TPC warm readout with the RCE system

Matt Graham, SLAC

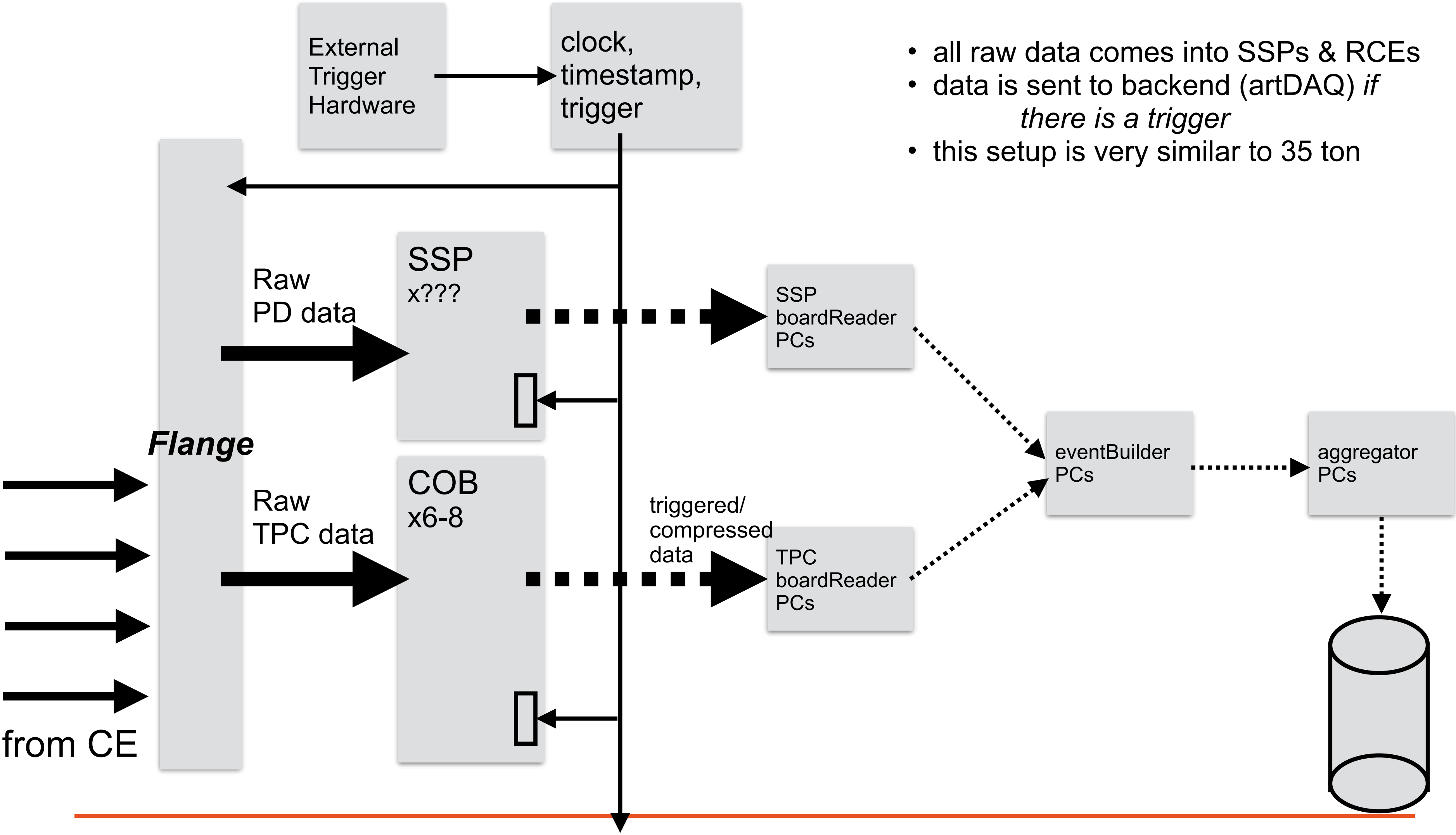
protoDUNE DAQ Review

November 3, 2016

Introduction

- the TPC produces a firehose of data … in protoDUNE, it's > 400 Gbps. This needs to be reduced!
- for protoDUNE we plan 2 methods of bandwidth reduction
  - external triggering on beam particles passing though detector
    - experiment require us to take 25 Hz per spill
  - lossless data compression
    - hopefully x4…depends on noise
- the "RCE solution" uses FPGAs packaged into an ATCA front-board to perform the compression, buffer the data, apply the trigger, and send data out via ethernet for event building
- Many of the items discussed here (and more) are in DUNE doc-db-1881

Roughly, protoDUNE DAQ



- all raw data comes into SSPs & RCEs
- data is sent to backend (artDAQ) *if there is a trigger*
- this setup is very similar to 35 ton

What do we mean by “RCE”

- RCE == Reconfigurable Cluster Element
  - it’s the processing unit
  - Xilinx ZYNQ SoC — dual core ARM; 1 GB DDR3;
- the RCE platform
  - the base suite of hardware/firmware/software that is used to develop your DAQ around
  - lots of acronyms: COB (cluster-on-board), DPM (data processing module), DTM (data transfer module), RTM (rear transmission module)…
- “the RCEs” ~ sloppy way to refer to the entire system

