*r e p o r t*

MIT Kavli Institute
1 Hampshire Street
Cambridge MA 02139–4307
Tel: 617-253-7277
Fax: 617-253-8084

To: ACIS Science Operations Team

From: Peter Ford, NE80-6071 <pgf@space.mit.edu>

Date: April 12th 2012

Subject: Selecting optimal parameters for the *txings* patch (v 1.1)

## 1. Introduction

The function of the *txings* patch to the ACIS instrument's flight software is to monitor threshold crossing rates—a measure of charged particle background flux—and to signal to the Chandra On-Board Computer (OBC) when the crossing rates increase by a significant amount. The patch uses an algorithm that depends on a number of parameters. This report describes how we go about determining the optimal values of those parameters.

## 2. The *txings* parameters

The patch contains three `TX` parameter structures, one (`TXblock`) for use by the current science run, a second (`TXnext`) for the next science run, and (`TXinit`) as a default. Immediately a new science run is started, the contents of `TXnext` replace `TXblock` and then `TXinit` replaces `TXnext`. Therefore, to change the parameters for the following science run, a *writeBep* command should be sent to update `TXnext`, whereas to change them for all subsequent runs, two *writeBep* commands must be issued, to update both `TXnext` and `TXinit`. Each structure contains 8 unsigned 32-bit integer fields, as shown in Fig. 1. The initial `TXinit` and `TXnext` values haven't changed since the patch was first uploaded in November 2011.

*Figure 1. The TX parameter structures and initial values*

```
struct TX {                         // parameter structure
    unsigned MINUTES;               // averaging 64-sec intervals
    unsigned TRIGGER_COUNT;         // threshold counter
    unsigned MAX_TX_PER_ROW;        // max crossings per row
    unsigned CC_TICKS;              // FEP ticks per frame in CC mode
    unsigned RATE_LIMIT[2];         // trigger thresholds/hundred-rows/sec
    unsigned TX_INCR[2];            // trigger threshold increments
} TXblock;

struct TX TXinit = { 5, 5, 512, 291840, { 700, 40 }, { 8, 8 } };
struct TX TXnext = { 5, 5, 512, 291840, { 700, 40 }, { 8, 8 } };
```

`MINUTES`

This specifies the length of each *txings* integration period. It is measured in units of software housekeeping interrupts. These occur every 640 BEP timer interrupts, each of which is at least 100 milliseconds. In practice, the housekeeping interrupts come at intervals of between 64 and 67 seconds depending on load.

`TRIGGER_COUNT`

This specifies the number of consecutive integration periods that the threshold crossing averages must remain above the rate limits before the patch triggers a radiation alert. There are separate rate limits for each CCD type: front- and back-illuminated, and either can trigger the alert. The current rates are determined by summing the threshold crossings reported by each CCD, dividing by its accumulated exposure time, and averaging across the CCDs of that type. *txings* will trigger if either of these rates remains above its `RATE_LIMIT` value *and* if it increases at least by its `TX_INCR` value for each integration period.

`MAX_TX_PER_ROW`

This parameter sets the maximum number of threshold crossings per CCD row. Above this, *txings* will assume that a FEP hardware error has occurred and will ignore the exposure. This feature was introduced to prevent a recurrence of known hardware errors from falsely triggering a radiation shutdown.

`CC_TICKS`

This specifies the number of 10-microsecond ticks per CCD readout in continuous clocking mode. It is used by *txings* to compute the exposure time of each output row. The nominal value is `291840`, which depends only on the *overclockPairsPerNode* and *outputRegisterMode* parameters, which are unlikely to change. There is no corresponding value for timed-exposure mode because it can easily be computed from internal parameters.

`RATE_LIMIT[2]`

This pair of parameters sets the limiting threshold rate values for front- and back-illuminated CCDs, respectively. Below these values, *txings* will not trigger. The rates are expressed in units of threshold crossings per CCD row per 100 seconds.

