Updates of the Readout System for 2019 MC

T. Cao

March 25th, 2020

Review of the Readout System

- The 2016 readout system developed by Kyle is robust. Overall, the system includes digitization of detector's response and emulation of the trigger system.
    - Digitization: Hits from detector simulation -> ADC samples
    - Emulation of the trigger system: Emulate GTP processing to generate triggers for readout of ADC samples.
- Drivers in the system are closely correlated:
    - Drivers for Ecal and Hodo work as a chain, separately.
        - Ecal chain: truth hits -> digitization -> raw conversion -> GTP cluster
        - Hodo chain (2019 readout system): truth hits -> hit processing -> digitization -> raw conversion -> hodo hit patterns
    - GTP clusters from the Ecal chain and hodo hit patterns from the hodo chain are input of trigger drivers.
    - A readout data manager driver manages all drivers:
        - Control time line of the system
        - Take care of time displacement among drivers, including time displacement due to NSA in digitization drivers, time displacement due to clustering temporal window in GTP cluster driver, and trigger displacement in trigger drivers.
        - Caches data from output of drivers so that the cached data could be used for the following drivers and be output after triggered.
        - Triggers signal are sent to the manager, and the manager determines what data are finally output.
- Besides ADC samples for hits of Ecal, hodo and SVT, other useful information can be collected in output of the readout system, like truth information (MC particles, truth hits), relation between truth hits and ADC samples, etc.

Several Issues in the Readout System

- All issues are logic problems, and there are no errors when compilation and running. But on the other hand, they are hard to be found and fixed.
    - Gains of hodo. channels
    - Threshold-crossing sample is a part of NSB or NSA
    - Deadtime for pulse integration
- Note: We need to estimate effects of the above issues on 2016 MC data to determine if re-production is necessary.

Issue 1: Gains of Hodo. Channel

- Ecal and hodo share the same digitization driver: DigitizationReadoutDriver. The driver processes truth hits to transform pulses into FADC hits, where pulse amplitude is proportional to hit energy.

- By default, unit of hit energy is MeV, like energy of Ecal hits. However, unit of hodo gain in the database is self-defined (pulse * gain = 1000 ADC), instead of MeV. To let the digitization driver can process hodo hits like Ecal hits, we need to do conversion for hodo gains from self-defined-unit/ADC to MeV/ADC. Energy of hits on holes is 0.833333 MeV on average (crystal energy / 2; referring to energy at peak in distribution), so the conversion factor for hodo gains in the database is 0.000833333 MeV/self-defined-unit.

- Notes:
  - It is the reason that there was no hodo hit output from readout when I checked the readout system at first.
  - The conversion factor changes as beam energy. To be used in the future, a variable “factorGainConversion” is defined in HodoscopeDigitizationReadoutDriver, and we can set it in steering files.
  - Hodo gains are set as uniform: 0.000833333 MeV/ADC when Kyle did his study, so no such issue at that time. But we updated the database for hodo gains according to Rafo's analysis later.
  - The issue has no effects on 2016 MC

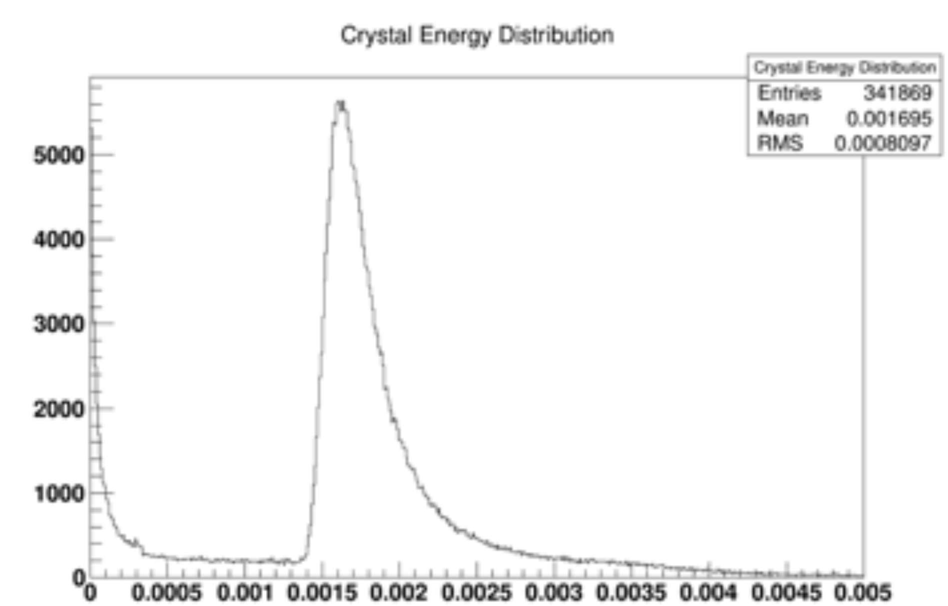


Figure is from Kyle

Issue 2: Threshold-Crossing Sample

- In the 2016 readout system, the threshold-crossing sample is a part of NSB, but actually it is a part of NSA.
- Comparison between two cases, the length of integration range is the same, but the range shifts one clock-cycle.
- Effects on 2016 MC is estimated to be slight
  - The issue has no effects on ADC readout, while only affects the trigger system, where hits after integration are applied to construct GTP clusters.
  - Change of integration result, which is caused by one clock-cycle shift, is very small.
    - The three-pole function is applied for conversion to a voltage pulse

