b2tt working document

Mikihiko Nakao (KEK, IPNS)

2017.8.2 (version 0.06)

1 About this document

This document describes the mechanism of b2tt in as much detail as possible. The main purposes are to understand:

- how to use b2tt in the frontend electronics firmware
- how to use FTSW and related software
- internal structure of FTSW firmware

It includes too detail information just to use b2tt and FTSW, which can be safely skipped.

2 Mechanism of b2tt

2.1 b2tt is/does

- a Trigger and Timing distribution and control protocol for Belle II
- bidirectional communication based on a 254 Mbps serial data link
- synchronized to a 127 MHz external clock derived from SuperKEKB RF
- distribute common data, while recipient may react differently if needed
- collect independent data and merge at every stage of collection

2.2 signal lines

- FTSWs are interconnected via one CAT-7 cable (4-pair signal lines) or via 2 pairs of optical fibers
  - 4 signal lines are for: clock signal, distribution signal, collection signal, modulated TDO signal
  - All signal lines are AC-coupled
- FTSW and frontend electronics (FEE) are connected via one or two CAT-7 cable(s).
  - Second CAT-7 cables is for JTAG programming, unused if unnecessary

2.3 Clock signal

- Always sent via CAT-7 7-8 pair or optical fiber 1
- The most upstream FTSW receives the clock at IN port from the clock master board
  - The clock master board receives 509 MHz RF signal from SuperKEKB and generates the 127 MHz clock (external mode)
  - When RF signal is absent, clock master generates the clock from its 127 MHz crystal oscillator (internal mode)
  - When the most upstream FTSW has no clock input, the clock is generated from the crystal oscillator of the FTSW (pocketdaq mode)
- All the other FTSWs receives the clock at the IN port (except for type-R) or upper SFP port (type-R)
  - At every stage, FTSW reduces the jitter of the clock by a jitter cleaner
  - Typical jitter of the clock signal after 10m CAT-7 transport is 20ps RMS (measured value by an oscilloscope)
  - If the clock is momentarily lost, it is detected as loss of the PLL lock of the jitter cleaner and marked as a clock error
  - Touching the input clock cable may cause a clock error
- Since jitter cleaner output has 4-fold phase ambiguity, FTSW detects the clock phase and resets the jitter cleaner until the phase becomes the expected one
  - Once the phase is fixed, it should not change

2.4 Distribution signal

- Always sent via CAT-7 3-6 pair or optical fibre 2
- 254 Mbps data signal synchronized to the clock signal
  - 8b10b encoded
  - 2560-bit frame data at the same cycle as the SuperKEKB revolution
  - One frame consists of 16 packets of 160 bit
  - One packet consists of 16 code of 8b10b
- Data packet structure
  - A packet starts with a start code K28.1 symbol (which uniquely defines the code boundary in 10b coding)
  - Trigger code is embedded at any 8b10b location of the packet
  - Since 24-clock minimum trigger interval is required, one packet can contain up to 4 trigger codes
  - When the trigger code is embedded at the start of the packet, the second location is filled with an alternative start code K28.3 symbol
  - 11 data codes are filled in the locations unused by the start code and trigger codes

  - When there are 0 to 2 trigger codes, 2 to 0 idle code (K28.3 symbol) is filled
  - When there are 0 to 3 trigger codes, the last location is filled with an 8-bit CRC (when there are 4 trigger codes, CRC code is omitted)
- Link-up condition
  - Packet has to start with a K28.1 symbol, or second code is an alternative start code K28.5
  - CRC matches
  - Link goes up after repeating the above condition for 240 packets
  - Link goes down immediately if the above condition fails
  - 8b10b disparity check and undefined code check are not done, but these lead to an illegal bit pattern of K28.1 or K28.5 and may cause link down

2.5 Trigger signal, trigger code

- There are fine-timing trigger with 1/4 clock unit (1.9ns) time information and coarse-timing trigger with 1 clock unit (7.8ns) time information.
- Trigger code is is embedded in the data packet of the distribution signal at any location of the 8b10b code boundary
  - Trigger code is an 8b10b code with the most significant bit of 8b is 1
  - Out of remaining 7-bit of the 8b code, 3-bit represents the timing within the 10b code (5 clock) and 4-bit represents the trigger type
  - Time (t_{file}=0,1,2,3) of a fine-timing trigger is represented by 4 specific trigger types
- Trigger types are as follows (defined in `b2tt_symbols.vhd`)

PID0	0x0	fine-timing trigger, t_{fine}=0
PID1	0x4	fine-timing trigger, t_{fine}=1
PID2	0x8	fine-timing trigger, t_{fine}=2
PID3	0xc	fine-timing trigger, t_{fine}=3
RSV0	0x2	reserve 0
RSV1	0x6	reserve 1
RSV2	0xa	reserve 2
RSV3	0xe	reserve 3
ECL	0x1	ECL trigger, coarse-timing
CDC	0x3	CDC trigger, coarse-timing
DPHY	0x5	delayed physics trigger
RAND	0x7	random trigger
TEST	0x9	test trigger
RSV4	0xb	reserve 4
RSV5	0xd	reserve 5
NONE	0xf	not a trigger

