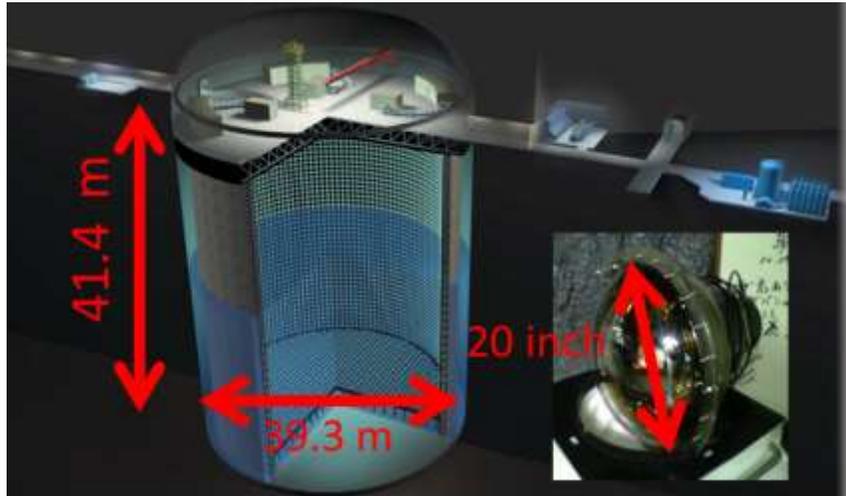


スーパーカミオカンデにおける 近傍超新星観測DAQの開発

Orii

スーパーカミオカンデと超新星爆発

- スーパーカミオカンデ(SK)
 - 水チェレンコフ検出器
 - 有感質量 32000ton
 - 光電子増倍管(PMT)13000本
ID:11129, OD:1885



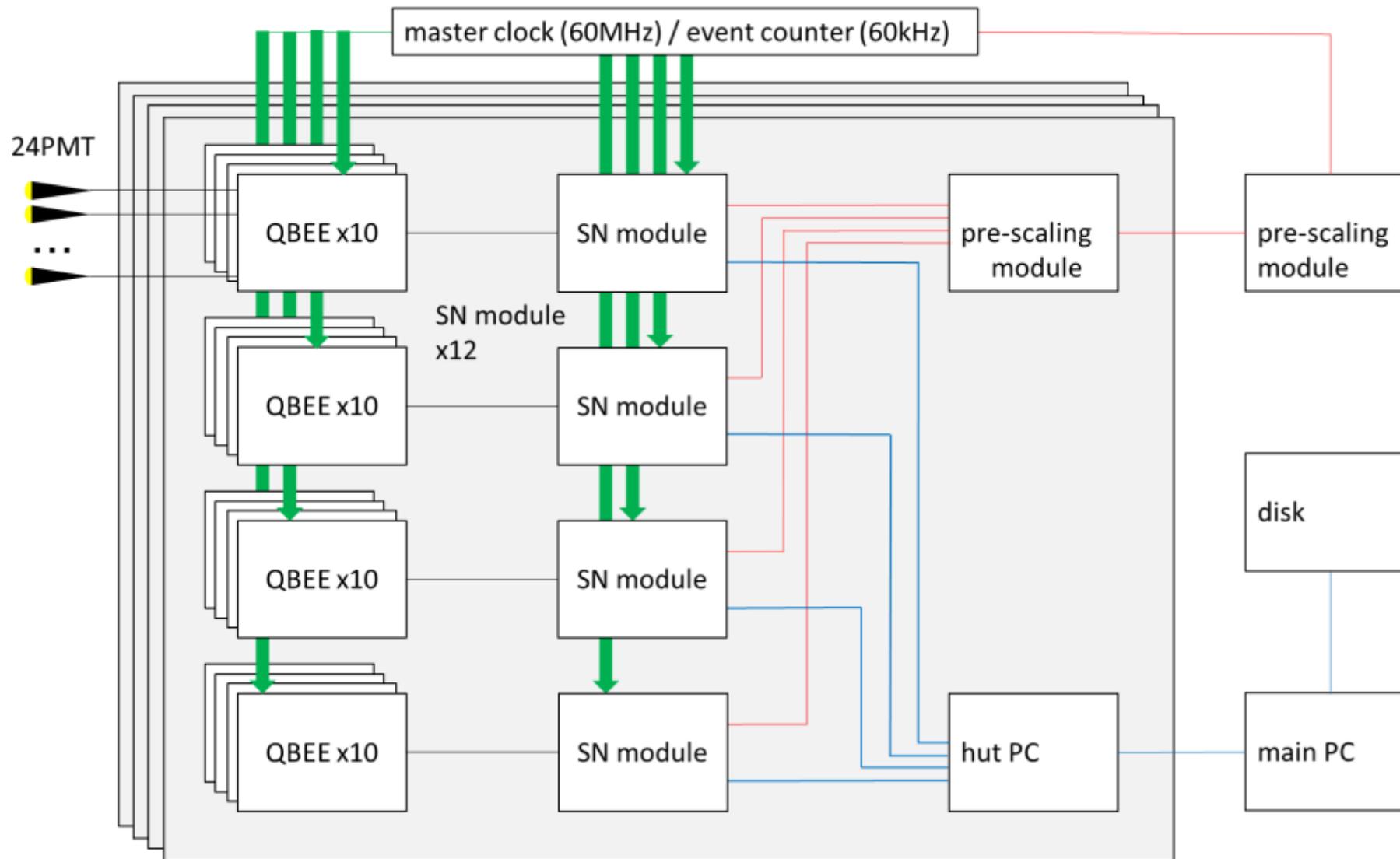
- 超新星爆発
 - 重い星($M > 8M_{\odot}$)が寿命を迎えて爆発する
 - エネルギーの99%がニュートリノとして放出される
 - 銀河系内で30~50年に1回
- 近傍で起こる可能性も予言されている
 - 候補
 - ベテルギウス(640光年)
 - アンタレス(550光年)
- 500光年で起こると...
 - SKにおいて 3×10^7 事象/10sec の観測が期待される
 - 既存DAQの処理能力は 6×10^6 事象/10sec まで
- DAQの改良が必要

