DUNE Timing System – Single Phase Development Progress and Plans

Stoyan Trilov

DUNE Timing Final Design Review

21/07/2020

University of BRISTOL

Overview

- **A summary of progress on DTS-SP prototyping so far, and plans for future development**
- **Topics which will be covered**
  - Hardware
  - Firmware
  - Software
  - Functionality
- **Mode details can be found in the test plan and results write-ups**
  - https://edms.cern.ch/document/100663458/1 – test plan
  - https://edms.cern.ch/document/100663459/1 – test results

Hardware

GPS receiver

- Prototyping with a SecureSync time reference system at Bristol: https://www.orolia.com/sites/default/files/document-files/SecureSync%20Standard%2005-29-20.pdf
  - N.B. access to lab at Bristol currently limited
- 10 MHz clock output verified using oscilloscope

- **Next steps**
  - Verify, and decode IRIG code output
  - Study how clock and IRIG code behave under normal and abnormal conditions, e.g. long-term stability, loss of signal
  - Verify output of IEEE-1588v2 (PTP) module

GPS Interface Board (GIB)

- **Board testing (Penn/Bristol)**
  - verify I2C communication with on-board ICs
  - verify that the 62.5 MHz clock can be generated from 10 MHz input (GPS receiver or otherwise)
  - verify timestamp generation and timing protocol downlink and uplink
  - verify decoding of IRIG signals in firmware

- **Firmware development**
  - IRIG decoding
  - Timestamp generation

mTCA Interface Board (MIB)

- **Board testing (Bristol/Penn)**
  - Board controller (MMC) programming and testing
  - verify I2C communication with on-board ICs
  - verify clock recovery mechanism and clock distribution tree
  - verify timestamp and clock generation and timing downlink and uplink in MIB standalone mode, i.e. no GIB

- **Firmware development**
  - Reception of timestamp from GIB
  - Injection of timing commands via software
  - Timing partitions

Fibre Interface Board (FIB)

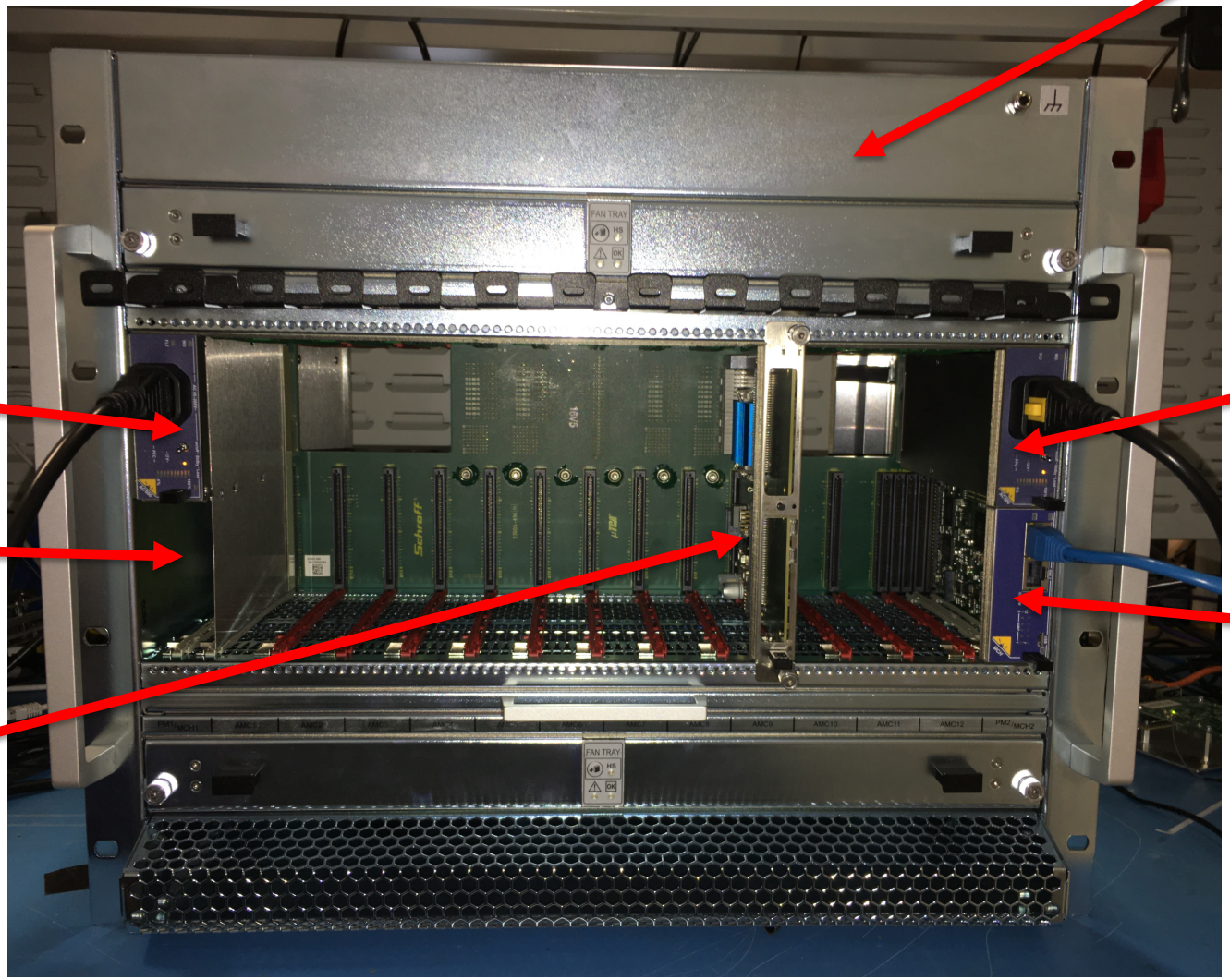
- **Board testing (Bristol)**
  - verify I2C communication with on-board ICs
  - verify downlink capability for each of the eight SFP transceivers
  - verify uplink capability for each of the eight SFP transceivers, via the 8:1 multiplexer
  - verify direct uplink capability for the four SFP which have such capability