2.6 JTAG signal, JTAG code

- JTAG signal consists of TCK TMS TDI distribution signals and TDO collection signal, and is asynchronous to the packet structure
- Distribution signal of TMS TDI is implemented as 2-bit status at the TCK timing represented by 4 different 8b10b K symbol
  - For (TMS TDI) being (0 0) (0 1) (1 0) (1 1), K-symbol K23.7 K27.7 K29.7 K30.7 are used, respectively
  - Up to 4 K-symbols can be embedded in a data packet, making the maximum JTAG speed to be 3.175 MHz, about the half of the default of the Xilinx Platform USB cable with the impact program
  - Trigger code and JTAG signals cannot co-exist, and therefore JTAG cannot be used while trigger is enabled
- To reduce the latency to receive TDO, the fourth line of CAT7 or optical fiber is used
  - Since TDO itself is a DC level signal, it is modulated by the 127 MHz clock to transfer over the AC-coupled CAT7/fiber connection

2.7 Distribution payload

- 77-bit payload is constructed from 11 lower 7-bit of 8b data codes whose most significant bit is 0
- See section `Distribution payload to FEE` for detail

2.8 Masking distribution signal

- Distribution signal can be masked without losing the link by replacing packet 4 through 7 with an idle pattern (`p_otrg.vhd`)
- Using this function, it is possible to send specific request (command in packet 4) to specific ports
- The disparity at the end of the idle pattern is adjusted by adding or not adding the CRC code to packet 2 and 6 for consistent disparity
- During a run or JTAG sequence, trigger code or JTAG code makes the disparity unpredictable, and the idle pattern causes inconsistency in the disparity
- Therefore distribution signal cannot be masked during the run and JTAG sequence.
- Since trigger and JTAG signals cannot be masked in the transmitter side, they are masked by setting the receiver side to ignore these signals.

2.9 Collection signal

- Always sent via CAT-7 3-6 pair or optical fibre 2
- 254 Mbps data signal synchronized to the clock signal, similar to the distribution signal
  - 8b10b encoded

  - 160-bit packet data at the 1/16 cycle of the SuperKEKB revolution
  - One packet consists of 16 code of 8b10b
- Since phase between the collection signal and the clock depends on the length of the connected cable and other things, the phase is scanned when the link is established in such a way that the phase difference is at the center of the range with stable link.
- Data packet structure
  - Packet begins with a start code K28.1 symbol (also known as a comma code to uniquely determine the code boundary in 10b bit sequence)
  - 14 data codes follows the start code, and therefore one packet consists of 112-bit decoded data
  - At the end an 8-bit CRC is embedded
  - At any location of 2-bit (clock) boundary, a busy code K28.5 symbol can be embedded
  - When a a busy code is embedded, CRC is not embedded and no CRC test is made
  - The packet is unused if more than one busy codes are embedded
- Link-up condition
  - 8b10b code boundary is determined when code boundary is unknown (`cntbit = 5`) and K28.1 start code is found, and reset to unknown when the start code is found at a non-boundary location
  - Packet is considered to be valid if 14 8b10b data codes follow and CRC matches when CRC exists, or less than 14 data codes are found because of more than one busy codes (`cntoctet < 16`)
  - Link is established if 256 consecutive packets are valid (`cntvalid = 255`)
  - Link goes down if any of the conditions is not fulfilled

2.10 Busy signal and busy state

- Busy signal is collected at each step of FTSW, and their logical-or is sent to upstream
- In order to minimize the latency, busy codes can be embedded at any clock boundary, regardless the 8b10b code boundary
- Busy on and off are defined by K28.5 disparity plus and minus codes
- K28.5 10b code pattern should not appear anywhere in the bit stream (if disparity handling is not correctly implemented, K28.5 code inserted at an arbitrary location can make a bit pattern equivalent to K28.1, and may cause ttlost or busy which cannot be reset)

2.11 Clock and link status

- `ckup=1` is the status when clock is correctly received and PLL is properly locked

- `ttup=1` is the status when b2tt protocol from upstream is correctly received (downstream-side signal)
- `alive=1` is the status when b2tt protocol from downstream is correctly received (upstream-side signal)
- `bound=1` is the status when `ttup=1` signal is properly propagated to upstream (upstream-side signal)
- `adown=1` / `bdown=1` indicate that it was `alive=1` / `bound=1` at the last reset, and next reset will make `alive=0` / `bound=0`
- `alost=1` and `adown=0` / `blost=1` and `bdown=0` indicate that it recovered from `adown=1` / `bdown=1` and the next reset will make `alost=0` / `blost=0`
- summary is as follows

`ckup`	input clock is locked by PLL/DCM of `b2tt_clk`
`ttup`	Link from upstream is established
`ttlost`	Link from upstream was lost
`ttdown`	Link from upstream is down
`alive`	D -> U link is established regardless of `dmask`
`alost`	D -> U link was lost and not disabled by `dmask`
`adown`	D -> U link is down and not disabled by `dmask`
`bound`	Both D <-> U links are established regardless of `dmask`
`blost`	Either of D <-> U link was lost and not disabled by `dmask`
`bdown`	Either of D <-> U link is down and not disabled by `dmask`

