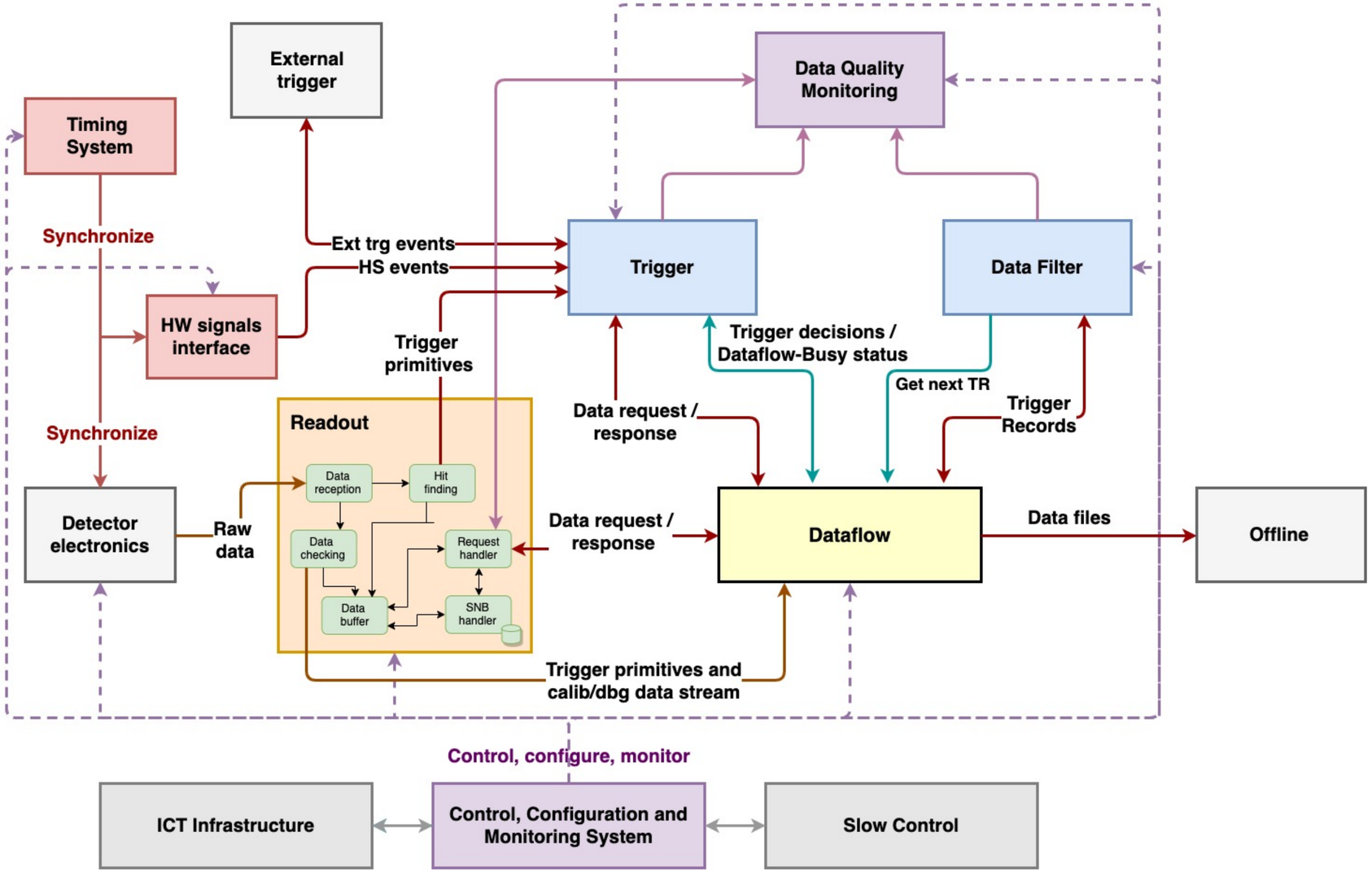
DAQ - Readout Developments & Firmware

Jim Brooke, University of Bristol, UK

DAQ Overview

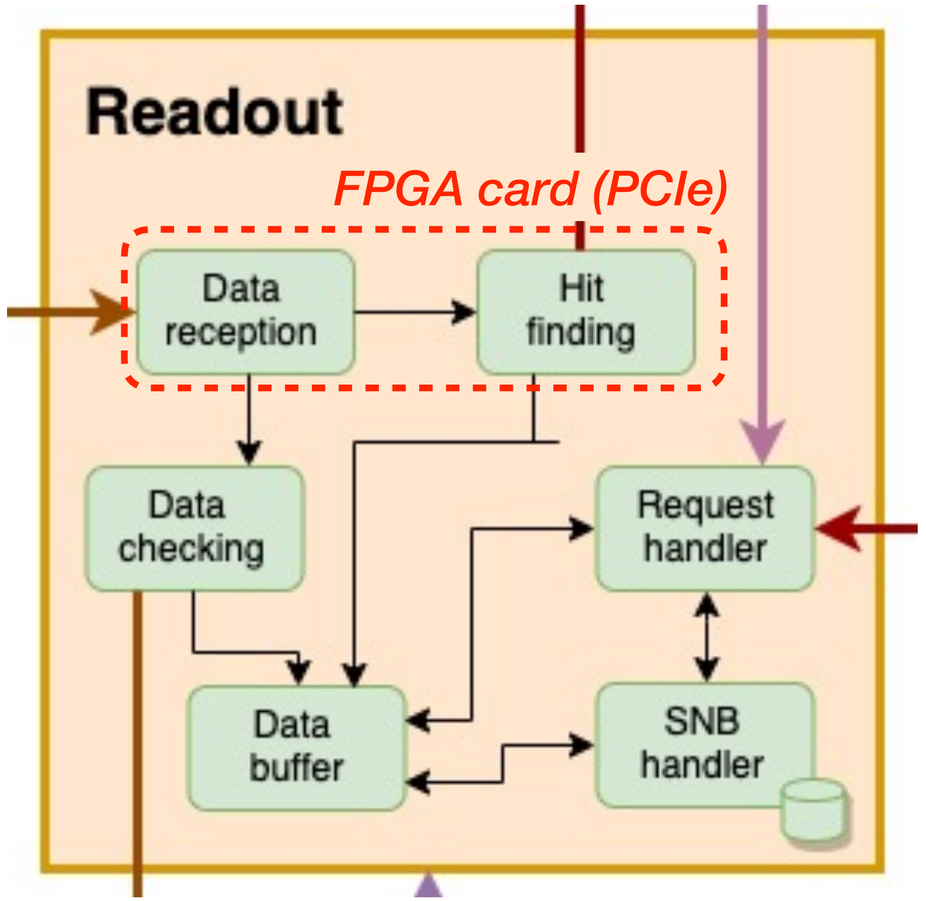


Readout Overview

- Data reception
- Trigger primitive generation

FPGA card (PCIe)

- Data buffer
- Data streams :
  - Triggered
    - Request/response
  - Non-triggered
    - TP/debug/calib data stream
  - Special SNB handling

