White Paper

Broadcast, Medical and Industrial
Agilex™ 7 FPGA

Designing Genlocked Video Systems with Deterministic Low Latency on FPGAs

Authors

Snaider Carrillo

Computer Vision Engineer
Embedded Acceleration Division
Altera Corporation
Lyon, France

Alexey Lopich

Director of Engineering
Embedded Acceleration Division
Altera Corporation
Marlow, UK

Abstract

Video data synchronization and low-latency support are two key features that require special attention when architecting an FPGA video processing pipeline. When input and output video data are correctly synchronized, it helps avoid any data corruption, and depending on the type of synchronization implemented on the video pipeline, it positively impacts memory footprint and low latency, as it could help reduce or entirely remove video frame buffers from the system. If video data synchronization or low-latency support features are needed in a video system, having an IP core library that allows the receiving, processing, moving, and transmitting of video data is insufficient. For a complete solution, a video library must be complemented with IP cores that allow extraction and routing timing information, thus facilitating synchronous clock generation at specific points in the video pipeline. This white paper introduces a new IP library and presents a methodology that uses these off-the-shelf IP cores to achieve deterministic video synchronization and low latency on FPGAs.

Introduction

Many imaging and video systems rely on precise timing and alignment between video inputs or between inputs and outputs. The ability to genlock video systems to maintain synchronization between video signals removes the need for video buffering, which introduces additional delay as well as resource, power, and memory bandwidth requirements [1]. Applications such as medical surgical robotics, broadcasting, multi-camera production, industrial and 3D stereoscopic imaging, virtual studios, video walls, and display installations are traditionally underpinned by precise alignment and timing between video sources [2]. FPGAs are widely used in all the above applications as hardware platform enablers due to their inherent reconfigurability, ability to support various interfaces, and parallel processing capabilities [3].

This white paper introduces a new set of clocked-video and genlock IP cores included in the Video and Vision Processing (VVP) Suite IP library [4] and describes various approaches to building systems with external VCXO components. It also covers VCXO-less genlock solutions, which implement similar VCXO functionality using only internal FPGA resources, such as fractional PLLs (fPLLs) and control logic. The latter approach delivers significant system savings due to lower PCB design and component cost. The rest of the content is organized as follows: Section II introduces a new IP library aiming to support clocked-video and genlock applications, Section III describes two different types of genlock solutions, Section IV discusses different architectures that can be used to achieve genlock on an FPGA video system, Section V presents a subsystem hierarchical approach to support genlock system integration, Sections VI briefly describes three example designs that demonstrate each of the genlock solutions discussed in this paper, and Section VII provides conclusions.

Table of Contents

Clocked Video and Genlock IP Cores Library

The VVP Suite is a set of IPs for video, image, and vision processing. The VVP IPs transport video using the Altera® FPGA streaming video protocol [5], built on top of the industry-standard AXI4-Stream protocol, with extensions for transporting control and video data.

The VVP Suite features IPs that range from simple building block functions such as clocked video and genlock, color space conversion, and mixer to sophisticated processing functions that can implement programmable scaling, arbitrary non-linear distortion correction, 3D look-up table, adaptive tone mapping, and many more. Specifically, for applications where clocked video interfaces, video synchronization, or low latency features are required, the VVP IP core library provides a subset of IPs, namely the Clocked Video and Genlock (CVG) IP cores. The CVG is a collection of IP cores derived from the VVP IP library that serves two purposes:

- To facilitate transferring video data between timing-aware (full-raster) clocked video and timing-agnostic (active video) video streaming protocols [5].
- To facilitate synchronization between video input and output pixel clocks and start of frame, based on video timing markers derived from video connectivity IP cores.

