RK3588 MIPI DSI-2 Developer's Guide

文件标识：RK-KF-YF-305

发布版本：V1.0.0

日期：2022-01-07

文件密级：□绝密 □秘密 □内部资料 ■公开

免责声明

商标声明

瑞芯微电子股份有限公司

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：　+86-4007-700-590

客户服务传真：　+86-591-83951833

客户服务邮箱：　fae@rock-chips.com

前言

文本主要对 RK3588 MIPI DSI2 各种场景应用进行软件配置说明。

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

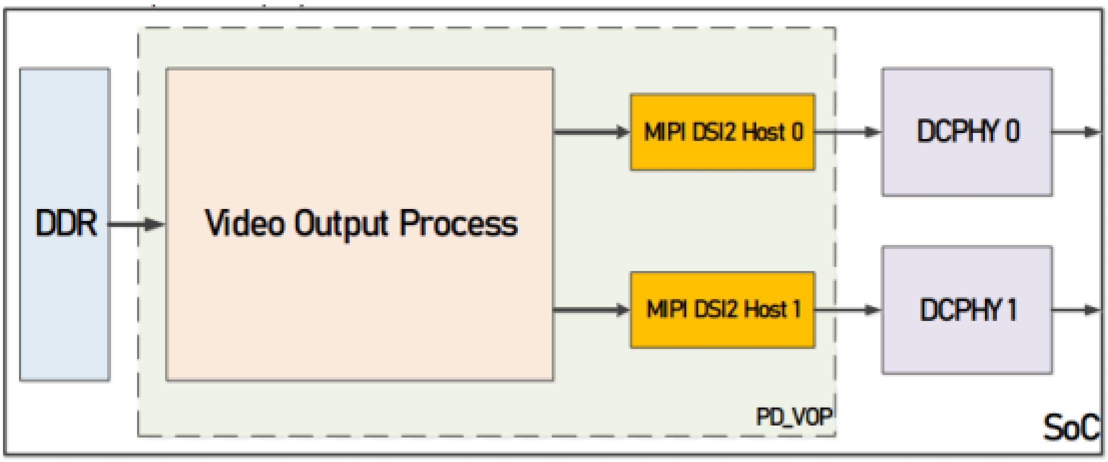
修订记录

版本号	作者	修改日期	修改说明
V1.0.0	黄国椿	2022-01-07	初始发布

目录

Introduction

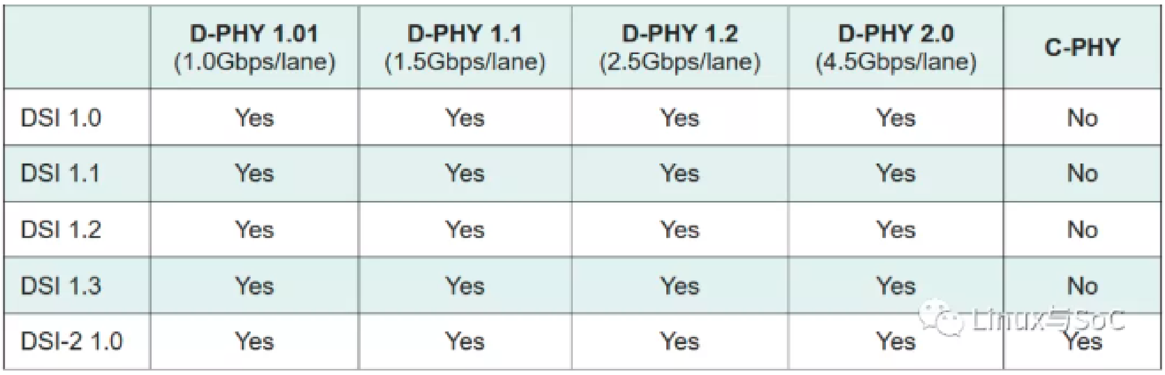
DSI-2 是 MIPI 联盟定义的一组通信协议， 可以兼容 D-PHY 和 C-PHY 的物理接口，RK3588 有两个 DSI-2
控制器和两个物理的 DCPHY, 可以同时最多支持两路 MIPI 输出。



DSI2 Host Controller In SoC

MIPI DSI-2 与 MIPI DSI 的差别

MIPI DSI-2 不同 MIPI DSI, 既可以像MIPI DSI 兼容 MIPI D-PHY,还可以兼容 MIPI C-PHY.



	D-PHY 1.01 (1.0Gbps/lane)	D-PHY 1.1 (1.5Gbps/lane)	D-PHY 1.2 (2.5Gbps/lane)	D-PHY 2.0 (4.5Gbps/lane)	C-PHY
DSI 1.0	Yes	Yes	Yes	Yes	No
DSI 1.1	Yes	Yes	Yes	Yes	No
DSI 1.2	Yes	Yes	Yes	Yes	No
DSI 1.3	Yes	Yes	Yes	Yes	No
DSI-2 1.0	Yes	Yes	Yes	Yes	Yes

MIPI DSI兼容性

Features

MIPI® Alliance Specification for Display Serial Interface 2 (DSI-2) Version 1.1
MIPI® Alliance Specification for Display Command Set (DCS) Version 1.4
MIPI® Alliance Specification for D-PHY v2.0
MIPI® Alliance Specification for C-PHY v1.1
Up to 4.5 Gbps per lane in D-PHY
Up to 2.0 Gsps per trio in C-PHY

应用领域

MIPI DSI基于差分信号传输，可以降低引脚数量和硬件设计复杂度，并保持良好的硬件兼容性。另外，基于MIPI DSI协议的IP还具备低功耗、低EMI的特性。

其应用领域主要包括：

超高清设备
嵌入式显示设备
智能仪表
IOT
头戴设备等

MIPI DSI应用领域

驱动代码说明：

uboot

驱动位置

```
drivers/video/drm/dw_mipi_dsi2.c
drivers/video/drm/samsung_mipi_dcphy.c
```

