# Developing DAQ and testing strip staves in SR1

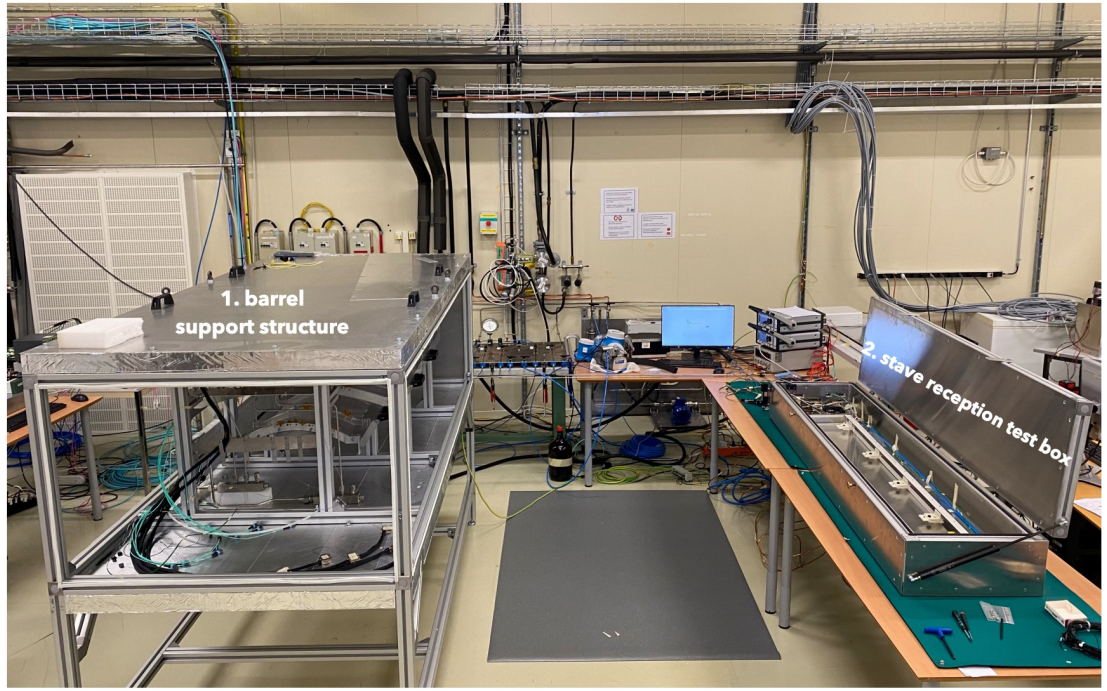
Elise Le Boulicaut, Duke University

LBL Instrumentation Meeting

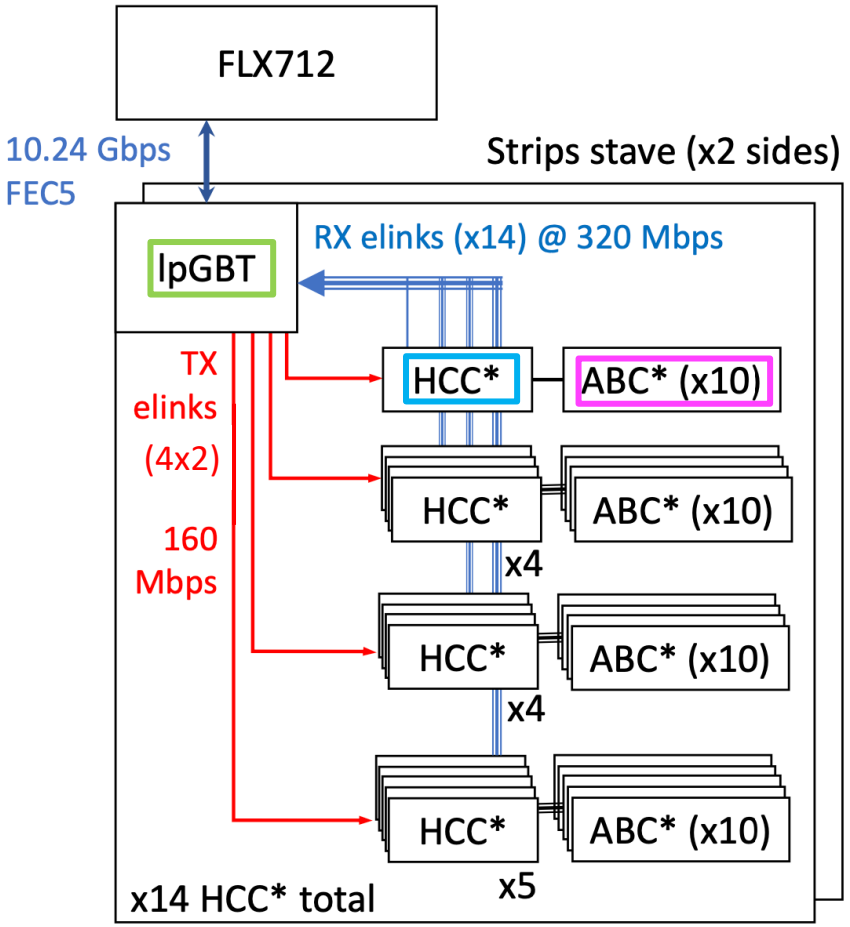
March 24, 2023

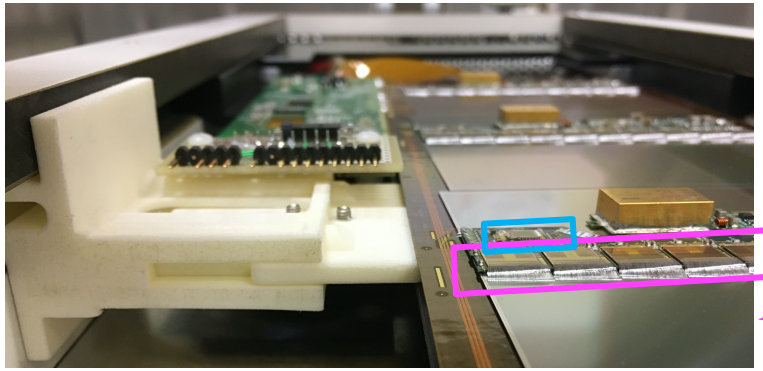
# Strips Barrel System Tests in SR1

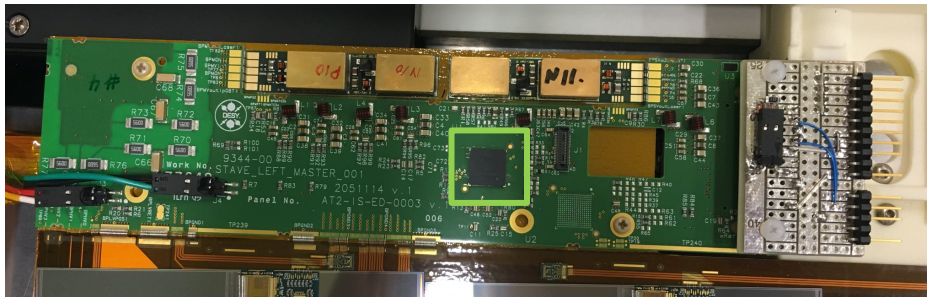
- **Goal**: Demonstration of full ITk Strip barrel system (powering, cooling, readout, …).
- Tests carried out in **SR1 radiation lab**.
- Results will be used for **Systems FDR**.
- Types of pre-production staves:
  - **PPA**: ABCStar v1, HCCStar v0, AMAC v2
    → preparation for PPB
  - **PPB**: ABCStar v1, HCCStar v1, AMACStar
    → most important for FDR
  - **Long-strip** (LS): one HCC per module, one lpGBT per side
    → 2 FELIX links per stave
  - **Short-strip** (SS): two HCCs per module, two lpGBTs per side
    → 4 FELIX links per stave
- Plans for system tests in SR1:
  - PPB: 1 LS + 3 SS → requires $1\times2 + 3\times4 = 14$ links
  - Currently, all 4 staves are mounted in barrel stave support structure.
- **Readout**: Genesys/ITSDAQ as reference (since used in other institutes), but also running FELIX/YARR in order to validate it.



