基于 FPGA 的系统通过合成两条视频流来提供 3D 视频

作者：Witold Kaczurba

简介

视频系统，目前已经深入消费应用的各个方面，在汽车、机器人和工业领域日益普遍。其在非消费应用中的增长主要源于 HDMI 标准以及更快、更高效的 DSP 和 FPGA 的出现。

本文将概要讨论利用模拟或 HDMI 摄像机实现立体视觉（3D 视频）的各种要求。文章将描述一个基于 FPGA 的系统，它将两个视频流结合成一个 3D 视频流，通过 HDMI 1.4 发射器进行传输，同时还要介绍一个基于 DSP 的系统，与通常需从两台摄像机接收数据相比，该系统可以节省 DMA 带宽。另外，本文还将描述一种方法，该方法可以实现一种并排格式，可供 3D 摄像机或要求 3D 视频的系统使用。

概述

立体视觉要求使用两台摄像机，二者相距大约 5.5 厘米，这是人类双眼之间的典型间距，如图 1 所示。

图 1 支架上的两台摄像机（已针对立体视觉对齐）

图 2 所示高级功能框图使用了采用相同视频标准的两台同步摄像机、两个视频解码器和一个 FPGA。为了确保完全一致的帧速率，摄像机必须行锁定到共同的参考时序。如果没有同步，不使用外部存储器，就不可能将输出组合起来并存储为完整的视频帧。

