T-HEAD

TH1520 Video Output User Manual

Revision	1.0.0
Security	Secret
Date	2023-08-26

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

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

Revisions

Rev	Description	Author(s)	Date
V1.0.0	Initial version	T-Head	2023-08-26

Contents

Figures & Tables

List of Abbreviations

Abbreviations	Full Spelling	Chinese Explanation
CSC	Consumer Electronics Control	消费电子控制
DHCP	High-bandwidth Digital Content Protection	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 Overview

The DPU is used to control image display. It converts the resolution of the image in the DDR to match that of the display device and feeds the pixel stream to the video interface (MIPI/HDMI) according to the timing required by the display device, and the video interface drives the display device to display the image. The main functions of the DPU include layer scale, layer rotation, layer overlay, and overlay image processing (including Gamma, Dither and Color).

The DPU supports a maximum of six image layers, two cursor layers input at the same time, and two-channel display in parallel. Each channel supports a maximum output resolution of 4K@60. The block diagram of DPU_TOP is shown in Figure & Table 1-1:

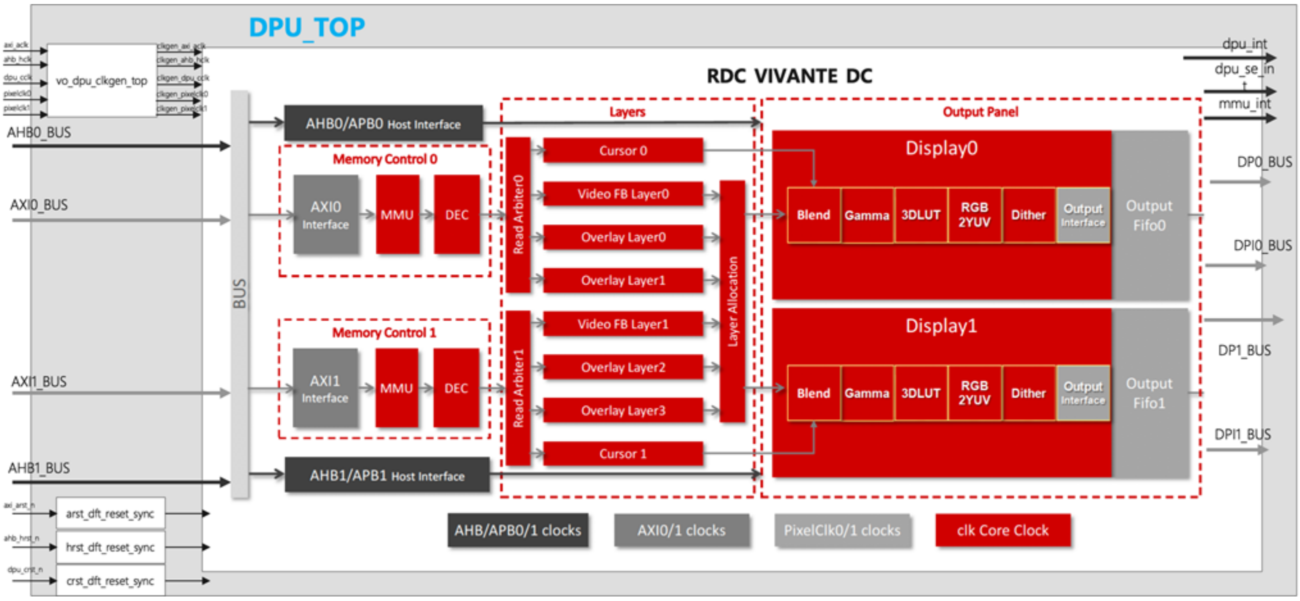


Figure & Table 1-1 DPU_TOP block diagram

1.2 Main Features

The DPU has the following features:

- Clock and Reset
  - AXI bus clock ACLK, the default frequency is 792M, its frequency depends on SoC configuration.
  - AHB configuration clock HCLK, the default frequency is 396M, its frequency depends on SoC configuration.
  - DPU Core clock, pixel processing clock, the default frequency is 594M, its frequency depends on the resolution of the current display device.

  - Pixelclk0/1 clock, pixel output clock, controlled by an independent PLL, the specific frequency depends on the resolution of the display device. Its calculation relationship and the relationship with the Core clock, see Section 1.4.1.
- AXI Bus
  - Data bus width 128bit
  - Address bus width 32bit, which can be extended to 40bit using the 8bit axi_user.
  - Burst features: In non-compression and non-MMU mode, the read trans types of the DPU include 16/128/256byte. In compression and MMU mode, the read trans types of the DPU include 6/32/64/96/128/160/192/ 224/256B.
  - Cacheable/re-order/protection transmission is not supported.
  - The read channel supports out-of-order and interleave.
  - Supports INC burst.
  - The width of AXI ARID is 5bit.
  - QoS features: It supports two-levels of QoS adjustment, High and Low. The corresponding level can be set through static configuration of QoS by the software. Dynamic adjustment of line buffer watermark is supported. When the number of pixels in the line buffer is lower than the watermark configured by the software, the DPU will issue high QoS. When the number of pixels in the line buffer is greater than the watermark configured by the software, the QoS of the DPU will be reduced to low level.
  - The maximum outstanding capability supported by the DPU is 64.
- Input Image Format
  - RGB: A2R10G10B0/ARGB8888/XRGB8888/RGB565/ARGB4444/XRGB4444
  - YUV: YUY2/UYVY/NV12/NV16/P010/YV12
  - Supports tile and linear memory formats. See Section 1.4.3 for details.
- Image Compression
  - Supports DEC400 compression. See DEC400 section description for details.
- Image Overlay
  - Supports eight mix modes defined by proter_duff: clr/src/dst/src_over/dst_over/ src_in/dst_in/src_out.
  - Supports global Alpha replacement.
- Image Scale
  - Horizontal scale uses a 3rd/5th order filter, the factor is configurable, and the scale factor is 15.16.
  - Vertical scale supports a 3rd order filter, the factor is configurable, and the scale factor is 15.16.
- Image Rotation
  - Supports 90/180/270/xflip/yflip/xyflip scale modes.
  - ARGB32bit: Only supports superTileY4x8; ARGB16bit: Only supports SuperTile8x8.
  - YUY2/UYVY/P010 TILE8X8_XMAJOR
  - Other formats do not support rotation.
- Gamma

  - Gamma table can be configured by software.
- Dither: Uses a software-configured dither table.
- Color: Uses a 17x17x17 color correction coefficient table.
- Output Image Format
  - DPI interface: Only supports RGB format, 30/24/18/16bit. See Section 1.4.4 for detailed formats.
  - DP interface: Supports RGB and YUV formats. See Section 1.4.4 for detailed formats.
- Interrupt Features
  - DC_INT: DPU output interrupt in the REE region. See Section 1.4.5 for details.
  - DC_SE_INT: DPU output interrupt in the TEE region.
  - MMU_INTR: DPU embedded MMU interrupt.

1.3 Interface

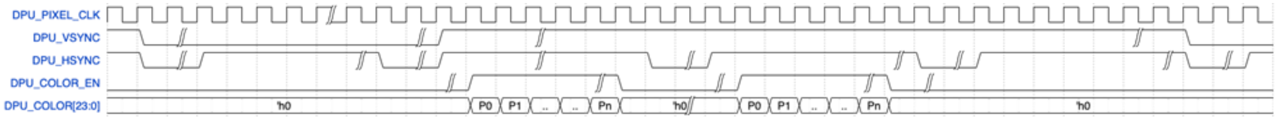


Figure & Table 1-2 Standard DPI interface timing diagram

Figure & Table 1-3 DPI interface definition

Pin Name	Direction	Width	Description
DPU_PIXELCLK	O	1	TTL output pixel clock
DPU_VSYNC	O	1	TTL output frame sync signal
DPU_HSYNC	O	1	TTL output row sync signal
DPU_COLOR_EN	O	1	TTL output pixel enable signal
DPU_COLOR_[0~23]	O	24	TTL output pixel data signal

The DPI interface of the DPU is compatible with DPI-2, but only supports RGB output. The relationship between the DPI interface and the standard DPI-2 interface is as follows:

CM(DPI)	<- ->	(not supported)
DE(DPI)	<- ->	DPU_COLOR_EN
Hsync(DPI)	<- ->	DPU_HSYNC
Vsync(DPI)	<- ->	DPU_VSYNC
SD(DPI)	<- ->	(not supported)
PCLK(DPI)	<- ->	DPU_PIXELCLK

The DPI interface timing diagram is as follows:

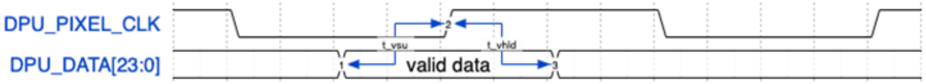


Figure & Table 1-4 DPI interface timing diagram

Figure & Table 1-5 TTL interface timing parameters

Symbol	Parameter	Condition	Standard Mode		Unit
			Min	Max	
Pixelclk	Pixel clock frequency		10	74.25	mHz
t_vsu	Pixel value setup time		1	--	ns
t_vhlod	Pixel value hold time		0.7	--	ns

1.4 Function Description

1.4.1 Clock Features

The DPU includes clocks of AXI bus and AHB bus, as well as core clock and pixel clock.

The bus clocks are determined by the SoC. The default frequency of the data bus is 792M and the default frequency of the configuration bus is 396M. The core clock and the pixel clock of the DPU are determined by the current output resolution.

