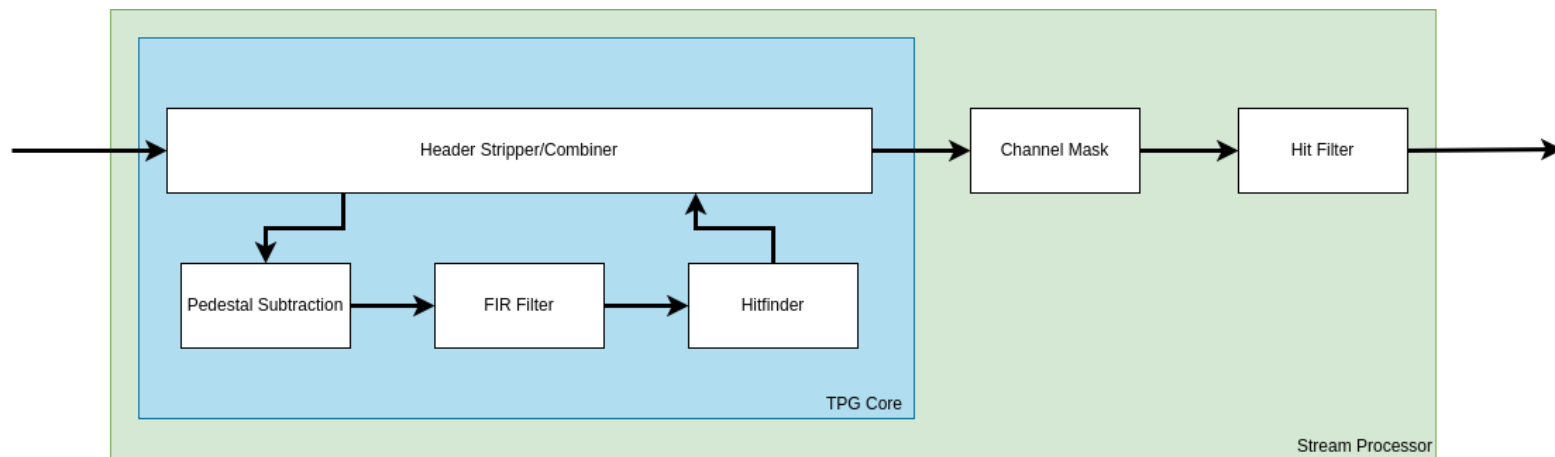
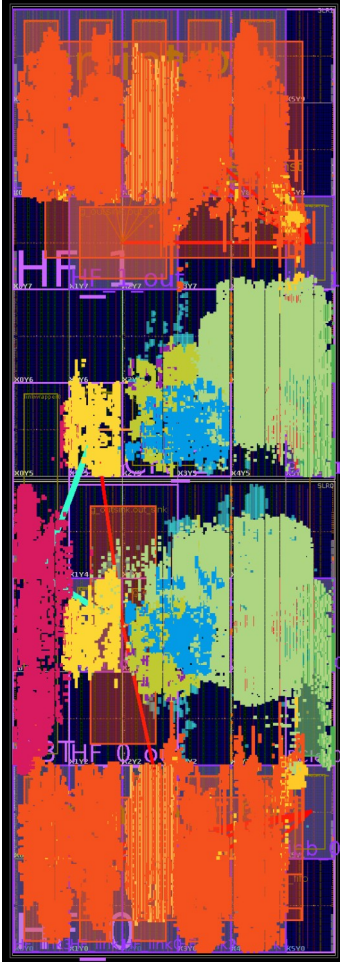


