

## 1 引言

本应用笔记介绍了如何在 i.MX RT1170 上使用 MIPI DSI 主机控制器和 LCDIFv2 控制器来驱动一个 Mobile Industry Processor Interface ( MIPI ) Display Serial Interface ( DSI ) 接口的 LCD 屏。

MIPI DSI 控制器是一个灵活的高性能模块，提供了与基于 MIPI DSI 接口的外围设备进行通信的串行接口。

Liquid Crystal Display Interface version 2 ( LCDIFv2 ) 是 i.MX RT1170 的显示控制器模块，该模块主要用于从内存中提取图片并将其显示在 LCD 屏上。

本应用笔记以 SDK 中的 `sd_jpeg` 工程为例，硬件环境主要是 MIMXRT1170-EVK 和一个 TFT LCD 屏。

## 2 MIPI DSI 主机控制器概述

MIPI DSI 是一种多功能的高速接口，主要用于智能手机，汽车及其他平台的 LCD 屏。

i.MX RT1170 的 MIPI DSI 控制器实现了 MIPI DSI 规范中定义的所有功能，并提供了 MCU 与基于 DSI 接口的 LCD 屏之间的通信接口。

i.MX RT1170 的 MIPI DSI D-PHY 是支持 MIPI 标准的高频和低功耗物理层，并为 DSI 提供了物理层实现。

### 2.1 DSI 主机控制器

DSI 是分层打包传输数据的，图 1 给出了 DSI 标准中定义的 DSI 接口层级。i.MX RT1170 的 DSI 控制器实现了上三层的功能（lane 管理层，低层协议层以及应用层）。

## 目录

|                                  |    |
|----------------------------------|----|
| 1 引言.....                        | 1  |
| 2 MIPI DSI 主机控制器概述.....          | 1  |
| 2.1 DSI 主机控制器.....               | 1  |
| 2.2 APB 接口.....                  | 4  |
| 2.3 DPI-2 接口模块.....              | 4  |
| 2.4 DSI D-PHY.....               | 5  |
| 2.5 DSI 主机时钟.....                | 6  |
| 2.6 LCD 显示系统.....                | 7  |
| 3 LCDIFv2 控制器.....               | 8  |
| 3.1 RGB 接口.....                  | 8  |
| 3.2 LCDIFv2 相关信号.....            | 10 |
| 3.3 LCDIFv2 相关时钟.....            | 11 |
| 3.4 Video Mux 控制器.....           | 11 |
| 4 示例运行.....                      | 11 |
| 4.1 <code>sd_jpeg</code> 例程..... | 12 |
| 4.2 LCD 刷新率.....                 | 14 |
| 5 参考资料.....                      | 14 |



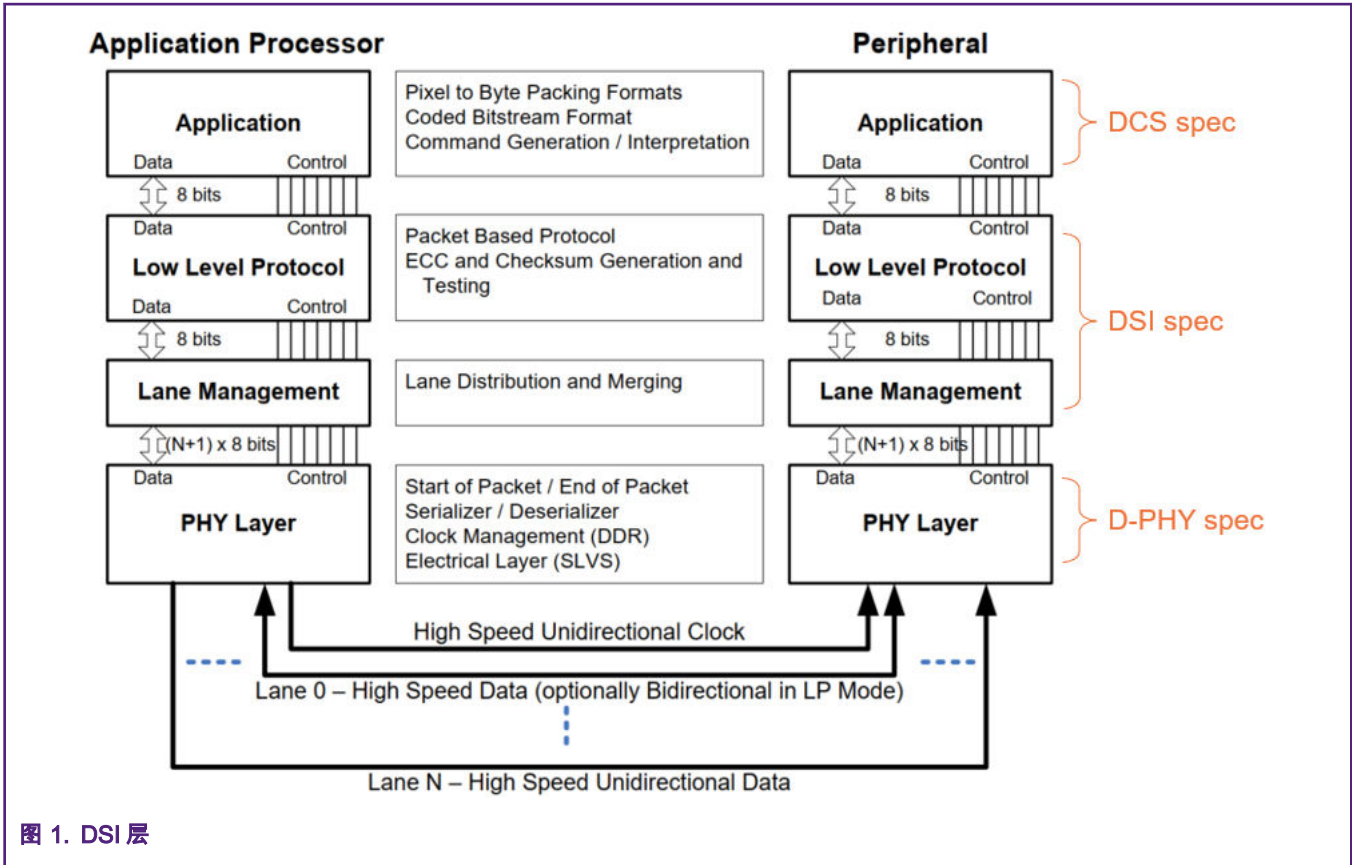


图 1. DSI 层

MIPI DSI 利用差分信号在 DSI 主机和显示模块之间进行时钟和数据的传输，其包含 1 条时钟 lane 和 1-4 条数据 lane。具体使用几条数据 lane 由应用中显示需要传输的数据量决定。RT1170 中仅支持最大 2 条数据 lane。DSI 接口与传统的 RGB 并行接口相比，其数据线大大减少了，节省了很多硬件资源。

兼容 DSI 接口的 LCD 至少支持 Command 模式和 video 模式两种操作模式中的一种。至于实际使用哪种模式是根据具体 LCD 的结构和性能来决定的。

