# Spurious Hits from APV25 Readout

Cameron Bravo (SLAC)

Introduction

- Looking into monster events with emulation of APV25 readout buffer

- 2 ns beam clock, 4 ns trigger clock, 24 ns APV25 clock, 840 ns "sync clock" which are when an APV25 readout potentially begins

- When a trigger signal is received, the APV25 puts 6 samples into readout buffer, each taking 3360 ns before the next readout can first potentially begin again

  - Only about 500-800 ns of time around each readout will be masked

- Study timing of events with > 1000 SVT hits (high purity sample of monsters)

- The later two runs were during the PEBAF accelerator configuration

- Used a single 2GB evio file per run
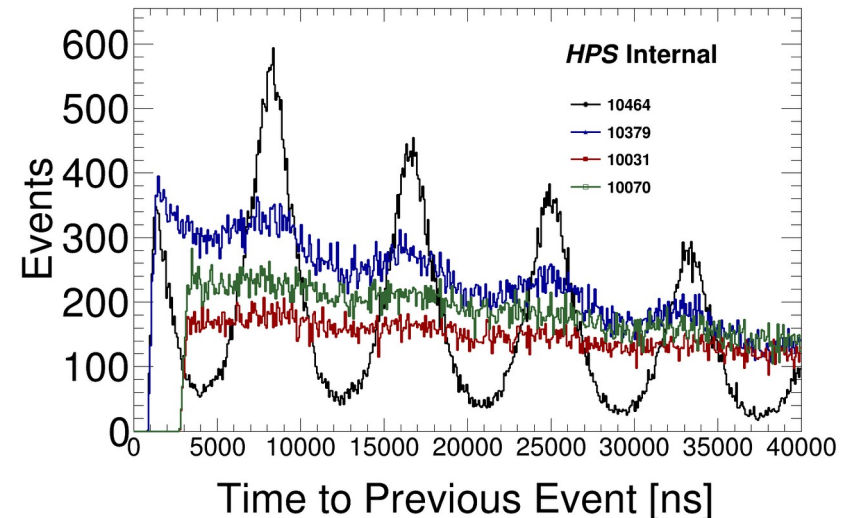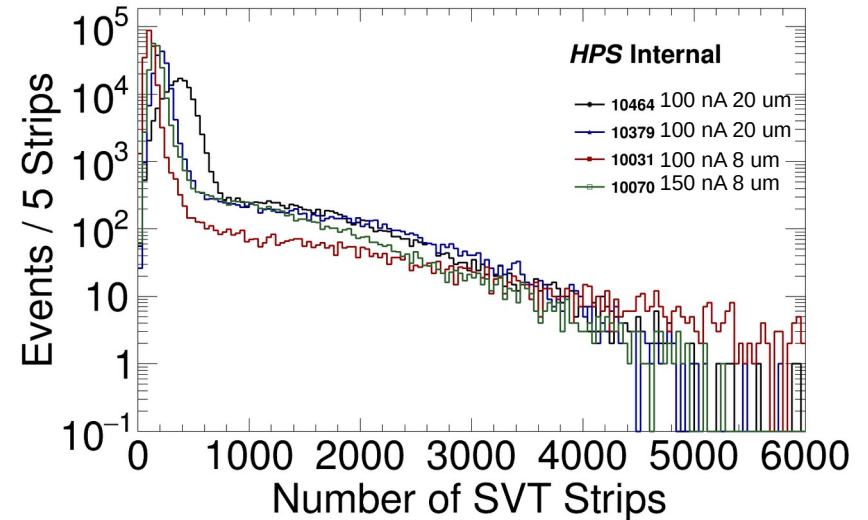
Calibrating the Sync Phase

- Accurately predicting APV25 read times requires knowing the phase of the "sync clock" of the chip, since readout can only begin at these clock edges

  - Must be calibrated for every run

  - 24 ns APV clock in phase with sync pulse: trigger phase = (sync phase)%24

- When using the correct sync phase, you will see the monster event times line up with the predicted read time for all relative sync phases for the time of the event which caused the read which caused the monster event

- Why is there this funny second band?