Firmware Trigger Primitives Lessons Learned

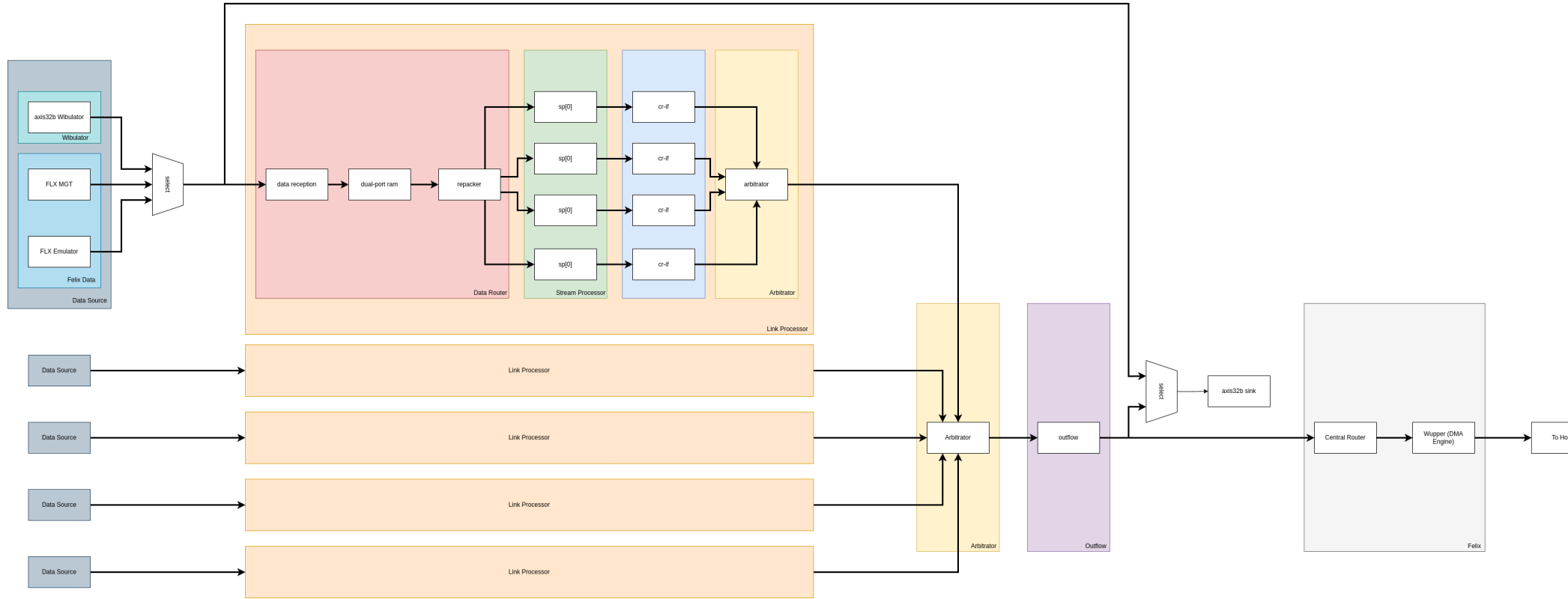
Antony Earle
DUNE UK
12/01/2023

- Firmware Based Trigger Primitive Generation, (TPG), produces trigger primitives from streamed ADC samples when ADC values on a wire cross a specified threshold value



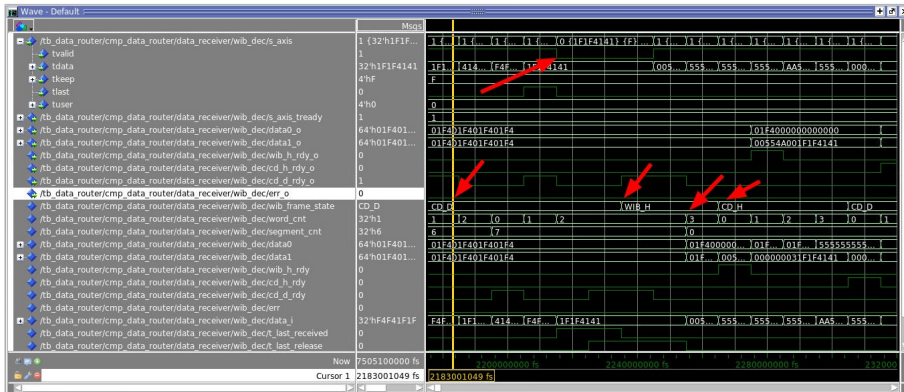


- FW TPG retrofitted into the top level design for the BNL712 variant of the Atlas Front End Link eXchange card, (FELIX)
- FELIX design is split into two endpoints with each half handling 5 ADC optical links

