AN13573

i.MX 8/RT MIPI DSI/CSI-2

第0版—2022年3月21日 应用笔记

目录

1 简介

本应用笔记提供了不同的i.MX 8和i.MX RT处理器上的MIPI DSI和CSI-2接口的详细信息，涵盖了处理器之间的实现差异，并提供了设计CSI-2和/或DSI系统以及调试操作的信息。

有关不同处理器的详细物理设计参考资料，请参阅《硬件开发人员指南》（Hardware Developers Guides）。

本文档也不包括详细的软件指南。

1.1 参考资料

- i.MX 8M参考手册
- i.MX 8M数据手册
- i.MX RT参考手册
- i.MX RT数据手册
- D-PHY 1.2版MIPI规范
- DSI 1.1版MIPI规范
- CSI-2 1.3版MIPI规范

1.2 综述

移动产业处理器接口（MIPI）联盟由Arm、Nokia ST和Texas Instruments于2003年成立，英特尔、摩托罗拉、三星和飞利浦随后加入。至今有300多名成员，涵盖IP供应商和芯片制造商，以及产品开发人员等。该组织专注于设计和推广硬件及软件接口，以简化芯片内置组件的集成。

本文档涵盖了以下三种规范：

- 摄像头串行接口2（CSI-2）规范，定义了外围设备（摄像头）和主处理器之间的标准数据传输和控制接口，着重于通过摄像头控制接口（CCI）传输图像数据（包括静止图像和视频）的协议以及基础图像传感器的控制。CSI-2规范不涵盖任何高级成像系统功能，例如图像曝光、聚焦、处理或成像系统中使用的任何其他常见功能。
- 显示串行接口（DSI）规范，定义了主处理器和外围设备（显示器）之间的数据传输协议以及接口。

  CSI-2和DSI规范虽然没有规定物理层，但它们是整个系统中的底层。

- D-PHY规范，定义了传输介质（电导体）、输入/输出电路以及从串行位流中捕获1和0的时钟机制。

图1是一个典型的系统设计示例，包括摄像头和触屏显示。摄像头和触屏显示都连接到一个应用处理器，这个应用处理器使用独立的四通道MIPI D-PHY接口进行数据传输并使用I2C接口进行控制。

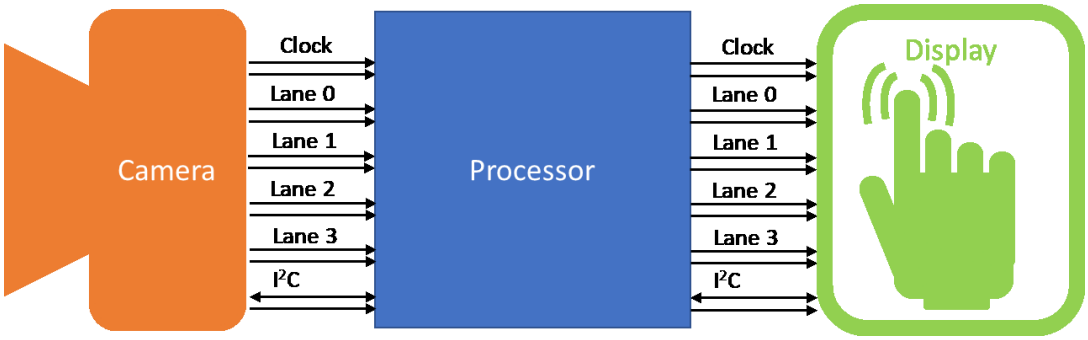


图1. 使用MIPI CSI-2和DSI接口的示例系统

注意

MIPI DSI、CSI-2和D-PHY规范适用于芯片间的接口和设计，其中摄像头和显示器是一个完全一体化产品的一部分，像是手机或平板电脑等。与某些热插拔或即插即用高速接口（如HDMI或USB）不同，MIPI DSI/CSI-2接口在最终设计中应该是固定的永久性连接。

2 D-PHY

D-PHY是CSI-2和DSI协议使用的源同步物理层，面向物理的及电气的通信。D-PHY中的D是罗马数字500的意思。D-PHY规范的1.0版是假设单个通道的通信速率为500Mbps而编写的，因此D-PHY这个名称诞生了！在最新版本的规范中，每个通道的实际最大比特率高达9Gbps。由于遵循规范版本和设计的实现，i.MX 8/RT部件的每通道最大为1.5 Gbps（有些部件甚至更低）。

2.1 链路基础知识

MIPI规范在讨论基于CSI-2或DSI组装系统所需的部件时使用了一组一致的术语。

- 线路：主处理器的单个引脚与外围设备的某个引脚之间的互连、导线或引线。
- 通道：需要两条线路来支持差分信号。D-PHY使用差分对进行高速数据和时钟传输。这种差分对被称为通道互连（Lane Interconnect），但在技术文档中大多简称为通道。
- 链路：包含一个时钟通道和至少一个数据通道的两个设备之间的连接。一个链路至少要包括两个PHY和两个通道的互连。

许多数据手册简化了链路信息和/或完全省略了链路这个术语。例如，在i.MX RT500数据手册中，框图显示了MIPI-DSI（双通道）。这意味着该部件包含一个时钟通道和两个数据通道，对应于在MIPI DSI接口中使用的六个引脚。

2.2 操作模式

共有四种操作模式：控制模式（Control）、高速传输模式（High-Speed）、脱离模式（Escape）和超低功耗状态（ULPS）模式。数据通道支持所有四种模式，但时钟通道仅支持控制模式、高速传输模式和超低功耗状态模式。正常运行时，数据通道处于控制模式或高速传输模式。高速数据传输模式在数据突发时采用，从控制模式的停止状态（LP-11）开始并在控制模式的停止状态（LP-11）结束。通道仅在数据突发时处于高速模式。表1描述了时钟和数据通道可以进入的六种物理状态。

表1. 通道状态描述

状态代码	线路电平		高速	低功耗	
	Dp线路	Dn线路	突发模式	控制模式	脱离模式
HS-0	HS低	HS高	差分0	无[1]	无[1]
HS-1	HS高	HS低	差分1	无[1]	无[1]
LP-00	LP低	LP低	无	桥接	空
LP-01	LP低	LP高	无	HS请求	Mark-0
LP-10	LP高	LP低	无	LP请求	Mark-1
LP-11	LP高	LP高	无	停止	无[2]

1. 在高速传输期间，低功耗接收器会检查线路上的LP-00状态码。
2. 如果在脱离模式下出现LP-11，通道将返回到停止状态（控制模式LP-11）。

2.3 I/O信号

D-PHY IO引脚有两种不同的电气操作模式：单端模式和差分模式。

- 单端模式称为低功耗（LP）模式，用于脱离模式和控制模式，在主处理器和外围设备之间进行系统配置/控制。在LP模式下，IO电路使用传统的CMOS输出缓冲器（带斜率控制），其在接地和VDD_IO之间切换。在LP模式下，组成通道的两个IO电路可以独立切换。它们提供四种不同的通道状态：LP-00、LP-01、LP-10和LP-11。
- 差分模式被称为高速（HS）模式，仅用于数据传输。差分模式只有HS-0和HS-1两种状态。在HS模式下，组成通道的两个IO电路输出极性始终相反，不能同时为逻辑高或逻辑低。图2说明了HS和LP模式之间信号电平的差异。

正常运行时，数据通道在低功耗控制模式和高速数据传输模式之间不断切换。

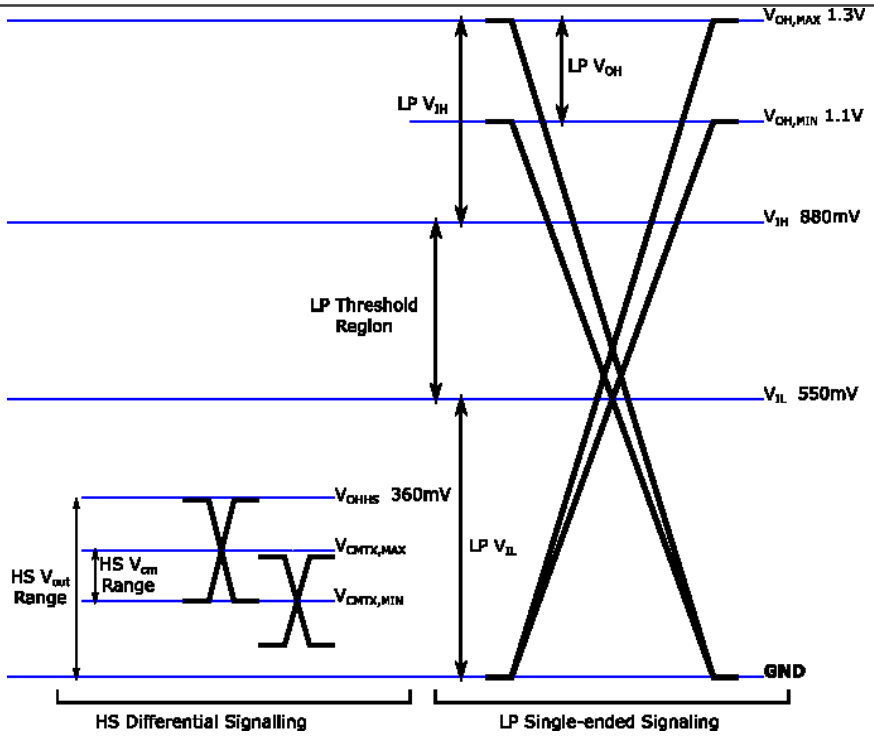


图2. D-PHY信号电平

表2列出了在LP和HS模式下都要使用的最关键IO信号的电压值。HS模式的工作电压范围可低于550 mV的V_{il}电平。如果链路发送器进入HS模式，而接收器没有进入（不管什么原因），接收器都会迅速标记错误，因为它不希望在LP模式看到长时间的逻辑0。

表2. D-PHY直流电气规范

参数	描述	最小值	类型值	最大值	单位
V_{OH}	输出高水平	1.1	1.2	1.3	V
V_{OL}	输出低电平	-50		50	mV
V_{IH}	输入高电平阈值	880			mV
V_{IL}	输入低电平阈值			550	mV
$V_{IL-ULPS}$	超低功耗状态模式下的输入低电平阈值			300	mV
V_{CMTX}	高速传输模式静态共模电压	150	200	250	mV
$\|V_{OD}\|$[1]	高速传输模式差分电压	140	200	270	mV
V_{OHHS}	高速传输模式输出高压			360	mV

1. 差分输出电压V_{OD}是D_P和D_N引脚上的V_{DP}和V_{DN}的电压差。

2.4 DDR时钟

HS模式下的数据传输使用双数据速率（DDR）传输方案。每当时钟通道信号改变状态时，接收器对数据进行采样。在单数据速率（SDR）同步数据总线架构（例如SDRAM）中，数据要么在上升沿要么在下降沿采样，但不会同时在上升沿或下降沿采样，这意味着数据速率等于时钟速度。各种低功耗模式使用单数据速率时钟结构。

DDR数据总线架构可以在每个时钟边缘传输数据，而不是仅在单个时钟边缘传输数据。这意味着对于给定的时钟频率（Fr），每通道的数据速率为2×Fr。例如，如果时钟频率为100MHz，则数据速率为2×100 MHz=200 MHz。图3显示了SDR和DDR两种时钟方案的数据和时钟之间的关系。

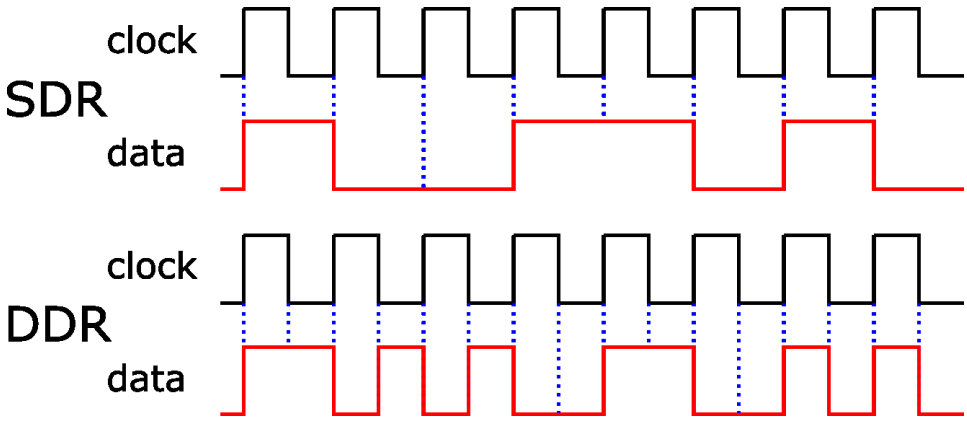


图3. SDR与DDR对比

由于HS模式使用差分信号，时序值是根据实际观察到的时钟差分信号的交叉来测量的。图4显示了数据和时钟通道的时序对齐。

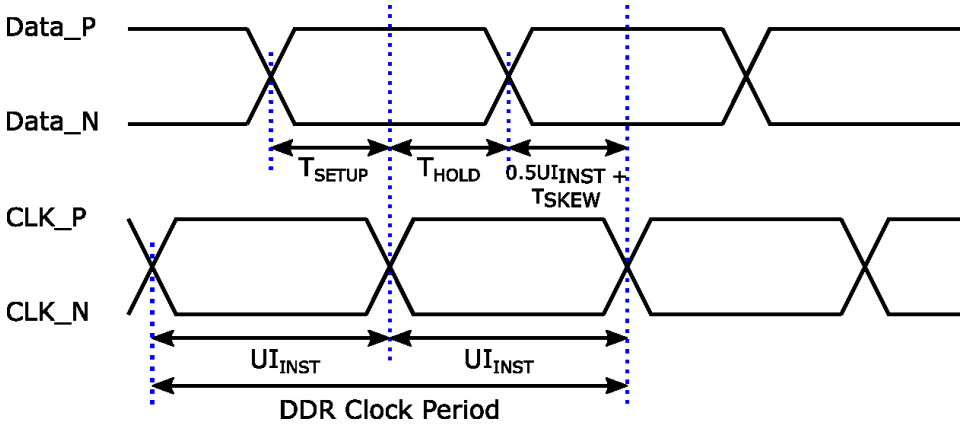


图4. D-PHY DDR时钟的定义

单位间隔（UI）等于时钟通道上HS状态的持续时间。DDR时钟周期等于2 × UI。

2.5 模式转换

为了有效利用D-PHY接口使用的少量引脚，没有传统的总线握手信号（例如数据使能、线路选择）。因此，各种LP和HS模式之间的转换是通过一系列握手顺序来处理的。

2.5.1 HS传输的开始

从停止状态（LP-11）转换到高速数据传输模式需要两个步骤。

1. 时钟通道进入高速时钟模式。
2. 数据通道进入高速模式。

2.5.1.1 时钟通道HS转换

表3描述了当时钟通道从低功耗状态转换到高速模式时的事件顺序。

表3. 启动高速时钟转换的过程

TX侧	RX侧
驱动停止状态（LP-11）	观察停止状态
驱动HS Req状态，维持TLPX时长（LP-01）	观察线路从LP-11到LP-01的转换
驱动桥接状态，维持T_{CLK}-PREPARE时长（LP-00）	观察线路从LP-01到LP-00的转换。在TCLK-TERM-EN时长之后开启线路端接
启用高速驱动并同时禁用低功耗驱动。	开启HS-RX并等待定时器$T_{CLK-SETTLE}$到期以忽略转换的影响。
驱动HS-0状态，维持$T_{CLK-ZERO}$时长	接收HS信号
在任何数据通道启动之前，驱动高速时钟信号，维持TCLK-PRE的时长	接收高速时钟信号

2.5.1.2 数据通道HS转换

表4描述了当数据通道从停止状态转换到高速模式时的“传输开始”（SoT）过程。

表4. 传输开始顺序

TX侧	RX侧
驱动停止状态（LP-11）	观察停止状态
驱动HS-Req状态, 维持TLPX时长（LP-01）	观察线路从LP-11到LP-01的转换
驱动桥接状态，维持THS-PREPARE时长（LP-00）	观察线路从LP-01到LP-00的转换。在TCLK-TERM-EN时长后开启线路端接
启用高速驱动并同时禁用低功耗驱动。	
驱动HS-0状态，维持THS-ZERO时长	启用HS-RX并等待定时器THS-SETTLE到期以忽略转换的影响。
	开始寻找前导序列
在上升时钟边缘插入HS同步序列‘00011101'	
	在识别到前导序列011101时同步

表格在下一页继续...

表4. 传输开始顺序（续）

TX侧	RX侧
继续传输高速数据载荷	
	接收数据载荷

2.5.1.3 时钟与数据SoT的关系

图5显示了时钟通道和数据通道的发送器/接收器对之间的时序关系。图5只显示了一个数据通道。多个数据通道遵循相同的顺序并同时开始。

为确保正确地进入HS模式，正确地配置发送器和接收器的时序参数是至关重要的。

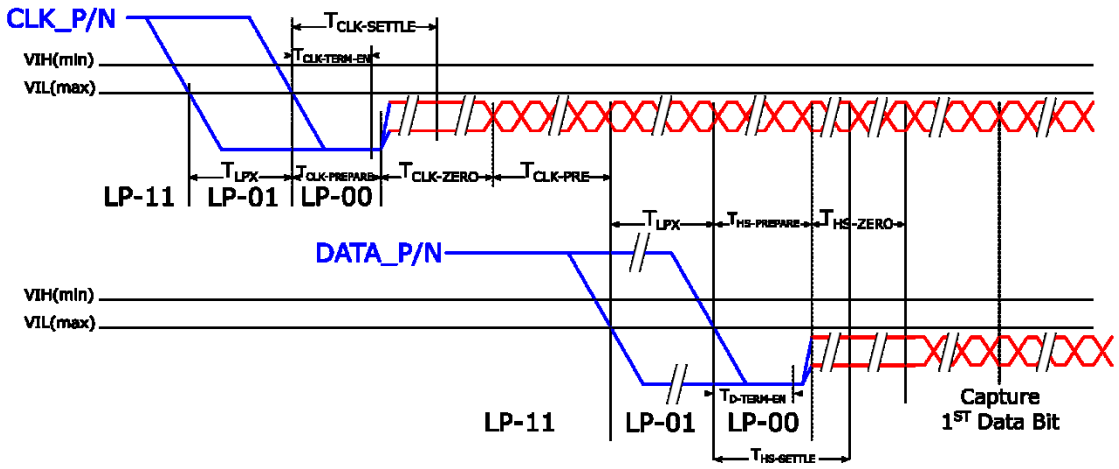


图5. 时钟和数据通道进入HS传输

图5所示的一些时序可以用本文档所涵盖的SOC的DSI/CSI-2/D-PHY配置寄存器进行调整。表5列出了允许调整的D-PHY规范、参数和相关的IP模块。

表5. D-PHY SoT时序参数

参数	描述	最小值	最大值	单位	IP
$T_{CLK-PRE}$	在任何相关数据通道开始从LP模式到HS模式的转换之前，发送器驱动HS时钟的时间。	8		UI[1]	DSI
$T_{CLK-PREPARE}$	紧接在HS-0线路状态开始HS传输之前，发送器驱动时钟通道LP-00线路状态的时间。	38	95	ns	DSI
$T_{CLK-SETTLE}$	从TCLK-PREPARE的开始起，HS接收器应忽略任何时钟通道HS转换的时间间隔。	95	300	ns	N/A[2]
$T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$	$T_{CLK-PREPARE}$ +发送器在开启时钟之前驱动HS-0状态的时间。	300		ns	N/A[2]

表格在下一页继续...

表5. D-PHY SoT时序参数（续）

参数	描述	最小值	最大值	单位	IP
$T_{CLK-TERM-EN}$	时钟通道接收器开启HS线路端接的时间，从Dn穿过V_{IL}最大值的时间点开始。	Dn到达VTERM-EN的时间		ns	无[2]
$T_{CLK-ZERO}$	发送器在开启时钟之前驱动HS-0状态的时间。				DSI
$T_{D-TERM-EN}$	数据通道接收器启用HS线路端接的时间，从D_n穿过VIL最大值的时间点开始。	Dn到达VTERM-EN的时间	35 ns + 4 × UI		无[2]
$T_{HS-PREPARE}$	紧接在HS-0线路状态开始HS传输之前，发送器驱动数据通道LP-00线路状态的时间。	40 ns + 4 × UI	85 ns + 6 × UI	ns	DSI
$T_{HS-PREPARE} + T_{HS-ZERO}$	THS-PREPARE+发送器在发送同步序列之前驱动HS-0状态的时间。	145 ns + 10 × UI		ns	
$T_{HS-SETTLE}$	HS接收器应忽略所有数据通道HS转换的时间间隔，从THS-PREPARE的开始算起。 HS接收器应在最小值之前忽略所有数据通道的转变，但HS接收器应该响应所有在最大值之后的数据通道的转变。	85 ns + 6 × UI	145 ns + 10 × UI	ns	CSI
$T_{HS-ZERO}$	发送器在发送同步序列之前驱动HS-0状态的时间。				DSI
T_{LPX}	所有在低功耗状态期间传输的长度	50		ns	N/A[2]

1. MIPI位时钟周期/2。
2. 这些参数必须符合D-PHY规范，但不能直接编程。

2.5.2 HS传输的结束

高速数据突发结束时，数据通道退出高速传输模式，并经过“传输结束”（EoT）过程进入停止状态。表6描述了这个事件顺序。

表6. 传输结束顺序

TX侧	RX侧
完成数据载荷的传输	接收数据载荷
在最后一个数据载荷位后立即翻转差分状态，并保持该状态至THS-TRAIL时长	
关闭HS-TX，开启LP-TX，并驱动停止状态（LP-11）至THS-EXIT时长	检测线路离开LP-00状态并进入停止状态（LP-11），并关闭端接
	忽略上一段THS-SKIP的数据为来隐藏转换的影响
	检测有效数据的最后一个转变，确定最后一个有效数据字节，并跳过拖尾序列

2.5.2.1 时钟与数据的EoT的关系

图6显示了时钟通道和数据通道的发送器/接收器对之间的EoT时序关系。图6仅显示了一条数据通道。多个数据通道遵循相同的顺序并同时结束。为确保正确地退出HS模式，正确地配置发送器和接收器的时序参数是至关重要的。

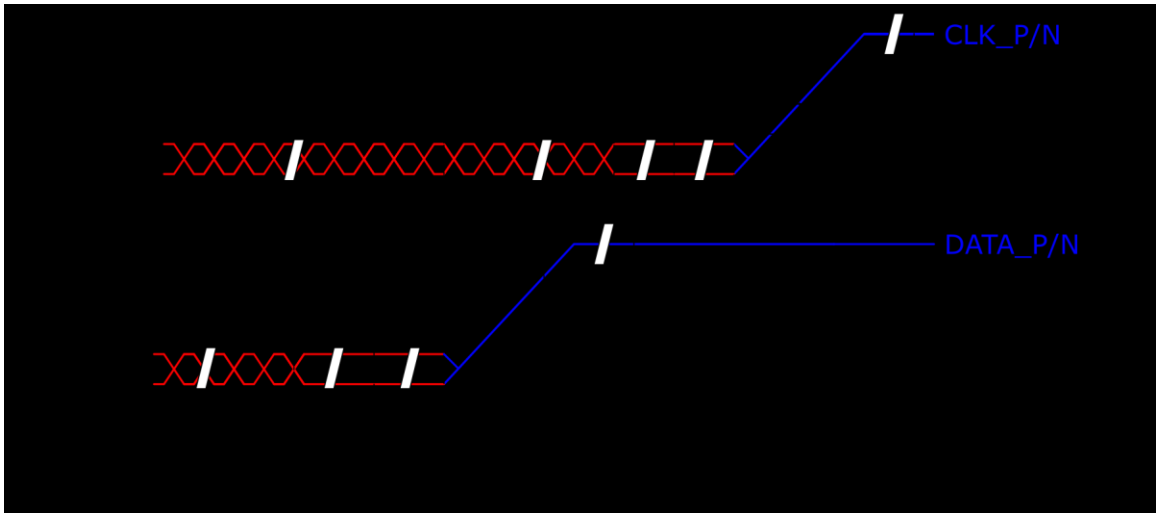


图6. 数据和时钟通道退出HS传输

图6所示的一些时序可以用本文档所涵盖的SOC的DSI/CSI-2/D-PHY配置寄存器进行调整。表7列出了允许调整的D-PHY规格、参数和相关IP模块。

表7. D-PHY的EoT时序参数

参数	描述	最小值	最大值	单位	IP
$T_{CLK-POST}$	最后一个相关数据通道转变为LP模式后，发送器继续发送HS时钟的时间。间隔定义为从$T_{HS-TRAIL}$结束到$T_{CLK-TRAIL}$开始的时间段。	60 ns + 52 × UI		ns	DSI
$T_{CLK-MISS}$	接收器检测到时钟转换缺失并禁用时钟通道HS-RX的超时时间。		60	ns	N/A[1]
$T_{CLK-TERM-EN}$	时钟通道接收器开启HS线路端接的时间，从Dn穿过V_{IL}最大值的时间点开始。		38	ns	N/A[1]
$T_{CLK-TRAIL}$	在HS传输突发的最后一个载荷的时钟位之后，发送器驱动HS-0状态的时间。	60		ns	DSI
T_{EOT}	从$T_{HS-TRAIL}$或$T_{CLK-TRAIL}$开始，到HS突发随后的LP-11状态开始之间的传输时间间隔。		105 ns + n × 12 × UI	ns[2]	DSI
$T_{HS-EXIT}$	HS突发随后的发送器驱动LP-11的时间。	100		ns	DSI
$T_{D-TERM-EN}$	数据通道接收器开启HS线路端接的时间，从Dn穿过V_{IL}最大值的时间点开始。		35 ns + 4 × UI	ns	N/A[1]

表格在下一页继续...

