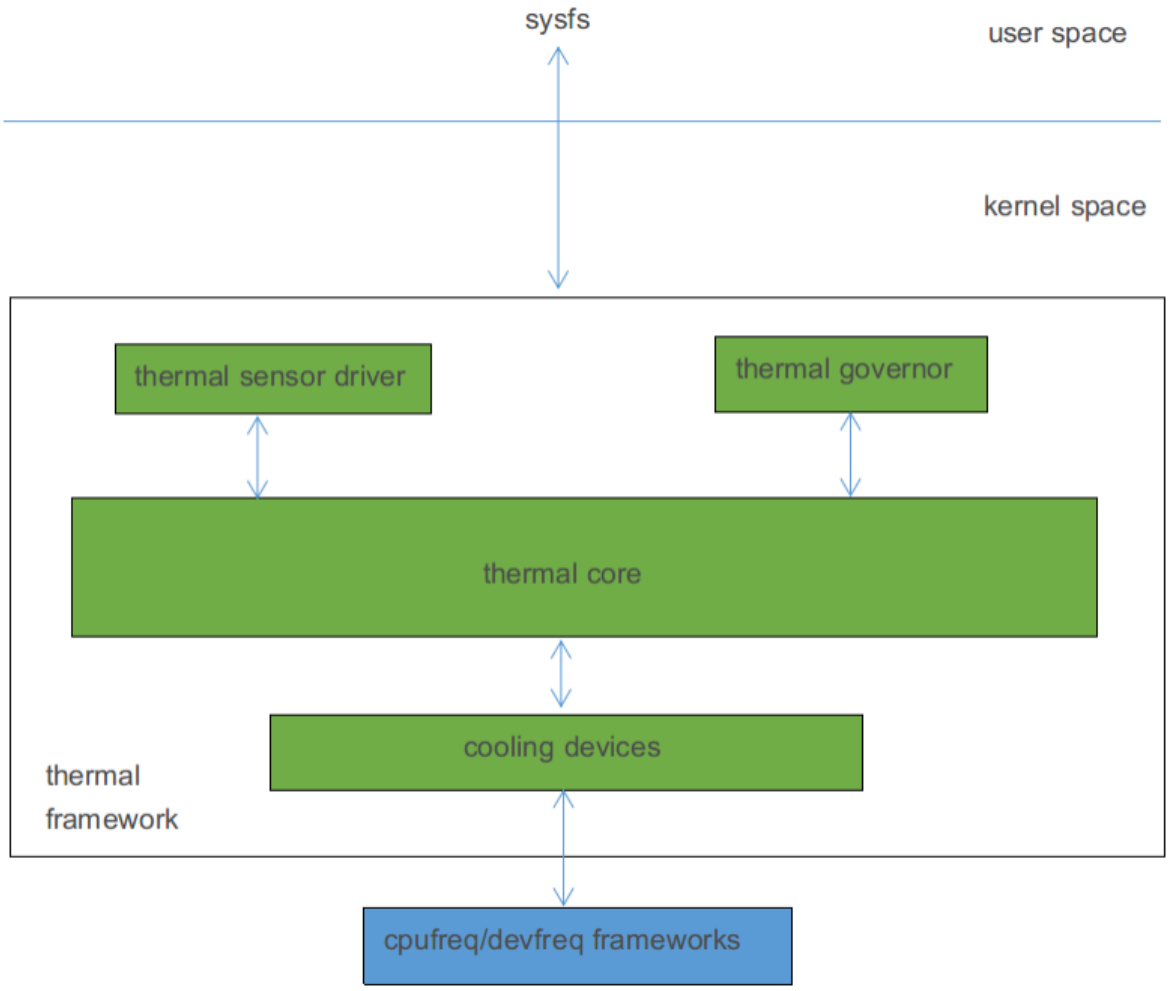
```
        };
    };
```

As for which power supply can be closed, need to check according to the actual hardware.

