Distributed digital data acquisition system with network time synchronization

Wolfgang Hennig & Shawn Hoover

(Presented by Bill Warburton)

XIA LLC, 31057 Genstar Rd, Hayward, CA

www.xia.com

Supported by DOE grant DE-SC0017223

Contents

Distributed digital data acquisition system with network time synchronization

Motivation

SBIR Solicitation

"*Software-Driven Network Architectures for Data Acquisition***"**

- Design a distributed DAQ system for radiation detectors
- Eliminate clock and trigger distribution cabling and synchronize DAQ units via high speed data network

XIA SBIR project

"*Distributed digital data acquisition system with network time synchronization"*

Phase I: 2017-2018, Phase II: 2018 – 2020

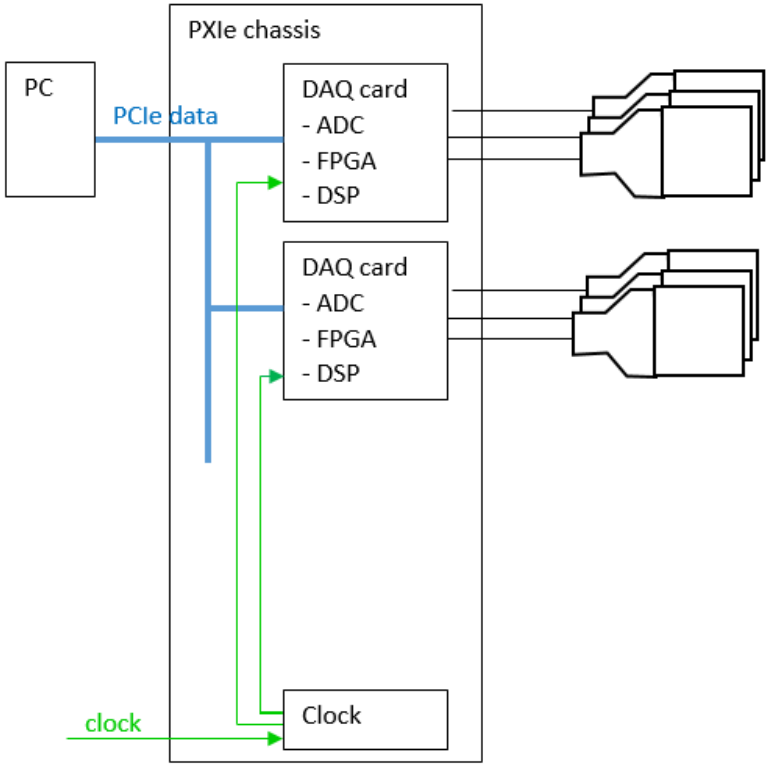
⇒ Adapt existing techniques to XIA's detector readout DAQ modules

⇒ Stay within standards, use open HW/SW environment, no "black box" for purchase

⇒ Requirements for timing precision depends on experiment

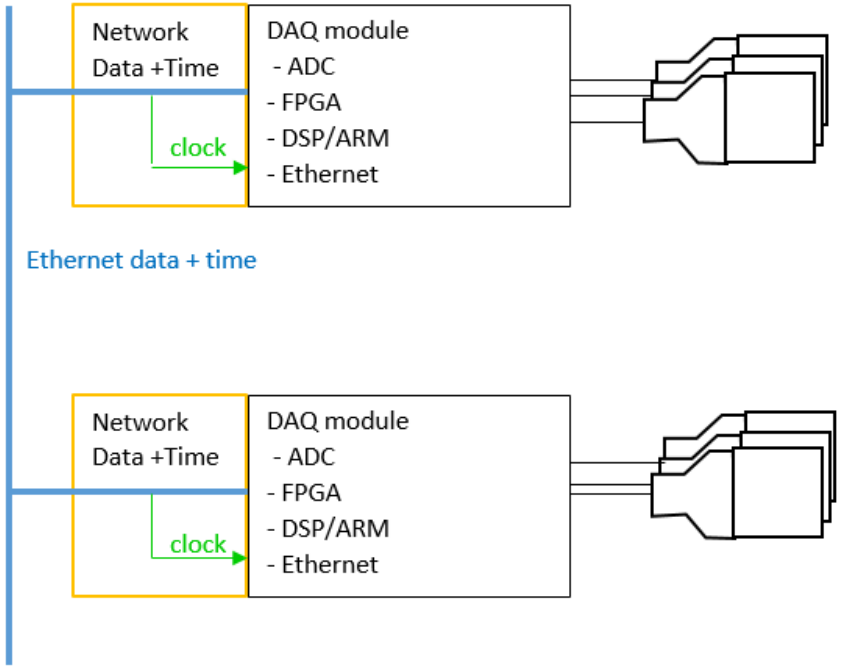
- Background reduction by coincidence: Hundreds of nanoseconds
- Event building for detector arrays: Tens of nanoseconds
- Time of flight measurements: Sub-nanosecond
- Goal in SBIR topics: "10 ns"; "1 ns or better"

Detector Readout Electronics Synchronized Through Ethernet Network Timing Techniques



Traditional

Crate with data I/O to host PC and local clock distribution



This Project

Independent modules with network data and derived clock

Distributed digital data acquisition system with network time synchronization

Existing Technologies

- **Shared Clock Signal**
  Works, but requires dedicated cabling. Not ideal for distributed DAQ systems.
  Reported time resolution: sub-nanosecond, even tens of picoseconds

- **IEEE 1588 Precision Time Protocol (PTP)**
  Processors exchange synchronization messages over network to measure delays
  PTP Time Stamping Units (TSU) built into several commercial Ethernet MACs, Ethernet physical layers (PHY). Commercial PTP switches available ($$).
  Open source software for managing time synchronization (LinuxPTP, ptpd)
  Reported time resolutions: milliseconds (software TSU)
  low nanoseconds (hardware TSU)

