Events in LBNE 35t DAQ

Jonathan Davies, Jeff Hartnell

University of Sussex

Alec Habig

U. Minnesota, Duluth

LBNE Phone Meeting

October 2014

Introduction

- Main running modes
- Blocks of data in DAQ
- Signal and overlap events
- Different “orders” of overlap events
- Required duplication to record 1st order overlap bkg events

- Mark Convery gave a very useful talk at a recent 35-ton phone meeting:
  - http://lbne2-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=9805

Giles Barr
DocDB-9677

Main overall running modes

(most ambitious ones first)

1. Continuous main DAQ mode: Similar to final 'triggerless' far detector running mode, i.e. parallel trigger farm looks for nice muons in real time. Needs ZS of TPC data to work.
2. Triggered main DAQ mode: Main selection is external trigger counters. Buffer all the data like continuous mode and collect all info in a window around any external scintillator trigger. Aim for window to start at least one drift earlier than trigger.
3. Immediate triggered mode [Fallback if bottleneck in either RCE or SSP data]: Use TPC triggered mode (only outputs data in window) or SSP triggered mode to avoid bottleneck
4. Triggered window mode [Use before we are happy with zero suppression in RCE] TPC in triggered non-zero suppressed mode
5. Wide-window burst mode [Special run mode for offline zero suppression studies] TPC in burst mode with the longest window possible (512MB buffer = 0.5 secs!)
6. Burst mode [Fallback before ZS works in RCE, this is also the mode for measuring noise, i.e. this is the mode for November vertical slice] Collect data in certain time window (could either determine this from a NOvA time hash function, or using Penn board trigger input)
7. Scope mode [Special run mode to look at long waveform FFTs etc]

More details of the overall modes

1. Continuous main DAQ mode: TPC: Continuous ZS mode; SSP: continuous mode; Penn-board: ? DAQinterface: Starts tasks incl. parallel trigger processors; Software trigger: Fully operationally configured, uses TPC data selection algorithms

2. Triggered main DAQ mode: TPC: Continuous ZS mode; SSP: continuous mode; Penn-board: ? DAQinterface: Starts all HW tasks and parallel trigger processors; Software trigger: Decode the Penn-board data and trigger when a trigger bit is found.

3. Immediate triggered mode [Fallback if bottleneck in either RCE or SSP data]: TPC: Triggered + ZS mode SSPs: Readout in trigger window; Penn-board: ? DAQinterface: Starts tasks; Software trigger: Trigger on Penn-board trig bit

4. Triggered window mode [Use before we are happy with zero suppression in RCE] TPC: Triggered + non-ZS mode SSPs: Readout in trigger window; Penn-board: ? DAQinterface: Starts tasks; Software trigger: Trigger on Penn-board trig bit.

5. Wide-window burst mode [Special run mode for offline zero suppression studies] TPC: Burst mode no ZS, long window SSPs: Special mode needed?; Penn-board: ? DAQinterface: Starts tasks; Software trigger: Trigger on burst data window (or just pass everything?).

6. Burst mode [Fallback before ZS works in RCE, this is also the mode for measuring noise, i.e. this is the mode for November vertical slice] TPC: Burst mode no ZS SSPs: Special mode needed?; Penn-board: ? DAQinterface: Starts tasks; Software trigger: Pass everything

7. Scope mode [Special mode to look at long waveform FFTs etc] TPC: Scope mode no ZS, SSPs: Off?; Penn board: ? DAQinterface: Starts tasks; Software trigger: Pass all to mon only

Just focusing on modes 1 and 2 in this talk

Also, focusing on TPC, not SSP or PennBoard

Blocks of Data

- DAQ systems by their nature are very "blocky"
  - they typically move data around in chunks
- artdaq is a generic DAQ framework that moves these blocks through the DAQ system and to disk
  - LBNE-specific artdaq "Board Reader" processes determine the block size and structure
- To capture all the necessary data from a given signal event we can intuitively work in units of a full **drift time** (call this a drift length)

A Starting Point

- The DAQ group has suggested the following:
  - microSlice = e.g. an RCE's data for 3200 digitisations (1 drift time at nominal HV), SSP could be different length
  - milliSlice = collection of microSlices, e.g. an RCE's data for 32000 digitisations (10 drift times at nominal HV)
  - milliBlock = “event” = all the milliSlices for **whole** detector (RCE **and** SSP **and** PennBoard).