表7. D-PHY的EoT时序参数（续）

参数	描述	最小值	最大值	单位	IP
$T_{HS-SKIP}$	HS-RX应忽略数据通道上的所有转变的时间间隔，随后为HS突发。时间间隔的结束点定义为HS突发随后的LP-11状态的开始。	40	55 + 4 × UI	ns	CSI
$T_{HS-TRAIL}$	在HS传输突发的最后一个数据载荷位之后，发送器驱动翻转后的差分状态的时间。	max (n × 8 × UI, 60 ns + n × 4 × UI)		ns[2]	DSI

1. 这些参数必须符合D-PHY规范，但不能直接编程。
2. 其中n=1表示正向HS模式，n=4表示反向HS模式。

2.5.3 连续时钟与非连续时钟

CSI-2和DSI规范均支持连续时钟模式。对于连续时钟的工作方式，其时钟通道会保持高速模式，并在HS数据包传输之间生成有效的时钟信号。对于非连续时钟的工作方式，其时钟通道在HS数据包传输之间进入LP-11状态。连续时钟模式允许更高的数据速率，因为取消了退出和重新进入时钟通道的HS模式的时序开销。

2.5.3.1 非连续时钟模式转换过程

一旦数据传输完成，时钟通道可以选择切换回低功耗状态（LP-11）。表8描述了时钟通道从高速模式转换到低功耗状态的事件顺序。

表8. 将时钟通道切换到低功耗模式的过程

主机侧	从机侧
驱动高速时钟信号（翻转HS-0/HS-1）	接收高速时钟信号（翻转HS-0/HS-1）
最后一个数据通道进入低功耗模式	
继续驱动高速时钟信号至TCLK-POST时长并以HS-0状态结束	
驱动HS-0状态至TCLK-TRAIL时长	在TCLK-MISS时间内检测时钟转换的缺失，关闭HS-RX然后等待转换到停止状态
关闭HS-TX，开启LP-TX，并驱动停止状态（LP-11）至THS-EXIT时长	
	检测线路转换到LP-11，关闭HS端接，并进入停止状态

2.5.4 超低功耗状态

从D-PHY的角度来看，超低功耗状态将时钟和数据通道都置于LP-00状态，并向外围设备发送信号以进入ULPS。实际的ULPS实现取决于不同的设备制造商。D-PHY规范没有描述主机或外围设备计算内核应该做些什么，或者它们在此期间如何操作。

2.5.4.1 时钟通道进入ULPS

表9描述了时钟通道进入ULPS的顺序。

表9. 时钟通道进入和退出ULPS的顺序。

TX	RX	备注
在TX停止状态（LP-11）下启动	RX停止（LP-11）	
通过发送LP-10状态至T_{LPXs}时长以请求ULPS		
驱动TX-ULPS状态（LP-00）至所需时长。		
通过驱动LP-10状态至T_{WAKEUP}时长来退出ULPS		

2.5.4.2 数据通道进入ULPS

数据通道进入ULPS的顺序稍微复杂一些，因为数据通道可以在具有链路回转过程的脱离模式期间支持双向通信。

注意

MIPI DSI规范将其称为总线回转过程（BTA）。CSI-2规范没有BTA过程，而是依赖I^2C链路进行双向通信。

TX数据通道发送握手序列作为ULPS模式的信号。表10描述了事件的顺序。

表10. 数据通道进入和退出ULPS的顺序

TX	RX	备注
自TX停止状态（LP-11）启动	RX处于停止状态（LP-11）	
通过发送LP-10状态至TLPX时长以请求LP		
发送LP-00至TLPX时长		
发送LP-01至TLPX时长		
发送LP-00至TLPX时长		
发送8位ULPS进入命令序列（00011110）		如果发送器在任何时候检测到LP-11状态，顺序将中止，通道恢复到停止状态（LP-11）
发送LP-00至所需的时长。		
通过驱动LP-10状态至T_{WAKEUP}时长来退出ULPS		

图7显示了数据通道进入ULPS时的逻辑状态。

注意

进入ULPS的命令使用Spaced-One-Hot编码。

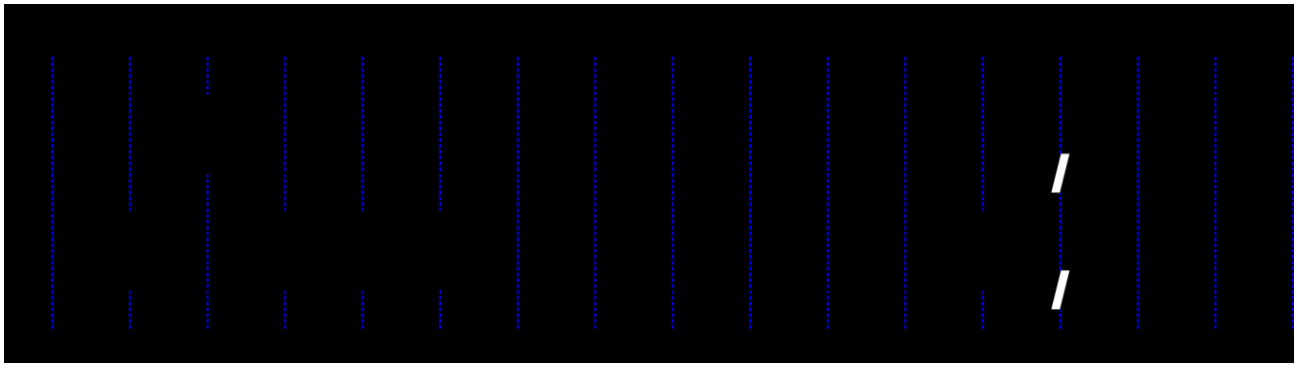


图7. 数据通道进入ULPS的顺序

Spaced-One-Hot 编码允许在接收器进行时钟恢复，因此不再需要时钟通道。工作原理是每个Mark（标记）状态与一个Space（空格）状态交织在一起。因此，每个符号由两部分组成：One-Hot阶段（标记0或标记1）和空格阶段。TX应发送标记0，后跟一个空格以发送数据位零，或应发送标记1，后跟一个空格以发送数据位1。后面没有空格的标记不代表数据位。退出带停止状态的脱离模式之前的最后阶段应为标记1的状态，该状态不是通信数据位的一部分，因为它后面没有空格状态。

2.6 故障检测

如果发送器和接收器的可配置时序参数都没有正确设置，则在SoT和EoT过程中可能会发生错误。D-PHY可以检测到某些错误，触发IRQ，并产生寄存器标志以帮助调试这些问题。

2.6.1 SoT错误

最常见的错误发生在传输顺序开始的期间。出现某些错误时，CSI-2和DSI IP模块会产生IRQ。

如果高速SoT先导序列已损坏，但仍然实现了正确的同步，则会发生`ErrSotHS`。该错误信号在`RxByteClkHS`的一个周期有效。这个错误被认为是先导序列中的软错误，导致数据载荷的可信度降低。

如果高速SoT先导序列已损坏，而且无法期望进行正确同步，则会发生`ErrSotSync_HS`。该错误信号对于HS接收置为高。

如果出现上述任一错误，最好重新检查SoT时序参数设置。

- 在发送器侧（SOC DSI或摄像头），可以通过调整$T_{CLK-PRE}$、$T_{CLK-PROPARE}$、$T_{HS-PREPARE}$、$T_{CL-ZERO}$和$T_{HS-ZERO}$参数来缩短进入HS模式的时间。
- 在接收器侧（SOC CSI-2或显示器），可以通过调整$T_{HS-SETTLE}$参数来增加寻找SoT头之前的等待时间。

3 DSI

本节介绍DSI接口以及显示设备如何连接到主处理器。阅读本节后，工程师可以评估各种显示器，并确定各种显示器对于本文档所述的所有i.MX处理器的适用性和兼容性。系统工程师可以确定整套显示系统的硬件。本节不介绍在软件中实现整套显示系统的细节，如图像处理（GPU/PXP）、颜色校正、驱动程序等。

3.1 综述

DSI指定主处理器和外围设备（如显示模块）之间的接口，它建立在现有MIPI联盟规范基础上，采用DPI-2、DBI-2和DCS标准指定的像素格式和命令集。DSI模块将所有像素数据、命令和事件序列化。在传统或旧有的接口中，数据通常利用附加控制信号通过并行数据总线发送到外设，或从外设接收。支持每像素24位（bpp）色深的旧有显示接口需要24个引脚的数据总线、vsync、hsync、enable和像素时钟线，需要28个单独的信号。DSI规范规定，根据D-PHY中使用的数据通道的数量，可以使用4-10个引脚来发送相同的数据。

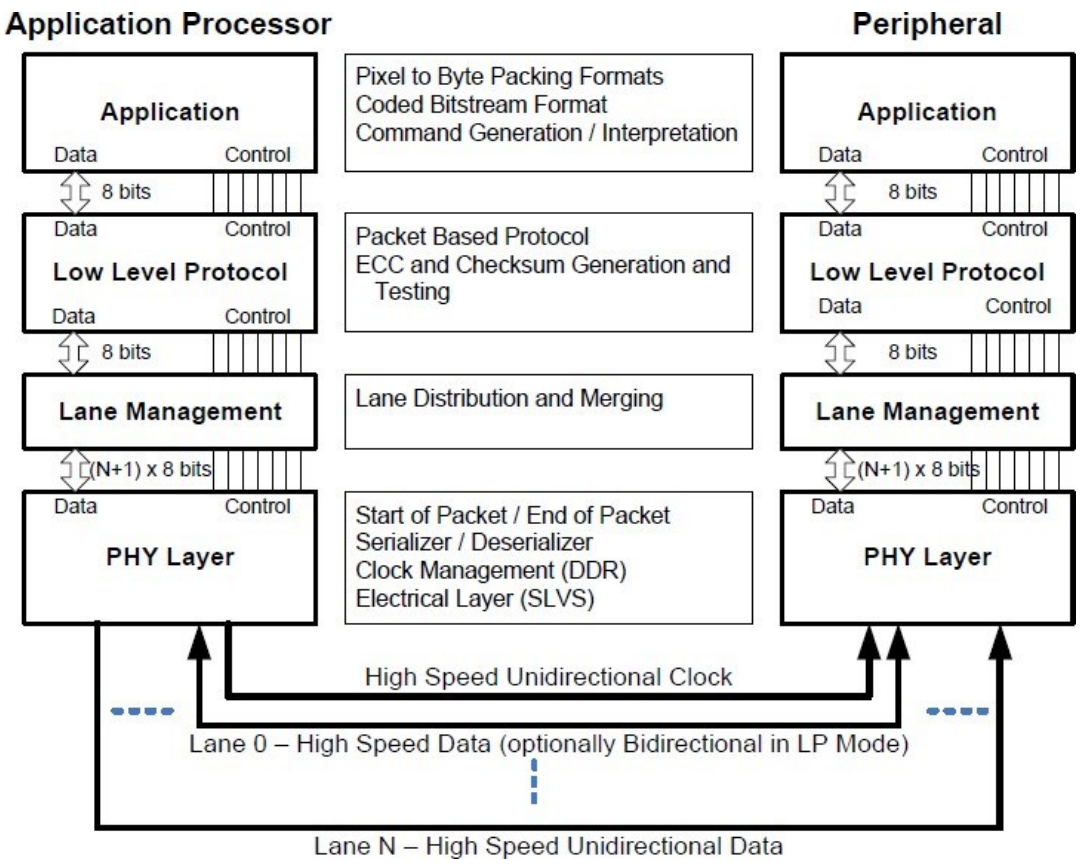


图8. DSI层

3.2 数字显示基础知识

数字显示基础知识有助于对数字显示器如何工作有个基本的了解，以了解数据传输的格式和时序。所有现代计算机/信息显示器都是光栅显示器，意味着显示的图像是在显示器中从左到右，从上到下，逐个像素、一次写入一行来创建的。显示的图像或帧由矩形像素阵列组成。

视频流是由带固定时序间隔的一系列显示帧组成。这种时序间隔可以使用每秒帧数（FPS）或Hz来描述。传统的电影院投影机使用24 FPS或24 Hz作为时序间隔。现代数字显示器（例如智能手机）使用60Hz或更高的帧间隔频率。

这一流程源自老式的CRT显示器，电子束不断穿过屏幕的背面来点亮图像。光束会从左向右扫描以生成图像的一行，然后关闭，以水平方向返回到左侧，在下一行开始时再次打开。光束从上到下继续这种模式，直到执行一个单次垂直折返。图9显示了电子束的路径。

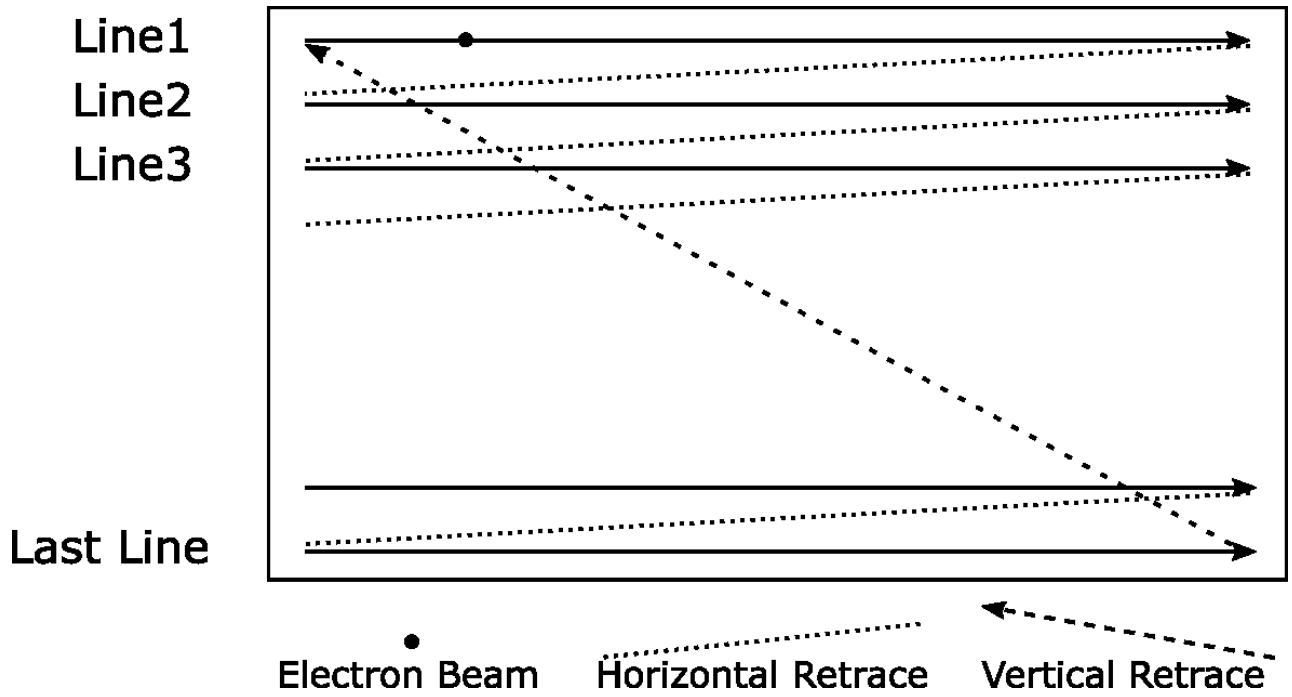


图9. CRT显示器光栅扫描

这个水平回扫不是瞬间的，需要一定的时间将光束从右转到左，再在下一行开始前稳定光束。整个过程称为水平消隐间隔，由三个不同的部分组成。

- **水平前廊**（HFP）：当光束到达显示行的末端时，关闭和回转光束的需要的时间间隔。
- **水平同步脉冲**（H_{sync}）：使光束越过屏幕返回所需的时间间隔。
- **水平后廊**（HBP）：打开光束以照亮下一行之前稳定光束所需的时间间隔。

垂直回扫需要一定的时间，称为垂直消隐间隔，由以下三部分组成。

- **垂直前廊**（VFP）：一旦显示了图像的最后一行，回转光束所需的时间间隔。
- **垂直同步脉冲**（V_{sync}）：将光束从屏幕底部转向顶部所需的时间间隔。
- **垂直后廊**（VBP）：在开始第一行之前，稳定垂直光束所需的时间间隔。

现代数字显示器不再需要时间来转动电子束，但仍然利用这些消隐时间来进行图像处理和发送附加数据（HDMI中的音频或CSI-2中的嵌入式图像数据）等工作。

3.2.1 像素、时钟和同步

电子束是一个有用的思路模型，便于理解数字显示数据是如何通过SOC和DSI链路传输的。像素数据从左到右、从上到下依次串行传输。像素数据中存在间隙用于各种消隐间隔。这些消隐间隔定义了图像宽度和相对于显示器的位置。

注意

（X，Y）像素坐标数据不存储或传输到任何地方。

图10显示了所有同步信号与有效显示区域的关系。

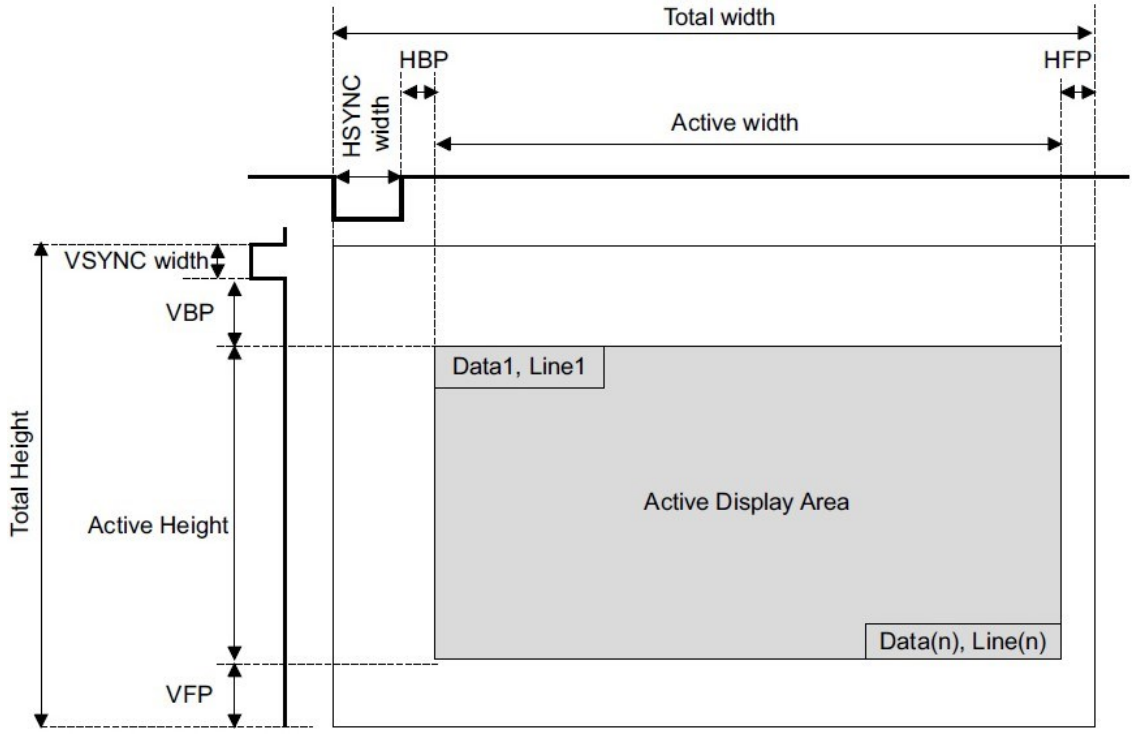


图10. 数字化显示的同步关系

图像数据的单个水平行以HSYNC脉冲作为信号标志。该脉冲之后是HBP，然后是有效像素数据，最后是HFP。HSYNC脉冲、HBP和HFP均使用像素作为测量单位。

VSYNC脉冲标志着新帧的开始。该脉冲之后是VBP，然后是有效图像行，最后是VFP。对下一个图像帧重复此流程。VSYNC、VBP和VFP均使用水平图像行作为测量单位。

