

# ***Timing Distribution Update***

Mikihiko Nakao (KEK)

`mikihiko.nakao@kek.jp`

*2017.8.24*

*Trigger/DAQ workshop*

*NTU, Taipei*

# Cold startup of TTD

## ● Experiences on 2017.7.27 and 2017.8.10

- Both case there were some reconfiguration work
- Trouble-shooting document was made, but it is hardly perfect

## ● Hardware problems

- ttrx had to be replaced after swapping FTSW 189 / 190 (CDC COPPER)
- A short CAT7 cable stopped working after changing connections

## ● Firmware problems

- Jitter cleaner setup had to be redone on one FTSW at each startup
- Some FTSW firmware did not boot from flash memory

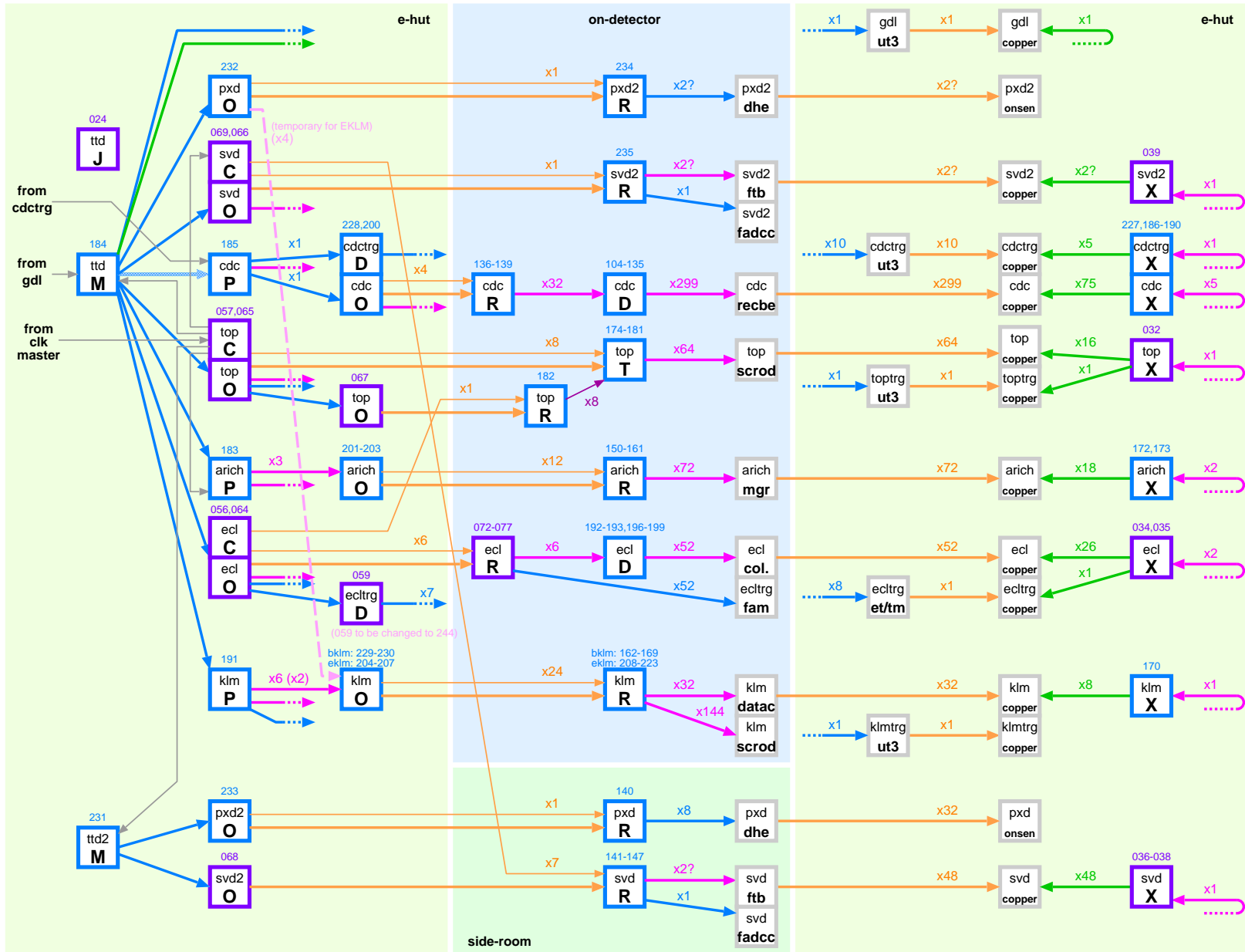
## ● Possible causes

- Noise in JTAG line did something nasty to existing configuration, so **better to turn on on-detector FTSW after E-hut FTSWs are up**
- Error in some configuration since it is not much tested

