

# LCD 之 mipi DSI 接口驱动调试流程

标签: [androidlcd 框架 mipiDSI](#)  
2017-03-31 22:152004 人阅读 [评论](#) (0) [收藏](#) [举报](#)

分类:  
Android 之 LCD 开发 (11)  
版权声明: 本文为博主原创文章, 未经博主允许不得转载。

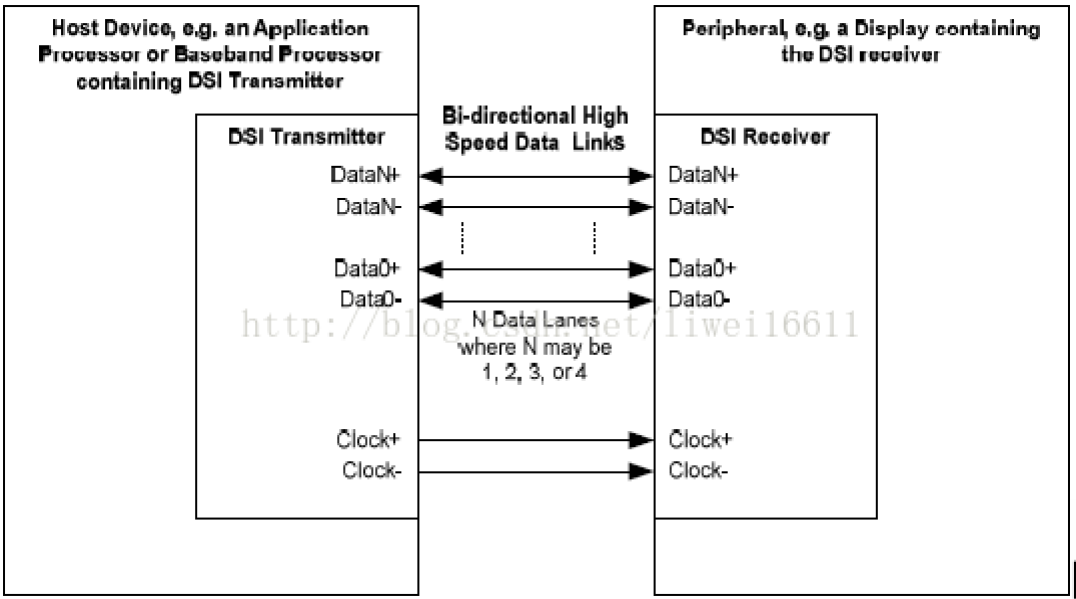
[目录](#) [\(?\)](#) [\[+\]](#)

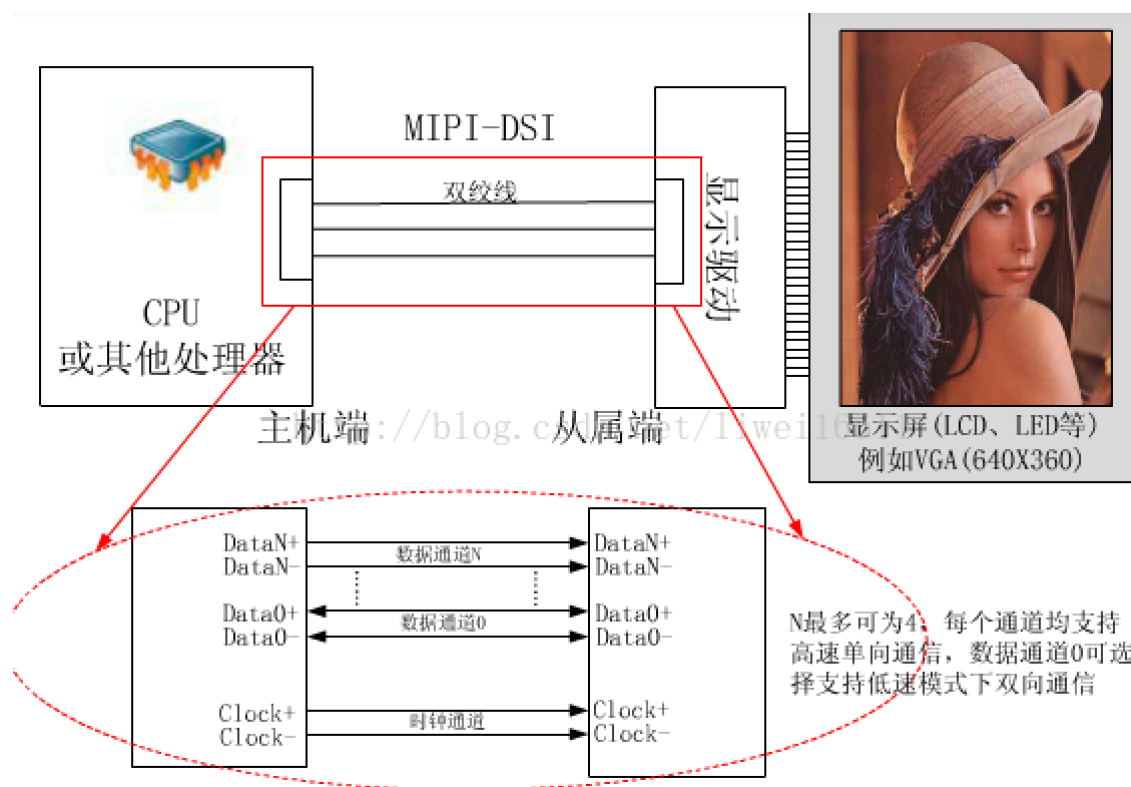
## 1、LCD MIPI DSI 协议

MIPI-DSI 是一种应用于显示技术的串行接口, 兼容 DPI (显示像素接口, Display Pixel Interface)、DBI (显示总线接口, Display Bus Interface) 和 DCS (显示命令集, Display Command Set), 以串行的方式发送像素信息或指令给外围, 而且从外围中读取状态信息或像素信息, 而且在传输的过程中享有自己独立的通信协议, 包括数据包格式和纠错检错机制。

MIPI-DSI 具备高速模式和低速模式两种工作模式, 全部数据通道都可以用于单向的高速传输, 但只有第一个数据通道才可用于低速双向传输, 从属端的状态信息、像素等式通过该数据通道返回。时钟通道专用于在高速传输数据的过程中传输同步时钟信号。此外, 一个主机端可允许同时与多个从属端进行通信。