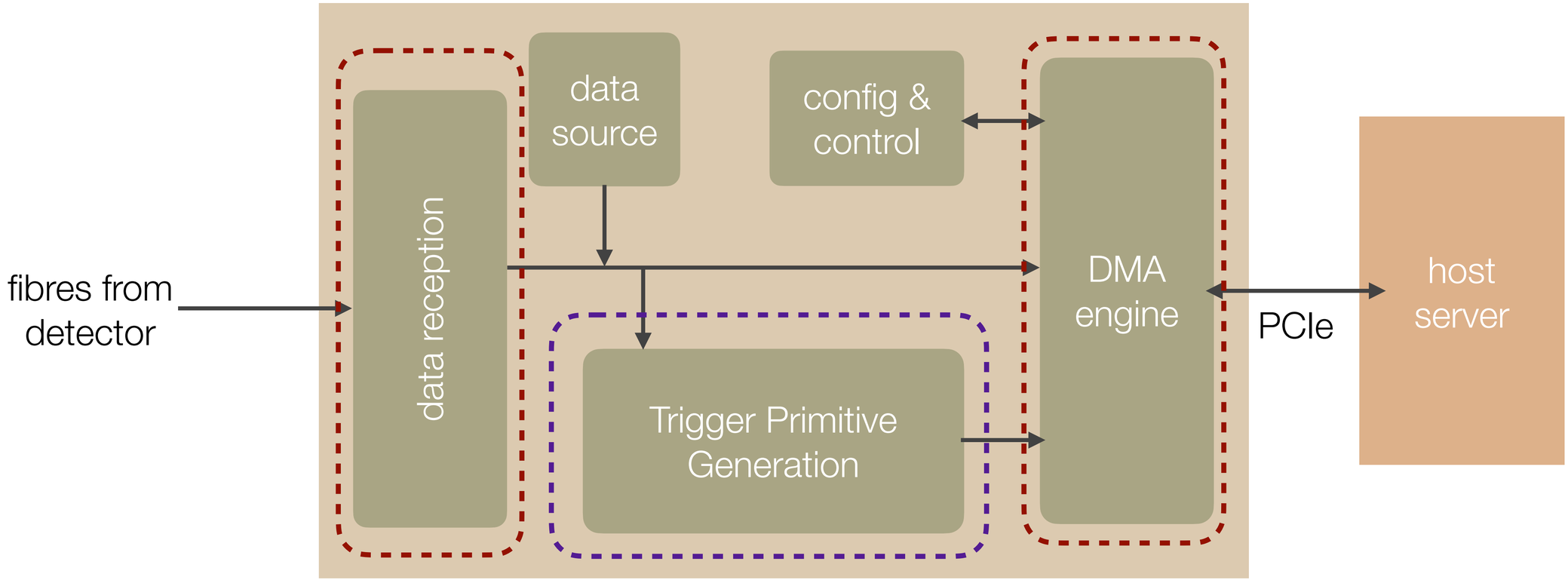


Data Reception

- **HD module front-end data rate ~12 Tbit/s**
  - 1500 × 9.6Gbit/s TPC links
  - 150 × 4.8Gbit/s PDS links
  - Received on 170 PCIe cards
- **VD module front-end data rate ~15 Tbit/s**
  - 320 × 40G links (TE)
  - 960 × 10G links (BE)
  - Received on 200 PCIe cards

- 2 flavours of card required - due to interface specifications
  - HD & VD BE - “FELIX FullMode”
  - VD TE - “Ethernet”

Readout Card Overview



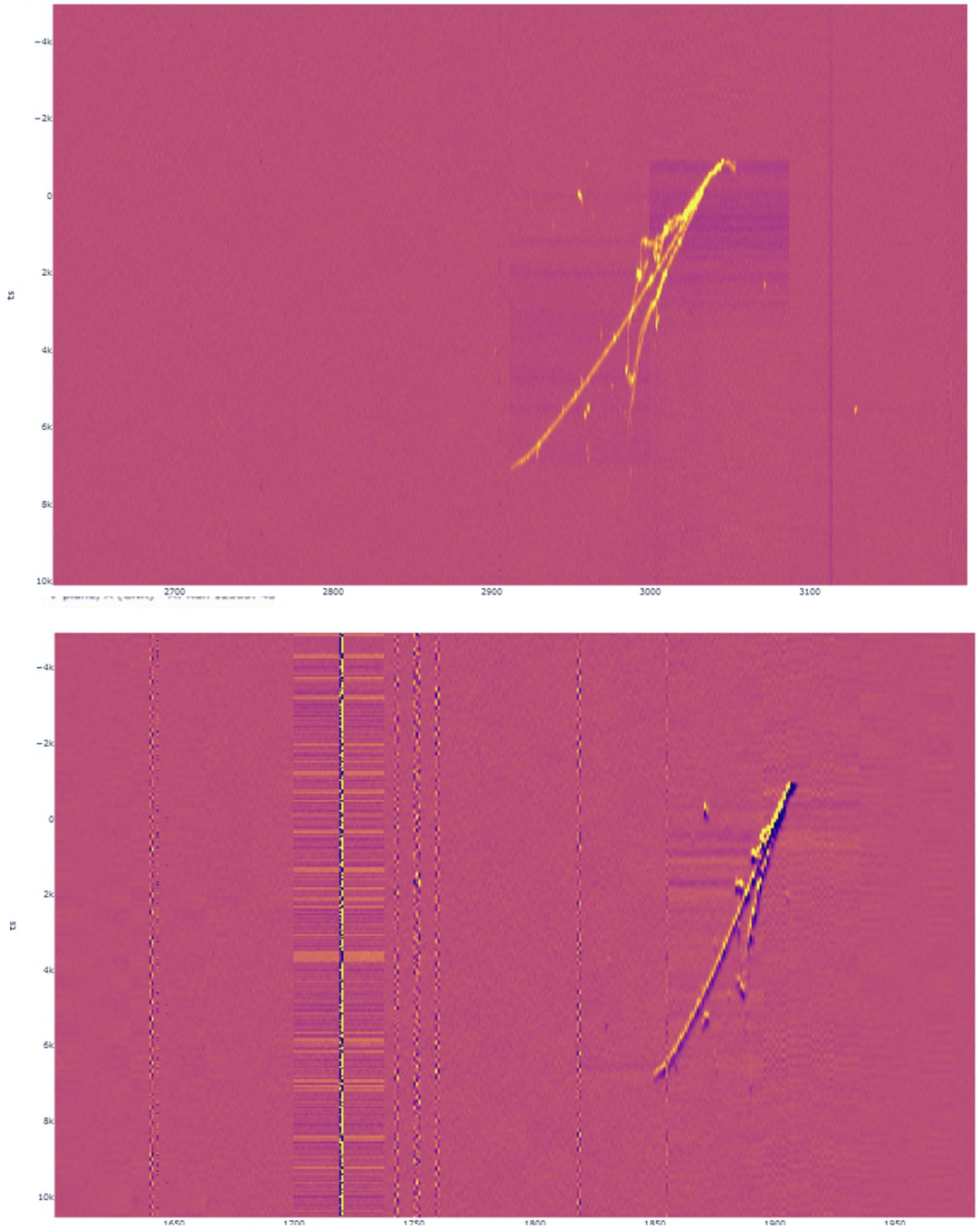
For HD/VD BE :

- FELIX components (with DUNE tweaks) in red
- DUNE components in purple

Recent Progress - Readout

- Readout deployed at CERN NP
  - “Phase 2” FELIX firmware
  - Software with essential functionality
  - Debug/calib streams
  - SNB writing
  - TPG not yet deployed

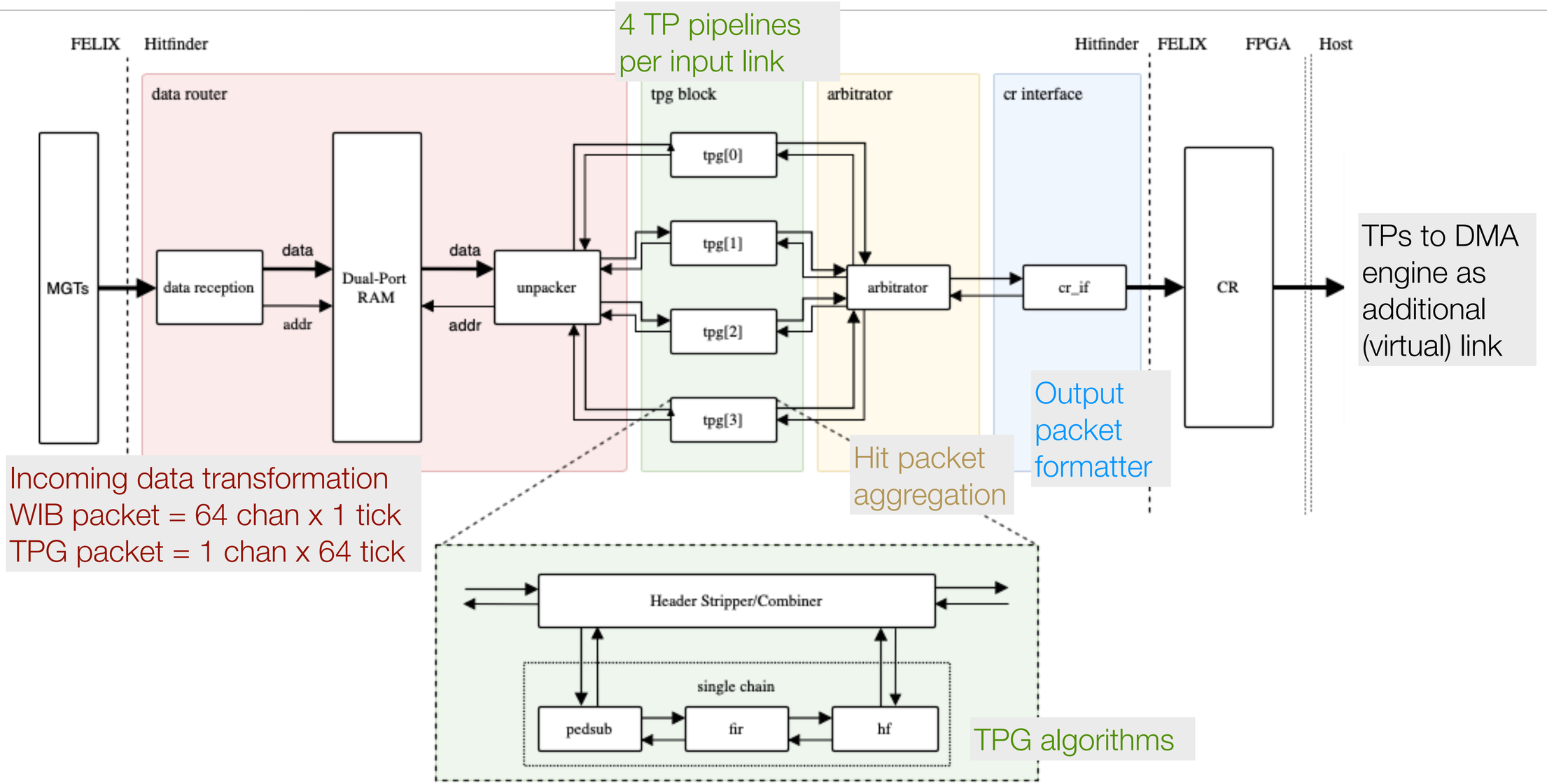
- Figure shows tracks captured with Vertical Drift cold box (BE)