# TT-RX problem

- **FTSW → TT-RX link was not established**
  - Deserializer chip cannot lock the ttrxlink, so it does not look like a firmware problem
  - Happens on certain combinations of FTSW and TT-RX, but each of them still works in other combinations, so it is not a broken part
- **How to debug / proceed?**
  - Problem reproduced at B2 setup
  - Problem is not reproduced with standalone TT-RX
  - No good way to probe TT-RX by an oscilloscope
  - Testing with various bit patterns may find a way to improve ttrxlink
  - Already ~10 TT-RXs are marked bad, but still enough spares
  - **For now it is not an urgent problem**

# New TTD tree



2017.08.17 version

with second SVD2 and PXD2 for phase2

ttd2/pxd2/svd2 FTSWs in Ehut and on-detector are unused in phase3

pxd/svd FTSWs in side-room will be moved to E-hut in phase3

**Glossary**

**FTSW nodes**

- M: master
- P: primary
- O: optical
- R: receiver
- D: distributor
- T: top-special
- X: copper/trrx
- J: jtag

- b2tt on CAT7
- b2tt/B2L optical
- clock optical
- b2tt+JTAG
- JTAG
- ttrxlink
- clock

Timing Distribution Update — Mikhiko Nakao — p.4

List of changes ⇒ in the next slide

# New TTD tree

- **Rearranged on 2017.8.10**
  - Figure is posted at Confluence
- **TOP is using second FTSW for JTAG**
  - TOP JTAG workaround is still usable in Phase II...
- **EKLM is divided from BKLM, using PXD's FTSW**
  - EKLM has to get merged into KLM before Phase II
- **Master FTSW – GDL connection is not as in the figure now**
- **For Phase II**
  - FTSWs for ARICH, [PXD + SVD] (beast), [PXD2 + SVD2] (sideroom) are installed, and are ready to connect
  - Second master FTSW for sideroom is ready

# New b2tt

- Details described in 2017.4 DAQ meeting (see backup slides)
- **b2tt was updated to provide better status information**, to add the injection-veto information, and to fix some of the known problems
- FTSW tree is now using new b2tt
- CDC and ECL firmware are now using new b2tt
- ftprogs are updated
- **still far from perfect**, mostly because of software

# ttaddr -p example

```
% ttaddr -65 -l
% ttaddr -65 -p
:
3=17700 reg=1a8000fd 1a8000fd anyerrtagerr=7..20 ← unnecessary tagerr
  0=17701 2000640b no-info [s05a cpr3005a]
  1!17702 20006589 no-info [s05b cpr3005b] ← masked
  2=17703 20006588 no-info [s05c cpr3005c]
  3=17704 20006589 no-info [s05d cpr3005d]
  4=17705 20006588 no-info [s06a cpr3006a]
  5=17706 2000654c no-info [s06b cpr3006b]
  6=17707 200063c9 no-info [s06c cpr3006c]
  7=17708 20006500 no-info [s06d cpr3006d]
4=17800 reg=1b800001 1b800001 anyerr ttlost=0 ← ttlost from port-0 of FTSW178
  0=17801 20006588 no-info [s07a cpr3007a]
  1!17802 2000658a no-info [s07b cpr3007b] ← masked
  2!17803 20006587 no-info [s07c cpr3007c] ← masked
  3=17804 20006585 no-info [s07d cpr3007d]
  4=17805 20006588 no-info [s08a cpr3008a]
  5=17806 2000624a no-info [s08b cpr3008b]
  6=17807 200065cc no-info [s08c cpr3008c]
  7=17808 20006581 no-info [s08d cpr3008d]
:
```

