Reverse Genlock for Synchronous Tiled Display Walls with Smart Internet Displays

Jochen Miroll, *Student Member, IEEE,* Alexander Löffler, Julian Metzger,
Philipp Slusallek, *Member, IEEE* and Thorsten Herfet, *Senior Member, IEEE*

***Abstract*—When it comes to creating a very large combined screen comprised of multiple individual displays, a so-called tiled display wall, dedicated graphics and synchronization hardware is commonly used for two reasons: firstly, without compromising display resolution neither spatially nor in the time domain, each display needs to be driven by one discrete frame buffer at the full native display resolution. Secondly, synchronization of the individual displays to a common clock is required for smooth and undistorted reproduction of motion spanning multiple displays. One example for synchronization of multiple frame buffers is digital television broadcast. Herein, audio-visual bitstreams are generated including clock information in order to enable receivers to align playback frequency and phase accordingly, which is termed genlock. In this paper we present an architecture for IP-based *reverse genlock*, whereas a display clock signal is used both for inter-display synchronization as well as for generator synchronization. The accuracy of the synchronization enables multi-HD active stereoscopy at 120 Hz across many solely IP-interconnected displays.**

***Index Terms*—genlock; software synchronization; active stereoscopy; display wall; virtual reality;**

I. Introduction

In display walls, specialized hardware is required for the individual displays to jointly display content and hence to form a combined screen due to the fact that display interfaces such as the Digital Visual Interface (DVI) are designed for one-to-one interconnection of a single pixel source and sink. Note that herein, a pixel sink is a *display*, whereas the term does not limit the scope to a specific display technology. With display interfaces, digital video data is transmitted in an uncompressed manner, whereas multi-HD will typically exceed the specified bandwidth limit of a single interconnection easily. On an $n \times 1080$p video wall composed of n individual full-HD displays e.g., displaying multi-HD resolution content would be possible (without compromising high refresh rates). Existing solutions such as *daisy-chaining*, in which one video signal is used to feed a set of displays, achieve a compromise of reduced resolution in either the spatial domain, the time domain, or both. Another common solution is the use of specialized hardware providing multiple display interface ports in order to multiply the available bandwidth (multi-head), whereas the number of ports (heads) is limited. A costly, yet extensible solution is the use of multiple personal computers, each equipped with a high-performance Graphics Processing Unit (GPU) which provides dedicated synchronization features by means of *separate wiring for clock distribution* between GPUs. This approach can be combined with multi-head and is termed *multi-node* display in the following.

The approach presented herein is an alternative to any of the above solutions. We focus on independent, low-cost display nodes that form a multi-node, synchronous composite screen. Accuracy of synchronization is achieved to enable at least 120 Hz time-interleaved (frame-sequential) stereoscopic display in conjunction with active-shutter glasses across the composite screen. Furthermore, video sources are synchronized by feedback of an IP synchronization signal from the display side. Video content is provided to the display side via standard protocols such as MPEG Transport Streams (TS), RTP and H.264. We term an individual display that meets the requirements of our architecture *Smart Internet Display* (SID) and we point out that existing consumer hardware such as current *smart/hybrid/connected TVs* are SIDs in principle. Audio rendition is outside the scope of this work.

In our system, we synchronize content sources and the display side to a display wall clock by *reverse genlock* over Internet protocols (IP).

The genlock paradigm is reversed due to the requirement of synchronizing both display frequency and phase when receiving video streams over IP. Reverse genlock includes clock propagation and round-trip-time measurement. At the content source side, each source will have its local *system time clock*, upon which it determines playout time instants. At the display side, we synchronize *display clocks*. By display clock we denote the refresh rate of the graphics pipeline, including the physical display itself.

II. Synchronization in Imaging Systems

With respect to synchronization in imaging systems, distinction of the terms *genlock, framelock* and *swaplock* is important.

Genlock: describes a mechanism by which one or multiple consumers synchronize to the clock of a single generator. It is common in broadcast, whereas e.g. in digital television an audio-visual bitstream is provided with in-band generator clock signaling. TV sets receiving this stream will decode and playback the audio-visual content at the exact same rate at which it has been generated.

Swaplock: describes a level of synchronization at which the content of any two currently displayed frame buffers are synchronized

Framelock : describes a level of synchronization at which visual refresh is synchronized (e.g. by genlock).