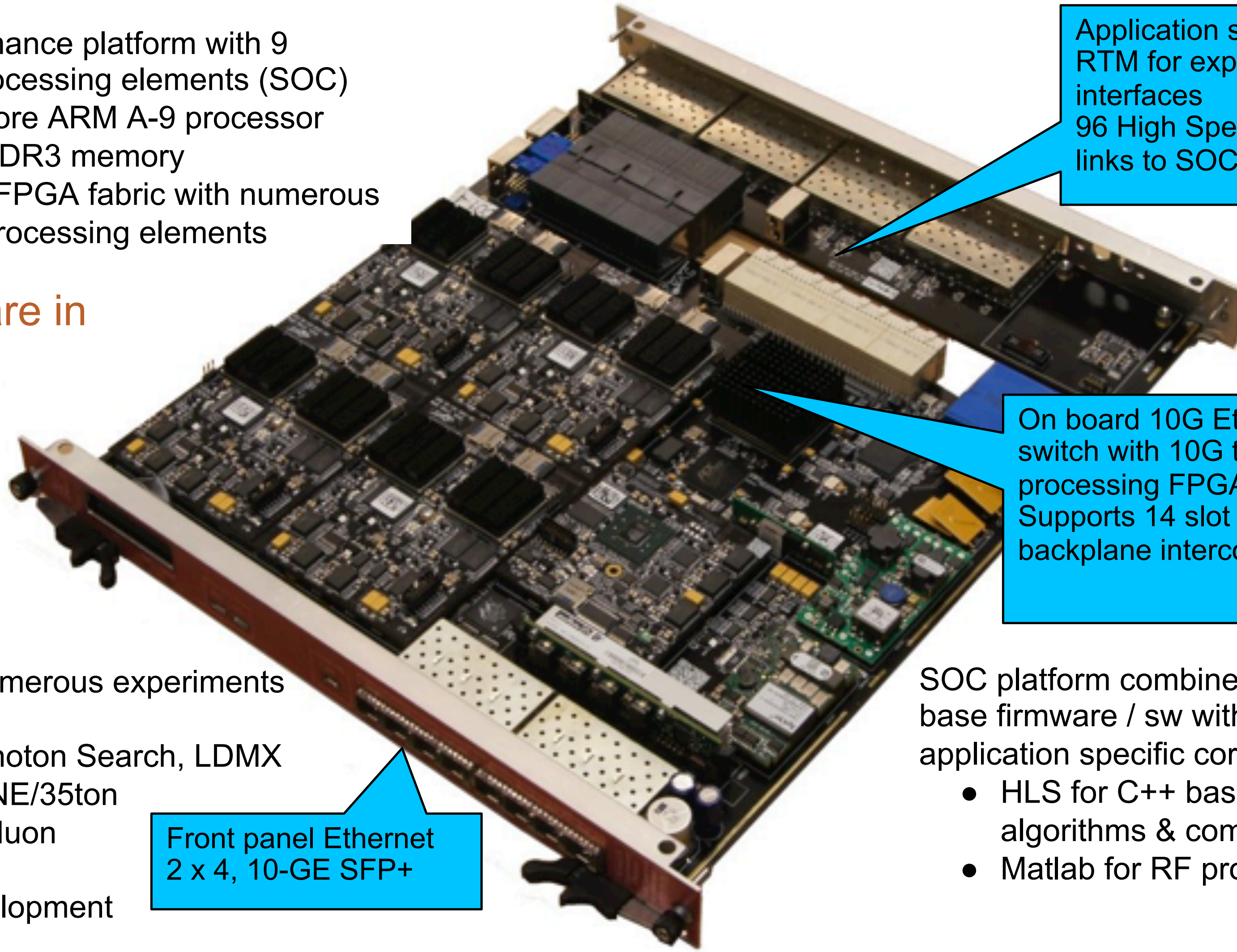
RCE platform hardware

High performance platform with 9 clustered processing elements (SOC)

- Dual core ARM A-9 processor
- 1GB DDR3 memory
- Large FPGA fabric with numerous DSP processing elements

Application specific RTM for experiment interfaces
96 High Speed bi-dir links to SOCs

schematics are in LBNE doc-db-9255



On board 10G Ethernet switch with 10G to each processing FPGA
Supports 14 slot full mesh backplane interconnect!

Deployed in numerous experiments

- LSST
- Heavy Photon Search, LDMX
- protoDUNE/35ton
- ATLAS Muon
- KOTO
- ITK Development