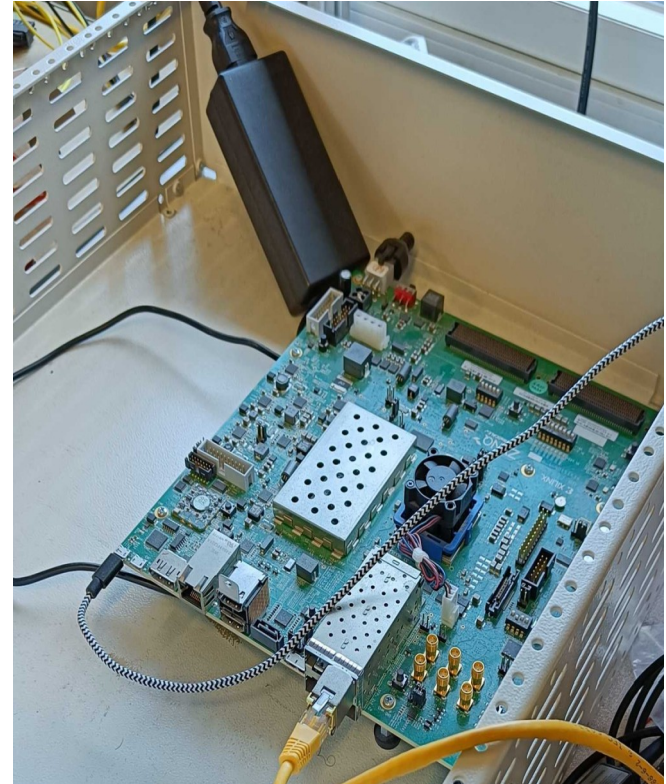


Trials and Tribulations

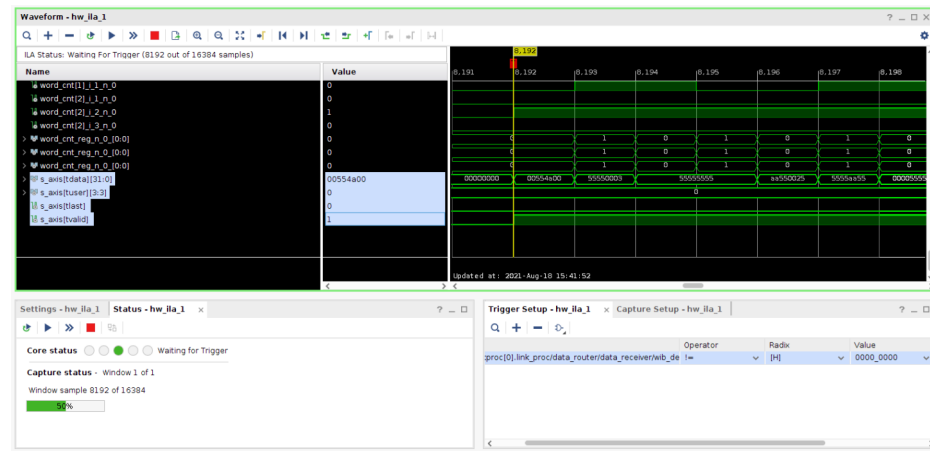
- Firmware testbench simulations used for testing individual firmware blocks and simulating the entire link processing chain
 - Used extensively in testing the hitfinding core in conjunction with a suite of test patterns
 - Successfully used to recreate some error states seen in FELIX running with greater insight into the issue
 - Individual testbench blocks very useful as part of the continuous integration strategy as it helped to spot changes that would break functionality