In general, swaplock and framelock are independent of each other. For example, a system could achieve synchronous display refresh (framelock) with the displayed content not being in sync (no swaplock). Framelock is an indispensable requirement for time-interleaved stereoscopic display on a composite screen such as a display wall.

For active stereoscopy, missing framelock will disturb the spatial perception of the composite image. Assume some active stereo shutter glasses are connected and synchronized to one of a set of displays (e.g. via infrared or Bluetooth). Without further measures, left-right stereo separation will suffer on all of the displays except this master. Note that a phase difference of 90° between any display and the master results in 100% crosstalk, while 180° results in left-right permutation. Without continuous frequency and phase equalization, for oscillators that deviate by e.g. 10 ppm in base frequency, 180° phase offset (starting from zero) will occur after just about 7 minutes in a system with refresh rate 120 Hz.

III. Display Synchronization

Any two independently running graphics devices are out of phase in display refresh. Even if parametrized to the same refresh rate

The authors are with Saarland University, Saarbrücken (Germany). This research has received funding from the Intel Visual Computing Institute at Saarland University and the DFKI Saarbrücken.
E-mail: see http://www.intel-vci.uni-saarland.de/en/team
Manuscript received June 30th, 2012

of e.g. 50 Hz, each derives this frequency from its local quartz crystal oscillator (XO) of some base frequency. Due to production tolerances of several ten parts per million (ppm), oscillator frequency temperature and aging dependence, any two free running XOs are highly likely to differ in frequency and phase at some point in time. With oscillator frequencies $f_1(t)$ and $f_2(t)$, frequency and phase difference are given as

$$|(f_1(t) - f_2(t))| = \Delta f(t) = \frac{d}{dt}\frac{1}{2\pi}\Delta\phi(t) = \frac{d}{dt}\frac{1}{2\pi}|\phi_1(t) - \phi_2(t)|$$

whereas $\Delta\phi(t)$, $\phi_1(t)$, $\phi_2(t) \in [0, 2\pi]$ are the phase difference and the phases, respectively.

For synchronization of multiple independently running clocks, as it is the case for display clocks in a display wall setup, two problems need to be solved: Firstly, definition of a wall clock and its propagation and secondly, adaptation of a local clock in frequency and phase to a wall clock.

A. Wall clock propagation

A number of system time clock synchronization mechanisms based on IP networking are available, such as NTP [1] and PTP [2]. Herein, a client may request one or more server(s) to send timestamps via IP packets, while a client itself may be a server to others. Synchronization is based on calculations with four timestamps, which are exchanged in IP packets on a pre-defined network port, potentially with operating system or Ethernet hardware support. Synchronization accuracy is increased by selecting more stable and less distant clock servers.

We use a straightforward approach to solve the first problem: One of the display nodes is elected as the clock master display (CMD) and it continuously broadcasts a message at each visual refresh time instant within display clock reference (DCR) UDP/IP packets. Similar to *adaptive mode* in TDMoIP [3], slave display nodes (SDNs) in our system receiving this stream of messages may assume them to be spaced equidistantly on the time axis of the CMD's display clock.

B. Clock adaptation

Slave display nodes need to determine frequency and phase of the wall clock, as well as of their own display clock. Both may be solved in the same manner: The CMD's exact refresh rate is deduced from arrival timestamps, while the local display refresh rate is deduced from vertical blanking interrupt timestamps, both measured in the operating system kernel space. Furthermore, propagation time of the DCR packets has to be determined. Finally, the local display clock needs to be adjusted without visible artifacts. The process is depicted in figure 1.

Of the above, fine granular adjustments of the display clock have turned out to be the biggest challenge in this work. The granularity has to be low enough for the connected display not to fall out of sync.

C. Hardware Solutions

In the most simple cases of daisy-chaining and multi-head, there is one single XO involved serving as the wall clock. In daisy chaining, a single video signal (e.g. HDMI) is repeated by the displays, possibly introducing some, but negligible delay. In the same way, a multi-head solution is clocked from a single XO on a multi-head GPU for example. Multi-node hardware solutions[1], i.e. for frame locking of several independent systems (also multi-node with multi-head), typically require the systems to be connected by dedicated wiring,

[1]Such as NVIDIA Quadro G-Sync and ATI FirePro S400, to name a few

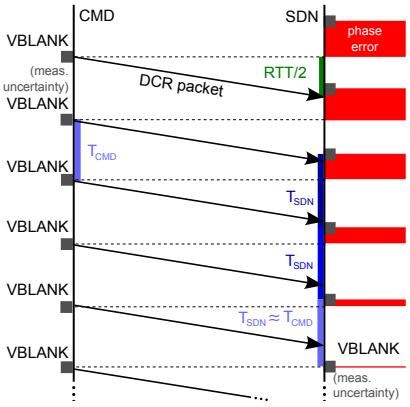

Figure 1. Slave Display Node (or identically: source) frequency and phase synchronization in the ideal case.

with the synchronization signal fed from the master to a number of slave display systems.

