



# Tina Linux 显示量产 排查指南

## 快速排查指南

版本号: 1.0

发布日期: 2024.12.7

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2024.12.7	AWA2257	初始版本



# 目 录

<b>1 前言</b>	<b>1</b>
<b>2 LCD 调试说明</b>	<b>2</b>
2.1 LCD 接口调试方法	2
2.1.1 LCD 显示状态调试	2
2.1.2 LCD 电源状态调试	3
2.1.3 LCD PWM 状态调试	3
2.1.4 LCD PIN 引脚状态调试	4
2.1.5 LCD CLK 时钟配置调试	5
2.1.6 LCD 接口 colorbar 调试	5
2.2 LCD 常见问题	7
2.2.1 黑屏（无背光）	7
2.2.2 黑屏（有背光）	7
2.2.3 闪屏	8
2.2.4 白屏	8
2.2.5 雪花屏	8
2.2.6 图像抖动	9
2.2.7 条形波纹	9
2.2.8 背光太亮或者太暗	9
<b>3 G2D 调试说明</b>	<b>10</b>
3.1 设备节点	10
3.2 传递参数信息	10
3.3 G2D 问题指南	11
3.3.1 invalid address	11
3.3.2 dmabuf get fd failed	11
3.3.3 dma_map_sg failed	11
3.3.4 旋转画面撕裂	12
3.3.5 旋转系统挂死	12
<b>4 KSC 调试说明</b>	<b>13</b>
4.1 ksc 调试手段	13
4.1.1 ksc 状态信息打印	13
4.1.1.1 内核	13
4.1.1.2 uboot	14
4.1.2 显示通路排查	14
4.2 ksc 常见问题排查	14
4.2.1 启动闪屏	14
4.2.1.1 uboot 启动后进入内核瞬间闪屏	14

# 1 前言

## 概述

- 本文档提供 DISP、HDMI、LCD、DE-Interlace、G2D、KSC 等显示基础模块 Linux 方案常规问题快速排查方案
- 本文档提供的快速排查方案模型如下：  
常规问题场景 + 基础硬件排查 + 基础软件配置排查 + 软件快速调试排查
- 本文档仅用于帮助开发人员快速排查和解决初级问题问题，如果按照文档中的指导仍无法解决问题，请联系全志原厂 FAE 进一步协助解决，并提供说明是否按照文档中的排查方案进行过必要的排查，提供排查过程的必要记录和数据

## 读者对象

本文档主要适用于以下工程师：

- 不具备显示技术基础的 RD 研发人员
- 不具备显示技术基础的 FAE 工程人员

## 作者信息

类别	作者
LCD	AW1221
G2D	AW0962
KSC	AW2257

## 2 LCD 调试说明

### 2.1 LCD 接口调试方法

以下大部分信息都需要 debugfs，如果没有挂载可以通过下面命令行挂载：`mount -t debugfs none /sys/kernel/debug`

#### 2.1.1 LCD 显示状态调试

```
cat /sys/class/disp/disp/attr/sys
```

**screen 0:**

```
screen 0:
de_rate 297000000 hz, ref_fps:60
mgr0: 1280x800 fmt[rgb] cs[0x204] range[full] eotf[0x4]
bits[8bits] err[0] force_sync[0] unblank direct_show[false]
  lcd output backlight( 50) fps:60.9 1280x 800
  err:0 skip:31 irq:1942 vsync:0 vsync_skip:0
  BUF enable ch[1] lyr[0] z[0] prem[N] a[global 255] fmt[ 8]
fb[1280, 800;1280, 800;1280, 800] crop[ 0, 0,1280, 800] frame[ 0,
0,1280, 800] addr[ 0, 0, 0] flags[0x 0] trd[0,0]
```

- **lcd output**

表示当前显示接口是 LCD 输出。如果不是，要么应用层要主动切换，要么你要在 `sys_config.fex` 设置好 `screenX_output_type` 属性。

- **1280x 800**

表示当前 LCD 的分辨率，需要和 `sys_config.fex` 中 `lcd0` 的设置一样，

- **ref\_fps:60**

表示当前 LCD 的显示帧率理论值，该值是根据 `sys_config.fex` 的 `lcd0` 填的时序算出来的**理论值**；

- **fps:60.9**

表示当前 LCD 的显示帧率实时统计值，正常来说应该是在 60(期望的 fps) 附近，如果差太多则不正常，重新检查屏时序，和在屏驱动的初始化序列是否有被调用到。