Command 模式下主控制器和 LCD 交换数据主要是通过发送命令和数据到 LCD 模块，该模块包含一个显示控制器，显示控制器包含本地寄存器和帧缓冲区。

Video 模式下主控制器到 LCD 间的数据传输是以实时像素流的方式进行的，且该像素流只能以 High-speed 模式进行传输。

图 2 给出了一个 Command 模式和 Video 模式的简单示例。

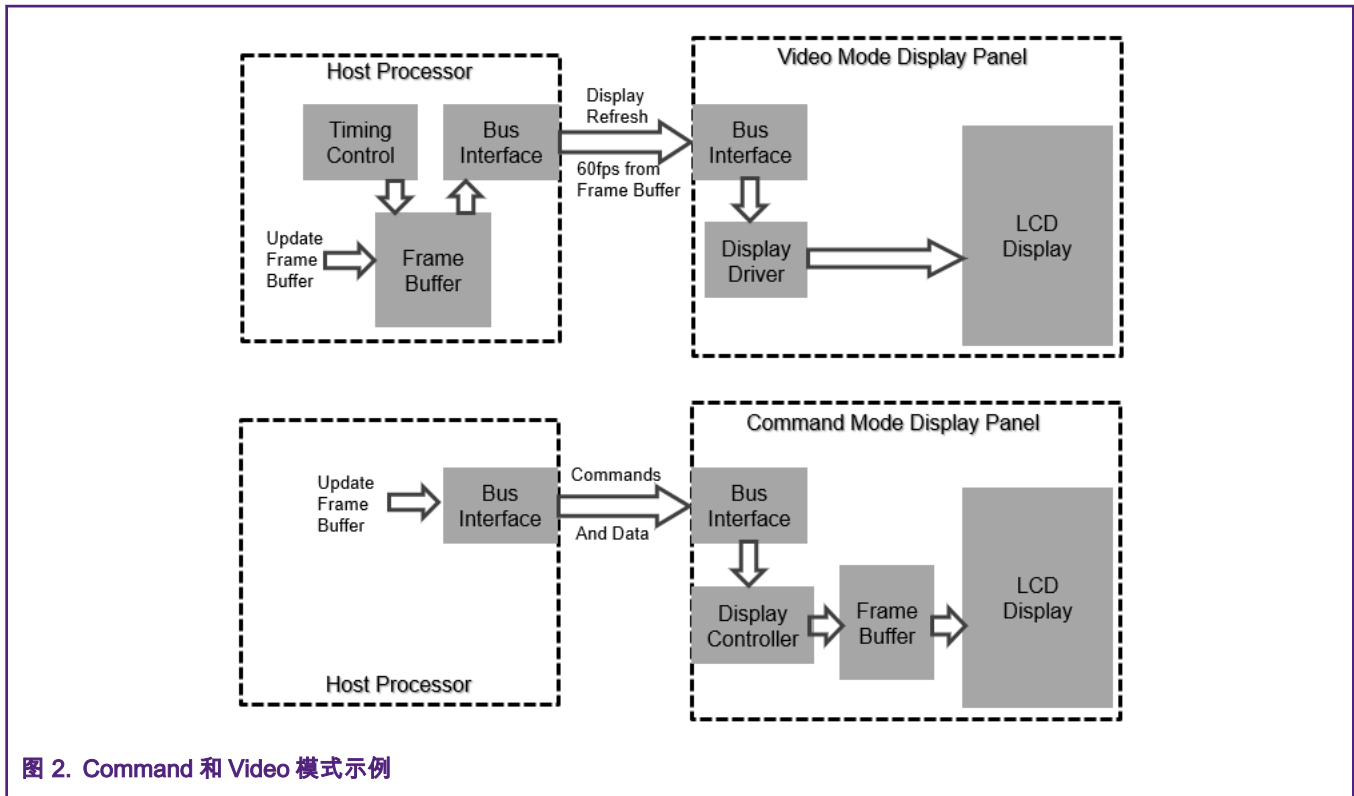


图 2. Command 和 Video 模式示例

总的来说，MIPI DSI 控制器提供以下主要功能：

- 支持 Command 和 Video 模式。
- 支持最大 2 个数据 lane。
- 支持 High-Speed 和 Low-Power 模式。
- 支持两种包接口：
  - APB 接口(状态和控制)。
  - 显示像素接口 (DPI-2)。

DSI 主机控制器主要结构如 图 3 所示。

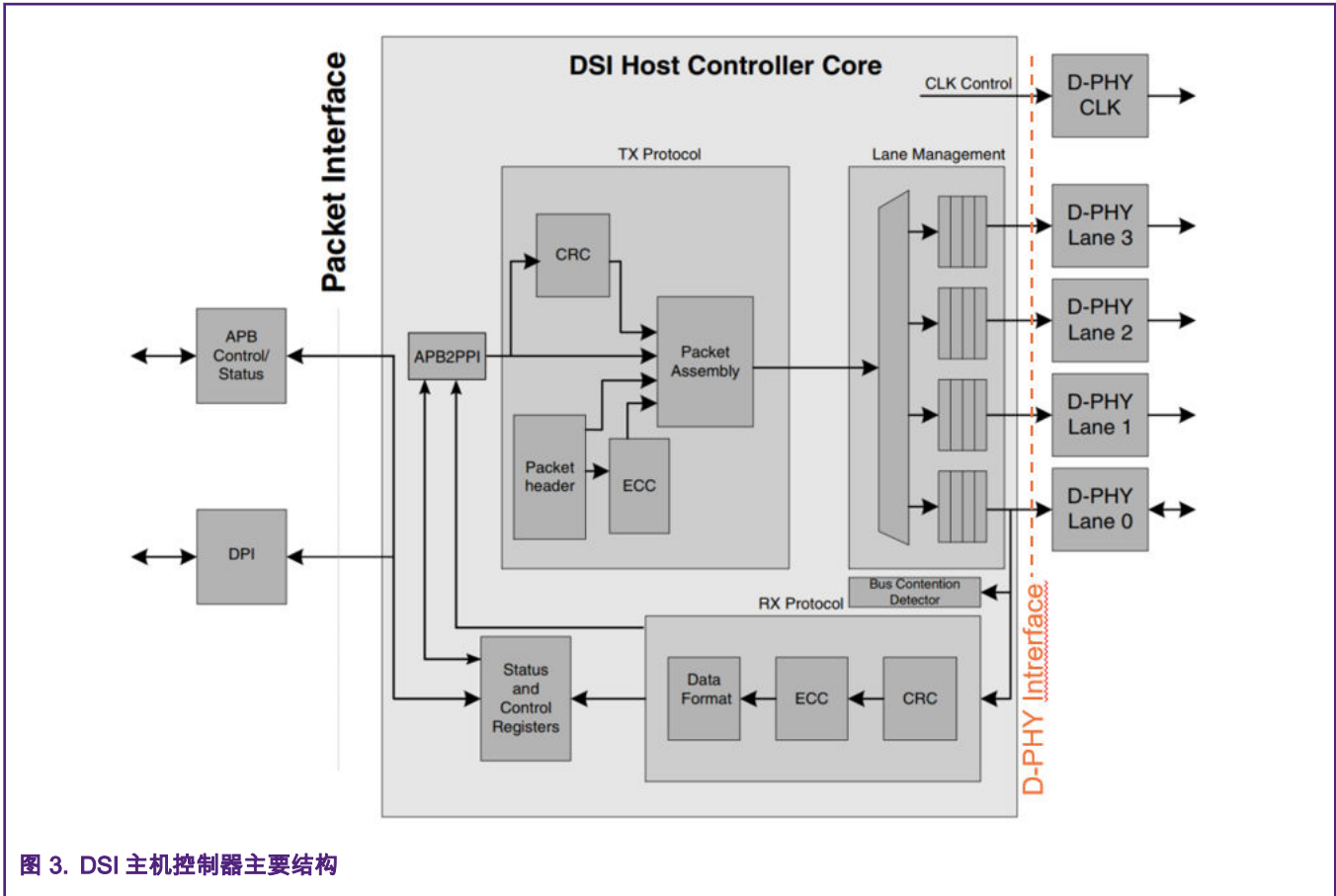


图 3. DSI 主机控制器主要结构

图 3 中，D-PHY 接口直接与 D-PHY 层相连，该层是物理层，会在后续章节中详细介绍，这里不做赘述。

DSI 主机控制器与显示控制器之间是通过 packet 接口发送和接收 DSI 命令以及数据的，主要支持两种 packet 接口：APB 接口和 DPI-2 接口。

## 2.2 APB 接口

APB 接口用于传输 DCS 和通用命令模式数据包。这些数据包是使用 APB 寄存器访问创建的。用户应用可以通过 DSI\_HOST\_APB\_PKT\_IF 寄存器访问 DSI 主机控制器数据包接口。

## 2.3 DPI-2 接口模块

DPI-2 接口模块主要为 DSI 主机控制器模块提供符合 DPI-2 标准的 packet 接口，同时处理对所有 DPI 数据类型的像素到字节的打包。

DPI-2 接口模块提供以下主要功能：

- 支持 16-，18- and 24- bit 像素数据。
- 支持 16，18，24，and 18 bit loosely packed DSI 数据类型。
- 已集成于 DSI 主机控制器。