5 Thermal control strategy introduction

5.1 Thermal overview

Thermal is a set of framework model defined by kernel developer which supports to control the system temperature according to the specified governor in order to prevent the chip from over-heating. Thermal framework consists of governor, core, cooling device, and sensor driver. The software architecture is as below:



Picture 1-10 Thermal framework

5.2 Thermal control parameter adjustment

Assuming that if we want thermal starts to work from 70 degree (more frequently acquire the

temperature), the max temperature not exceed 85 degree, and the system will reboot when over 115 degree. Then need to configure as below:

```
thermal_zones: thermal-zones {
    soc_thermal: soc-thermal {
        ....
        trips {
            threshold: trip-point-0 {
                /*
                 * 70度以上温控开始工作，缩短了获取温度的是时间间隔，但不一定马上降频，
                 * 还跟sustainable-power有关
                 */
                temperature = <70000>; /* millicelsius */
                hysteresis = <2000>; /* millicelsius */
                type = "passive";
            };
            target: trip-point-1 {
                /* 期望最高温度不超过85度 */
                temperature = <85000>; /* millicelsius */
                hysteresis = <2000>; /* millicelsius */
                type = "passive";
            };
            soc_crit: soc-crit {
                /* 超过115度系统重启 */
                temperature = <115000>; /* millicelsius */
                hysteresis = <2000>; /* millicelsius */
```

5.3 User mode interface introduction

The user mode interface is in the directory of /sys/class/thermal/. The detailed contents are corresponding to the configuration of thermal zone node in dtsi. RK3399 platform has two sub nodes, corresponding to the two sub directories thermal_zone0 and thermal_zone1 in the directory of /sys/class/thermal/. Through user mode interface, you can switch thermal control strategy, check current temperature and so on.

Take RK3399 as example, /sys/class/thermal/thermal_zone0/ directory includes the commonly used information as below:

```
temp                    /* 当前温度 */
available_policies      /* 支持的温控策略 */
policy                  /* 当前使用的温控策略 */
sustainable_power       /* 期望的最高温度下对应的power值 */
integral_cutoff         /* PID算法中I的触发条件：当前温度-期望的最高温度<integral_cutoff */
k_d                     /* PID算法中计算D的时候用的参数 */
k_i                     /* PID算法中计算I的时候用的参数 */
k_po                    /* PID算法中计算P的时候用的参数 */
k_pu                    /* PID算法中计算P的时候用的参数 */
mode                    /* enabled：自带定时获取温度，判断是否需要降频。disabled关闭该功能 */
type                    /* 当前thermal zone的类型 */
/* 不同的温度阀值，对应trips节点的配置 */
trip_point_0_hyst
trip_point_0_temp
trip_point_0_type
trip_point_1_hyst
trip_point_1_temp
trip_point_1_type
trip_point_2_hyst
trip_point_2_temp
trip_point_2_type
/* 不同cooling devic的状态，对应cooling-maps节点的配置 */
cdev0                   /* 代表一个cooling devic，有的平台还有cdev1、cdev2等 */
    cur_state           /* 该cooling device当前频率的档位 */
    max_state           /* 该cooling device最多有几个档位 */
    type                /* 该cooling device的类型 */
cdev0_weight            /* 该cooling devic在计算power时扩大的倍数 */
```

5.4 Close the thermal control

Method one: the default thermal control strategy in menuconfifig is set as user_space.

```
<*> Generic Thermal sysfs driver  --->
    --- Generic Thermal sysfs driver
    [*]   APIs to parse thermal data out of device tree
    [*]   Enable writable trip points
          Default Thermal governor (user_space)  --->  /* power_allocator改为user_space */
```

Method two: use command to close the thermal control after power on

Firstly, switch the thermal control strategy to user_space, that is, change the policy node of user mode interface to user_space. Or set the mode as disabled. Then remove the frequency limitation, that is, set cur_state of all cdev in user mode interface as 0.

Take RK3399 as example, switch the strategy to user_space:

```
echo user_space > /sys/class/thermal/thermal_zone0/policy
```

Or set the status of mode as disabled:

```
echo disabled > /sys/class/thermal/thermal_zone0/mode
```

Remove the frequency limitation:

```
/* 具体有多少个cdev，根据实际情况修改 */
echo 0 > /sys/class/thermal/thermal_zone0/cdev0/cur_state
echo 0 > /sys/class/thermal/thermal_zone0/cdev1/cur_state
echo 0 > /sys/class/thermal/thermal_zone0/cdev2/cur_state
```

5.5 Acquire current temperature

Directly check the temp node under the thermal_zone0 or thermal_zone1 directory of user mode interface.

