

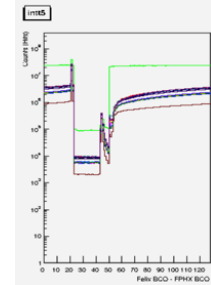
Software meeting yesterday

- I asked how I can make the BCO diff plot as the offline QA using the fully streaming data
- Two things
 - Try MVTX method. MVTX uses streaming mode only. (I will try)
 - Otherwise Fun4All may need to be modified slightly (Chris helps)

INTT QA plots for streaming mode

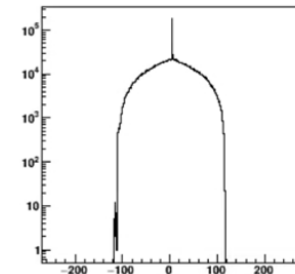
- BCO difference between GL1 and INTT hit
 - The plot is important to check how the INTT is sync'd with GL1
 - In the trigger mode, we checked it in the online monitor but cannot in the stream mode because no GL1 info in online
 - I made this by production GL1 and INTT DST separately and combine them later (using [TTree](#) for GL1)
- I would like to make this plot in the offline QA framework, but I cannot make this QA plot because of synchronization feature of Fun4All. Help needed!

Online Monitor in trig. mode



INTT HitBCO - GL1BCO

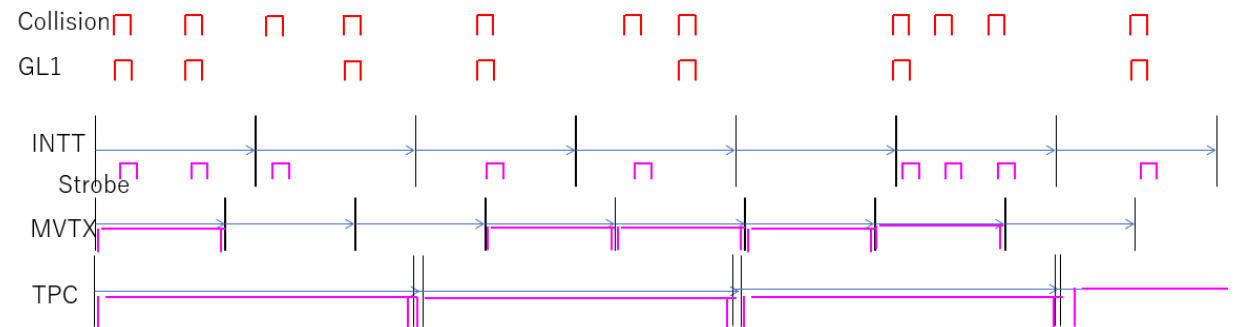
bcoreco diff_0



Streaming in the offline software

- I realized that the tracking group assumes data taken w/ the extended mode. No fully stream mode is expected by now
 - In Fun4All, the GL1 trigger is the driver to handle “event” and cover the data within the extended time
 - Data outside of the extended time is ignored
 - Need to modify the framework of “event” handling

BCO info used in Fun4All



- Fun4All has 2 places to use BCO info
 1. Combiner for event synchronization
 2. Time bucket in TrkrHitSet

How do we use them properly separately?

- Tracking group said that calling process_event take data in the time-frame (maybe after sync'd)
- Event combiner splits the event (process_event) based on the GL1 BCO (or INTT BCO)
- It seems that these two ideas are inconsistent (to me)

