

b2tt ワーキング・ドキュメント

中尾 幹彦 (高エネルギー加速器研究機構)

2017.8.2 (version 0.06)

1 この文章について

この文章では b2tt の仕組みを可能な限り詳細に解説し、以下の理解を手助けすることにある

- フロントエンド機器のファームウェアでの b2tt の使用方法
- FTSW および関連ソフトウェアの使用法
- FTSW のファームウェアの内部構造

FTSW 使用者のためには技術的すぎる情報も含まれているので、必要なところだけを参考にとすること

2 b2tt の仕組み

2.1 b2tt とは

- Belle II のトリガ・時間 (Trigger Timing) 分配制御用プロトコル
- 254 Mbps のシリアルデータリンクによる双方向通信
- SuperKEKB RF 由来の 127 MHz 外部クロックに同期
- 共通のデータを分配し、受信側で必要に応じて異なる動作をする
- 独立なデータを収集し、収集の各段階で決められた方式でまとめる

2.2 信号線

- FTSW どうしは 1 本の CAT-7 ケーブル (4 対の信号線) または 2 対の光ファイバで接続
 - 4 本の信号線の内訳: クロック信号、分配信号、収集信号、変調 TDO 信号
 - すべての信号線は AC カップル
- FTSW とフロントエンド (FEE) とは 1 本ないし 2 本の CAT-7 ケーブルで接続
 - 2 本目の CAT-7 ケーブルは JTAG プログラム用で、不用ならば必要ない

2.3 クロック信号

- CAT-7 7-8 ペアまたは 光ファイバ 1 で常に送り続ける

- 一番上流の FTSW は **clock master** ボードから **IN** ポートでクロックを受け取る
 - clock master ボードは SuperKEKB の 509 MHz RF 信号を受け取り、127 MHz のクロックを生成する (**external mode**)
 - RF 信号が不在の時には clock master 内の 127 MHz 水晶発振器からクロックを生成する (**internal mode**)
 - 一番上流の FTSW にクロック入力がない場合には、FTSW 内の 127 MHz 水晶発振器からクロックを生成する (**pocketdaq mode**)
- それ以外の FTSW は **IN** ポート (type-R 以外) あるいは **上段の SFP** ポート (type-R) でクロックを受け取る
 - クロック信号に対して各段の FTSW でジッタクリーナによってジッタの削減を行う
 - FTSW から 10m の CAT7 を通して受け取るクロックの典型的なジッタは RMS で約 20 ps (オシロスコープでの測定値)
 - クロックが一瞬でも途切れたらジッタクリーナの PLL ロックの外れとして検知されクロックエラーとなる
 - 入力のクロックケーブルへの接触などでクロックエラーになることがある
- ジッタクリーナの出力クロックの位相に 4 重の不定性があるので、FTSW は入力クロックとの位相差を検知して所定の位相差になるまでジッタクリーナを自動的にリセットする
 - 一旦位相差が固定されると、その後は変化しない

2.4 分配信号

- CAT-7 3-6 ペアまたは 光ファイバ 2 で常に送り続ける
- クロック信号に同期した 254 Mbps のデータ信号
 - 8b10b エンコードを行う
 - SuperKEKB の周回と同じ周期の 2560 ビットのフレームデータを使用
 - 1 フレームは 16 個の 160 ビットの packets から構成
 - 1 packet は 16 個の 8b10b コードから構成
- データパケットの構造
 - packet は開始コードの K28.1 シンボルで始まる (10b のパターンでコード境界がユニークに決まる)
 - packet 内の任意の 8b10b コードの場所にトリガコードを埋め込み可能
 - トリガ間隔は最低 24 クロックを要求しているので 1 フレームで最大 4 個のトリガコードが可能
 - 開始コード位置にトリガコードを埋め込む場合は次の場所に代替開始コード K28.5 シンボルを使用する
 - 開始コード、トリガコードの使用していない場所に 11 個のデータコードを埋め込む
 - トリガコードの数が 0-2 個の場合には 2-0 個のアイドルコード (K28.3 シンボル) を埋め込む
 - トリガコードの数が 0-3 個の場合には最後に 8 ビット CRC を埋め込む (トリガコードの数が 4 個の場合には CRC のチェックは行わない)