Take RK3399 as example, input the following command through the serial port to acquire CPU temperature:

```
cat /sys/class/thermal/thermal_zone0/temp
```

Input the following command through the serial port to acquire GPU temperature:

```
cat /sys/class/thermal/thermal_zone1/temp
```

6 System stability and power consumption self-analyzing CheckList

No.	Issue type	Check point	Remark	Result feedback
1	Stability	Check if each path voltage is normal or not	Ethernet disconnection, camera stress test crash, hdmi in stress test crash and other issues	
2	Power consumption/Stability	vdd_log actually measured is 0.95V		
3	Stability	LPDDR4:VDD_CENTER actually measured is 0.9V		
4	Stability	Power ripple of VDD_CPU_B, VDD_CPU_L, VDD_GPU, VDD_LOG, VDD_CENTER should be less than 10% (recommended within ±50mV)		
5	Stability	Try to verify with CPU/GPU/DDR fixed frequency		
6	Stability	Try to verify with CPU/GPU increased by 12.5mV times		
7	Stability	When PMUIO2 voltage is configured as 3.0V, need to pull up 10K for PMUIO2_VOLSEL pin	Pin V30 PMUIO2_VOLSEL is used to configure the voltage of PMUIO2	

8	Stability	Check if io-domain configuration in dts matches with schematic or not	Refer to the content in chapter 9.1.2 of RKDocs/rk3399/rk3399_android7.1_software_developer_guide_vxxx.pdf	
9	Power consumption	Check if the module not used in DTS is disabled or not	Disable the modules not used in actual project, such as ISP, EDP, pcie and so on. Appropriately reduce the refresh rate of lcd panel can reduce the power consumption	
10	Power consumption/Stability	Use TOP instruction to check CPU loading of APK	Run apk developed by customer, if the loading of some CPU core is very heavy, you can try to divide the apk task into several threads, which can balance the loading of each CPU core and reduce CPU frequency and the power consumption	
11	Power consumption/Stability	If the heat dissipation solution is reviewed by RK		
12	Performance/Power consumption/Stability	If the schematic/PCB is reviewed by RK		
13	Performance/Power consumption/Stability	Report SDK software version		
14	Performance/Power consumption/Stability	When using LPDDR4, check DDR version	Confirm it is updated to V2.3 or higher version	

7 Power consumption optimization point

The basic power consumption optimization points include: backlight brightness, reduce the panel refresh rate, disable some unused peripherals, adjust target load, limit CPU frequency according to the temperature, and so on. For LPDDR4 device, confirm if the ddr frequency scaling is enabled or not.

8 FAQ

8.1 CPU frequency adjustment failure

The set CPU voltage is not supported. The step voltage of RK pmic is 12.5mV, so when increasing the voltage, it should be 12.5mV times only.

```
[    3.132492] core: _opp_supported_by_regulators: OPP minuV: 880000 maxuV: 880000, not supported by regulator
[    3.132587] cpu cpu0: _opp_add: OPP not supported by regulators (408000000)
[    3.132903] core: _opp_supported_by_regulators: OPP minuV: 880000 maxuV: 880000, not supported by regulator
[    3.133048] cpu cpu0: _opp_add: OPP not supported by regulators (600000000)
[    3.133344] core: _opp_supported_by_regulators: OPP minuV: 880000 maxuV: 880000, not supported by regulator
[    3.133495] cpu cpu0: _opp_add: OPP not supported by regulators (816000000)
[    3.133820] core: _opp_supported_by_regulators: OPP minuV: 930000 maxuV: 930000, not supported by regulator
[    3.133941] cpu cpu0: _opp_add: OPP not supported by regulators (1008000000)
[    3.134245] core: _opp_supported_by_regulators: OPP minuV: 1005000 maxuV: 1005000, not supported by regulator
[    3.134388] cpu cpu0: _opp_add: OPP not supported by regulators (1200000000)
```