# Strip Stave overview

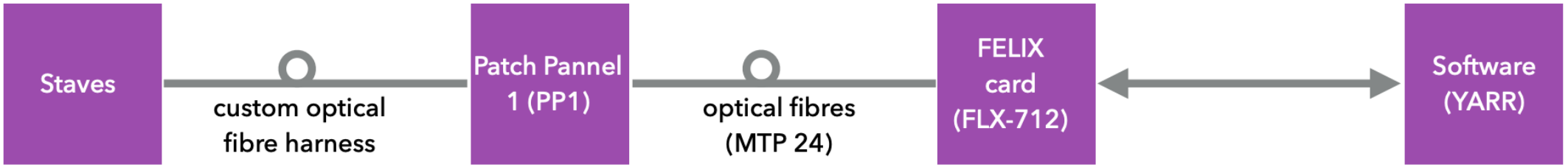


lpGBT

# FELIX Readout System




- Using **FELIX server** in the racks outside the radiation lab in SR1.
- Using **16-channel firmware** build.
- Using **Felixcore** (not yet switched to felix-star).

*Diagram from Dominique*

# Strips scans in YARR

...

# Trim scan

- **Goal:** optimize the per-channel (i.e. strip) thresholds in order to have a more uniform response.

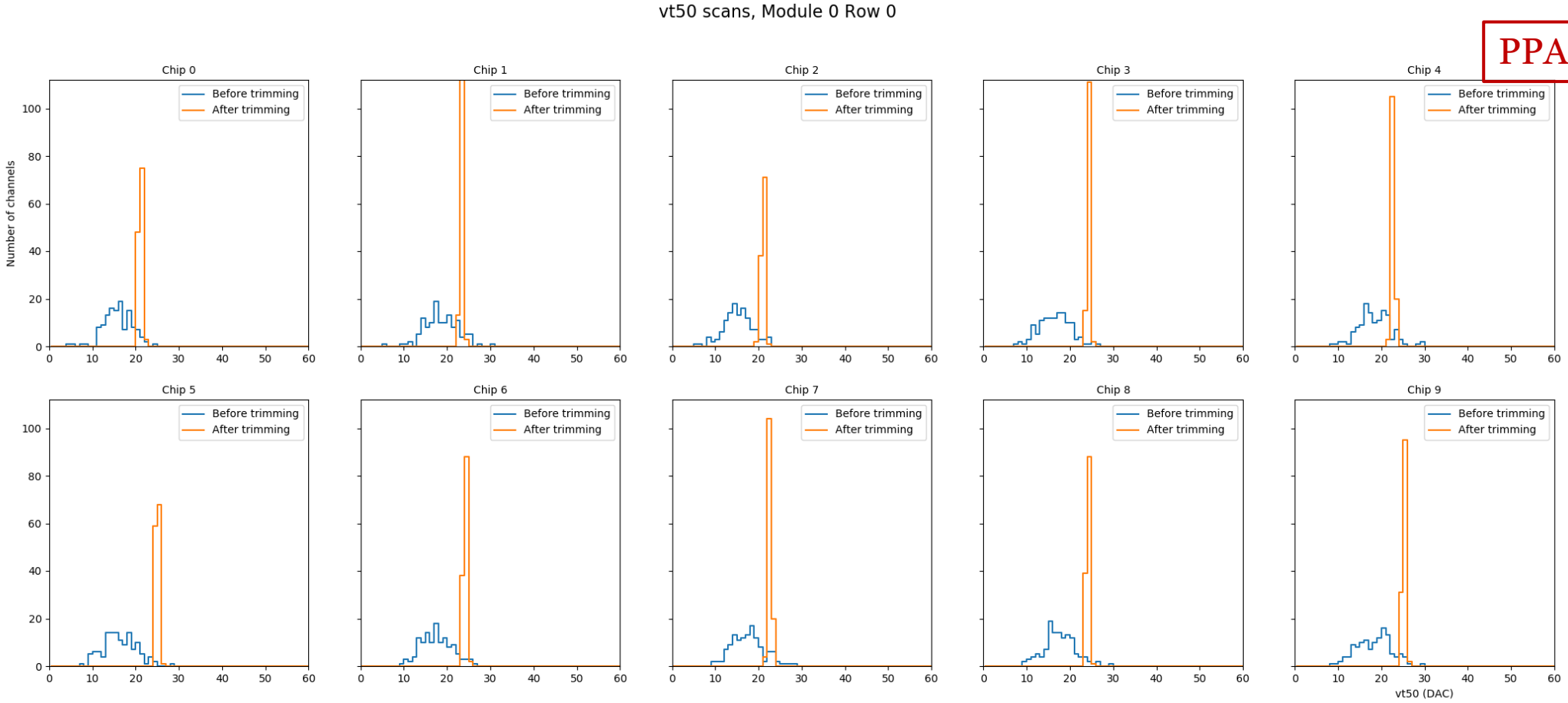
- Scan over **threshold** (BVT) for different values of **TrimDAC** and **trim range** (BTRANGE).
- For each strip, each TrimDAC, and each trim range, fit (reverse) s-curve and determine **vt50**.
- Find **best target threshold** by finding which vt50 value can be attained by a maximum number of strips.
  - Target can be determined per ABC or per HCC.
- Find **best trim range** for each chip and **best TrimDAC** for each strip and such that they attain the target.

- In practice, the trim range is kept fixed to a previously optimized value to decrease the scan time

# Trim scan results

Comparing vt50 distributions from threshold scan before and after trimming → trimming has the desired effect.

PPA LS stave

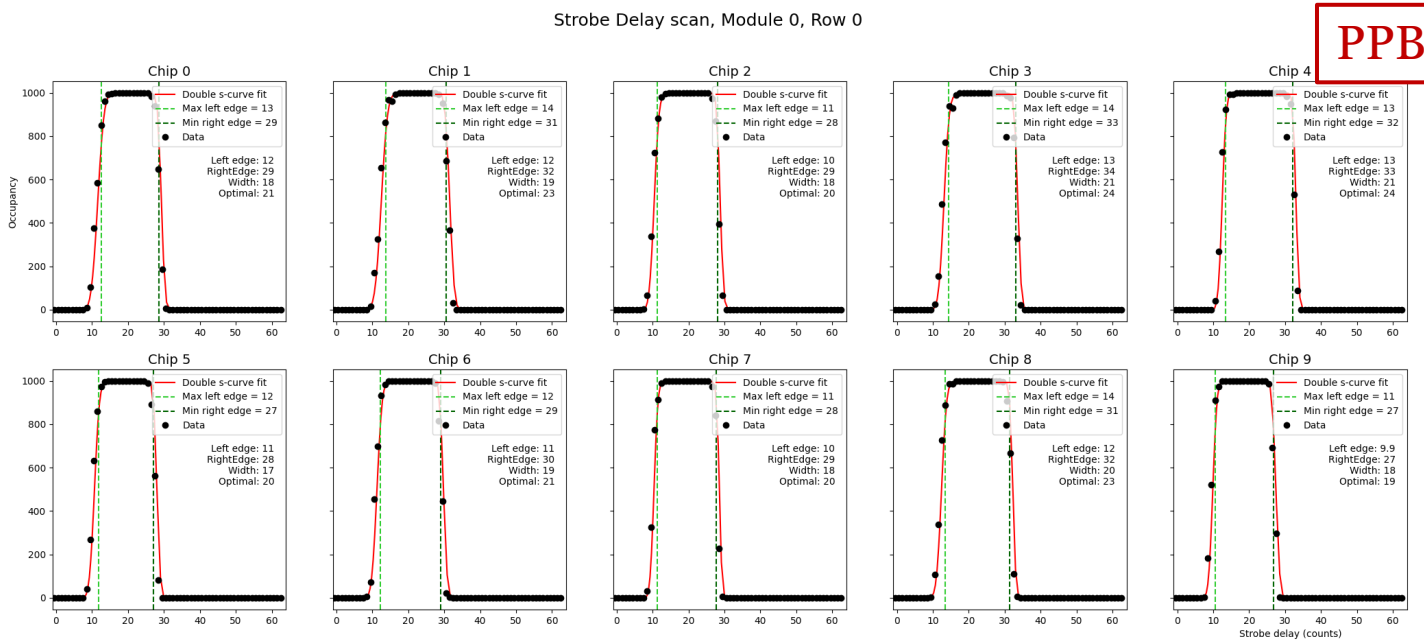


# Strobe delay scan

- **Goal:** optimize the delay of the charge injection with respect to the clock phase (strobe delay).