- **Firmware development**
  - Design already produced, to be tested with FIB

mTCA system and AMC (AFC)

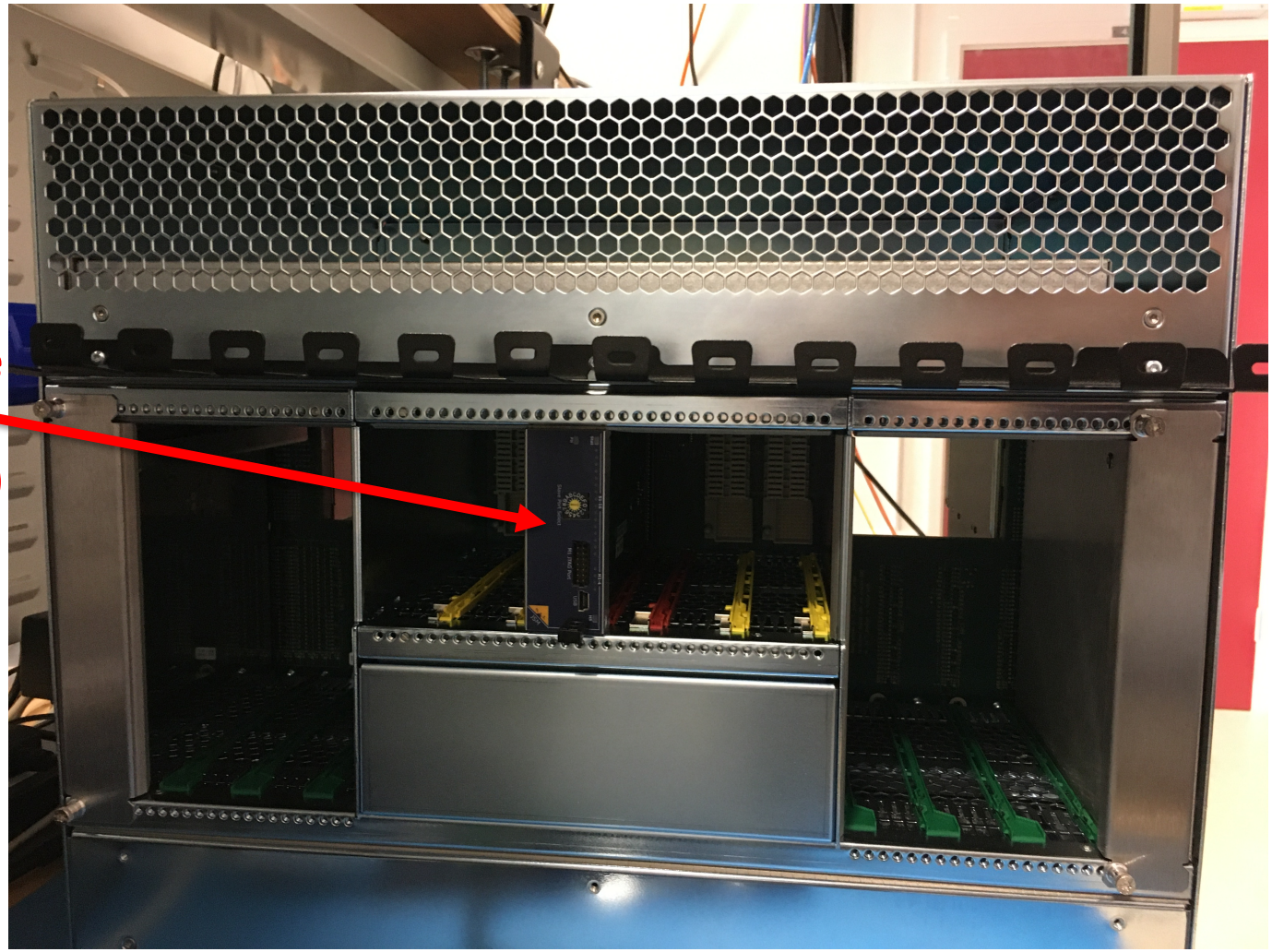
- mTCA system installed in Bristol lab
- Verified operation of AMC FMC Carrier (AFC), current FIB carrier board candidate, with mTCA system
  - Able to program AFC FPGA remotely (via crate controller (MCH) and JTAG switch module)
  - Ported existing endpoint firmware to AFC
    - IPBus control via Ethernet (crate controller acts as a switch)
    - Verified downlink and uplink with timing master (AIDA 2020 TLU) using an existing timing FMC design
- **Next steps**
  - Test AFC with FIB, MIB
    - Verify clock and signal routing of AFC

uTCA system (front)



uTCA system (back)

JTAG Switch Module
(JSM)
(remote JTAG access)

