Timing and synchronisation of the DUNE far detector

Stoyan Trilov, for the DUNE Collaboration

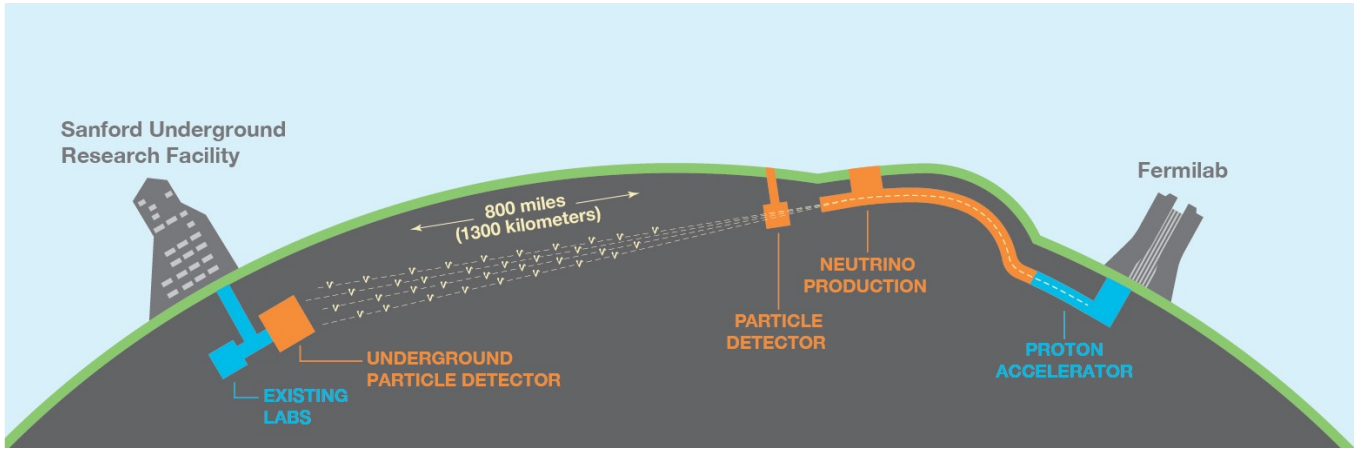
Technology in Instrumentation & Particle Physics (TIPP) Conference 2023

06/09/2023

University of BRISTOL

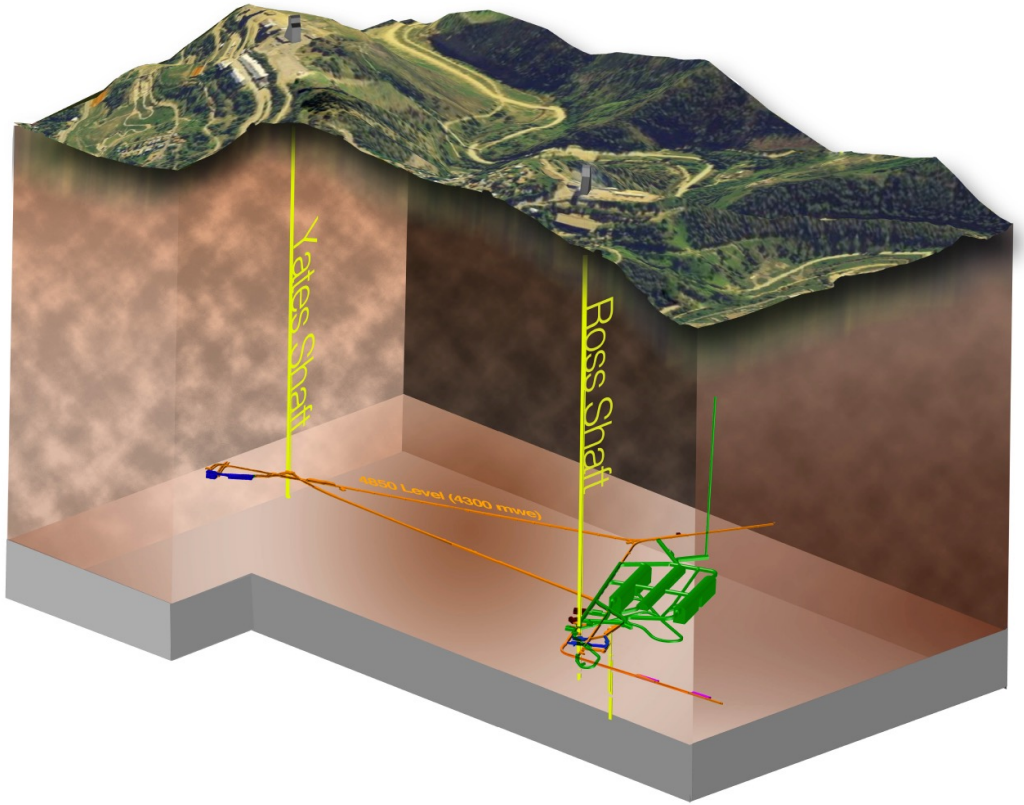
Deep Underground Neutrino Experiment

- Physics goals
  - Characterise neutrino sector properties: δ_{CP} and mass hierarchy
  - Supernova neutrino detection and measurement (expect to see ~1 in lifetime of DUNE)
  - Beyond standard model: e.g. proton decay
- Detector
  - Neutrino beam produced at Fermilab
  - Beam characterised by near-detector at Fermilab
  - Far-detector modules at SURF records neutrino interactions



DUNE far detector

- Far detector comprises of four 17kt liquid argon time projection chambers (LArTPC)
- Cold and warm custom electronics for charge and photon readout
  - TPC signal sampling: 14 bit @ 1.95 MHz
  - DAPHNE signal sampling: 14 bit @ 62.5 MHz
- Synchronisation considerations
  - Large, physically separated detectors
  - 1.5 km underground
    - Issues around accessibility, space, and power
  - 1300km away from neutrino source, and near detector
  - High uptime requirements (>98% for whole detector)