时钟必须保持像素数据和同步信号的顺序。该时钟被称为像素时钟，每个时钟周期传输一个像素。它是大多数图像显示传输系统的主时钟。

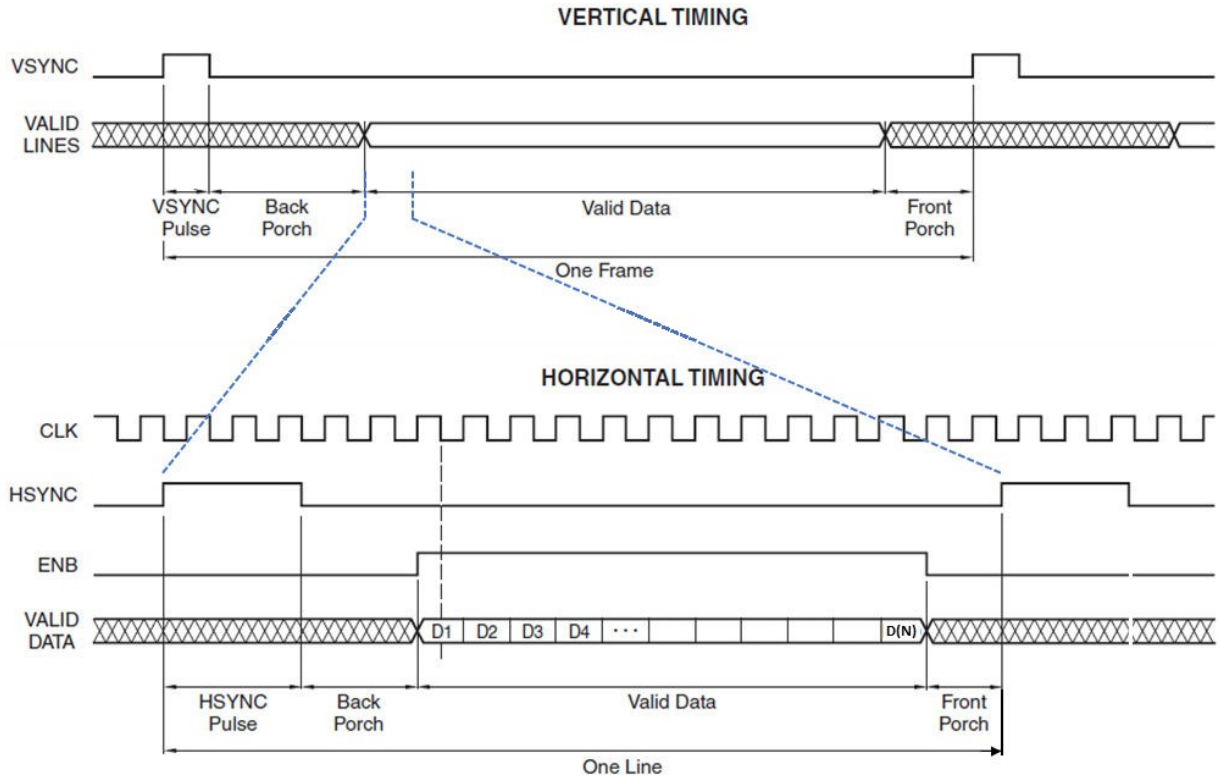


图11. 垂直和水平时序的关系

图11显示了HSYNC、VSYNC、像素时钟和数据之间的时序关系。每个像素时钟周期传输单个像素所需的数据。

ENB信号用于发出有效像素数据传输的信号。没有ENB信号，各种像素管道可能会将消隐数据误认为值为0的像素数据。

3.2.1.1 示例——像素时钟与分辨率的关系

SMPTE 274 M标准定义了所谓的全高清电视的分辨率和帧速率。该标准定义了1920个水平有效像素和1080个有效行，总计1920×1080=2073600个有效像素。对于2200像素的总行宽，该标准要求HBP为148像素，HFP为88像素，而HSYNC宽度为44像素。HD标准还规定1125个垂直行的VBP为36行，VFP为4行，而VSYNC宽度为5行。

因此，一整帧高清电视数据实际上包含2200×1125=247.5万像素或像素时钟。

在本例中，帧率为30 FPS所需的像素时钟可以通过将每帧的总像素乘以所需的帧率来计算。像素时钟=2.475Mpix*30fps=74.25 MHz。

3.3 MIPI显示器像素接口（DPI-2）

显示器像素接口（DPI-2）是MIPI联盟的一个标准，面向无显示控制器的显示板或无帧缓冲器的显示模块的并行接口。这些显示模块依赖从主处理器到显示器的稳定像素数据流，以保持没有闪烁或其他视觉伪影的图像。DSI规范将DPI-2操作模式称为视频模式。这个接口通常被用于在主处理器和（不兼容MIPI DSI的）LCD显示器之间传输和同步视频流。许多SOC使用这个接口（或类似的东西）在IP模块之间传输图像数据。

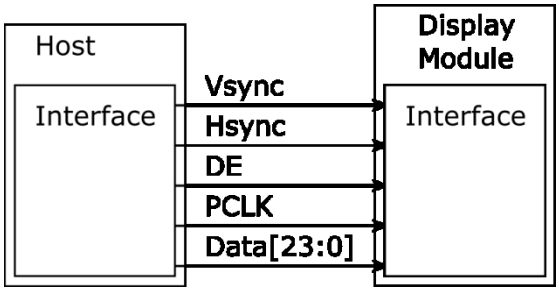


图12. DPI接口信号

图12显示了通用的DPI接口，以及主处理器和显示器之间的关键信号。

- **PCLK**：像素时钟是主系统时钟。像素数据在一个像素周期内传输。显示模块使用PCLK的上升沿来捕获像素数据。该时钟连续运行，是所有其他视频时序信号的根时钟。
- **数据[23:0]**：每个像素值通过并行数据总线从主处理器传输到显示模块。并行数据总线的宽度可以从16到24不等。
- **Hsync**：该水平同步信号标志每一水平像素行的开始，并使用像素时钟的数量来定义。
- **Vsync**：该垂直同步信号标志被显示图像的每帧的开始，并使用显示行的整数倍来定义。
- **DE**：该数据使能信号表示有效像素数据正在传输，有时也被称为线路有效（LV）或HREF信号。

3.4 MIPI显示架构

MIPI DPI规范定义了四种不同的显示模块架构。所有符合MIPI的显示器均遵循这四种架构中的一种，这四种架构有许多共性。显示驱动IC负责将输入的图像数据流转换为显示面板所需的控制信号。图13显示了基本的显示/主机系统架构。

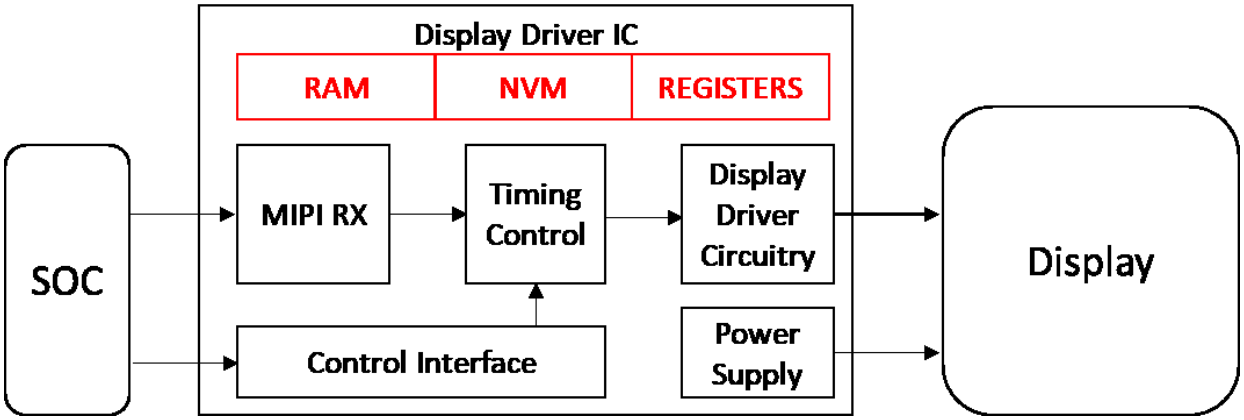


图13. MIPI显示架构

架构类型在内存配置上有所不同。显示类型2、3和4最为常见，并由SOC支持。表11列出了它们的差异。

表11. MIPI显示架构类型的差异

组件	类型1	类型2	类型3	类型4
全帧存储器	X			
部分帧存储器		X		
寄存器	X	X	X	
非易失性存储器	X	X	X	
视频流接口		X	X	X

3.5 DSI视频模式时序

图14显示了有效显示区域与水平和垂直时序的关系。在主机和显示器之间建立DSI链路还需要另外一些时序参数。表12定义了显示器制造商提供的参数。有时这些值不在显示器数据手册中，而是在显示器制造商使用的显示器控制器IC数据手册中。

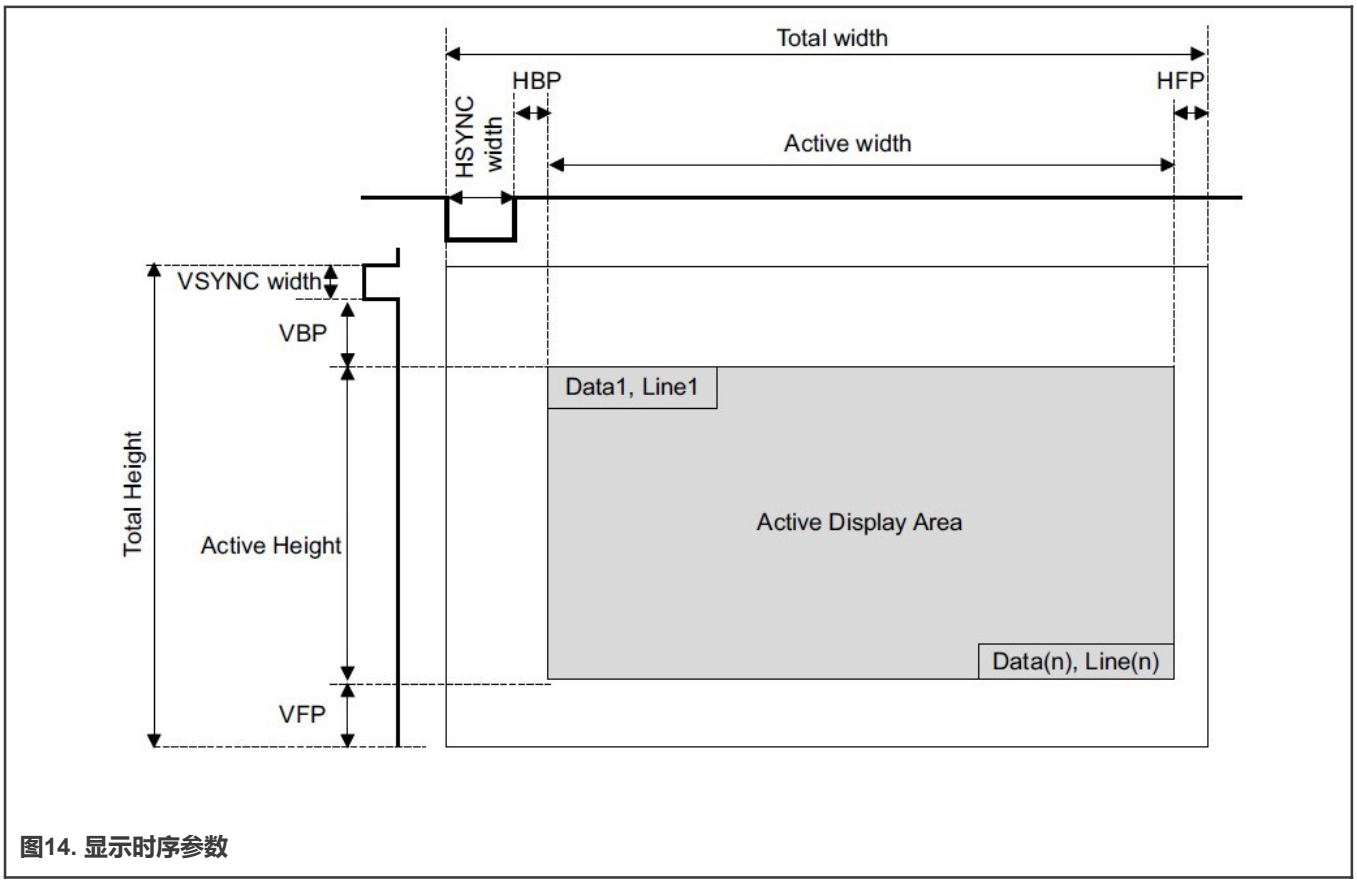


图14. 显示时序参数

表12. 所需的显示时序参数

参数	描述	最小值	最大值	单位	备注
t_L	行时间			μs	
t_{HSA}	水平同步有效			μs	H_{sync}宽度
t_{HBP}	水平后廊			μs	
t_{HACT}	图像数据时间			μs	
HACT	每行有效像素			Pixels	
t_{HFP}	水平前廊			μs	
VSA	垂直同步有效			行	V_{sync}宽度
VBP	垂直后廊			行	
VACT	每帧有效行			行	
VFP	垂直前廊			行	

3.6 视频分组

DSI规范的核心是简单地获取并行图像总线数据流并将其序列化，以便通过高速D-PHY链路传输。DSI接收并行数据、信号事件和命令，并将它们转换为数据包，然后添加数据包协议信息和标头，再将完整的字节发送到PHY，以便通过串行链路进行高速传输。

图像数据和同步信息使用D-PHY HS传输模式以字节大小的数据包发送，有两种数据包类型——短数据包和长数据包：

- 短数据包用于大多数命令模式的命令和相关参数。其他短数据包用于传送事件，如H_{sync}和V_{sync}的边缘。因为它们是短数据包，所以可以向外围设备的逻辑传送准确的时序信息。短数据包的长度均为4个字节，包括ECC。短数据包也是长数据包的标头。
- 长数据包用于传输大块像素或其他数据，它使用一个两字节的计数字段来指定载荷的长度。载荷的长度可以是从0到2^{16}-1字节中的任何值。每个长数据包以一个四字节的数据包标头开始，以一个两字节的数据包页脚结束。如果载荷为空，则长数据包最短可为6字节。最大数据包的长度是65541字节（2^16-1+6）。

图15显示了短数据包和长数据包的特点和区别。

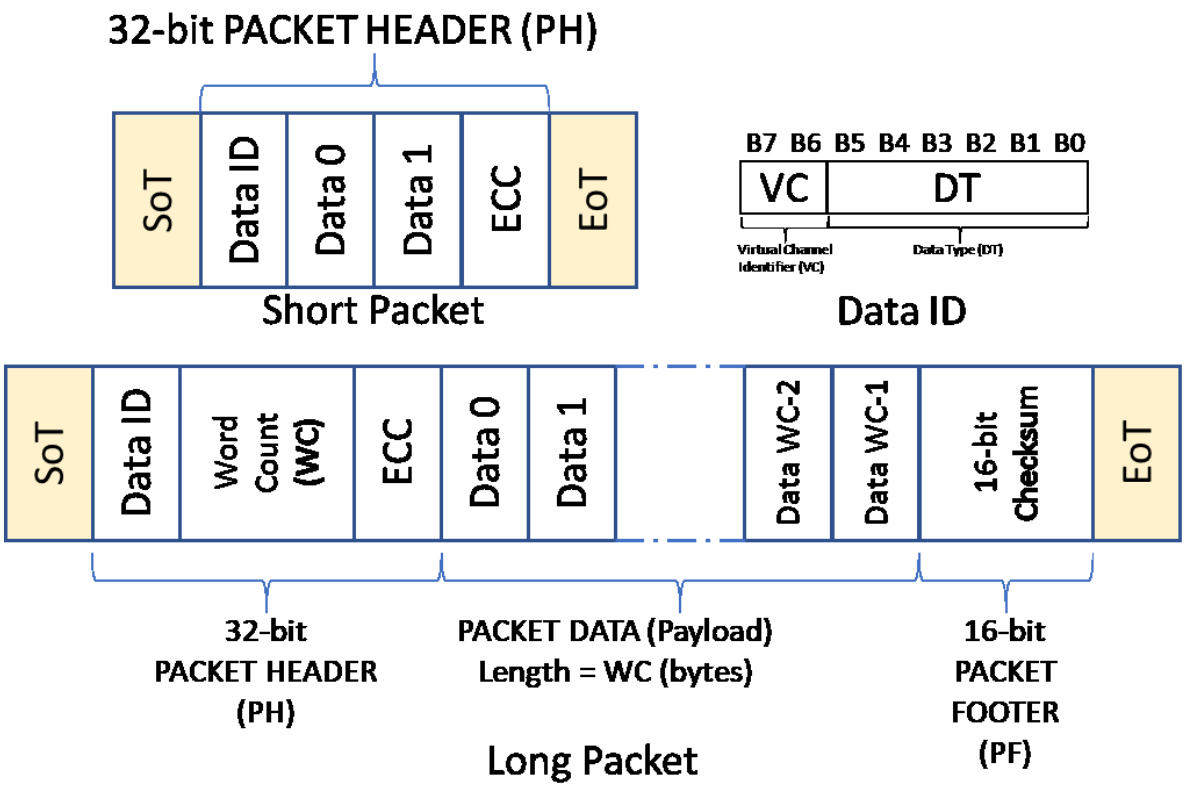


图15. 数据包结构

3.6.1 数据包标头的数据类型

每个数据包标头中的6位字段的数据ID[5:0]定义了数据包类型。表13列出了短数据包的和长数据包的不同数据类型。

表13. 数据类型

数据类型（十六进制）	数据类型（二进制）	描述	数据包类型
0x01	00 0001	同步事件，垂直同步开始	短
0x11	01 0001	同步事件，垂直同步结束	短
0x21	10 0001	同步事件，水平同步开始	短
0x31	11 0001	同步事件，水平同步结束	短
0x08	00 1000	传输包结束（EoTp）	短
0x02	00 0010	彩色模式（CM）关闭命令	短
0x12	01 0010	彩色模式（CM）开启命令	短
0x22	10 0010	关闭外设命令	短

表格在下一页继续...

表13. 数据类型（续）

数据类型（十六进制）	数据类型（二进制）	描述	数据包类型
0x32	11 0010	开启外设命令	短
0x03	00 0011	一般短数据包写入，无参数	短
0x13	01 0011	一般短数据包写入，1个参数	短
0x23	10 0011	一般短数据包写入，2个参数	短
0x04	00 0100	一般数据包读取，无参数	短
0x14	01 0100	一般数据包读取，1个参数	短
0x24	10 0100	一般数据包读取，2个参数	短
0x05	00 0101	DCS短数据包写入，无参数	短
0x15	01 0101	DCS短数据包写入，1个参数	短
0x06	00 0110	DCS读取，无参数	短
0x37	11 0111	设置最大返回数据包大小	短
0x09	00 1001	空数据包，无数据	长
0x19	01 1001	消隐数据包，无数据	长
0x29	10 1001	一般长数据包写入	长
0x39	11 1001	DCS长数据包写入/Write_LUT命令包	长
0x0C	00 1100	松散打包像素流，20位YCbCr，4:2:2格式	长
0x1C	01 1100	打包像素流，24位YCbCr，4:2:2格式	长
0x2C	10 1100	打包像素流，16位YCbCr，4:2:2格式	长
0x0D	00 1101	打包像素流，30位RGB，10-10-10格式	长
0x1D	01 1101	打包像素流，36位RGB，12-12-12格式	长
0x3D	11 1101	打包像素流，12位YCbCr，4:2:0格式	长
0x0E	00 1110	打包像素流，16位RGB，5-6-5格式	长
0x1E	01 1110	打包像素流，18位RGB，6-6-6格式	长
0x2E	10 1110	松散打包像素流，18位RGB，6-6-6格式	长
0x3E	11 1110	打包像素流，24位RGB，8-8-8格式	长

表格在下一页继续...

表13. 数据类型（续）

数据类型（十六进制）	数据类型（二进制）	描述	数据包类型
0xX0和0xXF，未指定	XX 0000 XX 1111	请勿使用 保留所有未指定的代码	

在短数据包中，数据ID字节后的2个字节仅在必要时包含数据。例如，在DCS短数据包写入中，数据ID0x15定义了一个参数命令。

- 数据0字节包含DCS写入命令。
- 数据1字节包含可选的写入参数。

由数据ID 0x22定义的关闭外设命令没有附加数据，因此数据0和数据1字节为空。

在长数据包中，数据ID字节后面的2个字节始终包含字的数量，取值范围为0至65535。

3.7 视频模式接口时序

视频模式是指从主处理器到外围设备的传输所采用的实时像素流形式的操作。正常运行时，为避免在显示图像中出现闪烁或其他可见伪影，显示模块依靠主处理器以足够的带宽来提供图像数据。

DSI对于视频模式数据的传输支持多种格式，或数据包顺序。外围设备的时序要求决定了哪种格式是合适的。在以下章节中，突发模式是指传输的RGB像素（有效视频）部分的时间压缩。此外，这些术语会在后续章节中使用：

- 带同步脉冲的非突发模式：使外围设备能够重建原始视频时序，包括同步脉冲宽度。
- 带同步事件的非突发模式：与上述模式类似，但不需要精确重建同步脉冲宽度，因此单个同步事件被替代。
- 突发模式：RGB像素数据包经过时间压缩，为LP模式（省电模式）或将其他传输复用到DSI链路的行扫描留出更多时间。

注意

为了精确地重建时序，请考虑数据包开销，包括数据ID、ECC，以及校验和字节。这实际上意味着D-PHY链路被配置的带宽有时大于视频流所需的最小带宽。

图16是以下三张分别显示三种视频模式差异的图片的关键。消隐或低功耗间隔（BLLP）是视频数据包（如像素流和同步事件包）不主动传输到外围设备的时间段。

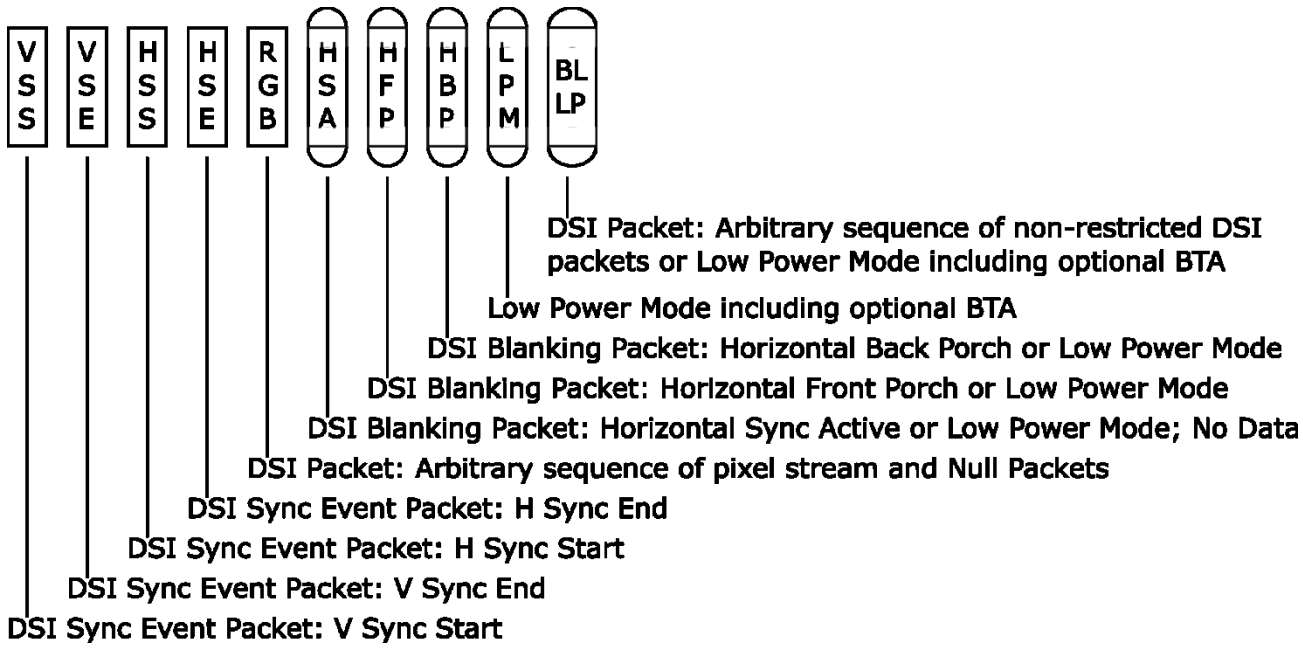


图16. 视频模式接口时序图例

3.7.1 带同步脉冲的非突发模式

以此格式，可以通过DSI串行链路来传送DPI类型的时序，包括匹配DPI像素传输速率和时序事件的宽度，如同步脉冲的宽度等。相应地，同步周期是使用数据包发送的开始同步脉冲和结束同步脉冲这两个来脉冲定义的。图17显示了DPI类型的时序和DSI数据包的关系。

