G2p Helicity Setup

Chao Gu

Outline

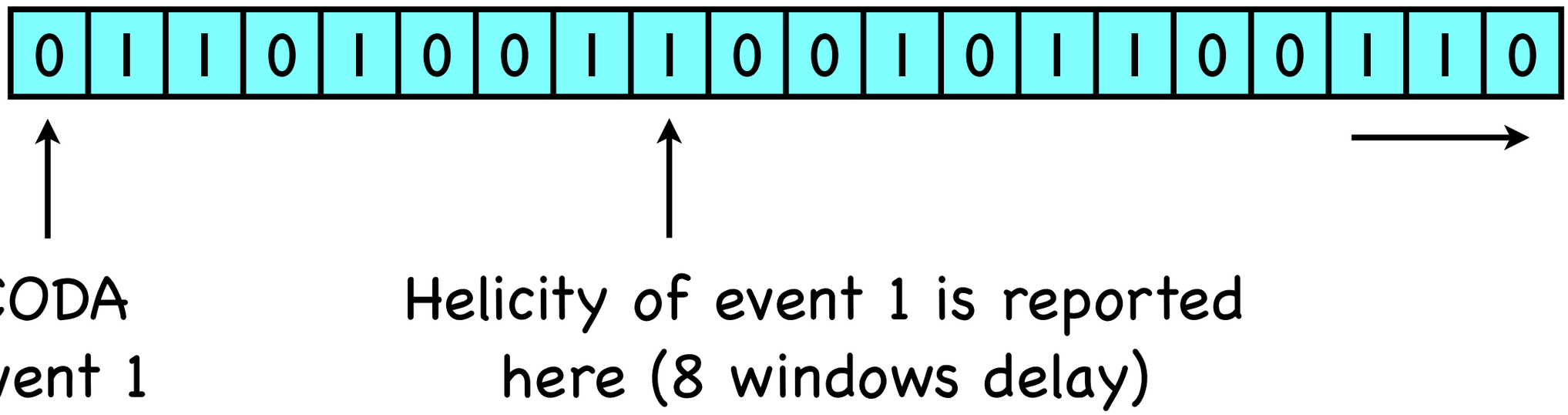
- Helicity Signal Scheme
- Hardware Setup
- Decode Package
- Charge Asymmetry Test
- Appendix

Helicity Signal Scheme

Back to Outline

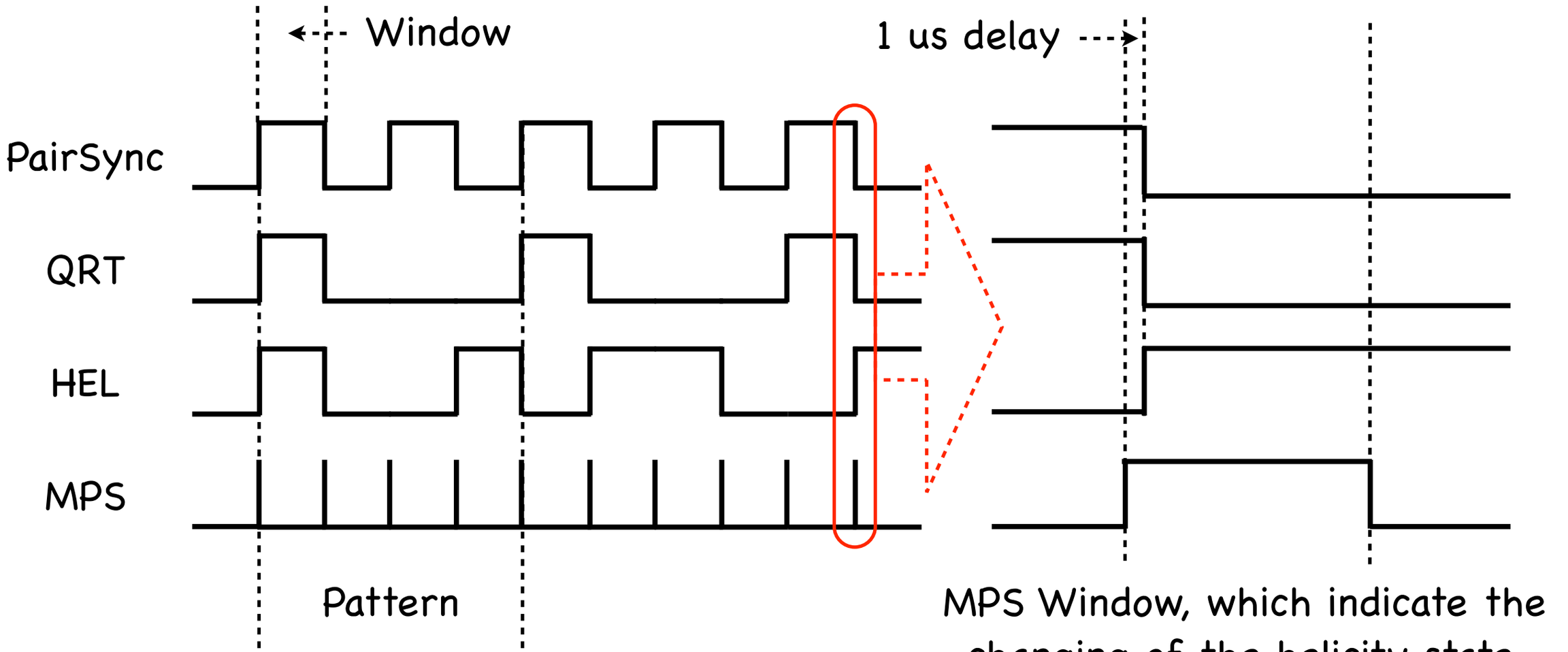
QWEAK Helicity Setting

- QWEAK runs with a helicity flip rate 960.02 Hz
- g2p registers CODA events at a rate up to 6 kHz
- The helicity reported from the helicity board is not the actual helicity of the beam at that time: delayed by 8 windows



Helicity Signal Scheme

- Four helicity signal: Helicity, QRT, MPS & PairSync
- The relationship of these 4 signals are shown in this plot:

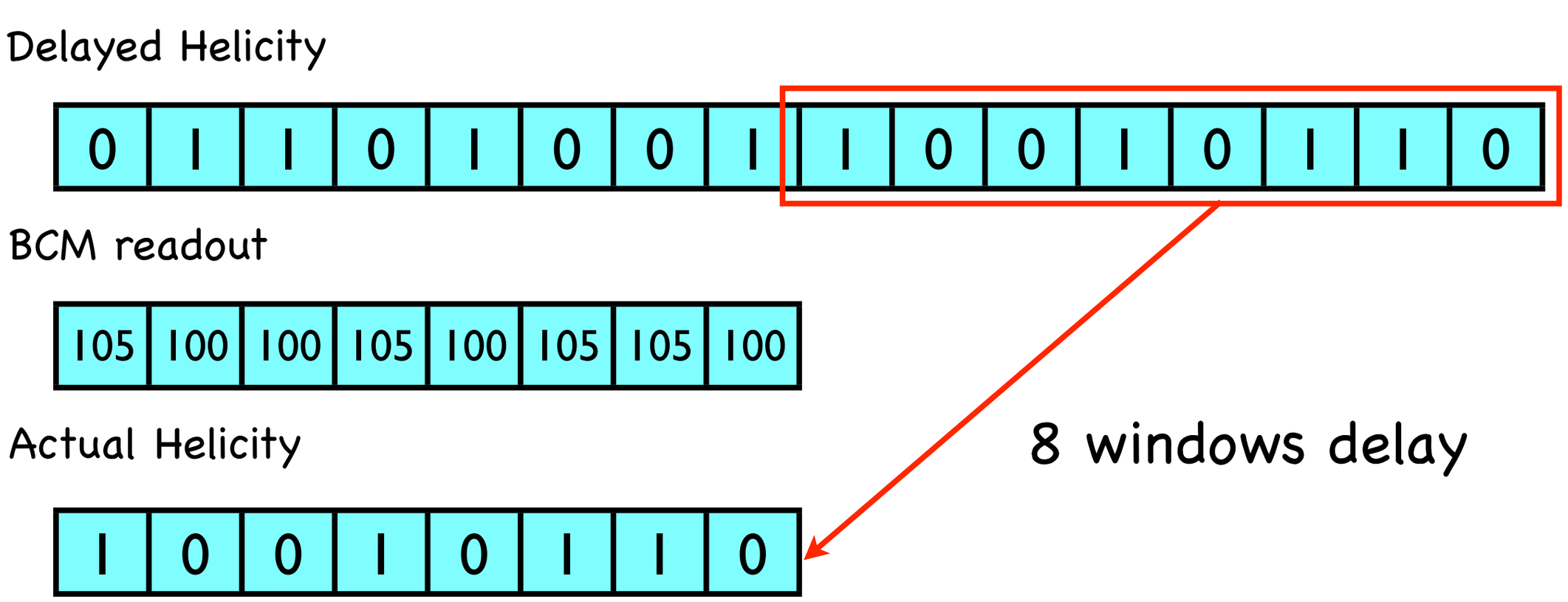


Helicity Signal Scheme: Definitions

- Helicity: indicate the polarization of the electron beam
- Window: a time region in which the helicity is stable, typically 971.65 us for QWEAK setting
- Pattern: to minimize system error, helicity is always changing by some symmetric sequence (like quartet +--+, octet +--+-++-, ...), which is called pattern, the pattern is composed by several helicity windows
- QRT: indicate the first window of a helicity pattern
- Polarity: the sign (positive or negative) of the first helicity window in each pattern
- MPS: indicate the changing time of the helicity state, in a MPS window, the helicity state is not well defined

Helicity Signal Scheme

- Readout data change with actual helicity (like BCM, Trigger, L1A, ..., will take BCM as example)
- Readout data is stored with delayed helicity
- Need to regenerate the actual helicity from the delayed helicity

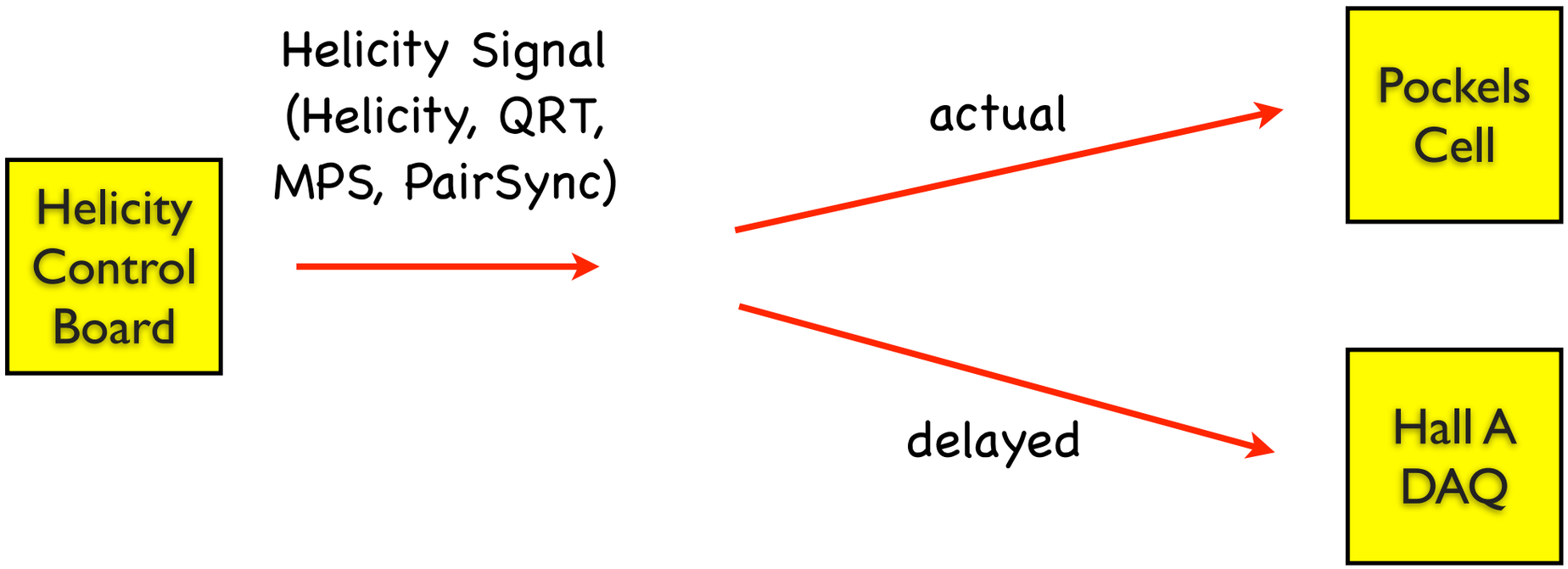


Hardware Setup

Back to Outline

Hardware Setup

- Helicity Signal is generated by Helicity Control Board installed at injector building
- Helicity board send actual helicity to injector to flip the real polarization of the electron beam
- Helicity board send delayed helicity to experiment hall