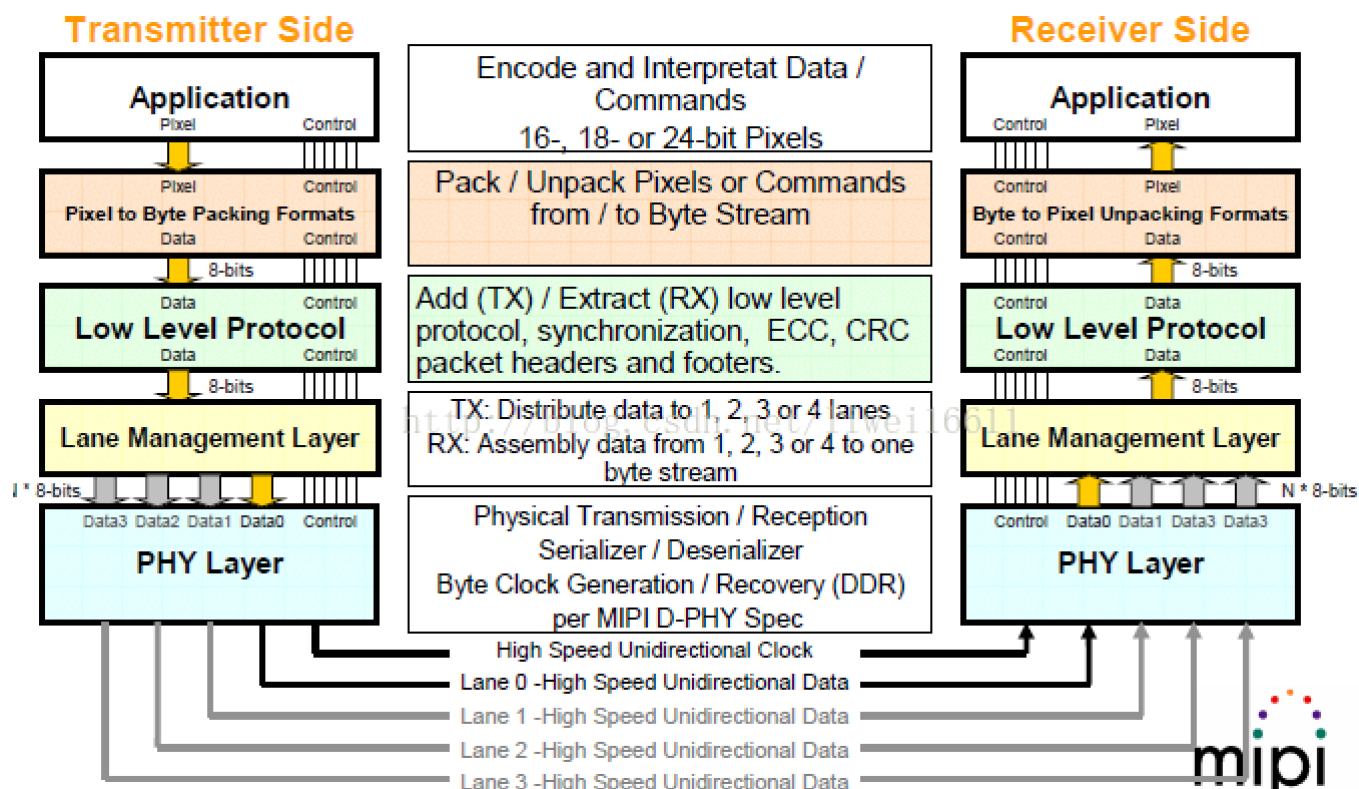
LCD mipi 系统简单示意图如下:

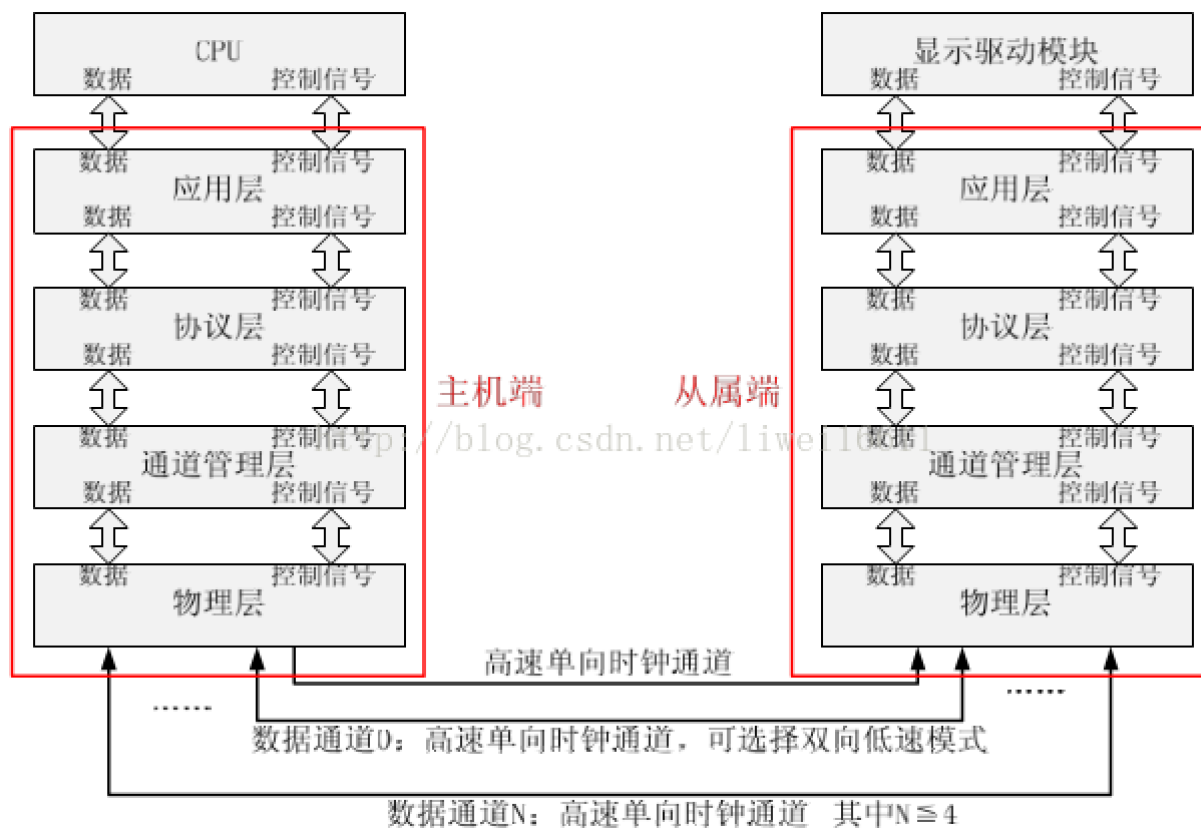




## 1.1、mipi DSI 协议分层

主要分为四个层：应用层、协议层、通道管理层、物理层





## 1.2、物理层

在接收到低速发送请求时，逻辑电路将待发送的指令或数据串行化，并控制输出高摆幅低功耗电平进行低功耗传送请求，随后将指令或数据串行地加载到通道，接收端则识别传送模式完成串行数据的并行转换。

