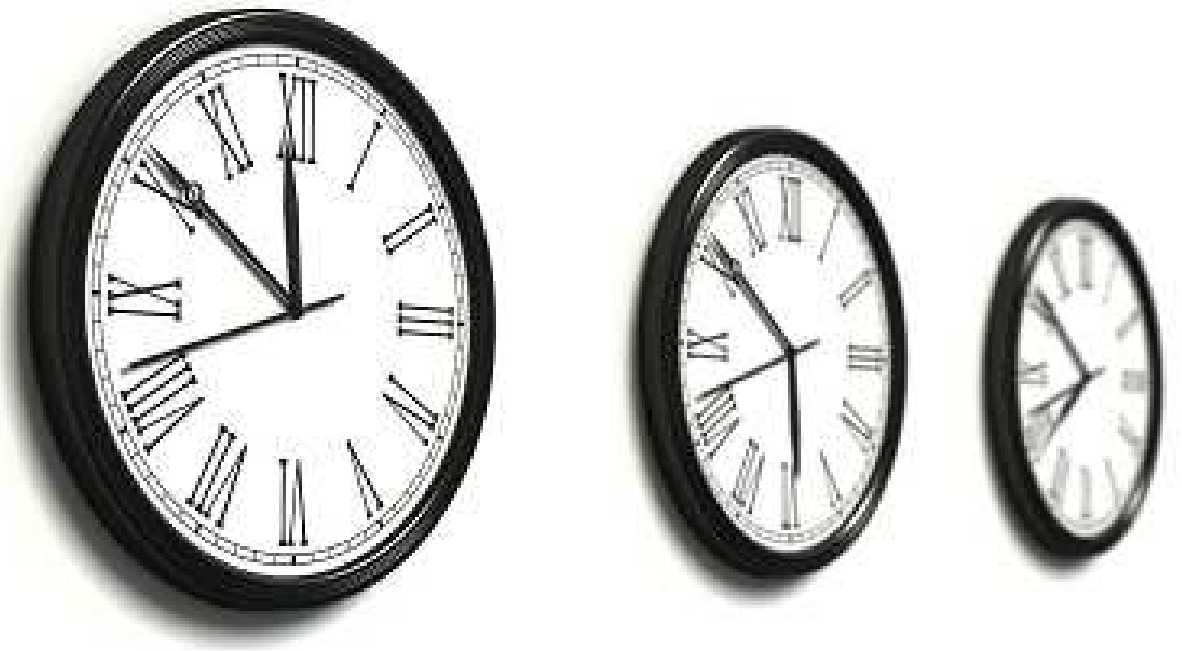
# TTD: event / run / time definitions

Mikihiko Nakao (KEK-IPNS)

mikihiko.nakao@kek.jp

*October 23, 2013*
*4th VXD workshop*
*at DESY*



# Belle II Trigger Timing Distribution

**Glossary:**

**TTD** (Trigger Timing Distribution)
name of the **system** — in DAQ group's private domain

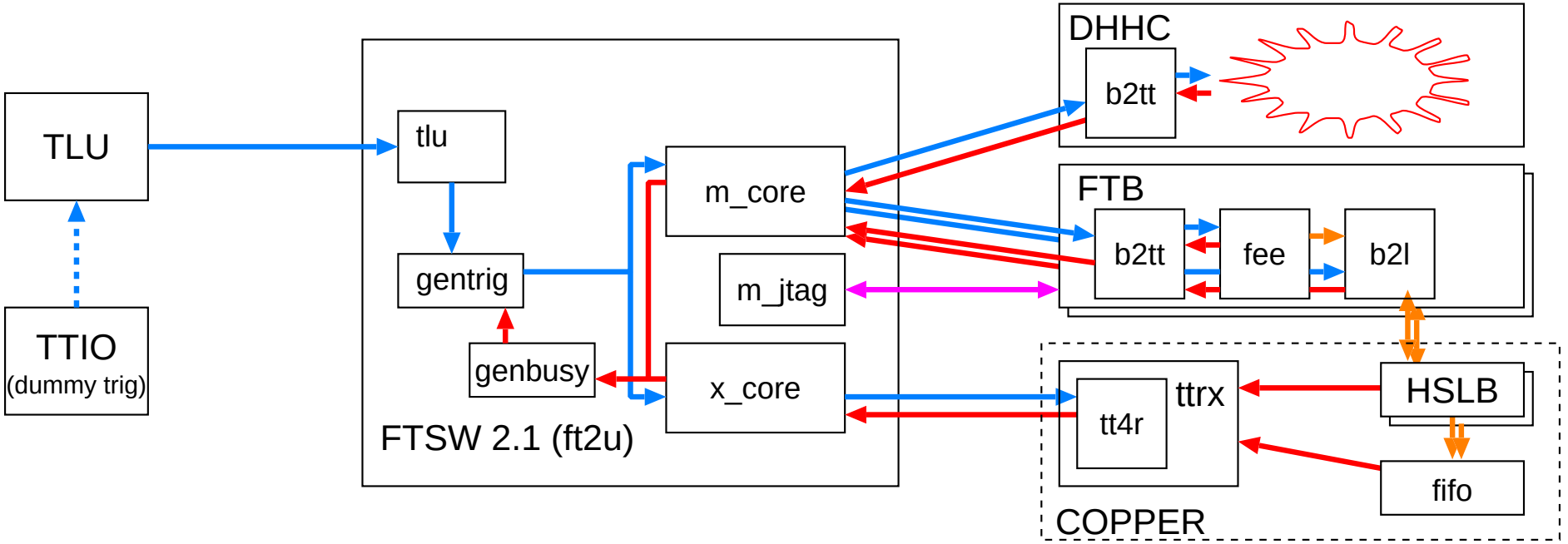
**b2tt** (Belle 2 trigger timing)
name of the **protocol** — announced to daq2_ml on 2013.10.16

