LUXE DAQ: Syncronisation

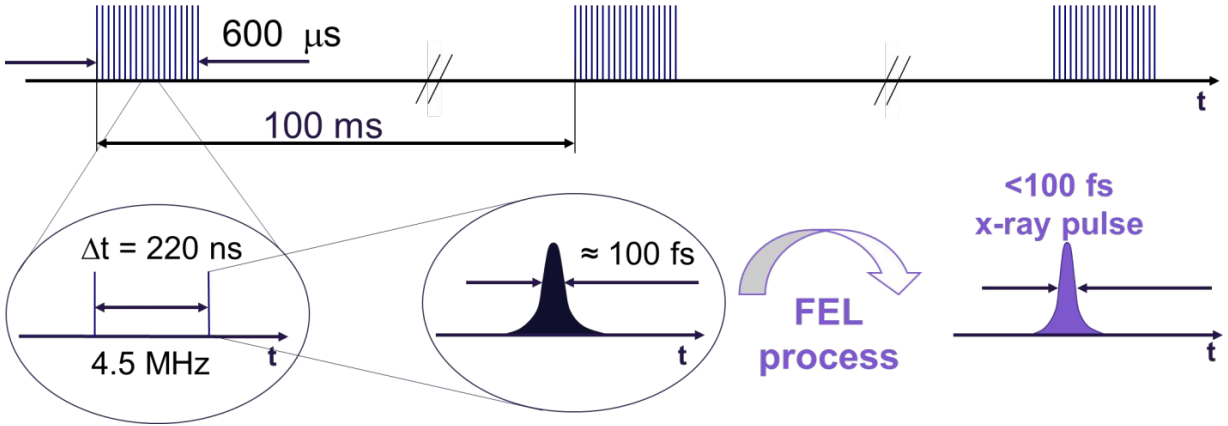
Matt Warren, Matthew Wing
UCL
1 Feb 2023

Introduction

- LUXE needs to interface with the machine
- LUXE has sub-detectors with different needs
- Presented here are the beginnings of a system design
  - Call it a “provocation”!
- We would like to find out what is expected/needed/available for each sub-det
  - Timing resolution
  - Synchronisation
  - Hardware interface
  - Busy (back-pressure)?
  - Needs for “upstream” communication
  - Scale - number of timing links needed

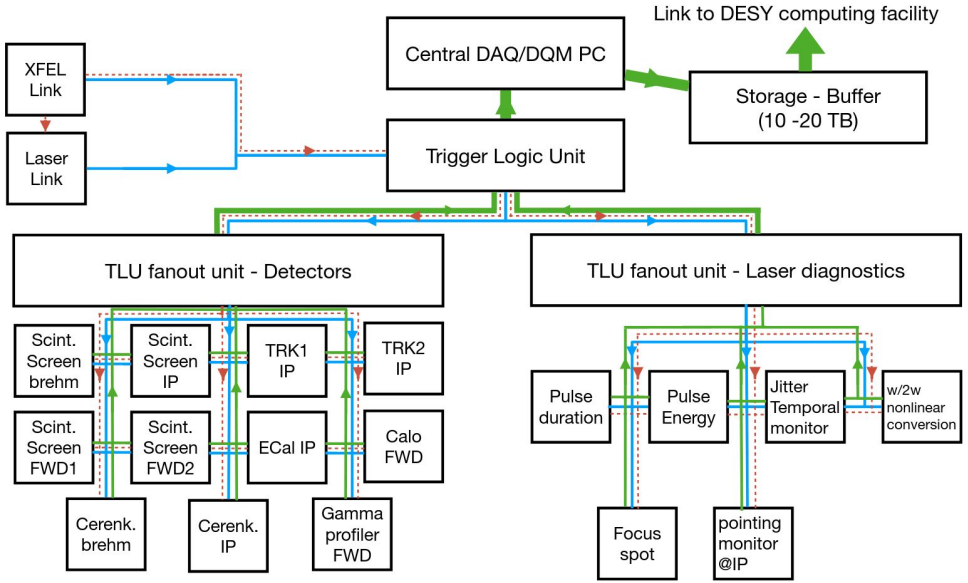
Reminder - bunch structure

- 4.5 MHz bunch clock
- Bunches in “trains” at 10 trains/second
  - 2700 bunches per train
- LUXE uses 1 bunch/second (1Hz laser)
  - I.e. 1 bunch from 1 in 10 trains
  - (called “LUXE-bunch” here)
- Readout is at 10Hz, matching bunch train frequency
  - Calibration triggers etc.



