



# 曳影 1520 视频输出 用户手册

|      |            |
|------|------------|
| 文档版本 | 1.0.0      |
| 保密等级 | 保密         |
| 发布日期 | 2023-08-26 |

**Copyright © 2022 T-HEAD (Shanghai) Semiconductor Co., Ltd. All rights reserved.**

This document is the property of T-HEAD (Shanghai) Semiconductor Co., Ltd. This document may only be distributed to: (i) a T-HEAD party having a legitimate business need for the information contained herein, or (ii) a non-T-HEAD party having a legitimate business need for the information contained herein. No license, expressed or implied, under any patent, copyright or trade secret right is granted or implied by the conveyance of this document. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of T-HEAD (Shanghai) Semiconductor Co., Ltd.

**Trademarks and Permissions**

The T-HEAD Logo and all other trademarks indicated as such herein are trademarks of T-HEAD (Shanghai) Semiconductor Co., Ltd. All other products or service names are the property of their respective owners.

**Notice**

The purchased products, services and features are stipulated by the contract made between T-HEAD and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

**平头哥（上海）半导体技术有限公司 T-HEAD (Shanghai) Semiconductor Co., LTD**

Address: 5th Floor Number 2 Chuan He Road 55, Number 366 Shang Ke Road, Shanghai free trade area, China  
Website: [www.t-head.cn](http://www.t-head.cn)

**Copyright © 2022 平头哥（上海）半导体技术有限公司，保留所有权利。**

本文档的所有权及知识产权归属于平头哥（上海）半导体技术有限公司及其关联公司(下称“平头哥”)。本文档仅能分派给：(i) 拥有合法雇佣关系，并需要本文档的信息的平头哥员工，或(ii)非平头哥组织但拥有合法合作关系，并且其需要本文档的信息的合作方。对于本文档，未经平头哥（上海）半导体技术有限公司明示同意，则不能使用该文档。在未经平头哥（上海）半导体技术有限公司的书面许可的情形下，不得复制本文档的任何部分，传播、转录、储存在检索系统中或翻译成任何语言或计算机语言。

**商标申明**

平头哥的 LOGO 和其它所有商标归平头哥（上海）半导体技术有限公司及其关联公司所有，未经平头哥（上海）半导体技术有限公司的书面同意，任何法律实体不得使用平头哥的商标或者商业标识。

**注意**

您购买的产品、服务或特性等应受平头哥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的购买或使用范围之内。除非合同另有约定，平头哥对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。平头哥（上海）半导体技术有限公司不对任何第三方使用本文档产生的损失承担任何法律责任。

**平头哥（上海）半导体技术有限公司 T-HEAD (Shanghai) Semiconductor Co., LTD**

地址：中国（上海）自由贸易试验区上科路 366 号、川和路 55 弄 2 号 5 层  
网址： [www.t-head.cn](http://www.t-head.cn)

## 版本历史

| 版本     | 说明   | 作者  | 日期         |
|--------|------|-----|------------|
| V1.0.0 | 初始版本 | 平头哥 | 2023-08-26 |

# 目录

|                 |     |
|-----------------|-----|
| 版本历史.....       | I   |
| 目录.....         | II  |
| 图表目录.....       | III |
| 术语与缩略语.....     | IV  |
| 1 DPU.....      | 1   |
| 1.1 概述.....     | 1   |
| 1.2 主要特性.....   | 1   |
| 1.3 接口.....     | 3   |
| 1.4 功能描述.....   | 4   |
| 1.5 使用.....     | 11  |
| 2 HDMI.....     | 18  |
| 2.1 概述.....     | 18  |
| 2.2 主要特性.....   | 19  |
| 2.3 接口.....     | 19  |
| 2.4 功能描述.....   | 20  |
| 2.5 使用.....     | 23  |
| 3 MIPI DSI..... | 25  |
| 3.1 概述.....     | 25  |
| 3.2 主要特性.....   | 25  |
| 3.3 接口.....     | 26  |
| 3.4 功能描述.....   | 26  |
| 3.5 使用.....     | 29  |

## 图表目录

|  |    |
|--|----|
| 图表 1-1 DPU_TOP 架构框图 .....                  | 1  |
| 图表 1-2 标准 DPI 接口时序图 .....                  | 3  |
| 图表 1-3 DPI 接口定义 .....                      | 3  |
| 图表 1-4 DPI 接口时序图 .....                     | 3  |
| 图表 1-5 TTL 接口时序参数 .....                    | 3  |
| 图表 1-6 DPU 典型分辨率的时钟配置 .....                | 4  |
| 图表 1-7 DPU PIXEL PLL 架构简图 .....            | 5  |
| 图表 1-8 典型显示分辨率对应的 PLL 配置参数参考 .....         | 5  |
| 图表 1-9 DPU 不用格式下输出的主要 trans 特性 .....       | 6  |
| 图表 1-10 DPU AXI 输出的 ID 对应列表 .....          | 6  |
| 图表 1-11 DPU 内存格式的对齐要求 .....                | 7  |
| 图表 1-12 像素内存摆放格式 .....                     | 8  |
| 图表 1-13 DPU DPI 输出引脚映射图 .....              | 8  |
| 图表 1-14 DP 输出引脚映射图 .....                   | 9  |
| 图表 1-15 DPU 中断事件列表 .....                   | 10 |
| 图表 1-16 DPU 输出设备连接图 .....                  | 11 |
| 图表 1-17 DPU 工作配置流程图 .....                  | 12 |
| 图表 1-18 DPU 复位时序图 .....                    | 12 |
| 图表 1-19 DEC400 stream 和 DPU 读取格式对应关系 ..... | 15 |
| 图表 1-20 显示屏参示意图 .....                      | 16 |
| 图表 2-1 DWC_hdmi_tx 控制器框图 .....             | 18 |
| 图表 2-2 I2S 模式 .....                        | 21 |
| 图表 2-3 HDMI 系统中的彩条 .....                   | 23 |
| 图表 2-4 HDMI_COLOR_BAR_CFG 寄存器字段说明 .....    | 23 |
| 图表 3-1 引脚描述表 .....                         | 26 |
| 图表 3-2 MIPI DSI 功能框图 .....                 | 27 |

## 术语与缩略语

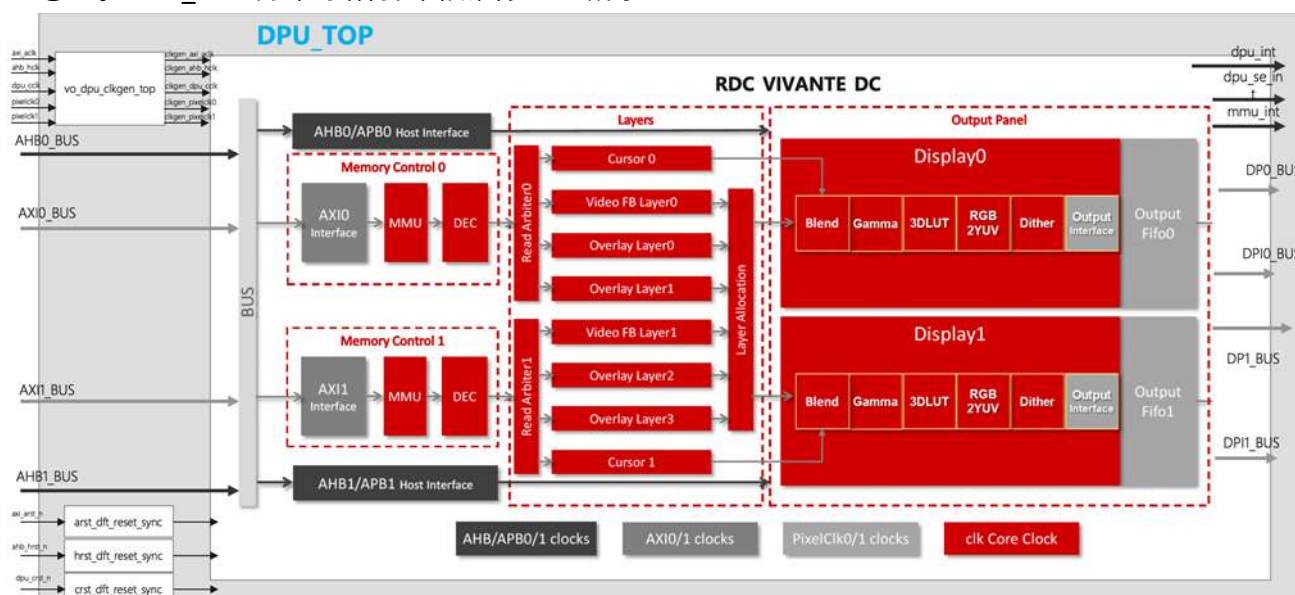
| 缩略语  | 英文全名                                     | 中文解释      |
|------|--|-----------|
| CSC  | Consumer Electronics Control             | 消费电子控制    |
| DHCP | High-bandwidth Digital ContentProtection | 高带宽数字内容保护 |
| DPI  | Display Pixel Interface                  | 显示像素接口    |
| DPU  | Display Process Unit                     | 显示处理单元    |
| DSI  | Display Serial Interface                 | 显示串行接口    |
| EoTP | End of Transmission Package              | 传输结束包     |
| SDR  | Single Data Rate                         | 单倍数据速率    |

# 1 DPU

## 1.1 概述

DPU 用来完成图像显示控制，将 DDR 中的图像转换成与显示设备匹配的分辨率，并按照显示设备需要的时序将像素流传输至视频接口（MIPI/HDMI/等），由视频接口驱动显示设备，完成显示。DPU 主要的功能包括图层缩放、图层旋转、图层叠加以及叠加后图像处理（包括 Gamma、Dither、Color）等。

DPU 最大支持 6 个图像层，2 个 cursor 层同时输入，支持两路并行显示，每路支持的最大输出分辨率为 4K@60。DPU\_TOP 内部的结构框图如图表 1-1 所示：