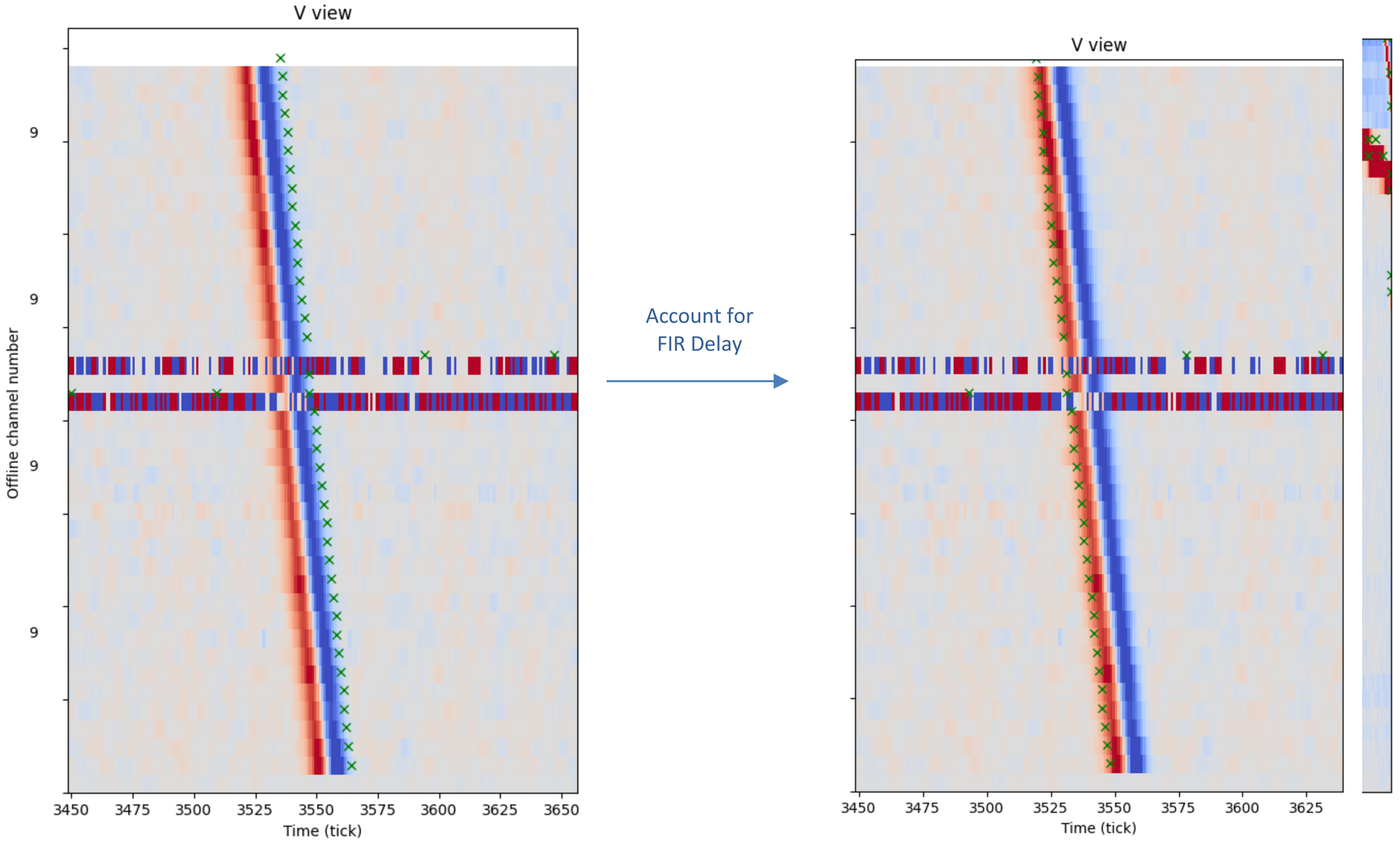
Recent Progress - SNB recording

- Special data handling for SNB triggers
  - Write full data stream to SSD for 100s
- Demonstrator using SW RAID 0 of 4 NVMe Samsung Pro 1TB
- Metrics shown for 1 readout host, during 3 SNB triggers
  - Also serving 50 Hz of random ‘regular’ triggers…