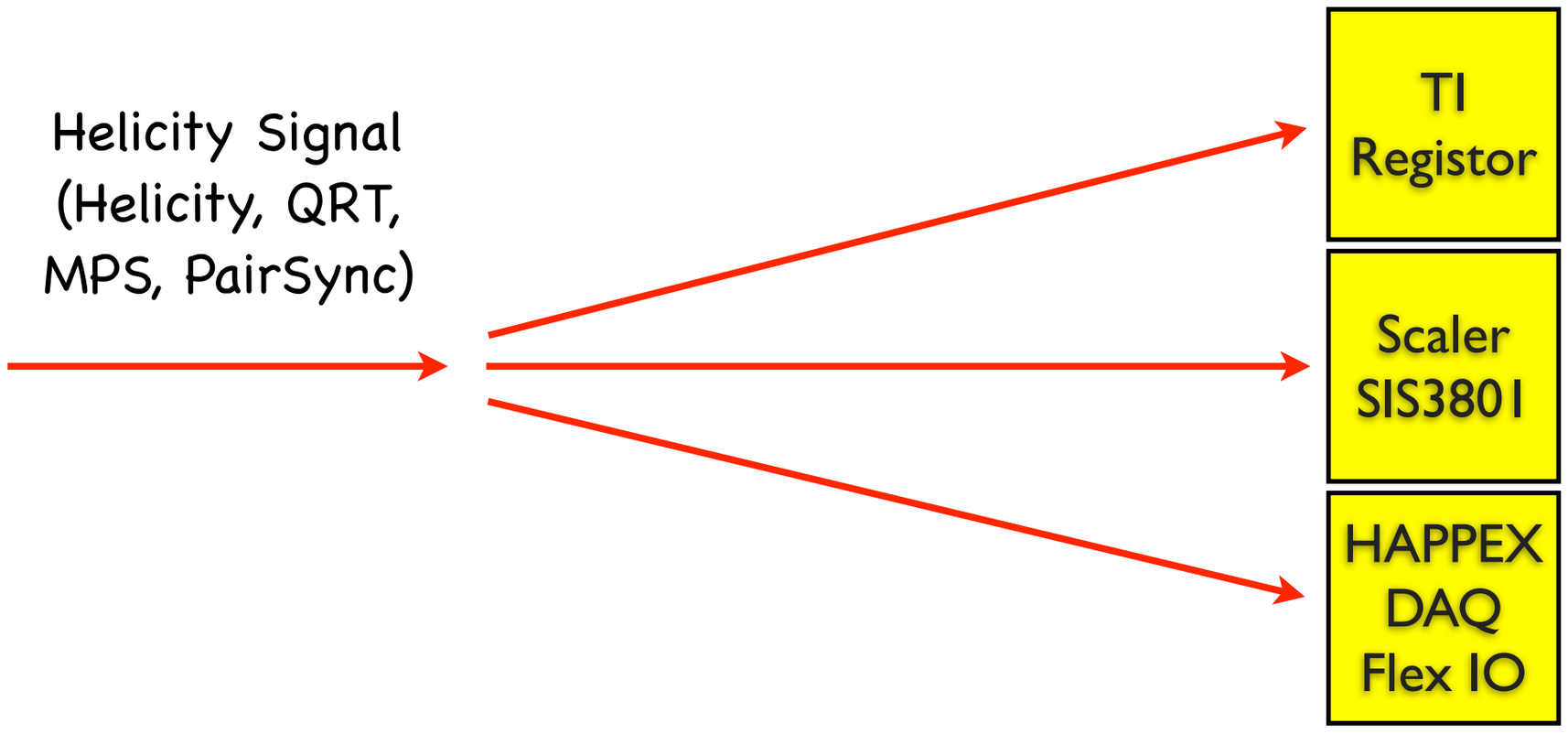


Hardware Setup

- Three ways to record helicity: Trigger Interface Register (TIR), Scaler Ring Buffer & HAPPEX DAQ
  - TIR: to record the helicity state for each physics event
  - Scaler: to record the charge information sorted by the helicity state
  - HAPPEX DAQ: same as Scaler but more accurate
- Electronics setup for TIR and Scaler: follow R. Michael's tech note
- HAPPEX Hardware Setup: I will skip this topic here, please see Pengjia's documents for details

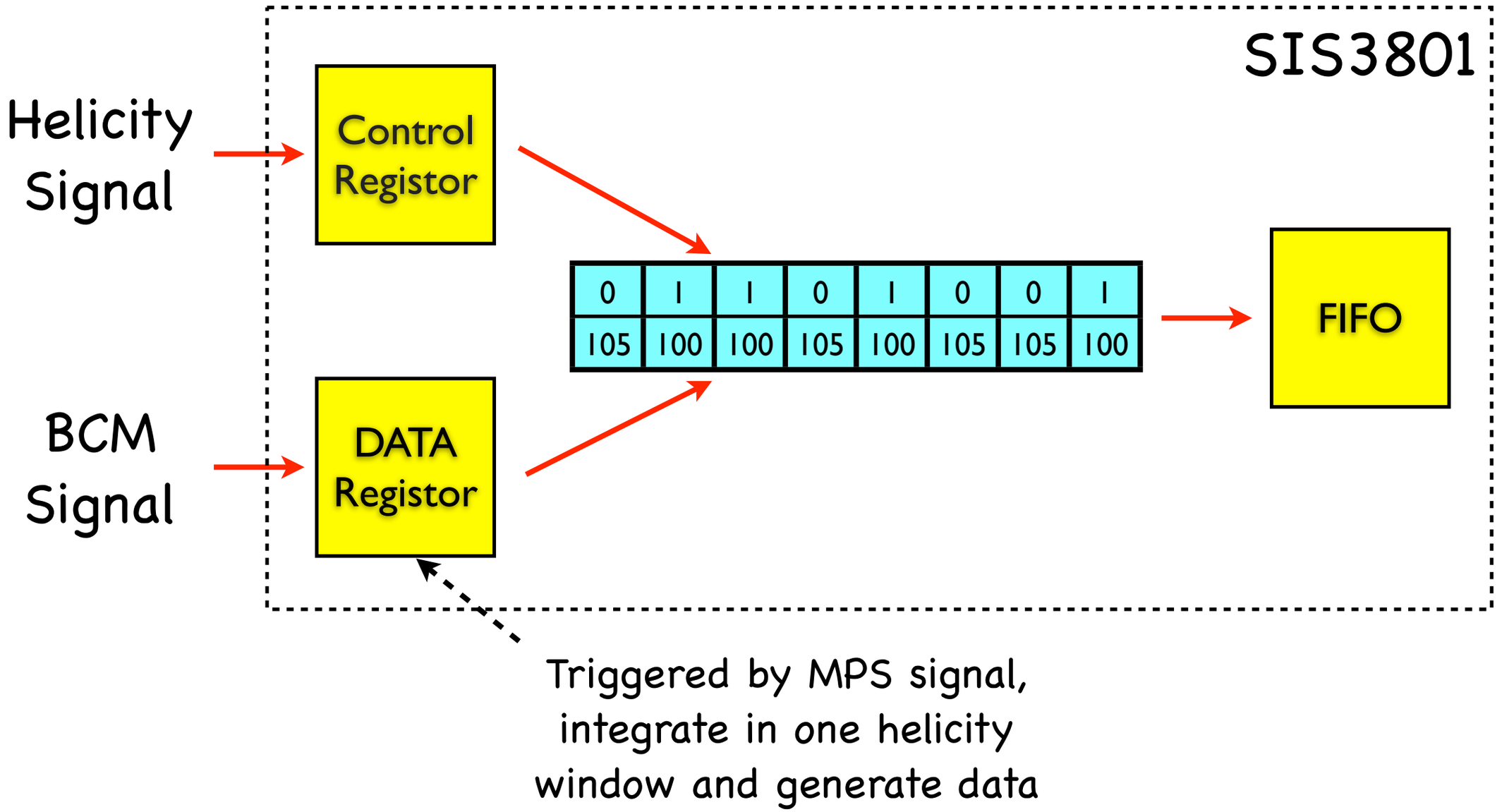
Hardware Setup: TIR

- 8 copies of helicity signal: 3 to Left, right and 3rd arm scaler SIS3801, 3 to L, R & 3rd arm TIR, 2 to L & R HAPPEX DAQ
- TIR only use 4 registors to record the helicity state when there is a trigger
- The helicity information of physics event can only be got from TIR

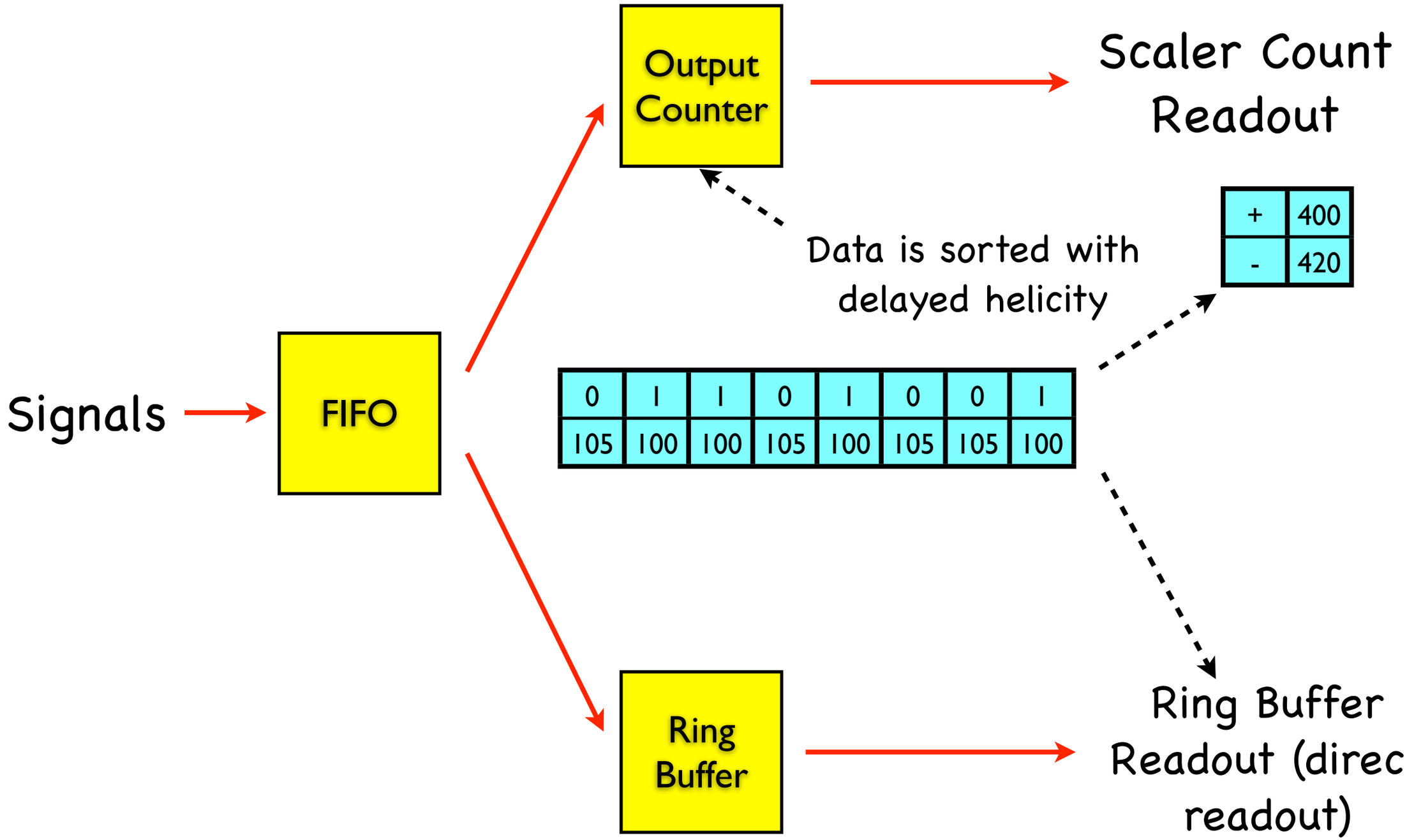


Hardware Setup: Scaler

- Electronics setup can be found here (by R. Michael) : http://hallaweb.jlab.org/equipment/daq/qweak_helicity.html (or in Appendix)

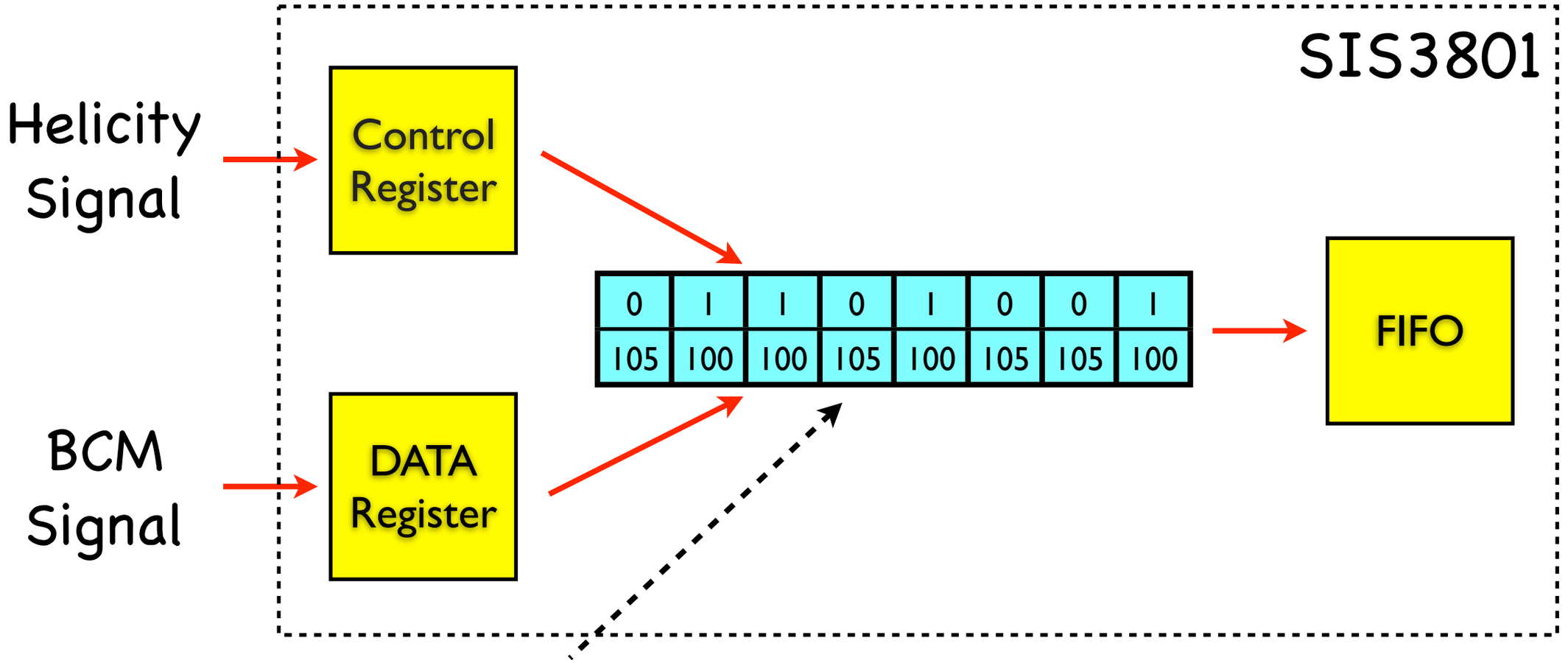


Hardware Setup: Scaler



Hardware Setup: Scaler

- One problem found during commission



The delayed helicity and BCM is not aligned inside SIS3801, it should be delayed by 8 windows but it as actually 7 windows