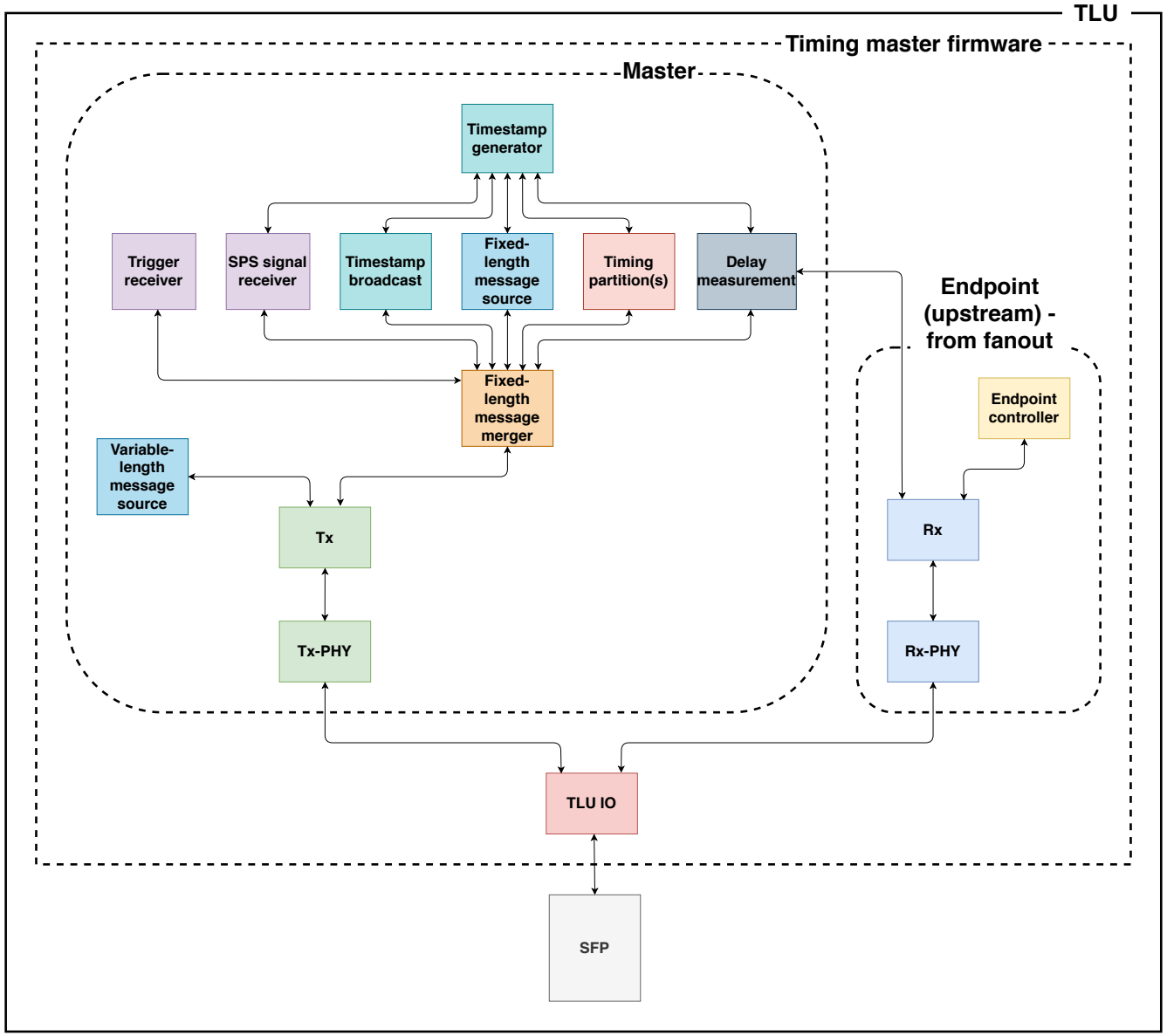


Firmware

Firmware

- **Existing firmware designs**
  - Timing master: timestamp and clock generation, injection point for timing commands (combines functionality of GIB and MIB)
  - Fanout: 1:8 distribution of timing data-stream (downstream), 8:1 uplink data-stream pass-through (only one active at a time)
  - Endpoint: maintain synchronised timestamp, receive incoming timing messages, send timing messages, adjustable delay (units of 20ns)
- **DTS-SP developments**
  - Endpoint firmware already ported and tested on AFC
    - Existing timing FMC used
  - Endpoint and fanout firmware for FIB produced
    - Will be tested once FIB is available
- **Next steps**
  - Test firmware at 62.5 MHz
  - Develop fine delay adjustment (units of 4 ns, 3.2 ns)
  - Develop GIB and MIB firmware (split functionality of existing timing master design)

