

密级状态：绝密() 秘密() 内部() 公开()

RK3368 VOP 模块总结

(图形与显示系统)

文件状态： [] 正在修改 [<input checked="" type="checkbox"/>] 正式发布	当前版本：	V0.1
	作 者：	黄家钗
	完成日期：	2015-06-14
	审 核：	
	完成日期：	

福州瑞芯微电子有限公司

Fuzhou Rockchips Semiconductor Co., Ltd

(版本所有,翻版必究)

版本历史

版本号	作者	修改日期	修改说明	备注
V0.1	黄家钗	20150614	初稿	

目 录

一、新特性介绍.....	2
二、问题及解决方法.....	2
1、IOMMU PAGE FAULT.....	2
2、iommu 使能.....	4
3、CABC 问题.....	5
4、Interlace output.....	6
5、HDMI 切换显示异常.....	7
6、缩小锯齿问题.....	7
7、average 算法产生绿线问题.....	8
三、遗留未解决的问题.....	9
四、后期优化及建议.....	10

一、新特性介绍

RK VOP ->VOP FULL->RK3288/RK3368

-> VOP LITE->RK3036/RK312X/SOFIA 3GR

RK3368 VOP 是基于 RK3288 VOP 基础上修正之前存在的问题同时加入一些新功能，feature 对比如下：

Feature	RK3288	RK3368
Vop number	2 个	1 个
10bit video input	支持	不支持
IFBDC	不支持	支持(2048x1536UI)
YUV overlay/output	不支持	支持
CABC	支持, 有问题	支持, 改善, 还存在问题
Mirror	Post x/y mirror	Win/post x/y mirror
Max Output solution	Big:3840x2160 Little:2560x1600	Big:4096x2160 ->MID:2048x1536 ->BOX:4096x2160
Interface output	不支持	加入 1080i 支持, 有问题
10bit output	支持(for EDP/HDMI/LVDS)	不支持

二、问题及解决方法

1、IOMMU PAGE FAULT

1).问题现象：死机/绿屏

2).异常 log:

```
rk_iommu ff930300.vop_mmu: 1.rawstat = 0x00000001,status = 0x00000001,reg_status = 0x0000018b
rk_iommu ff930300.vop_mmu: Page fault detected at 0x00000000119b0200 from bus id 0 of type read on vop
iova = 0x00000000119b0200: dte_index: 0x046 pte_index: 0x1b0 page_offset: 0x200
mmu_dte_addr: 0x0000000079ca1000 dte@0x0000000079ca1118: 0x7947a001 valid: 1 pte@0x000000007947a6c0: 0x000000 valid: 0
IOMMU(vop) map/unmap history list (total 20):
(00) map [12500000-130fffff]:00c00000 at 26.887858s
backtrace:
[<fffffc0006e3ab0>] rockchip_iovmm_map+0x364/0x45c
[<fffffc0006ce3b0>] ion_system_map_iommu+0x28/0xac
[<fffffc0006ca118>] ion_map_iommu+0x1dc/0x43c
[<fffffc0003311f4>] rk_fb_set_win_config+0x33c/0xdc4
[<fffffc0003323c4>] rk_fb_ioctl+0x748/0x828
[<fffffc00031d234>] fb_compat_ioctl+0x52c/0x544
[<fffffc0001bd91c>] compat_sys_ioctl+0x104/0x1e8
[<fffffc000084248>] cpu_switch_to+0x48/0x4c
[<ffffffffffffffff>] 0xffffffffffffffff
(01) unmap [11900000-124fffff]:00c00000 at 26.891678s
```

3).处理过程及方法:

a.加入 iommu debug 配置

echo 1 > /sys/module/rk_fb/parameters/rk_fb_iommu_debug 或者直接
将 driver/video/rockchip/rk_fb.c 中的 rk_fb_iommu_debug 设置为 1;