システムへの要請、設計

- エレキのデッドタイムがない
- main DAQから独立して動作する
- 超新星のburstが検出できる

- PMT hitを記録する
 - レート一定
 - 1hit~1p.e.
 - 超新星ニュートリノ~20MeV~120hit/event
 - occupancy = 120hit/11129PMT ~ 1%
 - energy depositが測定できる
- イベントの間引きを行う
 - main DAQの併用でfluxの時間変化が求められる

概要



設計詳細

- QBEEからのdigital hitsum信号を用いる
 - TDCで生成される
 - 17nsの間に光を検出したPMTの数(0~24)
→digital hitsumはevent rateによらず一定のデータ量
問題となるbufferよりも上流で生成される
- SN module, QBEEともにmaster clock, event counterに同期する
 - それぞれのデータが比較できる
- SN moduleからの信号を集めてveto信号を生成
master clockに送ってTDCを止めることでpre-scalingを行う

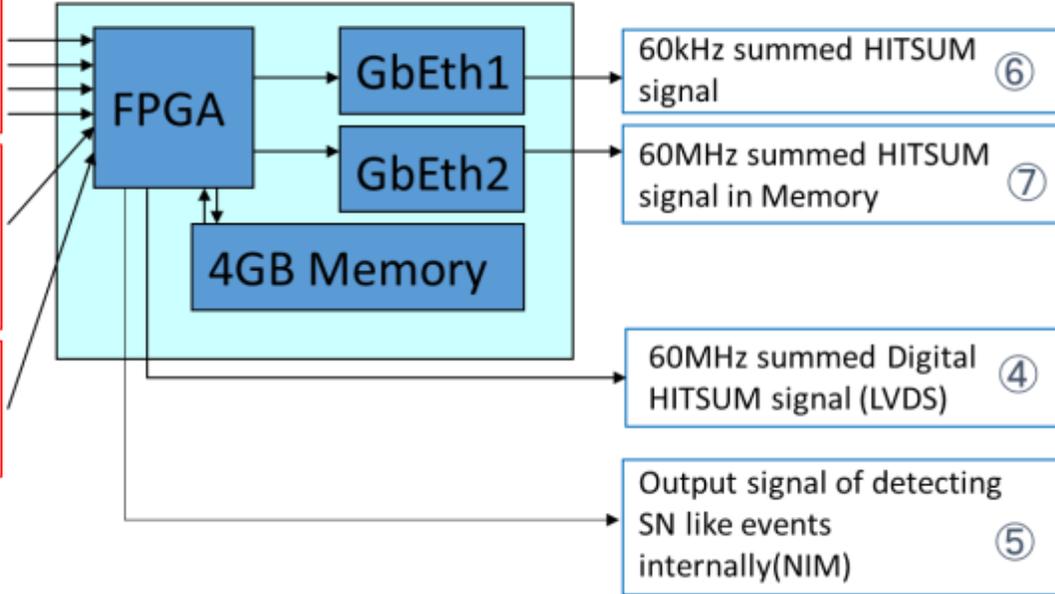
2種類のhitsum

- 60-MHz sum
 - digital hitsumをそのまま足したもの
 - データ量が多すぎる
 - 超新星爆発の周辺～1分を詳細に記録
- 60-kHz sum
 - 60-MHz sumを1024回足したもの
 - データ量を減らせるので、これを常に記録する
 - 超新星ニュートリノ@20 MeV ~ 120hit
 - PMTダークノイズ : 4kHz/PMT → $\sigma \sim 26$ hit/detector
 - イベントとノイズを区別できる

ボード

Input signal

- Digital HITSUM signal from QBEE(LVDS) × 10board (1crate) ①
- MASTER CLOCK (system clock of SK DAQ(60MHz) and event counter)(LVDS) ②
- Input signal of detecting SN like event externally(NIM) ③



Output signal

- 60kHz summed HITSUM signal ⑥
- 60MHz summed HITSUM signal in Memory ⑦
- 60MHz summed Digital HITSUM signal (LVDS) ④
- Output signal of detecting SN like events internally(NIM) ⑤

