HELP FILE FOR CALDB FILE USED IN XISPI
H.NAKAJIMA
2005/10/14 VER.1

1 Abstruct

This help file describes the structure and role of each component in CALDB file that is used by xispi.

2 Location and name

ISAS/JAXA:/usr/local/astroe/xis/calibration/caldb/ae_xi?_makepi_YYYYMMDD.fits

? : sensor ID (0 to 3)

YYYYMMDD : Date when this file was released (please use the newset one if there are several files)

3 Policy of the format

For every extension, calibration information at specific time are written in one row, contains `CalTime` in the first column. Whenever the calibration is done for the issues such as gain, CTI, and charge trail, we append a new row with the `CalTime` to corresponding extension and release updated file.

4 Description of each extension

4.1 Charge trail

This extension contains some parameters used for charge trail correction.

We call the charge left in the following pixel "trailing charge" and put it back to the original pixel considering the coordinate of the event.

The amount of trailing charges (hereafter *phtrail*) is derived as follows.

Horizontal Charge Trail:

$$phtrail_h = \mathrm{PH} * ((cti\_ph1500_h * (\mathrm{PH}/1500) * *(alpha_h - 1)) * (\mathrm{RAWX} + offset\_rawx)) \quad (1)$$

Vertical Charge Trail:

$$phtrail_v = \mathrm{PH} * ((cti\_ph1500_v * (\mathrm{PH}/1500) * *(alpha_v - 1)) * (\mathrm{ACTY} + offset\_acty)) \quad (2)$$

where

PH : pulse height of the pixel that the charges trail behind.

cti_ph1500 : Charge trail for PH=1500.

We first calculate the charge trail from the preceding pixels to the center pixel using (1) and (2), then correct them.

$$\begin{aligned} \mathrm{PH}[4] &= \mathrm{PH}[4] + phtrail_h \\ \mathrm{PH}[2] &= \mathrm{PH}[2] + phtrail_v \\ \mathrm{PH}[0] &= \mathrm{PH}[0] - phtrail_h - phtrail_v \end{aligned} \quad (3)$$

Next we calculate the charge trail from the center pixel to the following pixels and correct them.

$$\begin{aligned} \mathrm{PH}[0] &= \mathrm{PH}[0] + phtrail_h + phtrail_v \\ \mathrm{PH}[5] &= \mathrm{PH}[5] - phtrail_h \\ \mathrm{PH}[7] &= \mathrm{PH}[7] - phtrail_v \end{aligned} \quad (4)$$

For the disignation of each pixel, see Figure 1.

In this extnesion, we provide the parameters in equation (1) and (2).

Extension name is "CHARGETRAIL".

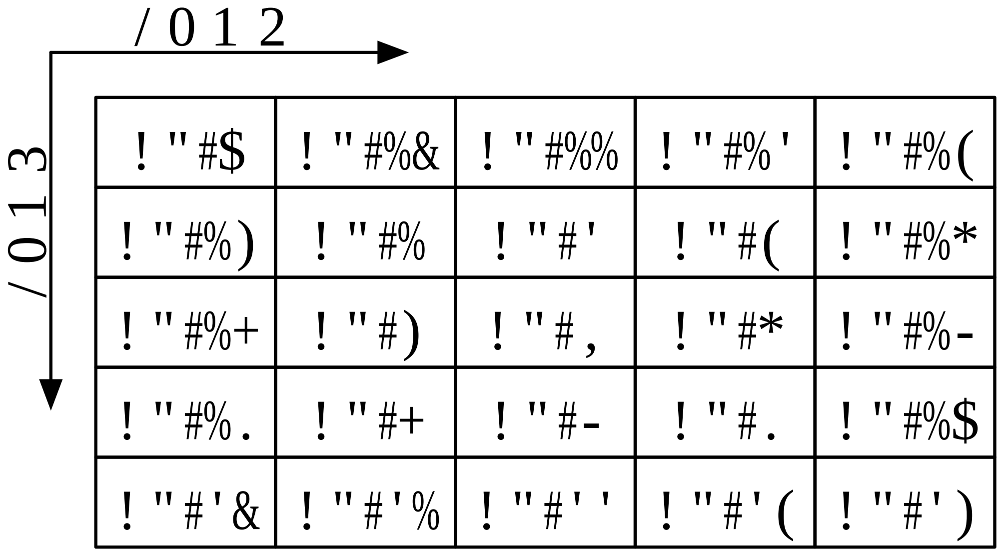

図 1: The disignation of each pixel for an event.