打开节点后内核会保存前面三帧的配置，当发生 iommu pagefault 后通过 iommu 的中

断回掉函数将 vop 当前的配置以及前面几帧的配置全部 dump 出来:

```
last config:
win[0]area[0],state=1,addr=0x13c94000, last_map_cnt=2
win[1]area[0],state=1,addr=0x13100000, last_map_cnt=2
win[2]area[0],state=1,addr=0x12500000, last_map_cnt=2
win[2]area[1],state=1,addr=0x12539000, last_map_cnt=2
win[2]area[2],state=0,addr=0x00000000, last_map_cnt=0
win[2]area[3],state=0,addr=0x00000000, last_map_cnt=0
win[3]area[0],state=0,addr=0x00000000, last_map_cnt=1
win[3]area[1],state=0,addr=0x00000000, last_map_cnt=0
win[3]area[2],state=0,addr=0x00000000, last_map_cnt=0
win[3]area[3],state=0,addr=0x00000000, last_map_cnt=0
win[4]area[0],state=0,addr=0x00000000, last_map_cnt=0
last freed buffer:
0:0x13c94000, unmap cnt=1
1:0x13100000, unmap cnt=1
2:0x12500000, unmap cnt=1
3:0x12539000, unmap cnt=1
4:0x11900000, unmap cnt=0
vop now state:
win[0]area[0],state=1,addr=0x13c94000
win[1]area[0],state=1,addr=0x13100000
win[2]area[0],state=1,addr=0x12500000
win[2]area[1],state=1,addr=0x12539000
win[2]area[2],state=0,addr=0x00000000
win[2]area[3],state=0,addr=0x00000000
win[3]area[0],state=1,addr=0x11900000
win[3]area[1],state=0,addr=0x00000000
win[3]area[2],state=0,addr=0x00000000
win[3]area[3],state=0,addr=0x00000000
win[4]area[0],state=0,addr=0x5ebfce68
now config:
win[0]area[0],state=1,addr=0x13c94000,after map cnt=2
win[1]area[0],state=1,addr=0x13100000,after map cnt=2
win[2]area[0],state=1,addr=0x12500000,after map cnt=2
win[2]area[1],state=1,addr=0x12539000,after map cnt=2
win[2]area[2],state=0,addr=0x00000000,after map cnt=0
win[2]area[3],state=0,addr=0x00000000,after map cnt=0
win[3]area[0],state=0,addr=0x00000000,after map cnt=0
win[3]area[1],state=0,addr=0x00000000,after map cnt=0
```

b.加入应用配置 debug 节点:

```
echo 1 > /sys/module/rk_fb/parameters/rk_fb_debug_lvl
```

```
[0: surfaceflinger: 197] -----frame start-----
[0: surfaceflinger: 197] user config:
[0: surfaceflinger: 197] win[0]:z_order=0,galhpa_v=0
[0: surfaceflinger: 197]   area[0]:fmt=2,ion_fd=91,phy_add=0x0,xoff=0,yoff=0
[0: surfaceflinger: 197]     xpos=0,ypos=0,xsize=1536,ysize=2048
[0: surfaceflinger: 197]     xact=1536,yact=2048,xvir=2048,yvir=2048
[0: surfaceflinger: 197] win[1]:z_order=1,galhpa_v=255
[0: surfaceflinger: 197]   area[0]:fmt=1,ion_fd=55,phy_add=0x0,xoff=0,yoff=0
[0: surfaceflinger: 197]     xpos=0,ypos=0,xsize=1536,ysize=2048
[0: surfaceflinger: 197]     xact=1536,yact=2048,xvir=1536,yvir=2048
[0: surfaceflinger: 197] win[2]:z_order=2,galhpa_v=255
[0: surfaceflinger: 197]   area[0]:fmt=1,ion_fd=44,phy_add=0x0,xoff=0,yoff=0
[0: surfaceflinger: 197]     xpos=0,ypos=0,xsize=1536,ysize=38
[0: surfaceflinger: 197]     xact=1536,yact=38,xvir=1536,yvir=38
[0: surfaceflinger: 197]   area[1]:fmt=1,ion_fd=47,phy_add=0x0,xoff=0,yoff=0
[0: surfaceflinger: 197]     xpos=0,ypos=1976,xsize=1536,ysize=72
[0: surfaceflinger: 197]     xact=1536,yact=72,xvir=1536,yvir=72
[0: surfaceflinger: 197] regs data:
[0: surfaceflinger: 197] win_num=3,buf_num=4
[0: surfaceflinger: 197] win[0]:z_order=0,area_num=1,area_buf_num=1
[0: surfaceflinger: 197]   area[0]:fmt=8,ion=ffffffc05e8da340,smem_star=0x11902000,cbr_star=0x0
[0: surfaceflinger: 197]     yoff=0x0,coff=0x0,area_data->buff_len=1000000
[0: surfaceflinger: 197]     xpos=0,ypos=0,xsize=1536,ysize=2048
[0: surfaceflinger: 197]     xact=1536,yact=2048,xvir=2048,yvir=2048
[0: surfaceflinger: 197] win[1]:z_order=1,area_num=1,area_buf_num=1
[0: surfaceflinger: 197]   area[0]:fmt=9,ion=ffffffc05e8da400,smem_star=0x135af000,cbr_star=0x0
[0: surfaceflinger: 197]     yoff=0x0,coff=0x0,area_data->buff_len=c00000
[0: surfaceflinger: 197]     xpos=0,ypos=0,xsize=1536,ysize=2048
[0: surfaceflinger: 197]     xact=1536,yact=2048,xvir=1536,yvir=2048
[0: surfaceflinger: 197] win[2]:z_order=2,area_num=2,area_buf_num=2
[0: surfaceflinger: 197]   area[0]:fmt=9,ion=ffffffc05e8da440,smem_star=0x13506000,cbr_star=0x0
[0: surfaceflinger: 197]     yoff=0x0,coff=0x0,area_data->buff_len=39000
[0: surfaceflinger: 197]     xpos=0,ypos=0,xsize=1536,ysize=38
[0: surfaceflinger: 197]     xact=1536,yact=38,xvir=1536,yvir=38
[0: surfaceflinger: 197]   area[1]:fmt=9,ion=ffffffc05e8da740,smem_star=0x13541000,cbr_star=0x0
[0: surfaceflinger: 197]     yoff=0x0,coff=0x0,area_data->buff_len=6c000
[0: surfaceflinger: 197]     xpos=0,ypos=1976,xsize=1536,ysize=72
[0: surfaceflinger: 197]     xact=1536,yact=72,xvir=1536,yvir=72
[0: surfaceflinger: 197] -----frame end-----
```

