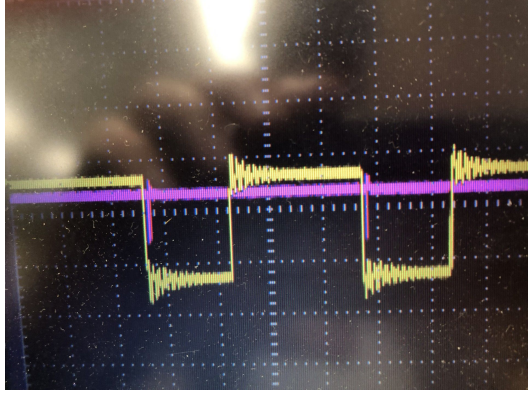
HDSoC and SAMPic Deadtime Studies Update

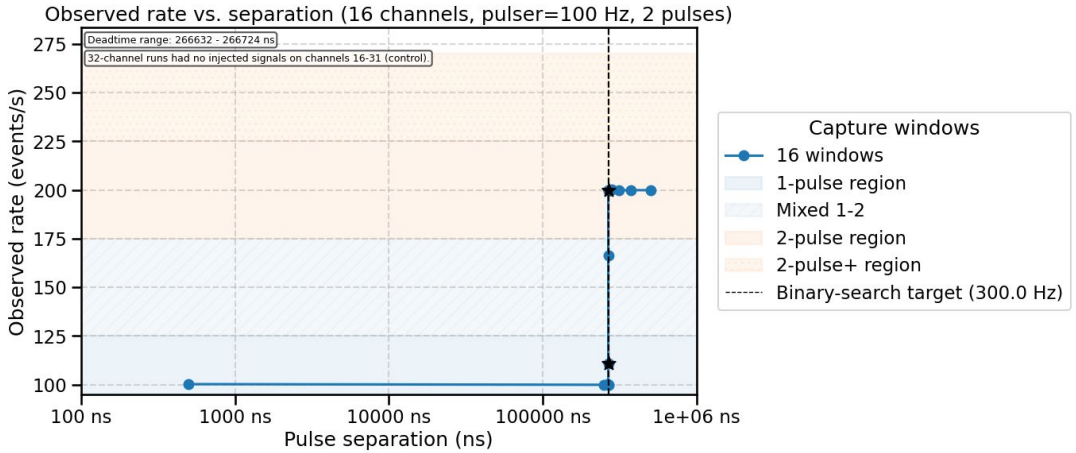
Jack Carlton
University of Kentucky

HDSoC Deadtime (Methods)

- Used Raspberry Pi Pico W + NIM modules to generated configurable double pulse signal
  - Later extended this to 3 and 4 “burst” pulses
- Used HDSoC DAQ to observe event rate for varying parameters
- See spike corresponding to deadtime



