

## STM32F4系列和STM32F7系列微控制器上的DSI主机

## 前言

嵌入式设备中对类似于智能手机的高级图形用户界面的需求不断增长，为嵌入式系统设计者带来了极大挑战。到目前为止，SPI、并口和RGB接口已被广泛应用于在MCU和显示屏之间建立连接。

由于对分辨率和刷新频率的要求越来越高，需要更多数量的引脚（对于16.7M色彩显示屏，最多为28个引脚）和更高的像素时钟频率。这些需求提高了对MCU侧的引脚数量要求和总体PCB复杂性和成本，这要归因于板尺寸、路由复杂性和时钟与数据之间的偏差问题。

为了解决这些挑战，意法半导体在市场上第一个提供了带有MIPI DSI主机的MCU产品（参见表1）。这些新的具有DSI主机的STM32产品实现了一种更有效的连接到显示屏的方法。MIPI-DSI是一种高速、引脚数量少的串行接口，用于原本针对移动产业的显示屏。由于在移动手机和平板电脑中的广泛应用，DSI接口越来越普及，这推动了DSI显示屏成本的下降，并使之对其他消费市场产生了吸引力。

STM32 MIPI-DSI主机大幅减少了设备的引脚数量，可与如今市场上随处可见的DSI显示屏实现轻松连接。得益于它的引脚数量少和低功耗特性，DSI主机是连接显示屏的最有效方式，特别是对于有严格的尺寸和功耗限制的设备，例如可穿戴设备。

本应用笔记描述了STM32F4系列和STM32F7系列微控制器的DSI主机接口，重点展示了DSI主机的不同操作模式，并提供了根据应用需要选择最佳工作模式的指南。还提供了关于如何根据操作模式配置DSI主机的实例。

## 相关文档

本应用笔记必须同以下文档（在[www.st.com](http://www.st.com)上获取）一起阅读：

- STM32F76xxx和STM32F77xxx基于32位MCU（RM0410）的高级ARM®
- STM32F469xx和STM32F479xx基于32位MCU（RM0386）的高级ARM®
- 相关的STM32F469/479、STM32F7x8和STM32F7x9数据手册

表1. 适用产品

类型	产品线
微控制器	STM32F469/479, STM32F7x8, STM32F7x9

# 目录

1	标准和参考文献 .....	10
2	概述 .....	11
2.1	显示屏连接 .....	11
2.2	MIPI显示屏规格标准 .....	13
2.3	STM32产品支持的显示接口 .....	13
2.4	不同STM32微控制器的DSI主机可用性 .....	16
2.5	DSI主机的优势 .....	17
2.6	智能架构中的DSI主机 .....	17
3	DSI简介 .....	19
3.1	DSI工作模式 .....	19
3.1.1	指令模式 .....	19
3.1.2	视频模式 .....	20
3.2	DSI物理层 .....	20
3.2.1	PHY配置 .....	20
3.2.2	PHY信令模式 .....	21
	数据通道状态 .....	22
3.2.3	数据通道工作模式 .....	22
	控制模式 .....	22
	高速传输模式 .....	23
	逸出模式 .....	24
3.2.4	双向通道和总线转向流程 .....	30
3.2.5	时钟通道功耗模式 .....	30
	低功耗模式 .....	30
	高速模式 .....	30
	超低功耗状态 (ULPS) .....	32
3.3	DSI协议 .....	33
3.3.1	数据包结构 .....	33
	长数据包 .....	34
	短数据包 .....	35
	数据标识符字节 .....	35
	数据保护 (ECC和校验和) .....	36
3.3.2	传输结束 (EoT) 数据包 .....	36
3.3.3	数据包传输模式 .....	36

3.3.4	主机至显示屏数据类型 . . . . .	38
	视频模式数据类型 . . . . .	40
	指令模式数据类型 . . . . .	43
3.3.5	显示屏至主机数据类型 . . . . .	47
3.3.6	视频模式接口时序 . . . . .	49
3.3.7	指令模式下的撕裂效应信号 . . . . .	50
<b>4</b>	<b>DSI主机描述 . . . . .</b>	<b>52</b>
4.1	DSI系统级架构 . . . . .	52
	DSI主机构建模块 . . . . .	52
4.2	工作模式 . . . . .	53
4.2.1	视频模式 . . . . .	54
	使用同步脉冲的非突发模式 . . . . .	54
	使用同步事件的非突发模式 . . . . .	58
	突发模式 . . . . .	60
	视频模式比较: . . . . .	62
4.2.2	适配指令模式 . . . . .	63
	适配指令模式下的显示刷新示例: . . . . .	64
	适配指令模式下的部分刷新示例 . . . . .	64
	适配指令模式的优势 . . . . .	66
	撕裂效应管理 . . . . .	67
4.2.3	APB指令模式 . . . . .	71
	何时使用APB指令模式? . . . . .	71
4.3	工作模式选择 . . . . .	72
4.4	DSI中断 . . . . .	72
4.5	低功耗模式 . . . . .	73
<b>5</b>	<b>DSI主机配置 . . . . .</b>	<b>74</b>
5.1	DSI主机全局配置 . . . . .	74
5.1.1	DSI调节器配置 . . . . .	74
5.1.2	时钟配置 . . . . .	74
	DSI PLL配置 . . . . .	75
	TX逸出时钟配置 . . . . .	76
	次级时钟源设置 . . . . .	76
5.1.3	DSI主机PHY参数 . . . . .	77
	通道数 . . . . .	77
	PHY时钟和数字部分控制 . . . . .	77
	时钟通道控制 . . . . .	77
5.1.4	DSI Wrapper PHY参数 . . . . .	79

HS位周期设置 .....	79
5.1.5 协议流控 .....	80
5.1.6 DSI主机LTDC接口配置 .....	81
颜色编码配置 .....	81
视频控制信号极性 .....	82
5.2 DSI工作模式配置 .....	83
5.2.1 通过LTDC接口实现视频模式 .....	83
视频模式选择 .....	83
视频模式下的LP状态 .....	83
PHY转换时间配置 .....	83
LP转换配置 .....	85
LTDC设置 .....	89
DSI主机视频时序 .....	91
DSI时钟设置 .....	93
DSI视频数据包参数 .....	93
视频模式下的指令传输 .....	96
帧回应 .....	103
5.2.2 通过LTDC接口实现适配指令模式 .....	105
DSI指令模式 .....	105
停止等待时序配置 .....	106
指令大小 (CMDSIZE) .....	106
撕裂效应设置 .....	106
刷新模式 .....	107
LTDC设置 .....	107
指令传输模式 .....	108
回应请求 .....	109
<b>6 STM32CubeMX配置示例 .....</b>	<b>111</b>
6.1 DSI主机视频突发模式 .....	111
6.1.1 引脚排列配置 .....	111
在RCC中使能HSE .....	111
以DSI模式使能LTDC .....	111
以视频模式使能DSI主机 .....	112
6.1.2 时钟配置 .....	112
LTDC时钟配置 .....	112
DSI时钟配置 .....	113
6.1.3 LTDC和DSI配置 .....	113
LTDC配置 .....	113
DSI主机配置 .....	116
6.1.4 视频突发模式的生成代码示例 .....	120

---

6.2	使用同步脉冲的DSI主机非突发模式 .....	123
6.2.1	指令配置 .....	123
6.2.2	显示接口配置 .....	123
6.3	DSI主机适配指令模式 .....	124
6.3.1	引脚配置 .....	124
6.3.2	时钟配置 .....	125
	LTDC时钟配置 .....	125
6.3.3	LTDC和DSI配置 .....	125
	LTDC配置 .....	125
	DSI主机配置 .....	127
6.3.4	适配指令模式的生成代码示例 .....	130
<b>7</b>	<b>DSI主机性能 .....</b>	<b>134</b>
7.1	DSI链路最大带宽对LTDC像素时钟的影响 .....	134
7.2	系统限制对LTDC像素时钟的影响 .....	134
7.3	DSI链路带宽估计 .....	134
7.3.1	视频模式 .....	134
7.3.2	适配指令模式 .....	135
<b>8</b>	<b>DSI主机应用示例 .....</b>	<b>137</b>
8.1	小尺寸显示屏驱动示例 .....	137
8.2	大尺寸显示屏驱动示例 .....	137
<b>9</b>	<b>支持的设备 .....</b>	<b>139</b>
<b>10</b>	<b>结论 .....</b>	<b>140</b>
<b>11</b>	<b>版本历史 .....</b>	<b>141</b>

## 表格索引

表1.	适用产品 . . . . .	1
表1.	STM32产品中的显示屏连接 . . . . .	15
表2.	以DSI主机为特色的STM32微控制器 . . . . .	16
表3.	DSI主机相较于其他显示接口的优势 . . . . .	17
表4.	数据通道状态和工作模式 . . . . .	22
表5.	逸出模式指令 . . . . .	24
表6.	主机至显示屏数据类型 . . . . .	38
表7.	DCS指令列表 . . . . .	46
表8.	显示屏至主机数据类型 . . . . .	48
表9.	DSI主机低功耗模式 . . . . .	73
表10.	DSI调节器配置寄存器 . . . . .	74
表11.	DSI PLL配置寄存器 . . . . .	75
表12.	TX逸出时钟配置寄存器 . . . . .	76
表13.	次级时钟源设置寄存器 . . . . .	77
表14.	通道数配置寄存器 . . . . .	77
表15.	PHY时钟和数字部分控制寄存器 . . . . .	77
表16.	时钟通道控制寄存器 . . . . .	78
表17.	HS位周期配置寄存器 . . . . .	79
表18.	协议流控配置寄存器 . . . . .	80
表19.	颜色编码配置寄存器 . . . . .	81
表20.	视频控制信号极性寄存器 . . . . .	82
表21.	视频模式选择寄存器 . . . . .	83
表22.	PHY转换时间配置寄存器 . . . . .	83
表23.	LP转换配置寄存器 . . . . .	85
表24.	显示时间示例 . . . . .	90
表25.	DSI主机视频时序寄存器 . . . . .	91
表26.	DSI视频数据包参数寄存器 . . . . .	93
表27.	指令传输模式寄存器 . . . . .	97
表28.	LP指令数据包大小寄存器 . . . . .	97
表29.	帧回应寄存器 . . . . .	103
表30.	DSI指令模式寄存器 . . . . .	105
表31.	停止等待时序配置寄存器 . . . . .	106
表32.	指令大小寄存器 . . . . .	106
表33.	撕裂效应设置寄存器 . . . . .	107
表34.	刷新模式寄存器 . . . . .	107
表35.	指令传输寄存器 . . . . .	108
表36.	回应请求寄存器 . . . . .	109
表37.	取决于颜色编码和DSI链路速度的最大像素时钟频率 . . . . .	134
表38.	文档版本历史 . . . . .	141
表39.	中文文档版本历史 . . . . .	141

# 图片索引

图1.	具有帧缓冲器和控制器的显示屏架构 . . . . .	12
图2.	没有控制器和帧缓冲器的显示屏架构 . . . . .	12
图3.	DBI接口示例 . . . . .	14
图4.	DPI接口示例 . . . . .	14
图5.	DSI接口示例 . . . . .	15
图6.	具有DSI主机系统架构的STM32 . . . . .	18
图7.	DSI接口概述 . . . . .	19
图8.	DSI主机和显示接口 . . . . .	21
图9.	HS和LP信号电平 . . . . .	22
图10.	基本的高速数据传输 . . . . .	23
图11.	高速数据传输模式 . . . . .	23
图12.	使用两个数据通道的高速传输 . . . . .	24
图13.	使用两个数据通道、字节数为奇数的高速传输 . . . . .	24
图14.	间隔独热编码 . . . . .	25
图15.	使用间隔独热编码通信的示例 . . . . .	26
图16.	LPDT逸出模式序列 . . . . .	26
图17.	LPDT有效负载数据 . . . . .	27
图18.	ULPS逸出模式序列 . . . . .	27
图19.	回应触发表示例 . . . . .	28
图20.	撕裂效应触发表示例 . . . . .	29
图21.	复位触发表示例 . . . . .	29
图22.	总线转向流程 . . . . .	30
图23.	时钟HS进入序列 . . . . .	31
图24.	时钟HS退出序列 . . . . .	31
图25.	HS模式下时钟通道和数据通道的关系 . . . . .	32
图26.	时钟通道ULPS进入序列 . . . . .	33
图27.	时钟通道ULPS退出序列 . . . . .	33
图28.	短和长数据包结构 . . . . .	34
图29.	长数据包示例 . . . . .	34
图30.	数据标识符字节 . . . . .	35
图31.	使用EoT数据包的HS传输 . . . . .	36
图32.	HS模式下使用一个数据通道的短数据包传输 . . . . .	36
图33.	HS模式下使用两个数据通道的短数据包传输 . . . . .	37
图34.	低功耗模式下的短数据包传输 . . . . .	37
图35.	HS模式下使用一个数据通道的长数据包传输 . . . . .	37
图36.	HS模式下使用两个数据通道的长数据包传输 . . . . .	37
图37.	HS模式下使用两个数据通道、有效负载数据具有奇数个字节的长数据包传输 . . . . .	38
图38.	低功耗模式下的长数据包传输 . . . . .	38
图39.	颜色模式视频数据包 . . . . .	40
图40.	关机/开启视频数据包 . . . . .	40
图41.	同步事件数据包 . . . . .	41
图42.	封装像素流, 16位格式, 数据类型 (0x0E) . . . . .	41
图43.	封装像素流, 18位格式, 数据类型= (0x1E) . . . . .	42
图44.	松散封装像素流, 18位格式, 数据类型= (0x2E) . . . . .	42
图45.	封装像素流, 24位格式, 数据类型= (0x3E) . . . . .	42
图46.	通用短写指令 . . . . .	43
图47.	通用长写指令 . . . . .	43
图48.	通用读指令 . . . . .	43

图49.	DCS短写指令 . . . . .	44
图50.	DCS长写指令 . . . . .	44
图51.	DCS读指令 . . . . .	45
图52.	反向通信序列 . . . . .	48
图53.	使用同步脉冲的非突发模式 . . . . .	49
图54.	使用同步事件的非突发模式 . . . . .	50
图55.	突发模式 . . . . .	50
图56.	TE序列 . . . . .	51
图57.	Set_tear_scanline DCS长数据包 . . . . .	51
图58.	DSI构建模块 . . . . .	53
图59.	使用同步脉冲的非突发模式下的帧 . . . . .	54
图60.	消隐或LP的定义 . . . . .	56
图61.	使用同步脉冲的非突发模式下的VACT区域 . . . . .	57
图62.	包含四个块且空包启用的激活行 . . . . .	58
图63.	使用同步事件的非突发模式下的帧 . . . . .	59
图64.	使用同步事件的突发模式下的VACT区域 . . . . .	60
图65.	视频突发模式帧 . . . . .	61
图66.	突发模式下的激活行 . . . . .	62
图67.	视频模式比较 . . . . .	63
图68.	通过WMS/WMC刷新整个屏幕 . . . . .	64
图69.	显示屏部分刷新 . . . . .	65
图70.	采用自动刷新的适配指令模式流程示例 . . . . .	65
图71.	适配指令模式下的显示刷新 . . . . .	66
图72.	撕裂效应请求与响应示例 . . . . .	67
图73.	具有双BTA的Set_tear_scanline DCS指令 . . . . .	68
图74.	来自显示屏的撕裂效应触发信号 . . . . .	69
图75.	通过引脚报告TE . . . . .	70
图76.	通过引脚报告撕裂效应放大图 . . . . .	71
图77.	DSI时钟方案 . . . . .	76
图78.	自动时钟通道控制 . . . . .	78
图79.	禁用了自动时钟通道控制并且时钟通道始终处于HS模式的示例 . . . . .	79
图80.	读指令后的BTA流程 . . . . .	81
图81.	LP模式进入流程 . . . . .	86
图82.	HBP区域内禁用LP . . . . .	86
图83.	HBP区域内使能LP . . . . .	87
图84.	VSA区域内仅使能LP模式 . . . . .	88
图85.	低功耗模式下最后一行 . . . . .	89
图86.	DSI视频时序配置寄存器 . . . . .	92
图87.	突发模式下的视频行 . . . . .	94
图88.	包含四个块且空包启用的配置 . . . . .	95
图89.	包含四个块且空包未启用的配置 . . . . .	96
图90.	使用同步脉冲的非突发模式下的LPSIZE计算 . . . . .	98
图91.	使用同步事件的突发或非突发模式下的LPSIZE计算 . . . . .	98
图92.	突发模式下VFP期间的28字节LP指令 . . . . .	100
图93.	延迟到最后一行的29字节LP指令 . . . . .	101
图94.	使用同步脉冲的非突发模式下的VLPSIZE计算 . . . . .	101
图95.	使用同步事件的非突发模式下的VLPSIZE计算 . . . . .	102
图96.	突发模式下的VLPSIZE . . . . .	102
图97.	突发模式下VACT期间的8字节LP指令 . . . . .	103
图98.	帧回应示例 . . . . .	104
图99.	帧回应放大图 . . . . .	105
图100.	使能了回应请求的通用短写指令 . . . . .	110

图101.	使用HSE的RCC配置	111
图102.	DSI模式下的LTDC配置	112
图103.	视频模式下的DSI主机配置	112
图104.	视频模式下使用PLLSAI1的LTDC像素时钟配置	113
图105.	视频模式下使用DSI PLL的DSI时钟配置	113
图106.	视频模式下的LTDC参数设置	114
图107.	视频模式下的LTDC层设置	115
图108.	视频模式下的数据和时钟通道配置	116
图109.	视频模式下的PHY时序配置	117
图110.	视频突发模式下的指令配置	118
图111.	视频突发模式下的显示接口配置	119
图112.	视频非突发模式下的指令配置	123
图113.	视频非突发模式下的显示接口配置	124
图114.	DSI适配指令模式的选择	125
图115.	适配指令模式下的LTDC时钟配置	125
图116.	适配指令模式下的LTDC参数配置	126
图117.	适配指令模式下的数据和时钟通道配置	127
图118.	适配指令模式下的PHY时序配置	128
图119.	适配指令模式下的指令传输配置	129
图120.	适配指令模式下的显示接口配置	130
图121.	小尺寸显示屏驱动示例	137
图122.	大尺寸显示屏驱动示例	138

## 1 标准和参考文献

本章列出了本文档中使用的标准和参考文献。

- MIPI®联盟显示串行接口（DSI）规范  
v1.1 - 2011年11月22日
- MIPI®联盟显示总线接口（DBI-2）规范  
v2.00 - 2005年11月16日
- MIPI®联盟显示指令集（DCS）规范  
v1.1 - 2011年11月22日
- MIPI®联盟显示像素接口（DPI-2）规范  
v2.00 - 2005年9月15日
- MIPI®联盟立体显示格式（SDF）规范  
v1.0 - 2011年11月22日
- MIPI®联盟D-PHY规范  
v1.1 - 2011年11月7日

## 2 概述

显示屏串行接口（DSI）是由MIPI（移动产业处理器接口）联盟定义的高速串行协议，提供显示模块与主机处理器之间的连接。

STM32是市场上首款具有集成DSI主机的MCU产品（参见[表 1](#)获取适用产品列表）。STM32 DSI主机提供高度集成的解决方案，这要得益于它的内部MIPI D-PHY、专用PLL和1.2 V调节器。

DSI主机提供速度高达1 Gb/s的高速通信接口。DSI主机允许微控制器使用更少的引脚与显示屏连接，无需外部电桥。DSI接口完全可配置，可与目前市售的DSI兼容显示屏实现轻松连接。

应用可受益于DSI主机支持的轻松连接和更少引脚数，降低了PCB复杂性和系统总成本。

DSI主机与LCD-TFT显示控制器（LTDC）深度集成，用于简化应用开发和移植。

STM32 DSI主机提供可扩展的架构。根据带宽要求，用户可以选择一个或两个数据通道。

### 2.1 显示屏连接

根据显示屏是否具有内部控制器和帧缓冲器，可将显示屏分为两类：

- 显示模块具有显示控制器和帧缓冲器：它们集成了图形RAM（GRAM），用于存储要显示的帧，并具有控制刷新操作的显示控制器。  
MCU使用一组指令更新显示屏的帧缓冲器内容。  
显示屏依赖其内部控制器和帧缓冲器来执行刷新操作，无需MCU的干预。
- 显示模块无显示控制器和帧缓冲器：这些显示屏依赖于MCU发送数据和视频时序信息的实时像素流，以便刷新显示。

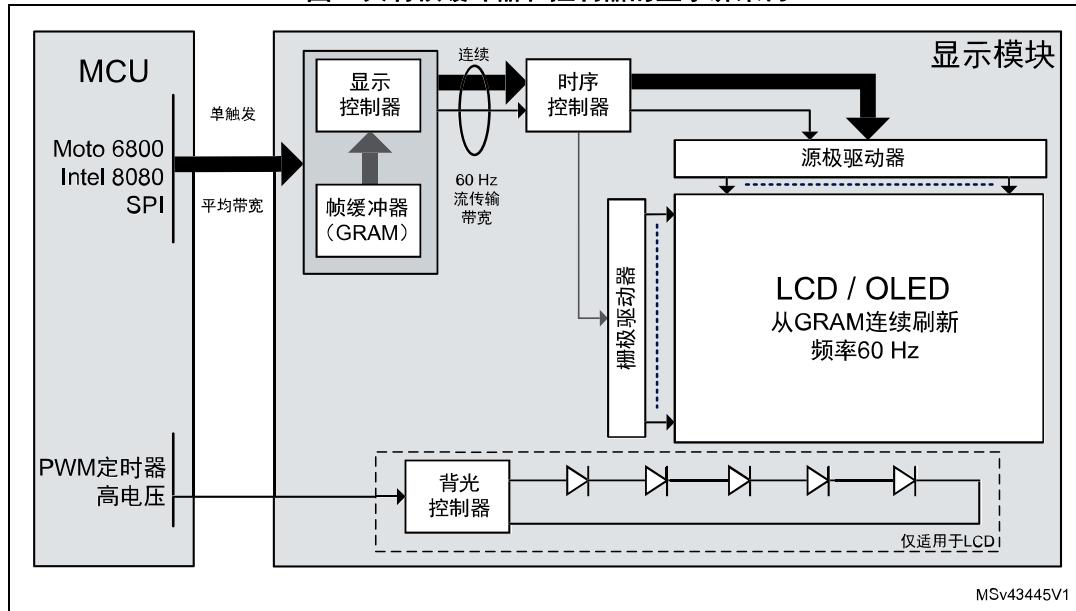
这些不同类型的显示屏采用不同类型的显示接口：

- 至有控制器和帧缓冲器的显示模块的接口：
  - SPI接口：使用的引脚很少（最多6个引脚），但速度很慢且不适合动画。
  - 并行接口：使用并行总线发送指令和数据，因此拥有的带宽高于SPI接口。  
常见的并行接口有Motorola 6800总线和Intel 8080总线。并行接口需要最多22个引脚（16个数据信号和6个控制信号）。

显示控制器和GRAM帧缓冲器位于显示侧。MCU发送更新显示屏GRAM的指令。更新将一次完成。

[图 1](#)显示的是集成了帧缓冲器和显示控制器的显示屏架构及其相应的接口方案：

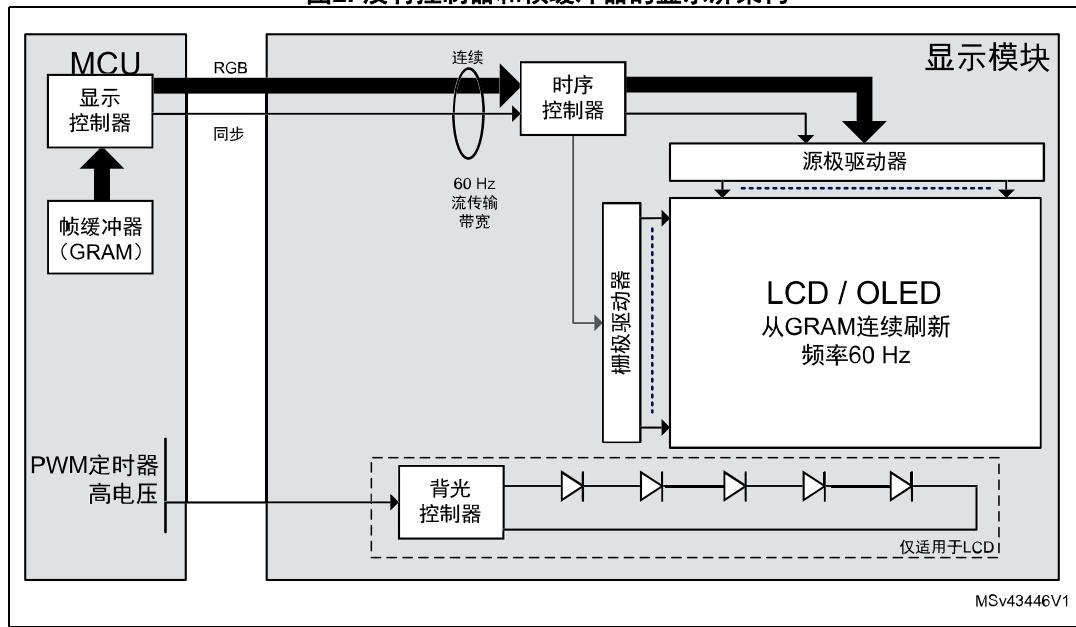
图1. 具有帧缓冲器和控制器的显示屏架构



- 至没有显示控制器和帧缓冲器的显示模块的接口：
  - RGB接口：MCU提供像素数据和视频时序信号。RGB接口具有很好的实时性能，但要求MCU侧具有高带宽，用于对显示屏的馈送。它还需要最多28个引脚：24 RGB数据（R[0:7]G[0:7]B[0:7]）和四个同步信号（PCLK、HSYNC、VSYNC、DATAEN）。MCU持续生成像素数据和视频控制信号来驱动显示屏。帧缓冲器位于MCU侧。

图2展示了没有帧缓冲器和显示控制器的显示屏的典型架构：

图2. 没有控制器和帧缓冲器的显示屏架构



## 2.2 MIPI显示屏规格标准

MIPI显示屏工作组规定了主机处理器与显示屏之间的互联。

该工作组定义了一组规范，目的是实现现有协议寻址显示的标准化：

- 显示总线接口（DBI）涵盖了有显示控制器和帧缓冲器的协议寻址显示模块。它定义了三种接口类型：
  - A类，支持并行Motorola 6800总线。
  - B类，支持并行Intel 8080总线。
  - C类，支持SPI接口。
- 显示指令集（DCS）规定了适用于支持MIPI-DBI接口的显示屏的指令。
- 显示像素接口（DPI）支持针对没有控制器和帧缓冲器的显示屏的RGB接口。

为了以更好的方式驱动现代显示屏，MIPI显示屏工作组开发了新的协议。

为了减少MCU和显示屏之间的连线数量，MIPI显示屏工作组定义了DSI。

DSI封装了DBI指令（称为指令模式）或DPI信号（称为视频模式），并将它们以信号的形式传输至显示屏。这允许只使用四个或六个引脚来连接标准显示屏，并获得与DPI相同性能。

DSI可通过单一接口支持所有可能的显示屏架构（有或没有显示控制器和帧缓冲器）。

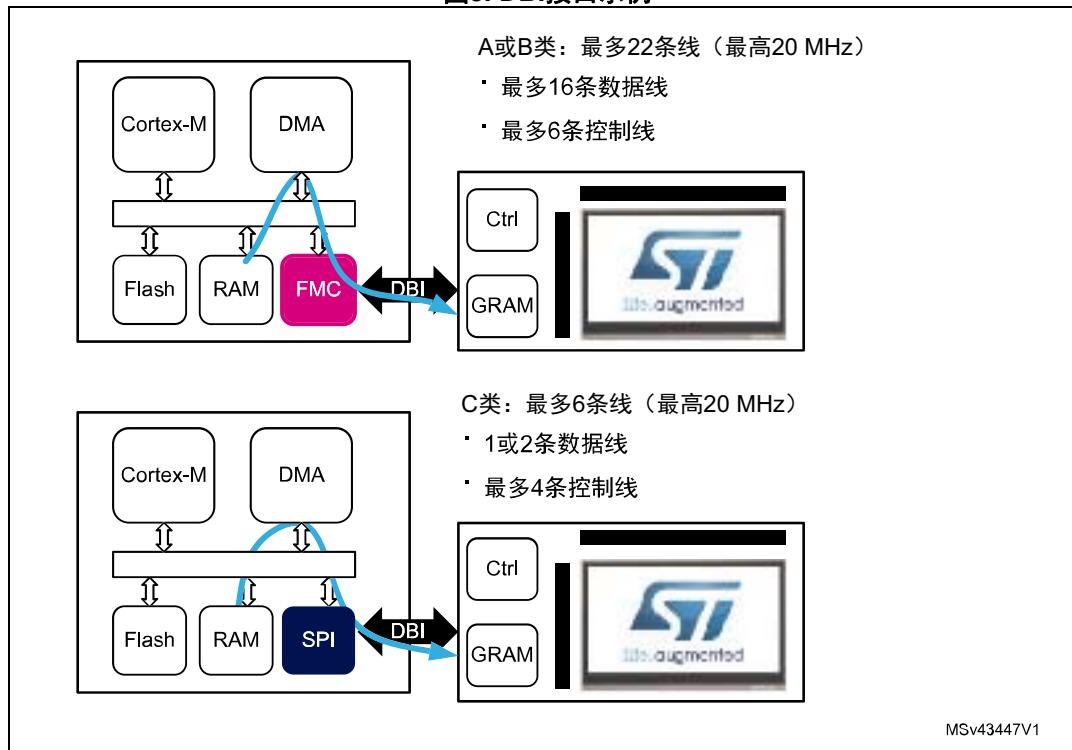
## 2.3 STM32产品支持的显示接口

所有STM32产品均通过SPI接口支持MIPI-DBI C类接口。

集成了F(S)MC（灵活的同步存储控制器）的STM32支持MIPI-DBI A类和B类接口。请参见应用笔记“*TFTLCD与高密度STM32F10xxxFSMC的连接*”（AN2790）了解关于通过FSMC连接显示屏的更多信息。

[图 3](#)显示了使用DBI接口连接显示屏的STM32微控制器：

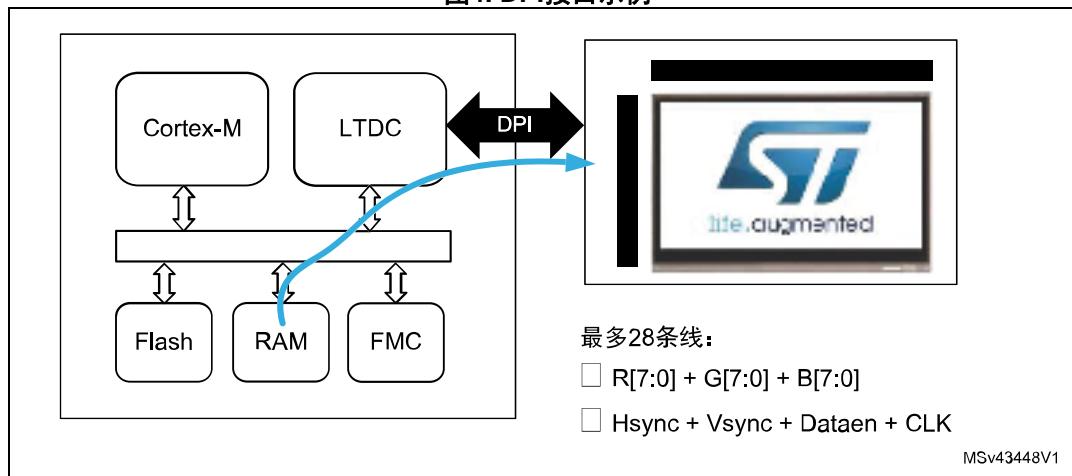
图3. DBI接口示例



集成了LTDC (LCD-TFT显示控制器) 的STM32支持MIPI-DPI。请参见应用笔记“STM32微控制器 (AN4861) 上的LCD-TFT显示控制器 (LTDC)”了解关于该主题的更多信息。

图4显示的是STM32微控制器使用DPI接口连接显示屏:

图4. DPI接口示例

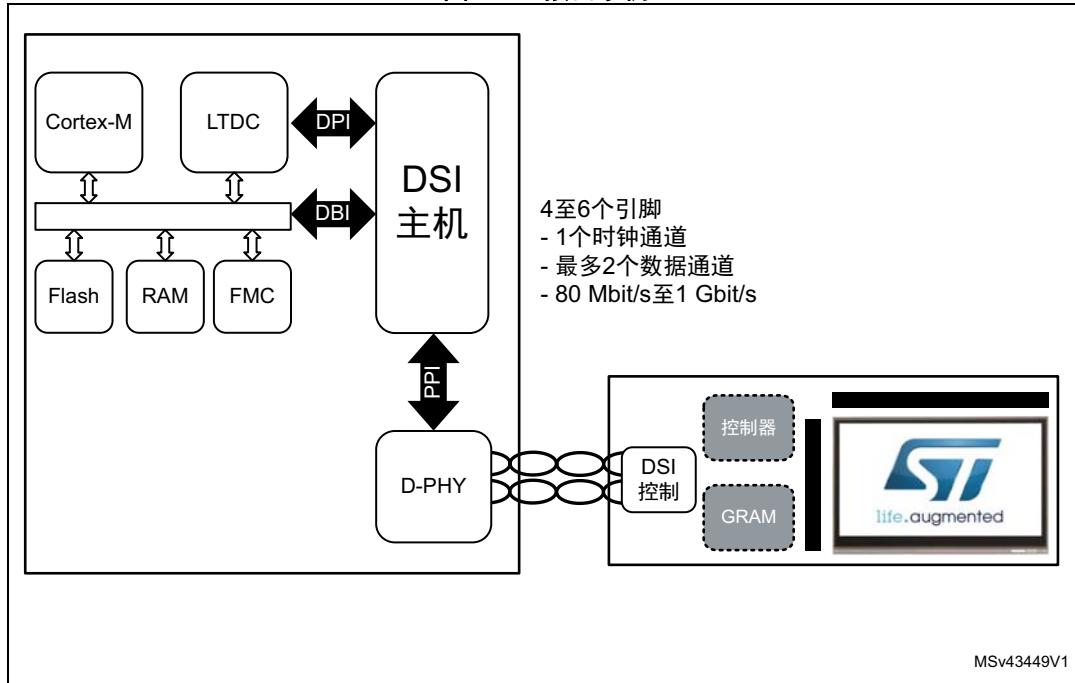


集成了DSI主机的最新STM32产品支持MIP-DSI (参见表1)。

DSI接口可以连接两种类型的显示屏。对于具有GRAM和显示控制器的显示屏，与DBI模式一样，DSI主机发送指令以刷新GRAM。对于没有GRAM和显示控制器的显示屏，与DPI模式一样，DSI主机发送像素数据和视频同步事件流。

[图 5](#)显示的是STM32微控制器使用DSI接口连接显示屏。

图5. DSI接口示例



下面的[表 1](#)总结了STM32产品中的不同显示接口方案。

表1. STM32产品中的显示屏连接

传统显示接口	MIPI显示接口标准	支持有帧缓冲器的显示屏	支持无帧缓冲器的显示屏	引脚数	并行/串行	STM32支持
Motorola 6800	DBI A类	有	无	12~22	并行	所有具有F(S)MC的STM32
Intel 8080	DBI B类	有	无	12~22	并行	所有具有F(S)MC的STM32
SPI	DBI C类	有	无	4~6	串行	所有STM32
RGB	DPI	无	有	20~28	并行	所有具有LTDC的STM32 (STM32F4x9、STM32F7x6、STM32F7x7、STM32F7x8、STM32F7x9)
-	DSI	有 (DSI: 命令模式)	有 (DSI视频模式)	4~6	串行	STM32F469、 STM32F479、 STM32F7x8、 STM32F7x9

## 2.4 不同STM32微控制器的DSI主机可用性

STM32F469/479系列、STM32F7x8系列和STM32F7x9系列微控制器支持DSI主机。[表 2](#)详细列出了STM32微控制器上的DSI特性。

**表2. 以DSI主机为特色的STM32微控制器**

产品	特性								
	FLASH (字节)	片上 SRAM (字节)	MIPI- DSI主 机	LTDC (1)	Quad-SPI (2)	JPEG 编解码 器 (3)	Chrom-Art 加速器 (4)	最高AHB频 率 (MHz) (5)	最高FMC SRAM和 SDRAM频率 (MHz)
STM32F469/479 系列	最多2M	384k	有	有	有	无	有	180	90
STM32F7x8系列 STM32F7x9系列	最多2M	512k	有	有	有	有	有	216	100

1. LTDC是一种TFT-LCD显示控制器。关于STM32的LTDC接口的更多信息，请参见应用笔记AN4861。
2. Quad-SPI接口可以连接外部存储器以扩大应用规模。关于STM32的QSPI接口的更详细信息，请参见应用笔记AN4760。
3. JPEG编解码器为JPEG编码和解码提供硬件加速。
4. Chrom-Art Accelerator®是一种ST专有的二维图形加速引擎。
5. LTDC以AHB速度获取图形数据。

## 2.5 DSI主机的优势

相较于其他显示接口，DSI主机具有许多优势。

**表3. DSI主机相较于其他显示接口的优势**

DSI优势	注释
引脚数少	DSI接口大幅降低了引脚要求。它可以提供与RGB接口相同的传输速率，但使用的引脚更少。 例如，LTDC需要28个引脚，而DSI只需4或6个引脚。
更低的PCB复杂性	由于要路由的信号更少，因此降低了PCB板复杂性，无需额外的PCB层。
低EMI（电磁干扰）	得益于其差分信令方案，DSI接口生成的EMI水平较低。
可扩展的架构	DSI主机提供可扩展的架构。根据带宽要求，用户可以选择一个或两个数据通道。
高度集成的解决方案	DSI主机具有集成的D-PHY、PLL和调节器，无需外部组件。
智能架构	DSI主机与LTDC控制器一起使用，后者相当于视频引擎。
标准化接口	DSI主机支持MIPI-DSI规范中描述的所有工作模式。标准化接口提高了与DSI显示屏的互操作性。
优化工作模式	STM32MCU DSI主机具有优化模式，可与集成了GRAM（图形RAM）的DSI显示屏一起使用。此模式降低了MCU侧的带宽和存储要求。
常见的DSI显示屏的优势	DSI显示屏被广泛应用于智能手机和平板电脑。对于许多其他嵌入式应用而言，它们正变得越来越有吸引力。

## 2.6 智能架构中的DSI主机

STM32系统架构主要由32位多层总线矩阵构成，它们将各种主设备与从设备互联。

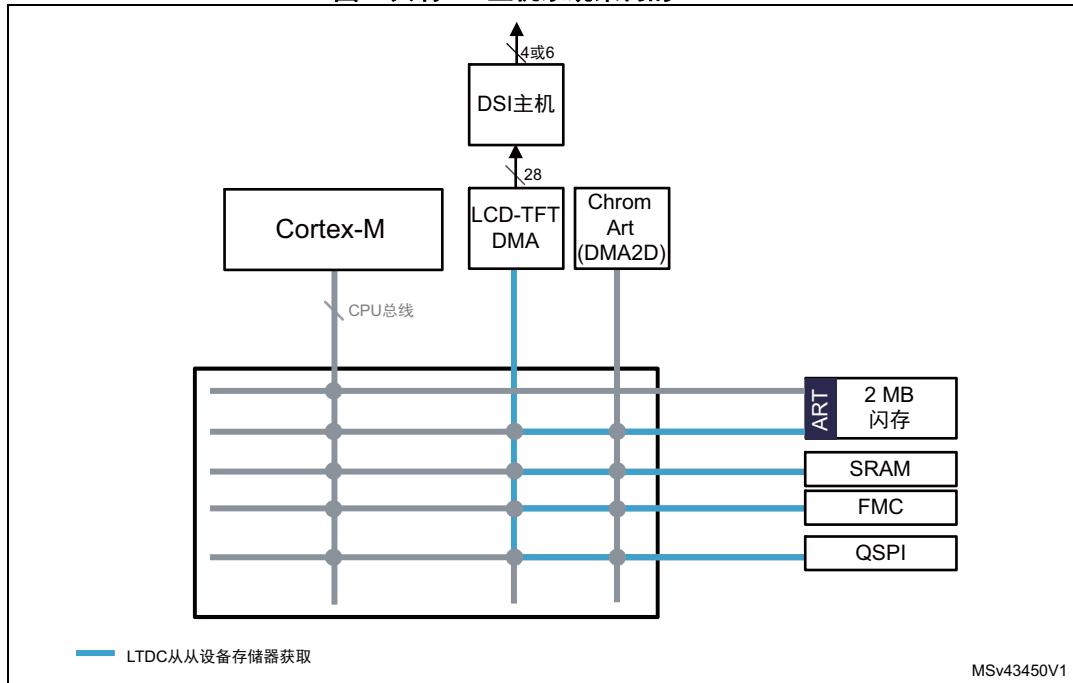
LTDC是总线矩阵上的主设备。它从各种存储器位置获取图形数据。

使用LTDC显示控制器作为视频推流器，可以轻松集成DSI主机。

来自LTDC的像素和视频时序数据由DSI主机串行化，并被发送至显示模块。

[图 6](#)显示了STM32系统中的DSI主机互联。

图6. 具有DSI主机系统架构的STM32



### 3 DSI简介

DSI详细说明了主机处理器和显示模块之间的接口。它建立在现有MIPI联盟规范的基础上，采用DPI-2、DBI-2和DCS标准中规定的像素格式和指令集。

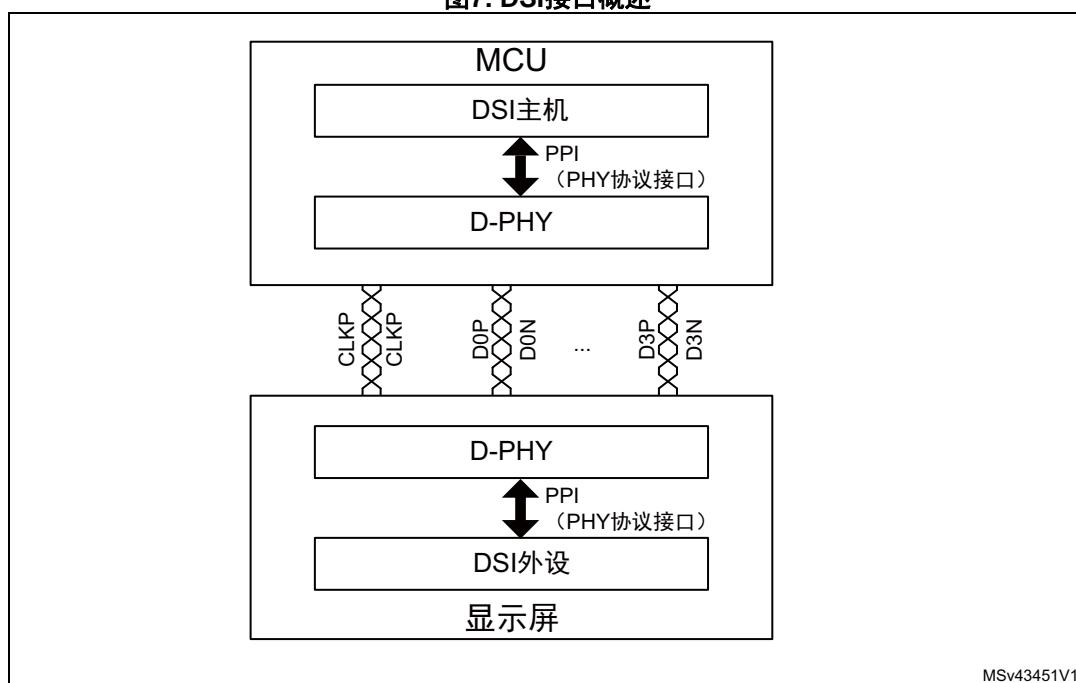
DSI主机将像素数据、信号事件或指令封装到DSI数据包中，然后发送给显示屏。DSI主机可以从显示屏回读状态或像素信息。

DSI主机通过PHY协议接口（PPI）将DSI数据包以并行数据的形式传输至D-PHY。D-PHY将数据包串行化并通过串行链路发送。

在显示屏侧，数据包被分解为并行数据、信号事件和指令。

[图 7](#)提供了主机和显示屏之间的DSI接口的总览：

图7. DSI接口概述



### 3.1 DSI工作模式

DSI标准定义了DSI主机和DSI显示屏的两种基本工作模式：指令模式和视频模式。

#### 3.1.1 指令模式

指令模式是指在一项操作中，表现为向显示模块发送指令和数据（如DCS规范所定义）。支持指令模式的显示模块包含控制器和帧缓冲器。

主机处理器通过向显示控制器发送指令、参数和数据来间接控制显示模块上的活动。

### 3.1.2 视频模式

视频模式是指在一项操作中，采用实时像素流的形式进行从主机处理器到显示模块的传输。显示模块依赖于主机处理器以足够的带宽提供图像数据，从而避免显示的图像上发生闪烁或其他可见异常现象。

## 3.2 DSI物理层

DSI物理层基于MIPI D-PHY规范，此规范描述了源同步的高速（HS）低功耗（LP）链路。

### 3.2.1 PHY配置

该链路包含主机处理器提供的一个双倍数据速率HS时钟通道和一至四个串行数据通道。

最小链路配置包含一个时钟通道和一个数据通道。

每个通道包含一对连线，在高速传输模式下以低摆幅差分模式驱动，在低功耗模式下以单端模式驱动。

DSI通信可以是正向（主机至显示屏）或反向（显示屏至主机）。

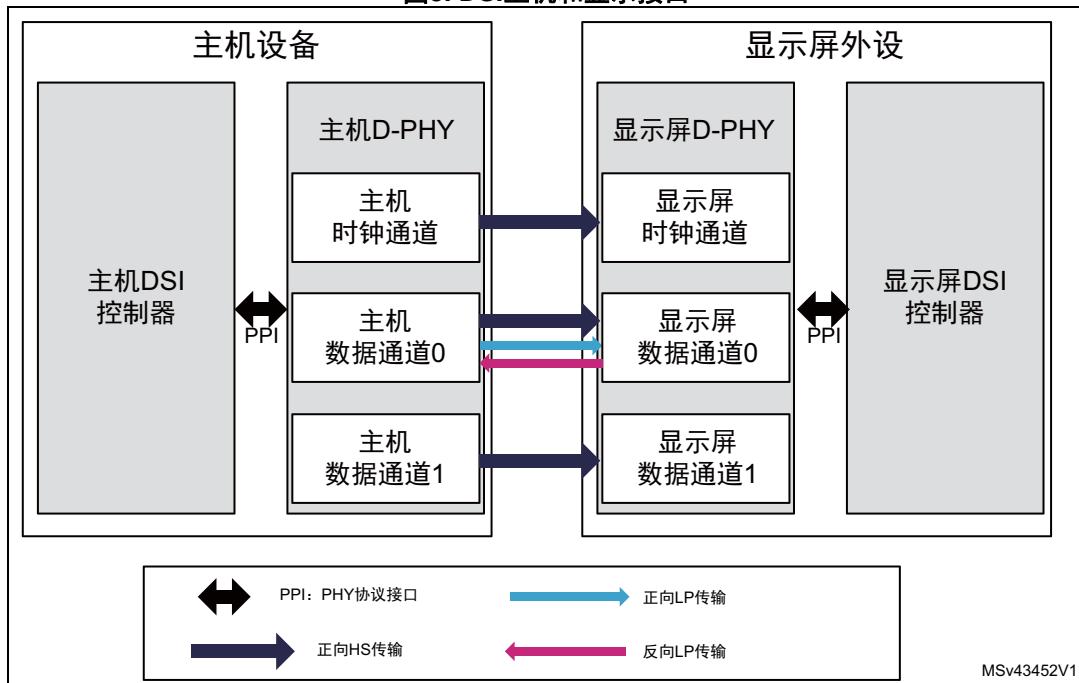
所有反向通信（从显示屏至主机）均在低功耗模式下完成，并且只使用数据通道0。

在指令模式系统中，通道0是双向的，而在视频模式系统中，既可以是单向的也可以双向的。

所有其他通道均为单向，仅支持高速模式。

[图 8](#)显示了具有一个时钟通道和两个数据通道的PHY配置示例。

图8. DSI主机和显示接口



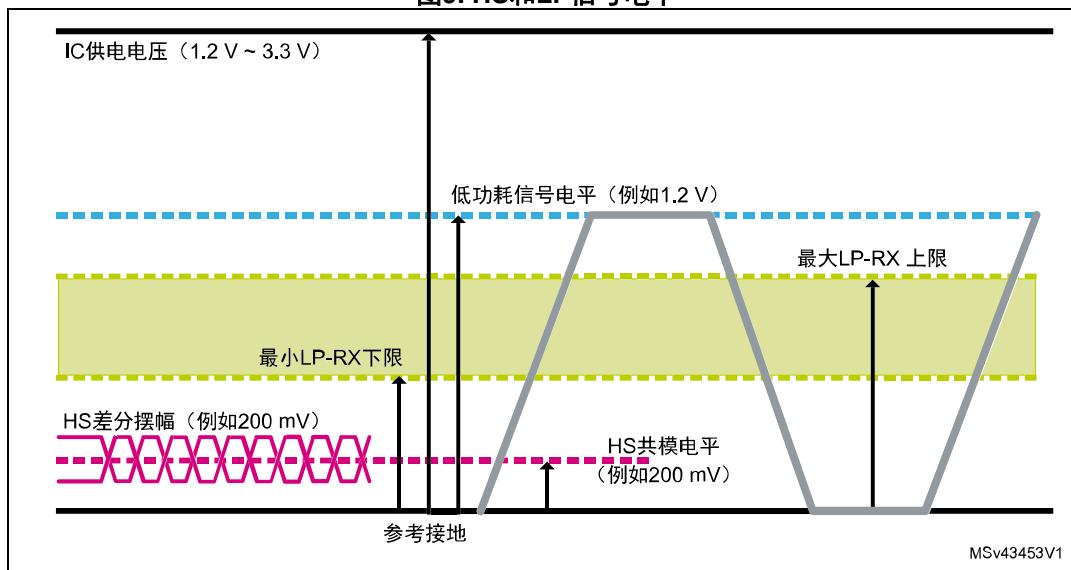
### 3.2.2 PHY信令模式

DSI物理层中有两个信令模式：

- 高速（HS）模式：用于快速数据传输，在差分模式下使用通道信号，速度最高1.5 Gbit/s。
- 低功耗（LP）模式：主要用于控制目的。可在单端模式下以10 Mbit/s的最大传输速率独立驱动通道信号对。

