

1st Time-In Attempt Data Analysis Summary

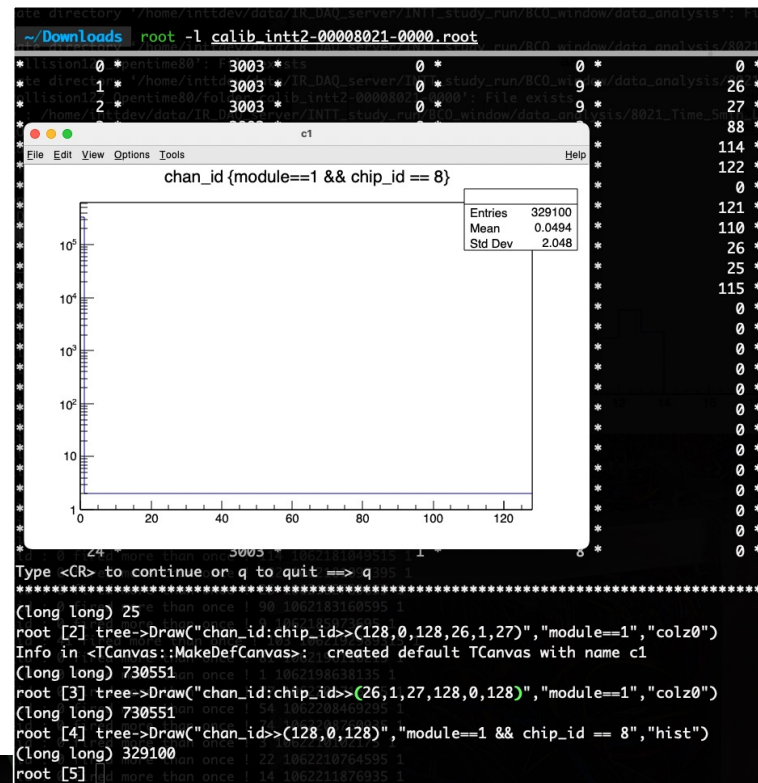
2023/5/30

Special thanks to INTT Team for quick feedback!

Clone Hits (Cheng-Wei)

- Contamination of duplicated data of exactly the same strip channel, chip-ID, half ladder, packet-ID within the same event.
- Potential double counts the hit multiplicity
- This was observed in the Tohoku beam test using FEM+Windows-10 DAQ.
- An indication of FPHX/ROC origin?

calib_intt2-00008021-0000.root (open time 80)
intt2, FC1, chip8



Attaching file calib_intt0-00008021-0000.root as _file0...

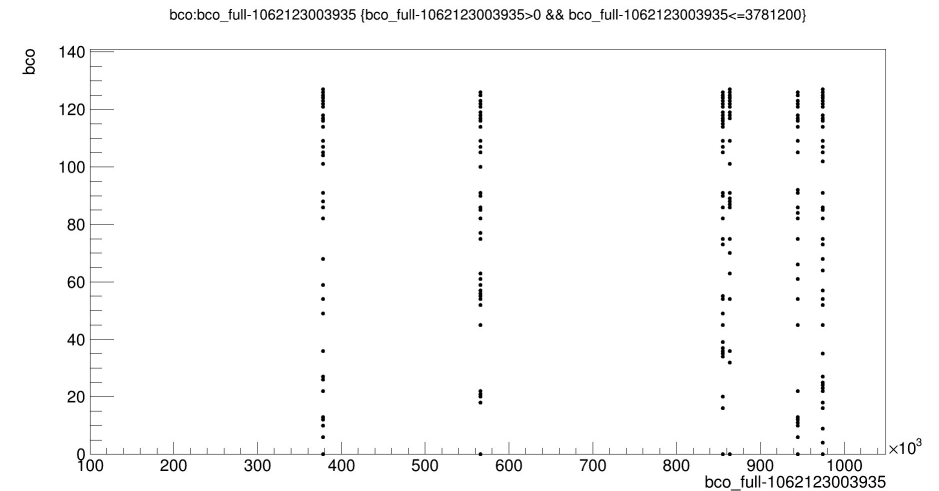
(TFile *) 0x7fc09723f600

root [1] tree->Scan("pid:module:chip_id:chan_id:adc:bco:bco_full", " Entry\$ > 261700", "colsize=15")