The theoretical basis of each clock select of the DPU is as follows:

- Bus Clock ACLK Select
  - Image bandwidth calculation based on image resolution and frame rate: BW(MB/s) = H_act*V_act*frame_rate* pixel_bytes/(1000*1000)
  - The DPU supports overlay of multiple layers, so the bandwidth of the AXI bus is required to be greater than the sum of the bandwidth required by multiple layers. For calculation of DPU AXI bus bandwidth, see Section 1.4.2.
- DPU Processing Clock CCLK Select
  - Since the DPU can read multiple layers at the same time, the DPU processing clock is determined by the maximum resolution layer.
  - The processing capability of the DPU is 1 clock to process 1 pixel (1T1P), so it is required: dpu_cclk >= H_act*V_act*frame_rate. Hact and Vact refer to the row and column effective pixels of the largest layer.
- The DPU pixel clock is determined by the screen parameters of the current display device.
  - Pixelclk = h_total*v_total*frame_rate
  - H_total= H_act+H_blanking, V_total= V_act+V_blanking

The select of dpu_cclk and dpu_pixel0/1clk at typical resolutions are shown in the table below.

Figure & Table 1-6 Clock configuration for typical resolutions of the DPU

Resolution	Frame Rate	DPU_ACLK	DPU_CCLK	DPI_PIXEL0/1_CLK
3840x2160	60	792	792	594
4096x2016	30	500	500	297
1920x1080	60	400	400	148.5

Resolution	Frame Rate	DPU_ACLK	DPU_CCLK	DPI_PIXEL0/1_CLK
1080x720	60	300	300	74.25
640x480	60	200	200	25.175

The pixel clock is configured by the DPU-specific PLL clock. The architecture of the DPU-specific PLL is as follows:

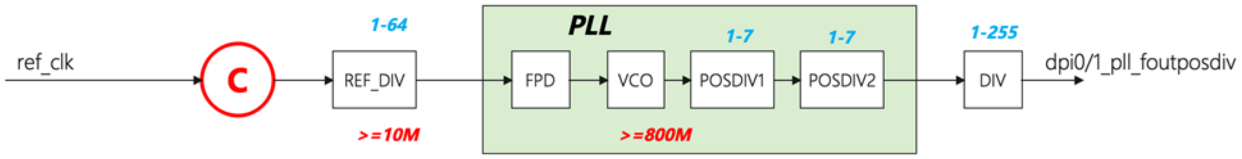


Figure & Table 1-7 DPU pixel PLL architecture diagram

Note the following for configuring the DPU:

- FOUTVCO = (FREF / REFDIV) × (FBDIV + FRAC/224)
- FOUTPOSTDIV = FOUTVCO / POSDIV1 / POSTDIV2
- Requires FREF/REF_DIV>=10M; FOUTVCO >= 800M
- Requires POSDIV1 >= POSDIV2

The following table is a reference list of PLL parameter configuration corresponding to typical output resolutions.

Figure & Table 1-8 Reference to configuration parameters corresponding to typical output resolutions

Resolution	Frame Rate	Pixel clk	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 Bus Features

Main features of the DPU AXI bus:

- Read and write two channels, the bus width is 128bit.
- Burst features:
  - In non-compression and non-MMU mode, the read trans types of the DPU include: 16/128/256byte.
  - In compression and MMU mode, the read trans types of the DPU include: 6/32/64/96/128/160/192/ 224/256B.

- The address output by the DPU core is 32bit, which can be extended to 40bit using the 8bit axi_user.
- Cacheable/re-order/protection transmission is not supported.
- The read channel supports out-of-order and interleave.
- Supports INC burst.
- The width of AXI ARID is 5bit.
- QoS features:
  - Supports two levels of QoS adjustment, High and Low. The corresponding levels can be statically configured by software.
  - QoS supports dynamic adjustment of the line buffer watermark. When the number of pixels in the line buffer is lower than the watermark configured by the software, the DPU will issue a high QoS. When the number of pixels in the line buffer is greater than the watermark, the QoS of the DPU will be reduced to low level.
- The maximum outstanding capability supported by the DPU is 64.

The main trans features output by the DPU in different image formats are as follows:

Figure & Table 1-9 Main Trans features output by the DPU in different image formats

Format/Type	**Burst Size (in byte)**	
	Compression	**Non-Compression**
32bit(A8R8G8B8,X8R8G8B8,A2R10G10B10)	128	128
16bit(A4R34G4B4, X4R4G4B4,A1R5G5B5,X1R5G5B5,R5G6B5)	\	128
UYVY	128	128
YUY2	128	128
P010	256	128
YV12	\	128
NV12	256	128
NV16	256	128

Figure & Table 1-10 IDs output by DPU AXI

Read Request	**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
Write Requests	**AWID ID**
Write back	0x0

Bus bandwidth calculation: The bandwidth of the DPU AXI bus is determined by the clock frequency freq(aclk) of the DPU AXI bus, the transmitted trans_size, the average read data latency (ns) and outstanding. It is calculated as follows:

BW(MB/s) = freq(aclk) * tran_size* outstanding/latency

In non-compression mode, the typical trans_size of the DPU is 128bits. The outstanding value of the DPU can be set by software and the maximum is 64. The latency is related to the current scene. When estimating the DPU bandwidth, it is assumed that the average latency of the DPU in the current scene is 1000.

The dynamic adjustment of QoS can improve the latency to achieve the purpose of increasing the bandwidth.

1.4.3 Input Image Format

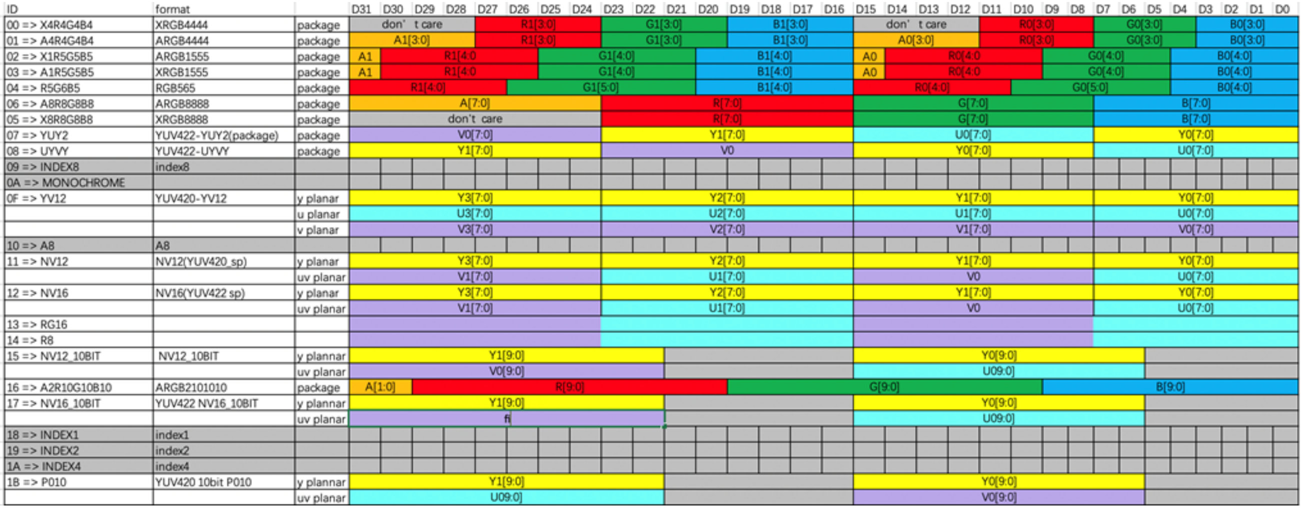
Different formats of the DPU have the following alignment requirements for image size, first address and stride:

Figure & Table 1-11 Alignment requirements for DPU memory formats

Linear/Tile	Burst Size (Bytes)	Format	Width x Height Alignment (in pixels)	Stride Alignment (Bytes)	Base Address (Bytes)
Linear	128	Planar YUV/Semi-planar YUV	2x2	128	128
	128	YUY2/UYVY	2x1	128	128
	128	RGB	\	128	128
	256	Planar YUV/Semi-planar YUV	2x2	256	256
	256	YUY2/UYVY	2x1	256	256

Linear/Tile	Burst Size (Bytes)	Format	Width x Height Alignment (in pixels)	Stride Alignment (Bytes)	Base Address (Bytes)
	256	RGB	\	256	256
SuperTileX8x8	128	16bpp RGB	64x64	128	128
SuperTileX8x4/SuperTileY4x8	128	32bpp RGB	64x64	128	128
8x8 tile	128	YUY2/UYVY	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
8x4 tile	128	NV12-UV planar	8x8	32	128
	128	P010-UV planar	8x8	32	128
	256	NV12-UV planar	8x8	64	256
	256	P010-UV planar	8x8	64	256

The pixel memory placement of different image formats is as follows:

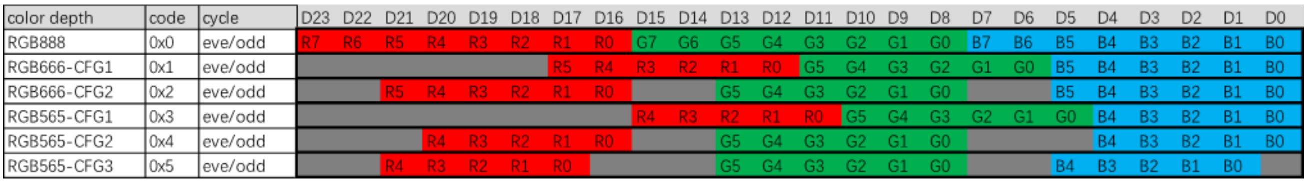


ID	format		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
00 => X4R4G4B4	XRGB4444	package	don’ t care				R1[3:0]				G1[3:0]				B1[3:0]				don’ t care				R0[3:0]				G0[3:0]				B0[3:0]			
01 => A4R4G4B4	ARGB4444	package	A1[3:0]				R1[3:0]				G1[3:0]				B1[3:0]				A0[3:0]				R0[3:0]				G0[3:0]				B0[3:0]			
02 => X1R5G5B5	ARGB1555	package	A1	R1[4:0					G1[4:0]					B1[4:0]					A0	R0[4:0					G0[4:0]					B0[4:0]				
03 => A1R5G5B5	XRGB1555	package	A1	R1[4:0					G1[4:0]					B1[4:0]					A0	R0[4:0					G0[4:0]					B0[4:0]				
04 => R5G6B5	RGB565	package	R1[4:0]					G1[5:0]						B1[4:0]					R0[4:0]					G0[5:0]						B0[4:0]				
06 => A8R8G8B8	ARGB8888	package	A[7:0]								R[7:0]								G[7:0]								B[7:0]							
05 => X8R8G8B8	XRGB8888	package	don't care								R[7:0]								G[7:0]								B[7:0]							
07 => YUY2	YUV422-YUY2(package)	package	V0[7:0]								Y1[7:0]								U0[7:0]								Y0[7:0]							
08 => UYVY	YUV422-UYVY	package	Y1[7:0]								V0								Y0[7:0]								U0[7:0]							
09 => INDEX8	index8																																	
0A => MONOCHROME																																		
0F => YV12	YUV420-YV12	y planar	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							
		u planar	U3[7:0]								U2[7:0]								U1[7:0]								U0[7:0]							
		v planar	V3[7:0]								V2[7:0]								V1[7:0]								V0[7:0]							
10 => A8	A8																																	
11 => NV12	NV12(YUV420_sp)	y planar	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							
		uv planar	V1[7:0]								U1[7:0]								V0								U0[7:0]							
12 => NV16	NV16(YUV422 sp)	y planar	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							
		uv planar	V1[7:0]								U1[7:0]								V0								U0[7:0]							
13 => RG16																																		
14 => R8																																		
15 => NV12_10BIT	NV12_10BIT	y plannar	Y1[9:0]																Y0[9:0]															
		uv planar	V0[9:0]																U09:0]															
16 => A2R10G10B10	ARGB2101010	package	A[1:0]		R[9:0]										G[9:0]										B[9:0]									
17 => NV16_10BIT	YUV422 NV16_10BIT	y plannar	Y1[9:0]																Y0[9:0]															
		uv planar	fi																U09:0]															
18 => INDEX1	index1																																	
19 => INDEX2	index2																																	
1A => INDEX4	index4																																	
1B => P010	YUV420 10bit P010	y plannar	Y1[9:0]																Y0[9:0]															
		uv planar	U09:0]																V0[9:0]															

Figure & Table 1-12 Pixel memory placement format

1.4.4 Output Image Format

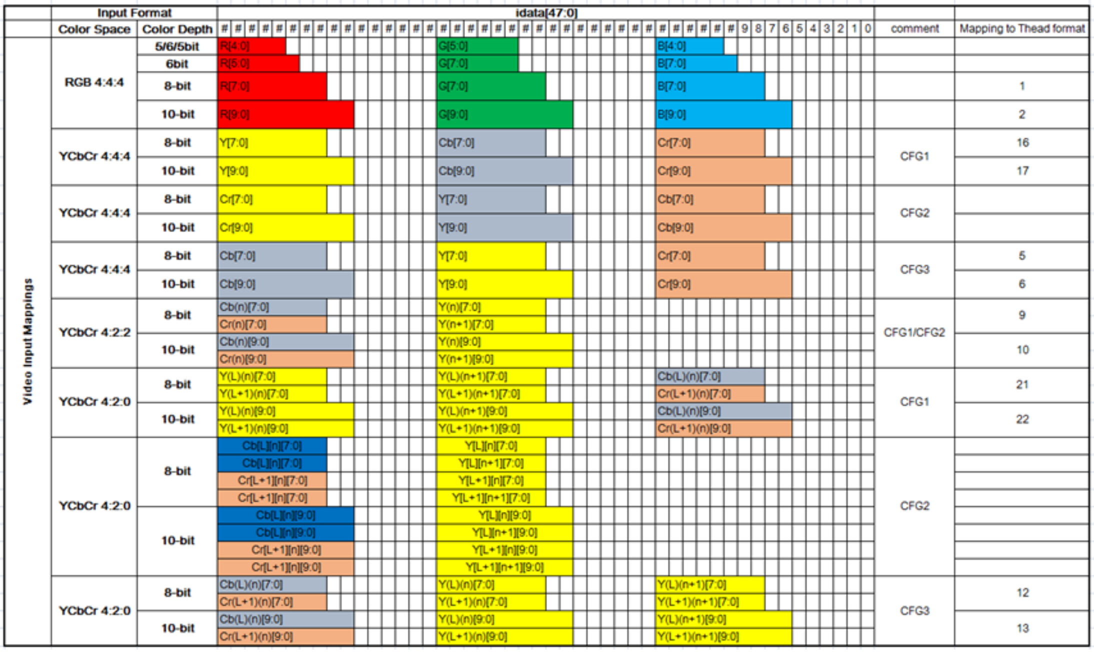
The DPU supports two kinds of interface output: DPI and DP formats. DPI format output to DSI to drive MIPI Panel or drive TTL Panel through GPIO, and DP interface output to HDMI.
The DPI output pin mapping format is shown in Figure & Table 1-13:



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							R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
RGB666-CFG2	0x2	eve/odd			R5	R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0			B5	B4	B3	B2	B1	B0
RGB565-CFG1	0x3	eve/odd									R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
RGB565-CFG2	0x4	eve/odd				R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0				B4	B3	B2	B1	B0
RGB565-CFG3	0x5	eve/odd			R4	R3	R2	R1	R0				G5	G4	G3	G2	G1	G0			B4	B3	B2	B1	B0	

Figure & Table 1-13 DPI output pin mapping diagram

The DP output pin mapping format is shown in Figure & Table 1-14:



	Input Format		idata[47:0]				
	Color Space	Color Depth				comment	Mapping to Thead format
Video Input Mappings	RGB 4:4:4	5/6/5bit	R[4:0]	G[5:0]	B[4:0]		
		6bit	R[5:0]	G[7:0]	B[7:0]		
		8-bit	R[7:0]	G[7:0]	B[7:0]		1
		10-bit	R[9:0]	G[9:0]	B[9:0]		2
	YCbCr 4:4:4	8-bit	Y[7:0]	Cb[7:0]	Cr[7:0]	CFG1	16
		10-bit	Y[9:0]	Cb[9:0]	Cr[9:0]		17
	YCbCr 4:4:4	8-bit	Cr[7:0]	Y[7:0]	Cb[7:0]	CFG2	
		10-bit	Cr[9:0]	Y[9:0]	Cb[9:0]		
	YCbCr 4:4:4	8-bit	Cb[7:0]	Y[7:0]	Cr[7:0]	CFG3	5
		10-bit	Cb[9:0]	Y[9:0]	Cr[9:0]		6
	YCbCr 4:2:2	8-bit	Cb(n)[7:0]	Y(n)[7:0]		CFG1/CFG2	9
			Cr(n)[7:0]	Y(n+1)[7:0]			
		10-bit	Cb(n)[9:0]	Y(n)[9:0]			10
			Cr(n)[9:0]	Y(n+1)[9:0]			
	YCbCr 4:2:0	8-bit	Y(L)(n)[7:0]	Y(L)(n+1)[7:0]	Cb(L)(n)[7:0]	CFG1	21
			Y(L+1)(n)[7:0]	Y(L+1)(n+1)[7:0]	Cr(L+1)(n)[7:0]		
		10-bit	Y(L)(n)[9:0]	Y(L)(n+1)[9:0]	Cb(L)(n)[9:0]		22
			Y(L+1)(n)[9:0]	Y(L+1)(n+1)[9:0]	Cr(L+1)(n)[9:0]		
	YCbCr 4:2:0	8-bit	Cb[L][n][7:0]	Y[L][n][7:0]		CFG2	
			Cb[L][n][7:0]	Y[L][n+1][7:0]			
			Cr[L+1][n][7:0]	Y[L+1][n][7:0]			
			Cr[L+1][n][7:0]	Y[L+1][n+1][7:0]			
		10-bit	Cb[L][n][9:0]	Y[L][n][9:0]			
			Cb[L][n][9:0]	Y[L][n+1][9:0]			
			Cr[L+1][n][9:0]	Y[L+1][n][9:0]			
			Cr[L+1][n][9:0]	Y[L+1][n+1][9:0]			
	YCbCr 4:2:0	8-bit	Cb(L)(n)[7:0]	Y(L)(n)[7:0]	Y(L)(n+1)[7:0]	CFG3	12
			Cr(L+1)(n)[7:0]	Y(L+1)(n)[7:0]	Y(L+1)(n+1)[7:0]		
		10-bit	Cb(L)(n)[9:0]	Y(L)(n)[9:0]	Y(L)(n+1)[9:0]		13
			Cr(L+1)(n)[9:0]	Y(L+1)(n)[9:0]	Y(L+1)(n+1)[9:0]		

Figure & Table 1-14 DP output pin mapping diagram

1.4.5 Interrupt

The DPU outputs three interrupts: MMU_INTR, DC_INTR, and DC_SE_INTR interrupts.