[图 9](#)显示了DSI物理层中HS和LP信号电平的示例：

图9. HS和LP信号电平



### 数据通道状态

DSI发射器有一个HS发射器和两个LP发射器，前者用于HS-0或HS-1的连线差分驱动，后者用于以单端模式独立驱动差分对 ( $D_P$  和  $D_N$ ) 的每条线。这导致HS发射器有两种可能的通道状态 (HS-0和HS-1)，而LP发射器有四种可能的通道状态 (LP-00、LP-10、LP-01和LP-11)。

通道状态反映了工作模式，用于从一种模式过渡到其他模式（参见表 4）。

表4. 数据通道状态和工作模式

状态码	$D_P$ 线 电平	$D_N$ 线 电平	高速 突发模式	低功耗 控制模式	低功耗逸出模 式
HS-0	HS low	HS high	差分0	x	x
HS-1	HS high	HS low	差分1	x	x
LP-00	LP low	LP low	x	电桥	空间
LP-01	LP low	LP high	x	HS-Rqst	Mark-0
LP-10	LP high	LP low	x	LP-Rqst	Mark-1
LP-11	LP high	LP high	x	STOP	(回到停止状 态)

### 3.2.3 数据通道工作模式

数据通道有三种工作模式：控制、高速传输和逸出。

#### 控制模式

复位后，数据通道处于控制模式 (LP-11停止状态)。所有其他模式始于并结束于控制模式。

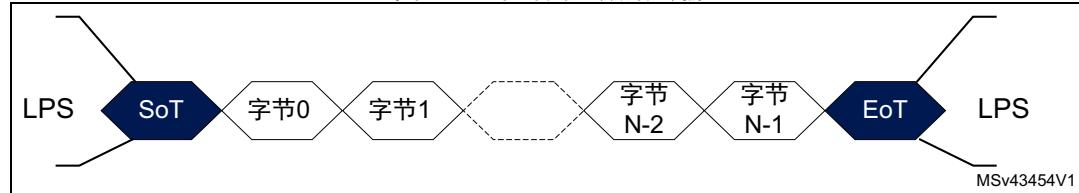
## 高速传输模式

高速传输始于并结束于停止状态（LP-11）。

为了实现主机与显示屏之间的同步，需添加头标和尾标序列。在接收器侧，它们会被移除，因为它们不属于实际有效负载数据。

[图 10](#)显示了基本的高速传输的示例。

图10. 基本的高速数据传输



### 1. 传输开始（SoT）流程

在收到HS请求（LP-11、LP-01、LP-00）时，数据通道进入HS模式。

主机首先驱动HS-0，然后驱动HS同步序列（00011101）以允许从设备同步。

然后，主机继续传输HS数据。

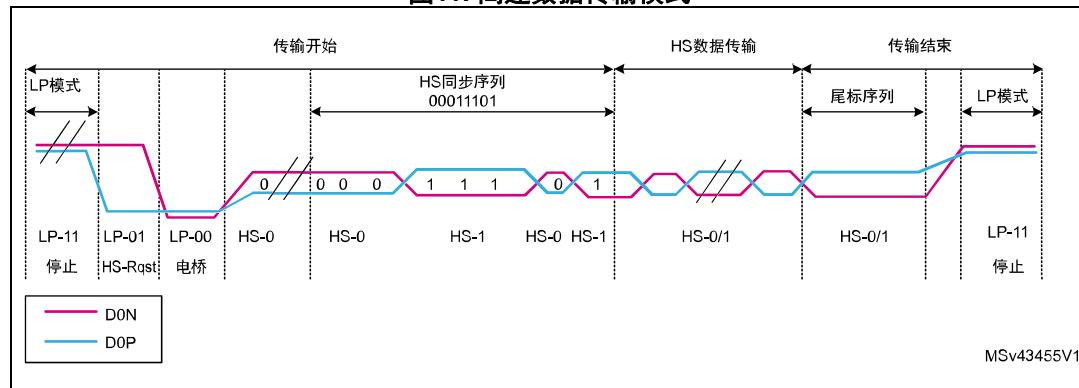
### 2. 传输结束（EoT）流程

在高速突发传输结束后，主机发送尾标序列。

尾标序列与传输的最后一个数据位相反：如果最后一个有效负载位是HS-0，那么发射器发送HS-1作为尾标序列，否则发送HS-0。

数据通道通过停止状态LP-11返回控制模式。

图11. 高速数据传输模式

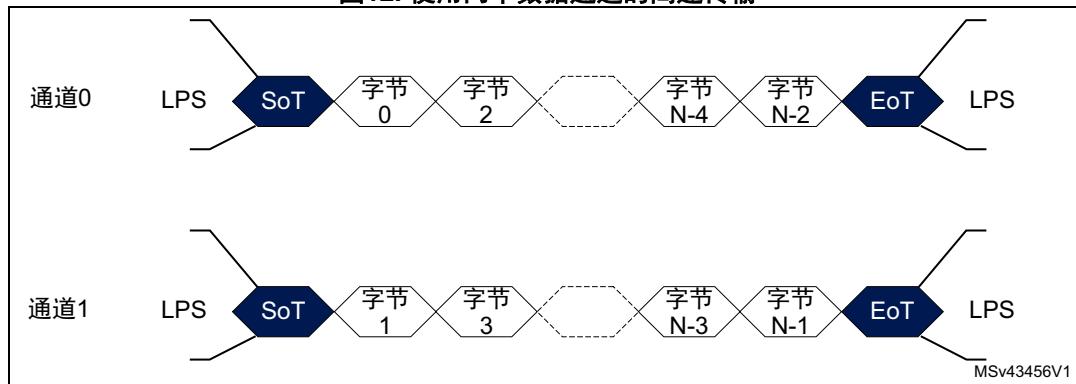


### 3. 多通道支持

可使用一个或更多数据通道完成高速传输。

[图 12](#)显示了使用两个数据通道的高速传输示例。

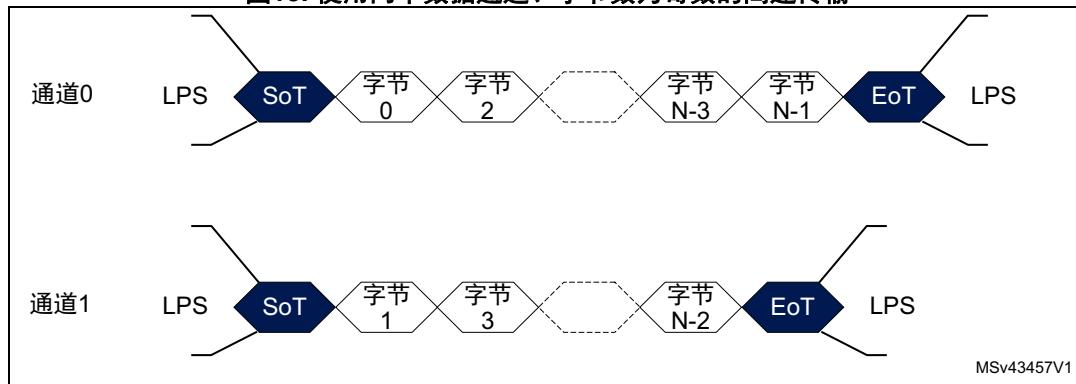
图12. 使用两个数据通道的高速传输



如果传输的字节数不是通道数的整数倍，则一些通道可能先于其他通道完成高速传输。

[图 13](#)显示了一个示例，其中有两个数据通道和奇数个字节，导致数据通道1先于数据通道0完成高速传输并发出EoT序列。

图13. 使用两个数据通道、字节数为奇数的高速传输



## 逸出模式

数据通道可通过逸出模式请求流程（LP-11、LP-10、LP-00、LP-01、LP-00）进入逸出模式。

在进入逸出模式后，发射器发送8位进入指令指示请求的操作。

逸出进入指令可以是：

- 低功耗数据传输（LPDT）
- 超低功耗状态（ULPS）
- 远程触发信号

[表 5](#)列出了逸出模式支持的不同进入指令：

表5. 逸出模式指令

逸出模式操作	命令类型	进入指令模式
低功耗数据传输	模式	11100001
超低功耗状态	模式	00011110
未定义-1	模式	10011111

表5. 逸出模式指令（续）

逸出模式操作	命令类型	进入指令模式
未定义-2	模式	11011110
复位触发	触发	01100010
撕裂效应触发	触发	01011101
回应触发	触发	00100001
未知-5	触发	10100000

注：仅通道0支持低功耗数据传输和触发消息。

通过逸出模式退出流程退出逸出模式（LP-10、LP-11）。

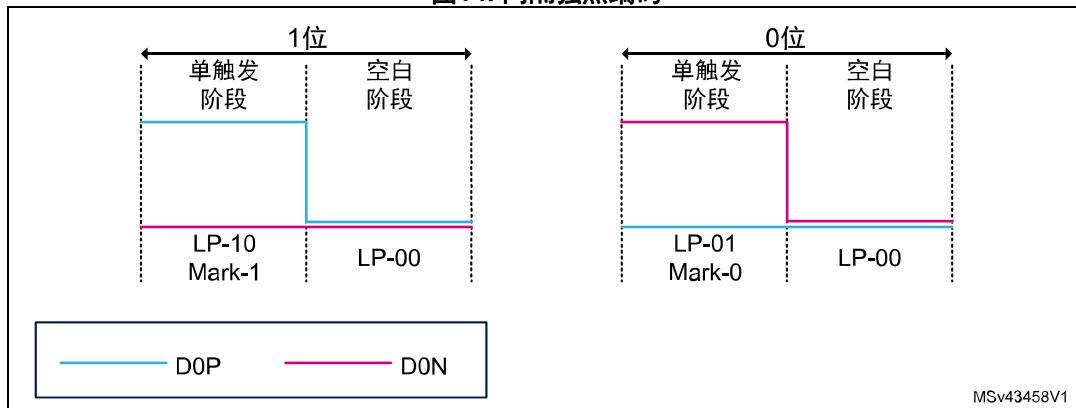
### 1. 间隔独热编码

在逸出模式下，使用间隔独热编码对进入指令和低功耗数据传输（LPDT）通信进行编码，这意味着每个标记状态与空白状态交替出现。

因此，每个符号包含两部分：独热阶段（Mark-0或Mark-1）和空白阶段。

间隔独热编码提供高可靠性和从数据流提取时钟的能力。但是，它需要双倍的数据传输带宽（参见图 14）。

图14. 间隔独热编码



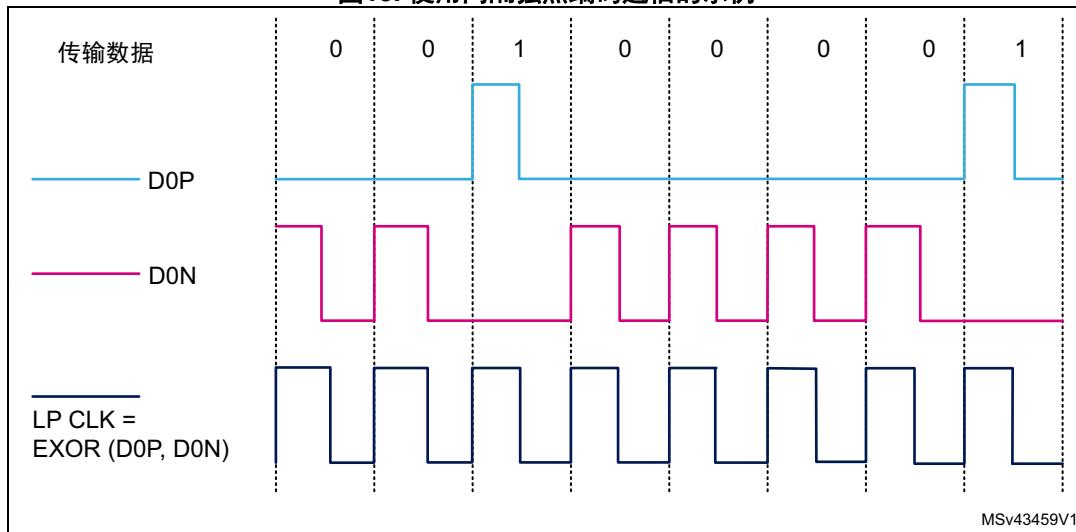
### 2. 低功耗时钟

发射器使用低功耗时钟信号进行低功耗通信，但是此时钟没有传输到接收侧。

数据通过间隔独热位编码进行自检，接收器可从使用异或函数的两路信号检索时钟。

图 15 显示了使用间隔独热编码的数据传输示例。通过对两个信号（DOP 和 DON）应用异或函数获取 LP 时钟。

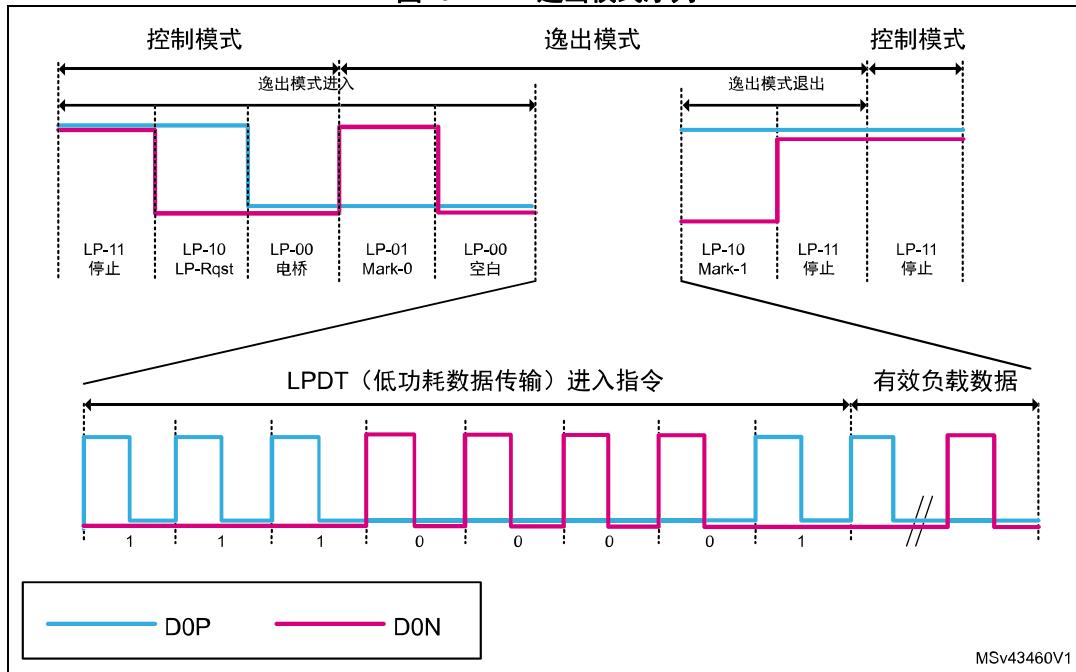
图15. 使用间隔独热编码通信的示例



### 3. 低功耗数据传输

在逸出模式进入序列后，发射器发送LPDT进入指令（11100001），然后是实际有效负载数据。图 16 显示了 LPDT 逸出模式序列。

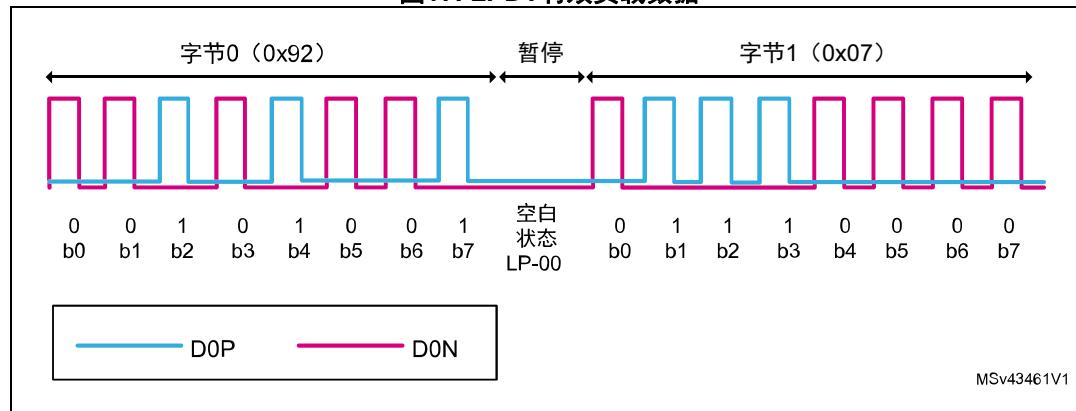
图16. LPDT逸出模式序列



数据按照LSB (least significant bit) 先行发送，对于多字节有效负载，首先传输最低有效字节。

在LPDT过程中，可通过维持线上的空白状态（LP-00）暂停通道。图 17显示了LPDT有效负载数据传输的示例。

图17. LPDT有效负载数据



#### 4. 超低功耗状态

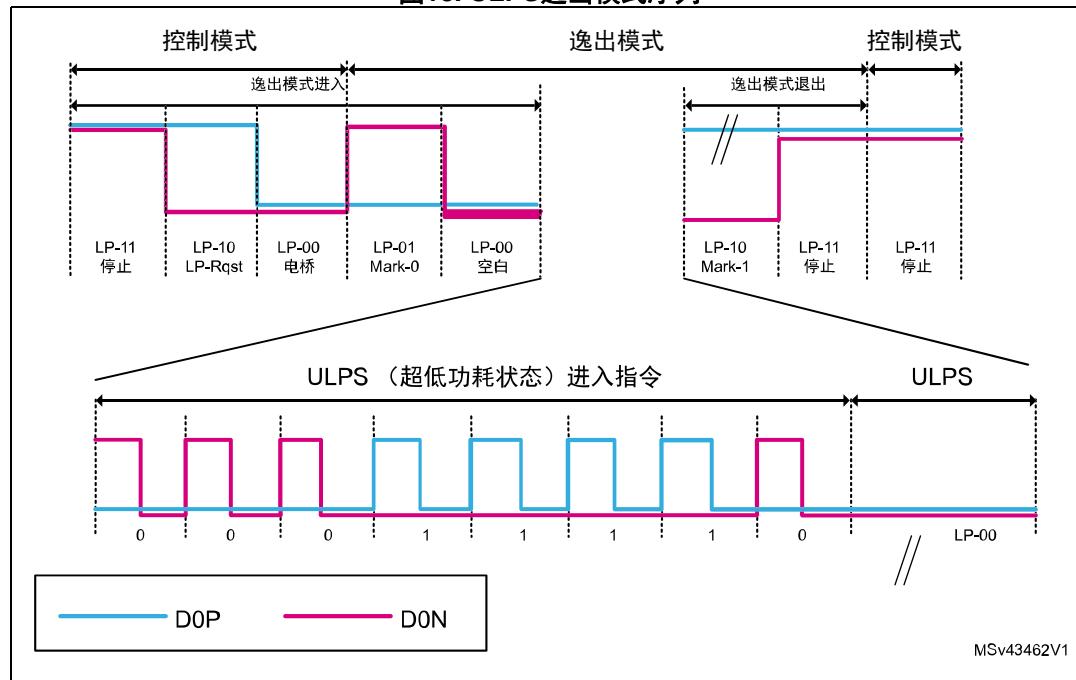
为了降低功耗，DSI主机可使数据通道进入超低功耗状态（ULPS）。这是通过以下流程实现的：

- 进入逸出模式（LP-11、LP-10、LP-00、LP-01、LP-00）。
- 发送ULPS进入指令（00011110）。
- 使通道信号保持LP-00状态。

通过Mark-1（LP-10）退出ULPS状态，之后处于停止状态（LP-11）。

图 18显示了ULPS逸出模式序列。

图18. ULPS逸出模式序列



## 5. 触发器

触发信令是一种消息传送系统，用于向接收侧发送标志。可以正向或反向发送。

DSI中使用了三种触发消息：

- 回应触发：显示屏向DSI主机发送的消息，用于指示最后一次传输已接收且无错误。这是反向通信（参见图 19）。
- 撕裂效应触发：被显示屏用来将内部时间通知给主机。这是反向通信（参见图 20）。
- 复位触发信号：由主机发送，用于复位显示（参见图 21）。

图19. 回应触发电示例

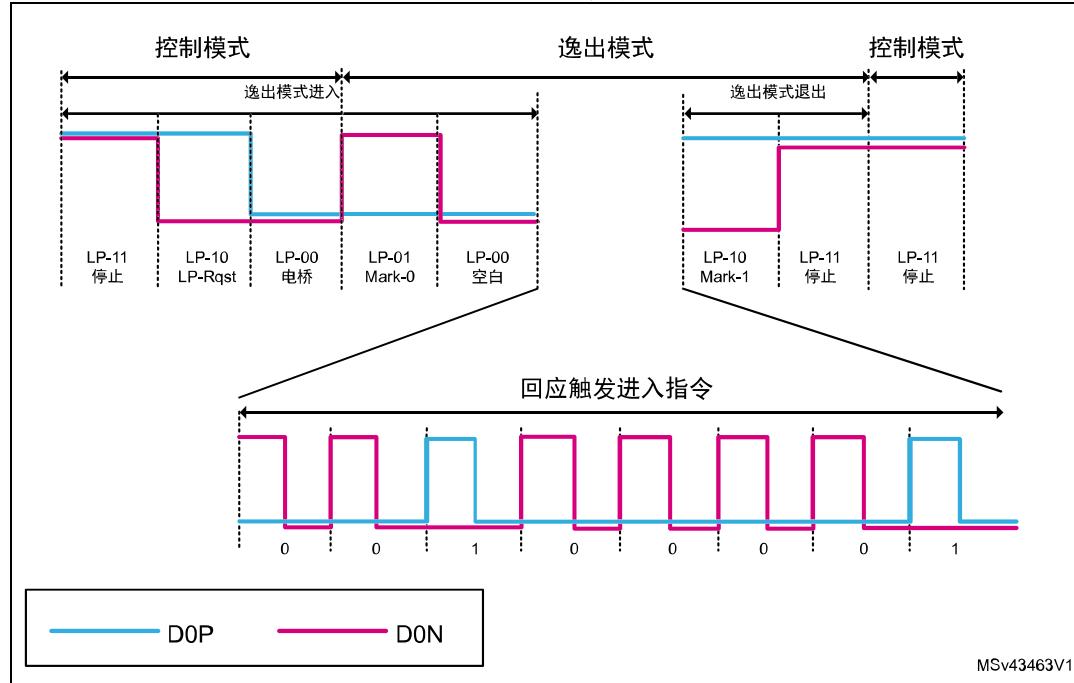


图20. 撕裂效应触发表例

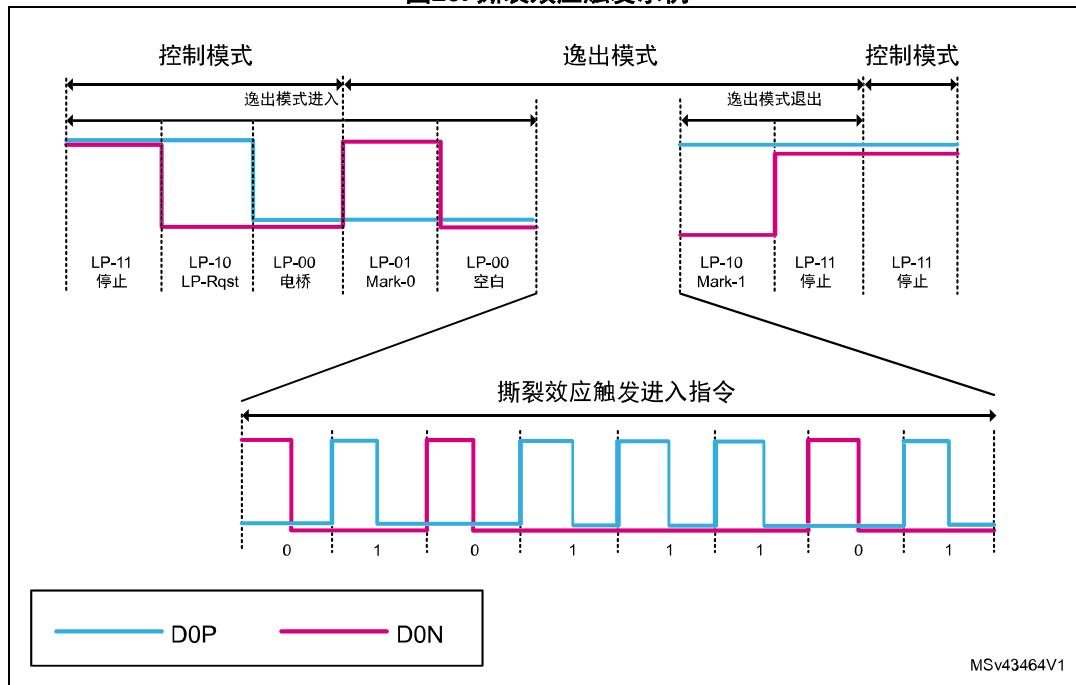
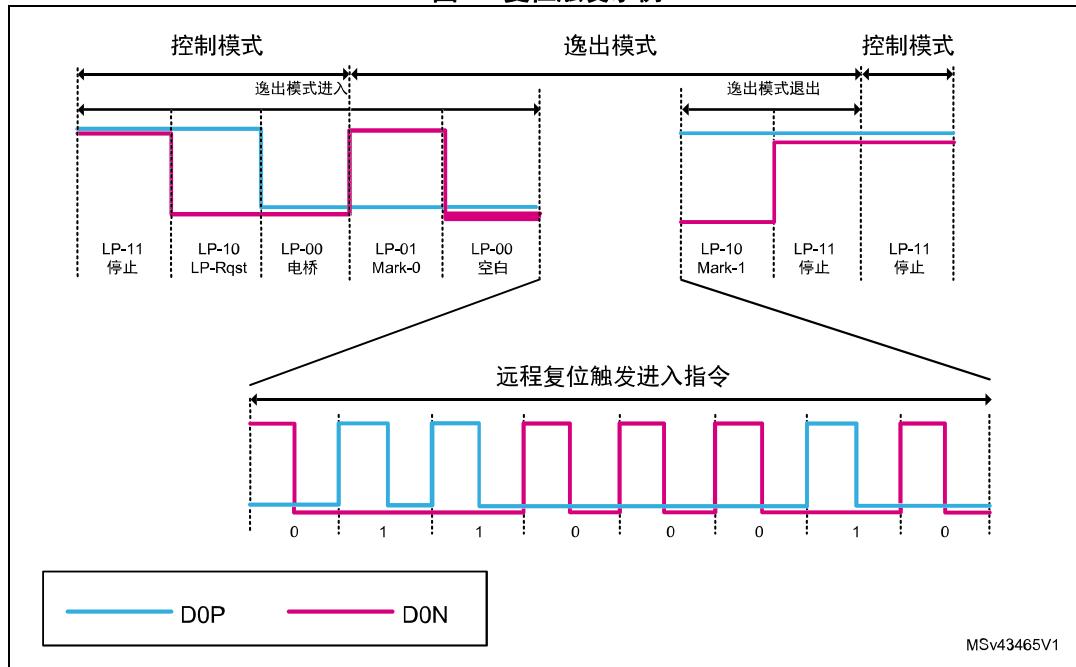


图21. 复位触发表例



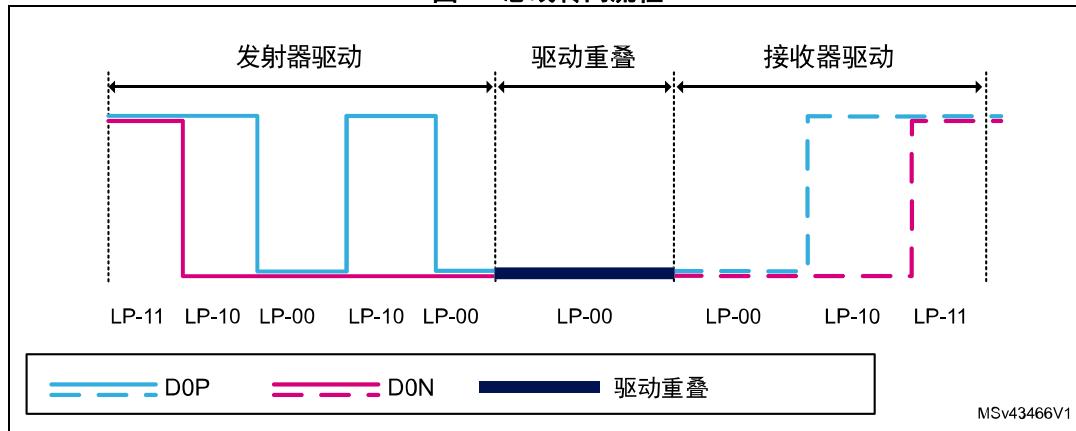
### 3.2.4 双向通道和总线转向流程

DSI在数据通道0上仅支持双向数据链路。为允许反向传输，可使用总线转向（BTA）流程调换数据通道方向。

从停止状态（LP-11）开始BTA。在接收器获得总线所有权后，反向传输可以开始。然后，接收器必须通过相同的转向流程将总线所有权还给主机。

[图 22](#)显示了BTA序列。

图22. 总线转向流程



### 3.2.5 时钟通道功耗模式

可用三种不同的功耗模式驱动DSI CLK通道：低功耗模式、超低功耗状态和高速时钟模式。

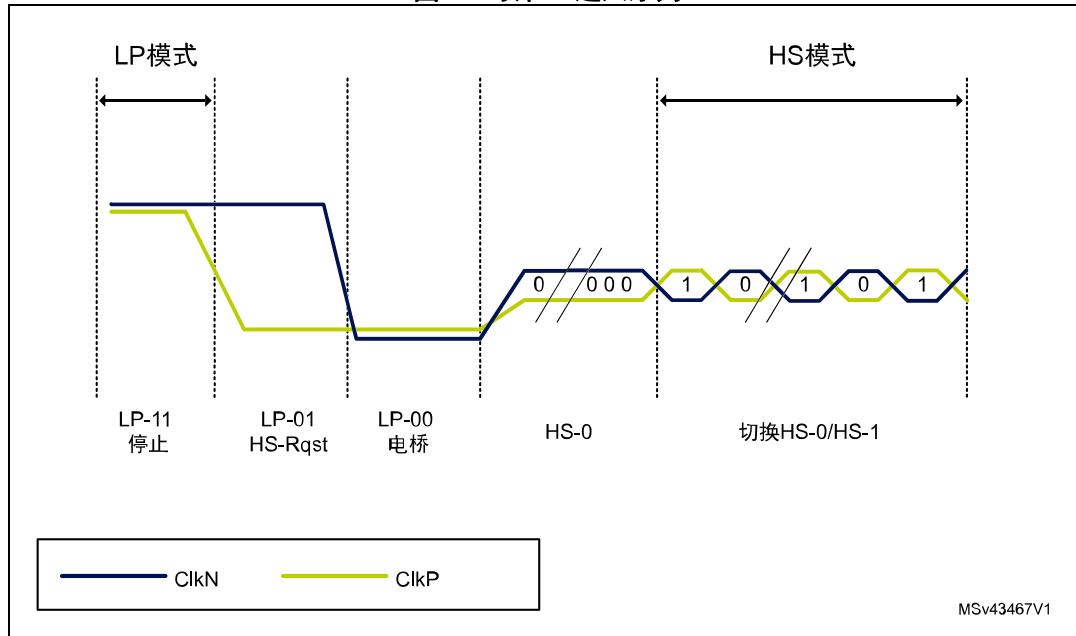
#### 低功耗模式

在低功耗模式下，时钟通道处于LP-11停止状态。所有其他模式始于并结束于LP模式。

#### 高速模式

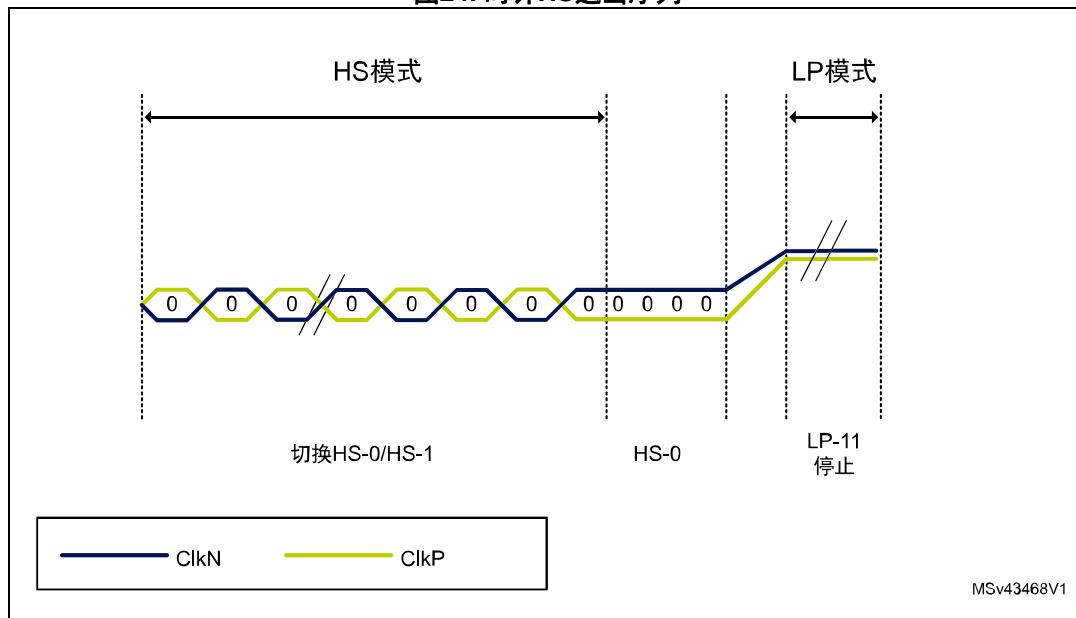
通过驱动HS进入序列（LP-11、LP-01、LP-00、HS-0），时钟通道从LP模式进入HS模式（参见[图 23](#)）。此后，时钟通道进入HS模式并开始切换HS-0、HS-1。

图23. 时钟HS进入序列



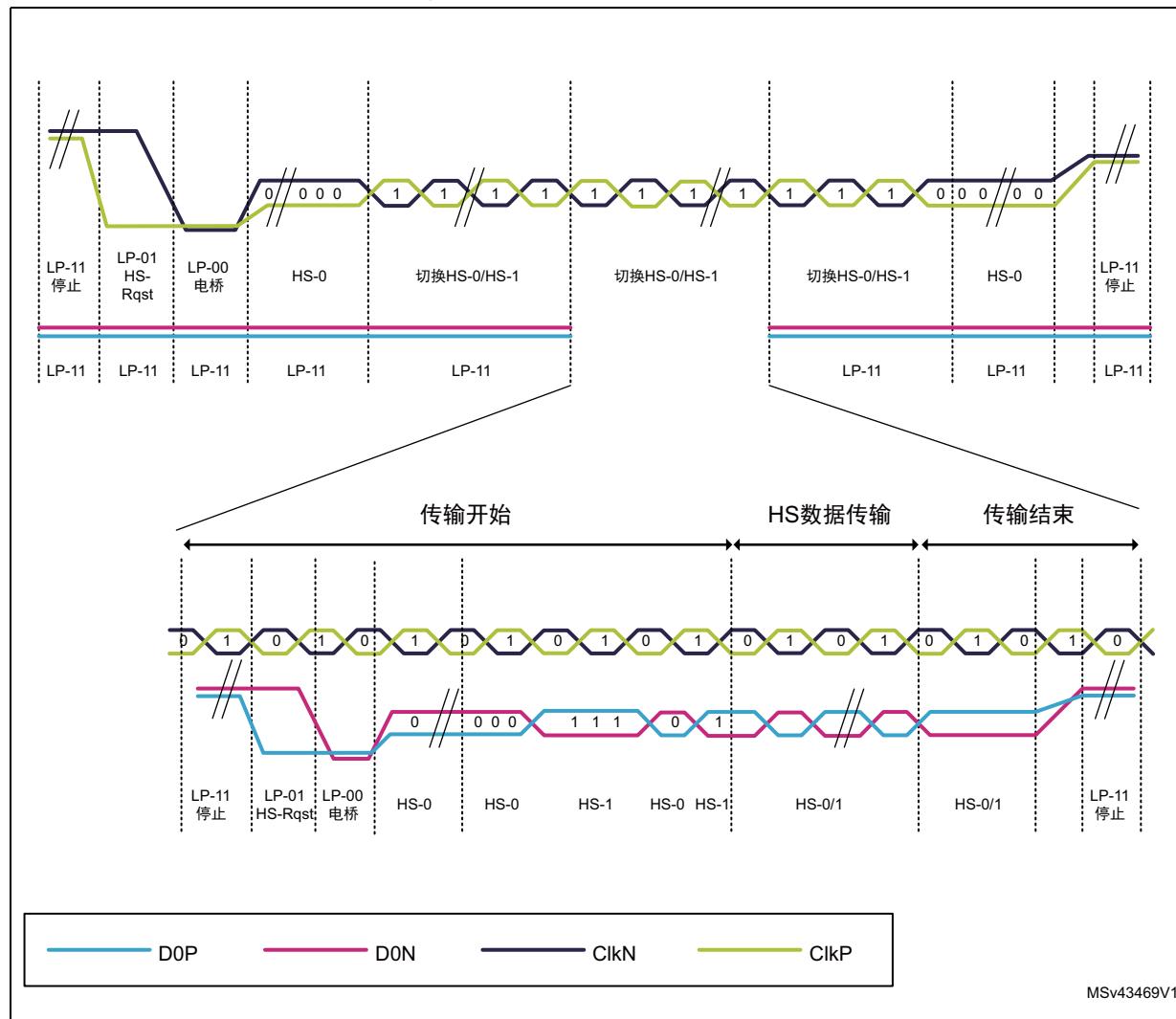
时钟通道通过退出序列（HS-0、LP-11）退出HS模式（参见图 24）。

图24. 时钟HS退出序列



高速时钟在通过数据通道发送高速数据之前启动。在高速数据通道停止后，高速时钟继续计时（参见图 25）。

图25. HS模式下时钟通道和数据通道的关系

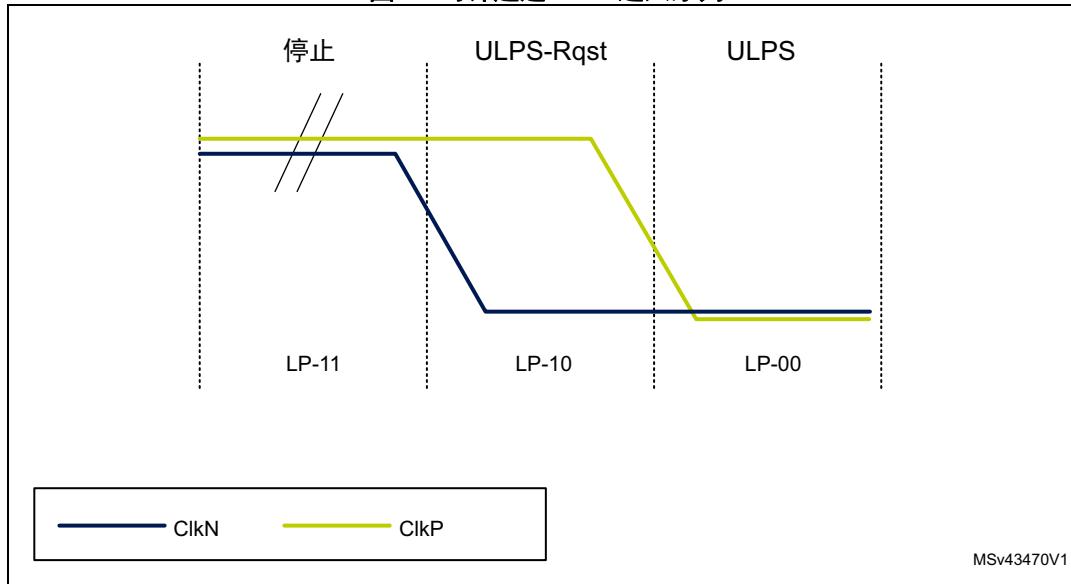


### 超低功耗状态（ULPS）

数据通道支持逸出模式，而时钟通道则不支持。但是，时钟通道支持ULPS（逸出模式的子集）。

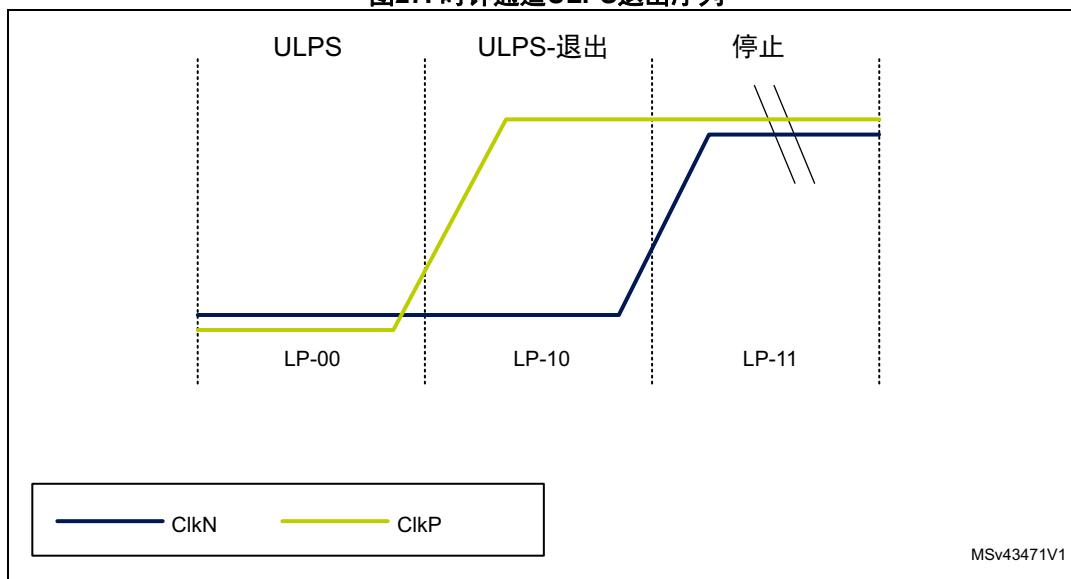
时钟通道可使用ULPS进入序列（LP-11、LP-10、LP-00）从停止状态进入ULPS，如图 26 所示。