- FPGA : vertex 5
- memory : DDR2 SO-DIMM
 - 60-MHz sum を保持する
 - 外部トリガ(NIM)が入力されると読み出す
- Ethernet : SiTCP
 - 60-kHz/60-MHzそれぞれに割り当てる



SN trigger for pre-scaling

- ボードごとにburst likeなイベントを検出し、SN triggerを生成
- 条件 : 100hit / module(240PMT) / 17us x 4回連続
- 高エネルギーミューオンによる、PMT-QBEE間での信号の反射がfake triggerの原因になる
- この条件で1MHz以上の超新星イベントが検出可能
- 既存のDAQでは6MHzまでのburstをロスなく記録できる
→bufferがあふれる前にpre-scalingを始められる
- burstの変化と共にpre-scalingの間隔を変える

status

- 2013. 3 モジュールのデザイン、生産
- 2013.12 全ボードの機能試験
- 2014. 4 SKへの影響測定
- 2014. 9 モジュールのインストール
commissioning開始
- 2015. 6 ファームウェアの修正
- 2015. 7 測定開始
 - 60-kHz sumの記録ができるようになった
 - 60-MHz sumは外部トリガが入れば読み出せる
 - pre-scalingはlogicを決めるところから行う
 - 60-MHz sum用のトリガもここで生成する

status

- 2013. 3 モジュールのデザイン、生産
- 2013.12 全ボードの機能試験
- 2014. 4 SKへの影響測定
- 2014. 9 モジュールのインストール
commissioning開始
- 2015. 6 ファームウェアの修正
- 2015. 7 測定開始
 - 60-kHz sumの記録ができるようになった
 - 60-MHz sumは外部トリガが入れば読み出せる
 - pre-scalingはlogicを決めるところから行う
 - 60-MHz sum用のトリガもここで生成する

memory error@プロトタイプ3

- 箱に入れた状態での冷却能力を調べていた
 - 箱には電源と冷却ファンx2がある
 - ボードを2枚入れる
 - memory, Ethernetを動かす
- メモリにエラーが出た
 - テストのたびに正常に動く時間が短くなった
 - 元のテスト環境にしても失敗した
 - 以前は通っていたテストでもエラーが出るようになった
- 温度が高すぎた？ → 70°C以下で問題なかった
- 電源のレギュレータが壊れている？ → Yes