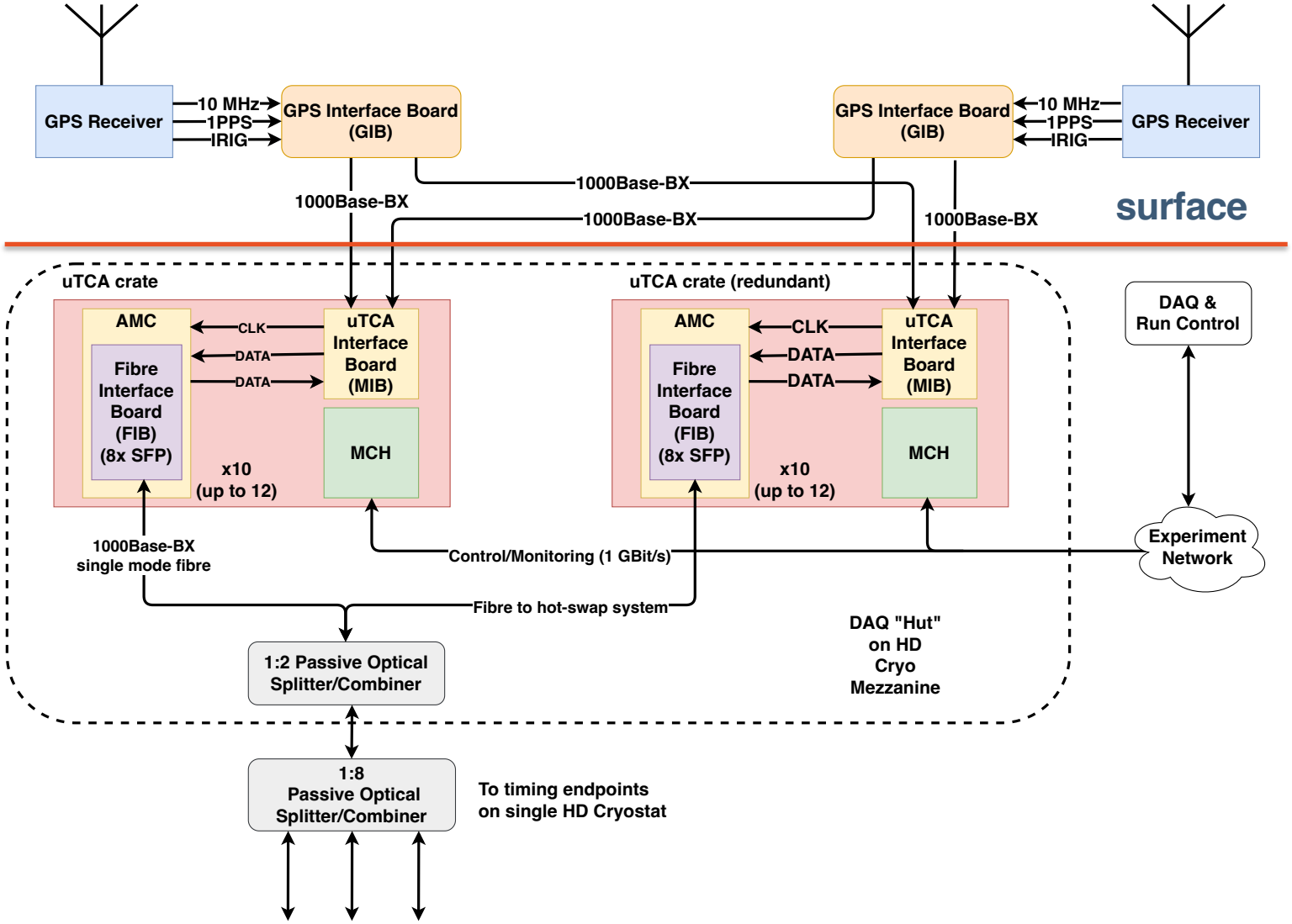


DUNE timing system (DTS) requirements

- Distribute a common 62.5 MHz clock and synchronised 64b timestamp to DUNE readout electronics
  - 62.5 MHz clock derived from the 10 MHz provided by GPS receiver
  - DTS timestamp initialised from GPS time
  - Relationship between DTS timestamp and GPS time continuously monitored and recorded
  - DTS timestamp can be translated to TIA/UTC “offline”
- O(μs) synchronisation between DTS timestamp and GPS time
- O(ns) synchronisation between DTS timestamps within a far detector module
- System uptime of >99%, driven by high-level DUNE’s uptime requirements
  - Redundancy and extensive monitoring designed into system from inception

