Scalable DAQ with Distributed Trigger*

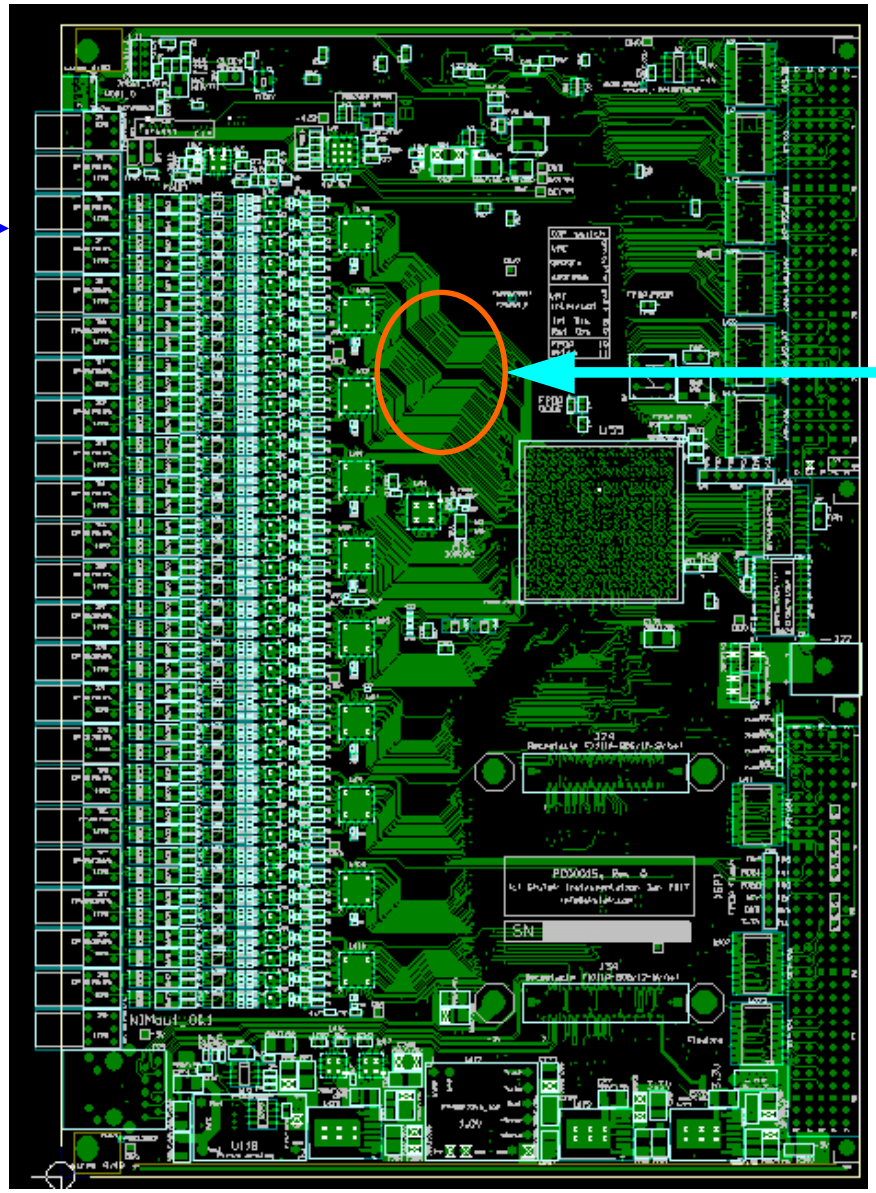
Wojtek Skulski
SkuTek Instrumentation
www.skutek.com
wojtek @ skutek.com

*Supported by the DOE Office of Science (Nuclear Physics) under the SBIR Grant DE-SC0009543

Outline

- Signal chain implementation.
- DAQ system architecture.
  - Digitizers, trigger modules, readout.
  - Time stamping and ADC clock synchronization.
- DDC-40 Digitizer with 40 channels per board (14 bits @ 100 MSPS).
- Embedded Linux System-on-Module (SOM).
- System Architecture options.
  1. VME readout.
  2. Embedded Linux with front-panel GbE readout per board.
  3. “Fast and dumb” front-panel GbE readout per board.
- R&D plans.
- Questions to the community.

Signal Chain Implementation



A

Input
BW = 10 MHz
Determined by:
preamp rise time,
detector physics,
and anti-aliasing

B

BW = ~10 GHz
A few inches

The demanding part of the design is kept local to the PCB.

40 * single channel rate

200 .. 300 Mbits/s
Gigabit Ethernet

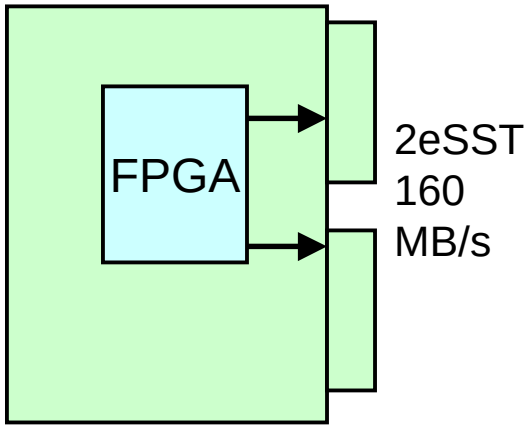
C

C

Up to 160 MB/s
VME backplane

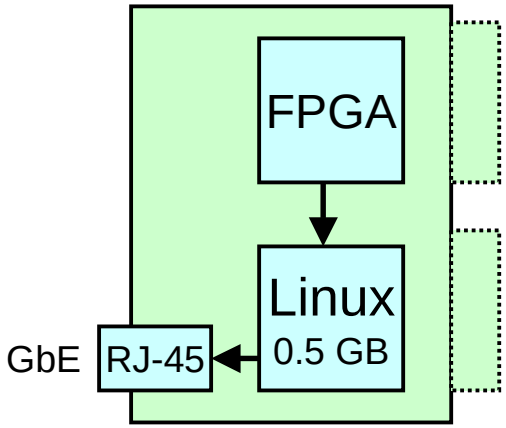
Readout Options

Option A VME



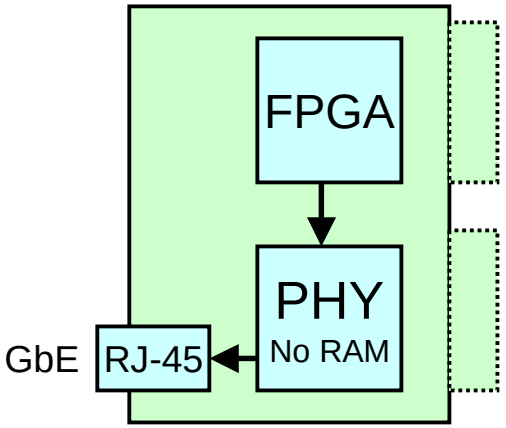
- Up to 160 MB/s with 2eSST (shared among modules)
- Single data stream per crate Convenient.
- Offline event building not required
- VME used for both power and readout