- Now mask status can be read back
- ttlost is detected, now I know it is from FEE to FTSW
- Cannot run ttaddr -p in GLOBAL mode, causing unnecessary tagerr

# ttaddr, tagerr and LOCAL/GLOBAL

- GLOBAL mode just distributes b2tt from upstream, no way to insert anything into it
- **Therefore no ttaddr, cmdft, resetft, jtagft in GLOBAL mode**
- Need to switch to LOCAL mode to diagnose by ttaddr -p
- **But then tagerr occurs** when switching LOCAL ↔ GLOBAL, because local and global trigger numbers are different
- tagerr can be masked by software to avoid this to happen, but current ttaddr fails to properly do it (or there may be a problem in firmware)
- **ttaddr should be updated to be more usable during GLOBAL** (some attempt has been already made, but so far not successful)
- **Automatically running ttaddr -p upon an error is the idea**, once this tagerr problem is fixed



# ttlost

- Frequently happening in FTSW 059 for ECL-TMM

- FTSW firmware for 059 still uses the old b2tt, to be updated

- In FTSW for TOP, too

- In the TOP ttlost case, it was in FEE to FTSW direction (direction was unknown in old b2tt)
- Updating b2tt of TMM/TOP firmware may help, but probably not

- Need to dump incoming bit stream — 512-bits for 3 packets

- Firmware in preparation (compiled but yet to be tested)
- TOP FTSW firmware has to be programmed on the detector (no remote)

```
statft-20170724 FTSW #064 / ft2o055i 2017.08.10-17:14:29 -> 08.22 05:54:29
ERROR (2017.08.22 04:29:41 -- 2017.08.22 05:53:57 for 5056s)
16 exprun=004eb400 exp 1 run 3764 sub 0
17 onask=000010c0 s3q=0 clk=00 o=10c0 GLOBFL
1f jpl1=cc008000 clk=in GOOD-CLOCK
28292c trg=00000001 aux limit 0 <-> last 0
2a2b27 cnt 436014 > 0 > 366563 > 0 (85.7 > 0.0 > 72.5Hz)
2d stafifo=10000000 empty trg-enabled
20 reset=80000000 08.22-04:29:41.788(start) no-FIFO
31 err=00000800 08.22-05:53:57.638(error) src=b
25 ereg=1b000800 ttlost=b
30 busy=80000000
39/25 me=06400008 1b000800 0f000000 anyerr ttlost=b
405468 00=07200000 0a000000 0f000000 ready tag=0
415569 01=07300000 0a000000 0f000000 ready tag=0
42566a 02=07400000 0a000000 0f000000 ready tag=0
43576b 03=07500000 0a000000 0f000000 ready tag=0
44586c 04=07600000 0a000000 0f000000 ready tag=0
45596d 05=07700000 0a000000 0f000000 ready tag=0
485c70 X8=03480000 0a0597e3 0f000000 ready tag=366563
495d71 X9=03580000 0a0597e3 0f000000 ready tag=366563
4a5e72 010=0640b000 0a0003e3 00006545 ready tag=995 no-info
4b5f73 011=05900008 1b000000 00006502 anyerr ttlost=me no-info
9f latency=0c00b000 maxtrig=12 maxtime=351.44us
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00%)
```

# Master FTSW

- **Master FTSW still running P (primary) firmware**
  - Therefore missing functions to handle trigger type, revolution signal and injection veto, all received from GDL
  - SVD's APV deadtime emulator to be included, too
- **First verison of ft3m firmware was made and compiled**
  - GDL counterpart has to be prepared, provided, **and tested**
  - FTSW register map is also updated for ft3o/ft3p firmware
  - Once confirmed to work, to be updated at next chance
- **Connection to GDL via 2 CAT-7 cables**
  - Trigger timing just as now, timing w.r.t. clock has to be guaranteed
  - 4-bit trigger type, parallel signal, need a mechanism to synchronize
  - b2tt-like encoded signal for both directions for revo/injection veto

