

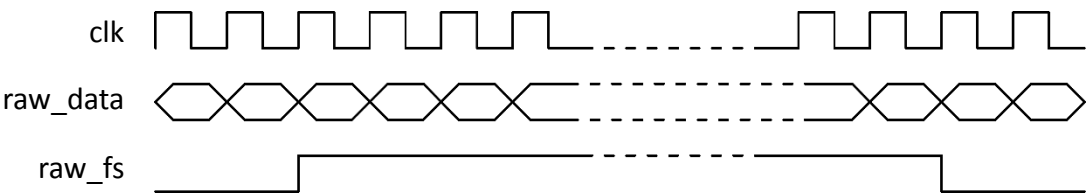
Raw-to-VIP Bridge IP Core

Raw-to-VIP Bridge IP Core 可以接收多种格式的视频流然后转换成 VIP 视频格式, Raw-to-VIP Bridge IP Core (简称 raw_vip_bridge)还提供了 一个可选的接口, 使您可以在 IP 核运行时实时更改它的参数.

raw_vip_bridge 总共可以接收三种视频流格式, 都通过 Avalon-ST 接口输入, 包括 FS 格式, PART_ST 格式和 RAW_ST 格式.

Raw Video Format

Figure 1-1: FS Format



FS 格式仅包含一组 data 总线和一个 FS 信号, 当 FS 信号有效时表示数据线上的数据有效. 特别地, 在 FS 模式中, 以 FS 有效的第一个数据做为图像的 第一个像素, FS 有效的最后一个数据做为图像的最后一个像素, 在此期间不允许拉低 FS 信号, 否则会 使图像数据传送出现错误.

PART_ST 视频格式为不支持反压的 Avalon-ST 格式, RAW_ST 视频格式为支持反压 的 Avalon-ST 格式, 这两种格式都符合标准的 Avalon-ST 定义, 因此不再详述.

Raw-to-VIP Bridge Parameter Settings

Table 1-1: raw_vip_bridge Parameter Settings

Parameter	Value	Description
Data Bits	4-32, Default = 8	每个颜色分量的数据宽度.
Data Planes	1-3, Default = 1	颜色分量的数量.
Data Mode	<ul style="list-style-type: none">• FS• PART_ST• RAW_ST	选择输入视频的格式.
Video In and Out Use Same Clock	On or Off	打开该选项会把输入与输出数据流同步到同一个时钟. 该选项仅在 Data Mode 选为 FS 或 PART_ST 时可用. 选为 RAW_ST 时该选项为 On.

Parameter	Value	Description
FIFO Depth	512-32768, Default = 4096	缓存数据所使用的 FIFO 的深度.
FIFO Use	On or Off	打开这个选项会导出内部 FIFO 所使用的容量. ⁽¹⁾
Video Width	Integer, Default = 720	输出视频流控制包中的宽度信息.
Video Height	Integer, Default = 576	输出视频流控制包中的高度信息.
Video Interlaced	<ul style="list-style-type: none"> • Progressive • F0 • F1 	输出视频流控制包中的交错信息.
Runtime Control	On or Off	打开该选项会增加一个接口用于实时更改参数.
Control Mode	<ul style="list-style-type: none"> • Avalon-MM • Export 	选择使用 Avalon-MM 接口还是简单的导出数据总线.

(1) 与 FIFO 有关的参数仅在 Data Mode 选择为 FS 或 PART_ST 时有效, 因为 RAW_ST 模式下不需要内部 FIFO.

Raw-to-VIP Bridge Signals

Table 1-2: Common Signals

这些信号会随着 raw_vip_bridge 的例化而生成.

Signal	Direction	Description
vst_clk	Input	输出视频流的主时钟.
vst_rst_n	Input	输出模块会在该复位信号为低时异步复位.
raw_clk	Input	输入原始视频流的主时钟. 在 Video In and Out-Use Same Clock 选择 Off 时生成
raw_rst_n	Input	输入模块的复位信号. 在 Video In and Out-Use Same Clock 选择 Off 时生成
raw_data	Input	raw 端口 Avalon-ST 的 data 总线, 视频信号通过该总线传输进 IP 核.
raw_fs	Input	该信号指示 raw_data 上的数据是否有效.