Option B Linux+GbE



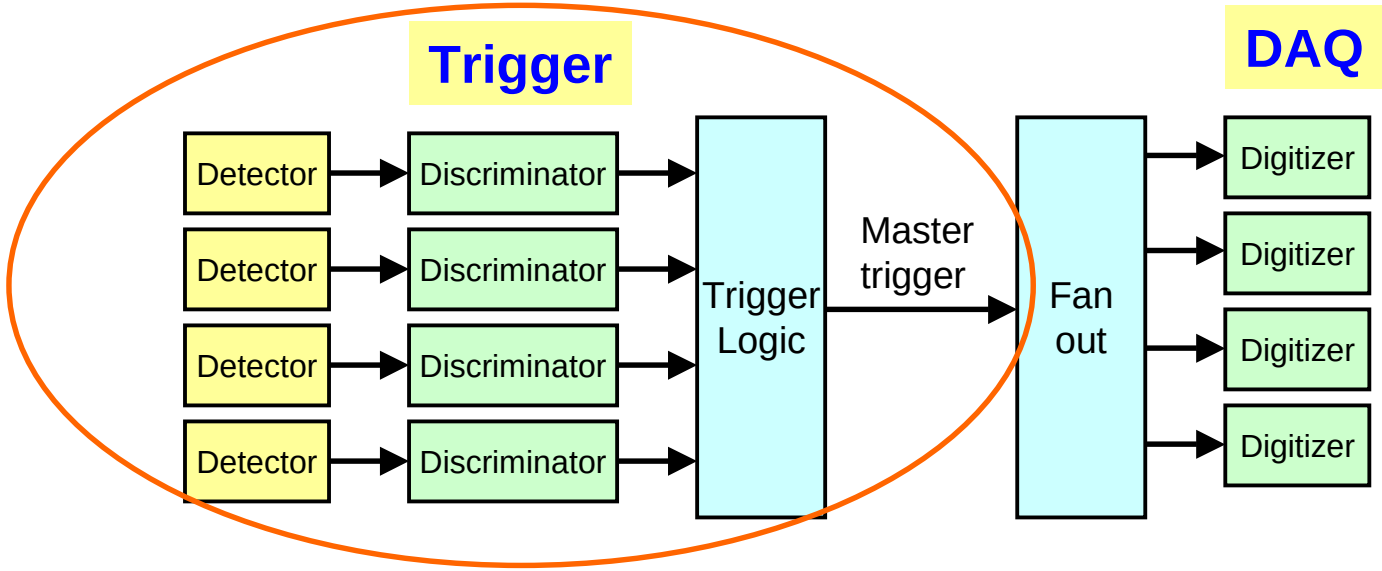
- ~30 MB/s per module Dedicated (not shared among modules)
- TCP/IP, UDP, RAW
- On-board event buffering, monitoring, histogramming.
- Offline event building is required
- VME used only for power

Option C "Dumb & Fast" GbE

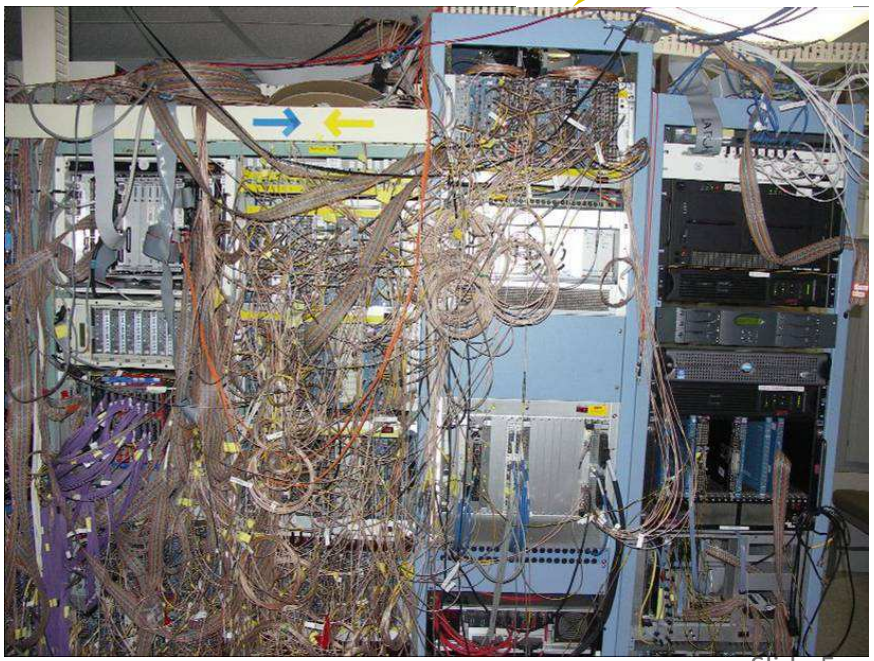


- Up to ~109 MB/s per module Dedicated (not shared among modules)
- UDP or RAW, but no TCP/IP
- No on-board monitoring (because no CPU)
- Lack of extensive buffering
- Offline event building is required
- VME used only for power

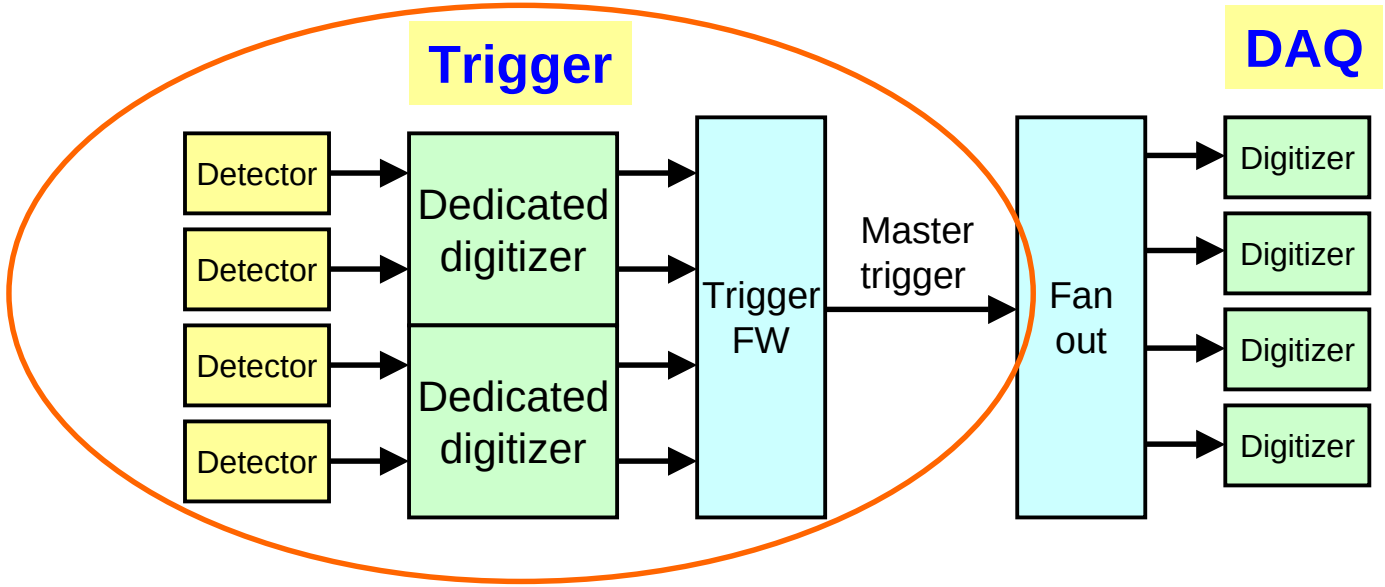
Trigger Options: Traditional Analog Trigger