レギュレータの故障

- 5つの電源線(5V, 3.3V, 2.5V, 1.8V, 1.0V)のうち1.8Vの線で約1.1Vしかなかった
- 他のボードでも約1.5Vで、電源の許容量を超えていた
- 1.8Vの線ではより多くの電流が必要になっていた
 - FPGAを変えた
 - Ethernetの通信速度で消費電力が変化
- このときの消費電力—レギュレータの容量 2A
 - FPGA: 1.776A (Xpower Analyzerによる解析) → 1.650A (termination最適化)
 - Ethernet (data sheet): 0.07A@100Mbps → 1Gbps@0.43A
 - 合計で read時: 2.2A, write時: 2.9A

解決

- 問題のレギュレータをより大きな電流で使えるものに変更
 - 回路もそれに合わせた
- 1.8Vの線に十分な電流が流れれば正常に動くことを確かめた
 - 0Ωの抵抗で置き換え、レギュレータをOFFに
 - モニターピンで1.8Vの外部電源に接続
 - memory, Ethernetを同時に動かし、1ヶ月問題がなかった



status

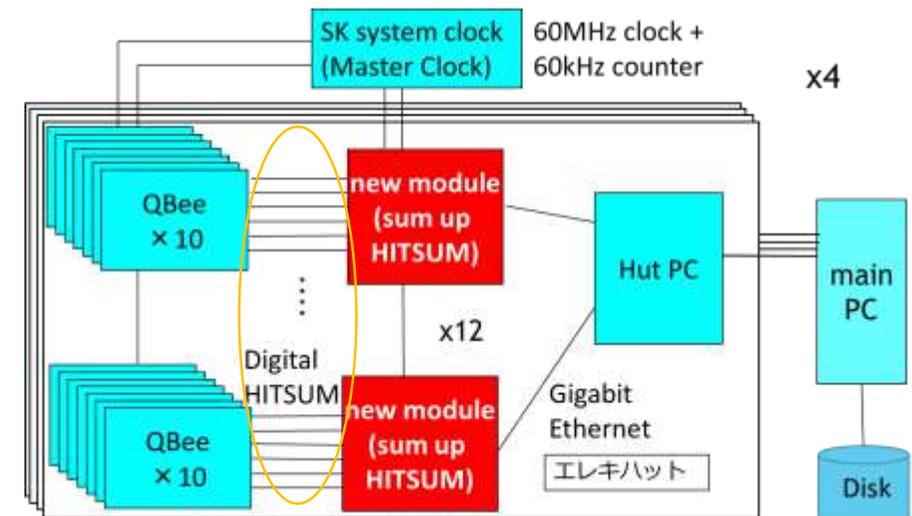
- 2013. 3 モジュールのデザイン、生産
- 2013.12 全ボードの機能試験
- 2014. 4 SKへの影響測定
- 2014. 9 モジュールのインストール
commissioning開始
- 2015. 6 ファームウェアの修正
- 2015. 7 測定開始
 - 60-kHz sumの記録ができるようになった
 - 60-MHz sumは外部トリガが入れば読み出せる
 - pre-scalingはlogicを決めるところから行う
 - 60-MHz sum用のトリガもここで生成する

機能試験

- FPGA, memory, SiTCPの温度を測定した
 - 70°C以下で動作温度を越えなかった
- 全モジュールについて各部品動作を確認した
 - LVDS in/out
 - NIM in/out
 - master/debug clock
 - DIP switch, jumper switch
 - Ethernet
 - memory
- 50枚中2枚にEthernet接続ができないなどの問題があった
 - 必要なのは48枚だが、予備として3枚を追加で製作した
- 試験用のファームウェアを1つにまとめたかったが、Ethernet, memoryが同時に使えず2つに分けた

