

mipi 调试经验

分享：
以下是最近几个月在调试 MIPI DSI / CSI 的一些经验总结，因为协议有专门的文档，所以这里就记录一些常用知识点：

一、D-PHY1、传输模式

LP (Low-Power) 模式：用于传输控制信号，最高速率 10 MHz
HS (High-Speed) 模式：用于高速传输数据，速率范围 [80 Mbps, 1Gbps] per Lane

传输的最小单元为 1 个字节，采用小端的方式及 LSB first, MSB last。

2、Lane States

* LP mode 有 4 种状态： LP00、LP01 (0)、LP10 (1)、LP11 (Dp、Dn)

* HS mode 有 2 种状态： HS-0、HS-1

HS 发送器发送的数据 LP 接收器看到的都是 LP00，

3、Lane Levels

* LP: 0 ~ 1.2V
* HS: 100 ~ 300mV, HS common level = 200mV, swing = 200 mv

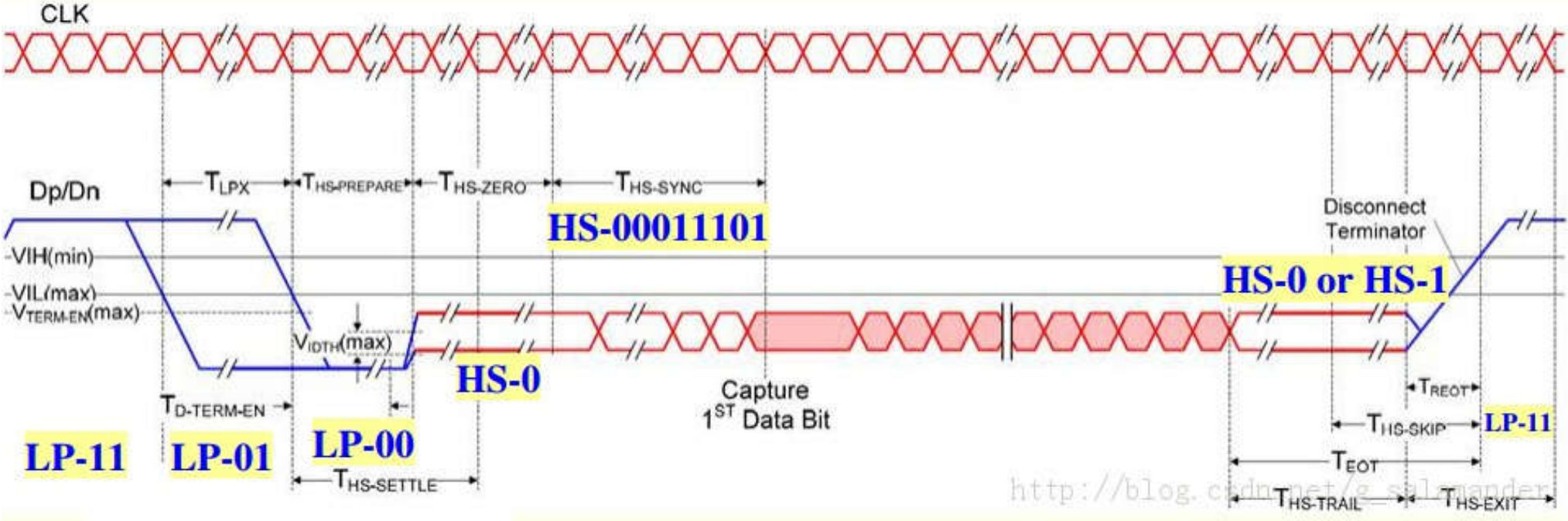
4、操作模式

在数据线上有 3 种可能的操作模式：Escape mode, High-Speed (Burst) mode and Control mode，下面是从停止状态进入相应模式需要的时序：