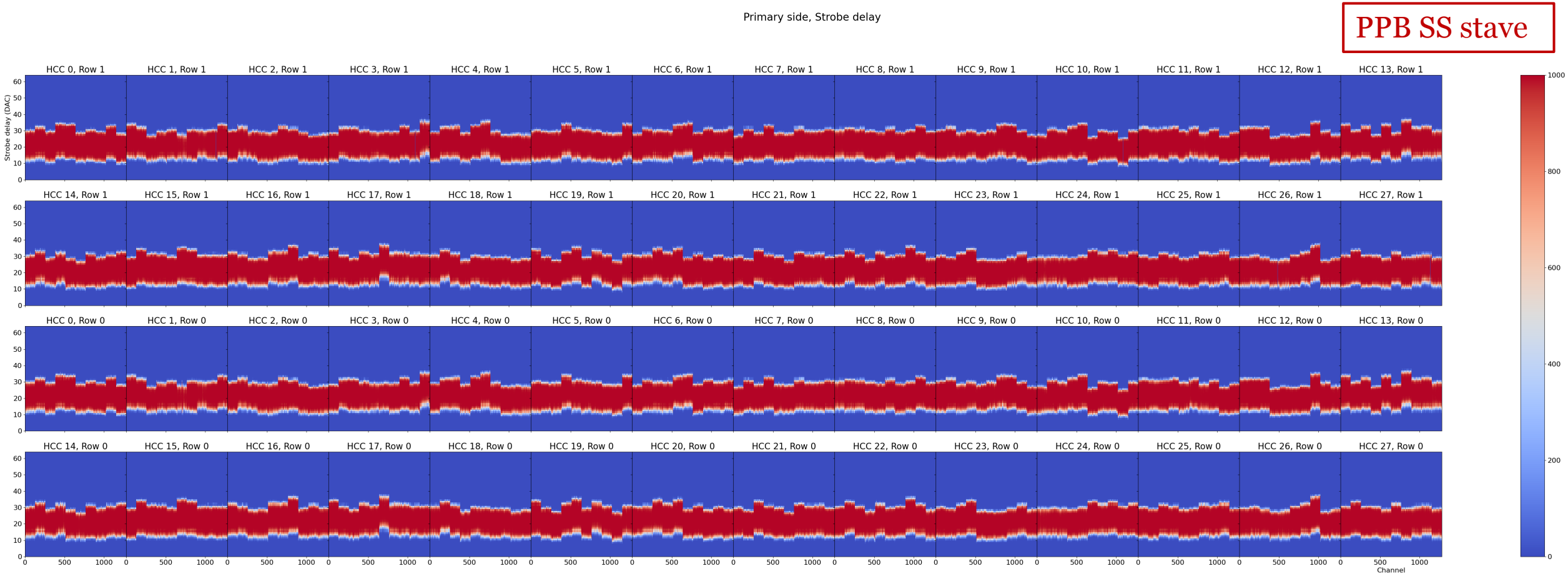
- Scan over **strobe delay** using edge detection mode.
- For each strip, obtain occupancy vs strobe delay (should look like a square pulse), fit double s-curve, and determine **rising and falling edges.**
- Find **maximum rising edge** and **minimum falling edge** for each ABC.
- Optimal strobe delay is 57% between these two points.

PPB SS stave



# Strobe delay scan results

PPB SS stave

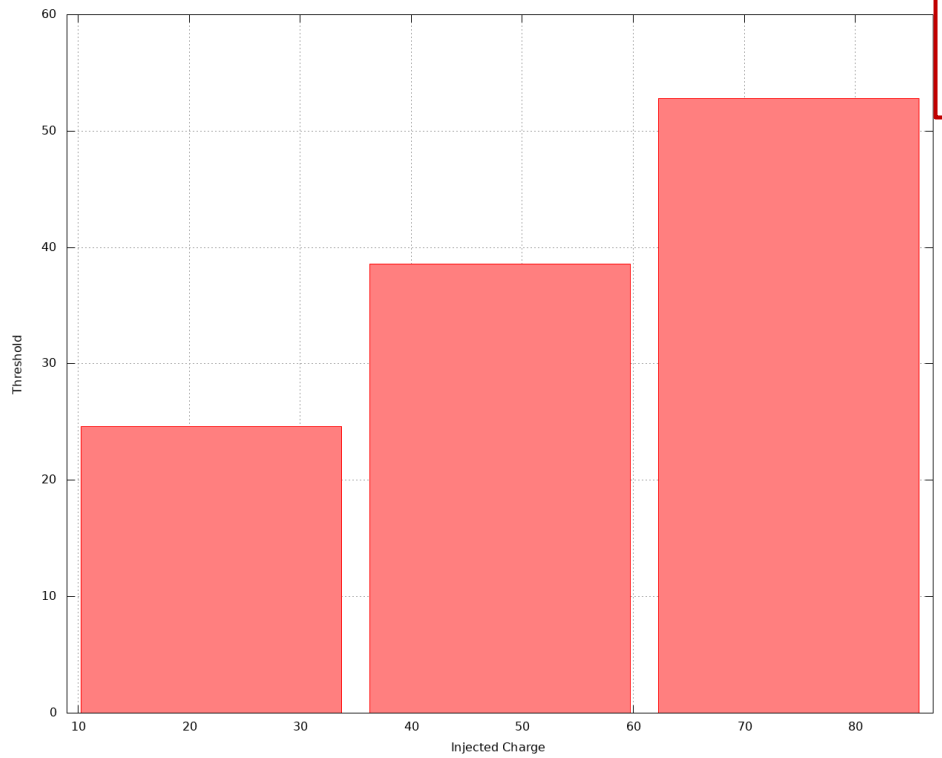


# N-point gain scan

- **Goal:** obtain the gain in order to provide a calibration to convert thresholds into fC.

- Scan over **threshold** (BVT) for N different **injected charges** (BCAL).
- For each strip and each BCAL, fit (reverse) s-curve and determine **vt50** and **output noise**.
- Plot vt50 vs charge (response curve) and fit to linear (or polynomial, or exponential) to determine the **gain**.
- Use the gain to determine the **input noise**.

- Response curve can be plotted and fitted either per strip, per ABC, or per HCC.