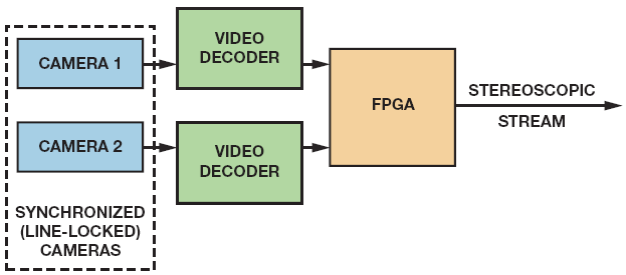

图 2 高级功能框图

图 3 显示两个行锁定视频流被合并成一个立体图像。图 4 显示，如果不将整个视频帧保存在外部存储器中，则异步视频流无法合并。

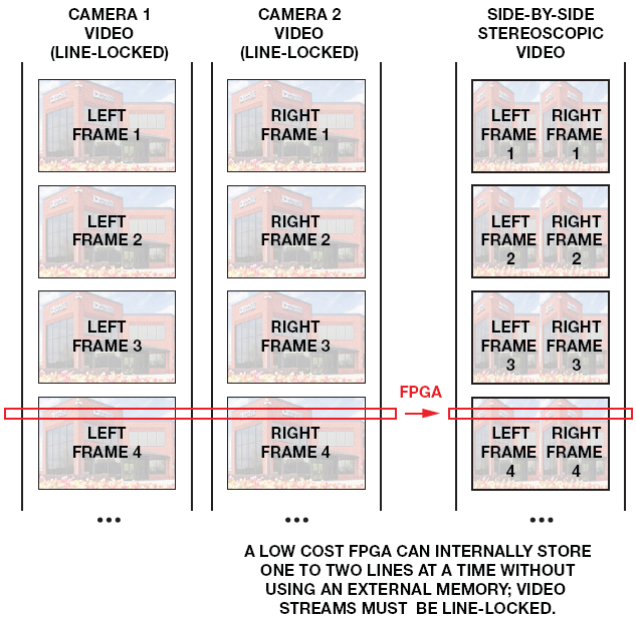

图 3 合并两个同步视频流

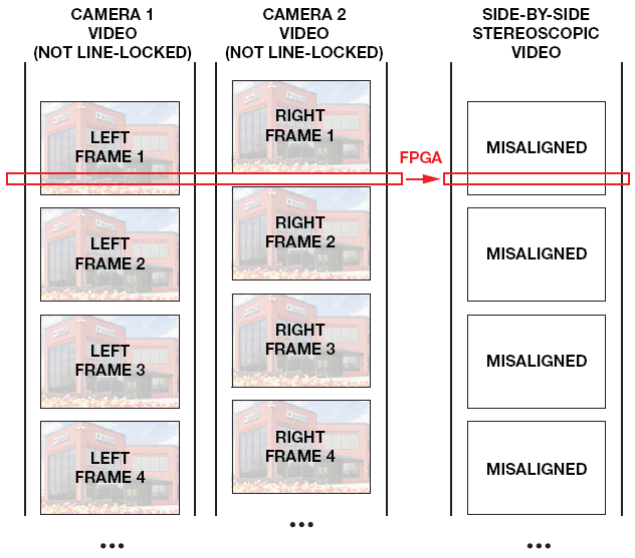

图 4 如果不使用外部存储器，则无法合并异步视频流

然后，两台同步摄像机的输出由视频解码器（如ADV7181D、ADV7182或ADV7186，用于模拟摄像机）进行数字化处理；也可由HDMI接收器（如ADV7610或ADV7611，用于数字摄像机）进行数字化处理。

视频解码器和HDMI接收器都采用内部锁相环(PLL)在其输出总线上产生时钟和像素数据。这意味着，在数字化模拟视频，或者接收HDMI流时，将为两台摄像机产生两个独立的时钟域。另外，两个视频流可能存在对齐误差。这些时序差异和对齐误差必须在后端器件（如FPGA）中进行补偿，先将数据带至共同的时钟域，然后再将两个视频图像结合成单个立体视频帧。然后，通过一个支持 3D 的HDMI 1.4 HDMI发射器（如ADV7511或ADV7513）发送同步后的视频流——也可以将其提供给DSP（如ADSP-BF609 Blackfin®处理器）以便进一步处理。

时钟架构

视频解码器有两种完全不同的时钟源，具体取决于其是否锁定。当视频 PLL 被锁定至输入同步信号时——水平同步（视频解码器）或 TMDS 时钟(HDMI)——结果会产生一个锁定至输入视频源的时钟。当视频失锁时，或者当 PLL 处于强制自由运行模式时，视频 PLL 不会锁定至输入同步信号，结果会产生一个锁定至晶振时钟的时钟输出。另外，时钟可能不会在复位后输出，因为 LLC 时钟驱动器在复位后设置为高阻抗模式。

因此，如果系统有两个或多个始于视频解码器或 HDMI 接收器的视频路径，即使将同一晶振时钟提供给两个视频解码器或 HDMI 接收器，仍会有两个不同频率、不同相位的不同时钟域，因为每个器件都会基于自己的 PLL 产生自己的时钟。

带锁定视频解码器的同步系统

典型的立体视频使用两个视频源，其中，每个视频解码器都会锁定至输入视频信号，并会基于输入水平同步或 TMDS 时钟产生自己的时钟。当两台摄像机同步——或行锁定至同一参考时序时——分帧线将始终对齐。由于两个独立的视频解码器会收到相同的水平同步信号，因此，像素时钟将拥有相同的像素时钟频率。这样，就可以将两条数据路径带入同一个时钟域，如图 5 所示。

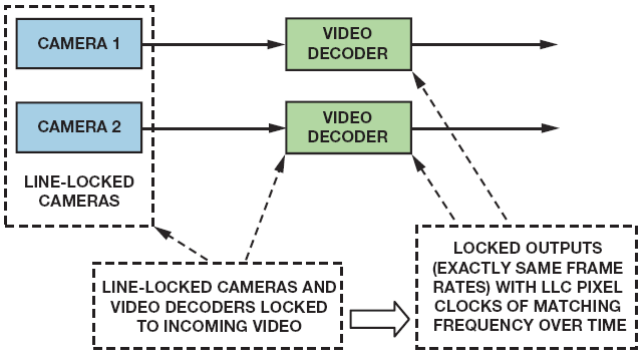

图 5 同步至同一参考源的两台摄像机。两个视频解码器都会收到同一同步信号，因此，它们也会锁定

异步视频系统

不幸的是，其中一个视频解码器可能因视频源信号质量欠佳而失锁，如图 6 所示；或者，摄像机因视频链路断开而失去同步性，如图 7 所示。这会在两条数据路径中导致不同的频率，结果又会导致进入后端中的数据量不对称。

