This spec, which is incomplete, describes some of processing steps for continuous-clocking mode data.

The following steps are performed once prior to the processing of the data for each input ACIS event.

1. obsfile:
   (a) Validation:
      i. Existence:
         If

         
         \[ \text{obsfile} \neq \text{none and } \text{obsfile} \neq \text{NONE} \]  \hspace{1cm} (1)

         \[ \text{obsfile} \neq \text{NONE} \]  \hspace{1cm} (2)

         and the obsfile does not exist, then obsfile is changed to “none” and acis_process_events produces a warning message.
ii. Permission:

If

\[
\text{obsfile} \neq \text{none} \quad \text{and} \quad \text{obsfile} \neq \text{NONE},
\]

and the file permissions do not allow the obsfile to be read, then obsfile is changed to “none” and acis\_process\_events produces a warning message.

iii. OBS\_MODE:

If

\[
\text{obsfile} \neq \text{none} \quad \text{and} \quad \text{obsfile} \neq \text{NONE},
\]

then

A. If the obsfile does not include the keyword obs\_mode, then OBS\_MODE is set to “none”.

B. If the obsfile includes the keyword obs\_mode and

\[
\begin{align*}
\text{obs\_mode} &= \text{pointing} \\
\text{obs\_mode} &= \text{POINTING} \\
\text{obs\_mode} &= \text{secondary} \\
\text{obs\_mode} &= \text{SECONDARY},
\end{align*}
\]

then OBS\_MODE is set to the value of obs\_mode. Hereafter this keyword is referred to as OBS\_MODE.

C. If the obsfile includes the keyword obs\_mode and

\[
\begin{align*}
\text{obs\_mode} \neq \text{pointing} \\
\text{obs\_mode} \neq \text{POINTING} \\
\text{obs\_mode} \neq \text{secondary} \\
\text{obs\_mode} \neq \text{SECONDARY},
\end{align*}
\]

then OBS\_MODE is set to “none”.

2. infile:

(a) Existence:

If the infile does not exist, then acis\_process\_events exits with an error message.

(b) Permission:

If the infile exists, but the file permissions do not allow it to be read, then acis\_process\_events exits with an error message.

(c) Validation:

i. OBS\_MODE:

If OBS\_MODE = none, then

A. The OBS\_MODE is read from the HDU \( h_{in} \) keyword of the same name. Hereafter this keyword is referred to as OBS\_MODE.

B. If the HDU \( h_{in} \) does not include the keyword OBS\_MODE, then OBS\_MODE is set to “none” and acis\_process\_events produces a warning message.
C. If the HDU $h_{in}$ includes the keyword OBS\_MODE and

$$\text{obs\_mode} \neq \text{pointing and}$$  

$$\text{obs\_mode} \neq \text{POINTING and}$$  

$$\text{obs\_mode} \neq \text{secondary and}$$  

$$\text{obs\_mode} \neq \text{SECONDARY},$$

then OBS\_MODE is set to “none” and acis\_process\_events produces a warning message.

ii. DATAMODE:
The DATAMODE is read from the HDU $h_{in}$ keyword of the same name. If the HDU $h_{in}$ does not include the keyword DATAMODE or if

$$\text{DATAMODE} \neq \text{CC33\_FAINT and}$$  

$$\text{DATAMODE} \neq \text{CC33\_GRADED and}$$  

$$\text{DATAMODE} \neq \text{FAINT and}$$  

$$\text{DATAMODE} \neq \text{FAINT\_BIAS and}$$  

$$\text{DATAMODE} \neq \text{GRADED and}$$  

$$\text{DATAMODE} \neq \text{VFAINT},$$

then acis\_process\_events exits with an error message. Hereafter, the value of this keyword is referred to as DATAMODE$_{in}$.

iii. CONTENT:
If the infile does not have an HDU $h_{in}$ with the keyword

$$\text{CONTENT} = \text{EVT0 or}$$  

$$\text{CONTENT} = \text{EVT1 or}$$  

$$\text{CONTENT} = \text{TGEVT1 or}$$  

$$\text{CONTENT} = \text{EVT2},$$

then acis\_process\_events exits with an error message. Hereafter, the value of this keyword is referred to as CONTENT$_{in}$.

iv. TIME:
If HDU $h_{in}$ of the infile does not include the column TIME, then acis\_process\_events exits with an error message. Hereafter, this column is referred to as TIME$_{in}$.

v. TIME\_RO:
If

$$\text{DATAMODE}_{in} = \text{CC33\_FAINT or}$$  

$$\text{DATAMODE}_{in} = \text{CC33\_GRADED},$$

if

$$\text{CONTENT}_{in} = \text{EVT1 or}$$  

$$\text{CONTENT}_{in} = \text{TGEVT1 or}$$  

$$\text{CONTENT}_{in} = \text{EVT2},$$

and if HDU $h_{in}$ of the infile does not include the column TIME\_RO, then acis\_process\_events exits with an error message. Hereafter, this column is referred to as TIME\_RO$_{in}$.

vi. EXPNO:
If HDU $h_{in}$ the infile does not include the column EXPNO, then acis\_process\_events exits with an error message. Hereafter, this column is referred to as EXPNO$_{in}$.
vii. CCD_ID:
A. If
\[
\text{CONTENT}_{\text{in}} = \text{EVT0}
\]
and if HDU \( h_{\text{in}} \) of the \text{infile} does not include the keyword \text{CCD_ID}, then \text{acis_process_events} exits with an error message. Hereafter, this keyword is referred to as \text{CCD_ID}_{\text{in}}.
B. If
\[
\begin{align*}
\text{CONTENT}_{\text{in}} &= \text{EVT1 or}\ (35) \\
\text{CONTENT}_{\text{in}} &= \text{TGEVT1 or}\ (36) \\
\text{CONTENT}_{\text{in}} &= \text{EVT2}\ (37)
\end{align*}
\]
and if HDU \( h_{\text{in}} \) of the \text{infile} does not include the column \text{CCD_ID}, then \text{acis_process_events} exits with an error message. Hereafter, this column is referred to as \text{CCD_ID}_{\text{in}}.

viii. CCDX:
A. If
\[
\text{CONTENT}_{\text{in}} = \text{EVT0}\ (38)
\]
and if HDU \( h_{\text{in}} \) of the \text{infile} does not include the column \text{CCDX}, then \text{acis_process_events} exits with an error message. Hereafter, this column is referred to as \text{CCDX}_{\text{in}}.

ix. CHIPX:
A. If
\[
\begin{align*}
\text{CONTENT}_{\text{in}} &= \text{EVT1 or}\ (39) \\
\text{CONTENT}_{\text{in}} &= \text{TGEVT1 or}\ (40) \\
\text{CONTENT}_{\text{in}} &= \text{EVT2}\ (41)
\end{align*}
\]
and if HDU \( h_{\text{in}} \) of the \text{infile} does not include the column \text{CHIPX}, then \text{acis_process_events} exits with an error message. Hereafter, this column is referred to as \text{CHIPX}_{\text{in}}.

tax. CCDY:
A. If
\[
\text{CONTENT}_{\text{in}} = \text{EVT0,}\ (42)
\]
if
\[
\begin{align*}
\text{DATAMODE}_{\text{in}} &= \text{FAINT or}\ (43) \\
\text{DATAMODE}_{\text{in}} &= \text{FAINT_BIAS or}\ (44) \\
\text{DATAMODE}_{\text{in}} &= \text{GRADED or}\ (45) \\
\text{DATAMODE}_{\text{in}} &= \text{VFAINT,}\ (46)
\end{align*}
\]
and if HDU \( h_{\text{in}} \) of the \text{infile} does not include the column \text{CCDY}, then \text{acis_process_events} exits with an error message. Hereafter, this column is referred to as \text{CCDY}_{\text{in}}.

xi. TROW:
A. If
\[
\text{CONTENT}_{\text{in}} = \text{EVT0,}\ (47)
\]
if
\[
\begin{align*}
\text{DATAMODE}_{\text{in}} &= \text{CC33_FAINT or}\ (48) \\
\text{DATAMODE}_{\text{in}} &= \text{CC33_GRADED}\ (49)
\end{align*}
\]
and if HDU \( h_{\text{in}} \) of the \text{infile} does not include the column \text{TROW}, then \text{acis_process_events} exits with an error message. Hereafter, this column is referred to as \text{TROW}_{\text{in}}.
xii. CHIPY:

A. If

\[
\begin{align*}
\text{CONTENT}_{in} & = \text{EVT1 or} \\
\text{CONTENT}_{in} & = \text{TGEVT1 or} \\
\text{CONTENT}_{in} & = \text{EVT2}
\end{align*}
\]

(50)
(51)
(52)

and if HDU \(h_{in}\) of the infile does not include the column CHIPY, then \texttt{acis\_process\_events} exits with an error message. Hereafter, this column is referred to as \text{CHIPY}_{in}.

xiii. TIMEDEL:

If

\[
\begin{align*}
\text{DATAMODE}_{in} & = \text{CC33\_FAINT or} \\
\text{DATAMODE}_{in} & = \text{CC33\_GRADED}
\end{align*}
\]

(53)
(54)

and HDU \(h_{in}\) of the infile does not include the keyword TIMEDEL, then \texttt{acis\_process\_events} exits with an error message. Hereafter this keyword is referred to as \text{TIMEDEL}_{in}.

xiv. RA\_TARG, DEC\_TARG, RA\_NOM, DEC\_NOM, RA\_PNT, DEC\_PNT, CHIPY\_TG, CHIPY\_ZO, and TG\_M:

If

\[
\begin{align*}
\text{OBS\_MODE} & = \text{pointing or} \\
\text{OBS\_MODE} & = \text{POINTING}
\end{align*}
\]

(55)
(56)

and

\[
\begin{align*}
\text{DATAMODE}_{in} & = \text{CC33\_FAINT or} \\
\text{DATAMODE}_{in} & = \text{CC33\_GRADED}
\end{align*}
\]

(57)
(58)

then

A. RA\_TARG, DEC\_TARG, RA\_NOM, DEC\_NOM, RA\_PNT, DEC\_PNT:

If HDU \(h_{in}\) of the infile does not include the keywords RA\_TARG, DEC\_TARG, RA\_NOM, DEC\_NOM, RA\_PNT, and DEC\_PNT, then \texttt{acis\_process\_events} exits with an error message. Hereafter these keywords are referred to as RA\_TARG\_in, DEC\_TARG\_in, RA\_NOM\_in, DEC\_NOM\_in, RA\_PNT\_in, and DEC\_PNT\_in, respectively.

B. CHIPY\_TG, CHIPY\_ZO, and TG\_M:

If

\[
\text{CONTENT}_{in} = \text{TGEVT1}
\]

(59)

and if HDU \(h_{in}\) of the infile does not include the columns CHIPY\_TG, CHIPY\_ZO, and TG\_M, then \texttt{acis\_process\_events} exits with an error message. Hereafter these columns are referred to as CHIPY\_TG\_in, CHIPY\_ZO\_in, and TG\_M\_in, respectively.

3. stop:

(a) Lowercase:

The parameter string is converted to contain only lower case letters.

(b) Validation:

i. Setting:

If

\[
\text{stop} \neq \text{none and}
\]

(60)
then $\text{stop}$ is changed to “none” and $\text{acis\_process\_events}$ produces a warning message.

ii. $\text{OBS\_MODE}$:
   If
   
   $\text{OBS\_MODE} \neq \text{pointing}$ and $\text{OBS\_MODE} \neq \text{POINTING}$

   and

   $\text{stop} \neq \text{none}$ and $\text{stop} \neq \text{chip}$ and $\text{stop} \neq \text{tdet}$,

   then $\text{stop}$ is changed to “none” and $\text{acis\_process\_events}$ produces a warning message.

4. $\text{acaofffile}$:
   (a) Validation for CC mode:
   If
   
   $\text{OBS\_MODE} = \text{pointing}$ or $\text{OBS\_MODE} = \text{POINTING}$

   and

   $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$,

   then

   i. Setting:
   If
   
   $\text{acaofffile} = \text{none}$ or $\text{acaofffile} = \text{NONE}$,

   then $\text{acis\_process\_events}$ exits with an error message.

   ii. Existence:
   If the $\text{acaofffile}$ does not exist, then $\text{acis\_process\_events}$ exits with an error message.

   iii. Permission:
   If the $\text{acaofffile}$ exists, but the file permissions do not allow it to be read, then $\text{acis\_process\_events}$ exits with an error message.

   iv. CONTENT:
   If the $\text{acaofffile}$ does not have an HDU $h_{\text{acaoff}}$ with the keyword
   
   \[
   \text{CONTENT} = \text{ASPSOL},
   \]

   then $\text{acis\_process\_events}$ exits with an error message.
v. Keyword:
If HDU hacaoff of the acaofffile does not include the keyword TSTART, then acis_process_events exits with an error message.

vi. Columns:
If HDU hacaoff of the acaofffile does not include the columns TIME, RA, DEC, and ROLL then acis_process_events exits with an error message. Hereafter, these columns are referred to as TIMEacaoff, Racaoff, Dacaoff, and ROLLaoff.

vii. Sequential:
If more than one valid acaofffile is specified, but the values TSTART are not in increasing order, then acis_process_events exits with an error message.