当接收到高速请求后，电路发送高摆幅低功耗电平进行高速传送请求，然后将待发送数据经过串行转换之后驱动差分驱动模块以低摆幅信号在通道上传送，接收端则识别高速模式后正确地检测其同步信号将高速串行数据完成串并转换成功地接收。

除此之外，总线控制权可在主机端与从属端之间交换，发送端发送放弃总线请求，接收端接收请求之后控制总线发起应答，总的过程为TA(turn around)操作，在LCD中我们称之为BTA (Bus Turn-Around)。当主机端需要获取从属端接收数据的状况或者读取显示模块的参数或显示数据的时候，主机端发起TA操作，在从属端发送数据完毕之后自动发起TA操作，归还总线控制权。

信号两种工作状态，高速模式状态 (LP) 和低速模式状态 (HS)。

物理层，就是要控制硬件信号的 (时序)，MIPI-DSI 两种信号类型：

时钟 lane, 两根线一个 Dp, Dn;

数据 lane, 两根线一个 Dp, Dn;

数据 lane 根据需求有多个，最少一个 lane (lane0)，最多 4 个 lane (lane0, lane1, lane2, lan3)，其中只有 lane0 是双向的 (低速模式双向，高速模式也只能是单向)，其他的都只能是单向 (高速模式单向)。

数据 lane 电平：

低速模式 LP: 0-1.2V，空闲电平 (LP11) Dp, Dn 是 1.2V。

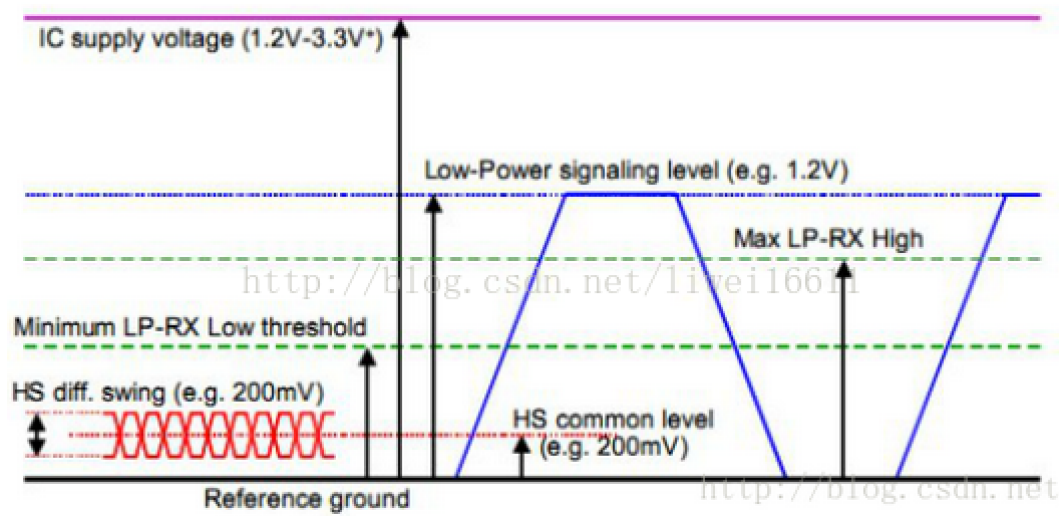
高速模式 HS: 100-300mV (200mV)

按照 D-PHY 协议，在整个协议的物理层中，在主机端和从属端之间采用的是同步连接，时钟通道用于传送高速时钟，

一个或多个数据通道用于传送低功耗数据信号或高速数据信号每一个通道都是利用两根互连线实现主机端和从属端的连接，并且支持高速模式和低速 ESCAPE 模式。

在高速模式下，发送端同时驱动该通道仅有的两根互连线，输出低摆幅差分信号，例如 200mV；

低速模式下，发送端分别驱动互连线，各自输出单端信号，但摆幅相对较大，例如 1.2V，如下图所示：



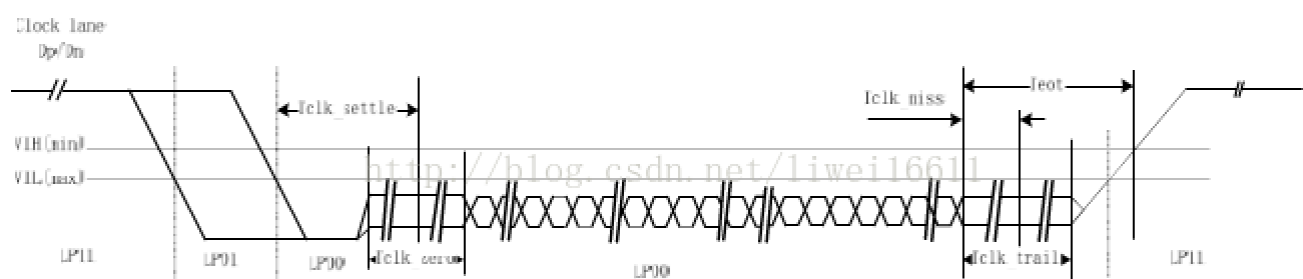
差分发送模块 (HS\_TX) 以差分信号驱动互连线，高速通道上呈现两种状态：differential-0 和 differential-1。低功耗单端发送模块 (LP\_TX) 独立地驱动两根互连线，通道上则有四种不同的状态：LP00、LP01、LP10 和 LP11。协议针对线路电平作了具体的定义并设置了三种工作模式：高速模式、控制模式和 ESCAPE 模式，通道默认为控制模式，线路处于 STOP 状态。当通道需要进入高速模式或 ESCAPE 模式，发送模块则需驱动线路为 LP01 或 LP10 向接收端发送请求，发送完请求序列之后便进行相应工作模式，发送 LP11 便可退出结束传送到 STOP 控制状态。接收端需要时刻进行 LP-RX 接收，侦测线路电平明确通道的工作模式。

state code	Line voltage level		High-speed	Low power	
	Dp-line	Dn-line	Burst mode	Control mode	Escape mode
HS-0	HS low	HS high	differential-0	N/A	N/A
HS-1	HS high	HS low	differential-1	N/A	N/A
LP-00	LP low	LP low	N/A	bridge	Space
LP-01	LP low	LP high	N/A	HS-Rqst	Mark-0
LP-10	LP high	LP low	N/A	LP-Rqst	Mark-1
LP-11	LP high	LP high	N/A	STOP	N/A