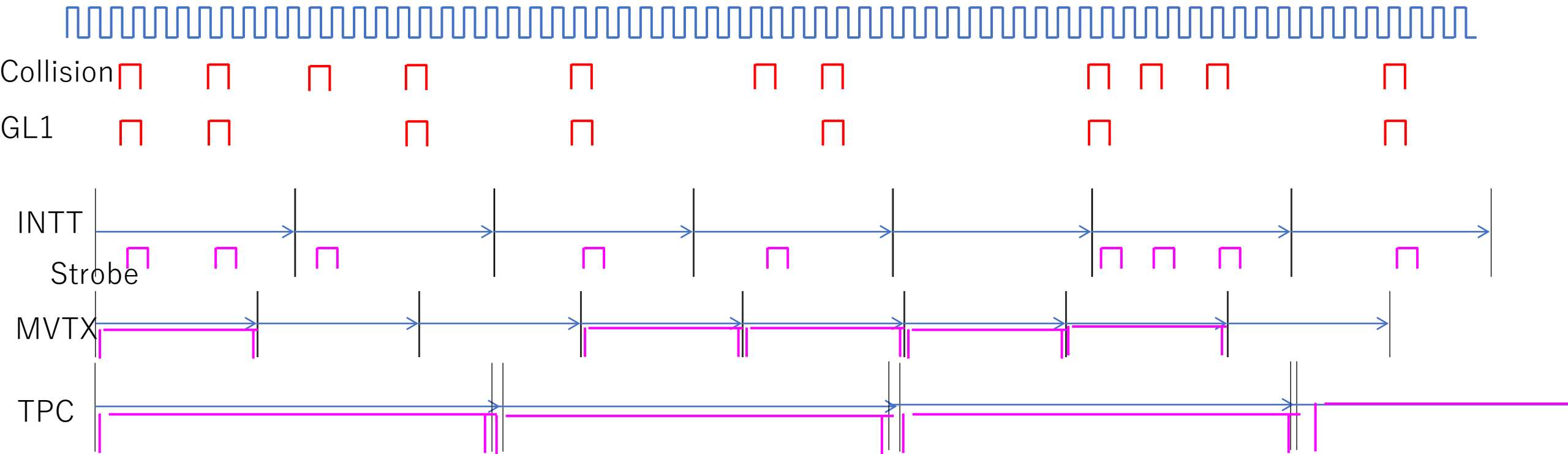
Issue: INTT production stopped ~70k events

- I tried to produce INTT DST but the production job stopped ~70k events. I tried multiple runs but all job stopped at the similar events.
- I ran w/ GDB. No error is caught
- I ran w/ Valgrind. It stopped (not finished/crashed yet). No error in the log

```
Reading symbols from .gnu_debugdata for /usr/lib64/libbz2.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libbz2.so.1)
Reading symbols from /usr/lib64/libz.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libz.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libz.so.1)
Reading symbols from /usr/lib64/libfontconfig.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libfontconfig.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libfontconfig.so.1)
Reading symbols from /usr/lib64/libfreetype.so.6...
Reading symbols from .gnu_debugdata for /usr/lib64/libfreetype.so.6...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libfreetype.so.6)
Reading symbols from /usr/lib64/libffi.so.6...
Reading symbols from .gnu_debugdata for /usr/lib64/libffi.so.6...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libffi.so.6)
Reading symbols from /usr/lib64/libfontconfig.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libfontconfig.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libfontconfig.so.1)
Reading symbols from /usr/lib64/librender.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/librender.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/librender.so.1)
Reading symbols from /usr/lib64/libcap.so.2...
Reading symbols from .gnu_debugdata for /usr/lib64/libcap.so.2...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libcap.so.2)
Reading symbols from /usr/lib64/liblz4.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/liblz4.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/liblz4.so.1)
Reading symbols from /usr/lib64/libcrypt.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libcrypt.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libcrypt.so.1)
Reading symbols from /usr/lib64/libpgp-error.so.0...
Reading symbols from .gnu_debugdata for /usr/lib64/libpgp-error.so.0...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libpgp-error.so.0)
Reading symbols from /usr/lib64/libdw.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libdw.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libdw.so.1)
Reading symbols from /usr/lib64/libcrypt.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libcrypt.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libcrypt.so.1)
Reading symbols from /usr/lib64/libattr.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libattr.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libattr.so.1)
Reading symbols from /usr/lib64/libelf.so.1...
Reading symbols from .gnu_debugdata for /usr/lib64/libelf.so.1...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libelf.so.1)
Reading symbols from /usr/lib64/libfreebl3.so...
Reading symbols from .gnu_debugdata for /usr/lib64/libfreebl3.so...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libfreebl3.so)
Reading symbols from /usr/lib64/libfreebl3.so...
Reading symbols from .gnu_debugdata for /usr/lib64/libfreebl3.so...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libfreebl3.so)
Reading symbols from /usr/lib64/libpthread.so.0...
Reading symbols from .gnu_debugdata for /usr/lib64/libpthread.so.0...
(No debugging symbols found in .gnu_debugdata for /usr/lib64/libpthread.so.0)
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib64/libthread_db.so.1".
0x00007fb3b2a67066 in __memmove_sse3 () from /usr/lib64/libc.so.6
(gdb) c
Continuing.
(gdb) █
good(const string &infile)
```

