bPID Beam Test Electronics – Integration Status and Plans

Matt Andrew, Kurtis Nishimura, Gary Varner

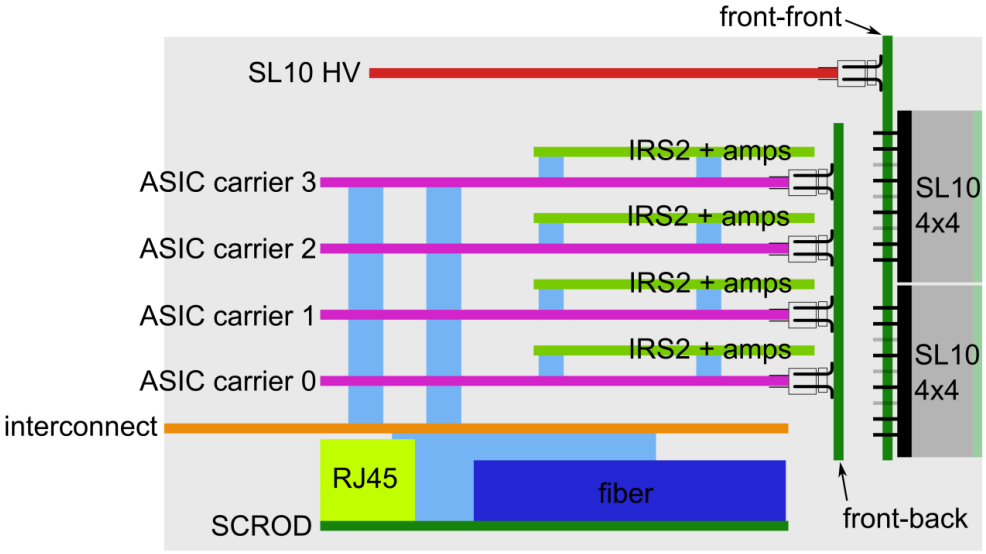
University of Hawaii

August 11, 2011

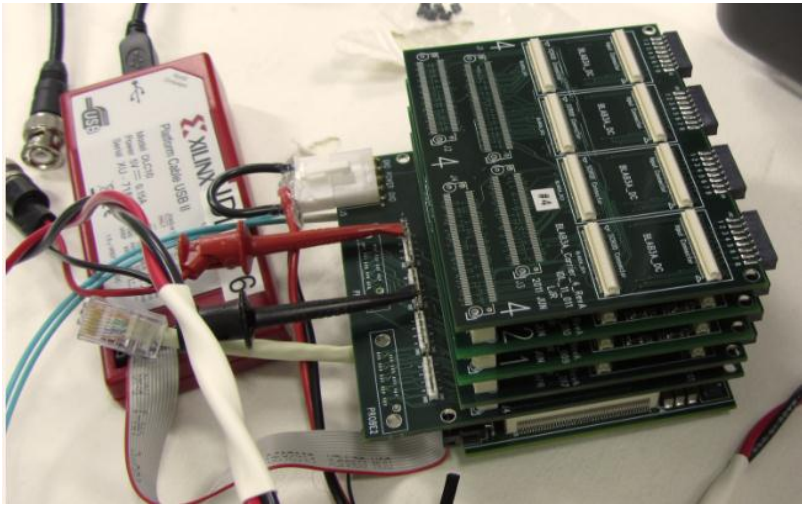
Overview

- Quick review of existing hardware pieces
  - Hawaii electronics.
  - Nakao-san's FTSW board.
- Primary goal this trip:
  - Integrate with Nakao-san's FTSW board and demonstrate low jitter clock distribution.
  - Will report existing test results and next testing plans.

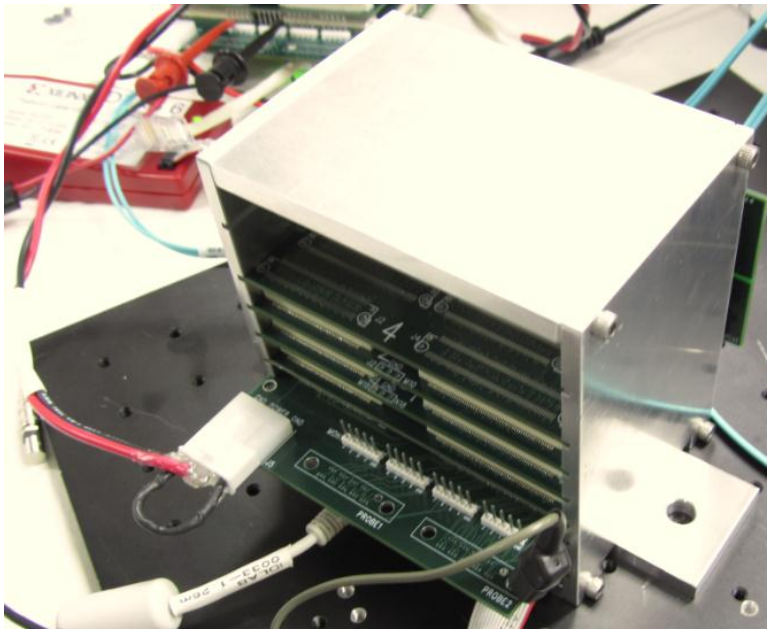
Beam Test Front-End Electronics



- Front-end modules digitize all SL10 signals and send data off by fiberoptic.



Front-end board stack (HV, front boards, daughter cards, not yet populated)



Front-end board stack in temporary mechanical enclosure.