5. doevtgrade:
   (a) Lowercase:
The parameter string is converted to contain only lower case letters.
   (b) Validation:
   If
   \[
   \text{doevtgrade} \neq \text{yes} \quad \text{and} \\
   \text{doevtgrade} \neq \text{no},
   \]
   then acis_process_events exits with an error message.

6. apply_cti:
   (a) Lowercase:
The parameter string is converted to contain only lower case letters.
   (b) Validation:
   i. Setting:
   If
   \[
   \text{apply}_\text{cti} \neq \text{yes} \quad \text{and} \\
   \text{apply}_\text{cti} \neq \text{no},
   \]
   then acis_process_events exits with an error message.
   ii. PHAS:
   If
   \[
   \text{apply}_\text{cti} = \text{yes}
   \]
   and if the infile does not include the column PHAS, then apply_cti is changed to “no” and acis_process_events produces a warning message.
   iii. doevtgrade:
   If
   \[
   \text{apply}_\text{cti} = \text{yes} \quad \text{and} \\
   \text{doevtgrade} = \text{no},
   \]
   then apply_cti is changed to “no” and acis_process_events produces a warning message.

7. alignmentfile:
(a) Validation for CC mode:

If

\[
\begin{align*}
\text{OBS\_MODE} &= \text{pointing or} \quad (85) \\
\text{OBS\_MODE} &= \text{POINTING} \quad (86)
\end{align*}
\]

and

\[
\begin{align*}
\text{DATAMODE\_in} &= \text{CC33\_FAINT or} \quad (87) \\
\text{DATAMODE\_in} &= \text{CC33\_GRADED}, \quad (88)
\end{align*}
\]

then

i. Setting:

If

\[
\begin{align*}
\text{alignmentfile} &= \text{none or} \quad (89) \\
\text{alignmentfile} &= \text{NONE}, \quad (90)
\end{align*}
\]

then \texttt{acis\_process\_events} exits with an error message.

ii. Existence:

If the \texttt{alignmentfile} does not exist, then \texttt{acis\_process\_events} exits with an error message.

iii. Permission:

If the \texttt{alignmentfile} exists, but the file permissions do not allow it to be read, then \texttt{acis\_process\_events} exits with an error message.

iv. CONTENT:

If the \texttt{alignmentfile} does not have an HDU \texttt{h\_alignment} with the keyword

\[
\text{CONTENT} = \text{ASPSOL}, \quad (91)
\]

then \texttt{acis\_process\_events} exits with an error message.

v. Keyword:

If HDU \texttt{h\_alignment} of the \texttt{alignmentfile} does not include the keyword \texttt{TSTART}, then \texttt{acis\_process\_events} exits with an error message.

vi. Columns:

If HDU \texttt{h\_alignment} of the \texttt{alignmentfile} does not include the columns \texttt{DY}, \texttt{DZ}, and \texttt{DTHETA} then \texttt{acis\_process\_events} exits with an error message.

vii. Sequential:

If more than one valid \texttt{alignmentfile} is specified, but the the values \texttt{TSTART} are not in increasing order, then \texttt{acis\_process\_events} exits with an error message.

8. badpixfile:

(a) Validation:

i. Existence:

If

\[
\begin{align*}
\text{badpixfile} &\neq \text{none and} \quad (92) \\
\text{badpixfile} &\neq \text{NONE} \quad (93)
\end{align*}
\]

and the \texttt{badpixfile} does not exist, then \texttt{badpixfile} is changed to “none” and \texttt{acis\_process\_events} produces a warning message.
ii. Permission:
   If
   \begin{align}
   \text{badpixfile} & \neq \text{none} \quad \text{(94)} \\
   \text{badpixfile} & \neq \text{NONE} \quad \text{(95)}
   \end{align}
   and the file permissions do not allow it to be read, then \text{badpixfile} is changed to “none” and \text{acis_process_events} produces a warning message.

iii. CONTENT:
   If
   \begin{align}
   \text{badpixfile} & \neq \text{none} \quad \text{(96)} \\
   \text{badpixfile} & \neq \text{NONE} \quad \text{(97)}
   \end{align}
   and the \text{badpixfile} does not have one or more HDUs \text{h}\text{badpix} with the keyword
   \begin{align}
   \text{CONTENT} & = \text{BADPIX} \quad \text{(98)} \\
   \text{CONTENT} & = \text{CDB\_ACIS\_BADPIX}, \quad \text{(99)}
   \end{align}
   then \text{badpixfile} is changed to “none” and \text{acis_process_events} produces a warning message.

iv. Keyword:
   If
   \begin{align}
   \text{badpixfile} & \neq \text{none} \quad \text{(100)} \\
   \text{badpixfile} & \neq \text{NONE} \quad \text{(101)}
   \end{align}
   and the HDU(s) \text{h}\text{badpix} of the \text{badpixfile} do not include the keyword \text{CCD\_ID}, then \text{badpixfile} is changed to “none” and \text{acis_process_events} produces a warning message. Hereafter this keyword is referred to as \text{CCD\_ID}\text{badpix}.

v. Columns:
   If
   \begin{align}
   \text{badpixfile} & \neq \text{none} \quad \text{(102)} \\
   \text{badpixfile} & \neq \text{NONE} \quad \text{(103)}
   \end{align}
   and the HDU(s) \text{h}\text{badpix} of the \text{badpixfile} do not include the columns \text{CHIPX}, \text{CHIPY}, \text{TIME}, \text{TIME\_STOP}, and \text{STATUS}, then \text{badpixfile} is changed to “none” and \text{acis_process_events} produces a warning message. Hereafter these columns are referred to as \text{CHIPX}\text{badpix}, \text{CHIPY}\text{badpix}, \text{TIME}\text{badpix}, \text{TIME\_STOP}\text{badpix}, and \text{STATUS}\text{badpix}, respectively.

9. \text{ctifile}:
   (a) Validation:
   If
   \begin{align}
   \text{ctifile} & \neq \text{caldb} \quad \text{(104)} \\
   \text{ctifile} & \neq \text{CALDB}, \quad \text{(105)}
   \end{align}
   then
   i. Existence:
   If the \text{ctifile} does not exist, then \text{apply\_cti} is changed to “no” and \text{acis_process_events} produces a warning message.
ii. Permission:
If the ctifile exists, but the file permissions do not allow it to be read, then apply_cti is changed to “no” and acis_process_events produces a warning message.

iii. CONTENT:
If the ctifile does not have one or more HDUs hcti with the keyword

$$\text{CONTENT} = \text{CDB\_ACIS\_CTI},$$  \hspace{1cm} (106)

then apply_cti is changed to “no” and acis_process_events produces a warning message.

iv. Columns:
If the first such HDU of the ctifile does not include the columns CCD\_ID, CHIPX\_LO, CHIPX\_HI, CHIPY\_LO, CHIPY\_HI, PHA, VOLUME\_X, VOLUME\_Y, FRCTRLX, FRCTRLY, TCTIX, and TCTIY, then apply_cti is changed to “no” and acis_process_events produces a warning message.

10. clobber:
(a) Lowercase:
The parameter string is converted to contain only lower case letters.

(b) Validation:
   i. Setting:
      If

$$\text{clobber} \neq \text{yes and}$$  \hspace{1cm} (107)
$$\text{clobber} \neq \text{no},$$  \hspace{1cm} (108)

then clobber is changed to “no” and acis_process_events produces a warning message.

ii. Permission:
If

$$\text{clobber} = \text{yes}$$  \hspace{1cm} (109)

and the outfile exists, but the file permissions of the outfile do not allow it to be overwritten, then acis_process_events exits with an error message.

iii. Don’t overwrite:
If

$$\text{clobber} = \text{no}$$  \hspace{1cm} (110)

and the outfile exists, then acis_process_events exits with an error message.

11. pix_adj:
(a) Lowercase:
The parameter string is converted to contain only lower case letters.

(b) Validation:
   i. Setting:
      If

$$\text{pix\_adj} \neq \text{centroid and}$$  \hspace{1cm} (111)
$$\text{pix\_adj} \neq \text{edser and}$$  \hspace{1cm} (112)
$$\text{pix\_adj} \neq \text{none and}$$  \hspace{1cm} (113)
$$\text{pix\_adj} \neq \text{randomize},$$  \hspace{1cm} (114)

then pix_adj is changed to “none” and acis_process_events produces a warning message.
ii. **OBS.MODE:**
   If
   
   \[
   \text{OBS.MODE} \neq \text{pointing and} \quad (115) \\
   \text{OBS.MODE} \neq \text{POINTING} \quad (116)
   \]
   
   and
   
   \[
   \text{pix.adj} \neq \text{none}, \quad (117)
   \]
   
   then pix.adj is changed to “none” and acis_process_events produces a warning message.

iii. **stop:**
   If
   
   \[
   \text{pix.adj} = \text{centroid or} \quad (118) \\
   \text{pix.adj} = \text{edser or} \quad (119) \\
   \text{pix.adj} = \text{randomize} \quad (120)
   \]
   
   and if
   
   \[
   \text{stop} \neq \text{sky}, \quad (121)
   \]
   
   then pix.adj is changed to “none” and acis_process_events produces a warning message.

iv. **PHAS:**
   If
   
   \[
   \text{pix.adj} = \text{centroid} \quad (122)
   \]
   
   and if the infile does not include the column PHAS, then pix.adj is changed to “none” and acis_process_events produces a warning message.

v. **FLTGRADE:**
   If
   
   \[
   \text{pix.adj} = \text{edser} \quad (123)
   \]
   
   and if the infile does not include the column FLTGRADE, then pix.adj is changed to “none” and acis_process_events produces a warning message.

12. **subpixfile:**
   (a) If
   
   \[
   \text{pix.adj} = \text{edser}, \quad (124)
   \]
   
   then
   
   i. **Existence:**
      If the subpixfile does not exist, then pix.adj is changed to “none” and acis_process_events produces a warning message.
   
   ii. **Permission:**
      If the subpixfile exists, but the file permissions do not allow it to be read, then pix.adj is changed to “none” and acis_process_events produces a warning message.
   
   iii. **Validation:**

   11
A. **CONTENT:**

If the `subpixfile` does not have one or more HDUs `h_{subpix}` with the keyword

\[
\text{CONTENT} = \text{AXAF\_SUBPIX},
\]

then `pix\_adj` is changed to “none” and `acis\_process\_events` produces a warning message.

B. **Keyword:**

If the HDUs `h_{subpix}` of the `subpixfile` do not include the keyword `CCD\_ID`, then `pix\_adj` is changed to “none” and `acis\_process\_events` produces a warning message.

C. **Columns:**

If the HDUs `h_{subpix}` of the `subpixfile` do not include binary tables with the columns `FLTGRADE`, `NPOINTS`, `ENERGY`, `CHIPX\_OFFSET`, and `CHIPY\_OFFSET`, then `pix\_adj` is changed to “none” and `acis\_process\_events` produces a warning message. Hereafter these columns are referred to as `FLTGRADE_{subpix}`, `NPOINTS_{subpix}`, `ENERGY_{subpix}`, `CHIPX\_OFFSET_{subpix}`, and `CHIPY\_OFFSET_{subpix}`, respectively.

### 1.5.2 Initializations

1. **Focal-point CCD:**
   
   If
   
   \[
   \text{OBS\_MODE} = \text{pointing or POINTING}
   \]
   
   then the values of `RA\_PNT_{in}` and `DEC\_PNT_{in}` are used to determine the `CCD\_ID` associated with the focal point. Hereafter this value is referred to as `CCD\_ID_{focus}`.

2. **Zeroth-order coordinates:**
   
   If
   
   \[
   \text{OBS\_MODE} = \text{pointing or POINTING}
   \]
   
   then the `CHIPY\_ZO_{in}` coordinates are processed to obtain the minimum, median, and maximum values:

   \[
   \begin{align*}
   \text{CHIPY\_ZO}_{\text{min}} &= \text{minimum} \left( \text{CHIPY\_ZO}_{\text{in}} \right), \\
   \text{CHIPY\_ZO}_{\text{med}} &= \text{median} \left( \text{CHIPY\_ZO}_{\text{in}} \right), \quad \text{and} \\
   \text{CHIPY\_ZO}_{\text{max}} &= \text{maximum} \left( \text{CHIPY\_ZO}_{\text{in}} \right).
   \end{align*}
   \]

   *The focal point is the location associated with the optical axis in the absence of dither. This location should not be confused with the aim point, which is the location illuminated by an undithered point source provided that the source is not offset from the target location.*
Only events in the good-time intervals are included in the computation of the values of \( \text{CHIPY}_{Z0_{\text{min}}}, \text{CHIPY}_{Z0_{\text{med}}}, \) and \( \text{CHIPY}_{Z0_{\text{max}}} \).