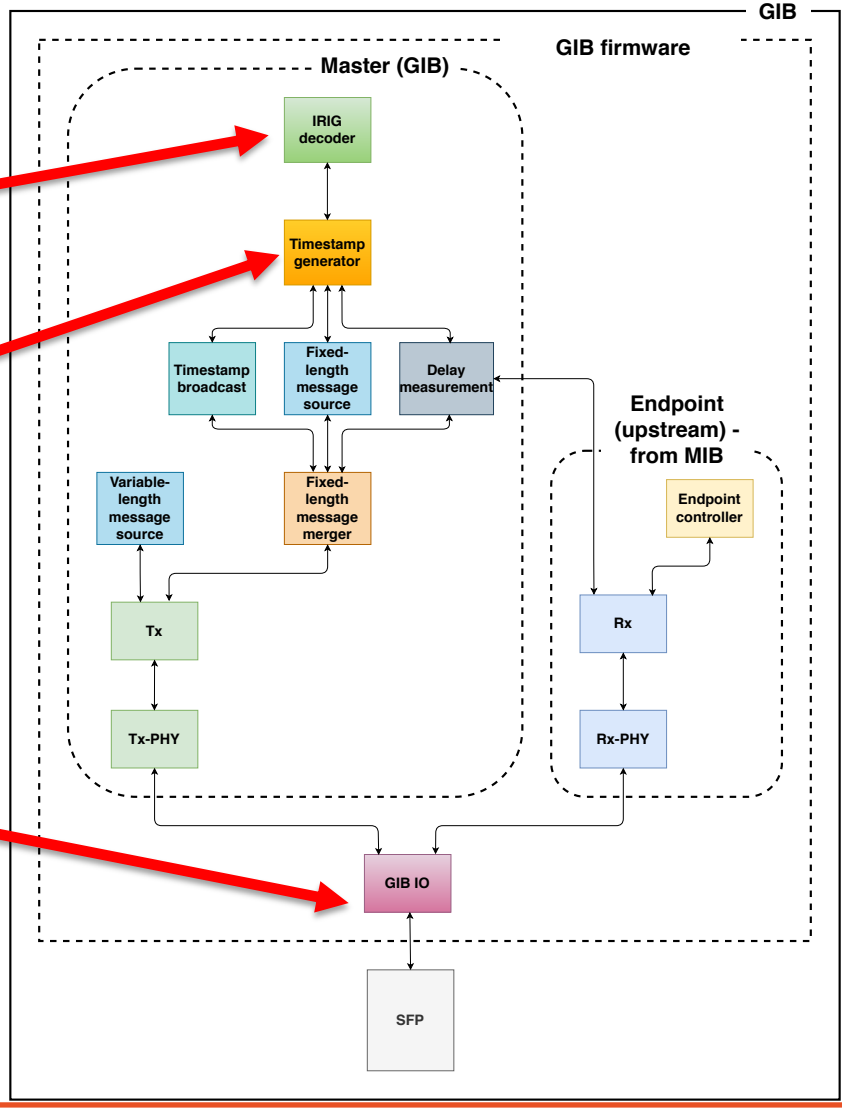
Existing master firmware design



GIB firmware design

New or modified firmware blocks

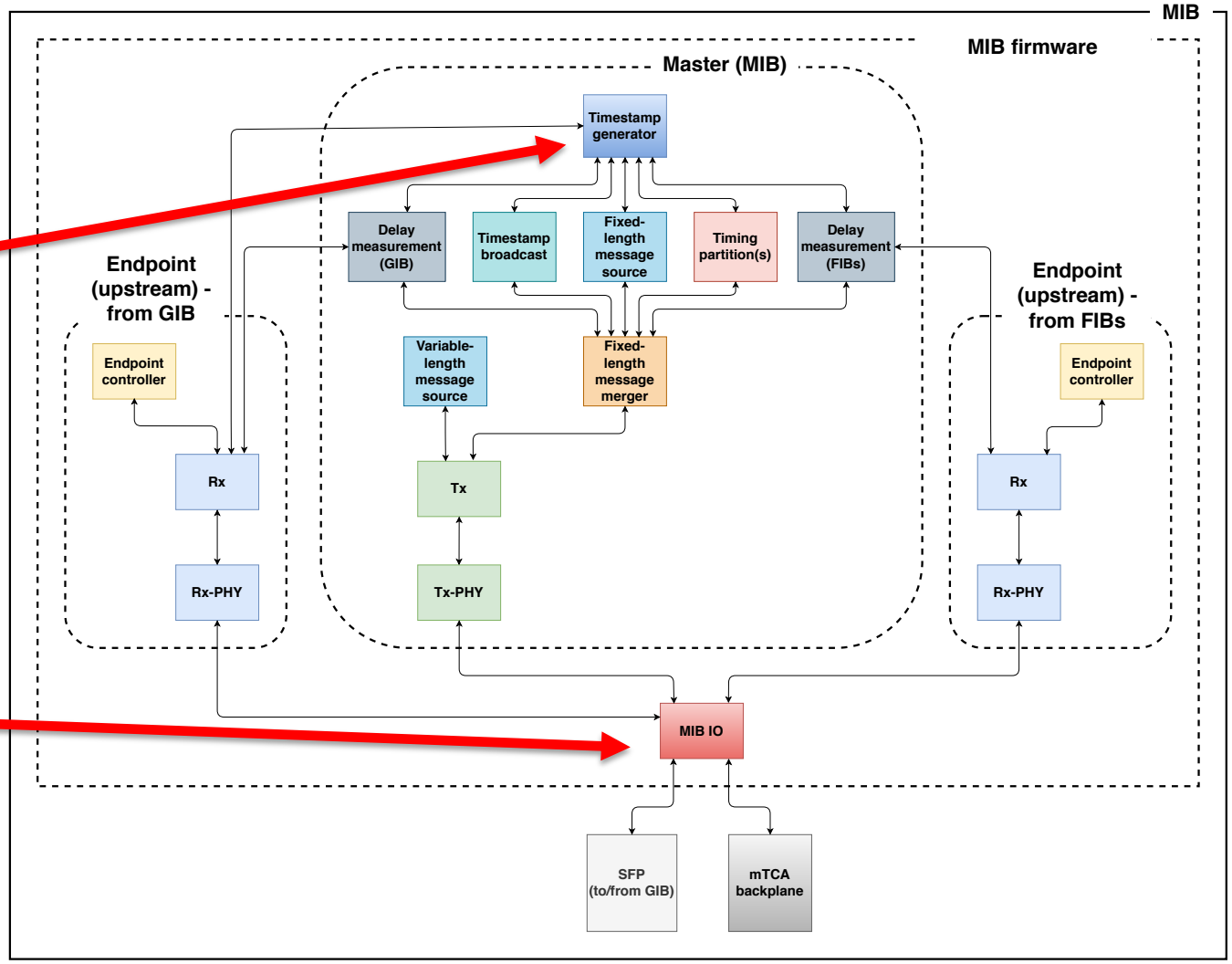
- IRIG decoder
- Timestamp generator
- GIB IO



MIB firmware design

New or modified firmware blocks

- Timestamp generator
- MIB IO



Software

Software

- **Existing timing core library (C++) functionality**
  - Implements IPBus hardware access layer
  - Implements higher-level functions composed of IPBus operations, e.g. configure FMC clock chip, configure timing partition, read data buffer
  - Interface to timing system via python CLI (pdtbutler), and DAQ software (artDAQ BoardReader)
- **Development of library towards DTS-SP so far**
  - IPMI communication
  - AFC integration, e.g. clock crossbar configuration
  - FIB integration
- **Next steps**
  - Develop GIB and MIB software control functions
  - Integrate library within PD2/DUNE DAQ application framework (when ready)