图表 1-1 DPU\_TOP 架构框图

## 1.2 主要特性

DPU 的主要特性如下：

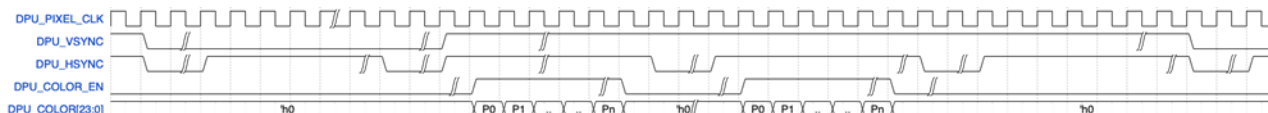
- 时钟与复位
  - AXI 总线时钟 ACLK，默认频率为 792M，其频率最终决定于 SoC 配置。
  - AHB 配置时钟 HCLK，默认频率为 396M，其频率最终取决于 SoC 配置。
  - DPU core 时钟，像素处理时钟，默认频率为 594M，其频率取决于当前显示设备的分辨率。
  - Pixelclk0/1 时钟，像素输出时钟，其由独立的 PLL 控制，具体频率取决于显示设备的分辨率，其计算关系以及和 core 时钟的关系，参见章节 1.4.1。
- AXI 总线
  - 数据总线位宽 128bit
  - 地址总线位宽 32bit，利用 8bit 的 user 可以扩展到 40bit。

- Burst 特性：在非压缩以及非 MMU 模式下，DPU 的读 trans 类型包括：16/128/256byte，在压缩以及 MMU 模式下，DPU 的读 trans 类型包括：6/32/64/96/128/160/192/ 224/256B。
- 不支持 cacheable/re-order/protection 传输。
- 读通道支持 out-of-order 以及 interleave。
- 支持 INC burst。
- AXI ARID 的位宽为 5bit。
- QoS 特性：支持两级 QoS 调整，High 和 Low，其对应的等级可以有软件静态配置 QoS；支持 line buffer 水线动态调整，当 line buffer 中的像素量低于软件配置的水线后，DPU 会发出高 QoS；当 line buffer 中像素量大于水线后，DPU 的 QoS 又会降低到 Low level。
- DPU 支持的最大 outstanding 能力是 64。
- 输入图像格式
  - RGB 格式：A2R10G10B0/ARGB8888/XRGB8888/RGB565/ARGB4444/XRGB4444
  - YUV 格式：YUY2/UYVY/NV12/NV16/P010/YV12
  - 支持 tile 以及 linear 内存格式，详细内容参见 1.4.3。
- 图像压缩
  - 支持 DEC400 压缩，详细信息参考 DEC400 章节描述。
- 图像叠加
  - 支持 proter\_duff 定义的 8 种 mix 方式：clr/src/dst/src\_over/dst\_over/src\_in/dst\_in/src\_out。
  - 支持全局 alpha 替换。
- 图像缩放
  - 水平缩放使用 3/5 阶滤波器；系数可配置，缩放系数为 15.16。
  - 垂直缩放支持 3 阶滤波器，系统可配置，缩放系数为 15.16。
- 图像旋转
  - 支持 90/180/270/xflip/yflip/xyflip 种缩放方式。
  - ARGB32bit：仅支持 superTileY4x8；ARGB16bit：仅支持 SuperTile8x8。
  - YUY2/UYVY/P010 TILE8X8\_XMAJOR
  - 其他格式都不支持 rotation。
- Gamma
  - Gamma table 可软件配置。
- Dither：使用软件配置的 dither table。
- Color：使用 17x17x17 的颜色校正系数表。
- 输出图像格式
  - DPI 接口：只支持 RGB 格式，30/24/18/16bit，详细格式参见章节 1.4.4。
  - DP 接口：支持 RGB 和 YUV 两种格式，详细格式参见章节 1.4.4。



- 中断特性
  - DC\_INT: REE 域 DPU 输出中断, 详细信息参见章节 1.4.5。
  - DC\_SE\_INT: TEE 域 DPU 输出中断
  - MMU\_INTR: DPU 内嵌 MMU 中断

## 1.3 接口



图表 1-2 标准 DPI 接口时序图

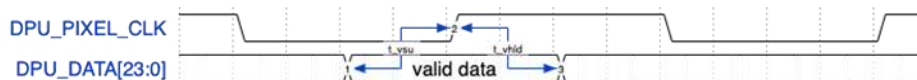
图表 1-3 DPI 接口定义

| 引脚名称             | 方向 | 宽度 | 说明           |
|------------------|----|----|--------------|
| DPU_PIXELCLK     | O  | 1  | TTL 输出像素时钟   |
| DPU_VSYNC        | O  | 1  | TTL 输出帧同步信号  |
| DPU_HSYNC        | O  | 1  | TTL 输出行同步信号  |
| DPU_COLOR_EN     | O  | 1  | TTL 输出像素使能信号 |
| DPU_COLOR_[0~23] | O  | 24 | TTL 输出像素数据信号 |

DPU 的 DPI 接口兼容 DPI-2, 但是只支持 RGB 输出, 其和标准 DPI-2 接口之间的对应关系如下:

|            |       |              |
|------------|-------|--------------|
| CM(DPI)    | <- -> | (不支持)        |
| DE(DPI)    | <- -> | DPU_COLOR_EN |
| Hsync(DPI) | <- -> | DPU_HSYNC    |
| Vsync(DPI) | <- -> | DPU_VSYNC    |
| SD(DPI)    | <- -> | (不支持)        |
| PCLK(DPI)  | <- -> | DPU_PIXELCLK |

DPI 接口输出的时序图如图表 1-4:



图表 1-4 DPI 接口时序图

图表 1-5 TTL 接口时序参数

| 符号       | 参数                    | 条件 | 标准模式 |       | 单位  |
|----------|-----------------------|----|------|-------|-----|
|          |                       |    | 最小值  | 最大值   |     |
| Pixelclk | pixel clock frequency |    | 10   | 74.25 | mHz |

| 符号     | 参数                     | 条件 | 标准模式 |     | 单位 |
|--------|------------------------|----|------|-----|----|
|        |                        |    | 最小值  | 最大值 |    |
| t_vsu  | Pixel value setup time |    | 1    | --  | ns |
| t_vhld | Pixel value hold time  |    | 0.7  | --  | ns |

## 1.4 功能描述

### 1.4.1 时钟特性

DPU 内部包括 AXI 总线和 AHB 总线的时钟，以及 core 时钟和像素时钟。总线时钟由 SoC 统一决定，数据总线的默认频率为 792M，配置总线的默认频率为 396M。DPU 的 core 时钟以及像素时钟由当前输出的分辨率决定。

DPU 的各个时钟选择的理论基础如下：

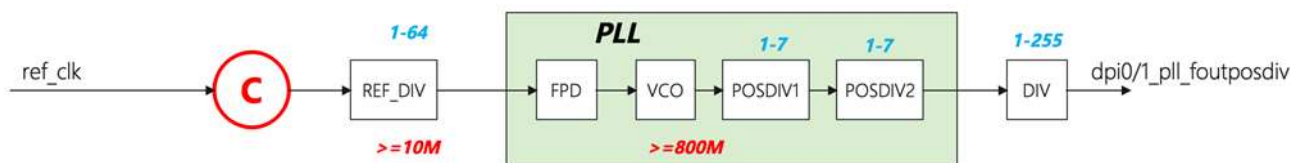
- 总线时钟 ACLK 的选择
  - 由图像分辨率以及帧率进行图像带宽计算  $BW(MB/s) = H_{act} * V_{act} * frame\_rate * pixel\_bytes / (1000 * 1000)$
  - DPU 支持多个图层的叠加，因此要求 AXI 总线的带宽大于多个图层所需的带宽总和。DPU AXI 总线带宽的计算参见章节 1.4.2。
- DPU 处理时钟 CCLK 的选择
  - 由于 DPU 可以同时读取多个图层，DPU 处理时钟由最大分辨率图层决定。
  - DPU 的处理能力是 1 个时钟处理 1 个像素（1T1P），因此要求： $dpu\_cclk \geq H_{act} * V_{act} * frame\_rate$ 。Hact 和 Vact 指的是最大图层的行有效像素和列有效像素。
- DPU 像素时钟由当前显示设备的屏参决定。
  - $Pixelclk = h\_total * v\_total * frame\_rate$
  - $H\_total = H_{act} + H\_blanking$ ,  $V\_total = V_{act} + V\_blanking$

典型分辨率下 dpu\_cclk 以及 dpu\_pixel0/1clk 的选择如图表 1-6 所示。

图表 1-6 DPU 典型分辨率的时钟配置

| 分辨率       | 帧率 | DPU_ACLK | DPU_CCLK | DPI_PIXEL0/1_CLK |
|-----------|----|----------|----------|------------------|
| 3840x2160 | 60 | 792      | 792      | 594              |
| 4096x2016 | 30 | 500      | 500      | 297              |
| 1920x1080 | 60 | 400      | 400      | 148.5            |
| 1080x720  | 60 | 300      | 300      | 74.25            |
| 640x480   | 60 | 200      | 200      | 25.175           |

像素时钟由 DPU 专用的 PLL 时钟进行配置，DPU 专用 PLL 的架构如图表 1-7：



图表 1-7 DPU PIXEL PLL 架构简图

对于 DPU 的配置需要注意事项如下：

- $FOUTVCO = (FREF / REF_{DIV}) \times (FBDIV + FRAC/224)$
- $FOUTPOSTDIV = FOUTVCO / POSDIV1 / POSTDIV2$
- 要求  $FREF/REF\_DIV \geq 10M$ ;  $FOUTVCO \geq 800M$
- 要求  $POSDIV1 \geq POSDIV2$

图表 1-8 是典型输出分辨率对应的 PLL 各个参数配置参考列表。

图表 1-8 典型显示分辨率对应的 PLL 配置参数参考

| 分辨率       | 帧率 | 像素时钟   | REFDIV | FBDIV | FRAC     | POSDIV1 | POSDIV2 | OUT_DIV |
|-----------|----|--------|--------|-------|----------|---------|---------|---------|
| 3840x2160 | 60 | 594    | 1      | 0x31  | 0x800000 | 1       | 1       | 2       |
| 4096x2016 | 30 | 297    | 1      | 0x31  | 0x800000 | 1       | 1       | 4       |
| 1920x1080 | 60 | 148.5  | 1      | 0x31  | 0x800000 | 1       | 1       | 8       |
| 1080x720  | 60 | 74.25  | 1      | 0x31  | 0x800000 | 1       | 1       | 16      |
| 640x480   | 60 | 25.175 | 200    | 0x43  | 0x222222 | 2       | 1       | 32      |

