COS DCE BOOT FSW v1.09 Component Test Results
Requirement 5.2.3.3a Counter for All Commands

Date: February 13, 2001
Document Number: COS-03-0027
Revision: Initial Release
Contract No.: NAS5-98043
CDRL No.: N/A

Prepared By: Tim Swanson, Software Test Engineer, Design_Net Eng. Date

Reviewed By: K. Brownsberger, COS Sr. Software Scientist, CU/CASA Date

Reviewed By: Grant Blue, COS Software & Operations Manager, BATC Date

Approved By: Barry Welsh, FUV Detector Program Manager, UCB Date

Approved By: John Andrews, COS Experiment Manager, CU/CASA Date

Center for Astrophysics & Space Astronomy
University of Colorado
Campus Box 593
Boulder, Colorado 80309
<table>
<thead>
<tr>
<th>Letter</th>
<th>ECO No.</th>
<th>Description</th>
<th>Check</th>
<th>Approved</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td></td>
<td>Initial Release</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THE UNIVERSITY OF COLORADO
At Boulder
The Center for Astrophysics and Space Astronomy

COS DCE BOOT FSW v1.09 Component Test Results
Requirement 5.2.3.3a Counter for All Commands

<table>
<thead>
<tr>
<th>Size</th>
<th>Code Indent No</th>
<th>Document No.</th>
<th>Rev</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>COS-03-0027</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

Scale: N/A
# Table of Contents

1. Introduction ................................................................................................................. 2
   1.1 Purpose .................................................................................................................... 2
   1.2 Scope ....................................................................................................................... 2
   1.3 Limitations and Constraints .................................................................................... 2
   1.4 Procedure Overview ................................................................................................. 2
   1.5 Theory of Test ......................................................................................................... 3
   1.6 Test Script Implementation ..................................................................................... 4
      1.6.1 Test Script Arguments ..................................................................................... 4
      1.6.2 Test Script Coding ........................................................................................... 4
2. Special Instructions ..................................................................................................... 4
   2.1 Quality Assurance ................................................................................................... 4
   2.2 Safety ....................................................................................................................... 4
      2.2.1 Personal Safety ................................................................................................ 4
      2.2.2 Test Article and Equipment Safety .................................................................. 5
   2.3 Contamination ......................................................................................................... 5
3. Support Requirements ................................................................................................. 5
   3.1 Personnel ................................................................................................................. 5
   3.2 Tools, Equipment, and Materials ............................................................................ 5
   3.3 Data/Software .......................................................................................................... 6
   3.4 Required Documentation ......................................................................................... 6
4. Procedure/Task Steps .................................................................................................. 6
   4.1 Pre-Operation Activities .......................................................................................... 6
      4.1.1 Make Sure that hks Tools Are Active ............................................................. 6
      4.1.2 Make Sure that the Proper ROM Is Installed .................................................. 7
      4.1.3 Log In to the EGSE ......................................................................................... 7
      4.1.4 Set Current Directory ...................................................................................... 7
      4.1.5 Slogin as eagcos .............................................................................................. 7
      4.1.6 Set Current Directory ...................................................................................... 8
      4.1.7 Ensure that Proper Files are Present............................................................... 8
   4.2 Operation Execution ................................................................................................ 8
      4.2.1 Establish Initial Test Conditions ..................................................................... 8
      4.2.2 Execute the Script ............................................................................................ 8
   4.3 Post-Operation Activities ...................................................................................... 11
      4.3.1 Copy Reports to PC Files and Print Them .................................................... 11
      4.3.2 Complete The Test Procedure Form ............................................................. 11
1. INTRODUCTION

1.1 PURPOSE

This document presents the Cosmic Origins Spectrograph (COS) Device Control Electronics (DCE) Flight Software (FSW) certification procedure. The purpose of this procedure is to verify that the FSW satisfies Software Requirements according to the method specified in the DCE FSW Test Plan (STP).

1.2 SCOPE

This test procedure comprises the steps necessary to verify that the FSW satisfies Software Requirements Document (SRD) paragraph 5.2.3.3 — Counters for All Commands.

1.3 LIMITATIONS AND CONSTRAINTS

This test cannot be run in parallel with any other commanding activity directed at the DCE FSW (such as, for example, the periodic transmission of NOOP commands). Test hardware shall be visually inspected, and its configuration noted, prior to conducting this test.