Functionality

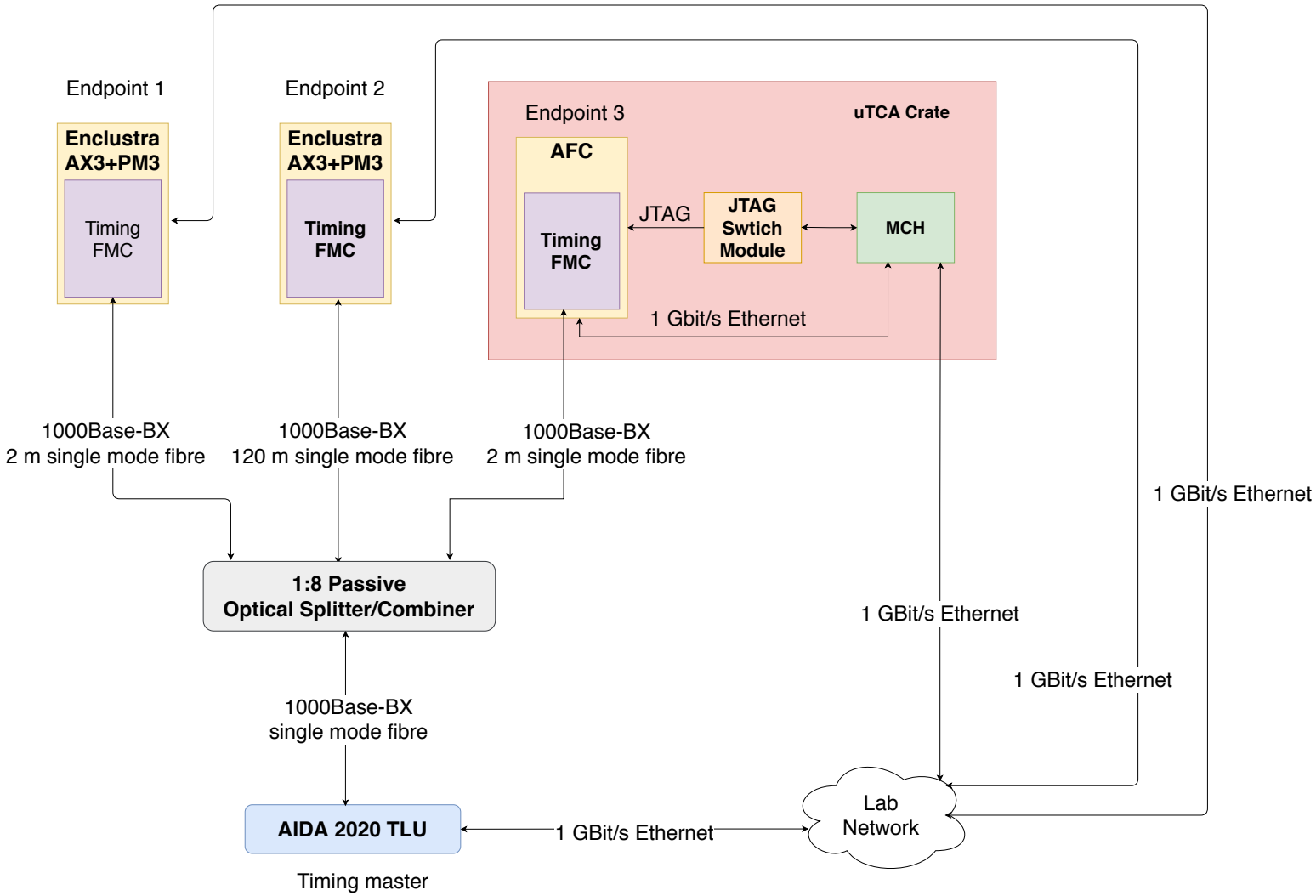
Endpoint alignment test

- Modified endpoint firmware outputs a “sync” pulse at the reception of a particular timing command
  - Two endpoints receiving commands from the same timing master, using a passive optical splitter
  - Endpoint 1 is connected to the master using a 2 m fibre
  - Endpoint 2 is connected to the master using a 120 m fibre
    - ~4.9 ns delay per 1 m of fibre