## 1.4.2 总线特性

DPU AXI 总线的主要特性：

- 读写两个通道，总线位宽为 128bit。
- Burst 特性：
  - 在非压缩以及非 MMU 模式下，DPU 的读 trans 类型包括：16/128/256byte。
  - 在压缩以及 MMU 模式下，DPU 的读 trans 类型包括：6/32/64/96/128/160/192/224/256B。
- DPU core 输出的地址为 32bit，集成上使用 8bit axi\_user 将其扩展到 40bit。
- 不支持 cacheable/re-order/protection 传输。
- 读通道支持 out-of-order 以及 interleave。
- 支持 INC burst。
- AXI ARID 的位宽为 5bit。
- QoS 特性：
  - 支持两级 QoS 调整，High 和 Low，其对应的等级可以由软件静态配置。

- QoS 支持 line buffer 水位动态调整，当 line buffer 中的像素低于软件配置的水位后，DPU 会发出高 QoS；当 line buffer 中像素大于水位后，DPU 的 QoS 又会降低到 Low level。
- DPU 支持的最大 outstanding 能力是 64。

DPU 在不同图像格式下输出的主要 trans 特性列表如下：

图表 1-9 DPU 不同格式下输出的主要 trans 特性

| 格式/类型  | 突发大小（单位：字节） |     |
|--|-------------|-----|
|  | 压缩          | 非压缩 |
| 32bit(A8R8G8B8,X8R8G8B8,A2R10G10B10)                   | 128         | 128 |
| 16bit(A4R34G4B4,<br>X4R4G4B4,A1R5G5B5,X1R5G5B5,R5G6B5) | \           | 128 |
| UYVY   | 128         | 128 |
| YUY2   | 128         | 128 |
| P010   | 256         | 128 |
| YV12   | \           | 128 |
| NV12   | 256         | 128 |
| NV16   | 256         | 128 |

图表 1-10 DPU AXI 输出的 ID 对应列表

| 读请求                | ARID |
|--------------------|------|
| Video0 layer       | 0x0  |
| Overlay0 layer     | 0x1  |
| Overlay1 layer     | 0x2  |
| Cursor0 layer      | 0x3  |
| Video1 layer       | 0x4  |
| Overlay2 layer     | 0x5  |
| Overlay3 layer     | 0x6  |
| Cursor1 layer      | 0x7  |
| Dec400 Tile status | 0x10 |
| MMU Request        | 0x1F |

| 写请求        | AWID ID |
|------------|---------|
| Write back | 0x0     |

总线带宽计算：DPU AXI 总线的带宽由 DPU 的 AXI 总线的时钟频率 freq(ack)、发出的 trans\_size、读数据的平均 latency(ns)以及 outstanding 共同决定。其计算如下：

$$BW(MB/s) = \text{freq(ack)} * \text{tran\_size} * \text{outstanding} / \text{latency}$$

对于 DPU，在非压缩模式下，典型 trans\_size 为 128 bit，DPU 的 outstanding 可以通过软件设定，最大为 64，latency 和当前场景有关，在 DPU 进行带宽估算时，假定当前场景下 DPU 的平均 latency 为 1000。

QoS 的动态调整可以改善 latency 来达到提高带宽的目的。

### 1.4.3 输入图像格式

DPU 不同格式对图像大小、首地址以及 stride 的对齐要求如下：

图表 1-11 DPU 内存格式的对齐要求

| 线性/图块                       | 突发大小<br>(字节) | 格式                         | 宽 x 高对齐 (单位: 像素) | Stride 对齐 (字节) | 基地址<br>(字节) |
|-----------------------------|--------------|----------------------------|------------------|----------------|-------------|
| Linear                      | 128          | Planar YUV/Semi-planar YUV | 2x2              | 128            | 128         |
|                             | 128          | YUY2/UYYV                  | 2x1              | 128            | 128         |
|                             | 128          | RGB                        | \                | 128            | 128         |
|                             | 256          | Planar YUV/Semi-planar YUV | 2x2              | 256            | 256         |
|                             | 256          | YUY2/UYYV                  | 2x1              | 256            | 256         |
|                             | 256          | RGB                        | \                | 256            | 256         |
| SuperTileX8x8               | 128          | 16bpp RGB                  | 64x64            | 128            | 128         |
| SuperTileX8x4/SuperTileY4x8 | 128          | 32bpp RGB                  | 64x64            | 128            | 128         |
| 8x8 tile                    | 128          | YUY2/UYYV                  | 8x8              | 128            | 128         |
|                             | 128          | NV12-Y planar              | 8x8              | 32             | 128         |
|                             | 128          | P010-Y planar              | 8x8              | 32             | 128         |
|                             | 256          | NV12-Y planar              | 8x8              | 64             | 256         |
|                             | 256          | P010-Y planar              | 8x8              | 64             | 256         |

| 线性/图块    | 突发大小<br>(字节) | 格式                | 宽 x 高对齐 (单位: 像素) | Stride 对齐 (字节) | 基地址<br>(字节) |
|----------|--------------|-------------------|------------------|----------------|-------------|
| 8x4 tile | 128          | NV12-UV<br>planar | 8x8              | 32             | 128         |
|          | 128          | P010-UV<br>planar | 8x8              | 32             | 128         |
|          | 256          | NV12-UV<br>planar | 8x8              | 64             | 256         |
|          | 256          | P010-UV<br>planar | 8x               | 64             | 256         |

不同图像格式的像素内存摆放如图表 1-12:

| ID                | format               |           | D31        | D30 | D29 | D28 | D27 | D26 | D25 | D24 | D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 | D15        | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|-------------------|----------------------|-----------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| 00 => X4R4G4B4    | XRG84444             | package   | don't care |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     | don't care |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 01 => A4R4G4B4    | ARG84444             | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 02 => X1R5G5B5    | ARG81555             | package   | A1         |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 03 => A1R5G5B5    | XRG81555             | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 04 => R5G6B5      | RG8565               | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 06 => A8R8G8B8    | ARG88888             | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 05 => X8R8G8B8    | XRG88888             | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 07 => YUY2        | YUV422-YUY2(package) | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 08 => UYVY        | YUV422-UYVY          | package   |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 09 => INDEX8      | index8               |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 0A => MONOCHROME  |                      |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 0F => YV12        | YUV420-YV12          | y planar  |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | u planar  |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | v planar  |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 10 => A8          | A8                   |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 11 => NV12        | NV12(YUV420_sp)      | y planar  |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | uv planar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 12 => NV16        | NV16(YUV422_sp)      | y planar  |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | uv planar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 13 => RG16        |                      |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 14 => R8          |                      |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 15 => NV12_10BIT  | NV12_10BIT           | y plannar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | uv planar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 16 => A2R10G10B10 | ARG82101010          | package   | A1         |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 17 => NV16_10BIT  | YUV422_NV16_10BIT    | y plannar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | uv planar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 18 => INDEX1      | index1               |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 19 => INDEX2      | index2               |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 1A => INDEX4      | index4               |           |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| 1B => P010        | YUV420_10bit_P010    | y plannar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
|                   |                      | uv planar |            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |            |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |

图表 1-12 像素内存摆放格式

## 1.4.4 输出图像格式

DPU 支持两种接口输出: DPI 和 DP 格式, DPI 格式输出至 DSI 驱动 MIPI Panel 或通过 GPIO 驱动 TTL Panel, 而 DP 接口输出至 HDMI。

DPU DPI 输出引脚映射格式如图表 1-13 所示:

| color depth | code | cycle   | D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|-------------|------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| RGB888      | 0x0  | eve/odd | R7  | R6  | R5  | R4  | R3  | R2  | R1  | R0  | G7  | G6  | G5  | G4  | G3  | G2  | G1 | G0 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| RGB666-CFG1 | 0x1  | eve/odd |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| RGB666-CFG2 | 0x2  | eve/odd |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| RGB565-CFG1 | 0x3  | eve/odd |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| RGB565-CFG2 | 0x4  | eve/odd |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |
| RGB565-CFG3 | 0x5  | eve/odd |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |    |    |    |    |

图表 1-13 DPU DPI 输出引脚映射图

对于 DP 输出格式对应的引脚映射如图表 1-14 所示:

### 图表 1-14 DP 输出引脚映射图

DPU 输出三种中断：MMU\_INTR/DC\_INTR 以及 DC\_SE\_INTR 中断。

DEC400 和 DPU Core 共用一个 DC\_INTR，当 CPU 获取到中断后，通过查询 AQIntrAcknowledge 和 gcregAHBDECIntrAcknowledgeEx2 来获取何种事件的发生。

DPU 常见的中断类型如下:

- ©【平头哥上海半导体技术有限公司】版权所有

- Reset done interrupt: 表示硬件复位完成中断。

对于 DPU 内部中断对用的所有事件如图表 1-15 所示：

图表 1-15 DPU 中断事件列表

| 位     | AQIntrAcknowledge<br>(0x0004)用于非安全内存 | gcregSeHiIntrAcknowledge<br>(0x4001)用于安全内存 | gcregSeMMUIIntrAcknowledge<br>(0x4012)用于 MMU |
|-------|--------------------------------------|--|--|
| 0     | DC Panel0 中断                         | DC Panel0 中断                               | MMU0 从不存在                                    |
| 1     | DC Panel1 中断                         | DC Panel1 中断                               | MMU0 客户端越界                                   |
| 2     | 安全复位完成中断                             | 安全复位完成中断                                   | MMU0 大小超出范围                                  |
| 3     | Video0 非安全配置冲突                       | Video0 非安全配置冲突                             | MMU0 读安全异常                                   |
| 4     | Overlay0 非安全配置冲突                     | Overlay0 非安全配置冲突                           | MMU0 写安全异常                                   |
| 5     | Overlay1 非安全配置冲突                     | Overlay1 非安全配置冲突                           | MMU0 页不存在                                    |
| 6     | Video1 非安全配置冲突                       | Video1 非安全配置冲突                             | MMU0 页写入不可写页                                 |
| 7     | Overlay2 非安全配置冲突                     | Overlay2 非安全配置冲突                           | MMU0 读安全内存冲突                                 |
| 8     | Overlay3 非安全配置冲突                     | Overlay3 非安全配置冲突                           | MMU1 从不存在                                    |
| 9     | Cursor0 非安全配置冲突                      |  | MMU1 客户端越界                                   |
| 10    | Cursor1 非安全配置冲突                      |  | MMU1 大小超出范围                                  |
| 11    | DC Panel0 数据下溢                       |  | MMU1 读安全异常                                   |
| 12    | DC Panel1 数据下溢                       |  | MMU1 写安全异常                                   |
| 13    |                                      |  | MMU1 页不存在                                    |
| 14    |                                      |  | MMU1 页写入不可写页                                 |
| 15    |                                      |  | MMU1 读安全内存冲突                                 |
| 16-29 |                                      |  |  |
| 30    | AXI Bus1 错误                          | AXI Bus1 错误                                |  |
| 31    | AXI Bus0 错误                          | AXI Bus0 错误                                |  |

## 说明

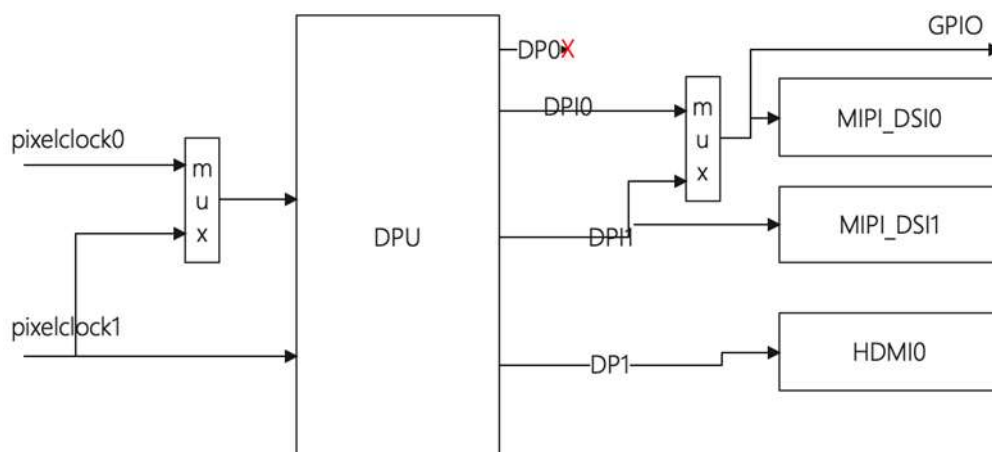
图表 1-15 没有列出 gcregAHBDECIntrAcknowledgeEx2 和 gcregAHBSeDECIntrAcknowledgeEx2 DEC TEE 和 REE 的中断事件，这两个中断的定义上述已经说明。



## 1.4.6 工作模式

DPU 的下游连接关系如下所示;对于显示通路 0,只可以驱动 mipi\_dsi0 以及 GPIO pad。而对于 panel1,它既可以驱动 HDMI (通过 DP 驱动),也可以驱动 mipi\_dsi0 以及 mipi\_dsi1 和 GPIO\_PAD,通过 DPI 接口进行驱动。这种连接关系对应的 DPU 三种显示模式:

- 双屏异显:所谓异显就是指两个显示通路互不干扰,使用不同的 PLL 为各自提供像素时钟,这个 case 下注意一个图层不能用在两个显示通路上。
- 双屏同显:两个屏幕显示的内容完全一致,这时候,需要将 DPI1 的输出通过 mux 切到第 0 路的输出接口上;Pixelclock0 的时钟可以不进行配置。
- 双屏部分同显:所谓双屏部分同显,是指两个通路有部分图层是一样的内容。这种方式需要满足两个约束:
  - 通路 0 使用的是通路 1 的像素时钟。
  - 同图层内容是指软件将同一个图像的内存地址给不同的图层,对硬件来说还是不同的图层。

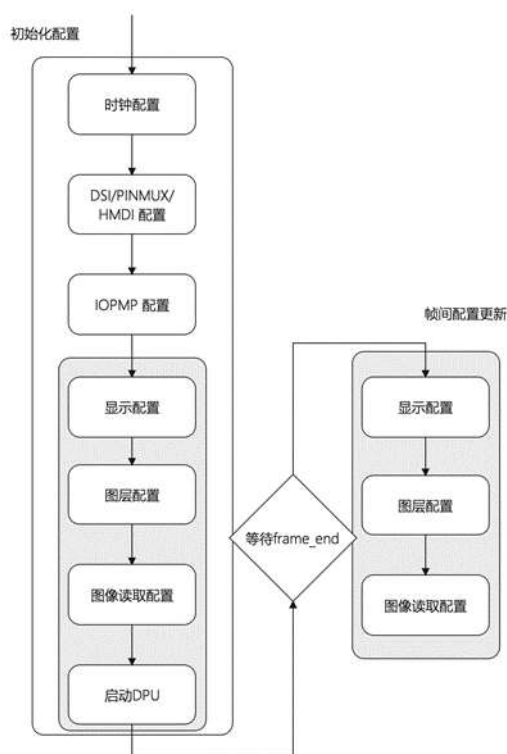


图表 1-16 DPU 输出设备连接图

## 1.5 使用

DPU 的软件配置流程如下:分为初始化配置和帧间配置更新两部分。

初始化配置包括时钟配置;像素流输出接口 (MIPI/HDMI 等配置在本文档中不体现,请参见相关模块的用户手册);DPU core 配置流程如下:



图表 1-17 DPU 工作配置流程图

## 说明

视频输出接口（DSI/PINMUX/HDMI）、IOPMP 配置以及 MMU 配置不在本文档范畴之内。

## 1.5.1 时钟和复位配置

### 1.5.1.1 复位流程

#### 1.5.1.1.1 正常复位流程

1. 打开 VO 子系统的复位，默认该复位是释放状态。vo\_rst\_n = AO\_RSTGEN\_BASE+0x30。
2. 释放 DPU 的复位：软复位控制，对应的是 VOSYS\_SYSREG\_APBIF\_APB3S\_BADDR+0x4[0:2]，详细内容参见对应的寄存器映射。
3. AQHiClockControl 的 soft\_reset 进行解复位。

DPU 的解复位要求满足如下时序：复位时间要求保持至少 32 个时钟周期，释放 128 个时钟周期后，才能进行配置。



图表 1-18 DPU 复位时序图

### 1.5.1.1.2 异常复位流程

在 DPU 出现异常后，需要对系统进行复位，需要查询 AQAxConfig 的 AXI\_REQ\_PENDING\_FLAG 的值，在得到该值为 0 时，按照正常的复位流程进行复位，如果该状态不为 0，则一直等待。如果长时间等不到，那么需要子系统或者全芯片复位，子系统复位以及全芯片复位不在本文档讨论范畴。DPU 不支持仅仅 core 域做复位，因此在 DPU 复位后，需要重新进行配置。

### 1.5.1.2 时钟配置

1. 根据当前的工作模式，以及输出的分辨率确定 PLL 的频率，并根据章节 1.4.1 PLL 的配置方法进行 PLL 频率设定。
2. 打开 VO 子系统的时钟；时钟默认是打开的。对应的寄存器如下：vo\_clk\_gen = AO\_CLKGEN\_BASE + 0x130。
3. 打开 DPU 子系统的复位和时钟；默认状态下，DPU 时钟是关闭的，对应的寄存器如下：  
VOSYS\_SYSREG\_APBIF\_APB3S\_BADDR+0x50，详细信息参见对应的寄存器映射。
4. 打开 DPU 内部时钟以及动态时钟开关，参见章节 1.4.1 DPU 内部的时钟控制。

## 1.5.2 DPU 配置

本章节只针对 REE 域进行配置，TEE 的配置方式类似。

### 1.5.2.1 图层配置

DPU 共有 2 个 Video 层、4 个 overlay 以及 2 个 Cursor 层。根据当前应用场景选择所需使用的图层进行配置，具体的配置如下：

1. 配置每个图层的属性：dcregFrameBufferConfig0~1、dcregOverlayConfig0~3；主要的配置信息包括：透明度配置（TRANSPARECY）、旋转角度配置（ROT\_ANGLE）、内存格式配置、色彩通道交换（SWIZZLE）、图像格式设定、ROI 区域设定、图像是否为压缩格式、gamma 使用、RGB2RGB 使能、YUV 嵌位控制以及图层使能。
2. 图像内存地址设定：根据步骤 1 配置的图形格式进行内存地址设定。如果是 package 格式，通过 dcregFrameBufferAddress0~1/dcregOverlayAddress0~3 进行内存地址分配，如果是 planar，则需要对对应的 planar 地址：  
dcregFrameBufferU/VPlanarAddress0~1/dcregOverlayU/VPlanarAddress0~3 进行配置。该信息需要每帧都进行配置。
3. 地址 Stride 配置：stride 是指图像在当前 planar 的某一行首地址到下一行首地址的 byte 距离。通过 dcregFrameBufferStride0~1、dcregFrameBufferUStride0~1、dcregFrameBufferVStride0~1、dcregOverlayStride0~3、dcregOverlayUStride0~3、dcregOverlayVStride0~3 进行配置。
4. 配置各个图层的位置信息。