DPI-2 接口模块给出了 DPI-2 接口模块的主要框图。DPI-2 模块接收兼容 MIPI 的 DPI-2 信号集（例如 VSYNC，HSYNC，像素数据），将时序信号和像素数据转换成相应的 DSI 数据包，并通过 DSI 主机控制器的 packet 接口将这些数据包传输至 DSI 控制器。

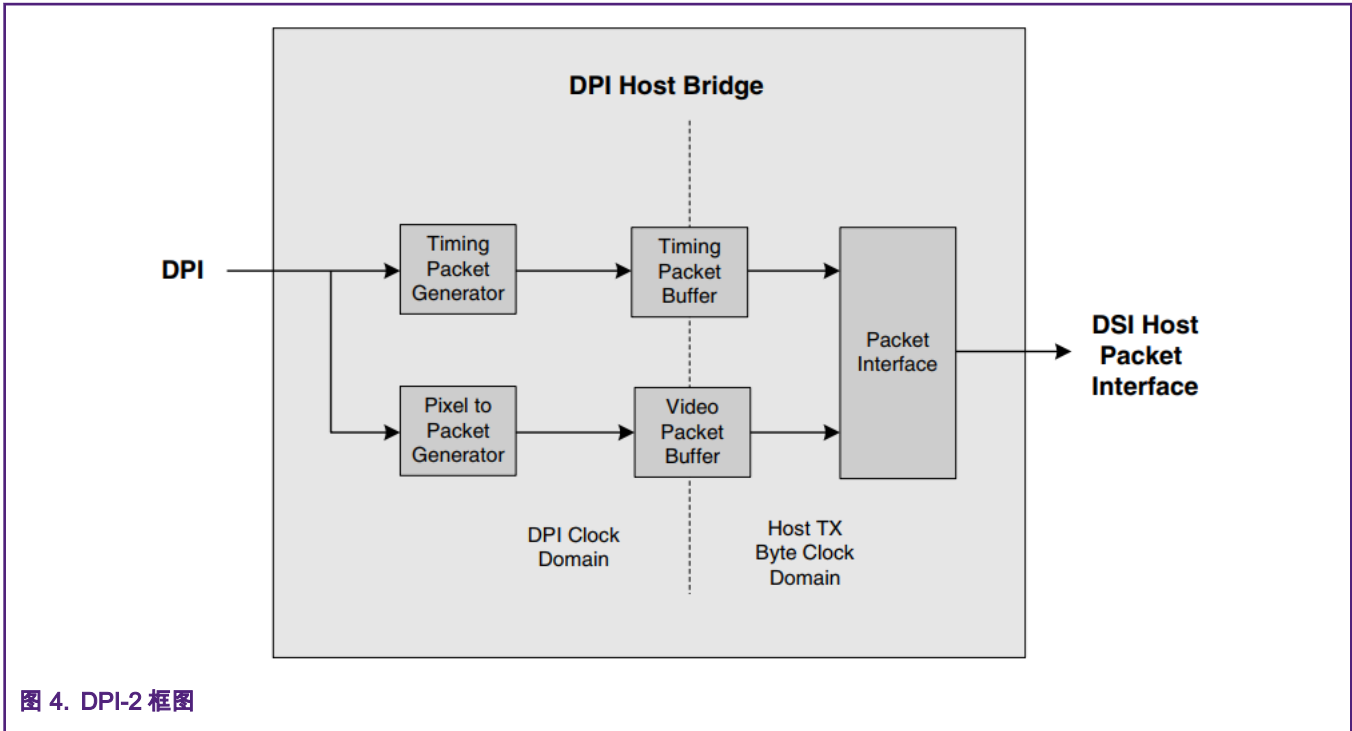


图 4. DPI-2 框图

DPI-2 接口通常也称为 RGB 接口，它使用 VSYNC，HSYNC，DOTCLK 和 ENABLE 信号来将数据传输至 LCD 屏。

## 2.4 DSI D-PHY

由前面介绍可知，D-PHY 是 DSI 的最底层—物理层，主要是采用差分时钟和数据 lane 进行数据传输。时钟为高速时钟 lane，以 DDR(双边沿采样)的方式进行高速数据传输，由主机发送至从机。图 5 是 MIPI DSI D-PHY 的具体框图，D-PHY 包含了 1 个时钟 lane 和最大 2 个数据 lane，其中每个 lane 由两根互连线组成。

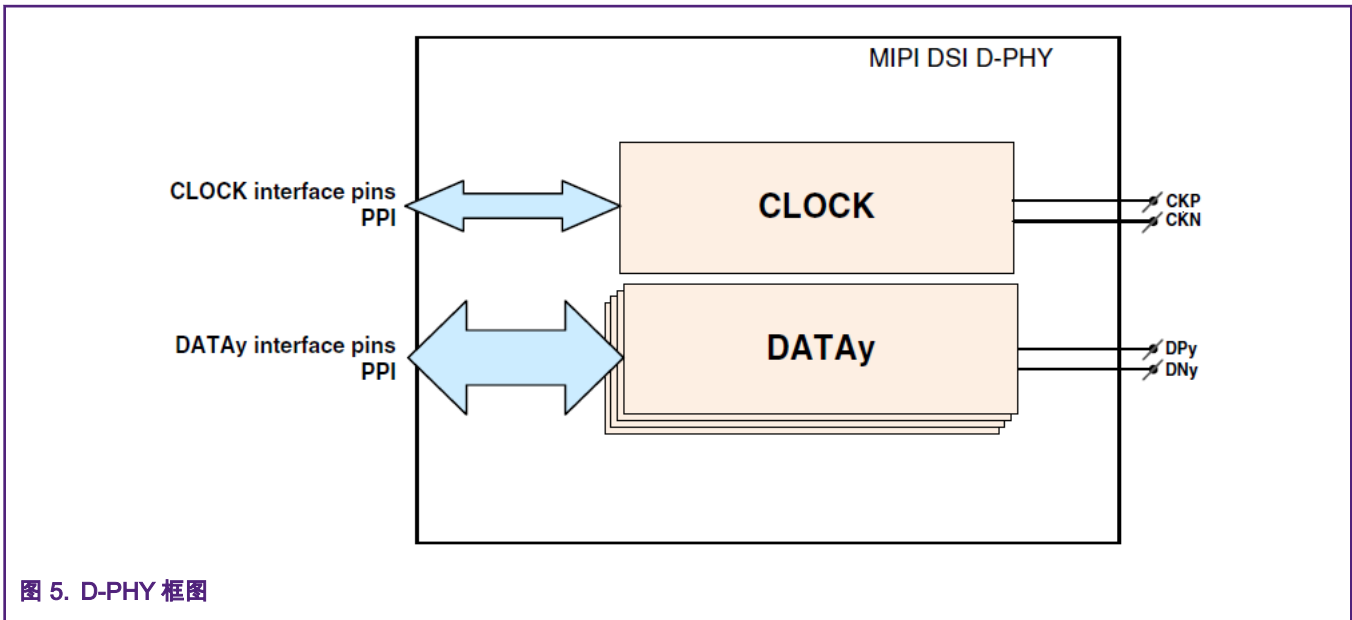


图 5. D-PHY 框图

D-PHY 支持两种传输模式：High-Speed 模式（简称 HS 模式）和 Low-Power 模式（简称 LP 模式）。

HS 模式用于高速同步数据传输，例如 Video 模式下的数据传输。HS 模式下的数据传输速率范围为 80 Mbps - 1.5 Gbps/lane。

LP 模式主要用于控制命令传输，该模式也可用于低速异步数据传输。

表 1 列出了 D-PHY 模块的主要接口信号。

表 1. DSI D-PHY 信号

| 信号          | 类型    | 描述                                   |
|-------------|-------|--------------------------------------|
| TxByteClkHS | Clock | HS 模式字节传输时钟，建议所有的数据 lane 共用该信号。      |
| TxCkEsc     | Clock | LP 模式发送时钟，该时钟信号的范围是 12 MHz 至 20 MHz。 |
| RxCkEsc     | Clock | LP 模式接收时钟，该时钟信号的最大值为 60 MHz。         |