Trigger Primitive Path Firmware

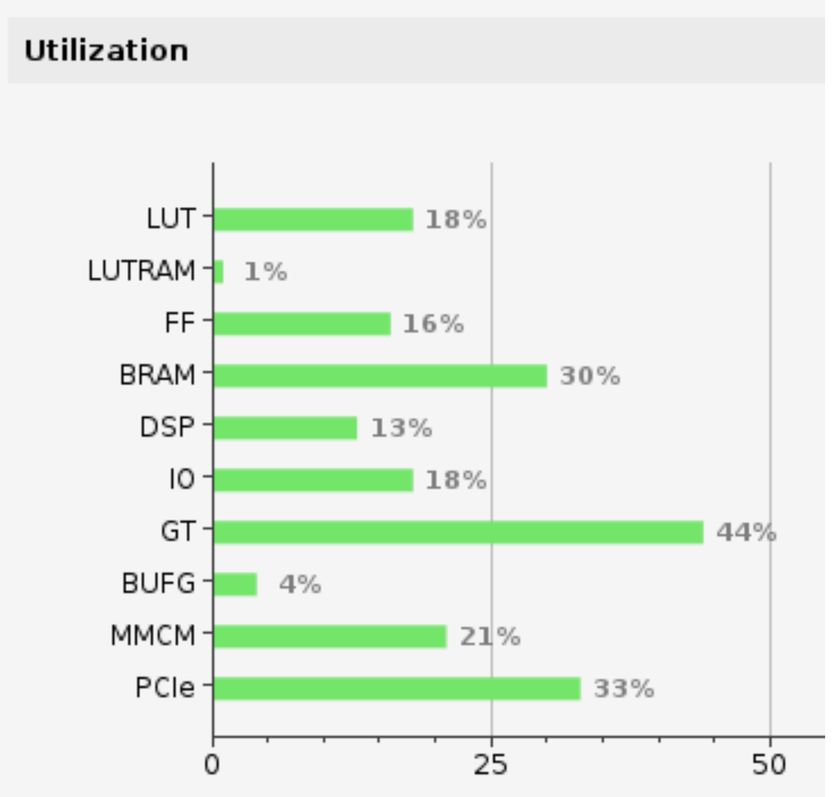


Trigger Primitive Firmware at PD1

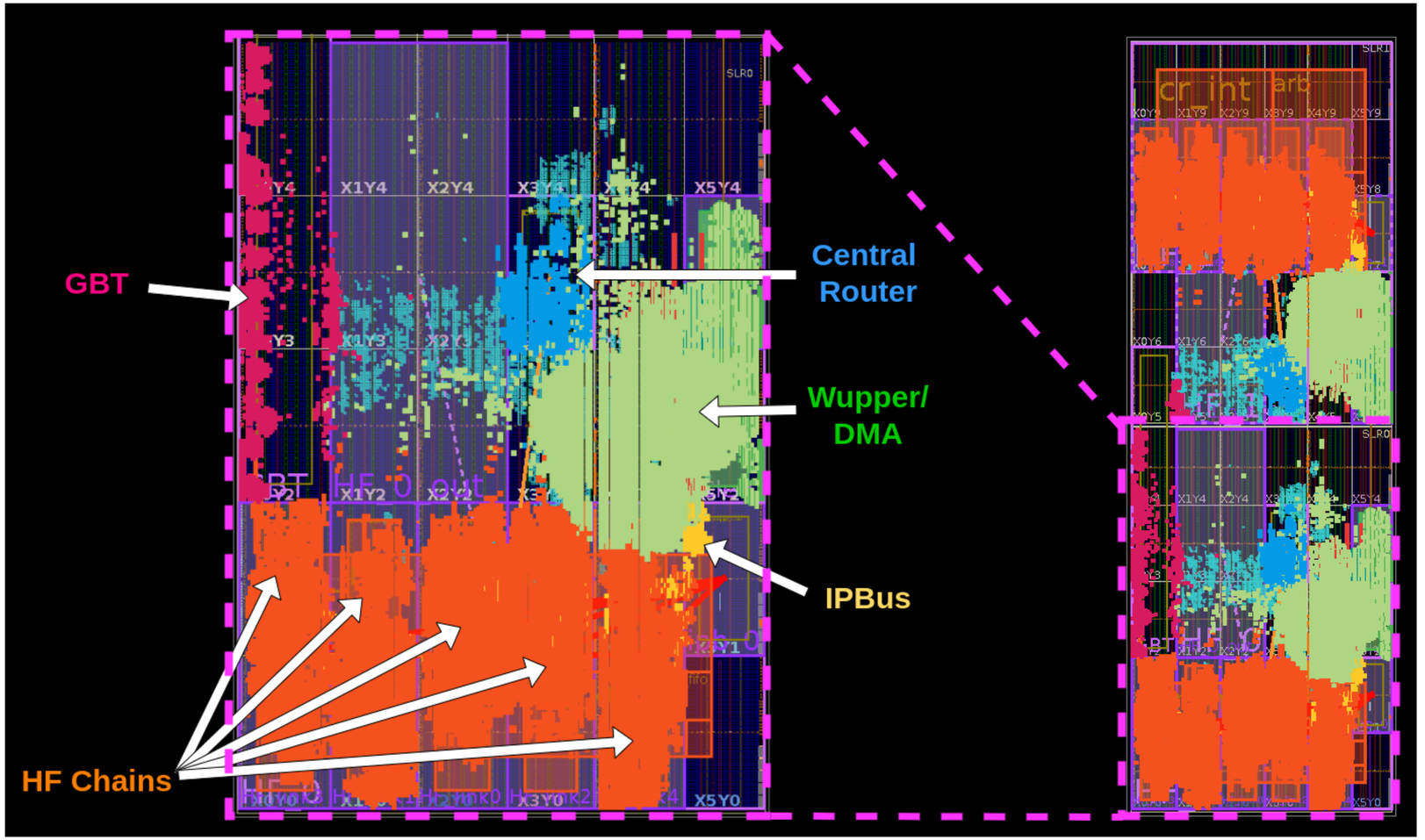


Recent Progress - Trigger Primitive Generation

- Migration to FELIX “phase 2” firmware
  - Baseline firmware for PD2 & used at coldboxes
- Numerous interface changes
  - Changes required in both DUNE TPG and FELIX firmware
  - Firmware builds successfully
- Integration tests underway
  - Several firmware issues identified & fixed
  - Needs to be fully integrated & deployed soon



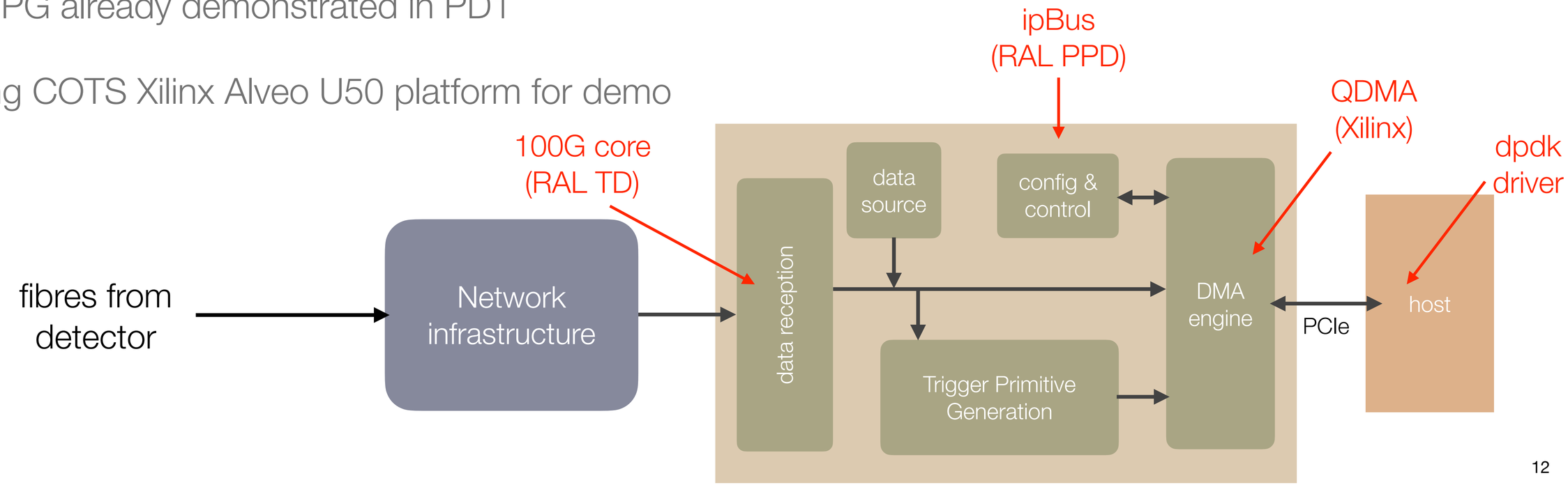
FPGA Resource Use



- **FPGA floorplan**
  - Xilinx KU115
  - FELIX FLX712
  - 10 link build (1 APA)
- **Functionality**
  - GBT - data reception
  - Central Router/Wupper - readout
  - HF chains - trigger primitives

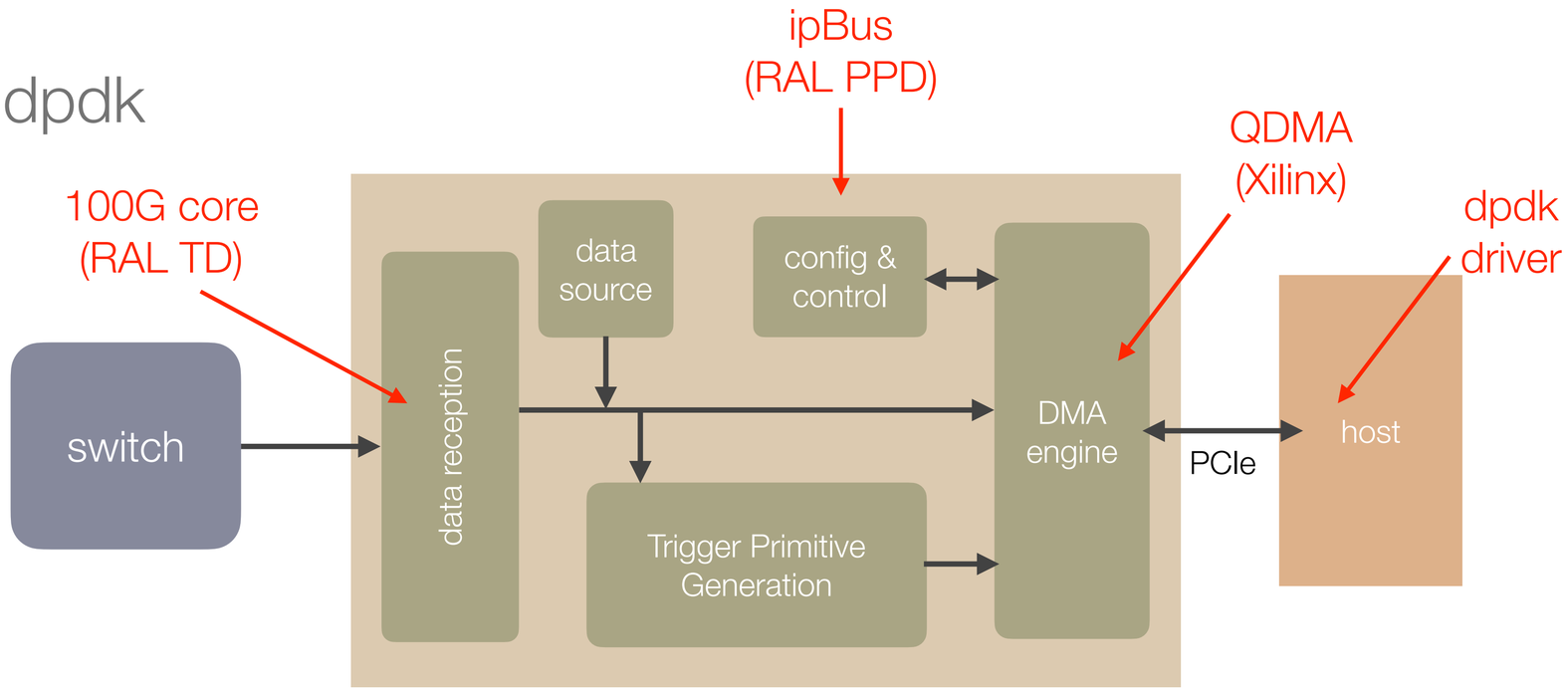
Ethernet Readout

- VD TE will use ethernet (UDP) packets for readout - legacy of DP TPC
- Baseline components have been identified for demonstrator - most demonstrated previously
  - 100G ethernet core run overnight; 98% throughput, no errors
  - ipBus-QDMA bridge demonstrated with individual read/writes
  - TPG already demonstrated in PD1
- Using COTS Xilinx Alveo U50 platform for demo