3. `acaofffile`:

If
\[
\begin{align*}
\text{OBS\_MODE} & = \text{pointing or} \quad (138) \\
\text{OBS\_MODE} & = \text{POINTING} \quad (139)
\end{align*}
\]

and
\[
\begin{align*}
\text{DATAMODE} & = \text{CC33\_FAINT or} \quad (140) \\
\text{DATAMODE} & = \text{CC33\_GRADED}, \quad (141)
\end{align*}
\]

then

(a) \( \text{TIME}_{\text{min}}, \text{TIME}_{\text{max}}, \text{RA}_c, \) and \( \text{DEC}_c \):

The `acaofffile` data are processed to determine the earliest and latest times for which there is aspect information and to determine the right ascension and declination coordinates near the center of the dither pattern:
\[
\begin{align*}
\text{TIME}_{\text{min}} & = \min (\text{TIME}_{\text{acaoff}}) , \quad (142) \\
\text{TIME}_{\text{max}} & = \max (\text{TIME}_{\text{acaoff}}) , \quad (143) \\
\text{RA}_c & = \text{median} (\text{RA}_{\text{acaoff}}) , \quad (144) \\
\text{DEC}_c & = \text{median} (\text{DEC}_{\text{acaoff}}) . \quad (145)
\end{align*}
\]

(b) \( \text{TIME}_c \):

The `acaofffile` data are processed to determine the time \( \text{TIME}_c \) at which the quantity
\[
\cos (\text{DEC}_{\text{acaoff}}) \cos (\text{DEC}_c) \cos (\text{RA}_{\text{acaoff}} - \text{RA}_c) + \sin (\text{DEC}_{\text{acaoff}}) \sin (\text{DEC}_c) \quad (146)
\]

is maximized (i.e. the time at which the telescope is pointed the closest to \( (\text{RA}_c, \text{DEC}_c) \)).

(c) \( \text{RA\_ADJ}_I, \text{DEC\_ADJ}_I, \text{RA\_ADJ}_S, \text{DEC\_ADJ}_S \):

The effective values of \( \text{RA} \) and \( \text{DEC} \) are computed for the ACIS-I and ACIS-S arrays. These coordinates are used to determine the values of \( \text{TIME} \) and \( \text{CHIPY}_{\text{ADJ}} \).

i. ACIS-I aim point:

For the ACIS-I array, the values of \( \text{RA\_ADJ}_I \) and \( \text{DEC\_ADJ}_I \) are initialized assuming that the source is at the ACIS-I aim point [i.e. that \( (\text{TIME}, \text{CCD\_ID}, \text{CHIPX}, \text{CHIPY}) = (\text{TIME}_c, 3, 965, 963) \)^†].

ii. ACIS-S aim point:

For the ACIS-S array, the values of \( \text{RA\_ADJ}_S \) and \( \text{DEC\_ADJ}_S \) are initialized assuming that the source is at the ACIS-S aim point [i.e. that \( (\text{TIME}, \text{CCD\_ID}, \text{CHIPX}, \text{CHIPY}) = (\text{TIME}_c, 7, 227, 509) \)^‡].

iii. Target location:

For the CCD at the focal point (i.e. \( \text{CCD\_ID}_{\text{focus}} \)), the values of \( \text{CHIPY} \) are computed for each row of the `acaofffile`, assuming that the source is at the location specified by \( \text{RA}_{\text{TARG}_{\text{in}}} \) and \( \text{DEC}_{\text{TARG}_{\text{in}}} \). These values of \( \text{CHIPY} \) are referred to as \( \text{CHIPY}_{\text{TARG}} \). If
\[
\begin{align*}
\text{median} (\text{CHIPY}_{\text{TARG}}) & \geq 16.5 \quad \text{(147)} \\
\text{median} (\text{CHIPY}_{\text{TARG}}) & < 1008.5 \quad \text{(148)}
\end{align*}
\]

then

---

† As described in the Proposers' Observatory Guide, the location of the aim point on the ACIS-I array has drifted with time. The location used here is within a few dozen pixels of the actual aim point, provided the default \( \text{SIM\_Y} \) and \( \text{SIM\_Z} \) offsets are used.

‡ Again, the location used here is within a few dozen pixels of the actual aim point, provided the default \( \text{SIM\_Y} \) and \( \text{SIM\_Z} \) offsets are used.
A. ACIS-I:
If
\[
\begin{align*}
\text{CCD ID}_\text{focus} & \geq 0 \text{ and } \text{CCD ID}_\text{focus} \leq 3, \\
\end{align*}
\]
then
\[
\begin{align*}
\text{RA}_{\text{ADJ}1} &= \text{RA}_{\text{TARG}in} \text{ and } \\
\text{DEC}_{\text{ADJ}1} &= \text{DEC}_{\text{TARG}in}.
\end{align*}
\]

B. ACIS-S:
If
\[
\begin{align*}
\text{CCD ID}_\text{focus} & \geq 4 \text{ and } \text{CCD ID}_\text{focus} \leq 9, \\
\end{align*}
\]
then
\[
\begin{align*}
\text{RA}_{\text{ADJS}} &= \text{RA}_{\text{TARG}in} \text{ and } \\
\text{DEC}_{\text{ADJS}} &= \text{DEC}_{\text{TARG}in}.
\end{align*}
\]

1.5.3 Loop over events

The following steps are performed, in sequence, for each event.

1. STATUS:
   (a) Exists:
   If HDU $h_{\text{in}}$ of the infile includes a 32-bit column named STATUS, then
   i. The values of the bits for an event are read from the infile.
   ii. The value of STATUS[$k$] is set to zero for bits $k = 1\text{–}5, 14, 16\text{–}19, \text{ and } 23 \text{ (of } 0\text{–}31\text{), bits that can be set by acis_process_events}.$
   iii. If $\text{doevtgrade} = \text{yes}$,
   then the value of STATUS[20], the other bit that can be set by acis_process_events, is set to zero.
   (b) Does not exist:
   If HDU $h_{\text{in}}$ does not include a 32-bit column named STATUS, then
   i. A set of 32 bits are allocated for the event.
   ii. The values of the 32 bits are initialized to zero.

2. EXPNO:
   (a) Read:
   The value of EXPNO for an event is given by EXPNO$_{\text{in}}$.
   (b) Validation:
   If
   \[
   \begin{align*}
   \text{EXPNO} & < 0 \text{ or } \\
   \text{EXPNO} & \geq 10^8,
   \end{align*}
   \]
   then acis_process_events produces a warning upon completion with a count of the total number
   of events for which one or the other of these conditions is true. These values should not occur.
3. **CCD_ID**:  
   (a) Read:  
   The value of CCD_ID for an event is given by CCD_ID_{in}. 
   (b) Validation:  
   If  
   \[
   \begin{align*}
   \text{CCD}_\text{ID} & < 0 \text{ or} \\
   \text{CCD}_\text{ID} & > 9,
   \end{align*}
   \]  
   then acis_process_events exits with an error message because CCD_ID-dependent computations could fail if the value of CCD_ID is unphysical.  

4. **CHIPX**:  
   (a) Read:  
   i. Level 0:  
   If  
   \[
   \text{CONTENT}_{\text{in}} = \text{EVT0},
   \]  
   then the value of CHIPX for an event is given by  
   \[
   \text{CHIPX} = \text{CCDX}_{\text{in}} + 1.
   \]  
   ii. Level 1, 1.5, or 2:  
   If  
   \[
   \begin{align*}
   \text{CONTENT}_{\text{in}} & = \text{EVT1}, \text{ or} \\
   \text{CONTENT}_{\text{in}} & = \text{TGEVT1}, \text{ or} \\
   \text{CONTENT}_{\text{in}} & = \text{EVT2},
   \end{align*}
   \]  
   then the value of CHIPX for an event is given by CHIPX_{in}.  
   (b) Validation:  
   i. Unphysical:  
   If  
   \[
   \begin{align*}
   \text{CHIPX} & < 1 \text{ or} \\
   \text{CHIPX} & > 1024,
   \end{align*}
   \]  
   then acis_process_events exits with an error message because CHIPX-dependent computations could fail if the value of CHIPX is unphysical.  
   ii. Unexpected:  
   If  
   \[
   \begin{align*}
   \text{CHIPX} & = 1 \text{ or} \\
   \text{CHIPX} & = 1024,
   \end{align*}
   \]  
   then acis_process_events produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. Although these values are not unphysical, they should not occur.  

5. **NODE_ID**:  

15
(a) Calculate:
The NODE ID of an event is given by
\[
\text{NODE ID} = \text{int} \left( \frac{\text{CHIPX} - 1}{256} \right),
\]  
(171)

where “int” means the integer portion of (i.e. truncate or round down) the quantity in parentheses.

6. CHIPY:
(a) Read:
i. Level 0:
If
\[
\text{CONTENT}_{\text{in}} = \text{EVT0},
\]  
(172)
then
A. TE mode:
If
\[
\text{DATAMODE}_{\text{in}} = \text{FAINT or FAINT_BIAS or GRADED or VFAINT},
\]  
(173)
(174)
(175)
(176)
then the value of CHIPY for an event is given by
\[
\text{CHIPY} = \text{CCDY}_{\text{in}} + 1.
\]  
(177)
B. CC mode:
If
\[
\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or CC33_GRADED},
\]  
(178)
(179)
then the value of CHIPY for an event is given by
\[
\text{CHIPY} = \text{TROW}_{\text{in}} + 1.
\]  
(180)

ii. Level 1, 1.5, or 2:
If
\[
\text{CONTENT}_{\text{in}} = \text{EVT1 or TGEVT1 or EVT2},
\]  
(181)
(182)
(183)
then the value of CHIPY for an event is given by \(\text{CHIPY}_{\text{in}}\).

(b) Validation:
i. Unphysical:
A. TE mode:
If
\[
\text{DATAMODE}_{\text{in}} = \text{FAINT or FAINT_BIAS or GRADED or VFAINT},
\]  
(184)
(185)
(186)
(187)
and if

\[
\begin{align*}
\text{CHIPY} &< 1 \text{ or } & (188) \\
\text{CHIPY} &> 1024, & (189)
\end{align*}
\]

then \texttt{acis\_process\_events} exits with an error message because CHIPY-dependent computations could fail if the value of CHIPY is unphysical.

B. CC mode:

If

\[
\text{DATAMODE}_{in} = \text{CC33\_FAINT} \text{ or } (190) \\
\text{DATAMODE}_{in} = \text{CC33\_GRADED} \quad (191)
\]

and if

\[
\begin{align*}
\text{CHIPY} &< 1 \text{ or } & (192) \\
\text{CHIPY} &> 512, & (193)
\end{align*}
\]

then \texttt{acis\_process\_events} exits with an error message because the CHIPY value is out of range and CHIPY-dependent computations could fail if the value of CHIPY is unphysical (especially if it is less than 1).

ii. Unexpected:

A. FAINT, FAINT\_BIAS, or GRADED:

If

\[
\begin{align*}
\text{DATAMODE}_{in} & = \text{FAINT} \text{ or } & (194) \\
\text{DATAMODE}_{in} & = \text{FAINT\_BIAS} \text{ or } & (195) \\
\text{DATAMODE}_{in} & = \text{GRADED} \quad (196)
\end{align*}
\]

and if

\[
\begin{align*}
\text{CHIPY} & = 1 \text{ or } & (197) \\
\text{CHIPY} & = 1024, & (198)
\end{align*}
\]

then \texttt{acis\_process\_events} produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. Although these values are not unphysical, they should not occur.

B. VFAINT:

If

\[
\text{DATAMODE}_{in} = \text{VFAINT} \quad (199)
\]

and if

\[
\begin{align*}
\text{CHIPY} & = 1 \text{ or } & (200) \\
\text{CHIPY} & = 2 \text{ or } & (201) \\
\text{CHIPY} & = 1023 \text{ or } & (202) \\
\text{CHIPY} & = 1024, & (203)
\end{align*}
\]

then \texttt{acis\_process\_events} produces a warning upon completion with a count of the total number of events for which one or another of these conditions is true. Although these values are not unphysical, they should not occur.
C. CC33\_FAINT or CC33\_GRADED:

If

\[
\text{DATAMODE}_n = \text{CC33\_FAINT or CC33\_GRADED} \tag{204}
\]

and if

\[
\begin{align*}
\text{CHIPY} &= 1 \text{ or } \tag{206} \\
\text{CHIPY} &= 512, \tag{207}
\end{align*}
\]

then \textit{acis\_process\_events} produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. Although these values are not unphysical, they should not occur.

