

Operating Strips Modules with ITSDAQ and YARR

Ryan Roberts, Timon Heim, Bruce
Gallop



Comparing ITSDAQ and YARR

We have two different DAQ systems in place for ITk strips modules

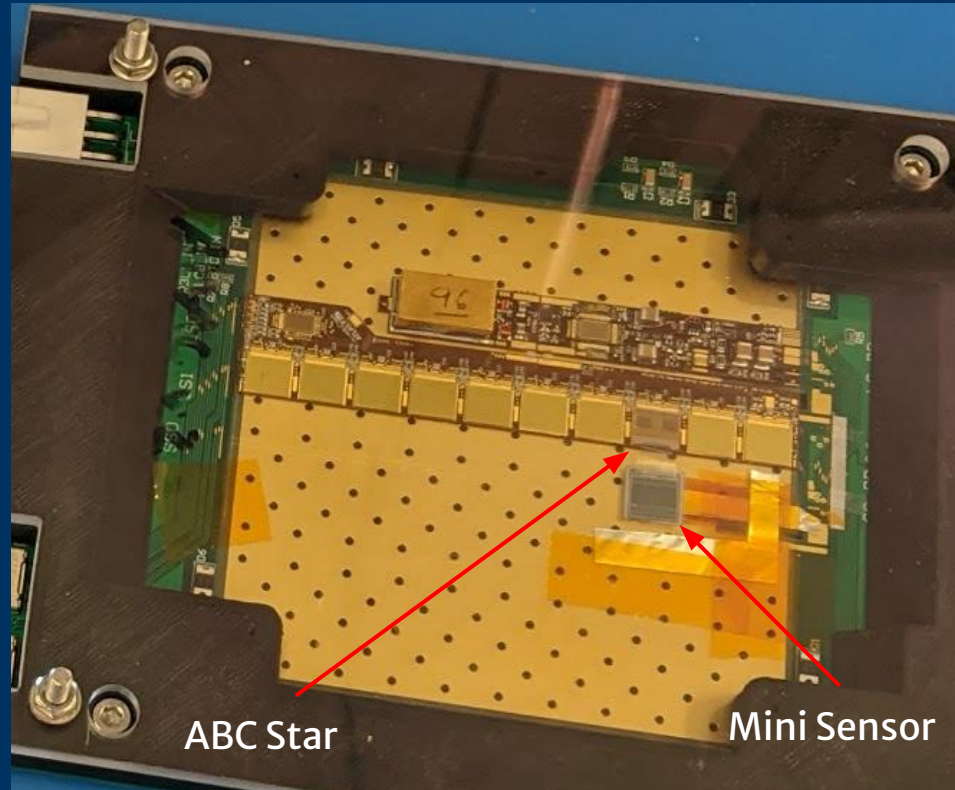
1. ITSDAQ, an iteration on SCTDAQ, has been used in the development and testing of ASICs and module construction. Based on ROOT. Operates with Nexys FPGA board as the readout hardware.
2. YARR, a more general DAQ software, will be used for both strips and pixels in ITk alongside the FELIX readout system, though YARR can also be used with different readout hardware.

We want to perform the QC scans in both systems and check for discrepancies.

Our Setup

We are using a DAQ load which consists of a power board and a hybrid which carries only a single ABCStar front end chip.

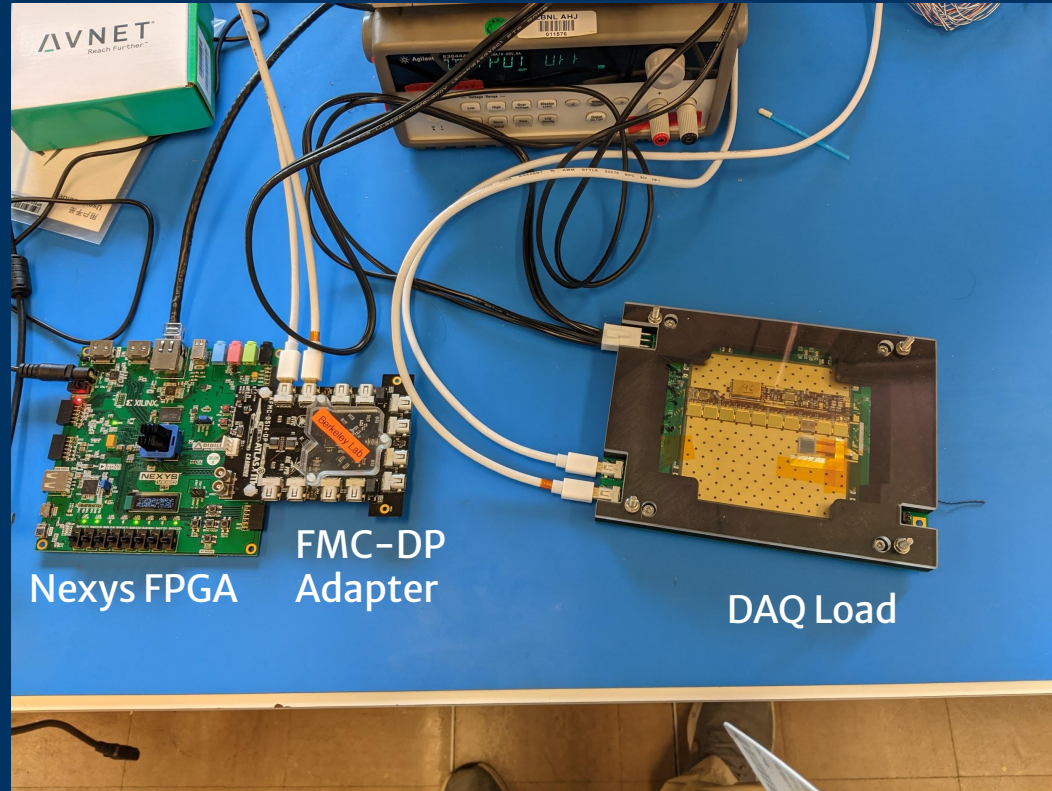
128 of the 256 channels on the ABCStar are connected to a “mini sensor” with ~2 cm strips



Our Setup

We are using a DAQ load which consists of a power board and a hybrid which carries only a single ABCStar front end chip.