Signal	Direction	Description
raw_valid	Input	raw 端口 Avalon-ST 的 valid 信号, 该信号指示此时 data 总线上的数据是否有效. 仅在 Data Mode 选择不为 FS 时生成.
raw_startofpacket	Input	raw 端口 Avalon-ST 的 startofpacket 信号, 该信号标志了一个 Avalon-ST 包的开始. 仅在 Data Mode 选择不为 FS 时生成.
raw_endofpacket	Input	raw 端口 Avalon-ST 的 endofpacket 信号, 该信号标志了一个 Avalon-ST 包的结束. 仅在 Data Mode 选择不为 FS 时生成.
raw_ready	Output	raw 端口 Avalon-ST 的 ready 信号, 当 IP 核准备好接收数据时该信号置位. 仅在 Data Mode 选择为 RAW_ST 时生成.
fifo_max_usedw	Output	内部 FIFO 所使用的空间的最大值.
dout_data	Output	dout 端口 Avalon-ST 的 data 总线, IP 核通过该总线输出视频信号.
dout_ready	Input	dout 端口 Avalon-ST 的 ready 信号, 当下游的器件准备好接收数据时置位该信号.
dout_valid	Output	dout 端口 Avalon-ST 的 valid 信号, 该信号指示此时 data 总线上的数据是否有效.
dout_startofpacket	Output	dout 端口 Avalon-ST 的 startofpacket 信号, 该信号标志了一个 Avalon-ST 包的开始.
dout_endofpacket	Output	dout 端口 Avalon-ST 的 endofpacket 信号, 该信号标志了一个 Avalon-ST 包的结束.

Table 1-3: Control Signals

这些信号只会在 `raw_vip_bridge` 参数编辑器里将 **runtime control** 选项打开时出现.

Signal	Direction	Description
<code>av_clk</code>	Input	<code>av_control</code> 从端口的主时钟.
<code>av_rst_n</code>	Input	<code>av_control</code> 从端口的复位信号, 低电平有效.
<code>av_address</code>	Input	<code>av_control</code> 从 Avalon-MM 的 <code>address</code> 总线, 该地址指向某一寄存器, 单位为字(word)偏移.
<code>av_read</code>	Input	<code>av_control</code> 从 Avalon-MM 的 <code>read</code> 信号, 当您置位该信号时, <code>av_control</code> 从端口会将读数据发送到 <code>readdata</code> 总线上.
<code>av_readdata</code>	Output	<code>av_control</code> 从 Avalon-MM 的 <code>readdata</code> 总线, <code>av_control</code> 从端口通过该总线输出读数据.
<code>av_readdatavalid</code>	Output	<code>av_control</code> 从 Avalon-MM 的 <code>readdatavalid</code> 信号, 该信号用来表明此时 <code>readdata</code> 总线上的数据是否有效.
<code>av_waitrequest</code>	Output	<code>av_control</code> 从 Avalon-MM 的 <code>waitrequest</code> 信号, 当该信号置位时, <code>av_control</code> 从端口会忽略一切读写请求.
<code>av_write</code>	Input	<code>av_control</code> 从 Avalon-MM 的 <code>write</code> 信号, 当您置位该信号时, <code>av_control</code> 从端口会从 <code>writedata</code> 上接收新数据.
<code>av_writedata</code>	Input	<code>av_control</code> 从 Avalon-MM 的 <code>writedata</code> 信号, <code>av_control</code> 从端口通过该总线接收写数据.
<code>av_irq</code>	Output	<code>av_control</code> 从 Avalon-MM 的 <code>interrupt</code> 信号, 该信号的置位意味着一帧处理完毕, 产生中断.
<code>im_width</code>	Input	视频宽度信息.
<code>im_height</code>	Input	视频高度信息.
<code>im_interlaced</code>	Input	视频交错信息.

Raw-to-VIP Bridge Control Registers

Table 1-4: raw_vip_bridge Control Register Map

视频的控制包参数会在每帧的开始读取到 IP 核内部缓存，因此有关的设置寄存器可以安全地在处理数据时更新.

Address	Register	Description
0	Control	<ul style="list-style-type: none">• Bit 0 为运行寄存器. 设置该位为 0 会使得 raw_vip_bridge 停止工作.• Bit 1 为中断使能寄存器. 设置该位为 1 使能每帧处理完毕时的中断.
1	Status	Bit 0 是状态位, 其他位都没有使用. 当 raw_vip_bridge 在处理数据时会将该位置 1.
2	Interrupt	Bit 1 为帧尾中断, 其他位都没有使用. 向该地址写任意数会复位帧尾中断.
3	Video Width	视频宽度信息.
4	Video Height	视频高度信息.
5	Video Interlaced	视频交错信息.

Table 1-5: Video Interlaced Data

视频交错信息取值与含义的对应关系如该表所示.

Hex	Bin	Description
4'h2	4'b0010	Progressive
4'hA	4'b1010	Interlaced F0
4'hE	4'b1110	Interlaced F1

Document Revision History

Data	Version	Changes
October 2015	1.3	增加了 Control Mode 选项
September 2015	1.2	将“计算 FIFO 最大使用容量”选项更改为“导出 FIFO 当前使用容量”
September 2015	1.1	增加了 Video In and Out Use Same Clock 选项
July 2015	1.0	第一次发布