7. \text{TG}_M:

(a) CC mode with gratings:

If

\[
\begin{align*}
\text{OBS\_MODE} &= \text{pointing or } \tag{208} \\
\text{OBS\_MODE} &= \text{POINTING} \tag{209}
\end{align*}
\]

and

\[
\begin{align*}
\text{DATAMODE}_n &= \text{CC33\_FAINT or } \tag{210} \\
\text{DATAMODE}_n &= \text{CC33\_GRADED} \tag{211}
\end{align*}
\]

and

\[
\text{CONTENT}_n = \text{TGEVT1}, \tag{212}
\]

then

i. Read:

The value of \text{TG}_M for an event is given by \text{TG}_M_n.

ii. Validation:

A. If

\[
\text{TG}_M < -99, \tag{213}
\]

then

\[
\text{TG}_M = -99 \tag{214}
\]

and \textit{acis\_process\_events} produces a warning upon completion with a count of the total number of events for which this condition is true. These values should not occur.

B. If

\[
\text{TG}_M > 99, \tag{215}
\]

then

\[
\text{TG}_M = 99 \tag{216}
\]

and \textit{acis\_process\_events} produces a warning upon completion with a count of the total number of events for which this condition is true. These values should not occur.
8. CHIPY\_TG:

(a) CC mode with gratings:

If

\[
\text{OBS\_MODE} = \text{pointing or POINTEING} \tag{217}
\]

\[
\text{OBS\_MODE} = \text{POINTING} \tag{218}
\]

and

\[
\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or CC33\_GRADED} \tag{219}
\]

\[
\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \tag{220}
\]

and

\[
\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \tag{221}
\]

then

i. Read:

The value of CHIPY\_TG for an event is given by CHIPY\_TG\_in.

ii. Validation:

A. If

\[
\text{TG\_M} > -99 \quad \text{and} \quad \text{TG\_M} < 99, \tag{222}
\]

\[
\text{TG\_M} > -99 \quad \text{and} \quad \text{TG\_M} < 99, \tag{223}
\]

and if

\[
\text{CHIPY\_TG} \leq 0 \quad \text{or} \quad \text{CHIPY\_TG} \geq 1025, \tag{224}
\]

\[
\text{CHIPY\_TG} \geq 1025, \tag{225}
\]

then acis\_process\_events exits with an error message because CHIPY\_TG-dependent computations could fail if the value of CHIPY\_TG is unphysical.

B. If

\[
\text{TG\_M} > -99, \tag{226}
\]

\[
\text{TG\_M} < 99, \text{and} \tag{227}
\]

\[
\text{CHIPY\_TG} < 1, \tag{228}
\]

then

\[
\text{CHIPY\_TG} = 1. \tag{229}
\]

C. If

\[
\text{TG\_M} > -99, \tag{230}
\]

\[
\text{TG\_M} < 99, \text{and} \tag{231}
\]

\[
\text{CHIPY\_TG} > 1024, \tag{232}
\]

then

\[
\text{CHIPY\_TG} = 1024. \tag{233}
\]
9. **CHIPY,Z0**:  
   (a) CC mode with gratings:  
   If  
   
   $\text{OBS\_MODE} = \text{pointing or}$ \hspace{1cm} (234)  
   $\text{OBS\_MODE} = \text{POINTING}$ \hspace{1cm} (235)  

   and  
   
   $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or}$ \hspace{1cm} (236)  
   $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$ \hspace{1cm} (237)  

   and  
   
   $\text{CONTENT}_{\text{in}} = \text{TGEVT1},$ \hspace{1cm} (238)  

   then  
   i. Read:  
   The value of \text{CHIPY,Z0} for an event is given by \text{CHIPY,Z0}_{\text{in}}.

10. **TIME,R0**:  
   (a) CC mode:  
   If  
   
   $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or}$ \hspace{1cm} (239)  
   $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED},$ \hspace{1cm} (240)  

   then  
   i. Read:  
   A. Level 0:  
   If  
   
   $\text{CONTENT}_{\text{in}} = \text{EVT0},$ \hspace{1cm} (241)  

   then the value of \text{TIME,R0} for an event is given by \text{TIME}_{\text{in}}.  
   B. Level 1, 1.5, or 2:  
   If  
   
   $\text{CONTENT}_{\text{in}} = \text{EVT1},$ \hspace{1cm} (242)  
   $\text{CONTENT}_{\text{in}} = \text{TGEVT1},$ \hspace{1cm} (243)  
   $\text{CONTENT}_{\text{in}} = \text{EVT2},$ \hspace{1cm} (244)  

   then the value of \text{TIME,R0} for an event is given by \text{TIME,R0}_{\text{in}}.  

   ii. Validation:  
   If  
   
   \begin{align*}  
   \text{TIME,R0} &\ < \ 0 \hspace{1cm} \text{(245)} 
   
   \text{TIME,R0} &\ \geq \ 3 \times 10^9, \hspace{1cm} \text{(246)}  
   \end{align*}  

   then \text{acis\_process\_events} produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. These values should not occur.
11. **TIME and CHIPY\_ADJ:**

(a) Read or calculate:

i. **TE mode:**

If

\[
\text{DATAMODE}_\text{in} = \text{FAINT or (247)} \\
\text{DATAMODE}_\text{in} = \text{FAINT\_BIAS or (248)} \\
\text{DATAMODE}_\text{in} = \text{GRADED or (249)} \\
\text{DATAMODE}_\text{in} = \text{VFAINT, (250)}
\]

then

\[
\text{TIME} = \text{TIME}_\text{in} \quad \text{and (251)} \\
\text{CHIPY\_ADJ} = \text{CHIPY. (252)}
\]

ii. **Pointing CC mode without grating data:**

If

\[
\text{OBS\_MODE} = \text{pointing or (253)} \\
\text{OBS\_MODE} = \text{POINTING (254)}
\]

and

\[
\text{DATAMODE}_\text{in} = \text{CC33\_FAINT or (255)} \\
\text{DATAMODE}_\text{in} = \text{CC33\_GRADED (256)}
\]

and

\[
\text{CONTENT}_\text{in} = \text{EVT0 or (257)} \\
\text{CONTENT}_\text{in} = \text{EVT1 or (258)} \\
\text{CONTENT}_\text{in} = \text{EVT2, (259)}
\]

then

A. **TIME\':**

The approximate time of arrival

\[
\text{TIME}' = \text{TIME}_\text{RO} - (512 + 1028) \times \text{TIMEDEL}_\text{in}. (260)
\]

B. **CHIPY\_ADJ\':**

If

\[
\text{CCD\_ID}_\text{focus} \geq 0 \quad \text{and (261)} \\
\text{CCD\_ID}_\text{focus} \leq 3, (262)
\]

then **CHIPY\_ADJ\'** (the approximate value of **CHIPY\_ADJ**) is given by the **CHIPY** location (on the focal-point CCD) of the coordinates **RA\_ADJ\_I** and **DEC\_ADJ\_I** at the time **TIME\'**. If \(\text{TIME}' < \text{TIME\_min or TIME}' > \text{TIME\_max}\), then \(\text{TIME}_{c}\) is used instead of **TIME\'**. If

\[
\text{CCD\_ID}_\text{focus} \geq 4 \quad \text{and (263)} \\
\text{CCD\_ID}_\text{focus} \leq 9, (264)
\]

then **CHIPY\_ADJ\'** is given by the **CHIPY** location (on the focal-point CCD) of the coordinates **RA\_ADJ\_S** and **DEC\_ADJ\_S** at the time **TIME\'**. If \(\text{TIME}' < \text{TIME\_min or TIME}' > \text{TIME\_max}\), then \(\text{TIME}_{c}\) is used instead of **TIME\'**.
C. TIME:
The value of CHIPY\_ADJ is used to obtain a better estimate of the time of arrival

\[ \text{TIME} = \text{TIME}_R - (\text{CHIPY\_ADJ} + 1028) \times \text{TIMEDEL}_{in}. \]  

(265)

D. CHIPY\_ADJ:
If

\[
\begin{align*}
\text{CCD\_ID}_{focus} & \geq 0 \quad \text{and} \\
\text{CCD\_ID}_{focus} & \leq 3,
\end{align*}
\]

(266) (267)

then the value of CHIPY\_ADJ is given by the CHIPY location (on the focal-point CCD) of the coordinates RA\_ADJ\_I and DEC\_ADJ\_I at the time \text{TIME}. If \text{TIME} < \text{TIME}_{\text{min}} or \text{TIME} > \text{TIME}_{\text{max}}, then \text{TIME}_{c} is used instead of \text{TIME}. If

\[
\begin{align*}
\text{CCD\_ID}_{focus} & \geq 4 \quad \text{and} \\
\text{CCD\_ID}_{focus} & \leq 9,
\end{align*}
\]

(268) (269)

then the value of CHIPY\_ADJ is given by the CHIPY location (on the focal-point CCD) of the coordinates RA\_ADJ\_S and DEC\_ADJ\_S at the time \text{TIME}. If \text{TIME} < \text{TIME}_{\text{min}} or \text{TIME} > \text{TIME}_{\text{max}}, then \text{TIME}_{c} is used instead of \text{TIME}.

iii. Pointing CC mode with ACIS-S grating data:
If

\[
\begin{align*}
\text{OBS\_MODE} = \text{pointing} \quad \text{or} \\
\text{OBS\_MODE} = \text{POINTING}
\end{align*}
\]

(270) (271)

and

\[
\begin{align*}
\text{DATAMODE}_{in} & = \text{CC33\_FAINT} \quad \text{or} \\
\text{DATAMODE}_{in} & = \text{CC33\_GRADED}
\end{align*}
\]

(272) (273)

and

\[
\text{CONENT}_{in} = \text{TGEVT1}
\]

(274)

and

\[
\begin{align*}
\text{CCD\_ID}_{focus} & \geq 4 \quad \text{and} \\
\text{CCD\_ID}_{focus} & \leq 9,
\end{align*}
\]

(275) (276)

then

A. Source events:
If

\[
\begin{align*}
\text{TG\_M} & > -99 \quad \text{and} \\
\text{TG\_M} & < 99,
\end{align*}
\]

(277) (278)

then

\[
\begin{align*}
\text{CHIPY\_ADJ} & = \text{CHIPY\_TG} \quad \text{and} \\
\text{TIME} & = \text{TIME}_R - (\text{CHIPY\_ADJ} + 1028) \times \text{TIMEDEL}_{in}
\end{align*}
\]

(279) (280)

If the event does not occur during a good-time interval, then CHIPY\_ADJ = CHIPY\_Z0\_med instead of CHIPY\_TG in equation 279.
B. Background events with zeroth order on the array:

If

\[ TG_M = -99 \text{ or } \]
\[ TG_M = 99 \] \hspace{1cm} (281)

and if

\[ CHIPY_{ZO_{min}} \geq 0.5 \text{ and } \]
\[ CHIPY_{ZO_{max}} < 1024.5, \] \hspace{1cm} (283)

then

\[ CHIPY_{ADJ} = CHIPY_{ZO} \text{ and } \]
\[ TIME = TIME_{RO} - (CHIPY_{ADJ} + 1028) \times TIMEDEL_{in}. \] \hspace{1cm} (285)

If the event does not occur during a good-time interval, then \( CHIPY_{ADJ} = CHIPY_{ZO_{med}} \) instead of \( CHIPY_{ZO} \) in equation 285.

C. Background events with zeroth order off the array:

If

\[ TG_M = -99 \text{ or } \]
\[ TG_M = 99 \] \hspace{1cm} (287)

and if

\[ CHIPY_{ZO_{max}} < 0.5 \text{ or } \]
\[ CHIPY_{ZO_{min}} \geq 1024.5, \] \hspace{1cm} (289)

then

\[ CHIPY_{ADJ} = 512 + (CHIPY_{ZO} - CHIPY_{ZO_{med}}) \text{ and } \]
\[ TIME = TIME_{RO} - (CHIPY_{ADJ} + 1028) \times TIMEDEL_{in}. \] \hspace{1cm} (291)

If the event does not occur during a good-time interval, then \( CHIPY_{ADJ} = 512 \) instead of \( 512 + CHIPY_{ZO} - CHIPY_{ZO_{med}} \) in equation 291.

iv. Pointing CC mode with ACIS-I grating data:

If

\[ OBS_{MODE} = \text{pointing or } \]
\[ OBS_{MODE} = \text{POINTING} \] \hspace{1cm} (293)

and

\[ DATAMODE_{in} = \text{CC33,FAINT or } \]
\[ DATAMODE_{in} = \text{CC33,GRADED} \] \hspace{1cm} (295)

and

\[ CONTENT_{in} = \text{TGEVT1} \] \hspace{1cm} (297)

and

\[ CCD_{ID_{focus}} \geq 0 \text{ and } \]
\[ CCD_{ID_{focus}} \leq 3, \] \hspace{1cm} (299)

then
A. **TIME’**:  
The approximate time of arrival  

\[
\text{TIME’} = \text{TIME}_\text{RO} - (512 + 1028) \times \text{TIMEDEL}_{\text{in}}. \tag{300}
\]

B. **CHIPY\_ADJ’**:  

CHIPY\_ADJ’ (the approximate value of CHIPY\_ADJ) is given by the CHIPY location (on the focal-point CCD) of the coordinates RA\_ADJ\_I and DEC\_ADJ\_I at the time **TIME’**. If the event does not occur during a good-time interval, then **TIMEc** is used instead of **TIME’**.