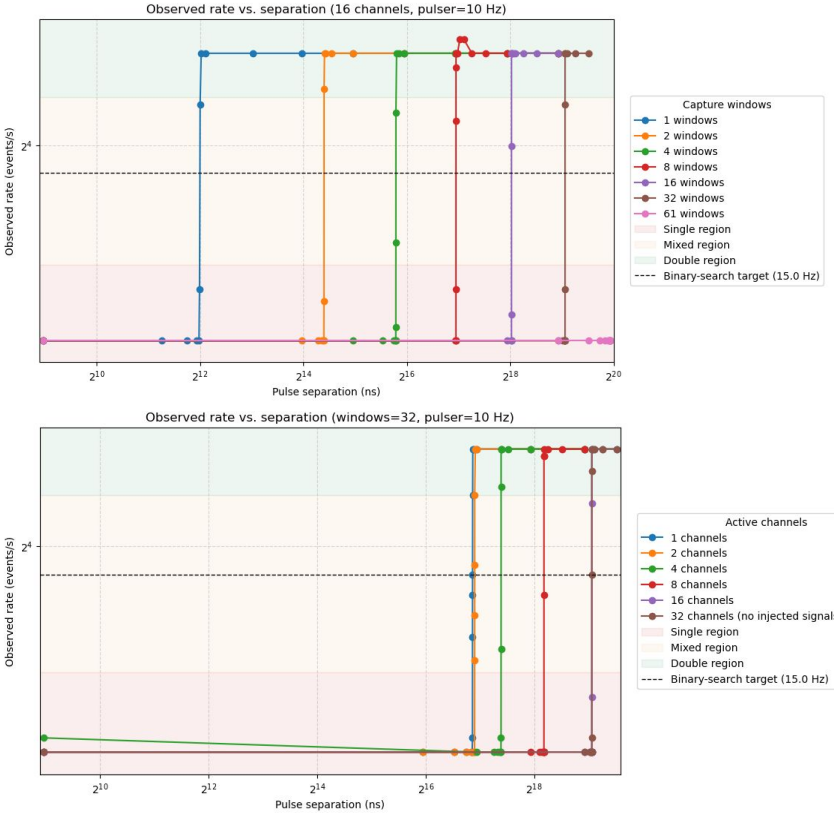
Example Double pulse on Oscilloscope



Example Rate Response to Double Pulse Separation

HDSoC Deadtime (Parameter Scan)

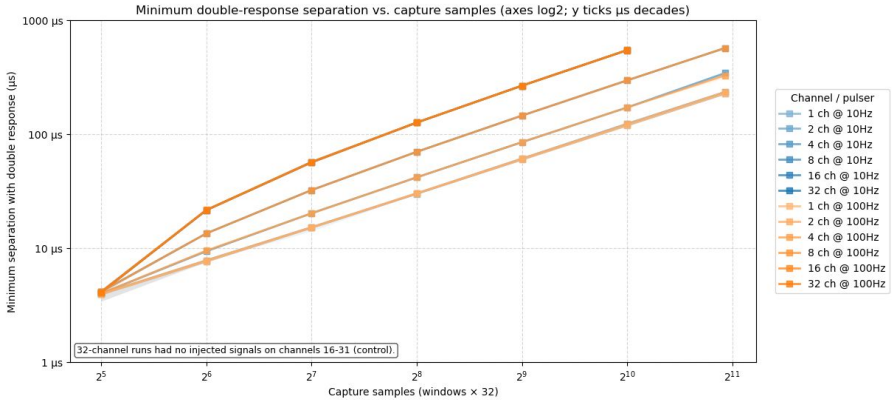
- Vary separation of pulses until each pulse is digitized
  - Binary search around expect event rate
- We tested the parameter space defined by
  - Windows = [1,2,4,8,16,32,61]
  - Channels = [1,2,4,8,16,32]
  - Input signal rate = [10 Hz, 100 Hz]
  - Number of pulses = [2,3,4]
- Space defined as:
  - Windows ⊗ Channels ⊗ Input signal rate ⊗ Number of pulses
- Notes:
  - Signals were fanned out to channels 0-15, no signals were in channels 16-31

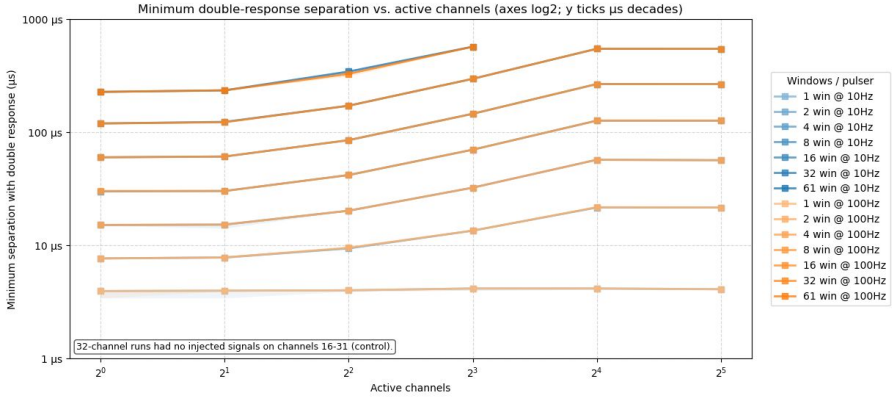


Examples showing channels and windows affect on deadtime

HDSoC Deadtime (Results)