In summary, a common clock (wall clock) must be present in any display-wall setup. Figure 2 shows the clock distribution schematically for the aforementioned solutions.

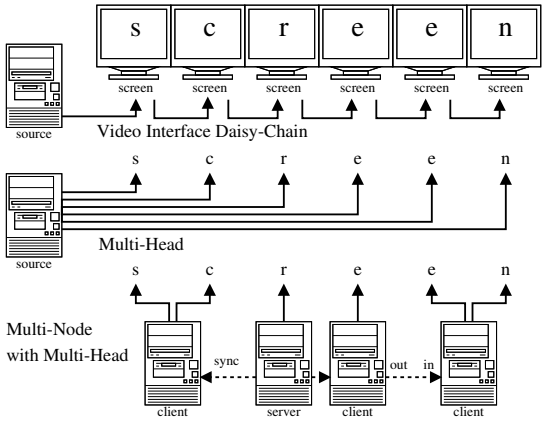

Figure 2. Hardware wall clock solutions differing in complexity and video bandwidth. *Top:* Daisy-Chain; *Middle:* multi-head; *Bottom:* multi-node with multi-head

IV. Modulation of Frame Duration

In the case of personal computers, upon reception of image data, the display refreshes the currently shown image with this data. Although physical creation of the visible image is completely different for DVI and HDMI, the format of the pixel data still closely follows VGA [4] and is specified by the Video Electronics Standards Association (VESA) by the general timing formula (GTF).

Figure 3 depicts signal layout, as signaled on digital interfaces DVI and HDMI. Blanking is a requirement specific to the display device; hence on DVI and HDMI, reduced blanking intervals are an option to reduce bandwidth overhead.

Consequently the modulation of the frame duration may increase or decrease the current refresh rate. In the following, we evaluate two methods to do so.

A. Frame Length

Due to the similarity of analog VGA signaling and the digital variants DVI and HDMI, we evaluated the possibility of adjusting the refresh rate of commodity PC graphics hardware [5] by modulation

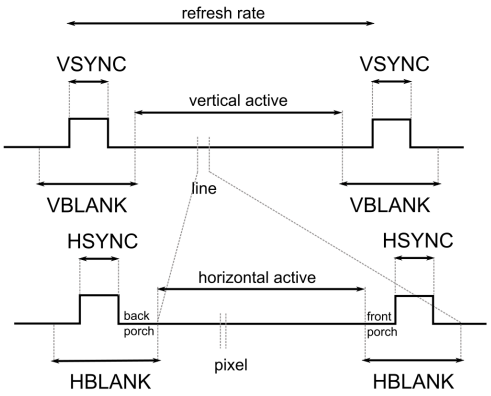

Figure 3. High level structure of signaling on a VGA, DVI and HDMI cable with horizontal and vertical synchronization

Table I
EXAMPLE VESA CVT PARAMETERS FOR 720P AT 120 HZ

pixelclock f_p	active (horizontal)	blanking (horizontal)		
		front	SYNC	back
162 MHz	1280	96	136	232
refresh rate r_v	active (vertical)	blanking (vertical)		
		front	SYNC	back
119.86 Hz	720	3	5	47

of frame duration. The video refresh rate is $r_v = f_p/p_{tot}$ with f_p being the pixel clock frequency, divided by the total number of pixels. Consequently, the frame duration is $T_{frame} = 1/r_v$.

With fixed f_p, changing the number of total pixels modifies the frame length (in pixels) and its duration. We have been able to modify the number of non-active pixel lines and columns within the VSYNC interval. The change in refresh rate is given as

$$\Delta r_v = f_p \cdot \left(\frac{1}{h_{tot_1} \cdot v_{tot_1}} - \frac{1}{h_{tot_2} \cdot h_{tot_2}} \right) \quad (1)$$

With the image parameters as given in table I one obtains from (1), within a reasonable refresh rate range, a granularity of refresh rate variation while changing only the number of lines by $\Delta r_v \approx 0.16Hz$, and while changing only the number of columns by $\Delta r_v \approx 0.06Hz$. Combining horizontal increase with vertical decrease and vice versa results in a granularity of about $0.015 - 0.02Hz$.