- リンクアップの条件
 - パケットが開始コード K28.1 で始まっているか 2 番目の 8b10b コードが代替開始コード K28.5 である
 - CRC が合っている
 - 以上の条件が 240 パケット以上続くとリンクアップとなる
 - 以上の条件が 1 回でもみたされないとリンクダウンとなる
 - 8b10b のディスパリティ不整合と未定義コードのチェックは行っていないが、これらが起ると K28.1 や K28.5 ビットパターンが不正に生成されてリンクダウンにつながる

2.5 トリガ信号、トリガコード

- トリガ信号には 1/4 クロック (1.9ns) の時間情報を持つ高精度 (fine-timing) トリガと 1 クロック単位 (7.8ns) の時間情報を持つ低精度 (coarse-timing) トリガがある。
- トリガコードは分配信号のデータパケットの構造内の任意の 8b10b コード境界に埋め込まれる
 - トリガコードは 8b10b データコードで 8b の最上位ビットが 1
 - 残り 7 ビットのうち 3 ビットで 10b コード内 (5 クロック) のタイミング、4 ビットでトリガタイプを表す
 - 高精度トリガの 1 クロック内の時間情報 ($t_{\text{fine}}=0,1,2,3$) は 4 種類のトリガタイプで識別する
- トリガタイプは以下の通り (b2tt_symbols.vhd で定義)

PID0	0x0	高精度トリガ、 $t_{\text{fine}}=0$
PID1	0x4	高精度トリガ、 $t_{\text{fine}}=1$
PID2	0x8	高精度トリガ、 $t_{\text{fine}}=2$
PID3	0xc	高精度トリガ、 $t_{\text{fine}}=3$
RSV0	0x2	予備 0
RSV1	0x6	予備 1
RSV2	0xa	予備 2
RSV3	0xe	予備 3
ECL	0x1	ECL トリガ、低精度
CDC	0x3	CDC トリガ、低精度
DPHY	0x5	物理トリガを遅延させたもの
RAND	0x7	ランダムトリガ
TEST	0x9	テストトリガ
RSV4	0xb	予備 4
RSV5	0xd	予備 5
NONE	0xf	トリガではない

2.6 JTAG 信号、JTAG コード

- JTAG 信号は TCK TMS TDI の分配信号と TDO の収集信号から構成されるデータパケットの構造とは非同期の信号である
- 分配信号は TMS TDI の 2 bit の状態を TCK のタイミングで 4 種類の 8b10b の K コードとしてデータパケットに埋め込む
 - (TMS TDI) が (0 0) (0 1) (1 0) (1 1) の時に、それぞれ K23.7 K27.7 K29.7 K30.7 の K コー

ドを使用する

- データパケットを破壊することなく 1 パケット内に最大 4 個の K コードを埋め込めるため、TCK は最大周波数は 3.175 MHz と Xilinx Platform USB ケーブルと impact プログラムでのデフォルトの約半分の速度となる
- トリガコードと JTAG 信号は共存できないので、トリガ生成中は JTAG を使用することができない
- TDO の受け取りには遅延を減らすために CAT7 または光ファイバの 4 本目の線を用いる
 - TDO 自体は DC レベルの信号なので、AC カップルされた CAT7/ファイバの接続で受け取れるように、127 MHz のクロックにより変調され、復調される。

2.7 分配ペイロード

- 8b10b データコードで 8b の最上位ビットが常に 0 のものの残り 7 ビットを 11 個集めて 77 ビットのペイロードとして扱う
- セクション FEE への分配信号のペイロード に詳細