4).原因

- A. 图层关闭没确认;
- B. 格式配置不对, 导致访问越界;
- C. 图层/区域状态不对;

```
--- a/drivers/video/rockchip/rk_fb.c
+++ b/drivers/video/rockchip/rk_fb.c
@@ -1841,6 +1841,8 @@ static void rk_fb_update_reg(struct rk_lcd_driver *dev_drv,
     } else {
         win->z_order = -1;
         win->state = 0;
         for (j = 0; j < 4; j++)
             win->area[j].state = 0;
     }
 #if defined(CONFIG_ROCKCHIP_IOMMU)
     if (dev_drv->iommu_enabled) {
         for (j = 0; j < 4; j++) {
```

2、iommu 使能

- 1).问题现象: uboot logo 切换到 kernel logo 时闪屏
- 2).原因: uboot logo 是物理地址, 内核起来后需要切到虚拟地址, 而 RK3368 的 mmu en 是实时生效, 图层配置是帧生效, 需要保证两个同步更改;
- 3).处理过程及方法:
 - a.保证图层配置不在 fp 阶段;
 - b.保证 mmu 是能在 blank 开始的时候打开;
 - c.保证不会中断抢占, 关闭系统中断;
 - d.保证 mmu 使能接口调用的时间小于 blank 时间, 打印导致延迟;

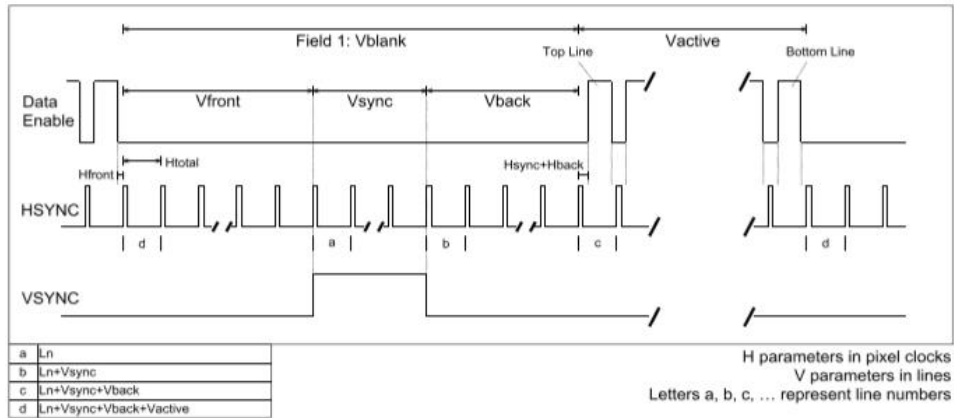
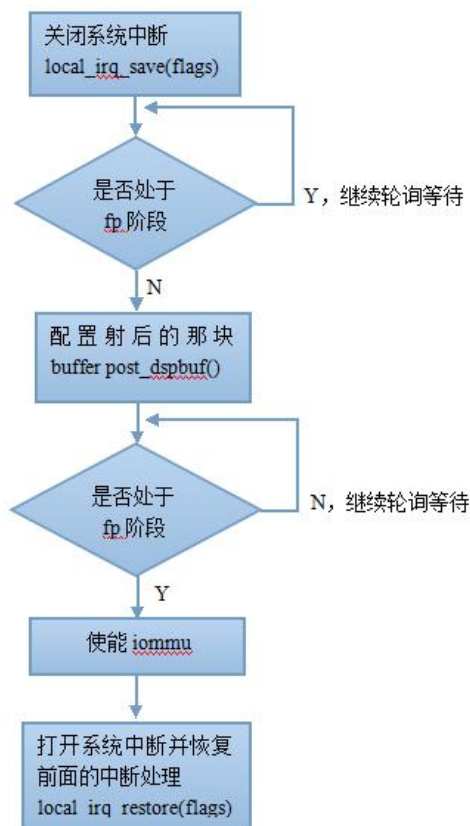