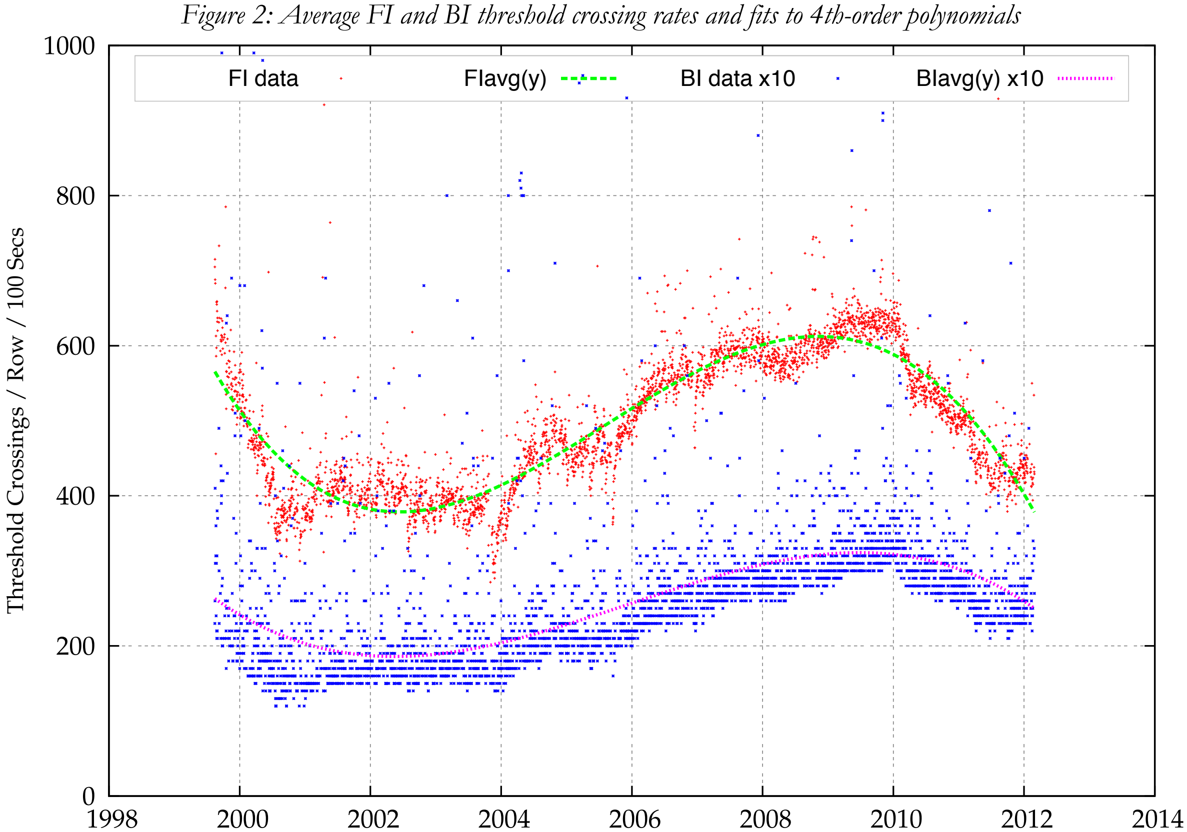
`TX_INCR[2]`

This pair of parameters sets the limiting threshold rate increment values for front- and back-illuminated CCDs, respectively. *txings* will not trigger if the average rate increments by less than this value from one integration period to the next. The rate increments are expressed in units of threshold crossings per CCD row per 100 seconds per integration period.

## 3. Variation in rate limits

The average threshold crossing rates vary with the charged particle background, which is principally determined by the 11-year solar cycle. The average background rates for front- and back-illuminated CCDs are shown in Fig. 2, along with 4th-order polynomial fits. The back-illuminated rates have been multiplied by 10 for better visibility. Only runs made in timed-exposure mode are shown, and runs that exhibited hardware anomalies and coincided with times of known high background radiation have also been excluded.