C. **TIME**:  
The value of CHIPY\_ADJ’ is used to obtain a better estimate of the time of arrival  

\[
\text{TIME} = \text{TIME}_\text{RO} - (\text{CHIPY\_ADJ’} + 1028) \times \text{TIMEDEL}_{\text{in}}. \tag{301}
\]

D. **CHIPY\_ADJ**:  
The value of CHIPY\_ADJ is given by the CHIPY location (on the focal-point CCD) of the coordinates RA\_ADJ\_I and DEC\_ADJ\_I at the time **TIME**. If the event does not occur during a good-time interval, then **TIMEc** is used instead of **TIME**.

v. Secondary CC mode:  
If  

\[
\text{OBS\_MODE} \neq \text{pointing and } \text{OBS\_MODE} \neq \text{POINTING}, \tag{302}
\]

then

A. **TIME**:  

\[
\text{TIME} = \text{TIME}_\text{RO} - (512 + 1028) \times \text{TIMEDEL}_{\text{in}}. \tag{304}
\]

B. **CHIPY\_ADJ**:  

\[
\text{CHIPY\_ADJ} = 512. \tag{305}
\]

(b) Validation:

i. If  

\[
\text{TIME} < 0 \text{ or } \text{TIME} \geq 3 \times 10^9, \tag{306}
\]

then **acis_process_events** produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. These values should not occur.

ii. If  

\[
\text{CHIPY\_ADJ} < 0.5 \text{ or } \text{CHIPY\_ADJ} \geq 1024.5, \tag{308}
\]

then **acis_process_events** exits with an error message because CHIPY\_ADJ-dependent computations could fail if the value of CHIPY\_ADJ is unphysical.

12. Bad pixel:
(a) If

\[ \text{badpixfile} \neq \text{none} \quad \text{and} \quad \text{badpixfile} \neq \text{NONE} \]

and if the \text{badpixfile} includes a valid HDU \( h_{\text{badpix}} \) where \( \text{CCD}_i \text{ID}_{\text{badpix}} = \text{CCD}_i \text{ID} \), then the HDU \( h_{\text{badpix}} \) is searched as follows to determine if the event should have one or more \text{STATUS} bits set to one.

i. If \( \text{DATAMODE}_m = \text{CC33\_FAINT} \) or \( \text{DATAMODE}_m = \text{CC33\_GRADED} \) and if there are one or more rows \( r \) in HDU \( h_{\text{badpix}} \) where

\[ \begin{align*}
\text{CHIPX} & \geq \text{CHIPX}_{\text{badpix},r}[0], \\
\text{CHIPX} & \leq \text{CHIPX}_{\text{badpix},r}[1], \\
\text{TIME} & \geq \text{TIME}_{\text{badpix},r}, \\
\text{TIME} & < \text{TIME\_STOP}_{\text{badpix},r}
\end{align*} \]

and where

\[ \begin{align*}
\text{STATUS}_{\text{badpix},r}[5] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[6] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[9] &= 1
\end{align*} \]

then

\[ \text{STATUS}[0] = 1 \]

for the event. Here \( \text{CCD}_i \text{ID}_{\text{badpix}} \) is the value of the keyword \( \text{CCD}_i \text{ID} \) in HDU \( h_{\text{badpix}} \) of the \text{badpixfile}, \( \text{CHIPX}_{\text{badpix},r}[0] \) and \( \text{CHIPX}_{\text{badpix},r}[1] \) are the first and second values in the vector column named \( \text{CHIPX} \) of row \( r \) of HDU \( h_{\text{badpix}} \) of the \text{badpixfile}, and \( \text{TIME}_{\text{badpix},r} \) and \( \text{TIME\_STOP}_{\text{badpix},r} \) are the values in the columns named \( \text{TIME} \) and \( \text{TIME\_STOP} \), respectively, of row \( r \) of HDU \( h_{\text{badpix}} \) of the \text{badpixfile}.

ii. If \( \text{DATAMODE}_m = \text{CC33\_FAINT} \) or \( \text{DATAMODE}_m = \text{CC33\_GRADED} \) and if there are one or more rows \( r \) in HDU \( h_{\text{badpix}} \) where

\[ \begin{align*}
\text{CHIPX} & \geq \text{CHIPX}_{\text{badpix},r}[0], \\
\text{CHIPX} & \leq \text{CHIPX}_{\text{badpix},r}[1], \\
\text{TIME} & \geq \text{TIME}_{\text{badpix},r}, \\
\text{TIME} & < \text{TIME\_STOP}_{\text{badpix},r}
\end{align*} \]

and where

\[ \begin{align*}
\text{STATUS}_{\text{badpix},r}[0] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[1] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[7] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[11] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[12] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[13] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[14] &= 1 \quad \text{or} \\
\text{STATUS}_{\text{badpix},r}[16] &= 1
\end{align*} \]

then

\[ \text{STATUS}[4] = 1 \]

for the event.
iii. If DATAMODE$_{in}$ = CC33.FAINT or DATAMODE$_{in}$ = CC33.GRADED and if there are one or more rows $r$ in HDU $h_{badpix}$ where

\[
\begin{align*}
\text{CHIPX} & \geq \text{CHIPX}_{badpix,r}[0], \\
\text{CHIPX} & \leq \text{CHIPX}_{badpix,r}[1], \\
\text{TIME} & \geq \text{TIM}_{badpix,r}, \\
\text{TIME} & < \text{TIM}_{STOP_{badpix,r}}
\end{align*}
\]  

and where

\[
\begin{align*}
\text{STATUS}_{badpix,r}[8] & = 1 \text{ or } \\
\text{STATUS}_{badpix,r}[10] & = 1,
\end{align*}
\]

then

\[\text{STATUS}[5] = 1\]

for the event.

iv. If DATAMODE$_{in}$ = CC33.FAINT or DATAMODE$_{in}$ = CC33.GRADED and if there are one or more rows $r$ in HDU $h_{badpix}$ where

\[
\begin{align*}
\text{CHIPX} & \geq \text{CHIPX}_{badpix,r}[0], \\
\text{CHIPX} & \leq \text{CHIPX}_{badpix,r}[1], \\
\text{TIME} & \geq \text{TIM}_{badpix,r}, \\
\text{TIME} & < \text{TIM}_{STOP_{badpix,r}}
\end{align*}
\]

and where

\[
\begin{align*}
\text{STATUS}_{badpix,r}[3] & = 1,
\end{align*}
\]

then

\[\text{STATUS}[6] = 1\]

for the event.

v. If DATAMODE$_{in}$ = CC33.FAINT or DATAMODE$_{in}$ = CC33.GRADED and if there are one or more rows $r$ in HDU $h_{badpix}$ where

\[
\begin{align*}
\text{CHIPX} & \geq \text{CHIPX}_{badpix,r}[0], \\
\text{CHIPX} & \leq \text{CHIPX}_{badpix,r}[1], \\
\text{TIME} & \geq \text{TIM}_{badpix,r}, \\
\text{TIME} & < \text{TIM}_{STOP_{badpix,r}}
\end{align*}
\]

and where

\[
\begin{align*}
\text{STATUS}_{badpix,r}[2] & = 1 \text{ or } \\
\text{STATUS}_{badpix,r}[4] & = 1,
\end{align*}
\]

then

\[\text{STATUS}[8] = 1\]

for the event.
vi. If $\text{DATAMODE}_{\text{in}} = \text{CC33}_\text{FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33}_\text{GRADED}$ and if there are one or more rows $r$ in HDU $h_{\text{badpix}}$ where

$$\begin{align*}
\text{CHIPX} & \geq \text{CHIPX}_{\text{badpix},r}[0], \\
\text{CHIPX} & \leq \text{CHIPX}_{\text{badpix},r}[1], \\
\text{TIME} & \geq \text{TIME}_{\text{badpix},r}, \\
\text{TIME} & < \text{TIME}_{\text{STOP}_{\text{badpix},r}}
\end{align*}$$

(353) (354) (355) (356)

and where

$$\text{STATUS}_{\text{badpix},r}[15] = 1,$$

(357)

then

$$\text{STATUS}[16] = 1$$

(358)

for the event.

vii. In summary, the mapping between a bad-pixel STATUS bit and the corresponding event STATUS bit is listed in Table 1.

<table>
<thead>
<tr>
<th>Bad-pixel STATUS bit</th>
<th>Event STATUS bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1: Bad-pixel to event STATUS bit mapping

13. PHAS:

(a) If HDU 1 of the infile includes the column PHAS, then

i. the values of PHAS for an event are read from the infile.

ii. If PHAS[4] < the split threshold, then STATUS[k] = 1 for bit $k = 1$.

iii. If PHAS[4] $\leq$ PHAS[j], for one or more $j = 0–3$ or $5–8$, then STATUS[k] = 1 for bit $k = 1$.

iv. If PHAS[j] > 4095, for one or more $j = 0–8$, then STATUS[k] = 1 for bit $k = 2$.

14. PHAS\_ADJ:

(a) If HDU 1 of the infile includes $\text{DATAMODE}_{\text{in}} = \text{CC33}_\text{FAINT}$, if the parameter apply\_cti = yes, and if the ctifile and mtlfile are specified, then the CTI-adjusted pulse heights are computed as follows.
i. The real-valued arrays for the serial CTI adjustment \( \Delta_x \), the parallel CTI adjustment \( \Delta_y \), and the adjusted pulse heights \( \text{PHAS}_\text{ADJ} \) are initialized such that

\[
\begin{align*}
\Delta_x[j] &= 0, \\
\Delta_y[j] &= 0, \text{ and} \\
\text{PHAS}_\text{ADJ}[j] &= \text{PHAS}[j]
\end{align*}
\]

(359) (360) (361)

for every element \( j = 0 \text{--} 8 \), where the starting point for the adjusted pulse heights are the unadjusted pulse heights \( \text{PHAS} \). Note that the values of the unadjusted pulse heights \( \text{PHAS} \) remain unchanged to ensure that it is possible to remove the CTI adjustment or to reapply the adjustment if the algorithm or calibration data are modified.

ii. The CTI iteration counter \( n \) is initialized such that

\[
n = 1.
\]

(362)

iii. The temporary variables \( \Delta_x', \Delta_y', \) and \( \text{PHAS}_\text{ADJ}' \) are set such that

\[
\begin{align*}
\Delta_x'[j] &= \Delta_x[j], \\
\Delta_y'[j] &= \Delta_y[j], \text{ and} \\
\text{PHAS}_\text{ADJ}'[j] &= \text{PHAS}_\text{ADJ}[j]
\end{align*}
\]

(363) (364) (365)

for each element \( j \).

iv. A. If there is a serial CTI trap-density map in the ctifile for \( \text{CCD}_\text{ID} \) and if \( \text{NODE}_\text{ID} = 0 \) or 2, then the values of \( \Delta_x \) are given by

\[
\begin{align*}
\Delta_x[0] &= c_x[0]s_x\rho_x[0]V_x[0], \\
\Delta_x[1] &= c_x[1]s_x\rho_x[1]V_x[1] - c_x'[0]s_x\rho_x[0]V_x[0], \\
\end{align*}
\]

(366) (367) (368) (369) (370) (371) (372) (373) (374)

where

\[
c_x[j] = \begin{cases} 
0 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{split threshold} \\
& \text{for all } j, \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \\
& \text{PHAS}[j - 1] + \Delta_x'[j - 1] + \Delta_y'[j - 1] \\
& \text{for } j = 1, 2, 4, 5, 7, 8, \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold} \\
& \text{for } j = 0, 3, 6 \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \\
& \text{PHAS}[j - 1] + \Delta_x'[j - 1] + \Delta_y'[j - 1] \\
& \text{for } j = 1, 2, 4, 5, 7, 8.
\end{cases}
\]
\[ V_j(t) = \frac{(PHAS_j + \Delta \|PHAS\|_j + \Delta \|PHAT\|_j)}{(VOLUME_{X+1} - VOLUME_X + 1)} \]

(381)

\[ \rho_j = \rho \text{ serial trap density} \]

(382)

\[ \Delta \|PHAS\|_j < \text{split threshold} \]

(375)

\[ \Delta \|PHAT\|_j < \text{split threshold} \]

(377)

\[ T = \frac{(T_{CCTIX} - T_{CCTIX} - 1)}{\Delta \|PHAT\|_j} \]

(379)

\[ T = \frac{t + \Delta \|PHAT\|_j}{\Delta \|PHAT\|_j} \]

(380)

\[ T = \frac{t + \Delta \|PHAT\|_j}{\Delta \|PHAT\|_j} \]

(381)

\[ \Delta \|PHAS\|_j < \text{split threshold} \]

(375)

\[ j \rightarrow \text{CHIPX} = \text{CHIPY} = 256, 513, \text{or 768} \]