Figure 2. General Progressive Video Format Timing (Positive Sync)



3、CABC 问题

问题一：开 CABC 后抖屏

原因：PWM 和 RGB 调整受 global limit 限制不同步导致

处理方法：不使用 vop pwm，在帧中断处理函数里读取中间变量控制外部 pwm 实现背光调整，有所改善，但对比 MTK 样机的效果还是会差点；

问题二：EDP 屏开关 CABC 黑屏

原因：VOP 和 EDP 间同步问题

处理方法：系统加载阶段就打开 CABC，通过设置 CABC 调节力度实现 CABC 的开关；

问题三：过曝问题：

原因：CABC 功能带来的质量损失；

处理方法：后续优化算法；

功耗数据：

on\off	cabc = 0		cabc = 1	
brightness	占空比	屏电流	占空比	屏电流
200	21.60%	627	60.80%	374
150	41.60%	494	71.20%	317
100	60.80%	372	80.80%	263
50	80.80%	262	90.40%	209
25	90.40%	208	95.20%	183
5	98.40%	167	99.20%	162
1	100%	159	100%	158

4、Interlace output

问题一：缩放后显示效果不好

原因：目前的缩放算法没考虑 interlace 模式；

处理方法：图层不做缩放，保证是 src:dst = 2:1,对于 box 的 over scan 由 gpu 处理；

问题二：显示效果不好

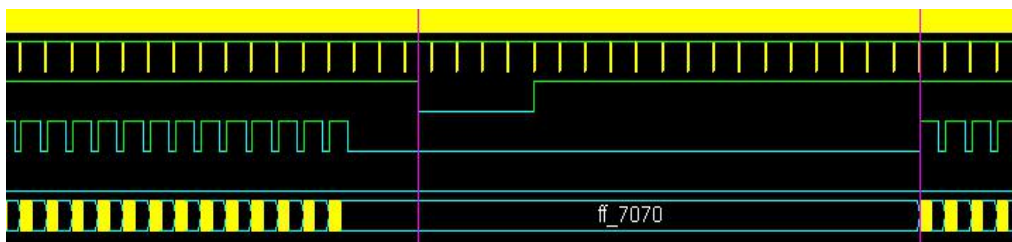
原因：场中断可以使寄存器生效，可能出现奇偶场刷不同 buffer 的奇偶行数据；