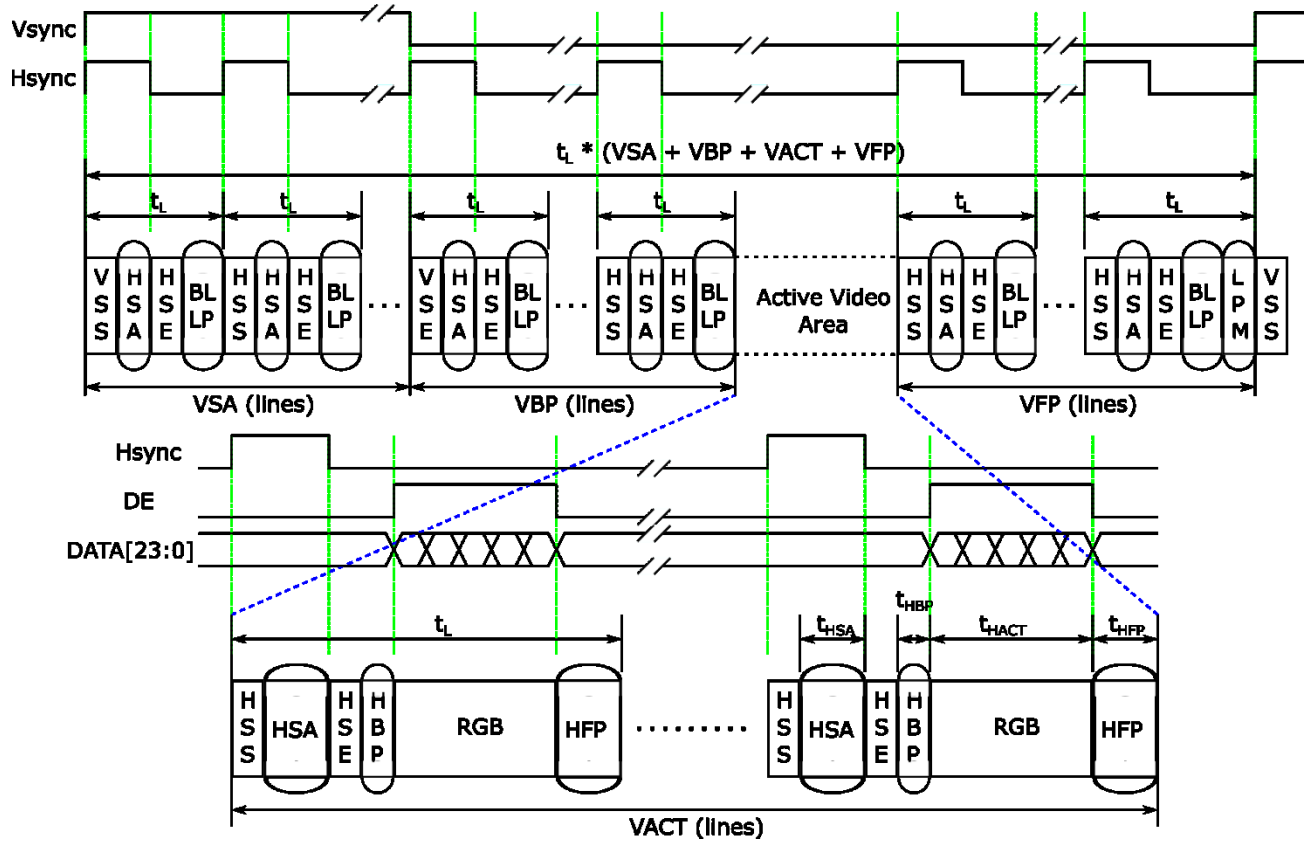


图17. 视频模式接口时序：带同步开始和同步结束的非突发传输

通常，显示为水平同步有效（HSA）、水平后廊（HBP）和水平前廊（HFP）的周期由消隐数据包填充，周期长度（包括数据包开销）的计算结果应与外围设备数据手册规定的周期相符。或者，如果有足够的时间从HS模式转换到LP模式并再次返回，LP模式中的时序间隔可以替代消隐数据包。这样可以省电。在HSA、HBP和HFP期间，总线应保持在LP-11状态。

3.7.2 带同步事件的非突发模式

这种模式是对上面描述的前一种格式的简化，只发送每个同步脉冲的开始。外围设备可以根据需要从接收到的每个同步事件数据包重新生成同步脉冲。像素的传输速率与它们在相应并行显示接口（如DPI-2）中的传输速率相同。图18显示了DPI类型时序与DSI数据包的关系。

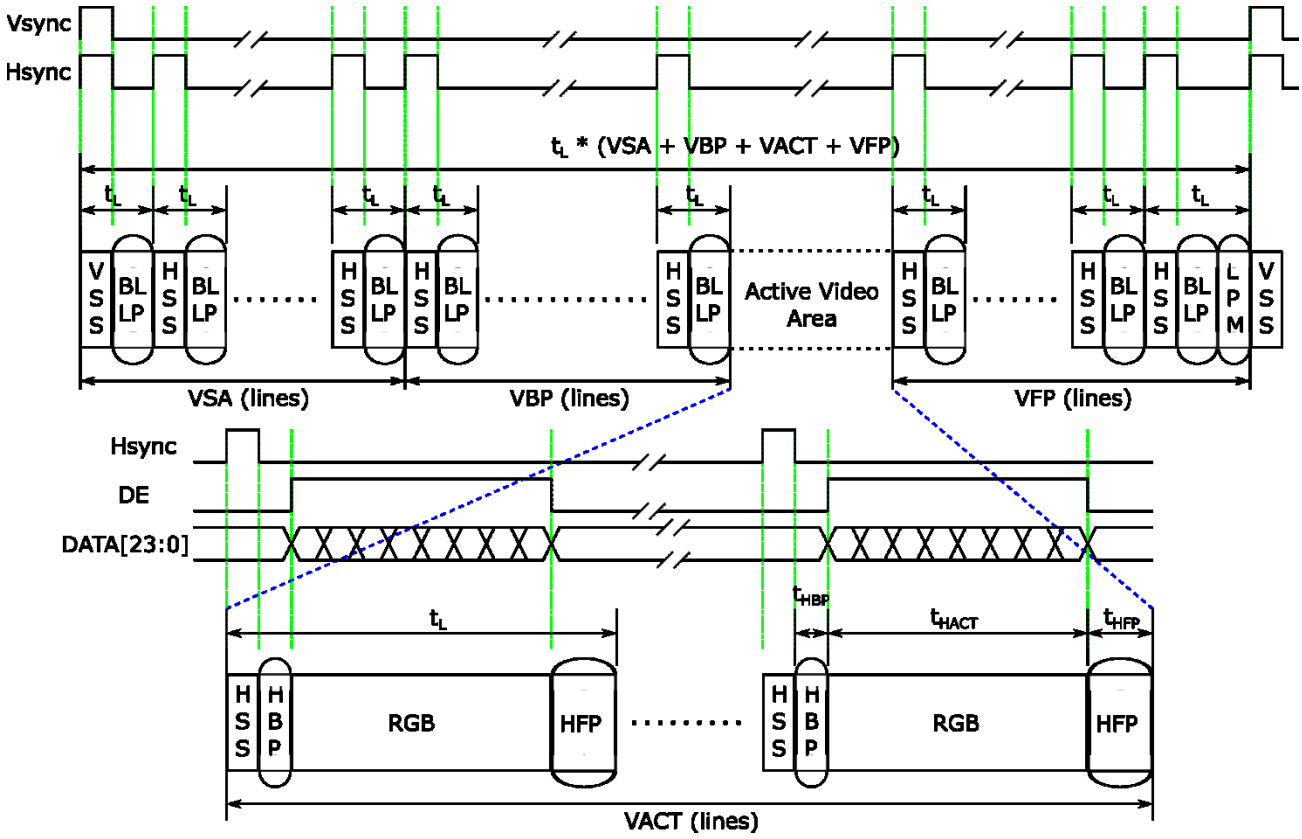


图18. 视频模式接口时序：带同步事件的非突发传输

与前一个非突发模式一样，如果有足够的时间从HS模式转换到LP模式并再次返回，LP模式中的时序间隔可以替代消隐数据包。这样可以省电。

3.7.3 突发模式

在这种模式下，像素数据块可以采用时间压缩突发格式以更短的时间传输。这种策略降低了DSI的总功耗，并使链路上的其他数据传输（无论方向如何）均有更大的时间块。图19显示了DPI类型的时序和DSI数据包的关系。

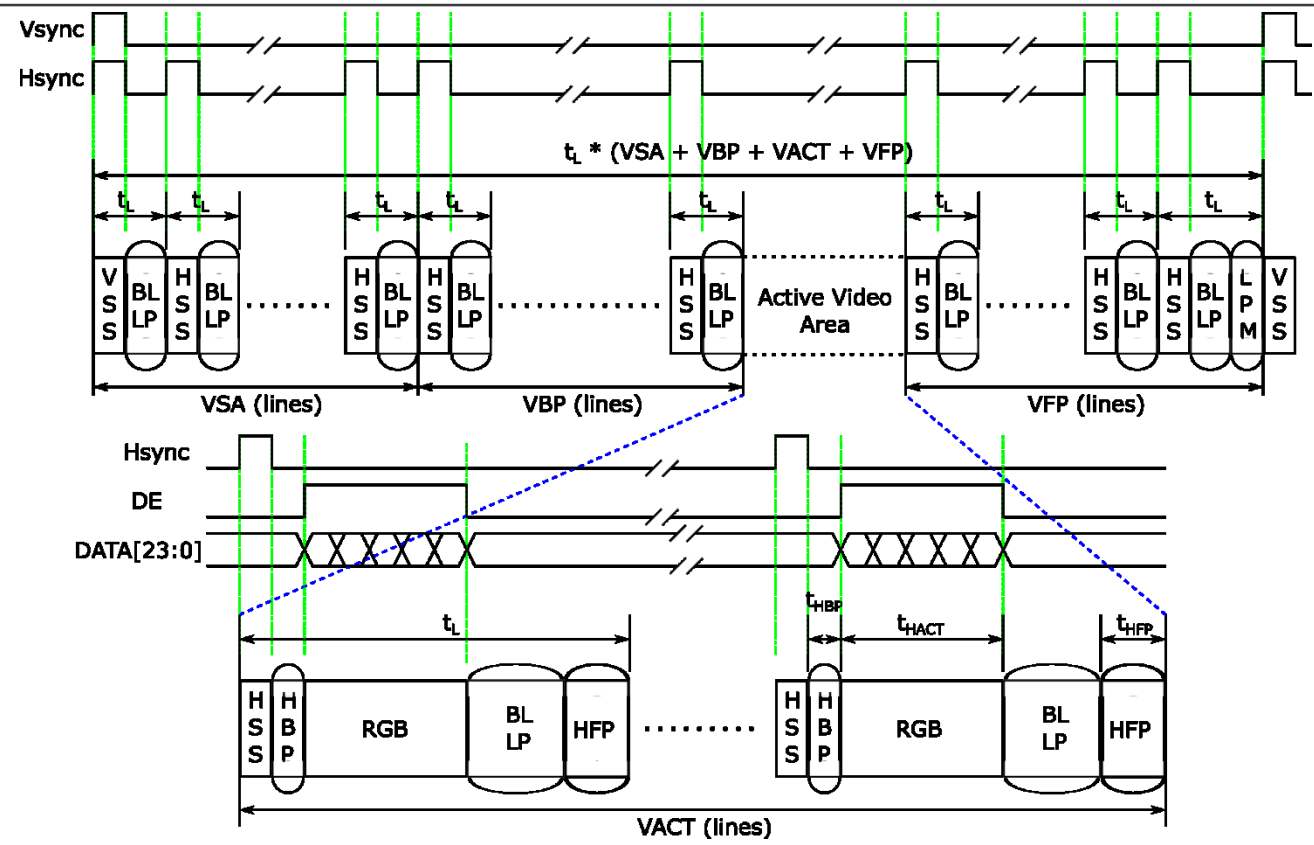


图19. 视频模式接口时序：突发传输

与非突发模式场景类似，如果有足够的时间从HS模式转换到LP模式并再次返回，LP模式中的时序间隔可以替代消隐数据包。这样可以省电。

3.7.4 像素格式

由于所有的D-PHY数据包的最小长度为1字节，DSI将各种像素格式打包为最小公约数的字节长度，从而允许完整传输单个的像素数据。这有时会导致多个像素跨多个字节打包。表14列出了所支持格式的像素位深度和最小字节值。这些值稍后会用于带宽的计算。

表14. DSI像素打包格式

数据格式	每像素位数	每数据包像素（最小）	数据包长度（字节）[1]
松散打包像素流，20位YCbCr，4:2:2	20	2	6
打包像素流，24位YCbCr，4:2:2	24	2	6
打包像素流，16位YCbCr，4:2:2	16	2	4
打包像素流，30位RGB，10-10-10	30	4	15
打包像素流，36位RGB，12-12-12	36	2	9
打包像素流，12位YCbCr，4:2:0	12	2	3

表格在下一页继续...

表14. DSI像素打包格式（续）

数据格式	每像素位数	每数据包像素数（最小）	数据包长度（字节）[1]
打包像素流，16位RGB，5-6-5	16	1	2
打包像素流，18位RGB，6-6-6	18	4	9
松散打包像素流，18位RGB，6-6-6	18	1	3
打包像素流，24位RGB，8-8-8格式	24	1	3

1. 该值是最小数据包长度，根据像素的整数对齐。总线宽（显示的像素加上未显示的像素）必须是该值的整数倍。

3.8 带宽计算

在系统设计期间评估各种显示器时，必须计算一些关键参数，并与主机和显示器数据手册中的参数进行比较。

像素时钟是各种显示处理的主时钟。计算所需像素时钟速率的公式为：

$$\text{像素时钟 (Hz)} = H_{TOT} * V_{TOT} * \text{FPS}$$

公式 1.

这里：

- H_{TOT} = 总行宽=有效像素+HSYNC+HFP+HBP（像素）
- V_{TOT} = 总帧高=有效行+VSYNC+VFP+VBP（行）
- FPS = 每秒帧数

总系统带宽是像素时钟乘以每个像素的位数，可使用以下公式计算：

$$\text{带宽（bps）} = \text{像素时钟} * \text{每个像素的位数}$$

公式 2.

有关每个像素的位数，请参见像素格式。

每通道数据速率（DRPL）是带宽除以系统设计中的数据通道的数量。根据DSI规范，数据通道最少为一个，最多为四个。

$$\text{每通道数据速率（bps）} = \text{带宽/数据通道的数量}$$

公式 3.

MIPI接口是双数据速率，因此D-PHY HS时钟输出频率是每通道数据速率的一半：

$$\text{MIPI D − PHY 时钟速率（Hz）} = \text{每通道数据速率}/2$$

公式 4.

3.9 DSI系统时钟配置

高级MIPI DSI/D-PHY架构在所有i.MX RT和i.MX 8系列芯片中都是通用的。信号/寄存器名称略有不同，但功能相同。图20是时钟系统和数据接口的上层概述。

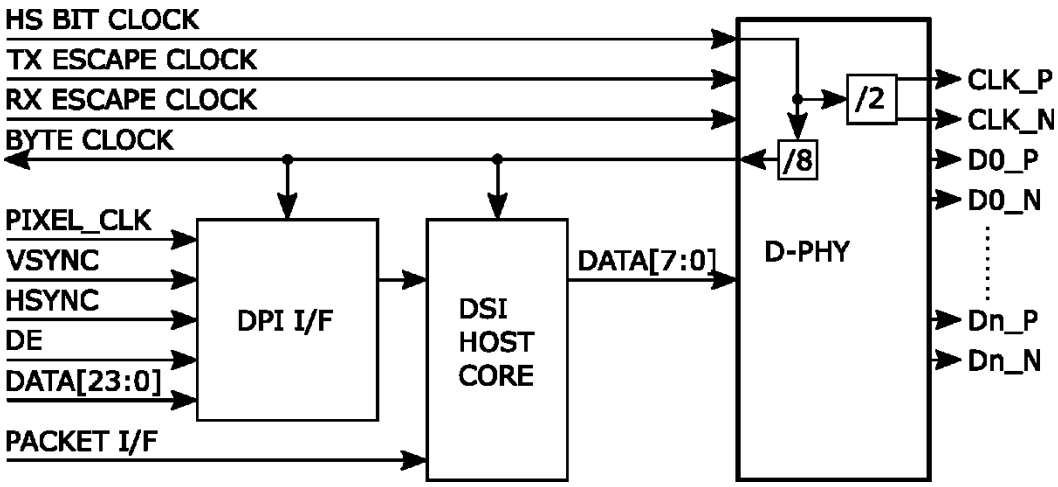


图20. MIPI DSI/D-PHY时钟结构

3.9.1 HS位时钟

D-PHY使用HS位时钟来生成高速DDR时钟和字节时钟。该时钟的速度等于每通道的数据速率。在大多数SOC中，专用的D-PHY锁相环（PLL）会生成该时钟，但i.MX RT500没有D-PHY锁相环。

3.9.2 字节时钟

D-PHY产生字节时钟，频率为HS位时钟数据速率的1/8。D-PHY被配置为生成1Gbps高速数据、生成125MHz的字节时钟。DSI主控制器的数据包接口与字节时钟同步。

3.9.3 脱离模式时钟

MIPI D-PHY使用TX脱离时钟进行状态控制和低功耗数据传输。该脱离时钟的频率范围在12MHz和20MHz之间。

MIPI D-PHY使用RX脱离时钟翻转低功耗数据的接收。MIPI规范要求该时钟频率为显示器LP时钟频率的67%至150%。

3.9.4 像素时钟

DPI-2接口使用像素时钟。所有DPI-2信号都与该时钟同步。DSI主控制器的DPI桥接模块负责将像素时钟域接收到的视频数据传输到字节时钟域。像素和字节时钟频率有以下公式中的关系：

字节时钟频率 > = 像素时钟频率 * 每像素位数 / (8 * (DPHY通道数))

公式 5.

这里：

- 字节时钟频率=字节时钟的频率=HS位时钟/8
- 像素时钟频率=DPI-2接口上像素时钟的频率。
- 每个像素的位数=DPI-2接口上的像素大小，单位为比特（见表14）。
- DPHY_LANE_NUM=MIPI D-PHY有效数据通道数（1-4）。

注意

如果字节时钟频率不满足这一要求，则MIPI接口无法跟上DPI-2接口上的视频流，视频线路将会中断。

4 DSI示例

i.MX 8和i.MX RT系列基于其底层工艺节点采用了两个不同的DSI/D-PHY模块。有关工艺节点和芯片差异的信息，请参见i.MX 8M和i.MX RT芯片的特性/差异。

4.1 i.MX RT1170 DSI示例

i.MX RT1170 EVK有一个可选的显示面板，可以在恩智浦网页单独购买。这个显示器是一块5.5英寸的LED TFT面板，分辨率为720像素乘1280像素，带有电容式触摸覆层。图21显示连接到RT1170 EVK的显示器。

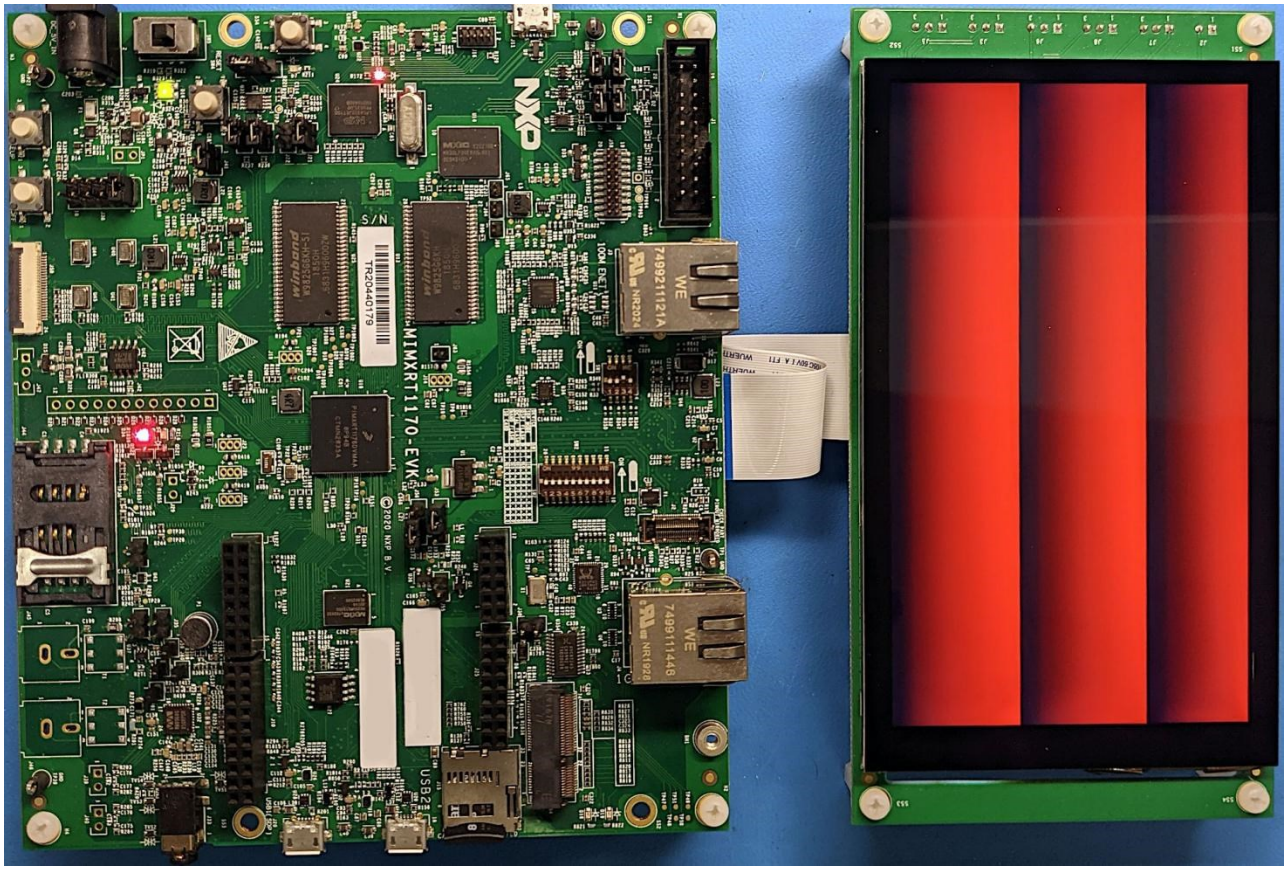


图21. 带5.5英寸TFT显示器的RT1170

对于本例，我们将通过SDK附带的ELCDIF RGB演示工程来了解所需的系统设置。ELCDIF RGB工程展示了如何使用ELCDIF驱动程序驱动TFT面板。如果此示例正确运行，一个矩形会在屏幕上移动，每次到达屏幕边缘时颜色都会发生变化。

