A General Purpose Telemetry Monitor (GPTM) for the Hubble Space Telescope (HST)

Repurposing an Existing Design

C. P. Hoffman, F. H. Schiffer 3rd
GSFC Code 441 HST NSSC-1 Flight Software Team
Introduction

- Hubble Space Telescope (HST) operating for 21 years with 24x7 ground support
- A proven design was modified to provide an autonomous telemetry monitor
- There are clear benefits to design reuse
- There are also challenges and pitfalls to avoid

First, some background
Background

• HST is a mix of old and new technology
  – New Science Instruments are state of the art
  – The NASA Standard Spacecraft Computer (NSSC-1) designed in mid-1970s, first flown in 1980
    • NSSC-1 is part of the Science Instrument Command & Data Handling (SI C&DH) subsystem
    • On-orbit SI C&DH replaced with “new” 20 year old SI C&DH during Servicing Mission 4 (May 2009)
    • Programmed in NSSC-1 Assembly Language (18-bit octal)
      – Many executive programs inherited with little change from earlier NSSC-1 based spacecraft, most notably Solar Maximum Mission (SMM)
Automation Problem

• Transition from fully manned to autonomous lights out operations
  • HST was continuously monitored from the ground for 21 years
  • Flight operations staffing reduced to 8x5 support on June 13, 2011
    – Planning began before the last servicing mission
    – Need for an onboard autonomous anomaly response was recognized
      » Minimize time between anomaly and response
      » Avoid Flight Operations Team involvement in routine “anomalies”
      » Reduce dependence on downlinked telemetry for anomaly response
Requirement

• HST Automation systems shall provide the ability to respond to specific spacecraft anomalies
Existing Monitoring Application Processors (APs)

MFMONTLM – Executive Telemetry Monitor Processor
- Limit checks up to 22 telemetry items
  - Posts a status message when limits are exceeded 5 times in a row
  - Safes Payload on loss of 1 Megahertz Clock
- Actions are hardcoded
- Runs every 60 seconds

MFGPEF – General Purpose Event Flag Processor
- Monitors up to 16 NSSC-1 addresses
  - Sets or clears one of 90 event flags when an action is required
  - Event Flags may be used to control a sequence of stored commanding
- Actions are controlled by a programmable table
- Runs every 0.5 second
Approaches with No Onboard Change

• MFMONTLM – ground system takes autonomous actions based on status messages
  – Status messages potentially transient and not visible due to telemetry limitations
• MFGPEF – monitor event flags in command sequence
  – One continuously running sequence per command processor
  – Independent actions tightly coupled by single command sequence
Approach with Onboard Change

- MFGPEF – Add optional ability to activate a command sequence
  - Make setting an Event Flag optional
  - Increase number of programmable slots to 32
  - The existing functionality would make this update fairly simple
Requirements Creep

- Original plan to update General Purpose Events Flag processor with ability to activate stored Relative Time Command Sequences (RTCS)s deemed insufficient
  - Decision was made to add MFGPTM as a new AP in addition to MFGPEF and MFMONTLM
  - Allow multiple actions per slot
Structural Changes for GPTM

• Event Flag Table cloned and modified
  – From 16 slots of 10 words, to 32 slots of 10 words
  – Spare fields used by Telemetry Monitor Table
    • 6 new fields added to existing 12 fields
      – New actions (Safing, RTCS, ESB, ESR), 1st Minor Frame, Track Mode
    • Modified meaning of two existing fields
      – Event flag field can disable Event Flag actions (0 now a legal value)
      – Science Instrument number expanded for use by Safing and RTCS

• Logic of Event Flag Processor modularized for Telemetry Monitor Processor
  – Divided into 5 subroutines