3 b2tt implementation into FEE

3.1 Firmware structure

- b2tt firmware has following tree structure
- some of the files contain more than one components and therefore components and files do not match one by one

```
b2tt                    (top module)
 +-- b2tt_clk             (clock management)
 +-- b2tt_fifo            (FIFO management)
 +-- b2tt_revo            (revolution signal handling)
 +-- b2tt_injv            (injection veto handling)
 +-- b2tt_decode          (decoder top)
      +-- b2tt_iddr            (DDR receiver: serial => 2-bit)
           +-- b2tt_iscan           (signal phase scanner)
      +-- b2tt_decomma         (comma/boundary finder)
      +-- b2tt_debit2          (2-bit decoder => 10-bit)
      +-- b2tt_debit10         (10-bit decoder => octet)
           +-- b2tt_de8b10b         (8b10b decoder)
      +-- b2tt_detrig          (trigger code finder)
      +-- b2tt_deoctet         (octet decoder => packet)
      +-- b2tt_depacket        (packet decoder => command)
      +-- b2tt_detag           (trigger tag checker)
 +-- b2tt_encode          (encoder top)
```

```
      +-- b2tt_encounter        (counters for encoding)
      +-- b2tt_enpacket         (packet encoder <= various info)
      +-- b2tt_enoctet          (octet encoder <= packet payload)
      +-- b2tt_enbit2           (2-bit encoder <= octet)
           +-- b2tt_en8b10b           (8b10b encoder)
      +-- b2tt_oddr             (DDR driver: serial <= 2-bit)
+-- b2tt_payload           (b2tt payload construction)
```

3.2 VHDL files

- All following files in b2tt directory of the b2tt package for the target FPGA are to be used

b2tt.vhd	main	common
b2tt_8b10b.vhd	8b10b conversion	common
b2tt_clk_s6.vhd	clock handling	Spartan-6
b2tt_clk_v5.vhd	clock handling	Virtex-5
b2tt_clk_v6.vhd	clock handling	Virtex-6
b2tt_clk_x7.vhd	clock handling	7-series
b2tt_ddr_s6.vhd	input and output	Spartan-6
b2tt_ddr_v5.vhd	input and output	Virtex-5
b2tt_ddr_v6.vhd	input and output	Virtex-6
b2tt_ddr_x7.vhd	input and output	7-series
b2tt_decode.vhd	b2tt decoding	common
b2tt_encode.vhd	b2tt encoding	common
b2tt_fifo.vhd	b2tt FIFO	other than Spartan-6
b2tt_fifo_s6.vhd	b2tt FIFO	Spartan-6
b2tt_iscan.vhd	phase scan	common
b2tt_revo.vhd	revolution signal	common
b2tt_injv.vhd	injection veto	common
b2tt_payload.vhd	b2tt payload	common
b2tt_symbols.vhd	types and constants	common

3.3 coregen files

- coregen files are needed only for chipscope
- Separately prepared for Spartan-6 (`b2tts6-coregen`), Virtex-5 (`b2ttv5-coregen`), Virtex-6 (`b2ttv6-coregen`) and 7-series- (`b2ttx7-coregen`) in the package ジ内に用意されている
- coregen files are not necessarily usable depending on the ISE version or FPGA types, and it may be necessary to regenerate
- These coregen files are not needed if signals for chipscope are included in other chipscope setup

3.4 Examples

- b2tt package includes simple firmware source files for supported FPGAs which uses only b2tt

- More realistic examples can be found in the Belle2link package, which is out of scope of this document
- ISE 12.4 is assumed for the compiling environment, and should be compatible with later ISE versions

3.5 b2tt interface

- b2tt can be used by including (only) one b2tt instance in the design
- b2tt has many generics and ports, but not all are used: some are mandatory, some are only for particular users and some are not meant to be used by users
- Below, signals with (∗) are mandatory, those with (∗∗) are mandatory when Belle2link is used, and those with (∗∗∗) are mandatory for firmware to be used inside a radiation area
- generics are defined as follows (do not set those not listed below)

`SUBSYSTEM`	(∗) set one of "TTFEE_" in `b2tt_symbols.vhd`
`FWTYPE`	(∗) 4-bit integer for user defined firmware type
`VERSION`	(∗) 16-bit integer for user defined firmware version
`FLIPTRG`	set to '1' when polarity is reversed for RJ-45 TRG signal
`FLIPACK`	set to '1' when polarity is reversed for RJ-45 ACK signal
`CLKDIV1`	denominator of reduced clock of `divclk1`
`CLKDIV2`	denominator of reduced clock of `divclk2`
`USEICTRL`	set to '0' when `IDELAYCTRL` is already used in firmware
`USEEXTCLK`	set to '1' to use externally generated clock
`USE254IN`	set to '1' to generate `dblclk` 254 MHz clock
`SIM_SPEEDUP`	set to '1' for fast simulation of link up procedure

- signals are define as follows (those not listed below can be kept open)

b2ttver	b2tt version
clkp/clkn	(*) RJ-45 7-8 pair for clock
trgp/trgn	(*) RJ-45 3-6 pair for b2tt input
ackp/ackn	(*) RJ-45 1-2 pair for b2tt output
extclk etc	external clock
id	(*) board ID
usrreg	8 bit user defined register, writable through b2tt
usrdat	16 bit user defined register, readable through b2tt
b2clkup	(*) b2tt recognizes the clock
b2ttup	(*) b2tt link is up
sysclk	(*) 127 MHz main clock
dblclk	254 MHz clock, only for Virtex-6
utime	Unix-like current time in the unit of second
ctime	current time within the second in clock unit
divclk1,2	reduced clock 1,2
exprun	exp(bit31:22) run(21:8) subrun(7:0)number
running	'1' during a run
runreset	(*) reset at the run start
feereset	reset signal just for FEE
b2lreset	(**) Belle2link reset
gtpreset	(**) GTP/GTX reset
rstmask	'1' if runreset is masked
trgout	(*) trigger signal (1 clock length)
trgtyp	trigger type
trgtag	trigger number (starting from 0)
trgmask	'1' if trgout is masked
frame	frame signal every 1280 clocks, synchronized in all FEEs
frame9	'1' for one-frame long in 1 out of 9 frames in all FEEs
revoloc	revosig location in the frame
revosig	SuperKEKB revolution signal
abortgap	'1' if in SuperKEKB abort gap
injveto etc	injection veto signal
busysrc	8 bit (up to 8 types of) busy signal
errsrc	8 bit (up to 8 types of) error signal
b2plllk	(**) Belle2link PLL lock status
b2linkup	(**) Belle2link link status
b2linkwe	(**) '1' if FEE is writing into Belle2link
b2lclk	(**) clock to drive Belle2link writing
semscan etc	(***) to communicate with the SEU mitigation code
fifordy	(*) ready to read FTSW FIFO data
fifodata	(*) 96 bit information to be readout from FIFO
fifonext	(*) signal to move to the next FIFO entry
regdbg	x"00" to be set
dbglink	96-bit signal for chipscope