- Trigger signals:
  - Either dedicated detectors, or
  - signals branched from regular detectors (“teed”).
- Discriminators are “*single-bit ADCs”.*
  - *Signal > threshold --> bit = 1.*
- Trigger logic assembled with NIM electronics.
  - Fast response (good).
  - FPGA programming not needed (good).
  - Substantial setup can be difficult to maintain.

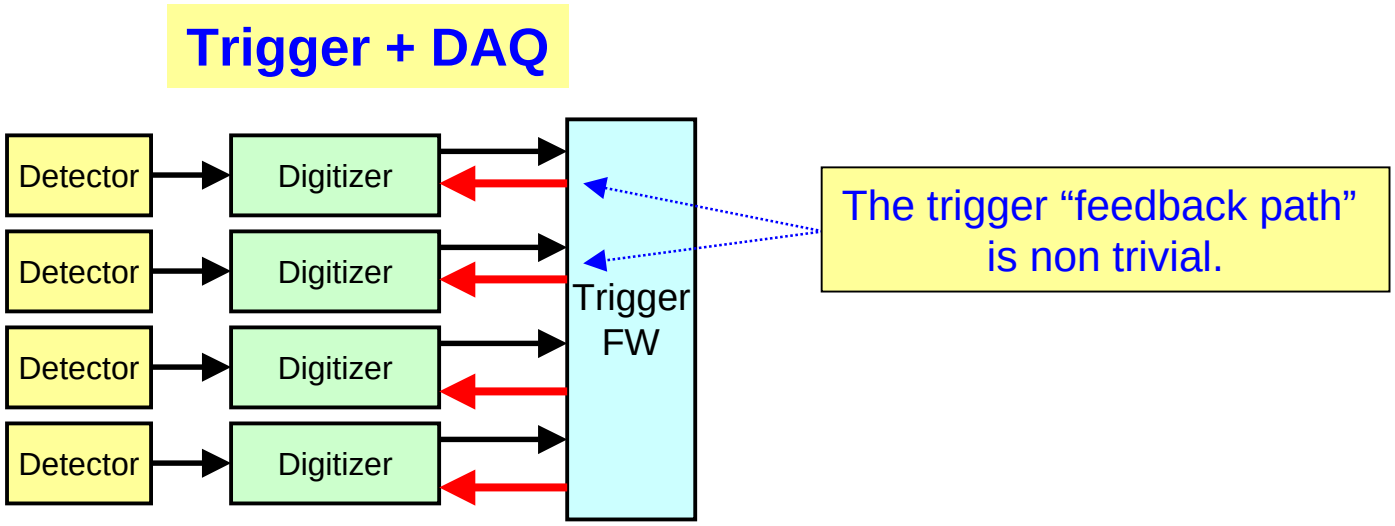


Trigger Options: Dedicated Digital Trigger



- Similar to traditional, but different implementation.
- Trigger signals similar to previous case:
  - Either dedicated detectors, or
  - signals branched from regular detectors (“teed”).
- Discriminators are implemented in FPGA firmware.
- Trigger logic assembled in FPGA firmware.
  - FPGA programming cannot be avoided.