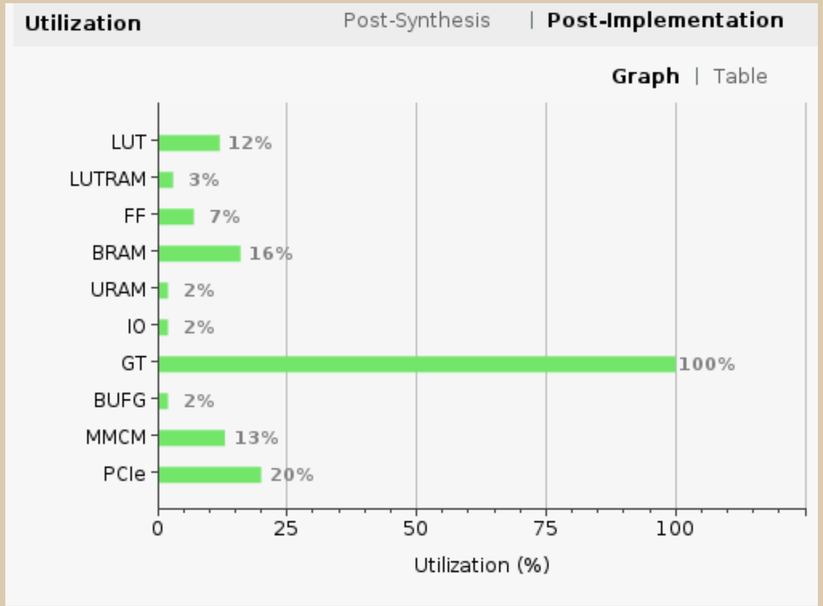


Ethernet Readout Firmware Status

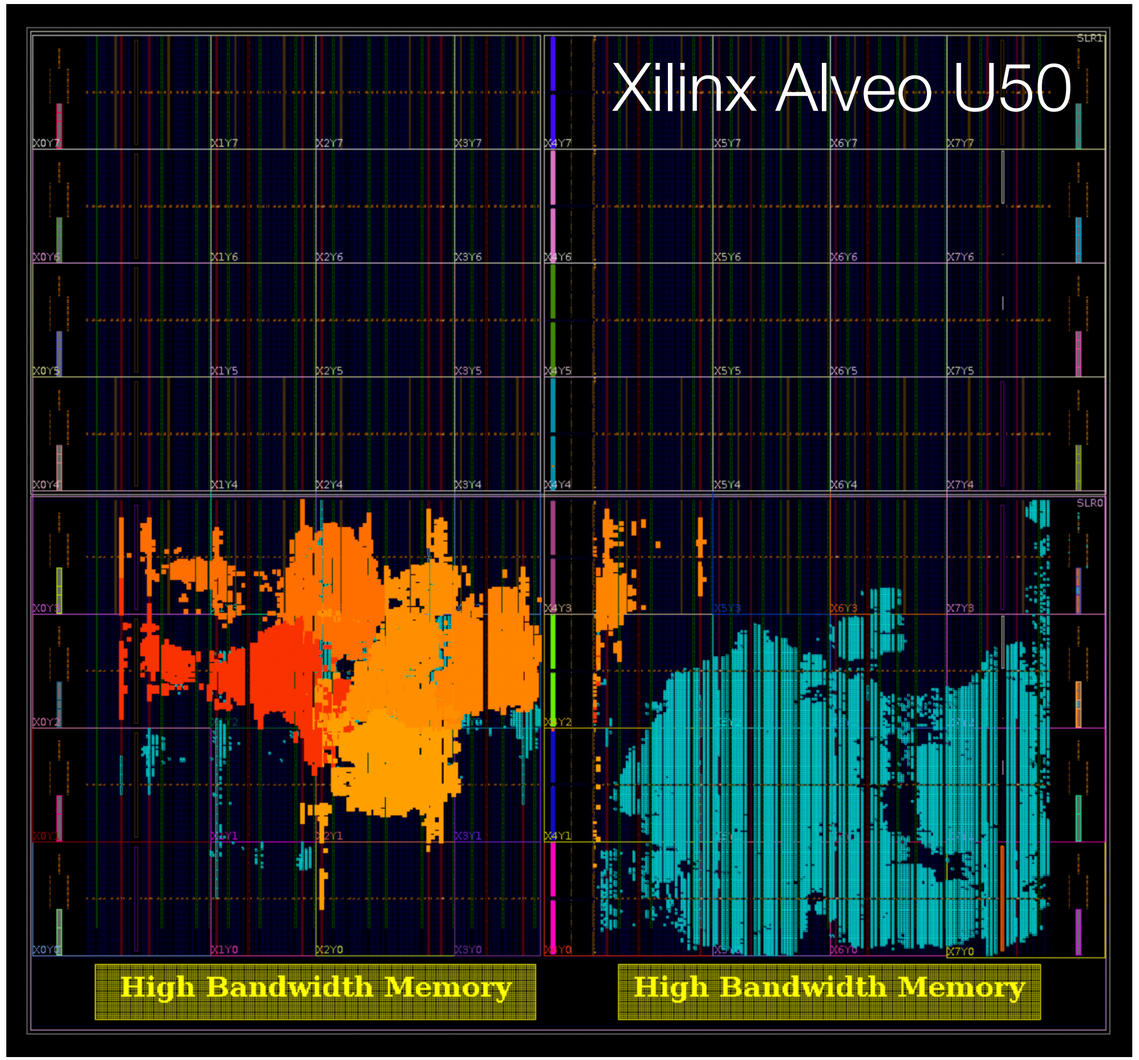
- Current firmware focus is integration of readout path
  - Successful tests of readout path (switch to host) - RAL TD
  - Tests of data transfers using smart NICs with dpdk - Oxford
- Next steps :
  - Integrate readout path + dpdk
  - Integrate ipBus and TPG
- VD TE data format
  - 1 channel per packet
  - Simplifies the TPG task



Ethernet FPGA Resource Use



Readout (100G + QDMA) resource use



- QDMA + TPG resources
- “5 link” TPG = 40% of requirement

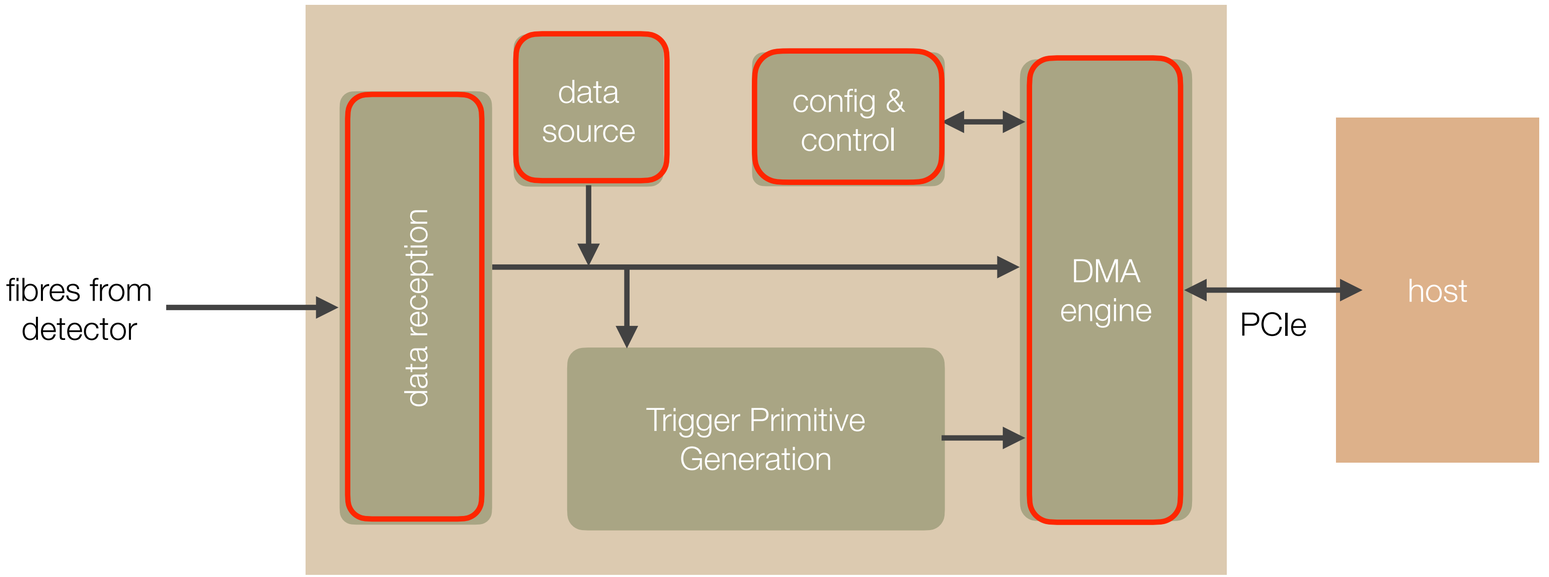
Summary

- **Readout system is now well advanced**
  - FELIX based version deployed at coldboxes with near full functionality
  - SNB writing demonstrated - reading from SSD to be integrated soon
- **Firmware trigger primitives are final piece**
  - Good progress made over recent months, but well behind schedule
- **Ethernet readout demonstrator underway**
  - All components of readout path demonstrated at some level
  - Aim to produce readout only demonstrator for use with VD TE cold box this year
  - Integration of TPG will follow