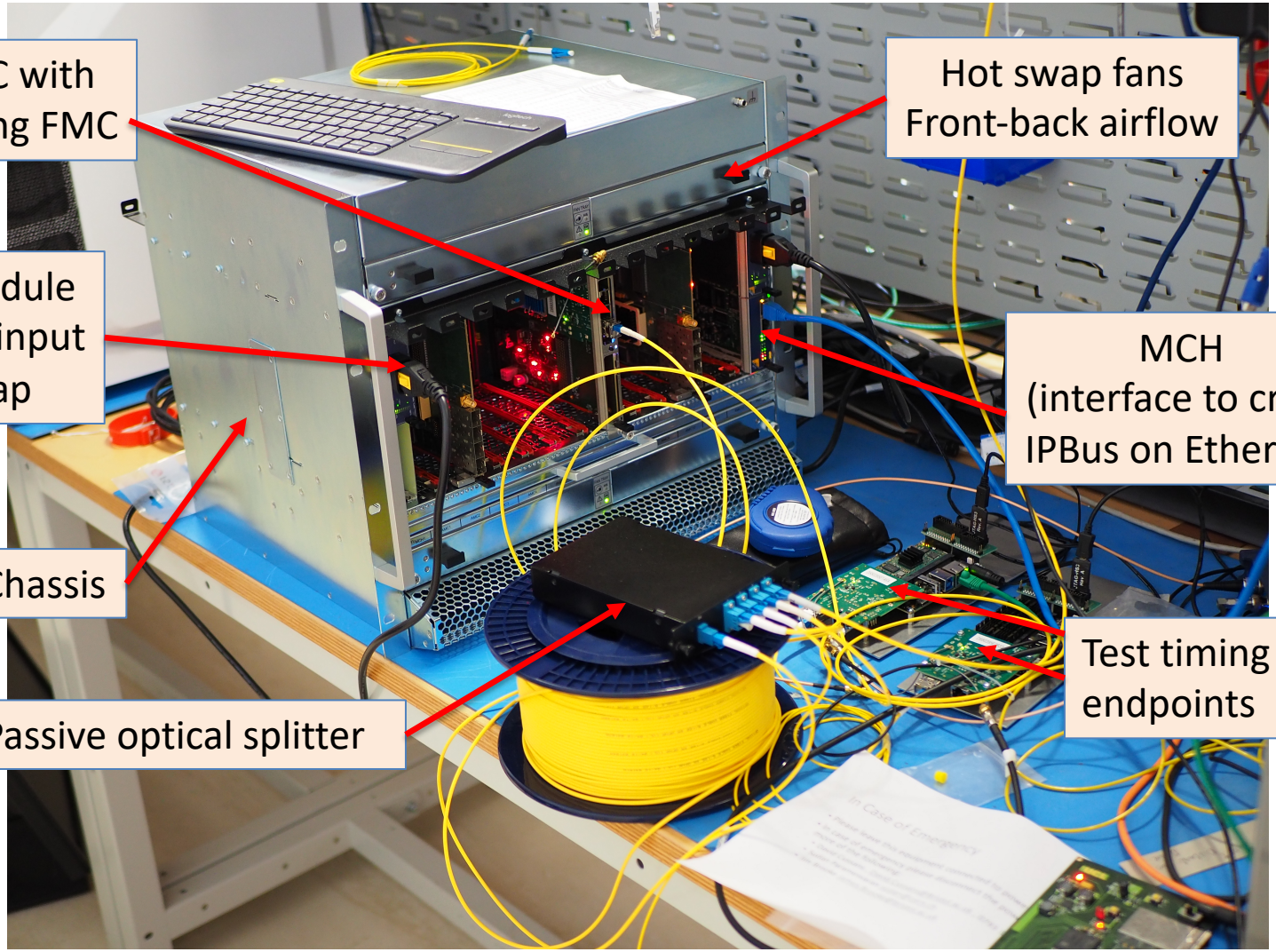
- Measure the time between the pulses emitted from the two endpoints with no delay adjustments
  - $\Delta t = t_2 - t_1$

- Adjust delay setting on endpoint with the short fibre, re-measure the time between the two pulses

