

Rockchip

DRM Display Driver Development Guide

发布版本:**1.0**

日期:**2019.01**

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自所有者所有。

版权所有 © 2019 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：www.rock-chips.com

客户服务电话：+86-591-83991906

客户服务传真：+86-591-83951833

客户服务邮箱：service@rock-chips.com

前言

概述

本文档主要介绍 Rockchip 处理器基于 DRM 显示框架 VOP 以及相关显示接口的基本属性、工作流程和常见问题分析。目的是为了相关工程师能对 DRM 显示驱动框架和硬件接口有更好的理解，并通过常见问题的分析能快速定位问题、解决问题。

产品版本

芯片名称	内核版本
RK3036	4.4
RK312X/PX3SE	4.4
RK3288X	4.4
RK322X/RK312XH	4.4
RK3308	4.4
RK322XH/RK332X	4.4
RK3326/PX30	4.4
RK3368/PX5	4.4
RK3399	4.4

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师
- 硬件开发工程师

修订记录

日期	版本	作者	审核	修改说明
2019-1-21	V1.0	黄家钗	黄德胜	正式发布版本

目录

前言.....	III
目录.....	IV
1 DRM 概述.....	1
1.1 基本概念.....	1
1.2 DRM 显示通路.....	1
1.3 DRM 驱动和 libdrm 交互过程.....	1
2 DRM 驱动.....	2
2.1 uboot 驱动.....	2
2.1.1 驱动目录.....	2
2.1.2 驱动文件.....	2
2.1.3 接口说明.....	2
2.2 kernel 驱动.....	3
2.2.1 驱动目录.....	3
2.2.2 驱动文件.....	3
2.2.3 内核驱动加载流程.....	4
3 Display feature.....	5
3.1 各平台支持的显示接口及属性.....	5
3.2 最大输出分辨率和协议标准.....	5
3.2.1 RK3036 平台.....	5
3.2.2 RK312X/PX3SE 平台.....	5
3.2.3 RK322X/RK312XH 平台.....	5
3.2.4 RK3288X 平台.....	5
3.2.5 RK3308 平台.....	6
3.2.6 RK322XH/RK332X 平台.....	6
3.2.7 RK3326/px30 平台.....	6
3.2.8 RK3368/PX5 平台.....	6
3.2.9 RK3399 平台.....	6
4 硬件相关.....	6
4.1 RGB 输出/TTL 模式硬件连接.....	6
4.1.1 RGB888 24BIT 屏.....	6
4.1.2 RGB666 18 bit 屏.....	7
4.1.3 RGB565 16 bit 屏.....	8
4.1.4 MCU 屏.....	8
4.2 LVDS Data Mapping.....	10
4.2.1 6 bit output mode.....	10
4.2.2 8 bit output mode.....	10
4.2.3 10 bit output mode.....	11
5 扫描时序说明.....	13
5.1 常见的扫描时序图.....	13
5.2 DRM 对扫描时序的定义.....	14

5.3 软件配置的对应关系.....	14
6 常用的 debug 手段.....	14
6.1 dump 当前的显示状态.....	14
6.1.1 使用命令.....	14
6.1.2 参数说明.....	15
6.2 调整 drm log 等级抓 log.....	16
6.3 查看当前的显示时钟.....	16
6.4 强行开关显示设备.....	16
6.5 查看 drm buffer 使用情况.....	17
6.6 查看 gpio 的状态.....	17
6.7 modetest 使用.....	17
6.8 界面暂停、启动.....	17
6.9 查看 EDID 信息.....	18
6.10 查看 HDMI 的状态信息.....	18
6.11 Dump 当前显示的 buffer.....	18
6.11.1 使用说明.....	18
6.11.2 例子.....	18
7 FAQ.....	18
7.1 uboot logo 切换到内核 logo 出现闪屏/无法显示问题.....	18
7.1.1 DDR 变频导致.....	18
7.1.2 clk tree 变化导致.....	18
7.1.3 时钟被关闭导致.....	19
7.1.4 uboot logo 图片和 kernel logo 图片大小不一致.....	19
7.1.5 内核初始化过程一些电源/GPIO 被重新初始化.....	19
7.1.6 VOP 优先级配置问题.....	19
7.1.7 测试相关电源和信号.....	19
7.2 如何开关 uboot logo 显示.....	19
7.3 VOP POST_BUF_EMPTY.....	20
7.3.1 带宽不够.....	20
7.3.2 iommu 出错.....	20
7.3.3 logic 电压太低.....	20
7.3.4 AFBDC/IFBDC 对齐要求.....	20
7.4 drm 驱动 bind.....	20
7.5 uboot logo 要求.....	21
7.6 部分机器开机动画显示异常，桌面显示正常.....	21
7.7 显示效果调节.....	22
7.8 屏无法点亮/休眠唤醒显示异常/不显示问题.....	22
7.9 rk3308 显示特殊性.....	22
7.9.1 uboot 显示.....	22
7.9.2 CMA_SIZE 的修改.....	22
7.10 OLED 屏余晖确认.....	23
7.11 各种接口屏配置.....	23
8 参考文档.....	23

1 DRM 概述

DRM 全称是 Direct Rendering Manager，进行显示输出管理、buffer 分配、帧缓冲。对应 userspace 库为 libdrm，libdrm 库提供了一系列友好的控制封装，使用户可以方便的进行显示的控制和 buffer 申请。DRM 的设备节点为 `"/dev/dri/cardX"`，X 为 0-15 的数值，默认使用的是 `/dev/dri/card0`。

从 linux 4.4 内核开始，Rockchip 显示驱动全部切到 DRM 显示框架。

1.1 基本概念

CRTC: 显示控制器，在 rockchip 平台是 SOC 内部 VOP (部分文档也称为 LCDC) 模块的抽象；

Plane: 图层，在 rockchip 平台是 SOC 内部 VOP (LCDC) 模块 win 图层的抽象；

Encoder: 输出转换器，指 RGB、LVDS、DSI、eDP、HDMI、CVBS、VGA 等显示接口；

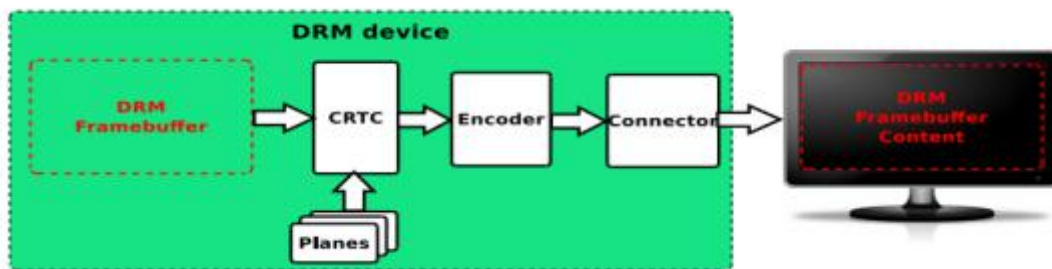
Connector: 连接器，指 encoder 和 panel 之间交互的接口部分；