- **Synchronous Ethernet (SyncE)**
  Extract clock from Ethernet link and use for local processing

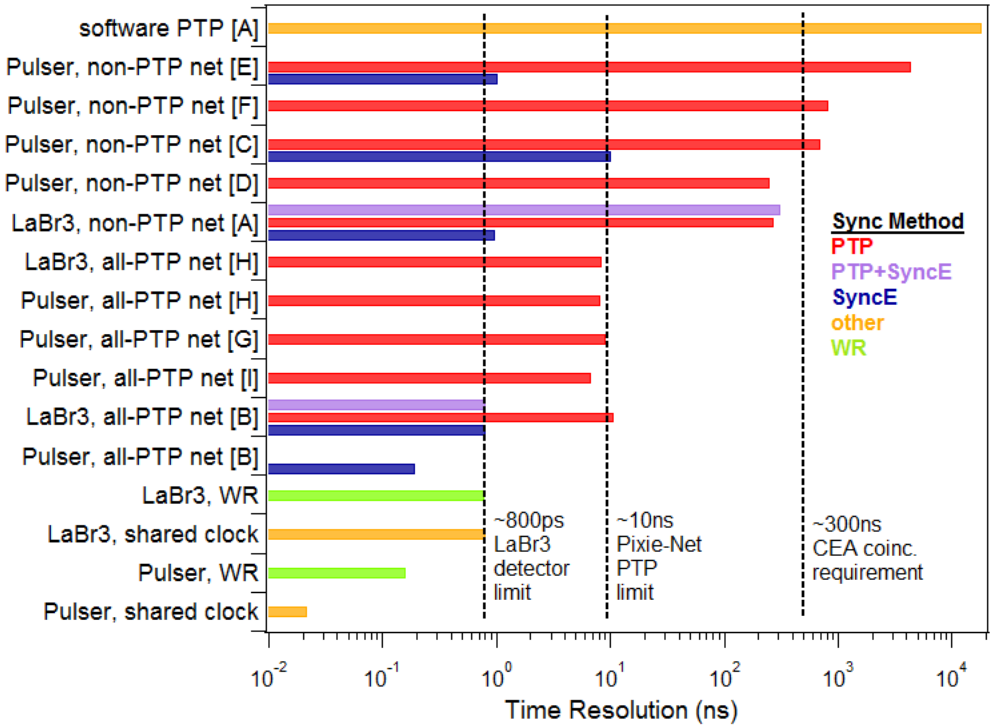
- **CERN's White Rabbit (WR)**
  Extension of PTP standard with synchronous Ethernet
  Open hardware project. WR network switch commercially available.
  WR modules developed by scientists
  Reported time resolution: sub-nanosecond, even tens of picoseconds

Distributed digital data acquisition system with network time synchronization

Phase I Results: Pixie-Net with PTP and SyncE

- Added a SyncE/PTP Ethernet PHY to recently developed Pixie-Net DAQ module
- Obtained timing resolutions of
  ~ 10ns FWHM with PTP
  ~ 800ps FWHM with SyncE using LaBr3 detector
  ~ 200ps FWHM with SyncE using pulser