4.1.1 初始系统设计

设置各种时钟和控制寄存器之前，必须在像素深度、帧速率和显示分辨率之间进行平衡。可以从带宽计算章节的四个带宽计算的解决开始。

此TFT显示器使用Raydium RM68200显示控制器IC。RM68200数据手册中的DSI视频模式时序参数如表15所示。

表15. RM68200的DSI参数

参数	描述	最小值	类型值	最大值	单位	条件
br_{PHY}	所有通道的总比特率	250		850	Mbps	WXGA
TBR2	总比特率			1.7	Gbps	WXGA分辨率，2通道
t_L	行扫描时间		12.8		µs	FPS = 60 Hz，VBP = 8，VFP = 8
t_{HSA}	水平同步有效时间				µs	
t_{HBP}	水平后廊	0.5			µs	WXGA
t_{HACT}	图像数据时间	9.6		19.2	µs	
HACT	每行有效像素		800[1]		像素	WXGA
t_{HFP}	水平前廊	0.5			µs	
VSA	垂直同步有效时间	1			行	
VBP	垂直后廊		8		行	
VACT	每帧有效扫瞄行数		1280		行	
VFP	垂直前廊	6	8		行	

1. RM68200可以支持每行800像素，但5.5英寸TFT LED显示器仅支持720像素。

在本例中，使用了以下参数（从源代码中提取）：

- 面板高度=1280
- 面板宽度=720
- HSW=8（又名HSYNC或HSA）
- HFP=32
- HBP=32
- VSW=2（又名VSYNC或VSA）
- VFP=16
- VBP=14

使用这些参数和60 Hz的帧速率，所需的像素时钟可以计算如下：

- 像素时钟 =（1280+2+16+14）*（720+8+32+32）*60 FPS = 62,346,240 Hz

本例的像素格式是RGB888，使用24 bpp的位深。这种设置的带宽可以计算为：

- 带宽 = 62,346,240*24 = 1,496,309,760 bps

该系统使用2通道MIPI接口，因此每个通道的数据速率为：

- DRPL = 1,496,309,760 bps/2 = 748,154,880 bps

因此，MIPI D-PHY HS位时钟必须等于或大于DRPL。

- MIPI HS位时钟 ≥ 748,154,880 Hz

4.1.2 时钟设置

时钟设置是系统问题的常见来源。必须选择根时钟。除法器、乘法器和/或锁相环必须正确配置。此系统示例要求正确设置以下时钟：

- 像素时钟
- 脱离模式时钟
- MIPI HS位时钟

4.1.2.1 像素时钟设置

首先为eLCDIF模块设置像素时钟。该模块将像素时钟与像素数据一起输出到DSI模块的DPI桥。CLOCK_ROOT69控制ELCDIF_CLK_ROOT。示例代码的检查会显示以下寄存器设置：

- CLOCK_ROOT69_CONTROL[MUX] = 4
- CLOCK_ROOT69_CONTROL[DIV] = 9

多路开关（mux）设置为4，选择SYS_PLL2_CLK运行至528 MHz。DIV设置为9时，eLCDIF像素时钟=528 MHz/9=58.666666667 MHz。

注意

58.67 MHz PCLK小于60 MHz，因此帧速率小于60 Hz。

4.1.2.2 脱离（Escape）时钟设置

CLOCK_ROOT72控制TX脱离模式时钟根（MIPI_ESC_CLK_ROOT）。示例代码检查会显示以下寄存器设置：

- CLOCK_ROOT72_CONTROL[MUX] = 1
- CLOCK_ROOT72_CONTROL[DIV] = 1

多路开关设置为1选择OSC24M，即24 MHz振荡器。DIV设置为1会导致除以1，因此脱离时钟根被设置为24MHz。RX和TX脱离时钟有不同的要求。有一个专门的时钟分频器，只用于TX脱离时钟。脱离时钟分频器的示例代码有以下设置：

- EscClockGroupConfig.DIV0 = 1

其结果是12MHz的TX脱离时钟和24MHz的RX脱离时钟。

注意

围绕i.MX RT系列的时钟分频器可能会有一些混淆：

1. 参考手册显示时钟根控制寄存器（clock_ROOT0_control-CLOCK_ROOT78_CONTROL）会将分频器选择加1：即所选时钟除以DIV + 1。
2. i.MX RT SDK函数CLOCK_SetRootClock（ ）从用户定义的DIV中减1。因此用户不必记住分频器实际上是DIV+1。
3. i.MX RT SDK函数CLOCK_SetGroupConfig不会从用户定义的DIV中减1。因此用户必须记住实际的分频数=DIV+1。

4.1.2.3 HS位时钟设置

RT1170 D-PHY有生成HS位时钟的内部锁相环。D-PHY会自动生成字节时钟。字节时钟总是HS位时钟频率的1/8。回想最初的系统设计，所需的HS位时钟必须 ≥ 748,154,880 Hz。

由于数据包的开销（数据ID、ECC、校验和字节，其他等等）与数据一起传输，将HS位时钟设置为748,154,880 Hz并不能提供足够的系统带宽。在本例中，我们使用的乘数值为1.125，这提供了额外的带宽余量。锁相环需要为所需频率配置一个参考时钟和三个分频器（M、N和O）。使用以下公式配置D-PHY锁相环时钟：

$$D-PHY\ PLL时钟输出 = REF\_CLK * (CM / (CN*CO))$$

公式 6.

图22显示了锁相环的基本操作。

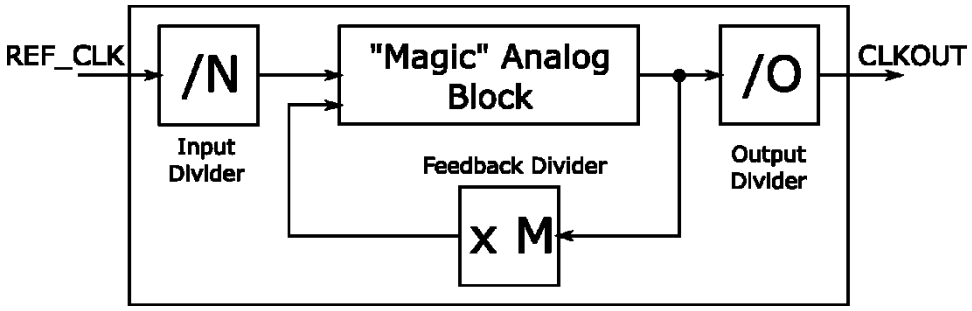


图22. 28FDSOI D-PHY锁相环框图

CLOCK_ROOT71控制D-PHY锁相环参考时钟（REF_CLK）根（MIPI_REF_CLK_ROOT）。示例代码的检查显示以下寄存器设置：

- CLOCK_ROOT71_CONTROL[MUX] = 1
- CLOCK_ROOT71_CONTROL[DIV] = 1

此设置将D-PHY锁相环参考根时钟配置为使用24 MHz振荡器。

使用像素时钟设置的58.667 MHz频率、24 bpp、2个数据通道和1.125倍率，得到以下期望的锁相环频率：

D-PHY位时钟 = ((58.66667 * 24) / 2) * 1.125 = 792 MHz

如果给定了期望输出频率和参考时钟频率，SDK函数DSI_DphyGetPllDivider（CN、CM、CO、refClkFreq_Hz、desiredOutFreq_Hz）会尝试为CM、CN和CO找到正确的分频值。如果找不到可行的值，则返回错误代码。示例代码使用以下分频值：

- CN = 0x1F /* = 1 */
- CM = 0xC1 /* = 33 */
- CO = 0x0 /* = 1 */

此代码的结果是D-PHY锁相环时钟输出= 24 MHz * (33 / (1 * 1)) = 792 MHz。

图23是D-PHY HS传输时钟的差分示波图。由于这是DDR时钟方案，实际频率为792 MHz/2 = 396 MHz。

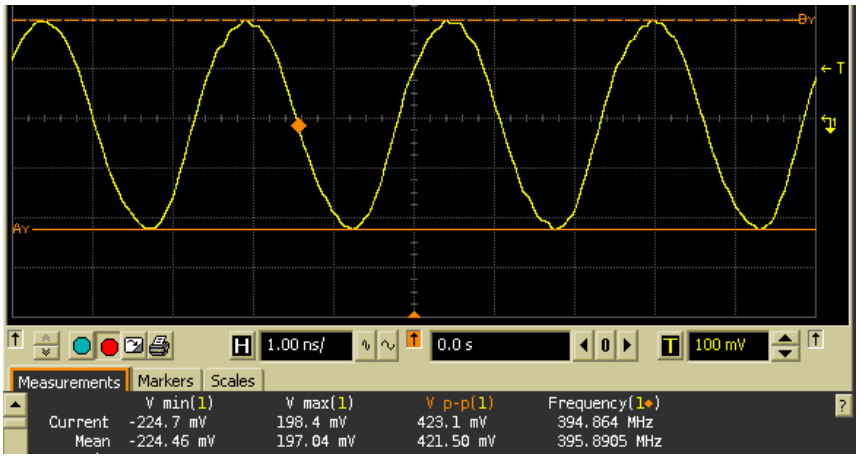


图23. MIPI DSI D-PHY时钟输出（差分）

4.1.3 D-PHY设置

如“HS传输开始”章节中所述，必须配置一些D-PHY参数才能正确地进入和退出HS传输模式。

表16列出了直接影响HS数据传输的D-PHY时序参数的寄存器。

表16. RT1170 D-PHY配置寄存器

参数	寄存器	描述
以下参数调整进入HS模式		
$T_{CLK-PRE}$	CFG_T_PRE[NUM_PERIODS]	使用字节时钟周期的D-PHY时序参数。此端口的最小值为1。 $T_{CLK-PRE}$ = 数字周期×（TxByteClkHS）[1] 0 = 零时钟周期（不支持） 1 = 一个时钟周期（所需最小值） 2 = 两个时钟周期 ... ...
$T_{HS-PREPARE}$	M_PRG_HS_PREPARE	00：$T_{HS-PREPARE}$ = 1×TxClkEsc周期 01：$T_{HS-PREPARE}$ = 1.5×TxClkEsc周期 10：$T_{HS-PREPARE}$ = 2×TxClkEsc周期 11：$T_{HS-PREPARE}$ = 2.5×TxClkEsc周期
$T_{CLK-PREPARE}$	MC_PRG_HS_PREPARE	0：$T_{CLK-PREPARE}$ = 1×TxClkEsc周期 1：$T_{CLK-PREPARE}$ = 1.5×TxClkEsc周期
$T_{HS-ZERO}$	M_PRG_HS_ZERO	$T_{HS-ZERO}$ =（M_PRG_HS_ZERO+6）×（TxByteClkHS周期）[1]

表格在下一页继续...

表16. RT1170 D-PHY配置寄存器（续）

参数	寄存器	描述
$T_{CLK-ZERO}$	MC_PRG_HS_ZERO	$T_{CLK-ZERO}$ =（MC_PRG_HS_ZERO+3）×（TxByteClkHS周期）[1]
以下参数调整退出HS模式		
$T_{CLK-POST}$	CFG_T_POST	$T_{CLK-POST}$ = NUM_PERIODS ×（TxByteClkHS）[1] 0 = 零时钟周期（不支持） 1 = 一个时钟周期（需要最小值） 2 = 两个时钟周期 ... ...
T_{EOT}	CFG_AUTOINSERT_EOTP	使主控制器能够在从HS模式切换到LP模式时自动插入EoTp短数据包。 0：EoTp不自动插入 1：EoTp自动插入
	CFG_EXTRA_CMDS_AFTER_EOTP	将DSI主控制器配置为在数据包结束后发送额外的传输结束数据包。该值是所发送的额外EOTP数据包的数量。 0 = 零个数据包 1 = 一个数据包 2 = 两个数据包 ... ...
$T_{HS-EXIT}$	CFG_TX_GAP	$T_{HS-EXIT}$ = NUM_PERIODS ×（TxByteClkHS）[1] 0 = 零个时钟周期（不支持） 1 = 一个时钟周期（所需最小值） 2 = 两个时钟周期 ... ...
$T_{HS-TRAIL}$	M_PRG_HS_TRAIL	$T_{HS-TRAIL}$ =（M_PRG_HS_TRAIL）×（TxByteClkHS周期）[1]
$T_{CLK-TRAIL}$	MC_PRG_HS_TRAIL	$T_{CLK-TRAIL}$ =（MC_PRG_HS_TRAIL）×（TxByteClkHS周期）[1]

1. TxByteClkHS=MIPI HS位时钟速率/8

4.1.3.1 HS传输模式启动设置

如表5所示，必须配置一些时序参数才能正确进入HS传输模式。图24是个波形图，显示了：

- 使用差分探针的MIPI DSI时钟通道
- 使用单端探针的MIPI_DSI_DP0和MIPI_DSI_DN0信号

图24显示了D-PHY从LP模式转换到HS模式时驱动程序代码配置的时序参数。

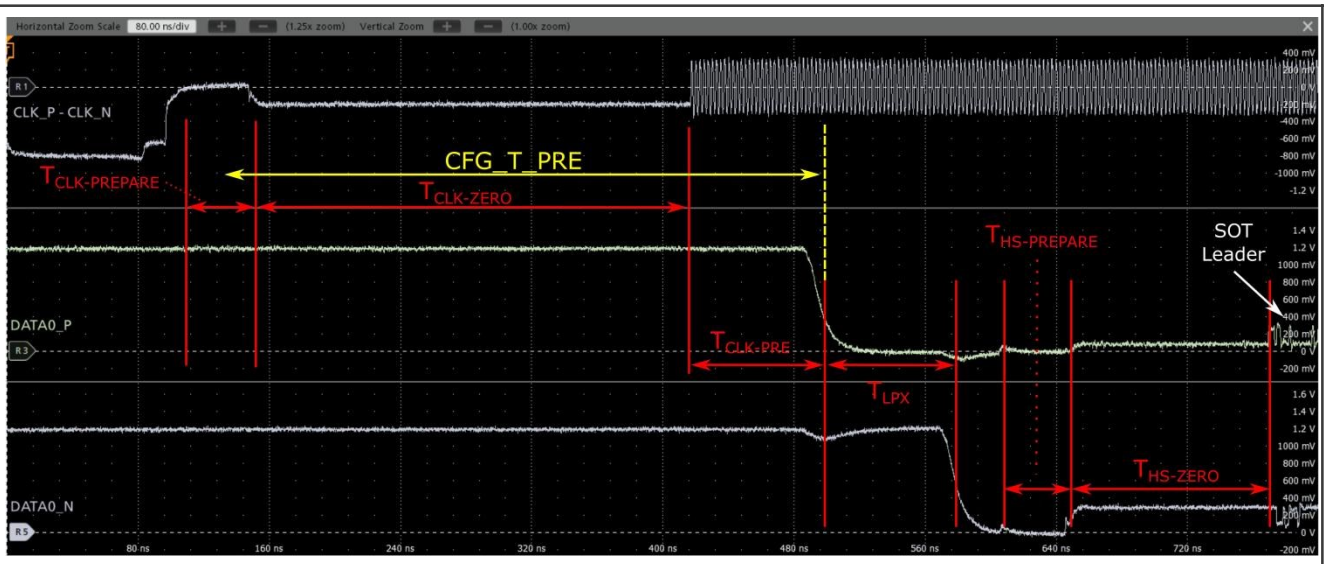


图24. 从LP模式转换到HS模式

对于本例，配置寄存器的目标是以下时间段：

- UI是MIPI HS位时钟周期，在本例中=1/792 MHz=1.26263 ns
- TxByteClkHS周期=8×MIPI HS位时钟=8×1.26263 ns=10.10 ns
- TxClkEsc周期=1/12 MHz=83.33 ns

4.1.3.1.1 TCLK-PRE

$T_{CLK-PRE}$参数通过`CFG_T_PRE[NUM_PERIOD]`寄存器进行调整。该参数设置的是在为HS操作开启时钟通道之后，且在为HS操作开启数据通道之前，控制器等待的TxByteClkHS时钟周期的数目。MIPI规范要求最小$T_{CLK-PRE}$为8×1.2626 ns=10.10 ns。在该示例中，$T_{CLK-PRE}$约为80ns。

CFG_T_PRE定时器不是在$T_{CLK-ZERO}$计时周期后启动，而是在$T_{CLK_PREPARE}$时间窗口附近开始。因此，CFT_T_PRE值要远大于所需的T_{CLK_PRE}值。

- CFG_T_PRE = 0 × 24 = 36
- CFG_T_PRE = 36 × 10.10 ns = 363.6 ns

4.1.3.1.2 HS准备定时器

$T_{CLK-PREPARE}$必须介于38 ns和95 ns之间。$T_{HS-PREPARE}$必须介于（40 ns+4×UI）= 45 ns和（85 ns+6×UI）= 92.576 ns之间。

- MC_PRG_HS_PREPARE[1:0] = 0
- $T_{CLK-PREPARE}$ = 1 × 41.67 ns = 41.67 ns
- M_PRG_HS_PREPARE[1:0] = 0
- $T_{HS-PREPARE}$ = 1 * 41.67 ns = 41.67 ns

4.1.3.1.3 HS零定时器

$T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$必须大于300 ns。由于$T_{CLK-PREPARE}$为41.67 ns，$T_{CLK-ZERO}$必须大于258.33 ns（300–41.67=258.33）。

- MC_PRG_HS_ZERO = 0 × 17 = 23
- $T_{CLK-ZERO}$ = (23 + 3) × (10.10 ns) = 262 ns

$T_{HS-ZERO}$必须大于（105 ns+6×UI）= 112.58 ns。

- M_PRG_HS_ZERO = 0x6
- $T_{HS-ZERO}$ = (6 + 6) × (10.10 ns) = 121.2 ns

4.1.3.2 HS传输模式退出设置

如表7所述，当HS传输转换回LP模式时，一些时序参数会影响HS传输的结束。图25说明了当D-PHY从HS模式转换到LP模式时驱动程序代码配置的这些参数。

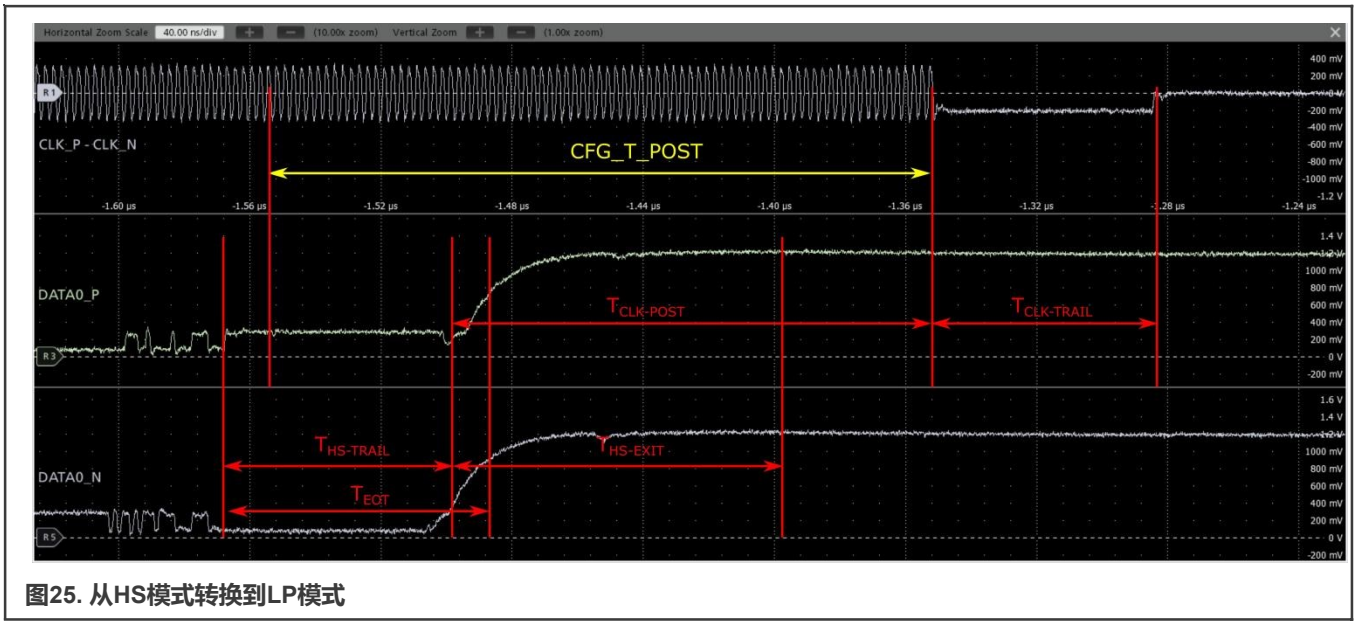


图25. 从HS模式转换到LP模式

4.1.3.2.1 $T_{CLK-POST}$

CFG_T_POST寄存器用于以TxByteClkHS周期的倍数调整$T_{CLK-POST}$参数。$T_{CLK-POST}$必须大于（60 ns+52×UI）=125.66 ns。在本例中，TCLK-POST约为148 ns。

- CFG_T_POST[NUM_PERIODS] = 0 × 14 = 20
- CFG_T_POST周期（ns）= 20 × 10.10 = 202 ns

4.1.3.2.2 T_{EOT}

T_{EOT}必须小于（105 ns+12×UI）= 120.15 ns。

可以使用`CFG_AUTOINSERT_EOTP`和`CFG_EXTRA_CMDS_AFTER_EOTP`寄存器调整此参数。在本例中，CFG_AUTOINSERT_EOTP=0，因此不发送额外的数据包，T_{EOT}约为84 ns。

4.1.3.2.3 $T_{HS-EXIT}$

CFG_TX_GAP寄存器用于以TxByteClkHS周期的倍数调整$T_{HS-EXIT}$参数。$T_{HS-EXIT}$必须大于100 ns。

- CFG_TX_GAP[NUM_PERIODS] = 0xA = 10
- $T_{HS-EXIT}$ = 10 * 10.10 = 101 ns.