*Figure 2: Average FI and BI threshold crossing rates and fits to 4th-order polynomials*




The coefficients of recent the fits (`FIavg(y)` and `BIavg(y)`, `y` = calendar year) to the FI and BI threshold crossing rate averages in Figure 2 are listed in the Appendix as the `FIparm[]` and `BIparm[]` arrays, *i.e.*,

`FIavg(y) = FIparm[0] + (y-2000)*FIparm[1] + (y-2000)*(y-2000)*FIparm[2]...`

## 4. Patch tests

All of the event-mode data used by the *txings* patch is also reported in telemetry. The patch can therefore be tested on the ground, using data from the ACIS archive maintained at MIT. To do this while keeping *txings.C,* the patch code, unchanged, a wrapper was written to define several external BEP classes, as shown in Table 1.

*Table 1. External classes defined for use in txings.C.*

| Class | Methods | Instance Variables | Description |
|---|---|---|---|
| `BepReg` | `showLeds` | | A null class method for `Test_Leds::show`(). |
| `DebugProbe` | `DebugProbe` | | Dummy constructor for `Test_Leds::show`(). |
| `EventExposure` | | `expNum`<br>`expThresholdCnt`<br>`expParityErrs` | These instance variables are defined to keep the compiler happy but are otherwise unused. |
| `FEPexpEndRec` | | `thresholds`<br>`parityerrs`<br>`expnum` | The variables are defined to keep the compiler happy but are unused. The wrapper calls `saveTX-ings`() directly with `expnum` and `thresholds`. |
| `MemoryServer` | `readBep` | | Method called when `triggerRadmon()` triggers. |
| `PramCc` | | `summedRows`<br>`colSum` | A single instance of this class is defined to pass values to `TXings::saveTXings`(). |
| `PramTe` | | `exposureTime`<br>`dutyCycle`<br>`summedRows`<br>`sumFlag` | A single instance of this class is defined to pass values to `TXings::saveTXings`(). |
| `ProcessMode` | `getMode`<br>`getCcdId` | | The methods are defined for `copyExpEnd`() but never invoked. |
| `ScienceManager` | `isIdle` | `idle` | A single instance of this class is defined in order to pass `idle`, the current science run state, to `TXings::triggerRadmon`(). |
| `ScienceMode` | | | An instance of an empty class for use as a pointer. |
| `SmContClocking` | | | An instance of an empty class for use as a pointer. |
| `SmTimedExposure` | | | An instance of an empty class for use as a pointer. |
| `SwHousekeeper` | `report` | | A null class method called by `triggerRadmon`(). |

In addition, the wrapper, *txings_test.C*, embeds *txings_parms.h* to define anomalous OBSIDs and FI and BI threshold polynomials, and a pair of macro definition files, *cmdtab.h* and *tlmtab.h*, which are part of the *psci* and *pmon* packages, to extract bit fields from ACIS telemetry packets, and *interface.h*, a C-language header file containing many useful definitions of ACIS FEP and BEP structures and fields.

The resulting binary command, *txings_test*, reads an ACIS packet stream, *e.g.*, as output by *getnrt*, *getPackets*, *getp*, etc., and precisely simulates the action of the onboard *txings* patch. Normally, it reports the contents of its `TXblock` whenever it triggers, but in verbose mode (the `–v` flag), it also reports it at the end of each science run, *i.e.,* when it comes to a *scienceReport* , *fatalMessage* or *bepStartupMessage* packet.

The full list of *txings_test* options and positional parameters is shown in Table 2. 13 arguments are required, although the last two are ignored if the `-P` flag is used. Without `-P`, the actual rate thresholds are computed at the start of each science run as

```
y = date - 2000.0
RATE_LIMIT[0] = FI_threshold + y * FI_RATE
RATE_LIMIT[1] = BI_threshold + y * BI_RATE
```

where *date* is the starting time of the run in years. With `-P`, *txings_test* uses 4th-order polynomials:

```
RATE_LIMIT[0] = FI_threshold + y*(FIparm[1] + y*(FIparm[2] + y*(... + y*FIparm[4])))
RATE_LIMIT[1] = BI_threshold + y*(BIparm[1] + y*(BIparm[2] + y*(... + y*BIparm[4])))
```

where `FIparm[]` and `BIparm[]` are defined in *txings_parms.h* from the fitting process shown in Fig. 2.