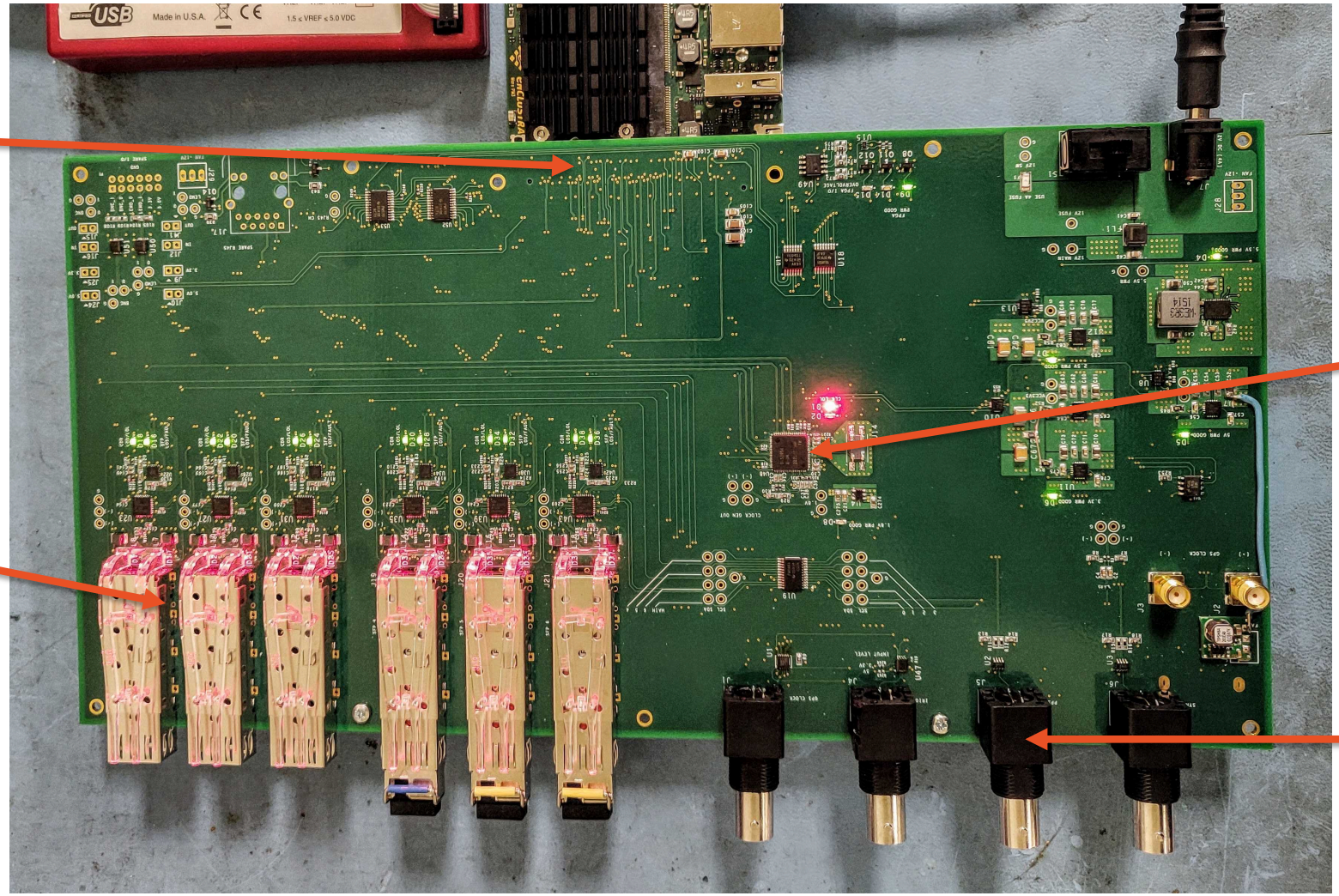
DTS topology

- **GPS interface board (GIB):**
  - Receives clock and time from GPS receiver
  - Generates DUNE clock and timestamp
- **uTCA interface board (MIB):**
  - Receives data-stream from GIB
  - Fan out clock and data to FIBs (via the AMCs)
- **Fibre Interface Board (FIB):**
  - An FMC with 8 SFP modules
  - Hosted by an AMC in a uTCA crate
  - Fans out timing data-stream to timing endpoints



GPS Interface Board (GIB)

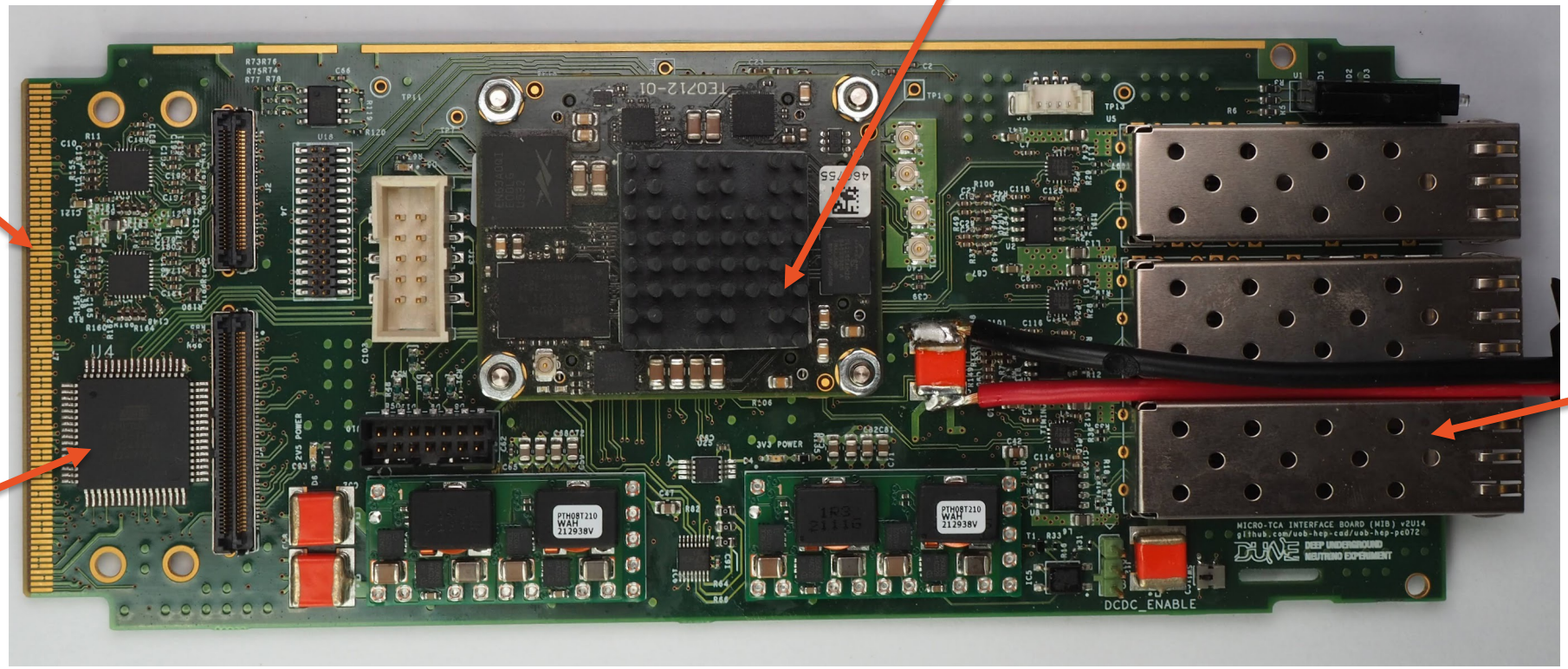
- GIB plugs onto an Enclustra Artix-7 A35 FPGA (via FMC connector)



- On-board PLL
  - SI5395
- SFPs
  - DTS IO
- BNC connectors
  - 10 MHz clock
  - IRIG-B
  - 1PPS

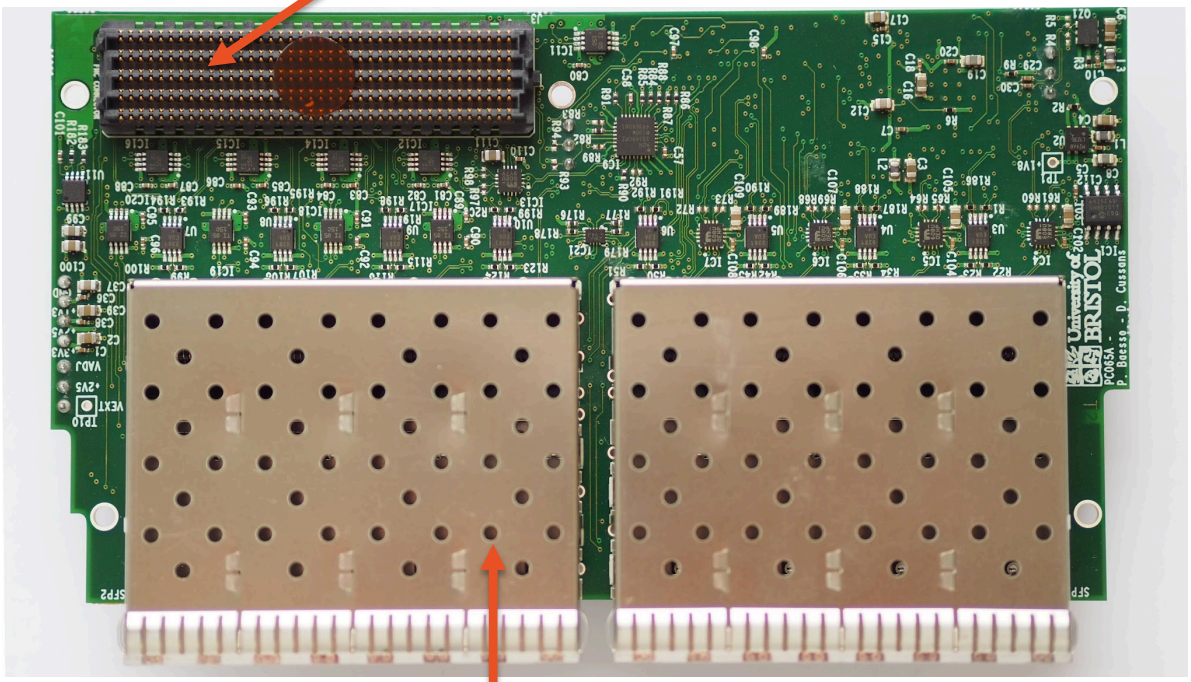
uTCA Interface Board (MIB)