# Revo signal

## ● **b2tt frame and SuperKEKB beam revo(lution)**

- frame is first fixed by FTSW, and then revo marker is defined w.r.t. the frame
- Previously they are somewhat mixed up
- “revoclk” signal (0..1279 counter for frame) was dropped in b2tt-0.52 while sorting frame and revo signals, but it will be restored with a new name, “frameloc”
- “revo9” is renamed to “frame9”

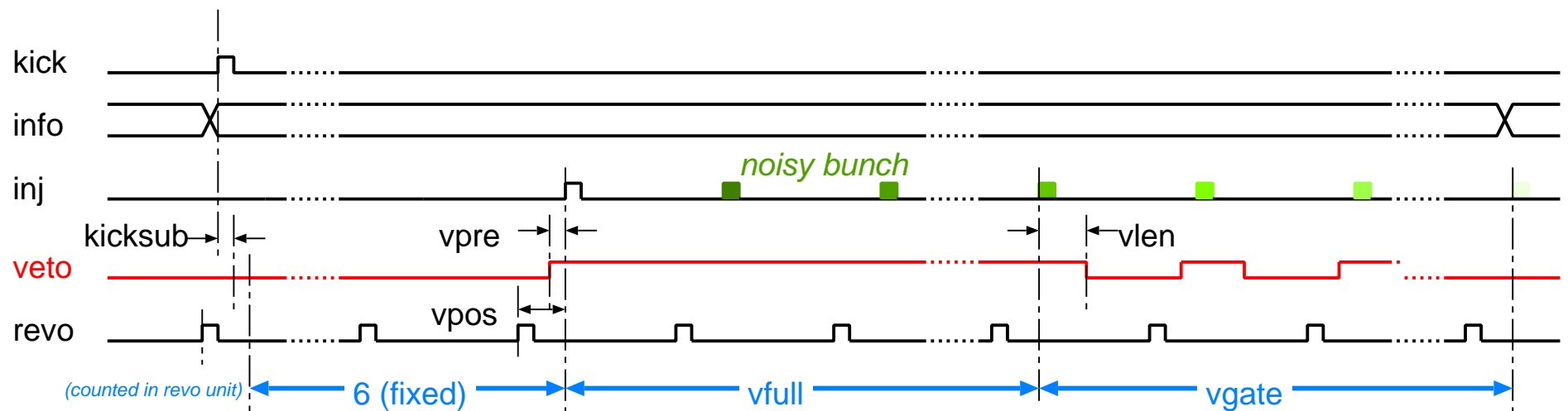
## ● **Distributed by b2tt (3 parameters, 33 bits)**

- “frame” is the reference timing of b2tt
- revolution marker location w.r.t. the frame is distributed
- abort gap information is also distributed

# Injection veto

- **Veto info distributed by b2tt (6 parameters, 63 bits)**
  - “kick” signal is generated by b2tt
  - all info available at kick timing
- **Structure defined by GDL and received by master FTSW**
  - Receiving info from the SuperKEKB “event system” is not yet working
  - Even if we use the “old” way (NIM signal), it has to be generated by GDL to veto the trigger

(figure updated since B2GM)



# utime adjustment

- **utime adjusted to NTP time by bootft or utimeft at beginning**
  - It starts drifting away if nothing is done (a few seconds a day)
  - Once started, utime / ctime should not be touched
- **Adjustment by changing the number of clocks / second (= assumed clock frequency)**
  - Clock frequency has to be multiple of 1280 (frame) or  $10\mu s$ , so adjustment cannot be better than  $5\mu s$
  - Adjusting every 10 minutes (600s) should keep a few ms difference
  - Previously Hanjin Kim made such a program, but not working well
  - A new program is made, but currently kept within 100ms
- **Frequency changes even during a run**
  - Time difference will be too large if waited for 8-hour long run

# Mask problem

- **Masking was probably not working properly all the time**
  - sometimes busy or error from masked port prohibits a run start
  - not yet clarified if it is a firmware problem or software problem
  - not much time was spent on this problem, but need to fix

# ttd1 problem

- ttd1 is not a powerful CPU, but many people relies on it
- (probably) memory or other resource is used up time to time
  - it kills some random processes, such as statft or pocket\_ttd
- **Action plans**
  - ttd daemon to handle all ftprogs / ttaddr requests from outside through NSM2
  - This should make it easier to coordinate a script between readout PC and TTD
  - And/or, update the ttd1 CPU to a more powerful one, which was purchased but not set up yet