ダミー信号が動かない@機能試験

- LVDS GPIOモジュールGNV-480をGNV-251に挿してダミー信号を生成、LVDS入力の試験に用いた
- これが動いていないようだった
- GNV-480のGNDがクレートのGNDから+1Vずれていた
- LVDS出力もその分ずれていた
- LVDS信号で同じ電源を利用していたのでなかなかGNDを疑わなかった

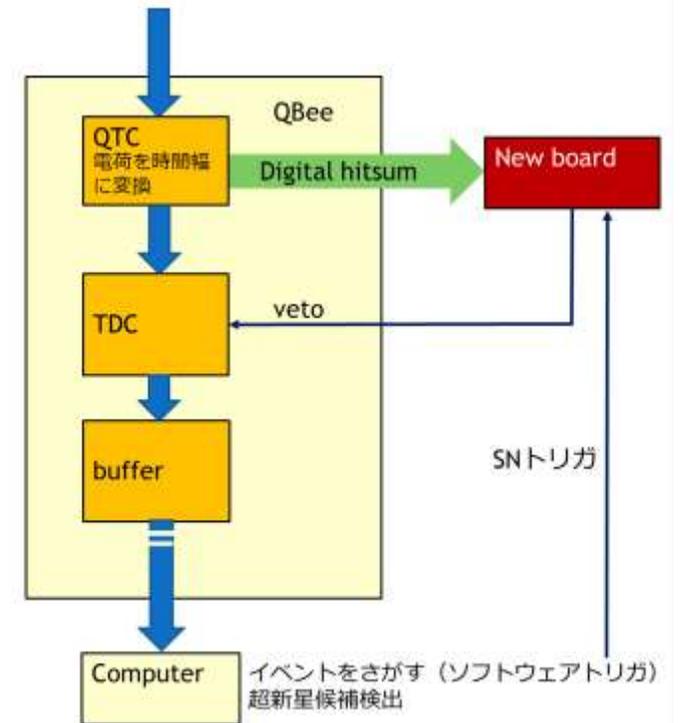


status

- 2013. 3 モジュールのデザイン、生産
- 2013.12 全ボードの機能試験
- 2014. 4 SKへの影響測定
- 2014. 9 モジュールのインストール
commissioning開始
- 2015. 6 ファームウェアの修正
- 2015. 7 測定開始
 - 60-kHz sumの記録ができるようになった
 - 60-MHz sumは外部トリガが入れば読み出せる
 - pre-scalingはlogicを決めるところから行う
 - 60-MHz sum用のトリガもここで生成する

既存DAQシステムへの影響確認

- 新モジュールの接続がQBeelによるデータ収集に影響を与えないか確認した→影響はなかった
- 確認項目
 - 電氣的ノイズ
 - ダークレート
 - ペDESTAL
 - CAL
 - タイミング
- 温度変化も測定した
- このときのファームウェア
 - 全ての入出力が実際と同じように動作する
 - 60kHz-sumの計算とイーサネットでの送信
 - メモリへのデータ書き込み



status

- 2013. 3 モジュールのデザイン、生産
- 2013.12 全ボードの機能試験
- 2014. 4 SKへの影響測定
- 2014. 9 モジュールのインストール
commissioning開始
- 2015. 6 ファームウェアの修正
- 2015. 7 測定開始
 - 60-kHz sumの記録ができるようになった
 - 60-MHz sumは外部トリガが入れば読み出せる
 - pre-scalingはlogicを決めるところから行う
 - 60-MHz sum用のトリガもここで生成する

memoryのデータ量が少ない@ファーム作成

- 全容量4GBを使いたい～1min 60-MHz sum
- 読み出してみたら2GBしかなかった
- memory部の生成にはXilinxのMIGを使った
- 使用したmemoryがライブラリに無かったので自分でパラメータを選んだ
- DDR2, MIG生成部とのやりとりは128bit単位
- 1アドレスで64bit