- Firmware type, version and board ID can be read out through b2tt, and are useful for troubleshooting to identify miscabling or incorrect firmware
- To use external clock, set USEEXTCLK to ='1'=, do not use clkp/clkn, and provide extclk extclkdbl extclkinv extdblinv corresponding to sysclk and dblclk and their inversion, and extclklck corresponding to b2clkup.

- `injveto` is an injection veto signal, and physics triggers are not generated while this signal is set
- `injveto` time structure is represented by parameters `injvpos` `injvpre` `injvplen` `injvpfull` `injvgate` at the `injkick` timing

4 Distribution payload to FEE

4.1 Payload structure

- Consists of 77-bit
- Bit 76 is a broadcast flag (1 for broadcast)
- Bit 75–64 for packet type (12-bit)
- Bit 63–0 for packet-type-dependent data

4.2 Packet types

- 8 types: TTAG FREQ RST CMD DISP INJV REVO SYNC
- Bit location is indicated by parenthesis below
- SYNC packet (broadcast only) —
  runreset(63) frame3(62) frame9(61) jtagen(60) trgen(59) utime(58–27) ctime(26–0)
- CMD packet (either broadcast or not) —
  target address(63–44) command(43–36) data(35–0)
- RST packet (either broadcast or not) —
  target address(63–44) feereset(43) b2lreset(42) gtpreset(41) incdelay(40) caldelay(39) setaddr(38) tagerr(37) errreset(36) clraddr(35) running(34) unused(33-0)
- FREQ packet (broadcast only) —
  unused(63–34) B2TTVER(33–24) clkfreq(23–0)
  (clock frequency is not exactly 127216000 Hz, but a multiple of 1280 and the best fit number to match the SuperKEKB RF frequency and time obtained by NTP, and can be dynamically changed when run is not taken)
- TTAG packet (broadcast only) —
  exprun(63–32) ttag(31–0)
  (ttag is the number of triggers sent, and it is compared with the number of received triggers in each frame)

5 Collection payload from FEE

5.1 Payload format

Defined in `b2tt/b2tt_payload.vhd`

111–92	myaddr	depends on dsel
91–84	cnt_payload	
83	dsel	always dsel=0 for FEE
82	ttdown	no dependence on dsel
81	anybsy	no dependence on dsel
80	anyerr	no dependence on dsel
79–70	reserve10	(10-bit)
69–64	cnt_seudet	no dependence on dsel
63–61	reserve3	reg1 has no dependence on dsel
60–56	reg1.type	(5-bit)
55–32	reg1.data	(24-bit) data
31–29	reg2.depth	(3-bit) incremented when dsel=1
28–24	reg2.type	(5-bit)
23–0	reg2.data	(24-bit) data

6 Error diagnosis

- When error happens, the error bit is raised (anyerr=1)
- Error type and details are shown by the priority-selective register
- Further detail can be investigated through user-selective register

6.1 TTREG type

- Priority is in the order of: CLOST TDOWN TLOST TERR EBIT FERR RERR LDOWN LLOST
- FERR is returned when current value of feeerr(7:0) at FEE is not 0
- RERR is returned when feeerr became nonzero but restored
- FERR and RERR are not generated by FTSW

6.2 Clock error (CLOST)

- Clock error has the highest priority since it affects everything else
- Clock error is identified by loss of the lock of PLL or DCM in various places
- Clock error is handled separately for the current status and past error (if clock has problem now, it is unlikely to be able to receive the status through b2tt)
- No clock error reported from COPPER/TTRX (ttrxlink)
- Following bit pattern is assigned

bit 23	jpdown	FTSW jitter cleaner PLL is down
bit 22	jplost	FTSW jitter cleaner PLL was lost
bit 21	ckdown	b2tt DCM lock is down (can it be visible?)
bit 20	cklost	b2tt DCM lock was lost
bit 19	plldown	Belle2link GTX/GTP PLL lock is down (FEE only)
bit 18	pllost	Belle2link GTX/GTP PLL lock was down (FEE only)
bit 17	icterr	ictrldelay error
bit 16	ictdown	DCM lock for ictrldelay is down
bit 15	clmany	Count of cklost exceeds 3
bit 14:13	cntlost	2-bit count of cklost (this FTSW only)
bit 12:0	clsrc	source of errors (0 if not from downstream)

6.3 Link error (TDOWN, TLOST, LDOWN, LLOST)

- b2tt error may affect other things and hence has a higher priority
- While prioritizing, since port bit pattern is shared, different TTREG types are used
- Maximum 20 ports of error sources to handle errors from ttrxlink
- Maximum 13 ports from b2tt, and remaining 7 bits are unused and set to 0
- Bit 23:20 has different definitions for ttrxlink and b2tt

bit 23	alost	tlost=alost if 1
bit 22	many	Count of cklost exceeds 3 (this FTSW only)
bit 21:20	cntlost	2-bit count of cklost
(bit 23:20)	(rxlost)	(HSLB port of LDOWN/LLOST, only for COPPER)
bit 19:0	src	source of errors (0 if not from downstream)