CalTime
Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment
Header name : `Segment`
Meaning : Segment information.

RAWX
Header name : `RAWX`
Meaning : RAWX information.

TrailH
Header name : `TrailH`
Meaning : Horizontal charge trail for PH=1500. cti_ph1500_h in the equation (1).

TrailV
Header name : `TrailV`
Meaning : Vertical charge trail for PH=1500. cti_ph1500_v in the equation (2).

AlphaH
Header name : `AlphaH`
Meaning : $alpha_h$ in the equation (1).

AlphaV
Header name : `AlphaV`
Meaning : $alpha_v$ in the equation (2).

OffsetRAWX
Header name : `OffsetRAWX`
Meaning : $offset\_rawx$ in the equation (1).

OffsetACTY
Header name : `OffsetACTY`
Meaning : $offset\_actx$ in the equation (2).

4.2 Parallel CTI

This extension contains some parameters used for parallel CTI correction.
We derive the amount of parallel CTI as follows.

$$\mathrm{CTI} = cti\_const + cti\_norm * \mathrm{PHAS} * *cti\_pow \tag{5}$$

Then we correct the CTI as follows.

$$\mathrm{PHAS} = \mathrm{PHAS}/(1 - \mathrm{CTI}) * *(\mathrm{ACTY} + 1)) \tag{6}$$

where
PHAS : pulse height of each pixel for one event.
In this extnesion, we provide the parameters in equation (5).
Extension name is "PARALLEL_CTI".

CalTime
Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment
Header name : `Segment`
Meaning : Segment information.

RAWX
Header name : `RAWX`
Meaning : RAWX information.

CTI_CONST
Header name : `CTI_CONST`
Meaning : Constant coponent of CTI. *cti_const* in the equation (5).

CTI_NORM
Header name : `CTI_NORM`
Meaning : Normalization coponent of CTI. *cti_norm* in the equation (5).

CTI_POW
Header name : `CTI_POW`
Meaning : Power-law index coponent of CTI. *cti_pow* in the equation (5).

4.3 Serial CTI

This extension contains some parameters used for serial CTI correction.
How to derive CTI is the same as that of the parallel CTI.
Then we correct the CTI as follows.

$$\mathrm{PHAS} = \mathrm{PHAS}/(1 - \mathrm{CTI}) * *(\mathrm{RAWX} + 1)) \tag{7}$$

Extension name is "SERIAL_CTI".

CalTime
Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment
Header name : `Segment`
Meaning : Segment information.

CTI_CONST
Header name : `CTI_CONST`
Meaning : Constant coponent of CTI. cti_const in the equation (5).

CTI_NORM
Header name : `CTI_NORM`
Meaning : Normalization coponent of CTI. cti_norm in the equation (5).

CTI_POW
Header name : `CTI_POW`
Meaning : Power-law index coponent of CTI. cti_pow in the equation (5).

4.4 Split threshold

This extension contains some parameters used for determining the PHA-dependent split threshold.

We utilize PHA-dependent split threshold for BI chip to determine GRADE. The threshold is defined as follows.

$$\begin{aligned} &\mathrm{Sp.Th.} = spth\_min + (spth\_slope * log10(\mathrm{PHA}) + spth\_offset) \\ &if \quad (\ \mathrm{Sp.Th.} < spth\_min\ ), \quad \mathrm{Sp.TH.} = spth\_min \end{aligned} \tag{8}$$

In the case of FI chips, even if you choose PHA-dependent split threshold in xispi (in other words, even if you choose to use CALDB values), CALDB values are set to make threshold to be 20 independent of PHA.

In this extnesion, we provide the parameters in equation (8).

Extension name is "SPTH_PARAM"

CalTime
Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment
Header name : `Segment`
Meaning : Segment information.

OFFSET
Header name : `OFFSET`
Meaning : $spth\_offset$ in the equation (8).

SLOPE
Header name : `SLOPE`
Meaning : $spth\_slope$ in the equation (8).

MINIMUM
Header name : `MINIMUM`
Meaning : $spth\_min$ in the equation (8).

4.5 Gain variation depending on AE temperature

This extension contains some parameters relate detector gain to AE temperature.