• Basic Design added to, but not changed
## Event Flags Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ</td>
<td>Frequency to monitor Location (0 – 8191) 0.5 sec. incr.</td>
</tr>
<tr>
<td>FREQCNTR</td>
<td>Internal Frequency Counter from 0 to FREQ</td>
</tr>
<tr>
<td>CONSCNTR</td>
<td>Consecutive Counter from 0 to MINCOUNT, -1 to initialize slot</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Address to monitor (0 – 65535)</td>
</tr>
<tr>
<td>MASK</td>
<td>Bit Mask ANDed with LOCATION for bits to monitor</td>
</tr>
<tr>
<td>HILIM</td>
<td>Higher Limit for Limit Check</td>
</tr>
<tr>
<td>LOLIM</td>
<td>Lower Limit for Limit Check</td>
</tr>
<tr>
<td>EVNTFLAG</td>
<td>Event Flag number (1 – 15, 0 is illegal value)</td>
</tr>
<tr>
<td>SI</td>
<td>Science Instrument number (1 - 5, or 0 for Global Flag)</td>
</tr>
<tr>
<td>CHKTYPE</td>
<td>Limit Check Type (0 = In Limits, 1 = Out of Limits)</td>
</tr>
<tr>
<td>ACTTYPE</td>
<td>Action Type (0 = Latching, 1 = Tracking)</td>
</tr>
<tr>
<td>MINCOUNT</td>
<td>Minimum Consecutive times limit criteria must be met before taking action (1 - 4095)</td>
</tr>
</tbody>
</table>
New Fields for GPTM Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAFING</td>
<td>Action to Safe (0, 1). Also disables GPTM slot</td>
</tr>
<tr>
<td>RTCS</td>
<td>Action to activate specified RTCS (1-144, 0 = no RTCS)</td>
</tr>
<tr>
<td>ESB</td>
<td>Action to post Executive Status Buffer Message (0, 1)</td>
</tr>
<tr>
<td>ESR</td>
<td>Action to set GPTM Executive Status Report flag (0, 1)</td>
</tr>
<tr>
<td>TRKMODE</td>
<td>0= Continuous when action set, 1= on Transitions only</td>
</tr>
<tr>
<td>FSTMNFRM</td>
<td>First Minor Frame to sample location</td>
</tr>
<tr>
<td>EVNTFLAG</td>
<td>Event Flag number (1 – 15, 0 = no Event Flag)</td>
</tr>
<tr>
<td>SI</td>
<td>Science Instrument number (1 - 5, or 0 for no SI, 6 or 7 for Global Flag, 6 for ASCS Safing or RTCS, 7 for System RTCS or Payload Safing)</td>
</tr>
</tbody>
</table>

Existing fields modified:

| EVNTFLAG    | Event Flag number (1 – 15, 0 = no Event Flag)                              |
| SI          | Science Instrument number (1 - 5, or 0 for no SI, 6 or 7 for Global Flag, 6 for ASCS Safing or RTCS, 7 for System RTCS or Payload Safing) |
# GPTM Data Structure

<table>
<thead>
<tr>
<th>bits</th>
<th>word 1</th>
<th>word 2</th>
<th>word 3</th>
<th>word 4</th>
<th>word 5</th>
<th>word 6</th>
<th>word 7</th>
<th>word 8</th>
<th>word 9</th>
<th>word 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>SPARE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>FREQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SPARE**

**FREQ**

**FREQCNTR** (internal counter)

**CONSCNTR** (internal counter)

**SPARE**

**LOCATION**

**MASK**

**HILIM**

**LOLIM**

**RTCS**

**B**

**R**

**SI**

**EVNTFLAG**

**SPARE**

**FSTMNFRM**

**A**

**C**

**MINCOUNT**

**S** - SAFING

**B** - ESB

**R** - ESR

**T** - TRKMODE

**A** - ACTTYPE

**C** - CHKTYPE

*Red = new fields for GPTM*

This presentation contains no US Export Controlled (ITAR) information.

10/19-21/2011
Advantages of Design Reuse

• Quicker transition into development
  – Much of the desired functionality was already present

• Reuse of existing tests

• Reuse of Operational procedures and data structures

• Existing knowledge base of real world usage
Disadvantages of Design Reuse

- Limited requirements analysis of existing functions
- Old test scripts required extensive reworking before they would run in current test environment
- Schedule assumed only originally agreed-upon modifications would be added
Requirements Creep During Design Reviews

- Derived requirements were added to design
  - Need to stagger activation of RTCSs
    - Reduce processing load
    - Avoid nesting of one RTCS by a new RTCS
  - All slots must finish processing before next Master Timing Pulse (MTP, 500 millisecond cycle)
Requirements Creep During Design Reviews

• Other requirements were slipped in as minor modifications
  – Error checking when slot initialized, and also at action time
    • Common Error Response:
      – Set ESR, Post ESB, disable GPTM Slot
        » If Safing Initiated, or if Error initiating Safing, activating RTCS, or Setting/Resetting Event Flags
  – Prevent RTCS Nesting
    • Only activate RTCS if no other RTCS active for that SI
      – Slot will remain active if RTCS activation is delayed
  – Synchronize Monitoring to Telemetry
    • Optional first minor frame field added
      – Allow checking of subcommutated telemetry
      – Allows programming of slots for more efficient use of resources
  – Add two new tracking modes
    • Transition – only invokes RTCS or ESB on transition into action state
    • Continuous – continuously invokes RTCS or ESB while in action state
General Purpose Telemetry Monitor
Structure (initial, as modification of MFGPEF)

