

密级状态：绝密() 秘密() 内部() 公开(☒)

Rockchip_DRM_Panel_Porting_Guide

(技术部，第二系统产品部)

文件状态： [] 正在修改 [<input checked="" type="checkbox"/>] 正式发布	当前版本：	V1.5
	作 者：	闭伟勇
	完成日期：	2018-8-30
	审 核：	
	完成日期：	

福州瑞芯微电子股份有限公司

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
V1.0	闭伟勇	2017-4-15	初始版本	
V1.1	黄家钗	2017-4-17	加入 LVDS 屏配置说明	
V1.2	闭伟勇	2017-8-15	同步代码，更新 MIPI 章节。	
V1.3	闭伟勇	2017-12-9	更新文档，增加 Display Routing&FAQ 章节。	
V1.4	黄家钗	2018-7-9	加入 MCU/CPU 屏配置说明	
V1.5	闭伟勇	2018-8-30	更新 RGB 配置说明	

目 录

1	Documentation and Source Code.....	1
1.1	kernel.....	1
1.2	u-boot	2
1.3	Example	3
2	MIPI-DSI	3
2.1	DT Bindings	3
2.1.1	MIPI-DSI	3
2.1.2	MIPI-DPHY	4
2.1.3	Logo	4
2.1.4	Panel.....	5
2.2	Command	7
3	eDP.....	12
3.1	DT Bindings	12
3.1.1	eDP Host	12
3.1.2	eDP PHY	13
3.1.3	Logo	13
3.1.4	Panel.....	14
4	LVDS/RGB	15
4.1	DT Bindings	15
4.1.1	LVDS.....	15
4.1.2	RGB	15
4.1.3	Logo	17
4.1.4	Panel.....	18
4.2	Data mapping	22
5	DP	26

5.1	DT Bindings	26
5.1.1	DP_TX	26
6	MCU/CPU.....	26
6.1	Interface	26
6.2	Panel.....	27
6.3	MCU timing	28
7	Display Routing	28
7.1	DT Bindings	28
7.2	Dual-Display	30
7.3	Note.....	31
8	FAQ.....	31

1 Documentation and Source Code

1.1 kernel

Source Code Dir:

drivers/gpu/drm/rockchip/

drivers/gpu/drm/bridge/

drivers/gpu/drm/panel/

drivers/phy/rockchip/

Documentation Dir:

Documentation/devicetree/bindings/display/rockchip/

Documentation/devicetree/bindings/display/bridge/

Documentation/devicetree/bindings/display/panel/

Documentation/devicetree/bindings/phy/rockchip/

Driver	File	Doc
Core	rockchip_drm_drv.c	rockchip-drm.txt
VOP	rockchip_drm_vop.c rockchip_vop_reg.c	rockchip-vop.txt
LVDS	rockchip_lvds.c	rockchip-lvds.txt
RGB	rockchip_rgb.c	rockchip-rgb.txt
MIPI-DSI	dw-mipi-dsi.c phy-rockchip-inno-mipi-dphy.c	dw_mipi_dsi_rockchip.txt phy-rockchip-inno-mipi-dphy.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
DP	cdn-dp-core.c cdn-dp-reg.c cdn-dp-link-training.c phy-rockchip-typec.c	cdn-dp-rockchip.txt phy-rockchip-typec.txt

Panel	panel-simple.c	simple-panel.txt
-------	----------------	------------------

1.2 u-boot

Source Code Dir:

[rkdevelop]: drivers/video/rockchip*

[next-dev]: drivers/video/drm/*

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 rockchip-inno-mipi-dphy.c
Panel	panel_simple.c (rkdevelop) rockchip_dsi_panel.c (rkdevelop) rockchip_panel.c (next-dev)
LVDS	rockchip_lvds.c
RGB	rockchip_rgb.c

1.3 Example

只要在源代码以下目录搜索关键字，就可以找到现成的参考配置文件：

arch/arm64/boot/dts/rockchip/

arch/arm/boot/dts/rk*

关键字：panel/edp/dsi/lvds/vop/rgb/cdn_dp

2 MIPI-DSI

2.1 DT Bindings

2.1.1 MIPI-DSI

(1) Single-channel

```
&dsi {
    status = "okay";
};
```

Note: 在 rk3288/rk3399 平台上，对于单通道 MIPI 屏，硬件上请接到 mipi_tx0。

(2) Dual-channel (for rk3288/rk3399)

```
&dsi {
    status = "okay";
    rockchip,dual-channel = <&dsi1>;
};
```

```
&dsi1 {
    status = "okay";
};
```

(3) Dual-link (for rk3399)

```
&dsi {
    status = "okay";
};
```

```
&dsi1 {
    status = "okay";
};
```