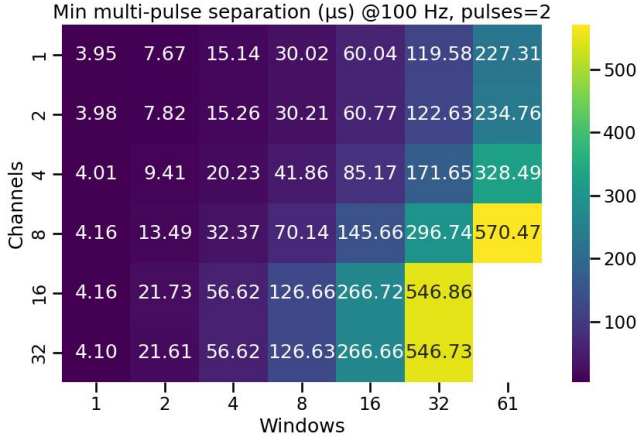
- Deadtime increases with
  - # of active channels
  - # of windows (# of samples, 1 window == 32 samples, this is nalu's terminology)
- Deadtime unaffected by:
  - Rate
    - 10Hz and 100Hz input signal curves overlap
  - Inactive channels
    - Curve flattens when channels 16-31 (no input) are enabled in software
- Notes:
  - JSON file of deadtime upper and lower bounds available
  - Systematically lower deadtime than reality
    - Need to re-run scans binary searching around expected *data rate*, not event rate

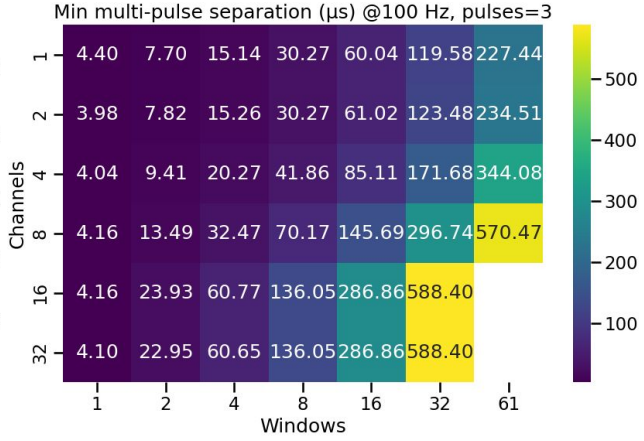


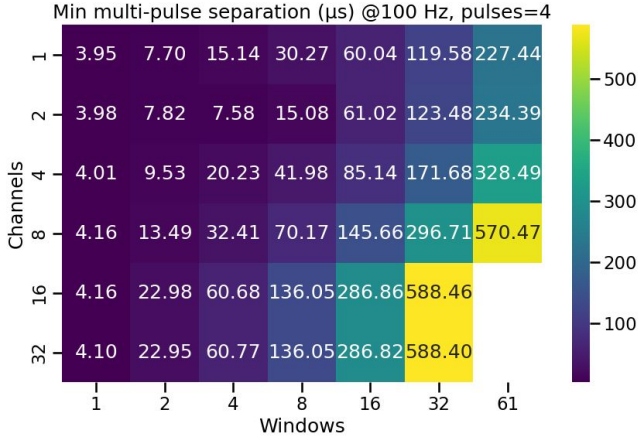


HDSoC Deadtime (Results)

- Deadtime unaffected by:
  - Number of pulses (“congestion”)
- Repeated double pulse tests with 3 and 4 pulses
  - Separation between pulses changed with binary search
  - No meaningful change outside a few isolated bugs in the 4 pulse test