Trigger Options: Integrated Digital Trigger

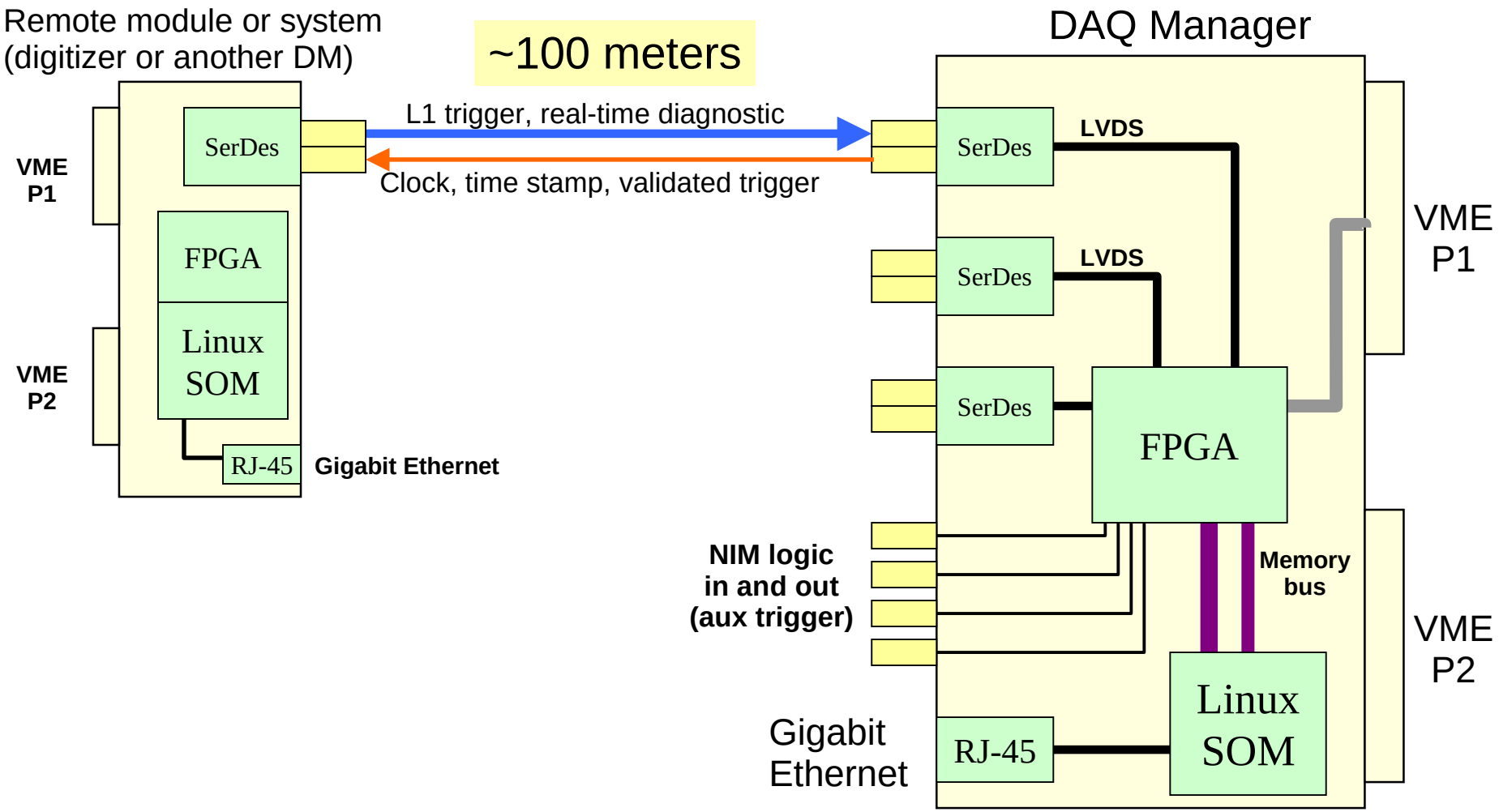


- Trigger becomes an integral part of the DAQ.
- Trigger signals can be branched off from DAQ signals inside DAQ.
- Discriminators & logic are implemented in DAQ firmware.
  - Integrated trigger logic is **highly non trivial**.
  - FPGA programming has to be mastered. (Any problem with that?)
- “Trigger signals” are the same as main signals.

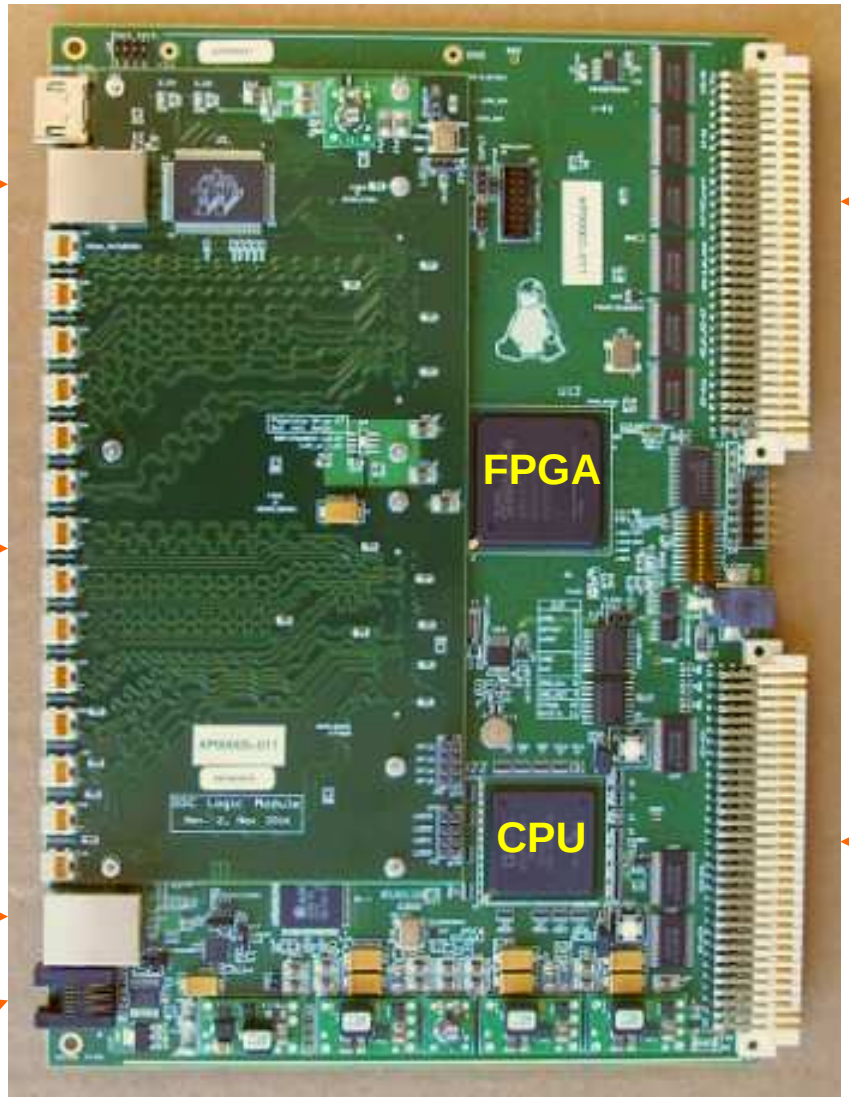
  Trigger information is automatically recorded in the DAQ stream.