Control system overview

- *“Common aspects”* technote system is based on the AIDA2020 Trigger Logic Unit - **TLU** *
- Mainly the TLU provides interfaces and logic for clocks, triggering, other controls and busy
- TLU stores book-keeping info on triggers for readout with data
- LUXE needs to extend TLU functionality
- Sub-detectors need to implement a hardware interface to it



* https://indico.desy.de/event/33748/contributions/119097/attachments/72419/92848/LUXE_tech_note___common_aspects_21Mar2022.pdf

Control system - *Machine interface*

The LUXE-TLU is expected to:

- Receive the bunch clock
  - 4.5MHz
  - Multiply as needed, with means of extracting original clock
  - Or receive already-multiplied clock + sync signal
- Receive a LUXE-train-start, or “LUXE-bunch” trigger
  - Synchronous with the XFEL+Laser
- Receive the XFEL-train-ID?
  - Additional trigger meta-data
  - Could be used to determine LUXE-trains?
- The physical interface(s) for this still need to be identified
- The Laser system seems best placed to forward all timing signals to the TLU?

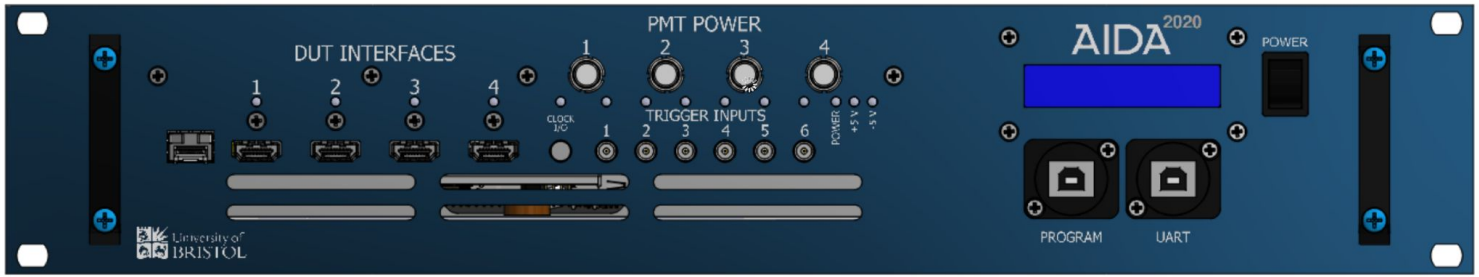
Control system - *Sub-detector interface (+stand-alone)*

The LUXE-TLU is expected to:

- Send/forward LUXE-bunch triggers to sub-detectors
  - And train-start etc. as needed
- Send trigger-ID, including trigger-type (~ trigger source)
- Provide means to delay triggers to synchronise with next train(s) - e.g. provide early trigger/sync pulse
  - May need to be part of sub-det receiver logic for fine tuning
- Receive external signals (e.g scintillator/cosmics) to generate local trigger
  - Make use of coincidence logic if needed
- Provide software interface (via IPBus)
- Generate stand-alone/local triggers (under software control)
  - Also any train-IDs etc, if used
- Distribute timing signals via fan-outs
- Receive a busy reception from sub-dets
  - Stops trigger distribution (and generation)
  - Via fan-ins

AIDA2020 TLU timing system

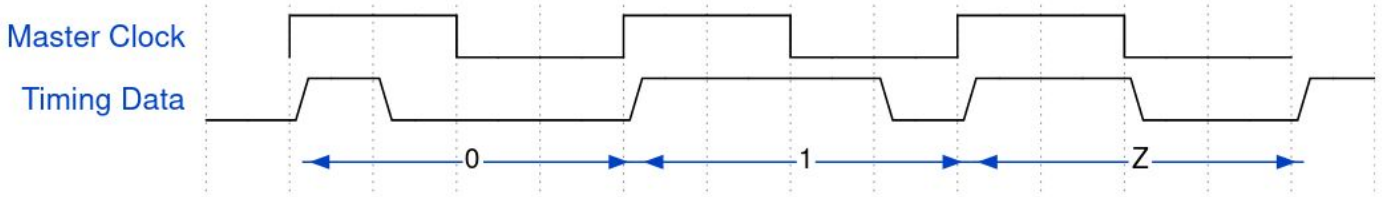
- Well known in the community (i.e. not too much detail here)
- Receives or generates clock (incl. multipliers etc.)
- Interfaces to trigger sources via Lemo inputs
  - Can be used to accept machine signals with updated firmware
- Triggers and control info distributed electrically
  - Custom electronic interfaces needed
- Can distribute trigger numbers, but little more
- Busy options are present
- Fanout options available (but I ***think*** only Busy fan-in)



See: *The AIDA-2020 TLU* https://iopscience.iop.org/article/10.1088/1748-0221/14/09/P09019/pdf

(proto)DUNE timing system

- The TLU has been adapted for use by protoDUNE
  - Alternative firmware
  - Evolved into DUNE Timing System
- Triggers and control info distributed via fibre (SFP) + fanout/in
  - 8b/10b encoded data - can send more than triggers/IDs (asynchro)
  - Duty Cycle Shift Keying used to allow clock recovery to be performed on FPGA
    - Does not require extra CDR hardware (FPGA + SFP is sufficient)
    - <200ps clock jitter (~20ps with dedicated CDR hardware) [predicted]
  - Can be optically split
  - Bi-directional interface
    - Sub-det can send data to TLU
    - If optically split then bi-dir then needs higher level time-slice allocation, reducing BW