- irq:1942

表示 LCD 的显示中断的次数，正常来说是一秒 60（期望的 fps）次，重复 cat sys，如果无变化，则异常。

- BUF

表示图层信息，一行 BUF 表示一个图层，如果一个 BUF 都没有出现，那么将是黑屏，不过和屏驱动本身关系就不大了，应该查看应用层 & 框架层。

## 2.1.2 LCD 电源状态调试

对于 axp 方案，如果需要检查 LCD 的供电状态，可以通过以下命令检查 LCD 电源是否有 enable，同并结合实际万用表量取电压值进行比较，以帮助排查初级硬件问题。

```
cat /sys/class/regulator/dump
```

```
pmu1736_ldoio2: disabled 0 700000 supply_name:  
pmu1736_ldoio1: disabled 0 700000 supply_name:  
pmu1736_dc1sw: enabled 1 3300000 supply_name: vcc-lcd  
pmu1736_cpus: enabled 0 900000 supply_name:  
pmu1736_cldo4: disabled 0 700000 supply_name:  
pmu1736_cldo3: disabled 0 700000 supply_name:  
pmu1736_cldo2: enabled 1 3300000 supply_name: vcc-pf  
pmu1736_cldo1: disabled 0 700000 supply_name:  
pmu1736_bldo5: enabled 2 1800000 supply_name: vcc-cpvin vcc-pc  
pmu1736_bldo4: disabled 0 700000 supply_name:  
pmu1736_bldo3: disabled 0 700000 supply_name:  
pmu1736_bldo2: disabled 0 700000 supply_name:  
pmu1736_bldo1: disabled 0 700000 supply_name:  
pmu1736_aldo5: enabled 0 2500000 supply_name:  
pmu1736_aldo4: enabled 0 3300000 supply_name:  
pmu1736_aldo3: enabled 1 1800000 supply_name: avcc  
pmu1736_aldo2: enabled 0 1800000 supply_name:  
pmu1736_aldo1: disabled 0 700000 supply_name:  
pmu1736_rtc: enabled 0 1800000 supply_name:  
pmu1736_dcdc6: disabled 0 500000 supply_name:  
pmu1736_dcdc5: enabled 0 1480000 supply_name:  
pmu1736_dcdc4: enabled 1 900000 supply_name: vdd-sys  
pmu1736_dcdc3: enabled 0 900000 supply_name:  
pmu1736_dcdc2: enabled 0 1160000 supply_name:  
pmu1736_dcdc1: enabled 4 3300000 supply_name: vcc-emmc vcc-io vcc-io vcc-io
```

## 2.1.3 LCD PWM 状态调试

LCD 的 PWM 接口用于提供背光电源，调试方法如下

```
cat /sys/kernel/debug/pwm
```

打印如下信息：

```
platform/7020c00.s_pwm, 1 PWM device
pwm-0 ((null) ): period: 0 ns duty: 0 ns polarity: normal

platform/300a000.pwm, 2 PWM devices
pwm-0 (lcd ): requested enabled period: 20000 ns duty: 3984 ns polarity: normal
pwm-1 ((null) ): period: 0 ns duty: 0 ns polarity: normal
```

上面输出的信息表示：通道 0 的 pwm 被请求和使能了，使能者是” LCD” ，后面还有 pwm 的周期频率信息。

注意：方案配置中存在两种类型的 pwm，一种为是 CPUS(supper standby) 的 pwm 对应上面的 **platform/7020c00.s\_pwm**，这种 pwm 在软件上通常对应通道 8 的 pwm 上。一种是 CPUX(主控) 的 pwm，对应上面的 **platform/300a000.pwm**，软件上一般是 pwm0 到 pwm7。

## 2.1.4 LCD PIN 引脚状态调试

LCD 屏一般需要配置部分管脚的复用功能，一般来说如果是 RGB 或者 LVDS 接口需要配置对应的数据脚，时钟脚和同步信号脚。

此外，有些屏会用到一些管脚用于复位、电源使能、背光使能、甚至是用 gpio 模拟 SPI，模拟 I2C 等。

这个时候想知道自己设置的管脚有没有设置成功可以参考下面的打印。

```
cat /sys/kernel/debug/pinctrl/pio/pinmux-pins
```