3
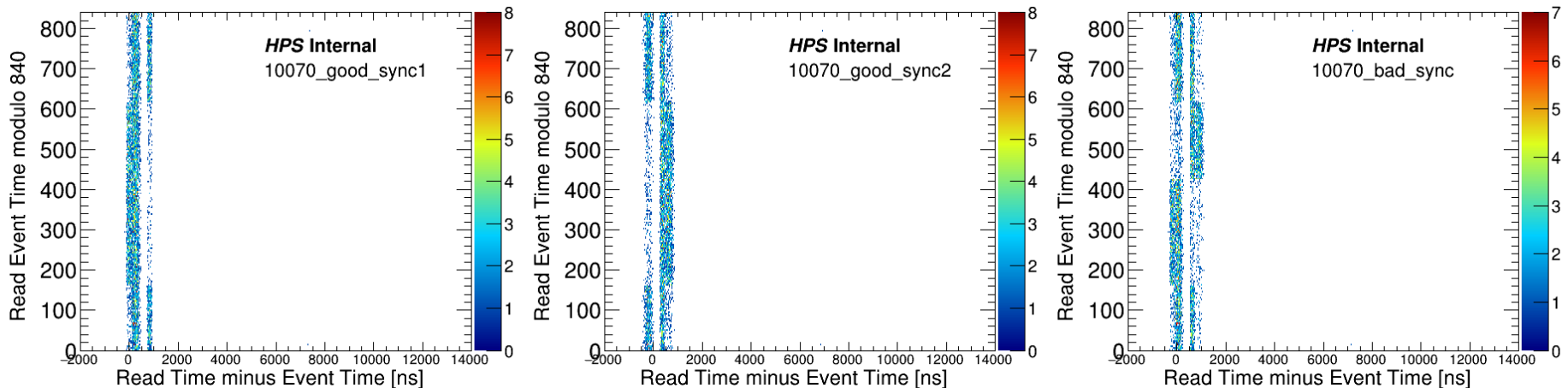
Multiple Sync Phases
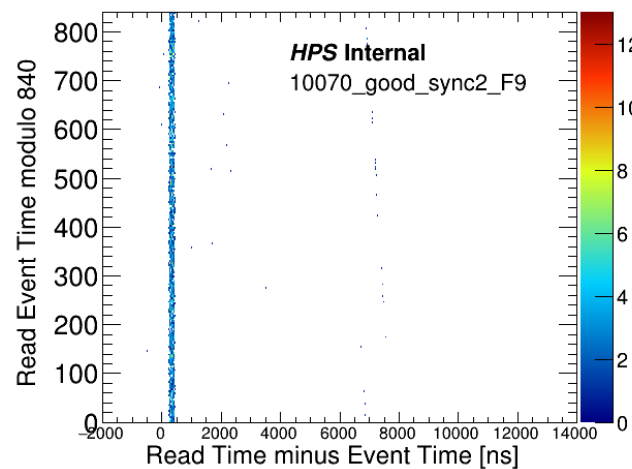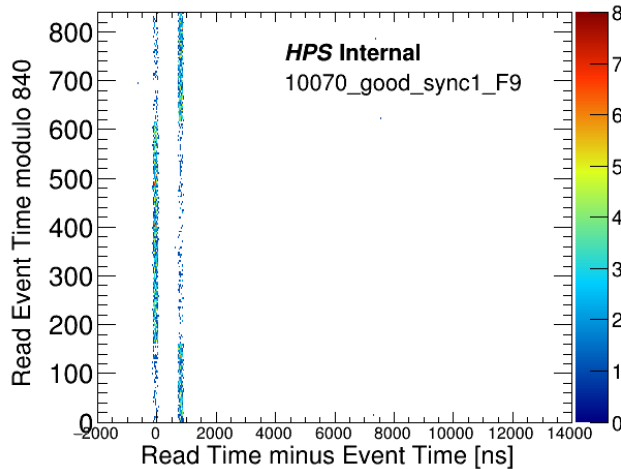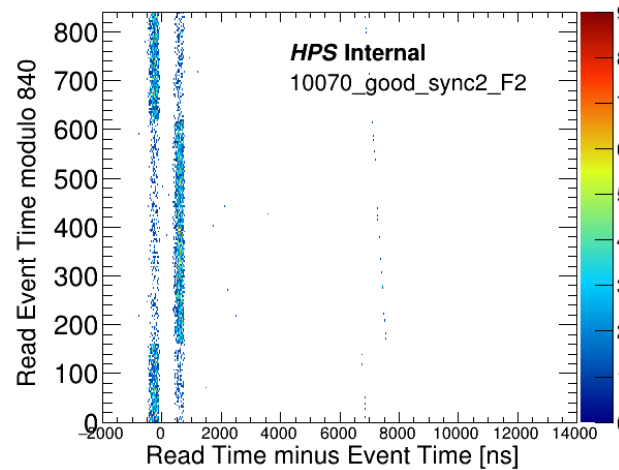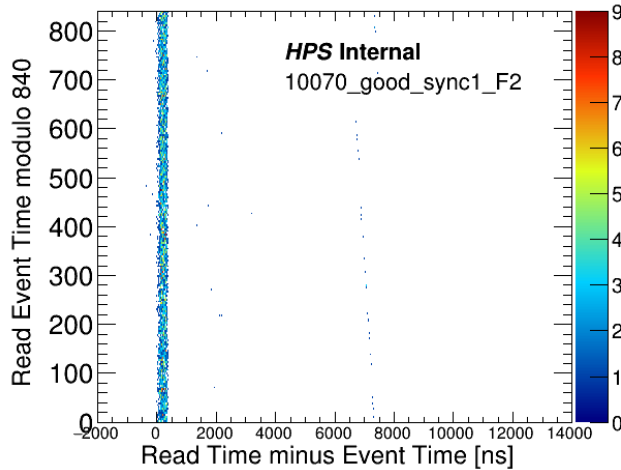
- Multiple runs have 5 FEBs in one sync phase and the other 5 in a different sync phase

