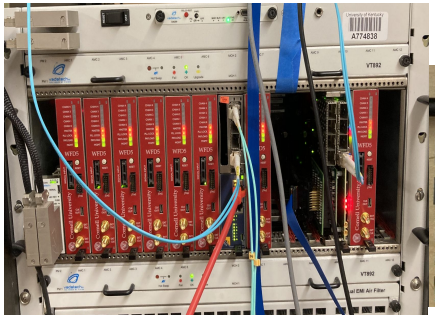
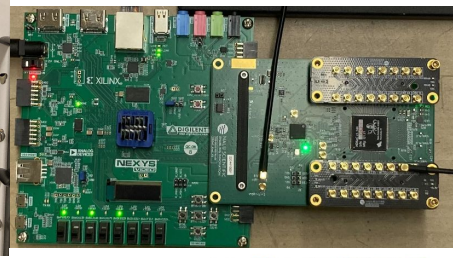
PIONEER DAQ Development

Jack Carlton
University of Kentucky
January 7th, 2025

The Four DAQs in this Talk

1. g-2 modified (Calo Teststand) DAQ
   - Uses g-2 hardware
   - Used for calorimeters tests
2. HDSoC (Nalu) DAQ
   - Uses Nalu Scientific's HDSoC FMC board
   - Used for ATAR digitization
3. PCIe based (PIONEER) DAQ
   - Will use Cornell designed hardware
   - UKY testing with development FPGA boards
   - Will be used for PIONEER DAQ
4. Belle II PCIe Based DAQ
   - Uses PCIe readout of FPGA boards
   - Used in Belle II experiment
   - Useful design to piggyback off of for PIONEER



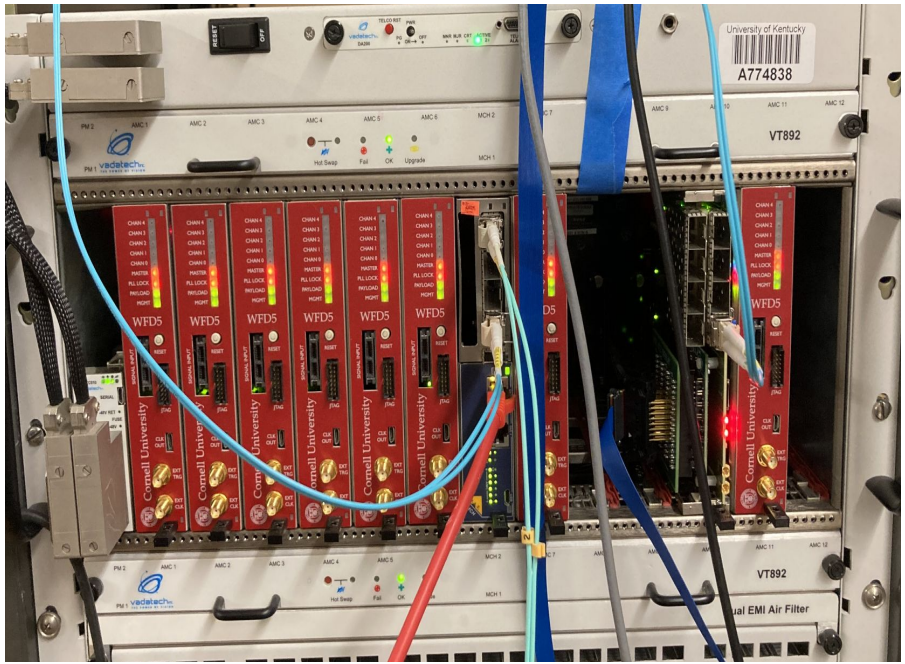


Some hardware from each DAQ system mentioned

Top left: g-2 modified electronics crate
Top right: Nalu's HDSoC FMC on Nexys A7
Bottom left: PCIe based test board
Bottom right: PCIe40 board

g-2 modified (Calo Teststand) DAQ - Updates

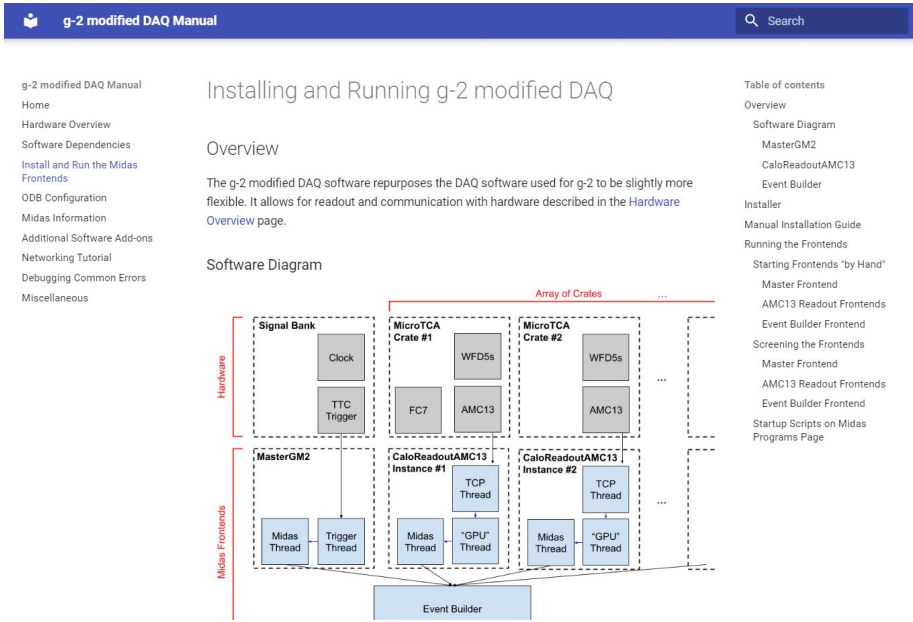
- Updated operating system for the project to ALMA9
  - CentOS7 reached EOL
- Added features
  - Multiple crates
  - WFD5 self triggering mode
  - Laser study parameters added to ODB
- Improved rate capabilities
  - Removed meinberg dependency
  - Tested ~10kHz at UKY
- Some quality of life scripts added
- Updated midas to a 2024 build