**FTSW** (Frontend Timing SWitch)
name of the **VME module** — provided to people who requested

**ft2u** (FTSW version 2 user firmware)
name of the **firmware** on FTSW — announcedd to daq2_ml on 2013.10.22

*Igor came to me asked: "did you change the name of the system?"*
*— my answer: "no, each part has a different name to avoid(?) confusion"*

# TTD for telescope test




- **Hardware (FTSW version 2.1)** is unchanged since 2011
- **Firmware (ft2u/b2tt)** is ready including connections to TLU, DHHC, FTB, and minor problems (if found) can be quickly fixed (also remotely)
- **Software (pocket_ttd)** — NSM based control / data-acquisition exists, used and tested at Spring-8 CDC beam-test, lots of improvement to be added

# TTD providing run/event/time info at b2tt

```
entity b2tt is
  port (
    -- system clock and time
    signal sysclk   : out std_logic;
    signal utime    : out std_logic_vector (31 downto 0);
    signal ctime    : out std_logic_vector (26 downto 0);

    -- exp- / run-number
    signal exprun   : out std_logic_vector (31 downto 0);

    -- run reset
    signal runreset : out std_logic;

    -- data for Belle2link header
    signal fifordy  : out std_logic;
    signal fifodata : out std_logic_vector (63 downto 0);
    signal fifonext : in  std_logic;
      : );
end b2tt;
```

# Trigger number

- **XXX → 0:** `runreset` to clear the event number
  - 1-clock wide pulse
  - always at the fixed position within the revolution cycle
  - **available for users, action has to be completed within 10 $\mu$s**
- **0:** First trigger (trigger number 0) is generated with trgtyp = 0xf at run start (begin-of-run trigger) regardless the trigger source
- **1 → XXX: b2tt** provides a 32-bit trigger number, counted inside b2tt
- b2tt trigger number is compared with FTSW event number every 10$\mu$s
- b2tt trigger number is attached to Belle2link header
  - Time the trigger is received and time the header is sent are different
  - Event info are once stored in FIFO (using 1 FEE's RAMB18)
- Current CDC Belle2link firmware sends only 16-bit trigger number to HSLB (will be increased to 32-bit)

# Trigger time

- **Time** (32-bit utime + 27-bit ctime) info is attached to every event
- **Trigger number can be faked by a process and still can match the other, while trigger time is not easy to generate at match**
- **synchronization** — all b2tt receivers are synchronized to central FTSW in such a way that the same trigger has the same time
- **utime** — Unix-time in second, now automatically set when FTSW firmware is loaded by `bootft`
- **ctime** — Clock within a second, up to 127,215,999
- Minor problems:
  - Time is screwed up if clock source is screwed up
    (happened at Spring-8 when clock is given by TRG group)
    $\Rightarrow$ a slow system should be checking it time to time
  - utime count-up is a bit delayed in the current b2tt, need to be fixed

# Run number

- A 32-bit word, subdivided into experiment number and run number
  - In current Pocket DAQ system it is 10-bit exp + 22-bit run number
  - Proposal:
    - 10-bit experiment number (up to 1,023)
    - 14-bit cold-start number (up to 16,383)
    - 8-bit hot-restart number (up to 255)
- Available at the frontend by b2tt
- **Run number plays an important role for a fast recovery (hot-restart) from a local crash**
- A new cold-start number is given when configuration is updated and all subsystem are restarted

# Local Run Crash

- **Possible Timeline**
  1. Somebody crashes and back pressure to TTD stops the trigger
  2. Faulty system is identified (by a fast/slow system)
  3. Recovery is made automatically by a slow system (or by hand), **but only locally** — relevant systems may be re-started
  4. **Meanwhile the buffers in the healthy frontend become empty**
  5. The data of the crashed run are suspended somewhere in the buffer of event building at **COPPER, EB0, EB1, EB2 and ONSEN** (these are the systems which have more than one data sources)
- **Speed grades**
  1. **Fast** by TTD to the frontend, data-driven to the backend
  2. **Slow** NSM, EPICS, any other network/software control
  3. **Very slow** GUI and human interaction
  4. **Ultra slow** Login via SSH, running shell script by hand
  5. **XXXX slow** Hardware replacement, recomping firmware,. . .