处理方法：软件保持在刷偶场数据的时候更新寄存器配置，后续芯片区别场中断和帧中断

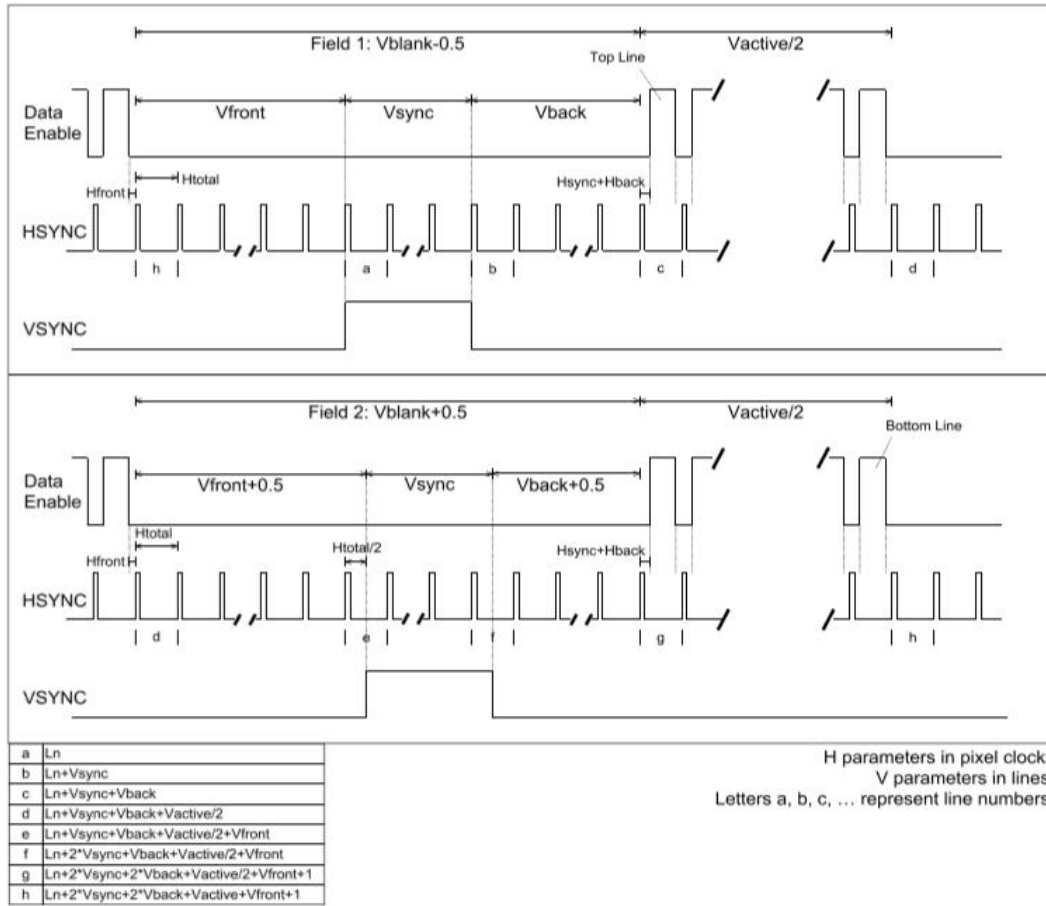
问题三：HDMI 1080i CTS 测试无法通过

原因：奇偶场时序错误

RK3368 时序：



标准时序：



5、HDMI/CVBS 切换显示异常

原因：a.RK3368 单个 vop，应用接收主辅屏/分辨率切换有延迟，给出错误的配置；

b.interlace 模式下如果 src 的 act_h 为奇数出现访问越界；

解决方法：

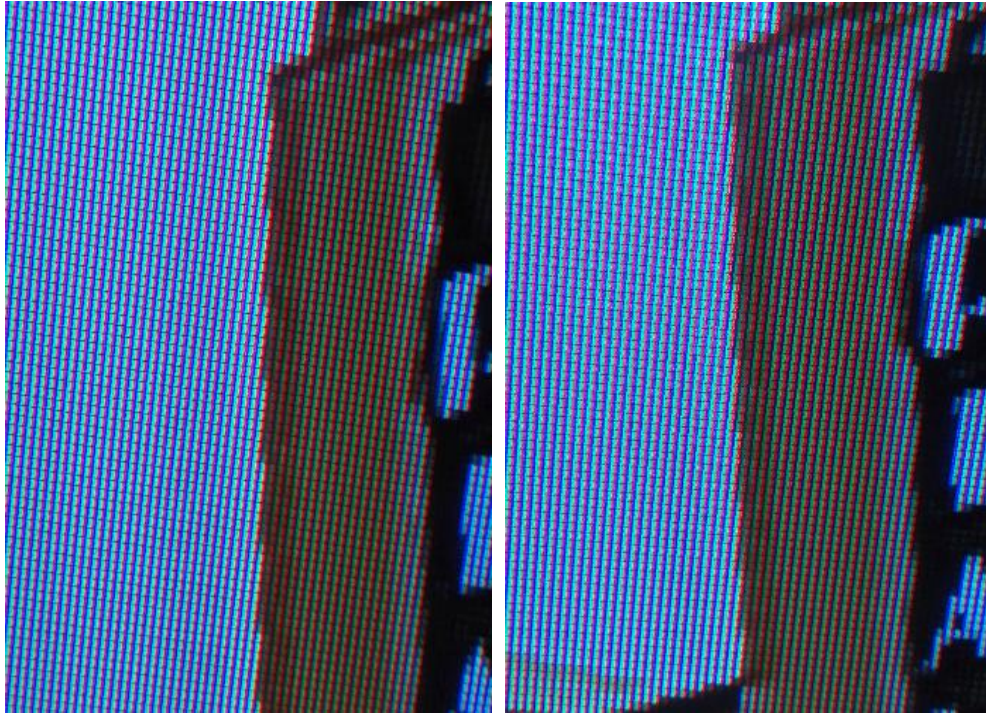
a.rk fb 做一层过滤，发现异常的配置不往下送；

b.在切屏过程中截掉应用的配置；

c.src act_h 驱动做 2pixel 对齐处理；

6、缩小锯齿问题

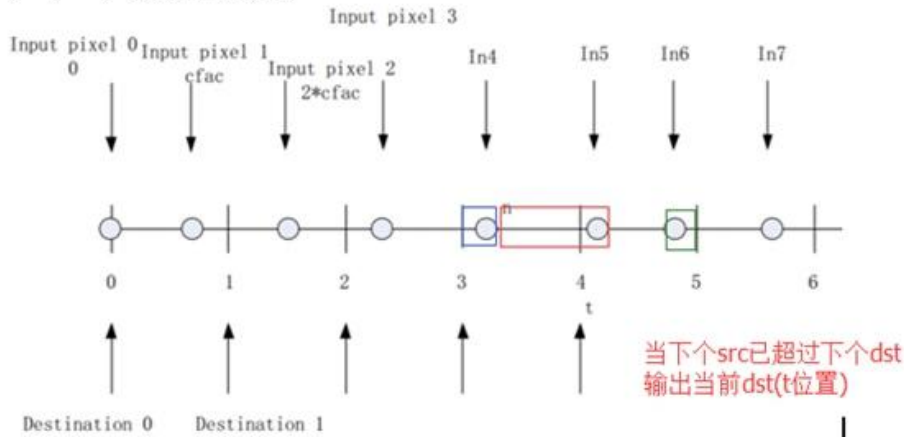
在播放 4k 视频 HDMI 1080p 输出的时候，锯齿现象比较明显，RK3368 的效果比海思 3798M 的较差，配置 offset 后有所改善，但是全局看整个画面的时候锯齿还是比较明显，在画面更新的时候会有抖动，下面左图 offset=0，右图 offset=0.5：



7、average 算法产生绿线问题

YUV420 数据从 3840x2160 缩小到 1918x1079 采用 average 算法产生绿线问题，算法部何平征分析如下：

确定是average算法实现时,定点化精度8位引入的bug,在之前几个版本中(VPU/VOP/RGA)可能都存在,需要check。



1920缩小到1918, scaleFactor=65484; scaleFactorSR8=255, src x=1255 开始, 16位精度累积的坐标xAcc在整数像素右侧, 而8位精度累积的坐标xAccSR8 = 前次 xAcc>>8 + scaleFactorSR8, 却可能在整像素左侧。

