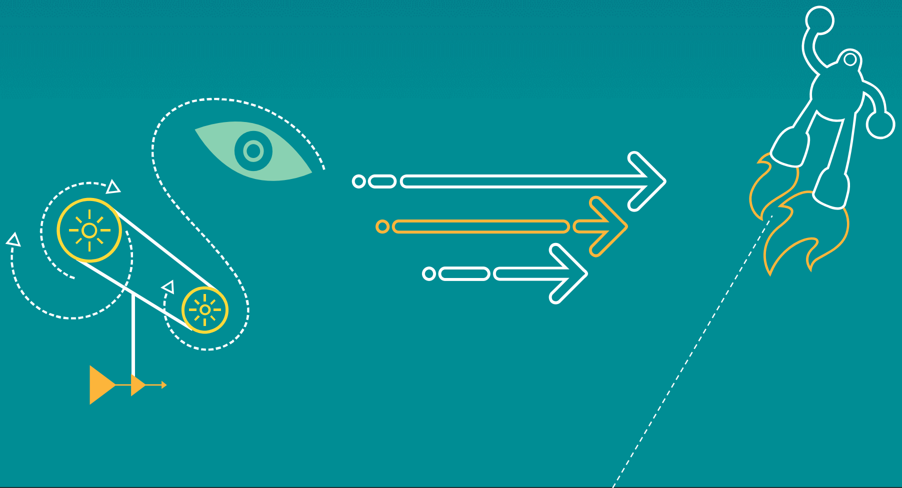
高通多媒体技术期刊　20150923

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. **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.

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文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	Sep 2015	Initial release

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

内容

- Display
  - MDSS HW cursor 讨论和关键fix推送
  - MDSS kernel 线程讨论和关键fix推送
  - MDSS debugfs 和sysfs应用最新总结
- Audio
  - Important docs update
  - Audio bring up of external WCD codec
  - Power supply for external WCD codec
  - MCLK configuration for external WCD codec
  - Audio common issues

Display

HW cursor 相关的关键fix

- **在最近的8994，8992 以及 8996平台上，我们对HW_cursor 提供了支持，对系统的功耗和性能有所改善。**

HW cursor软件引入了一些bug，下面的fix提供了解决办法。

1.ARM: dts: msm: add clk ctrl configs for cursor pipe on msm8992/msm8994

https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?id=9f833e7a57d55dd4751765945cbb24c9fd6417cb

2. msm: mdss: always assign virtual address to cursor buffer

https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?id=ad2f67ccf2e11c8c398b645cb255280d21e85f4c

3. msm: mdss: add cursor logic to get the src crop

https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?h=LA.BF.2.1_rb1.10&id=4282a4bf1b764f61c2e7dbfaa36e483328a343c5

4.msm: mdss: exclude cursor from dirty pipe logic

https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?h=msm-3.10&id=700adab07419cc2d867a55e205cd895a4a165870

MDSS kernel 线程讨论和关键fix推送

- **MDSS_fb0和ESD DSI event thread是 MDSS kernel驱动中的两个重要线程。他们中间的同步如果发生问题或者是死锁，很容易导致SurfaceFlinger service出现问题，进而导致framework reboot。**

  这个fix主要解决了mdss_fb0 线程和ESD线程之间的同步问题，问题的根源在于Linux kernel本身的同步方法的bug。

  1. workqueue: fix hang involving racing cancel[_delayed]_work_sync()'s for PREEMPT_NONE

  https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/commit/?id=8603e1b30027f943cc9c1eef2b291d42c3347af1

MDSS debugfs 和sysfs应用最新总结

- **使用 adb shell直接操作debugfs 和sysfs进行MDSS的相关调试，较为便捷，避免了重新编译的繁琐。**

1. 首先可以使用 debugfs，打开MDSS的 kernel logs。

比如

```
adb shell "echo -n 'file mdss_dsi.c +tp'> /d/dynamic_debug/control"
adb shell " echo -n ‘func mdss_mdp_ad_input +tp’ > /d/dynamic_debug/control “
```