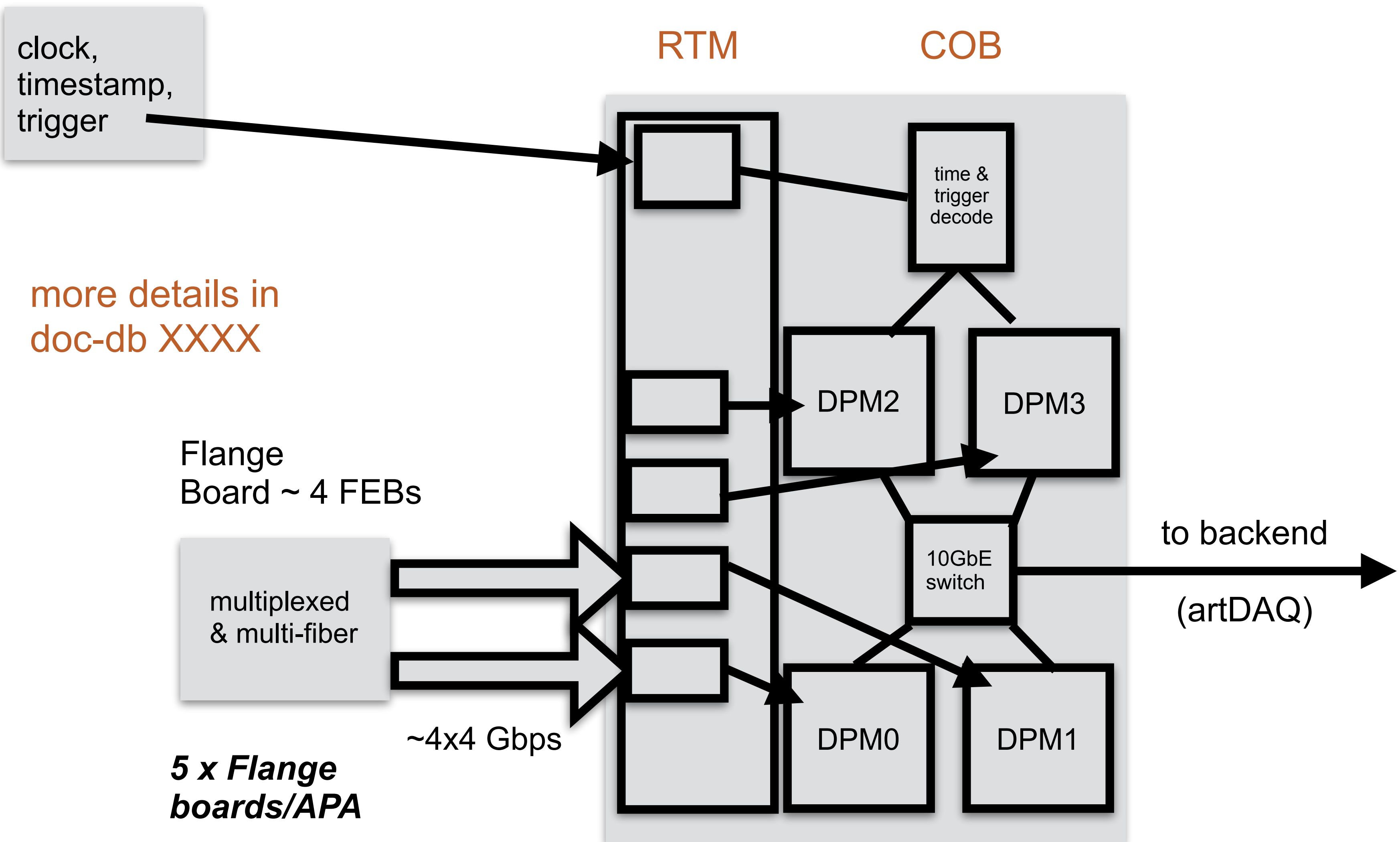
Front panel Ethernet
2 x 4, 10-GE SFP+

SOC platform combines stable base firmware / sw with application specific cores

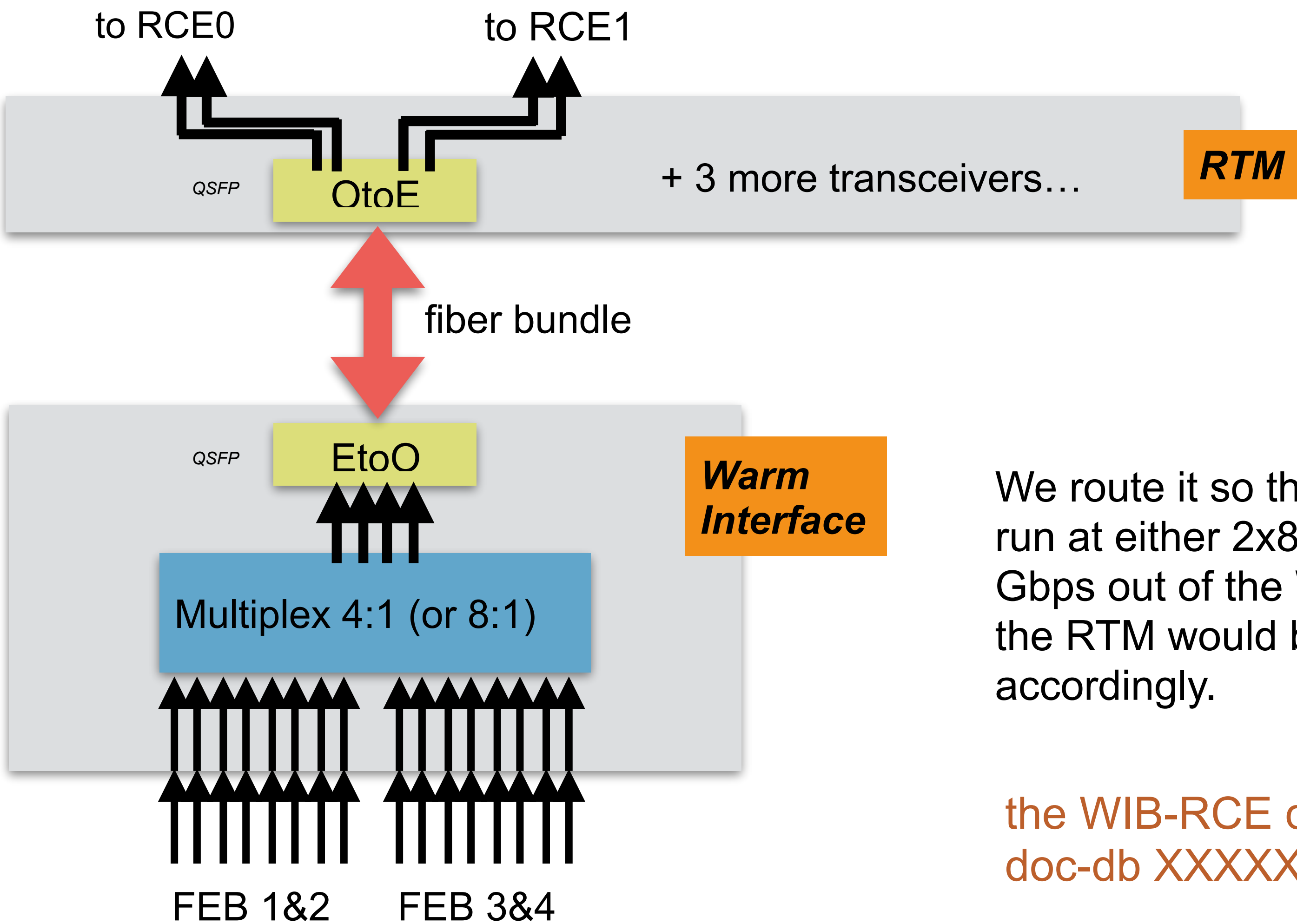
- HLS for C++ based algorithms & compression
- Matlab for RF processing