- Hardware test platform used ZCU102 development board
 - Had FW TPG blocks without the overhead of the FELIX infrastructure
 - Build time much shorter than FELIX, allows quick iterations for debugging
- Used to replicate issues seen in FELIX hardware
 - Channel masking issue was replicated on ZCU102 hardware and patch fix tested before deployment on FELIX



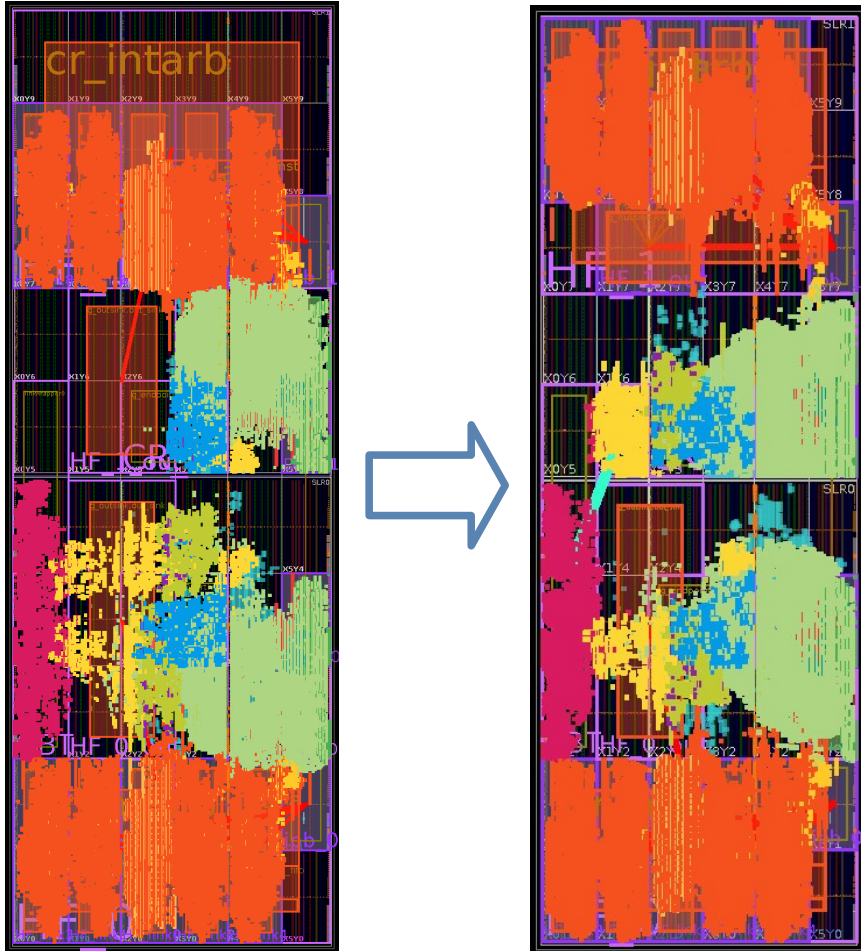
- Coverage between sim and ZCU102 test solutions helped with spotting issues
 - Example issue with data reception state machine not replicated in simulation but spotted in ZCU102 testing