1.2.1、时钟通道

时钟通道支持高速模式和 ULPS（超低功耗模式）模式：

高速模式应用于整个接口工作于高速数据传送模式，用于传送高速时钟：

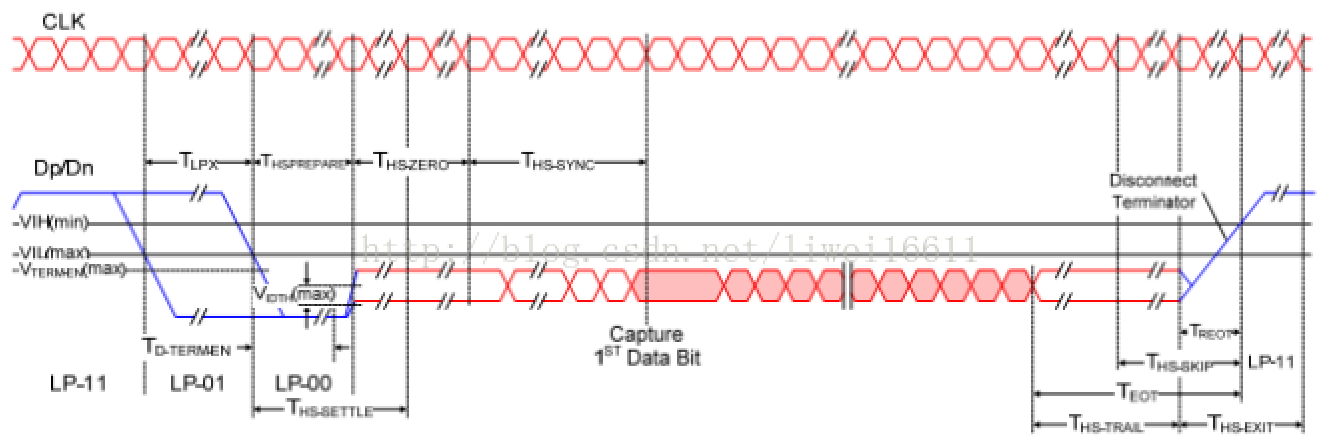


ULPS 模式用于外围（如整个显示模块）要进入休眠模式时将时钟通道置于超低功耗状态，节省外围接口功耗。

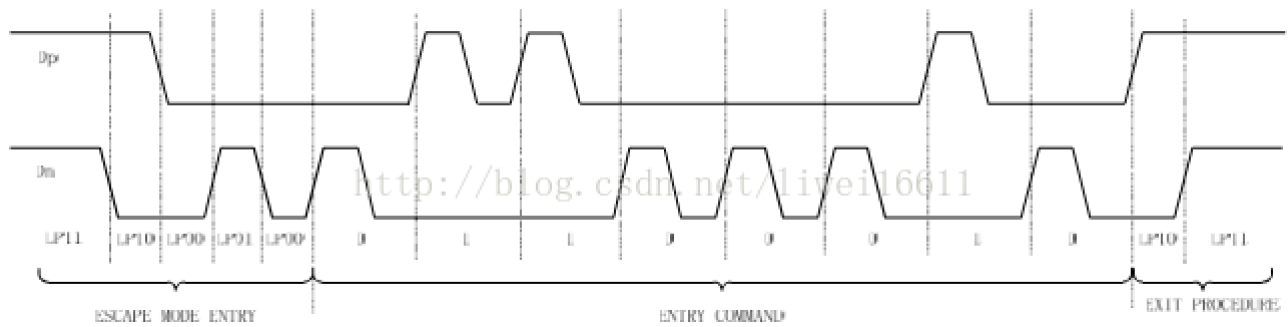


1.2.2、数据通道

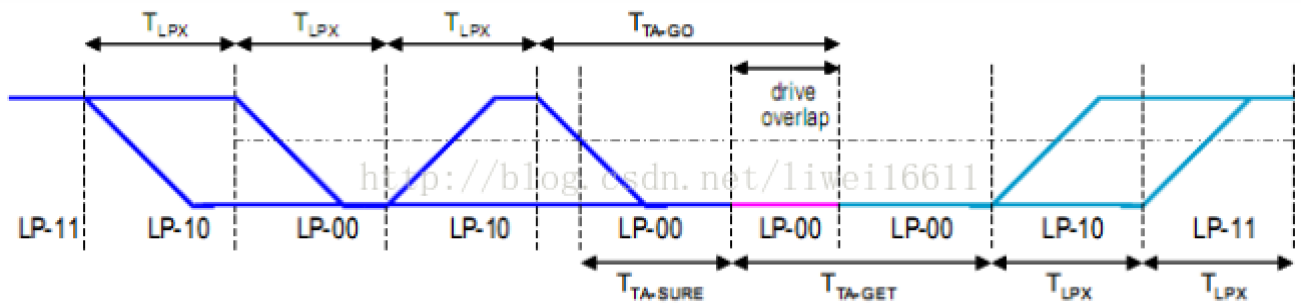
数据通道支持高速模式和 ESCAPE 模式，在此还包括另外一个重要的功能总线控制权交换——BTA 模式。高速模式是以高速的低摆幅差分信号传送数据，速度快，最高可达 1Gps。在经过请求序列 LP11→LP01→LP00 之后，HS\_TX 开始发送串行高速数据，发送结束之后，LP\_TX 发送 LP11 回到正常的控制模式。



ESCAPE 模式是由 LP\_TX 直接驱动线路通道发送电平序列传送低功耗数据。其进入方式为 LP11→LP10→LP00→LP01→LP00，退出方式为 LP10→LP11。经过请求序列之后，LP\_TX 发送模式命令，包括 trigger 命令和低功耗数据传送（LPDT）等，如下如所示：