128 of the 256 channels on the ABCStar are connected to a “mini sensor” with ~2 cm strips



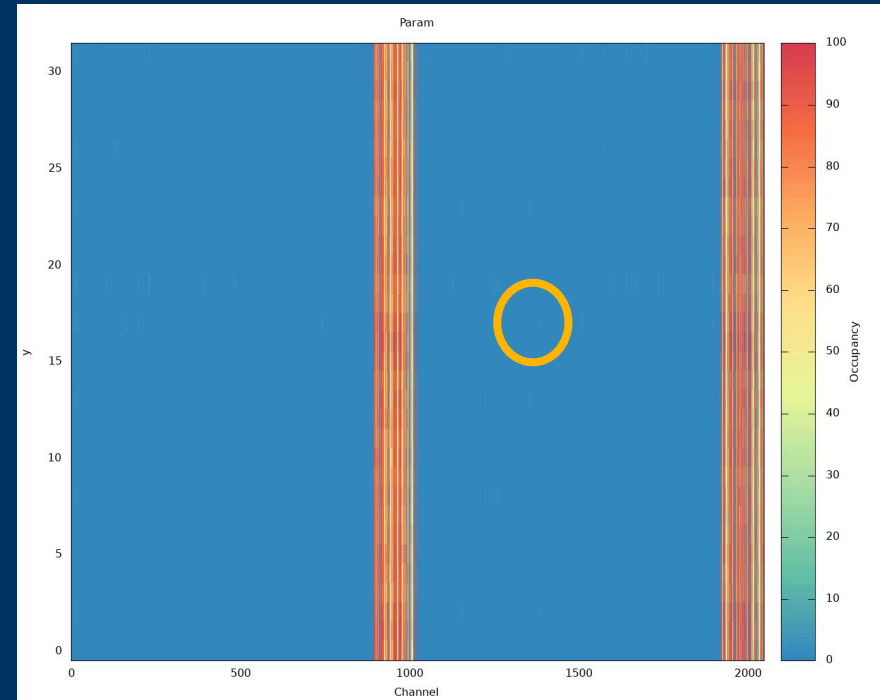
Efforts to Develop Strips QC in Yarr

Multiple people are working or have worked on this

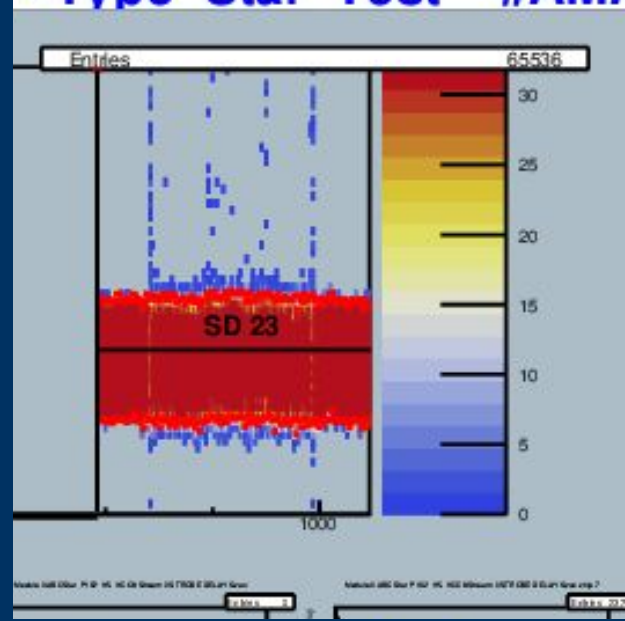
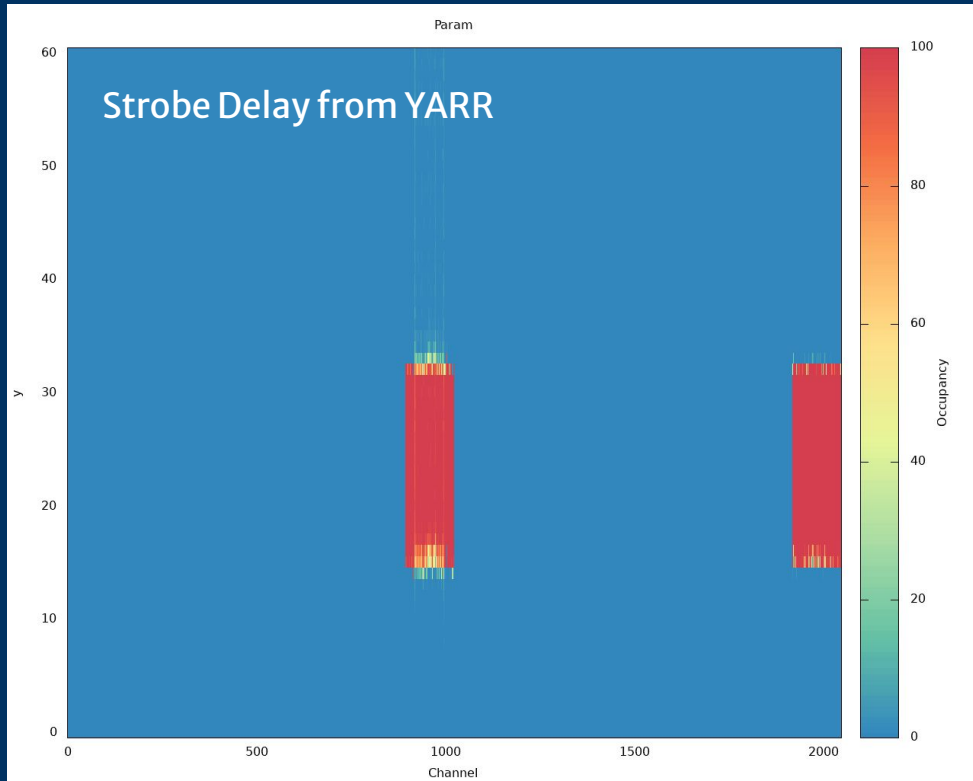
- Different setups and focuses
- J.J. worked on this last year with a FELIX/NetIO setup
 - Some scans implemented with external analysis via ROOT
 - Branch Name: devel_FelixNetIO_StarChip
- Zhengcheng working on this with FELIX/NetIO
 - Implementing chained analysis for n Point Gain scans
 - Branch Name: devel_chained_analysis
- I have been working with a Nexys FPGA setup
 - Mostly debugging the setup and ItsdaqFW
 - Have implemented register reads
 - Partially implemented hit counter data taking
 - Implemented 2D Histogram Analogous to Itsdaq

Selected Debugging

- Strobe delay scan showed no dependence on the strobe delay
- Same story with calibration pulse size; digital scans empty
 - Indicates no injection is happening
 - Discovered that order of trigger and pulse were reversed
- Packets that can't be parsed+hits outside of existing ABC
 - Solved by turning of HPR packets
 - ItsdaqFW communication implementation in YARR has trouble with packets close together



