

NICMOS Memory/Hardware Anomalies

A.B. Schultz, I. Dashevsky, and C. Townsley
January 31, 2003

ABSTRACT

NICMOS has experienced Error Detection and Correction (EDAC) single-bit EDAC errors, TPG resets, and Intel Exception problems. Except for the last error, the EDAC errors have been specific to the same memory address location. They are handled by software. A Timing Pattern Generator (TPG) reset occurs when a spurious signal is generated caused by a high energy particle event. The NICMOS flight software (FSW) was modified to handle TPG resets. An Intel Exception will suspend the instrument. There is all likelihood that additional similar EDAC errors, TPG resets, and suspends due to an Intel Exception problem will occur during the on-orbit life time of NICMOS. This ISR is intended to document the hardware errors and to be a resource when additional errors occur.

Introduction

The Near Infrared Camera and Multi-Object Spectrometer (NICMOS) was installed on the *Hubble Space Telescope* (HST) in February 1997 during the Second HST Servicing Mission (SM2). Due to a thermal leak, the Nitrogen ice cryogen was exhausted on January 4, 1999, and data taking was suspended on January 11, 1999. NICMOS was reactivated following installation of the NICMOS Cooling System (NCS) during HST Servicing Mission 3B on March 8, 2002.

The NICMOS flight electronics was built and tested by Ball Aerospace, Boulder, Co. The flight electronics consist of four electronic boards housed in the Main Electronics Box (MEB); i.e. a CPU board, a System memory board, a Buffer memory board, and a Communications board (serial interfaces to non-MEB electronics). The CPU board contains a

16 MHz Intel 80386-based processor, an Intel 80387 math coprocessor, an Intel 82380 DMA controller, 2 Mbytes of RAM, and 0.5 Mbytes of error correction memory. NICMOS memory is mapped into Error Detection and Correction (EDAC) Random Access Memory (RAM) (2 Mbytes), Control Section (CS) memory, Buffer memory (16 Mbytes), Memory Mapped I/O, EEPROM (1.5 Mbytes), Debug & Real-Time Performance (RTP) memory, and PROM (32 Kbytes).

NICMOS Flight Software and Commanding

The NICMOS Flight Software (FSW) is installed in the NICMOS CS Flight Hardware as a compiled, executable image. It is embedded software written in the C language and has two states: Boot and Operate. The Boot state starts executing when the CS is powered on and provides some of the basic functionality with no detector or mechanism operations. It does not need a kernel or an operating system to regulate the execution of tasks. The Boot code is not modifiable (contained in PROM). The Operate code performs all the functions necessary to control the NICMOS instrument. The Operate code is modifiable (contained in EEPROM). The C-code source files are maintained at the NASA Goddard Space Flight Center (GSFC) by the HST Payload Flight Software Team (Code 441). The most recent update to the NICMOS FSW, Version 4.0Cb, was installed on October 4, 2002.

The C-code requires function calls as well as the corresponding parameters and data to command the science instruments. This task is fulfilled by the science commanding, which is written and maintained at the STScI by the Flight Systems Engineering branch using the Science Command Programming Language (SCPL). The output of the STScI ground system (i.e., proposal processing, associated scheduling, etc.), including time sequenced science commanding, is translated from ASCII to a memory load consisting of bits, and transmitted to the HST on a weekly schedule using the Tracking and Data Relay System (TDRS) satellites. The science instrument commanding is modifiable on an ongoing basis.

EDAC error

The Error Detection and Correction (EDAC) refresh software is a background task that cycles through EDAC memory and reads all of the memory locations approximately once every 4-10 seconds depending on CPU usage. The purpose of EDAC is to detect and allow correction of single-bit memory errors. EDAC can not handle a multi-bit error and a multi-bit error will suspend the instrument. When an error is detected the EDAC refresh task will re-read the EDAC memory address where the error occurred. The hardware reports back the corrected bit pattern, and the software writes the corrected value back into EDAC memory, effectively refreshing the memory location. A single-bit error could be caused by a radiation hit, or possibly, due to a memory location that has failed.