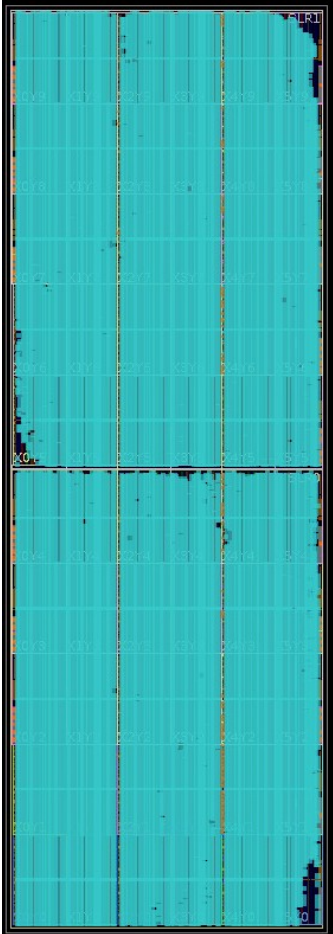
- Test tools became limited in reproducing bugs seen in hardware when encountering real detector data
 - Firmware blocks used as test sources, (Wibulator), did not behave in the same way as FELIX Emulator or external data from the decoder block
 - Specific example being the control of axis flags which revealed a bug in the data reception block. Bug was not caught in testing as the Wibulator test source data was not replicating tvalid drops in the same way as real data caused inside the FELIX
- Test tools did not keep up with issues as we encountered them
 - Test source should either generate data in the same fashion as will be seen in the hardware implementation or test patterns should be adjusted to do so

- Relied more on using in-production FELIX cards as test benches for trialling firmware fixes
 - Can be tricky to organise as it occupies real detector hardware which is needed by others
 - Limited debugging capability with lack of JTAG functions for quick reprogramming and data spying
- Would suggest dedicated test bench hardware with fully representative data source if this was attempted again

Felix integration

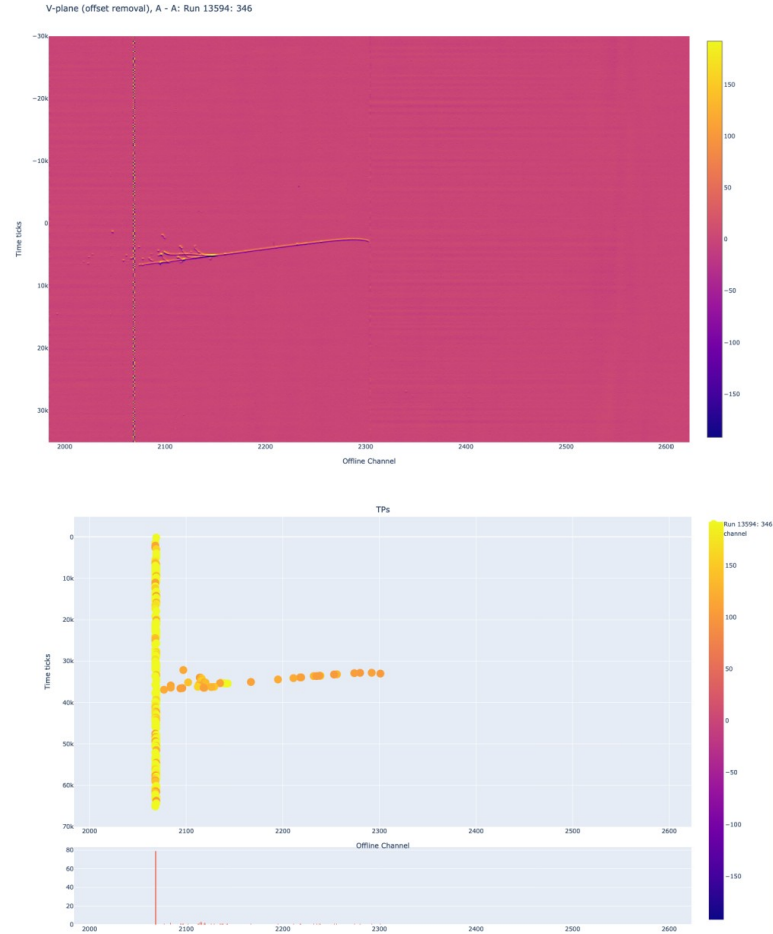


- Rebased FELIX FW to incorporate features provided by FELIX developers
 - Rebasing a bit of a lottery in terms of time required to implement
 - Changes could require updates to build scripting to make compatible
 - Large functional changes could require repeat build testing with adjustments to firmware block placement to ensure timing restrictions were achieved. Costly due to long build times of FW, (2-3 hours on a fast machine)