Pixie-Net

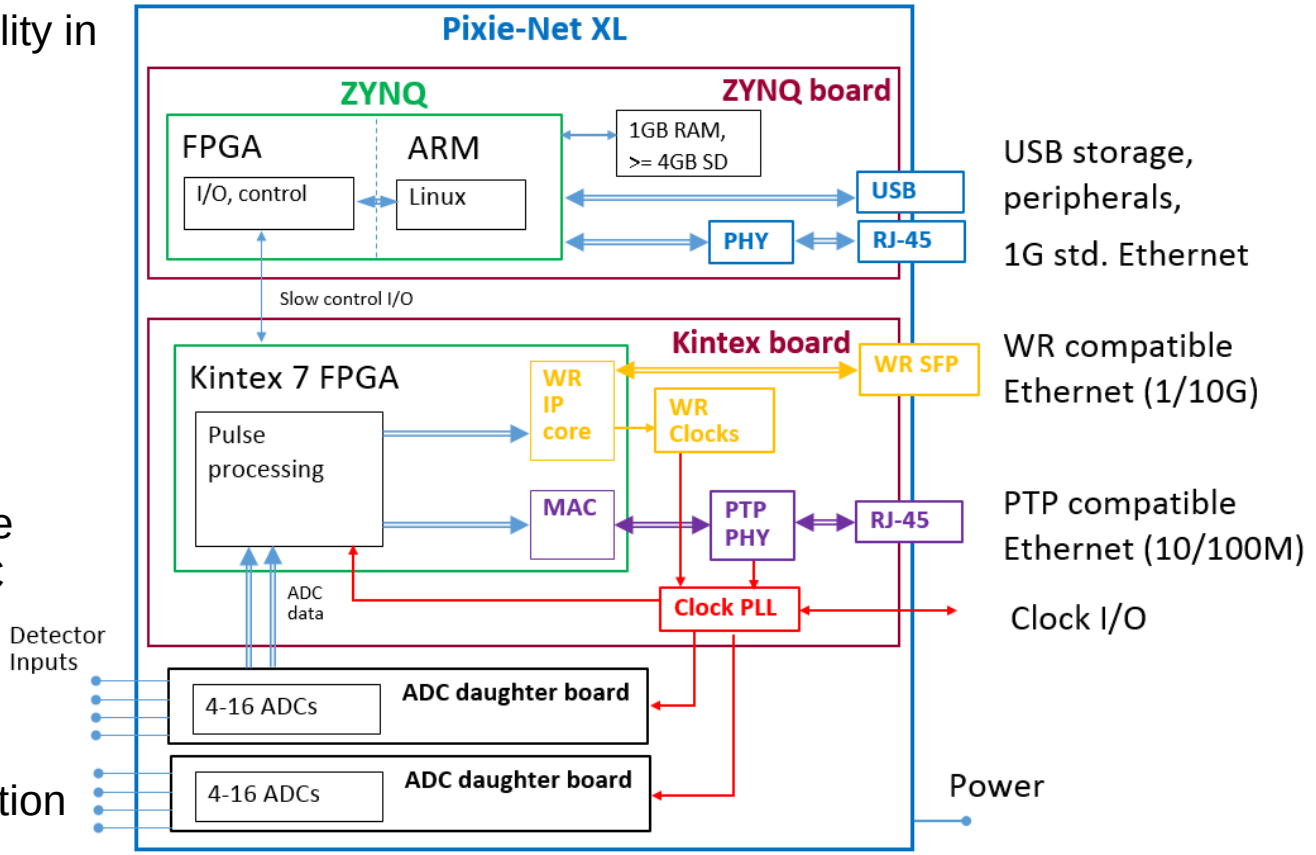


[A] Dell PowerConn
2216,
non-PTP
[B] back to back,
PTP
[C] Netgear ProSAFE
GS108,
non-PTP
[D] Toplink TK 1005G,
non-PTP
[E] Linksys EZXS55W ,
non-PTP
[F] Moxa EDS-405A-
PTP,
non-PTP (disabled)
[G] Moxa EDS-405A-
PTP,
PTP
[H] Oregano syn1588,
PTP
[I] Artel Quarra 2800,
PTP

Phase II DAQ hardware: Pixie-Net XL

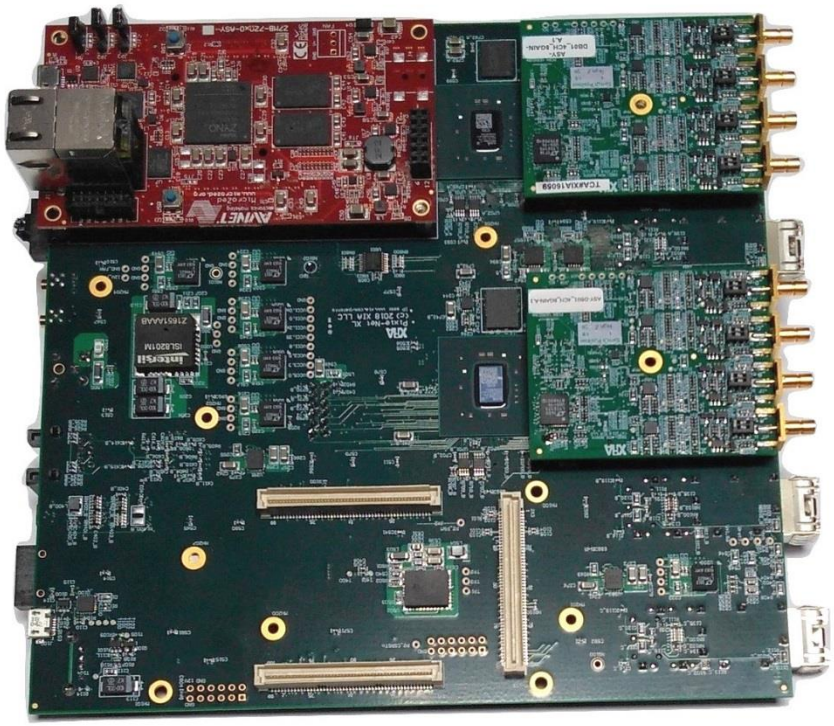
- New pulse processor board using Kintex 7 FPGA
- ADC daughter cards for detector readout (flexibility in ADC channels, rate, precision, or non-ADC functions)
- Zynq controller board reused from Pixie-Net
- High speed data flow from ADC to FPGA to Ethernet output
- WR, PTP, SyncE can be used as source for ADC and FPGA clocking

- Targets:
  - <100ps timing resolution
  - 10G Ethernet
  - high rate pulse processing



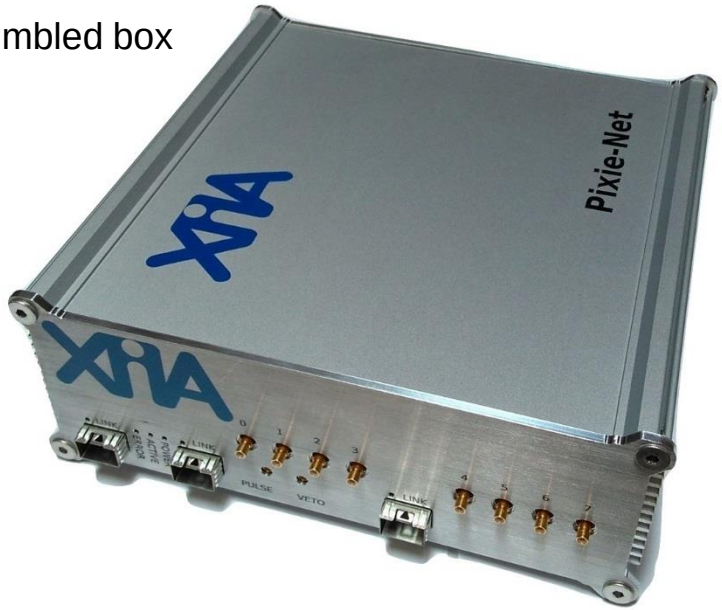
Distributed digital data acquisition system with network time synchronization