- Note: milliBlocks are the basic blocks that flow through artdaq
  - DAQ group configures these to provide the performance and data structure that is desired by the experiment as a whole

Approximate TPC data sizes

- Without zero suppression (ZS): ~64 Gbit/s (was 72 but do 16/18 FEBs)
- Cosmic rays with ZS: 270 Mbit/s
  - (My back of the envelope calculation came up with ~100 Mbit/s)
- Assuming a factor of several hundred from zero suppression
  - a factor of a few worse here would be significant!

- 10 drift lengths at nominal voltage = 32000 digitisations = 4.3Mbit with ZS, (800 Mbit = 100 MB without ZS)

- Other numbers (only approximate)
  - 2048 wires in 35t, 400 Hz cosmic rays, 2 MHz digitisation, 12 bits/digitisation, 1.6 ms drift is 3200 digitisations (or "ticks") at nominal voltage
  - ASIC reads 16 wires, FEB has 8 ASICs (16x8=128 wires), 2048 wires = 16 FEBs
  - An ATCA crate nominally has 8 RCEs, we have 1 ATCA crate (?)

Zeroth order TPC data collection

- To collect all the hits from a **signal event*** you need simply 1 drift length of data

  - * signal event = prompt ionisation deposited well within 1 clock tick of anode wire readout (that you're interested in)

Overlaps in a TPC

- Define: $\mathbf{t_0}$ = the true start time of the prompt ionisation in the TPC
- Define: $\mathbf{t_{end}}$ = the last possible time a hit could occur on a wire from prompt ionisation in the TPC (i.e. ionisation by the cathode plane)

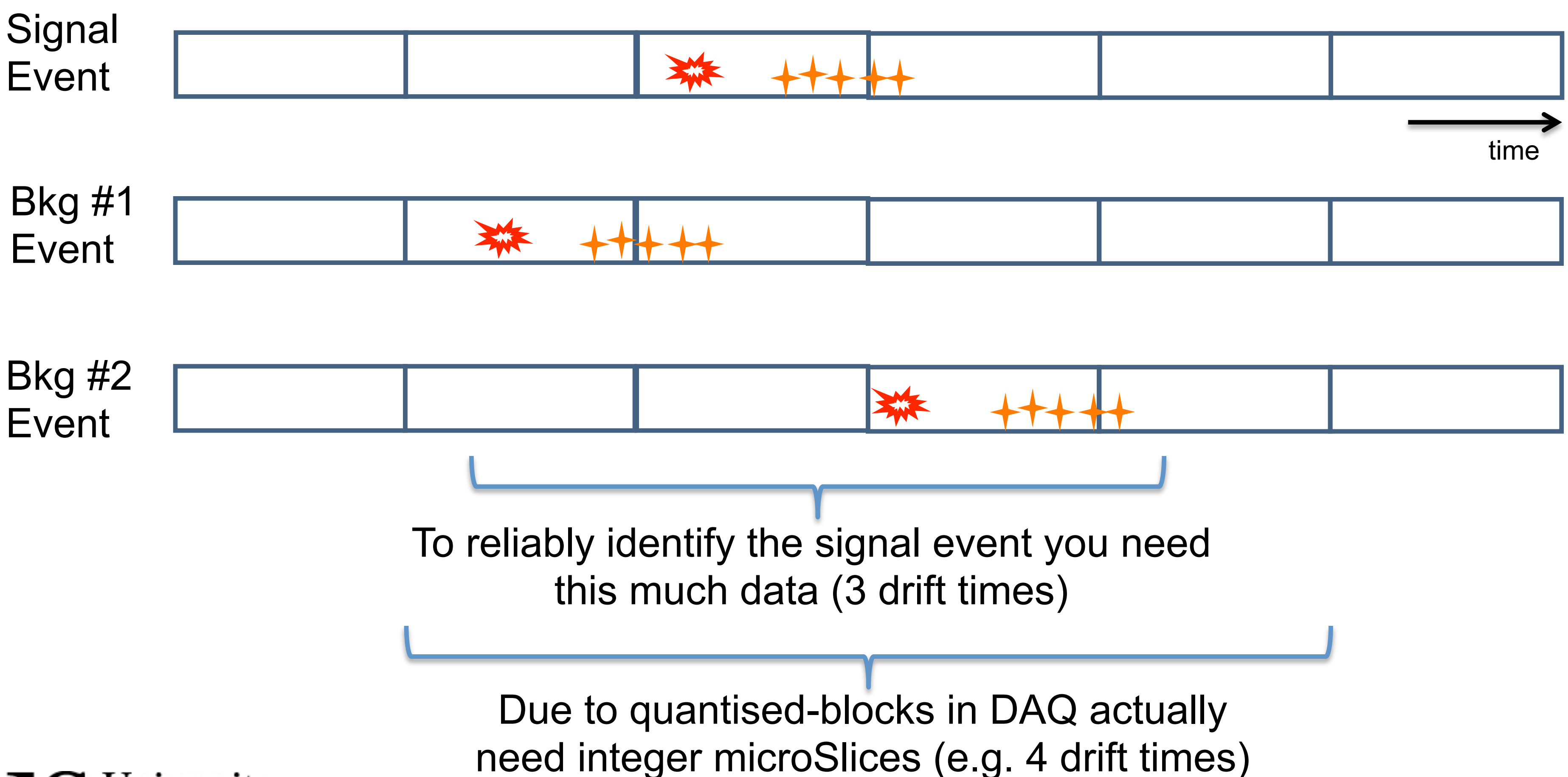
- In a high rate environment you can have background events that overlap the drift time of the signal event, either before or after
- To be able to subtract/remove the background overlap event you want EVERY piece of it (not just the hits that overlap in time the signal event drift time) so you can better identify it, say, as a cosmic muon
  - A bkg event with t_0 up to 1 drift length **before** the signal event t_0 can have hits that overlap hits from the signal event. You need the entire drift time of data for the bkg event.
  - A bkg event with a t_0 as late as the t_{end} of the signal event needs to be recorded in full. Such a bkg event would have a t_{end} = 2 *x* drift time after the signal event t_0
- MicroBooNE use 3 drift lengths of data for each signal event