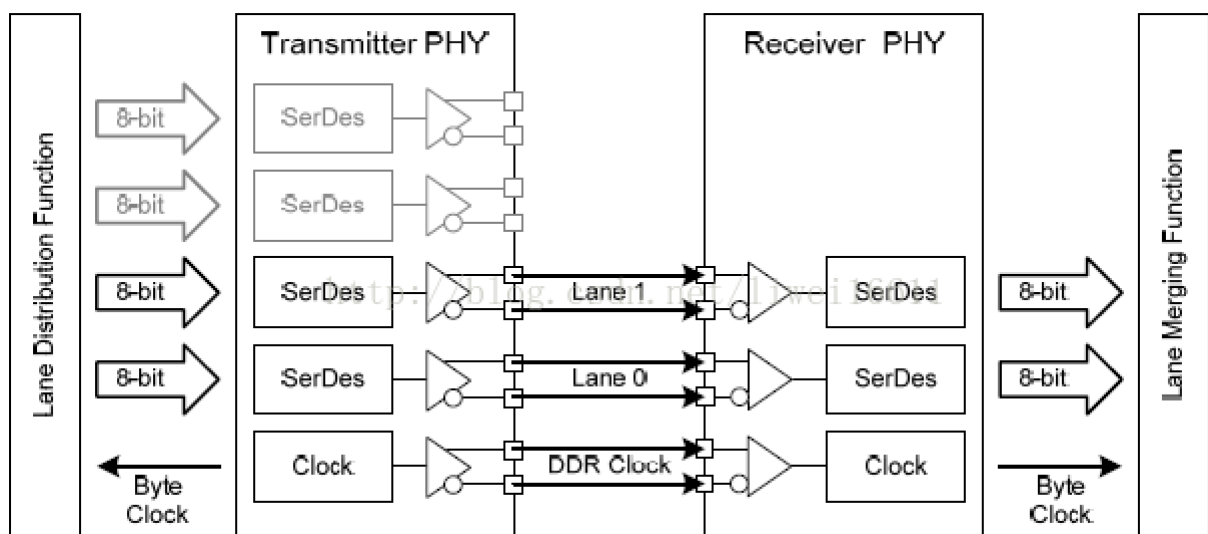
BTA 模式用于交换总线控制权，在控制模式下完成。LP\_TX 发送请求序列 LP11→LP10→LP00→LP10→LP00，经过 BTA-Go 之后停止驱动总线。相对于总线另一端来说，在接收到请求序列并且经过 BTA-SURE 之后，LP\_TX 开始驱动总线，发送确认信号，接管总线。



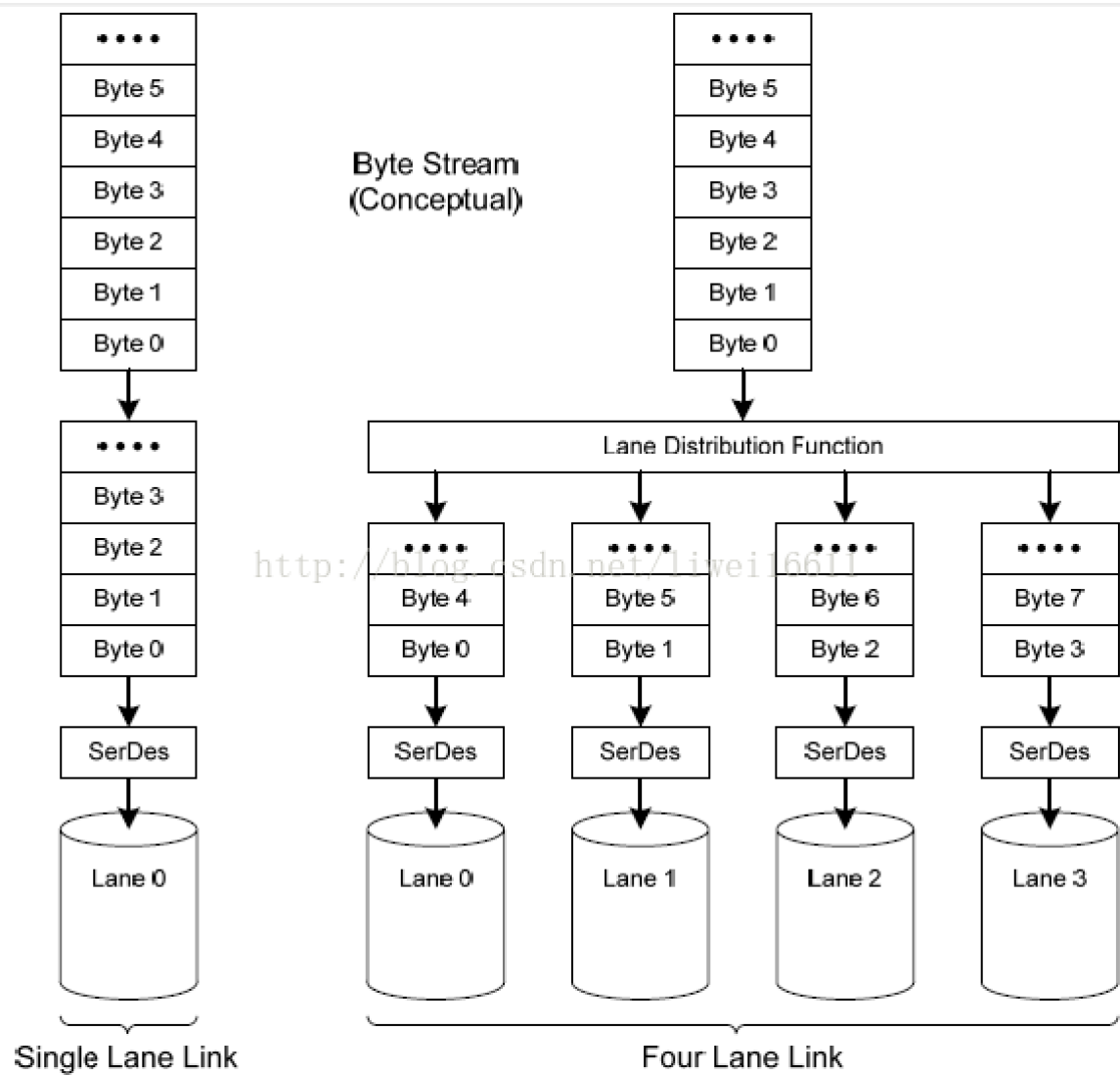
### 1.3、通道管理层

依据设计所设定最多为 4 个数据通道，该层在发送端将需要发送的数据按照通道次序分成 N 组输送到相应的数据通道，使其经数据通道同时发送至从属端。在接收端，该层需要做的是将接收到的 N 组数据组合在一起恢复成原始的数据序列。