1.4 PROCEDURE OVERVIEW

The procedure requires the hks tools running on the Sun SparcStation Electronic Ground Support Equipment (EGSE) whose network IP address is one of

shorty.ssl.berkeley.edu
taiyo.ssl.berkeley.edu
ginger.ssl.berkeley.edu

Test time shall be scheduled in advance. The Test Conductor must be logged into the Unix system as user eagcos, and be commanding from the appropriate directory. This directory contains both the test script file and the shell script file; these two files control test execution. The test is conducted by invoking the shell script. This shell script in turn invokes the Perl 5 program UniScript.pl, which resides in its own distinct directory. The test procedure steps have been pre-recorded in the test script file, and are executed interpretively by the UniScript program. The shell script and test script are attached to this document as appendices. As UniScript executes the test script it sends results to the operator console and to two report files, which are also placed in the current directory. After completion of the test script, the Test Conductor can certify successful test.
execution by examining the contents of the report files and determining that required outputs are present in them. Printed copies of the report files are attached to the manually completed checklist (Paragraph 4 below) as documentation of the test.

1.5 THEORY OF TEST

The requirement is to “send all non-hazardous commands sequentially separated by 0.9 seconds each. Verify that Commands Received and Commands Executed counters agree, and are equal to the number of commands sent.” This test requirement cannot be met literally as stated for the following reasons:

1. Not all “non-hazardous” commands are accepted/executed in Boot State.
2. The "reset" commands LFDRSTP and LFDRSTW demolish the ongoing packet counts since they reset FSW variables.
3. The LFDJMPCS command takes FSW out of Boot State, hence leaves FSW in no condition to report success of the test in Boot State.
4. The LFDNOOP command is counted as neither “received” nor "executed" per software design.

Consequently, the script issues only the commands summarized in the following Table 1-1. After each command, the variables LFDCMDR and LFDCMDX are compared for equality with the hard-coded correct number of commands issued.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>ARGUMENTS</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFDCOPY</td>
<td>SOURCE =0xC000 DEST =0xC100 NBYTES =16 BANK =0</td>
<td>SOURCE and DEST are locations in 8051 “scratch” memory.</td>
</tr>
<tr>
<td>LFDCRC</td>
<td>SOURCE =0xC000 NBYTES =16 CODE =0</td>
<td>SOURCE is a location in 8051 “scratch” memory.</td>
</tr>
<tr>
<td>LFDDIAGC</td>
<td>SOURCE =0xC000 NBYTES =16</td>
<td>SOURCE is a location in 8051 “scratch” memory.</td>
</tr>
<tr>
<td>LFDDNLOD</td>
<td>SOURCE =0xC000 NBYTES =16</td>
<td>SOURCE is a location in 8051 “scratch” memory.</td>
</tr>
<tr>
<td>LFDGOTO</td>
<td>ISR23 =0x0023</td>
<td>Location of interrupt service routine.</td>
</tr>
<tr>
<td>LFDHKREQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFDMADDR</td>
<td>0 =index SOURCE =0xC000</td>
<td>0=index in LFDMONS/LFDMADD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOURCE is a location in 8051 “scratch”</td>
</tr>
</tbody>
</table>
1.6 TEST SCRIPT IMPLEMENTATION

1.6.1 Test Script Arguments

The script is parameterized as shown in the following Table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Correct Argument for Version 1.09</th>
</tr>
</thead>
<tbody>
<tr>
<td>#0</td>
<td>Absolute hex storage address of 8051 “scratch” area</td>
<td>C000</td>
</tr>
<tr>
<td>#1</td>
<td>Absolute hex storage address of another “scratch” area</td>
<td>C100</td>
</tr>
<tr>
<td>#2</td>
<td>Absolute hex storage address of unused ISR</td>
<td>0023</td>
</tr>
</tbody>
</table>

These parameters must be encoded into the shell script u (see Appendix A).

1.6.2 Test Script Coding

The script uses standard UniScript commands and directives.

2. SPECIAL INSTRUCTIONS

2.1 QUALITY ASSURANCE

QA support is required to verify the configuration and setup environment as well as monitoring test steps and verifying results.

2.2 SAFETY

2.2.1 Personal Safety
To ensure the safety of the test personnel during test execution the guidelines contained in Paragraph 3.4 Reference [1] will be adhered to.