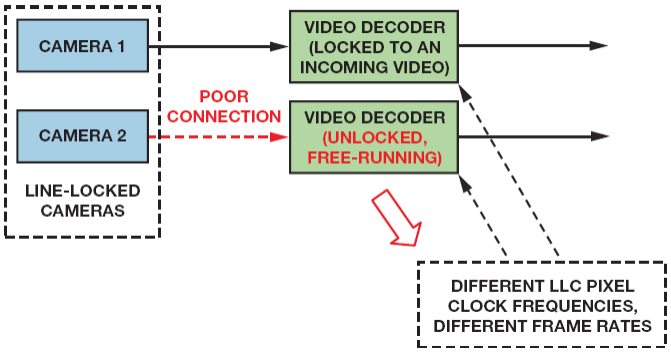

图 6 带未锁定视频解码器的行锁定摄像机

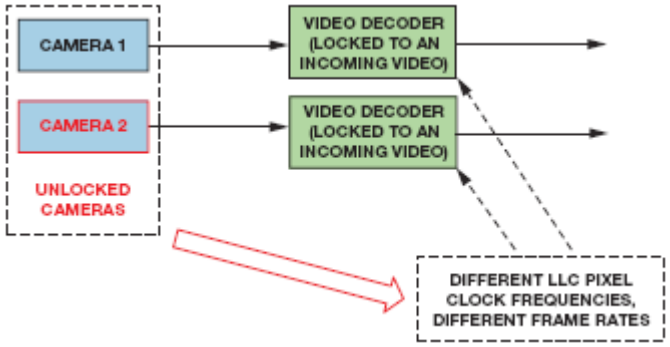

图 7 带锁定视频解码器的未锁定摄像机

视频失锁可以通过使用一个中断(SD 视频解码器为 SD_UNLOCK，分量视频解码器为 CP_UNLOCK，或 HDMI 接收器中的 TMDSPLL_LCK 寄存器)来检测，该中断会在一定延迟后介入。视频解码器集成了不稳定水平同步平滑机制，因此，视频失锁的检测可能需要两三行。该延迟可通过控制 FPGA 中的失锁来减少。

时钟三态模式

在设计 FPGA 时钟资源时，必须知道，默认情况下，许多视频解码器和 HDMI 产品在复位后将时钟和数据线路置为三态模式。因此，LLC 像素时钟不适用于同步复位。

两条视频流中的数据对齐误差

为了简化系统并减少合并两幅图像所需存储器，到达 FPGA 的数据应进行同步，以使来自第一台摄像机的第 M 行第 N 个 像素与来自第二台摄像机的第 M 行第 N 个 像素同时收到。

在 FPGA 输入端，这可能很难实现，因为两条视频路径可能具有不同的延迟：行锁定摄像机可能输出存在对齐误差的行，不同的连接长度可能加大对齐误差，而视频解码器则可能带来可变启动延迟。受这些延迟影响，采用行锁定摄像机的系统会有一些存在对齐误差的像素。

行锁定摄像机对齐误差

即使是行锁定摄像机也可能输出存在对齐误差的视频行。图 8 显示来自两台摄像机的 CVBS 输出端的垂直同步信号。一台摄像机（同步主机）为第二台摄像机（同步从机）提供行锁定信号。380 ns 的对齐误差是清楚可见的。图 9 展示的是这些摄像机输出端的视频解码器传输的数据。可以看到 11 个像素的位移。