数据分组与恢复如下图：

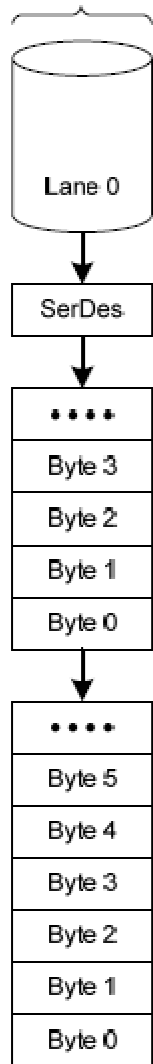


数据分组：

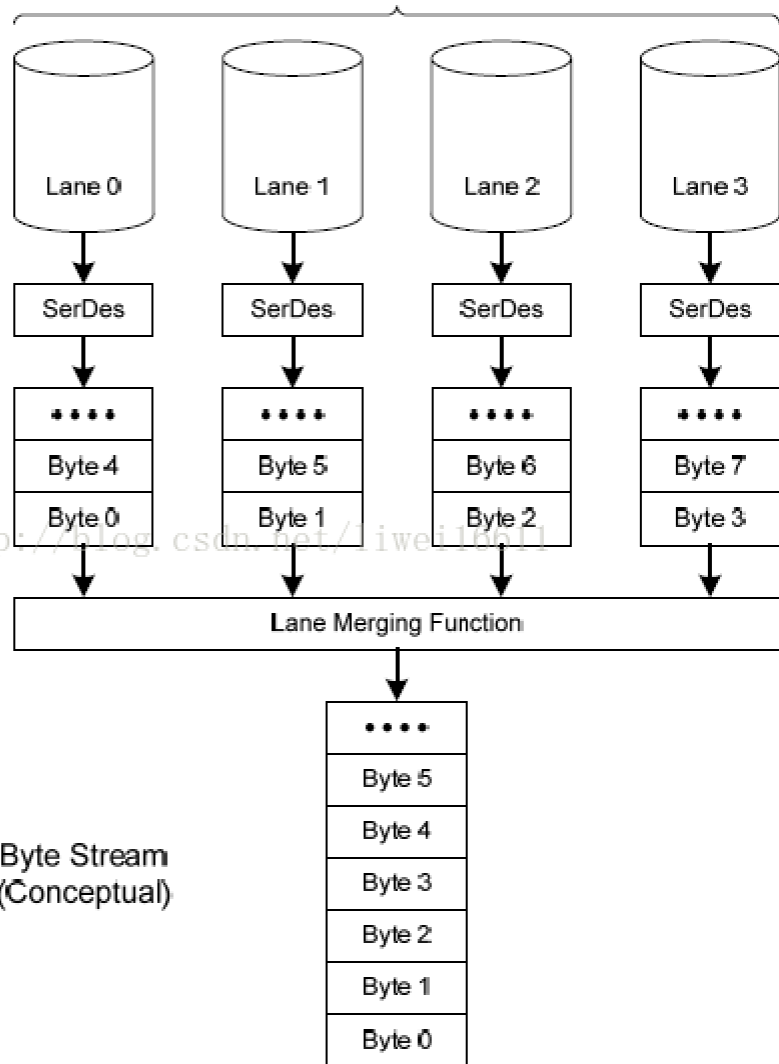


数据恢复:

### Single Lane Link



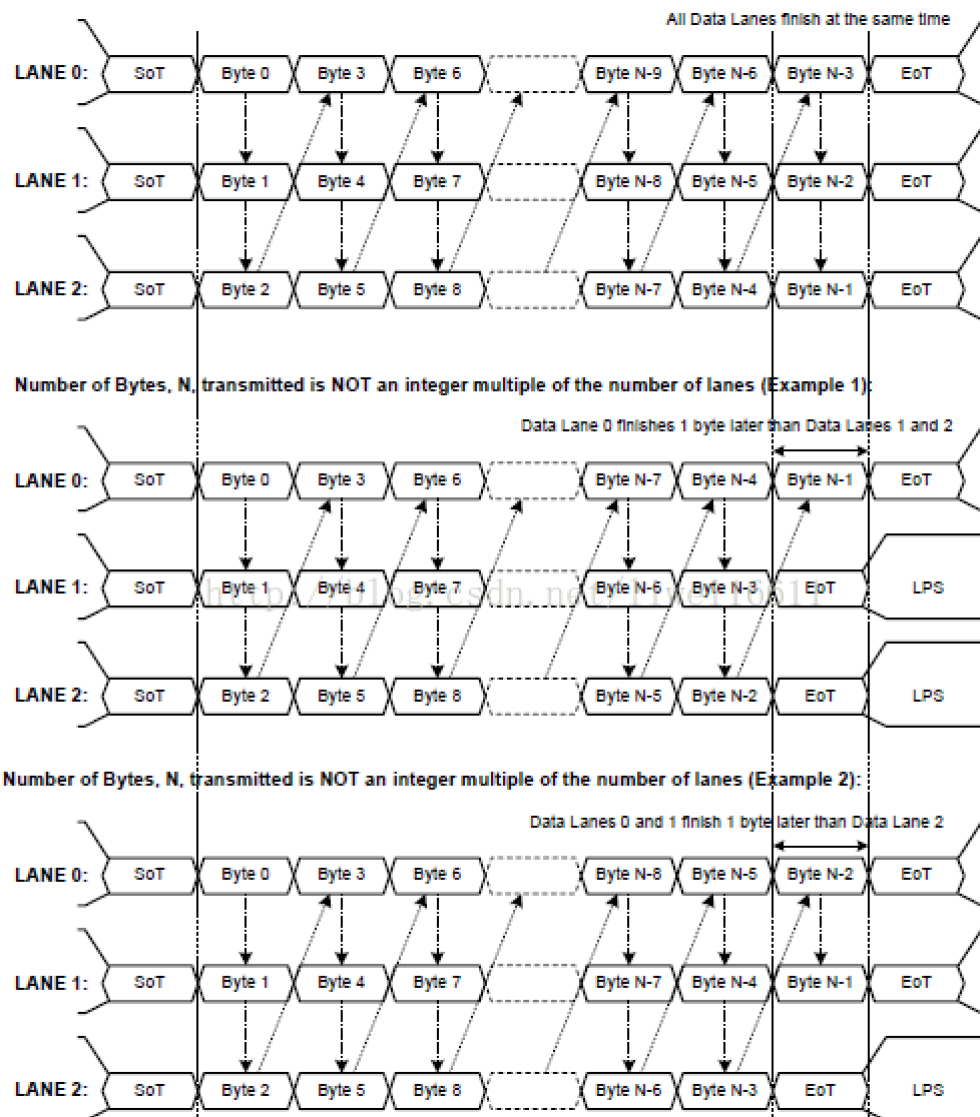
### Four Lane Link



Byte Stream  
(Conceptual)