*Table 2. Options and parameters for txings_test*

| Flag | Argument | Instance Variables |
|---|---|---|
| `-e` | `ticks` | Time the interrupts by "`ticks`" of the FEP clock. |
| `-s` | `frames` | Time the interrupts by minor frames, each lasting "`ticks`" ticks of the FEP clock. |
| `-D` | | List the contents of `TXblock` to *stdout* after every science run. |
| `-P` | | Ignore `FIrate` and `BIrate` in favor of `FIparm[]` and `BIparm[]` in *txings_parms.h*. |
| `-p` | | List the average rates after each call to `triggerRadmon`(). (Caution: much output.) |
| `-v` | | List the contents of `TXblock` after every science run and report telemetry gaps. |
| **Positional Arguments (required)** | | |
| `phase` | | Processing phase of first input science run |
| `run` | | Run number of first input science run |
| `year` | | Year of origin of first input science run |
| `minutes` | | Number of 64-second integration periods (*i.e.*, `TX.MINUTES`) |
| `triggers` | | Number of consecutive over-limit periods for trigger (*i.e.*, `TX.TRIGGER_COUNT`) |
| `maxPerRow` | | Maximum threshold crossings per CCD row (*i.e.*, `MAX_TX_PER_ROW`) |
| `ccTicks` | | Number of 10 microsecond ticks per full CC frame (*i.e.*, `CC_TICKS`) |
| `FI_threshold` | | Average rate threshold for front-illuminated CCDs in 2000 (*i.e.*, `RATE_LIMIT[0]`) |
| `BI_threshold` | | Average rate threshold for back-illuminated CCDs in 2000 (*i.e.*, `RATE_LIMIT[1]`) |
| `FI_increment` | | Minimum rate increase for front-illuminated CCDs (*i.e.*, `TX_INCR[0]`) |
| `BI_increment` | | Minimum rate increase for back-illuminated CCDs (*i.e.*, `TX_INCR[1]`) |
| `FI_rate` | | Rate of increase of `FI_threshold` per year (ignored if `-P` specified) |
| `BI_rate` | | Rate of increase of `BI_threshold` per year (ignored if `-P` specified) |

## 5. Results

The testing proceeded in three steps.

1. The threshold crossing rates for front- and back-illuminated CCDs were averaged for all science runs in the ACIS archive and these rates were fitted to 4th order polynomials, as illustrated in Fig. 2.
2. *txings_test* was run on all data since the end of the in-orbit checkout period in October 1999, a total of 13596 science runs. The *txings* parameters are shown in the "Archive" column of Table 3. Further testing was restricted to a subset of 521 observations that either (a) triggered during the first test, or (b) was known to have included a period of high background radiation (e.g., because it terminated in SCS107 from an EPHIN or HRC trigger, or ground command), or (c) observed the Crab Nebula, or (d) was known to possess an ACIS hardware anomaly that might result in a false *txings* trigger.
3. Each of the 521 science runs was processed by *txings_test* over each combination of parameters in the "Values Tested" column of Table 3, a total of 4,220,100 runs (`FI_increment` and `BI_increment` were varied together, and a total of 10 `FI_rate`/`BI_rate` sets were tested). The parameter values in the "Optimum" column show the values that resulted in the most "good" triggers (15) with zero false positives.

*Table 3. txings_test parameters*

| Parameter | Onboard | Archive | Values Tested | Optimum | Upload | Comment |
|---|---|---|---|---|---|---|
| `minutes` | 5 | 3 | 3,4,5 | 3 | 3 | |
| `triggers` | 5 | 3 | 3,4,5 | 5 | 5 | |
| `maxPerRow` | 512 | 512 | 512 | 512 | 512 | Not varied |
| `ccTicks` | 291840 | 291840 | 291840 | 291840 | 291840 | Not varied |
| `FI_threshold` | 700 | 600 | 670,680,690,700,710,720 | 700 | 626 | Varies with date |
| `BI_threshold` | 40 | 30 | 35,40,45,50,55 | 45 | 47 | Varies with date |
| `FI_increment` | 8 | 0 | 0,1,2 | 1 | 1 | |
| `BI_increment` | 8 | 0 | 0,1,2 | 1 | 1 | |
| `FI_rate` | 0 | FIparm | 1.0,1.75,2.5,FIparm | FIparm | – | See Appendix |
| `BI_rate` | 0 | BIparm | 1.0,1.5,2.0,BIparm | BIparm | – | See Appendix |

