

---

# 高通多媒体技术期刊 20150408

---



Qualcomm Technologies, Inc.

Confidential and Proprietary – Qualcomm Technologies, Inc.

机密和专有信息——高通技术股份有限公司



# Confidential and Proprietary – Qualcomm Technologies, Inc.

---

## Confidential and Proprietary – Qualcomm Technologies, Inc.

**NO PUBLIC DISCLOSURE PERMITTED:** Please report postings of this document on public servers or web sites to: [DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com). **禁止公开：**如在公共服务器或网站上发现本文档，请报告至：[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com).

**Restricted Distribution:** Not to be distributed to anyone who is not an employee of either Qualcomm or its affiliated without the express approval of Qualcomm's Configuration Management. **限制分发：**未经高通配置管理部门的明示批准，不得发布给任何非高通或高通附属及关联公司员工的人。 Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc. 未经高通技术股份有限公司明示的书面允许，不得使用、复印、复制、或修改全部或部分文档，不得以任何形式向他人透露其内容。

The user of this documentation acknowledges and agrees that any Chinese text and/or translation herein shall be for reference purposes only and that in the event of any conflict between the English text and/or version and the Chinese text and/or version, the English text and/or version shall be controlling. 本文档的用户知悉并同意中文文本和/或翻译仅供参考之目的，如英文文本和/或版本和中文文本和/或版本之间存在冲突，以英文文本和/或版本为准。

This document contains confidential and proprietary information and must be shredded when discarded. 未经高通明示的书面允许，不得使用、复印、复制全部或部分文档，不得以任何形式向他人透露其内容。本文档含有高通机密和专有信息，丢弃时必须粉碎销毁。

Qualcomm reserves the right to make changes to the product(s) or information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an "as is" basis. 高通保留未经通知即修改本档中提及的产品或信息的权利。本公司对使用或应用本文档所产生的直接或间接损失概不负责。本文档中的信息为基于现状所提供，使用风险由用户自行承担。

Qualcomm is a trademark of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners. Qualcomm是高通公司在美国及其它国家注册的商标。所有高通公司的商标皆获得使用许可。其它产品和品牌名称可能为其各自所有者的商标或注册商标。

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited. 本文档及所含技术资料可能受美国和国际出口、再出口或转移出口法律的 限制。严禁违反或偏离美国和国际的相关法律。

**Qualcomm Technologies, Inc. 5775 Morehouse Drive San Diego, CA 92121 U.S.A.**

高通技术股份有限公司，美国加利福尼亚州圣地亚哥市莫豪斯路 5775 号，邮编 92121

# Revision History

---

Revision	Date	Description
A	April 2015	Initial release

**Note:** There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

# 内容

---

- Display
  - 如何debug wait4pingpong timeout 问题
    - 查看Panel的TE 信号
    - 查看TE pin脚的配置
    - 调试wait4pingpong timeout的方法
    - 对log的分析
    - Dump MDP相关的TE config的寄存器
  - 背光调试
    - 背光的使能
    - 背光常见问题的调试



---

# Display

---

# 如何debug wait4pingpong timeout 问题

- 对于command mode panel , 经常会发生wait4pingpong timeout的问题 , 比如下面log :

```
W/Kernel ( 0): <4>[ 2261.021620] -----[ cut here ]-----
W/Kernel ( 0): <4>[ 2261.021656] WARNING: at
/home/project/kernel/drivers/video/msm/mdss/mdss_mdp_intf_cmd.c:617
mdss_mdp_cmd_wait4pingpong+0x244/0x3b8()
I/Kernel ( 0): <6>[ 2261.021667] cmd kickoff timed out (0) ctl=0
I/Kernel ( 0): <6>[ 2261.021671] Modules linked in: wlan(O) texfat(PO)
I/Kernel ( 0): <6>[ 2261.021692] CPU: 4 PID: 5782 Comm: mdss_fb0 Tainted: P      O
3.10.49-perf-g6f05a8e-00363-ga132c4f-dirty #455
I/Kernel ( 0): <6>[ 2261.021698] Call trace:
I/Kernel ( 0): <6>[ 2261.021711] [<fffffc0000879b0>] dump_backtrace+0x0/0x214
I/Kernel ( 0): <6>[ 2261.021720] [<fffffc000087bd4>] show_stack+0x10/0x1c
I/Kernel ( 0): <6>[ 2261.021731] [<fffffc00099a920>] dump_stack+0x1c/0x28
I/Kernel ( 0): <6>[ 2261.021740] [<fffffc00009b3b4>] warn_slowpath_common+0x74/0x9c
I/Kernel ( 0): <6>[ 2261.021748] [<fffffc00009b428>] warn_slowpath_fmt+0x4c/0x58
I/Kernel ( 0): <6>[ 2261.021757] [<fffffc00030b210>]
mdss_mdp_cmd_wait4pingpong+0x240/0x3b8
I/Kernel ( 0): <6>[ 2261.021767] [<fffffc0002f6c50>] mdss_mdp_display_commit+0x710/0xaac
I/Kernel ( 0): <6>[ 2261.021776] [<fffffc00031285c>] mdss_mdp_overlay_kickoff+0x578/0xadc
I/Kernel ( 0): <6>[ 2261.021787] [<fffffc00032e89c>] __mdss_fb_display_thread+0x12c/0x2c4
I/Kernel ( 0): <6>[ 2261.021797] [<fffffc0000bbe60>] kthread+0xac/0xb8
```