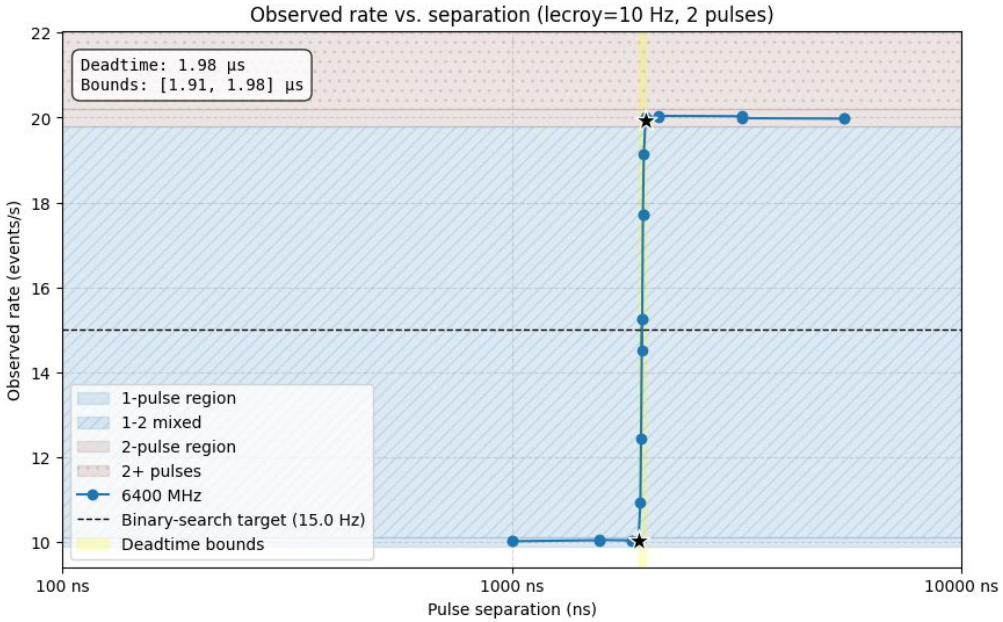






SAMPic Deadtime (Methods)

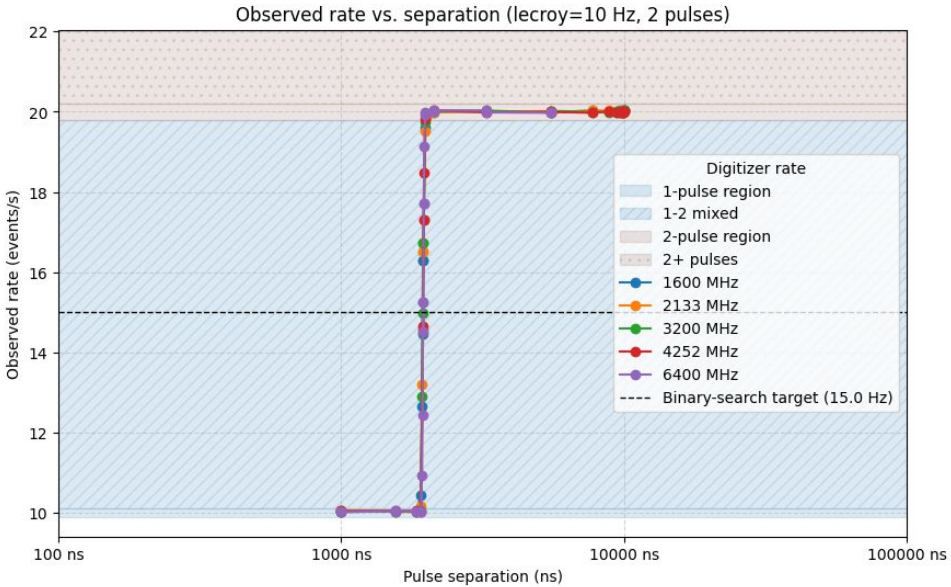
- Used Lecroy 9210 Pulse Generator to create a configurable double pulse signal
- Used sampic_256ch_lib observe hit rate for varying parameters
- See spike corresponding to deadtime



Example rate response to double pulse separation

SAMPic Deadtime (Parameter Scan)