The DPU has a separate MMU interrupt, which is controlled by gcregSeMMUIntrEnbl (0x10044). Interrupt information can be got through gcregSeMMUIntrAcknowledge (0x10048). For the defined interrupt events, refer to Figure & Table 1-15 DPU interrupt event list.

DEC400 and DPU core share a DC_INTR. When CPU gets the interrupt, it can get what kind of event happened by checking AQIntrAcknowledge and gcregAHBDECIntrAcknowledgeEx2.

DC_SE_INTR is DC_INTR in the TEE region. When the interrupt is triggered, CPU gets what kind of event happened by checking gcregSeHiIntrAcknowledge and gcregAHBSeDECIntrAcknowledgeEx2.

The common interrupts of the DPU are as follows:

- AXI bus error intr: The response returned by the AXI read channel is not OKEY.
- Flush done intr: DEC400 flush done interrupt. DEC400 completes the decompression of 32 streams through a hardware cache. When using hardware flush done, each bit of 31:0 of AcknowledgeEx2 corresponds to the flush done of each corresponding stream. When using software flush done, all bit0 corresponds to the flush done of all client0 streams, and bit1 corresponds to the flush done of all client1 streams, and in turn to max client.
- Frame done intr: The interrupt is pulled high at the last pixel of the valid area of each frame, and CPU is automatically cleaned when AQIntrAcknowledge is read. This interrupt is used for

configuration of the next frame operation. This configuration takes effect after the next vsync arrives, and the DMA also performs memory access after the next vsync arrives.

- Layer allocation intr: Each layer can only be used to TEE or REE. If both TEE and REE registers are configured to be valid, the DPU will use the TEE configuration and generate this interrupt.
- Reset done interrupt: Indicates hardware reset done interrupt.

All the events corresponding to DPU internal interrupts are shown in Figure & Table 1-15:

Figure & Table 1-15 DPU interrupt event list

Bit	AQIntrAcknowledge (0x0004) for Non-secure Memory	gcregSeHiIntrAcknowledge (0x4001) for Secure Memory	gcregSeMMUIntrAcknowledge (0x4012) for MMU
0	DC Panel0 interrupt	DC Panel0 interrupt	MMU0 slave not present
1	DC Panel1 interrupt	DC Panel1 interrupt	MMU0 client out of bounds
2	Secure reset done interrupt	Secure reset done interrupt	MMU0 size out of bounds
3	Video0 non-secure configuration conflict	Video0 non-secure configuration conflict	MMU0 read security exception
4	Overlay0 non-secure configuration conflict	Overlay0 non-secure configuration conflict	MMU0 write security exception
5	Overlay1 non-secure configuration conflict	Overlay1 non-secure configuration conflict	MMU0 page not present
6	Video1 non-secure configuration conflict	Video1 non-secure configuration conflict	MMU0 page write to unwritable page
7	Overlay2 non-secure configuration conflict	Overlay2 non-secure configuration conflict	MMU0 read secure memory violation
8	Overlay3 non-secure configuration conflict	Overlay3 non-secure configuration conflict	MMU1 slave not present
9	Cursor0 non-secure configuration conflict		MMU1 client out of bounds
10	Cursor1 non-secure configuration conflict		MMU1 size out of bounds
11	DC Panel0 data underflow		MMU1 read security exception
12	DC Panel1 data underflow		MMU1 write security exception
13			MMU1 page not present
14			MMU1 page write to unwritable page
15			MMU1 read secure memory violation

Bit	AQIntrAcknowledge (0x0004) for Non-secure Memory	gcregSeHiIntrAcknowledge (0x4001) for Secure Memory	gcregSeMMUIntrAcknowledge (0x4012) for MMU
16-29			
30	AXI Bus1 error	AXI Bus1 error	
31	AXI Bus0 error	AXI Bus0 error	

NOTE

Figure & Table 1-15 does not list the interrupt events gcregAHBDECIntrAcknowledgeEx2 DEC TEE and gcregAHBSeDECIntrAcknowledgeEx2 DEC REE. The definitions of these two interrupts have been explained above.

1.4.6 Working Mode

The downstream connections of the DPU are as follows: Display channel 0 can only drive mipi_dsi0 and GPIO pad. Panel 1 can drive either HDMI (via DP), or mipi_dsi0, mipi_dsi1 and GPIO_PAD (via DPI interface). The three display modes corresponding to this connection relationship are as follows:

- Dual screen different display: The different display means that the two display channels do not interfere with each other, and different PLLs provide pixel clocks for each. In this case, note that one layer cannot be used on two display channels.
- Dual screen same display: The content displayed on the two screens is the same. At this time, the output of DPI1 needs to be switched to the output interface of channel 0 through mux. Pixelclock0 is not necessary to be configured.
- Dual screen partially same display: The dual screen partially same display means that some layers of the two channels have the same content. This mode needs to satisfy two constraints:
  - Channel 0 uses the pixel clock of Channel 1
  - The content of the same layer means that the software assigns the memory address of the same image to different layers, which is still a different layer for the hardware.

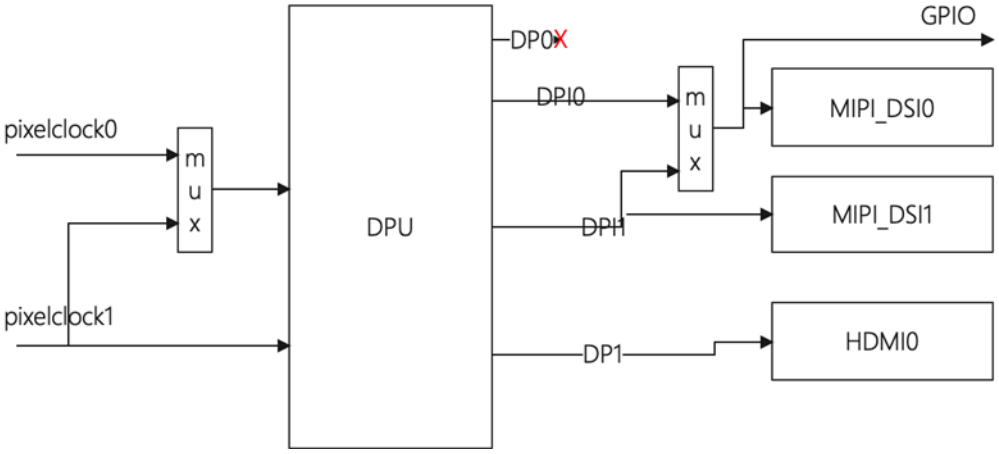


Figure & Table 1-16 DPU output device connection diagram

1.5 Usage

The software configuration process of the DPU consists of two parts: initialization configuration and inter-frame configuration update.

The initialization configuration includes clock configuration and pixel stream output interface configuration (MIPI/HDMI configurations are not reflected in this document, refer to the user guides of relevant modules). The DPU core configuration process is as follows:

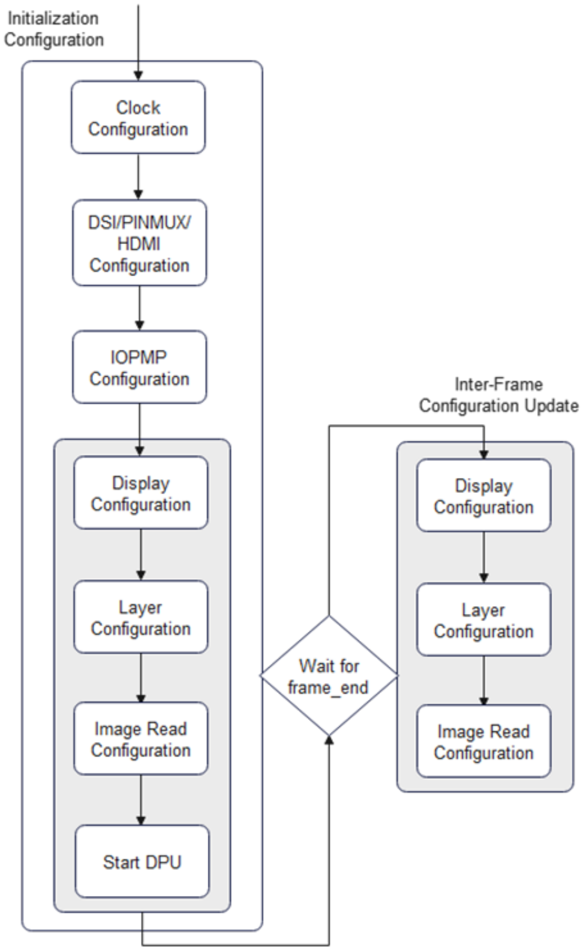


Figure & Table 1-17 DPU configuration flow chart

NOTE

Video output interface (DSI/PINMUX/HDMI) configuration, IOPMP configuration, and MMU configuration are not covered in this document.

1.5.1 Clock and Reset Configuration

1.5.1.1 Reset Process

1.5.1.1.1 Normal Reset Process

1. Turn on the reset of the VO subsystem, which is the release state by default, vo_rst_n = AO_RSTGEN_BASE+0x30.
2. Release the reset of DPU: soft reset control, corresponding to VOSYS_SYSREG_APBIF_APB3S_BADDR+0x4[0:2], see the corresponding register map for details.
3. Release soft_reset of AQHiClockControl.

The reset of the DPU is required to meet the following timing: The reset time is required to hold at least 32 clock cycles, and the configuration can only be performed after the reset is released for 128 clock cycles.