µTCA crate with WFD5s, AMC13, FC7, and MCH

g-2 modified (Calo Teststand) DAQ - Documentation

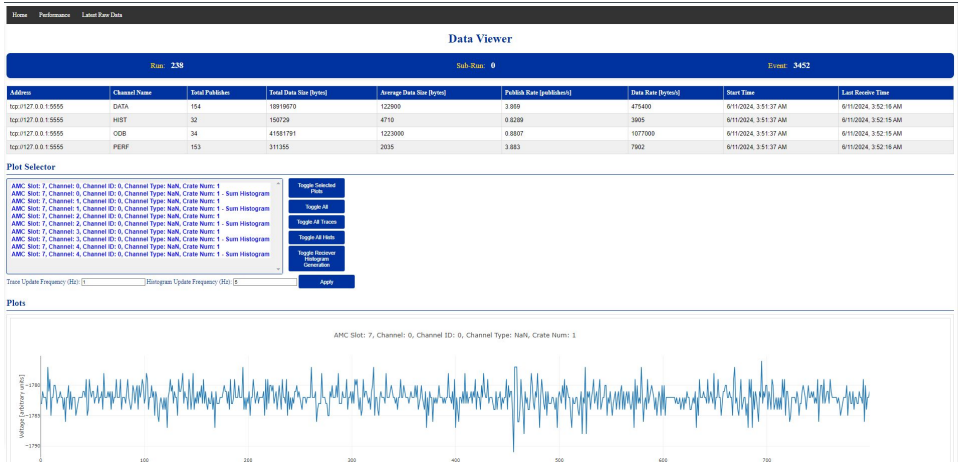
- Setup of the teststand DAQ is not straightforward
  - Custom software and hardware
  - Specific software and hardware configurations
- Created documentation to aid setup and configuration
  - Website version on github pages
  - Includes explanation of relevant ODB parameters
  - Living document (easy to update)



A page from the manual webpage

g-2 modified (Calo Teststand) DAQ - Software Add-ons

- Status webpage
  - Timing monitoring
  - Data quality Monitoring
  - Crate content status page revived (separate webpage)
- More event level info
  - System resource monitoring
- To do
  - Optimize Event Publisher
    - Integrate my midas_receiver library, created as per a suggestion on the midas forums

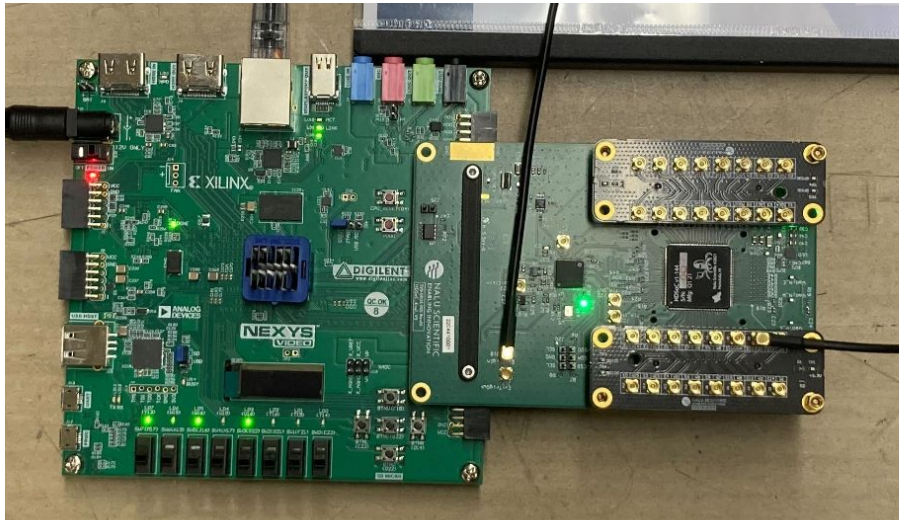


“Generalized” Teststand DAQ DQM Webpage

Integrating HDSoC into Midas (naludaq) - Current Status

- Can control board via midas
  - Initialize board
  - Configure (external) trigger and channel settings
  - Begin and end collections