Ground Segment
- HST GS
- SNAS
- STScI Systems

Space Segment
- HST486 FSW
- Payload FSW

HST Mission Operations

Task Management
- Commands
- Engineering Data
- Science Data
- Diagnostics, Verification & Monitoring

MFGPEF

Key
- New
- Modified
- No Change
- Not Focus of Topic

10/19-21/2011

This presentation contains no US Export Controlled (ITAR) information
General Purpose Telemetry Monitor Structure (PDR)

Ground Segment
- HST GS
- SNAS
- STScI Systems

Space Segment
- HST486 FSW
- Payload FSW

Key:
- New
- Modified
- No Change
- Not Focus of Topic

Task Management
- Commands
- Engineering Data
- Science Data
- Diagnostics, Verification & Monitoring
- Science Instrument Application Processors

Scheduler Table

10/19-21/2011

This presentation contains no US Export Controlled (ITAR) information
General Purpose Telemetry Monitor Structure (final)

Ground Segment
- HST GS
- SNAS
- STScI Systems

Space Segment
- HST486 FSW
- Payload FSW

Key:
- New
- Modified
- No Change
- Not Focus of Topic

This presentation contains no US Export Controlled (ITAR) information

10/19-21/2011
Problems due to Requirements Creep

• Modifications were made to the design without identifying the changes to the requirements
• There was no step back to re-examine the overall design in light of the changed requirements
• Unit tests matched the original/flawed design
• Design Errors slipped through as a result
Design Errors

• Continuous tracking generated actions whether or not actions were triggered for the monitored location
  – Needed to move Event Flag code to a separate subroutine
    • Called from Main routine or Action Response subroutine
• Tracking could cause the consecutive counter to wrap
  – Caused actions to be unexpectedly triggered
  – Needed to set consecutive counter to 2 after tracking limits satisfied
• If in Tracking Mode an RTCS was not activated due to another RTCS active, the RTCS would never be activated
  – Needed to reset consecutive counter to zero after tracking limits were satisfied
Design Error Impacts

• This was a double impact to the schedule
  – Created much more work than originally estimated
  – Flight software had to be modified and retested late in the development process
  – Symbol Of Interest (mapping of FSW variables to physical memory addresses) had already been delivered for the Project Database

• Is a change to the SOI after delivery a big problem?
Yes!

Creating a new Project Database is an exacting process that requires a long lead time. Time we did not have.
Minimizing the Impact

• Adding the MFGPTM code to the end of memory minimized the impact to the SOI
  – Order of linking assembled NSSC-1 code to create an executable image is controlled by a linker
  – GPTM was linked into end of NSSC-1 memory, with control table first
    • No symbols used operationally were affected
  – This is standard procedure, when possible, to mitigate just such a problem
Team Standard

- Design
  - Write/update PDL
  - Peer review *
- Code and Unit Test
  - Write/update Code
  - Write/update Unit Test script
  - Run Unit Test
  - Peer review code and unit test *
- System Test
  - Write/update System Test
  - Dry run System Test
  - Peer review System Test *
  - Formal run of System Test
- Delivery of Symbol Of Interest
- Delivery of executable

As implemented

- Design
  - Write/update PDL
  - Peer review
    - minor updates requested
- Code and Unit Test
  - Write/update Code
  - Write/update Unit Test script
  - Run Unit Test
  - Peer review code and unit test
    - addition of new requirements change design
- System Test
  - Write/update System Test
  - Dry run System Test
  - Peer review System Test
    - Test deemed barely adequate, but no time to fix before SOI delivery
      - Formal run of System Test - passed
- Delivery of Symbol Of Interest
- More rigorous System Test – failed
- New Design and Code updates with peer reviews
- Rerun of new System Tests - passed
- Delivery of executable, and updated SOI

* Repeat above steps as needed
Lessons Learned

- Identify requirements changes implied by changes made during design reviews
- Review entire design in light of requirements changes
Uses of General Purpose Telemetry Monitor

- Can replace tasks that have been done by special purpose RTCSs
- Reduces need for Ground Operations monitoring of telemetry

- GPTM is currently executing with an empty Table until HST Operations personnel develop and validate the actual slot definitions to be used on-orbit

- GPTM Programming Examples:
  - Latching example: Autonomous Safing
  - Latching example: Accommodate Yellow and Red limit actions
    - Program 2 slots to look at same telemetry point
  - Tracking example: CMOS Single Bit Error Flag Reset
# GPTM Programming