6.4 errors from FEE (FERR, RERR)

- FEE can pass 8 different error types in an 8-bit signal
- Since they are merged as logical-or during collection, it is not recommended to express more than 8 types by different bit patterns
- FTSW does not care about FEE error types
- FERR is for the current status, and RERR keeps the error status since the previous reset.

bit 23:16	feeerr	8-bit error from FEE
bit 12:0	src	source of errors (0 if not from downstream)

6.5 other errors (TERR)

- TERR covers all errors other than those from clock, link and FEE
- All errors are covered by one of TTREG type from CLOST to TERR

bit 23	tagerr	tag mismatch (not checked in FTSW)
bit 22	timerr	time mismatch
bit 21	badver	b2tt version mismatch
bit 20	fifoerr	b2tt fifo error
bit 19	semmbe	multi-bit error in SEU mitigation
bit 18	semcrc	CRC error in SEU mitigation
bit 17:13	rsv	reserve
bit 12:0	src	source of errors (0 if not from downstream)

6.6 error summary (EBIT)

- A register to see the summary of all error types
- This is one of possible user selectable register, and is not used as an error diagnosis register
- It does not contain down stream status

bit 23	tagerr	tag mismatch
bit 22	timerr	time mismatch
bit 21	badver	b2tt version mismatch
bit 20	fifoerr	b2tt fifo error
bit 19	semmbe	multi-bit error in SEU mitigation
bit 18	semcrc	CRC error in SEU mitigation
bit 17:13	rsv	
bit 12	clost	
bit 11	tdown	
bit 10	tlost	
bit 9	ldown	
bit 8	llost	
bit 7:0	rerr	8-bit error from FEE

6.7 error mapping from old b2tt protocol

- Error bits from a FEE using old b2tt protocol are mapped as follows
- Major error bits are covered, although not all errors are mapped

old payload	signal	new error
payload(91)=0	not sta_linkup	bdown
payload(90)=0	not sta_b2lup	drega=LDOWN
payload(17)=0	not sta_plllk	drega=CLOST bit-18
payload(80)=0	sta_err	drega=FERR
payload(79)=1	sta_ttlost	drega=TLOST
payload(78)=1	sta_b2llost	drega=LLOST
payload(77)=1	sta_tagerr	drega=TERR bit-23
payload(76)=1	sta_fifoerr	drega=TERR bit-20
payload(75)=1	sta_fifoful	drega=TERR bit-20
payload(69:68)/=0	sta_seuerr(1:0)	drega=TERR bit-19:18
payload(13)=1	sta_badver	drega=TERR bit-21
payload(12)=1	sta_timerr	drega=TERR bit-22

7 Link status register (dstat/xstat/pstat)

- `dstat` for each b2link represents the general status of the link destination
- `addr` represents the source of the currently selected `dregb` register (default is that of the directly connected FEE or FTSW)
- generated in `o_collect`

bit 31:12	addr	address of currently selected dregb
bit 11	dsel	if 0, addr is for the directly connected one
bit 10	anyerr	error anywhere in this downstream
bit 9:4	seucnt	total number of SEU in this downstream
bit 3	bsyin	busy anywhere in this downstream
bit 2	bdown	b2tt from this FTSW is down
bit 1	blost	b2tt from this FTSW was lost
bit 0	alost	b2tt from direct downstream was lost

- `xstat` for each ttrxlink if the directly connection is to a TT-RX
- generated in `x_collect`
- TT-RX can be connected only to ft2u ft2x ft3x firmware

bit 31:17	addr-hi	highest 15 bit of the receiver FTSW address
bit 16:12	addr-lo	port number of the receiver FTSW (1..20)
bit 11	(dsel)	always 0
bit 10	bsy	busy (i.e., COPPER FIFO full)
bit 9:4	(cntseu)	always 0
bit 3	anyerr	error (belle2link lost or down, etc)
bit 2	bdown	ttrxlink from this FTSW is down
bit 1	blost	ttrxlink from this FTSW was lost
bit 0	alost	ttrxlink from TT-RX was lost

- combined status `pstat` is generated in `o_b2tt`
- pstat corresponds to dstat viewed from upstream

bit 31:12	addr	address of currently selected pregb
bit 11	dsel	if 0, addr is for this FTSW
bit 10	anyerr	error anywhere in this FTSW or downstream
bit 9:4	seucnt	total number of SEU in the downstream
bit 3	bsyin	busy anywhere in this FTSW or downstream
bit 2	ttdown	b2tt from upstream is down
bit 1	ttlost	b2tt from upstream was lost
bit 0	cklost	clock was lost (unused in upstream)

8 Error diagnosis register / user selectable register

8.1 Structure of error diagnosis register and user selectable register

- Error diagnosis register and user selectable register have 3-bit `depth (bit 31:29)`, 5-bit `type (bit 28:24)` and 24-bit `data (bit 23:0)`
- 29 type registers are defined as below

- TTREG with a larger hex(adecimal) number has a higher priority
- Defined in `b2tt_symbols.vhd` as `ttreg_t` type with `TTREG_` prefix
- Not all TTREG types are described in this document yet