驱动配置

```
CONFIG_DRM_ROCKCHIP_DW_MIPI_DSI2=y
CONFIG_DRM_ROCKCHIP_SAMSUNG_MIPI_DCPHY=y
```

kernel

驱动位置

```
MIPI DSI-2 host controller:
drivers/gpu/drm/rockchip/dw-mipi-dsi2-rockchip.c

MIPI DCPHY:
drivers/phy/rockchip/phy-rockchip-samsung-dcphy.c
```

驱动配置

```
CONFIG_ROCKCHIP_DW_MIPI_DSI=y
CONFIG_PHY_ROCKCHIP_SAMSUNG_DCPHY=y
```

参考设备树

DTS 路径：

```
arch/arm64/boot/dts/rockchip/rk3588-evb.dtsi
arch/arm64/boot/dts/rockchip/rk3588-evb1-lp4.dtsi
arch/arm64/boot/dts/rockchip/rk3588-evb2-lp4.dtsi
arch/arm64/boot/dts/rockchip/rk3588-evb3-lp5.dtsi
arch/arm64/boot/dts/rockchip/rk3588-evb4-lp4.dtsi
arch/arm64/boot/dts/rockchip/rk3588s-evb.dtsi
arch/arm64/boot/dts/rockchip/rk3588s-evb1-lp4x.dtsi
arch/arm64/boot/dts/rockchip/rk3588s-evb2-lp5.dtsi
arch/arm64/boot/dts/rockchip/rk3588s-evb4-lp4x.dtsi
```

dts 配置用例场景说明:

```
rk3588-evb1: dsi0->dphy->1080p_panel && dsi1->dphy->1080p_panel;
rk3588-evb2: dsi1->dphy->1080p_panel;
rk3588-evb3: dsi0->dphy->1080p_panel && dsi1->cphy->cphy_panel;
rk3588-evb4: dsi0->dphy->1080p_panel;
rk3588s-evb1: dsi0->dphy->1080p_panel && dsi1->dphy->cmd_no_dsc_panel;
rk3588s-evb2: dsi0->cphy->cphy_panel & dsi1->dphy->1080p_panel;
rk3588s-evb4: dsi0->dphy->1080p_panel && dsi1->dphy->cmd_dsc_panel;
```

屏端配置

DTS 配置

```
dsi0_panel: panel@0 {
    status = "okay";
    compatible = "simple-panel-dsi";
    reg = <0>;
    power-supply = <&vcc3v3_lcd_n>;
    backlight = <&backlight>;
    reset-gpios = <&gpio2 RK_PB4 GPIO_ACTIVE_LOW>;
    reset-delay-ms = <10>;
    enable-delay-ms = <10>;
    prepare-delay-ms = <10>;
    unprepare-delay-ms = <10>;
    disable-delay-ms = <60>;
    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>;
    //phy-c-option;
    //compressed-data;
    //slice-width = <720>;
    //slice-height = <65>;
    //version-major = <1>;
    //version-minor = <1>;

    panel-init-sequence = [
        ...
        05 78 01 11
        05 00 01 29
    ];

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

    disp_timings0: display-timings {
        native-mode = <&dsi0_timing0>;
        dsi0_timing0: timing0 {
            clock-frequency = <132000000>;
            hactive = <1080>;
            vactive = <1920>;
            hfront-porch = <15>;
            hsync-len = <4>;
            hback-porch = <30>;
            vfront-porch = <15>;
            vsync-len = <2>;
            vback-porch = <15>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };
};
```