- SI53342 clock buffer ICs
  - clock fanout to backplane
- FPGA connector
  - Trenz TE0712 FPGA board- Artix-7 A200
- SI5395 PLL IC

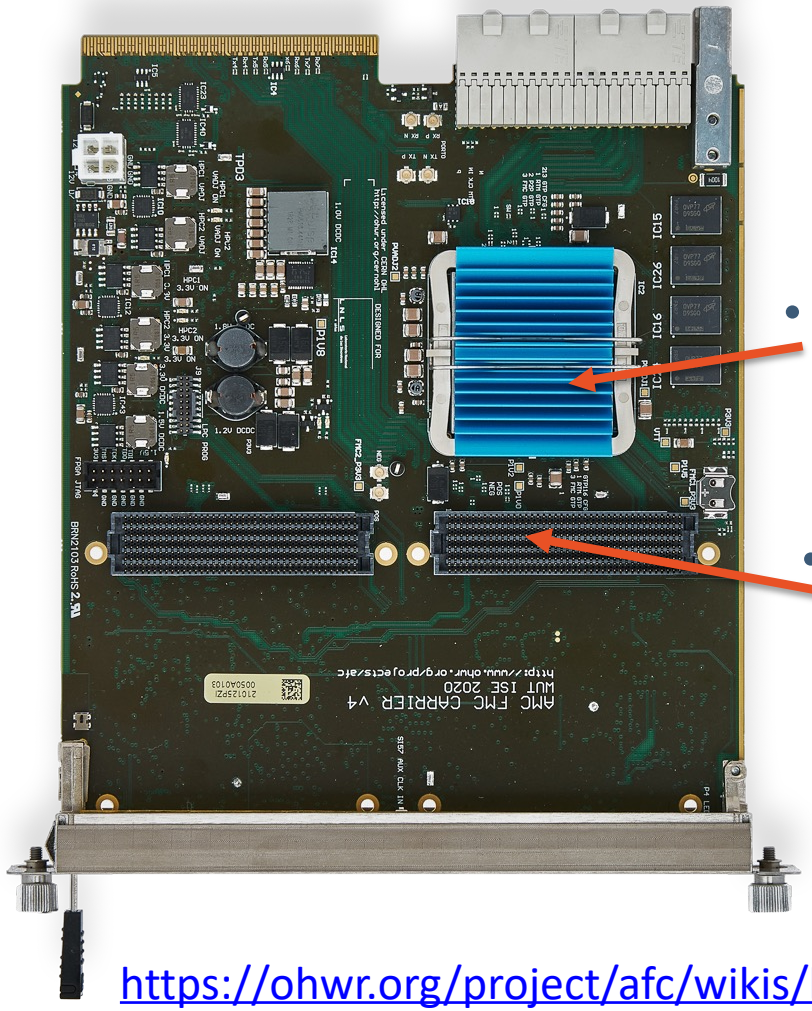


Fibre Interface Board (FIB) and AFC v4

- LPC FMC connector



- 8 SFPs
  - DTS IO

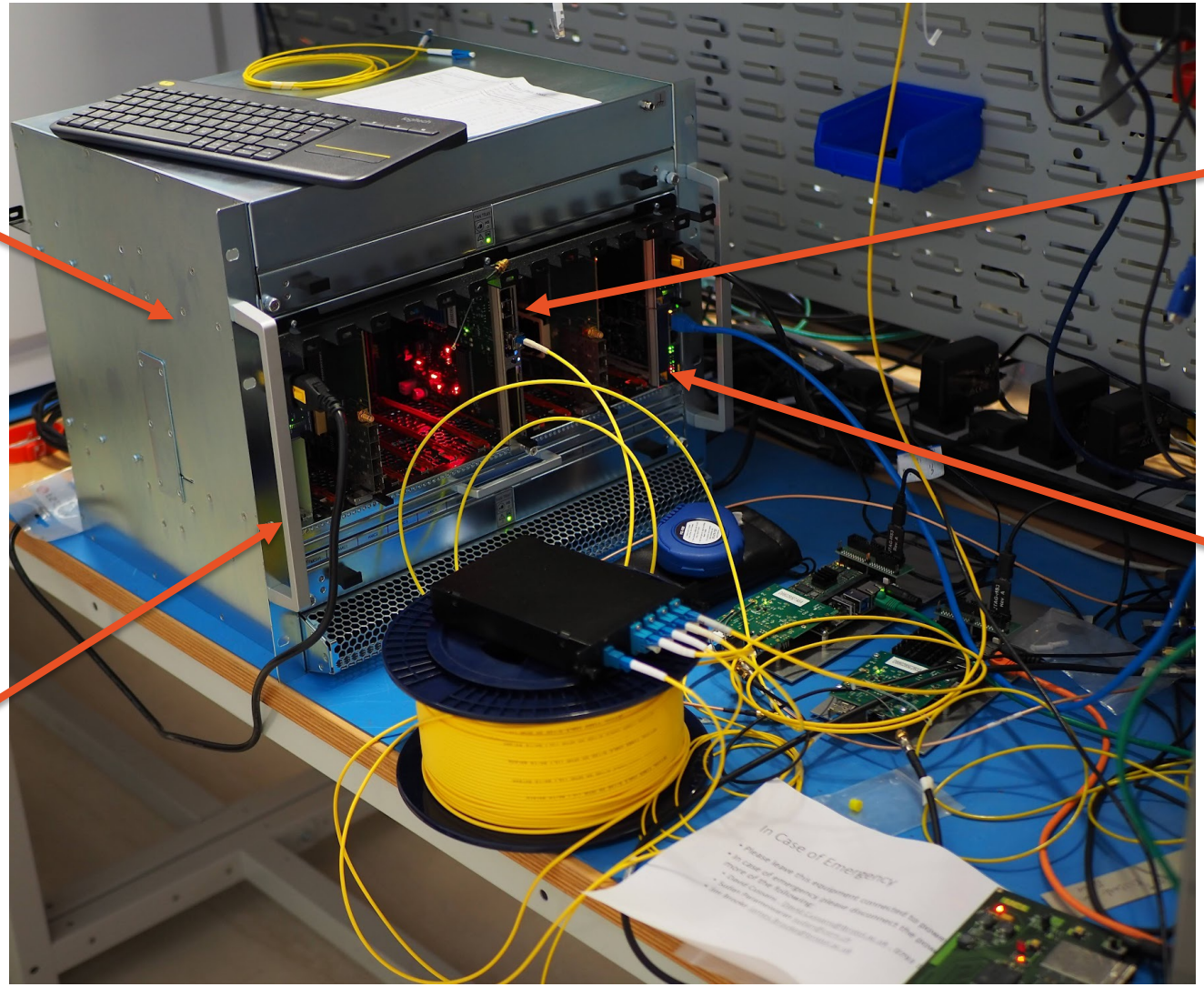


- Xilinx Artix-7 A200 FPGA
- 2 HPC FMC connectors

https://ohwr.org/project/afc/wikis/home

uTCA crate

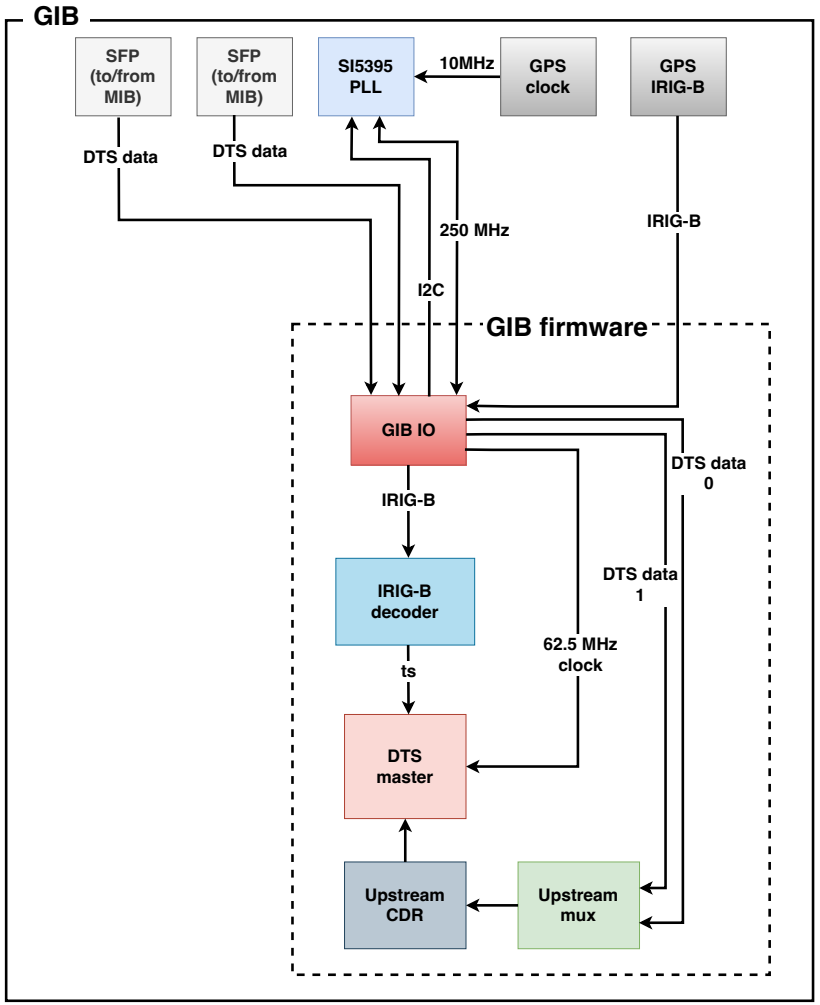
- uTCA crate
- AFC into an AMC slot
- MCH slot for MCH
- MCH slot for MIB