图26. 时钟通道ULPS进入序列



时钟通道使用ULPS退出序列（LP-11、LP-10、LP-00）从ULPS状态进入LPM，如图27所示。

图27. 时钟通道ULPS退出序列



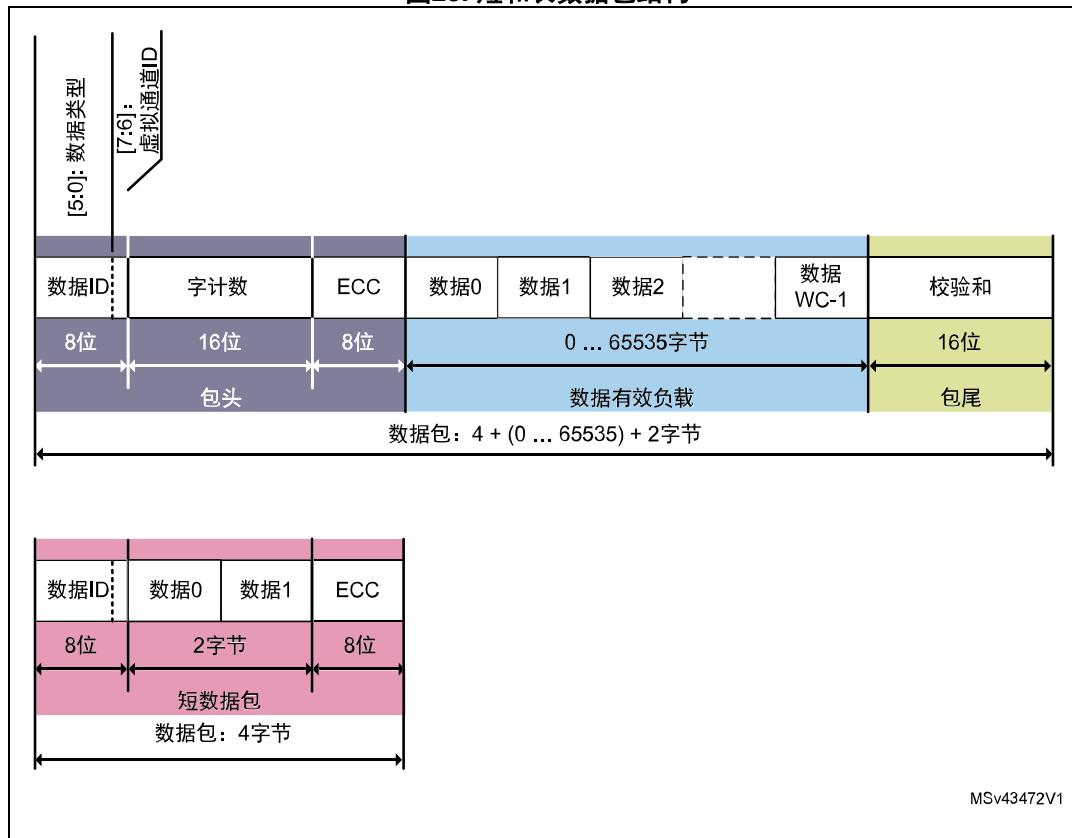
### 3.3 DSI协议

DSI是一种基于数据包的协议。并行数据和指令通过封包协议和包头封装到数据包并被解包。

#### 3.3.1 数据包结构

为底层协议通信定义了两种数据包结构：长数据包和短数据包。请参见图28：

图28. 短和长数据包结构



按字节发送数据，最低有效位优先。对于多字节字段，例如字计数和校验和，先发送最低有效字节。

图 29 显示了长数据包的示例。

图29. 长数据包示例



### 长数据包

长数据包主要用于大数据块（例如像素数据）的传输。它们由三部分组成：包头（PH）、有效负载数据和包尾（PF）。

32位数据包包头包含：

- 8位数据ID。
- 16位字计数，定义了有效负载数据的字节长度。
- 8位ECC（纠错码），用于保护数据包包头。

有效负载数据包含应用特定的数据。它主要用于传输像素数据或指令参数。通过字计数定义其长度。其长度可介于0至65535个字节之间。

包尾包含16位校验和（CS）。它由发射器计算，被接收器用来检查接收的数据是否无错误。

长数据包的最小长度为6字节，有效负载数据为0

- 4个PH字节。
- 2个PF字节。

最大长度为65541字节

- 4个PH字节
- 有效负载数据为65534字节
- 2个PF字节。

### 短数据包

短数据包由四个字节组成：

- 一个字节用于数据ID。
- 两个字节用于指令或有效负载数据。
- 一个字节用于ECC。

它们主要用于短指令传输和对时间敏感的信息（例如视频同步事件）

### 数据标识符字节

任何数据包的第一个字节均为DI（数据标识符）字节。DI字节由虚拟通道（VC）标识符和数据类型（DT）组成（参见图 30）。

图30. 数据标识符字节



#### 1. 虚拟通道标识符

DSI主机可使用包头的虚拟通道ID字段，通过标记指令或数据块服务于最多四台外设。  
VC标识数据被定向到的外设。

#### 2. 数据类型字段DT[5:0]

数据类型字段指定数据包是长还是短数据包类型。它还指定有效负载数据的数据包格式和内容。

### 数据保护（ECC和校验和）

DSI标准提供两种数据保护机制：ECC和校验和。

#### 1. 纠错码

纠错码（ECC）字节允许修正单比特位错误和检测二比特位错误。

ECC为短数据包提供稳定的保护，短数据包通常用于传输关键信息。它也为长数据包包头提供了很好的保护，长数据包包头包括数据标识符和字计数字段。

#### 2. 校验和

长数据包中的有效负载数据通过16位校验和提供保护，校验和只能指示有效负载中存在一个还是更多错误。

### 3.3.2 传输结束（EoT）数据包

D-PHY使用EoT序列发送HS传输结束信号，但此序列可能被接收器解读为有效数据。

为增加协议的稳定性，DSI发射器可以发送EoTp（传输结束数据包）以通知高速传输结束。

此机制以增加开销（每次传输增加四个额外字节）为代价提供更稳定的环境。

[图 31](#)显示了使能EoT数据包的HS传输示例。

图31. 使用EoT数据包的HS传输

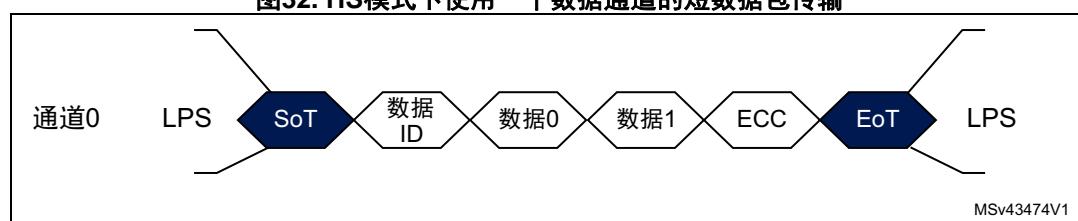


### 3.3.3 数据包传输模式

短和长数据包都可以在HS或LP模式下传输。此外，在HS模式下发送的数据包可以在可用数据通道间进行分割。

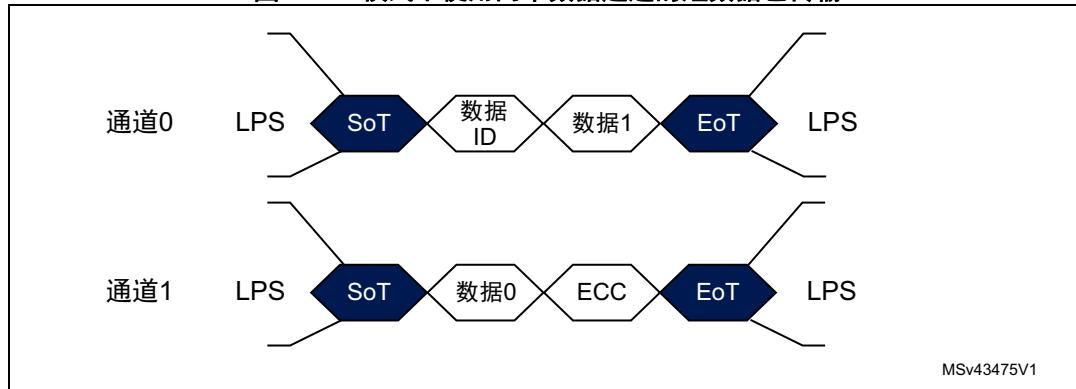
[图 32](#)显示了HS模式下使用一个数据通道的短数据包传输。

图32. HS模式下使用一个数据通道的短数据包传输



[图 33](#)显示了HS模式下使用两个数据通道的短数据包传输。

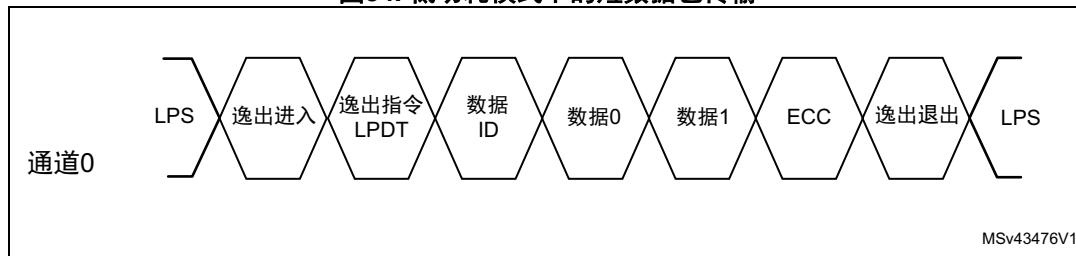
图33. HS模式下使用两个数据通道的短数据包传输



[图 34](#)显示了LP模式下的短数据包传输。

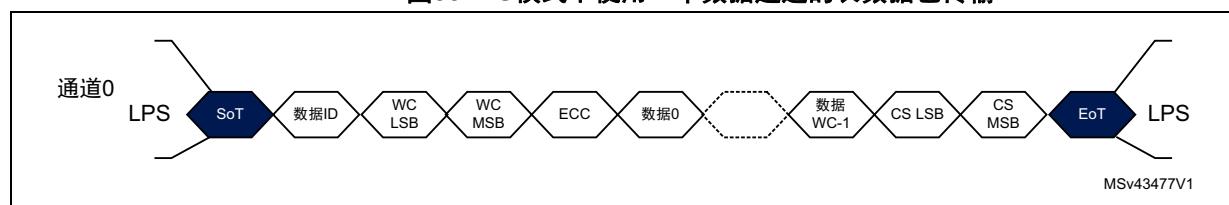
注：  
在LP模式下，只有数据通道0用于传输。

图34. 低功耗模式下的短数据包传输



[图 35](#)显示了HS模式下使用一个数据通道的长数据包传输。

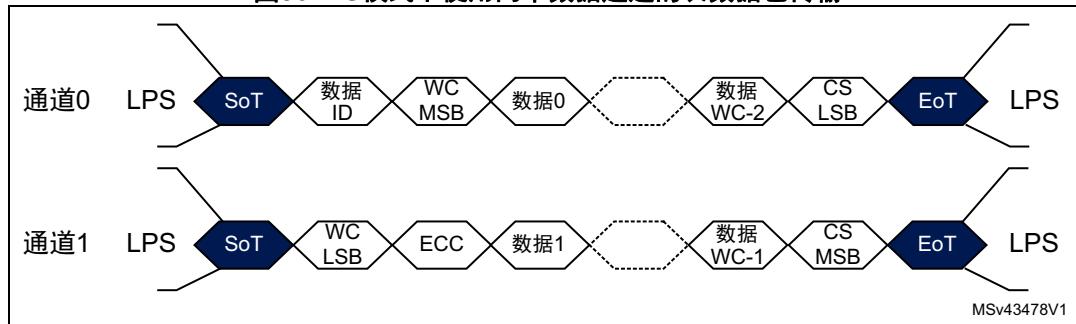
图35. HS模式下使用一个数据通道的长数据包传输



在HS模式下，只能使用多个通道进行数据传输。

[图 36](#)显示了HS模式下使用两个数据通道的长数据包传输。

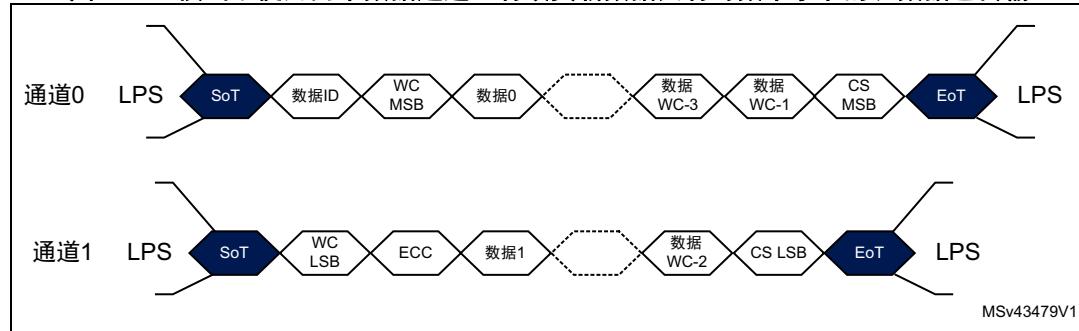
图36. HS模式下使用两个数据通道的长数据包传输



如果有效负载数据的数量不是通道数的整数倍，则一些通道可能先于其他通道完成高速传输，提早一个周期（字节）发送了EoT。

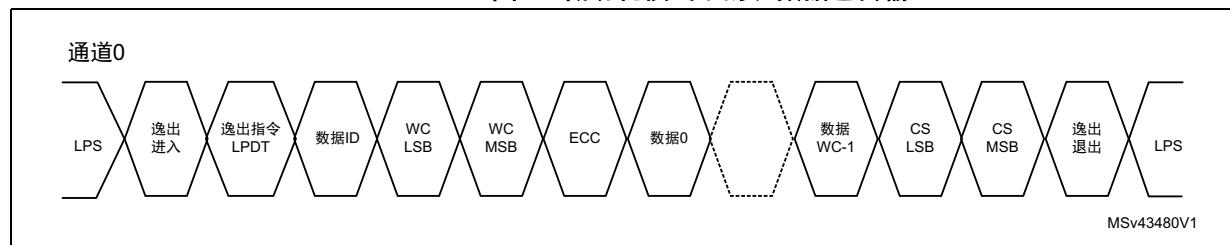
[图 37](#)显示了使用两个数据通道的示例。在本例中，有效负载数据具有奇数个字节，导致数据通道1在数据通道0之前完成高速传输。

**图37. HS模式下使用两个数据通道、有效负载数据具有奇数个字节的长数据包传输**



[图 38](#)显示了LP模式下的长数据包传输。

**图38. 低功耗模式下的长数据包传输**



### 3.3.4 主机至显示屏数据类型

主机至显示屏数据类型可以是短数据包或长数据包。它们可以是视频或指令数据类型。[表 6](#)中显示了主机至显示屏数据类型：

**表6. 主机至显示屏数据类型**

数据类型	说明	Packet	DSI模式
0x01	同步事件, V同步开始	短	视频
0x11	同步事件, V同步结束	短	视频
0x21	同步事件, H同步开始	短	视频
0x31	同步事件, H同步结束	短	视频
0x08	传输结束数据包 (EoTp)	短	二者兼有
0x02	颜色模式 (CM) 关闭指令	短	视频
0x12	颜色模式 (CM) 打开指令	短	视频
0x22	关闭外设指令	短	视频
0x32	开启外设指令	短	视频
0x03	通用短写指令, 无参数	短	指令
0x13	通用短写指令, 1个参数	短	指令

表6. 主机至显示屏数据类型（续）

数据类型	说明	Packet	DSI模式
0x23	通用短写指令, 2个参数	短	指令
0x04	通用读指令, 无参数	短	指令
0x14	通用读指令, 1个参数	短	指令
0x24	通用读指令, 2个参数	短	指令
0x05	DCS短写指令, 无参数	短	指令
0x15	DCS短写指令, 1个参数	短	指令
0x06	DCS读指令, 无参数	短	指令
0x37	设置最大返回数据包大小	短	指令
0x09	空包, 无数据	长	视频
0x19	消隐数据包, 无数据	长	视频
0x29	通用长写指令	长	指令
0x39	DCS长写指令/Write_LUT	长	指令
0x0C	松散封装像素流 20位YCbCr, 4:2:2格式	长	视频
0x1C	封装像素流 24位YCbCr, 4:2:2格式	长	视频
0x2C	数据包像素流 16位YCbCr, 4:2:2格式	长	视频
0x0D	数据包像素流 30位RGB, 10-10-10格式	长	视频
0x1D	数据包像素流 36位RGB, 12-12-12格式	长	视频
0x3D	数据包像素流 12位YCbCr, 4:2:0格式	长	视频
0x0E	数据包像素流 16位RGB, 5-6-5格式	长	视频
0x1E	数据包像素流 18位RGB, 6-6-6格式	长	视频
0x2E	松散封装像素流 18位RGB, 6-6-6格式	长	视频
0x3E	封装像素流 24位RGB, 8-8-8格式	长	视频

## 视频模式数据类型

视频模式数据类型主要用于传输同步事件和像素数据。同步事件和像素数据包以HS模式发送，因为它们传输的是时间关键信息。

其他视频模式数据类型用于颜色模式和关闭控制。它们可以在HS或LP模式下发送。

注：

在下面显示的所有示例中， $VC = 0$ 。

### 1. 关机和颜色模式

颜色模式指令是允许在正常模式和低颜色模式之间切换显示模块的短数据包。低频颜色模式用于节能（参见图 39）。

图39. 颜色模式视频数据包

颜色模式关闭				颜色模式开启			
数据ID	数据0	数据1	ECC	数据ID	数据0	数据1	ECC
0x02	0x00	0x00	0x0B	0x12	0x00	0x00	0x18

MSv43481V1

关机和开启指令是用于关闭或开启显示模块的短数据包（参见图 40）。

图40. 关机/开启视频数据包

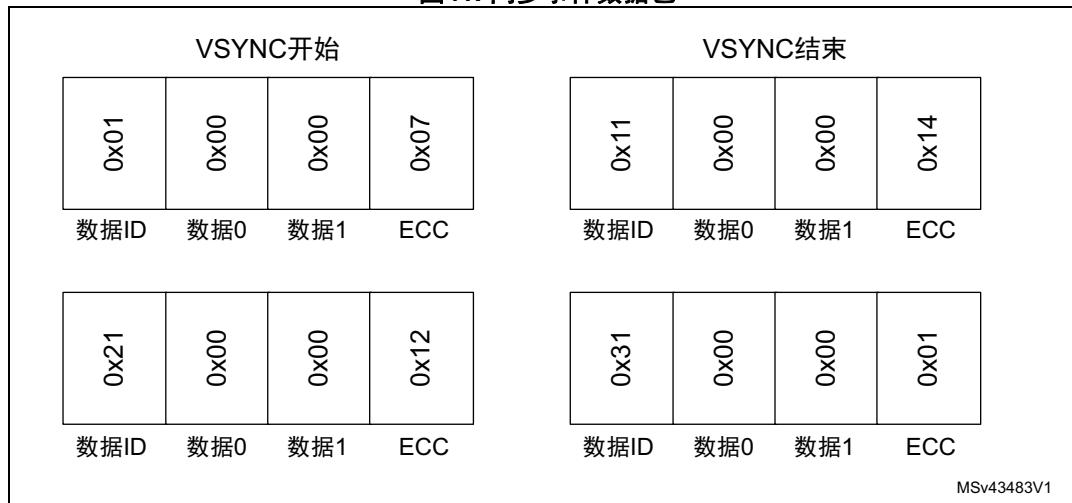
关闭外设				开启外设			
数据ID	数据0	数据1	ECC	数据ID	数据0	数据1	ECC
0x22	0x00	0x00	0x1E	0x32	0x00	0x00	0x0D

MSv43482V1

### 2. 同步事件

同步事件通过短数据包发送，因为短数据包更适合传输时间精确的信息（参见图 41）。

图41. 同步事件数据包



### 3. 封装像素流

封装像素流（PPS）数据包是以像素的格式向视频模式显示模块传输RGB图像数据的长数据包。数据包包含DI字节、2字节WC、ECC字节、长度WC字节的有效负载和2字节校验和。

DSI协议定义了不同颜色编码的封装像素流：

- 封装像素流，16位格式。请参见图 42。
- 封装像素流，18位格式。请参见图 43。
- 松散封装像素流，18位格式。请参见图 44。
- 封装像素流，24位格式。请参见图 45。

图42. 封装像素流，16位格式，数据类型（0x0E）

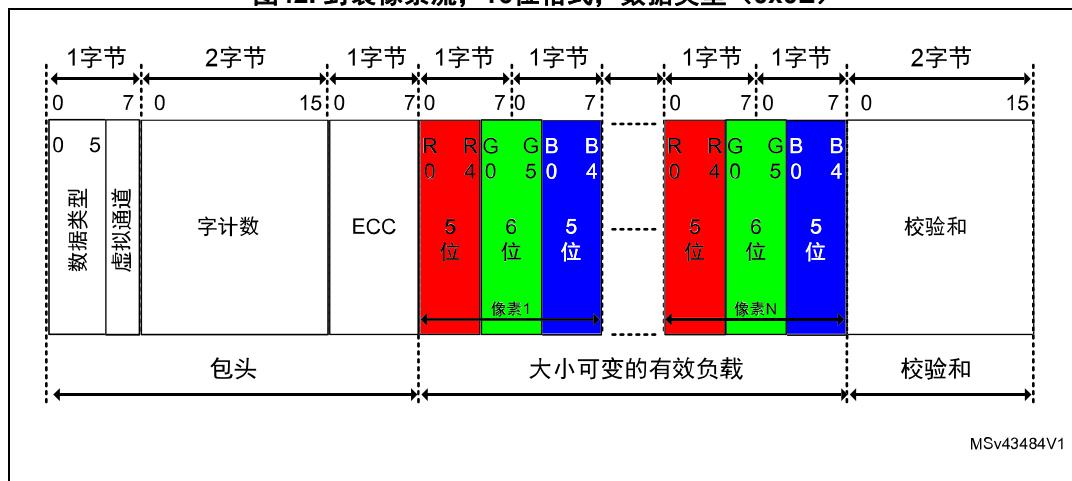


图43. 封装像素流，18位格式，数据类型= (0x1E)

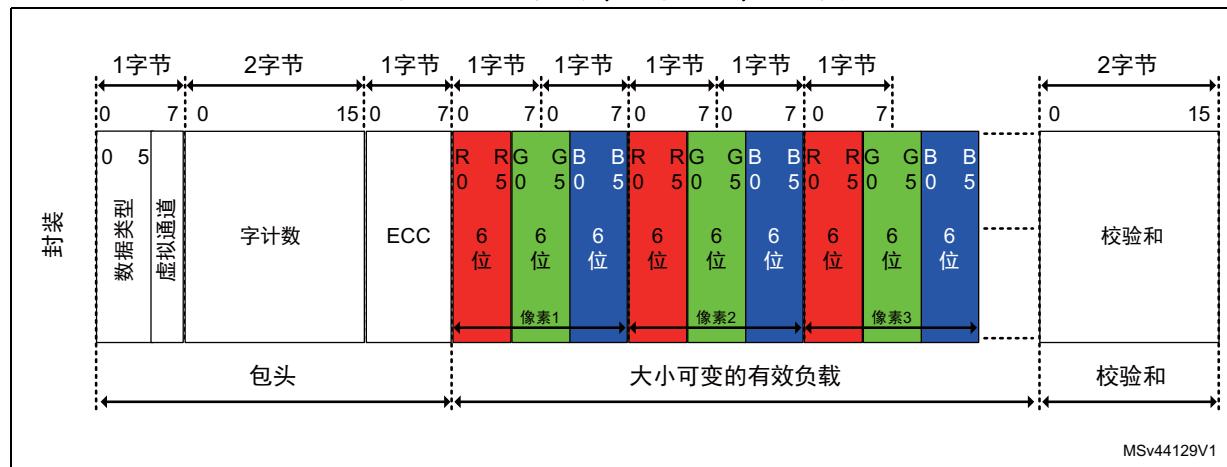


图44. 松散封装像素流，18位格式，数据类型= (0x2E)

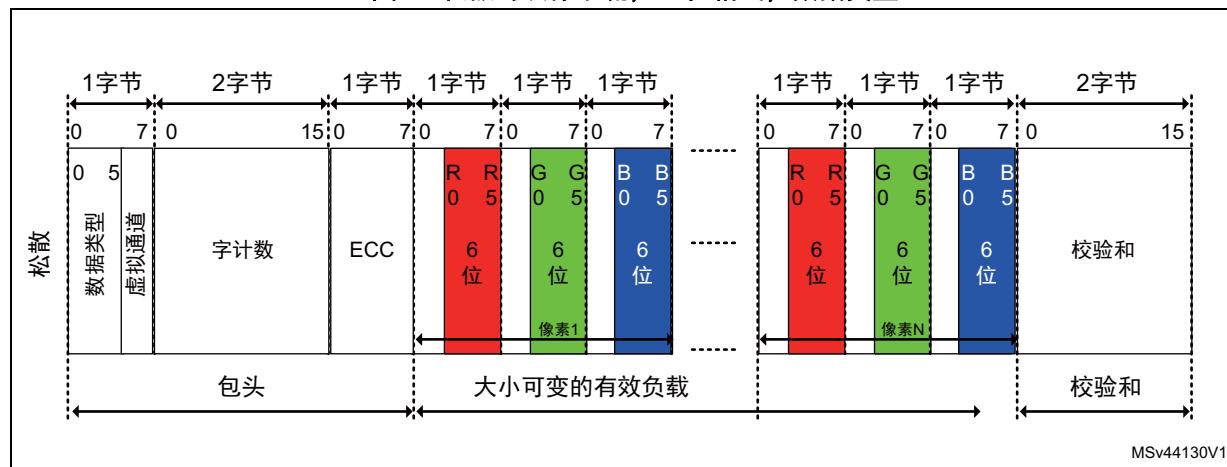
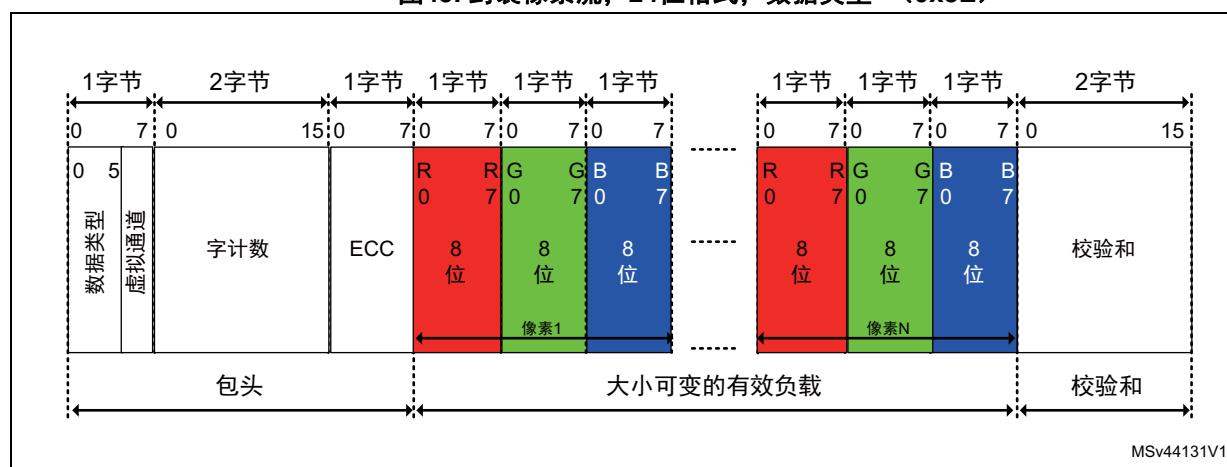


图45. 封装像素流，24位格式，数据类型= (0x3E)



### 指令模式数据类型

指令模式数据类型以HS或LP模式发送。它们用于对/从显示屏寄存器和帧缓冲器执行写入/读取。

#### 1. 通用指令

有三种类型的通用指令：

- 通用短写指令。请参见图 46。
- 通用长写指令。请参见图 47。
- 通用读指令。请参见图 48。

图46. 通用短写指令

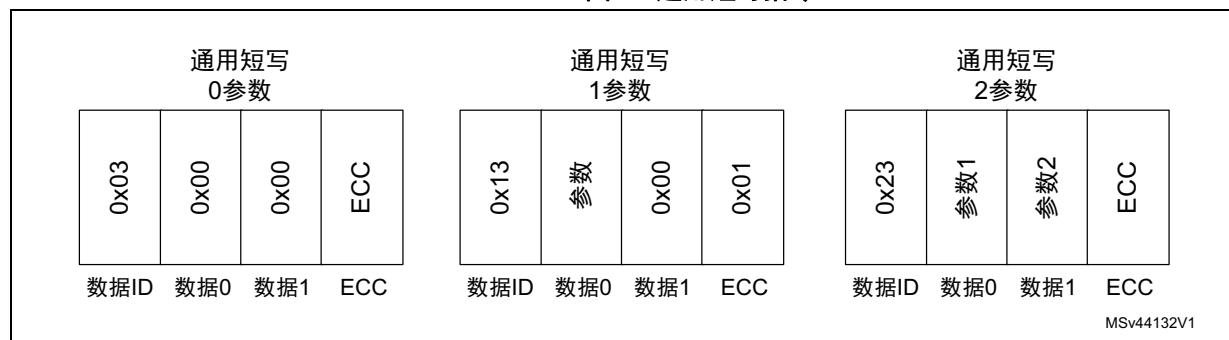
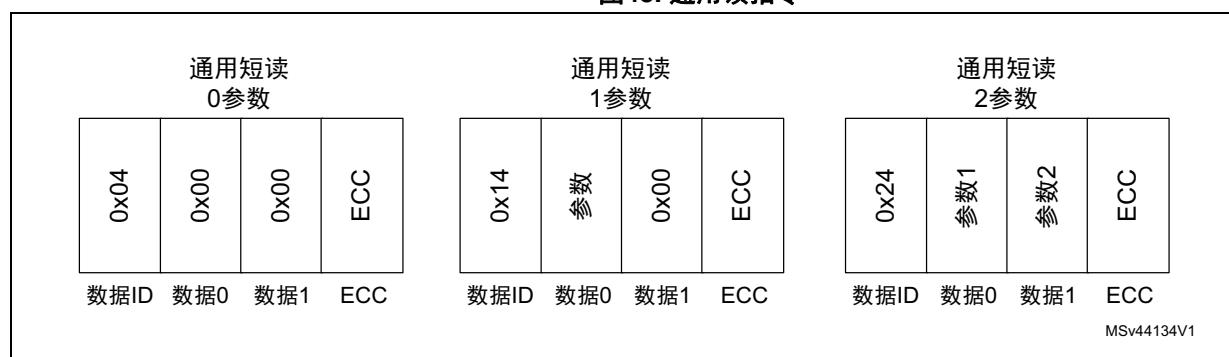


图47. 通用长写指令



图48. 通用读指令



## 2. DCS指令

DCS是MIPI联盟定义的标准化指令集，旨在用于指令模式显示屏。DCS指令列于[表 7](#)中。

包含0或1个参数的DCS指令使用短数据包发送，包含1个以上参数的DCS指令使用长数据包发送。

有三种类型的DCS指令：

- DCS短写指令。请参见[图 49](#)。

使用短数据包以DT 0x05发送无参数DCS短写指令。DCS指令索引位于数据0字段。数据1字段未使用，它被置为0。

具有1个参数的DCS短写指令采用DT 0x15。DCS指令索引位于数据0字段，参数位于数据1字段。

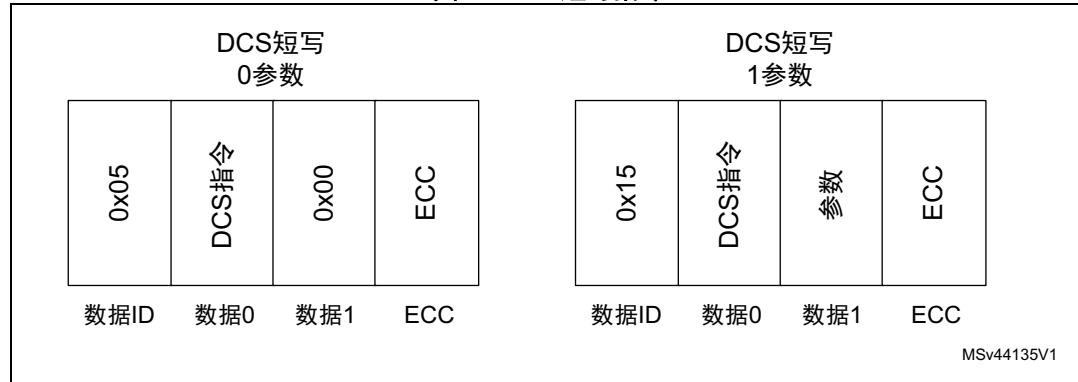
- DCS长写指令。请参见[图 50](#)。

- DCS读取。请参见[图 51](#)。

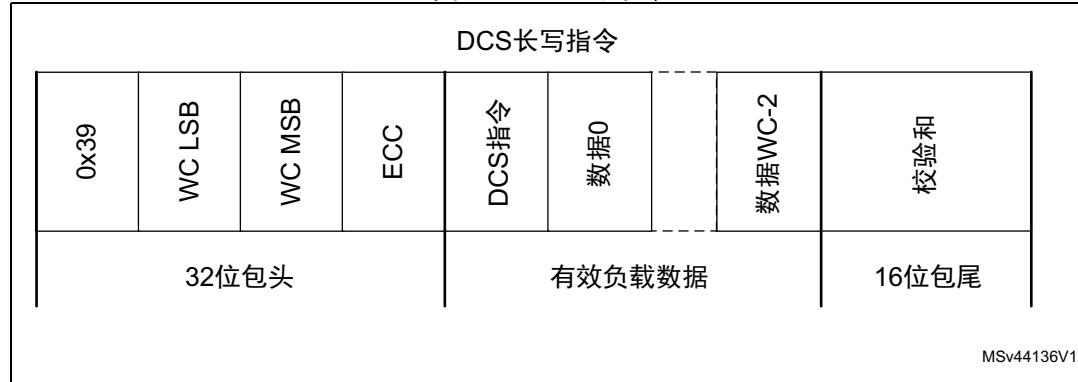
读指令后必须有BTA，用于将总线所有权交给显示屏，以便显示屏发送响应。

响应可以是DCS短或长读响应（参见[表 8](#)）。

**图49. DCS短写指令**



**图50. DCS长写指令**



**图51. DCS读指令**

下面的[表 7](#)显示了DCS指令列表。

表7. DCS指令列表

指令	十六进制码	说明
enter_idle_mode	39h	显示面板上使用部分色深。
enter_invert_mode	21h	显示的图像颜色反转。
enter_normal_mode	13h	整个显示区用于图像显示。
enter_partial_mode	12h	部分显示区用于图像显示。
enter_sleep_mode	10h	显示面板的电源断开。
exit_idle_mode	38h	显示面板上使用完整色深。
exit_invert_mode	20h	显示的图像颜色不反转。
exit_sleep_mode	11h	显示面板的电源接通。
get_3D_control	3Fh	获取显示模块三维模式。
get_address_mode	0Bh	获取从主机到显示模块和从帧存储器到显示设备的数据传输顺序。
get_blue_channel	08h	获取位于 (0, 0) 的像素的蓝色分量。
get_diagnostic_result	0Fh	获取外设自诊断结果。
get_display_mode	0Dh	从外设获取当前显示模式。
get_green_channel	07h	获取位于 (0, 0) 的像素的绿色分量。
get_pixel_format	0Ch	获取当前像素格式。
get_power_mode	0Ah	获取当前功耗模式。
get_red_channel	06h	获取位于 (0, 0) 的像素的红色分量。
get_scanline	45h	获取当前扫描行。
get_signal_mode	0Eh	获取显示模块信令模式。
nop	00h	无操作。
read_DDB_continue	A8h	从最后一个读取位置继续读取DDB。
read_DDB_start	A1h	从提供的位置读取DDB。
read_memory_continue	3Eh	在最后一个read_memory_continue或read_memory_start后继续从外设读取图像数据。
read_memory_start	2Eh	从set_column_address和set_page_address提供的位置开始，从外设向主机处理器接口传输图像数据。
set_3D_control	3Dh	显示面板上使用三维模式。
set_address_mode	36h	设置从主机到显示模块和从帧存储器到显示设备的数据传输顺序。
set_column_address	2Ah	设置列范围。
set_display_off	28h	消隐显示设备。
set_display_on	29h	在显示设备上显示图像。
set_gamma_curve	26h	选择显示设备使用的Gamma曲线。
set_page_address	2Bh	设置页面范围。

表7. DCS指令列表（续）

指令	十六进制码	说明
set_partial_columns	31h	定义显示设备上部分显示区域中的列数。
set_partial_rows	30h	定义显示设备上部分显示区域中的行数。
set_pixel_format	3Ah	定义界面上使用的每像素位数。
set_scroll_area	33h	定义显示设备上的垂直滚动和固定区域。
set_scroll_start	37h	定义垂直滚动起点。
set_tear_off	34h	同步信息不从显示模块发送至主机处理器。
set_tear_on	35h	在VFP开始时，同步信息从显示模块发送至主机处理器。
set_tear_scanline	44h	当显示设备刷新到提供的扫描行时，同步信息从显示模块发送至主机处理器。
set_vsync_timing	40h	Set VSYNC timing.
soft_reset	01h	软件复位。
write_LUT	2Dh	使用提供的数据填写外设查找表。
write_memory_continue	3Ch	从最后一个写入位置开始，从主机处理器接口向外设传输图像信息。
write_memory_start	2Ch	从set_column_address和set_page_address提供的位置开始，从主机处理器向外设传输图像数据。

### 3. 设置最大返回数据包大小

“*set maximum return packet size*”指令允许主机处理器对来自外设的响应数据包的大小进行限制。

最大返回数据包大小的初始值为一个字节。为了获得一个以上的字节，主机必须在发送读指令之前发送该指令，以所需数据包大小为参数。

注：最大返回数据包大小不得超过主机读FIFO的大小，以免发生上溢。DSI主机读FIFO的大小为32 x 32位字。

### 3.3.5 显示屏至主机数据类型

所有指令模式系统均支持反向通信。另一方面，对于视频模式系统而言，对反向通信的支持是可选的。

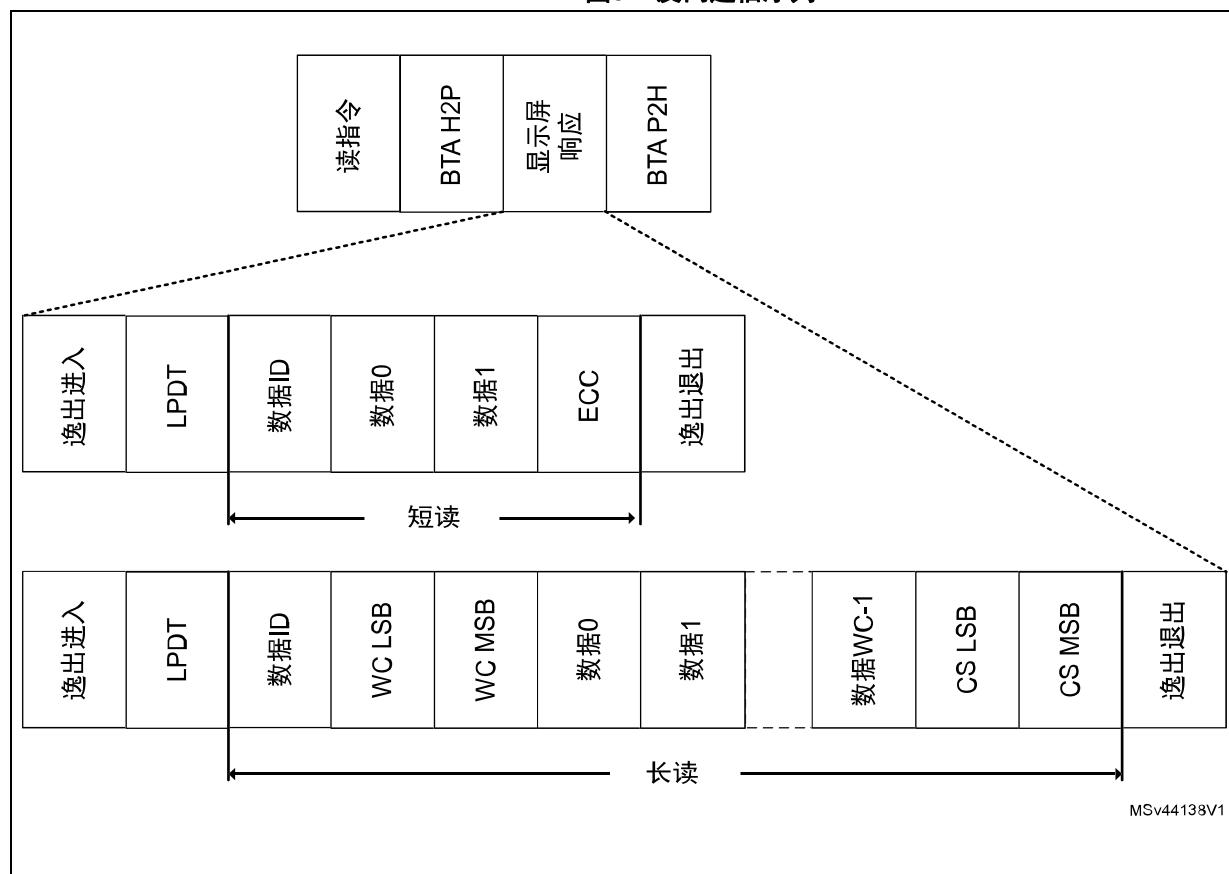
与正向传输一样，显示屏至主机的传输使用相同的短和长数据包结构。

显示屏至主机的通信只能在主机使用BTA序列将总线所有权给予显示屏后开始。

在发送响应后，显示屏通过BTA序列将总线所有权还给主机。

[图 52](#)显示了反向通信序列的示例。请注意，显示屏至主机的传输发生在LP模式下，使用数据通道0。

图52. 反向通信序列



下面的[表 8](#)列出了显示屏至主机的数据包类型。

表8. 显示屏至主机数据类型

数据类型	说明	Packet
0x02	回应和报告错误	短
0x08	传输结束数据包 (EoTp)	短
0x11	通用短读响应 返回1字节	短
0x12	通用短读响应 返回2字节	短
0x1A	通用长读响应	长
0x1C	DCS长读响应	长

表8. 显示屏至主机数据类型（续）

数据类型	说明	Packet
0x21	DCS短读响应 返回1字节	短
0x22	DCS短读响应 返回2字节	短

### 3.3.6 视频模式接口时序

视频模式外设要求实时传输像素数据。DSI支持多种格式的视频模式数据传输：

- 使用同步脉冲的非突发模式：使外设能够精确地重建原始视频时序，包括同步脉宽。请参见图 53。如果没有足够的时间切换至LP模式，则使用消隐数据包取代LP模式下的时间间隔。
- 使用同步事件的非突发模式：类似于上述模式，但无需同步脉宽的精确重建，因此取代了单一同步事件。请参见图 54。
- 突发模式：RGB像素数据包经过时间压缩，在扫描行期间留下更多时间用于LP模式（节能）或用于将其他传输复用到DSI链路上。请参见图 55。

注：

为了精确的时间重建，应考虑数据包开销（包括数据ID、ECC和校验和字节）。

DSI主机支持全部三种视频模式数据包序列。DSI显示屏支持至少一种上述模式。

图53. 使用同步脉冲的非突发模式

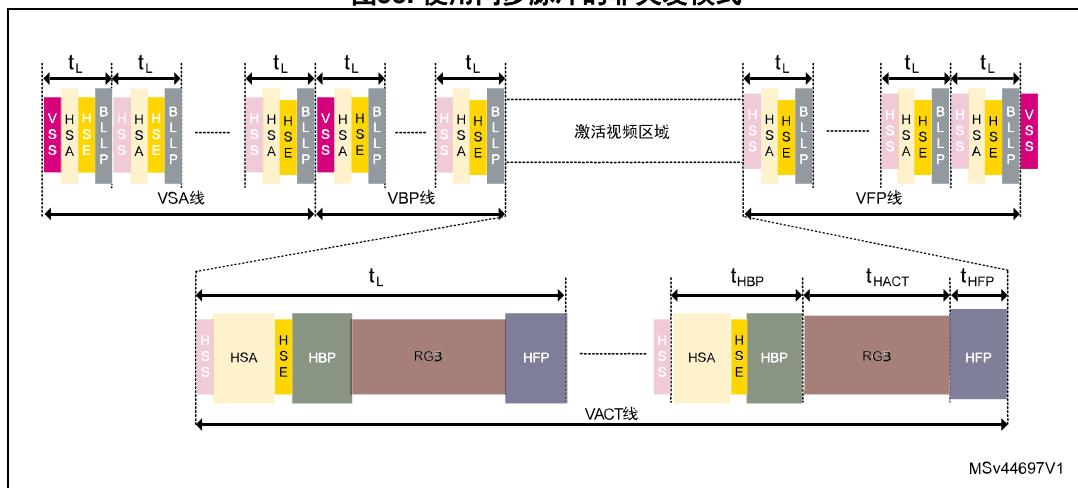


图54. 使用同步事件的非突发模式

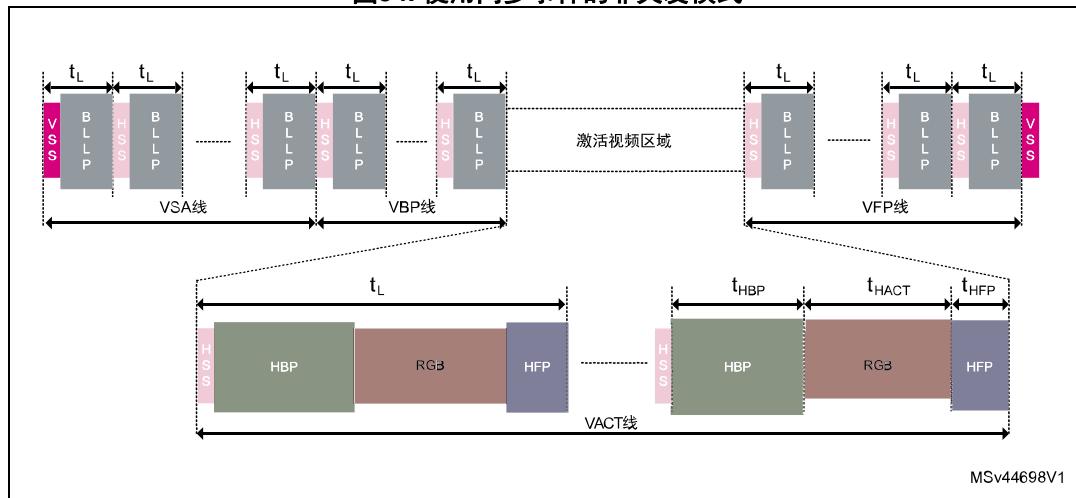
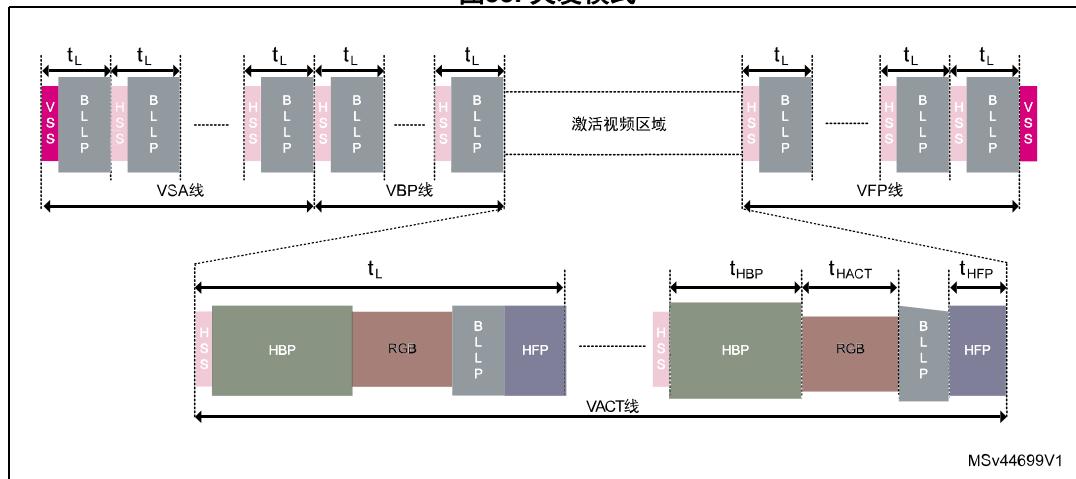


图55. 突发模式



### 3.3.7 指令模式下的撕裂效应信号

撕裂效应（TE）是一种视觉失真，即显示设备在一个屏幕画面上显示来自两个或更多帧的信息。当主机执行的显示屏GRAM更新与显示屏的扫描进度不同步时，将产生伪影。

撕裂效应信令是一种同步机制，允许将显示屏侧的时间事件通知主机。它用在显示屏具有自己的控制器并使用其内部GRAM刷新显示的指令模式系统中。

在指令模式系统中，显示屏在每次到达特定扫描行时通过TE信令通知主机。此操作使主机知道何时更新显示屏GRAM不会导致撕裂效应。

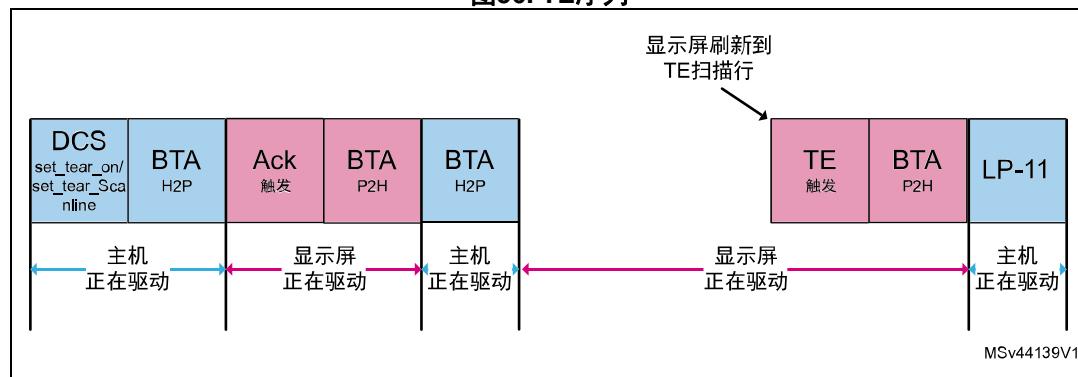
当需要TE报告时，DSI主机发送set\_tear\_on或set\_tear\_scanline DCS指令，然后执行双BTA（总线转向）流程将总线所有权交给显示屏。

需要双BTA，因为显示屏的DSI协议层以回应触发信号或错误报告来响应第一个BTA，然后通过第二个BTA将总线所有权还给主机（因为它不解读DCS指令）。

为允许TE报告，主机必须执行第二BTA以将总线所有权交给显示屏。在到达扫描行后，显示屏立即以TE触发信号（01011101）做出响应。

[图 56](#)显示了撕裂效应序列。

图56. TE序列



1. BTA H2P是由主机发起的BTA流程，用于将总线所有权交给外设。BTAP2H是由外设发起的BTA流程，用于将总线所有权交给主机。

Set\_tear\_scanline以与set\_tear\_on相同的方式使能TE报告，但Set\_tear\_scanline还定义应发生TE报告的扫描行。[图 57](#)中显示了Set\_tear\_scanline DCS长数据包，当显示到533行时需要TE报告（就本例而言）。