Trigger/Timing Distribution (FTSW)

From Nakao-san's documentation:

20110805 version

Timing signals over CAT7 cables

7 ports, O1 to O7

ACK → ACK: 254 Mbps serialized, unused
TRG ← TRG: 254 Mbps serialized
RSV ← RSV: pulled down to GND
CLK ← CLK: 127 Mhz

JTAG signals over CAT7 cables

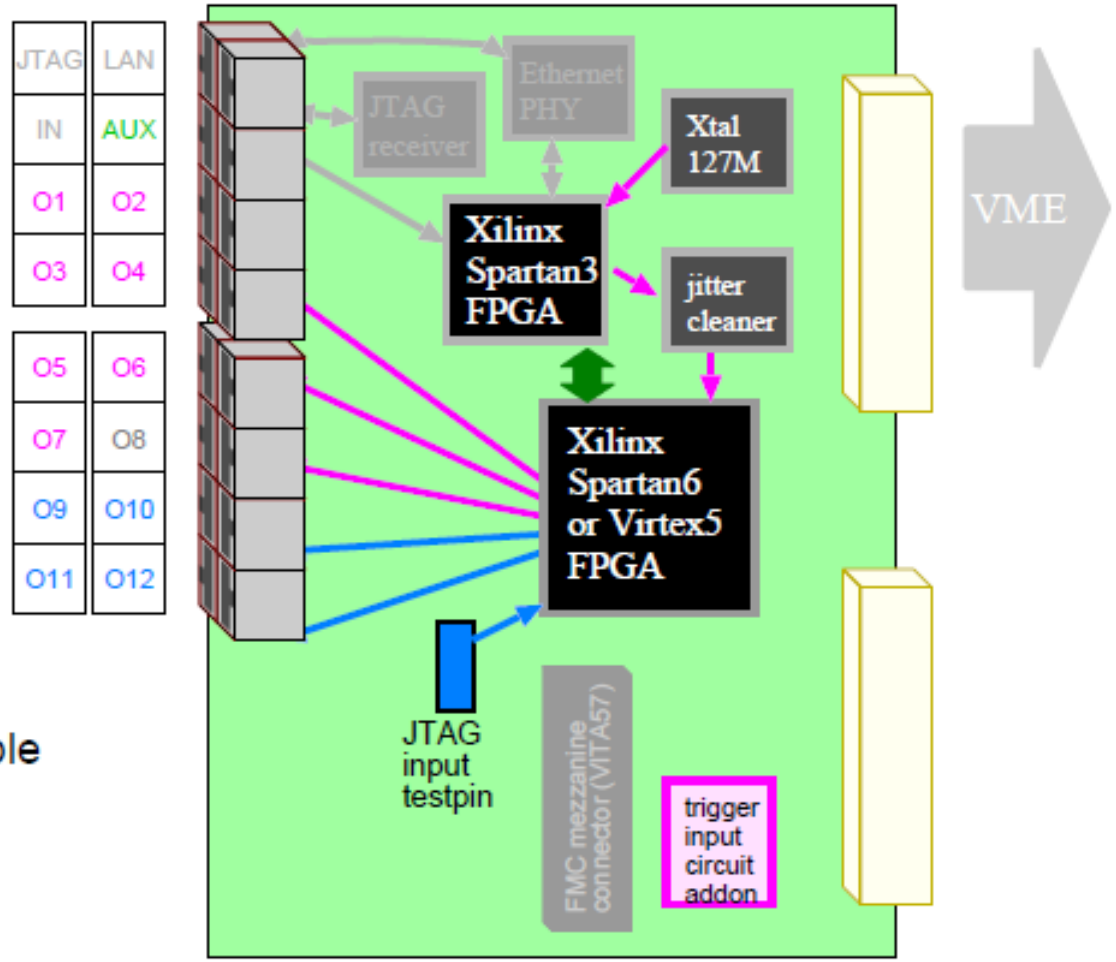
4 ports, O9 to O12

TCK ←
TMS ←
TDI ←
TDO →

Monitoring signals over a CAT7 cable

AUX port

trgin ← copy of trigger input
trg21 ← latched with 21MHz clock
trgpulse ← trgin and (not trg21)
clk21 ← 21MHz clock



Trigger/Timing Distribution (FTSW)

From Nakao-san's documentation:

20110805 version

Timing signals over CAT7 cables

7 ports, O1 to O7

- ACK → ACK: 254 Mbps serialized, unused
- TRG ← TRG: 254 Mbps serialized
- RSV ← RSV: pulled down to GND
- CLK ← CLK: 127 Mhz

JTAG	LAN
IN	AUX
O1	O2
O3	O4
O5	O6
O7	O8
O9	O10
O11	O12

JTAG signals over CAT7 cables

4 ports, O9 to O12

- TCK ←
- TMS ←
- TDI ←
- TDO →

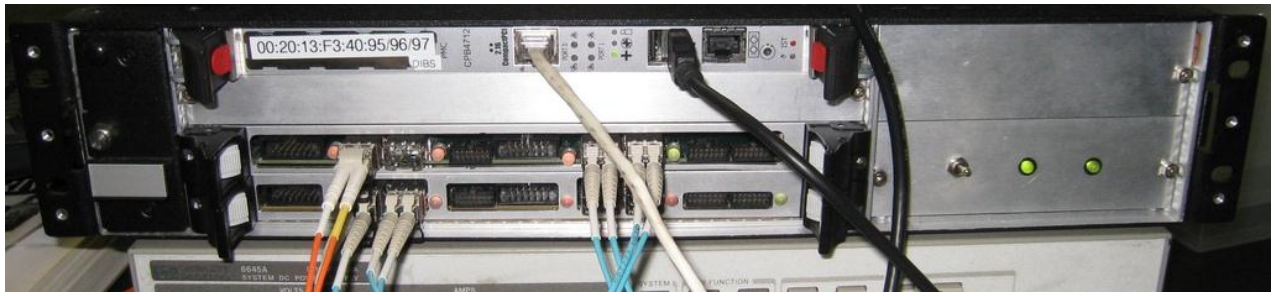
Monitoring signals over a CAT7 cable

AUX port

- trgin ← copy of trigger input
- trg21 ← latched with 21MHz clock
- trgpulse ← trgin and (not trg21)
- clk21 ← 21MHz clock