* Row *	* pid *	* module *	* chip_id *	* chan_id *	* adc *	* bco *	* bco_full *
* 261787 *	3001 *	0 *	21 *	61 *	0 *	106 *	1062182603015 *
* 261788 *	3001 *	0 *	21 *	61 *	0 *	106 *	1062182603015 *
* 261789 *	3001 *	0 *	18 *	15 *	2 *	106 *	1062182603015 *
* 261790 *	3001 *	0 *	18 *	16 *	2 *	106 *	1062182603015 *

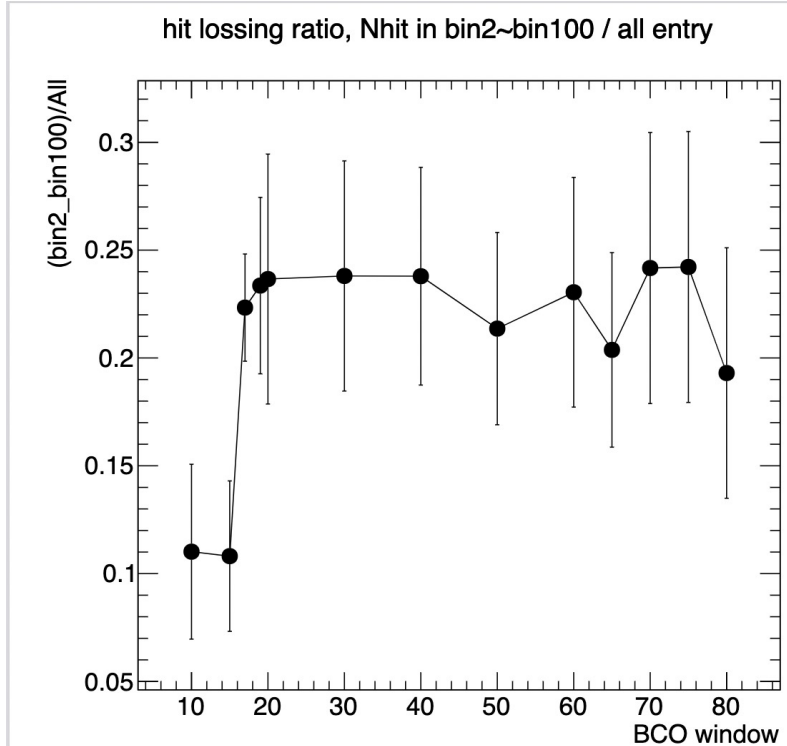
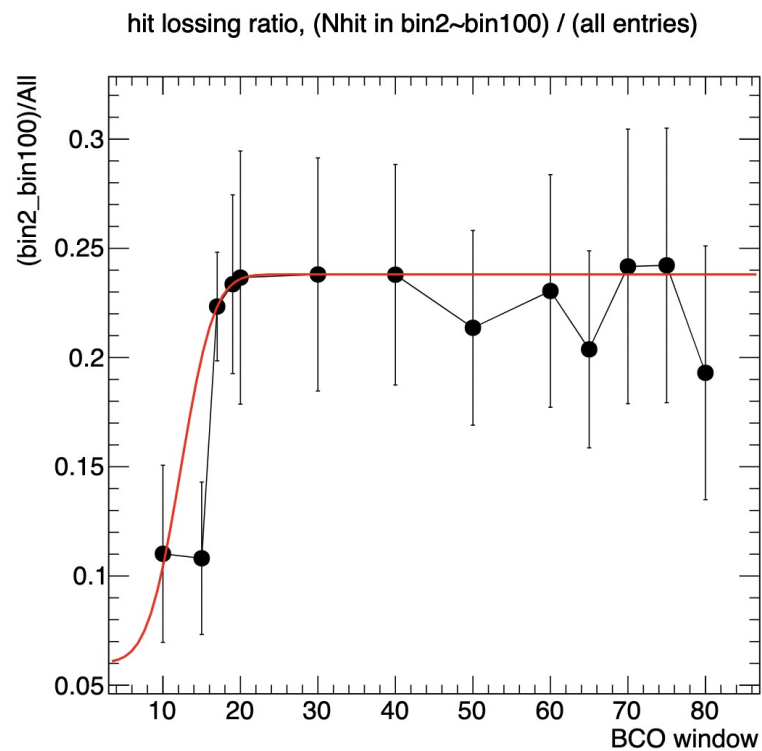
Mix-up of multi-BCO number within a single event (Cheng-Wei)

```
Attaching file calib_intt0-00008021-0000.root as _file0...
(TFile *) 0x7ff4d7512900
root [1] tree->Scan("pid,module:chip_id:chan_id:adc:bco:bco_full","", "colsize=15")
*****
* Row *      pid *      module *      chip_id *      chan_id *      adc *      bco *      bco_full *
*****
* 0 *      3001 *      0 *      0 *      0 *      0 *      0 *      1062123003935 *
* 1 *      3001 *      0 *      24 *      71 *      1 *      2 *      1062123003935 *
* 2 *      3001 *      0 *      24 *      72 *      5 *      2 *      1062123003935 *
* 3 *      3001 *      0 *      25 *      71 *      7 *      2 *      1062123003935 *
* 4 *      3001 *      0 *      25 *      72 *      2 *      2 *      1062123003935 *
* 5 *      3001 *      1 *      0 *      0 *      0 *      0 *      1062123003935 *
* 6 *      3001 *      1 *      7 *      43 *      7 *      2 *      1062123003935 *
* 7 *      3001 *      1 *      17 *      38 *      7 *      2 *      1062123003935 *
* 8 *      3001 *      1 *      2 *      65 *      5 *      2 *      1062123003935 *