图 6 给出了 D-PHY PLL 的框图。D-PHY PLL 是基于高性能 PLL 的频率合成器，具有锁定检测器，独立的输出分频器并支持掉电模式。该模块主要是用于输出 HS 模式下的字节传输时钟即上表中的 TxByteClkHS 信号。D-PHY PLL 模块的输入时钟为 *ref\_clk*，其范围为 24 - 200 MHz。VCO 的最大输出频率为 1.5 GHz。PLL 输出频率由公式  $(ref\_clk * (CM / (CN * CO))$  计算得出。最终 TxByteClkHS 信号频率就等于 PLL 输出频率除以 8。

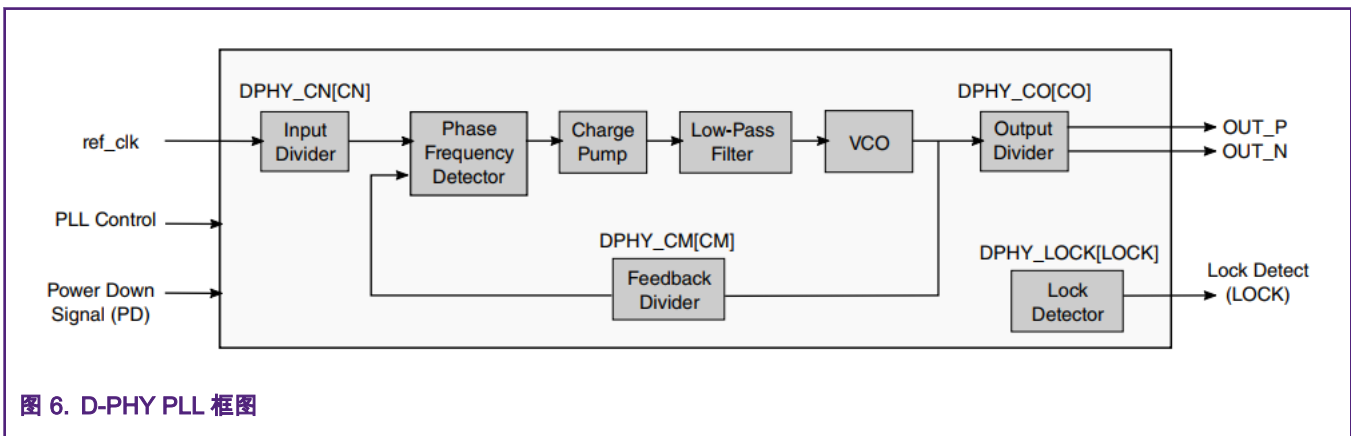


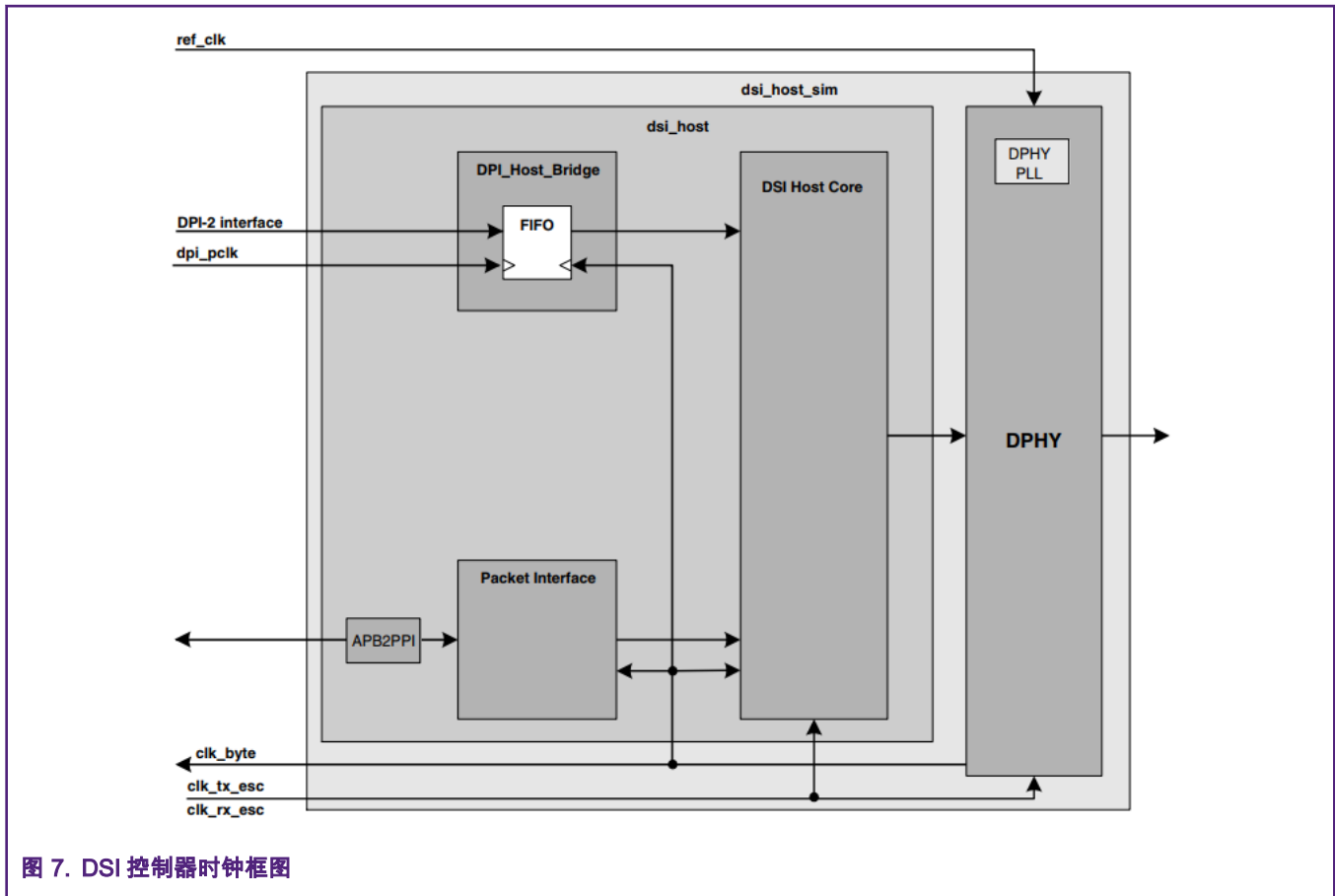
图 6. D-PHY PLL 框图

## 2.5 DSI 主机时钟

DSI 控制器需要以下时钟信号：

- 参考时钟 (*ref\_clk*) – D-PHY PLL 模块的输入时钟。
- 字节时钟 (*clk\_byte*) – 由 PLL 模块生成，用于 HS 模式下的高速数据传输。
- LP 模式时钟 (*clk\_tx\_esc*, *clk\_rx\_esc*) – 用于内部 LPDT 状态机和 LP 模式下的传输。
- Pixel clock (*dpi\_pclk*) – 用于 DPI-2 接口。

上述各时钟的分布如 图 7 所示。



## 2.6 LCD 显示系统

本节主要介绍一种基本的嵌入式 LCD 显示系统，该 LCD 支持 Video 模式。该系统包含 MCU，帧缓冲区，显示控制器，Video Mux 控制器，MIPI DSI 控制器以及一个 LCD 屏。

- MCU 计算要在帧缓冲区中显示的图像或者 GUI。MCU 计算能力越强，那么更新帧缓冲区的频率就越高，则显示越流畅。
- 帧缓冲区是用于存储图片或 GUI 的像素数据的一段内存。帧缓冲区的大小通常由 LCD 分辨率和显示色彩深度决定。一般会使用至少两个帧缓冲区来避免破坏当前显示的数据。
- 显示控制器连续不断的刷新 LCD 屏并将帧缓冲区中的数据每秒在屏上刷新多次（例如每秒刷新 60 次即为 60 Hz）。显示控制器集成于显示模块或者 MCU 内部。RT1170 有两个显示控制模块，分别为 eLCDIF 控制器和 LCDIFv2 控制器。
- Video Mux 控制器用于决定选用哪一个显示控制器来控制显示接口。例如可以选取 eLCDIF 来控制 MIPI DSI，也可以选择 LCDIFv2 来控制 MIPI DSI。
- MIPI DSI 控制器提供了一个高速串行接口，该接口允许 MCU 与兼容 MIPI DSI 接口的 LCD 之间进行通信。
- LCD 屏由显示控制器驱动并显示相应的图片或 GUI。驱动 LCD 屏主要需要配置以下关键参数：
  - 显示大小
 