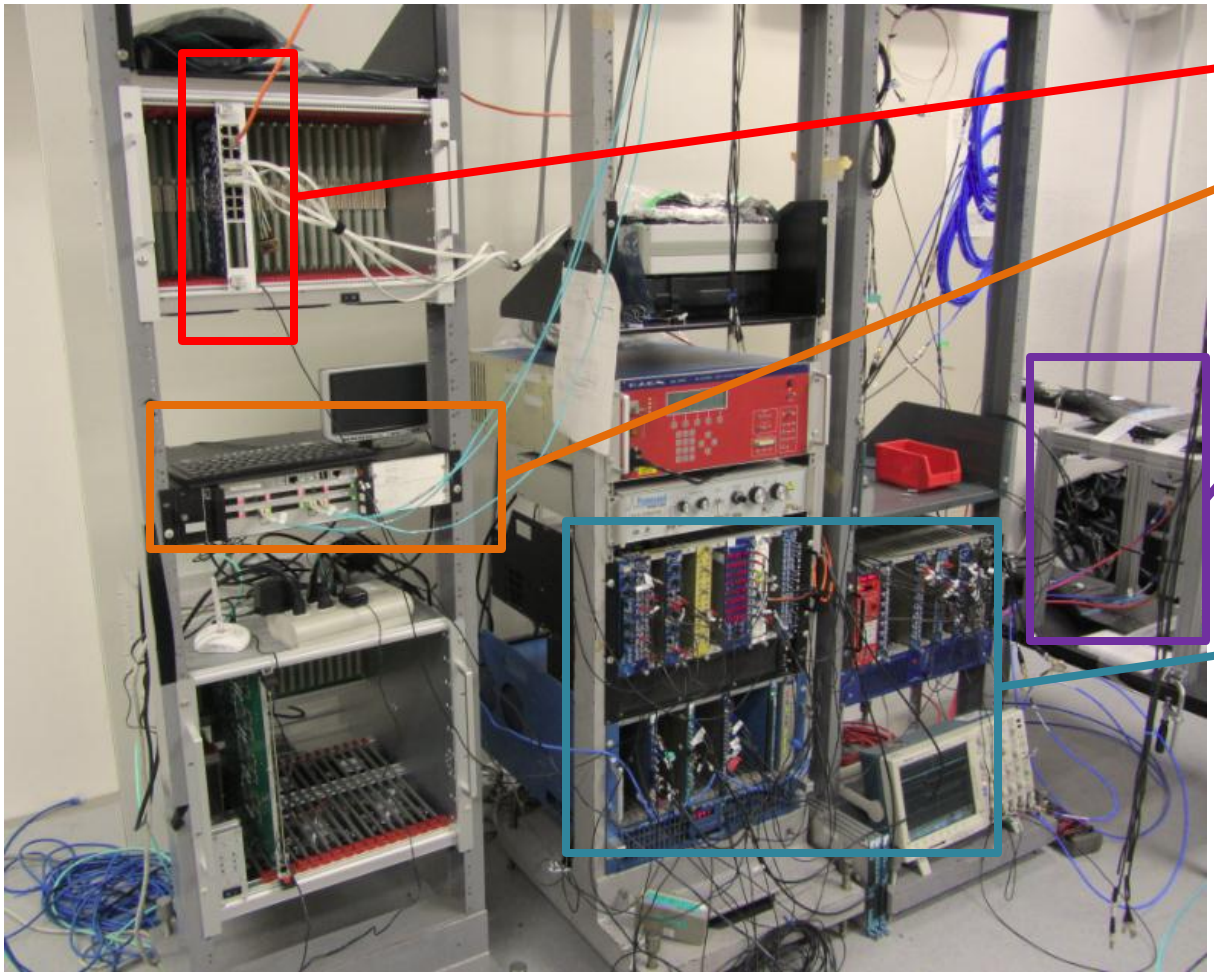
VME

Back-End Data Acquisition



- cPCI crate (above) with custom card (DSP_cPCI, left), that receives data from front end modules.
- This crate will collect all data for the cosmic and beam tests.
  - Collects data from front-end modules by fiberoptic.
  - Collects CAMAC data via USB.