* 9 *      3001 *      1 *      22 *      91 *      7 *      2 *      1062123003935 *
* 10 *      3001 *      2 *      0 *      0 *      0 *      0 *      1062123003935 *
* 11 *      3001 *      2 *      22 *      5 *      7 *      2 *      1062123003935 *
* 12 *      3001 *      2 *      23 *      34 *      6 *      2 *      1062123003935 *
* 13 *      3001 *      2 *      23 *      35 *      4 *      2 *      1062123003935 *
* 14 *      3001 *      2 *      13 *      90 *      2 *      2 *      1062123003935 *
* 15 *      3001 *      2 *      12 *      90 *      7 *      2 *      1062123003935 *
* 16 *      3001 *      2 *      19 *      110 *      7 *      2 *      1062123003935 *
* 17 *      3001 *      2 *      20 *      8 *      0 *      2 *      1062123003935 *
* 18 *      3001 *      2 *      20 *      9 *      1 *      2 *      1062123003935 *
* 19 *      3001 *      2 *      22 *      4 *      1 *      2 *      1062123003935 *
* 20 *      3001 *      2 *      20 *      10 *      2 *      2 *      1062123003935 *
* 21 *      3001 *      2 *      15 *      29 *      1 *      13 *      1062123003935 *
* 22 *      3001 *      3 *      10 *      127 *      1 *      0 *      1060594465395 *
* 23 *      3001 *      3 *      16 *      127 *      0 *      35 *      1060594465395 *
* 24 *      3001 *      3 *      9 *      27 *      1 *      35 *      1060594465395 *
*****
Type <CR> to continue or q to quit ==> q
```



