Network Time Synchronization for Detector Data Acquisition Electronics

Wolfgang Hennig, *Member, IEEE*

***Abstract*–Large scale nuclear physics experiments often use arrays of radiation detectors, which can be physically separated, for example in separate rooms or along a beamline. Time synchronization of the detector readout electronics, which is essential to detect related events, is often accomplished by sharing clock, trigger and reset signals within and between racks of digitizing electronics. This works well over short distances, but requires dedicated cabling and/or modules and becomes cumbersome for widely separated arrays. As the detector readout electronics is in many cases operated by computers linked over standard data networks, an alternative to dedicated clock distribution trees is the synchronization of clocks over the network. Network time synchronization protocols like IEEE 1588 or White Rabbit are reported to achieve low/sub nanosecond timing resolution, but these results generally refer to clock offsets computed in software or to once-per-second reference pulses, not to the capture of detector pulses with amplitudes and waveforms.**

The work reported here thus focuses on integration of network time synchronization techniques with digital data acquisition for radiation detectors. Detector waveforms are captured with analog to digital converters that are synchronized to the network master clock and/or are tagged with time stamps related to the network time. Key to this integration is the handling of network clock and time in the same FPGA firmware that processes the analog to digital converter's data stream. We tested implementations based on FPGA firmware and on network hardware, and report timing resolutions obtained for two synchronized detectors capturing coincident radiation.

I. INTRODUCTION

TRADITIONALLY, time synchronization between multiple channels of digital data acquisition is accomplished by sharing clocks, clock reset signals, and triggers, which can become quite complex [1]. For coincident detector signals, such as the 1.17 MeV and 1.33 MeV gammas from a ^{60}Co source detected in two fast scintillators, the variation in measured time-of-arrival difference ΔT (time resolution) can be a few hundred picoseconds full width at half maximum (FWHM) for digitization rates of 100-500 MSPS [2]. The time resolution can approach less than 10 ps FWHM for idealized signals from a pulser [2], [3], [4].

As the detector readout electronics are operated by computers linked over standard data networks, an alternative to dedicated clock distribution trees is the synchronization of clocks over the network, and an alternative to dedicated trigger connections is the processing of time stamped event data by a central computer issuing accept or reject decisions to the whole system. A current standard in network time synchronization is the IEEE 1588 precision time protocol (PTP). It has been implemented on several Ethernet controllers and PHY devices (e.g. Texas Instrument's DP83640 [5]), Xilinx' Zynq processor [6], and many other devices. Precisions are reported to reach the low/sub nanosecond range, depending on the implementation. However, techniques like PTP are primarily designed to synchronize clocks for processors, not for real time processing in field programmable gate arrays (FPGA) or application specific integrated circuits (ASIC). Clocks are synchronized on the nanosecond level in internal counters, but the processor can access these counters only with software limited latencies. In addition, detector waveform data, usually digitized by analog to digital converters (ADC) and captured by FPGA firmware, are not available to the processor in real time. Even if a processor could respond "immediately" to one data event, a second event can follow closer than the readout time (especially when one processor serves multiple channels. Therefore, it is desirable to integrate the PTP synchronization with the data capture and processing of digitized detector signals in the FPGA.