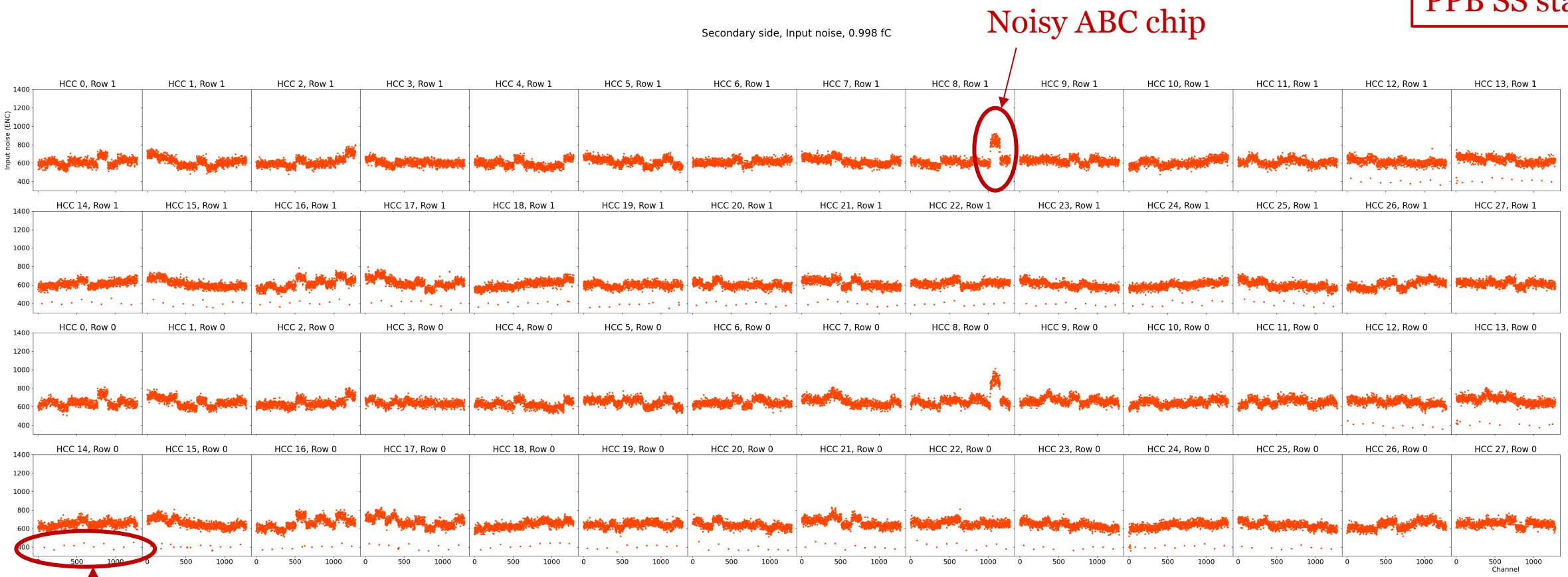



PPB SS stave
One HCC

# N-point gain scan results

## Input noise as a function of channel number for a full stave side:

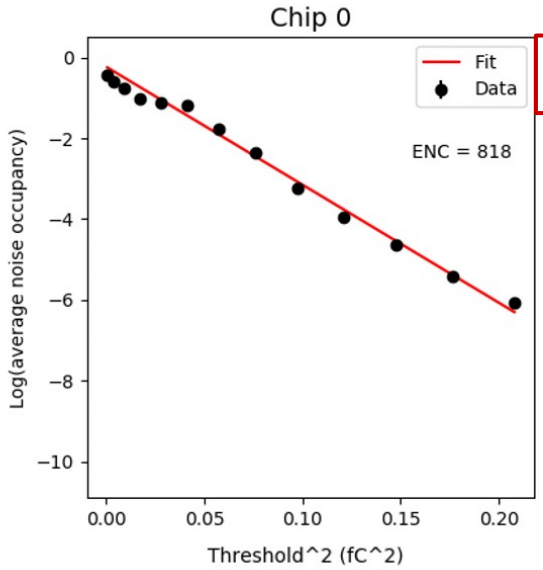
PPB SS stave



# Noise occupancy scan

- **Goal:** determine the equivalent noise charge (ENC) for each ABC.

- Scan over **threshold** with **no injected charge**.
- Throttle the number of triggers so that higher thresholds get larger number of triggers (to get sufficient statistics).
- Plot log(average relative noise occupancy per ABC) vs threshold² (in fC²).
- Fit to a linear function and extract the **ENC** from the slope.




PPA LS stave

# Running YARR scans in SR1

- Created a config-based script to do the following:
  - Get tx/rx/efuseID mapping given FELIX device and optical link numbers.
  - Get the register values from ITSDAQ for registers that need to be tuned per-chip (BVREF, BIREF, B8REF, COMBIAS, ADC_BIAS, LDOD, LDOA).
  - Select which HCCs we wish to run on and write their chip configuration files.
  - Get data from previous scans if needed (optimized trims and strobe delay, response curve calibration).
  - Send the `scanConsole` command to launch the scan (can launch multiple processes in parallel if running on multiple staves).

  ⇒ At this stage in system tests, it is not feasible to do all this by hand! However, this is still only a temporary solution.

  See this git repo for code and README.

# YARR/ITSDAQ comparison

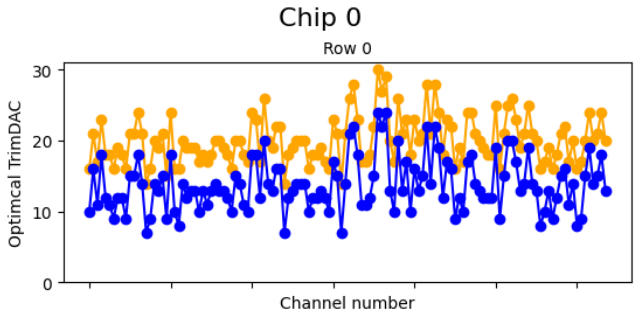
# Introduction

- Two main **readout systems** for ITk strips
  - **FELIX/YARR**:
    - Uses “official” TDAQ software
    - Common with pixels
    - In principle can be scaled to the entire ITk
    - Not widely used for strips
  - **ITSDAQ**:
    - Only for strips
    - Not easily scalable to multiple staves
    - Used in many institutes for QA/QC
- Ran some quantitative comparisons between YARR and ITSDAQ scan results to 1) validate YARR and 2) identify any potential algorithm differences that could be discussed.

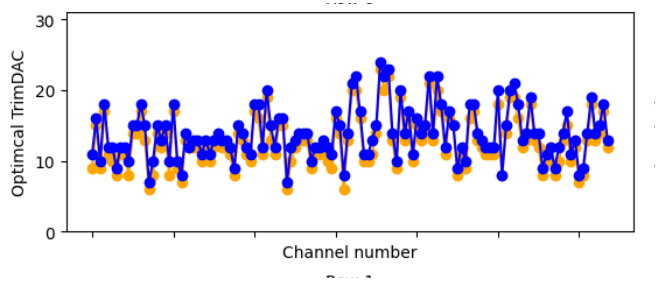
# Tim scan

- ITSDAQ uses a simplified approach: fix BVT to 15, loop over TrimDAC only, and choose the value which leads to 50% occupancy.
- Results should be similar if the optimized target in YARR is close to 15. But this is not always the case.
- Modified the YARR algorithm slightly to find a target as close as possible to 15 → good agreement. Still unsure if we want to keep this.

