MIT CSR
ACIS FLIGHT SOFTWARE

Massachusetts Institute of Technology
Center for Space Research
Cambridge, Massachusetts 02139

To: Claude Canizares

From: Peter Ford

Date: May 7, 1996

Subject: Long-Term ACIS Maintenance and Verification

Introduction

By any standards, ACIS is a complex instrument whose long-term maintenance will pose many problems. It is clear that this task, which is the responsibility of the AXAF Science Center, will require some sort of instrument simulator and personnel trained in its use. I start by addressing the requirements that will be expected of such an installation, proceed to describe the necessary hardware, software, and documentation, and conclude by describing how it might be managed.

I began this memo after reading the March 29 draft of Dan Schwartz' ACIS Operations Management Plan, and have tried to remain compatible with his concepts. I have also had detailed discussions with Bill Mayer, Bob Goeke, and Jim Francis, but the recommendations and the errors are all mine.

Tasks

I separate the ACIS maintenance tasks into the following broad categories:

- *Verify the correct functioning of routine command loads*—if we discover that the ACIS flight hardware is particularly sensitive to command parameters, *e.g.* to clocking biases or to sequencer commands, it will be advisable to run new configurations on a hardware simulator before uplinking them.
- *Participate in Orbit Verification Activities*—during that critical period, as many people as possible with knowledge of ACIS flight software should be on hand to recognize anomalies in the test results and to work to respond to them.
- *Investigate hardware anomalies*—a tricky business, but one made much easier with a hardware simulator. Suspect an under-voltage on a particular board? Re-create the condition in the simulator and figure out a work-around.
- *Simulate and study the effects of hardware ageing*—as flight components begin to grow old and die, we would observe the consequences on a simulator and develop ways to recover. In reverse, if any of our simulator components fail, we would be warned that the same fate could befall the flight unit and we could try to prevent this from happening.
- *Contribute to a spacecraft simulation*—if the ASC intends to build a spacecraft simulator, we could, in principle, supply hardware and/or software to simulate some parts of ACIS. If the simulation is to be in software, we would need to develop the necessary interfaces; if in hardware, I could imagine using a hardware simulator at MIT to talk through an RCTU to a spacecraft simulator's CTU at SAO.

- *Locate and patch errors in existing flight software*—if the errors occur within high-level functions, *e.g.* in science management or in the instrument's response to abnormal conditions, suitable patches can be developed using a software simulator. For a final verification, or if the error involves low-level hardware interfaces, a hardware simulator would be more useful.
- *Develop software patches to increase the capabilities of the instrument*—the flight software has been designed to respond to various anticipated anomalies, *e.g.* per-output-node changes in DC signal level, random changes in pixel bias levels, SEUs in bias memory, radiation monitor alarms, etc. We have not included code to respond to less likely conditions, *e.g.* time-varying light leaks, excessive pixel smearing, etc., but this could easily be developed and verified on a simulator.

There is considerable overlap between these tasks, but they clearly require both hardware and software components. Another important distinction lies between routine tasks, tasks conducted in response to some perceived anomaly, and tasks intended to improve the instrument beyond its initial flight configuration.

Routine Maintenance

During normal flight operations, ORs will be created by the ASC, compiled into ACIS parameter blocks, and passed to OFLS for scheduling. No special maintenance should be required. The resulting ACIS telemetry will be processed into event files and derived products. Anomalies will be noted by the Science Operations Team (see below), which will be responsible for preparing and uplinking changes to various ACIS parameter blocks, as shown in Table 1.