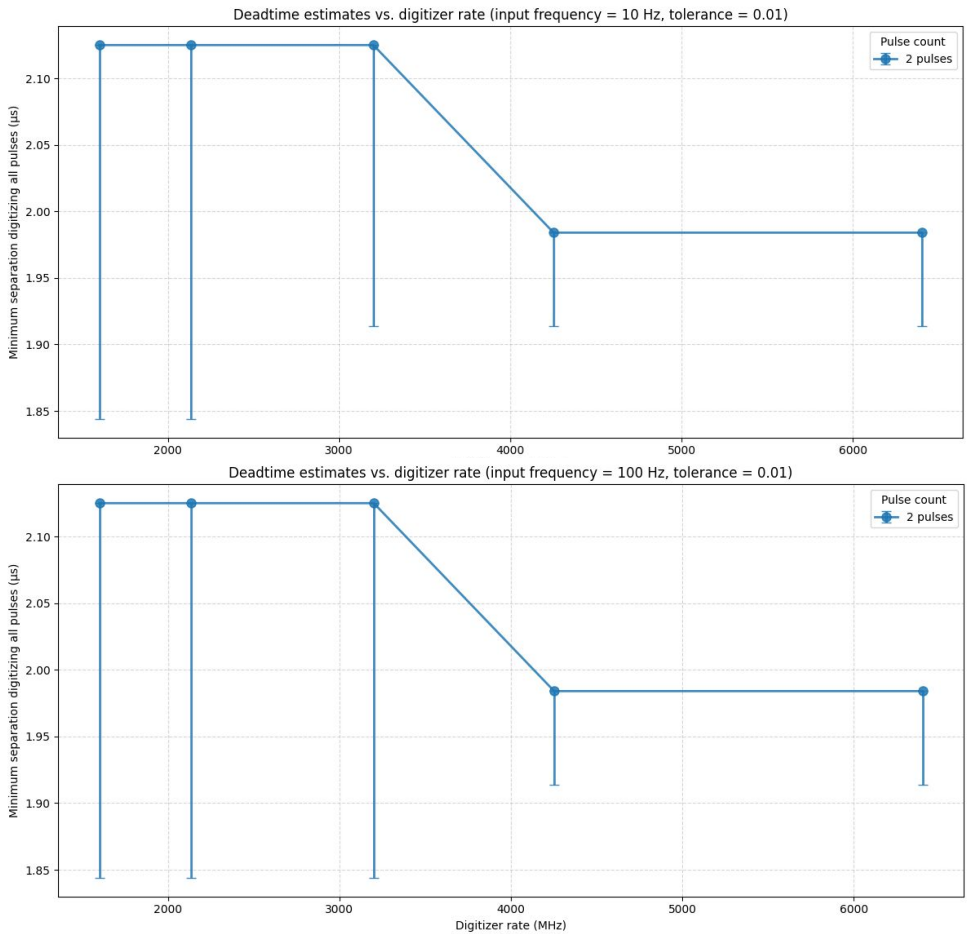
- Vary separation of pulses until each pulse is digitized
  - Binary search around expect hit rate
- We tested the parameter space defined by
  - Sampic digitization rate = [1600, 2133, 3200, 4252, 6400] MHz
  - Input signal rate = [10, 100] Hz
- Space defined as:
  - Input signal rate ⊗ Sampic digitization rate
- Notes:
  - For this scan only one channel was connected to the Lecroy module



Examples showing the effect of digitizer rate on deadtime

SAMPic Deadtime (Results)

- Deadtime unaffected by:
  - Rate
    - 10Hz and 100Hz input signal curves overlap
  - Digitization Rate
    - No significant change in deadtime with digitization rate
- Need to do more scans to see how things scale with number of channels. For one channel, best estimate is **~2µs deadtime**.



Effect of digitization rate on measured deadtime.
Lower bound based on maximum pulse separation where only one pulse was seen.