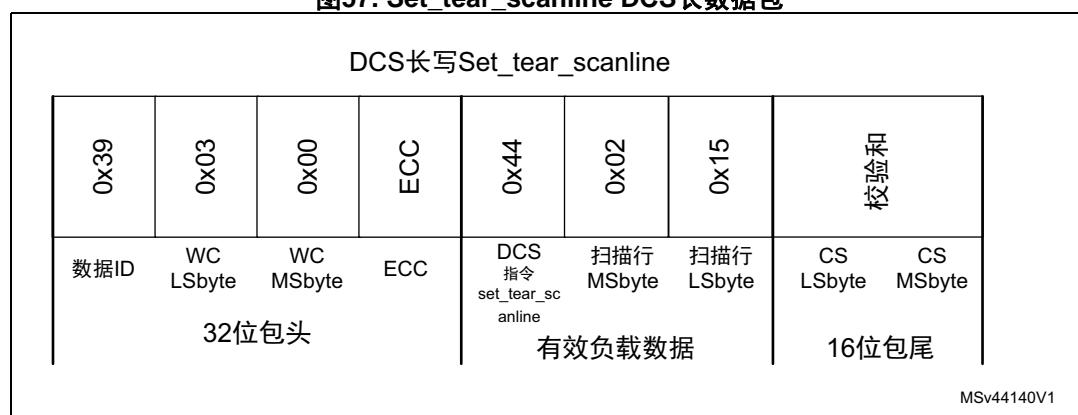
set\_tear\_scanline通常用于初始化阶段的TE扫描行编程，然后，主机在每次需要使能TE报告时发送set\_tear\_on。

Set\_tear\_scanline指令是长DCS写指令。需要两个参数来定义TE扫描行。

注：

使用两个字节发送TE扫描行，最高有效字节优先。

图57. Set\_tear\_scanline DCS长数据包



## 4 DSI主机描述

本节描述STM32 DSI主机系统级架构和工作模式，并提供一些根据应用需要选择合适工作模式的指南。

### 4.1 DSI系统级架构

DSI主机与LTDC深度集成。它依赖于LTDC获取像素数据并提供视频同步信号。

DSI主机有两种系统接口：帧缓冲器

- LTDC接口。  
此接口允许DSI主机从LTDC捕获像素数据和视频同步信号，并将它们封装到DSI数据包中。
  - 当处于视频模式时，为视频数据包（视频同步事件、像素数据包流）。
  - 当处于适配指令模式时，为memory\_write\_start（WMS）和memory\_write\_continue（WMC）DCS指令。
- APB接口。  
此接口用于DCS和通用指令模式数据包的传输。这些数据包使用APB寄存器访问进行构建。  
它可以在适配指令模式或视频模式下并发访问，以便将DCS或通用数据包传输给显示屏。

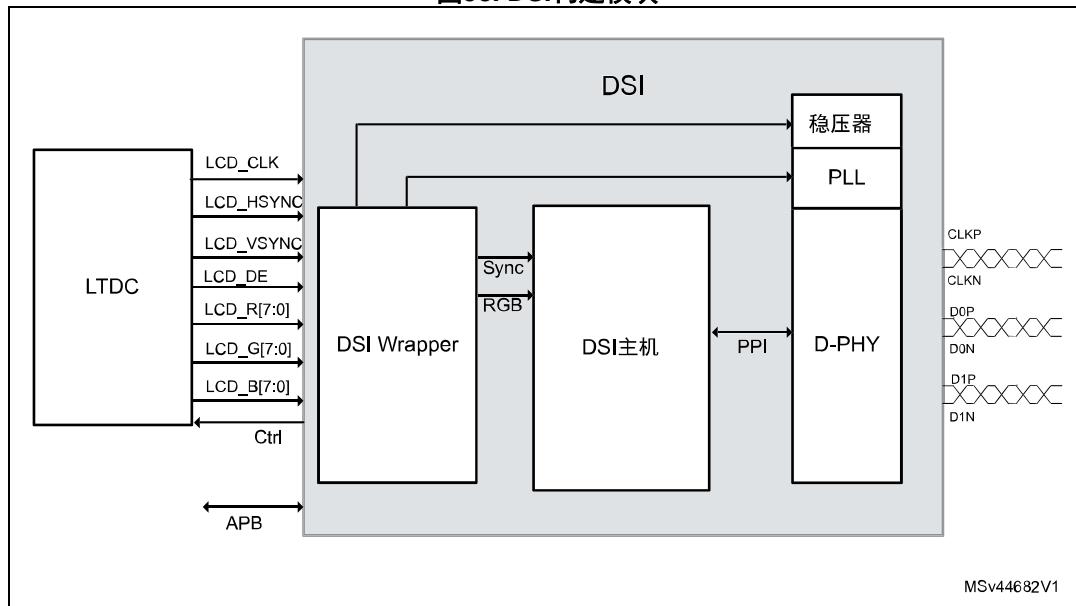
#### DSI主机构建模块

DSI系统构建模块为：

- DSWrapper：将LTDC与DSI主机互连，以便在视频或适配指令模式下重定向像素数据和视频控制信号。  
DSI Wrapper还控制DSI调节器、DSI PLL和D-PHY的一些特定功能。
- DSI主机控制器：构建长或短DSI数据包并生成相应的ECC和CRC代码。数据包按字节发送至D-PHY用于序列化。  
如果使用多个通道，DSI主机还在可用的D-PHY通道之间执行数据分割。
- D-PHY：将来自DSI主机控制器的数据串行化并通过串行链路发送。
- 内部PLL：生成D-PHY使用的HS时钟。
- 内部调节器：为PLL和D-PHY提供1v2电源。

[图 58](#)所示为DSI构建模块。

图58. DSI构建模块



## 4.2 工作模式

STM32 DSI主机支持DSI规范1.1版中定义的所有工作模式。它支持指令模式和视频模式序列（突发模式、使用同步脉冲的非突发模式、使用同步事件的非突发模式）。

STM32 DSI主机还支持增强的指令模式，以便使用集成GRAM的显示屏实现优化操作。

可用于将图形数据传输至显示屏的三种工作模式为：

- 视频模式。  
它用于通过高速链路进行由LTDC直接生成的RGB数据和相关同步信号（Vsync、Hsync）的流传输。  
LTDC接口捕获数据和同步信号，并将它们传输至FIFO接口。  
接口上存在两种不同数据流：视频控制信号和像素数据。  
DSI主机使用两种数据流构建DSI视频数据包，然后通过DSI链路传输数据包。  
流传输在DSI主机和LTDC使能后立即开始。这种连续刷新是连接无图形RAM的显示屏的最佳方式。
- 适配指令模式。  
此模式是连接具有自己的内部图形RAM的显示屏的最佳方式。DSI主机只捕获来自LTDC的一个完整帧并将其转换为一系列DCS写指令以更新显示屏图形RAM。如果在DSI主机中置位了控制位，将自动完成此独热刷新。
- APB指令模式  
此模式用于通过高速链路发送用于配置的指令，与使用SPI时一样。使用DSI主机APB接口发出指令。

## 4.2.1 视频模式

DSI主机视频模式支持DSI规范中定义的三种工作模式。

视频模式用于没有集成能够保留整个帧的图形RAM的显示屏。

视频模式显示屏没有内部控制器，它依赖于DSI主机提供连续像素流。

### 使用同步脉冲的非突发模式

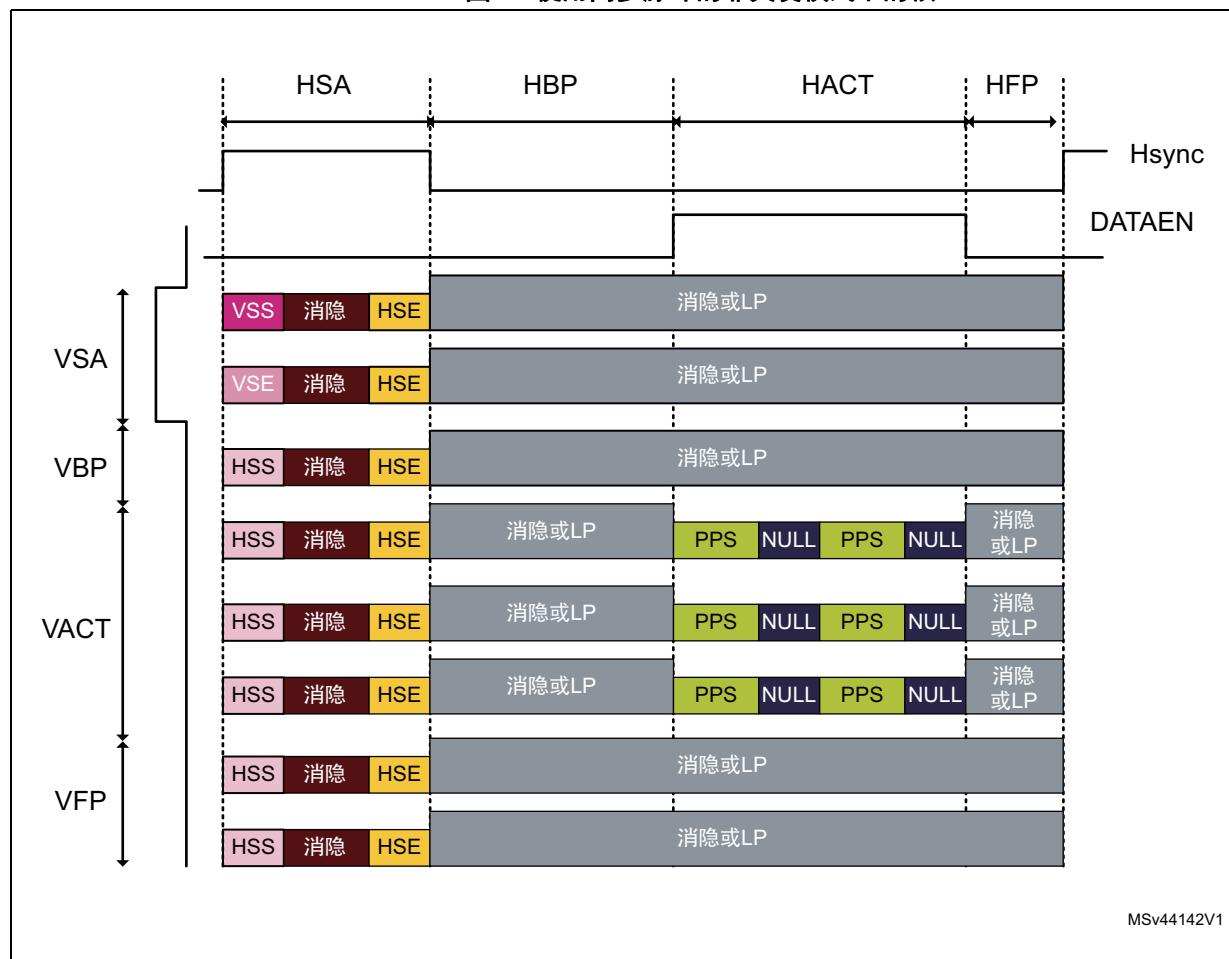
在非突发模式下，处理器使用DSI主机的分区属性将视频行传输划分为若干像素块，选择性地将它们与空包交替排列。这样做的目的是使LTDC接口输入像素带宽与DSI链路带宽相匹配。

在该模式下，主机控制器和显示屏无需存储整行像素数据。此模式只要求存储一个视频数据包的内容，最大限度降低了存储要求。

此模式使外设能够精确地重建原始视频时序，包括同步脉宽。

[图 59](#)显示了使用同步脉冲的非突发模式下的典型帧。

图59. 使用同步脉冲的非突发模式下的帧



HSA (Hsync激活) 周期由HSS (Hsync开始)、消隐和HSE (Hsync结束) 数据包组成。此区域中的数据包以高速模式传输，HSA周期内链路不进入LP。

DSI主机自动计算使HSS和HSE数据包之间的时间与HSA周期相匹配所需的消隐数据包大小。

当DSI主机检测到Vsync上升沿（假设Vsync信号高电平有效）时，DSI主机通过Vsync开始（VSS）数据包而不是HSS数据包启动HSA周期。

当检测到Vsync下降沿时，DSI主机发送Vsync结束（VSE）数据包而不是HSS数据包标记VSA周期结束。

Vsync激活（VSA）区域内的其他行以HSS数据包开头。

在垂直激活（VACT）周期以外，链路在HSA周期后进入LP模式，直至水平行结束。

在VACT区域内，DSI主机传输上述HSA周期，然后链路进入LP或发送消隐数据包，持续的时间周期等于水平后沿（HBP）周期。然后，DSI主机分一块或多块发送PPS（封装像素流）数据包，最后的空包用于将像素传输时间与水平激活（HACT）周期相匹配。

一旦HACT周期结束，DSI主机进入LP或发送消隐数据包，持续时间等于水平前沿（HFP）周期。

注：当区域标记有消隐或LP时，表示DSI主机可以在整个周期时间内发送高速消隐数据包，或者DSI主机可以将链路置于低功耗模式。

DSI主机能够计算为了匹配周期时序，消隐数据包中需要的字节数。

如果是低功耗模式，DSI PHY将在进入低功耗模式之前发出传输结束序列。然后，在开始新的高速传输之前，DSI PHY发出传输开始序列。

DSI主机需要一些来自用户的输入，以便了解低功耗转换（EoT和SoT序列）的开销，以及在特定周期内是否可以切换至低功耗模式。

[图 60](#)显示了消隐或低功耗区域的两种可能方案。

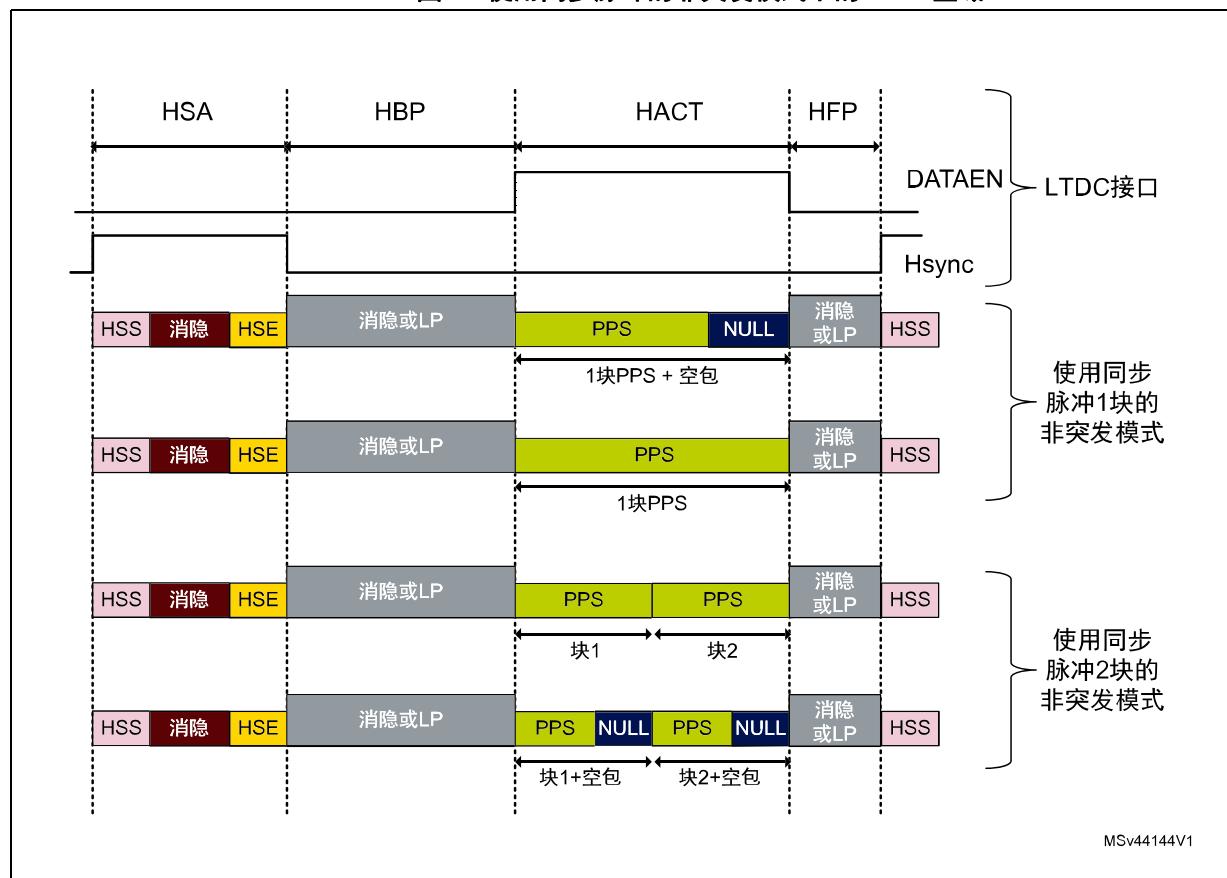
图60. 消隐或LP的定义



在VACT周期内，有许多种不同的水平行配置。

[图 61](#)显示了VACT区域行可能有的不同配置。

图61. 使用同步脉冲的非突发模式下的VACT区域



链路可以在HBP和HFP周期内进入LP，或者DSI主机在该周期内没有足够时间进行HS和LP模式之间的转换时发送消隐数据包。

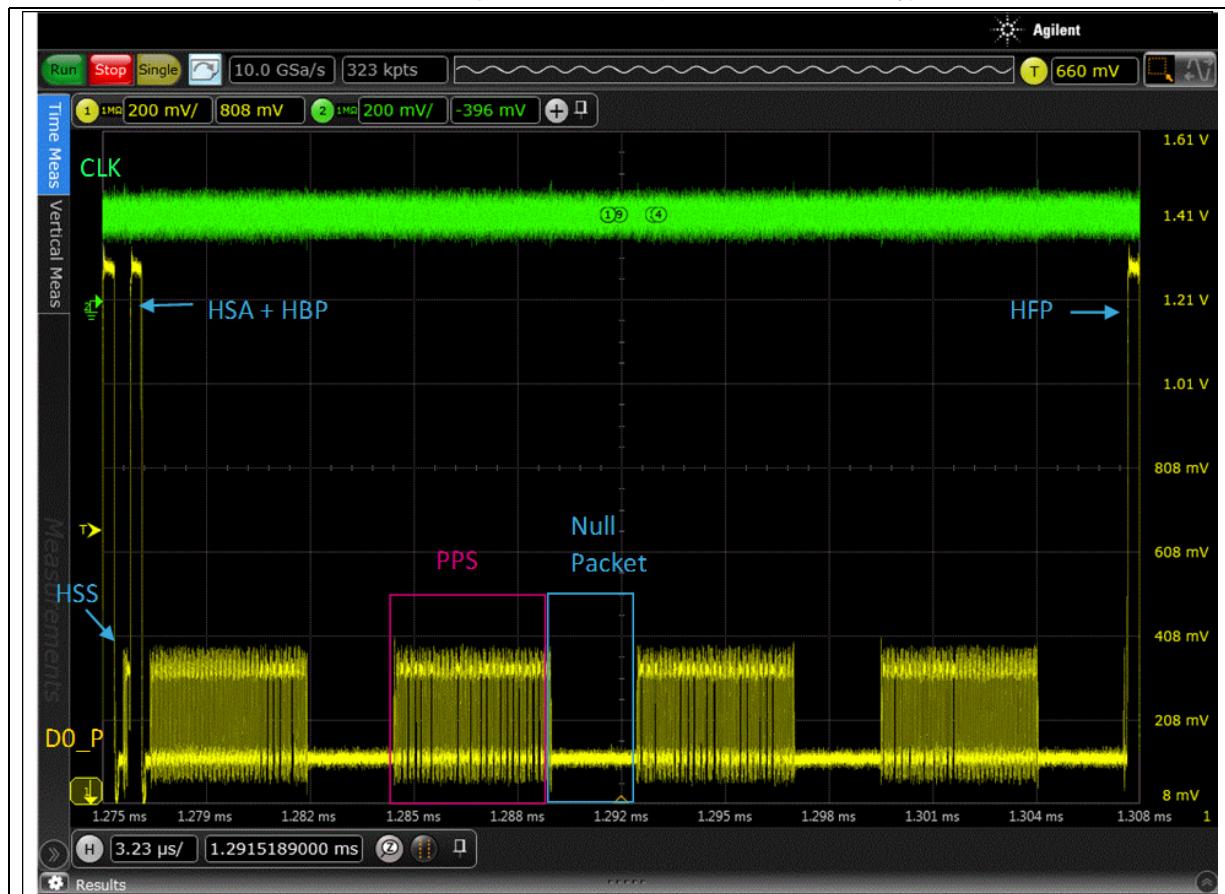
在HACT周期内，DSI主机必须在与LTCDHACT周期相匹配的时间周期内发送像素数据。根据DSI和像素时钟频率，DSI主机可使用一个或更多块发送像素数据。

每块只能包含一个封装像素流（PPS）数据包或具有空包的PPS数据包。

[第 5.2.1 节：通过LTDC接口实现视频模式](#)中讨论了非突发模式下块数和空包大小的选择。

[图 62](#) 显示了包含四个块的激活行。每块包含一个PPS数据包和一个空包。

图62. 包含四个块且空包启用的激活行



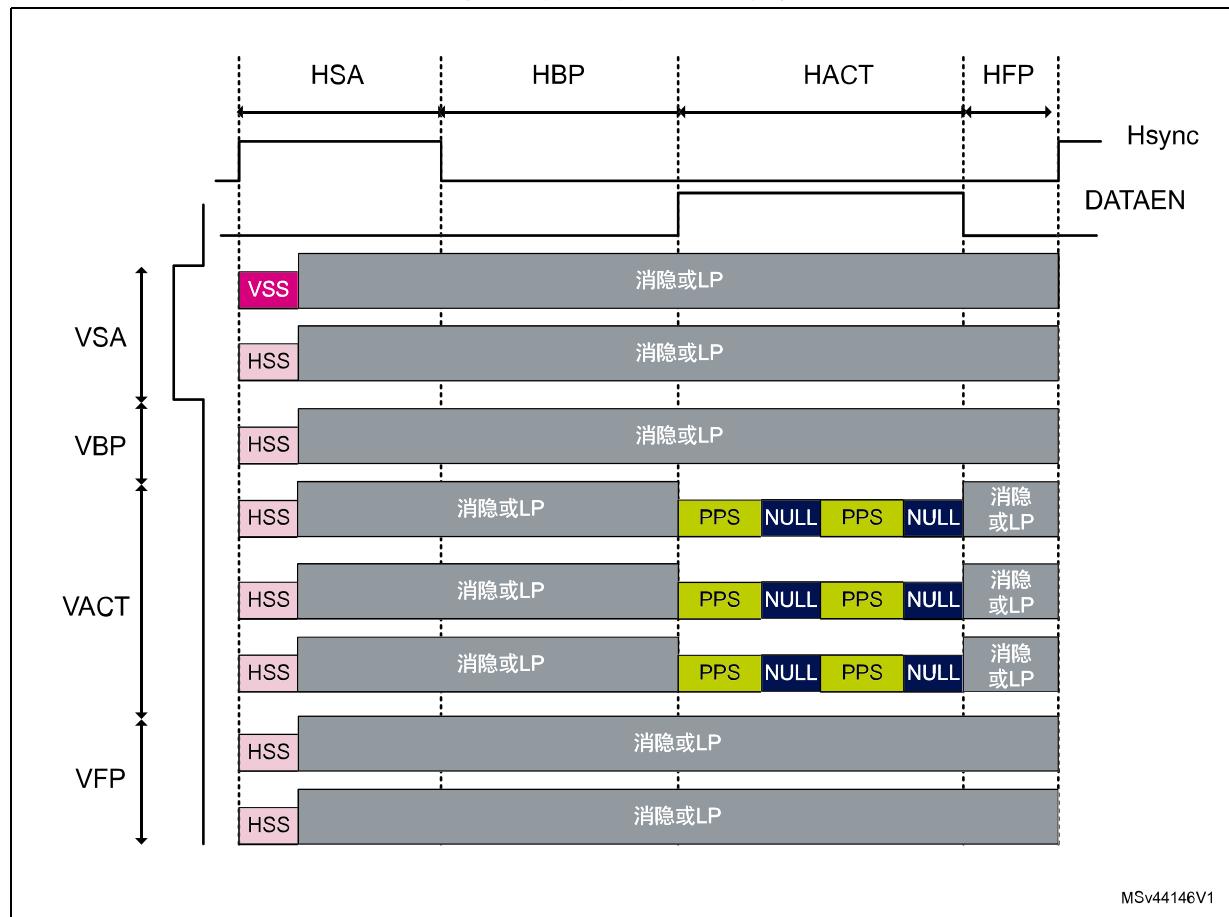
### 使用同步事件的非突发模式

在该模式下，无需同步脉宽的精确重建，因此取代了单一同步事件。在该模式下：

- 只传输每个同步脉冲的开头。
- 以与LTDC相同的速率执行传输。
- 如果没有足够的时间切换至LP模式，则使用消隐数据包取代LP模式下的时间间隔。

[图 63](#)显示了使用同步事件的非突发模式下的典型帧。

图63. 使用同步事件的非突发模式下的帧



以VSS数据包开头的行决定了VSA区域的开始。帧中的所有其他行以HSS数据包开头。

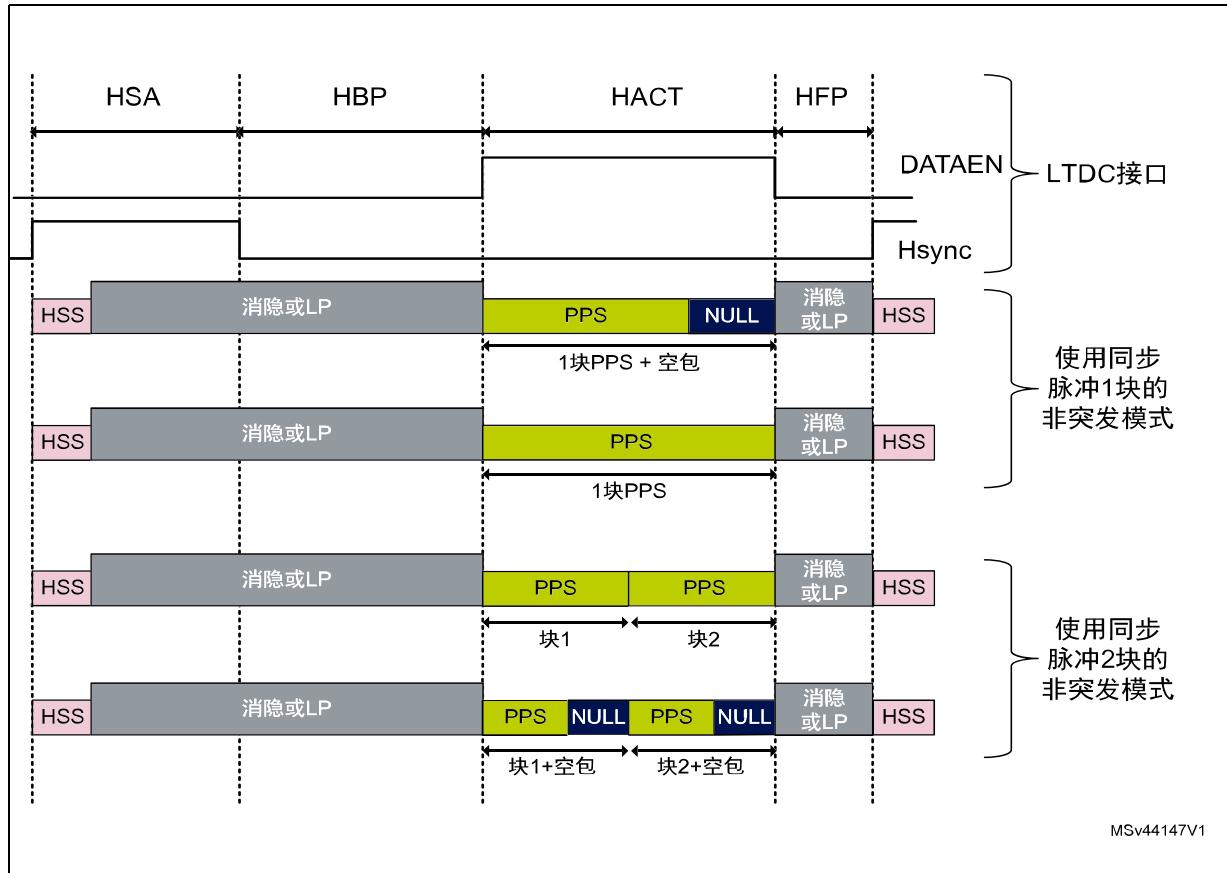
在未激活区域（VSA、VBP、VFP），链路在发送HSS数据包后进入LP，直至下一行。

在VACT区域，DSI主机发送HSS数据包，然后进入LP或发送消隐数据包，直至HSA+HBP周期结束。

HACT区域与使用同步脉冲的非突发模式相同。

[图 64](#)显示了使用同步事件的突发模式下的VACT区域。

图64. 使用同步事件的突发模式下的VACT区域



对于非突发模式的VACT区域，同步事件和同步脉冲的区别仅在HSA周期内。

DSI主机发送HSS数据包以通知HSA周期的开始。然后，链路进入LP或主机发送消隐数据包，直至HACT区域开始。

#### 何时使用非突发模式？

非突发模式为像素传输提供了更好的速率匹配。此模式能够：

- 只在存储器中存储一定量的像素，无需完整像素行（对DSI主机中RAM的要求更低）。
- 使用仅支持少量像素缓冲（少于一个完整像素行）的设备进行操作。

#### 突发模式

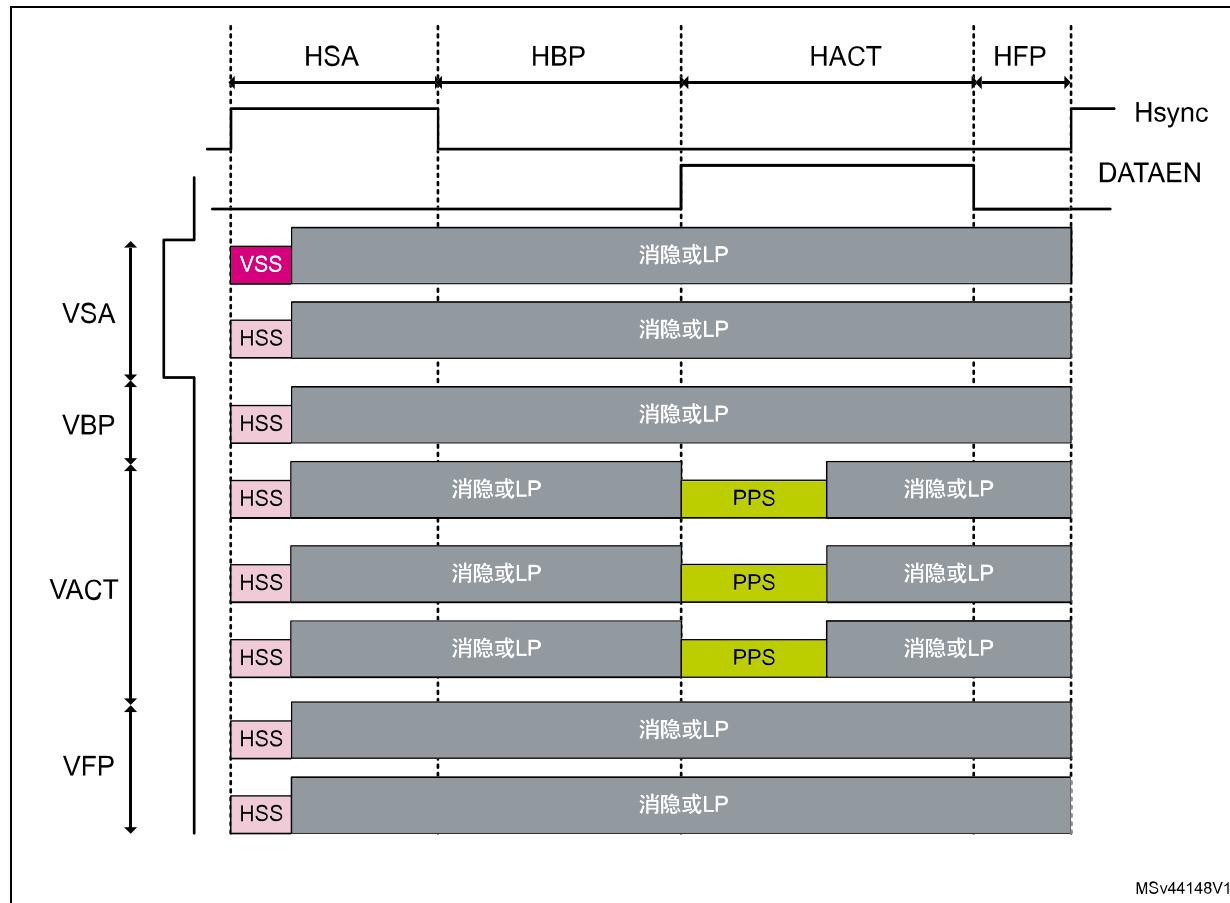
在视频突发模式下，RGB像素数据包经过时间压缩，在扫描行期间留下更多时间用于LP模式（节能）或用于将其他传输复用到DSI链路上。

在该模式下，整个激活像素行在FIFO中进行缓冲并以单个数据包进行传输，无中断。此传输模式要求DSI有效负载像素FIFO具有在其内部存储整行激活像素数据的容量。

最好在LTDC带宽与DSI链路带宽的差异较大时使用此模式。突发模式使DSI主机能够在一次数据突发传输中快速发出整个激活视频行，然后返回低功耗模式。

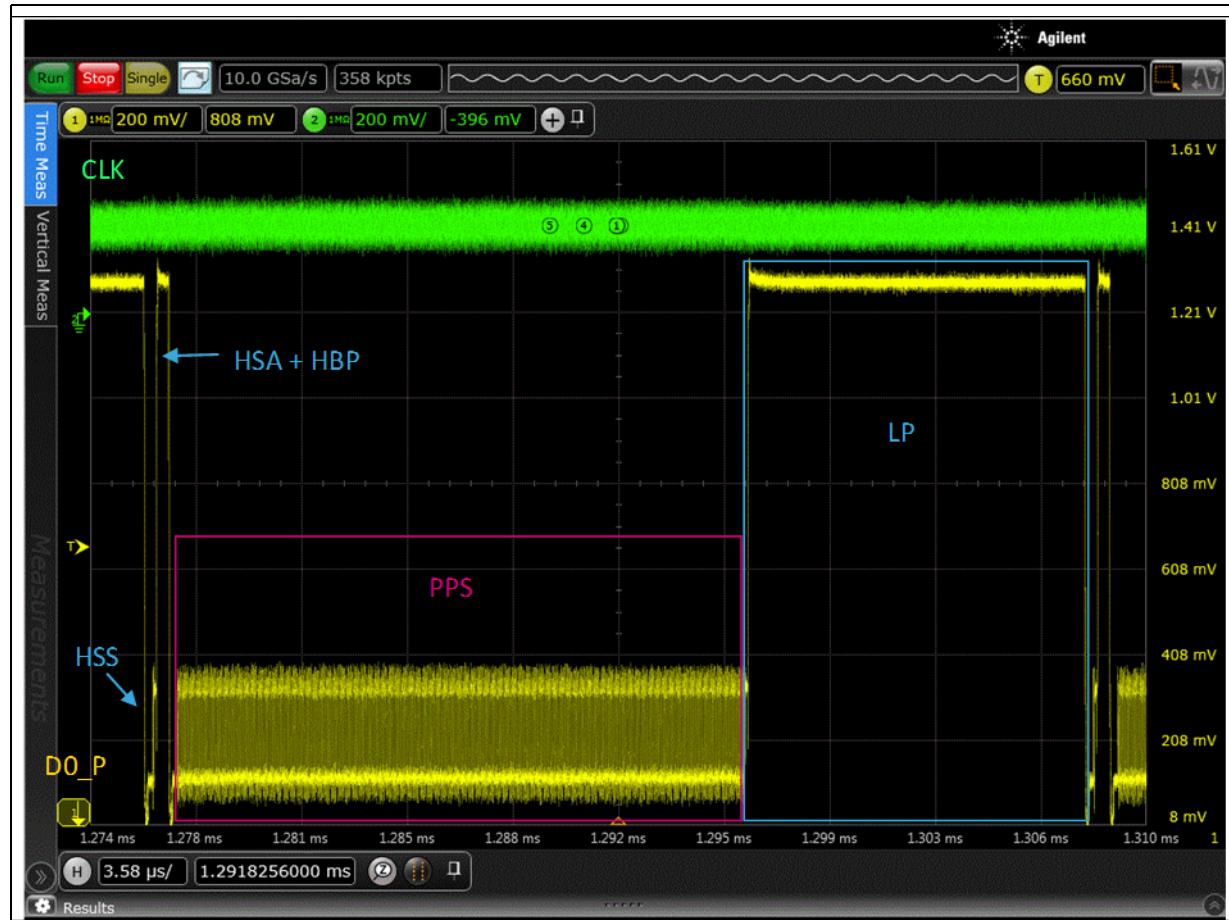
**图 65**显示了视频突发模式下的典型帧。与使用同步事件的非突发模式的区别在于HACT区域。在突发模式下，在HACT区域内，使用一个数据包以最大速度发送RGB数据，然后链路进入LP模式。

图65. 视频突发模式帧



**图 66**显示了包含使用PPS数据包发送的RGB像素数据的视频激活行。由于DSI链路带宽高于来自LTDC的DSI输入带宽，链路长时间处于LP模式。

图66. 突发模式下的激活行



### 何时使用突发模式？

在下列情况下使用突发模式：

- 显示设备支持在一次数据包突发传输中接收完整像素行，以避免接收缓冲区发生上溢。
- DSI输出带宽应高于LTDC接口输入带宽，以使链路每行进入低功耗状态一次。

### 视频模式比较：

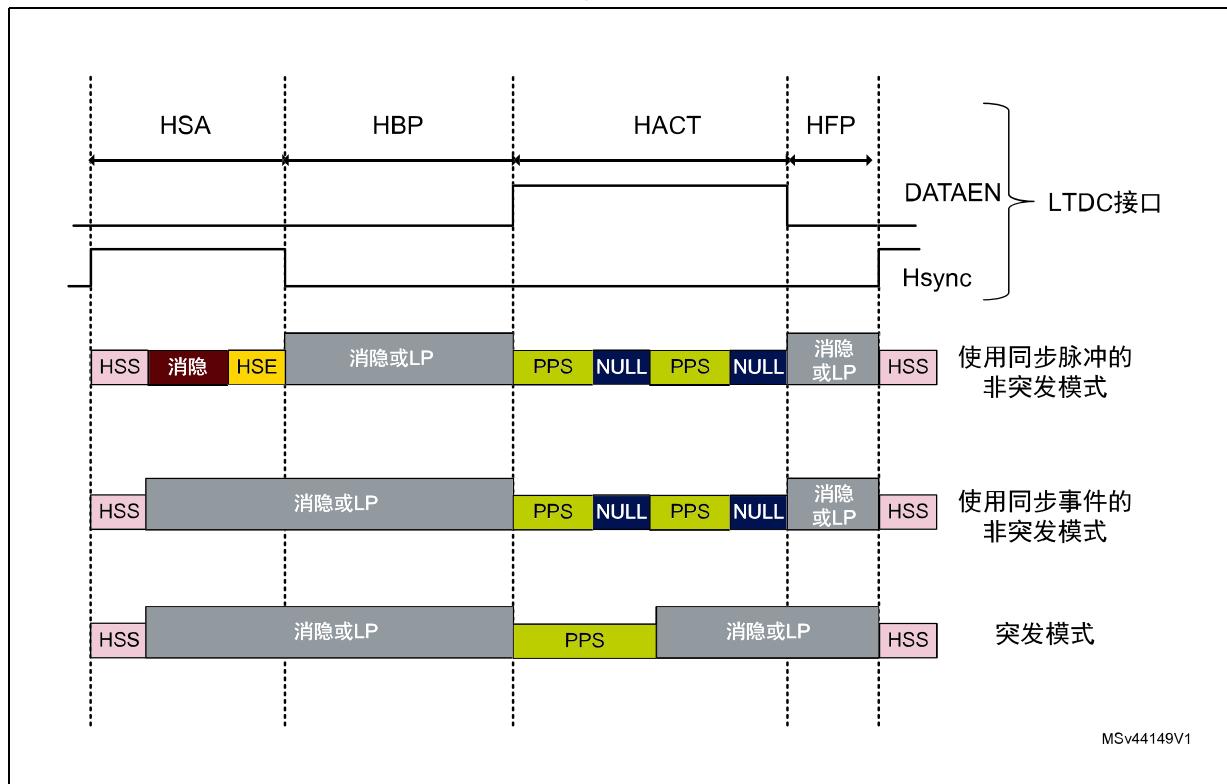
[图 67](#)显示了三种不同视频模式格式之间的比较。

使用同步脉冲的非突发模式是传输视频时序信息的最准确模式，但它不允许在HSA和HACT区域内进入LP。

使用同步事件的非突发模式允许在HSA区域内进入LP。

突发模式是能效最高的模式，因为即使是在HACT区域内，也允许进入LP模式。

图67. 视频模式比较



#### 4.2.2 适配指令模式

适配指令模式是一种经过充分优化的工作模式，用于连接具有自己的图形RAM的显示屏。

此模式自动刷新具有LTDC的显示屏图形RAM，不会产生任何CPU或DMA负载。

使用LTDC执行GRAM刷新操作：

- DSI主机控制LTDC，使能它并持续一个帧。
- 来自LTDC的RGB数据被捕获，并通过一系列的DCS长写指令数据包发送至显示屏。
- 一旦图形RAM完全刷新，DSI主机自动停止LTDC且DSI链路进入低功耗模式。

当帧缓冲器准备发送时，用户只通过置位一个位来控制显示屏的刷新操作。

只要无需更新帧缓冲器内容，显示屏即使用其内部图形RAM进行刷新操作。当显示屏的更新并不频繁时，这一点十分有用，可最大限度减少MCU侧的带宽使用量和总功耗。

为了缩短刷新持续时间，可以链路的最大速度执行刷新。必须特别注意LTDC侧的带宽要求。

用户可以只刷新帧缓冲器的一部分，以便提高动画速度并降低MCU侧的带宽要求。

适配指令模式仅支持DCS（显示指令集）、WMS（write\_memory\_start）和WMC（write\_memory\_continue）指令。为实现正常操作，还需要其他指令（例如显示配置指令）和撕裂效应初始化。这些指令必须通过APB接口发送。

### 适配指令模式下的显示刷新示例：

在适配指令模式下，DSI主机自动执行刷新过程。

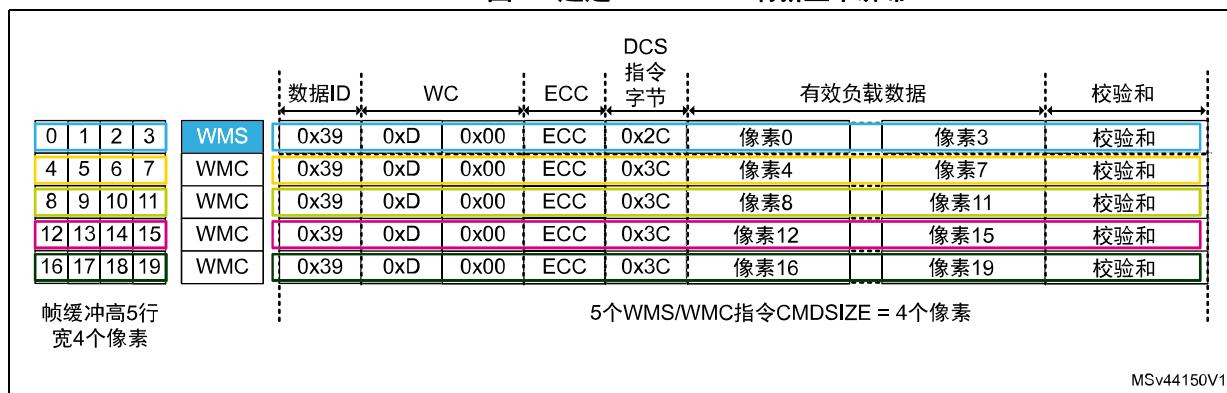
DSI主机使能LTDC，后者从帧缓冲器获取像素数据并传递给DSI主机。DSI主机将像素数据存储到专用FIFO中，然后将其封装到WMS和WMC DCS指令中。

在图 68所示的示例中，一个五行的帧可分五个DCS长写指令进行传输：一个Write\_Memory\_Start（WMS）和四个Write\_Memory\_Continue（WMC）。

CMDSIZE参数定义了每个WMS/WMC指令中要传输的像素数。

在本例中，CMDSIZE被设置为等于行长（以像素为单位），因此每个DCS指令封装了包含四个像素的一整行。

图68. 通过WMS/WMC刷新整个屏幕



### 适配指令模式下的部分刷新示例

适配指令模式支持部分刷新功能，允许刷新屏幕的特定部分，无需发送整个屏幕。

只重新发送部分屏幕而非整个屏幕可以：

- 减少系统带宽，因为只需从帧缓冲器获取一部分像素。
- 缩短刷新时间，提高动画速度。
- 降低功耗，因为DSI链路处于LP模式的时间更长。
- 最后，减少帧缓冲器存储所需的内存占用（可以只存储帧的一部分）。

在图 69所示的示例中，只传输了中心的像素。

CMDSIZE被设置为每个DCS指令中要发送的像素数，在本例中为2。

**注:** 为了选择显示屏侧要刷新的区域, 用户可能需要通过APB接口发送DCS指令(`set_column_address`和`set_page_address`)。此外, 用户必须重新编程LTDC参数, 以便选择要获取的像素窗口。  
请参见STM32cube指令模式示例了解关于所需设置的更详细信息。

图69. 显示屏部分刷新

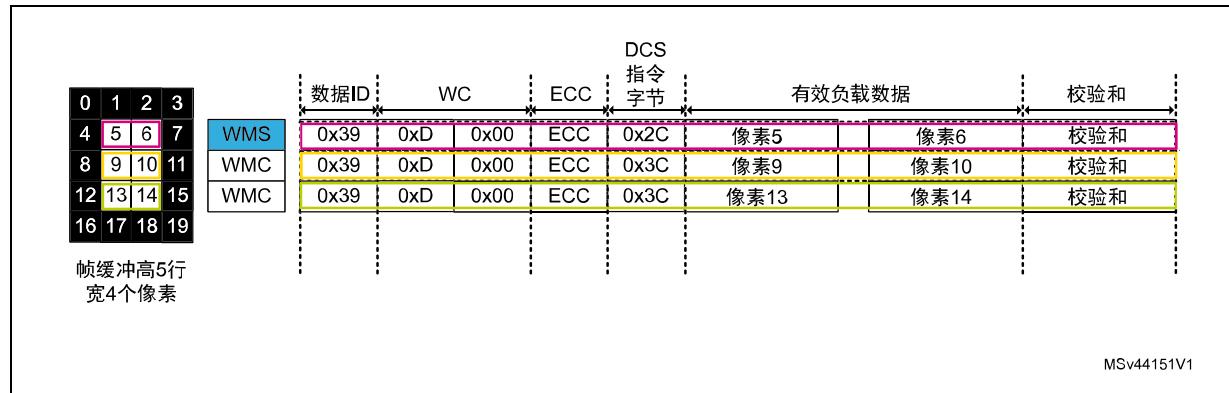


图 70 显示了适配指令模式的基本流程。

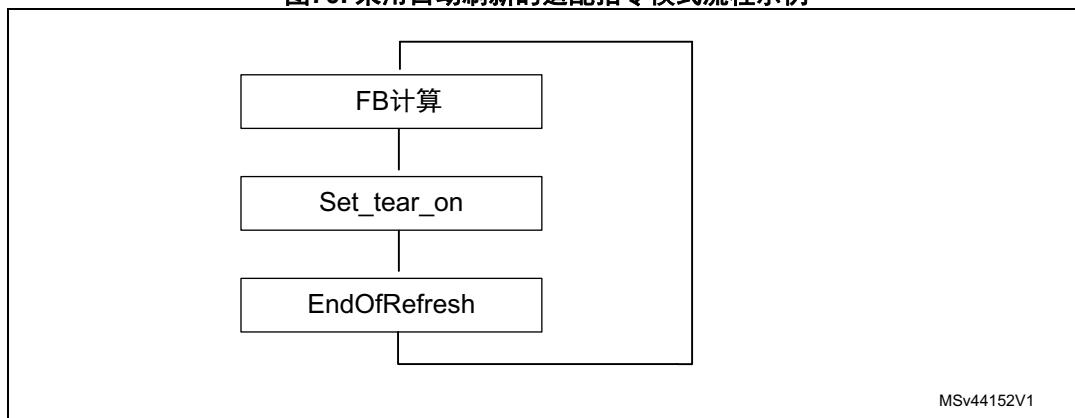
一旦帧缓冲器准备就绪, 用户发送`Set_tear_on`指令以激活显示屏侧的TE报告。当显示屏刷新到编程的扫描行时, 它发送TE触发消息。