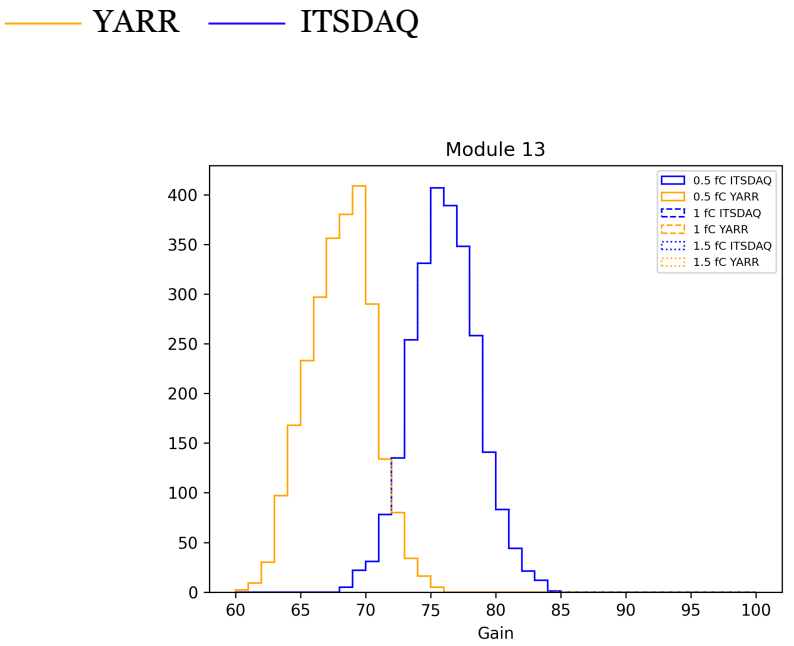
PPA LS stave



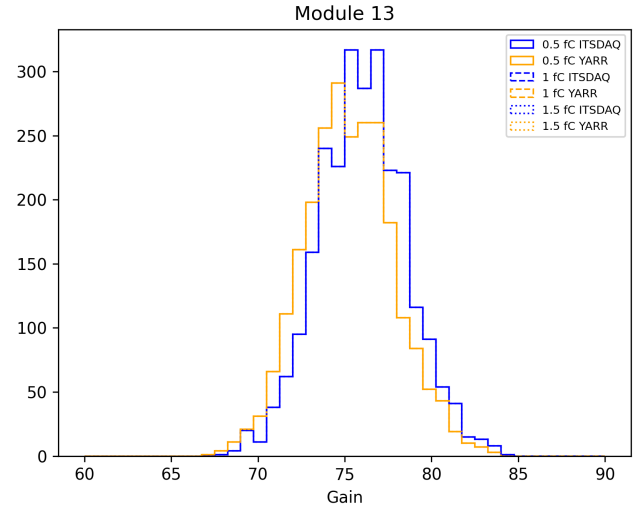

*YARR with optimized trim target*




*YARR with trim target as close as possible to ITSDAQ*



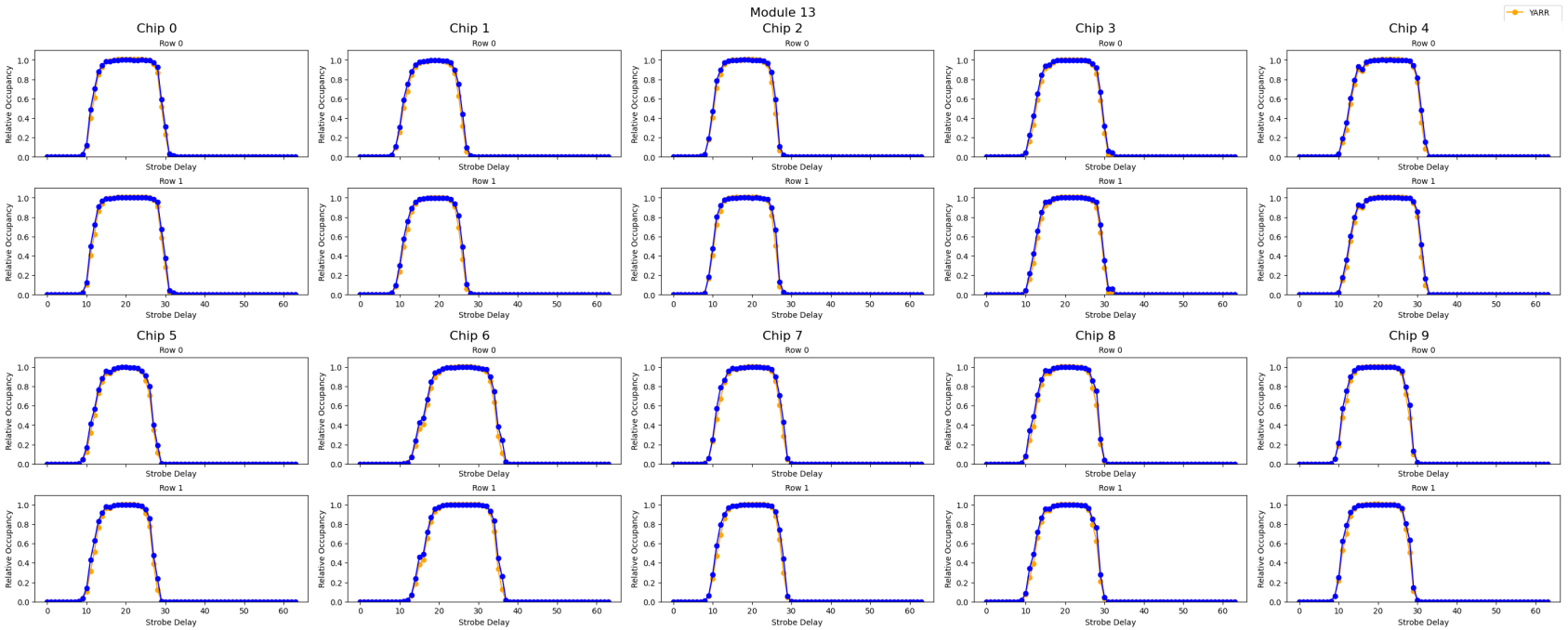

*Gain distribution with YARR using optimized trim target*

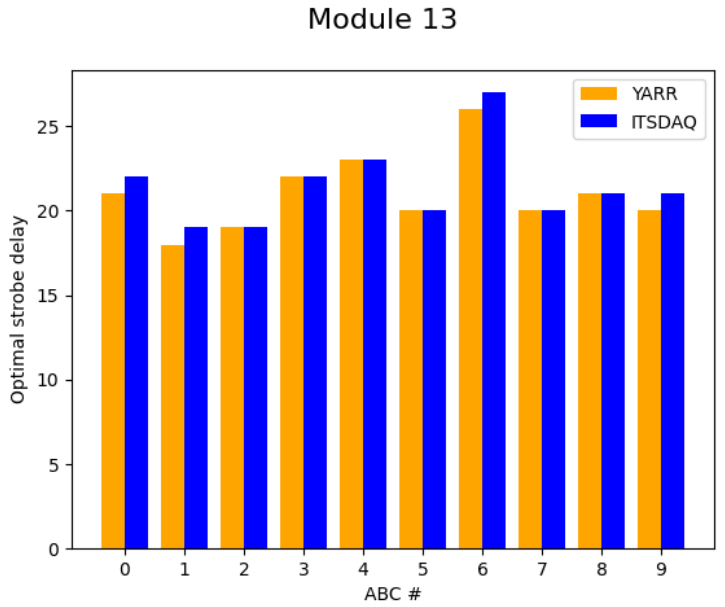



*Gain distribution with YARR using trim target as close as possible to ITSDAQ*