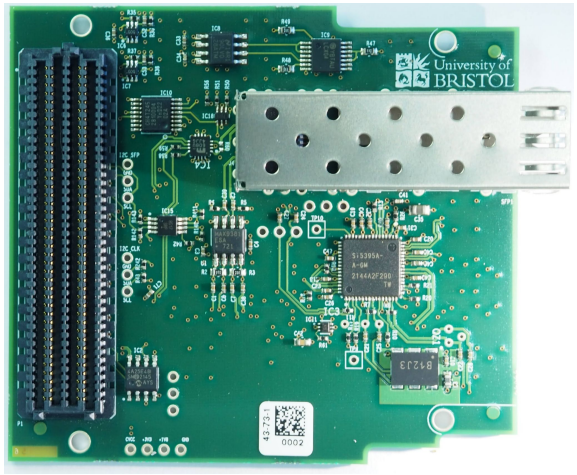


Duty Cycle Shift Keying (DCSK) showing the representation of '0' , '1' and 'Z' *

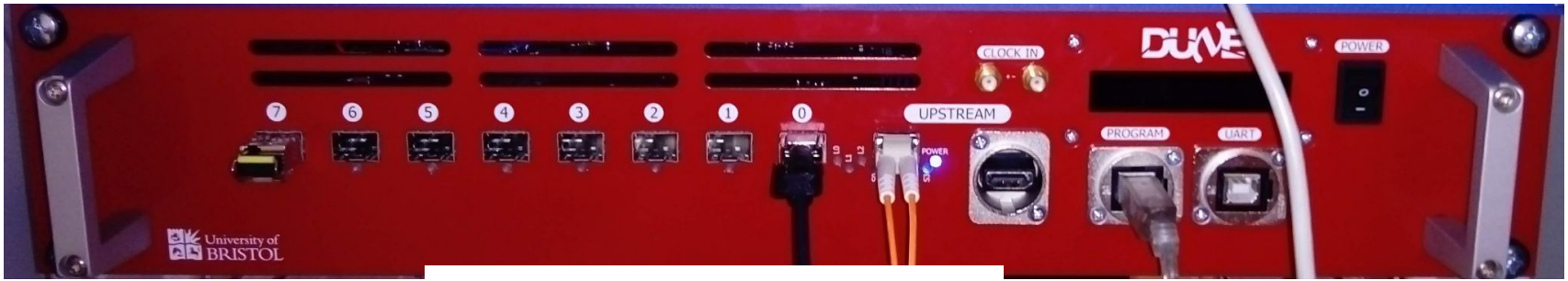
* from: *Timing and Synchronization of the DUNE Neutrino Detector* https://arxiv.org/pdf/2210.15517.pdf

(proto)DUNE based hardware

- Some hardware is available now (thank-you Bristol/DUNE!)
  - "Reference hardware" FMCs that can act as either timing master or endpoint (depending on firmware)
  - "Fanout" units that can drive up to 8 SFPs (x8 if optical splitter used)
- These will be used to create a test system
- And possibly be part of the final system!



Reference hardware FMC



Fanout

Initial conclusions (what does all this mean??)

- We need to understand machine interface better
  - Physical: Optical, electrical? LVDS? etc.
  - Signals: Clock, sync? etc.
- We need a TLU for cosmics
- The fibre based (DUNE) system is desirable for distribution to sub-detectors
  - Simplified hardware
  - Additional bandwidth for distributing data (in both directions)
- Customisation of the firmware will be required in all cases
  - Probably need to find someone for this
- Hardware exists for initial testing but more will be required
  - DUNE fanouts exist but need more TLUs (expecially for labs)
- Sub-detectors need to implement (FPGA) hardware interfaces to talk to the control system
  - With SFP for fibre
  - Plus dedicated CDR chip for best timing precession
- May need fine and course delays to align with laser
  - Achievable with FPGA resources

Conclusion: A possible system design…

- TLU receives from the machine/laser:
  - LVDS Clock
  - Bunch or train sync pulse
  - Train-ID (serialised)
- TLU distributes to sub-dets, via a fanout:
  - A clock synchronous with the XFEL bunches (down to 20ps level if needed)
  - A trigger synchronous with the machine (or other sources)
  - Trigger identification number and trigger-type
  - Async data at ~5MBytes/s (depending on clock multiplier)
- Sub-detectors implement required hardware interface
- Sub-dets send back
  - A busy
  - Async data at ~5MB/s divided by fanout factor (what will this be?)
- Back to the original questions - will this cover needs?
  - Timing resolution
  - Synchronisation information
  - Hardware interface
  - Busy
  - What data needs to be sent upstream?
  - Number of timing links needed by each sub-detector