Bridge: 桥接设备，一般用于注册 encoder 后面另外再接的转换芯片，如 DSI2HDMI 转换芯片。

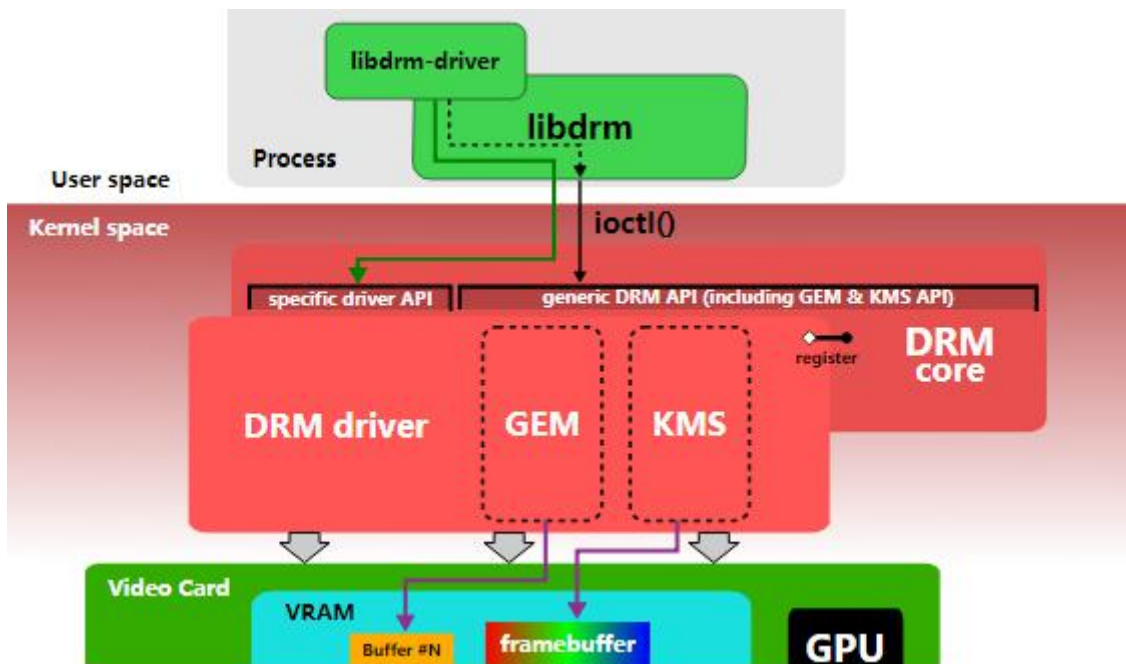
Panel: 泛指屏，各种 LCD、HDMI 等显示设备的抽象；

GEM: buffer 管理和分配，类似 android 下的 ion。

1.2 DRM 显示通路



1.3 DRM 驱动和 libdrm 交互过程



2 DRM 驱动

2.1 uboot 驱动

2.1.1 驱动目录

drivers/video/drm/

2.1.2 驱动文件

Driver	File
Core	rockchip_display.c rockchip_crtc.c rockchip_connector.c rockchip_phy.c rockchip_panel.c
VOP	rockchip_vop.c rockchip_vop_reg.c
eDP	rockchip_analogix_dp.c rockchip_analogix_dp_reg.c
MIPI-DSI	rockchip_mipi_dsi.c rockchip-dw-mipi-dsi.c inno_video_phy.c
LVDS	rockchip_lvds.c inno_video_phy.c
RGB	rockchip_rgb.c inno_video_phy.c
HDMI	dw_hdmi.c rockchip_dw_hdmi.c rockchip-inno-hdmi-phy.c
TVE/CVBS	rockchip_drm_tve.c
Panel	panel_simple.c rockchip_dsi_panel.c

2.1.3 接口说明

- (1) void rockchip_show_logo(void)
显示 U-Boot logo 和 kernel logo
- (2) void rockchip_show_bmp(const char *bmp)
显示指定的 bmp 图片，目前主要用于充电 logo 的显示：
- (3) rockchip_display_fixup(void *blob)
将 U-Boot 中确定的一些变量通过 dtb 文件传递给内核，包括 kernel logo 的大小、地址、格式、输出扫描时序以及过扫描的配置等信息。

2.2 kernel 驱动

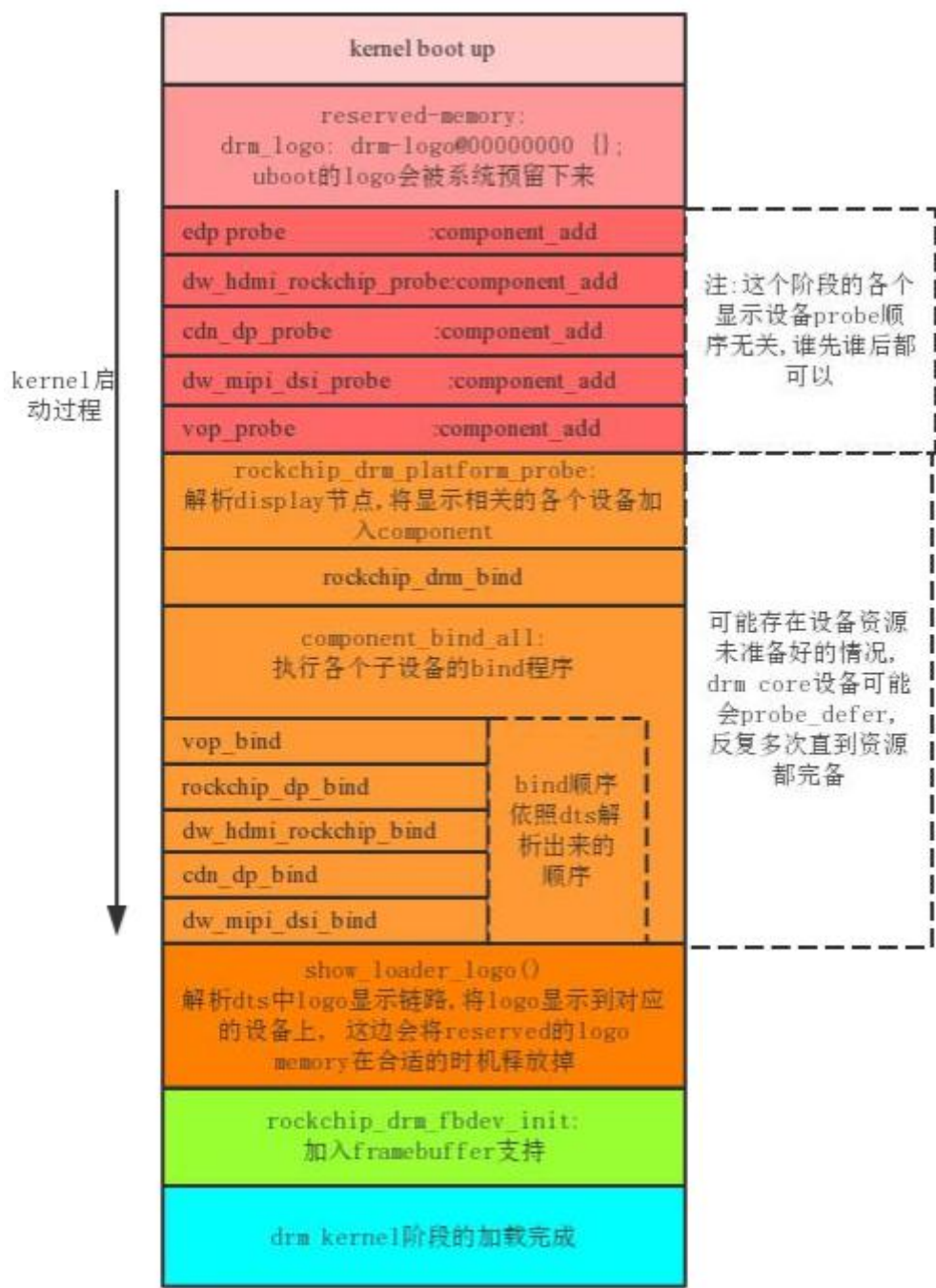
2.2.1 驱动目录

drivers/gpu/drm/rockchip/
drivers/gpu/drm/bridge/analogix/
drivers/gpu/drm/bridge/synopsys/
drivers/phy/rockchip/

2.2.2 驱动文件

Driver	File	Doc
Core	rockchip_drm_drv.c rockchip_drm_fb.c rockchip_drm_fbdev.c	rockchip-drm.txt
VOP	rockchip_drm_vop.c rockchip_vop_reg.c	rockchip-vop.txt
LVDS	rockchip_lvds.c phy-rockchip-inno-video-phy.c	rockchip-lvds.txt
MIPI-DSI	dw-mipi-dsi.c phy-rockchip-inno-video-phy.c	dw_mipi_dsi_rockchip.txt phy-rockchip-inno-video-combo-phy.txt
eDP	analogix_dp-rockchip.c analogix_dp_core.c analogix_dp_reg.c phy-rockchip-dp.c	analogix_dp-rockchip.txt analogix_dp.txt rockchip-dp-phy.txt
HDMI	inno_hdmi.c dw-hdmi.c dw_hdmi-rockchip.c dw-hdmi-hdcp.c dw-hdmi-cec.c dw-hdmi-i2s-audio.c phy-rockchip-inno-hdmi-phy.c	inno_hdmi-rockchip.txt dw_hdmi-rockchip.txt phy-rockchip-inno-hdmi-phy.txt
TVE/CVBS	rockchip_drm_tve.c	rockchip_drm_tve.txt
Panel	panel-simple.c	simple-panel.txt

2.2.3 内核驱动加载流程



3 Display feature

3.1 各平台支持的显示接口及属性

SOC	Display output interface										Other feature								
	RGB	MCU	LVDS	DUAL LVDS	MIPI	DUAL MIPI	EDP	DP	HDMI	CVBS	4K	i-MODE	FBDC	BCSH	POST SCAL	xMIR	yMIR	MULI AREA	MMU
RK3066/PX3	√	×	√	×	×	×	×	×	√	×	×	×	×	×	×	×	×	×	×
RK3188/PX5	√	√	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
RK3126/RK3126C	√	×	√	×	√	×	×	×	×	×	×	×	×	√	×	×	×	×	√
RK3128/PX3SE	√	×	√	×	√	×	×	×	√	√	×	×	×	√	×	×	×	×	√
RK3036	×	×	×	×	×	×	×	×	√	√	×	×	×	√	×	×	×	×	√
RK322X/RK312XH	×	×	×	×	×	×	×	×	√	√	√	√	×	√	√	√	√	×	√
RK322XH/RK332X	×	×	×	×	×	×	×	×	√	√	√	√	×	√	√	√	√	×	√
SOFLA 3GR	√	×	√	×	√	×	×	×	×	×	×	×	×	√	×	×	×	×	√
RV1108	√	×	×	×	√	×	×	×	√	√	×	√	×	√	×	×	×	×	×
RK3288	√	×	√	√	√	√	√	×	√	×	√	×	×	√	√	√	√	√	√
RK3368	√	×	√	×	√	√	√	×	√	×	√	×	√	√	√	√	√	√	√
RK3399/RK3399 Pro	×	×	×	×	√	√	√	√	√	×	√	√	√	√	√	√	√	√	√
RK3326/PX30	√	√	√	×	√	×	×	×	√	×	×	√	√	√	×	×	×	√	√
RK3308	√	√	×	×	×	×	×	×	×	×	×	×	×	√	×	×	×	×	×