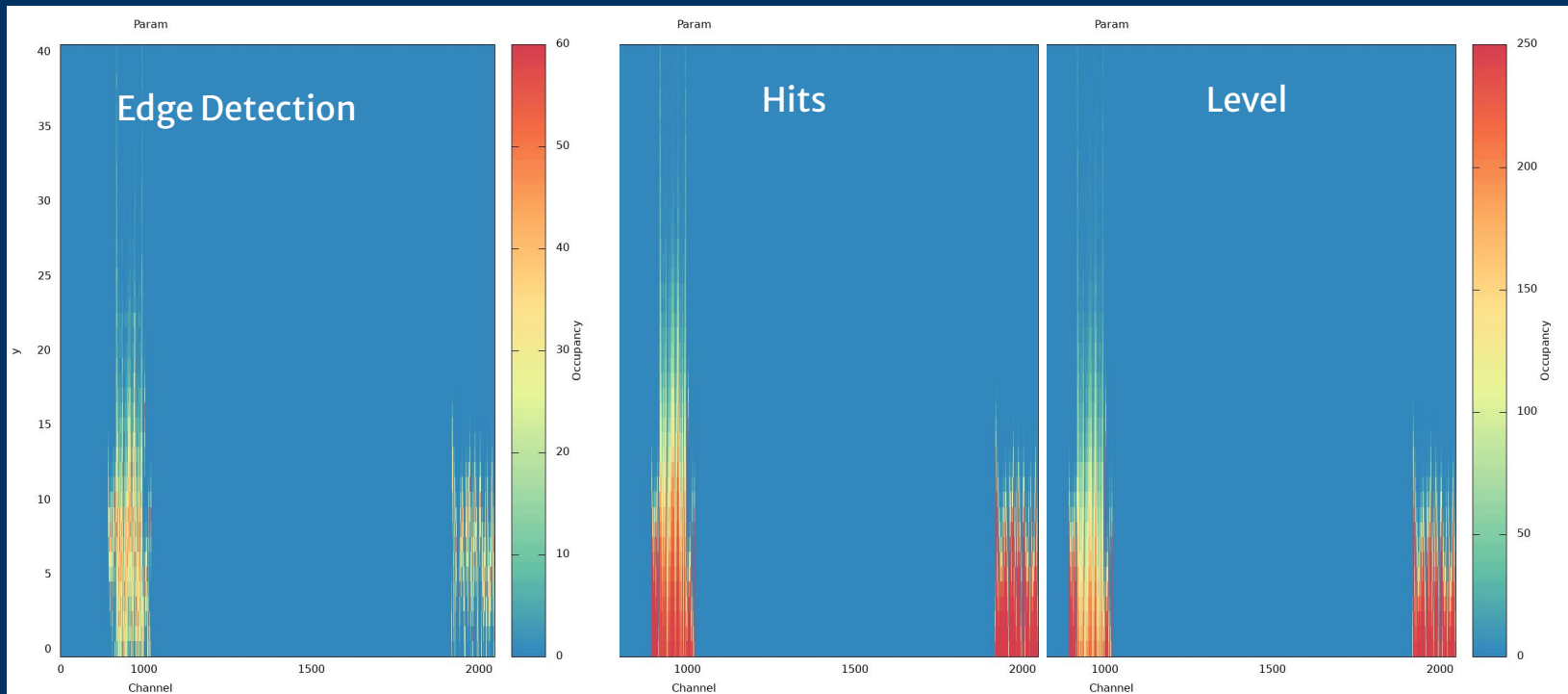
Strobe Delay in YARR



Strobe Delay from Itsdaq (stream 0 only)

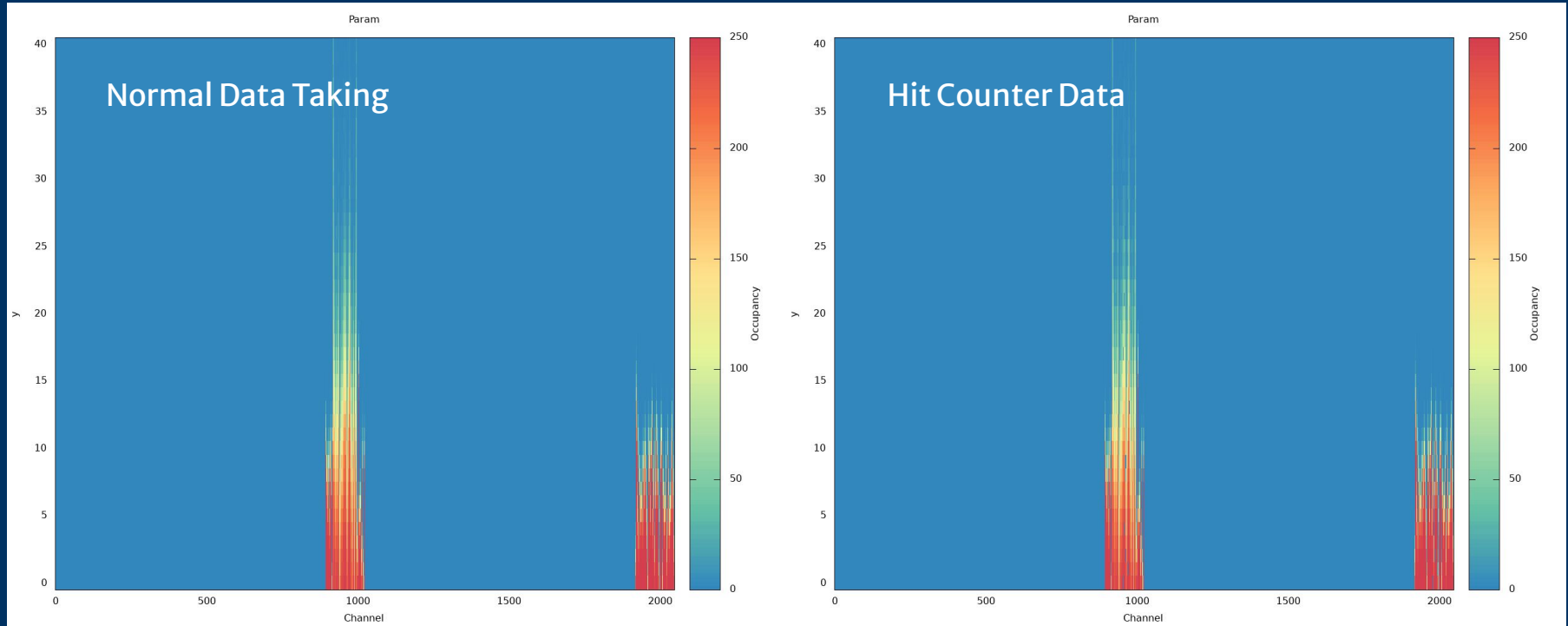
Analysis2D YARR Analysis gives plots similar to Itsdaq

Noise at Low Threshold



At some point, noise occupancy at low thresholds was not 1 – seems to depend on the hit detection mode

Hit Counters (NO Scan)