① 属性说明

Property	Value	Comment
rockchip, lane-rate (optional)	80~1000	如果没有配置该属性，驱动会根据 display_mode 自动计算 lane-rate。单位为 mbps/lane。
rockchip, dual-channel (optional)	&dsi1	对于 Dual-channel mode，该属性必须配置。

2.1.2 MIPI-DPHY

```
&mipi_dphy {
    status = "okay";
};
```

Note: rk3288/rk3399 没有该节点，不需要配置。

① 属性说明

Property	Value	Comment
inno, lanes	1~4	如果没有配置该属性，默认值为 4lanes。

2.1.3 Logo

```
&route_dsi {
    status = "okay";
};
```

Note: 如果没有开启 u-boot 阶段 Logo，只有 Android 启动才会显示 Android Logo，不会显示内核 Logo，所以一般需要开启。

2.1.4 Panel

```
&dsi {
    status = "okay";

    panel@0 {
        compatible = "simple-panel-dsi";
        reg = <0>;
        power-supply = <&vcc_lcd>;
        backlight = <&backlight>;
        reset-gpios = <&gpio3 13 GPIO_ACTIVE_LOW>;
        enable-gpios = <&gpio2 27 GPIO_ACTIVE_HIGH>;

        dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
                     MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET)>;
        dsi,format = <MIPI_DSI_FMT_RGB888>;
        dsi,lanes = <4>;

        prepare-delay-ms = <20>;
        reset-delay-ms = <20>;
        init-delay-ms = <20>;
        enable-delay-ms = <20>;
        disable-delay-ms = <20>;
        unprepare-delay-ms = <20>;

        panel-init-sequence = [
            39 00 04 b9 ff 83 94
            ...
            15 00 02 df 8e
            ...
            05 78 01 11
            05 14 01 29
        ];

        panel-exit-sequence = [
            05 00 01 28
            05 78 01 10
        ];
    };
};
```

```

display-timings {
    native-mode = <&timing0>;

    timing0: timing0 {
        clock-frequency = <74000000>;
        hactive = <800>;
        hfront-porch = <68>;
        hsync-len = <18>;
        hback-porch = <68>;
        vactive = <1280>;
        vfront-porch = <6>;
        vsync-len = <4>;
        vback-porch = <6>;
        hsync-active = <0>;
        vsync-active = <0>;
        de-active = <0>;
        pixelclk-active = <0>;
    };
};
};

```

① 属性说明

Property	Value	Comment
compatible	simple-panel-dsi	默认，不需要修改。
reg	0	virtual channel，默认，不需要修改。
backlight	&backlight	默认，不需要修改。
power-supply	&vcc_lcd	可选，regulator 配置。
reset-gpios	&gpio0 21 GPIO_ACTIVE_LOW	可选，reset 脚 GPIO 配置。
enable-gpios	&gpio0 22 GPIO_ACTIVE_HIGH	可选，enable 脚 GPIO 配置。
dsi,flags	(MIPI_DSI_MODE_VIDEO MIPI_DSI_MODE_VIDEO_BURST MIPI_DSI_MODE_EOT_PACKET MIPI_DSI_MODE_LPM)	默认，不需要修改。 MIPI_DSI_MODE_VIDEO, MIPI_DSI_MODE_VIDEO_BURST, 表示 Video Burst Mode。 MIPI_DSI_MODE_LPM 表示在 LP 模式下 发送初始化序列。 MIPI_DSI_MODE_EOT_PACKET 表示关闭 EOTP 特性，当前驱动默认关闭。

dsi,format	MIPI_DSI_FMT_RGB888	Pixel Format
dsi,lanes	4	Lane Number (1 ~ 8), 大于 4 表示为 Dual-channel MIPI-DSI Panel。
prepare-delay-ms	20	可选，具体时序参考屏驱动。
reset-delay-ms	20	可选，具体时序参考屏驱动。
init-delay-ms	20	可选，具体时序参考屏驱动。
enable-delay-ms	20	可选，具体时序参考屏驱动。
unprepare-delay-ms	20	可选，具体时序参考屏驱动。
disable-delay-ms	20	可选，具体时序参考屏驱动。
panel-init-sequence	...	屏的上电初始化序列，具体参数配置方式参考下文说明。
panel-exit-sequence	...	屏的下电初始化序列，具体参数配置方式参考下文说明。
display-timings	...	Display-Timing，参考屏规格书。

Note: 除了以上部分所列 Property，还有其他可选的 Property 可以配置，具体参考内核相关文档或者屏驱动的代码。

2.2 Command

```
panel-init-sequence = [
    39 00 04 b9 ff 83 94
    ...
    15 00 02 df 8e
    ...
    05 78 01 11
    05 14 01 29
];
```