* Escape mode 进入时序：LP11→LP10→LP00→LP01→LP00，退出时序：LP10→LP11

当进入 Escape mode 需要发送 8-bit entry command 表明请求的动作，比如要进行低速数据传输则需要发送 cmd: 0x87，进入超低功耗模式则发送 cmd: 0x78。在 DSI 中 LP 通讯只用 Data Lane 0。

* High-Speed mode 进入时序：LP11→LP01→LP00→SoT(0001_1101)，退出时序：EoT→LP11，时序图如下：



* Turnaround 进入时序：LP11→LP10→LP00→LP10→LP00，退出时序：LP00→LP10→LP11

这是开启 BTA 的时序，一般用于从 slave 返回数据如 ACK: 0x84。

5、时序要求

在调试 DSI 或者 CSI 的时候， HS mode 下的几个时序非常重要：T_LPX, T_HS-SETTLE ≈ T_HS-PREPARE + T_HS-ZERO, T_HS-TRAIL，一般遵循的原则为：Host 端的 T_HS-SETTLE > Slave 端的 T_HS-SETTLE。

二、DSI1、线路构成

在 DSI 中需要 1 根时钟线以及 1 ~ 4 根数据线。

2、两种接口的 LCD

* Comman mode (对应 MPU 接口)

* Video mode (对应 RGB 接口)

该模式下视频数据只能通过 HS mode 传输。

3、数据包类型

短包：4 bytes，由 3 部分组成：

* Data Identifier (DI) * 1byte: Contains the Virtual Channel[7:6] and Data Type[5:0].

* Packet Data * 2byte: Length is fixed at two bytes

* Error Correction Code (ECC) * 1byte: allows single-bit errors to be corrected and 2-bit errors to be detected.

长包：6 ~ 65541 bytes，同样由 3 部分组成：

* Packet Header(4 bytes) - 包头

Data Identifier (DI) * 1byte: Contains the Virtual Channel[7:6] and Data Type[5:0].

Word Count (WC) * 2byte: defines the number of bytes in the Data Payload.

Error Correction Code (ECC) * 1byte: allows single-bit errors to be corrected and 2-bit errors to be detected.

* Data Payload(0~65535 bytes) - 有效数据

Length = WC × bytes

* Packet Footer(2 bytes): Checksum - 包尾

If the payload has length 0, then the Checksum calculation results in FFFFh

If the Checksum isn’ t calculated, the Checksum value is 0000h

4、从控制器到外设发送的包类型

Data Type		Description	Size	mode
01h	00 0001	Sync Event, V Sync Start (01, 00, 00, 07)	Short	Video mode only
11h	01 0001	Sync Event, V Sync End (11, 00, 00, 14)	Short	Video mode only
21h	10 0001	Sync Event, H Sync Start (21, 00, 00, 12)	Short	Video mode only
31h	11 0001	Sync Event, H Sync End (31, 00, 00, 01)	Short	Video mode only
02h	00 0010	Color Mode (CM) Off Command (02, 00, 00, 0B)	Short	Video mode only
12h	01 0010	Color Mode (CM) On Command (12, 00, 00, 18)	Short	Video mode only
22h	10 0010	Shut Down Peripheral Command (22, 00, 00, 1E)	Short	Video mode only
32h	11 0010	Turn On Peripheral Command (32, 00, 00, 0D)	Short	Video mode only
03h	00 0011	Generic Short WRITE, no parameters	Short	HX8357 dont support
13h	01 0011	Generic Short WRITE, 1 parameters	Short	HX8357 dont support
23h	10 0011	Generic Short WRITE, 2 parameters	Short	HX8357 dont support
04h	00 0100	Generic READ, no parameters	Short	HX8357 dont support
14h	01 0100	Generic READ, 1 parameters	Short	HX8357 dont support
24h	10 0100	Generic READ, 2 parameters	Short	HX8357 dont support
05h	00 0101	DCS WRITE, no parameters	Short	Command, Video
15h	01 0101	DCS WRITE, 1 parameters	Short	Command, Video
06h	00 0110	DCS READ, no parameters	Short	Command, Video
37h	11 0111	Set Maximum Return Packet Size	Short	Command, Video
08h	00 1000	End of Transmission (EoTp) packet (08, 0F, 0F, 01)	Short	Command, Video
0Eh	00 1110	Packed Pixel Stream, 16-bit RGB, 5-6-5 Format	Long	Video mode only
1Eh	01 1110	Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long	Video mode only
2Eh	10 1110	Loosely Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long	Video mode only
3Eh	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	Long	Video mode only
19h	01 1001	Blanking Packet, no data	Long	Video mode only
29h	10 1001	Generic Long Write	Long	HX8357 dont support
09h	00 1001	Null Packet, no data	Long	Command, Video
39h	11 1001	DCS Long Write/write LUT Command Packet	Long	Command, Video
x0h and xFh	xx 0000	DO NOT USE	http://blog.csdn.net/g_salamander	
unspecified	xx 1111	All unspecified codes are reserved		