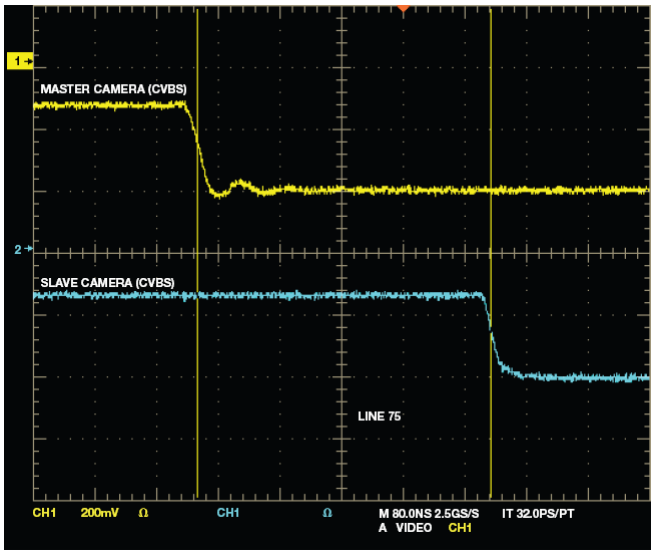

图 8 行锁定视频摄像机之间的 380 ns 视频对齐误差

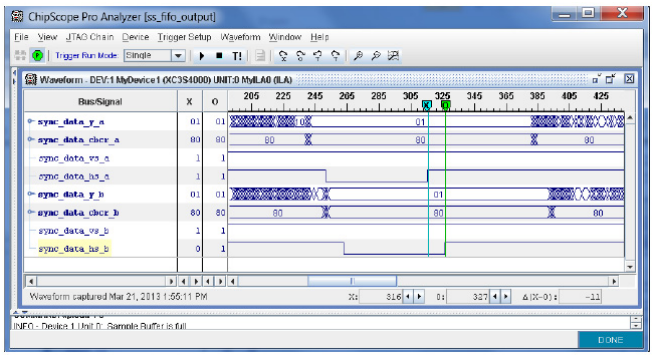

图 9 数字域中未补偿的 11 个像素的视频对齐误差

不同的连接长度

所有电气连接都会带来传播延迟，因此，要确保两条视频路径具有相同的轨道和电缆长度。

视频解码器/HDMI 接收器延迟

所有视频解码器都会带来可能因启用的功能而异的延迟。另外，有些视频器件含有可能增加随机启动延迟的因素——如深色 FIFO。采用视频解码器的典型立体系统的随机启动延迟大约为 5 个像素时钟。含有 HDMI 发射器和接收器的系统（如图 10 所示）的随机启动延迟可能为 40 个像素时钟左右。

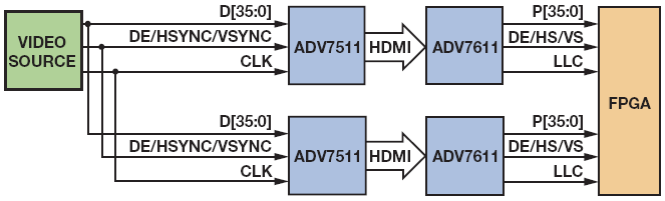

图 10 流水线延迟测量设置

对齐误差补偿

图 11 所示系统中，一个视频解码器对来自各摄像机的模拟信号进行数字化处理。各视频路径的数据和时钟是独立的。两条视频路径都连接至 FIFO，后者对输入数据进行缓冲，以补偿数据对齐误差。在输出数据时，FIFO 使用来自其中一个解码器的共用时钟。在锁定系统中，两条数据路径应具有完全相同的时钟频率，以确保在摄像机行锁定且视频解码器锁定的情况下，不会出现 FIFO 溢出或下溢现象。

通过启用或禁用 FIFO 输出，控制模块可以维持 FIFO 电平以尽量减少像素对齐误差。如果采取了正确的补偿措施，则 FPGA 模块的输出应为与第一个像素对齐的两条数据路径。然后该数据提供给 FPGA 后端，以生成 3D 格式。

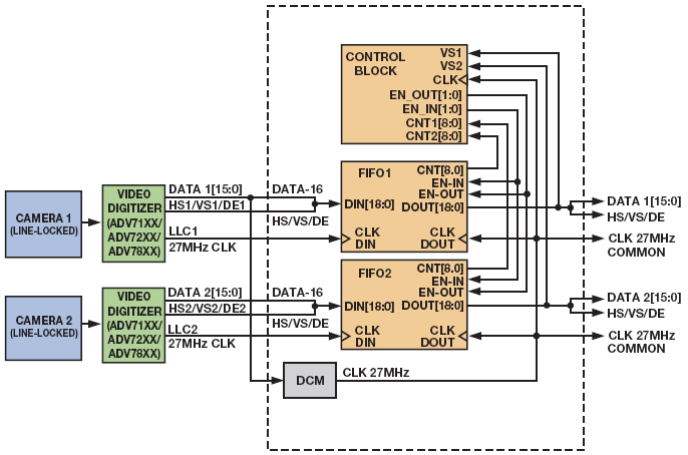

图 11 使用数字 FIFO 来重新对齐视频图像

对齐误差测量

两个数字化数据流之间的对齐误差可以在视频FIFO输出端进行测量，其方法是使用一个单一时钟计数器，该计数器在输入信号之一的垂直同步(VS)脉冲上复位。图 12 所示两个视频流(vs_a_in 和 vs_b_in）的对齐误差为 4 个像素。计数器使用列表 1 中所示方法测量对齐误差。计数从 VS1 的上升沿开始，并在 VS2 的上升沿终止。