8.2 Standby power consumption is too high

The ddr frequency scaling is not enabled in dts. It is running with the highest frequency.

```
&dmc {
        center-supply = <&vdd_logic>;
        status = "okay";
};
```

8.3 Overall running power consumption is too high

We find some customers increase cpu, gpu, ddr voltage too much and cause the power consumption much higher than RK sdk board, because the ripple of their hardware board is relatively large. The voltage should be increased by 12.5mV each time. As long as the system is stable, the less increase, the better.

8.4 Image composition strategy: the power consumption of HWC composition is lower than GPU composition

Check system property sys.hwc.compose_policy, 0 means to use GPU composition, and 6 means to use hwc composition. It is recommended to use hwc composition if possible. For example, for the scenario of playing 4K video, the power consumption of hwc composition is around 4.2V-200mA less than gpu composition.

8.5 Suspend power consumption is too high

Check the voltage configuration in regulators. Below shows an example of vdd_gpu voltage configuration issue which will lead to the high power consumption in suspend. Recommend customers not to modify the voltage in regulators arbitrarily.

```
vdd_gpu: DCDC_REG2 {
        regulator-name = "vdd_gpu";
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <800000>;
        regulator-max-microvolt = <1250000>;
        regulator-ramp-delay = <6001>;
        regulator-state-mem {
                regulator-off-in-suspend;
                regulator-suspend-microvolt = <950000>;
        };
};
```

正确配置

```
vdd_2v3: DCDC_REG2 {
        regulator-name = "vdd_2v3";
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <1100000>;
        regulator-max-microvolt = <1100000>;
        regulator-ramp-delay = <6001>;
        regulator-state-mem {
                regulator-off-in-suspend;
                regulator-suspend-microvolt = <1100000>;
        };
};
```

8.6 Device reset during high temperature stress test with 60℃

Check if the over-temperature protection temperature is correct or not, configured in kernel/arch/arm64/boot/dts/rockchip/rk3399.dtsi.

```
trips {
        threshold: trip-point-0 {
                temperature = <70000>; /* millicelsius */
                hysteresis = <2000>; /* millicelsius */
                type = "passive";
        };
        target: trip-point-1 {
                temperature = <85000>; /* millicelsius */
                hysteresis = <2000>; /* millicelsius */
                type = "passive";
        };
        soc_crit: soc-crit {
                temperature = <115000>; /* millicelsius */
                hysteresis = <2000>; /* millicelsius */
                type = "critical";
        };
};
```

软件过温保护

```
tsadc: tsadc@ff260000 {
        compatible = "rockchip,rk3399-tsadc";
        reg = <0x0 0xff260000 0x0 0x100>;
        interrupts = <GIC_SPI 97 IRQ_TYPE_LEVEL_HIGH 0>;
        rockchip,grf = <&grf>;
        clocks = <&cru SCLK_TSADC>, <&cru PCLK_TSADC>;
        clock-names = "tsadc", "apb_pclk";
        assigned-clocks = <&cru SCLK_TSADC>;
        assigned-clock-rates = <750000>;
        resets = <&cru SRST_TSADC>;
        reset-names = "tsadc-apb";
        pinctrl-names = "init", "default", "sleep";
        pinctrl-0 = <&otp_gpio>;
        pinctrl-1 = <&otp_out>;
        pinctrl-2 = <&otp_gpio>;
        #thermal-sensor-cells = <1>;
        rockchip,hw-tshut-temp = <120000>;
        status = "disabled";
};
```

硬件保护关机温度

8.7 Device reset when CPU temperature is around 80℃ during high temperature stress test

The over-temperature protection of software is already configured as 115℃, and the over-temperature protection of hardware is already configured as 120℃, but when CPU temperature is 80℃ the hardware shutdown happens without any log. The phenomenon may be because the MOS tube in below picture is turned on directly at 80℃, which will pull up the reset pin of RK808, and cause the hardware reset. The solution is: change better mos tube.