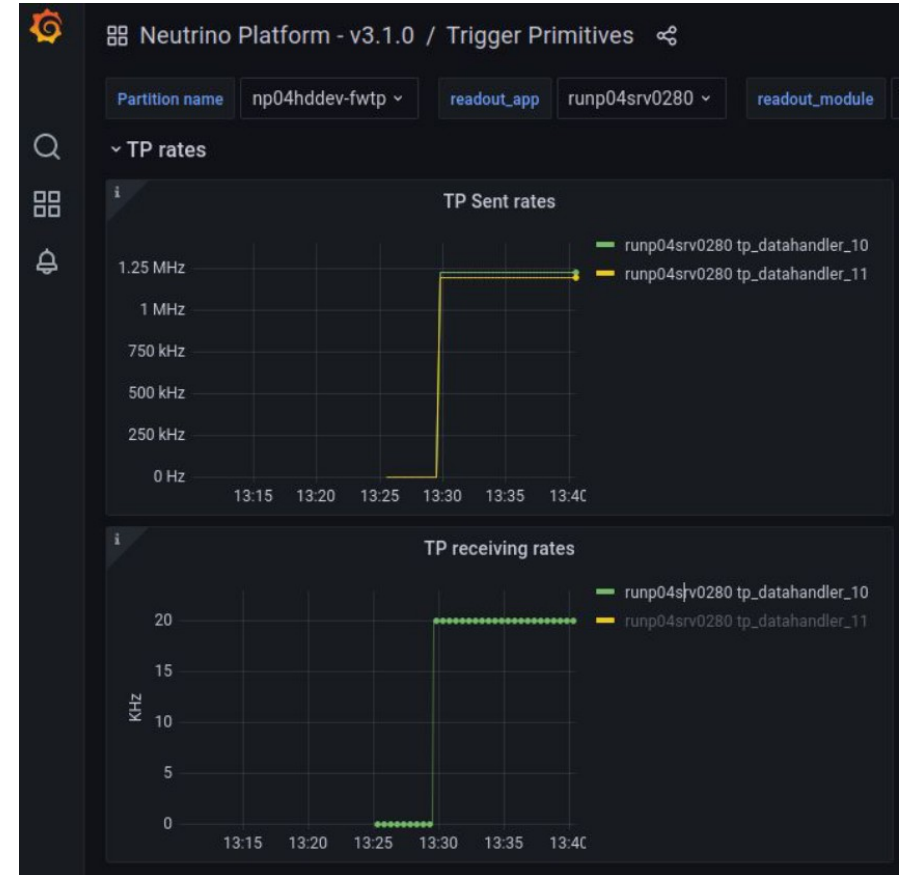
- Patch fixes for DUNE specific implementation were necessary to overcome some changes made by upstream developers
 - Would be beneficial if upstream developers could be given our use case to run as part of their continuous integration tests
 - Not necessary for it to pass or for them to roll back changes if the build fails, but can be used as an early warning system when looking to rebase firmware

- First operation of phase-II FELIX FW TPG chain and readout software with real detector hardware, (see Shyam Bhuller's talk for details)
- Managed trigger records with generated hits evidenced on event display



- Issues with stability of system
 - System halts due to data back pressure were common, especially at processing start
- Processor configuration required arcane start up procedure reliant on command line scripting
- Difficulty in quickly analysing run data to check for issues
 - Difficult to discern if run data was good or not, or if the system was misconfigured in some way

- Developed multiple firmware builds to support DAQ testing during the week
 - Having all developers on one hallway enabled quick turn around and discussion of issues
- TPG configuration streamlined and used as part of run control
 - Double edged sword as over-reliance on using run control config constrained testing in some aspects
- Monitoring and FW TP metrics added to Grafana dashboards for better feedback and monitoring



