

TTD software/firmware for non-stop DAQ

Mikihiko Nakao (KEK)

mikihiko.nakao@kek.jp

2022.11.30

Belle II Trigger/DAQ Workshop 2022

Disclaimer:

this talk is not so a productive one, but it may be useful to share our concerns...

Wish

● FEE error recovery without SALS

- **Same** run number, **increment** sub run number, event number just **kept incrementing**
- Original DAQ plan included such a wish
- DAQ system has been too complex and unreliable to work on it
- Now, this workshop is a chance to revisit (similar wishes also in subdetector talks)

● What can be actually done

- Recovery from ttlost / b2llost may be realized in a **O(1sec)** time scale
- Recovery from other FEE errors demands reprogramming and would take longer time of **O(1min)**
- Use **PAUSED** state like HV recovery — still some gain against full SALS

● What have to be involved

- **TTD** is the main player
- FEE errors are also detected as corrupt data at **readout PC**
- Therefore, the recovery sequence has to be handled by an upper level — **RUNCONTROL?**
- And **FEE** should be able to start from a non-zero event number

How to pause the run

● TTD processes

- Two processes are involved: **pocket_ttd** and **ttctrld**
- **pocket_ttd** sends FATAL log message to logcollectord
- **ttctrld** sets TTD state to ERROR

● Readout PCs

- Readout PC's **runcontrold** sets subdetector's state to ERROR
- TTD receives back-pressure and become BUSY
- I found that I don't know how the error is generated and propagated

● statft

- FTSW status may be ERROR, but it should look like PAUSED on statft
- One more register bit is necessary on FTSW

● COPPER, PCIe40

- Probably we don't need to consider COPPER
- Need a mechanism to flush the unprocessed event fragments

Resetting errors

● ttlost, b2llost

- **runreset** to all FEEs for simplicity

● SEU, data error, or most of other errors

- Firmware reprogramming script (e.g., CDC)
- Subdetector becomes LOCAL mode while reprogramming

● Hanging event fragments?

- Better to reset for PCIe40? Anyway, runreset signal will be delivered
- No need to reset for event builder or readout PC?
- Data-driven handling would be smarter than additional slow control procedure

● Local run during PAUSED?

- TTD does not care, so it should be possible

How to resume the run

● TTD actions

- TTD distributes the run number, incremented sub-run number and event number
- Firmware has to be corrected (current firmware cannot distribute non-zero event number)
- No change in the connection to EB

● FEE check item

- Any code that depends on starting from non-zero event number should be modified
- Would like to avoid any run-start special handling inside FEEs
- Starting from a multiple of 2^n may be helpful for some subdetector

Thoughts

● Is it really needed / wanted?

- "Restarting without incrementing run number" vs "Partial SALS with many short runs"?
- Partial SALS with new run number would be still possible without changing TTD programs
- Deadtime may be the same, if the short runs are not rejected by offline processing

● Work on TTD programs

- Current pocket_ttd and ttctrld program codes are pretty badly organized
- I failed to figure out how to implement PAUSED state while preparing this talk
- Probably we need an overhaul of the TTD programs in a similar way that FTSW firmware is being reorganized

● Schedule

- Starting this work before the end of this FY is not so likely
- Overhaul of the TTD programs would be benefit for a longer time span of Belle II data taking
- Need to revisit at future meetings