3.2 最大输出分辨率和协议标准

3.2.1 RK3036 平台

显示接口	最大输出	协议标准
HDMI	1920x1080@60hz	支持 HDMI 1.4a 协议标准
CVBS	720x480i@60hz/720x576@50hz	支持 NTSC/PAL 标准输出

3.2.2 RK312X/PX3SE 平台

显示接口	最大输出	协议标准
RGB	1280x800@60hz	支持 RGB666/sRGB888
LVDS	1280x800@60hz	支持 VESA 和 JEIDA LVDS 数据格式
MIPI	1280x800@60hz	支持 DSI v1.0, DCS v1.0, DPHY v1.0 协议标准
HDMI	1920x1080@60hz	支持 HDMI 1.4a 协议标准

3.2.3 RK322X/RK312XH 平台

显示接口	最大输出	协议标准
HDMI	4096x2160@60hz	支持 HDMI 1.4a 和 2.0 协议标准
CVBS	720x480i@60hz/720x576@50hz	支持 NTSC/PAL 标准输出

3.2.4 RK3288X 平台

显示接口	最大输出	协议标准
RGB	1920x1080@60hz	支持 RGB888/RGB666/sRGB888
LVDS	单通道: 1280x800@60hz 双通道: 1920x1080@60hz	支持 VESA 和 JEIDA LVDS 数据格式
eDP	2560x1600@60hz	支持 DP1.2a 和 eDP1.3 协议标准
MIPI	单通道: 1920x1080@60hz 双通道: 2560x1600@60hz	支持 DSI v1.1, DCS v1.1, DPHY v1.1 协议标准

HDMI	VOP BIG:: 3840x2160@60hz VOP LIT: 1920x1080@60hz	支持 HDMI 1.4a 和 2.0 协议标准
------	---	-------------------------

3.2.5 RK3308 平台

显示接口	最大输出	协议标准
RGB	1280x800@60hz	支持 RGB888/RGB666/sRGB888/MCU

3.2.6 RK322XH/RK332X 平台

显示接口	最大输出	协议标准
HDMI	4096x2160@60hz	支持 HDMI 1.4a 和 2.0 协议标准
CVBS	720x480i@60hz/720x576@50hz	支持 NTSC/PAL 标准输出

3.2.7 RK3326/px30 平台

显示接口	最大输出	协议标准
RGB	1280x800@60hz	支持 RGB888/RGB666/sRGB888/MCU
LVDS	1280x800@60hz	支持 VESA 和 JEIDA LVDS 数据格式
MIPI	1920x1080@60hz	支持 DSI v1.0, DCS v1.0, DPHY v1.0 协议标准

3.2.8 RK3368/PX5 平台

显示接口	最大输出	协议标准
RGB	1920X1080@60hz	支持 RGB888/RGB666/sRGB888
LVDS	1280x800@60hz	支持 VESA 和 JEIDA LVDS 数据格式
eDP	2560x1600@60hz	支持 DP1.2a 和 eDP1.3 协议标准
MIPI	1920x1080@60hz	支持 DSI v1.0, DCS v1.0, DPHY v1.0 协议标准
HDMI	VOP BIG:: 4096X2160@60hz	支持 HDMI 1.4a 和 2.0 协议标准

3.2.9 RK3399 平台

显示接口	最大输出	协议标准
eDP	VOP BIG: 3840x2160@60hz VOP LITE: 2560x1600@60hz	支持 DP1.2a 和 eDP1.3 协议标准
MIPI	单通道: 1920x1080@60hz 双通道: 2560x1600@60hz	支持 DSI v1.1, DCS v1.1, DPHY v1.1 协议标准
HDMI	VOP BIG:: 4096X2160@60hz VOP LITE: 2560x1600@60hz	支持 HDMI 1.4a 和 2.0a 协议标准
DP	VOP BIG:: 4096X2160@60hz VOP LITE: 2560x1600@60hz	支持 DP 1.2 协议标准

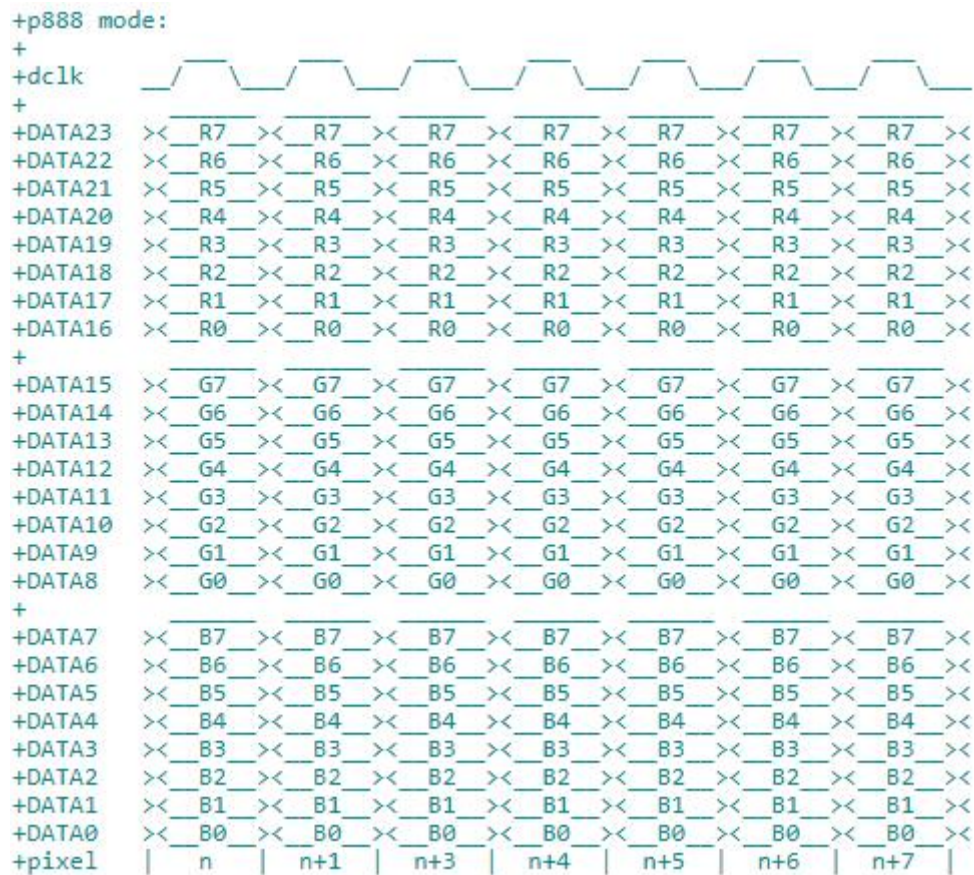
4 硬件相关

4.1 RGB 输出/TTL 模式硬件连接

4.1.1 RGB888 24BIT 屏

硬件可以参考下面表格 screen_face 为 OUT_P888 的连接, 软件在 RGB 节点配置: bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;

传输时序:

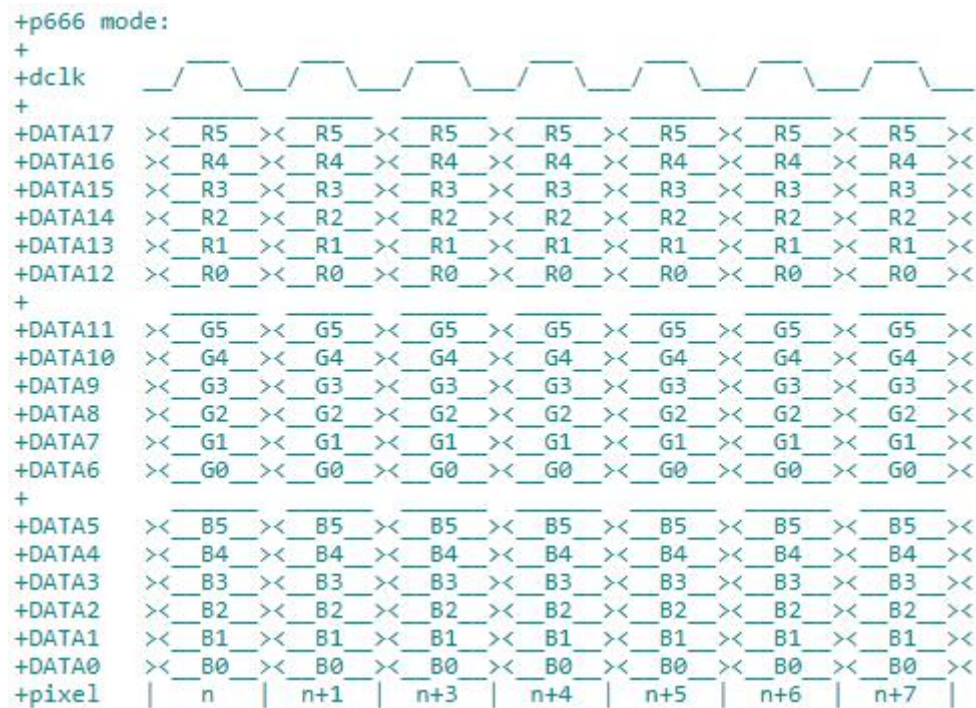


4.1.2 RGB666 18 bit 屏

硬件有两种连接方式:

- (1). 硬件按下面表格 screen_face 为 OUT_D888_P666 的方式连接, 软件在 RGB 节点配置:
bus-format = <MEDIA_BUS_FMT_RGB666_1X24_CPADHI>;
- (2). 硬件按下面表格 screen_face 为 OUT_P666 的方式连接, 软件在 RGB 节点配置: bus-format = <MEDIA_BUS_FMT_RGB666_1X18>;

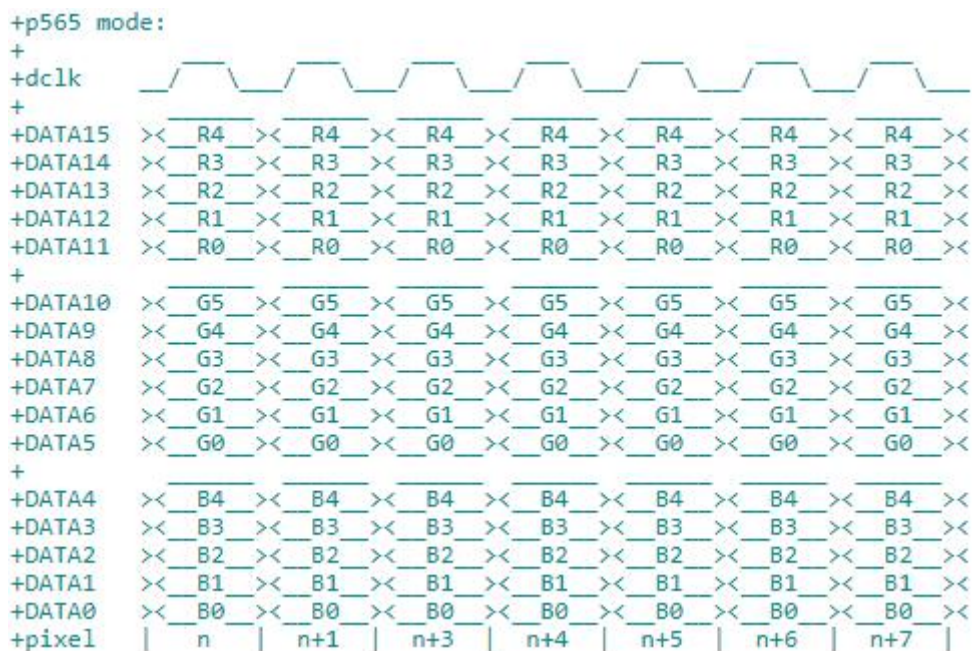
传输时序:



4.1.3 RGB565 16 bit 屏

按下面表格 screen_face 为 OUT_D888_P565 的方式连接，软件在 RGB 节点配置：bus-format = <MEDIA_BUS_FMT_RGB565_1X16>;

传输时序：



4.1.4 MCU 屏

硬件连接可以参考上面(1),(2),(3)的连接方式，MCU 比较特殊的是 RGB 模式下的 DCLK/VSYNC/HSYNC/DEN 在 MCU 模式下对应 RS/CS/WEN/REN 引脚；

Rockchip DRM Display Driver Development guide

Display mode	RGB Parallel 24-bit	RGB Parallel 18-bit	RGB Parallel 18-bit	RGB Parallel 16-bit	RGB Parallel 16-bit	ITU656 Mode0	ITU656 Mode1	ITU656 Mode2	MCU mode
Screen_ face	OUT_P888	OUT_D888_P666	OUT_P6666	OUT_D888_P565	OUT_P565	OUT_S888/OUT_S888DUMY	OUT_S8888/OUT_S888DUMY	OUT_S88888/OUT_S8888DUMY	OUT_P8888
DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	RS
VSYNC	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC				CS
HSYNC	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC				WEN
DEN	DEN	DEN	DEN	DEN	DEN				REN
DATA	DATA[23:0] DATA[15:1] DATA[7:2]	DATA[23:1] DATA[15:1] DATA[7:2]	DATA[17:0] DATA[15:1] DATA[7:2]	DATA[23:1] DATA[15:1] DATA[7:2]	DATA[15:0] DATA[15:1] DATA[7:2]	DATA[7:0] DATA[15:1] DATA[7:2]	DATA[15:8] DATA[15:1] DATA[7:2]	DATA[14:7] DATA[15:1] DATA[7:2]	
D23	R7	R5	-	R4	-	-	-	-	D23
D22	R6	R4	-	R3	-	-	-	-	D22
D21	R5	R3	-	R2	-	-	-	-	D21
D20	R4	R2	-	R1	-	-	-	-	D20
D19	R3	R1	-	R0	-	-	-	-	D19
D18	R2	R0	-	-	-	-	-	-	D18
D17	R1	-	R5	-	-	-	-	-	D17
D16	R0	-	R4	-	-	-	-	-	D16
D15	G7	G5	R3	G5	R4	-	D7	-	D15
D14	G6	G4	R2	G4	R3	-	D6	D7	D14
D13	G5	G3	R1	G3	R2	-	D5	D6	D13
D12	G4	G2	R0	G2	R1	-	D4	D5	D12
D11	G3	G1	G5	G1	R0	-	D3	D4	D11
D10	G2	G0	G4	G0	G5	-	D2	D3	D10
D9	G1	-	G3	-	G4	-	D1	D2	D9
D8	G0	-	G2	-	G3	-	D0	D1	D8
D7	B7	B5	G1	B4	G2	D7	-	D0	D7
D6	B6	B4	G0	B3	G1	D6	-	-	D6
D5	B5	B3	B5	B2	G0	D5	-	-	D5
D4	B4	B2	B4	B1	B4	D4	-	-	D4
D3	B3	B1	B3	B0	B3	D3	-	-	D3
D2	B2	B0	B2	-	B2	D2	-	-	D2
D1	B1	-	B1	-	B1	D1	-	-	D1
D0	B0	-	B0	-	B0	D0	-	-	D0

4.2 LVDS Data Mapping

可以通过 dts 文件 LVDS 节点下的 rockchip,data-mapping 属性和 rockchip,data-width 属性来配置哪种模式输出。

4.2.1 6 bit output mode

采用 4+1 的传输模式，即 4 组数据信号加一组时钟信号，最后一组数据信号传输无效数据。

JEIDA_6BIT:

bus-format = <MEDIA_BUS_FMT_RGB666_1X7X3_SPWG>;

*		JEIDA_6BIT	
Y 0	TX0	R0	
	TX1	R1	
	TX2	R2	
	TX3	R3	
	TX4	R4	
	TX6	R5	
	TX7	G0	
Y 1	TX8	G1	
	TX9	G2	
	TX12	G3	
	TX13	G4	
	TX14	G5	
	TX15	B0	
	TX18	B1	
Y 2	TX19	B2	
	TX20	B3	
	TX21	B4	
	TX22	B5	
	TX24	HSYNC	
	TX25	VSYNC	
	TX26	ENABLE	
Y 3	TX27	GND	
	TX5	GND	
	TX10	GND	
	TX11	GND	
	TX16	GND	
	TX17	GND	
	TX23	RSVD	

4.2.2 8 bit output mode

采用 4+1 的传输模式，即 4 组数据信号加一组时钟信号。

VESA_8BIT:

bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;

JEIDA_8BIT:

bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_JEIDA>;

		VESA_8BIT	JEIDA_8BIT
Y 0	TX0	R0	R2
	TX1	R1	R3
	TX2	R2	R4
	TX3	R3	R5
	TX4	R4	R6
	TX6	R5	R7
	TX7	G0	G2
Y 1	TX8	G1	G3
	TX9	G2	G4
	TX12	G3	G5
	TX13	G4	G6
	TX14	G5	G7
	TX15	B0	B2
	TX18	B1	B3
Y 2	TX19	B2	B4
	TX20	B3	B5
	TX21	B4	B6
	TX22	B5	B7
	TX24	HSYNC	HSYNC
	TX25	VSYNC	VSYNC
	TX26	ENABLE	ENABLE
Y 3	TX27	R6	R0
	TX5	R7	R1
	TX10	G6	G0
	TX11	G7	G1
	TX16	B6	B0
	TX17	B7	B1
	TX23	RSVD	RSVD