# And restarting a new run

- **Slow control of O(1000) subsystems is not acceptable**
- A fatal error at HSLB firmware/COPPER driver $\Rightarrow$ EOF to basf2
  (e.g., unrecoverable FIFO full, surpassing the almost-full level)
- A fatal error at COPPER's basf2 $\Rightarrow$ connection closed to EB0
  - EB0 was block by read() for the faulty COPPER
    (select() can't be used due to time penalty)
  - Upon receiving connection closed, EB0 can start reading other COPPER data and throw away if the data of the crashed run number
  - EB1,2 can do the same, **but how about ONSEN?**
- At frontend,
  - Run number is updated and no trigger is guaranteed for next 10$\mu$s
  - Run (restart) number is incremented, **without resetting event num.**
    *or is it better to reset to zero? To avoid event# skips at HLT/EB2*
  - If reset to zero, run number can be treated as a higher digit of the event number (?) — **Does it work?**

# Data format (plan?)

*from b2link_ml:0058 on 2013.07.06. . .*

```
-------------------------------------------------
HSL: 0xFFAA(16) --- B2L header  |      reserved(16)
HSL: HSLB event count(24)              | TTRX-tag(8)
-------------------------------------------------
B2L: TT-tag(16)                  |      TT-utime(16)
B2L: TT-stime(28)                         | TT-type(4)
-------------------------------------------------
FEE: Data #0 (32)
FEE: Data #1 (32)
FEE:    ....
FEE: Data #n (32)
-------------------------------------------------
B2L: TT-tag-copy(16)             |  B2L checksum(16)
-------------------------------------------------
HSL: 0xFF55(16)                  | HSLB checksum(16)
-------------------------------------------------
```

*but this was not correct reality, and a few more things have been / have to be added.*

# Data format (CDC-test)

**The format actually used at the CDC beam test**

```
-------------------------------------------------
HSL: 0xFFAA(16) --- B2L header  |        0x0000(16)
-------------------------------------------------
B2L: '0'(1) | TT-ctime(27)              | TT-type(4)
B2L: TT-utime(16)                |        TT-tag(16)
B2L:  TT-exprun(32)
B2L: '0'(1) | B2L-ctime(27)             |  "0000"(4)
-------------------------------------------------
FEE: Data #0 (32)
FEE: Data #1 (32)
FEE:    ....
FEE: Data #n (32)
-------------------------------------------------
B2L: TT-utime(16)                |        TT-tag(16)
-------------------------------------------------
HSL: 0xFF55(16)                  | HSLB checksum(16)
-------------------------------------------------
```

- Second header word of HSLB does not exist (from beginning)
- TT-exprun and second B2L-ctime was added

# Data format (Final?)

**The format used at the telescope test**

```
-----------------------------------------------
HSL: 0xFFAA(16) --- B2L header  |      HSLB-tag(16)
-----------------------------------------------
B2L: '0'(1) | TT-ctime(27)              | TT-type(4)
B2L: TT-tag(32)
B2L: TT-utime(32)
B2L: TT-exprun(32)
B2L: '0' | B2L-ctime(27)                |  "0000"(4)
-----------------------------------------------
FEE: Data #0 (32)
FEE: Data #1 (32)
FEE:    ....
FEE: Data #n (32)
-----------------------------------------------
B2L: TT-tag(16)                  |  B2L-checksum(16)
-----------------------------------------------
HSL: 0xFF55(16)                  | HSLB checksum(16)
-----------------------------------------------
```

- tag (event number) and utime to be increased to 32-bit, HSLB-tag, B2L-checksum to be added

# Back pressure from ONSEN?

*from discussion on the boat. . .*

- Currently PXD back-pressure can come only through DHH
- ONSEN cannot directly talk to FTSW because of lacking (non-Ethernet) RJ-45 port and a 127 MHz reference clock source
- **How about compromising at the FTSW side?**
  - FTSW is able to receive “low speed” (O(a few 100Mbps)) optical serial link with SFP
  - FTSW’s serial link is on fabric, can be driven by a 156.25 MHz clock
  - RocketIO GTP can handle 0.1–0.5 Gbps with oversampling (from Virtex-5 datasheet)
  - Just need two SFP ports (one to deliver 156.25 MHz clock to FTSW) on ONSEN front panel, and bit of logic resources *and firmware work*

# Summary and plan

- At the moment we are running the system at “Ultra slow” speed grade
- At the telescope test, at least it should be in the “Slow” speed grade
- We definitely need “Fast” speed grade run start at Belle II, hopefully we can test it at the telescope test