显示大小由水平（像素数）乘以垂直（行数）决定。
  - 颜色深度
 

颜色深度决定了一个像素可以表示的颜色数量，以每个像素的比特数进行表示（bpp）。例如，一个像素的颜色深度为 24bpp（也可表示为 RGB888），则该像素可以表示 16777216 种颜色。
  - 刷新率（Hz）

刷新率是指一帧数据每秒钟在屏幕上刷新的次数。一般刷新率都为 60 Hz，即一帧数据每秒钟在屏上刷 60 次，较低的刷新率可能会造成不良的视觉效果。

图 8 给出了整个 LCD 显示系统的基本框架。

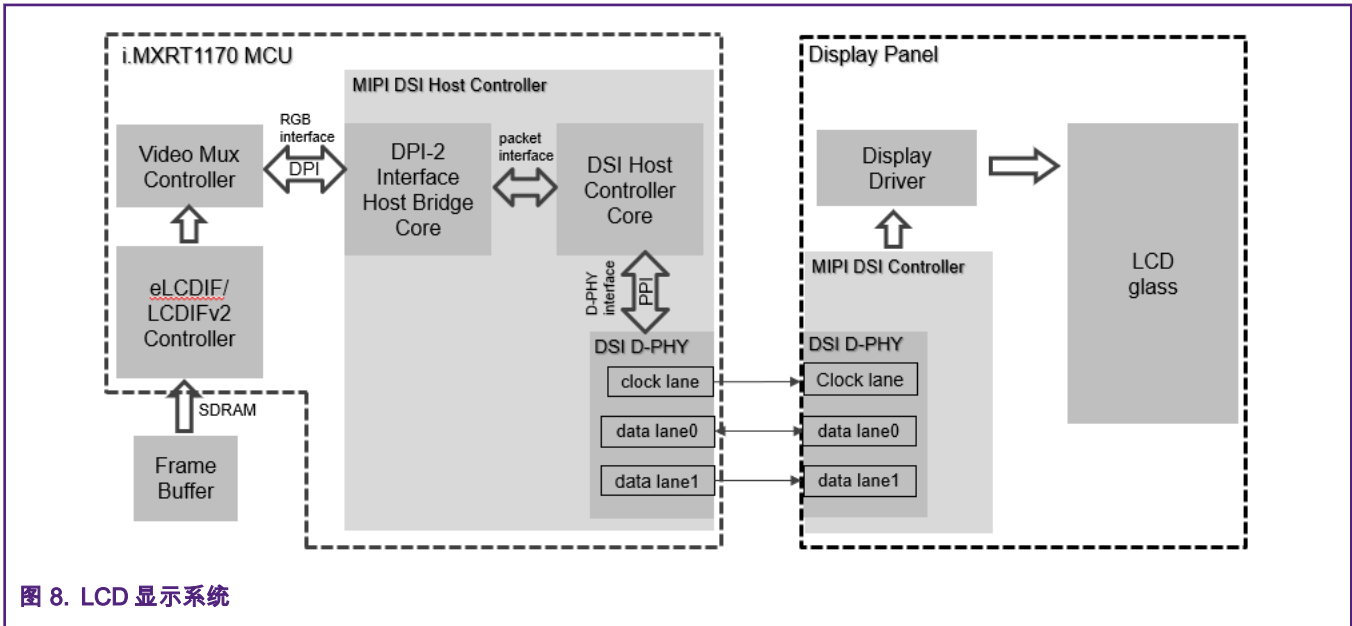


图 8. LCD 显示系统

对于整个 LCD 系统来说，帧缓冲区是比较关键的，会影响整个系统的性能以及视觉效果。

- 分配给帧缓冲区的内存空间通常与其他系统设备（如 CPU，DMA，网络等）共享。
- 帧缓冲区中的数据以位，字节，半字或字的数据形式进行组织，具体组织方式取决于颜色深度以及颜色位的排列方式。
- 缓冲区大小由公式 (长×宽×颜色深度) 计算得出。

例如：

$$720 \times 1280 \text{ display @24bpp (RGB888)} = 720 \text{ columns} \times 1280 \text{ rows} \times 3 \text{ bytes/pixel} = 2764800 \text{ bytes}$$

为了更好的进行显示，一般使用双缓冲区，此时缓冲区大小要增加一倍。双缓冲区一个用于存储当前显示的图像，另一个用于准备下一帧要显示的图像。

## 3 LCDIFv2 控制器

i.MX RT1170 包含两个显示控制器：eLCDIF 和 LCDIFv2，其功能均为从内存中取图像并显示到 LCD 屏上。

由于这两个显示控制器主体功能类似，本章仅介绍 LCDIFv2 控制器。LCDIFv2 控制器包含以下主要功能：

- 仅支持 RGB 接口。
- 显示图层最多可支持 8 个图层混合。
- 支持并行视频接口输入，且支持典型的视频接口 CSI-2 数据格式：16 bpp (YUV422 8-bit)，24 bpp (RGB888)，18 bpp (RGB666)，16 bpp (RGB565)，15 bpp (RGB555)，12 bpp (RGB444)。

### 3.1 RGB 接口

上文提到 LCDIFv2 仅支持 RGB 接口，本节主要介绍一下 RGB 接口的基本知识。

RGB 接口是直接对 LCD 屏的各像素点进行操作的，它利用 VSYNC，HSYNC，DOTCLK 和 ENABLE 信号对各像素点进行颜色填充操作，填充速度快，常用于视频或动画显示。一个像素包含红，绿，蓝三部分，每个部分最大分别占 8bit 大小，一般像素编码形式为 RGB565，RGB666，RGB888 等。



对于 RGB 接口，在每个像素时钟沿（上升或下降），显示控制器从缓冲区中获取一个像素数据，将其转换为相应的像素格式（如 RGB888），并输出到 RGB 接口。然后，像素数据便会显示在 LCD 屏的有效区域上。

为了在 LCD 屏上正确显示图像，需要根据 LCD 屏的手册对相应的时序进行配置。图 9 给出了 LCD 屏相关时序参数以及分辨率的示意图。表 2 给出了 1280 × 800 分辨率的 LCD 屏的时序参数示例。