# Strobe delay scan

PPA LS stave



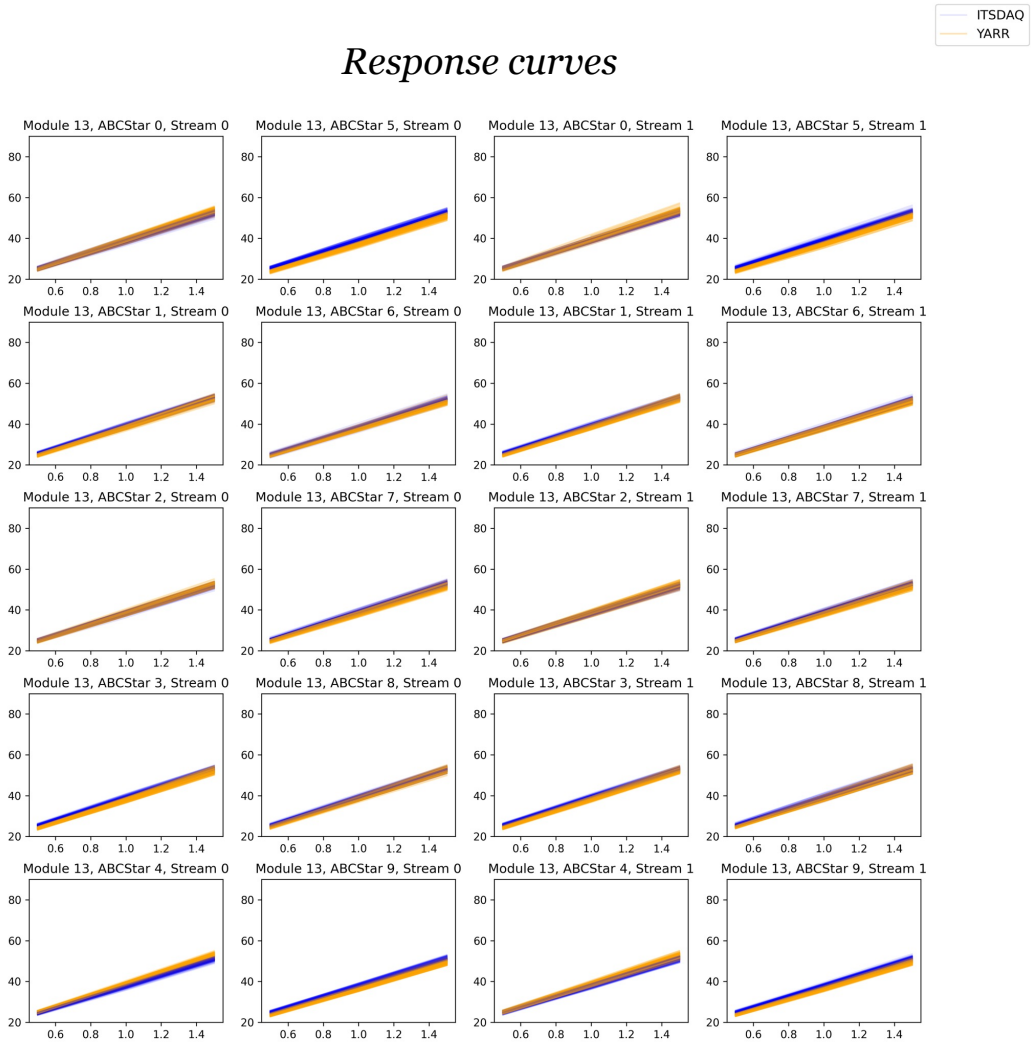




⇒ Good agreement overall.

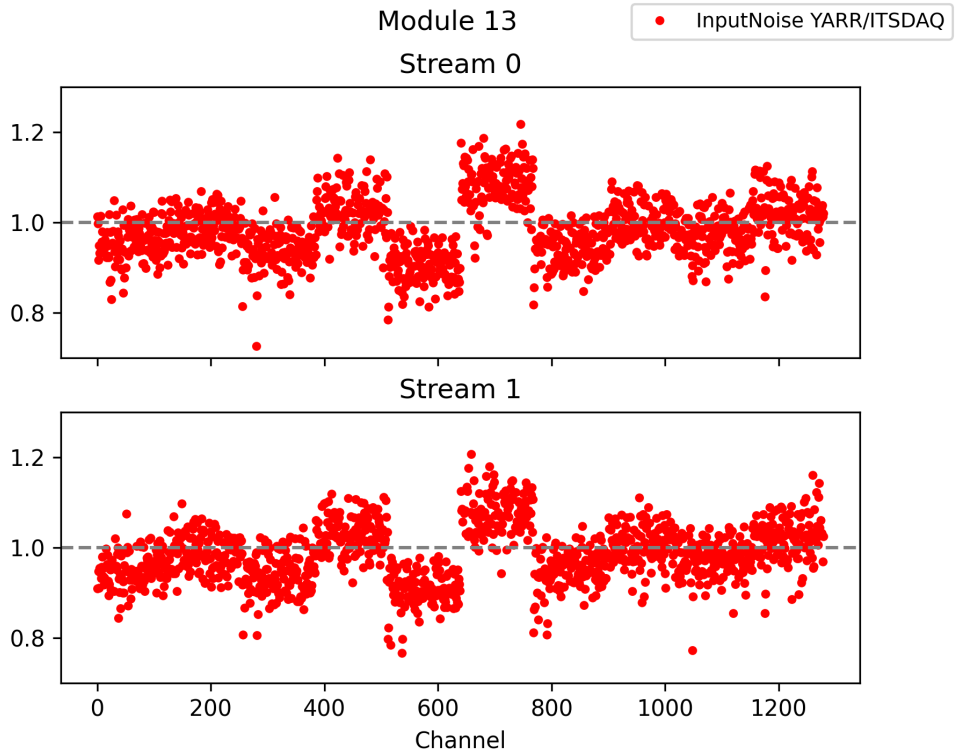
# N-point-gain scan

PPB SS stave

*Response curves*



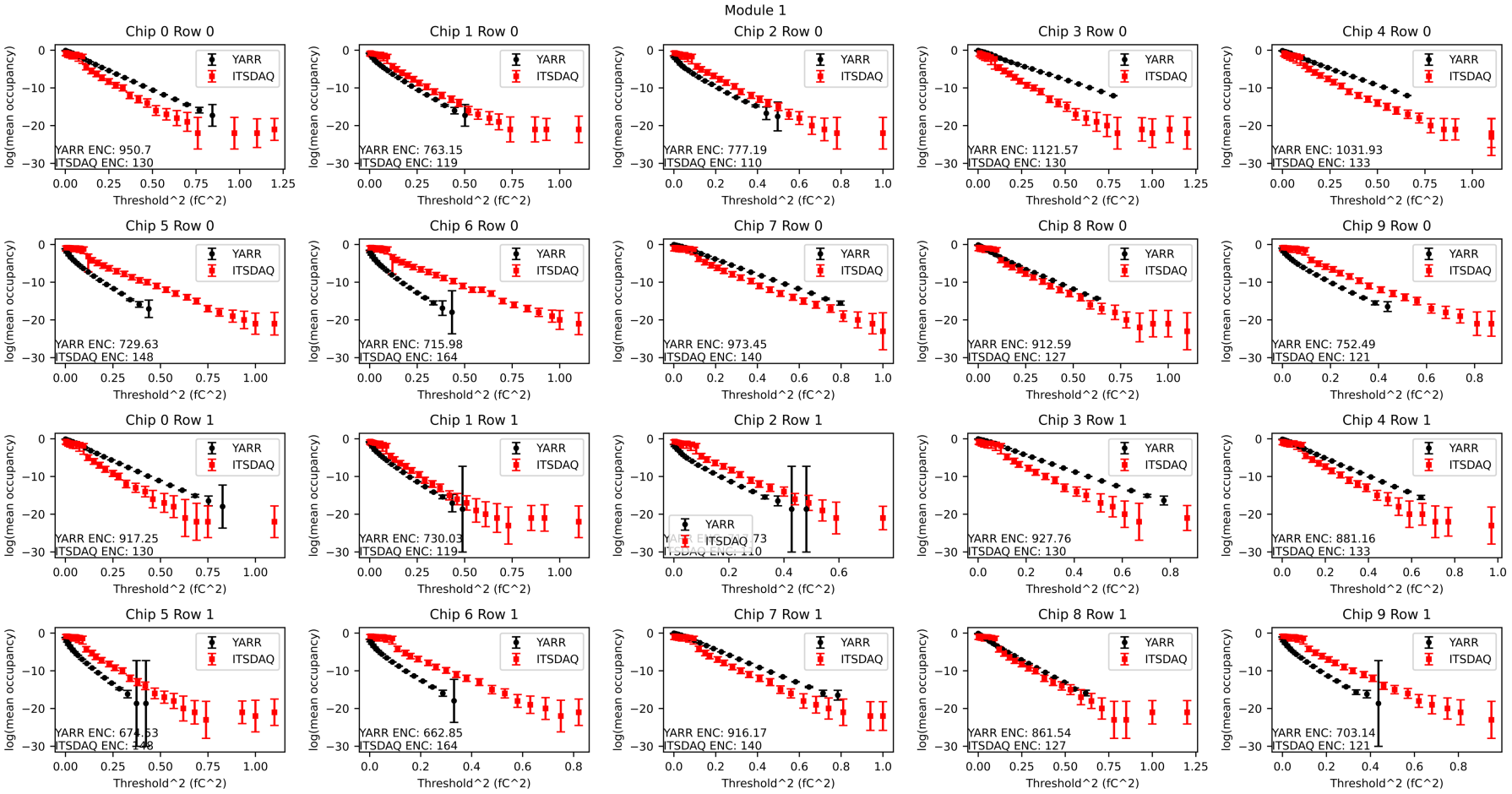

*Input noise ratio*




⇒ Overall agreement is OK but some ABCs have larger differences than others.

