

DAQ WG Status Report

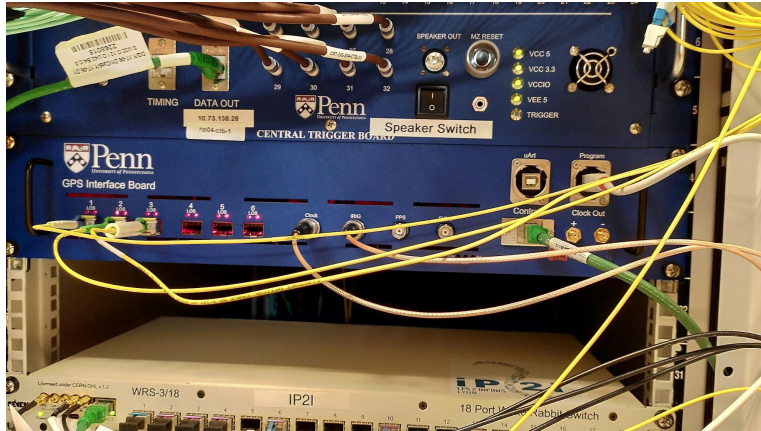
27/8/24

Nicolò Tosi for the DAQ WG

Updates primarily from visit at CERN last week

Observed and operated ProtoDUNE timing system components, with

- David Cussans, responsible for the timing system
- Alessandro Thea, responsible for the FD DAQ



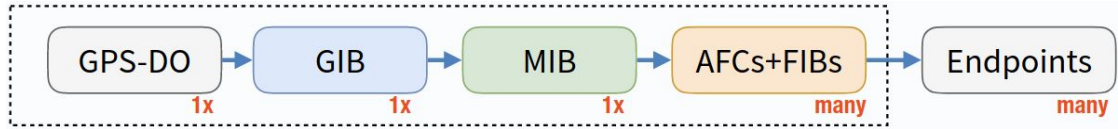
GIB



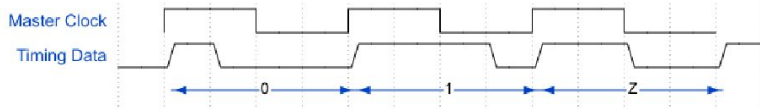
Temporary 1:8 Fanout

-Reminder of Timing architecture

uTCA Crate (1 for all NDs)



Fiber transmits encoded clock (62.5 MHz and data (timestamps, sync command, spill, ...))



Endpoints transmit back for delay compensation

Endpoint reference design available as FMC



8 Fibers per FIB, 8 Endpoints per fiber



What we learned in extreme synthesis

- The hardware (for the timing system) is simpler than expected
 - Mostly because almost everything is actually done in software, including time alignment
 - **It also does not do everything we want it to**
- The software is more complex than anticipated
 - Perhaps unnecessarily so for the ND
- The DAQ group has no resources to spare for SAND
 - To the point that they asked us to pay for test hardware (AFAIK, FD and NDLar got it for free)
 - **... and when it was time to actually provide it, they did not, citing critical lack of spares.**
 - They did say it may become available again after end of NP04 run, maybe...

Timing hardware and firmware

- The complexity of the timing receiver has been further reduced w.r.t. what we knew. The CDR chip has been eliminated.
 - Integrating the timing receiver on our on-detector electronics will be almost trivial.
 - One timing endpoint costs <100 EUR in dedicated components.

- Firmware code is well organized and modular.
 - Built with ipbb with an automated toolchain (originally for CMS).
 - It would be nice to have the same thing, but it's not free (in terms of labor).
 - Easy to integrate in our boards as long as hardware is similar

Timing hardware and firmware

- Biggest known problem is with I2C communications in some boards
 - We encountered this live, in the “spare” boards we were supposed to get.
 - This depends on the board design and will likely not be present in our own if we do it well.
 - We could also use SPI instead, more custom code to re-write though

- We have obtained access to
 - design files for the reference endpoint <https://github.com/uob-hep-cad/uob-hep-pc069>
 - Fw source for the various boards <https://gitlab.cern.ch/dune-daq/timing/dune-timing-firmware>
 - .bin <https://pdts-fw.web.cern.ch/pdts-fw/index.php?p=tags%2Fv7.3.0%2Fpipeline7313687>

A critical feature is missing from the timing System

Or rather, originally implemented and then removed...

The timing system as is does not distribute accelerator state as commands

- No start of spill
- No end of spill
- No “beam will come in xxx ms” warning
 - GRAIN **needs** the latter for ASIC power gating

D. Cussans said these features were removed as “unnecessary for FD”

Absurd to me that ND or at least SAND were not asked about this?!

We must have this feature back (and updated to new ACLK)

Software

- Everything has a dependency on the whole DAQ development environment.
 - Everything! Even reading the firmware version from the FPGA requires installing the whole dune daq from cvmfs
 - Nontrivial barrier to entry in terms of required OS, expertise, time
- On the other hand, you get a rich environment with lots of tools.

- Timing is controlled through a single application “dtsbutler”
 - However, this is made for the timing system NOT for the endpoints.
- I have not yet begun to explore the rest of the DAQ software.
 - Help most likely needed from actual software experts to move forward.

DAQ Architecture: more paths possible

- Original plan was to go with Zynq with links
 - 1 GbE to the PS-CPU, for control, monitoring, local acq... using any software protocol we like
 - 10 GbE to the PL-fabric for data. Firmware block (IP core) provided by DAQ
 - This closely mirrors the FD, but is a huge overkill for us. Also costs somewhat more.

- A. Thea however is perfectly happy if we do 1 GbE for both
 - or even just a single 1 GbE

- They also don't care whether we use a Zynq, a CPU, an FPGA or a toaster...
 - Zynqs however have no downsides I can think of.
 - AMD just announced support extended until 2045 (!) for the current generation Ultrascale+.

DAQ Architecture: more work needed

- The original plan (1GbE+10GbE) requires
 - Least “upfront” work for GRAIN and STT
 - No idea for ECAL, may or may not be compatible with CAEN concentrator
 - Highest cost (but we are only talking 30-50k difference in parts for all of SAND)

- Other options offer different compromises.
 - It is definitely possible to make something that is simpler and cheaper
 - Saving potentially up to 1kEUR/DAQ endpoint
 - Requires more original work starting now (or rather next year)
 - Our system will be more tailor-made, with all the pros and cons

Outlook and plans

- It is necessary to have a high level discussion within the ND community on the need for Accelerator Timing and info. This needs to happen soon.
- I could not organize this at the last CM, few people were responsive on this issue.
- **Resources need to be allocated to ensure that the Timing system actually meets our requirements. At the moment it does not for what concerns accelerator interface.**

Backup

Exercise: timing circuit for GRAIN Warm Interface Board

Based on the latest reference design, I drafted this design and showed it to David, which agreed it was sound. Using a single PLL is beneficial for phase noise.