RCE platform in protoDUNE

more details in
doc-db XXXX



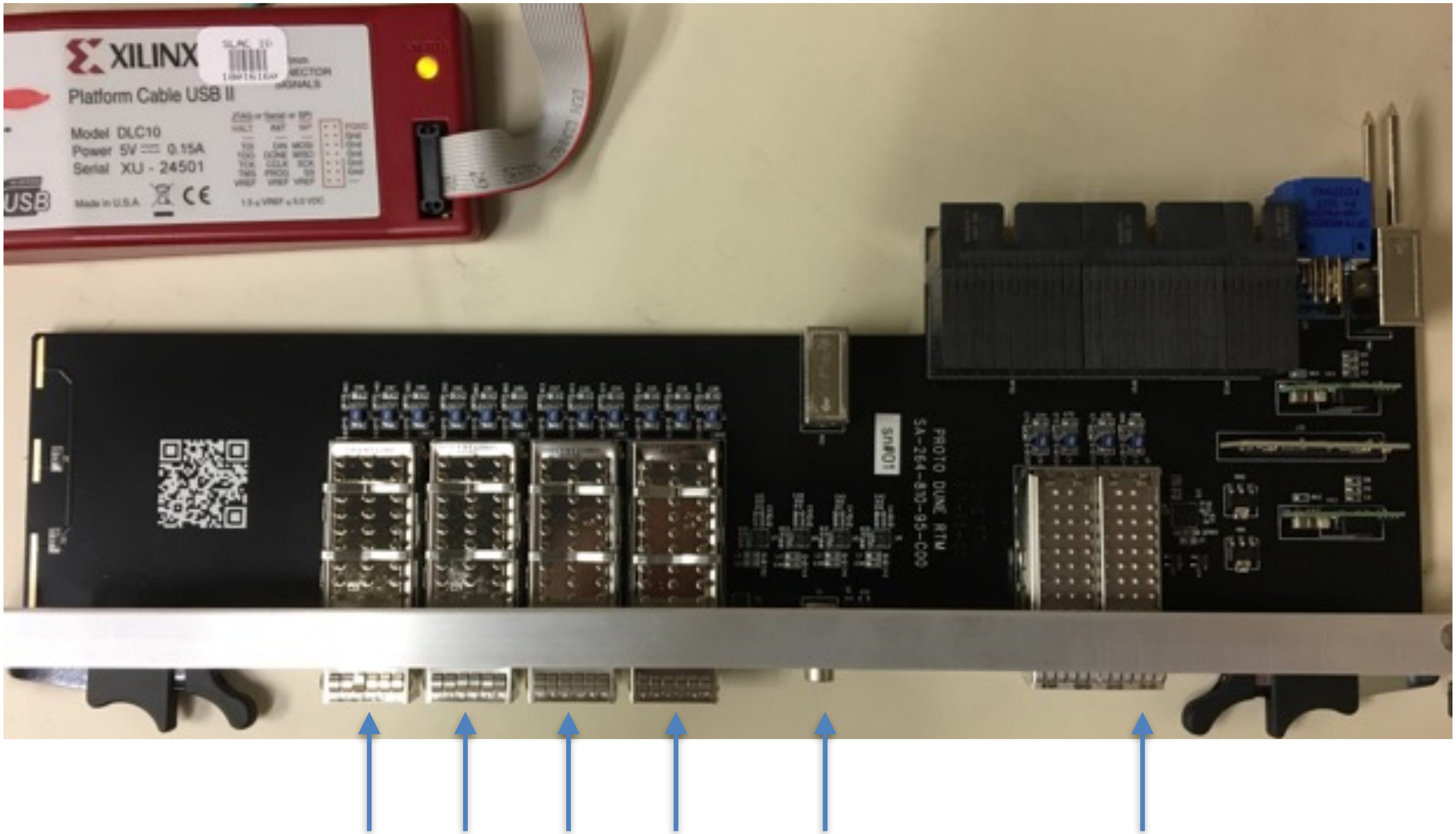
WIB-RCE physical interface



We route it so there is an option to run at either 2x8 Gbps or 4x4 Gbps out of the WIB FPGA and the RTM would be routed accordingly.

the WIB-RCE data format doc-db XXXXX

protoDUNE RTM (first pass)