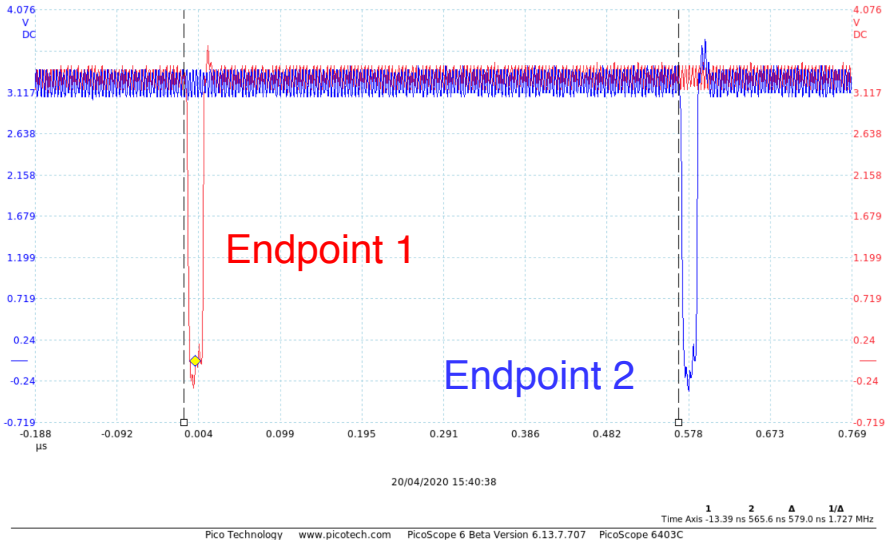
Test setup - diagram

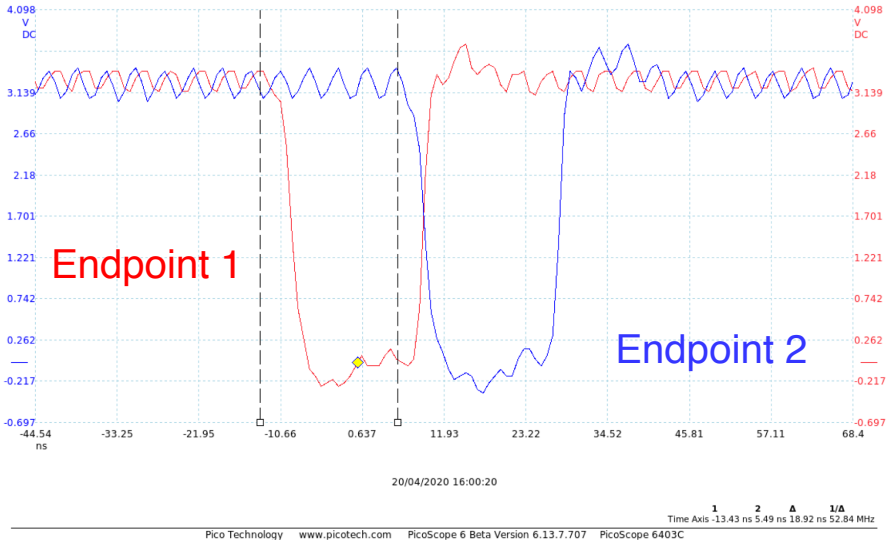


Test setup - photo



Endpoint alignment test – results





- No delay on endpoint 1
- $\Delta t = 579$ ns, as expected

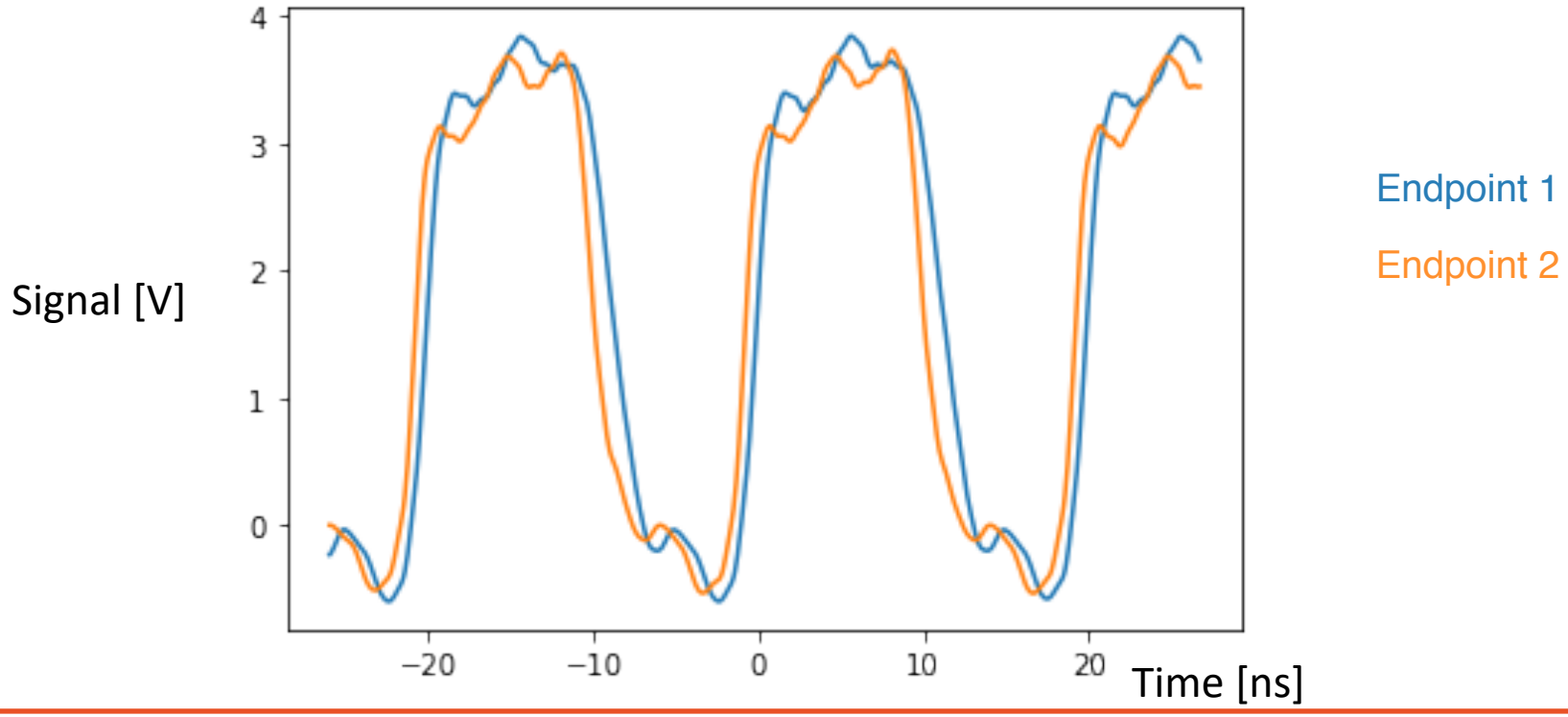
- Delay of 28 clock ticks; 28*20 ns = 560 ns for endpoint 1
- $\Delta t = 19$ ns, as expected

- Endpoints will eventually be aligned further
  - Current functionality only allows delay adjustment in units of 50MHz clock cycles
    - Feature still being refined and tested, only certain delay values work
  - In future there will be finer alignment in units of 250MHz, and eventually 312.5MHz