2.8 分配信号のマスク

- パケット 4 から 7 までのデータ信号をアイドルパターンに置きかえることによりリンクを保持したまま分配信号のマスクを行うことができる (p_otrg.vhd)
- この機能により、特定のポートにだけ特定の要求 (パケット 4 のコマンド) を送ることができる
- 常に一定であるアイドルパターンの disparity に合わせるために、アイドルパターンであるデータ信号のパケット 2 と 6 の CRC の付加の有無によって disparity の値を調整する (o_encode.vhd)
- ラン中と JTAG シーケンス中にはトリガコードまたは JTAG コードにより disparity が不定になるのでアイドルパターンで置きかえてしまうと disparity が合わなくなる
- したがってラン中または JTAG シーケンス中はデータ信号をマスクすることができない
- トリガ信号および JTAG 信号を送信側でマスクすることはできないため、受けとったこれらの信号を受信側で無視するように設定することによりマスクする

2.9 収集信号

- CAT-7 1-2 ペアまたは 光ファイバ 2 で常に送り続ける
- 分配信号と同様、クロック信号に同期した 254 Mbps のデータ信号
 - 8b10b エンコードを行う
 - SuperKEKB の周回を 16 分割した 160 ビットのパケットデータを使用
 - 1 パケットは 16 個の 8b10b コードから構成
- 収集信号とクロックとの位相差は接続されるケーブル長などに依存するので、リンク確立時に位相のスキャンを行い通信が安定する位相差の範囲の中央になるように自動調整される
- データパケットの構造

- パケットは開始コードの K28.1 シンボルで始まる (カンマコードとも呼ばれ、10b のビット列でコード境界がユニークに決まる)
- 開始コードのあとにデータコード 14 個が続き、従って 1 パケットで 112 ビットの復元データから構成される
- 最後に 8 ビット CRC を埋め込む
- パケット内の任意の 2 ビット (クロック) 単位の場所にビジーコードである K28.5 を埋め込み可能
- ビジー信号が埋め込まれたパケットには CRC は埋め込まれず、CRC の照合も行われない
- ビジー信号が 2 個以上埋め込まれたパケットは破棄される
- リンクアップの条件
 - 8b10b コード境界が正しく認識されていない状態 ($\text{cntbit2} = 5$) あるいは K28.1 開始コードが見つかり 8b10b コード境界を設定し ($\text{cntbit2} < 5$)、正しいコード境界でない場所で K28.1 開始コードが見つかりリセットする
 - 開始コードのあとに 14 個連続した 8b10b データコードが続き、CRC があれば CRC が符合するか、ビジーコードによりデータコードが 14 個未満になっている場合にはパケットが正当であると判定 ($\text{cntoctet} < 16$)
 - 256 個連続してパケットが正当だとリンクアップと判定 ($\text{cntvalid} = 255$)
 - 以上の条件が 1 回でも満たされないとリンクダウンになる

2.10 ビジー信号とビジー状態

- ビジー信号は FTSW の各段階で一度収集され、論理和としてまとめた結果を上流に送り出す
- 遅延信号を最小にするために 8b10b コード境界に限らず任意のクロック端界に埋め込むことができる
- ビジー信号のオン・オフには K28.5 の disparity プラスとマイナスのコードを用いる
- K28.5 の 10b コードと同じパターンがビット列には現われることはない (disparity の制御に間違いがあると 任意の場所に挿入された K28.5 により K28.1 と見なせるビット列が生成され、tlost やリセットできないビジー状態となる可能性がある)
- 各ノードは現在のビジー状態を runreset 信号により初期化する

2.11 クロックとリンクの状態

- $\text{ckup}=1$ はクロックが正しく受けとれていて PLL が正常にロックされている状態
- $\text{ttup}=1$ は上流からの b2t プロトコルが正常に受けとれている状態 (下流側の信号)
- $\text{alive}=1$ は下流からの b2t プロトコルが正常に受けとれている状態 (上流側で見た信号)
- $\text{bound}=1$ は下流での $\text{ttup}=1$ 信号が up-stream に正しく届いている状態 (上流側で見た信号)
- $\text{adown}=1$ / $\text{bdown}=1$ は reset 時に $\text{alive}=1$ / $\text{bound}=1$ だった信号が落ちている状態で、リセットすると $\text{alive}=0$ / $\text{bound}=0$ になる
- $\text{alost}=1$ かつ $\text{adown}=0$ / $\text{blost}=1$ かつ $\text{bdown}=0$ の状態は reset 時に $\text{adown}=1$ / $\text{bdown}=1$ からリンクが復活した状態で、リセットにより $\text{alost}=0$ / $\text{blost}=0$ になる