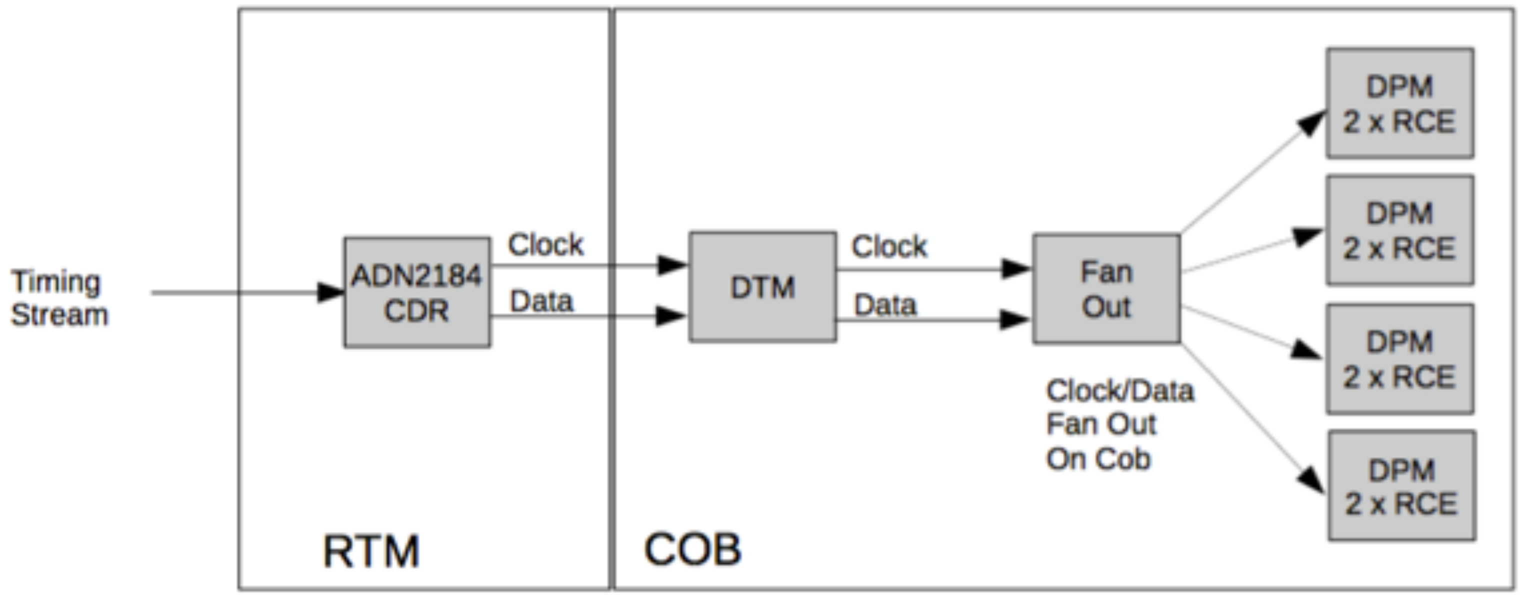
post schematics doc-db

from WIB

GPIO

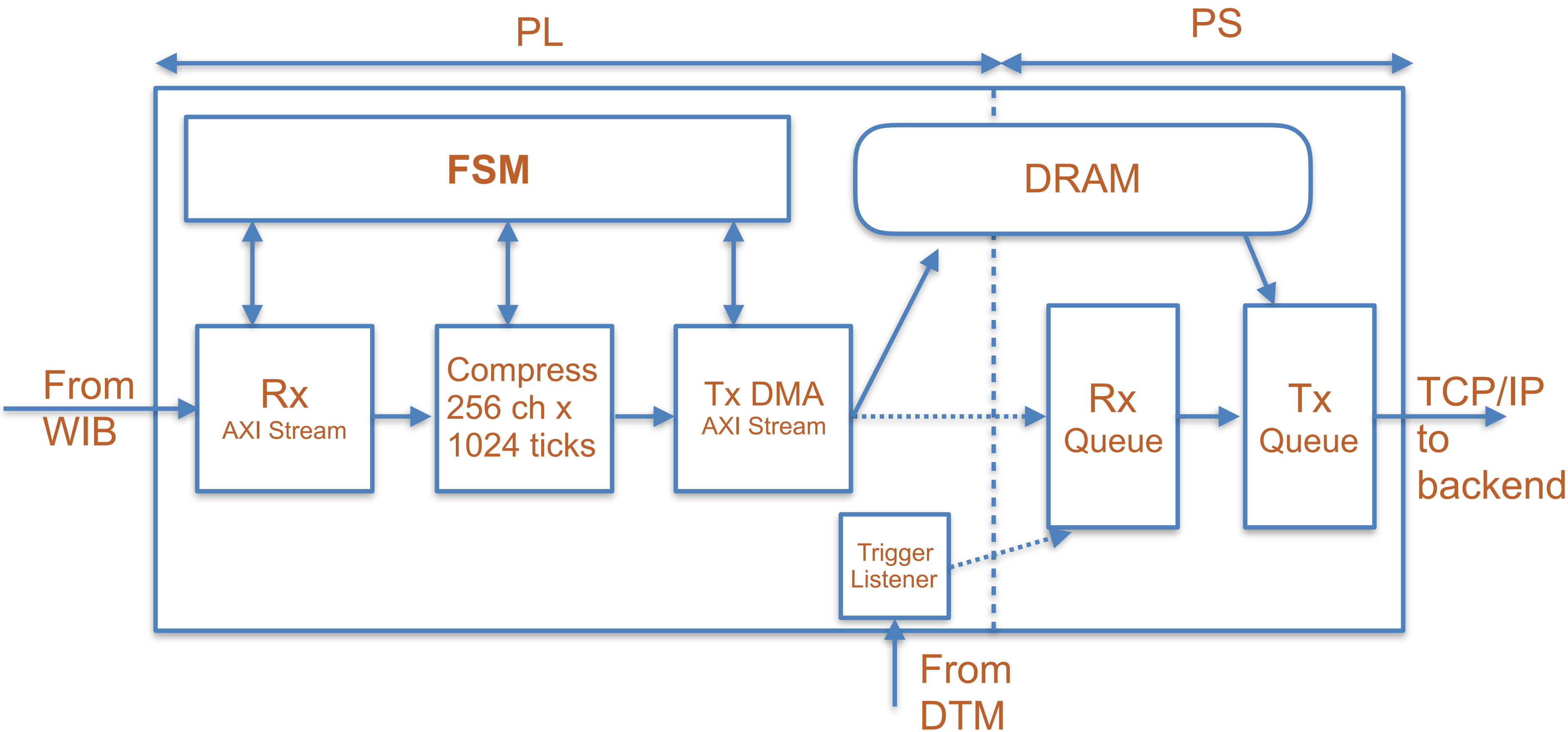
from timing

on-COB timing/trigger distribution



- Clock and trigger/timestamp data separated on the RTM and sent to DTM which fans them out to each RCE (…this is really what the DTM is there for…)

RCE data flow



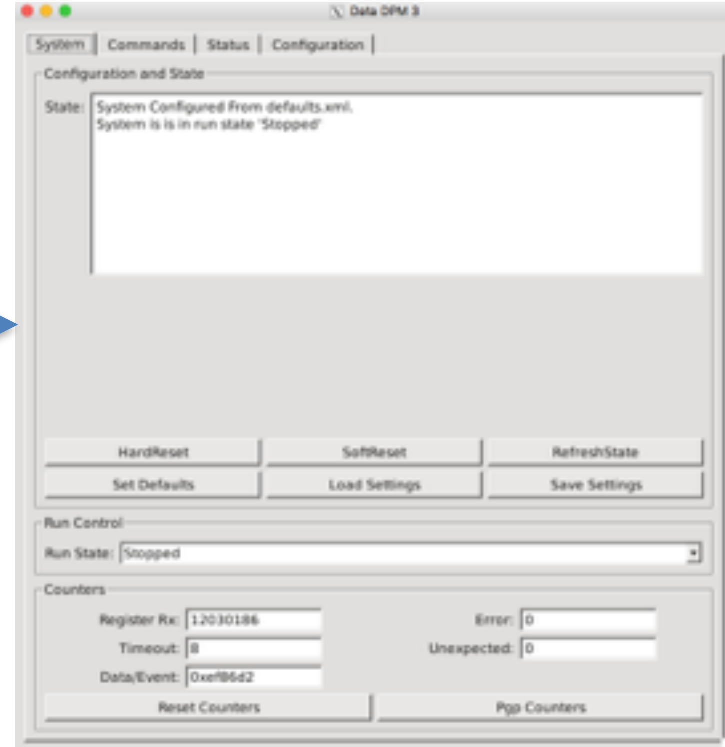
Compression & other tricks…

- SLAC is currently working on the compression firmware block
- we will use Arithmetic Probability Encoding (APE)…it's an entropy encoder like Huffman
- data will be blocked into 1024 tick chunks (0.5ms) and each chunk will have probability tables computed and data encoded before being "DMA'ed"
  - this is done on a per-channel basis, with large parallelization in the FPGA
- we are developing this using Vivado HLS … write the algorithm C++ and the package converts to VHDL, simulation, synthesis and testing.
  - Our experience has been fairly positive (we also wrote a waveform extraction IP)…you have to think a little different than programming for a PC though!
- also looking into implementing:
  - a hit finder that would go out as a separate stream, potentially used for triggering: Sussex/Oxford
  - pre-compression frequency filtering and/or coherent noise suppression (still lossless): UCDavis/SLAC