打印以下信息：

```
pin 227 (PH3): twi1 (GPIO UNCLAIMED) function io_disabled group PH3
pin 228 (PH4): (MUX UNCLAIMED) (GPIO UNCLAIMED)
pin 229 (PH5): (MUX UNCLAIMED) pio:229
pin 230 (PH6): (MUX UNCLAIMED) pio:230
pin 231 (PH7): (MUX UNCLAIMED) pio:231
pin 106 (PD10): lcd0 (GPIO UNCLAIMED) function lvds1 group PD10
```

上面的信息表示：

如果一行里面同时出现 (MUX UNCLAIMED) (GPIO UNCLAIMED)，说明该管脚处于初始状态没有声明也没有复用为某个功能。

如果 (MUX UNCLAIMED) 被换成其它字符，比如 twi1 或者 lvds 说明该管脚被声明为改字符串的功能。如果 (GPIO UNCLAIMED) 被换成 pio:xxx，说明该管脚被复用为普通 GPIO，而且处于使能状态。

PH5，PH6 这些 IO 被申请为普通 GPIO 功能，而 PD10 被声明而且复用为 lvds 管脚功能。

PH3 虽然被声明为 twi1，但是后面 function io\_disabled，表示当前为 io disable 状态。

## 2.1.5 LCD CLK 时钟配置调试

抓取与显示 Tcon 相关的 CLK 时钟信息 `cat /sys/kernel/debug/clk/clk_summary | grep tcon`

抓取与显示 pll\_video 相关的 CLK 时钟信息 `cat /sys/kernel/debug/clk/clk_summary | grep pll_video`

抓取与显示 mipi 抓取与显示 `cat /sys/kernel/debug/clk/clk_summary | grep mipi`

## 2.1.6 LCD 接口 colorbar 调试

显示是一整条链路，中间任何一个环节出错，最终的表现都是显示异常，图像显示异常几个可能原因，一、图像本身异常。二、图像经过 DE (Display Engine) 后异常。三、图像经过接口模块后异常。这是我们关注的点。

有一个简单的方法可以初步判断，接口模块 (tcon 和 dsi 等) 可以自己输出内置的一些 patten，比如说彩条，灰阶图，棋盘图等。当接口输出这些内置 patten 的时候，如果这时候显示就异常，这说明了：

1. LCD 的驱动或者配置有问题
2. LCD 屏由于外部环境导致显示异常

显示自带 patten 的方式：在 linux-4.9 上，disp 的 sysfs 中有一个 attr 可以直接操作显示：`echo X > /sys/class/disp/disp/attr/colorbar`

上面的操作是显示 colorbar，其中的 X 可以是 0 到 8，0 到 7 对应的含义如下图所示：

LCD\_SRC\_SEL

000: DE

001: Color Check

010: Grayscale Check

011: Black by White Check

100: Test Data all 0

101: Test Data all 1

110: Reversed

111: Gridding Check

图 2-1: colorbar 参数说明

8 表示让 DE 绘制 colorbar。如果有多个显示，想让第二个显示显示 colorbar 的话，那么先 echo 1 > /sys/class/disp/disp/attr/disp 然后再执行上面操作。

如果没有这个 attr 的话，可以直接操作寄存器，也就是操作 tcon 寄存器的 040 偏移的最低 3 位。在 linux 下，cd /sys/class/sunxi\_dump 然后 echo 0x06511040 > dump;cat dump 这样会打印当前 tcon 的 040 偏移寄存器的值，然后在上面值的基础上修改最低 3 位为上图的值即可，修改方式，示例：echo 0x06511040 0x800001f1 > write 注意 tcon 的基地址不一定是 0x06511000，不同平台不一样，请参考 SOC 文档获取 tcon 的基地址。