Backup

Readout Path Firmware

- Demonstrated in context of HD CE
  - 10 links @ 9.6 Gbps, FELIX FullMode format
  - Also, 12 links @ 9.6 Gbps demonstrated in lab for VD BE readout
- Uses FLX712 hardware platform + FELIX FW/SW ecosystem
  - Developed for ATLAS upgrades
  - Broad existing user base
- Some DUNE-specific FW
  - “Jumbo block super chunk”
  - 4.8Gbps input links
  - Dual output data rx stage, for TPG
- All integrated into FELIX repo

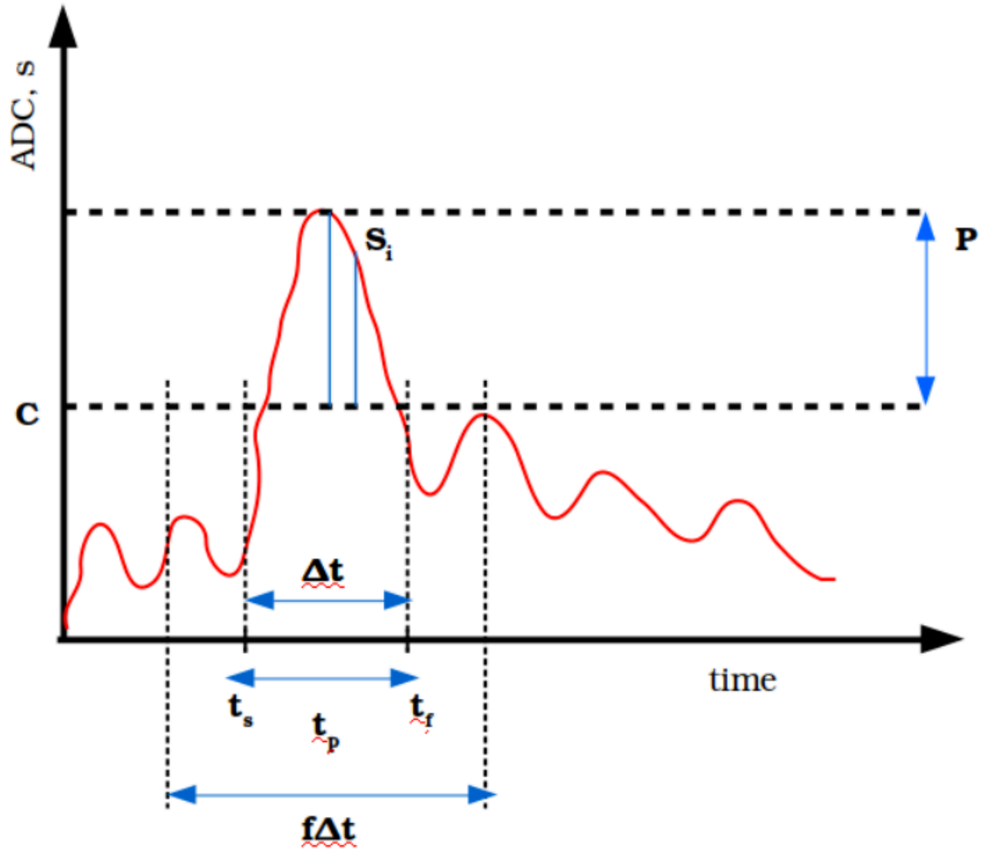


Firmware Integration

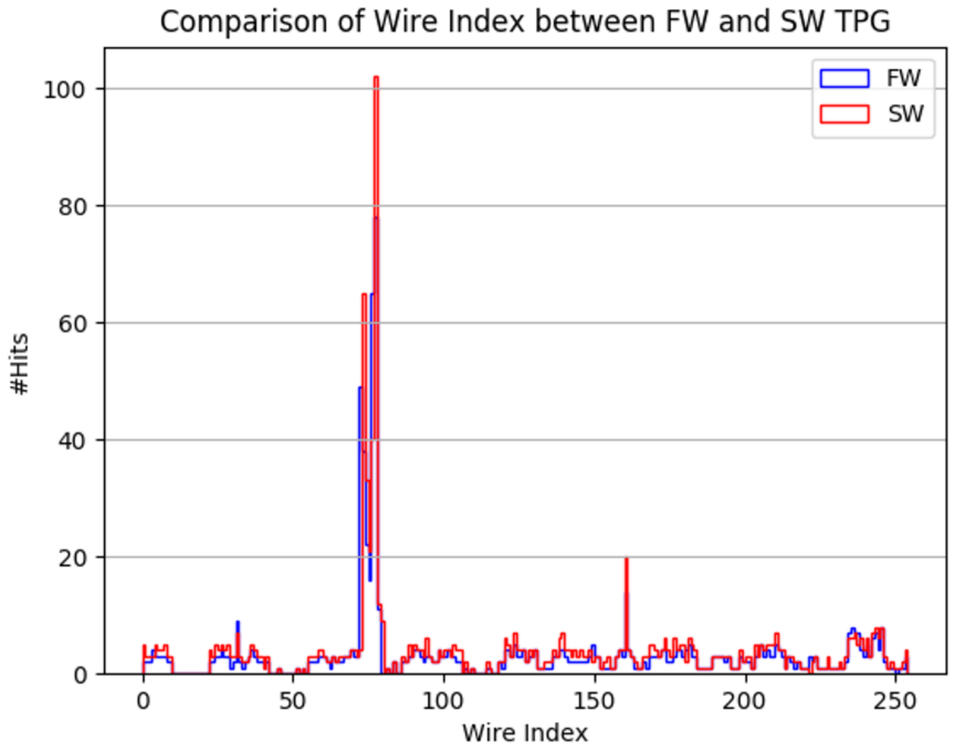
- Some lessons from PD1
  - Firmware integration was challenging
  - Readout and TPG firmware developed in different environments
- Base FELIX FW
  - PD1 demo based on FELIX “phase 1”
  - Integration with FELIX “phase 2” took time due to changes in underlying architecture
  - Now have a good working relationship with FLX developers
  - DUNE-specific features have been merged with main FELIX repo when needed
- Different build systems
  - Conversion of FELIX build scripts to ipbb dependency files for DUNE
- Different config/control mechanisms
  - TPG uses ipBus
  - Bridge ipBus over FELIX DMA engine, using memory mapped ipBus master
- Introduced FPGA area constraints
  - Allows standalone development of components, including placement, without a full time-consuming build

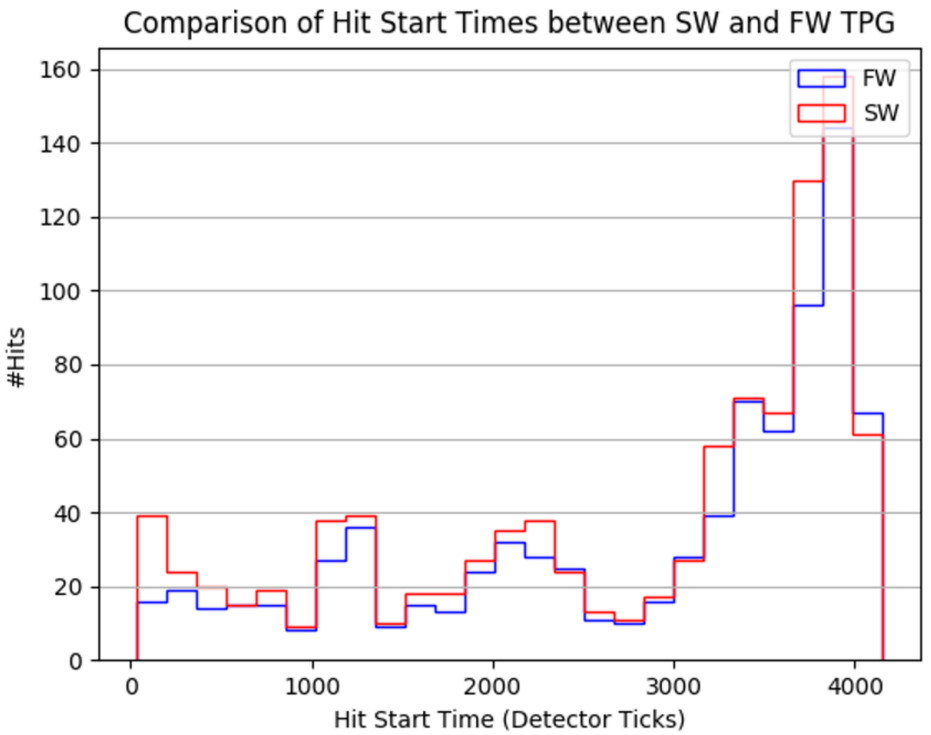
Trigger Primitive Generation

- TPG processing for each channel must :
  - Apply FIR for noise removal
  - Subtract pedestal
  - Identify hits (signal above threshold)
- TPG demonstrated in both SW and FW in 2020
  - Firmware TPG chosen for final system
  - Host resources can be used for other processing tasks
  - Software TPG will be used for validation, simulation, and as fallback