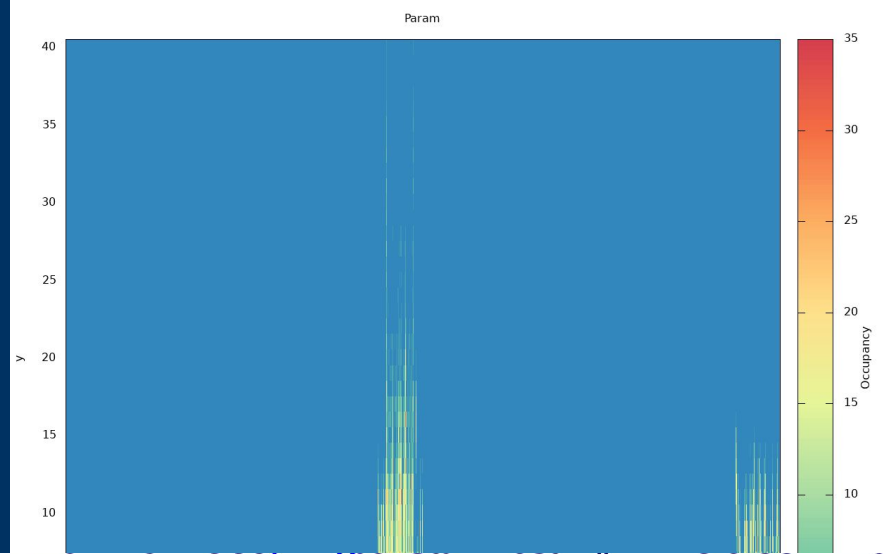


Able to take data from hit counters. (StarCounterLoop)
Should be faster for some scans: Less communication with
FPGA needed.

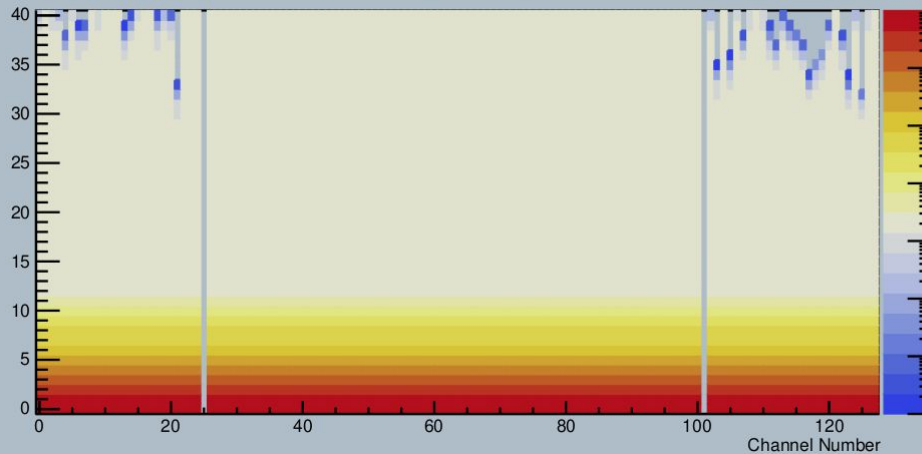
Status of Strips QC in YARR

QC Step*	YARR Status
Chip Communication ¹²	Communication established, need to write tests based on register reads and HPR's
Strobe Delay ¹²	Data taking available, J.J.'s branch has offline analysis
Trim ²	Config can take in the trim values and scan can be performed. Has anyone implemented this in YARR Feedback?
3 Point Gain ¹	J.J. has offline analysis and Zhengcheng is working on this with chained analysis.
Response Curve (10 pt gain) ²	Generalization of 3 Point Gain
Noise Occupancy ¹²	I am working on this - see next slide.
I-V Curve	Outside the scope of YARR?

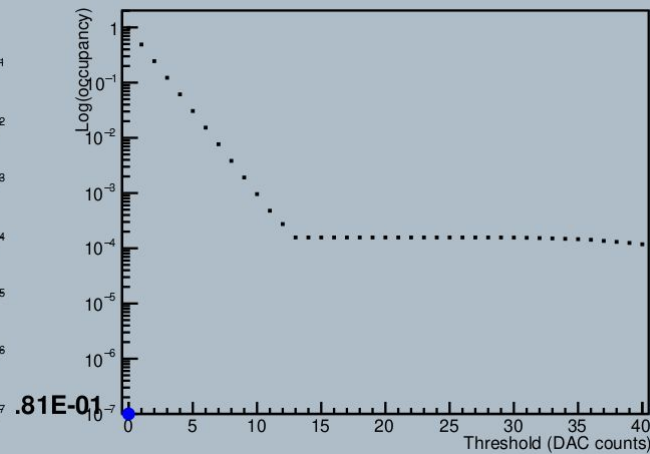
Noise Occupancy Needs



Evens Module 0 ABCStar_P162_H5_HCC8 Occupancy Data



Evens Module 0 ABCStar_P162_H5_HCC8 Occupancy Data



Closer Look at Noise Occupancy

- Want high stats in high threshold region
 - Need to use higher triggering rate (hopefully no problem)
 - Want to use hit counters
- High occupancy at low thresholds
 - Counters fill up quickly, need fewer triggers per burst
- Itsdaq handles this by actively varying the number of triggers per burst
 - Still considering how to implement in YARR