- Blocking of link processing due to back pressure was a common issue during data runs
- Mitigation strategies:
 - Implementation of a pedestal capture mechanism which sampled the first ADC value for each wire to be processed and assigned the same value to that wires median.
 - Increased bandwidth of processing chain after stream processor blocks by increasing bit width form 16b to 32b

pipelines

probe	0	1	2	3
p0: upck >> hsc	4040 [bsy] (l) 0	4039 [bsy] (l) 0	4040 [bsy] (l) 255	4040 [bsy] (l) 0
p1: hsc >> psub	4040 [bsy] (l) 255	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4040 [bsy] (l) 0
p2: psub >> fir	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4040 [bsy] (l) 0
p3: fir >> hf	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4040 [bsy] (l) 0
p4: hf >> hsc	4039 [bsy] () 0	4039 [bsy] () 0	4039 [bsy] () 0	4040 [bsy] () 0
p5: hsc >> mask	4039 [bsy] (ul) 0	4039 [bsy] (ul) 0	4039 [bsy] (ul) 0	4040 [bsy] (ul) 0
p6: mask >> filt	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4039 [bsy] (l) 0	4040 [bsy] (l) 0
p7: filt >> arb	1933 [bsy] (l) 0	1933 [bsy] (l) 0	1933 [bsy] (l) 0	1934 [bsy] (l) 0

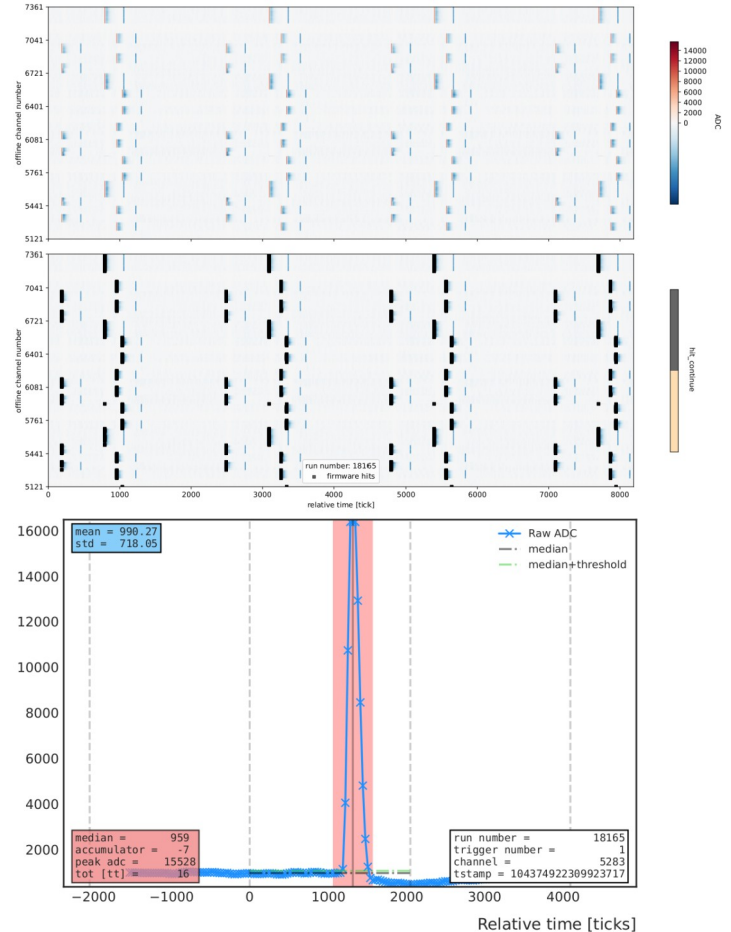
- Occasional back pressure blocking could lead to a degraded state where a processing pipeline does not recover after reset signal is issued
 - This renders the link unusable until a power cycle is issued to the card
 - Patch builds were developed and tested to fix the issue, but it is still present