## Example 1: Safe Payload if 1 MHz Clock Lost

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>ACTION:</th>
<th>TYPEFMF:</th>
<th>MINCOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a - SAFING</td>
<td>a - TRKMODE</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b - RTCS</td>
<td>b - FSTMNFRM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0777777 (-1)</td>
<td>0</td>
<td>No 1 MHz Clock flag telemetry word</td>
<td>000002 Monitor bit 2</td>
<td>000000</td>
<td>0400160</td>
<td>a=1 - Safing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>init slot</td>
<td>always load zero</td>
<td>No 1 MHz Clock flag</td>
<td>High limit = 0</td>
<td>Low limit = 0</td>
<td>Continuous</td>
<td>=0 1st mf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>always load zero</td>
<td>No 1 MHz Clock flag</td>
<td>000000</td>
<td>000000</td>
<td>0400160</td>
<td>a=0 Continuous</td>
<td>=0 1st mf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ye</td>
<td>0777777 (-1)</td>
<td>0</td>
<td>No 1 MHz Clock flag</td>
<td>000002 Monitor bit 2</td>
<td>000000</td>
<td>0400160</td>
<td>a=1 - Safing</td>
</tr>
<tr>
<td></td>
<td>Every 0.5 sec</td>
<td>init slot</td>
<td>always load zero</td>
<td>No 1 MHz Clock flag</td>
<td>000002 Monitor bit 2</td>
<td>000000</td>
<td>0400160</td>
<td>a=1 - Safing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No 1 MHz Clock flag</td>
<td>High limit = 0</td>
<td>Low limit = 0</td>
<td>Continuous</td>
<td>=0 1st mf</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a - SAFING</td>
<td>a - TRKMODE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b - RTCS</td>
<td>b - FSTMNFRM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c - ESB</td>
<td>c - ACTTYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>d - ESR</td>
<td>d - CHKTYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>e - SI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>f - EVNTFLAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a - SAFING</td>
<td>a - TRKMODE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b - RTCS</td>
<td>b - FSTMNFRM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c - ESB</td>
<td>c - ACTTYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>d - ESR</td>
<td>d - CHKTYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>e - SI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>f - EVNTFLAG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Initialize Payload Safing when “No 1 MHz Clock” flag is set 5 consecutive times
- Due to Latching, FREQ will be set to 0, disabling the slot until it is reset

Note: Values listed are “for example only”.

10/19-21/2011 This presentation contains no US Export Controlled (ITAR) information
# GPTM Programming

## Example 2: Turn Off SDF if Side B Temperature is Out of Limits

### Slot Programming

<table>
<thead>
<tr>
<th>Slot</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot</td>
<td>FREQ</td>
<td>FREQCNTR</td>
<td>CONSCNTR</td>
<td>LOCATION</td>
<td>MASK</td>
<td>HILIM</td>
<td>LOLIM</td>
<td>ACTION:</td>
<td>TYPEFMF:</td>
<td>MINCOUNT</td>
</tr>
<tr>
<td>n+1</td>
<td>024 (20)</td>
<td>0777777 (-1)</td>
<td>0</td>
<td>NSSC-1 address of SDF Side B Temperature telemetry</td>
<td>000377</td>
<td>0000342 Red</td>
<td>000052 Red</td>
<td>0217760</td>
<td>000025</td>
<td>Every 10 sec. Initialize slot</td>
</tr>
<tr>
<td>n+2</td>
<td>024 (20)</td>
<td>0777777 (-1)</td>
<td>0</td>
<td>NSSC-1 address of SDF Side B Temperature telemetry</td>
<td>000377</td>
<td>0000322 Yellow</td>
<td>000060 Yellow</td>
<td>000560</td>
<td>000025</td>
<td>Every 10 sec. Initialize slot</td>
</tr>
</tbody>
</table>

### Notes

- Programming two slots to monitor the same telemetry allows red and yellow limit actions
  - Slot n+2 Post an ESB only
  - Slot n+1 activates RTCS 143 to Turn Off SDF, sets ESR flag and Posts an ESB
- Due to Latching, FREQ will be set to 0, disabling the triggered slot until it is reset

Note: Values listed are “for example only”.