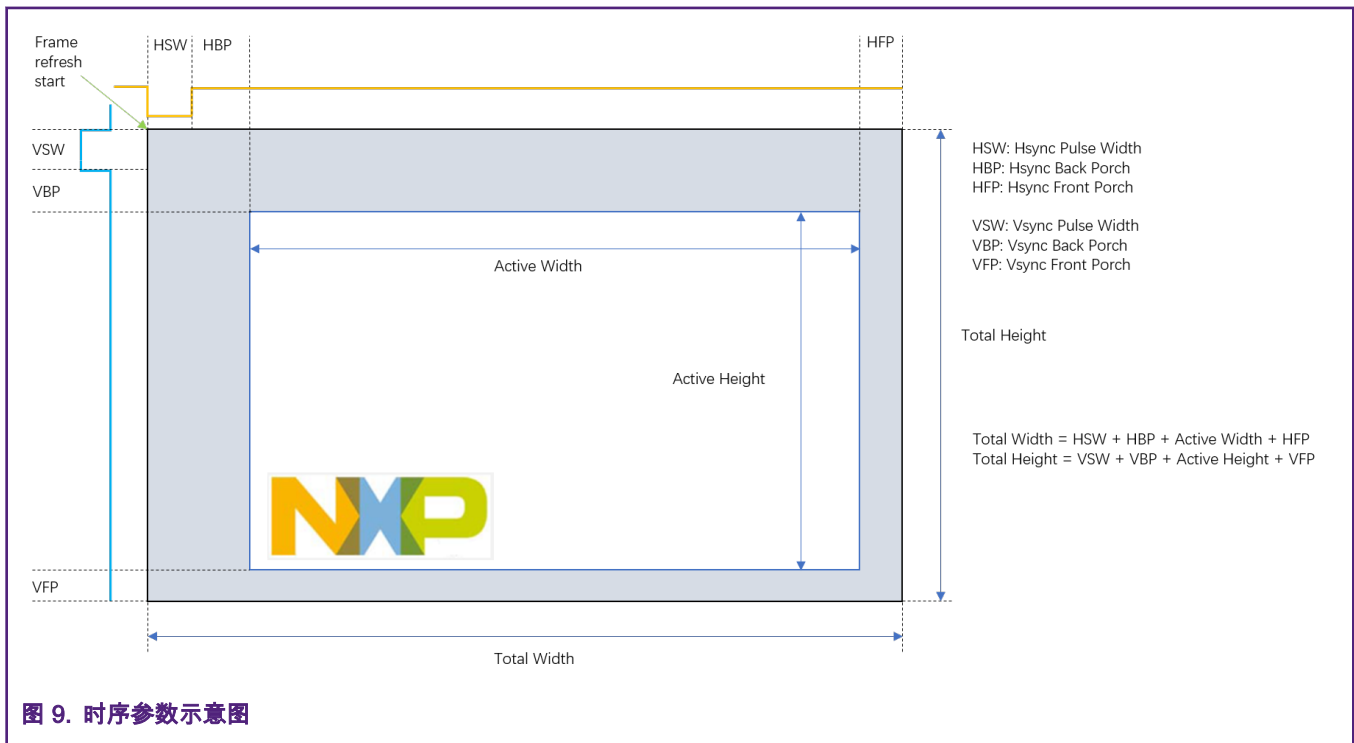


图 9. 时序参数示意图

表 2. 1280 × 800 LCD 屏的时序参数

| Parameter                        |                | Symbol | Min. | Type | Max. | Unit |
|----------------------------------|----------------|--------|------|------|------|------|
| DCLK frequency@ Frame rate=60 Hz |                | DCLK   | 68.9 | 71.1 | 73.4 | MHz  |
| Hsync                            | Period time    | th     | 1340 | 1440 | 1470 | DCLK |
|                                  | Display period | thd    | —    | 1280 | —    | DCLK |
|                                  | Back porch     | thb    | —    | 80   | —    | DCLK |
|                                  | Front porch    | thfp   | —    | 70   | —    | DCLK |
|                                  | Pulse width    | thpw   | —    | 10   | —    | DCLK |
| Vsync                            | Period time    | tv     | 815  | 823  | 833  | H    |
|                                  | Display period | tvd    | —    | 800  | —    | H    |
|                                  | Back porch     | tvb    | —    | 10   | —    | H    |
|                                  | Front porch    | tvfp   | —    | 10   | —    | H    |
|                                  | Pulse width    | tvpw   | —    | 3    | —    | H    |

主机控制器通过 RGB 接口持续高速的向 LCD 发送数据，利用 VSYNC，HSYNC，ENABLE 和 DOTCLK 信号以确保数据的同步。图 10 给出了显示一帧数据的全过程。

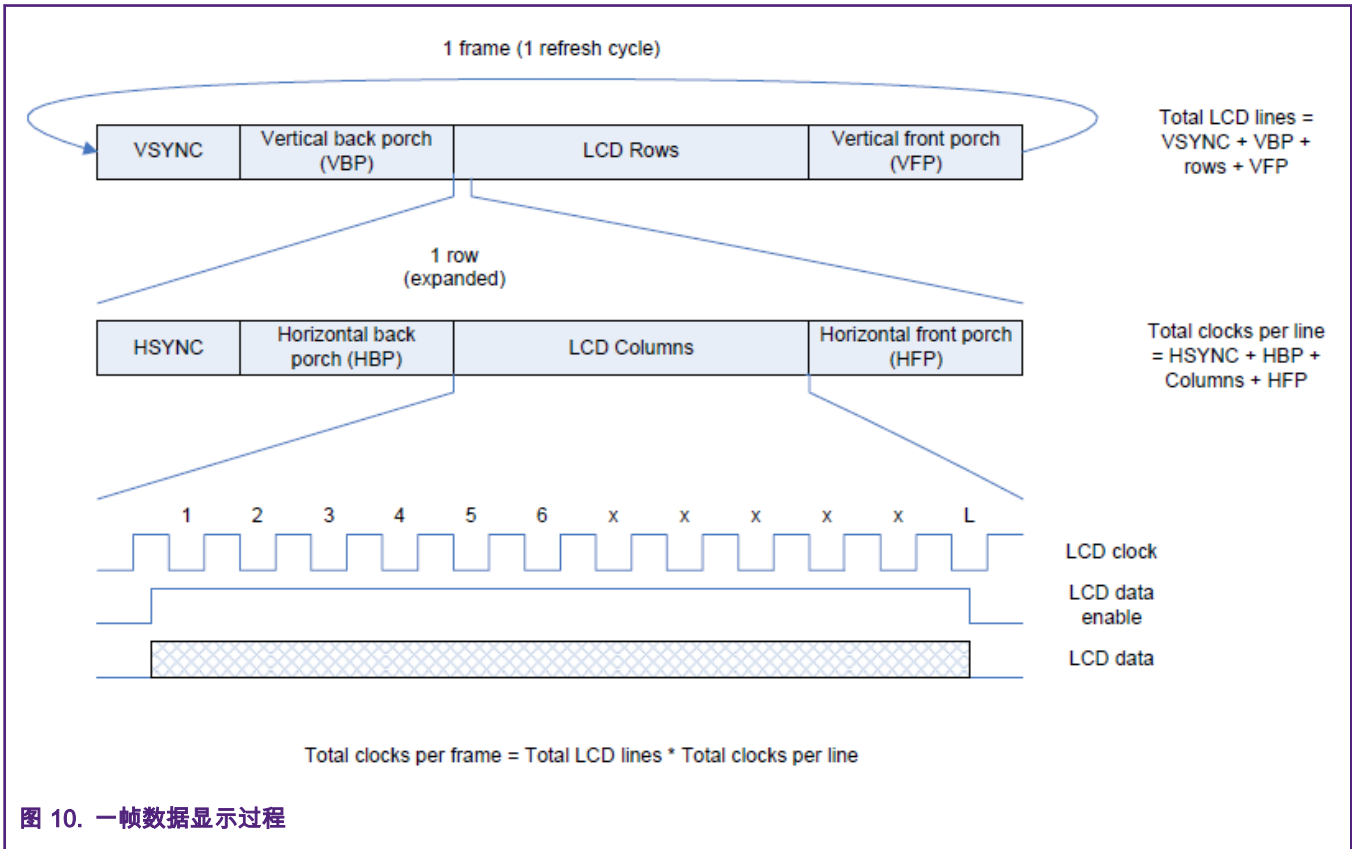


图 10. 一帧数据显示过程

### 3.2 LCDIFv2 相关信号

由 RGB 接口可知，为驱动图像正确显示在 LCD 屏上，LCDIFv2 控制器需要提供数据以及相应的控制信号。表 3 列出了详细的信号及相关描述。在 RT1170 中，LCDIFv2 控制器是通过 DPI-2 接口（也即 RGB 接口）将这些信号输出到 MIPI DSI 控制器的。

表 3. 详细信号描述

| 信号           | 方向  | 描述                |
|--------------|-----|-------------------|
| DCLK         | OUT | 驱动数据的像素时钟。        |
| VSYNC        | OUT | 帧同步信号，表示新的一帧的起始点。 |
| HSYNC        | OUT | 行同步信号，表示新的一行的起点。  |
| ENABLE       | OUT | 数据使能信号，表示有效像素数据。  |
| TXDATA[23:0] | OUT | 像素数据。             |

通常，驱动 LCD 屏还需要包含其他信号，这些信号不属于表 3 中所述的 LCDIFv2 信号，但也是 LCD 屏必需的。LCDIFv2 控制器只能驱动表 3 中所述的信号。一般使用 GPIO 来管理其他信号。LCD 屏通常内嵌一个背光单元，该背光单元需要一个额外的背光控制电路和 GPIO 来控制。此外，某些 LCD 屏需要通 I<sup>2</sup>C 或 SPI 来实现触屏功能。