- This is the consequence of the $n_{\text{collision}} > 0$. The Run=8021 was taken with $n_{\text{collision}}=127$. So any BCO number can be mixed up within the given event (=given trigger).
- We suppose to see a lot less mixed up with $n_{\text{collision}}=2$ data sets.
- And should no mix-up in $n_{\text{collision}}=0$ data taking. Must be checked when we take such a data.

Open Time Window Scan (Cheng-Wei)

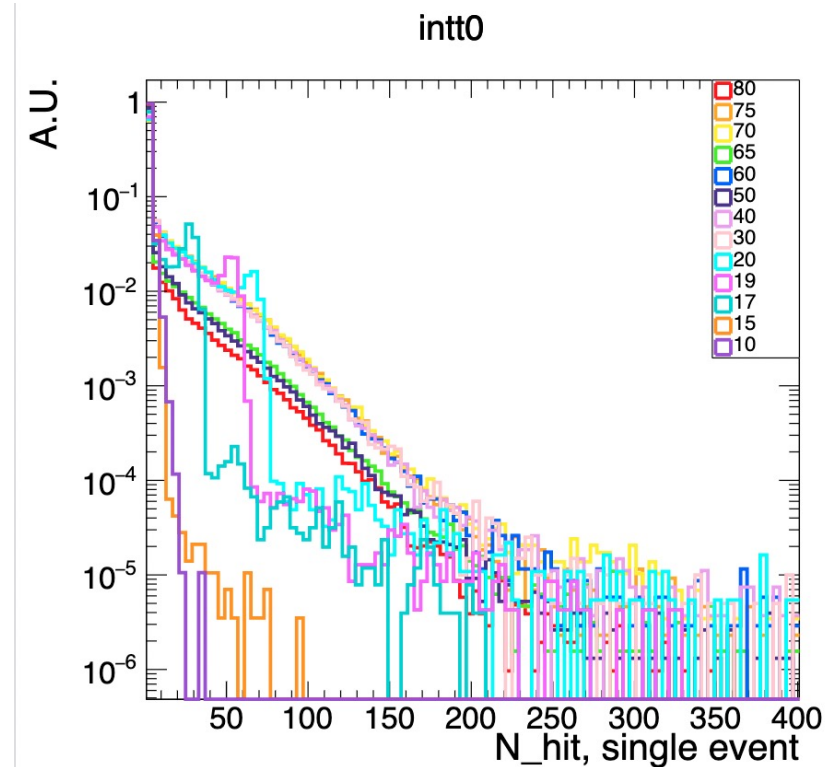


Open Time Window Scan (Cheng-Wei)

- My theory is this open time scan with $n_{\text{collision}}=128$ won't allow me to predict L1delay value to time-in. However, we could test my high multiplicity drop prediction as a function of open time. We'll investigate if our understanding of transmission speed is consistent with this observation.

ACTION ITEM: Calculate the #hit/BCO/datafiber Data transfer limit and compare with calculated limit

/home/inttdev/data/IR_DAQ_server/INTT_study_run/BCO_window/data_analysis/8020_Time_5
min_L1Delay00_Ncollision127_Opertime120



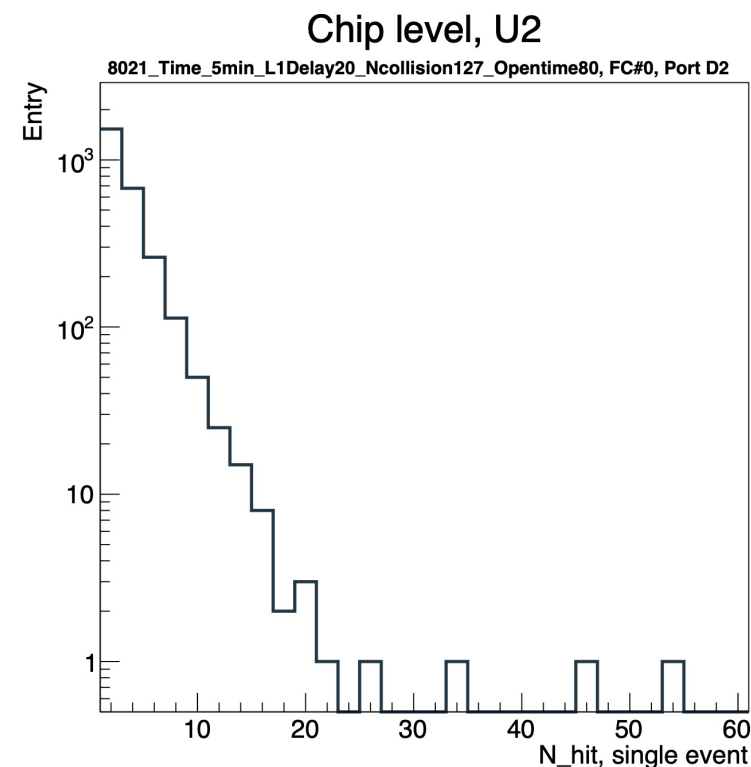
Chip Multiplicity (Cheng-Wei)