- a) 对于 video 层, 通过 dcregFrameBufferSize0~1 进行图像的宽高设定, 如果有使能 ROI, 则通过 dcregFrameBufferROIOrigin0~1、dcregFrameBufferROISize0~1、dcregFrameBufferConfigEx0~1 进行 ROI 区域进行配置。
  - b) 通过 dcregOverlayTL0~3 以及 dcregOverlayBR0~3 进行图层位置和大小设定, 如果有使能 ROI, 则进行 dcregOverlayROIOrigin0~3、dcregOverlayROISize0~3 进行 ROI 区域进行配置。
5. 根据透明图信息进行 blending 以及 color key 的设定。
- a) Color key 设定: 通过 dcregFrameBufferColorKey0~1、dcregFrameBufferColorKeyHigh0~1、dcregOverlayColorKey0~3、dcregOverlayColorKeyHigh0~3 进行关键颜色的指定。通过 dcregFrameBufferBGColor0~1 对非关键色进行替换。注意: video 0 以及 layer0-1 共用一个 BGColor0, video1 和 layer2-3 共用 BGColor1。
6. 缩放控制配置:
- a) 缩放倍数的控制: dcregFrameBufferScaleFactorX0~1、dcregFrameBufferScaleFactorY0~1。
  - b) 滤波器类型的选择: dcregFrameBufferScaleConfig0~1 进行滤波器类型进行配置。
  - c) 滤波器系数的配置: 通过 dcregHoriFilterKernelIndex0~1 和 dcregHoriFilterKernel0~1 完成对水平滤波系数的配置。通过 dcregVertiFilterKernelIndex0~1 和 dcregVertiFilterKernel0~1 完成对垂直滤波系数的配置。
  - d) 同理对 layer0 和 layer2 图层进行缩放配置, 注意 layer1 和 layer3 不支持缩放。
7. Color space 转化配置: 如果读入的图像格式为 YUV 格式, 则需要通过 dcregFrameYUVToRGBCoef\*0~1 相关寄存器对 color space 的系数进行配置, 如果有潜入位需求, 也需要对 dcregFrameY/UVClampBound0~1 值进行相关配置。
8. 如果在图层属性中使能 gamma, 则需:
- a) 通过 dcregDeGammaIndex0~1、dcregDeGammaData0~1 以及 dcregDeGammaExData0~1 完成对 gamma table 系数的配置。
9. 通过 dcregSeFrameBufferConfigEx0~1、dcregOverlayConfigEx0~3 对图层叠加顺序的选择以及对应 panel 的选择。
- a) 图层叠加顺序的选择: 默认叠加顺序是背景层和第 0 层叠加后作为第 1 层的背景层与第 1 层叠加, 依次完成对所有图层的叠加。因此通过 Layer\_sel 进行当前图层对应叠加的第几层进行选择。
  - b) 通过 PANEL\_SEL 选择对应的 panel。
10. 需要注意: 图层选择具有排他性, 举例来说, video0 用于 panel0 的 source, 那么就不能用于 panel1 的 source。
11. 由于显示有实时性的要求, 为了保证 DPU 在内存及时取数, 有设置自动流水线机制:
- a) 通过 dcregFrameBufferWaterMark0~1 设置硬件中 cache 的水线位置。
  - b) 通过 dcregQos 设计 low latency 以及 hight latency 的 QoS 的值。

- c) 当 DPU 内部的 Cache 水位低于软件设置的水线时，DPU 自动使用 high QoS 向总线要数据，在 DPU 高于设置的水线时，自动跳转到 Low QoS 向总线要数据。
- d) 通过配置 gcReadOT 来设定 DPU 的读 DDR 带宽能力，计算公式为： $BW(MB/s) = \text{freq}(\text{axi\_clk}) * \text{readOT} * \text{trans\_size} / \text{latency}$ 。其中 trans\_size 表示 DPU AXI 读写的 trans 大小，在 DPU trans 典型值为 128B，latency 是从 DPU 发出读请求到 DPU 拿到本次传输的最后一个数据所需的时间，单位 ns。带宽分析是个系统性工程，本文档只给出 DPU 带宽计算的基本公式。OT 的设置要在整个场景中平衡设定。

### 1.5.2.2 MMU 配置

NA

### 1.5.2.3 DEC400 配置

1. 根据属性决定是否需要是能 DEC400，通过 gcregAHBDECControl 的 Disable\_compression 进行设置，该设定需要和图层定义的属性匹配。
2. 如果需要使能 DEC，则打开时钟以及使能 clock gating。
3. 通过 gcregAHBDECControlEx、gcregAHBDECControlEx2 进行属性配置。
4. 配置 gcregAHBDECReadConfig0~15 对不同流进行压缩格式的确认。不同流对应的图层信息如图表 1-19 所示：

图表 1-19 DEC400 stream 和 DPU 读取格式对应关系

| DC Layers      | Format/Planar              | DEC400 Stream |
|----------------|----------------------------|---------------|
| Video layer0   | Package format or Y planar | stream0       |
| Video layer0   | U/UV planar                | stream1       |
| Video layer0   | V planar                   | stream2       |
| Overlay layer0 | Package format or Y planar | stream3       |
| Overlay layer0 | U/UV planar                | stream4       |
| Overlay layer0 | V planar                   | stream5       |
| Overlay layer1 | Package format or Y planar | stream6       |
| Overlay layer1 | U/UV planar                | stream7       |
| Overlay layer1 | V planar                   | stream8       |
| Cursor layer0  |                            | stream9       |
| Video layer1   | Package format or Y planar | stream16      |
| Video layer1   | U/UV planar                | stream17      |

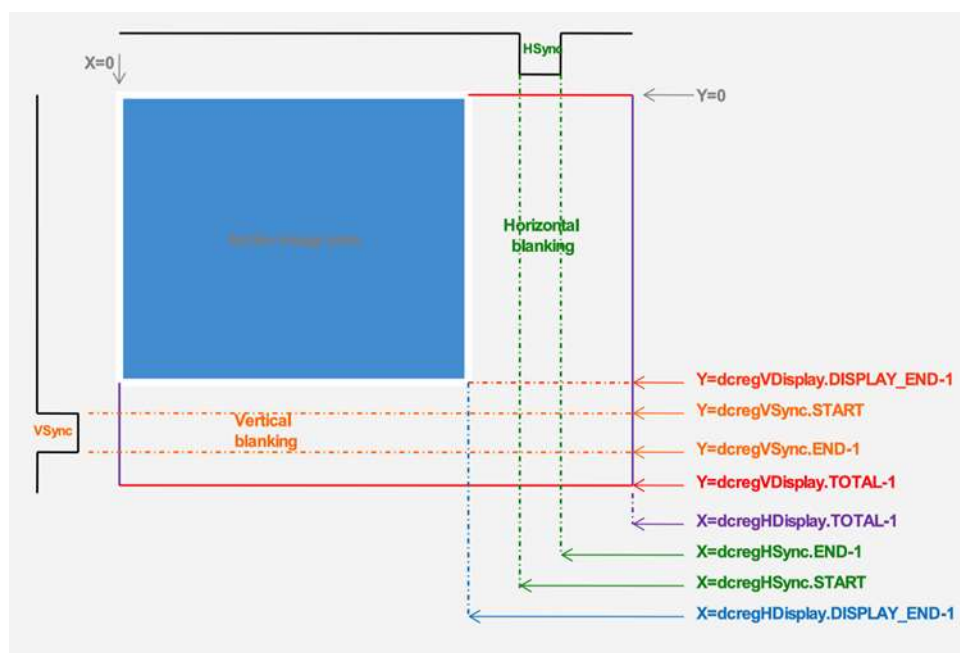
| DC Layers      | Format/Planar              | DEC400 Stream |
|----------------|----------------------------|---------------|
| Video layer1   | V planar                   | stream18      |
| Overlay layer2 | Package format or Y planar | stream19      |
| Overlay layer2 | U/UV planar                | stream20      |
| Overlay layer2 | V planar                   | stream21      |
| Overlay layer3 | Package format or Y planar | stream22      |
| Overlay layer3 | U/UV planar                | stream23      |
| Overlay layer3 | V planar                   | stream24      |
| Cursor layer1  |                            | stream24      |

详细的配置流程参见 DEC400 配置文档文档。

### 1.5.2.4 显示配置

在完成对图像图层的属性配置后，需要对两个显示通路进行配置：

1. 根据显示设备手册，进行时序参数配置，时序确定后，帧间配置时不允许修改该参数。可以参见图表 1-20：



图表 1-20 显示屏参示意图

2. 根据 dcregPanelConfig0~1 对显示通路的属性进行配置。
  - a) 对输出时序控制信号的极性进行控制，注意这些属性在初始化确定后，不允许调整。
  - b) 对输出的后图像处理功能进行选择：包括 gamma 以及 3D LUT 功能的选择。

3. 通过 dcregDpConfig0~1 的 bus\_output\_sel 选择当前是 DPI 输出还是 DP 输出,如果选择的是 DP, 使用 DP\_DATA\_FORMAT 和 DP\_DATA\_YUV\_FORMAT 进行输出格式选择, 如果是 DPI 格式输出, 通过 dcregDpiConfig0~1 对 DPI 输出格式进行选择。注意需要和后续模块使用的输入格式保持一致。输出格式可以参照章节 1.4.4 的描述。
4. 如果输出格式为 YUV, 则需要通过 dcregRGBToYUVCoef[0-3]0~1 进行系数表配置。
5. 如果选通 3D-LUT, 通过 dcregThreedLutScale 相关寄存器进行配置。
6. 通过 dcregPanelConfigEx0~1 配置配置参数生效时间点, default0 表示所有新的设定在下一帧 vsync 后才会生效。如果设置成 1, 需要软件保证配置的设定在下一帧 vsync 到来前完成配置。建议设置成 0, 该寄存器可以理解为是否使能 shadow 功能。

### 1.5.2.5 启动 DPU

在保证所有的配置正确无误后,

1. 使能所有的中断, AQIntrEnbl、gcregAHBDECIntrEnblEx2。
2. 启动 DPU: dcregDisplayPanelStart。