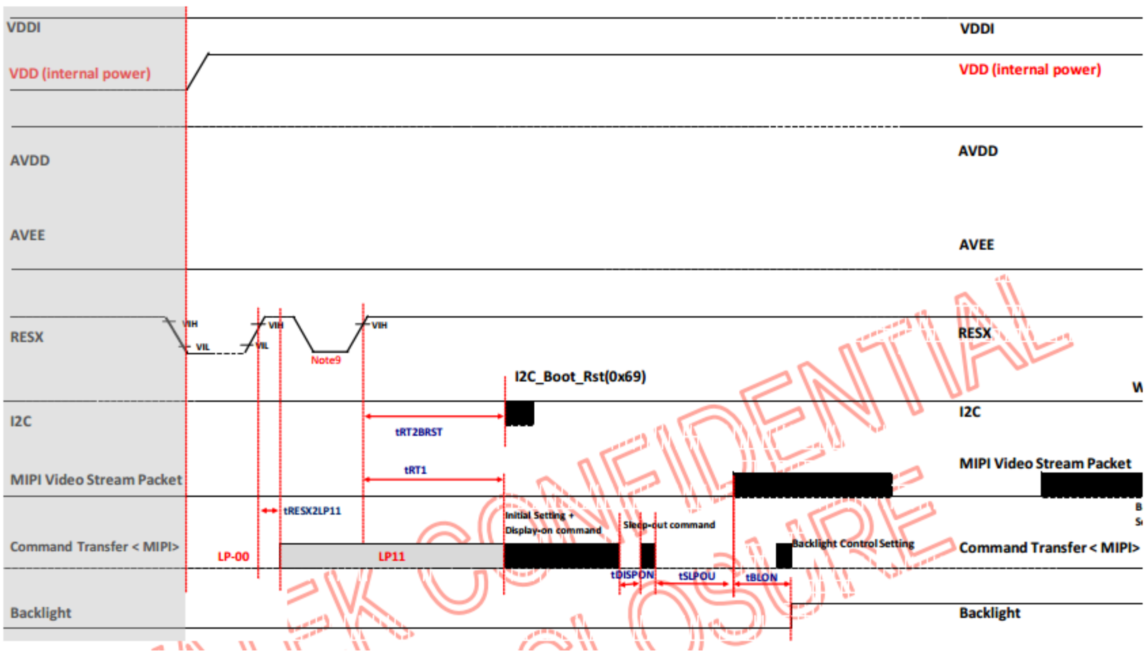
配置说明

通用配置

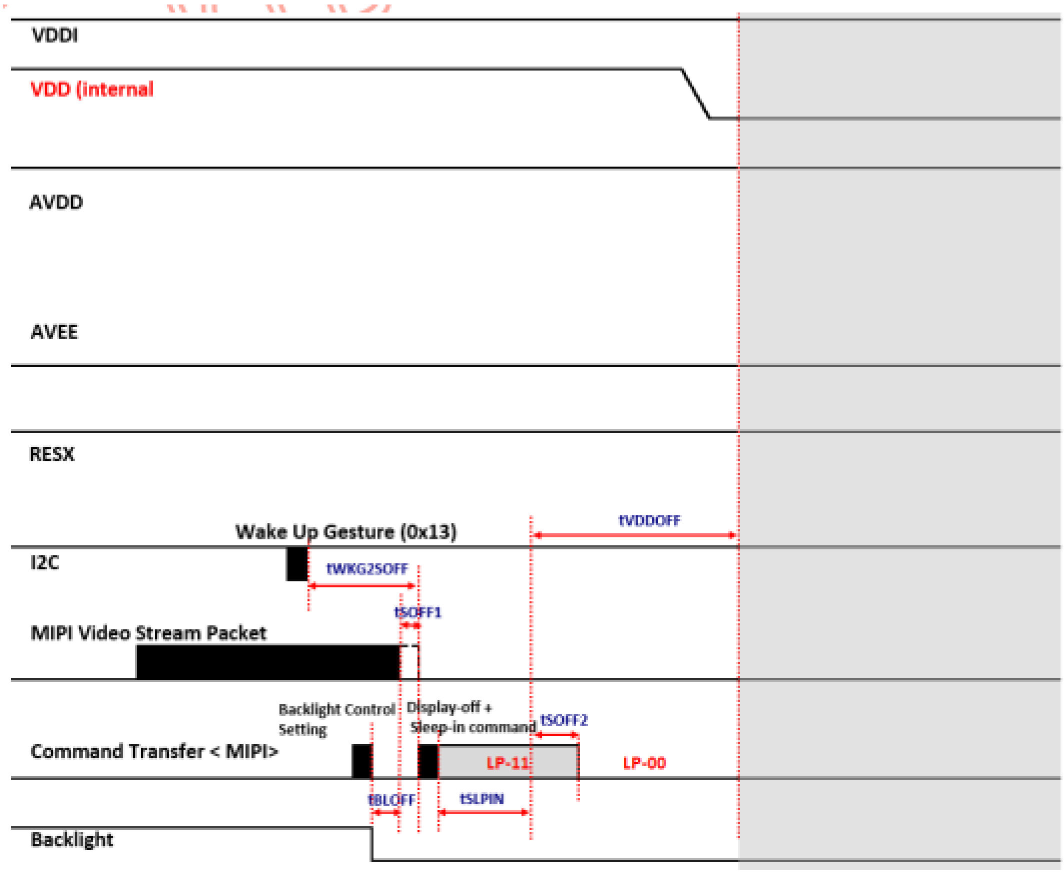
Property	Description	Value
compatible	compatible	simple-panel-dsi
power-supply	屏端供电 [option]	相关regulator引用
backlight	背光	背光引用
enable-gpios	屏使能GPIO [option]	GPIO引用描述
reset-gpios	屏复位GPIO	GPIO引用描述
reset-delay-ms	panel sequence delay	参考 panel spec
enable-delay-ms		
prepare-delay-ms		
unprepare-delay-ms		
disable-delay-ms		
dsi,flags	DSI2 工作模式	cmd mode: MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET
		video mode: MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST | MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET
dsi,format	像素数据格式	MIPI_DSI_FMT_RGB888
		MIPI_DSI_FMT_RGB666
		MIPI_DSI_FMT_RGB666_PACKED
		MIPI_DSI_FMT_RGB565
dsi,lanes	mipi data 通道数	1/2/3 trios [cphy]
		6 trios [cphy 双通道]
		1/2/3/4 lanes [dphy]
		8 lanes [dphy 双通道]
phy-c-option	C-PHY panel [option]	布尔类型string
compressed-data	带dscpanel [option]	布尔类型string
slice-width	定义dsc slice宽 [option]	参照panel spec
slice-height	定义dsc slice高 [option]	
version-major	定义dsc版本 [option]	参照panel spec

version-minor		
panel-init-sequence	屏上电初始化序列	[hex] data_type delay_ms payload_lenth payload
panel-exit-sequence	屏下电初始化序列	
display-timing	panel timing	参考panel spec