下xAcc余	下xAccSR8余	factorR(sfSR8-下xAccSR8余)	后factorL(下xAcc余SR8)
276	1		0
224	0	255 (得到dst=4, 还正确)	0
172	255 error	0 (得到dst=5的结果0绿线)	0

8位精度累积的坐标计算方法应改为: xAccSR8 = 这次xAcc>>8。
目前还在进一步测试这个修改, 未来IC版本可能要做相应修改。

8、放大效果不好问题

在 720p 放大到 4k 输出效果不好, 字体周边毛刺较多, 看起来有模糊感, 采用 bicubic 算法驱动默认使用 PRECISE 配置, 改成 CATROM 后效果有所改善, 但 RK3368 效果比海思 3798M 还是差一些。

三、遗留未解决的问题

- 1、CABC 显示效果不好问题;
- 2、HDMI 1080i 模式时序错误问题;
- 3、average 算法绿线问题
- 4、720P to 4K 显示效果不好问题;
- 5、鼠标层出现带宽不足问题;

四、后期优化及建议

- 1、解决第三项中未解决的问题；
- 2、mmu 使能改成帧中断；
- 3、interlace 模式下区别帧中断和场中断，图层配置改成帧有效；
- 4、interlace 模式下缩放效果改善；
- 5、interlace 模式下 yact 偶数行的限制；
- 6、加入 480i、576i 分辨率支持；
- 7、增加 debug 接口，获取 VOP 当前送给 LVDS/EDP/MIPI/HDMI 的数据；
- 8、black en 在 yuv overlay 下的处理；
- 9、gamma 调整做到无缝切换；
- 10、layer0 alpha 支持；