0	1	1	0	1	0	0	1
105	100	100	105	100	105	105	100

	0	1	1	0	1	0	0
105	100	100	105	100	105	105	100

Decode Package

Back to Outline

Decoder: Pseudo-random Helicity Generator

- The polarity of each helicity pattern is generated by a 30 bits shift registor, which is a pseudo-random sequence

Polarity sequence generated by the pseudo-random number generator

0
1
0
0
0
1
0
1
1

Each bit in this sequence will be expended with the helicity pattern

0	1	1	0
1	0	0	1
0	1	1	0
0	1	1	0
0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0
1	0	0	1
1	0	0	1

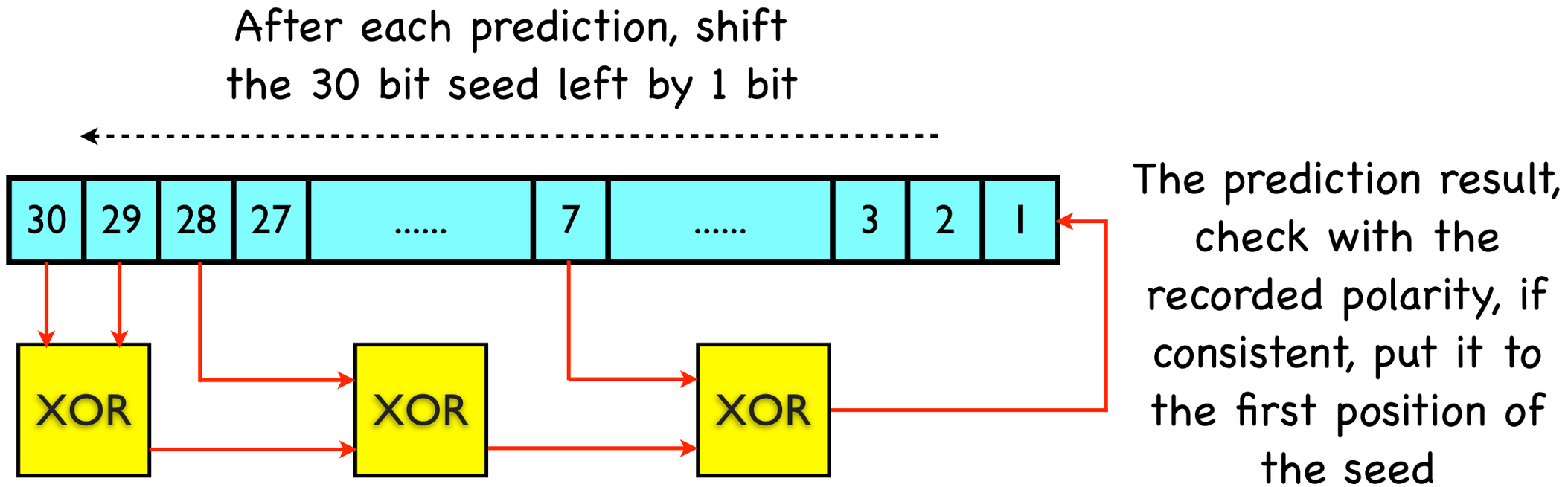
0	1	1	0
1	0	0	1
0	1	1	0
0	1	1	0
0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0
1	0	0	1
1	0	0	1

0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	1

Connect all of the helicity patterns to generate the final helicity flipping sequence

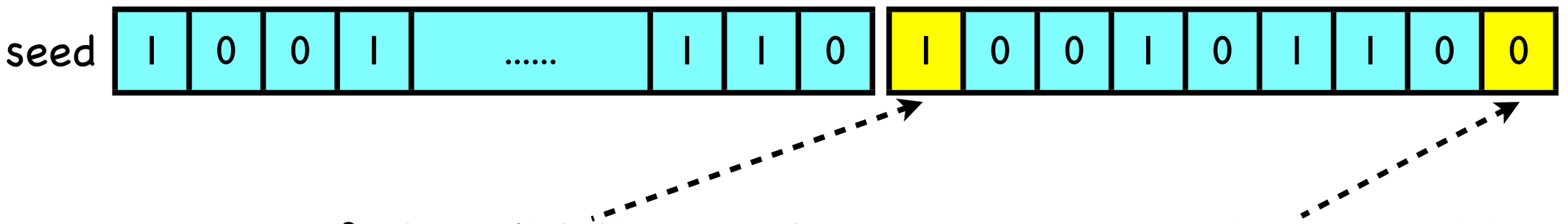
Decoder: Prediction

- If we knew the polarity of continuous 30 helicity pattern, we will be able to predict the polarity of following helicity pattern using the pseudo-random number generator
- The plot shows the algorithm to predict polarity of helicity pattern



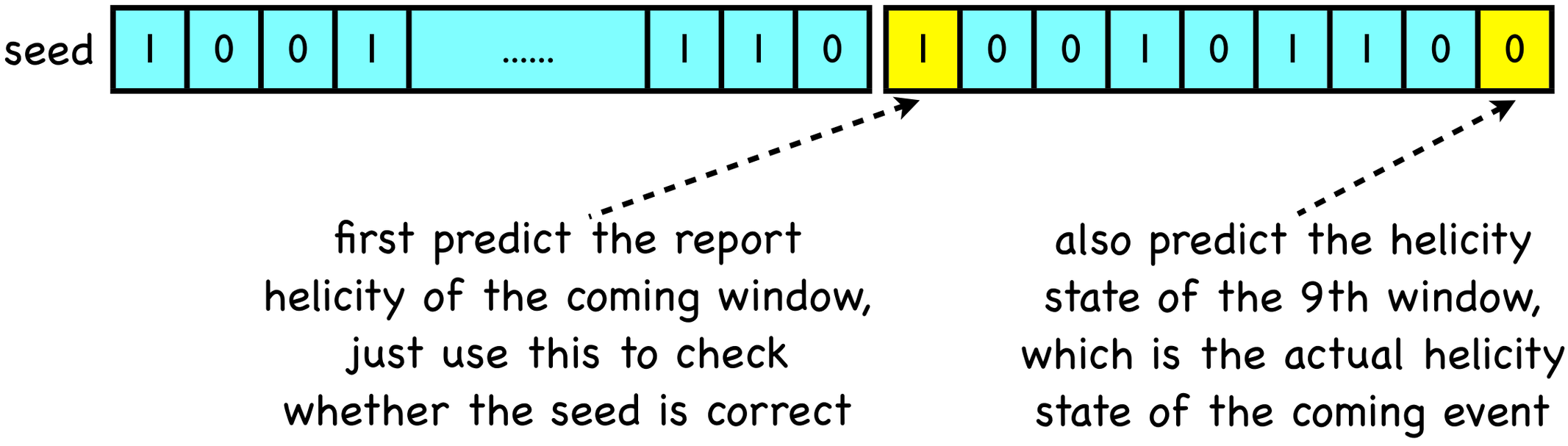
Decoder: Prediction

- Since the report helicity is delayed by eight windows, the basic idea of the decoder is to gather 30 helicity patterns to generate the pseudo-random seed, then use this seed to predict the actual helicity
- Will do 2 times of prediction for each helicity window after it got the random seed:



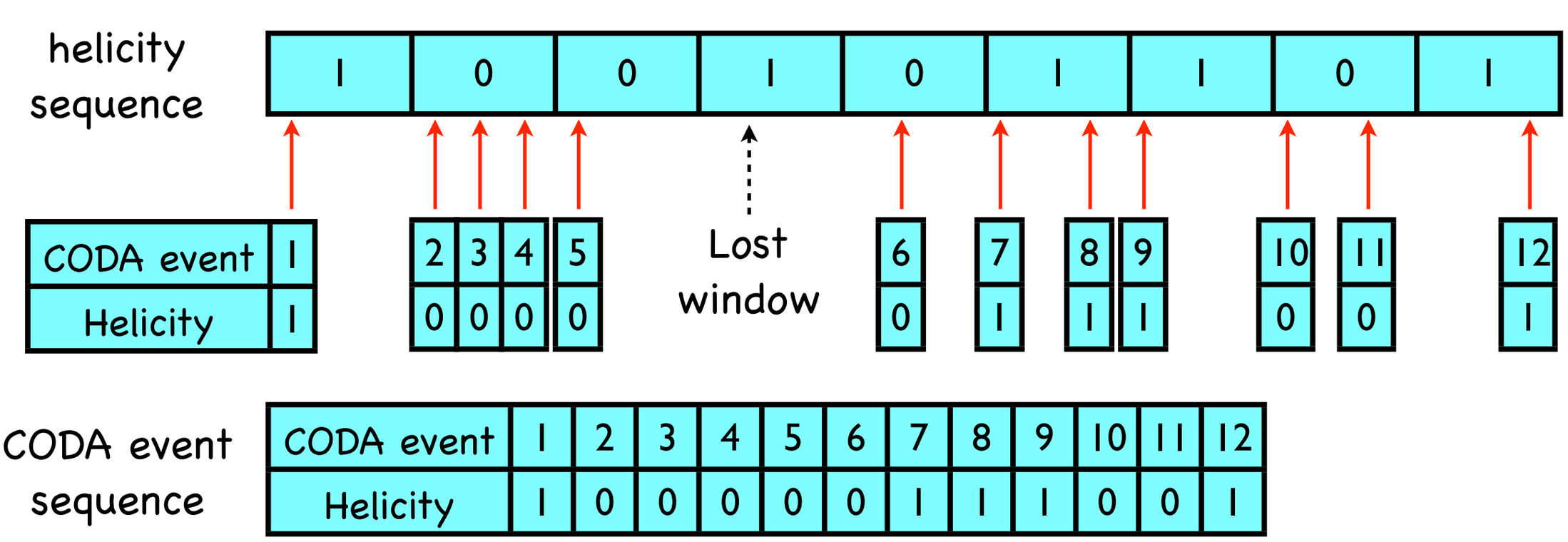
Decoder: Ring Buffer

- Since the scaler is triggered by MPS signal, it will only record one event for each helicity window, and will not lose any helicity window if normal
- Helicity information in scaler ring buffer automatically satisfies the requirement to do prediction (complete information, no repeat)
- No special treatment is required here, just process the prediction as described in previous slide



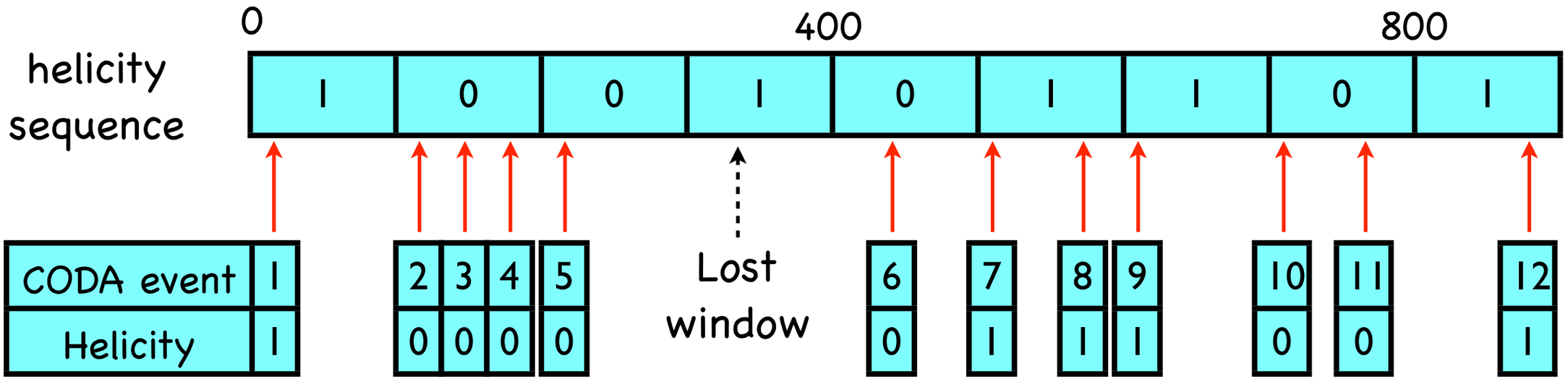
Decoder: TIR

- TIR uses physics trigger, it may record several events for each helicity window, and may lose some helicity information when there is no physics trigger as shown in the plot
- Direct prediction will FAIL



