



Science and
Technology
Facilities Council



University of
BRISTOL



TPG - Software Status

Joel Greer, Kunal Kotheekar, Ivana Hristova

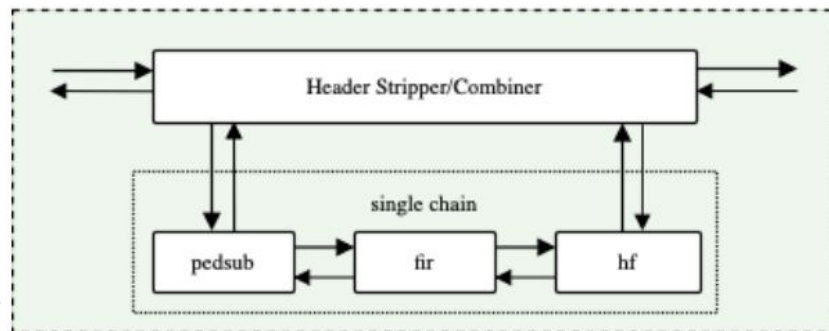
8 October 2020
Upstream DAQ Firmware

- Status of software validation tools
 - [dataflow-software](#) - repository of various modules and tools
 - [tpgSim](#) - Python simulation of TPG algorithm
 - Format conversion
 - For ADCs
 - For Hits
 - Data pattern generation
- Update since July's technology review
 - [tpgSim validation](#)
 - [tpgSim changelog](#)
 - [Binary capture data from ProtoDUNE-I \(C++ tools + tpgSim\)](#)
- Outlook and next steps
 - [dataflow-software](#) developments and future plans
 - Thoughts on long-term validation and online monitoring

tpgSim - TPG Python Simulation Status

- tpgSim implements TPG core firmware algorithm
 - Pedestal subtraction (PEDSUB)
 - 32-tap FIR filter (FIR)
 - Hit finder (HF)
 - Save State Restore (SSR) capability to improve HF
- tpgSim validation against LArSoft completed - 100% match except for a few corner cases
 - [Tag - v2.0.0](#)
 - [Documentation - Issue #57](#)
- Further tpgSim development currently driven by
 - Validation of TPG core firmware
 - Analysis of capture data from ProtoDUNE-I

TPG core



- Ongoing
 - Match firmware behaviour
 - Add 'hooks' to ease HF firmware validation
 - Fix remaining bugs if any
 - Clean obsolete and stale branches

tpgSim - TPG Python Simulation

Short-Term Plans

What is the goal of tpgSim?

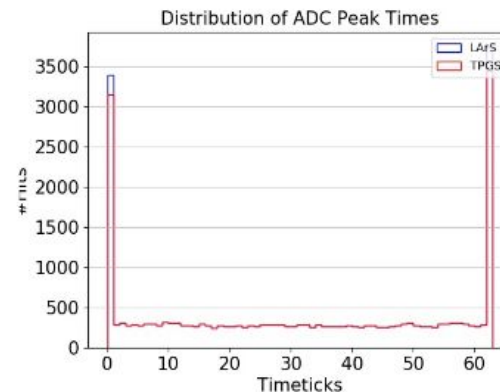
- tpgSim as part of dataflow-software will be supported and developed in parallel to the TPG firmware evolution
 - Run TPG testbenches - input/output text files, bitwise comparison between FW and SW
 - Setup CI tests - automatic validation of any TPG FW changes
 - As part of the (ProtoDUNE-I/II) capture data validation workflow
- What do we plan to do next? We propose the following
 - Run existing (or add new) testbenches which break down the TPG core functionality into standard testable blocks
 - Prepare scripts to run and validate each block
 - Iterate until we reach 100% match between FW and SW
 - Add scripts and testbenches to CI
 - Who - Joel, Ivana (SW) + Kostas, Kunal, Tony (FW) (?)
 - When - starting now (?)
 - Caveats - ongoing [TPG FW review](#) as reported by Kostas (is the FW still a moving target?)

tpgSim - Update Since ~July

Validation against LArSoft (Kunal)

[Detailed report](#) presented by Kunal at Upstream DAQ meeting

- Scripts
 - `test_fmt_larsoft.py` added to `dataflow-software` repository
- Data patterns
 - Pattern files specifically generated for TPG debugging during LArSoft-tpgSim comparisons
 - Long complex pattern generated to cover most hit distributions as expected in real data
 - Newly generated patterns stored in [dataflow-patterns](#)



Pattern: UniqueHits_F

LarSoft Hits = 24339

TPG Hits = 23791

tpgSim - Update Since ~July

Changelog (Ivana)

tpgSim was modified in order to match either LArSoft or TPG firmware outputs

- LArSoft comparison
 - Replace SciPy's `signal.lfilter()` with LArSoft's C++ implementation of the 32-tap FIR filter
 - Output axi4s files containing ADCs either after PEDSUB or after PEDSUB+FIR
- Firmware comparison
 - Append 6-word packet with pedestal information to each axi4s packet which has a hit
 - Append end-of-packet (EOP) word when hit(s) fully contained within an axi4s packet
 - 'hitContinues' is LSB in 16-bit hit packet data word
- Motivated by binary capture data validation
 - Initialise pedestal to either the 1st ADC value or the 1st ADC value of the 64-tick packet in which the 1st hit was found
 - ...

Tools to Interact with ProtoDUNE-I Binary Capture Data (Joel)

Tools have been developed to:

- Interpret the hit and ADC binary files, including those with pedestal words
 - Including extraction of ADC data from only the region of time where hits and ADCs overlap - for use in validation
 - Convert to standard dataflow-software formats (debugging remaining case)
- Extract 'impossible' hits:
 - Identify bugs in hits from the TPG firmware used at protoDUNE-I
 - Provide summary information which allows these hits to be located in hit capture
- Plot distributions of hit quantities
 - Including comparisons between matched Python emulated hits and the counterparts found in firmware (debugging ongoing)
 - Create input files for code which plots event displays (needs extension to multiple fibres)
 - Link to [presentation](#)

Summary and Outlook

- This meeting
 - tpgSim is ready to be used for the TPG FW validation and CI
 - tpgSim also part of the ongoing binary capture data from protoDUNE-I workflow
 - Both likely to require and result in further development - tpgSim not final yet
- Next meeting
 - More detailed view of the full set of dataflow-software modules and tools
 - Status and plans for their future direction and development
- Longer-term view
 - tpgSim useful for validating low-level implementation of TPG algorithm(s)
 - Analysis of binary data captures useful for validating full data processing and readout chain
 - Validation of bulk data during stable operation will require use of DUNE software framework and online monitoring tools