屏上电时序



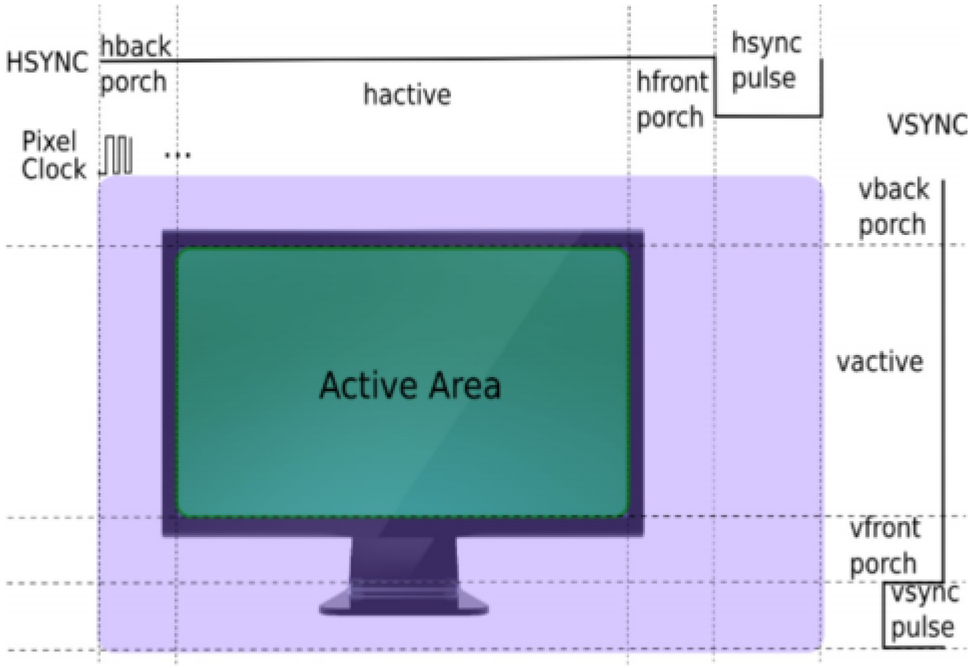
屏下电时序



初始化序列常见数据类型

data type	description	packet size
0x03	Generic Short WRITE, no parameters	short
0x13	Generic Short WRITE, 1 parameters	short
0x23	Generic Short WRITE, 2 parameters	short
0x29	Generic long WRITE,	long
0x05	DCS Short WRITE, no parameters	short
0x15	DCS Short WRITE, 1 parameters	short
0x07	DCS Short WRITE, 1 parameters, DSC EN	short
0x0a	DCS long WRITE, PPS, 128 bytes	long

Display Timing

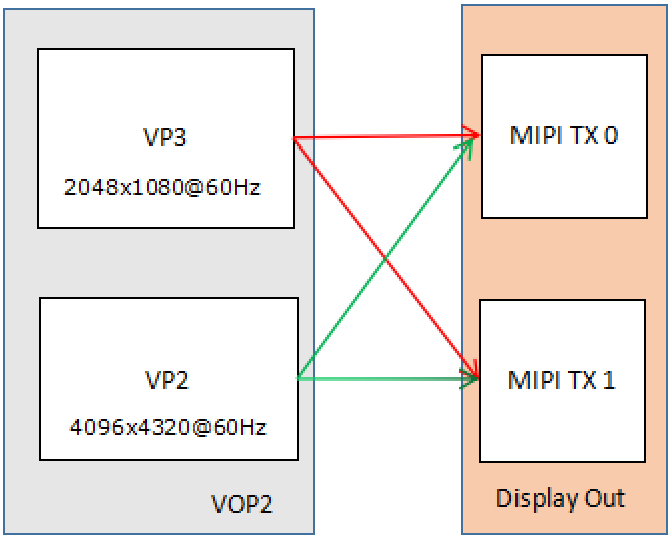


Display Timing

```
htotal = hactive+hfront-porch+hsync-len+hback-porch;
vtotal = vactive+vfront-porch+vsync-len+vback-porch;
clock-frequency = htotal x vtotal x frame_rate (KHz)
```

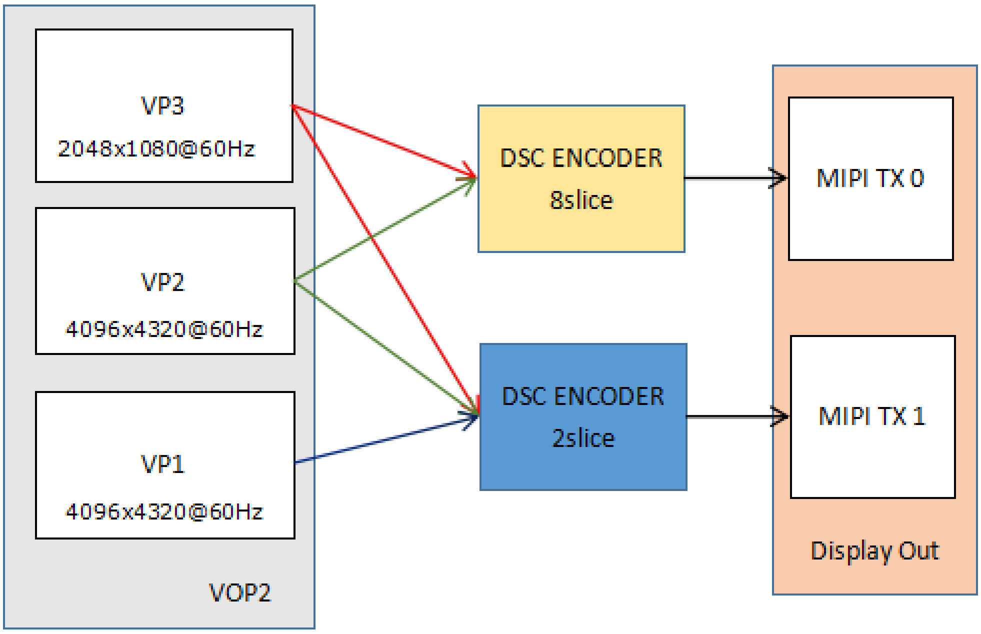
Display Route

Without DSC



Schematic Diagram of Display Output Interface

With DSC



Schematic Diagram of Display Output Interface

DTS 配置

如 DSI0 挂载在 VP3:

```
&dsi0_in_vp2 {
    status = "disabled";
};

&dsi0_in_vp3 {
    status = "okay";
};
```

如 DSI1 挂载 VP2

```
&dsi1_in_vp2 {
    status = "okay";
};

&dsi1_in_vp3 {
    status = "disabled";
};
```

