



Mikihiko Nakao (KEK IPNS / SOKENDAI)

`mikihiko.nakao@kek.jp`

2025.10.22

Belle II Trigger/DAQ Workshop 2025

List of TTD-related issues

Stock of spares and long term maintenance

FTSW patch — status and plan

Further known hardware-related issues

Non-stop DAQ related investigations

Possible improvements

Unified Trigger Timing Distribution (TTD)

127 MHz system clock (1/4 of SuperKEKB RF)

“b2tt” — custom 254 Mbps bidirectional serial protocol

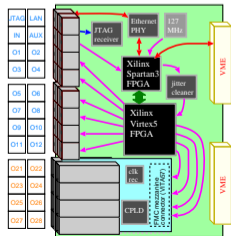
“FTSW” (frontend timing switch) — single PC-board for multiple TTD functions

Up to 20 RJ-45 / 8 optical output from 6U double-width VME

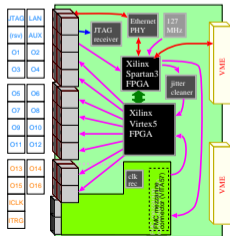
Common receiver firmware everywhere for Xilinx/Altera FPGAs

JTAG connections to frontend using RJ-45 ports

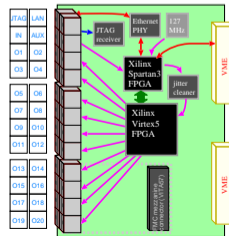
12x RJ45 + 8x SFP (Tx)



16x RJ45 + 2x SFP (Rx)



20x RJ45



FTSW modules

Module types

- Type S FTSW, version 2 and version 3 — 20 port RJ-45 output
- Type R FTSW and FTOR — 16 port RJ-45 / 2 port SFP card for optical receiver
- Type P FTSW and FTOP — 12 port RJ-45 / 8 port SFP (optical) distributor

Difference between FTSW version and types

- Clock output: version 2 from Virtex 5 FPGA, version 3 from Spartan 3 FPGA to reduce interference
- Several other minor differences, almost common firmware source files
- Type S, R, P can be interchanged by (re-)soldering the RJ-45 connectors

Special modules

- No 5V power connection for ECL, driven by 3.3V (6 FTSW2 + 6 FTSW3)
 - 19-inch rack mount front panel for PXD and TOP-JTAG (no VME crate, driven by 5V 2A AC adaptor)
- 2 PXD FTSW were changed to the front panel type in summer 2025**

Spare modules

- Currently no ready-to-go spare in the shelf
- Many type-S modules can be taken from E-hut COPPER crates (3 of them were already taken)
- FTSW stock management is not up to date — **plan: netbox to store module and connection info**

Long-term maintenance of FTSW modules

Last spare module production

- In 2019, 10 FTSW2 and 10 FTSW3 modules were produced
- There are still many of them not configured as spare
- JTAG does not work, reason not understood yet (no time yet for detailed investigation)

Top 3 vulnerable devices on FTSW

- Virtex 5 — 13 in stock, 17-week lead time for new procurement
- DC-DC converter — 20 in stock, not so clear if new procurement is possible
- Jitter cleaner — some in stock, usually easy to procure

Broken modules

- More than 5 broken modules in stock, half of them due to Virtex 5, the other due to jitter cleaner
- Possible to send to company for chip replacement

More production?

- Considering the retired modules from COPPER crates and repairable modules, no need for further production for the Belle II lifetime

FTSW patch for power-on issue

FTSW does not always correctly start upon power-on

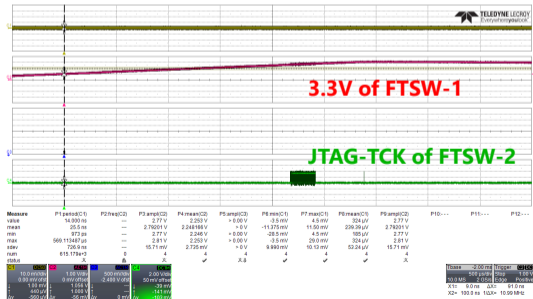
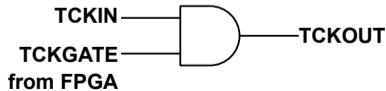
- JTAG related problem (need to reprogram FPGA)
- Jitter cleaner related problem (need to reprogram jitter cleaner)