Current Testing Setup



FTSW Module

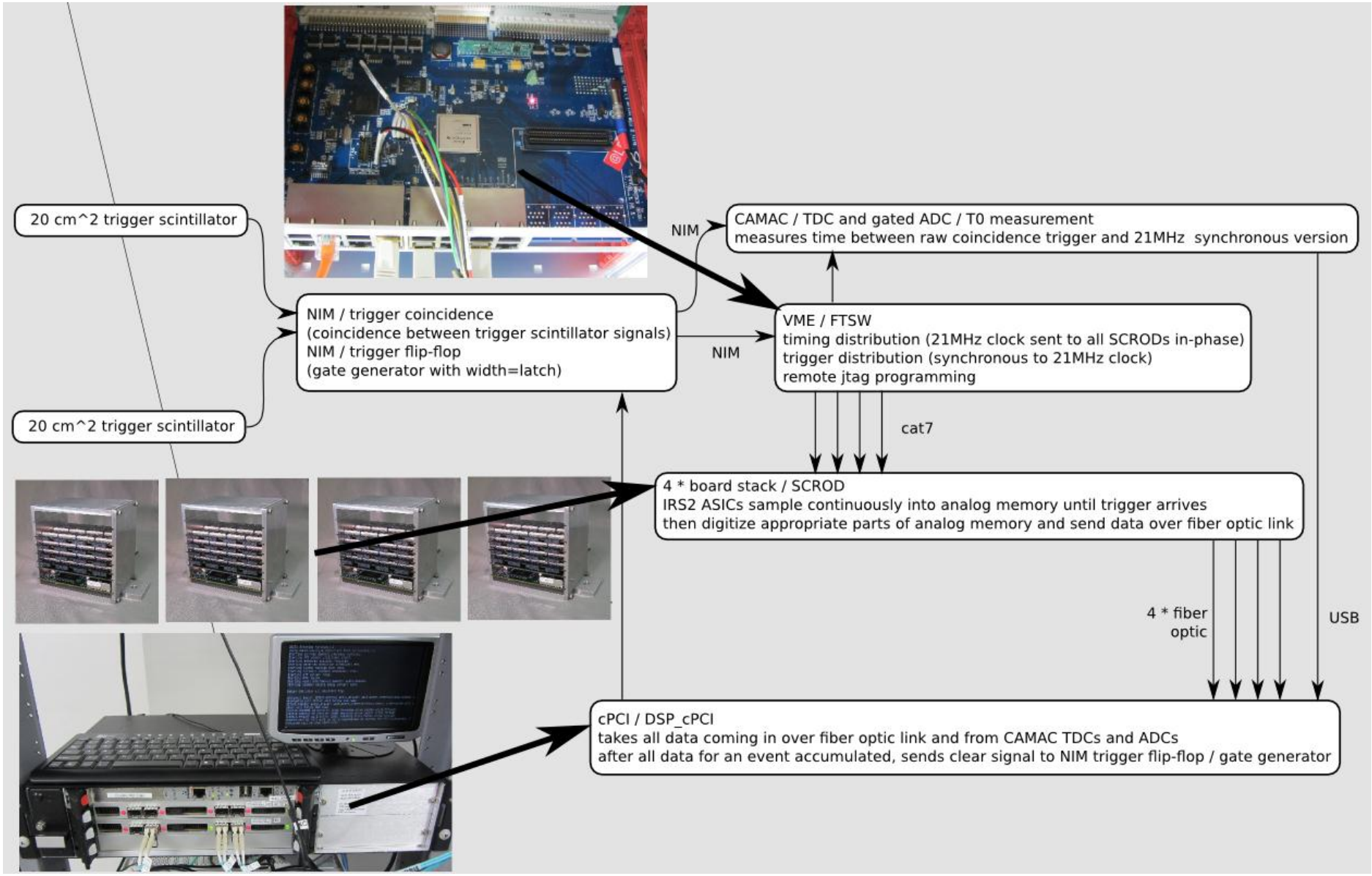
Back-end fiberoptic System

Scintillator paddles & timing counters

NIM modules & CAMAC crate

Thank you for very nice setup and use of equipment.

Cosmic Test System Diagram



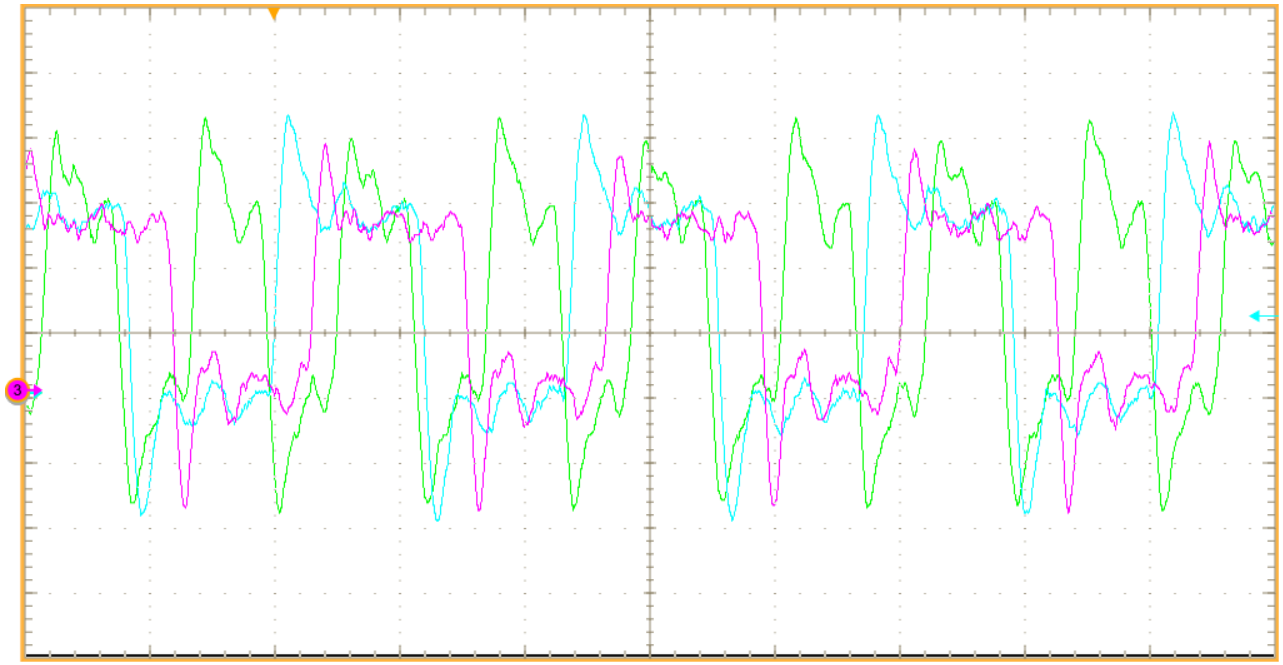
FTSW Integration Report

- Nakao-san visited Nagoya on August 5-6:
  - Provided 1 FTSW for use in our beam test.
  - Provided firmware and support in integration effort.