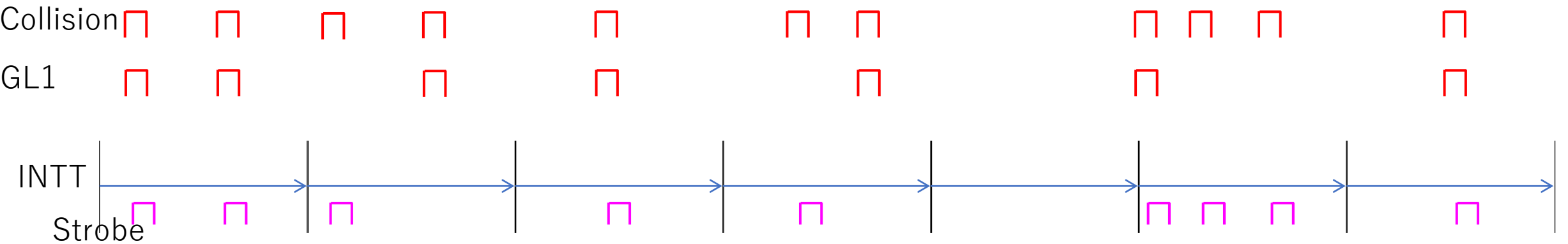
```
PHNodeReset: Resetting INTTRAWHIT
executing run for input master DefaultSyncManager
Fun4AllServer::run - processing event 78246 from run 47972
INTTRAWHIT, Node Count: 1
Writing Event for out
Fun4AllServer::process_event Resetting Event for Sync Manager DefaultSyncManager
PHNodeReset: Resetting DST
PHNodeReset: Resetting INTT
PHNodeReset: Resetting INTTRAWHIT
executing run for input master DefaultSyncManager
Fun4AllServer::run - processing event 78247 from run 47972
INTTRAWHIT, Node Count: 1
Writing Event for out
Fun4AllServer::process_event Resetting Event for Sync Manager DefaultSyncManager
PHNodeReset: Resetting DST
PHNodeReset: Resetting INTT
PHNodeReset: Resetting INTTRAWHIT
executing run for input master DefaultSyncManager
Fun4AllServer::run - processing event 78248 from run 47972
INTTRAWHIT, Node Count: 1
Writing Event for out
Fun4AllServer::process_event Resetting Event for Sync Manager DefaultSyncManager
PHNodeReset: Resetting DST
PHNodeReset: Resetting INTT
PHNodeReset: Resetting INTTRAWHIT
executing run for input master DefaultSyncManager
Fun4AllServer::run - processing event 78249 from run 47972
INTTRAWHIT, Node Count: 1
Writing Event for out
Fun4AllServer::process_event Resetting Event for Sync Manager DefaultSyncManager
PHNodeReset: Resetting DST
PHNodeReset: Resetting INTT
PHNodeReset: Resetting INTTRAWHIT
executing run for input master DefaultSyncManager
waiting for data (interrupt to abort)
```