TTREG	hex	description
CLOST	1d	clock error
TDOWN	1c	b2tt is down
TLOST	1b	b2tt was lost
TERR	1a	one of tagerr timerr badver clklost
FERR	19	feeerr is asserted now
RERR	18	feeerr was reported but cleared now
LDOWN	17	b2link down
LLOST	16	b2link lost
EBIT	15	error bit
BUSY	14	busy
IDLY	13	uplink b2tt delay wrt clock
RDLY	12	collect b2tt delay wrt clock
ALIVE	11	alive bits of connected nodes
BOUND	10	bound bits of connected nodes
TMASK	0f	Trigger distribution mask
EMASK	0e	Error collection mask
JTAGE	0d	JTAG-over-b2tt/lvds mask
SEM	0c	SEU mitigation status
TTAG	0b	Tag of distributed trigger
FTAG	0a	Tag of processed trigger by FEE
WCNTH	09	Belle2link write count higher 24-bit
WCNTL	08	Belle2link write count lower 24-bit
ETIME	07	utime when error occurred
BTIME	06	utime of boot of the firmware + ttup
EREG	05	error register when error occurred
PREG	04	current error register sent to up-link
ID	03	`type(23:16), board(15:0)` (type=0x4f for ft3o, board=0x102 for FTSW#102)
VER	02	`ftsw(23:20), firmware(17:8), build(5:0)` (ftsw=3, firmware=43, build=1 for ft3o043a)
USR	01	user defined register for debug

8.2 Error diagnosis register

- `prega` is a prioritized status register to show selected information depending on the error type
- Therefore, information about lower priority errors are masked
- Priority is: `TMASK CLOST TDOWN TLOST` (generated in `o_payload`) `TERR FERR RERR LDOWN LLOST` (generated in `b2tt_payload`)
- `TMASK` of `prega` is used to inform the LOCAL mode to the upstream
- When there is no error, `FTAG` is used to inform the id of the last processed trigger

8.3 User selectable register

- `pregb` is a user selectable register to retrieve further information over b2tt
- Section of the register is made via route selection command 5 `dsel` and register selection command 7 `regsel`

9 ttrxlink payload (xpayload)

- xpayload is the payload of ttrxlink from ttrx to FTSW
  - Raw data consists of 18-bit of xen3:0 xbsy3:0 xful3:0 xemp3:0 xlf xup
  - Since number of bit is limited, xen, xbsy, xful, xemp combinations which do not logically occur are used for other purposes
- ttrxlink status itself are described by three signals: alive adown bound
- ttrxlink information is merged into TTREG of b2tt by FTSW
- ttrxlink information can be viewed by following two registers
- Most of 4-bit information corresponds to 4 HSLBs on a COPPER
  - xrega (generated in `x_dataout`)

bit 31:29	rsv	always 0
bit 28:24	ttreg	fixed to EBIT (0x15)
bit 23:20	hsmask	HSLB is masked
bit 19:16	hsnemp	COPPER FIFO is not empty
bit 15:12	hsful	COPPER FIFO is full
bit 11:8	hsdown	Belle2link is down at HSLB
bit 7:4	hslost	Belle2link was lost at HSLB
bit 3:0	regerr	software error set into TTRX

  - xregb (generated in `x_dataout`), regsel=0 by default, unspecified bit is always 0

regsel=0		
bit 28:24	ttreg	TMASK (0x0f)
bit 23:6	xdata	raw data of xpayload
bit 5:4	staiddr	3 if link is up
bit 3	busyin	busy (COPPER FIFO is full)
bit 2	alost	same as xstat bit 0
bit 1	alive	1 if link is up
bit 0	async	1 if link is up
regsel=1		
bit 28:24	ttreg	TTAG (0x0b) for debug
reg 23:12	cntbsy	
reg 11:0	cntful	
regsel=2		
bit 28:24	ttreg	IDLY (0x13) for debug
bit 15:14	stacntb2	
bit 13:8	cntwidth	
bit 6:0	cntdelay	
regsel=3		
bit 28:24	ttreg	RDLY (0x12) for debug
bit 18	regerr4	
bit 17	lbusy	
bit 16	sbusy	
bit 13:12	staiddr	same as regsel=0 bit 5:4
bit 11:10	sigbit2	
bit 9:8	cntbit2	
bit 7:0	buflink	

10 List of FTSW registers

10.1 #00 ftswid (RW)

bit 31:0 ftswid value 0x46545357 (ASCII code for "FTSW")

Initial value can be changed to anything else to test read/write function. Not to be changed since the value is checked by ftprogs for existence of the FTSW.

10.2 #01 cpldver (R)

bit 31:16 board-id board id + 0x300 (FTSW3) / 0 (FTSW2)
bit 15:0 cpldver CPLD (FTSW3) / S3-FPGA (FTSW2) firmware version

10.3 #02 conf (RW)

write 0x41	to disable configuration flash memory
write 0x87	to assert PROGRAM* of FPGA
write 0x86	to deassert PROGRAM* of FPGA
write 0x08+m012	to set M(2:0) of FPGA to m012
read bit 7	DONE of FPGA (program is done and FPGA is ready)
read bit 6	not-busy (FPGA is not busy for programming)
read bit 5	always 0
read bit 4	program is controlled by the user program
read bit 3	INIT* of FPGA
read bit 2:0	M(2:0) of FPGA, i.e., 7 for serial-slave and 6 for selectMAP-slave (parallel)

10.4 #03 cclk (W)

Write to this register generates CCLK to FPGA with lowest 8-bit (or 1-bit) for programming in selectMAP-slave (serial-slave) mode

10.5 #10 fpgaid (RW)

bit 31:0 fpgaid ASCII code of FTSW FPGA firmware name

e.g., 0x46543255 for FT2U firmware, can be changed to anything else, but not recommended to change