Endpoint clock studies

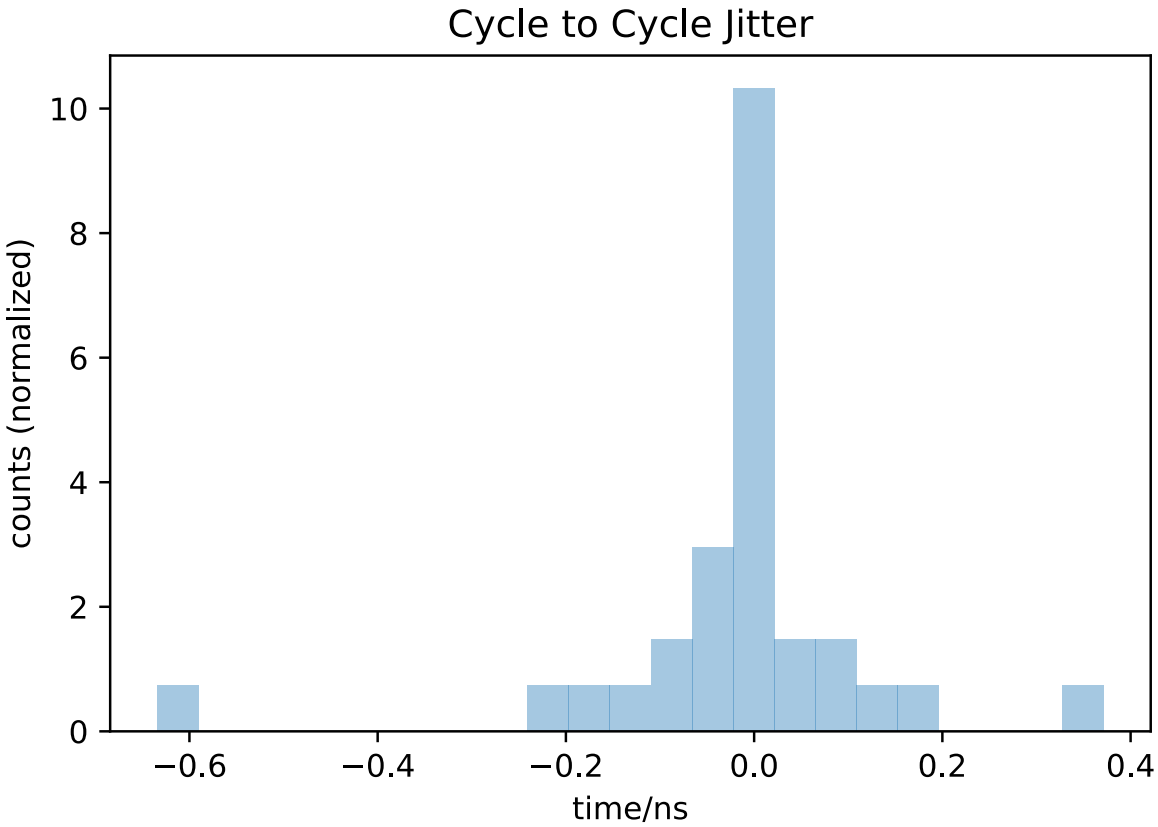
- Study the properties of the clocks produced by two endpoints
  - Cycle-to-cycle jitter
  - Difference in time of clock edges

Example of data used to perform studies on following slides



Endpoint cycle-to-cycle jitter

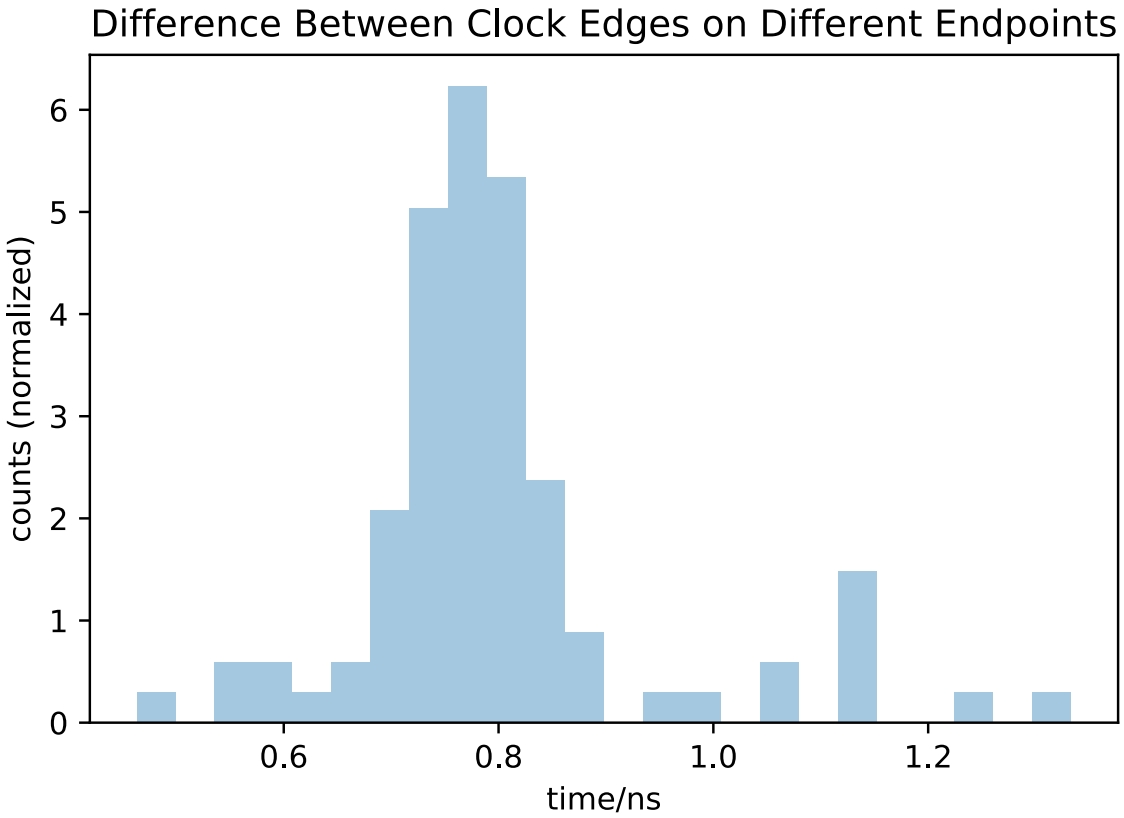
- Cycle-to-cycle jitter (J_{CC})
  - Difference in period between consecutive cycles



- $\overline{J_{cc}} = -0.019$ ns
- $\sigma_{J_{CC}} = 0.155$ ns

Endpoint clock alignment

- Clock alignment (Δt_r)
  - Difference in time of clock edges of the two endpoints



- $\overline{\Delta t_r} = 0.802$ ns
- $\sigma_{\Delta t_r} = 0.140$ ns

Summary

- **Short/medium hardware test plan complete**
- **Firmware**
  - A lot of basic functionality already present
  - Work underway to transition to new hardware and operational paradigm
- **Software**
  - A lot of basic functionality already present
  - New hardware and communication protocols being integrated
- **Preliminary studies of endpoint alignment and clock properties**
  - Encouraging results of clock quality (< 5 ns alignment requirement)
  - Oscilloscope used not ideally suited for task (due COVID-19), tests will be repeated