Prototypes built so far



Main board with ADC daughter cards and Zynq controller module card stack

Fully assembled box



4-channel,
75-125 MHz,
14bit ADC daughter card

8-channel,
250 MHz,
12bit ADC daughter card
Differential inputs via HDMI cable

Distributed digital data acquisition system with network time synchronization

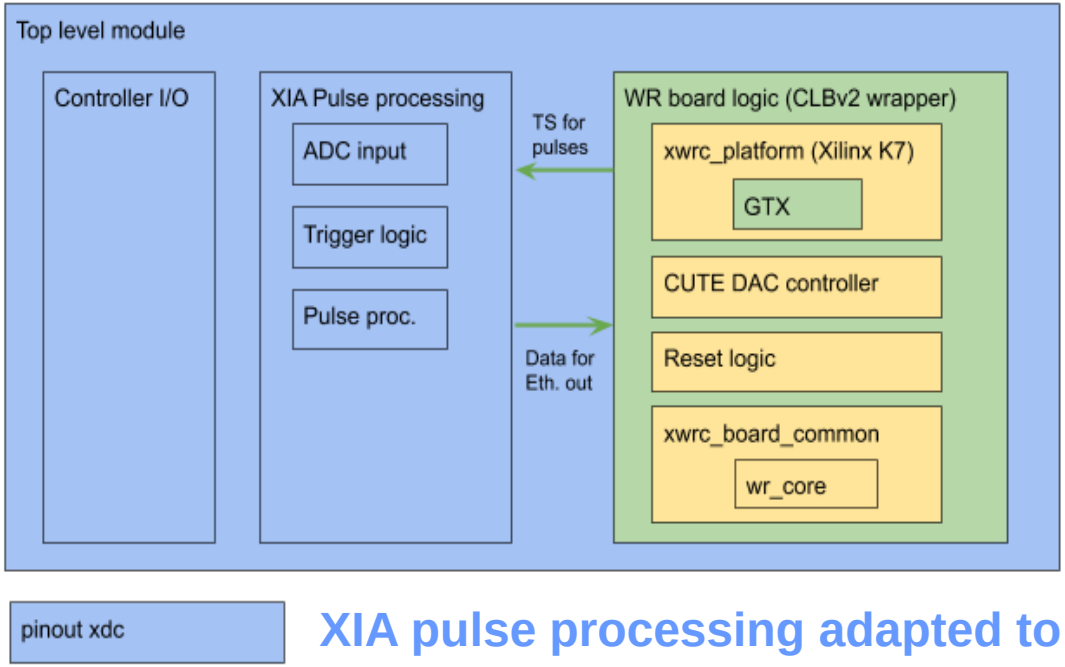
Firmware implementation

Integrated WR open source IP core into XIA pulse processing

=> Pixie-Net XL can

- capture detector pulse data and compute energy, MCA histograms, CFD timing, etc.
- act as WR clock master or WR clock slave (tuning local clocks to master frequency & phase)
- record WR date/time in captured detector data
- send out Ethernet data via WR core (work in progress)

FW block diagram



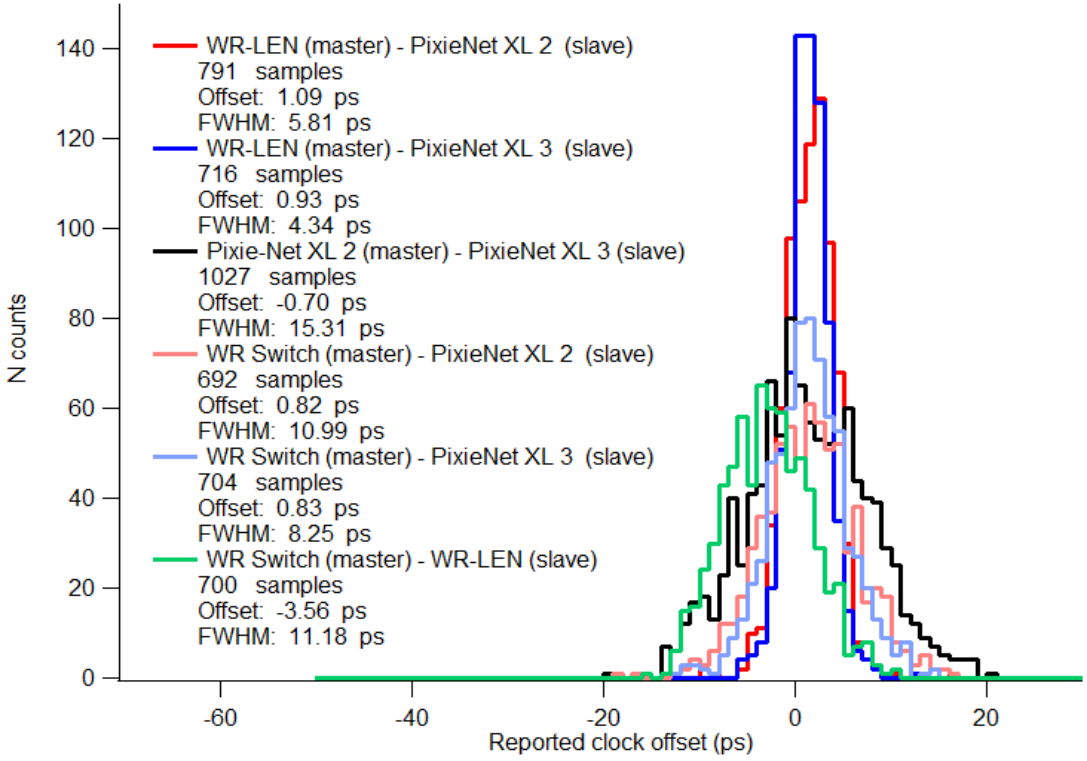
XIA pulse processing adapted to Kintex board

WR open source unchanged

WR open source adapted

Timing Measurements – SW/FW reports

- WR “SoftPLL” core reports performance values, e.g. the clock offset of WR slave to WR master.
- Measured for a variety of commercial WR modules and the Pixie-Net XL
- Histogram offsets, apply Gauss fit



⇒ Timing resolutions 4.3-15.3 ps FWHM

⇒ No significant difference between commercial WR modules and Pixie-Net XL

⇒ But is a SW report a good measure for actual performance ??