The following core functions are needed to implement such functionality in a video processing system on FPGA:

- Clocked Video Input (CVI) – to transfer video data between full-raster and active data streaming interfaces. The IP provides a seamless conversion by removing video timing data from a full-raster bus and leaving just the active pixel data in a streaming video format. Typically, the IP is a bridge to transfer data from video connectivity to the video processing pipeline.
- Clocked Video Output (CVO) – to merge active pixel data from a streaming video bus with the real-time video timing signals from a reference full-raster bus. The output is a streaming video full-raster bus that can be connected to a video connectivity IP.
- Genlock Controller – to provide a control loop system that matches the voltage-controlled crystal oscillator (VCXO) frequency of the selected reference input pixel clock. This IP can be used in an FPGA to support external VCXO tracking to a reference input pixel clock on a video pipeline in genlock mode, avoiding drifting or rolling effects on the output video stream.
- Genlock Signal Router – to extract and route multichannel genlock strobes. This IP passes genlock timing signals to internal or external FPGA multi-rate pixel clock generators. The IP also generates video synchronization signals based on video timing markers derived from the video connectivity IPs.
- Video Timing Generator – to generate real-time timing markers signals (for example, Fsync, Hsync, Vsync) that define a full-raster video. The IP can create any video raster, including interlaced and progressive standards.
- VCXO-less Controller – to generate an output pixel clock synchronized (i.e., genlocked) with an input pixel clock by using only internal FPGA resources, specifically by utilizing transmitter transceiver PHY with dynamic reconfiguration and fractional PLL (fPLL) capabilities. The IP provides a phase frequency detector (PFD) and a proportional integral and derivative (PID) controller to measure differences between the input and output pixel clocks and to generate a control word that allows tracking and locking the phase of the output pixel clock relative to the input pixel clock.

Types of Genlock Solutions

Genlock Definition

The output video timing in a typical video pipeline is controlled by a Video Timing Generator (Vtiming) IP. The Vtiming IP adds horizontal and vertical blanking intervals to the active pixel data coming from the video processing system. These blanking periods, along with active pixels and the output pixel clock, determine the video output's frame period. Assuming that input and output pixel clocks are asynchronous, the difference in clock period caused by clock drift can be corrected by generating an output pixel clock that is phase and frequency locked to a reference input pixel clock.

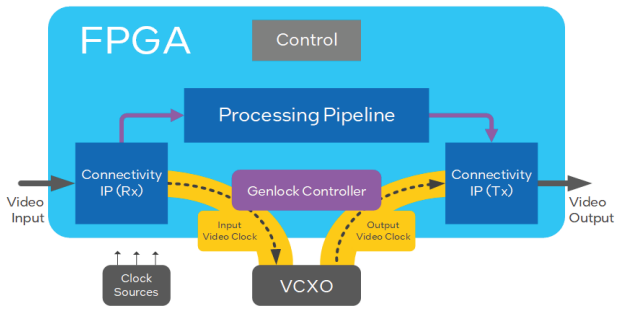

Figure 1. Generic video system architecture with independent video input and output clocks. Genlock allows generating a synchronous output pixel clock by using an input pixel clock as a reference, to ensure the two clocks do not drift apart.

A video processing pipeline is said to be in genlock mode when i) the output pixel clock is generated-locked (i.e., genlock) to the input pixel clock, and ii) the start of each output video frame consistently aligns to the start of each input video frame or a configurable fixed offset from the start of each input video frame. Depending on how the output pixel clock is generated, genlock solutions can be classified as VCXO-based or VCXO-less. The rest of this section describes the two genlock solutions. It shows how to enable video data synchronization and low latency support for FPGA-based video processing systems with the help of off-the-shelf IP cores from the VVP IPs.

VCXO-based Video Synchronization

VCXO-based solutions use external devices to generate the output pixel clock based on a reference input signal, such as a pixel clock or Fsync. Depending on the external VCXO device, the interface to connect the input reference signal can vary; however, two of the most common input reference interfaces are Fsync, Vsync, and Hsync (FVH), and voltage control (Vc).