- まとめると以下の通り

ckup	入力クロックが b2tt_clk の PLL/DCM でロックされている状態
ttup	上流からのリンクが成立している状態
ttlost	上流からのリンクが切れた跡
tttdown	上流からのリンクが切れている状態
alive	条件によらず D -> U のリンクが成立している状態
alost	dmask されておらず D -> U リンクが一度でも切れた跡
adown	dmask されておらず D -> U リンクが切れている状態
bound	条件によらず D <-> U 双方の リンクが成立している状態
blost	dmask されておらず D <-> U どちらかのリンクが切れた跡
bdown	dmask されておらず D <-> U のリンクが切れている状態

3 b2tt の FEE への組み込み

3.1 ファームウェアの構造

- b2tt ファームウェアは以下のツリー構造になっている
- なお一部のファイルには複数のコンポーネントが含まれており、必ずしもファイルと対応していない

```

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_depaket                    (packet decoder => command)
    +-- b2tt_detag                      (trigger tag checker)
+-- b2tt_encode                         (encoder top)
    +-- b2tt_encounter                  (counters for encoding)
    +-- b2tt_enpaket                    (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 ファイル

- b2tt のパッケージ内の b2tt ディレクトリに含まれる以下のファイルのうち使用する FPGA の種類に対応するものをすべて使用する

b2tt.vhd	メイン	共通
b2tt_8b10b.vhd	8b10b 変換	共通
b2tt_clk_s6.vhd	クロック処理	Spartan-6
b2tt_clk_v5.vhd	クロック処理	Virtex-5
b2tt_clk_v6.vhd	クロック処理	Virtex-6
b2tt_clk_x7.vhd	クロック処理	7-series
b2tt_ddr_s6.vhd	入出力処理	Spartan-6
b2tt_ddr_v5.vhd	入出力処理	Virtex-5
b2tt_ddr_v6.vhd	入出力処理	Virtex-6
b2tt_ddr_x7.vhd	入出力処理	7-series
b2tt_decode.vhd	b2tt デコード	共通
b2tt_encode.vhd	b2tt エンコード	共通
b2tt_fifo.vhd	b2tt FIFO	Spartan-6 以外共通
b2tt_fifo_s6.vhd	b2tt FIFO	Spartan-6
b2tt_iscan.vhd	位相スキャン	共通
b2tt_revo.vhd	周回信号処理	共通
b2tt_injv.vhd	入射 veto 処理	共通
b2tt_payload.vhd	b2tt ペイロード	共通
b2tt_symbols.vhd	信号型と定数の定義	共通

3.3 coregen ファイル

- coregen ファイルは chipscope のためだけに必要
- Spartan-6 (b2tts6-coregen)、Virtex-5 (b2ttv5-coregen)、Virtex-6 (b2ttv6-coregen)、7-series- (b2ttx7-coregen) 用のファイルがパッケージ内に用意されている
- 必ずしもパッケージ内の coregen ファイルがそのまま使用できるとは限らないので、必要に応じて再生成する
- 他の chipscope の中に組み込む場合にはパッケージ内の coregen ファイルを使用する必要は特にない

3.4 使用例

- 対応する各 FPGA 用に b2tt のみを組み込んだファームウェアのソースファイルを用意している
- 実際には Belle2link のパッケージに含まれる使用例を参考にした方がより現実的である
- コンパイル環境としては ISE 12.4 を想定しており、それ以降のバージョンの ISE でも問題なく使用できる

3.5 b2tt インターフェイス

- b2tt のインスタンスをひとつ (だけ) 生成することにより b2tt を使用できる

- b2tt には以下のように数多くの generic と port が使用されており、必須のもの、特定のユーザ専用のものおよびユーザの使用を想定していないものがあり、すべての信号を使用するわけではない
- 以下で (*) のついた信号は必須、(**) のついた信号は Belle2link 使用時には必須、(***) のついた信号は放射線エリア内で使用するファームウェアについて必須となる
- 各 generic の使い方は以下の通り (以下に記述のないものを変更してはいけない)

SUBSYSTEM	(*) b2tt_symbols.vhd 内の TTTEE_ のいずれかを設定