Table 1. Updatable Parameter Blocks

Type of Block	Anticipated Update Frequency	Reason for Update	Source of Update Information
Exposure Parameter	Hours, Days	Science Observation	OR
1-D and 2-D Windows	Hours, Days	Science Observation	OR
DEA Configuration	Weeks	DEA ageing, CCD radiation damage	Overclock, bias map and corner pixel monitoring
Bad Columns and Pixels	Weeks	CCD radiation damage	Event monitoring
Fiducial Pixels[a]	Weeks	CCD testing	Bias map and event monitoring
Custom SRAM and PRAM	Unknown	Science Observation, response to anomaly, etc.	OR, bias map, event monitoring, etc.
Huffman Compression	Years	CCD radiation damage	Bias map monitoring

a. If implemented

The great majority of these tasks will be performed on a routine basis by the SOT without the need for hardware or software simulators.[1] The major exception is in the area of customized Sequencer Instructions (SRAM) and Program Instructions (PRAM) that may be required to achieve the designed ACIS performance in various configurations. Present indications are that it is unlikely that the BEP's PRAM synthesizer will be capable of generating optimal DEA instructions in for possible values of exposure parameters—if it is necessary to perform such a science observation, it may be necessary to hand-craft the PRAM (and perhaps SRAM) loads on the ground

1 although the SOT might choose to use one, if available.

and uplink them to ACIS. Since incorrect commands may overload the DEA circuitry, it would be essential to verify the command loads within a simulator before uplinking them, and a software simulator alone won't suffice since the quality of a PRAM load cannot be judged fully until it is used to simultaneously drive all CCDs to determine whether their clocking signals are interfering with each other.

If discrete PRAM loads are permitted (and they may be necessary for some types of observation), the SOT must develop the expertise to design and test these loads. This calls for the closest collaboration between the SOT and the people who will be most experienced in designing clocking sequences for ACIS CCDs, who will most probably be found within the CSR CCD laboratory working on new sensors.

Response to Anomalies

Since so much of the success of AXAF depends on the continuous and reliable functioning of ACIS, the search for anomalies must be relentless and the response to finding them must be swift and decisive. However the anomaly is first detected, experience in past missions has shown that hardware simulation is the fastest and most comprehensive way to understand the problems and to determine their optimal solutions.

A described more fully below, the ACIS hardware simulator would use components that were very similar to their flight counterparts. If simulator components fail, we might anticipate trouble with the flight units too, and develop work-around, e.g., an ageing power supply might be preserved by running at a lesser load, using fewer CCDs. Once assembled, the hardware simulator should not be switched off!

During an ACIS emergency, there will be heavy demands on the hardware simulator and on the staff who are trained in its use. We can prepare for this in several ways:

- assign a small room at CSR to the simulator and its GSE and make someone responsible for seeing that it stays there except when taking part in experiments in the CCD lab.
- maintain all documentation related to the instrument and its GSE in both printed and machine-readable format.
- provide access to each printed circuit board during normal operation, *e.g.*, construct a set of extender cards that would permit us to insert probes and logic analyzers at all possible test points.
- create a reliable network interface to the EGSE so that simulator tests can be prepared and analyzed by the SOT.
- keep key members of the original ACIS design team aware of the progress of the flight experiment so that they are better prepared if called back for emergency consultations.

Software simulation will also have its use during anomalies, both to prepare diagnostic patches to run in the flight instrument and in the hardware simulator, and later to circumvent the anomaly itself. Although generally less useful than hardware simulation, multiple software simulations may be run concurrently.

Instrument Improvements

As our ACIS experience accumulates, we may wish to add new functions to the instrument, or improve existing functions. While prudence and thrift argue against actually doing about this, a software patch can sometimes be justified if a strong enough case can be made, *e.g.* in significantly reduced mission operations costs.

The ACIS science team has already identified a number of instrument modes that have not been included in the CEI specification, but which would be highly beneficial, *viz*:

- a "graded histogram mode" in which the BEP generates histograms of event energies;
- the ability to report the values of "fiducial pixels" from specific locations in the FEP input buffers;
- an event-finding mode that returns 5x5 blocks of pixels surrounding possible events, instead of the normal 3x3 blocks.

Any software change introduced after SIM integration would of necessity be in the form of a patch to be uplinked through the OBC, so it is unlikely that entire new modes such as these would be implemented. I think that the improvements are more likely to be incremental—small changes to existing functions to improve their throughput or scientific accuracy.