Status of Strips QC in YARR

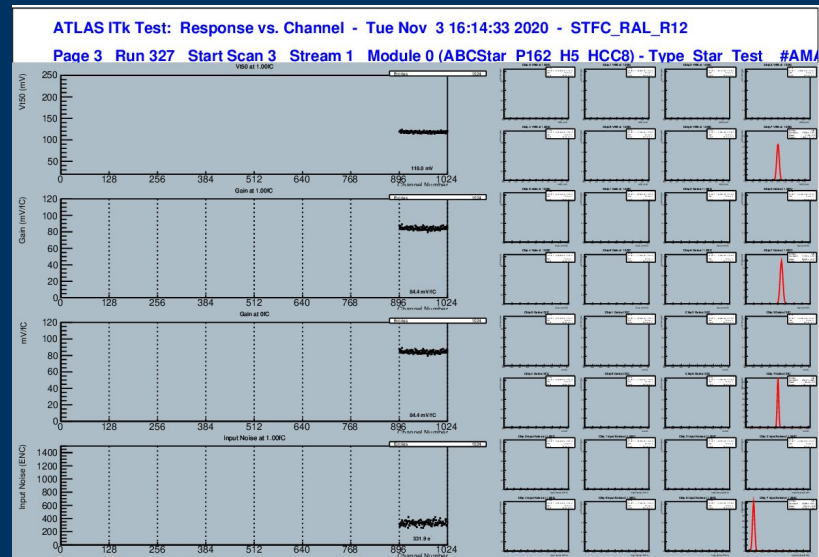
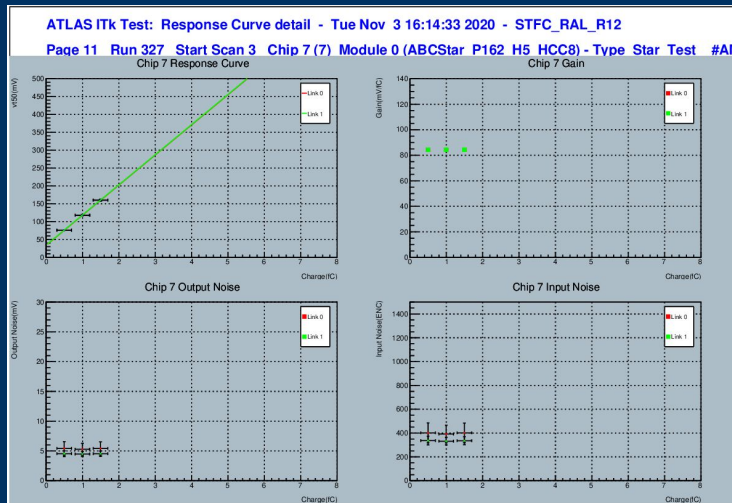
QC Step*	YARR Status
Chip Communication ¹²	Communication established, need to write tests based on register reads and HPR's
Strobe Delay ¹²	Data taking available, J.J.'s branch has offline analysis
Trim ²	Config can take in the trim values and scan can be performed. Has anyone implemented this in YARR Feedback?
3 Point Gain ¹	J.J. has offline analysis and Zhengcheng is working on this with chained analysis.
Response Curve (10 pt gain) ²	Generalization of 3 Point Gain
Noise Occupancy ¹²	I am working on this - more later
I-V Curve	Outside the scope of YARR?

Thanks!

Backup

Current Status with ITSDAQ

ITSDAQ is mostly working as expected now with our setup and we have fitted results which can be compared to YARR



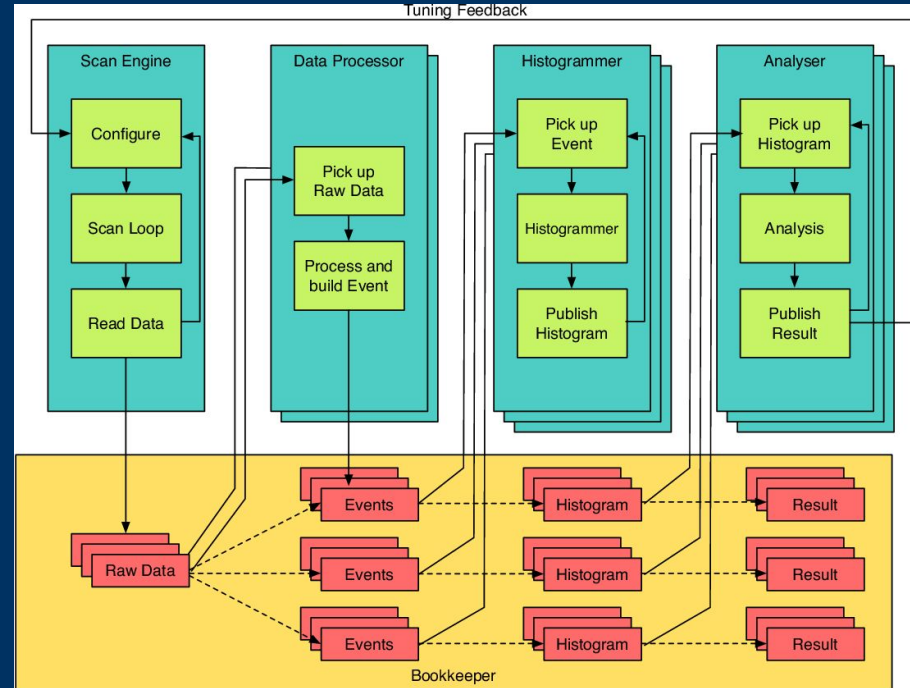
Results of a three point gain response curve measurement

Berkeley

First Steps with YARR

Additionally, the suite of scans used in the QC process has not been fully recreated in YARR, though YARR's scan configuration makes many of these data taking operations simple to implement.

More challenging is the implementation of the histogramming and fitting required to produce directly comparable results.

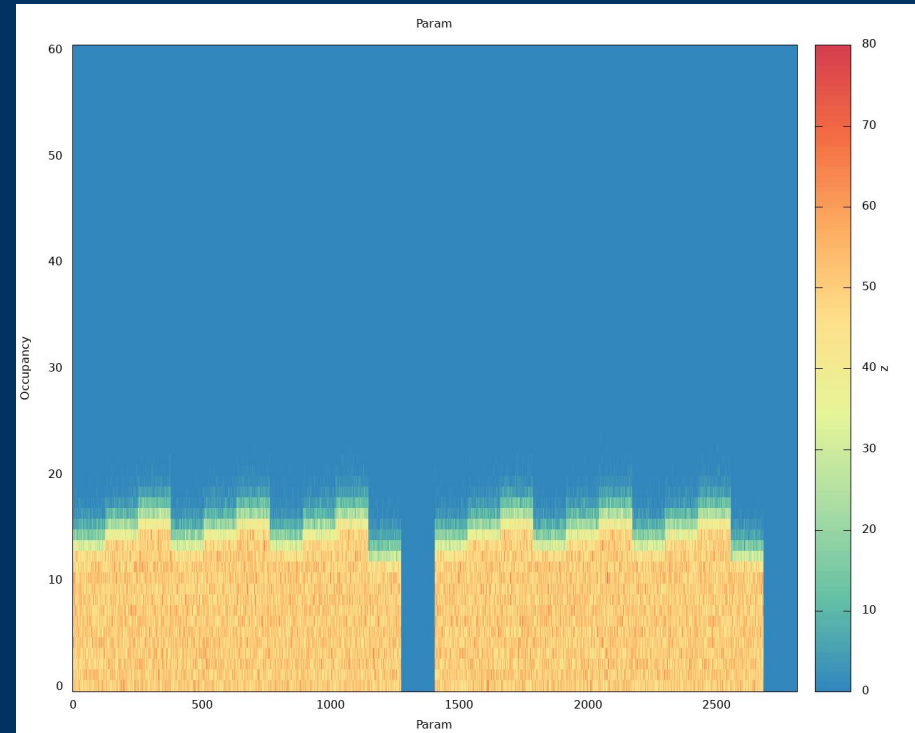


Towards Full Scans in YARR

The next step was to create a histogrammer which arranges data in a similar layout to ITSDAQ.

The channel number is on the horizontal axis and the vertical axis can be used for a scan variable, in this case the threshold.