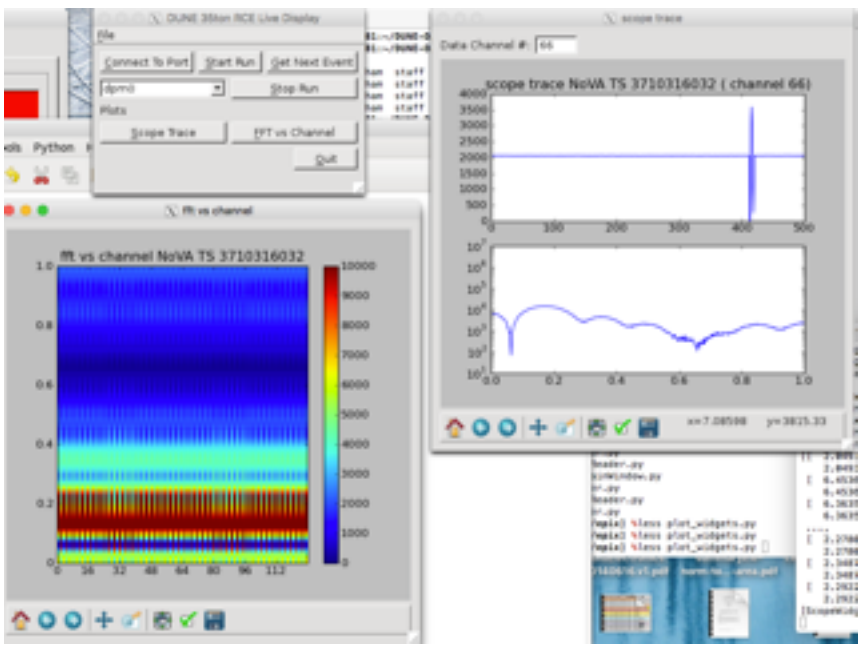
Backend software tools

We have tools that work independent of the official DAQ that are used for debugging & development…

Control & Status GUI



(very) simple run control & online monitor



protoDUNE RCE system numbers

	per RCE	per COB	per protoDUNE
# RCEs	1	8	60
# COBs	—	1	8
input data bandwidth	~7 Gbps	~56 Gbps	~420 Gbps
max output data bandwidth*	~0.4 Gbps	~3.2 Gbps	~24 Gbps
max in-spill trigger rate**	~130 Hz		
max steady-state trigger rate	~45 Hz		

* we currently use the 1 Gbps link to the switch but we're limited by the fairly inefficient ARM/archLinux TCP/IP stack…development is ongoing at SLAC to (a) implement hardware-assisted TCP/IP (pretty much ready) and (b) implement fully hardware based 10 Gbps ethernet using a reliable udp protocol (probably a few months off and not really necessary for us)

** assumes we send 5ms/trigger, x4 compression, and no out-of-spill triggers…as a reminder the baseline requirement is 25 Hz but we want to push this up to 50 Hz or even 100 Hz

link to simple rate calculator (ask me for permisison to edit)

RCE system production & testing

- the relationship between TID/AIR and DUNE is somewhat of a "producer/consumer" one… they provide the hardware and ***base*** firmware & software (for $$$, of course!) and it is all tested and validated
- hardware:
  - COB/DPMs— we have 3 full COBs (2 from 35-ton, 1 at SLAC); 5 additional boards have been ordered…the COBs are being loaded now, DPMs to follow soon; expected delivery (@SLAC) ~ mid-December
  - RTM — as shown, we have a prototype RTM…we've built 3 of them and will purchase more this FY after the first round of testing (in case we need to make changes)
- firmware:
  - base WIB-receiver and DMA engine firmware are ready now
  - pass-through + FSM firmware are ready now (use for WIB-interface testing)
  - compression firmware ~ 50% done; ready by mid-January
  - waiting for Bristol-provided firmware blocks before we work on DTM firmware
- software:
  - basic framework exists (part of base provided by TID/AIR)
  - specific data flow control, including triggering ~ 50% done; ready by mid-January

Interfaces

- Five interfaces between the RCE TPC readout and other (sub-)systems
  - TPC readout electronics
    - physical: multi-strand fiber with QSFP+
    - logical: WIB data format
    - first testing of interface currently proceeding at BU
  - Backend computing
    - physical/logical: SFP+/10Gbps ethernet to artDAQ boardReader
    - we work with Oxford/RAL/FNAL to get the board reader code for both the data receiver part and the RCE configuration
  - Timing/Trigger
    - physical/logical: SFP+/custom protocol (Bristol)
    - first testing will take place at Oxford
  - Offline/Online Monitoring
    - logical: data format & decompression routine
    - SLAC will provide interfaces ("getters") once things are a bit more settles; users will not need to know the ordering of bits