Software Simulator

The current flight software simulator runs on a MIPS R3000 or R4000 processor under the Ultrix operating system in a Decstation. The native architectures of these CPUs are supersets of the Mongoose processors used in the ACIS flight units, so no emulation is required. The simulator is actually a set of UNIX processes—one for each FEP and one for the BEP. The latter uses POSIX

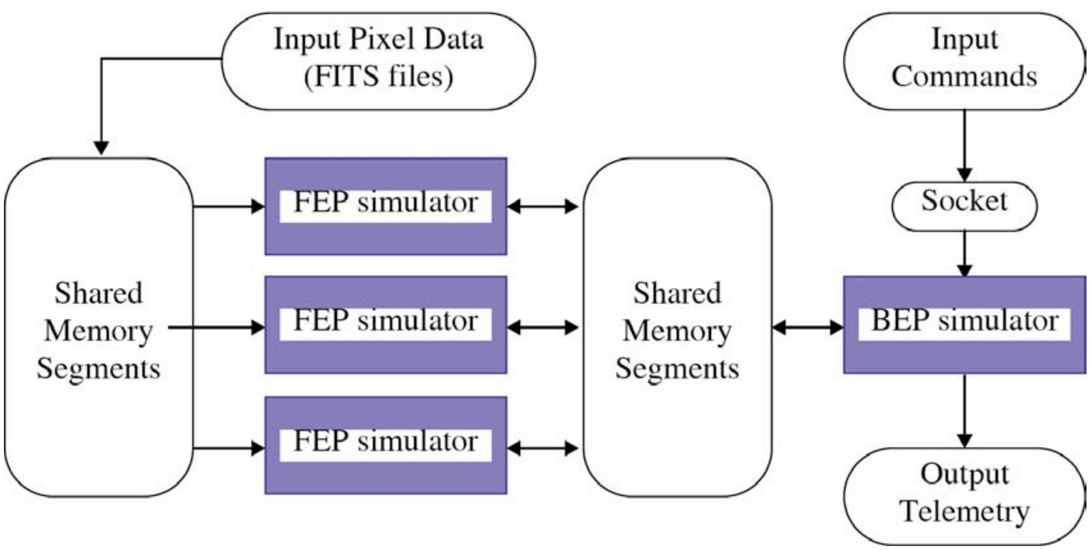

Figure 1. ACIS Software Simulator

threads to implement the multiprocessing aspects of the Mongoose RTX operating system. Commands are sent to the BEP via INET sockets; communication with the FEPs is through shared memory, and the BEP process writes the telemetry to its standard output.

Since the simulator processes must run in a single machine, its CPU should be the fastest available. Current Decstations using the R4000 processor are able to simulate a single FEP in one third real time, which is adequate for most purposes, but the ASC should consider upgrading them if faster Mongoose-compatible Decstations become available.

The throughput of the simulator is increased by off-loading the input and output tasks onto a second workstation, *e.g.* Sun SPARC, which is responsible for loading commands and data, and interpreting the telemetry. This workstation would also be responsible for controlling the flight software configuration and for providing an interface to the rest of the Science Center.

Hardware Simulator

Since the ACIS engineering units will be almost identical to the flight units, they can be assembled into a device that closely simulates the flight hardware. Table 2 lists the hardware components that should be available at MIT after the flight units is shipped, and the minimum number required to construct a useful simulator.