10/19-21/2011

This presentation contains no US Export Controlled (ITAR) information
### GPTM Programming

#### Example 3: Reset CMOS Single Bit Error Flag when Set

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ</td>
<td>FREQCNTR</td>
<td>CONSCNTR</td>
<td>LOCATION</td>
<td>MASK</td>
<td>HILIM (EU)</td>
<td>LOLIM (EU)</td>
<td>ACTION:</td>
<td>TYPEFMF:</td>
<td>MINCOUNT</td>
</tr>
<tr>
<td>024 (20)</td>
<td>0777777 (-1)</td>
<td>0</td>
<td>NSSC-1 address of CMOS Single Bit Error Flag telemetry word</td>
<td>000040</td>
<td>000000</td>
<td>0220160</td>
<td>a=0 - NoSafing b=144 - RTCS c-0 - NoPost d=0 - NoSet e=7 - SYSTEM f=0 - NoEF</td>
<td>a - TRKMODE b - FSTMNFRM c - ACTTYPE d - CHKTYPE</td>
<td>0</td>
</tr>
<tr>
<td>Every 10 sec.</td>
<td>Initialize slot</td>
<td>Always load 0</td>
<td>Monitor CMOS Single Bit Error Flag</td>
<td>000000</td>
<td>000000</td>
<td>0220160</td>
<td>a=0 - NoSafing b=144 - RTCS c-0 - NoPost d=0 - NoSet e=7 - SYSTEM f=0 - NoEF</td>
<td>a - TRKMODE b - FSTMNFRM c - ACTTYPE d - CHKTYPE</td>
<td>0</td>
</tr>
</tbody>
</table>

- When there is a Single Bit Error in NSSC-1 CMOS memory, the built-in EDAC function corrects the error and sets a flag in telemetry
- This GPTM slot will activate RTCS 144 to clear the SBE flag when it is set
  - Action Type = Tracking – the SBE telemetry is always monitored
  - Track Mode = Transition – the action takes place just when the SBE flag is set

Note: Values listed are “for example only”.

10/19-21/2011

This presentation contains no US Export Controlled (ITAR) information
Conclusion

• Design reuse has distinct advantages
  – Design is already tested and understood
  – A user knowledge base already exists
  – Test scripts already exist

• A reused design should be treated as new in terms of requirements analysis, and testing
  – Minor changes can have unexpected design implications

• Design for future maintainability
That’s a wrap.

Detail of one of the simulator boards in our NSSC-1 FSW Lab.
BACKUP SLIDES
Acronyms

ACS – Advanced Camera for Surveys (SI 1)
AP – Application Processor
BCU – Bus Coupler Unit
C&DH – Command & Data Handling
COS – Cosmic Origins Spectrograph (SI 4)
CPM – Central Processor Module
CU – Control Unit
ESB – Executive Status Buffer message
ESR – Executive Status Report flag
FSW – Flight Software
GPTM – General Purpose Telemetry Monitor
HST – Hubble Space Telescope
MOSES – Mission Operations, System Engineering & Software
NASA – National Aeronautics and Space Administration
NICMOS – Near Infrared Camera and Multi-Object Spectrometer (SI 2)
NSSC-1 – NASA Standard Spacecraft Computer, Model 1
PCU – Power Control Unit
PDL – Program Design Language
RIU – Remote Interface Unit
RM – Remote Module
RTCS – Relative Time Command Sequence
SAA – South Atlantic Anomaly
SI – Science Instrument
SDF – Science Data Formatter
SMM – Solar Maximum Mission
SSM – Support Systems Module
STINT – Standard Interface
STIS – Space Telescope Imaging Spectrograph (SI 3)
WFC3 – Wide Field Camera 3 (SI 5)
GPTM Code Modules

- **MFGPTM, General Purpose Telemetry Monitor (New)**
  - General Purpose Telemetry Monitor (GPTM) Application Processor
  - 569 Lines of code
- **MFGPTMAR, GPTM Action Response (New)**
  - Handles any action programmed into a GPTM control table slot
  - 294 Lines of code = 259 Lines in module + 35 Lines in calling macro
- **MFGPTMCT, GPTM Control Table (New)**
  - Contains the 32 slot programmable GPTM Control Table
    - GPTM Control Table will be delivered empty
  - 320 word table - 10 words per slot
- **MFGPTMEF, GPTM Event Flag (New)**
  - Controls the setting and clearing of event flags by the GPTM AP
  - 192 Lines of code = 157 Lines in module + 35 lines in calling macro
- **MFGPTMVS, GPTM Validate Slot (New)**
  - Examines GPTM Control Table slots
  - Disables a slot found to have coding errors
  - 320 Lines of code
- **MFGPTMER, GPTM Error Response (New)**
  - Posts an ESB message with a parameter containing an error return code in bits 10-7, and GPTM Control Table slot number in bits 6-1
  - 114 Lines of code = 75 Lines in module + 39 Lines in calling macro
- **MFUNIQUE, Mission Unique Data (Modified)**
  - Add Executive Processor 9, General Purpose Telemetry Monitor, to the Scheduler Table
  - 1405 Lines of existing code, 11 lines modified