说明：前 3 个字节（16 进制），分别代表 Data Type，Delay，Payload Length。

从第四个字节开始的数据代表长度为 Length 的实际有效 Payload。

第一条命令的解析如下：

39 00 04 b9 ff 83 94

Data Type: 0x39 (DCS Long Write)

Delay: 0x00 (0 ms)

Payload Length: 0x04 (4 Bytes)

Payload: 0xb9 0xff 0x83 0x94

最后一条命令的解析如下:

05 14 01 29

Data Type: 0x05 (DCS Short Write, no parameters)

Delay: 0x14 (20 ms)

Payload Length: 0x01 (1 Bytes)

Payload: 0x29

1) Data Type

Table 16 Data Types for Processor-sourced Packets

Data Type, hex	Data Type, binary	Description	Packet Size
0x01	00 0001	Sync Event, V Sync Start	Short
0x11	01 0001	Sync Event, V Sync End	Short
0x21	10 0001	Sync Event, H Sync Start	Short
0x31	11 0001	Sync Event, H Sync End	Short
0x08	00 1000	End of Transmission packet (EoTp)	Short
0x02	00 0010	Color Mode (CM) Off Command	Short
0x12	01 0010	Color Mode (CM) On Command	Short
0x22	10 0010	Shut Down Peripheral Command	Short
0x32	11 0010	Turn On Peripheral Command	Short
0x03	00 0011	Generic Short WRITE, no parameters	Short
0x13	01 0011	Generic Short WRITE, 1 parameter	Short
0x23	10 0011	Generic Short WRITE, 2 parameters	Short
0x04	00 0100	Generic READ, no parameters	Short
0x14	01 0100	Generic READ, 1 parameter	Short
0x24	10 0100	Generic READ, 2 parameters	Short
0x05	00 0101	DCS Short WRITE, no parameters	Short
0x15	01 0101	DCS Short WRITE, 1 parameter	Short
0x06	00 0110	DCS READ, no parameters	Short
0x37	11 0111	Set Maximum Return Packet Size	Short
0x09	00 1001	Null Packet, no data	Long
0x19	01 1001	Blanking Packet, no data	Long
0x29	10 1001	Generic Long Write	Long
0x39	11 1001	DCS Long Write/write_LUT Command Packet	Long
0x0C	00 1100	Loosely Packed Pixel Stream, 20-bit YCbCr, 4:2:2 Format	Long
0x1C	01 1100	Packed Pixel Stream, 24-bit YCbCr, 4:2:2 Format	Long
0x2C	10 1100	Packed Pixel Stream, 16-bit YCbCr, 4:2:2 Format	Long
0x0D	00 1101	Packed Pixel Stream, 30-bit RGB, 10-10-10 Format	Long
0x1D	01 1101	Packed Pixel Stream, 36-bit RGB, 12-12-12 Format	Long

Data Type, hex	Data Type, binary	Description	Packet Size
0x3D	11 1101	Packed Pixel Stream, 12-bit YCbCr, 4:2:0 Format	Long
0x0E	00 1110	Packed Pixel Stream, 16-bit RGB, 5-6-5 Format	Long
0x1E	01 1110	Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x2E	10 1110	Loosely Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x3E	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	Long
0xX0 and 0xFF, unspecified	XX 0000 XX 1111	DO NOT USE All unspecified codes are reserved	

① DCS Write

0x05	00 0101	DCS Short WRITE, no parameters	Short
0x15	01 0101	DCS Short WRITE, 1 parameter	Short
0x39	11 1001	DCS Long Write/write_LUT Command Packet	Long

DCS packet 包括一个字节的 dcs 命令，以及 n 个字节的 parameters。

如果 $n < 2$ ，将以 Short Packet 的形式对 Payload 进行打包。 $n = 0$ ，表示只发送 dcs 命令，不带参数，Data Type 为 0x05； $n = 1$ ，表示发送 dcs 命令，带一个参数，Data Type 为 0x15。

如果 $n \geq 2$ ，将以 Long Packet 的形式对 Payload 进行打包。此时发送 dcs 命令，带 n 个参数，Data Type 为 0x39。

② Generic Write

0x03	00 0011	Generic Short WRITE, no parameters	Short
0x13	01 0011	Generic Short WRITE, 1 parameter	Short
0x23	10 0011	Generic Short WRITE, 2 parameters	Short
0x29	10 1001	Generic Long Write	Long

Generic Packet 包括 n 个字节的 parameters。

如果 $n < 3$ ，将以 Short Packet 的形式对 Payload 进行打包。 $n = 0$ ，表示 no parameters，Data Type 为 0x03； $n = 1$ ，表示 1 parameter，Data Type 为 0x13； $n = 2$ ，表示 2 parameters，Data Type 为 0x23。