Example of signal + overlap bkg events

[box] = 1 microSlice (assume it's 1 drift length of time)

[red burst] = t_0 (from SSP/PD or External Counters)

[orange star] = t of TPC hit (from RCE)



University of Sussex

Different "orders" of overlaps

- Define **first** order background overlap events
  - those that could have hits that overlap the 1 drift time of **signal event**
- Define **second** order background overlap events
  - those that could have hits that overlap the 1 drift time of **first order background events**
- Can define **third** order... and onwards
- In an ideal world... record 2^{nd}, 3^{rd}, ...(?) order overlap events too (perhaps for beam spills?)
  - e.g. reconstructing 1^{st} order overlap bkg events could be aided by having all the information on the 2^{nd} order bkg event that partially overlaps it in time.

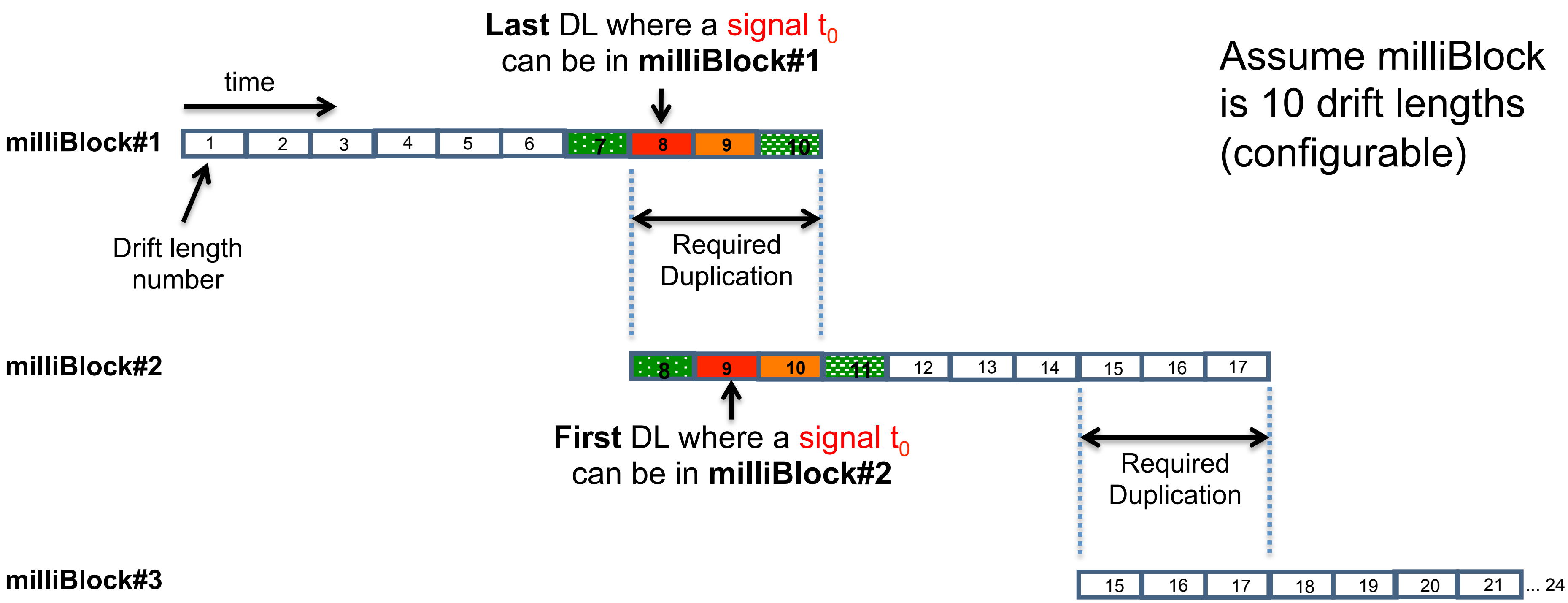
Independent triggering on milliBlocks

- Current scheme...
- Online triggering is done in parallel by multiple processes
- milliBlocks will be sent to independent processes (potentially on multiple computers)
  - they can't talk to each other
  - so have to send extra (duplicate) information about overlap background events inside each milliBlock
  - see next slide...

Required duplication to record 1st order overlap bkg events

1 Drift length (DL)	Signal t_0	Signal hits	1st order overlap bkg t_0	1st order overlap bkg hits
	Yes	Possibly	Possibly	Possibly
	No	Possibly	Possibly	Possibly
	No	No	Possibly	Possibly
	No	No	No	Possibly

Assume milliBlock is 10 drift lengths (configurable)

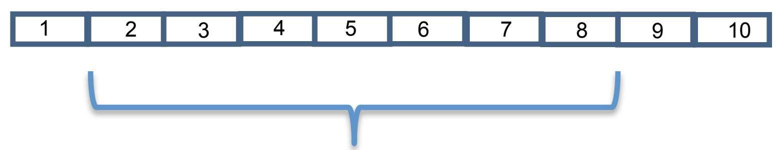


University of Sussex

Trigger “fiducial” region in each milliBlock