数据操作过程：





#### 1.4、协议层

信息传送采用数据包格式，包括长数据包和短数据包。发送数据的时候，将数据按照信息类型及内容进行压包，完成 ECC 码的生成和 CRC 码的添加。接收数据的时候，依据 ECC 码和 CRC 码就整个数据包进行检错纠错，完成对包头和数据内容的译码并合理输送到应用层中。

DSI 协议是一个基于数据包传送的通信协议，主机端和显示模块之间传送的命令和数据基本上都以数据包格式进行。DSI 所定义的数据包有两种：端数据包 (short packet) 和长数据包 (long packet)，短数据包主要用于传输命令、读写寄存器，长数据包主要用于传输大量图象数据或部分控制命令。

##### short packet

LPS	SoT	Data ID	Data 0	Data 1	ECC	EoT	LPS
-----	-----	---------	--------	--------	-----	-----	-----

##### long packet

LPS	SoT	Data ID	Word Count (WC)	ECC	Data 0	Data 1	Data wc-2	Data wc-1	16-Bit Check Sum	EoT	LPS
Packet Header(PH)				Packet Data(PayLoad)					Packet Footer(PF)		

## 1.5、应用层

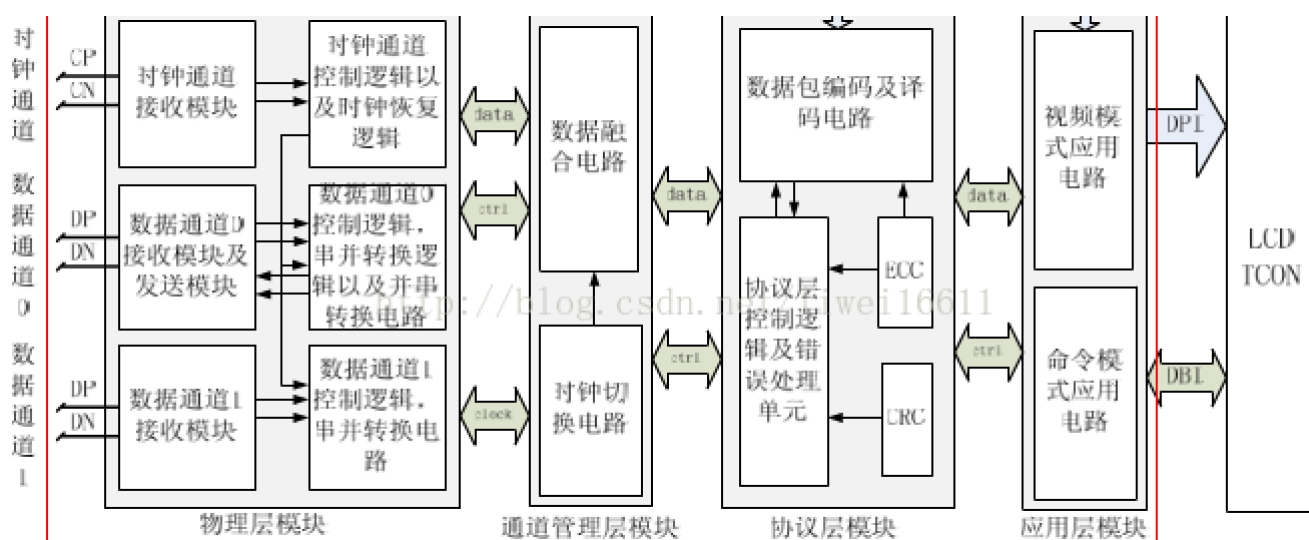
根据应用模块的需要，在发送端对发送的命令和数据进行初步编码转化为 MIPI-DSI 所规定的格式，而在接收端则将接收的数据还原为应用模块所支持的数据格式及时序要求。

应用层模块直接连接到显示模块，负责着与显示模块的通信。按照协议要求，接口电路支持命令模式和视频模式，命令模式则是经过接口转换之后主机端依然得以“命令+参数”格式对显示模块进行读写操作，而视频模式下，主机端可以直接将显示数据写入显示模块，无需命令操作，实时显示视频。该模块在设计的过程中应支持命令模式和视频模式。

根据协议所定义的数据类型的意义，其中一些数据类型是专门应用于 video 模式，如 0x01 指代的是场同步信号 (Vsync) 开始。所以在此模块电路的设计中，分成了视频模式应用电路和命令模式应用电路两个模块，视频模式应用电路模块仅接收视频模式相关的命令和数据并以 DPI 时序格式传送到驱动中。

cmd 模式应用电路模块适用于非视频模式的控制命令和数据，对其进行译码并以 DBI 时序格式写入到显示模块中的寄存器或 SRAM 中。当需要读取数据的时候，命令模式应用电路模块通过 DBI 口可从显示模块中读取相关寄存器的信息等。

mipi 总体框图：



## 2、MIPI DSI 实践

### 2.1、mipi 速率

$$\text{mipi\_clk} = \text{resX} * \text{resY} * \text{bps} * 1.25 * \text{fps} / \text{lane\_num}$$