2. /d/mdp 里面可以直接dump mdp和DSI相关的各种寄存器。

```
adb shell "echo 0x0 0x90000 > /d/mdp/off "
adb shell cat /d/mdp/reg > C:\Desktop\mdss_reg.txt
0x00000000: 10090000 00000000 00000000 00000000
0x00000010: 00000000 00000000 00000000 00000000
 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 adb shell "echo 0x0 0x260 > /d/mdp/dsi0_ctrl_off"
 adb shell cat /d/mdp/dsi0_ctrl_reg > C:\Desktop\dsi_reg.txt
 0x00000000: 10030000 000001f5 00000000 11111300
 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

MDSS debugfs 和sysfs应用最新总结

3. /d/mdp/stat 可以查看当前mdp的运行状态

```
root@msm8992:/d/mdp # cat stat
cat stat
mdp:
intf2: play: 00000000   vsync: 00004002        underrun: 00000000
user_bl: 00000000      mod_bl: 00000000

VIG0 :   00000002      VIG1 :   00000002      VIG2 :   00000000
RGB0 :   00000342      RGB1 :   00000002      RGB2 :   00000000
DMA0 :   00000030      DMA1 :   00000002
```

4.

serialize_wait4pp可以控制mdss kernel frame kick off 的 流程，用于复杂场景抓取寄存器.在设置完串行pingpong之后，可以去复现问题，抓取Xlog，这样得到的MDSS 寄存器dump ，便于debug 分析，防止前后帧之间的寄存器覆盖。

```
root@msm8992:/d/mdp # echo 1 > serialize_wait4pp
echo 1 > serialize_wait4pp
```

MDSS debugfs 和sysfs应用最新总结

5.

/d/mdp/dump可以查看当前mixer和SSPP的分配情况。

```
cat dump
--[ Control path #0 - MIPI_VIDEO]--
MDP Clk=259459200  Final BW=1359072000
Play Count=15  Underrun Count=0

Intf Mixer #0  res=1600x2560

SSPP #1 type=VIG ndx=2 flags=0x00000000 play_cnt=2
 stage=0 alpha=0x0 transp=0x0 blend_op=0
 src w=0 h=0 format=5 (RGB_888)
 src_rect x=0 y=0 w=800 h=2560 H.dec=0 V.dec=0
 dst_rect x=0 y=0 w=800 h=2560
 SMP allocated=[0 0] reserved=[0 0]
  Data:

SSPP #3 type=RGB ndx=8 flags=0x00220000 play_cnt=1
stage=2 alpha=0xff transp=0xffffffff blend_op=2
src w=1664 h=2560 format=13 (RGBA_8888)
src_rect x=0 y=0 w=1600 h=2560 H.dec=0 V.dec=0
dst_rect x=0 y=0 w=1600 h=2560
SMP allocated=[0 0] reserved=[0 0]
Data:
state=ACTIVE  addr=0x0000000001400000 size=17039360 ihdl=0xffffffc0336fea80 alloc_time=22056.665811306
```

MDSS debugfs 和sysfs应用最新总结

6.

/sys/class/graphics/fb0/mdp # cat caps 可以查看当前 MDSS 软硬件的状态，以及某些feature是否开启。

```
root@msm8992:/sys/class/graphics/fb0/mdp # cat caps
cat caps
mdp_version=5
hw_rev=269090816
rgb_pipes=3
vig_pipes=3
dma_pipes=2
blending_stages=4
cursor_pipes=1
max_cursor_size=128
smp_count=34
smp_size=8192
smp_mb_per_pipe=0
max_downscale_ratio=4
max_upscale_ratio=20
props=31488
max_bandwidth_low=7000000
max_bandwidth_high=7000000
max_pipe_width=2048
max_mixer_width=4096
features= decimation tile_format src_split rotator_downscale
```

Audio

Important docs update

- 80-NV396-50 (B) MSM8996 AUDIO BRINGUP GUIDE
- Solution#00031096 - How to enable PM8956 internal codec on MSM8956/MSM8976 platform

Audio bring up of external WCD codec

- 检查是否ADSP6已经起来。如有如下打印，表明ADSP6已经起来了
  - apr_tal:Q6 Is Up
- 检查是否Slimbus已经起来。如有如下打印，表明Slimbus已经起来了；如果没有，检查codec的供电，没有给codec供电可能是原因之一，见下页（page 4）检查codec的供电配置。
  - [30.281382] slimbus:1 laddr:0xcb, EAPC:0x1:0xa0
  - [30.282444] wcd9335-slim tasha-slim-pgd: wcd9xxx_slim_device_up: slim device up
  - [30.282933] slimbus:1 laddr:0xca, EAPC:0x0:0xa0
  - [30.285257] wcd9335-slim tasha-slim-pgd: wcd9xxx_check_codec_type: detected tasha_codec, major 0x0, minor 0x0, ver 0x1
- 查看声卡是否已经注册成功
  - root@msm8996:/ # cat /proc/asound/cards
  - 0 [msm8996tashacdp]: msm8996-tasha-c - msm8996-tasha-cdp-snd-card
- 用tinymix、tinyplay命令验证是否播放有声音，如果没有，检查mixer path是否和硬件匹配；检查是否有MCLK，见page 5

Power supply for external WCD codec

- 首先查看硬件原理图，VDD_BUCK是codec的供电， VDD_BUCK一般和LDO连接来控制是否供电，检查LDO的IO控制引脚的连接，8956参考设计其连接PM的GPIO_07
- 配置dtsi文件
  - 查阅PMIC的GPIO对应的地址，根据spec配置(请联系PMIC team) ，可参考已经有的dtsi文件配置
  - 例如：连接PM8952的gpio7作为codec的控制供电的LDO的IO