  - FEBs 0-4

  - FEBs 5-9

  - Need to confirm it is always separated into these two groups of FEBs, could make sense considering we have two DTMs

- Need to emulate the APV buffer twice

4

# After Calibrating Trigger Delay and Sync Phase

Cut these two time windows relative to each predicted read
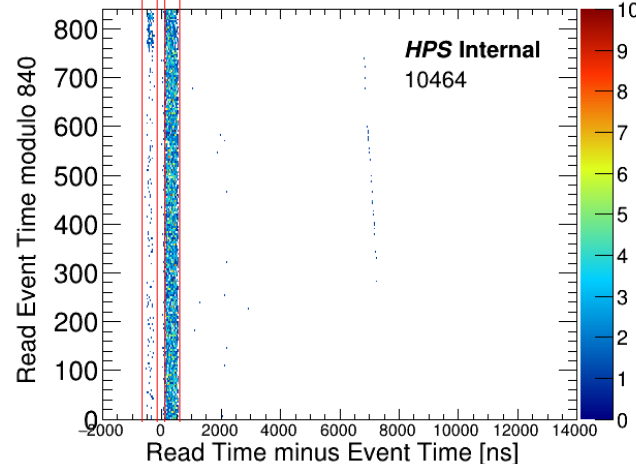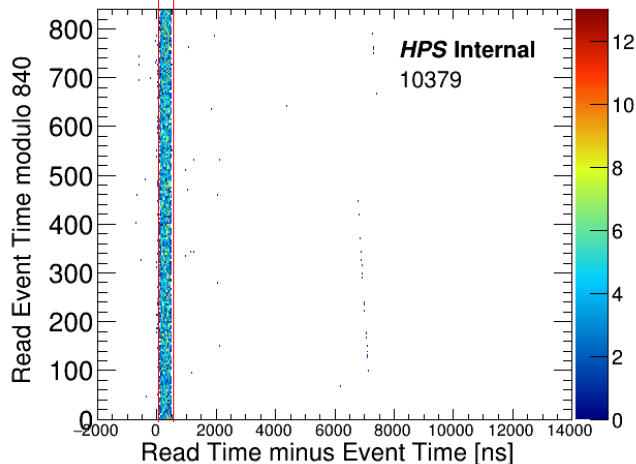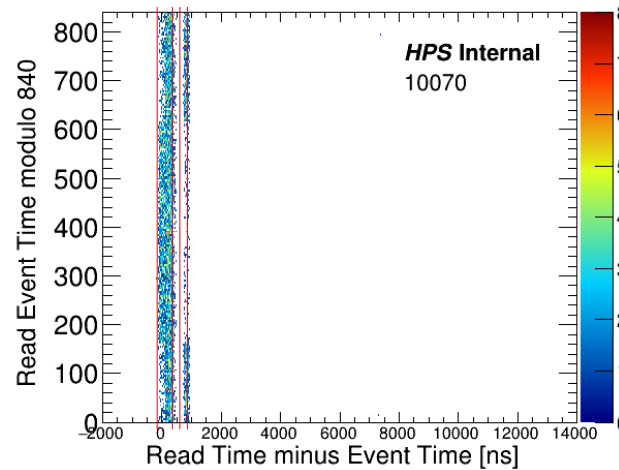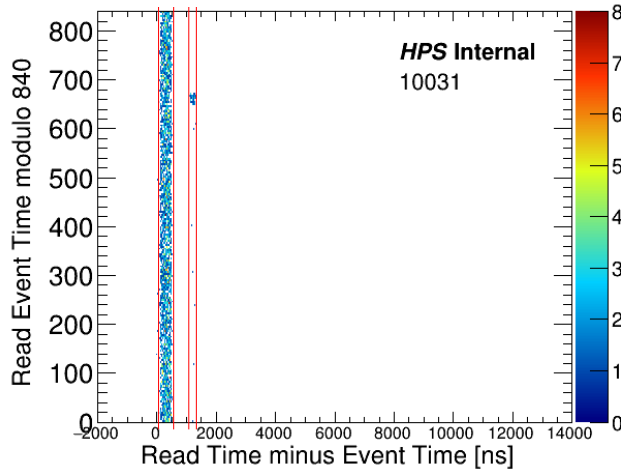
Only one window for run 10379