The 15 triggers are listed in Table 4. Under "FI" and "BI", the "Thresh" column shows the triggering rate, *i.e.*, `FI_threshold` or `BI_threshold`, and the "Value" column shows the rate (in boldface) at which *txings* triggered. Runs using `FIparm` and `BIparm` were found to be less sensitive to the values of the other parameters than those using `FI_rate` and `BI_rate`. In the rightmost column, `SCS107` denotes a run terminated by ground command, `EPHIN` one triggered onboard, either by the EPHIN or HRC instruments, and `HIRAD` describes runs that, in retrospect, should probably have been terminated due to high background radiation. The "Upload" column shows the values that should be uploaded early in 2012 in order to achieve the optimal performance at that time.

*Table 4. Observations that triggered with optimal txings parameters*

| | OBSID | Phase | Run | Run Start Time | ACIS Mode | FI† | | BI† | | Cause of Trigger* |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Thresh | Value | Thresh | Value | |
| 1 | 61996 | acis4 | 102 | 2000-196-11:01:55 | Te3x3 | 644 | – | 43 | **755** | SCS107 |
| 2 | 2344 | acis6 | 41 | 2000-330-07:36:23 | Te3x3 | 615 | **768** | 42 | – | HIRAD |
| 3 | 61839 | acis6 | 45 | 2000-331-19:46:33 | Te3x3 | 614 | **860** | 42 | 39 | EPHIN |
| 4 | 1578 | acis7 | 180 | 2001-093-02:22:53 | Te3x3 | 594 | **940** | 41 | 19 | EPHIN* |
| 5 | 1890 | acis10 | 79 | 2001-267-14:29:13 | Te3x3 | 575 | **863** | 40 | 19 | EPHIN |
| 6 | 2010 | acis10 | 165 | 2001-308-17:38:28 | Te3x3 | 572 | **10953** | 40 | **330** | EPHIN* |
| 7 | 3389 | acis11 | 29 | 2001-327-01:45:35 | Te5x5 | 571 | **726** | 40 | – | EPHIN* |
| 8 | 61227 | acis13 | 40 | 2002-111-04:51:32 | Te3x3 | 566 | **3332** | 40 | **97** | EPHIN* |
| 9 | 4365 | acis14 | 210 | 2002-234-10:48:39 | Te3x3 | 566 | **864** | 40 | 19 | HIRAD |
| 10 | 2783 | acis14 | 217 | 2002-236-02:11:45 | Te5x5 | 566 | – | 40 | **350** | EPHIN* |
| 11 | 5760 | acis31 | 107 | 2005-251-01:58:27 | Te5x5 | 686 | **780** | 45 | – | EPHIN |
| 12 | 58650 | acis38 | 163 | 2006-347-06:45:52 | Te3x3 | 750 | **4599** | 49 | 165 | HIRAD* |
| 13 | 12613 | acis68 | 56 | 2011-216-07:03:56 | Cc3x3 | 645 | **959** | 48 | 38 | EPHIN |
| 14 | 13822 | acis70 | 209 | 2012-023-06:02:14 | Te5x5 | 580 | **881** | 46 | 34 | EPHIN |
| 15 | 13819 | acis70 | 213 | 2012-027-19:42:51 | Te5x5 | 578 | **1222** | 46 | 48 | EPHIN* |

† The values of the rate averages that resulted in a trigger are shown in **boldface** type.

* These runs would have triggered with the default *txings* parameter set (the "Onboard" column of Table 3) that have been active since the patch was first loaded in November 2011.