2.2.2 Test Article and Equipment Safety

- If access within one (1) meter of COS bench electronics is necessary, wrist straps attached to technical ground shall be used by all personnel involved in handling of any COS test article. Overcurrent and overvoltage shall be set to remove power if nominal limits are exceeded.
- Emergency Power Shutdown — If, during the COS DCE FSW test, power is ON and a severe test equipment failure results in the power system exceeding specified limits, the Test Conductor shall direct or perform shutdown of power.

2.3 CONTAMINATION

All flight hardware shall be handled with clean latex gloves; it shall be covered with clean ESD material and/or stored in a clean flow-bench.

3. SUPPORT REQUIREMENTS

3.1 PERSONNEL

Execution of the COS DCE FSW certification procedure requires the following personnel (to be completed at the Test Readiness Review (TRR):

- Test Director: ____________________________________________
- Test Conductor: ____________________________________________
- Test Technician: ____________________________________________
- QA: ____________________________________________

3.2 TOOLS, EQUIPMENT, AND MATERIALS

The following is a list of tools, equipment, or materials required in this test. Record manufacturer and model, metrology, or property numbers of equipment used, where appropriate. Record calibration due dates where appropriate.

- Boot Mode ROM: schematic 27C256
Engineering Ground Support Equipment (see paragraph [1.4]). Indicate specific configuration:

<table>
<thead>
<tr>
<th>EGSE</th>
<th>DCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>taiyo shorty</td>
<td>ginger ETU</td>
</tr>
<tr>
<td>X</td>
<td>DCE #1 DCE #2</td>
</tr>
</tbody>
</table>

3.3 DATA/SOFTWARE

The following files must be present:

**Table 3-1: Required Program and Data Files**

<table>
<thead>
<tr>
<th>EGSE (shorty) Directory</th>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\disks\galex\users\galex\tcs\uniscript\</td>
<td>UniScript.pl</td>
<td>UniScript interpreter</td>
</tr>
<tr>
<td>\disks\galex\users\galex\tcs\uniscript\stp5_2_3_3a\</td>
<td>u</td>
<td>Shell script for this procedure</td>
</tr>
<tr>
<td>Ditto</td>
<td>stp5_2_3_3a.tst</td>
<td>Test script for this procedure [Appendix B]</td>
</tr>
</tbody>
</table>

In addition, the hks tools must be active. Directions for activating hks are given in UCB-COS-DOC-1118 (Paragraph [3.4] Reference [4]).

3.4 REQUIRED DOCUMENTATION

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document Number</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHB 1700.1(V1-A)</td>
<td><em>NASA Basic Safety Manual</em></td>
</tr>
<tr>
<td>2</td>
<td>COS-03-0027</td>
<td><em>DCE FSW Test Procedure 5.2.3.3a (this document)</em></td>
</tr>
<tr>
<td>3</td>
<td>UCB-COS-008</td>
<td><em>COS FUV Detector Software Test Plan</em></td>
</tr>
<tr>
<td>4</td>
<td>UCB-COS-DOC-1118</td>
<td><em>COS EGSE Startup Procedure</em></td>
</tr>
</tbody>
</table>

4. PROCEDURE/TASK STEPS

4.1 PRE-OPERATION ACTIVITIES

4.1.1 Make Sure that hks Tools Are Active

Follow the procedure given in Paragraph [3.4] Reference [4].
4.1.2 Make Sure that the Proper ROM Is Installed

Visually verify that the ROM under test is installed: if EEPROM, in U18: if PROM, in U2 and U7.

4.1.3 Log In to the EGSE

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C:	cs\us&gt; telnet shorty.ssl.berkeley.edu</td>
<td>Establish connection to shorty via Telnet client program</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Login: tcs Password:</td>
<td>Using telnet window, login as user tcs</td>
<td></td>
</tr>
</tbody>
</table>

4.1.4 Set Current Directory

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>tcs@shorty% cd ~galex/tcs tcs@shorty% pwd /disks/galex/users/galex/tcs</td>
<td>Change current directory as shown</td>
<td></td>
</tr>
</tbody>
</table>