- VCXO-based Video Synchronization with an FVH interface: The easy way to interface with FVH-type devices is by using the Genlock Signal Router IP, which is typically connected to an RX connectivity IP (for example, HDMI or SDI), from which it extracts FVH signals and output them to an external VCXO device. Fig. 2 shows the input and output interfaces for the Genlock Signal Router IP. This IP can be paired with an external VCXO device, such as the LMH1983 [6], to provide a genlock output pixel clock.
- VCXO-based Video Synchronization with a Voltage Control (Vc) Pin Interface: To connect with VCXO devices with control voltage input capabilities (for example, Si516 [7]), the Genlock Controller IP takes the reference input pixel clock and the generated output pixel clock as inputs, and outputs a pulse width modulation (PWM) type signal proportional to the difference in the input and output clock period caused by clock drift. The PWM-type signal is connected to the Vc input pin of the external VCXO device to reduce the clock drift, so the generated output pixel clock is locked to the reference input clock. Fig. 3 shows the input and output interfaces for the genlock controller IP.

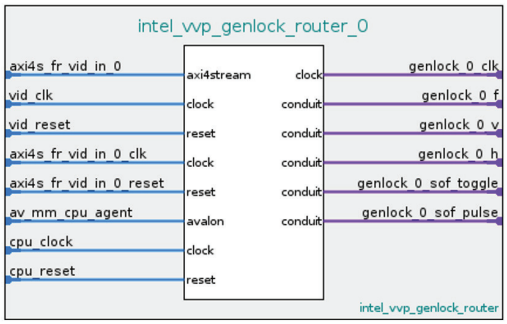

Figure 2. Genlock Router VVP IP with FVH interface enabled.

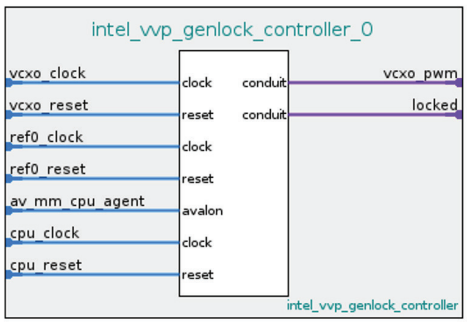

Figure 3. Genlock Controller VVP IP with Vc interface enabled.

VCXO-less Video Synchronization

VCXO-less solutions offer the possibility of eliminating the use of external VCXO devices and instead using the dynamic reconfiguration feature and fPLL capabilities of the transceivers included on the FPGAs to generate a synchronized output pixel clock based on the input reference signal.

To implement such a solution, you must create a VCXO-less Controller Subsystem (Fig. 4), which works with an FPGA transceiver to compare and adjust the input and output pixel clock phase.

The VCXO-less Controller's inputs are connected to the input and output pixel clocks and to the FVH interface provided by the input connectivity IP core (such as HDMI or SDI), while its output is connected to the reconfiguration interface of the TX PHY transceiver.

Suppose there is any difference in terms of frequency between these input and output clock domains. In that case, the IP generates an Avalon® memory-mapped interface transaction to reconfigure the TX fPLL so that both input and output pixel clock frequencies are as close as possible. A VCXO-less Subsystem is a combination of a TX PHY instance and a VCXO-less controller, as shown in Fig. 4. Such subsystem generally consists of the following components:

- A transceiver with dynamic reconfiguration and fPLL capabilities to generate an output pixel clock.
- A PFD and a PID low-pass filter (LPF) controller to measure differences between the input and output pixel clocks and to generate a control word that allows tracking/locking the phase of the output pixel clock relative to the input pixel clock.
- A set of debug control registers to diagnose the status of the video system.