Block Diagram (w/PLL and w/o TB)

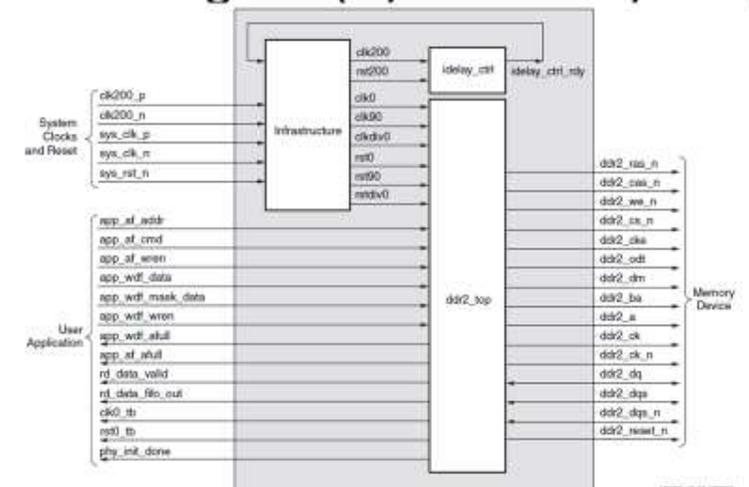


Figure 9-4: Top-Level Block Diagram of the DDR2 SDRAM Design with a PLL but without a Testbench

- 書き込み時

- アドレスを4ずつ増やす

- × 1サイクルで128bit

- 1サイクルで128 x 2 bit: **DDR2!**

- write enableを立てるクロックとその次のクロックで別のデータを書き込む

- 読み出し時

- × アドレスを4ずつ増やす

- アドレスを2ずつ増やす

- 1サイクルで128bit

- 読み書きとも128bit飛ばしだったのでデータの抜けや重複がなく、データ総量を見るまで気が付かなかった

memoryから読み出したデータが戻る

- $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 1000 \rightarrow 0 \rightarrow \dots$ のようなデータを書き込んでいるはず
- 読み出すと $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 100 \rightarrow 0 \rightarrow \dots$
- アドレスのbit幅を打ち間違えていた
- `address[28:0] <= address[18:0] + 29'b2;`

SiTCPでの通信ができない

- 2つのSiTCPを同時に実装しようとするとうまく動かないことがあった
- ファイルをクリアしてコンパイルをやり直すと動くことも
→配線の問題？
- 配線を直接触るのは最後にしたい.....
- constraint を変更したら動いた
 - 主にクロック周り

