EDTM Study Report

Jacob Murphy

2022/06/28

Over the course of the 2021 Pion LT experiment, several studies were conducted to examine strange behavior in the Electronic Dead Time Measurements (EDTM). For physics production in this experiment, EDTM worked as expected. However, in calibration runs without coincidence triggers, Total Live Time (TLT) calculations were found to be greater than 1, leading to several studies.

Branches Used for Live Time Calculations:

- T.coin.p(h)EDTM_tdcTimeRaw
- fEvtHdr.fEvtType
- T.coin.pTRIG()_ROC1(2)_tdcTime
- H.bcm4a.AvgCurrent
- P.BCM4A.scalerCurrent(Charge)
- P.1MHz.scalerTime
- P.pTRIG.scaler
- P.EDTM.scaler
- P.pL1ACCP.scaler

1 TDC Timing

In the T tree, for a given replay, there are a series of TDC variable branches. For each trigger, there are TDC branches for ROC1 and ROC2. For each of these, there is a time and time-raw (or raw time and non-raw time). The raw time is not reference-time subtracted and has channels with each channel equating to a fraction of a nanosecond. In the non-raw time, the TDC spectrum IS reference-time subtracted. Within each trigger's non-raw time spectrum, there should be a self-tming peak that corresponds to events of that trigger type. Cutting along this peak allows for selecting events from individual triggers.

There are additional TDC branches for EDTM events. In the COIN DAQ, EDTM events from either arm are combined into one histogram and saved. There is a location for each arm (pEDTM and hEDTM). Paralleling the other TDCs, there are also time and time-raw branches. In the non-raw time, there is indeed a peak containing most EDTM events. The events, or grass, outside of this peak are dependent on delays in the events being received due to front-end build-up. This increases with higher rates. Typically, the SHMS has higher rates than the HMS and therefore there is higher 'grass' in the pEDTM non-raw time as compared to hEDTM for the same run. This is also a sign of some electronic dead time. The events outside of the non-raw time peak are NOT to be cut. Attempting to do so results in severely low and unrealistic live times, indicating there are valid and good EDTM events in the 'grass'. Additionally, grass-events in one arm are often within the non-raw time peak of the other. Therefore, cuts along the non-raw time for selecting EDTM events are not beneficial.

The EDTM TDC raw time spectrum is not reference-time subtracted, which is very important. Because EDTM events are always of every trigger type, they appear in the self-timing peak of every pTRIG TDC non-raw time spectrum. To select EDTM events from specific triggers, the self-timing peaks cannot be used. An alternative is using these raw time EDTM spectra. Each trigger has a range of acceptance. The width is well-defined to be 300 channels for every trigger, corresponding to the 20 ns window of acceptance for each trigger. Additionally, the higher the channel, the earlier the event arrived. This allowed for conformation of the timing-order for the triggers in this experiment:

- pTRIG 6
- pTRIG 5
- pTRIG 3
- pTRIG 1
- pTRIG 2
- pTRIG 4

Unfortunately, this TDC spectrum is also limited in allowing for cuts on EDTM. The COIN, 3/4, and ELREAL triggers overlap within each pair. To properly extract the trigger origin of a given EDTM event, we must rely on event types.

2 Event Types

When Brad setup the trigger apparatus for this experiment, he enabled three trigger bit types:

- Bit 1: SHMS
- Bit 2: HMS
- Bit 3: COIN

These bit-types then feed into the seven event types found in the event handler:

- Event Type 1: SHMS (Bit 1)

- Event Type 2: HMS (Bit 2)
- Event Type 3: SHMS+HMS (Bit 1+2)
- Event Type 4: COIN ONLY (Bit 3)
- Event Type 5: SHMS COIN ONLY (Bit 1+3)
- Event Type 6: HMS COIN ONLY (Bit 2+3)
- Event Type 7: ALL COIN (Bit 1+2+3)