如果一个帧的总像素长度是已知的，则可以通过从帧长中减去计数值，从而算出负偏斜（VS2 位于 VS1 之前）。该负值应在偏斜超过像素帧长的一半时计算。结果应用来重新对齐 FIFO 中存储的数据。

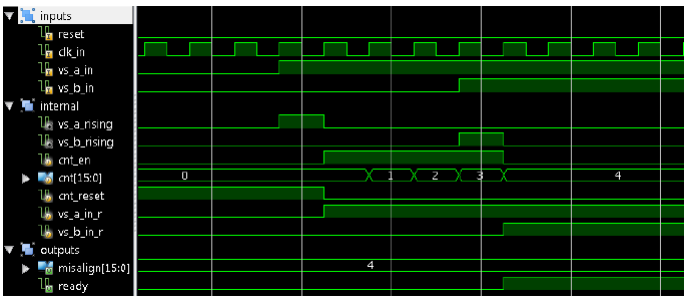

图 12 对齐误差测量

列表 1 简单对齐误差测量(Verilog®)。

```
module misalign_measurement(
 input wire reset,
 input wire clk_in,
 input wire vs_a_in,
 input wire vs_b_in,
 output reg [15:0] misalign,
 output reg ready);

reg [15:0] cnt;
reg cnt_en、cnt_reset;
reg vs_a_in_r、vs_b_in_r;
使 vs_a_rising = vs_a_in > vs_a_in_r;
使 vs_b_rising = vs_b_in > vs_b_in_r;

always @(posedge clk_in)
begin
 vs_a_in_r <= vs_a_in;
 vs_b_in_r <= vs_b_in;
end

always @(posedge clk_in)
 if (reset)
 begin
 { ready, cnt_en } <= 2'b00;
 misalign <= 0;
 end else begin
 if ((vs_a_in == 1'b0) && (vs_b_in == 1'b0))
 { ready, cnt_reset } <= 2'b01;
 else
 cnt_reset <= 1'b0;
 /* beginning */
 if (vs_a_rising && vs_b_rising)
 begin
 misalign <= 0;
 { ready, cnt_en } <= 2'b10;
 end
 else if ((vs_a_rising > vs_b_in) || (vs_b_
 rising > vs_a_in))
 { ready, cnt_en } <= 2'b01;

 /* ending */
 if ((cnt_en == 1'b1) && (vs_a_rising || vs_b_
 rising))
 begin
 { ready, cnt_en } <= 2'b10;
  misalign <= vs_a_rising ? (-(cnt + 1)) : (cnt + 1);
 end
 end

always @(posedge clk_in) /* counter */
 if ((cnt_reset) || (reset))

 cnt <= 0;
 else if (cnt_en)
 cnt <= cnt + 1;

endmodule
```

从两个对齐视频流生成 3D 视频

一旦像素、行和帧数据都真正同步，FPGA 可以将视频数据转换成 3D 视频流，如图 13 所示。

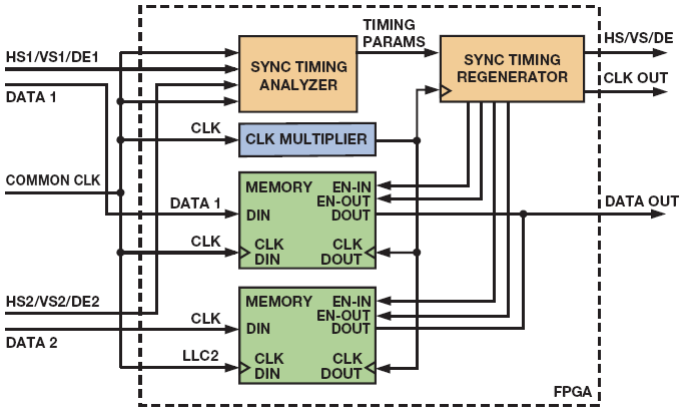

图 13 用于实现 3D 格式的简化架构

输入数据由共用时钟读入存储器。同步时序分析仪检查输入的同步信号，并抽取视频时序，包括水平前后沿长度、垂直前后沿、水平和垂直同步长度、水平有效行长、垂直有效行数和同步信号极化。将该信息与当前水平和垂直像素位置一起传给同步时序再发生器，这样可以生成经修改的时序，以便支持所需 3D 视频结构。新生成的时序应延迟，以确保 FIFO 含有所需数据量。