## 6. Recommendations

The default values of the *txings* parameters onboard as of November 7th 2011 would have resulted in 7 radiation triggers over the past 11 years, with no false alarms. The "optimum" parameter set determined by the process described in the Section 5 would have caught a further 8 events, again without any false triggers. It uses the `FIparm` and `BIparm` coefficients described in Section 3 (see the Appendix for their values), and the onboard `FI_threshold` and `BI_threshold` values must therefore be updated from time to time. Note that the zeroth-order coefficients `FIparm[0]` and `BIparm[0]` are not used by *txings_test*, but are reliable indicators of the "optimal" choice for `FI_threshold` and `BI_threshold`, *i.e.*, for a period in which the average rates are predicted to be `FIavg` and `BIavg`, the best choice for `TXblock` parameters will be

```
FI_threshold = FIavg + FIopt – FIparm[0] = 440 + 700 – 514 = 626
BI_threshold = BIavg + BIopt – BIparm[0] =  26 +  45 –  24 =  47
```

The numerical values of `FIavg` and `BIavg` are estimated by extrapolating the data in Fig.1, `FIopt` and `BIopt` are from the "Optimum" column of Table 3., and `FIparm[0]` and `BIparm[0]` are from the polyomial fits and are listed in the Appendix. This is how the values in the "Upload" column of Table 3 were determined.

From the 11-year trend in Fig. 2, I estimate that the threshold rates will currently (February 2012) need to be updated at 4–6 month intervals, guided by a recalculation of `FIparm` and `BIparm` and a weighted average of recent and 11-year-old rates. The "Upload" parameters in Table 3 should provide optimal *txings* performance for the period February–July 2012, since we expect the rates to be slowly varying over this part of the solar cycle. The command packet definition that copies these values to `TXinit` is shown in Fig. 3. Its name in the ACIS tables is `WBTXING002`. The `TXinit` address coded into this command is only valid for the current patch level (Standard E, Optional F, software version 50).

*Figure 3. Command to update TXinit (bcmd format)*

```
  write 0 0x8003dc30 { # update the BEP's TXinit structure
        3              # MINUTES: number of averaging 64-sec integration intervals
        5              # TRIGGER_COUNT: number of successive intervals to trigger
        512            # MAX_TX_PER_ROW: maximum threshold crossings per row
        291840         # CCD_TICKS: FEP clock ticks per 1024 rows in CC mode
        626 47         # RATE_LIMIT[]: FI,BI threshold rates (per row, per 100 secs)
        1 1            # TX_INCR[]: minimum FI,BI threshold rate increments
  }
```

## 7. References

"ACIS Software User's Guide," MIT 36-54003, Rev. A, (NAS8-37716/DR/SDM05) July 21, 1999.

"Using ACIS on the Chandra X-ray Observatory as a particle radiation monitor," C. E. Grant, B. LaMarr, M. W. Bautz and S. L. O'Dell, SPIE, June 2010.

"Using ACIS to detect and report high radiation conditions," MIT 36-1044, v. 1.3, April 15, 2011.

"Using the Chandra ACIS X-Ray Imager as a Background Particle Flux Detector," P. G. Ford & C. E. Grant, submitted to SPIE, January 2012.

## 8. Glossary

Back-Illuminated ............... A CCD that detects x-rays incident on the face opposite to that of its junctions.
BEP ..................................... ACIS Back End Processor — the unit that interfaces between RCTU and FEPs.
BI ........................................ An abbreviation for Back-Illuminated. (*q.v.*)
Bi-Level .............................. A data channel from DPA to RCTU reporting only OFF or ON.
CC ....................................... An abbreviation for Continuous Clocking (*q.v.*)
CCD .................................... Charge-Coupled Device — the x-ray detectors used by ACIS.
DPA .................................... Digital Processor Assembly (the digital part of ACIS, i.e., BEP and FEPs).
EPHIN ............................... Electron, Proton and Helium Instrument — flown on Chandra and SOHO.
FEP ...................................... ACIS Front-End Processor (one of 6 units that identify x-ray candidates).
FI ......................................... An abbreviation for Front-Illuminated. (*q.v.*)
Front-Illuminated ............. A CCD that detects x-rays incident on the same face as its junctions.
OBC .................................... On-Board Computer — the Chandra spacecraft's central controller.
OBSID ............................... Observation Identifier — a unique number assigned by the CXC to a science run.
RADMON ......................... Excessive radiation alert signal from EPHIN or ground to OBC and/or ACIS.
RCTU.................................. Remote Communications Telemetry Unit (between ACIS and the OBC, etc.).
SCS107 ............................... Command to OBC from EPHIN or ground to safe the science instruments.
TE ....................................... An abbreviation for Timed Exposure (*q.v.*)
Threshold ........................... Value by which a pixel exceeds its corresponding bias value to become interesting.
TX ....................................... The structure used to describe a set of *txings* parameters.
TXblock ............................. The parameters used by the *txings* patch during the current science run.
TXinit ................................. The default set of TX parameters copied to TXnext at the start of a science run.
TXnext ............................... The set of TX parameters copied to TX at the start of a science run.
writeBep ............................. External command to ACIS to update specified contents of BEP memory.