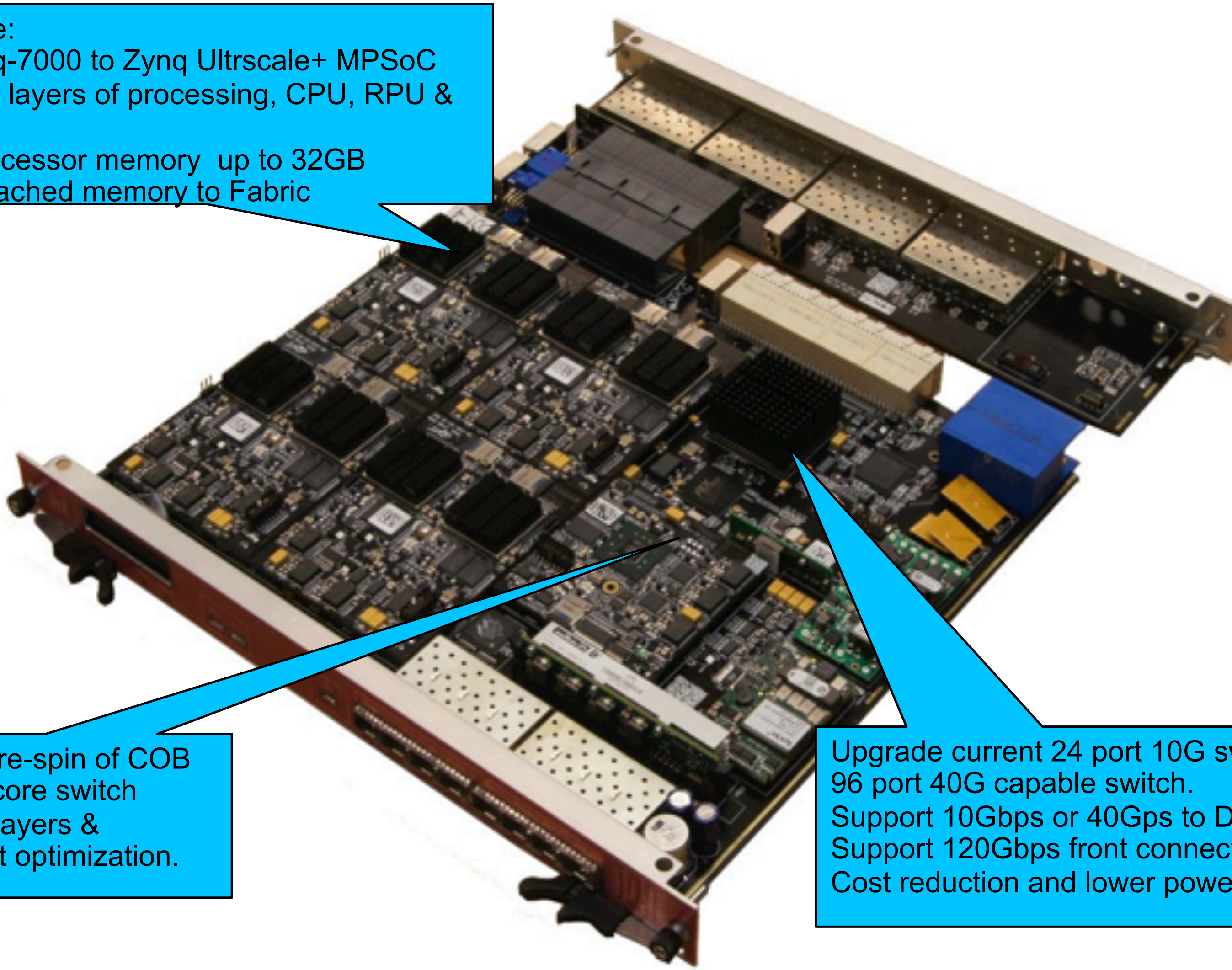
Conclusion

- The RCE system as designed should easily meet the science requirements for protoDUNE
  - We have experience with this system from 35-ton prototype and other experiments and a good team actively working on it
  - beyond the base requirements, we hope to test more advanced techniques (hit finding, noise filtering) that could be very useful for full DUNE
  - there is also a planned development to put the artDAQ boardReader directly on the RCE ARM
- We have a number of COBs out in the wild and the production for the remaining hardware has begun; testing and integration is currently happening at BU (WIB), Oxford (timing & artDAQ), and SLAC (compression and base firmware/software)
  - we should be well ahead of the game by the time of the VST ~mid January

Planned RCE-platform upgrades

***post protoDUNE*

DPM Upgrade:
Upgrade Zynq-7000 to Zynq Ultrscale+ MPSoC
3 layers of processing, CPU, RPU & GPU
Additional processor memory up to 32GB
Add direct attached memory to Fabric



Cost reduction re-spin of COB coincides with core switch upgrade. Less layers & component cost optimization.

Upgrade current 24 port 10G switch to 96 port 40G capable switch.
Support 10Gbps or 40Gps to DPMs
Support 120Gbps front connection
Cost reduction and lower power