(382)

\[ \text{PHAS}\|j + 1| + \Delta \|PHAS\|_j + \Delta \|PHAT\|_j < \text{split threshold} \]

(377)
B. If there is a serial CTI trap-density map in the `ctifile` for `CCD_ID` and if `NODE_ID = 1` or 3, then the values of $\Delta_x$ are given by

\[
\begin{align*}
\Delta_x[0] &= c_x[0]s_x\rho_x[0]V_x[0] - c_x'[0]s_x\rho_x[1]V_x[1], \\
\Delta_x[8] &= c_x[8]s_x\rho_x[8]V_x[8].
\end{align*}
\]  

where

\[
c_x[j] = \begin{cases} 
0 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{split threshold} \\
& \text{for all } j, \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \\
& \text{PHAS}[j + 1] + \Delta_x'[j + 1] + \Delta_y'[j + 1] \\
& \text{(for } j = 0, 1, 3, 4, 6, 7), \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold} \\
& \text{(for } j = 2, 5, 8) \\
1 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \\
& \text{PHAS}[j + 1] + \Delta_x'[j + 1] + \Delta_y'[j + 1] \\
& \text{(for } j = 0, 1, 3, 4, 6, 7), \\
\end{cases}
\]
and \( s_x, T, t', t'_k, t'_{k+1}, \rho_x[j] \), and \( V_x[j] \) are given by equations. 375, 376, 377, 378, 379, 380, and 381, respectively.

v. If there is a parallel CTI trap-density map in the ctifile for CCD_ID, then the values of \( \Delta_y \) are given by

\[
\Delta_y[0] = c_y[0]s_y\rho_y[0]V_y[0], \quad (391)
\]
\[
\Delta_y[1] = c_y[1]s_y\rho_y[1]V_y[1], \quad (392)
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]

where

\[
c_y[j] = \begin{cases} 
0 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{split threshold} \\
 & \text{(for all } j) \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold} \\text{and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{PHAS}[j-3] + \Delta_x'[j-3] + \Delta_y'[j-3] \\
& \text{(for } j = 3, 4, 5, 6, 7, 8) \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold} \\text{and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{PHAS}[j-3] + \Delta_x'[j-3] + \Delta_y'[j-3] \\
& \text{(for } j = 0, 1, 2) \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold} \\text{and} \\
1 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold} \\text{and} \\
& \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{PHAS}[j-3] + \Delta_x'[j-3] + \Delta_y'[j-3] \\
& \text{(for } j = 3, 4, 5, 6, 7, 8) 
\end{cases}
\]
vii. A. If

$$|\text{PHAS}_{\text{ADJ}}[j] - \text{PHAS}_{\text{ADJ}}[j]| < \text{cticonverge} \text{ (for all } j\text{)}$$

and

$$n \leq \text{max\_cti\_iter},$$

then the computation of \text{PHAS}_{\text{ADJ}} is complete for the event.

\[s_y = 1 + \text{TCTIY}(T - \text{FP\_TEMPO}),\]  

\[(400)\]

\[V_y[j] = \left(\frac{\text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l}\right) (\text{VOLUME}_{Y_{l+1}} - \text{VOLUME}_{Y_l}) + \text{VOLUME}_{Y_l},\]

\[(402)\]

\[c'_y[j] = \begin{cases} 
0 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] < \text{split threshold or } \text{PHA}[j] \rightarrow \text{CHIPY} = 1 \text{ or } 1024 \text{ (for } j = 1, 2, 3, 4, 5), \\
\text{FRCTRL} & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] > \text{PHAS}[j + 3] + \Delta_x'[j + 3] + \Delta_y'[j + 3] \text{ and } \text{PHAS}[j + 3] + \Delta_x'[j + 3] + \Delta_y'[j + 3] \geq \text{split threshold (for } j = 0, 1, 2, 3, 4, 5), \\
1 & \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \leq \text{PHAS}[j + 3] + \Delta_x'[j + 3] + \Delta_y'[j + 3] \text{ and } \text{PHAS}[j] + \Delta_x'[j] + \Delta_y'[j] \geq \text{split threshold (for } j = 0, 1, 2, 3, 4, 5), 
\end{cases}\]

and \(T, t', t'_k, \text{ and } t'_{k+1}\), are given by equations. 376, 377, 378, and 379, respectively.

vi. The CTI-adjusted pulse heights

\[\text{PHAS}_{\text{ADJ}}[j] = \text{PHAS}[j] + \Delta_x[j] + \Delta_y[j]\]

\[(403)\]
B. If
\[ |\text{PHAS}\_\text{ADJ}'[j] - \text{PHAS}\_\text{ADJ}[j]| \geq \text{cticonverge} \text{ for one or more } j \text{ and } \]
\[ n < \max_{\text{cti}\_\text{iter}}, \]
then \( n = n + 1 \) and steps 1.5.14(a)iii–1.5.14(a)vii are repeated.

C. If
\[ |\text{PHAS}\_\text{ADJ}'[j] - \text{PHAS}\_\text{ADJ}[j]| \geq \text{cticonverge} \text{ for one or more } j \text{ and } \]
\[ n \geq \max_{\text{cti}\_\text{iter}}, \]
then no additional iterations are performed, the values of \text{PHAS}\_\text{ADJ}[j] from the most recent iteration are used as are, and \text{STATUS}[k] = 1 \text{ for bit } k = 20 \text{ to indicate that the CTI adjustment did not converge.}

15. \text{FLTGRADE}:

(a) If
\[ \text{DATAMODE}_n = \text{CC33\_FAINT and} \]
\[ \text{apply\_cti} = \text{yes}, \]
then
where
\[ c_1[j] = \begin{cases} 0 & \text{if \text{PHAS}\_\text{ADJ}[j] < \text{split threshold}} \\ 1 & \text{otherwise,} \end{cases} \]
and the elements \( j = 0–3 \) and \( 5–8 \) of \text{PHAS}\_\text{ADJ} are depicted in Figure 1.

(b) If
\[ \text{DATAMODE}_n = \text{CC33\_FAINT and} \]
\[ \text{apply\_cti} = \text{no}, \]
then
where
\[ c_1[j] = \begin{cases} 0 & \text{if \text{PHAS}[j] < \text{split threshold}} \\ 0 & \text{if \text{PHAS}[j] > 4095} \\ 0 & \text{if \text{PHAS}[j] > \text{PHAS}[4] \text{ for } j = 0–3} \\ 0 & \text{if \text{PHAS}[j] > \text{PHAS}[4] \text{ for } j = 5–8} \\ 1 & \text{otherwise.} \end{cases} \]

(c) If
\[ \text{DATAMODE}_n = \text{CC33\_GRADED}, \]
then the \text{FLTGRADE} of an event is equal to the value of \text{FLTGRADE} for the event in the infile.

16. \text{GRADE}:

(a) If the \text{gradesfile} is specified, then the \text{GRADE} of an event is determined from the \text{FLTGRADE} of the event as follows.
Figure 1: The relative CHIPX and CHI PY coordinates of the nine elements $j = 0–8$ of a 3 pixel × 3 pixel event island PHAS$_j$ or PHAS,ADJ$_j$.

i. The appropriate HDU of the grade file is identified. This HDU is the one where the header keyword CB10001 includes the DATAMODE of HDU 1 of the infile.

ii. The row $i$ of the appropriate HDU of the grade file is identified. This row is the one where

$$\text{FLTGRADE}_{\text{grade},i} = \text{FLTGRADE},$$

where FLTGRADE$_{\text{grade}}$ is a column in the grade file.

iii. The GRADE of the event is given by

$$\text{GRADE} = \text{GRADE}_{\text{grade},i},$$

where GRADE$_{\text{grade}}$ is a column in the grade file.

17. PHA,RO:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33,FAINT},$$

then

$$\text{PHA,RO} = \sum_{j=0}^{8} \beta[j]p[j],$$

where

i. $p[j] = \text{PHAS}[j]$.

ii. The elements $j = 0–8$ of PHAS are depicted in Figure 1.
iii. \( \beta[j] = 0 \) if \( p[j] < \) split threshold.  \( (424) \)

iv. \( \beta[j] = 0 \) if \( \begin{cases} p[j] > p[4] & (\text{for } j = 0-3) \\ p[j] \geq p[4] & (\text{for } j = 5-8) \end{cases} \)  \( (425) \)

v. If \( \text{CORNERS} = -1 \), then

\( \beta[0] = \beta[2] = \beta[6] = \beta[8] = 0. \)  \( (426) \)

vi. If \( \text{CORNERS} = 0 \), then there are no additional constraints on \( \beta[0], \beta[2], \beta[6], \) and \( \beta[8] \).

vii. If \( \text{CORNERS} = 1 \), then

\( \begin{align*} 
\beta[0] &= 0 \quad \text{if } \beta[1] = 0 \text{ and } \beta[3] = 0. \\
\beta[2] &= 0 \quad \text{if } \beta[1] = 0 \text{ and } \beta[5] = 0. \\
\beta[6] &= 0 \quad \text{if } \beta[3] = 0 \text{ and } \beta[7] = 0. \\
\beta[8] &= 0 \quad \text{if } \beta[5] = 0 \text{ and } \beta[7] = 0. 
\end{align*} \)

vi. If \( \text{CORNERS} = 2 \), then

\( \begin{align*} 
\beta[0] &= 0 \quad \text{if } \beta[1] = 0 \text{ or } \beta[3] = 0 \text{ or } \text{GRADE} \neq 6. \\
\beta[2] &= 0 \quad \text{if } \beta[1] = 0 \text{ or } \beta[5] = 0 \text{ or } \text{GRADE} \neq 6. \\
\beta[6] &= 0 \quad \text{if } \beta[3] = 0 \text{ or } \beta[7] = 0 \text{ or } \text{GRADE} \neq 6. \\
\beta[8] &= 0 \quad \text{if } \beta[5] = 0 \text{ or } \beta[7] = 0 \text{ or } \text{GRADE} \neq 6. 
\end{align*} \)

(b) If \( \text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \), then

i. If \( \text{CONTENT}_{\text{in}} = \text{EVT0} \), then the value of \( \text{PHA}_{\text{RO}} \) for the event is the value of \( \text{PHA} \) in the infile.

ii. If \( \text{CONTENT}_{\text{in}} = \text{EVT1}, \text{TGEVT1}, \) or \( \text{EVT2} \), then the value of \( \text{PHA}_{\text{RO}} \) for the event is the value of \( \text{PHA}_{\text{RO}} \) in the infile.

18. \( \text{PHA} \), including time-dependent gain:

(a) If \( \text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \), then

\( \text{PHA} = \sum_{j=0}^{8} \beta[j]p[j], \)  \( (439) \)

where

i. \( p[j] = \begin{cases} 
\text{PHAS}_{\text{ADJ}}[j] & \text{if } \text{apply}_\text{cti} = \text{yes} \\
\text{PHAS}[j] & \text{if } \text{apply}_\text{cti} = \text{no} 
\end{cases} \)  \( (440) \)

ii. The elements \( j = 0-8 \) of \( \text{PHAS}_{\text{ADJ}} \) (or \( \text{PHAS} \)) are depicted in Figure 1.
\[ \beta[j] = 0 \quad \text{if} \quad p[j] < \text{split threshold}. \quad (441) \]

iv. If the CTI adjustment is not performed, then

\[
\beta[j] = 0 \quad \text{if} \quad \begin{cases} 
p[j] > p[4] \quad \text{for} \quad j = 0 - 3 \\
p[j] \geq p[4] \quad \text{for} \quad j = 5 - 8
\end{cases} \quad (442)
\]

v. If CORNERS = –1, then

\[ \beta[0] = \beta[2] = \beta[6] = \beta[8] = 0. \quad (443) \]

vi. If CORNERS = 0, then there are no additional constraints on \(\beta[0], \beta[2], \beta[6],\) and \(\beta[8]\).

vii. If CORNERS = 1, then

\[
\begin{align*}
\beta[0] &= 0 \quad \text{if} \quad \beta[1] = 0 \text{ and } \beta[3] = 0. \\
\beta[2] &= 0 \quad \text{if} \quad \beta[1] = 0 \text{ and } \beta[5] = 0. \\
\beta[6] &= 0 \quad \text{if} \quad \beta[3] = 0 \text{ and } \beta[7] = 0. \\
\beta[8] &= 0 \quad \text{if} \quad \beta[5] = 0 \text{ and } \beta[7] = 0.
\end{align*} \quad (444 - 447) \]

viii. If CORNERS = 2, then

\[
\begin{align*}
\beta[0] &= 0 \quad \text{if} \quad \beta[1] = 0 \text{ or } \beta[3] = 0 \text{ or } \text{GRADE} \neq 6. \\
\beta[2] &= 0 \quad \text{if} \quad \beta[1] = 0 \text{ or } \beta[5] = 0 \text{ or } \text{GRADE} \neq 6. \\
\beta[6] &= 0 \quad \text{if} \quad \beta[3] = 0 \text{ or } \beta[7] = 0 \text{ or } \text{GRADE} \neq 6. \\
\beta[8] &= 0 \quad \text{if} \quad \beta[5] = 0 \text{ or } \beta[7] = 0 \text{ or } \text{GRADE} \neq 6. 
\end{align*} \quad (448 - 451) \]

(b) If \(\text{DATAMODE}_{\text{in}} = \text{CC33, GRADED},\) \(452\)

then the value of PHA for the event is read from the infile.