B. Pixelclock

From eq. (1) we see that also a modification of f_p, the pixelclock, results in a different frame duration. The pixelclock is typically produced by a frequency synthesizer on the graphics device. It is generated by applying multipliers and divisors to a reference frequency that is provided by a quartz oscillator. Eq. (2) gives an example [5]:

$$f_p = f_{ref} \frac{(5 \cdot (M_1 + 2) + (M_2 + 2))}{(N + 2) \cdot (P_1 \cdot P_2)} \quad (2)$$

whereas f_{ref} is a reference XO frequency (e.g. 96 MHz for some Intel devices). $M_{1,2}$, N and $P_{1,2}$ are integer parameters that can be adjusted within predetermined limits. Without going into detail of parameter selection and validation of frequency synthesizers, the granularity of refresh rate modification using the pixelclock has been around $0.14 - 0.36Hz$ for the parameters in table I.

It can be seen that with access to f_p, some combination of parameters may enable even more fine grained frame length modulation.

Above presented approaches have been chosen using VGA by Allard et al. [6] with analog monitors and by Waschbüsch et al. [7] with digital monitors.

We have been able to modify the image dimensions (within any part of the total pixel area) as well as the pixelclock within the VBLANK pause using Intel integrated graphics platforms without resetting the whole display pipeline using Intel Linux graphics drivers [5]. But none of the tested displays tolerated ongoing modifications of image parameters without clearly visible artifacts.

Hence, we deem the modulation of frame duration by the above mentioned methods unusable for synchronization of display walls, as it seems that tolerance highly depends on the specific display hardware.

V. (FORWARD) GENLOCK IN MPEG SYSTEMS

Audio-visual coding and transmission according to Moving Pictures Experts Group (MPEG) standards are used in many systems such as digital television (DTV). DTV sets are required to playback audio-visual content at the exact same rate at which it is received. Therefore, the receivers' clocks must be synchronized to that of the broadcast station, which is accomplished by a (forward) genlock mechanism. By some phase-locked loop (PLL) implementation, a DTV receiver constantly monitors and adapts its frequency and phase compared to a reference clock signal.

DTV broadcast according to the Advanced Television Systems Committee (ATSC) and Digital Video Broadcasting (DVB) standards is transmitted as an MPEG Transport Stream (TS) multiplex. Genlock is implemented by including reference timestamps at an interval of less than $100ms$ and less than $\pm 500ns$ of timing jitter into an MPEG TS. They comprise the so-called program clock reference (PCR). Each PCR timestamp represents a clock tick counter value of a clock running at 27 MHz. The tick counter value is taken during TS multiplex at the point in time of timestamp insertion. PCR timestamps are encoded as a 90 kHz and a 27 MHz portion. ISO/IEC 13818-1 describes the insertion of timestamps into an MPEG TS. Any receiving devices (TVs, set-top-boxes) likewise deduce their local system frequency from a 27 MHz quartz, driving both audio and video outputs. Thus, the 27 MHz portion of the PCR timestamps may be compared directly after compensation for some offset due to non-identical start values. In a typical digital TV set-top-box (STB) design, frequency and phase is estimated by use of a PLL implementation in either hardware or software, while the output of the PLL is used to adjust a voltage controlled crystal oscillator (VCXO) circuit.

Additionally presentation as well as decoding timestamps (PTS, DTS) are provided for audio and video frames within their respective packetized elementary stream (PES) headers at a resolution of 90 kHz and relative to the PCR. On a 90 kHz resolution clock, e.g. 3003 ticks elapse per frame for an approximate 29.97 Hz video frame rate (NTSC TV system), while 3600 elapse for 25 Hz (PAL TV system).

VI. SMART INTERNET DISPLAYS

In our system, entities are solely IP-interconnected. A display entity is composed of a single physical display device as well as some accelerated processing unit (APU) with the following capabilities: decoding and display of (multi-view) video content, IP connectivity and adjustability of the visual refresh rate. Currently available television sets are so-called *smart TVs* that fit nicely into this concept, satisfying the basic requirements. We call display hardware that meets these genlock requirements *Smart Internet Displays* (SIDs). The concept of an SID can also be abstracted by a regular display and a connected IP-enabled video source, such as a digital TV set-top-box.