Timing Measurements – Clock jitter tests

- Probing actual clock or PPS signals on Pixie-Net XL vs PPS reference pulse from WR master (commercial WR switch)
- Oscilloscope reports std.dev. of delay from edge to edge (= jitter in our definition)

Blue: WR Master PPS (commercial WR switch)

Red: WR Slave PPS (commercial WR-LEN

Yellow: WR Slave PPS (Pixie-Net XL)

⇒ Pixie-Net XL jitter is ~100ps, but commercial WR module reaches 10-15ps

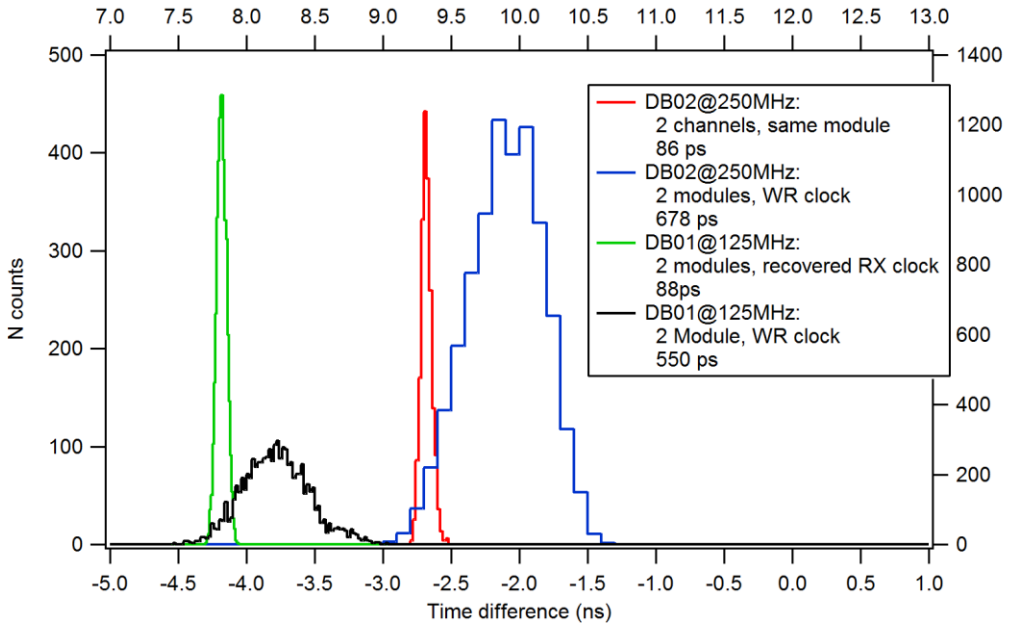
⇒ Further probing shows tuned WR clock on Pixie-Net XL is unusually jittery, and clock fanout for ADCs adds even more jitter (~300ps**) – More Study**

⇒ But fortunately the Recovered RX Ethernet clock is low jitter (17ps) and can be routed from FPGA to ADC as a workaround

Distributed digital data acquisition system with network time synchronization

Timing Measurements – Time of Flight

- Two detector signals (or split pulser) connected to two Pixie-Net XL synchronized via WR
- Capture waveforms and timestamps, compute sub-sample time of arrival by interpolation on rising edge (CFD)
- Histogram the difference of time of arrival in both modules, apply Gauss fit



Time Resolutions

⇒ With 75 MHz ADC daughter card, pulser, tuned WR clock: 550 ps FWHM

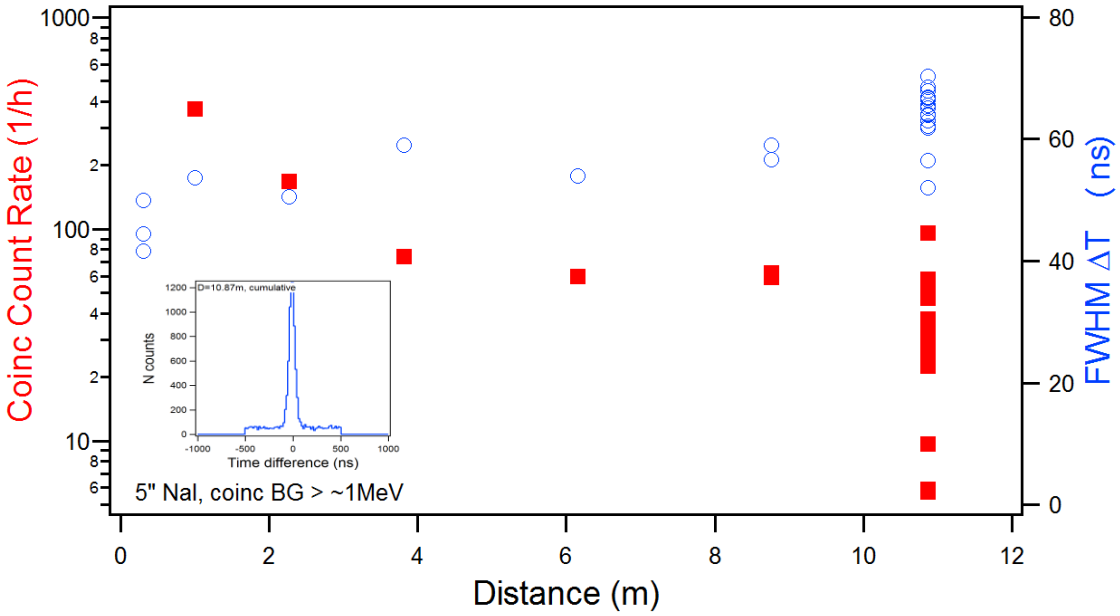
⇒ With 250 MHz ADC daughter card, pulser, tuned WR clock: 678 ps FWHM
(2 channels in same module: 86 ps FWHM)

⇒ With 125MHz ADC daughter card, pulser, recovered RX clock: 88 ps FWHM

(Best “real world” measure of performance.)