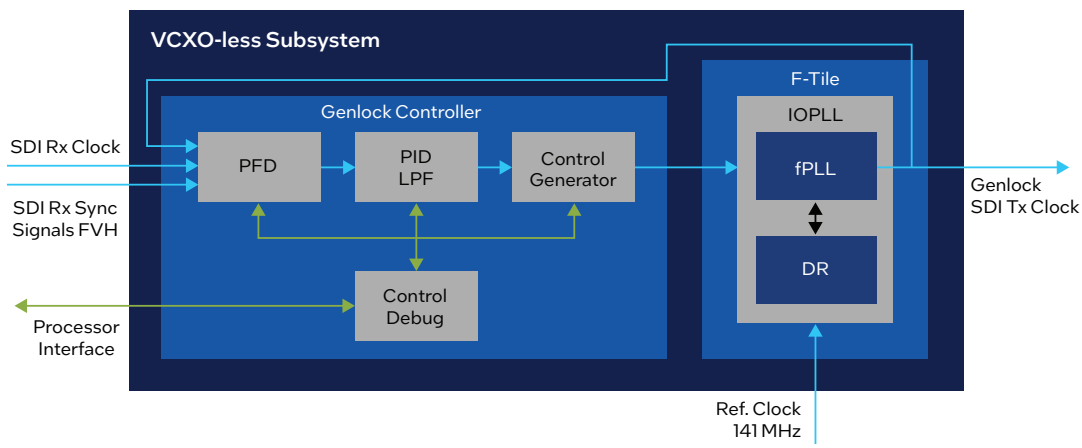

Figure 4. VCXO-less Subsystem

Both VCXO-based and VCXO-less solutions can be used to synchronize pixel clocks for a variety of video applications, including video over IP, where the synchronization of the video and audio data packets usually relies on precision time protocol (PTP) time server (usually referred to as Grandmaster), and its corresponding clock followers. The key to keeping video over IP properly synchronized requires a standardized timestamp mechanism to determine the clock drift (or jitter) between the input and output pixel clock, as defined in ST2059 [8], and adjusting the frequency value of the follower clock that generates the output pixel clock, so that the latency between data packets is kept below the upper limits dictated by the ST2059 standard. Such adjusting of the clock can be achieved by genlocking both clocks using a VCXO-based or VCXO-less approach as described above.

FPGA Genlock System Architecture

Depending on the relationship between input and output pixel clocks, a video processing pipeline can be architected to support three video synchronization modes: free-running, frame-sync, and genlock.

Video Pipeline in Free-Running Mode

In a free-running mode, input and output pixel clocks are independent and asynchronous. Hence, even if the total output frame size, including blankings, is configured to be the same as the input frame size, and the frequency value of the output pixel clock is the same as the input pixel clock, the frame periods of the video input and output would be different due to the intrinsic clock fluctuation (commonly referred as jitter) present in every clock generator.

Consequently, over time, input and output start-of-frame (SOF) signals drift apart, as shown in Fig. 5, where arrows highlighted in blue indicate input (RX) and output (TX) SOF alignment, while arrows in black indicate frame drift at a specific point in time. If not corrected, frame drift could cause the output video frame to jump or roll over sporadically or continuously.

One way to address such drifting over time is to add a video frame buffer to the processing pipeline to decouple the input and output and handle the video synchronization. The frame buffer must support features to allow different input and output video resolutions and frame rates depending on the video pipeline requirements. Such features include double or triple buffering for progressive and interlaced video frames, a configurable memory packing scheme, optional dropping of broken frames, and frame statistics counters. For example, the Video Frame Buffer IP provides such features and allows the creation of video pipelines in free-running mode.

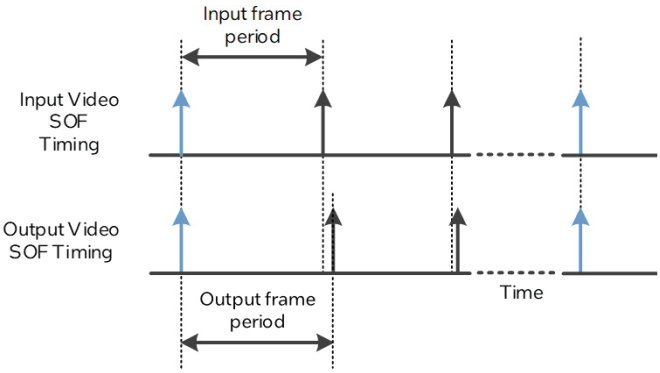

Figure 5. Timing diagram comparing input and output SOF in free-running mode

Video Pipeline in Frame-Sync Mode

Video pipelines in frame-sync mode allow the output raster to track an external timing reference, for example, the Fsync toggle signal, but the output clock does not need to be locked to the input pixel clock. This mode's typical use case scenario includes V-by-One, MIPI, or DisplayPort interfaces, as their packetized-based data exchange methods do not require a pixel-accurate clock.