# Noise occupancy scan

PPA LS stave



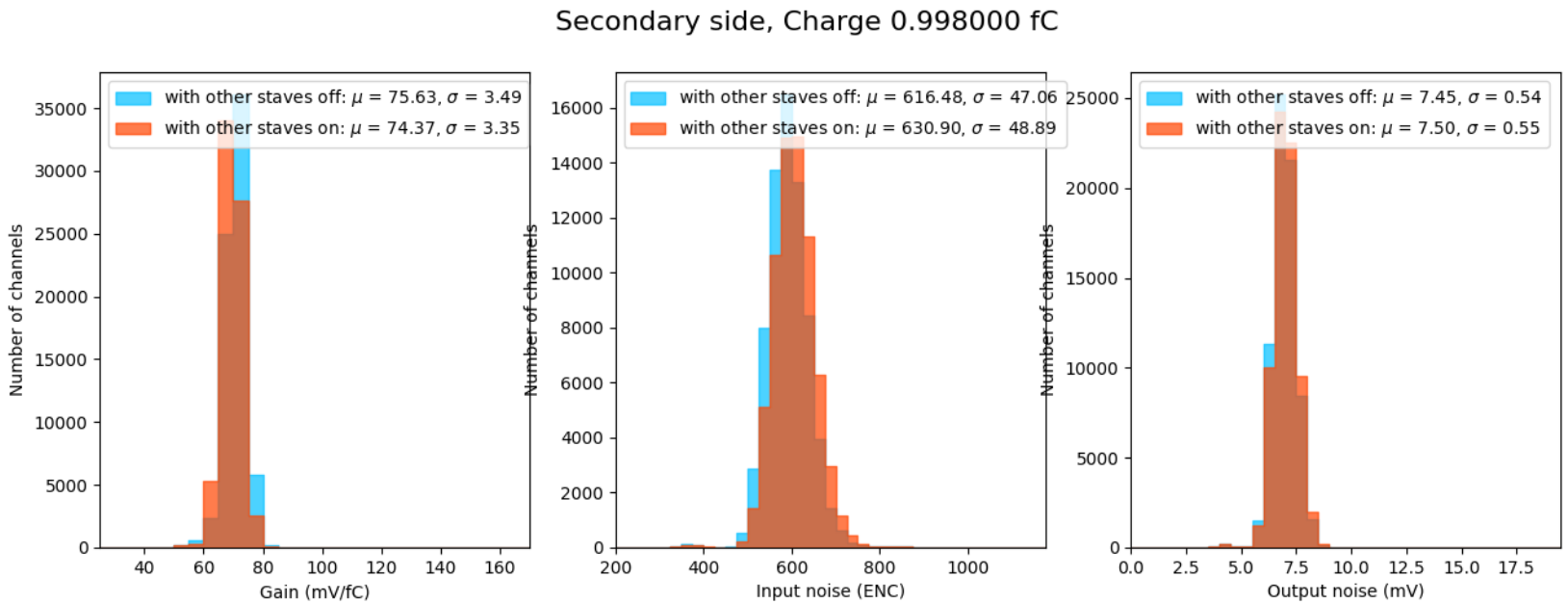

Identified some significant differences:

- Different response curves → **different calibrations** for the x-axis.
- **Triggering** is different, allowing ITSDAQ to send more triggers faster.
- **Error bars** are calculated differently.

# Performance

# Cross talk tests

To test whether we see noise due to **cross-talk** between neighbouring staves, ran series of scans on one stave side (secondary side of SS PPB stave) with all other stave sides off. Compared three-point-gain results with the same stave side when all other staves were on and running:

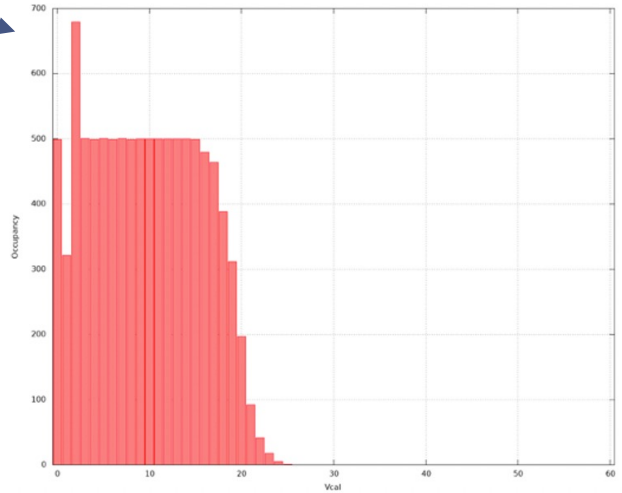



⇒ No change in output noise. Slightly lower gain and higher noise when other staves are on, but difference is small (1/3 of a sigma) and could simply be due to temperature.

# Running on 2 staves in parallel

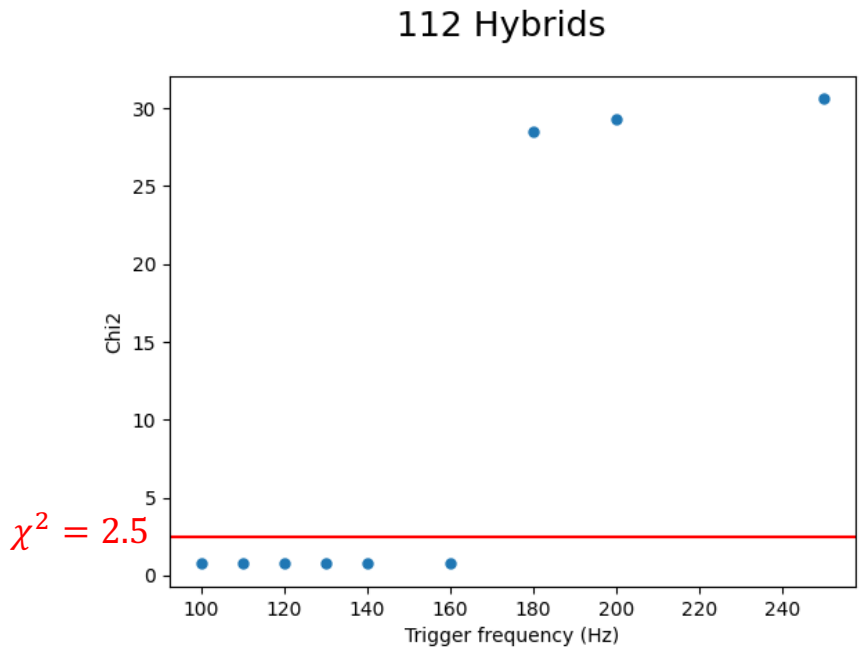
- The largest scale tests that have been run so far are trim, strobe delay, and three-point-gain scans on **two SS PPB staves in parallel** (112 HCCs).
- Used script shown on slide 13 to launch one YARR process per stave (in principle possible to run on two staves with one process but it's a bit slower).
- Noticed a **data migration** problem, which got worse with increasing number of hybrids → needed to decrease the trigger frequency to 100 Hz.
- Scan times (at 100 Hz):
  - Trim scan: 2.5 h
  - Strobe delay scan: 13 min
  - Three-point-gain scan: 15 min

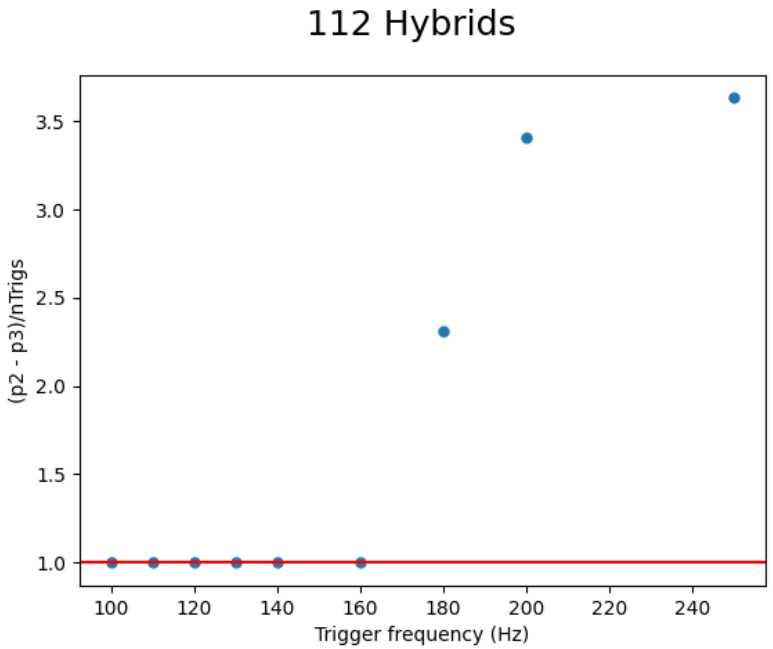
*Example s-curve showing data migration*



# Trigger frequency limitation

- Investigated in more detail the **dependence on the number of FEs** of the trigger frequency limitation.
- For different numbers of FEs, ran vt50 scan and measured average $\chi^2$ of s-curve fits as well as $\frac{p_2-p_3}{N_{trigs}}$ (which should be as close to 1 as possible, since $p_2$ =max and $p_3$ = min ).
- Used these measures to determine when the fits start getting bad (removed all cuts on s-curve fit parameters for this test).