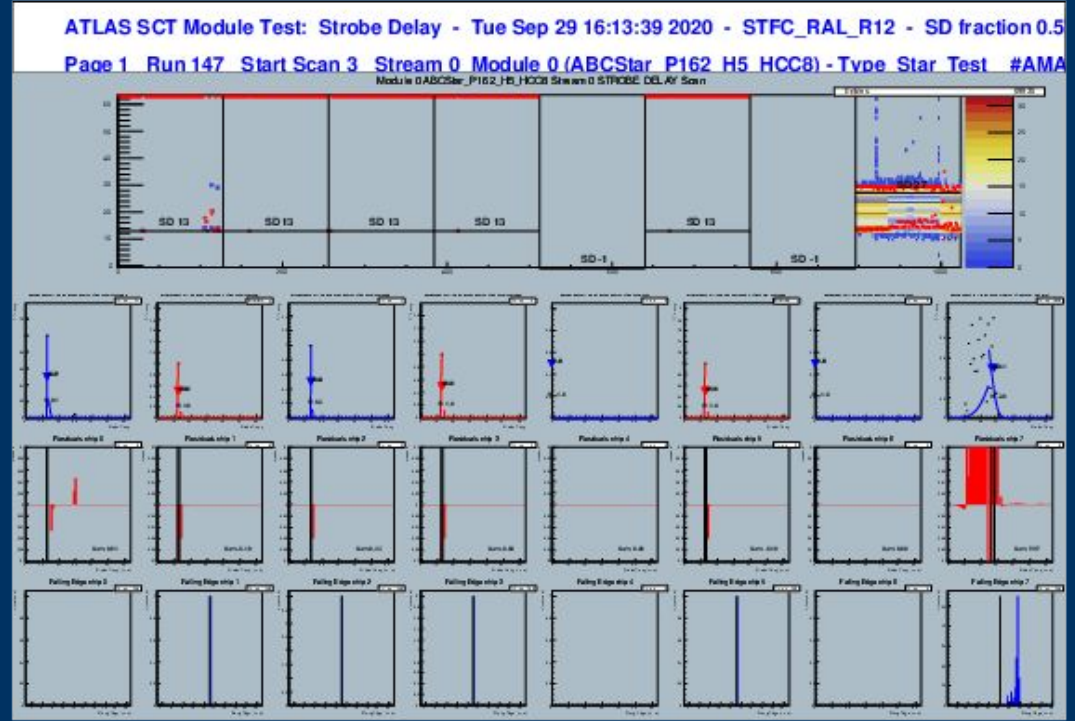
This scan is a simple threshold scan at a fixed charged injection, performed in emulation. The stair step pattern is due to the emulated ABC's having different settings for the threshold scale.



First Scans from ITSDAQ

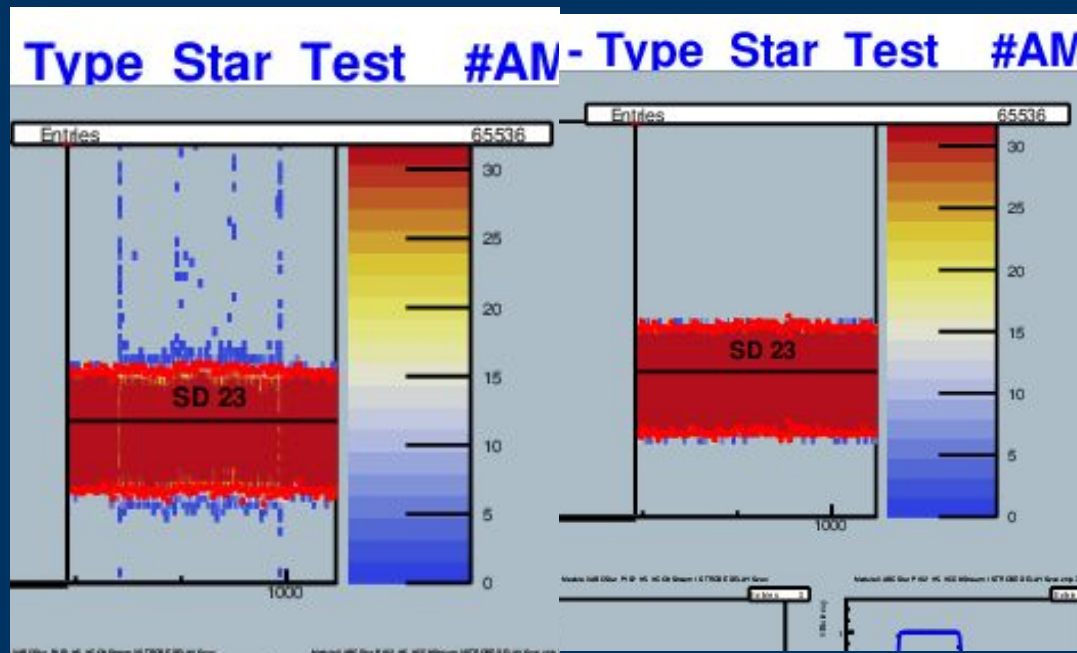
Initial scans showed problems with missing packets and data points showing up in channels that don't exist.

The problem was traced to the synchronization of the output stream and was fixed by setting the stream to be 8b10b encoded.



Next Steps with ITSDAQ

Fixing the encoding removes the missing data, but we still see large noise on the channels connected to the mini sensor because the sensor is not biased.