FPHX Manual

The experiment will use a 106 ns beam clock. Each FPHX chip integrates and shapes (CR-RC) signals from 128 channels of mini-strips, digitizes and sparsifies the hit channels each beam crossing, and serially reads out the digitized data. The chip is designed only for positive charge (hole) collection from p-on-n detectors. The strips used by the experiment will have different lengths, resulting in a varying distribution of input capacitance across different chips. For the most part, activity is rare, but when there is an event, it is expected that there will be an average of 2.8% hit pixels per chip, with the possibility of much higher occupancy. Long latency cannot be tolerated, and it is therefore required that four hits will be read out within four beam cross-over periods of an event. As shown in Figure 1, a number of FPHX chips will be controlled on a single High Density Interconnect (HDI). Space on the HDI is limited, which is why the data from events must be serialized onto one or two pairs of LVDS digital lines per chip. In addition, the FPHX must be as “self-sufficient” as possible and provide its own internal bias voltages and currents with minimal external support circuitry. The user must be able to control internal parameters and biases, therefore a digital slow control interface is provided on each chip to enable programming.

FPHX is designed to be able to handle up to 4 hits/chip/collision. However, the observed hit multiplicity distribution demonstrates more than 4 hits from the given collision. Same observed in Tohoku beam data as well.

* Row *	pid *	module *	chip_id *	chan_id *	adc *	bco *	bco_full *
189 *	3001 *	13 *	7 *	90 *	1 *	4 *	1062123003935 *
190 *	3001 *	13 *	5 *	52 *	5 *	4 *	1062123003935 *
191 *	3001 *	13 *	17 *	19 *	5 *	4 *	1062123003935 *
192 *	3001 *	13 *	17 *	20 *	0 *	4 *	1062123003935 *
193 *	3001 *	13 *	2 *	109 *	7 *	4 *	1062123003935 *
194 *	3001 *	13 *	2 *	110 *	7 *	4 *	1062123003935 *
195 *	3001 *	13 *	5 *	51 *	7 *	4 *	1062123003935 *
196 *	3001 *	13 *	17 *	21 *	1 *	4 *	1062123003935 *
197 *	3001 *	13 *	17 *	22 *	7 *	4 *	1062123003935 *
198 *	3001 *	13 *	2 *	111 *	7 *	4 *	1062123003935 *
199 *	3001 *	13 *	2 *	112 *	1 *	4 *	1062123003935 *



ACTION ITEM: Any clone hits are to be removed from the multiplicity counting.

FPHX Manual

Phase Block

A primary requirement of the FPhx architecture is that it be able to read out within four beam cross-over periods an event that contained four hit strips. In other words, regardless of activity level, long latency cannot be tolerated. Hits must be sensed, amplified, discriminated, captured, sorted, serialized and read out of the chip. Moreover, the requirements do not allow for dead time, so if an FPhx chip receives an event in beam cross-over period “N”, it must be able to deal with an event in beam cross over period “N+1” and in beam cross-over period “N+2”, etc.

These twin requirements of low-latency and zero dead time give rise to the notion of phase architecture. During any given phase, amplification, discrimination, acquisition, sorting, serialization and output must happen. However, amplification, discrimination and acquisition are happening for hits that are occurring in *this* phase. Sorting is happening for hits that occurred in the last phase. Serialization and output are happening for hits that occurred at least two phases ago.

The requirement of “four-hits-in-four-beam-crossings” suggests that a cyclic progression of four phases (Phase 1 -> Phase 2 -> Phase 3 -> Phase 4 -> Phase 1 -> etc) could be used as the cornerstone of an architecture built to work for Phenix. During Phase 1, some circuitry would deal with amplification, discrimination and acquisition of hits that occur in Phase 1. Other circuitry would deal with the sorting of hits acquired in Phase 4. Still other circuitry would deal with the serialization and output of hits that were acquired in Phase 3 and earlier. Similarly, during Phase 2, some circuitry would deal with amplification, discrimination and acquisition of hits that occur in Phase 2. Other circuitry would deal with the sorting of hits acquired in Phase 1. Still other circuitry would deal with the serialization and output of hits that were acquired in Phase 4 and earlier.