4.1.5 Slogin as eagcos

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>tcs@shorty% slogin -l eagcos shorty.ssl.berkeley.edu <a href="mailto:eagcos@shorty.ssl.berkeley.edu">eagcos@shorty.ssl.berkeley.edu</a>’s password: (get from SSL personnel) Last login: Sat Oct 7 10:41:05 2000 from auntem.ssl.berke Sun Microsystems Inc. SunOS 5.8 Generic February 2000 You have mail. COS EGSE software version: devel</td>
<td>slogin as eagcos; get password from SSL personnel</td>
<td></td>
</tr>
</tbody>
</table>
## 4.1.6 Set Current Directory

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>eagcos:shorty% cd /disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a</td>
<td>Change current directory as shown</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eagcos:shorty% pwd /disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a</td>
<td></td>
</tr>
</tbody>
</table>

## 4.1.7 Ensure that Proper Files are Present

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>eagcos@shorty% ls –l</td>
<td>List files; the .tst file and the shell script u should be present</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total 12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-rw-r--r-- 1 tcs eag 1398 Oct 8 18:03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>stp5_2_3_3a.tst</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-rw-r--r-- 1 tcs eag 62 Oct 9 17:44 u</td>
<td></td>
</tr>
</tbody>
</table>

## 4.2 OPERATION EXECUTION

### 4.2.1 Establish Initial Test Conditions

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td>eagcos:shorty% set path=($path ~dbb/scripts/bin)</td>
<td>Set path as shown to enable access to hks tools</td>
</tr>
</tbody>
</table>

### 4.2.2 Execute the Script

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td>eagcos:taiy0% sh u $pstring=C000,C100,0023,0,0,0,0,0 Parameters are: Script File: stp5_2_3_3a</td>
<td>Shell to u. You should see the accompanying output as UniScript executes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#0: C000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#1: C100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#2: 0023</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#3: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#4: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#5: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#6: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>#7: 0</td>
<td></td>
</tr>
</tbody>
</table>
Report file
> /disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a
/stp5_2_3_3a.rp1
  successfully opened.
Report file
> /disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a
/stp5_2_3_3a.rp2
  successfully opened.
Script file
/disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a/s
tp5_2_3_3a.tst
  successfully opened at level 0.

"Press Y when ready to conduct test 5.2.3.3a"
Y
Continuing.
"Sending POR, collecting initial HK"

LFDNOOP

WAIT  0: HKV0=1; HKV1=135; wc=5
LFDNOOP

"Sending LFDCOPY"

LFDCOPY      SOURCE,DEST,NBYTES,BANK

WAIT  0: HKV0=3; HKV1=1; wc=5
WAIT  1: HKV1=2; wc=4
WAIT  1: HKV1=3; wc=3
"Sending LFMCRC"

LFDCRC       SOURCE,NBYTES,CODE

WAIT  0: HKV0=6; HKV1=4; wc=5

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Report file</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; /disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a/stp5_2_3_3a.rp1</td>
<td>successfully opened.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; /disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a/stp5_2_3_3a.rp2</td>
<td>successfully opened.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/disks/galex/users/galex/tcs/uniscript/stp5_2_3_3a/stp5_2_3_3a.tst</td>
<td>successfully opened at level 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Press Y when ready to conduct test 5.2.3.3a&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Continuing.</td>
<td>&quot;Sending POR, collecting initial HK&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDNOOP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT  0: HKV0=1; HKV1=135; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDNOOP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Sending LFDCOPY&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDCOPY      SOURCE,DEST,NBYTES,BANK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT  0: HKV0=3; HKV1=1; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT  1: HKV1=2; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT  1: HKV1=3; wc=3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Sending LFMCRC&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDCRC       SOURCE,NBYTES,CODE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT  0: HKV0=6; HKV1=4; wc=5</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>QA</td>
<td>Operator Entry/System Response</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=5; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=6; wc=3</td>
<td>&quot;Sending LFDDIAGC&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDDIAGC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=9; HKV1=7; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=8; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=9; wc=3</td>
<td>&quot;Sending LFDDNLOD&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDDNLOD SOURCE,NBYTES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=12; HKV1=11; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=11; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=12; wc=3</td>
<td>&quot;Sending LFDGOTO&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDGOTO ISR23</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=15; HKV1=13; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=14; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=15; wc=3</td>
<td>&quot;Sending LFDHKREQ&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDHKREQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=18; HKV1=16; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=17; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=18; wc=3</td>
<td>&quot;Sending LFDMADDR&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDMADDR 0,SOURCE,DATA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=21; HKV1=19; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=20; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=21; wc=3</td>
<td>&quot;Sending LFDUPLOD&quot;</td>
</tr>
</tbody>
</table>
### Requirement 5.2.3.3a Counter for All Commands