如果 $n \geq 3$ ，将以 Long Packet 的形式进行对 Payload 打包，表示 n parameters，Data Type 为 0x29。

2) Delay

表示当前 Packet 发送完成之后，需要延时多少 ms，再开始发送下一条命令。

3) Payload Length

表示 Packet 的有效负载长度。

4) Payload

表示 Packet 的有效负载，长度为 Payload Length。

5) Example

(B) On sequence

sequence	DataType (hex)	index (hex)	parameters # (hex)	description	comment
SLEEP MODE					
↓					
DCDC_EN L->H				DCDC_EN L->H (VSP,VSN on)	
wait 20ms					
command	05	01	-	soft reset	
wait 5ms					
command	23	B0	1 00	MCAP	
command	29	B3	1 04	Interface setting	
			2 08		
			3 00		
			4 22		
			5 00		
command	29	B4	1 0C	Interface ID setting	
command	29	B6	1 3A	DSI control	
			2 D3		
command	15	51	1 E6	write display brightness	
command	15	53	1 2C	write control display	
command	15	3A	1 77	set pixel format	
command	39	2A	1 00	set column address	
			2 00		
			3 04		
			4 AF		
command	39	2B	1 00	set page address	
			2 00		
			3 07		
			4 7F		
send image	39	2C/3C		write memory / write memory continue	
command	05	11	-	exit sleep mode	
wait 120ms					
command	05	29	-	set display on	
wait min 0ms					
LED_EN L->H				LED_EN L->H	
↓					
NORMAL MODE					

```

panel-init-sequence = [
    05 05 00 01 01
    23 00 00 02 b0 00
    23 00 00 02 d6 01
    29 00 00 06 b3 14 08 00 22 00
    29 00 00 02 b4 0c
    29 00 00 03 b6 3a c3
    15 00 00 02 51 e6
    15 00 00 02 53 2c
    15 00 00 02 3a 77
    39 00 00 05 2a 00 00 04 af
    39 00 00 05 2b 00 00 07 7f
    05 78 00 01 29
    05 00 00 01 11
];

```

(C) Off sequence

sequence	DataTyp (hex)	index (hex)	parameters # (hex)	description	comment
NORMAL MODE					
↓					
command	05	28	- -	set display off	
wait 20ms					
command	05	10	- -	enter sleep mode	
wait 80ms					
DCDC_EN H->L				DCDC_EN H->L (VSP,VSN off)	
wait 20ms					
↓					
SLEEP MODE					

```
panel-exit-sequence = [
    05 14 00 01 28
    05 50 00 01 10
];
```

3 eDP

3.1 DT Bindings

3.1.1 eDP Host

```
&edp {
    force-hpd;
    status = "okay";

    ports {
        port@1 {
            reg = <1>;

            edp_out_panel: endpoint {
                remote-endpoint = <&panel_in_edp>;
            };
        };
    };
};
```

① 属性说明

Property	Value	Comment
force-hpd		一般需要配置该属性，避免因 hpd 检测不到，导致 eDP 无法正常工作。

3.1.2 eDP PHY

```
&edp_phy {  
    status = "okay";  
};
```

Note: 只有 rk3288 需要配置。

3.1.3 Logo

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

Note: 如果没有开启 u-boot 阶段 Logo，只有 Android 启动才会显示 Android Logo，不会显示内核 Logo，所以一般需要开启。

3.1.4 Panel

```
panel: panel {
    compatible = "simple-panel";
    backlight = <&backlight>;
    power-supply = <&vcc_lcd>;
    enable-gpios = <&gpio1 13 GPIO_ACTIVE_HIGH>;
    prepare-delay-ms = <20>;
    enable-delay-ms = <20>;
    bpc = <8>;

    display-timings {
        native-mode = <&timing0>;

        timing0: timing0 {
            clock-frequency = <200000000>;
            hactive = <1536>;
            vactive = <2048>;
            hfront-porch = <12>;
            hsync-len = <16>;
            hback-porch = <48>;
            vfront-porch = <8>;
            vsync-len = <4>;
            vback-porch = <8>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };

    ports {
        panel_in: endpoint {
            remote-endpoint = <&edp_out>;
        };
    };
};
```

① 属性说明

Property	Value	Comment
bpc	6 or 8 or 10	Bit pixel component

4 LVDS/RGB

4.1 DT Bindings

4.1.1 LVDS

```
&lvds {
    status = "okay";

    ports {
        port@1 {
            reg = <1>;

            lvds_out_panel: endpoint {
                remote-endpoint = <&panel_in_lvds>;
            };
        };
    };
};
```