Sketch of the Target Architecture

Integrated DAQ + Trigger



Trigger Implementation: DAQ Manager Prototype

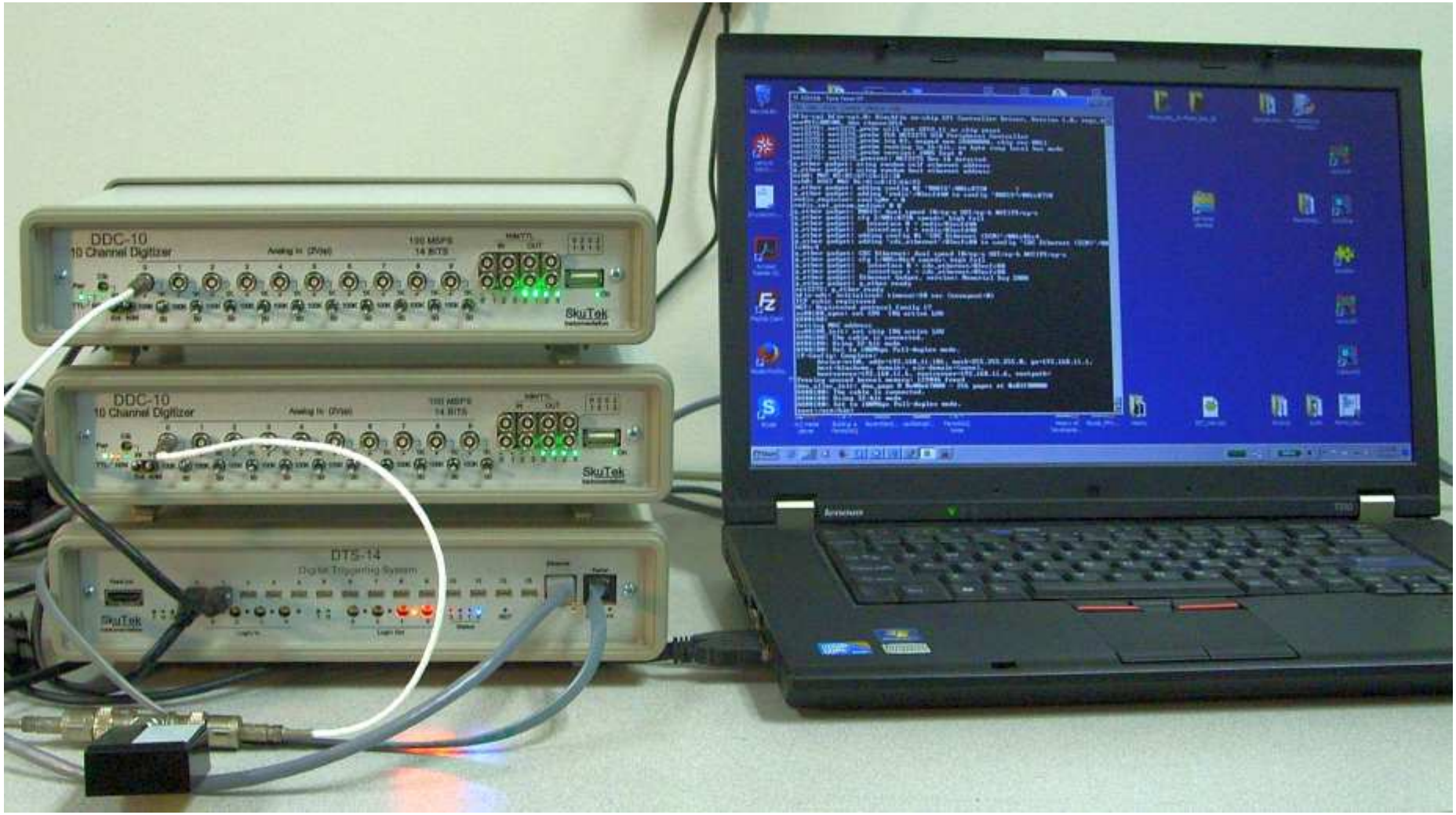


VME is optional.
DM can be mounted in a box.