TPG Results from PD1

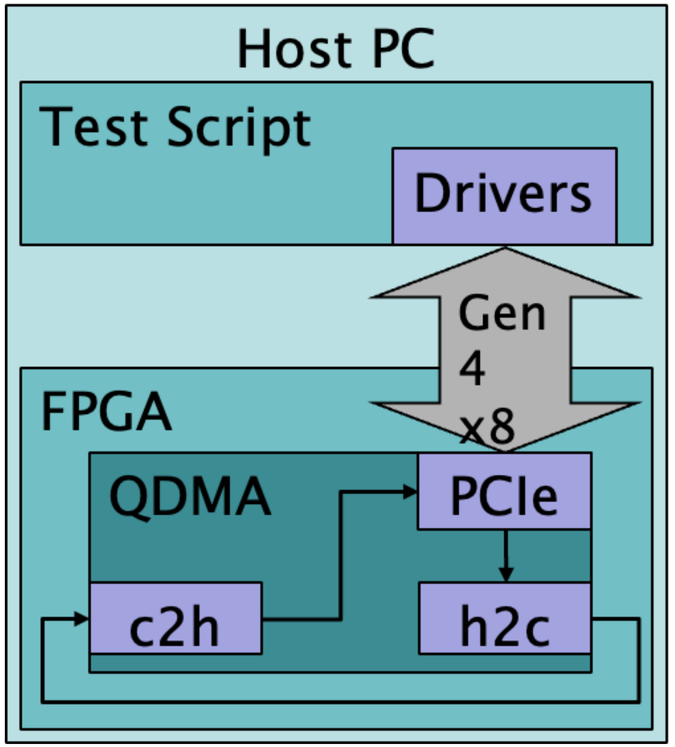




- Comparison of FW TPs with SW TPs shows reasonable agreement

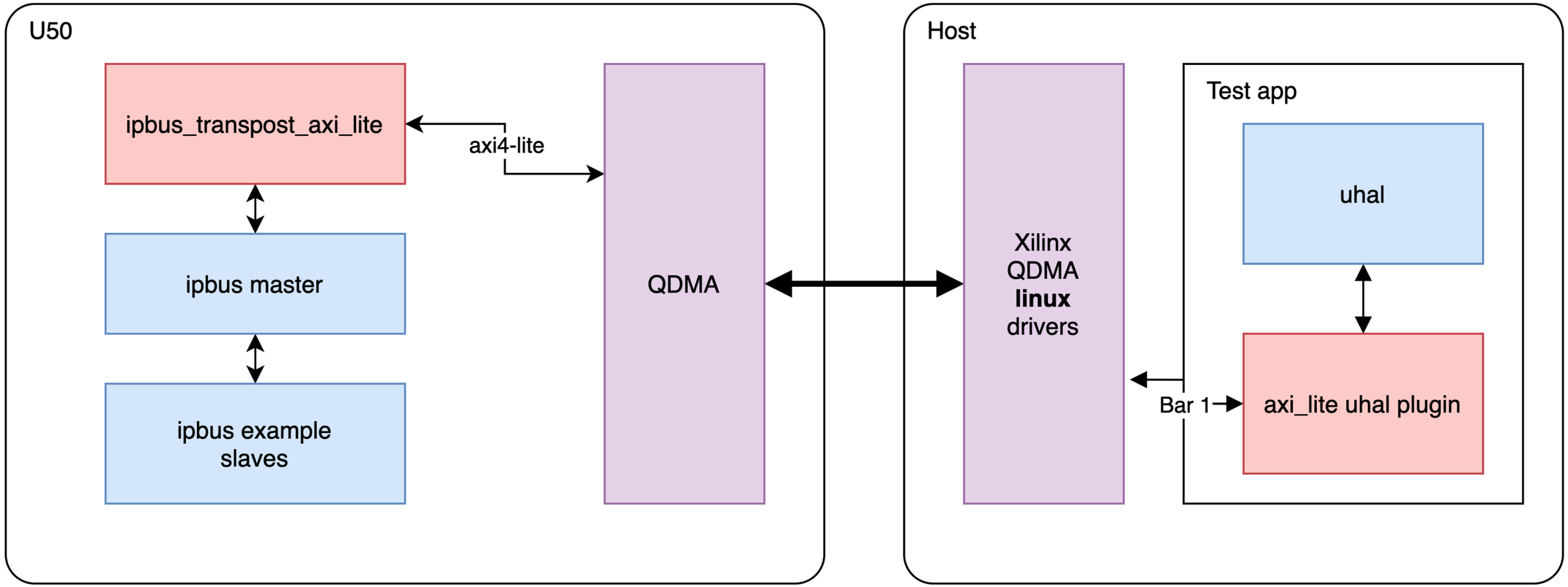
Ethernet Readout - QDMA Test

- **QDMA**
  - PCIe connectivity between host PC and FPGA
  - PCIe gen4 x8 or gen3 x16 for ~128 Gb/s transfer speeds
  - Up to 2048 ‘queues’ of data
    - Can be dynamically enabled and disabled via software
    - Hardware implementation up to FPGA designer
  - Compatible with DPDK software library
- Tested using Xilinx reference drivers
  - PCIe gen4 x8 on Alveo U50 and gen3 x16 on U50-LV
  - Streaming loopback test data from PC <-> FPGA
    - 32 kB transfer with custom design logic
  - Memory-mapped read/write to axi4-lite registers