Decoder: TIR

- Solution: Time Stamp
- Use time stamp to decide if there are multiple CODA events in one helicity window or if there are lost windows



CODA event sequence

CODA event	1	2	3	4	5	6	7	8	9	10	11	12
Helicity	1	0	0	0	0	0	1	1	1	0	0	1
Timestamp	21	109	147	185	216	432	508	582	605	707	764	879

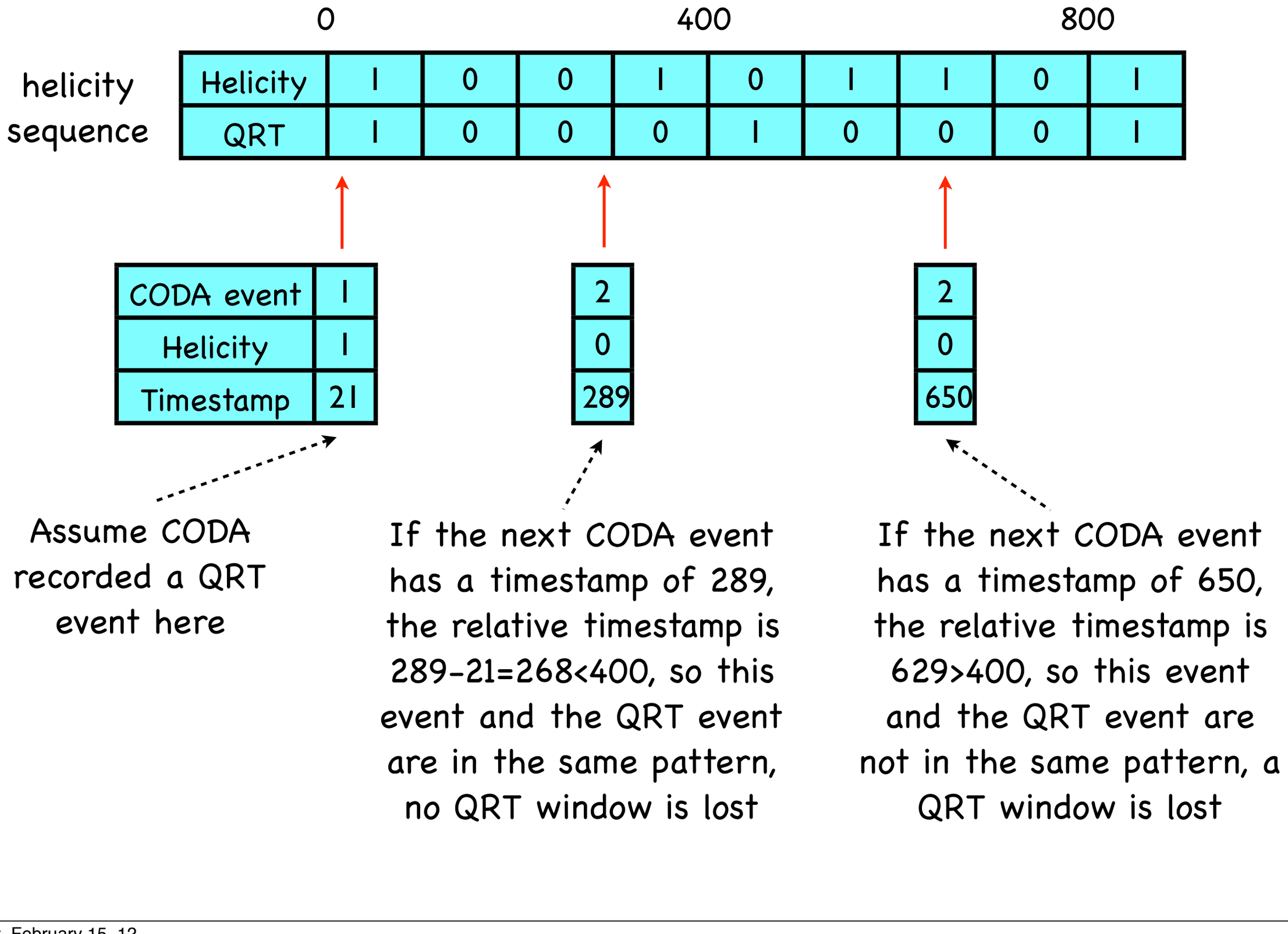
First Pattern | Second Pattern

0 400 800

Decoder: TIR

- Under real situation, there is no absolute time stamp for the helicity sequence
- The first helicity window in each helicity pattern is the most important for prediction because it decides the polarity
- From now on, we will call a CODA event in the first helicity window of one pattern a "QRT event", and call this window a "QRT window"
- Assume there is already one QRT event, then we can use relative time stamp to decide whether there is a lost QRT window

Decoder: TIR


0
400
800
helicity sequence
Helicity 1 0 0 1 0 1 1 0 1
QRT 1 0 0 0 1 0 0 0 1
CODA event 1
Helicity 1
Timestamp 21
2
0
289
2
0
650
Assume CODA recorded a QRT event here
If the next CODA event has a timestamp of 289, the relative timestamp is 289-21=268<400, so this event and the QRT event are in the same pattern, no QRT window is lost
If the next CODA event has a timestamp of 650, the relative timestamp is 629>400, so this event and the QRT event are not in the same pattern, a QRT window is lost

Decoder: TIR

- Use the method described in previous slide, we could decide if there is lost QRT windows
- Once we decide how many QRT windows is lost, we could apply some correction for the prediction to get the correct result
- To decide if there are multiple events inside one helicity window, one could compare the time stamp of 2 events (2 events with the same Helicity, QRT, PairSync value), if the difference is smaller than window time, then these 2 events belong to the same helicity window
- For helicity information form HAPPEX DAQ, since HAPPEX DAQ works as a ring buffer, I use the same method as decoding ring buffer helicity to decode it

Decoder

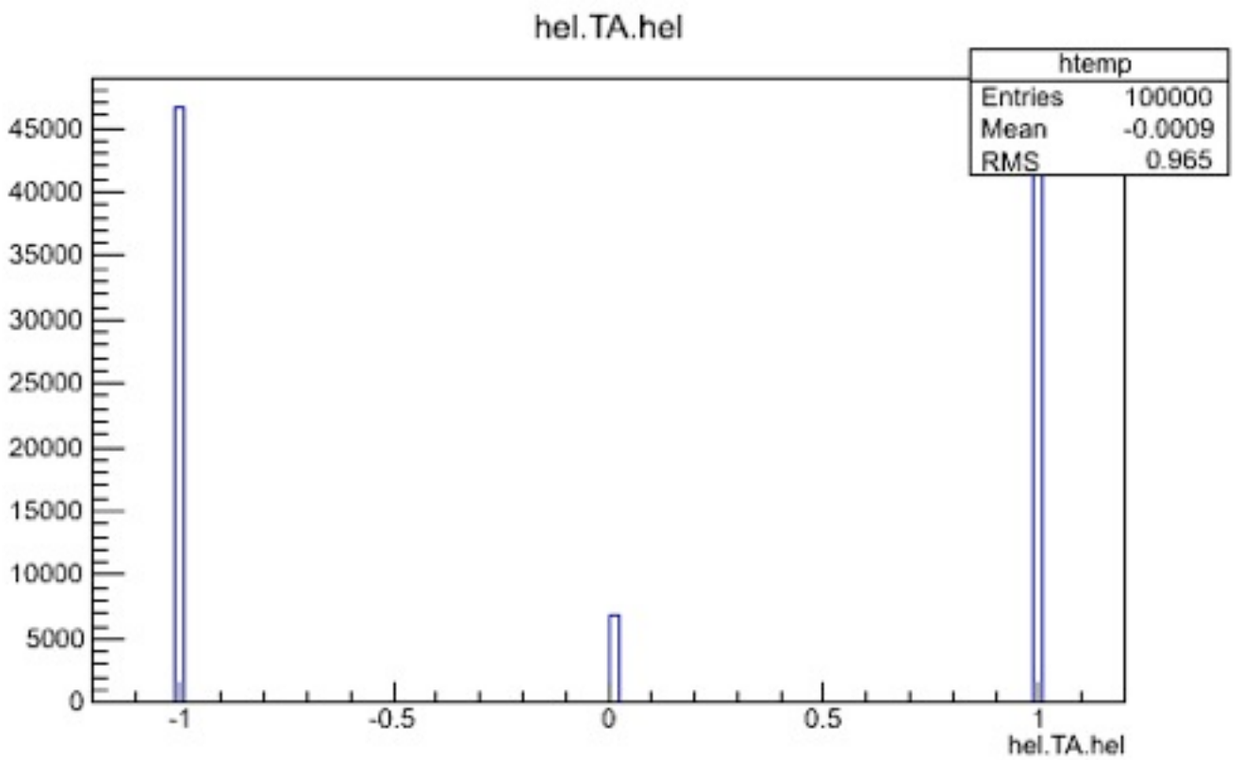
- Since we have 3 sources of helicity, the decoder package will compare the pseudo-random seeds of these 3 sources to see if there is any difference
- If we lost the helicity information from one source, we can use the other 2 sources to make a recovery of the seed of this source
- The new decoder package THaG2PHelicity follows the algorithm described above, see the comments in the source code to get more details

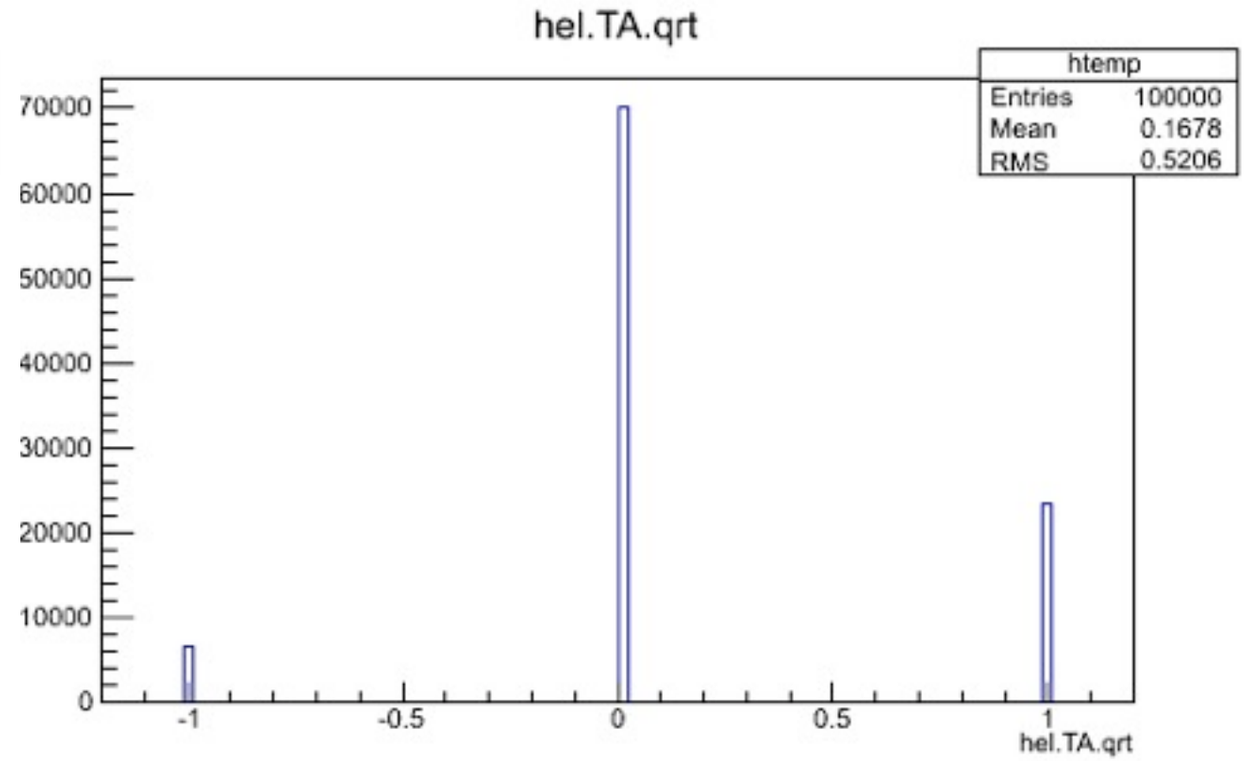
Decoder: Efficiency Test

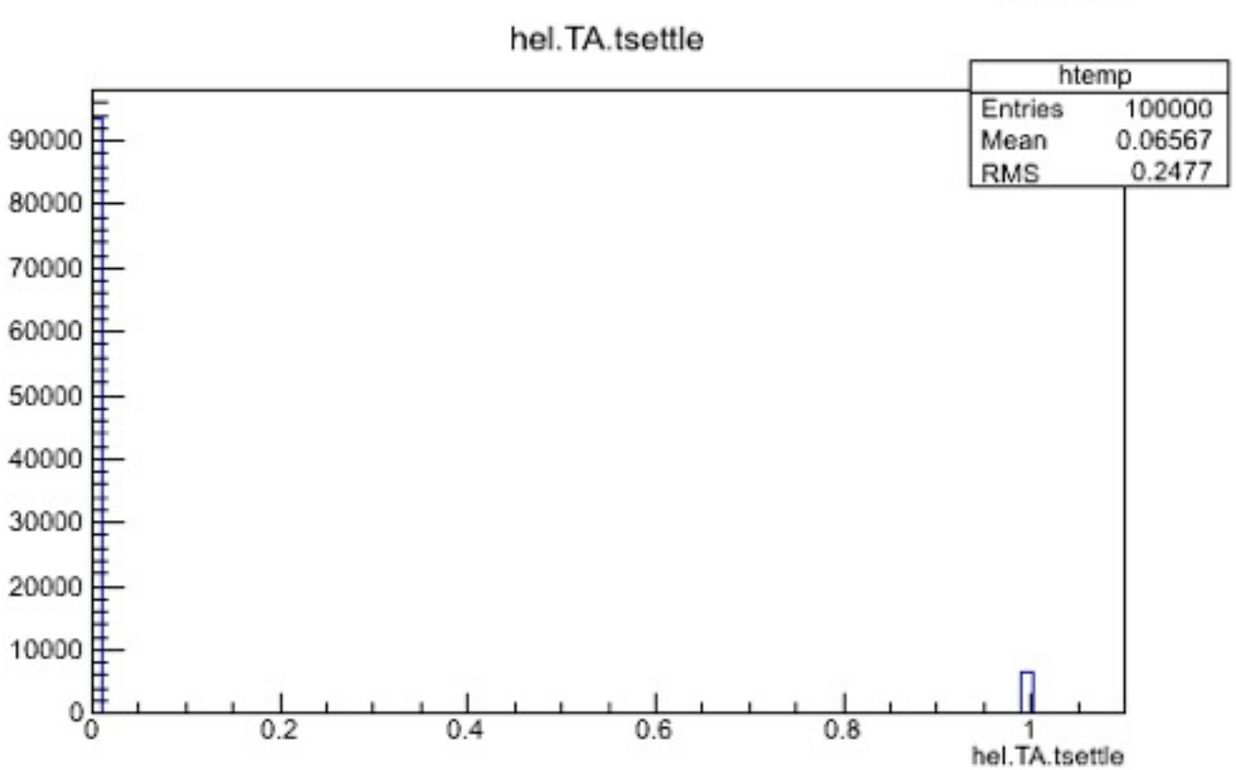
- Test the package with different event rate to see how many event will lose during decode
- The result shows that typically the lost event amount is 0.2-0.5 s times event rate