如果使能了自动刷新功能, 在收到TE事件后, 主机通过发送WMS/WNDCS指令自动刷新显示屏。

如果自动刷新功能未使能, 则这种情况下只需通过在DSI Wrapper中置位一个位(LTDC\_EN位)来执行刷新。

当刷新操作终止时, `EndOfRefresh`事件被发送至DSI主机, 现在开始计算下一帧。

图70. 采用自动刷新的适配指令模式流程示例



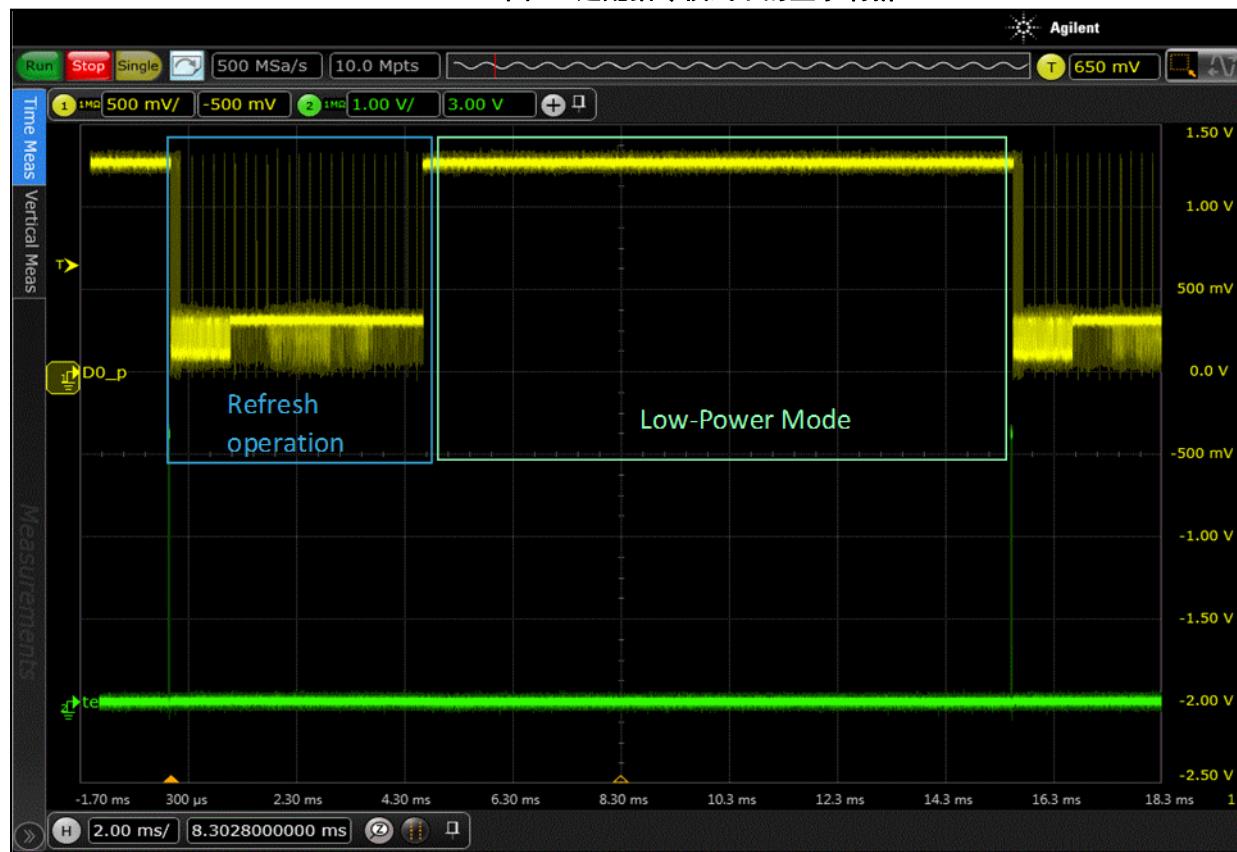
## 适配指令模式的优势

适配指令模式提供许多优势：

- 它可以实现集成度更高的解决方案。
- 图像计算和GRAM更新不得同时发生。这样可避免从帧缓冲器获取数据时LTDC与DMA2D图形引擎之间的并发问题。
- MCU侧有一个帧缓冲器足以。
- 帧缓冲器可嵌入内部MCUSRAM（色深为16bpp的320x320显示屏需要约200KB的帧缓冲器）。
- 低功耗：只要无需更新帧，显示屏即使用其图形RAM进行刷新，并且可以将链路置于低功耗模式。
- 支持部分刷新：用户可以选择只更新屏幕的一部分。这样可以缩短刷新时间，降低MCU侧的带宽使用率和功耗。
- 像素时钟可设置为最高频率，从而允许在更短周期内使用最大带宽。

[图 71](#)显示了适配指令模式下的显示刷新示例。DSI主机发送DCS指令以更新帧，然后回到LP模式。

图71. 适配指令模式下的显示刷新



## 撕裂效应管理

撕裂效应允许显示屏与DSI主机之间的完美同步，以便在具有自己的图形RAM的显示屏上执行刷新操作。

撕裂效应信令可通过两种方式来执行：

- 通过链路，无需任何额外的引脚。
- 或使用额外的引脚。

### 通过链路报告撕裂效应

当通过链路报告撕裂效应时，DSI主机发送set\_tear\_on或set\_tear\_scanline指令，并通过BTA流程将总线所有权交给显示屏。在首个BTA后，显示屏将以回应触发信号做出响应，并将总线所有权还给主机。主机将开始第二个BTA流程，以便将总线所有权交给显示屏。一旦显示屏刷新到编程的扫描行，将向DSI主机发送触发信号，并将总线所有权还给DSI主机。

可以发起中断以启动图形RAM刷新，或者用户可以选择执行自动刷新。

[图 73](#)显示了通过DSI链路报告的撕裂效应。

图72. 撕裂效应请求与响应示例



[图 73](#)显示了具有双BTA的Set\_tear\_scanline指令的放大图。

图73. 具有双BTA的Set\_tear\_scanline DCS指令

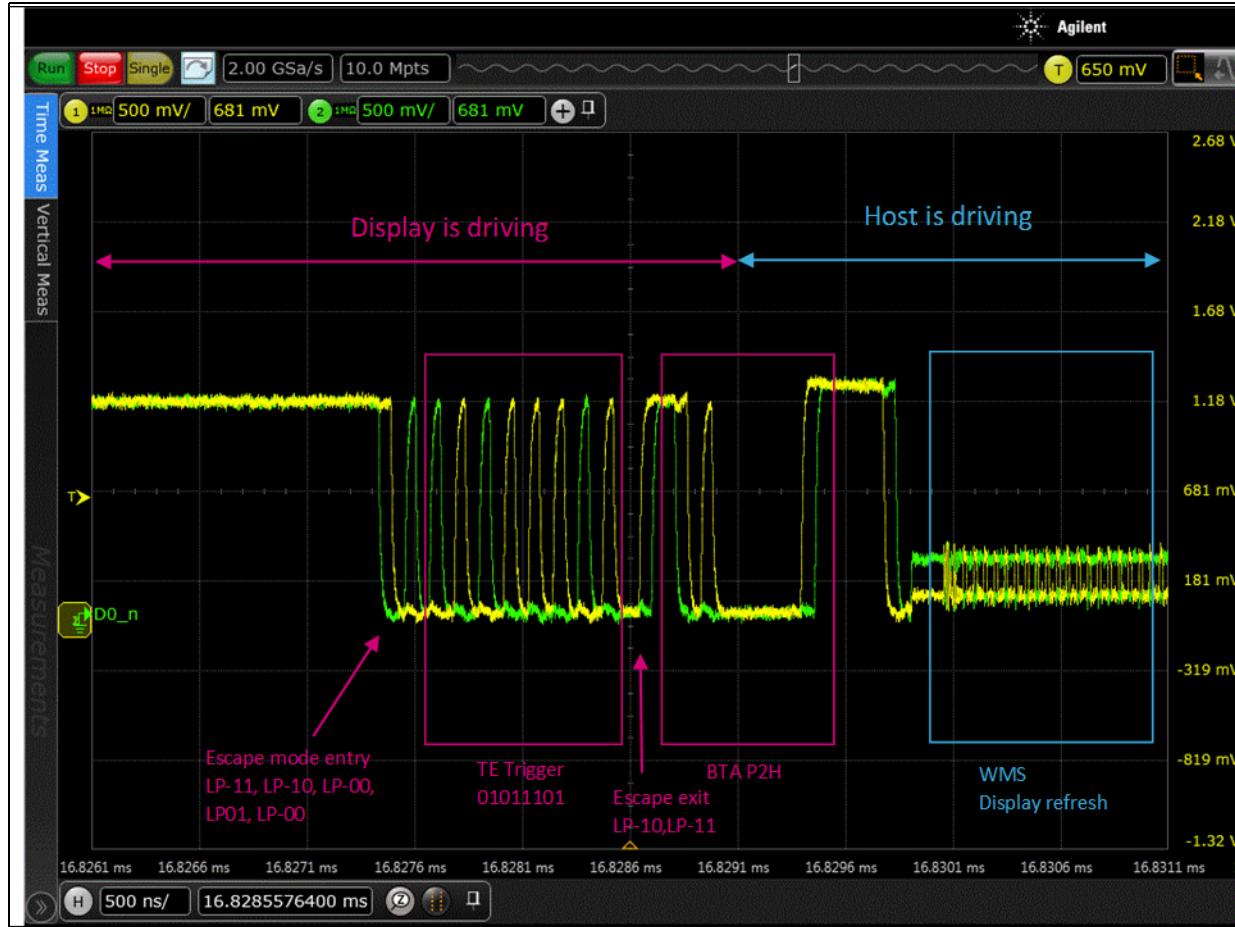


当显示屏刷新到指定的扫描行时，它通过BTA流程发送撕裂效应触发信号，将总线所有权还给DSI主机。

[图 74](#)显示了撕裂效应触发消息的放大图。

然后，DSI主机可以使用WMS/WMC DCS指令开始显示刷新操作。

图74. 来自显示屏的撕裂效应触发信号



#### 通过引脚报告撕裂效应

当通过引脚报告撕裂效应时，DSI主机发送set\_tear\_on或set\_tear\_scanline DCS指令。然后，显示屏在刷新到编程的扫描行时切换专用引脚以触发DSI主机。

set\_tear\_on或set\_tear\_scanline DCS指令只能发送一次，然后显示屏将生成连续同步信号。

即使需要额外的引脚，此机制也能避免通过DSI主机与显示屏之间的链路进行多次交换。

可通过引脚切换发起中断以启动图形RAM刷新。

然后，DSI主机通过发送WMS/WMC DCS指令开始刷新操作。

可在检测到TE中时自动执行或通过软件使能刷新。

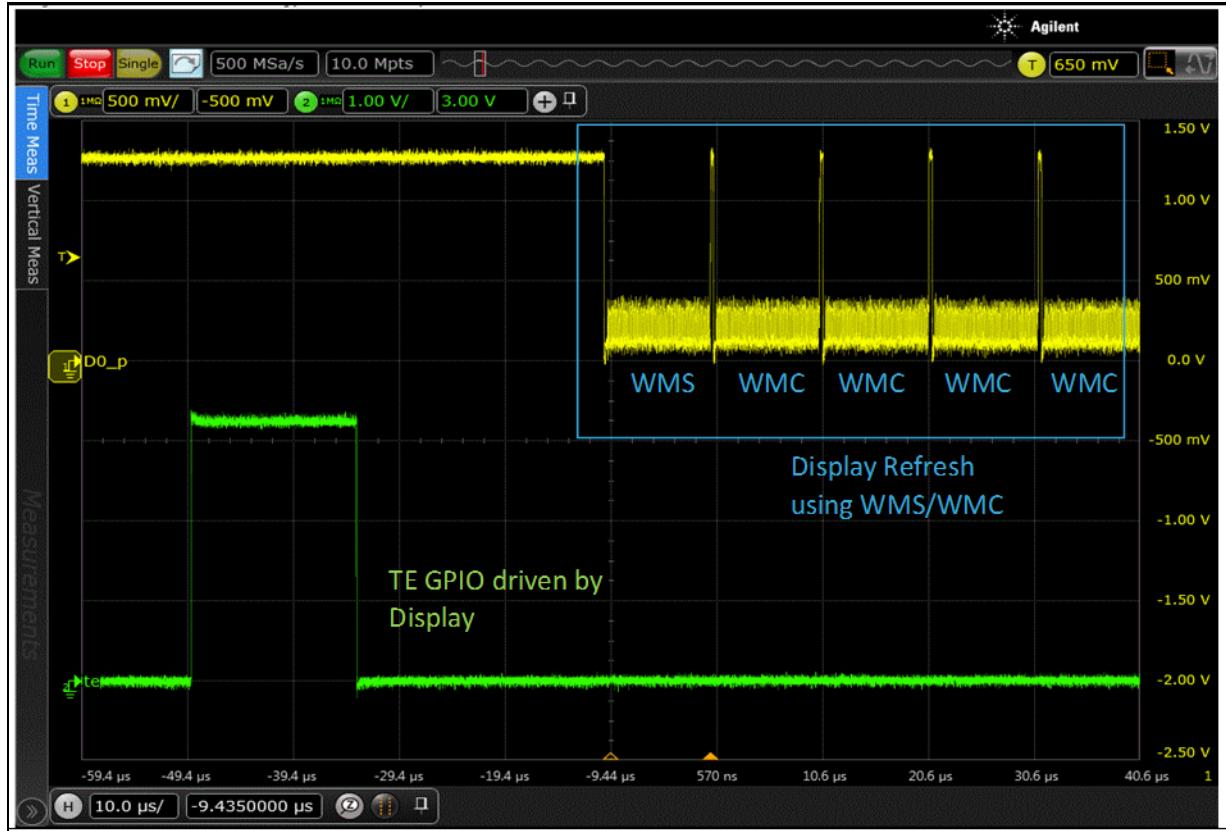
[图 75](#)显示的是通过GPIO报告TE。当显示模块刷新到编程的扫描行时，会在TE引脚上生成脉冲。

[图 76](#)显示了通过GPIO报告的撕裂效应放大图。

图75. 通过引脚报告TE



图76. 通过引脚报告撕裂效应放大图



#### 4.2.3 APB指令模式

APB指令模式用于通过DSI主机APB寄存器接口发送指令。

通用指令或DCS指令可在启动时发送用于显示配置或在应用运行时发送用于维护操作。

所有指令可在高速或低功耗模式下发送，因为某些显示屏在启动时只接受低功耗通信。

也可在视频流传输期间发送指令。DSI主机调度器根据编程时序自动评估是否有足够的时间在视频传输期间插入指令。

所有指令可通过软件全面编程，这使DSI主机能够支持所有标准DCS指令和所有取决于显示屏的自定义指令。

##### 何时使用APB指令模式？

APB指令模式主要用在显示屏初始化阶段和显示屏上的维护操作。它与视频模式或适配指令模式同时工作。

## 4.3 工作模式选择

选择视频模式或适配指令模式对解决方案的架构和成本有很大影响。

从微控制器的角度来看，适配指令模式是成本优化解决方案的首选。

由于视频模式不要求显示屏侧有图形RAM，此解决方案适合大尺寸显示屏，大尺寸显示屏出于成本优化的考虑不携带图形RAM。MCU侧的带宽和内存使用限制与当前基于LTDC的解决方案相同。大部分情况下，帧缓冲器的双倍缓冲需要外部RAM。

适配指令模式需要具有图形RAM的显示屏。这种显示屏的成本稍高，但大多数低于 $480 \times 480$ 的小尺寸显示屏都嵌入了图形RAM。因此，小尺寸显示屏的适配指令模式不需要外部RAM，因为帧缓冲可以存储在内部MCUBRAM中。由于无需外部RAM，这样可以显著减少MCU上的带宽问题、总BOM成本和解决方案集成。

## 4.4 DSI中断

DSI主机有许多种中断，用于监控所有通信时序和事件。请参见相关的STM32参考手册了解所有中断源的详细描述。

DSI Wrapper主机或DSI主机均可以生成中断。所有中断均合并成一个中断行进入中断控制器。

除了协议相关中断，DSI主机还提供中断用于管理：

- 调整器事件。
- PLL事件。
- 撕裂效应事件（仅在指令模式下）。
- 刷新结束事件（仅在指令模式下）。

## 4.5 低功耗模式

在运行和睡眠模式下，DSI主机激活。DSI主机中断可导致设备退出睡眠模式。

在停止模式下，DSI主机冻结，其寄存器内容被保留。在待机模式下，DSI主机掉电，之后必须重新初始化。

[表 9](#)显示了DSI主机低功耗模式。

表9. DSI主机低功耗模式

模式	说明
运行	激活。
睡眠	激活。外设中断导致设备退出睡眠模式。
停止	冻结。外设寄存器内容被保留。
待机	掉电。在退出待机模式后，必须重新初始化外设。

## 5 DSI主机配置

本节描述了配置和使用DSI主机所需的底层硬件寄存器。关于完整的编程序列，请参见相关的STM32参考手册。

配置过程分为两部分：

- 全局初始化：通用于所有工作模式：视频模式或适配指令模式。
- 工作模式配置：本节取决于选择的工作模式，即视频模式或适配指令模式。

**注：**APB指令模式与视频模式和适配指令模式同时使用。由于会导致高延迟，它未被优化用来单独刷新显示屏，即使可以实现。因此，这里只重点介绍视频模式和适配指令工作模式配置。

### 5.1 DSI主机全局配置

本节描述通用于所有工作模式的DSI主机配置。下面将详细描述调节器、PLL、PHY、流控和LTDC接口配置。

#### 5.1.1 DSI调节器配置

DSI主机具有专用的1.2 V内部调节器，用于向D-PHY和PLL提供电源。[表 10](#)显示了配置DSI调节器所需的寄存器。

**表10. DSI调节器配置寄存器**

说明	寄存器字段
使能调节器	DSI_WRPCR.REGEN
等待调节器就绪	DSI_WISR.RRS

通过DSI Wrapper控制提供1.2 V电压的DSI调节器。

通过将DSI\_WRPCR寄存器的REGEN位置来使能调节器。

当调节器就绪时，置位DSI\_WISR寄存器的RRIF位。

通过直接使能1.2 V调节器完成D-PHY的电源开/关。

#### 5.1.2 时钟配置

本节显示了DSI主机需要的不同时钟配置：

- 使用DSI主机内部PLL的HS时钟。
- 用于LP通信的TX逸出时钟。
- DSI PLL关闭时的超低功耗状态模式需要次级时钟源。

## DSI PLL配置

STM32 DSI主机通过DSI Wrapper控制专用DSI PLL。表 11 显示了DSI PLL配置寄存器。

表11. DSI PLL配置寄存器

说明	寄存器字段
配置PLL环路分频因子	DSI_WRPCR.NDIV
配置PLL输出分频因子	DSI_WRPCR.ODF
配置PLL输入分频因子	DSI_WRPCR.IDF
使能PLL	DSI_WRPCR.PLLEN
等待PLL锁定	DSI_WISR.PLLS

DSI PLL的输入时钟为HSE（高速外部）振荡器时钟。

PLL输出时钟是馈送至D-PHY的HS时钟。HS时钟是全频时钟。它必须处于80MHz至500MHz的范围内。

D-PHY使用HS时钟生成半频DDR时钟，并使用时钟通道传输至显示屏。这意味着对于500 Mbit/s每通道的速率，PLL输出必须是500 MHz，并且传输的时钟通道为250 MHz DDR时钟。

D-PHY将HS时钟除以8以生成lane\_byte\_clk，并将其馈送至DSI主机。

使用以下公式计算lane\_byte\_clk：

$$FVCO = (HSE / IDF) \times 2 \times NDIV$$

Lane\_Bit\_CLK = FVCO / (2xODFx8)，FVCO必须处于500MHz至1GHz的范围内。

### 计算示例

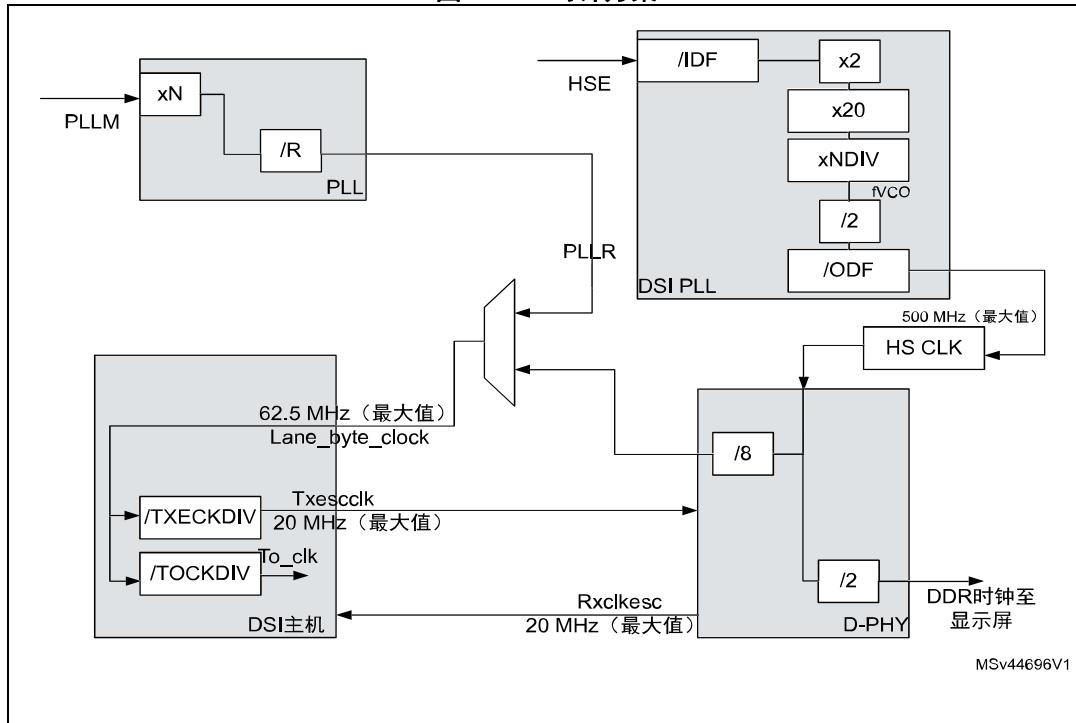
要获得每通道500 Mbit/s的速率，必须配置DSI PLL以输出500 MHz HS时钟，并且馈送至DSI主机的lane\_byte\_clk必须等于500 MHz / 8 = 62.5 MHz。使用25 MHz HSE振荡器时，可使用这些参数得到lane\_byte\_clk = 62.5 MHz：

IDF = 1, NDIV = 20, ODF = 1

$$\text{Lane}_\text{byte}_\text{clk} = ((25/1) \times 2 \times 20) / (2 \times 1 \times 8) = 62.5 \text{ MHz}$$

图 77 显示了DSI时钟方案。

图77. DSI时钟方案



### TX逸出时钟配置

TX逸出时钟用在LP模式下。在LP模式下，需要两个周期的tx\_esc\_clk传输一个比特位。  
[表 12](#)显示了用于配置TX逸出时钟的寄存器。

表12. TX逸出时钟配置寄存器

说明	寄存器字段
TX逸出时钟分频	DSI_CCR.TXECKDIV

TX逸出时钟的计算如下：

$$\text{TX escape CLK} = \text{Lane\_byte\_clk} / \text{TXECKDIV}$$

**注：**必须编程TXECKDIV以确保TX逸出时钟小于20MHz。必须将TX预分频器设置为大于2的值。预分频器值0或1将禁用TX逸出时钟的生成。

### 次级时钟源设置

可以从DSI-PHY或从主PLL的特定输出（PLLR）馈送DSI主机lane\_byte\_clock源。在超低功耗状态下，DSI PLL可能已关闭。这种情况下，lane\_byte\_clock源为PLLR。

[表 13](#)显示了在两个时钟源之间进行选择时使用的寄存器位。

**表13. 次级时钟源设置寄存器**

说明	寄存器字段
DSI时钟源选择	DCKCFGR2.DSISEL

当DSI-PHY用作DSI lane\_byte\_clk源时，必须清除此位。这是正常工作模式。

在DSI PLL和DSI-PHY关闭的情况下，当PLLR用作DSI lane\_byte\_clk源时，必须置位此位。

### 5.1.3 DSI主机PHY参数

本节介绍从DSI主机控制的D-PHY参数。

#### 通道数

DSI主机提供使用一个或两个数据通道的可扩展架构。[表 14](#)显示了用于通道数编程的寄存器字段。

**表14. 通道数配置寄存器**

说明	寄存器字段
设置激活数据通道的数量	DSI_PCONFR.NL

通道数取决于应用的带宽要求。

每个DSI通道的最大数据速率为500 Mbit/s，在双数据通道模式下，总速率为1Gbit/s。请参见[第 7 节：DSI主机性能](#)来获取更多关于如何评估所需通道数的信息。

#### PHY时钟和数字部分控制

DSI主机提供对D-PHY数字部分和时钟通道模块的控制。

[表 15](#)显示了用于控制D-PHY时钟和数字部分的寄存器字段。

**表15. PHY时钟和数字部分控制寄存器**

说明	寄存器字段
D-PHY数字部分控制	DSI_PCTRLR.DEN
使能D-PHY时钟通道模块	DSI_PCTRLR.CKE

DSI\_PCTRLR.DEN字段可使D-PHY退出复位。

必须置位DSI\_PCTRLR.CKE以使能D-PHY时钟通道。

#### 时钟通道控制

DSI主机提供特定的功能用于时钟通道控制。[表 16](#)显示了用于时钟通道控制的不同寄存器字段。

表16. 时钟通道控制寄存器

说明	寄存器字段
自动时钟通道控制	DSI_CLCR.ACR
D-PHY时钟控制	DSI_CLCR.DPCC

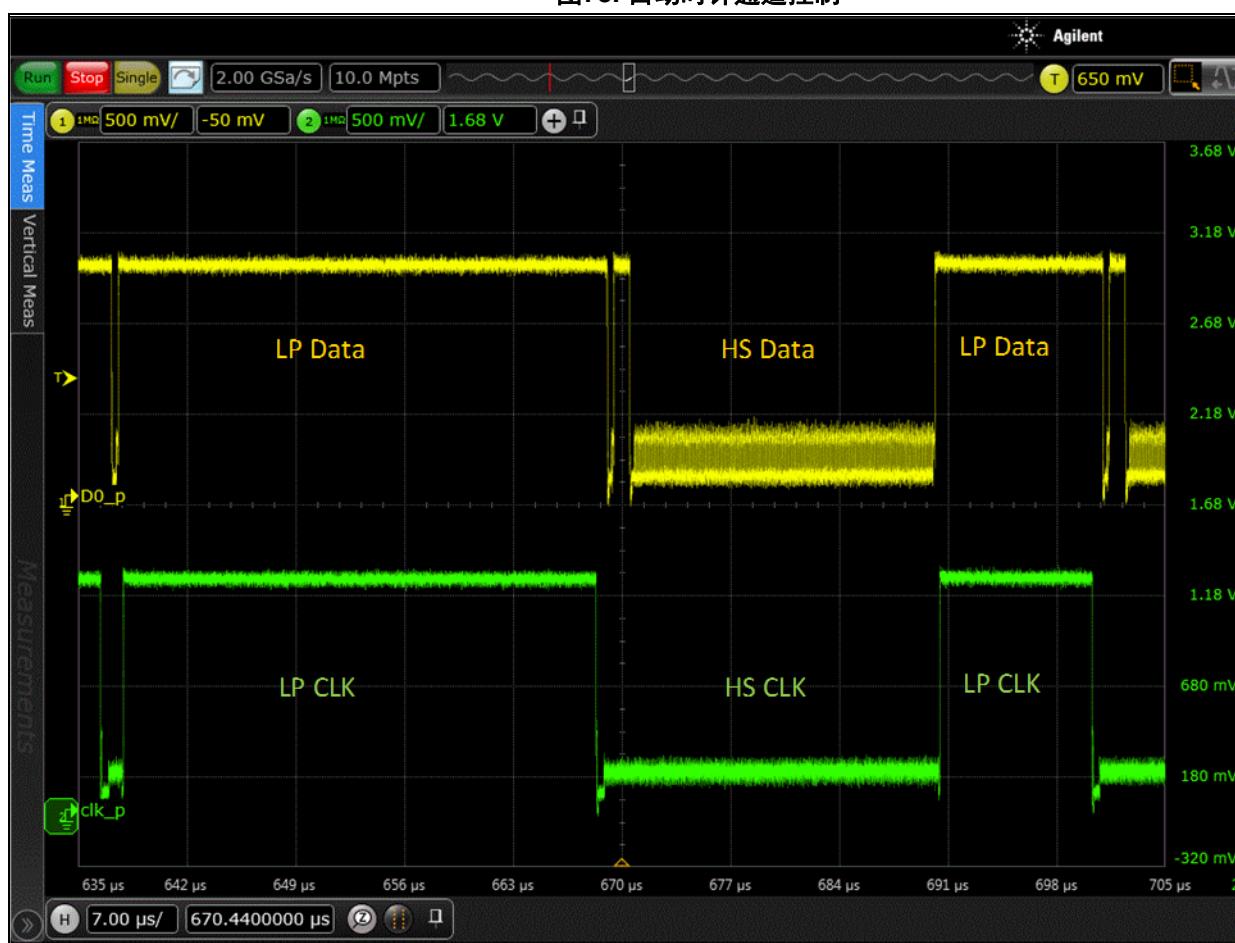
### 自动时钟通道控制

用户可以选择让DSI主机自动控制何时让时钟通道进入LP状态。

这是通过将DSI\_CLCR.ACR位置为1来实现的，并且还必须将DSI\_CLCR.DPCC置为1以使能时钟的HS模式。

图78显示的是使能了自动时钟通道控制的示例。DSI主机自动停止以HS模式提供时钟，并在时间允许时将其置于停止状态LP-11。

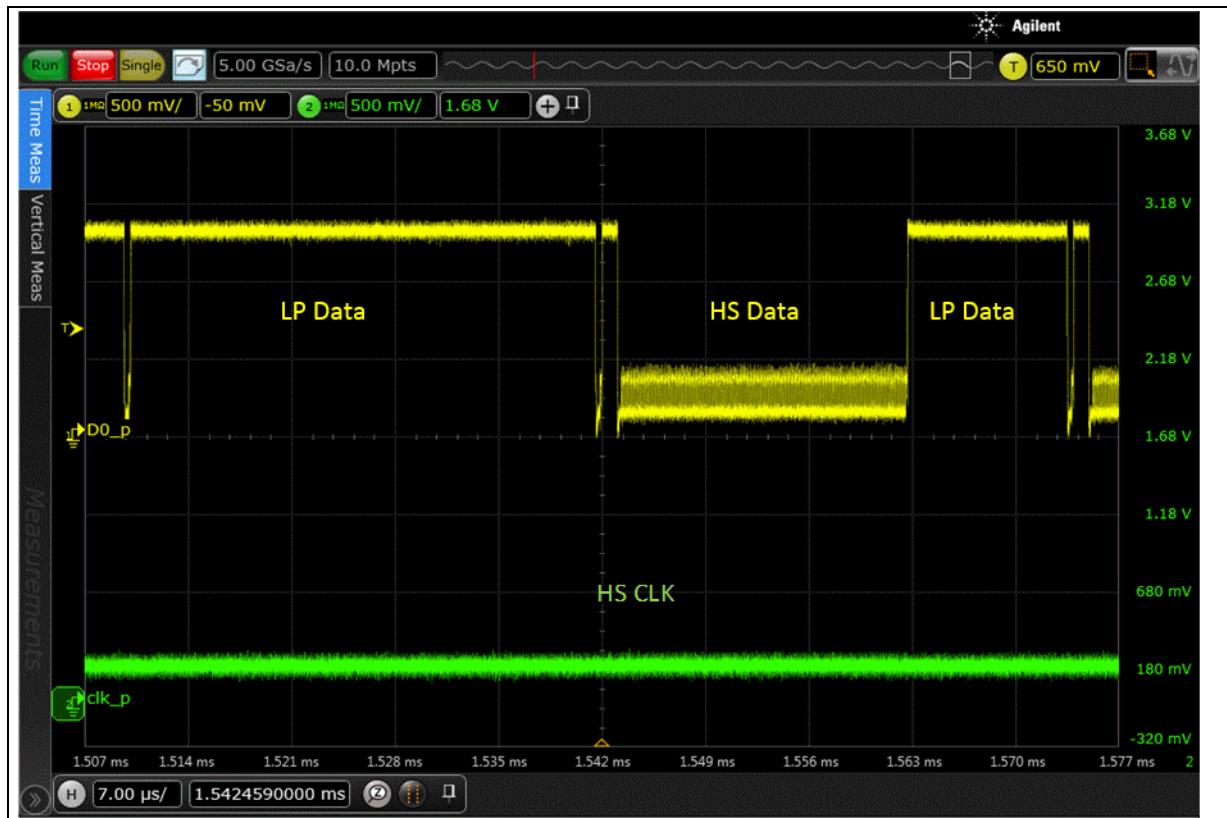
图78. 自动时钟通道控制



注：在视频模式下，DSI主机根据PHY转换时间提供的时间（DSI\_CLTCR.HS2LP\_TIME和DSI\_CLTCR.LP2HS\_TIME）来决定是否可以让时钟通道进入LP模式，具体讨论见第 5.2.1节：通过LTDC接口实现视频模式。

图 79 显示的是禁用了自动时钟通道控制并且时钟通道始终处于HS模式的示例。

**图79. 禁用了自动时钟通道控制并且时钟通道始终处于HS模式的示例**



#### 手动时钟通道控制

还可以通过DSI主机时钟通道配置寄存器的DSI\_CLCR.DPCC位手动控制时钟通道状态。可通过清除DSI\_CLCR.DPCC将时钟通道手动置于LP状态。

### 5.1.4 DSI Wrapper PHY参数

本节介绍从DSI Wrapper控制的D-PHY参数。

#### HS位周期设置

D-PHY需要知道PLL提供的高速时钟频率。该值通过DSI Wrapper指示给D-PHY。表 17 显示了DSI Wrapper内部用于设置HS位周期的寄存器字段。

**表17. HS位周期配置寄存器**

说明	寄存器字段
设置HS模式下的位周期（0.25 ns的倍数）	DSI_WPCR0.UIX4

此字段以0.25 ns的形式定义了高速模式下的位周期，它被用作由D-PHY管理的所有时间的时基。

通过DSI\_WPCR0.UIX4字段配置单位间隔。如果该周期不是0.25ns的倍数，则应将驱动值向下舍入。

**注：**此字段为必填字段。为避免DSI主机与PHY之间的时间不匹配问题，必须正确设定此字段。

**示例：**对于500 Mbit/s的链路速度，来自PLL并被馈送到PHY的HS时钟输出为500 MHz，位周期为2 ns。因此，将UIX4设定为 $2 / 0.25 = 8$ 。

单位间隔是时钟周期的一半，因此

$UI = 1 / (2 \times 250 \text{ MHz}) = 2 \text{ ns}$ ，因此将UIX4设定为8。

## 5.1.5 协议流控

DSI主机提供流控功能，包括EoTp发送和接收、ECC和CRC接收以及总线转向控制。

**注：**正向的ECC和CRC生成是强制性的，并且始终使能。

[表 18](#)显示了用于流控设置的寄存器字段。

**表18. 协议流控配置寄存器**

说明	寄存器字段
EoTp发送使能	DSI_PCR.ETTXE
EoTp接收使能	DSI_PCR.ETRXE
总线转向使能	DSI_PCR.BTAE
ECC接收使能	DSI_PCR.ECCRXE
CRC接收使能	DSI_PCR.CRCRXE

### EoTp发送和接收

一次HS传输中可能存在多个数据包。始终在PHY层使用专用EoT序列发送传输结束信号。为了增强系统的总体稳定性，DSI在协议层定义了专用的EoT数据包（EoTp）。

此机制以增加开销（每次传输增加四个额外字节）为代价提供更稳定的环境。

在HS传输结束时发送EOTp。之后，链路进入LP状态（请参见[第 3.3.2 节：传输结束（EoT）数据包](#)获取更详细信息）。

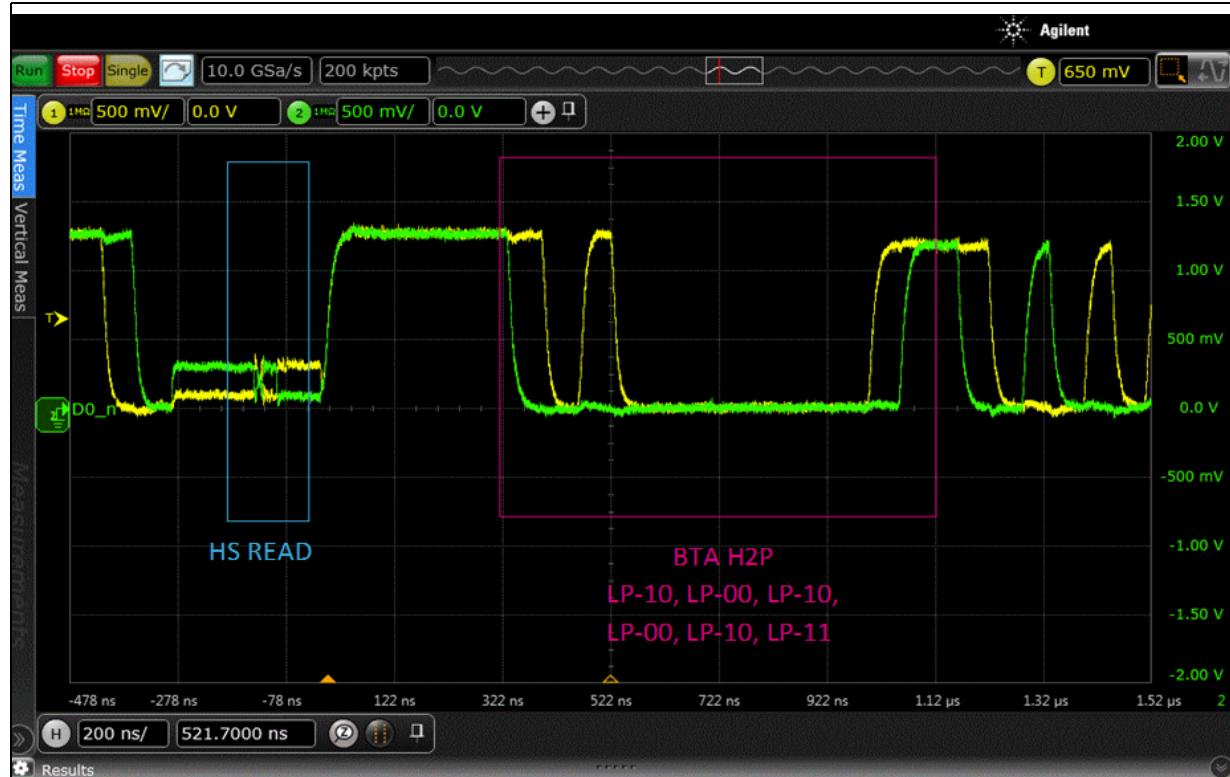
DSI主机通过分别置位DSI\_PCR.ETTXE和DSI\_PCR.ETRXE支持EoTp的发送和接收。

### 总线转向

当需要反向通信（例如读取、回应和撕裂效应请求）时，必须使能总线转向。

[图 80](#)显示了读指令的示例，DSI\_PCR.BTAE置位时DSI主机自动启动BTA流程。

图80. 读指令后的BTA流程



### ECC和CRC接收

DSI主机可检查已接收数据包的ECC和CRC字段的值。分别通过置位DSI\_PCR.ECCRXE和DSI\_PCR.CRCRXE来使能此检查。

## 5.1.6 DSI主机LTDC接口配置

DSI主机LTDC接口允许颜色编码和视频信号极性控制。

### 颜色编码配置

[表 19](#)显示了用于控制颜色编码的寄存器字段。

表19. 颜色编码配置寄存器

说明	寄存器字段
DSI Wrapper颜色编码配置	DSI_WCFGR.COLMUX
DSI主机颜色编码配置	DSI_LCOLCR.COLC
松散封装版本，18位配置	DSI_LCOLCR.LPE

LTDC输出始终为24位RGB R[7:0]G[7:0]B[7:0]。

DSI Wrapper允许将LTDC输出映射至DSI主机输入。必须为DSI Wrapper和DSI主机选择相同的颜色格式。此颜色格式将应用于DSI输出数据包。

可用的颜色编码为：

- 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

不同的16位和18位配置对DSI输出数据包无影响。用户可以任选其一，只要在DSI Wrapper (DSI\_WCFG&OLMUX) 和DSI主机 (DSI\_LCOLCR.COLC) 中设定了相同配置。

如果选择了任意16位配置，DSI主机将输出565颜色编码。如果选择了任意18位配置，DSI主机将输出666颜色编码。

但视频模式18位配置是一个例外，DSI\_LCOLCR.LPE位在松散和非松散数据包配置之间进行选择（请参见图 43和图 44了解非松散和松散18位数据包）。

对于视频模式18位颜色格式，如果使用18位松散数据包版本，则用户必须置位DSI\_LCOLCR.LPE。

**注：** 在视频模式下，得益于封装像素流包头中的数据类型字段，显示屏能够获知使用的颜色格式（请参见第 1 节视频模式数据类型 第 40页）。在指令模式下，主机需使用set\_pixel\_format DCS指令将颜色格式通知显示屏。

**注：** DSI主机颜色编码定义了通过DSI链路传输的DSI数据包中的像素编码方式。它不同于图像源颜色格式。图像源颜色格式必须与LTDC层输入颜色格式一致。可以同时使用16位编码的图像源和24位编码的DSI主机输出数据包。

### 视频控制信号极性

可通过LTDC接口控制视频控制信号 (Hsync、Vsync和DATA ENABLE) 的极性。

表 20显示了视频控制信号极性的寄存器字段。

表20. 视频控制信号极性寄存器

说明	寄存器字段
配置HSYNC极性	DSI_LPCR.HSP
配置VSYNC极性	DSI_LPCR.VSP
配置DATA ENABLE极性	DSI_LPCR.DEP

可通过DSI LTDC接口进行视频控制信号 (Hsync、Vsync和DE) 极性的编程。

编程值必须与LTDC控制信号极性设置一致，这意味着DSI和LTDC中的极性必须相同，但DE信号除外。

LTDC输出（非DE）非DE信号，因此如果LTDC DE极性为高电平有效，则DSI中的DE极性必须为低电平有效，反之亦然。

## 5.2 DSI工作模式配置

本节介绍特定于所使用工作模式的配置设置。用户可以选择视频模式或适配指令模式。

### 5.2.1 通过LTDC接口实现视频模式

本节介绍特定于视频模式的DSI主机配置。

#### 视频模式选择

用户必须从三种可用的视频模式中选择一种。视频模式选择寄存器如[表 21](#)所示。

**表21. 视频模式选择寄存器**

说明	寄存器字段
配置视频模式类型	DSI_VMC.R.VMT

三种视频模式类型为：

- 00：使用同步脉冲的非突发模式
- 01：使用同步事件的非突发模式
- 1x：突发

#### 视频模式下的LP状态

DSI规范建议定期结束HS传输并使数据通道进入LP状态。这样可实现PHY同步。

每个扫描行应发生一次向LP状态的转换。如果不能，则必须每帧发生一次向LP状态的转换。

#### PHY转换时间配置

在视频模式下，DSI主机需要关于HS和LP状态之间PHY转换时间开销的信息。在[表 22](#)中描述的寄存器中设置这些时间。

**表22. PHY转换时间配置寄存器**

说明	寄存器字段
时钟通道计时器	
配置D-PHY时钟通道从高速到低功耗模式转换所需的最长时间，以通道字节时钟周期来衡量	DSI_CLTCR.HS2LP_TIME
配置D-PHY时钟通道从低功耗到高速模式转换所需的最长时间，以通道字节时钟周期来衡量	DSI_CLTCR.LP2HS_TIME
数据通道计时器	
配置D-PHY数据通道从高速到低功耗模式转换所需的最长时间，以通道字节时钟周期来衡量	DSI_DLTCR.HS2LP_TIME
配置D-PHY数据通道从低功耗到高速模式转换所需的最长时间，以通道字节时钟周期来衡量	DSI_DLTCR.LP2HS_TIME

这些时序允许DSI主机获知LP向HS和HS向LP转换的开销，以便了解在视频消隐周期内是否有足够的时间进入LP模式。

必须在DSI\_DLTCR寄存器中为数据通道置位LP2HS\_TIME和HS2LP\_TIME。

对于时钟通道，仅当使能了“自动时钟通道控制”时，要在DSI\_CLTCR寄存器中置位LP2HS\_TIME和HS2LP\_TIME。

LP2HS\_TIME反映了PHY从LP切换到HS需要的最长时间，而HS2LP\_TIME反映了PHY从HS切换到LP需要的最长时间。

DSI主机将消隐周期与总转换时间进行比较，以便了解是否能够在消隐周期内切换到LP。

如果（周期时序）>（总转换时间），则DSI主机请求D-PHY进入LP。

如果（周期时序）<（总转换时间），则DSI主机在该周期内以HS模式发送消隐数据包。

- PHY转换时间：

- 数据通道

LP2HS= 17 lanebyteclk

HS2LP= 18 lanebyteclk

- 时钟通道

LP2HS= 36 lanebyteclk

HS2LP= 28 lanebyteclk

- 总转换时间的计算：

- 仅数据通道进入LP

转换时间=HS2LP\_TIME（数据）+ LP2HS\_TIME（数据）

- 时钟通道和数据通道进入LP

转换时间= HS2LP\_TIME（时钟）+ LP2HS\_TIME（时钟）

### 计算示例

本例假设：

- HS2LP\_TIME = 18个lane\_byte时钟周期
- LP2HS\_TIME = 17个lane\_byte时钟周期
- HFP周期 = 35个pixel\_clock周期

HS2LP\_TIME +LP2HS\_TIME = 35个lane\_byte时钟周期。

HFP > (HS2LP + LP2HS)，因此DSI主机在HFP区域进入LP。

**注:** 必须认真设置HS2LP\_TIME和LP2HS\_TIME值,以便反映PHY在低功耗和高速模式之间切换所需的最长时间,否则,如果提供给DSI主机的时序短于转换所需的实际时序,则DSI主机可以在短于PHY转换时间的周期内请求LP转换;这会导致视频时序违规。

编程时序不得远大于PHY实际需要的时间,因为DSI主机将使用此时间进行内部计算。

### LP转换配置

在DSI\_VMCR中配置低功耗转换以定义允许进入低功耗模式的视频周期-如果有时间这样做。

[表 23](#)显示了用于每个区域的LP设置编程的寄存器字段。

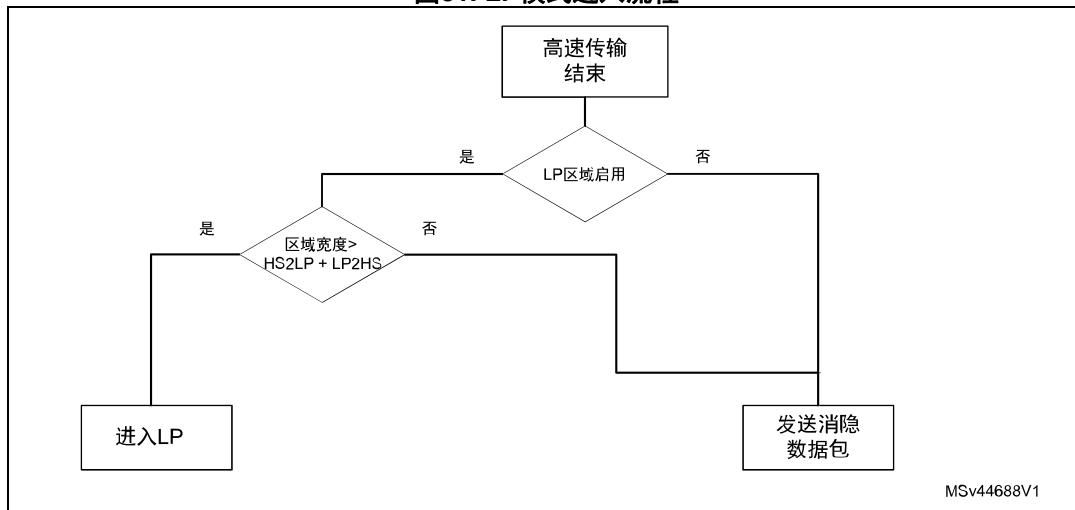
**表23. LP转换配置寄存器**

说明	寄存器字段
低功耗水平前沿使能	DSI_VMCR.LPHFPE
低功耗水平后沿使能	DSI_VMCR.LPHBPE
低功耗垂直激活使能	DSI_VMCR.LPVAE
低功耗垂直前沿使能	DSI_VMCR.LPVFPE
低功耗垂直后沿使能	DSI_VMCR.LPVBPPE
低功耗垂直同步激活使能	DSI_VMCR.LPVSAE