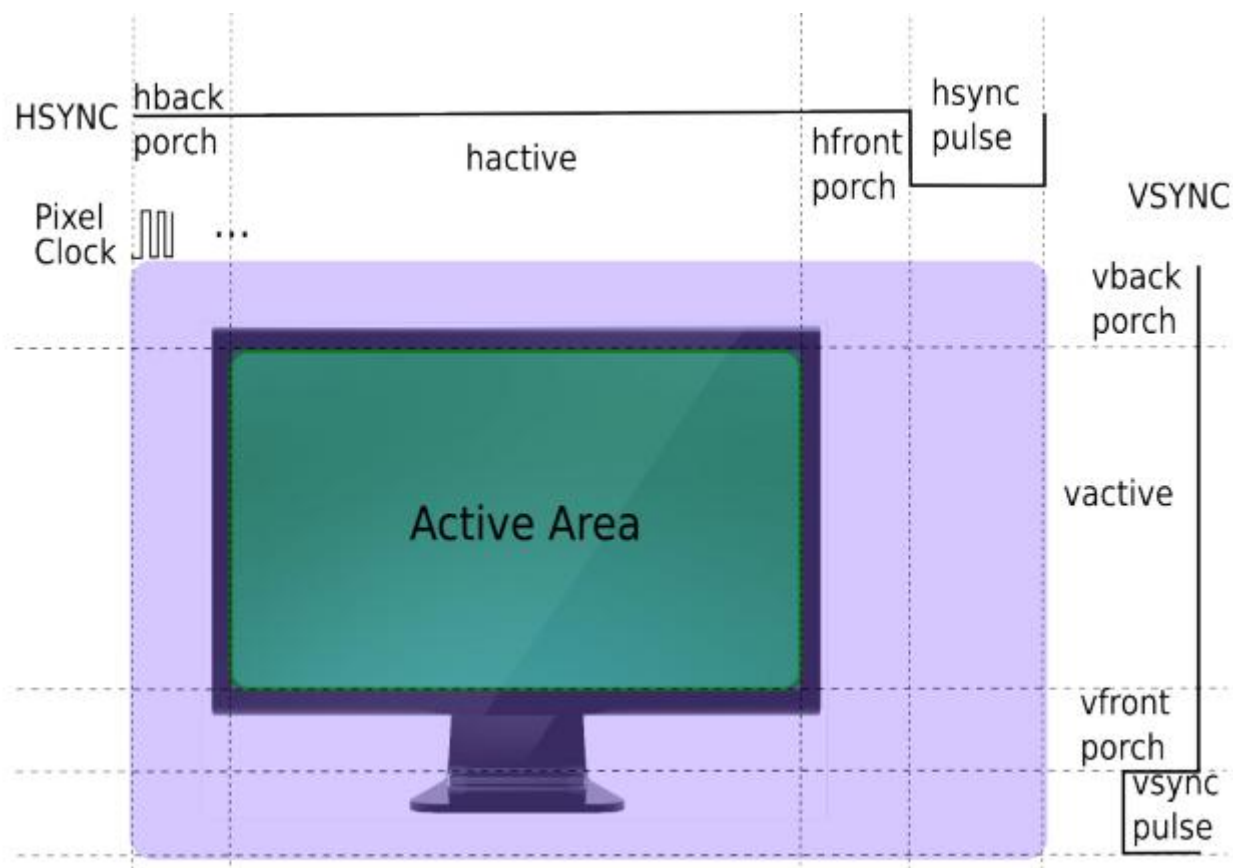
4.2.3 10 bit output mode

采用 5+1 的传输模式，即 5 组数据信号加一组时钟信号。
目前 DRM 驱动尚未加入该支持。

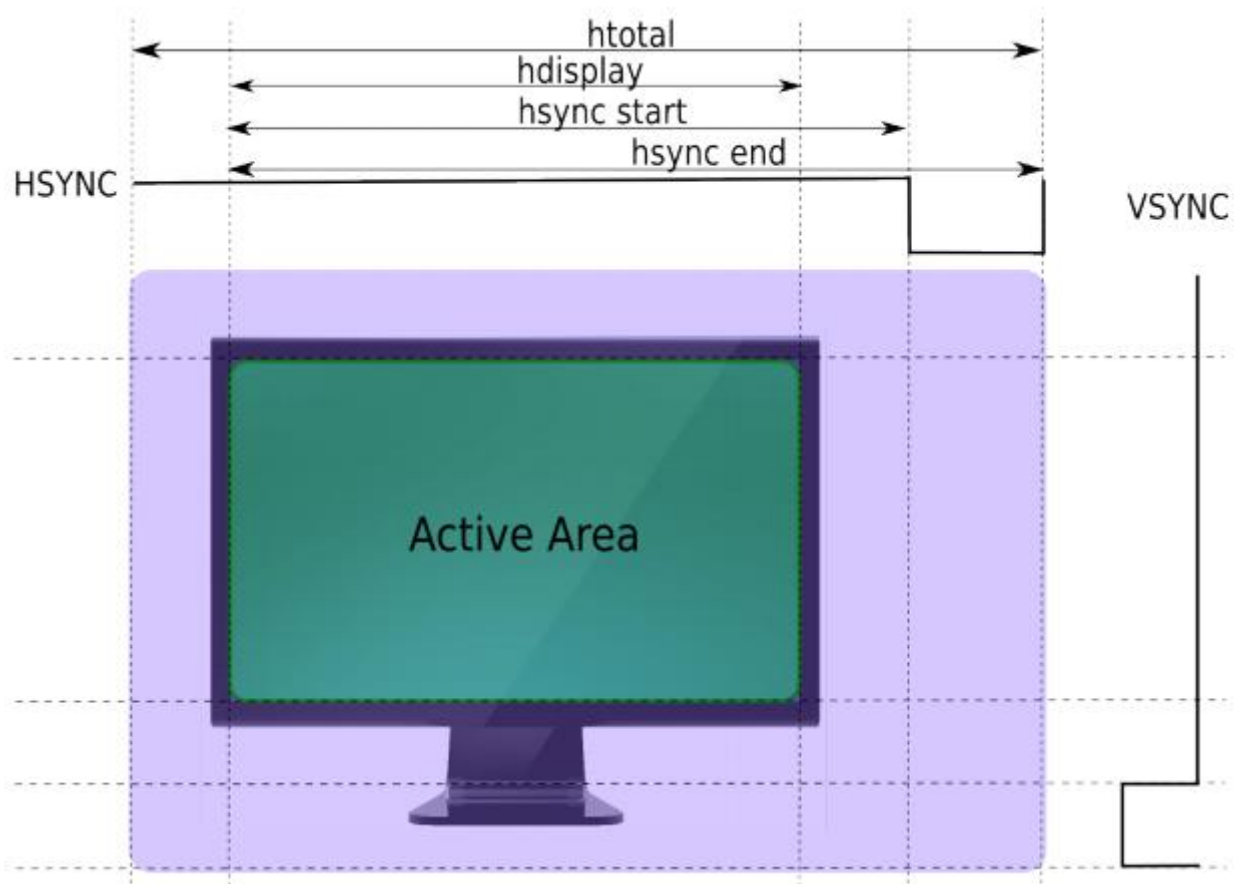
		VESA_10BIT	JEIDA_10BIT
Y 0	TX0	R0	R4
	TX1	R1	R5
	TX2	R2	R6
	TX3	R3	R7
	TX4	R4	R8
	TX6	R5	R9
	TX7	G0	G4
Y 1	TX8	G1	G5
	TX9	G2	G6
	TX12	G3	G7
	TX13	G4	G8
	TX14	G5	G9
	TX15	B0	B4
	TX18	B1	B5
Y 2	TX19	B2	B6
	TX20	B3	B7
	TX21	B4	B8
	TX22	B5	B9
	TX24	HSYNC	HSYNC
	TX25	VSNC	VSNC
	TX26	ENABLE	ENABLE
Y 3	TX27	R6	R2
	TX5	R7	R3
	TX10	G6	G2
	TX11	G7	G3
	TX16	B6	B2
	TX17	B7	B3
	TX23	GND	GND
Y 4	TX27	R8	R0
	TX5	R9	R1
	TX10	G8	G0
	TX11	G9	G1
	TX16	B8	B0
	TX17	B9	B1
	TX23	GND	GND