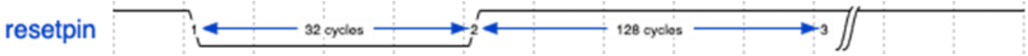


Figure & Table 1-18 DPU reset timing diagram

1.5.1.1.2 Abnormal Reset Process

After the DPU is abnormal, the system needs to be reset. You need to check the AXI_REQ_PENDING_FLAG value of AQAxiConfig. If the value is 0, reset the system according to the normal reset process. If the value is not 0, keep waiting. If having waited for a long time and the value is still not 0, the subsystem reset or chip reset is required. The subsystem reset and chip reset are not covered in this document. The DPU does not support resetting only the core domain, so the DPU needs to be reconfigured after it is reset.

1.5.1.2 Clock Configuration

1. Determine the PLL frequency according to the current working mode and the output resolution, and set the PLL frequency according to Section 1.4.1 description of PLL configuration methods.
2. Turn on the clock of the VO subsystem, which is on by default. The corresponding register is as follows: vo_clk_gen= AO_CLKGEN_BASE+0x130.
3. Turn on the reset and clock of the DPU subsystem. The DPU clock is off by default. The corresponding register is as follows: VOSYS_SYSREG_APBIF_APB3S_BADDR+0x50, see the corresponding register map for details.
4. Turn on the DPU internal clock and dynamic clock switch, see Section 1.4.1 description of DPU internal clock control.

1.5.2 DPU Configuration

This section mainly configures the REE region and the configuration method of the TEE region is similar to that of the REE region.

1.5.2.1 Layer Configuration

The DPU has two video layers, four overlays, and two cursor layers. Select the layer to be used for configuration according to the current application scenario. The specific configuration is as follows:

1. Configure properties of each layer: dcregFrameBufferConfig0~1, dcregOverlayConfig0~3. The main configuration information includes: transparency configuration (TRANSPARENCY), rotation angle configuration (RO_ANGLE), memory format configuration, color channel swizzle (SWIZZLE), image format setting, ROI area setting, image compression format, gamma usage, RGB2RGB enable, YUV clamp control and layer enable.
2. Image memory address setting: Set the memory address according to the image format configured in step 1. If it is in package format, use dcregFrameBufferAddress0~1/ dcregOverlayAddress0~3 for memory address allocation. If it is in planar format, you need to configure the corresponding planar addresses: dcregFrameBufferU/VPlanarAddress0~1/ dcregOverlayU/VPlanarAddress0~3. This information needs to be configured for each frame.
3. Address stride configuration: Stride refers to the byte distance from the first address of an image in a certain row of the current planar to the first address of the next row. Configure through dcregFrameBufferStride0~1, dcregFrameBufferUStride0~1, dcregFrameBufferVStride0~1, dcregOverlayStride0~3, dcregOverlayUStride0~3, and dcregOverlayVStride0~3.
4. Configure the location information of each layer.
   a) For the video layer, set the width and height of the image through dcregFrameBufferSize0~1. If ROI is enabled, use dcregFrameBufferROIOrigin0~1, configure the ROI area through dcregFrameBufferROIOrigin0~1, dcregFrameBufferROISize0~1, and dcregFrameBufferConfigEx0~1.
   b) Set the layer location and size through dcregOverlayTL0~3 and dcregOverlayBR0~3. If ROI is enabled, configure the ROI area through dcregOverlayROIOrigin0~3 and dcregOverlayROISize0~3.
5. Set blending and color key according to the transparent image information.
   a) Color key setting: Specify the key color through dcregFrameBufferColorKey0~1, dcregFrameBufferColorKeyHigh0~1, dcregOverlayColorKey0~3, and dcregOverlayColorKeyHigh0~3. Replace the non-key color through dcregFrameBufferBGColor0~1. Note: Video 0 and layer0-1 share a BGColor0. Video1 and layer2-3 share a BGColor1.
6. Scale control configuration:
   a) Scale factor control: dcregFrameBufferScaleFactorX0~1, dcregFrameBufferScaleFactorY0~1.
   b) Filter type select: Configure the filter type through dcregFrameBufferScaleConfig0~1.
   c) Filter coefficient configuration: Configure the horizontal filter coefficient through dcregHoriFilterKernelIndex0~1 and dcregHoriFilterKernel0~1. Configure the vertical filter coefficient through dcregVertiFilterKernelIndex0~1 and dcregVertiFilterKernel0~1.
   d) Configure scale for layer0 and layer2. Note that layer1 and layer3 do not support scale.

7. Color space conversion configuration: If the read image is in YUV format, you need to configure the color space coefficient through dcregFrameYUVToRGBCoef*0~1. If there is a clamp bit requirement, you also need to configure the dcregFrameY/UVClampBound0~1 value.
8. To enable gamma in the layer properties, the following is required:
   a) Configure the gamma table coefficient through dcregDeGammaIndex0~1, dcregDeGammaData0~1, and dcregDeGammaExData0~1.
9. Select the layer overlay order and the corresponding panel through dcregSeFrameBufferConfigEx0~1 and dcregOverlayConfigEx0~3.
   a) Layer overlay order select: The default overlay order is that the background layer and layer0 are overlayed together as the background layer of layer1. The overlay of all layers is completed in turn. Therefore, select the layer that the current layer is to be overlayed with through Layer_sel.
   b) Select the corresponding panel through PANEL_SEL.
10. Note: Layer select is exclusive. For example, if video0 is used as the source of panel0, it cannot be used as the source of panel1.
11. Due to the real-time requirements of the display, there is an automatic watermark mechanism to ensure that DPU fetches data in memory in time:
    a) Set the watermark position of the cache in the hardware through dcregFrameBufferWaterMark0~1.
    b) Design the QoS values of low latency and high latency through dcregQos.
    c) When the watermark of the cache in the DPU is lower than that set by the software, the DPU automatically uses high QoS to request data from the bus. When the watermark of the cache in the DPU is higher than that set by the software, the DPU automatically jumps to low QoS to request data from the bus.
    d) Set the read DDR bandwidth capability of the DPU by configuring gcReadOT. The calculation formula is: BW（MB/s）= freq(axi_clk)*readOT*trans_size/latency. trans_size represents the trans size of DPU AXI read and write. The typical value of trans is 128B. The latency is the time from when the DPU sends the read request to when the DPU gets the last data of this transmission, the unit is ns. Bandwidth analysis is a systematic project, this document only gives the DPU bandwidth calculation formula. OT needs to be set in a balanced manner across the entire scene.

1.5.2.2 MMU Configuration

NA

1.5.2.3 DEC400 Configuration

1. Determine whether DEC400 is enabled according to the property, and set it through Disable_compression of gcregAHBDECControl. This setting needs to match the property defined by the layer.
2. If you need to enable DEC, turn on the clock and enable clock gating.

3. Configure properties through gcregAHBDECControlEx and gcregAHBDECControlEx2.
4. Configure gcregAHBDECReadConfig0~15 to confirm the compression formats of different streams. The layer information corresponding to different streams is shown in Figure & Table 1-19:

Figure & Table 1-19 Relationship between DEC400 stream and DPU read format

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 layer 0		stream9
Video layer1	Package format or Y planar	stream16
Video layer1	U/UV planar	stream17
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 layer 1		stream24

For the detailed configuration process, see the DEC400 configuration document.

1.5.2.4 Display Configuration

After configuring the properties of the image layer, two display channels need to be configured:

1. Configure the timing parameter according to the display device manual. After the timing is determined, it is not allowed to modify this parameter during inter-frame configuration. See Figure & Table 1-20:

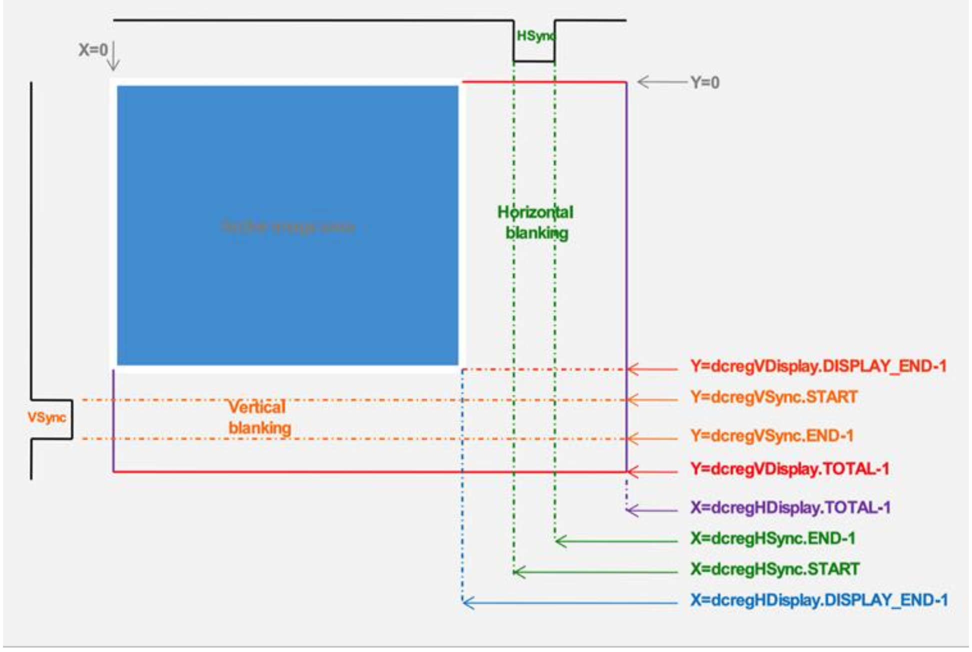


Figure & Table 1-20 Display screen parameter diagram

2. Configure the properties of the display channel according to dcregPanelConfig0~1.
   a) Control the polarity of the output timing control signal. Note that these properties are not allowed to be adjusted after the initialization is determined.
   b) Select the post-image processing function of the output image: includes gamma and 3D LUT select.
3. Select DPI output or DP output through bus_output_sel of dcregDpConfig0~1. If DP output is selected, use DP_DATA_FORMAT and DP_DATA_YUV_FORMAT to select the output format. If DPI output is selected, select the output format through dcregDpiConfig0~1. Note that the output format needs to be consistent with the input format used by subsequent modules. For the output format, see the description in Section 1.4.4.
4. If the output format is YUV, you need to configure the coefficient table through dcregRGBToYUVCoef[0-3]0~1.
5. If 3D-LUT is selected, configure it through dcregThreedLutScale.
6. Configure the effective time of the configured parameters through dcregPanelConfigEx0~1. The default is 0, which means that all new settings will take effect after sync of the next frame. If it is set to 1, software is required to ensure that the settings are configured before the arrival of vsync of the next frame. It is recommended to set it to 0. This register is used to enable or disable the shadow function.