#### Step QA Operator Entry/System Response Description

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LFDUPLod  DEST,NBYTES,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=24; HKV1=22; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=23; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=24; wc=3</td>
<td>&quot;Sending LFDWDOG&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDWDOG 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=27; HKV1=25; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=26; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=27; wc=3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=27; HKV1=28; wc=5</td>
<td></td>
</tr>
</tbody>
</table>

#### 4.3 POST-OPERATION ACTIVITIES

**4.3.1 Copy Reports to PC Files and Print Them**

Using an FTP client, copy the \texttt{u, stp5\_2\_3\_3a.tst, stp5\_2\_3\_3a.rp1, and}\texttt{ stp5\_2\_3\_3a.rp2} files to appropriate PC files. Include these files as Appendices A, B, C, and D with this completed form.

**4.3.2 Complete The Test Procedure Form**

Ensure that all blank fields in this report are completed correctly and submit the completed report to QA.
SUMMARY SHEET

OPERATION TITLE: _____________________________ WOA# ______________

TEST ARTICLES IDENTIFICATION (including serial and/or part numbers):
____________________________________________________________________
____________________________________________________________________

TASKS/STEPS COMPLETED: __________________________________________
____________________________________________________________________

LOCATION: _________________________________________________________

TEST STARTED: TEST TERMINATED

TIME: _______ Hr/Min TIME: _______ Hr/Min

DATE: _______ DATE: _______

LOGS USED: ________________________________________________________

ANOMALY REPORTS GENERATED: ___________________________________
____________________________________________________________________