## 查看Panel的TE信号和配置

---

- 对于command mode panel，HW TE信号来自于panel端，故通过示波器查看TE的信号是否正常。
- 如果TE信号不正常，同时需要与Panel vendor来一起查看。
- 如果TE信号的波形正常，接下来需要查看TE的pin脚是否配置正确。
- 对于不同的chipset，TE 的GPIO有所不同，请查看相关的GPIO手册。

# 查看TE pin的config

- 在dtsi中，举例如下，对于8994来说，TE的pin 脚为GPIO 10

```
--- a/arch/arm/boot/dts/qcom/msm8994-xxx.dtsi
+++ b/arch/arm/boot/dts/qcom/msm8994-xxx.dtsi
@@ -293,11 +293,16 @@
qcom,pins = <&gp 78>;
};
```

```
+&pmx_mdss_te {
+ qcom,num-grp-pins = <1>;
+ qcom,pins = <&gp 10>;
+};
```

```
&mdss_dsi0 {
qcom,dsi-pref-prim-pan = <&dsi_r63419_wqhd_cmd0>;
pinctrl-names = "mdss_default", "mdss_sleep";
- pinctrl-0 = <&mdss_dsi_active>;
- pinctrl-1 = <&mdss_dsi_suspend>;
+ pinctrl-0 = <&mdss_dsi_active &mdss_te_active>;
+ pinctrl-1 = <&mdss_dsi_suspend &mdss_te_suspend>;
qcom,dsi-panel-bias-vreg;
qcom,platform-reset-gpio = <&msm_gpio 78 0>;
qcom,regulator-ldo-mode;
```



# 调试wait4pingpong timeout的方法

---

- 对于此问题，xlog的信息可以有效地帮助定位问题的所在。
- 目前xlog默认为enable，请看下面link：

msm: mdss: debugfs: xlog: enable xlog debug by default

[https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?h=LA.BF.2.1\\_rb1.10&id=3a24c32d82ac25d6801c54aa7dd546b9430a5be3](https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?h=LA.BF.2.1_rb1.10&id=3a24c32d82ac25d6801c54aa7dd546b9430a5be3)

- 1：提高 panel 的FPS 来查看是否对问题有所改善，例如  
qcom,mdss-dsi-panel-framerate= 65 或者70

# 调试wait4pingpong timeout的方法 – 续一

- 2 : 增加patch来查看TE信号是否到来 , 如下 ,

- qcom,platform-te-gpio = <&msm\_gpio 10 0>; // 以8994为例

- 增加 HW vsync test 函数

```
diff --git a/drivers/video/msm/mdss/mdss_dsi.c b/drivers/video/msm/mdss/mdss_dsi.c
```

```
index 660e400..7c249bf 100644
```

```
--- a/drivers/video/msm/mdss/mdss_dsi.c
```

```
+++ b/drivers/video/msm/mdss/mdss_dsi.c
```

```
@@ -926,6 +926,13 @@ end:
```

```
    return dsi_pan_node;
```

```
}
```

```
+ irqreturn_t test_hw_vsync_handler(int irq, void *data)
```

```
+ {
```

```
+ pr_err("HW VSYNC\n");
```

```
+ 
```

```
+ return IRQ_HANDLED;
```

```
+ }
```

■

## 调试wait4pingpong timeout的方法 – 续二

```
static int __devinit mdss_dsi_ctrl_probe(struct platform_device *pdev)
{
    int rc = 0;
@@ -1047,8 +1054,12 @@ static int __devinit mdss_dsi_ctrl_probe(struct platform_device
*pdev)
    }

    pr_debug("%s: Dsi Ctrl->%d initialized\n", __func__, index);
-    return 0;

+    f (devm_request_irq(&pdev->dev, gpio_to_irq(ctrl_pdata->disp_te_gpio),
test_hw_vsync_handler, IRQF_TRIGGER_FALLING, "VSYNC_GPIO", NULL))
+    {
+        pr_err("request_irq failed.\n");
+    }
+    return 0;

error_pan_node:
    of_node_put(dsi_pan_node);
error_vreg:
```