<i>pipelines</i>				
probe	0	1	2	3
p0: upck >> hsc	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p1: hsc >> psub	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p2: psub >> fir	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p3: fir >> hf	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p4: hf >> hsc	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p5: hsc >> mask	1 [rdy] () 255	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p6: mask >> filt	1 [rdy] () 255	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p7: filt >> arb	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0

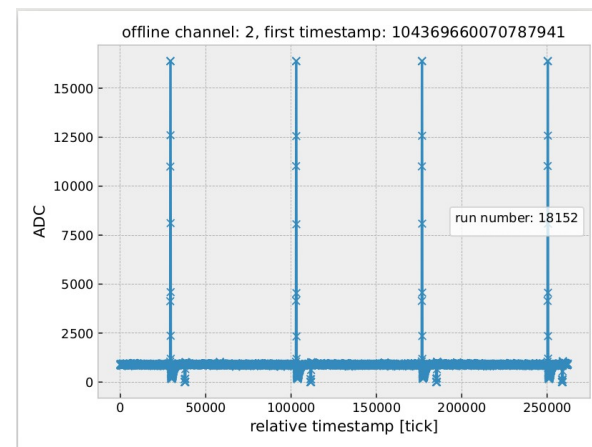
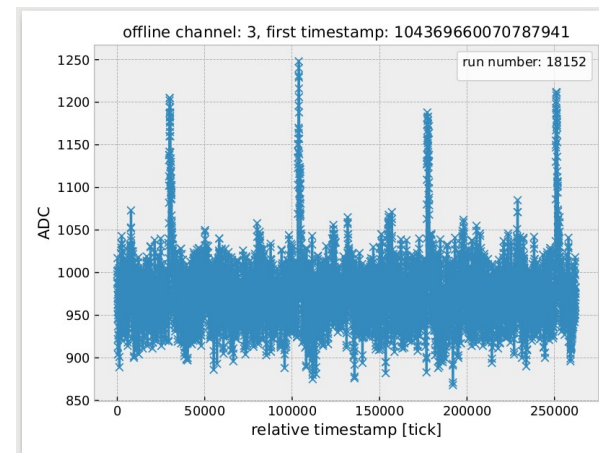
- Found that the current level of monitoring in FW is insufficient to support debugging of problems like this when encountered
- Test solutions, (simulation/ZCU102) failed in replicating the bug, leaving no option for isolating the issue

<i>pipelines</i>				
probe	0	1	2	3
p0: upck >> hsc	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p1: hsc >> psub	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p2: psub >> fir	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p3: fir >> hf	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p4: hf >> hsc	0 [bsy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p5: hsc >> mask	1 [rdy] () 255	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p6: mask >> filt	1 [rdy] () 255	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0
p7: filt >> arb	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0	0 [rdy] () 0

- New build developed after DAQ integration week which improved 32b converter block with monitoring stats and an additional fifo
- Testing showed improvement in stability with multiple runs taken without link blocking error and event display evidencing TPs correctly



- Run config very streamlined by this point
- Good example of overall improvements in data feedback tools as these were used to confirm the misconfiguration of WIB calibration pulses and helped to remove some doubt around firmware functionality



- SW and HW test benches were useful this year right up until they weren't - Test solutions did not keep up with the times so missed a few edge cases. A dedicated test bench that replicated WIB pulser setup/not relying on testing with an APA/NP04 server
- Felix integration could get painful when rebasing upstream developers changes. Try to get them to include your test case as an early warning system when merge testing
- The monitoring infrastructure built into the FW didn't contain enough information so debugging backpressure and locking were problematic to diagnose and couldn't be replicated outside of NP04
- Data validation was invaluable as quick verification things worked, or that there were config issues elsewhere in the detector, (again another problem with testing firmware fixes in an APA card rather than a static testbench)
- This talk highlights difficulties last year but entire FW TPG team should be proud of the work done and improvements made