COMMENTS: ________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________

TEST CONDUCTOR: _________________________________________________

Signature/Date

QA REPRESENTATIVE:_______________________________________________

Signature/Date
Appendix A. Shell Script u

#!/bin/sh
pkill cosnoopy
perl ../UniScript.pl stp5_2_3_3a "C000,C100,0023,0,0,0,0,0"
cosnoopy&
Appendix B. Test Script stp5_2_3_3a.tst

```
; ****************************************************************************
; ****************************************************************************
; * ***
; ** BBBB OOO OOO TTTTT OOO N N L Y Y **
; ** B B B O O O T O O N N N L Y Y **
; ** B B B O O O T O O N N L Y Y **
; ** BBBB O O O O O T O O N N L Y **
; ** B B B O O O T O O N NN L Y **
; ** B B B O O O T O O N N L Y **
; ** BBBB O O O O T O O O N N L Y **
; **
; ****************************************************************************
; ****************************************************************************
; ****************************************************************************
; STP 5.2.3.3a
; ------------------------------------------------------------------------------ *
; Send all non-hazardous commands sequentially separated by 0.9 seconds each.
; Verify that Commands Received and Commands Executed counters agree, and are
; equal to the number of commands sent.
; * NOTE: This test requirement cannot be met literally as stated for the follow-
; ing reasons:
; * 1. Not all commands are accepted/executed in Boot State
; * 2. The "reset" commands LFDRSTP and LFDRSTW demolish the ongoing packet counts
; * since they reset FSW variables.
; * 3. The LFDJMPCS command takes FSW out of Boot State, hence leaves FSW in no
; * condition to report success of the test in Boot State.
; * 4. The LFDNOOP command is not counted as "executed" per software design.
; * Parameters:
; * #0 SOURCE ("scratch area")
; * #1 DEST ("scratch area + 256")
; * #2 ISR23 (address of LFDNOOP command routine)
; ****************************************************************************

SYM SOURCE =0x#0
SYM DEST =0x#1
SYM ISR23 =0x#2
SYM NBYTES =16
SYM CODE =0
SYM ID =0
SYM SRTTTING =0
SYM BANK =0
SYM RATE =0
SYM SEGMENT =0
SYM DIR =0
SYM POWER =0
SYM STATE =0
SYM HIVCLOT =0
SYM LIMIT =0
SYM VOLTAGE =0
SYM ACTUATOR =0
SYM ABORT =0
SYM OVERRIDE =0
SYM DOOR =0
SYM MOVE =0
SYM BANK =0
SYM DATA =0
SYM DELTA =90
SYM NSEC =5
SYM DIAG0031=0x0031

;LFDNOOP
```
;LFDRSTP
;LFDRSTW
;
; *****************************************************************************
; * Wait until setup (if any) is complete *
; ****************************************************************************
;
WTOR  "Press Y when ready to conduct test 5.2.3.3a"
;
; ****************************************************************************
; * Force Boot State *
; ****************************************************************************
;
DTG  3,"(0) Sending POR, collecting initial HK"
WTO  "Sending POR, collecting initial HK"
POR
DELAY DELTA
WAIT NSRC,HK
LOG 1,LFCTIME,LFDCMDX,LFDCMDR
LDN00P
;
DELAY DELTA
DTG  3,"(1) Sending LFDCOPY"
WTO  "Sending LFDCOPY"
LFDCOPY SOURCE,DEST,NBYTES,BANK
WAIT NSRC,HK
LOG 1,LFDCMDR,LFDCMDX
CHECK 1,($LFDCMDX == $LFDCMDR)
CHECK 1,($LFDCMDX == 1)
;
DELAY DELTA
DTG  3,"(2) Sending LFDCRC"
WTO  "Sending LFDCRC"
LFDCRC SOURCE,NBYTES,CODE
WAIT NSRC,HK
LOG 1,LFDCMDR,LFDCMDX
CHECK 1,($LFDCMDX == $LFDCMDR)
CHECK 1,($LFDCMDX == 2)
;
DELAY DELTA
DTG  3,"(3) Sending LFDDIAGC"
WTO  "Sending LFDDIAGC"
LFDDIAGC
WAIT NSRC,HK
LOG 1,LFDCMDR,LFDCMDX
CHECK 1,($LFDCMDX == $LFDCMDR)
CHECK 1,($LFDCMDX == 3)
;
DELAY DELTA
DTG  3,"(4) Sending LFDDNLOD"
WTO  "Sending LFDDNLOD"
LFDDNLOD SOURCE,NBYTES
WAIT NSRC,HK
LOG 1,LFDCMDR,LFDCMDX
CHECK 1,($LFDCMDX == $LFDCMDR)
CHECK 1,($LFDCMDX == 4)
;
DELAY DELTA
DTG  3,"(5) Sending LFDGOTO"
WTO  "Sending LFDGOTO"
LFDGOTO ISR23
WAIT NSRC,HK
LOG 1,LFDCMDR,LFDCMDX
DIAG 1,ANY,DIAG0031
CHECK 1,($LFDCMDX == $LFDCMDR)
CHECK 1,($LFDCMDX == 5)
;
DELAY DELTA
DTG  3,"(6) Sending LFDHKREQ"
WTO "Sending LFDHKREQ"
LFDHKREQ
WAIT NSC, HK
LOG 1, LFDCMDR, LFDCMDX
CHECK 1, ($LFDCMDX == $LFDCMDR)
CHECK 1, ($LFDCMDX == 6)
;
DELAY DELTA
DTG 3, "(7) Sending LFDMADDR"
WTO "Sending LFDMADDR"
LFDMADDR 0, SOURCE, DATA
WAIT NSC, HK
LOG 1, LFDCMDR, LFDCMDX
CHECK 1, ($LFDCMDX == $LFDCMDR)
CHECK 1, ($LFDCMDX == 7)
;
DELAY DELTA
DTG 3, "(8) Sending LFDUPLOD"
WTO "Sending LFDUPLOD"
LFDUPLOD DEST, NBYTES, 0
WAIT NSC, HK
LOG 1, LFDCMDR, LFDCMDX
CHECK 1, ($LFDCMDX == $LFDCMDR)
CHECK 1, ($LFDCMDX == 8)
;
DELAY DELTA
DTG 3, "(9) Sending LFDWDOG"
WTO "Sending LFDWDOG"
LFDMDOG 1
WAIT NSC, HK
LOG 1, LFDCMDR, LFDCMDX
CHECK 1, ($LFDCMDX == $LFDCMDR)
CHECK 1, ($LFDCMDX == 9)
;
DELAY DELTA
DTG 3, "(10) Sending LFDJMPCS"
WTO "Sending LFDMPCS"
LFDMPCS 0

