

# CAL July 2010 channel failure

This page describes the failure of the HE channels of a single GCFE. There are 3072 GCFEs in CAL, two for each of the 1536 xtals.

The problem occurred in the run starting at MET = 301753824, which is MJD 55402.52108796, or 2010 Jul 25 at 12:30:22 UTC, or 2010 day 206 at 12:30:22 UTC.

## Symptoms of the failure

The problem occurred in the run starting at MET = 301753824, which is MJD 55402.52108796, or 2010 Jul 25 at 12:30:22 UTC, or 2010 day 206 at 12:30:22 UTC.

On 27 Jul 2010 (Day 208), at 1:55 PM EDT, Anders Borgland wrote:

*Starting with run 301753824 we have two problems:*

*1/*

*We do not see any signal in the high energy diode in tower 4, layer X1, column 4, + side. From Eric Siskind: "Whether the failure is within the diode itself or in the GCFE electronics chain (HE preamp or slow shaper) is currently unknown". You can read the whole thread here:*

<https://www-glast.stanford.edu/protected/mail/datamon/4835.html>

*2/ While problem 1/ affects all (high energy) events in that channel, GCR events (4-range and zero-suppressed events) tickle FSW bug 1156:*

<https://jira.slac.stanford.edu/browse/FSW-1156>

*This means that about 10 events per run will fail in the decompression. Because of the way the Halfpipe works we lose the complete datagram for each of these events. Since a datagram contains about 110 events we are currently losing about 1100 events per run. This corresponds to about 2.5 seconds of data for each 90 minute run.*

*The FSW group have a fix for bug 1156 and will upload a new build asap.*

*Note that currently there is no failure mode in CalRecon so events from this channel is not treated in any special way. NRL is working on this.*

*It should also be noted that the problem was caught immediately by two separate parts of the Data monitoring. The automatic alarms caught both the missing datagrams and the missing signal from the diode. These runs are marked as 'GOOD' by the DQM shifter, but with a comment attached to them. Obviously we will have to live with the missing diode signal from now on.*

*Some of you will not have failed to notice the irony that it's GCR events tickling FSW bug 1156 (hint: SSC-258) 😊*

*anders*

## What has failed?

As soon as we understand exactly what has failed in this GCFE, I'll type something here.

## Consequences of this failure

On 27 Jul 2010 (Day 208), at 2:15 PM EDT, J. Eric Grove wrote:

Additional clarification:

*We do not see any signal in the high energy diode in tower 4, layer X1, column 4, + side. From Eric Siskind: "Whether*  
*[]*

*Note that currently there is no failure mode in CalRecon so events from this channel is not treated in any special way. NRL is working on this.*

What this means is that (until xtal recon is fixed):

1. Any photon that MISSES this *one* crystal is *not* affected. It is correctly and properly reconstructed.
2. Any photon that deposits LESS THAN about 1 GeV in this *one* crystal is *not* affected. It is correctly and properly reconstructed.
3. Any photon that deposits MORE THAN about 1 GeV in this *one* crystal has an incorrect energy and position measurement in this *one* crystal, and therefore has an incorrect reconstructed incident energy and direction. The level of error in reconstructed incident energy and direction is surely energy-dependent, and I don't have an estimate of the magnitude yet.

In the above sentences, "photon" means "any event that is not read out by Trigger Engine 4, i.e. any event that is not read out in 4-range, zero-suppressed mode". I used the word photon to focus the discussion.

The combination of cases (1) and (2) covers the overwhelming majority of photons in the LAT dataset, so most events are perfectly fine, but clearly we need to implement a fix for this particular failure in the code that reconstructs *crystal* energy and position.

Eric

Once CAL xtal recon has been modified with the changes outlined below, the performance of this xtal will be essentially nominal, and the performance of the LAT will be unaffected.