Overview

EDAC error correction is performed by a combination of hardware and software. Anytime a write to memory is performed the 32-bit CMOS error detection and correction hardware computes a 7-bit modified Hamming code and writes it to a 7-bit word which is associated with every 32-bit memory address (i.e. every memory location consist of a 32-bit address and a 7-bit check code). An EDAC memory error detection occurs when a memory location is read and the recomputed check bits do not match the 7-bit check code.

The NICMOS EDAC single-bit error counter (NSBITERR) is incremented by 1 each time a single-bit error is detected, and the number of single-bit errors detected is included in the normal engineering telemetry. No status buffer message is generated for this error. The counter flag is reset from 1 to 0 on entry to boot mode and at entry to operate mode.

NICMOS has experienced EDAC single-bit errors and one random multi-bit error. All of the 1998 single-bit errors have occurred in the location 8405DE5C while the 2002 errors occurred in memory location 84083E5C, where “84” refers to the problem bits and “05DE5C” and “083E5C” refers to the memory address. Each occurrence has been logged and the logs are available on the GSFC HST Payload FSW web page. The last event occurred on October 22, 2002 (day 2002.295) at 09:55:57z. The web page URLs are:

1998:

http://hstplfsw.hst.nasa.gov/payload1/NICMOS/docs/NicmosDoc/nic_singlebit_evts.pdf

2002:

http://hstplfsw.hst.nasa.gov/payload1/NICMOS/docs/NicmosDoc/nicsbevts_postsm3b.pdf

The section of code that was getting the single-bit error was moved to a different location in memory with the installation of NICMOS FSW 4.0C (June 2000). At that time, an extra check was added to the EDAC refresh task and implemented to recheck the memory location getting a single-bit error to verify the error. If the error was still in the same location, then an error message is issued to the NICMOS internal log. The reappearance of the single-bit error in the same section of code, but at a different memory location, indicates the problem is in the code and not in hardware.

TPG resets

A Timing Pattern Generator (TPG) reset occurs when a spurious signal is generated caused by a high energy particle event. The TPG reset sets all the digital-to-analog converter (DAC) voltages to zero which effectively powers off the detector. This was traced to a component (Opto-Isolator) on the TPG board. An opto-isolator is an electronic component used to reduce electronic noise in a signal path by electrically isolating one part of a circuit from another. This isolation is accomplished by converting electrical pulses to opti-

cal pulses and back. An energetic charged particle event can trigger an optical pulse which results in a TPG reset. In the past, a TPG reset would suspend the instrument.

The NICMOS flight software (FSW) was modified to handle TPG resets. The new FSW detects a TPG reset, sets a flag, and allows the observations in the other two NICMOS cameras to continue. Stored command instructions will conditionally recover the reset TPG during normal instrument mode transitions. If a TPG reset occurs before the start of an observation, the TPG timing pattern is loaded nominally by the FSW and data are read from the analog-to-digital converter (A/D). However, the detector will be powered off so the data collected is noise. NICMOS data sets that are affected by a TPG reset will be flagged in OPUS: the NICMOS keyword EXPFLAG is set to read DETOFF any time a TPG reset occurs during an exposure.

The NICMOS 4.0C FSW release was installed on June 27, 2000 and activation was completed at 19:28 GMT. Each TPG reset occurrence has been logged and the log is available on the GSFC HST Payload FSW web page. The last event occurred on October 15, 2002 (day 2002.288) at 15:51:32z. The web page URL is:

<http://edocs1.hst.nasa.gov/doc/publish/mop/NICMOSReset.htm>

Intel Exception

Processor interrupts and exceptions are control transfers which alter the normal program flow to handle external events or to report errors or exceptional conditions. The difference between interrupts and exceptions is that interrupts are used to handle asynchronous events external to the processor, but exceptions handle conditions detected by the processor itself in the course of executing instructions.