; -----------------------------------------
; LFGBWK SETTING, SEGMENT, 0
; LFGEWK SETTING, SEGMENT, 0
; LFQLT SETTING, SEGMENT
; LFQHSFT SETTING, SEGMENT, 0
; LFQSTIM RATE, SEGMENT
; LFQSTR SETTING, SEGMENT, 0
; LFQT SETTING, SEGMENT, DIR
; LFQT SETTING, SEGMENT
; LFQWR 0
; LFQRAMPT RATE
; LFQSTATE STATE
; LFQVENA HIVAL
; LFQVILIM LIMIT
; LFQVLOW VOLTAGE, SEGMENT
; LFQVMAX VOLTAGE, SEGMENT
; LFQVOM VOLTAGE, SEGMENT
; LFQVPWR POWER
; LFQVSET VOLTAGE, 0
; LFQCPR INTERVAL, SEGMENT, COUNT
; LFQRACT1 POWER
; LFQRACT2 POWER
; LFQRACTEN ACTUATOR
; LFQRACTRS 0
; LFQRPWR POWER
; LFQRLIM LIMIT
; LFQRSOVD OVERRIDE
; LFQRDNR DIR
; LFQMDNA DOOR
; LFQMPWR MOVE
;
WAIT NSC, HK
LOG 1,LFCTIME,LFDCMDX,LFDCMDR
CHECK 1,((LFDCMDR==LFDCMDX) && (LFDCMDX==9))
;
DTG 1, "(10) Test stp5.2.3.3a completed successfully"
## Appendix C. Test Report stp5_2_3_3a.rp1

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Test Description</th>
<th>Value1</th>
<th>Value2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.3.3a</td>
<td>Counter for All Commands</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Ver 01.09 Fri Dec 1 00:52:52 2000 "(0) Sending POR, collecting initial HK"

```
LFDNOOP
Addr Addr HK-Name Value
----- ---- -------------- ----- 
1680-1683 LFCTIME 00000149 
170C-170D LFDCMDX 0001 
1718-1719 LFDCMDR 0001 
```

### Ver 01.09 Fri Dec 1 00:52:54 2000 "(1) Sending LFDCOPY"

```
LFDCOPY SOURCE,DEST,NBYTES,BANK
Addr Addr HK-Name Value
----- ---- -------------- ----- 
1718-1719 LFDCMDR 0001 
170C-170D LFDCMDX 0001 
```

CHECK: ($LFDCMDX == $LFDCMDR)  
eval:  (0001 == 0001)  
SUCCESS

CHECK: ($LFDCMDX == 1)  
eval:  (0001 == 1)  
SUCCESS

### Ver 01.09 Fri Dec 1 00:52:57 2000 "(2) Sending LFDCRC"

```
LFDCRC SOURCE,NBYTES,CODE
Addr Addr HK-Name Value
----- ---- -------------- ----- 
1718-1719 LFDCMDR 0002 
170C-170D LFDCMDX 0002 
```

CHECK: ($LFDCMDX == $LFDCMDR)  
eval:  (0002 == 0002)  
SUCCESS

CHECK: ($LFDCMDX == 2)  
eval:  (0002 == 2)  
SUCCESS

### Ver 01.09 Fri Dec 1 00:53:01 2000 "(3) Sending LFDDIAGC"

```
LFDDIAGC 
```
## Requirement 5.2.3.3a Counter for All Commands

### Ver 01.09 Fri Dec 1 00:53:04 2000

"(4) Sending LFDDNLQOD"

<table>
<thead>
<tr>
<th>Addr</th>
<th>Addr</th>
<th>HK-Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1718-1719</td>
<td>LFDCMDR</td>
<td>0003</td>
<td></td>
</tr>
<tr>
<td>170C-170D</td>
<td>LFDCMDX</td>
<td>0003</td>
<td></td>
</tr>
</tbody>
</table>

CHECK: ($LFDCMDX == $LFDCMDR)
eval: (0003 == 0003)
SUCCESS

CHECK: ($LFDCMDX == 3)
eval: (0003 == 3)
SUCCESS

### Ver 01.09 Fri Dec 1 00:53:07 2000

"(5) Sending LFDDGOTO"

<table>
<thead>
<tr>
<th>Addr</th>
<th>Addr</th>
<th>HK-Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1718-1719</td>
<td>LFDCMDR</td>
<td>0004</td>
<td></td>
</tr>
<tr>
<td>170C-170D</td>
<td>LFDCMDX</td>
<td>0004</td>
<td></td>
</tr>
</tbody>
</table>