- Readout is currently unprocessed UDP packets over 1GbE

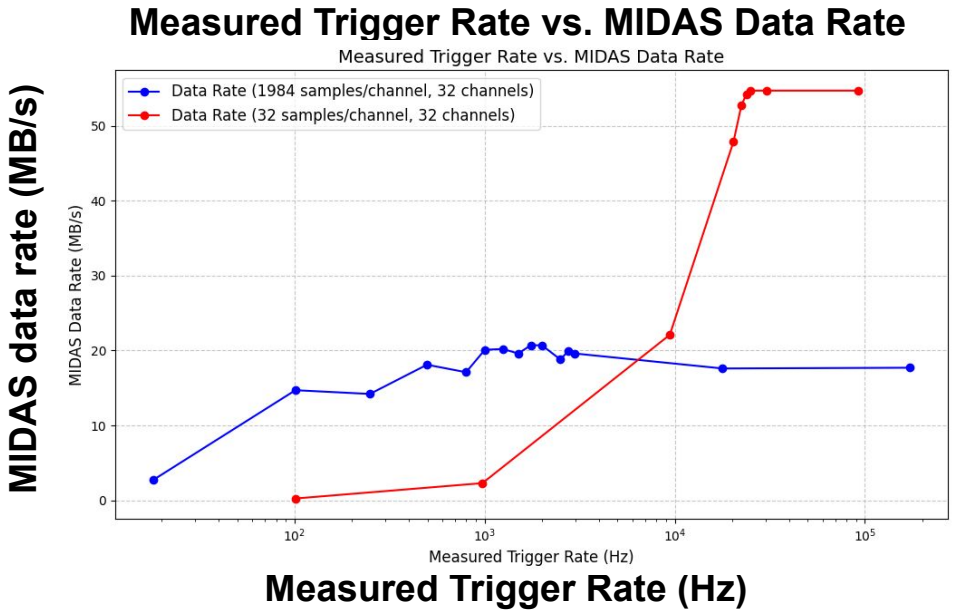


Nalu's HDSoC FMC attached to a Nexys A7 Video Card

Integrating HDSoC into Midas (naludaq) - Rates

- Highest data rate achieved is **~55 MB/s** through 32 channels at ~20kHz trigger rate
  - Slower in practice, no event building yet

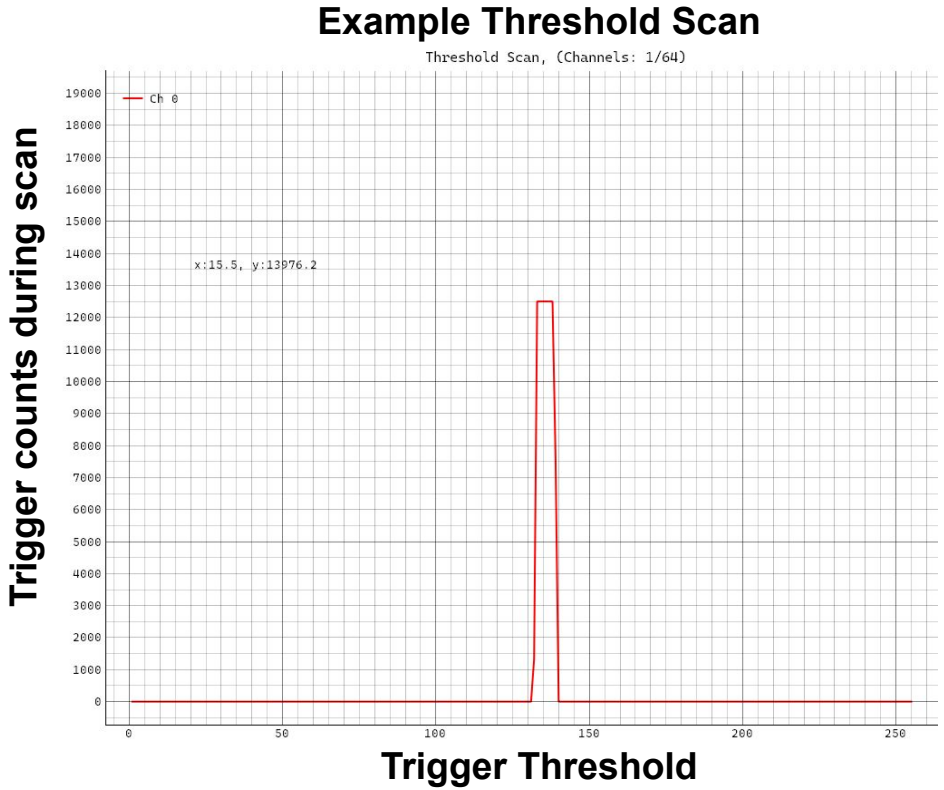
- Data rate dependencies
  - Trigger rate
  - Number of channels
  - “Island” size
    - Affects max data rate



Integrating HDSoC into Midas (naludaq) - Next Steps

- Incorporate features from NaluScope into Midas:
  - Construct events from UDP data
  - Configuring internal trigger settings
  - Threshold scan
  - Pedestal scan and subtraction

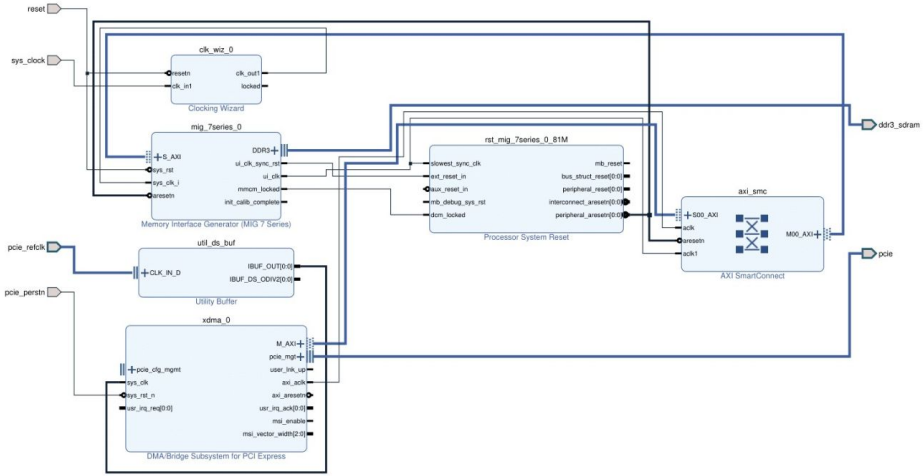
- Marcus Luck (Nalu Software Head) has been helping