      $V(t) = \frac{1}{2}t^2 e^{-t}$ (t >= 0 and denotes clock-cycle #)
    - Integration from t_i to t_f

      $f(t) = \int V(t)dt = -(\frac{1}{2}t^2 + t + 1)e^{-t}$

      $\int_{t_i}^{t_f} V(t)dt = f(t_f) - f(t_i)$

- For 2016 MC, NSB = 5 and NSA = 25.
- Clock-cycle # of threshold crossing sample depends on energy of hits. Generally, it is less than NSB, so $f(t_i) = f(0) = -1$ with or without one clock-cycle shift
- t_f >= NSA, so $f(t_f) \approx 0$

Issue 3: Deadtime for Pulse Integration

- Issue: Deadtime for pulse integration is 32 ns for hardware. But in the code, we use deadtime as 32 clock-cycles.
- Logic in hardware: If a threshold crossing sample is detected at time T, another threshold sample cannot be detected crossing until T + 32 ns and there would have to be an actual threshold sample crossing at T + 32 ns for that to work (e.g. if the pulse goes over threshold and is still over threshold 32 ns later it will not trigger another detection of the pulse).
- Deadtime is programmable in the digitization driver, so there are three cases :
    - NSA < deadtime / 4: An integration stops at the end of NSA, and then reach deadtime. After deadtime, a new integration can start.
    - NSA == deadtime / 4: An integration stops at the end of NSA and deadtime, and a new integration can start immediately after that.
    - NSA > deadtime / 4: When an integration has not completed, a new integration may start between the end of deadtime and the end of NSA (if a sample is over threshold, while its previous sample is lower than threshold)
- How about effects on 2016 MC?
    - Interval of signal events is set as 500 ns (i.e. 125 clock-cycles), so effects on hits by signal events should be negligible.
    - Effects on hits by beam background events?

New Readout Driver Development for 2019 MC

- HodoscopePatternReadoutDriver
- SinglesTrigger2019ReadoutDriver
- PairsTrigger2019ReadoutDriver
- TriggerModule2019: updated based on TriggerModule for 2016; the class handles trigger cuts for 2019

HodoscopePatternReadoutDriver (1)

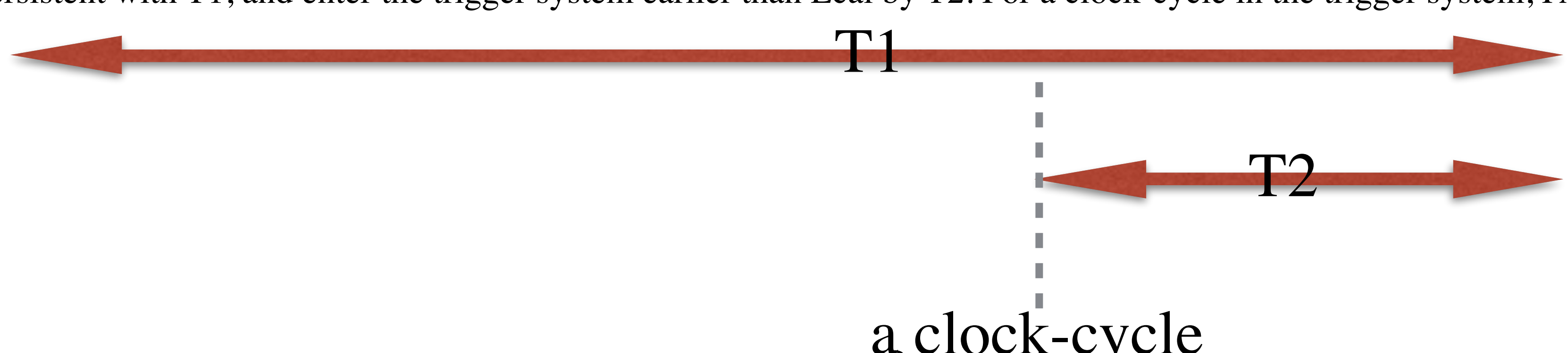
- The hodoscope logic in the trigger system:

1. Time-stretch hodoscope hits from the FADC so they persist long enough to be in time coincidence with other correlated hits in the hodoscope as well as with clusters in the ECAL
2. Cut FADC hits – this is after the FADC TET cut and gain, so it is the first cut that can be made on raw hodoscope hits to reject noise, dark current pulses, and/or low energy background.
3. Hits that have been time-stretched and hits discriminated by the previous 2 steps are then combined to form total energy deposited in each of the individual hodoscope tiles and hodoscope clusters.