# Summary / more items

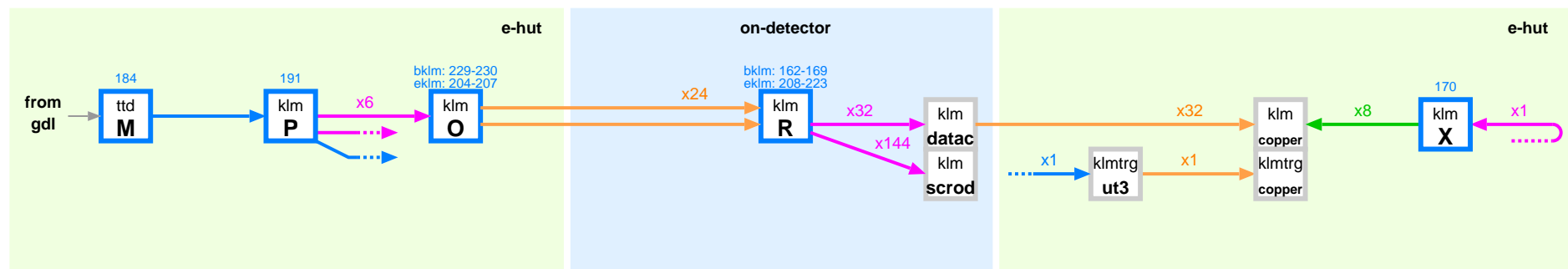
- Lots of things have been updated regarding b2tt, but still more adjustment is needed
- Hopefully a better handle to resolve existing problems with new b2tt firmware
- Tools for deadtime minimization are ready, tuning to be started
- Tools for non-stop DAQ is ready for ttlost/b2llost, software work to be started
- TOP ZYNQ programming has to be still made



# Backup

# b2tt and FTSW

- All FTSW/FEE has a unique 20-bit address
- b2tt distributes a frame of data every 100  $\mu\text{s}$ 
  - All FTSW/FEE receives same data
  - “address” inside data to send specific command to one FTSW/FEE
  - 😊 (already well-defined, no change this time)
- b2tt collects a packet of 112-bit every 6  $\mu\text{s}$ 
  - One FTSW receives from up to 12 FTSWs/FEEs
  - They have to be merged into another 128-bit packet to upstream
  - 🚧 **badly designed, to be revised this time**



# Link status

- Each FTSW/FEE can tell if the link is “**up**” or “**down**” now
- It is also needed to know if it has been “**lost**”
- For a FTSW (**me**) to other FTSW/FEE (**you**), need status (**up/down/lost**) of the link to **you** and your link to **me**
- It is also needed to know **down/lost** anywhere in the tree
- **Link status “names” and handling are rearranged**
  - **alive** — link to you is up
  - **bound** — link to me is up at you
  - **adown/bdown** — link to me/you was up at reset, but not now
  - **alost/blost** — link to me/you was lost since last reset
  - **ttdown/ttlost** — link is down/lost anywhere in the tree

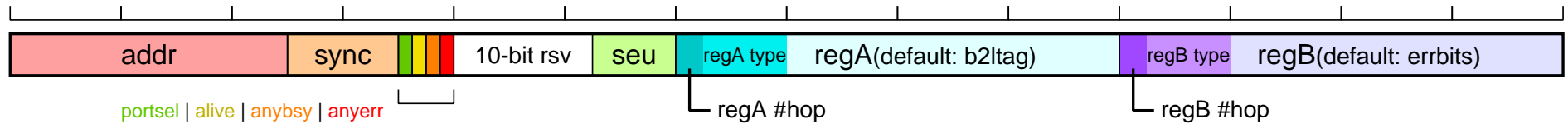
# How to merge?