Timing Measurements – Cosmic coincidences

- Use cosmic showers as source for coincident radiation separated by large distance.
- Using large, slow detectors to demonstrate synchronization over long time and large distance (goal is not high precision)
- Only use WR time stamps, not waveform interpolation



⇒ Background rate ~500 counts/s each detector, recording ~8.5 million records (>1 MeV) per day.

⇒ Coincidence rate decreases with distance. Hundreds of coincidence events per day at ~11m distance.

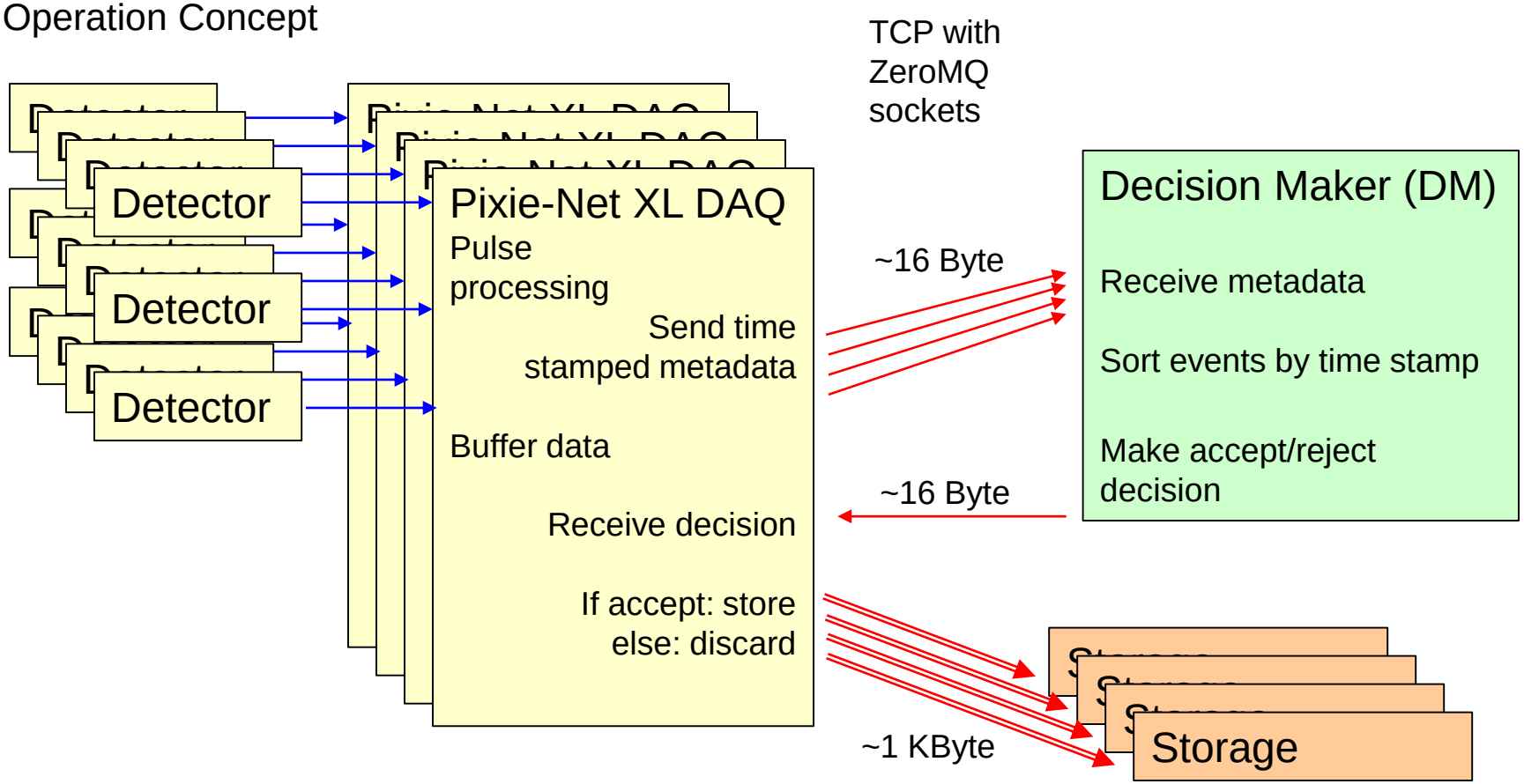
⇒ Timing resolution ~70 ns FWHM

⇒ If modules could share trigger information besides clock synchronization, we would not have to record the 99.984% waste data => use Software Triggering

Distributed digital data acquisition system with network time synchronization

Software Triggering

Operation Concept



Demonstration and coding example – users will have their own ideas

Distributed digital data acquisition system with network time synchronization

Software Triggering

Initial implementation

- **DAQ: Pixie-Net Zynq ARM Linux**
  Instrumented acquisition routine (C) to open socket connections and exchange trigger metadata with DM.
  Buffer event data in a ring buffer (~200 events).
  Store waveform to file on accept.
  Rejected/ignored triggers expire as the buffer wraps around.

- **DM: Ruby script on a Linux VM**
  Bind socket servers, sync run start across modules, and analyze trigger metadata.
  Track pending timestamps as counts in sliding window partitions (e.g. 2ms buffer, 1us windows, 10ns stride).
  Frequently sum sliding windows to find multiple hits and accept ranges.
  Initial tests: 1.5 kcps x 2 modules for proof of concept. Convert to C for production.