(急に通信できなくなったと思ったらクロックケーブルが壊れていた)

```
38 #####
39 # I/O STANDARDS
40 #####
41 # Clock Signals
42 // Add on 2014 Aug. 09
43 NET "CLK" CLOCK_DEDICATED_ROUTE = TRUE;
44 NET "TRIG N" CLOCK_DEDICATED_ROUTE = TRUE;
45
46 NET "CLK" LOC = AC8;
47 NET "CLK" IOSTANDARD = LVCMOS33;
48
49
50
51 # Ethernet
52 // ETH1
53 // Change to TRUE on 2014 Aug. 09
54 //NET "GMII_RX_CLK_1" CLOCK_DEDICATED_ROUTE = FALSE;
55 //NET "GMII_TX_CLK_1" CLOCK_DEDICATED_ROUTE = FALSE;
56 NET "GMII_RX_CLK_1" CLOCK_DEDICATED_ROUTE = TRUE;
57 NET "GMII_TX_CLK_1" CLOCK_DEDICATED_ROUTE = TRUE;
58 NET "GMII_GTXCLK_1" CLOCK_DEDICATED_ROUTE = TRUE;
59
60
61 NET "GMII_RX_CLK_1" LOC = D15;
62 NET "GMII_TXD_1[7]" LOC = V26;
63
64 NET "GMII_TXD_1[7]" IOSTANDARD = LVCMOS33;
65
66
67
68 // ETH2
69 // Change to TRUE on 2014 Aug. 09
70 //NET "GMII_RX_CLK_2" CLOCK_DEDICATED_ROUTE = FALSE;
71 //NET "GMII_TX_CLK_2" CLOCK_DEDICATED_ROUTE = FALSE;
72 NET "GMII_RX_CLK_2" CLOCK_DEDICATED_ROUTE = TRUE;
73 NET "GMII_TX_CLK_2" CLOCK_DEDICATED_ROUTE = TRUE;
74 NET "GMII_GTXCLK_2" CLOCK_DEDICATED_ROUTE = TRUE;
75
76 NET "GMII_RX_CLK_2" LOC = E18;
77 NET "GMII_TX_CLK_2" IOSTANDARD = LVCMOS33;
```

```
1247
1248 * PlanAhead Generated IO constraints
1249 NET "GMII_GTXCLK_1" SLEW = FAST;
1250 NET "GMII_GTXCLK_2" SLEW = FAST;
1251 NET "GMII_TXD_1[7]" SLEW = FAST;
1252 NET "GMII_TXD_1[6]" SLEW = FAST;
1253 NET "GMII_TXD_1[5]" SLEW = FAST;
1254 NET "GMII_TXD_1[4]" SLEW = FAST;
1255 NET "GMII_TXD_1[3]" SLEW = FAST;
1256 NET "GMII_TXD_1[2]" SLEW = FAST;
1257 NET "GMII_TXD_1[1]" SLEW = FAST;
1258 NET "GMII_TXD_1[0]" SLEW = FAST;
1259 NET "GMII_TXD_2[7]" SLEW = FAST;
1260 NET "GMII_TXD_2[6]" SLEW = FAST;
1261 NET "GMII_TXD_2[5]" SLEW = FAST;
1262 NET "GMII_TXD_2[4]" SLEW = FAST;
1263 NET "GMII_TXD_2[3]" SLEW = FAST;
1264 NET "GMII_TXD_2[2]" SLEW = FAST;
1265 NET "GMII_TXD_2[1]" SLEW = FAST;
1266 NET "GMII_TXD_2[0]" SLEW = FAST;
1267 #Created by Constraints Editor (xc5v1x50-ff676-1) - 2014/08/09
1268 #Created by Constraints Editor (xc5v1x50-ff676-1) - 2014/08/09
1269 TIMEGRP "GMII_GTXCLK_1" OFFSET = OUT 2 ns AFTER "CLK" REFERENCE_PIN "GMII_GTXCLK_1";
1270 TIMEGRP "GMII_GTXCLK_2" OFFSET = OUT 2 ns AFTER "CLK" REFERENCE_PIN "GMII_GTXCLK_2";
1271 #Created by Constraints Editor (xc5v1x50-ff676-1) - 2014/08/09
1272 NET "GMII_TX_CLK_1" TNM_NET = GMII_TX_CLK_1;
1273 TIMESPEC TS_GMII_TX_CLK_1 = PERIOD "GMII_TX_CLK_1" 8 ns HIGH 50%;
1274 NET "GMII_TX_CLK_2" TNM_NET = GMII_TX_CLK_2;
1275 TIMESPEC TS_GMII_TX_CLK_2 = PERIOD "GMII_TX_CLK_2" 8 ns HIGH 50%;
1276 NET "CLK" TNM_NET = CLK;
```