如果希望从外设读取数据或者状态，则在处理器发送完读取命令后还需要发送 BTA 命令，非读取命令在外设接收成功后会返回 trigger message 0x84。

5、从外设到处理器数据包类型

Data Type (Hex)	Data Type (Binary)	Description	Packet Size
00h - 01h	00 000x	Reserved	short
02h	00 0010	Acknowledge and Error Report	short
03h - 07h	00 0011 - 00 0111	Reserved	
08h	00 1000	End of Transmission (EoT) packet	short
09h - 10h	00 1001 - 01 0000	Reserved	
11h	01 0001	Generic short READ Response, 1 byte returned	short
12h	01 0010	Generic short READ Response, 2 byte returned	short
13h - 19h	01 0011 - 01 1001	Reserved	
1Ah	01 1010	Generic Long READ Response	Long
1Bh	01 1011	Reserved	
1Ch	01 1100	DCS Long READ Response	Long
1Dh - 20h	01 1101 - 10 0000	Reserved	
21h	10 0001	DCS Short READ Response, 1 byte returned	short
22h	10 0010	DCS Short READ Response, 2 byte returned	short
23h - 3Fh	10 0011 - 11 1111	Reserved	

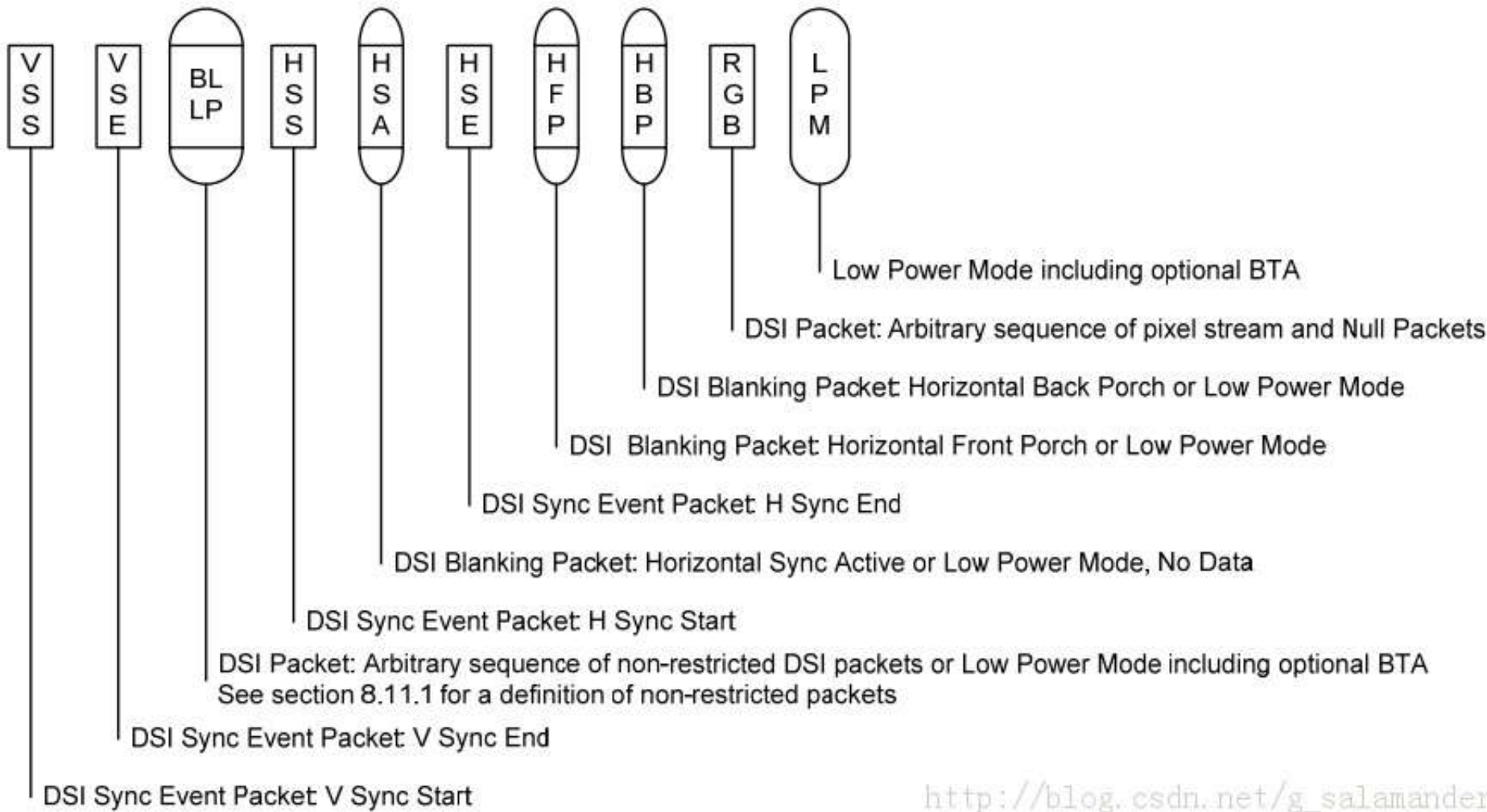
返回的数据一般分为 4 个类型：

- * Tearing Effect (TE)：trigger message (BAh)
 - * Acknowledge：trigger message (84h)
 - * Acknowledge and Error Report：short packet (Data Type is 02h)
 - * Response to Read Request：short packet or long packet
- Generic Read Response、DCS Read Response (1byte, 2byte, multi byte)