II. EXPERIMENTAL SETUP

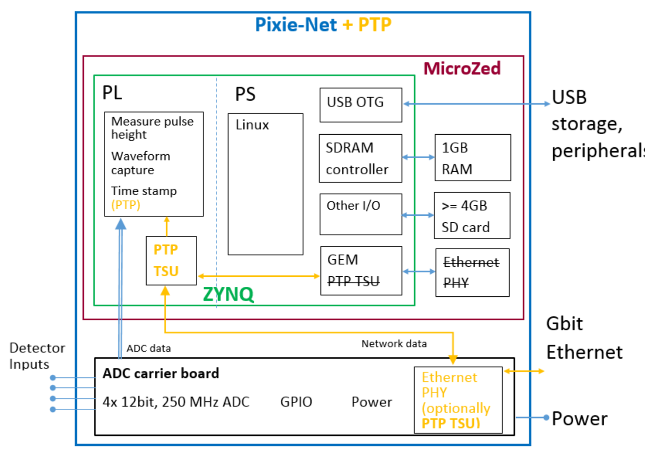

Fig. 1. Pixie-Net block diagram

We approach this integration in two ways: 1) Derive from the PTP circuitry a high speed clock that can be used to clock the ADCs digitizing the detector signal. This clock is synchronized to the network grand master clock and so all ADCs in a network effectively run on the same clock, but without physical clock connections. 2) Derive from the PTP process a timestamp with which a detector event is tagged. The timestamp is related to the network grand master time with nanosecond precision. Either

Manuscript received November 10, 2017. This work was supported in part by the U.S. Department of Energy under Grant No. DE-SC0017223.

W. Hennig is with XIA LLC, Hayward, CA 94544 USA (whennig@xia.com). 978-1-5386-2282-7/17/$31.00 ©2017 IEEE

978-1-5386-2282-7/17/$31.00 ©2017 IEEE

approach would result in synchronized detector pulses, though in 1) an additional signal or process is required to define time zero. For implementation, we chose the Pixie-Net spectrometer module [7]. Its block diagram is shown in Fig. 1. It features a Xilinx Zynq system-on-chip, combining an ARM processor (PS) running Linux, with an FPGA (PL) containing the detector pulse processing. For this project, the Zynq Ethernet interface is redirected to the DP83460 Ethernet PHY (approach 1) or a PTP time stamping core from SoC-e [8] is added to the PL (approach 2). The PTP process is controlled by the LinuxPTP [9] utility in the PS. The DP83460 Ethernet can also be operated in synchronous Ethernet (sync-E) mode. The Zynq's built-in PTP functions unfortunately make no outputs available to the FPGA and therefore are not used in this work.

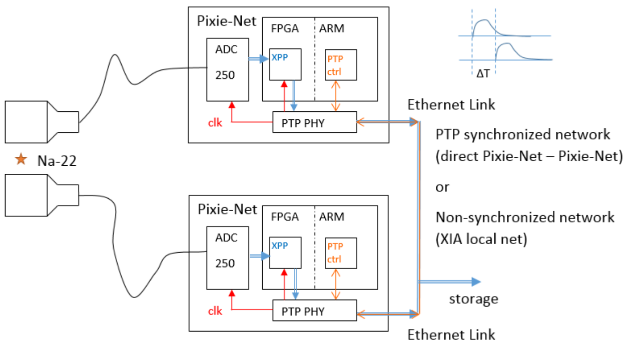

Fig. 2: Experimental setup.

Timing measurements were performed as in [2], with two $LaBr_3$ detectors read out by two Pixie-Net modules synchronized over the network (Fig. 2). The time difference ΔT between coincident gamma rays from a ^{22}Na source was measured using time stamps or constant fraction algorithms. A pulser was used in place of the scintillator pair in cases where the detector limited the precision. Two channels of a Pixie-4e module [2], [10] operating on a shared clock were connected in parallel as a reference in some of the measurements. The network was either *non-synchronized* (the standard local network), or *PTP synchronized* in which every node runs on a synchronized clock (here implemented as connecting the two Pixie-Net modules back to back). Table I summarizes the synchronization techniques explored.

TABLE I. SYNCHRONIZATION TECHNIQUES

Technique	Legend	Synchronization via
PTP FPGA timestamping	SoC-e	PTP adjusted fractional time stamp increments [8]
PTP PHY timestamping	PHY	PTP adjusted clock frequency using DP83640 [5]
PTP PHY timestamping with sync-E	PHY+Sync-E	Sync-E clock frequency adjusted by PTP
Sync-E only	Sync-E	Sync-E fixed clock frequency
White Rabbit	WR	WR adjusted clock frequency [11], [12]
Shared Clock	shared clk	Distributed fixed clock frequency