### 1.5.2.6 帧间配置更新说明

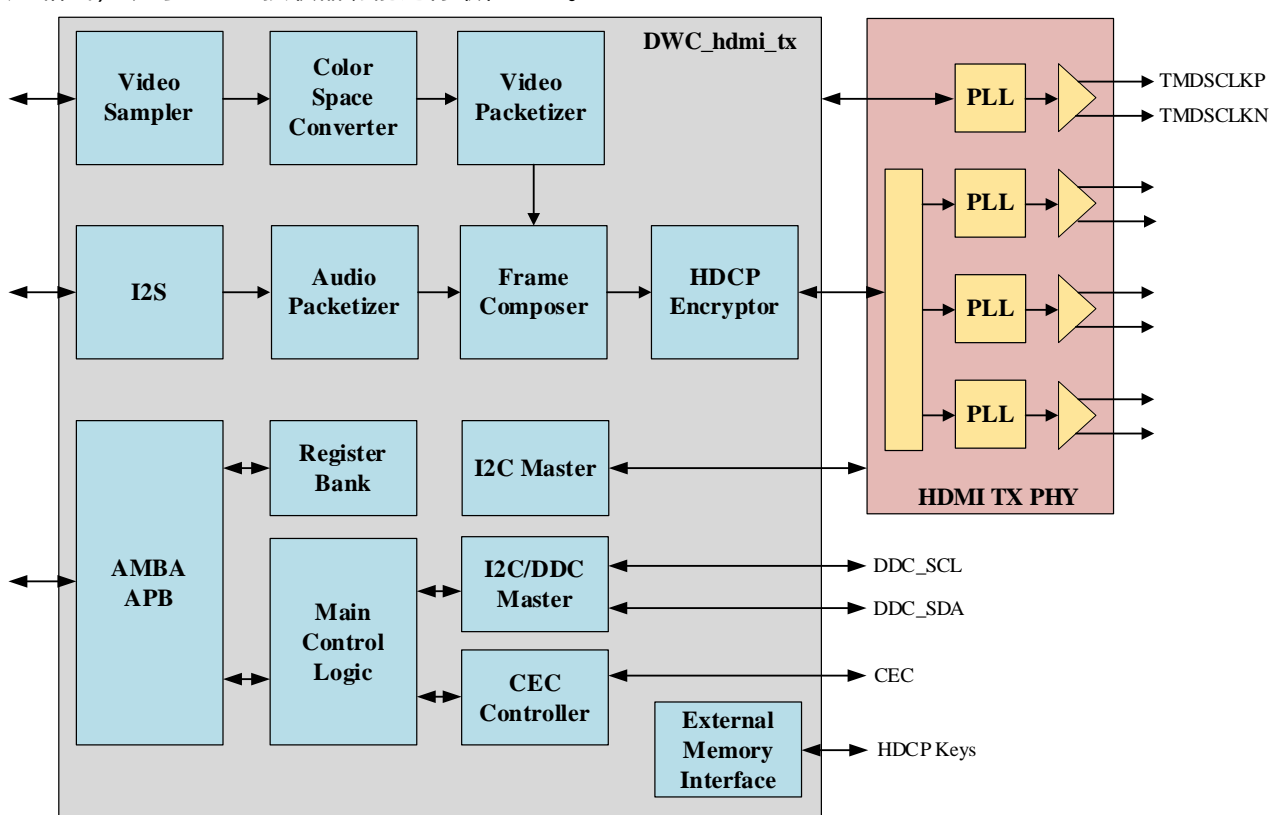
在 CPU 响应 DPU 中断后, 查询当前的中断是否为 frame end, 如果是, 则参照 1.5.2.1 和 1.5.2.4 进行配置参数的更新。

## 2 HDMI

### 2.1 概述

DWC\_hdmi\_tx 提供各种标准音频、视频和系统接口。

输入视频流可以是单倍数据速率 (SDR) 总线格式的 RGB 4:4:4、YCbCr 4:2:2、YCbCr 4:4:4 或 YCbCr 4:2:0。视频模式下的时序格式必须遵循 CEA-861-E 规范。嵌入式色彩空间转换允许在 HDMI 源侧转换像素颜色格式，以与 HDMI 接收器功能进行最佳匹配。



图表 2-1 DWC\_hdmi\_tx 控制器框图

输入音频流可以通过标准 I2S 格式接口提供（适用于所有音频类型：L-PCM、NL-PCM 和 HBR）。

系统接口（连接到处理器总线的接口）是 AMBA APB。

最后，该控制器可以输出全高清视频，色彩模式高达 48 位。此外，通过对输入视频流执行自动像素重复，在低分辨率视频格式上插入多达 8 通道的高保真音频。



## 2.2 主要特性

DWC\_hdmi\_tx 具有以下特性：

- AMBA APB 3.0，地址宽度为 16 位，数据宽度为 8 位
- HDMI 主机中断，包括 HDCP1.4 和 SCDC
- CEC 引擎专用中断信号，由唤醒事件触发
- 640x480p@59.94Hz/60Hz
- 720x480p@59.94Hz/60Hz
- 1280x720p@59.94Hz/60Hz
- 1920x1080p@59.94Hz/60Hz
- 3840x2160p@29.97Hz/30Hz
- 3840x2160p@59.94Hz/60Hz
- 4096x2160p@29.97Hz/30Hz
- 4096x2160p@59.94Hz/60Hz
- 24/30-bit RGB 4:4:4
- 24/30-bit YCbCr 4:4:4
- 16/20-bit YCbCr 4:2:2
- 24/30-bit YCbCr 4:2:0
- 支持色彩空间转换
- 支持内部像素重复
- I2S 音频接口，多达 8 个通道
- 支持消费电子控制（CEC）
- 支持 TMDS 数据加扰
- 支持 HDCP，HDCP 版本为 HDCP1.4
- SCDC I2C DDC 接入
- I2C DDC、EDID 块读取模式

## 2.3 接口

| 引脚名称      | 方向 | 宽度 | 说明                                   |
|-----------|----|----|--------------------------------------|
| HDMI_CEC  | IO | 1  | CEC 数据总线                             |
| HDMI_HPD  | IO | 1  | HDMI 热插拔检测信号                         |
| HDMI_REXT | IO | 1  | 用于 HDMI 操作模式的所有 TMDS 数据通道的 Rx 存在检测信号 |
| HDMI_SCL  | IO | 1  | 用于 HDCP 和 E-EDID 的 HDMI DDC I2C 从时钟  |

| 引脚名称            | 方向 | 宽度 | 说明                                  |
|-----------------|----|----|-------------------------------------|
| HDMI_SDA        | IO | 1  | 用于 HDCP 和 E-EDID 的 HDMI DDC I2C 从数据 |
| HDMI_TMDSCCLKN  | O  | 1  | 负 TMDS 差分线路驱动器时钟输出                  |
| HDMI_TMDSCCLKP  | O  | 1  | 正 TMDS 差分线路驱动器时钟输出                  |
| HDMI_TMDSDATAN0 | O  | 1  | 数据通道 0 的负 TMDS 差分线路驱动器数据输出          |
| HDMI_TMDSDATAN1 | O  | 1  | 数据通道 1 的负 TMDS 差分线路驱动器数据输出          |
| HDMI_TMDSDATAN2 | O  | 1  | 数据通道 2 的负 TMDS 差分线路驱动器数据输出          |
| HDMI_TMDSDATAP0 | O  | 1  | 数据通道 0 的正 TMDS 差分线路驱动器数据输出          |
| HDMI_TMDSDATAP1 | O  | 1  | 数据通道 1 的正 TMDS 差分线路驱动器数据输出          |
| HDMI_TMDSDATAP2 | O  | 1  | 数据通道 2 的正 TMDS 差分线路驱动器数据输出          |

## 2.4 功能描述

### 2.4.1 视频像素采样器

视频像素采样器块根据色彩深度（深色）和格式配置定义的视频数据输入映射同步视频数据。

### 2.4.2 视频打包器

该块主要进行以下操作：

- 像素重复（如果未在输入视频流中执行并且是您需要的）
- 深色模式下 10 位封装
- 根据 HDMI 规范重新映射 YCbCr 422
- 从像素或重复时钟到最终 TMDS 时钟域的时钟速率转换（通过 FIFO）

### 2.4.3 HDMI 2.0 TMDS 加扰功能

DWC\_hdmi\_tx 支持 TMDS 数据加扰，这是在 RGB4:4:4、YCbCr 4:4:4 或 YCbCr 4:2:2 视频格式中传输 2160p@60Hz 所需要的。当启用 HDMI 2.0 支持功能时，TMDS 频率高于 340MHz 的所有输出视频模式都需要通过将 `fc_invidconf.HDCP_keepout` 寄存器字段设置为 1'b1 和 `fc_scrambler_ctrl.scrembler_en` 设置为 1'b1 来激活加扰功能。对于频率等于或低于 340MHz 的视频模式，激活加扰功能是可选的。激活与否取决于您的要求和连接的 HDMI Sink 设备。

## 2.4.4 色彩空间转换

该块具有以下视频色彩空间转换功能：

- RGB 到/从 YcbCr 的转换
- 4:2:2 到/从 4:4:4 向上（像素重复或线性插值）/向下转换
- 限制到/从全量化范围转换

CSC 支持 CEA-861-E 规范中报告的所有时序和以下像素模式：

- RGB 444 和 YCbCr 444：24、30 位
- YCbCr 422：16、20 位

不支持与 YCbCr 4:2:0 之间的色彩空间转换。

## 2.4.5 音频接口

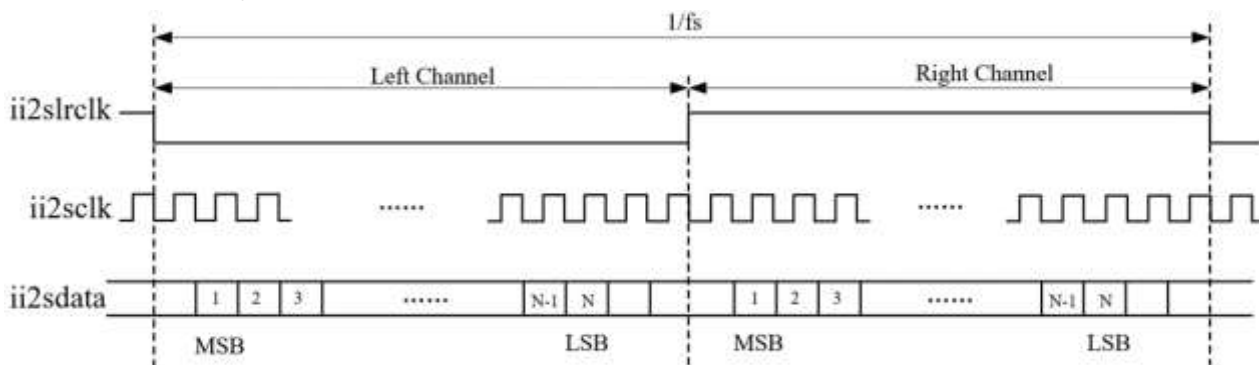
I2S 接口使用 I2S 音频时钟输入对线性 PCM 输入数据进行采样，并将其存储在输入音频 FIFO 中。有四条 I2S 数据线，支持高达 192kHz 的采样率（支持双通道标准 I2S 的最大理论音频速率为 768kHz 或 1536kHz（仅在 HDMI 2.0 中支持））。I2S 接口符合 NXP 的 I2S 规范。