## Changes necessary to recon

As of July 2010, CAL recon code contained no mitigation against failures, despite our having discussed adding such code for the last 10+ years. This failure requires that we do something. The intent here is to fix the reconstruction of events that hit this xtal at the level of xtal recon so that the failure and fix are transparent to downstream energy reconstruction and clustering.

We're adding status bits to xtal recon to indicate that the energy and position information returned by a xtal with compromised readout have been modified. Any reconstruction code that follows xtal recon could make use of those bits to accept/reject that xtal or modify higher level recon algorithms, if the authors and architects of that code wish.

On 4 Aug 2010 (Day 216), at 4:59 PM EDT, J. Eric Grove wrote:

All,

We at NRL had a very productive discussion with Tracy and Leon last week about how to mitigate against the failure in a CAL channel on 25 July. This email fleshes out a proposal for a new calibration DB quantity, how it'll be used, and requests your comments. And it requests that we start the process of adding this new calib quantity to the DB.

Over the years, we've discussed having a status word to indicate known h/w failures, and now we have a good reason to implement it. I wish we'd done it before. Our CAL online s/w used an "exception list" XML file to document failures or out-of-family conditions, and this is similar – but it's not XML.

Each of the 1536 xtals will have a 32-bit status word indicating its hardware and configuration status, and that 1536-element array will be stored as a calibration database quantity. The initial value of the status word for each event comes from the calibration database. In addition, some of the bits in that status word are used by xtal recon to indicate whether the energy and position calculated by xtal recon for a given event are good, need to be corrected, or have been corrected for the appropriate failure mode indicated by the hardware and config status bits.

The bit map for the status word is given at the end of the email. Your comments or proposed modifications to the word are welcome.

The status word array needs to have an epoch of validity, like all other calib quantities. Until this failure, the status word for each xtal end was 0x0. Now we have one xtal with two bits set, the bits that indicate bad plus-face HEX8 and HEX1, and that condition began on 25 July.

In this status word, we need to encode both a priori status information – including hardware status that changes slowly in time, easily tracked as a new calibration DB file, and configuration information that can change from run to run and really should be extracted from MOOT – as well as status information from the xtal recon member functions that indicate whether the energy and position data coming out of xtal recon are good or bad, or whether they were derived by a special algorithm.

For example, for our current failure, we have bad HEX8 and HEX1 data, but we've done nothing to the config registers of this GCFC to prevent HEX8 and HEX1 from appearing. We *could* configure to prevent them from being selected as the best (i.e. only) range, and we're building that config now as a contingency. The code to detect whether a given event is suffering from the failure will depend on how we've configured the GCFC, and I'm convinced for this particular failure it's better if we don't touch the config. I'd prefer to read the config information out of MOOT for the appropriate run, but I'll bet for the moment we should just merge the h/w status and config into a calibration DB file. That's fine. We won't be changing the relevant configuration parameters on a run-by-run basis, so it'll change slowly if at all, consistent with our usage of calib DB quantities.

— Usage —

Tracy and Chul have started coding following the thoughts laid out below.

Recall that the current recon sequence has two CAL passes, Cal1 and Cal2, surrounding an intervening TKR pass that is TkrReconAlg/FirstPass. We will modify the second CAL pass to include a CalXtalRecAlg member function that fixes the response of each CAL xtal that has a failure. Thus:

```
Cal1.Members = { "CalXtalRecAlg/first",  
                 "CalClustersAlg/first",  
                 "CalEventEnergyAlg/RawEnergy" };  
Cal2.Members = { "CalXtalRecAlg/FixXtalResp",  
                 "CalEventEnergyAlg/second" };
```

On the first pass, for the current failure, CalXtalRecAlg::calculate() will see the status word indicating bad HEX8 and HEX1 data, so it'll set status bits indicating that it has generated bad energy and bad longitudinal (X or Y) position information.