System Prototype w/o VME

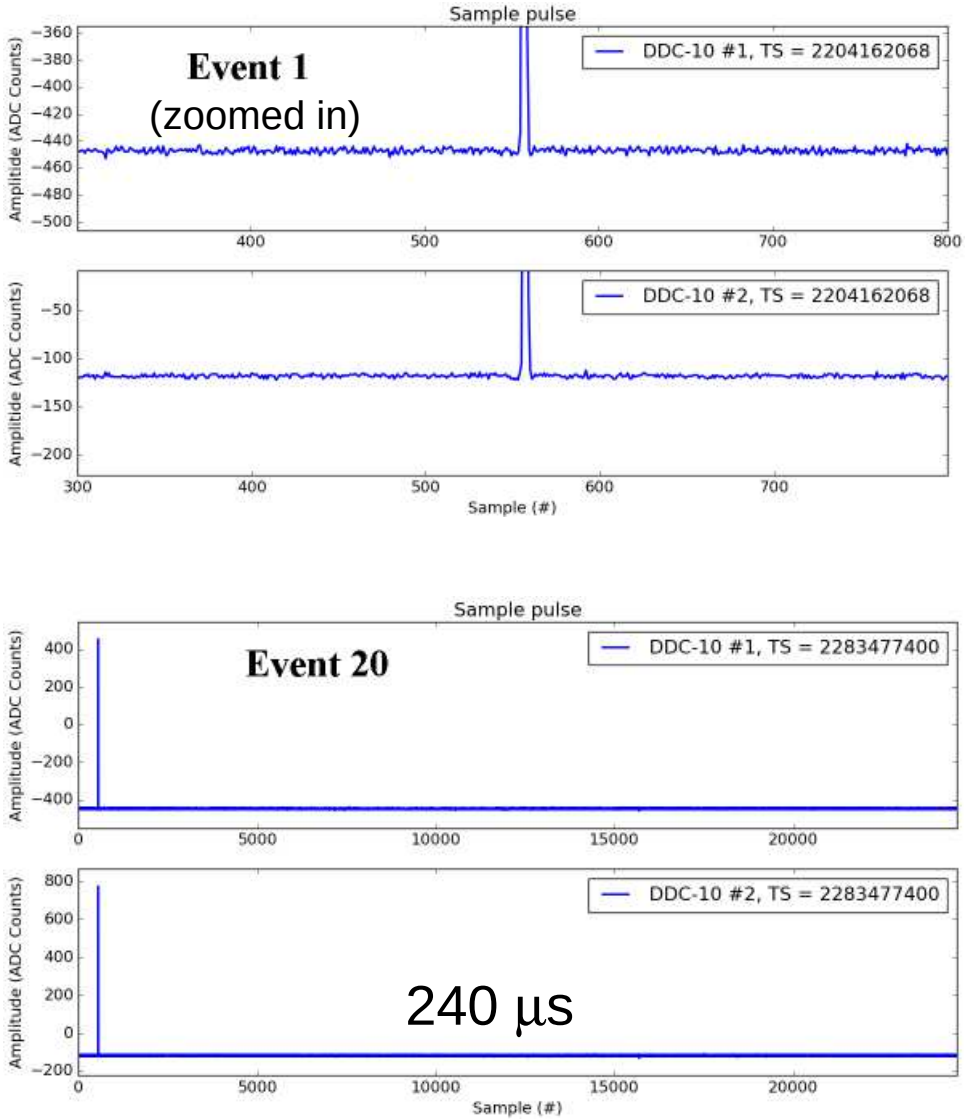
Two 10-channel Digitizers + DAQ Manager

Both digitizers write separate binary event files with time stamped waveforms

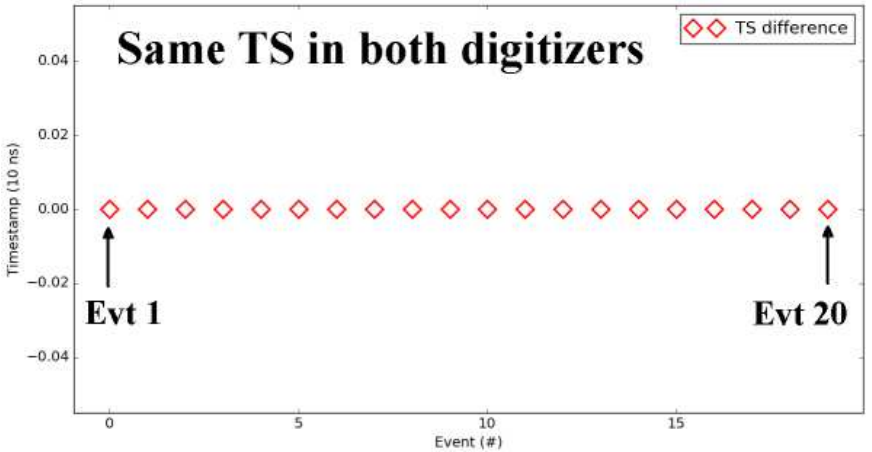


Time Stamp and ADC Clock Synchronization

Time stamps and ADC clock synchronized in both digitizers and DAQ Manager

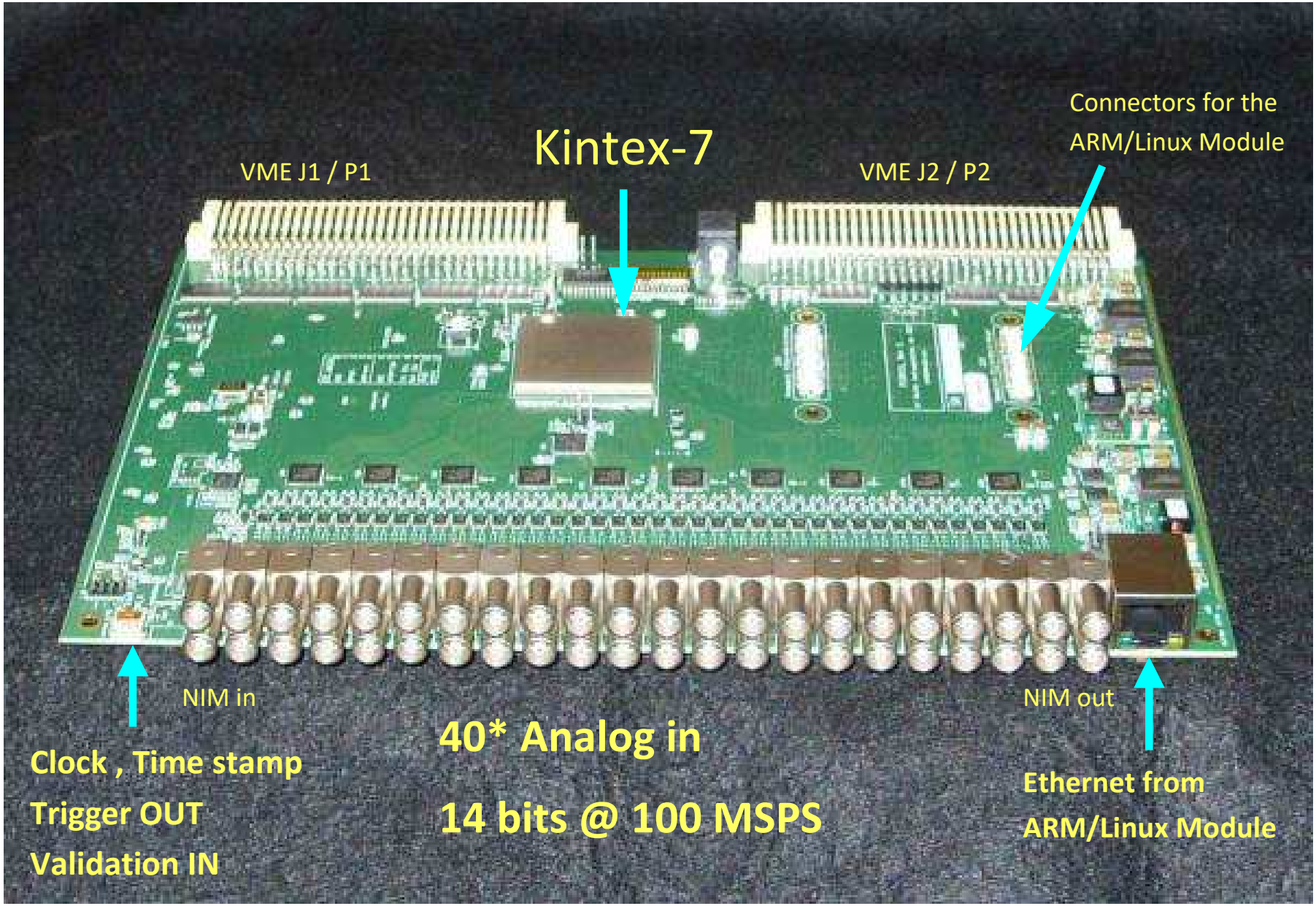


TS granularity = 10 ns