Event types 3-7 all involve multiple triggers, though not necessarily multiple bit types. If an event passes through trigger logic AND meets the prescale condition, it is then given a bit type. If only pTRIG5 were enabled, all events would be Type 4 with Bit value of 3. Though the events had to have passed the pTRIG1 and pTRIG4 trigger logic to become a COIN, the prescale condition is never met. If, however, the single-arm triggers were not prescaled at all either, then COIN events would all be type 7. With prescaled single arms, COIN events are a mixture of 4-7, with 5-7 being far less likely the higher the prescales. Event Type 3 only occurs in single-arm running or a prescaled COIN trigger. Applying cuts along these event types, we can determine which trigger an EDTM event passed through. Differentiating between the COIN Event Types (4-7) is not consequential because all EDTM are COIN when COIN triggers are enabled (see 4). For Event Type 3, whatever is the earlier single-arm trigger that is enabled, then it will save all of that event type. For example, in Luminosity and HeeP Singles runs, we enable pTRIG2 and pTRIG4 only. Because pTRIG2 is the earlier trigger, Event Types 1 and 3 are SHMS and only Event Type 2 is HMS. With this information, the combined EDTM events in the COIN DAQ can be separated into individual arm EDTM events.

3 EDTM Sent and Accepted

Before moving on to define the Total Live Time, it is important to understand how EDTM Sent and EDTM Accepted are defined. The 'sent' value refers to how many EDTM pulses were generated by the pulse clock during the experiment run. This value is tracked by the EDTM scaler variable in the TSP and TSH trees, where we typically use the former. We also apply current cuts to this scaler variable to trim out beam trips and ramps, where charge accumulated is unreliable, and beam downtime, where the TLT is effectively 100% artificially. At present, we use the BCM4A variable to determine the current cuts. The scaler events, which iterate every 1000 trigger events or 2 seconds, are a running counter for charge, time, and scaler events. To apply current cuts to EDTM sent, we must subtract from the end-of-run total the sum of EDTM scaler counts from all cut scaler events. This final count is what is used as $EDTM_{sent}$ or $EDTM_{scaler}$.

EDTM	pTRIG1	pTRIG2	pTRIG3	pTRIG4	pTRIG5	pTRIG6
Channel Start	3100	2755	3250	2600	3650	3810
Channel Stop	3400	3055	3550	2900	3950	4110

Tab. 1: Channel Ranges in T tree for pEDTM TDC Time Raw

For EDTM accepted, we use the raw time EDTM TDC histograms. All events where the raw time is 0 are cut. Additionally, current cuts are again placed as before, now using the T tree's BCM4A variable. Cuts are then placed in the 300-channel width regions corresponding to all active triggers. These regions are defined as shown in Table 1. Finally, cuts are placed along event types. For PionLT single-arm calibration runs, only pTRIG2 and pTRIG4 were used. In this instance, Event Type 2 is associated with pTRIG4 and Event Types 1 AND 3 are associated with pTRIG2.

4 Prescaling

An event passing through the trigger logic is saved by the DAQ only if two conditions are met:

- Prescale conditions are met
- DAQ is not busy

The DAQ is busy when an event is being saved, hence only one event may be saved at a time. This is the CPUDT. If multiple triggers are met at the same time, the timing order will determine which is saved. Note that the COIN triggers are intentionally given the highest priority by Brad.

If an event is triggered within the time window of the DAQ saving the previous event, it will not be recorded (hence the dead-time). Physics events are effectively random in timing, so the EDTM can be used to track this deadtime as its origin is a pulse clock.

The issue with EDTM and prescaling is that the EDTM pulses through every trigger at once. Because of the trigger timing, only one event is ever recorded per pulse (this is debatable from EDTM study 3 where triple pulses were recorded, but it was never reproducible). This means that, if prescale conditions are met, the earliest trigger always records the EDTM event. If the earliest trigger's prescale condition is not met, then and only then can the DAQ save the EDTM pulse into the next-timed trigger (assuming that trigger's prescale conditions are also met). In other words, only EDTM events that are 'rejected' by eariler trigger prescale conditions can be recorded by later triggers.