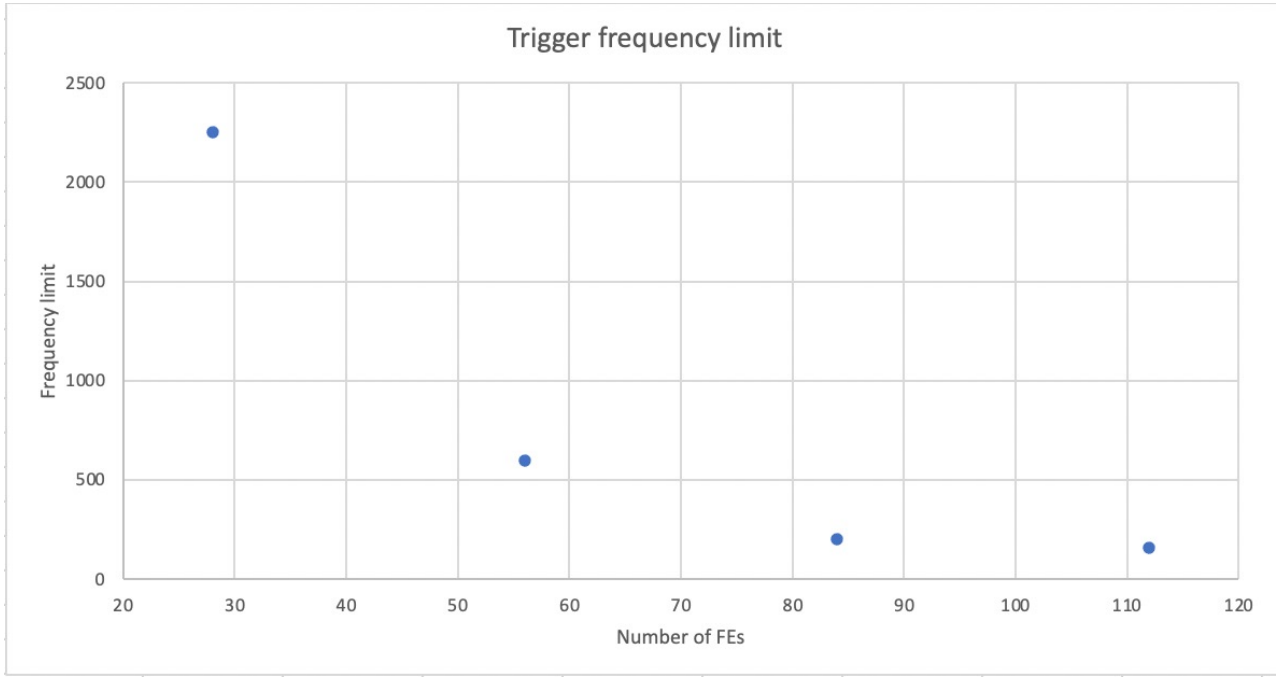


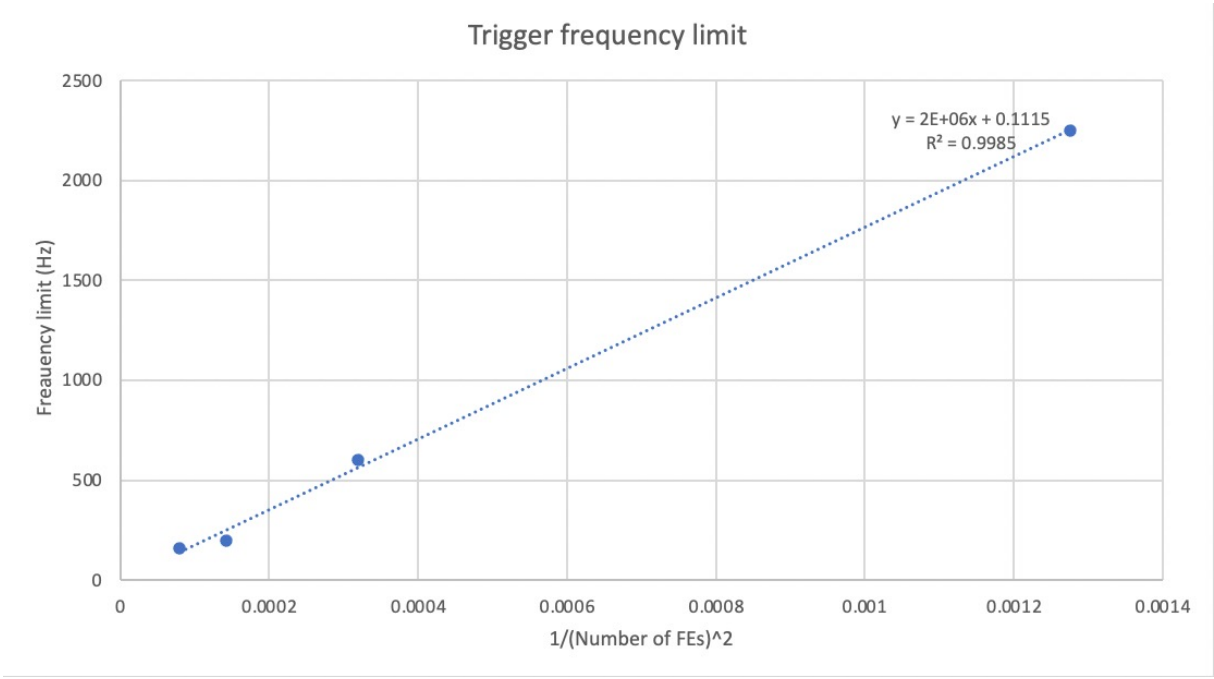




*Example diagnostic plots for 112 FEs, showing that the frequency limit is between 160 and 180 Hz.*

# Trigger frequency limit vs number of FEs

Determined trigger frequency limit (highest frequency for which we don't see problems in the s-curves) for different numbers of front ends:




⇒ Dependence is $\sim 1/N^2$

# Possible solutions to tackle trigger frequency limitation

- **Feedback-based readout** (MR): count number of triggers received and make sure it matches number of triggers sent before moving to next iteration.
- **Broadcast elink** (FELIX JIRA): would allow to send one trigger command to all FEs at once instead of duplicating the command and sending it on all elinks in parallel.
- **Trickle memory triggers** (MR): Load the trigger sequence into the trickle memory and release it on demand.
- **Firmware triggers**.

# Conclusion and next steps

- All scans needed for strips system tests have been implemented in YARR and successfully run on staves in SR1.
- YARR/ITSDAQ comparisons show good agreement.
- Ran large-scale tests on two SS PPB staves in parallel with YARR.
- Trigger frequency limitation is currently the main bottleneck.
- Next steps:
  - Fix trigger frequency limitation (also necessary for low-occupancy “physics-like” tests needed for the module PRR).
  - Automatically write scan results to chip configs to avoid the need for external scripts between scans.
  - Link to database to avoid having to use ITSDAQ configs.
  - Re-organize/optimize some of the code and merge it into YARR devel.
- Useful links:
  - SR1 documentation website
  - ITk Strips System Tests INT note
  - Latest scan results for two staves in parallel
  - Latest YARR/ITSDAQ comparisons