All three modules in synch



Time span = 79.315 ms

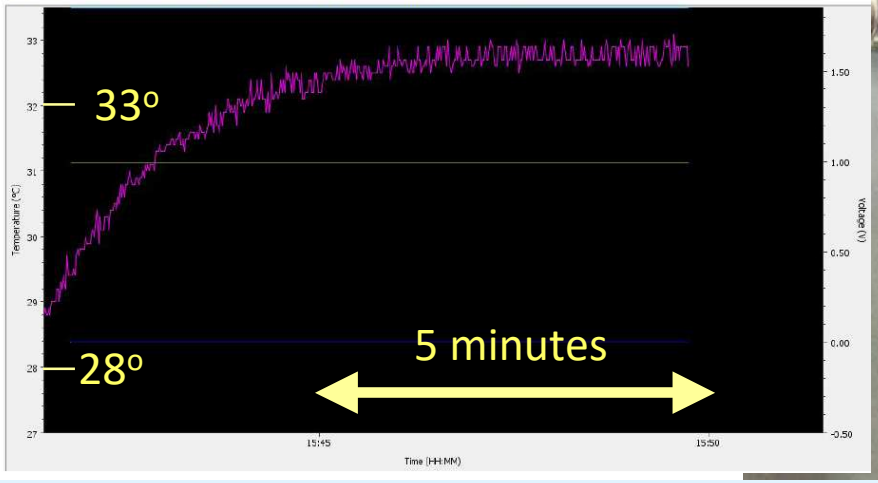
The 40-Channel Digitizer With Kintex-7



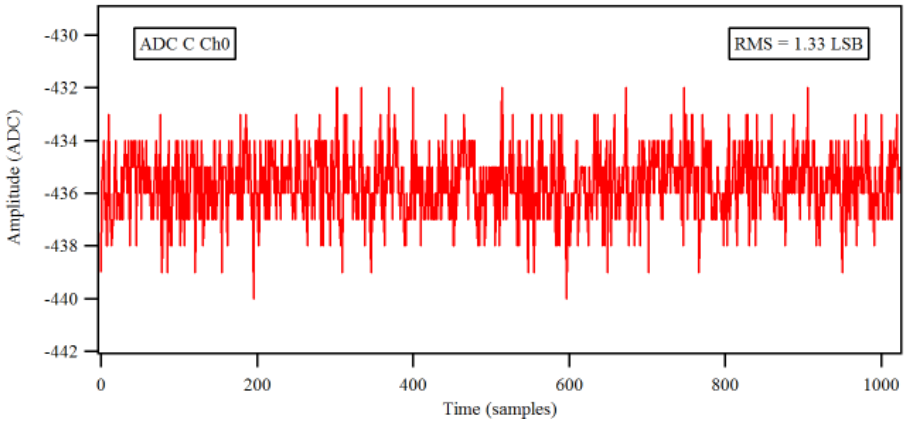
The DDC-40 Digitizer in the VME Crate

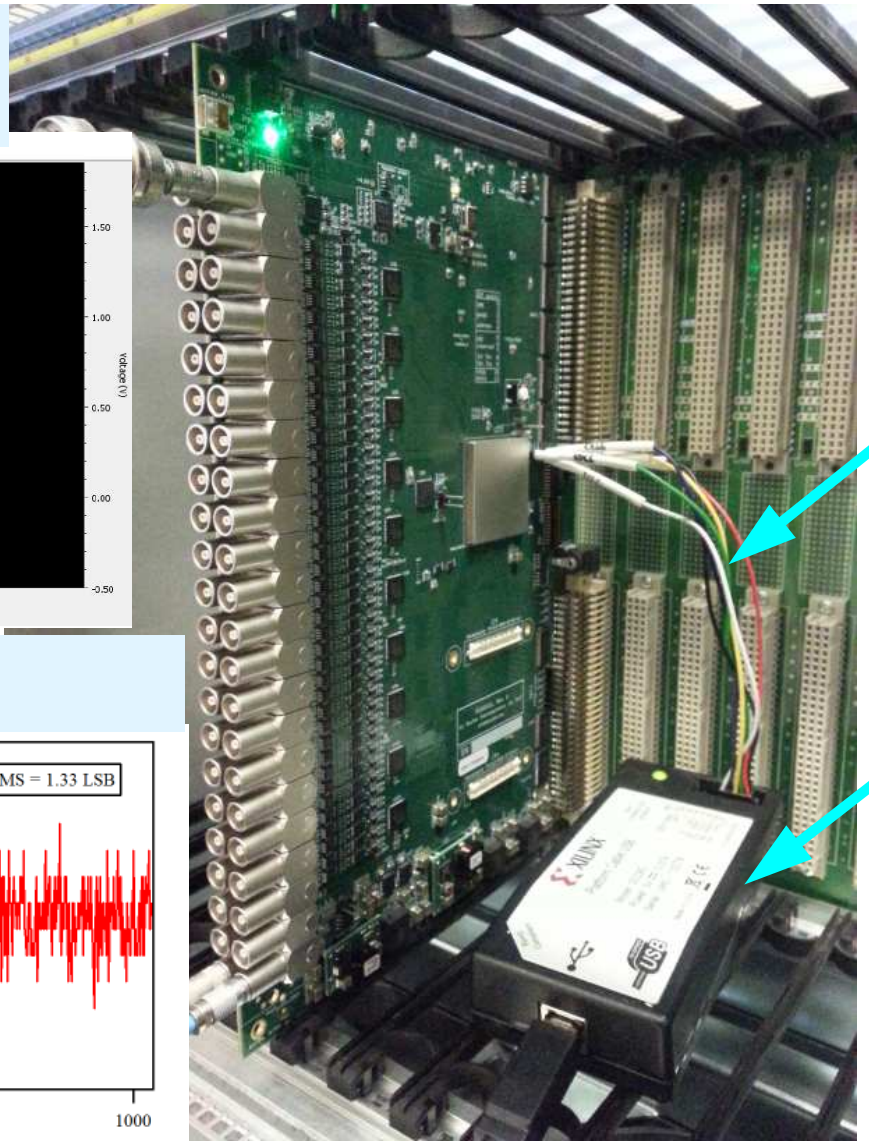
One board **is here at the meeting**, the 2nd prototype is now being tested.

Temperature of the FPGA measured with the internal FPGA sensor.