<u>Initial testing</u>:

- JTAG programming:
  - Verified that we can program front-end modules from FTSW through 15 m of flat CAT7 cable.
    - 1 at a time or multiple simultaneously.
    - Results sensitive to grounding ➔ FTSW and front-end modules should share a common ground.
- Simple trigger distribution:
  - FTSW module can accept a NIM trigger and pass on the raw signal through 15 m flat CAT7 cable.

Front-End Clock Requirements



Front end digitizing requires three critical clocks with specific phases:

Clock Signal	Freq (rel. to sampl. rate))	Freq (@2.7 GSa/s)	Comments
Sampling track	f_s / 128	21.17 MHz	
Sampling hold	f_s / 128	21.17 MHz	Must be low jitter
Write strobe	f_s / 64	42.3 MHz	

Front-End Clocks and FTSW

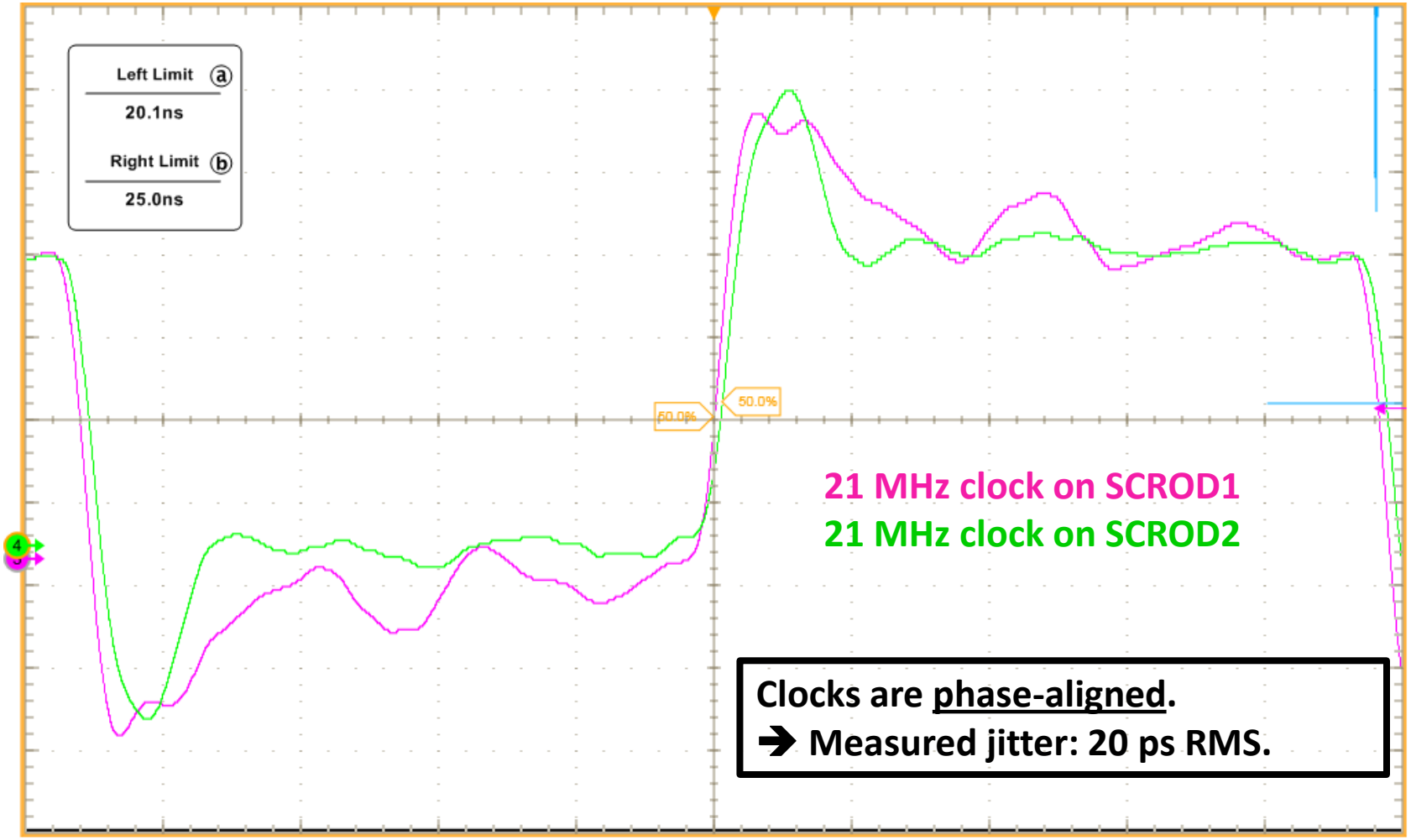
- FTSW delivers a 127 MHz clock via CAT7 cable to front-end modules.
- Front-end modules must generate these required clocks…
  - Clocks must be phase aligned for all front-end modules!

- Integrated Nakao-san's receiver firmware into front-end firmware to test jitter between the most critical clock signal between different front-end modules.