(c) If \(\text{apply tgain} = \text{yes},\) \(453\)

then

\[ \text{PHA} = \text{PHA} + \text{int} \left[ \frac{(\text{TIME} - \text{EPOCH1})}{(\text{EPOCH2} - \text{EPOCH1})} (\delta_2 - \delta_1) + \delta_1 + \epsilon \right], \quad (454) \]

where

\[ \text{int} = \text{the integer portion of (i.e. truncate or round down)}, \quad (455) \]
\[ \text{TIME} = \text{the time of the event}, \quad (456) \]
\[ \text{EPOCH1} = \text{a keyword in the tgainfile}, \quad (457) \]
\[ \text{EPOCH2} = \text{a keyword in the tgainfile}, \quad (458) \]
\[ \delta_1 = \left( \frac{\text{PHA} - \text{PHA}_m[r]}{\text{PHA}_{m+1}[r] - \text{PHA}_m[r]} \right) (\text{DELTPHA1}_{m}[r] - \text{DELTPHA1}_{m}[r]) + \delta_1 \quad (459) \]
\[ \text{DELTPHA1}_{m}[r], \quad (460) \]
\[ \delta_2 = \frac{(\text{PHA} - \text{PHA}_m[\text{r}])}{(\text{PHA}_{m+1}[\text{r}] - \text{PHA}_m[\text{r}])} \left( \text{DELTPHA2}_{m+1}[\text{r}] - \text{DELTPHA2}_m[\text{r}] \right) + \delta_1 \]  
\[ \epsilon = \text{a uniform random deviate in the range} \ [0, 1) \]  
\[ \{ \text{If rand}_{\text{pha}} = \text{no}, \text{then} \epsilon = 0. \} \]

(d) If \( \text{PHA} \geq 32767 \), then \( \text{STATUS}[k] = 1 \) for bit \( k = 3 \).

19. CORN\_PHA:

(a) If \( \text{DATAMODE}_{\text{in}} \neq \text{CC33\_GRADED} \), then the value of CORN\_PHA is read from the \text{infile}.

20. ENERGY:

(a) If the parameter \text{calculate}_{\text{pi}} = \text{yes}, if the parameter \text{gainfile} is specified, and if \( \text{PHA} > 0 \), then

i. The row \( i \) in the \text{gainfile} is identified such that

\[ \text{CCD\_ID} = \text{CCD\_ID}_{\text{gain},i}, \]
\[ \text{CHIPX\_MIN}_{\text{gain},i} \leq \text{CHIPX} \leq \text{CHIPX\_MAX}_{\text{gain},i}, \] and
\[ \text{CHIPY\_MIN}_{\text{gain},i} \leq \text{nint(\text{CHIPY\_ADJ})} \leq \text{CHIPY\_MAX}_{\text{gain},i}, \]

where CCD\_ID\text{gain}, CHIPX\_MIN\text{gain}, CHIPX\_MAX\text{gain}, CHIPY\_MIN\text{gain}, and CHIPY\_MAX\text{gain} are columns in the gainfile.

ii. A uniform random deviate \( \Delta p \) is computed over the interval from \([-0.5, +0.5)\).

iii. The element \( j \) of row \( i \) of PHA\text{gain} is identified such that

\[ \text{PHA}_{\text{gain},i}[j] \leq (\text{PHA} + \Delta p) < \text{PHA}_{\text{gain},i}[j+1], \]

where PHA\text{gain} is a vector column in the gainfile. If \( \text{PHA} + \Delta p < \text{PHA}_{\text{gain},i}[0] \), then \( j = 0 \). If \( \text{PHA}_{\text{gain},i}[\text{NPOINTS} - 2] \leq \text{PHA} + \Delta p \), then \( j = \text{NPOINTS} - 2 \), where NPOINTS is a column in the gainfile.
iv. The \textbf{ENERGY} of an event is computed from the PHA of the event:

\[
\text{ENERGY} = \left( \frac{\text{PHA} + \Delta p - \text{PHA}_{\text{gain},i}[j]}{\text{PHA}_{\text{gain},i}[j + 1] - \text{PHA}_{\text{gain},i}[j]} \right) (\text{ENERGY}_{\text{gain},i}[j + 1] - \text{ENERGY}_{\text{gain},i}[j]) + \text{ENERGY}_{\text{gain},i}[j],
\]

where \text{ENERGY}_{\text{gain}} is a vector column in the \textit{gainfile}.

v. If ENERGY < 0, then ENERGY = 0.

(b) If the parameter \textit{calculate\_pi} = yes, if the parameter \textit{gainfile} is specified, and if \text{PHA} \leq 0, then ENERGY = 0.

(c) If the parameter \textit{calculate\_pi} = no or if the parameter \textit{gainfile} is not specified, then

i. If the \textit{infile} includes the ENERGY of an event, then the ENERGY of the event is equal to the ENERGY in the \textit{infile}.

ii. If the \textit{infile} does not include the ENERGY of an event, then ENERGY = 0.

21. PI:

(a) If \textit{calculate\_pi} = yes,

\[
\text{calculate\_pi} = \text{yes},
\]

then

i. \[\text{PI} = \text{int} \left( \frac{\text{ENERGY}}{\text{pi\_bin\_width}} \right) + 1,\]

where “int” indicates the integer portion of what is in parentheses (i.e. the value is truncated or rounded down).

ii. If \[\text{PI} < 1,\]

then \text{PI} = 1.

iii. If \[\text{PI} > \text{pi\_num\_bins},\]

then \text{PI} = \text{pi\_num\_bins}.

(b) If \textit{calculate\_pi} = no

and if the \textit{infile} includes the value of PI for an event, then the value of PI is read from the \textit{infile}.

22. pix\_adj:

(a) centroid:

If \text{pix\_adj} = \text{centroid},

\[
\text{pix\_adj} = \text{centroid},
\]

then

\[
\text{CHIPX\_ADJ} = \text{CHIPX\_ADJ} - w'[0] + w'[2] - w'[3] + w'[5] - w'[6] + w'[8]
\]

and if

\[
\text{DATAMODE} = \text{FAINT} \text{ or } \text{FAINT\_BIAS} \text{ or } \text{GRADED} \text{ or } \text{VFAINT},
\]

38
then
\[ \text{CHIPY\_ADJ} = \text{CHIPY\_ADJ} - w'[0] - w'[1] - w'[2] + w'[6] + w'[7] + w'[8] \] (484)

and if
\[ \text{DATAMODE} = \text{CC33\_FAINT} \text{ or } \text{CC33\_GRADED}, \] (485)

then
\[ \text{TIME} = \text{TIME} + (w'[0] + w'[1] + w'[2] - w'[6] - w'[7] - w'[8]) \times \text{TIMEDEL}_{\text{in}}, \] (487)

where
\[ w'[j] = \frac{w[j]}{\sum_{j=0}^{8} w[j]}, \] (488)
\[ w[j] = \begin{cases} p[j] & \text{if the pixel is valid} \\ 0 & \text{if the pixel is invalid}, \end{cases} \] (489)
\[ p[j] = \begin{cases} \text{PHAS\_ADJ}[j] & \text{if apply\_cti = yes} \\ \text{PHAS}[j] & \text{if apply\_cti = no}, \end{cases} \] (490)

and the pixel is invalid if
\[ \beta[j] = 0 \text{ or } \text{STATUS}[0] = 1 \text{ or } \text{STATUS}[1] = 1 \text{ or } \text{STATUS}[2] = 1 \text{ or } \text{STATUS}[3] = 1 \text{ or } \text{STATUS}[4] = 1 \text{ or } \text{STATUS}[11] = 1 \text{ or } \text{STATUS}[13] = 1 \text{ or } \text{STATUS}[14] = 1 \text{ or } \text{STATUS}[15] = 1 \text{ or } \text{STATUS}[16] = 1. \] (501)

Note that it is possible for the centroid algorithm to yield an adjustment to CHIPX\_ADJ and/or CHIPY\_ADJ that is greater than half a pixel. However, the adjustment cannot equal or exceed one pixel.

(b) edser:
If
\[ \text{pix\_adj} = \text{edser}, \] (502)
then
\[ \text{CHIPX\_ADJ} = \text{CHIPX\_ADJ} + \left( \frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta X[k + 1] - \Delta X[k]) + \Delta X[k] \] (503)

and if
\[ \text{DATAMODE} = \text{FAINT} \text{ or } \text{FAINT\_BIAS} \text{ or } \text{GRADED} \text{ or } \text{VFAINT}, \] (504)

39
\[ \text{CHIPY_ADJ} = \text{CHIPY_ADJ} + \left( \frac{\text{ENERGY} - E[k]}{E[k + 1] - E[k]} \right) (\Delta Y[k + 1] - \Delta Y[k]) + \Delta Y[k] \] (508)

and if

\[ \text{DATAMODE} = \text{CC33_FAINT} \text{ or } \text{DATAMODE} = \text{CC33_GRADED} \] (509)

then

\[ \text{TIME} = \text{TIME} - \left( \frac{\text{ENERGY} - E[k]}{E[k + 1] - E[k]} \right) (\Delta Y[k + 1] - \Delta Y[k]) + \Delta Y[k] \times \text{TIMEDEL}_{\text{int}} \] (511)

where \( E[k] \) and \( E[k + 1] \), \( X[k] \) and \( X[k + 1] \), and \( Y[k] \) and \( Y[k + 1] \) are the \( k \) and \( (k+1) \)th elements of the vector columns \( \text{ENERGY}_{\text{subpix}} \), \( \text{CHIPX OFFSET}_{\text{subpix}} \), and \( \text{CHIPY OFFSET}_{\text{subpix}} \), respectively. These columns are in the HDU of the \text{subpixfile} where the value of the keyword \text{CCD ID} is equal to the value of the \text{CCD ID} of the event. The appropriate row of these columns is the one where \( \text{FLTGRADE}_{\text{subpix}} = \text{FLTGRADE} \). The values of \( k \) are the ones where

\[ \text{ENERGY} \geq E[k] \text{ and } \text{ENERGY} < E[k + 1]. \] (512)

Note that if

\[ \text{ENERGY} \leq E[0], \] (514)

then \( k = 0 \). Similarly, if

\[ \text{ENERGY} \geq E[NPOINTS_{\text{subpix}} - 2], \] (515)

then \( k = NPOINTS_{\text{subpix}} - 2 \).

(c) none:
If \( \text{pix adj} = \text{none} \),
then the values of \( \text{CHIPX_ADJ} \) and \( \text{CHIPY_ADJ} \) remain unchanged.

(d) randomize:
If \( \text{pix adj} = \text{randomize} \),
then
\[ \text{CHIPX_ADJ} = \text{CHIPX_ADJ} + \epsilon_x \] (518)

and if

\[ \text{DATAMODE} = \text{FAINT} \text{ or } \text{DATAMODE} = \text{FAINT_BIAS} \text{ or } \text{DATAMODE} = \text{GRADED} \text{ or } \text{DATAMODE} = \text{VFAINT}, \] (519)

then
\[ \text{CHIPY_ADJ} = \text{CHIPY_ADJ} + \epsilon_y \] (523)
and if

\[
\begin{align*}
\text{DATAMODE} &= \text{CC33}_\text{FAINT} \text{ or } \text{CC33}_\text{GRADED}, \\
\text{DATAMODE} &= \text{CC33}_\text{FAINT} \text{ or } \text{CC33}_\text{GRADED},
\end{align*}
\]

then

\[
\text{TIME} = \text{TIME} - \epsilon_y \times \text{TIMEDEL}_{\text{in}},
\]

where \(\epsilon_x\) and \(\epsilon_y\) are uniform random deviates in the range \([-0.5, 0.5]\) pixel.

(e) If

\[
\text{CHIPX}_{\text{ADJ}} < 0.5,
\]

then

\[
\text{CHIPX}_{\text{ADJ}} = 1.
\]

(f) If

\[
\text{CHIPX}_{\text{ADJ}} \geq 1024.5,
\]

then

\[
\text{CHIPX}_{\text{ADJ}} = 1024.
\]

(g) If

\[
\text{CHIPY}_{\text{ADJ}} < 0.5,
\]

then

\[
\text{CHIPY}_{\text{ADJ}} = 1.
\]

(h) If

\[
\text{CHIPY}_{\text{ADJ}} \geq 1024.5,
\]

then

\[
\text{CHIPY}_{\text{ADJ}} = 1024.
\]

23. \text{TDETX} and \text{TDETY}:

(a) If

\[
\begin{align*}
\text{stop} &= \text{tdet or } \\
\text{stop} &= \text{det or } \\
\text{stop} &= \text{tan or } \\
\text{stop} &= \text{sky},
\end{align*}
\]

then the values of \text{TDETX} and \text{TDETY} are computed using the values of \text{nint}((\text{CHIPX}_{\text{ADJ}}) and \text{nint}((\text{CHIPY}_{\text{ADJ}}). Here, “nint” indicates that the real-valued coordinate is rounded to the nearest integer before the computation of the TDET coordinate.