开机LOGO

DSI0

例如 vp3->dsi0 或 vp3->dsc0->dsi0:

```
&route_dsi0 {
    status = "okay";
    connect = <&vp3_out_dsi0>;
};
```

DSI1

例如 vp2->dsi1 或 vp2->dsc1->dsi1:

```
&route_dsi1 {
    status = "okay";
    connect = <&vp2_out_dsi1>;
};
```

DSI0 && DSI1

例如 （vp3->dsi0 或 vp3->dsc0->dsi0） && （vp2->dsi1 或 vp2->dsc1->dsi1）：

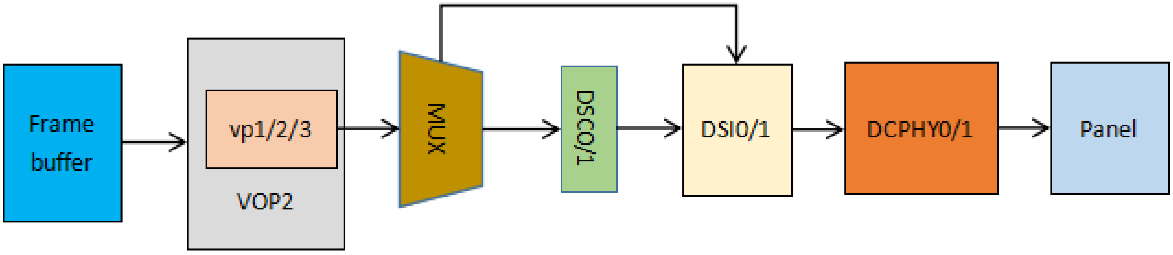
```
&route_dsi0 {
    status = "okay";
    connect = <&vp3_out_dsi0>;
};

&route_dsi1 {
    status = "okay";
    connect = <&vp2_out_dsi1>;
};
```

DSI HOST

ports
以下实例中 ports 是用来 Dispaly Interface 和 panel 之间进行关联.
详细配置说明参阅如下文档:

```
Documentation/devicetree/bindings/graph.txt
```

单DSI



单 DSI Display Route

DSI0

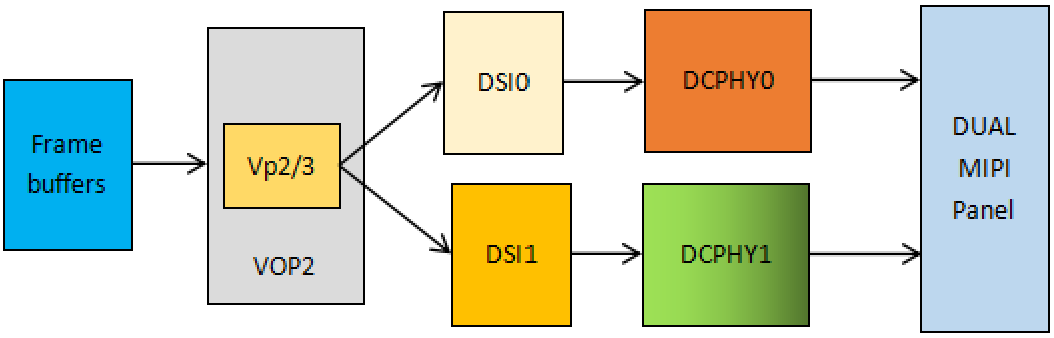
```
&dsi0 {
    status = "okay";
    //rockchip,lane-rate = <1000>;
    dsi0_panel: panel@0 {
        status = "okay";
        compatible = "simple-panel-dsi";
        ...

        ports {
            #address-cells = <1>;
            #size-cells = <0>;

            port@0 {
                reg = <0>;
                panel_in_dsi: endpoint {
                    remote-endpoint = <&dsi_out_panel>;
                };
            };
        };
    };

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@1 {
            reg = <1>;
            dsi_out_panel: endpoint {
                remote-endpoint = <&panel_in_dsi>;
            };
        };
    };

};

&mipi_dcphy0 {
    status = "okay";
};
```

单DSI1

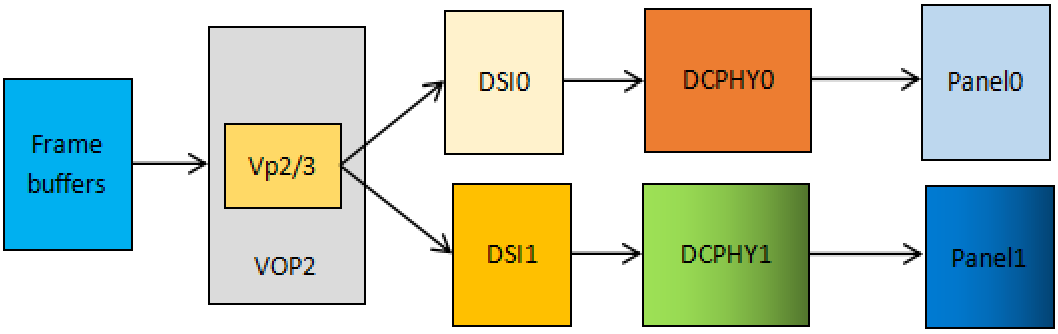
```
&dsi1 {
    status = "okay";
    //rockchip,lane-rate = <1000>;
    dsi1_panel: panel@0 {
        status = "okay";
        compatible = "simple-panel-dsi";
        ...

        ports {
            #address-cells = <1>;
            #size-cells = <0>;

            port@0 {
                reg = <0>;
                panel_in_dsi1: endpoint {
                    remote-endpoint = <&dsi1_out_panel>;
                };
            };
        };
    };

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@1 {
            reg = <1>;
            dsi1_out_panel: endpoint {
                remote-endpoint = <&panel_in_dsi1>;
            };
        };
    };
};

&mipi_dcphy1 {
    status = "okay";
};
```