4.1.2 RGB

Note: 在 rockchip 平台中，RGB 接口主要有两种设计：

LCDC_M0: 所有 pin 由 iomux 配置复用，因此需要使用 rgb 驱动，并且配置 pinctrl。

LCDC_M1: 一部分 pin 由 lvds_phy 配置复用，因此需要使用 lvds 驱动。一部分 pin 由 iomux 配置复用，因此需要配置 pinctrl。

4.1.2.1 LCDC_M0

```
&rgb {
    pinctrl-names = "default";
    pinctrl-0 = <&lcdc_rgb_dclk_pin
                &lcdc_rgb_m0_den_pin
                &lcdc_rgb_m0_hsync_pin
                &lcdc_rgb_m0_vsync_pin
                &lcdc_rgb888_m0_data_pins>;
    status = "okay";

    ports {
        port@1 {
            reg = <1>;

            rgb_out_panel: endpoint {
                remote-endpoint = <&panel_in_rgb>;
            };
        };
    };
};
```

4.1.2.2 LCDC_M1

```
&lvds {
    pinctrl-names = "default";
    pinctrl-0 = <&lcdc_rgb_dclk_pin &lcdc_rgb666_m1_data_pins>;
    status = "okay";

    ports {
        port@1 {
            reg = <1>;

            lvds_out_panel: endpoint {
                remote-endpoint = <&panel_in_lvds>;
            };
        };
    };
};
```

4.1.3 Logo

4.1.3.1 LCDC_M0

```
&route_rgb {  
    status = "okay";  
};
```

4.1.3.2 LCDC_M1/LVDS

```
&route_lvds {  
    status = "okay";  
};
```

4.1.4 Panel

4.1.4.1 LVDS

```
panel {
    compatible = "simple-panel";
    power-supply = <&vcc33_lcd>;
    backlight = <&backlight>;
    enable-gpios = <&gpio0 RK_PB6 GPIO_ACTIVE_HIGH>;
    prepare-delay-ms = <100>;
    unprepare-delay-ms = <100>;
    enable-delay-ms = <100>;
    disable-delay-ms = <100>;
    rockchip,data-mapping = "vesa";
    rockchip,data-width = <24>;
    rockchip,output = "lvds";
    status = "okay";

    display-timings {
        native-mode = <&timing0>;

        timing0: timing0 {
            clock-frequency = <48000000>;
            hactive = <1024>;
            vactive = <600>;
            hback-porch = <90>;
            hfront-porch = <90>;
            vback-porch = <10>;
            vfront-porch = <10>;
            hsync-len = <90>;
            vsync-len = <10>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };

    port {
        panel_in_lvds: endpoint {
            remote-endpoint = <&lvds_out_panel>;
        };
    };
};
```

① 属性说明

Property	Value	Comment
rockchip,data-mapping	vesa or jeida	LVDS 信号的两种编码方式，具体对应关系参考 data mapping 说明。
rockchip,data-width	18 or 24 or 30	LVDS 的数据位，RGB 三个分量都是 6bit 的填 18,RGB 三个分量都是 8bit 的填 24, RGB 三个分量都是 10bit 的填 30。
rockchip,output	lvds or dualldvs	双通道 LVDS，目前只有 rk3288 支持。

4.1.4.2 LCDC_M0

```
panel {
    compatible = "simple-panel";
    backlight = <&backlight>;
    power-supply = <&vcc3v3_lcd>;
    bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;
    enable-delay-ms = <20>;
    prepare-delay-ms = <20>;
    unprepare-delay-ms = <20>;
    disable-delay-ms = <20>;

    width-mm = <217>;
    height-mm = <136>;

    display-timings {
        native-mode = <&timing0>;

        timing1: timing1 {
            clock-frequency = <49500000>;
            hactive = <1024>;
            vactive = <600>;
            hback-porch = <90>;
            hfront-porch = <90>;
            vback-porch = <10>;
            vfront-porch = <10>;
            hsync-len = <90>;
            vsync-len = <10>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };

    port {
        panel_in_rgb: endpoint {
            remote-endpoint = <&rgb_out_panel>;
        };
    };
};
```

① 属性说明

Property	Value	Comment
bus-format	MEDIA_BUS_FMT_RGB888_1X24	接口类型，

	MEDIA_BUS_FMT_RGB666_1X24_CPADHI MEDIA_BUS_FMT_RGB666_1X18 MEDIA_BUS_FMT_RGB565_1X16	RGB888/RGB666/RGB565, 其中 RGB666 有两种子类型。
--	--	--