Noise (many glitches) found on JTAG-line upon power-on

- When FPGA is not programmed, LVDS output for JTAG is undefined
- Just reprogramming FPGA does not cause the problem, so probably the glitch happens only when the supply voltage is marginal
- LVDS receiver then generates random **harmful** noise sequence

How to fix

- Disable JTAG-TCK line (**tckgate**)
- Enable only when FPGA is down or b2tt is down
- Replace TCK jumper with an active AND logic



More known issues of FTSW hardware

Jitter cleaner stops working

- Jitter cleaner is a programmable device with SPI (Serial Peripheral Interface)
- SPI line may be in an undefined state during power-on, and random noise sequence may reprogram the jitter cleaner into a wrong configuration
- Pulldown resistor on power-down line can shut down the jitter cleaner when FPGA is not programmed
(a new patch to add a pulldown resistor)

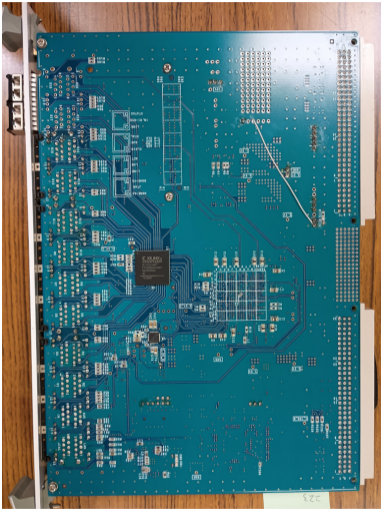
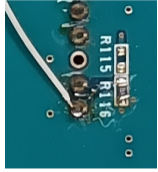
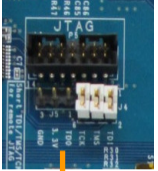
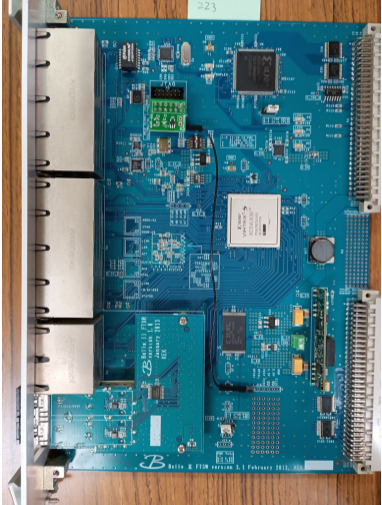
Other cases requiring power cycle

- Sometimes an unstable b2tt is not fixed by reprogramming the FPGA, but fixed by a power cycle.
— **mechanism not understood**
- Power cycle of single FTSW module without a power cycle of the entire crate would be convenient
- DC-DC converter has an ENABLE pin which could be negated by firmware, if it is connected to an FPGA pin **(a new patch connection)**

Unstable link / JTAG recovers by cable reconnection

- Why an unstable link recovers? — **mechanism not understood**
- Hypothesis: malfunction of IO buffer inside FPGA reset by an electric shock?

Patched FTSW



Status of FTSW patches

Patches completed for PXD, EKLM, ARICH

- 8 modules for BKLM were not made during this summer
- No plan yet for SVD, CDC, TOP, ECL
- **There was a power-on issue for TOP this summer** (but it was solved by cable reconnection)

Firmware modification

Software modification

- JTAG has to be enabled before checking the JTAG connection or reprogramming FTSW
- A new register to control it via regft
- ttaddr and jtagft are not needed to modify
- `jtag-chain-{eklm|ari}.sh` and `jtag-program-{eklm|ari}.sh` were modified
- Script cleanup and modification for all subdetectors to be done
- Then, it will be pushed into `ttt` repo

Operation tips

- On-detector FTSWs have to be turned on after E-hut FTSWs are turned on
- This is because the tckgate patch works only when b2tt link is up

On non-stop DAQ

Restarting a subrun without runreset will save large downtime

- Possible errors to be rescued — ttlost, b2lllost, ferr (?)