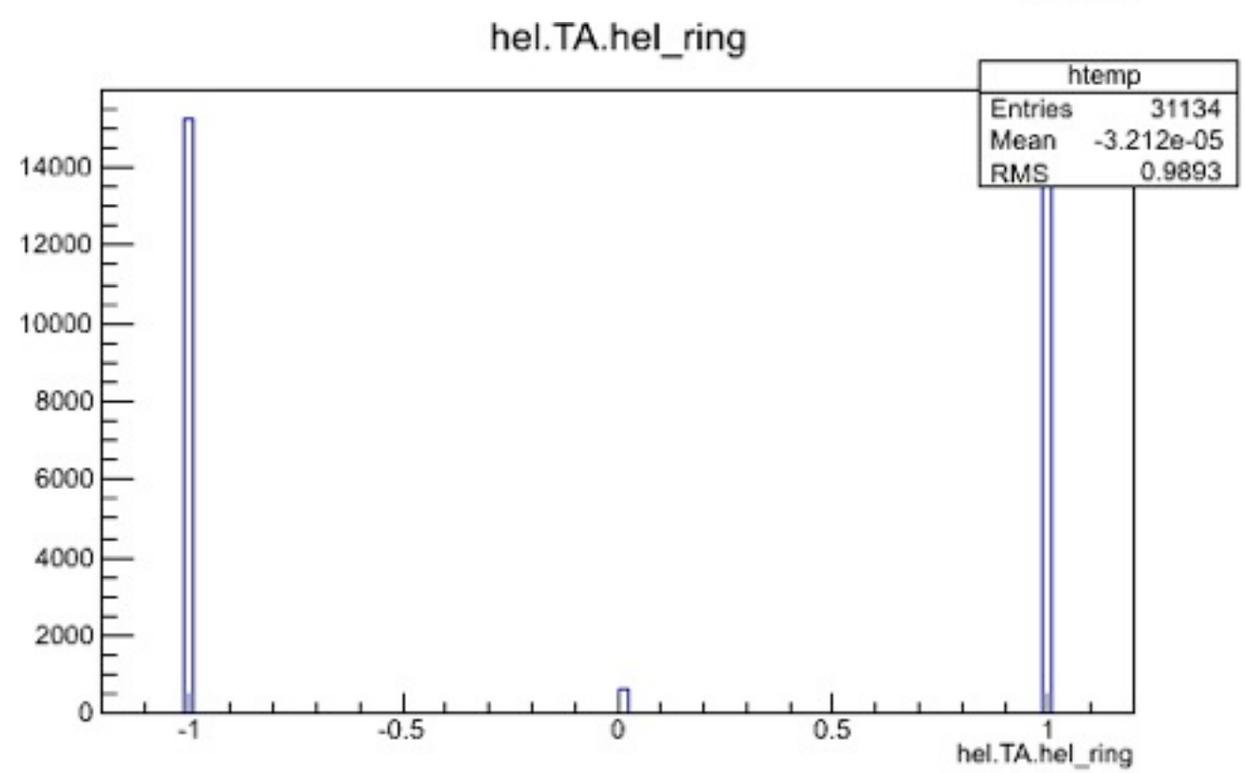
Event Rate/Hz	Event Amount	Event lost for MPS window	Event lost for decode
320	100k	6552	188
1k	100k	6714	153
2k	100k	6709	143
3k	100k	6567	307
5.5k	100k	6715	677

Decoder: Efficiency Test









Charge Asymmetry Test

Back to Outline

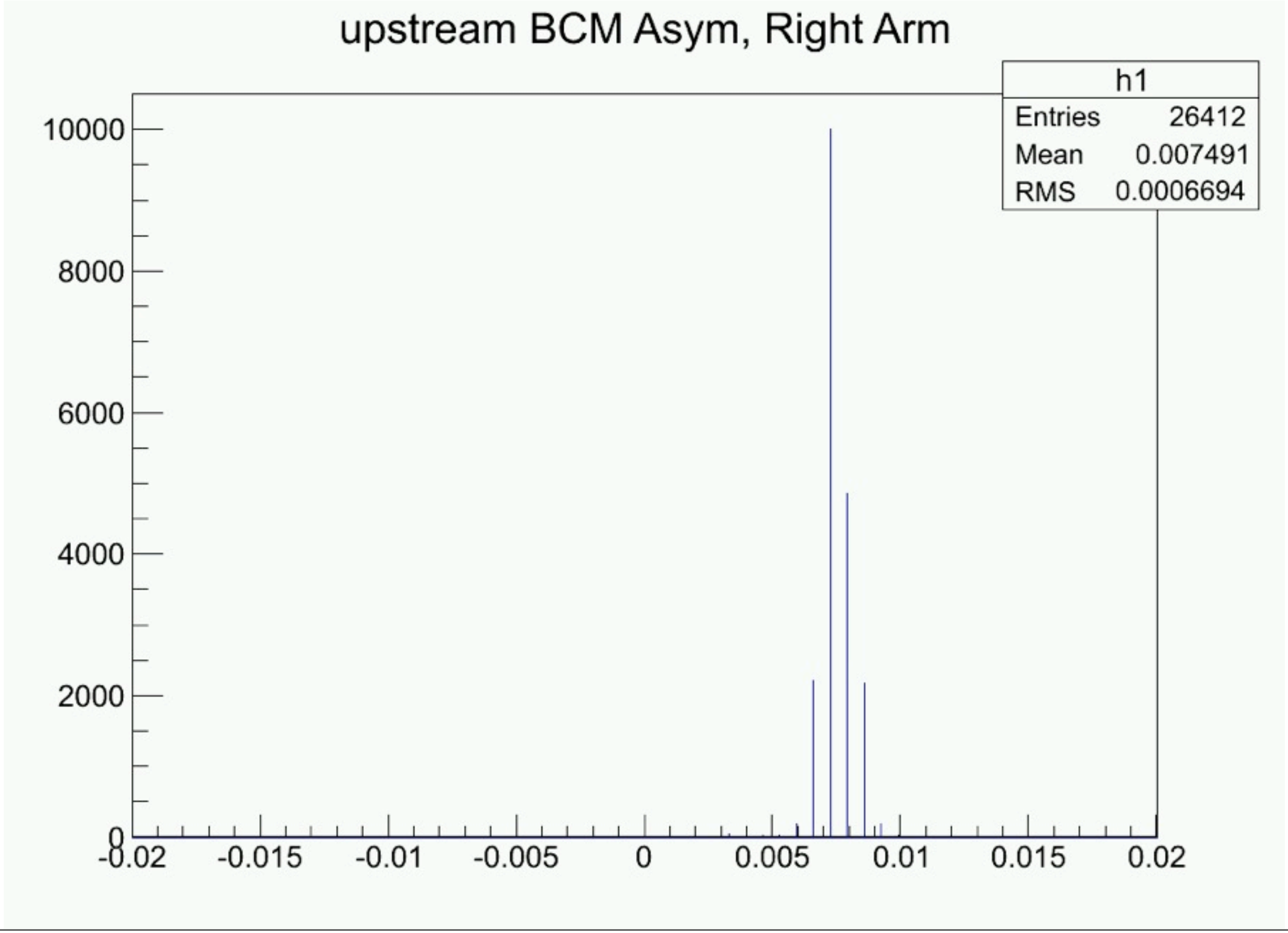
Test Setup

- Use a local helicity board to generate a fake helicity signal (delayed and no-delayed)
- Use DAC to generate a fake BCM signal which has a large asymmetry, this signal is based on actual helicity
- Put delayed helicity actual helicity, fake BCM signal into our normal DAQ system (HRS, HAPPEX, Moller, 3rd Arm...), and calculate the charge asymmetry
- See Pengjia's talk: http://hallaweb.jlab.org/experiment/g2p/collaborators/pzhu/02082012/happex_asym_report.pptx for more details about the test setup

Charge Asymmetry: Calculation

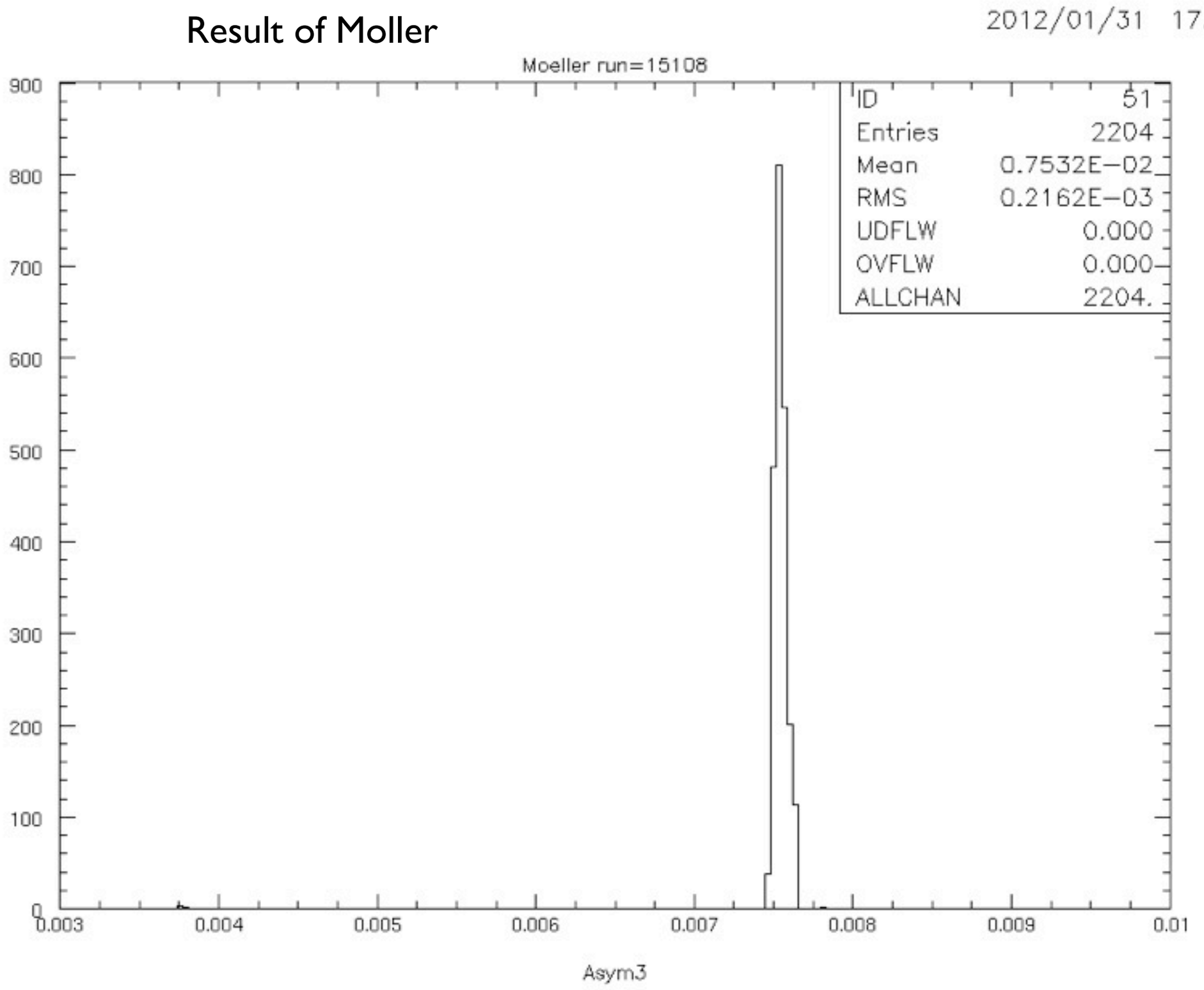
- Use a helicity pattern as one unit to calculate the charge asymmetry
- Plot the charge asymmetry of each helicity pattern in one histogram to get the distribution
- Also get the total charge in positive helicity state and negative helicity state for the whole run and calculate a charge asymmetry
- Compare the charge asymmetry got from these two methods, the value should match