显示出现异常的时候，有可能是下面三个原因：

1. SOC 端屏接口模块 +LCD 屏出现了问题。
2. 图像经过 SOC 端图像合成模块 (DE) 处理后出现了问题。
3. 图像源本身就有问题。

本节介绍，确认图像源本身没问题的前提下，如何进一步确认是否经过 DE 处理之后图像是否有问题。

语法：echo 屏幕索引 > /sys/class/disp/disp/attr/disp echo 路径/bmp 文件名 > /sys/class/disp/disp/attr/capture\_dump 第一个 echo 的作用是确定捕捉哪个显示的，“屏幕索引”可取值是 0 或者 1 第二个 echo 作用是生成截屏 bmp 文件，确保“路径”是可写的，有剩余空间的。

比如：`echo 0 > /sys/class/disp/disp/attr/disp echo /data/xx.bmp > /sys/class/disp/disp/attr/capture_dump`

这样就会在/data 目录下生成 screen 0 的 bmp 截图，文件名是 xx.bmp。

注意：这个功能只有 linux-4.9 以及后续的内核才支持这个功能。

## 2.2 LCD 常见问题

### 2.2.1 黑屏（无背光）

背光一般是由 pwm 来控制的，没有背光的话有多种原因，请按下面步骤依次检查：

1. 屏驱动没有加载成功。屏驱动编写有问题，没有加载，自然不会执行屏背光开启的操作，你可以通过查看显示信息一小节，确定屏驱动是否加载成功了。
2. PWM 通道没有使能。检查原理图，确定所用 pwm 通道编号，查看 pwm 信息一节确认对应 pwm 有没有使能。软件确定 sys\_config.fex 中的 [lcd0] 中 lcd\_pwm\_ 开头的属性配置正确。
3. 背光使能脚没有使能。不少设计中都用一根 gpio 来控制背光的使能，当这根脚处于无效状态时，背光自然无法亮，根据屏手册得知背光使能脚有效状态（高低电平，以及电压值），然后用万用表确认下。软件上确认 sys\_config.fex 中的 [lcd0] 的 lcd\_bl\_en 的定义是准确的。
4. 背光值设置太低。不少屏，但 pwm 的占空比为接近 0% 的时候就会完全黑屏。
5. 原理图或者 PCB 有错导致背光电压不够。屏手册里面会规定背光所需要的电压范围，直接用万用表确认下是否符合屏的要求。

### 2.2.2 黑屏（有背光）

黑屏但是有背光，可能有多种原因导致，请依次按以下步骤检查：

1. 没送图层。如果应用没有送任何图层那么表现的现象就是黑屏，通过查看显示信息一小节可以确定有没有送图层。如果确定没有图层，可以通过查看接口自带 colorbar，确认屏能否正常显示。
2. SOC 端的显示接口模块没有供电。SOC 端模块没有供电自然无法传输视频信号到屏上。一般 SOC 端模块供电的 axp 名字叫做 vcc-lcd, vcc-dsi, vcc33-lcd, vcc18-dsi 等。
3. 复位脚没有复位。如果有复位脚，请确保硬件连接正确，确保复位脚的复位操作有放到屏驱动中。
4. sys\_config.fex 中 [lcd0] 有严重错误。第一个是 lcd 的 timing 太离谱，请严格按照屏手册中的提示来写！手册没写就找屏厂，这个时序只有屏厂清楚。第二个就是，接口类型搞错，比如接的 DSI 屏，配置却写成 LVDS 的。
5. 屏的初始化命令不对。包括各个步骤先后顺序，延时等，这个时候请找屏厂确认初始化命令。

## 2.2.3 闪屏

[问题表现]

1. 整个屏幕闪，忽明忽暗的感觉，一般为背光或智能背光的问题。
2. 屏的部分区域闪，有的时候在开屏后闪，一段时间后消失

[分析/试验]

1. 检查智能背光是否开着，如果开着，关掉是否还会出现，澄清智能背光的问题
2. 如果不是智能背光问题，用万用表测量下 vled+/- 的电压是否稳定，如果稳定说明不是背光原因；如果不稳定，跳下一步
3. 查一下 pwmpin 的电压，看是否稳定，如果稳定，则要查一下背光电路或背光 IC 是否存在问题，或者是否有其他电路会影响到 vled+/-