PCIe based (PIONEER) DAQ - Status

- Using Nereid Development board
  - Kintex-7 FPGA
  - Max throughput 2 GB/s over PCIe
- Firmware using Xilinx intellectual property (IP) blocks in Vivado
  - Creates DMA link between onboard RAM and host (desktop) RAM
- Have C++ library for readout
  - Effectively a wrapper around Xilinx XDMA driver
- Integrated C++ library in a midas frontend for rate testing

Nereid K7 PCI Express FPGA Development Board



Block diagram for DMA transfer between board RAM and host (desktop) RAM

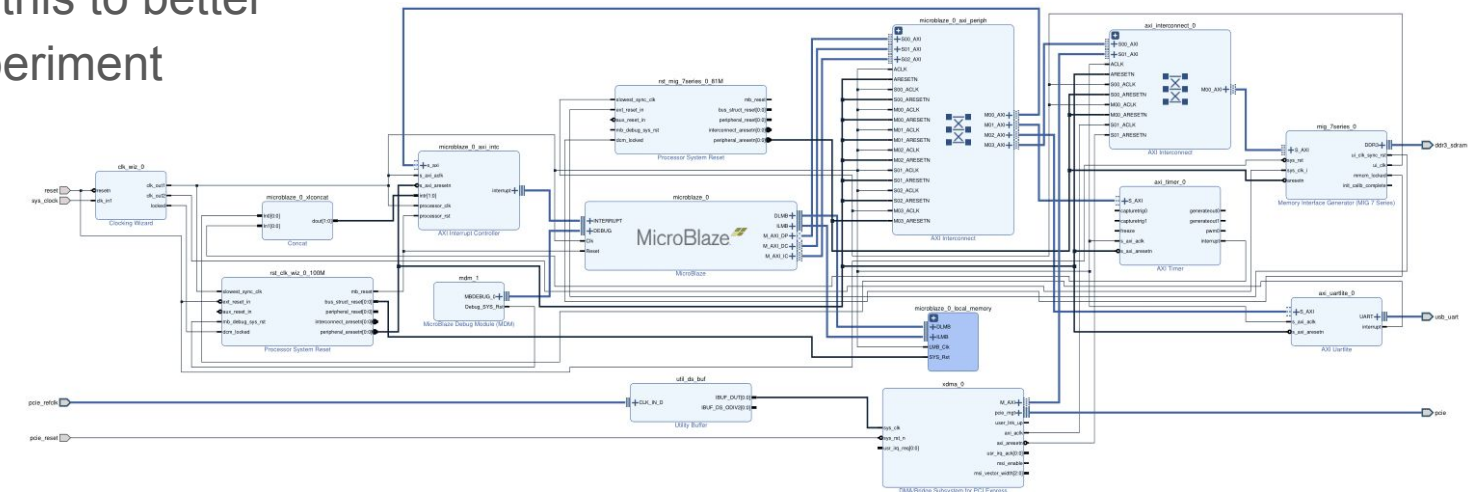
PCIe based (PIONEER) DAQ - Rates

- More interested in read/Card-to-host (c2h) transfer rates
- Transfer rates are faster for larger data transfer sizes
- Highest throughput in MIDAS was **~1.2 GB/s**
  - Using two c2h channels
- 1.2 GB/s is largely limited by nereid board hardware

DMA transfer rate vs transfer size over one channel using custom C++ Library

PCIe based (PIONEER) DAQ - Next Steps

- Added MicroBlaze IP Block
- Allows the FPGA to run C++ code to edit onboard DDR3 RAM
  - Can code data generation simulators
- Want to use this to better simulate experiment



Block diagram for PCIe DMA transfer with microblaze

Belle II PCIe Based DAQ - Overview

- Belle II is an experiment studying B mesons at SuperKEKB in Japan
  - Similar data rate needs to PIONEER
  - Recently (~2020) upgraded their DAQ system to be PCIe based

- PCIe based upgrade involved a “PCIe40” FPGA board
  - Similar to UKY test FPGA boards like [numato’s nereid development board]