每个 I2S 接口支持两个音频通道。DWC\_hdmi\_tx 有四个 I2S 接口，可同时支持多达八个 192kHz 的音频通道。每个音频样本宽度可以配置为 16 位到 24 位。aud\_conf1 寄存器的 I2S\_width 字段选择每个右/左采样的位宽。每个右/左通道可以携带 1 到 N 位（N = 16 ~ 24）。

ii2slrclk 输入信号的频率必须与音频采样率  $f_s$  相同，即 32kHz 至 192kHz。INPUTCLKFS 设置必须与 ii2sclk 的频率一致。

在 I2S 模式下（如图表 2-2 所示），子帧数据的最有效位（MSB）必须在 ii2slrclk 转换后的 ii2sclk 第二个有效沿发送。然后按顺序发送直到最低有效位（LSB）的其他位。根据字长、ii2sclk 频率和采样率，在一个样本的 LSB 和下一个样本的 MSB 之间可能存在未使用的 ii2sclk 周期。

子帧可以与子帧数据大小一样宽或更大（例如，BPCUV + 24 位音频采样 = 29 个时钟周期）。在典型的场景中，子帧可以是 32 个时钟宽度，并且音频采样 LSB 之后的所有时钟周期都未使用，ii2sdata 设置为 0，与音频采样宽度无关。



图表 2-2 I2S 模式

## 2.4.6 帧合成器

该块将视频、音频和数据包组装成一个一致的帧，然后最终传输到 HDMI TX PHY。

## 2.4.7 E-EDID/HDCP/SCDC I2C E-DDC 接口

E-DDC 通道是一个专用的 I2C 主接口，允许根据系统需要读取 Sink E-EDID。通过该接口，可以访问 HDMI 2.0 子系统 Sink EDID 和 SCDC 结构。

DWC\_hdmi\_tx 中包含的 I2C 主接口符合 I2C 总线规范的 2.1 版，可通过从地址 0x7E00 开始的 APB 接口进行访问。

提供了以下 I2C 操作：

- 单个数据字节写
- 单个数据字节读
- 连续数据字节读（8 字节）
- 单个数据字节扩展读
- 连续数据字节扩展读（8 字节）
- SCDC 更新读
- SCDC 读请求检测
- SCDC 轮询机制
- 总线清除机制

当 DWC\_hdmi\_tx 的外部设备将 SDA 线路驱动为低电平时，SCDC 读请求检测就会运行。这种检测是准确的，直到只有 DWC\_hdmi\_tx I2C 主接口驱动 I2C DDC 总线。

## 2.4.8 CEC 硬件引擎

CEC 是一种协议，可在环境中的所有各种视听产品之间提供高级控制功能。它是 HDMI 规范中的可选功能。仅使用一条双向线路进行发送和接收。

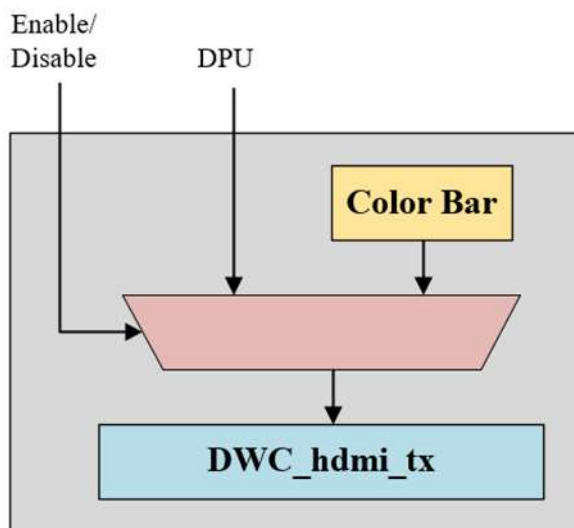
CEC 线路上的所有事务都由一个发起方和一个或多个跟随者组成。发起方发送消息结构和数据。跟随者 是任何数据的接收者，并负责设置任何确认位。

## 2.4.9 内置彩条

在 HDMI 系统中增加了一个内置的彩条模块进行测试。默认情况下，禁用内置的彩条功能。一旦用户启用它，有 7 种分辨率的彩条可供选择。

- 640\*480p@60Hz
- 720\*480p@60Hz
- 1280\*720p@24Hz
- 1280\*720p@60Hz

- 1920\*1080p@30Hz
- 3840\*2160p@30Hz
- 4096\*2160p@30Hz



图表 2-3 HDMI 系统中的彩条

## 2.5 使用

内置彩条编程模型。

请使用图表 2-4 所示的寄存器启用彩条功能。

- 寄存器名: HDMI\_COLOR\_BAR\_CFG
- 描述: 该寄存器用于启用或禁用内置的彩条功能，并在启用此功能时选择其中一个分辨率。
- 大小: 32 位
- 偏移量: 0xD0
- 存在: 是

图表 2-4 HDMI\_COLOR\_BAR\_CFG 寄存器字段说明

| 位段   | 名称   | 类型 | 说明                             |
|------|------|----|--------------------------------|
| 31:7 | RSVD | R  | 保留位 - 只读<br>复位值: 0x0<br>存在: 一直 |

| 位段  | 名称                | 类型  | 说明   |
|-----|-------------------|-----|--|
| 6:4 | RESOLUTION_SELECT | R/W | HDMI 彩条分辨率选择信号<br>3'b000: 640_480_60Hz<br>3'b001: 720_480_60Hz<br>3'b010: 1280_720_24Hz<br>3'b011: 1280_720_60Hz<br>3'b100: 1920_1080_30Hz<br>3'b101: 3840_2160_30Hz<br>3'b110: 4096_2160_30Hz<br>3'b111: 640_480_60Hz<br>复位值: 0x0<br>存在: 一直 |
| 3:1 | RSVD              | R   | 保留位 - 只读<br>复位值: 0x0<br>存在: 一直   |
| 0   | COLOR_BAR_ENA     | R/W | HDMI 彩条使能信号<br>0: 禁用彩条。<br>1: 启用彩条。<br>复位值: 0x0<br>存在: 一直  |

## 3 MIPI DSI

### 3.1 概述

MIPI DSI 模块由一个 MIPI DSI 主控制器和一个 4 通道 D-PHY TX 组成。

MIPI DSI 主机控制器遵循 MIPI 联盟规范 *MIPI Alliance Specification for Display Serial Interface (DSI), Version 1.2-16 June 2014*, 提供与 MIPI 联盟标准兼容的 PPI 接口, 用于连接 PHY 对接端。像素输入端提供一个与 MIPI 联盟标准 *MIPI Alliance Specification for Display Pixel Interface v2.00 (DPI-2)-15 September 2015* 兼容的 DPI 接口, 用于对接显示控制模块的输入。同时, 控制器提供标准的 AMBA APB2.0 接口, 用于控制 MIPI DSI 控制器和 D-PHY, 并提供标准的 DCS 命令的发送功能。

D-PHY TX 遵循 MIPI 联盟规范 *MIPI Alliance Specification for D-PHY, Version 1.2, 01 August 2014*。D-PHY TX 具有多达 4 个数据通道, 每个数据通道的最大速率为 2.5Gbps, 提供与 MIPI 联盟标准兼容的 PPI 接口, 用于连接外部 MIPI 控制器。

### 3.2 主要特性

MIPI DSI 主机控制器具有以下特点:

- 符合 MIPI 联盟标准
- 标准 DPI 接口, 用于接收输入显示图像
- 支持高达 4K@60fps 输入显示图像
- 支持各种 bpp 的 RGB 图像输入显示, 如 RGB101010
- 支持通过 AMBA APB 接口进行标准的 DCS 命令传输
- 通过非突发脉冲模式实现帧和行同步包的时序精确信令
- ECC 和校验和功能

MIPI DSI D-PHY TX 具有以下特点:

- 符合 MIPI 联盟标准
- 多达 4 个 D-PHY TX 数据通道
- 每个数据通道的最大速率为 2.5Gbps
- 通过数据通道 0 支持双向通信和逃避模式 (高达 10Mbps)
- 支持禁用 PLL 的超低功耗模式
- 支持传输结束包 (EoTP)