### 3.3 LCDIFv2 相关时钟

为进行控制，LCDIFv2 控制器需要配置相应的时钟，LCDIFv2 主要需要四个输入时钟：

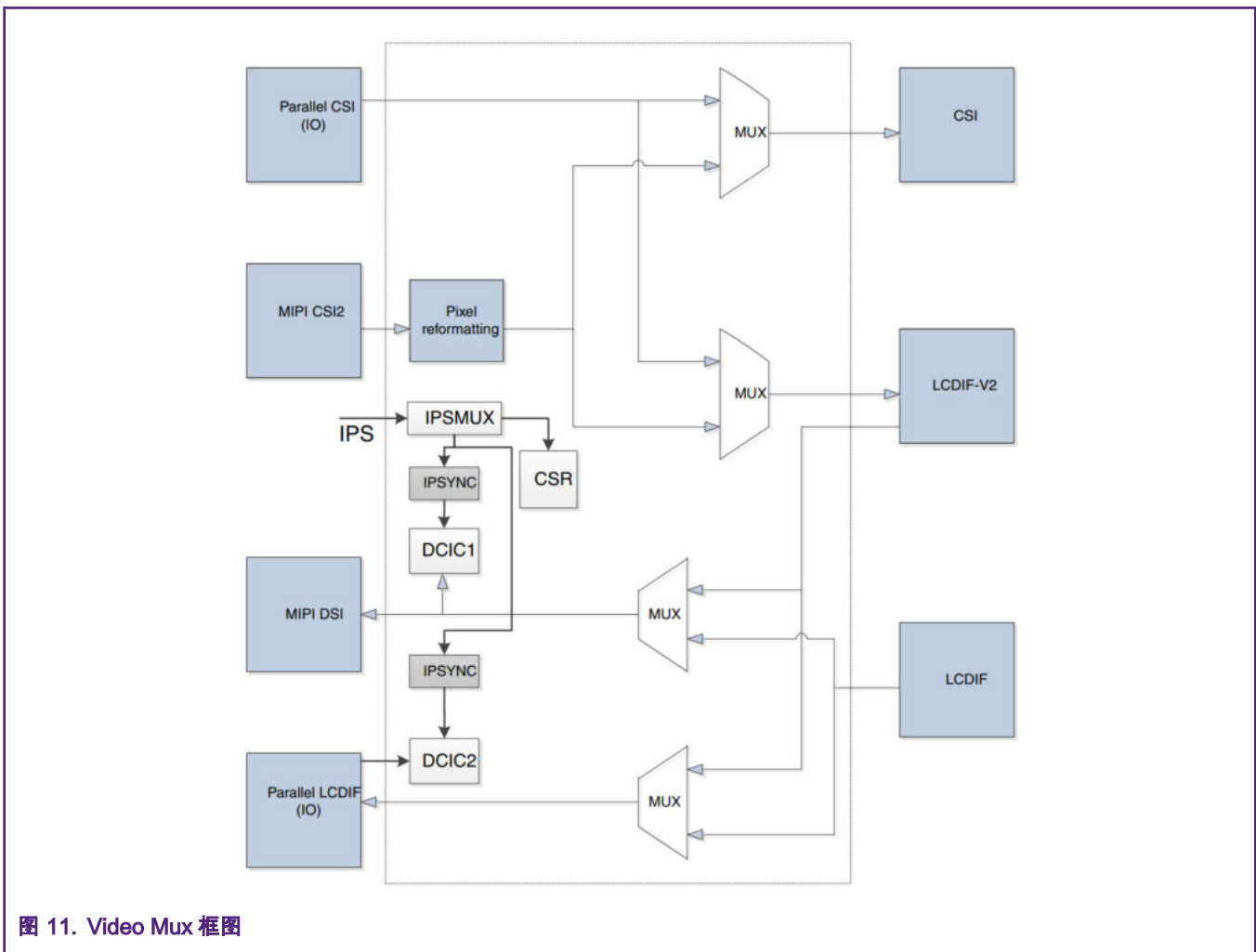
- `apb_clk`, `b_clk` – 两个来自 SOC 的相同时钟，用于寄存器和总线访问。
- `pix_clk` – 像素时钟，根据具体的 LCD 屏的分辨率参数进行配置。
- `pdi_clk` – 和像素时钟数值相同，专用于 LCDIFv2 的 PASS\_THROUGH 模式。

LCDIFv2 控制器的 `pix_clk` 和 `b_clk` 之间没有频率限制，但是为了获得更好的性能，设置较高的 `b_clk` 可以为系统内存中提供更大的可用带宽。

### 3.4 Video Mux 控制器

RT1170 支持多种 LCD 或视频接口，同时也支持两种显示控制器，video mux 控制器提供了在这些接口和控制器之间进行选择的功能。

图 11 给出了 video mux 的功能框图。例如，可以选择由 LCDIFv2 控制器来控制 MIPI DSI 接口输出。



## 4 示例运行

RT1170 SDK 包中有一个 `sd_jpeg` 例程，该例程展示了 LCDIFv2 控制器和 MIPI DSI 控制器的具体功能以及使用方法。SDK 包可以从 NXP 官网上进行下载。

## 4.1 sd\_jpeg 例程

SDK 包中，`sd_jpeg` 例程位于 `boards\evkmimxrt1170\jpeg_examples\sd_jpeg` 下。该例程从 SD 卡中读取 JPEG 图片，解码后一张一张显示在 LCD 屏上。

在运行例程之前，需要修改堆和栈的大小，为  $720 \times 1280$  的 LCD 屏分配更大的帧缓冲区。图 12 给出了栈和堆大小的更改示例，具体要根据实际需求进行修改。

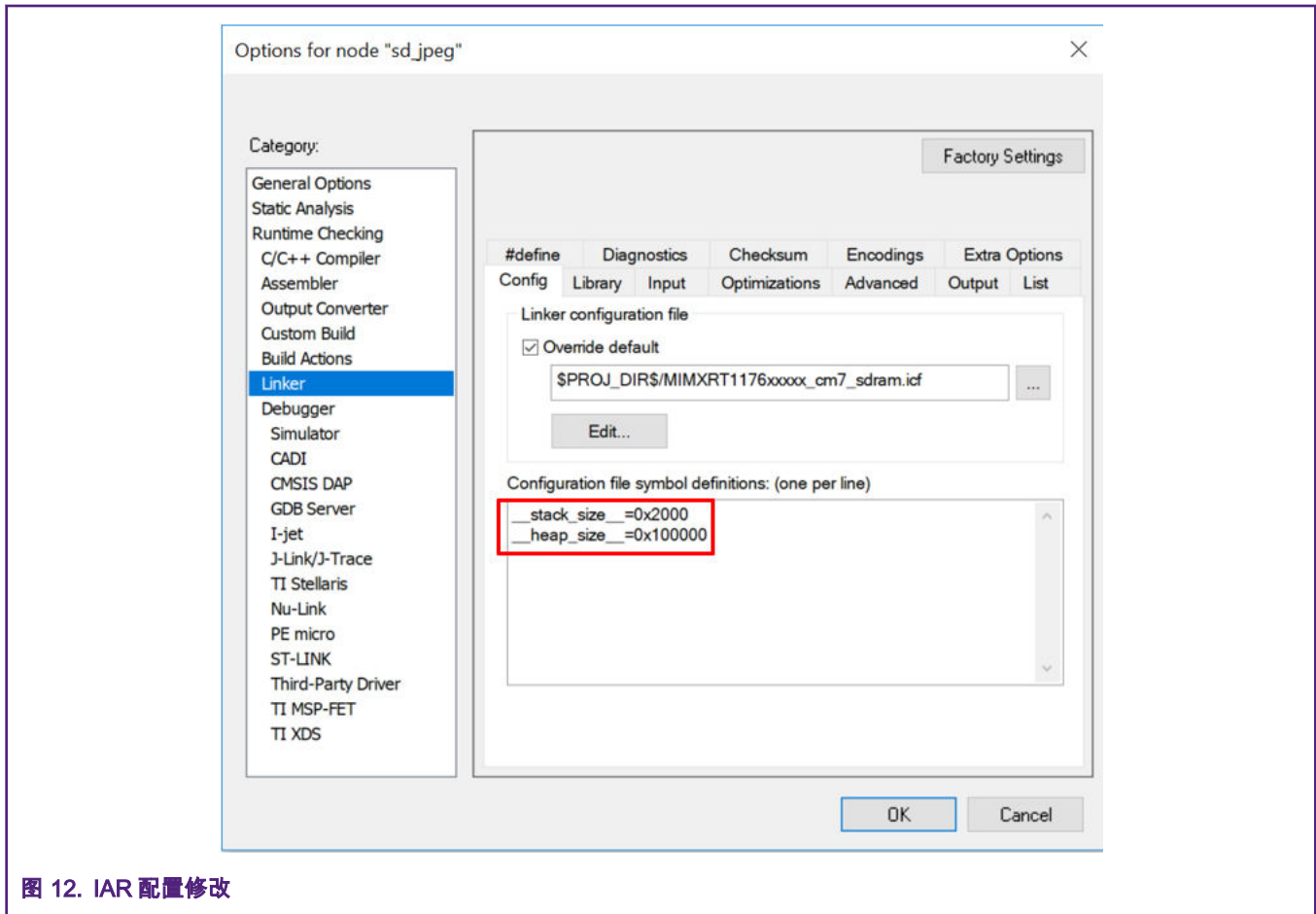


图 12. IAR 配置修改

编译，下载，运行例程到 RT1170-EVK 板上以驱动  $720 \times 1280$  的 LCD 屏。该屏通过 MIPI DSI 接口连接 EVK 板上，然后 SD 卡中的图片就会一张一张的显示在 LCD 屏上，如图 13 所示。



