Baseline Vs High Rate Runs (Pulse Generator)
Effective deadtime and Tkr Trigger Combinations

Trigger Rates for 1-hour runs:

<table>
<thead>
<tr>
<th>Rate (Khz)</th>
<th>Num total Events</th>
<th>Tkr Exclusive</th>
<th>Pulse Exclusive</th>
<th>Tkr Inclusive</th>
<th>Pulse Inclusive</th>
<th>Tkr &amp; Pulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulse 20 Khz</td>
<td>126846</td>
<td>98575</td>
<td>24810</td>
<td>101139</td>
<td>27259</td>
<td>2320</td>
</tr>
<tr>
<td>Pulse 10 Khz</td>
<td>154557</td>
<td>121021</td>
<td>31096</td>
<td>122427</td>
<td>32282</td>
<td>1110</td>
</tr>
<tr>
<td>Pulse 1 Khz</td>
<td>172199</td>
<td>134922</td>
<td>35296</td>
<td>135761</td>
<td>35832</td>
<td>509</td>
</tr>
<tr>
<td>Baseline</td>
<td>191586</td>
<td>150212</td>
<td>39725</td>
<td>150677</td>
<td>39830</td>
<td>94</td>
</tr>
</tbody>
</table>

- Pulse triggered events are pre-scaled, i.e. just a small fraction of those events get recorded. Nevertheless, the instrument triggers and reads out.

- With high rates, a cosmic particle is more likely to find a busy instrument. Thus, the fraction of time the instrument is dead increases.

Assuming that a single trigger produces a deadtime of 26 μs, one can calculate the expected fraction of time the instrument is dead, having in mind that the pulse generator is Poisson-like and that the uncertainty of the nominal rate is 3%:

<table>
<thead>
<tr>
<th>Rate (Khz)</th>
<th>Expected (%)</th>
<th>From Data (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6 + 0.1</td>
<td>4.3</td>
</tr>
<tr>
<td>5</td>
<td>12.2 + 0.3</td>
<td>13.7</td>
</tr>
<tr>
<td>10</td>
<td>22.9 + 0.6</td>
<td>22.2</td>
</tr>
<tr>
<td>20</td>
<td>40.5 + 1</td>
<td>35.7</td>
</tr>
</tbody>
</table>

From Data = 1 − \( \frac{Tkr_{inc} \text{ Triggers (@ X Khz)}}{Tkr_{inc} \text{ Triggers (baseline)}} \)
Trigger Request VS Readout
Case 1: Trigger Req has 3-in-a-row but readout doesn't

- Small effect

Events with Tkr Trigger (inclusive)

Events with Tkr Trigger (exclusive)

Tkr triggered events with additional pulse trigger are more likely to be inconsistent
Trigger Request VS Readout
Case 2: Readout has 3-in-a-row but trigger Req doesn't

Why does it get worse for top layers?
How can the pulse affect events that have pure tkr trigger?

Luis Reyes
Attempting an Explanation for events that have 3-in-a-row trgreq but no hits when a pulser trigger and a tkr trigger are present...

Considerations:

- Those events have 5+ (4 a very few times) hits somewhere in the tracker, it's not like the pulser is making up all the 6 requests necessary for the 3-in-a-rows...
- TEM diagnostics latching and readout are “almost” simultaneous...

Imagine a hit with TOT so small that with normal settings doesn’t produce a request or a hit (a 3-in-a-row happens somewhere else):

![Diagram]

Let's assume now that both tkr and pulse trigger are present, but the pulse happens first. With respect to the hit, the latching of TemDiag and Readout comes faster...

![Diagram]

→ Pulse trigger moves the trigger acknowledge forward
What about this?

Threshold

Tem

Readout

No TrgReq
Hit present

- Slow channels could be specially susceptible to this.

(Pulse trigger moves the trigger acknowledge forward)