```
gpio@c600 {
        status = "ok";
        qcom,mode = <1>;
        qcom,pull = <5>;
        qcom,vin-sel = <0>;
        qcom,src-sel = <0>;
        qcom,master-en = <1>;
};
```

MCLK configuration for external WCD codec

- 首先查看硬件原理图，确认WCD Codec的MCLK连接，8956参考设计连接PM的GPIO_01
- 配置dtsi文件
  - 查阅PMIC的MCLK的GPIO对应的地址，根据spec配置(请联系PMIC team)，可参考已经有的dtsi文件配置
  - 例如：连接PM8952的gpio1作为MCLK源
    - &pm8950_gpios {
    - gpio@c000 {
    - status = "ok";
    - qcom,mode = <1>;
    - qcom,pull = <5>;
    - qcom,vin-sel = <2>;
    - qcom,src-sel = <2>;
    - qcom,master-en = <1>;
    - qcom,out-strength = <2>;
    - };
  - 可Dump GPIO寄存器查看是否配置正确
    - mount -t debugfs none /sys/kernel/debug
    - cd /sys/kernel/debug/spmi/spmi-0
    - echo 0x100 > count
    - echo 0xc000 > address
    - cat data
- MCLK的使能，代码在audio-ext-clk.c，不需要修改，默认代码可以工作

Audio common issues（1）

- 描述：no ringtone and music sound during QQ video call
- 复现步骤和现象：
  - 1. Open QQ (version 5.8.1), set up a QQ video call, turn off loudspeaker.
  - 2. Click back key (3 times) to run QQ in background.
  - 3. Make a MT call, DUT will play ringtone, the ringtone is normal (have sound), pick up the call, say anything then hang up the call.
  - 4. Make a second MT call, the ringtone have no sound >>> ISSUE.
  - 5. Play music, music also have no sound. >>>ISSUE.
- 基线： MSM8939.LA.2.0.2, LA.BR.1
- CR： 899756
- 代码修改：
  - https://www.codeaurora.org/cgit/quic/la/platform/frameworks/av/commit/?id=b3a87049b373cecee53e16fc8463a07389300832
  - https://www.codeaurora.org/cgit/quic/la/platform/hardware/qcom/audio/commit/?id=0b04de357865bd39d30414b5e131506ff90b9386

Audio common issues （2）

- 描述：pop noise when switch to next music with DTS SRS open
- 复现步骤和现象：
  - 1. Enable DTS SRS, using the DSP solution
  - 2. Play music in headset
  - 3. Repeat switch to next song
  - 4. Will hear some pop noise in headset sometimes.
- 基线： MSM8909.LA.1.1, MPSS.JO.1.0
- CR： 873740
- 代码修改：
  - 在DSP，请申请SBA

Audio common issues（3）

- 描述：play next muic, progress bard doesn't move for a while
- 复现步骤和现象：
  - 1. Enter Music Player(default Music Player), play one music
  - 2. Seek to about 10s before the end
  - 3. Wait for this song is end and player swithes to next song
  - 4. Next song plays normal, but progress bad dosn't move for several seconds
- 基线： MSM8939.LA.2.0, LA.BR.1
- CR： 866102
- 代码修改：
  - https://www.codeaurora.org/cgit/quic/la/platform/frameworks/av/commit/?h=LA.BR.1.3.2&id=0463bc2c49be36862c15b39855f61887ea90bd41

Questions?

https://support.cdmatech.com