A. Digital Television Set-Top-Boxes

In our system we use VCXO features of digital TV set-top-boxes[2] for frequency adjustments in the order of ± 150 ppm [8], whereas the crystal oscillator quality requirements are ± 30 ppm according to DVB [9],[10]. In case display refresh rate adjustment at this granularity is available at the display nodes, a software phase-locked loop (SPLL) implementation may be used for display clock frequency and phase synchronization. Herein we assume the display to be slave to the input HDMI/DVI video signal, whereas frequency and phase changes of the video signal are followed by the display.

B. Digital Television on Personal Computers

Contrary to dedicated DTV platforms, personal computer (PC) clock frequencies typically are not derived from a single 27 MHz XO. Different components such as CPU, GPU, sound card and memory may even use different base frequencies. Non CPU-integrated graphics cards may use 25 MHz, as PCI-express bus communication is typically clocked at 100 MHz. Sound cards and digital audio interfaces on the other hand typically employ 24.576 MHz crystals for integer multiples of 48 kHz sampling frequencies (or 22.579 MHz for 44.1 kHz).

VII. Reverse Genlock

In our system, we reverse the roles of generator and consumer with respect to synchronization: The display side provides a clock signal to the content generating side. This setup enables feeding the display wall with different, independently generated streams.

As stated above, we synchronize display refresh rates and times across displays (framelock) using an IP-based clock signal. Using the same signal, sources are synchronized. Virtual timestamps are further provided within the clock signal packets. With sources providing timestamped IP video streams to the display side, a closed loop that enables swaplock is created. Phase is estimated using ICMP [11] *ping*, which is assumed available at any IP host and supported throughout the Internet.

A. Architecture

For the reasons motivated in section III, we have chosen to implement a reverse genlock architecture in which continuous timing information is provided via IP networking. The architecture consists of three types of entities:

- one clock master display (CMD)
- slave display nodes (SDN)
- reverse genlock frame deadline predictor

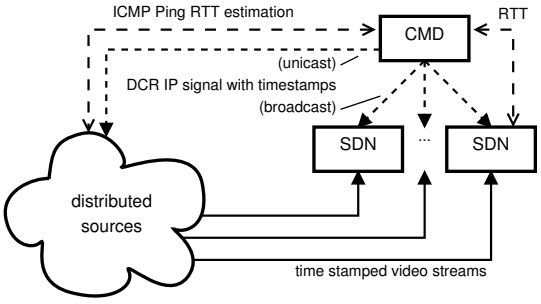

Figure 4. Example reverse genlock scenario. SDNs and sources are synchronized to a wall clock provided by one of the display nodes.

[2]Intel CE4100: http://intelconsumerelectronics.com/

The reverse genlock architecture is depicted in figure 4. One of the displays serves as the clock master by providing the wall clock reference for the composite display. A reference time base is directly derived from the CMD's digital display signal and is provided to other entities via short IP packets, which we term display clock reference (DCR) packets. DCR packets are generated periodically with period larger than (at integer multiples) or equal to the CMD's visual refresh period. Due to the assumption of equidistance with those packets, timing jitter as introduced by the network can be compensated, and the visual refresh rate of the CMD can be determined. This may be supported by IEEE 1588 hardware timestamps.

Non-clock-master displays (slave display nodes, SDNs) measure their own visual refresh rate and compare it with the reference clock received via IP, whereas packet loss is compensated for. Refresh rate equalization is done by VCXO control (genlock) based on the DCR signal. Phase differences are compensated for by round trip time (RTT) estimation using the standard IP RTT measurement method based on ICMP [11] *ping*.

In addition, the DCR packets may also be received by sources that shall synchronize content generation with the composite display, which is the motivation for the term *reverse genlock*. In case of forwarded DCR packets, the current cumulative RTT for each forwarding hop is signaled in each DCR packet. The packet structure is depicted in figure 5.

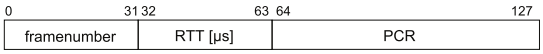

Figure 5. Frame structure of periodic display clock reference (DCR) frames.