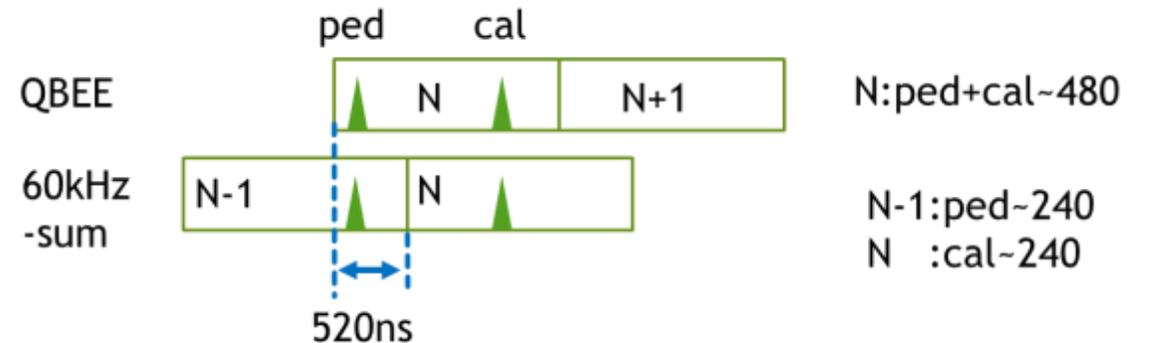
status

- 2013. 3 モジュールのデザイン、生産
- 2013.12 全ボードの機能試験
- 2014. 4 SKへの影響測定
- 2014. 9 モジュールのインストール
commissioning開始
- 2015. 6 ファームウェアの修正
- 2015. 7 測定開始
 - 60-kHz sumの記録ができるようになった
 - 60-MHz sumは外部トリガが入れば読み出せる
 - pre-scalingはlogicを決めるところから行う
 - 60-MHz sum用のトリガもここで生成する

QBEEとの同期

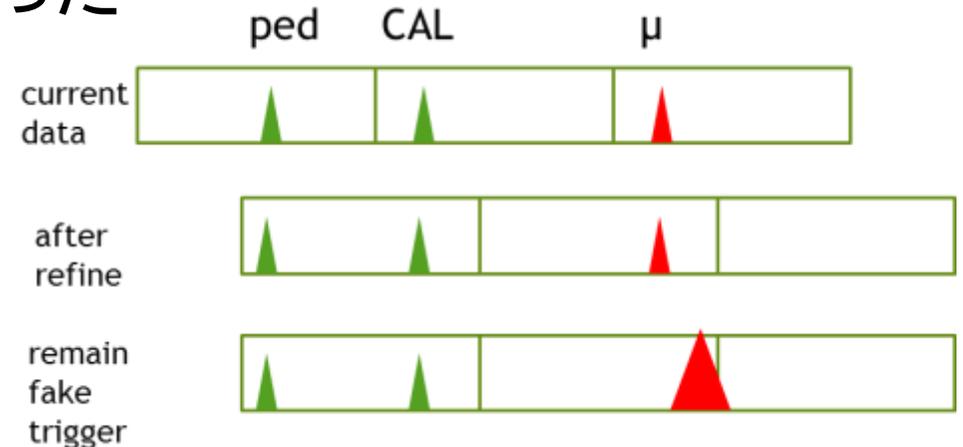
- 2014.9 ver.1
 - 60-kHz sumはQBEEデータと合っていたが、タイミングのズレがあった
 - 1ブロックに入っているはずのQBEE用の較正信号(pedestal, CAL)が2ブロックにまたがっていた
 - fake triggerを生成しやすくなっていた

- 2015.6 ver.2
 - タイミングシフトは17ns以下に抑えられた
 - 60-kHz sumと60-MHz sumの同期を行った



SN trigger生成条件の変更

- 最初の条件 : 100hit / module / 17us x 3回
- ゲートの端にある高エネルギーミュオンを排除するため3回とした
- タイミングのズレによりfake triggerの増加
→ firmware updateで解消
- その後、24時間の測定で6回のfakeがあった
- 条件を4回連続に変更
 - 同じデータでfake 0回
 - burstの検出に問題はないため、この条件に変更した



予定 — pre-scaling logic

- QBEEの処理能力を超えない範囲でできるだけ多くのデータを記録したい
- pre-scaling logicを考える
 - いくつのSN triggerが1になったらpre-scalingを始めるか？
 - 間隔を時間/SN triggerのレートによってどのように変化させるか？
- logic決定のため、レーザーダイオードを用いた試験やシミュレーションを行う