    - **Hodoscope Tile Hits**
      All channels of the same hodoscope layer and tile are added together so that the VTP has an energy for each tile that is updated every clock-cycle. Tiles are considered to be hit for any clock cycle that the total tile energy >= hodoscope tile threshold. This produces a 5-bit vector indicating which tiles are hit (so there are 4 sets of these arrays corresponding to layer 1,2 and top, bottom).

    - **Hodoscope Cluster Hits**
      Another 4-bit vector is created (4 sets as well corresponding to layer 1,2 and top, bottom) that indicate hodoscope clusters (e.g tile 1+2, 2+3, 3+4, and 4+5). When two adjacent tiles have energy deposited (can be above or below the hodoscope tile threshold) and their sum is >= hodoscope tile threshold then the corresponding hodoscope cluster bit is set.

- For each layer, there are a 5-bit vector for tile hits and a 4-bit vector for cluster hits. So a 9-bit pattern with hit status is created for each layer and updated every clock-cycle.
- Hodo. FADC hits is persistent with T1, and enter the trigger system earlier than Ecal by T2. For a clock-cycle in the trigger system, FADC hits in a time range are considered.



HodoscopePatternReadoutDriver (2)

Logic for building patterns at each clock-cycle in the driver:

1. At a clock-cycle t, collect all FADC hits in the range [t + T2 - T1, t + T2]
2. For each channel of each layer, pick up FADC hits (>= TET) from the collection and save their energy in a list
3. Extract maximized energy in the list for each channel of each layer
4. For a layer, build a pattern using information of the maximized energy for channels of the layer.

SinglesTrigger2019ReadoutDriver

- Comparing to the 2016 singles trigger, there are two more limits for the 2019 singles trigger
  - Cluster position-dependent cuts

$$x_{cluster} \geq x_{min}$$

$$E_{cluster} \geq C_0 + C_1 x + C_2 x^2 + C_3 x^3$$

  - Geometry matching between Ecal-Hodo. The hodoscope patterns are used in algorithm for matching
- Parameters for singles trigger are reset based on the DAQ configuration.

PairsTrigger2019ReadoutDriver

- In view of software, the pairs trigger driver for 2019 is the same as the one for 2016.

- Parameters for pairs trigger are reset based on the DAQ configuration.

Other Trigger Drivers

- FEE trigger driver: In the DAQ configuration, there is prescale setup for various regions of Ecal. For MC, we should keep all events, so do not need to set prescale in the trigger system

- Cluster multiplicity driver

- Development of these two drivers is in the plan list.

Readout Steering Files for 2019 MC

- Three readout steering files with full chain have been developed.
    - Singles trigger: PhysicsRun2019TrigSingles.lcsim
    - Pairs trigger: PhysicsRun2019TrigPairs.lcsim.
    - Pulse trigger: PhysicsRun2019TrigPulse.lcsim
- Parameter setup in the files are based on the DAQ configuration hps_trigger_v12_1.cnf

Branches for Updates in hps-java

- My updates shown in the above slides are in iss655
- Omar's updates for SVT readout are in iss652
- All updates are merged in iss677, which will be used for producing pass0 of 2019 MC data

Question and Discussion

- What's trigger we use for pass0 of 2019 MC? Singles 3 trigger with Ecal-hodo matching or pairs 1 trigger for ap?
- Should we re-produce 2016 MC from the readout level for all samples due to the issues mentioned above? Effects are estimated to be slight. It is better to do reproduction, but time needs to be considered.
- An extra issue, not urgent to be fixed:
  - For the case NSA > deatime/4, a new integration could start between the end of deadtime and the end of NSA. In the code, the current integration stops immediately once a new integration starts, but actually, the current integration should continue to NSA.
  - The issue is not so easy to be fixed since the digitization driver is designed for integration one by one.
  - It rarely happens that multiple integrations are on-going at the same time on a single channel. Even if it happens, the current code just causes a slight difference for the integration value according to our calculation about the three-pole function model. Therefore, the issue's effects on MC data is negligible. For now, the updated readout system is good enough to be applied.
  - Another branch will be built to fix the issue later. Probably, a surgery for the digitization driver is needed.

Summary

- The 2016 readout system has been carefully re-checked. Some issues are fixed.

- Some new readout drivers and support classes for 2019 MC have been developed and tested well.

- Updates of SVT readout driver have been done by Omar.

- Three readout steering files with different triggers are developed.

- The readout system is ready for 2019 MC and has been used to produce large-scale samples.

Backup Slides

Scheme of System

Event (every 2 ns):

EcalHits HodoscopeHits TrackerHits TrackerHits MCParticle Manager

HitProcessing

Digitization Digitization SVT Readout

RawConverter RawConverter

GTP cluster HodoscopePattern

trigger