5 扫描时序说明

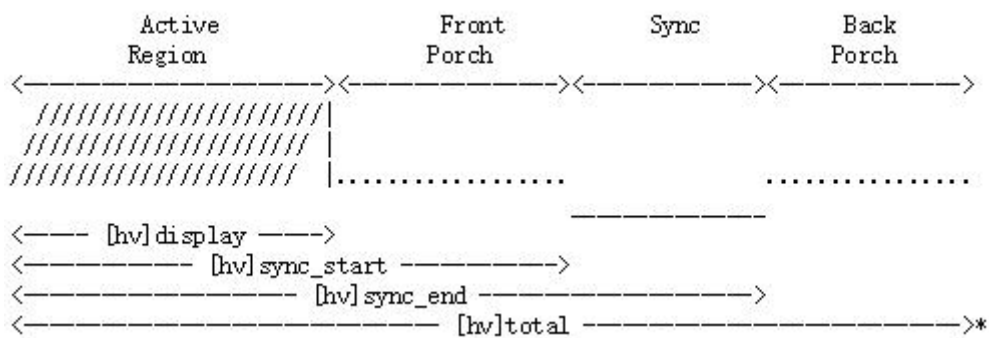
5.1 常见的扫描时序图



5.2 DRM 对扫描时序的定义



5.3 软件配置的对应关系



6 常用的 debug 手段

6.1 dump 当前的显示状态

6.1.1 使用命令

```
cat /sys/kernel/debug/dri/0/summary
```

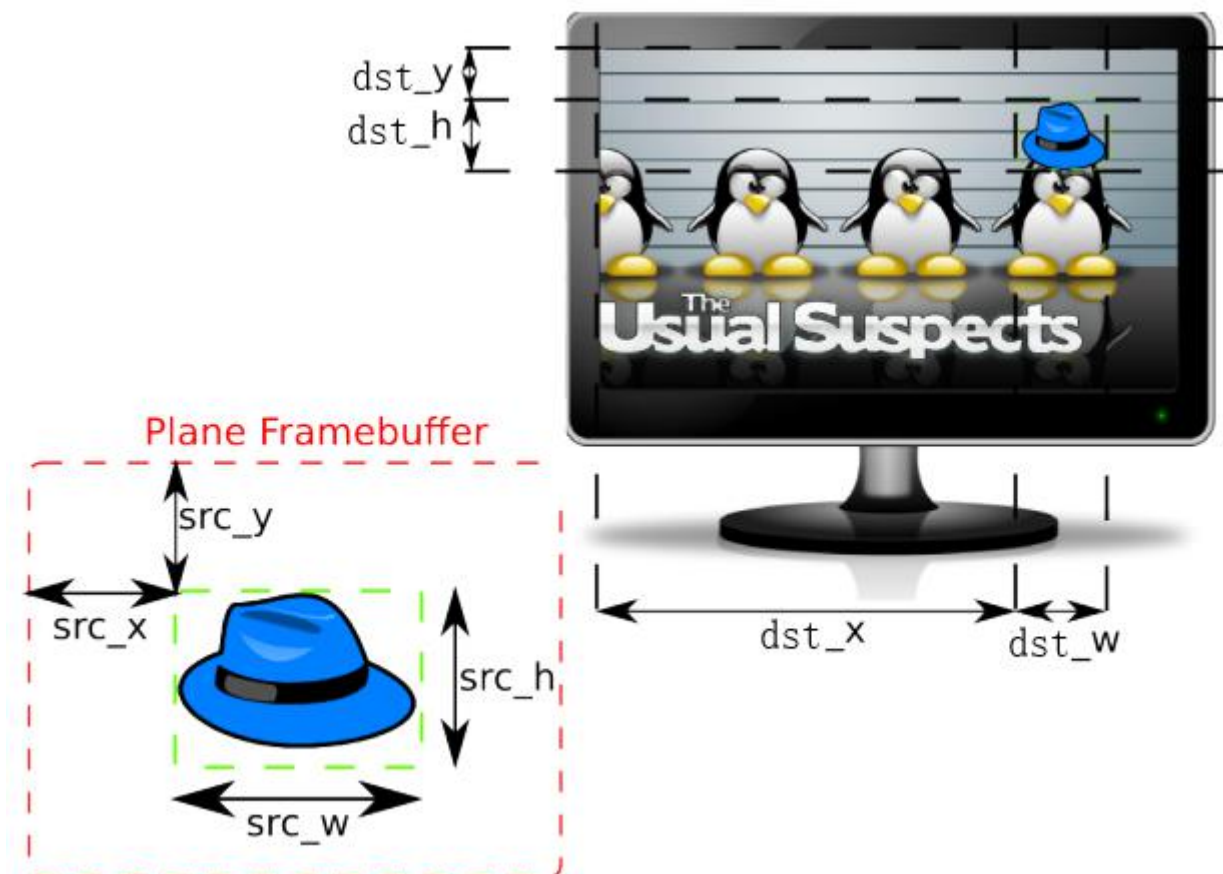
```

/ # cat /sys/kernel/debug/dri/0/summary
VOP [ff900000.vop]: ACTIVE
Connector: eDP
  overlay_mode[0] bus_format[1009] output_mode[f] color_space[0]
Display mode: 1536x2048p60
  clk[200000] real_clk[200000] type[0] flag[a]
  H: 1536 1548 1564 1612
  V: 2048 2056 2060 2068
win0-0: ACTIVE
  format: XR24 little-endian (0x34325258) SDR[0] color_space[0]
  csc: y2r[0] r2r[0] r2y[0] csc mode[0]
  zpos: 0
  src: pos[0x0] rect[1536x2048]
  dst: pos[0x0] rect[1536x2048]
  buf[0]: addr: 0x0000000000000000 pitch: 6144 offset: 0
win1-0: DISABLED
win2-0: DISABLED
win2-1: DISABLED
win2-2: DISABLED
win2-3: DISABLED
win3-0: DISABLED
win3-1: DISABLED
win3-2: DISABLED
win3-3: DISABLED
post: sdr2hdr[0] hdr2sdr[0]
pre : sdr2hdr[0]
post CSC: r2y[0] y2r[0] CSC mode[1]
VOP [ff8f0000.vop]: ACTIVE
Connector: DSI
  overlay_mode[0] bus_format[100a] output_mode[0] color_space[0]
Display mode: 1280x720p29
  clk[96000] real_clk[96000] type[8] flag[a]
  H: 1280 2280 2480 2680
  V: 720 920 1120 1220
win0-0: ACTIVE
  format: XR24 little-endian (0x34325258) SDR[0] color_space[0]
  csc: y2r[0] r2r[0] r2y[0] csc mode[0]
  zpos: 0
  src: pos[0x0] rect[1280x720]
  dst: pos[0x0] rect[1280x720]
  buf[0]: addr: 0x0000000000000000 pitch: 6144 offset: 0
win2-0: DISABLED
win2-1: DISABLED

```

6.1.2 参数说明

- (1) 两个红色方框表示两个显示设备使用的 vop 分别是 ff900000.vop 和 ff8f0000.vop;
- (2) 绿色部分表示 connector 信息，两个显示设备分别为 eDP 屏和 MIPI 屏;
- (3) 粉色部分为显示模式，可以知道具体的时序、DCLK 以及帧率，上图中两个设备分别为分辨率为 1536x2048p60 的 eDP 屏和分辨率 1280x720p29 的 MIPI 屏;
- (4) 蓝色部分是 VOP 图层信息，第一个显示设备打开 win0 图层，大小为 1536x2048 格式为 XRGB，第二个显示设备打开 win0 图层，大小 1280x720 格式为 XRGB，src 和 dst 表示源数据和显示的大小和位置，如下图所示:



(5) 橙色部分为 VOP HDR、CSC 的一些状态信息;

6.2 调整 drm log 等级抓 log

debug: Enable debug output, where each bit enables a debug category.

Bit 0 (0x01) will enable CORE messages (drm core code)

Bit 1 (0x02) will enable DRIVER messages (drm controller code)

Bit 2 (0x04) will enable KMS messages (modesetting code)

Bit 3 (0x08) will enable PRIME messages (prime code)

Bit 4 (0x10) will enable ATOMIC messages (atomic code)

Bit 5 (0x20) will enable VBL messages (vblank code) (int)

示例:

echo 0x0c > /sys/module/drm/parameters/debug 打开了 KMS, PRIME 这些的打印, kernel 驱动中使用 DRM_DEBUG_KMS 和 DRM_DEBUG_PRIME 打印的信息都可以打印出来。

6.3 查看当前的显示时钟

有时候需要知道 VOP dclk 和 aclk 的值, 以及 pll 源, 可以通过下面的命令获取:

```
cat /sys/kernel/debug/clk/clk_summary
```

如果只要关注 vop 的时钟, 可以使用:

```
cat /sys/kernel/debug/clk/clk_summary | grep vop
```

6.4 强行开关显示设备

以 LVDS 为例:

关 LVDS: echo off > /sys/class/drm/card0-LVDS-1/status

开 LVDS: echo on > /sys/class/drm/card0-LVDS-1/status

6.5 查看 drm buffer 使用情况

cat /sys/kernel/debug/dri/0/mm_dump, 可以知道当前 drm 这边 buffer 的使用情况。

```
/ # cat /sys/kernel/debug/dri/0/mm_dump
0x0000000000000000-0x000000000000c00000: 12582912: used
0x000000000000c00000-0x000000001000000000: 4282384384: free
total: 4294967296, used 12582912 free 4282384384
```