并排 3D 视频

对存储器要求最低的架构是并排格式，只需要一个两行缓冲器(FIFO)即可存储来自两个视频源的行内容。并排格式的宽度应为原始输入模式的两倍。为此，应使用一个双倍时钟来为拥有双倍水平行长度的再生同步时序提供时钟。用于为后端提供时钟的双倍时钟将以双倍速率清空第一个 FIFO 和第二个 FIFO，这样即可并排显示图像，如图 14 所示。并排图像如图 15 所示。

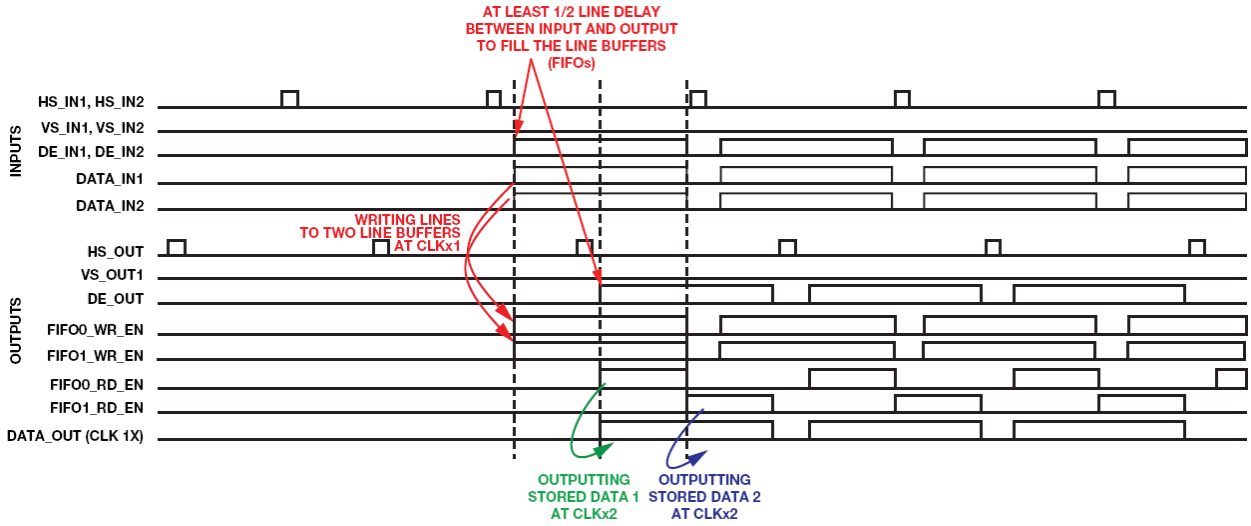

图 14 使用简单的 FPGA 行缓冲器来并排合并两幅图像

图 15 视频时序下的并排 576p 图像

结论

ADI 公司的解码器和 HDMI 产品以及简单的后处理技术可以打造出真正的立体 3D 视频，并为其传输提供条件。如本文所示，用简单的数字模块，无需使用昂贵的存储器，即可实现 3D 视频。这种系统可用于需要 3D 视觉的任何类型的系统中，从简单的摄像机，到基于 ADSP-BF609 DSP 的可以跟踪物体及其距离的专业系统。

作者简介

Witold Kaczurba [witold.kaczurba@analog.com]是ADI公司高级电视部（爱尔兰利默里克）资深应用工程师，负责支持解码器和HDMI产品。他从波兰弗罗茨瓦夫理工大学毕业并获得电气工程硕士学位后，于 2007 年加入ADI公司。学生期间，他曾为小型电子和IT公司工作，后来作为工读生加入ADI公司（爱尔兰），随后成为应用工程师。