Front end digitizing requires three critical clocks with specific phases:

Clock Signal	Freq (rel. to sampl. rate))	Freq (@2.7 GSa/s)	Comments
Sampling track	f_s / 128	21.17 MHz	
Sampling hold	f_s / 128	21.17 MHz	Must be low jitter
Write strobe	f_s / 64	42.3 MHz	

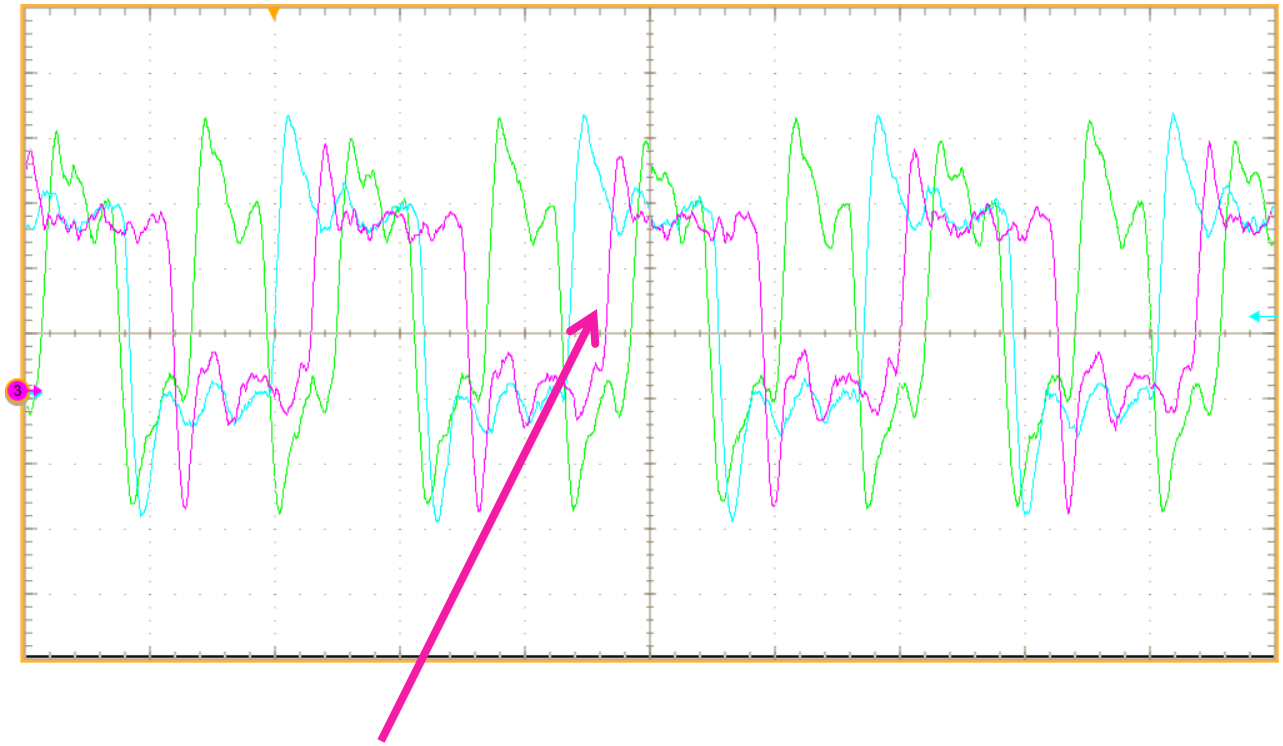
Timing Distribution Results



Backend Data Collection

- For cosmic and beam test, we must collect data from two systems:
  - Front-end modules
  - CAMAC
- Proper synchronization between events is critical!
- Strategy: veto new NIM triggers until we've read out all relevant data for each event.
  - System is set up and nominally working, with FTSW trigger distribution.
  - Currently limited to ~50 Hz instantaneous event rate (with a 4:56 spill duty cycle, ~100k events per 8 hour shift).