Itsdaq Crashing

Itsdaq has a habit of crashing after running a scan while writing data

Have tried updating Itsdaq, Nexys firmware, and OS

```
st_configure_variable: module -1 typ 1509 value 12.00
st_execute_variable: Send configuration for scan variable: ABCStar variable (8)
st_configure_variable: module -1 typ 3085 value 1.00
st_configure_variable: module -1 typ 3079 value 1.00
st_execute_variable: Send configuration for scan variable: HCCStar variable (79)
hsio_flush_event_data_buffer: throw away 3 items in opcode buffer
hsio_flush_event_data_buffer: throw away 3 items in opcode buffer
st_configure_variable: module -1 typ 1508 value 0.00
st_configure_variable: module -1 typ 3079 value 0.00
hsio_flush_event_data_buffer: throw away 2 items in opcode buffer
hsio_flush_event_data_buffer: throw away 3 items in opcode buffer
No DCS tree to write (nothing has been recorded)

*** Break *** segmentation violation

=====
There was a crash.
This is the entire stack trace of all threads:
=====
gdb.printing.register_pretty_printer(gdb.current_objfile()),
gdb.printing.register_pretty_printer(gdb.current_objfile()),

Thread 4 (Thread 0x7f66ac3ac700 (LWP 18601)):
#0 0x00007f66c09cc3953 in select () at /lib64/libc.so.6
#1 0x00007f66afdc04b7 in UdpSocket::receive(RawPacket&, int) () at /home/rroberts/itsdaq-sw/bin/libstdll.so
#2 0x00007f66afdc2aba in HsioAccess::receive(std::vector<HsioOpcode>, std::allocator<HsioOpcode> >>, unsigned short&, int) () at /home/rroberts/itsdaq-sw/bin/libstdll.so
#3 0x00007f66afda4787 in HsioBuffer::run() () at /home/rroberts/itsdaq-sw/bin/libstdll.so
#4 0x00007f66c0f67070 in () at /lib64/libstdc++.so.6
#5 0x00007f66bf729e65 in start_thread () at /lib64/libpthread.so.0
#6 0x00007f66c09cc88d in clone () at /lib64/libc.so.6

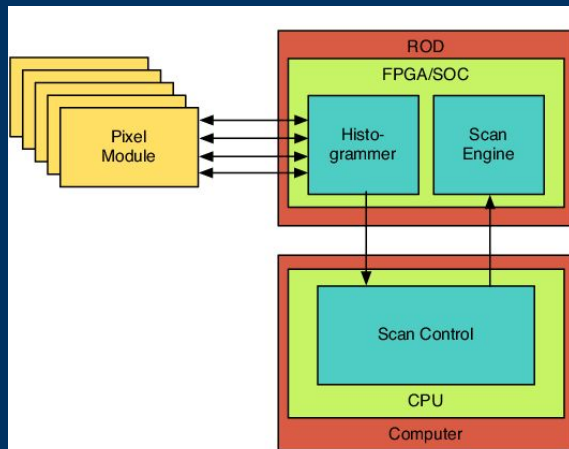
Thread 3 (Thread 0x7f66a227b700 (LWP 18804)):
#0 0x00007f66bf72dda2 in pthread_cond_timedwait

GLIBC 2.3.2 () at /lib64/libpthread.so.0
#1 0x00007f66afda1b51 in std::thread::Impl<std::Bind_simple<work_initWorkerThreads()::{lambda()#1} ()> >::M_run() () at /home/rroberts/itsdaq-sw/bin/libstdll.so
#2 0x00007f66c0f67070 in () at /lib64/libstdc++.so.6
#3 0x00007f66bf729e65 in start_thread () at /lib64/libpthread.so.0
#4 0x00007f66c09cc88d in clone () at /lib64/libc.so.6
```

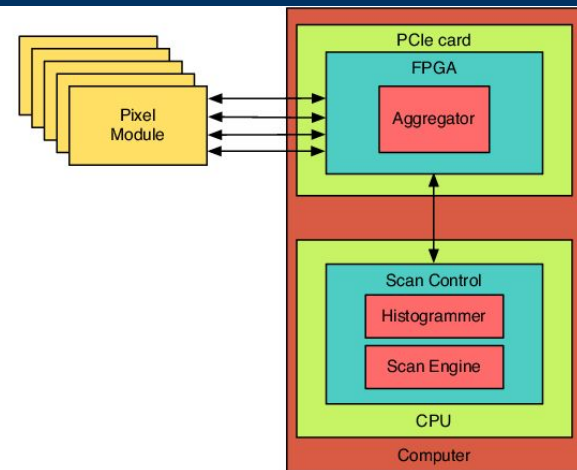
First Steps with YARR

We don't have access to FELIX hardware yet, so we have been running YARR in two other configurations

1. Nexys board running the same firmware as ITSDAQ
2. Using YARR's built in emulation



(a) Traditionally used readout architecture.

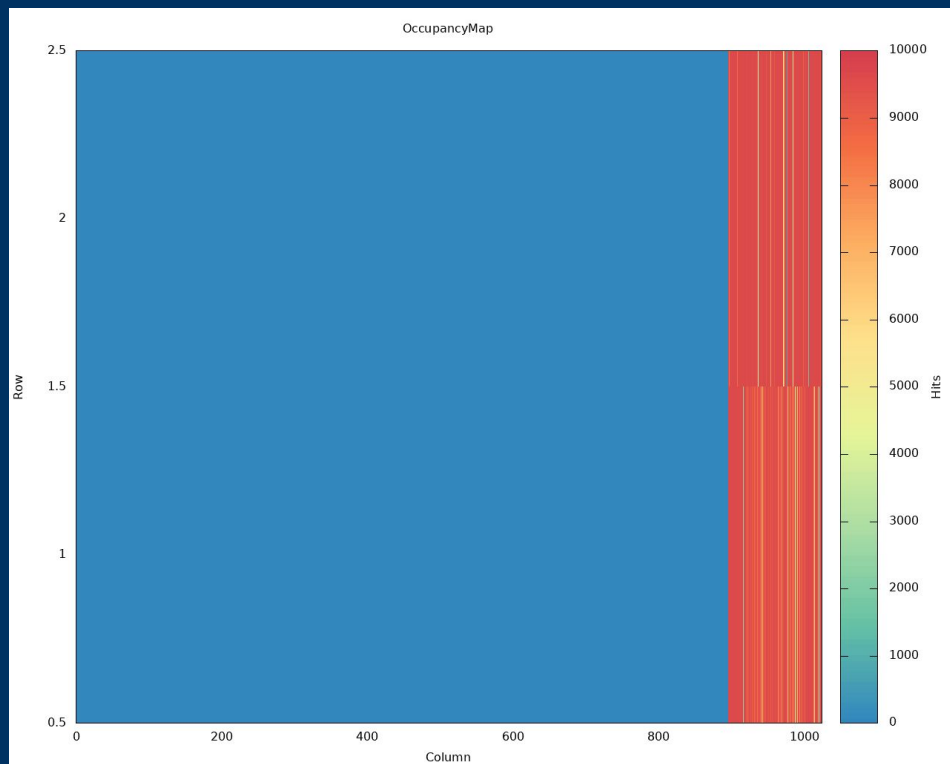


(b) Readout architecture enabled by YARR.

First Data from YARR

Our first data in YARR was a simple occupancy map to confirm that the data is being collected correctly by the software.

This data from a run of 10,000 triggers sent to the DAQ load shows hits only in the extant ABCStar and most channels have exactly 10,000 hits.