注：bps = 3 \* 8 = 24 位

1.25 为余量

如：

分辨率：2560\*1440，fps = 65

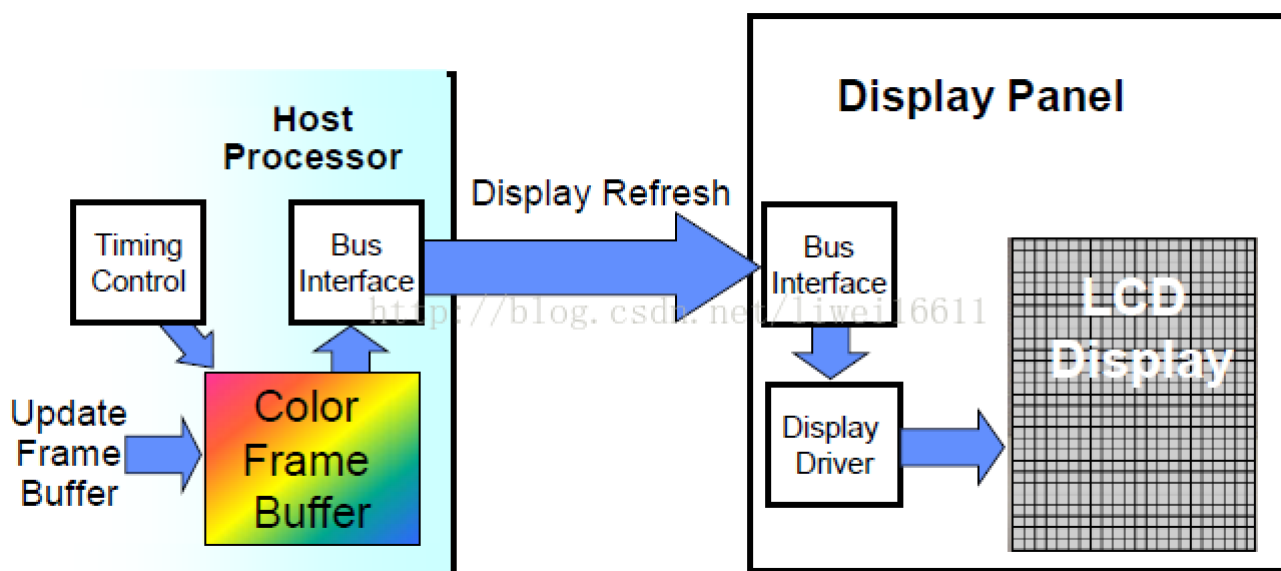
MIPI lane：2 条数据 lane，1 条时钟 lane

$$\text{mip\_clk} = 2560 * 1440 * 60 \text{fps} * (3 * 8) * 1.25 * 65 / 2 / 8 / 104 / 1024 = 429 \text{Mhz}$$

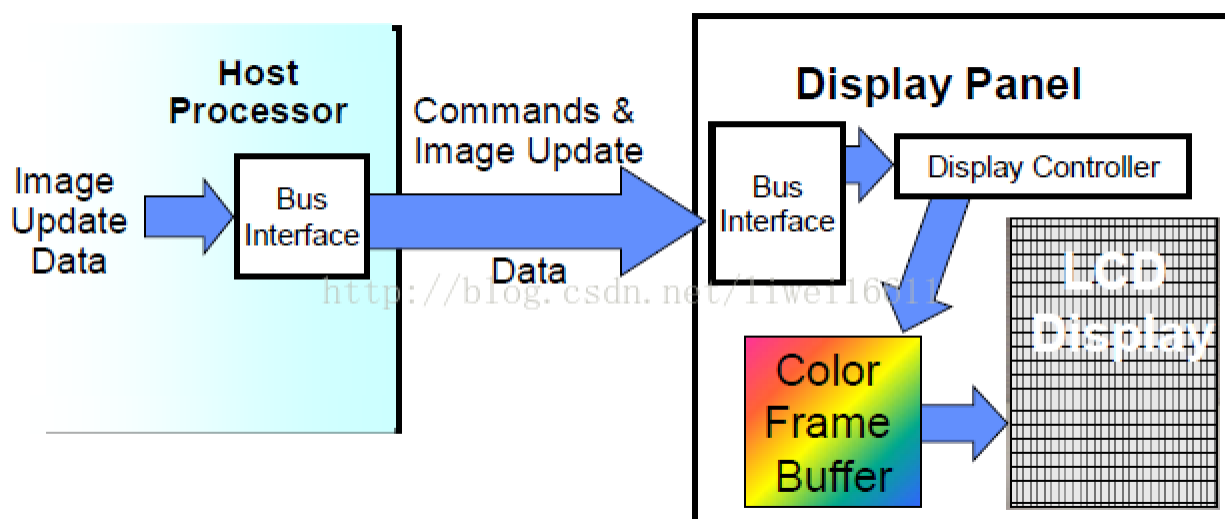
### 2.2、LCD dsi 简图

今天以 LCD MIPI 为例做简要讨论。Display 对应 MIPI DSI，Camera 对应 MIPI CSI。和 Display 相关的就是 MIPI DSI，DPI，DBI 等，规范了 host display controller 到 panel 之间通信时从物理层，链路层到应用层的协议。

DPI: Video-mode displays



DBI: Command-mode displays



## 2.3、mipi 导致花屏问题

### 2.3.1、切换屏幕画面，中间闪一条竖线

原因：mipi clk\_post 偏低

分析：data lane 在传输完成最后一帧数据后进入 lp 模式，clk lane 继续传输时钟信号的时间过短导致 datalane 未完全进入 lp11，状态不稳定。

解决：

- 1、增大 clk\_post
- 2、时钟改为 continue 模式，功耗大

### 2.3.2、LCD 右半边闪屏

原因：mipi clk 偏低，lk 阶段 clk 偏低，kernel 阶段 clk 正常

分析：数据包长短不一致会导致进入 lp11 时间点不一致，属于正常现象 --- 排除 1k 频率偏低导致问题

解决：修整频率

### 2.3.3、panel 电荷无法释放导致闪屏 --- 硬件设计

## 2.4、时序异常

### 2.4.1、IOVCC 上电瞬间，mipi 异常脉冲

此时 mipi 接口还没准备好，lp-tx 被打开，输出异常脉冲

### 2.4.2、reset 低电平时，mipi 信号被拉低

器件由于低功耗考虑，reset 为低时，进入深度待机模式，regulator 关闭，保护二极管分流，拉低信号。

## 2.5、mipi 测试

需要引出 clkp、clkn、data0p, data0n

参考：

### 1、mipi DSI

<http://blog.csdn.NET/redredbird/article/details/12648847>

<http://blog.csdn.net/eZiMu/article/details/56279847?locationNum=14&fps=1>

### 2、LCD 调试流程

<http://blog.csdn.Net/dearsq/article/details/52354593?locationNum=9&fps=1>

### 3、lcd DSI 论文

<http://www.doc88.com/p-0877384948107.html>