## 调试wait4pingpong timeout的方法 – 续三

- 3 : 在mdss\_mdp\_cmd\_pingpong\_done() 函数中，把pr\_debug改为pr\_err，可以帮助跟踪PP done interrupts 是否从MDP h/w 端触发。

- 4 : 为了debug的意图，可以使用模拟的TE信号，如下：

```
qcom,mdss-tear-check-sync-cfg-height = <...>; /* Height + VBP + VFP + VSW */
qcom,mdss-tear-check-sync-init-val = <...>; /* Height */
qcom,mdss-tear-check-sync-threshold-start = <4>;
qcom,mdss-tear-check-sync-threshold-continue = <4>;
qcom,mdss-tear-check-start-pos = <...>; /* Height */
qcom,mdss-tear-check-rd-ptr-trigger-intr = <...>; /* Height + 1 */
```

对于上面的变量的描述如下：

```
mdss-tear-check-sync-init-val = ( Panel active height )
mdss-tear-check-sync-cfg-height = ( Panel active height + VBP, VFP, VSW)
mdss-tear-check-rd-ptr-trigger-intr = ( Panel active height + 1 )
mdss-tear-check-start-pos = ( Panel active height )
```

同时，可以阅读下面的文件：

/kernel/Documentation/devicetree/bindings/fb/mdss-dsi-panel.txt

# 对wait4pingpong timeout log的分析

- 比如，suspend-resume 测试，有时候panel不能被唤醒，同时有wait4pingpong timeout log，请看下面log的分析：
- **Good logs：**
  - E/Kernel ( 0): <3>[ 596.696892] mdss\_dsi\_panel\_reset: enable 5 rst\_gpio=1
  - E/Kernel ( 0): <3>[ 596.717262] mdss\_dsi\_start\_hs\_clk\_lane: ndx=0, set\_hs, cnt=1
  - E/Kernel ( 0): <3>[ 596.848092] mdss\_dsi\_stop\_hs\_clk\_lane: ndx=0, cnt=0
  - E/Kernel ( 0): <3>[ 596.848105] lcd::mipi\_disp\_inc sharp 1080p panel
  - E/Kernel ( 0): <3>[ 596.848119] mdss\_dsi\_start\_hs\_clk\_lane: ndx=0, set\_hs, cnt=1
  - E/Kernel ( 0): <3>[ **596.855791**] HW VSYNC // **TE signal的时间点**
  - E/Kernel ( 0): <3>[ 596.857468] mdss\_dsi\_stop\_hs\_clk\_lane: ndx=0, cnt=0
  - E/Kernel ( 0): <3>[ 596.857488] mdss\_dsi\_start\_hs\_clk\_lane: ndx=0, set\_hs, cnt=1
  - E/Kernel ( 0): <3>[ 596.865846] mdss\_dsi\_stop\_hs\_clk\_lane: ndx=0, cnt=0
  - E/Kernel ( 0): <3>[ **596.872651**] HW VSYNC // **596.872651 - 596.855791 = 16.86ms**
  - E/Kernel ( 0): <3>[ 596.876463] mdss\_dsi\_start\_hs\_clk\_lane: ndx=0, set\_hs, cnt=1
- 从上面logs来看，HW TE信号的产生正常。

# 对wait4pingpong timeout log的分析 –续–

## ▪ Bad logs:

- E/Kernel ( 0): <3>[ **2260.657187**] mdss\_dsi\_panel\_reset: enable 5 rst\_gpio=1 -> resume for the issue case
  - E/Kernel ( 0): <3>[ 2260.873857] mdss\_dsi\_start\_hs\_clk\_lane: ndx=0, set\_hs, cnt=1 -> frame kickoff happened and not TE signal till next suspend/resume
  - .....
  - I/Kernel ( 0): <6>[ 2261.021667] cmd kickoff timed out (0) ctl=0 -> **issue happens**
  - .....
  - E/Kernel ( 0): <3>[ **4341.546745**] mdss\_dsi\_panel\_reset: enable 5 rst\_gpio=1 -> next resume, no issue is seen
  - E/Kernel ( 0): <3>[ 4341.769660] HW VSYNC -> first TE signal from the panel after resume.
- 
- 从上面logs来看，当问题发生时，没有收到 panel TE的信号，Not see the log “HW VSYNC” when the issue is seen after resume.
  - 实际上，No TE signal coming from the panel until the next suspend/resume. (No TE signal from the panel from **2260.657187** to **4341.546745**).