During frame-sync mode operation, blanking pixels can be dropped or repeated to compensate for frame drift so that the output SOF occurs at a fixed interval. In this mode, there is usually a reference input toggle signal to indicate that a new input SOF has occurred, and it is used to keep input and output SOF signals synchronized.

The Vtiming IP can receive a reference input Fsync toggle signal (frame_start) and use it to support the operation of a video pipeline in frame-sync mode. In this mode, the Vtiming IP adjusts the output video timing to align the output SOF to the frame_start signal on every frame. It causes the output video timing to jump from one area of video blanking to another, resulting in several video blanking pixels being added or removed from the total frame size, compensating for output pixel clock drift. The Vtiming IP must be configured to have one extra line of blanking than is necessary; for example, a total number of 1,125 lines becomes 1,126. It allows output video frames to overrun and add extra video blanking pixels if the frame_start is late.

Additionally, the Vtiming IP can generate an optional pulse signal (genlock_dly) connected to a video frame buffer and used to pace the ingress/egress of video data based on a specific input and output SOF genlock latency value. Fig. 6 shows a typical timing diagram for a video system configured in frame-sync or genlock mode. The figure depicts an Fsync toggle signal (frame_start) and Vtiming IP genlock pulse (genlokc_dly) and their relationship with the input and output SOF signals. The difference between the input and output SOF determines the input-to-output genlock latency.

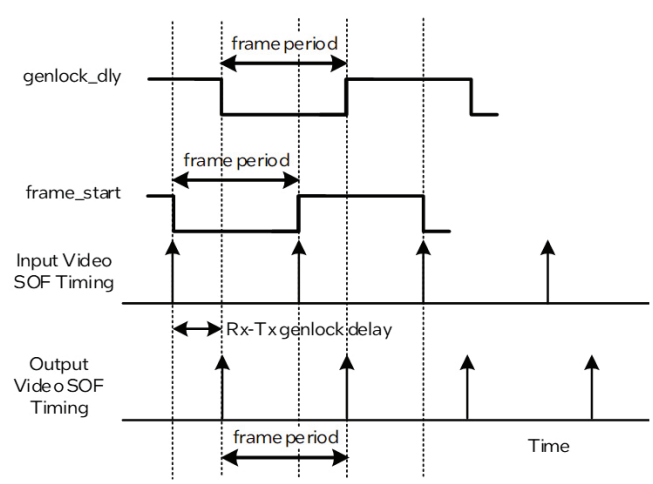

Figure 6. Timing diagram comparing Rx and Tx SOF signal in Frame-Sync and Genlock mode.

Video Pipeline in Genlock Mode

For a video pipeline to be in genlock mode, it needs to satisfy two conditions: 1) the output pixel clock is synchronized with the input pixel clock, and 2) the start of each output video frame consistently aligns with the start of each input video frame or a configurable offset from the start of each input video frame.

Condition #1 can be satisfied by using any of the two solutions discussed in Section II, i.e., VCXO-based or

VCXO-less. Condition #2 can be satisfied by feeding the input frame synchronization signal (Fsync toggle signal) derived from the RX video connectivity core into the Vtiming IP via the frame_start input port. By doing this, the Vtiming IP uses the frame_start to synchronize and generate the output vertical sync/blank, horizontal sync/blank, and SOF signals used by the TX core.

Depending on the low-latency requirements, we could have two possible scenarios to implement a video pipeline in genlock mode:

- Video Pipeline without Video Frame Buffer: For this scenario, the latency is generally less than a frame (i.e., sub-frame latency). However, several video line buffers must satisfy the minimum genlock delay between (input) RX and (output) TX SOF signals.

  Two factors determine the final latency value between RX and TX SOF: i) how fast the output clock generator can lock to the input reference signal and ii) the actual clock frequency value of the RX and TX pixel clocks. Typically, this type of video pipeline requires access to the output clock generator so its frequency value can be fine-tuned to fulfill a specific low latency value. An IP like the Genlock Controller can manually fine-tune the output pixel clock by injecting a specified constant value via its control interface. This is then reflected as a variation on the PWM-type output signal driving the external VCXO device.