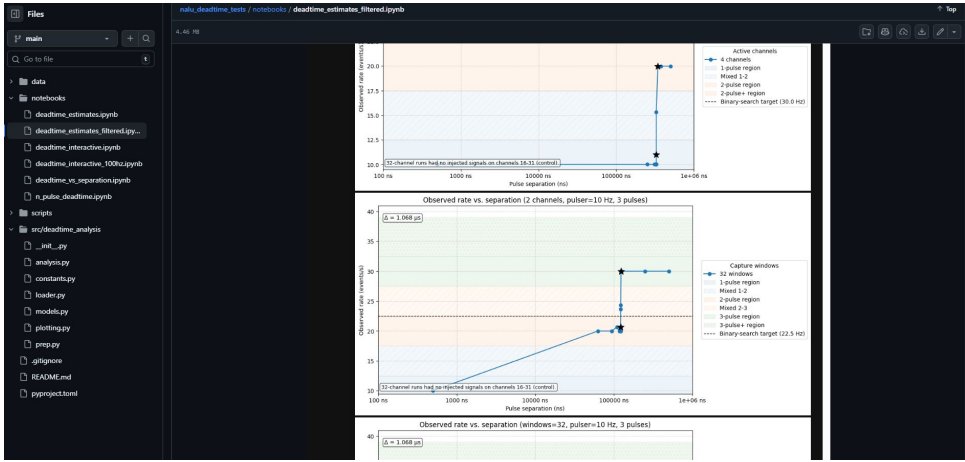
TODO

- Rerun HDSoC deadtime scans with fixes
  - Make sure all 16 channels are getting signals
  - Binary search on data rate as opposed to event rate
- Run the SAMPic deadtime scans with more parameter(s)
  - Number of channels
- Run more realistic deadtime scans with both systems
  - Right now all pulses are synchronized
  - Some configurable offset on pulses/groups of pulses between channels gives insight into channel level vs. chip level deadtimes

Auxiliary Slides

HDSoC Deadtime (Caveats)

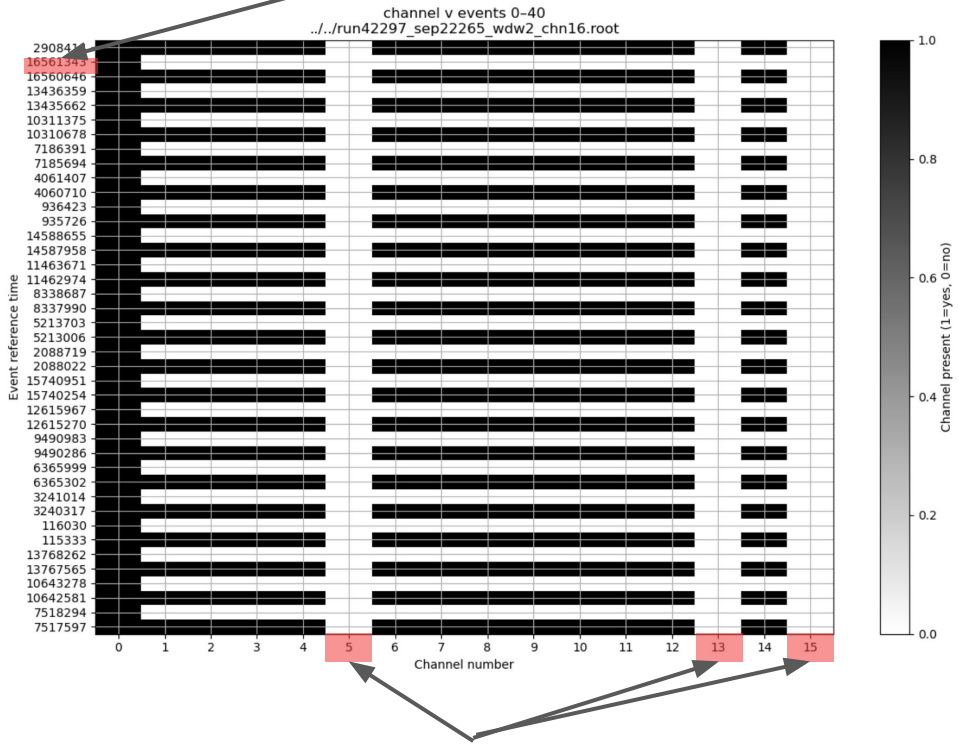
- More plots are available in notebooks hosted on github
  - HDSoC
  - SAMPic
- All data is also hosted on github
  - Could pull repo, make sure you python environment has all dependencies and run interactive notebooks to explore data



HDSoC Deadtime (Caveats)

- Problems with the scans
  - Some channels inadvertently inactivate
    - Probably a faulty connection
  - Binary searches on event rate
    - Some events don't contain the full expected dataset
    - Need to binary search on data rate instead of event rate
- Will re-run scans with these improvements
  - Quantitative results are likely in the correct "ballpack", just systematically underestimated
  - Qualitative results are likely correct