双通道 DSI

MODE1:



MODE2:



Dual Channel DSI Display Route

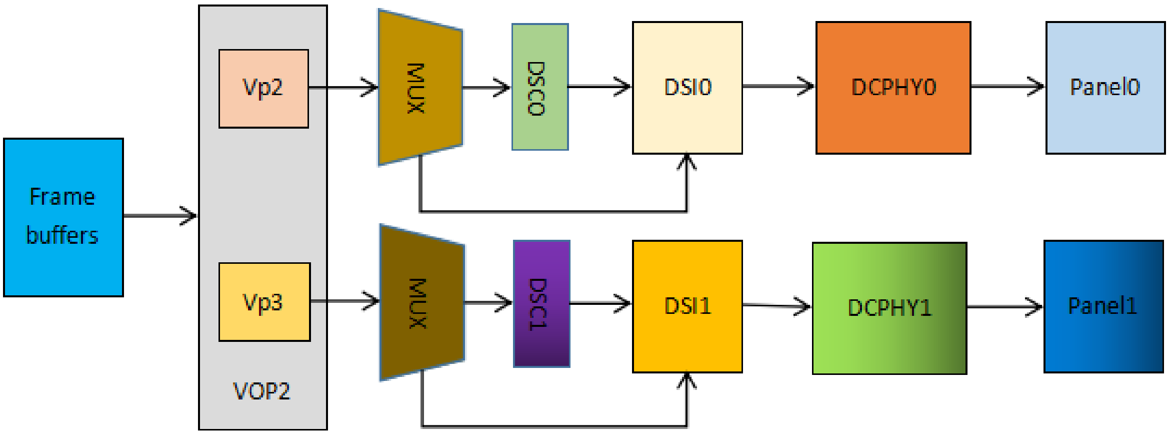
双通道的配置注意如下标红属性：

rockchip,dual-channel = <&dsi1>

dsi,lanes = <8>;//DPHY 屏

```
&dsi0 {
    status = "okay";
    rockchip,dual-channel = <&dsi1>;

    dsi0_panel {
        status = "okay";
        compatible = "simple-panel-dsi";
        dsi,lanes  = <8>;
        ...

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

            timing0: timing0 {
                clock-frequency = <260000000>;
                hactive = <1440>;
                vactive = <2560>;
                hfront-porch = <150>;
                hsync-len = <30>;
                hback-porch = <60>;
                vfront-porch = <8>;
                vsync-len = <4>;
                vback-porch = <4>;
                hsync-active = <0>;
                vsync-active = <0>;
                de-active = <0>;
                pixelclk-active = <0>;
            };
        };

        ports {
            #address-cells = <1>;
            #size-cells = <0>;

            port@0 {
                reg = <0>;
                panel_in_dsi0: endpoint {
                    remote-endpoint = <&dsi0_out_panel>;
                };
            };
        };
    };

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@1 {
            reg = <1>;
            dsi0_out_panel: endpoint {
                remote-endpoint = <&panel_in_dsi0>;
            };
        };
    };
};

&dsi1 {
    status = "okay";
};

&mipi_dcphy0 {
    status = "okay";
};

&mipi_dcphy1 {
    status = "okay";
};
```

Dual-link DSI



Dual Link DSI Display Route

```
&dsi0 {
    status = "okay";
    //rockchip,lane-rate = <1000>;
    dsi0_panel: panel@0 {
        status = "okay";
        compatible = "simple-panel-dsi";
        ...

        ports {
            #address-cells = <1>;
            #size-cells = <0>;
            port@0 {
                reg = <0>;
                panel_in_dsi: endpoint {
                    remote-endpoint = <&dsi_out_panel>;
                };
            };
        };
    };

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@1 {
            reg = <1>;
            dsi_out_panel: endpoint {
                remote-endpoint = <&panel_in_dsi>;
            };
        };
    };

};

&dsi1 { status = "okay";
    //rockchip,lane-rate = <1000>;
    dsi1_panel: panel@0 {
        status = "okay";
        compatible = "simple-panel-dsi";
        ...

        ports {
            #address-cells = <1>;
            #size-cells = <0>;

            port@0 {
                reg = <0>;
                panel_in_dsi1: endpoint {
                    remote-endpoint = <&dsi1_out_panel>;
                };
            };
        };
    };

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@1 {
            reg = <1>;
            dsi1_out_panel: endpoint {
                remote-endpoint = <&panel_in_dsi1>;
            };
        };
    };
};

&mipi_dcphy0 {
    status = "okay";
```

```
};

&mipi_dcphy1 {
    status = "okay";
};
```

DCPHY

实际应用配置中默认是配置成D-PHY，通过屏端配置介绍可知，通过下面可以配置成 C-PHY:

```
dsi0_panel: panel@0 {
    ...
    phy-c-option;
    ...
};
```