- Video Pipeline with Video Frame Buffer: For this scenario, it is possible to achieve sub-frame latency and latencies equal to or bigger than one video frame. Typically, for this type of scenario, the final low latency value can be controlled via a combination of i) internally generated SOF signals (genlock_dly) by the Vtiming IP, which can then be used to pace the video frame buffer and/or ii) a specialized control logic implemented on the frame buffer that allows synchronizing input and output video frames.

Video Subsystem Hierarchical Approach

Fig. 7 shows the proposed hierarchical genlock processing pipeline, which consists of four subsystem hierarchies: video RX, video processing, video TX, and genlock. Even though it is not shown explicitly, it is assumed that the video pipeline also includes a CPU control subsystem that allows system configuration, control, and debugging based on a stack of drivers and software applications.

Video RX Subsystem

This subsystem contains the necessary input (RX) video connectivity IP core (e.g., HMDI, DP, MIPI, or SDI), CVI, color space converter, and other IPs to provide the correct video data format for downstream subsystems. Typically, the video RX subsystem interfaces with the video processing pipeline via a video switch, and this is usually done using a timing-agnostic video protocol, such as the Altera FPGA streaming video [5], as the majority of video processing IP in a typical pipeline does not require any knowledge about the full-raster timing, but just plain active pixel data. The video RX subsystem also interfaces with the genlock subsystem hierarchy to provide the Genlock Signal Router IP with the correct video timing markers derived from the input video interfaces. This timing information is transferred via a timing-aware video protocol, such as clocked video (CV) or Altera streaming video full-raster interfaces.

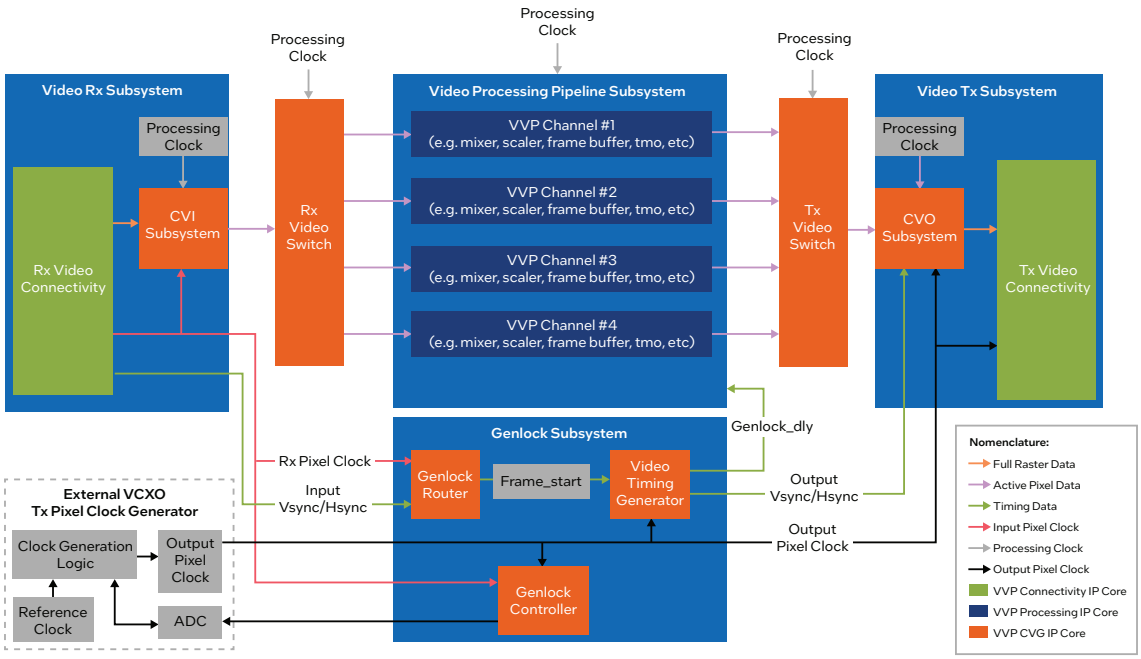

Figure 7. Video System Hierarchy.