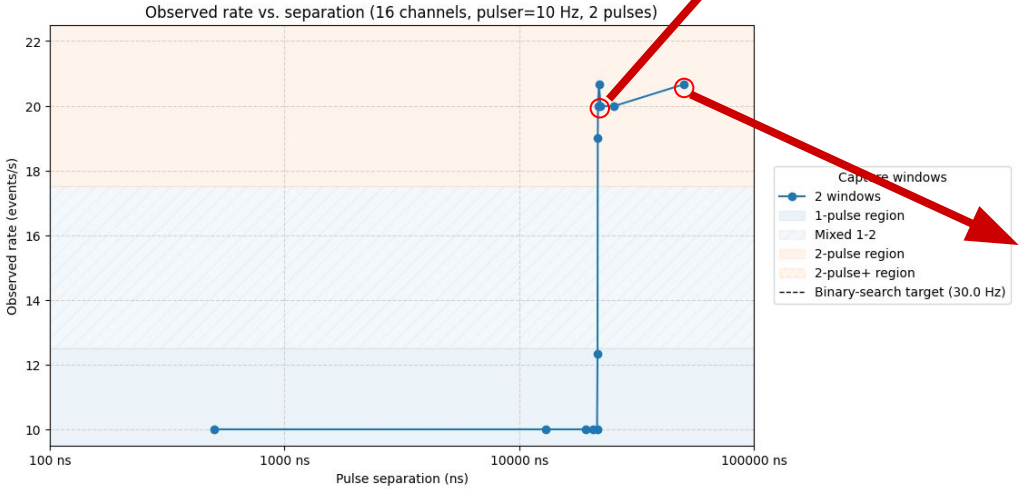
Some events missing majority of channel data

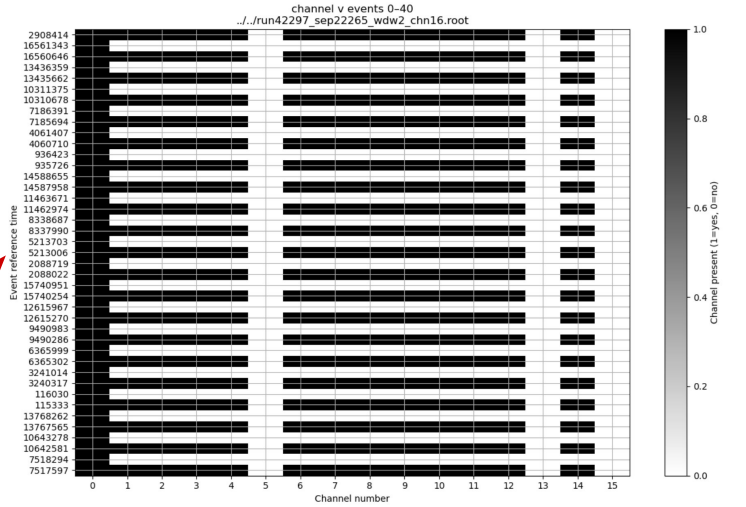


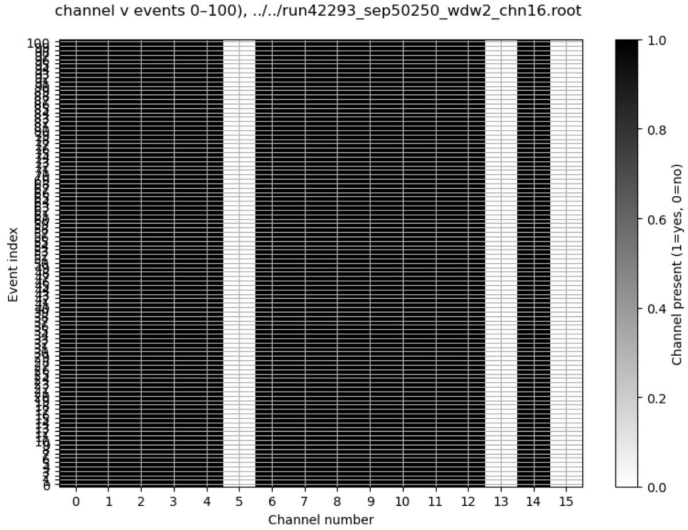
Channels not digitized, maybe bad connections(?)