This approach would require redundant circuitry. For example, there would be Phase 1 acquisition circuitry and Phase 2 acquisition circuitry and Phase 3 acquisition circuitry and Phase 4 acquisition circuitry. Moreover, there would have to be additional circuitry to manage the flow of data through these different phases. However, this approach would enable the job to be done without requiring excessive speed and the consequent power that would require.

One problem with this approach, however, is how to make it deal with variable event sizes. For example, even though “four-hits-in-four-beam-crossings” is the measuring stick for Phenix, events of only one hit will be very common and events with more than four hits are actually desirable.

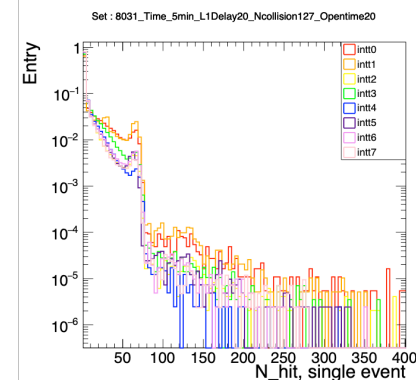
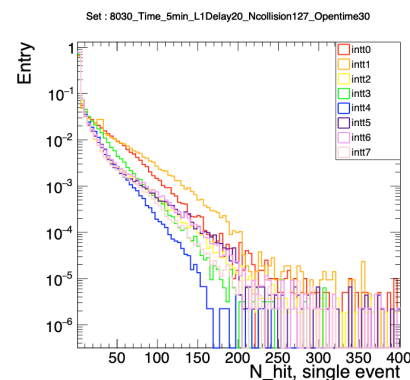
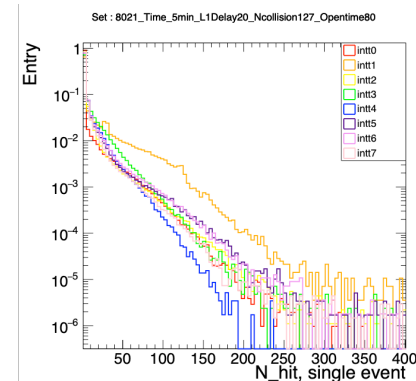
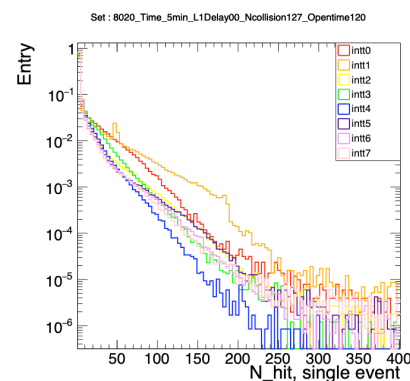
Fortunately, this architecture can deal with events of less-than four hits easily. The only consequence is a slight inefficiency in the readout bandwidth. However, since synchronization words will be output whenever there is no data to be output, this slight inefficiency should allow the data acquisition system to remain in sync with the FPhx chips.

Larger hits pose a bigger problem. Of necessity, amplification and discrimination must occur in parallel for all 128 channels. Therefore, larger events will have no impact on this activity. Acquisition, even though it has four phases, also occurs

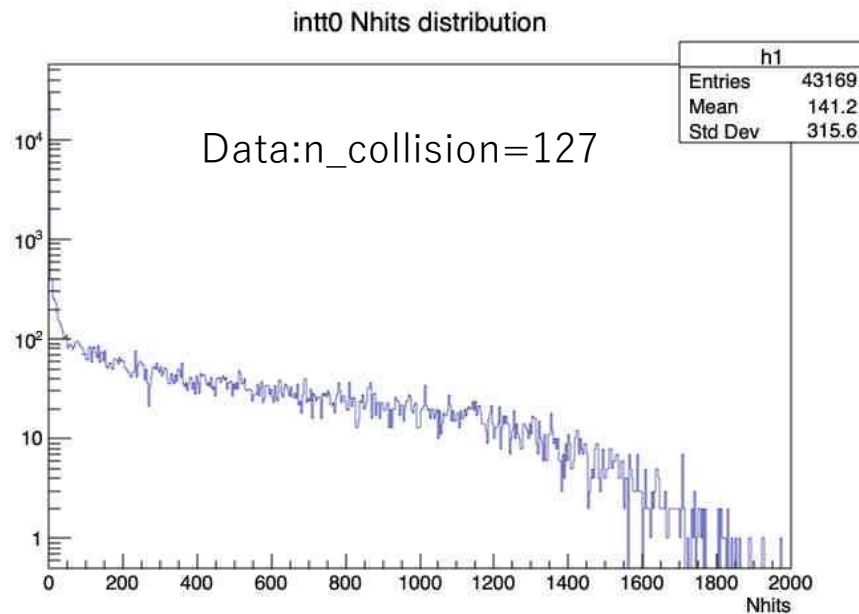
Felix Dependent Multiplicity (Cheng-Wei)