[图 81](#)显示了基本的LP模式进入流程。如果在一个区域内禁用了LP进入,DSI主机将发送消息而不是进入LP模式。

如果在一个区域内使能了LP进入,DSI主机将检查周期长度是否足够用来进入和退出LP模式。这是通过将区域的周期与HS2LP和LP2HS转换时间进行比较来实现的。

图81. LP模式进入流程



- HBP区域内禁用LP模式  
当VMCR.LPHBPE=0时，DSI主机在HBP周期内以HS模式发送消隐数据包。然后，它使用封装像素流数据包发送像素数据（参见图 82）。

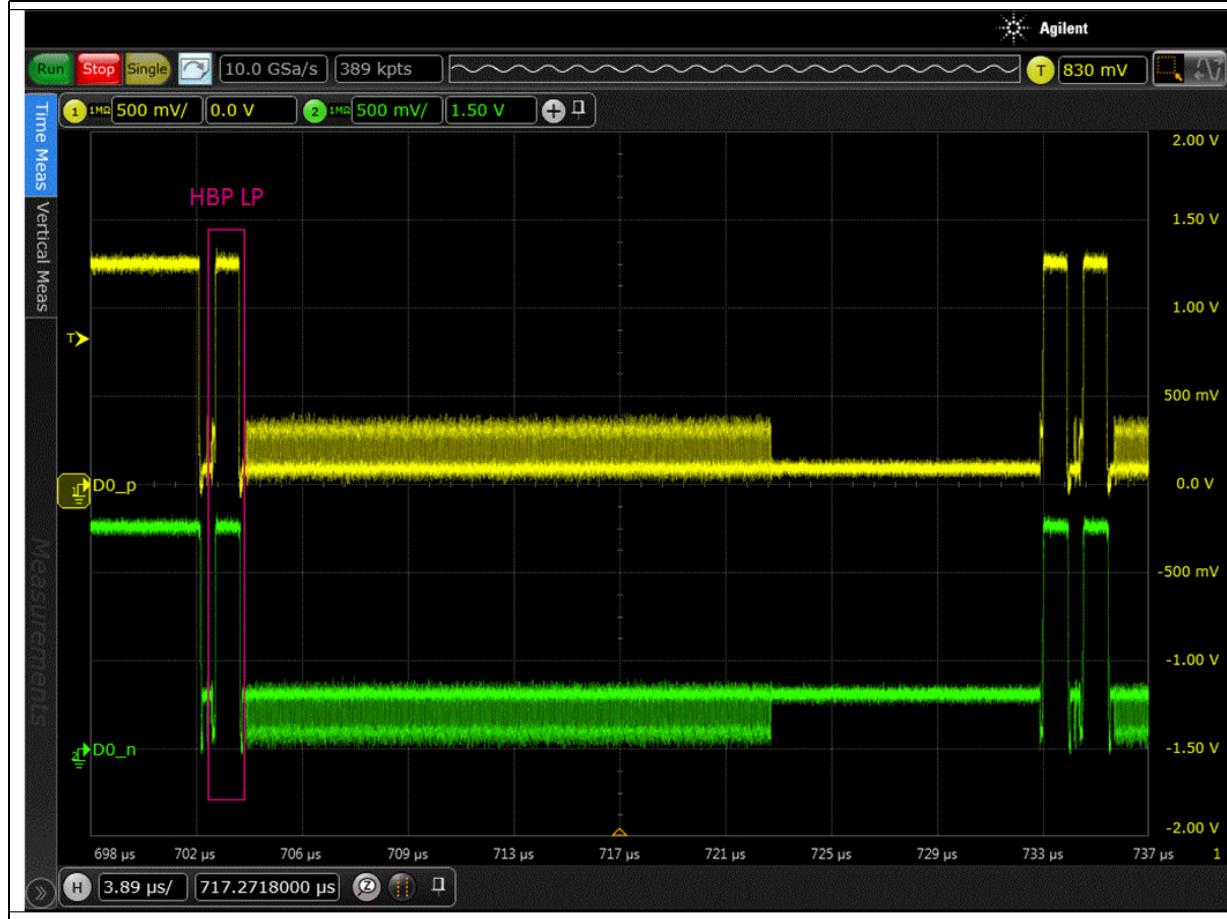
图82. HBP区域内禁用LP



- HBP区域内使能LP模式

当VMCR.LPHBPE=1时，如果HBP周期>HS2LP+LP2HS，则DSI主机在HBP周期内进入LP。然后，它返回HS模式并使用封装像素流数据包发送像素数据（参见图 83）。

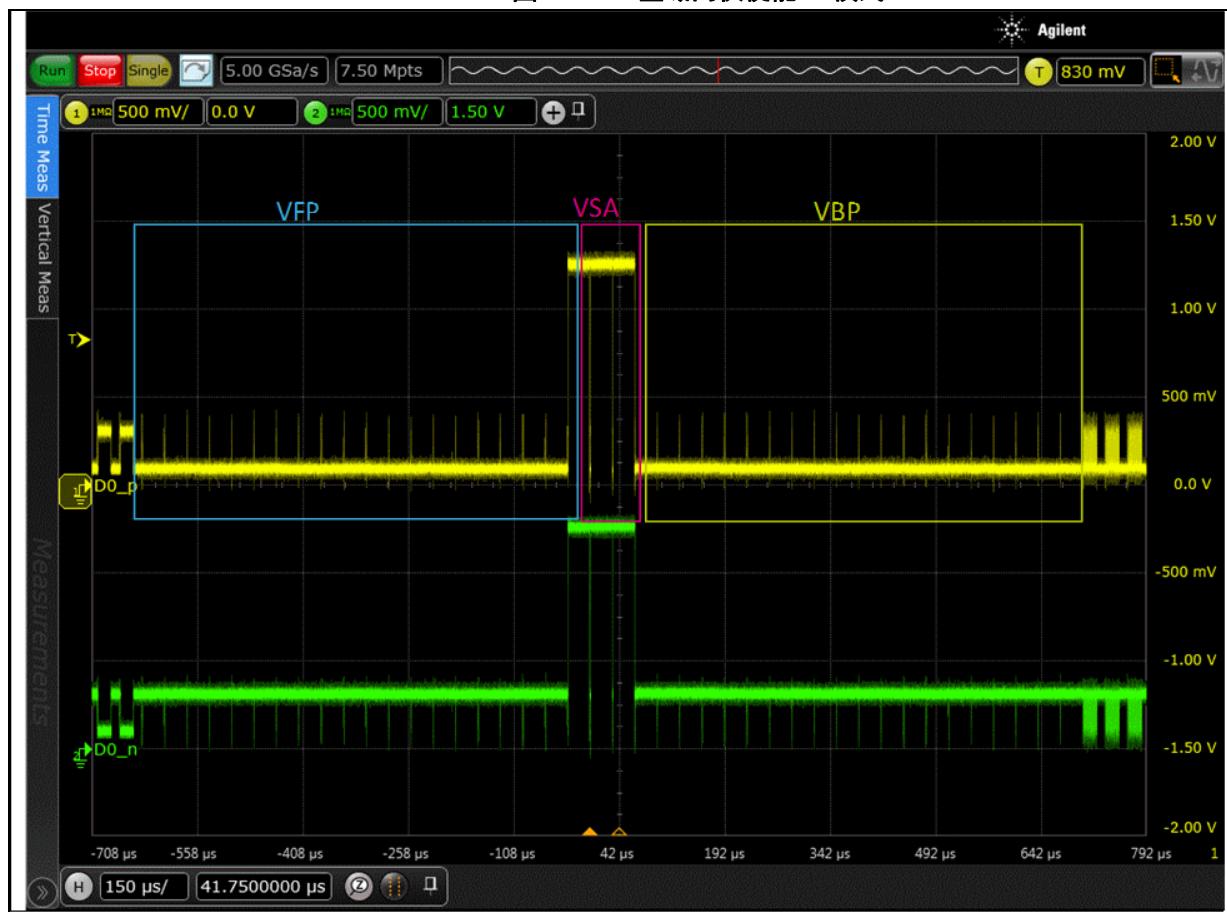
图83. HBP区域内使能LP



- VSA区域内仅使能LP模式

图 84 显示了VSA区域内仅使能LP模式的示例

图84. VSA区域内仅使能LP模式

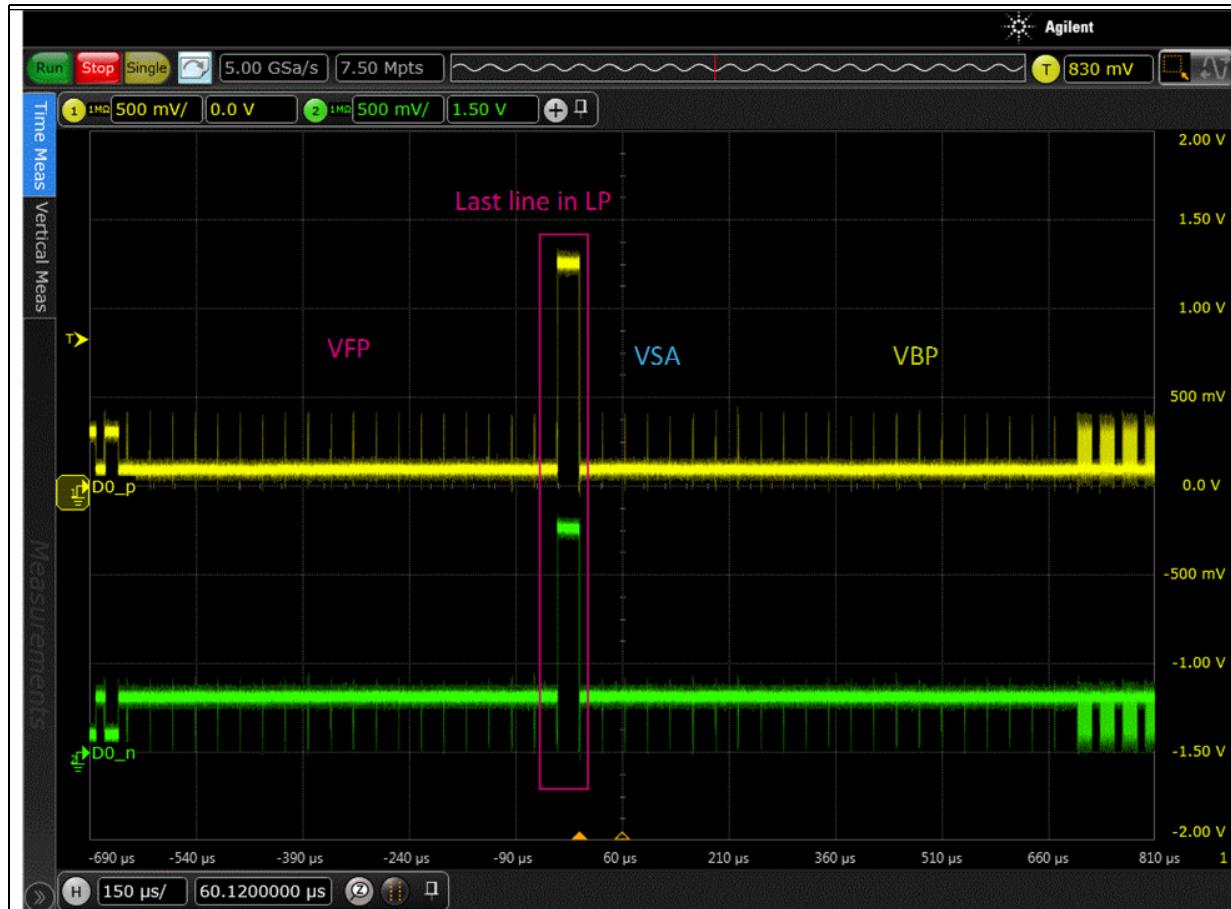


注： 帧的最后一行总是进入LP模式，即使没有为VFP区域使能LP模式。这是为了确保DSI链路在每一帧至少进入LP模式一次。

- 对所有区域禁用LP模式

STM32 DSI主机确保主机在每一帧至少进入LP状态一次。这发生在帧的最后一行，即使没有在DSI\_VMCR寄存器中配置任何要进入LP的区域（参见图 85）。

图85. 低功耗模式下最后一行



### LTDC设置

DSI主机依赖LTDC进行像素数据和视频控制信号的流传输。LTDC配置对DSI主机的正常工作至关重要。

- LTDC视频时序

从显示屏数据手册获取帧的垂直和水平时序。

由于DSI规范建议在每个扫描行进入LP状态，用户可以选择一个水平时序，以使DSI链路能够在每个扫描行进入LP状态一次（只要时间符合显示时间规范）。

关于显示时间示例，请参见 [表 24](#)。

表24. 显示时间示例

项目	规格			单位
	最小值	典型	最大值	
垂直时序				
VSA	2	2	63	HS
VBP	20	20	255	HS
VFP	18	20	255	HS
垂直消隐周期	40	42	1024	HS
VACT	-	480	-	HS
垂直刷新速率	-	60	-	Hz
水平时序				
HSA	1	10	63	PCLK
HBP	3	15	63	PCLK
HFP	4	16	63	PCLK
水平消隐周期	8	32	128	PCLK
HACT		800		PCLK
fPCLK	24	26.36	30.74	MHz

选择水平时序，以允许每个扫描行转换到LP模式一次。

- 以像素时钟为单位的水平时序：  
 $HSA = 5, HBP = 35, HACT = 800, HFP = 35$   
 具有这些HBP和HFP值时，链路在HBP和HFP周期均进入LP模式，详细说明见  
[PHY转换时间配置第 83页](#)。
  - 以行为单位的垂直时序：  
 $VSA = 2, VBP = 20, VACT = 480, VFP = 20$ 
    - LTDC像素时钟设置  
 根据以下公式设置像素时钟：  
 $pixel\_clock = (VSA+VBP+VACT+VFP) \times (HSA+HBP+HACT+HFP) \times \text{帧率}$   
计算示例：
      - 以 $pixel\_clock$ 为单位的水平时序：  
 $HSA = 5, HBP = 35, HACT = 800, HFP = 35$
      - 以行为单位的垂直时序：  
 $VSA = 2, VBP = 20, VACT = 480, VFP = 20$
      - 刷新速率 = 60 fps。
      - $pixel\_clock = (2+20+480+20) \times (5+35+800+35) \times 60 = 522 \times 875 \times 60 = 27.4 \text{ MHz}$
- 请参见[第 7节：DSI主机性能](#)以获取支持的最大像素时钟的更详细信息。

## DSI主机视频时序

必须为DSI主机提供显示视频时序信息。[表 25](#) 显示了用于配置视频时序的DSI主机寄存器。

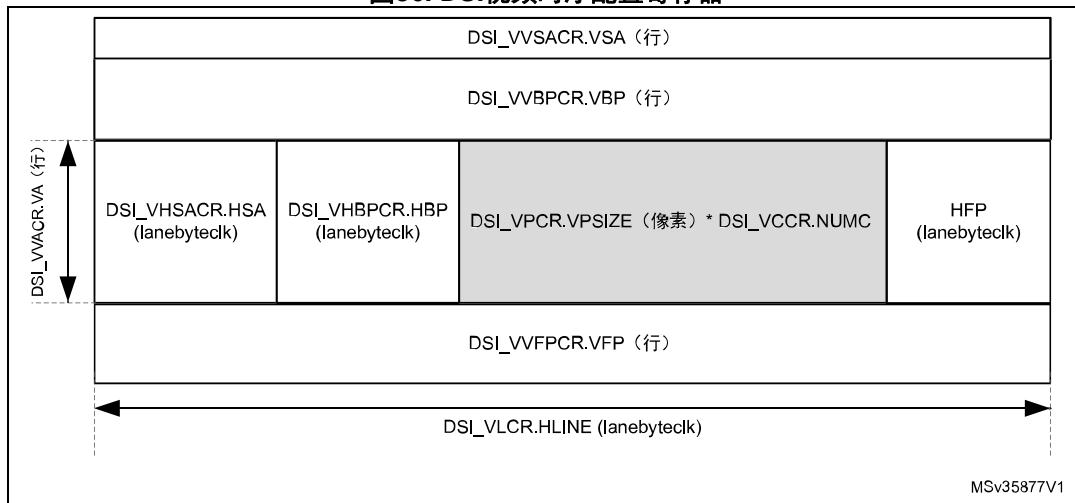
**表25. DSI主机视频时序寄存器**

说明	寄存器字段
<b>帧水平时序</b>	
设置以通道字节时钟周期为单位的水平同步激活（HSA）持续时间	DSI_VHSACR.HSA
设置以通道字节时钟周期为单位的水平后沿（HBP）持续时间	DSI_VHBPCR.HBP
设置以通道字节时钟周期进行计数的水平行时间（HSA+HBP+HACT+HFP）	DSI_VLCR.HLINE
<b>帧垂直时序</b>	
配置以水平行数来衡量的垂直激活（VSA）持续时间	DSI_VVSACR.VSA
配置以水平行数来衡量的垂直后沿（VBP）持续时间	DSI_VVBPCR.VBP
配置以水平行数来衡量的垂直前沿（VFP）持续时间	DSI_VVFPCR.VFP
配置以水平行数来衡量的垂直激活（VACT）持续时间	DSI_VVACR.VA

DSI主机中视频时序的长度必须与LTDC中的相同。LTDC中的值表示为pixel\_clock周期数，DSI中的编程值表示为通道字节时钟周期数。

[图 90](#)中描述了视频时序寄存器。

图86. DSI视频时序配置寄存器



- 帧水平时序
  - 配置水平同步持续时序 (DSI\_VHSACR.HSA)，以通道字节时钟周期数表示LTDC水平同步激活周期的时间长度。
  - 配置水平后沿持续时序 (DSI\_VHBPCR.HBP)，以通道字节时钟周期数表示LTDC水平后沿周期的时间长度。
  - 配置水平行时序 (DSI\_VLCR.HLINE)，以通道字节时钟周期数表示LTDC视频行的时间长度。
- 帧垂直时序
  - 使用LTDC垂直同步激活周期中现有的行数配置垂直同步持续时序 (DSI\_VVSACR.VSA)。
  - 使用LTDC垂直后沿周期中现有的行数配置垂直后沿持续时序 (DSI\_VVBPCR.VBP)。
  - 使用LTDC垂直前沿周期中现有的行数配置垂直前沿持续时序 (DSI\_VVFFPCR.VFP)。
  - 使用LTDC垂直激活周期中现有的行数配置垂直激活持续时序 (DSI\_VVACR.VA)。

配置示例：

$DSI\_VLCR.HLINE = (HSA + HBP + HACT + HFP) \times (\text{lane\_byte\_clk} / \text{pixel\_clock}) = (5 + 35 + 800 + 35) \times (62.5 / 27.4) = 1995$

$DSI\_VHSACR.HSA = HSA \times (\text{lane\_byte\_clk}/\text{pixel\_clock}) = 5 \times (62.5 / 27.4) = 11$

$DSI\_VHDPCR.HBP = HBP \times (\text{lane\_byte\_clk}/\text{pixel\_clock}) = 35 \times (62.5 / 27.4) = 80$

垂直行配置：

$DSI\_VVSACR.VSA = 2$

$DSI\_VVBPCR.VBP = 20$

$DSI\_VVFFPCR.VFP = 20$

$DSI\_VVACR.VA = 480$

## DSI时钟设置

DSI时钟取决于使用的视频模式：

- 突发模式：

在突发模式下，可以将DSI时钟设置为最大值，以使链路处于低功耗模式的周期更长。

支持的最大lane\_byte\_clock为62.5 MHz。

- 非突发模式：

应配置DSI非突发模式，使DSI输出像素比与LTDC接口输入像素比相匹配。下一节对此进行了讨论。

## DSI视频数据包参数

[表 26](#) 显示了配置视频激活区域内的视频数据包所需的DSI主机寄存器。

**表26. DSI视频数据包参数寄存器**

说明	寄存器字段
以像素为单位配置视频数据包大小	DSI_VPCR.VPSIZE
配置块数	DSI_VCCR.NUMC
以字节为单位配置空包大小	DSI_VNPCR.NPSIZE

视频数据包参数配置取决于选择的视频模式：

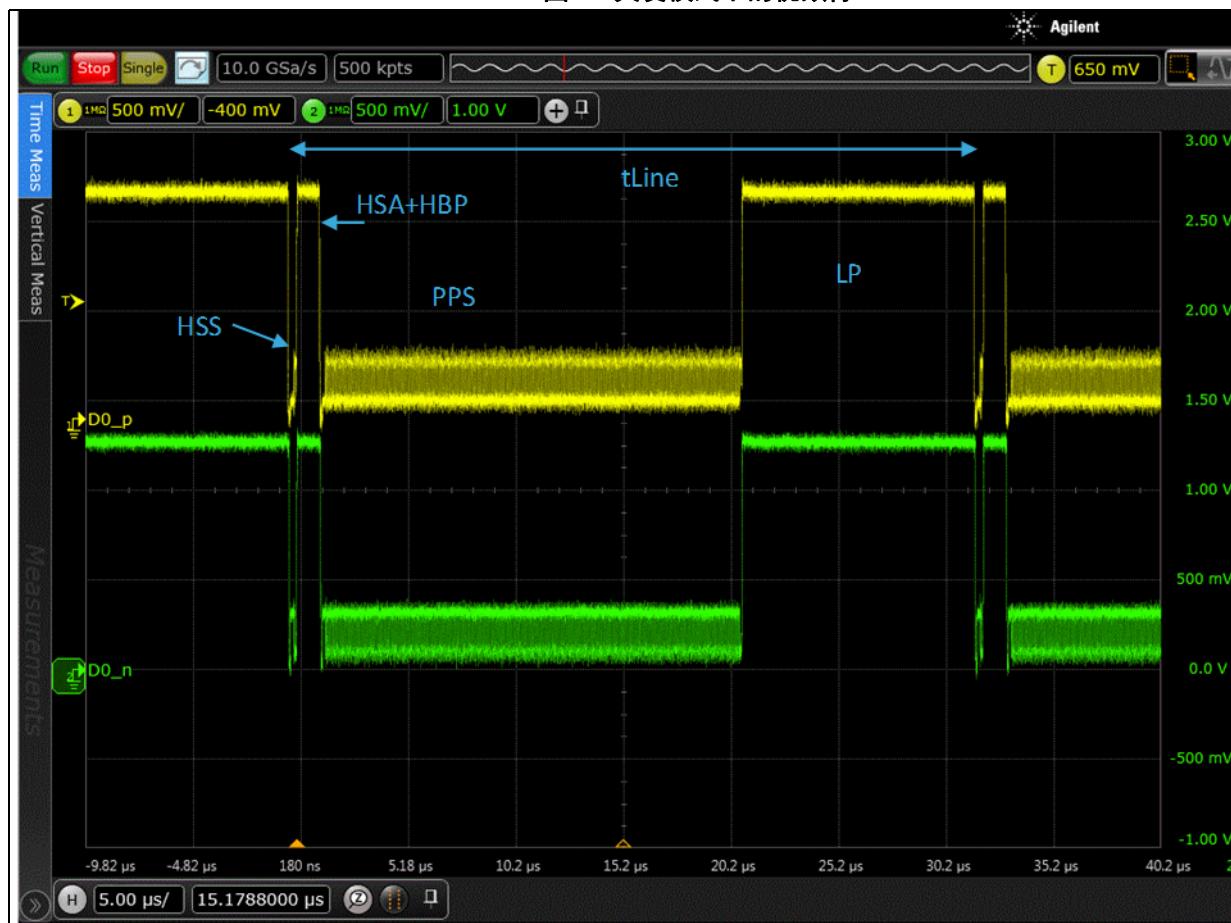
- 突发模式

在突发模式下，VPSIZE必须等于整行的长度（以像素为单位）。因此，可将块数设置为0，以便在一个数据包中传输视频行。

在发送封装像素流后，为了节能，链路进入LP模式。无需空包（NPSIZE = 0）。

[图 87](#)显示了突发模式下的视频行。

图87. 突发模式下的视频行



- 非突发模式

应配置DSI非突发模式，使DSI输出像素比与LTDC接口输入像素比相匹配。这是通过将像素行（HACT区域）划分为若干像素块并选择性地将它们与空包进行交织来实现的。

以下等式允许在非突发模式下设置DSI主机传输参数。两个等式用于平衡在DSI上输出像素所需的时间（等式右侧）与从LTDC输入像素所需的时间（等式左侧）。

- 当空包启用时：

**公式1**

$$\text{lanebyteclkperiod} \times \text{NUMC} (\text{VPSIZE} \times \text{bytes\_per\_pixel} + 12 + \text{NPSIZE}) / \text{number\_of\_lanes} = \text{pixels\_per\_line} \times \text{LTDC\_clock\_period}$$

- 当空包禁用时：

**公式2**

$$\text{lanebyteclkperiod} \times \text{NUMC} (\text{VPSIZE} \times \text{bytes\_per\_pixel} + 6) / \text{number\_of\_lanes} = \text{pixels\_per\_line} \times \text{LTDC\_clock\_period}$$

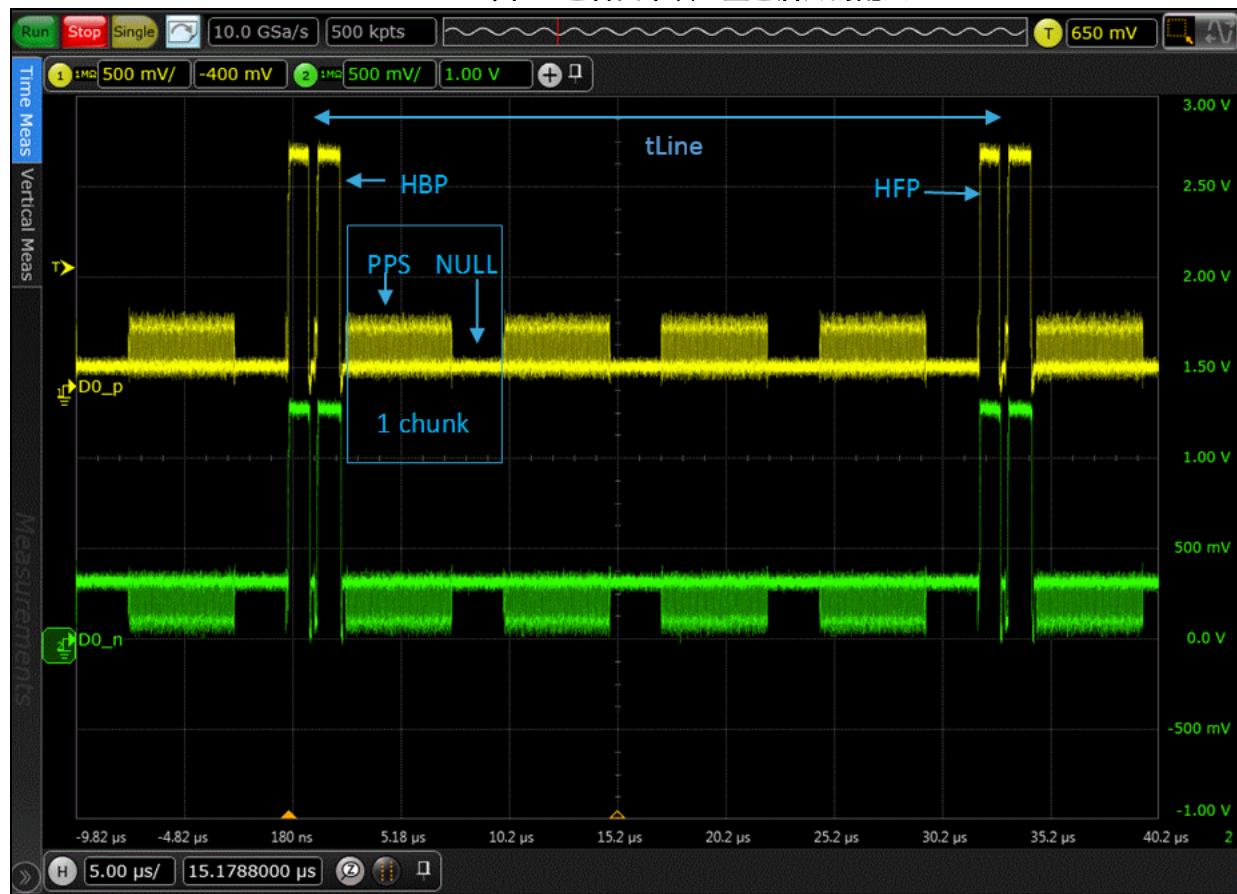
注:  $VPSIZE = \text{视频数据包大小}, NUMC = \text{块数}.$

**包含四个块且空包启用的配置示例:**

- 使用视频时序参数计算Pixel\_clock (参见[LTDC设置第 89页](#)) ; pixel\_clock = 27.429 MHz。
- 在本例中, Lane\_byte\_clk被设置为它的最大速度: 62.5 MHz。
- 设置每个数据包的像素和块数以与行宽匹配:  
以像素为单位的行宽 =  $VPSIZE \times NUMC$ 。  
 $VPSIZE$ 取决于显示屏中的FIFO大小。在本例中, 我们将VPSIZE设置为200像素, 并将块数设置为4。
- 使用上述等式2, DSI输出像素的速度快于从LTDC输入像素的速度:  
DSI像素输出时间 =  $4(200 \times 3 + 6) / (2 \times 62.5 \text{ MHz}) = 19 \mu\text{s}$   
LTDC接口像素输入时间 =  $800 / 27.4 \text{ MHz} = 29 \mu\text{s}$   
这要求使用空包, 以便使用上述等式1平衡LTDC和DSI吞吐率。
- 计算空包大小:  
使用上述等式1计算NPSIZE。  
 $2 \times (400 \times 3 + 12 + \text{NPSIZE}) / (62.5 \times 2) = 800 \times 1 / 27.4$   
NPSIZE=299字节。

[图 87](#)显示了包含四个块且空包启用的视频激活行:

**图88. 包含四个块且空包启用的配置**



### 包含四个块且空包未启用的配置示例

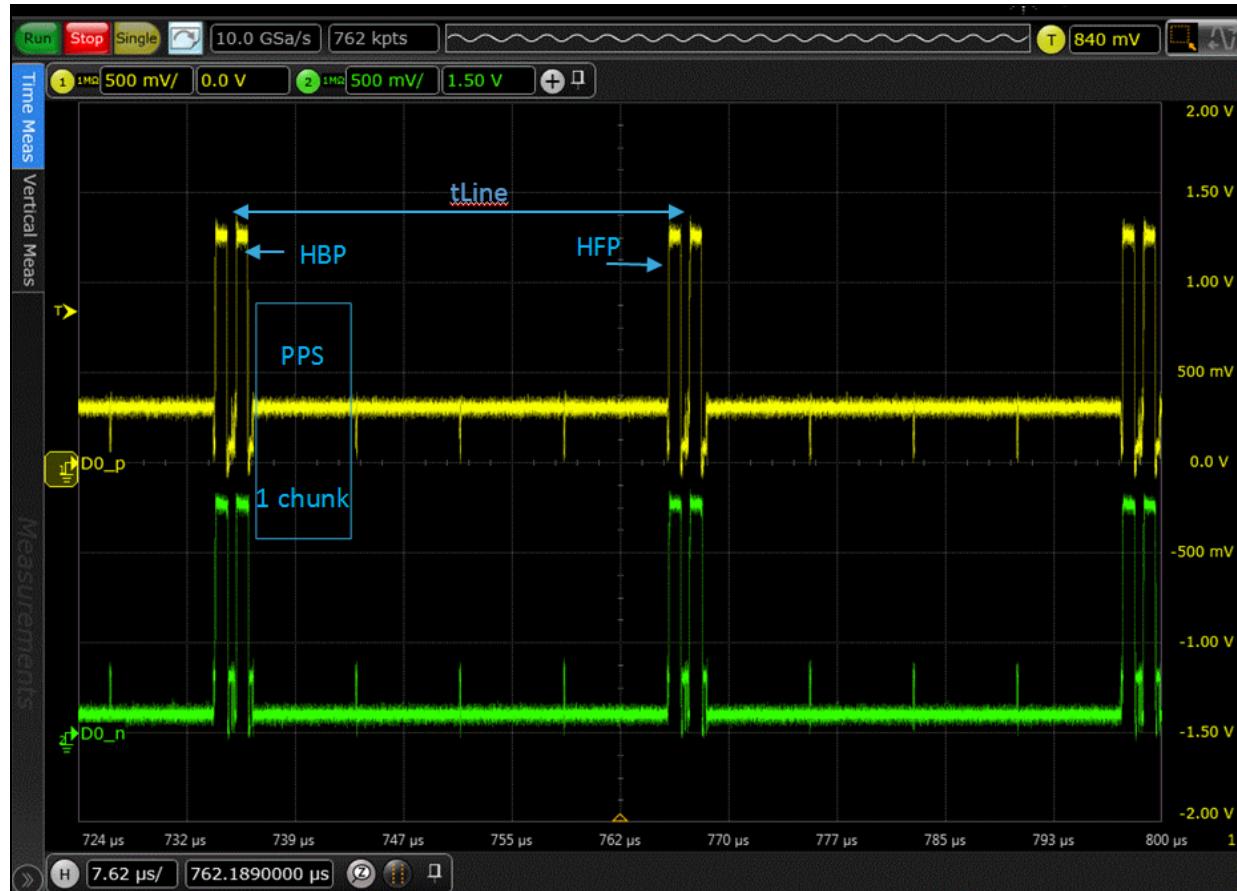
为避免使用空包，必须使用上述等式2平衡DSI和LTDC吞吐率。

本例使用四个块的配置，lane\_byte\_clk的计算如下：

- $\text{Lane\_byte\_clk} = \text{pixel\_clock} \times \text{NUMC} \times (\text{VPSIZE} \times \text{bytes\_per\_pixel} + 6) / (\text{pixels\_per\_line} \times \text{num\_lanes}) = 41.24 \text{ MHz}$

[图 89](#)显示了包含四个块且没有空包的视频激活行，DSI链路为41.2 MHz。

**图89. 包含四个块且空包未启用的配置**



注：必须根据显示屏的内部FIFO大小设置视频数据包大小（VPSIZE）和后续的块数（NUMC）。例如，如果内部FIFO仅能容纳200个像素，则块数必须大于4，以确保VPSIZE等于或小于200个像素。

### 视频模式下的指令传输

在视频传输过程中，DSI主机允许在消隐区域通过APB通用接口发送指令。可在低功耗或高速模式下发送指令。

如果在HS模式下传输指令，DSI主机将自动确定每个指令可以发送的区域，无需编程或计算。

对于LP指令，DSI主机需要来自用户的输入来确定可以传输指令的合适区域。

- 指令传输模式

[表 27](#) 显示了视频模式下在HS和LP指令传输模式之间进行选择时使用的寄存器位。

**表27. 指令传输模式寄存器**

说明	寄存器字段
配置是否在低功耗模式下传输指令	DSI_VMCR.LPCE

如果DSI\_VMCR.LPE = 1，则在LP模式下发送指令，否则在HS模式下发送指令。

**注：**一些显示屏需要在LP模式下发送初始化指令。这种情况下，必须在初始化阶段使能LP指令传输。

- LP指令数据包大小

当LP指令传输使能时，用户必须使用[表 28](#)中定义的寄存器字段，将最大允许数据包大小（以字节为单位）通知DSI主机。

**表28. LP指令数据包大小寄存器**

说明	寄存器字段
最大数据包大小	DSI_LPMCR LPSIZE
VACT最大数据包大小	DSI_LPMCR VLPSIZE

当在HS视频模式传输（DSI\_VMCR.LPCE=1）期间将DSI主机配置为发送低功耗（LP）指令时，必须以字节为单位计算在水平前沿（HFP）、垂直同步激活（VSA）、垂直后沿（VBP）和垂直前沿（VFP）区域内以LP模式传输指令的可用时间。

- LPSIZE：VACT区域的最大数据包大小  
此字段用于LP模式下的指令传输。它定义在VSA、VBP和VFP区域内可嵌入行中的最大数据包的大小（以字节为单位）。
- VLPSIZE：VACT区域内的最大数据包大小  
此字段用于LP模式下的指令传输。它定义VACT区域的HFP期间可嵌入行中的最大数据包的大小（以字节为单位）。

**注：**正确地配置LPSIZE和VLPSIZE很重要。如果LPSIZE和VLPSIZE大于实际可用时间，则可能导致视频时序违规。另一方面，如果LPSIZE和VLPSIZE大幅小于可用时间，则许多指令会被延迟到帧的最后一行。

**注：**建议避免在帧的最后一行发送多行指令。如果必须发送大量指令，用户可以禁用DSI视频模式并在DSI指令模式下发送指令，然后重新使能视频模式。

LPSIZE计算

图 90 显示了使用同步脉冲的视频非突发模式下的LPSIZE计算。

**图90. 使用同步脉冲的非突发模式下的LPSIZE计算**

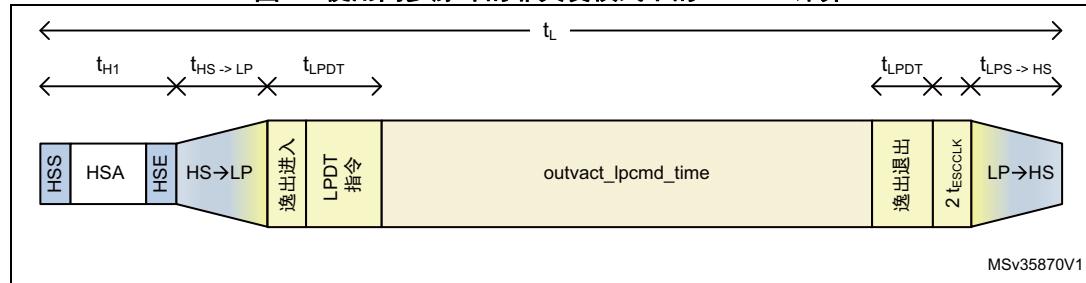
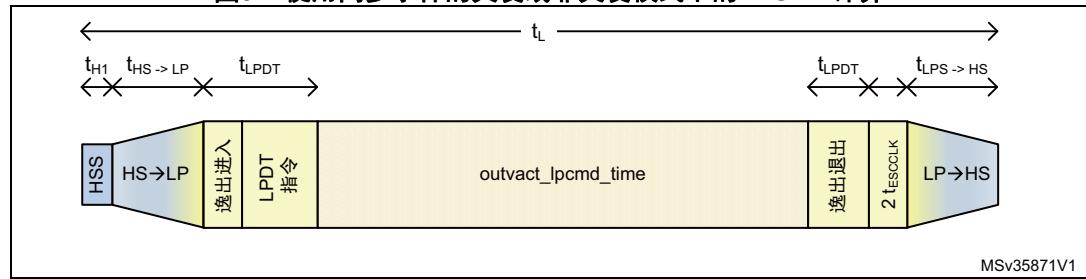


图 91 显示了使用同步事件的视频突发模式和视频非突发模式下的LPSIZE计算。

**图91. 使用同步事件的突发或非突发模式下的LPSIZE计算**



其中

$$t_L = \text{行时序} = (\text{HSA} + \text{HBP} + \text{HACT} + \text{HFP}) / \text{PCLK};$$

$$t_{H1} = \text{HSA脉冲的时序} = t_{\text{HSA}} = \text{HSA} / \text{PCLK}$$

使用同步脉冲的非突发模式：

$$t_{HS \rightarrow LP} = \text{进入低功耗模式的时序};$$

使用同步事件的突发模式或非突发模式：

$$t_{H1} = \text{发送HSS数据包} = 4 \text{字节的时序} / (\text{lane\_byte\_clk} \times \text{Number\_Lanes})$$

$$t_{HS \rightarrow LP} = \text{进入低功耗模式的时序};$$

$$t_{LP \rightarrow HS} = \text{退出低功耗模式的时序};$$

$t_{LPDT}$ =与逸出模式逸出、LPDT指令和逸出退出相关的D-PHY时序。根据D-PHY规范，此值在LP模式下始终为11位（或22个TX逸出时钟周期）；

$$t_{ESCLK} = \text{逸出时钟周期} = \text{DSI\_CCR.TXECKDIV} / \text{lane\_byteclk}$$

$2 t_{ESCLK}$  = DSI主机实现引起的延迟。

计算示例：

$$t_L = (5 + 35 + 800 + 35) / 27.429 \text{ MHz} = 31.9 \text{ us}$$

$$t_{H1}$$

使用同步脉冲的非突发模式：

$$t_{H1}=t_{HSA} = 5/27,429 \text{ MHz} = 0.182 \text{ us}$$

使用同步事件的突发模式或非突发模式：

$$t_{H1} = 4 / (62.5 \times 2) = 0.032 \text{ us}$$

$t_{HS \rightarrow LP} = 291 \text{ ns}$  (这是D-PHY特定的时序)

$t_{LP \rightarrow HS} = 264 \text{ ns}$  (这是PHY特定的时序)

$$t_{ESCCLK} = 4 / 62.5 = 0.064 \text{ ns}$$

$$t_{LPDT} = 22 \times t_{ESCCLK} = 1.408$$

使用同步事件的突发和非突发模式下的LPSIZE计算：

$$\text{LPSIZE} = (31.9 - (0.032 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 \times 8 \times 0.064) = 29.07$$

使用同步脉冲的非突发模式下的LPSIZE计算：

$$\text{LPSIZE} = (31.9 - (0.182 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 \times 8 \times 0.064) = 28.93$$

为了获得一些余量，两种情况下要编程的LPSIZE均为28字节。

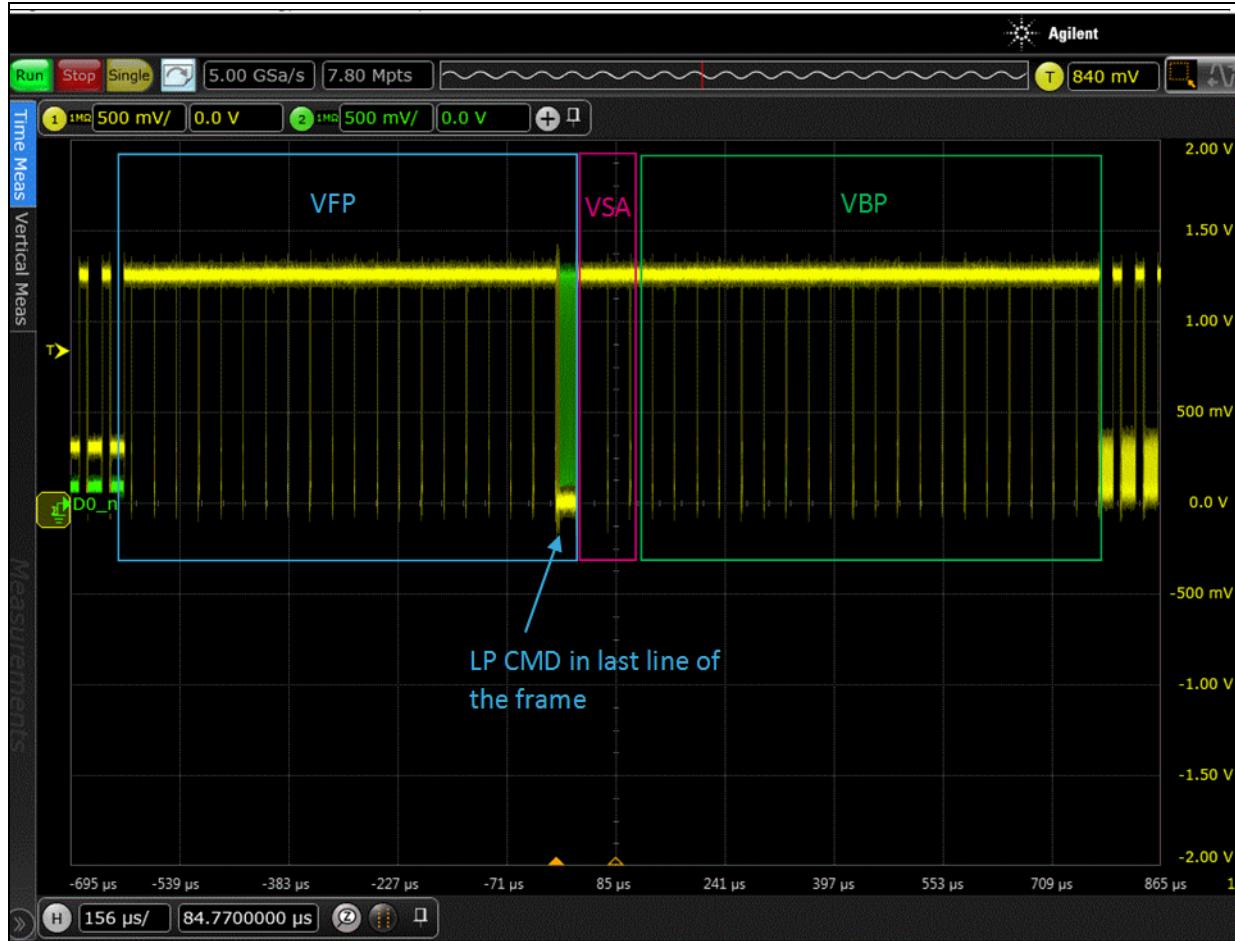
[图 92](#) 显示了消隐区域内以LP模式发送的28字节指令。指令嵌入消隐周期不会导致时线违规。

图92. 突发模式下VFP期间的28字节LP指令



图 93 显示了未嵌入激活或消隐区域的29字节指令。这种情况下，DSI 主机将指令延迟到帧的最后一行。

图93. 延迟到最后一行的29字节LP指令



#### VLPSIZE计算

图 94 显示了使用同步脉冲的视频非突发模式下的VLPSIZE计算。

图94. 使用同步脉冲的非突发模式下的VLPSIZE计算

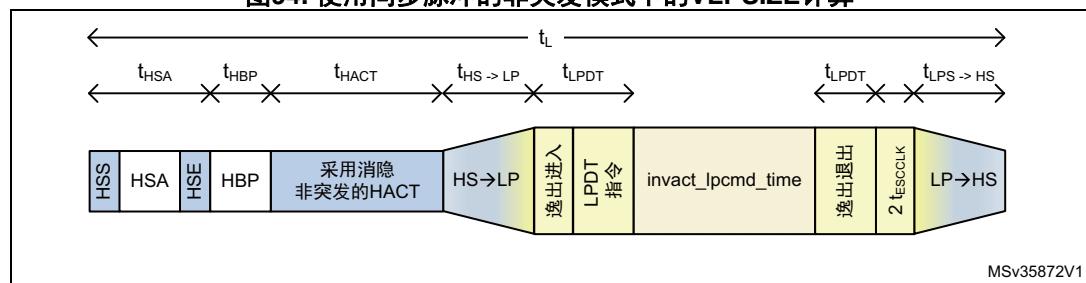


图 95 显示了使用同步事件的视频非突发模式下的VLPSIZE计算。

图95. 使用同步事件的非突发模式下的VLPSIZE计算

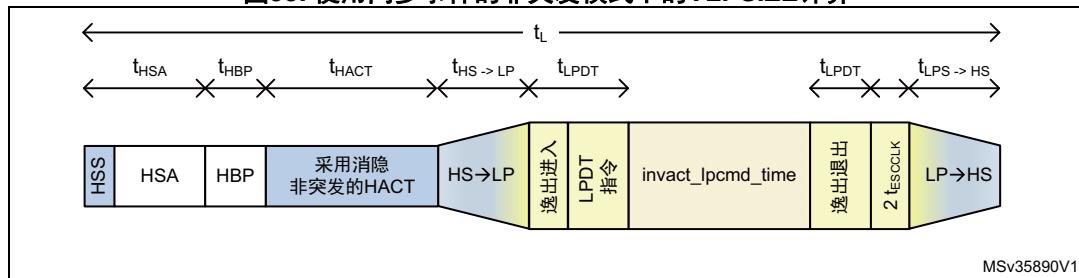
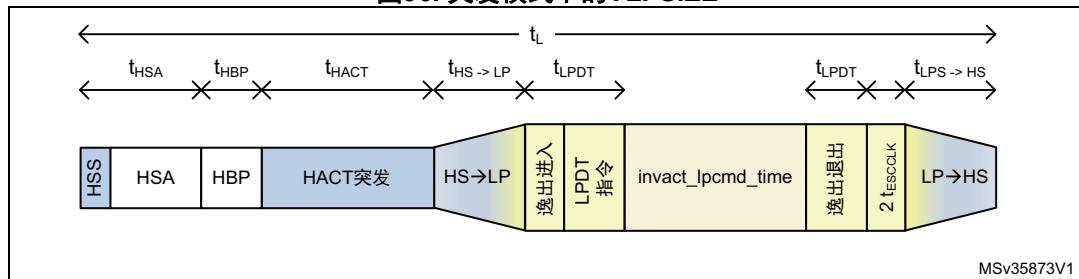