HDSoC Deadtime (Caveats)

- As pulse separation increase, we begin to see data from the missing channels again



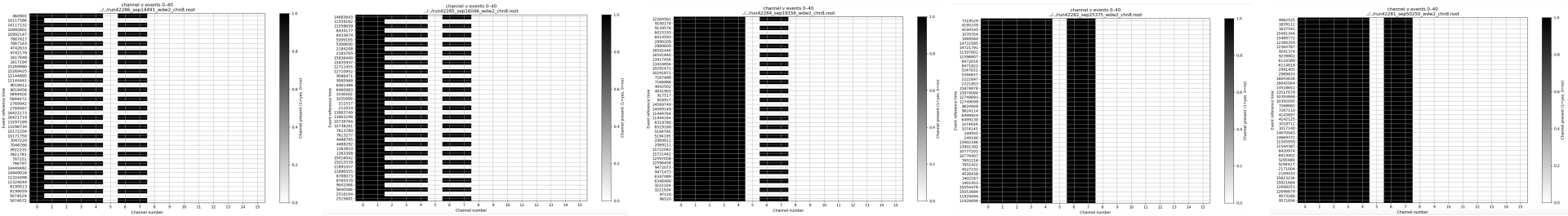




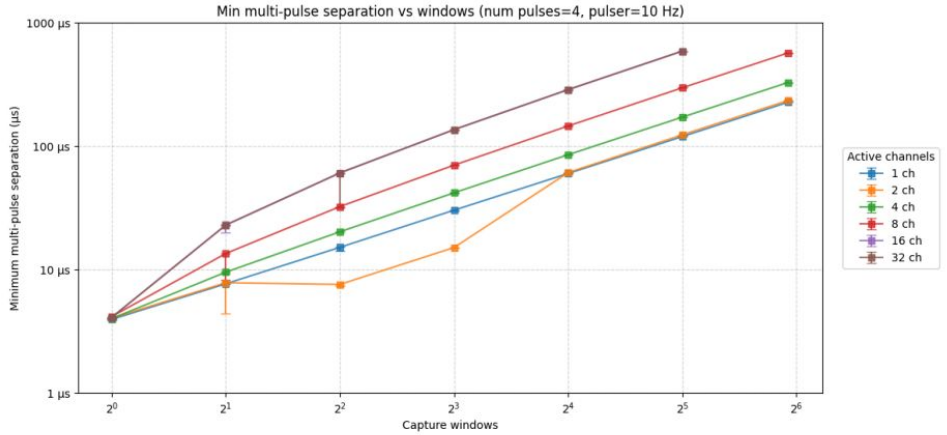
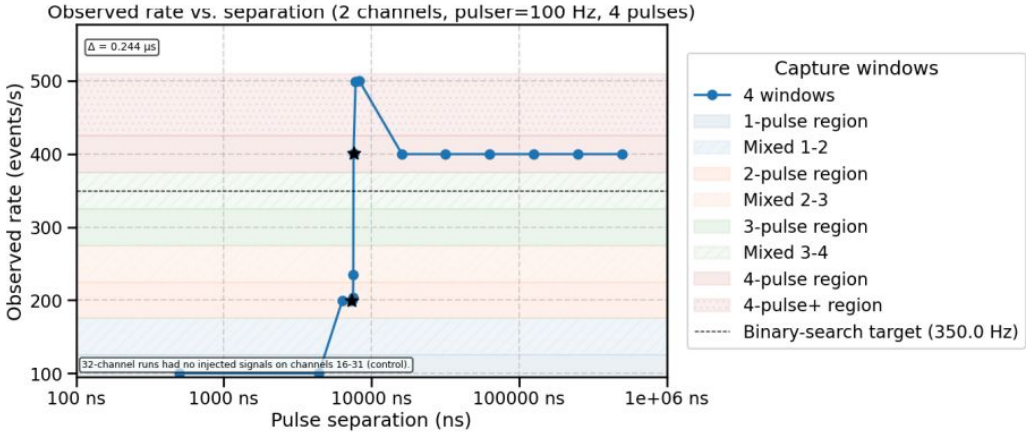
HDSoC Deadtime (Caveats)

- As pulse separation increase, we begin to see data from the missing channels again
  - This is a gradual effect

Increasing separation between the two “burst” pulses



Chanel occupancy plots.
Organized by increasing double pulse separation.
(windows = 2, enabled channels = 8, double pulse frequency = 10 Hz)

HDSoC Deadtime (Caveats)

- Issues using HDSoC DAQ collector for 4 pulse data
- See unexpected dips in some deadtime curves
- Somehow the HDSoC DAQ collector is splitting events, causing more recorded events than actual
- This lowers the point where all 4 pulses are “seen”, causing deadtime to lower