- Looked at 4 runs

- Most runs have two sync phases

- Remove time windows around reads where we see the events with high svt hit counts

- Band structure is from different FEB contributions

- Difference in sync phase not the same

After Calibrating Trigger Delay and Sync Phase
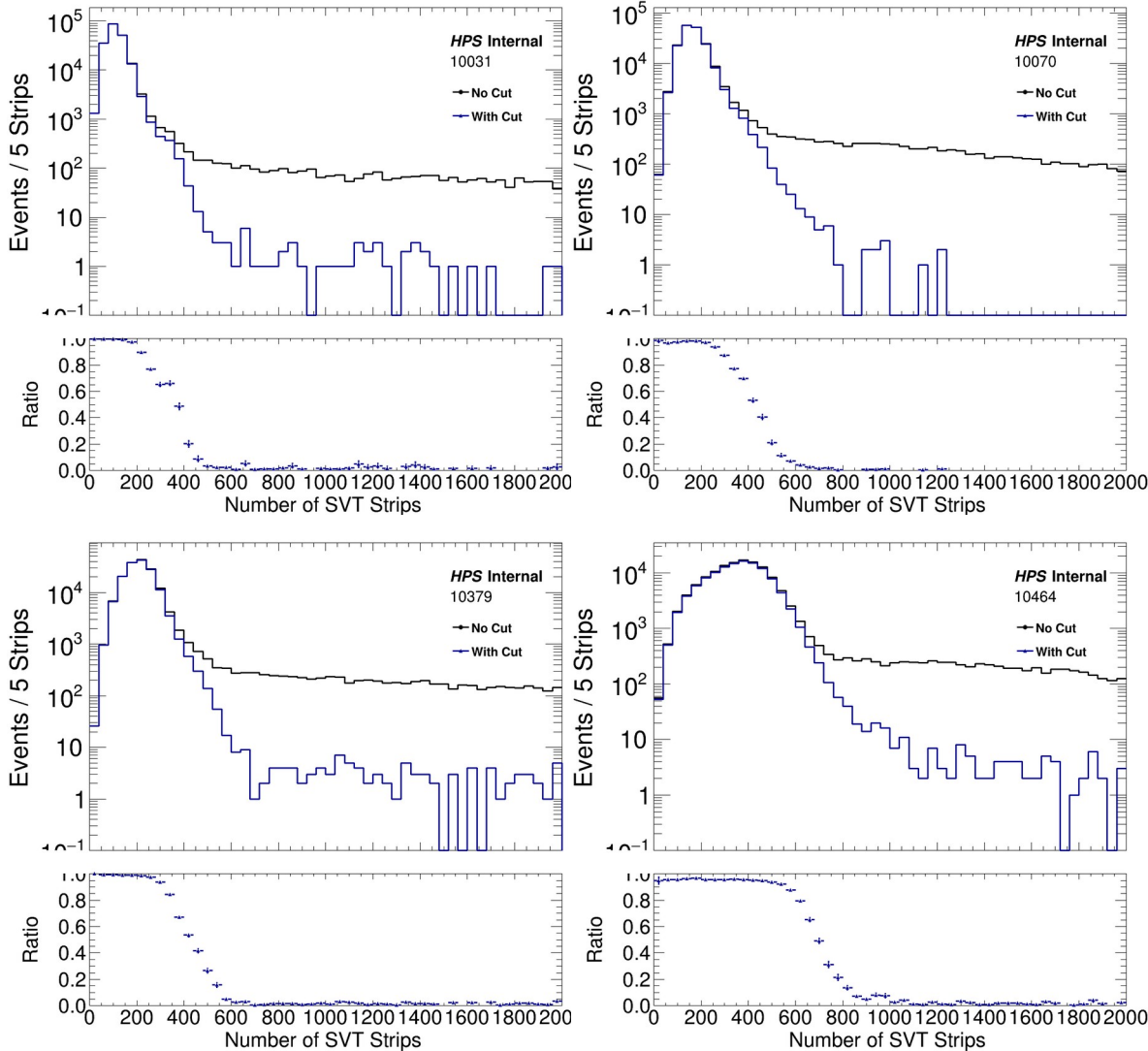
- Cut events within time window seen on last slide of predicted reads

- Could extend cut to include both sync phases, but this will also cut more good events

- Still a bit of a high N hit tail, but it is mostly gone

- Plan to at least try using emulation of buffers in both sync phases to see how that behaves

6

Cut Around Predicted Reads and Next Steps

- Remove events with difference to read time with ranges found on last slide

- Still need to figure out calibrating the sync phase for all of the runs

  - Need to add sql part of Java still

  - Add second buffer to separate emulation of reads on different FEBs

- Still a lot to do to, but mostly understand what we are seeing now