图 96 显示了视频突发模式下的VLPSIZE计算。

图96. 突发模式下的VLPSIZE



其中

$$t_L = \text{行时序} = (\text{HSA} + \text{HBP} + \text{HACT} + \text{HFP}) / \text{PCLK};$$

$$t_{\text{HSA}} = \text{HSA脉冲的时序} = \text{HSA} / \text{PCLK};$$

$$t_{\text{HBP}} = \text{水平后沿的时序} = \text{HBP} / \text{PCLK};$$

$$t_{\text{HACT}} = \text{视频激活的时序}.$$

非突发模式: pixels\_per\_line / PCLK

突发模式: 视频激活经过时序压缩, 计算如下

$$t_{\text{HACT}} = (\text{VPSIZE} \times \text{Bytes_per_Pixel}) / (\text{Number_Lanes} \times t_{\text{Lane_byte_clk}});$$

$$t_{\text{ESCLK}} = \text{逸出时钟周期} = \text{DSI_CCR.TXECKDIV} / \text{lane_byteclk}$$

计算示例:

$$t_L = \text{行时序} = (5 + 35 + 800 + 35) / 27.4 \text{ MHz} = 31.9 \text{ us}$$

$$t_{\text{HSA}} = 5 / 27.4 \text{ MHz} = 0.182 \text{ us}$$

$$t_{\text{HBP}} = 35 / 27.4 \text{ MHz} = 1.27 \text{ us}$$

$$t_{\text{HACT}}$$

非突发模式:

$$t_{\text{HACT}} = 800 / 27.4 \text{ MHz} = 29.16 \text{ us}$$

突发模式:

$$t_{\text{HACT}} = (800 \times 3) / (2 \times 62.5 \text{ MHz}) = 19.2 \text{ us}$$

突发模式下的VLPSIZE

$$\text{VLPSIZE} = (31.9 - (0.182 + 1.27 + 19.2 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 \times 8 \times 0.064) = 8.94$$

突发模式VACT区域允许8字节长数据包

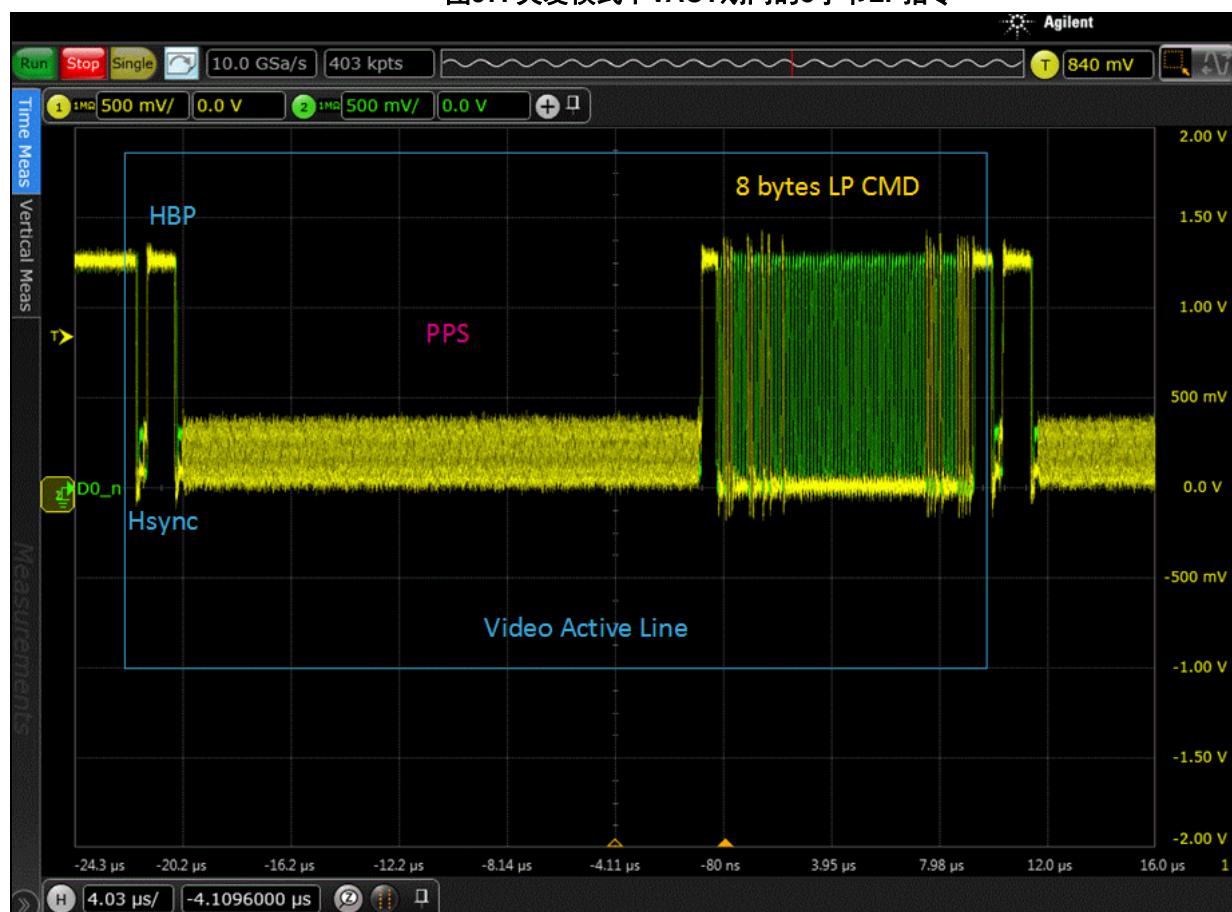
非突发模式下的VLPSIZE

$$\text{VLPSIZE} = (31.9 - (0.182 + 1.27 + 29.16 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 \times 8 \times 0.064) = -0.78$$

è VACT区域不允许数据包

[图 97](#)显示了视频激活行期间以LP模式发送的8字节指令。发送指令不会导致激活行的时序违规。

图97. 突发模式下VACT期间的8字节LP指令



注：对于允许的字节数（LPSIZE和VLPSIZE），用户可使用10%的余量来避免视频时序问题。

### 帧回应

为确保显示屏正确地接收了帧，DSI主机可以要求帧回应。

[表 29](#)显示了视频模式下用于使能帧回应的寄存器位。

表29. 帧回应寄存器

说明	寄存器字段
帧总线转向回应使能	DSI_VMCR.FBTAAE

在帧的最后一行，主机执行BTA流程。然后，显示屏获得总线所有权，如果之前的所有数据包均已接收且无错误（意味着显示屏已正确接收帧），则发送回应触发信号。之后，显示屏执行BTA序列以将总线所有权还给DSI主机。如果显示屏遇到之前数据包中的错误，它将以错误报告做出响应。

注：必须将DSI\_PCR.BTAE置为1，以便使能总线转向（BTA）请求。

图98和图99显示了视频模式下帧回应触发信号的示例。

图98. 帧回应示例



图99. 帧回应放大图



## 5.2.2 通过LTDC接口实现适配指令模式

本节描述与适配指令模式相关的DSI主机设置。

### DSI指令模式

必须同时在DSI主机和DSI Wrapper中选择适配指令模式（参见表 30）。

表30. DSI指令模式寄存器

说明	寄存器字段
选择视频传输模式：视频模式或适配指令模式	DSI_WCFGR.DSIM
将DSI主机设置为视频或指令模式	DSI_MCR.CMDM

在DSI Wrapper配置寄存器中选择适配指令模式（DSI\_WCFGR.DSIM=1）。

在DSI主机模式配置寄存器中选择指令模式（DSI\_MCR.CMDM=1）。

### 停止等待时序配置

[表 31](#) 中的寄存器字段用于配置停止状态后请求高速传输的最小等待周期。

**表31. 停止等待时序配置寄存器**

说明	寄存器字段
配置停止状态后请求高速传输的最小等待周期。	DSI_PCONFR.SW_TIME

发射器D-PHY为了保证所有数据通道在新的高速传输开始前处于停止状态，必须使用 SW\_TIME。

显示屏在接收新的高速传输之前可能还需要一段特定的时序。必须在显示屏的数据手册中检查此时序。

编程值必须是主机SW\_Time和显示屏SW\_TIME中的最大者。

**注：**DSI主机的最小SW\_Time是10 lanebyclk指令模式。

### 指令大小 (CMDSIZE)

用户必须以像素为单位，定义适配指令模式中用于刷新显示屏的GRAM的DCS长写指令（WMS 和WMC）大小。在[表 32](#)所述的寄存器中设置指令大小。

**表32. 指令大小寄存器**

说明	寄存器字段
该字段以像素为单位配置LTDC Write memory指令的最大允许大小	DSI_LCCR.CMDSIZE

DSI主机像素FIFO大小为960个32位字。这意味着：

- 在24 bpp模式下，该字段应不超过1280像素。
- 在16 bpp模式下，该字段应不超过1920像素。

### 撕裂效应设置

使用[表 33](#)中所述的寄存器管理指令模式下的撕裂效应。

表33. 撕裂效应设置寄存器

说明	寄存器字段
TE源	DSI_WCFG.R.TESRC
TE极性	DSI_WCFG.R.TEPOL
撕裂效应回应请求使能	DSI_CMCR.TEARE

- 通过链路报告撕裂效应

- TE源

为了通过链路报告TE，必须将DSI Wrapper配置寄存器的撕裂效应源（DSI\_WCFG.R.TESRC）置为0。

- TE回应请求

当通过链路报告撕裂效应时，必须置位DSI主机指令模式配置寄存器的DSI\_CMCR.TEARE位。

注：

注意：必须置位DSI主机协议配置寄存器的DSI\_PCR.BTAE位，以便允许TE报告的反向通信。

- 通过引脚报告撕裂效应

- TE源

为了通过外部引脚报告TE，必须将撕裂效应源（TESRC）置为1。

- TE极性

当通过GPIO链路使用TE时，在编程时必须考虑显示屏中的TE极性。DSI主机支持两种极性（低和高）

### 刷新模式

DSI主机支持两种开始显示屏刷新操作的模式。[表 33](#)显示了用于选择刷新模式的寄存器字段。

表34. 刷新模式寄存器

说明	寄存器字段
自动刷新	DSI_WCFG.R.AR

如果每次接收到撕裂效应事件时需要自动更新显示屏，则置位DSI Wrapper配置寄存器（DSI\_WCFG.R）的自动刷新（AR）位。

- 自动刷新：在接收到TE事件后自动置位DSI\_WCR.LTDCEN。
- 手动刷新：在接收到TE事件后，软件必须通过置位DSI\_WCR.LTDCEN位以刷新GRAM。

### LTDC设置

在适配指令模式下，DSI主机从LTDC输入像素流。当DSI以适配指令模式工作时，LTDC像素速率和视频时序具有特定的配置。

- LTDC像素时钟设置

像素时钟频率无需匹配显示屏像素时钟，因为它依赖其内部控制器生成时序信息。

在适配指令模式下，必须选择LTDC像素时钟以确保以下要求：

- 最小像素时钟必须足够快，以确保GRAM刷新时间短于显示屏内部刷新频率，从而避免视觉伪影。
- 最大像素时钟必须与系统限值一致，以免LTDC侧发生FIFO欠载问题。

请参见[第 7 节：DSI 主机性能](#)了解关于适配指令模式下像素时钟最小值和最大值的更详细信息。

- LTDC视频时序

由于显示屏不依赖于主机获取时序信息，因此可以将所有垂直和水平消隐周期（HSA、HBP、HFP、VSA、VBP、VFP）设置为最小值1，但用户必须分别以行长和每帧行数正确地设置HACT和VACT。

### 指令传输模式

可在高速或低功耗模式下发送或接收所有指令，具体取决于指令类型。在DSI主机指令模式配置寄存器（DSI\_CMCR）中为其中的每一个编程专用配置位（参见[表 35](#)）。

**表35. 指令传输寄存器**

说明	寄存器字段
最大读数据包大小	DSI_CMCR.MRDPS
DCS长写传输	DSI_CMCR.DLWTX
DCS短读0参数传输	DSI_CMCR.DSR0TX
DCS短读1参数传输	DSI_CMCR.DSW1TX
DCS短写0参数传输	DSI_CMCR.DSW0TX
通用长写传输	DSI_CMCR.GLWTX
通用短读2参数传输	DSI_CMCR.GSR2TX
通用短读1参数传输	DSI_CMCR.GSR1TX
通用短读0参数传输	DSI_CMCR.GSR0TX
通用短写2参数传输	DSI_CMCR.GSW2TX
通用短写1参数传输	DSI_CMCR.GSW1TX
通用短写0参数传输	DSI_CMCR.GSW0TX

注：

一些显示屏需要在LP模式下发送初始化指令。这种情况下，必须在初始化阶段的LP模式下配置用于显示屏初始化的指令。

在初始化阶段后，可以重新配置指令以在高速模式下发送。这对于DCS长写指令特别重要，这类指令用于容纳显示屏刷新期间使用的WMS和WMC DCS指令。

## 回应请求

DSI主机可在发送每条指令后请求回应。这是通过置位DSI\_CMCR.ARE位来使能的（参见 [表 36](#)）。

**表36. 回应请求寄存器**

说明	寄存器字段
在每次数据包传输后使能回应请求	DSI_CMCR.ARE

当此功能使能时，DSI主机在发送每个指令后开始BTA流程。显示屏获得总线所有权，并回复一个应答出发，在出错时回复错误报告。

然后，显示屏发送BTA以将总线所有权还给DSI主机。

**注：**在适配指令模式下的刷新操作期间，应避免使用此功能，因为它会增加许多开销，延长显示屏刷新时间。

此功能十分有用，可提高系统安全性并尽快检测到错误。因此，其使用情况取决于应用需要。

[图 100](#)显示的是使能了回应请求的写指令示例。

图100. 使能了回应请求的通用短写指令



## 6 STM32CubeMX配置示例

STM32CubeMX工具可用于配置DSI主机外设。本节介绍在不同工作模式下配置DSI主机所需的基本配置步骤：视频突发模式、具有同步脉冲的视频非突发模式和适配指令模式。本节还提供在不同工作模式下配置LTDC和DSI主机的软件代码示例。关于更多完整示例，用户可参考STM32Cube示例。在STM32F769I探索板上有生成好的示例。

由于DSI主机使用LTDC作为视频推流器，因此必须完成LTDC配置。这些示例显示了基本的LTDC配置。应用笔记AN4861中提供了关于LTDC配置的更多信息。

### 6.1 DSI主机视频突发模式

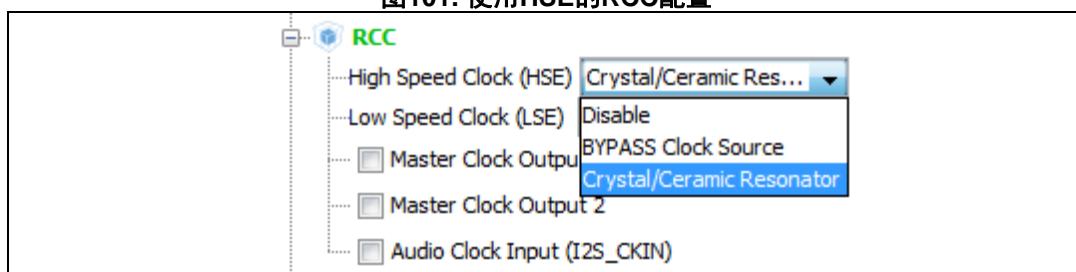
本节介绍了在视频突发模式下配置DSI主机所需的不同步骤。仅介绍了强制设置。

#### 6.1.1 引脚排列配置

##### 在RCC中使能HSE

DSI要求使用HSE（高速外部）振荡器作为DSIPLL的时钟输入。[图 101](#)显示了如何配置RCC（复位和时钟控制）以使能HSE。

图101. 使用HSE的RCC配置

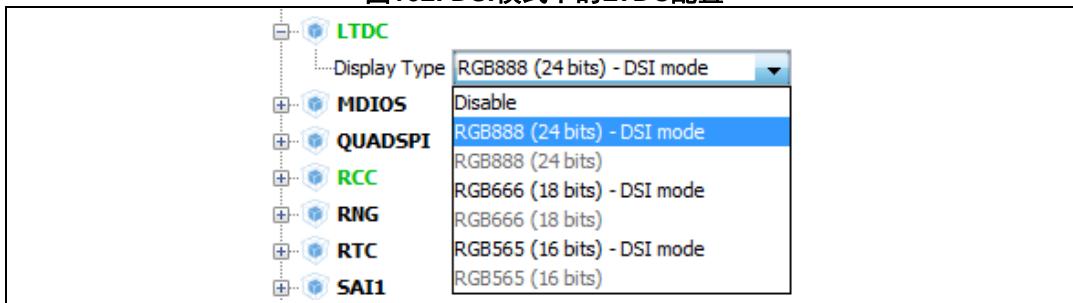


##### 以DSI模式使能LTDC

必须使能LTDC并设置为DSI模式。

请注意，LTDC用作视频推流器，用于对DSI的馈送。因此在DSI模式下，LTDC输出通过芯片内部连接到DSI主机，这意味着LTDC不需要引脚配置。[图 102](#)所示为LTDC配置。

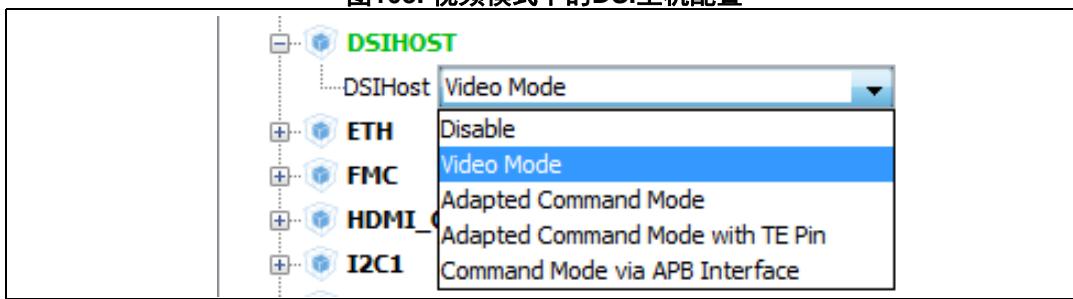
图102. DSI模式下的LTDC配置



### 以视频模式使能DSI主机

为DSI主机选择视频模式（参见 [图 103](#)）。

图103. 视频模式下的DSI主机配置



注：  
DSI主机使用专用引脚，无需复用功能配置。

## 6.1.2 时钟配置

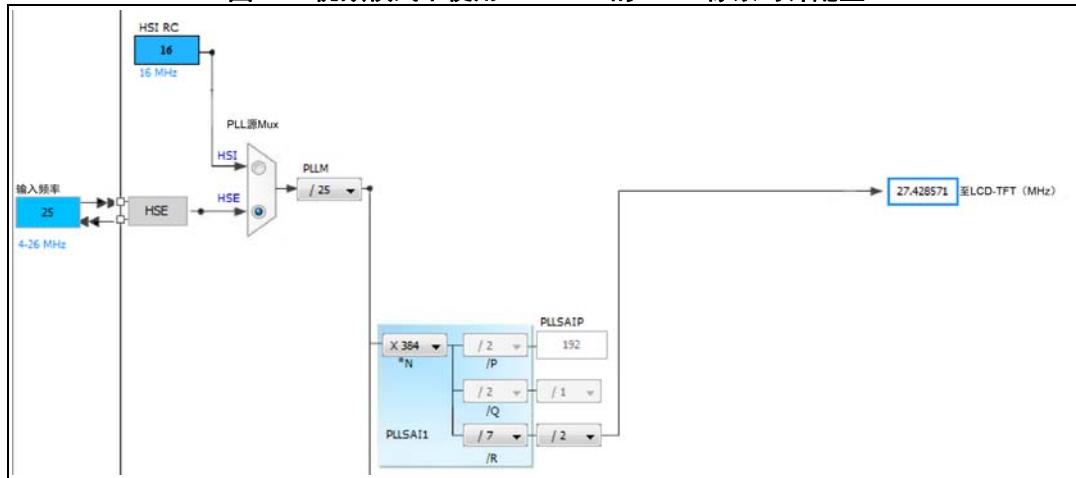
在时钟配置阶段，必须配置DSI PLL（对于DSI主机和PHY）和PLLSAI（对于LTDC）。

### LTDC时钟配置

根据显示要求设置LTDC像素时钟频率。请参见 [LTDC设置第 89页](#) 中的 LTDC像素时钟设置了解视频模式下的像素时钟设置。

在本例中，像素时钟频率设置为27.4 MHz。[图 104](#)显示了将像素时钟设置为27.4 MHz所需的PLLSAI1设置。

图104. 视频模式下使用PLLSAI1的LTDC像素时钟配置

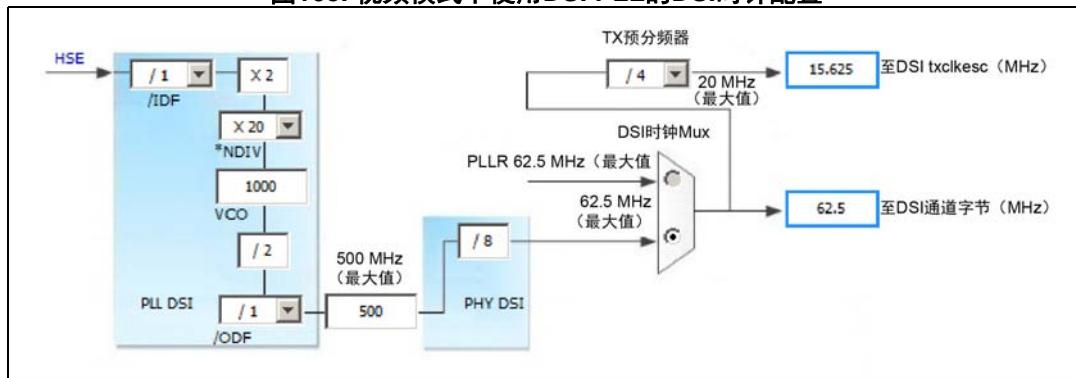


### DSI时钟配置

本例显示了用于生成不同DSI时钟的DSI PLL配置。

[图 105](#) 显示了DSI PLL的配置示例。本例中的链路速率为每通道500 Mbit/s, lane\_byte\_clock设置为62.5 MHz。

图105. 视频模式下使用DSI PLL的DSI时钟配置



必须选择逸出模式时钟预分频器值以生成TX LP传输期间使用的逸出时钟。此时钟不得超过20 MHz。

注：必须将TX预分频器设置为大于2的值。预分频器值0或1将禁用TX逸出时钟的生成。

## 6.1.3 LTDC和DSI配置

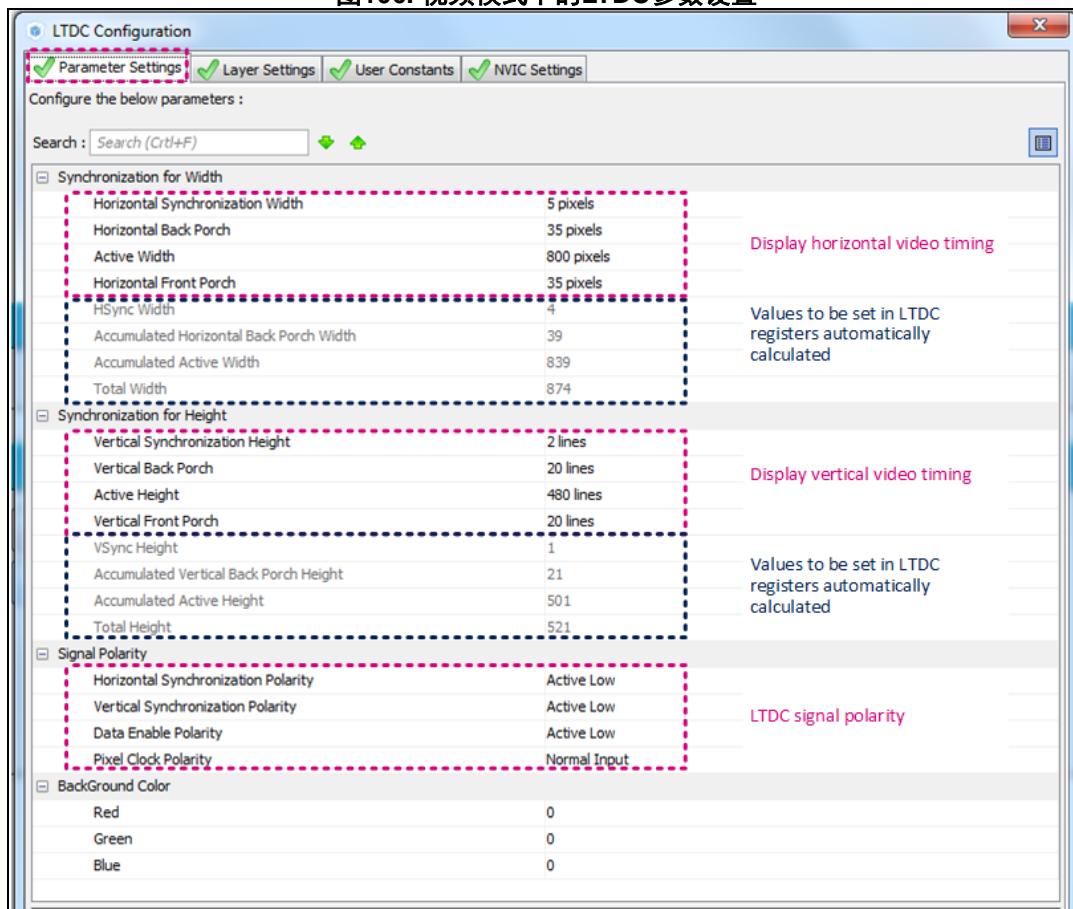
### LTDC配置

本节介绍使用DSI接口显示图像所需的LTDC设置。

- LTDC参数设置

[图 105](#)显示了LTDC参数设置。

图106. 视频模式下的LTDC参数设置



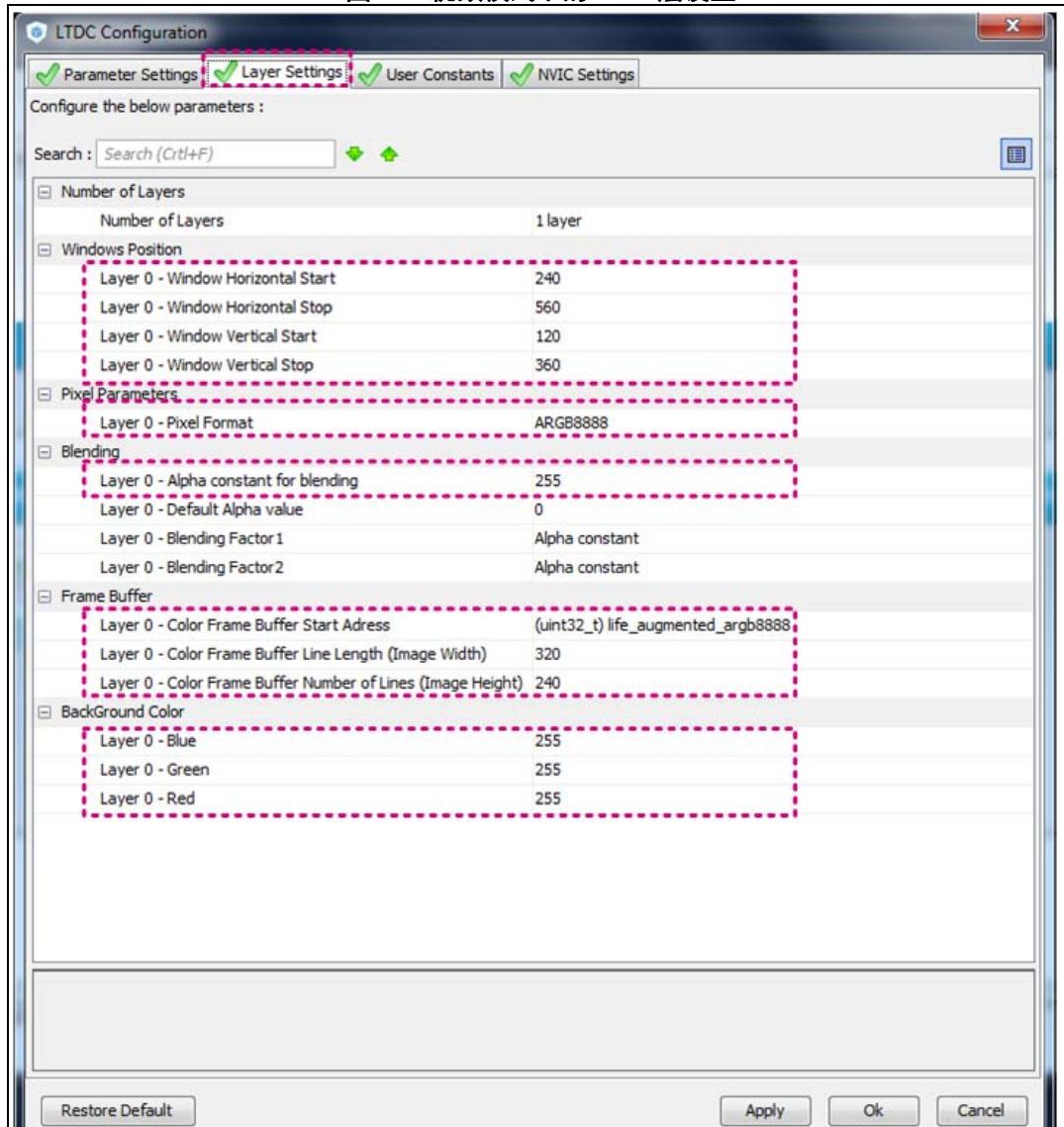
垂直和水平视频时序：用户输入显示时序，STM32CubeMX自动生成要在LTDC寄存器中编程的相应值。

视频信号极性：这些信号被馈送到DSI主机。唯一的限制在于，必须在LTDC和DSI中设定相同极性。用户必须输入LTDC的极性，然后STM32CubeMX才能确保在DSI中自动设置相应极性。

- LTDC层设置

[图 107](#)显示了LTDC层设置。本例中只使用了一层。

图107. 视频模式下的LTDC层设置



**窗口位置定义：**本例使用宽320个像素、高240行的图像，屏幕的剩余部分填充以层默认颜色。选择窗口位置，使图像显示在屏幕中心。

**像素参数设置：**必须根据源图像颜色格式设置像素参数。本例中使用的图像的颜色格式为ARGB8888。

**混合参数设置：**本例中只使用一层，用于混合的常量 $\alpha$ 参数为255。

### 帧缓冲器参数：

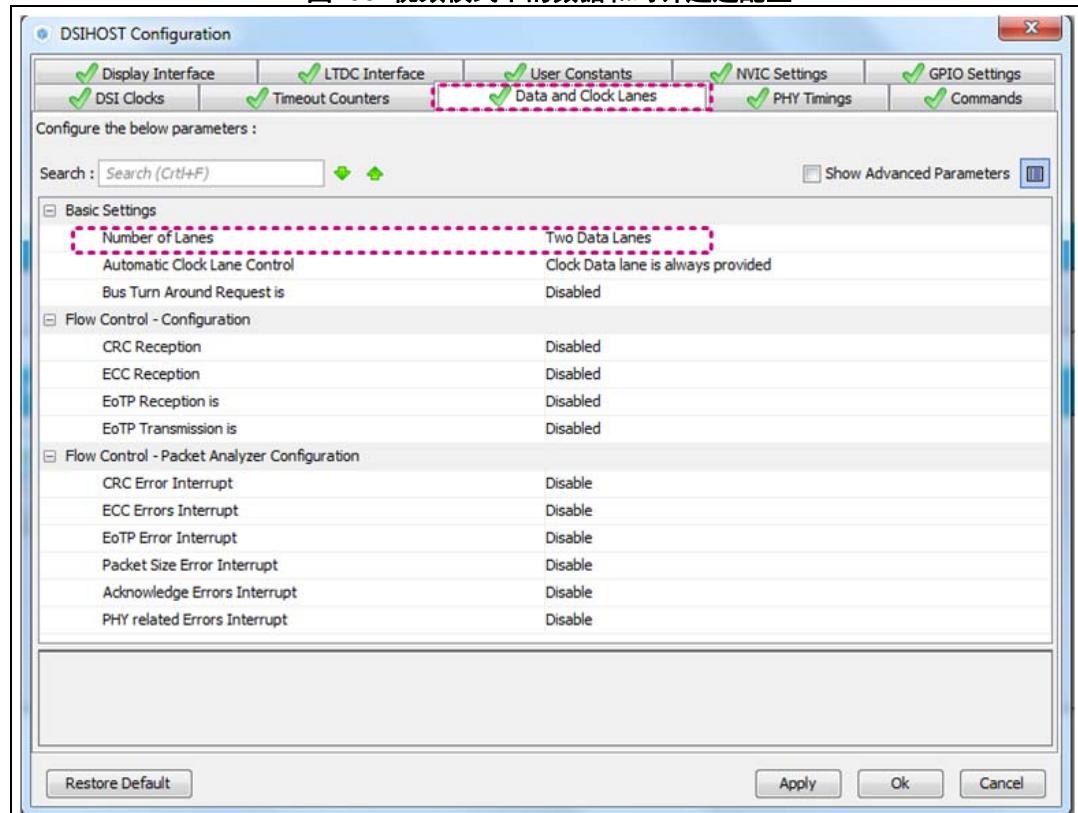
- 将帧缓冲器的开始地址设置为图像源开始地址。在本例中，将其设置为源图像上的一个指针（life\_augmented\_argb8888）。
- 分别以图像宽度和高度设置帧缓冲器长度和行数。本例中使用320 x 240图像。
- 层默认颜色设置：所显示图像以外的所有区域均采用默认颜色，在本例中为白色。

### DSI主机配置

- 数据和时钟通道配置

[图 108](#)显示了DSI数据和时钟通道设置。

**图108. 视频模式下的数据和时钟通道配置**



通道数选择：本例中使用了2个通道。

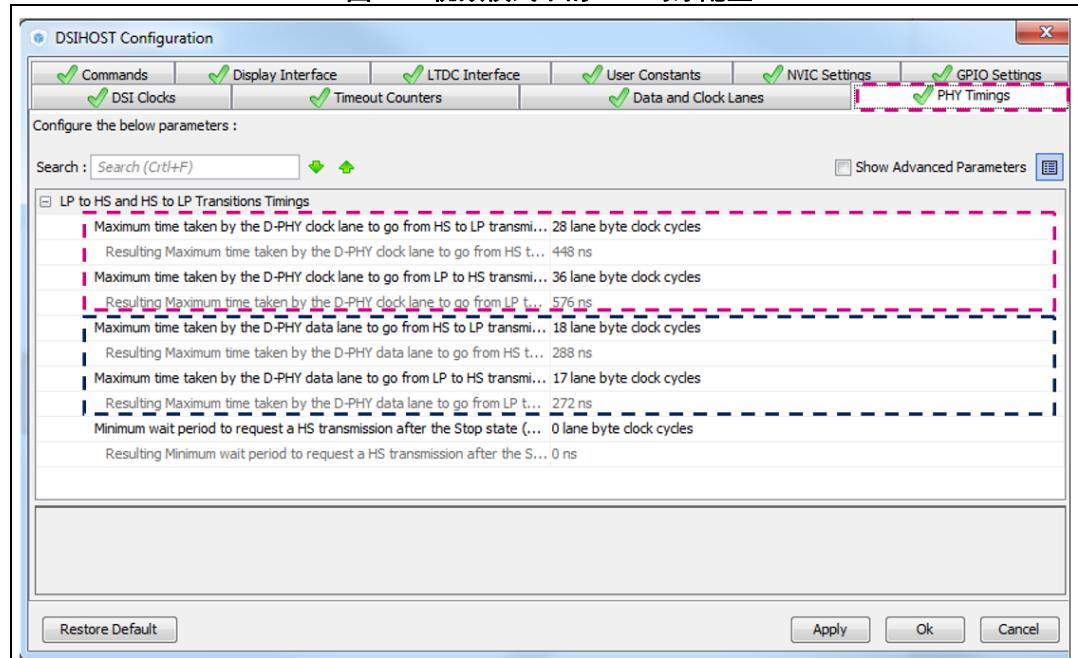
时钟通道控制模式：本例中始终提供时钟通道。

BTA请求：如需反向通信（读请求、帧回应请求或其他），则必须使能BTA请求。在本例中不需要。

- **PHY时序配置**

[图 109](#)显示了DSI PHY时序配置。

**图109. 视频模式下的PHY时序配置**

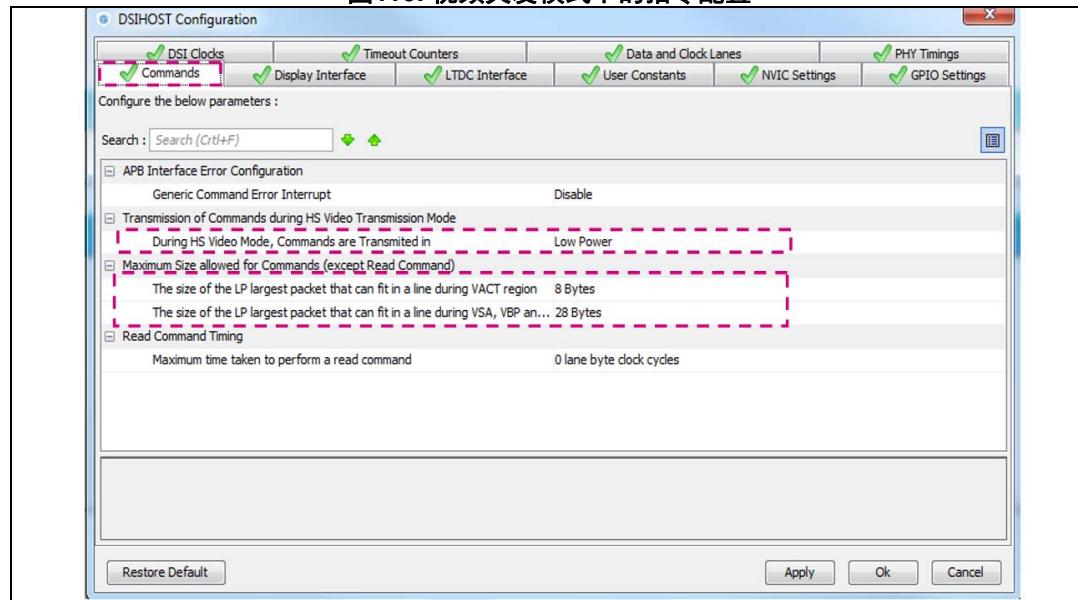


- **数据通道转换时序：**  
保留默认的LP至HS和HS至LP转换时序，如上图所示。  
必须完成数据通道转换时序配置。
- **时钟通道转换时序：**仅当“自动时钟通道控制”设置为“时间允许时不提供时钟通道”时，才需要这些时序。
- 将视频模式下停止状态（SW\_TIME）后请求高速传输的最小等待周期设置为0。

- 指令传输配置。

[图 110](#)显示了指令传输配置。

图110. 视频突发模式下的指令配置

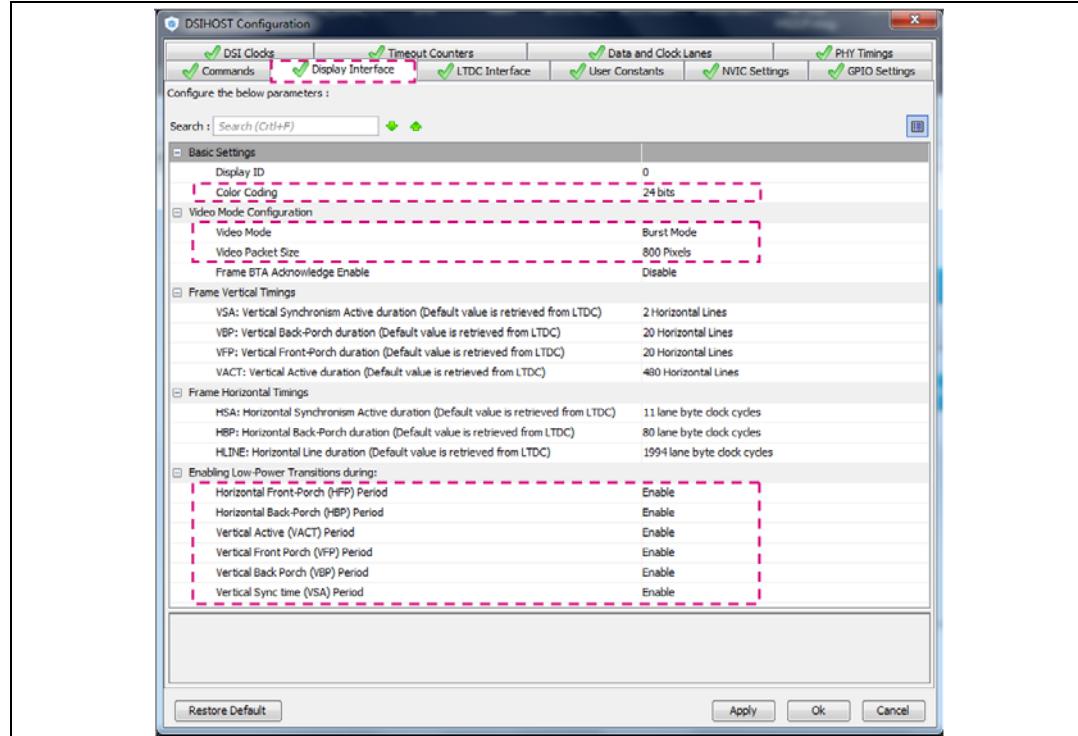


- 指令传输模式：在低功耗模式下使能指令传输。在本例中，由于必须在LP模式下发送显示屏初始化指令，因此必须配置。
- 最大LP指令大小：必须按[DSI视频数据包参数](#)章节所示进行计算。  
设置VACT区域内的LP最大数据包大小 = 8字节  
设置垂直消隐区域内的LP最大数据包大小 = 28字节 更多详细信息，请参见[视频模式下的指令传输](#)。
- 在本例中无需读指令，因此保留读指令时序0。

- 显示接口配置

[图 111](#)显示了DSI显示接口配置：

图111. 视频突发模式下的显示接口配置



- 颜色编码选择：允许用户选择通过链路传输的DSI数据包的颜色格式。此设置独立于图像源的LTDC层颜色格式。图像源可使用565颜色格式，DSI主机设置为24位颜色格式。  
在本例中，使用封装像素流以24位格式发送像素流。
- 视频模式配置：  
选择视频突发模式。  
设置视频数据包大小：在突发模式下，将视频数据包大小设置为完整视频行长（以像素为单位）。  
如果需要，使能帧BTA回应。本例中不使用此设置。
- 从LTDC配置检索帧的垂直和水平时序。STM32CubeMX根据LTDC视频设置、通道字节时钟和LTDC像素时钟自动计算值。
- 建议对所有区域使能LP转换。DSI主机自动根据区域长度和PHY转换检查是否能够执行转换。如果需要，用户可以禁用特定区域内的LP转换，但不建议这样做。

## 6.1.4 视频突发模式的生成代码示例

在完成配置阶段后，STM32CubeMX为用户提供使用不同工具链生成代码的能力。本节显示了STM32CubeMX生成的部分代码，并突出显示了需要用户添加到“用户代码”部分的用于显示屏复位和初始化的代码部分。

```
/* 主要功能 */
/* 使能I-Cache */
SCB_EnableICache();
/* 使能D-Cache */
SCB_EnableDCache();
/* MCU配置 */
/* 复位所有外设，初始化Flash接口和Systick。 */
HAL_Init();
/* 配置系统时钟 */
SystemClock_Config();
/* 初始化所有配置的外设 */
/*HSE的GPIO初始化*/
MX_GPIO_Init();
/*DSI主机初始化，DSI在该阶段尚未启动*/
MX_DSIHOST_DSI_Init();
/*LTDC初始化。在该阶段使能LTDC。*/
MX_LTDC_Init();
/* 用户代码开始 2 这是唯一需要用户修改的部分*/
/* 通过XRES信号进行LCD显示屏硬件复位*/
BSP_LCD_Reset();
/*启动DSI主机。这将使能DSI主机和Wrapper。必须在LTDC使能后执行*/
HAL_DSI_Start(&hdsi);
/*LCD显示屏初始化：这将发送显示屏初始化指令。参数为颜色格式和方向。LCD显示屏颜色格式必须
与DSI主机颜色格式一致*/
OTM8009A_Init(OTM8009A_FORMAT_RGB888 , OTM8009A_ORIENTATION_LANDSCAPE);
/* 用户代码结束 2*/

/* 全局初始化 */
static void MX_DSIHOST_DSI_Init(void){
    hdsi.Instance = DSI;
    hdsi.Init.AutomaticClockLaneControl = DSI_AUTO_CLK_LANE_CTRL_DISABLE;
    hdsi.Init.TXEscapeCkdiv = 4;
    hdsi.Init.NumberOfLanes = DSI_TWO_DATA_LANES;
    /*每通道500Mbps速率的DSI PLL配置。Lane_byte_clock为62,5 Mhz*/
    PLLInit.PLLNDIV = 20;
    PLLInit.PLLIDF = DSI_PLL_IN_DIV1;
    PLLInit.PLLODF = DSI_PLL_OUT_DIV1;
```

```
if (HAL_DSI_Init(&hdsi, &PLLInit) != HAL_OK)
{
    Error_Handler();
}

/*视频模式初始化*/
VidCfg.VirtualChannelID = 0;
/*DSI主机颜色格式配置。使用封装像素流以24位格式发送图像数据。*/
VidCfg.ColorCoding = DSI_RGB888;
VidCfg.LoselyPacked = DSI_LOOSELY_PACKED_DISABLE;
/*视频模式配置*/
VidCfg.Mode = DSI_VID_MODE_BURST;
/*视频数据包配置*/
VidCfg.PacketSize = 800;
VidCfg.NumberOfChunks = 0;
VidCfg.NullPacketSize = 0;
/*信号极性配置。LTDC和DSI极性相同，具有相反极性的DATA ENABLE除外 */
VidCfg.HSPolarity = DSI_HSYNC_ACTIVE_LOW;
VidCfg.VSPolarity = DSI_VSYNC_ACTIVE_LOW;
VidCfg.DEPolarity = DSI_DATA_ENABLE_ACTIVE_HIGH;
/*视频时序配置。从LTDC配置进行检索*/
VidCfg.HorizontalSyncActive = 11;
VidCfg.HorizontalBackPorch = 80;
VidCfg.HorizontalLine = 1994;
VidCfg.VerticalSyncActive = 2;
VidCfg.VerticalBackPorch = 20;
VidCfg.VerticalFrontPorch = 20;
VidCfg.VerticalActive = 480;
/*指令传输模式和最大LP数据包大小*/
VidCfg.LPCommandEnable = DSI_LP_COMMAND_ENABLE;
VidCfg.LPLargestPacketSize = 29;
VidCfg.LPVACTLargestPacketSize = 8;
/*LP转换配置。建议在所有区域使能LP转换*/
VidCfg.LPHorizontalFrontPorchEnable = DSI_LP_HFP_ENABLE;
VidCfg.LPHorizontalBackPorchEnable = DSI_LP_HBP_ENABLE;
VidCfg.LPVerticalActiveEnable = DSI_LP_VACT_ENABLE;
VidCfg.LPVerticalFrontPorchEnable = DSI_LP_VFP_ENABLE;
VidCfg.LPVerticalBackPorchEnable = DSI_LP_VBP_ENABLE;
VidCfg.LPVerticalSyncActiveEnable = DSI_LP_VSYNC_ENABLE;
/*流控配置*/
VidCfg.FrameBTAAcknowledgeEnable = DSI_FBTAA_DISABLE;
if (HAL_DSI_ConfigVideoMode(&hdsi, &VidCfg) != HAL_OK)
{
```

```
        Error_Handler();
    }

/*LTDC初始化*/
LTDC_LayerCfgTypeDef pLayerCfg;
hLtdc.Instance = LTDC;
hLtdc.Init.HSPolarity = LTDC_HSPOLARITY_AL;
hLtdc.Init.VSPolarity = LTDC_VSPOLARITY_AL;
hLtdc.Init.DEPolarity = LTDC_DEPOLARITY_AL;
hLtdc.Init.PCPolarity = LTDC_PCPOLARITY_IPC;
/*符合显示时序的视频时序配置*/
hLtdc.Init.HorizontalSync = 4;
hLtdc.Init.VerticalSync = 1;
hLtdc.Init.AccumulatedHBP = 39;
hLtdc.Init.AccumulatedVBP = 21;
hLtdc.Init.AccumulatedActiveW = 839;
hLtdc.Init.AccumulatedActiveH = 501;
hLtdc.Init.TotalWidth = 874;
hLtdc.Init.TotalHeigh = 521;
if (HAL_LTDC_Init(&hLtdc) != HAL_OK)
{
    Error_Handler();
}

/*LTDC层初始化*/
/*窗口位置配置*/
pLayerCfg.WindowX0 = 240;
pLayerCfg.WindowX1 = 560;
pLayerCfg.WindowY0 = 120;
pLayerCfg.WindowY1 = 360;
/*输入像素颜色格式。它必须与源图像颜色格式一致*/
pLayerCfg.PixelFormat = LTDC_PIXEL_FORMAT_ARGB8888;
/*混合参数*/
pLayerCfg.Alpha = 255;
pLayerCfg.Alpha0 = 0;
pLayerCfg.BlendingFactor1 = LTDC_BLENDING_FACTOR1_CA;
pLayerCfg.BlendingFactor2 = LTDC_BLENDING_FACTOR2_CA;
/*帧缓冲器参数*/
pLayerCfg.FBStartAdress = (uint32_t) life_augmented_argb8888;
pLayerCfg.ImageWidth = 320;
pLayerCfg.ImageHeight = 240;
/*背景颜色配置。本例中为白色背景*/
pLayerCfg.Backcolor.Blue = 255;
```

```

pLayerCfg.Backcolor.Green = 255;
pLayerCfg.Backcolor.Red = 255;
if (HAL_LTDC_ConfigLayer(&hltdc, &pLayerCfg, 0) != HAL_OK)
{
    Error_Handler();
}