如果不稳定，先检查下软件原因，pwm 的配置是否一直变化：如果有变化则为软件原因；如果没有变化，再查硬件原因，检查 pwm 受到哪些因素的干扰。

4. 如果是智能背光原因，则如果检查智能背光调节问题。智能背光会对背光及显示内容进行调节，可以分开检查，对背光的调节是否正确合理/对显示内容调节是否正确合理。
5. 针对表现 2，很大的可能是屏的上电顺序不正确或者初始化未正确引起。

## 2.2.4 白屏

不少屏初始化失败的，但是电源有上（至少背光有上），那么呈现的现象是白屏。

1. 检查所有屏相关电源。通过查看电源信息一小节确认需要开 axp 电源有开启。
2. 检测复位脚电压。如果有的话。
3. 检查初始化代码以及调整 LCD 的 timing (lcd\_dclk\_freq, lcd\_ht, lcd\_vt, lcd\_hbp, lcd\_vbp, lcd\_hspw, 和 lcd\_vspw)，咨询屏厂。

## 2.2.5 雪花屏

一部分屏再初始化成功之后，由于 SOC 端没有发送数据，会呈现雪花屏的状态，一般是那种需要 te 脚 (tear effect) 同步的屏。

这种情况需要，

1. 检查 te 脚有没有连到 SOC 端的 lcd\_vsync 脚
2. lcd\_vsync 所对应的脚有没有被设置成 vsync 功能
3. 检查 te 脚的电压是否够大，是否足以触发中断
4. 雪花屏的另外一个原因，就是图像数据被串改，这个时候和屏本身就没关系了。可以通过截屏一小节来确认图像本身是否正常，通过查看接口自带 colorbar 一小节来确认接口信号是否正常。

## 2.2.6 图像抖动

图像抖动可能是屏的问题也可能是图像数据本身的问题。

1. 这时候可以通过查看接口自带 colorbar，确认图像本身是否正常
2. 通过查看接口自带 colorbar 一小节来确认接口信号是否正常。
3. 如果是屏的问题，那么可能原因就是时钟管脚以及其它数据信号脚的驱动能力不足。这个时候需要修改 sys\_config.fex 中 [lcd0] 的管脚部分的设置，详情查看《sunxi display2 LCD 调试指南》。

## 2.2.7 条形波纹

有些 LCD 屏的像素格式是 18bit 色深 (RGB666) 或 16bit 色深 (RGB565)，建议打开 FRM 功能，通过 dither 的方式弥补色深，使显示达到 24bit 色深 (RGB888) 的效果。如下图所示，上图是色深为 RGB66 的 LCD 屏显示，下图是打开 dither 后的显示，打开 dither 后色彩渐变的地方过度平滑。

设置 [lcd0] 的 lcd\_frm 属性可以改善这种现象。

## 2.2.8 背光太亮或者太暗

背光是模拟信号，有些电路无法保证线性，所以出现某些背光值过亮或者过暗的情况。

设置 [lcd0] 的 Lcd\_bl\_n\_percent 属性可以改善这种现象。

## 3 G2D 调试说明

### 3.1 设备节点

dev/g2d

### 3.2 传递参数信息

1. echo 1 > /sys/class/g2d/g2d/attr/debug 打开调试信息
2. echo 0 > /sys/class/g2d/g2d/attr/debug 关闭调试信息