1.5.2.5 Start DPU

After ensuring that all configurations are correct,

1. Enable all interrupts through AQIntrEnbl and gcregAHBDECIntrEnblEx2.

2. Start DPU through dcregDisplayPanelStart.

1.5.2.6 Inter-Frame Configuration Update Description

After CPU responds to the DPU interrupt, it checks whether the current interrupt is the frame end. If it is the frame end, refer to Sections 1.5.2.1 and 1.5.2.4 to update the configuration parameters.

2 HDMI

2.1 Overview

The DWC_hdmi_tx provides a variety of standard audio, video, and system interfaces.
The input video stream can be either RGB 4:4:4, YCbCr 4:2:2, YCbCr 4:4:4, or YCbCr 4:2:0 in single data rate (SDR) bus formats. The timing format in the video mode must follow the CEA-861-E specification. An embedded color space conversion allows the pixel color format to be converted on the HDMI source side to match the best with the HDMI sink capabilities.

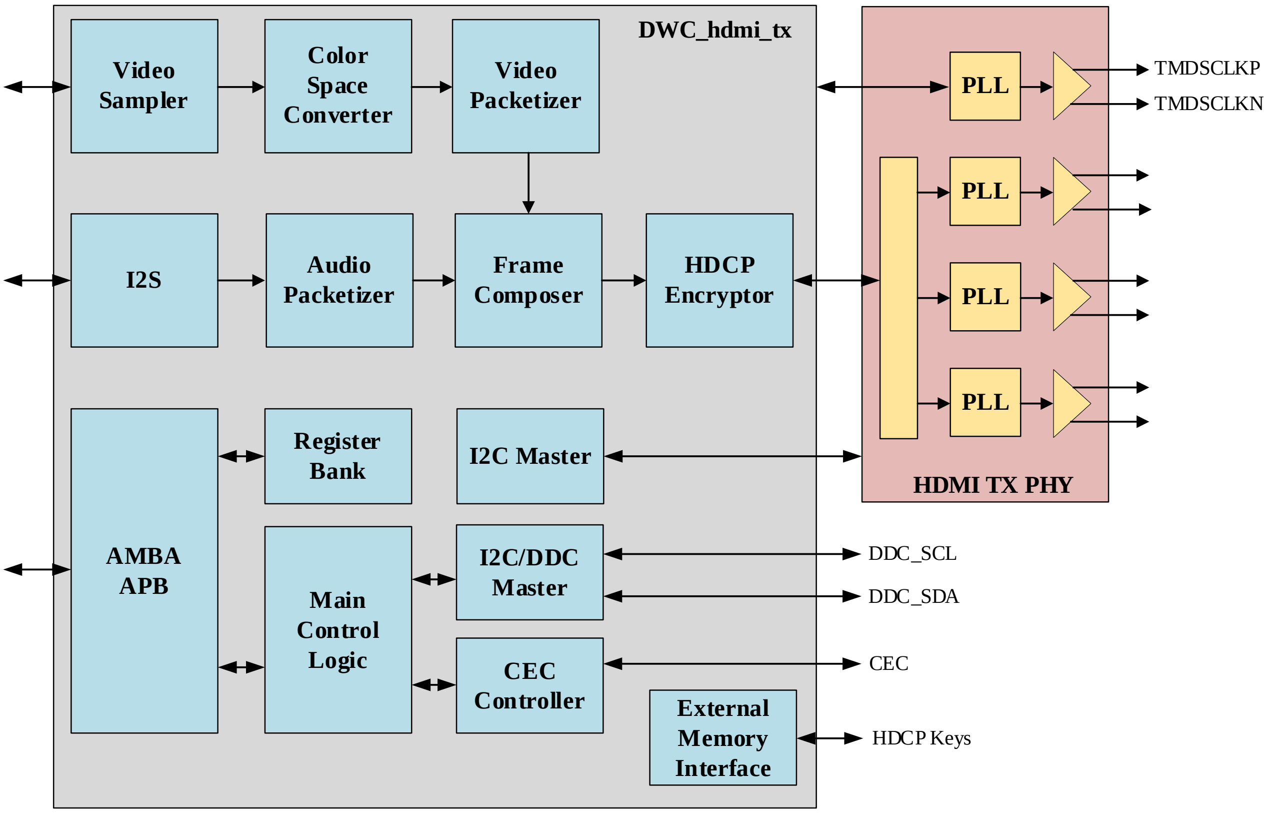


Figure & Table 2-1 DWC_hdmi_tx controller block diagram

The input audio stream can be provided through a standard I2S format interface (for all audio types: L-PCM, NL-PCM, and HBR).
The system interface (the interface that connects to the processor bus) is an AMBA APB.
Finally, the controller can output the video in full HD with color mode up to 48-bit. Also, inserts high fidelity audio up to eight-channels over low resolution video formats by performing automatic pixel repetition on the input video stream.

2.2 Main Features

DWC_hdmi_tx has the following features:

- AMBA APB 3.0, address width is 16bit, data width is 8bit

- HDMI host interrupt, including HDCP1.4 and SCDC
- CEC engine dedicated interrupt signal raised by a wake-up event
- 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
- Supports color space converter
- Supports internal pixel repetition
- I2S audio interface，up to eight-channels
- Supports Consumer Electronics Control (CEC)
- Supports TMDS data scrambling
- Supports HDCP，HDCP version is HDCP1.4
- SCDC I2C DDC access
- I2C DDC, EDID block read mode

2.3 Interface

Pin Name	Direction	Width	Description
HDMI_CEC	IO	1	CEC data bus
HDMI_HPD	IO	1	Hot plug detect signal for HDMI
HDMI_REXT	IO	1	Rx presence detection signal for all TMDS data lanes for HDMI mode of operation
HDMI_SCL	IO	1	HDMI DDC I2C slave clock for HDCP and E-EDID
HDMI_SDA	IO	1	HDMI DDC I2C slave data for HDCP and E-EDID
HDMI_TMDSCLKN	O	1	Negative TMDS differential line driver clock output
HDMI_TMDSCLKP	O	1	Positive TMDS differential line driver clock output
HDMI_TMDSDATAN0	O	1	Negative TMDS differential line driver data output for data channel 0
HDMI_TMDSDATAN1	O	1	Negative TMDS differential line driver data output for data channel 1

Pin Name	Direction	Width	Description
HDMI_TMDSDATAN2	O	1	Negative TMDS differential line driver data output for data channel 2
HDMI_TMDSDATAP0	O	1	Positive TMDS differential line driver data output for data channel 0
HDMI_TMDSDATAP1	O	1	Positive TMDS differential line driver data output for data channel 1
HDMI_TMDSDATAP2	O	1	Positive TMDS differential line driver data output for data channel 2

2.4 Function Description

2.4.1 Video Pixel Sampler

The Video pixel sampler block synchronizes the video data, as per the video data input mapping defined by the Color Depth (Deep Color) and format configuration.

2.4.2 Video Packetizer

This block is responsible for the following:

- Pixel repetition (if not performed in the input video stream and is required by you)
- 10-bit packing when in deep color modes
- YCbCr 422 re-mapping according to the HDMI specification
- Clock rate transformation from pixel or repetition clock to the final TMDS clock domain (by means of FIFOs)

2.4.3 HDMI 2.0 TMDS Scrambling Feature

The DWC_hdmi_tx supports the TMDS data scrambling that is required to transmit 2160p@60Hz in RGB4:4:4, YCbCr 4:4:4, or YCbCr 4:2:2 video formats. When the HDMI 2.0 support feature is enabled, all output video modes with a TMDS frequency higher than 340MHz require the scrambler feature be activated by setting the fc_invidconf.HDCP_keepout register field to 1‘b1 and fc_scrambler_ctrl.scrambler_en to 1‘b1. For video modes at frequencies equal or below 340MHz, activating the scrambler feature is optional. The activation depends on your requirements and the connected HDMI Sink device.

2.4.4 Color Space Conversion

This block is responsible for the following video color space conversion functions:

- RGB to/from YCbCr
- 4:2:2 to/from 4:4:4 up (pixel repetition or linear interpolation)/down-converter
- Limited to/from full quantization range conversion

The CSC supports all the timings reported in the CEA-861-E specification and the following pixel modes:

- RGB 444 and YCbCr 444: 24, 30 bits
- YCbCr 422: 16, 20bits

Color Space Conversion to and from YCbCr 4:2:0 is not supported.