图 13. RT1170-EVK 和 720 × 1280 LCD 屏

为了使用 LCDIFv2 控制器和 MIPI DSI 控制器来驱动 DSI 兼容的 LCD 屏，用户应用程序应执行以下步骤：

- 配置 LCD 屏的电源，复位，背光引脚信号。
- 按 LCD 屏手册配置相关时序参数以及信号极性等。

例如，sd\_jpeg 例程配置 LCD 分辨率及时序参数如下所示：

```
#define DEMO_PANEL_WIDTH (720)
#define DEMO_PANEL_HEIGHT (1280)
#define DEMO_HSW 8
#define DEMO_HFP 32
#define DEMO_HBP 32
#define DEMO_VSW 2
#define DEMO_VFP 16
#define DEMO_VBP 14
#define DEMO_POL_FLAGS \
    (kLCDIFV2_DataEnableActiveHigh | kLCDIFV2_VsyncActiveLow | kLCDIFV2_HsyncActiveLow |
     kLCDIFV2_DriveDataOnRisingClkEdge)
```

- 配置 Video Mux 控制器相关寄存器。

例如，sd\_jpeg 例程定义了利用 LCDIFv2 控制器来控制 MIPI DSI 输出：

```
CLOCK_EnableClock(kCLOCK_Video_Mux);
VIDEO_MUX->VID_MUX_CTRL.SET = VIDEO_MUX_VID_MUX_CTRL_MIPI_DSI_SEL_MASK;
```



- 计算并配置像素时钟，MIPI DSI 的 RxClkEsc, TxClkEsc 时钟以及 DPHY PLL 模块的参考时钟。像素时钟由公式  $(height + VSW + VFP + VBP) \times (width + HSW + HFP + HBP) \times refresh\ rate$  计算得出。

图 14 给出了 `sd_jpeg` 例程中的用到的主要时钟。

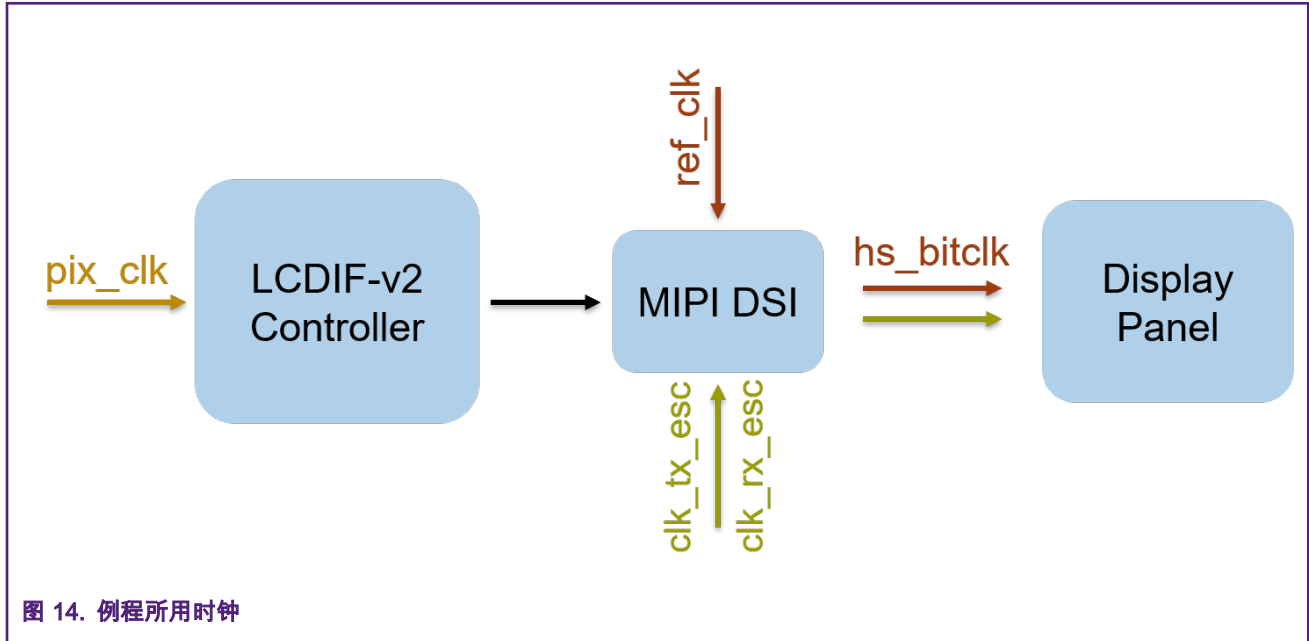


图 14. 例程所用时钟

`sd_jpeg` 例程中对  $720 \times 1280$  的 LCD 屏的详细时钟配置如下：

```
/* pix_clk = 528MHz / 9 = 58MHz. */
/* clk_rx_esc = 528MHz / 11 = 48MHz. */
/* clk_tx_esc = 528MHz / 11 / 3 = 16MHz. */
/* ref_clk = 24MHz / 1 = 24MHz. */
```

- 计算并配置 DPHY 高速传输时钟， $hs\_bitclk = TxByteClkHS \times 8$ 。 `hs_bitclk` 是由 DPHY PLL 模块通过 `ref_clk` 生成的，且速率必须足够快才能在 LCD 屏上刷新正确的图像数据，它的值必须大于以下数值：

$$(pix\_clk \times bit\ per\ output\ pixel) / number\ of\ MIPI\ data\ lane$$

- 配置 DSI 模块，DPHY 模块以及 DPI-2 模块的其他相关参数。
- 配置 TFT LCD 屏的内置显示驱动，例如本示例使用的 LCD 屏的显示驱动为 RM68200。

通过上述的配置，LCD 屏就能够正确的显示图像或 GUI 了。

## 4.2 LCD 刷新率

LCD 刷新率是指每秒刷新整个 LCD 屏的次数。一般来说，刷新率的典型值是 60 Hz，较低的刷新率可能会导致不良的视觉效果。LCD 刷新率与 LCD 屏和 MCU 的性能息息相关，本节使用 `sd_jpeg` 例程来测试一下 RT1170 支持的最大刷新率。

LCD 屏的刷新率由以下公式计算得到：

$$refresh\ rate = pix\_clk / ((height + VSW + VFP + VBP) \times (width + HSW + HFP + HBP))$$

为了测试最大刷新率，采用不断提高像素时钟 (`pix_clk`)，直到 LCD 屏无法显示正确的图像为止的测试方法。

测试结果表明当像素时钟提高至 88 Mhz 时，LCD 屏无法显示正确的图像，因此，对于  $720 \times 1280$  的 LCD 屏来说，当前显示系统支持的最大刷新率为 84 Hz。

## 5 参考资料

1. i.MX RT1170 Processor Reference Manual (Rev. E, 12/2019)

2. *i.MX RT eLCDIF RGB Mode Use Case* (document [AN12302](#)).

## ***How To Reach Us***

### **Home Page:**

[nxp.com](http://nxp.com)

### **Web Support:**

[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 08/2020

Document identifier: AN12940

