



Science and
Technology
Facilities Council



University of
BRISTOL



dataflow-software review

Joel Greer, Kunal Kotheekar, Ivana Hristova

15 October 2020
Upstream DAQ Firmware

- Status of software validation tools
 - [dataflow-software](#) - repository of various modules and tools
 - tpgSim - Python simulation of TPG algorithm (previous meeting)
 - Format conversion
 - For ADCs
 - For Hits
 - Data pattern generation
 - FW+HW framework - not covered today
 - Binary data capture analysis - standalone tools + dataflow-sw usage at final stages
- Review
 - Recent timeline
 - Code status
 - Issues
- Summary and discussion

dataformats: re-organised script structure according to what discussed with Jim and Kunal

Initial CI

Covid-19 lockdown

ADCs / Hits formatters + tpgSim rapid development, response to FW requests

14 Nov 2019

3 Feb 2020

March 2020

April 2020

May 2020

Initial tag for tests at NP04
June 2020

Ivana start Maintainer

Major refactoring and renaming

Stop major refactoring

Final tag for tests at NP04

FW Technology Review

1 June 2020

3 June 2020

7 June 2020

26 June 2020

29 July 2020

30 July 2020

protoDUNE-I data taking

tpgSim-LArSoft validation done

tpgSim fixes + binary capture data results

dataflow-software review (today)

dataflow-software Recent Timeline

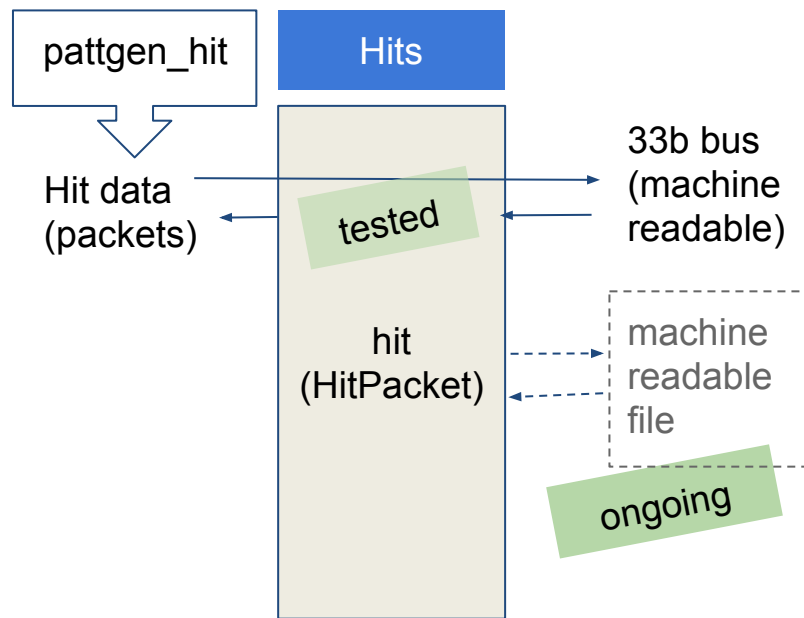
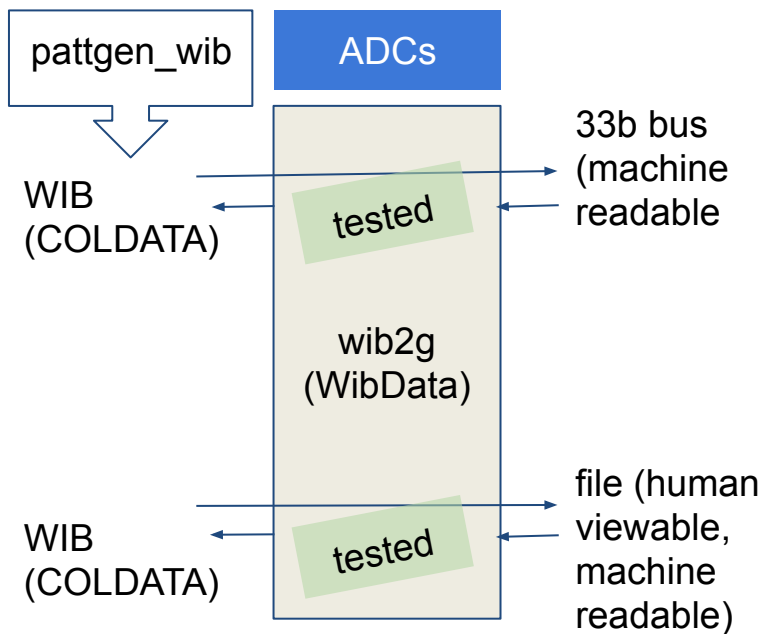
Aug 2020