Sample message flow
window=200us, timestamps in ns

DAQ 1

```
> TRIGGER mod0,ch0,8477648
> TRIGGER mod0,ch0,9133008
...
< ACCEPT 8300000,8500000
  (store 8477648 waveform)
```

DAQ 2

```
> TRIGGER mod4,ch1,8369496
> TRIGGER mod4,ch1,9524856
...
< ACCEPT 8300000,8500000
  (store 8369496)
```

DM

```
< TRIGGER mod0,ch0,8477648
< TRIGGER mod4,ch1,8369496
> ACCEPT 8300000,8500000
< TRIGGER mod4,ch1,9524856
< TRIGGER mod0,ch0,9133008
  (ignore)
```

Distributed digital data acquisition system with network time synchronization

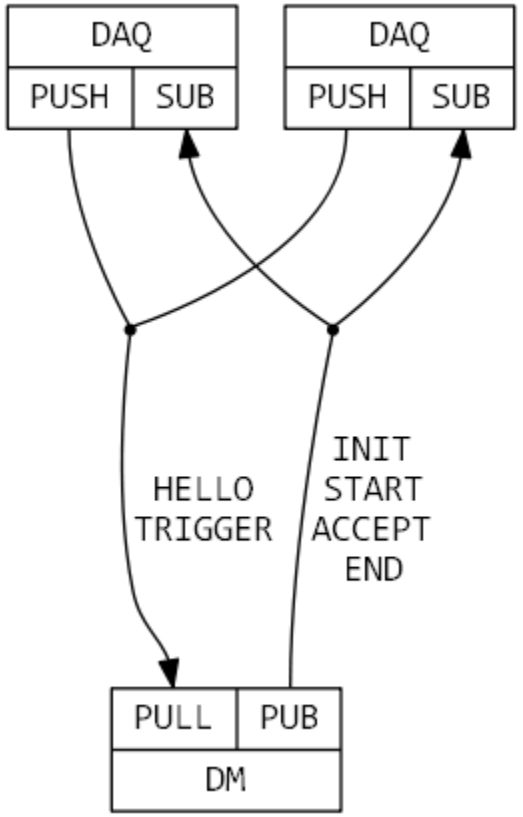
Software Triggering

Initial implementation

- **ZeroMQ sockets over TCP**
  Message-oriented programming model with TCP reliability.
  Multithreaded IO built in.
  ZeroMQ benchmarks on Zynq Linux:
  - Mean latency: 170us one-way
  - Mean throughput: 525k msg/s
  Select socket types according to system topology and communication patterns.
  - Triggers: PUSH/PULL pipeline style
  - Decisions: PUB/SUB broadcast style

- **ASCII encoding on the wire initially**
  ~30 Bytes per message including overhead.
- To go fastest, encode binary (~16 bytes), pack multiple triggers into 1500 MTU, and try UDP.

ZeroMQ socket types



Distributed digital data acquisition system with network time synchronization

Commercial Branches: SBIR work adopted in other XIA products*

- **Pixie-Net PTP**
  Phase I prototype commercially available.
  Several units sold

- **Pixie-16 MZ-TrigIO**
  Trigger I/O module for XIA's 6U PXI pulse processor boards with PTP clock option.
  First unit sold.
  Also available as desktop PTP GPIO module

- **Pixie-4 Hybrid**
  Update of XIA's 3U PXIe pulse processor board with WR synchronization and data output.
  Prototype in testing.
  Better jitter than Pixie-Net XL

* Mainly funded by XIA

Summary

Hardware platform

- Implemented and characterized WR network time synchronization on new detector DAQ electronics, the Pixie-Net XL
- Easily reaches “<1 ns” timing resolution goal, demonstrated <100ps, but needs more tweaks and revisions to reach WR reference performance of <20ps.
- Pixie-Net XL is now being evaluated at a collaborating research lab.

Software triggering

- Fairly simple protocol
- Intended as demo and building block for user code using Pixie resources

Outlook

- Improving Ethernet data out via WR, then upgrade to 10Gbps rates
- Porting software triggering to Pixie-Net XL, develop user demo and GUI
- More ADC daughter cards, performance testing, …

Detector Readout Electronics Synchronized Through Ethernet Network Timing Techniques

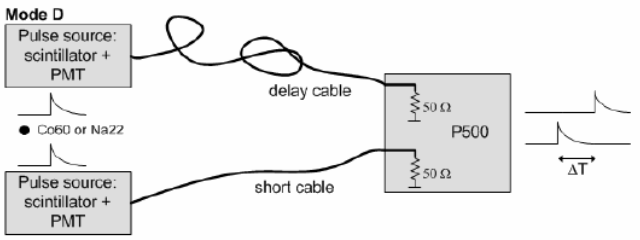
Thank You

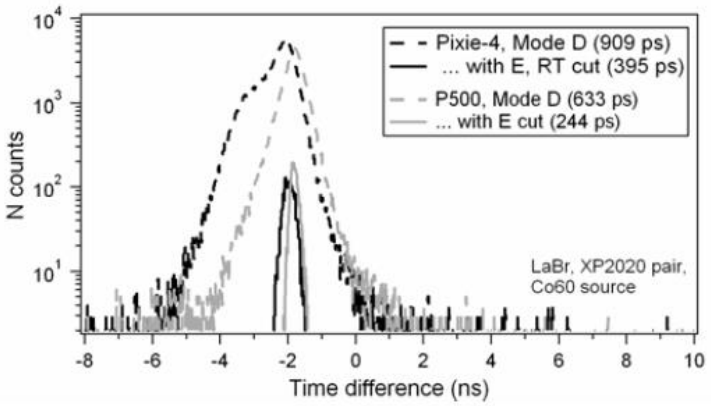
Questions?