A typical waveform





JTAG Cable

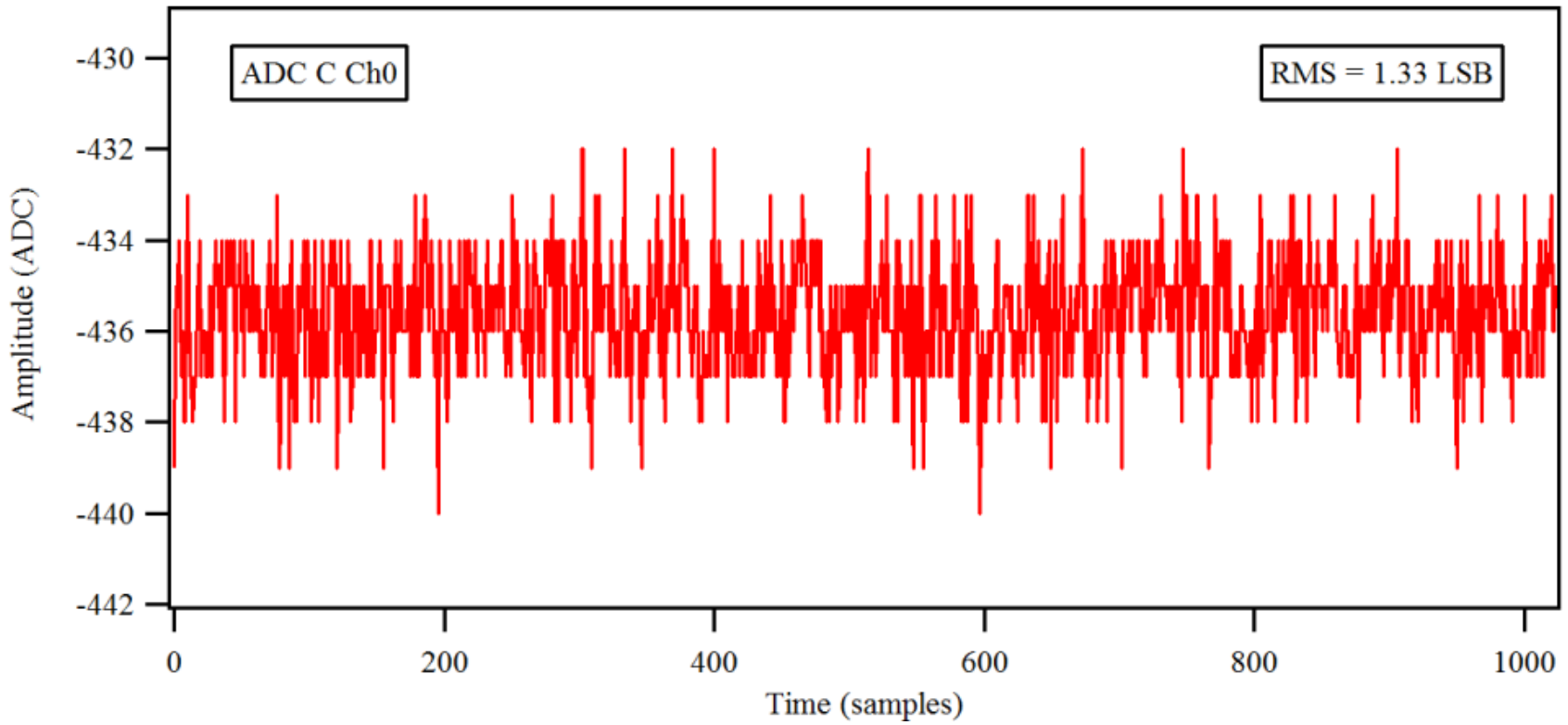
Xilinx JTAG interface

A Typical Waveform

- Excellent baseline stability.
- Low RMS noise: 0.162 mV = 1.33 LSB

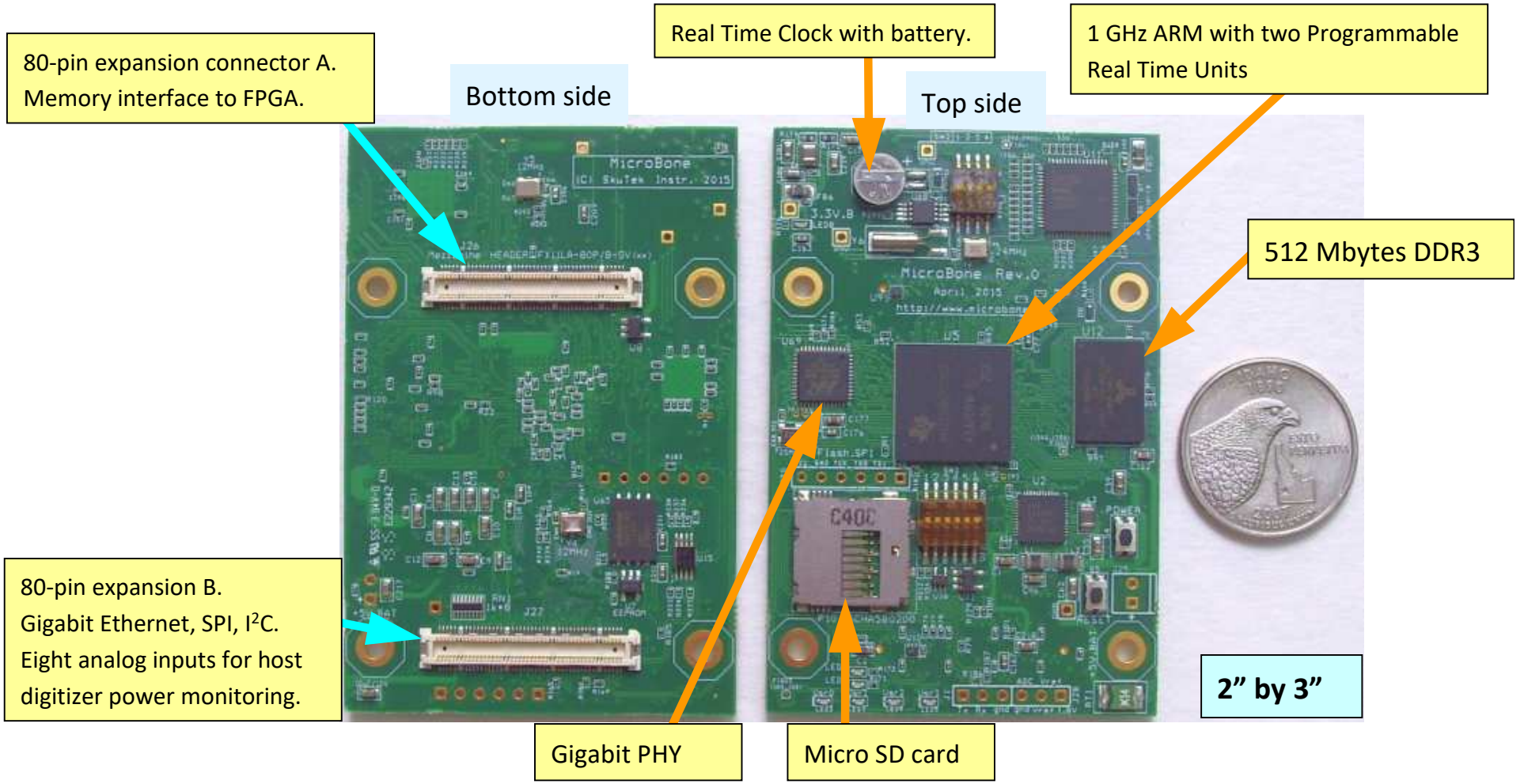
  1 LSB = 2000 mV / 16,384 = 0.122 mV
- Waveform time span: **222.5** µs or **397.5** µs, depending on the FPGA size.

ADC chip spec is 1.0 LSB



Embedded Linux System-On-Module

- Optional 1 GHz ARM **workstation** running Linux plugs into the Digitizer.
- It can provide **real time, instantaneous** histogramming, diagnostics, and monitoring.
- Do you want to know if detectors work as expected? Monitor them *in situ!*



Linux SOM Mounted Inside 10-Channel Digitizer

- System-on-Module running Linux is setting up the FPGA and then reading data.
- The control software was written in C and Python. (Specifically: in Jupyter.)
- The FPGA is mapped as memory in the Linux address space. Easy for the programmer.

This digitizer was used during the test.

Linux SOM

R&D Plans

- Increasing the reach of the Fast Trigger Link from a few meters to ~hundred meters.
- New, more powerful DAQ Manager Module.
- Turnkey firmware and software “of reasonable complexity”.

  We plan to develop useful FW/SW tools for the users.

- <u>System Architecture options</u>.

  1. Classic VME readout.
     a) DAQ systems with up to 40*20 = 800 channels per crate.

        Advantage: Event building is automatic.
  2. Embedded Linux SOM with GbE readout per board.

        Advantage: Convenient monitoring & control.

     b) Crates “full of modules” with GbE cables, one from each digitizer.
     c) Standalone “DAQ in a box” with Ethernet interface.
  3. “Fast and dumb” GbE readout per board: Full GbE speed, but UDP only.

        Advantage: Similar to the previous, but 3x faster.

        But… Convenient monitoring not provided.

Acknowledgments

Contributions by Eryk Druszkiewicz and David Hunter are gratefully acknowledged.

Questions To the Community

- Which of the System Architecture options do you prefer?
  1. Classic VME?
     - Low level drivers, controllers, registers, hex addresses, ough.
  2. Embedded Linux? Each board is like a workstation.
     - Do you need Linux tools and applications *embedded* in each module?
     - Do you need in situ diagnostic and monitoring? (E.g., histogramming in situ.)
  3. "Fast and dumb" GbE readout per board:
     - Full GbE speed, but only UDP.
     - Do you need fast readout?
     - All the burden is at the receiving end.
     - We send you UDP packets, you receive them and you do the rest. OK?

Please tell us what you need!