Timing Alignment Between Front-end and Global Trigger

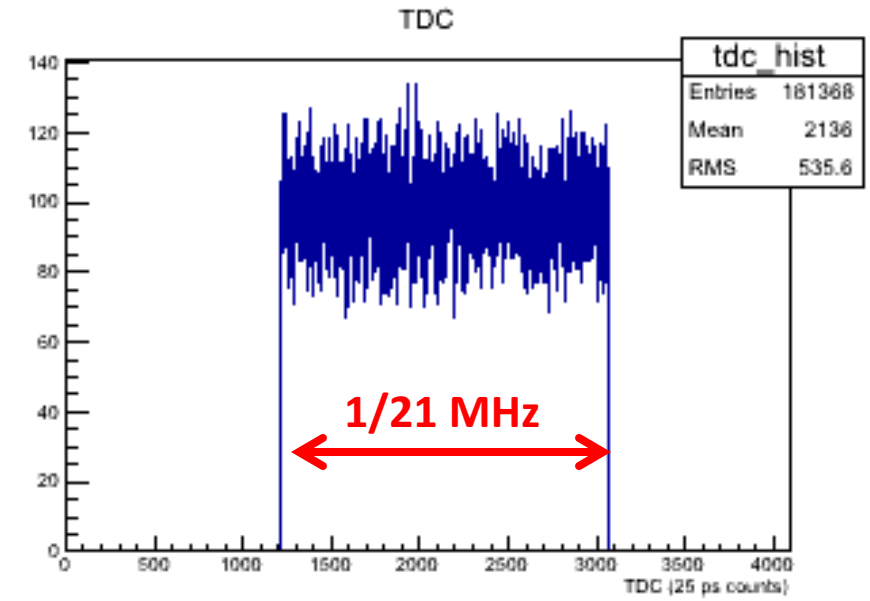


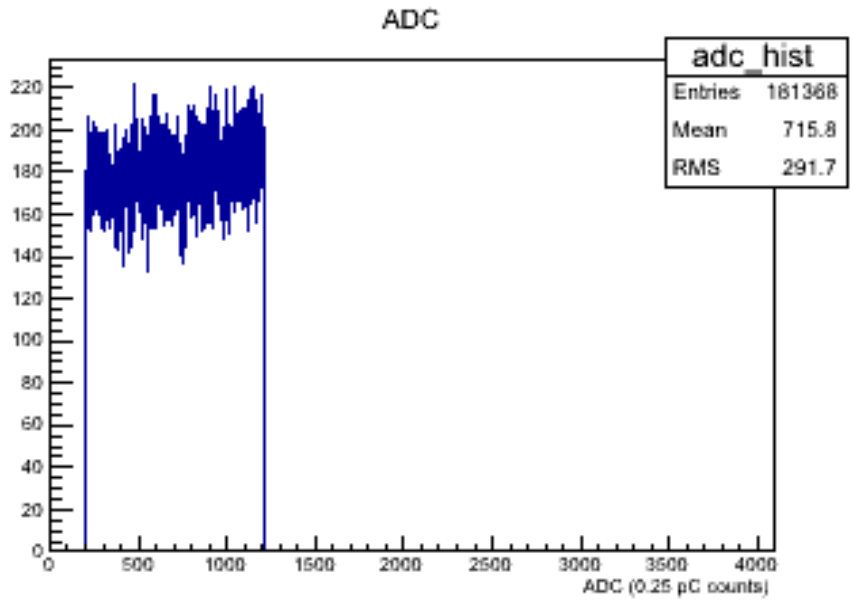
Front-end electronics measures timing relative to these clock edges…

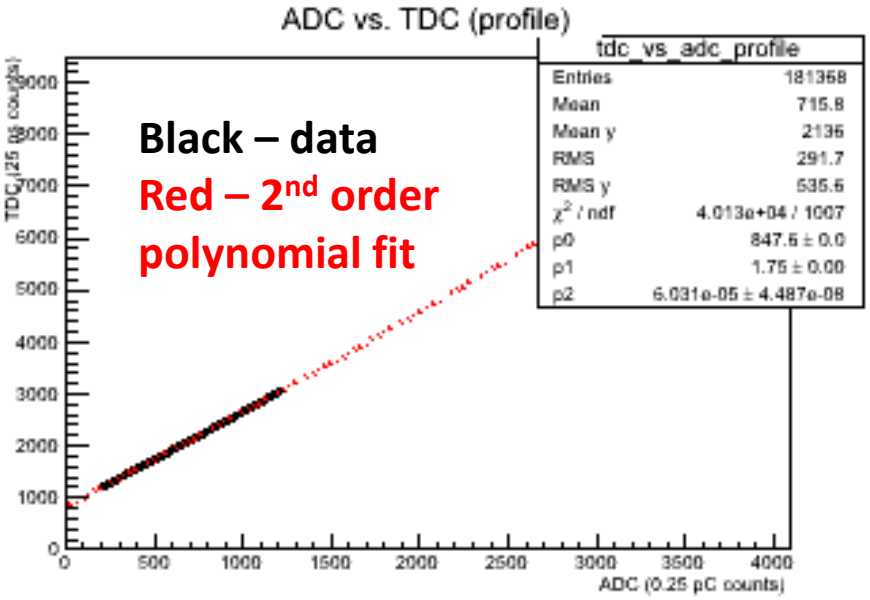
But for beam test and cosmic ray test we need to align to the global trigger.

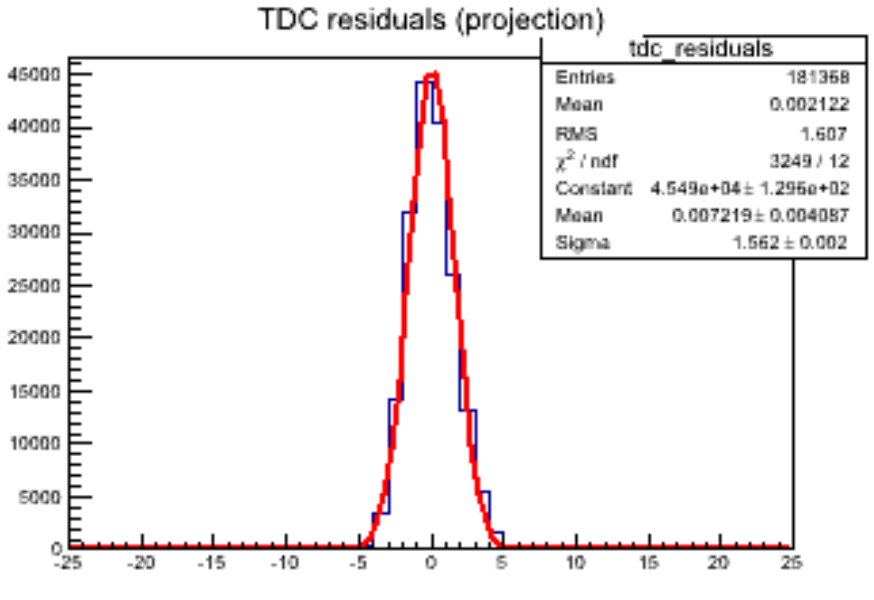
Timing Alignment w/ Global Trigger

- FTSW provides two signals that can be used for alignment:
  - A copy of the trigger synchronous with 21 MHz clock.
    - The trigger starts a TDC, this signal stops the TDC.
  - A signal that rises asynchronously with trigger and falls synchronously with 21 MHz clock.
    - This signal is sent to an ADC, which effectively measures its width.
- These TDC and ADC values are readout by CAMAC USB crate controller.
  - Measuring both will give some redundancy, possibly improve errors (or at least provide a cross check).