4.1.4.3 LCDC_M1

```

panel {
    compatible = "simple-panel";
    enable-gpios = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
    bus-format = <MEDIA_BUS_FMT_RGB666_1X18>;
    rockchip,data-width = <18>;
    rockchip,output = "rgb";

    display-timings {
        native-mode = <&timing0>;

        timing0: timing0 {
            clock-frequency = <51200000>;
            hactive = <1024>;
            vactive = <600>;
            hback-porch = <100>;
            hfront-porch = <120>;
            vback-porch = <10>;
            vfront-porch = <15>;
            hsync-len = <100>;
            vsync-len = <10>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };

    port {
        panel_in_lvds: endpoint {
            remote-endpoint = <&lvds_out_panel>;
        };
    };
};

```

① 属性说明

Property	Value	Comment
----------	-------	---------

bus-format	<p>MEDIA_BUS_FMT_RGB888_1X24</p> <p>MEDIA_BUS_FMT_RGB666_1X24_CPADHI</p> <p>MEDIA_BUS_FMT_RGB666_1X18</p> <p>MEDIA_BUS_FMT_RGB565_1X16</p>	<p>总线类型，</p> <p>RGB888/RGB666/RGB565，</p> <p>其中 RGB666 有两种子类型。</p>
rockchip,data-width	18 or 24 or 30	
rockchip,output	rgb	<p>Lvds 驱动会根据输出类型，将</p> <p>phy 配置成 rgb 模式。</p>

4.2 Data mapping

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。

		VESA_6BIT	JEIDA_6BIT
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	GND	GND
	TX5	GND	GND
	TX10	GND	GND
	TX11	GND	GND
	TX16	GND	GND
	TX17	GND	GND
	TX23	RSVD	RSVD

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	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

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。

		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	VSYNC	VSYNC
	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 DP

5.1 DT Bindings

5.1.1 DP_TX

```
&cdn_dp {
    status = "okay";
    extcon = <&fusb0>;
    phys = <&tcphy0_dp>;
};
```

6 MCU/CPU

6.1 Interface

MCU 屏又叫 CPU 屏基于 i80 总线协议，具有 CS、RS、RD、WR 四根控制信号线和 8/16/18/24 数据线，MCU 屏的优点是控制简单，无需同步信号和时钟信号，缺点是屏内部需要集成 GRAM，成本较高，无法做到大分辨率的屏。目前 rk3188、rk3308、rk3326/px30 支持 MCU 屏。

控制信号：

CS: 屏的片选信号，低有效，和 VSYNC 复用。

RS: 数据和命令区分信号，1 表示发送数据，0 表示发送命令，和 DCLK 复用。

RD: 1 表示发数据到屏，0 表示从屏读数据(RK 平台不支持)，和 DEN 复用。

WR: 写使能信号，上升沿有效，和 HSYNC 复用。

6.2 Panel

```
panel: panel {
    compatible = "simple-panel";
    bus-format = <MEDIA_BUS_FMT_RGB565_1X16>;
    backlight = <&backlight>;
    enable-gpios = <&gpio0 RK_PB7 GPIO_ACTIVE_LOW>;
    enable-delay-ms = <20>;
    reset-gpios = <&gpio0 RK_PC4 GPIO_ACTIVE_LOW>;
    reset-value = <0>;
    reset-delay-ms = <10>;
    prepare-delay-ms = <20>;
    unprepare-delay-ms = <20>;
    disable-delay-ms = <20>;
    width-mm = <217>;
    height-mm = <136>;
    rgb-mode = "p565";
    status = "okay";
    rockchip,cmd-type = "mcu";

    // type:0 is cmd, 1 is data
    panel-init-sequence = [
        //type delay num val1 val2 val3
        00 00 01 e0
    ];

    panel-exit-sequence = [
        00 78 01 10
    ];

    display-timings {
        native-mode = <&kd050fwfba002_timing>;

        kd050fwfba002_timing: timing0 {
            clock-frequency = <32000000>;
            hactive = <320>;
        }
    }
}
```

特殊属性说明

Property	Value	Comment
rgb-mode	p888、p666、p565、s888、 s888_dummy	屏的数据接口类型
rockchip,cmd-type	spi or mcu	spi: 通过 spi 接口发送初始化命令 mcu: 通过 mcu 接口发送初始化命令

6.3 MCU timing

```
&vop {
    status = "okay";

    mcu-timing {
        mcu-pix-total = <9>;
        mcu-cs-pst = <1>;
        mcu-cs-pend = <8>;
        mcu-rw-pst = <2>;
        mcu-rw-pend = <5>;

        mcu-hold-mode = <0>;
    };
};
```

1) mcu-pix-total: 发送一次数据/命令需要几个 DCLK 周期;

2) mcu-cs-pst/mcu-cs-pend: 片选开始和结束位置;

3) mcu-rw-pst/mcu-rw-pend: 数据发送开始和结束位置;

时序图:

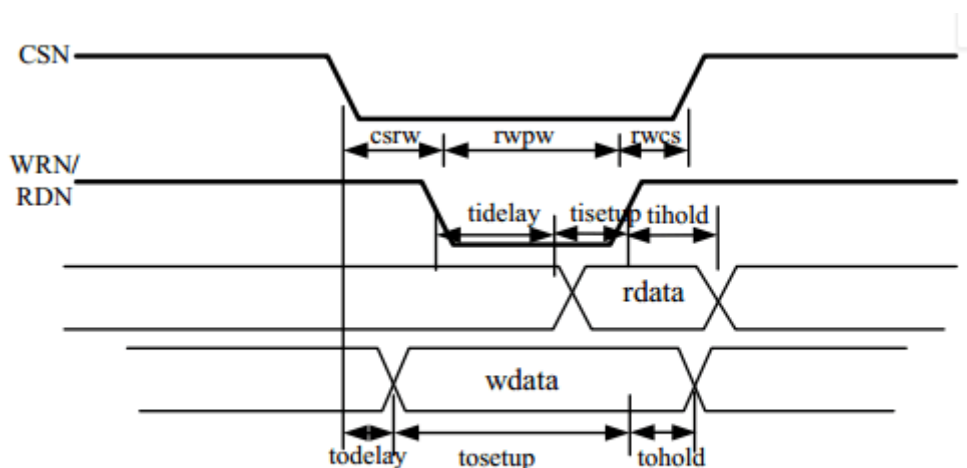


Figure 3-1 i8080 r/w timing

7 Display Routing

对于有两个 vop 的平台 (rk3288/rk3399), 可以灵活选择显示路径。

7.1 DT Bindings

下面以 rk3399 平台 VOPB→eDP→Panel 为例进行说明。

(1) VOPB → eDP


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

&edp_in_vopb {
    status = "okay";
};
```

```
&route_edp {
    connect = <&vopb_out_edp>;
    status = "okay";
};
```

(2) eDP → Panel

```
&edp {
    ports {
        port@1 {
            reg = <1>;

            edp_out_panel: endpoint {
                remote-endpoint = <&panel_in_edp>;
            };
        };
    };
};
```

```
panel {
    port {
        panel_in_edp: endpoint {
            remote-endpoint = <&edp_out_panel>;
        };
    };
};
```

下面以 rk3399 平台 VOPL→DSI→Panel 为例进行说明。

(1) VOPL → DSI

```
&dsi_in_vopl {
    status = "okay";
};

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

```
&route_dsi {
    status = "okay";
    connect = <&vopl_out_dsi>;
};
```

(2) DSI → Panel

因为 Panel 已经挂在 DSI 总线下面，所以绑定关系也就确定了，不需要再做配置。

7.2 Dual-Display

对于有两个 vop 的平台（rk3288/rk3399），同时支持两个显示设备。只要在 dts 中正确配置显示路径，VOP 选择互不冲突，就可以实现。比如按照 5.1 和 5.2 进行配置，就可以实现 eDP Panel 和 DSI Panel 的双显。

目前最新的多显方案，build.prop 需做以下参数设置：

1、edp/dsi + hdmi:

默认配置即可

2、edp + dsi:

sys.hwc.device.primary=eDP

sys.hwc.device.extend=DSI

以上为 edp 为主显，dsi 为副显，如需 dsi 做主显，调换一下即可。

3、edp + dsi/hdmi:

sys.hwc.device.primary=eDP

sys.hwc.device.extend=HDMI-A,DSI

hdmi 插拔会抢占副显的显示。

4、edp/hdmi + dsi:

sys.hwc.device.primary=HDMI-A,eDP

sys.hwc.device.extend=DSI

hdmi 插拔会抢占主显的显示。

7.3 Note

1、如果需要配置 VOPL→eDP/DSI→Panel，上层还需要关掉 AFBC。

```
wlq@loita:~/rk3399/1_source_code/5_RK3399_Android_7.1_release/hardware/rockchip/libgralloc$
diff --git a/Android.mk b/Android.mk
index 2ea9765..937dfe6 100644
--- a/Android.mk
+++ b/Android.mk
@@ -161,7 +161,7 @@ @@ endif

ifeq ($(strip $(TARGET_BOARD_PLATFORM_GPU)), mali-t860)
MALI_AFBC_GRALLOC := 1
-USE_AFBC_LAYER = 1
+USE_AFBC_LAYER = 0
LOCAL_CFLAGS += -DMALI_PRODUCT_ID_T86X=1
endif
```