- The hit multiplicity suppose to be felix independent.
- The possible dependence should be originated from dead/hot/masked channel variation between different felix.

ACTION ITEM: Once time-in is completed, to be tested by taking data with constant rate DAC0 threshold/chip to verify above theory.

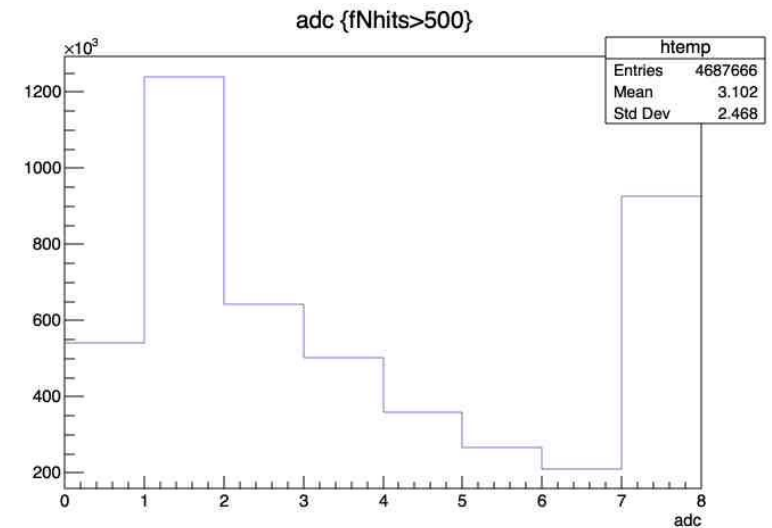
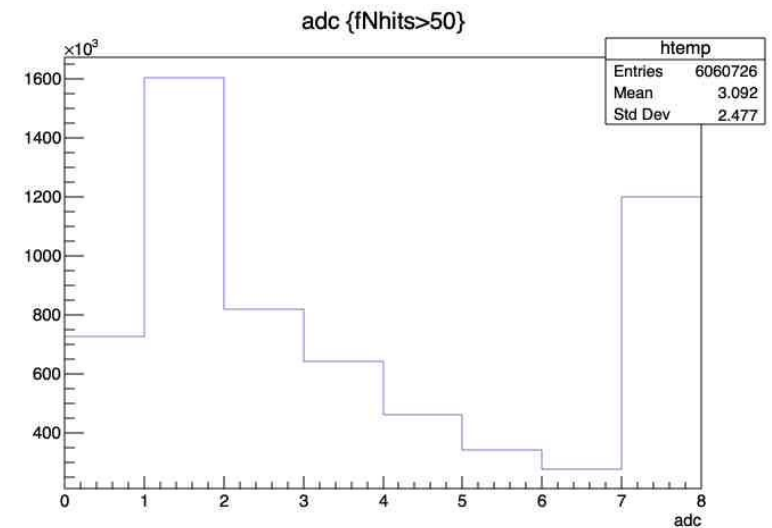


Multiplicity Dependent ADC distribution (Maya)



Hoping high multiplicity events are associated with the trigger though, the observed ADC distribution is yet not distinctive enough for the high multiplicity events.

ACTION ITEM: Compare ADC distributions Nhits>50 and Nhits<50.



Correlation between BCO-Full and FPHX BCO

- Has anyone taken a look the difference between FPHX BCO and the last 7 bits of BCO Full?
- **Joseph** should give an instruction how to access the last 7 digits of full BCO in the root tree, if some special function has already implemented.