- Merging the error status
  - It was logical-or, but then the number of error bits is limited, and no additional info accompanied
  - Only one “**persistent**” error bit, other bits are just to see the status
  - “**Prioritized**” the error status, with 20-bit error location info
- Merging processed number of events — take the smallest
- Merging the SEU counts — sum is calculated at each step

```
payload(111 downto 92) <= myaddr;  
payload(91)           <= b2ttup;  
payload(90)           <= b2linkup;  
payload(89 downto 82) <= cnt_payload;  
payload(81)           <= busy;  
payload(80)           <= sta_err;  
payload(79)           <= sta_ttlost;  
payload(78)           <= sta_b2llost;  
payload(77)           <= sta_tagerr;  
payload(76)           <= sta_fifoerr;  
payload(75)           <= sta_fifoful;  
:
```

- Old packet was like this, all error bits were logical-or
- It was not possible to access the individual FTSW/FEE status without generating “ttlost”

# New b2tt ACK-payload format



- **20-bit address** to identify the source of the payload
  - Address is assigned by `ttaddr`
  - Usually the directly connected FTSW/FEE is visible
  - It could be the hops behind by selecting port (`selport=1`)
- **3 Error bits** are persistent regardless the `selport`
- **SEU count** to collect the sum of the number of SEUs
- **Two registers** as window to access more bits
  - 2-bit #hop, 6-bit type, 24-bit data
  - regA is state dependent, to tell error source upon error
  - If no error, tag count = how many events are written to belle2link
  - regB is user controllable to access more info

# New statft format

- 3 registers (stat regA regB) for each link and **myself** at statft
- **stat** contains address, seu-count, error/busy, link status
- **regA** tells the status of the tree behind the link
- **regB** is the window to see the status of any FTSW/FEE

```
statft-20170419 FTSW #103 / ft3o052e 2017.04.28-00:10:35 -> 04.28 07:27:10
16  exprun=00000000 exp 0 run 0 sub 0
17  omask=00008000 s3q=0 clk=00 o=0000 LOCAL
1f   jpll=cc008000 clk=in GOOD-CLOCK
28292c trg=00000000 none limit 0 <-> last 0
2a2b27 cnt 0 > 0 > 0 > 0 (0.0 > 0.0 > 0.0Hz)
2d stafifo=10000000 empty trg-DISABLED
20  reset=80000000 04.28-00:14:01.585(runrst) no-FIFO
3d   err=00000000 04.28-00:14:01.585(error) src=none
39/25 me=10300400 17000010 ERROR-or ferr
405468 00=10200000 0a000000 00000000 ftag=0 null
44586c 04=22500000 10000001 00000000 ebit=o=0 null
```

# Miscellaneous new features

- **FTSW has its address at startup, no need to run “ttaddr”**
  - board number, e.g., #103, is converted into address 0x10300)
  - ttaddr is a recursive program to assign all addresses in the tree
- **FEE address still has to be assigned from a linear list**
  - e.g., FEE at FTSW #103 port 2 has address 0x10302
  - Much easier to implement in the run control software
- **More info about “time” in statft**
  - When FTSW was programmed (btime)
  - When last reset occurred (rstutime)
  - When last error or run start occurred (etime)
  - When it is now (utime)

# Firmware code merging

- **Need to change: b2tt, ft2u, ft3o, ft3d, ft3x**
  - b2tt: 10 VHDL files, 7 of them are also used in ftxx firmware
  - ft2u: 39 VHDL files, 7 files are ft2u specific
  - ft3o: 37 VHDL files, 4 files are ft3o specific
  - ft3d: 24 VHDL files, 2 files are ft3d specific
  - ft3x: 23 VHDL files, 1 file is ft3x specific
- **24 files were updated for the case of ft2u**



# Firmware / software status / plan

- **Need to change: b2tt, ft2u, ft3o, ft3d, ft3x, statft**  
(and more variants ft2o ft3p ft3r ft2r ft2x from the same source)
- **b2tt / ft2u were there since last TRGDAQ WS / B2GM**
- **Last three weeks were devoted to implement the rest**
  - All firmware are finally built and being tested in B2
  - Virtex-5 resource is very tight (chipscope is given up for ft3d now)
  - PocketDAQ version works, full chain version is under debug
- **Plan**
  - Finish debugging during golden week
  - Try with CDC firmware first
  - Release new ft2u 0.85 and b2tt 0.50
  - FEEs with old b2tt can be still used but with limited error info