This issue is avoided, however, by our physics production. A COIN trigger is always used in production, and it always has a prescale of 0, meaning every event is accepted to the DAQ. This means the EDTM pulses can never be registered by an trigger other than the COIN trigger. It also means that any difference between the accepted and sent EDTM events is due to dead-time. This is not the case in singles-running.

During HeeP singles and Luminosity scans, there is no COIN trigger. Usually, the triggers that are used (pTRIG2 and pTRIG4) will be prescaled. From how our calibration runs are set, almost always the SHMS trigger, pTRIG2, is at a higher rate and higher prescale. It is also the earlier trigger. As discussed earlier, the later triggers only can save EDTM events that are rejected by the earlier triggers. Therefore, pTRIG 4 will only save EDTM events when pTRIG2's prescale condition is not met.

The first issue with this system is that the majority of the EDTM pulses will be prescaled away (assuming a factor grater than 1 is used). To compensate for this, Brad designed the EDTM GUI to set the EDTM scaler (sent) rate to be equal to the desired rate multiplied by the lowest prescale factor. In physics production, the lowest prescale factor is always 1 so EDTM desired and sent rates are equal. For singles runs, these rates are rarely equal.

This means that if we simplistically say:

$$TLT = \frac{EDTM_{acc}}{EDTM_{sent}} \tag{1}$$

then our TLT will be abysmal thanks to the prescale conditions. A prescale-correction must be applied to account for this, but it is not a simple as using the lowest prescale factor. EDTM events will pass through both trigger's prescales, unless the earliest trigger is not prescaled. TLT is then given by:

$$TLT^{\#} = \frac{EDTM^{\#}_{acc}}{C^{\#} * EDTM_{sent}} \tag{2}$$

where # refers to $pTRIG\#$ and $C^{\#}$ is a correction factor to account for prescaling. The value of $C^{\#}$ is dependent on the prescale factors $PSF^{\#}$ for $pTRIG\#$ and all earlier triggers. Note that $EDTM_{sent}$ is NOT dependent on the trigger.

Assuming that there is only one active trigger per Bit type:

- $$C^6 = \frac{1}{PSF^6} \tag{3}$$
- $$C^5 = \frac{1}{PSF^5} \tag{4}$$
- $$C^3 = \frac{1}{PSF^3} - \frac{1}{PSF^3 * PSF^6} \tag{5}$$
- For a 3/4 run:

$$C^1 = \frac{1}{PSF^1} - \frac{1}{PSF^1 * PSF^3} - \frac{1}{PSF^1 * PSF^6} \tag{6}$$

For a normal production run:

$$C^1 = \frac{1}{PSF^1} - \frac{1}{PSF^1 * PSF^5} \tag{7}$$

- $$C^2 = \frac{1}{PSF^2} - \frac{1}{PSF^2 * PSF^3} \tag{8}$$

- For Electron Singles:

$$C^4 = \frac{1}{PSF^4} - \frac{1}{PSF^4 * PSF^2} \tag{9}$$

For Normal Production:

$$C^4 = \frac{1}{PSF^4} - \frac{1}{PSF^4 * PSF^1} - \frac{1}{PSF^4 * PSF^5} \tag{10}$$

It is important to note that these correction factors are due to the EDTM pulse affecting every trigger at once. Each EDTM pulse is one 'real' event, so as long as the event is recorded in one of the triggers, there is no deadtime. This behavior is the same as with a COIN trigger, which does not save the single-arm versions of its event.

5 Errors

A reoccurring issue we found was $TLT > 1$ in single-arm analysis. The correction factor $C^{\#}$ has improved our calculations immensely. The final step is to apply appropriate error to the TLT. Using binomial errors (and Bill Henry), the error can be written as:

$$\frac{\delta TLT}{TLT} = \frac{\sqrt{EDTM^{\#}_{acc} * (1 - \frac{EDTM^{\#}_{acc}}{EDTM_{sent}})}}{C^{\#} * EDTM_{sent}} \tag{11}$$