2.4.5 Audio Interface

The I2S interface uses the I2S audio clock input to sample Linear-PCM input data and stores it in an input audio FIFO. There are four I2S data lines that support up to 192kHz sampling rates (supports a maximum theoretical audio rate of 768kHz or 1536kHz (supported in HDMI 2.0 only) for a two-channel standard I2S). The I2S interface is compliant with the I2S specification from NXP.

Each I2S interface supports two audio channels. The DWC_hdmi_tx has four I2S interfaces that supports up to eight audio channels simultaneously at 192kHz. Each audio sample width can be configured to be from 16 bits up to 24 bits. The I2S_width field of the aud_conf1 register selects the bit width for each right/left sample. Each right/left channel can carry 1 to N bits (N = 16 to 24).

The ii2slrclk input signal must have the same frequency as the Audio Sampling Rate fs, that is 32kHz to 192kHz. The INPUTCLKFS setting must be consistent with the frequency of ii2sclk.

In I2S mode (as depicted in Figure & Table 2-2), the most significant bit (MSB) of the sub-frame data must be sent on the second active edge of ii2sclk following an ii2slrclk transition. The other bits up to the least significant bit (LSB) are then transmitted in order. Depending on word length, ii2sclk frequency, and sample rate, there may be unused ii2sclk cycles between the LSB of one sample and the MSB of the next.

The sub-frame can be as wide or larger than the sub-frame data size (such as, BPCUV+24 bits audio sample = 29 clock periods). In a typical scenario, the sub-frame can be 32 clocks wide, and all clock periods after the audio sample LSB are unused and i2sdata set to 0, regardless of audio sample width.

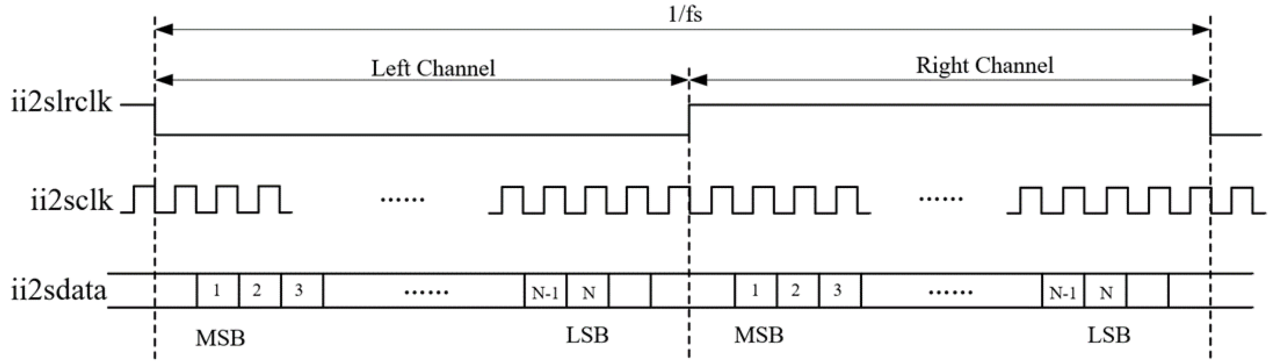


Figure & Table 2-2 I2S format

2.4.6 Frame Composer

This block assembles the video, audio, and data packets in a consistent frame and finally transmitted to the HDMI TX PHY.

2.4.7 E-EDID/HDCP/SCDC I2C E-DDC Interface

The E-DDC channel is a dedicated I2C master interface that allows the read of a Sink E-EDID based on system needs. Through this interface, it is possible to access the Sink EDID and the SCDC structure in an HDMI 2.0 subsystem.

The I2C master included in DWC_hdmi_tx complies with the version 2.1 of the I2C Bus Specification and can be accessed through the APB interface starting at address 0x7E00.

The following I2C operations are provided:

- Single data byte write
- Single data byte read
- Sequential data byte read (8 bytes)
- Single data byte extended read
- Sequential data byte extended read (8 bytes)
- SCDC update read
- SCDC read request detection
- SCDC polling mechanism
- Bus Clear mechanism

The SCDC read request detection operates, when a device external to DWC_hdmi_tx drives the SDA line low. This detection is accurate until the DWC_hdmi_tx I2C master is the only one driving the I2C DDC bus.

2.4.8 CEC Hardware Engine

CEC is a protocol that provides high-level control functions between all of the various audiovisual products in your environment. It is an optional feature in the HDMI Specification. It uses only one bidirectional line for transmission and reception.

All transactions on the CEC line consist of an initiator and one or more followers. The initiator sends the message structure and the data. The follower is the recipient of any data and is responsible for setting any acknowledgment bits.

2.4.9 Built-in Color Bar

There is a built-in color bar module added into HDMI system for test. The built-in color bar function is disabled by default. Once the user enables it, color bars of 7 resolutions are available and can be selected.

- 640*480p@60Hz
- 720*480p@60Hz
- 1280*720p@24Hz
- 1280*720p@60Hz
- 1920*1080p@30Hz
- 3840*2160p@30Hz
- 4096*2160p@30Hz

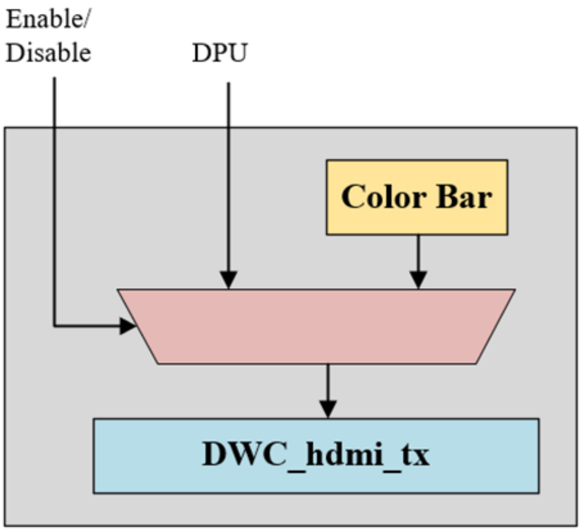


Figure & Table 2-3 Color bar in HDMI system

2.5 Usage

Built-in color bar programming model.
Please use the register shown in Figure & Table 2-4 to enable color bar function.

- Name: HDMI_COLOR_BAR_CFG
- Description: This register is used to enable or disable built-in color bar function and select one of resolutions if this function is enabled.
- Size: 32 bits
- Offset: 0xD0
- Exists: Yes

Figure & Table 2-4 Fields for register: HDMI_COLOR_BAR_CFG

Bits	Name	Access	Description
31:7	RSVD	R	Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
6:4	RESOLUTION_SELECT	R/W	HDMI color bar resolution select signal 3'b000: 640_480_60Hz 3'b001: 720_480_60Hz 3'b010: 1280_720_24Hz 3'b011: 1280_720_60Hz 3'b100: 1920_1080_30Hz 3'b101: 3840_2160_30Hz 3'b110: 4096_2160_30Hz 3'b111: 640_480_60Hz

Bits	Name	Access	Description
			Value After Reset: 0x0 Exists: Always
3:1	RSVD	R	Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
0	COLOR_BAR_ENA	R/W	HDMI color bar enable signal 0: Color bar is disabled. 1: Color bar is enabled. Value After Reset: 0x0 Exists: Always

3 MIPI DSI

3.1 Overview

The MIPI DSI module is composed of a MIPI DSI host controller and a 4lane D-PHY TX.
The MIPI DSI host controller follows MIPI Alliance Specification *MIPI Alliance Specification for Display Serial Interface (DSI), Version 1.2-16 June 2014*, providing a PPI interface compatible with MIPI Alliance standards with the PHY docking end. The pixel input terminal provides a DPI interface compatible with the MIPI Alliance Standard *MIPI Alliance Specification for Display Pixel Interface v2.00 (DPI-2)-15 September 2015* for docking the input of the display control module. At the same time, the controller provides standard AMBA APB2.0 interface for the control of MIPI DSI controller and D-PHY and provides the sending function of standard DCS commands.
The D-PHY TX follows MIPI Alliance Specification *MIPI Alliance Specification for D-PHY, Version 1.2, 01 August 2014*. DPHY TX has a maximum of 4 data lanes, and the maximum rate of each data lane is 2.5Gbps. It provides a PPI interface compatible with MIPI Alliance standards to interface with external MIPI controllers.

3.2 Main Features

The MIPI DSI host controller has the following features:

- Compliant with MIPI Alliance standards
- Standard DPI port for reception of input display image
- Supports up to 4K@60fps input display image
- Supports a variety of bpp of RGB image input to display, such as RGB101010
- Supports the standard DCS command transmission through the AMBA APB interface
- Timing accurate signaling of frame and line synchronization packets through non-burst pulse mode
- ECC and checksum capabilities

The MIPI DSI D-PHY TX has the following features:

- Compliant with MIPI Alliance standards
- Up to four D-PHY TX data lanes
- Up to 2.5Gbps per lane
- Bidirectional communication and escape mode (up to 10Mbps) support through data lane0
- Supports ultra low-power mode with PLL disable
- Supports for End of Transmission Package (EoTP)

3.3 Interface

Figure & Table 3-1 Pin description table

Pin Name	Direction	Width	Description
REXT	IO	1	Reference resistor connection PAD
CLKP	IO	1	Positive D-PHY differential clock Line
CLKN	IO	1	Negative D-PHY differential clock line
DATAP0	IO	1	Positive D-PHY Differential data line
DATAN0	IO	1	Negative D-PHY differential data line
DATAP1	IO	1	Positive D-PHY differential data line
DATAN1	IO	1	Negative D-PHY differential data line
DATAP2	IO	1	Positive D-PHY differential data line
DATAN2	IO	1	Negative D-PHY differential data line
DATAP3	IO	1	Positive D-PHY differential data line
DATAN3	IO	1	Negative D-PHY differential data line

3.4 Function Description

3.4.1 Function Overview

The overall functional block diagram of MIPI DSI is shown in Figure & Table 3-2.