- In a given milliBlock a trigger process would ideally only trigger on events with a t_0 in drift lengths 2-8 inclusive.
  - These are the “fiducial” drift times.

milliBlock



Ideally only trigger on signal events with t_0 in drift lengths 2-8 in this 10 drift length long milliBlock

Not enough information to write out bkg overlaps events for drift length numbers 1, 9 and 10. These drift lengths will be triggered on in the milliBlocks that come before and after.

Conclusions

- Offline analysis requires that (whole, full drift length) 1st order overlap background events are also recorded alongside the signal event
- To do this in the DAQ in mode-1 a duplication region that is 3 drift lengths long is required
  - if we just write everything overlap is not required

Backup Slides

Making trigger decisions

- With 270 Mbit/s of ZS cosmic data this could all be written to disk (3 TB/day, 100 TB/month)
- With 64 Gbit/s **non-ZS data** it's not feasible to write it all (700 TB/day, 21 PB/month)

- A ZS 32000 tick milliBlock is 4.3 Mbit but simple unpacking removes ZS so it blows up to ~100 MB
  - there are 62.5 such milliBlocks per second
  - 6 GB/s is about a DVD of data per second
    - that's a lot to push through the CPU/RAM
    - Googling, RAM write speeds are perhaps 30 GB/s
  - Bottom line: simple unpacking in a trigger would probably require multiple servers
  - on a single server, triggering on header information (nhit above ZS threshold, PennBoard trigger bits, large PD hits) should be feasible

Example of signal + overlap bkg events

= 1 drift length of time

= t_0 (from SSP/PD or External Counters)

= t of TPC hit (from RCE)