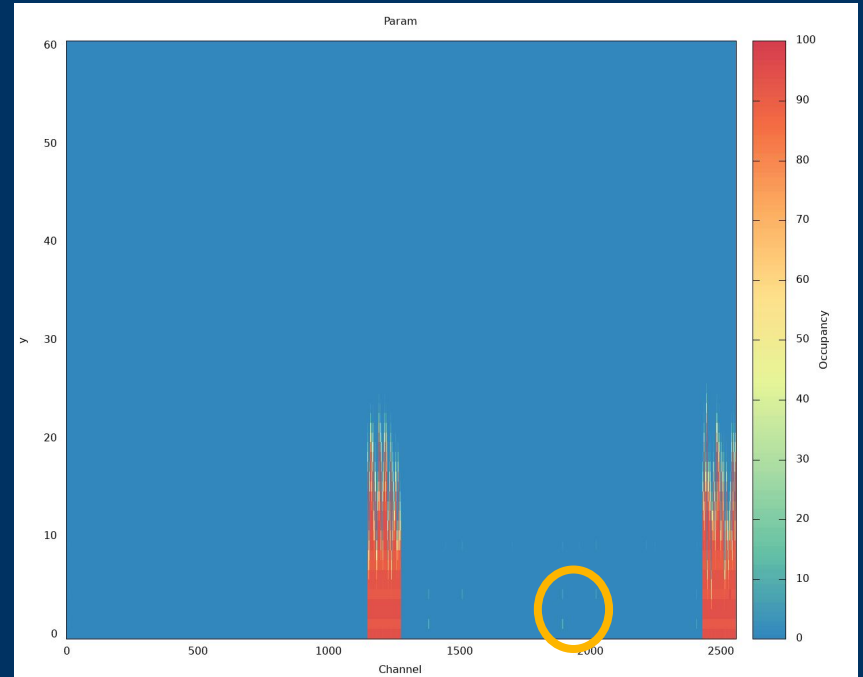


Features added to YARR

- Histogram analysis to show hit occupancy by channel and scan value
- Loop to dump HCCstar and ABCstar register values (specified or all registers)
- Trigger loop using hit counters rather than LP physics packets to transmit data
 - Not able to histogram data yet

Threshold scan in YARR

Basic behavior is as expected.
Horizontal stripes indicate missing data corresponding to packets with missing words.
Some data migrates to non-existent channels.

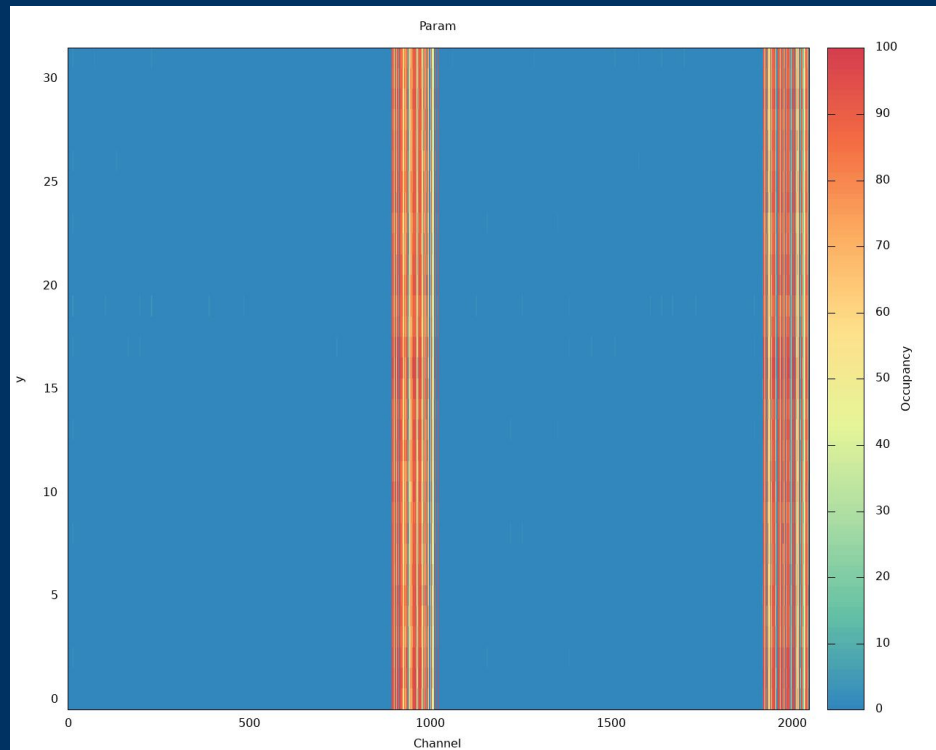


Strobe Delay/Pulse Size scan in YARR

Same behavior scanning over strobe delay or calibration pulse size.

Channels with most activity are those with higher noise in threshold scan.

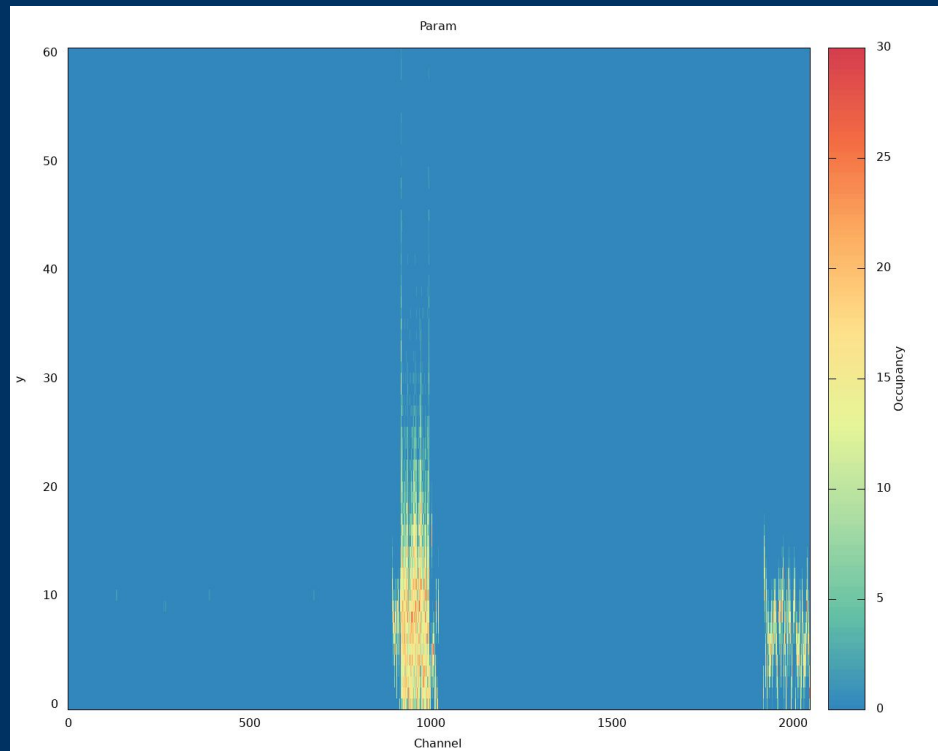
Indicates injection not happening?



Recent new behavior: even less data

After refreshing the setup (including some changes to initial register values), things look even worse.

Threshold scan shown here.



Future Work

- Working on a scan using digital injection
- Need to implement pedestal trim
- Need to get Itsdaq functioning again
- Need to understand missing words