When the processor detects an exception, the exception handler will determine if the exception is a fault, trap, or an abort.

- Faults are exceptions that are reported “before” the instruction causing the exception. Faults are either detected before the instruction begins to execute, or during execution of the instruction. If detected during the instruction, the fault is reported with the machine restored to a state that permits the instruction to be restarted.
- A trap is an exception that is reported at the instruction boundary immediately after the instruction in which the exception was detected.
- An abort is an exception that permits neither precise location of the instruction causing the exception nor restart of the program that caused the exception. Aborts are used to report severe errors, such as hardware errors and inconsistent or illegal values in system tables.

The occurrence of interrupts is not generally predictable, and normally, when the processor detects an exception, the 80386 invokes a handling procedure to process the exception. The processor services interrupts and exceptions only between the end of one

instruction and the beginning of the next. If the handling procedure can not handle the interrupt or exception, a shutdown “suspend” occurs. A manual restart of the processor is necessary at this point. This is a known bug of the NICMOS FSW.

Intel Exception Events

On September 27, 1997 (day 1997.270) at 11:06z the NICMOS FSW suspended due to an Intel Exception “debug error.” NICMOS recovery was on day 1997.271 at 04:18z per Operation Request.

On January 17, 1998 (day 1998.017) at 01:10z the NICMOS FSW suspended due to an Intel Exception “debug error” (HSTAR 6316). NICMOS recovery was on day 1998.017 at 07:01z per Operation Request.

On April 7, 1998 (day 1998.097) at 11:03z the NICMOS FSW suspended because it encountered an Intel Exception (HSTAR 6429). A three detector MULTIACCUM observation was in progress at the time which was also the case during the two previous suspends. NICMOS was recovered on day 1998.097 at 16:40z per Operation Request.

On May 3, 2001 (day 2001.123) at 07:19z the NICMOS FSW suspended due to an Intel Exception, a “page fault” (HSTAR 8201). NICMOS was in SAAOPER when the event occurred. NICMOS was recovered on day 2001.127 by real-time commanding, transitioned from suspend to SAAOper.

On April 21, 2002 (day 2002.111) at 17:05z the NICMOS FSW suspended due to an Intel Exception “debug error” (HSTAR 8623). NICMOS was returned to SAAOPER and then to OPERATE by stored commanding on day 2002.112 at 03:10z.

On June 19, 2002 (day 2002.170) at 09:35z NICMOS FSW suspended due to an Intel Exception while in boot mode.

On July 23, 2002 (day 2002.204) at 03:42z NICMOS FSW suspended due to an Intel Exception “debug error” (HSTAR 8744).

On October 16, 2002 (day 2002.289) NICMOS FSW suspended due to an Intel Exception “debug error” (HSTAR 8816). NICMOS was recovered on day 2002.289 at 15:51:50z.

HSTARs are HST (Operational) Anomaly Reports and are the responsibility of the Flight Operation Team (FOT) and HST GSFC personnel.

Conclusions and Recommendations

The three anomalies discussed above, EDAC single bit error, TPG reset, and Intel Exception, will occur in the future. An EDAC single bit error is handled by a combination of hardware and software, whereas TPG resets are only handled by software, neither are a major concern to the operation of NICMOS. Any data loss due to a TPG reset should be tracked and the PI informed about the lost of data. An Intel Exception event does suspend NICMOS data taking requiring real time commanding to re-boot the instrument. The obvi-

ous recommendation is to relocate the affected code into a different memory location and restrict the use of the affected memory locations. These events are spurious interrupts and there are no obvious clear recommendations that can be implemented to avoid them. The one clear recommendation is for STScI Commanding and Engineering to monitor their occurrences for any possible long term solutions or recommendations to avoid them in future instruments.

Acknowledgements

We would like to thank STScI Commanding and Engineering for reviewing the draft of this ISR and their assistance in clarifying each of the anomalies discussed in this ISR.