FWTYPE	(*) 4 ビット整数で、ファームウェアのタイプの識別に使用
VERSION	(*) 16 ビット整数で、ファームウェアのバージョンを設定
FLIPTRG	RJ-45 の TRG 信号の極性が反転している時にのみ '1' を設定
FLIPACK	RJ-45 の ACK 信号の極性が反転している時にのみ '1' を設定
CLKDIV1	divclk1 に出力する分周クロックの分母
CLKDIV2	divclk2 に出力する分周クロックの分母
USEICTRL	ファームウェアですでに IDELAYCTRL を使用済みの時には '0'
USEEXTCLK	クロックを外部で生成する場合には '1'
USE254IN	dblclk に 254 MHz クロックを出力する時にのみ '1'
SIM_SPEEDUP	シミュレーション用にリンクの手順を速くする場合 '1'

- 各信号は以下の通り (記述のない出力信号は open のままで良い)

b2ttver	b2tt のバージョン
clkp/clkn	(*) RJ-45 7-8 ペアからの クロック
trgp/trgn	(*) RJ-45 3-6 ペアからの b2tt 信号
ackp/ackn	(*) RJ-45 1-2 ペアへの b2tt 信号
extclk 他	外部クロック信号入力
id	(*) ボード ID
usrreg	8 bit、b2tt から書き込み可
usrdat	16 bit、b2tt から読み出し可で不要なら x"00" を設定
b2clkup	(*) b2tt がクロックを認識
b2ttup	(*) b2tt リンクが確立
sysclk	(*) 127 MHz の基準クロック
dblclk	254 MHz のクロック、Virtex-6 のみ
utime	Unix 準拠の秒単位の現在時間
ctime	1 秒内のクロック単位の現在時間
divclk1,2	分周クロック 1,2
exprun	実験 (bit31:22) ラン (21:8) サブラン (7:0) 番号
running	'1' ならばラン中
runreset	(*) ラン開始時のリセット
feereset	FEE 専用のリセット
b2lreset	(**) Belle2link のリセット
gtpreset	(**) GTP/GTX のリセット
rstmask	'1' ならば runreset はマスクされている
trgout	(*) トリガ信号 (1 クロック幅)
trgtyp	トリガタイプ
trgtag	トリガ番号 (0 から始まる)
trgmask	'1' ならば trgout はマスクされている

frame	1280 クロックごとのフレーム信号、全 FEE で同期
frame9	9 回に 1 回、全 FEE でフレームの間 '1' になる
revoloc	revosig のフレーム内での位置
revosig	SuperKEKB の周回信号
abortgap	SuperKEKB のアボートギャップ
injveto 他	入射 veto 信号
busysrc	8 ビット (最大 8 種類) のビジー信号
errsrc	8 ビット (最大 8 種類) のエラー信号
b2pllck	(**) Belle2link の PLL ロック状態
b2linkup	(**) Belle2link のリンクのステータス
b2linkwe	(**) FEE が Belle2link にデータを書き出している状態
b2lclk	(**) FEE が Belle2link 書き出しに使用するクロック
semscan 他	(***) SEU 対策コードとの信号のやりとりを行う
fifordy	(*) FTSW の FIFO データを読める状態
fifodata	(*) FIFO から読み出すトリガ毎の 96 ビット情報
fifonext	(*) FIFO 読み出し終了の信号
regdbg	x"00" を設定しておく
dbglink	Chipscope につなげておく 96 ビット情報

- ファームウェアのタイプ、バージョン、ボード ID は b2t 経由で読み取ることができ、ケーブル配線間違いやファームウェアの間違いなどの問題解決に役に立つので正しく設定すること
- 外部クロックを使用する時には USEEXTCLK='1' とし、~clkp/clkn を使用せず sysclk dbllck とその反転信号に対応する extclk extclkdbl extclkinv extdblinv および b2clkup に対応する extclkclk 信号を供給する
- injveto は入射 veto 信号で、この間は物理トリガは生成されない
- injveto の時間構造は injkick のタイミングで injvpos injvpre injvplen injvpfull injvgate の各パラメータで表わされる

4 FEE への分配ペイロード

4.1 ペイロードの構造

- 77 ビットで構成
- ビット 76 はブロードキャストフラグ (1 ならばブロードキャスト)
- ビット 75-64 はパケットの種類 (12 ビット)
- ビット 63-0 がパケットの種類ごとに定義されるデータ

4.2 パケットの種類

- TTAG FREQ RST CMD DISP INJV REVO SYNC の 8 種類を使用