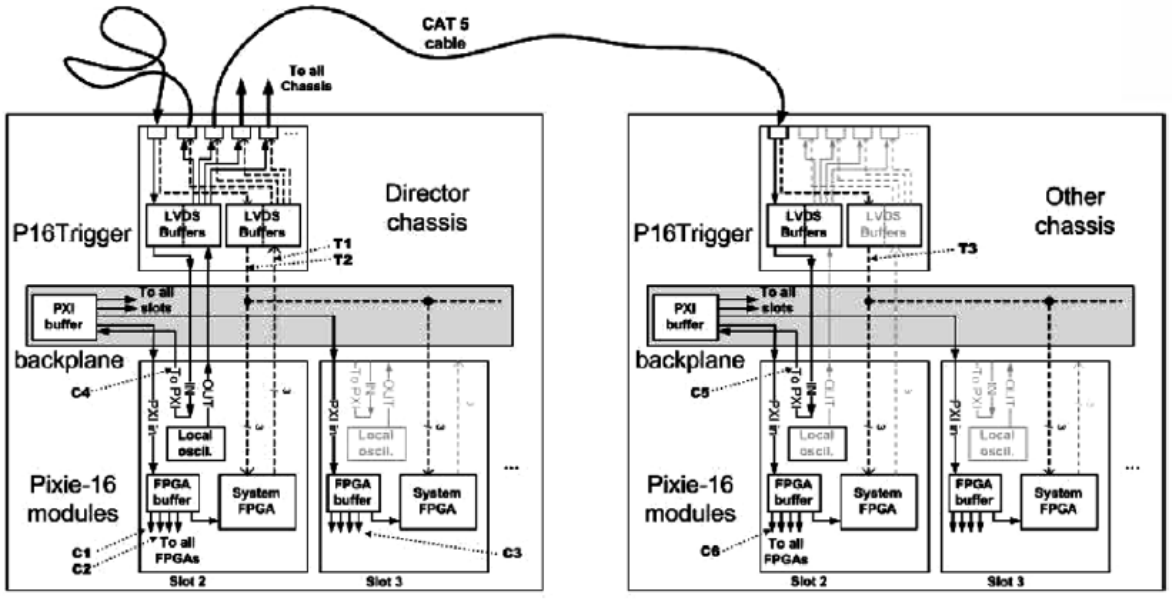
Traditional Synchronization

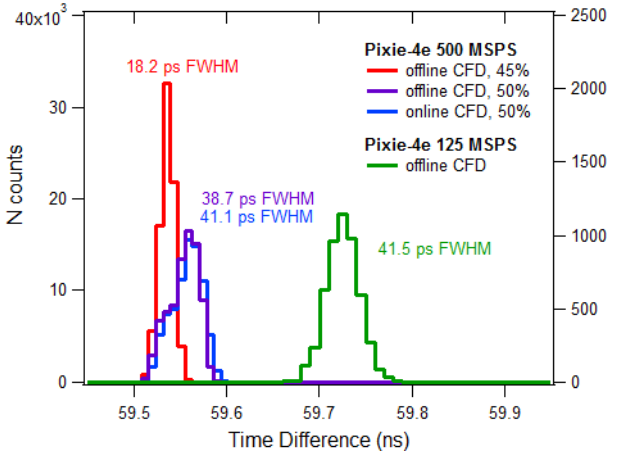
Traditionally, time synchronization between multiple channels of digital data acquisition is accomplished by sharing clocks, clock reset signals, and triggers.

With suitable algorithms (CFD), timing resolutions can be ~20 ps for idealized signals and a few hundred ps for detector signals digitized with 100-5000 MSPS.

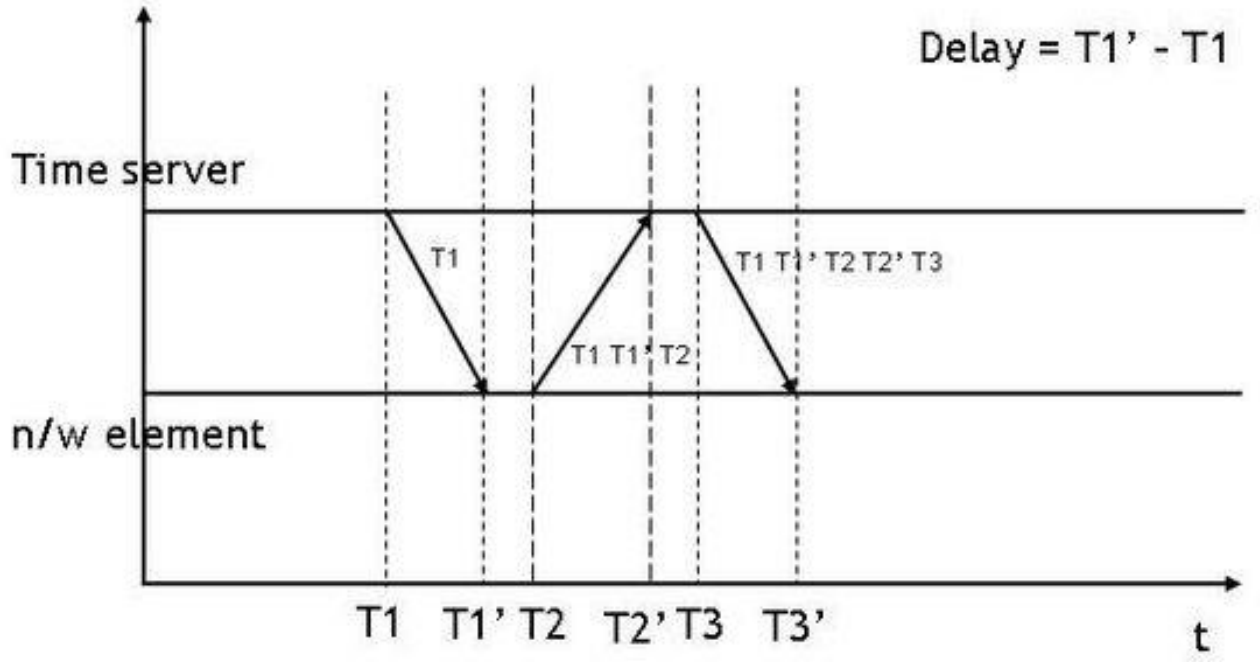








PTP synchronization (Wikipedia)



If d is the transit time for the *Sync* message, and $\tilde{o}$ is the constant offset between master and slave clocks, then

$$T1' - T1 = \tilde{o} + d \text{ and } T2' - T2 = -\tilde{o} + d$$

Combining the above two equations, we find that

$$\tilde{o} = (T1' - T1 - T2' + T2)/2$$