Any participating pixel source can synchronize content generation rate (or playout rate for stored content) to the composite display refresh rate. In order to do so, a frame deadline predictor provides linear extrapolation of the point in time at which a frame with some number $b > a$, whereas a is the last displayed frame, is due for transmission. By this, a joint playout phase can be achieved across multiple content-generating nodes, which may be subject to a constant phase offset due to content generation management. Furthermore, audio-visual frames must be timestamped to achieve swaplock. Therefore, a broadcast-compatible PCR timestamp is included in the DCR packets, see figure 5. This creates a closed loop, as each display can compare the PCR received via the DCR directly (current time of the master) with PCR values as seen from incoming audio-visual bitstream(s), and consequently adjust a constant delay. If this constant delay is chosen identically on all display devices, swaplock is achieved.

VIII. Results

We implemented a phase-locked Loop for VBLANK and DCR frequency and phase estimation in software on i686 32-bit architecture. The PLL is depicted in figure 6. For an in-depth discussion

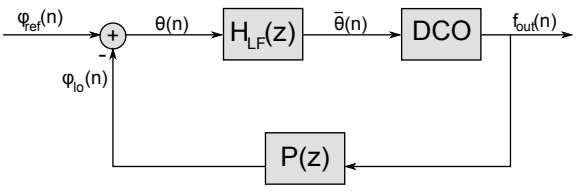

Figure 6. Phase-Locked-Loop

on PLL design, the reader is referred to the literature [12]. Our implementation compensates for undetected VBLANK interrupts and lost DCR packets by linear interpolation. The PLL loop filter is of first

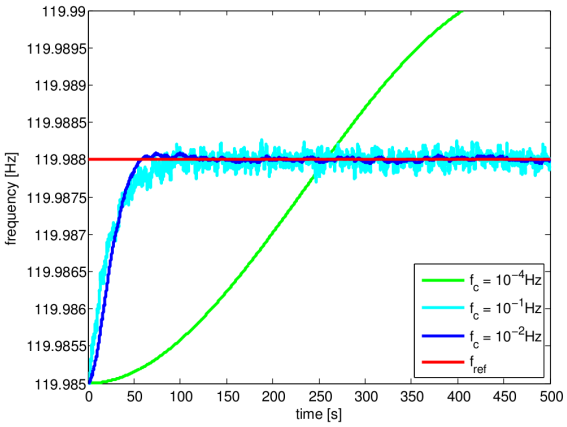

Figure 7. PLL simulation with loop filter of first order

order with cutoff frequency 10^{-2} Hz. Figure 7 shows a simulation of the loop filter.

The used PLL takes roughly below 75 s to acquire lock, i.e. a display-wall is powered up and running after less than two minutes, including boot time. We achieve software-measured residual phase error of around $\pm 100\,\mu$s with the used set-top-boxes and long term frequency stability. Absolute phase offset due to ICMP LAN round-trip time estimation error is assumed negligible [13] with proper filtering. Said magnitude of phase error corresponds to only $\pm 1.2\%$ of T_{frame} at 120 Hz, which we deem sufficient for good time-interleaved stereo separation. Note that left/right discrimination is provided by HDMI 1.4a [14].

For determining the frequency and phase stability of the system synchronized via an IP network, we perform two tests. During the tests, the VSYNC pulses of the video signal are extracted from the DVI/HDMI signal and measured using an oscilloscope. Tests are carried out with two set-top-boxes connected to a gigabit Ethernet. Via this Ethernet, both receive a single DCR signal. In the first test, there is no background traffic on the network. An exemplary but consistent result is presented in figure 8. Herein, the phase difference

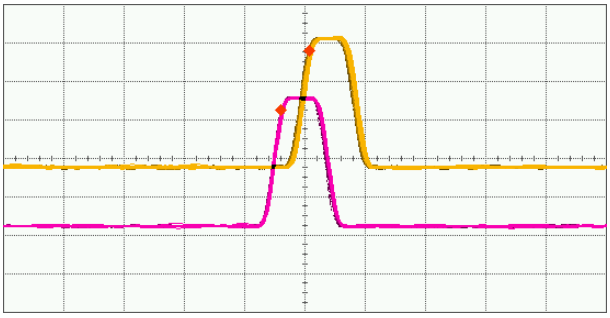

Figure 8. Oscilloscope measurement of VSYNC pulses on DVI/HDMI cables connected to a SDN each. SPLLs are locked. No background traffic. Scale: 50μs/div. Mean phase difference $\left|\Delta\bar{T}(\diamond)\right| = 29.5\mu$s with std. dev. 6.7μs.

is below 30μs on average.