10.6 #11 fpgaver (R)

bit 31:16	boardid	board ID
bit 15:0	fpgaver	FPGA firmware version

10.7 #12 setutim (RW)

(31:0) (32-bit) unix-time, i.e., time as the number of seconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC), as the reference for the time of the system, and also to record when the time was set

10.8 #13 clkfreq (RW)

bit 23:0 initial value: 0x0952980, lowest 24-bit of 0x7952980 = 127216000 to define the clock frequency to increment utime counted in the unit of second

10.9 #14 utime (R(W))

bit 31:0 utime current unix-time counted by FTSW

Writing any value to this register will hold the value of UTIME, CTIME, UDEAD, CDEAD, TINCNT and TOUTCNT for one second, to properly calculate the trigger rate, dead time, etc (see frozen bit below). Read CTIME as the last register and checking bit-31 to be '1' to ensure the all register are held.

10.10 #15 ctime (R)

bit 26:0	ctime	fine time in clock unit within utime
bit 31	frozen	CTIME etc are frozen

10.11 #16 exprun (R/W)

bit 31:22	exp	experiment number (up to 1023)
bit 21:8	run	run number (up to 16385)
bit 7:0	sub	sub-run number (up to 255

10.12 #17 omask for ftu (R/W)

bit 29	clkmask	(for experts only)
bit 28	cmdmask	command broadcast flag
bit 27:24	rxmask	mask to disable each hslb
bit 23:16	omask2	(for experts only)
bit 15:12	xmask	mask to disable the ttrxlink port
bit 7:0	omask	mask to disable the b2tt port

10.13 #17 omask for fto (R/W)

bit 31	query	(for experts only)
bit 30	usecmask	(for experts only)
bit 29	usetmask	(for experts only)
bit 28	bcast	broadcast flag
bit 27:16	tmask	(for experts only)
bit 15	uselocal	local (1) or global (0) mode
bit 11:0	omask	mask to disable the b2tt port

10.14 #17 omask for ftx (R)

bit 31:27	dselout	xpayload source choice
bit 26	errmask	
bit 24	cmdmask	enable sending rxmask
bit 23:20	rxmask	mask to disable each hslb
bit 19:0	omask	disabled port

10.15 #18 addr (RW)

bit 31:12	addr	destination address of command and reset
bit 11:4	cmd	8-bit command
bit 3:0	cmdhi	bit 35:32 of 36-bit data for the command

10.16 #19 cmd (RW)

Command is distributed by writing any value to this register.

bit 31:0	cmdlo	bit 31:0 of 36-bit data for the command

10.17 #1a jtag (RW)

Register to be used by the `jtagft` program

10.18 #1b tdo (R)

Register to be used by the `jtagft` program

10.19 #1c jctl (RW)

Register to control the jitter cleaner chip, do not touch

10.20 #1d jreg (R)

Register to control the jitter cleaner chip, do not touch

10.21 #1e jpd (RW)

Register to reset the jitter cleaner chip to resolve the four-fold ambiguity in the jitter cleaned clock phase

10.22 #1f jpll (R)

bit 31	stajpll	'1' if PLL is locked
bit 30	stajdcm	'1' if DCM is locked
bit 29:28	ckmux	clock source - 0: IN, 1: on-board, 2: optical
bit 27:24	phase	should be x"c" if stable
bit 23:16	retry	should be 0 if stable
bit 15:0	count	should be x"8000" if stable

10.23 #20 reset ((R)/W)

(* bit is to generate a short pulse)

bit 31	nofifo	no FIFO full to block the trigger
bit 30(u)	autorst	autoreset mode (runreset upon link changes)
bit 29	notagerr	tag error is ignored
bit 28	selreset	reset not broadcasted
bit 24(u)	usetlu	connect to TLU at AUX port
bit 23	ebup	EB is connected (for software use)
bit 21	paused	Run is paused (for software use)
bit 20	running	Run is running (for software use)
*bit 17	clraddr	clear address
bit 16	regbusy	artificial busy by software
*bit 15	clrictrl	(clear ictrl not in ft3o)
*bit 14	incdelay	
*bit 13	caldelay	
*bit 12	setaddr	set address
*bit 11	errreset	distributed error reset (but not runreset)
*bit 10	gtpreset	Belle2link GTP/GTX reset
*bit 9	feereset	FEE reset
*bit 8	b2lreset	Belle2link reset
*bit 5	locreset	local error reset (not distributed)
*bit 4	cntreset	FTSW counter reset
*bit 3	trgstop	stop the run
bit 2	genbor	generate begin-of-run trigger at run start
*bit 1	trgstart	start a run
*bit 0	runreset	run reset

10.24 #21 rstutim (R)

`utime` when last `runreset` was issued

10.25 #22 rstctim (R)

`ctime` when last `runreset` was issued

10.26 #23 etime (R)

`utime` when first error occurred after reset

10.27 #24 etimc (R)

`ctime` when first error occurred after reset

10.28 #25 erega (R)

`prega` when first error occurred after reset

10.29 #26 btime (R)

`utime` when firmware was programmed and b2tt was established for the first time

10.30 #27 cnttt (R)

Trigger tag of the last event written into Belle2link

10.31 #28 seltrg (RW)

write anything: reset dummy trigger cycle (see trigft.c for more detail of dummy trigger setting)

bit 31:20	trgopt	optional parameter for dummy trigger
bit 17:8	rateval	dummy trigger rate, linear part
bit 7:4)	rateexp	dummy trigger rate, exponent part
bit 2:0	seltrg	type of trigger source 0: none 1: IN 1-2 pair 2: AUX 7-8 pair 3: TLU trigger 4: pulse dummy trigger (uniform over revolution) 5: revo dummy trigger (fixed phase in revolution) 6: random dummy trigger (uniform random interval) 7: poisson dummy trigger (poisson interval)