Results, Asymmetry Set Point +0.75%



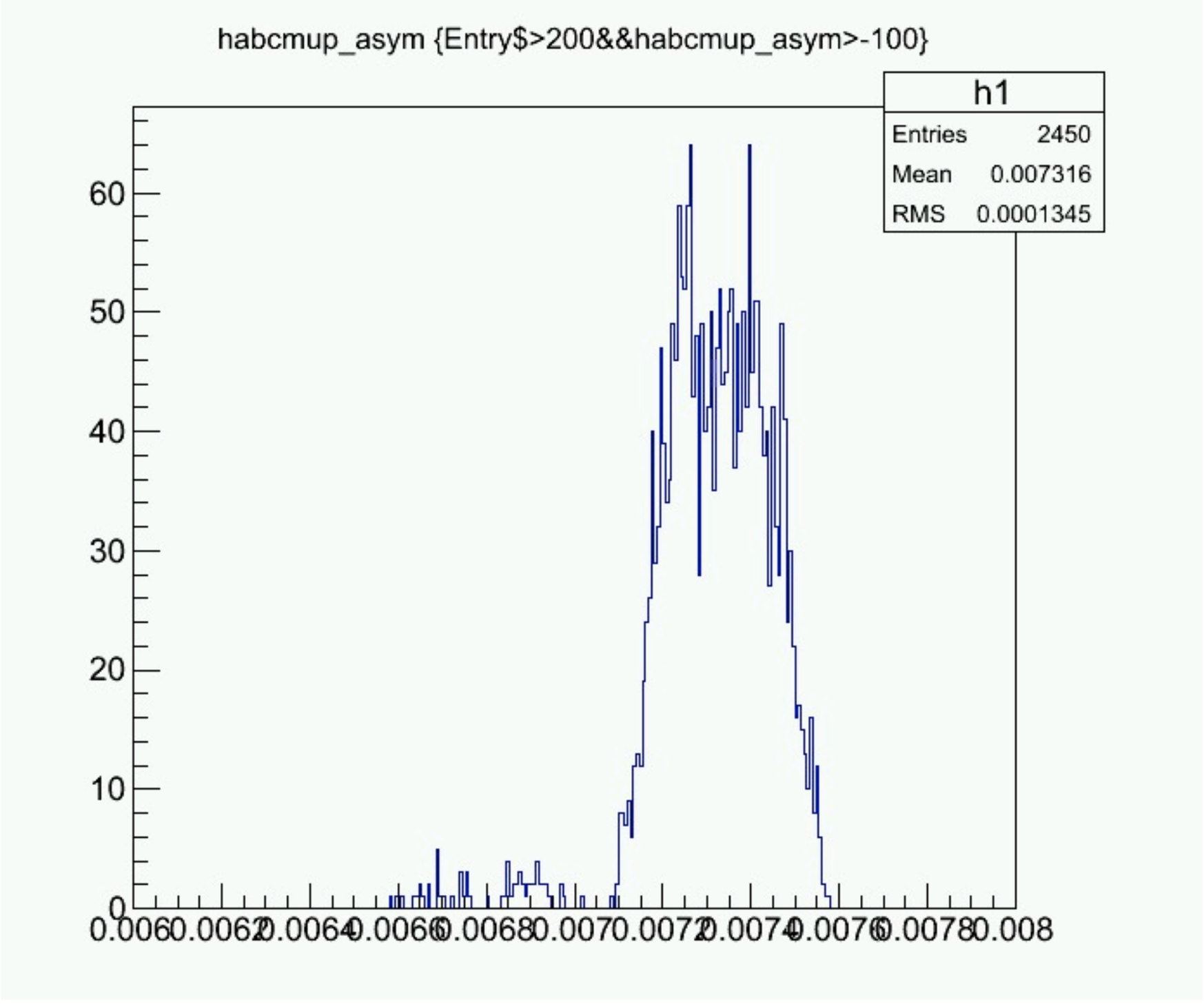
Results, Asymmetry Set Point +0.75%

Result of Moller

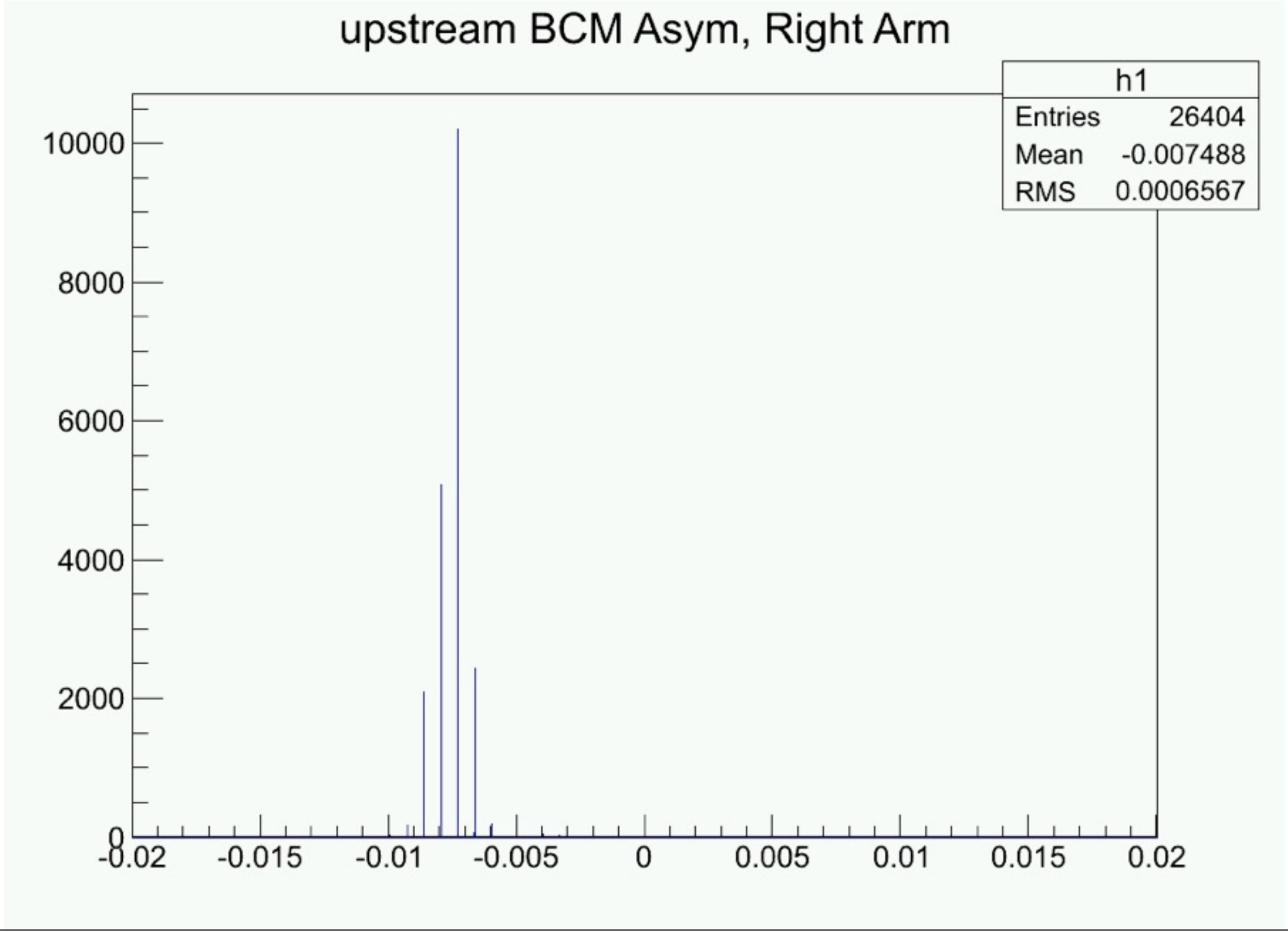


Results, Asymmetry Set Point +0.75%

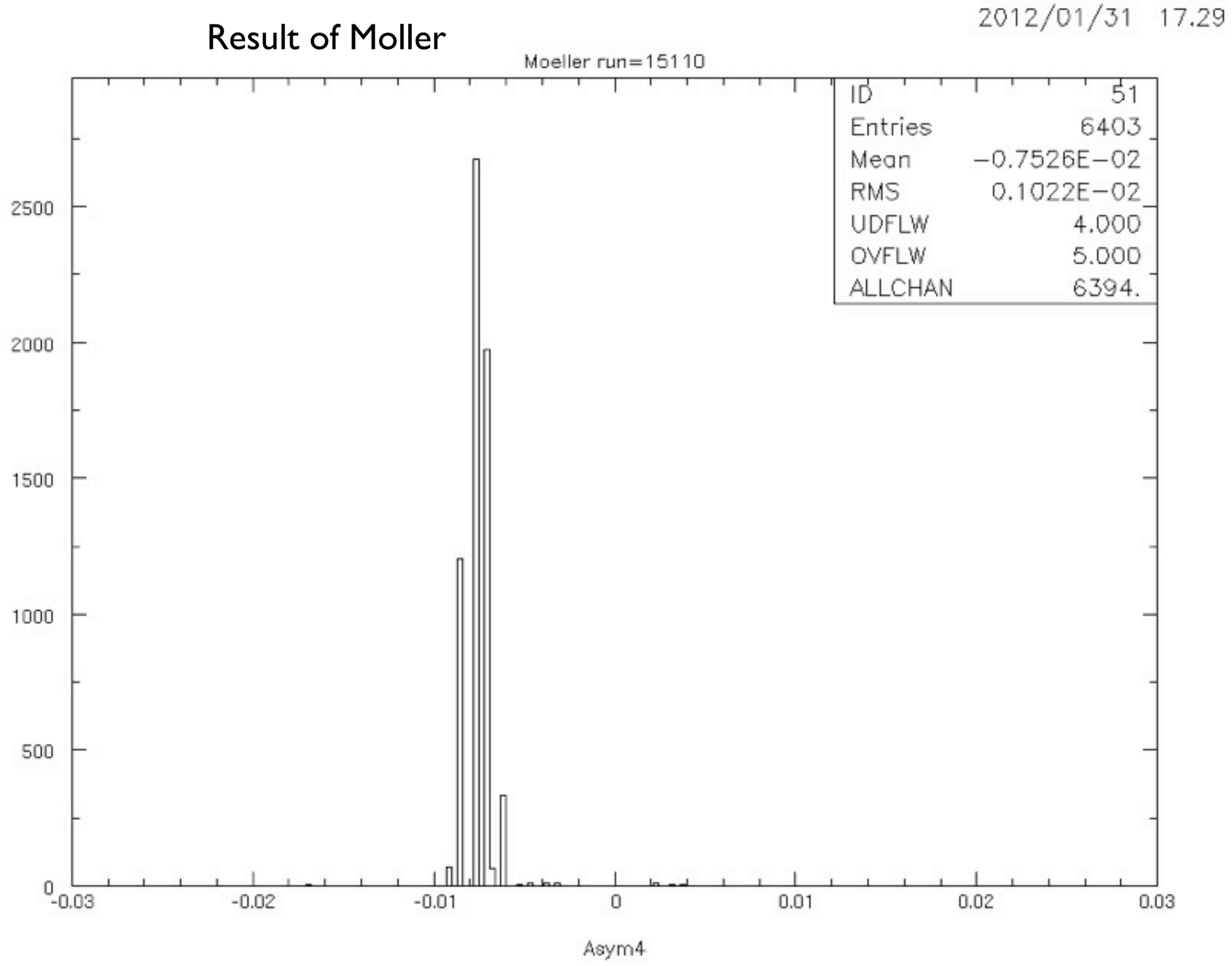
Result of HAPPEX DAQ, Right Arm



Results, Asymmetry Set Point -0.75%

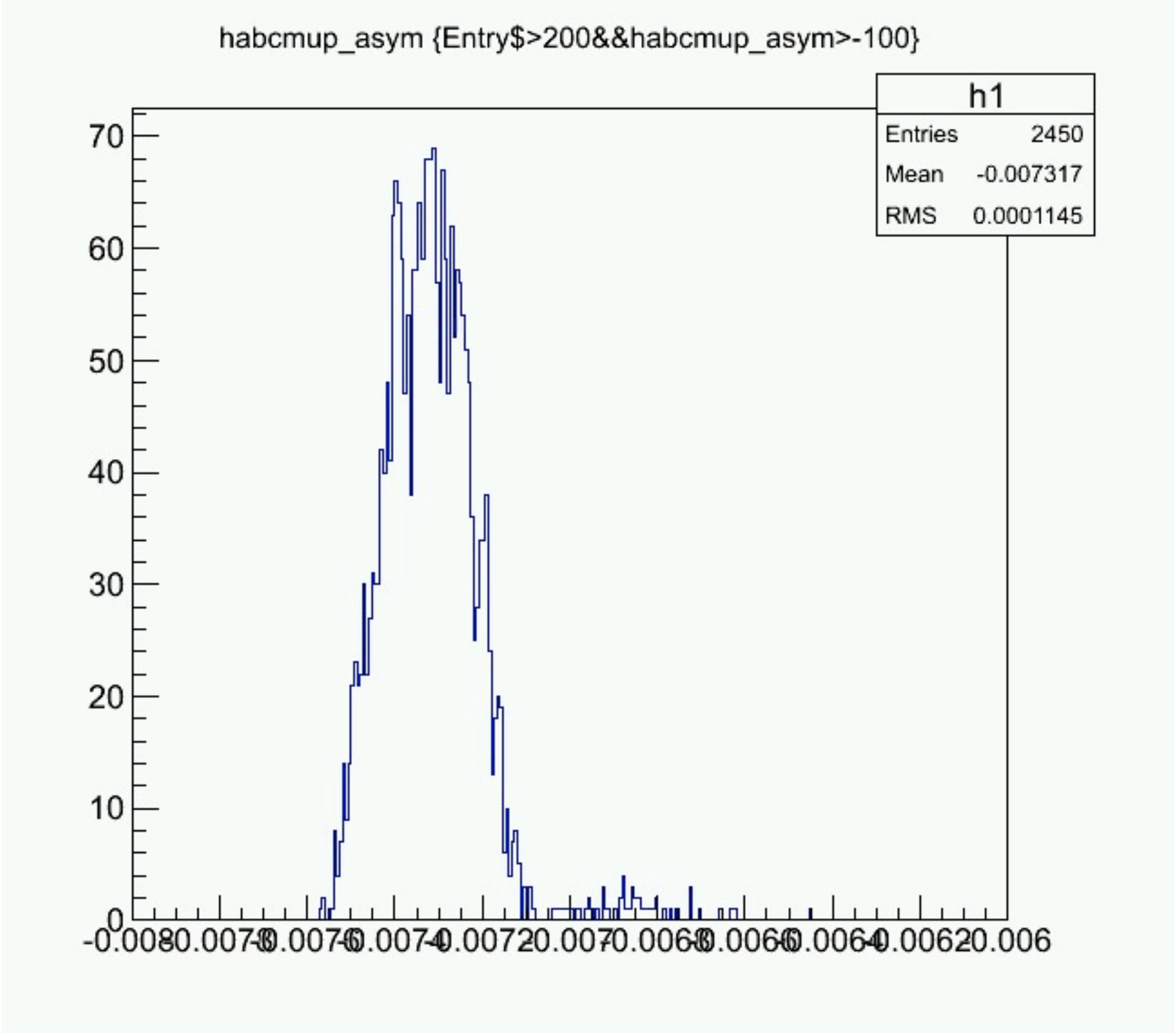


Results, Asymmetry Set Point -0.75%

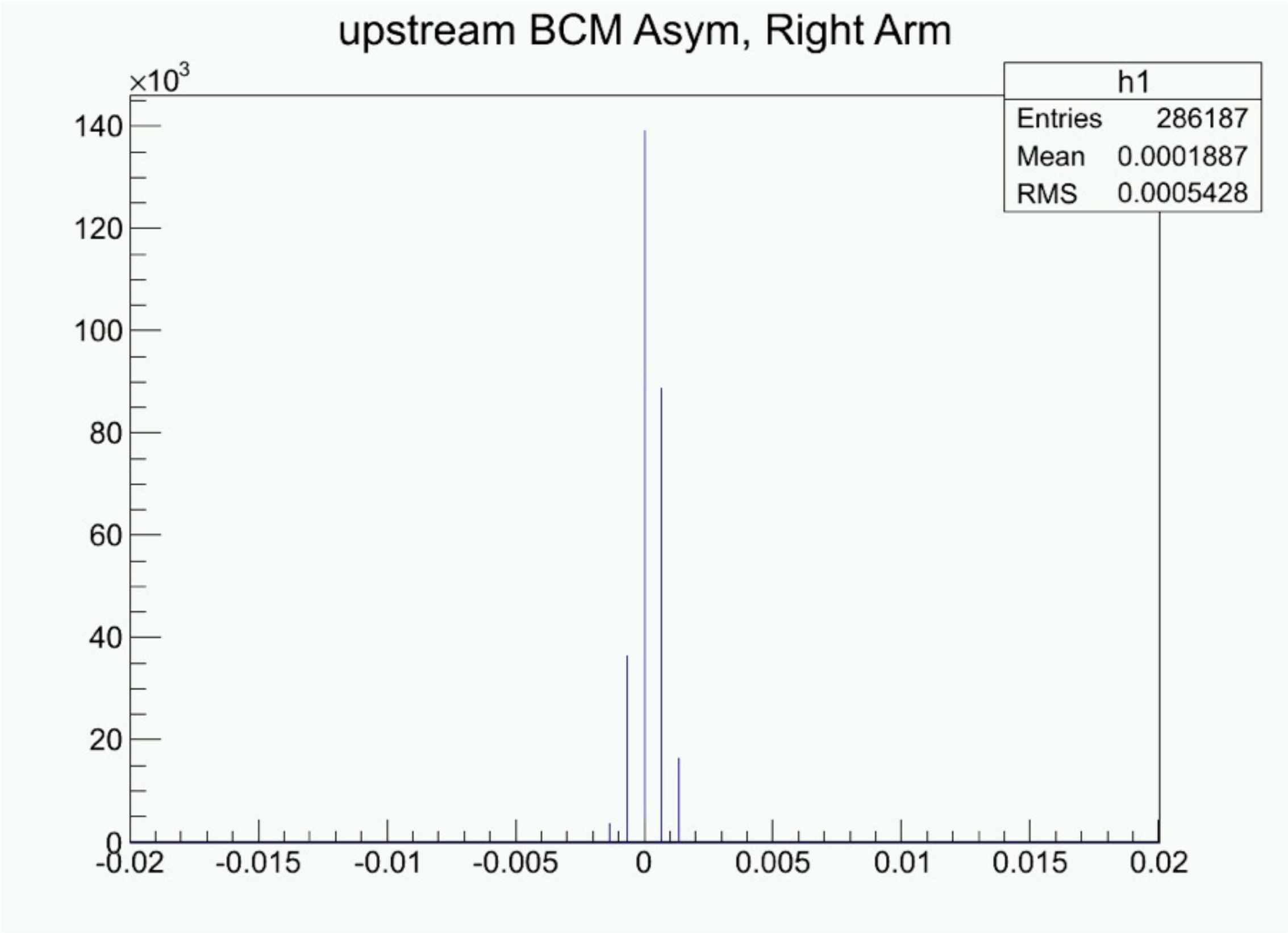


Results, Asymmetry Set Point -0.75%

Result of HAPPEX DAQ, Right Arm

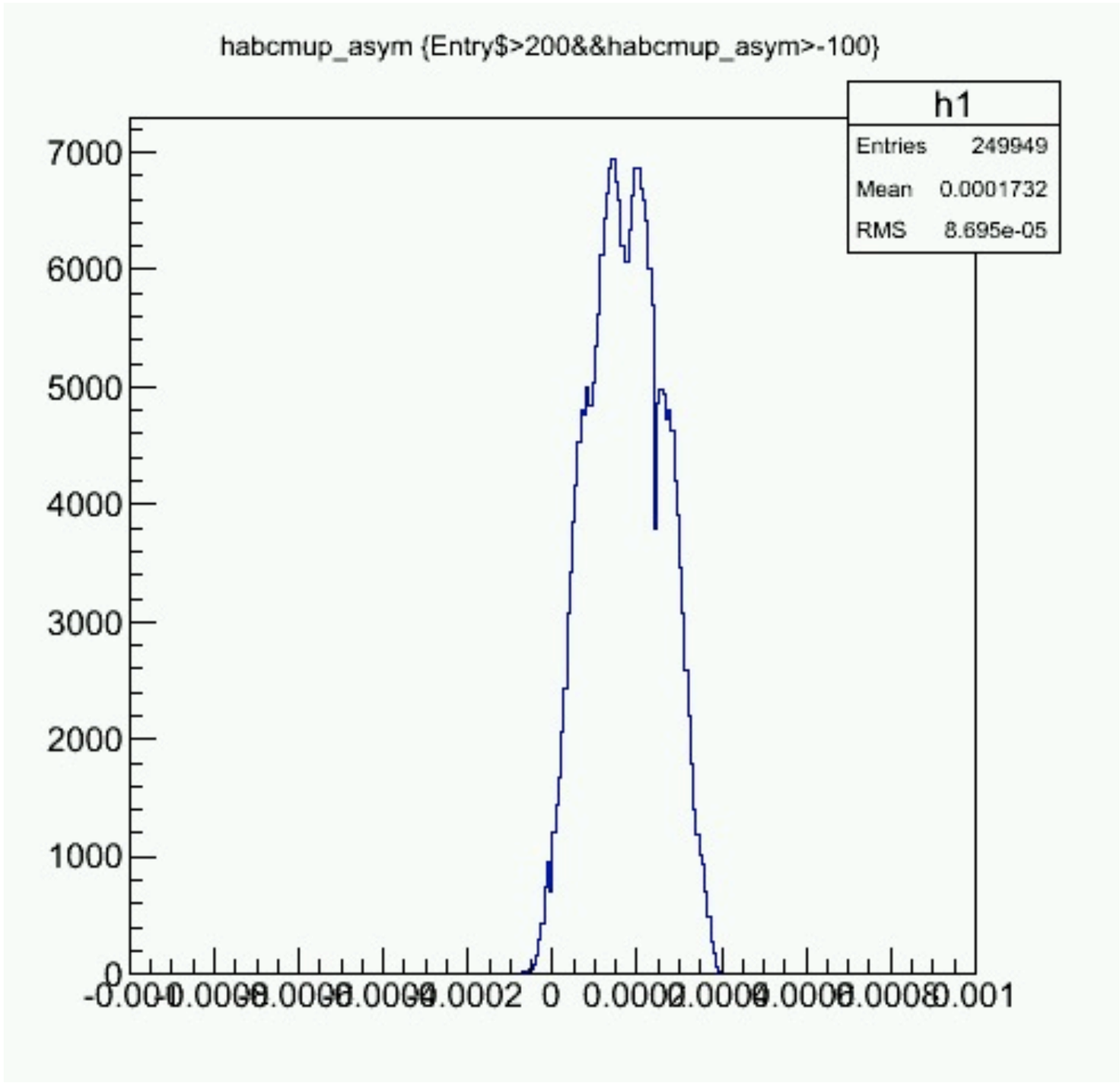


Results, Asymmetry Set Point +190ppm

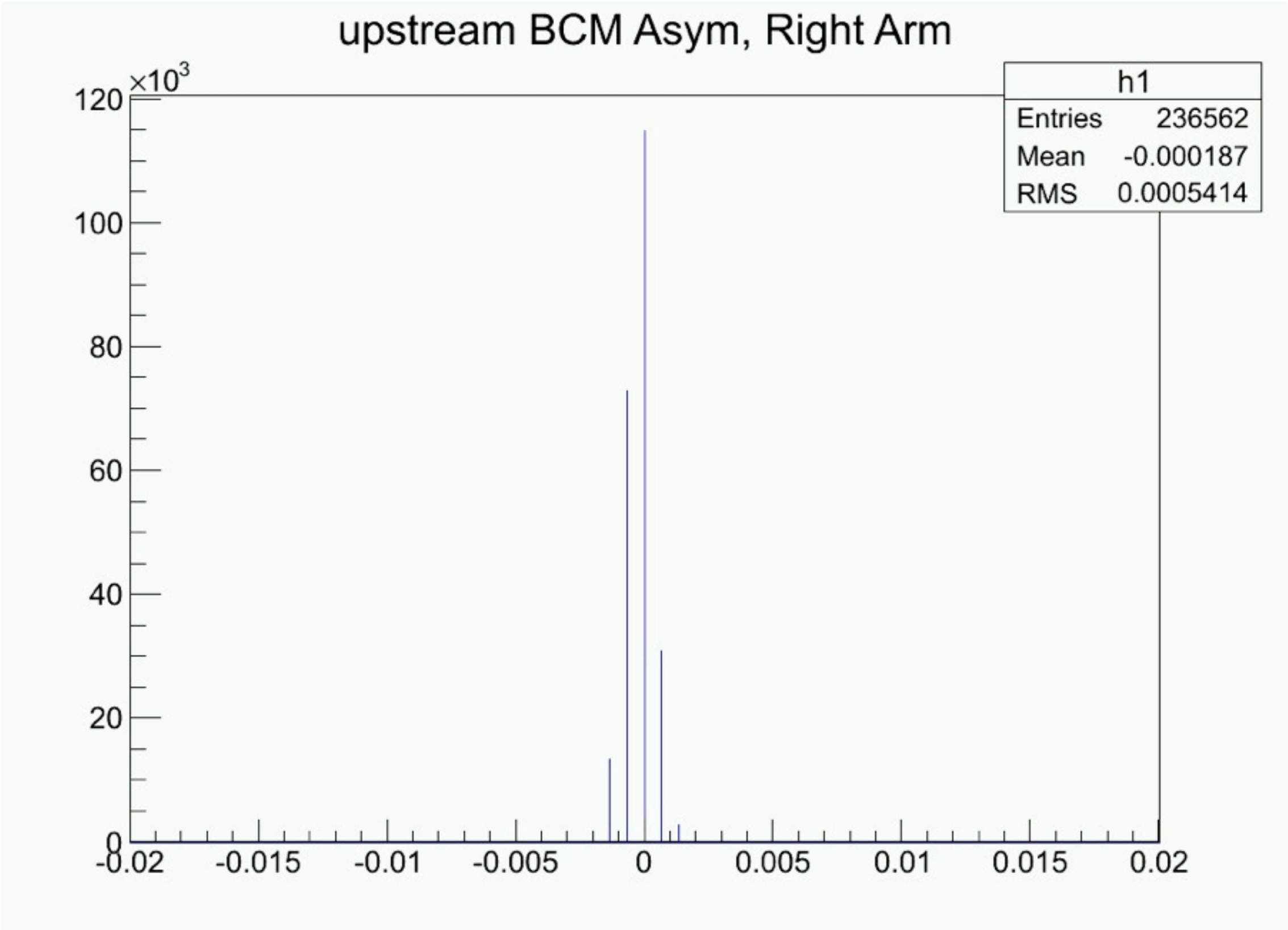


Results, Asymmetry Set Point +190ppm

Result of HAPPEX DAQ, Right Arm

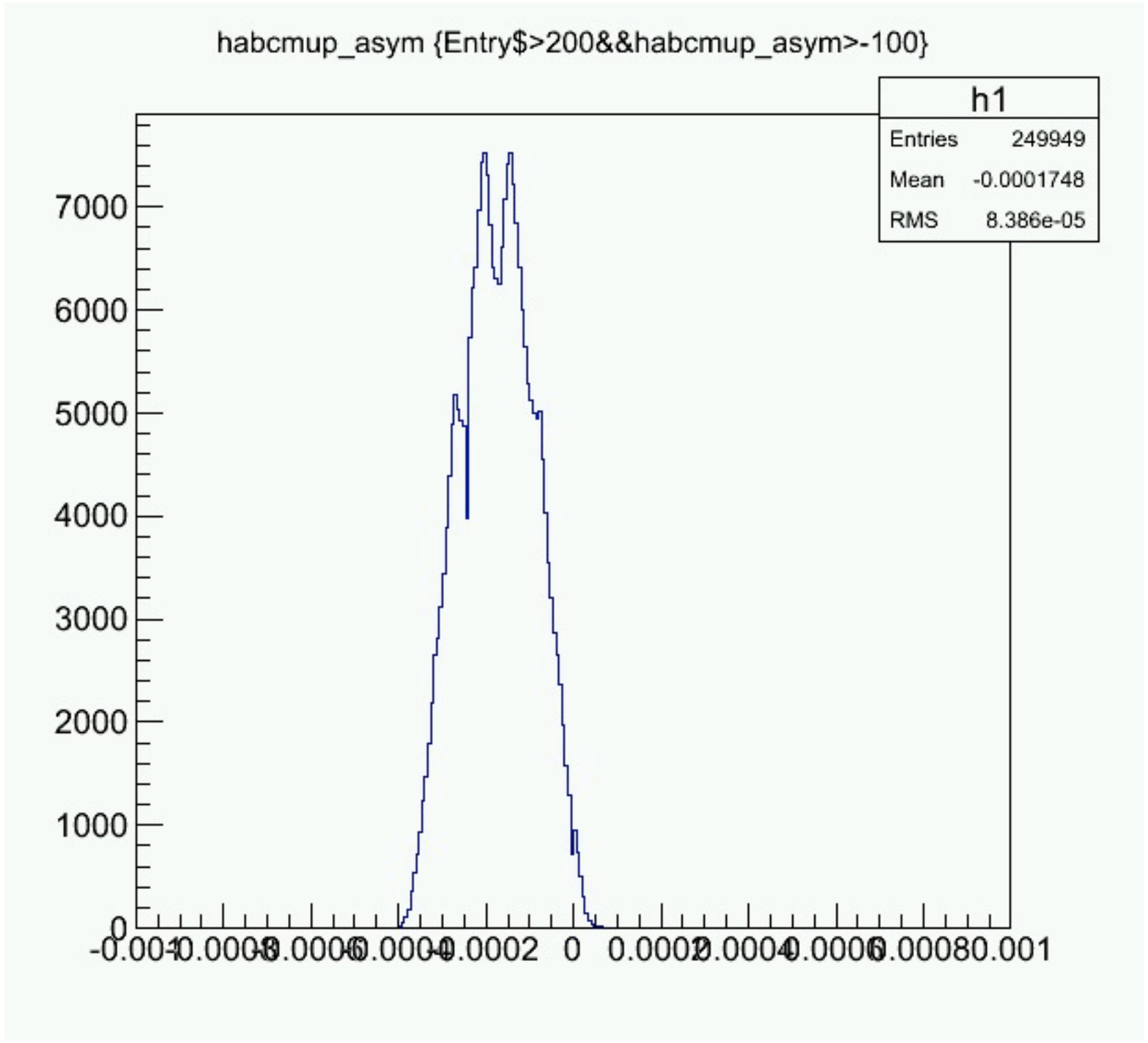


Results, Asymmetry Set Point -190ppm



Results, Asymmetry Set Point -190ppm

Result of HAPPEX DAQ, Right Arm



Charge Asymmetry: Analysis

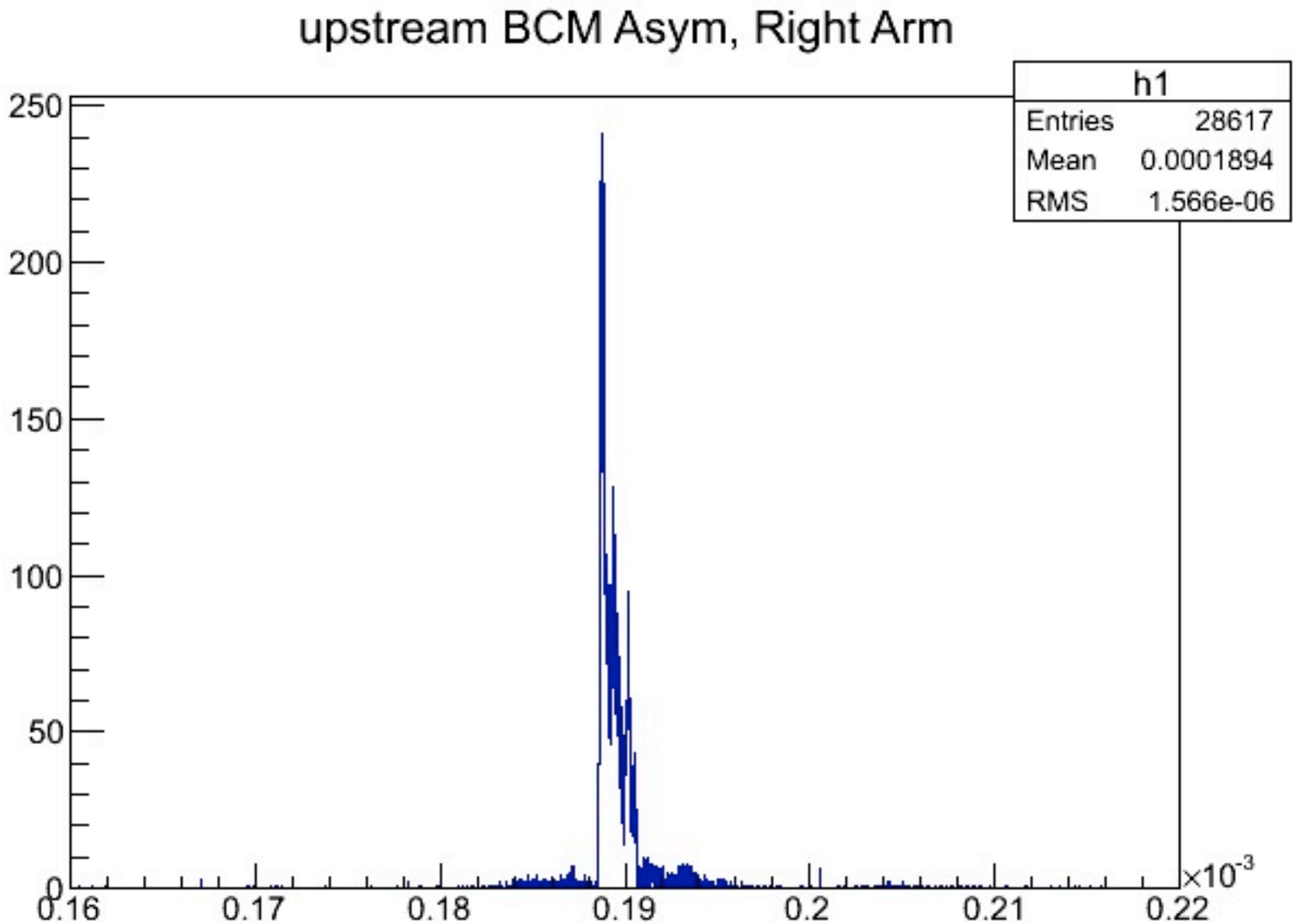
- The HRS DAQ, Moller DAQ and HAPPEX DAQ's results looks consistent with each other
- From the HAPPEX DAQ's result, the bad resolution is because the reflection between the 2 copies of the BCM signal for HAPPEX DAQ and Scaler, which is caused by the T-style splitter