# Dump MDP相关的TE config的寄存器

- 以8994为例，MDP register dump如下：
- // 0xFD971000 ~ 0xFD97103C  
# Register offset and length in hex #echo 71000 3C > /d/mdp/off  
#cat /d/mdp/reg
  - 0x00071000: 00000001 0018007c 0000fff0 00000000
  - 0x00071010: 00000a00 0000767b 00040004 00000a00
  - 0x00071020: 00000a01 00000000 00000000 00000000
  - 0x00071030: 00000000 00000000 00000000
- // 0xFD971800 ~ 0xFD97183C  
# Register offset and length in hex #echo 71800 3C > /d/mdp/off  
#cat /d/mdp/reg
  - 0x00071800: 00000001 0018007c 0000fff0 00000000
  - 0x00071810: 00000a00 00006621 00040004 00000a00
  - 0x00071820: 00000a01 00000000 00000000 00000000
  - 0x00071830: 00000000 00000000 00000000

# 背光的使能

- 背光的使能是显示bring up中重要的一环，在使能背光之前，请先和H/W工程师确认屏的背光类型
- 背光控制类型
  - - qcom,mdss-dsi-bl-pmic-control-type:dtsti文件中指明背光控制类型的参数，一般有三种类型，如下
  - "bl\_ctrl\_pwm" = 背光由PWM控制.
  - "bl\_ctrl\_wled" = 背光由WLED控制.
  - "bl\_ctrl\_dcs" = 背光由DCS命令控制.
  - other: Unknown backlight control. (default)
  - 您也可以在如下帮助文件中找到具体的说明

Kernel/Documentation/devicetree/bindings/fb/mdss-dsi-panel.txt

- 对于bl\_ctrl\_pwm背光控制类型，需要在dtsti文件中另外设置如下参数：
  - - qcom,mdss-dsi-bl-pmic-bank-select: LPG channel for backlight.
  - - qcom,mdss-dsi-bl-pmic-pwm-frequency:PWM period in microseconds.
  - - qcom,mdss-dsi-pwm-gpio:PMIC gpio binding to backlight.



# 背光常见问题的调试

- 背光抖动问题

- 首先请使能mdss\_fb.c中的dynamic debug信息来查看背光等级的变化

```
echo -n "file mdss_fb.c +tp" > d/dynamic_debug/control
```

- 如果背光等级变化非常平滑，则可排除背光软件问题，请确认是否是显示抖动，或者背光IC的硬件问题
- 如果背光等级变化不平滑，举例如下

```
[ 10.244546] mdss_fb_set_backlight: mdss_fb_set_backlight bkl_lvl = 0
```

```
[ 18.314944] mdss_fb_set_backlight: mdss_fb_set_backlight bkl_lvl = 255
```

```
[ 18.913691] mdss_fb_set_backlight: mdss_fb_set_backlight bkl_lvl = 10
```

```
[ 18.996094] mdss_fb_set_backlight: mdss_fb_set_backlight bkl_lvl = 255
```

我们可以看到背光等级跳动在短时间内非常厉害，此时请先排除如下影响

- 1). disable CABL:可从QualcommSetting关闭或者在build.prop设置ro.qualcomm.cabl=0
- 2). 排除lightsensor的影响:请关闭根据环境光调节背光功能

# 背光常见问题的调试- 续一

- 唤醒时背光比显示先亮问题，具体表现就是唤醒时的白屏或者黑屏问题

- 请添加如下debug信息

```
--- a/drivers/video/msm/mdss/mdss_fb.c
+++ b/drivers/video/msm/mdss/mdss_fb.c
@@ -1005,6 +1011,7 @@ static int mdss_fb_blank_sub(int blank_mode, struct fb_info *info,
    switch (blank_mode) {
    case FB_BLANK_UNBLANK:
+
+                pr_err("[DEBUG] UNBLANK START!\n");
+                if (!mfd->panel_power_on && mfd->mdp.on_fnc) {
+                    ret = mfd->mdp.on_fnc(mfd);
+                    if (ret == 0) {
@@ -1028,6 +1035,7 @@ static int mdss_fb_blank_sub(int blank_mode, struct fb_info *info,
+                    mdss_fb_set_backlight(mfd, mfd->unset_bl_level);
+                }
+                mutex_unlock(&mfd->bl_lock);
+                pr_err("[DEBUG] UNBLANK DONE!\n");
+                break;
--- a/drivers/video/msm/mdss/mdss_mdp_overlay.c
+++ b/drivers/video/msm/mdss/mdss_mdp_overlay.c
@@ -1365,6 +1365,7 @@ int mdss_mdp_overlay_kickoff(struct msm_fb_data_type *mfd,
    bool need_cleanup = false;
    struct mdss_mdp_commit_cb commit_cb;
+
+    pr_err("[DEBUG] KICKOFF!\n");
+    ATRACE_BEGIN(__func__);
    if (ctl->shared_lock)
        mutex_lock(ctl->shared_lock);
```

- `echo -n "file mdss_fb.c +tp" > d/dynamic_debug/control`

- 请提case到高通来帮忙解决

---

## Questions?

<https://support.cdmatech.com>