- 以下、括弧内はビット位置
- SYNC パケット (ブロードキャストのみ) —
runreset(63) frame3(62) frame9(61) jtagen(60) trgen(59) utime(58-27) ctime(26-0)

- CMD パケット (ブロードキャスト有無どちらも可) —
対象アドレス (63–44) コマンド (43–36) データ (35–0)
- RST パケット (ブロードキャスト有無どちらも可) —
対象アドレス (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 パケット (ブロードキャストのみ) —
未使用 (63–34) B2TVER(33–24) clkfreq(23–0)
(クロック信号は厳密に 127216000 Hz ではなく、SuperKEKB の RF 周波数と NTP で得られる
時間とが最も良く同期が取れる 1280 の倍数を使用し、ラン中以外のタイミングで動的に変わ
りうる)
- TAG パケット (ブロードキャストのみ) —
exprun(63–32) ttag(31–0)
(ttag は送信したトリガ数で、フレームごとに受信したトリガの数との比較を行う)

5 FEE からの収集ペイロード

5.1 ペイロードのフォーマット

b2tt/b2tt_payload.vhd の中で定義されている。

111–92	myaddr	dsel に依存する
91–84	cnt_payload	
83	dsel	FEE では必ず dsel=0
82	ttdown	dsel に依存しない
81	anybsy	dsel に依存しない
80	anyerr	dsel に依存しない
79–70	reserve10	(10-bit)
69–64	cnt_seudet	dsel に依存しない
63–61	reserve3	reg1 は dsel に依存しない
60–56	reg1.type	(5-bit)
55–32	reg1.data	(24-bit) データ
31–29	reg2.depth	(3-bit) dsel=1 で +1 増える
28–24	reg2.type	(5-bit)
23–0	reg2.data	(24-bit) データ

6 エラー診断

- エラー時にはエラービットが立ち (anyerr=1)
- エラーの状態が優先度式レジスタで表示される
- ユーザー選択式レジスタでさらに詳細を調べることができる

6.1 TTREG のタイプ

- 優先度は CLOST TDOWN TLOST TERR EBIT FERR RERR LDOWN LLOST の順

- FEE での feerr(7:0) の現在値が 0 でない時には FERR を返す
- feerr が起きたが現在は 0 の場合には RERR を返す
- FERR、RERR は FTSW では生成されない

6.2 クロックのエラー (CLOST)

- クロックエラーは他のすべてのことに影響を与えるので最優先とする
- クロックのエラーはさまざまな PLL や DCM のロックが外れたことをもって判定する
- 主なものについては現在の状態および過去に起きたエラーを別々に表す (現在クロックに異常がある時にはそもそも b2tt を通して状態を得られない可能性が高い)
- COPPER/TTRX (tttxlink) からはクロックエラーは受け取らない
- 以下のビットパターンが定義されている

bit 23	jpdwn	FTSW jitter cleaner PLL is down
bit 22	jplst	FTSW jitter cleaner PLL was lost
bit 21	ckdwn	b2tt DCM lock is down (can it be visible?)
bit 20	cklst	b2tt DCM lock was lost
bit 19	plldwn	Belle2link GTX/GTP PLL lock is down (FEE only)
bit 18	pllst	Belle2link GTX/GTP PLL lock was down (FEE only)
bit 17	icterr	ictrl delay error
bit 16	ictdwn	DCM lock for ictrl delay 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 リンクのエラー (TDOWN, TLOST, LDOWN, LLOST)

- b2tt のエラーは他のことに影響を与える可能性が高いので優先度を高くする
- 優先度をつけるにあたって、ポートのビットパターンを共有しているので別々の TTREG タイプとして扱う
- tttxlink からのエラーを処理するためにエラーソースは最大 20 ポート
- b2tt からは最大 13 ポートで、残り 7 ビット未使用で常に 0
- bit 23:20 は tttxlink と b2tt とで意味が異なる

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 FEE からのエラー (FERR, RERR)