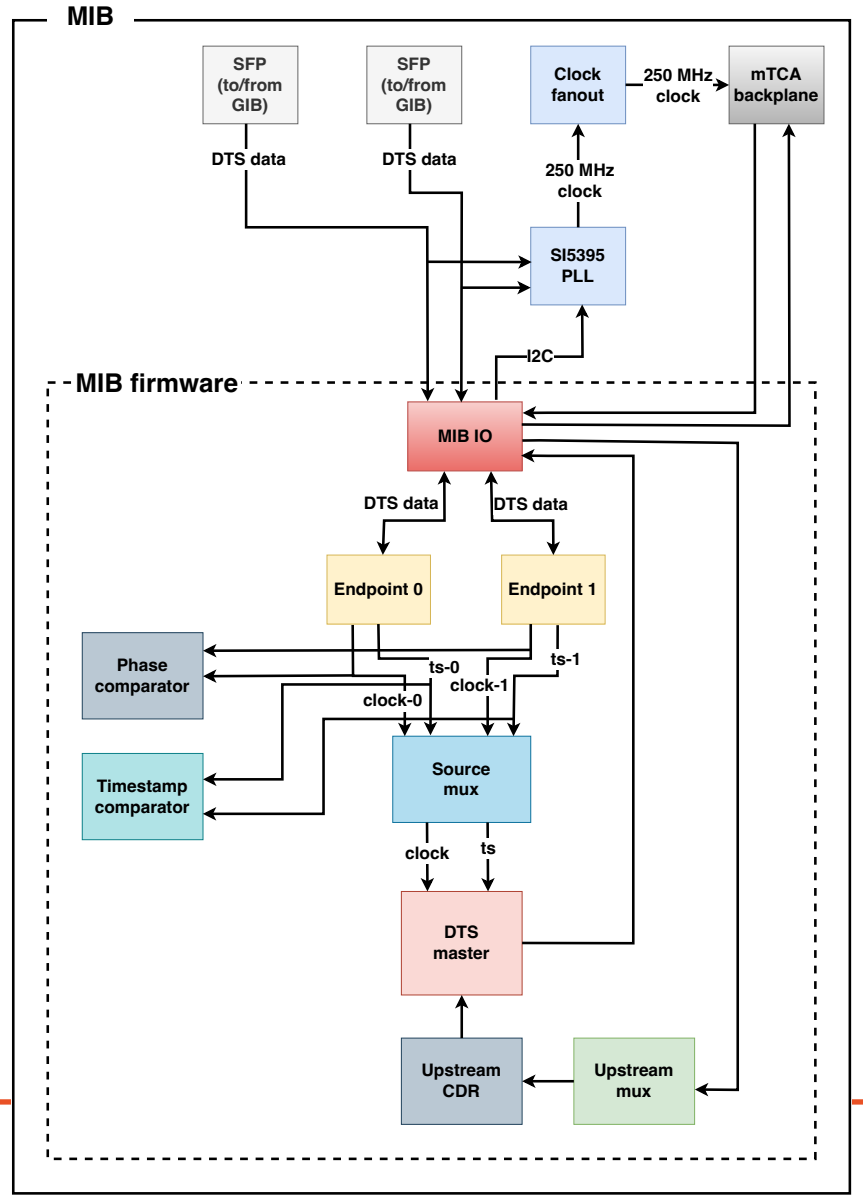


Firmware

- Highly modular architecture
- Several core firmware blocks with well-defined functionalities combined in different configurations to produce top-level designs with varying functionalities on varying hardware platforms
  - Master, endpoint, upstream CDR, IRIG-B decoder, hardware IO
  - Endpoint block provided to end-users (RO firmware developers) for integration into their own readout firmware designs
- Makes use of Duty-Cycle-Shift-Key encoding
    - Clock recovery in FPGA without dedicated external IC
- Ability to send synchronised addressed messages from master → endpoint via software interface
- Adjustment for different fibre lengths in 2ns units
- Scriptable project management and building using the IPbus Build tool
  - https://ipbus.web.cern.ch/doc/user/html/firmware/ipbb-primer.html