读取数据返回值解析示例如下：

1. - Acknowledge and Error report (if error occurs)
2. Byte 0 is 0x87 (escape mode low power data transmission header)
3. Byte 1 is 0x02 (Data type, 8.10 of “MIPI Alliance Specification for DSI”)
4. Byte 3,2 are error report bits[15:0] (8.9.5 of “MIPI Alliance Specification for DSI”)
5. Byte 4 is the ECC, calculated from byte 1,2,3
6. - Generic Short READ response
7. Byte 0 is 0x87 (escape mode low power data transmission header)
8. Byte 1 is 0x11 or 0x12 (8.10 of “MIPI Alliance Specification for DSI”)
9. Byte 2,3 are the read data. If only 1 byte is returned, byte 3 will be 0x00
10. Byte 4 is the ECC, calculated from byte 1,2,3
11. - Long READ packet response
12. Byte 0 is 0x87 (escape mode low power data transmission header)
13. Byte 1 is 0x1A (8.10 of “MIPI Alliance Specification for DSI”)
14. Byte 3,2 are the word count N (N=0 to 65535)
15. Byte 4 is the ECC, calculated from byte 1,2,3
16. Byte 5 to byte 5+N-1 are the N-byte read data
17. Byte 5+N+1, byte 5+N are the checksum, calculated on byte 5 to byte 5+N-1. If
18. checksum is not calculated by peripheral, this field is 0x0000.