Error recovery mechanism should be different for different type of errors

- It is also important to separately consider the two directions of ttlost and b2lllost

Do we accept event skip in the back end?

- Life is easier if PCIe40 can correctly skip incomplete event and merge the event based on the event number instead of arrival order

In general, b2tt firmware is **not** designed to be non-stop DAQ ready

- So far, no attempt has been made to study the b2tt / belle2link firmware and make a possible plan to update

b2tt behaviour for ttlost

ttlost recovery (FEE to FTSW direction)

- FEE does not notice, error state inside FTSW can be cleared by errreset
- New subrun will always be successfully resumed
- It may not be necessary to stop the run from the beginning, if it is for a short period

ttlost recovery (FTSW to FEE direction)

- ttdown is detected in an 80-clock-long packet (up to $6\mu s$)
- FEE status is embeded in an 80-clock-long packet, and has to be decoded and encode at every step (3–4 steps \times 6–12 μs)
- 20–50 μs latency is comparable to the trigger interval at 30 kHz,
probability is high to get one trigger lost during error propagation

If one trigger is lost and ttlost is cleared

- b2tt delivers the last sent trigger (tagin)
- tagerr occurs when tagin is different from internal trigger count cnt_trig (aka trgtag) but need to wait for the tagin arrival within 1280 clock frame
- Before tagerr happens, a few triggers may be recorded with trgtag smaller by one
- There is a way to set tagtag to be tagin (tagset), by setting and releasing the trigger mask (trgmask)
- **Even if we use tagset, one trigger is anyway missing from the b2tt output**

b2tt behaviour for b2llost

b2llost recovery (PCIe40 to FEE direction)

- b2llost is detected by FEE and then propagated to FTSW
- No crucial data is usually sent from PCIe40 during data taking, hence no impact on resuming a subrun

b2llost recovery (FEE to PCIe40 direction, when the event is sent)

- b2llost is detected by PCIe40 and then propagated to FTSW
- PCIe40 will anyway receive a broken event — can it be thrown out and continue?

b2llost recovery (FEE to PCIe40 direction, when IDLE is sent)

- b2llost is detected by PCIe40 and then propagated to FTSW
- Trying to read the source code, but not yet successful to understand what happens

b2llost/b2llost is very poorly implemented

- b2llost is based on invalid K symbol (only)
- Nothing happens in belle2link even if CRC error is detected for both directions (!)
- Nothing happens in belle2link even if receiver FIFO becomes full (!)

It will be an interesting project to make belle2link more robust, with protocol upgrade in mind

More miscellaneous topics

CAT7 cable quality can be very poor

- There are poor flat CAT7 cable with electric resistance of a few to 10 ohms
- One of the lot that I bought several years ago was all bad
- Recently, KLM's short CAT7 cables seem to have the same problem (O(100) cables, different product)
- Better to check the cable resistance before using

Injection kicker signal test setup (upon request from ECL)

- Injection kicker signal is now an important handle to mitigate background in frontend
- Now one can generate a dummy kicker signal for test (need special privilege on ttd12 system)
- An example code is also prepared in ftsw_test repo (ft3f_injtest directory)

30 kHz limit on ttd11 VME read

- Probably limitation in VME cycle, but it's only 180 kHz × 32-bit read (6 word / event)
- If the user/kernel space switching is a bottleneck, one can largely improve by introducing a new system call to read 6 words in one system call
- **An interesting small project?**

ttd11 SSD project

- NFS is slowing down the user experience of ttd11, which will be improved by booting from SSD

Summary

**TTD / FTSW / b2tt may be a matured project,
but there are still interesting topics to work on**

**In particular, probably it's good time to seriously
think about Belle2link overhaul**

**More topics related to FTSW version 4 and b2tt overhaul
to be covered in a separate talk**