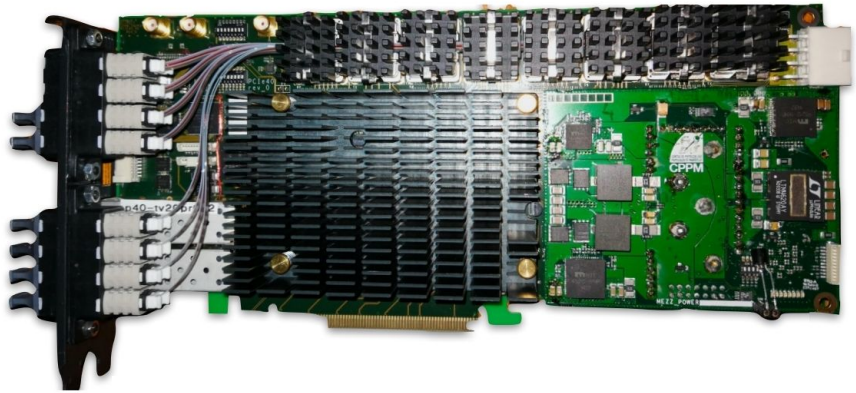
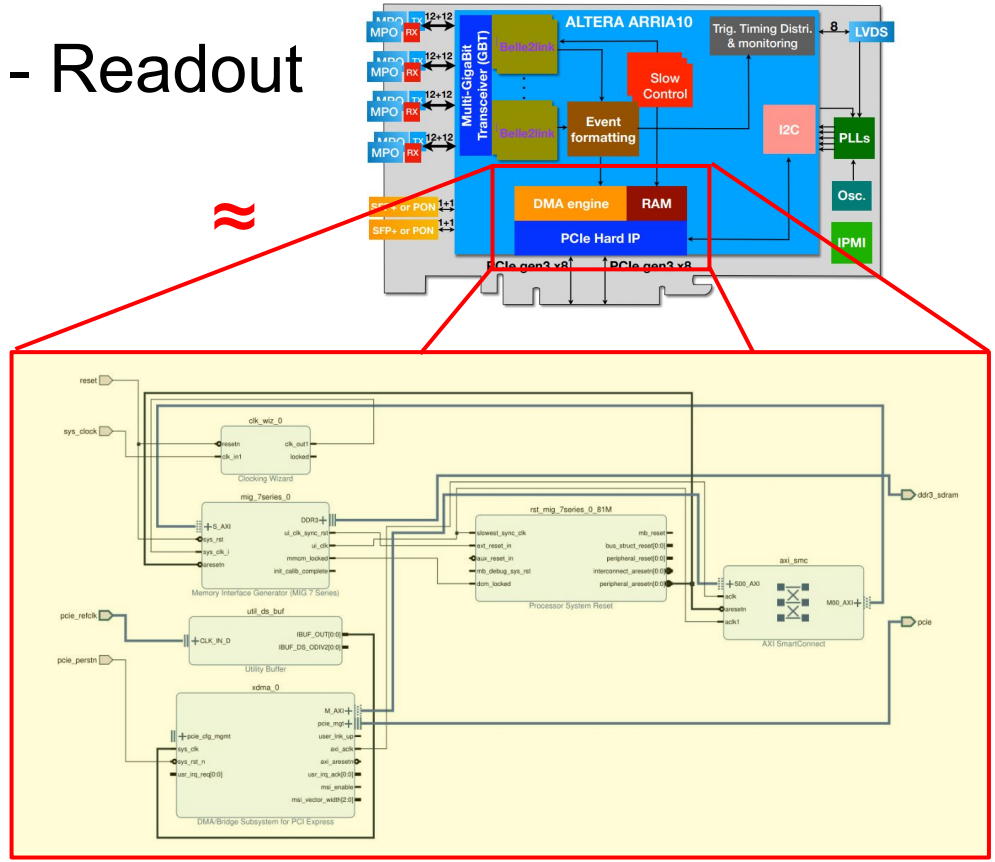


Photo and block diagram of the PCIe40 board

Belle II PCIe Based DAQ - Readout

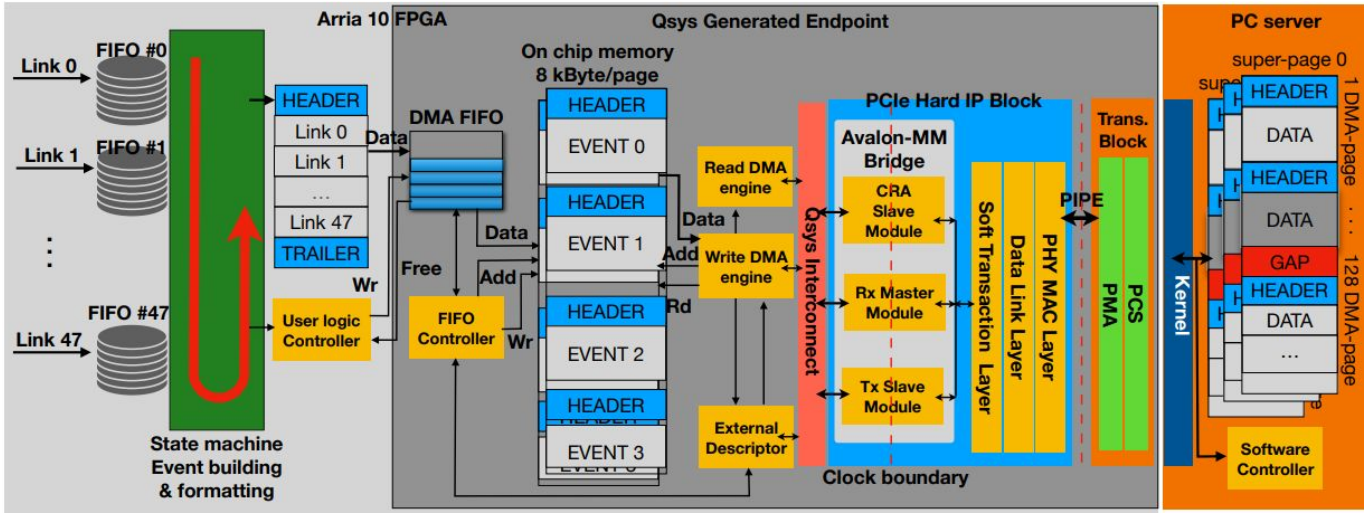
- Belle II design uses DMA engine to move data out of RAM buffers
  - Use PCIe Hard IP (Intel)
  - Use custom written drivers
- Doing very similar at UKY
  - Xilinx PCIe DMA engine facilitates data transfer between card and host
  - Using XIlinx XDMA driver



UKY block diagram for PCIe based DMA is similar to the Belle II PCIe readout system

Belle II PCIe Based DAQ - Control and Event Processing

- FPGA handles event building from multiple sources (links)
  - Events constructed before PCIe transfer
- Their “Qsys Generated Endpoint” is similar to the Xilinx IP blocks in the last slide
- Allows user control of low level components via DMA transfer over PCIe