III. RESULTS

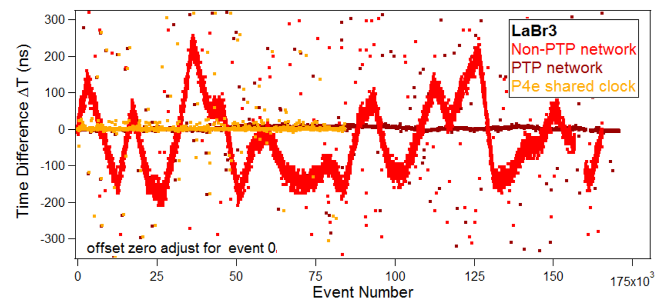

Fig. 3: Measured time difference ΔT for coincident pulses using DP83640 time stamping.

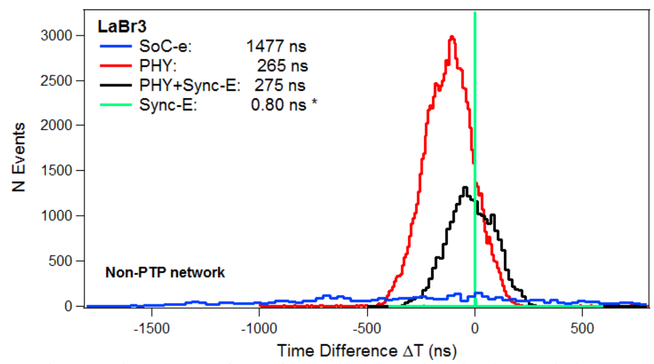

Fig. 4: Histograms of ΔT for various synchronization techniques, using $LaBr_3$ with non-PTP network. The SoC-e distribution extends beyond the range of the plot

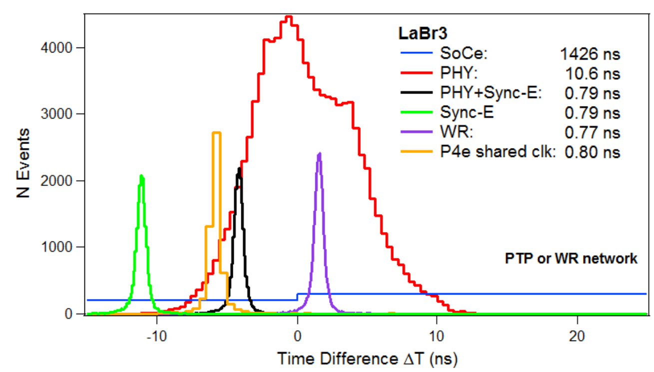

Fig. 5: Histograms of ΔT for various synchronization techniques, using $LaBr_3$ with PTP network.

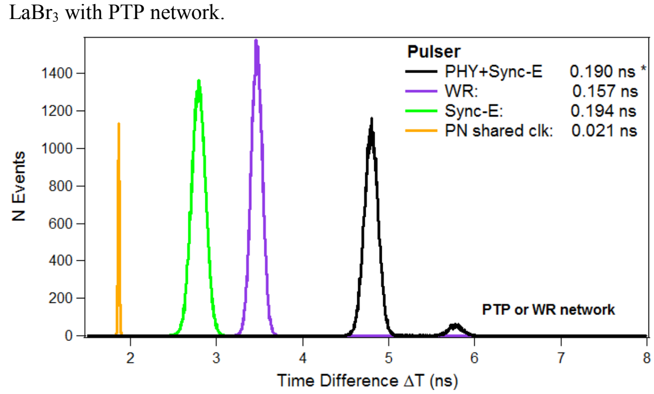

Fig. 6: Histograms of ΔT for various synchronization techniques, using pulser with PTP network. The PHY+Sync-E distribution shows a secondary peak from the initialization phase of the measurement.