D-PHY

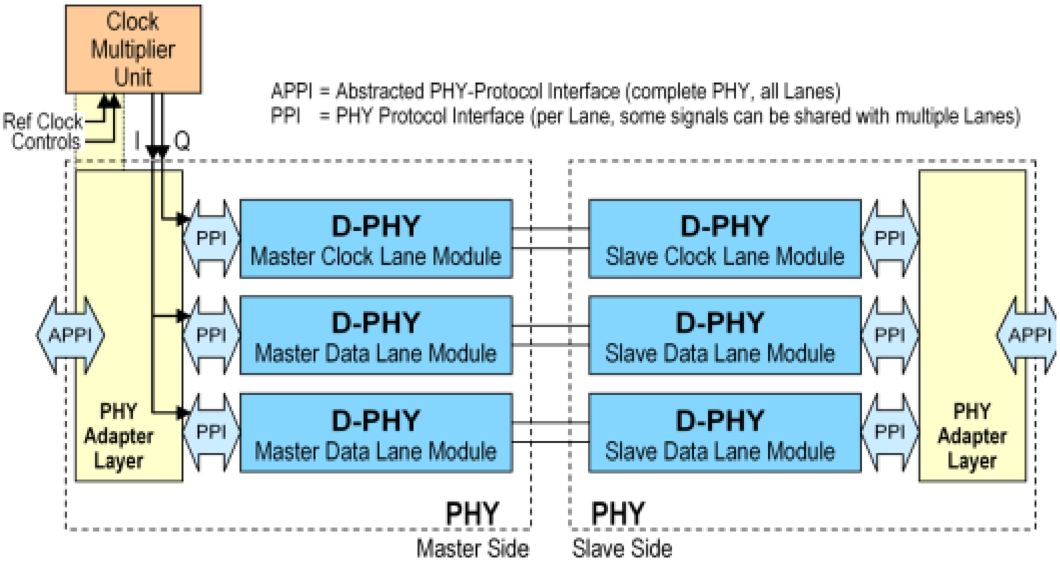
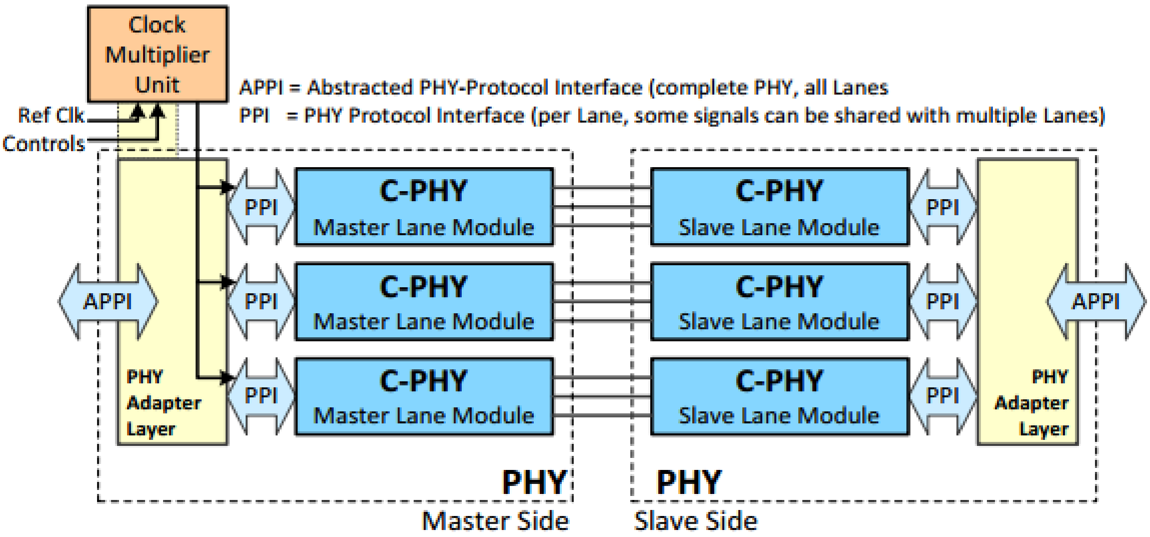


Figure 2 Two Data Lane PHY Configuration

Up to 4.5 Gbps per lane in D-PHY
一个D-PHY port 最多4lanes，每个lane由两条差分线组成

C-PHY



APPI = Abstracted PHY-Protocol Interface (complete PHY, all Lanes
PPI = PHY Protocol Interface (per Lane, some signals can be shared with multiple Lanes)

Figure 5 Three Lane PHY Configuration

Up to 2.0 Gsps per trio in C-PHY
一个C-PHY port 最多3lanes，每个lane由 tree-wire-trios 组成

常见问题

1、查看VOP timing 和 Connector 信息

```
console:/ # cat /d/dri/0/summary
Video Port0: DISABLED
Video Port1: DISABLED
Video Port2: DISABLED
Video Port3: ACTIVE
    Connector: DSI-2
        bus_format[100a]: RGB888_1X24
        overlay_mode[0] output_mode[0] color_space[0]
    Display mode: 1080x1920p60
        clk[132000] real_clk[132000] type[48] flag[a]
        H: 1080 1095 1099 1129
        V: 1920 1935 1937 1952
    Cluster3-win0: ACTIVE
        win_id: 6
        format: AB24 little-endian (0x34324241) SDR[0] color_space[0]
glb_alpha[0xff]
        rotate: xmirror: 0 ymirror: 0 rotate_90: 0 rotate_270: 0
        csc: y2r[0] r2y[0] csc mode[0]
        zpos: 0
        src: pos[0, 0] rect[1080 x 1920]
        dst: pos[0, 0] rect[1080 x 1920]
        buf[0]: addr: 0x000000000376e000 pitch: 4352 offset: 0
```