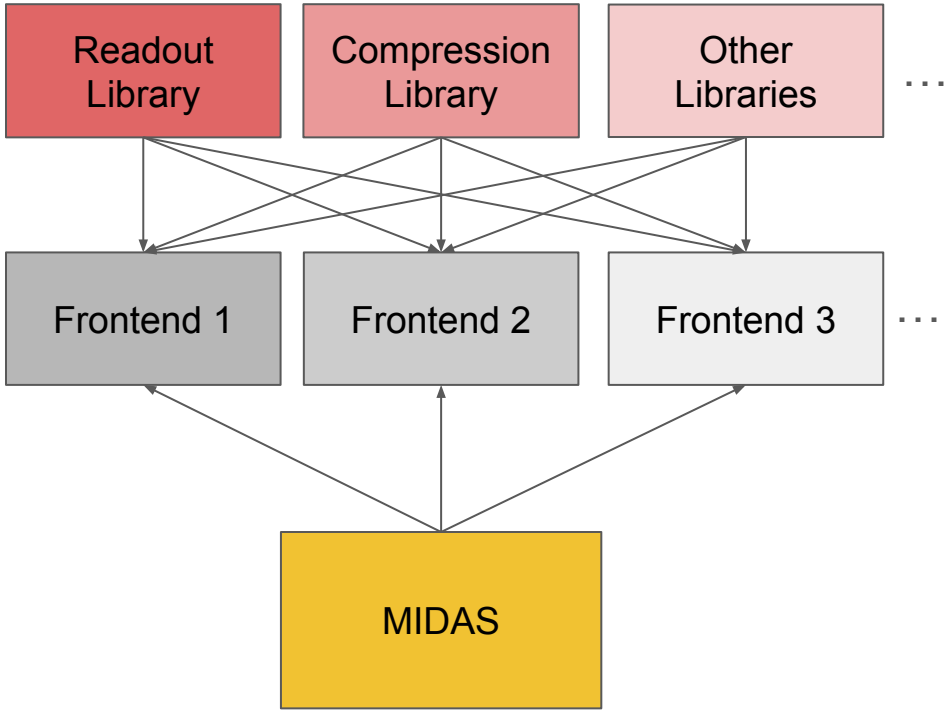


Auxiliary Slides

j.carlton@uky.edu

Software Philosophy

- Write modular software
  - Will make experiment DAQ code much more manageable in the future
- Optimize and adjust readout, compression, and other libraries (as needed)
- Write simple and scalable midas frontends
  - Implement libraries above

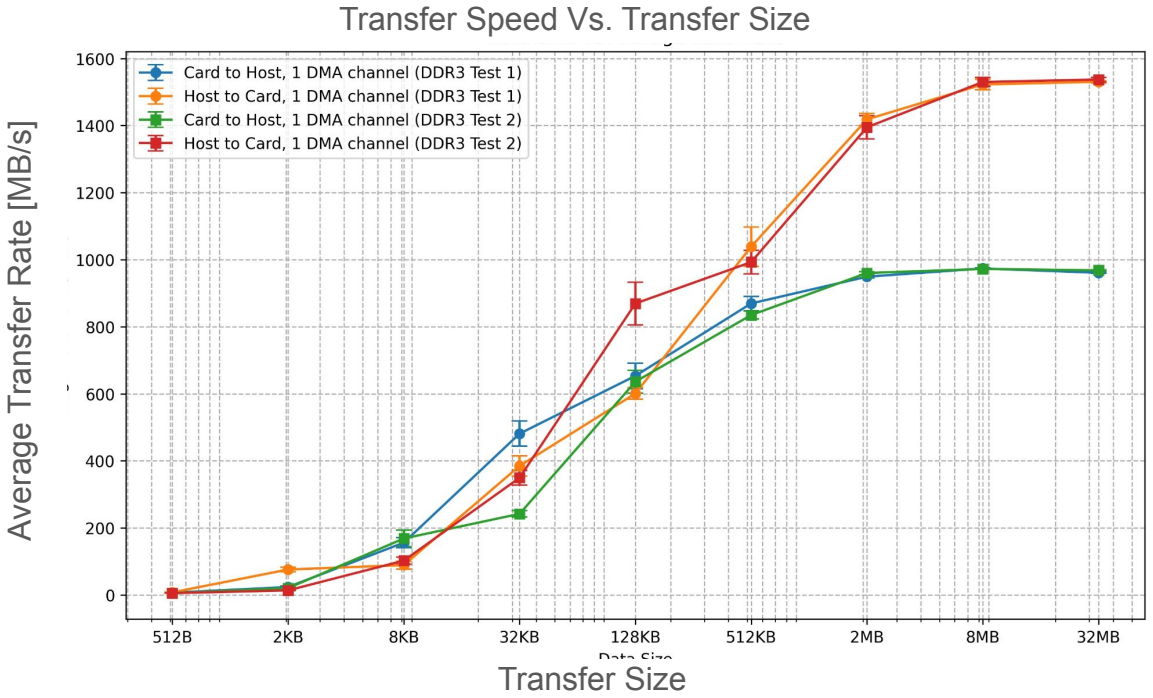


Dependency Diagram

j.carlton@uky.edu

Older (2017) Xilinx XDMA Driver Gives Better Results

- Transfer rates using block ram in a computer with an older OS (CentOS7)
- XDMA driver by Xilinx changes with kernel version
  - Driver version causing performance difference
- Can make slight edits to old driver to see compile on ALMA9 and see performance gains