4.1.3.2.4 HS跟踪定时器

$T_{CLK-TRAIL}$必须大于60 ns，并且必须配置。它介于（60 ns+4×UI）=65.05 ns和（105 ns+12×UI）=120.15 ns之间。

- MC_PRG_HS_TRAIL = 0x7
- M_PRG_HS_TRAIL = 0x7
- $T_{CLK-TRAIL}$ = 7 × 10.10 ns = 70.7 ns
- $T_{HS-TRAIL}$ = 7 × 10.10 ns = 70.7 ns

4.1.4 DSI上的DPI-2视频时序再造

DSI模块有一个DPI-2主机桥内核，它：

- 接受兼容MIPI的DPI-2信号集（V_{sync}、H_{sync}、像素数据等）
- 为视频时序事件和视频数据创建DSI数据包
- 通过DSI主控制器的数据包接口传输这些数据包

该模块通过适当地分隔携带DPI视频信息的DSI数据包并将消隐数据包插入数据包流来保持视频时序。

为了实现这一目标，DSI主控制器必须正确设置HFP、HBP和HSA配置寄存器。这些寄存器的值以DSI数据包字节的形式反映了HFP、HBP和HSA的绝对时间。图26显示了对于单个显示行的DPI接口（基于像素时钟的时序）和DSI接口（基于字节大小数据包的时序）之间的关系。

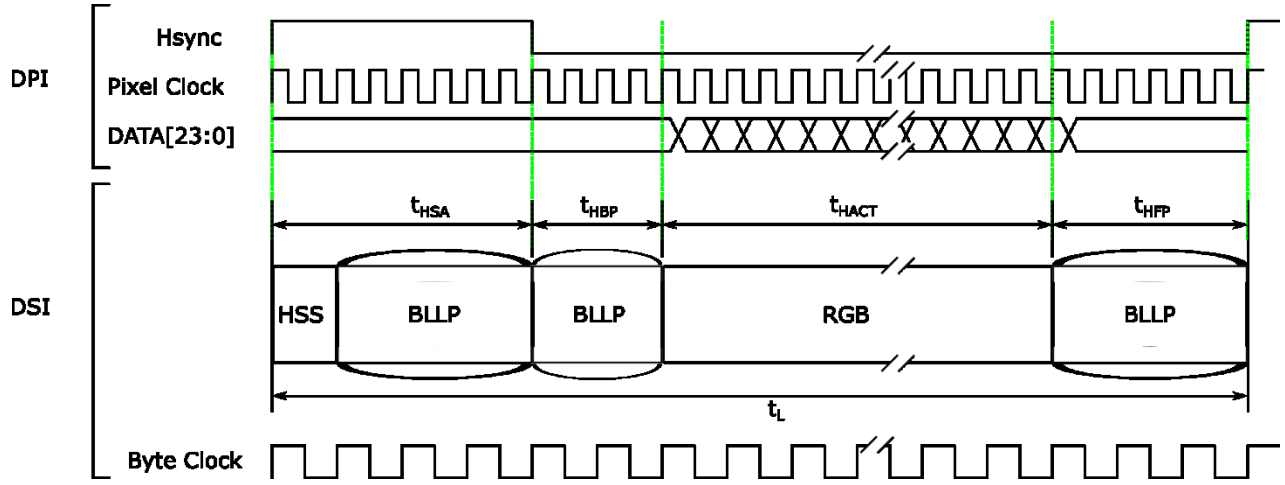


图26. DPI视频时序与DSI数据包的关系——突发模式

要将DPI接口上的像素大小与DSI数据包字节相关联，请使用以下关系：

DPI事件（时间）= [发送时间] * [DSI字节数]

公式 7.

DPI事件所需的时长由HSA、HFP或HBP等DSI参数确定。

DPI事件 = DPI事件大小（像素）/DPI_Pclk频率（Hz）

公式 8.

传输时间 = 1/（MIPI字节时钟（Hz）* 通道数）

公式 9.

要求DSI字节数`number_of_dsi_bytes`，请使用以下公式：

DSI_Bytes = (DPI_Event_size * MIPI 字节时钟 * cfg_num_lanes) / DPI_Pclk_Frequency

公式 10.

这里：

- `DSI_Bytes`：在寄存器中设置的HFP、HBP或HSA的值。
- `DPI_Event_size`：以像素为单位的视频事件（HFP、HBP、HSA）大小。
- `DPI_Pclk_frequency`：像素时钟频率（Hz）。
- MIPI Byte Clock：高速模式下的MIPI数据通道字节时钟频率（Hz）。
- `cfg_num_lanes`：有效D-PHY通道数。

4.1.4.1 DPI-2/DSI桥接设置

表17. 28 nm DPI-2/DSI桥接配置寄存器

寄存器	描述
PIXEL_PAYLOAD_SIZE[15:0]	作为一个DSI数据包发送的最大像素数。建议可以被行数（以像素为单位）整除。
PIXEL_FIFO_SEND_LEVEL[15:0]	为了优化DSI应用，DPI桥会在启动DSI数据包之前缓冲一些DPI像素。此配置端口控制DPI主机桥开始发送像素的水平。
INTERFACE_COLOR_CODING[2:0]	按照DPI规范的规定，设置24位数据总线中RGB位的分布。 000：16位配置1 001：16位配置2 010：16位配置3 011：18位配置1 100：18位配置2 101：24位 110、111：保留
PIXEL_FORMAT[1:0]	设置像素的DSI数据包类型 00：16位 01：18位 10：18位松散打包

表格在下一页继续...

表17. 28 nm DPI-2/DSI桥配置寄存器（续）

寄存器	描述
	11：24位
VSYNC_POLARITY[0]	设置dpi_vsync_input的极性 0：低有效 1：高有效
HSYNC_POLARITY[0]	设置dpi_hsync_input的极性 0：低有效 1：高有效
VIDEO_MODE[1:0]	选择主机DPI模块为其生成数据包的DSI视频模式。 00：带同步脉冲的非突发模式 01：带同步事件的非突发模式 10：突发模式 11：保留，无效
HFP[15:0]	设置水平前廊消隐数据包的DSI数据包载荷大小（以字节为单位）
HBP[15:0]	设置水平后廊消隐数据包的DSI数据包载荷大小（以字节为单位）。
HSA[15:0]	设置水平同步宽度填充消隐数据包的DSI数据包载荷大小（以字节为单位）。
ENABLE_MULT_PKTS[0]	启用每个视频线路多个数据包。启用后，必须将PIXEL_PAYLOAD_SIZE设置为视频线路大小的一半。 0：视频线路以单个数据包发送 1：视频线路分两个数据包发送
VBP[7:0]	设置垂直后廊中的行数。
VFP[7:0]	设置垂直前廊中的行数。
BLLP_MODE[0]	在可能时将BLLP期间优化为低功耗模式。 0：在BLLP期间发送消隐数据包 1：LP模式用于BLLP期间
USE_NULL_PKT_BLLP[0]	选择BLLP期间要发送的消隐数据包类型。 0：BLLP区域1中使用消隐数据包 1：BLLP区域中使用空数据包
VACTIVE[13:0]	设置垂直有效区域的行数。

表格在下一页继续...

表17. 28 nm DPI-2/DSI桥接配置寄存器（续）

寄存器	描述
VC	设置要发送到接收数据包接口的数据包虚拟通道（VC）。虚拟通道不等于该值的数据包会被丢弃，并在DSI错误报告中设置DSI VC ID无效位（位12）。

`DSI_SetDpiConfig()`函数用于设置DPI配置寄存器。下面我们检查表17中的各个寄存器设置。

- `PIXEL_PAYLOAD_SIZE = 720`

  该寄存器是以像素为单位的有效水平行的宽度（HACT）。

- `PIXEL_FIFO_SEND_LEVEL = 720`

  该寄存器设置DSI启动数据包传输的限制点。

- `INTERFACE_COLOR_CODING = 101`

  该寄存器将DPI输入配置为24位RGB像素格式。

- `PIXEL_FORMAT = 11`

  该寄存器将DSI输出包配置为24位RGB像素格式。

- `VIDEO_MODE = 10`

  该寄存器将视频模式接口配置为使用突发模式。

- `ENABLE_MULT_PKTS = 0x0`

  用单个数据包发送一整行像素。

- `VBP = 0xE = 14`

  垂直后廊有14行。

- `VFP = 0x10 = 16`

  垂直前廊有16行。

- `BLLP_MODE = 0x1`

  该寄存器将MIPI通道配置为在BLLP期间进入LP模式。

- `USE_NULL_PKT_BLLP = 0x0`

  对于BLLP期间的`0x0`，DSI发送消隐数据包，而不是空数据包。

- `VACTIVE = 0x4ff = 1279`

  该寄存器设置垂直有效行的总数。由于第一行从0开始，因此该数字等于VACT–1。

RT1170不支持虚拟信道，但其他28FDSOI芯片（如i.MX 8Q）支持。

4.1.4.2 字节长度—消隐参数转换

大多数配置相当直接，可从显示器数据手册中提取，但水平消隐参数确实需要一些操作才能在像素时间和DSI字节时间之间转换。回顾本章节前面的内容，DPI和DSI事件之间的关系由以下公式确定：

$$\text{DPI事件} = [\text{传输时间}] * [\text{DSI字节}]$$

公式 11.

用公式11，我们可以创建简单的乘法系数，将DPI位长度转换为DSI字节长度：

$$\text{UINT32 coff} = (\text{numLanes} * \text{dsiHsBitClkFreq\_Hz}) / (\text{dpiPixelClkFreq\_Hz} * 8\text{U});$$

公式 12.

在本例中，系数计算为：

*coff = (2 * 792 MHz) / (58.667 MHz * 8) = 3*

回顾RM68200数据手册中定义的初始水平消隐参数：

- HSA = 8
- HFP = 32
- HBP = 32

将这三个值乘以系数（3）后，我们得到：

```
HSA = 0x18 = 24
HFP = 0x60 = 96
HBP = 0x60 = 96
```

如图26所示，HSA使用HSS短字节数据包类型及随后的长消隐数据包作为信号。两个数据包的宽度为24字节。

HFP和HBP均使用长度为96字节的长消隐数据包来作为信号。

4.1.5 DSI示例时序总结

图27显示了MIPI时钟和DATA0通道从LP模式转换到HS模式，并返回以处理一行像素。我们示例的实际帧速率为58.6667 MHz / 1,039,104像素 = 56.459 Hz。

由于有1312个垂直行，每行需要大约1/（帧速率）/Vtot = 1/56.459 Hz/1312 = 13.5 us的时间来传输像素数据。图27中的标记正好显示了一个水平图像行的时序。

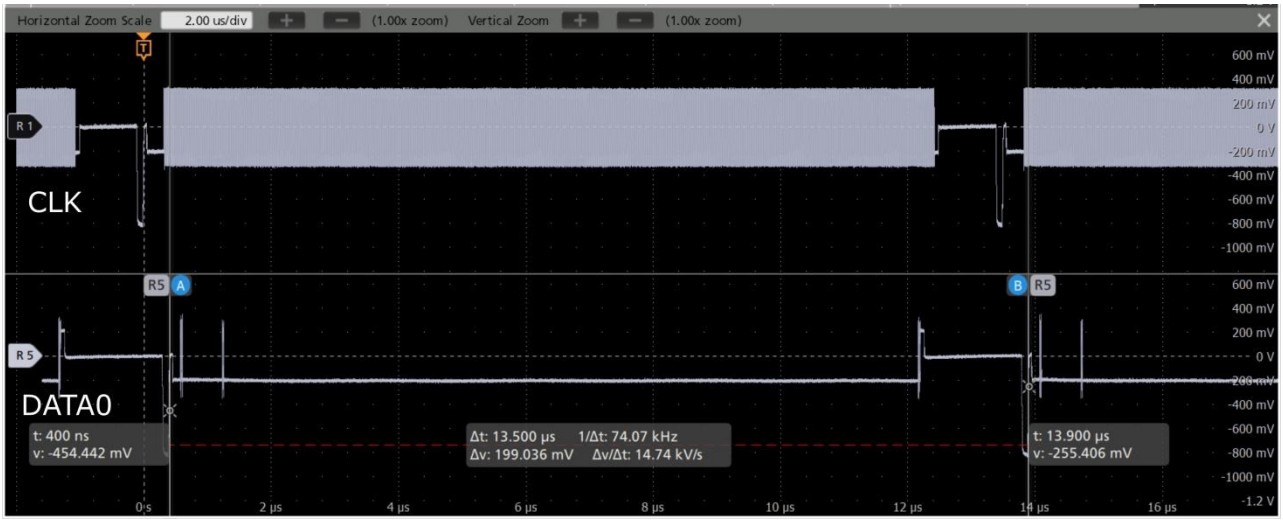


图27. DSI单个有效行时序

5 CSI-2

本节介绍CSI-2接口以及摄像头或图像传感器如何连接到处理器。读过本节后，工程师可以评估各种图像传感器或摄像头模块，确定与本文档中描述的所有i.MX处理器的适用性和兼容性。系统工程师可以指定一个完整的图像系统的硬件。本节不描述在软件中实现完整图像系统的细节，不涵盖诸如ISP校准、曝光控制、白平衡、软件驱动程序等信息。

5.1 综述

CSI-2规范遵循与DSI规范相同的通用架构，采用并行图像数据总线，并将其序列化为字节大小的数据块，以便在PHY上进行高速传输。最大的区别是增加了单独的控制界面并支持在图像帧中嵌入数据。

摄像头控制接口（CCI）是I^2C协议的一个子集，包括I^2C规范中规定的I^2C从设备必备功能的最小组合。因此，符合CCI规范的发送器也可以连接到系统I^2C总线。重要的是，I^2C主机不会试图使用CCI主机和CCI从机所不支持的I^2C功能。

数据传输协议层由多个层组成，每个层履行不同的职责。CSI-2协议可使用主处理器上的单个接口开启多个数据流。协议层指定如何对多个数据流进行标记和交错，以便能够正确地重构每个数据流。图28显示了在发送器（图像传感器）和接收器（主处理器）之间的组成数据传输系统的各个层。

- **像素/字节打包/解包层。**CSI-2规范支持像素格式为每像素6位到24位的图像应用。在发送器中，该层将应用层的像素打包成字节，再将数据发送到底层协议层。在接收器中，该层将底层协议层的字节解压为像素，再将数据发送到应用层。此层不会改变每像素8位的传输数据。
- **底层协议**（LLP）包括建立位级和字节级同步的方法，用于在传输开始（SoT）和传输结束（EoT）事件之间传输串行数据，并将数据传递到下一层。LLP的最小数据粒度为1字节。LLP还包括字节内位值解释的赋值，如Endian的赋值。
- **通道管理。**CSI-2具有通道可扩展性，可提高性能。根据应用的带宽要求，数据通道的数量可以是一个、两个、三个或四个。接口的发送端将输出数据流中的字节分配（分配器功能）到一个或多个通道。在接收端，接口从通道收集字节并将其合并（合并功能）为重新组合的数据流。数据流恢复原始的数据流顺序。

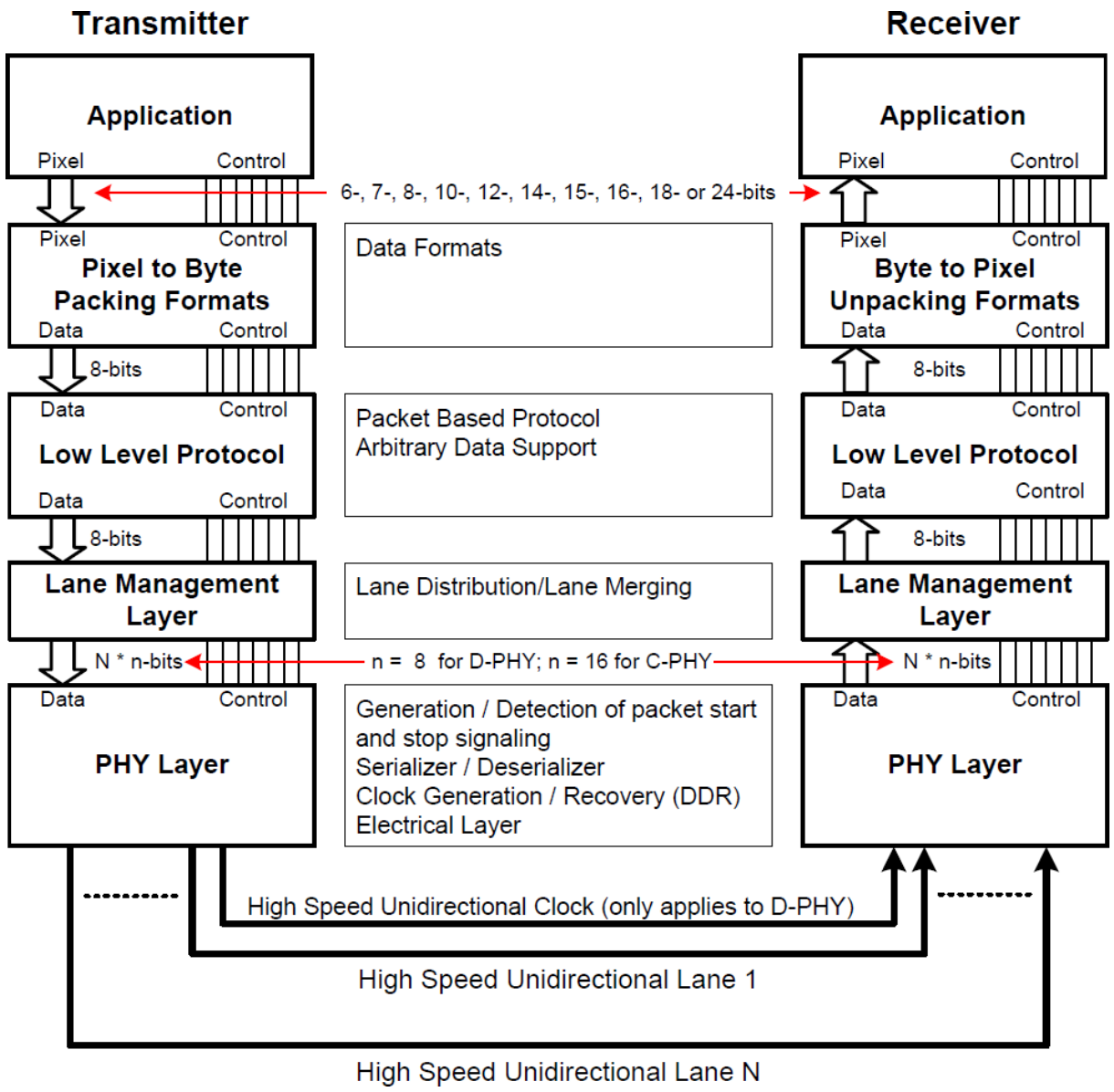


图28. CSI-2层的定义

5.2 图像传感器基础知识

采用CSI-2系统时，对图像传感器有个基本的了解会很有帮助。本文以OmniVision的OV5640图像传感器为例。该传感器为500万像素，支持两个MIPI数据通道，是许多i.MX 6/7/8/RT EVK常用的OV5640MRFL摄像头模块的基础。

图像传感器包含一个光敏单元阵列。每个单元将光转换为原始（RAW）像素。OV5640具有2624列乘1964行的像素阵列，总共5,153,536个像素。并非所有像素都用于图像捕获。如图29所示，有效列为第16–2607列，有效行为第14–1957行，最大可用图像阵列为2592 × 1944 = 5,038,848个有效像素。

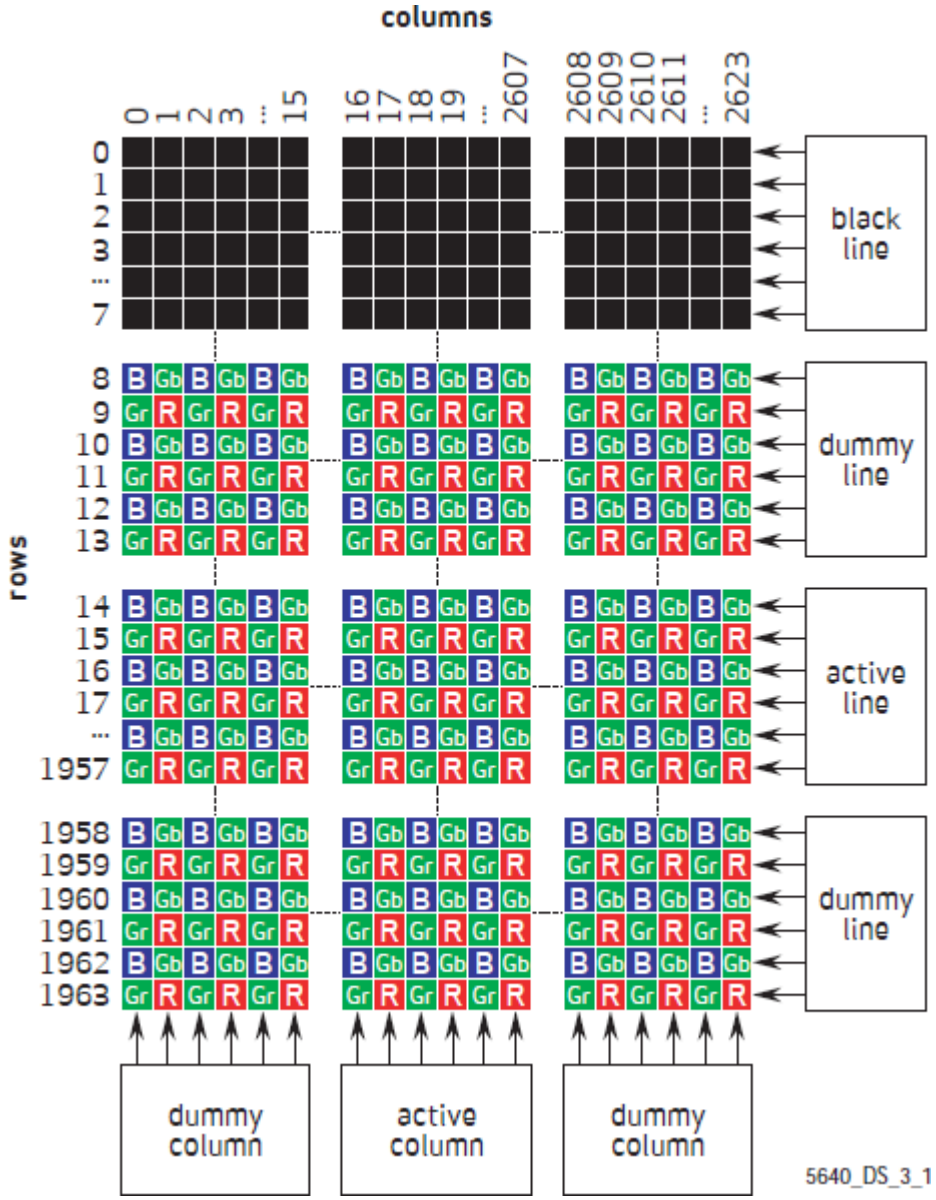


图29. OV5640图像传感器的布局

图像传感器上的ADC一次一个地采样每个像素的电压，并将数字值输出到数据总线上。总线宽度等于ADC分辨率位深。这种未处理的数字值称为原始（RAW）格式。OV5640有一个10位ADC和一个10位宽的数据总线，因此输出采用本地RAW10格式。图像传感器的分辨率范围为6位到24位不等。CSI-2规范仅支持最大14位的RAW位深。ADC读取一整行每个像素的值，移动到下一行，并重复这一过程，直到读取了整个传感器像素阵列，或称之为帧。这个过程听起来很熟悉——与DSI中讲解的光栅扫描显示方法相同。