24. \text{DETX} and \text{DETY}:
(a) If

\[
\begin{align*}
\text{stop} &= \text{det or} \\
\text{stop} &= \text{tan or} \\
\text{stop} &= \text{sky},
\end{align*}
\]

then the values of DETX and DETY are computed using the real-valued coordinates CHIPX\_ADJ and CHIPY\_ADJ.

25. X and Y:
   (a) If

\[
\text{stop} = \text{sky},
\]

then the values of X and Y are computed using the real-valued coordinates CHIPX\_ADJ and CHIPY\_ADJ.

26. SKY\_1D:
   (a) If

\[
\begin{align*}
\text{DATAMODE}_{in} &= \text{CC33\_FAINT or} \\
\text{DATAMODE}_{in} &= \text{CC33\_GRADED}
\end{align*}
\]

and if

\[
\text{stop} = \text{sky},
\]

then the value of SKY\_1D is computed.

1.5.4 Write outfile

1. PIX\_ADJ:
   (a) If

\[
\text{pix\_adj} = \text{centroid},
\]

then

\[
\text{PIX\_ADJ} = \text{CENTROID},
\]

(b) If

\[
\text{pix\_adj} = \text{edser},
\]

then

\[
\text{PIX\_ADJ} = \text{EDSER},
\]

(c) If

\[
\text{pix\_adj} = \text{none},
\]

then

\[
\text{PIX\_ADJ} = \text{NONE},
\]
(d) If
\[ \text{pix}_\text{adj} = \text{randomize}, \]
then
\[ \text{PIX}_\text{ADJ} = \text{RANDOMIZE}, \]

2. **RAND\_SKY:**
(a) If
\[ \text{pix}_\text{adj} = \text{centroid}, \]
then
\[ \text{RAND\_SKY} = 0.0, \]

(b) If
\[ \text{pix}_\text{adj} = \text{edser}, \]
then
\[ \text{RAND\_SKY} = 0.0, \]

(c) If
\[ \text{pix}_\text{adj} = \text{none}, \]
then
\[ \text{RAND\_SKY} = 0.0, \]

(d) If
\[ \text{pix}_\text{adj} = \text{randomize}, \]
then
\[ \text{RAND\_SKY} = 0.5, \]

3. **TIME\_ADJ:**
(a) **TE mode:**
If
\[
\begin{align*}
\text{DATAMODE}_{\text{in}} & = \text{FAINT} \text{ or } \text{VFAINT}, \\
\text{DATAMODE}_{\text{in}} & = \text{FAINT\_BIAS} \text{ or } \text{GRADED}, \\
\end{align*}
\]
then
\[ \text{TIME\_ADJ} = \text{NONE}. \]
(b) Pointing CC mode without grating data:

i. If

\[
\text{OBS\_MODE} = \text{pointing or}\quad (567) \\
\text{OBS\_MODE} = \text{POINTING} \quad (568)
\]

and

\[
\text{DATAMODE}_\text{in} = \text{CC33\_FAINT or}\quad (569) \\
\text{DATAMODE}_\text{in} = \text{CC33\_GRADED} \quad (570)
\]

and

\[
\text{CONTENT}_\text{in} = \text{EVT0 or} \quad (571) \\
\text{CONTENT}_\text{in} = \text{EVT1 or} \quad (572) \\
\text{CONTENT}_\text{in} = \text{EVT2} \quad (573)
\]

and

\[
\text{CCD\_ID}_{\text{focus}} \geq 0 \quad \text{and} \quad (574) \\
\text{CCD\_ID}_{\text{focus}} \leq 3 \quad (575)
\]

and

\[
\cos(\text{DEC\_ADJ}_I) \cos(\text{DEC\_TARG}_\text{in}) \cos(\text{RA\_ADJ}_I - \text{RA\_TARG}_\text{in}) + \sin(\text{DEC\_ADJ}_I) \sin(\text{DEC\_TARG}_\text{in}) < 4.855 \times 10^{-11}, \quad (576)
\]

then

\[
\text{TIME\_ADJ} = \text{TARGET}. \quad (579)
\]

ii. If

\[
\text{OBS\_MODE} = \text{pointing or} \quad (580) \\
\text{OBS\_MODE} = \text{POINTING} \quad (581)
\]

and

\[
\text{DATAMODE}_\text{in} = \text{CC33\_FAINT or} \quad (582) \\
\text{DATAMODE}_\text{in} = \text{CC33\_GRADED} \quad (583)
\]

and

\[
\text{CONTENT}_\text{in} = \text{EVT0 or} \quad (584) \\
\text{CONTENT}_\text{in} = \text{EVT1 or} \quad (585) \\
\text{CONTENT}_\text{in} = \text{EVT2} \quad (586)
\]

and

\[
\text{CCD\_ID}_{\text{focus}} \geq 4 \quad \text{and} \quad (587) \\
\text{CCD\_ID}_{\text{focus}} \leq 9 \quad (588)
\]
and
\[
\cos(DEC_{\text{ADJ}}) \cos(DEC_{\text{TARGin}}) \cos(RA_{\text{ADJ}} - RA_{\text{TARGin}}) + \sin(DEC_{\text{ADJ}}) \sin(DEC_{\text{TARGin}}) < 4.855 \times 10^{-11},
\]

then
\[
\text{TIME}_{\text{ADJ}} = \text{TARGET}.
\]

iii. If
\[
\text{OBS}_{\text{MODE}} = \text{pointing or} \quad (593)
\]
\[
\text{OBS}_{\text{MODE}} = \text{POINTING} \quad (594)
\]
and
\[
\text{DATAMODE}_{\text{in}} = \text{CC33_FAIN} \text{T or} \quad (595)
\]
\[
\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (596)
\]
and
\[
\text{CONTENT}_{\text{in}} = \text{EVT0 or} \quad (597)
\]
\[
\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (598)
\]
\[
\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (599)
\]
and
\[
\text{CCD}_{\text{ID focus}} \geq 0 \text{ and} \quad (600)
\]
\[
\text{CCD}_{\text{ID focus}} \leq 3 \quad (601)
\]
and
\[
\cos(DEC_{\text{ADJ}}) \cos(DEC_{\text{TARGin}}) \cos(RA_{\text{ADJ}} - RA_{\text{TARGin}}) + \sin(DEC_{\text{ADJ}}) \sin(DEC_{\text{TARGin}}) \geq 4.855 \times 10^{-11},
\]

then
\[
\text{TIME}_{\text{ADJ}} = \text{AIMPOINT}.
\]

iv. If
\[
\text{OBS}_{\text{MODE}} = \text{pointing or} \quad (606)
\]
\[
\text{OBS}_{\text{MODE}} = \text{POINTING} \quad (607)
\]
and
\[
\text{DATAMODE}_{\text{in}} = \text{CC33_FAIN} \text{T or} \quad (608)
\]
\[
\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (609)
\]
and
\[
\text{CONTENT}_{\text{in}} = \text{EVT0 or} \quad (610)
\]
\[
\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (611)
\]
\[
\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (612)
\]
and

\[
\begin{align*}
\text{CCD\_ID\_focus} & \geq 4 \quad \text{and} \\
\text{CCD\_ID\_focus} & \leq 9
\end{align*}
\] (613)

and

\[
\begin{align*}
\cos (\text{DEC\_ADJS}) \cos (\text{DEC\_TARG}_{\text{in}}) \cos (\text{RA\_ADJS} - \text{RA\_TARG}_{\text{in}}) + \\
\sin (\text{DEC\_ADJS}) \sin (\text{DEC\_TARG}_{\text{in}}) & \geq 4.855 \times 10^{-11},
\end{align*}
\] (615)

then

\[
\text{TIME\_ADJ} = \text{AIMPOINT}.
\] (618)

(c) Pointing CC mode with ACIS-S grating data:

If

\[
\begin{align*}
\text{OBS\_MODE} & = \text{pointing or} \\
\text{OBS\_MODE} & = \text{POINTING}
\end{align*}
\] (619)

and

\[
\begin{align*}
\text{DATAMODE\_in} & = \text{CC33\_FAINT or} \\
\text{DATAMODE\_in} & = \text{CC33\_GRADED}
\end{align*}
\] (621)

and

\[
\text{CONTENT\_in} = \text{TGEVT1}
\] (623)

and

\[
\begin{align*}
\text{CCD\_ID\_focus} & \geq 4 \quad \text{and} \\
\text{CCD\_ID\_focus} & \leq 9,
\end{align*}
\] (624)

then

\[
\text{TIME\_ADJ} = \text{GRATING}.
\] (626)

(d) Pointing CC mode with ACIS-I grating data:

i. If

\[
\begin{align*}
\text{OBS\_MODE} & = \text{pointing or} \\
\text{OBS\_MODE} & = \text{POINTING}
\end{align*}
\] (627)

and

\[
\begin{align*}
\text{DATAMODE\_in} & = \text{CC33\_FAINT or} \\
\text{DATAMODE\_in} & = \text{CC33\_GRADED}
\end{align*}
\] (629)

and

\[
\text{CONTENT\_in} = \text{TGEVT1}
\] (631)
and

\[ \text{CCD}_\text{i} \text{focus} \geq 0 \quad \text{and} \quad \text{CCD}_\text{i} \text{focus} \leq 3 \]  

(632)

(633)

and

\[
\cos(\text{DEC}_\text{ADJ}_\text{i}) \cos(\text{DEC}_\text{TARG}_\text{in}) \cos(\text{RA}_\text{ADJ}_\text{i} - \text{RA}_\text{TARG}_\text{in}) + \\
\sin(\text{DEC}_\text{ADJ}_\text{i}) \sin(\text{DEC}_\text{TARG}_\text{in}) < 4.855 \times 10^{-11},
\]

(634)

(635)

then

\[ \text{TIME}_\text{ADJ} = \text{TARG}. \]  

(637)

ii. If

\[
\text{OBS}_\text{MODE} = \text{pointing or} \quad \text{OBS}_\text{MODE} = \text{POINTING}
\]

(638)

(639)

and

\[ \text{DATAMODE}_{\text{in}} = \text{CC33 FAINT} \quad \text{or} \quad \text{DATAMODE}_{\text{in}} = \text{CC33 GRADED} \]

(640)

(641)

and

\[ \text{CONTENT}_{\text{in}} = \text{TGEVT1} \]

(642)

and

\[ \text{CCD}_\text{i} \text{focus} \geq 0 \quad \text{and} \quad \text{CCD}_\text{i} \text{focus} \leq 3 \]  

(643)

(644)

and

\[
\cos(\text{DEC}_\text{ADJ}_\text{i}) \cos(\text{DEC}_\text{TARG}_\text{in}) \cos(\text{RA}_\text{ADJ}_\text{i} - \text{RA}_\text{TARG}_\text{in}) + \\
\sin(\text{DEC}_\text{ADJ}_\text{i}) \sin(\text{DEC}_\text{TARG}_\text{in}) \geq 4.855 \times 10^{-11},
\]

(645)

(646)

then

\[ \text{TIME}_\text{ADJ} = \text{AIMPOINT}. \]  

(648)

(c) Secondary CC mode:

If

\[
\text{OBS}_\text{MODE} \neq \text{pointing and} \quad \text{OBS}_\text{MODE} \neq \text{POINTING}
\]

(649)

(650)

and

\[ \text{DATAMODE}_{\text{in}} = \text{CC33 FAINT} \quad \text{or} \quad \text{DATAMODE}_{\text{in}} = \text{CC33 GRADED}. \]

(651)

(652)

then

\[ \text{TIME}_\text{ADJ} = \text{MIDCHIP}. \]  

(653)
2 TBD

- Complete the spec to include all of the timed exposure mode processing.
- Complete sections 1.1, 1.2, 1.3, and 1.4.
- Should CONTENTs other than EVT0, EVT1, TGEVT1, and EVT2 be included?
- Should CONTENT = EVT2 be dropped?
- Should DATAMODEs other than CC33_FAINT, CC33_GRADED, FAINT, FAINT_BIAS, GRADED, and VFAINT be included?
- Are the RA_TARG, DEC_TARG, RA_NOM, DEC_NOM, and TIMEDEL keywords in the output of afe (need obsfile sometimes)?
- What if TIME_RO is not in the infile (output of afe? EVT2 files?)?
- What if a small fraction of the values of CHIPY_TARG are off the chip due to bad aspect?
- Make sure that the STATUS bits are unset and reset properly.
- What about aoff and soff files instead of asol files?
- Are the $\beta$ in PHA_RO the same as the $\beta$ in PHA?
- Should something be done about SKY_1D?