DMA transfer rate vs transfer size over one channel using Xilinx XDMA tools on CentOS 7

j.carlton@uky.edu

Best Guesses for Plot Shape

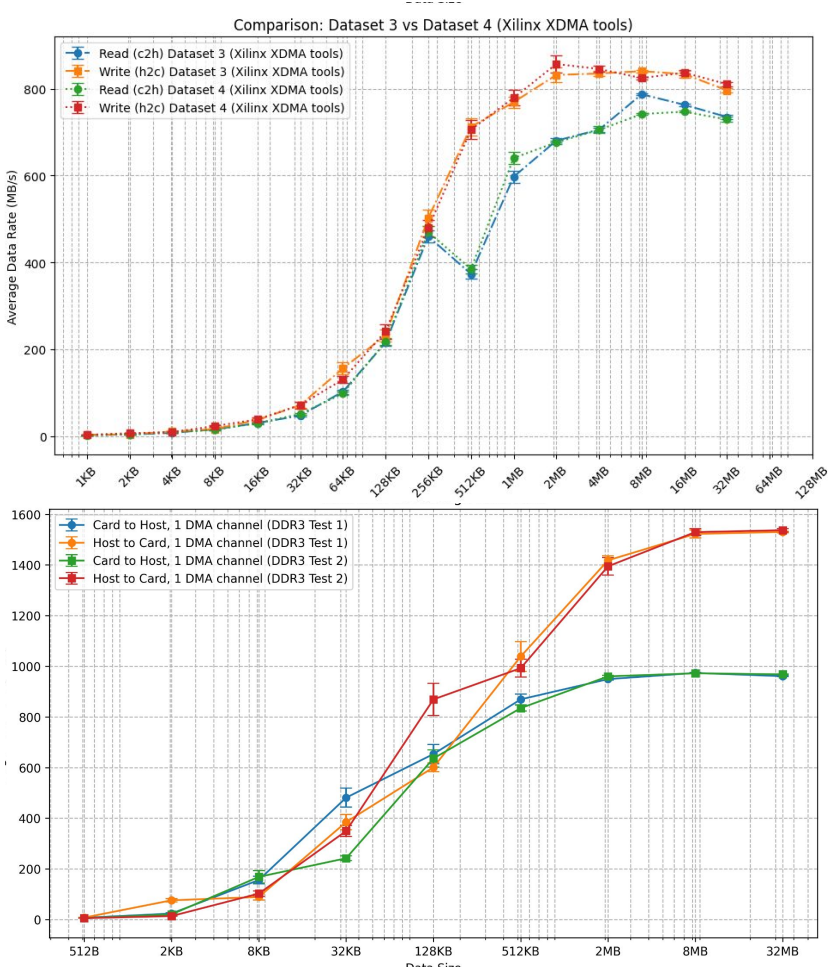
- Rates tests are often “bumpy”
- Timing results at the KB scale are “bumpy”
  - May have to do with system resource management (How data transfers are optimized to DDR3 RAM(?))
  - Could also be due to PCIe bus congestions

DMA transfer rate vs transfer size over one channel using custom C++ Library

j.carlton@uky.edu

Xilinx XDMA Driver Issues

- On CentOS7 and ALMA9 different versions of Xilinx XDMA drivers are used
  - Newer tests were done on ALMA9 using a newer version
- Performance was evidently affected
  - Unsure on the exact reason
- Problem can be remedied by using old driver (from 2017)

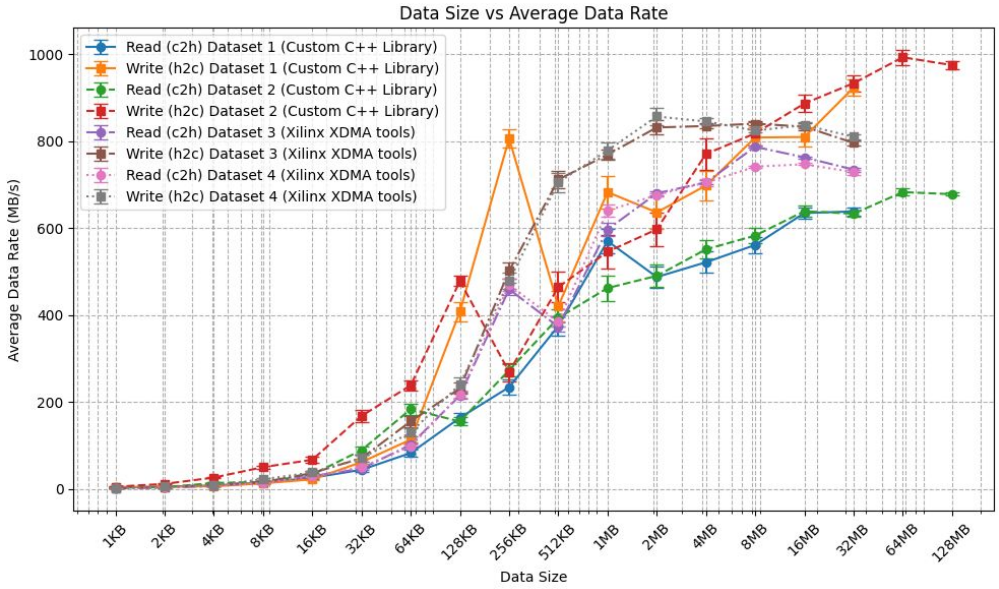


Plots to the right →
DMA transfer rate vs transfer size over one channel
Top: ALMA9 (newer driver)
Bottom: CentOS7 (older driver)

j.carlton@uky.edu

Differences Between C++ Library and Xilinx XDMA Tools

- For these plots Xilinx XDMA tools are artificially inflated by $2^{20}/10^6 \sim 1.05$
  - Rate calculated differently for each program
- Xilinx XDMA Tools beforms “better”
  - Has the more “expected” leveling off shape
  - Also reports faster read speeds
- Unsure what’s causing the discrepency
  - Both pieces of software interface with the boards very similarly