```

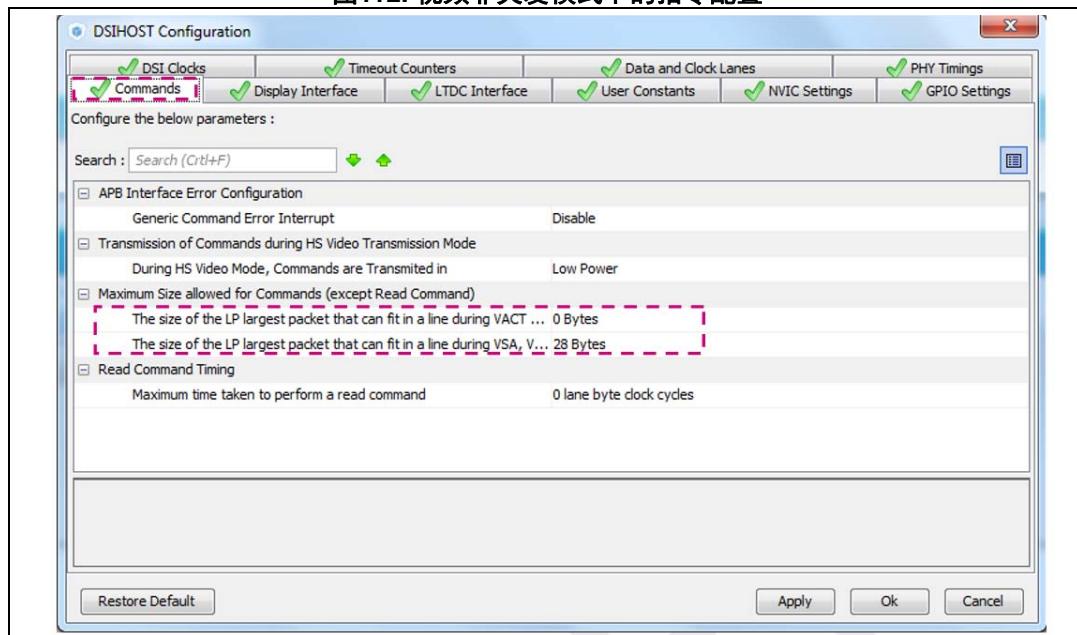
## 6.2 使用同步脉冲的DSI主机非突发模式

大部分配置与突发模式配置类似，指令配置和显示接口配置除外。

### 6.2.1 指令配置

[图 112](#)显示了指令传输配置。

图112. 视频非突发模式下的指令配置



相比于视频突发模式，在非突发模式下，VACT区域内用于指令传输的余量较少。

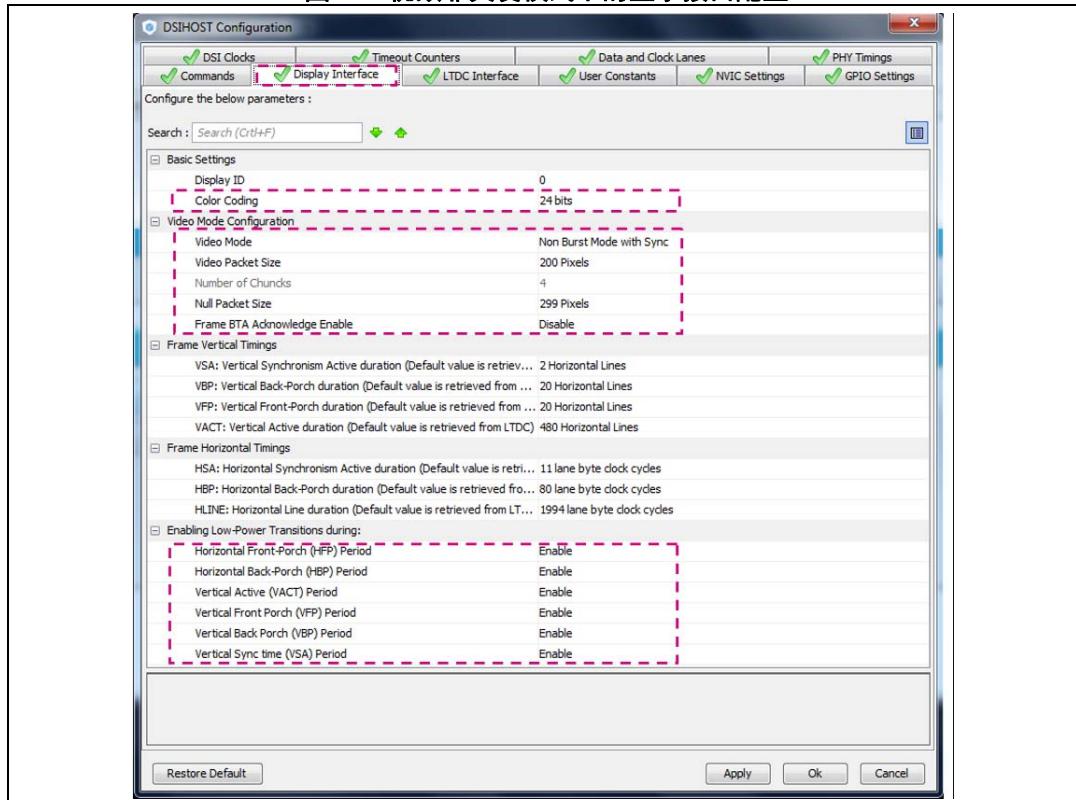
在本例中，该周期内不允许传输指令，因此必须将VACT期间的LP最大数据包设置为0。

更多详细信息，请参见[视频模式下的指令传输](#)。

### 6.2.2 显示接口配置

[图 113](#)显示了DSI显示接口配置。

图113. 视频非突发模式下的显示接口配置



视频模式选择：在本例中，选择使用同步脉冲的非突发模式。

视频数据包大小：必须根据显示屏中的可用行缓冲区大小进行设置。在本例中，视频数据包大小设置为200像素。

自动根据以下参数计算块数：在LTDC中配置的激活宽度和数据包大小（块数 = 激活宽度 / 视频数据包大小）。

空包大小：必须计算以字节为单位的空包大小。本例中需要299字节。关于计算的更多详细信息，请参见 [DSI视频数据包参数](#)。

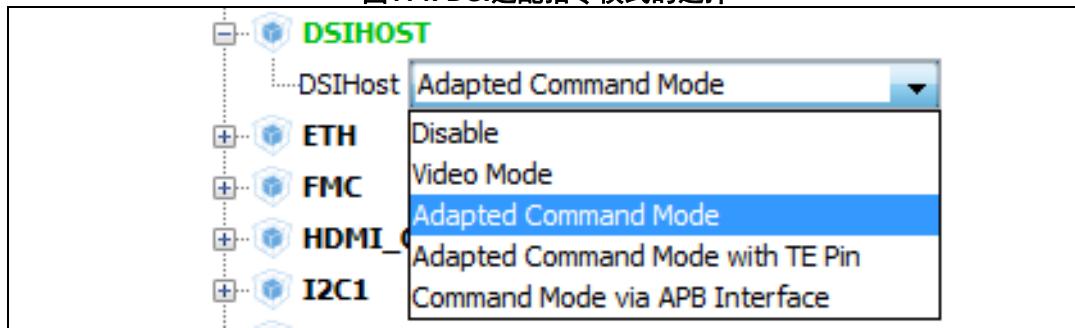
## 6.3 DSI主机适配指令模式

### 6.3.1 引脚配置

RCC和LTDC引脚排列配置与视频突发模式相同。只需更改DSI主机配置并设置为适配指令模式。

在引脚排列配置部分选择DSI主机适配指令模式（参见图 114）。

图114. DSI适配指令模式的选择



注：如需通过引脚报告撕裂效应，则必须选择“使用TE引脚的适配指令模式”。此选项配置TE使用引脚。

### 6.3.2 时钟配置

DSI时钟配置与视频模式相同。本例中保留相同的lane\_byte\_clock频率，即62.5 MHz。

#### LTDC时钟配置

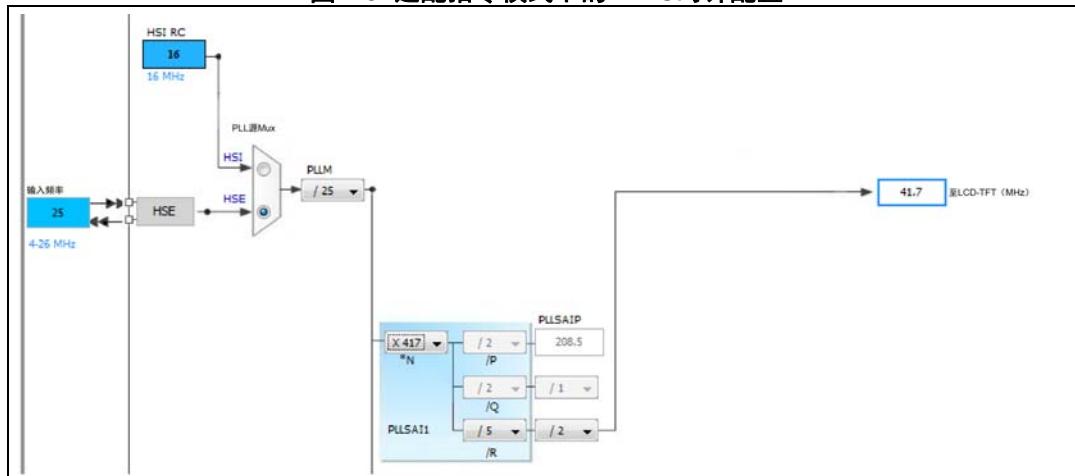
在适配指令模式下，为了缩短刷新时间，将LTDC pixel\_clock设置为支持的最高频率。

像素时钟 = 通道速率 × 通道数 / bit\_per\_pixel

在24位颜色模式下，使用两个数据通道，每个通道的速率为500 Mbit/s，则LTDC像素时钟为 $500 \text{ mbs} \times 2 / 24 = 41.7 \text{ MHz}$ 。

[图 115](#)显示了如何配置PLLSAI以将LTDC像素时钟设置为41.7 MHz。

图115. 适配指令模式下的LTDC时钟配置



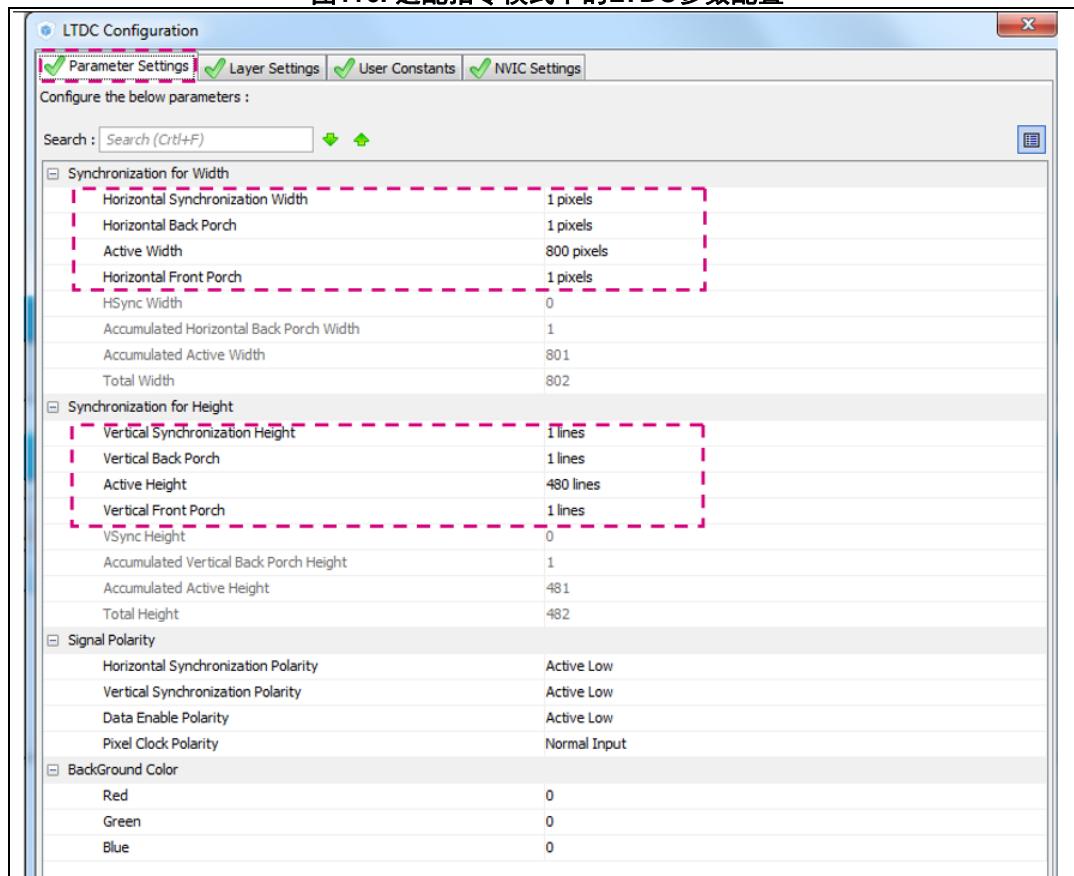
### 6.3.3 LTDC和DSI配置

#### LTDC配置

关于LTDC层设置，请参见视频模式示例（参见[第 6.1 节：DSI主机视频突发模式](#)）。

LTDC参数的配置与视频模式相同，视频消隐时序除外（参见图 116）。

图116. 适配指令模式下的LTDC参数配置



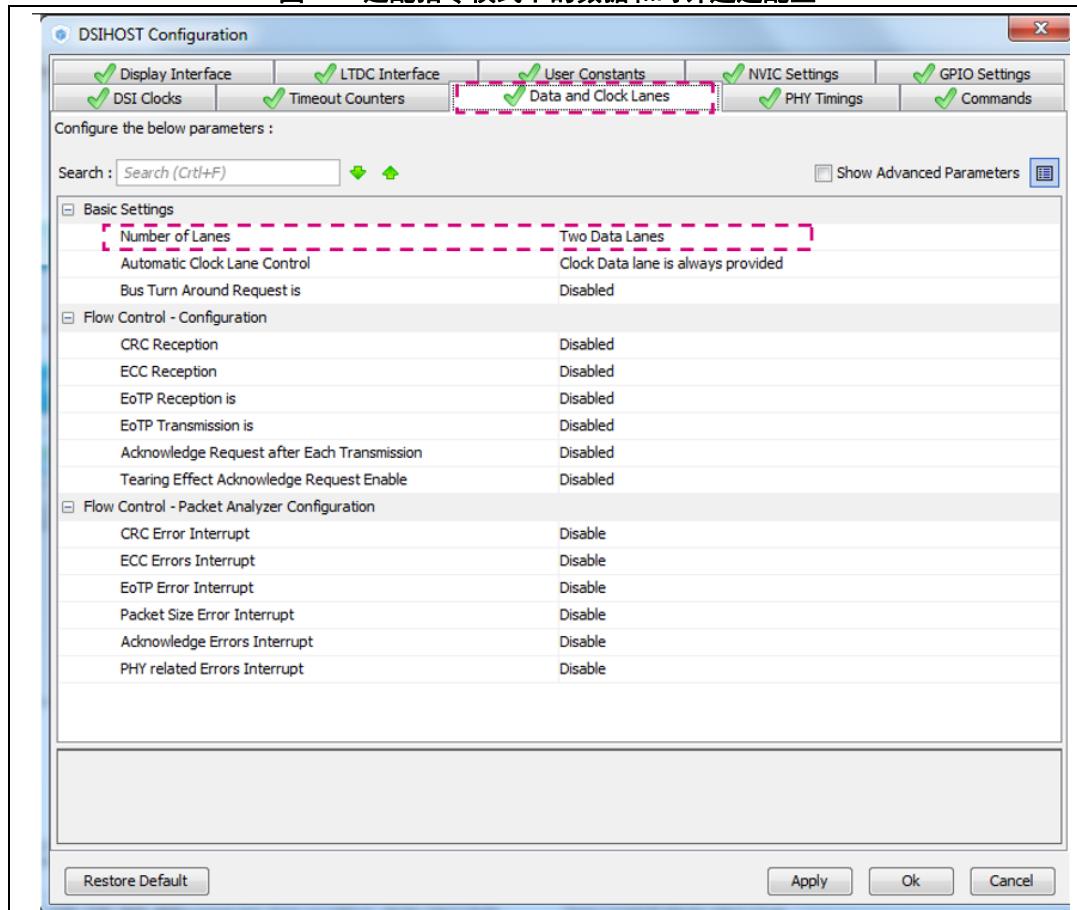
在适配指令模式下，可将水平消隐时序设置为最小值，即1个像素时钟，并将垂直消隐时序设置为最小值，即1行。这是因为显示屏依赖其内部显示控制器生成视频时序。

## DSI主机配置

- 数据和时钟通道配置

[图 117](#)显示了适配指令模式下的DSI数据和时钟通道设置。

图117. 适配指令模式下的数据和时钟通道配置



数据通道数选择：本例中使用了2个数据通道。

如需通过链路报告撕裂效应，则用户必须使能：

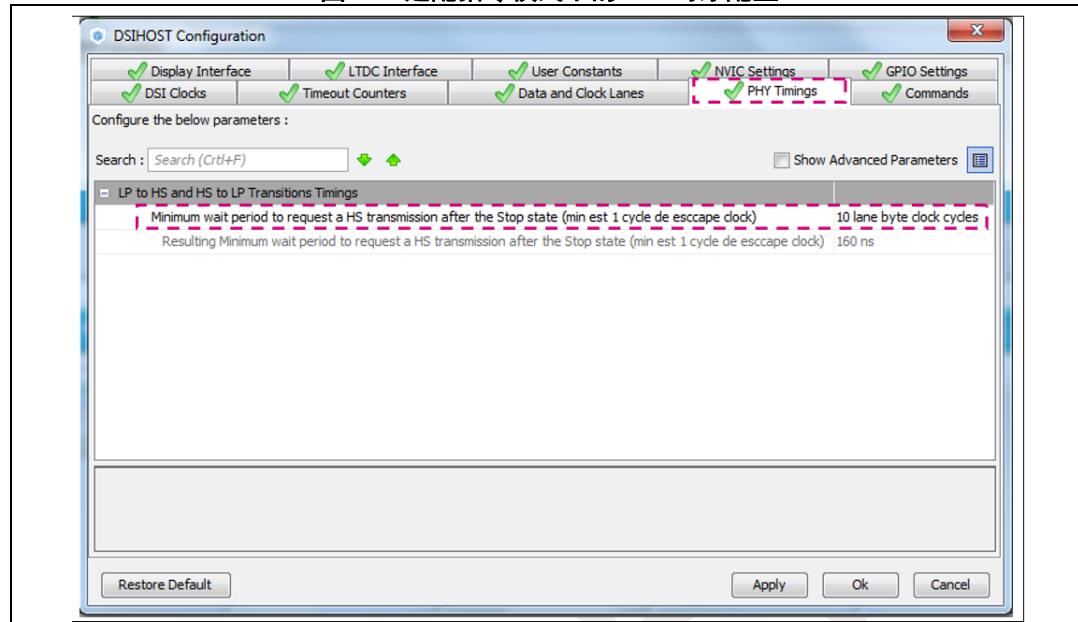
- 总线转向请求
- 撕裂效应回应请求

本例中不使用撕裂效应回报。

- **PHY时序配置**

图 118展示了DSI PHY时序配置。要配置的唯一一个参数是停止等待时序。

图118. 适配指令模式下的PHY时序配置

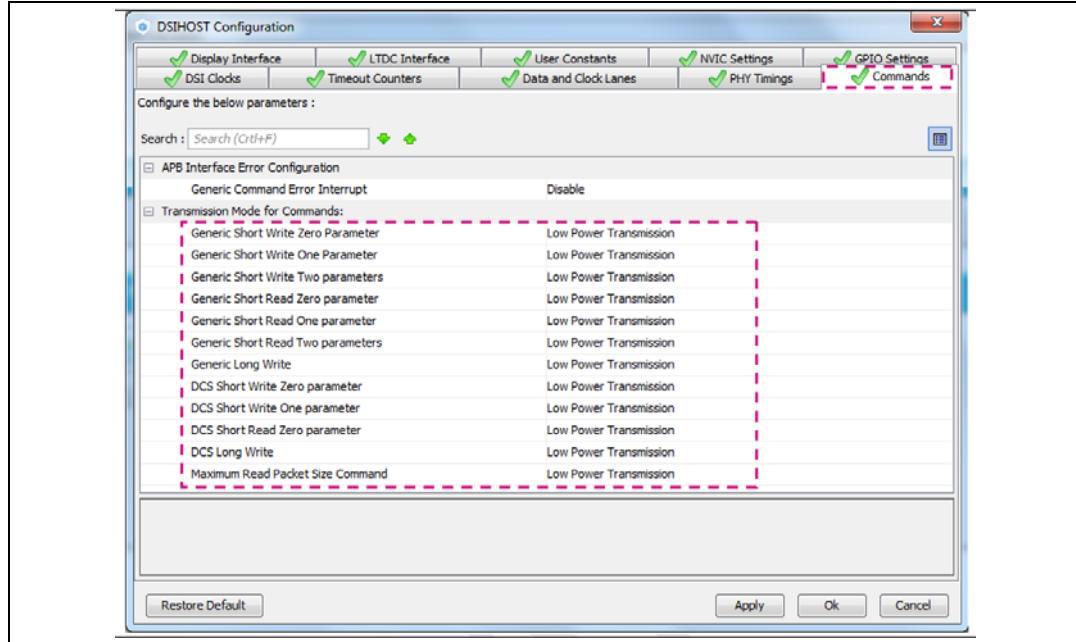


在指令模式下，必须设置停止状态后请求高速传输的最小等待周期，即停止等待时序（SW\_Time）。DSI主机需要的SW\_Time为10个通道字节时钟周期。如果显示屏需要更长的停止等待时序，则用户必须使用显示屏SW\_TIME编程此字段。

- 指令传输配置

图 119 显示了指令传输配置。

图119. 适配指令模式下的指令传输配置



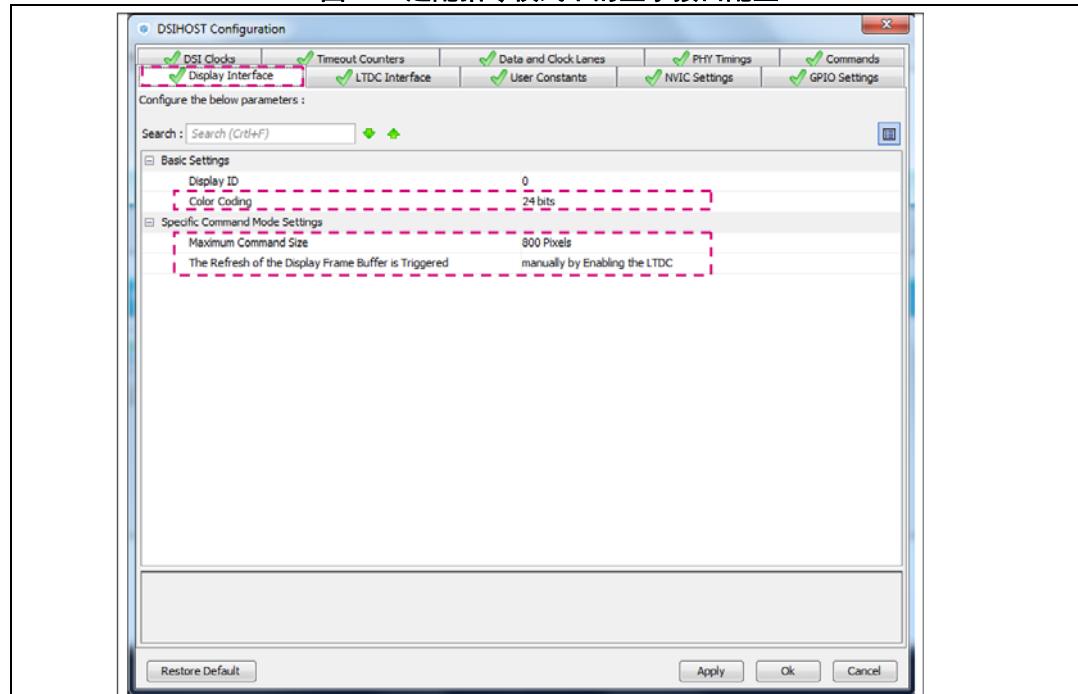
可单独将每种指令类型设置为以LP或HS模式发送。

一些显示屏需要在LP模式下发送初始化指令。这种情况下，用户必须使能以低功耗模式传输指令。然后，在显示屏的初始化完成后，用户必须重新使能高速传输（特别是用于帧刷新的DCS长写指令）。

- 显示接口配置

图 120 显示了DSI显示接口配置。

图120. 适配指令模式下的显示接口配置



颜色编码选择：本例中使用了24位颜色模式。

在指令模式下，主机必须将使用的颜色格式通知显示屏。这可以通过发送DCS短指令set\_pixel\_format来完成。可在显示屏初始化阶段执行。

在本例中，最大指令大小设置为800像素，将帧的每一行封装在1个WMS或WMC DCS指令中。

刷新模式选择：在本例中，通过使能LTDC手动完成刷新。

### 6.3.4 适配指令模式的生成代码示例

本节显示了STM32CubeMX为适配指令模式配置生成的代码。用户必须在“用户代码”部分添加与显示屏复位和初始化相关的几部分。

```
/* 主要功能 */
/* 使能I-Cache */
SCB_EnableICache();

/* 使能D-Cache */
SCB_EnableDCache();

/* MCU配置 */
```

```
/* 复位所有外设， 初始化Flash接口和Systick。 */
HAL_Init();

/* 配置系统时钟 */
SystemClock_Config();

/* 初始化所有配置的外设 */
/*HSE的GPIO初始化*/
MX_GPIO_Init();
/*DSI主机初始化， DSI在该阶段尚未启动*/
MX_DSIHOST_DSI_Init();
/*LTDC初始化。在该阶段使能LTDC。*/
MX_LTDC_Init();
/* 用户代码开始 2。这是唯一需要用户修改的部分 */
/* 通过XRES信号进行LCD显示屏硬件复位*/
BSP_LCD_Reset();
/*启动DSI主机。这将使能DSI主机和Wrapper。必须在LTDC使能后执行*/
HAL_DSI_Start(&hdssi);
/*LCD显示屏初始化：这将发送显示屏初始化指令。参数为颜色格式和方向。颜色格式必须与DSI主机
颜色格式一致*/
OTM8009A_Init(OTM8009A_FORMAT_RGB888, OTM8009A_ORIENTATION_LANDSCAPE);

/*使能指令的高速传输。这发生在根据一些显示屏的要求在LP模式下完成显示屏初始化时*/
/*由于使用DCS长写指令完成显示屏刷新，停止DSI并仅在高速模式下重新配置DCS长写指令*/
HAL_DSI_Stop(&hdssi_eval);
/* 配置指令模式 */
dsiLPCmdInit.LPGenShortWriteNoP      = DSI_LP_GSW0P_ENABLE;
dsiLPCmdInit.LPGenShortWriteOneP     = DSI_LP_GSW1P_ENABLE;
dsiLPCmdInit.LPGenShortWriteTwoP     = DSI_LP_GSW2P_ENABLE;
dsiLPCmdInit.LPGenShortReadNoP       = DSI_LP_GSR0P_ENABLE;
dsiLPCmdInit.LPGenShortReadOneP      = DSI_LP_GSR1P_ENABLE;
dsiLPCmdInit.LPGenShortReadTwoP      = DSI_LP_GSR2P_ENABLE;
dsiLPCmdInit.LPGenLongWrite         = DSI_LP_GLW_ENABLE;
dsiLPCmdInit.LPDcsShortWriteNoP     = DSI_LP_DSW0P_ENABLE;
dsiLPCmdInit.LPDcsShortWriteOneP    = DSI_LP_DSW1P_ENABLE;
dsiLPCmdInit.LPDcsShortReadNoP      = DSI_LP_DSR0P_ENABLE;
dsiLPCmdInit.LPDcsLongWrite        = DSI_LP_DLW_DISABLE;
dsiLPCmdInit.LPMaxReadPacket       = DSI_LP_MRDP_ENABLE;
dsiLPCmdInit.AcknowledgeRequest    = DSI_ACKNOWLEDGE_DISABLE;
/* 初始化命令模式 */
HAL_DSI_ConfigCommand(&hdssi_eval, &dsiLPCmdInit);
```

```
HAL_DSI_Start(&hdsi_eval);

/* 手动刷新。这将在DSI Wrapper控制寄存器中使能LTDCEN位 */
HAL_DSI_Refresh(&hdsi);
/* 用户代码结束 2 */

/* 全局初始化 */
hdsi.Instance = DSI;
hdsi.Init.AutomaticClockLaneControl = DSI_AUTO_CLK_LANE_CTRL_DISABLE;
hdsi.Init.TXEscapeCkdiv = 4;
hdsi.Init.NumberOfLanes = DSI_TWO_DATA_LANES;
/*DSI PLL配置。这会将链路速率设置为500 mbps。Lane_byte_clock为62,5 Mhz*/
PLLInit.PLLNDIV = 20;
PLLInit.PLLIDF = DSI_PLL_IN_DIV1;
PLLInit.PLLODF = DSI_PLL_OUT_DIV1;
if (HAL_DSI_Init(&hdsi, &PLLInit) != HAL_OK)
{
    Error_Handler();
}

/*指令传输配置*/
/*使能LP模式的指令传输。对于某些显示屏，必须在初始化阶段完成。在显示屏初始化完成后，必须
为DCS长写指令禁用LP传输*/
LPCmd.LPGenShortWriteNoP = DSI_LP_GSW0P_ENABLE;
LPCmd.LPGenShortWriteOneP = DSI_LP_GSW1P_ENABLE;
LPCmd.LPGenShortWriteTwoP = DSI_LP_GSW2P_ENABLE;
LPCmd.LPGenShortReadNoP = DSI_LP_GSR0P_ENABLE;
LPCmd.LPGenShortReadOneP = DSI_LP_GSR1P_ENABLE;
LPCmd.LPGenShortReadTwoP = DSI_LP_GSR2P_ENABLE;
LPCmd.LPGenLongWrite = DSI_LP_GLW_ENABLE;
LPCmd.LPDcsShortWriteNoP = DSI_LP_DSW0P_ENABLE;
LPCmd.LPDcsShortWriteOneP = DSI_LP_DSW1P_ENABLE;
LPCmd.LPDcsShortReadNoP = DSI_LP_DSR0P_ENABLE;
LPCmd.LPDcsLongWrite = DSI_LP_DLW_ENABLE;
LPCmd.LPMaxReadPacket = DSI_LP_MRDP_ENABLE;
LPCmd.AcknowledgeRequest = DSI_ACKNOWLEDGE_DISABLE;
if (HAL_DSI_ConfigCommand(&hdsi, &LPCmd) != HAL_OK)
{
    Error_Handler();
}

/*适配指令配置*/
```

```
CmdCfg.VirtualChannelID = 0;  
/*选择DSI主机颜色格式*/  
CmdCfg.ColorCoding = DSI_RGB888;  
CmdCfg.CommandSize = 800;  
CmdCfg.TearingEffectSource = DSI_TE_DSILINK;  
CmdCfg.TearingEffectPolarity = DSI_TE_RISING_EDGE;  
/* DS1主机信号极性。LTDC和DSI极性相同，具有相反极性的DATA ENABLE除外 */  
CmdCfg.HSPolarity = DSI_HSYNC_ACTIVE_LOW;  
CmdCfg.VSPolarity = DSI_VSYNC_ACTIVE_LOW;  
CmdCfg.DEPolarity = DSI_DATA_ENABLE_ACTIVE_HIGH;  
CmdCfg.VSyncPol = DSI_VSYNC_FALLING;  
/*使用手动刷新*/  
CmdCfg.AutomaticRefresh = DSI_AR_DISABLE;  
CmdCfg.TEAcknowledgeRequest = DSI_TE_ACKNOWLEDGE_DISABLE;  
if (HAL_DSI_ConfigAdaptedCommandMode(&hdsi, &CmdCfg) != HAL_OK)  
{  
    Error_Handler();  
}  
  
/*LTDC配置*/  
hltdc.Instance = LTDC;  
hltdc.Init.HSPolarity = LTDC_HSPOLARITY_AL;  
hltdc.Init.VSPolarity = LTDC_VSPOLARITY_AL;  
hltdc.Init.DEPolarity = LTDC_DEPOLARITY_AL;  
hltdc.Init.PCPolarity = LTDC_PCPOLARITY_IPC;  
/*视频时序配置。在适配指令模式下，可将垂直和水平消隐设置为1。*/  
hltdc.Init.HorizontalSync = 0;  
hltdc.Init.VerticalSync = 0;  
hltdc.Init.AccumulatedHBP = 1;  
hltdc.Init.AccumulatedVBP = 1;  
hltdc.Init.AccumulatedActiveW = 801;  
hltdc.Init.AccumulatedActiveH = 481;  
hltdc.Init.TotalWidth = 802;  
hltdc.Init.TotalHeigh = 482;  
/*背景颜色配置*/  
hltdc.Init.Backcolor.Blue = 0;  
hltdc.Init.Backcolor.Green = 0;  
hltdc.Init.Backcolor.Red = 0;  
if (HAL_LTDC_Init(&hltdc) != HAL_OK)  
{  
    Error_Handler();  
}  
/*LTDC层配置与视频突发模式示例相同。*/
```

## 7 DSI主机性能

DSI主机性能受DSI链路带宽的物理极限和系统限制的影响。

### 7.1 DSI链路最大带宽对LTDC像素时钟的影响

每个通道支持的最大DSI链路速度为500 Mbit/s。当使用两个数据通道时，总链路速度为1 Gbit/s。

这是等效像素时钟与DSI主机配置之间的关系。根据颜色编码、使用的数据通道数和数据通道速率，可按以下方式计算等效像素时钟：

$$\text{像素时钟} = (\text{lane_rate} \times \text{number_of_lanes}) / \text{bits_per_pixel}.$$

例如，当使用两个速率为500 Mbit/s的数据通道时，总数据速率为1 Gbit/s：

- 每个像素编码16位：

最大等效像素时钟为 $1 \text{ Gbit/s} / 16 \text{ bpp} = 62.5 \text{ MHz}$

- 每个像素编码24位：

最大等效像素时钟为 $1 \text{ Gbit/s} / 24 \text{ bpp} = 41.5 \text{ MHz}$

[表 37](#)显示了根据颜色编码和DSI链路速度得到的最大像素时钟频率。

**表37. 取决于颜色编码和DSI链路速度的最大像素时钟频率**

-	1个数据通道（500 Mbit/s）	2个数据通道（1 Gbit/s）
24 bpp	20.83 MHz	41.66 MHz
16 bpp	31.25 MHz	62.5 MHz

### 7.2 系统限制对LTDC像素时钟的影响

最大等效像素时钟可能受系统限制的影响。根据应用使用情况，系统限制可能要求减小LTDC像素时钟。

请参见应用笔记“STM32微控制器上的LCD-TFT显示控制器（LTDC）”了解关于系统限制对支持的最大像素时钟的影响的更多信息。

### 7.3 DSI链路带宽估计

本节旨在评估视频模式和适配指令模式需要的DSI链路带宽。

#### 7.3.1 视频模式

在视频模式下，LTDC像素时钟和最小链路BW（带宽）受显示时序的影响。

可使用以下公式计算像素时钟：

$$\text{像素时钟} = \text{总宽} \times \text{总高} \times \text{刷新频率}$$

使用以下公式计算驱动显示屏所需的最小DSI链路BW：

$$\text{最小DSI链路BW} = \text{像素时钟} \times \text{色深}$$

以具有以下时序的显示屏为例：

- HSA = 5, HBP = 35, HACT = 800, HFP = 35
- VSA = 2, VBP = 20, VACT = 480, VFP = 20
- 刷新速率 = 60 fps
- 色深 = 24 bpp
  - 像素时钟 =  $875 \times 522 \times 60 = 27.4 \text{ MHz}$
  - 最小链路BW = 657 Mbit/s

这是驱动显示屏所需的最小链路BW。

由于最大通道速率为500 Mbit/s，仅使用一个通道无法驱动此显示屏。

可通过两个最小通道速率为328 Mbit/s的通道来驱动此显示屏。

注：

上述链路BW只给出了要使用的最小必要BW和通道数的估算值。视频模式类型（突发对比非突发）和协议开销会影响链路BW的选择。

在突发模式下，DSI链路BW可能会增加。这确保了DSI主机以足够快的速度发送像素数据，从而长时序处于LP模式。

在非突发模式下，必须对DSI链路BW计算进行细微调整，以考虑[DSI视频数据包参数](#)中等式1和2所示的协议开销。

请注意，在两种情况下，即使DSI BW增加，LTDC像素时钟仍必须保留使用显示视频时序计算的值。

### 7.3.2 适配指令模式

在适配指令模式下，最大GRAM刷新时间受显示屏内部刷新频率的影响。

为避免视觉伪影和撕裂，最长允许刷新时间必须小于1/(显示屏刷新频率)。

使用以下公式计算最长允许刷新时间：

$$\text{最长刷新时间} = 1 / \text{显示屏刷新频率}$$

例如，如果显示屏的内部刷新频率为60 Hz，则最长允许刷新时间为 $1/60 \text{ Hz} = 16.6 \text{ ms}$ 。

可使用以下公式计算DSI链路BW：

$$\text{DSI链路BW} = \text{FB大小} / \text{刷新时间} = \text{HACT} \times \text{VACT} \times \text{色深} / \text{刷新时间}$$

使用最长刷新时间计算最小链路BW。它是可以避免显示屏侧视觉伪影的最小BW。

$$\text{最小DSI链路BW} = \text{HACT} \times \text{VACT} \times \text{色深} / \text{最长刷新时间}$$

例如对于320 x 320显示屏，色深为16 bpp，显示屏内部刷新频率为60 Hz：

$$\text{FB大小 (KB)} = 320 \times 320 \times 2/1024 = 200 \text{ KByte}$$

$$\text{最小链路宽度} = 320 \times 320 \times 16 / 0.016 = 102 \text{ Mbit/s}$$

使用一个速率为102 Mbit/s的通道可支持此带宽。

可使用以下公式计算LTDC像素时钟：

$$\text{LTDC像素时钟} = \text{DSI链路BW} / \text{色深}$$

在本例中，LTDC像素时钟 =  $102 / 16 = 6.3 \text{ MHz}$

如果系统使用单缓冲区存储帧缓冲，则必须使GRAM刷新时间最小化，以便提供足够的时间用于图形计算。这可以通过增加DSI链路速度来实现。

GRAM刷新时间的计算如下：

$$\text{刷新时间} = \text{HACT} \times \text{VACT} \times \text{色深} / \text{DSI链路BW}$$

如果链路速度增加至500 Mbit/s，则刷新显示屏GRAM所需的时间为：

$$\text{刷新时间} = 320 \times 320 \times 16 / 500 \sim 3 \text{ ms.}$$

这为图形计算留下了 $16.6\text{ms} - 3\text{ms} = 13.6\text{ms}$ 的时序。

注：

当DSI链路速度增加时，像素时钟也会增加。

在本例中，LTDC像素时钟增加至 $500 / 16 = 31.25 \text{ MHz}$ ，但用户必须注意系统限制，以免LTDC侧发生FIFO欠载问题。

必须根据以下因素评估可保证最短刷新时间的最大链路速度：

- 最大DSI链路物理限制（请参见[第 7.1 节：DSI链路最大带宽对LTDC像素时钟的影响](#)）。
- 系统限制对LTDC像素时钟的影响（请参见[第 7.2 节：系统限制对LTDC像素时钟的影响](#)）。

## 8 DSI主机应用示例

本节提供了使用DSI主机驱动显示屏的一些用例。DSI主机工作模式是根据显示屏特性来选择的。[第7节：DSI主机性能](#)中介绍了对链路带宽和帧缓冲器大小的要求。

### 8.1 小尺寸显示屏驱动示例

在本例中，驱动的是像素分辨率为 $320 \times 320$ 、色深为16bpp的小尺寸显示屏。

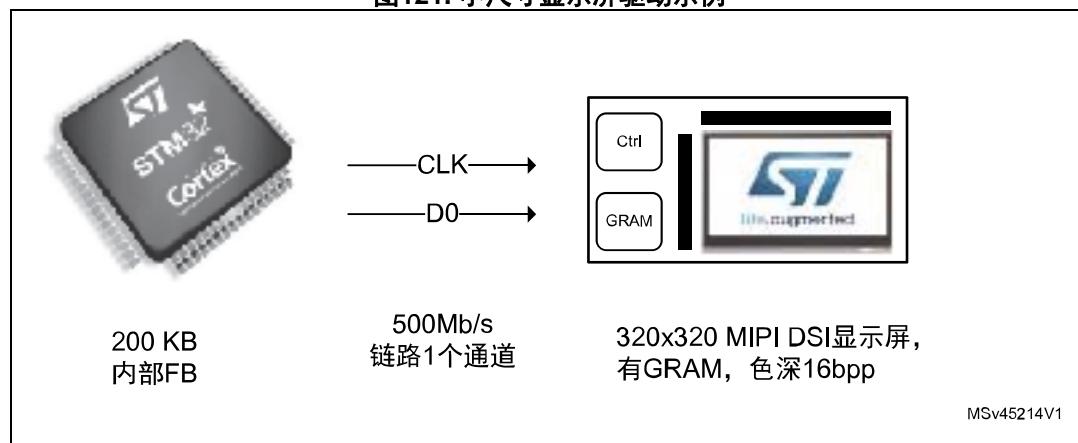
显示屏集成了内部GRAM，用于存储帧缓冲。因此，可使用指令模式驱动显示屏。

需要的帧缓冲器大小为200 KByte，可嵌入内部SRAM。

为了在MCU侧使用单缓冲区，将通道速率设置为500 Mbit/s，以便加快刷新速度并为图形计算留下更多时间。

[图 121](#)显示了以指令模式驱动的小尺寸显示屏的配置示例。

图121. 小尺寸显示屏驱动示例



### 8.2 大尺寸显示屏驱动示例

在本例中，驱动的是色深为24 bpp的 $800 \times 480$ 显示屏。

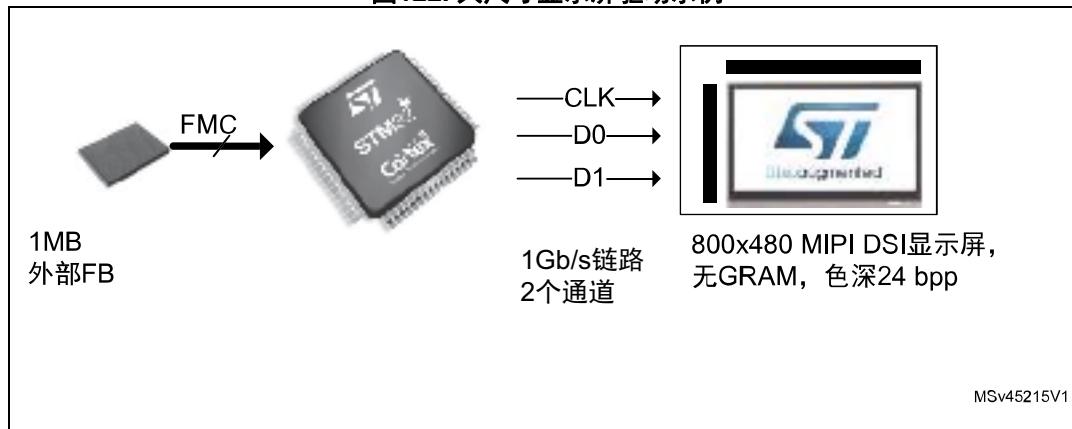
该显示屏没有集成内部GRAM，因此必须使用视频模式。

需要的帧缓冲器大小约为1MByte，因此必须使用外部存储设备。可通过FMC驱动外部SRAM或SDRAM，以便存储帧缓冲。

为确保显示屏侧所需的刷新速率，必须使用两个通道。

[图 122](#)显示了以视频模式驱动的大尺寸显示屏的配置示例。

图122. 大尺寸显示屏驱动示例



## 9 支持的设备

STM32 DSI主机支持MIPI DSI规范中定义的所有工作模式。

它支持指令模式和所有视频模式类型（突发模式、使用同步脉冲的非突发模式、使用同步事件的非突发模式）。

它支持每个通道最高500 Mbit/s的数据速率，当使用两个数据通道时，总链路速度为1 Gbit/s。

它可以连接任何DSI兼容的显示屏。

## 10 结论

使用DSI主机的STM32 MCU代表了连接显示屏的最有效方式，这要得益于DSI协议的引脚数量少和低功耗特性。

本应用笔记介绍了支持STM32 DSI主机连接各种显示屏的功能和特性。

此外，还介绍了STM32 DSI主机的可扩展架构和各种工作模式，它们为客户提供了灵活性。

DSI主机适配指令模式是连接显示屏的最合适方式，这要得益于独热刷新功能及其低功耗特性，但是这会增加显示屏侧的成本，因为需要完整帧缓冲GRAM。

对于没有集成控制器和GRAM的显示屏，最佳选择是视频突发模式，因为它是能效最高的视频模式。但是，它需要在显示屏侧存储视频行。

对于没有GRAM和行缓冲区的低成本显示屏，STM32DSI主机支持非突发模式，以便连接此类显示屏。

介绍了使用STM32CubeMX时的分步配置，并提供了代码示例以便用户迅速开始应用开发。

## 11 版本历史

表38. 文档版本历史

日期	版本	变更
2017年2月10日	1	初始版本。

表39. 中文文档版本历史

日期	版本	变更
2017年11月14日	1	中文初始版本。

**重要通知 - 请仔细阅读**

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利