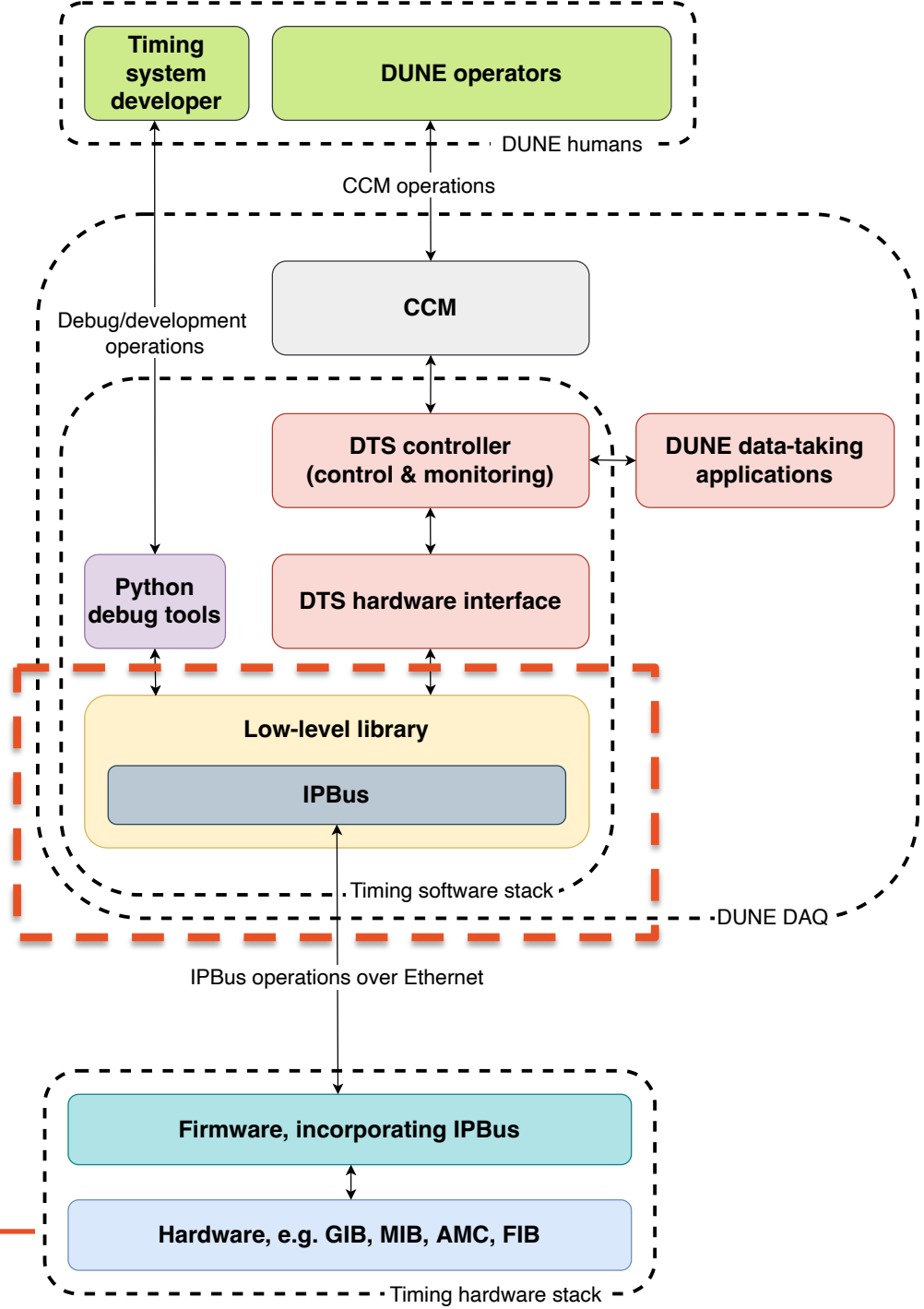
Firmware design examples





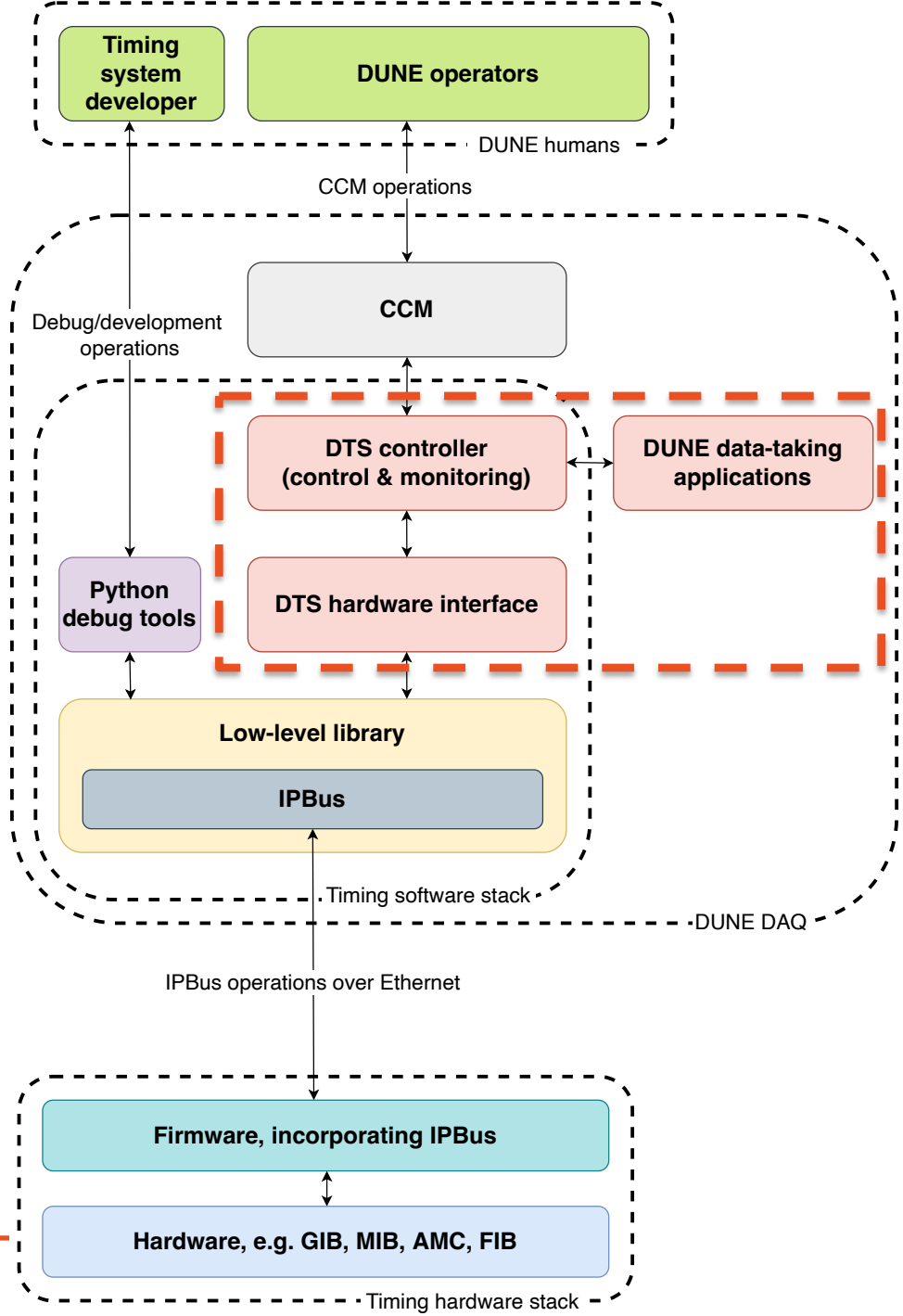
DTS low-level software

- C++ library provides OOP low-level interface to firmware registers
- Each firmware block has corresponding C++ class
- Register read and writes grouped together to form logical member methods, e.g. reset, configure
- Several methods from several classes aggregated under an umbrella top-level design firmware class
- Top-level interface classes rely on inheritance to facilitate code re-use and consistency of interface



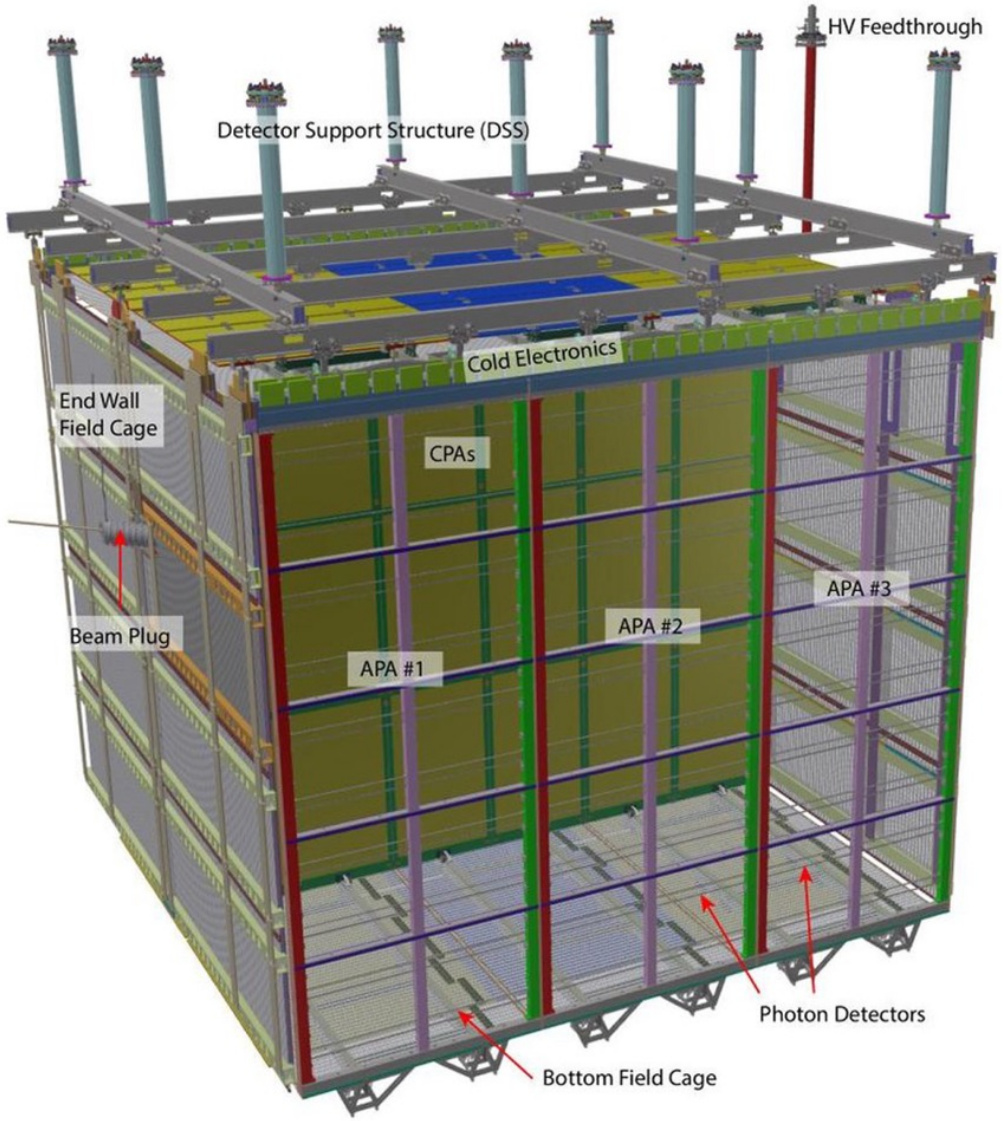
DTS high-level software

- For each DTS device, there is a DTS controller providing a control and monitoring interface
  - Collects operational monitoring information from firmware & hardware, parsing it and updating its state accordingly
  - Receives high-level commands from run/slow control, and translates them to the corresponding low-level commands (C++ library calls)
- System manager controllers co-ordinates actions between controllers for tasks involving multiple hardware devices, e.g. round trip time measurement
- Monitoring and control systems responsible for initiating any kind of redundancy operation → DTS controllers responsible for providing necessary summarised information to make decisions