We fit the relation of PHA of Mn Kα as a function of AE temperature as follows.

$$\mathrm{PHA}(\mathrm{MnK}\alpha)(T) = C + aegainnorm * (T - aegainoffset) * *aegainpow \tag{9}$$

C : Constant component

T : AE temperature

On the other hand, PHA of Mn Kα at T_o can be alternatively expressed as follows (see section 4.6).

$$\text{MnK}\alpha \quad energy(eV) = \text{PHA}(\text{MnK}\alpha)(T_o) * *2.0 * quad + \text{PHA}(\text{MnK}\alpha)(T_o) * linr + offset \tag{10}$$

Then we set gainfactor and correct with it against the event taken at T'.

$$\begin{aligned} Gainfactor &= 1 + (\ \text{PHA}(\text{MnK}\alpha)(T') - \text{PHA}(\text{MnK}\alpha)(T_o)\ )\ /\ \text{PHA}(\text{MnK}\alpha)(T_o) & (11) \\ &= 1 + aegainnorm * (\ (\ T' - aegainoffset\ ) * *\ aegainpow - \\ &\quad (\ T_o - aegainoffset\ ) * *\ aegainpow\ )\ /\ \text{PHA}(\text{MnK}\alpha)(T_o) \\ \text{PHA}(\text{MnK}\alpha)\_cor &= \text{PHA}(\text{MnK}\alpha)\ /\ Gainfactor & (12) \end{aligned}$$

In this extnesion, we provide the parameters in equation (9).
Extension name is "GAIN-AETEMP".

CalTime
Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment
Header name : `Segment`
Meaning : Segment information.

NORM
Header name : `NORM`
Meaning : *aegainnorm* in the equation (9).

OFFSET
Header name : `OFFSET`
Meaning : *aegainoffset* in the equation (9).

POW
Header name : `POW`
Meaning : *aegainpow* in the equation (9).

4.6 Gain parameters for Normal mode

This extension contains gain parameters for Normal mode.

We convert PHA to PI after correcting the AE temperature effect as follows. In the current version we utilize separate second polynomials for the energy below and above Si-Kedge.

$$\text{PI}(\text{MnK}\alpha) = \text{PHA}(\text{MnK}\alpha)\_cor * *2.0 * quad + \text{PHA}(\text{MnK}\alpha)\_cor * linr + offset\ /\ XIS\_PI\_UNIT \tag{13}$$

XIS_PI_UNIT : 3.65 eV/ch

In this extnesion, we provide the parameters in equation (13).
Extension name is "GAIN_NORMAL".

CalTime
Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment
Header name : `Segment`
Meaning : Segment information.

QUAD_LOW

Header name : `QUAD_LOW`
Meaning : *quad* in the equation (13) for the energy below Si-Kedge.

LINR_LOW

Header name : `LINR_LOW`
Meaning : *linr* in the equation (13) for the energy below Si-Kedge.

OFFSET_LOW

Header name : `OFFSET_LOW`
Meaning : $offset$ in the equation (13) for the energy below Si-Kedge.

QUAD_HIGH

Header name : `QUAD_HIGH`
Meaning : *quad* in the equation (13) for the energy above Si-Kedge.

LINR_HIGH

Header name : `LINR_HIGH`
Meaning : *linr* in the equation (13) for the energy above Si-Kedge.

OFFSET_HIGH

Header name : `OFFSET_HIGH`
Meaning : $offset$ in the equation (13) for the energy above Si-Kedge.

AETemp

Header name : `AETemp`
Meaning : AE temperature at the time we obtained calibration data.

4.7 Gain parameters for P-sum mode

This extension contains gain parameters for P-sum mode. The contents are basically same as section 4.6 but for P-sum mode.

Extension name is “GAIN_PSUM”.

CalTime

Header name : `Caltime`
Meaning : Time when the calibration was done in AETIME.

Segment

Header name : `Segment`
Meaning : Segment information.

QUAD_LOW

Header name : `QUAD_LOW`
Meaning : *quad* in the equation (13) for the energy below Si-Kedge.

LINR_LOW

Header name : `LINR_LOW`
Meaning : *linr* in the equation (13) for the energy below Si-Kedge.

OFFSET_LOW

Header name : `OFFSET_LOW`
Meaning : $offset$ in the equation (13) for the energy below Si-Kedge.

QUAD_HIGH

Header name : `QUAD_HIGH`
Meaning : $quad$ in the equation (13) for the energy above Si-Kedge.

LINR_HIGH

Header name : `LINR_HIGH`
Meaning : $linr$ in the equation (13) for the energy above Si-Kedge.

OFFSET_HIGH

Header name : `OFFSET_HIGH`
Meaning : $offset$ in the equation (13) for the energy above Si-Kedge.

AETemp

Header name : `AETemp`
Meaning : AE temperature at the time we obtained calibration data.

5 Contact

Questions and comments should be addressed to
nakajima@cr.scphys.kyoto-u.ac.jp