10.32 #29 tlimit (RW)

Number of triggers to be generated, or "-1" for unlimitted number of triggers

10.33 #2a tincnt (R)

Number of trigger input since last run reset, including those that cannot be generated while busy

10.34 #2b toutcnt (R)

Number of trigger output since last run reset

10.35 #2c tlast (R)

Number of remaining triggers when `tlimit` is set

10.36 #2d stafifo (R)

bit 31	fifoful	FIFO is full
bit 30	fifoorun	FIFO overrun
bit 28	fifoemp	FIFO is empty
bit 25:24	fifoahi	n-th 32-bit is being read

10.37 #2e fifo (R)

bit 31:0	fifo	four 32-bit words are sequentially read out 0xffffffff if empty
word 0		
–bit 31		0 if fifo data is valid
–bit 30:4	ctime	ctime of trigger
–bit 3:0	trgtyp	type of trigger
word 1	utime	utime of trigger
word 2	trgtag	trigger tag starting from 0
word 3	tlutag	TLU tag (only in ft2u, using TLU)

10.38 #2f useold (R)

bit 31:24	regsel	selected register for `pregb`
bit 12:0	useold	old b2tt firmware is found at port

10.39 #30 dbsy (R)

bit 31	busy	trigger is temporarily stopped
bit 30	anydbsy	any busy from downstream
bit 29	pipebsy	trigger pipeline busy
bit 28	runreset	run reset signal, should not stay high
bit 27	errreset	error reset signal, should not stay high
bit 26	reset	any reset signal, should not stay high
bit 25	trig	trigger signal, should not stay high
bit 24	trgmask	trigger is ignored if high
bit 23:20	trgtype	trigger type of the last trigger
bit 19:0	dbsy	each busy from downstream

10.40 #31 derr (R)

bit 31	uselocal	in local mode
bit 28	running	during a run
bit 27	clkerr	clock error
bit 26	ckdown	clock is down
bit 25:24	noictrl	ICTRLDELAY status is bad
bit 23	ttdown	b2tt from upstream is down
bit 20	trigshort	trigger interval is too short
bit 19:0	derr	each error from downstream

10.41 #34 alive (R)

`alive` status of each downstream port

10.42 #35 bound (R)

`bound` status of each downstream port

10.43 #39 pstat (R)

Status register to be delivered to the upstream b2tt connection

bit 31:12	addr	address of this ftsw or forwarded node
bit 11	dsel	status forwarded (1) or of this ftsw (0)
bit 10	busy	status of fast busy signal
bit 9:4	seucnt	sum of SEU occurred in this tree
bit 3	anyerr	any error occurred in this tree
bit 2	ttdown	b2tt is down somewhere in this tree
bit 1	ttlost	b2tt was lost somewhere in this tree
bit 0	cklost	clock was lost in this ftsw

10.44 #3a prega (R)

Error diagnosis register, for details see elsewhere

10.45 #3b pregb (R)

User selectable register, for details see elsewhere

10.46 #40–#53 dstat (R)

Status register received from each downstream port

10.47 #54–#67 drega (R)

Error diagnosis register received from each downstream port

10.48 #68–#7b dregb (R)

User selectable register received from each downstream port

10.49 #7c-#8f odead (R)

Dead time (duration of busy) from each downstream port

10.50 #90 atime (R)

Time reference to calculate the trigger rate and dead-time fraction

10.51 #91 atrigi (R)

Trigger count to calculate the trigger rate

10.52 #92 abusy (R)

Total busy

10.53 #94 cbusy (R)

Busy from COPPER (this register is not properly working yet)

10.54 #95 pbusy (R)

Busy due to `maxtrig` setting

10.55 #96 fbusy (R)

Busy due to FTSW FIFO full

10.56 #97 rbusy (R)

Total time of PAUSE state

10.57 #9e trgdelay (RW)

bit 31:16	trgdelay	delay for AUX trigger in clock unit
bit 11:8	selila	chipscope source selection
bit 7:0	selreg	pregb selection

10.58 #9f maxtrig (RW)

bit 31:24	maxtrig	maximum allowed number of triggers within latency
bit 23:0	latency	period for maxtrig in clock unit

11 8b10b code table

K28.5 0011111010 1100000101 = bsydn bsyup

K28.0	K.1c	(rd+) 001111 0100	(rd-) 110000 1011
K28.1	K.3c	(rd+) 001111 1001	(rd-) 110000 0110
K28.2	K.5c	(rd+) 001111 0101	(rd-) 110000 1010
K28.3	K.7c	(rd+) 001111 0011	(rd-) 110000 1100
K28.4	K.9c	(rd+) 001111 0010	(rd-) 110000 1101
K28.5	K.bc	(rd+) 001111 1010	(rd-) 110000 0101
K28.6	K.dc	(rd+) 001111 0110	(rd-) 110000 1001
K28.7	K.fc	(rd+) 001111 1000	(rd-) 110000 0111
K23.7	K.f7	(rd+) 111010 1000	(rd-) 000101 0111
K27.7	K.fb	(rd+) 110110 1000	(rd-) 001001 0111
K29.7	K.fd	(rd+) 101110 1000	(rd-) 010001 0111
K30.7	K.fe	(rd+) 011110 1000	(rd-) 100001 0111