Streaming mode



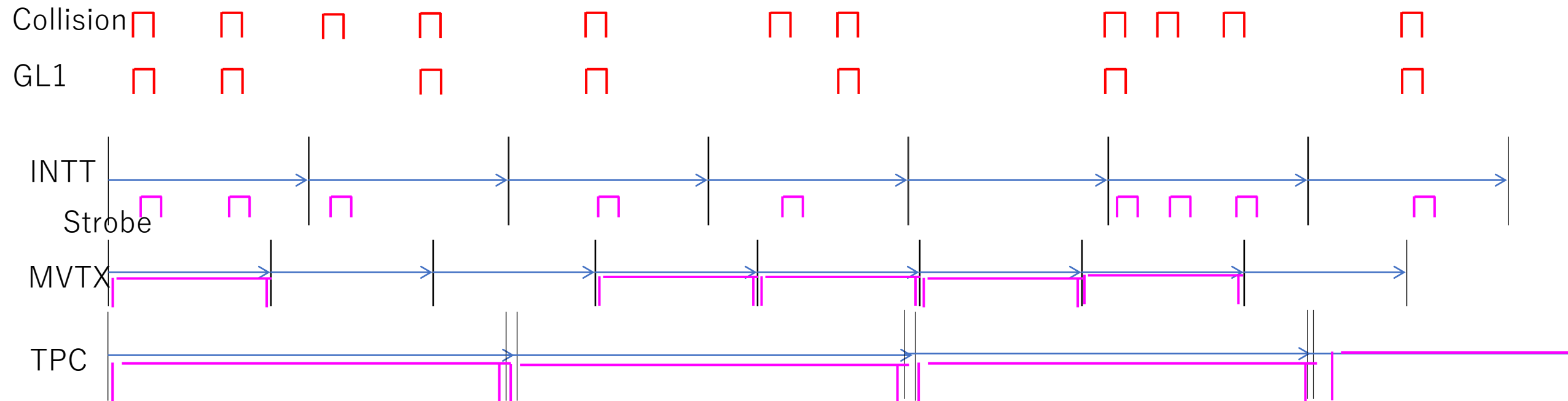
- 2 questions for synchronizing the detectors in Fun4All
 - Sync streaming detector (INTT) and GL1
 - Sync among streaming detectors

Sync streaming detector (INTT) and GL1



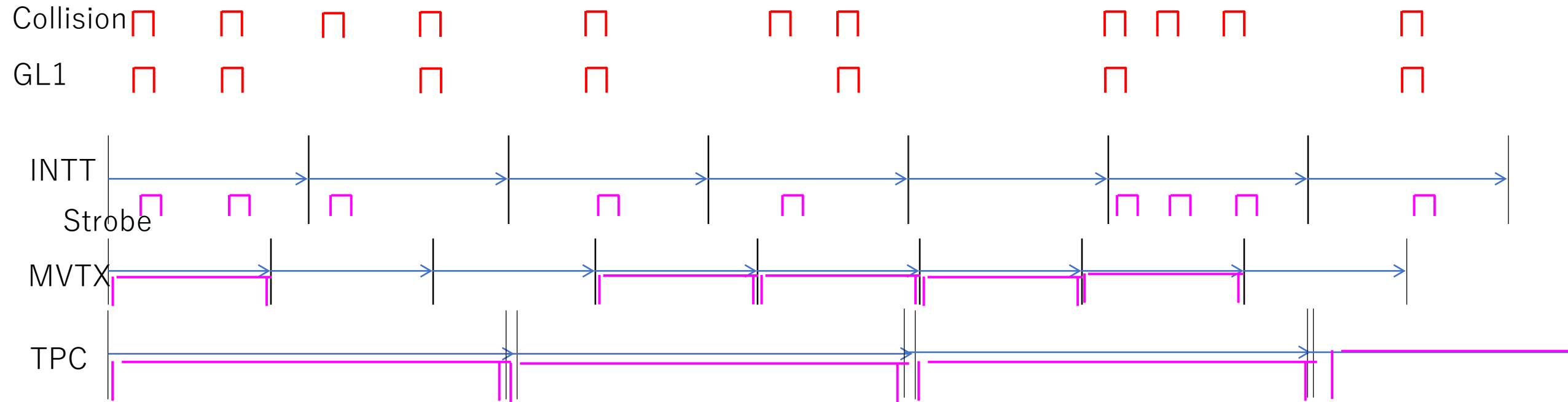
- For trigger mode, GL1 is the driver to sync
- For stream, INTT has more hits in BCO because of no deadtime. But GL1 finding efficiency is 95~97% in INTT stream data.
 - Last week, I talked w/ Chris and Jin and was suggested to use INTT BCO (w/ 7bit BCO addition) as driver instead of GL1BCO
- If we use INTT BCO as event driver, we lose 3~5% events

Sync among streaming detectors



- In streaming, we don't have the same heart beat among the streaming detectors
- INTT time frame (strobe cycle) is 120 RHIC clock (same as 1 RHIC cycle)
 - MVTX time frame is 10us,
 - TPC = 2 INTT time frame
- How is the time frame defined ?