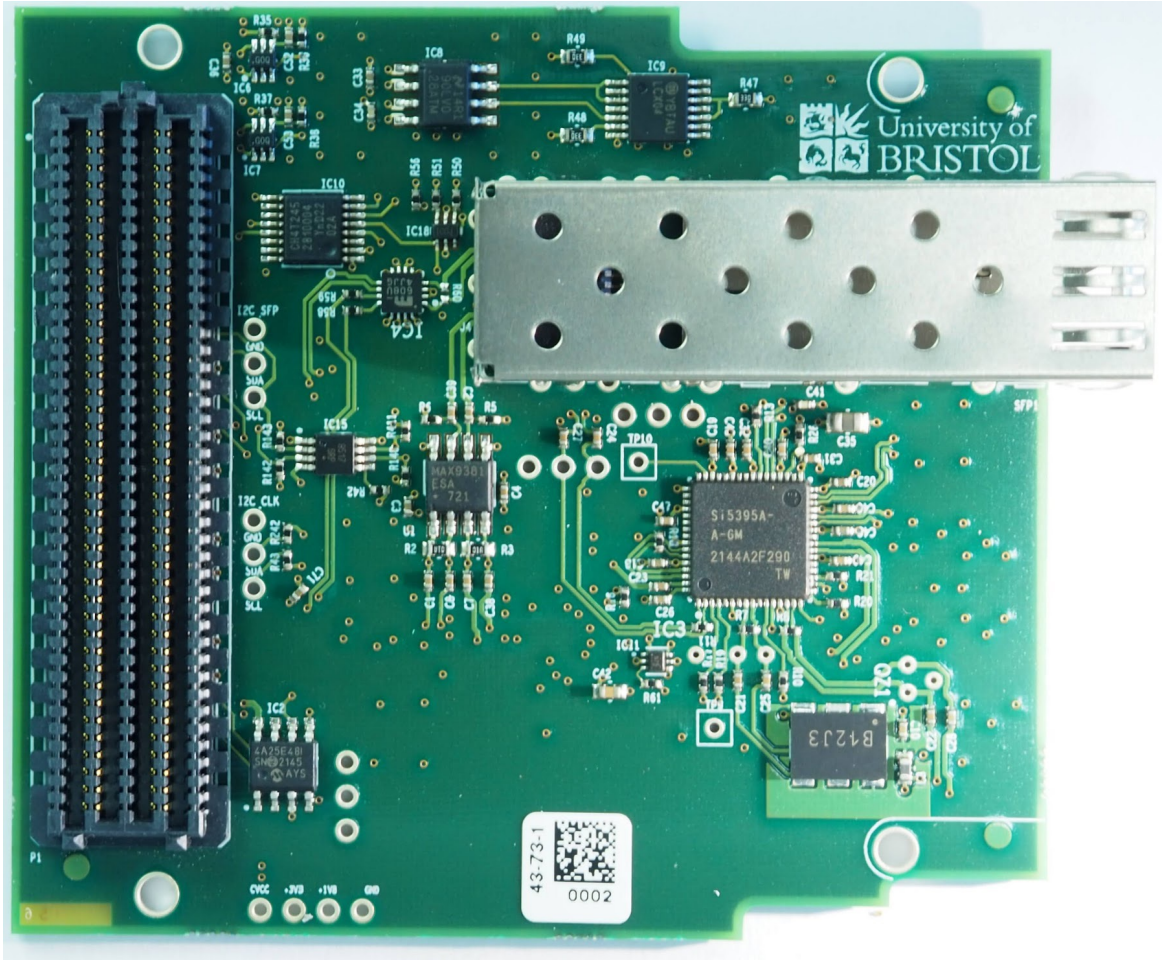
DTS prototypes

- ProtoDUNE HD is 0.77kt LArTPC based at CERN
- DTS prototypes/precursors successfully used to in ProtoDUNE run I
  - Master board based on already existing AIDA TLU (analogous to GIB+MIB)
  - Three active fanout boards (precursor to FIBs)
  - Prototype firmware and low-level core library
- For ProtoDUNE II, the full DTS prototype (hardware, firmware, and software) will be used



DTS reference hardware

- For small scale readout electronics integration, DTS group provides reference hardware FMCs
  - Mounts onto an FPGA board (Enclustra, Nexys Video)
  - Acts as master or endpoint depending on loaded firmware
  - GPIO
  - DTS IO via SFP
  - SI5395 PLL
    - Stand-alone or external clock input



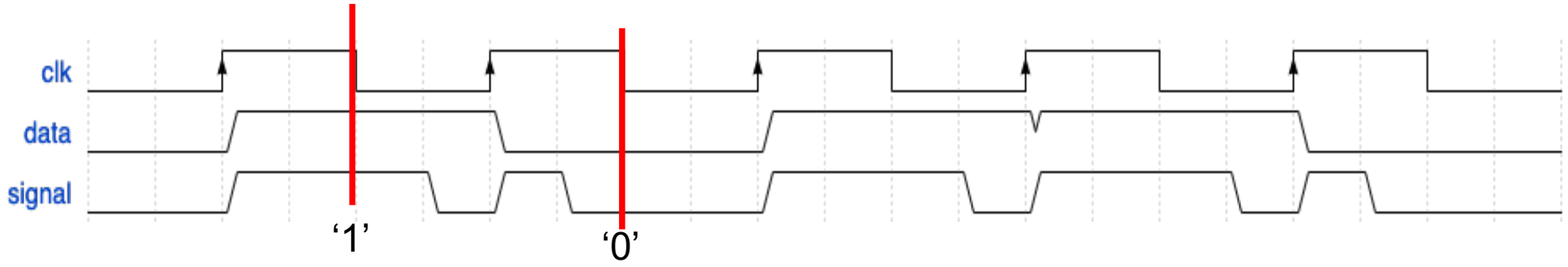
Conclusion

- Core DTS functionality has been extensively tested with DTS prototypes at ProtoDUNE-I, and other test-stands including Fermilab and Argonne
- ProtoDUNE-II will run with full chain DTS prototype (GPS→GIB→MIB→FIB)
- First production hardware procurement later in the year
  - For commercial components with long-lead time and for ICs at risk of shortage
- Firmware and software in advanced stages of development
  - By end of 2024, aim to have
    - Final design and implementation
    - Fully tested and documented

Backup

DTS data encoding

- Transfer data by modulating duty cycle of a “clock” signal
  - **Duty Cycle Shift Keying**
- Using 25% = 0 , 75% = 1 , 50% = “Z”



- Use 8b10b encoding for DC balance
- No external CDR chip required
  - Clock recovered from data inside FPGA