## 3.3 接口

图表 3-1 引脚描述表

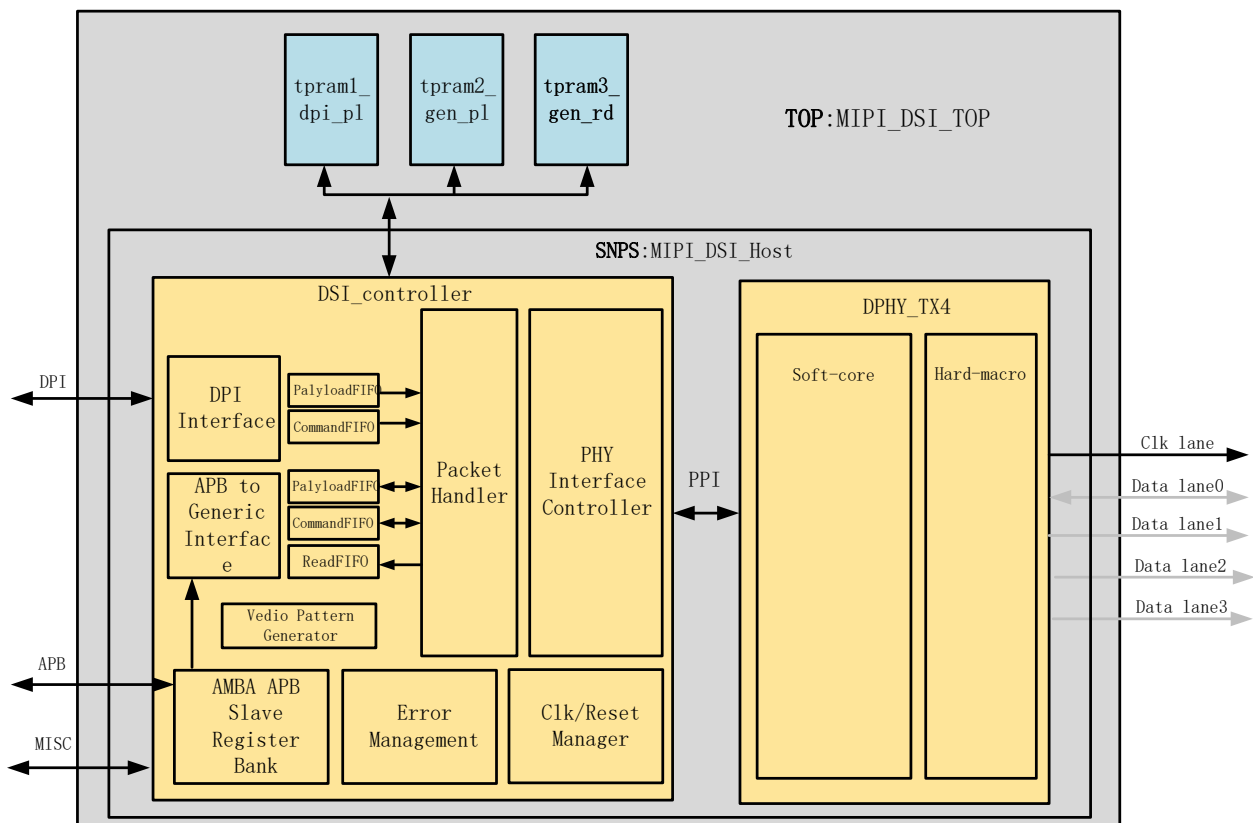
| 引脚名称   | 方向 | 宽度 | 说明            |
|--------|----|----|---------------|
| REXT   | IO | 1  | 参考电阻连接 PAD    |
| CLKP   | IO | 1  | 正 D-PHY 差分时钟线 |
| CLKN   | IO | 1  | 负 D-PHY 差分时钟线 |
| DATAP0 | IO | 1  | 正 D-PHY 差分数据线 |
| DATAN0 | IO | 1  | 负 D-PHY 差分数据线 |
| DATAP1 | IO | 1  | 正 D-PHY 差分数据线 |
| DATAN1 | IO | 1  | 负 D-PHY 差分数据线 |
| DATAP2 | IO | 1  | 正 D-PHY 差分数据线 |
| DATAN2 | IO | 1  | 负 D-PHY 差分数据线 |
| DATAP3 | IO | 1  | 正 D-PHY 差分数据线 |
| DATAN3 | IO | 1  | 负 D-PHY 差分数据线 |

## 3.4 功能描述

### 3.4.1 功能概述

MIPI DSI 的整体功能框图如[图表 3-2](#) 所示。





图表 3-2 MIPI DSI 功能框图

MIPI DSI 主机的主要模块如下：

- DPI 接口：捕获数据和控制信号，并将它们传送到用于视频控制信号的 FIFO 和用于像素数据的另一个 FIFO。当处于视频模式时，这些数据被用来构建视频数据包。
- 寄存器组可通过标准 AMBA-APB 从接口访问，提供对 MIPI DSI 寄存器的访问以进行配置和控制。还有一个完全可编程的中断发生器，用于通知系统某些事件。
- PHY 接口控制器负责管理 D-PHY PPI 接口。它确认当前操作并启用低功耗传输/接收或高速传输。它还在可用的 D-PHY 通道之间执行数据分割，以实现高速传输。
- 数据包处理程序调度链路内的活动。它根据所使用的视频传输模式（突发模式或具有同步脉冲或同步事件的非突发模式）执行多种功能。它构建长或短数据包，生成相应的 ECC 和 CRC 代码。该块还执行以下功能：
  - 包接收
  - 通过检查 ECC 来验证包头
  - 单比特错误的包头校正和通知
  - 终止接收
  - 多包头错误通知
- APB-to-Generic 块将 APB 操作桥接到持有 Generic 命令的 FIFO 中。
- 错误管理通知并监控 DSI 链路上的错误情况。它控制用于确定是否发生超时情况的计时器，执行内部软复位并触发中断通知。

### 3.4.2 DPI 接口

DPI 接口遵循 MIPI DPI-2 规范，像素数据总线宽度高达 30 位。它用于在视频模式下传输信息，其中从主机处理器到外设的传输采用实时像素流的形式。

可以对 DPI 接口进行配置，以增加灵活性并促进该接口在多个系统中的正确使用。这些配置选项如下：

- 极性控制：所有的控制信号都是可编程的，可根据系统要求改变极性。
- 核复位后，DPI 等待第一个 VSYNC 有效转换开始信号采样，包括像素数据，从而避免在帧中间开始传输图像数据。
- 为避免 FIFO 下溢和溢出，假设在任何时候都能接收到配置的像素数。即使 dpidataen 引脚处于活动状态的时间超过或少于所需的时间，也会发生这种情况。
- 为了保持内存在数据包调度方面的有序性，每个数据包的像素数参数被用来分隔不同视频数据包的内存空间。

当关闭显示器时，DPI 视频必须在命令发出后的一帧内保持活动状态。这确保了在实际禁用 DPI 接口上的视频生成之前正确传输命令。

可以在不影响当前帧的情况下动态更新 DPI 配置。这是在阴影寄存器的帮助下完成的。该功能由 VID\_SHADOW\_CTRL 寄存器控制。

### 3.4.3 APB 从通用接口

APB 从接口允许在命令模式下传输通用信息。通过该接口发送的命令不受 DCS 规范的约束，并且可以包括 DSI 规范中描述为特定于制造商的通用命令。

### 3.4.4 命令的传输

MIPI DSI 在视频模式下支持高速和低功耗的命令传输。MIPI DSI 使用消隐或低功耗（BLLP）周期来传输通过 APB 通用接口插入的命令。

- 如果 VID\_MODE\_CFG 寄存器的 lp\_cmd\_en 位为 0，则在视频模式下高速发送命令。在这种情况下，MIPI DSI 自动确定每个命令可以发送的区域，不需要编程或计算。
- MIPI DSI 可以配置为在高速视频模式传输期间发送低功耗（LP）命令。要启用该功能，请将 VID\_MODE\_CFG 寄存器的 lp\_cmd\_en 位设置为 1。在这种情况下，有必要计算在 LP 模式下向水平前沿（HFP）、垂直同步有效（VSA）、垂直后沿（VBP）和垂直前沿（VFP）区域传输命令的可用时间，以字节数为单位。
- MIPI DSI 也可以发送读命令，如果在帧的最后一行发出读命令，则 edpihalt 信号生效并保持生效状态，直到读命令正在进行。在此期间应停止视频传输。
- 为了降低 D-PHY 的功耗，MIPI DSI 不以高速模式传输时，允许时钟通道进入 LP 模式。

### 3.4.5 虚拟通道

MIPI DSI 支持为每个接口选择虚拟通道 (VC)。使用多个 VC，当每个显示器具有不同的 VC 标识符时，系统可以同时寻址多个显示器。

### 3.4.6 视频模式图案产生器

视频模式图案生成器允许在没有任何刺激的情况下传输水平/垂直彩条和 D-PHY BER 测试图案。

### 3.4.7 超时计时器和错误控制

外设可能不会立即正确响应某些接收到的数据包。例如，一个外设接收到一个读请求，但是由于它的体系结构暂时不能访问 RAM。这可能是因为面板正在刷新，需要一些时间来响应。在这种情况下，设置超时以确保主机等待足够长的时间，以便设备能够在接收新数据或正确响应新请求之前处理以前的数据。

INT\_ST0 和 INT\_ST1 寄存器与错误条件报告相关联。这些寄存器可以触发中断引脚，通知系统错误发生。

### 3.4.8 PHY PLL 可编程性

数据速率由 PLL 输出时钟相位频率的两倍给出：数据速率 (Gbps) = PLL Fout (GHz) \* 2。输出频率 (Fout) 是输入参考频率 (Fclk\_in) 和倍频/分频比的函数。可以通过以下方式确定：Fout = (M/N) \* fclk\_in。

其中：

- M：反馈倍频比
- N：输入分频比

## 3.5 使用

### 3.5.1 编程流程

MIPI DSI 编程流程如下：

1. 复位 APB 接口。
2. 配置 DPI。
3. 配置 D-PHY 参数。
4. 唤醒核和 D-PHY。
5. 等待 D-PHY 上电。
6. 如果改变视频模式，配置 DPI。

### 3.5.2 视频模式图案生成器

发送测试图案的 MIPI DSI 编程顺序如下：

1. 选择视频模式。
2. 配置颜色编码。
3. 配置视频帧。
4. 配置视频图案生成器。

### 3.5.3 停止冲突场景处理

DSI 有一个中断用于指示错误场景，当 `edpihalt` 被拉高时，DPI 在下一帧中仍然输入。当这样的场景发生时，将触发此中断。软件需要复位 DSI 并重新初始化 DSI 以继续使用 DSI。

产生错误中断的 `dpivsync` 极性可以通过 `vo_subsys sysreg` 的寄存器 `vsync_pol` 来配置，中断掩码可以通过 `vo_subsys sysreg` 的寄存器 `halt_err_mask` 来配置，而中断错误状态可以通过读取 `vo_subsys sysreg` 的寄存器 `halt_err` 获得。