6、Video 模式的 3 种数据格式



- * Non-Burst Mode with Sync Pulses
- * Non-Burst Mode with Sync Events
- * Burst Mode

* 调试记录

LCD半边闪屏问题，原厂给的信息：分析了系统板送出的 video mode timing，資訊摘要如下

HSCLK: 160MHz
Per lane bit-rate: 320Mbps (UI=3.125ns)
HS SoT HS-prepare + HS-zero 約 155ns

由上述的 timing 懷疑與現象是因為 IC HS data settle timing 搭配不當所導致

看来是我们输出的mipi信号 HS-prepare + HS-zero 比 LCD 默认设置短引起的。还有随机整屏闪动的问题通过调节 VFP 和 VBP 的值调到了理想状态。另外 LCD 的 VCC 在使用 mos 管控制后休眠后会有 2.0V 的悬浮电压，通过 RC 电路将电压放掉，将 C78 换成了 10K 电阻。

LCD电路上有几个比较重要的电压：AVDD、VCC、VGH、VGL、HAVDD、VCOM（由AVDD通过电阻分压得到）

* 唤醒慢的问题

在最初调试的几款 LCD 里面初始化 cmd 都比较少，后来在调试一款 IPS 屏的时候发现唤醒需要 3 秒左右，这款 LCD 初始化 cmd 有100多条，之前在调试一款 LCD 的时候每条 cmd 发送之后需要 delay 10ms 再发下一条 cmd，所以在这款 LCD 这里不能有 delay，并且经过调试在确保发送成功的情况下将 LP 的传输速度提高了 3 倍（这里需要读取每条 cmd 的返回值 0x84 确认命令是否发送成功），优化后唤醒时间不到 1 秒。

* LCD 参数理解更正

才发现之前一直对 LCD 的几个参数 HFP、HBP、VFP、VBP 理解有错误，正确的应该是以同步信号（HSYNC、VSYNC）为基准，在同步信号之前的称为 Front，在同步信号之后的称为 Back，而不是之前理解的以有效像素为基准。

* LCD 显示呈锯齿状问题

这两天（12.11）还调试了一款 540 x 960 分辨率的 mipi LCD，在开始的时候一直点不亮，和供应商确认了好久无意间才发现是他们给的初始化代码是错的，使用正确的初始化代码就能点亮了，不过显示出来的图像却是呈锯齿状的，即没有对齐。之前在别的平台也遇到过类似问题，也就是分辨率不是 16 的整数倍，LCD controller 在取数据的时候会对不齐。边研究 Datasheet 边和 ASIC 同事讨论，后来确定了一个方案：即在 DSI、LCD 寄存器里面设置分辨率为 540 x 960 以让 LCD 正确识别信号，但 framebuffer 需要设置为 544 x 960 以对齐，并且设置 Source pitch 寄存器为 544，这样显示就正常了，相当于 framebuffer 里每一行的最后 4 个 pixel 会被 LCD controller 丢掉。

今天（12.12）在和 ASIC 同事的讨论下更正了之前的理解：LCD controller 在计算取数据的时候，地址是根据（x，y）坐标来算的，差不多是address = y * pitch + x + base，pitch 就是一行 pixel 在内存里的大小，这个至少是要对齐到 8byte，因为 bus 宽度是 8byte，如 Data sheet 中的描述 “Source pitch for RGB channel, QWORD aligned if linear mode”。之前计算 pitch 值的公式为：xres / 8 * bits_per_pixel / 8，如果 xres = 540，bits_per_pixel = 32，计算的结果因为取整的原因为 0x10c，实际上正确的值应该是 0x10e，所以需要将公式改为：xres * (bits_per_pixel / 8) / 8，即在每个像素占 4byte 的情况下只要 xres 为偶数就可以满足对齐的要求，而不用改为 544。