Video Processing Subsystem

This subsystem contains all the necessary video processing IP cores to achieve the required video and vision processing capabilities. Typically, such subsystem expects its input and output interfaces to comply with Lite or Full variants of the Altera FPGA streaming video protocol. Both input and output interfaces use a Video Switch IP core to receive video from the RX subsystem and send video to the video TX subsystem. Video switches at this hierarchy's front and back end provide switching capability between multiple sources (e.g., TPG, frame reader, etc.). Also, it offers a natural separation between video connectivity and video processing, creating more abstract and reusable hierarchical templates.

Video TX Subsystem

This subsystem contains necessary output (TX) video connectivity IP cores (for example, HMDI, DP, MIPI-DSI, VoIP, or SDI) and video protocol converter bridge, such as the CVO IP core, to handle and merge timing-agnostics video signals from the processing subsystem and timing-aware signals to generate a full video raster consumed by corresponding connectivity TX IP.

Video Genlock Subsystem

This subsystem contains a Genlock Signal Router IP core, which receives timing information from the video RX subsystem, decodes it, and generates an Fsync toggle signal (i.e., frame_start) used by the Vtiming IP to generate synchronous output video timing signals. The Vtiming IP could be instantiated as a stand-alone component or as part of the CVO IP core.

Due to the inherent jitter present in the clock generators, in genlock mode, the Vtiming accepts a configurable amount of drift in the frame_start signal before resynchronizing the output video timing. This is because the output video clock does not track the input video clock precisely, and there may be some drift before it is aligned. The Vtiming IP has a genlock confidence window centered around the output video SOF. If the frame_start signal changes outside the window, the Vtiming IP immediately resets the output video timing to align the SOF to the frame_start signal. The window can be defined as +/- N pixels.

Finally, and depending on the type of external VCXO used by the video pipeline, we could use the Genlock Signal Router IP core to drive a VCXO with an FVH interface or use the Genlock Controller IP to drive a VCXO that accepts a single-bit (PWM type) voltage control reference input signal to generate the output pixel clock, as shown in Fig. 2 and Fig. 3, respectively.

VCXO and VCXO-Less FPGA Design Examples

To demonstrate the functionality of a video genlock pipeline working on a real-life system, we implemented the following three design examples [9]:

SDI VCXO-less Design Example

The Video SDI VCXO-less design example uses a VCXO-less subsystem and several VVP IP cores as part of its processing pipeline to demonstrate how to implement a genlock and low-latency video system without frame buffers. The design is implemented on an Agilex™ 7 FPGA developer kit, supports video resolutions from 720p up to 4K60, and provides on-chip buffering for up to 10 lines of videos.

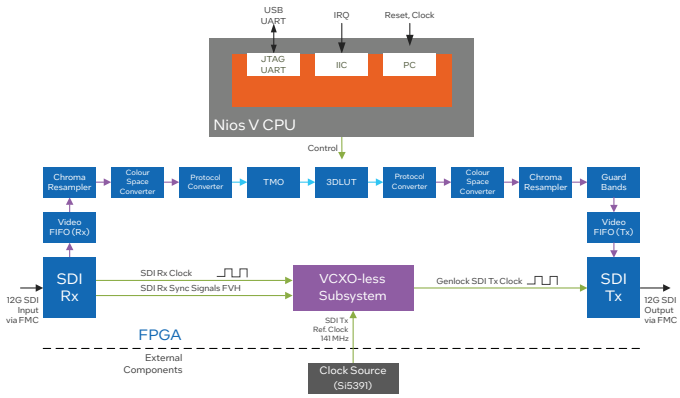

Figure 8. Genlock VCXO-Less Design using on-chip fPLL.

SDI VCXO with FVH Interface Design Example

The Video SDI Genlock Design Example demonstrates SDI Parallel loopback using an external VCXO component. Similar to the VCXO-less design, this design example was also implemented on an Agilex 7 FPGA developer kit. It supports video resolutions from 720p up to 4K60 and provides on-chip buffering for up to 10 lines of videos.