Then the moments analysis will see a status word indicating bad energy and longitudinal position information, so it'll ignore that xtal in the moments calculation. If the energy info is bad (status & 0x01 == 1), then the moments analysis should ignore the xtal entirely. If energy info is good – as it will be if the moments analysis were called a second time – then moments can use the two transverse coordinates for this xtal, i.e. those provided by geometry, along with the energy info in the moments analysis.

The new function that fixes the xtal recon data from this xtal after the moments analysis will calculate a good energy and set the position equal to the closest approach of the shower axis (from the moments trajectory). Thus, it'll clear the bits indicating bad energy and bad position, and it'll set the bit indicating that the position it's returning has been derived from external information (i.e. the moments analysis). Any recon code downstream can then decide whether to use that xtal longitudinal position information or not.

I've divided the 32 bits into 8 for recon status (and used 5), 16 for hardware status (and used 8 so far), and 8 for config status (and used 4). Perhaps it'd make more sense to reserve more bits for recon status. Note that I've included two bits for Bill's mods to the longitudinal position calculation to indicate whether he's corrected for direct light in the near diode or corrected for an ambiguous asymmetry. Those can be set in the initial CalXtalRecAlg pass and used as desired later.

Comments?

I'd like to proceed with getting this new calib quantity in the database so we can extract it in recon with CalCalibSvc.

Eric

– bit map for CalXtalRecData status word –

bit	function
-----	----------

status of recon calculation

0	bad energy
1	bad longitudinal position
2	energy has been calculated by failure mitigation algorithm
3	energy has been calculated for corrected longitudinal position
4	position has been provided by external means
5	longitudinal position has been corrected for direct light
6	longitudinal position has been corrected for ambiguous ratio
7	unused

status of h/w

8	bad minus-face LEX8
9	bad minus-face LEX1
10	bad minus-face HEX8
11	bad minus-face HEX1
12	bad plus-face LEX8
13	bad plus-face LEX1
14	bad plus-face HEX8
15	bad plus-face HEX1
16-23	unused

status of config

24	minus-face LE autoranging disabled
25	minus-face HE autoranging disabled
26	plus-face LE autoranging disabled
27	plus-face HE autoranging disabled
28-31	unused

Algorithms for correcting current and not-unlikely future failures in FixXtalResp

On 5 Aug 2010 (Day 217), at 5:24 PM EDT, J. Eric Grove wrote:

Chul,  
Here I've detailed the actions for FixXtalResp in the cases we currently understand. I've added a new CalXtalRecData status word definition [at the bottom of this email] adding Bill's corrected-energy bit and a bit to indicate that we've mitigated the failure.  
Eric

```

case == bad plus-face HEX8 && bad plus-face HEX1 && ! plus-face HE autoranging disabled
/* this is the case for the current failure and current configuration */
/* if one of the HEX ranges is best range, fix it */
/* if plus-face first range == (HEX8 || HEX1),
    calculate E using the opposite face and the externally provided longitudinal position
    clear the bad energy bit
    set the externally provided longitudinal position bit
    set the failure-mitigation energy bit
*/

case == bad plus-face HEX8 && bad plus-face HEX1 && plus-face HE autoranging disabled
/* this is the case for the current failure, but it requires a configuration we have not used */
/* if LEX1 is best range and saturated, fix it */
/* if plus-face first range == LEX1 && plus-face > 4050 (i.e. saturated),
    calculate E using the opposite face and the externally provided longitudinal position
    clear the bad energy bit
    set the externally provided longitudinal position bit
    set the failure-mitigation energy bit
*/

case == bad plus-face LEX8 && bad plus-face LEX1 && bad plus-face HEX8 && bad plus-face HEX1
/* this is the case we may get to soon if this GCFE continues to degrade and the LE ranges fail */
/* in all cases,
    calculate E using the opposite face and the externally provided longitudinal position
    clear the bad energy bit
    set the externally provided longitudinal position bit
    set the failure-mitigation energy bit
*/

```

## Fix for Failed Electronics on Cal Crystal

[Slides](#) for fixing Cal Response for T4L2C4