一帧RAW图像数据会经过一系列的后处理步骤，如去马赛克、颜色校正和白平衡。以前，后处理步骤被认为是有用的或赏心悦目的。这些算法要么在图像传感器上处理，要么在采用ISP进行后处理的期间处理。本文档不涵盖这些主题，它们不会影响带宽、数据通道和时钟等系统规范。本文档不赘述各种图像系统的细节，例如滚动与机械快门的对比、闪光同步和曝光等。

5.2.1 图像传感器视频时序

大多数CMOS图像传感器是可编程的，以生成单个图像（如静态摄像头）或视频流。本文档着重讲述视频流。大多数图像传感器发出的视频流遵循在DSI章节描述的相同视频格式。

像素时钟时间周期设置了每个像素的ADC转换的时间间隔。它是HSYNC和VSYNC信号的根时钟。HSYNC是每一行的开始信号，VSYNC是每一个新帧的开始信号。

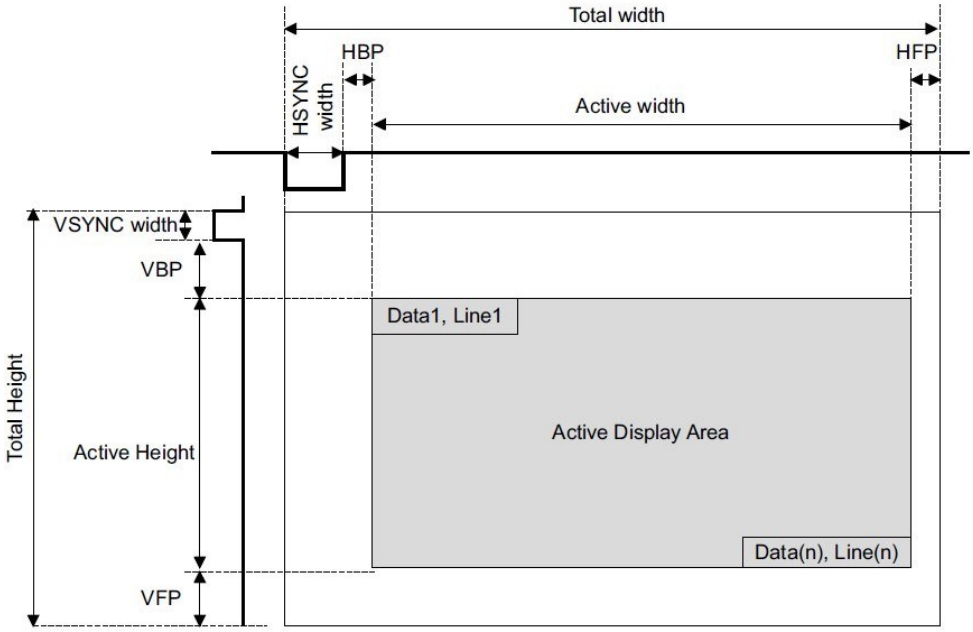


图30. 图像传感器视频时序

5.3 CSI-2视频时序

CSI-2底层协议将像素数据、VSYNC和HSYNC脉冲转换为一系列数据包。数据包通过物理D-PHY接口发送，同时保持关键时序参数之间的关系。图31显示了使用RGB888像素格式的视频流的基本CSI-2帧结构。

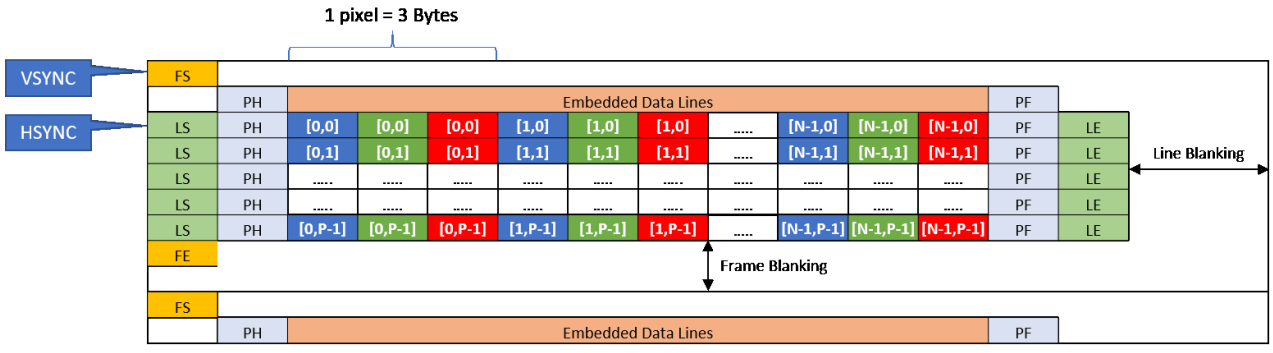
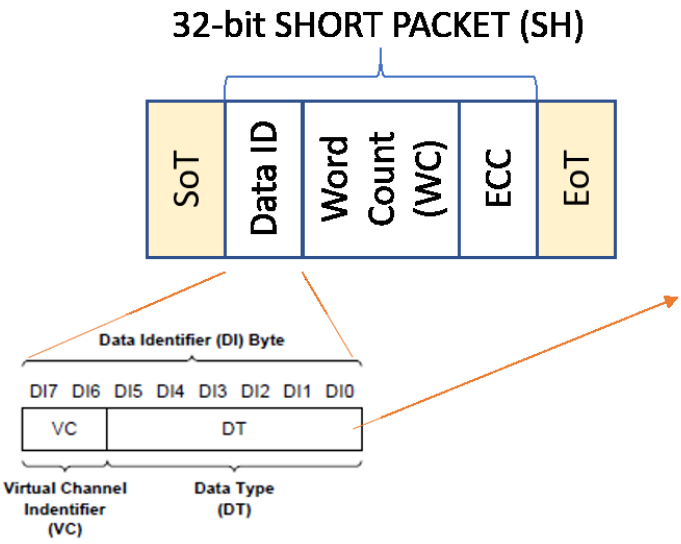


图31. CSI-2帧结构

V_{sync}信号使用帧开始（FS）数据包来传送。H_{sync}信号使用行开始（LS）数据包来传送。

5.3.1 数据包详细信息

在传输开始（SOT）过程结束之后，会发送一个短数据包，告诉接收器下一步是什么：同步信号、数据、虚拟通道信息还是用户定义的数据。图32显示了各种数据包解码选项。



Data Type	Description
0x00	Frame Start Code
0x01	Frame End Code
0x02	Line Start Code (Optional)
0x03	Line End Code (Optional)
0x04 to 0x07	Reserved
0x08 to 0x0F	Generic Short Packet Data Types
0x10 to 0x17	Generic Long Packet Data Types
0x18 to 0x1F	YUV Data
0x20 to 0x27	RGB Data
0x28 to 0x2F	RAW Data
0x30 to 0x37	User Defined Byte-based Data
0x38 to 0x3F	Reserved

图32. 短数据包解码器

图33显示了组成HS包和LP数据包的数据包流是如何定义图像帧开始、行开始和数据的。

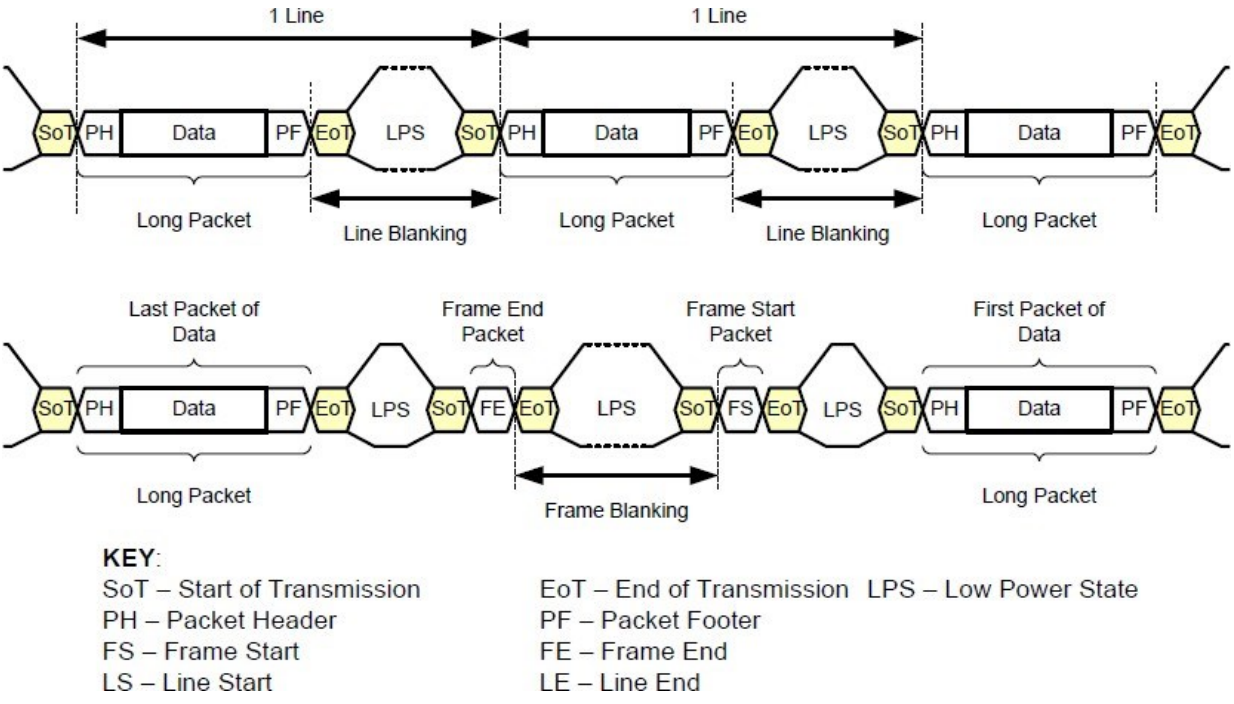


图33. 行和帧消隐定义

5.4 像素/字节打包

旧的图像传感器（在CSI-2标准发布之前）使用并行总线将ADC数据输出到SOC或专用图像处理器。10位宽的数据总线加上行、帧和时钟信号，至少有13个信号宽。闪存或其他功能需要额外的引脚。发送器中的CSI-2硬件模块从并行总线获取图像数据，并将其打包成串行字节流，该串行字节流可以通过DPHY通道传输。

CSI-2标准支持各种像素格式和位长。8位ADC的RAW输出整齐地打包成每字节一个像素。图34显示了一个640位宽图像行的数据包流。

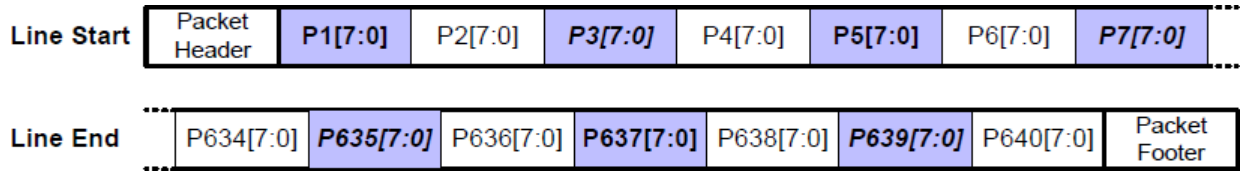


图34. RAW8数据包传输

此RAW8像素/字节打包示例是CSI-2所涵盖的所有各种图像数据标准中最简单、最容易理解的格式。每个数据字节包含一个像素的数值。

OV5640的RAW10格式具有10位的像素值，但CSI-2标准规定最小数据宽度为1字节。由于我们无法传输部分像素数据，CSI-2基于所用像素格式规定了各种像素封装标准。10位RAW像素数据的传输是用5个字节封装4个10位像素值完成的。在这种情况下，最小数据包长度总是5个字节。如果要转换的行的像素宽度不能被5整除，多余的像素将被零填充。图35显示了使用RAW10格式的640位宽图像行的数据包流。

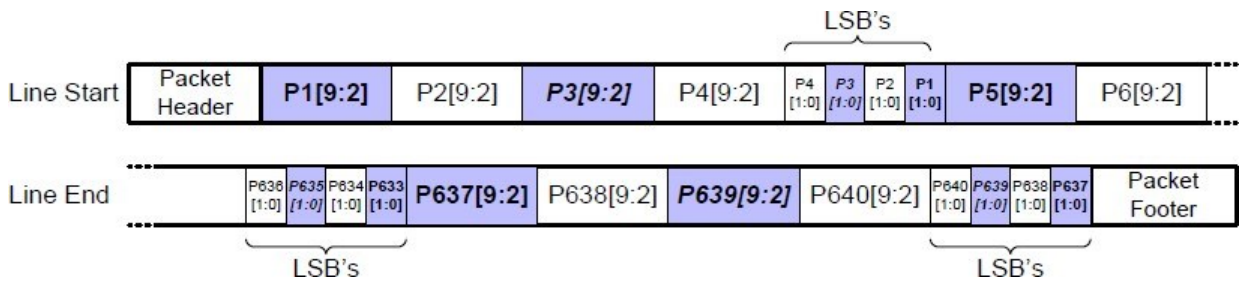


图35. RAW10数据包的传输

表18列出了对于所支持的格式，每像素的位数（bpp）和最小数据包大小（字节）。它们是给定格式的最小数据长度的大小。例如，即使RAW6格式仅使用6位数据来描述每个像素，也必须在最小长度为3字节的数据包中发送至少4个像素。

表18. CSI-2支持的格的式的数据包的数据大小限制

数据格式	每像素位数	每数据包像素数（最小）	数据包长度（字节）	备注
YUV420 8位（旧式）	12	2	3	
YUV420 8位	12	2	2/4	数据包长度 — 偶数/奇数行
YUV420 10位	15	4	5/10	数据包长度 — 偶数/奇数行
YUV422 8位	16	2	4	

表格在下一页继续...

表18. CSI-2支持的格式的数据包的数据大小限制（续）

数据格式	每像素位数	每数据包像素数（最小）	数据包长度（字节）	备注
YUV422 10位	20	2	5	YCbCr 4:2:2
RGB888	24	1	3	
RGB666	18	4	9	
RGB565	16	1	2	
RGB555	15	1	2	在绿色组件的LSB后添加的填充位。
RGB444	12	1	2	在三个颜色组件的LSB后添加的填充位。
RAW6	6	4	3	
RAW7	7	8	7	
RAW8	8	1	1	
RAW10	10	4	5	
RAW12	12	2	3	
RAW14	12	4	7	

5.5 带宽计算

要设置或评估图像捕获系统，必须计算系统带宽，以确保各种时钟支持所需的系统。所有图像系统设计的基础是像素时钟，可以使用公式13计算：

$$\text{像素时钟（Hz）} = \text{HTOT} * \text{VTOT} * \text{FPS}$$

公式 13.

这里：

- H_{TOT} = 总行宽=有效像素 + 水平消隐（像素）
- V_{TOT} = 总帧高度 = 有效行 + 垂直消隐（线）
- FPS = 每秒帧数

总系统带宽等于像素时钟乘以每像素的位数。可以使用公式14计算：

$$\text{带宽（bps）} = \text{像素时钟} * \text{每像素位数}$$

公式 14.

输出格式决定了每个像素的位数。有关CSI-2支持的各种像素格式的位深，请参见表18。

每个通道的数据速率是带宽除以系统设计中的数据通道数。CSI-2规范允许的最小通道数为1，最大通道数为4。

每通道数据速率（bps）= 带宽/数据通道数

公式 15.

MIPI接口为双数据速率，因此D-PHY HS输出时钟频率是每通道数据速率的一半：

MIPI D−PHY时钟速率（Hz）= 每通道数据速率/2

公式 16.

5.5.1 OV5640带宽示例

有关时序信息的示例，请参见OV5640数据手册。图36显示了V_{sync}、H_{sync}和有效数据（HREF）信号之间的关系。该数据手册使用像素时钟作为所有一切的测量单位。必要时转换成行。

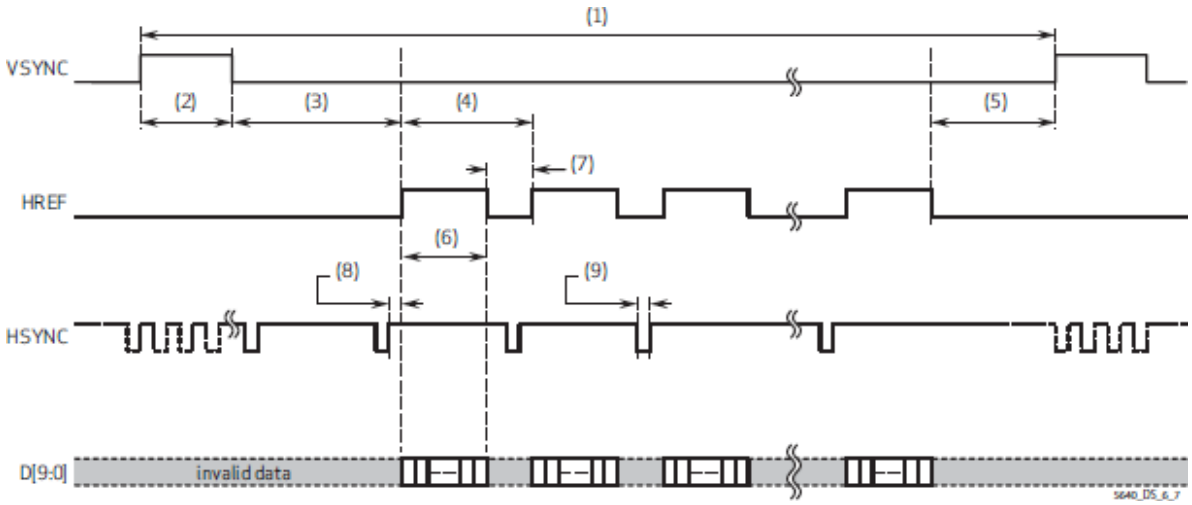


图36. OV5640时序

表19. OV5640的时序值

模式	1	2	3	4	5	6	7	8	9
		V_{sync}	垂直后廊		垂直前廊	有效像素	水平消隐		
5 Mpix (2592 × 1944)	5596992	5688	48276	2844	14544	2592	252	0	252
720 p (1280 × 720)	5596992	5688	3530644	2844	14544	1280	1564	0	1564

OV5640数据手册使用2592像素宽和1944行高的全分辨率，给出了252像素时钟的H_{sync}脉冲宽度。

- H_{TOT} = 2592 + 252 = 2844像素

V_{sync}脉冲被定义为5688个像素时钟，但5688/2844 = 2行。垂直前廊和后廊的时序为22行。如果减去一个H_{sync}脉冲，就可以得到总垂直边廊的时钟数。

- Vbp + Vfp – Hsync = 48276 + 14544 – 252 = 62,568像素时钟
- 62,568/2844 = 22行
- V_{TOT} = 1944 + 2 + 22 = 1968

最大分辨率的帧速率为15FPS。默认输出格式为YUV422（8位），其像素位深为16 bpp。根据这些值，我们可以计算带宽：

- 像素时钟=2844*1968*15=83,954,880 Hz
- 带宽=83,954,880*16=1,343,278,080 bps
- 每通道数据速率=1,343,278,080 bps/2=671,639,040 bps
- MIPI位时钟=671,639,040 bps/2=335.9 MHz

图37是运行在335.9 MHz的OV5640摄像头模块MIPI位时钟的波形图。

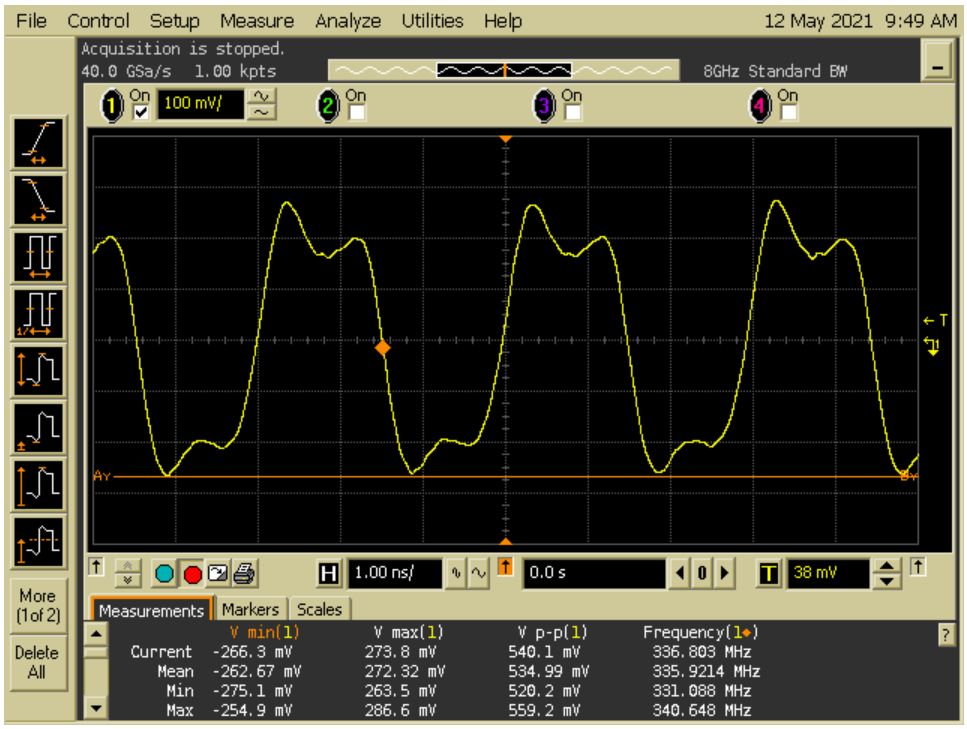


图37. OV5640 MIPI位时钟，15FPS下分辨率为5 Mp

5.6 CSI-2系统时钟配置

上层MIPI CSI-2/D-PHY时钟架构在所有i.MX RT和i.MX 8系列芯片中都是通用的。信号和寄存器名称不同，但功能相同。图38是时钟系统和数据接口的上层概览。

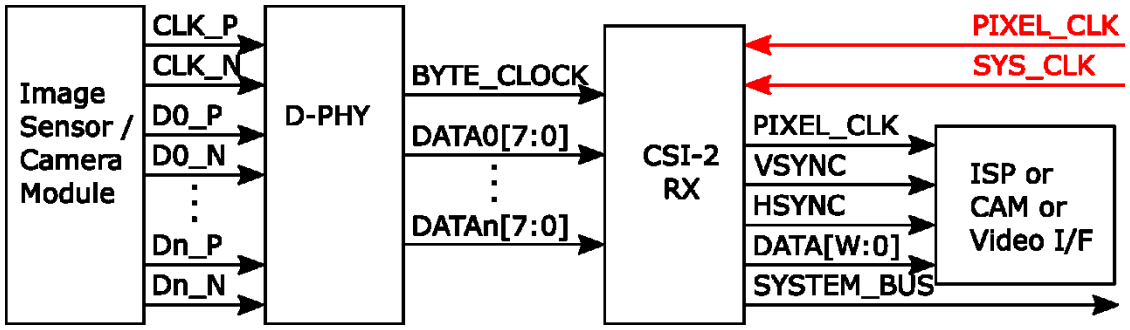


图38. MIPI CSI-/D-PHY时钟结构

时序关系和要求可能令人困惑，但如果记住一条规则就很简单：

- 进入部件（来自图像传感器）的数据带宽必须与CSI-2模块发出的数据带宽平衡。

5.6.1 字节时钟

D-PHY自动生成图38所示的字节时钟（BYTE_CLOCK）。BYTE_CLOCK始终是HS位时钟速度的1/8（也是每通道的数据速率）。

5.6.2 像素时钟

SOC生成像素时钟（`PIXEL_CLK`）。必须设置`PIXEL_CLK`，使CSI-2模块发出的图像带宽大于或等于进入该模块的图像带宽。28FDSOI标准的部件将该时钟称为用户接口时钟或`CLK_UI`。16FF标准的部件将该时钟称为`I_ACLK`或`ACLK`。