如果打印太多，可以先调高打印等级 echo 0 > /proc/sys/kernel/printk, 然后 dmesg 查看信息

```
[G2D-g2d_bsp_bitblt] line:2036: BITBLT_flag: 0x100 -----设置旋转角度
0x0 不旋转
0x400 0度
0x100 90度
0x200 180度
0x300 270度
[G2D-g2d_bsp_bitblt] line:2163: ROT input info:
[G2D-g2d_bsp_bitblt] line:2164: Src_fd: 21 -----源图片buf fd
[G2D-g2d_bsp_bitblt] line:2165: Format: 0x6 -----输入format
[G2D-g2d_bsp_bitblt] line:2166: Flag: 0x100
[G2D-g2d_bsp_bitblt] line:2167: inClipRectX: 0 -----crop x
[G2D-g2d_bsp_bitblt] line:2168: inClipRectY: 0 -----crop y
[G2D-g2d_bsp_bitblt] line:2169: inClipRectW: 1280 -----crop w
[G2D-g2d_bsp_bitblt] line:2170: inClipRectH: 800 -----crop h
[G2D-g2d_bsp_bitblt] line:2179: ROT_IFMT: 0x6
[G2D-g2d_bsp_bitblt] line:2180: ROT_ISIZE: 0x31f04ff -----w h写入寄存器的值
[G2D-g2d_bsp_bitblt] line:2182: SRC_align: 64, 0, 0 -----源对齐方式
[G2D-g2d_bsp_bitblt] line:2184: DST_align: 64, 0, 0 -----输出图片对齐方式
[G2D-g2d_bsp_bitblt] line:2195: ROT_InPITCH: 5120, 0, 0 -----输入pitch
[G2D-g2d_bsp_bitblt] line:2237: ROT output info: -----
[G2D-g2d_bsp_bitblt] line:2238: Dst_fd: 40 -----输出图片buf fd
[G2D-g2d_bsp_bitblt] line:2239: Format: 0x6 -----输出format
[G2D-g2d_bsp_bitblt] line:2249: ROT_OutPITCH: 3200, 0, 0 -----输出pitch
[G2D-g2d_bsp_bitblt] line:2250: outClipRectX: 0 -----crop x
[G2D-g2d_bsp_bitblt] line:2251: outClipRectY: 0 -----crop y
[G2D-g2d_bsp_bitblt] line:2252: outClipRectW: 800 -----crop w
[G2D-g2d_bsp_bitblt] line:2253: outClipRectH: 1280 -----crop h
[G2D-g2d_bsp_bitblt] line:2270: DST_ADDR0: 0x2800000 -----输出映射后的地址
[G2D-g2d_bsp_bitblt] line:2271: DST_ADDR1: 0x2904000 -----同上
[G2D-g2d_bsp_bitblt] line:2272: DST_ADDR2: 0x29fe000 -----同上
[G2D-g2d_bsp_bitblt] line:2278: init_module: 0x80000011 -----写到寄存器控制器值
[G2D-g2d_bsp_bitblt] line:2036: BITBLT_flag: 0x100
```

## 3.3 G2D 问题指南

### 3.3.1 invalid address

- 问题现象：

串口狂打印” DE invalid address: 0x0, data:0x0, id:0x1”