- FEE から b2tt に 8-bit で 8 種類のエラーを渡すことができる

- 収集する際に論理和をとるので、ビットパターンを変えて 8 種類以上のエラーを表現することは薦められない
- FTSW はエラーの種類ごとに動作を変えることはしない
- FERR は現在値、RERR は前回のリセットから一回でも起きた場合の状態を保持

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

6.5 その他のエラー (TERR)

- TERR はクロックとリンクの状態によるものでなく、FEE からのものでないエラー
- CLOST から TERR までの TTREG タイプですべてのエラーをカバーする

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 エラーのまとめ (EBIT)

- すべてのエラーを一度に見ることのできるまとめのレジスタ
- ユーザ選択レジスタのひとつだが、エラー診断レジスタには使用されない
- 下流の情報は含まない

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 旧 b2tt プロトコルのエラーの読みかえ

- 旧 b2tt プロトコルを使用する FEE との接続時には以下のエラー情報の読みかえを行う
- すべてのエラーを網羅できるわけではないが、主なものはカバーされる

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_ttllost	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 リンク状態レジスタ (dstat/xstat/pstat)

- b2tt リンクごとの dstat はリンク先全体の状態を表す
- addr は現在選択している dregb レジスタのソース (デフォルトは直接接続の FTSW または FEE)
- o_collect で生成

bit 31:12	addr	現在選択中の dregb のアドレス
bit 11	dsel	0 ならば addr は直近の下流のアドレス
bit 10	anyerr	下流のどこかでエラー
bit 9:4	seucnt	下流の SEU 数の総和
bit 3	bsyin	下流のどこかでビジー
bit 2	bdown	直近の下流で上流からの b2tt が down
bit 1	blost	直近の下流で上流からの b2tt が lost
bit 0	alost	直近の下流からの b2tt が lost

- 直近のリンク先が TT-RX の場合のステータス xstat となる
- xstat は x_collect で定義
- TT-RX は ft2u ft2x ft3x のファームウェアにのみ接続可能

bit 31:17	addr-hi	受信側 FTSW のアドレスの上位 15 bit
bit 16:12	addr-lo	受信側 FTSW のポート番号 (1..20)
bit 11	(dsel)	常に 0
bit 10	bsy	ビジー (COPPER FIFO がフル)
bit 9:4	(cntseu)	常に 0
bit 3	anyerr	エラー (Belle2link down/lost など)
bit 2	bdown	ttrx 側で見た ttrxlink が down
bit 1	blost	ttrx 側で見た ttrxlink が lost
bit 0	alost	ftsw 側で見た ttrxlink が lost

- これらをまとめた自分自身のステータス pstat は o_b2tt で生成
- pstat が直近の上流で見た dstat に相当

bit 31:12	addr	現在選択中の pregb のアドレス
bit 11	dsel	0 ならば addr は自分自身のアドレス
bit 10	anyerr	なんらかのエラー
bit 9:4	seucnt	下流の SEU 数の総和
bit 3	bsyin	なんらかのビジー
bit 2	ttdown	直近の上流からの b2tt が down
bit 1	ttlost	直近の上流からの b2tt が lost
bit 0	cklost	クロックが lost (上流では使用しない)

8 エラー診断レジスタ・ユーザ選択レジスタ

8.1 エラー診断レジスタとユーザ選択レジスタの構造

- エラー診断レジスタおよびユーザ選択レジスタは 3-bit の depth (bit 31:29)、5-bit の type (bit 28:24)、24-bit の data (bit 23:0) から構成される
- 以下のように 29 種類のレジスタが定義されている
- 16 進 (hex) 番号の大きな TTREG がより高い優先度を持つ
- b2tt_symbols.vhd 内で ttreg_t 型で TTREG_ のプレフィックスをつけて定義される
- この文章ではまだすべての TTREG の説明は網羅されていない

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
JTAG	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

TTREG	hex	description