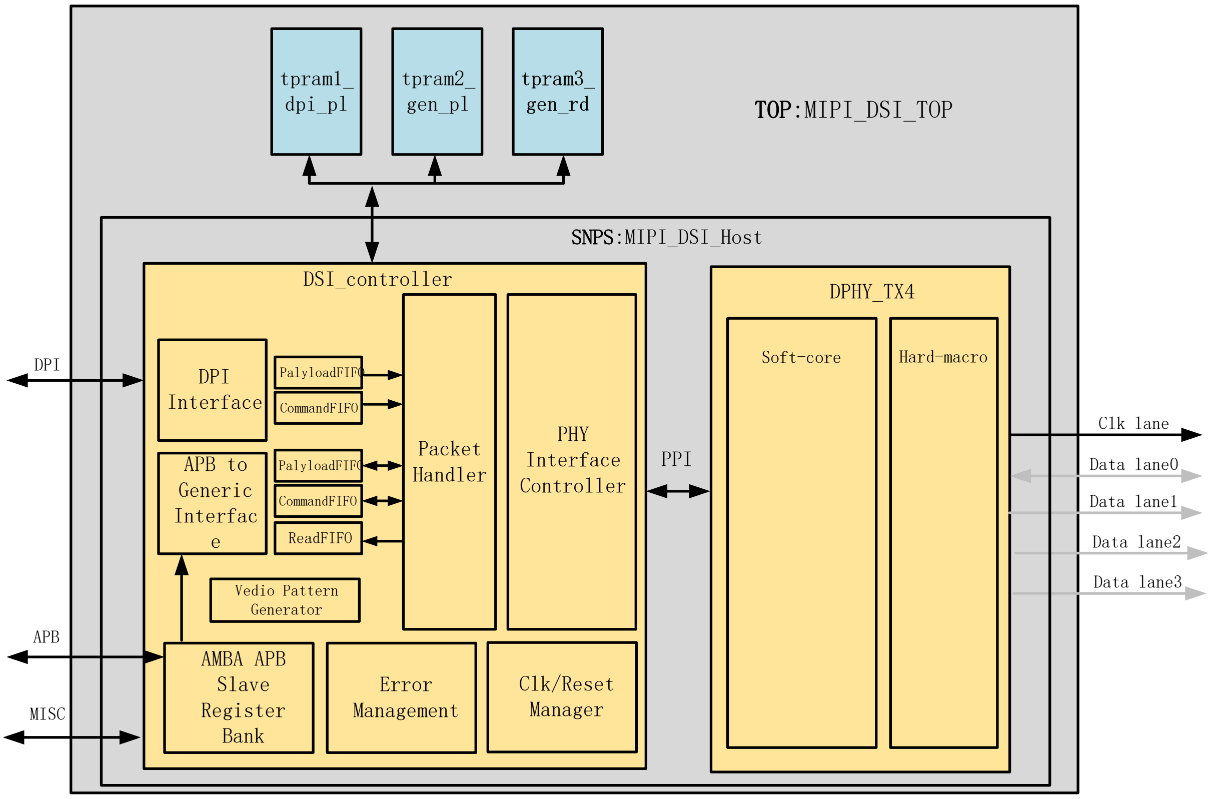


Figure & Table 3-2 MIPI DSI functional block diagram

The main blocks of MIPI DSI host are as follows:

- DPI interface: Captures the data and control signals and conveys them to a FIFO for video control signals and another one for the pixel data. This data is then used to build video packets, when in video mode.
- The register bank is accessible through a standard AMBA-APB slave interface, providing access to the MIPI DSI registers for configuration and control. There is also a fully programmable interrupt generator to inform the system about certain events.
- The PHY interface controller is responsible for managing the D-PHY PPI interface. It acknowledges the current operation and enables low-power transmission/reception or a high-speed transmission. It also performs data splitting between available D-PHY lanes for high-speed transmission.
- The packet handler schedules the activities inside the link. It performs several functions based on the video transmission mode that is used (burst mode or non-burst mode with sync pulses or sync events). It builds long or short packet generating correspondent ECC and CRC codes. This block also performs the following functions:
  - Packet reception
  - Validation of packet header by checking the ECC
  - Header correction and notification for single-bit errors
  - Termination of reception
  - Multiple header error notification
- The APB-to-Generic block bridges the APB operations into FIFOs holding the Generic commands.

- The error management notifies and monitors the error conditions on the DSI link. It controls the timers used to determine if a timeout condition occurred, performing an internal soft reset and triggering an interruption notification.

3.4.2 DPI Interface

The DPI interface follows the MIPI DPI-2 specification with pixel data bus width up to 30 bits. It is used to transmit the information in video mode in which the transfers from the host processor to the peripheral take the form of a real-time pixel stream.

The DPI interface can be configured to increase flexibility and promote correct usage of this interface for several systems. These configuration options are as follows:

- Polarity control: All the control signals are programmable to change the polarity depending on system requirements.
- After the core reset, DPI waits for the first VSYNC active transition to start signal sampling, including pixel data, thus avoiding starting the transmission of the image data in the middle of a frame.
- To avoid FIFO underflows and overflows, the configured number of pixels is assumed to be received at all times. This happens even if the dpidataen pin is active for more or less time than necessary.
- To keep the memory organized with respect to the packet scheduling, the number of pixels per packet parameter is used to separate the memory space of different video packets.

When shutting down the display, it is necessary for the DPI video to keep active for one frame after the command being issued. This ensures that the commands are correctly transmitted before actually disabling the video generation at the DPI interface.

It is possible to update the DPI configuration on the fly without impacting the current frame. It is done with the help of shadow registers. This feature is controlled by the VID_SHADOW_CTRL register.

3.4.3 APB Slave Generic Interface

The APB slave interface allows the transmission of generic information in command mode. Commands sent through this interface are not constrained to comply with the DCS specification, and can include generic commands described in the DSI specification as manufacturer-specific.

3.4.4 Transmission of Commands

The MIPI DSI supports the transmission of commands, both in high-speed and low-power, while in video mode. The MIPI DSI uses Blanking or Low-Power (BLLP) periods to transmit commands inserted through the APB Generic interface.

- If the lp_cmd_en bit of the VID_MODE_CFG register is 0, the commands are sent in high-speed in video Mode. In this case, the MIPI DSI automatically determines the area where each command can be sent and no programming or calculation is required.
- MIPI DSI can be configured to send the low-power (LP) commands during the HS video mode transmission. To enable this feature, set the lp_cmd_en bit of the VID_MODE_CFG register to 1.

In this case, it is necessary to calculate the time available, in bytes, to transmit a command in LP mode to Horizontal Front Porch (HFP), Vertical Sync Active (VSA), Vertical Back Porch (VBP), and Vertical Front Porch (VFP) regions.

- MIPI DSI can also send read command and if a read command is issued on the last line of a frame, the edpihalt signal gets asserted and stays asserted until the read command is in progress. The video transmission should be stopped during this period.
- To reduce the power consumption of the D-PHY, the MIPI DSI, when not transmitting in the high-speed mode, allows the clock lane to enter into the LP mode.

3.4.5 Virtual Channel

The MIPI DSI supports choosing the Virtual Channel (VC) for use for each interface. Using multiple VCs, the system can address multiples displays at the same time, when each display has a different VC identifier.

3.4.6 Video Mode Pattern Generator

The video mode pattern generator allows the transmission of horizontal/vertical color bar and D-PHY BER testing pattern without any stimuli.

3.4.7 Timeout Timers and Error Control

A peripheral may not immediately respond correctly to some received packets. For example, a peripheral receives a read request, but due to its architecture cannot access the RAM for a while. It may be because the panel is being refreshed and takes some time to respond. In this case, set a timeout to ensure that the host waits long enough so that the device is able to process the previous data before receiving the new data or responding correctly to new requests.

The INT_ST0 and INT_ST1 registers are associated with error condition reporting. These registers can trigger an interrupt pin to inform the system about the occurrence of errors.

3.4.8 PHY PLL Programmability

The data rate is given by the double of the PLL output clock phases frequency: Data rate (Gbps) = PLL Fout(GHz)*2. Output frequency (Fout) is a function of the input reference frequency (Fclkin) and of the multiplication/division ratios. It can be determined in the following way: Fout = (M/N)*fclkin. Where:

- M: Feedback multiplication ratio
- N: Input frequency division ratio

3.5 Usage

3.5.1 Program Flow

The MIPI DSI program process is as follows:

1. Reset the APB interface.
2. DPI configuration.
3. Configure D-PHY parameters.
4. Wake up the core and D-PHY.
5. Wait for D-PHY power-up.
6. If change video mode, DPI configuration.

3.5.2 Video Mode Pattern Generator

The MIPI DSI programming sequence to send a test pattern is as follows:

1. Video mode selection
2. Color coding configuration
3. Video frame configuration
4. Video pattern generator configuration

3.5.3 Halt conflict Scenario Processing

DSI has an interrupt used to indicate the error scene that DPI is still input in the next frame when edpihalt is pulled high. This interrupt will be triggered when such a scene occurs. The software needs to reset DSI and reinitialize DSI to continue using DSI.

The dpivsync polarity that generates the error interrupt can be configured through the register vsync_pol of vo_subsys sysreg, the interrupt mask can be configured through the register halt_err_mask of vo_subsys sysreg, and the interrupt error status can be obtained by reading the register halt_err of vo_subsys sysreg.