## Appendix

The following data are compiled into *txings_test* via the *txings_parms.h* include file.

```
/* Special observations */

/* OBSIDs in which EPHIN/HRC triggers occurred */
int radmon[] = {
   433, 648, 1578, 1890, 1900, 2010, 2077, 2080, 2547, 2783, 2943, 2986,
   3057, 3389, 3577, 3713, 3764, 3871, 3896, 3997, 4063, 4591, 4642, 4690,
   4757, 4912, 4996, 5060, 5475, 5760, 5814, 5934, 6390, 6432, 6438, 6904,
   7410, 12613, 12902, 13822, 13819, 60117, 60497, 61116, 61227, 61452,
   61665, 61839,
};

/* OBSIDs in which SCS107 was commanded from the ground */
int scs107[] = {
   642, 2042, 2135, 2365, 2751, 2953, 3130, 3182, 3363, 3463, 3732, 4016,
   4175, 4408, 4756, 5296, 5954, 6152, 13393, 14361, 61427, 61996,
};

/* OBSIDs with other high-radiation conditions */
int hirad[] = {
   2344, 4365, 10113, 56866, 56867, 57597, 58650,
};

/* Crab observations */
int crab[] = {
   168, 169, 170, 769, 770, 771, 772, 773, 1994, 1995, 1996, 1997, 1998,
   1999, 2000, 2001, 2798, 4607, 4621, 4622, 4623, 4624, 5284, 5285, 5550,
   5551, 5552, 5553, 5554, 5555, 5556, 5557, 5558, 6140, 6141, 6142, 6143,
   7587, 13139, 13146, 13147, 13150, 13151, 13152, 13153, 13154, 13204,
   13205, 13206, 13207, 13208, 13209, 13210, 13750, 13751, 13752, 13753,
   13754, 13755, 13756, 13757, 13758, 13759, 13760, 13761, 13762,
};

/* Hardware anomalies */
int fep0_latchup[] = {
   18, 333, 510, 965, 1383, 62327, 62333, 62338, 62340, 62353, 62363, 62502,
};

int tplane_latchup[] = {
   371, 2010, 3403, 5560, 6730,
};

/* FI threshold polynomial from acis1:1 thru acis71:74, 02-28-2012 */
double FIparm[] = {
   5.13778, -1.22987, 0.327996, -0.0213403, 0.000158621,
};

/* BI threshold polynomial from acis1:1 thru acis71:74, 02-28-2012 */
double BIparm[] = {
   0.241664, -0.0511798, 0.0134273, -0.000737483, -1.31463e-06,
};
```

*continued overleaf*

```
/* Addresses of patch blocks  */
unsigned *txinit_addr = (unsigned *)0x8003dc30; /* address of TXinit */
unsigned *txnext_addr = (unsigned *)0x8003dc50; /* address of TXnext */
unsigned *txings_addr = (unsigned *)0x8003dc70; /* address of txings */
unsigned txversion    = 50;                     /* patch version number */
char     *txloadblock = "WBTXING002";           /* name of pblock */

/* "Optimum" txings parameters */
unsigned txopts[8] = { 3, 5, 512, 291840, 626, 47, 1, 1 };
```