The second test is performed while transmitting 100 Mbps UDP streams destined to each of the set-top-boxes IPs from another gigabit Ethernet host in the network. This test is important since the network cannot be assumed idle in practice due to the visual content being streamed via the same network also. Note that the background traffic is initiated after the PLLs were "locked" over an idle Ethernet. Due to the Ethernet backbone being a gigabit switch and the set-top-boxes being equipped with 100 Mbps Ethernet, the links to the set-top-boxes are assumed to be saturated. Again, an exemplary but consistent result is presented in figure 9. In this test, the phase difference

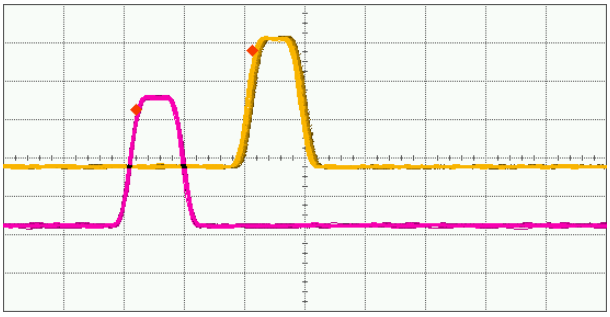

Figure 9. Oscilloscope measurement of VSYNC pulses on DVI/HDMI cables connected to a SDN each. SPLLs are locked.Very high background traffic. Scale: 50μs/div. Mean phase difference $\left|\Delta\bar{T}(\diamond)\right| = 105.8\mu$s with std. dev. 20.6μs.

increases to slightly above 100 μs on average with increased standard deviation. In both tests, RTT estimation has been switched off, thus the results validate the use of the above described PLL. Note that in figures 9 and 8, the grid position relative to the oscilloscope trigger, which is connected to the DCR source (master clock signal) is unchanged.

IX. Conclusion

Very high resolution stereoscopic visualization on a tiled display wall is possible even with time-interleaved active shutter technology. Smart Internet Displays as defined in this paper are a requirement to build such display walls and consumer hardware is available. Providing both synchronization and content via IP networking is a scalable and cost-effective solution.

References

[1] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification." RFC 5905 (Proposed Standard), June 2010.

[2] P. Ohly, D. Lombard, and K. Stanton, "Hardware Assisted Precision Time Protocol. Design and case study.," in *Proceedings of LCI International Conference on High-Performance Clustered Computing. Urbana, IL, USA: Linux Cluster Institute*, vol. 5, pp. 121–131, 2008.

[3] Y. Stein, R. Shashoua, R. Insler, and M. Anavi, "Time Division Multiplexing over IP (TDMoIP)." RFC 5087 (Informational), Dec. 2007.

[4] Digital Display Working Group Promoters, "Digital Visual Interface – DVI, Revision 1.0," 1999.

[5] Linux Graphics Drivers from Intel, March 2012.

[6] J. Allard, V. Gouranton, G. Lamarque, E. Melin, and B. Raffin, "SoftGenLock: active stereo and genlock for PC cluster," in *Proceedings of the workshop on Virtual environments*, p. 260, ACM, 2003.

[7] M. Waschbüsch, D. Cotting, M. Duller, and M. Gross, "WinSGL: software genlocking for cost-effective display synchronization under Microsoft Windows," *Proceedings of the Sixth Eurographics Symposium on Parallel Graphics and Visualization*, 2006.

[8] AKM AK8136a Data Sheet, March 2012.

[9] ETSI TR 101 290 V.1.2.1, "Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems," 2001.

[10] ISO 13818-1, "Information technology – Generic coding of moving pictures and associated audio information: Systems," 2000.

[11] J. Postel, "Internet Control Message Protocol." RFC 792 (Standard), Sept. 1981. Updated by RFCs 950, 4884.

[12] R. Best, *Phase-Locked Loops: Design, Simulation, and Applications*. McGraw-Hill professional engineering, McGraw-Hill, 2007.

[13] A. Günther and C. Hoene, "Measuring round trip times to determine the distance between WLAN nodes," *In Proc. of 4th Int. IFIP-TC6 Networking Conference*, pp. 768–779, 2005.

[14] Hitachi Ltd, Panasonic Corporation, Philips Consumer Electronics International, B.V., Silicon Image, Inc., Sony Corporation, Technicolor, S.A. and Toshiba Corporation, "High-Definition Multimedia Interface Specification – Version 1.4a," 2010.