  - This problem has been fixed, will update some plots to show the change
  - Also see Pengjia's talk: http://hallaweb.jlab.org/experiment/g2p/collaborators/pzhu/02082012/happex_asym_report.pptx for more details

Charge Asymmetry: Analysis

- The Resolution of Scaler Ring Buffer can not be improved if we keep quartet as the unit to calculate the asymmetry
- This is because the flipping rate of helicity is too high so the BCM count in each helicity window is relatively small, which causes a huge round-off error (scaler data count can only be integers)
- To prove this is actually a round-off error, one way to do this is to combine several quartets together to calculate asymmetry, however, this will reduce the total event amount

Charge Asymmetry: Analysis

- In this plot, I combined 10 quartets as one unit to calculate charge asymmetry for the asymmetry set point 190ppm run. The RMS changes from 5.4e-4 to 1.6e-6, this run took about 15 mins.



Charge Asymmetry: Conclusion and Todo

- The plot in the previous slide shows the influence of the round-off error, it dominate the RMS of the scaler result
- The HRS DAQ, Moller DAQ and HAPPEX DAQ's results looks consistent with each other
- Hall C also received one BCM signal from Hall A, so we also run a charge asymmetry test with them, we got consistent result
- Need to do a large charge asymmetry test together with Hall C when beam comes back

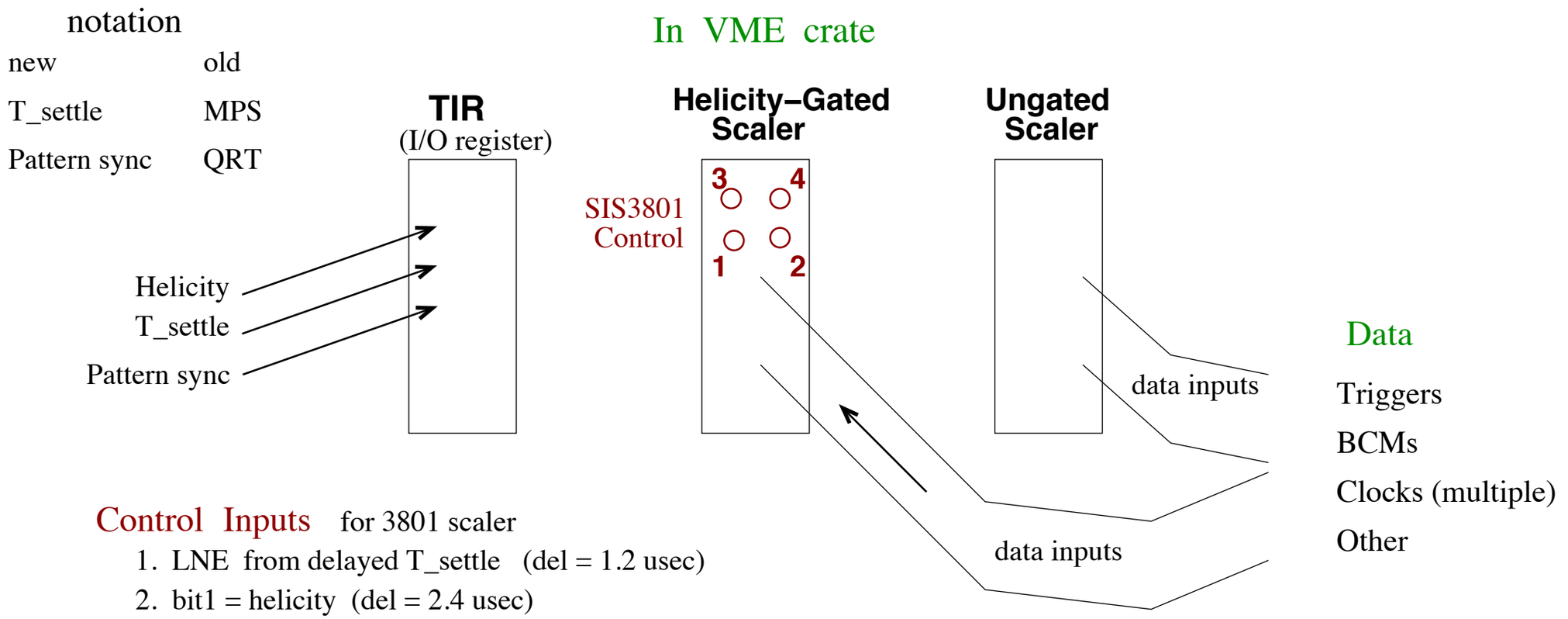
Appendix

Back to Outline

Electronics Scheme for TIR & Scaler

Schematic of Electronics for Helicity Info in Hall A HRS during Qweak

R. Michaels
July 27, 2010



Control Inputs for 3801 scaler

1. LNE from delayed T_settle (del = 1.2 usec)
2. bit1 = helicity (del = 2.4 usec)
3. bit2 = pattern sync (del = 2.4 usec)
4. Gate = T_settle with appro. polarity and undelayed. (TRUE disables counting)

At frequency "f" (f = 1 kHz normally) we must:

1. Read helicity–gated scaler with zero deadtime.
2. Keep a queue in memory of the cpu of [helicity, pattern sync, and clock(s)]
3. Sort the plus and minus helicity, adding the data to "virtual" scalers for online consumption.
4. Check data for errors. Issue warnings.

For each trigger accepted by Trigger Supervisor, we must:

1. Read the (a) TIR bits and (b) clocks from ungated scaler.
2. Flush the memory queue (see 2 above), to have full sequence of helicity data.

Back to Slide