BCO info used in Fun4All



- Fun4All has 2 places to use BCO info
 1. Combiner for event synchronization
 2. Time bucket in TrkrHitSet

How do we use them properly separately?

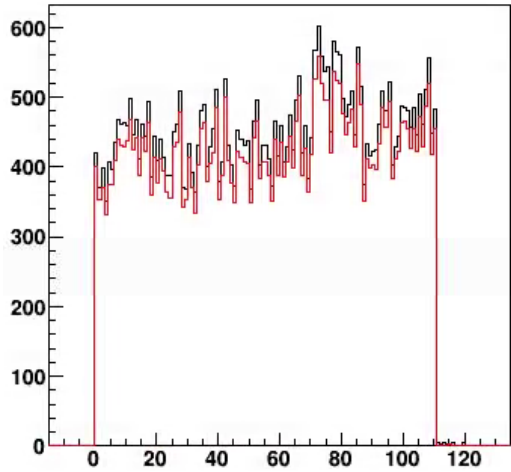
- Tracking group said that calling `process_event` take data in the time-frame (maybe after sync'd)
- Event combiner splits the event (`process_event`) based on the GL1 BCO (or INTT BCO)
- It seems that these two ideas are inconsistent (to me)

GL1 finding efficiency

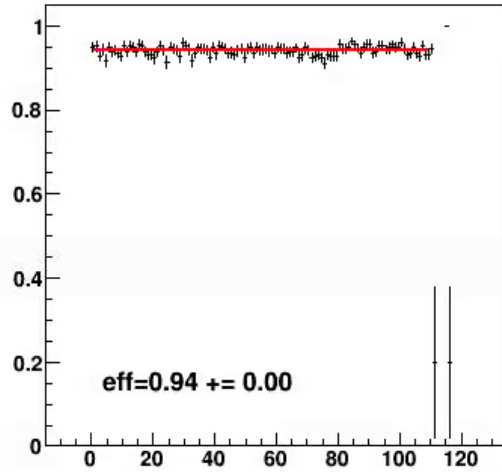
Streaming (run47977)

Triggered (run47982)

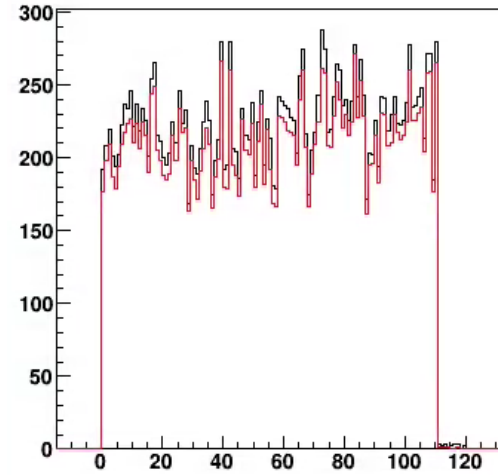
bunch @ gl1



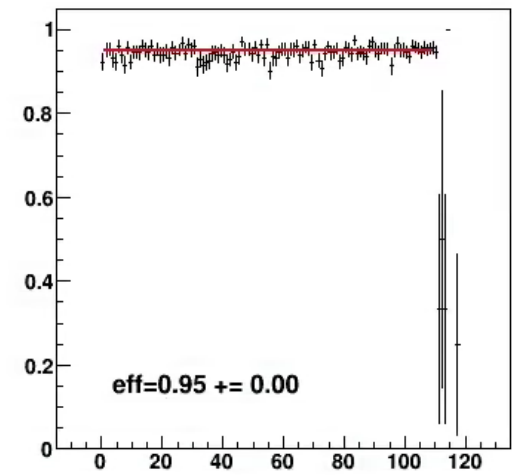
bunch @ evt all felix



bunch @ gl1



bunch @ evt all felix



- Finding efficiency is similar for both the stream and triggered
 - It looks that the stream mode works properly
 - Previous study : 97%. Why?
- Offset between GL1 and recoBCO is changed. Why?
 - previous offset was 19, now it is 23
 - If we find the reason, it can use it. Otherwise this should be a calibration parameter