DMA transfer rate vs transfer size over one channel.
Two datasets for each “method”
(Custom C++ library vs Xilinx XDMA tools)

j.carlton@uky.edu

Motivation for PCIe Based Readout

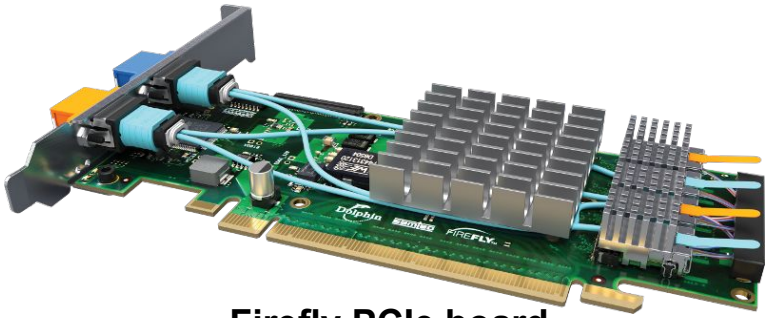
- Using APOLLO system (no more µTCA crates)
- Data received by desktop through Firefly PCIe cards
  - An optical link to communicate with FPGA

Command Module (CU)

+

Service Module (BU)

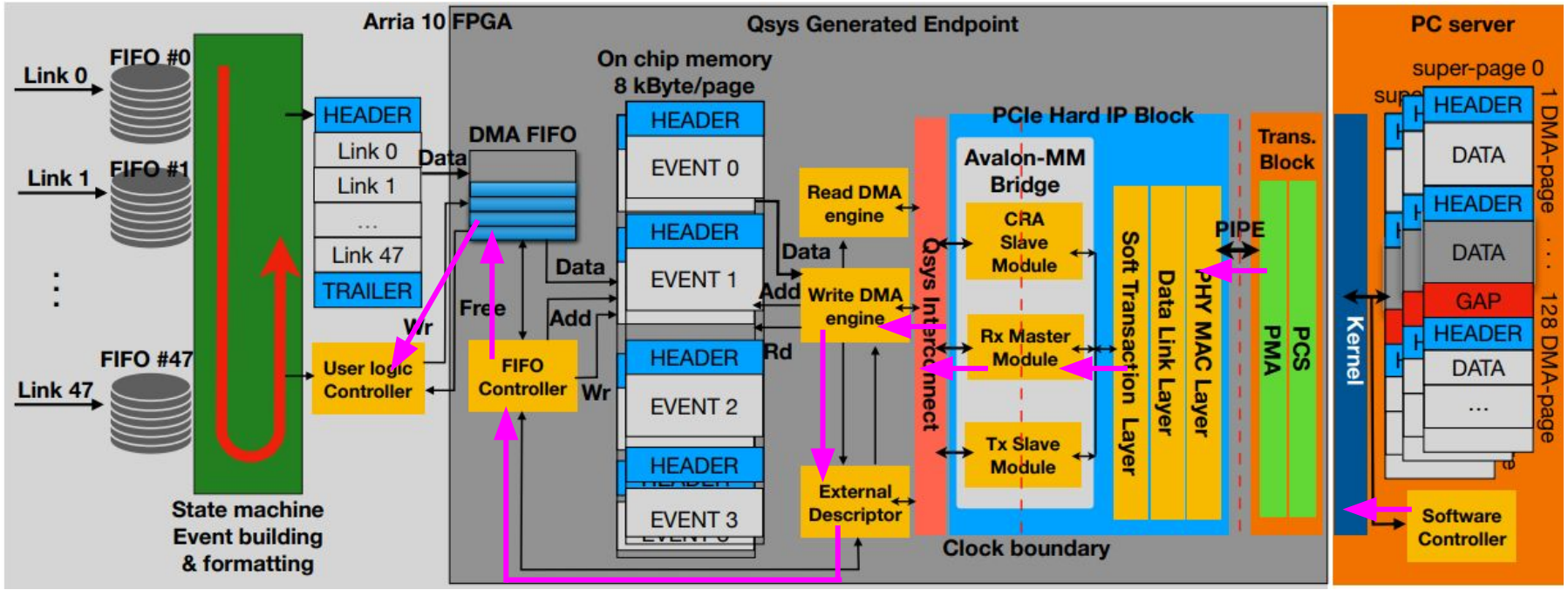
=



Firefly PCIe board

Belle II DAQ control signal flow

= Path to write to user logic controller



How Belle II Data Flow Works

- Data Collection
  - Each Belle2link receives raw data from a sub-detector and buffers it into a FIFO memory
- Event Building
  - The firmware merges fragments from 48 links into a single event, formats it, and stores it in a DMA FIFO (32 kB)
- DMA Transfer
  - Data are transferred from the FPGA's memory to the PC's memory using PCIe DMA om 8kB pages
- PC Processing
  - Data are received in 1 MB super pages in a large buffer on the PC

= Have similar system working at UKY test stand

= Do not have similar system working at UKY test stand

j.carlton@uky.edu

What is a PLL? (Phase Locked Loop)

- A phase lock loop (PLL) is a control system that generates an output signal whose phase is fixed relative to the phase of an input signal