Fig. 3 shows the measured ΔT distributions for tests with the DP83640 PTP time stamping. True coincidences fall in a tight distribution, unrelated events close in time form a random background. The non-PTP network measurement clearly shows how ΔT drifts according to small clock frequency differences as the PTP software periodically adjusts the clock. This drift is of lower magnitude in the PTP network measurement, and absent in the Pixie-4e shared clock measurement.

Histogramming hundreds of thousands of timing measurements, we obtain a distribution of ΔT around an average value (Fig. 4-6). The FWHM of a Gaussian fit to these distributions is our primary measure of performance for the synchronization techniques (values shown in plot legends).

IV. Summary and Conclusions

As is to be expected, resolutions with the PTP network are better than with the non-PTP network, and better again with sync-E or WR, by an order of magnitude. Hardware timestamping is better than firmware timestamping, though the SoC-e performs much worse than advertised and may not have been set up correctly in all aspects.

Thus the two most promising network time synchronization techniques for nuclear physics applications were WR and PTP PHY timestamping with the DP83640. The DP83640 obtained time resolutions of ~10 (~300) ns in a (non-)PTP network. In sync-E mode, it reached ~800 ps with a pair of $LaBr_3$ detectors and ~200 ps with a pulser. With the WR demo kit, resolutions were ~800 ps with a pair of $LaBr_3$ detectors and ~190 ps with a pulser. A traditional distributed clock reached ~800 ps with a pair of $LaBr_3$ detectors and ~20 ps with a pulser.

The DP83640 is therefore a good solution for applications that do not need the most precise timing (e.g. coincidence background rejection); it is a ~$10 part that can easily be integrated into the DAQ electronics and does not necessarily need a special network infrastructure. For more demanding applications, such as detector array event building, the DP83640 can be used in a PTP enabled network (which adds cost to the network infrastructure, but not the DAQ electronics). For highest precision applications such as time-of-flight measurements (with detectors better suited for timing than the $LaBr_3$ used here), White Rabbit or synchronous Ethernet can be used, but traditional shared clock solutions still provide better performance which may be necessary for some applications.

This precision network time synchronization allows a new kind of data acquisition, with software based trigger and recording decisions rather than hard wired logic.

References

[1] W. Hennig, H. Tan, M. Walby, P. Grudberg, A. Fallu-Labruyere, W. K. Warburton, C. Vaman, K. Starosta, D. Miller "Clock and Trigger Synchronization between Several Chassis of Digital Data Acquisition Modules", NIM B 261 (2007) 1000–1004

[2] W. Hennig, S. Asztalos, D. Breus, K. Sabourov, W.K. Warburton, "Development of 500 MHz Multi-Channel Readout Electronics for Fast Radiation Detectors", IEEE Trans. Nucl. Sci, Vol. 57, No. 4, August 2010, p. 2365-2370

[3] W. K. Warburton, W. Hennig, "New Algorithms For Improved Digital Pulse Arrival Timing With Sub-GSps ADCs", to be published in IEEE Trans. Nucl. Sci; DOI: 10.1109/TNS.2017.2766074

[4] C. Hellesen, M. Skiba, G. Ericsson, E. Andersson Sundén, F. Binda, S. Conroy, J. Eriksson, M. Weiszflog, "Impact of digitization for timing and pulse shape analysis of scintillator detector signals", NIM A 720 (2013), 135-140

[5] www.ti.com/product/DP83640/technicaldocuments

[6] "Zynq-7000 AP SoC - Precision Timing with IEEE1588 v2 Protocol", www.xilinx.com

[7] www.xia.com/Pixie-Net.html

[8] http://soc-e.com/ieee-1588v2-ordinary-and-boundary-clock/

[9] http://linuxptp.sourceforge.net

[10] W. Hennig, Y. X. Chu, H. Tan, A. Fallu-Labruyere, W. K. Warburton and R. Grzywacz, "The DGF Pixie-4 spectrometer – Compact Digital Readout Electronics for HPGe Clover Detectors", NIM B, 263 (2007) 175-178

[11] http://sevensols.com/index.php/products/wr-len/

[12] https://www.ohwr.org/projects/white-rabbit/wiki