Sep 2020

Oct 2020

framework	dataformats	bin	test	dataflow-software
<p>Tools for FW/HW control, configuration, testing, debugging</p>	<p>Data pattern generation</p> <p>Format conversion (for ADCs / Hits)</p> <p>FW validation</p>	<p>Commands (CLI), executables</p>	<p>(Unit) tests</p> <p>CI</p>	<p>archive data etc formatters → some legacy code + scripts</p> <p>env.sh env_sim.sh genTPGInPatts.sh test_fmt_larsoft.py</p>
<p>config.py cr.py dpr.py dr.py setup.py sink.py termui.py texttable.py toolbox.py wibulator.py</p>	<p>axi4s.py bus33b.py channel_map.py fibrepattern.py hit.py wib2g.py fmt_hit_33b.py fmt_hit_axi4s.py fmt_hit_file.py fmt_philfile.py fmt_tpg.py fmt_unpacker.py fmt_wib_33b.py fmt_wib_dpr.py fmt_wib_file.py pattgen_hit.py pattgen_wib.py wibdump_to_patternfiles.py</p>	<p>enableMGTplayback.py hfTester.py hfWatcher.py hfButler.py hfDriver.py hfTPConfig.py hfWibulator.py peekSinks.py</p> <p>adcBinaryMangler.py hfDataflow.py hitBinaryMangler.py pattGenHit.py pattGenWib.py rawCrawlerHitBinaryCheck.py tpgSim.py adcFormatter.py hitFormatter.py pattGenPhil.py tpg_investigation.py</p>	<p>dtpc_dpr_test.py dtpc_readDPR.py dtpc_writeDPR.py playback.py readWIB.py test_config.py test.py writeConfigPacket.py</p> <p>circularTest.py test_33b.py test_axi4s.py test_channel_map.py (CI) test_diff_philgen_wibgen.py test_fmt_hit_33b.py (CI) test_fmt_hit_axi4s.py test_fmt_hit_file.py test_fmt_philfile.py test_fmt_unpacker.py test_fmt_wib_33b.py (CI) test_fmt_wib_dpr.py test_fmt_wib_file.py (CI) test_tpg.py (CI)</p>	<p>18k Python 1k XML 150 bash 225 files</p> <p>data classes ADCs / Hits 'central' class</p> <p>formatter classes</p> <p>old naming</p> <p>standalone</p>

Review of ADCs/Hits Formatters



philfile
unpacker
dpr

not tested recently
(old naming)

to be fixed

difference

not tested recently (old naming)

axi4s

file (human
viewable)

List of (Open) Issues

Consolidate Hit File Format	Ivana	Ongoing
Impossible values in protodune hit binary captures	Joel, Kunal	Ongoing
TPG LArSoft Validation (Corner Cases and Documentation)	done	Documentation
Clean up channel_map	to be fixed	Pointed out at the previous meeting
scripts	Ivana	Ongoing
Allow disabling of individual parts of TPG chain	done	To be closed
Pattern Requirements	done	To be confirmed
Comprehensive pattern generation	done	To be confirmed
Handle state saving in tpgSim	done	To be closed
Implement unit tests for formatters in CI	Ivana, ...	To be tested and fixed
Fix unpacker to wib object conversion	Ivana, ...	To be tested and fixed
Finish fmt_dpr	Ivana, ...	To be tested and fixed

- Most open issues related to data formatters
 - Keep opening new issues to track ongoing and future tasks

- dataflow-software repository
 - Keep 'FW+HW' and 'Simulation' code in separate `bin / test` (sub-)directories similar to having separate 'framework' and 'dataformats' directories (?)
 - Create 'scripts' directory to store bash files and other executables ([#55](#))
 - **Fix unit tests (broken due to module renaming)**
 - Set complete set of tests in CI ([#25](#))
- Updates for protoDUNE-II
 - Channel mapping (suggested at previous meeting)
 - Understand which TPG formats to support in the simulation (raised at previous meeting)
 - WIB, DPR formats (?)
 - ...
- Current and next step(s)
 - Save and read HitPacket-s to and from a file
 - Finish tasks related to protoDUNE-I - testbenches, format conversion

- Testbenches
 - Prepare list of TPG testbenches - input/output files, scripts
 - Kunal
- Format conversion
 - Make sure all data formatters are up-to-date and running - unit tests, CI
 - Ivana
- Binary capture data analysis with protoDUNE-I
 - Finalise analysis and store code in a separate branch/repository
 - Joel
- Monitoring tools
 - Integration of existing and new tools in DUNE's app fwk
- Documentation
 - Data formats - live document maintained by software and firmware developers
 - Actual file formats produced by the dataflow-software conversion tools
- Resources
 - Prepare list of tasks and people needed for dataflow-software development in the next months