Streaming loopback test

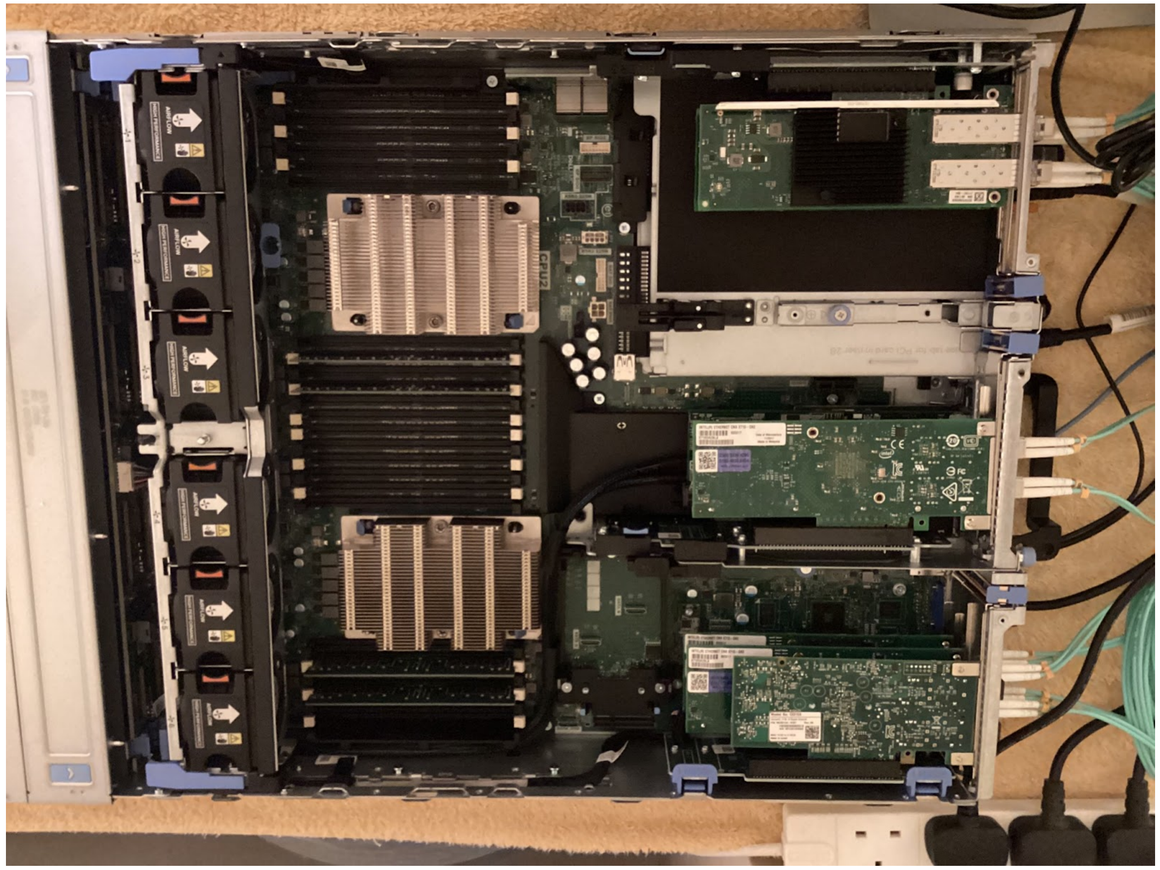
Ethernet Readout - ipBus + QDMA test

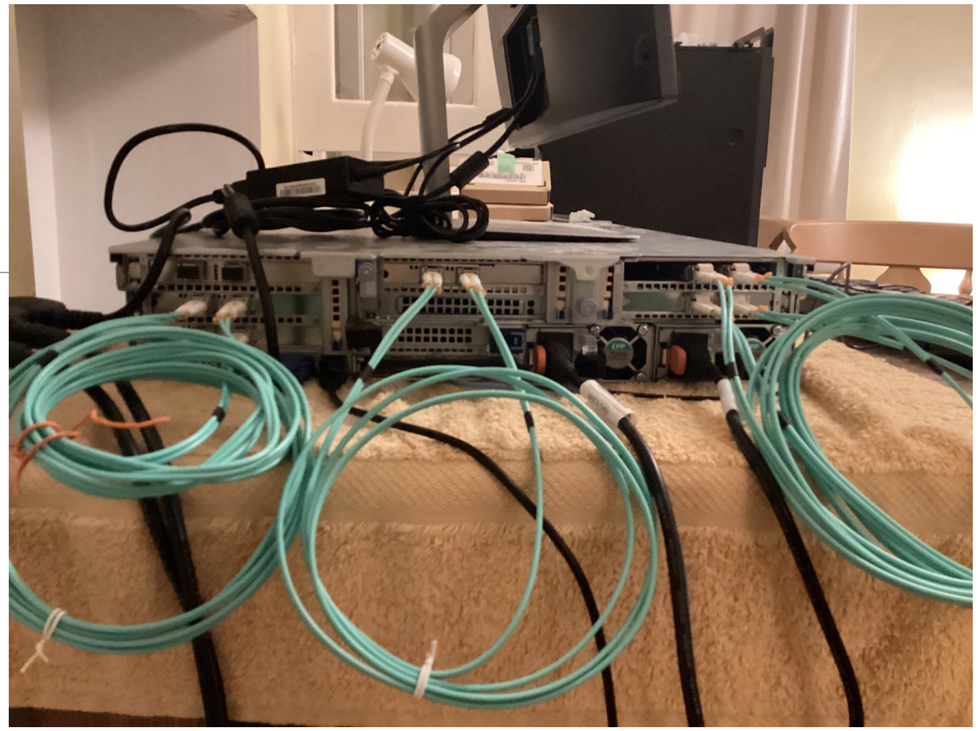


Data transfers with dpdk - test setup

- Setup uses five Intel XL710 dual 10Gb Ethernet cards with optical inputs, which allows ten loopback tests to be run concurrently.
- The XL710 cards have 8-lane PCIe Gen3. Gen3 is 8GT/s, so each card can do 64Gb/s in principle, which is larger than the 2 x 10Gb/s that the devices can receive on their inputs (i.e. the PCIe interfaces on the cards are not a bottleneck).
- The server is a dual socket Dell R740. Processors believed to be silver processors at the low-end of the selection from Dell.
- The machine currently has only limited memory (two sticks for each processor) - so the memory bandwidth is a factor 3 lower than in a fully utilised Xeon processor (which has six DDR4 interfaces).
- Since there is only one machine, it is used as both source and sink, so the data rate and packet rate correspond to both outgoing (Tx) and incoming (Rx) packets. In the real DUNE configuration, we will only have the incoming Rx packets. The numbers here are for the incoming data only.

Data transfers with dpdk - test setup





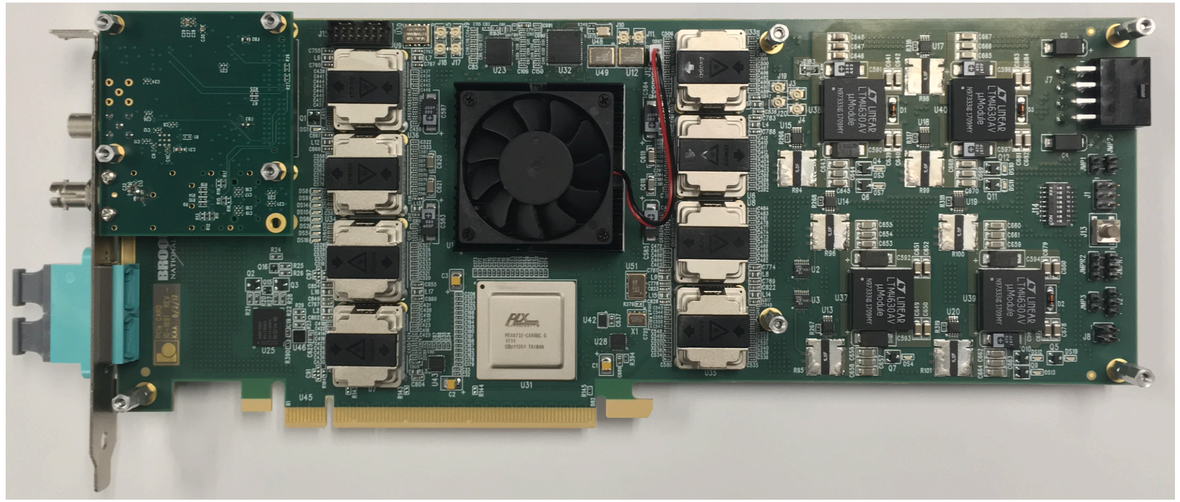
The output has similar numbers
for all 10 Rx and 10 Tx ports

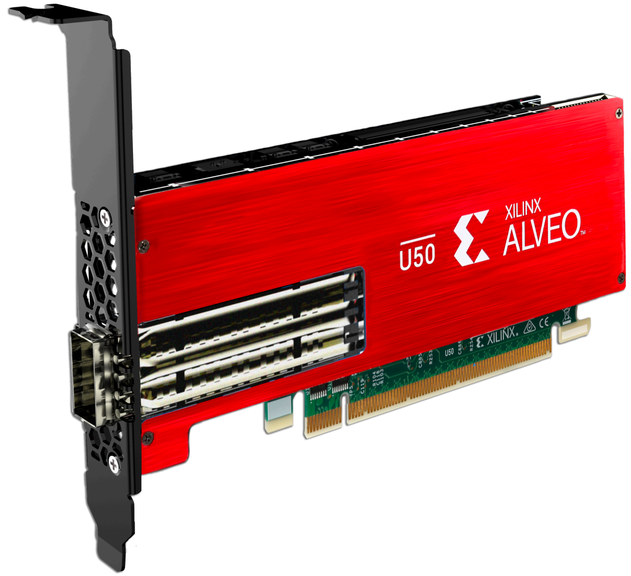
```
Throughput (since last show)
Rx-pps:      14183625          Rx-bps:   7262015920
Tx-pps:      14186103          Tx-bps:   7263289336
############################################################################

########################## NIC statistics for port 9  ########################
RX-packets: 1236895566 RX-missed: 0          RX-bytes:  79161315680
RX-errors: 0
RX-nombuf:  0
TX-packets: 1236900385 TX-errors: 0          TX-bytes:  79161623824

Throughput (since last show)
Rx-pps:      14186115          Rx-bps:   7263291976
Tx-pps:      14183620          Tx-bps:   7262013888
############################################################################
```

Data transfers with dpdk - tests so far

- Used the latest DPDK 20.11.3 LTS version (2021 Sept 6th), with Centos 7
- So far just used it out of the box with no real attempt to optimise. Used the example program 'testpmd' (which is the 'flagship' example program that comes with DPDK).
- Used the vfio driver in the kernel, which plays a similar role to the /dev/flx etc drivers in FELIX, i.e. it allows the DPDK userspace access to what it needs in the hardware to make the data flow. This was done by just following the instructions in the DPDK documentation.
- The test was an out of the box loopback test on all ten sets of ports. A LC fibre cable was connected between ports A and B on each of the five cards and data were sent from the Tx to the Rx in both directions along each fibre.
- To make testpmd forward packets in a loop, used ‘set nbcore 20’ and ‘start tx_first 4’ (will continue to investigate to see how non-optimal it is).
- The data rate reported by testpmd is around 7,200,000,000b/s. This is 7.2e9/1024^3 = 6.7 Gb/s on each link, so for all ten links 67Gb/s = 8.3GB/s data received. The expected rate on an APA is 7.68GB/s.
- The packet rate on each of the 10 Tx and 10 Rx ports is 14 Mpackets/s. That is only 514 bits per packet, which is 64 bytes which is the length of the tx_packets in testpmd.

Readout Hardware





- FELIX FullMode interface
  - Currently using FLX 712
  - Hardware evolution for DUNE in close collaboration with ATLAS
- Ethernet interface
  - Currently using Xilinx U50
  - Will revisit after demonstrators