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 エラー診断レジスタ (prega)

- prega は優先度式ステータスレジスタで、エラーの種類により、その特定のエラーに関する情報を選択して表示する
- したがってより優先度の低いエラーの表示はマスクされる
- 優先度は: TMASK CLOST TDOWN TLOST (これらは o_payload で生成) TERR FERR RERR LDOWN LLOST (これらは b2tt_payload で生成)
- prega の TMASK は上流に LOCAL モードであることを伝えるために使用
- エラーのない時には FTAG として処理された最後のトリガ番号を返す

8.3 ユーザー選択レジスタ

- pregb はユーザ選択レジスタで、より多くの情報を b2tt 越しに得ることができる
- 表示するレジスタの選択は、コマンド 5 の dsel による経路選択とコマンド 7 の regsel によるレジスタ選択で行う

9 ttrxlink のペイロード (xpayload)

- xpayload は ttrxlink の ttrx から FTSW へ送られる payload
 - 生情報は xen3:0 xbsy3:0 xful3:0 xemp3:0 xlf xup の 18-bit で構成
 - bit 数が限られているため、xen、xbsy、xful、xemp の組み合わせのうち実際には生じない組み合わせを別の意味に使用
- ttrxlink 自体の情報は alive adown bound の 3 つの信号で表す
- ttrxlink の情報を FTSW によりまとめる際には b2tt の TTREG に置き換えられる
- ttrxlink の情報は以下の 2 つのレジスタで見られる
- ほとんどの 4-ビット情報は COPPER 上の 4 枚の HSLB に対応
 - xrega (x_dataout で生成)

bit 31:29	rsv	常に 0
bit 28:24	ttreg	EBIT (0x15) に固定
bit 23:20	hsmask	HSLB がマスクされている
bit 19:16	hsnemp	COPPER FIFO が空でない
bit 15:12	hsful	COPPER FIFO がフル
bit 11:8	hsdown	HSLB 側で Belle2link が down
bit 7:4	hslost	HSLB 側で Belle2link が lost
bit 3:0	regerr	TTRX で設定するソフトウェアエラー

– xregb (x_dataout で生成)、デフォルトは regsel=0、指定のないビットは常に 0

regsel=0 bit 28:24 bit 23:6 bit 5:4 bit 3 bit 2 bit 1 bit 0	ttreg xdata staiddr busyin alost alive async	TMASK (0x0f) 生の xpayload のデータ linkup 時には 3 ビジー (COPPER FIFO がフル) xstat bit 0 と同じ linkup 時には 1 linkup 時には 1
regsel=1 bit 28:24 reg 23:12 reg 11:0	ttreg cntbsy cntful	TTAG (0x0b) デバッグ用
regsel=2 bit 28:24 bit 15:14 bit 13:8 bit 6:0	ttreg stacntb2 cntwidth cntdelay	IDLY (0x13) デバッグ用
regsel=3 bit 28:24 bit 18 bit 17 bit 16 bit 13:12 bit 11:10 bit 9:8 bit 7:0	ttreg regerr4 lbusy sbusy staiddr sigbit2 cntbit2 buflink	RDLY (0x12) デバッグ用 regsel=0 bit 5:4 と同じ

10 FTSW レジスター一覧

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	loreset	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 flimit (RW)

Number of triggers to be generated, or "-1" for unlimited 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 dbusy (R)

bit 31	busy	trigger is temporarily stopped
bit 30	anydbusy	any busy from downstream
bit 29	pipebusy	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	dbusy	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 コード表

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