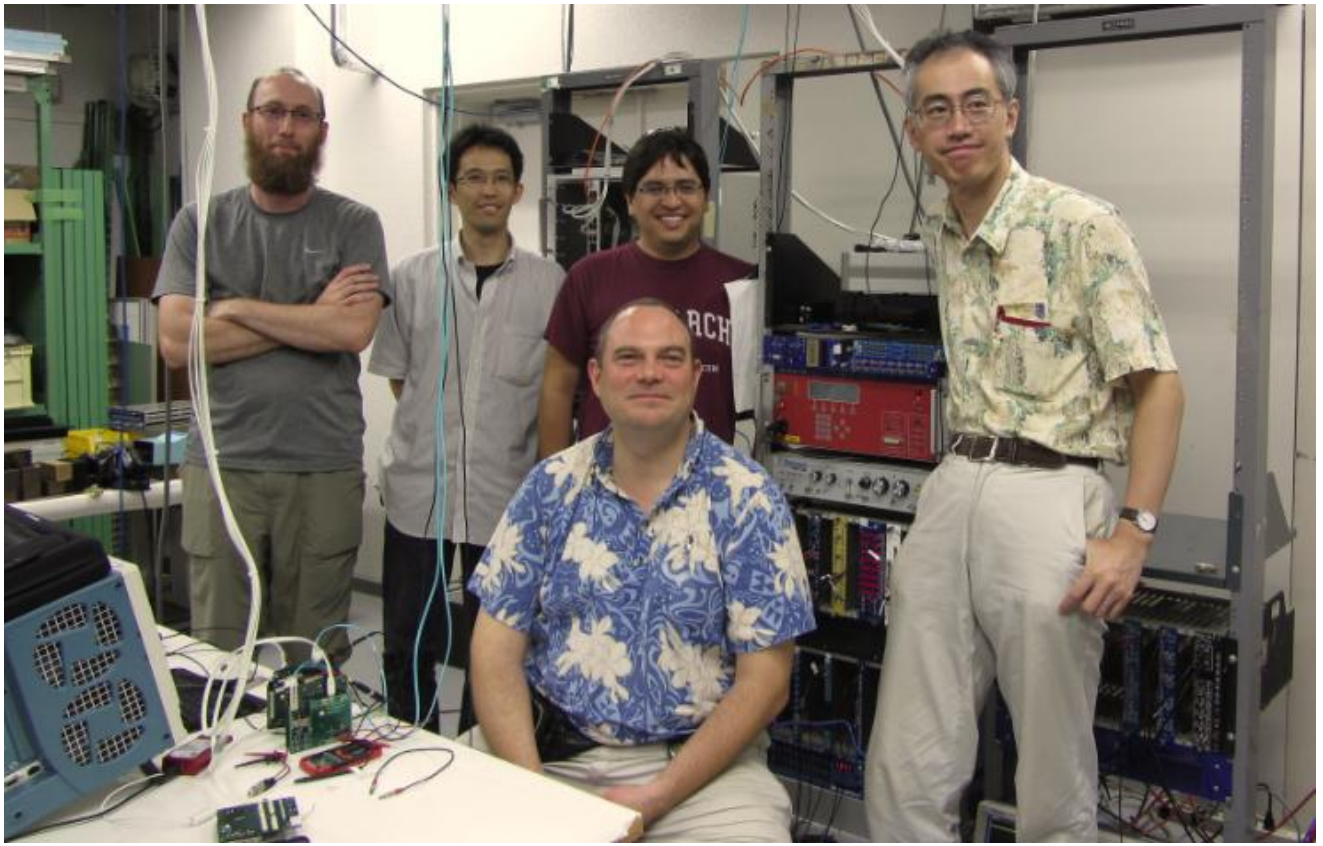

TDC
tdc_hist
Entries 181368
Mean 2136
RMS 535.6
1/21 MHz
TDC (25 ps counts)


ADC
adc_hist
Entries 181368
Mean 715.8
RMS 291.7
ADC (0.25 pC counts)


ADC vs. TDC (profile)
tdc_vs_adc_profile
Entries 181368
Mean 715.8
Mean y 2136
RMS 291.7
RMS y 535.6
χ² / ndf 4.013e+04 / 1007
p0 847.6 ± 0.0
p1 1.75 ± 0.00
p2 6.031e-05 ± 4.487e-08
Black – data
Red – 2nd order polynomial fit
ADC (0.25 pC counts)


TDC residuals (projection)
tdc_residuals
Entries 181368
Mean 0.002122
RMS 1.607
χ² / ndf 3249 / 12
Constant 4.549e+04 ± 1.295e+02
Mean 0.007219 ± 0.004087
Sigma 1.562 ± 0.002

FTSW Summary



➔ **With some care required (timing in firmware, grounding between modules, and cable quality), FTSW meets our needs for timing and trigger distribution as well as remote programming.**

Next Steps

- Integrate readout of real front-end data with fiberoptic protocol (currently done by USB).
- Verify global timing alignment scheme.
- Try measuring timing resolution between different front-end modules using MCPs.
  - Either cosmic timing counters or SL10s with laser input.

- Still lots of firmware/software to integrate and debug…
  - …but we've made lots of progress during the past week.