8 FAQ

1、如何确认显示驱动已经正常加载？

```
px5:/ # dmesg | grep drm
[ 0.000000] Reserved memory: failed to reserve memory for node 'drm-logo@00000000': base 0x0000000000000000, size 0 MiB
[ 0.954629] platform vpu_service: allocator is drm
[ 0.955398] platform hevc_service: allocator is drm
[ 0.964883] [drm:drm_core_init] Initialized drm 1.1.0 20060810
[ 0.969040] rockchip-drm display-subsystem: bound ff930000.vop (ops vop_component_ops)
[ 0.969108] [drm:rockchip_lvds bind] *ERROR* failed to find panel and bridge node
[ 0.969129] rockchip-drm display-subsystem: failed to bind ff968000.lvds (ops rockchip_lvds_component_ops): -517
[ 0.969400] rockchip-drm display-subsystem: master bind failed: -517
[ 4.885955] rockchip-drm display-subsystem: bound ff930000.vop (ops vop_component_ops)
[ 4.886031] [drm:rockchip_lvds bind] *ERROR* failed to find panel and bridge node
[ 4.886044] rockchip-drm display-subsystem: failed to bind ff968000.lvds (ops rockchip_lvds_component_ops): -517
[ 4.886222] rockchip-drm display-subsystem: master bind failed: -517
[ 5.401052] rockchip-drm display-subsystem: bound ff930000.vop (ops vop_component_ops)
[ 5.401185] rockchip-drm display-subsystem: bound ff968000.lvds (ops rockchip_lvds_component_ops)
[ 5.401194] [drm:drm_vblank_init] Supports vblank timestamp caching Rev 2 (21.10.2013).
[ 5.401200] [drm:drm_vblank_init] No driver support for vblank timestamp query.
[ 5.401305] rockchip-drm display-subsystem: failed to parse loader memory
[ 5.404415] rockchip-drm display-subsystem: fb0: frame buffer device
[ 5.453879] [drm:drm_get_platform_dev] Initialized pvr 1.8.4610191 20110701 on minor 1
```

可以通过 Log 确认，只有配置正确，相关 component 最后才会 bound 成功。

（可能会多次因为驱动资源暂时获取不到而 EPROBE_DEFER，但只要配置正确，待相关组件驱动能够完整获取到资源后，最终就会 bind 成功）

2、如果显示驱动无法正常加载，如何排查？（现象为一直打印 drm 相关的 log）

检查 dts 显示相关配置。重点排查 panel 相关资源是否与其他模块冲突，导致 panel 驱动因为资源冲突而无法正常加载（可以导出 log，搜索关键字 **panel**），从而导致整个显示驱动无法正常加载。

（一般为 panel 节点下的 GPIO，或者 backlight 节点下的 GPIO 与其他模块配置冲突）。

3、如何查看当前显示信息？

```
# cat /d/dri/0/summary
```

```
rk3399_box:/ # cat /d/dri/0/summary
VOP [ff900000.vop]: DISABLED
VOP [ff8f0000.vop]: ACTIVE
Connector: DP
    bus_format[0] output_mode[0]
Display mode: 1360x768p60
    clk[86000] real_clk[86000] type[48] flag[5]
    H: 1360 1424 1536 1792
    V: 768 771 777 795
win0-0: ACTIVE
    format: AB24 little-endian (0x34324241)
    zpos: 0
    src: pos[0x0] rect[1920x1080]
    dst: pos[0x0] rect[1360x768]
    buf[0]: addr: 0x00000000001eb0000 pitch: 7680 offset: 0
win2-0: DISABLED
win2-0: DISABLED
win2-1: DISABLED
win2-2: DISABLED
```

4、如果已经根据文档配置完成相关配置，显示驱动也已经正常加载，但是屏不亮，如何排查？

如果按照文档正确配置，屏应该可以正常显示。如果屏不亮，除非驱动有功能缺陷，否则就应该往屏方向排查。

- 1) 确认背光是否正常。
- 2) 确认屏相关供电及复位是否正常。
- 3) 根据规格书检查 panel 的上电时序。

如果是 DSI Panel，可以先粗略的做下面配置：

增加时序相关延时参数：

```
prepare-delay-ms = <120>;
reset-delay-ms = <120>;
init-delay-ms = <120>;
enable-delay-ms = <120>;
disable-delay-ms = <120>;
unprepare-delay-ms = <120>;
```

```
panel-init-sequence = [
    ...
    05 78 01 11
    05 ff 01 29
];
```

在最后一条 cmd 发送之后，延时 0xff (255) ms，表示发送完所有 cmd，到开始传输图像数据的延时。

- 4) &dsi (注意不是在 panel 下面配置) 下配置 rockchip, lane-rate 属性值, 指定所需 lane-rate (一般小于等于 1000), 不通过驱动自动计算 (自动计算出来的值不一定满足需求)。
- 5) 尝试配置修改 timing, 比如水平和垂直方向的消隐值、像素时钟值, 帧率小于等于 60Hz。