Table 2. ACIS Flight and Engineering Hardware

Unit Type	Flight Units	Engineering Units	
		Total	Minimum
Analog (DEA) boards	10	# 6[a]	2
Analog Control Boards	2	2	1
DEA backplane and housing	1	1	1
Front End Processor (FEP) boards	6	# 6	2
Back End Processor (BEP) boards	2	3	1[b]
DPA backplane and housing	1	1	1
Focal Plane Assembly (FPA)	1	1	*[c]
Power Supply (PSMC)	1	1	1

a. plus calibration boards that can be retro-fitted with flight firmware
b. two BEPs are needed for testing redundancy; otherwise, one only
c. may be replaced by image loader and/or individual CCDs

Sufficient engineering units would be assembled into a single instrument, which would be supported by all or part of the ACIS GSE. If the FPA were integrated with the other components, the entire Vacuum GSE would be employed to maintain it. Alternatively, the FPA would be replaced by a set of cables connecting the DEA units to separate CCD installations, such as are currently used for ACIS calibration. In either case, the ACIS electrical GSE would be inherited and the assembly would be made sufficiently portable to permit it to be connected to laboratory vacuum systems. A sturdy trolley should suffice. I think that the most suitable place to keep it would be close to the CCD laboratory on the 5th floor of Building 37.

The hardware simulator would be interfaced to the following devices:

- 1 Remote Control Telemetry Unit (RCTU)
- 4 Digital Image Loaders
- 2 High-Speed Tap frame grabbers

It will be important to retain the RCTU that was supplied to MIT/ACIS for testing the instrument, and, to avoid later misunderstandings, the ASC should negotiate this with MSFC and TRW as soon as possible. The other devices form part of the existing ACIS EGSE. It would also be useful to construct a number of board extenders that would give access to individual DEA and DPA boards while the instrument is operating. Since the engineering boards will not be conformally coated, it would be a simple matter to probe any part of the circuitry.

Documentation

The ACIS project has created an extensive library of engineering documents, drawings, some of which are also available electronically, others not. I suggest that the entire collection be transferred to the ASC and housed close to the hardware simulator and its GSE and to the workstations belonging to the software simulator. A large flatbed plotter should be available to print additional drawings should the need arise.

Personnel

The ACIS project plans to lay off much of the engineering team even before the instrument leaves MIT. We must make very sure that we retain sufficient people to construct the simulators and then either remain at MIT to run them or to train others to do so. The tasks that I have outlined above call for at least four separate people:

- 50% of a staff programmer to remain with ASC for the duration of the mission to exercise the ACIS software and hardware simulators, prepare simple flight-software patches, etc.
- 25% of a scientist or engineer with intimate knowledge of the flight software who would be available to work with the programmer and with other AXAF teams to monitor the health of the instrument, develop more sophisticated patches to the flight software, and consult to the SOT on non-standard command loads, etc.
- 15% of a technician to perform routine maintenance on the ACIS hardware simulator.
- 15% of an engineer with intimate knowledge of the flight hardware who would inspect ACIS housekeeping data on a regular basis and perform diagnostic tests on the hardware simulator.

In addition these positions, several members of the ACIS project will be available for the next 12-24 months, in particular Jim Francis to complete the software documentation and make any necessary post-XRCF software changes, Steve Kissel for issues relating to CCD clocking and analog electronics, and Dorothy Gordon to consult on digital hardware problems.

Transition

The first step is to disseminate this memo, gauge the reaction, and, hopefully, develop it into a long-range plan for ACIS maintenance within the ASC. Then we should identify those resources, people and things, that we're going to lose unless we speak up very soon. In particular, if we wish to retain any of our software engineers, we must decide to do so within the next month or two.

On the hardware side, the ASC should announce its intention of retaining all ACIS engineering and calibration hardware, including the spare Lockheed-Martin's power supply, all GSE equipment, and the RCTU and CTUE supplied by TRW. The engineering units should be assembled into a hardware simulator as soon as possible, both to take advantage of the engineers who are still working for the project, and because the simulator should be run for as long as possible in order to let whatever is going to wear out, wear out. The ASC should also plan to acquire one or two fast R4000 Decstations to use for software simulation.

From a management viewpoint, I recommend that the ASC take steps to co-opt ACIS personnel to train the SOT in the finer points of the instrument operations. This should best be done while we write the "Software Maintenance Manual", so that this document will be of the most assistance in the years to come.