6.6 查看 gpio 的状态

cat /sys/kernel/debug/gpio, 通过这个命令可以知道当前一些 GPIO 的状态, 确认和屏相关的 GPIO 是否有被正常设置。

```
GPIOs 0-31, platform/pinctrl, gpio0:
gpio-1  (          |vcc_sd          ) out lo
gpio-4  (          |bt_default_wake_host) in  lo
gpio-5  (          |GPIO Key Power  ) in  hi
gpio-9  (          |bt_default_reset  ) out lo
gpio-10 (          |reset           ) out hi

GPIOs 32-63, platform/pinctrl, gpio1:
gpio-34 (          |int-n           ) in  hi
gpio-45 (          |enable          ) out hi
gpio-46 (          |vsel            ) out lo
gpio-49 (          |vsel            ) out lo
```

6.7 modetest 使用

```
/ # modetest -h
usage: modetest [-cDdefMPpsCvw]

Query options:

    -c      list connectors
    -e      list encoders
    -f      list framebuffers
    -p      list CRTCs and planes (pipes)

Test options:

    -P <plane_id>@<crtc_id>:<w>x<h>[+<x>+<y>][*<scale>][@<format>] set a plane
    -s <connector_id>[,<connector_id>][@<crtc_id>]:<mode>[-<vrefresh>][@<format>] set a mode
    -C      test hw cursor
    -v      test vsynced page flipping
    -w <obj_id>:<prop_name>:<value> set property

Generic options:

    -d      drop master after mode set
    -M module      use the given driver
    -D device      use the given device
```

6.8 界面暂停、启动

暂停进程(只能短暂暂停, 一段时间后会自动恢复)

kill -STOP `pgrep surfaceflinger`

恢复进程

kill -CONT `pgrep surfaceflinger`

6.9 查看 EDID 信息

以 HDMI 为例: `cat /sys/class/drm/card0-HDMI-A-1/edid > /data/edid.bin`

6.10 查看 HDMI 的状态信息

`cat /sys/kernel/debug/dw-hdmi/status`

```
/ # cat /sys/kernel/debug/dw-hdmi/status
PHY: enabled           Mode: HDMI
Pixel Clk: 148500000Hz  TMDs Clk: 148500000Hz
Color Format: RGB       Color Depth: 8 bit
Colorimetry: ITU.BT709 EOTF: Off
```

6.11 Dump 当前显示的 buffer

6.11.1 使用说明

```
/sys/kernel/debug/dri/0/ff900000.vop/vop_dump # cat dump
echo dump > dump to dump one frame
echo dumpn > dump to start vop keep dumping
echo dumpoff > dump to stop keep dumping
echo dumpn > dump n is the number of dump times
dump path is /data/vop_buf
if fd err = -3 try rm -r /data/vopbuf echo dump1 > dump can fix it
if fd err = -28 save needed data try rm -r /data/vopbuf
```

6.11.2 例子

- (1) dump 一帧当前显示的 buffer:
`echo dump > /sys/kernel/debug/dri/0/ff900000.vop/vop_dump/dump`
- (2) 连续 dump n 帧显示的 buffer
`echo dumpn > /sys/kernel/debug/dri/0/ff900000.vop/vop_dump/dump`
- (3) dump 出来的文件保存在/data/vop_buf/, 可以使用 7yuv 软件查看。

7 FAQ

7.1 uboot logo 切换到内核 logo 出现闪屏/无法显示问题

7.1.1 DDR 变频导致

DDR 变频导致, 可以做如下修改尝试:

- (1) 关闭 dts 文件中 ddr 变频节点, 保证内核阶段不做 DDR 变频, 修改方法如下:

```
&dmc {
    status = "disabled";
};
```

- (2) 关闭 DDR 变频时对 DCLK 的调整, 修改方法如下:

```
&dmc {
    vop-dclk-mode = <1>;
};
```

7.1.2 clk tree 变化导致

部分平台 uboot 中的 clk tree 配置和内核的 clk tree 配置是独立的, 如果两个驱动的 clk 策略不一致,

就会出现在 clk 重新初始化的时候出现闪屏问题，以 RK3399 为例，可以按如下方法确认：

(1) 在 rk3399_clk_init()@kernel/drivers/clk/rockchip/clk-rk3399.c 函数入口处加上 while(1)；确认是否会有闪屏问题；

(2) 在 rk3399_clk_init()@kernel/drivers/clk/rockchip/clk-rk3399.c 函数结束处加上 while(1)，确认是否会有闪屏问题；

(3) 如果步骤 a 中无闪屏现象步骤 b 中有闪屏现象，那基本可以确认是 clk tree 变化导致闪屏问题，可以直接找对应平台 cru 负责人或者提交 redmine 并说明转给 pll 相关负责人。

7.1.3 时钟被关闭导致

有一些流程上的问题或者软件上的 bug 可能存在一些必要的时钟在驱动注册的时候没有被使能，导致这些时钟在内核跑完后被框架自动关闭从而导致显示异常，可以尝试按下面的修改默认不关闭时钟做测试：

在 dts 文件中，找到 chosen 节点，在 bootargs 末尾加上 clk_ignore_unused：如 bootargs = "xxxx clk_ignore_unused"；

```
--- a/arch/arm64/boot/dts/rockchip/rk3399-linux.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-linux.dtsi
@@ -47,7 +47,7 @@
     compatible = "rockchip,linux", "rockchip,rk3399";

     chosen {
-        bootargs = "earlycon=uart8250,mmio32,0xff1a0000";
+        bootargs = "earlycon=uart8250,mmio32,0xff1a0000 clk_ignore_unused";
     };
```

7.1.4 uboot logo 图片和 kernel logo 图片大小不一致

rockchip 平台有要求 uboot logo 和 kernel logo 的图片分辨率大小一样，如果出现 uboot logo 显示正常，到内核阶段显示异常，可以确认下 kernel 目录下 logo.bmp 和 logo_kernel.bmp 分辨率是否一致：

```
$file logo.bmp logo_kernel.bmp
logo.bmp:      PC bitmap, windows 3.x format, 654 x 258 x 8
logo_kernel.bmp: PC bitmap, windows 3.x format, 654 x 258 x 8
```

7.1.5 内核初始化过程一些电源/GPIO 被重新初始化

该问题涉及的可能性很多，总之在新项目 porting 过程中要及时确认显示相关的 GPIO 电源是否和其他模块有冲突；

如果 dts 确认无误，可以在内核代码搜索串口中的关键字，通过二分法在各个模块加载的位置 while 住，逐步确认导致闪屏问题；

7.1.6 VOP 优先级配置问题

如果 VOP 优先级没有被配置最高，有可能在内核加载阶段被其他 IP 抢占总线导致闪屏问题，该问题一般会在 SDK 发布前 Fix。

7.1.7 测试相关电源和信号

如果以上还未找到闪屏问题，请使用示波器抓取 clk、data 和电源等相关信号从 uboot 到内核阶段的波形图并提交 redmine。

7.2 如何开关 uboot logo 显示

以 eDP 为例，可以在 dts 文件里找到 route_edp 节点，配置 status 状态：


```
&route_edp {
    //status = "okay";
};
```

如果 uboot logo 关闭，需要等到安卓起来后屏才会正常显示；

7.3 VOP POST_BUF_EMPTY

```
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
```

7.3.1 带宽不够

如果系统带宽不够会导致 VOP 不能及时取到数据，从而报 post empty，可以做如下尝试：

- (1) 尝试将 ddr 固定最高频率；
- (2) 将屏的消隐期加长，提高一行的取数时间；

7.3.2 iommu 出错

确认是否如下图所示的 iommu pagefault 错误，如果有，请先尝试更新到最新代码测试，如果最新代码还存在该问题，请提交 remine 系统，并附上相关 log；

```
rockchip-vop ff8f0000.vop: [drm:vop_isr] *ERROR* POST_BUF_EMPTY irq err
rk_iommu ff8f3f00.iommu: Page fault at 0x00000000f0000400 of type read
rk_iommu ff8f3f00.iommu: iova = 0x00000000f0000400: dte_index: 0x3c0 pte_index: 0x0
rk_iommu ff8f3f00.iommu: mmu_dte_addr: 0x00000000f1692000 dte@0x00000000f1692f00: 0x
: 0 page@0x0000000000000000 flags: 0x0

0x00000000: 00000000 03058896 20805800 0003d000
0x00000010: 0000000f 0800780c 00000000 00711c08
0x00000020: ed000000 00000000 00000000 00000000
0x00000030: 3a001085 00400000 00000000 01400147
0x00000040: 00082000 00000000 0101028d 0101028d
```

7.3.3 logic 电压太低

logic 电压太低会导致 VOP 异常，可以尝试提高 100mv 测试；

7.3.4 AFBDC/IFBDC 对齐要求

对于 px30/rk3326、rk3368、rk3399 平台如果屏的分辨率非 16pixel 对齐可以尝试关闭 afbdc/ifbdc 功能，修改方法参考文档《FAQ-DRM-HWC》 1.3.1 章节。