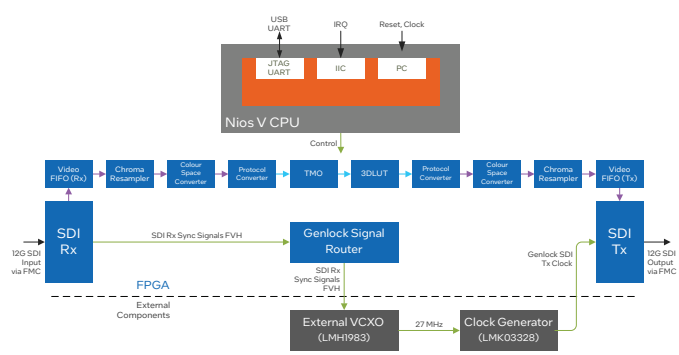

Figure 9. Genlock VCXO-Based Example Design using FVH Interface.

HDMI VCXO with Vc Interface Design Example

The capabilities of the VCXO-based genlock video pipeline solution without using a video frame buffer are demonstrated on an HDMI-based design example implemented on an Arria® 10 GX FPGA development kit equipped with HDMI 2.0 FMC card, supporting 1080p60 and 4K60 video resolutions.

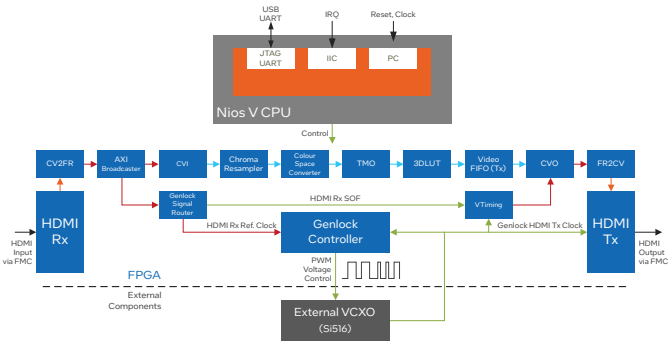

Figure 10. Genlock VCXO-Based Design using PWM-type Interface.

Conclusions

This paper presents an FPGA design methodology for developing genlocked video and imaging systems for latency-sensitive applications using off-the-shelf IP cores. We cover various approaches to building systems with external VCXO components and describe VCXO-less architectures, which implement similar VCXO functionality using only internal FPGA resources, such as fPLLs and control logic.

References

[1] M. Poimboeuf, "Genlock and Timing Regeneration for Multiple Formats of High-Definition Video and Digital Cinema", SMPTE J., vol. 110, pp. 240-247, Apr. 2001.

[2] P. Centen, T. Moelands, J. van Rooy, M. Stekelenburg, "A Multiformat HDTV Camera Head", SMPTE Journal, vol.110, no.8, pp.510-516, 2001.

[3] Thomas Edwards, Warren Belkin, and Andy Bechtolsheim, "Video Processing in an FPGA-Enabled Ethernet Switch", SMPTE Mot. Imag. J, vol. 123, no. 2, pp. 48-52, March 2014..

[4] Altera, "Video and Vision Processing Suite," [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/dsp/video-vision-processing-suite.html

[5] Altera, " Altera FPGA streaming video protocol," [Online]. Available: https://www.intel.com/content/www/us/en/docs/programmable/683397/current/about-the-intel-fpga-streaming-video.html

[6] Texas Instrument, " LMH1983 3G/HD/SD video clock generator with audio clock" [Online]. Available: https://www.ti.com/lit/gpn/LMH1983

[7] Skyworks, " Si516 Dual Frequency General Purpose Voltage Controlled Oscillator (VCXO)" [Online]. Available: https://www.skyworksinc.com/en/products/timing/general-purpose-voltage-controlled-crystal-oscillators-vcxo/si516

[8] "SMPTE ST 2059-1:2021", Generation and Alignment of Interface Signals to the SMPTE Epoch, 2021.

[9] Altera, "CVG Example Designs," [Online]. Available: https://www.intel.com/content/www/us/en/content-details/794825/clocked-video-genlock-cvg-demo-on-agilex-7-fpga-i-series-development-kit-video.html

WP-01336-1.1