5.6.3 系统时钟

SOC生成图38中的系统时钟（`SYS_CLK`）。`SYS_CLK`用于CSI-2控制接口的命令时钟。28FDSOI标准的部件将该时钟称为CLK。16FF标准的部件将其称为`I_PCLK`或APB时钟。

6 CSI-2示例

i.MX 8和i.MX RT系列基于底层工艺节点使用两个不同的CSI-2/D-PHY模块。有关工艺节点和芯片差异的信息，请参见i.MX 8M和i.MX RT芯片的特性/差异。

6.1 RT1170摄像头示例

i.MX RT1170 EVK发运时随附一个摄像头模块，该模块使用OmniVision的OV5640图像传感器。图39显示了正确安装的摄像头模块，其720×1280 LED面板上显示的图像在i.MX RT1170 DSI示例中有描述。

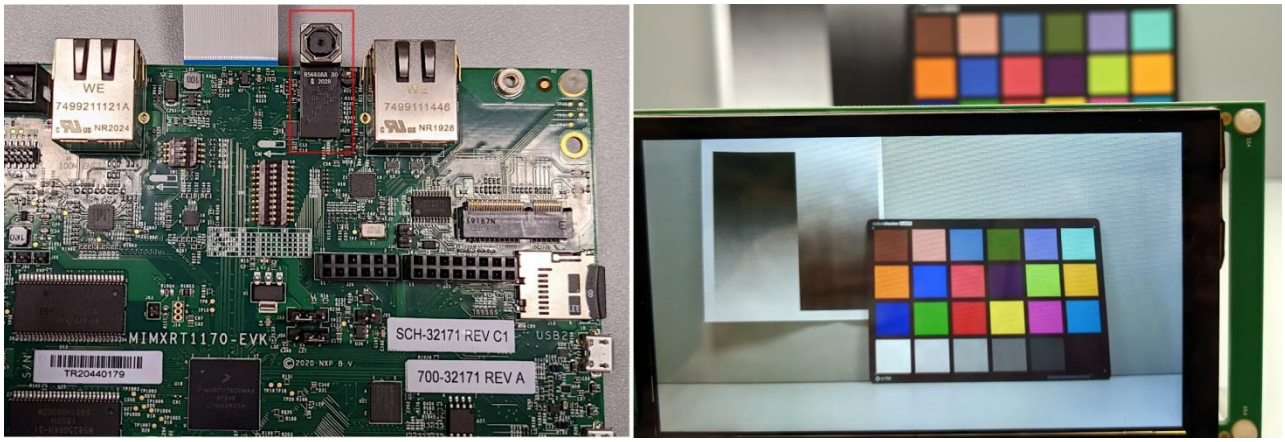


图39. 安装摄像头的i.MX RT1170

在本例中，我们将检查SDK中包含的`CSI_MIPI_RGB_CM7`演示过程。在本例中，摄像头输出像素格式为RGB565。`MIPI_CSI-2`在内部将其转换为RGB888并发送给CSI。换句话说，CSI输入数据总线宽度为24位。CSI将帧保存为32位格式的XRGB8888。PXP用于将XRGB88转换为RGB565并输出到LCD面板。

6.1.1 初始系统设计

首先要做的是计算OV5640所需的图像带宽。在本例中，有效图像大小为1280×720，帧速率为30 FPS。

H_{sync}是612像素，使得H_{TOT} = 1892。

V_{sync}是2行，Vfp+Vbp = 18。因此，V_{TOT} = 720+18+2 = 740行。

使用这些值和30 FPS的帧速率，我们可以计算像素时钟：

- 像素时钟 = 1892*740*30 = 42,002,400 Hz

RGB565格式需要16 bpp，因此带宽为：

- 带宽 = 42,002,400 Hz*16 bpp = 672,038,400 bps

该系统使用2通道MIPI接口，因此每通道的数据速率为：

- DRPL = 672,038,400 bps / 2 = 336,019,200 bps

MIPI D-PHY HS时钟必须为：

- MIPI HS时钟 = 336,019,200 / 2 = 168,009,600 Hz

6.1.2 CSI-2 RX时钟设置

RT1170 CSI-2 RX模块需要为摄像头操作设置三个时钟：

- 系统时钟（CLK）
- 像素时钟（CLK_UI）
- 脱离模式时钟（CLK_ESC）

6.1.2.1 系统时钟

在28FDSOI CSI-2模块中，系统时钟简称为CLK。系统时钟必须等于或快于字节时钟。本例中的字节时钟可以计算为：

Byte_clock = DRPL / 8 = 336,019,200 / 8 = 42,002,400 Hz

CLOCK_ROOT73控制CSI2_CLK_ROOT。示例代码的检查显示以下寄存器设置：

- CLOCK_ROOT73_CONTROL[MUX] = 5
- CLOCK_ROOT73_CONTROL[DIV] = 8

多路开关设置为5，将根时钟设置为以480MHz运行的SYS_PLL3_CLK。DIV设置为8时，时钟频率为480 MHz/8=60 MHz。

6.1.2.2 像素时钟

CLOCK_ROOT75控制CSI2_UI_CLK_ROOT。

- CLOCK_ROOT75_CONTROL[MUX] = 5
- CLOCK_ROOT75_CONTROL[DIV] = 8

多路开关设置为5，将根时钟设置为以480MHz运行的SYS_PLL3_CLK。DIV设置为8时，时钟频率为480 MHz/8=60 MHz。

6.1.2.3 脱离模式时钟

脱离模式时钟必须在60–80 MHz之间。CLOCK_ROOT74控制CSI2_ESC_CLK_ROOT。

- CLOCK_ROOT74_CONTROL[MUX] = 5
- CLOCK_ROOT74_CONTROL[DIV] = 8

多路开关设置为5，将根时钟设置为以480MHz运行的SYS_PLL3_CLK。DIV设置为8时，时钟频率为480/8=60 MHz。

6.1.3 带宽平衡

使用这些时钟设置，我们可以检查输入和输出带宽的关系。输出带宽必须大于或等于输入带宽。

- 输入带宽 = DRPL * 通道数

  公式 17.

- 输出带宽 =（像素时钟）*（图像格式bpp）*（每个时钟周期的像素数）

  公式 18.

- 输入带宽 = 336,019,200 bps * 2 = 672,038,400 bps
- 输出带宽 = 60 MHz * 16 bpp * 1 = 960 Mbps

由于输出带宽（960 Mbps）大于等于输入带宽（672 Mbps)，因此不会丢失图像数据。

6.1.4 摄像头时钟设置

RT1170 EVK上的24 MHz晶振提供OV5640的参考时钟（XVCLK）。根据初始系统设计的计算，我们需要42 MHz的像素时钟和168 MHz的MIPI位时钟。

表20. OV5640时钟设置寄存器

寄存器地址	寄存器名称	描述	编程值
0x3034	SC PLL CONTRL0	位[3:0]：MIPI位模式	0x1A

表格在下一页继续...

表20. OV5640时钟设置寄存器（续）

寄存器地址	寄存器名称	描述	编程值
		0x8：8位模式 0xA：10位模式	
0x3035	SC PLL CONTRL1	位[7:4]：系统时钟分频器——减慢所有时钟（SDIV0） 位[3:0]：MIPI PCLK的分频器（MIPI_DIV）	0x21
0x3036	SC PLL CONTRL2	锁相环倍数（4~252）：可以是4~127的任意整数，只能是128~252的偶数	0x54
0x3037	SC PLL CONTRL3	位[4]：PLL根分频器（PLL_RDIV） 0：旁路 1：除以2 位[3:0]：锁相环预分频器：1、2、3、4、6、8	0x13
0x4837	PCLK Period	像素时钟周期，pclk_div=1，1位小数	0x0A
0x3108	SYSTEM ROOT DIVIDER	位[5:4]：PCLK根分频器（PCLK_DIV） 00：PCLK = pll_clki 01：PCLK = pll_clki/2 10：PCLK = pll_clki/4 11：PCLK = pll_clki/8	0x16

系统时钟基于所需的系统带宽，可计算如下：

- SYSCLK = XVCLK/PLL预分频数 * PLL倍数 = 24/3*84 = 672 MHz

PLL预分频数的映射没有文档描述，但3似乎是常用的。

MIPI位时钟使用以下公式从系统时钟导出：

- MIPI_BIT_CLK = SYSCLK / MIPI_DIV / 2 = 672 / 2 / 2 = 168 MHz

像素时钟也使用以下公式从系统时钟导出：

- PIXEL_CLK = 系统时钟/（SDIV0*PLL_RDIV*BIT_DIV*PCLK_DIV*P_DIV）=672/（2*2*2*1）=42 MHz

图40显示了正确设置为1280×720×30 FPS和16 bpp后高速模式下的MIPI位时钟。如先前计算的，频率为168 MHz。

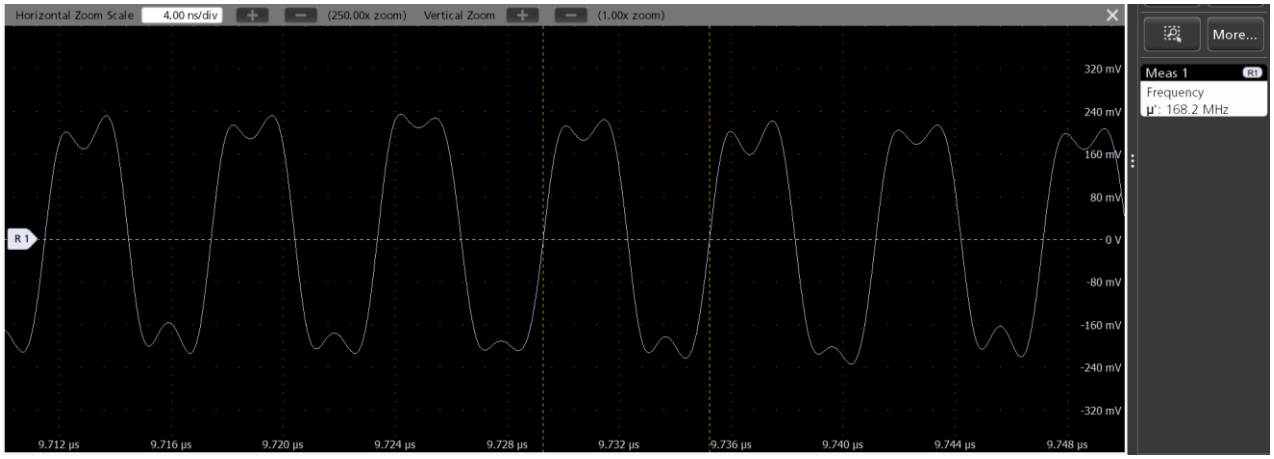


图40. 1280 x 720@30 FPS、16 bpp-168 MHz的OV5640 MIPI时钟

6.1.5 D-PHY设置

唯一需要配置的D-PHY参数是$T_{HS-SETTLE}$。

在本例中，`RxClkInEsc`和`CLK_UI`（像素时钟）都是60 MHz。两者的时间周期均为1/60 MHz=16.67 ns。

D-PHY规范要求$T_{HS-SETTLE}$在85 ns + 6 * UI到145 ns + 10 * UI的范围内。计算结果的范围为185 ns–311.67 ns。

对于RT1170，编程$T_{HS-SETTLE}$的公式为：

$$T\_HS\_SETTLE = (PRG\_RXHS\_SETTRE+1) * RxClkInEsc的Tperiod$$

公式 19.

CSI-2 RX模块的驱动程序代码设置PRG_RXHS_SETTLE = 0x11 = 17。

$T_{HS-SETTLE}$= (17+1) * 16.67 ns = 300 ns

`RxClkInEsc`的计算误差高达1个周期。选择远离边界的编程值。

6.1.6 OV5640 MIPI波形

使用上一节中的设置，我们可以通过查看时钟和数据通道的示波图来验证部分时序。

图41中的示波图显示了从LP数据传输转换为HS数据传输时，时钟（以差分方式显示）和数据0通道（P和N为单端显示）之间的关系。

300ns的THS-SETTLE周期被着重显示，然后是传输开始（SoT）报头和帧开始（FS）数据包。

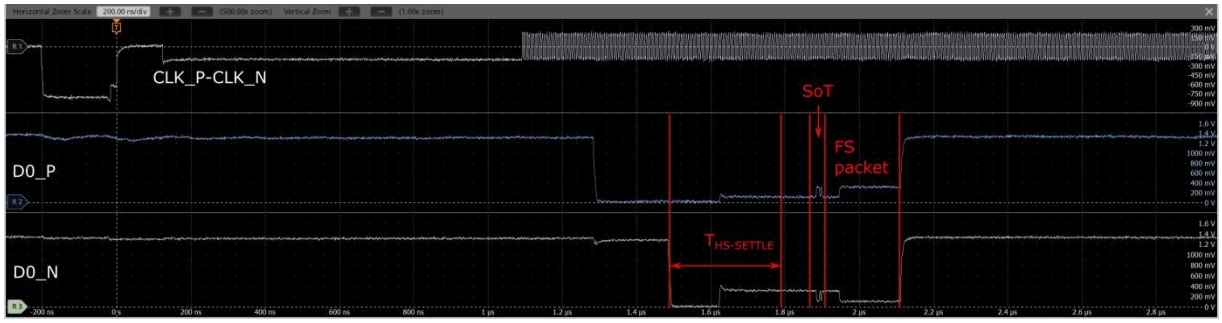


图41. OV5640启动

由于在扫描行的末端数据通道从HS转换到LP一次，我们可以很容易地使用示波器来检查时序。图42中的示波图显示了新帧的开始和前13个有效图像行。单行（tL）所需的时间可以用帧速率除以VTOT来计算。

tL = 1 / 帧速率 / VTOT = (1/30fps) / 740行 = 45.045 us

V_{sync} + VBP = 2 + 7 = 9行。因此，从帧开始到有效像素数据的总时间为：

*9行 * 45.045us = 405 us*

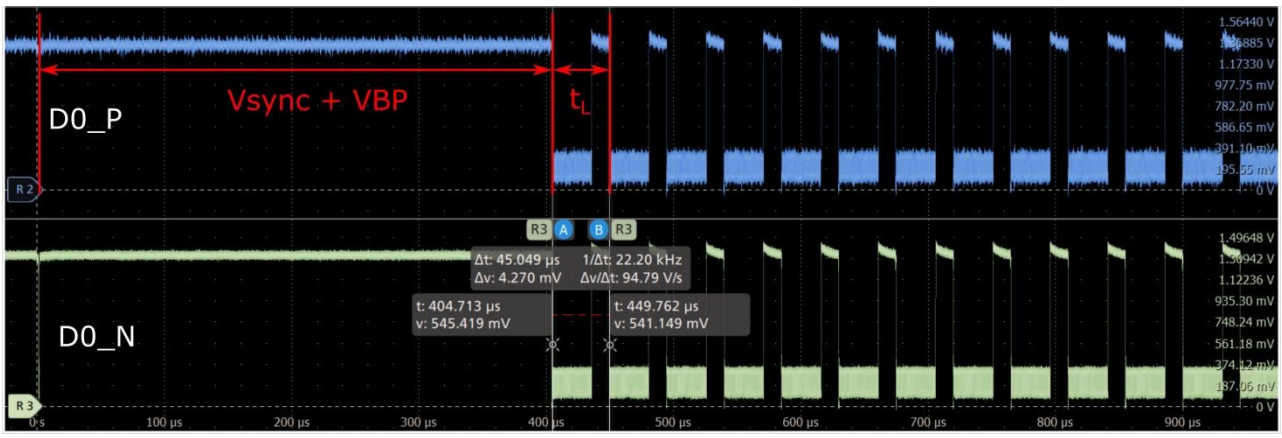


图42. OV5640帧开始时序

7 i.MX 8M和i.MX RT芯片的特性/差异

表21. 特性差异

型号	DSI			CSI-2			制程
	通道数	最大通道速度（Gbps）	最大总带宽（Gbps）	通道数	最大通道速度（Gbps）	最大总带宽（Gbps）	
i.MX RT1170	2	0.8	1.6	2	1.5	3	28FDSOI
i.MX RT500[1]	2	0.8951	1.79	无	无	无	28FDSOI

表格在下一页继续...

表21. 特性差异（续）

型号	DSI			CSI-2			制程
	通道数	最大通道速度（Gbps）	最大总带宽（Gbps）	通道数	最大通道速度（Gbps）	最大总带宽（Gbps）	
i.MX 8	4	1.5	6	4	1.5	6	28FDSOI
i.MX 8M	4	1.5	6	4 (x2)	1.5	6	28FDSOI
i.MX 8M Mini	4	1.5	6	4	1.5	6	16FF
i.MX 8M Nano	4	1.5	6	4	1.5	6	16FF
i.MX 8M Plus	4	1.5	6	4 (x2)	1.5	6	16FF
i.MX 8ULP	4	1.5	6	2	1.5	3	28FDSOI
i.MX 8ULP-CS	4	1.5	6	2	1.5	3	28FDSOI
i.MX 8Q/D/X	4 (x2)	1.05	4.2	4	1.5	6	28FDSOI

1. D-PHY不支持PLL或ULPS。

8 调试

本节介绍了常见的启用问题，并提供了调试技巧。

8.1 DSI启用检查清单

1. 检查所有电源电压。
2. 确保I^2C通信正常：
    a. 验证发送和接收数据的能力。可以使用Linux I^2C命令，如`i2cdetect`、`i2cdump`、`i2cget`、`i2cset`。
3. 确保显示初始化寄存器是正确的。
4. 检查MIPI时钟：
    b. MIPI HS位时钟
    c. 像素时钟
    d. 脱离模式时钟
5. 确保显示器和SOC视频模式是同步的。例如，显示器和SOC都选择突发视频模式。
6. 检查显示驱动器中是否正确设置了消隐参数：H_{sync}、HBP、HFP、V_{sync}、VBP、VFP。
7. 检查MIPI低功耗通信：频率、IO电源电压。
8. 检查MIPI高速通信：频率和差分电压。
9. 检查MIPI错误寄存器：`ErrSotHS`和`ErrSot_Sync_HS`。

8.2 DSI常见问题

8.2.1 不连续时钟

有些显示器不需要专用的外部晶体/时钟。它们使用MIPI时钟作为系统时钟。这些显示器需要SOC的连续时钟模式来为显示器提供适当的时钟，以执行初始化和/或运行时功能。

8.2.2 缺少周期性低功耗转换

常见的显示故障是由BLLP操作的DSI源配置引起的。为了使PHY同步，主处理器周期性地结束HS传输并将数据通道驱动到LP状态。这种转换每帧至少发生一次。一些常见DSI驱动程序示例在视频流的任何BLLP期间均默认不启用LP模式。

8.2.3 错误的DSI数据包时序

从D-PHY物理层来看，大多数标准DSI接口都确保兼容MIPI。与DSI数据包顺序/时序相关的问题发生在驱动程序开发阶段，特别是在为自定义视频解决方案实现非标准视频时序或时钟速率时。这意味着视频数据包（尤其是同步数据包）的接收时序对于保持系统正常工作至关重要。

8.2.4 $T_{HS-SKIP}$配置

MIPI D-PHY接收器规范要求信宿设备在高速数据包传输结束时忽略DSI数据通道上的活动，以便重新进入低功耗状态（LP-11），其目的是在传输结束（EoT）序列中屏蔽转换的影响。如果$T_{HS-SKIP}$时序参数配置错误，可能会导致DSI视频流数据错误。

8.2.5 传输结束数据包（EoTp）

兼容MIPI DSI v1.0规范和更高版本的DSI设备需要在所有HS数据传输之后生成传输结束数据包（EoTp）。EoTp的主要目标是在从HS模式转换到LP模式期间增强DSI接口的鲁棒性，以便即使在非最佳信号条件的情况下，接收器也可以清晰地检测到HS传输的结束。为了支持DSI外围设备之间的向后兼容性和互操作性，这一标准要求发送器和接收器设备具有选择采用或不采用EoTp功能的能力。这个EoTp的插入是可选的。如果正在使用的显示器需要这一功能，请在16FF和28FDSOI的DSI模块上启用。

8.3 CSI-2启用检查清单

1. 检查所有电源电压。
2. 确保I^2C通信正常：
   - 验证发送和接收数据的能力。可以使用Linux I^2C命令，如`i2cdetect`、`i2cdump`、`i2cget`、`i2cset`。
3. 确保传感器初始化寄存器是正确的。
4. 检查摄像头和SOC MIPI时钟：
   a. 摄像头MIPI HS位时钟、像素时钟和脱离模式时钟。
   b. SOC像素时钟和脱离模式时钟。
5. 检查MIPI低功耗通信、低频和非差分（示波器或MIPI分析仪）。
6. 检查MIPI高速通信、高频和差分（MIPI分析仪）。
7. 检查MIPI状态寄存器，如：
   a. ECC和CRC错误状态寄存器
   b. IRQ状态寄存器
   c. `ErrSot` HS状态寄存器

d. ErrSotSync 状态寄存器

e. ErrEsc状态寄存器

f. ErrSyncEsc状态寄存器

g. ErrControl状态寄存器

8.4 CSI-2常见问题

8.4.1 像素格式错误

MIPI CSI-2规范支持多种图像像素格式。完整列表见表18。但是在选定的具体格式中，应用层和图像传感器可能会有轻微的错位。例如，图像传感器可能使用RGB888格式发送数据，但是此图像传感器可能用RGB、BGR、GRB等6种不同的方式发送像素数据。如果应用以为是BGR，而图像传感器使用GRB发送数据，则没有数据传输错误，但接收到的图像的颜色出现错误。

9 修订历史

版本号	日期	描述
0	2022年3月21日	初版发布

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

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.

Airfast — is a trademark of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Cadence — the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.

ColdFire+ — is a trademark of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

EdgeScale — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

FeliCa — is a trademark of Sony Corporation.

Freescale — is a trademark of NXP B.V.

HITAG — is a trademark of NXP B.V.

ICODE and I-CODE — are trademarks of NXP B.V.

Immersiv3D — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Layerscape — is a trademark of NXP B.V.

Mantis — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MOBILEGT — is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

Processor Expert — is a trademark of NXP B.V.

QorIQ — is a trademark of NXP B.V.

SafeAssure — is a trademark of NXP B.V.

SafeAssure — logo is a trademark of NXP B.V.

StarCore — is a trademark of NXP B.V.

Synopsys — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

Tower — is a trademark of NXP B.V.

UCODE — is a trademark of NXP B.V.

VortiQa — is a trademark of NXP B.V.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 March 2022
Document identifier: AN13573