7.4 drm 驱动 bind

如果看到下面 log 红色部分说明 drm 驱动已经正常加载成功：

```

rockchip-dp ff970000.edp: using lookup tables for GPIO lookup
rockchip-dp ff970000.edp: lookup for GPIO hpd failed
i2c i2c-9: of_i2c: modalias failure on /edp@ff970000/ports
rockchip-drm display-subsystem: bound ff970000.edp (ops 0xffffffff8008ae9840)
rockchip-drm display-subsystem: bound ff960000.dsi (ops 0xffffffff8008ae8808)
i2c i2c-10: of_i2c: modalias failure on /hdmi@ff940000/ports
dwhdmi-rockchip ff940000.hdmi: registered DesignWare HDMI I2C bus driver
dwhdmi-rockchip ff940000.hdmi: Detected HDMI TX controller v2.11a with HDCP
rockchip-drm display-subsystem: bound ff940000.hdmi (ops 0xffffffff8008ae71a0)
[drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[drm] No driver support for vblank timestamp query.
mmc_host mmc2: Bus speed (slot 0) = 150000000Hz (slot req 150000000Hz, actual
rockchip-drm display-subsystem: fb0: frame buffer device
of_get_named_gpiod_flags: can't parse 'simple-audio-card,hp-det-gpio' proper
of_get_named_gpiod_flags: can't parse 'simple-audio-card,mic-det-gpio' proper
asoc-simple-card hdmi-sound: i2s-hifi <-> ff8a0000.i2s mapping ok

```

在驱动加载过程中，可能会出现一些显示设备加载失败，那可能是因为显示驱动依赖的一些资源没准备好导致，如果最后能看到上面的 log 我们就认为 drm 驱动已经正常加载。

如果一直出现 drm 驱动加载失败 log，产品上比较可能出现的是 GPIO 口被其他设备先注册或者 panel 的 compatible 未正确配置导致 panel 注册失败。

7.5 uboot logo 要求

(1) uboot logo 和内核 logo 的大小一直；

(2) 支持 8bit rle8 压缩的 bmp 图片(生成方式: convert -compress rle -colors 256 logo_old.bmp logo_new.bmp)和 24bit 非压缩的 bmp 图片；

7.6 部分机器开机动画显示异常，桌面显示正常

(1) 异常 logo

verified-boot 对固件校验时发现校验失败才会重新调用屏端的接口，对应的 log 为：

init: Starting service 'exec 1 (/sbin/slideshow)'.

(2) 原因

system 校验失败,可能是 system 分区太小导致结尾校验 hash 信息丢失或者烧写工具，会导致如下调用关系：

```
draw()@system/extras/slideshow/slideshow.cpp
```

```
->drm_flip()bootable/recovery/minui/graphics_drm.cpp
```

```
->drmModePageFlip()
```

导致在一些屏上显示黑屏

(3) 处理方法

可以通过关闭这个加密临时处理，真正解决需要确认烧写工具的版本，具体关闭方法如下：

```
a/fstab.rk30board.bootmode.forceencrypt.emmc
```

```
b/fstab.rk30board.bootmode.forceencrypt.emmc
```

```
-/dev/block/platform/fe330000.sdhci/by-name/system /system ext4
```

```
ro,noatime,nodiratime,noauto_da_alloc
```

```
wait,resize,verify=/dev/block/platform/fe330000.sdhci/by-name/verity_mode
```

```
+ /dev/block/platform/fe330000.sdhci/by-name/system /system ext4
```

```
ro,noatime,nodiratime,noauto_da_alloc wait,resize # use this line below instead to enable verity
```

7.7 显示效果调节

RK 平台支持亮度，对比度，饱和度的调节，从 linux 4.4 开始我们默认使用 BCSH 模块来调节效果，可以使用下面命令调节，默认值为 50：

```
setprop persist.sys.brightness.main 50;setprop persist.sys.contrast.main 50;setprop
persist.sys.hue.main 50;setprop persist.sys.saturation.main 50;
setprop sys.display.timeline N -->N 的值每次调节加 1
```

7.8 屏无法点亮/休眠唤醒显示异常/不显示问题

按以下几个方面做进一步确认：

- (1) 确认是否有背光；
- (2) 确认屏的相关电源及复位控制是否正常；
- (3) 确认上下电时序是否满足屏的 spec 要求；

7.9 rk3308 显示特殊性

7.9.1 uboot 显示

RK3308 默认关闭 uboot 显示，如果要打开 uboot 显示需要参考如下配置 uboot 中的 defconfig

```
--- a/configs/evb-rk3308_defconfig
+++ b/configs/evb-rk3308_defconfig
@@ -4,7 +4,6 @@ CONFIG_SYS_MALLOC_F_LEN=0x2000
CONFIG_ROCKCHIP_RK3308=y
CONFIG_ROCKCHIP_SPL_RESERVE_IRAM=0x0
CONFIG_RKIMG_BOOTLOADER=y
-# CONFIG_USING_KERNEL_DTB is not set
CONFIG_TARGET_EVB_RK3308=y
CONFIG_DEFAULT_DEVICE_TREE="rk3308-evb"
CONFIG_DEBUG_UART=y
@@ -55,6 +54,11 @@ CONFIG_USB_GADGET_DOWNLOAD=y
CONFIG_G_DNL_MANUFACTURER="Rockchip"
CONFIG_G_DNL_VENDOR_NUM=0x2207
CONFIG_G_DNL_PRODUCT_NUM=0x330d
+CONFIG_DM_VIDEO=y
+CONFIG_DISPLAY=y
+CONFIG_DRM_ROCKCHIP=y
+CONFIG_DRM_ROCKCHIP_RGB=y
+CONFIG_LCD=y
CONFIG_USE_TINY_PRINTF=y
CONFIG_SPL_TINY_MEMSET=y
CONFIG_ERRNO_STR=y
```

7.9.2 CMA_SIZE 的修改

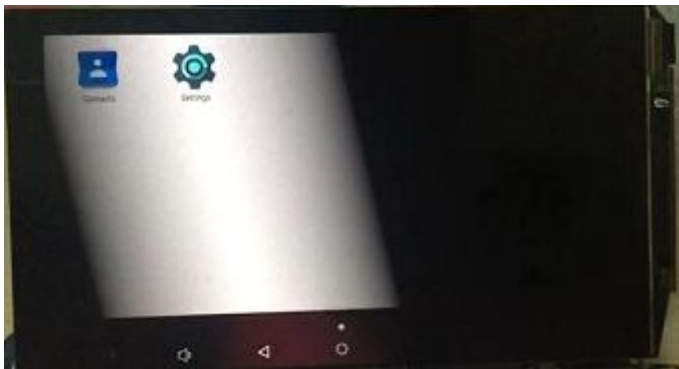
RK3308 VOP 不支持 IOMMU，所以分配内存需要从预留的 cma buffer 分配，默认 CMA_SIZE 为 16M，如果出现分配内存失败，可以参考如下方法修改 CMA_SIZE：


```
--- a/arch/arm64/configs/rk3308_linux_defconfig
+++ b/arch/arm64/configs/rk3308_linux_defconfig
@@ -44,6 +44,7 @@ CONFIG_HZ_1000=y
 # CONFIG_COMPACTION is not set
 # CONFIG_BOUNCE is not set
 CONFIG_DEFAULT_MMAP_MIN_ADDR=32768
+CONFIG_CMA=y
 # CONFIG_UNMAP_KERNEL_AT_EL0 is not set
 # CONFIG_ARM64_HW_AFDBM is not set
 # CONFIG_ARM64_PAN is not set
@@ -89,6 +90,8 @@ CONFIG_RFKILL=y
 CONFIG_DEVTMPFS=y
 CONFIG_DEVTMPFS_MOUNT=y
 # CONFIG_ALLOW_DEV_COREDUMP is not set
+CONFIG_DMA_CMA=y
+CONFIG_CMA_SIZE_MBYTES=32
 # CONFIG_BLK_DEV is not set
 CONFIG_NETDEVICES=y
 # CONFIG_NET_CORE is not set
```

7.10 OLED 屏余晖确认

传统的 LCD 屏是 16ms 全余晖，而 OLED 一大优势就是低余晖，这可以给 VR 产品带来更舒适的体验。

一般 OLED 屏可以调整余晖值，在没有高速 DV 的情况下可以使用 iPhone 手机的慢动作模式拍摄屏幕，根据亮度区间和全屏幕的比例关系初步估算出余晖时间，如下面这张图，如果屏按 60fps 扫描，刷一帧需要的时间是 16.6ms，这个时候亮部区域差不多占全屏区域一半，这样可以估算余晖时间大概为 8ms 左右。



7.11 各种接口屏配置

请参考文档《Rockchip_DRM_Panel_Porting_Guide》。

8 参考文档

- (1) Rockchip_drm_integration_helper-zh.pdf
- (2) Rockchip_DRM_Panel_Porting_Guide.pdf
- (3) Rockchip rk fb development guide.pdf
- (4) https://en.wikipedia.org/wiki/Direct_Rendering_Manager
- (5) <https://01.org/linuxgraphics/gfx-docs/drm/gpu/index.html>
- (6) <https://kernel.readthedocs.io/en/latest/gpu/drm-kms.html#mode-setting>
- (7) <https://events.static.linuxfound.org/sites/events/files/slides/brezillon-drm-kms.pdf>