CHECK: ($LFDCMDX == $LFDCMDR)
eval: (0004 == 0004)
SUCCESS

CHECK: ($LFDCMDX == 4)
eval: (0004 == 4)
SUCCESS

### Ver 01.09 Fri Dec 1 00:53:10 2000

"(6) Sending LFDDHKREQ"

<table>
<thead>
<tr>
<th>Addr</th>
<th>Addr</th>
<th>HK-Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1718-1719</td>
<td>LFDCMDR</td>
<td>0005</td>
<td></td>
</tr>
<tr>
<td>170C-170D</td>
<td>LFDCMDX</td>
<td>0005</td>
<td></td>
</tr>
</tbody>
</table>

CHECK: ($LFDCMDX == $LFDCMDR)
eval: (0005 == 0005)
SUCCESS

CHECK: ($LFDCMDX == 5)
eval: (0005 == 5)
SUCCESS
SUCCESS
CHECK:  ($LFDCMDX == 6)
eval:   (0006 == 6)

SUCCESS
Ver 01.09 Fri Dec 1 00:53:13 2000  "(7) Sending LFDMADDR"
LFDMADDR  0,SOURCE,DATA
Addr Addr HK-Name Value
---- ---- -------------- -----  
1718-1719 LFDCMDR 0007  
170C-170D LFDCMDX 0007  
CHECK:  ($LFDCMDX == $LFDCMDR)
eval:   (0007 == 0007)

SUCCESS
CHECK:  ($LFDCMDX == 7)
eval:   (0007 == 7)

SUCCESS
Ver 01.09 Fri Dec 1 00:53:16 2000  "(8) Sending LFDUPLOD"
LFDUPLOD  DEST,NBYTES,0
Addr Addr HK-Name Value
---- ---- -------------- -----  
1718-1719 LFDCMDR 0008  
170C-170D LFDCMDX 0008  
CHECK:  ($LFDCMDX == $LFDCMDR)
eval:   (0008 == 0008)

SUCCESS
CHECK:  ($LFDCMDX == 8)
eval:   (0008 == 8)

SUCCESS
Ver 01.09 Fri Dec 1 00:53:20 2000  "(9) Sending LFDWDOG"
LFDWDOG  1
Addr Addr HK-Name Value
---- ---- -------------- -----  
1718-1719 LFDCMDR 0009  
170C-170D LFDCMDX 0009  
CHECK:  ($LFDCMDX == $LFDCMDR)
eval:   (0009 == 0009)

SUCCESS
CHECK:  ($LFDCMDX == 9)
eval:   (0009 == 9)

SUCCESS
Addr Addr HK-Name Value
---- ---- -------------- -----  
1680-1683 LFCTIME 0000001C  
170C-170D LFDCMDX 0009  
1718-1719 LFDCMDR 0009  

COS DCE BOOT FSW v1.09 Component Test Results
Requirement 5.2.3.3a Counter for All Commands
CHECK: (($LFDCMDR==$LFDCMDX) && ($LFDCMDX==9))
eval:  ((0009==0009) && (0009==9))
        SUCCESS

Ver 01.09 Fri Dec  1 00:53:22 2000 "(10) Test stp5.2.3.3a completed successfully"
Appendix D. Test Report stp5_2_3_3a.rp2

Ver 01.09 Fri Dec 1 00:52:52 2000 "(0) Sending POR, collecting initial HK"
Ver 01.09 Fri Dec 1 00:52:54 2000 "(1) Sending LFDCOPY"
Ver 01.09 Fri Dec 1 00:52:57 2000 "(2) Sending LFDCRC"
Ver 01.09 Fri Dec 1 00:53:01 2000 "(3) Sending LFDDIAGC"
Ver 01.09 Fri Dec 1 00:53:04 2000 "(4) Sending LFDDNLOD"
Ver 01.09 Fri Dec 1 00:53:07 2000 "(5) Sending LFDGOTO"
Ver 01.09 Fri Dec 1 00:53:08 2000 "(6) Sending LFDDHREQ"
Ver 01.09 Fri Dec 1 00:53:12 2000 "(7) Sending LFDMADDR"
Ver 01.09 Fri Dec 1 00:53:16 2000 "(8) Sending LFDDUPLOD"
Ver 01.09 Fri Dec 1 00:53:20 2000 "(9) Sending LFDWDOG"