- 排查步骤：

一般这种情况是访问了一片没有分配到内存的区域，导致越界报错

要排查出现问题的地址是否是 g2d 访问的那片区域，如果是，那就要看是否内存申请小了，或者对齐没做好；如果不是，那就找 iommu 同事联调。

### 3.3.2 dmabuf get fd failed

- 问题现象：

串口打印 get fd failed。

- 排查方法：

显示 buffer 句柄 fd 提前释放或不存在，非驱动问题，排查显示应用 buffer 调用流程。

### 3.3.3 dma\_map\_sg failed

- 问题现象：

串口打印 get fd failed。

- 排查方法：

ION 申请 buffer 用错 flag，要检查

Cma 内存：ION\_HEAP\_TYPE\_DMA

Iommu：ION\_HEAP\_TYPE\_SYSTEM

### 3.3.4 旋转画面撕裂

一般出现这个原因，多半是对齐没做好，尤其是 YV12 这种，旋转之后 Y、U、V 三个分量也要对齐

### 3.3.5 旋转系统挂死

一般出现这种情况，可能是上电时序出了问题，这个时候检查几个总线 gating 是否开启

对应 spec 寄存器 G2D\_SCLK\_GATE、G2D\_HCLK\_GATE、G2D\_AHB\_RESET。



## 4 KSC 调试说明

### 4.1 ksc 调试手段

#### 4.1.1 ksc 状态信息打印

##### 4.1.1.1 内核

```
cat /sys/class/ksc/ksc/attr/info
```

```
root@TinaLinux:/# cat /sys/class/ksc/ksc/attr/info
Enabled:1 irq:0 irq_no:240 be err:0 fe err:0 svp2ksc_dmt:0 de2ksc_data_volume_err:0
===== ksc reg value =====
0xffffffffc800200000: 0x14000001 0x14001001 0x00010000 0x00011010
0xffffffffc800200010: 0x00000000 0x00101100 0x00000111 0x02800280
0xffffffffc800200020: 0x02800280 0x02800140 0x000000b0 0x00000000
0xffffffffc800200030: 0x050002d0 0x050002d0 0x0fff0fff 0x04ff0000
0xffffffffc800200040: 0x02cf0000 0x00000000 0x44908000 0x449e9000
0xffffffffc800200050: 0x00000000 0x44aca000 0x44bab000 0x00000000
0xffffffffc800200060: 0x00000000 0x00000000 0x00000000 0x44708000
0xffffffffc800200070: 0x00000000 0x00000000 0x00000000 0x44718000
0xffffffffc800200080: 0x00000000 0x44719000 0x0119000c 0x00000000
0xffffffffc800200090: 0x00180046 0x017a0147 0x00000000 0x00000000
0xffffffffc8002000a0: 0x00000000 0x00000000 0x00808000 0x40302010
0xffffffffc8002000b0: 0x00001210 0x01001511 0x0040005c 0x40041810
0xffffffffc8002000c0: 0x00200020 0x00000000 0x00010010 0x00010010
0xffffffffc8002000d0: 0x0000001f 0x10800400 0x02170218 0x00000218
0xffffffffc8002000e0: 0x01a501f6 0x00020000 0x00000000 0x00000000
===== ksc clk info =====
clk0 rate:30000000
Reset0 state:0
clk1 rate:24000000
Reset1 state:0
===== ksc bandwidth control configuration info =====
bandwidth control Enabled: 1
bandwidth control num: 8
```

图 4-1: KSC 状态信息打印

- Enabled: ksc 开启/关闭状态
- irq: 中断触发次数
- be err: ksc be 模块 ERROR 中断上报次数
- fe err: ksc fe 模块 ERROR 中断上报次数
- svp2ksc\_dmt: ksc100 SVP 到 ksc 数据传输缺数错误触发中断次数

- de2ksc\_data\_volume\_err: ksc110 de 到 ksc 数据传输缺数错误触发中断次数
- ksc reg value : ksc 寄存器打印, 基地址为 0x05300000
- ksc clk info : ksc 时钟信息
- ksc bandwidth control configuration info: ksc 带宽控制配置信息, 包括是否开启带宽控制, 带宽控制数

#### 4.1.1.2 uboot

为实现平滑显示, 在 uboot 阶段已初始化并使能 ksc, uboot command line 可输入' ksc '命令打印 KSC 状态信息, 状态信息格式与内核一致。

### 4.1.2 显示通路排查

```
cd /sys/class/disp/disp/attr
echo 1 > colorbar
echo 8 > colorbar
```

ksc 使能情况下, 在显示数据通路中处于 DE 与 TCON 中间。通过 colorbar 可确定是否是 KSC 错误导致显示异常。

## 4.2 ksc 常见问题排查

### 4.2.1 启动闪屏

#### 4.2.1.1 uboot 启动后进入内核瞬间闪屏

解决方法: 开启 ksc 带宽控制, 在 board.dts 和 uboot-board.dts 的加入 ksc 节点, bw\_ctrl\_en = <1> 开启 ksc 带宽控制。

```
&ksc {
    ...
    bw_ctrl_en = <1>;
};
```

若开启带宽控制后, 出现**绿屏/花屏/画面停顿**现象, 打印 ksc 状态信息, 观察到 de2ksc\_data\_volume\_err 数不断增加, 说明 ksc 带宽控制对 de 出现了过度的反压作用, 导致了 DE 到 KSC 通路出现数据缺数, 此时应加入 bw\_ctrl\_num 属性配置 ksc 带宽控制数:

```
&ksc {
    ...
    bw_ctrl_en = <1>;
```

```
bw_ctrl_num = <8>;  
};
```

bw\_ctrl\_num 的临界值计算公式如下：

$\text{bw\_ctrl\_num (临界值)} = \text{src\_w} * \text{src\_h} * \text{fps} * 32 / \text{freq}$  (向上取整)

src\_w、src\_h 分别表示 DE 输出图像宽高，fps 帧率，freq 为 ksc 工作时钟频率 300M。

bw\_ctrl\_num 范围为 0~15，必须大于临界值，如果仍出现绿屏/花屏现象，加大 bw\_ctrl\_num，直到正常显示。






## 著作权声明

版权所有 ©2024 珠海全志科技股份有限公司。保留一切权利。

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

、、**全志科技**、（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。