2、查看 DSI2 相关 clk tree

```
console:/ # cat /d/clk/clk_summary | grep vop
    clk_vop_pmu                       0        0        0    24000000          0
    0  50000
          dclk_vop3                   1        2        0    33000000          0
    0  50000
          dclk_vop1_src               0        1        0   594000000          0
    0  50000
```

```
             dclk_vop1                0        1        0   594000000          0
    0  50000
          dclk_vop0_src               0        1        0   594000000          0
    0  50000
             dclk_vop0                0        1        0   594000000          0
    0  50000
             aclk_vop_low_root        1        1        0   396000000          0
    0  50000
             hclk_vop_root            2        4        0   198000000          0
    0  50000
                hclk_vop              1        3        0   198000000          0
    0  50000
          aclk_vop_root               1        1        0   500000000          0
    0  50000
             aclk_vop_doby            0        0        0   500000000          0
    0  50000
             aclk_vop_sub_src         1        1        0   500000000          0
    0  50000
                aclk_vop              1        4        0   500000000          0
    0  50000
             pclk_vop_root            3        5        0   100000000          0
    0  50000
          dclk_vop2_src               1        1        0   148500000          0
    0  50000
             dclk_vop2                1        2        0   148500000          0
    0  50000
console:/ # cat /d/clk/clk_summary | grep dsi
                pclk_dsihost1         1        2        0   100000000          0
    0  50000
                pclk_dsihost0         1        2        0   100000000          0
    0  50000
    clk_dsihost1                      1        2        0   351000000          0
    0  50000
    clk_dsihost0                      1        2        0   351000000          0
    0  50000
console:/ # cat /d/clk/clk_summary | grep mipi
          mipi1_clk_src               0        0        0    33000000          0
    0  50000
             mipi1_pixclk             0        0        0    33000000          0
    0  50000
          mipi0_clk_src               0        0        0   148500000          0
    0  50000
             mipi0_pixclk             0        0        0   148500000          0
    0  50000
    clk_usbdpphy_mipidcpphy_ref       5        5        0    24000000          0
    0  50000
    clk_mipi_camaraout_m4             0        0        0    24000000          0
    0  50000
    clk_mipi_camaraout_m3             0        0        0    24000000          0
    0  50000
    clk_mipi_camaraout_m2             0        0        0    24000000          0
    0  50000
    clk_mipi_camaraout_m0             0        0        0    24000000          0
    0  50000
          clk_mipi_camaraout_m1       0        0        0    37125000          0
    0  50000
                pclk_mipi_dcphy1       1        1        0    100000000
 0     0  50000
```

```
                pclk_mipi_dcphy0       1         1         0    100000000
 0      0   50000
console:/ #
```

3、如何查看指定 DSI lane 速率
DSI lane 速率的指定有两种，一种是驱动自动计算:

```
dmesg | grep dsi

[   77.369812] dw-mipi-dsi2 fde20000.dsi: [drm:dw_mipi_dsi2_encoder_enable]
final DSI-Link bandwidth: 879 x 4 Mbps
```

一种是通过如下属性手动指定:

```
&dsi0 {
    ...
    rockchip,lane-rate = <1000>;
    ...
}
```

4、MIPI DSI2 HOST 没有自己color bar 功能，通过如下命令可以让 VOP2 投显 color bar
根据显示路由选择对应的命令:

```
vp0:
io -4 0xfdd90c08 0x1 && io -4 0xfdd90000 0xffffffff

vp1:
io -4 0xfdd90d08 0x1 && io -4 0xfdd90000 0xffffffff

vp2:
io -4 0xfdd90e08 0x1 && io -4 0xfdd90000 0xffffffff

vp3:
io -4 0xfdd90f08 0x1 && io -4 0xfdd90000 0xffffffff
```

5、如何判断显示异常的时候，MIPI DSI2 和 panel 是否通信正常

uboot:

```
--- a/drivers/video/drm/rockchip_panel.c
+++ b/drivers/video/drm/rockchip_panel.c
@@ -260,6 +260,7 @@ static void panel_simple_prepare(struct rockchip_panel
*panel)
    struct rockchip_panel_priv *priv = dev_get_priv(panel->dev);
    struct mipi_dsi_device *dsi = dev_get_parent_platdata(panel->dev);
    int ret;
+   u8 mode;

    if (priv->prepared)
        return;
@@ -285,6 +286,8 @@ static void panel_simple_prepare(struct rockchip_panel
*panel)
    if (plat->delay.init)
        mdelay(plat->delay.init);
```

```
+   mipi_dsi_dcs_get_power_mode(dsi, &mode);
+   printf("===>mode: 0x%x\n", mode);
    if (plat->on_cmds) {
        if (priv->cmd_type == CMD_TYPE_SPI)
            ret = rockchip_panel_send_spi_cmds(panel->state,
@@ -298,6 +301,8 @@ static void panel_simple_prepare(struct rockchip_panel *panel)
            printf("failed to send on cmds: %d\n", ret);
    }

+   mipi_dsi_dcs_get_power_mode(dsi, &mode);
+   printf("===>mode: 0x%x\n", mode);
    priv->prepared = true;
 }
```

kernel

```
--- a/drivers/gpu/drm/panel/panel-simple.c
+++ b/drivers/gpu/drm/panel/panel-simple.c
@@ -506,6 +506,7 @@ static int panel_simple_prepare(struct drm_panel *panel)
    unsigned int delay;
    int err;
    int hpd_asserted;
+   u8 mode;

    if (p->prepared)
        return 0;
@@ -554,6 +555,8 @@ static int panel_simple_prepare(struct drm_panel *panel)
        }
    }

+   mipi_dsi_dcs_get_power_mode(p->dsi, &mode);
+   printk("===>mode: 0x%x\n, mode");
    if (p->desc->init_seq)
        if (p->dsi)
            panel_simple_xfer_dsi_cmd_seq(p, p->desc->init_seq);
@@ -561,6 +564,9 @@ static int panel_simple_prepare(struct drm_panel *panel)
    if (p->desc->delay.init)
        msleep(p->desc->delay.init);

+   mipi_dsi_dcs_get_power_mode(p->dsi, &mode);
+   printk("===>mode: 0x%x\n, mode");
+
    p->prepared = true;

    return 0;
```

通信正常会有如下打印,否者排查屏端时序确保屏是否就绪:

```
===> mode: 0x8
===> mode: 0x9c
```

6、backlight驱动probe失败

```
console:/ # dmesg | grep backlight
[    3.164274] pwm-backlight: probe of backlight failed with error -16
```